

Computer lab A

Instructions

- The lab is assumed to be done in groups.
- Create a report to the lab solutions in PDF.
- Be concise and do not include unnecessary printouts and figures produced by the software and not required in the assignments.
- **Include all your codes as an appendix into your report.**
- A typical lab report should 2-4 pages of text plus some number of figures plus appendix with codes.
- The group lab report should be submitted via LISAM before the deadline specified in LISAM.
- **Use 12345 as a random seed everywhere where the result of the simulation differs with the run unless stated otherwise.**

Assignment 1. Computations with simulated data

- a) Generate two time series $x_t = -0.8x_{t-2} + w_t$, where $x_0 = x_1 = 0$ and $x_t = \cos\left(\frac{2\pi t}{5}\right)$ with 100 observations each. Apply a smoothing filter $v_t = 0.2(x_t + x_{t-1} + x_{t-2} + x_{t-3} + x_{t-4})$ to these two series and compare how the filter has affected them.
- b) Consider time series $x_t - 4x_{t-1} + 2x_{t-2} + x_{t-5} = w_t + 3w_{t-2} + w_{t-4} - 4w_{t-6}$. Write an appropriate R code to investigate whether this time series is causal and invertible.
- c) Use built-in R functions to simulate 100 observations from the process $x_t + \frac{3}{4}x_{t-1} = w_t - \frac{1}{9}w_{t-2}$, compute sample ACF and theoretical ACF, use seed 54321. Compare the ACF plots.

Assignment 2. Visualization, detrending and residual analysis of Rhine data.

The data set **Rhine.csv** contains monthly concentrations of total nitrogen in the Rhine River in the period 1989-2002.

- a) Import the data to R, convert it appropriately to *ts* object (use function *ts()*) and explore it by plotting the time series, creating scatter plots of x_t against x_{t-1}, \dots, x_{t-12} . Analyze the time series plot and the scatter plots: Are there any trends, linear or seasonal, in the time series? When during the year is the concentration highest? Are there any special patterns in the data or scatterplots? Does the variance seem to change over time? Which variables in the scatterplots seem to have a significant relation to each other?
- b) Eliminate the trend by fitting a linear model with respect to t to the time series. Is there a significant time trend? Look at the residual pattern and the sample ACF of the residuals and comment how this pattern might be related to seasonality of the series.
- c) Eliminate the trend by fitting a kernel smoother with respect to t to the time series (choose a reasonable bandwidth yourself so the fit looks reasonable). Analyze the residual pattern and the sample ACF of the residuals and compare it to the ACF from step b). Conclusions? Do residuals seem to represent a stationary series?
- d) Eliminate the trend by fitting the following so-called seasonal means model:
$$x_t = \alpha_0 + \alpha_1 t + \beta_1 I(month = 2) + \dots + \beta_{12} I(month = 12) + w_t,$$

- where $I(x) = 1$ if x is true and 0 otherwise. Fitting of this model will require you to augment data with a categorical variable showing the current month, and then fitting a usual linear regression.
- Analyze the residual pattern and the ACF of residuals.
- e) Perform stepwise variable selection in model from step d). Which model gives you the lowest AIC value? Which variables are left in the model?

Assignment 3. Analysis of oil and gas time series.

Weekly time series *oil* and *gas* present in the package *astsa* show the oil prices in dollars per barrel and gas prices in cents per dollar.

- a) Plot the given time series in the same graph. Do they look like stationary series? Do the processes seem to be related to each other? Motivate your answer.
- b) Apply log-transform to the time series and plot the transformed data. In what respect did this transformation make the data easier for the analysis?
- c) To eliminate trend, compute the first difference of the transformed data, plot the detrended series, check their ACFs and analyze the obtained plots. Denote the data obtained here as x_t (oil) and y_t (gas).
- d) Exhibit scatterplots of x_t and y_t for up to three weeks of lead time of x_t ; include a nonparametric smoother in each plot and comment the results: are there outliers? Are the relationships linear? Are there changes in the trend?
- e) Fit the following model: $y_t = \alpha_0 + \alpha_1 I(x_t > 0) + \beta_1 x_t + \beta_2 x_{t-1} + w_t$ and check which coefficients seem to be significant. How can this be interpreted? Analyze the residual pattern and the ACF of the residuals.

Time Series Analysis Lab 1

Omkar Bhutra (omkbh878), Nadezhda Green (nadgr258)

18 September 2019

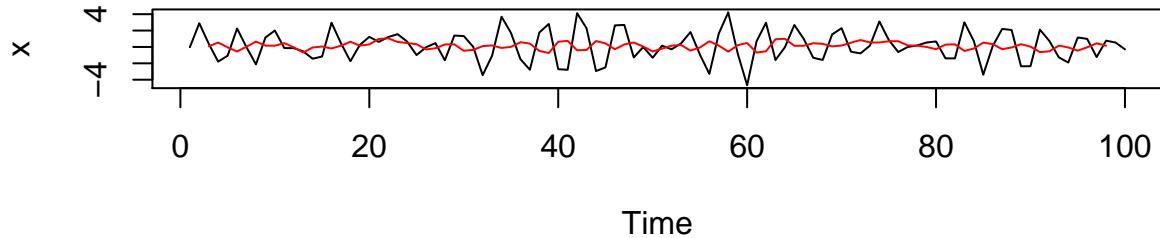
Assignment 1

a)

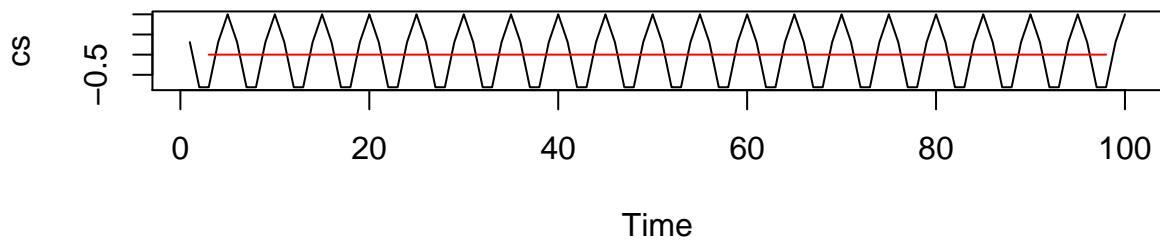
For the first time series, we can say that the filter has made the time series more smooth. Further, the series has less variation on the horizontal axis. We can see in the function of the filtering that it takes the average of neighbour observations in the past. Regarding the cosine time series, we can see that smoothing filter makes the time series completely horizontal. The reason for this is that the cosine series have the same pattern as a function of time and if we do not use an even number of lags in the smoothing filter we will always get a straight line.

```
set.seed(12345)
par(mfrow=c(2,1), cex.main=1.5)
#First Time Series
w = rnorm(110,0,1) #10 extra to avoid startup problems
x = stats::filter(w, filter = c(0,-0.8), method="recursive")[-(1:10)] # remove first 10
plot.ts(x, main = "-0.8x_t-2 + w_t , smoothing in red")
v1 = stats::filter(x,sides = 2, filter = rep(0.2,5))
lines(v1, type = "l", col = "red")
#Second Time Series
cs = cos(2*pi*1:100/5); w = rnorm(100,0,1)
v2 = stats::filter(cs,sides = 2, filter = rep(0.2,5))
plot.ts(cs, main = expression("cos(2*pi*t/5),smoothing in red"))
lines(v2, type = "l", col = "red")
```

-0.8x_t-2 + w_t , smoothing in red



cos(2*pi*t/5),smoothing in red



The first time series is quite irregular with large fluctuations in values and looks choppy, the smoothing filter takes the average of 5 the previous points of the past and makes its smoother and also closer to the centerline. The second time series is a cosine function, the cosine wave has regular patterns and the smoothing filter completely reduces the series to 0 which is seen as the horizontal line. If odd number of lags are used in the smoothing filter this will be the result.

b)

Given the following equation with the autoregressive function on the left hand side and the moving average on the right hand side. In order to check if the series is causal and invertible we have to check the polynomial root of the series. This is to see that if any of the resulting parameters are inside the unit circle then they are non causal and non invertible.

$$x_t - 4x_{t-1} + 2x_{t-2} + x_{t-5} = w_t + 3wt - 2 + w_{t-4} - 4w_{t-6}$$

Simplified:

$$(1 - 4B + 2B^2 + B^5)x_t = (1 + 3B^2 + B^4 - 4B^6)w_t$$

We have to check if all of the parameters satisfy the constraint:

$$\sqrt{(R^2 + I^2)}$$

where R is the real number and I is the imaginary number.

The process is not invertible and not causal.

```
#Check Invertibility
Invertible <- polyroot(c(1,0,3,0,1,0,4))
#Check causality
Causal <- polyroot(c(1,-4,2,0,0,1))
rootFun <- function(y){
  res <- c()
  for(i in 1:length(y)){
    res[i] <- sqrt(Im(y[i])^2 + Re(y[i])^2)
  }
  res <- as.numeric(res)
  if(any(res<1)){
    return("Non Causal/Non Invertible")
  } else {
    return("Causal/Invertible")
  }
}
root1 <- rootFun(Invertible)
root2 <- rootFun(Causal)
root1

## [1] "Non Causal/Non Invertible"

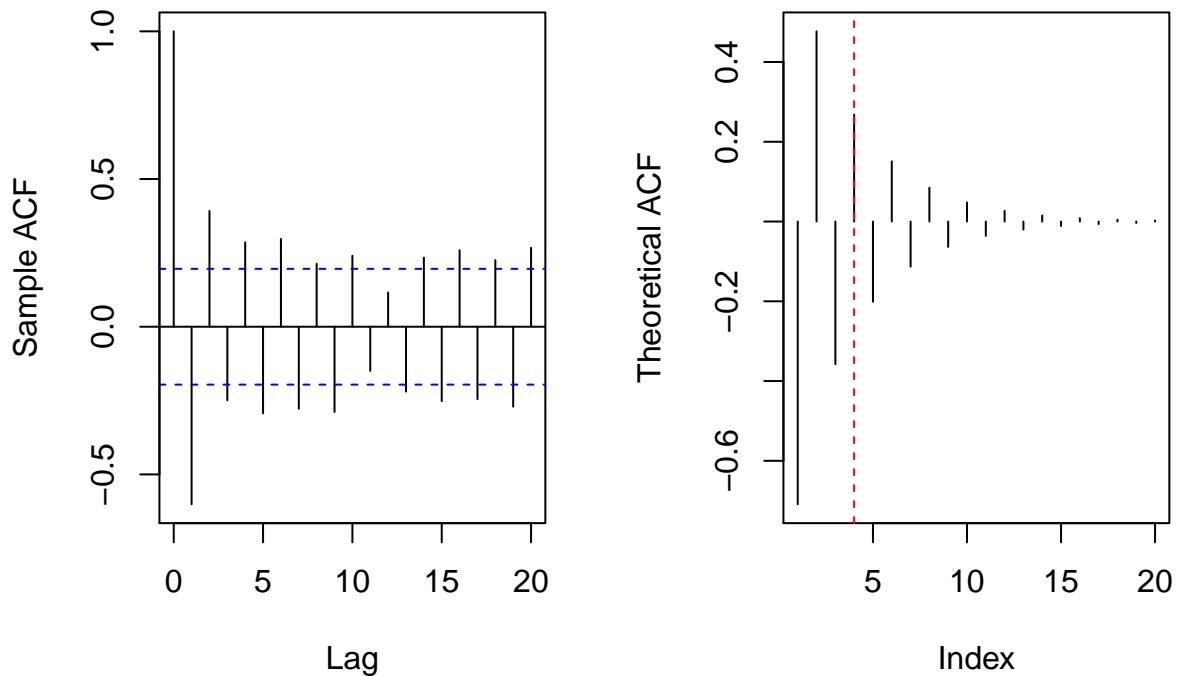
root2

## [1] "Non Causal/Non Invertible"

c)

set.seed(54321)
#model
arimasimmodel<-arima.sim(model = list(ar = -3/4, ma = c(0,-1/9)), n = 100 )
#plotting
par(mfrow = c(1,2))
acftest<-acf(arimasimmodel,ylab="Sample ACF")
arimaacfobj<-ARMAacf(ar = c(-3/4), ma = c(0,-1/9), lag.max=20)[-1]
plot(arimaacfobj , type = "n",ylab="Theoretical ACF")
lines(arimaacfobj , type = "h")
abline(col = "red", v = 4, lty = 2)
```

Series arimasimmodel



Difference between the theoretical ACF and the sample ACF is that the autocorrelations in the theoretical ACF plot are given by a recursive filter when our $lag > 3$.

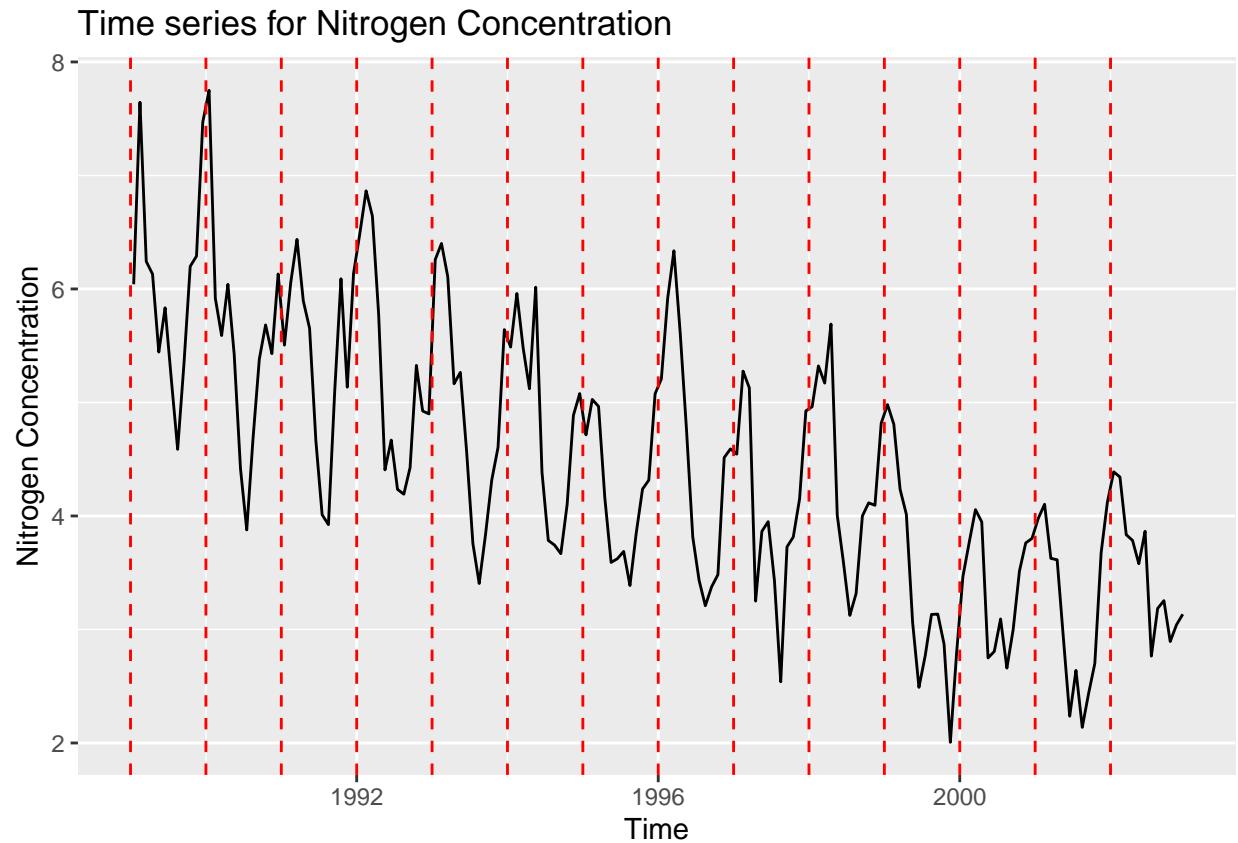
Assignment 2

a)

From the plot of the time series, it is observed that there is a general decreasing trend over the years observed and also a regular seasonality where the N conc. rises sharply towards the end of the year peaking at the beginning of the next year and the falls sharply. The variance does not seem to change over time. This is confirmed from the scatterplots where it is seen that there is a linear trend in the first and last months of the year.

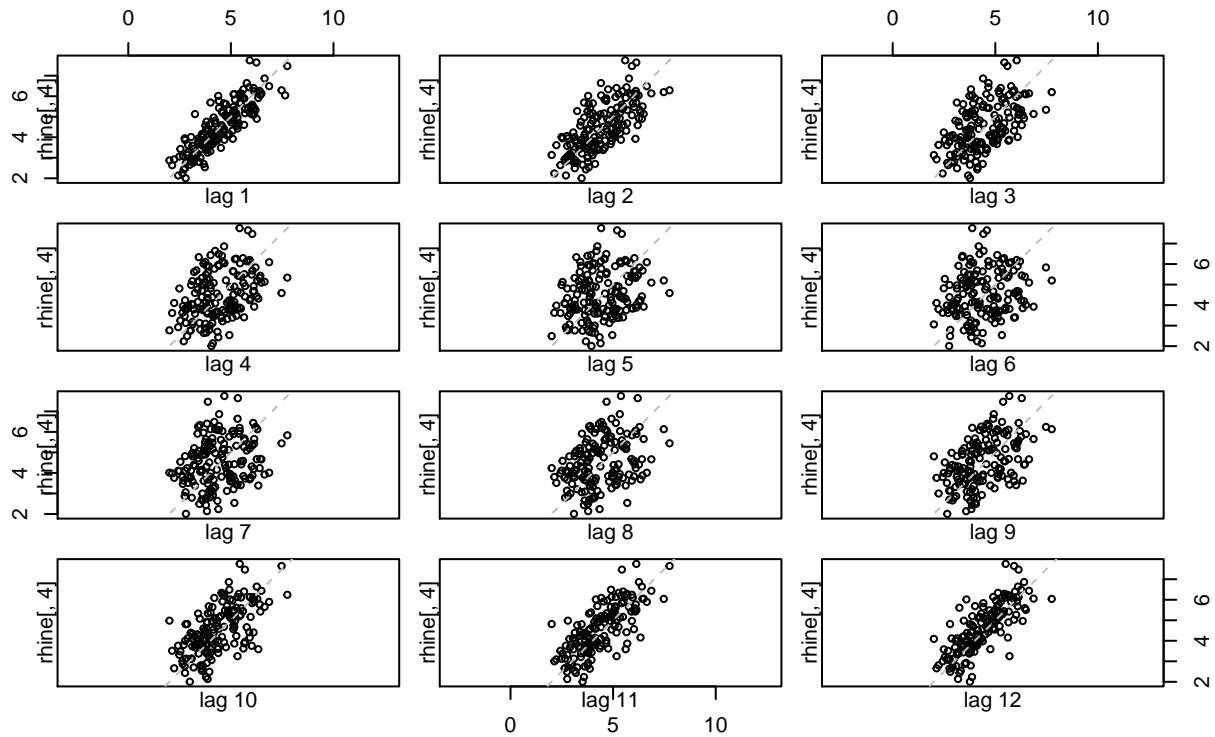
```
rhine <- ts(read.csv2("C:/Users/Omkar/Documents/Rhine.csv"))
rhinedf <- as.data.frame( read.csv2("C:/Users/Omkar/Documents/Rhine.csv") )

ggplot(rhinedf, aes(x = rhinedf$Time)) + geom_line(aes(y = rhinedf$TotN_conc)) + geom_vline(aes(xintercept =
```



```
lag.plot(x = rhine[,4], lag = 12, main = "Scatterplots of different lags of Nitrogen composition" )
```

Scatterplots of different lags of Nitrogen composition



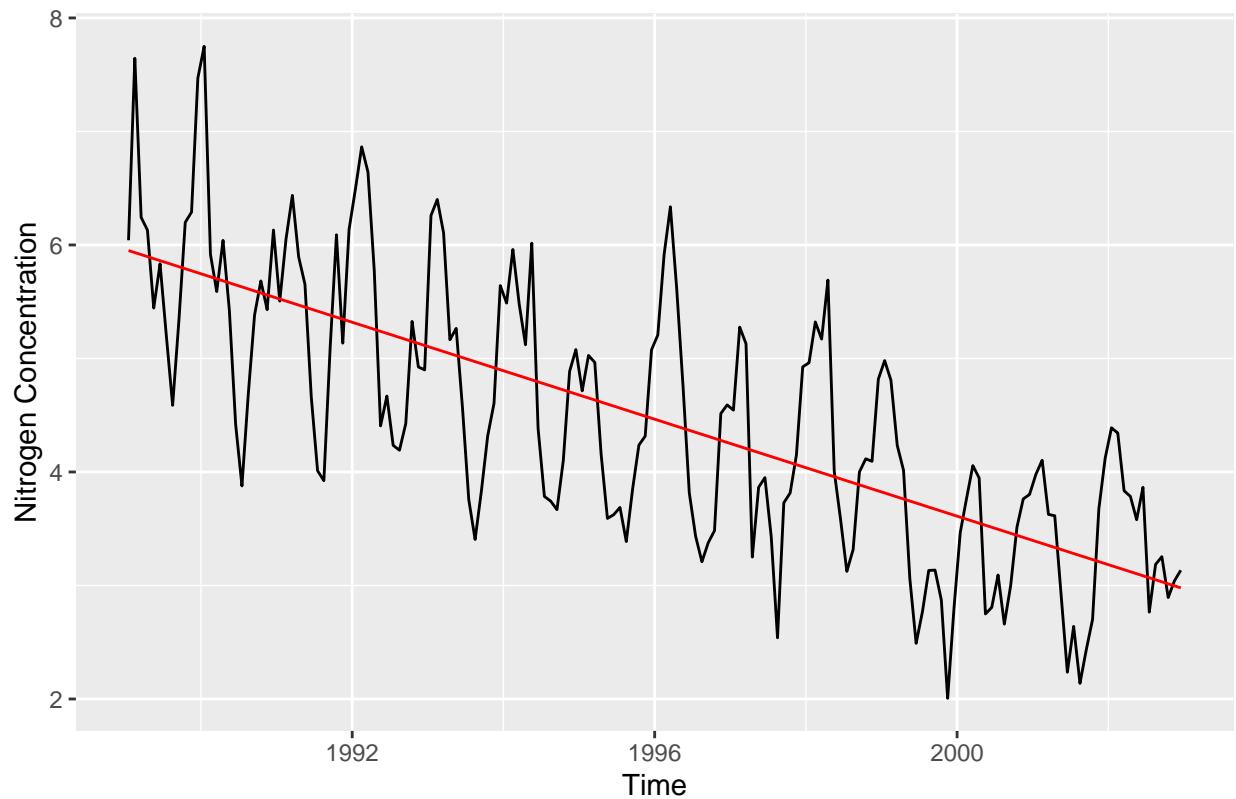
b)

The fitted values in figure show that there is a decreasing trend in Nitrogen Concentration over time. For each month the slope has a significant trend of approximately -0.2 Nitrogen concentration units. On observation of the ACF we can see that the autocorrelation has a recurring pattern every year which is sign of seasonality. By looking at figure we can see the seasonality in the difference between the horizontal line at 0 and the data.

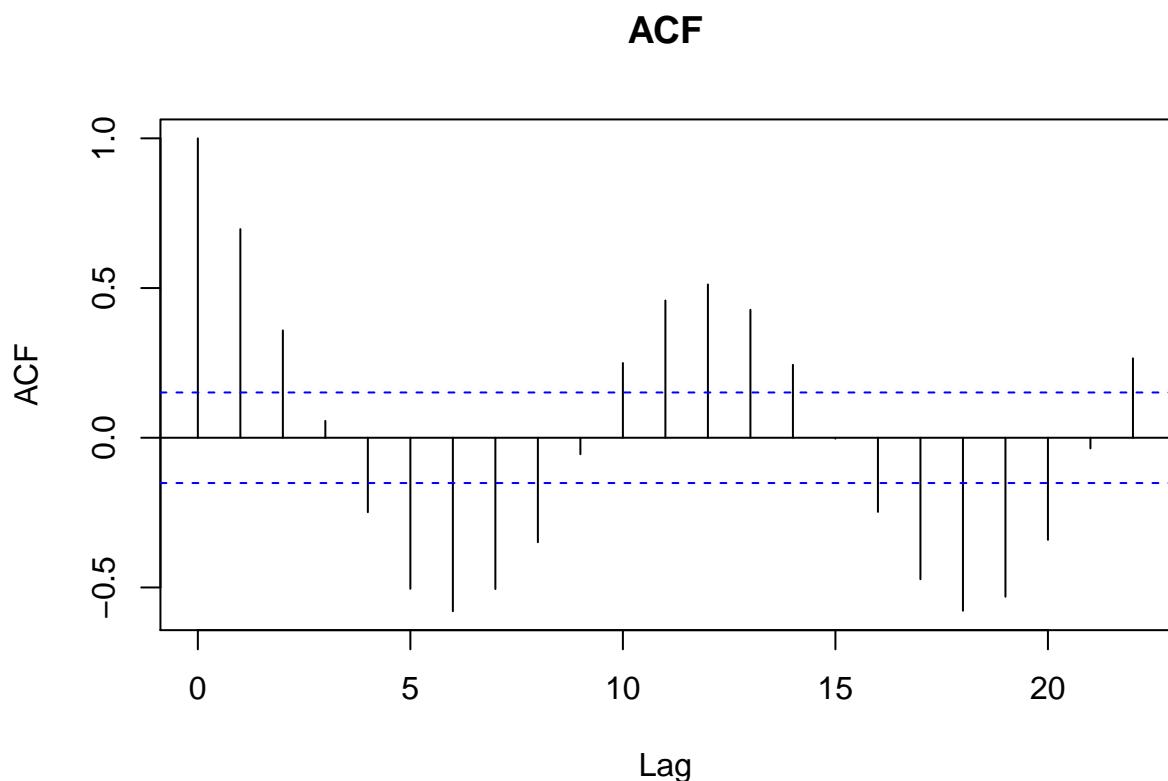
```
#Eliminating the trend by fitting a linear model
linear.model <- lm(TotN_conc ~ Time, data = rhinedf)
```

```
ggplot(rhinedf, aes(x = rhinedf$Time)) + geom_line(aes(y = rhinedf$TotN_conc)) + geom_line(aes(y = linear.mod
```

Linear trend elimination and ACF of residuals



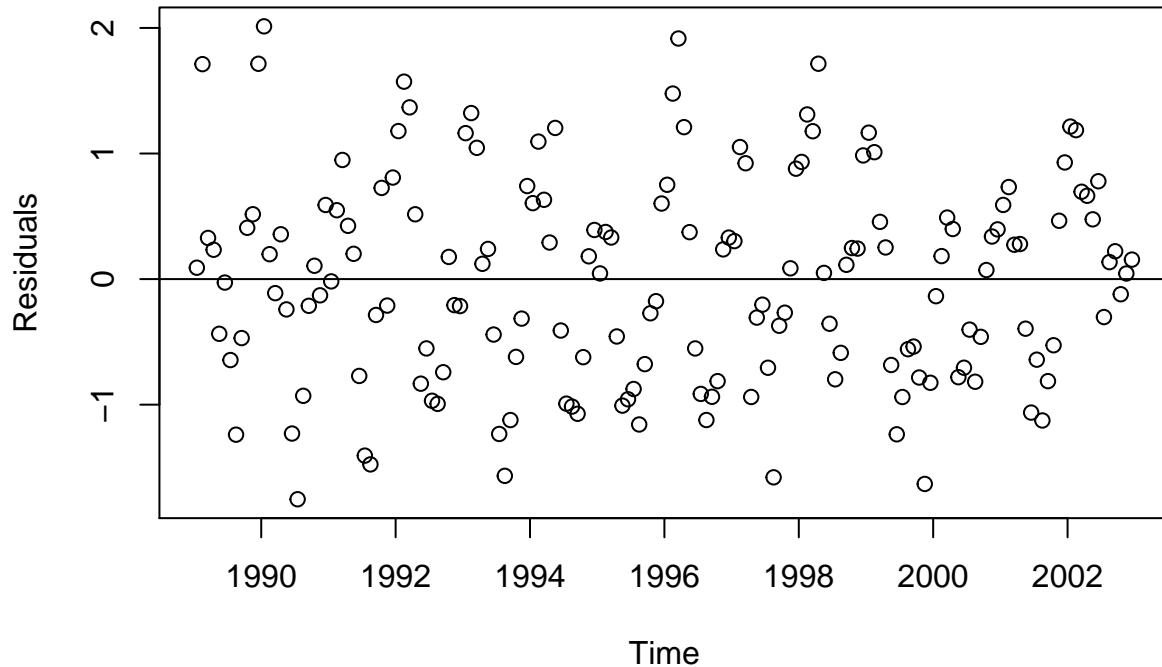
```
#Plot ACF
acf(linear.model$residuals, main = "ACF")
```



```
#Plot Residuals
plot(linear.model$residuals, main = "Residual plot",
      x = linear.model$model$Time, ylab = "Residuals",
      xlab = "Time", type = "p")

abline(h = 0)
```

Residual plot



```
summaryModel <- summary(linear.model)
```

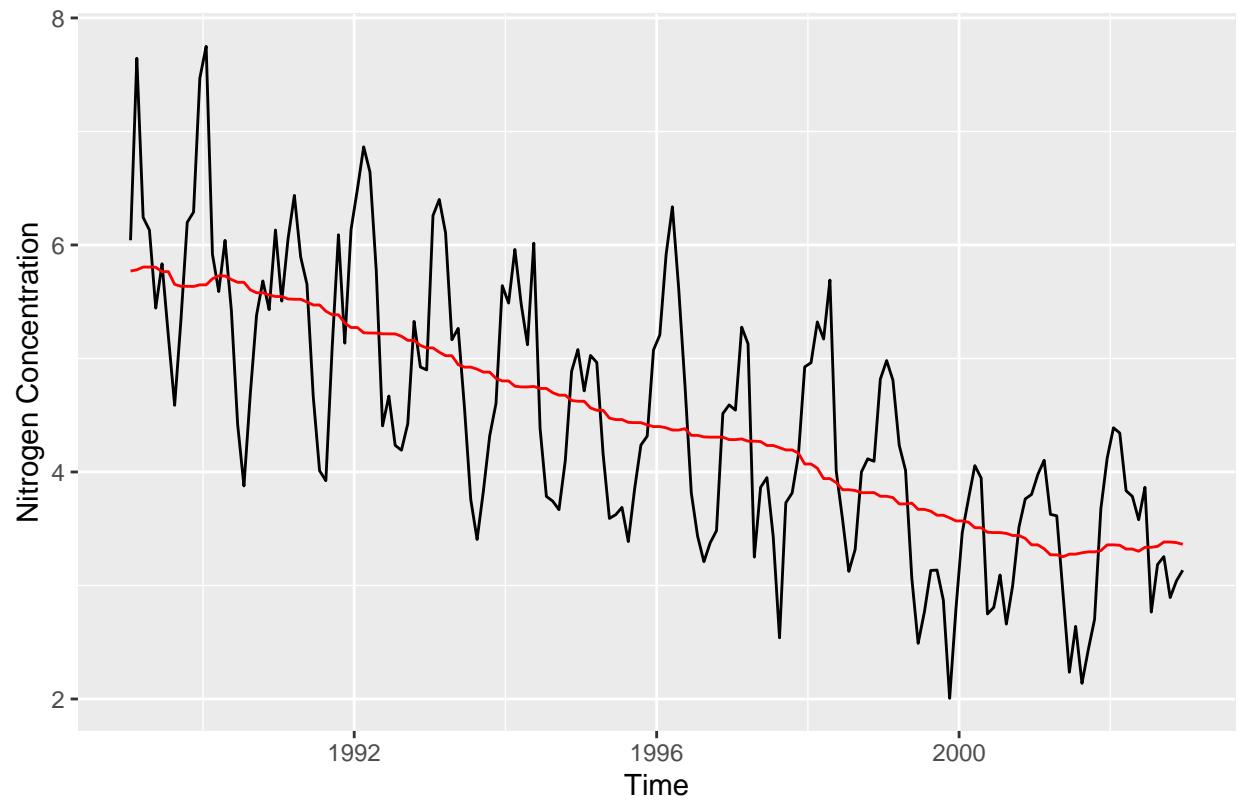
c)

In Figure 1 we have compared the residual pattern from the linear model, kernel smoother with bandwidth 4 and kernel smoother with bandwidth 20. As we increase the bandwidth the fitted line becomes underfitted and the residual pattern changes. The residuals does not seem to be stationary in the models since they seem to follow a pattern that is dependent on time which we can see in Figure 1, this can be interpreted as a seasonal effect.

```
#Eliminating the trend by fitting a kernel smoother
kernel.model <- ksmooth(x = rhine[, 3], y = rhine[, 4], bandwidth=4)
kernel.model20 <- ksmooth(x = rhine[, 3], y = rhine[, 4], bandwidth=20)
residuals.kernel <- rhine[, 4] - kernel.model$y
residuals.kernel20 <- rhine[, 4] - kernel.model20$y

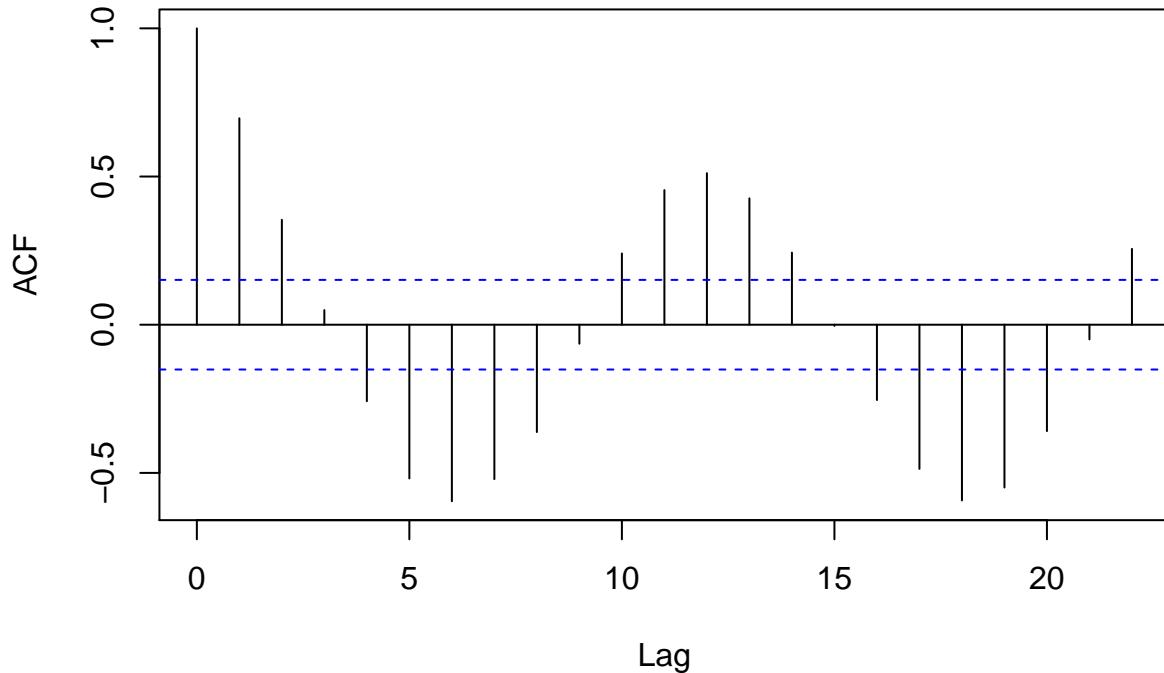
ggplot(rhinedf, aes(x = rhinedf$Time)) + geom_line(aes(y = rhinedf$TotN_conc)) + geom_line(aes(y = kernel.mod
```

Fitted values with kernel smoother vs Data



```
#ACF plot for residuals of kernel smoother  
acf(residuals.kernel, main = "ACF for residuals from kernel smoother with B = 4")
```

ACF for residuals from kernel smoother with B = 4



```
#Extract & plot residuals from kernel
plot(y = residuals.kernel, x = rhine[,3], type = "l",
      main = "Residuals from kernel smoother",
      xlab = "Time", ylab = "Residuals")
lines(linear.model$residuals,
      x = linear.model$model$Time, xlab = "Time",
      type = "l", col = "red")
lines(residuals.kernel20,
      x = linear.model$model$Time, xlab = "Time",
      type = "l", col = "green")
abline(h = 0)
legend("topright", col = c("red", "black", "green"),
       legend = c("Linear Residuals", "Kernel Smoother B = 4", "Kernel Smoother B = 20"),
       lty = c(1, 1, 1), cex = 0.7)
```

d)

It is observed that the residual plot does not look dependent on time anymore and this looks like noise and it is visually confirmed that its stationary. From the ACF plot it is seen that the seasonality has disappeared since we now included the month in the model and this is not affected in the residuals as before.

```
linear.model.d <- lm(rhine[, 4] ~ rhine[, 3] + as.factor(rhine[, 2]))
par(mfrow = c(1, 2))
acf(linear.model.d$residuals,
```

Residuals from kernel smoother

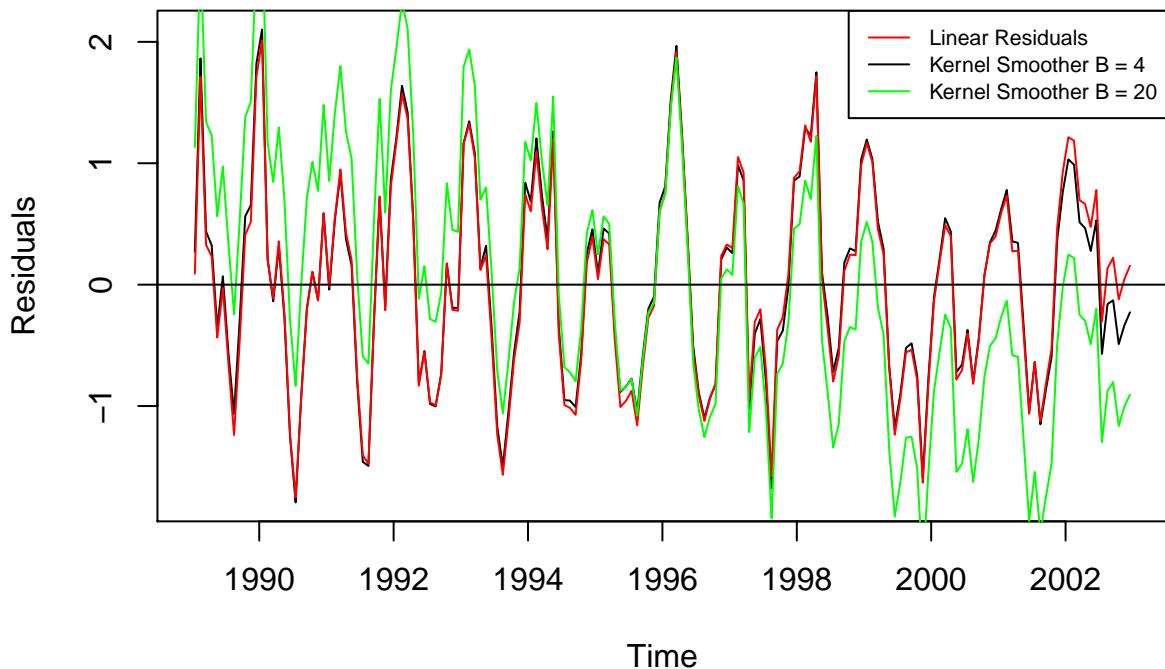


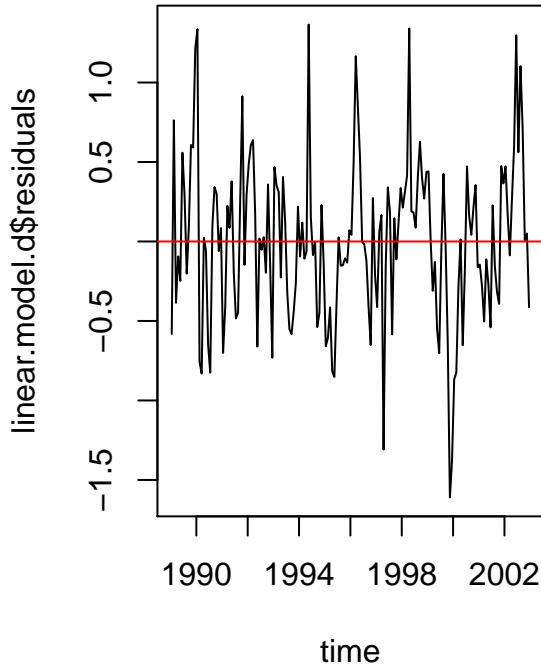
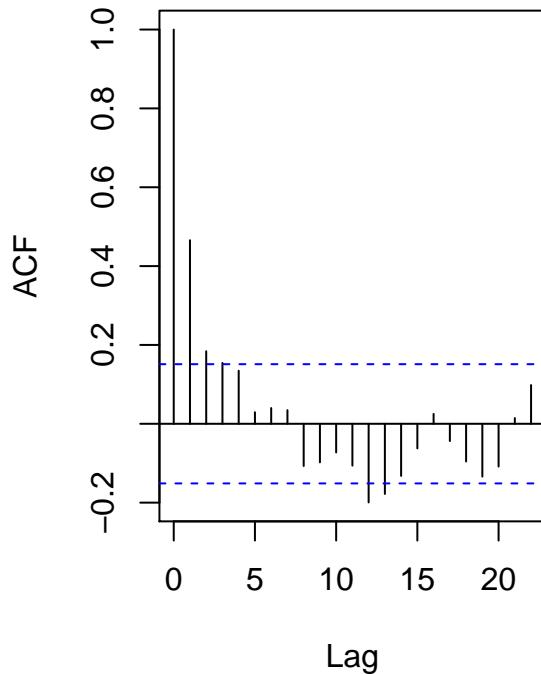
Figure 1: Comparing residuals for different kernel smoothers and linear model

```

main = "ACF for seasonal means model")
plot(y = linear.model.d$residuals, x = rhine[,3],
      type = "l", xlab="time")
abline(h=0, col = "red")

```

ACF for seasonal means model



```
par(mfrow = c(1,1))
```

e)

The previous model is the best one since we only consider two variables, time and month to estimate nitrogen concentration. The variable month is a factor and hence will not be removed even if it is significant. From the summary, but the best model given AIC is the same model as before.

```
#AIC used for stepwise feature selection
stepwise <- step(linear.model.d, direction = "backward")
```

```

## Start:  AIC=-202.02
## rhine[, 4] ~ rhine[, 3] + as.factor(rhine[, 2])
##
##                               Df Sum of Sq      RSS      AIC
## <none>                           43.237 -202.023
## - as.factor(rhine[, 2]) 11     68.524 111.761   -64.477
## - rhine[, 3]                  1    118.387 161.624    17.499

```

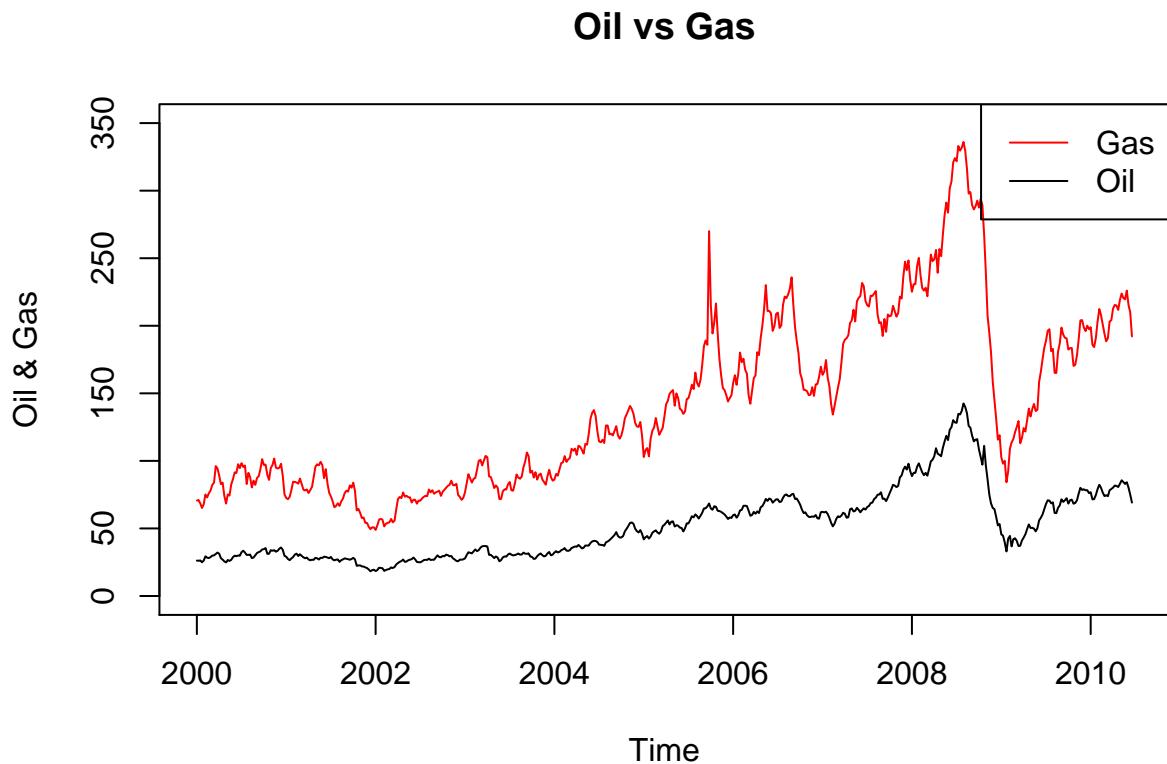
```
stepwiseSummary <- summary(stepwise)
```

Assignment 3

a)

Both Oil and Gas do not look stationary, especially the variable Gas since it is observed that the variance is dependent on time. The time series look like they are related as they both have an increasing trend followed by a crash after 2008.

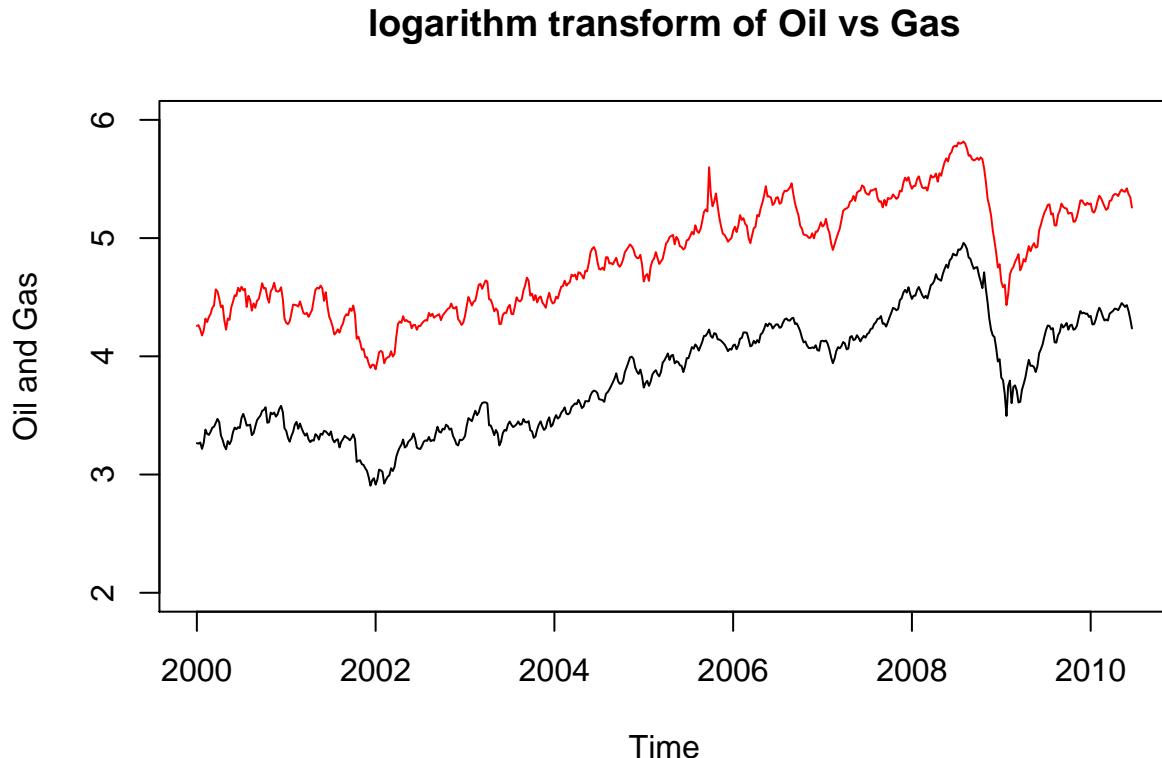
```
plot(oil, ylab = "Oil & Gas",
      xlab = "Time", main = "Oil vs Gas",
      ylim = c(0,350))
lines(gas, col = "red")
legend("topright", legend = c("Gas", "Oil"), lty = c(1,1), col = c("red", "black"))
```



b)

The log function reduces the exponential volatility in the data and hence, Log stabilizes the variance so its easier to compare the two different time series.

```
#Log Transform of both time series
logOil <- log(oil)
logGas <- log(gas)
plot(logOil, ylab = "Oil and Gas",
     xlab = "Time", main = "logarithm transform of Oil vs Gas",
     ylim = c(2,6))
lines(logGas, col = "red")
```

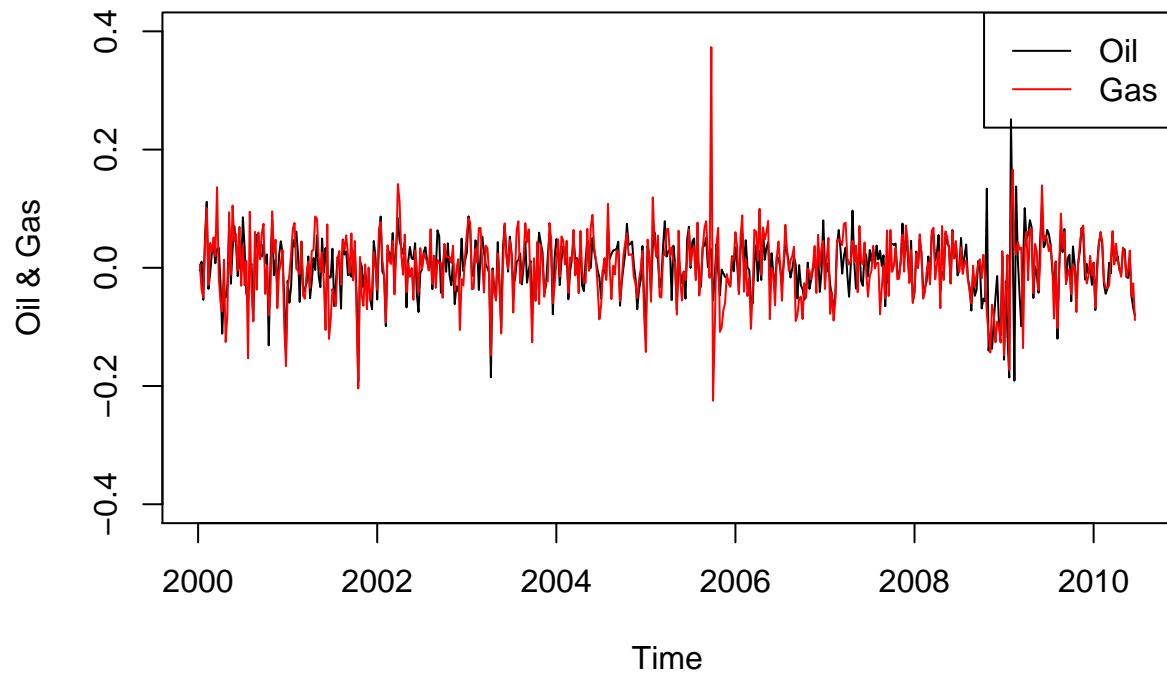


c)

From Figure, the first difference of the series seem to be stationary even though we have a couple of peaks.

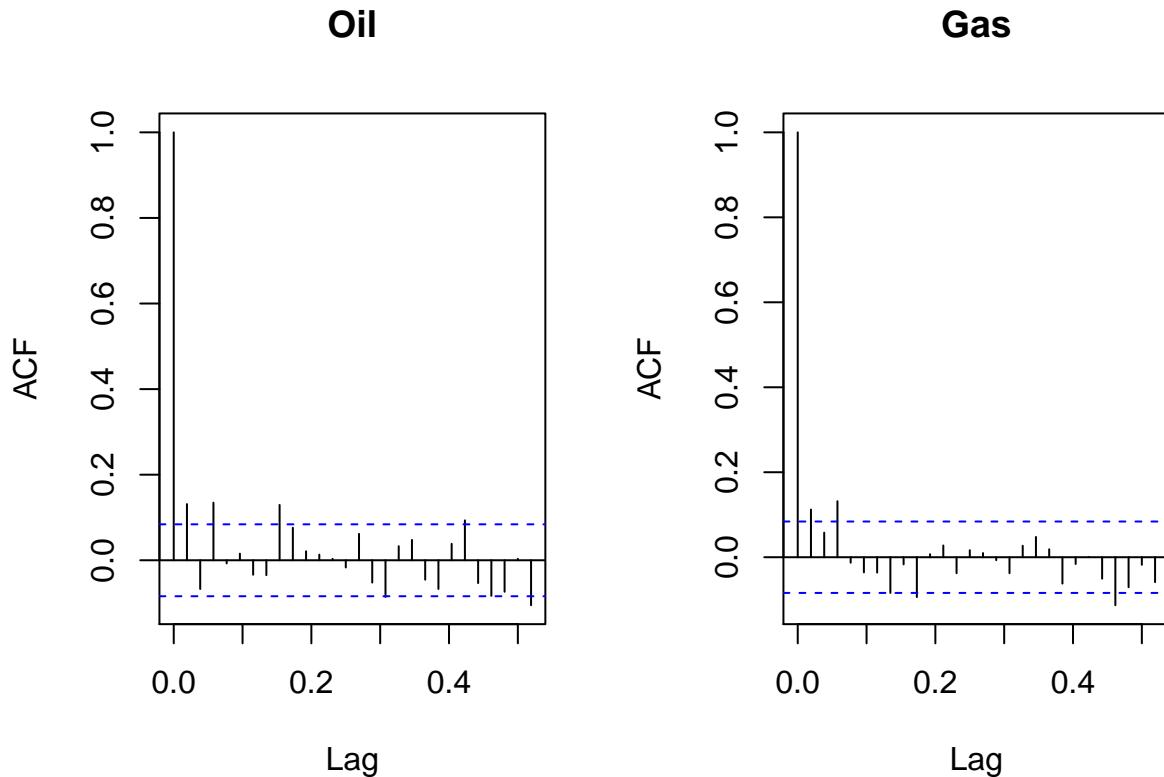
```
oilDiff <- diff(logOil)
gasDiff <- diff(logGas)
plot(oilDiff, ylab = "Oil & Gas",
      xlab = "Time", main = "First difference of Oil vs Gas",
      ylim = c(-0.4,0.4))
lines(gasDiff, col = "red")
legend("topright", legend = c("Oil","Gas"), col = c("black", "red"),
      lty = c(1,1))
```

First difference of Oil vs Gas



From the ACF plots it is observed that there isn't a significant correlation.

```
par(mfrow = c(1,2))
acf(oilDiff, main = "Oil")
acf(gasDiff, main = "Gas")
```

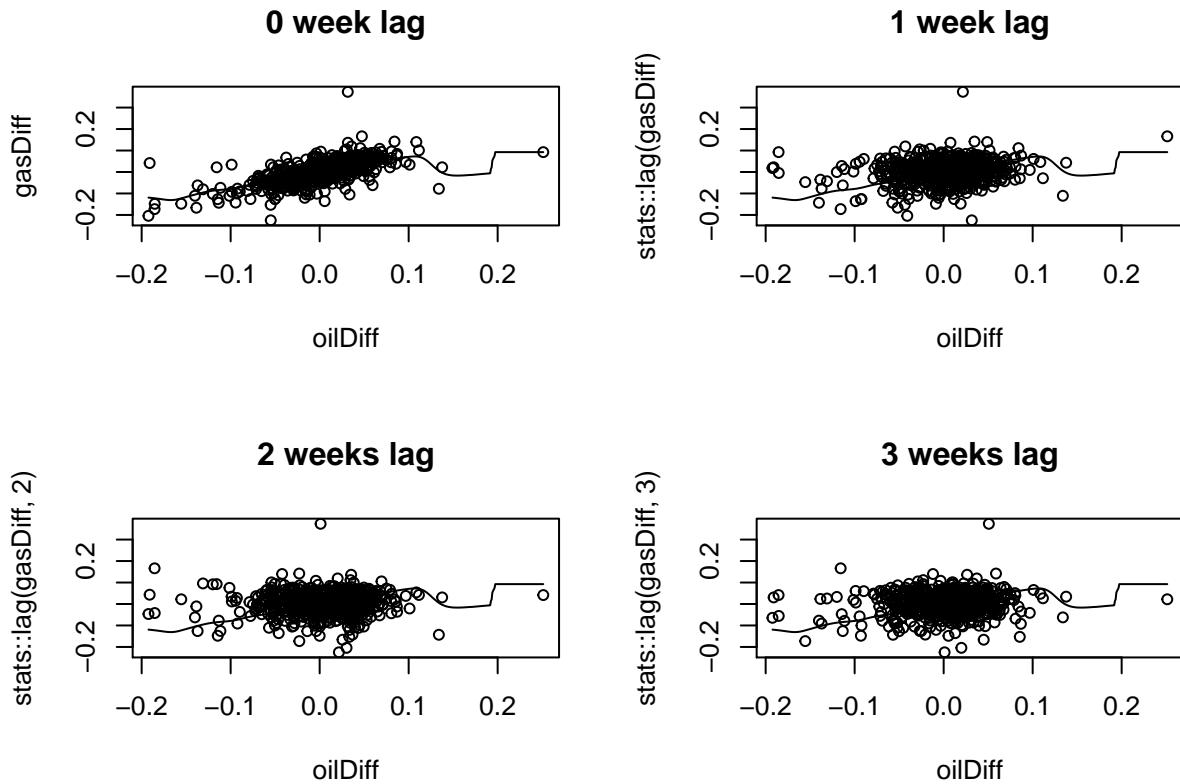


```
par(mfrow = c(1,1))
```

d)

It is observed from the fig of scatterplots, that the relationship seems to decrease as we increase the weekly lag but the difference is not large. The relationship seems to be somewhat linear with 0 lag. We do not have a linear relationship and few outliers are present.

```
par(mfrow = c(2,2))
plot(x = oilDiff , y = gasDiff, main = "0 week lag")
lines(ksmooth(x = oilDiff, y = gasDiff, bandwidth = 0.04, kernel = "normal"))
plot(x = oilDiff, y = stats::lag(gasDiff), main = "1 week lag")
lines(ksmooth(x = oilDiff, y = stats::lag(gasDiff), bandwidth = 0.04, kernel = "normal"))
plot(x = oilDiff, y = stats::lag(gasDiff, 2), main = "2 weeks lag")
lines(ksmooth(x = oilDiff, y = stats::lag(gasDiff,2), bandwidth = 0.04, kernel = "normal"))
plot(x = oilDiff, y = stats::lag(gasDiff, 3), main = "3 weeks lag")
lines(ksmooth(x = oilDiff, y = stats::lag(gasDiff,3), bandwidth = 0.04, kernel = "normal"))
```



```
par(mfrow = c(1,1))
```

e)

The following model is fit :

$$y_t = \alpha_0 + \alpha_1 I(x_t > 0) + \beta_1 x_t + \beta_2 x_{t-1} + w_t$$

According to the summary the lagged variable is not significant. The binary variable $I(x_t > 0)$ is significant at the 2nd confidence interval. The dummy variable has a positive coefficient which means that an increase in oil will also increase the gas. We can also see in Figure that the residuals seem to be constant over time with the exception of some outliers.

```
df <- ts.intersect(y = gasDiff, xt = oilDiff,
                    xt1 = stats::lag(oilDiff, 1), xtbin = oilDiff>0)
model3 <- lm(y ~ xt + xt1 + xtbin, data = df)
summary(model3)
```

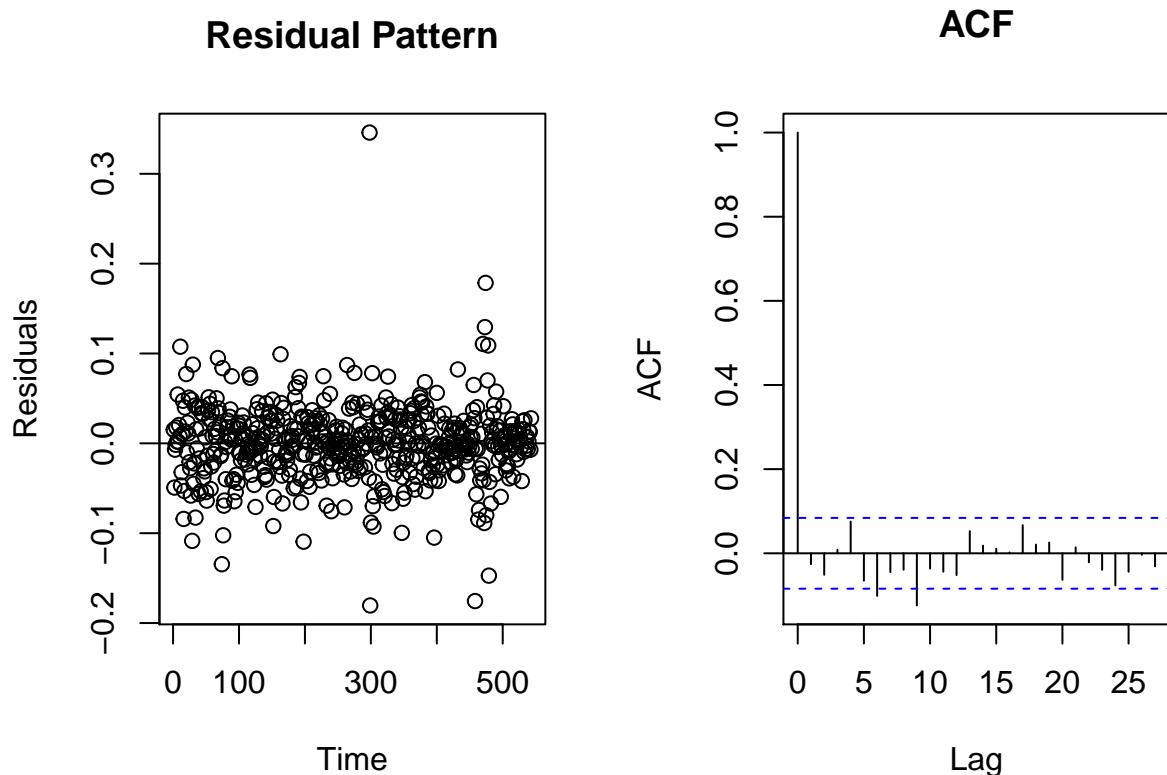
```
##
## Call:
## lm(formula = y ~ xt + xt1 + xtbin, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18044 -0.02103  0.00003  0.02170  0.34592
##
```

```

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006176   0.003470  -1.780  0.0757 .
## xt          0.694200   0.058898  11.786 <2e-16 ***
## xt1         0.012660   0.038729   0.327  0.7439
## xtnbin      0.012376   0.005542   2.233  0.0259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04202 on 539 degrees of freedom
## Multiple R-squared:  0.445, Adjusted R-squared:  0.4419
## F-statistic: 144.1 on 3 and 539 DF,  p-value: < 2.2e-16

par(mfrow = c(1,2))
plot(residuals(model3), ylab = "Residuals",
     xlab = "Time", main = "Residual Pattern")
abline(h = 0)
acf(residuals(model3), main = "ACF")

```



Computer lab B

Instructions

- The lab is assumed to be done in groups.
- Create a report to the lab solutions in PDF.
- Be concise and do not include unnecessary printouts and figures produced by the software and not required in the assignments.
- **Include all your codes as an appendix into your report.**
- A typical lab report should 2-4 pages of text plus some number of figures plus appendix with codes.
- The group lab report should be submitted via LISAM before the deadline specified in LISAM.
- **Use 12345 as a random seed everywhere where the result of the simulation differs with the run unless stated otherwise.**

Assignment 1. Computations with simulated data

- a. Generate 1000 observations from AR(3) process with $\phi_1 = 0.8, \phi_2 = -0.2, \phi_3 = 0.1$. Use these data and the definition of PACF to compute ϕ_{33} from the sample, i.e. write your own code that performs linear regressions on necessarily lagged variables and then computes an appropriate correlation. Compare the result with the output of function pacf() and with the theoretical value of ϕ_{33}
- b. Simulate an AR(2) series with $\phi_1 = 0.8, \phi_2 = 0.1$ and $n = 100$. Compute the estimated parameters and their standard errors by using three methods: method of moments (Yule-Walker equations), conditional least squares and maximum likelihood (ML) and compare their results to the true values. Which method does seem to give the best result? Does theoretical value for ϕ_2 fall within confidence interval for ML estimate?
- c. Generate 200 observations of a seasonal $ARIMA(0,0,1) \times (0,0,1)_{12}$ model with coefficients $\Theta = 0.6$ and $\theta = 0.3$ by using arima.sim(). Plot sample ACF and PACF and also theoretical ACF and PACF. Which patterns can you see at the theoretical ACF and PACF? Are they repeated at the sample ACF and PACF?
- d. Generate 200 observations of a seasonal $ARIMA(0,0,1) \times (0,0,1)_{12}$ model with coefficients $\Theta = 0.6$ and $\theta = 0.3$ by using arima.sim(). Fit $ARIMA(0,0,1) \times (0,0,1)_{12}$ model to the data, compute forecasts and a prediction band 30 points ahead and plot the original data and the forecast with the prediction band. Fit the same data with function *gausspr* from package **kernlab** (use default settings). Plot the original data and predicted data from $t = 1$ to $t = 230$. Compare the two plots and make conclusions.
- e. Generate 50 observations from ARMA(1,1) process with $\phi = 0.7, \theta = 0.5$. Use first 40 values to fit an ARMA(1,1) model with $\mu = 0$. Plot the data, the 95% prediction band and plot also the true 10 values that you initially dropped. How many of them are outside the prediction band? How can this be interpreted?

Assignment 2. ACF and PACF diagnostics.

- a. For data series *chicken* in package **astsa** (denote it by x_t), plot 4 following graphs up to 40 lags: $ACF(x_t)$, $PACF(x_t)$, $ACF(\nabla x_t)$, $PACF(\nabla x_t)$ (group them in one graph). Which $ARIMA(p, d, q)$ or $ARIMA(p, d, q) \times (P, D, Q)_s$ models can be suggested based on this information only? Motivate your choice.
- b. Repeat step 1 for the following datasets: *so2*, *Eqcount*, *HCT* in package **astsa**.

Assignment 3. ARIMA modeling cycle.

In this assignment, you are assumed to apply a complete ARIMA modeling cycle starting from visualization and detrending and ending up with a forecasting.

- a. Find a suitable $ARIMA(p, d, q)$ model for the data set *oil* present in the library **astsa**. Your modeling should include the following steps in an appropriate order: visualization, unit root test, detrending by differencing (if necessary), transformations (if necessary), ACF and PACF plots when needed, EACF analysis, Q-Q plots, Box-Ljung test, ARIMA fit analysis, control of the parameter redundancy in the fitted model. When performing these steps, always have 2 tentative models at hand and select one of them in the end. Validate your choice by AIC and BIC and write down the equation of the selected model. Finally, perform forecasting of the model 20 observations ahead and provide a suitable plot showing the forecast and its uncertainty.
- b. Find a suitable $ARIMA(p, d, q) \times (P, D, Q)_s$ model for the data set *unemp* present in the library **astsa**. Your modeling should include the following steps in an appropriate order: visualization, detrending by differencing (if necessary), transformations (if necessary), ACF and PACF plots when needed, EACF analysis, Q-Q plots, Box-Ljung test, ARIMA fit analysis, control of the parameter redundancy in the fitted model. When performing these steps, always have 2 tentative models at hand and select one of them in the end. Validate your choice by AIC and BIC and write down the equation of the selected model (write in the backshift operator notation without expanding the brackets). Finally, perform forecasting of the model 20 observations ahead and provide a suitable plot showing the forecast and its uncertainty.

Time series Lab 2

Omkar Bhutra (omkbh878)

24 September 2019

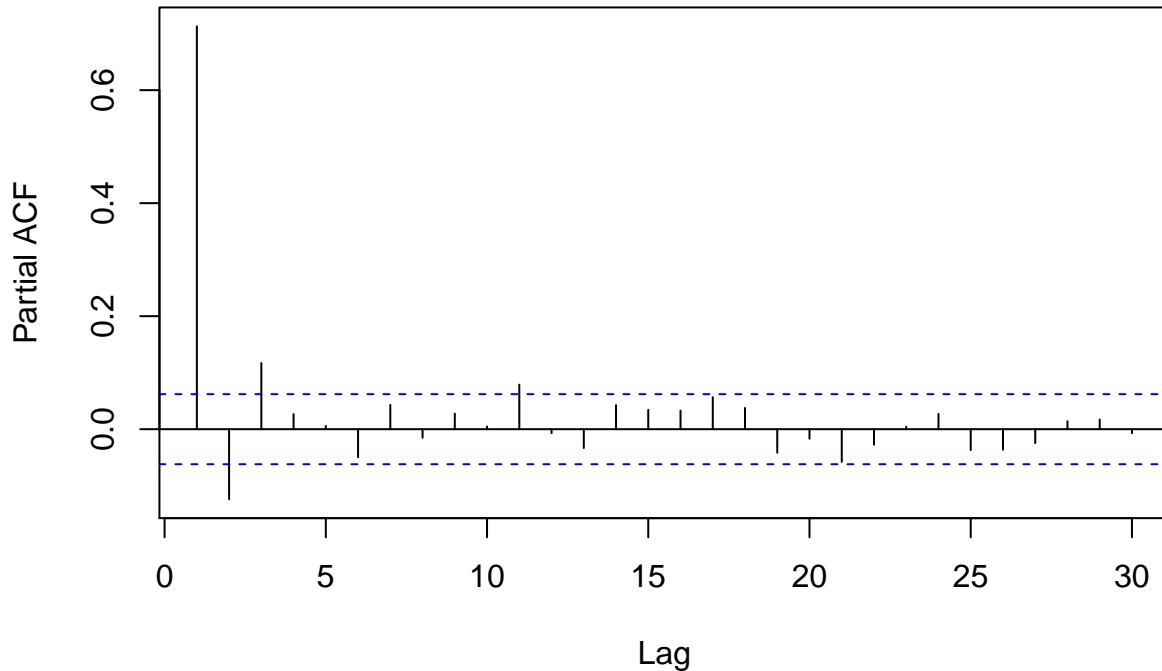
Assignment 1. Computations with simulated data

1a. Generate 1000 observations from AR(3) process with phi's $1=0.8, 2=-0.2, 3=0.1$. Use these data and the definition of PACF to compute 33 from the sample, i.e. write your own code that performs linear regressions on necessarily lagged variables and then computes an appropriate correlation. Compare the result with the output of function pacf() and with the theoretical value of 33

1000 observations are generated from $AR(3)$ with $\phi_1 = 0.8\phi_2 = -0.2\phi_3 = 0.1$ with the *arima.sim*. And we used the correlation from the first lag from the *PACF* from the generated data to calculate the theoretical PACF using the *ARMAacf* function. To compare the theoretical with the simulated data we used a linear regression we created a *dataframe*. X is the simulated data from $ARIMA(0.8, -0.2, 0.1)$, $X1$ is X minus one lag, $X2$ is X minus 2 lags and $X3$ is x minus 3 lags. Then saving the residuals from residuals from the linear regression where X is the dependent variable explained by $x1$ and $x2$. Also when $x3$ is the dependent variable explained by $x1$ and $x2$. The manual calculated is preformed by the correlation between the both residuals from the linear regressions.

```
set.seed(12345)
#simulate
AR3 <- arima.sim(list(ar=c(0.8,-0.2,0.1)), n=1000)
#theoretical pacf
AR3pacf <- pacf(AR3)
```

Series AR3



```

AR3data <- ts.intersect(x = AR3, x1=stats::lag(AR3,1), x2=stats::lag(AR3,2), x3=stats::lag(AR3,3))

AR_lm <- lm(x ~ x1+x2,data=AR3data)
AR_lm_lag3 <- lm(x3 ~ x2+x1,data=AR3data)
r1=residuals(AR_lm)
r2=residuals(AR_lm_lag3)
cor(cbind(r1,r2))

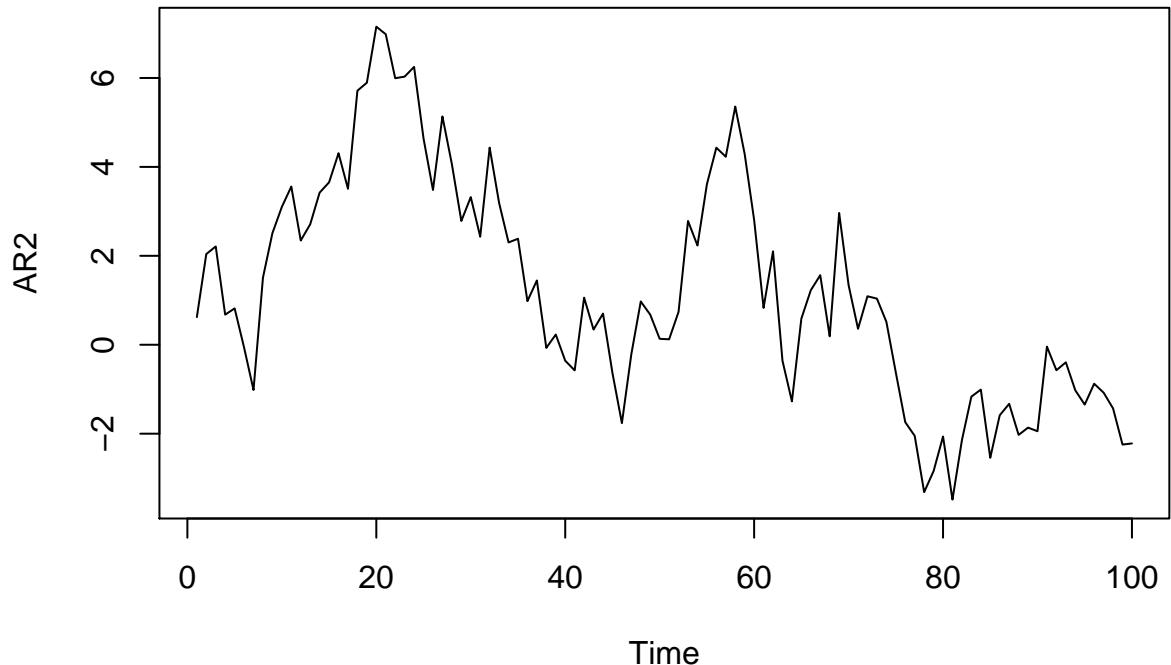
##          r1          r2
## r1 1.0000000 0.1146076
## r2 0.1146076 1.0000000

cat(paste("The theoretical pacf by lag 3:", cor(cbind(r1,r2))[1,2]))
```

The theoretical pacf by lag 3: 0.114607639249897

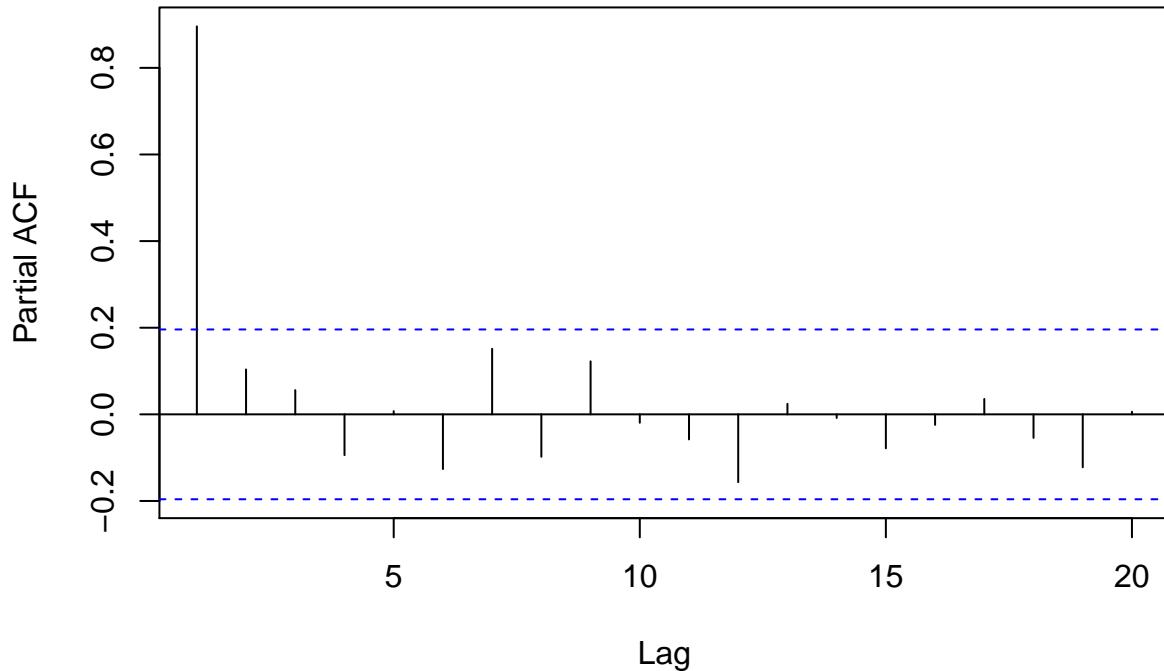
1b. Simulate an AR(2) series with $\phi_1=0.8$, $\phi_2=0.1$ and $n=100$. Compute the estimated parameters and their standard errors by using three methods: method of moments (Yule-Walker equations), conditional least squares and maximum likelihood (ML) and compare their results to the true values. Which method does seem to give the best result? Does theoretical value for ϕ_2 fall within confidence interval for ML estimate?

```
set.seed(12345)
AR2 <- arima.sim(list(ar=c(0.8,0.1)), n=100)
plot(AR2)
```



```
pacf(AR2)
```

Series AR2



```
paste("method of moments (Yule-Walker equations)")

## [1] "method of moments (Yule-Walker equations)"

yulewalker <- ar.yw(AR2, aic=F, order.max=2)
paste("estimated parameters")

## [1] "estimated parameters"

yulewalker$ar

## [1] 0.8029146 0.1037053

paste("standard errors")

## [1] "standard errors"

yulewalker$asy.var.coef

## [,1]      [,2]
## [1,] 0.010198404 -0.009135888
## [2,] -0.009135888  0.010198404
```

```

paste("conditional least squares")

## [1] "conditional least squares"

condleastsquares <- arima(AR2,order=c(2,0,0), method = c("CSS"))
paste("coefficient estimates")

## [1] "coefficient estimates"

condleastsquares$model$phi

## [1] 0.8066846 0.1205352

paste("standard errors")

## [1] "standard errors"

condleastsquares$var.coef

##               ar1          ar2      intercept
## ar1      0.009691316 -0.00883412 -0.004071024
## ar2     -0.008834120  0.00988207 -0.013287047
## intercept -0.004071024 -0.01328705  2.290286649

paste("maximum likelihood")

## [1] "maximum likelihood"

maxliklihood <- ar.mle(AR2, aic=F, order.max=2)
paste("ar coefficient estimates")

## [1] "ar coefficient estimates"

maxliklihood$ar

##       ar1      ar2
## 0.7968774 0.1189369

paste("standard errors")

## [1] "standard errors"

maxliklihood$asy.var.coef

##            [,1]      [,2]
## [1,] 0.009072681 -0.008127448
## [2,] -0.008127448  0.009072681

```

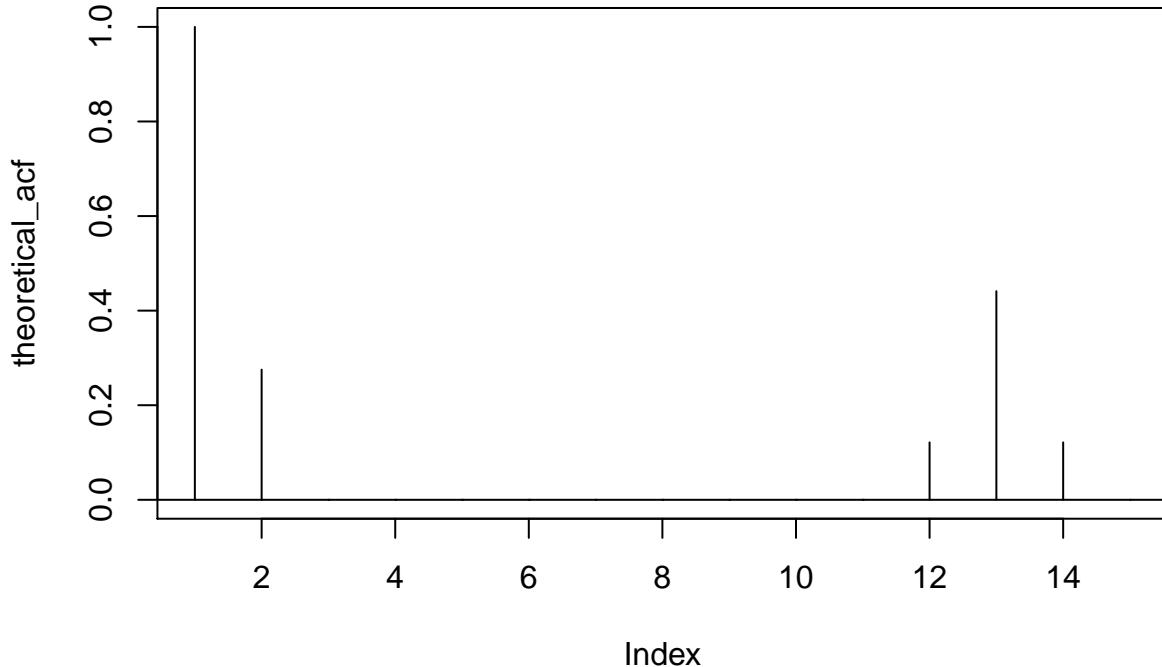
The results from the Yule walker seem to be the best parameters obtained. Yes, theoretical ϕ_2 is inside the confidence interval for the maximum likelihood estimate.

1c. Generate 200 observations of a seasonal ARIMA $(0,0,1) \times (0,0,1)12$ model with coefficients $\Theta=0.6$ and $=0.3$ by using arima.sim(). Plot sample ACF and PACF and also theoretical ACF and PACF. Which patterns can you see at the theoretical ACF and PACF? Are they repeated at the sample ACF and PACF?

```
set.seed(12345)
ma.coeff <- c(0.3, rep(0, 10), 0.6)
ar <- arima.sim(n=200, model = list(order=c(0,0,12), ma = ma.coeff))
#ar.seasonal<- sarima(ar, 0, 0, 1, P = 0, D = 0, Q = 1, S = 12, Model = TRUE)
theoretical_acf <- ARMAacf(ma = c(ma.coeff, 0.3*0.6))
theoretical_pacf <- ARMAacf(ma = c(ma.coeff, 0.3*0.6), pacf = TRUE)

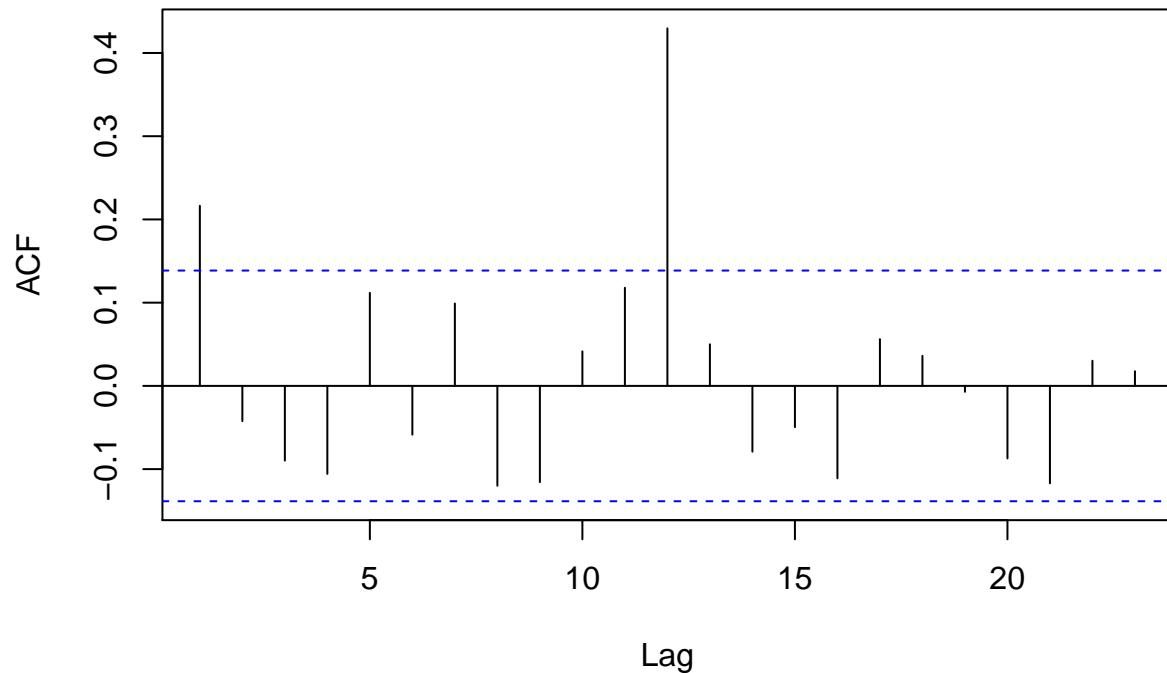
plot(theoretical_acf, type="h", main="Theoretical ACF")
abline(h=0)
```

Theoretical ACF



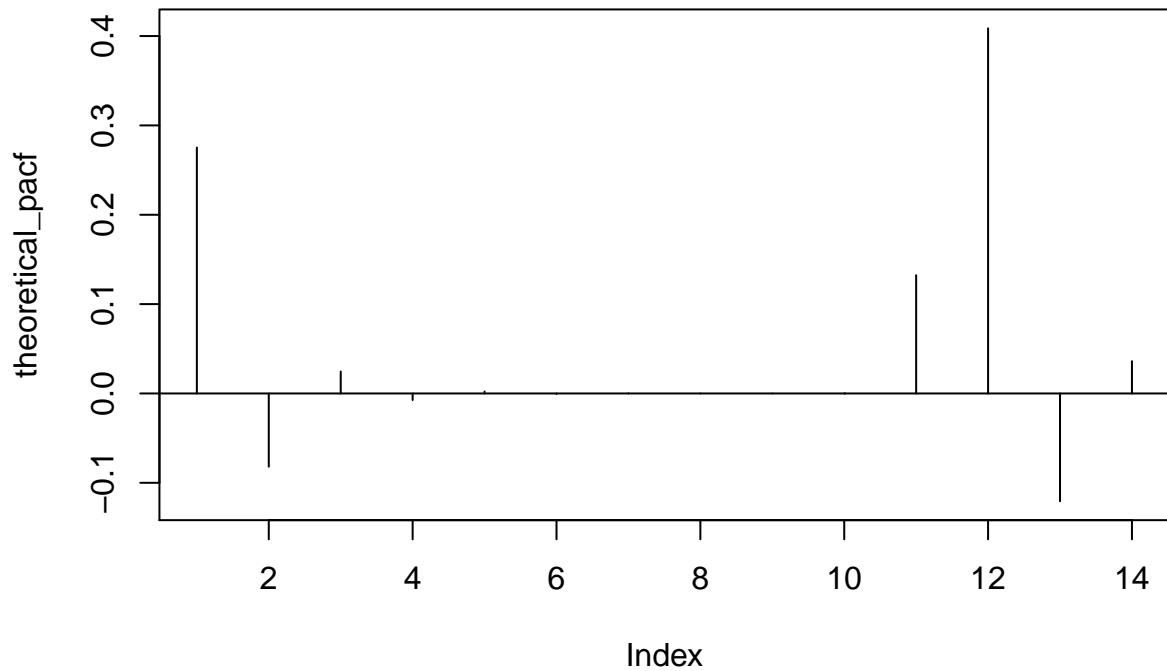
```
sample_acf <- acf(ar, main="Sample ACF")
```

Sample ACF



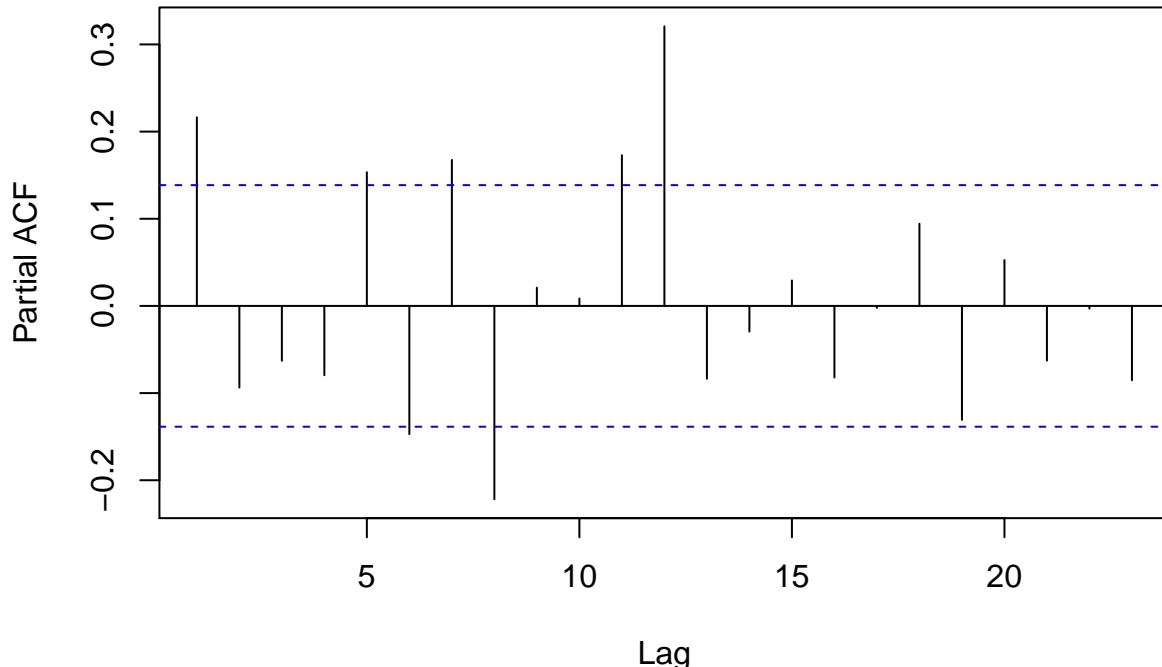
```
plot(theoretical_pacf, type="h", main="Theoretical PACF")
abline(h=0)
```

Theoretical PACF



```
sample_pacf <- pacf(ar, main="Sample PACF")
```

Sample PACF



The ACF patterns are similar between the theoretical and sample observations. In theoretical ACF, there are large spikes at lags 1 and 13 while in sample ACF we have large spikes at 1 and 12 suggesting some correlation exists along the lags. In PACF patterns, the spikes are at lags 1 and 12 for both theoretical and sample.

1c. Generate 200 observations of a seasonal ARIMA $(0,0,1)\times(0,0,1)12$ model with coefficients $\Theta=0.6$ and $=0.3$ by using arima.sim(). Fit $(0,0,1)\times(0,0,1)12$ model to the data, compute forecasts and a prediction band 30 points ahead and plot the original data and the forecast with the prediction band. Fit the same data with function gausspr from package kernlab (use default settings). Plot the original data and predicted data from $=1$ to $=230$. Compare the two plots and make conclusions.

```

set.seed(12345)
ma.coeff <- c(0.3,rep(0,10),0.6)
ar <- arima.sim(n=200, model = list(order=c(0,0,12), ma = ma.coeff))
ar_fit <- arima(ar,order = c(0,0,1),seasonal = list(order = c(0,0,1),period = 12))
ar_pred <- predict(ar_fit,n.ahead = 30, se.fit = TRUE)

#gausspr is an implementation of Gaussian processes for classification and regression
gausspr_data <- data.frame(y = ar, x = 1:200)
gausspr_fit <- kernlab::gausspr(y ~ x, gausspr_data)

## Using automatic sigma estimation (sigest) for RBF or laplace kernel

gausspr_pred <- predict(gausspr_fit, data.frame(x=201:230))

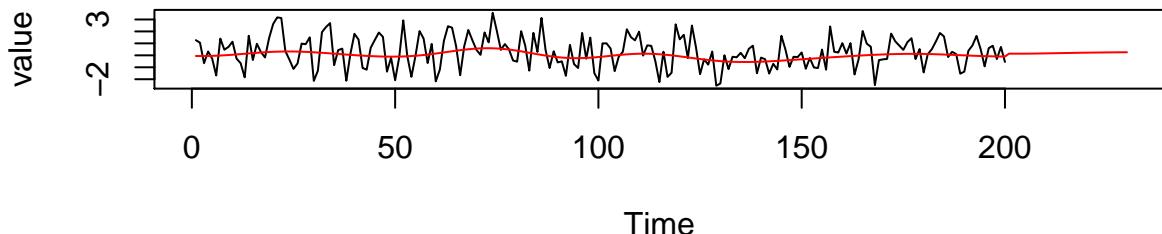
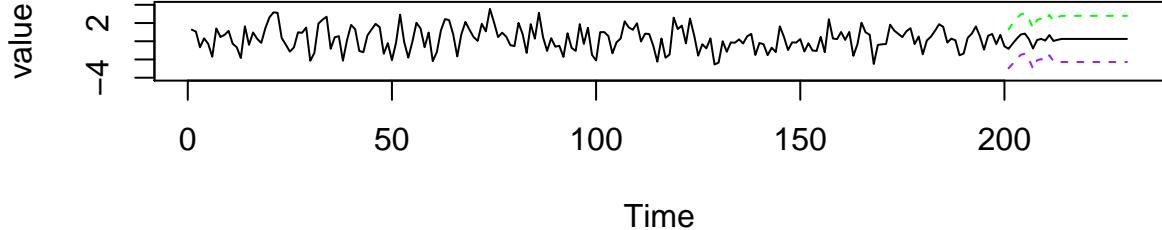
par(mfrow=c(2, 1))

```

```

plot(ts(c(ar, ar_pred$pred)), ylim=c(-4, 4), ylab="value", title = "Moving average model with prediction")
lines(200 + 1:length(ar_pred$pred), ar_pred$pred + 1.96 * ar_pred$se, lty=2, col="green")
lines(200 + 1:length(ar_pred$pred), ar_pred$pred - 1.96 * ar_pred$se, lty=2, col="purple")
plot(ar, xlim=c(0, 230), ylab="value", title = "Moving average model")
lines(c(fitted(gausspr_fit), gausspr_pred), , col="red")

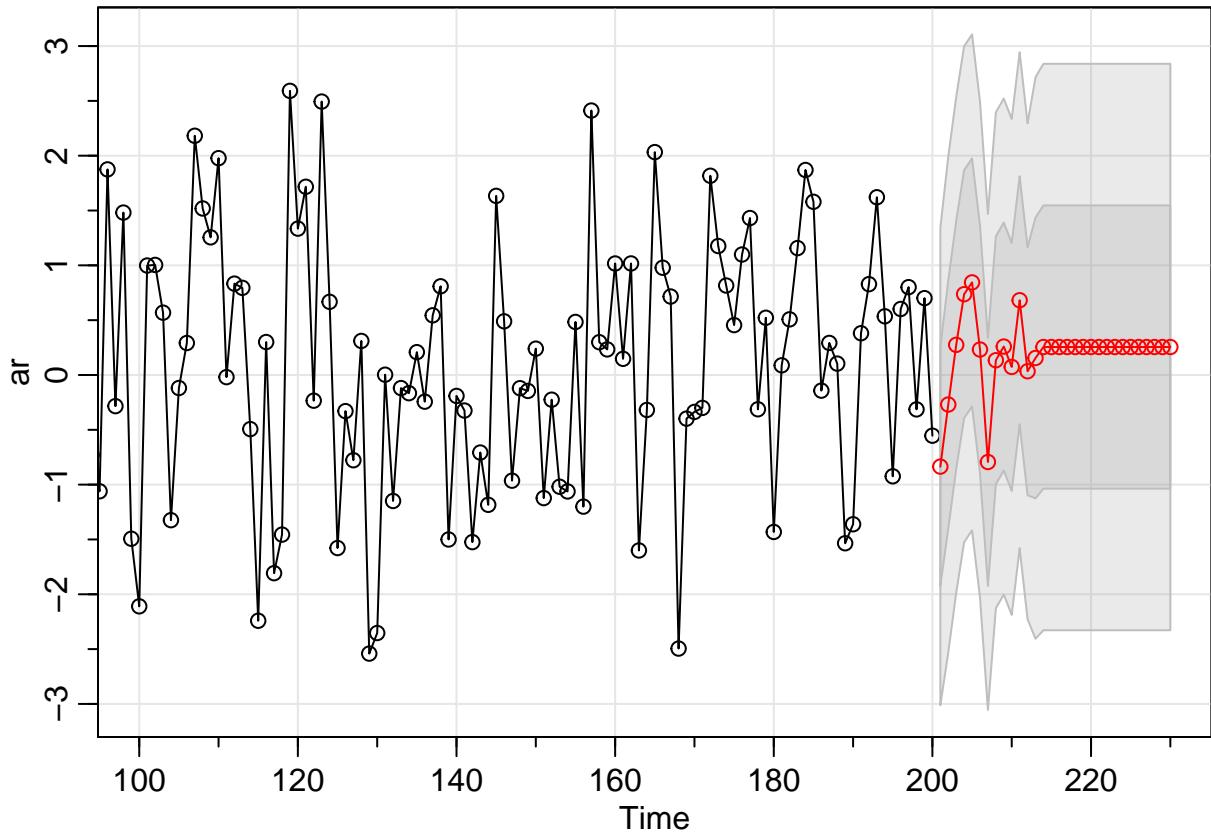
```



```

par(mfrow=c(1, 1))
#sarima for prediciton
sarima.for(ar, 30, 0, 0, 1, 0, 0, 1, 12)

```



```

## $pred
## Time Series:
## Start = 201
## End = 230
## Frequency = 1
## [1] -0.83557722 -0.26951120  0.27478352  0.73678068  0.84455443
## [6]  0.23224666 -0.79358727  0.13448054  0.25951986  0.07335756
## [11]  0.68187185  0.03408012  0.15501563  0.25496000  0.25496000
## [16]  0.25496000  0.25496000  0.25496000  0.25496000  0.25496000
## [21]  0.25496000  0.25496000  0.25496000  0.25496000  0.25496000
## [26]  0.25496000  0.25496000  0.25496000  0.25496000  0.25496000
##
## $se
## Time Series:
## Start = 201
## End = 230
## Frequency = 1
## [1] 1.088418 1.130806 1.130806 1.130806 1.130806 1.130806 1.130806
## [8] 1.130806 1.130806 1.130806 1.130806 1.130806 1.280439 1.291578
## [15] 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578
## [22] 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578 1.291578
## [29] 1.291578 1.291578

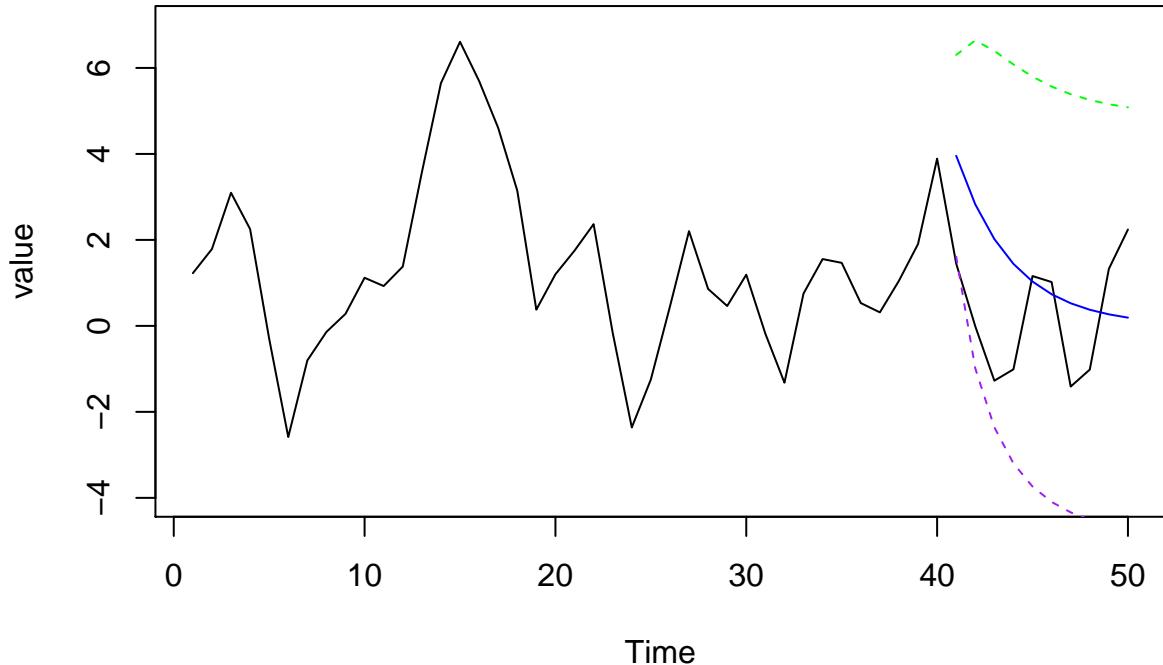
```

The prediction is reasonable for 12 months but then the model predicts the mean values for the next 18 months. The Gaussian process has an initial kink in the prediction and after that it remains linear due to the smooth fit to the observed data.

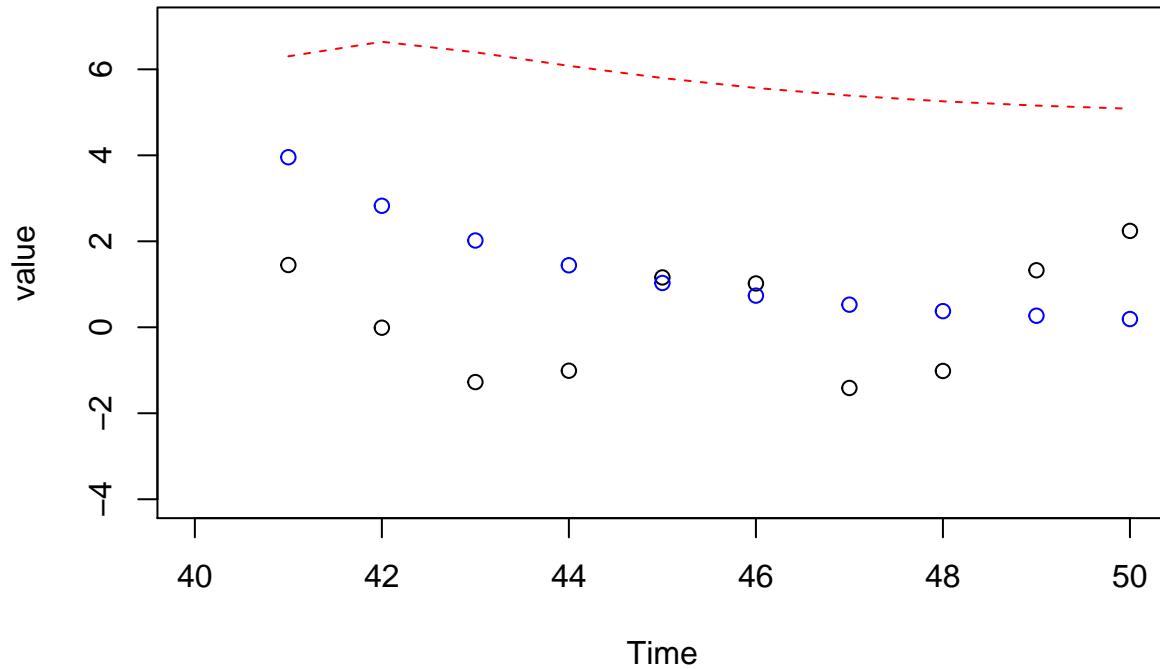
1e. Generate 50 observations from ARMA(1,1) process with $\phi = 0.7$, $\mu = 0.5$. Use first 40 values to fit an ARMA(1,1) model with $\mu = 0$. Plot the data, the 95% prediction band and plot also the true 10 values that you initially dropped. How many of them are outside the prediction band? How can this be interpreted?

```
set.seed(12345)
ar <- arima.sim(model=list(ma=c(0.5), ar=c(0.7)), n=50)
train <- ts(ar[1:40])
test <- ts(ar[41:50])
ar_fit <- arima(train, order=c(1, 0, 1), include.mean = F)
ar_pred <- predict(ar_fit, n.ahead=10)

plot(ts(c(train, test)), ylim=c(-4, 7), type="l", ylab="value")
lines(40 + 1:length(test), ar_pred$pred, col="blue")
lines(40 + 1:length(test), ar_pred$pred + 1.96 * ar_pred$se, lty=2, col="green")
lines(40 + 1:length(test), ar_pred$pred - 1.96 * ar_pred$se, lty=2, col="purple")
```



```
plot(40 + 1:length(test), test, ylim=c(-4, 7), xlim=c(40, 50), type="p", ylab="value", xlab="Time")
points(40 + 1:length(test), ar_pred$pred, col="blue")
lines(40 + 1:length(test), ar_pred$pred + 1.96 * ar_pred$se, lty=2, col="red")
```

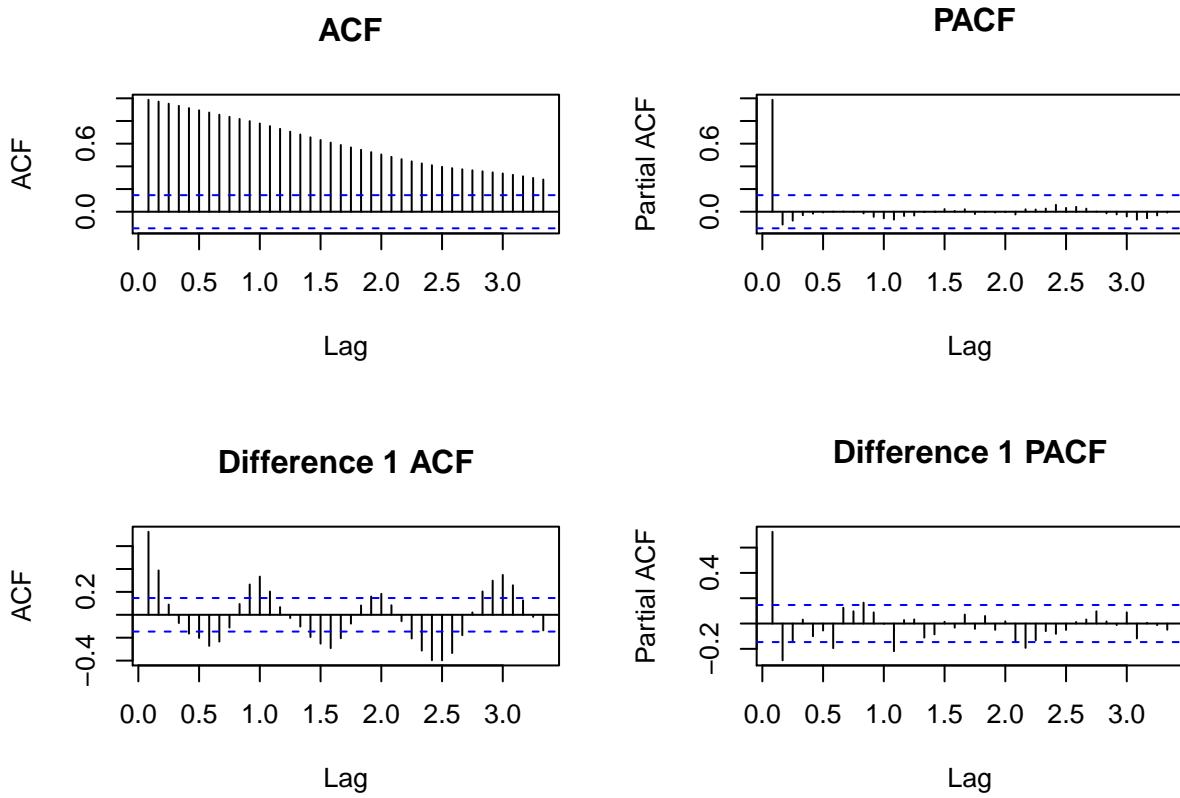


Only one observation lies outside the confidence interval , this means that since we expected 5/100 observations to be outside the 95% confidence interval, it is reasonable to find 1/10 to be outside.

Assignment 2. ACF and PACF diagnostics

```
acf_pacf_diag<- function(data){
  par(mfrow = c(2, 2))
  acf(data, lag.max = 40, main=" ACF")
  pacf(data, lag.max = 40, main=" PACF")
  acf(diff(data, lag = 1), lag.max = 40, main= "Difference 1 ACF")
  pacf(diff(data, lag = 1), lag.max = 40, main= "Difference 1 PACF")
  par(mfrow = c(2, 2))
}

acf_pacf_diag(chicken)
```



ACF : The ACF on the original data suggests an AR or ARMA model since the ACF tails off.

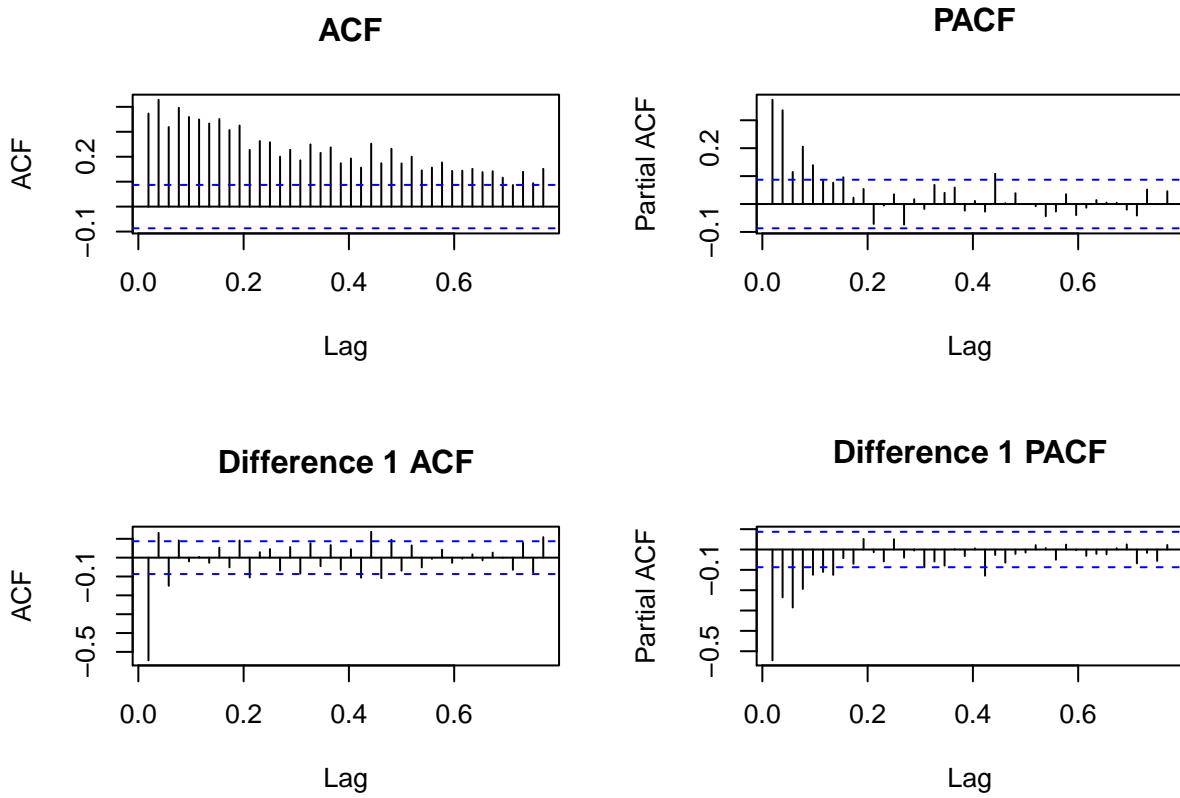
PACF: The PACF on the original data cuts off after lag 1 suggesting an AR(1) model.

Difference 1 ACF: 1st Differencing suggests seasonality in the data. The ACF tails off which suggests an AR model.

Difference 1 PACF: The PACF shows that the seasonality cuts off after lag 12 which is $1 * 12$ indicating an AR(1) model.

ARIMA(1, 0, 0) $x(1, 1, 0)_{12}$ can be suggested.

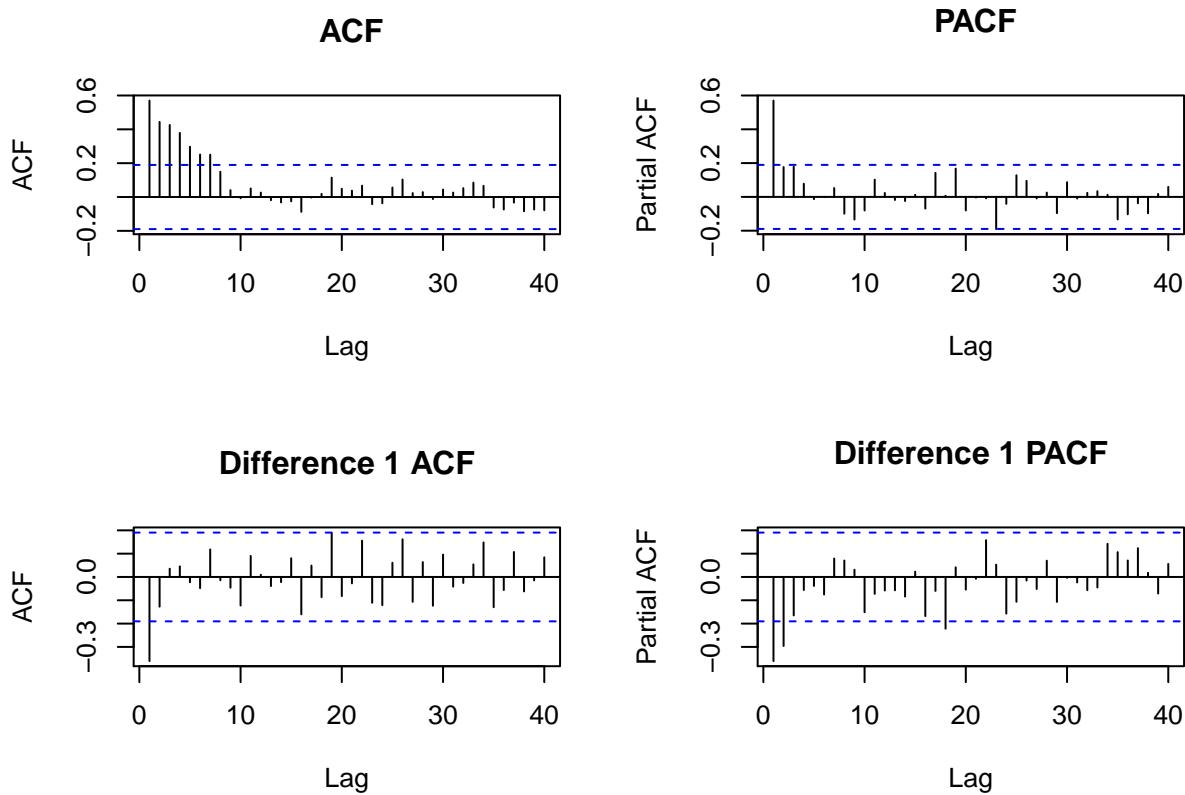
```
acf_pacf_diag(so2)
```



ACF: The ACF tails off suggesting either an AR or ARMA model. PACF: The PACF tails off , also suggesting an ARMA model. Difference 1 ACF: The ACF after difference cuts off after lag 1 suggesting a MA(1) model. Difference 1 PACF: The PACF after difference tails off further suggesting a MA(1) model.

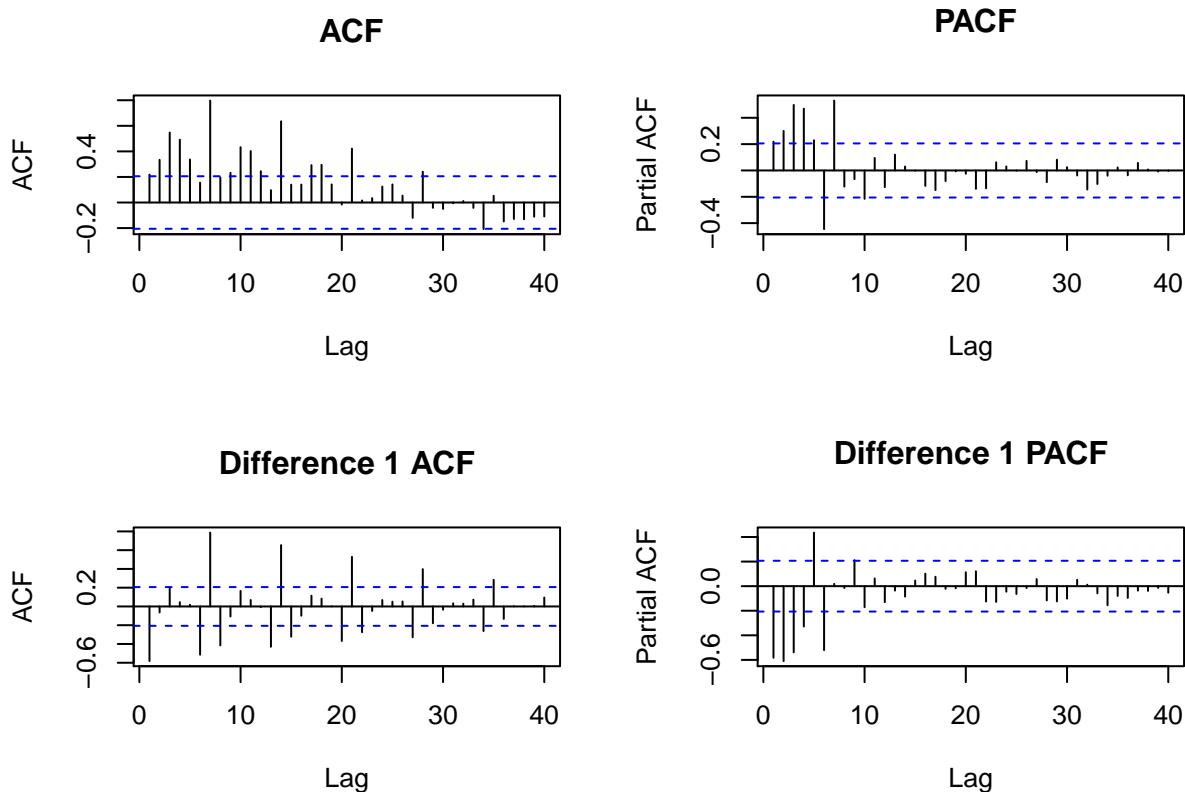
ARIMA(0, 1, 1) can be suggested.

```
acf_pacf_diag(EQcount)
```



ACF: The ACF tails off suggesting an AR or ARMA model. PACF: The PACF cuts off after lag 1 suggesting AR(1) model. Difference 1 Data ACF: The ACF after difference cuts off after lag 1 suggesting a MA(1) model. Difference 1 Data PACF: The PACF after difference tails off further suggesting a MA model. Either a ARMA(1, 0, 0) or ARMA(0, 1, 1) is suggested.

```
acf_pacf_diag(HCT)
```



ACF: The ACF tails off suggesting either an AR or ARMA model. PACF: The PACF cuts off after lag 7 suggesting an AR(7) model. Difference 1 ACF: The ACF suggests seasonality that tails off after lag 7 suggesting an seasonality of 7. Difference 1 PACF: The PACF cuts off after 6 lags suggesting an AR(6) seasonality model.

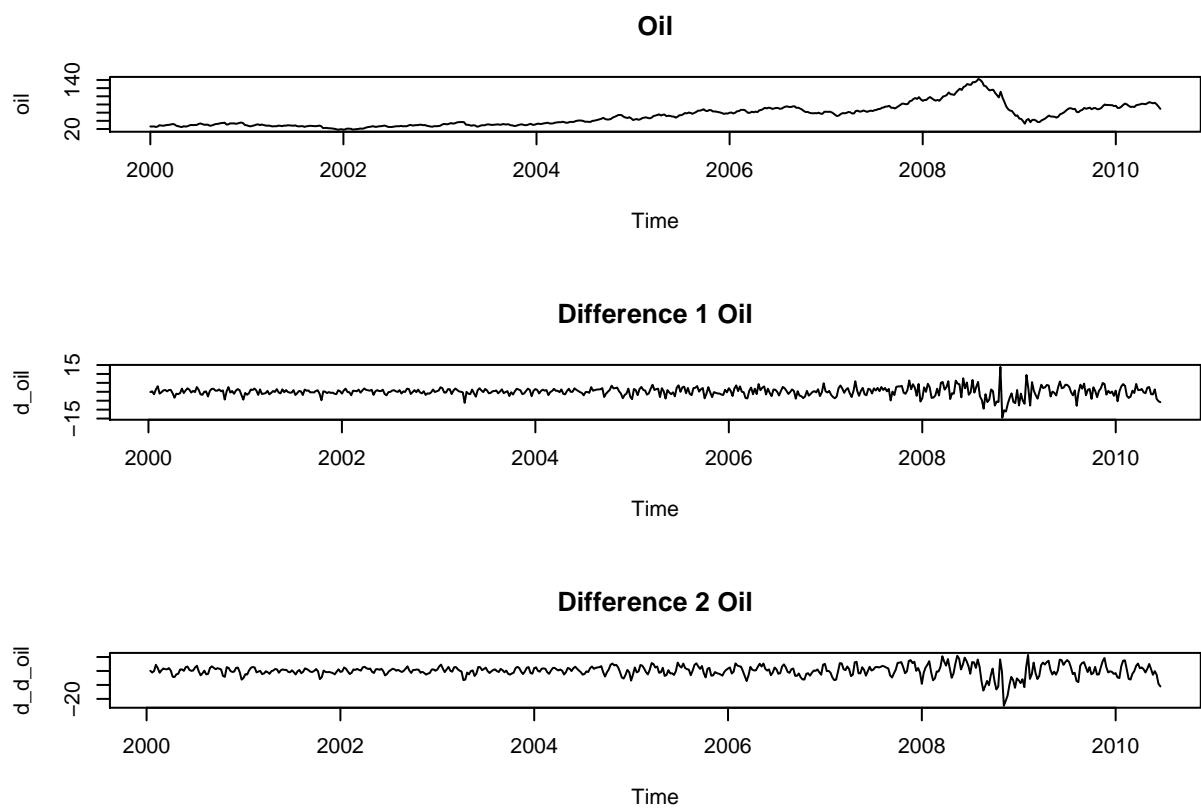
ARIMA(7, 0, 0)*X*(1, 1, 0)₇ model can be suggested.

Assignment 3. ARIMA modelling cycle

Question 1

```
log_oil <- log(oil)
d_oil <- diff(oil)
d_d_oil <- diff(oil, 2)
d_log_oil <- diff(log_oil)
d_d_log_oil <- diff(log_oil, 2)

par(mfrow=c(3, 1))
plot(oil, main="Oil")
plot(d_oil, main="Difference 1 Oil")
plot(d_d_oil, main="Difference 2 Oil")
```

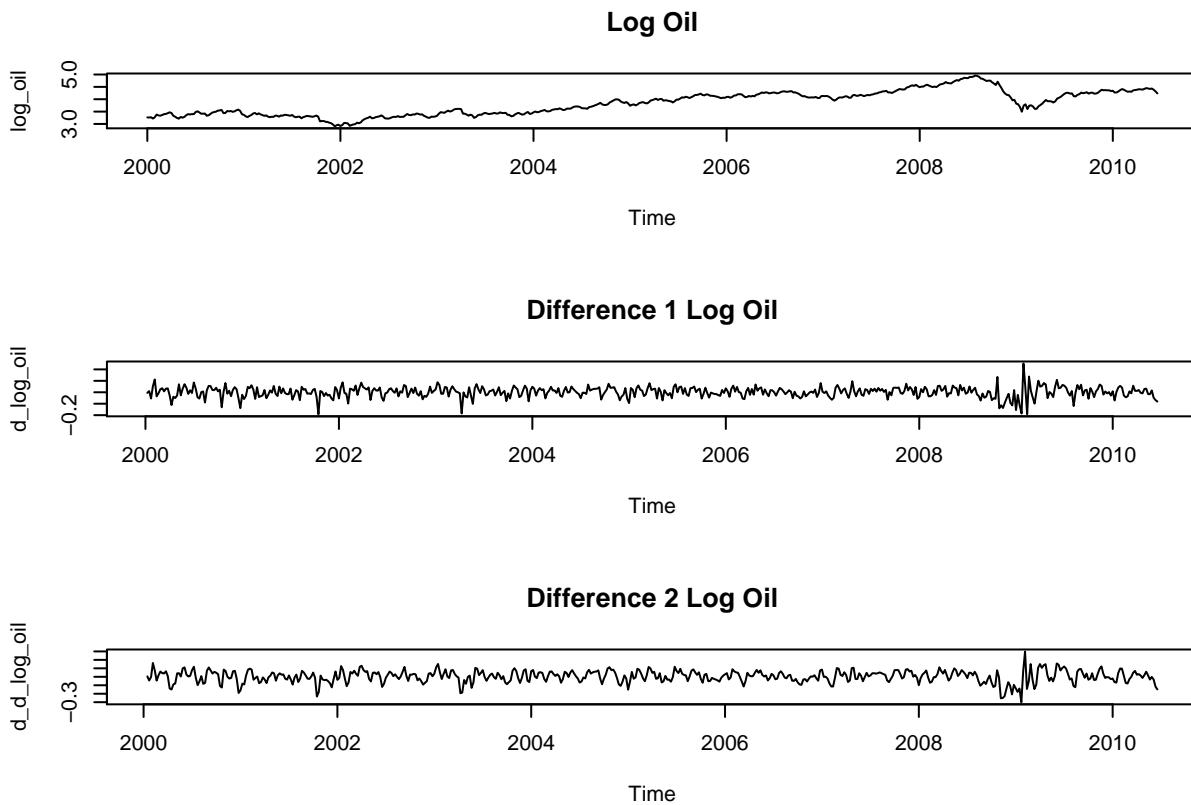


```
par(mfrow=c(3, 1))
```

```

par(mfrow=c(3, 1))
plot(log_oil, main="Log Oil")
plot(d_log_oil, main="Difference 1 Log Oil")
plot(d_d_log_oil, main="Difference 2 Log Oil")

```



```

par(mfrow=c(3, 1))

```

The logarithm transformed data is what we have to work with as it adjusts the scale appropriately for the price fluctuations. The first difference can be considered stationary and while the 2nd difference does not do much visual change. We can keep both models tentatively.

```

adf.test(oil)

```

```

## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag      ADF p.value
## [1,] 0  0.0628  0.662
## [2,] 1 -0.1556  0.599
## [3,] 2 -0.1983  0.587
## [4,] 3 -0.3284  0.549
## [5,] 4 -0.3675  0.538
## [6,] 5 -0.4799  0.506

```

```

## Type 2: with drift no trend
##      lag   ADF p.value
## [1,] 0 -1.29  0.600
## [2,] 1 -1.51  0.521
## [3,] 2 -1.55  0.506
## [4,] 3 -1.74  0.432
## [5,] 4 -1.75  0.429
## [6,] 5 -1.82  0.401
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,] 0 -1.77  0.674
## [2,] 1 -2.15  0.513
## [3,] 2 -2.23  0.481
## [4,] 3 -2.45  0.385
## [5,] 4 -2.53  0.353
## [6,] 5 -2.74  0.263
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01

```

```
adf.test(log_oil)
```

```

## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,] 0 0.717  0.850
## [2,] 1 0.582  0.811
## [3,] 2 0.665  0.835
## [4,] 3 0.527  0.796
## [5,] 4 0.539  0.799
## [6,] 5 0.405  0.761
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,] 0 -1.30  0.596
## [2,] 1 -1.46  0.539
## [3,] 2 -1.34  0.580
## [4,] 3 -1.63  0.477
## [5,] 4 -1.48  0.533
## [6,] 5 -1.46  0.539
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,] 0 -2.03  0.563
## [2,] 1 -2.35  0.428
## [3,] 2 -2.16  0.509
## [4,] 3 -2.54  0.350
## [5,] 4 -2.42  0.399
## [6,] 5 -2.63  0.308
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01

```

```
adf.test(d_log_oil)
```

```

## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag      ADF p.value
## [1,] 0 -20.30    0.01
## [2,] 1 -16.56    0.01
## [3,] 2 -11.36    0.01
## [4,] 3 -10.77    0.01
## [5,] 4 -9.22     0.01
## [6,] 5 -9.19     0.01
## Type 2: with drift no trend
##      lag      ADF p.value
## [1,] 0 -20.31    0.01
## [2,] 1 -16.58    0.01
## [3,] 2 -11.38    0.01
## [4,] 3 -10.79    0.01
## [5,] 4 -9.23     0.01
## [6,] 5 -9.21     0.01
## Type 3: with drift and trend
##      lag      ADF p.value
## [1,] 0 -20.29    0.01
## [2,] 1 -16.57    0.01
## [3,] 2 -11.37    0.01
## [4,] 3 -10.78    0.01
## [5,] 4 -9.22     0.01
## [6,] 5 -9.20     0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01

adf.test(d_d_log_oil)

```

```

## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag      ADF p.value
## [1,] 0 -12.80    0.01
## [2,] 1 -15.34    0.01
## [3,] 2 -9.09     0.01
## [4,] 3 -10.93    0.01
## [5,] 4 -8.12     0.01
## [6,] 5 -9.56     0.01
## Type 2: with drift no trend
##      lag      ADF p.value
## [1,] 0 -12.81    0.01
## [2,] 1 -15.37    0.01
## [3,] 2 -9.11     0.01
## [4,] 3 -10.95    0.01
## [5,] 4 -8.14     0.01
## [6,] 5 -9.59     0.01
## Type 3: with drift and trend
##      lag      ADF p.value
## [1,] 0 -12.80    0.01

```

```

## [2,] 1 -15.36 0.01
## [3,] 2 -9.10 0.01
## [4,] 3 -10.94 0.01
## [5,] 4 -8.13 0.01
## [6,] 5 -9.58 0.01
## -----
## Note: in fact, p.value = 0.01 means p.value <= 0.01

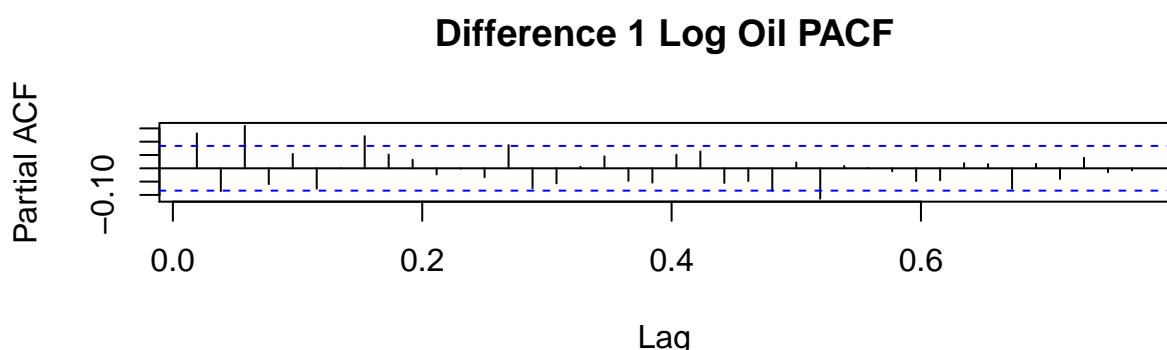
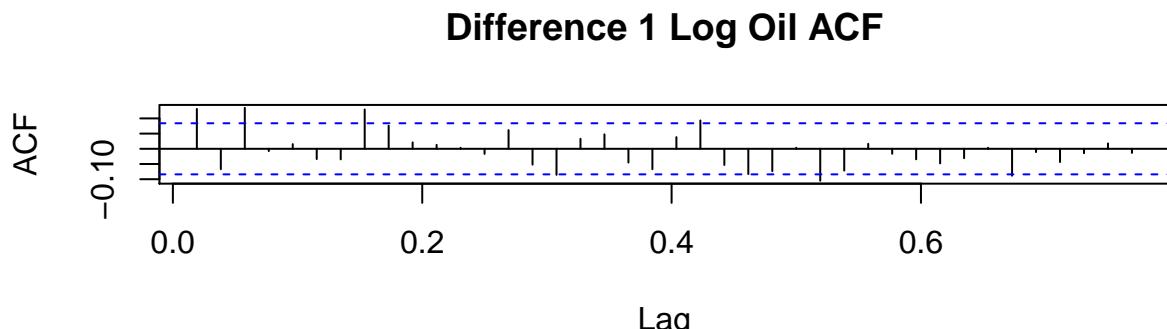
```

The augmented Dickey-Fuller test indicates that we perform differencing to make the data stationary.

```

par(mfrow=c(2, 1))
acf(d_log_oil, lag.max=40, main="Difference 1 Log Oil ACF")
pacf(d_log_oil, lag.max=40, main="Difference 1 Log Oil PACF")

```

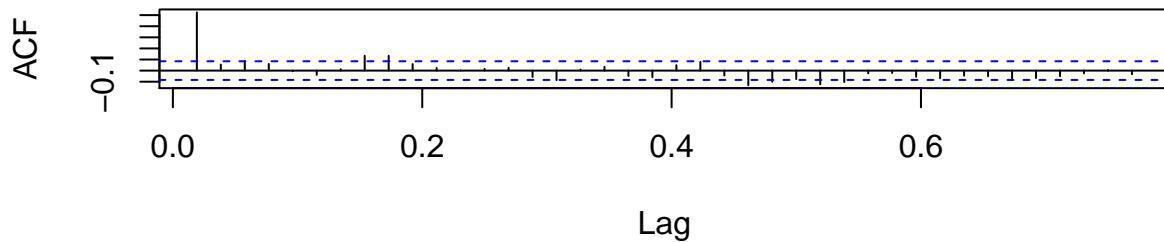


```

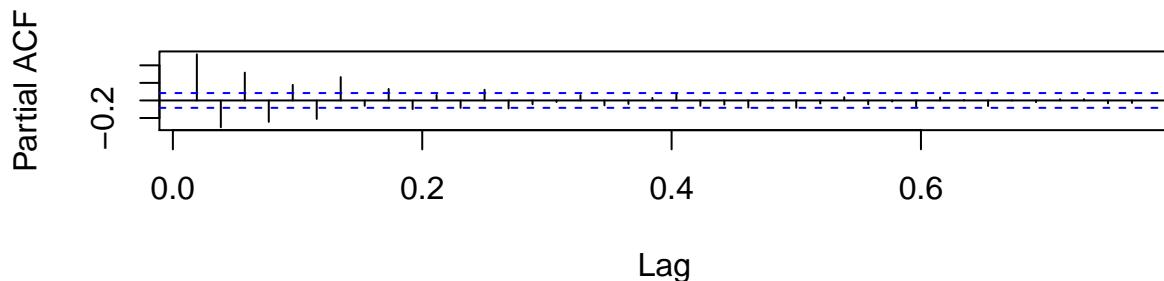
acf(d_d_log_oil, lag.max=40, main="Difference 2 Log Oil ACF")
pacf(d_d_log_oil, lag.max=40, main="Difference 2 Log Oil PACF")

```

Difference 2 Log Oil ACF



Difference 2 Log Oil PACF



ACF and PACF for the 1st order differencing do not have any clearly defined patterns compared to the 2nd order differencing, where it is observed that ACF cuts off at lag 1 and trails off in the PACF suggesting an ARIMA(0,2,1) model.

Computing the sample extended ACF: EACF analysis

```
eacf(d_log_oil)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o x o o o o x o o o o o o o
## 1 x o x o o o o o x o o o o o o o
## 2 x x x o o o o o x o o o o o o o
## 3 x x x o o o o o x o o o o o o o
## 4 x o x o o o o o x o o o o o o o
## 5 x x x o x o o x o o o o o o o
## 6 o x x o x x o x o o o o o o x
## 7 o x x x x x x x o x o o o o o o
```

```
eacf(d_d_log_oil)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o o o o o o x x o o o o o o o
## 1 x x o o o o o o x x o o o o o o
```

```

## 2 x x x o o o o x x o o   o   o   o
## 3 x x x x o o o o x x o o   o   o   o
## 4 x x x x o o o o x x o o   o   o   o
## 5 x o x x x o o o x x o o   o   o   o
## 6 x o x x x x x x o x o   o   o   o
## 7 x x x x x x x o x o   o   o   o

```

The EACF matrix for 1st order differencing make a general pattern of rectangles as compared to triangles. The EACF matrix for the 2nd order differencing has the triangular pattern with a point at AR(1) which confirms the previous statements in the ACF.

```

ljungbox<- function(data) {
print(Box.test(data, lag = 1, type = "Ljung-Box"))
}
ljungbox(oil)

##
## Box-Ljung test
##
## data: data
## X-squared = 541.1, df = 1, p-value < 2.2e-16

fit1 <- Arima(log_oil, order=c(1, 1, 1))
fit1

## Series: log_oil
## ARIMA(1,1,1)
##
## Coefficients:
##             ar1      ma1
##           -0.5253  0.7142
## s.e.    0.0872  0.0683
##
## sigma^2 estimated as 0.002112: log likelihood=904.58
## AIC=-1803.15  AICc=-1803.11  BIC=-1790.25

fit2 <- Arima(log_oil, order=c(1, 2, 2))
fit2

## Series: log_oil
## ARIMA(1,2,2)
##
## Coefficients:
##             ar1      ma1      ma2
##           -0.5247  -0.2861  -0.7139
## s.e.    0.0873  0.0686  0.0685
##
## sigma^2 estimated as 0.002118: log likelihood=899.7
## AIC=-1791.4  AICc=-1791.32  BIC=-1774.21

```

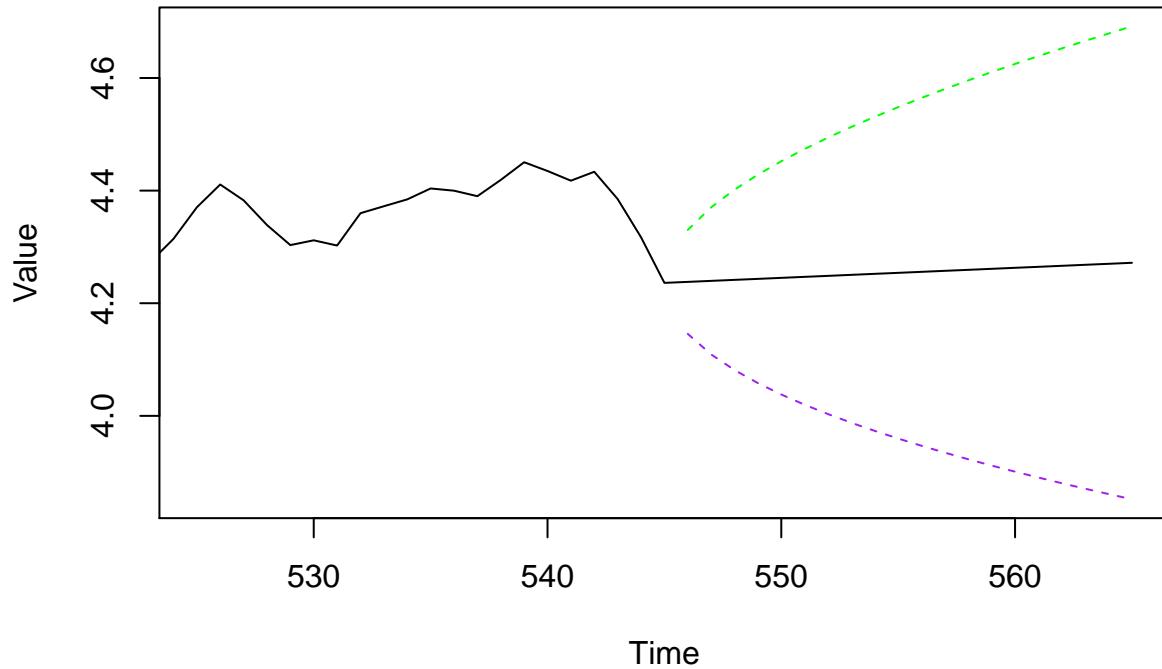
```

fit3 <- Arima(log_oil, order=c(0, 2, 1))
fit3

## Series: log_oil
## ARIMA(0,2,1)
##
## Coefficients:
##             ma1
##             -1.0000
## s.e.    0.0061
##
## sigma^2 estimated as 0.002213:  log likelihood=886.63
## AIC=-1769.26   AICc=-1769.24   BIC=-1760.67

fit_plot <- function(model) {
  pred <- predict(model, n.ahead=20, se.fit=TRUE)
  upper_band <- pred$pred + 1.96 * pred$se
  lower_band <- pred$pred - 1.96 * pred$se
  n <- length(model$x)
  plot(c(model$x, pred$pred), type="l",
       xlim=c(n - 20, n + 20),
       ylim=c(min(lower_band), max(upper_band)), ylab="Value", xlab="Time")
  lines(n + 1:20, upper_band, lty=2, col="green")
  lines(n + 1:20, lower_band, lty=2, col="purple")
}
fit_plot(fit3)

```



The best model if we evaluate the BIC criterion we choose the ARIMA(0,2,1) which looks like following

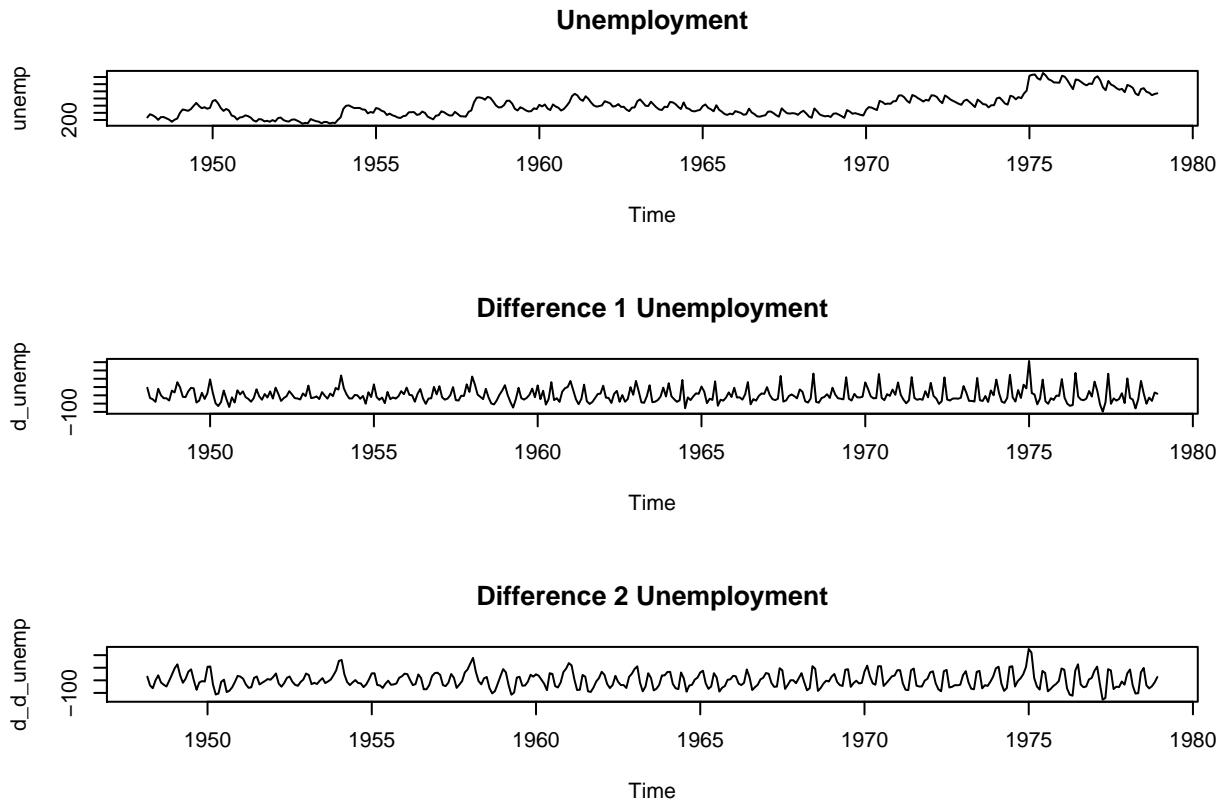
$$\nabla^2 x_t = (1 + \theta B) w_t$$

The fig. above displays a prediction of 20 timesteps. The prediction is marked by a kink and a linear prediction value. The intervals also look reasonable

Question 2

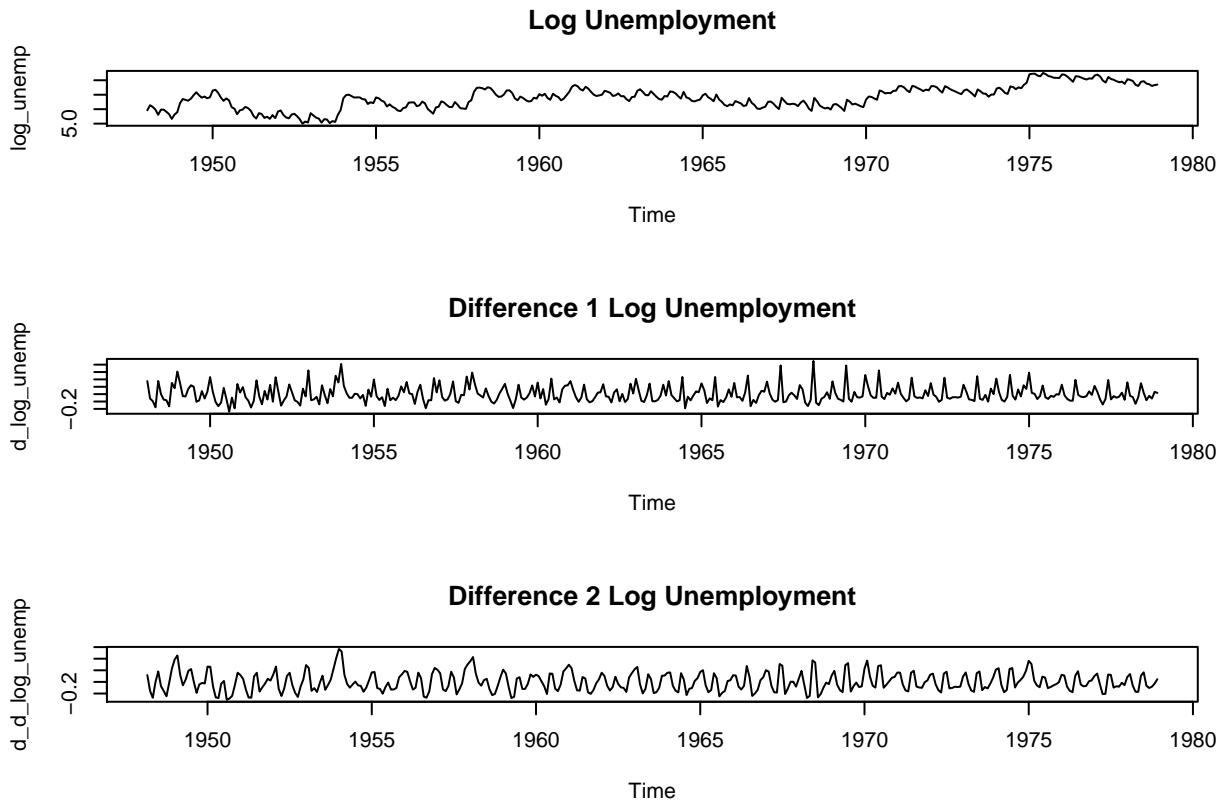
```
log_unemp <- log(unemp)
d_unemp <- diff(unemp)
d_d_unemp <- diff(unemp, 2)
d_log_unemp <- diff(log_unemp)
d_d_log_unemp <- diff(log_unemp, 2)

par(mfrow=c(3, 1))
plot(unemp, main="Unemployment")
plot(d_unemp, main="Difference 1 Unemployment")
plot(d_d_unemp, main="Difference 2 Unemployment")
```



```
par(mfrow=c(3, 1))
```

```
par(mfrow=c(3, 1))
plot(log_unemp, main="Log Unemployment")
plot(d_log_unemp, main="Difference 1 Log Unemployment")
plot(d_d_log_unemp, main="Difference 2 Log Unemployment")
```

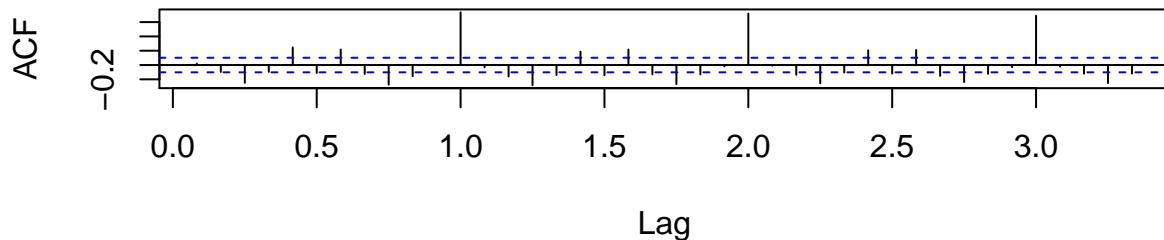


```
par(mfrow=c(3, 1))
```

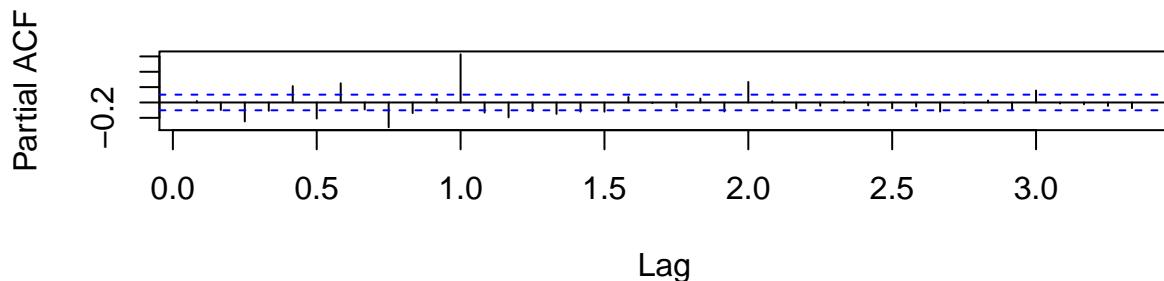
The data is not stationary visually and hence, differencing is done by the order of 1. Variance seems to be increasing with time which is reduced on transformation with the log scale. Differencing of the order 2 gives a smoother result.

```
par(mfrow=c(2, 1))
acf(d_log_unemp, lag.max=40, main="Difference 1 Log Unemployment ACF")
pacf(d_log_unemp, lag.max=40, main="Difference 1 Log Unemployment PACF")
```

Difference 1 Log Unemployment ACF

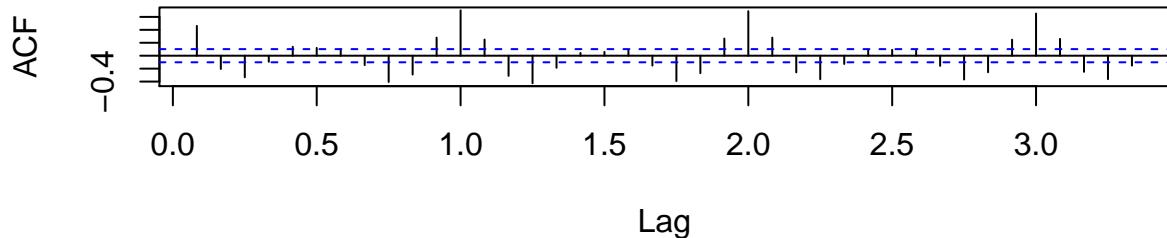


Difference 1 Log Unemployment PACF

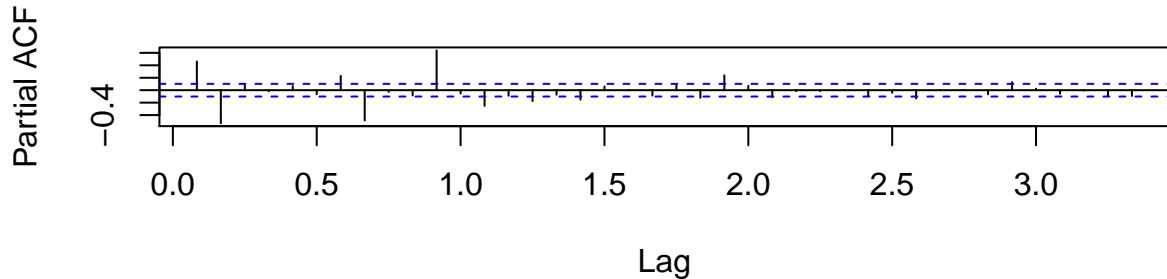


```
acf(d_d_log_unemp, lag.max=40, main="Difference 2 Log Unemployment ACF")
pacf(d_d_log_unemp, lag.max=40, main="Difference 2 Log Unemployment PACF")
```

Difference 2 Log Unemployment ACF



Difference 2 Log Unemployment PACF



Difference 1 Seasonality behavior: The ACF plot suggests seasonality at 12 lags that tails off both in the ACF and the PACF. This suggests an $ARMA_{1,2}$ seasonality component. The PACF spikes at 3 multiples which is indicative of AR(3). **Non-seasonality behavior:** There is no distinct non-seasonal pattern to be seen. Our model for this data is SARMA(3,1,0)₁₂.

Difference 2 Seasonality behavior: Large spikes at lags 9, 12, 15 are observed and then it fades till the pattern repeats at lags 21, 24, 27. So there is definitely a seasonality pattern in the data. The PACF shows that the pattern tails off over time with spikes at two 2 multiples which indicates AR(2).

Non-seasonality behavior: Apart from the seasonal behavior the ACF tails off and the PACF cuts off after lag 2 indicating an AR(2) model.

```
adf.test(d_log_oil)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag      ADF p.value
## [1,]   0 -20.30   0.01
## [2,]   1 -16.56   0.01
## [3,]   2 -11.36   0.01
## [4,]   3 -10.77   0.01
## [5,]   4  -9.22   0.01
## [6,]   5  -9.19   0.01
## Type 2: with drift no trend
##      lag      ADF p.value
```

```

## [1,] 0 -20.31 0.01
## [2,] 1 -16.58 0.01
## [3,] 2 -11.38 0.01
## [4,] 3 -10.79 0.01
## [5,] 4 -9.23 0.01
## [6,] 5 -9.21 0.01
## Type 3: with drift and trend
##      lag ADF p.value
## [1,] 0 -20.29 0.01
## [2,] 1 -16.57 0.01
## [3,] 2 -11.37 0.01
## [4,] 3 -10.78 0.01
## [5,] 4 -9.22 0.01
## [6,] 5 -9.20 0.01
## -----
## Note: in fact, p.value = 0.01 means p.value <= 0.01

adf.test(d_d_log_oil)

## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag ADF p.value
## [1,] 0 -12.80 0.01
## [2,] 1 -15.34 0.01
## [3,] 2 -9.09 0.01
## [4,] 3 -10.93 0.01
## [5,] 4 -8.12 0.01
## [6,] 5 -9.56 0.01
## Type 2: with drift no trend
##      lag ADF p.value
## [1,] 0 -12.81 0.01
## [2,] 1 -15.37 0.01
## [3,] 2 -9.11 0.01
## [4,] 3 -10.95 0.01
## [5,] 4 -8.14 0.01
## [6,] 5 -9.59 0.01
## Type 3: with drift and trend
##      lag ADF p.value
## [1,] 0 -12.80 0.01
## [2,] 1 -15.36 0.01
## [3,] 2 -9.10 0.01
## [4,] 3 -10.94 0.01
## [5,] 4 -8.13 0.01
## [6,] 5 -9.58 0.01
## -----
## Note: in fact, p.value = 0.01 means p.value <= 0.01

```

The EACF suggests that there no ARMA model well suited to this data.

```

fit1 <- Arima(log_unemp, order=c(0, 0, 0), seasonal=c(3, 1, 0))
fit1

## Series: log_unemp
## ARIMA(0,0,0)(3,1,0) [12]
##
## Coefficients:
##             sar1      sar2      sar3
##           -0.2589   -0.2801   -0.2872
## s.e.     0.0522    0.0526    0.0547
##
## sigma^2 estimated as 0.06287:  log likelihood=-13.62
## AIC=35.24    AICc=35.36    BIC=50.79

fit2 <- Arima(log_unemp, order=c(0, 0, 0), seasonal=c(2, 2, 0))
fit2

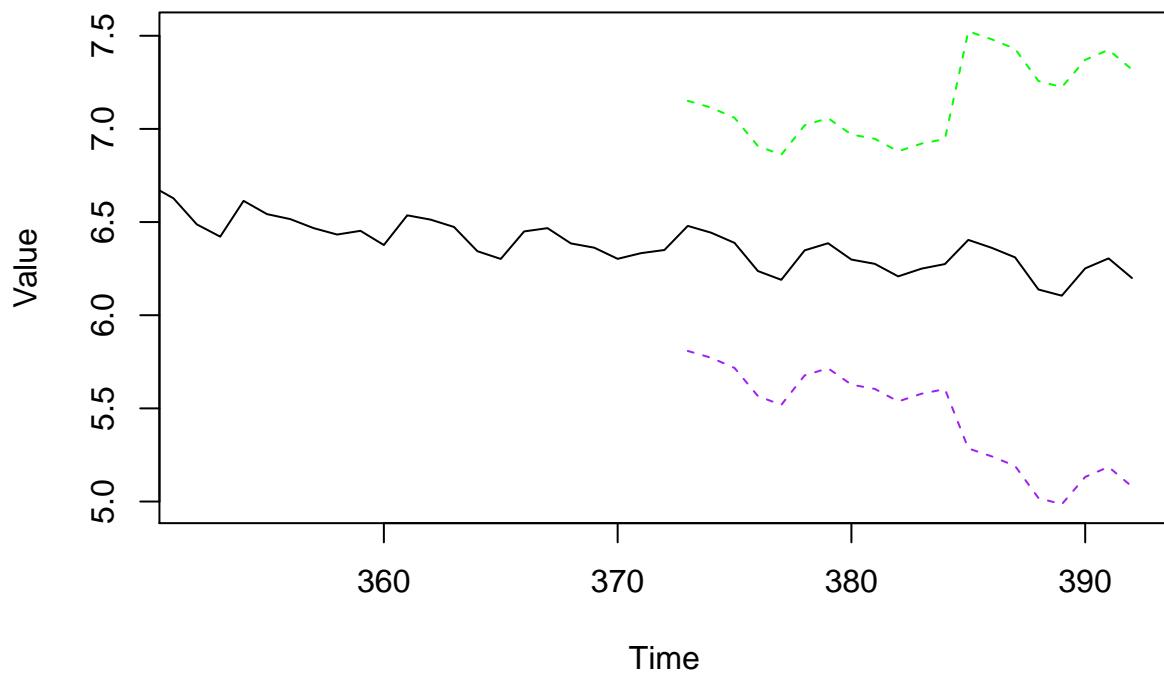
## Series: log_unemp
## ARIMA(0,0,0)(2,2,0) [12]
##
## Coefficients:
##             sar1      sar2
##           -0.6655   -0.3370
## s.e.     0.0548    0.0553
##
## sigma^2 estimated as 0.1173:  log likelihood=-123.05
## AIC=252.1    AICc=252.17    BIC=263.66

```

The second fit has better BIC/AIC which is then our final model. It can be written formally as

$$(1 + 0.6655B + 0.3370B^2) \nabla^2 x_t = w_t.$$

```
fit_plot(fit2)
```



Time series Lab 3

Omkar Bhutra (omkbh878)

12 October 2019

Assignment 1:

In table 1 a script for generation of data from simulation of the following state space model and implementation of the Kalman filter on the data is given. Kalman filter: $z_t = A_{t-1}z_{t-1} + e_t$, $x_t = C_t z_t + v_t$, $v_t \sim N(0, R_t)$, $e_t \sim N(0, Q_t)$

Prediction step: $m_{t+1|t} = A_t m_{t|t}$ $P_{t+1|t} = A_t P_{t|t} A_t^T + Q_{t+1}$

Observation Update: $G_t = \frac{P_t C^T}{A P_t A^T + R} = \frac{P_{t|t} A_t^T}{P_{t+1|t}}$ $\tilde{m}_{t|t} = m_t + G_t(z_{t+1} - A m_{t|t})$ $\tilde{P}_t = P_{t|t} - G_t(A P_{t|t} A^T + Q) G_t^T = P_{t|t} - G_t P_{t+1|t} G_t^T$

a. Write down the expression for the state space model that is being simulated.

observation update using $x_t = C_t z_t + v_t \sim N(z_t; m_{t|t}, P_{t|t})$ prediction using $z_{t+1} = Az_t + e_{t+1} \sim N(z_t; m_{t+1|t}, P_{t+1|t})$ $\theta = \{A, C, R, Q, m_0, P_0\} = \{1, 1, 1, 1, 0, 1\}$

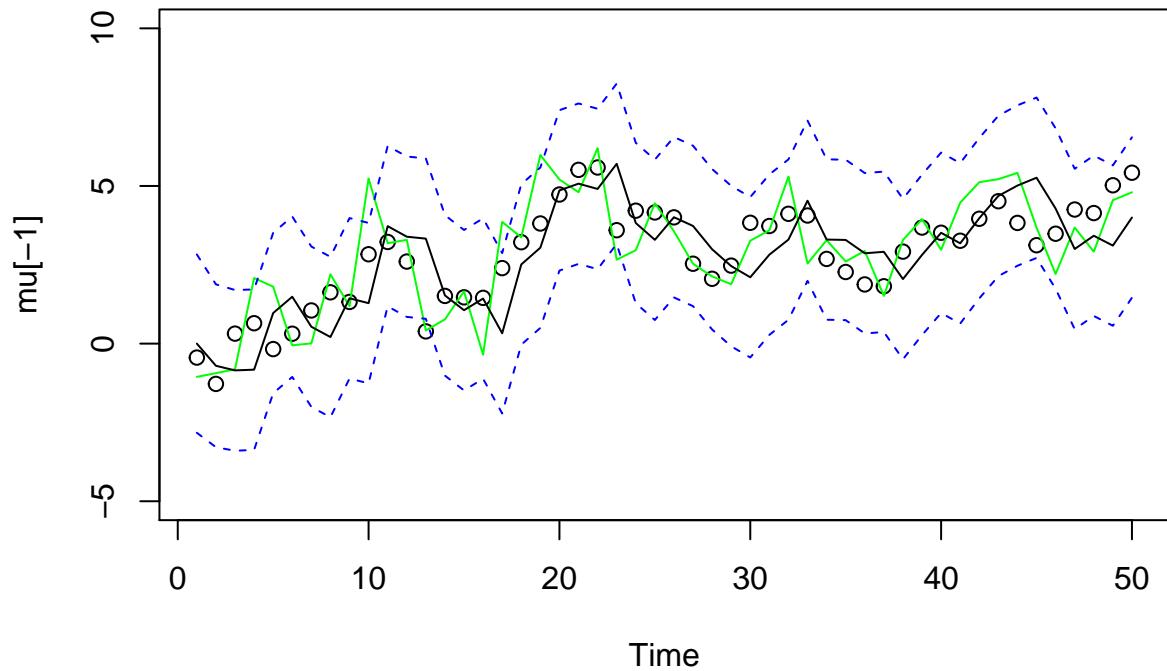
b. Run this script and compare the filtering results with a moving average smoother of order 5.

```
# generate data
set.seed(1); num = 50

smoother<-function(x,n){
  stats::filter(as.vector(x),rep(1/n,n),sides = 2)
}

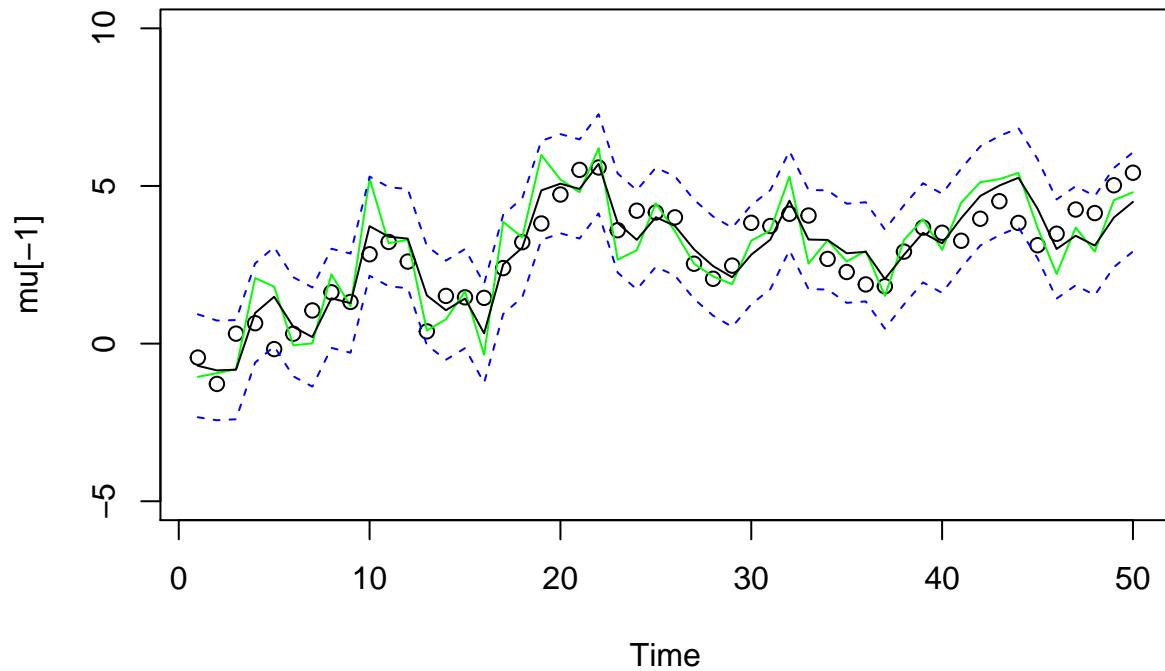
w = rnorm(num+1,0,1); v = rnorm(num ,0,1)
mu = cumsum(w) # state : mu[0], mu[1] ,... , mu[50]
y = mu[-1] + v # obs: y[1] ,... , y[50]
# filter and smooth ( Ksmooth O does both )
ks = Ksmooth0(num , y, A=1, mu0=0, Sigma0=1, Phi=1, cQ=1, cR=1)
# start figure
Time = 1:num
plot(Time , mu[-1], main ='Predict ', ylim =c(-5,10))
lines(Time ,y,col=" green ")
lines(ks$xp)
lines(ks$xp+2* sqrt(ks$Pp), lty =2, col=4)
lines(ks$xp -2* sqrt(ks$Pp), lty =2, col=4)
```

Predict



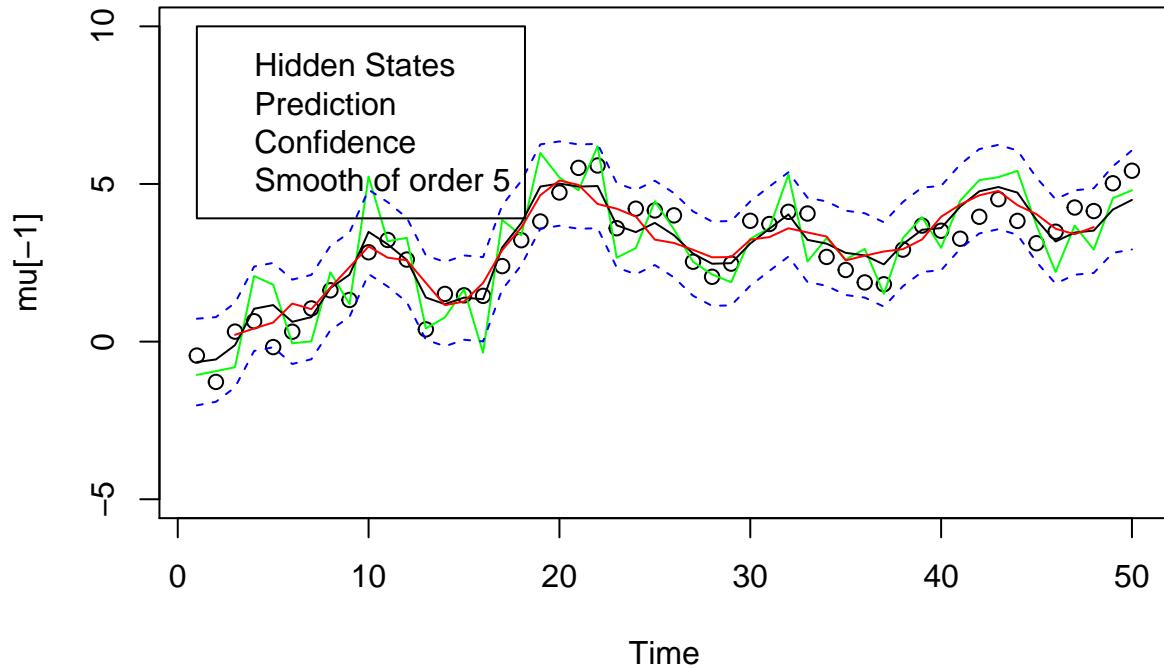
```
plot(Time , mu[-1], main ='Filter ', ylim =c(-5,10))
lines(Time ,y,col=" green ")
lines(ks$xf)
lines(ks$xf+2* sqrt(ks$Pf), lty =2, col=4)
lines(ks$xf -2* sqrt(ks$Pf), lty =2, col=4)
```

Filter



```
plot(Time , mu[-1], main ='Smooth ', ylim =c(-5,10))
lines(Time ,y,col=" green ")
lines(ks$xs)
lines(ks$xs+2* sqrt(ks$Ps), lty =2, col=4)
lines(ks$xs -2* sqrt(ks$Ps), lty =2, col=4)
lines(Time,smoother(y,5),col = "red") # Moving average smoother order 5
legend(1, 10, legend = c("Hidden States", "Prediction",
"Confidence", "Smooth of order 5"), col = c("black", "green",
"blue", "red"))
```

Smooth



```

mu[1]; #initial mean smoother

## [1] -0.6264538

ks$xOn; #initial smoother covariance

##          [,1]
## [1,] -0.3241541

sqrt(ks$P0n) # initial value info

##          [,1]
## [1,] 0.7861514

```

It can be visually observed that the prediction is most volatile but stays within the confidence intervals. Filtering reduces the volatility and the moving average smoothing of order 5 which is seen as the red line is the smoothest of all the functions.

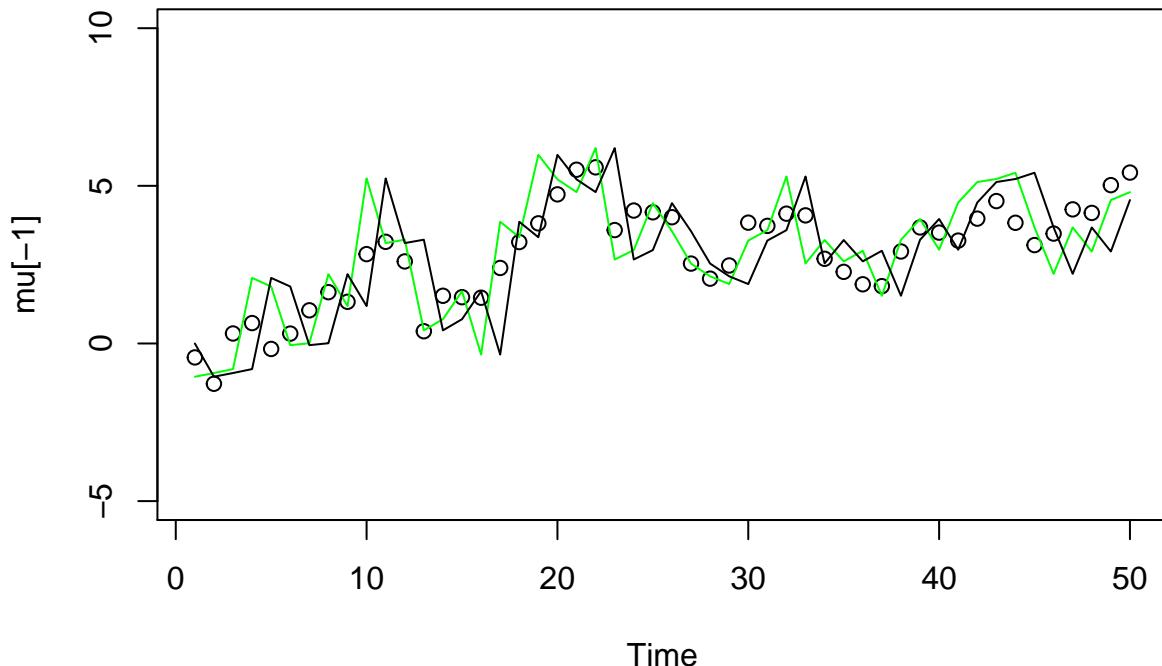
c) Also, compare the filtering outcome when R in the filter is 10 times smaller than its actual value while Q in the filter is 10 times larger than its actual value. How does the filtering outcome vary?

```

set.seed(1)
num = 50
w = rnorm(num+1,0,1)
v = rnorm(num ,0,1)
mu = cumsum(w) # state : mu[0] , mu[1] ,... , mu[50]
y = mu[-1] + v # obs: y[1] ,..., y[50]
# filter and smooth ( Ksmooth 0 does both )
ks = Ksmooth0(num,y,A=1, mu0=0, Sigma0=1, Phi=1, cQ=10, cR = .1)
# start figure
par(mfrow=c(1,1))
Time = 1: num
# predicted
plot(Time , mu[-1] , main='Predict',ylim=c(-5,10))
lines(Time,y,col="green")
lines(ks$xp)
lines(ks$xp+2*sqrt(ks$Pp) , lty =2, col=4)
lines(ks$xp-2*sqrt(ks$Pp) ,lty=2,col=4)

```

Predict

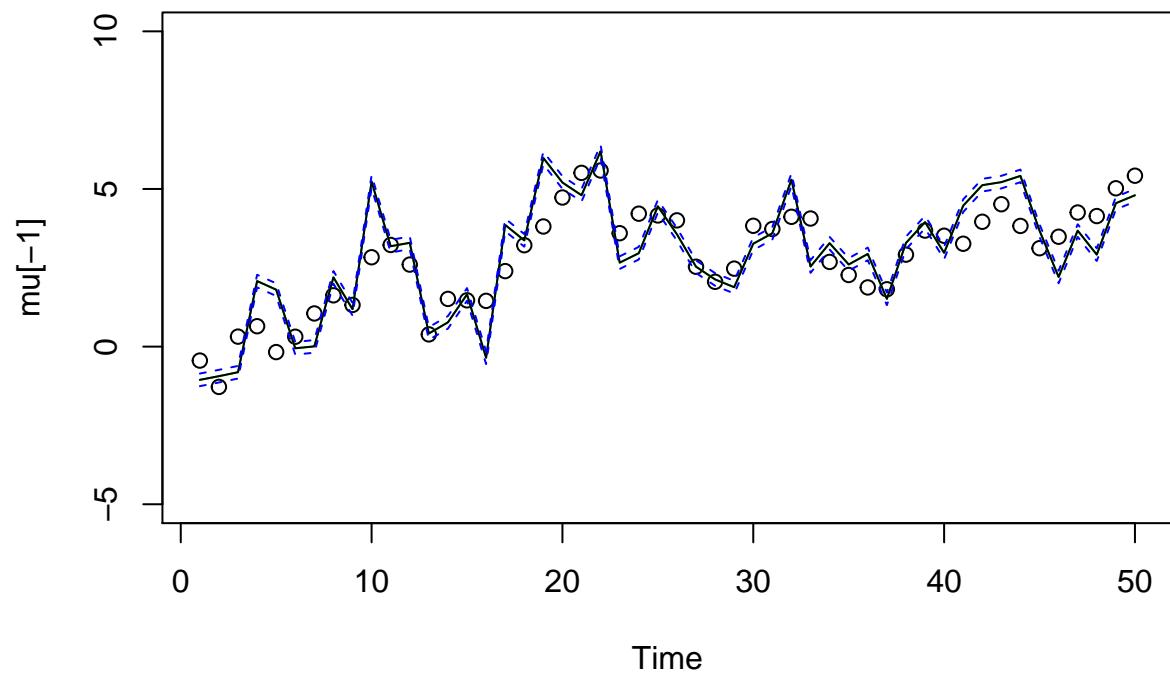


```

# Filter
plot(Time,mu[ -1] , main ='Filter',ylim=c(-5,10))
lines(Time ,y,col="green")
lines(ks$xf )
lines(ks$xf+2*sqrt(ks$Pf) , lty =2, col=4)
lines(ks$xf-2*sqrt(ks$Pf) , lty =2, col=4)

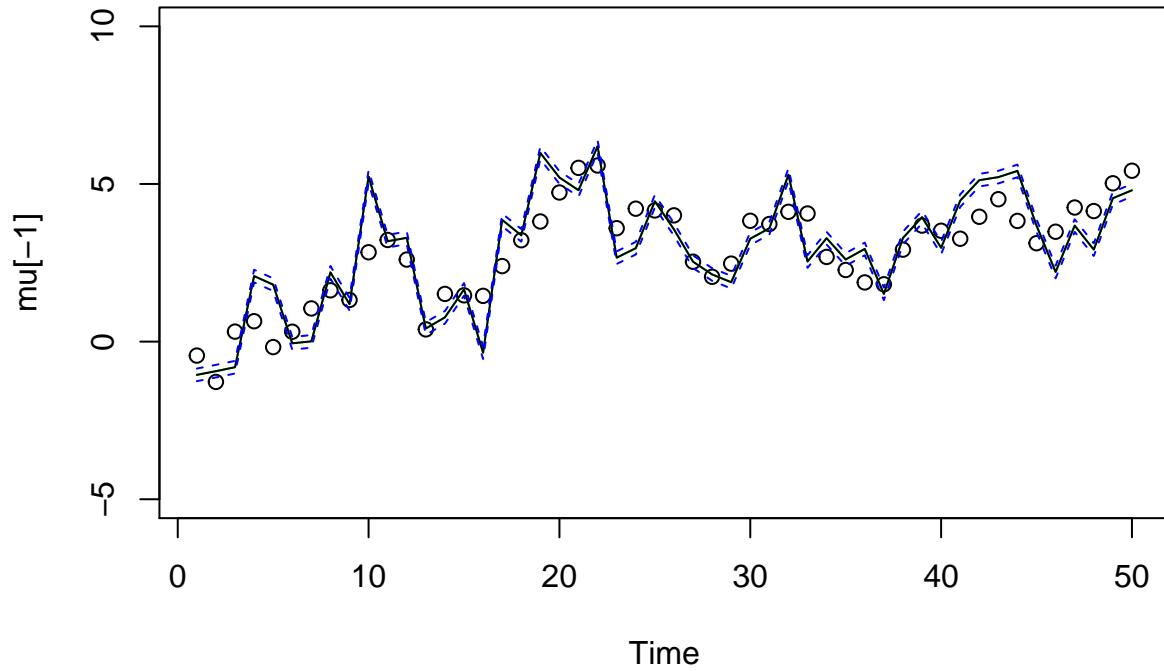
```

Filter



```
# Smooth
plot(Time , mu[-1] , main ='Smooth', ylim =c( -5,10))
lines(Time ,y ,col="green")
lines(ks$xs )
lines(ks$xs+2*sqrt(ks$Ps) , lty =2, col=4)
lines(ks$xs-2*sqrt(ks$Ps) , lty =2, col=4)
```

Smooth



```
#Initial mean smoother
mu[1]

## [1] -0.6264538

#Initial smoother covariance
ks$x0n

## [,1]
## [1,] -0.01044278

#Initial value info
sqrt(ks$P0n)

## [,1]
## [1,] 0.9950377
```

The data and filtering model as in 1b is taken but the R in the filter is 10 times smaller and Q in the filter is larger by 10 times, both compared to their actual values. The filtering and smoothed lines are closely similar.

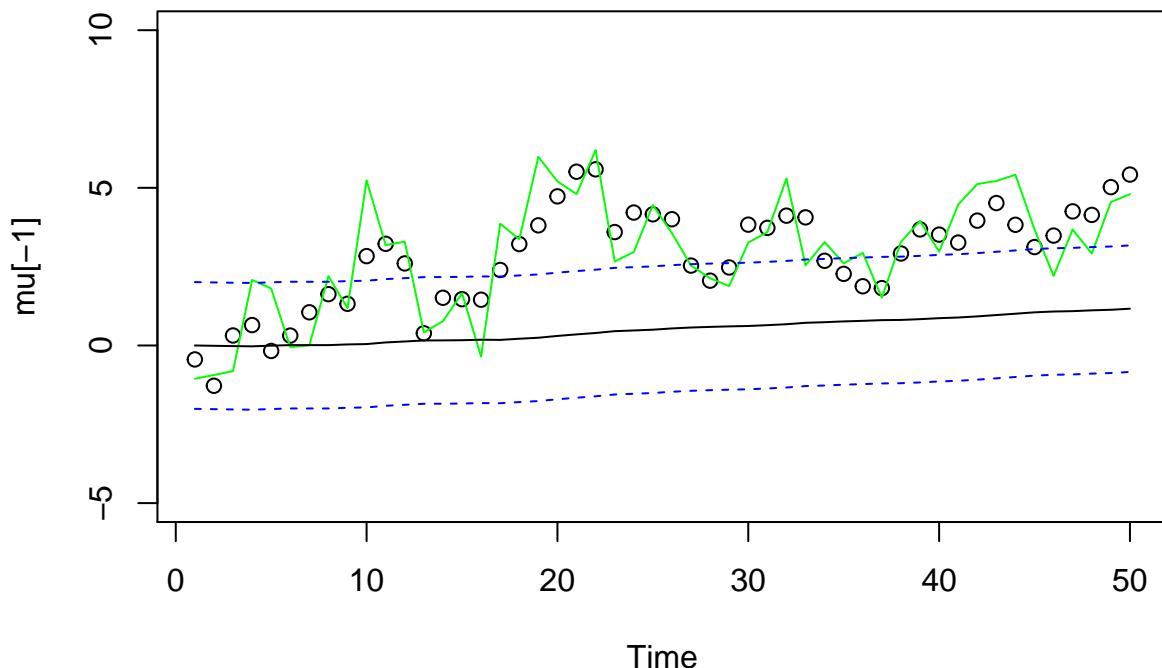
d. Now compare the filtering outcome when R in the filter is 10 times larger than its actual value while Q in the filter is 10 times smaller than its actual value. How does the filtering outcome varies?

```

set.seed(1)
num = 50
w = rnorm(num+1,0,1)
v = rnorm(num ,0,1)
mu = cumsum(w) # state : mu[0] , mu[1] ,... , mu[50]
y = mu[-1] + v # obs: y[1] ,..., y[50]
# filter and smooth ( Ksmooth 0 does both )
ks = Ksmooth0(num,y,A=1, mu0=0, Sigma0=1, Phi=1, cQ=0.10, cR =10)
# start figure
par(mfrow=c(1,1))
Time = 1: num
# predicted
plot(Time , mu[-1] , main='Predict',ylim=c(-5,10))
lines(Time,y,col="green")
lines(ks$xp)
lines(ks$xp+2*sqrt(ks$Pp) , lty =2, col=4)
lines(ks$xp-2*sqrt(ks$Pp) ,lty=2,col=4)

```

Predict

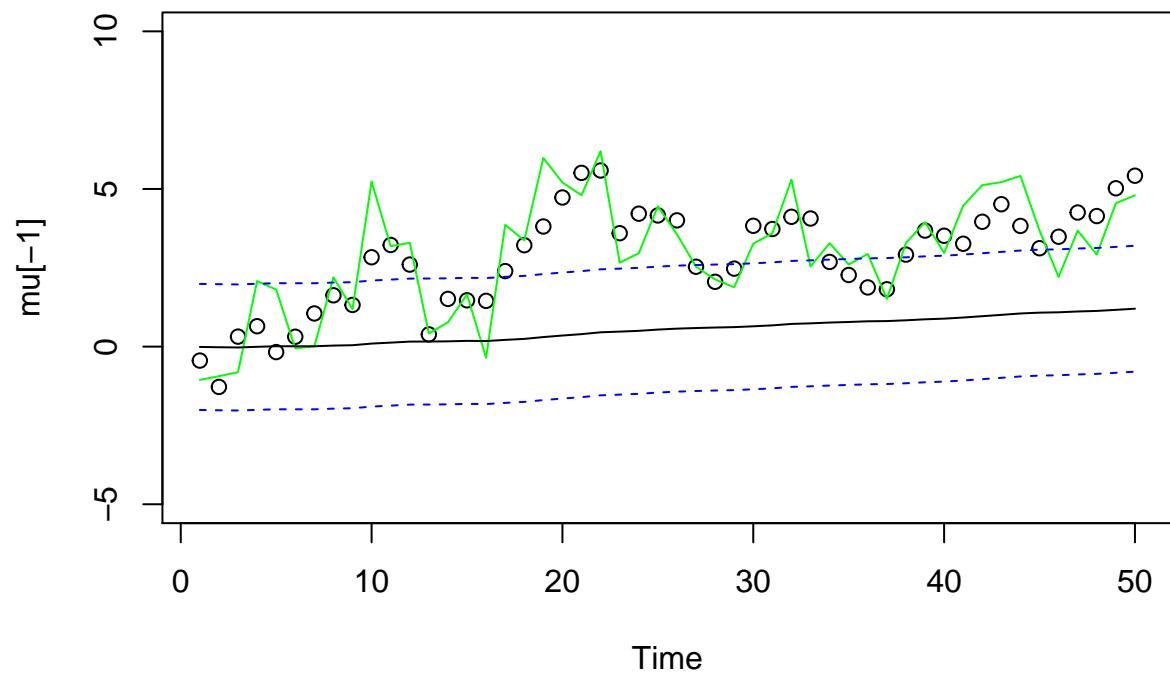


```

# Filter
plot(Time,mu[ -1] , main ='Filter',ylim=c(-5,10))
lines(Time ,y,col="green")
lines(ks$xf )
lines(ks$xf+2*sqrt(ks$Pf) , lty =2, col=4)
lines(ks$xf-2*sqrt(ks$Pf) , lty =2, col=4)

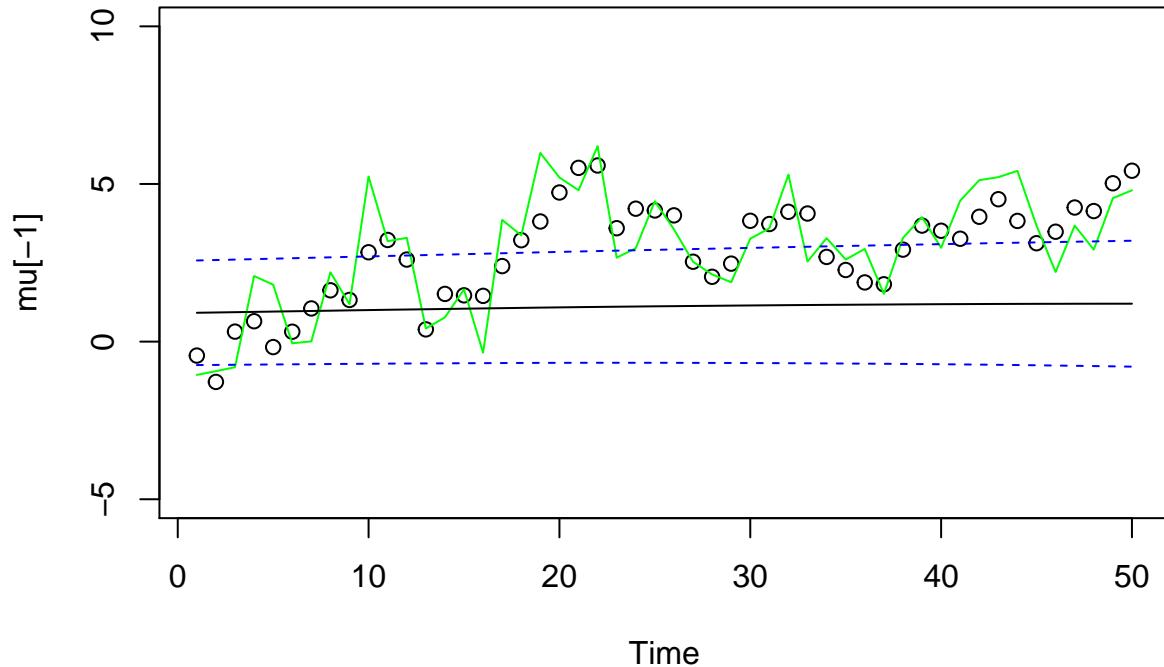
```

Filter



```
# Smooth
plot(Time , mu[-1] , main ='Smooth', ylim =c( -5,10))
lines(Time ,y ,col="green")
lines(ks$xs )
lines(ks$xs+2*sqrt(ks$Ps) , lty =2, col=4)
lines(ks$xs-2*sqrt(ks$Ps) , lty =2, col=4)
```

Smooth



```
#Initial mean smoother
mu[1]

## [1] -0.6264538

#Initial smoother covariance
ks$x0n

##          [,1]
## [1,] 0.905755

#Initial value info
sqrt(ks$P0n)

##          [,1]
## [1,] 0.82731
```

The data and filtering model as in 1b is taken but the Q in the filter is 10 times smaller and R in the filter is larger by 10 times, both compared to their actual values. The model considers data as error when the distance from the observations to the line are smaller than 10.

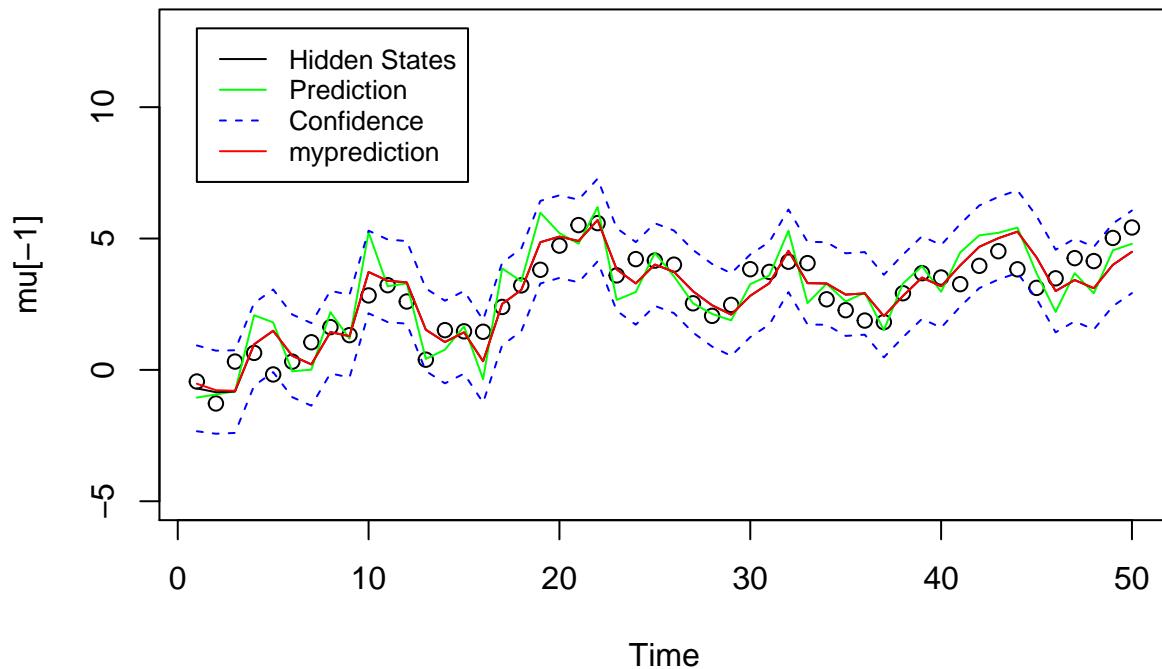
1e. Implement your own Kalman filter and replace ksmooth0 function with your script.

```

kalmanfilter <- function(num, y, A, mu0, Sigma0, Phi, cQ, cR){
# init
T <- num
xs <- rep(0,T)
Ps <- rep(0,T)
I <- diag(1,dim(matrix(mu0))[1])
for(t in 1:T){
  #obs update step
  K <- Sigma0%*% t(Phi) %*% solve(Phi%*%Sigma0%*%t(Phi)+cR)
  xs[t] <- mu0 + K%*%(y[t] - Phi%*%mu0)
  Ps[t] <- (I - K%*%Phi)%*%Sigma0
  #prediction step
  mu0 <- A%*%xs[t]
  Sigma0 <- A%*%Ps[t]%*%t(A)+cQ
}
filteroutput <- list()
filteroutput$xs <- xs
filteroutput$Ps <- Ps
filteroutput$x0n <- mu0
filteroutput$P0n <- Sigma0
return(filteroutput)
}
# Q=1, R=1
ourkf <- kalmanfilter(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,
cQ = 1, cR = 1)
ks = Ksmooth0(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,
cQ = 1, cR = 1)
plot(Time, mu[-1], main = "Filter Q=1 R=1", ylim = c(-5,13))
lines(Time, y, col = "green")
lines(ks$xf)
lines(ks$xf + 2 * sqrt(ks$Pf), lty = 2, col = 4)
lines(ks$xf - 2 * sqrt(ks$Pf), lty = 2, col = 4)
lines(ourkf$xs, col = "red")
legend(1, 13, legend = c("Hidden States", "Prediction", "Confidence", "myprediction"), col = c("black",

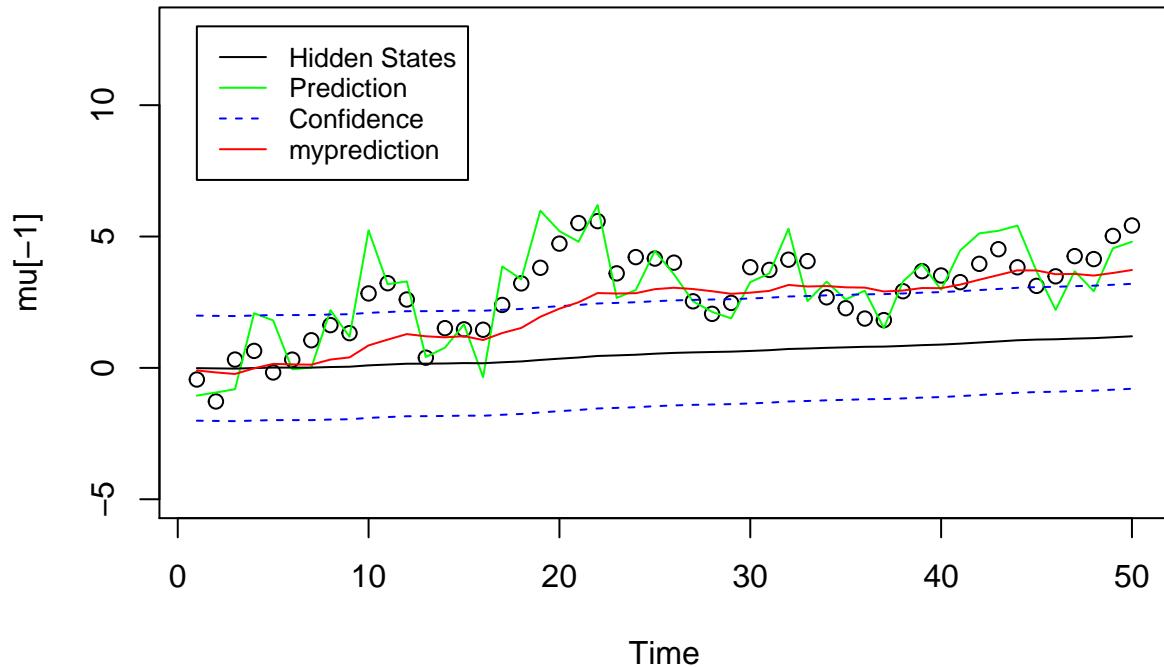
```

Filter Q=1 R=1



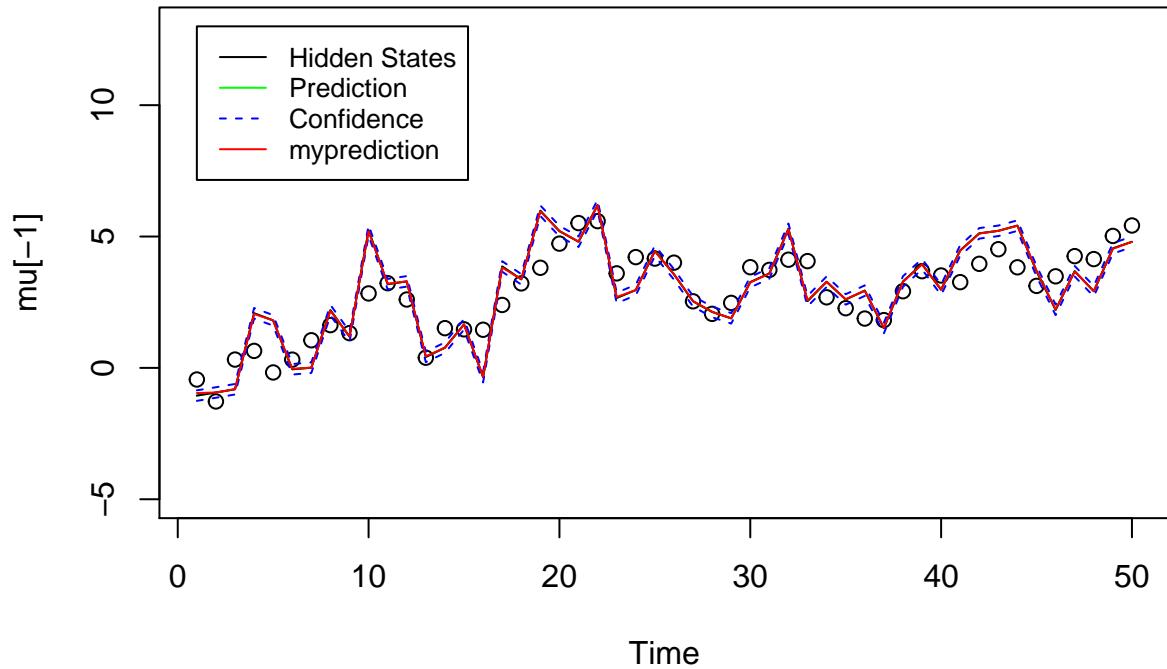
```
# Q=0.10,R=10
ourkf1 <- kalmanfilter(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = .1, cR = 10)
ks1 = Ksmooth0(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = .1, cR = 10)
plot(Time, mu[-1], main = "Filter Q=.1 R=10", ylim = c(-5,13))
lines(Time, y, col = "green")
lines(ks1$xf)
lines(ks1$xf + 2 * sqrt(ks1$Pf), lty = 2, col = 4)
lines(ks1$xf - 2 * sqrt(ks1$Pf), lty = 2, col = 4)
lines(ourkf1$xs, col = "red")
legend(1, 13, legend = c("Hidden States", "Prediction",
"Confidence", "myprediction"), col = c("black", "green",
"blue", "red"), lty = c(1, 1, 2), cex = 0.8)
```

Filter Q=.1 R=10



```
# Q=10,R=0.10
ourkf2 <- kalmanfilter(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = 10, cR = 0.1)
ks2 = Ksmooth0(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = 10, cR = 0.1)
plot(Time, mu[-1], main = "Filter Q=10 R=.1", ylim = c(-5,13))
lines(Time, y, col = "green")
lines(ks2$xf)
lines(ks2$xf + 2 * sqrt(ks2$Pf), lty = 2, col = 4)
lines(ks2$xf - 2 * sqrt(ks2$Pf), lty = 2, col = 4)
lines(ourkf2$xs, col = "red")
legend(1, 13, legend = c("Hidden States", "Prediction",
"Confidence", "myprediction"), col = c("black", "green",
"blue", "red"), lty = c(1, 1, 2), cex = 0.8)
```

Filter Q=10 R=.1



The kalman function was constructed according to the Lab questions provided. The function varies from the function ksmooth0 from the package astsa as it appears to be smoother than our implementation.

f. How do you interpret the Kalman gain?

The Kalman gain, G_t is expressing a belief parameter for the predicted prior $P_{t+1|t}$ accounting for the next state ($t + 1$). It is a value $\in (0, 1)$ and is computed as $G_t = \frac{P_t C^T}{AP_t A^T + R} = \frac{P_{t|t} A_t^T}{P_{t+1|t}}$. Magnitude of belief is decided by the autocovariance matrix. Higher uncertainty in the prior suggests that it should be believed lesser.

Robert H. Shumway
David S. Stoffer

Time Series Analysis and Its Applications

With R Examples

Fourth Edition



Springer

Robert H. Shumway
David S. Stoffer

Time Series Analysis and Its Applications

With R Examples

Fourth Edition



[live free or bark](#)

Preface to the Fourth Edition

The fourth edition follows the general layout of the third edition but includes some modernization of topics as well as the coverage of additional topics. The preface to the third edition—which follows—still applies, so we concentrate on the differences between the two editions here. As in the third edition, R code for each example is given in the text, even if the code is excruciatingly long. Most of the examples with seemingly endless coding are in the latter chapters. The R package for the text, `astsa`, is still supported and details may be found in [Appendix R](#). A number of data sets have been updated. For example, the global temperature deviation series have been updated to 2015 and are included in the newest version of the package; the corresponding examples and problems have been updated accordingly.

[Chapter 1](#) of this edition is similar to the previous edition, but we have included the definition of trend stationarity and the the concept of prewhitening when using cross-correlation. The New York Stock Exchange data set, which focused on an old financial crisis, was replaced with a more current series of the Dow Jones Industrial Average, which focuses on a newer financial crisis. In [Chapter 2](#), we rewrote some of the regression review, changed the smoothing examples from the mortality data example to the Southern Oscillation Index and finding El Niño. We also expanded the discussion of lagged regression to [Chapter 3](#) to include the possibility of autocorrelated errors.

In [Chapter 3](#), we removed normality from definition of ARMA models; while the assumption is not necessary for the definition, it is essential for inference and prediction. We added a section on regression with ARMA errors and the corresponding problems; this section was previously in [Chapter 5](#). Some of the examples have been modified and we added some examples in the seasonal ARMA section.

In [Chapter 4](#), we improved and added some examples. The idea of modulated series is discussed using the classic star magnitude data set. We moved some of the filtering section forward for easier access to information when needed. We removed the reliance on `spec.pgram` (from the `stats` package) to `mvspec` (from the `astsa` package) so we can avoid having to spend pages explaining the quirks of `spec.pgram`, which tended to take over the narrative. The section on wavelets was removed because

there are so many accessible texts available. The spectral representation theorems are discussed in a little more detail using examples based on simple harmonic processes.

The general layout of [Chapter 5](#) and of [Chapter 7](#) is the same, although we have revised some of the examples. As previously mentioned, we moved regression with ARMA errors to [Chapter 3](#).

[Chapter 6](#) sees the biggest change in this edition. We have added a section on smoothing splines, and a section on hidden Markov models and switching autoregressions. The Bayesian section is completely rewritten and is on linear Gaussian state space models only. The nonlinear material in the previous edition is removed because it was old, and the newer material is in Douc, Moulines, and Stoffer (2014). Many of the examples have been rewritten to make the chapter more accessible. Our goal was to be able to have a course on state space models based primarily on the material in [Chapter 6](#).

The Appendices are similar, with some minor changes to [Appendix A](#) and [Appendix B](#). We added material to [Appendix C](#), including a discussion of Riemann–Stieltjes and stochastic integration, a proof of the fact that the spectra of autoregressive processes are dense in the space of spectral densities, and a proof of the fact that spectra are approximately the eigenvalues of the covariance matrix of a stationary process.

We tweaked, rewrote, improved, or revised some of the exercises, but the overall ordering and coverage is roughly the same. And, of course, we moved regression with ARMA errors problems to [Chapter 3](#) and removed the [Chapter 4](#) wavelet problems. The exercises for [Chapter 6](#) have been updated accordingly to reflect the new and improved version of the chapter.

Davis, CA
Pittsburgh, PA
September 2016

*Robert H. Shumway
David S. Stoffer*

Preface to the Third Edition

The goals of this book are to develop an appreciation for the richness and versatility of modern time series analysis as a tool for analyzing data, and still maintain a commitment to theoretical integrity, as exemplified by the seminal works of Brillinger (1975) and Hannan (1970) and the texts by Brockwell and Davis (1991) and Fuller (1995). The advent of inexpensive powerful computing has provided both real data and new software that can take one considerably beyond the fitting of simple time domain models, such as have been elegantly described in the landmark work of Box and Jenkins (1970). This book is designed to be useful as a text for courses in time series on several different levels and as a reference work for practitioners facing the analysis of time-correlated data in the physical, biological, and social sciences.

We have used earlier versions of the text at both the undergraduate and graduate levels over the past decade. Our experience is that an undergraduate course can be accessible to students with a background in regression analysis and may include [Section 1.1–Section 1.5](#), [Section 2.1–Section 2.3](#), the results and numerical parts of [Section 3.1–Section 3.9](#), and briefly the results and numerical parts of [Section 4.1–Section 4.4](#). At the advanced undergraduate or master’s level, where the students have some mathematical statistics background, more detailed coverage of the same sections, with the inclusion of extra topics from [Chapter 5](#) or [Chapter 6](#) can be used as a one-semester course. Often, the extra topics are chosen by the students according to their interests. Finally, a two-semester upper-level graduate course for mathematics, statistics, and engineering graduate students can be crafted by adding selected theoretical appendices. For the upper-level graduate course, we should mention that we are striving for a broader but less rigorous level of coverage than that which is attained by Brockwell and Davis (1991), the classic entry at this level.

The major difference between this third edition of the text and the second edition is that we provide R code for almost all of the numerical examples. An R package called `astsa` is provided for use with the text; see [Section R.2](#) for details. R code is provided simply to enhance the exposition by making the numerical examples reproducible.

We have tried, where possible, to keep the problem sets in order so that an instructor may have an easy time moving from the second edition to the third edition.

However, some of the old problems have been revised and there are some new problems. Also, some of the data sets have been updated. We added one section in [Chapter 5](#) on unit roots and enhanced some of the presentations throughout the text. The exposition on state-space modeling, ARMAX models, and (multivariate) regression with autocorrelated errors in [Chapter 6](#) have been expanded. In this edition, we use standard R functions as much as possible, but we use our own scripts (included in [astsa](#)) when we feel it is necessary to avoid problems with a particular R function; these problems are discussed in detail on the website for the text under R Issues.

We thank John Kimmel, Executive Editor, Springer Statistics, for his guidance in the preparation and production of this edition of the text. We are grateful to Don Percival, University of Washington, for numerous suggestions that led to substantial improvement to the presentation in the second edition, and consequently in this edition. We thank Doug Wiens, University of Alberta, for help with some of the R code in [Chapter 4](#) and [Chapter 7](#), and for his many suggestions for improvement of the exposition. We are grateful for the continued help and advice of Pierre Duchesne, University of Montreal, and Alexander Aue, University of California, Davis. We also thank the many students and other readers who took the time to mention typographical errors and other corrections to the first and second editions. Finally, work on the this edition was supported by the National Science Foundation while one of us (D.S.S.) was working at the Foundation under the Intergovernmental Personnel Act.

Davis, CA
Pittsburgh, PA
September 2010

*Robert H. Shumway
David S. Stoffer*

Contents

Preface to the Fourth Edition	v
Preface to the Third Edition	vii
1 Characteristics of Time Series	1
1.1 The Nature of Time Series Data	2
1.2 Time Series Statistical Models	8
1.3 Measures of Dependence	14
1.4 Stationary Time Series	19
1.5 Estimation of Correlation	26
1.6 Vector-Valued and Multidimensional Series	33
Problems	38
2 Time Series Regression and Exploratory Data Analysis	47
2.1 Classical Regression in the Time Series Context	47
2.2 Exploratory Data Analysis	56
2.3 Smoothing in the Time Series Context	67
Problems	72
3 ARIMA Models	77
3.1 Autoregressive Moving Average Models	77
3.2 Difference Equations	90
3.3 Autocorrelation and Partial Autocorrelation	96
3.4 Forecasting	102
3.5 Estimation	115
3.6 Integrated Models for Nonstationary Data	133
3.7 Building ARIMA Models	137
3.8 Regression with Autocorrelated Errors	145
3.9 Multiplicative Seasonal ARIMA Models	148
Problems	156

4	Spectral Analysis and Filtering	167
4.1	Cyclical Behavior and Periodicity	168
4.2	The Spectral Density	174
4.3	Periodogram and Discrete Fourier Transform	181
4.4	Nonparametric Spectral Estimation	191
4.5	Parametric Spectral Estimation	205
4.6	Multiple Series and Cross-Spectra	208
4.7	Linear Filters	213
4.8	Lagged Regression Models	218
4.9	Signal Extraction and Optimum Filtering	223
4.10	Spectral Analysis of Multidimensional Series	227
	Problems	230
5	Additional Time Domain Topics	241
5.1	Long Memory ARMA and Fractional Differencing	241
5.2	Unit Root Testing	250
5.3	GARCH Models	253
5.4	Threshold Models	261
5.5	Lagged Regression and Transfer Function Modeling	265
5.6	Multivariate ARMAX Models	271
	Problems	284
6	State Space Models	287
6.1	Linear Gaussian Model	288
6.2	Filtering, Smoothing, and Forecasting	292
6.3	Maximum Likelihood Estimation	302
6.4	Missing Data Modifications	310
6.5	Structural Models: Signal Extraction and Forecasting	315
6.6	State-Space Models with Correlated Errors	319
6.6.1	ARMAX Models	320
6.6.2	Multivariate Regression with Autocorrelated Errors	322
6.7	Bootstrapping State Space Models	325
6.8	Smoothing Splines and the Kalman Smoother	331
6.9	Hidden Markov Models and Switching Autoregression	334
6.10	Dynamic Linear Models with Switching	345
6.11	Stochastic Volatility	357
6.12	Bayesian Analysis of State Space Models	365
	Problems	375
7	Statistical Methods in the Frequency Domain	383
7.1	Introduction	383
7.2	Spectral Matrices and Likelihood Functions	386
7.3	Regression for Jointly Stationary Series	388
7.4	Regression with Deterministic Inputs	397
7.5	Random Coefficient Regression	405

7.6 Analysis of Designed Experiments	407
7.7 Discriminant and Cluster Analysis	421
7.8 Principal Components and Factor Analysis	437
7.9 The Spectral Envelope	453
Problems	464
Appendix A Large Sample Theory	471
A.1 Convergence Modes	471
A.2 Central Limit Theorems	478
A.3 The Mean and Autocorrelation Functions	482
Appendix B Time Domain Theory	491
B.1 Hilbert Spaces and the Projection Theorem	491
B.2 Causal Conditions for ARMA Models	495
B.3 Large Sample Distribution of the AR Conditional Least Squares Estimators	497
B.4 The Wold Decomposition	500
Appendix C Spectral Domain Theory	503
C.1 Spectral Representation Theorems	503
C.2 Large Sample Distribution of the Smoothed Periodogram	507
C.3 The Complex Multivariate Normal Distribution	517
C.4 Integration	522
C.4.1 Riemann–Stieltjes Integration	522
C.4.2 Stochastic Integration	524
C.5 Spectral Analysis as Principal Component Analysis	525
C.6 Parametric Spectral Estimation	529
Appendix R R Supplement	531
R.1 First Things First	531
R.2 astsa	531
R.3 Getting Started	532
R.4 Time Series Primer	536
R.4.1 Graphics	539
References	541
Index	553

Chapter 1

Characteristics of Time Series

The analysis of experimental data that have been observed at different points in time leads to new and unique problems in statistical modeling and inference. The obvious correlation introduced by the sampling of adjacent points in time can severely restrict the applicability of the many conventional statistical methods traditionally dependent on the assumption that these adjacent observations are independent and identically distributed. The systematic approach by which one goes about answering the mathematical and statistical questions posed by these time correlations is commonly referred to as time series analysis.

The impact of time series analysis on scientific applications can be partially documented by producing an abbreviated listing of the diverse fields in which important time series problems may arise. For example, many familiar time series occur in the field of economics, where we are continually exposed to daily stock market quotations or monthly unemployment figures. Social scientists follow population series, such as birthrates or school enrollments. An epidemiologist might be interested in the number of influenza cases observed over some time period. In medicine, blood pressure measurements traced over time could be useful for evaluating drugs used in treating hypertension. Functional magnetic resonance imaging of brain-wave time series patterns might be used to study how the brain reacts to certain stimuli under various experimental conditions.

In our view, the first step in any time series investigation always involves careful examination of the recorded data plotted over time. This scrutiny often suggests the method of analysis as well as statistics that will be of use in summarizing the information in the data. Before looking more closely at the particular statistical methods, it is appropriate to mention that two separate, but not necessarily mutually exclusive, approaches to time series analysis exist, commonly identified as the *time domain approach* and the *frequency domain approach*. The time domain approach views the investigation of lagged relationships as most important (e.g., how does what happened today affect what will happen tomorrow), whereas the frequency domain approach views the investigation of cycles as most important (e.g., what is the economic cycle through periods of expansion and recession). We will explore both types of approaches in the following sections.

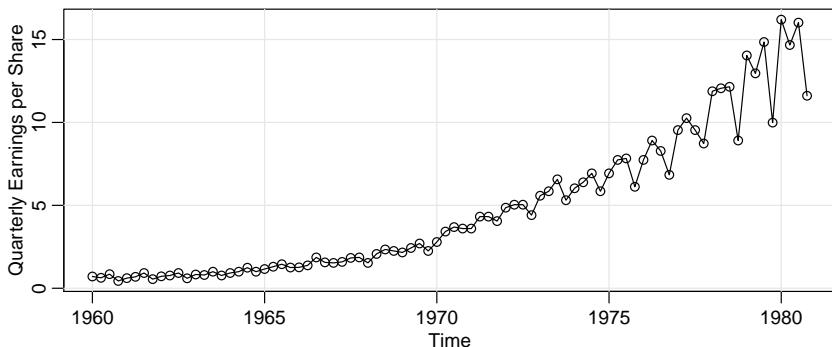


Fig. 1.1. Johnson & Johnson quarterly earnings per share, 84 quarters, 1960-I to 1980-IV.

1.1 The Nature of Time Series Data

Some of the problems and questions of interest to the prospective time series analyst can best be exposed by considering real experimental data taken from different subject areas. The following cases illustrate some of the common kinds of experimental time series data as well as some of the statistical questions that might be asked about such data.

Example 1.1 Johnson & Johnson Quarterly Earnings

Figure 1.1 shows quarterly earnings per share for the U.S. company Johnson & Johnson, furnished by Professor Paul Griffin (personal communication) of the Graduate School of Management, University of California, Davis. There are 84 quarters (21 years) measured from the first quarter of 1960 to the last quarter of 1980. Modeling such series begins by observing the primary patterns in the time history. In this case, note the gradually increasing underlying trend and the rather regular variation superimposed on the trend that seems to repeat over quarters. Methods for analyzing data such as these are explored in Chapter 2 and Chapter 6. To plot the data using the R statistical package, type the following:^{1.1}

```
library(astsa)      # SEE THE FOOTNOTE
plot(jj, type="o", ylab="Quarterly Earnings per Share")
```

Example 1.2 Global Warming

Consider the global temperature series record shown in Figure 1.2. The data are the global mean land–ocean temperature index from 1880 to 2015, with the base period 1951–1980. In particular, the data are deviations, measured in degrees centigrade, from the 1951–1980 average, and are an update of Hansen et al. (2006). We note an apparent upward trend in the series during the latter part of the twentieth century that has been used as an argument for the global warming hypothesis. Note also the leveling off at about 1935 and then another rather sharp upward trend at about

^{1.1} Throughout the text, we assume that the R package for the book, `astsa`, has been installed and loaded. See Section R.2 for further details.

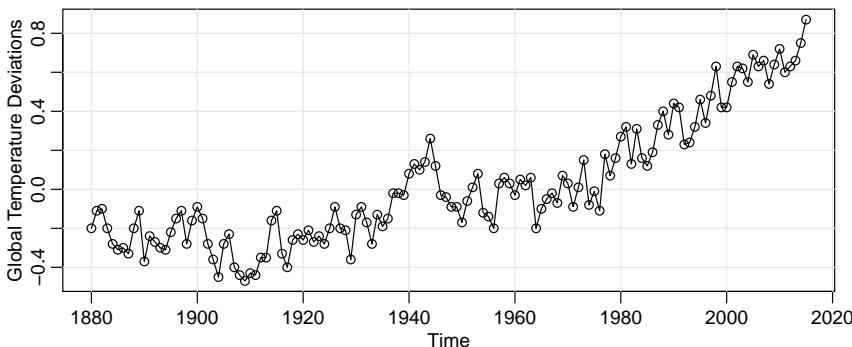


Fig. 1.2. Yearly average global temperature deviations (1880–2015) in degrees centigrade.

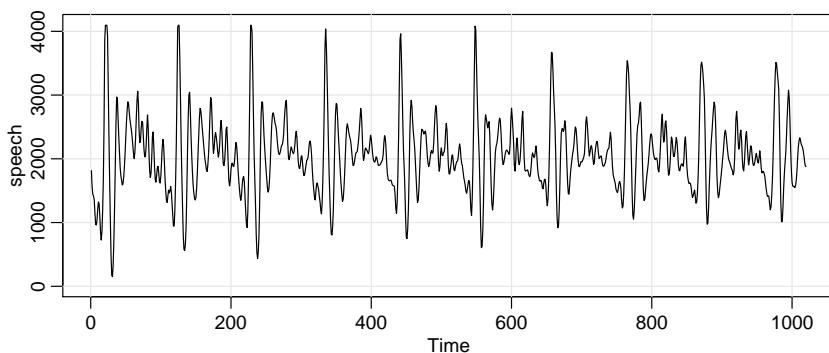


Fig. 1.3. Speech recording of the syllable *aaa ··· hhh* sampled at 10,000 points per second with $n = 1020$ points.

1970. The question of interest for global warming proponents and opponents is whether the overall trend is natural or whether it is caused by some human-induced interface. **Problem 2.8** examines 634 years of glacial sediment data that might be taken as a long-term temperature proxy. Such percentage changes in temperature do not seem to be unusual over a time period of 100 years. Again, the question of trend is of more interest than particular periodicities. The R code for this example is similar to the code in [Example 1.1](#):

```
plot(globtemp, type="o", ylab="Global Temperature Deviations")
```

Example 1.3 Speech Data

[Figure 1.3](#) shows a small .1 second (1000 point) sample of recorded speech for the phrase *aaa ··· hhh*, and we note the repetitive nature of the signal and the rather regular periodicities. One current problem of great interest is computer recognition of speech, which would require converting this particular signal into the recorded phrase *aaa ··· hhh*. Spectral analysis can be used in this context to produce a signature of this phrase that can be compared with signatures of various

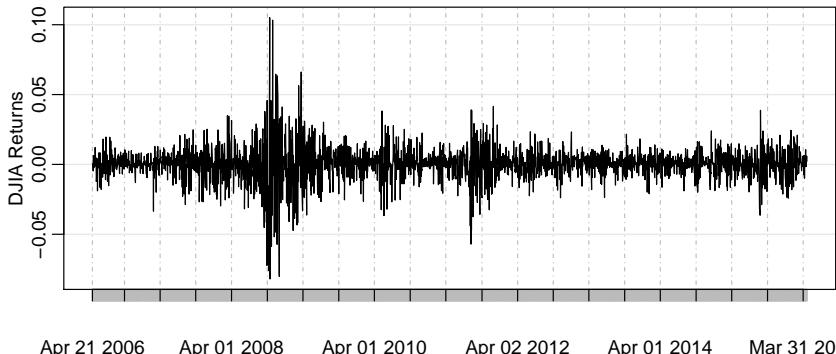


Fig. 1.4. The daily returns of the Dow Jones Industrial Average (DJIA) from April 20, 2006 to April 20, 2016.

library syllables to look for a match. One can immediately notice the rather regular repetition of small wavelets. The separation between the packets is known as the *pitch period* and represents the response of the vocal tract filter to a periodic sequence of pulses stimulated by the opening and closing of the glottis. In R, you can reproduce Figure 1.3 using `plot(speech)`.

Example 1.4 Dow Jones Industrial Average

As an example of financial time series data, Figure 1.4 shows the daily *returns* (or percent change) of the Dow Jones Industrial Average (DJIA) from April 20, 2006, to April 20, 2016. It is easy to spot the financial crisis of 2008 in the figure. The data shown in Figure 1.4 are typical of return data. The mean of the series appears to be stable with an average return of approximately zero, however, highly volatile (variable) periods tend to be clustered together. A problem in the analysis of these type of financial data is to forecast the volatility of future returns. Models such as *ARCH* and *GARCH* models (Engle, 1982; Bollerslev, 1986) and *stochastic volatility* models (Harvey, Ruiz and Shephard, 1994) have been developed to handle these problems. We will discuss these models and the analysis of financial data in Chapter 5 and Chapter 6. The data were obtained using the Technical Trading Rules (TTR) package to download the data from YahooTM and then plot it. We then used the fact that if x_t is the actual value of the DJIA and $r_t = (x_t - x_{t-1})/x_{t-1}$ is the return, then $1 + r_t = x_t/x_{t-1}$ and $\log(1 + r_t) = \log(x_t/x_{t-1}) = \log(x_t) - \log(x_{t-1}) \approx r_t$.^{1.2} The data set is also available in `astsa`, but `xts` must be loaded.

```
# library(TTR)
# djia = getYahooData("^DJI", start=20060420, end=20160420, freq="daily")
library(xts)
djiar = diff(log(djia$Close))[-1] # approximate returns
plot(djiar, main="DJIA Returns", type="n")
lines(djiar)
```

^{1.2} $\log(1 + p) = p - \frac{p^2}{2} + \frac{p^3}{3} - \dots$ for $-1 < p \leq 1$. If p is near zero, the higher-order terms in the expansion are negligible.

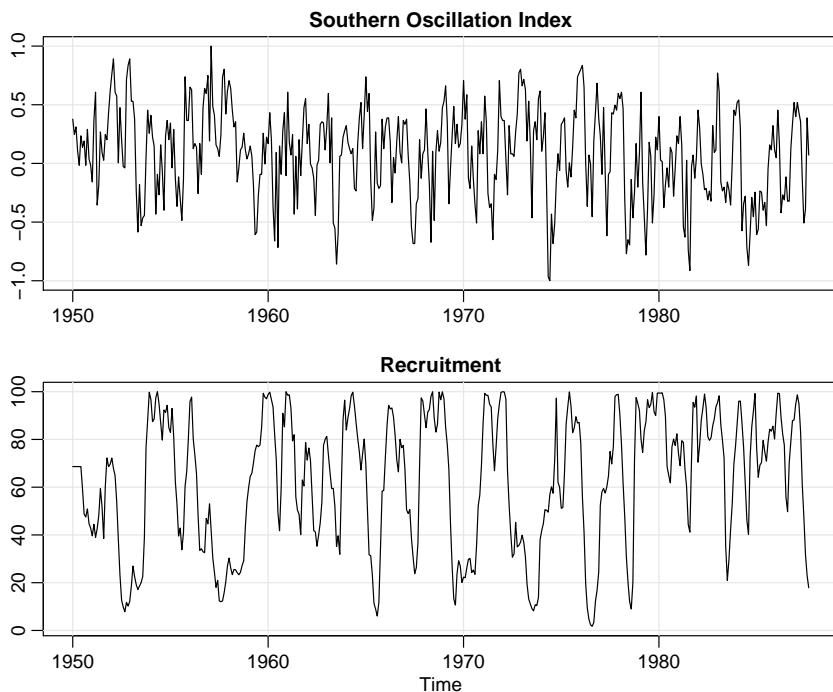


Fig. 1.5. Monthly SOI and Recruitment (estimated new fish), 1950–1987.

Example 1.5 El Niño and Fish Population

We may also be interested in analyzing several time series at once. Figure 1.5 shows monthly values of an environmental series called the *Southern Oscillation Index* (SOI) and associated Recruitment (number of new fish) furnished by Dr. Roy Mendelsohn of the Pacific Environmental Fisheries Group (personal communication). Both series are for a period of 453 months ranging over the years 1950–1987. The SOI measures changes in air pressure, related to sea surface temperatures in the central Pacific Ocean. The central Pacific warms every three to seven years due to the El Niño effect, which has been blamed for various global extreme weather events. Both series in Figure 1.5 exhibit repetitive behavior, with regularly repeating cycles that are easily visible. This periodic behavior is of interest because underlying processes of interest may be regular and the rate or *frequency* of oscillation characterizing the behavior of the underlying series would help to identify them. The series show two basic oscillations types, an obvious annual cycle (hot in the summer, cold in the winter), and a slower frequency that seems to repeat about every 4 years. The study of the kinds of cycles and their strengths is the subject of Chapter 4. The two series are also related; it is easy to imagine the fish population is dependent on the ocean temperature. This possibility suggests trying some version of regression analysis as a procedure for relating the two series. *Transfer function*

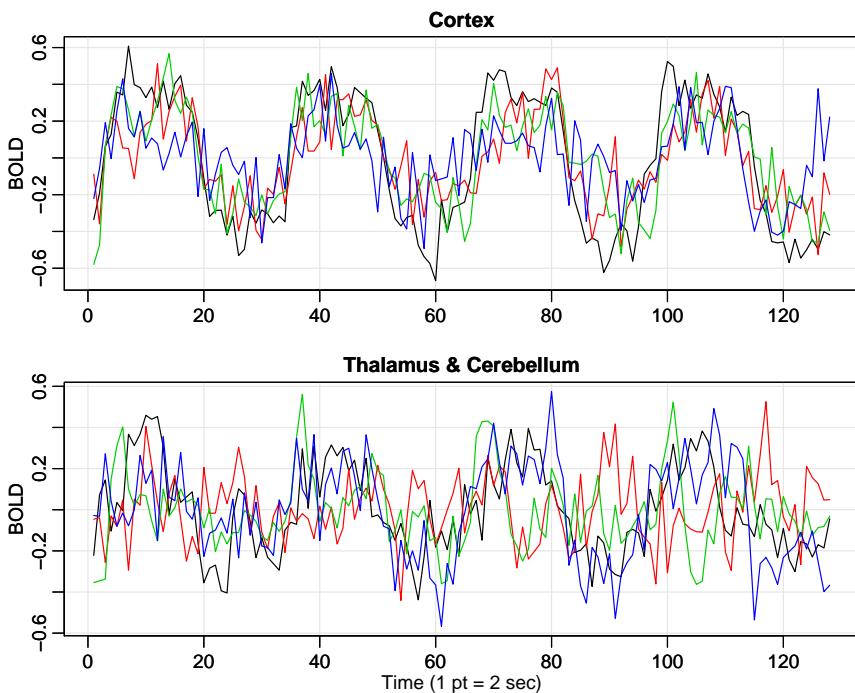


Fig. 1.6. fMRI data from various locations in the cortex, thalamus, and cerebellum; $n = 128$ points, one observation taken every 2 seconds.

modeling, as considered in [Chapter 5](#), can also be applied in this case. The following R code will reproduce [Figure 1.5](#):

```
par(mfrow = c(2,1)) # set up the graphics
plot(soi, ylab="", xlab="", main="Southern Oscillation Index")
plot(rec, ylab="", xlab="", main="Recruitment")
```

Example 1.6 fMRI Imaging

A fundamental problem in classical statistics occurs when we are given a collection of independent series or vectors of series, generated under varying experimental conditions or treatment configurations. Such a set of series is shown in [Figure 1.6](#), where we observe data collected from various locations in the brain via functional magnetic resonance imaging (fMRI). In this example, five subjects were given periodic brushing on the hand. The stimulus was applied for 32 seconds and then stopped for 32 seconds; thus, the signal period is 64 seconds. The sampling rate was one observation every 2 seconds for 256 seconds ($n = 128$). For this example, we averaged the results over subjects (these were evoked responses, and all subjects were in phase). The series shown in [Figure 1.6](#) are consecutive measures of blood oxygenation-level dependent (BOLD) signal intensity, which measures areas of activation in the brain. Notice that the periodicities appear strongly in the motor cortex

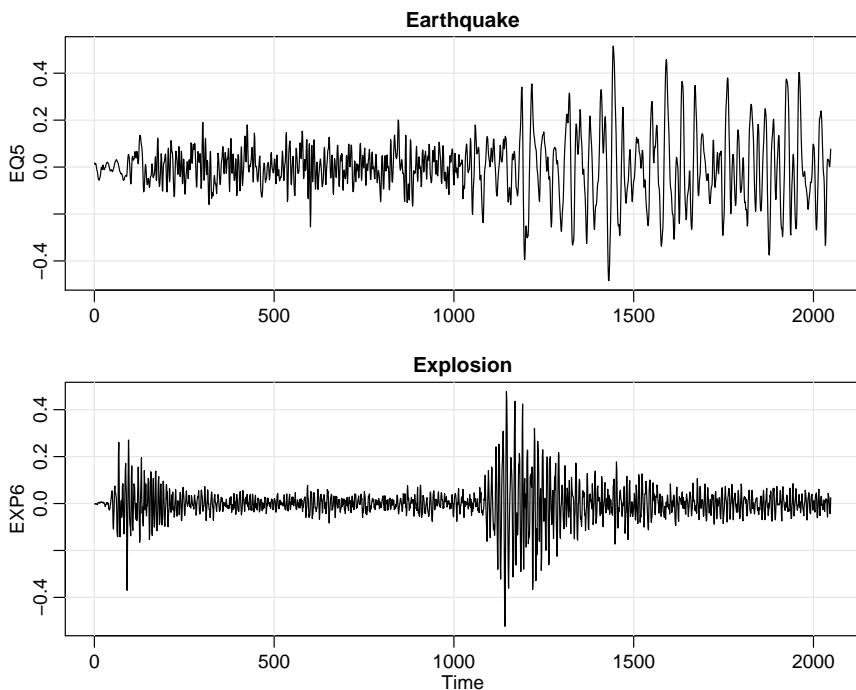


Fig. 1.7. Arrival phases from an earthquake (top) and explosion (bottom) at 40 points per second.

series and less strongly in the thalamus and cerebellum. The fact that one has series from different areas of the brain suggests testing whether the areas are responding differently to the brush stimulus. Analysis of variance techniques accomplish this in classical statistics, and we show in Chapter 7 how these classical techniques extend to the time series case, leading to a spectral analysis of variance. The following R commands can be used to plot the data:

```
par(mfrow=c(2,1))
ts.plot(fmri1[,2:5], col=1:4, ylab="BOLD", main="Cortex")
ts.plot(fmri1[,6:9], col=1:4, ylab="BOLD", main="Thalamus & Cerebellum")
```

Example 1.7 Earthquakes and Explosions

As a final example, the series in Figure 1.7 represent two phases or arrivals along the surface, denoted by P ($t = 1, \dots, 1024$) and S ($t = 1025, \dots, 2048$), at a seismic recording station. The recording instruments in Scandinavia are observing earthquakes and mining explosions with one of each shown in Figure 1.7. The general problem of interest is in distinguishing or discriminating between waveforms generated by earthquakes and those generated by explosions. Features that may be important are the rough amplitude ratios of the first phase P to the second phase S, which tend to be smaller for earthquakes than for explosions. In the case of the

two events in [Figure 1.7](#), the ratio of maximum amplitudes appears to be somewhat less than .5 for the earthquake and about 1 for the explosion. Otherwise, note a subtle difference exists in the periodic nature of the S phase for the earthquake. We can again think about spectral analysis of variance for testing the equality of the periodic components of earthquakes and explosions. We would also like to be able to classify future P and S components from events of unknown origin, leading to the *time series discriminant analysis* developed in [Chapter 7](#).

To plot the data as in this example, use the following commands in R:

```
par(mfrow=c(2,1))
plot(EQ5, main="Earthquake")
plot(EXP6, main="Explosion")
```

1.2 Time Series Statistical Models

The primary objective of time series analysis is to develop mathematical models that provide plausible descriptions for sample data, like that encountered in the previous section. In order to provide a statistical setting for describing the character of data that seemingly fluctuate in a random fashion over time, we assume a *time series* can be defined as a collection of random variables indexed according to the order they are obtained in time. For example, we may consider a time series as a sequence of random variables, x_1, x_2, x_3, \dots , where the random variable x_1 denotes the value taken by the series at the first time point, the variable x_2 denotes the value for the second time period, x_3 denotes the value for the third time period, and so on. In general, a collection of random variables, $\{x_t\}$, indexed by t is referred to as a *stochastic process*. In this text, t will typically be discrete and vary over the integers $t = 0, \pm 1, \pm 2, \dots$, or some subset of the integers. The observed values of a stochastic process are referred to as a *realization* of the stochastic process. Because it will be clear from the context of our discussions, we use the term *time series* whether we are referring generically to the process or to a particular realization and make no notational distinction between the two concepts.

It is conventional to display a sample time series graphically by plotting the values of the random variables on the vertical axis, or ordinate, with the time scale as the abscissa. It is usually convenient to connect the values at adjacent time periods to reconstruct visually some original hypothetical continuous time series that might have produced these values as a discrete sample. Many of the series discussed in the previous section, for example, could have been observed at any continuous point in time and are conceptually more properly treated as *continuous time series*. The approximation of these series by *discrete time parameter series* sampled at equally spaced points in time is simply an acknowledgment that sampled data will, for the most part, be discrete because of restrictions inherent in the method of collection. Furthermore, the analysis techniques are then feasible using computers, which are limited to digital computations. Theoretical developments also rest on the idea that a continuous parameter time series should be specified in terms of finite-dimensional *distribution functions* defined over a finite number of points in time. This is not to

say that the selection of the sampling interval or rate is not an extremely important consideration. The appearance of data can be changed completely by adopting an insufficient sampling rate. We have all seen wheels in movies appear to be turning backwards because of the insufficient number of frames sampled by the camera. This phenomenon leads to a distortion called *aliasing* (see [Section 4.1](#)).

The fundamental visual characteristic distinguishing the different series shown in [Example 1.1](#)–[Example 1.7](#) is their differing degrees of smoothness. One possible explanation for this smoothness is that it is being induced by the supposition that adjacent points in time are *correlated*, so the value of the series at time t , say, x_t , depends in some way on the past values x_{t-1}, x_{t-2}, \dots . This model expresses a fundamental way in which we might think about generating realistic-looking time series. To begin to develop an approach to using collections of random variables to model time series, consider [Example 1.8](#).

Example 1.8 White Noise (3 flavors)

A simple kind of generated series might be a collection of uncorrelated random variables, w_t , with mean 0 and finite variance σ_w^2 . The time series generated from uncorrelated variables is used as a model for noise in engineering applications, where it is called *white noise*; we shall denote this process as $w_t \sim \text{wn}(0, \sigma_w^2)$. The designation white originates from the analogy with white light and indicates that all possible periodic oscillations are present with equal strength.

We will sometimes require the noise to be independent and identically distributed (iid) random variables with mean 0 and variance σ_w^2 . We distinguish this by writing $w_t \sim \text{iid}(0, \sigma_w^2)$ or by saying *white independent noise* or *iid noise*. A particularly useful white noise series is *Gaussian white noise*, wherein the w_t are independent normal random variables, with mean 0 and variance σ_w^2 ; or more succinctly, $w_t \sim \text{iid N}(0, \sigma_w^2)$. [Figure 1.8](#) shows in the upper panel a collection of 500 such random variables, with $\sigma_w^2 = 1$, plotted in the order in which they were drawn. The resulting series bears a slight resemblance to the explosion in [Figure 1.7](#) but is not smooth enough to serve as a plausible model for any of the other experimental series. The plot tends to show visually a mixture of many different kinds of oscillations in the white noise series.

If the stochastic behavior of all time series could be explained in terms of the white noise model, classical statistical methods would suffice. Two ways of introducing serial correlation and more smoothness into time series models are given in [Example 1.9](#) and [Example 1.10](#).

Example 1.9 Moving Averages and Filtering

We might replace the white noise series w_t by a *moving average* that smooths the series. For example, consider replacing w_t in [Example 1.8](#) by an average of its current value and its immediate neighbors in the past and future. That is, let

$$v_t = \frac{1}{3}(w_{t-1} + w_t + w_{t+1}), \quad (1.1)$$

which leads to the series shown in the lower panel of [Figure 1.8](#). Inspecting the series shows a smoother version of the first series, reflecting the fact that the slower

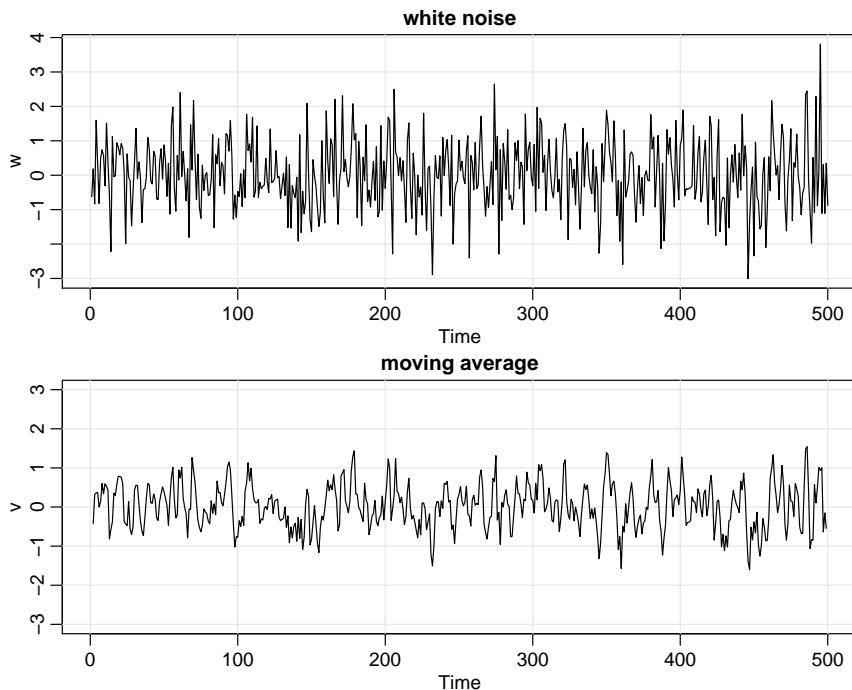


Fig. 1.8. Gaussian white noise series (top) and three-point moving average of the Gaussian white noise series (bottom).

oscillations are more apparent and some of the faster oscillations are taken out. We begin to notice a similarity to the SOI in Figure 1.5, or perhaps, to some of the fMRI series in Figure 1.6.

A linear combination of values in a time series such as in (1.1) is referred to, generically, as a filtered series; hence the command `filter` in the following code for Figure 1.8.

```
w = rnorm(500,0,1)          # 500 N(0,1) variates
v = filter(w, sides=2, filter=rep(1/3,3)) # moving average
par(mfrow=c(2,1))
plot.ts(w, main="white noise")
plot.ts(v, ylim=c(-3,3), main="moving average")
```

The speech series in Figure 1.3 and the Recruitment series in Figure 1.5, as well as some of the MRI series in Figure 1.6, differ from the moving average series because one particular kind of oscillatory behavior seems to predominate, producing a sinusoidal type of behavior. A number of methods exist for generating series with this quasi-periodic behavior; we illustrate a popular one based on the autoregressive model considered in Chapter 3.

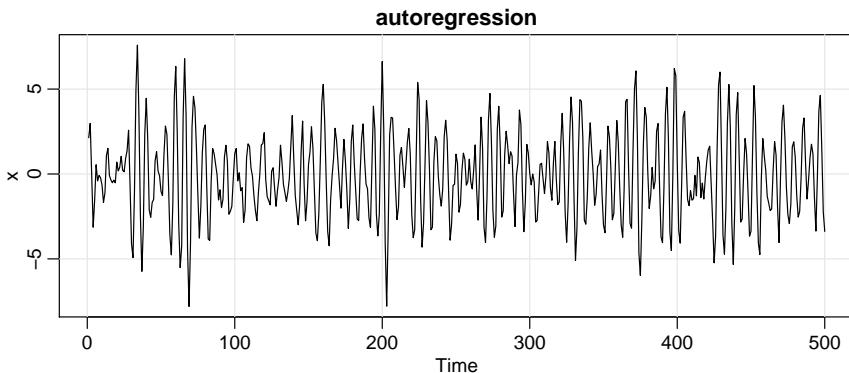


Fig. 1.9. Autoregressive series generated from model (1.2).

Example 1.10 Autoregressions

Suppose we consider the white noise series w_t of Example 1.8 as input and calculate the output using the second-order equation

$$x_t = x_{t-1} - .9x_{t-2} + w_t \quad (1.2)$$

successively for $t = 1, 2, \dots, 500$. Equation (1.2) represents a regression or prediction of the current value x_t of a time series as a function of the past two values of the series, and, hence, the term *autoregression* is suggested for this model. A problem with startup values exists here because (1.2) also depends on the initial conditions x_0 and x_{-1} , but assuming we have the values, we generate the succeeding values by substituting into (1.2). The resulting output series is shown in Figure 1.9, and we note the periodic behavior of the series, which is similar to that displayed by the speech series in Figure 1.3. The autoregressive model above and its generalizations can be used as an underlying model for many observed series and will be studied in detail in Chapter 3.

As in the previous example, the data are obtained by a filter of white noise. The function `filter` uses zeros for the initial values. In this case, $x_1 = w_1$, and $x_2 = x_1 + w_2 = w_1 + w_2$, and so on, so that the values do not satisfy (1.2). An easy fix is to run the filter for longer than needed and remove the initial values.

```
w = rnorm(550, 0, 1) # 50 extra to avoid startup problems
x = filter(w, filter=c(1, -.9), method="recursive")[-(1:50)] # remove first 50
plot.ts(x, main="autoregression")
```

Example 1.11 Random Walk with Drift

A model for analyzing trend such as seen in the global temperature data in Figure 1.2, is the *random walk with drift* model given by

$$x_t = \delta + x_{t-1} + w_t \quad (1.3)$$

for $t = 1, 2, \dots$, with initial condition $x_0 = 0$, and where w_t is white noise. The constant δ is called the *drift*, and when $\delta = 0$, (1.3) is called simply a *random walk*.

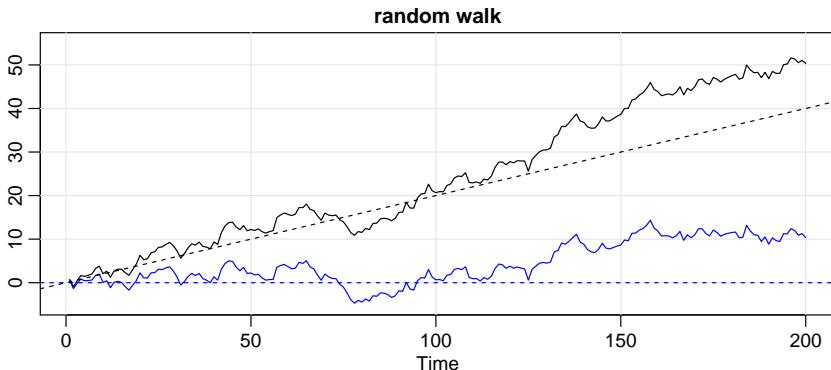


Fig. 1.10. Random walk, $\sigma_w = 1$, with drift $\delta = .2$ (upper jagged line), without drift, $\delta = 0$ (lower jagged line), and straight (dashed) lines with slope δ .

The term random walk comes from the fact that, when $\delta = 0$, the value of the time series at time t is the value of the series at time $t - 1$ plus a completely random movement determined by w_t . Note that we may rewrite (1.3) as a cumulative sum of white noise variates. That is,

$$x_t = \delta t + \sum_{j=1}^t w_j \quad (1.4)$$

for $t = 1, 2, \dots$; either use induction, or plug (1.4) into (1.3) to verify this statement. Figure 1.10 shows 200 observations generated from the model with $\delta = 0$ and $.2$, and with $\sigma_w = 1$. For comparison, we also superimposed the straight line $.2t$ on the graph. To reproduce Figure 1.10 in R use the following code (notice the use of multiple commands per line using a semicolon).

```
set.seed(154)      # so you can reproduce the results
w = rnorm(200);  x = cumsum(w)  # two commands in one line
wd = w + .2;    xd = cumsum(wd)
plot.ts(xd, ylim=c(-5,55), main="random walk", ylab='')
lines(x, col=4); abline(h=0, col=4, lty=2); abline(a=0, b=.2, lty=2)
```

Example 1.12 Signal in Noise

Many realistic models for generating time series assume an underlying signal with some consistent periodic variation, contaminated by adding a random noise. For example, it is easy to detect the regular cycle fMRI series displayed on the top of Figure 1.6. Consider the model

$$x_t = 2 \cos(2\pi \frac{t+15}{50}) + w_t \quad (1.5)$$

for $t = 1, 2, \dots, 500$, where the first term is regarded as the signal, shown in the upper panel of Figure 1.11. We note that a sinusoidal waveform can be written as

$$A \cos(2\pi\omega t + \phi), \quad (1.6)$$

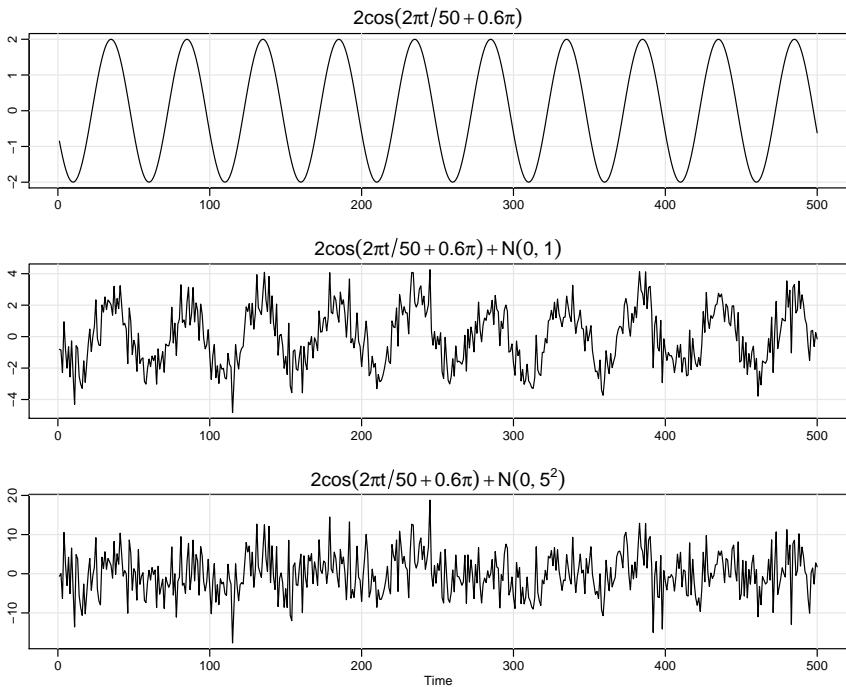


Fig. 1.11. Cosine wave with period 50 points (top panel) compared with the cosine wave contaminated with additive white Gaussian noise, $\sigma_w = 1$ (middle panel) and $\sigma_w = 5$ (bottom panel); see (1.5).

where A is the amplitude, ω is the frequency of oscillation, and ϕ is a phase shift. In (1.5), $A = 2$, $\omega = 1/50$ (one cycle every 50 time points), and $\phi = 2\pi 15/50 = .6\pi$.

An additive noise term was taken to be white noise with $\sigma_w = 1$ (middle panel) and $\sigma_w = 5$ (bottom panel), drawn from a normal distribution. Adding the two together obscures the signal, as shown in the lower panels of Figure 1.11. Of course, the degree to which the signal is obscured depends on the amplitude of the signal and the size of σ_w . The ratio of the amplitude of the signal to σ_w (or some function of the ratio) is sometimes called the *signal-to-noise ratio (SNR)*; the larger the SNR, the easier it is to detect the signal. Note that the signal is easily discernible in the middle panel of Figure 1.11, whereas the signal is obscured in the bottom panel. Typically, we will not observe the signal but the signal obscured by noise.

To reproduce Figure 1.11 in R, use the following commands:

```
cs = 2*cos(2*pi*1:500/50 + .6*pi); w = rnorm(500,0,1)
par(mfrow=c(3,1), mar=c(3,2,2,1), cex.main=1.5)
plot.ts(cs, main=expression(2*cos(2*pi*t/50+.6*pi)))
plot.ts(cs+w, main=expression(2*cos(2*pi*t/50+.6*pi) + N(0,1)))
plot.ts(cs+5*w, main=expression(2*cos(2*pi*t/50+.6*pi) + N(0,25)))
```

In [Chapter 4](#), we will study the use of *spectral analysis* as a possible technique for detecting regular or periodic signals, such as the one described in [Example 1.12](#). In general, we would emphasize the importance of simple additive models such as given above in the form

$$x_t = s_t + v_t, \quad (1.7)$$

where s_t denotes some unknown signal and v_t denotes a time series that may be white or correlated over time. The problems of detecting a signal and then in estimating or extracting the waveform of s_t are of great interest in many areas of engineering and the physical and biological sciences. In economics, the underlying signal may be a trend or it may be a seasonal component of a series. Models such as (1.7), where the signal has an autoregressive structure, form the motivation for the state-space model of [Chapter 6](#).

In the above examples, we have tried to motivate the use of various combinations of random variables emulating real time series data. Smoothness characteristics of observed time series were introduced by combining the random variables in various ways. Averaging independent random variables over adjacent time points, as in [Example 1.9](#), or looking at the output of difference equations that respond to white noise inputs, as in [Example 1.10](#), are common ways of generating correlated data. In the next section, we introduce various theoretical measures used for describing how time series behave. As is usual in statistics, the complete description involves the multivariate distribution function of the jointly sampled values x_1, x_2, \dots, x_n , whereas more economical descriptions can be had in terms of the mean and autocorrelation functions. Because correlation is an essential feature of time series analysis, the most useful descriptive measures are those expressed in terms of covariance and correlation functions.

1.3 Measures of Dependence

A complete description of a time series, observed as a collection of n random variables at arbitrary time points t_1, t_2, \dots, t_n , for any positive integer n , is provided by the joint distribution function, evaluated as the probability that the values of the series are jointly less than the n constants, c_1, c_2, \dots, c_n ; i.e.,

$$F_{t_1, t_2, \dots, t_n}(c_1, c_2, \dots, c_n) = \Pr(x_{t_1} \leq c_1, x_{t_2} \leq c_2, \dots, x_{t_n} \leq c_n). \quad (1.8)$$

Unfortunately, these multidimensional distribution functions cannot usually be written easily unless the random variables are jointly normal, in which case the joint density has the well-known form displayed in (1.33).

Although the joint distribution function describes the data completely, it is an unwieldy tool for displaying and analyzing time series data. The distribution function (1.8) must be evaluated as a function of n arguments, so any plotting of the corresponding multivariate density functions is virtually impossible. The marginal distribution functions

$$F_t(x) = P\{x_t \leq x\}$$

or the corresponding marginal density functions

$$f_t(x) = \frac{\partial F_t(x)}{\partial x},$$

when they exist, are often informative for examining the marginal behavior of a series.^{1.3} Another informative marginal descriptive measure is the mean function.

Definition 1.11 *The mean function is defined as*

$$\mu_{xt} = E(x_t) = \int_{-\infty}^{\infty} x f_t(x) dx, \quad (1.9)$$

provided it exists, where E denotes the usual expected value operator. When no confusion exists about which time series we are referring to, we will drop a subscript and write μ_{xt} as μ_t .

Example 1.13 Mean Function of a Moving Average Series

If w_t denotes a white noise series, then $\mu_{wt} = E(w_t) = 0$ for all t . The top series in Figure 1.8 reflects this, as the series clearly fluctuates around a mean value of zero. Smoothing the series as in Example 1.9 does not change the mean because we can write

$$\mu_{vt} = E(v_t) = \frac{1}{3}[E(w_{t-1}) + E(w_t) + E(w_{t+1})] = 0.$$

Example 1.14 Mean Function of a Random Walk with Drift

Consider the random walk with drift model given in (1.4),

$$x_t = \delta t + \sum_{j=1}^t w_j, \quad t = 1, 2, \dots .$$

Because $E(w_t) = 0$ for all t , and δ is a constant, we have

$$\mu_{xt} = E(x_t) = \delta t + \sum_{j=1}^t E(w_j) = \delta t$$

which is a straight line with slope δ . A realization of a random walk with drift can be compared to its mean function in Figure 1.10.

^{1.3} If x_t is Gaussian with mean μ_t and variance σ_t^2 , abbreviated as $x_t \sim N(\mu_t, \sigma_t^2)$, the marginal density is given by $f_t(x) = \frac{1}{\sigma_t \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma_t^2}(x - \mu_t)^2 \right\}$, $x \in \mathbb{R}$.

Example 1.15 Mean Function of Signal Plus Noise

A great many practical applications depend on assuming the observed data have been generated by a fixed signal waveform superimposed on a zero-mean noise process, leading to an additive signal model of the form (1.5). It is clear, because the signal in (1.5) is a fixed function of time, we will have

$$\begin{aligned}\mu_{xt} &= E(x_t) = E\left[2 \cos(2\pi \frac{t+15}{50}) + w_t\right] \\ &= 2 \cos(2\pi \frac{t+15}{50}) + E(w_t) \\ &= 2 \cos(2\pi \frac{t+15}{50}),\end{aligned}$$

and the mean function is just the cosine wave.

The lack of independence between two adjacent values x_s and x_t can be assessed numerically, as in classical statistics, using the notions of covariance and correlation. Assuming the variance of x_t is finite, we have the following definition.

Definition 1.2 *The autocovariance function is defined as the second moment product*

$$\gamma_x(s, t) = \text{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)], \quad (1.10)$$

for all s and t . When no possible confusion exists about which time series we are referring to, we will drop the subscript and write $\gamma_x(s, t)$ as $\gamma(s, t)$. Note that $\gamma_x(s, t) = \gamma_x(t, s)$ for all time points s and t .

The autocovariance measures the *linear* dependence between two points on the same series observed at different times. Very smooth series exhibit autocovariance functions that stay large even when the t and s are far apart, whereas choppy series tend to have autocovariance functions that are nearly zero for large separations. Recall from classical statistics that if $\gamma_x(s, t) = 0$, x_s and x_t are not linearly related, but there still may be some dependence structure between them. If, however, x_s and x_t are bivariate normal, $\gamma_x(s, t) = 0$ ensures their independence. It is clear that, for $s = t$, the autocovariance reduces to the (assumed finite) *variance*, because

$$\gamma_x(t, t) = E[(x_t - \mu_t)^2] = \text{var}(x_t). \quad (1.11)$$

Example 1.16 Autocovariance of White Noise

The white noise series w_t has $E(w_t) = 0$ and

$$\gamma_w(s, t) = \text{cov}(w_s, w_t) = \begin{cases} \sigma_w^2 & s = t, \\ 0 & s \neq t. \end{cases} \quad (1.12)$$

A realization of white noise with $\sigma_w^2 = 1$ is shown in the top panel of Figure 1.8.

We often have to calculate the autocovariance between filtered series. A useful result is given in the following proposition.

Property 1.1 Covariance of Linear Combinations

If the random variables

$$U = \sum_{j=1}^m a_j X_j \quad \text{and} \quad V = \sum_{k=1}^r b_k Y_k$$

are linear combinations of (finite variance) random variables $\{X_j\}$ and $\{Y_k\}$, respectively, then

$$\text{cov}(U, V) = \sum_{j=1}^m \sum_{k=1}^r a_j b_k \text{cov}(X_j, Y_k). \quad (1.13)$$

Furthermore, $\text{var}(U) = \text{cov}(U, U)$.

Example 1.17 Autocovariance of a Moving Average

Consider applying a three-point moving average to the white noise series w_t of the previous example as in [Example 1.9](#). In this case,

$$\gamma_v(s, t) = \text{cov}(v_s, v_t) = \text{cov}\left\{\frac{1}{3}(w_{s-1} + w_s + w_{s+1}), \frac{1}{3}(w_{t-1} + w_t + w_{t+1})\right\}.$$

When $s = t$ we have

$$\begin{aligned} \gamma_v(t, t) &= \frac{1}{9} \text{cov}\{(w_{t-1} + w_t + w_{t+1}), (w_{t-1} + w_t + w_{t+1})\} \\ &= \frac{1}{9} [\text{cov}(w_{t-1}, w_{t-1}) + \text{cov}(w_t, w_t) + \text{cov}(w_{t+1}, w_{t+1})] \\ &= \frac{3}{9} \sigma_w^2. \end{aligned}$$

When $s = t + 1$,

$$\begin{aligned} \gamma_v(t+1, t) &= \frac{1}{9} \text{cov}\{(w_t + w_{t+1} + w_{t+2}), (w_{t-1} + w_t + w_{t+1})\} \\ &= \frac{1}{9} [\text{cov}(w_t, w_t) + \text{cov}(w_{t+1}, w_{t+1})] \\ &= \frac{2}{9} \sigma_w^2, \end{aligned}$$

using [\(1.12\)](#). Similar computations give $\gamma_v(t-1, t) = 2\sigma_w^2/9$, $\gamma_v(t+2, t) = \gamma_v(t-2, t) = \sigma_w^2/9$, and 0 when $|t - s| > 2$. We summarize the values for all s and t as

$$\gamma_v(s, t) = \begin{cases} \frac{3}{9} \sigma_w^2 & s = t, \\ \frac{2}{9} \sigma_w^2 & |s - t| = 1, \\ \frac{1}{9} \sigma_w^2 & |s - t| = 2, \\ 0 & |s - t| > 2. \end{cases} \quad (1.14)$$

[Example 1.17](#) shows clearly that the smoothing operation introduces a covariance function that decreases as the separation between the two time points increases and disappears completely when the time points are separated by three or more time points. This particular autocovariance is interesting because it only depends on the time separation or *lag* and not on the absolute location of the points along the series. We shall see later that this dependence suggests a mathematical model for the concept of *weak stationarity*.

Example 1.18 Autocovariance of a Random Walk

For the random walk model, $x_t = \sum_{j=1}^t w_j$, we have

$$\gamma_x(s, t) = \text{cov}(x_s, x_t) = \text{cov}\left(\sum_{j=1}^s w_j, \sum_{k=1}^t w_k\right) = \min\{s, t\} \sigma_w^2,$$

because the w_t are uncorrelated random variables. Note that, as opposed to the previous examples, the autocovariance function of a random walk depends on the particular time values s and t , and not on the time separation or lag. Also, notice that the variance of the random walk, $\text{var}(x_t) = \gamma_x(t, t) = t \sigma_w^2$, increases without bound as time t increases. The effect of this variance increase can be seen in [Figure 1.10](#) where the processes start to move away from their mean functions δt (note that $\delta = 0$ and $.2$ in that example).

As in classical statistics, it is more convenient to deal with a measure of association between -1 and 1 , and this leads to the following definition.

Definition 1.3 *The autocorrelation function (ACF) is defined as*

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}}. \quad (1.15)$$

The ACF measures the linear predictability of the series at time t , say x_t , using only the value x_s . We can show easily that $-1 \leq \rho(s, t) \leq 1$ using the Cauchy–Schwarz inequality.^{1.4} If we can predict x_t perfectly from x_s through a linear relationship, $x_t = \beta_0 + \beta_1 x_s$, then the correlation will be $+1$ when $\beta_1 > 0$, and -1 when $\beta_1 < 0$. Hence, we have a rough measure of the ability to forecast the series at time t from the value at time s .

Often, we would like to measure the predictability of another series y_t from the series x_s . Assuming both series have finite variances, we have the following definition.

Definition 1.4 *The cross-covariance function between two series, x_t and y_t , is*

$$\gamma_{xy}(s, t) = \text{cov}(x_s, y_t) = E[(x_s - \mu_{xs})(y_t - \mu_{yt})]. \quad (1.16)$$

There is also a scaled version of the cross-covariance function.

Definition 1.5 *The cross-correlation function (CCF) is given by*

$$\rho_{xy}(s, t) = \frac{\gamma_{xy}(s, t)}{\sqrt{\gamma_x(s, s)\gamma_y(t, t)}}. \quad (1.17)$$

^{1.4} The Cauchy–Schwarz inequality implies $|\gamma(s, t)|^2 \leq \gamma(s, s)\gamma(t, t)$.

We may easily extend the above ideas to the case of more than two series, say, $x_{t1}, x_{t2}, \dots, x_{tr}$; that is, *multivariate time series* with r components. For example, the extension of (1.10) in this case is

$$\gamma_{jk}(s, t) = E[(x_{sj} - \mu_{sj})(x_{tk} - \mu_{tk})] \quad j, k = 1, 2, \dots, r. \quad (1.18)$$

In the definitions above, the autocovariance and cross-covariance functions may change as one moves along the series because the values depend on both s and t , the locations of the points in time. In [Example 1.17](#), the autocovariance function depends on the separation of x_s and x_t , say, $h = |s - t|$, and not on where the points are located in time. As long as the points are separated by h units, the location of the two points does not matter. This notion, called *weak stationarity*, when the mean is constant, is fundamental in allowing us to analyze sample time series data when only a single series is available.

1.4 Stationary Time Series

The preceding definitions of the mean and autocovariance functions are completely general. Although we have not made any special assumptions about the behavior of the time series, many of the preceding examples have hinted that a sort of regularity may exist over time in the behavior of a time series. We introduce the notion of regularity using a concept called *stationarity*.

Definition 1.6 A *strictly stationary time series* is one for which the probabilistic behavior of every collection of values

$$\{x_{t_1}, x_{t_2}, \dots, x_{t_k}\}$$

is identical to that of the time shifted set

$$\{x_{t_1+h}, x_{t_2+h}, \dots, x_{t_k+h}\}.$$

That is,

$$\Pr\{x_{t_1} \leq c_1, \dots, x_{t_k} \leq c_k\} = \Pr\{x_{t_1+h} \leq c_1, \dots, x_{t_k+h} \leq c_k\} \quad (1.19)$$

for all $k = 1, 2, \dots$, all time points t_1, t_2, \dots, t_k , all numbers c_1, c_2, \dots, c_k , and all time shifts $h = 0, \pm 1, \pm 2, \dots$.

If a time series is strictly stationary, then all of the multivariate distribution functions for subsets of variables must agree with their counterparts in the shifted set for all values of the shift parameter h . For example, when $k = 1$, (1.19) implies that

$$\Pr\{x_s \leq c\} = \Pr\{x_t \leq c\} \quad (1.20)$$

for any time points s and t . This statement implies, for example, that the probability the value of a time series sampled hourly is negative at 1 AM is the same as at 10 AM.

In addition, if the mean function, μ_t , of the series exists, (1.20) implies that $\mu_s = \mu_t$ for all s and t , and hence μ_t must be constant. Note, for example, that a random walk process with drift is *not* strictly stationary because its mean function changes with time; see [Example 1.14](#).

When $k = 2$, we can write (1.19) as

$$\Pr\{x_s \leq c_1, x_t \leq c_2\} = \Pr\{x_{s+h} \leq c_1, x_{t+h} \leq c_2\} \quad (1.21)$$

for any time points s and t and shift h . Thus, if the variance function of the process exists, (1.20)–(1.21) imply that the autocovariance function of the series x_t satisfies

$$\gamma(s, t) = \gamma(s + h, t + h)$$

for all s and t and h . We may interpret this result by saying the autocovariance function of the process depends only on the time difference between s and t , and not on the actual times.

The version of stationarity in [Definition 1.6](#) is too strong for most applications. Moreover, it is difficult to assess strict stationarity from a single data set. Rather than imposing conditions on all possible distributions of a time series, we will use a milder version that imposes conditions only on the first two moments of the series. We now have the following definition.

Definition 1.7 A **weakly stationary** time series, x_t , is a finite variance process such that

- (i) the mean value function, μ_t , defined in (1.9) is constant and does not depend on time t , and
- (ii) the autocovariance function, $\gamma(s, t)$, defined in (1.10) depends on s and t only through their difference $|s - t|$.

Henceforth, we will use the term **stationary** to mean weakly stationary; if a process is stationary in the strict sense, we will use the term *strictly stationary*.

Stationarity requires regularity in the mean and autocorrelation functions so that these quantities (at least) may be estimated by averaging. It should be clear from the discussion of strict stationarity following [Definition 1.6](#) that a strictly stationary, finite variance, time series is also stationary. The converse is not true unless there are further conditions. One important case where stationarity implies strict stationarity is if the time series is Gaussian [meaning all finite distributions, (1.19), of the series are Gaussian]. We will make this concept more precise at the end of this section.

Because the mean function, $E(x_t) = \mu_t$, of a stationary time series is independent of time t , we will write

$$\mu_t = \mu. \quad (1.22)$$

Also, because the autocovariance function, $\gamma(s, t)$, of a stationary time series, x_t , depends on s and t only through their difference $|s - t|$, we may simplify the notation. Let $s = t + h$, where h represents the time shift or *lag*. Then

$$\gamma(t + h, t) = \text{cov}(x_{t+h}, x_t) = \text{cov}(x_h, x_0) = \gamma(h, 0)$$

because the time difference between times $t+h$ and t is the same as the time difference between times h and 0. Thus, the autocovariance function of a stationary time series does not depend on the time argument t . Henceforth, for convenience, we will drop the second argument of $\gamma(h, 0)$.

Definition 1.8 *The autocovariance function of a stationary time series will be written as*

$$\gamma(h) = \text{cov}(x_{t+h}, x_t) = E[(x_{t+h} - \mu)(x_t - \mu)]. \quad (1.23)$$

Definition 1.9 *The autocorrelation function (ACF) of a stationary time series will be written using (1.15) as*

$$\rho(h) = \frac{\gamma(t+h, t)}{\sqrt{\gamma(t+h, t+h)\gamma(t, t)}} = \frac{\gamma(h)}{\gamma(0)}. \quad (1.24)$$

The Cauchy–Schwarz inequality shows again that $-1 \leq \rho(h) \leq 1$ for all h , enabling one to assess the relative importance of a given autocorrelation value by comparing with the extreme values -1 and 1 .

Example 1.19 Stationarity of White Noise

The mean and autocovariance functions of the white noise series discussed in Example 1.8 and Example 1.16 are easily evaluated as $\mu_{wt} = 0$ and

$$\gamma_w(h) = \text{cov}(w_{t+h}, w_t) = \begin{cases} \sigma_w^2 & h = 0, \\ 0 & h \neq 0. \end{cases}$$

Thus, white noise satisfies the conditions of Definition 1.7 and is weakly stationary or stationary. If the white noise variates are also normally distributed or Gaussian, the series is also strictly stationary, as can be seen by evaluating (1.19) using the fact that the noise would also be iid. The autocorrelation function is given by $\rho_w(0) = 1$ and $\rho(h) = 0$ for $h \neq 0$.

Example 1.20 Stationarity of a Moving Average

The three-point moving average process of Example 1.9 is stationary because, from Example 1.13 and Example 1.17, the mean and autocovariance functions $\mu_{vt} = 0$, and

$$\gamma_v(h) = \begin{cases} \frac{3}{9}\sigma_w^2 & h = 0, \\ \frac{2}{9}\sigma_w^2 & h = \pm 1, \\ \frac{1}{9}\sigma_w^2 & h = \pm 2, \\ 0 & |h| > 2 \end{cases}$$

are independent of time t , satisfying the conditions of Definition 1.7.

The autocorrelation function is given by

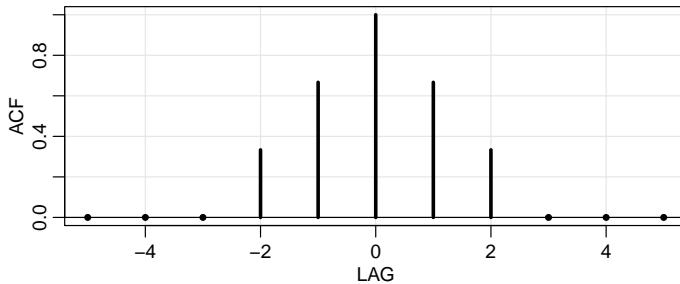


Fig. 1.12. Autocorrelation function of a three-point moving average.

$$\rho_v(h) = \begin{cases} 1 & h = 0, \\ \frac{2}{3} & h = \pm 1, \\ \frac{1}{3} & h = \pm 2, \\ 0 & |h| > 2. \end{cases}$$

Figure 1.12 shows a plot of the autocorrelations as a function of lag h . Note that the ACF is symmetric about lag zero.

Example 1.21 A Random Walk is Not Stationary

A random walk is not stationary because its autocovariance function, $\gamma(s, t) = \min\{s, t\}\sigma_w^2$, depends on time; see Example 1.18 and Problem 1.8. Also, the random walk with drift violates both conditions of Definition 1.7 because, as shown in Example 1.14, the mean function, $\mu_{xt} = \delta t$, is also a function of time t .

Example 1.22 Trend Stationarity

For example, if $x_t = \alpha + \beta t + y_t$, where y_t is stationary, then the mean function is $\mu_{x,t} = E(x_t) = \alpha + \beta t + \mu_y$, which is not independent of time. Therefore, the process is not stationary. The autocovariance function, however, is independent of time, because $\gamma_x(h) = \text{cov}(x_{t+h}, x_t) = E[(x_{t+h} - \mu_{x,t+h})(x_t - \mu_{x,t})] = E[(y_{t+h} - \mu_y)(y_t - \mu_y)] = \gamma_y(h)$. Thus, the model may be considered as having stationary behavior around a linear trend; this behavior is sometimes called *trend stationarity*. An example of such a process is the price of chicken series displayed in Figure 2.1.

The autocovariance function of a stationary process has several special properties. First, $\gamma(h)$ is *non-negative definite* (see Problem 1.25) ensuring that variances of linear combinations of the variates x_t will never be negative. That is, for any $n \geq 1$, and constants a_1, \dots, a_n ,

$$0 \leq \text{var}(a_1 x_1 + \dots + a_n x_n) = \sum_{j=1}^n \sum_{k=1}^n a_j a_k \gamma(j-k), \quad (1.25)$$

using Property 1.1. Also, the value at $h = 0$, namely

$$\gamma(0) = E[(x_t - \mu)^2] \quad (1.26)$$

is the variance of the time series and the Cauchy–Schwarz inequality implies

$$|\gamma(h)| \leq \gamma(0).$$

A final useful property, noted in a previous example, is that the autocovariance function of a stationary series is symmetric around the origin; that is,

$$\gamma(h) = \gamma(-h) \quad (1.27)$$

for all h . This property follows because

$$\gamma((t+h)-t) = \text{cov}(x_{t+h}, x_t) = \text{cov}(x_t, x_{t+h}) = \gamma(t-(t+h)),$$

which shows how to use the notation as well as proving the result.

When several series are available, a notion of stationarity still applies with additional conditions.

Definition 1.10 Two time series, say, x_t and y_t , are said to be **jointly stationary** if they are each stationary, and the cross-covariance function

$$\gamma_{xy}(h) = \text{cov}(x_{t+h}, y_t) = E[(x_{t+h} - \mu_x)(y_t - \mu_y)] \quad (1.28)$$

is a function only of lag h .

Definition 1.11 The **cross-correlation function (CCF)** of jointly stationary time series x_t and y_t is defined as

$$\rho_{xy}(h) = \frac{\gamma_{xy}(h)}{\sqrt{\gamma_x(0)\gamma_y(0)}}. \quad (1.29)$$

Again, we have the result $-1 \leq \rho_{xy}(h) \leq 1$ which enables comparison with the extreme values -1 and 1 when looking at the relation between x_{t+h} and y_t . The cross-correlation function is not generally symmetric about zero, i.e., typically $\rho_{xy}(h) \neq \rho_{xy}(-h)$. This is an important concept; it should be clear that $\text{cov}(x_2, y_1)$ and $\text{cov}(x_1, y_2)$ need not be the same. It is the case, however, that

$$\rho_{xy}(h) = \rho_{yx}(-h), \quad (1.30)$$

which can be shown by manipulations similar to those used to show (1.27).

Example 1.23 Joint Stationarity

Consider the two series, x_t and y_t , formed from the sum and difference of two successive values of a white noise process, say,

$$x_t = w_t + w_{t-1} \quad \text{and} \quad y_t = w_t - w_{t-1},$$

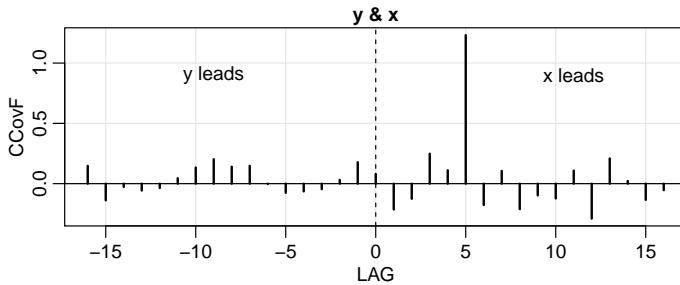


Fig. 1.13. Demonstration of the results of Example 1.24 when $\ell = 5$. The title shows which side leads.

where w_t are independent random variables with zero means and variance σ_w^2 . It is easy to show that $\gamma_x(0) = \gamma_y(0) = 2\sigma_w^2$ and $\gamma_x(1) = \gamma_x(-1) = \sigma_w^2$, $\gamma_y(1) = \gamma_y(-1) = -\sigma_w^2$. Also,

$$\gamma_{xy}(1) = \text{cov}(x_{t+1}, y_t) = \text{cov}(w_{t+1} + w_t, w_t - w_{t-1}) = \sigma_w^2$$

because only one term is nonzero. Similarly, $\gamma_{xy}(0) = 0$, $\gamma_{xy}(-1) = -\sigma_w^2$. We obtain, using (1.29),

$$\rho_{xy}(h) = \begin{cases} 0 & h = 0, \\ 1/2 & h = 1, \\ -1/2 & h = -1, \\ 0 & |h| \geq 2. \end{cases}$$

Clearly, the autocovariance and cross-covariance functions depend only on the lag separation, h , so the series are jointly stationary.

Example 1.24 Prediction Using Cross-Correlation

As a simple example of cross-correlation, consider the problem of determining possible leading or lagging relations between two series x_t and y_t . If the model

$$y_t = Ax_{t-\ell} + w_t$$

holds, the series x_t is said to *lead* y_t for $\ell > 0$ and is said to *lag* y_t for $\ell < 0$. Hence, the analysis of leading and lagging relations might be important in predicting the value of y_t from x_t . Assuming that the noise w_t is uncorrelated with the x_t series, the cross-covariance function can be computed as

$$\begin{aligned} \gamma_{yx}(h) &= \text{cov}(y_{t+h}, x_t) = \text{cov}(Ax_{t+h-\ell} + w_{t+h}, x_t) \\ &= \text{cov}(Ax_{t+h-\ell}, x_t) = A\gamma_x(h - \ell). \end{aligned}$$

Since (Cauchy–Schwarz) the largest absolute value of $\gamma_x(h - \ell)$ is $\gamma_x(0)$, i.e., when $h = \ell$, the cross-covariance function will look like the autocovariance of the input series x_t , and it will have a peak on the positive side if x_t leads y_t and a peak on the negative side if x_t lags y_t . Below is the R code of an example where x_t is white noise, $\ell = 5$, and with $\hat{\gamma}_{yx}(h)$ shown in Figure 1.13.

```
x = rnorm(100)
y = lag(x, -5) + rnorm(100)
ccf(y, x, ylab='CCovF', type='covariance')
```

The concept of weak stationarity forms the basis for much of the analysis performed with time series. The fundamental properties of the mean and autocovariance functions (1.22) and (1.23) are satisfied by many theoretical models that appear to generate plausible sample realizations. In Example 1.9 and Example 1.10, two series were generated that produced stationary looking realizations, and in Example 1.20, we showed that the series in Example 1.9 was, in fact, weakly stationary. Both examples are special cases of the so-called linear process.

Definition 1.12 A linear process, x_t , is defined to be a linear combination of white noise variates w_t , and is given by

$$x_t = \mu + \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}, \quad \sum_{j=-\infty}^{\infty} |\psi_j| < \infty. \quad (1.31)$$

For the linear process (see Problem 1.11), we may show that the autocovariance function is given by

$$\gamma_x(h) = \sigma_w^2 \sum_{j=-\infty}^{\infty} \psi_{j+h} \psi_j \quad (1.32)$$

for $h \geq 0$; recall that $\gamma_x(-h) = \gamma_x(h)$. This method exhibits the autocovariance function of the process in terms of the lagged products of the coefficients. We only need $\sum_{j=-\infty}^{\infty} \psi_j^2 < \infty$ for the process to have finite variance, but we will discuss this further in Chapter 5. Note that, for Example 1.9, we have $\psi_0 = \psi_{-1} = \psi_1 = 1/3$ and the result in Example 1.20 comes out immediately. The autoregressive series in Example 1.10 can also be put in this form, as can the general autoregressive moving average processes considered in Chapter 3.

Notice that the linear process (1.31) is dependent on the future ($j < 0$), the present ($j = 0$), and the past ($j > 0$). For the purpose of forecasting, a future dependent model will be useless. Consequently, we will focus on processes that do not depend on the future. Such models are called *causal*, and a causal linear process has $\psi_j = 0$ for $j < 0$; we will discuss this further in Chapter 3.

Finally, as previously mentioned, an important case in which a weakly stationary series is also strictly stationary is the normal or Gaussian series.

Definition 1.13 A process, $\{x_t\}$, is said to be a **Gaussian process** if the n -dimensional vectors $x = (x_{t_1}, x_{t_2}, \dots, x_{t_n})'$, for every collection of distinct time points t_1, t_2, \dots, t_n , and every positive integer n , have a multivariate normal distribution.

Defining the $n \times 1$ mean vector $E(x) \equiv \mu = (\mu_{t_1}, \mu_{t_2}, \dots, \mu_{t_n})'$ and the $n \times n$ covariance matrix as $\text{var}(x) \equiv \Gamma = \{\gamma(t_i, t_j); i, j = 1, \dots, n\}$, which is assumed to be positive definite, the multivariate normal density function can be written as

$$f(x) = (2\pi)^{-n/2} |\Gamma|^{-1/2} \exp \left\{ -\frac{1}{2} (x - \mu)' \Gamma^{-1} (x - \mu) \right\}, \quad (1.33)$$

for $x \in \mathbb{R}^n$, where $|\cdot|$ denotes the determinant.

We list some important items regarding linear and Gaussian processes.

- If a Gaussian time series, $\{x_t\}$, is weakly stationary, then μ_t is constant and $\gamma(t_i, t_j) = \gamma(|t_i - t_j|)$, so that the vector μ and the matrix Γ are independent of time. These facts imply that all the finite distributions, (1.33), of the series $\{x_t\}$ depend only on time lag and not on the actual times, and hence the series must be strictly stationary. In a sense, weak stationarity and normality go hand-in-hand in that we will base our analyses on the idea that it is enough for the first two moments to behave nicely. We use the multivariate normal density in the form given above as well as in a modified version, applicable to complex random variables throughout the text.
- A result called the *Wold Decomposition* (Theorem B.5) states that a stationary non-deterministic time series is a causal linear process (but with $\sum \psi_j^2 < \infty$). A linear process need not be Gaussian, but if a time series is Gaussian, then it is a causal linear process with $w_t \sim \text{iid } N(0, \sigma_w^2)$. Hence, stationary Gaussian processes form the basis of modeling many time series.
- It is not enough for the marginal distributions to be Gaussian for the process to be Gaussian. It is easy to construct a situation where X and Y are normal, but (X, Y) is not bivariate normal; e.g., let X and Z be independent normals and let $Y = Z$ if $XZ > 0$ and $Y = -Z$ if $XZ \leq 0$.

1.5 Estimation of Correlation

Although the theoretical autocorrelation and cross-correlation functions are useful for describing the properties of certain hypothesized models, most of the analyses must be performed using sampled data. This limitation means the sampled points x_1, x_2, \dots, x_n only are available for estimating the mean, autocovariance, and autocorrelation functions. From the point of view of classical statistics, this poses a problem because we will typically not have iid copies of x_t that are available for estimating the covariance and correlation functions. In the usual situation with only one realization, however, the assumption of stationarity becomes critical. Somehow, we must use averages over this single realization to estimate the population means and covariance functions.

Accordingly, if a time series is stationary, the mean function (1.22) $\mu_t = \mu$ is constant so that we can estimate it by the *sample mean*,

$$\bar{x} = \frac{1}{n} \sum_{t=1}^n x_t. \quad (1.34)$$

In our case, $E(\bar{x}) = \mu$, and the standard error of the estimate is the square root of $\text{var}(\bar{x})$, which can be computed using first principles (recall Property 1.1), and is given by

$$\begin{aligned}
\text{var}(\bar{x}) &= \text{var}\left(\frac{1}{n} \sum_{t=1}^n x_t\right) = \frac{1}{n^2} \text{cov}\left(\sum_{t=1}^n x_t, \sum_{s=1}^n x_s\right) \\
&= \frac{1}{n^2} \left(n\gamma_x(0) + (n-1)\gamma_x(1) + (n-2)\gamma_x(2) + \cdots + \gamma_x(n-1) \right. \\
&\quad \left. + (n-1)\gamma_x(-1) + (n-2)\gamma_x(-2) + \cdots + \gamma_x(1-n) \right) \\
&= \frac{1}{n} \sum_{h=-n}^n \left(1 - \frac{|h|}{n}\right) \gamma_x(h).
\end{aligned} \tag{1.35}$$

If the process is white noise, (1.35) reduces to the familiar σ_x^2/n recalling that $\gamma_x(0) = \sigma_x^2$. Note that, in the case of dependence, the standard error of \bar{x} may be smaller or larger than the white noise case depending on the nature of the correlation structure (see [Problem 1.19](#))

The theoretical autocovariance function, (1.23), is estimated by the sample autocovariance function defined as follows.

Definition 1.14 *The sample autocovariance function is defined as*

$$\hat{\gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x}), \tag{1.36}$$

with $\hat{\gamma}(-h) = \hat{\gamma}(h)$ for $h = 0, 1, \dots, n-1$.

The sum in (1.36) runs over a restricted range because x_{t+h} is not available for $t+h > n$. The estimator in (1.36) is preferred to the one that would be obtained by dividing by $n-h$ because (1.36) is a non-negative definite function. Recall that the autocovariance function of a stationary process is non-negative definite [(1.25); also, see [Problem 1.25](#)] ensuring that variances of linear combinations of the variates x_t will never be negative. And because a variance is never negative, the estimate of that variance

$$\widehat{\text{var}}(a_1 x_1 + \cdots + a_n x_n) = \sum_{j=1}^n \sum_{k=1}^n a_j a_k \hat{\gamma}(j-k),$$

should also be non-negative. The estimator in (1.36) guarantees this result, but no such guarantee exists if we divide by $n-h$. Note that neither dividing by n nor $n-h$ in (1.36) yields an unbiased estimator of $\gamma(h)$.

Definition 1.15 *The sample autocorrelation function is defined, analogously to (1.24), as*

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}. \tag{1.37}$$

The sample autocorrelation function has a sampling distribution that allows us to assess whether the data comes from a completely random or white series or whether correlations are statistically significant at some lags.

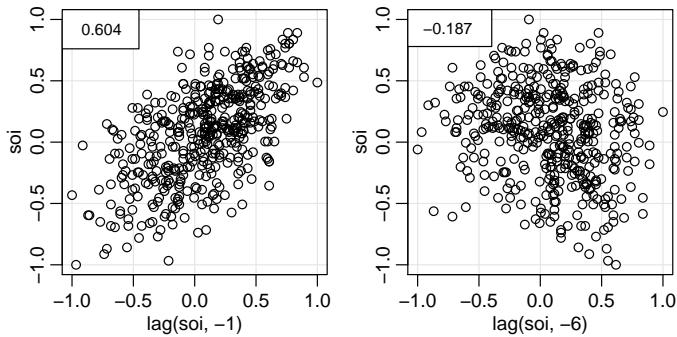


Fig. 1.14. Display for [Example 1.25](#). For the SOI series, the scatterplots show pairs of values one month apart (left) and six months apart (right). The estimated correlation is displayed in the box.

Example 1.25 Sample ACF and Scatterplots

Estimating autocorrelation is similar to estimating correlation in the usual setup where we have pairs of observations, say (x_i, y_i) , for $i = 1, \dots, n$. For example, if we have time series data x_t for $t = 1, \dots, n$, then the pairs of observations for estimating $\rho(h)$ are the $n - h$ pairs given by $\{(x_t, x_{t+h}); t = 1, \dots, n-h\}$. [Figure 1.14](#) shows an example using the SOI series where $\hat{\rho}(1) = .604$ and $\hat{\rho}(6) = -.187$. The following code was used for [Figure 1.14](#).

```
(r = round(acf(soi, 6, plot=FALSE)$acf[-1], 3)) # first 6 sample acf values
[1] 0.604 0.374 0.214 0.050 -0.107 -0.187
par(mfrow=c(1,2))
plot(lag(soi,-1), soi); legend('topleft', legend=r[1])
plot(lag(soi,-6), soi); legend('topleft', legend=r[6])
```

Property 1.2 Large-Sample Distribution of the ACF

Under general conditions,^{1.5} if x_t is white noise, then for n large, the sample ACF, $\hat{\rho}_x(h)$, for $h = 1, 2, \dots, H$, where H is fixed but arbitrary, is approximately normally distributed with zero mean and standard deviation given by

$$\sigma_{\hat{\rho}_x(h)} = \frac{1}{\sqrt{n}}. \quad (1.38)$$

Based on the previous result, we obtain a rough method of assessing whether peaks in $\hat{\rho}(h)$ are significant by determining whether the observed peak is outside the interval $\pm 2/\sqrt{n}$ (or plus/minus two standard errors); for a white noise sequence, approximately 95% of the sample ACFs should be within these limits. The applications of this property develop because many statistical modeling procedures depend on reducing a time series to a white noise series using various kinds of transformations. After such a procedure is applied, the plotted ACFs of the residuals should then lie roughly within the limits given above.

^{1.5} The general conditions are that x_t is iid with finite fourth moment. A sufficient condition for this to hold is that x_t is white Gaussian noise. Precise details are given in [Theorem A.7](#) in [Appendix A](#).

Example 1.26 A Simulated Time Series

To compare the sample ACF for various sample sizes to the theoretical ACF, consider a contrived set of data generated by tossing a fair coin, letting $x_t = 1$ when a head is obtained and $x_t = -1$ when a tail is obtained. Then, construct y_t as

$$y_t = 5 + x_t - .7x_{t-1}. \quad (1.39)$$

To simulate data, we consider two cases, one with a small sample size ($n = 10$) and another with a moderate sample size ($n = 100$).

```
set.seed(101010)
x1 = 2*rbinom(11, 1, .5) - 1    # simulated sequence of coin tosses
x2 = 2*rbinom(101, 1, .5) - 1
y1 = 5 + filter(x1, sides=1, filter=c(1,-.7))[-1]
y2 = 5 + filter(x2, sides=1, filter=c(1,-.7))[-1]
plot.ts(y1, type='s'); plot.ts(y2, type='s') # plot both series (not shown)
c(mean(y1), mean(y2))                      # the sample means
[1] 5.080  5.002
acf(y1, lag.max=4, plot=FALSE) # 1/sqrt(10) = .32
  Autocorrelations of series 'y1', by lag
    0     1     2     3     4
  1.000 -0.688  0.425 -0.306 -0.007
acf(y2, lag.max=4, plot=FALSE) # 1/sqrt(100) = .1
  Autocorrelations of series 'y2', by lag
    0     1     2     3     4
  1.000 -0.480 -0.002 -0.004  0.000
# Note that the sample ACF at lag zero is always 1 (Why?).
```

The theoretical ACF can be obtained from the model (1.39) using the fact that the mean of x_t is zero and the variance of x_t is one. It can be shown that

$$\rho_y(1) = \frac{-0.7}{1 + 0.7^2} = -0.47$$

and $\rho_y(h) = 0$ for $|h| > 1$ (Problem 1.24). It is interesting to compare the theoretical ACF with sample ACFs for the realization where $n = 10$ and the other realization where $n = 100$; note the increased variability in the smaller size sample.

Example 1.27 ACF of a Speech Signal

Computing the sample ACF as in the previous example can be thought of as matching the time series h units in the future, say, x_{t+h} against itself, x_t . Figure 1.15 shows the ACF of the speech series of Figure 1.3. The original series appears to contain a sequence of repeating short signals. The ACF confirms this behavior, showing repeating peaks spaced at about 106-109 points. Autocorrelation functions of the short signals appear, spaced at the intervals mentioned above. The distance between the repeating signals is known as the *pitch period* and is a fundamental parameter of interest in systems that encode and decipher speech. Because the series is sampled at 10,000 points per second, the pitch period appears to be between .0106 and .0109 seconds. To compute the sample ACF in R, use `acf(speech, 250)`.

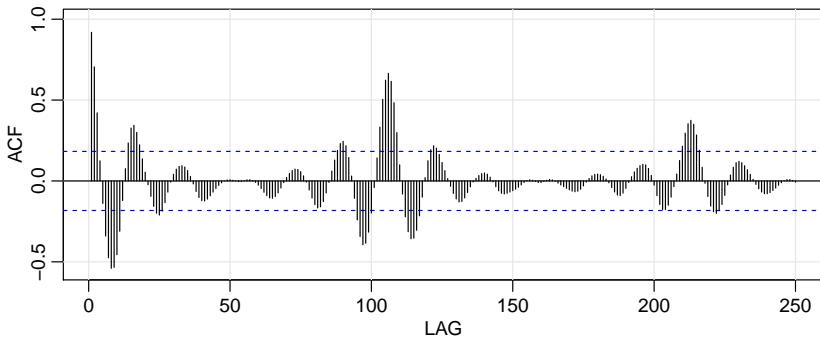


Fig. 1.15. ACF of the speech series.

Definition 1.16 The estimators for the cross-covariance function, $\hat{\gamma}_{xy}(h)$, as given in (1.28) and the cross-correlation, $\hat{\rho}_{xy}(h)$, in (1.11) are given, respectively, by the **sample cross-covariance function**

$$\hat{\gamma}_{xy}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(y_t - \bar{y}), \quad (1.40)$$

where $\hat{\gamma}_{xy}(-h) = \hat{\gamma}_{yx}(h)$ determines the function for negative lags, and the **sample cross-correlation function**

$$\hat{\rho}_{xy}(h) = \frac{\hat{\gamma}_{xy}(h)}{\sqrt{\hat{\gamma}_x(0)\hat{\gamma}_y(0)}}. \quad (1.41)$$

The sample cross-correlation function can be examined graphically as a function of lag h to search for leading or lagging relations in the data using the property mentioned in [Example 1.24](#) for the theoretical cross-covariance function. Because $-1 \leq \hat{\rho}_{xy}(h) \leq 1$, the practical importance of peaks can be assessed by comparing their magnitudes with their theoretical maximum values. Furthermore, for x_t and y_t independent linear processes of the form (1.31), we have the following property.

Property 1.3 Large-Sample Distribution of Cross-Correlation

The large sample distribution of $\hat{\rho}_{xy}(h)$ is normal with mean zero and

$$\sigma_{\hat{\rho}_{xy}} = \frac{1}{\sqrt{n}} \quad (1.42)$$

if at least one of the processes is independent white noise (see [Theorem A.8](#)).

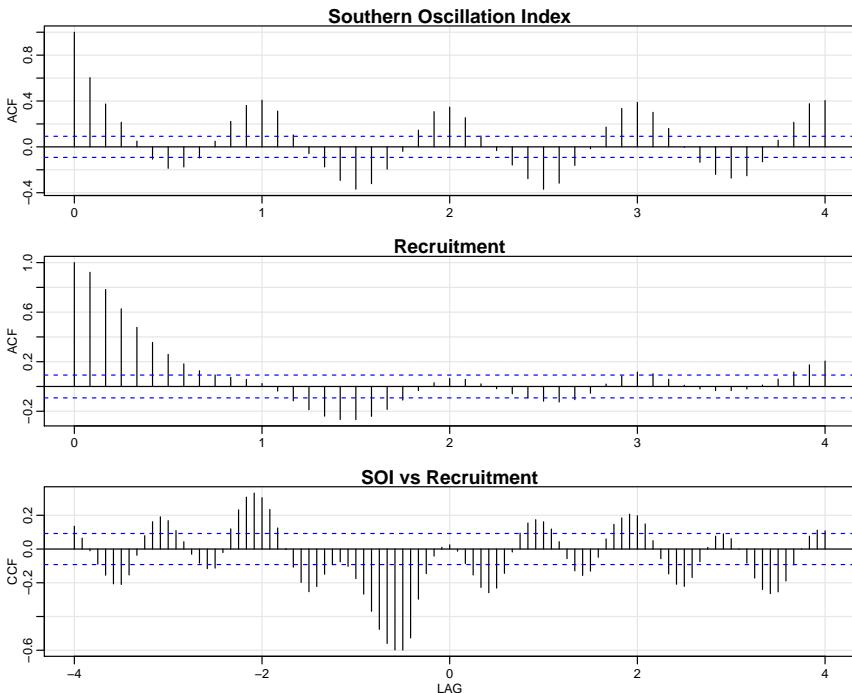


Fig. 1.16. Sample ACFs of the SOI series (top) and of the Recruitment series (middle), and the sample CCF of the two series (bottom); negative lags indicate SOI leads Recruitment. The lag axes are in terms of seasons (12 months).

Example 1.28 SOI and Recruitment Correlation Analysis

The autocorrelation and cross-correlation functions are also useful for analyzing the joint behavior of two stationary series whose behavior may be related in some unspecified way. In Example 1.5 (see Figure 1.5), we have considered simultaneous monthly readings of the SOI and the number of new fish (Recruitment) computed from a model. Figure 1.16 shows the autocorrelation and cross-correlation functions (ACFs and CCF) for these two series. Both of the ACFs exhibit periodicities corresponding to the correlation between values separated by 12 units. Observations 12 months or one year apart are strongly positively correlated, as are observations at multiples such as 24, 36, 48, Observations separated by six months are negatively correlated, showing that positive excursions tend to be associated with negative excursions six months removed.

The sample CCF in Figure 1.16, however, shows some departure from the cyclic component of each series and there is an obvious peak at $h = -6$. This result implies that SOI measured at time $t - 6$ months is associated with the Recruitment series at time t . We could say the SOI leads the Recruitment series by six months. The sign of the CCF is negative, leading to the conclusion that the two series move in different directions; that is, increases in SOI lead to decreases in Recruitment.

and vice versa. We will discover in [Chapter 2](#) that there is a relationship between the series, but the relationship is nonlinear. The dashed lines shown on the plots indicate $\pm 2/\sqrt{453}$ [see [\(1.42\)](#)], but since neither series is noise, these lines do not apply. To reproduce [Figure 1.16](#) in R, use the following commands:

```
par(mfrow=c(3,1))
acf(soi, 48, main="Southern Oscillation Index")
acf(rec, 48, main="Recruitment")
ccf(soi, rec, 48, main="SOI vs Recruitment", ylab="CCF")
```

Example 1.29 Prewhtening and Cross Correlation Analysis

Although we do not have all the tools necessary yet, it is worthwhile to discuss the idea of prewhitening a series prior to a cross-correlation analysis. The basic idea is simple; in order to use [Property 1.3](#), at least one of the series must be white noise. If this is not the case, there is no simple way to tell if a cross-correlation estimate is significantly different from zero. Hence, in [Example 1.28](#), we were only guessing at the linear dependence relationship between SOI and Recruitment.

For example, in [Figure 1.17](#) we generated two series, x_t and y_t , for $t = 1, \dots, 120$ independently as

$$x_t = 2 \cos(2\pi t \frac{1}{12}) + w_{t1} \quad \text{and} \quad y_t = 2 \cos(2\pi [t+5] \frac{1}{12}) + w_{t2}$$

where $\{w_{t1}, w_{t2}; t = 1, \dots, 120\}$ are all independent standard normals. The series are made to resemble SOI and Recruitment. The generated data are shown in the top row of the figure. The middle row of [Figure 1.17](#) shows the sample ACF of each series, each of which exhibits the cyclic nature of each series. The bottom row (left) of [Figure 1.17](#) shows the sample CCF between x_t and y_t , which appears to show cross-correlation even though the series are independent. The bottom row (right) also displays the sample CCF between x_t and the prewhitened y_t , which shows that the two sequences are uncorrelated. By prewhitening y_t , we mean that the signal has been removed from the data by running a regression of y_t on $\cos(2\pi t)$ and $\sin(2\pi t)$ [see [Example 2.10](#)] and then putting $\tilde{y}_t = y_t - \hat{y}_t$, where \hat{y}_t are the predicted values from the regression.

The following code will reproduce [Figure 1.17](#).

```
set.seed(1492)
num=120; t=1:num
X = ts(2*cos(2*pi*t/12) + rnorm(num), freq=12)
Y = ts(2*cos(2*pi*(t+5)/12) + rnorm(num), freq=12)
Yw = resid(lm(Y~ cos(2*pi*t/12) + sin(2*pi*t/12), na.action=NULL))
par(mfrow=c(3,2), mgp=c(1.6,.6,0), mar=c(3,3,1,1) )
plot(X)
plot(Y)
acf(X,48, ylab='ACF(X)')
acf(Y,48, ylab='ACF(Y)')
ccf(X,Y,24, ylab='CCF(X,Y)')
ccf(X,Yw,24, ylab='CCF(X,Yw)', ylim=c(-.6,.6))
```

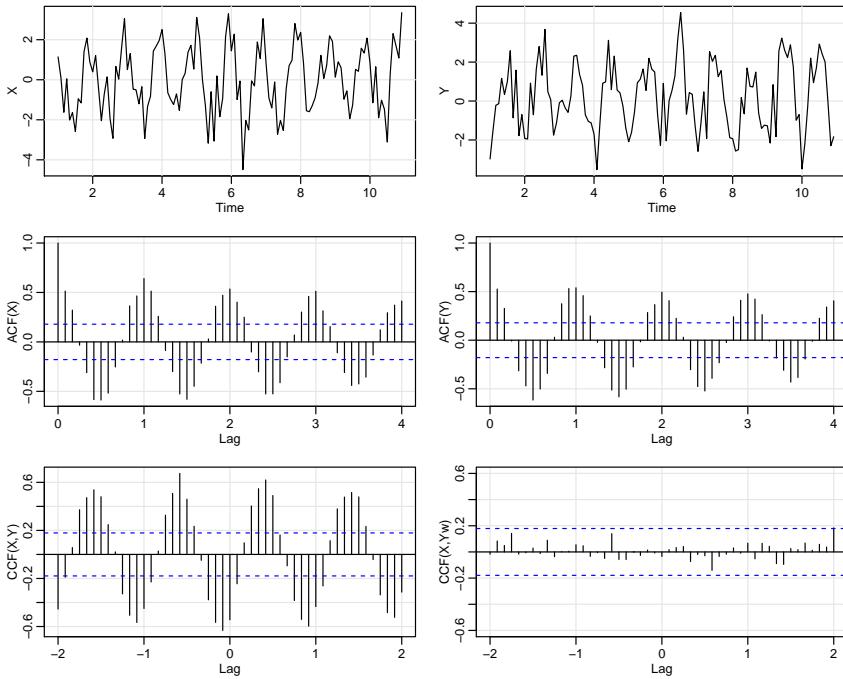


Fig. 1.17. Display for *Example 1.29*. Top row; The generated series. Middle row: The sample ACF of each series. Bottom row; The sample CCF of the series (left) and the sample CCF of the first series with the prewhitened second series (right).

1.6 Vector-Valued and Multidimensional Series

We frequently encounter situations in which the relationships between a number of jointly measured time series are of interest. For example, in the previous sections, we considered discovering the relationships between the SOI and Recruitment series. Hence, it will be useful to consider the notion of a *vector time series* $x_t = (x_{t1}, x_{t2}, \dots, x_{tp})'$, which contains as its components p univariate time series. We denote the $p \times 1$ column vector of the observed series as x_t . The row vector x_t' is its transpose. For the stationary case, the $p \times 1$ mean vector

$$\mu = E(x_t) \quad (1.43)$$

of the form $\mu = (\mu_{t1}, \mu_{t2}, \dots, \mu_{tp})'$ and the $p \times p$ autocovariance matrix

$$\Gamma(h) = E[(x_{t+h} - \mu)(x_t - \mu)'] \quad (1.44)$$

can be defined, where the elements of the matrix $\Gamma(h)$ are the cross-covariance functions

$$\gamma_{ij}(h) = E[(x_{t+h,i} - \mu_i)(x_{tj} - \mu_j)] \quad (1.45)$$

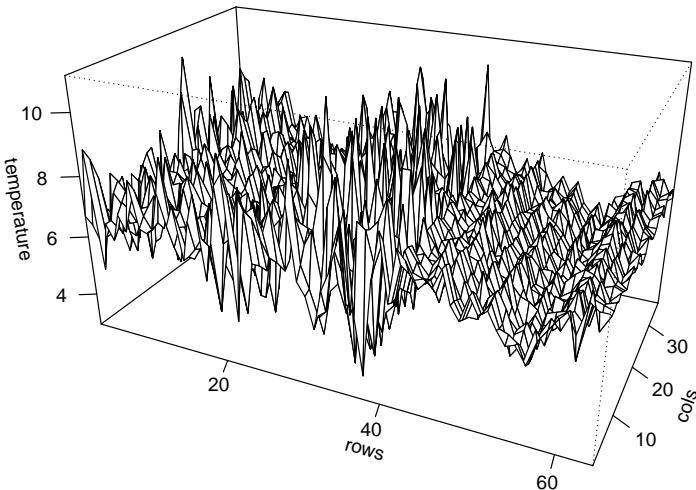


Fig. 1.18. Two-dimensional time series of temperature measurements taken on a rectangular field (64×36 with 17-foot spacing). Data are from Bazza et al. (1988).

for $i, j = 1, \dots, p$. Because $\gamma_{ij}(h) = \gamma_{ji}(-h)$, it follows that

$$\Gamma(-h) = \Gamma'(h). \quad (1.46)$$

Now, the *sample autocovariance matrix* of the vector series x_t is the $p \times p$ matrix of sample cross-covariances, defined as

$$\hat{\Gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x})', \quad (1.47)$$

where

$$\bar{x} = n^{-1} \sum_{t=1}^n x_t \quad (1.48)$$

denotes the $p \times 1$ *sample mean vector*. The symmetry property of the theoretical autocovariance (1.46) extends to the sample autocovariance (1.47), which is defined for negative values by taking

$$\hat{\Gamma}(-h) = \hat{\Gamma}(h)'. \quad (1.49)$$

In many applied problems, an observed series may be indexed by more than time alone. For example, the position in space of an experimental unit might be described by two coordinates, say, s_1 and s_2 . We may proceed in these cases by defining a *multidimensional process* x_s as a function of the $r \times 1$ vector $s = (s_1, s_2, \dots, s_r)'$, where s_i denotes the coordinate of the i th index.

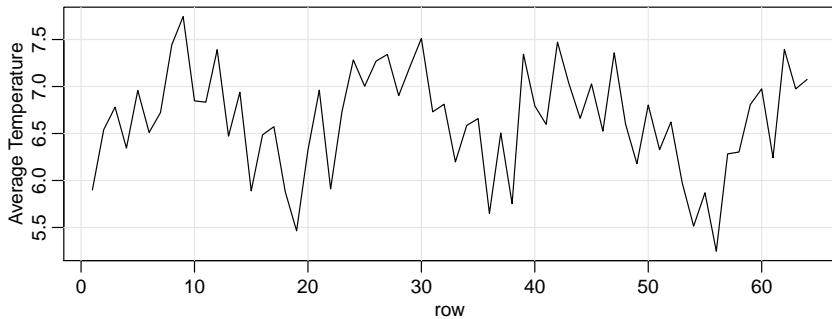


Fig. 1.19. Row averages of the two-dimensional soil temperature profile. $\bar{x}_{s_1 \cdot} = \sum_{s_2} x_{s_1, s_2} / 36$.

Example 1.30 Soil Surface Temperatures

As an example, the two-dimensional ($r = 2$) temperature series x_{s_1, s_2} in Figure 1.18 is indexed by a row number s_1 and a column number s_2 that represent positions on a 64×36 spatial grid set out on an agricultural field. The value of the temperature measured at row s_1 and column s_2 , is denoted by $x_s = x_{s_1, s_2}$. We can note from the two-dimensional plot that a distinct change occurs in the character of the two-dimensional surface starting at about row 40, where the oscillations along the row axis become fairly stable and periodic. For example, averaging over the 36 columns, we may compute an average value for each s_1 as in Figure 1.19. It is clear that the noise present in the first part of the two-dimensional series is nicely averaged out, and we see a clear and consistent temperature signal.

To generate Figure 1.18 and Figure 1.19 in R, use the following commands:

```
persp(1:64, 1:36, soiltemp, phi=25, theta=25, scale=FALSE, expand=4,
      ticktype="detailed", xlab="rows", ylab="cols", zlab="temperature")
plot.ts(rowMeans(soiltemp), xlab="row", ylab="Average Temperature")
```

The *autocovariance function* of a stationary multidimensional process, x_s , can be defined as a function of the multidimensional lag vector, say, $h = (h_1, h_2, \dots, h_r)'$, as

$$\gamma(h) = E[(x_{s+h} - \mu)(x_s - \mu)], \quad (1.50)$$

where

$$\mu = E(x_s) \quad (1.51)$$

does not depend on the spatial coordinate s . For the two dimensional temperature process, (1.50) becomes

$$\gamma(h_1, h_2) = E[(x_{s_1+h_1, s_2+h_2} - \mu)(x_{s_1, s_2} - \mu)], \quad (1.52)$$

which is a function of lag, both in the row (h_1) and column (h_2) directions.

The *multidimensional sample autocovariance function* is defined as

$$\hat{\gamma}(h) = (S_1 S_2 \cdots S_r)^{-1} \sum_{s_1} \sum_{s_2} \cdots \sum_{s_r} (x_{s+h} - \bar{x})(x_s - \bar{x}), \quad (1.53)$$

where $s = (s_1, s_2, \dots, s_r)'$ and the range of summation for each argument is $1 \leq s_i \leq S_i - h_i$, for $i = 1, \dots, r$. The mean is computed over the r -dimensional array, that is,

$$\bar{x} = (S_1 S_2 \cdots S_r)^{-1} \sum_{s_1} \sum_{s_2} \cdots \sum_{s_r} x_{s_1, s_2, \dots, s_r}, \quad (1.54)$$

where the arguments s_i are summed over $1 \leq s_i \leq S_i$. The multidimensional sample autocorrelation function follows, as usual, by taking the scaled ratio

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}. \quad (1.55)$$

Example 1.31 Sample ACF of the Soil Temperature Series

The autocorrelation function of the two-dimensional (2d) temperature process can be written in the form

$$\hat{\rho}(h_1, h_2) = \frac{\hat{\gamma}(h_1, h_2)}{\hat{\gamma}(0, 0)},$$

where

$$\hat{\gamma}(h_1, h_2) = (S_1 S_2)^{-1} \sum_{s_1} \sum_{s_2} (x_{s_1+h_1, s_2+h_2} - \bar{x})(x_{s_1, s_2} - \bar{x})$$

Figure 1.20 shows the autocorrelation function for the temperature data, and we note the systematic periodic variation that appears along the rows. The autocovariance over columns seems to be strongest for $h_1 = 0$, implying columns may form replicates of some underlying process that has a periodicity over the rows. This idea can be investigated by examining the mean series over columns as shown in **Figure 1.19**.

The easiest way (that we know of) to calculate a 2d ACF in R is by using the fast Fourier transform (FFT) as shown below. Unfortunately, the material needed to understand this approach is given in [Chapter 4, Section 4.3](#). The 2d autocovariance function is obtained in two steps and is contained in `cs` below; $\hat{\gamma}(0, 0)$ is the (1,1) element so that $\hat{\rho}(h_1, h_2)$ is obtained by dividing each element by that value. The 2d ACF is contained in `rs` below, and the rest of the code is simply to arrange the results to yield a nice display.

```
fs = Mod(fft(soiltemp-mean(soiltemp)))^2/(64*36)
cs = Re(fs, inverse=TRUE)/sqrt(64*36) # ACovF
rs = cs/cs[1,1] # ACF
rs2 = cbind(rs[1:41,21:2], rs[1:41,1:21])
rs3 = rbind(rs2[41:2,], rs2)
par(mar = c(1,2,5,0)+.1)
persp(-40:40, -20:20, rs3, phi=30, theta=30, expand=30, scale="FALSE",
      ticktype="detailed", xlab="row lags", ylab="column lags",
      zlab="ACF")
```

The sampling requirements for multidimensional processes are rather severe because values must be available over some uniform grid in order to compute the ACF. In some areas of application, such as in soil science, we may prefer to sample a limited number of rows or *transects* and hope these are essentially replicates of the

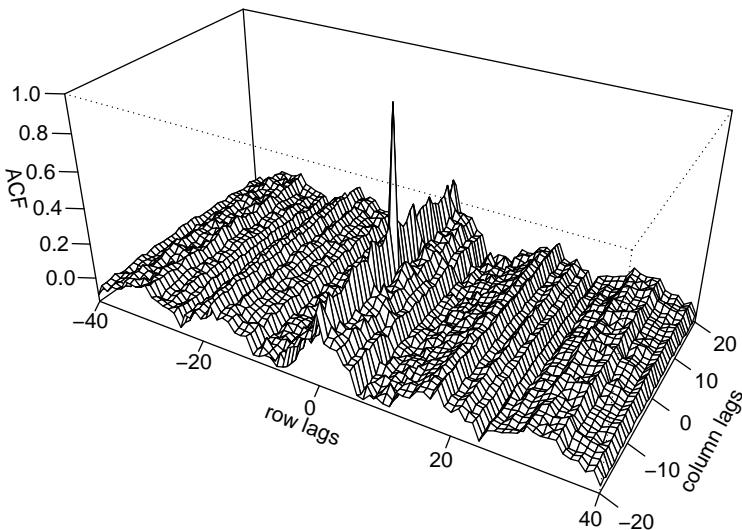


Fig. 1.20. Two-dimensional autocorrelation function for the soil temperature data.

basic underlying phenomenon of interest. One-dimensional methods can then be applied. When observations are irregular in time space, modifications to the estimators need to be made. Systematic approaches to the problems introduced by irregularly spaced observations have been developed by Journel and Huijbregts (1978) or Cressie (1993). We shall not pursue such methods in detail here, but it is worth noting that the introduction of the *variogram*

$$2V_x(h) = \text{var}\{x_{s+h} - x_s\} \quad (1.56)$$

and its sample estimator

$$2\hat{V}_x(h) = \frac{1}{N(h)} \sum_s (x_{s+h} - x_s)^2 \quad (1.57)$$

play key roles, where $N(h)$ denotes both the number of points located within h , and the sum runs over the points in the neighborhood. Clearly, substantial indexing difficulties will develop from estimators of the kind, and often it will be difficult to find non-negative definite estimators for the covariance function. **Problem 1.27** investigates the relation between the variogram and the autocovariance function in the stationary case.

Problems

Section 1.1

1.1 To compare the earthquake and explosion signals, plot the data displayed in [Figure 1.7](#) on the same graph using different colors or different line types and comment on the results. (The R code in [Example 1.11](#) may be of help on how to add lines to existing plots.)

1.2 Consider a signal-plus-noise model of the general form $x_t = s_t + w_t$, where w_t is Gaussian white noise with $\sigma_w^2 = 1$. Simulate and plot $n = 200$ observations from each of the following two models.

(a) $x_t = s_t + w_t$, for $t = 1, \dots, 200$, where

$$s_t = \begin{cases} 0, & t = 1, \dots, 100 \\ 10 \exp\left\{-\frac{(t-100)}{20}\right\} \cos(2\pi t/4), & t = 101, \dots, 200. \end{cases}$$

Hint:

```
s = c(rep(0, 100), 10*exp(-(1:100)/20)*cos(2*pi*1:100/4))
x = s + rnorm(200)
plot.ts(x)
```

(b) $x_t = s_t + w_t$, for $t = 1, \dots, 200$, where

$$s_t = \begin{cases} 0, & t = 1, \dots, 100 \\ 10 \exp\left\{-\frac{(t-100)}{200}\right\} \cos(2\pi t/4), & t = 101, \dots, 200. \end{cases}$$

(c) Compare the general appearance of the series (a) and (b) with the earthquake series and the explosion series shown in [Figure 1.7](#). In addition, plot (or sketch) and compare the signal modulators (a) $\exp\{-t/20\}$ and (b) $\exp\{-t/200\}$, for $t = 1, 2, \dots, 100$.

Section 1.2

1.3 (a) Generate $n = 100$ observations from the autoregression

$$x_t = -0.9x_{t-2} + w_t$$

with $\sigma_w = 1$, using the method described in [Example 1.10](#). Next, apply the moving average filter

$$v_t = (x_t + x_{t-1} + x_{t-2} + x_{t-3})/4$$

to x_t , the data you generated. Now plot x_t as a line and superimpose v_t as a dashed line. Comment on the behavior of x_t and how applying the moving average filter changes that behavior. [Hints: Use `v = filter(x, rep(1/4, 4), sides = 1)` for the filter and note that the R code in [Example 1.11](#) may be of help on how to add lines to existing plots.]

(b) Repeat (a) but with

$$x_t = \cos(2\pi t/4).$$

(c) Repeat (b) but with added $N(0, 1)$ noise,

$$x_t = \cos(2\pi t/4) + w_t.$$

(d) Compare and contrast (a)–(c); i.e., how does the moving average change each series.

Section 1.3

1.4 Show that the autocovariance function can be written as

$$\gamma(s, t) = E[(x_s - \mu_s)(x_t - \mu_t)] = E(x_s x_t) - \mu_s \mu_t,$$

where $E[x_t] = \mu_t$.

1.5 For the two series, x_t , in [Problem 1.2](#) (a) and (b):

- (a) Compute and plot the mean functions $\mu_x(t)$, for $t = 1, \dots, 200$.
- (b) Calculate the autocovariance functions, $\gamma_x(s, t)$, for $s, t = 1, \dots, 200$.

Section 1.4

1.6 Consider the time series

$$x_t = \beta_1 + \beta_2 t + w_t,$$

where β_1 and β_2 are known constants and w_t is a white noise process with variance σ_w^2 .

- (a) Determine whether x_t is stationary.
- (b) Show that the process $y_t = x_t - x_{t-1}$ is stationary.
- (c) Show that the mean of the moving average

$$v_t = \frac{1}{2q+1} \sum_{j=-q}^q x_{t-j}$$

is $\beta_1 + \beta_2 t$, and give a simplified expression for the autocovariance function.

1.7 For a moving average process of the form

$$x_t = w_{t-1} + 2w_t + w_{t+1},$$

where w_t are independent with zero means and variance σ_w^2 , determine the autocovariance and autocorrelation functions as a function of lag $h = s - t$ and plot the ACF as a function of h .

1.8 Consider the random walk with drift model

$$x_t = \delta + x_{t-1} + w_t,$$

for $t = 1, 2, \dots$, with $x_0 = 0$, where w_t is white noise with variance σ_w^2 .

- (a) Show that the model can be written as $x_t = \delta t + \sum_{k=1}^t w_k$.
- (b) Find the mean function and the autocovariance function of x_t .
- (c) Argue that x_t is not stationary.
- (d) Show $\rho_x(t-1, t) = \sqrt{\frac{t-1}{t}} \rightarrow 1$ as $t \rightarrow \infty$. What is the implication of this result?
- (e) Suggest a transformation to make the series stationary, and prove that the transformed series is stationary. (Hint: See [Problem 1.6b](#).)

1.9 A time series with a periodic component can be constructed from

$$x_t = U_1 \sin(2\pi\omega_0 t) + U_2 \cos(2\pi\omega_0 t),$$

where U_1 and U_2 are independent random variables with zero means and $E(U_1^2) = E(U_2^2) = \sigma^2$. The constant ω_0 determines the period or time it takes the process to make one complete cycle. Show that this series is weakly stationary with autocovariance function

$$\gamma(h) = \sigma^2 \cos(2\pi\omega_0 h).$$

1.10 Suppose we would like to predict a single stationary series x_t with zero mean and autocorrelation function $\gamma(h)$ at some time in the future, say, $t + \ell$, for $\ell > 0$.

- (a) If we predict using only x_t and some scale multiplier A , show that the mean-square prediction error

$$MSE(A) = E[(x_{t+\ell} - Ax_t)^2]$$

is minimized by the value

$$A = \rho(\ell).$$

- (b) Show that the minimum mean-square prediction error is

$$MSE(A) = \gamma(0)[1 - \rho^2(\ell)].$$

- (c) Show that if $x_{t+\ell} = Ax_t$, then $\rho(\ell) = 1$ if $A > 0$, and $\rho(\ell) = -1$ if $A < 0$.

1.11 Consider the linear process defined in [\(1.31\)](#).

- (a) Verify that the autocovariance function of the process is given by [\(1.32\)](#). Use the result to verify your answer to [Problem 1.7](#). Hint: For $h \geq 0$, $\text{cov}(x_{t+h}, x_t) = \text{cov}(\sum_k \psi_k w_{t+h-k}, \sum_j \psi_j w_{t-j})$. For each $j \in \mathbb{Z}$, the only “survivor” will be when $k = h + j$.
- (b) Show that x_t exists as a limit in mean square (see [Appendix A](#)).

1.12 For two weakly stationary series x_t and y_t , verify [\(1.30\)](#).

1.13 Consider the two series

$$x_t = w_t$$

$$y_t = w_t - \theta w_{t-1} + u_t,$$

where w_t and u_t are independent white noise series with variances σ_w^2 and σ_u^2 , respectively, and θ is an unspecified constant.

- (a) Express the ACF, $\rho_y(h)$, for $h = 0, \pm 1, \pm 2, \dots$ of the series y_t as a function of σ_w^2 , σ_u^2 , and θ .
- (b) Determine the CCF, $\rho_{xy}(h)$ relating x_t and y_t .
- (c) Show that x_t and y_t are jointly stationary.

1.14 Let x_t be a stationary normal process with mean μ_x and autocovariance function $\gamma(h)$. Define the nonlinear time series

$$y_t = \exp\{x_t\}.$$

- (a) Express the mean function $E(y_t)$ in terms of μ_x and $\gamma(0)$. The moment generating function of a normal random variable x with mean μ and variance σ^2 is

$$M_x(\lambda) = E[\exp\{\lambda x\}] = \exp\left\{\mu\lambda + \frac{1}{2}\sigma^2\lambda^2\right\}.$$

- (b) Determine the autocovariance function of y_t . The sum of the two normal random variables $x_{t+h} + x_t$ is still a normal random variable.

1.15 Let w_t , for $t = 0, \pm 1, \pm 2, \dots$ be a normal white noise process, and consider the series

$$x_t = w_t w_{t-1}.$$

Determine the mean and autocovariance function of x_t , and state whether it is stationary.

1.16 Consider the series

$$x_t = \sin(2\pi Ut),$$

$t = 1, 2, \dots$, where U has a uniform distribution on the interval $(0, 1)$.

- (a) Prove x_t is weakly stationary.
- (b) Prove x_t is not strictly stationary.

1.17 Suppose we have the linear process x_t generated by

$$x_t = w_t - \theta w_{t-1},$$

$t = 0, 1, 2, \dots$, where $\{w_t\}$ is independent and identically distributed with characteristic function $\phi_w(\cdot)$, and θ is a fixed constant. [Replace "characteristic function" with "moment generating function" if instructed to do so.]

- (a) Express the joint characteristic function of x_1, x_2, \dots, x_n , say,

$$\phi_{x_1, x_2, \dots, x_n}(\lambda_1, \lambda_2, \dots, \lambda_n),$$

in terms of $\phi_w(\cdot)$.

- (b) Deduce from (a) that x_t is strictly stationary.

- 1.18** Suppose that x_t is a linear process of the form (1.31). Prove

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty.$$

Section 1.5

- 1.19** Suppose $x_t = \mu + w_t + \theta w_{t-1}$, where $w_t \sim wn(0, \sigma_w^2)$.

- (a) Show that mean function is $E(x_t) = \mu$.
- (b) Show that the autocovariance function of x_t is given by $\gamma_x(0) = \sigma_w^2(1 + \theta^2)$, $\gamma_x(\pm 1) = \sigma_w^2\theta$, and $\gamma_x(h) = 0$ otherwise.
- (c) Show that x_t is stationary for all values of $\theta \in \mathbb{R}$.
- (d) Use (1.35) to calculate $\text{var}(\bar{x})$ for estimating μ when (i) $\theta = 1$, (ii) $\theta = 0$, and (iii) $\theta = -1$
- (e) In time series, the sample size n is typically large, so that $\frac{(n-1)}{n} \approx 1$. With this as a consideration, comment on the results of part (d); in particular, how does the accuracy in the estimate of the mean μ change for the three different cases?

- 1.20** (a) Simulate a series of $n = 500$ Gaussian white noise observations as in [Example 1.8](#) and compute the sample ACF, $\hat{\rho}(h)$, to lag 20. Compare the sample ACF you obtain to the actual ACF, $\rho(h)$. [Recall [Example 1.19](#).]

- (b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

- 1.21** (a) Simulate a series of $n = 500$ moving average observations as in [Example 1.9](#) and compute the sample ACF, $\hat{\rho}(h)$, to lag 20. Compare the sample ACF you obtain to the actual ACF, $\rho(h)$. [Recall [Example 1.20](#).]

- (b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

- 1.22** Although the model in [Problem 1.2\(a\)](#) is not stationary (Why?), the sample ACF can be informative. For the data you generated in that problem, calculate and plot the sample ACF, and then comment.

- 1.23** Simulate a series of $n = 500$ observations from the signal-plus-noise model presented in [Example 1.12](#) with $\sigma_w^2 = 1$. Compute the sample ACF to lag 100 of the data you generated and comment.

- 1.24** For the time series y_t described in [Example 1.26](#), verify the stated result that $\rho_y(1) = -0.47$ and $\rho_y(h) = 0$ for $h > 1$.

1.25 A real-valued function $g(t)$, defined on the integers, is non-negative definite if and only if

$$\sum_{i=1}^n \sum_{j=1}^n a_i g(t_i - t_j) a_j \geq 0$$

for all positive integers n and for all vectors $a = (a_1, a_2, \dots, a_n)'$ and $t = (t_1, t_2, \dots, t_n)'$. For the matrix $G = \{g(t_i - t_j); i, j = 1, 2, \dots, n\}$, this implies that $a' G a \geq 0$ for all vectors a . It is called positive definite if we can replace ' \geq ' with ' $>$ ' for all $a \neq 0$, the zero vector.

- (a) Prove that $\gamma(h)$, the autocovariance function of a stationary process, is a non-negative definite function.
- (b) Verify that the sample autocovariance $\hat{\gamma}(h)$ is a non-negative definite function.

Section 1.6

1.26 Consider a collection of time series $x_{1t}, x_{2t}, \dots, x_{Nt}$ that are observing some common signal μ_t observed in noise processes $e_{1t}, e_{2t}, \dots, e_{Nt}$, with a model for the j -th observed series given by

$$x_{jt} = \mu_t + e_{jt}.$$

Suppose the noise series have zero means and are uncorrelated for different j . The common autocovariance functions of all series are given by $\gamma_e(s, t)$. Define the sample mean

$$\bar{x}_t = \frac{1}{N} \sum_{j=1}^N x_{jt}.$$

- (a) Show that $E[\bar{x}_t] = \mu_t$.
- (b) Show that $E[(\bar{x}_t - \mu_t)^2] = N^{-1} \gamma_e(t, t)$.
- (c) How can we use the results in estimating the common signal?

1.27 A concept used in *geostatistics*, see Journel and Huijbregts (1978) or Cressie (1993), is that of the *variogram*, defined for a spatial process x_s , $s = (s_1, s_2)$, for $s_1, s_2 = 0, \pm 1, \pm 2, \dots$, as

$$V_x(h) = \frac{1}{2} E[(x_{s+h} - x_s)^2],$$

where $h = (h_1, h_2)$, for $h_1, h_2 = 0, \pm 1, \pm 2, \dots$. Show that, for a stationary process, the variogram and autocovariance functions can be related through

$$V_x(h) = \gamma(0) - \gamma(h),$$

where $\gamma(h)$ is the usual lag h covariance function and $0 = (0, 0)$. Note the easy extension to any spatial dimension.

The following problems require the material given in Appendix A

1.28 Suppose $x_t = \beta_0 + \beta_1 t$, where β_0 and β_1 are constants. Prove as $n \rightarrow \infty$, $\hat{\rho}_x(h) \rightarrow 1$ for fixed h , where $\hat{\rho}_x(h)$ is the ACF (1.37).

1.29 (a) Suppose x_t is a weakly stationary time series with mean zero and with absolutely summable autocovariance function, $\gamma(h)$, such that

$$\sum_{h=-\infty}^{\infty} \gamma(h) = 0.$$

Prove that $\sqrt{n} \bar{x} \xrightarrow{P} 0$, where \bar{x} is the sample mean (1.34).

(b) Give an example of a process that satisfies the conditions of part (a). What is special about this process?

1.30 Let x_t be a linear process of the form (A.43)–(A.44). If we define

$$\tilde{\gamma}(h) = n^{-1} \sum_{t=1}^n (x_{t+h} - \mu_x)(x_t - \mu_x),$$

show that

$$n^{1/2} (\tilde{\gamma}(h) - \hat{\gamma}(h)) = o_p(1).$$

Hint: The Markov Inequality

$$\Pr\{|x| \geq \epsilon\} < \frac{\text{E}|x|}{\epsilon}$$

can be helpful for the cross-product terms.

1.31 For a linear process of the form

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j},$$

where $\{w_t\}$ satisfies the conditions of Theorem A.7 and $|\phi| < 1$, show that

$$\sqrt{n} \frac{(\hat{\rho}_x(1) - \rho_x(1))}{\sqrt{1 - \rho_x^2(1)}} \xrightarrow{d} N(0, 1),$$

and construct a 95% confidence interval for ϕ when $\hat{\rho}_x(1) = .64$ and $n = 100$.

1.32 Let $\{x_t; t = 0, \pm 1, \pm 2, \dots\}$ be iid $(0, \sigma^2)$.

- (a) For $h \geq 1$ and $k \geq 1$, show that $x_t x_{t+h}$ and $x_s x_{s+k}$ are uncorrelated for all $s \neq t$.
- (b) For fixed $h \geq 1$, show that the $h \times 1$ vector

$$\sigma^{-2} n^{-1/2} \sum_{t=1}^n (x_t x_{t+1}, \dots, x_t x_{t+h})' \xrightarrow{d} (z_1, \dots, z_h)'$$

where z_1, \dots, z_h are iid $N(0, 1)$ random variables. [Hint: Use the Cramér-Wold device.]

(c) Show, for each $h \geq 1$,

$$n^{-1/2} \left[\sum_{t=1}^n x_t x_{t+h} - \sum_{t=1}^{n-h} (x_t - \bar{x})(x_{t+h} - \bar{x}) \right] \xrightarrow{P} 0 \quad \text{as } n \rightarrow \infty$$

where $\bar{x} = n^{-1} \sum_{t=1}^n x_t$.

(d) Noting that $n^{-1} \sum_{t=1}^n x_t^2 \xrightarrow{P} \sigma^2$ by the WLLN, conclude that

$$n^{1/2} [\hat{\rho}(1), \dots, \hat{\rho}(h)]' \xrightarrow{d} (z_1, \dots, z_h)'$$

where $\hat{\rho}(h)$ is the sample ACF of the data x_1, \dots, x_n .

Chapter 2

Time Series Regression and Exploratory Data Analysis

In this chapter we introduce classical multiple linear regression in a time series context, model selection, exploratory data analysis for preprocessing nonstationary time series (for example trend removal), the concept of differencing and the backshift operator, variance stabilization, and nonparametric smoothing of time series.

2.1 Classical Regression in the Time Series Context

We begin our discussion of linear regression in the time series context by assuming some output or *dependent* time series, say, x_t , for $t = 1, \dots, n$, is being influenced by a collection of possible inputs or *independent* series, say, $z_{t1}, z_{t2}, \dots, z_{tq}$, where we first regard the inputs as fixed and known. This assumption, necessary for applying conventional linear regression, will be relaxed later on. We express this relation through the *linear regression model*

$$x_t = \beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \dots + \beta_q z_{tq} + w_t, \quad (2.1)$$

where $\beta_0, \beta_1, \dots, \beta_q$ are unknown fixed regression coefficients, and $\{w_t\}$ is a random error or noise process consisting of independent and identically distributed (iid) normal variables with mean zero and variance σ_w^2 . For time series regression, it is rarely the case that the noise is white, and we will need to eventually relax that assumption. A more general setting within which to embed mean square estimation and linear regression is given in [Appendix B](#), where we introduce Hilbert spaces and the Projection Theorem.

Example 2.1 Estimating a Linear Trend

Consider the monthly price (per pound) of a chicken in the US from mid-2001 to mid-2016 (180 months), say x_t , shown in [Figure 2.1](#). There is an obvious upward trend in the series, and we might use simple linear regression to estimate that trend by fitting the model

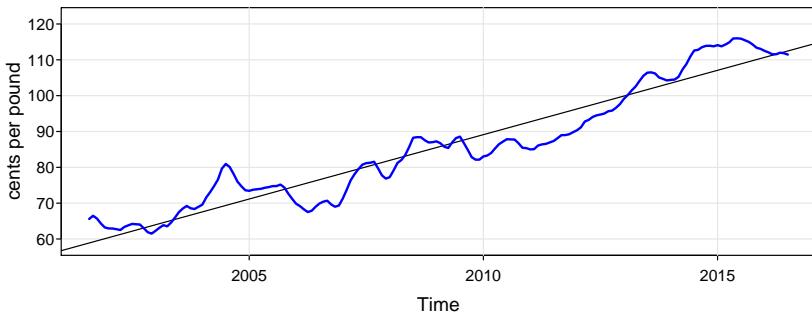


Fig. 2.1. The price of chicken: monthly whole bird spot price, Georgia docks, US cents per pound, August 2001 to July 2016, with fitted linear trend line.

$$x_t = \beta_0 + \beta_1 z_t + w_t, \quad z_t = 2001\frac{7}{12}, 2001\frac{8}{12}, \dots, 2016\frac{6}{12}.$$

This is in the form of the regression model (2.1) with $q = 1$. Note that we are making the assumption that the errors, w_t , are an iid normal sequence, which may not be true; the problem of autocorrelated errors is discussed in detail in [Chapter 3](#).

In ordinary least squares (OLS), we minimize the error sum of squares

$$Q = \sum_{t=1}^n w_t^2 = \sum_{t=1}^n (x_t - [\beta_0 + \beta_1 z_t])^2$$

with respect to β_i for $i = 0, 1$. In this case we can use simple calculus to evaluate $\partial Q / \partial \beta_i = 0$ for $i = 0, 1$, to obtain two equations to solve for the β s. The OLS estimates of the coefficients are explicit and given by

$$\hat{\beta}_1 = \frac{\sum_{t=1}^n (x_t - \bar{x})(z_t - \bar{z})}{\sum_{t=1}^n (z_t - \bar{z})^2} \quad \text{and} \quad \hat{\beta}_0 = \bar{x} - \hat{\beta}_1 \bar{z},$$

where $\bar{x} = \sum_t x_t / n$ and $\bar{z} = \sum_t z_t / n$ are the respective sample means.

Using R, we obtained the estimated slope coefficient of $\hat{\beta}_1 = 3.59$ (with a standard error of .08) yielding a significant estimated increase of about 3.6 cents per year. Finally, [Figure 2.1](#) shows the data with the estimated trend line superimposed. R code with partial output:

```
summary(fit <- lm(chicken~time(chicken), na.action=NULL))
      Estimate Std. Error t.value
(Intercept) -7131.02     162.41   -43.9
time(chicken)    3.59      0.08    44.4
--
Residual standard error: 4.7 on 178 degrees of freedom
plot(chicken, ylab="cents per pound")
abline(fit)      # add the fitted line
```

The multiple linear regression model described by (2.1) can be conveniently written in a more general notation by defining the column vectors $\mathbf{z}_t = (1, z_{t1}, z_{t2}, \dots, z_{tq})'$

and $\beta = (\beta_0, \beta_1, \dots, \beta_q)'$, where ' denotes transpose, so (2.1) can be written in the alternate form

$$x_t = \beta_0 + \beta_1 z_{t1} + \dots + \beta_q z_{tq} + w_t = \beta' z_t + w_t. \quad (2.2)$$

where $w_t \sim \text{iid } N(0, \sigma_w^2)$. As in the previous example, OLS estimation finds the coefficient vector β that minimizes the error sum of squares

$$Q = \sum_{t=1}^n w_t^2 = \sum_{t=1}^n (x_t - \beta' z_t)^2, \quad (2.3)$$

with respect to $\beta_0, \beta_1, \dots, \beta_q$. This minimization can be accomplished by differentiating (2.3) with respect to the vector β or by using the properties of projections. Either way, the solution must satisfy $\sum_{t=1}^n (x_t - \hat{\beta}' z_t) z_t' = 0$. This procedure gives the *normal equations*

$$\left(\sum_{t=1}^n z_t z_t' \right) \hat{\beta} = \sum_{t=1}^n z_t x_t. \quad (2.4)$$

If $\sum_{t=1}^n z_t z_t'$ is non-singular, the least squares estimate of β is

$$\hat{\beta} = \left(\sum_{t=1}^n z_t z_t' \right)^{-1} \sum_{t=1}^n z_t x_t.$$

The minimized error sum of squares (2.3), denoted *SSE*, can be written as

$$SSE = \sum_{t=1}^n (x_t - \hat{\beta}' z_t)^2. \quad (2.5)$$

The ordinary least squares estimators are unbiased, i.e., $E(\hat{\beta}) = \beta$, and have the smallest variance within the class of linear unbiased estimators.

If the errors w_t are normally distributed, $\hat{\beta}$ is also the maximum likelihood estimator for β and is normally distributed with

$$\text{cov}(\hat{\beta}) = \sigma_w^2 C, \quad (2.6)$$

where

$$C = \left(\sum_{t=1}^n z_t z_t' \right)^{-1} \quad (2.7)$$

is a convenient notation. An unbiased estimator for the variance σ_w^2 is

$$s_w^2 = MSE = \frac{SSE}{n - (q + 1)}, \quad (2.8)$$

where *MSE* denotes the *mean squared error*. Under the normal assumption,

$$t = \frac{(\hat{\beta}_i - \beta_i)}{s_w \sqrt{c_{ii}}} \quad (2.9)$$

Table 2.1. Analysis of Variance for Regression

Source	df	Sum of Squares	Mean Square	F
$z_{t,r+1:q}$	$q - r$	$SSR = SSE_r - SSE$	$MSR = SSR/(q - r)$	$F = \frac{MSR}{MSE}$
Error	$n - (q + 1)$	SSE	$MSE = SSE/(n - q - 1)$	

has the t-distribution with $n - (q + 1)$ degrees of freedom; c_{ii} denotes the i -th diagonal element of C , as defined in (2.7). This result is often used for individual tests of the null hypothesis $H_0: \beta_i = 0$ for $i = 1, \dots, q$.

Various competing models are often of interest to isolate or select the best subset of independent variables. Suppose a proposed model specifies that only a subset $r < q$ independent variables, say, $z_{t,1:r} = \{z_{t1}, z_{t2}, \dots, z_{tr}\}$ is influencing the dependent variable x_t . The reduced model is

$$x_t = \beta_0 + \beta_1 z_{t1} + \dots + \beta_r z_{tr} + w_t \quad (2.10)$$

where $\beta_1, \beta_2, \dots, \beta_r$ are a subset of coefficients of the original q variables.

The null hypothesis in this case is $H_0: \beta_{r+1} = \dots = \beta_q = 0$. We can test the reduced model (2.10) against the full model (2.2) by comparing the error sums of squares under the two models using the F -statistic

$$F = \frac{(SSE_r - SSE)/(q - r)}{SSE/(n - q - 1)} = \frac{MSR}{MSE}, \quad (2.11)$$

where SSE_r is the error sum of squares under the reduced model (2.10). Note that $SSE_r \geq SSE$ because the full model has more parameters. If $H_0: \beta_{r+1} = \dots = \beta_q = 0$ is true, then $SSE_r \approx SSE$ because the estimates of those β s will be close to 0. Hence, we do not believe H_0 if $SSR = SSE_r - SSE$ is big. Under the null hypothesis, (2.11) has a central F -distribution with $q - r$ and $n - q - 1$ degrees of freedom when (2.10) is the correct model.

These results are often summarized in an *Analysis of Variance (ANOVA)* table as given in Table 2.1 for this particular case. The difference in the numerator is often called the regression sum of squares (SSR). The null hypothesis is rejected at level α if $F > F_{n-q-1}^{q-r}(\alpha)$, the $1 - \alpha$ percentile of the F distribution with $q - r$ numerator and $n - q - 1$ denominator degrees of freedom.

A special case of interest is the null hypothesis $H_0: \beta_1 = \dots = \beta_q = 0$. In this case $r = 0$, and the model in (2.10) becomes

$$x_t = \beta_0 + w_t.$$

We may measure the proportion of variation accounted for by all the variables using

$$R^2 = \frac{SSE_0 - SSE}{SSE_0}, \quad (2.12)$$

where the residual sum of squares under the reduced model is

$$SSE_0 = \sum_{t=1}^n (x_t - \bar{x})^2. \quad (2.13)$$

In this case SSE_0 is the sum of squared deviations from the mean \bar{x} and is otherwise known as the adjusted total sum of squares. The measure R^2 is called the *coefficient of determination*.

The techniques discussed in the previous paragraph can be used to test various models against one another using the F test given in (2.11). These tests have been used in the past in a stepwise manner, where variables are added or deleted when the values from the F -test either exceed or fail to exceed some predetermined levels. The procedure, called *stepwise multiple regression*, is useful in arriving at a set of useful variables. An alternative is to focus on a procedure for *model selection* that does not proceed sequentially, but simply evaluates each model on its own merits. Suppose we consider a normal regression model with k coefficients and denote the *maximum likelihood estimator* for the variance as

$$\hat{\sigma}_k^2 = \frac{SSE(k)}{n}, \quad (2.14)$$

where $SSE(k)$ denotes the residual sum of squares under the model with k regression coefficients. Then, Akaike (1969, 1973, 1974) suggested measuring the goodness of fit for this particular model by balancing the error of the fit against the number of parameters in the model; we define the following.^{2.1}

Definition 2.1 Akaike's Information Criterion (AIC)

$$AIC = \log \hat{\sigma}_k^2 + \frac{n + 2k}{n}, \quad (2.15)$$

where $\hat{\sigma}_k^2$ is given by (2.14) and k is the number of parameters in the model.

The value of k yielding the minimum AIC specifies the best model. The idea is roughly that minimizing $\hat{\sigma}_k^2$ would be a reasonable objective, except that it decreases monotonically as k increases. Therefore, we ought to penalize the error variance by a term proportional to the number of parameters. The choice for the penalty term given by (2.15) is not the only one, and a considerable literature is available advocating different penalty terms. A corrected form, suggested by Sugiura (1978), and expanded by Hurvich and Tsai (1989), can be based on small-sample distributional results for the linear regression model (details are provided in Problem 2.4 and Problem 2.5). The corrected form is defined as follows.

Definition 2.2 AIC, Bias Corrected (AICc)

$$AICc = \log \hat{\sigma}_k^2 + \frac{n + k}{n - k - 2}, \quad (2.16)$$

^{2.1} Formally, AIC is defined as $-2 \log L_k + 2k$ where L_k is the maximized likelihood and k is the number of parameters in the model. For the normal regression problem, AIC can be reduced to the form given by (2.15). AIC is an estimate of the Kullback-Leibler discrepancy between a true model and a candidate model; see Problem 2.4 and Problem 2.5 for further details.

where $\hat{\sigma}_k^2$ is given by (2.14), k is the number of parameters in the model, and n is the sample size.

We may also derive a correction term based on Bayesian arguments, as in Schwarz (1978), which leads to the following.

Definition 2.3 Bayesian Information Criterion (BIC)

$$\text{BIC} = \log \hat{\sigma}_k^2 + \frac{k \log n}{n}, \quad (2.17)$$

using the same notation as in Definition 2.2.

BIC is also called the Schwarz Information Criterion (SIC); see also Rissanen (1978) for an approach yielding the same statistic based on a minimum description length argument. Notice that the penalty term in BIC is much larger than in AIC, consequently, BIC tends to choose smaller models. Various simulation studies have tended to verify that BIC does well at getting the correct order in large samples, whereas AICc tends to be superior in smaller samples where the relative number of parameters is large; see McQuarrie and Tsai (1998) for detailed comparisons. In fitting regression models, two measures that have been used in the past are adjusted R-squared, which is essentially s_w^2 , and Mallows C_p , Mallows (1973), which we do not consider in this context.

Example 2.2 Pollution, Temperature and Mortality

The data shown in Figure 2.2 are extracted series from a study by Shumway et al. (1988) of the possible effects of temperature and pollution on weekly mortality in Los Angeles County. Note the strong seasonal components in all of the series, corresponding to winter-summer variations and the downward trend in the cardiovascular mortality over the 10-year period.

A scatterplot matrix, shown in Figure 2.3, indicates a possible linear relation between mortality and the pollutant particulates and a possible relation to temperature. Note the curvilinear shape of the temperature mortality curve, indicating that higher temperatures as well as lower temperatures are associated with increases in cardiovascular mortality.

Based on the scatterplot matrix, we entertain, tentatively, four models where M_t denotes cardiovascular mortality, T_t denotes temperature and P_t denotes the particulate levels. They are

$$M_t = \beta_0 + \beta_1 t + w_t \quad (2.18)$$

$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + w_t \quad (2.19)$$

$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + \beta_3(T_t - T.)^2 + w_t \quad (2.20)$$

$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + \beta_3(T_t - T.)^2 + \beta_4 P_t + w_t \quad (2.21)$$

where we adjust temperature for its mean, $T. = 74.26$, to avoid collinearity problems. It is clear that (2.18) is a trend only model, (2.19) is linear temperature, (2.20)

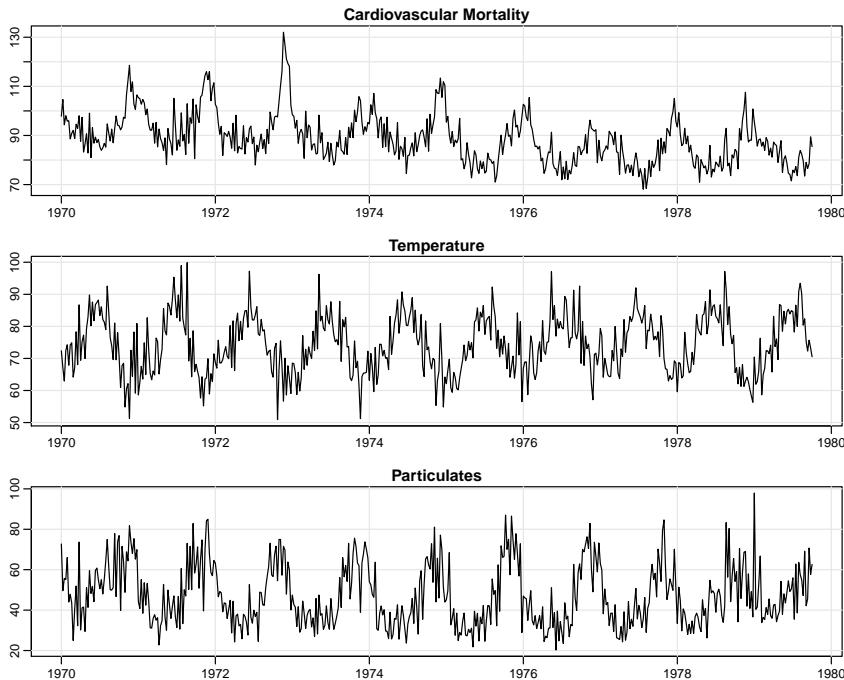


Fig. 2.2. Average weekly cardiovascular mortality (top), temperature (middle) and particulate pollution (bottom) in Los Angeles County. There are 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979.

Table 2.2. Summary Statistics for Mortality Models

Model	k	SSE	df	MSE	R^2	AIC	BIC
(2.18)	2	40,020	506	79.0	.21	5.38	5.40
(2.19)	3	31,413	505	62.2	.38	5.14	5.17
(2.20)	4	27,985	504	55.5	.45	5.03	5.07
(2.21)	5	20,508	503	40.8	.60	4.72	4.77

is curvilinear temperature and (2.21) is curvilinear temperature and pollution. We summarize some of the statistics given for this particular case in Table 2.2.

We note that each model does substantially better than the one before it and that the model including temperature, temperature squared, and particulates does the best, accounting for some 60% of the variability and with the best value for AIC and BIC (because of the large sample size, AIC and AICC are nearly the same). Note that one can compare any two models using the residual sums of squares and (2.11). Hence, a model with only trend could be compared to the full model, $H_0: \beta_2 = \beta_3 = \beta_4 = 0$, using $q = 4, r = 1, n = 508$, and

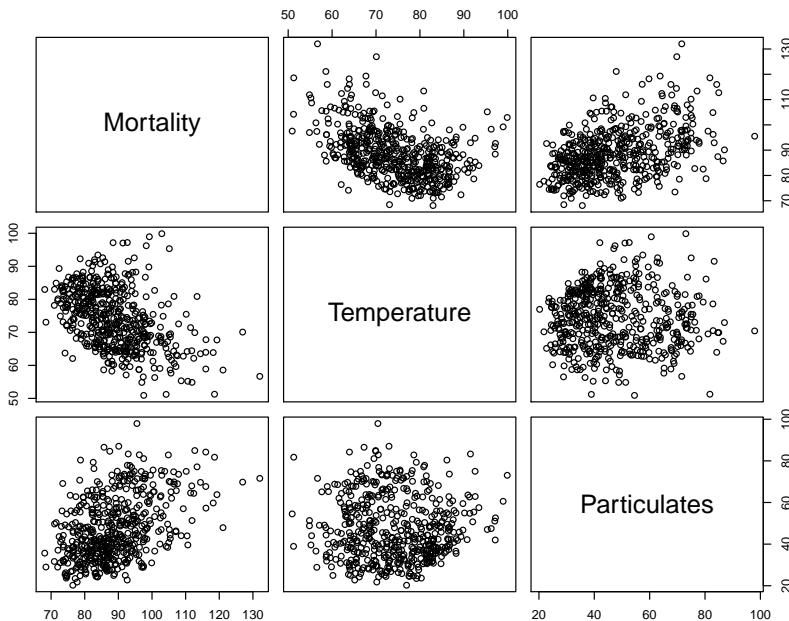


Fig. 2.3. Scatterplot matrix showing relations between mortality, temperature, and pollution.

$$F_{3,503} = \frac{(40,020 - 20,508)/3}{20,508/503} = 160,$$

which exceeds $F_{3,503}(.001) = 5.51$. We obtain the best prediction model,

$$\begin{aligned}\hat{M}_t &= 2831.5 - 1.396_{(.10)} t - .472_{(.032)}(T_t - 74.26) \\ &\quad + .023_{(.003)}(T_t - 74.26)^2 + .255_{(.019)}P_t,\end{aligned}$$

for mortality, where the standard errors, computed from (2.6)–(2.8), are given in parentheses. As expected, a negative trend is present in time as well as a negative coefficient for adjusted temperature. The quadratic effect of temperature can clearly be seen in the scatterplots of Figure 2.3. Pollution weights positively and can be interpreted as the incremental contribution to daily deaths per unit of particulate pollution. It would still be essential to check the residuals $\hat{w}_t = M_t - \hat{M}_t$ for autocorrelation (of which there is a substantial amount), but we defer this question to Section 3.8 when we discuss regression with correlated errors.

Below is the R code to plot the series, display the scatterplot matrix, fit the final regression model (2.21), and compute the corresponding values of AIC, AICc and BIC.^{2.2} Finally, the use of `na.action` in `lm()` is to retain the time series attributes for the residuals and fitted values.

^{2.2} The easiest way to extract AIC and BIC from an `lm()` run in R is to use the command `AIC()` or `BIC()`. Our definitions differ from R by terms that do not change from model to model. In the example, we show how to obtain (2.15) and (2.17) from the R output. It is more difficult to obtain AICc.

```

par(mfrow=c(3,1)) # plot the data
plot(cmort, main="Cardiovascular Mortality", xlab="", ylab="")
plot(temp, main="Temperature", xlab="", ylab="")
plot(part, main="Particulates", xlab="", ylab="")
dev.new()          # open a new graphic device
ts.plot(cmort,temp,part, col=1:3) # all on same plot (not shown)
dev.new()
pairs(cbind(Mortality=cmort, Temperature=temp, Particulates=part))
temp = temp-mean(temp) # center temperature
temp2 = temp^2
trend = time(cmort)    # time
fit = lm(cmort~ trend + temp + temp2 + part, na.action=NULL)
summary(fit)           # regression results
summary(aov(fit))      # ANOVA table (compare to next line)
summary(aov(lm(cmort~cbind(trend, temp, temp2, part)))) # Table 2.1
num = length(cmort)    # sample size
AIC(fit)/num - log(2*pi) # AIC
BIC(fit)/num - log(2*pi) # BIC
(AICC = log(sum(resid(fit)^2)/num) + (num+5)/(num-5-2)) # AICC

```

As previously mentioned, it is possible to include lagged variables in time series regression models and we will continue to discuss this type of problem throughout the text. This concept is explored further in [Problem 2.2](#) and [Problem 2.10](#). The following is a simple example of lagged regression.

Example 2.3 Regression With Lagged Variables

In [Example 1.28](#), we discovered that the Southern Oscillation Index (SOI) measured at time $t - 6$ months is associated with the Recruitment series at time t , indicating that the SOI leads the Recruitment series by six months. Although there is evidence that the relationship is not linear (this is discussed further in [Example 2.8](#) and [Example 2.9](#)), consider the following regression,

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t, \quad (2.22)$$

where R_t denotes Recruitment for month t and S_{t-6} denotes SOI six months prior. Assuming the w_t sequence is white, the fitted model is

$$\hat{R}_t = 65.79 - 44.28_{(2.78)} S_{t-6} \quad (2.23)$$

with $\hat{\sigma}_w = 22.5$ on 445 degrees of freedom. This result indicates the strong predictive ability of SOI for Recruitment six months in advance. Of course, it is still essential to check the model assumptions, but again we defer this until later.

Performing lagged regression in R is a little difficult because the series must be aligned prior to running the regression. The easiest way to do this is to create a data frame (that we call `fish`) using `ts.intersect`, which aligns the lagged series.

```

fish = ts.intersect(rec, soiL6=lag(soi, -6), dframe=TRUE)
summary(fit1 <- lm(rec~soiL6, data=fish, na.action=NULL))

```

The headache of aligning the lagged series can be avoided by using the R package `dynlm`, which must be downloaded and installed.

```

library(dynlm)
summary(fit2 <- dynlm(rec~ L(soi, 6)))

```

We note that `fit2` is similar to the `fit1` object, but the time series attributes are retained without any additional commands.

2.2 Exploratory Data Analysis

In general, it is necessary for time series data to be stationary so that averaging lagged products over time, as in the previous section, will be a sensible thing to do. With time series data, it is the dependence between the values of the series that is important to measure; we must, at least, be able to estimate autocorrelations with precision. It would be difficult to measure that dependence if the dependence structure is not regular or is changing at every time point. Hence, to achieve any meaningful statistical analysis of time series data, it will be crucial that, if nothing else, the mean and the autocovariance functions satisfy the conditions of stationarity (for at least some reasonable stretch of time) stated in [Definition 1.7](#). Often, this is not the case, and we will mention some methods in this section for playing down the effects of nonstationarity so the stationary properties of the series may be studied.

A number of our examples came from clearly nonstationary series. The Johnson & Johnson series in [Figure 1.1](#) has a mean that increases exponentially over time, and the increase in the magnitude of the fluctuations around this trend causes changes in the covariance function; the variance of the process, for example, clearly increases as one progresses over the length of the series. Also, the global temperature series shown in [Figure 1.2](#) contains some evidence of a trend over time; human-induced global warming advocates seize on this as empirical evidence to advance the hypothesis that temperatures are increasing.

Perhaps the easiest form of nonstationarity to work with is the *trend stationary* model wherein the process has stationary behavior around a trend. We may write this type of model as

$$x_t = \mu_t + y_t \quad (2.24)$$

where x_t are the observations, μ_t denotes the trend, and y_t is a stationary process. Quite often, strong trend will obscure the behavior of the stationary process, y_t , as we shall see in numerous examples. Hence, there is some advantage to removing the trend as a first step in an exploratory analysis of such time series. The steps involved are to obtain a reasonable estimate of the trend component, say $\hat{\mu}_t$, and then work with the residuals

$$\hat{y}_t = x_t - \hat{\mu}_t. \quad (2.25)$$

Example 2.4 Detrending Chicken Prices

Here we suppose the model is of the form of (2.24),

$$x_t = \mu_t + y_t,$$

where, as we suggested in the analysis of the chicken price data presented in [Example 2.1](#), a straight line might be useful for detrending the data; i.e.,

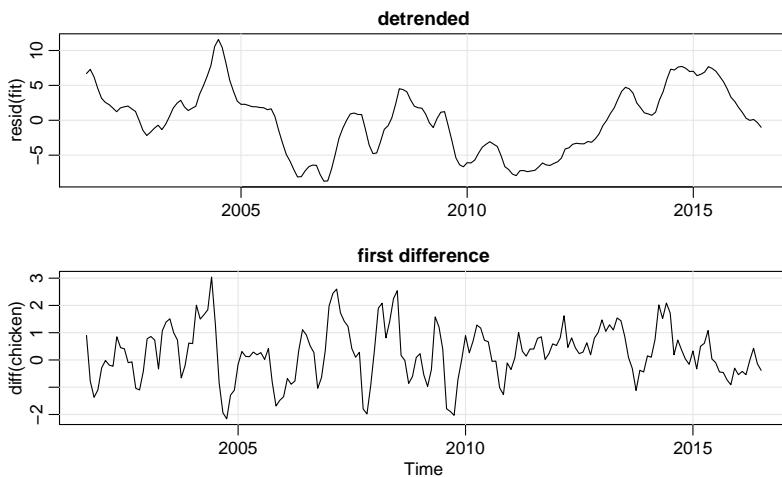


Fig. 2.4. Detrended (top) and differenced (bottom) chicken price series. The original data are shown in [Figure 2.1](#).

$$\mu_t = \beta_0 + \beta_1 t.$$

In that example, we estimated the trend using ordinary least squares and found

$$\hat{\mu}_t = -7131 + 3.59 t$$

where we are using t instead of z_t for time. [Figure 2.1](#) shows the data with the estimated trend line superimposed. To obtain the detrended series we simply subtract $\hat{\mu}_t$ from the observations, x_t , to obtain the detrended series^{2.3}

$$\hat{y}_t = x_t + 7131 - 3.59 t.$$

The top graph of [Figure 2.4](#) shows the detrended series. [Figure 2.5](#) shows the ACF of the original data (top panel) as well as the ACF of the detrended data (middle panel).

In [Example 1.11](#) and the corresponding [Figure 1.10](#) we saw that a random walk might also be a good model for trend. That is, rather than modeling trend as fixed (as in [Example 2.4](#)), we might model trend as a stochastic component using the random walk with drift model,

$$\mu_t = \delta + \mu_{t-1} + w_t, \quad (2.26)$$

where w_t is white noise and is independent of y_t . If the appropriate model is (2.24), then *differencing* the data, x_t , yields a stationary process; that is,

^{2.3} Because the error term, y_t , is not assumed to be iid, the reader may feel that weighted least squares is called for in this case. The problem is, we do not know the behavior of y_t and that is precisely what we are trying to assess at this stage. A notable result by Grenander and Rosenblatt (1957, Ch 7), however, is that under mild conditions on y_t , for polynomial regression or periodic regression, asymptotically, ordinary least squares is equivalent to weighted least squares with regard to efficiency.

$$\begin{aligned}x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) \\&= \delta + w_t + y_t - y_{t-1}.\end{aligned}\tag{2.27}$$

It is easy to show $z_t = y_t - y_{t-1}$ is stationary using [Property 1.1](#). That is, because y_t is stationary,

$$\begin{aligned}\gamma_z(h) &= \text{cov}(z_{t+h}, z_t) = \text{cov}(y_{t+h} - y_{t+h-1}, y_t - y_{t-1}) \\&= 2\gamma_y(h) - \gamma_y(h+1) - \gamma_y(h-1)\end{aligned}$$

is independent of time; we leave it as an exercise ([Problem 2.7](#)) to show that $x_t - x_{t-1}$ in [\(2.27\)](#) is stationary.

One advantage of differencing over detrending to remove trend is that no parameters are estimated in the differencing operation. One disadvantage, however, is that differencing does not yield an estimate of the stationary process y_t as can be seen in [\(2.27\)](#). If an estimate of y_t is essential, then detrending may be more appropriate. If the goal is to coerce the data to stationarity, then differencing may be more appropriate. Differencing is also a viable tool if the trend is fixed, as in [Example 2.4](#). That is, e.g., if $\mu_t = \beta_0 + \beta_1 t$ in the model [\(2.24\)](#), differencing the data produces stationarity (see [Problem 2.6](#)):

$$x_t - x_{t-1} = (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) = \beta_1 + y_t - y_{t-1}.$$

Because differencing plays a central role in time series analysis, it receives its own notation. The first difference is denoted as

$$\nabla x_t = x_t - x_{t-1}.\tag{2.28}$$

As we have seen, the first difference eliminates a linear trend. A second difference, that is, the difference of [\(2.28\)](#), can eliminate a quadratic trend, and so on. In order to define higher differences, we need a variation in notation that we will use often in our discussion of ARIMA models in [Chapter 3](#).

Definition 2.4 We define the **backshift operator** by

$$Bx_t = x_{t-1}$$

and extend it to powers $B^2 x_t = B(Bx_t) = Bx_{t-1} = x_{t-2}$, and so on. Thus,

$$B^k x_t = x_{t-k}.\tag{2.29}$$

The idea of an inverse operator can also be given if we require $B^{-1}B = 1$, so that

$$x_t = B^{-1}Bx_t = B^{-1}x_{t-1}.$$

That is, B^{-1} is the *forward-shift operator*. In addition, it is clear that we may rewrite [\(2.28\)](#) as

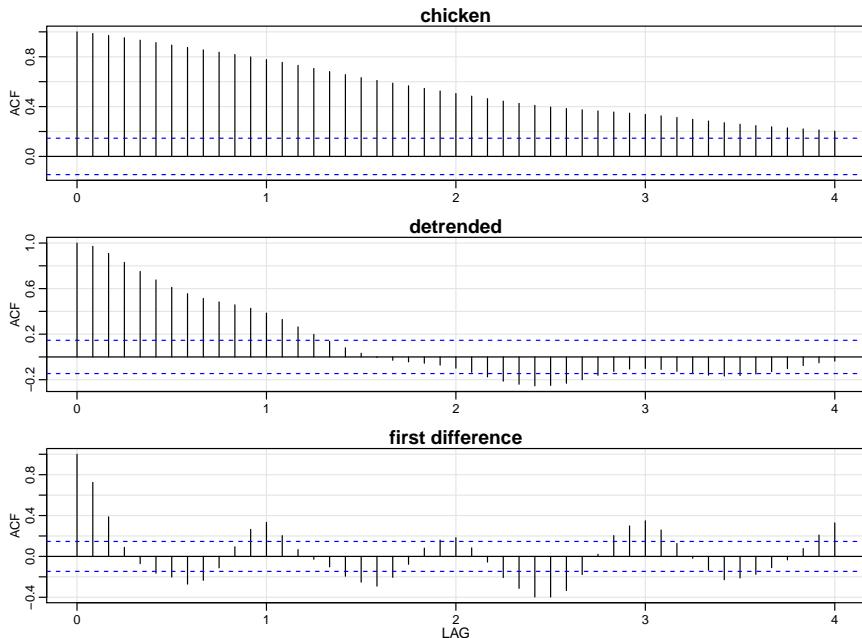


Fig. 2.5. Sample ACFs of chicken prices (top), and of the detrended (middle) and the differenced (bottom) series. Compare the top plot with the sample ACF of a straight line: `acf(1:100)`.

$$\nabla x_t = (1 - B)x_t, \quad (2.30)$$

and we may extend the notion further. For example, the second difference becomes

$$\nabla^2 x_t = (1 - B)^2 x_t = (1 - 2B + B^2)x_t = x_t - 2x_{t-1} + x_{t-2} \quad (2.31)$$

by the linearity of the operator. To check, just take the difference of the first difference $\nabla(\nabla x_t) = \nabla(x_t - x_{t-1}) = (x_t - x_{t-1}) - (x_{t-1} - x_{t-2})$.

Definition 2.5 Differences of order d are defined as

$$\nabla^d = (1 - B)^d, \quad (2.32)$$

where we may expand the operator $(1 - B)^d$ algebraically to evaluate for higher integer values of d . When $d = 1$, we drop it from the notation.

The first difference (2.28) is an example of a *linear filter* applied to eliminate a trend. Other filters, formed by averaging values near x_t , can produce adjusted series that eliminate other kinds of unwanted fluctuations, as in Chapter 4. The differencing technique is an important component of the ARIMA model of Box and Jenkins (1970) (see also Box et al., 1994), to be discussed in Chapter 3.

Example 2.5 Differencing Chicken Prices

The first difference of the chicken prices series, also shown in [Figure 2.4](#), produces different results than removing trend by detrending via regression. For example, the differenced series does not contain the long (five-year) cycle we observe in the detrended series. The ACF of this series is also shown in [Figure 2.5](#). In this case, the differenced series exhibits an annual cycle that was obscured in the original or detrended data.

The R code to reproduce [Figure 2.4](#) and [Figure 2.5](#) is as follows.

```
fit = lm(chicken~time(chicken), na.action=NULL) # regress chicken on time
par(mfrow=c(2,1))
plot(resid(fit), type="o", main="detrended")
plot(diff(chicken), type="o", main="first difference")
par(mfrow=c(3,1)) # plot ACFs
acf(chicken, 48, main="chicken")
acf(resid(fit), 48, main="detrended")
acf(diff(chicken), 48, main="first difference")
```

Example 2.6 Differencing Global Temperature

The global temperature series shown in [Figure 1.2](#) appears to behave more as a random walk than a trend stationary series. Hence, rather than detrend the data, it would be more appropriate to use differencing to coerce it into stationarity. The detreded data are shown in [Figure 2.6](#) along with the corresponding sample ACF. In this case it appears that the differenced process shows minimal autocorrelation, which may imply the global temperature series is nearly a random walk with drift. It is interesting to note that if the series is a random walk with drift, the mean of the differenced series, which is an estimate of the drift, is about .008, or an increase of about one degree centigrade per 100 years.

The R code to reproduce [Figure 2.4](#) and [Figure 2.5](#) is as follows.

```
par(mfrow=c(2,1))
plot(diff(globtemp), type="o")
mean(diff(globtemp)) # drift estimate = .008
acf(diff(gtemp), 48)
```

An alternative to differencing is a less-severe operation that still assumes stationarity of the underlying time series. This alternative, called *fractional differencing*, extends the notion of the difference operator (2.32) to fractional powers $-.5 < d < .5$, which still define stationary processes. Granger and Joyeux (1980) and Hosking (1981) introduced long memory time series, which corresponds to the case when $0 < d < .5$. This model is often used for environmental time series arising in hydrology. We will discuss long memory processes in more detail in [Section 5.1](#).

Often, obvious aberrations are present that can contribute nonstationary as well as nonlinear behavior in observed time series. In such cases, *transformations* may be useful to equalize the variability over the length of a single series. A particularly useful transformation is

$$y_t = \log x_t, \quad (2.33)$$

which tends to suppress larger fluctuations that occur over portions of the series where the underlying values are larger. Other possibilities are *power transformations* in the

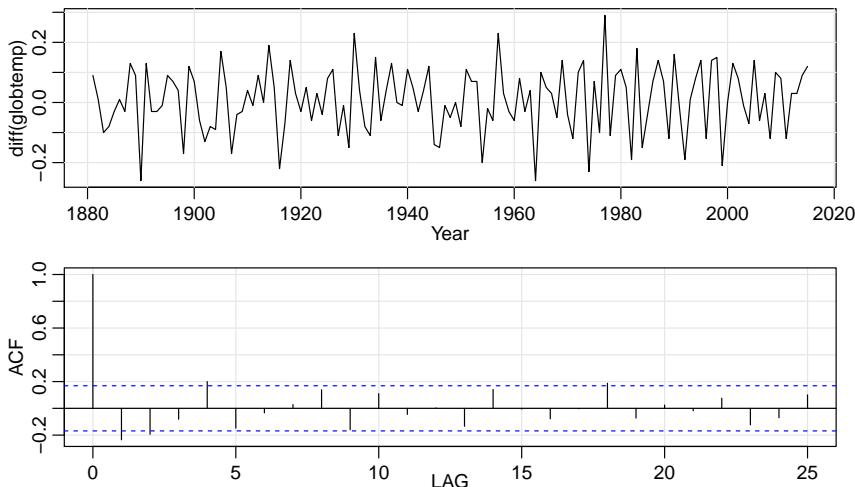


Fig. 2.6. Differenced global temperature series and its sample ACF.

Box–Cox family of the form

$$y_t = \begin{cases} (x_t^\lambda - 1)/\lambda & \lambda \neq 0, \\ \log x_t & \lambda = 0. \end{cases} \quad (2.34)$$

Methods for choosing the power λ are available (see Johnson and Wichern, 1992, §4.7) but we do not pursue them here. Often, transformations are also used to improve the approximation to normality or to improve linearity in predicting the value of one series from another.

Example 2.7 Paleoclimatic Glacial Varves

Melting glaciers deposit yearly layers of sand and silt during the spring melting seasons, which can be reconstructed yearly over a period ranging from the time deglaciation began in New England (about 12,600 years ago) to the time it ended (about 6,000 years ago). Such sedimentary deposits, called *varves*, can be used as proxies for paleoclimatic parameters, such as temperature, because, in a warm year, more sand and silt are deposited from the receding glacier. Figure 2.7 shows the thicknesses of the yearly varves collected from one location in Massachusetts for 634 years, beginning 11,834 years ago. For further information, see Shumway and Verosub (1992). Because the variation in thicknesses increases in proportion to the amount deposited, a logarithmic transformation could remove the nonstationarity observable in the variance as a function of time. Figure 2.7 shows the original and transformed varves, and it is clear that this improvement has occurred. We may also plot the histogram of the original and transformed data, as in Problem 2.8, to argue that the approximation to normality is improved. The ordinary first differences (2.30) are also computed in Problem 2.8, and we note that the first differences have

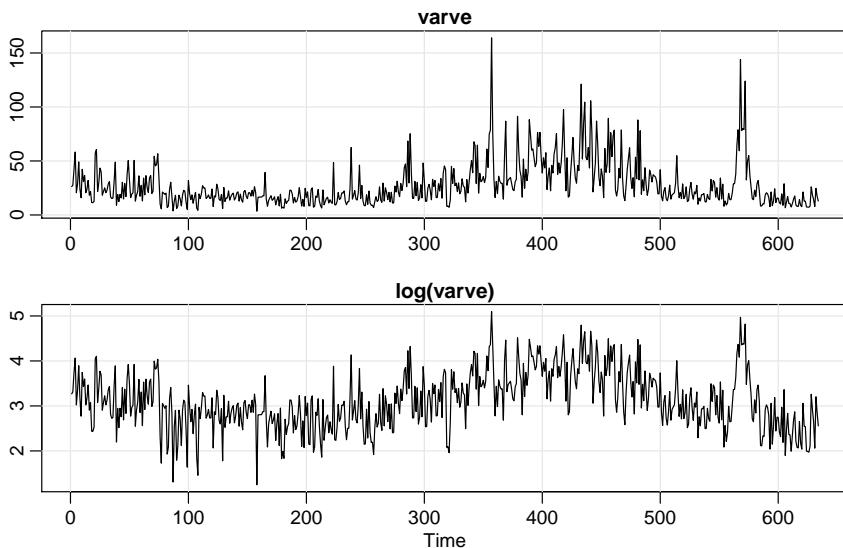


Fig. 2.7. Glacial varve thicknesses (top) from Massachusetts for $n = 634$ years compared with log transformed thicknesses (bottom).

a significant negative correlation at lag $h = 1$. Later, in [Chapter 5](#), we will show that perhaps the varve series has long memory and will propose using fractional differencing. [Figure 2.7](#) was generated in R as follows:

```
par(mfrow=c(2,1))
plot(varve, main="varve", ylab="")
plot(log(varve), main="log(varve)", ylab="" )
```

Next, we consider another preliminary data processing technique that is used for the purpose of visualizing the relations between series at different lags, namely, *scatterplot matrices*. In the definition of the ACF, we are essentially interested in relations between x_t and x_{t-h} ; the autocorrelation function tells us whether a substantial linear relation exists between the series and its own lagged values. The ACF gives a profile of the linear correlation at all possible lags and shows which values of h lead to the best predictability. The restriction of this idea to linear predictability, however, may mask a possible nonlinear relation between current values, x_t , and past values, x_{t-h} . This idea extends to two series where one may be interested in examining scatterplots of y_t versus x_{t-h} .

Example 2.8 Scatterplot Matrices, SOI and Recruitment

To check for nonlinear relations of this form, it is convenient to display a lagged scatterplot matrix, as in [Figure 2.8](#), that displays values of the SOI, S_t , on the vertical axis plotted against S_{t-h} on the horizontal axis. The sample autocorrelations are displayed in the upper right-hand corner and superimposed on the scatterplots are locally weighted scatterplot smoothing (lowess) lines that can be used to help

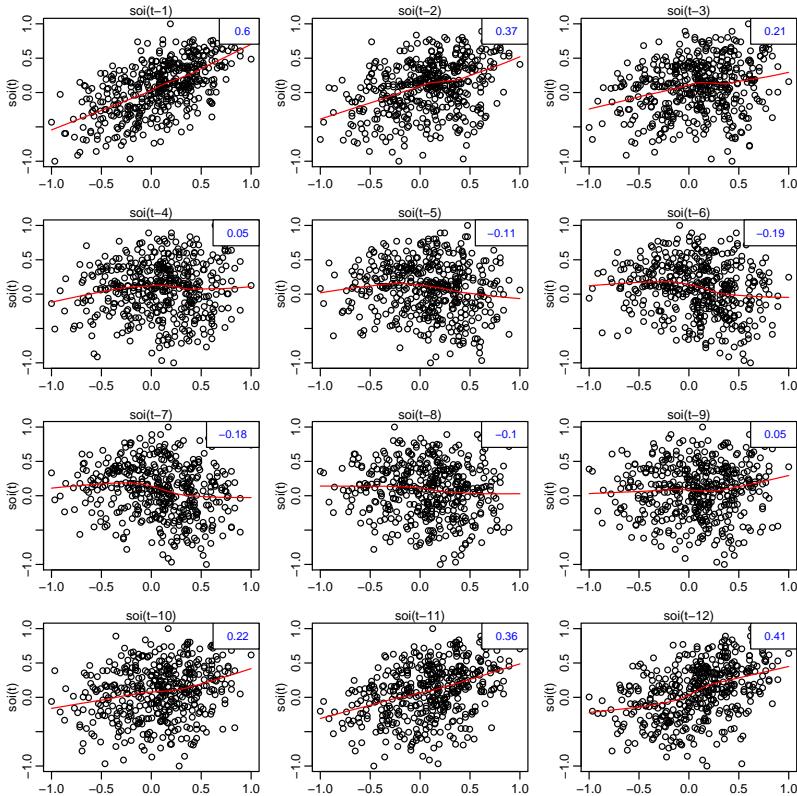


Fig. 2.8. Scatterplot matrix relating current SOI values, S_t , to past SOI values, S_{t-h} , at lags $h = 1, 2, \dots, 12$. The values in the upper right corner are the sample autocorrelations and the lines are a lowess fit.

discover any nonlinearities. We discuss smoothing in the next section, but for now, think of lowess as a robust method for fitting local regression.

In Figure 2.8, we notice that the lowess fits are approximately linear, so that the sample autocorrelations are meaningful. Also, we see strong positive linear relations at lags $h = 1, 2, 11, 12$, that is, between S_t and $S_{t-1}, S_{t-2}, S_{t-11}, S_{t-12}$, and a negative linear relation at lags $h = 6, 7$. These results match up well with peaks noticed in the ACF in Figure 1.16.

Similarly, we might want to look at values of one series, say Recruitment, denoted R_t plotted against another series at various lags, say the SOI, S_{t-h} , to look for possible nonlinear relations between the two series. Because, for example, we might wish to predict the Recruitment series, R_t , from current or past values of the SOI series, S_{t-h} , for $h = 0, 1, 2, \dots$ it would be worthwhile to examine the scatterplot matrix. Figure 2.9 shows the lagged scatterplot of the Recruitment series R_t on the

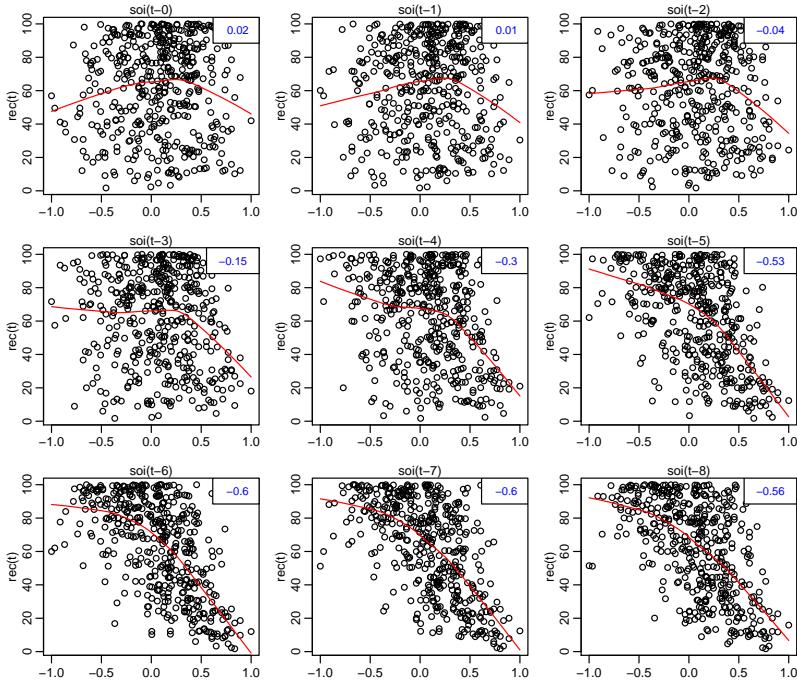


Fig. 2.9. Scatterplot matrix of the Recruitment series, R_t , on the vertical axis plotted against the SOI series, S_{t-h} , on the horizontal axis at lags $h = 0, 1, \dots, 8$. The values in the upper right corner are the sample cross-correlations and the lines are a lowess fit.

vertical axis plotted against the SOI index S_{t-h} on the horizontal axis. In addition, the figure exhibits the sample cross-correlations as well as lowess fits.

Figure 2.9 shows a fairly strong nonlinear relationship between Recruitment, R_t , and the SOI series at $S_{t-5}, S_{t-6}, S_{t-7}, S_{t-8}$, indicating the SOI series tends to lead the Recruitment series and the coefficients are negative, implying that increases in the SOI lead to decreases in the Recruitment. The nonlinearity observed in the scatterplots (with the help of the superimposed lowess fits) indicates that the behavior between Recruitment and the SOI is different for positive values of SOI than for negative values of SOI.

Simple scatterplot matrices for one series can be obtained in R using the `lag.plot` command. Figure 2.8 and Figure 2.9 may be reproduced using the following scripts provided with `astsa`:

```
lag1.plot(soi, 12)      # Figure 2.8
lag2.plot(soi, rec, 8)  # Figure 2.9
```

Example 2.9 Regression with Lagged Variables (cont)

In Example 2.3 we regressed Recruitment on lagged SOI,

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t.$$

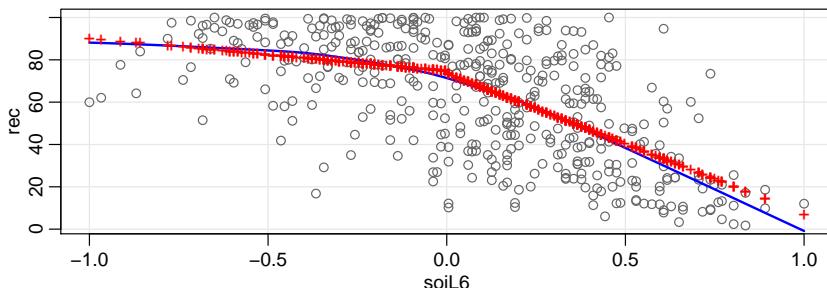


Fig. 2.10. Display for [Example 2.9](#): Plot of Recruitment (R_t) vs SOI lagged 6 months (S_{t-6}) with the fitted values of the regression as points (+) and a lowess fit (—).

However, in [Example 2.8](#), we saw that the relationship is nonlinear and different when SOI is positive or negative. In this case, we may consider adding a dummy variable to account for this change. In particular, we fit the model

$$R_t = \beta_0 + \beta_1 S_{t-6} + \beta_2 D_{t-6} + \beta_3 D_{t-6} S_{t-6} + w_t,$$

where D_t is a dummy variable that is 0 if $S_t < 0$ and 1 otherwise. This means that

$$R_t = \begin{cases} \beta_0 + \beta_1 S_{t-6} + w_t & \text{if } S_{t-6} < 0, \\ (\beta_0 + \beta_2) + (\beta_1 + \beta_3)S_{t-6} + w_t & \text{if } S_{t-6} \geq 0. \end{cases}$$

The result of the fit is given in the R code below. [Figure 2.10](#) shows R_t vs S_{t-6} with the fitted values of the regression and a lowess fit superimposed. The piecewise regression fit is similar to the lowess fit, but we note that the residuals are not white noise (see the code below). This is followed up in [Example 3.45](#).

```
dummy = ifelse(soi<0, 0, 1)
fish = ts.intersect(rec, soiL6=lag(soi, -6), dL6=lag(dummy, -6), dframe=TRUE)
summary(fit <- lm(rec~soiL6*dL6, data=fish, na.action=NULL))

Coefficients:
            Estimate Std. Error t.value
(Intercept) 74.479     2.865  25.998
soiL6       -15.358    7.401  -2.075
dL6         -1.139    3.711  -0.307
soiL6:dL6   -51.244    9.523  -5.381
---
Residual standard error: 21.84 on 443 degrees of freedom
Multiple R-squared:  0.4024
F-statistic: 99.43 on 3 and 443 DF

attach(fish)
plot(soiL6, rec)
lines(lowess(soiL6, rec), col=4, lwd=2)
points(soiL6, fitted(fit), pch='+', col=2)
plot(resid(fit)) # not shown ...
acf(resid(fit)) # ... but obviously not noise
```

As a final exploratory tool, we discuss assessing periodic behavior in time series data using regression analysis. In [Example 1.12](#), we briefly discussed the problem of

identifying cyclic or periodic signals in time series. A number of the time series we have seen so far exhibit periodic behavior. For example, the data from the pollution study example shown in [Figure 2.2](#) exhibit strong yearly cycles. The Johnson & Johnson data shown in [Figure 1.1](#) make one cycle every year (four quarters) on top of an increasing trend and the speech data in [Figure 1.2](#) is highly repetitive. The monthly SOI and Recruitment series in [Figure 1.6](#) show strong yearly cycles, which obscures the slower El Niño cycle.

Example 2.10 Using Regression to Discover a Signal in Noise

In [Example 1.12](#), we generated $n = 500$ observations from the model

$$x_t = A \cos(2\pi\omega t + \phi) + w_t, \quad (2.35)$$

where $\omega = 1/50$, $A = 2$, $\phi = .6\pi$, and $\sigma_w = 5$; the data are shown on the bottom panel of [Figure 1.11](#). At this point we assume the frequency of oscillation $\omega = 1/50$ is known, but A and ϕ are unknown parameters. In this case the parameters appear in (2.35) in a nonlinear way, so we use a trigonometric identity^{2.4} and write

$$A \cos(2\pi\omega t + \phi) = \beta_1 \cos(2\pi\omega t) + \beta_2 \sin(2\pi\omega t),$$

where $\beta_1 = A \cos(\phi)$ and $\beta_2 = -A \sin(\phi)$. Now the model (2.35) can be written in the usual linear regression form given by (no intercept term is needed here)

$$x_t = \beta_1 \cos(2\pi t/50) + \beta_2 \sin(2\pi t/50) + w_t. \quad (2.36)$$

Using linear regression, we find $\hat{\beta}_1 = -.74_{(.33)}$, $\hat{\beta}_2 = -1.99_{(.33)}$ with $\hat{\sigma}_w = 5.18$; the values in parentheses are the standard errors. We note the actual values of the coefficients for this example are $\beta_1 = 2 \cos(.6\pi) = -.62$, and $\beta_2 = -2 \sin(.6\pi) = -1.90$. It is clear that we are able to detect the signal in the noise using regression, even though the signal-to-noise ratio is small. [Figure 2.11](#) shows data generated by (2.35) with the fitted line superimposed.

To reproduce the analysis and [Figure 2.11](#) in R, use the following:

```
set.seed(90210)          # so you can reproduce these results
x = 2*cos(2*pi*1:500/50 + .6*pi) + rnorm(500,0,5)
z1 = cos(2*pi*1:500/50)
z2 = sin(2*pi*1:500/50)
summary(fit <- lm(x~0+z1+z2)) # zero to exclude the intercept
Coefficients:
            Estimate Std. Error t value
z1   -0.7442     0.3274  -2.273
z2   -1.9949     0.3274  -6.093
Residual standard error: 5.177 on 498 degrees of freedom
par(mfrow=c(2,1))
plot.ts(x)
plot.ts(x, col=8, ylab=expression(hat(x)))
lines(fitted(fit), col=2)
```

We will discuss this and related approaches in more detail in [Chapter 4](#).

^{2.4} $\cos(\alpha \pm \beta) = \cos(\alpha) \cos(\beta) \mp \sin(\alpha) \sin(\beta)$.

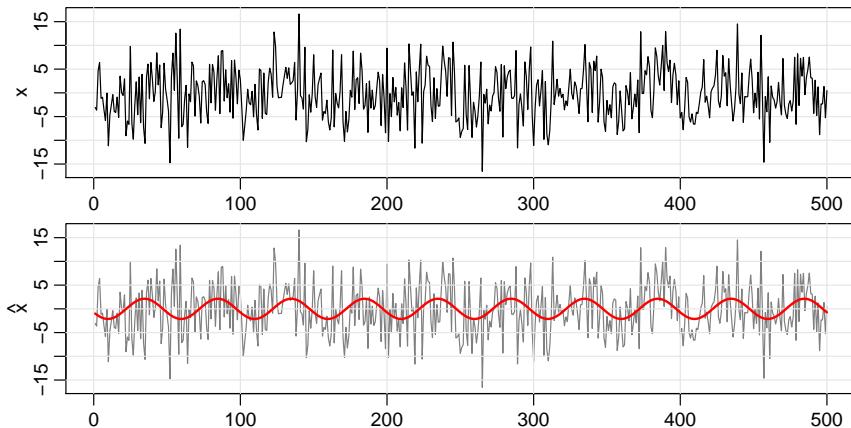


Fig. 2.11. Data generated by (2.35) [top] and the fitted line superimposed on the data [bottom].

2.3 Smoothing in the Time Series Context

In [Section 1.2](#), we introduced the concept of filtering or smoothing a time series, and in [Example 1.9](#), we discussed using a moving average to smooth white noise. This method is useful in discovering certain traits in a time series, such as long-term trend and seasonal components. In particular, if x_t represents the observations, then

$$m_t = \sum_{j=-k}^k a_j x_{t-j}, \quad (2.37)$$

where $a_j = a_{-j} \geq 0$ and $\sum_{j=-k}^k a_j = 1$ is a symmetric moving average of the data.

Example 2.11 Moving Average Smoother

For example, [Figure 2.12](#) shows the monthly SOI series discussed in [Example 1.5](#) smoothed using (2.37) with weights $a_0 = a_{\pm 1} = \dots = a_{\pm 5} = 1/12$, and $a_{\pm 6} = 1/24$; $k = 6$. This particular method removes (filters out) the obvious annual temperature cycle and helps emphasize the El Niño cycle. To reproduce [Figure 2.12](#) in R:

```
wgts = c(.5, rep(1,11), .5)/12
soif = filter(soi, sides=2, filter=wgts)
plot(soi)
lines(soif, lwd=2, col=4)
par(fig = c(.65, 1, .65, 1), new = TRUE) # the insert
nwgts = c(rep(0,20), wgts, rep(0,20))
plot(nwgts, type="l", ylim = c(-.02,.1), xaxt='n', yaxt='n', ann=FALSE)
```

Although the moving average smoother does a good job in highlighting the El Niño effect, it might be considered too choppy. We can obtain a smoother fit using the normal distribution for the weights, instead of boxcar-type weights of (2.37).

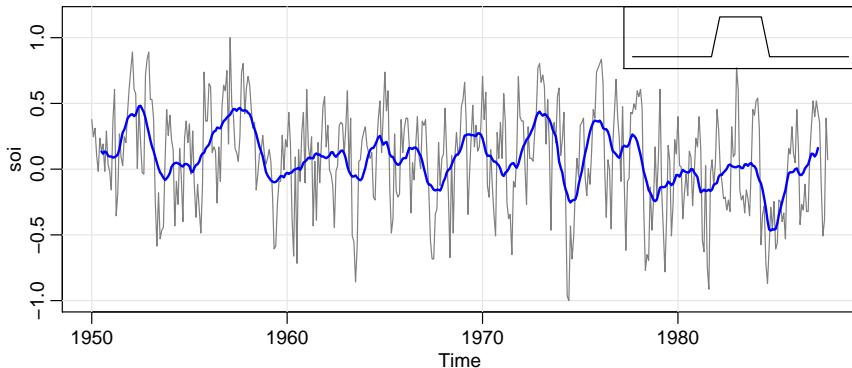


Fig. 2.12. Moving average smoother of SOI. The insert shows the shape of the moving average (“boxcar”) kernel [not drawn to scale] described in (2.39).

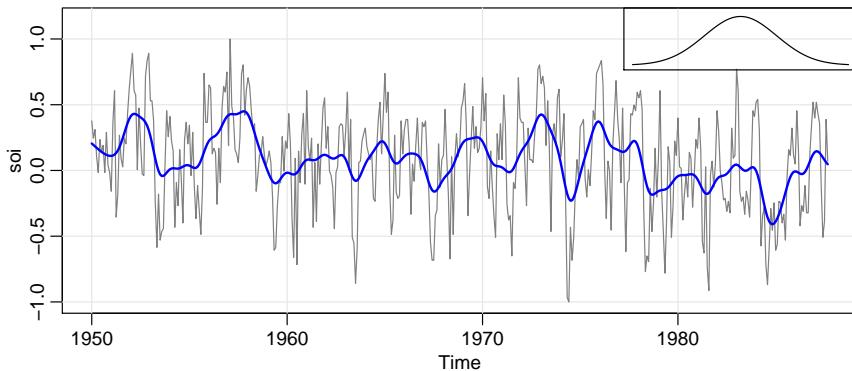


Fig. 2.13. Kernel smoother of SOI. The insert shows the shape of the normal kernel [not drawn to scale].

Example 2.12 Kernel Smoothing

Kernel smoothing is a moving average smoother that uses a weight function, or kernel, to average the observations. Figure 2.13 shows kernel smoothing of the SOI series, where m_t is now

$$m_t = \sum_{i=1}^n w_i(t) x_i, \quad (2.38)$$

where

$$w_i(t) = K\left(\frac{t-i}{b}\right) / \sum_{j=1}^n K\left(\frac{t-j}{b}\right) \quad (2.39)$$

are the weights and $K(\cdot)$ is a kernel function. This estimator, which was originally explored by Parzen (1962) and Rosenblatt (1956b), is often called the Nadaraya–Watson estimator (Watson, 1966). In this example, and typically, the normal kernel, $K(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)$, is used.

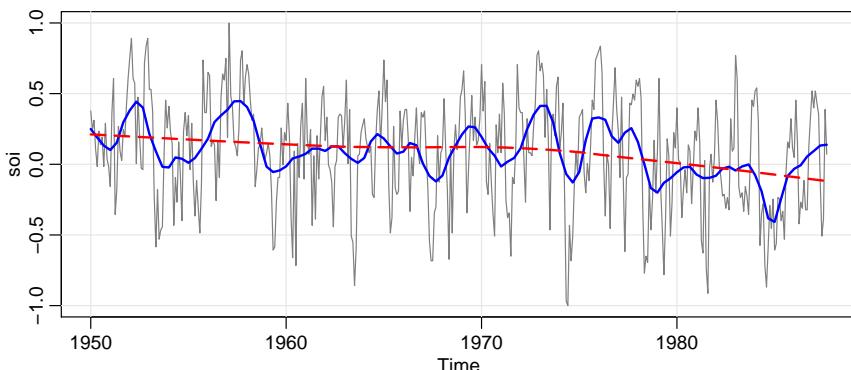


Fig. 2.14. Locally weighted scatterplot smoothers (`lowess`) of the SOI series.

To implement this in R, use the `ksmooth` function where a bandwidth can be chosen. The wider the bandwidth, b , the smoother the result. From the R `ksmooth` help file: The kernels are scaled so that their quartiles (viewed as probability densities) are at $\pm 0.25 \times \text{bandwidth}$. For the standard normal distribution, the quartiles are ± 0.674 . In our case, we are smoothing over time, which is of the form $t/12$ for the SOI time series. In Figure 2.13, we used the value of $b = 1$ to correspond to approximately smoothing a little over one year. Figure 2.13 can be reproduced in R as follows.

```
plot(soi)
lines(ksmooth(time(soi), soi, "normal", bandwidth=1), lwd=2, col=4)
par(fig = c(.65, 1, .65, 1), new = TRUE) # the insert
gauss = function(x) { 1/sqrt(2*pi) * exp(-(x^2)/2) }
x = seq(from = -3, to = 3, by = 0.001)
plot(x, gauss(x), type = "l", ylim=c(-.02,.45), xaxt='n', yaxt='n', ann=FALSE)
```

Example 2.13 Lowess

Another approach to smoothing a time plot is nearest neighbor regression. The technique is based on k -nearest neighbors regression, wherein one uses only the data $\{x_{t-k/2}, \dots, x_t, \dots, x_{t+k/2}\}$ to predict x_t via regression, and then sets $m_t = \hat{x}_t$.

Lowess is a method of smoothing that is rather complex, but the basic idea is close to nearest neighbor regression. Figure 2.14 shows smoothing of SOI using the R function `lowess` (see Cleveland, 1979). First, a certain proportion of nearest neighbors to x_t are included in a weighting scheme; values closer to x_t in time get more weight. Then, a robust weighted regression is used to predict x_t and obtain the smoothed values m_t . The larger the fraction of nearest neighbors included, the smoother the fit will be. In Figure 2.14, one smoother uses 5% of the data to obtain an estimate of the El Niño cycle of the data.

In addition, a (negative) trend in SOI would indicate the long-term warming of the Pacific Ocean. To investigate this, we used `lowess` with the default smoother span of `f=2/3` of the data. Figure 2.14 can be reproduced in R as follows.

```
plot(soi)
lines(lowess(soi, f=.05), lwd=2, col=4) # El Nino cycle
```

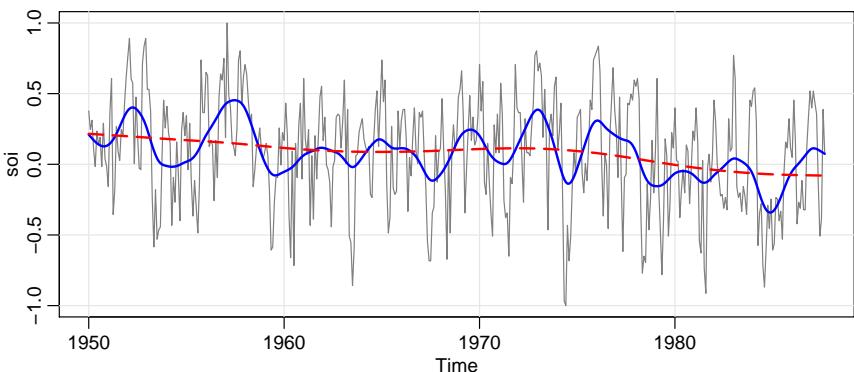


Fig. 2.15. Smoothing splines fit to the SOI series.

```
lines(lowess(soi), lty=2, lwd=2, col=2) # trend (with default span)
```

Example 2.14 Smoothing Splines

An obvious way to smooth data would be to fit a polynomial regression in terms of time. For example, a cubic polynomial would have $x_t = m_t + w_t$ where

$$m_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3.$$

We could then fit m_t via ordinary least squares.

An extension of polynomial regression is to first divide time $t = 1, \dots, n$, into k intervals, $[t_0 = 1, t_1], [t_1 + 1, t_2], \dots, [t_{k-1} + 1, t_k = n]$; the values t_0, t_1, \dots, t_k are called *knots*. Then, in each interval, one fits a polynomial regression, typically the order is 3, and this is called *cubic splines*.

A related method is *smoothing splines*, which minimizes a compromise between the fit and the degree of smoothness given by

$$\sum_{t=1}^n [x_t - m_t]^2 + \lambda \int (m_t'')^2 dt, \quad (2.40)$$

where m_t is a cubic spline with a knot at each t and primes denote differentiation. The degree of smoothness is controlled by $\lambda > 0$.

Think of taking a long drive where m_t is the position of your car at time t . In this case, m_t'' is instantaneous acceleration/deceleration, and $\int (m_t'')^2 dt$ is a measure of the total amount of acceleration and deceleration on your trip. A smooth drive would be one where a constant velocity, is maintained (i.e., $m_t'' = 0$). A choppy ride would be when the driver is constantly accelerating and decelerating, such as beginning drivers tend to do.

If $\lambda = 0$, we don't care how choppy the ride is, and this leads to $m_t = x_t$, the data, which are not smooth. If $\lambda = \infty$, we insist on no acceleration or deceleration ($m_t'' = 0$); in this case, our drive must be at constant velocity, $m_t = c + vt$, and

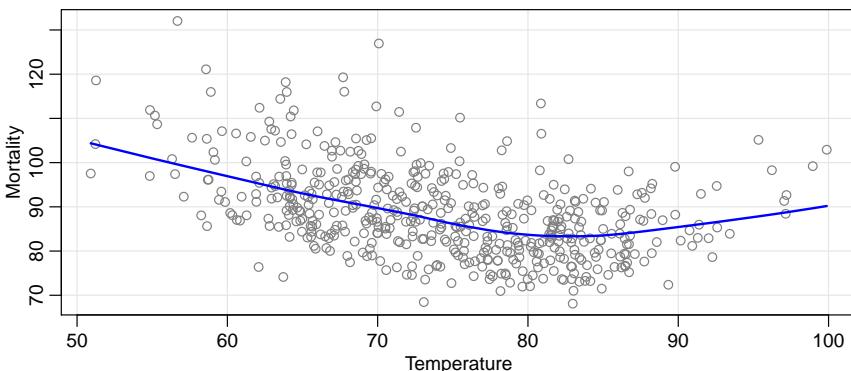


Fig. 2.16. Smooth of mortality as a function of temperature using lowess.

consequently very smooth. Thus, λ is seen as a trade-off between linear regression (completely smooth) and the data itself (no smoothness). The larger the value of λ , the smoother the fit.

In R, the smoothing parameter is called `spar` and it is monotonically related to λ ; type `?smooth.spline` to view the help file for details. [Figure 2.15](#) shows smoothing spline fits on the SOI series using `spar=.5` to emphasize the El Niño cycle, and `spar=1` to emphasize the trend. The figure can be reproduced in R as follows.

```
plot(soi)
lines(smooth.spline(time(soi), soi, spar=.5), lwd=2, col=4)
lines(smooth.spline(time(soi), soi, spar= 1), lty=2, lwd=2, col=2)
```

Example 2.15 Smoothing One Series as a Function of Another

In addition to smoothing time plots, smoothing techniques can be applied to smoothing a time series as a function of another time series. We have already seen this idea used in [Example 2.8](#) when we used lowess to visualize the nonlinear relationship between Recruitment and SOI at various lags. In this example, we smooth the scatterplot of two contemporaneously measured time series, mortality as a function of temperature. In [Example 2.2](#), we discovered a nonlinear relationship between mortality and temperature. Continuing along these lines, [Figure 2.16](#) show a scatterplot of mortality, M_t , and temperature, T_t , along with M_t smoothed as a function of T_t using lowess. Note that mortality increases at extreme temperatures, but in an asymmetric way; mortality is higher at colder temperatures than at hotter temperatures. The minimum mortality rate seems to occur at approximately 83° F.

[Figure 2.16](#) can be reproduced in R as follows using the defaults.

```
plot(temp, cmort, xlab="Temperature", ylab="Mortality")
lines(lowess(temp, cmort))
```

Problems

Section 2.1

2.1 A Structural Model For the Johnson & Johnson data, say y_t , shown in Figure 1.1, let $x_t = \log(y_t)$. In this problem, we are going to fit a special type of structural model, $x_t = T_t + S_t + N_t$ where T_t is a trend component, S_t is a seasonal component, and N_t is noise. In our case, time t is in quarters (1960.00, 1960.25, ...) so one unit of time is a year.

- (a) Fit the regression model

$$x_t = \underbrace{\beta t}_{\text{trend}} + \underbrace{\alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t)}_{\text{seasonal}} + \underbrace{w_t}_{\text{noise}}$$

where $Q_i(t) = 1$ if time t corresponds to quarter $i = 1, 2, 3, 4$, and zero otherwise. The $Q_i(t)$'s are called indicator variables. We will assume for now that w_t is a Gaussian white noise sequence. *Hint:* Detailed code is given in Code R.4, the last example of Section R.4.

- (b) If the model is correct, what is the estimated average annual increase in the logged earnings per share?
- (c) If the model is correct, does the average logged earnings rate increase or decrease from the third quarter to the fourth quarter? And, by what percentage does it increase or decrease?
- (d) What happens if you include an intercept term in the model in (a)? Explain why there was a problem.
- (e) Graph the data, x_t , and superimpose the fitted values, say \hat{x}_t , on the graph. Examine the residuals, $x_t - \hat{x}_t$, and state your conclusions. Does it appear that the model fits the data well (do the residuals look white)?

2.2 For the mortality data examined in Example 2.2:

- (a) Add another component to the regression in (2.21) that accounts for the particulate count four weeks prior; that is, add P_{t-4} to the regression in (2.21). State your conclusion.
- (b) Draw a scatterplot matrix of M_t, T_t, P_t and P_{t-4} and then calculate the pairwise correlations between the series. Compare the relationship between M_t and P_t versus M_t and P_{t-4} .

2.3 In this problem, we explore the difference between a random walk and a trend stationary process.

- (a) Generate *four* series that are random walk with drift, (1.4), of length $n = 100$ with $\delta = .01$ and $\sigma_w = 1$. Call the data x_t for $t = 1, \dots, 100$. Fit the regression $x_t = \beta t + w_t$ using least squares. Plot the data, the true mean function (i.e., $\mu_t = .01 t$) and the fitted line, $\hat{x}_t = \hat{\beta} t$, on the same graph. *Hint:* The following R code may be useful.

```

par(mfrow=c(2,2), mar=c(2.5,2.5,0,0)+.5, mgp=c(1.6,.6,0)) # set up
for (i in 1:4){
  x = ts(cumsum(rnorm(100,.01,1))) # data
  regx = lm(x~0+time(x), na.action=NULL) # regression
  plot(x, ylab='Random Walk w Drift') # plots
  abline(a=0, b=.01, col=2, lty=2) # true mean (red - dashed)
  abline(regx, col=4) # fitted line (blue - solid)
}

```

- (b) Generate *four* series of length $n = 100$ that are linear trend plus noise, say $y_t = .01t + w_t$, where t and w_t are as in part (a). Fit the regression $y_t = \beta t + w_t$ using least squares. Plot the data, the true mean function (i.e., $\mu_t = .01t$) and the fitted line, $\hat{y}_t = \hat{\beta}t$, on the same graph.
- (c) Comment (what did you learn from this assignment).

2.4 Kullback-Leibler Information Given the random $n \times 1$ vector y , we define the information for discriminating between two densities in the same family, indexed by a parameter θ , say $f(y; \theta_1)$ and $f(y; \theta_2)$, as

$$I(\theta_1; \theta_2) = n^{-1} E_1 \log \frac{f(y; \theta_1)}{f(y; \theta_2)}, \quad (2.41)$$

where E_1 denotes expectation with respect to the density determined by θ_1 . For the Gaussian regression model, the parameters are $\theta = (\beta', \sigma^2)'$. Show that

$$I(\theta_1; \theta_2) = \frac{1}{2} \left(\frac{\sigma_1^2}{\sigma_2^2} - \log \frac{\sigma_1^2}{\sigma_2^2} - 1 \right) + \frac{1}{2} \frac{(\beta_1 - \beta_2)' Z' Z (\beta_1 - \beta_2)}{n \sigma_2^2}. \quad (2.42)$$

2.5 Model Selection Both selection criteria (2.15) and (2.16) are derived from information theoretic arguments, based on the well-known *Kullback-Leibler discrimination information* numbers (see Kullback and Leibler, 1951, Kullback, 1958). We give an argument due to Hurvich and Tsai (1989). We think of the measure (2.42) as measuring the discrepancy between the two densities, characterized by the parameter values $\theta'_1 = (\beta'_1, \sigma_1^2)'$ and $\theta'_2 = (\beta'_2, \sigma_2^2)'$. Now, if the true value of the parameter vector is θ_1 , we argue that the best model would be one that minimizes the discrepancy between the theoretical value and the sample, say $I(\theta_1; \hat{\theta})$. Because θ_1 will not be known, Hurvich and Tsai (1989) considered finding an unbiased estimator for $E_1[I(\beta_1, \sigma_1^2; \hat{\beta}, \hat{\sigma}^2)]$, where

$$I(\beta_1, \sigma_1^2; \hat{\beta}, \hat{\sigma}^2) = \frac{1}{2} \left(\frac{\sigma_1^2}{\hat{\sigma}^2} - \log \frac{\sigma_1^2}{\hat{\sigma}^2} - 1 \right) + \frac{1}{2} \frac{(\beta_1 - \hat{\beta})' Z' Z (\beta_1 - \hat{\beta})}{n \hat{\sigma}^2}$$

and β is a $k \times 1$ regression vector. Show that

$$E_1[I(\beta_1, \sigma_1^2; \hat{\beta}, \hat{\sigma}^2)] = \frac{1}{2} \left(-\log \sigma_1^2 + E_1 \log \hat{\sigma}^2 + \frac{n+k}{n-k-2} - 1 \right), \quad (2.43)$$

using the distributional properties of the regression coefficients and error variance. An unbiased estimator for $E_1 \log \hat{\sigma}^2$ is $\log \hat{\sigma}^2$. Hence, we have shown that the expectation

of the above discrimination information is as claimed. As models with differing dimensions k are considered, only the second and third terms in (2.43) will vary and we only need unbiased estimators for those two terms. This gives the form of AICc quoted in (2.16) in the chapter. You will need the two distributional results

$$\frac{n\hat{\sigma}^2}{\sigma_1^2} \sim \chi_{n-k}^2 \quad \text{and} \quad \frac{(\hat{\beta} - \beta_1)'Z'Z(\hat{\beta} - \beta_1)}{\sigma_1^2} \sim \chi_k^2$$

The two quantities are distributed independently as chi-squared distributions with the indicated degrees of freedom. If $x \sim \chi_n^2$, $E(1/x) = 1/(n-2)$.

Section 2.2

2.6 Consider a process consisting of a linear trend with an additive noise term consisting of independent random variables w_t with zero means and variances σ_w^2 , that is,

$$x_t = \beta_0 + \beta_1 t + w_t,$$

where β_0, β_1 are fixed constants.

- (a) Prove x_t is nonstationary.
- (b) Prove that the first difference series $\nabla x_t = x_t - x_{t-1}$ is stationary by finding its mean and autocovariance function.
- (c) Repeat part (b) if w_t is replaced by a general stationary process, say y_t , with mean function μ_y and autocovariance function $\gamma_y(h)$.

2.7 Show (2.27) is stationary.

2.8 The glacial varve record plotted in Figure 2.7 exhibits some nonstationarity that can be improved by transforming to logarithms and some additional nonstationarity that can be corrected by differencing the logarithms.

- (a) Argue that the glacial varves series, say x_t , exhibits heteroscedasticity by computing the sample variance over the first half and the second half of the data. Argue that the transformation $y_t = \log x_t$ stabilizes the variance over the series. Plot the histograms of x_t and y_t to see whether the approximation to normality is improved by transforming the data.
- (b) Plot the series y_t . Do any time intervals, of the order 100 years, exist where one can observe behavior comparable to that observed in the global temperature records in Figure 1.2?
- (c) Examine the sample ACF of y_t and comment.
- (d) Compute the difference $u_t = y_t - y_{t-1}$, examine its time plot and sample ACF, and argue that differencing the logged varve data produces a reasonably stationary series. Can you think of a practical interpretation for u_t ? Hint: Recall Footnote 1.2.

- (e) Based on the sample ACF of the differenced transformed series computed in (c), argue that a generalization of the model given by Example 1.26 might be reasonable. Assume

$$u_t = \mu + w_t + \theta w_{t-1}$$

is stationary when the inputs w_t are assumed independent with mean 0 and variance σ_w^2 . Show that

$$\gamma_u(h) = \begin{cases} \sigma_w^2(1 + \theta^2) & \text{if } h = 0, \\ \theta \sigma_w^2 & \text{if } h = \pm 1, \\ 0 & \text{if } |h| > 1. \end{cases}$$

- (f) Based on part (e), use $\hat{\rho}_u(1)$ and the estimate of the variance of u_t , $\hat{\gamma}_u(0)$, to derive estimates of θ and σ_w^2 . This is an application of the method of moments from classical statistics, where estimators of the parameters are derived by equating sample moments to theoretical moments.

2.9 In this problem, we will explore the periodic nature of S_t , the SOI series displayed in Figure 1.5.

- (a) Detrend the series by fitting a regression of S_t on time t . Is there a significant trend in the sea surface temperature? Comment.
- (b) Calculate the periodogram for the detrended series obtained in part (a). Identify the frequencies of the two main peaks (with an obvious one at the frequency of one cycle every 12 months). What is the probable El Niño cycle indicated by the minor peak?

Section 2.3

2.10 Consider the two weekly time series `oil` and `gas`. The oil series is in dollars per barrel, while the gas series is in cents per gallon.

- (a) Plot the data on the same graph. Which of the simulated series displayed in Section 1.2 do these series most resemble? Do you believe the series are stationary (explain your answer)?
- (b) In economics, it is often the percentage change in price (termed *growth rate* or *return*), rather than the absolute price change, that is important. Argue that a transformation of the form $y_t = \nabla \log x_t$ might be applied to the data, where x_t is the oil or gas price series. Hint: Recall Footnote 1.2.
- (c) Transform the data as described in part (b), plot the data on the same graph, look at the sample ACFs of the transformed data, and comment.
- (d) Plot the CCF of the transformed data and comment. The small, but significant values when `gas` leads `oil` might be considered as feedback.
- (e) Exhibit scatterplots of the oil and gas growth rate series for up to three weeks of lead time of oil prices; include a nonparametric smoother in each plot and comment on the results (e.g., Are there outliers? Are the relationships linear?).

(f) There have been a number of studies questioning whether gasoline prices respond more quickly when oil prices are rising than when oil prices are falling (“asymmetry”). We will attempt to explore this question here with simple lagged regression; we will ignore some obvious problems such as outliers and autocorrelated errors, so this will not be a definitive analysis. Let G_t and O_t denote the gas and oil growth rates.

- (i) Fit the regression (and comment on the results)

$$G_t = \alpha_1 + \alpha_2 I_t + \beta_1 O_t + \beta_2 O_{t-1} + w_t,$$

where $I_t = 1$ if $O_t \geq 0$ and 0 otherwise (I_t is the indicator of no growth or positive growth in oil price). *Hint:*

```
poil = diff(log(oil))
pgas = diff(log(gas))
indi = ifelse(poil < 0, 0, 1)
mess = ts.intersect(pgas, poil, poill = lag(poil, -1), indi)
summary(fit <- lm(pgas~ poil + poill + indi, data=mess))
```

- (ii) What is the fitted model when there is negative growth in oil price at time t ? What is the fitted model when there is no or positive growth in oil price? Do these results support the asymmetry hypothesis?
- (iii) Analyze the residuals from the fit and comment.

2.11 Use two different smoothing techniques described in Section 2.3 to estimate the trend in the global temperature series `globtemp`. Comment.

Chapter 3

ARIMA Models

Classical regression is often insufficient for explaining all of the interesting dynamics of a time series. For example, the ACF of the residuals of the simple linear regression fit to the price of chicken data (see [Example 2.4](#)) reveals additional structure in the data that regression did not capture. Instead, the introduction of correlation that may be generated through lagged linear relations leads to proposing the *autoregressive* (*AR*) and *autoregressive moving average* (*ARMA*) models that were presented in Whittle (1951). Adding nonstationary models to the mix leads to the *autoregressive integrated moving average* (*ARIMA*) model popularized in the landmark work by Box and Jenkins (1970). The *Box–Jenkins method* for identifying ARIMA models is given in this chapter along with techniques for *parameter estimation* and *forecasting* for these models. A partial theoretical justification of the use of ARMA models is discussed in [Section B.4](#).

3.1 Autoregressive Moving Average Models

The classical regression model of [Chapter 2](#) was developed for the static case, namely, we only allow the dependent variable to be influenced by current values of the independent variables. In the time series case, it is desirable to allow the dependent variable to be influenced by the past values of the independent variables and possibly by its own past values. If the present can be plausibly modeled in terms of only the past values of the independent inputs, we have the enticing prospect that forecasting will be possible.

INTRODUCTION TO AUTOREGRESSIVE MODELS

Autoregressive models are based on the idea that the current value of the series, x_t , can be explained as a function of p past values, $x_{t-1}, x_{t-2}, \dots, x_{t-p}$, where p determines the number of steps into the past needed to forecast the current value. As a typical case, recall [Example 1.10](#) in which data were generated using the model

$$x_t = x_{t-1} - .90x_{t-2} + w_t,$$

where w_t is white Gaussian noise with $\sigma_w^2 = 1$. We have now assumed the current value is a particular *linear* function of past values. The regularity that persists in [Figure 1.9](#) gives an indication that forecasting for such a model might be a distinct possibility, say, through some version such as

$$x_{n+1}^n = x_n - .90x_{n-1},$$

where the quantity on the left-hand side denotes the forecast at the next period $n + 1$ based on the observed data, x_1, x_2, \dots, x_n . We will make this notion more precise in our discussion of forecasting ([Section 3.4](#)).

The extent to which it might be possible to forecast a real data series from its own past values can be assessed by looking at the autocorrelation function and the lagged scatterplot matrices discussed in [Chapter 2](#). For example, the lagged scatterplot matrix for the Southern Oscillation Index (SOI), shown in [Figure 2.8](#), gives a distinct indication that lags 1 and 2, for example, are linearly associated with the current value. The ACF shown in [Figure 1.16](#) shows relatively large positive values at lags 1, 2, 12, 24, and 36 and large negative values at 18, 30, and 42. We note also the possible relation between the SOI and Recruitment series indicated in the scatterplot matrix shown in [Figure 2.9](#). We will indicate in later sections on transfer function and vector AR modeling how to handle the dependence on values taken by other series.

The preceding discussion motivates the following definition.

Definition 3.1 An autoregressive model of order p , abbreviated **AR(p)**, is of the form

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \quad (3.1)$$

where x_t is stationary, $w_t \sim \text{wn}(0, \sigma_w^2)$, and $\phi_1, \phi_2, \dots, \phi_p$ are constants ($\phi_p \neq 0$). The mean of x_t in (3.1) is zero. If the mean, μ , of x_t is not zero, replace x_t by $x_t - \mu$ in (3.1),

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + \phi_2(x_{t-2} - \mu) + \cdots + \phi_p(x_{t-p} - \mu) + w_t,$$

or write

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \quad (3.2)$$

where $\alpha = \mu(1 - \phi_1 - \cdots - \phi_p)$.

We note that (3.2) is similar to the regression model of [Section 2.1](#), and hence the term auto (or self) regression. Some technical difficulties, however, develop from applying that model because the regressors, x_{t-1}, \dots, x_{t-p} , are random components, whereas z_t was assumed to be fixed. A useful form follows by using the backshift operator (2.29) to write the AR(p) model, (3.1), as

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)x_t = w_t, \quad (3.3)$$

or even more concisely as

$$\phi(B)x_t = w_t. \quad (3.4)$$

The properties of $\phi(B)$ are important in solving (3.4) for x_t . This leads to the following definition.

Definition 3.2 The autoregressive operator is defined to be

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p. \quad (3.5)$$

Example 3.1 The AR(1) Model

We initiate the investigation of AR models by considering the first-order model, AR(1), given by $x_t = \phi x_{t-1} + w_t$. Iterating backwards k times, we get

$$\begin{aligned} x_t &= \phi x_{t-1} + w_t = \phi(\phi x_{t-2} + w_{t-1}) + w_t \\ &= \phi^2 x_{t-2} + \phi w_{t-1} + w_t \\ &\vdots \\ &= \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j w_{t-j}. \end{aligned}$$

This method suggests that, by continuing to iterate backward, and provided that $|\phi| < 1$ and $\sup_t \text{var}(x_t) < \infty$, we can represent an AR(1) model as a linear process given by^{3.1}

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j}. \quad (3.6)$$

Representation (3.6) is called the stationary solution of the model. In fact, by simple substitution,

$$\underbrace{\sum_{j=0}^{\infty} \phi^j w_{t-j}}_{x_t} = \phi \underbrace{\left(\sum_{k=0}^{\infty} \phi^k w_{t-1-k} \right)}_{x_{t-1}} + w_t.$$

The AR(1) process defined by (3.6) is stationary with mean

$$E(x_t) = \sum_{j=0}^{\infty} \phi^j E(w_{t-j}) = 0,$$

and autocovariance function,

$$\begin{aligned} \gamma(h) &= \text{cov}(x_{t+h}, x_t) = E \left[\left(\sum_{j=0}^{\infty} \phi^j w_{t+h-j} \right) \left(\sum_{k=0}^{\infty} \phi^k w_{t-k} \right) \right] \\ &= E \left[\left(w_{t+h} + \cdots + \phi^h w_t + \phi^{h+1} w_{t-1} + \cdots \right) (w_t + \phi w_{t-1} + \cdots) \right] \quad (3.7) \\ &= \sigma_w^2 \sum_{j=0}^{\infty} \phi^{h+j} \phi^j = \sigma_w^2 \phi^h \sum_{j=0}^{\infty} \phi^{2j} = \frac{\sigma_w^2 \phi^h}{1 - \phi^2}, \quad h \geq 0. \end{aligned}$$

^{3.1} Note that $\lim_{k \rightarrow \infty} E \left(x_t - \sum_{j=0}^{k-1} \phi^j w_{t-j} \right)^2 = \lim_{k \rightarrow \infty} \phi^{2k} E \left(x_{t-k}^2 \right) = 0$, so (3.6) exists in the mean square sense (see Appendix A for a definition).

Recall that $\gamma(h) = \gamma(-h)$, so we will only exhibit the autocovariance function for $h \geq 0$. From (3.7), the ACF of an AR(1) is

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \phi^h, \quad h \geq 0, \quad (3.8)$$

and $\rho(h)$ satisfies the recursion

$$\rho(h) = \phi \rho(h-1), \quad h = 1, 2, \dots . \quad (3.9)$$

We will discuss the ACF of a general AR(p) model in [Section 3.3](#).

Example 3.2 The Sample Path of an AR(1) Process

[Figure 3.1](#) shows a time plot of two AR(1) processes, one with $\phi = .9$ and one with $\phi = -.9$; in both cases, $\sigma_w^2 = 1$. In the first case, $\rho(h) = .9^h$, for $h \geq 0$, so observations close together in time are positively correlated with each other. This result means that observations at contiguous time points will tend to be close in value to each other; this fact shows up in the top of [Figure 3.1](#) as a very smooth sample path for x_t . Now, contrast this with the case in which $\phi = -.9$, so that $\rho(h) = (-.9)^h$, for $h \geq 0$. This result means that observations at contiguous time points are negatively correlated but observations two time points apart are positively correlated. This fact shows up in the bottom of [Figure 3.1](#), where, for example, if an observation, x_t , is positive, the next observation, x_{t+1} , is typically negative, and the next observation, x_{t+2} , is typically positive. Thus, in this case, the sample path is very choppy.

The following R code can be used to obtain a figure similar to [Figure 3.1](#):

```
par(mfrow=c(2,1))
plot(arima.sim(list(order=c(1,0,0), ar=.9), n=100), ylab="x",
      main=(expression(AR(1)~~~phi==+.9)))
plot(arima.sim(list(order=c(1,0,0), ar=-.9), n=100), ylab="x",
      main=(expression(AR(1)~~~phi==-.9)))
```

Example 3.3 Explosive AR Models and Causality

In [Example 1.18](#), it was discovered that the random walk $x_t = x_{t-1} + w_t$ is not stationary. We might wonder whether there is a stationary AR(1) process with $|\phi| > 1$. Such processes are called explosive because the values of the time series quickly become large in magnitude. Clearly, because $|\phi|^j$ increases without bound as $j \rightarrow \infty$, $\sum_{j=0}^{k-1} \phi^j w_{t-j}$ will not converge (in mean square) as $k \rightarrow \infty$, so the intuition used to get (3.6) will not work directly. We can, however, modify that argument to obtain a stationary model as follows. Write $x_{t+1} = \phi x_t + w_{t+1}$, in which case,

$$\begin{aligned} x_t &= \phi^{-1} x_{t+1} - \phi^{-1} w_{t+1} = \phi^{-1} (\phi^{-1} x_{t+2} - \phi^{-1} w_{t+2}) - \phi^{-1} w_{t+1} \\ &\quad \vdots \\ &= \phi^{-k} x_{t+k} - \sum_{j=1}^{k-1} \phi^{-j} w_{t+j}, \end{aligned} \quad (3.10)$$

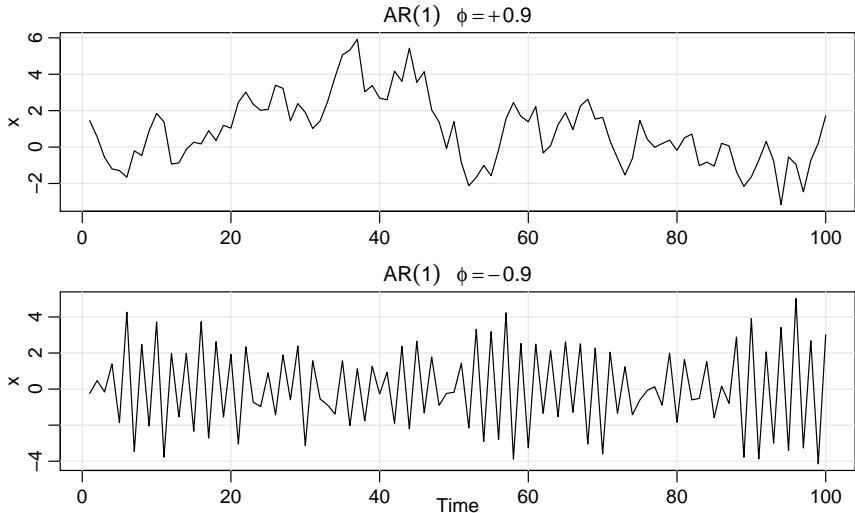


Fig. 3.1. Simulated AR(1) models: $\phi = .9$ (top); $\phi = -.9$ (bottom).

by iterating forward k steps. Because $|\phi|^{-1} < 1$, this result suggests the stationary future dependent AR(1) model

$$x_t = - \sum_{j=1}^{\infty} \phi^{-j} w_{t+j}. \quad (3.11)$$

The reader can verify that this is stationary and of the AR(1) form $x_t = \phi x_{t-1} + w_t$. Unfortunately, this model is useless because it requires us to know the future to be able to predict the future. When a process does not depend on the future, such as the AR(1) when $|\phi| < 1$, we will say the process is *causal*. In the explosive case of this example, the process is stationary, but it is also future dependent, and not causal.

Example 3.4 Every Explosion Has a Cause

Excluding explosive models from consideration is not a problem because the models have causal counterparts. For example, if

$$x_t = \phi x_{t-1} + w_t \quad \text{with} \quad |\phi| > 1$$

and $w_t \sim \text{iid } N(0, \sigma_w^2)$, then using (3.11), $\{x_t\}$ is a non-causal stationary Gaussian process with $E(x_t) = 0$ and

$$\begin{aligned} \gamma_x(h) &= \text{cov}(x_{t+h}, x_t) = \text{cov}\left(- \sum_{j=1}^{\infty} \phi^{-j} w_{t+h+j}, - \sum_{k=1}^{\infty} \phi^{-k} w_{t+k}\right) \\ &= \sigma_w^2 \phi^{-2} \phi^{-h} / (1 - \phi^{-2}). \end{aligned}$$

Thus, using (3.7), the causal process defined by

$$y_t = \phi^{-1} y_{t-1} + v_t$$

where $v_t \sim \text{iid } N(0, \sigma_w^2 \phi^{-2})$ is stochastically equal to the x_t process (i.e., all finite distributions of the processes are the same). For example, if $x_t = 2x_{t-1} + w_t$ with $\sigma_w^2 = 1$, then $y_t = \frac{1}{2}y_{t-1} + v_t$ with $\sigma_v^2 = 1/4$ is an equivalent causal process (see Problem 3.3). This concept generalizes to higher orders, but it is easier to show using Chapter 4 techniques; see Example 4.8.

The technique of iterating backward to get an idea of the stationary solution of AR models works well when $p = 1$, but not for larger orders. A general technique is that of matching coefficients. Consider the AR(1) model in operator form

$$\phi(B)x_t = w_t, \quad (3.12)$$

where $\phi(B) = 1 - \phi B$, and $|\phi| < 1$. Also, write the model in equation (3.6) using operator form as

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B)w_t, \quad (3.13)$$

where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$ and $\psi_j = \phi^j$. Suppose we did not know that $\psi_j = \phi^j$. We could substitute $\psi(B)w_t$ from (3.13) for x_t in (3.12) to obtain

$$\phi(B)\psi(B)w_t = w_t. \quad (3.14)$$

The coefficients of B on the left-hand side of (3.14) must be equal to those on right-hand side of (3.14), which means

$$(1 - \phi B)(1 + \psi_1 B + \psi_2 B^2 + \cdots + \psi_j B^j + \cdots) = 1. \quad (3.15)$$

Reorganizing the coefficients in (3.15),

$$1 + (\psi_1 - \phi)B + (\psi_2 - \psi_1\phi)B^2 + \cdots + (\psi_j - \psi_{j-1}\phi)B^j + \cdots = 1,$$

we see that for each $j = 1, 2, \dots$, the coefficient of B^j on the left must be zero because it is zero on the right. The coefficient of B on the left is $(\psi_1 - \phi)$, and equating this to zero, $\psi_1 - \phi = 0$, leads to $\psi_1 = \phi$. Continuing, the coefficient of B^2 is $(\psi_2 - \psi_1\phi)$, so $\psi_2 = \phi^2$. In general,

$$\psi_j = \psi_{j-1}\phi,$$

with $\psi_0 = 1$, which leads to the solution $\psi_j = \phi^j$.

Another way to think about the operations we just performed is to consider the AR(1) model in operator form, $\phi(B)x_t = w_t$. Now multiply both sides by $\phi^{-1}(B)$ (assuming the inverse operator exists) to get

$$\phi^{-1}(B)\phi(B)x_t = \phi^{-1}(B)w_t,$$

or

$$x_t = \phi^{-1}(B)w_t.$$

We know already that

$$\phi^{-1}(B) = 1 + \phi B + \phi^2 B^2 + \cdots + \phi^j B^j + \cdots,$$

that is, $\phi^{-1}(B)$ is $\psi(B)$ in (3.13). Thus, we notice that working with operators is like working with polynomials. That is, consider the polynomial $\phi(z) = 1 - \phi z$, where z is a complex number and $|\phi| < 1$. Then,

$$\phi^{-1}(z) = \frac{1}{(1 - \phi z)} = 1 + \phi z + \phi^2 z^2 + \cdots + \phi^j z^j + \cdots, \quad |z| \leq 1,$$

and the coefficients of B^j in $\phi^{-1}(B)$ are the same as the coefficients of z^j in $\phi^{-1}(z)$. In other words, we may treat the backshift operator, B , as a complex number, z . These results will be generalized in our discussion of ARMA models. We will find the polynomials corresponding to the operators useful in exploring the general properties of ARMA models.

INTRODUCTION TO MOVING AVERAGE MODELS

As an alternative to the autoregressive representation in which the x_t on the left-hand side of the equation are assumed to be combined linearly, the moving average model of order q , abbreviated as MA(q), assumes the white noise w_t on the right-hand side of the defining equation are combined linearly to form the observed data.

Definition 3.3 *The moving average model of order q , or MA(q) model, is defined to be*

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}, \quad (3.16)$$

where $w_t \sim \text{wn}(0, \sigma_w^2)$, and $\theta_1, \theta_2, \dots, \theta_q$ ($\theta_q \neq 0$) are parameters.^{3.2}

The system is the same as the infinite moving average defined as the linear process (3.13), where $\psi_0 = 1$, $\psi_j = \theta_j$, for $j = 1, \dots, q$, and $\psi_j = 0$ for other values. We may also write the MA(q) process in the equivalent form

$$x_t = \theta(B)w_t, \quad (3.17)$$

using the following definition.

Definition 3.4 *The moving average operator is*

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q. \quad (3.18)$$

Unlike the autoregressive process, the moving average process is stationary for any values of the parameters $\theta_1, \dots, \theta_q$; details of this result are provided in Section 3.3.

^{3.2} Some texts and software packages write the MA model with negative coefficients; that is, $x_t = w_t - \theta_1 w_{t-1} - \theta_2 w_{t-2} - \cdots - \theta_q w_{t-q}$.

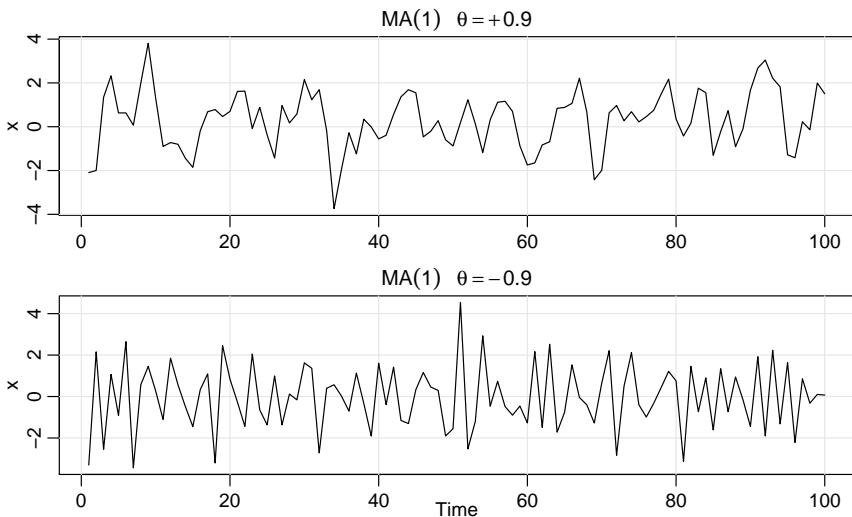


Fig. 3.2. Simulated MA(1) models: $\theta = .9$ (top); $\theta = -.9$ (bottom).

Example 3.5 The MA(1) Process

Consider the MA(1) model $x_t = w_t + \theta w_{t-1}$. Then, $E(x_t) = 0$,

$$\gamma(h) = \begin{cases} (1 + \theta^2)\sigma_w^2 & h = 0, \\ \theta\sigma_w^2 & h = 1, \\ 0 & h > 1, \end{cases}$$

and the ACF is

$$\rho(h) = \begin{cases} \frac{\theta}{(1+\theta^2)} & h = 1, \\ 0 & h > 1. \end{cases}$$

Note $|\rho(1)| \leq 1/2$ for all values of θ (Problem 3.1). Also, x_t is correlated with x_{t-1} , but not with x_{t-2}, x_{t-3}, \dots . Contrast this with the case of the AR(1) model in which the correlation between x_t and x_{t-k} is never zero. When $\theta = .9$, for example, x_t and x_{t-1} are positively correlated, and $\rho(1) = .497$. When $\theta = -.9$, x_t and x_{t-1} are negatively correlated, $\rho(1) = -.497$. Figure 3.2 shows a time plot of these two processes with $\sigma_w^2 = 1$. The series for which $\theta = .9$ is smoother than the series for which $\theta = -.9$.

A figure similar to Figure 3.2 can be created in R as follows:

```
par(mfrow = c(2,1))
plot(arima.sim(list(order=c(0,0,1), ma=.9), n=100), ylab="x",
      main=(expression(MA(1)~~~theta==+.5)))
plot(arima.sim(list(order=c(0,0,1), ma=-.9), n=100), ylab="x",
      main=(expression(MA(1)~~~theta==-.5)))
```

Example 3.6 Non-uniqueness of MA Models and Invertibility

Using Example 3.5, we note that for an MA(1) model, $\rho(h)$ is the same for θ and $\frac{1}{\theta}$; try 5 and $\frac{1}{5}$, for example. In addition, the pair $\sigma_w^2 = 1$ and $\theta = 5$ yield the same autocovariance function as the pair $\sigma_w^2 = 25$ and $\theta = 1/5$, namely,

$$\gamma(h) = \begin{cases} 26 & h = 0, \\ 5 & h = 1, \\ 0 & h > 1. \end{cases}$$

Thus, the MA(1) processes

$$x_t = w_t + \frac{1}{5}w_{t-1}, \quad w_t \sim \text{iid } N(0, 25)$$

and

$$y_t = v_t + 5v_{t-1}, \quad v_t \sim \text{iid } N(0, 1)$$

are the same because of normality (i.e., all finite distributions are the same). We can only observe the time series, x_t or y_t , and not the noise, w_t or v_t , so we cannot distinguish between the models. Hence, we will have to choose only one of them. For convenience, by mimicking the criterion of causality for AR models, we will choose the model with an infinite AR representation. Such a process is called an *invertible* process.

To discover which model is the invertible model, we can reverse the roles of x_t and w_t (because we are mimicking the AR case) and write the MA(1) model as $w_t = -\theta w_{t-1} + x_t$. Following the steps that led to (3.6), if $|\theta| < 1$, then $w_t = \sum_{j=0}^{\infty} (-\theta)^j x_{t-j}$, which is the desired infinite AR representation of the model. Hence, given a choice, we will choose the model with $\sigma_w^2 = 25$ and $\theta = 1/5$ because it is invertible.

As in the AR case, the polynomial, $\theta(z)$, corresponding to the moving average operators, $\theta(B)$, will be useful in exploring general properties of MA processes. For example, following the steps of equations (3.12)–(3.15), we can write the MA(1) model as $x_t = \theta(B)w_t$, where $\theta(B) = 1 + \theta B$. If $|\theta| < 1$, then we can write the model as $\pi(B)x_t = w_t$, where $\pi(B) = \theta^{-1}(B)$. Let $\theta(z) = 1 + \theta z$, for $|z| \leq 1$, then $\pi(z) = \theta^{-1}(z) = 1/(1 + \theta z) = \sum_{j=0}^{\infty} (-\theta)^j z^j$, and we determine that $\pi(B) = \sum_{j=0}^{\infty} (-\theta)^j B^j$.

AUTOREGRESSIVE MOVING AVERAGE MODELS

We now proceed with the general development of autoregressive, moving average, and mixed *autoregressive moving average* (ARMA), models for stationary time series.

Definition 3.5 A time series $\{x_t; t = 0, \pm 1, \pm 2, \dots\}$ is ARMA(p, q) if it is stationary and

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}, \quad (3.19)$$

with $\phi_p \neq 0$, $\theta_q \neq 0$, and $\sigma_w^2 > 0$. The parameters p and q are called the autoregressive and the moving average orders, respectively. If x_t has a nonzero mean μ , we set $\alpha = \mu(1 - \phi_1 - \dots - \phi_p)$ and write the model as

$$x_t = \alpha + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}, \quad (3.20)$$

where $w_t \sim \text{wn}(0, \sigma_w^2)$.

As previously noted, when $q = 0$, the model is called an autoregressive model of order p , AR(p), and when $p = 0$, the model is called a moving average model of order q , MA(q). To aid in the investigation of ARMA models, it will be useful to write them using the AR operator, (3.5), and the MA operator, (3.18). In particular, the ARMA(p, q) model in (3.19) can then be written in concise form as

$$\phi(B)x_t = \theta(B)w_t. \quad (3.21)$$

The concise form of the model points to a potential problem in that we can unnecessarily complicate the model by multiplying both sides by another operator, say

$$\eta(B)\phi(B)x_t = \eta(B)\theta(B)w_t,$$

without changing the dynamics. Consider the following example.

Example 3.7 Parameter Redundancy

Consider a white noise process $x_t = w_t$. If we multiply both sides of the equation by $\eta(B) = 1 - .5B$, then the model becomes $(1 - .5B)x_t = (1 - .5B)w_t$, or

$$x_t = .5x_{t-1} - .5w_{t-1} + w_t, \quad (3.22)$$

which looks like an ARMA(1, 1) model. Of course, x_t is still white noise; nothing has changed in this regard [i.e., $x_t = w_t$ is the solution to (3.22)], but we have hidden the fact that x_t is white noise because of the *parameter redundancy* or over-parameterization.

The consideration of parameter redundancy will be crucial when we discuss estimation for general ARMA models. As this example points out, we might fit an ARMA(1, 1) model to white noise data and find that the parameter estimates are significant. If we were unaware of parameter redundancy, we might claim the data are correlated when in fact they are not (Problem 3.20). Although we have not yet discussed estimation, we present the following demonstration of the problem. We generated 150 iid normals and then fit an ARMA(1, 1) to the data. Note that $\hat{\phi} = -.96$ and $\hat{\theta} = .95$, and both are significant. Below is the R code (note that the estimate called ‘intercept’ is really the estimate of the mean).

```
set.seed(8675309)      # Jenny, I got your number
x = rnorm(150, mean=5) # generate iid N(5, 1)s
arima(x, order=c(1, 0, 1)) # estimation
Coefficients:
            ar1      ma1   intercept<= misnomer
            -0.9595  0.9527    5.0462
            s.e.    0.1688  0.1750    0.0727
```

Thus, forgetting the mean estimate, the fitted model looks like

$$(1 + .96B)x_t = (1 + .95B)w_t,$$

which we should recognize as an over-parametrized model.

Example 3.3, **Example 3.6**, and **Example 3.7** point to a number of problems with the general definition of ARMA(p, q) models, as given by (3.19), or, equivalently, by (3.21). To summarize, we have seen the following problems:

- (i) parameter redundant models,
- (ii) stationary AR models that depend on the future, and
- (iii) MA models that are not unique.

To overcome these problems, we will require some additional restrictions on the model parameters. First, we make the following definitions.

Definition 3.6 *The AR and MA polynomials are defined as*

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p, \quad \phi_p \neq 0, \quad (3.23)$$

and

$$\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q, \quad \theta_q \neq 0, \quad (3.24)$$

respectively, where z is a complex number.

To address the first problem, we will henceforth refer to an ARMA(p, q) model to mean that it is in its simplest form. That is, in addition to the original definition given in equation (3.19), we will also require that $\phi(z)$ and $\theta(z)$ have no common factors. So, the process, $x_t = .5x_{t-1} - .5w_{t-1} + w_t$, discussed in **Example 3.7** is not referred to as an ARMA(1, 1) process because, in its reduced form, x_t is white noise.

To address the problem of future-dependent models, we formally introduce the concept of *causality*.

Definition 3.7 *An ARMA(p, q) model is said to be causal, if the time series $\{x_t; t = 0, \pm 1, \pm 2, \dots\}$ can be written as a one-sided linear process:*

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B)w_t, \quad (3.25)$$

where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$, and $\sum_{j=0}^{\infty} |\psi_j| < \infty$; we set $\psi_0 = 1$.

In **Example 3.3**, the AR(1) process, $x_t = \phi x_{t-1} + w_t$, is causal only when $|\phi| < 1$. Equivalently, the process is causal only when the root of $\phi(z) = 1 - \phi z$ is bigger than one in absolute value. That is, the root, say, z_0 , of $\phi(z)$ is $z_0 = 1/\phi$ (because $\phi(z_0) = 0$) and $|z_0| > 1$ because $|\phi| < 1$. In general, we have the following property.

Property 3.1 Causality of an ARMA(p, q) Process

An ARMA(p, q) model is causal if and only if $\phi(z) \neq 0$ for $|z| \leq 1$. The coefficients of the linear process given in (3.25) can be determined by solving

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\theta(z)}{\phi(z)}, \quad |z| \leq 1.$$

Another way to phrase **Property 3.1** is that *an ARMA process is causal only when the roots of $\phi(z)$ lie outside the unit circle*; that is, $\phi(z) = 0$ only when $|z| > 1$. Finally, to address the problem of uniqueness discussed in **Example 3.6**, we choose the model that allows an infinite autoregressive representation.

Definition 3.8 An ARMA(p, q) model is said to be **invertible**, if the time series $\{x_t; t = 0, \pm 1, \pm 2, \dots\}$ can be written as

$$\pi(B)x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} = w_t, \quad (3.26)$$

where $\pi(B) = \sum_{j=0}^{\infty} \pi_j B^j$, and $\sum_{j=0}^{\infty} |\pi_j| < \infty$; we set $\pi_0 = 1$.

Analogous to **Property 3.1**, we have the following property.

Property 3.2 Invertibility of an ARMA(p, q) Process

An ARMA(p, q) model is invertible if and only if $\theta(z) \neq 0$ for $|z| \leq 1$. The coefficients π_j of $\pi(B)$ given in (3.26) can be determined by solving

$$\pi(z) = \sum_{j=0}^{\infty} \pi_j z^j = \frac{\phi(z)}{\theta(z)}, \quad |z| \leq 1.$$

Another way to phrase **Property 3.2** is that *an ARMA process is invertible only when the roots of $\theta(z)$ lie outside the unit circle*; that is, $\theta(z) = 0$ only when $|z| > 1$. The proof of **Property 3.1** is given in **Section B.2** (the proof of **Property 3.2** is similar). The following examples illustrate these concepts.

Example 3.8 Parameter Redundancy, Causality, Invertibility

Consider the process

$$x_t = .4x_{t-1} + .45x_{t-2} + w_t + w_{t-1} + .25w_{t-2},$$

or, in operator form,

$$(1 - .4B - .45B^2)x_t = (1 + B + .25B^2)w_t.$$

At first, x_t appears to be an ARMA(2, 2) process. But notice that

$$\phi(B) = 1 - .4B - .45B^2 = (1 + .5B)(1 - .9B)$$

and

$$\theta(B) = (1 + B + .25B^2) = (1 + .5B)^2$$

have a common factor that can be canceled. After cancellation, the operators are $\phi(B) = (1 - .9B)$ and $\theta(B) = (1 + .5B)$, so the model is an ARMA(1, 1) model, $(1 - .9B)x_t = (1 + .5B)w_t$, or

$$x_t = .9x_{t-1} + .5w_{t-1} + w_t. \quad (3.27)$$

The model is causal because $\phi(z) = (1 - .9z) = 0$ when $z = 10/9$, which is outside the unit circle. The model is also invertible because the root of $\theta(z) = (1 + .5z)$ is $z = -2$, which is outside the unit circle.

To write the model as a linear process, we can obtain the ψ -weights using **Property 3.1**, $\phi(z)\psi(z) = \theta(z)$, or

$$(1 - .9z)(1 + \psi_1 z + \psi_2 z^2 + \cdots + \psi_j z^j + \cdots) = 1 + .5z.$$

Rearranging, we get

$$1 + (\psi_1 - .9)z + (\psi_2 - .9\psi_1)z^2 + \cdots + (\psi_j - .9\psi_{j-1})z^j + \cdots = 1 + .5z.$$

Matching the coefficients of z on the left and right sides we get $\psi_1 - .9 = .5$ and $\psi_j - .9\psi_{j-1} = 0$ for $j > 1$. Thus, $\psi_j = 1.4(.9)^{j-1}$ for $j \geq 1$ and (3.27) can be written as

$$x_t = w_t + 1.4 \sum_{j=1}^{\infty} .9^{j-1} w_{t-j}.$$

The values of ψ_j may be calculated in R as follows:

```
ARMAtoMA(ar = .9, ma = .5, 10) # first 10 psi-weights
[1] 1.40 1.26 1.13 1.02 0.92 0.83 0.74 0.67 0.60 0.54
```

The invertible representation using **Property 3.1** is obtained by matching coefficients in $\theta(z)\pi(z) = \phi(z)$,

$$(1 + .5z)(1 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + \cdots) = 1 - .9z.$$

In this case, the π -weights are given by $\pi_j = (-1)^j 1.4 (.5)^{j-1}$, for $j \geq 1$, and hence, because $w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$, we can also write (3.27) as

$$x_t = 1.4 \sum_{j=1}^{\infty} (-.5)^{j-1} x_{t-j} + w_t.$$

The values of π_j may be calculated in R as follows by reversing the roles of w_t and x_t ; i.e., write the model as $w_t = -.5w_{t-1} + x_t - .9x_{t-1}$:

```
ARMAtoMA(ar = -.5, ma = -.9, 10) # first 10 pi-weights
[1] -1.400 .700 -.350 .175 -.087 .044 -.022 .011 -.006 .003
```

Example 3.9 Causal Conditions for an AR(2) Process

For an AR(1) model, $(1 - \phi B)x_t = w_t$, to be causal, the root of $\phi(z) = 1 - \phi z$ must lie outside of the unit circle. In this case, $\phi(z) = 0$ when $z = 1/\phi$, so it is easy to go from the causal requirement on the root, $|1/\phi| > 1$, to a requirement on the parameter, $|\phi| < 1$. It is not so easy to establish this relationship for higher order models.

For example, the AR(2) model, $(1 - \phi_1 B - \phi_2 B^2)x_t = w_t$, is causal when the two roots of $\phi(z) = 1 - \phi_1 z - \phi_2 z^2$ lie outside of the unit circle. Using the quadratic formula, this requirement can be written as

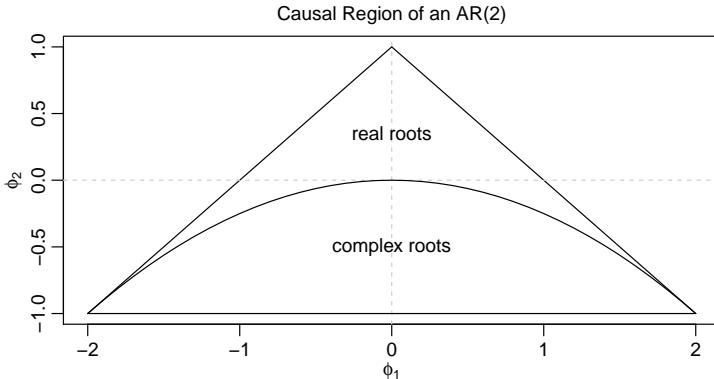


Fig. 3.3. Causal region for an AR(2) in terms of the parameters.

$$\left| \frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2} \right| > 1.$$

The roots of $\phi(z)$ may be real and distinct, real and equal, or a complex conjugate pair. If we denote those roots by z_1 and z_2 , we can write $\phi(z) = (1 - z_1^{-1}z)(1 - z_2^{-1}z)$; note that $\phi(z_1) = \phi(z_2) = 0$. The model can be written in operator form as $(1 - z_1^{-1}B)(1 - z_2^{-1}B)x_t = w_t$. From this representation, it follows that $\phi_1 = (z_1^{-1} + z_2^{-1})$ and $\phi_2 = -(z_1 z_2)^{-1}$. This relationship and the fact that $|z_1| > 1$ and $|z_2| > 1$ can be used to establish the following equivalent condition for causality:

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad |\phi_2| < 1. \quad (3.28)$$

This causality condition specifies a triangular region in the parameter space; see [Figure 3.3](#). We leave the details of the equivalence to the reader ([Problem 3.5](#)).

3.2 Difference Equations

The study of the behavior of ARMA processes and their ACFs is greatly enhanced by a basic knowledge of difference equations, simply because they are difference equations. We will give a brief and heuristic account of the topic along with some examples of the usefulness of the theory. For details, the reader is referred to Mickens (1990).

Suppose we have a sequence of numbers u_0, u_1, u_2, \dots such that

$$u_n - \alpha u_{n-1} = 0, \quad \alpha \neq 0, \quad n = 1, 2, \dots . \quad (3.29)$$

For example, recall [\(3.9\)](#) in which we showed that the ACF of an AR(1) process is a sequence, $\rho(h)$, satisfying

$$\rho(h) - \phi\rho(h-1) = 0, \quad h = 1, 2, \dots.$$

Equation (3.29) represents a *homogeneous difference equation of order 1*. To solve the equation, we write:

$$\begin{aligned} u_1 &= \alpha u_0 \\ u_2 &= \alpha u_1 = \alpha^2 u_0 \\ &\vdots \\ u_n &= \alpha u_{n-1} = \alpha^n u_0. \end{aligned}$$

Given an initial condition $u_0 = c$, we may solve (3.29), namely, $u_n = \alpha^n c$.

In operator notation, (3.29) can be written as $(1 - \alpha B)u_n = 0$. The polynomial associated with (3.29) is $\alpha(z) = 1 - \alpha z$, and the root, say, z_0 , of this polynomial is $z_0 = 1/\alpha$; that is $\alpha(z_0) = 0$. We know a solution (in fact, *the* solution) to (3.29), with initial condition $u_0 = c$, is

$$u_n = \alpha^n c = \left(z_0^{-1}\right)^n c. \quad (3.30)$$

That is, the solution to the difference equation (3.29) depends only on the initial condition and the inverse of the root to the associated polynomial $\alpha(z)$.

Now suppose that the sequence satisfies

$$u_n - \alpha_1 u_{n-1} - \alpha_2 u_{n-2} = 0, \quad \alpha_2 \neq 0, \quad n = 2, 3, \dots \quad (3.31)$$

This equation is a *homogeneous difference equation of order 2*. The corresponding polynomial is

$$\alpha(z) = 1 - \alpha_1 z - \alpha_2 z^2,$$

which has two roots, say, z_1 and z_2 ; that is, $\alpha(z_1) = \alpha(z_2) = 0$. We will consider two cases. First suppose $z_1 \neq z_2$. Then the general solution to (3.31) is

$$u_n = c_1 z_1^{-n} + c_2 z_2^{-n}, \quad (3.32)$$

where c_1 and c_2 depend on the initial conditions. The claim it is a solution can be verified by direct substitution of (3.32) into (3.31):

$$\begin{aligned} &\underbrace{(c_1 z_1^{-n} + c_2 z_2^{-n})}_{u_n} - \alpha_1 \underbrace{(c_1 z_1^{-(n-1)} + c_2 z_2^{-(n-1)})}_{u_{n-1}} - \alpha_2 \underbrace{(c_1 z_1^{-(n-2)} + c_2 z_2^{-(n-2)})}_{u_{n-2}} \\ &= c_1 z_1^{-n} \left(1 - \alpha_1 z_1 - \alpha_2 z_1^2\right) + c_2 z_2^{-n} \left(1 - \alpha_1 z_2 - \alpha_2 z_2^2\right) \\ &= c_1 z_1^{-n} \alpha(z_1) + c_2 z_2^{-n} \alpha(z_2) = 0. \end{aligned}$$

Given two initial conditions u_0 and u_1 , we may solve for c_1 and c_2 :

$$u_0 = c_1 + c_2 \quad \text{and} \quad u_1 = c_1 z_1^{-1} + c_2 z_2^{-1},$$

where z_1 and z_2 can be solved for in terms of α_1 and α_2 using the quadratic formula, for example.

When the roots are equal, $z_1 = z_2 (= z_0)$, a general solution to (3.31) is

$$u_n = z_0^{-n}(c_1 + c_2 n). \quad (3.33)$$

This claim can also be verified by direct substitution of (3.33) into (3.31):

$$\begin{aligned} & \underbrace{z_0^{-n}(c_1 + c_2 n)}_{u_n} - \alpha_1 \underbrace{(z_0^{-(n-1)}[c_1 + c_2(n-1)])}_{u_{n-1}} - \alpha_2 \underbrace{(z_0^{-(n-2)}[c_1 + c_2(n-2)])}_{u_{n-2}} \\ &= z_0^{-n}(c_1 + c_2 n) \left(1 - \alpha_1 z_0 - \alpha_2 z_0^2\right) + c_2 z_0^{-n+1} (\alpha_1 + 2\alpha_2 z_0) \\ &= c_2 z_0^{-n+1} (\alpha_1 + 2\alpha_2 z_0). \end{aligned}$$

To show that $(\alpha_1 + 2\alpha_2 z_0) = 0$, write $1 - \alpha_1 z - \alpha_2 z^2 = (1 - z_0^{-1} z)^2$, and take derivatives with respect to z on both sides of the equation to obtain $(\alpha_1 + 2\alpha_2 z) = 2z_0^{-1}(1 - z_0^{-1} z)$. Thus, $(\alpha_1 + 2\alpha_2 z_0) = 2z_0^{-1}(1 - z_0^{-1} z_0) = 0$, as was to be shown. Finally, given two initial conditions, u_0 and u_1 , we can solve for c_1 and c_2 :

$$u_0 = c_1 \quad \text{and} \quad u_1 = (c_1 + c_2)z_0^{-1}.$$

It can also be shown that these solutions are unique.

To summarize these results, in the case of distinct roots, the solution to the homogeneous difference equation of degree two was

$$\begin{aligned} u_n &= z_1^{-n} \times (\text{a polynomial in } n \text{ of degree } m_1 - 1) \\ &\quad + z_2^{-n} \times (\text{a polynomial in } n \text{ of degree } m_2 - 1), \end{aligned} \quad (3.34)$$

where m_1 is the multiplicity of the root z_1 and m_2 is the multiplicity of the root z_2 . In this example, of course, $m_1 = m_2 = 1$, and we called the polynomials of degree zero c_1 and c_2 , respectively. In the case of the repeated root, the solution was

$$u_n = z_0^{-n} \times (\text{a polynomial in } n \text{ of degree } m_0 - 1), \quad (3.35)$$

where m_0 is the multiplicity of the root z_0 ; that is, $m_0 = 2$. In this case, we wrote the polynomial of degree one as $c_1 + c_2 n$. In both cases, we solved for c_1 and c_2 given two initial conditions, u_0 and u_1 .

These results generalize to the homogeneous difference equation of order p :

$$u_n - \alpha_1 u_{n-1} - \cdots - \alpha_p u_{n-p} = 0, \quad \alpha_p \neq 0, \quad n = p, p+1, \dots. \quad (3.36)$$

The associated polynomial is $\alpha(z) = 1 - \alpha_1 z - \cdots - \alpha_p z^p$. Suppose $\alpha(z)$ has r distinct roots, z_1 with multiplicity m_1 , z_2 with multiplicity m_2 , \dots , and z_r with multiplicity m_r , such that $m_1 + m_2 + \cdots + m_r = p$. The general solution to the difference equation (3.36) is

$$u_n = z_1^{-n} P_1(n) + z_2^{-n} P_2(n) + \cdots + z_r^{-n} P_r(n), \quad (3.37)$$

where $P_j(n)$, for $j = 1, 2, \dots, r$, is a polynomial in n , of degree $m_j - 1$. Given p initial conditions u_0, \dots, u_{p-1} , we can solve for the $P_j(n)$ explicitly.

Example 3.10 The ACF of an AR(2) Process

Suppose $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$ is a causal AR(2) process. Multiply each side of the model by x_{t-h} for $h > 0$, and take expectation:

$$\text{E}(x_t x_{t-h}) = \phi_1 \text{E}(x_{t-1} x_{t-h}) + \phi_2 \text{E}(x_{t-2} x_{t-h}) + \text{E}(w_t x_{t-h}).$$

The result is

$$\gamma(h) = \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2), \quad h = 1, 2, \dots . \quad (3.38)$$

In (3.38), we used the fact that $\text{E}(x_t) = 0$ and for $h > 0$,

$$\text{E}(w_t x_{t-h}) = \text{E}\left(w_t \sum_{j=0}^{\infty} \psi_j w_{t-h-j}\right) = 0.$$

Divide (3.38) through by $\gamma(0)$ to obtain the difference equation for the ACF of the process:

$$\rho(h) - \phi_1 \rho(h-1) - \phi_2 \rho(h-2) = 0, \quad h = 1, 2, \dots . \quad (3.39)$$

The initial conditions are $\rho(0) = 1$ and $\rho(-1) = \phi_1 / (1 - \phi_2)$, which is obtained by evaluating (3.39) for $h = 1$ and noting that $\rho(1) = \rho(-1)$.

Using the results for the homogeneous difference equation of order two, let z_1 and z_2 be the roots of the associated polynomial, $\phi(z) = 1 - \phi_1 z - \phi_2 z^2$. Because the model is causal, we know the roots are outside the unit circle: $|z_1| > 1$ and $|z_2| > 1$. Now, consider the solution for three cases:

(i) When z_1 and z_2 are real and distinct, then

$$\rho(h) = c_1 z_1^{-h} + c_2 z_2^{-h},$$

so $\rho(h) \rightarrow 0$ exponentially fast as $h \rightarrow \infty$.

(ii) When $z_1 = z_2 (= z_0)$ are real and equal, then

$$\rho(h) = z_0^{-h} (c_1 + c_2 h),$$

so $\rho(h) \rightarrow 0$ exponentially fast as $h \rightarrow \infty$.

(iii) When $z_1 = \bar{z}_2$ are a complex conjugate pair, then $c_2 = \bar{c}_1$ (because $\rho(h)$ is real), and

$$\rho(h) = c_1 z_1^{-h} + \bar{c}_1 \bar{z}_1^{-h}.$$

Write c_1 and z_1 in polar coordinates, for example, $z_1 = |z_1| e^{i\theta}$, where θ is the angle whose tangent is the ratio of the imaginary part and the real part of z_1 (sometimes called $\arg(z_1)$; the range of θ is $[-\pi, \pi]$). Then, using the fact that $e^{i\alpha} + e^{-i\alpha} = 2 \cos(\alpha)$, the solution has the form

$$\rho(h) = a |z_1|^{-h} \cos(h\theta + b),$$

where a and b are determined by the initial conditions. Again, $\rho(h)$ dampens to zero exponentially fast as $h \rightarrow \infty$, but it does so in a sinusoidal fashion. The implication of this result is shown in the next example.

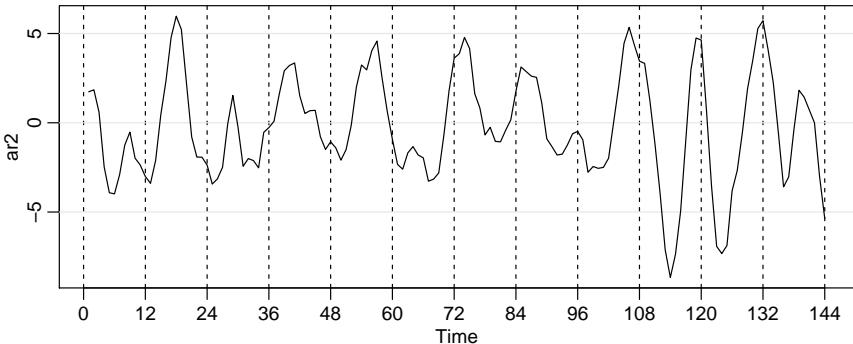


Fig. 3.4. Simulated AR(2) model, $n = 144$ with $\phi_1 = 1.5$ and $\phi_2 = -.75$.

Example 3.11 An AR(2) with Complex Roots

Figure 3.4 shows $n = 144$ observations from the AR(2) model

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t,$$

with $\sigma_w^2 = 1$, and with complex roots chosen so the process exhibits pseudo-cyclic behavior at the rate of one cycle every 12 time points. The autoregressive polynomial for this model is $\phi(z) = 1 - 1.5z + .75z^2$. The roots of $\phi(z)$ are $1 \pm i/\sqrt{3}$, and $\theta = \tan^{-1}(1/\sqrt{3}) = 2\pi/12$ radians per unit time. To convert the angle to cycles per unit time, divide by 2π to get 1/12 cycles per unit time. The ACF for this model is shown in left-hand-side of Figure 3.5.

To calculate the roots of the polynomial and solve for arg in R:

```
z = c(1, -1.5, .75)      # coefficients of the polynomial
(a = polyroot(z)[1])    # print one root = 1 + i/sqrt(3)
[1] 1+0.57735i
arg = Arg(a)/(2*pi)     # arg in cycles/pt
1/arg                   # the pseudo period
[1] 12
```

To reproduce Figure 3.4:

```
set.seed(8675309)
ar2 = arima.sim(list(order=c(2,0,0), ar=c(1.5,-.75)), n = 144)
plot(ar2, axes=FALSE, xlab="Time")
axis(2); axis(1, at=seq(0,144,by=12));  box()
abline(v=seq(0,144,by=12), lty=2)
```

To calculate and display the ACF for this model:

```
ACF = ARMAacf(ar=c(1.5,-.75), ma=0, 50)
plot(ACF, type="h", xlab="lag")
abline(h=0)
```

Example 3.12 The ψ -weights for an ARMA Model

For a causal ARMA(p, q) model, $\phi(B)x_t = \theta(B)w_t$, where the zeros of $\phi(z)$ are outside the unit circle, recall that we may write

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j},$$

where the ψ -weights are determined using [Property 3.1](#).

For the pure MA(q) model, $\psi_0 = 1$, $\psi_j = \theta_j$, for $j = 1, \dots, q$, and $\psi_j = 0$, otherwise. For the general case of ARMA(p, q) models, the task of solving for the ψ -weights is much more complicated, as was demonstrated in [Example 3.8](#). The use of the theory of homogeneous difference equations can help here. To solve for the ψ -weights in general, we must match the coefficients in $\phi(z)\psi(z) = \theta(z)$:

$$(1 - \phi_1 z - \phi_2 z^2 - \dots)(\psi_0 + \psi_1 z + \psi_2 z^2 + \dots) = (1 + \theta_1 z + \theta_2 z^2 + \dots).$$

The first few values are

$$\begin{aligned}\psi_0 &= 1 \\ \psi_1 - \phi_1 \psi_0 &= \theta_1 \\ \psi_2 - \phi_1 \psi_1 - \phi_2 \psi_0 &= \theta_2 \\ \psi_3 - \phi_1 \psi_2 - \phi_2 \psi_1 - \phi_3 \psi_0 &= \theta_3 \\ &\vdots\end{aligned}$$

where we would take $\phi_j = 0$ for $j > p$, and $\theta_j = 0$ for $j > q$. The ψ -weights satisfy the homogeneous difference equation given by

$$\psi_j - \sum_{k=1}^p \phi_k \psi_{j-k} = 0, \quad j \geq \max(p, q+1), \quad (3.40)$$

with initial conditions

$$\psi_j - \sum_{k=1}^j \phi_k \psi_{j-k} = \theta_j, \quad 0 \leq j < \max(p, q+1). \quad (3.41)$$

The general solution depends on the roots of the AR polynomial $\phi(z) = 1 - \phi_1 z - \dots - \phi_p z^p$, as seen from (3.40). The specific solution will, of course, depend on the initial conditions.

Consider the ARMA process given in (3.27), $x_t = .9x_{t-1} + .5w_{t-1} + w_t$. Because $\max(p, q+1) = 2$, using (3.41), we have $\psi_0 = 1$ and $\psi_1 = .9 + .5 = 1.4$. By (3.40), for $j = 2, 3, \dots$, the ψ -weights satisfy $\psi_j - .9\psi_{j-1} = 0$. The general solution is $\psi_j = c \cdot .9^j$. To find the specific solution, use the initial condition $\psi_1 = 1.4$, so $1.4 = .9c$ or $c = 1.4/.9$. Finally, $\psi_j = 1.4(.9)^{j-1}$, for $j \geq 1$, as we saw in [Example 3.8](#).

To view, for example, the first 50 ψ -weights in R, use:

```
ARMAtoMA(ar=.9, ma=.5, 50)      # for a list
plot(ARMAtoMA(ar=.9, ma=.5, 50)) # for a graph
```

3.3 Autocorrelation and Partial Autocorrelation

We begin by exhibiting the ACF of an MA(q) process, $x_t = \theta(B)w_t$, where $\theta(B) = 1 + \theta_1B + \dots + \theta_qB^q$. Because x_t is a finite linear combination of white noise terms, the process is stationary with mean

$$\mathbb{E}(x_t) = \sum_{j=0}^q \theta_j \mathbb{E}(w_{t-j}) = 0,$$

where we have written $\theta_0 = 1$, and with autocovariance function

$$\begin{aligned} \gamma(h) &= \text{cov}(x_{t+h}, x_t) = \text{cov}\left(\sum_{j=0}^q \theta_j w_{t+h-j}, \sum_{k=0}^q \theta_k w_{t-k}\right) \\ &= \begin{cases} \sigma_w^2 \sum_{j=0}^{q-h} \theta_j \theta_{j+h}, & 0 \leq h \leq q \\ 0 & h > q. \end{cases} \end{aligned} \quad (3.42)$$

Recall that $\gamma(h) = \gamma(-h)$, so we will only display the values for $h \geq 0$. Note that $\gamma(q)$ cannot be zero because $\theta_q \neq 0$. The cutting off of $\gamma(h)$ after q lags is the signature of the MA(q) model. Dividing (3.42) by $\gamma(0)$ yields the *ACF of an MA(q)*:

$$\rho(h) = \begin{cases} \frac{\sum_{j=0}^{q-h} \theta_j \theta_{j+h}}{1 + \theta_1^2 + \dots + \theta_q^2} & 1 \leq h \leq q \\ 0 & h > q. \end{cases} \quad (3.43)$$

For a causal ARMA(p, q) model, $\phi(B)x_t = \theta(B)w_t$, where the zeros of $\phi(z)$ are outside the unit circle, write

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}. \quad (3.44)$$

It follows immediately that $\mathbb{E}(x_t) = 0$ and the autocovariance function of x_t is

$$\gamma(h) = \text{cov}(x_{t+h}, x_t) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+h}, \quad h \geq 0. \quad (3.45)$$

We could then use (3.40) and (3.41) to solve for the ψ -weights. In turn, we could solve for $\gamma(h)$, and the ACF $\rho(h) = \gamma(h)/\gamma(0)$. As in Example 3.10, it is also possible to obtain a homogeneous difference equation directly in terms of $\gamma(h)$. First, we write

$$\begin{aligned} \gamma(h) &= \text{cov}(x_{t+h}, x_t) = \text{cov}\left(\sum_{j=1}^p \phi_j x_{t+h-j} + \sum_{j=0}^q \theta_j w_{t+h-j}, x_t\right) \\ &= \sum_{j=1}^p \phi_j \gamma(h-j) + \sigma_w^2 \sum_{j=h}^q \theta_j \psi_{j-h}, \quad h \geq 0, \end{aligned} \quad (3.46)$$

where we have used the fact that, for $h \geq 0$,

$$\text{cov}(w_{t+h-j}, x_t) = \text{cov}\left(w_{t+h-j}, \sum_{k=0}^{\infty} \psi_k w_{t-k}\right) = \psi_{j-h} \sigma_w^2.$$

From (3.46), we can write a *general homogeneous equation for the ACF of a causal ARMA process*:

$$\gamma(h) - \phi_1 \gamma(h-1) - \cdots - \phi_p \gamma(h-p) = 0, \quad h \geq \max(p, q+1), \quad (3.47)$$

with initial conditions

$$\gamma(h) - \sum_{j=1}^p \phi_j \gamma(h-j) = \sigma_w^2 \sum_{j=h}^q \theta_j \psi_{j-h}, \quad 0 \leq h < \max(p, q+1). \quad (3.48)$$

Dividing (3.47) and (3.48) through by $\gamma(0)$ will allow us to solve for the ACF, $\rho(h) = \gamma(h)/\gamma(0)$.

Example 3.13 The ACF of an AR(p)

In Example 3.10 we considered the case where $p = 2$. For the general case, it follows immediately from (3.47) that

$$\rho(h) - \phi_1 \rho(h-1) - \cdots - \phi_p \rho(h-p) = 0, \quad h \geq p. \quad (3.49)$$

Let z_1, \dots, z_r denote the roots of $\phi(z)$, each with multiplicity m_1, \dots, m_r , respectively, where $m_1 + \cdots + m_r = p$. Then, from (3.37), the general solution is

$$\rho(h) = z_1^{-h} P_1(h) + z_2^{-h} P_2(h) + \cdots + z_r^{-h} P_r(h), \quad h \geq p, \quad (3.50)$$

where $P_j(h)$ is a polynomial in h of degree $m_j - 1$.

Recall that for a causal model, all of the roots are outside the unit circle, $|z_i| > 1$, for $i = 1, \dots, r$. If all the roots are real, then $\rho(h)$ dampens exponentially fast to zero as $h \rightarrow \infty$. If some of the roots are complex, then they will be in conjugate pairs and $\rho(h)$ will dampen, in a sinusoidal fashion, exponentially fast to zero as $h \rightarrow \infty$. In the case of complex roots, the time series will appear to be cyclic in nature. This, of course, is also true for ARMA models in which the AR part has complex roots.

Example 3.14 The ACF of an ARMA(1, 1)

Consider the ARMA(1, 1) process $x_t = \phi x_{t-1} + \theta w_{t-1} + w_t$, where $|\phi| < 1$. Based on (3.47), the autocovariance function satisfies

$$\gamma(h) - \phi \gamma(h-1) = 0, \quad h = 2, 3, \dots,$$

and it follows from (3.29)–(3.30) that the general solution is

$$\gamma(h) = c \phi^h, \quad h = 1, 2, \dots . \quad (3.51)$$

To obtain the initial conditions, we use (3.48):

$$\gamma(0) = \phi\gamma(1) + \sigma_w^2[1 + \theta\phi + \theta^2] \quad \text{and} \quad \gamma(1) = \phi\gamma(0) + \sigma_w^2\theta.$$

Solving for $\gamma(0)$ and $\gamma(1)$, we obtain:

$$\gamma(0) = \sigma_w^2 \frac{1 + 2\theta\phi + \theta^2}{1 - \phi^2} \quad \text{and} \quad \gamma(1) = \sigma_w^2 \frac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2}.$$

To solve for c , note that from (3.51), $\gamma(1) = c\phi$ or $c = \gamma(1)/\phi$. Hence, the specific solution for $h \geq 1$ is

$$\gamma(h) = \frac{\gamma(1)}{\phi} \phi^h = \sigma_w^2 \frac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2} \phi^{h-1}.$$

Finally, dividing through by $\gamma(0)$ yields the ACF

$$\rho(h) = \frac{(1 + \theta\phi)(\phi + \theta)}{1 + 2\theta\phi + \theta^2} \phi^{h-1}, \quad h \geq 1. \quad (3.52)$$

Notice that the general pattern of $\rho(h)$ versus h in (3.52) is not different from that of an AR(1) given in (3.8). Hence, it is unlikely that we will be able to tell the difference between an ARMA(1,1) and an AR(1) based solely on an ACF estimated from a sample. This consideration will lead us to the partial autocorrelation function.

THE PARTIAL AUTOCORRELATION FUNCTION (PACF)

We have seen in (3.43), for MA(q) models, the ACF will be zero for lags greater than q . Moreover, because $\theta_q \neq 0$, the ACF will not be zero at lag q . Thus, the ACF provides a considerable amount of information about the order of the dependence when the process is a moving average process. If the process, however, is ARMA or AR, the ACF alone tells us little about the orders of dependence. Hence, it is worthwhile pursuing a function that will behave like the ACF of MA models, but for AR models, namely, the *partial autocorrelation function (PACF)*.

Recall that if X , Y , and Z are random variables, then the partial correlation between X and Y given Z is obtained by regressing X on Z to obtain \hat{X} , regressing Y on Z to obtain \hat{Y} , and then calculating

$$\rho_{XY|Z} = \text{corr}\{X - \hat{X}, Y - \hat{Y}\}.$$

The idea is that $\rho_{XY|Z}$ measures the correlation between X and Y with the linear effect of Z removed (or partialled out). If the variables are multivariate normal, then this definition coincides with $\rho_{XY|Z} = \text{corr}(X, Y | Z)$.

To motivate the idea for time series, consider a causal AR(1) model, $x_t = \phi x_{t-1} + w_t$. Then,

$$\begin{aligned} \gamma_x(2) &= \text{cov}(x_t, x_{t-2}) = \text{cov}(\phi x_{t-1} + w_t, x_{t-2}) \\ &= \text{cov}(\phi^2 x_{t-2} + \phi w_{t-1} + w_t, x_{t-2}) = \phi^2 \gamma_x(0). \end{aligned}$$

This result follows from causality because x_{t-2} involves $\{w_{t-2}, w_{t-3}, \dots\}$, which are all uncorrelated with w_t and w_{t-1} . The correlation between x_t and x_{t-2} is not zero, as it would be for an MA(1), because x_t is dependent on x_{t-2} through x_{t-1} . Suppose we break this chain of dependence by removing (or partial out) the effect x_{t-1} . That is, we consider the correlation between $x_t - \phi x_{t-1}$ and $x_{t-2} - \phi x_{t-1}$, because it is the correlation between x_t and x_{t-2} with the linear dependence of each on x_{t-1} removed. In this way, we have broken the dependence chain between x_t and x_{t-2} . In fact,

$$\text{cov}(x_t - \phi x_{t-1}, x_{t-2} - \phi x_{t-1}) = \text{cov}(w_t, x_{t-2} - \phi x_{t-1}) = 0.$$

Hence, the tool we need is partial autocorrelation, which is the correlation between x_s and x_t with the linear effect of everything “in the middle” removed.

To formally define the PACF for mean-zero stationary time series, let \hat{x}_{t+h} , for $h \geq 2$, denote the regression^{3.3} of x_{t+h} on $\{x_{t+h-1}, x_{t+h-2}, \dots, x_{t+1}\}$, which we write as

$$\hat{x}_{t+h} = \beta_1 x_{t+h-1} + \beta_2 x_{t+h-2} + \dots + \beta_{h-1} x_{t+1}. \quad (3.53)$$

No intercept term is needed in (3.53) because the mean of x_t is zero (otherwise, replace x_t by $x_t - \mu_x$ in this discussion). In addition, let \hat{x}_t denote the regression of x_t on $\{x_{t+1}, x_{t+2}, \dots, x_{t+h-1}\}$, then

$$\hat{x}_t = \beta_1 x_{t+1} + \beta_2 x_{t+2} + \dots + \beta_{h-1} x_{t+h-1}. \quad (3.54)$$

Because of stationarity, the coefficients, $\beta_1, \dots, \beta_{h-1}$ are the same in (3.53) and (3.54); we will explain this result in the next section, but it will be evident from the examples.

Definition 3.9 *The partial autocorrelation function (PACF) of a stationary process, x_t , denoted ϕ_{hh} , for $h = 1, 2, \dots$, is*

$$\phi_{11} = \text{corr}(x_{t+1}, x_t) = \rho(1) \quad (3.55)$$

and

$$\phi_{hh} = \text{corr}(x_{t+h} - \hat{x}_{t+h}, x_t - \hat{x}_t), \quad h \geq 2. \quad (3.56)$$

The reason for using a double subscript will become evident in the next section. The PACF, ϕ_{hh} , is the correlation between x_{t+h} and x_t with the linear dependence of $\{x_{t+1}, \dots, x_{t+h-1}\}$ on each, removed. If the process x_t is Gaussian, then $\phi_{hh} = \text{corr}(x_{t+h}, x_t | x_{t+1}, \dots, x_{t+h-1})$; that is, ϕ_{hh} is the correlation coefficient between x_{t+h} and x_t in the bivariate distribution of (x_{t+h}, x_t) conditional on $\{x_{t+1}, \dots, x_{t+h-1}\}$.

^{3.3} The term regression here refers to regression in the population sense. That is, \hat{x}_{t+h} is the linear combination of $\{x_{t+h-1}, x_{t+h-2}, \dots, x_{t+1}\}$ that minimizes the mean squared error $E(x_{t+h} - \sum_{j=1}^{h-1} \alpha_j x_{t+j})^2$.

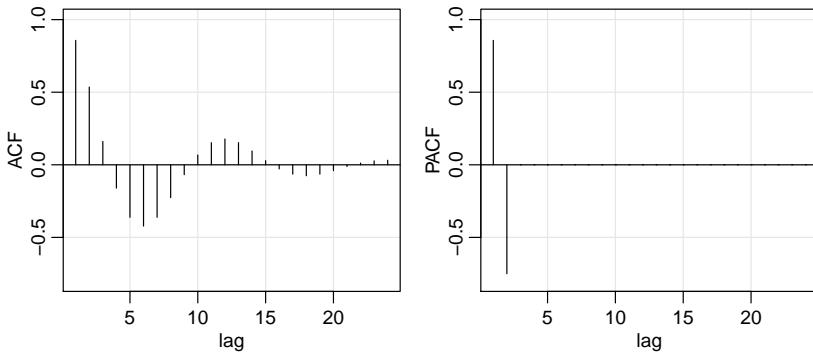


Fig. 3.5. The ACF and PACF of an AR(2) model with $\phi_1 = 1.5$ and $\phi_2 = -.75$.

Example 3.15 The PACF of an AR(1)

Consider the PACF of the AR(1) process given by $x_t = \phi x_{t-1} + w_t$, with $|\phi| < 1$. By definition, $\phi_{11} = \rho(1) = \phi$. To calculate ϕ_{22} , consider the regression of x_{t+2} on x_{t+1} , say, $\hat{x}_{t+2} = \beta x_{t+1}$. We choose β to minimize

$$\text{E}(x_{t+2} - \hat{x}_{t+2})^2 = \text{E}(x_{t+2} - \beta x_{t+1})^2 = \gamma(0) - 2\beta\gamma(1) + \beta^2\gamma(0).$$

Taking derivatives with respect to β and setting the result equal to zero, we have $\beta = \gamma(1)/\gamma(0) = \rho(1) = \phi$. Next, consider the regression of x_t on x_{t+1} , say $\hat{x}_t = \beta x_{t+1}$. We choose β to minimize

$$\text{E}(x_t - \hat{x}_t)^2 = \text{E}(x_t - \beta x_{t+1})^2 = \gamma(0) - 2\beta\gamma(1) + \beta^2\gamma(0).$$

This is the same equation as before, so $\beta = \phi$. Hence,

$$\begin{aligned}\phi_{22} &= \text{corr}(x_{t+2} - \hat{x}_{t+2}, x_t - \hat{x}_t) = \text{corr}(x_{t+2} - \phi x_{t+1}, x_t - \phi x_{t+1}) \\ &= \text{corr}(w_{t+2}, x_t - \phi x_{t+1}) = 0\end{aligned}$$

by causality. Thus, $\phi_{22} = 0$. In the next example, we will see that in this case, $\phi_{hh} = 0$ for all $h > 1$.

Example 3.16 The PACF of an AR(p)

The model implies $x_{t+h} = \sum_{j=1}^p \phi_j x_{t+h-j} + w_{t+h}$, where the roots of $\phi(z)$ are outside the unit circle. When $h > p$, the regression of x_{t+h} on $\{x_{t+1}, \dots, x_{t+h-1}\}$, is

$$\hat{x}_{t+h} = \sum_{j=1}^p \phi_j x_{t+h-j}.$$

We have not proved this obvious result yet, but we will prove it in the next section. Thus, when $h > p$,

Table 3.1. Behavior of the ACF and PACF for ARMA Models

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

$$\phi_{hh} = \text{corr}(x_{t+h} - \hat{x}_{t+h}, x_t - \hat{x}_t) = \text{corr}(w_{t+h}, x_t - \hat{x}_t) = 0,$$

because, by causality, $x_t - \hat{x}_t$ depends only on $\{w_{t+h-1}, w_{t+h-2}, \dots\}$; recall equation (3.54). When $h \leq p$, ϕ_{pp} is not zero, and $\phi_{11}, \dots, \phi_{p-1,p-1}$ are not necessarily zero. We will see later that, in fact, $\phi_{pp} = \phi_p$. Figure 3.5 shows the ACF and the PACF of the AR(2) model presented in Example 3.11. To reproduce Figure 3.5 in R, use the following commands:

```
ACF = ARMAacf(ar=c(1.5,-.75), ma=0, 24)[-1]
PACF = ARMAacf(ar=c(1.5,-.75), ma=0, 24, pacf=TRUE)
par(mfrow=c(1,2))
plot(ACF, type="h", xlab="lag", ylim=c(-.8,1)); abline(h=0)
plot(PACF, type="h", xlab="lag", ylim=c(-.8,1)); abline(h=0)
```

Example 3.17 The PACF of an Invertible MA(q)

For an invertible MA(q), we can write $x_t = -\sum_{j=1}^{\infty} \pi_j x_{t-j} + w_t$. Moreover, no finite representation exists. From this result, it should be apparent that the PACF will never cut off, as in the case of an AR(p).

For an MA(1), $x_t = w_t + \theta w_{t-1}$, with $|\theta| < 1$, calculations similar to Example 3.15 will yield $\phi_{22} = -\theta^2/(1+\theta^2+\theta^4)$. For the MA(1) in general, we can show that

$$\phi_{hh} = -\frac{(-\theta)^h(1-\theta^2)}{1-\theta^{2(h+1)}}, \quad h \geq 1.$$

In the next section, we will discuss methods of calculating the PACF. The PACF for MA models behaves much like the ACF for AR models. Also, the PACF for AR models behaves much like the ACF for MA models. Because an invertible ARMA model has an infinite AR representation, the PACF will not cut off. We may summarize these results in Table 3.1.

Example 3.18 Preliminary Analysis of the Recruitment Series

We consider the problem of modeling the Recruitment series shown in Figure 1.5. There are 453 months of observed recruitment ranging over the years 1950-1987. The ACF and the PACF given in Figure 3.6 are consistent with the behavior of an AR(2). The ACF has cycles corresponding roughly to a 12-month period, and the PACF has large values for $h = 1, 2$ and then is essentially zero for higher order lags. Based on Table 3.1, these results suggest that a second-order ($p = 2$) autoregressive model might provide a good fit. Although we will discuss estimation

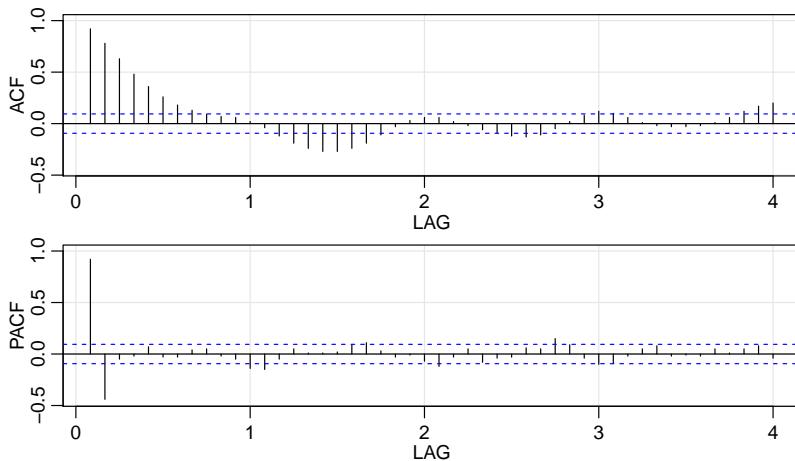


Fig. 3.6. ACF and PACF of the Recruitment series. Note that the lag axes are in terms of season (12 months in this case).

in detail in [Section 3.5](#), we ran a regression (see [Section 2.1](#)) using the data triplets $\{(x; z_1, z_2) : (x_3; x_1, x_2), (x_4; x_2, x_1), \dots, (x_{453}; x_{451}, x_{452})\}$ to fit a model of the form

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

for $t = 3, 4, \dots, 453$. The estimates and standard errors (in parentheses) are $\hat{\phi}_0 = 6.74_{(1.11)}$, $\hat{\phi}_1 = 1.35_{(.04)}$, $\hat{\phi}_2 = -.46_{(.04)}$, and $\hat{\sigma}_w^2 = 89.72$.

The following R code can be used for this analysis. We use `acf2` from `astsa` to print and plot the ACF and PACF.

```
acf2(rec, 48)      # will produce values and a graphic
(regr = ar.ols(rec, order=2, demean=FALSE, intercept=TRUE))
regr$asy.se.coef # standard errors of the estimates
```

3.4 Forecasting

In forecasting, the goal is to predict future values of a time series, x_{n+m} , $m = 1, 2, \dots$, based on the data collected to the present, $x_{1:n} = \{x_1, x_2, \dots, x_n\}$. Throughout this section, we will assume x_t is stationary and the model parameters are known. The problem of forecasting when the model parameters are unknown will be discussed in the next section; also, see [Problem 3.26](#). The minimum mean square error predictor of x_{n+m} is

$$x_{n+m}^n = E(x_{n+m} | x_{1:n}) \tag{3.57}$$

because the conditional expectation minimizes the mean square error

$$E [x_{n+m} - g(x_{1:n})]^2, \tag{3.58}$$

where $g(x_{1:n})$ is a function of the observations $x_{1:n}$; see [Problem 3.14](#).

First, we will restrict attention to predictors that are linear functions of the data, that is, predictors of the form

$$x_{n+m}^n = \alpha_0 + \sum_{k=1}^n \alpha_k x_k, \quad (3.59)$$

where $\alpha_0, \alpha_1, \dots, \alpha_n$ are real numbers. We note that the α s depend on n and m , but for now we drop the dependence from the notation. For example, if $n = m = 1$, then x_2^1 is the one-step-ahead linear forecast of x_2 given x_1 . In terms of (3.59), $x_2^1 = \alpha_0 + \alpha_1 x_1$. But if $n = 2$, x_3^2 is the one-step-ahead linear forecast of x_3 given x_1 and x_2 . In terms of (3.59), $x_3^2 = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2$, and in general, the α s in x_2^1 and x_3^2 will be different.

Linear predictors of the form (3.59) that minimize the mean square prediction error (3.58) are called *best linear predictors* (BLPs). As we shall see, linear prediction depends only on the second-order moments of the process, which are easy to estimate from the data. Much of the material in this section is enhanced by the theoretical material presented in [Appendix B](#). For example, [Theorem B.3](#) states that if the process is Gaussian, minimum mean square error predictors and best linear predictors are the same. The following property, which is based on the Projection Theorem, [Theorem B.1](#), is a key result.

Property 3.3 Best Linear Prediction for Stationary Processes

Given data x_1, \dots, x_n , the best linear predictor, $x_{n+m}^n = \alpha_0 + \sum_{k=1}^n \alpha_k x_k$, of x_{n+m} , for $m \geq 1$, is found by solving

$$E[(x_{n+m} - x_{n+m}^n) x_k] = 0, \quad k = 0, 1, \dots, n, \quad (3.60)$$

where $x_0 = 1$, for $\alpha_0, \alpha_1, \dots, \alpha_n$.

The equations specified in (3.60) are called the *prediction equations*, and they are used to solve for the coefficients $\{\alpha_0, \alpha_1, \dots, \alpha_n\}$. The results of [Property 3.3](#) can also be obtained via least squares; i.e., to minimize $Q = E(x_{n+m} - \sum_{k=0}^n \alpha_k x_k)^2$ with respect to the α s, solve $\partial Q / \partial \alpha_j = 0$ for the α_j , $j = 0, 1, \dots, n$. This leads to (3.60).

If $E(x_t) = \mu$, the first equation ($k = 0$) of (3.60) implies

$$E(x_{n+m}^n) = E(x_{n+m}) = \mu.$$

Thus, taking expectation in (3.59), we have

$$\mu = \alpha_0 + \sum_{k=1}^n \alpha_k \mu \quad \text{or} \quad \alpha_0 = \mu \left(1 - \sum_{k=1}^n \alpha_k \right).$$

Hence, the form of the BLP is

$$x_{n+m}^n = \mu + \sum_{k=1}^n \alpha_k (x_k - \mu).$$

Thus, until we discuss estimation, there is no loss of generality in considering the case that $\mu = 0$, in which case, $\alpha_0 = 0$.

First, consider *one-step-ahead prediction*. That is, given $\{x_1, \dots, x_n\}$, we wish to forecast the value of the time series at the next time point, x_{n+1} . The BLP of x_{n+1} is of the form

$$x_{n+1}^n = \phi_{n1}x_n + \phi_{n2}x_{n-1} + \dots + \phi_{nn}x_1, \quad (3.61)$$

where we now display the dependence of the coefficients on n ; in this case, α_k in (3.59) is $\phi_{n,n+1-k}$ in (3.61), for $k = 1, \dots, n$. Using [Property 3.3](#), the coefficients $\{\phi_{n1}, \phi_{n2}, \dots, \phi_{nn}\}$ satisfy

$$\mathbb{E}\left[\left(x_{n+1} - \sum_{j=1}^n \phi_{nj}x_{n+1-j}\right)x_{n+1-k}\right] = 0, \quad k = 1, \dots, n,$$

or

$$\sum_{j=1}^n \phi_{nj}\gamma(k-j) = \gamma(k), \quad k = 1, \dots, n. \quad (3.62)$$

The prediction equations (3.62) can be written in matrix notation as

$$\Gamma_n \phi_n = \gamma_n, \quad (3.63)$$

where $\Gamma_n = \{\gamma(k-j)\}_{j,k=1}^n$ is an $n \times n$ matrix, $\phi_n = (\phi_{n1}, \dots, \phi_{nn})'$ is an $n \times 1$ vector, and $\gamma_n = (\gamma(1), \dots, \gamma(n))'$ is an $n \times 1$ vector.

The matrix Γ_n is nonnegative definite. If Γ_n is singular, there are many solutions to (3.63), but, by the Projection Theorem ([Theorem B.1](#)), x_{n+1}^n is unique. If Γ_n is nonsingular, the elements of ϕ_n are unique, and are given by

$$\phi_n = \Gamma_n^{-1} \gamma_n. \quad (3.64)$$

For ARMA models, the fact that $\sigma_w^2 > 0$ and $\gamma(h) \rightarrow 0$ as $h \rightarrow \infty$ is enough to ensure that Γ_n is positive definite ([Problem 3.12](#)). It is sometimes convenient to write the one-step-ahead forecast in vector notation

$$x_{n+1}^n = \phi_n' x, \quad (3.65)$$

where $x = (x_n, x_{n-1}, \dots, x_1)'$.

The *mean square one-step-ahead prediction error* is

$$P_{n+1}^n = \mathbb{E}(x_{n+1} - x_{n+1}^n)^2 = \gamma(0) - \gamma_n' \Gamma_n^{-1} \gamma_n. \quad (3.66)$$

To verify (3.66) using (3.64) and (3.65),

$$\begin{aligned} \mathbb{E}(x_{n+1} - x_{n+1}^n)^2 &= \mathbb{E}(x_{n+1} - \phi_n' x)^2 = \mathbb{E}(x_{n+1} - \gamma_n' \Gamma_n^{-1} x)^2 \\ &= \mathbb{E}(x_{n+1}^2 - 2\gamma_n' \Gamma_n^{-1} x x_{n+1} + \gamma_n' \Gamma_n^{-1} x x' \Gamma_n^{-1} \gamma_n) \\ &= \gamma(0) - 2\gamma_n' \Gamma_n^{-1} \gamma_n + \gamma_n' \Gamma_n^{-1} \Gamma_n \Gamma_n^{-1} \gamma_n \\ &= \gamma(0) - \gamma_n' \Gamma_n^{-1} \gamma_n. \end{aligned}$$

Example 3.19 Prediction for an AR(2)

Suppose we have a causal AR(2) process $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$, and one observation x_1 . Then, using equation (3.64), the one-step-ahead prediction of x_2 based on x_1 is

$$x_2^1 = \phi_{11} x_1 = \frac{\gamma(1)}{\gamma(0)} x_1 = \rho(1)x_1.$$

Now, suppose we want the one-step-ahead prediction of x_3 based on two observations x_1 and x_2 ; i.e., $x_3^2 = \phi_{21}x_2 + \phi_{22}x_1$. We could use (3.62)

$$\begin{aligned}\phi_{21}\gamma(0) + \phi_{22}\gamma(1) &= \gamma(1) \\ \phi_{21}\gamma(1) + \phi_{22}\gamma(0) &= \gamma(2)\end{aligned}$$

to solve for ϕ_{21} and ϕ_{22} , or use the matrix form in (3.64) and solve

$$\begin{pmatrix} \phi_{21} \\ \phi_{22} \end{pmatrix} = \begin{pmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{pmatrix}^{-1} \begin{pmatrix} \gamma(1) \\ \gamma(2) \end{pmatrix},$$

but, it should be apparent from the model that $x_3^2 = \phi_1 x_2 + \phi_2 x_1$. Because $\phi_1 x_2 + \phi_2 x_1$ satisfies the prediction equations (3.60),

$$E\{[x_3 - (\phi_1 x_2 + \phi_2 x_1)]x_1\} = E(w_3 x_1) = 0,$$

$$E\{[x_3 - (\phi_1 x_2 + \phi_2 x_1)]x_2\} = E(w_3 x_2) = 0,$$

it follows that, indeed, $x_3^2 = \phi_1 x_2 + \phi_2 x_1$, and by the uniqueness of the coefficients in this case, that $\phi_{21} = \phi_1$ and $\phi_{22} = \phi_2$. Continuing in this way, it is easy to verify that, for $n \geq 2$,

$$x_{n+1}^n = \phi_1 x_n + \phi_2 x_{n-1}.$$

That is, $\phi_{n1} = \phi_1$, $\phi_{n2} = \phi_2$, and $\phi_{nj} = 0$, for $j = 3, 4, \dots, n$.

From Example 3.19, it should be clear (Problem 3.45) that, if the time series is a causal AR(p) process, then, for $n \geq p$,

$$x_{n+1}^n = \phi_1 x_n + \phi_2 x_{n-1} + \cdots + \phi_p x_{n-p+1}. \quad (3.67)$$

For ARMA models in general, the prediction equations will not be as simple as the pure AR case. In addition, for n large, the use of (3.64) is prohibitive because it requires the inversion of a large matrix. There are, however, iterative solutions that do not require any matrix inversion. In particular, we mention the recursive solution due to Levinson (1947) and Durbin (1960).

Property 3.4 The Durbin–Levinson Algorithm

Equations (3.64) and (3.66) can be solved iteratively as follows:

$$\phi_{00} = 0, \quad P_1^0 = \gamma(0). \quad (3.68)$$

For $n \geq 1$,

$$\phi_{nn} = \frac{\rho(n) - \sum_{k=1}^{n-1} \phi_{n-1,k} \rho(n-k)}{1 - \sum_{k=1}^{n-1} \phi_{n-1,k} \rho(k)}, \quad P_{n+1}^n = P_n^{n-1}(1 - \phi_{nn}^2), \quad (3.69)$$

where, for $n \geq 2$,

$$\phi_{nk} = \phi_{n-1,k} - \phi_{nn} \phi_{n-1,n-k}, \quad k = 1, 2, \dots, n-1. \quad (3.70)$$

The proof of [Property 3.4](#) is left as an exercise; see [Problem 3.13](#).

Example 3.20 Using the Durbin–Levinson Algorithm

To use the algorithm, start with $\phi_{00} = 0$, $P_1^0 = \gamma(0)$. Then, for $n = 1$,

$$\phi_{11} = \rho(1), \quad P_2^1 = \gamma(0)[1 - \phi_{11}^2].$$

For $n = 2$,

$$\begin{aligned} \phi_{22} &= \frac{\rho(2) - \phi_{11} \rho(1)}{1 - \phi_{11} \rho(1)}, \quad \phi_{21} = \phi_{11} - \phi_{22} \phi_{11}, \\ P_3^2 &= P_2^1[1 - \phi_{22}^2] = \gamma(0)[1 - \phi_{11}^2][1 - \phi_{22}^2]. \end{aligned}$$

For $n = 3$,

$$\begin{aligned} \phi_{33} &= \frac{\rho(3) - \phi_{21} \rho(2) - \phi_{22} \rho(1)}{1 - \phi_{21} \rho(1) - \phi_{22} \rho(2)}, \\ \phi_{32} &= \phi_{22} - \phi_{33} \phi_{21}, \quad \phi_{31} = \phi_{21} - \phi_{33} \phi_{22}, \\ P_4^3 &= P_3^2[1 - \phi_{33}^2] = \gamma(0)[1 - \phi_{11}^2][1 - \phi_{22}^2][1 - \phi_{33}^2], \end{aligned}$$

and so on. Note that, in general, the standard error of the one-step-ahead forecast is the square root of

$$P_{n+1}^n = \gamma(0) \prod_{j=1}^n [1 - \phi_{jj}^2]. \quad (3.71)$$

An important consequence of the Durbin–Levinson algorithm is (see [Problem 3.13](#)) as follows.

Property 3.5 Iterative Solution for the PACF

The PACF of a stationary process x_t , can be obtained iteratively via (3.69) as ϕ_{nn} , for $n = 1, 2, \dots$

Using [Property 3.5](#) and putting $n = p$ in (3.61) and (3.67), it follows that for an AR(p) model,

$$\begin{aligned} x_{p+1}^p &= \phi_{p1} x_p + \phi_{p2} x_{p-1} + \cdots + \phi_{pp} x_1 \\ &= \phi_1 x_p + \phi_2 x_{p-1} + \cdots + \phi_p x_1. \end{aligned} \quad (3.72)$$

Result (3.72) shows that for an AR(p) model, the partial autocorrelation coefficient at lag p , ϕ_{pp} , is also the last coefficient in the model, ϕ_p , as was claimed in [Example 3.16](#).

Example 3.21 The PACF of an AR(2)

We will use the results of [Example 3.20](#) and [Property 3.5](#) to calculate the first three values, ϕ_{11} , ϕ_{22} , ϕ_{33} , of the PACF. Recall from [Example 3.10](#) that $\rho(h) - \phi_1\rho(h-1) - \phi_2\rho(h-2) = 0$ for $h \geq 1$. When $h = 1, 2, 3$, we have $\rho(1) = \phi_1/(1-\phi_2)$, $\rho(2) = \phi_1\rho(1) + \phi_2$, $\rho(3) - \phi_1\rho(2) - \phi_2\rho(1) = 0$. Thus,

$$\begin{aligned}\phi_{11} &= \rho(1) = \frac{\phi_1}{1-\phi_2} \\ \phi_{22} &= \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2} = \frac{\left[\phi_1\left(\frac{\phi_1}{1-\phi_2}\right) + \phi_2\right] - \left(\frac{\phi_1}{1-\phi_2}\right)^2}{1 - \left(\frac{\phi_1}{1-\phi_2}\right)^2} = \phi_2 \\ \phi_{21} &= \rho(1)[1 - \phi_2] = \phi_1 \\ \phi_{33} &= \frac{\rho(3) - \phi_1\rho(2) - \phi_2\rho(1)}{1 - \phi_1\rho(1) - \phi_2\rho(2)} = 0.\end{aligned}$$

Notice that, as shown in [\(3.72\)](#), $\phi_{22} = \phi_2$ for an AR(2) model.

So far, we have concentrated on one-step-ahead prediction, but [Property 3.3](#) allows us to calculate the BLP of x_{n+m} for any $m \geq 1$. Given data, $\{x_1, \dots, x_n\}$, the m -step-ahead predictor is

$$x_{n+m}^n = \phi_{n1}^{(m)} x_n + \phi_{n2}^{(m)} x_{n-1} + \dots + \phi_{nn}^{(m)} x_1, \quad (3.73)$$

where $\{\phi_{n1}^{(m)}, \phi_{n2}^{(m)}, \dots, \phi_{nn}^{(m)}\}$ satisfy the prediction equations,

$$\sum_{j=1}^n \phi_{nj}^{(m)} E(x_{n+1-j} x_{n+1-k}) = E(x_{n+m} x_{n+1-k}), \quad k = 1, \dots, n,$$

or

$$\sum_{j=1}^n \phi_{nj}^{(m)} \gamma(k-j) = \gamma(m+k-1), \quad k = 1, \dots, n. \quad (3.74)$$

The prediction equations can again be written in matrix notation as

$$\Gamma_n \phi_n^{(m)} = \gamma_n^{(m)}, \quad (3.75)$$

where $\gamma_n^{(m)} = (\gamma(m), \dots, \gamma(m+n-1))'$, and $\phi_n^{(m)} = (\phi_{n1}^{(m)}, \dots, \phi_{nn}^{(m)})'$ are $n \times 1$ vectors. The *mean square m-step-ahead prediction error* is

$$P_{n+m}^n = E(x_{n+m} - x_{n+m}^n)^2 = \gamma(0) - \gamma_n^{(m)'} \Gamma_n^{-1} \gamma_n^{(m)}. \quad (3.76)$$

Another useful algorithm for calculating forecasts was given by Brockwell and Davis (1991, Chapter 5). This algorithm follows directly from applying the projection theorem ([Theorem B.1](#)) to the *innovations*, $x_t - x_t^{t-1}$, for $t = 1, \dots, n$, using the fact that the innovations $x_t - x_t^{t-1}$ and $x_s - x_s^{s-1}$ are uncorrelated for $s \neq t$ (see [Problem 3.46](#)). We present the case in which x_t is a mean-zero stationary time series.

Property 3.6 The Innovations Algorithm

The one-step-ahead predictors, x_{t+1}^t , and their mean-squared errors, P_{t+1}^t , can be calculated iteratively as

$$\begin{aligned} x_1^0 &= 0, \quad P_1^0 = \gamma(0) \\ x_{t+1}^t &= \sum_{j=1}^t \theta_{tj}(x_{t+1-j} - x_{t+1-j}^{t-j}), \quad t = 1, 2, \dots \end{aligned} \quad (3.77)$$

$$P_{t+1}^t = \gamma(0) - \sum_{j=0}^{t-1} \theta_{t,t-j}^2 P_{j+1}^j \quad t = 1, 2, \dots, \quad (3.78)$$

where, for $j = 0, 1, \dots, t-1$,

$$\theta_{t,t-j} = \left(\gamma(t-j) - \sum_{k=0}^{j-1} \theta_{j,j-k} \theta_{t,t-k} P_{k+1}^k \right) / P_{j+1}^j. \quad (3.79)$$

Given data x_1, \dots, x_n , the innovations algorithm can be calculated successively for $t = 1$, then $t = 2$ and so on, in which case the calculation of x_{n+1}^n and P_{n+1}^n is made at the final step $t = n$. The m -step-ahead predictor and its mean-square error based on the innovations algorithm (Problem 3.46) are given by

$$x_{n+m}^n = \sum_{j=m}^{n+m-1} \theta_{n+m-1,j}(x_{n+m-j} - x_{n+m-j}^{n+m-j-1}), \quad (3.80)$$

$$P_{n+m}^n = \gamma(0) - \sum_{j=m}^{n+m-1} \theta_{n+m-1,j}^2 P_{n+m-j}^{n+m-j-1}, \quad (3.81)$$

where the $\theta_{n+m-1,j}$ are obtained by continued iteration of (3.79).

Example 3.22 Prediction for an MA(1)

The innovations algorithm lends itself well to prediction for moving average processes. Consider an MA(1) model, $x_t = w_t + \theta w_{t-1}$. Recall that $\gamma(0) = (1 + \theta^2)\sigma_w^2$, $\gamma(1) = \theta\sigma_w^2$, and $\gamma(h) = 0$ for $h > 1$. Then, using Property 3.6, we have

$$\begin{aligned} \theta_{n1} &= \theta\sigma_w^2 / P_n^{n-1} \\ \theta_{nj} &= 0, \quad j = 2, \dots, n \\ P_1^0 &= (1 + \theta^2)\sigma_w^2 \\ P_{n+1}^n &= (1 + \theta^2 - \theta\theta_{n1})\sigma_w^2. \end{aligned}$$

Finally, from (3.77), the one-step-ahead predictor is

$$x_{n+1}^n = \theta \left(x_n - x_n^{n-1} \right) \sigma_w^2 / P_n^{n-1}.$$

FORECASTING ARMA PROCESSES

The general prediction equations (3.60) provide little insight into forecasting for ARMA models in general. There are a number of different ways to express these forecasts, and each aids in understanding the special structure of ARMA prediction. Throughout, we assume x_t is a causal and invertible ARMA(p, q) process, $\phi(B)x_t = \theta(B)w_t$, where $w_t \sim \text{iid } N(0, \sigma_w^2)$. In the non-zero mean case, $E(x_t) = \mu_x$, simply replace x_t with $x_t - \mu_x$ in the model. First, we consider two types of forecasts. We write x_{n+m}^n to mean the minimum mean square error predictor of x_{n+m} based on the data $\{x_n, \dots, x_1\}$, that is,

$$x_{n+m}^n = E(x_{n+m} \mid x_n, \dots, x_1).$$

For ARMA models, it is easier to calculate the predictor of x_{n+m} , assuming we have the complete history of the process $\{x_n, x_{n-1}, \dots, x_1, x_0, x_{-1}, \dots\}$. We will denote the predictor of x_{n+m} based on the infinite past as

$$\tilde{x}_{n+m} = E(x_{n+m} \mid x_n, x_{n-1}, \dots, x_1, x_0, x_{-1}, \dots).$$

In general, x_{n+m}^n and \tilde{x}_{n+m} are not the same, but the idea here is that, for large samples, \tilde{x}_{n+m} will provide a good approximation to x_{n+m}^n .

Now, write x_{n+m} in its causal and invertible forms:

$$x_{n+m} = \sum_{j=0}^{\infty} \psi_j w_{n+m-j}, \quad \psi_0 = 1 \quad (3.82)$$

$$w_{n+m} = \sum_{j=0}^{\infty} \pi_j x_{n+m-j}, \quad \pi_0 = 1. \quad (3.83)$$

Then, taking conditional expectations in (3.82), we have

$$\tilde{x}_{n+m} = \sum_{j=0}^{\infty} \psi_j \tilde{w}_{n+m-j} = \sum_{j=m}^{\infty} \psi_j w_{n+m-j}, \quad (3.84)$$

because, by causality and invertibility,

$$\tilde{w}_t = E(w_t \mid x_n, x_{n-1}, \dots, x_0, x_{-1}, \dots) = \begin{cases} 0 & t > n \\ w_t & t \leq n. \end{cases}$$

Similarly, taking conditional expectations in (3.83), we have

$$0 = \tilde{x}_{n+m} + \sum_{j=1}^{\infty} \pi_j \tilde{x}_{n+m-j},$$

or

$$\tilde{x}_{n+m} = - \sum_{j=1}^{m-1} \pi_j \tilde{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j x_{n+m-j}, \quad (3.85)$$

using the fact $E(x_t \mid x_n, x_{n-1}, \dots, x_0, x_{-1}, \dots) = x_t$, for $t \leq n$. Prediction is accomplished recursively using (3.85), starting with the one-step-ahead predictor, $m = 1$, and then continuing for $m = 2, 3, \dots$. Using (3.84), we can write

$$x_{n+m} - \tilde{x}_{n+m} = \sum_{j=0}^{m-1} \psi_j w_{n+m-j},$$

so the *mean-square prediction error* can be written as

$$P_{n+m}^n = E(x_{n+m} - \tilde{x}_{n+m})^2 = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2. \quad (3.86)$$

Also, we note, for a fixed sample size, n , the prediction errors are correlated. That is, for $k \geq 1$,

$$E\{(x_{n+m} - \tilde{x}_{n+m})(x_{n+m+k} - \tilde{x}_{n+m+k})\} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j \psi_{j+k}. \quad (3.87)$$

Example 3.23 Long-Range Forecasts

Consider forecasting an ARMA process with mean μ_x . Replacing x_{n+m} with $x_{n+m} - \mu_x$ in (3.82), and taking conditional expectation as in (3.84), we deduce that the m -step-ahead forecast can be written as

$$\tilde{x}_{n+m} = \mu_x + \sum_{j=m}^{\infty} \psi_j w_{n+m-j}. \quad (3.88)$$

Noting that the ψ -weights dampen to zero exponentially fast, it is clear that

$$\tilde{x}_{n+m} \rightarrow \mu_x \quad (3.89)$$

exponentially fast (in the mean square sense) as $m \rightarrow \infty$. Moreover, by (3.86), the mean square prediction error

$$P_{n+m}^n \rightarrow \sigma_w^2 \sum_{j=0}^{\infty} \psi_j^2 = \gamma_x(0) = \sigma_x^2, \quad (3.90)$$

exponentially fast as $m \rightarrow \infty$.

It should be clear from (3.89) and (3.90) that ARMA forecasts quickly settle to the mean with a constant prediction error as the forecast horizon, m , grows. This effect can be seen in Figure 3.7 where the Recruitment series is forecast for 24 months; see Example 3.25.

When n is small, the general prediction equations (3.60) can be used easily. When n is large, we would use (3.85) by truncating, because we do not observe

x_0, x_1, x_2, \dots , and only the data x_1, x_2, \dots, x_n are available. In this case, we can truncate (3.85) by setting $\sum_{j=n+m}^{\infty} \pi_j x_{n+m-j} = 0$. The *truncated predictor* is then written as

$$\tilde{x}_{n+m}^n = - \sum_{j=1}^{m-1} \pi_j \tilde{x}_{n+m-j}^n - \sum_{j=m}^{n+m-1} \pi_j x_{n+m-j}, \quad (3.91)$$

which is also calculated recursively, $m = 1, 2, \dots$. The mean square prediction error, in this case, is approximated using (3.86).

For AR(p) models, and when $n > p$, equation (3.67) yields the exact predictor, x_{n+m}^n , of x_{n+m} , and there is no need for approximations. That is, for $n > p$, $\tilde{x}_{n+m}^n = \tilde{x}_{n+m} = x_{n+m}^n$. Also, in this case, the one-step-ahead prediction error is $E(x_{n+1} - x_{n+1}^n)^2 = \sigma_w^2$. For pure MA(q) or ARMA(p, q) models, truncated prediction has a fairly nice form.

Property 3.7 Truncated Prediction for ARMA

For ARMA(p, q) models, the truncated predictors for $m = 1, 2, \dots$, are

$$\tilde{x}_{n+m}^n = \phi_1 \tilde{x}_{n+m-1}^n + \cdots + \phi_p \tilde{x}_{n+m-p}^n + \theta_1 \tilde{w}_{n+m-1}^n + \cdots + \theta_q \tilde{w}_{n+m-q}^n, \quad (3.92)$$

where $\tilde{x}_t^n = x_t$ for $1 \leq t \leq n$ and $\tilde{x}_t^n = 0$ for $t \leq 0$. The truncated prediction errors are given by: $\tilde{w}_t^n = 0$ for $t \leq 0$ or $t > n$, and

$$\tilde{w}_t^n = \phi(B) \tilde{x}_t^n - \theta_1 \tilde{w}_{t-1}^n - \cdots - \theta_q \tilde{w}_{t-q}^n$$

for $1 \leq t \leq n$.

Example 3.24 Forecasting an ARMA(1, 1) Series

Given data x_1, \dots, x_n , for forecasting purposes, write the model as

$$x_{n+1} = \phi x_n + w_{n+1} + \theta w_n.$$

Then, based on (3.92), the one-step-ahead truncated forecast is

$$\tilde{x}_{n+1}^n = \phi x_n + 0 + \theta \tilde{w}_n^n.$$

For $m \geq 2$, we have

$$\tilde{x}_{n+m}^n = \phi \tilde{x}_{n+m-1}^n,$$

which can be calculated recursively, $m = 2, 3, \dots$.

To calculate \tilde{w}_n^n , which is needed to initialize the successive forecasts, the model can be written as $w_t = x_t - \phi x_{t-1} - \theta w_{t-1}$ for $t = 1, \dots, n$. For truncated forecasting using (3.92), put $\tilde{w}_0^n = 0$, $x_0 = 0$, and then iterate the errors forward in time

$$\tilde{w}_t^n = x_t - \phi x_{t-1} - \theta \tilde{w}_{t-1}^n, \quad t = 1, \dots, n.$$

The approximate forecast variance is computed from (3.86) using the ψ -weights determined as in Example 3.12. In particular, the ψ -weights satisfy $\psi_j = (\phi + \theta)\phi^{j-1}$, for $j \geq 1$. This result gives

$$P_{n+m}^n = \sigma_w^2 \left[1 + (\phi + \theta)^2 \sum_{j=1}^{m-1} \phi^{2(j-1)} \right] = \sigma_w^2 \left[1 + \frac{(\phi + \theta)^2 (1 - \phi^{2(m-1)})}{(1 - \phi^2)} \right].$$

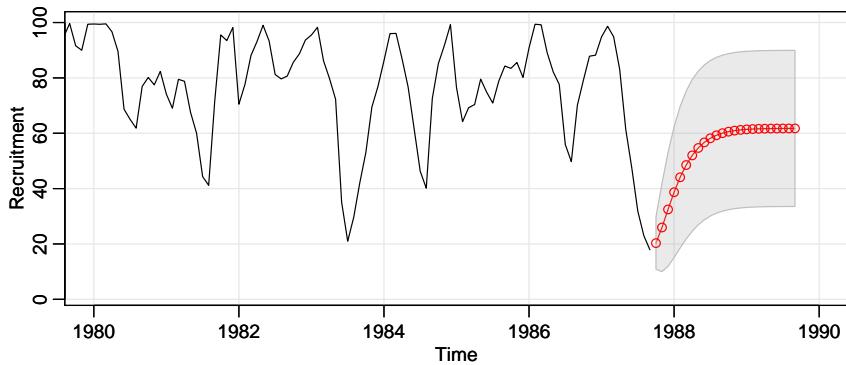


Fig. 3.7. Twenty-four month forecasts for the Recruitment series. The actual data shown are from about January 1980 to September 1987, and then the forecasts plus and minus one standard error are displayed.

To assess the precision of the forecasts, *prediction intervals* are typically calculated along with the forecasts. In general, $(1 - \alpha)$ prediction intervals are of the form

$$x_{n+m}^n \pm c_{\frac{\alpha}{2}} \sqrt{P_{n+m}^n}, \quad (3.93)$$

where $c_{\alpha/2}$ is chosen to get the desired degree of confidence. For example, if the process is Gaussian, then choosing $c_{\alpha/2} = 2$ will yield an approximate 95% prediction interval for x_{n+m} . If we are interested in establishing prediction intervals over more than one time period, then $c_{\alpha/2}$ should be adjusted appropriately, for example, by using Bonferroni's inequality [see (4.63) in Chapter 4 or Johnson and Wichern, 1992, Chapter 5].

Example 3.25 Forecasting the Recruitment Series

Using the parameter estimates as the actual parameter values, Figure 3.7 shows the result of forecasting the Recruitment series given in Example 3.18 over a 24-month horizon, $m = 1, 2, \dots, 24$. The actual forecasts are calculated as

$$x_{n+m}^n = 6.74 + 1.35x_{n+m-1}^n - .46x_{n+m-2}^n$$

for $n = 453$ and $m = 1, 2, \dots, 12$. Recall that $x_t^s = x_t$ when $t \leq s$. The forecasts errors P_{n+m}^n are calculated using (3.86). Recall that $\hat{\sigma}_w^2 = 89.72$, and using (3.40) from Example 3.12, we have $\psi_j = 1.35\psi_{j-1} - .46\psi_{j-2}$ for $j \geq 2$, where $\psi_0 = 1$ and $\psi_1 = 1.35$. Thus, for $n = 453$,

$$\begin{aligned} P_{n+1}^n &= 89.72, \\ P_{n+2}^n &= 89.72(1 + 1.35^2), \\ P_{n+3}^n &= 89.72(1 + 1.35^2 + [1.35^2 - .46]^2), \end{aligned}$$

and so on.

Note how the forecast levels off quickly and the prediction intervals are wide, even though in this case the forecast limits are only based on one standard error; that is, $x_{n+m}^n \pm \sqrt{P_{n+m}^n}$.

To reproduce the analysis and Figure 3.7, use the following commands:

```
regr = ar.ols(rec, order=2, demean=FALSE, intercept=TRUE)
fore = predict(regr, n.ahead=24)
ts.plot(rec, fore$pred, col=1:2, xlim=c(1980, 1990), ylab="Recruitment")
U = fore$pred+fore$se; L = fore$pred-fore$se
xx = c(time(U), rev(time(U))); yy = c(L, rev(U))
polygon(xx, yy, border = 8, col = gray(.6, alpha = .2))
lines(fore$pred, type="p", col=2)
```

We complete this section with a brief discussion of *backcasting*. In backcasting, we want to predict x_{1-m} , for $m = 1, 2, \dots$, based on the data $\{x_1, \dots, x_n\}$. Write the backcast as

$$x_{1-m}^n = \sum_{j=1}^n \alpha_j x_j. \quad (3.94)$$

Analogous to (3.74), the prediction equations (assuming $\mu_x = 0$) are

$$\sum_{j=1}^n \alpha_j E(x_j x_k) = E(x_{1-m} x_k), \quad k = 1, \dots, n, \quad (3.95)$$

or

$$\sum_{j=1}^n \alpha_j \gamma(k-j) = \gamma(m+k-1), \quad k = 1, \dots, n. \quad (3.96)$$

These equations are precisely the prediction equations for forward prediction. That is, $\alpha_j \equiv \phi_{nj}^{(m)}$, for $j = 1, \dots, n$, where the $\phi_{nj}^{(m)}$ are given by (3.75). Finally, the backcasts are given by

$$x_{1-m}^n = \phi_{n1}^{(m)} x_1 + \dots + \phi_{nn}^{(m)} x_n, \quad m = 1, 2, \dots. \quad (3.97)$$

Example 3.26 Backcasting an ARMA(1, 1)

Consider an ARMA(1, 1) process, $x_t = \phi x_{t-1} + \theta w_t + w_t$; we will call this the *forward model*. We have just seen that best linear prediction backward in time is the same as best linear prediction forward in time for stationary models. Assuming the models are Gaussian, we also have that minimum mean square error prediction backward in time is the same as forward in time for ARMA models.^{3.4} Thus, the process can equivalently be generated by the *backward model*,

$$x_t = \phi x_{t+1} + \theta v_{t+1} + v_t,$$

^{3.4} In the stationary Gaussian case, (a) the distribution of $\{x_{n+1}, x_n, \dots, x_1\}$ is the same as (b) the distribution of $\{x_0, x_1, \dots, x_n\}$. In forecasting we use (a) to obtain $E(x_{n+1} | x_n, \dots, x_1)$; in backcasting we use (b) to obtain $E(x_0 | x_1, \dots, x_n)$. Because (a) and (b) are the same, the two problems are equivalent.

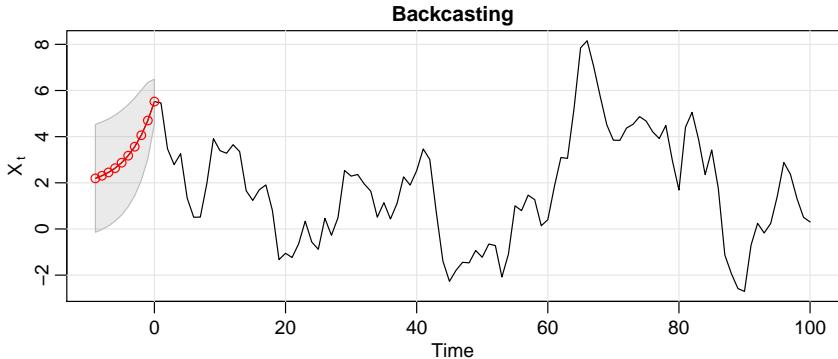


Fig. 3.8. Display for [Example 3.26](#); backcasts from a simulated ARMA(1, 1).

where $\{v_t\}$ is a Gaussian white noise process with variance σ_w^2 . We may write $x_t = \sum_{j=0}^{\infty} \psi_j v_{t+j}$, where $\psi_0 = 1$; this means that x_t is uncorrelated with $\{v_{t-1}, v_{t-2}, \dots\}$, in analogy to the forward model.

Given data $\{x_1, \dots, x_n\}$, truncate $v_n^n = E(v_n | x_1, \dots, x_n)$ to zero and then iterate backward. That is, put $\tilde{v}_n^n = 0$, as an initial approximation, and then generate the errors backward

$$\tilde{v}_t^n = x_t - \phi x_{t+1} - \theta \tilde{v}_{t+1}^n, \quad t = (n-1), (n-2), \dots, 1.$$

Then,

$$\tilde{x}_0^n = \phi x_1 + \theta \tilde{v}_1^n + \tilde{v}_0^n = \phi x_1 + \theta \tilde{v}_1^n,$$

because $\tilde{v}_t^n = 0$ for $t \leq 0$. Continuing, the general truncated backcasts are given by

$$\tilde{x}_{1-m}^n = \phi \tilde{x}_{2-m}^n, \quad m = 2, 3, \dots.$$

To backcast data in R, simply reverse the data, fit the model and predict. In the following, we backcasted a simulated ARMA(1,1) process; see [Figure 3.8](#).

```
set.seed(90210)
x      = arima.sim(list(order = c(1,0,1), ar = .9, ma=.5), n = 100)
xr     = rev(x)                                # xr is the reversed data
pxr   = predict(arima(xr, order=c(1,0,1)), 10)  # predict the reversed data
pxrp  = rev(pxr$pred)                          # reorder the predictors (for plotting)
pxrse = rev(pxr$se)                            # reorder the SEs
nx    = ts(c(pxrp, x), start=-9)              # attach the backcasts to the data
plot(nx, ylab=expression(X[-t]), main='Backcasting')
U = nx[1:10] + pxrse; L = nx[1:10] - pxrse
xx = c(-9:0, 0:-9); yy = c(L, rev(U))
polygon(xx, yy, border = 8, col = gray(0.6, alpha = 0.2))
lines(-9:0, nx[1:10], col=2, type='o')
```

3.5 Estimation

Throughout this section, we assume we have n observations, x_1, \dots, x_n , from a causal and invertible Gaussian ARMA(p, q) process in which, initially, the order parameters, p and q , are known. Our goal is to estimate the parameters, $\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$, and σ_w^2 . We will discuss the problem of determining p and q later in this section.

We begin with *method of moments* estimators. The idea behind these estimators is that of equating population moments to sample moments and then solving for the parameters in terms of the sample moments. We immediately see that, if $E(x_t) = \mu$, then the method of moments estimator of μ is the sample average, \bar{x} . Thus, while discussing method of moments, we will assume $\mu = 0$. Although the method of moments can produce good estimators, they can sometimes lead to suboptimal estimators. We first consider the case in which the method leads to optimal (efficient) estimators, that is, AR(p) models,

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t,$$

where the first $p + 1$ equations of (3.47) and (3.48) lead to the following:

Definition 3.10 *The Yule–Walker equations are given by*

$$\gamma(h) = \phi_1 \gamma(h-1) + \cdots + \phi_p \gamma(h-p), \quad h = 1, 2, \dots, p, \quad (3.98)$$

$$\sigma_w^2 = \gamma(0) - \phi_1 \gamma(1) - \cdots - \phi_p \gamma(p). \quad (3.99)$$

In matrix notation, the Yule–Walker equations are

$$\Gamma_p \phi = \gamma_p, \quad \sigma_w^2 = \gamma(0) - \phi' \gamma_p, \quad (3.100)$$

where $\Gamma_p = \{\gamma(k-j)\}_{j,k=1}^p$ is a $p \times p$ matrix, $\phi = (\phi_1, \dots, \phi_p)'$ is a $p \times 1$ vector, and $\gamma_p = (\gamma(1), \dots, \gamma(p))'$ is a $p \times 1$ vector. Using the method of moments, we replace $\gamma(h)$ in (3.100) by $\hat{\gamma}(h)$ [see equation (1.36)] and solve

$$\hat{\phi} = \hat{I}_p^{-1} \hat{\gamma}_p, \quad \hat{\sigma}_w^2 = \hat{\gamma}(0) - \hat{\gamma}'_p \hat{I}_p^{-1} \hat{\gamma}_p. \quad (3.101)$$

These estimators are typically called the *Yule–Walker estimators*. For calculation purposes, it is sometimes more convenient to work with the sample ACF. By factoring $\hat{\gamma}(0)$ in (3.101), we can write the Yule–Walker estimates as

$$\hat{\phi} = \hat{R}_p^{-1} \hat{\rho}_p, \quad \hat{\sigma}_w^2 = \hat{\gamma}(0) [1 - \hat{\rho}'_p \hat{R}_p^{-1} \hat{\rho}_p], \quad (3.102)$$

where $\hat{R}_p = \{\hat{\rho}(k-j)\}_{j,k=1}^p$ is a $p \times p$ matrix and $\hat{\rho}_p = (\hat{\rho}(1), \dots, \hat{\rho}(p))'$ is a $p \times 1$ vector.

For AR(p) models, if the sample size is large, the Yule–Walker estimators are approximately normally distributed, and $\hat{\sigma}_w^2$ is close to the true value of σ_w^2 . We state these results in [Property 3.8](#); for details, see [Section B.3](#).

Property 3.8 Large Sample Results for Yule–Walker Estimators

The asymptotic ($n \rightarrow \infty$) behavior of the Yule–Walker estimators in the case of causal AR(p) processes is as follows:

$$\sqrt{n} (\hat{\phi} - \phi) \xrightarrow{d} N\left(0, \sigma_w^2 \Gamma_p^{-1}\right), \quad \hat{\sigma}_w^2 \xrightarrow{P} \sigma_w^2. \quad (3.103)$$

The Durbin–Levinson algorithm, (3.68)–(3.70), can be used to calculate $\hat{\phi}$ without inverting $\hat{\Gamma}_p$ or \hat{R}_p , by replacing $\gamma(h)$ by $\hat{\gamma}(h)$ in the algorithm. In running the algorithm, we will iteratively calculate the $h \times 1$ vector, $\hat{\phi}_h = (\hat{\phi}_{h1}, \dots, \hat{\phi}_{hh})'$, for $h = 1, 2, \dots$. Thus, in addition to obtaining the desired forecasts, the Durbin–Levinson algorithm yields $\hat{\phi}_{hh}$, the sample PACF. Using (3.103), we can show the following property.

Property 3.9 Large Sample Distribution of the PACF

For a causal AR(p) process, asymptotically ($n \rightarrow \infty$),

$$\sqrt{n} \hat{\phi}_{hh} \xrightarrow{d} N(0, 1), \quad \text{for } h > p. \quad (3.104)$$

Example 3.27 Yule–Walker Estimation for an AR(2) Process

The data shown in Figure 3.4 were $n = 144$ simulated observations from the AR(2) model

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t,$$

where $w_t \sim \text{iid } N(0, 1)$. For these data, $\hat{\gamma}(0) = 8.903$, $\hat{\rho}(1) = .849$, and $\hat{\rho}(2) = .519$. Thus,

$$\hat{\phi} = \begin{pmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \end{pmatrix} = \begin{bmatrix} 1 & .849 \\ .849 & 1 \end{bmatrix}^{-1} \begin{pmatrix} .849 \\ .519 \end{pmatrix} = \begin{pmatrix} 1.463 \\ -.723 \end{pmatrix}$$

and

$$\hat{\sigma}_w^2 = 8.903 \left[1 - (.849, .519) \begin{pmatrix} 1.463 \\ -.723 \end{pmatrix} \right] = 1.187.$$

By Property 3.8, the asymptotic variance–covariance matrix of $\hat{\phi}$ is

$$\frac{1}{144} \frac{1.187}{8.903} \begin{bmatrix} 1 & .849 \\ .849 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} .058^2 & -.003 \\ -.003 & .058^2 \end{bmatrix},$$

and it can be used to get confidence regions for, or make inferences about $\hat{\phi}$ and its components. For example, an approximate 95% confidence interval for ϕ_2 is $-.723 \pm 2(.058)$, or $(-.838, -.608)$, which contains the true value of $\phi_2 = -.75$.

For these data, the first three sample partial autocorrelations are $\hat{\phi}_{11} = \hat{\rho}(1) = .849$, $\hat{\phi}_{22} = \hat{\phi}_2 = -.721$, and $\hat{\phi}_{33} = -.085$. According to Property 3.9, the asymptotic standard error of $\hat{\phi}_{33}$ is $1/\sqrt{144} = .083$, and the observed value, $-.085$, is about only one standard deviation from $\phi_{33} = 0$.

Example 3.28 Yule–Walker Estimation of the Recruitment Series

In Example 3.18 we fit an AR(2) model to the recruitment series using ordinary least squares (OLS). For AR models, the estimators obtained via OLS and Yule–Walker are nearly identical; we will see this when we discuss conditional sum of squares estimation in (3.111)–(3.116).

Below are the results of fitting the same model using Yule–Walker estimation in R, which are nearly identical to the values in Example 3.18.

```
rec.yw = ar.yw(rec, order=2)
rec.yw$x.mean    # = 62.26 (mean estimate)
rec.yw$ar         # = 1.33, -.44 (coefficient estimates)
sqrt(diag(rec.yw$asy.var.coef)) # = .04, .04 (standard errors)
rec.yw$var.pred   # = 94.80 (error variance estimate)
```

To obtain the 24 month ahead predictions and their standard errors, and then plot the results (not shown) as in Example 3.25, use the R commands:

```
rec.pr = predict(rec.yw, n.ahead=24)
ts.plot(rec, rec.pr$pred, col=1:2)
lines(rec.pr$pred + rec.pr$se, col=4, lty=2)
lines(rec.pr$pred - rec.pr$se, col=4, lty=2)
```

In the case of AR(p) models, the Yule–Walker estimators given in (3.102) are optimal in the sense that the asymptotic distribution, (3.103), is the best asymptotic normal distribution. This is because, given initial conditions, AR(p) models are linear models, and the Yule–Walker estimators are essentially least squares estimators. If we use method of moments for MA or ARMA models, we will not get optimal estimators because such processes are nonlinear in the parameters.

Example 3.29 Method of Moments Estimation for an MA(1)

Consider the time series

$$x_t = w_t + \theta w_{t-1},$$

where $|\theta| < 1$. The model can then be written as

$$x_t = \sum_{j=1}^{\infty} (-\theta)^j x_{t-j} + w_t,$$

which is nonlinear in θ . The first two population autocovariances are $\gamma(0) = \sigma_w^2(1 + \theta^2)$ and $\gamma(1) = \sigma_w^2\theta$, so the estimate of θ is found by solving:

$$\hat{\rho}(1) = \frac{\hat{\gamma}(1)}{\hat{\gamma}(0)} = \frac{\hat{\theta}}{1 + \hat{\theta}^2}.$$

Two solutions exist, so we would pick the invertible one. If $|\hat{\rho}(1)| \leq \frac{1}{2}$, the solutions are real, otherwise, a real solution does not exist. Even though $|\rho(1)| < \frac{1}{2}$ for an invertible MA(1), it may happen that $|\hat{\rho}(1)| \geq \frac{1}{2}$ because it is an estimator. For example, the following simulation in R produces a value of $\hat{\rho}(1) = .507$ when the true value is $\rho(1) = .9/(1 + .9^2) = .497$.

```
set.seed(2)
ma1 = arima.sim(list(order = c(0,0,1), ma = 0.9), n = 50)
acf(ma1, plot=FALSE)[1] # = .507 (lag 1 sample ACF)
```

When $|\hat{\rho}(1)| < \frac{1}{2}$, the invertible estimate is

$$\hat{\theta} = \frac{1 - \sqrt{1 - 4\hat{\rho}(1)^2}}{2\hat{\rho}(1)}. \quad (3.105)$$

It can be shown that^{3.5}

$$\hat{\theta} \sim \text{AN}\left(\theta, \frac{1 + \theta^2 + 4\theta^4 + \theta^6 + \theta^8}{n(1 - \theta^2)^2}\right);$$

AN is read *asymptotically normal* and is defined in [Definition A.5](#). The maximum likelihood estimator (which we discuss next) of θ , in this case, has an asymptotic variance of $(1 - \theta^2)/n$. When $\theta = .5$, for example, the ratio of the asymptotic variance of the method of moments estimator to the maximum likelihood estimator of θ is about 3.5. That is, for large samples, the variance of the method of moments estimator is about 3.5 times larger than the variance of the MLE of θ when $\theta = .5$.

MAXIMUM LIKELIHOOD AND LEAST SQUARES ESTIMATION

To fix ideas, we first focus on the causal AR(1) case. Let

$$x_t = \mu + \phi(x_{t-1} - \mu) + w_t \quad (3.106)$$

where $|\phi| < 1$ and $w_t \sim \text{iid } N(0, \sigma_w^2)$. Given data x_1, x_2, \dots, x_n , we seek the likelihood

$$L(\mu, \phi, \sigma_w^2) = f\left(x_1, x_2, \dots, x_n \mid \mu, \phi, \sigma_w^2\right).$$

In the case of an AR(1), we may write the likelihood as

$$L(\mu, \phi, \sigma_w^2) = f(x_1)f(x_2 \mid x_1) \cdots f(x_n \mid x_{n-1}),$$

where we have dropped the parameters in the densities, $f(\cdot)$, to ease the notation. Because $x_t \mid x_{t-1} \sim N(\mu + \phi(x_{t-1} - \mu), \sigma_w^2)$, we have

$$f(x_t \mid x_{t-1}) = f_w[(x_t - \mu) - \phi(x_{t-1} - \mu)],$$

where $f_w(\cdot)$ is the density of w_t , that is, the normal density with mean zero and variance σ_w^2 . We may then write the likelihood as

$$L(\mu, \phi, \sigma_w^2) = f(x_1) \prod_{t=2}^n f_w[(x_t - \mu) - \phi(x_{t-1} - \mu)].$$

^{3.5} The result follows from [Theorem A.7](#) and the delta method. See the proof of [Theorem A.7](#) for details on the delta method.

To find $f(x_1)$, we can use the causal representation

$$x_1 = \mu + \sum_{j=0}^{\infty} \phi^j w_{1-j}$$

to see that x_1 is normal, with mean μ and variance $\sigma_w^2/(1-\phi^2)$. Finally, for an AR(1), the likelihood is

$$L(\mu, \phi, \sigma_w^2) = (2\pi\sigma_w^2)^{-n/2} (1-\phi^2)^{1/2} \exp \left[-\frac{S(\mu, \phi)}{2\sigma_w^2} \right], \quad (3.107)$$

where

$$S(\mu, \phi) = (1-\phi^2)(x_1 - \mu)^2 + \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2. \quad (3.108)$$

Typically, $S(\mu, \phi)$ is called the *unconditional sum of squares*. We could have also considered the estimation of μ and ϕ using *unconditional least squares*, that is, estimation by minimizing $S(\mu, \phi)$.

Taking the partial derivative of the log of (3.107) with respect to σ_w^2 and setting the result equal to zero, we get the typical normal result that for any given values of μ and ϕ in the parameter space, $\sigma_w^2 = n^{-1}S(\mu, \phi)$ maximizes the likelihood. Thus, the maximum likelihood estimate of σ_w^2 is

$$\hat{\sigma}_w^2 = n^{-1}S(\hat{\mu}, \hat{\phi}), \quad (3.109)$$

where $\hat{\mu}$ and $\hat{\phi}$ are the MLEs of μ and ϕ , respectively. If we replace n in (3.109) by $n-2$, we would obtain the unconditional least squares estimate of σ_w^2 .

If, in (3.107), we take logs, replace σ_w^2 by $\hat{\sigma}_w^2$, and ignore constants, $\hat{\mu}$ and $\hat{\phi}$ are the values that minimize the criterion function

$$l(\mu, \phi) = \log [n^{-1}S(\mu, \phi)] - n^{-1}\log(1-\phi^2); \quad (3.110)$$

that is, $l(\mu, \phi) \propto -2\log L(\mu, \phi, \hat{\sigma}_w^2)$.^{3.6} Because (3.108) and (3.110) are complicated functions of the parameters, the minimization of $l(\mu, \phi)$ or $S(\mu, \phi)$ is accomplished numerically. In the case of AR models, we have the advantage that, conditional on initial values, they are linear models. That is, we can drop the term in the likelihood that causes the nonlinearity. Conditioning on x_1 , the *conditional likelihood* becomes

$$\begin{aligned} L(\mu, \phi, \sigma_w^2 \mid x_1) &= \prod_{t=2}^n f_w [(x_t - \mu) - \phi(x_{t-1} - \mu)] \\ &= (2\pi\sigma_w^2)^{-(n-1)/2} \exp \left[-\frac{S_c(\mu, \phi)}{2\sigma_w^2} \right], \end{aligned} \quad (3.111)$$

where the *conditional sum of squares* is

^{3.6} The criterion function is sometimes called the profile or concentrated likelihood.

$$S_c(\mu, \phi) = \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2. \quad (3.112)$$

The conditional MLE of σ_w^2 is

$$\hat{\sigma}_w^2 = S_c(\hat{\mu}, \hat{\phi})/(n - 1), \quad (3.113)$$

and $\hat{\mu}$ and $\hat{\phi}$ are the values that minimize the conditional sum of squares, $S_c(\mu, \phi)$. Letting $\alpha = \mu(1 - \phi)$, the conditional sum of squares can be written as

$$S_c(\mu, \phi) = \sum_{t=2}^n [x_t - (\alpha + \phi x_{t-1})]^2. \quad (3.114)$$

The problem is now the linear regression problem stated in [Section 2.1](#). Following the results from least squares estimation, we have $\hat{\alpha} = \bar{x}_{(2)} - \hat{\phi}\bar{x}_{(1)}$, where $\bar{x}_{(1)} = (n - 1)^{-1} \sum_{t=1}^{n-1} x_t$, and $\bar{x}_{(2)} = (n - 1)^{-1} \sum_{t=2}^n x_t$, and the conditional estimates are then

$$\hat{\mu} = \frac{\bar{x}_{(2)} - \hat{\phi}\bar{x}_{(1)}}{1 - \hat{\phi}} \quad (3.115)$$

$$\hat{\phi} = \frac{\sum_{t=2}^n (x_t - \bar{x}_{(2)})(x_{t-1} - \bar{x}_{(1)})}{\sum_{t=2}^n (x_{t-1} - \bar{x}_{(1)})^2}. \quad (3.116)$$

From (3.115) and (3.116), we see that $\hat{\mu} \approx \bar{x}$ and $\hat{\phi} \approx \hat{\rho}(1)$. That is, the Yule–Walker estimators and the conditional least squares estimators are approximately the same. The only difference is the inclusion or exclusion of terms involving the endpoints, x_1 and x_n . We can also adjust the estimate of σ_w^2 in (3.113) to be equivalent to the least squares estimator, that is, divide $S_c(\hat{\mu}, \hat{\phi})$ by $(n - 3)$ instead of $(n - 1)$ in (3.113).

For general AR(p) models, maximum likelihood estimation, unconditional least squares, and conditional least squares follow analogously to the AR(1) example. For general ARMA models, it is difficult to write the likelihood as an explicit function of the parameters. Instead, it is advantageous to write the likelihood in terms of the *innovations*, or one-step-ahead prediction errors, $x_t - x_t^{t-1}$. This will also be useful in [Chapter 6](#) when we study state-space models.

For a normal ARMA(p, q) model, let $\beta = (\mu, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q)'$ be the $(p + q + 1)$ -dimensional vector of the model parameters. The likelihood can be written as

$$L(\beta, \sigma_w^2) = \prod_{t=1}^n f(x_t \mid x_{t-1}, \dots, x_1).$$

The conditional distribution of x_t given x_{t-1}, \dots, x_1 is Gaussian with mean x_t^{t-1} and variance P_t^{t-1} . Recall from (3.71) that $P_t^{t-1} = \gamma(0) \prod_{j=1}^{t-1} (1 - \phi_{jj}^2)$. For ARMA models, $\gamma(0) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_j^2$, in which case we may write

$$P_t^{t-1} = \sigma_w^2 \left\{ \left[\sum_{j=0}^{\infty} \psi_j^2 \right] \left[\prod_{j=1}^{t-1} (1 - \phi_{jj}^2) \right] \right\} \stackrel{\text{def}}{=} \sigma_w^2 r_t,$$

where r_t is the term in the braces. Note that the r_t terms are functions only of the regression parameters and that they may be computed recursively as $r_{t+1} = (1 - \phi_{tt}^2)r_t$ with initial condition $r_1 = \sum_{j=0}^{\infty} \psi_j^2$. The likelihood of the data can now be written as

$$L(\beta, \sigma_w^2) = (2\pi\sigma_w^2)^{-n/2} [r_1(\beta)r_2(\beta)\cdots r_n(\beta)]^{-1/2} \exp\left[-\frac{S(\beta)}{2\sigma_w^2}\right], \quad (3.117)$$

where

$$S(\beta) = \sum_{t=1}^n \left[\frac{(x_t - x_t^{t-1}(\beta))^2}{r_t(\beta)} \right]. \quad (3.118)$$

Both x_t^{t-1} and r_t are functions of β alone, and we make that fact explicit in (3.117)–(3.118). Given values for β and σ_w^2 , the likelihood may be evaluated using the techniques of Section 3.4. Maximum likelihood estimation would now proceed by maximizing (3.117) with respect to β and σ_w^2 . As in the AR(1) example, we have

$$\hat{\sigma}_w^2 = n^{-1} S(\hat{\beta}), \quad (3.119)$$

where $\hat{\beta}$ is the value of β that minimizes the concentrated likelihood

$$l(\beta) = \log [n^{-1} S(\beta)] + n^{-1} \sum_{t=1}^n \log r_t(\beta). \quad (3.120)$$

For the AR(1) model (3.106) discussed previously, recall that $x_1^0 = \mu$ and $x_t^{t-1} = \mu + \phi(x_{t-1} - \mu)$, for $t = 2, \dots, n$. Also, using the fact that $\phi_{11} = \phi$ and $\phi_{hh} = 0$ for $h > 1$, we have $r_1 = \sum_{j=0}^{\infty} \phi^{2j} = (1 - \phi^2)^{-1}$, $r_2 = (1 - \phi^2)^{-1}(1 - \phi^2) = 1$, and in general, $r_t = 1$ for $t = 2, \dots, n$. Hence, the likelihood presented in (3.107) is identical to the innovations form of the likelihood given by (3.117). Moreover, the generic $S(\beta)$ in (3.118) is $S(\mu, \phi)$ given in (3.108) and the generic $l(\beta)$ in (3.120) is $l(\mu, \phi)$ in (3.110).

Unconditional least squares would be performed by minimizing (3.118) with respect to β . Conditional least squares estimation would involve minimizing (3.118) with respect to β but where, to ease the computational burden, the predictions and their errors are obtained by conditioning on initial values of the data. In general, numerical optimization routines are used to obtain the actual estimates and their standard errors.

Example 3.30 The Newton–Raphson and Scoring Algorithms

Two common numerical optimization routines for accomplishing maximum likelihood estimation are Newton–Raphson and scoring. We will give a brief account of the mathematical ideas here. The actual implementation of these algorithms is much more complicated than our discussion might imply. For details, the reader is referred to any of the *Numerical Recipes* books, for example, Press et al. (1993).

Let $l(\beta)$ be a criterion function of k parameters $\beta = (\beta_1, \dots, \beta_k)$ that we wish to minimize with respect to β . For example, consider the likelihood function given by (3.110) or by (3.120). Suppose $l(\hat{\beta})$ is the extremum that we are interested in

finding, and $\hat{\beta}$ is found by solving $\partial l(\beta)/\partial \beta_j = 0$, for $j = 1, \dots, k$. Let $l^{(1)}(\beta)$ denote the $k \times 1$ vector of partials

$$l^{(1)}(\beta) = \left(\frac{\partial l(\beta)}{\partial \beta_1}, \dots, \frac{\partial l(\beta)}{\partial \beta_k} \right)'.$$

Note, $l^{(1)}(\hat{\beta}) = 0$, the $k \times 1$ zero vector. Let $l^{(2)}(\beta)$ denote the $k \times k$ matrix of second-order partials

$$l^{(2)}(\beta) = \left\{ -\frac{\partial^2 l(\beta)}{\partial \beta_i \partial \beta_j} \right\}_{i,j=1}^k,$$

and assume $l^{(2)}(\beta)$ is nonsingular. Let $\beta_{(0)}$ be a “sufficiently good” initial estimator of β . Then, using a Taylor expansion, we have the following approximation:

$$0 = l^{(1)}(\hat{\beta}) \approx l^{(1)}(\beta_{(0)}) - l^{(2)}(\beta_{(0)}) [\hat{\beta} - \beta_{(0)}].$$

Setting the right-hand side equal to zero and solving for $\hat{\beta}$ [call the solution $\beta_{(1)}$], we get

$$\beta_{(1)} = \beta_{(0)} + \left[l^{(2)}(\beta_{(0)}) \right]^{-1} l^{(1)}(\beta_{(0)}).$$

The Newton–Raphson algorithm proceeds by iterating this result, replacing $\beta_{(0)}$ by $\beta_{(1)}$ to get $\beta_{(2)}$, and so on, until convergence. Under a set of appropriate conditions, the sequence of estimators, $\beta_{(1)}, \beta_{(2)}, \dots$, will converge to $\hat{\beta}$, the MLE of β .

For maximum likelihood estimation, the criterion function used is $l(\beta)$ given by (3.120); $l^{(1)}(\beta)$ is called the score vector, and $l^{(2)}(\beta)$ is called the *Hessian*. In the method of scoring, we replace $l^{(2)}(\beta)$ by $E[l^{(2)}(\beta)]$, the *information* matrix. Under appropriate conditions, the inverse of the information matrix is the asymptotic variance–covariance matrix of the estimator $\hat{\beta}$. This is sometimes approximated by the inverse of the Hessian at $\hat{\beta}$. If the derivatives are difficult to obtain, it is possible to use quasi-maximum likelihood estimation where numerical techniques are used to approximate the derivatives.

Example 3.31 MLE for the Recruitment Series

So far, we have fit an AR(2) model to the Recruitment series using ordinary least squares ([Example 3.18](#)) and using Yule–Walker ([Example 3.28](#)). The following is an R session used to fit an AR(2) model via maximum likelihood estimation to the Recruitment series; these results can be compared to the results in [Example 3.18](#) and [Example 3.28](#).

```
rec.mle = ar.mle(rec, order=2)
rec.mle$x.mean    # 62.26
rec.mle$ar        # 1.35, -.46
sqrt(diag(rec.mle$asy.var.coef)) # .04, .04
rec.mle$var.pred   # 89.34
```

We now discuss least squares for ARMA(p, q) models via *Gauss–Newton*. For general and complete details of the Gauss–Newton procedure, the reader is referred

to Fuller (1996). As before, write $\beta = (\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q)'$, and for the ease of discussion, we will put $\mu = 0$. We write the model in terms of the errors

$$w_t(\beta) = x_t - \sum_{j=1}^p \phi_j x_{t-j} - \sum_{k=1}^q \theta_k w_{t-k}(\beta), \quad (3.121)$$

emphasizing the dependence of the errors on the parameters.

For conditional least squares, we approximate the residual sum of squares by conditioning on x_1, \dots, x_p (if $p > 0$) and $w_p = w_{p-1} = w_{p-2} = \dots = w_{1-q} = 0$ (if $q > 0$), in which case, given β , we may evaluate (3.121) for $t = p+1, p+2, \dots, n$. Using this conditioning argument, the conditional error sum of squares is

$$S_c(\beta) = \sum_{t=p+1}^n w_t^2(\beta). \quad (3.122)$$

Minimizing $S_c(\beta)$ with respect to β yields the conditional least squares estimates. If $q = 0$, the problem is linear regression and no iterative technique is needed to minimize $S_c(\phi_1, \dots, \phi_p)$. If $q > 0$, the problem becomes nonlinear regression and we will have to rely on numerical optimization.

When n is large, conditioning on a few initial values will have little influence on the final parameter estimates. In the case of small to moderate sample sizes, one may wish to rely on unconditional least squares. The unconditional least squares problem is to choose β to minimize the unconditional sum of squares, which we have generically denoted by $S(\beta)$ in this section. The unconditional sum of squares can be written in various ways, and one useful form in the case of ARMA(p, q) models is derived in Box et al. (1994, Appendix A7.3). They showed (see Problem 3.19) the unconditional sum of squares can be written as

$$S(\beta) = \sum_{t=-\infty}^n \tilde{w}_t^2(\beta), \quad (3.123)$$

where $\tilde{w}_t(\beta) = E(w_t | x_1, \dots, x_n)$. When $t \leq 0$, the $\hat{w}_t(\beta)$ are obtained by backcasting. As a practical matter, we approximate $S(\beta)$ by starting the sum at $t = -M+1$, where M is chosen large enough to guarantee $\sum_{t=-\infty}^{-M} \tilde{w}_t^2(\beta) \approx 0$. In the case of unconditional least squares estimation, a numerical optimization technique is needed even when $q = 0$.

To employ Gauss–Newton, let $\beta_{(0)} = (\phi_1^{(0)}, \dots, \phi_p^{(0)}, \theta_1^{(0)}, \dots, \theta_q^{(0)})'$ be an initial estimate of β . For example, we could obtain $\beta_{(0)}$ by method of moments. The first-order Taylor expansion of $w_t(\beta)$ is

$$w_t(\beta) \approx w_t(\beta_{(0)}) - (\beta - \beta_{(0)})' z_t(\beta_{(0)}), \quad (3.124)$$

where

$$z'_t(\beta_{(0)}) = \left(-\frac{\partial w_t(\beta)}{\partial \beta_1}, \dots, -\frac{\partial w_t(\beta)}{\partial \beta_{p+q}} \right) \Bigg|_{\beta=\beta_{(0)}}, \quad t = 1, \dots, n.$$

The linear approximation of $S_c(\beta)$ is

$$Q(\beta) = \sum_{t=p+1}^n [w_t(\beta_{(0)}) - (\beta - \beta_{(0)})' z_t(\beta_{(0)})]^2 \quad (3.125)$$

and this is the quantity that we will minimize. For approximate unconditional least squares, we would start the sum in (3.125) at $t = -M + 1$, for a large value of M , and work with the backcasted values.

Using the results of ordinary least squares (Section 2.1), we know

$$\widehat{(\beta - \beta_{(0)})} = \left(n^{-1} \sum_{t=p+1}^n z_t(\beta_{(0)}) z_t'(\beta_{(0)}) \right)^{-1} \left(n^{-1} \sum_{t=p+1}^n z_t(\beta_{(0)}) w_t(\beta_{(0)}) \right) \quad (3.126)$$

minimizes $Q(\beta)$. From (3.126), we write the *one-step Gauss–Newton estimate* as

$$\beta_{(1)} = \beta_{(0)} + A(\beta_{(0)}), \quad (3.127)$$

where $A(\beta_{(0)})$ denotes the right-hand side of (3.126). Gauss–Newton estimation is accomplished by replacing $\beta_{(0)}$ by $\beta_{(1)}$ in (3.127). This process is repeated by calculating, at iteration $j = 2, 3, \dots$,

$$\beta_{(j)} = \beta_{(j-1)} + A(\beta_{(j-1)})$$

until convergence.

Example 3.32 Gauss–Newton for an MA(1)

Consider an invertible MA(1) process, $x_t = w_t + \theta w_{t-1}$. Write the truncated errors as

$$w_t(\theta) = x_t - \theta w_{t-1}(\theta), \quad t = 1, \dots, n, \quad (3.128)$$

where we condition on $w_0(\theta) = 0$. Taking derivatives and negating,

$$-\frac{\partial w_t(\theta)}{\partial \theta} = w_{t-1}(\theta) + \theta \frac{\partial w_{t-1}(\theta)}{\partial \theta}, \quad t = 1, \dots, n, \quad (3.129)$$

where $\partial w_0(\theta)/\partial \theta = 0$. We can also write (3.129) as

$$z_t(\theta) = w_{t-1}(\theta) - \theta z_{t-1}(\theta), \quad t = 1, \dots, n, \quad (3.130)$$

where $z_t(\theta) = -\partial w_t(\theta)/\partial \theta$ and $z_0(\theta) = 0$.

Let $\theta_{(0)}$ be an initial estimate of θ , for example, the estimate given in Example 3.29. Then, the Gauss–Newton procedure for conditional least squares is given by

$$\theta_{(j+1)} = \theta_{(j)} + \frac{\sum_{t=1}^n z_t(\theta_{(j)}) w_t(\theta_{(j)})}{\sum_{t=1}^n z_t^2(\theta_{(j)})}, \quad j = 0, 1, 2, \dots, \quad (3.131)$$

where the values in (3.131) are calculated recursively using (3.128) and (3.130). The calculations are stopped when $|\theta_{(j+1)} - \theta_{(j)}|$, or $|Q(\theta_{(j+1)}) - Q(\theta_{(j)})|$, are smaller than some preset amount.

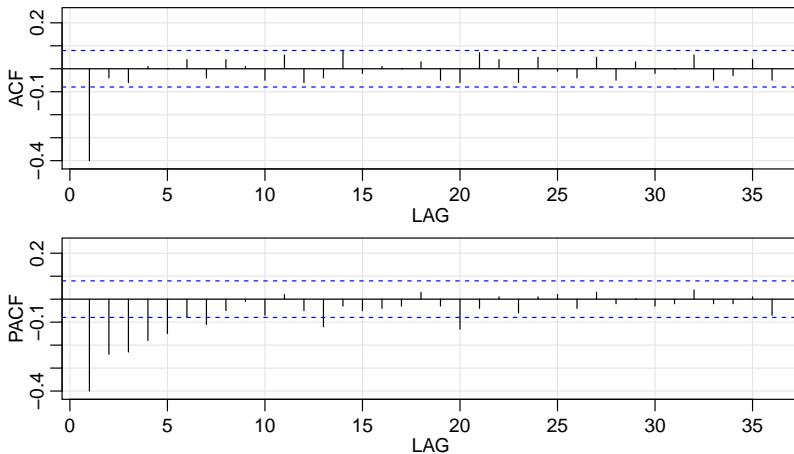


Fig. 3.9. ACF and PACF of transformed glacial varves.

Example 3.33 Fitting the Glacial Varve Series

Consider the series of glacial varve thicknesses from Massachusetts for $n = 634$ years, as analyzed in [Example 2.7](#) and in [Problem 2.8](#), where it was argued that a first-order moving average model might fit the logarithmically transformed and differenced varve series, say,

$$\nabla \log(x_t) = \log(x_t) - \log(x_{t-1}) = \log\left(\frac{x_t}{x_{t-1}}\right),$$

which can be interpreted as being approximately the percentage change in the thickness.

The sample ACF and PACF, shown in [Figure 3.9](#), confirm the tendency of $\nabla \log(x_t)$ to behave as a first-order moving average process as the ACF has only a significant peak at lag one and the PACF decreases exponentially. Using [Table 3.1](#), this sample behavior fits that of the MA(1) very well.

Since $\hat{\rho}(1) = -.397$, our initial estimate is $\theta_{(0)} = -.495$ using [\(3.105\)](#). The results of eleven iterations of the Gauss–Newton procedure, [\(3.131\)](#), starting with $\theta_{(0)}$ are given in [Table 3.2](#). The final estimate is $\hat{\theta} = \theta_{(11)} = -.773$; interim values and the corresponding value of the conditional sum of squares, $S_c(\theta)$ given in [\(3.122\)](#), are also displayed in the table. The final estimate of the error variance is $\hat{\sigma}_w^2 = 148.98/632 = .236$ with 632 degrees of freedom (one is lost in differencing). The value of the sum of the squared derivatives at convergence is $\sum_{t=1}^n z_t^2(\theta_{(11)}) = 368.741$, and consequently, the estimated standard error of $\hat{\theta}$ is $\sqrt{.236/368.741} = .025$ ^{3.7}; this leads to a t -value of $-.773/.025 = -30.92$ with 632 degrees of freedom.

[Figure 3.10](#) displays the conditional sum of squares, $S_c(\theta)$ as a function of θ , as well as indicating the values of each step of the Gauss–Newton algorithm. Note

^{3.7} To estimate the standard error, we are using the standard regression results from [\(2.6\)](#) as an approximation

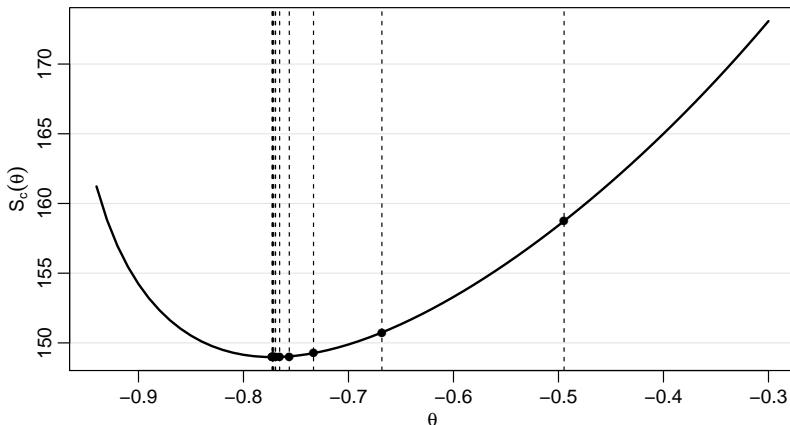


Fig. 3.10. Conditional sum of squares versus values of the moving average parameter for the glacial varve example, [Example 3.33](#). Vertical lines indicate the values of the parameter obtained via Gauss–Newton; see [Table 3.2](#) for the actual values.

that the Gauss–Newton procedure takes large steps toward the minimum initially, and then takes very small steps as it gets close to the minimizing value. When there is only one parameter, as in this case, it would be easy to evaluate $S_c(\theta)$ on a grid of points, and then choose the appropriate value of θ from the grid search. It would be difficult, however, to perform grid searches when there are many parameters.

The following code was used in this example.

```

x = diff(log(varve))
# Evaluate Sc on a Grid
c(0) -> w -> z
c() -> Sc -> Sz -> Szw
num = length(x)
th = seq(-.3,-.94,-.01)
for (p in 1:length(th)){
  for (i in 2:num){ w[i] = x[i]-th[p]*w[i-1] }
  Sc[p] = sum(w^2) }
plot(th, Sc, type="l", ylab=expression(S[c](theta)), xlab=expression(theta),
  lwd=2)
# Gauss–Newton Estimation
r = acf(x, lag=1, plot=FALSE)$acf[-1]
rstart = (1-sqrt(1-4*(r^2)))/(2*r)      # from (3.105)
c(0) -> w -> z
c() -> Sc -> Sz -> Szw -> para
niter = 12
para[1] = rstart
for (p in 1:niter){
  for (i in 2:num){ w[i] = x[i]-para[p]*w[i-1]
    z[i] = w[i-1]-para[p]*z[i-1] }
  Sc[p] = sum(w^2)
  Sz[p] = sum(z^2)
  Szw[p] = sum(z*w)
  para[p+1] = para[p] + Szw[p]/Sz[p]  }

```

Table 3.2. Gauss–Newton Results for Example 3.33

j	$\theta_{(j)}$	$S_c(\theta_{(j)})$	$\sum_{t=1}^n z_t^2(\theta_{(j)})$
0	-0.495	158.739	171.240
1	-0.668	150.747	235.266
2	-0.733	149.264	300.562
3	-0.756	149.031	336.823
4	-0.766	148.990	354.173
5	-0.769	148.982	362.167
6	-0.771	148.980	365.801
7	-0.772	148.980	367.446
8	-0.772	148.980	368.188
9	-0.772	148.980	368.522
10	-0.773	148.980	368.673
11	-0.773	148.980	368.741

```
round(cbind(iteration=0:(niter-1), thetahat=para[1:niter] , Sc , Sz ), 3)
abline(v = para[1:12], lty=2)
points(para[1:12], Sc[1:12], pch=16)
```

In the general case of causal and invertible ARMA(p, q) models, maximum likelihood estimation and conditional and unconditional least squares estimation (and Yule–Walker estimation in the case of AR models) all lead to optimal estimators. The proof of this general result can be found in a number of texts on theoretical time series analysis (for example, Brockwell and Davis, 1991, or Hannan, 1970, to mention a few). We will denote the ARMA coefficient parameters by $\beta = (\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q)'$.

Property 3.10 Large Sample Distribution of the Estimators

Under appropriate conditions, for causal and invertible ARMA processes, the maximum likelihood, the unconditional least squares, and the conditional least squares estimators, each initialized by the method of moments estimator, all provide optimal estimators of σ_w^2 and β , in the sense that $\hat{\sigma}_w^2$ is consistent, and the asymptotic distribution of $\hat{\beta}$ is the best asymptotic normal distribution. In particular, as $n \rightarrow \infty$,

$$\sqrt{n} (\hat{\beta} - \beta) \xrightarrow{d} N\left(0, \sigma_w^2 \Gamma_{p,q}^{-1}\right). \quad (3.132)$$

The asymptotic variance–covariance matrix of the estimator $\hat{\beta}$ is the inverse of the information matrix. In particular, the $(p+q) \times (p+q)$ matrix $\Gamma_{p,q}$, has the form

$$\Gamma_{p,q} = \begin{pmatrix} \Gamma_{\phi\phi} & \Gamma_{\phi\theta} \\ \Gamma_{\theta\phi} & \Gamma_{\theta\theta} \end{pmatrix}. \quad (3.133)$$

The $p \times p$ matrix $\Gamma_{\phi\phi}$ is given by (3.100), that is, the ij -th element of $\Gamma_{\phi\phi}$, for $i, j = 1, \dots, p$, is $\gamma_x(i-j)$ from an AR(p) process, $\phi(B)x_t = w_t$. Similarly, $\Gamma_{\theta\theta}$ is a $q \times q$ matrix with the ij -th element, for $i, j = 1, \dots, q$, equal to $\gamma_y(i-j)$ from an AR(q) process, $\theta(B)y_t = w_t$. The $p \times q$ matrix $\Gamma_{\phi\theta} = \{\gamma_{xy}(i-j)\}$, for $i = 1, \dots, p$; $j = 1, \dots, q$; that is, the ij -th element is the cross-covariance between the two AR processes given by $\phi(B)x_t = w_t$ and $\theta(B)y_t = w_t$. Finally, $\Gamma_{\theta\phi} = \Gamma'_{\phi\theta}$ is $q \times p$.

Further discussion of [Property 3.10](#), including a proof for the case of least squares estimators for $\text{AR}(p)$ processes, can be found in [Section B.3](#).

Example 3.34 Some Specific Asymptotic Distributions

The following are some specific cases of [Property 3.10](#).

AR(1): $\gamma_x(0) = \sigma_w^2 / (1 - \phi^2)$, so $\sigma_w^2 \Gamma_{1,0}^{-1} = (1 - \phi^2)$. Thus,

$$\hat{\phi} \sim \text{AN} [\phi, n^{-1}(1 - \phi^2)]. \quad (3.134)$$

AR(2): The reader can verify that

$$\gamma_x(0) = \left(\frac{1 - \phi_2}{1 + \phi_2} \right) \frac{\sigma_w^2}{(1 - \phi_2)^2 - \phi_1^2}$$

and $\gamma_x(1) = \phi_1 \gamma_x(0) + \phi_2 \gamma_x(1)$. From these facts, we can compute $\Gamma_{2,0}^{-1}$. In particular, we have

$$\begin{pmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \end{pmatrix} \sim \text{AN} \left[\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}, n^{-1} \begin{pmatrix} 1 - \phi_2^2 & -\phi_1(1 + \phi_2) \\ \text{sym} & 1 - \phi_2^2 \end{pmatrix} \right]. \quad (3.135)$$

MA(1): In this case, write $\theta(B)y_t = w_t$, or $y_t + \theta y_{t-1} = w_t$. Then, analogous to the AR(1) case, $\gamma_y(0) = \sigma_w^2 / (1 - \theta^2)$, so $\sigma_w^2 \Gamma_{0,1}^{-1} = (1 - \theta^2)$. Thus,

$$\hat{\theta} \sim \text{AN} [\theta, n^{-1}(1 - \theta^2)]. \quad (3.136)$$

MA(2): Write $y_t + \theta_1 y_{t-1} + \theta_2 y_{t-2} = w_t$, so , analogous to the AR(2) case, we have

$$\begin{pmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \end{pmatrix} \sim \text{AN} \left[\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, n^{-1} \begin{pmatrix} 1 - \theta_2^2 & \theta_1(1 + \theta_2) \\ \text{sym} & 1 - \theta_2^2 \end{pmatrix} \right]. \quad (3.137)$$

ARMA(1,1): To calculate $\Gamma_{\phi\theta}$, we must find $\gamma_{xy}(0)$, where $x_t - \phi x_{t-1} = w_t$ and $y_t + \theta y_{t-1} = w_t$. We have

$$\begin{aligned} \gamma_{xy}(0) &= \text{cov}(x_t, y_t) = \text{cov}(\phi x_{t-1} + w_t, -\theta y_{t-1} + w_t) \\ &= -\phi\theta\gamma_{xy}(0) + \sigma_w^2. \end{aligned}$$

Solving, we find, $\gamma_{xy}(0) = \sigma_w^2 / (1 + \phi\theta)$. Thus,

$$\begin{pmatrix} \hat{\phi} \\ \hat{\theta} \end{pmatrix} \sim \text{AN} \left[\begin{pmatrix} \phi \\ \theta \end{pmatrix}, n^{-1} \begin{bmatrix} (1 - \phi^2)^{-1} & (1 + \phi\theta)^{-1} \\ \text{sym} & (1 - \theta^2)^{-1} \end{bmatrix}^{-1} \right]. \quad (3.138)$$

Example 3.35 Overfitting Caveat

The asymptotic behavior of the parameter estimators gives us an additional insight into the problem of fitting ARMA models to data. For example, suppose a time series follows an AR(1) process and we decide to fit an AR(2) to the data. Do any problems occur in doing this? More generally, why not simply fit large-order AR models to make sure that we capture the dynamics of the process? After all,

if the process is truly an AR(1), the other autoregressive parameters will not be significant. The answer is that if we *overfit*, we obtain less efficient, or less precise parameter estimates. For example, if we fit an AR(1) to an AR(1) process, for large n , $\text{var}(\hat{\phi}_1) \approx n^{-1}(1 - \phi_1^2)$. But, if we fit an AR(2) to the AR(1) process, for large n , $\text{var}(\hat{\phi}_1) \approx n^{-1}(1 - \phi_2^2) = n^{-1}$ because $\phi_2 = 0$. Thus, the variance of ϕ_1 has been inflated, making the estimator less precise.

We do want to mention, however, that overfitting can be used as a diagnostic tool. For example, if we fit an AR(2) model to the data and are satisfied with that model, then adding one more parameter and fitting an AR(3) should lead to approximately the same model as in the AR(2) fit. We will discuss model diagnostics in more detail in [Section 3.7](#).

The reader might wonder, for example, why the asymptotic distributions of $\hat{\phi}$ from an AR(1) and $\hat{\theta}$ from an MA(1) are of the same form; compare [\(3.134\)](#) to [\(3.136\)](#). It is possible to explain this unexpected result heuristically using the intuition of linear regression. That is, for the normal regression model presented in [Section 2.1](#) with no intercept term, $x_t = \beta z_t + w_t$, we know $\hat{\beta}$ is normally distributed with mean β , and from [\(2.6\)](#),

$$\text{var}\left\{\sqrt{n}(\hat{\beta} - \beta)\right\} = n\sigma_w^2 \left(\sum_{t=1}^n z_t^2\right)^{-1} = \sigma_w^2 \left(n^{-1} \sum_{t=1}^n z_t^2\right)^{-1}.$$

For the causal AR(1) model given by $x_t = \phi x_{t-1} + w_t$, the intuition of regression tells us to expect that, for n large,

$$\sqrt{n}(\hat{\phi} - \phi)$$

is approximately normal with mean zero and with variance given by

$$\sigma_w^2 \left(n^{-1} \sum_{t=2}^n x_{t-1}^2\right)^{-1}.$$

Now, $n^{-1} \sum_{t=2}^n x_{t-1}^2$ is the sample variance (recall that the mean of x_t is zero) of the x_t , so as n becomes large we would expect it to approach $\text{var}(x_t) = \gamma(0) = \sigma_w^2/(1 - \phi^2)$. Thus, the large sample variance of $\sqrt{n}(\hat{\phi} - \phi)$ is

$$\sigma_w^2 \gamma_x(0)^{-1} = \sigma_w^2 \left(\frac{\sigma_w^2}{1 - \phi^2}\right)^{-1} = (1 - \phi^2);$$

that is, [\(3.134\)](#) holds.

In the case of an MA(1), we may use the discussion of [Example 3.32](#) to write an approximate regression model for the MA(1). That is, consider the approximation [\(3.130\)](#) as the regression model

$$z_t(\hat{\theta}) = -\theta z_{t-1}(\hat{\theta}) + w_{t-1},$$

where now, $z_{t-1}(\hat{\theta})$ as defined in [Example 3.32](#), plays the role of the regressor. Continuing with the analogy, we would expect the asymptotic distribution of $\sqrt{n}(\hat{\theta} - \theta)$ to be normal, with mean zero, and approximate variance

$$\sigma_w^2 \left(n^{-1} \sum_{t=2}^n z_{t-1}^2(\hat{\theta}) \right)^{-1}.$$

As in the AR(1) case, $n^{-1} \sum_{t=2}^n z_{t-1}^2(\hat{\theta})$ is the sample variance of the $z_t(\hat{\theta})$ so, for large n , this should be $\text{var}\{z_t(\theta)\} = \gamma_z(0)$, say. But note, as seen from [\(3.130\)](#), $z_t(\theta)$ is approximately an AR(1) process with parameter $-\theta$. Thus,

$$\sigma_w^2 \gamma_z(0)^{-1} = \sigma_w^2 \left(\frac{\sigma_w^2}{1 - (-\theta)^2} \right)^{-1} = (1 - \theta^2),$$

which agrees with [\(3.136\)](#). Finally, the asymptotic distributions of the AR parameter estimates and the MA parameter estimates are of the same form because in the MA case, the “regressors” are the differential processes $z_t(\theta)$ that have AR structure, and it is this structure that determines the asymptotic variance of the estimators. For a rigorous account of this approach for the general case, see [Fuller \(1996, Theorem 5.5.4\)](#).

In [Example 3.33](#), the estimated standard error of $\hat{\theta}$ was .025. In that example, we used regression results to estimate the standard error as the square root of

$$n^{-1} \hat{\sigma}_w^2 \left(n^{-1} \sum_{t=1}^n z_t^2(\hat{\theta}) \right)^{-1} = \frac{\hat{\sigma}_w^2}{\sum_{t=1}^n z_t^2(\hat{\theta})},$$

where $n = 632$, $\hat{\sigma}_w^2 = .236$, $\sum_{t=1}^n z_t^2(\hat{\theta}) = 368.74$ and $\hat{\theta} = -.773$. Using [\(3.136\)](#), we could have also calculated this value using the asymptotic approximation, the square root of $(1 - (-.773)^2)/632$, which is also .025.

If n is small, or if the parameters are close to the boundaries, the asymptotic approximations can be quite poor. The *bootstrap* can be helpful in this case; for a broad treatment of the bootstrap, see [Efron and Tibshirani \(1994\)](#). We discuss the case of an AR(1) here and leave the general discussion for [Chapter 6](#). For now, we give a simple example of the bootstrap for an AR(1) process.

Example 3.36 Bootstrapping an AR(1)

We consider an AR(1) model with a regression coefficient near the boundary of causality and an error process that is symmetric but not normal. Specifically, consider the causal model

$$x_t = \mu + \phi(x_{t-1} - \mu) + w_t, \tag{3.139}$$

where $\mu = 50$, $\phi = .95$, and w_t are iid double exponential (Laplace) with location zero, and scale parameter $\beta = 2$. The density of w_t is given by

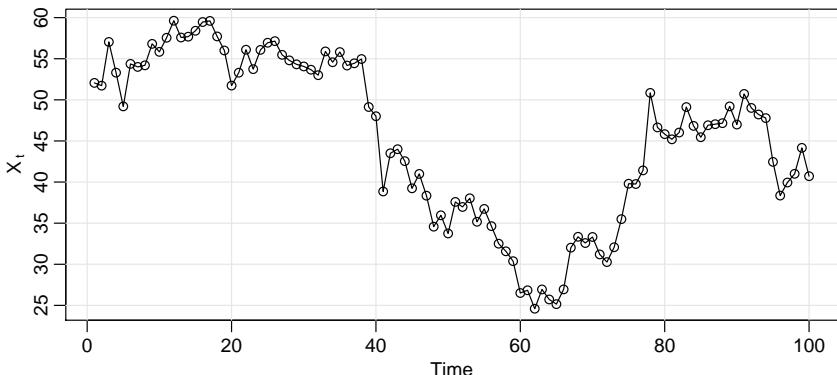


Fig. 3.11. One hundred observations generated from the model in Example 3.36.

$$f(w) = \frac{1}{2\beta} \exp \{-|w|/\beta\} \quad -\infty < w < \infty.$$

In this example, $E(w_t) = 0$ and $\text{var}(w_t) = 2\beta^2 = 8$. Figure 3.11 shows $n = 100$ simulated observations from this process. This particular realization is interesting; the data look like they were generated from a nonstationary process with three different mean levels. In fact, the data were generated from a well-behaved, albeit non-normal, stationary and causal model. To show the advantages of the bootstrap, we will act as if we do not know the actual error distribution. The data in Figure 3.11 were generated as follows.

```
set.seed(101010)
e = rexp(150, rate=.5); u = runif(150,-1,1); de = e*sign(u)
dex = 50 + arima.sim(n=100, list(ar=.95), innov=de, n.start=50)
plot.ts(dex, type='o', ylab=expression(X[~t]))
```

Using these data, we obtained the Yule–Walker estimates $\hat{\mu} = 45.25$, $\hat{\phi} = .96$, and $\hat{\sigma}_w^2 = 7.88$, as follows.

```
fit = ar.yw(dex, order=1)
round(cbind(fit$x.mean, fit$ar, fit$var.pred), 2)
[1,] 45.25 0.96 7.88
```

To assess the finite sample distribution of $\hat{\phi}$ when $n = 100$, we simulated 1000 realizations of this AR(1) process and estimated the parameters via Yule–Walker. The finite sampling density of the Yule–Walker estimate of ϕ , based on the 1000 repeated simulations, is shown in Figure 3.12. Based on Property 3.10, we would say that $\hat{\phi}$ is approximately normal with mean ϕ (which we supposedly do not know) and variance $(1 - \phi^2)/100$, which we would approximate by $(1 - .96^2)/100 = .03^2$; this distribution is superimposed on Figure 3.12. Clearly the sampling distribution is not close to normality for this sample size. The R code to perform the simulation is as follows. We use the results at the end of the example

```
set.seed(111)
phi.yw = rep(NA, 1000)
for (i in 1:1000){
```

```
e = rexp(150, rate=.5); u = runif(150,-1,1); de = e*sign(u)
x = 50 + arima.sim(n=100, list(ar=.95), innov=de, n.start=50)
phi.yw[i] = ar.yw(x, order=1)$ar }
```

The preceding simulation required full knowledge of the model, the parameter values and the noise distribution. Of course, in a sampling situation, we would not have the information necessary to do the preceding simulation and consequently would not be able to generate a figure like [Figure 3.12](#). The bootstrap, however, gives us a way to attack the problem.

To simplify the discussion and the notation, we condition on x_1 throughout the example. In this case, the one-step-ahead predictors have a simple form,

$$x_t^{t-1} = \mu + \phi(x_{t-1} - \mu), \quad t = 2, \dots, 100.$$

Consequently, the innovations, $\epsilon_t = x_t - x_t^{t-1}$, are given by

$$\epsilon_t = (x_t - \mu) - \phi(x_{t-1} - \mu), \quad t = 2, \dots, 100, \quad (3.140)$$

each with MSPE $P_t^{t-1} = E(\epsilon_t^2) = E(w_t^2) = \sigma_w^2$ for $t = 2, \dots, 100$. We can use (3.140) to write the model in terms of the innovations,

$$x_t = x_t^{t-1} + \epsilon_t = \mu + \phi(x_{t-1} - \mu) + \epsilon_t \quad t = 2, \dots, 100. \quad (3.141)$$

To perform the bootstrap simulation, we replace the parameters with their estimates in (3.141), that is, $\hat{\mu} = 45.25$ and $\hat{\phi} = .96$, and denote the resulting sample innovations as $\{\hat{\epsilon}_2, \dots, \hat{\epsilon}_{100}\}$. To obtain one bootstrap sample, first randomly sample, with replacement, $n = 99$ values from the set of sample innovations; call the sampled values $\{\epsilon_2^*, \dots, \epsilon_{100}^*\}$. Now, generate a bootstrapped data set sequentially by setting

$$x_t^* = 45.25 + .96(x_{t-1}^* - 45.25) + \epsilon_t^*, \quad t = 2, \dots, 100. \quad (3.142)$$

with x_1^* held fixed at x_1 . Next, estimate the parameters as if the data were x_t^* . Call these estimates $\hat{\mu}(1)$, $\hat{\phi}(1)$, and $\sigma_w^2(1)$. Repeat this process a large number, B , of times, generating a collection of bootstrapped parameter estimates, $\{\hat{\mu}(b), \hat{\phi}(b), \sigma_w^2(b); b = 1, \dots, B\}$. We can then approximate the finite sample distribution of an estimator from the bootstrapped parameter values. For example, we can approximate the distribution of $\hat{\phi} - \phi$ by the empirical distribution of $\hat{\phi}(b) - \hat{\phi}$, for $b = 1, \dots, B$.

[Figure 3.12](#) shows the bootstrap histogram of 500 bootstrapped estimates of ϕ using the data shown in [Figure 3.11](#). Note that the bootstrap distribution of $\hat{\phi}$ is close to the distribution of $\hat{\phi}$ shown in [Figure 3.12](#). The following code was used to perform the bootstrap.

```
set.seed(666) # not that 666
fit = ar.yw(dex, order=1) # assumes the data were retained
m = fit$x.mean # estimate of mean
phi = fit$ar # estimate of phi
nboot = 500 # number of bootstrap replicates
resids = fit$resid[-1] # the 99 innovations
```

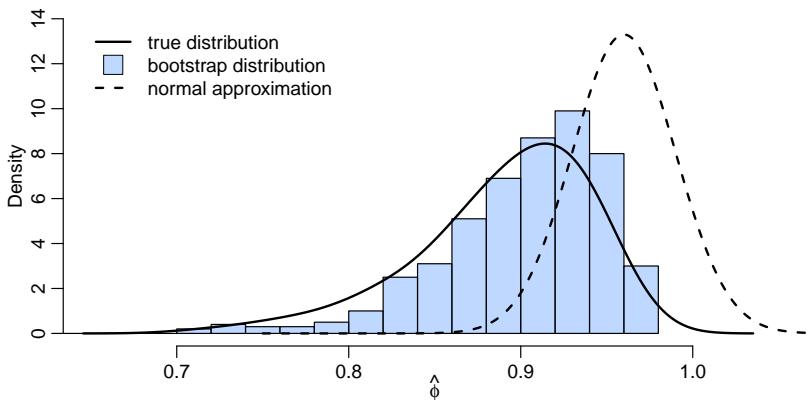


Fig. 3.12. Finite sample density of the Yule–Walker estimate of ϕ (solid line) in Example 3.36 and the corresponding asymptotic normal density (dashed line). Bootstrap histogram of $\hat{\phi}$ based on 500 bootstrapped samples.

```

x.star = dex                      # initialize x*
phi.star.yw = rep(NA, nboot)
# Bootstrap
for (i in 1:nboot) {
  resid.star = sample(resids, replace=TRUE)
  for (t in 1:99){ x.star[t+1] = m + phi*yw(x.star[t]-m) + resid.star[t] }
  phi.star.yw[i] = ar.yw(x.star, order=1)$ar
}
# Picture
culer = rgb(.5,.7,1,.5)
hist(phi.star.yw, 15, main="", prob=TRUE, xlim=c(.65,1.05), ylim=c(0,14),
  col=culer, xlab=expression(hat(phi)))
lines(density(phi.yw, bw=.02), lwd=2)  # from previous simulation
u = seq(.75, 1.1, by=.001)           # normal approximation
lines(u, dnorm(u, mean=.96, sd=.03), lty=2, lwd=2)
legend(.65, 14, legend=c('true distribution', 'bootstrap distribution',
  'normal approximation'), bty='n', lty=c(1,0,2), lwd=c(2,0,2),
  col=1, pch=c(NA,22,NA), pt.bg=c(NA,culer,NA), pt.cex=2.5)

```

3.6 Integrated Models for Nonstationary Data

In Chapter 1 and Chapter 2, we saw that if x_t is a random walk, $x_t = x_{t-1} + w_t$, then by differencing x_t , we find that $\nabla x_t = w_t$ is stationary. In many situations, time series can be thought of as being composed of two components, a nonstationary trend component and a zero-mean stationary component. For example, in Section 2.1 we considered the model

$$x_t = \mu_t + y_t, \quad (3.143)$$

where $\mu_t = \beta_0 + \beta_1 t$ and y_t is stationary. Differencing such a process will lead to a stationary process:

$$\nabla x_t = x_t - x_{t-1} = \beta_1 + y_t - y_{t-1} = \beta_1 + \nabla y_t.$$

Another model that leads to first differencing is the case in which μ_t in (3.143) is stochastic and slowly varying according to a random walk. That is,

$$\mu_t = \mu_{t-1} + v_t$$

where v_t is stationary. In this case,

$$\nabla x_t = v_t + \nabla y_t,$$

is stationary. If μ_t in (3.143) is a k -th order polynomial, $\mu_t = \sum_{j=0}^k \beta_j t^j$, then (Problem 3.27) the differenced series $\nabla^k x_t$ is stationary. Stochastic trend models can also lead to higher order differencing. For example, suppose

$$\mu_t = \mu_{t-1} + v_t \quad \text{and} \quad v_t = v_{t-1} + e_t,$$

where e_t is stationary. Then, $\nabla x_t = v_t + \nabla y_t$ is not stationary, but

$$\nabla^2 x_t = e_t + \nabla^2 y_t$$

is stationary.

The *integrated* ARMA, or ARIMA, model is a broadening of the class of ARMA models to include differencing.

Definition 3.11 A process x_t is said to be **ARIMA**(p, d, q) if

$$\nabla^d x_t = (1 - B)^d x_t$$

is ARMA(p, q). In general, we will write the model as

$$\phi(B)(1 - B)^d x_t = \theta(B)w_t. \quad (3.144)$$

If $E(\nabla^d x_t) = \mu$, we write the model as

$$\phi(B)(1 - B)^d x_t = \delta + \theta(B)w_t,$$

where $\delta = \mu(1 - \phi_1 - \cdots - \phi_p)$.

Because of the nonstationarity, care must be taken when deriving forecasts. For the sake of completeness, we discuss this issue briefly here, but we stress the fact that both the theoretical and computational aspects of the problem are best handled via state-space models. We discuss the theoretical details in Chapter 6. For information on the state-space based computational aspects in R, see the ARIMA help files (`?arima` and `?predict.Arima`); our scripts `sarima` and `sarima.for` are basically wrappers for these R scripts.

It should be clear that, since $y_t = \nabla^d x_t$ is ARMA, we can use Section 3.4 methods to obtain forecasts of y_t , which in turn lead to forecasts for x_t . For example, if $d = 1$, given forecasts y_{n+m}^n for $m = 1, 2, \dots$, we have $y_{n+m}^n = x_{n+m}^n - x_{n+m-1}^n$, so that

$$x_{n+m}^n = y_{n+m}^n + x_{n+m-1}^n$$

with initial condition $x_{n+1}^n = y_{n+1}^n + x_n$ (noting $x_n^n = x_n$).

It is a little more difficult to obtain the prediction errors P_{n+m}^n , but for large n , the approximation used in [Section 3.4](#), equation (3.86), works well. That is, the mean-squared prediction error can be approximated by

$$P_{n+m}^n = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^{*2}, \quad (3.145)$$

where ψ_j^* is the coefficient of z^j in $\psi^*(z) = \theta(z)/\phi(z)(1-z)^d$.

To better understand integrated models, we examine the properties of some simple cases; [Problem 3.29](#) covers the ARIMA(1, 1, 0) case.

Example 3.37 Random Walk with Drift

To fix ideas, we begin by considering the random walk with drift model first presented in [Example 1.11](#), that is,

$$x_t = \delta + x_{t-1} + w_t,$$

for $t = 1, 2, \dots$, and $x_0 = 0$. Technically, the model is not ARIMA, but we could include it trivially as an ARIMA(0, 1, 0) model. Given data x_1, \dots, x_n , the one-step-ahead forecast is given by

$$x_{n+1}^n = E(x_{n+1} \mid x_n, \dots, x_1) = E(\delta + x_n + w_{n+1} \mid x_n, \dots, x_1) = \delta + x_n.$$

The two-step-ahead forecast is given by $x_{n+2}^n = \delta + x_{n+1}^n = 2\delta + x_n$, and consequently, the m -step-ahead forecast, for $m = 1, 2, \dots$, is

$$x_{n+m}^n = m\delta + x_n, \quad (3.146)$$

To obtain the forecast errors, it is convenient to recall equation (1.4); i.e., $x_n = n\delta + \sum_{j=1}^n w_j$, in which case we may write

$$x_{n+m} = (n+m)\delta + \sum_{j=1}^{n+m} w_j = m\delta + x_n + \sum_{j=n+1}^{n+m} w_j.$$

From this it follows that the m -step-ahead prediction error is given by

$$P_{n+m}^n = E(x_{n+m} - x_{n+m}^n)^2 = E\left(\sum_{j=n+1}^{n+m} w_j\right)^2 = m\sigma_w^2. \quad (3.147)$$

Hence, unlike the stationary case (see [Example 3.23](#)), as the forecast horizon grows, the prediction errors, (3.147), increase without bound and the forecasts follow a straight line with slope δ emanating from x_n . We note that (3.145) is exact in this case because $\psi^*(z) = 1/(1-z) = \sum_{j=0}^{\infty} z^j$ for $|z| < 1$, so that $\psi_j^* = 1$ for all j .

The w_t are Gaussian, so estimation is straightforward because the differenced data, say $y_t = \nabla x_t$, are independent and identically distributed normal variates with mean δ and variance σ_w^2 . Consequently, optimal estimates of δ and σ_w^2 are the sample mean and variance of the y_t , respectively.

Example 3.38 IMA(1, 1) and EWMA

The ARIMA(0,1,1), or IMA(1,1) model is of interest because many economic time series can be successfully modeled this way. In addition, the model leads to a frequently used, and abused, forecasting method called exponentially weighted moving averages (EWMA). We will write the model as

$$x_t = x_{t-1} + w_t - \lambda w_{t-1}, \quad (3.148)$$

with $|\lambda| < 1$, for $t = 1, 2, \dots$, and $x_0 = 0$, because this model formulation is easier to work with here, and it leads to the standard representation for EWMA. We could have included a drift term in (3.148), as was done in the previous example, but for the sake of simplicity, we leave it out of the discussion. If we write

$$y_t = w_t - \lambda w_{t-1},$$

we may write (3.148) as $x_t = x_{t-1} + y_t$. Because $|\lambda| < 1$, y_t has an invertible representation, $y_t = \sum_{j=1}^{\infty} \lambda^j y_{t-j} + w_t$, and substituting $y_t = x_t - x_{t-1}$, we may write

$$x_t = \sum_{j=1}^{\infty} (1 - \lambda) \lambda^{j-1} x_{t-j} + w_t. \quad (3.149)$$

as an approximation for large t (put $x_t = 0$ for $t \leq 0$). Verification of (3.149) is left to the reader (Problem 3.28). Using the approximation (3.149), we have that the approximate one-step-ahead predictor, using the notation of Section 3.4, is

$$\begin{aligned} \tilde{x}_{n+1} &= \sum_{j=1}^{\infty} (1 - \lambda) \lambda^{j-1} x_{n+1-j} \\ &= (1 - \lambda)x_n + \lambda \sum_{j=1}^{\infty} (1 - \lambda) \lambda^{j-1} x_{n-j} \\ &= (1 - \lambda)x_n + \lambda \tilde{x}_n. \end{aligned} \quad (3.150)$$

From (3.150), we see that the new forecast is a linear combination of the old forecast and the new observation. Based on (3.150) and the fact that we only observe x_1, \dots, x_n , and consequently y_1, \dots, y_n (because $y_t = x_t - x_{t-1}$; $x_0 = 0$), the truncated forecasts are

$$\tilde{x}_{n+1}^n = (1 - \lambda)x_n + \lambda \tilde{x}_n^{n-1}, \quad n \geq 1, \quad (3.151)$$

with $\tilde{x}_1^0 = x_1$ as an initial value. The mean-square prediction error can be approximated using (3.145) by noting that $\psi^*(z) = (1 - \lambda z)/(1 - z) = 1 + (1 - \lambda) \sum_{j=1}^{\infty} z^j$ for $|z| < 1$; consequently, for large n , (3.145) leads to

$$P_{n+m}^n \approx \sigma_w^2 [1 + (m-1)(1 - \lambda)^2].$$

In EWMA, the parameter $1 - \lambda$ is often called the smoothing parameter and is restricted to be between zero and one. Larger values of λ lead to smoother forecasts.

This method of forecasting is popular because it is easy to use; we need only retain the previous forecast value and the current observation to forecast the next time period. Unfortunately, as previously suggested, the method is often abused because some forecasters do not verify that the observations follow an IMA(1, 1) process, and often arbitrarily pick values of λ . In the following, we show how to generate 100 observations from an IMA(1,1) model with $\lambda = -\theta = .8$ and then calculate and display the fitted EWMA superimposed on the data. This is accomplished using the Holt-Winters command in R (see the help file [?HoltWinters](#) for details; no output is shown):

```
set.seed(666)
x = arima.sim(list(order = c(0,1,1), ma = -0.8), n = 100)
(x.ima = HoltWinters(x, beta=FALSE, gamma=FALSE)) # alpha below is 1 - lambda
  Smoothing parameter: alpha:  0.1663072
plot(x.ima)
```

3.7 Building ARIMA Models

There are a few basic steps to fitting ARIMA models to time series data. These steps involve

- plotting the data,
- possibly transforming the data,
- identifying the dependence orders of the model,
- parameter estimation,
- diagnostics, and
- model choice.

First, as with any data analysis, we should construct a time plot of the data, and inspect the graph for any anomalies. If, for example, the variability in the data grows with time, it will be necessary to transform the data to stabilize the variance. In such cases, the Box–Cox class of power transformations, equation (2.34), could be employed. Also, the particular application might suggest an appropriate transformation. For example, we have seen numerous examples where the data behave as $x_t = (1 + p_t)x_{t-1}$, where p_t is a small percentage change from period $t - 1$ to t , which may be negative. If p_t is a relatively stable process, then $\nabla \log(x_t) \approx p_t$ will be relatively stable. Frequently, $\nabla \log(x_t)$ is called the *return* or *growth rate*. This general idea was used in [Example 3.33](#), and we will use it again in [Example 3.39](#).

After suitably transforming the data, the next step is to identify preliminary values of the autoregressive order, p , the order of differencing, d , and the moving average order, q . A time plot of the data will typically suggest whether any differencing is needed. If differencing is called for, then difference the data once, $d = 1$, and inspect the time plot of ∇x_t . If additional differencing is necessary, then try differencing again and inspect a time plot of $\nabla^2 x_t$. Be careful not to overdifference because this may introduce dependence where none exists. For example, $x_t = w_t$ is serially uncorrelated, but $\nabla x_t = w_t - w_{t-1}$ is MA(1). In addition to time plots, the sample

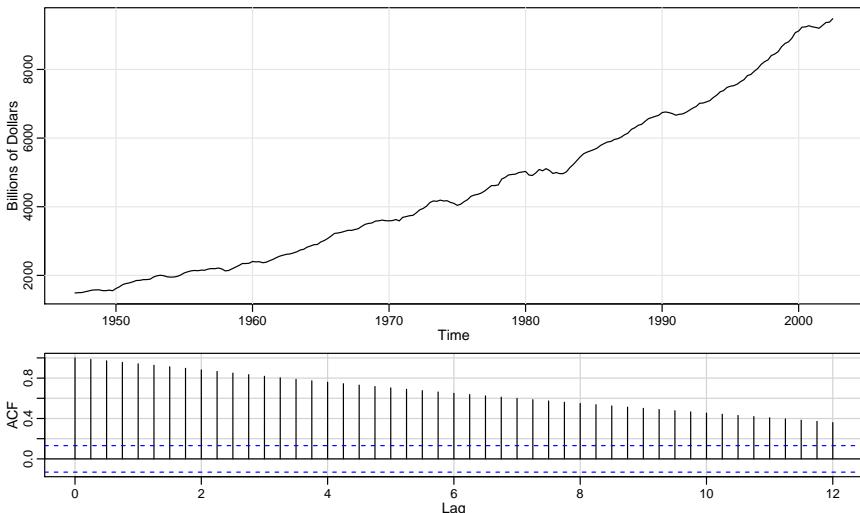


Fig. 3.13. Top: Quarterly U.S. GNP from 1947(1) to 2002(3). Bottom: Sample ACF of the GNP data. Lag is in terms of years.

ACF can help in indicating whether differencing is needed. Because the polynomial $\phi(z)(1 - z)^d$ has a unit root, the sample ACF, $\hat{\rho}(h)$, will not decay to zero fast as h increases. Thus, a slow decay in $\hat{\rho}(h)$ is an indication that differencing may be needed.

When preliminary values of d have been settled, the next step is to look at the sample ACF and PACF of $\nabla^d x_t$ for whatever values of d have been chosen. Using Table 3.1 as a guide, preliminary values of p and q are chosen. Note that it cannot be the case that both the ACF and PACF cut off. Because we are dealing with estimates, it will not always be clear whether the sample ACF or PACF is tailing off or cutting off. Also, two models that are seemingly different can actually be very similar. With this in mind, we should not worry about being so precise at this stage of the model fitting. At this point, a few preliminary values of p , d , and q should be at hand, and we can start estimating the parameters.

Example 3.39 Analysis of GNP Data

In this example, we consider the analysis of quarterly U.S. GNP from 1947(1) to 2002(3), $n = 223$ observations. The data are real U.S. gross national product in billions of chained 1996 dollars and have been seasonally adjusted. The data were obtained from the Federal Reserve Bank of St. Louis (<http://research.stlouisfed.org/>). Figure 3.13 shows a plot of the data, say, y_t . Because strong trend tends to obscure other effects, it is difficult to see any other variability in data except for periodic large dips in the economy. When reports of GNP and similar economic indicators are given, it is often in growth rate (percent change) rather than in actual (or adjusted) values that is of interest. The growth rate, say, $x_t = \nabla \log(y_t)$, is plotted in Figure 3.14, and it appears to be a stable process.

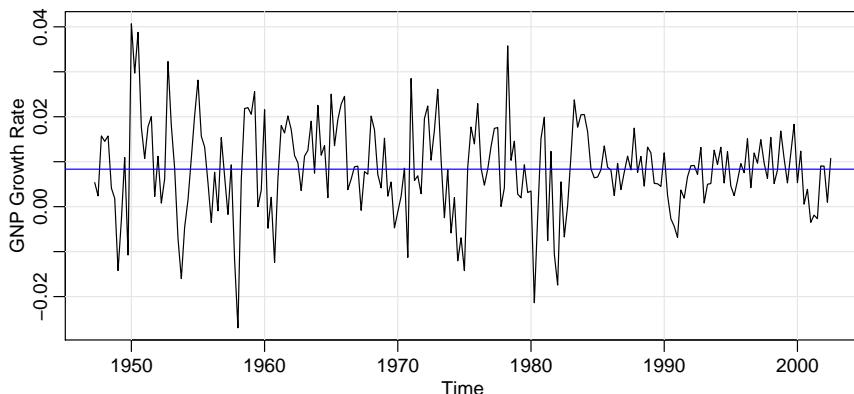


Fig. 3.14. U.S. GNP quarterly growth rate. The horizontal line displays the average growth of the process, which is close to 1%.

The sample ACF and PACF of the quarterly growth rate are plotted in [Figure 3.15](#). Inspecting the sample ACF and PACF, we might feel that the ACF is cutting off at lag 2 and the PACF is tailing off. This would suggest the GNP growth rate follows an MA(2) process, or log GNP follows an ARIMA(0, 1, 2) model. Rather than focus on one model, we will also suggest that it appears that the ACF is tailing off and the PACF is cutting off at lag 1. This suggests an AR(1) model for the growth rate, or ARIMA(1, 1, 0) for log GNP. As a preliminary analysis, we will fit both models.

Using MLE to fit the MA(2) model for the growth rate, x_t , the estimated model is

$$\hat{x}_t = .008_{(.001)} + .303_{(.065)}\hat{w}_{t-1} + .204_{(.064)}\hat{w}_{t-2} + \hat{w}_t, \quad (3.152)$$

where $\hat{\sigma}_w = .0094$ is based on 219 degrees of freedom. The values in parentheses are the corresponding estimated standard errors. All of the regression coefficients are significant, including the constant. *We make a special note of this because, as a default, some computer packages do not fit a constant in a differenced model.* That is, these packages assume, by default, that there is no drift. In this example, not including a constant leads to the wrong conclusions about the nature of the U.S. economy. Not including a constant assumes the average quarterly growth rate is zero, whereas the U.S. GNP average quarterly growth rate is about 1% (which can be seen easily in [Figure 3.14](#)). We leave it to the reader to investigate what happens when the constant is not included.

The estimated AR(1) model is

$$\hat{x}_t = .008_{(.001)}(1 - .347) + .347_{(.063)}\hat{x}_{t-1} + \hat{w}_t, \quad (3.153)$$

where $\hat{\sigma}_w = .0095$ on 220 degrees of freedom; note that the constant in (3.153) is $.008(1 - .347) = .005$.

We will discuss diagnostics next, but assuming both of these models fit well, how are we to reconcile the apparent differences of the estimated models (3.152)

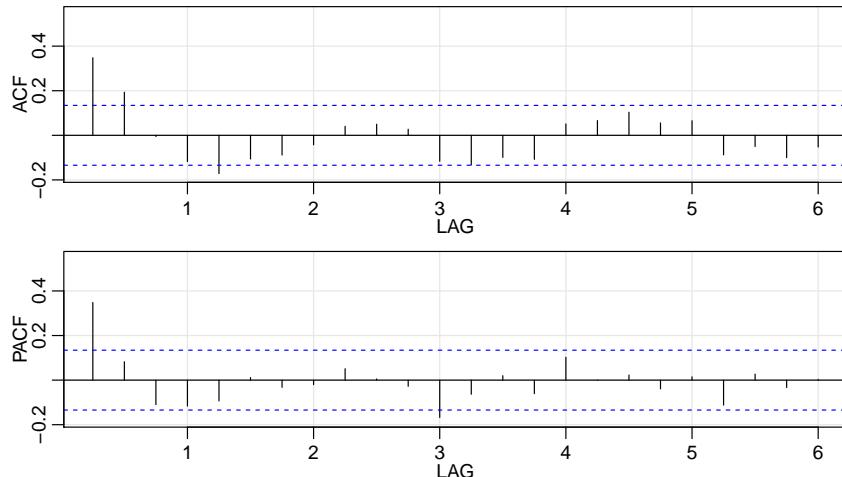


Fig. 3.15. Sample ACF and PACF of the GNP quarterly growth rate. Lag is in terms of years.

and (3.153)? In fact, the fitted models are nearly the same. To show this, consider an AR(1) model of the form in (3.153) without a constant term; that is,

$$x_t = .35x_{t-1} + w_t,$$

and write it in its causal form, $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$, where we recall $\psi_j = .35^j$. Thus, $\psi_0 = 1, \psi_1 = .350, \psi_2 = .123, \psi_3 = .043, \psi_4 = .015, \psi_5 = .005, \psi_6 = .002, \psi_7 = .001, \psi_8 = 0, \psi_9 = 0, \psi_{10} = 0$, and so forth. Thus,

$$x_t \approx .35w_{t-1} + .12w_{t-2} + w_t,$$

which is similar to the fitted MA(2) model in (3.153).

The analysis can be performed in R as follows.

```
plot(gnp)
acf2(gnp, 50)
gnpgr = diff(log(gnp)) # growth rate
plot(gnpgr)
acf2(gnpgr, 24)
sarima(gnpgr, 1, 0, 0) # AR(1)
sarima(gnpgr, 0, 0, 2) # MA(2)
ARMAtoMA(ar=.35, ma=0, 10) # prints psi-weights
```

The next step in model fitting is diagnostics. This investigation includes the analysis of the residuals as well as model comparisons. Again, the first step involves a time plot of the *innovations* (or residuals), $x_t - \hat{x}_t^{t-1}$, or of the *standardized innovations*

$$e_t = \left(x_t - \hat{x}_t^{t-1} \right) / \sqrt{\hat{P}_t^{t-1}}, \quad (3.154)$$

where \hat{x}_t^{t-1} is the one-step-ahead prediction of x_t based on the fitted model and \hat{P}_t^{t-1} is the estimated one-step-ahead error variance. If the model fits well, the standardized

residuals should behave as an iid sequence with mean zero and variance one. The time plot should be inspected for any obvious departures from this assumption. Unless the time series is Gaussian, it is not enough that the residuals are uncorrelated. For example, it is possible in the non-Gaussian case to have an uncorrelated process for which values contiguous in time are highly dependent. As an example, we mention the family of GARCH models that are discussed in [Chapter 5](#).

Investigation of marginal normality can be accomplished visually by looking at a histogram of the residuals. In addition to this, a normal probability plot or a Q-Q plot can help in identifying departures from normality. See Johnson and Wichern (1992, Chapter 4) for details of this test as well as additional tests for multivariate normality.

There are several tests of randomness, for example the runs test, that could be applied to the residuals. We could also inspect the sample autocorrelations of the residuals, say, $\hat{\rho}_e(h)$, for any patterns or large values. Recall that, for a white noise sequence, the sample autocorrelations are approximately independently and normally distributed with zero means and variances $1/n$. Hence, a good check on the correlation structure of the residuals is to plot $\hat{\rho}_e(h)$ versus h along with the error bounds of $\pm 2/\sqrt{n}$. The residuals from a model fit, however, will not quite have the properties of a white noise sequence and the variance of $\hat{\rho}_e(h)$ can be much less than $1/n$. Details can be found in Box and Pierce (1970) and McLeod (1978). This part of the diagnostics can be viewed as a visual inspection of $\hat{\rho}_e(h)$ with the main concern being the detection of obvious departures from the independence assumption.

In addition to plotting $\hat{\rho}_e(h)$, we can perform a general test that takes into consideration the magnitudes of $\hat{\rho}_e(h)$ as a group. For example, it may be the case that, individually, each $\hat{\rho}_e(h)$ is small in magnitude, say, each one is just slightly less than $2/\sqrt{n}$ in magnitude, but, collectively, the values are large. The *Ljung–Box–Pierce Q-statistic* given by

$$Q = n(n + 2) \sum_{h=1}^H \frac{\hat{\rho}_e^2(h)}{n - h} \quad (3.155)$$

can be used to perform such a test. The value H in (3.155) is chosen somewhat arbitrarily, typically, $H = 20$. Under the null hypothesis of model adequacy, asymptotically ($n \rightarrow \infty$), $Q \sim \chi_{H-p-q}^2$. Thus, we would reject the null hypothesis at level α if the value of Q exceeds the $(1 - \alpha)$ -quantile of the χ_{H-p-q}^2 distribution. Details can be found in Box and Pierce (1970), Ljung and Box (1978), and Davies et al. (1977). The basic idea is that if w_t is white noise, then by [Property 1.2](#), $n\hat{\rho}_w^2(h)$, for $h = 1, \dots, H$, are asymptotically independent χ_1^2 random variables. This means that $n \sum_{h=1}^H \hat{\rho}_w^2(h)$ is approximately a χ_H^2 random variable. Because the test involves the ACF of residuals from a model fit, there is a loss of $p + q$ degrees of freedom; the other values in (3.155) are used to adjust the statistic to better match the asymptotic chi-squared distribution.

Example 3.40 Diagnostics for GNP Growth Rate Example

We will focus on the MA(2) fit from [Example 3.39](#); the analysis of the AR(1) residuals is similar. [Figure 3.16](#) displays a plot of the standardized residuals, the ACF of the residuals, a boxplot of the standardized residuals, and the p-values

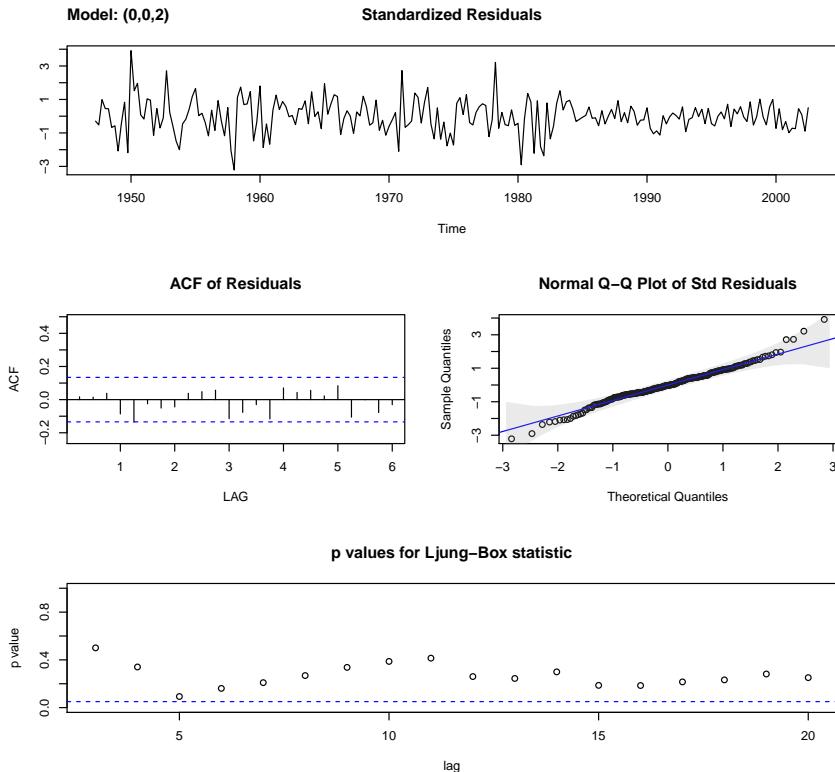


Fig. 3.16. Diagnostics of the residuals from MA(2) fit on GNP growth rate.

associated with the Q-statistic, (3.155), at lags $H = 3$ through $H = 20$ (with corresponding degrees of freedom $H - 2$).

Inspection of the time plot of the standardized residuals in Figure 3.16 shows no obvious patterns. Notice that there may be outliers, with a few values exceeding 3 standard deviations in magnitude. The ACF of the standardized residuals shows no apparent departure from the model assumptions, and the Q-statistic is never significant at the lags shown. The normal Q-Q plot of the residuals shows that the assumption of normality is reasonable, with the exception of the possible outliers.

The model appears to fit well. The diagnostics shown in Figure 3.16 are a by-product of the `sarima` command from the previous example.^{3.8}

^{3.8} The script `tsdiag` is available in R to run diagnostics for an ARIMA object, however, the script has errors and we do not recommend using it.

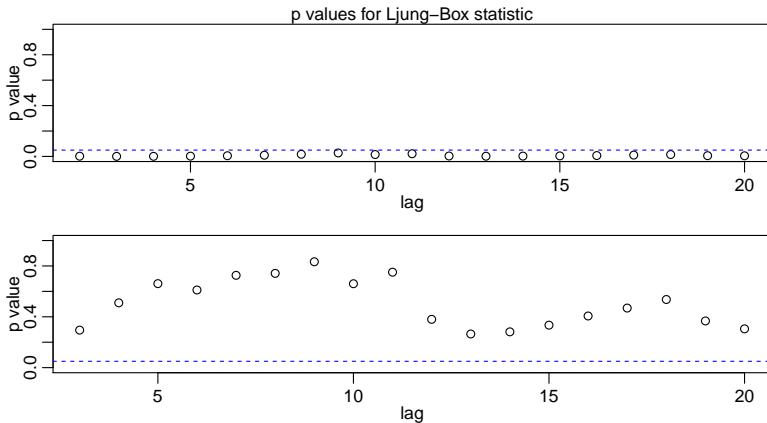


Fig. 3.17. Q-statistic p-values for the ARIMA(0, 1, 1) fit (top) and the ARIMA(1, 1, 1) fit (bottom) to the logged varve data.

Example 3.41 Diagnostics for the Glacial Varve Series

In Example 3.33, we fit an ARIMA(0, 1, 1) model to the logarithms of the glacial varve data and there appears to be a small amount of autocorrelation left in the residuals and the Q-tests are all significant; see Figure 3.17.

To adjust for this problem, we fit an ARIMA(1, 1, 1) to the logged varve data and obtained the estimates

$$\hat{\phi} = .23_{(.05)}, \hat{\theta} = -.89_{(.03)}, \hat{\sigma}_w^2 = .23.$$

Hence the AR term is significant. The Q-statistic p-values for this model are also displayed in Figure 3.17, and it appears this model fits the data well.

As previously stated, the diagnostics are byproducts of the individual `sarima` runs. We note that we did not fit a constant in either model because there is no apparent drift in the differenced, logged varve series. This fact can be verified by noting the constant is not significant when the command `no.constant=TRUE` is removed in the code:

```
sarima(log(varve), 0, 1, 1, no.constant=TRUE) # ARIMA(0, 1, 1)
sarima(log(varve), 1, 1, 1, no.constant=TRUE) # ARIMA(1, 1, 1)
```

In Example 3.39, we have two competing models, an AR(1) and an MA(2) on the GNP growth rate, that each appear to fit the data well. In addition, we might also consider that an AR(2) or an MA(3) might do better for forecasting. Perhaps combining both models, that is, fitting an ARMA(1, 2) to the GNP growth rate, would be the best. As previously mentioned, we have to be concerned with *overfitting* the model; it is not always the case that more is better. Overfitting leads to less-precise estimators, and adding more parameters may fit the data better but may also lead to bad forecasts. This result is illustrated in the following example.

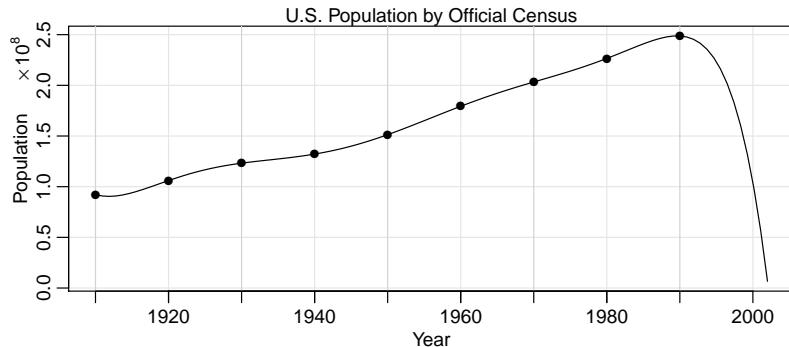


Fig. 3.18. A perfect fit and a terrible forecast.

Example 3.42 A Problem with Overfitting

Figure 3.18 shows the U.S. population by official census, every ten years from 1910 to 1990, as points. If we use these nine observations to predict the future population, we can use an eight-degree polynomial so the fit to the nine observations is perfect. The model in this case is

$$x_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots + \beta_8 t^8 + w_t.$$

The fitted line, which is plotted in the figure, passes through the nine observations. The model predicts that the population of the United States will be close to zero in the year 2000, and will cross zero sometime in the year 2002!

The final step of model fitting is model choice or model selection. That is, we must decide which model we will retain for forecasting. The most popular techniques, AIC, AICc, and BIC, were described in Section 2.1 in the context of regression models.

Example 3.43 Model Choice for the U.S. GNP Series

Returning to the analysis of the U.S. GNP data presented in Example 3.39 and Example 3.40, recall that two models, an AR(1) and an MA(2), fit the GNP growth rate well. To choose the final model, we compare the AIC, the AICc, and the BIC for both models. These values are a byproduct of the `sarima` runs displayed at the end of Example 3.39, but for convenience, we display them again here (recall the growth rate data are in `gnpgr`):

```
sarima(gnpgr, 1, 0, 0) # AR(1)
$AIC: -8.294403 $AICc: -8.284898 $BIC: -9.263748
sarima(gnpgr, 0, 0, 2) # MA(2)
$AIC: -8.297693 $AICc: -8.287854 $BIC: -9.251711
```

The AIC and AICc both prefer the MA(2) fit, whereas the BIC prefers the simpler AR(1) model. It is often the case that the BIC will select a model of smaller order than the AIC or AICc. In either case, it is not unreasonable to retain the AR(1) because pure autoregressive models are easier to work with.

3.8 Regression with Autocorrelated Errors

In [Section 2.1](#), we covered the classical regression model with uncorrelated errors w_t . In this section, we discuss the modifications that might be considered when the errors are correlated. That is, consider the regression model

$$y_t = \sum_{j=1}^r \beta_j z_{tj} + x_t \quad (3.156)$$

where x_t is a process with some covariance function $\gamma_x(s, t)$. In ordinary least squares, the assumption is that x_t is white Gaussian noise, in which case $\gamma_x(s, t) = 0$ for $s \neq t$ and $\gamma_x(t, t) = \sigma^2$, independent of t . If this is not the case, then weighted least squares should be used.

Write the model in vector notation, $y = Z\beta + x$, where $y = (y_1, \dots, y_n)'$ and $x = (x_1, \dots, x_n)'$ are $n \times 1$ vectors, $\beta = (\beta_1, \dots, \beta_r)'$ is $r \times 1$, and $Z = [z_1 | z_2 | \dots | z_n]'$ is the $n \times r$ matrix composed of the input variables. Let $\Gamma = \{\gamma_x(s, t)\}$, then $\Gamma^{-1/2}y = \Gamma^{-1/2}Z\beta + \Gamma^{-1/2}x$, so that we can write the model as

$$y^* = Z^*\beta + \delta,$$

where $y^* = \Gamma^{-1/2}y$, $Z^* = \Gamma^{-1/2}Z$, and $\delta = \Gamma^{-1/2}x$. Consequently, the covariance matrix of δ is the identity and the model is in the classical linear model form. It follows that the weighted estimate of β is $\hat{\beta}_w = (Z^{*'}Z^*)^{-1}Z^{*'}y^* = (Z'\Gamma^{-1}Z)^{-1}Z'\Gamma^{-1}y$, and the variance-covariance matrix of the estimator is $\text{var}(\hat{\beta}_w) = (Z'\Gamma^{-1}Z)^{-1}$. If x_t is white noise, then $\Gamma = \sigma^2 I$ and these results reduce to the usual least squares results.

In the time series case, it is often possible to assume a stationary covariance structure for the error process x_t that corresponds to a linear process and try to find an ARMA representation for x_t . For example, if we have a pure AR(p) error, then

$$\phi(B)x_t = w_t,$$

and $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ is the linear transformation that, when applied to the error process, produces the white noise w_t . Multiplying the regression equation through by the transformation $\phi(B)$ yields,

$$\underbrace{\phi(B)y_t}_{y_t^*} = \sum_{j=1}^r \beta_j \underbrace{\phi(B)z_{tj}}_{z_{tj}^*} + \underbrace{\phi(B)x_t}_{w_t},$$

and we are back to the linear regression model where the observations have been transformed so that $y_t^* = \phi(B)y_t$ is the dependent variable, $z_{tj}^* = \phi(B)z_{tj}$ for $j = 1, \dots, r$, are the independent variables, but the β s are the same as in the original model. For example, if $p = 1$, then $y_t^* = y_t - \phi y_{t-1}$ and $z_{tj}^* = z_{tj} - \phi z_{t-1,j}$.

In the AR case, we may set up the least squares problem as minimizing the error sum of squares

$$S(\phi, \beta) = \sum_{t=1}^n w_t^2 = \sum_{t=1}^n \left[\phi(B)y_t - \sum_{j=1}^r \beta_j \phi(B)z_{tj} \right]^2$$

with respect to all the parameters, $\phi = \{\phi_1, \dots, \phi_p\}$ and $\beta = \{\beta_1, \dots, \beta_r\}$. Of course, the optimization is performed using numerical methods.

If the error process is ARMA(p, q), i.e., $\phi(B)x_t = \theta(B)w_t$, then in the above discussion, we transform by $\pi(B)x_t = w_t$, where $\pi(B) = \theta(B)^{-1}\phi(B)$. In this case the error sum of squares also depends on $\theta = \{\theta_1, \dots, \theta_q\}$:

$$S(\phi, \theta, \beta) = \sum_{t=1}^n w_t^2 = \sum_{t=1}^n \left[\pi(B)y_t - \sum_{j=1}^r \beta_j \pi(B)z_{tj} \right]^2$$

At this point, the main problem is that we do not typically know the behavior of the noise x_t prior to the analysis. An easy way to tackle this problem was first presented in Cochrane and Orcutt (1949), and with the advent of cheap computing is modernized below:

- (i) First, run an ordinary regression of y_t on z_{t1}, \dots, z_{tr} (acting as if the errors are uncorrelated). Retain the residuals, $\hat{x}_t = y_t - \sum_{j=1}^r \hat{\beta}_j z_{tj}$.
- (ii) Identify ARMA model(s) for the residuals \hat{x}_t .
- (iii) Run weighted least squares (or MLE) on the regression model with autocorrelated errors using the model specified in step (ii).
- (iv) Inspect the residuals \hat{w}_t for whiteness, and adjust the model if necessary.

Example 3.44 Mortality, Temperature and Pollution

We consider the analyses presented in [Example 2.2](#), relating mean adjusted temperature T_t , and particulate levels P_t to cardiovascular mortality M_t . We consider the regression model

$$M_t = \beta_1 + \beta_2 t + \beta_3 T_t + \beta_4 T_t^2 + \beta_5 P_t + x_t, \quad (3.157)$$

where, for now, we assume that x_t is white noise. The sample ACF and PACF of the residuals from the ordinary least squares fit of (3.157) are shown in [Figure 3.19](#), and the results suggest an AR(2) model for the residuals.

Our next step is to fit the correlated error model (3.157), but where x_t is AR(2),

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

and w_t is white noise. The model can be fit using the `sarima` function as follows (partial output shown).

```
trend = time(cmort); temp = temp - mean(temp); temp2 = temp^2
summary(fit <- lm(cmort~trend + temp + temp2 + part, na.action=NULL))
acf2(resid(fit), 52) # implies AR2
sarima(cmort, 2,0,0, xreg=cbind(trend,temp,temp2,part))
Coefficients:
      ar1     ar2   intercept    trend      temp      temp2      part
      0.3848  0.4326  80.2116 -1.5165 -0.0190  0.0154  0.1545
  s.e.  0.0436  0.0400   1.8072  0.4226  0.0495  0.0020  0.0272
sigma^2 estimated as 26.01: loglikelihood = -1549.04, aic = 3114.07
```

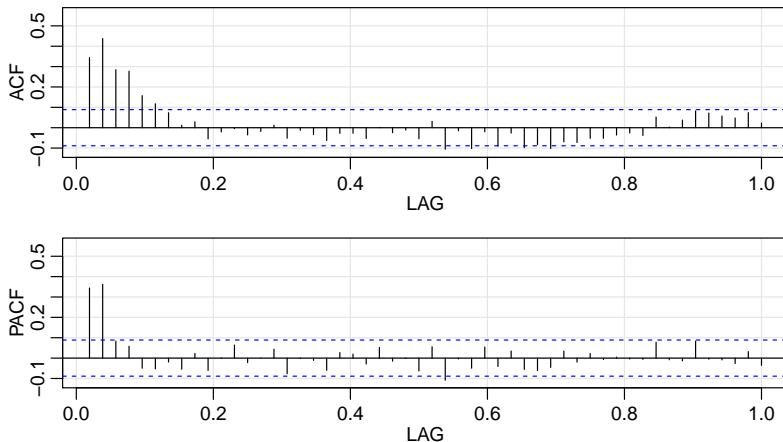


Fig. 3.19. Sample ACF and PACF of the mortality residuals indicating an AR(2) process.

The residual analysis output from `sarima` (not shown) shows no obvious departure of the residuals from whiteness.

Example 3.45 Regression with Lagged Variables (cont)

In Example 2.9 we fit the model

$$R_t = \beta_0 + \beta_1 S_{t-6} + \beta_2 D_{t-6} + \beta_3 D_{t-6} S_{t-6} + w_t,$$

where R_t is Recruitment, S_t is SOI, and D_t is a dummy variable that is 0 if $S_t < 0$ and 1 otherwise. However, residual analysis indicates that the residuals are not white noise. The sample (P)ACF of the residuals indicates that an AR(2) model might be appropriate, which is similar to the results of Example 3.44. We display partial results of the final model below.

```
dummy = ifelse(soi<0, 0, 1)
fish = ts.intersect(rec, soiL6=lag(soi,-6), dL6=lag(dummy,-6), dframe=TRUE)
summary(fit <- lm(rec ~soiL6*dL6, data=fish, na.action=NULL))
attach(fish)
plot(resid(fit))
acf2(resid(fit))      # indicates AR(2)
intract = soiL6*dL6   # interaction term
sarima(rec,2,0,0, xreg = cbind(soiL6, dL6, intract))
$ttable
  Estimate     SE  t.value p.value
ar1    1.3624 0.0440 30.9303 0.0000
ar2   -0.4703 0.0444 -10.5902 0.0000
intercept 64.8028 4.1121 15.7590 0.0000
soiL6    8.6671 2.2205  3.9033 0.0001
dL6    -2.5945 0.9535 -2.7209 0.0068
intract -10.3092 2.8311 -3.6415 0.0003
```

3.9 Multiplicative Seasonal ARIMA Models

In this section, we introduce several modifications made to the ARIMA model to account for seasonal and nonstationary behavior. Often, the dependence on the past tends to occur most strongly at multiples of some underlying seasonal lag s . For example, with monthly economic data, there is a strong yearly component occurring at lags that are multiples of $s = 12$, because of the strong connections of all activity to the calendar year. Data taken quarterly will exhibit the yearly repetitive period at $s = 4$ quarters. Natural phenomena such as temperature also have strong components corresponding to seasons. Hence, the natural variability of many physical, biological, and economic processes tends to match with seasonal fluctuations. Because of this, it is appropriate to introduce autoregressive and moving average polynomials that identify with the seasonal lags. The resulting *pure seasonal autoregressive moving average model*, say, $\text{ARMA}(P, Q)_s$, then takes the form

$$\Phi_P(B^s)x_t = \Theta_Q(B^s)w_t, \quad (3.158)$$

where the operators

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps} \quad (3.159)$$

and

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs} \quad (3.160)$$

are the **seasonal autoregressive operator** and the **seasonal moving average operator** of orders P and Q , respectively, with seasonal period s .

Analogous to the properties of nonseasonal ARMA models, the pure seasonal $\text{ARMA}(P, Q)_s$ is *causal* only when the roots of $\Phi_P(z^s)$ lie outside the unit circle, and it is *invertible* only when the roots of $\Theta_Q(z^s)$ lie outside the unit circle.

Example 3.46 A Seasonal AR Series

A first-order seasonal autoregressive series that might run over months could be written as

$$(1 - \Phi B^{12})x_t = w_t$$

or

$$x_t = \Phi x_{t-12} + w_t.$$

This model exhibits the series x_t in terms of past lags at the multiple of the yearly seasonal period $s = 12$ months. It is clear from the above form that estimation and forecasting for such a process involves only straightforward modifications of the unit lag case already treated. In particular, the causal condition requires $|\Phi| < 1$.

We simulated 3 years of data from the model with $\Phi = .9$, and exhibit the *theoretical ACF* and *PACF* of the model. See [Figure 3.20](#).

```
set.seed(666)
phi = c(rep(0,11),.9)
sAR = arima.sim(list(order=c(12,0,0), ar=phi), n=37)
sAR = ts(sAR, freq=12)
layout(matrix(c(1,1,2, 1,1,3), nc=2))
```

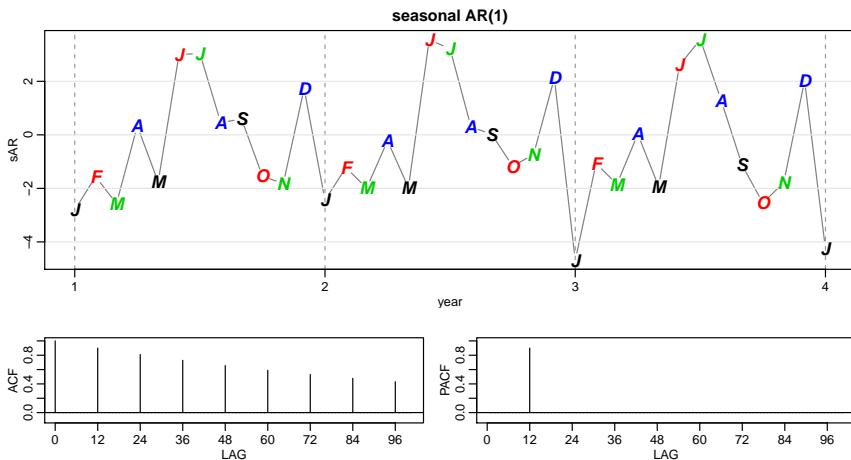


Fig. 3.20. Data generated from a seasonal ($s = 12$) AR(1), and the true ACF and PACF of the model $x_t = .9x_{t-12} + w_t$.

```

par(mar=c(3,3,2,1), mgp=c(1.6,.6,0))
plot(sAR, axes=FALSE, main='seasonal AR(1)', xlab="year", type='c')
Months = c("J","F","M","A","M","J","J","S","O","N","D")
points(sAR, pch=Months, cex=1.25, font=4, col=1:4)
axis(1, 1:4); abline(v=1:4, lty=2, col=gray(.7))
axis(2); box()
ACF = ARMAacf(ar=phi, ma=0, 100)
PACF = ARMAacf(ar=phi, ma=0, 100, pacf=TRUE)
plot(ACF, type="h", xlab="LAG", ylim=c(-.1,1)); abline(h=0)
plot(PACF, type="h", xlab="LAG", ylim=c(-.1,1)); abline(h=0)

```

For the first-order seasonal ($s = 12$) MA model, $x_t = w_t + \Theta w_{t-12}$, it is easy to verify that

$$\begin{aligned}\gamma(0) &= (1 + \Theta^2)\sigma^2 \\ \gamma(\pm 12) &= \Theta\sigma^2 \\ \gamma(h) &= 0, \quad \text{otherwise.}\end{aligned}$$

Thus, the only nonzero correlation, aside from lag zero, is

$$\rho(\pm 12) = \Theta/(1 + \Theta^2).$$

For the first-order seasonal ($s = 12$) AR model, using the techniques of the nonseasonal AR(1), we have

$$\begin{aligned}\gamma(0) &= \sigma^2/(1 - \Phi^2) \\ \gamma(\pm 12k) &= \sigma^2\Phi^k/(1 - \Phi^2) \quad k = 1, 2, \dots \\ \gamma(h) &= 0, \quad \text{otherwise.}\end{aligned}$$

In this case, the only non-zero correlations are

Table 3.3. Behavior of the ACF and PACF for Pure SARMA Models

	$\text{AR}(P)_s$	$\text{MA}(Q)_s$	$\text{ARMA}(P, Q)_s$
ACF*	Tails off at lags ks , $k = 1, 2, \dots$,	Cuts off after lag Qs	Tails off at lags ks
PACF*		Tails off at lags ks $k = 1, 2, \dots$,	Tails off at lags ks

*The values at nonseasonal lags $h \neq ks$, for $k = 1, 2, \dots$, are zero.

$$\rho(\pm 12k) = \Phi^k, \quad k = 0, 1, 2, \dots .$$

These results can be verified using the general result that $\gamma(h) = \Phi\gamma(h - 12)$, for $h \geq 1$. For example, when $h = 1$, $\gamma(1) = \Phi\gamma(11)$, but when $h = 11$, we have $\gamma(11) = \Phi\gamma(1)$, which implies that $\gamma(1) = \gamma(11) = 0$. In addition to these results, the PACF have the analogous extensions from nonseasonal to seasonal models. These results are demonstrated in [Figure 3.20](#).

As an initial diagnostic criterion, we can use the properties for the pure seasonal autoregressive and moving average series listed in [Table 3.3](#). These properties may be considered as generalizations of the properties for nonseasonal models that were presented in [Table 3.1](#).

In general, we can combine the seasonal and nonseasonal operators into a *multiplicative seasonal autoregressive moving average model*, denoted by $\text{ARMA}(p, q) \times (P, Q)_s$, and write

$$\Phi_P(B^s)\phi(B)x_t = \Theta_Q(B^s)\theta(B)w_t \quad (3.161)$$

as the overall model. Although the diagnostic properties in [Table 3.3](#) are not strictly true for the overall mixed model, the behavior of the ACF and PACF tends to show rough patterns of the indicated form. In fact, for mixed models, we tend to see a mixture of the facts listed in [Table 3.1](#) and [Table 3.3](#). In fitting such models, focusing on the seasonal autoregressive and moving average components first generally leads to more satisfactory results.

Example 3.47 A Mixed Seasonal Model

Consider an $\text{ARMA}(0, 1) \times (1, 0)_{12}$ model

$$x_t = \Phi x_{t-12} + w_t + \theta w_{t-1},$$

where $|\Phi| < 1$ and $|\theta| < 1$. Then, because x_{t-12} , w_t , and w_{t-1} are uncorrelated, and x_t is stationary, $\gamma(0) = \Phi^2\gamma(0) + \sigma_w^2 + \theta^2\sigma_w^2$, or

$$\gamma(0) = \frac{1 + \theta^2}{1 - \Phi^2} \sigma_w^2.$$

In addition, multiplying the model by x_{t-h} , $h > 0$, and taking expectations, we have $\gamma(1) = \Phi\gamma(11) + \theta\sigma_w^2$, and $\gamma(h) = \Phi\gamma(h - 12)$, for $h \geq 2$. Thus, the ACF for this model is

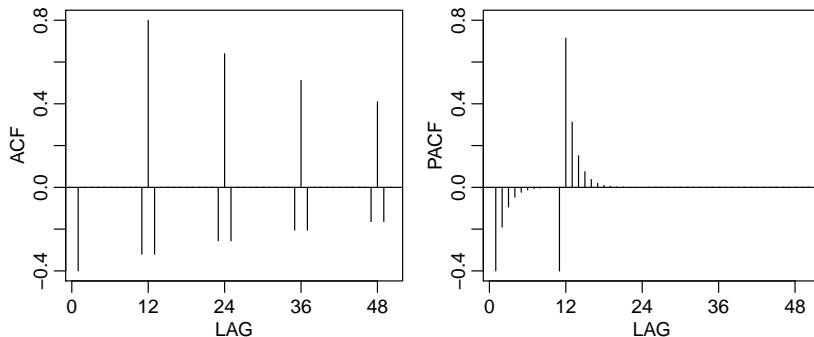


Fig. 3.21. ACF and PACF of the mixed seasonal ARMA model $x_t = .8x_{t-12} + w_t - .5w_{t-1}$.

$$\begin{aligned}\rho(12h) &= \Phi^h \quad h = 1, 2, \dots \\ \rho(12h-1) &= \rho(12h+1) = \frac{\theta}{1+\theta^2} \Phi^h \quad h = 0, 1, 2, \dots, \\ \rho(h) &= 0, \quad \text{otherwise.}\end{aligned}$$

The ACF and PACF for this model, with $\Phi = .8$ and $\theta = -.5$, are shown in Figure 3.21. These type of correlation relationships, although idealized here, are typically seen with seasonal data.

To reproduce Figure 3.21 in R, use the following commands:

```
phi = c(rep(.8, 11), .8)
ACF = ARMAacf(ar=phi, ma=-.5, 50)[-1]      # [-1] removes 0 lag
PACF = ARMAacf(ar=phi, ma=-.5, 50, pacf=TRUE)
par(mfrow=c(1,2))
plot(ACF, type="h", xlab="LAG", ylim=c(-.4,.8)); abline(h=0)
plot(PACF, type="h", xlab="LAG", ylim=c(-.4,.8)); abline(h=0)
```

Seasonal persistence occurs when the process is nearly periodic in the season. For example, with average monthly temperatures over the years, each January would be approximately the same, each February would be approximately the same, and so on. In this case, we might think of average monthly temperature x_t as being modeled as

$$x_t = S_t + w_t,$$

where S_t is a seasonal component that varies a little from one year to the next, according to a random walk,

$$S_t = S_{t-12} + v_t.$$

In this model, w_t and v_t are uncorrelated white noise processes. The tendency of data to follow this type of model will be exhibited in a sample ACF that is large and decays very slowly at lags $h = 12k$, for $k = 1, 2, \dots$. If we subtract the effect of successive years from each other, we find that

$$(1 - B^{12})x_t = x_t - x_{t-12} = v_t + w_t - w_{t-12}.$$

This model is a stationary MA(1)₁₂, and its ACF will have a peak only at lag 12. In general, seasonal differencing can be indicated when the ACF decays slowly at multiples of some season s , but is negligible between the periods. Then, a *seasonal difference of order D* is defined as

$$\nabla_s^D x_t = (1 - B^s)^D x_t, \quad (3.162)$$

where $D = 1, 2, \dots$, takes positive integer values. Typically, $D = 1$ is sufficient to obtain seasonal stationarity. Incorporating these ideas into a general model leads to the following definition.

Definition 3.12 *The multiplicative seasonal autoregressive integrated moving average model, or SARIMA model is given by*

$$\Phi_P(B^s)\phi(B)\nabla_s^D \nabla^d x_t = \delta + \Theta_Q(B^s)\theta(B)w_t, \quad (3.163)$$

where w_t is the usual Gaussian white noise process. The general model is denoted as **ARIMA**(p, d, q) \times (P, D, Q) _{s} . The ordinary autoregressive and moving average components are represented by polynomials $\phi(B)$ and $\theta(B)$ of orders p and q , respectively, and the seasonal autoregressive and moving average components by $\Phi_P(B^s)$ and $\Theta_Q(B^s)$ of orders P and Q and ordinary and seasonal difference components by $\nabla^d = (1 - B)^d$ and $\nabla_s^D = (1 - B^s)^D$.

Example 3.48 An SARIMA Model

Consider the following model, which often provides a reasonable representation for seasonal, nonstationary, economic time series. We exhibit the equations for the model, denoted by ARIMA(0, 1, 1) \times (0, 1, 1)₁₂ in the notation given above, where the seasonal fluctuations occur every 12 months. Then, with $\delta = 0$, the model (3.163) becomes

$$\nabla_{12}\nabla x_t = \Theta(B^{12})\theta(B)w_t$$

or

$$(1 - B^{12})(1 - B)x_t = (1 + \Theta B^{12})(1 + \theta B)w_t. \quad (3.164)$$

Expanding both sides of (3.164) leads to the representation

$$(1 - B - B^{12} + B^{13})x_t = (1 + \theta B + \Theta B^{12} + \Theta\theta B^{13})w_t,$$

or in difference equation form

$$x_t = x_{t-1} + x_{t-12} - x_{t-13} + w_t + \theta w_{t-1} + \Theta w_{t-12} + \Theta\theta w_{t-13}.$$

Note that the multiplicative nature of the model implies that the coefficient of w_{t-13} is the product of the coefficients of w_{t-1} and w_{t-12} rather than a free parameter. The multiplicative model assumption seems to work well with many seasonal time series data sets while reducing the number of parameters that must be estimated.

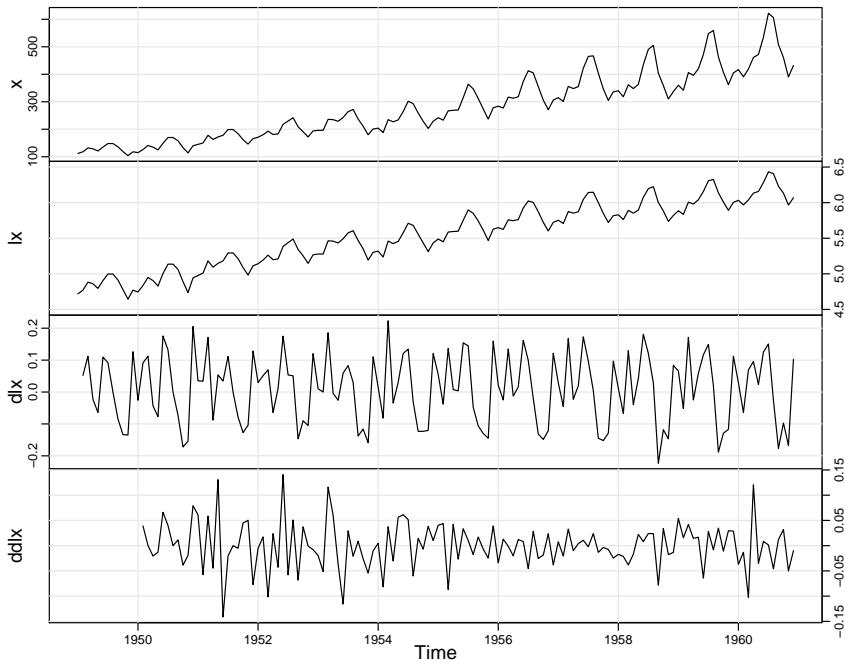


Fig. 3.22. R data set `AirPassengers`, which are the monthly totals of international airline passengers x , and the transformed data: $lx = \log x_t$, $dlx = \nabla \log x_t$, and $ddlx = \nabla_{12} \nabla \log x_t$.

Selecting the appropriate model for a given set of data from all of those represented by the general form (3.163) is a daunting task, and we usually think first in terms of finding difference operators that produce a roughly stationary series and then in terms of finding a set of simple autoregressive moving average or multiplicative seasonal ARMA to fit the resulting residual series. Differencing operations are applied first, and then the residuals are constructed from a series of reduced length. Next, the ACF and the PACF of these residuals are evaluated. Peaks that appear in these functions can often be eliminated by fitting an autoregressive or moving average component in accordance with the general properties of Table 3.1 and Table 3.3. In considering whether the model is satisfactory, the diagnostic techniques discussed in Section 3.7 still apply.

Example 3.49 Air Passengers

We consider the R data set `AirPassengers`, which are the monthly totals of international airline passengers, 1949 to 1960, taken from Box & Jenkins (1970). Various plots of the data and transformed data are shown in Figure 3.22 and were obtained as follows:

```
x = AirPassengers
lx = log(x); dlx = diff(lx); ddx = diff(dlx, 12)
```

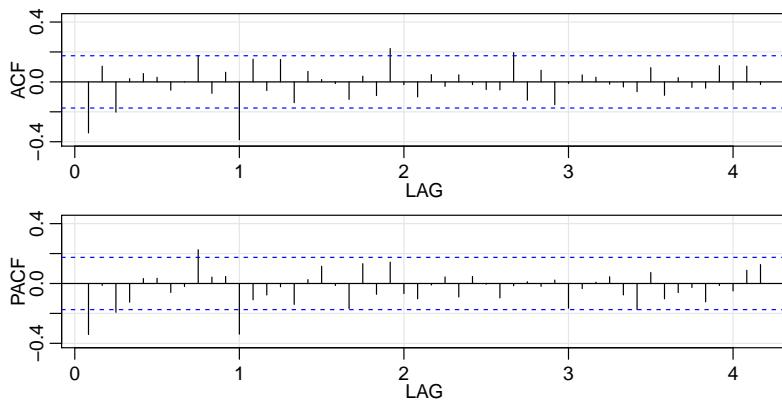


Fig. 3.23. Sample ACF and PACF of dd1x ($\nabla_{12} \nabla \log x_t$).

```
plot.ts(cbind(x, lx, dlx, dd1x), main="")
# below of interest for showing seasonal RW (not shown here):
par(mfrow=c(2,1))
monthplot(dlx); monthplot(dd1x)
```

Note that x is the original series, which shows trend plus increasing variance. The logged data are in lx , and the transformation stabilizes the variance. The logged data are then differenced to remove trend, and are stored in dlx . It is clear there is still persistence in the seasons (i.e., $dlx_t \approx dlx_{t-12}$), so that a twelfth-order difference is applied and stored in $dd1x$. The transformed data appears to be stationary and we are now ready to fit a model.

The sample ACF and PACF of dd1x ($\nabla_{12} \nabla \log x_t$) are shown in Figure 3.23. The R code is:

```
acf2(dd1x, 50)
```

Seasonal Component: It appears that at the seasons, the ACF is cutting off a lag $1s$ ($s = 12$), whereas the PACF is tailing off at lags $1s, 2s, 3s, 4s, \dots$. These results implies an SMA(1), $P = 0$, $Q = 1$, in the season ($s = 12$).

Non-Seasonal Component: Inspecting the sample ACF and PACF at the lower lags, it appears as though both are tailing off. This suggests an ARMA(1, 1) within the seasons, $p = q = 1$.

Thus, we first try an $\text{ARIMA}(1, 1, 1) \times (0, 1, 1)_{12}$ on the logged data:

```
 sarima(lx, 1, 1, 1, 0, 1, 1, 12)
Coefficients:
      ar1      ma1      sma1
    0.1960   -0.5784   -0.5643
  s.e.  0.2475    0.2132    0.0747
sigma^2 estimated as 0.001341
$AIC -5.5726 $AICc -5.556713 $BIC -6.510729
```

However, the AR parameter is not significant, so we should try dropping one parameter from the within seasons part. In this case, we try both an $\text{ARIMA}(0, 1, 1) \times (0, 1, 1)_{12}$ and an $\text{ARIMA}(1, 1, 0) \times (0, 1, 1)_{12}$ model:

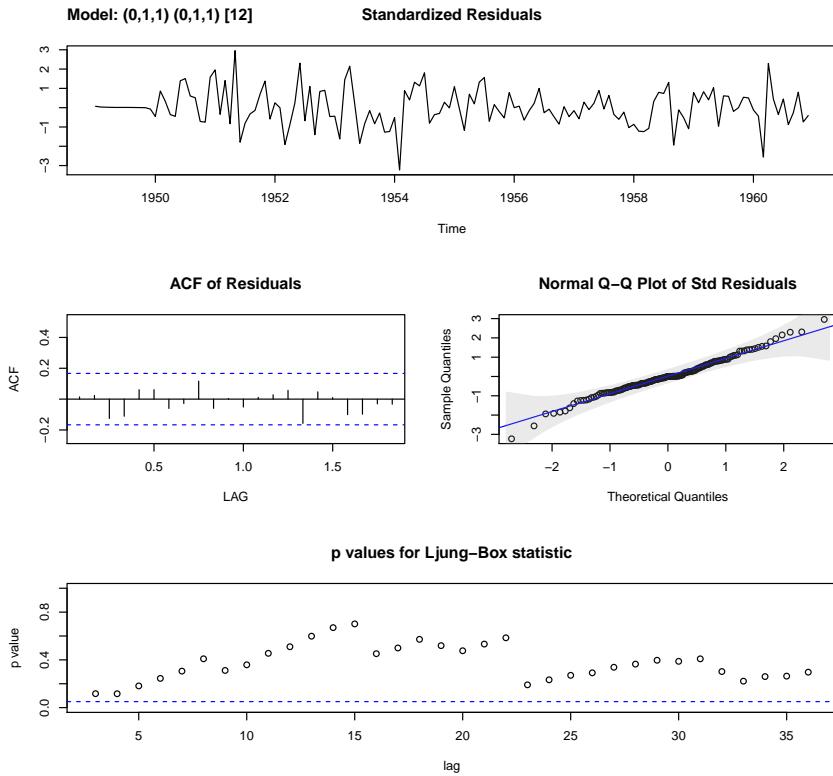


Fig. 3.24. Residual analysis for the $\text{ARIMA}(0, 1, 1) \times (0, 1, 1)_{12}$ fit to the logged air passengers data set.

```

sarima(lx, 0,1,1, 0,1,1,12)
Coefficients:
      m1       sm1
     -0.4018   -0.5569
  s.e.  0.0896   0.0731
sigma^2 estimated as 0.001348
$AIC -5.58133 $AICc -5.56625 $BIC -6.540082
sarima(lx, 1,1,0, 0,1,1,12)
Coefficients:
      ar1       sm1
     -0.3395   -0.5619
  s.e.  0.0822   0.0748
sigma^2 estimated as 0.001367
$AIC -5.567081 $AICc -5.552002 $BIC -6.525834

```

All information criteria prefer the $\text{ARIMA}(0, 1, 1) \times (0, 1, 1)_{12}$ model, which is the model displayed in (3.164). The residual diagnostics are shown in Figure 3.24, and except for one or two outliers, the model seems to fit well.

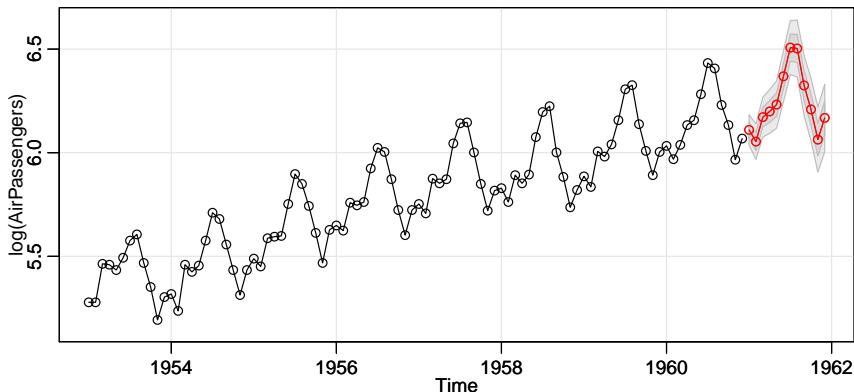


Fig. 3.25. Twelve month forecast using the ARIMA(0, 1, 1) \times (0, 1, 1)₁₂ model on the logged air passenger data set.

Finally, we forecast the logged data out twelve months, and the results are shown in Figure 3.25.

```
sarima.fore(lx, 12, 0, 1, 1, 0, 1, 1, 12)
```

Problems

Section 3.1

3.1 For an MA(1), $x_t = w_t + \theta w_{t-1}$, show that $|\rho_x(1)| \leq 1/2$ for any number θ . For which values of θ does $\rho_x(1)$ attain its maximum and minimum?

3.2 Let $\{w_t; t = 0, 1, \dots\}$ be a white noise process with variance σ_w^2 and let $|\phi| < 1$ be a constant. Consider the process $x_0 = w_0$, and

$$x_t = \phi x_{t-1} + w_t, \quad t = 1, 2, \dots.$$

We might use this method to simulate an AR(1) process from simulated white noise.

- (a) Show that $x_t = \sum_{j=0}^t \phi^j w_{t-j}$ for any $t = 0, 1, \dots$.
- (b) Find the $E(x_t)$.
- (c) Show that, for $t = 0, 1, \dots$,

$$\text{var}(x_t) = \frac{\sigma_w^2}{1 - \phi^2} (1 - \phi^{2(t+1)})$$

- (d) Show that, for $h \geq 0$,

$$\text{cov}(x_{t+h}, x_t) = \phi^h \text{var}(x_t)$$

- (e) Is x_t stationary?

- (f) Argue that, as $t \rightarrow \infty$, the process becomes stationary, so in a sense, x_t is “asymptotically stationary.”
- (g) Comment on how you could use these results to simulate n observations of a stationary Gaussian AR(1) model from simulated iid $N(0,1)$ values.
- (h) Now suppose $x_0 = w_0/\sqrt{1 - \phi^2}$. Is this process stationary? Hint: Show $\text{var}(x_t)$ is constant.

3.3 Verify the calculations made in [Example 3.4](#) as follows.

- (a) Let $x_t = \phi x_{t-1} + w_t$ where $|\phi| > 1$ and $w_t \sim \text{iid } N(0, \sigma_w^2)$. Show $E(x_t) = 0$ and $\gamma_x(h) = \sigma_w^2 \phi^{-2} \phi^{-h}/(1 - \phi^{-2})$ for $h \geq 0$.
- (b) Let $y_t = \phi^{-1} y_{t-1} + v_t$ where $v_t \sim \text{iid } N(0, \sigma_w^2 \phi^{-2})$ and ϕ and σ_w are as in part (a). Argue that y_t is causal with the same mean function and autocovariance function as x_t .

3.4 Identify the following models as ARMA(p, q) models (watch out for parameter redundancy), and determine whether they are causal and/or invertible:

- (a) $x_t = .80x_{t-1} - .15x_{t-2} + w_t - .30w_{t-1}$.
- (b) $x_t = x_{t-1} - .50x_{t-2} + w_t - w_{t-1}$.

3.5 Verify the causal conditions for an AR(2) model given in [\(3.28\)](#). That is, show that an AR(2) is causal if and only if [\(3.28\)](#) holds.

Section 3.2

3.6 For the AR(2) model given by $x_t = -.9x_{t-2} + w_t$, find the roots of the autoregressive polynomial, and then plot the ACF, $\rho(h)$.

3.7 For the AR(2) series shown below, use the results of [Example 3.10](#) to determine a set of difference equations that can be used to find the ACF $\rho(h)$, $h = 0, 1, \dots$; solve for the constants in the ACF using the initial conditions. Then plot the ACF values to lag 10 (use [ARMAacf](#) as a check on your answers).

- (a) $x_t + 1.6x_{t-1} + .64x_{t-2} = w_t$.
- (b) $x_t - .40x_{t-1} - .45x_{t-2} = w_t$.
- (c) $x_t - 1.2x_{t-1} + .85x_{t-2} = w_t$.

Section 3.3

3.8 Verify the calculations for the autocorrelation function of an ARMA(1, 1) process given in [Example 3.14](#). Compare the form with that of the ACF for the ARMA(1, 0) and the ARMA(0, 1) series. Plot the ACFs of the three series on the same graph for $\phi = .6$, $\theta = .9$, and comment on the diagnostic capabilities of the ACF in this case.

3.9 Generate $n = 100$ observations from each of the three models discussed in [Problem 3.8](#). Compute the sample ACF for each model and compare it to the theoretical values. Compute the sample PACF for each of the generated series and compare the sample ACFs and PACFs with the general results given in [Table 3.1](#).

Section 3.4

3.10 Let x_t represent the cardiovascular mortality series (`cmort`) discussed in Example 2.2.

- (a) Fit an AR(2) to x_t using linear regression as in Example 3.18.
- (b) Assuming the fitted model in (a) is the true model, find the forecasts over a four-week horizon, x_{n+m}^n , for $m = 1, 2, 3, 4$, and the corresponding 95% prediction intervals.

3.11 Consider the MA(1) series

$$x_t = w_t + \theta w_{t-1},$$

where w_t is white noise with variance σ_w^2 .

- (a) Derive the minimum mean-square error one-step forecast based on the infinite past, and determine the mean-square error of this forecast.
- (b) Let \tilde{x}_{n+1}^n be the truncated one-step-ahead forecast as given in (3.92). Show that

$$E[(x_{n+1} - \tilde{x}_{n+1}^n)^2] = \sigma^2(1 + \theta^{2+2n}).$$

Compare the result with (a), and indicate how well the finite approximation works in this case.

3.12 In the context of equation (3.63), show that, if $\gamma(0) > 0$ and $\gamma(h) \rightarrow 0$ as $h \rightarrow \infty$, then Γ_n is positive definite.

3.13 Suppose x_t is stationary with zero mean and recall the definition of the PACF given by (3.55) and (3.56). That is, let

$$\epsilon_t = x_t - \sum_{i=1}^{h-1} a_i x_{t-i} \quad \text{and} \quad \delta_{t-h} = x_{t-h} - \sum_{j=1}^{h-1} b_j x_{t-j}$$

be the two residuals where $\{a_1, \dots, a_{h-1}\}$ and $\{b_1, \dots, b_{h-1}\}$ are chosen so that they minimize the mean-squared errors

$$E[\epsilon_t^2] \quad \text{and} \quad E[\delta_{t-h}^2].$$

The PACF at lag h was defined as the cross-correlation between ϵ_t and δ_{t-h} ; that is,

$$\phi_{hh} = \frac{E(\epsilon_t \delta_{t-h})}{\sqrt{E(\epsilon_t^2)E(\delta_{t-h}^2)}}.$$

Let R_h be the $h \times h$ matrix with elements $\rho(i-j)$ for $i, j = 1, \dots, h$, and let $\rho_h = (\rho(1), \rho(2), \dots, \rho(h))'$ be the vector of lagged autocorrelations, $\rho(h) = \text{corr}(x_{t+h}, x_t)$. Let $\tilde{\rho}_h = (\rho(h), \rho(h-1), \dots, \rho(1))'$ be the reversed vector. In addition, let x_t^h denote the BLP of x_t given $\{x_{t-1}, \dots, x_{t-h}\}$:

$$x_t^h = \alpha_{h1}x_{t-1} + \cdots + \alpha_{hh}x_{t-h},$$

as described in [Property 3.3](#). Prove

$$\phi_{hh} = \frac{\rho(h) - \tilde{\rho}'_{h-1} R_{h-1}^{-1} \rho_h}{1 - \tilde{\rho}'_{h-1} R_{h-1}^{-1} \tilde{\rho}_{h-1}} = \alpha_{hh}.$$

In particular, this result proves [Property 3.4](#).

Hint: Divide the prediction equations [see (3.63)] by $\gamma(0)$ and write the matrix equation in the partitioned form as

$$\begin{pmatrix} R_{h-1} & \tilde{\rho}_{h-1} \\ \tilde{\rho}'_{h-1} & \rho(0) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_{hh} \end{pmatrix} = \begin{pmatrix} \rho_{h-1} \\ \rho(h) \end{pmatrix},$$

where the $h \times 1$ vector of coefficients $\alpha = (\alpha_{h1}, \dots, \alpha_{hh})'$ is partitioned as $\alpha = (\alpha'_1, \alpha_{hh})'$.

3.14 Suppose we wish to find a prediction function $g(x)$ that minimizes

$$MSE = E[(y - g(x))^2],$$

where x and y are jointly distributed random variables with density function $f(x, y)$.

(a) Show that MSE is minimized by the choice

$$g(x) = E(y \mid x).$$

Hint:

$$MSE = EE[(y - g(x))^2 \mid x].$$

(b) Apply the above result to the model

$$y = x^2 + z,$$

where x and z are independent zero-mean normal variables with variance one.

Show that $MSE = 1$.

(c) Suppose we restrict our choices for the function $g(x)$ to linear functions of the form

$$g(x) = a + bx$$

and determine a and b to minimize MSE . Show that $a = 1$ and

$$b = \frac{E(xy)}{E(x^2)} = 0$$

and $MSE = 3$. What do you interpret this to mean?

3.15 For an AR(1) model, determine the general form of the m -step-ahead forecast x_{t+m}^t and show

$$E[(x_{t+m} - x_{t+m}^t)^2] = \sigma_w^2 \frac{1 - \phi^{2m}}{1 - \phi^2}.$$

3.16 Consider the ARMA(1,1) model discussed in [Example 3.8](#), equation (3.27); that is, $x_t = .9x_{t-1} + .5w_{t-1} + w_t$. Show that truncated prediction as defined in (3.91) is equivalent to truncated prediction using the recursive formula (3.92).

3.17 Verify statement (3.87), that for a fixed sample size, the ARMA prediction errors are correlated.

Section 3.5

3.18 Fit an AR(2) model to the cardiovascular mortality series ([cmort](#)) discussed in [Example 2.2](#). using linear regression and using Yule–Walker.

- (a) Compare the parameter estimates obtained by the two methods.
- (b) Compare the estimated standard errors of the coefficients obtained by linear regression with their corresponding asymptotic approximations, as given in [Property 3.10](#).

3.19 Suppose x_1, \dots, x_n are observations from an AR(1) process with $\mu = 0$.

- (a) Show the backcasts can be written as $x_t^n = \phi^{1-t} x_1$, for $t \leq 1$.
- (b) In turn, show, for $t \leq 1$, the backcasted errors are

$$\tilde{w}_t(\phi) = x_t^n - \phi x_{t-1}^n = \phi^{1-t}(1 - \phi^2)x_1.$$

- (c) Use the result of (b) to show $\sum_{t=-\infty}^1 \tilde{w}_t^2(\phi) = (1 - \phi^2)x_1^2$.
- (d) Use the result of (c) to verify the unconditional sum of squares, $S(\phi)$, can be written as $\sum_{t=-\infty}^n \tilde{w}_t^2(\phi)$.
- (e) Find x_t^{t-1} and r_t for $1 \leq t \leq n$, and show that

$$S(\phi) = \sum_{t=1}^n (x_t - x_t^{t-1})^2 / r_t.$$

3.20 Repeat the following numerical exercise three times. Generate $n = 500$ observations from the ARMA model given by

$$x_t = .9x_{t-1} + w_t - .9w_{t-1},$$

with $w_t \sim \text{iid } N(0, 1)$. Plot the simulated data, compute the sample ACF and PACF of the simulated data, and fit an ARMA(1, 1) model to the data. What happened and how do you explain the results?

3.21 Generate 10 realizations of length $n = 200$ each of an ARMA(1,1) process with $\phi = .9, \theta = .5$ and $\sigma^2 = 1$. Find the MLEs of the three parameters in each case and compare the estimators to the true values.

3.22 Generate $n = 50$ observations from a Gaussian AR(1) model with $\phi = .99$ and $\sigma_w = 1$. Using an estimation technique of your choice, compare the approximate asymptotic distribution of your estimate (the one you would use for inference) with the results of a bootstrap experiment (use $B = 200$).

3.23 Using Example 3.32 as your guide, find the Gauss–Newton procedure for estimating the autoregressive parameter, ϕ , from the AR(1) model, $x_t = \phi x_{t-1} + w_t$, given data x_1, \dots, x_n . Does this procedure produce the unconditional or the conditional estimator? Hint: Write the model as $w_t(\phi) = x_t - \phi x_{t-1}$; your solution should work out to be a non-recursive procedure.

3.24 Consider the stationary series generated by

$$x_t = \alpha + \phi x_{t-1} + w_t + \theta w_{t-1},$$

where $E(x_t) = \mu$, $|\theta| < 1$, $|\phi| < 1$ and the w_t are iid random variables with zero mean and variance σ_w^2 .

- (a) Determine the mean as a function of α for the above model. Find the autocovariance and ACF of the process x_t , and show that the process is weakly stationary. Is the process strictly stationary?
- (b) Prove the limiting distribution as $n \rightarrow \infty$ of the sample mean,

$$\bar{x} = n^{-1} \sum_{t=1}^n x_t,$$

is normal, and find its limiting mean and variance in terms of α , ϕ , θ , and σ_w^2 .
(Note: This part uses results from Appendix A.)

3.25 A problem of interest in the analysis of geophysical time series involves a simple model for observed data containing a signal and a reflected version of the signal with unknown amplification factor a and unknown time delay δ . For example, the depth of an earthquake is proportional to the time delay δ for the P wave and its reflected form pP on a seismic record. Assume the signal, say s_t , is white and Gaussian with variance σ_s^2 , and consider the generating model

$$x_t = s_t + a s_{t-\delta}.$$

- (a) Prove the process x_t is stationary. If $|a| < 1$, show that

$$s_t = \sum_{j=0}^{\infty} (-a)^j x_{t-\delta j}$$

is a mean square convergent representation for the signal s_t , for $t = 1, \pm 1, \pm 2, \dots$

- (b) If the time delay δ is assumed to be known, suggest an approximate computational method for estimating the parameters a and σ_s^2 using maximum likelihood and the Gauss–Newton method.
- (c) If the time delay δ is an unknown integer, specify how we could estimate the parameters including δ . Generate a $n = 500$ point series with $a = .9$, $\sigma_w^2 = 1$ and $\delta = 5$. Estimate the integer time delay δ by searching over $\delta = 3, 4, \dots, 7$.

3.26 Forecasting with estimated parameters: Let x_1, x_2, \dots, x_n be a sample of size n from a causal AR(1) process, $x_t = \phi x_{t-1} + w_t$. Let $\hat{\phi}$ be the Yule–Walker estimator of ϕ .

- (a) Show $\hat{\phi} - \phi = O_p(n^{-1/2})$. See [Appendix A](#) for the definition of $O_p(\cdot)$.
- (b) Let x_{n+1}^n be the one-step-ahead forecast of x_{n+1} given the data x_1, \dots, x_n , based on the known parameter, ϕ , and let \hat{x}_{n+1}^n be the one-step-ahead forecast when the parameter is replaced by $\hat{\phi}$. Show $x_{n+1}^n - \hat{x}_{n+1}^n = O_p(n^{-1/2})$.

Section 3.6

3.27 Suppose

$$y_t = \beta_0 + \beta_1 t + \cdots + \beta_q t^q + x_t, \quad \beta_q \neq 0,$$

where x_t is stationary. First, show that $\nabla^k x_t$ is stationary for any $k = 1, 2, \dots$, and then show that $\nabla^k y_t$ is not stationary for $k < q$, but is stationary for $k \geq q$.

3.28 Verify that the IMA(1,1) model given in [\(3.148\)](#) can be inverted and written as [\(3.149\)](#).

3.29 For the ARIMA(1, 1, 0) model with drift, $(1 - \phi B)(1 - B)x_t = \delta + w_t$, let $y_t = (1 - B)x_t = \nabla x_t$.

- (a) Noting that y_t is AR(1), show that, for $j \geq 1$,

$$y_{n+j}^n = \delta [1 + \phi + \cdots + \phi^{j-1}] + \phi^j y_n.$$

- (b) Use part (a) to show that, for $m = 1, 2, \dots$,

$$x_{n+m}^n = x_n + \frac{\delta}{1 - \phi} \left[m - \frac{\phi(1 - \phi^m)}{(1 - \phi)} \right] + (x_n - x_{n-1}) \frac{\phi(1 - \phi^m)}{(1 - \phi)}.$$

Hint: From (a), $x_{n+j}^n - x_{n+j-1}^n = \delta \frac{1 - \phi^j}{1 - \phi} + \phi^j(x_n - x_{n-1})$. Now sum both sides over j from 1 to m .

- (c) Use [\(3.145\)](#) to find P_{n+m}^n by first showing that $\psi_0^* = 1$, $\psi_1^* = (1 + \phi)$, and $\psi_j^* - (1 + \phi)\psi_{j-1}^* + \phi\psi_{j-2}^* = 0$ for $j \geq 2$, in which case $\psi_j^* = \frac{1 - \phi^{j+1}}{1 - \phi}$, for $j \geq 1$. Note that, as in [Example 3.37](#), equation [\(3.145\)](#) is exact here.

3.30 For the logarithm of the glacial varve data, say, x_t , presented in [Example 3.33](#), use the first 100 observations and calculate the EWMA, \tilde{x}_{t+1}^t , given in [\(3.151\)](#) for $t = 1, \dots, 100$, using $\lambda = .25, .50$, and $.75$, and plot the EWMA and the data superimposed on each other. Comment on the results.

Section 3.7

3.31 In Example 3.40, we presented the diagnostics for the MA(2) fit to the GNP growth rate series. Using that example as a guide, complete the diagnostics for the AR(1) fit.

3.32 Crude oil prices in dollars per barrel are in `oil`. Fit an ARIMA(p, d, q) model to the growth rate performing all necessary diagnostics. Comment.

3.33 Fit an ARIMA(p, d, q) model to the global temperature data `globtemp` performing all of the necessary diagnostics. After deciding on an appropriate model, forecast (with limits) the next 10 years. Comment.

3.34 Fit an ARIMA(p, d, q) model to the sulfur dioxide series, `so2`, performing all of the necessary diagnostics. After deciding on an appropriate model, forecast the data into the future four time periods ahead (about one month) and calculate 95% prediction intervals for each of the four forecasts. Comment. (Sulfur dioxide is one of the pollutants monitored in the mortality study described in Example 2.2.)

Section 3.8

3.35 Let S_t represent the monthly sales data in `sales` ($n = 150$), and let L_t be the leading indicator in `lead`.

- (a) Fit an ARIMA model to S_t , the monthly sales data. Discuss your model fitting in a step-by-step fashion, presenting your (A) initial examination of the data, (B) transformations, if necessary, (C) initial identification of the dependence orders and degree of differencing, (D) parameter estimation, (E) residual diagnostics and model choice.
- (b) Use the CCF and lag plots between ∇S_t and ∇L_t to argue that a regression of ∇S_t on ∇L_{t-3} is reasonable. [Note that in `lag2.plot()`, the first named series is the one that gets lagged.]
- (c) Fit the regression model $\nabla S_t = \beta_0 + \beta_1 \nabla L_{t-3} + x_t$, where x_t is an ARMA process (explain how you decided on your model for x_t). Discuss your results. [See Example 3.45 for help on coding this problem.]

3.36 One of the remarkable technological developments in the computer industry has been the ability to store information densely on a hard drive. In addition, the cost of storage has steadily declined causing problems of *too much data* as opposed to *big data*. The data set for this assignment is `cpg`, which consists of the median annual retail price per GB of hard drives, say c_t , taken from a sample of manufacturers from 1980 to 2008.

- (a) Plot c_t and describe what you see.
- (b) Argue that the curve c_t versus t behaves like $c_t \approx \alpha e^{\beta t}$ by fitting a linear regression of $\log c_t$ on t and then plotting the fitted line to compare it to the logged data. Comment.

- (c) Inspect the residuals of the linear regression fit and comment.
- (d) Fit the regression again, but now using the fact that the errors are autocorrelated. Comment.

3.37 Redo [Problem 2.2](#) without assuming the error term is white noise.

Section 3.9

3.38 Consider the ARIMA model

$$x_t = w_t + \theta w_{t-2}.$$

- (a) Identify the model using the notation $\text{ARIMA}(p, d, q) \times (P, D, Q)_s$.
- (b) Show that the series is invertible for $|\theta| < 1$, and find the coefficients in the representation

$$w_t = \sum_{k=0}^{\infty} \pi_k x_{t-k}.$$

- (c) Develop equations for the m -step ahead forecast, \tilde{x}_{n+m} , and its variance based on the infinite past, x_n, x_{n-1}, \dots

3.39 Plot the ACF of the seasonal ARIMA(0, 1) \times (1, 0)₁₂ model with $\Phi = .8$ and $\theta = .5$.

3.40 Fit a seasonal ARIMA model of your choice to the chicken price data in [chicken](#). Use the estimated model to forecast the next 12 months.

3.41 Fit a seasonal ARIMA model of your choice to the unemployment data in [unemp](#). Use the estimated model to forecast the next 12 months.

3.42 Fit a seasonal ARIMA model of your choice to the unemployment data in [UnempRate](#). Use the estimated model to forecast the next 12 months.

3.43 Fit a seasonal ARIMA model of your choice to the U.S. Live Birth Series ([birth](#)). Use the estimated model to forecast the next 12 months.

3.44 Fit an appropriate seasonal ARIMA model to the log-transformed Johnson and Johnson earnings series ([jj](#)) of [Example 1.1](#). Use the estimated model to forecast the next 4 quarters.

The following problems require supplemental material given in [Appendix B](#).

3.45 Suppose $x_t = \sum_{j=1}^p \phi_j x_{t-j} + w_t$, where $\phi_p \neq 0$ and w_t is white noise such that w_t is uncorrelated with $\{x_k; k < t\}$. Use the Projection Theorem, [Theorem B.1](#), to show that, for $n > p$, the BLP of x_{n+1} on $\overline{\text{sp}}\{x_k, k \leq n\}$ is

$$\hat{x}_{n+1} = \sum_{j=1}^p \phi_j x_{n+1-j}.$$

3.46 Use the Projection Theorem to derive the Innovations Algorithm, **Property 3.6**, equations (3.77)-(3.79). Then, use **Theorem B.2** to derive the m -step-ahead forecast results given in (3.80) and (3.81).

3.47 Consider the series $x_t = w_t - w_{t-1}$, where w_t is a white noise process with mean zero and variance σ_w^2 . Suppose we consider the problem of predicting x_{n+1} , based on only x_1, \dots, x_n . Use the Projection Theorem to answer the questions below.

(a) Show the best linear predictor is

$$x_{n+1}^n = -\frac{1}{n+1} \sum_{k=1}^n k x_k.$$

(b) Prove the mean square error is

$$\mathbb{E}(x_{n+1} - x_{n+1}^n)^2 = \frac{n+2}{n+1} \sigma_w^2.$$

3.48 Use **Theorem B.2** and **Theorem B.3** to verify (3.117).

3.49 Prove **Theorem B.2**.

3.50 Prove **Property 3.2**.

Chapter 4

Spectral Analysis and Filtering

In this chapter, we focus on the *frequency domain* approach to time series analysis. We argue that the concept of regularity of a series can best be expressed in terms of periodic variations of the underlying phenomenon that produced the series. Many of the examples in [Section 1.1](#) are time series that are driven by periodic components. For example, the speech recording in [Figure 1.3](#) contains a complicated mixture of frequencies related to the opening and closing of the glottis. The monthly SOI displayed in [Figure 1.5](#) contains two periodicities, a seasonal periodic component of 12 months and an El Niño component of about three to seven years. Of fundamental interest is the return period of the El Niño phenomenon, which can have profound effects on local climate.

An important part of analyzing data in the frequency domain, as well as the time domain, is the investigation and exploitation of the properties of the time-invariant linear filter. This special linear transformation is used similarly to linear regression in conventional statistics, and we use many of the same terms in the time series context.

We also introduce coherency as a tool for relating the common periodic behavior of two series. Coherency is a frequency based measure of the correlation between two series at a given frequency, and we show later that it measures the performance of the best linear filter relating the two series.

Many frequency scales will often coexist, depending on the nature of the problem. For example, in the Johnson & Johnson data set in [Figure 1.1](#), the predominant frequency of oscillation is one cycle per year (4 quarters), or $\omega = .25$ cycles per observation. The predominant frequency in the SOI and fish populations series in [Figure 1.5](#) is also one cycle per year, but this corresponds to 1 cycle every 12 months, or $\omega = .083$ cycles per observation. Throughout the text, we measure frequency, ω , at cycles per time point rather than the alternative $\lambda = 2\pi\omega$ that would give radians per point. Of descriptive interest is the *period* of a time series, defined as the number of points in a cycle, i.e., $1/\omega$. Hence, the predominant period of the Johnson & Johnson series is $1/.25$ or 4 quarters per cycle, whereas the predominant period of the SOI series is 12 months per cycle.

4.1 Cyclical Behavior and Periodicity

We have already encountered the notion of periodicity in numerous examples in Chapters 1, 2 and 3. The general notion of periodicity can be made more precise by introducing some terminology. In order to define the rate at which a series oscillates, we first define a *cycle* as one complete period of a sine or cosine function defined over a unit time interval. As in (1.5), we consider the periodic process

$$x_t = A \cos(2\pi\omega t + \phi) \quad (4.1)$$

for $t = 0, \pm 1, \pm 2, \dots$, where ω is a *frequency* index, defined in cycles per unit time with A determining the height or *amplitude* of the function and ϕ , called the *phase*, determining the start point of the cosine function. We can introduce random variation in this time series by allowing the amplitude and phase to vary randomly.

As discussed in Example 2.10, for purposes of data analysis, it is easier to use a trigonometric identity^{4.1} and write (4.1) as

$$x_t = U_1 \cos(2\pi\omega t) + U_2 \sin(2\pi\omega t), \quad (4.2)$$

where $U_1 = A \cos \phi$ and $U_2 = -A \sin \phi$ are often taken to be normally distributed random variables. In this case, the amplitude is $A = \sqrt{U_1^2 + U_2^2}$ and the phase is $\phi = \tan^{-1}(-U_2/U_1)$. From these facts we can show that if, and only if, in (4.1), A and ϕ are independent random variables, where A^2 is chi-squared with 2 degrees of freedom, and ϕ is uniformly distributed on $(-\pi, \pi)$, then U_1 and U_2 are independent, standard normal random variables (see Problem 4.3).

If we assume that U_1 and U_2 are uncorrelated random variables with mean 0 and variance σ^2 , then x_t in (4.2) is stationary with mean $E(x_t) = 0$ and, writing $c_t = \cos(2\pi\omega t)$ and $s_t = \sin(2\pi\omega t)$, autocovariance function

$$\begin{aligned} \gamma_x(h) &= \text{cov}(x_{t+h}, x_t) = \text{cov}(U_1 c_{t+h} + U_2 s_{t+h}, U_1 c_t + U_2 s_t) \\ &= \text{cov}(U_1 c_{t+h}, U_1 c_t) + \text{cov}(U_1 c_{t+h}, U_2 s_t) \\ &\quad + \text{cov}(U_2 s_{t+h}, U_1 c_t) + \text{cov}(U_2 s_{t+h}, U_2 s_t) \\ &= \sigma^2 c_{t+h} c_t + 0 + 0 + \sigma^2 s_{t+h} s_t = \sigma^2 \cos(2\pi\omega h), \end{aligned} \quad (4.3)$$

using Footnote 4.1 and noting that $\text{cov}(U_1, U_2) = 0$. From (4.3), we see that

$$\text{var}(x_t) = \gamma_x(0) = \sigma^2.$$

Thus, if we observe $U_1 = a$ and $U_2 = b$, an estimate of σ^2 is the sample variance of these two observations, which in this case is simply $S^2 = \frac{a^2+b^2}{2-1} = a^2 + b^2$.

The random process in (4.2) is function of its frequency, ω . For $\omega = 1$, the series makes one cycle per time unit; for $\omega = .50$, the series makes a cycle every two time units; for $\omega = .25$, every four units, and so on. In general, for data that occur at discrete time points, we will need at least two points to determine a cycle, so the

^{4.1} $\cos(\alpha \pm \beta) = \cos(\alpha) \cos(\beta) \mp \sin(\alpha) \sin(\beta)$.

highest frequency of interest is .5 cycles per point. This frequency is called the *folding frequency* and defines the highest frequency that can be seen in discrete sampling. Higher frequencies sampled this way will appear at lower frequencies, called *aliases*; an example is the way a camera samples a rotating wheel on a moving automobile in a movie, in which the wheel appears to be rotating at a different rate, and sometimes backwards (the *wagon wheel effect*). For example, most movies are recorded at 24 frames per second (or 24 Hertz). If the camera is filming a wheel that is rotating at 24 Hertz, the wheel will appear to stand still.

Consider a generalization of (4.2) that allows mixtures of periodic series with multiple frequencies and amplitudes,

$$x_t = \sum_{k=1}^q [U_{k1} \cos(2\pi\omega_k t) + U_{k2} \sin(2\pi\omega_k t)], \quad (4.4)$$

where U_{k1}, U_{k2} , for $k = 1, 2, \dots, q$, are uncorrelated zero-mean random variables with variances σ_k^2 , and the ω_k are distinct frequencies. Notice that (4.4) exhibits the process as a sum of uncorrelated components, with variance σ_k^2 for frequency ω_k . As in (4.3), it is easy to show (Problem 4.4) that the autocovariance function of the process is

$$\gamma_x(h) = \sum_{k=1}^q \sigma_k^2 \cos(2\pi\omega_k h), \quad (4.5)$$

and we note the autocovariance function is the sum of periodic components with weights proportional to the variances σ_k^2 . Hence, x_t is a mean-zero stationary processes with variance

$$\gamma_x(0) = \text{var}(x_t) = \sum_{k=1}^q \sigma_k^2, \quad (4.6)$$

exhibiting the overall variance as a sum of variances of each of the component parts.

As in the simple case, if we observe $U_{k1} = a_k$ and $U_{k2} = b_k$ for $k = 1, \dots, q$, then an estimate of the k th variance component, σ_k^2 , of $\text{var}(x_t)$, would be the sample variance $S_k^2 = a_k^2 + b_k^2$. In addition, an estimate of the total variance of x_t , namely, $\gamma_x(0)$ would be the sum of the sample variances,

$$\hat{\gamma}_x(0) = \hat{\text{var}}(x_t) = \sum_{k=1}^q (a_k^2 + b_k^2). \quad (4.7)$$

Hold on to this idea because we will use it in Example 4.2.

Example 4.1 A Periodic Series

Figure 4.1 shows an example of the mixture (4.4) with $q = 3$ constructed in the following way. First, for $t = 1, \dots, 100$, we generated three series

$$x_{t1} = 2 \cos(2\pi t 6/100) + 3 \sin(2\pi t 6/100)$$

$$x_{t2} = 4 \cos(2\pi t 10/100) + 5 \sin(2\pi t 10/100)$$

$$x_{t3} = 6 \cos(2\pi t 40/100) + 7 \sin(2\pi t 40/100)$$

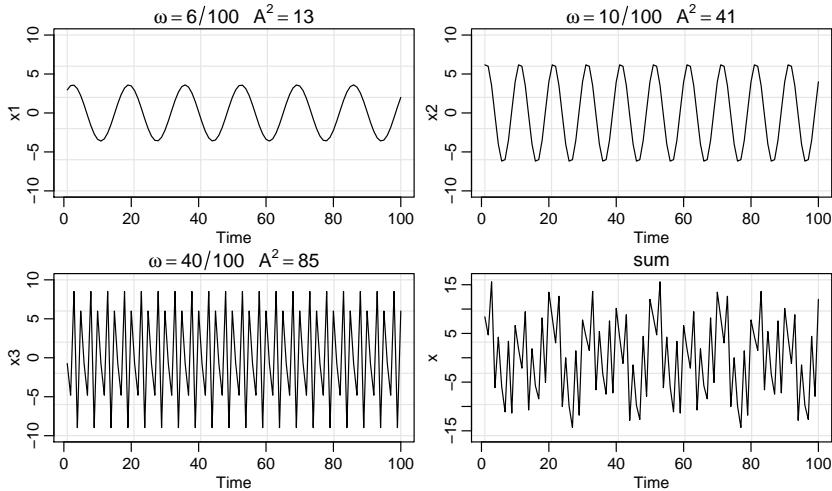


Fig. 4.1. Periodic components and their sum as described in Example 4.1.

These three series are displayed in Figure 4.1 along with the corresponding frequencies and squared amplitudes. For example, the squared amplitude of x_{t1} is $A^2 = 2^2 + 3^2 = 13$. Hence, the maximum and minimum values that x_{t1} will attain are $\pm\sqrt{13} = \pm 3.61$.

Finally, we constructed

$$x_t = x_{t1} + x_{t2} + x_{t3}$$

and this series is also displayed in Figure 4.1. We note that x_t appears to behave as some of the periodic series we saw in Chapters 1 and 2. The systematic sorting out of the essential frequency components in a time series, including their relative contributions, constitutes one of the main objectives of spectral analysis. The R code to reproduce Figure 4.1 is

```

x1 = 2*cos(2*pi*1:100*6/100) + 3*sin(2*pi*1:100*6/100)
x2 = 4*cos(2*pi*1:100*10/100) + 5*sin(2*pi*1:100*10/100)
x3 = 6*cos(2*pi*1:100*40/100) + 7*sin(2*pi*1:100*40/100)
x = x1 + x2 + x3
par(mfrow=c(2,2))
plot.ts(x1, ylim=c(-10,10), main=expression(omega==6/100~~~A^2==13))
plot.ts(x2, ylim=c(-10,10), main=expression(omega==10/100~~~A^2==41))
plot.ts(x3, ylim=c(-10,10), main=expression(omega==40/100~~~A^2==85))
plot.ts(x, ylim=c(-16,16), main="sum")

```

The model given in (4.4) along with the corresponding autocovariance function given in (4.5) are population constructs. Although, in (4.7), we hinted as to how we would estimate the variance components, we now discuss the practical aspects of how, given data x_1, \dots, x_n , to actually estimate the variance components σ_k^2 in (4.6).

Example 4.2 Estimation and the Periodogram

For any time series sample x_1, \dots, x_n , where n is odd, we may write, *exactly*

$$x_t = a_0 + \sum_{j=1}^{(n-1)/2} [a_j \cos(2\pi t j/n) + b_j \sin(2\pi t j/n)], \quad (4.8)$$

for $t = 1, \dots, n$ and suitably chosen coefficients. If n is even, the representation (4.8) can be modified by summing to $(n/2 - 1)$ and adding an additional component given by $a_{n/2} \cos(2\pi t \frac{1}{2}) = a_{n/2}(-1)^t$. The crucial point here is that (4.8) is exact for any sample. Hence (4.4) may be thought of as an approximation to (4.8), the idea being that many of the coefficients in (4.8) may be close to zero.

Using the regression results from Chapter 2, the coefficients a_j and b_j are of the form $\sum_{t=1}^n x_t z_{tj} / \sum_{t=1}^n z_{tj}^2$, where z_{tj} is either $\cos(2\pi t j/n)$ or $\sin(2\pi t j/n)$. Using Problem 4.1, $\sum_{t=1}^n z_{tj}^2 = n/2$ when $j/n \neq 0, 1/2$, so the regression coefficients in (4.8) can be written as $(a_0 = \bar{x})$,

$$a_j = \frac{2}{n} \sum_{t=1}^n x_t \cos(2\pi t j/n) \quad \text{and} \quad b_j = \frac{2}{n} \sum_{t=1}^n x_t \sin(2\pi t j/n).$$

We then define the *scaled periodogram* to be

$$P(j/n) = a_j^2 + b_j^2, \quad (4.9)$$

and it is of interest because it indicates which frequency components in (4.8) are large in magnitude and which components are small. *The scaled periodogram is simply the sample variance at each frequency component and consequently is an estimate of σ_j^2 corresponding to the sinusoid oscillating at a frequency of $\omega_j = j/n$.* These particular frequencies are called the *Fourier or fundamental frequencies*. Large values of $P(j/n)$ indicate which frequencies $\omega_j = j/n$ are predominant in the series, whereas small values of $P(j/n)$ may be associated with noise. The periodogram was introduced in Schuster (1898) and used in Schuster (1906) for studying the periodicities in the sunspot series (shown in Figure 4.22).

Fortunately, it is not necessary to run a large regression to obtain the values of a_j and b_j because they can be computed quickly if n is a highly composite integer. Although we will discuss it in more detail in Section 4.3, the *discrete Fourier transform (DFT)* is a complex-valued weighted average of the data given by^{4.2}

$$\begin{aligned} d(j/n) &= n^{-1/2} \sum_{t=1}^n x_t \exp(-2\pi i t j/n) \\ &= n^{-1/2} \left(\sum_{t=1}^n x_t \cos(2\pi t j/n) - i \sum_{t=1}^n x_t \sin(2\pi t j/n) \right), \end{aligned} \quad (4.10)$$

^{4.2} Euler's formula: $e^{i\alpha} = \cos(\alpha) + i \sin(\alpha)$. Consequently, $\cos(\alpha) = \frac{e^{i\alpha} + e^{-i\alpha}}{2}$, and $\sin(\alpha) = \frac{e^{i\alpha} - e^{-i\alpha}}{2i}$. Also, $\frac{1}{i} = -i$ because $-i \times i = 1$. If $z = a+ib$ is complex, then $|z|^2 = z z^* = (a+ib)(a-ib) = a^2+b^2$; the * denotes conjugation.

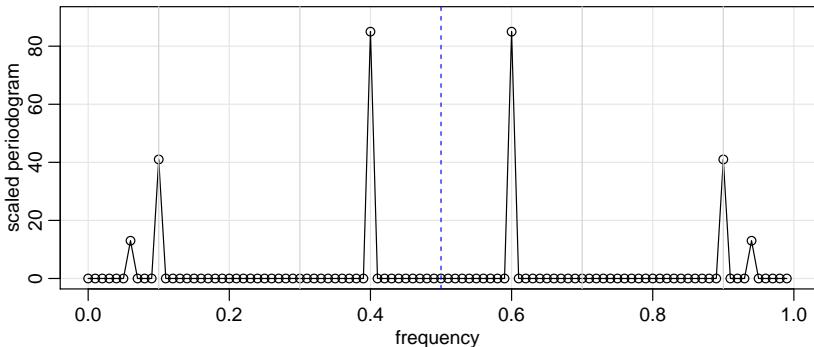


Fig. 4.2. The scaled periodogram (4.12) of the data generated in Example 4.1.

for $j = 0, 1, \dots, n - 1$, where the frequencies j/n are the Fourier or fundamental frequencies. Because of a large number of redundancies in the calculation, (4.10) may be computed quickly using the *fast Fourier transform (FFT)*. Note that

$$|d(j/n)|^2 = \frac{1}{n} \left(\sum_{t=1}^n x_t \cos(2\pi t j/n) \right)^2 + \frac{1}{n} \left(\sum_{t=1}^n x_t \sin(2\pi t j/n) \right)^2 \quad (4.11)$$

and it is this quantity that is called the *periodogram*. We may calculate the scaled periodogram, (4.9), using the periodogram as

$$P(j/n) = \frac{4}{n} |d(j/n)|^2. \quad (4.12)$$

The scaled periodogram of the data, x_t , simulated in Example 4.1 is shown in Figure 4.2, and it clearly identifies the three components x_{t1} , x_{t2} , and x_{t3} of x_t . Note that

$$P(j/n) = P(1 - j/n), \quad j = 0, 1, \dots, n - 1,$$

so there is a mirroring effect at the folding frequency of 1/2; consequently, the periodogram is typically not plotted for frequencies higher than the folding frequency. In addition, note that the heights of the scaled periodogram shown in the figure are

$$P(\frac{6}{100}) = P(\frac{94}{100}) = 13, \quad P(\frac{10}{100}) = P(\frac{90}{100}) = 41, \quad P(\frac{40}{100}) = P(\frac{60}{100}) = 85,$$

and $P(j/n) = 0$ otherwise. These are exactly the values of the squared amplitudes of the components generated in Example 4.1.

Assuming the simulated data, x , were retained from the previous example, the R code to reproduce Figure 4.2 is

```
P = Mod(2*fft(x)/100)^2; Fr = 0:99/100
plot(Fr, P, type="o", xlab="frequency", ylab="scaled periodogram")
```

Different packages scale the FFT differently, so it is a good idea to consult the documentation. R computes it without the factor $n^{-1/2}$ and with an additional factor of $e^{2\pi i \omega_j}$ that can be ignored because we will be interested in the squared modulus.

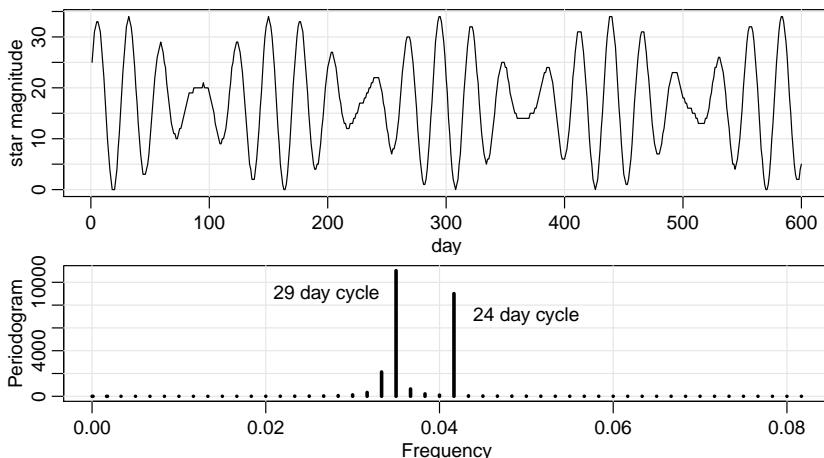


Fig. 4.3. Star magnitudes and part of the corresponding periodogram.

If we consider the data x_t in [Example 4.1](#) as a color (waveform) made up of primary colors x_{t1}, x_{t2}, x_{t3} at various strengths (amplitudes), then we might consider the periodogram as a prism that decomposes the color x_t into its primary colors (spectrum). Hence the term *spectral analysis*. The following is an example using actual data.

Example 4.3 Star Magnitude

The data in [Figure 4.3](#) are the magnitude of a star taken at midnight for 600 consecutive days. The data are taken from the classic text, *The Calculus of Observations, a Treatise on Numerical Mathematics*, by E.T. Whittaker and G. Robinson, (1923, Blackie & Son, Ltd.).

The periodogram for frequencies less than .08 is also displayed in the figure; the periodogram ordinates for frequencies higher than .08 are essentially zero. Note that the $29 (\approx 1/.035)$ day cycle and the $24 (\approx 1/.041)$ day cycle are the most prominent periodic components of the data.

We can interpret this result as we are observing an *amplitude modulated* signal. For example, suppose we are observing signal-plus-noise, $x_t = s_t + v_t$, where $s_t = \cos(2\pi\omega t) \cos(2\pi\delta t)$, and δ is very small. In this case, the process will oscillate at frequency ω , but the amplitude will be modulated by $\cos(2\pi\delta t)$. Since $2 \cos(\alpha) \cos(\delta) = \cos(\alpha + \delta) + \cos(\alpha - \delta)$, the periodogram of data generated as x_t will have two peaks close to each other at $\alpha \pm \delta$. Try this on your own:

```
t = 1:200
plot.ts(x <- 2*cos(2*pi*.2*t)*cos(2*pi*.01*t))    # not shown
lines(cos(2*pi*.19*t)+cos(2*pi*.21*t), col=2)      # the same
Px = Mod(fft(x))^2; plot(0:199/200, Px, type='o') # the periodogram
```

The R code to reproduce [Figure 4.3](#) is

```
n = length(star)
par(mfrow=c(2,1), mar=c(3,3,1,1), mgp=c(1.6,.6,0))
plot(star, ylab="star magnitude", xlab="day")
```

```

Per = Mod(fft(star-mean(star)))^2/n
Freq = (1:n - 1)/n
plot(Freq[1:50], Per[1:50], type='h', lwd=3, ylab="Periodogram",
      xlab="Frequency")
u = which.max(Per[1:50])           # 22   freq=21/600=.035 cycles/day
uu = which.max(Per[1:50][-u])      # 25   freq=25/600=.041 cycles/day
1/Freq[22]; 1/Freq[26]            # period = days/cycle
text(.05, 7000, "24 day cycle"); text(.027, 9000, "29 day cycle")
### another way to find the two peaks is to order on Per
y = cbind(1:50, Freq[1:50], Per[1:50]); y[order(y[,3]),]

```

4.2 The Spectral Density

In this section, we define the fundamental frequency domain tool, the spectral density. In addition, we discuss the spectral representations for stationary processes. Just as the Wold decomposition (Theorem B.5) theoretically justified the use of regression for analyzing time series, the spectral representation theorems supply the theoretical justifications for decomposing stationary time series into periodic components appearing in proportion to their underlying variances. This material is enhanced by the results presented in Appendix C.

Example 4.4 A Periodic Stationary Process

Consider a periodic stationary random process given by (4.2), with a fixed frequency ω_0 , say,

$$x_t = U_1 \cos(2\pi\omega_0 t) + U_2 \sin(2\pi\omega_0 t), \quad (4.13)$$

where U_1 and U_2 are uncorrelated zero-mean random variables with equal variance σ^2 . The number of time periods needed for the above series to complete one cycle is exactly $1/\omega_0$, and the process makes exactly ω_0 cycles per point for $t = 0, \pm 1, \pm 2, \dots$. Recalling (4.3) and using Footnote 4.2, we have

$$\begin{aligned} \gamma(h) &= \sigma^2 \cos(2\pi\omega_0 h) = \frac{\sigma^2}{2} e^{-2\pi i \omega_0 h} + \frac{\sigma^2}{2} e^{2\pi i \omega_0 h} \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF(\omega) \end{aligned}$$

using Riemann–Stieltjes integration (see Section C.4.1), where $F(\omega)$ is the function defined by

$$F(\omega) = \begin{cases} 0 & \omega < -\omega_0, \\ \sigma^2/2 & -\omega_0 \leq \omega < \omega_0, \\ \sigma^2 & \omega \geq \omega_0. \end{cases}$$

The function $F(\omega)$ behaves like a cumulative distribution function for a discrete random variable, except that $F(\infty) = \sigma^2 = \text{var}(x_t)$ instead of one. In fact, $F(\omega)$ is a cumulative distribution function, not of probabilities, but rather of variances, with $F(\infty)$ being the total variance of the process x_t . Hence, we term $F(\omega)$ the *spectral distribution function*. This example is continued in Example 4.9.

A representation such as the one given in [Example 4.4](#) always exists for a stationary process. For details, see [Theorem C.1](#) and its proof; Riemann–Stieltjes integration is described in [Section C.4.1](#).

Property 4.1 Spectral Representation of an Autocovariance Function

If $\{x_t\}$ is stationary with autocovariance $\gamma(h) = \text{cov}(x_{t+h}, x_t)$, then there exists a unique monotonically increasing function $F(\omega)$, called the spectral distribution function, with $F(-\infty) = F(-1/2) = 0$, and $F(\infty) = F(1/2) = \gamma(0)$ such that

$$\gamma(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF(\omega). \quad (4.14)$$

An important situation we use repeatedly is the case when the autocovariance function is absolutely summable, in which case the spectral distribution function is absolutely continuous with $dF(\omega) = f(\omega) d\omega$, and the representation (4.14) becomes the motivation for the property given below.

Property 4.2 The Spectral Density

If the autocovariance function, $\gamma(h)$, of a stationary process satisfies

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty, \quad (4.15)$$

then it has the representation

$$\gamma(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} f(\omega) d\omega \quad h = 0, \pm 1, \pm 2, \dots \quad (4.16)$$

as the inverse transform of the spectral density,

$$f(\omega) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega h} \quad -1/2 \leq \omega \leq 1/2. \quad (4.17)$$

This spectral density is the analogue of the probability density function; the fact that $\gamma(h)$ is non-negative definite ensures

$$f(\omega) \geq 0$$

for all ω . It follows immediately from (4.17) that

$$f(\omega) = f(-\omega)$$

verifying the spectral density is an even function. Because of the evenness, we will typically only plot $f(\omega)$ for $0 \leq \omega \leq 1/2$. In addition, putting $h = 0$ in (4.16) yields

$$\gamma(0) = \text{var}(x_t) = \int_{-\frac{1}{2}}^{\frac{1}{2}} f(\omega) d\omega,$$

which expresses the total variance as the integrated spectral density over all of the frequencies. We show later on, that a linear filter can isolate the variance in certain frequency intervals or *bands*.

It should now be clear that the autocovariance and the spectral distribution functions contain the same information. That information, however, is expressed in different ways. The autocovariance function expresses information in terms of lags, whereas the spectral distribution expresses the same information in terms of cycles. Some problems are easier to work with when considering lagged information and we would tend to handle those problems in the time domain. Nevertheless, other problems are easier to work with when considering periodic information and we would tend to handle those problems in the spectral domain.

We note that the autocovariance function, $\gamma(h)$, in (4.16) and the spectral density, $f(\omega)$, in (4.17) are Fourier transform pairs. In particular, this means that if $f(\omega)$ and $g(\omega)$ are two spectral densities for which

$$\gamma_f(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} f(\omega) e^{2\pi i \omega h} d\omega = \int_{-\frac{1}{2}}^{\frac{1}{2}} g(\omega) e^{2\pi i \omega h} d\omega = \gamma_g(h) \quad (4.18)$$

for all $h = 0, \pm 1, \pm 2, \dots$, then

$$f(\omega) = g(\omega). \quad (4.19)$$

Finally, the absolute summability condition, (4.15), is not satisfied by (4.5), the example that we have used to introduce the idea of a spectral representation. The condition, however, is satisfied for ARMA models.

It is illuminating to examine the spectral density for the series that we have looked at in earlier discussions.

Example 4.5 White Noise Series

As a simple example, consider the theoretical power spectrum of a sequence of uncorrelated random variables, w_t , with variance σ_w^2 . A simulated set of data is displayed in the top of Figure 1.8. Because the autocovariance function was computed in Example 1.16 as $\gamma_w(h) = \sigma_w^2$ for $h = 0$, and zero, otherwise, it follows from (4.17), that

$$f_w(\omega) = \sigma_w^2$$

for $-1/2 \leq \omega \leq 1/2$. Hence the process contains equal power at all frequencies. This property is seen in the realization, which seems to contain all different frequencies in a roughly equal mix. In fact, the name white noise comes from the analogy to white light, which contains all frequencies in the color spectrum at the same level of intensity. The top of Figure 4.4 shows a plot of the white noise spectrum for $\sigma_w^2 = 1$. The R code to reproduce the figure is given at the end of Example 4.7.

Since the linear process is an essential tool, it is worthwhile investigating the spectrum of such a process. In general, a linear filter uses a set of specified coefficients, say a_j , for $j = 0, \pm 1, \pm 2, \dots$, to transform an input series, x_t , producing an output series, y_t , of the form

$$y_t = \sum_{j=-\infty}^{\infty} a_j x_{t-j}, \quad \sum_{j=-\infty}^{\infty} |a_j| < \infty. \quad (4.20)$$

The form (4.20) is also called a *convolution* in some statistical contexts. The coefficients are collectively called the *impulse response function*, and the Fourier transform

$$A(\omega) = \sum_{j=-\infty}^{\infty} a_j e^{-2\pi i \omega j}, \quad (4.21)$$

is called the *frequency response function*. If, in (4.20), x_t has spectral density $f_x(\omega)$, we have the following result.

Property 4.3 Output Spectrum of a Filtered Stationary Series

For the process in (4.20), if x_t has spectrum $f_x(\omega)$, then the spectrum of the filtered output, y_t , say $f_y(\omega)$, is related to the spectrum of the input x_t by

$$f_y(\omega) = |A(\omega)|^2 f_x(\omega), \quad (4.22)$$

where the frequency response function $A(\omega)$ is defined in (4.21).

Proof: The autocovariance function of the filtered output y_t in (4.20) is

$$\begin{aligned} \gamma_y(h) &= \text{cov}(x_{t+h}, x_t) \\ &= \text{cov}\left(\sum_r a_r x_{t+h-r}, \sum_s a_s x_{t-s}\right) \\ &= \sum_r \sum_s a_r \gamma_x(h - r + s) a_s \\ &\stackrel{(1)}{=} \sum_r \sum_s a_r \left[\int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega(h-r+s)} f_x(\omega) d\omega \right] a_s \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\sum_r a_r e^{-2\pi i \omega r} \right) \left(\sum_s a_s e^{2\pi i \omega s} \right) e^{2\pi i \omega h} f_x(\omega) d\omega \\ &\stackrel{(2)}{=} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} \underbrace{|A(\omega)|^2 f_x(\omega)}_{f_y(\omega)} d\omega, \end{aligned}$$

where we have, (1) replaced $\gamma_x(\cdot)$ by its representation (4.16), and (2) substituted $A(\omega)$ from (4.21). The result holds by exploiting the uniqueness of the Fourier transform. \square

The use of Property 4.3 is explored further in Section 4.7. If x_t is ARMA, its spectral density can be obtained explicitly using the fact that it is a linear process, i.e., $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$, where $\sum_{j=0}^{\infty} |\psi_j| < \infty$. The following property is a direct consequence of Property 4.3, by using the additional facts that the spectral density of white noise is $f_w(\omega) = \sigma_w^2$, and by Property 3.1, $\psi(z) = \theta(z)/\phi(z)$.

Property 4.4 The Spectral Density of ARMA

If x_t is ARMA(p, q), $\phi(B)x_t = \theta(B)w_t$, its spectral density is given by

$$f_x(\omega) = \sigma_w^2 \frac{|\theta(e^{-2\pi i \omega})|^2}{|\phi(e^{-2\pi i \omega})|^2} \quad (4.23)$$

where $\phi(z) = 1 - \sum_{k=1}^p \phi_k z^k$ and $\theta(z) = 1 + \sum_{k=1}^q \theta_k z^k$.

Example 4.6 Moving Average

As an example of a series that does not have an equal mix of frequencies, we consider a moving average model. Specifically, consider the MA(1) model given by

$$x_t = w_t + .5w_{t-1}.$$

A sample realization is shown in the top of [Figure 3.2](#) and we note that the series has less of the higher or faster frequencies. The spectral density will verify this observation.

The autocovariance function is displayed in [Example 3.5](#), and for this particular example, we have

$$\gamma(0) = (1 + .5^2)\sigma_w^2 = 1.25\sigma_w^2; \quad \gamma(\pm 1) = .5\sigma_w^2; \quad \gamma(\pm h) = 0 \text{ for } h > 1.$$

Substituting this directly into the definition given in [\(4.17\)](#), we have

$$\begin{aligned} f(\omega) &= \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega h} = \sigma_w^2 \left[1.25 + .5 \left(e^{-2\pi i \omega} + e^{2\pi i \omega} \right) \right] \\ &= \sigma_w^2 [1.25 + \cos(2\pi\omega)]. \end{aligned} \quad (4.24)$$

We can also compute the spectral density using [Property 4.4](#), which states that for an MA, $f(\omega) = \sigma_w^2 |\theta(e^{-2\pi i \omega})|^2$. Because $\theta(z) = 1 + .5z$, we have

$$\begin{aligned} |\theta(e^{-2\pi i \omega})|^2 &= |1 + .5e^{-2\pi i \omega}|^2 = (1 + .5e^{-2\pi i \omega})(1 + .5e^{2\pi i \omega}) \\ &= 1.25 + .5 \left(e^{-2\pi i \omega} + e^{2\pi i \omega} \right) \end{aligned}$$

which leads to agreement with [\(4.24\)](#).

Plotting the spectrum for $\sigma_w^2 = 1$, as in the middle of [Figure 4.4](#), shows the lower or slower frequencies have greater power than the higher or faster frequencies.

Example 4.7 A Second-Order Autoregressive Series

We now consider the spectrum of an AR(2) series of the form

$$x_t - \phi_1 x_{t-1} - \phi_2 x_{t-2} = w_t,$$

for the special case $\phi_1 = 1$ and $\phi_2 = -.9$. [Figure 1.9](#) shows a sample realization of such a process for $\sigma_w = 1$. We note the data exhibit a strong periodic component that makes a cycle about every six points.

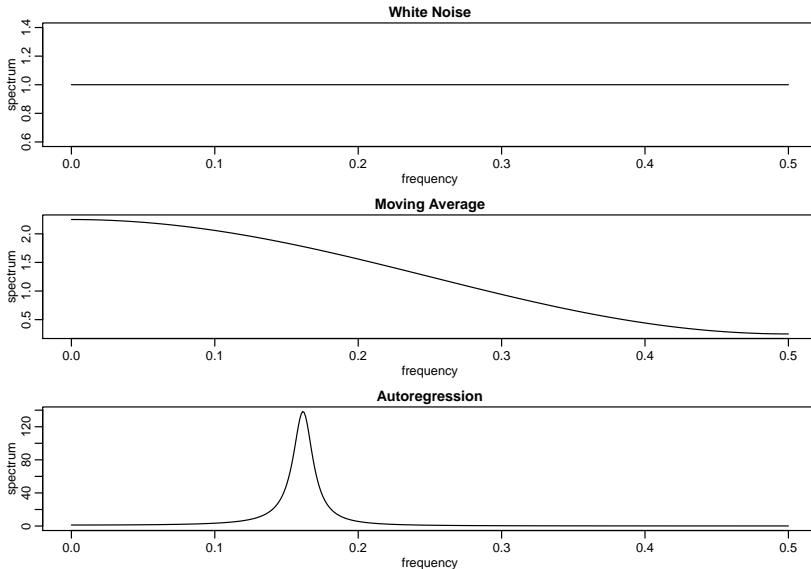


Fig. 4.4. Theoretical spectra of white noise (top), a first-order moving average (middle), and a second-order autoregressive process (bottom).

To use [Property 4.4](#), note that $\theta(z) = 1$, $\phi(z) = 1 - z + .9z^2$ and

$$\begin{aligned} |\phi(e^{-2\pi i\omega})|^2 &= (1 - e^{-2\pi i\omega} + .9e^{-4\pi i\omega})(1 - e^{2\pi i\omega} + .9e^{4\pi i\omega}) \\ &= 2.81 - 1.9(e^{2\pi i\omega} + e^{-2\pi i\omega}) + .9(e^{4\pi i\omega} + e^{-4\pi i\omega}) \\ &= 2.81 - 3.8 \cos(2\pi\omega) + 1.8 \cos(4\pi\omega). \end{aligned}$$

Using this result in [\(4.23\)](#), we have that the spectral density of x_t is

$$f_x(\omega) = \frac{\sigma_w^2}{2.81 - 3.8 \cos(2\pi\omega) + 1.8 \cos(4\pi\omega)}.$$

Setting $\sigma_w = 1$, the bottom of [Figure 4.4](#) displays $f_x(\omega)$ and shows a strong power component at about $\omega = .16$ cycles per point or a period between six and seven cycles per point and very little power at other frequencies. In this case, modifying the white noise series by applying the second-order AR operator has concentrated the power or variance of the resulting series in a very narrow frequency band.

The spectral density can also be obtained from first principles, without having to use [Property 4.4](#). Because $w_t = x_t - x_{t-1} + .9x_{t-2}$ in this example, we have

$$\begin{aligned} \gamma_w(h) &= \text{cov}(w_{t+h}, w_t) \\ &= \text{cov}(x_{t+h} - x_{t+h-1} + .9x_{t+h-2}, x_t - x_{t-1} + .9x_{t-2}) \\ &= 2.81\gamma_x(h) - 1.9[\gamma_x(h+1) + \gamma_x(h-1)] + .9[\gamma_x(h+2) + \gamma_x(h-2)] \end{aligned}$$

Now, substituting the spectral representation (4.16) for $\gamma_x(h)$ in the above equation yields

$$\begin{aligned}\gamma_w(h) &= \int_{-\frac{1}{2}}^{\frac{1}{2}} [2.81 - 1.9(e^{2\pi i\omega} + e^{-2\pi i\omega}) + .9(e^{4\pi i\omega} + e^{-4\pi i\omega})] e^{2\pi i\omega h} f_x(\omega) d\omega \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} [2.81 - 3.8 \cos(2\pi\omega) + 1.8 \cos(4\pi\omega)] e^{2\pi i\omega h} f_x(\omega) d\omega.\end{aligned}$$

If the spectrum of the white noise process, w_t , is $g_w(\omega)$, the uniqueness of the Fourier transform allows us to identify

$$g_w(\omega) = [2.81 - 3.8 \cos(2\pi\omega) + 1.8 \cos(4\pi\omega)] f_x(\omega).$$

But, as we have already seen, $g_w(\omega) = \sigma_w^2$, from which we deduce that

$$f_x(\omega) = \frac{\sigma_w^2}{2.81 - 3.8 \cos(2\pi\omega) + 1.8 \cos(4\pi\omega)}$$

is the spectrum of the autoregressive series.

To reproduce Figure 4.4, use `arma.spec` from `astsa`:

```
par(mfrow=c(3,1))
arma.spec(log="no", main="White Noise")
arma.spec(ma=.5, log="no", main="Moving Average")
arma.spec(ar=c(1,-.9), log="no", main="Autoregression")
```

Example 4.8 Every Explosion has a Cause (cont)

In Example 3.4, we discussed the fact that explosive models have causal counterparts. In that example, we also indicated that it was easier to show this result in general in the spectral domain. In this example, we give the details for an AR(1) model, but the techniques used here will indicate how to generalize the result.

As in Example 3.4, we suppose that $x_t = 2x_{t-1} + w_t$, where $w_t \sim \text{iid } N(0, \sigma_w^2)$. Then, the spectral density of x_t is

$$f_x(\omega) = \sigma_w^2 |1 - 2e^{-2\pi i\omega}|^{-2}. \quad (4.25)$$

But, $|1 - 2e^{-2\pi i\omega}| = |1 - 2e^{2\pi i\omega}| = |(2e^{2\pi i\omega})(\frac{1}{2}e^{-2\pi i\omega} - 1)| = 2|1 - \frac{1}{2}e^{-2\pi i\omega}|$. Thus, (4.25) can be written as

$$f_x(\omega) = \frac{1}{4}\sigma_w^2 |1 - \frac{1}{2}e^{-2\pi i\omega}|^{-2},$$

which implies that $x_t = \frac{1}{2}x_{t-1} + v_t$, with $v_t \sim \text{iid } N(0, \frac{1}{4}\sigma_w^2)$ is an equivalent form of the model.

We end this section by mentioning another spectral representation that deals with the process directly. In nontechnical terms, the result suggests that (4.4) is approximately true for any stationary time series, and this gives an additional theoretical justification for decomposing time series into harmonic components.

Example 4.9 A Periodic Stationary Process (cont)

In Example 4.4, we considered the periodic stationary process given in (4.13), namely, $x_t = U_1 \cos(2\pi\omega_0 t) + U_2 \sin(2\pi\omega_0 t)$. Using Footnote 4.2, we may write this as

$$x_t = \frac{1}{2}(U_1 + iU_2)e^{-2\pi i\omega_0 t} + \frac{1}{2}(U_1 - iU_2)e^{2\pi i\omega_0 t},$$

where we recall that U_1 and U_2 are uncorrelated, mean-zero, random variables each with variance σ^2 . If we call $Z = \frac{1}{2}(U_1 + iU_2)$, then $Z^* = \frac{1}{2}(U_1 - iU_2)$, where $*$ denotes conjugation. In this case, $E(Z) = \frac{1}{2}[E(U_1) + iE(U_2)] = 0$ and similarly $E(Z^*) = 0$. For mean-zero complex random variables, say X and Y , $\text{cov}(X, Y) = E(XY^*)$. Thus

$$\begin{aligned}\text{var}(Z) &= E(|Z|^2) = E(ZZ^*) = \frac{1}{4}E[(U_1 + iU_2)(U_1 - iU_2)] \\ &= \frac{1}{4}[E(U_1^2) + E(U_2^2)] = \frac{\sigma^2}{2}.\end{aligned}$$

Similarly, $\text{var}(Z^*) = \sigma^2/2$. Moreover, since $Z^{**} = Z$,

$$\text{cov}(Z, Z^*) = E(ZZ^{**}) = \frac{1}{4}E[(U_1 + iU_2)(U_1 + iU_2)] = \frac{1}{4}[E(U_1^2) - E(U_2^2)] = 0.$$

Hence, (4.13) may be written as

$$x_t = Z e^{-2\pi i\omega_0 t} + Z^* e^{2\pi i\omega_0 t} = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i\omega t} dZ(\omega),$$

where $Z(\omega)$ is a complex-valued random process that makes uncorrelated jumps at $-\omega_0$ and ω_0 with mean-zero and variance $\sigma^2/2$. Stochastic integration is discussed further in Section C.4.2. This notion generalizes to all stationary series in the following property (also, see Theorem C.2).

Property 4.5 Spectral Representation of a Stationary Process

If x_t is a mean-zero stationary process, with spectral distribution $F(\omega)$ as given in Property 4.1, then there exists a complex-valued stochastic process $Z(\omega)$, on the interval $\omega \in [-1/2, 1/2]$, having stationary uncorrelated non-overlapping increments, such that x_t can be written as the stochastic integral (see Section C.4.2)

$$x_t = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i\omega t} dZ(\omega),$$

where, for $-1/2 \leq \omega_1 \leq \omega_2 \leq 1/2$,

$$\text{var}\{Z(\omega_2) - Z(\omega_1)\} = F(\omega_2) - F(\omega_1).$$

4.3 Periodogram and Discrete Fourier Transform

We are now ready to tie together the periodogram, which is the sample-based concept presented in Section 4.1, with the spectral density, which is the population-based concept of Section 4.2.

Definition 4.1 Given data x_1, \dots, x_n , we define the **discrete Fourier transform (DFT)** to be

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^n x_t e^{-2\pi i \omega_j t} \quad (4.26)$$

for $j = 0, 1, \dots, n - 1$, where the frequencies $\omega_j = j/n$ are called the **Fourier or fundamental frequencies**.

If n is a highly composite integer (i.e., it has many factors), the DFT can be computed by the fast Fourier transform (FFT) introduced in Cooley and Tukey (1965). Also, different packages scale the FFT differently, so it is a good idea to consult the documentation. R computes the DFT defined in (4.26) without the factor $n^{-1/2}$, but with an additional factor of $e^{2\pi i \omega_j}$ that can be ignored because we will be interested in the squared modulus of the DFT. Sometimes it is helpful to exploit the inversion result for DFTs, which shows the linear transformation is one-to-one. For the *inverse DFT* we have,

$$x_t = n^{-1/2} \sum_{j=0}^{n-1} d(\omega_j) e^{2\pi i \omega_j t} \quad (4.27)$$

for $t = 1, \dots, n$. The following example shows how to calculate the DFT and its inverse in R for the data set $\{1, 2, 3, 4\}$; note that R writes a complex number $z = a + ib$ as `a+bi`.

```
(dft = fft(1:4)/sqrt(4))
[1] 5+0i -1+1i -1+0i -1-1i
(idft = fft(dft, inverse=TRUE)/sqrt(4))
[1] 1+0i 2+0i 3+0i 4+0i
(Re(idft)) # keep it real
[1] 1 2 3 4
```

We now define the periodogram as the squared modulus of the DFT.

Definition 4.2 Given data x_1, \dots, x_n , we define the **periodogram** to be

$$I(\omega_j) = |d(\omega_j)|^2 \quad (4.28)$$

for $j = 0, 1, 2, \dots, n - 1$.

Note that $I(0) = n\bar{x}^2$, where \bar{x} is the sample mean. Also, $\sum_{t=1}^n \exp(-2\pi i t \frac{j}{n}) = 0$ for $j \neq 0$,^{4.3} so we can write the DFT as

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^n (x_t - \bar{x}) e^{-2\pi i \omega_j t} \quad (4.29)$$

for $j \neq 0$. Thus,

^{4.3} $\overline{\sum_{t=1}^n z^t} = z \frac{1-z^n}{1-z}$ for $z \neq 1$. In this case, $z^n = e^{-2\pi i j} = 1$.

$$\begin{aligned}
I(\omega_j) &= |d(\omega_j)|^2 = n^{-1} \sum_{t=1}^n \sum_{s=1}^n (x_t - \bar{x})(x_s - \bar{x}) e^{-2\pi i \omega_j(t-s)} \\
&= n^{-1} \sum_{h=-(n-1)}^{n-1} \sum_{t=1}^{n-|h|} (x_{t+|h|} - \bar{x})(x_t - \bar{x}) e^{-2\pi i \omega_j h} \\
&= \sum_{h=-(n-1)}^{n-1} \hat{\gamma}(h) e^{-2\pi i \omega_j h}
\end{aligned} \tag{4.30}$$

for $j \neq 0$, where we have put $h = t - s$, with $\hat{\gamma}(h)$ as given in (1.36).^{4.4} In view of (4.30), the periodogram, $I(\omega_j)$, is the sample version of $f(\omega_j)$ given in (4.17). That is, we may think of the periodogram as the *sample spectral density* of x_t .

At first, (4.30) seems to be an obvious way to estimate a spectral density (4.17); i.e., simply put a hat on $\gamma(h)$ and sum as far as the sample size will allow. However, after further consideration, it turns out that this is not a very good estimator because it uses some bad estimates of $\gamma(h)$. For example, there is only one pair of observations, (x_1, x_n) for estimating $\gamma(n-1)$, and only two pairs (x_1, x_{n-1}) , and (x_2, x_n) that can be used to estimate $\gamma(n-2)$, and so on. We will discuss this problem further as we progress, but an obvious improvement over (4.30) would be something like $\hat{f}(\omega) = \sum_{|h| \leq m} \hat{\gamma}(h) e^{-2\pi i \omega h}$, where m is much smaller than n .

It is sometimes useful to work with the real and imaginary parts of the DFT individually. To this end, we define the following transforms.

Definition 4.3 Given data x_1, \dots, x_n , we define the **cosine transform**

$$d_c(\omega_j) = n^{-1/2} \sum_{t=1}^n x_t \cos(2\pi \omega_j t) \tag{4.31}$$

and the **sine transform**

$$d_s(\omega_j) = n^{-1/2} \sum_{t=1}^n x_t \sin(2\pi \omega_j t) \tag{4.32}$$

where $\omega_j = j/n$ for $j = 0, 1, \dots, n-1$.

We note that $d(\omega_j) = d_c(\omega_j) - i d_s(\omega_j)$ and hence

$$I(\omega_j) = d_c^2(\omega_j) + d_s^2(\omega_j). \tag{4.33}$$

We have also discussed the fact that spectral analysis can be thought of as an analysis of variance. The next example examines this notion.

^{4.4} Note that (4.30) can be used to obtain $\hat{\gamma}(h)$ by taking the inverse DFT of $I(\omega_j)$. This approach was used in Example 1.31 to obtain a two-dimensional ACF.

Example 4.10 Spectral ANOVA

Let x_1, \dots, x_n be a sample of size n , where for ease, n is odd. Then, recalling Example 4.2,

$$x_t = a_0 + \sum_{j=1}^m [a_j \cos(2\pi\omega_j t) + b_j \sin(2\pi\omega_j t)], \quad (4.34)$$

where $m = (n-1)/2$, is exact for $t = 1, \dots, n$. In particular, using multiple regression formulas, we have $a_0 = \bar{x}$,

$$a_j = \frac{2}{n} \sum_{t=1}^n x_t \cos(2\pi\omega_j t) = \frac{2}{\sqrt{n}} d_c(\omega_j)$$

$$b_j = \frac{2}{n} \sum_{t=1}^n x_t \sin(2\pi\omega_j t) = \frac{2}{\sqrt{n}} d_s(\omega_j).$$

Hence, we may write

$$(x_t - \bar{x}) = \frac{2}{\sqrt{n}} \sum_{j=1}^m [d_c(\omega_j) \cos(2\pi\omega_j t) + d_s(\omega_j) \sin(2\pi\omega_j t)]$$

for $t = 1, \dots, n$. Squaring both sides and summing we obtain

$$\sum_{t=1}^n (x_t - \bar{x})^2 = 2 \sum_{j=1}^m [d_c^2(\omega_j) + d_s^2(\omega_j)] = 2 \sum_{j=1}^m I(\omega_j)$$

using the results of Problem 4.1. Thus, we have partitioned the sum of squares into harmonic components represented by frequency ω_j with the periodogram, $I(\omega_j)$, being the mean square regression. This leads to the ANOVA table for n odd:

Source	df	SS	MS
ω_1	2	$2I(\omega_1)$	$I(\omega_1)$
ω_2	2	$2I(\omega_2)$	$I(\omega_2)$
\vdots	\vdots	\vdots	\vdots
ω_m	2	$2I(\omega_m)$	$I(\omega_m)$
Total	$n - 1$	$\sum_{t=1}^n (x_t - \bar{x})^2$	

The following is an R example to help explain this concept. We consider $n = 5$ observations given by $x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 2, x_5 = 1$. Note that the data complete one cycle, but not in a sinusoidal way. Thus, we should expect the $\omega_1 = 1/5$ component to be relatively large but not exhaustive, and the $\omega_2 = 2/5$ component to be small.

```
x = c(1, 2, 3, 2, 1)
c1 = cos(2*pi*1:5*1/5); s1 = sin(2*pi*1:5*1/5)
c2 = cos(2*pi*1:5*2/5); s2 = sin(2*pi*1:5*2/5)
```

```

omega1 = cbind(c1, s1); omega2 = cbind(c2, s2)
anova(lm(x~omega1+omega2)) # ANOVA Table
  Df  Sum Sq Mean Sq
omega1    2  2.74164  1.37082
omega2    2  .05836   .02918
Residuals 0  .00000
Mod(fft(x))^2/5 # the periodogram (as a check)
[1] 16.2  1.37082  .029179  .029179  1.37082
# I(0)  I(1/5)  I(2/5)  I(3/5)  I(4/5)

```

Note that $I(0) = n\bar{x}^2 = 5 \times 1.8^2 = 16.2$. Also, the sum of squares associated with the residuals (SSE) is zero, indicating an exact fit.

Example 4.11 Spectral Analysis as Principal Component Analysis

It is also possible to think of spectral analysis as a principal component analysis. In Section C.5, we show that the spectral density may be thought of as the approximate eigenvalues of the covariance matrix of a stationary process. If $X = (x_1, \dots, x_n)$ are n values of a mean-zero time series, x_t with spectral density $f_x(\omega)$, then

$$\text{cov}(X) = \Gamma_n = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(n-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(n-1) & \gamma(n-2) & \cdots & \gamma(0) \end{bmatrix}.$$

For n sufficiently large, the eigenvalues of Γ_n are

$$\lambda_j \approx f(\omega_j) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i h j / n},$$

with approximate eigenvectors

$$g_j^* = \frac{1}{\sqrt{n}} (e^{-2\pi i 0j/n}, e^{-2\pi i 1j/n}, \dots, e^{-2\pi i (n-1)j/n}),$$

for $j = 0, 1, \dots, n-1$. If we let G be the complex matrix with columns g_j , then the complex vector $Y = G^* X$ has elements that are the DFTs,

$$y_j = \frac{1}{\sqrt{n}} \sum_{t=1}^n x_t e^{-2\pi i t j / n}$$

for $j = 0, 1, \dots, n-1$. In this case, the elements of Y are asymptotically uncorrelated complex random variables, with mean-zero and variance $f(\omega_j)$. Also, X may be recovered as $X = GY$, so that $x_t = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} y_j e^{2\pi i t j / n}$.

We are now ready to present some large sample properties of the periodogram. First, let μ be the mean of a stationary process x_t with absolutely summable autocovariance function $\gamma(h)$ and spectral density $f(\omega)$. We can use the same argument as in (4.30), replacing \bar{x} by μ in (4.29), to write

$$I(\omega_j) = n^{-1} \sum_{h=-(n-1)}^{n-1} \sum_{t=1}^{n-|h|} (x_{t+|h|} - \mu)(x_t - \mu) e^{-2\pi i \omega_j h} \quad (4.35)$$

where ω_j is a non-zero fundamental frequency. Taking expectation in (4.35) we obtain

$$\mathbb{E}[I(\omega_j)] = \sum_{h=-(n-1)}^{n-1} \left(\frac{n-|h|}{n} \right) \gamma(h) e^{-2\pi i \omega_j h}. \quad (4.36)$$

For any given $\omega \neq 0$, choose a sequence of fundamental frequencies $\omega_{j:n} \rightarrow \omega$ ^{4.5} from which it follows by (4.36) that, as $n \rightarrow \infty$ ^{4.6}

$$\mathbb{E}[I(\omega_{j:n})] \rightarrow f(\omega) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i h \omega}. \quad (4.37)$$

In other words, under absolute summability of $\gamma(h)$, the spectral density is the long-term average of the periodogram.

Additional asymptotic properties may be established under the condition that the autocovariance function satisfies

$$\theta = \sum_{h=-\infty}^{\infty} |h| |\gamma(h)| < \infty. \quad (4.38)$$

First, we note that straight-forward calculations lead to

$$\text{cov}[d_c(\omega_j), d_c(\omega_k)] = n^{-1} \sum_{s=1}^n \sum_{t=1}^n \gamma(s-t) \cos(2\pi \omega_j s) \cos(2\pi \omega_k t), \quad (4.39)$$

$$\text{cov}[d_c(\omega_j), d_s(\omega_k)] = n^{-1} \sum_{s=1}^n \sum_{t=1}^n \gamma(s-t) \cos(2\pi \omega_j s) \sin(2\pi \omega_k t), \quad (4.40)$$

$$\text{cov}[d_s(\omega_j), d_s(\omega_k)] = n^{-1} \sum_{s=1}^n \sum_{t=1}^n \gamma(s-t) \sin(2\pi \omega_j s) \sin(2\pi \omega_k t), \quad (4.41)$$

where the variance terms are obtained by setting $\omega_j = \omega_k$ in (4.39) and (4.41). In [Appendix C, Section C.2](#), we show the terms in (4.39)–(4.41) have interesting properties under assumption that (4.38) holds. In particular, for $\omega_j, \omega_k \neq 0$ or $1/2$,

$$\text{cov}[d_c(\omega_j), d_c(\omega_k)] = \begin{cases} f(\omega_j)/2 + \varepsilon_n & \omega_j = \omega_k, \\ \varepsilon_n & \omega_j \neq \omega_k, \end{cases} \quad (4.42)$$

^{4.5} By this we mean $\omega_{j:n} = j_n/n$, where $\{j_n\}$ is a sequence of integers chosen so that j_n/n is the closest Fourier frequency to ω ; consequently, $|j_n/n - \omega| \leq \frac{1}{2n}$.

^{4.6} From [Definition 4.2](#) we have $I(0) = n\bar{x}^2$, so the analogous result of (4.37) for the case $\omega = 0$ is $\mathbb{E}[I(0)] - n\mu^2 = n \text{var}(\bar{x}) \rightarrow f(0)$ as $n \rightarrow \infty$.

$$\text{cov}[d_s(\omega_j), d_s(\omega_k)] = \begin{cases} f(\omega_j)/2 + \varepsilon_n & \omega_j = \omega_k, \\ \varepsilon_n & \omega_j \neq \omega_k, \end{cases} \quad (4.43)$$

and

$$\text{cov}[d_c(\omega_j), d_s(\omega_k)] = \varepsilon_n, \quad (4.44)$$

where the error term ε_n in the approximations can be bounded,

$$|\varepsilon_n| \leq \theta/n, \quad (4.45)$$

and θ is given by (4.38). If $\omega_j = \omega_k = 0$ or $1/2$ in (4.42), the multiplier $1/2$ disappears; note that $d_s(0) = d_s(1/2) = 0$, so (4.43) does not apply in these cases.

Example 4.12 Covariance of Sine and Cosine Transforms

For the three-point moving average series of Example 1.9 and $n = 256$ observations, the theoretical covariance matrix of the vector $D = (d_c(\omega_{26}), d_s(\omega_{26}), d_c(\omega_{27}), d_s(\omega_{27}))'$ using (4.39)–(4.41) is

$$\text{cov}(D) = \begin{pmatrix} .3752 & -.0009 & -.0022 & -.0010 \\ -.0009 & .3777 & -.0009 & .0003 \\ -.0022 & -.0009 & .3667 & -.0010 \\ -.0010 & .0003 & -.0010 & .3692 \end{pmatrix}.$$

The diagonal elements can be compared with half the theoretical spectral values of $\frac{1}{2}f(\omega_{26}) = .3774$ for the spectrum at frequency $\omega_{26} = 26/256$, and of $\frac{1}{2}f(\omega_{27}) = .3689$ for the spectrum at $\omega_{27} = 27/256$. Hence, the cosine and sine transforms produce nearly uncorrelated variables with variances approximately equal to one half of the theoretical spectrum. For this particular case, the uniform bound is determined from $\theta = 8/9$, yielding $|\varepsilon_{256}| \leq .0035$ for the bound on the approximation error.

If $x_t \sim \text{iid}(0, \sigma^2)$, then it follows from (4.38)–(4.44), and a central limit theorem^{4.7} that

$$d_c(\omega_{j:n}) \sim \text{AN}(0, \sigma^2/2) \quad \text{and} \quad d_s(\omega_{j:n}) \sim \text{AN}(0, \sigma^2/2) \quad (4.46)$$

jointly and independently, and independent of $d_c(\omega_{k:n})$ and $d_s(\omega_{k:n})$ provided $\omega_{j:n} \rightarrow \omega_1$ and $\omega_{k:n} \rightarrow \omega_2$ where $0 < \omega_1 \neq \omega_2 < 1/2$. We note that in this case, $f_x(\omega) = \sigma^2$. In view of (4.46), it follows immediately that as $n \rightarrow \infty$,

$$\frac{2I(\omega_{j:n})}{\sigma^2} \xrightarrow{d} \chi_2^2 \quad \text{and} \quad \frac{2I(\omega_{k:n})}{\sigma^2} \xrightarrow{d} \chi_2^2 \quad (4.47)$$

with $I(\omega_{j:n})$ and $I(\omega_{k:n})$ being asymptotically independent, where χ_v^2 denotes a chi-squared random variable with v degrees of freedom. If the process is also Gaussian, then the above statements are true for any sample size.

Using the central limit theory of Section C.2, it is fairly easy to extend the results of the iid case to the case of a linear process.

^{4.7} If $\{Y_j\} \sim \text{iid}(0, \sigma^2)$ and $\{a_j\}$ are constants for which $\sum_{j=1}^n a_j^2 / \max_{1 \leq j \leq n} a_j^2 \rightarrow \infty$ as $n \rightarrow \infty$, then $\sum_{j=1}^n a_j Y_j \sim \text{AN}\left(0, \sigma^2 \sum_{j=1}^n a_j^2\right)$. AN is read *asymptotically normal*; see Definition A.5.

Property 4.6 Distribution of the Periodogram Ordinates

If

$$x_t = \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}, \quad \sum_{j=-\infty}^{\infty} |\psi_j| < \infty \quad (4.48)$$

where $w_t \sim iid(0, \sigma_w^2)$, and (4.38) holds, then for any collection of m distinct frequencies $\omega_j \in (0, 1/2)$ with $\omega_{j:n} \rightarrow \omega_j$

$$\frac{2I(\omega_{j:n})}{f(\omega_j)} \xrightarrow{d} iid \chi_2^2 \quad (4.49)$$

provided $f(\omega_j) > 0$, for $j = 1, \dots, m$.

This result is stated more precisely in [Theorem C.7](#). Other approaches to large sample normality of the periodogram ordinates are in terms of cumulants, as in Brillinger (1981), or in terms of mixing conditions, such as in Rosenblatt (1956a). Here, we adopt the approach used by Hannan (1970), Fuller (1996), and Brockwell and Davis (1991).

The distributional result (4.49) can be used to derive an approximate *confidence interval for the spectrum* in the usual way. Let $\chi_v^2(\alpha)$ denote the lower α probability tail for the chi-squared distribution with v degrees of freedom; that is,

$$\Pr\{\chi_v^2 \leq \chi_v^2(\alpha)\} = \alpha. \quad (4.50)$$

Then, an approximate $100(1-\alpha)\%$ confidence interval for the spectral density function would be of the form

$$\frac{2 I(\omega_{j:n})}{\chi_2^2(1 - \alpha/2)} \leq f(\omega) \leq \frac{2 I(\omega_{j:n})}{\chi_2^2(\alpha/2)}. \quad (4.51)$$

Often, trends are present that should be eliminated before computing the periodogram. Trends introduce extremely low frequency components in the periodogram that tend to obscure the appearance at higher frequencies. For this reason, it is usually conventional to center the data prior to a spectral analysis using either mean-adjusted data of the form $x_t - \bar{x}$ to eliminate the zero or d-c component or to use detrended data of the form $x_t - \hat{\beta}_1 - \hat{\beta}_2 t$ to eliminate the term that will be considered a half cycle by the spectral analysis. Note that higher order polynomial regressions in t or nonparametric smoothing (linear filtering) could be used in cases where the trend is nonlinear.

As previously indicated, it is often convenient to calculate the DFTs, and hence the periodogram, using the fast Fourier transform algorithm. The FFT utilizes a number of redundancies in the calculation of the DFT when n is highly composite; that is, an integer with many factors of 2, 3, or 5, the best case being when $n = 2^p$ is a factor of 2. Details may be found in Cooley and Tukey (1965). To accommodate this property, we can pad the centered (or detrended) data of length n to the next highly composite integer n' by adding zeros, i.e., setting $x_{n+1}^c = x_{n+2}^c = \dots = x_{n'}^c = 0$, where x_t^c

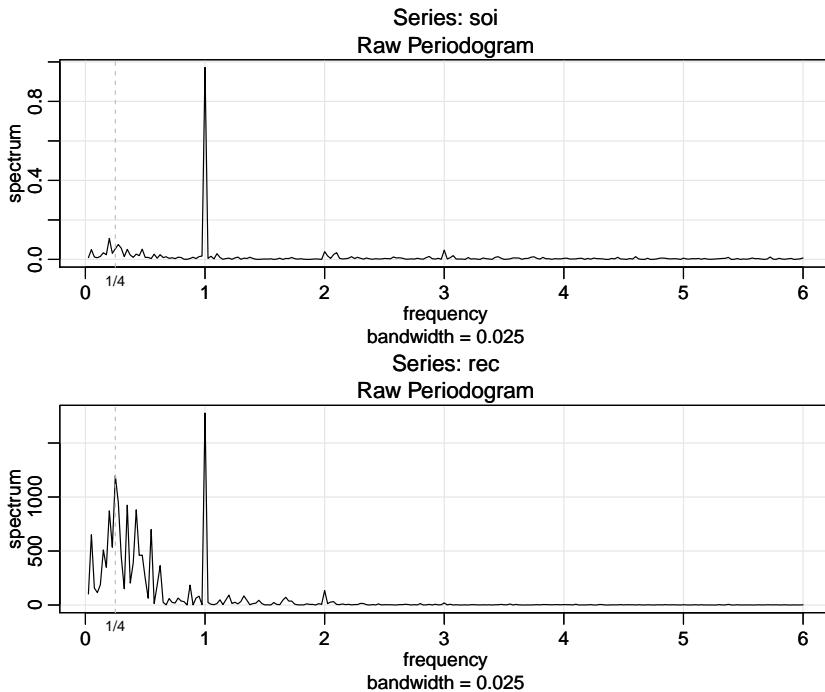


Fig. 4.5. Periodogram of SOI and Recruitment, $n = 453$ ($n' = 480$), where the frequency axis is labeled in multiples of $\Delta = 1/12$. Note the common peaks at $\omega = 1\Delta = 1/12$, or one cycle per year (12 months), and some larger values near $\omega = \frac{1}{4}\Delta = 1/48$, or one cycle every four years (48 months).

denotes the centered data. This means that the fundamental frequency ordinates will be $\omega_j = j/n'$ instead of j/n . We illustrate by considering the periodogram of the SOI and Recruitment series shown in Figure 1.5. Recall that they are monthly series and $n = 453$ months. To find n' in R, use the command `nextn(453)` to see that $n' = 480$ will be used in the spectral analyses by default.

Example 4.13 Periodogram of SOI and Recruitment Series

Figure 4.5 shows the periodograms of each series, where the frequency axis is labeled in multiples of $\Delta = 1/12$. As previously indicated, the centered data have been padded to a series of length 480. We notice a narrow-band peak at the obvious yearly (12 month) cycle, $\omega = 1\Delta = 1/12$. In addition, there is considerable power in a wide band at the lower frequencies that is centered around the four-year (48 month) cycle $\omega = \frac{1}{4}\Delta = 1/48$ representing a possible El Niño effect. This wide band activity suggests that the possible El Niño cycle is irregular, but tends to be around four years on average. We will continue to address this problem as we move to more sophisticated analyses.

Noting $\chi^2_2(.025) = .05$ and $\chi^2_2(.975) = 7.38$, we can obtain approximate 95% confidence intervals for the frequencies of interest. For example, the periodogram

of the SOI series is $I_S(1/12) = .97$ at the yearly cycle. An approximate 95% confidence interval for the spectrum $f_S(1/12)$ is then

$$[2(.97)/7.38, 2(.97)/.05] = [.26, 38.4],$$

which is too wide to be of much use. We do notice, however, that the lower value of .26 is higher than any other periodogram ordinate, so it is safe to say that this value is significant. On the other hand, an approximate 95% confidence interval for the spectrum at the four-year cycle, $f_S(1/48)$, is

$$[2(.05)/7.38, 2(.05)/.05] = [.01, 2.12],$$

which again is extremely wide, and with which we are unable to establish significance of the peak.

We now give the R commands that can be used to reproduce Figure 4.5. To calculate and graph the periodogram, we used the `mvspec` command in available from `astsa`. We note that the value of Δ is the reciprocal of the value of `frequency` for the data of a time series object. If the data are not a time series object, `frequency` is set to 1. Also, we set `log="no"` because the periodogram is plotted on a \log_{10} scale by default. Figure 4.5 displays a `bandwidth`. We will discuss bandwidth in the next section, so ignore this for the time being.

```
par(mfrow=c(2,1))
soi.per = mvspec(soi, log="no")
abline(v=1/4, lty=2)
rec.per = mvspec(rec, log="no")
abline(v=1/4, lty=2)
```

The confidence intervals for the SOI series at the yearly cycle, $\omega = 1/12 = 40/480$, and the possible El Niño cycle of four years $\omega = 1/48 = 10/480$ can be computed in R as follows:

```
soi.per$spec[40] # 0.97223; soi pgram at freq 1/12 = 40/480
soi.per$spec[10] # 0.05372; soi pgram at freq 1/48 = 10/480
# conf intervals - returned value:
U = qchisq(.025,2) # 0.05063
L = qchisq(.975,2) # 7.37775
2*soi.per$spec[10]/L # 0.01456
2*soi.per$spec[10]/U # 2.12220
2*soi.per$spec[40]/L # 0.26355
2*soi.per$spec[40]/U # 38.40108
```

The preceding example made it clear that the periodogram as an estimator is susceptible to large uncertainties, and we need to find a way to reduce the variance. Not surprisingly, this result follows if consider (4.49) and the fact that, for any n , the periodogram is based on only two observations. Recall that the mean and variance of the χ^2_ν distribution are ν and 2ν , respectively. Thus, using (4.49), we have $I(\omega) \sim \frac{1}{2}f(\omega)\chi^2_2$, implying

$$\text{E}[I(\omega)] \approx f(\omega) \quad \text{and} \quad \text{var}[I(\omega)] \approx f^2(\omega).$$

Consequently, $\text{var}[I(\omega)] \not\rightarrow 0$ as $n \rightarrow \infty$ and thus the periodogram is not a consistent estimator of the spectral density. The solution to this dilemma can be resolved by smoothing the periodogram.

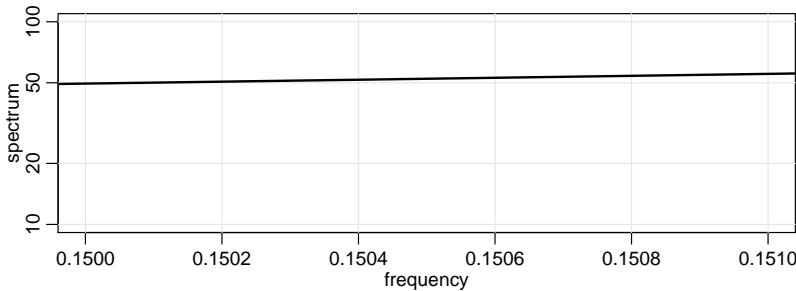


Fig. 4.6. A small section (near the peak) of the AR(2) spectrum shown in Figure 4.4 .

4.4 Nonparametric Spectral Estimation

To continue the discussion that ended the previous section, we introduce a *frequency band*, \mathcal{B} , of $L \ll n$ contiguous fundamental frequencies, centered around frequency $\omega_j = j/n$, which is chosen close to a frequency of interest, ω . For frequencies of the form $\omega^* = \omega_j + k/n$, let

$$\mathcal{B} = \left\{ \omega^*: \omega_j - \frac{m}{n} \leq \omega^* \leq \omega_j + \frac{m}{n} \right\}, \quad (4.52)$$

where

$$L = 2m + 1 \quad (4.53)$$

is an odd number, chosen such that the spectral values in the interval \mathcal{B} ,

$$f(\omega_j + k/n), \quad k = -m, \dots, 0, \dots, m$$

are approximately equal to $f(\omega)$. This structure can be realized for large sample sizes, as shown formally in Section C.2. Values of the spectrum in this band should be relatively constant for the smoothed spectra defined below to be good estimators. For example, to see a small section of the AR(2) spectrum (near the peak) shown in Figure 4.4, use

```
arma.spec(ar=c(1,-.9), xlim=c(.15,.151), n.freq=100000)
```

which is displayed in Figure 4.6.

We now define an averaged (or smoothed) periodogram as the average of the periodogram values, say,

$$\bar{f}(\omega) = \frac{1}{L} \sum_{k=-m}^m I(\omega_j + k/n), \quad (4.54)$$

over the band \mathcal{B} . Under the assumption that the spectral density is fairly constant in the band \mathcal{B} , and in view of (4.49) we can show that under appropriate conditions,^{4.8}

^{4.8} The conditions, which are sufficient, are that x_t is a linear process, as described in Property 4.6, with $\sum_j \sqrt{|j|} |\psi_j| < \infty$, and w_t has a finite fourth moment.

for large n , the periodograms in (4.54) are approximately distributed as independent $f(\omega)\chi_2^2/2$ random variables, for $0 < \omega < 1/2$, as long as we keep L fairly small relative to n . This result is discussed formally in [Section C.2](#). Thus, under these conditions, $L\bar{f}(\omega)$ is the sum of L approximately independent $f(\omega)\chi_2^2/2$ random variables. It follows that, for large n ,

$$\frac{2L\bar{f}(\omega)}{f(\omega)} \sim \chi_{2L}^2 \quad (4.55)$$

where \sim means *is approximately distributed as*.

In this scenario, where we smooth the periodogram by simple averaging, it seems reasonable to call the width of the frequency interval defined by (4.52),

$$B = \frac{L}{n}, \quad (4.56)$$

the *bandwidth*.^{4.9} The concept of bandwidth, however, becomes more complicated with the introduction of spectral estimators that smooth with unequal weights. Note that (4.56) implies the degrees of freedom can be expressed as

$$2L = 2Bn, \quad (4.57)$$

or twice the *time-bandwidth product*. The result (4.55) can be rearranged to obtain an approximate $100(1 - \alpha)\%$ confidence interval of the form

$$\frac{2L\bar{f}(\omega)}{\chi_{2L}^2(1 - \alpha/2)} \leq f(\omega) \leq \frac{2L\bar{f}(\omega)}{\chi_{2L}^2(\alpha/2)} \quad (4.58)$$

for the true spectrum, $f(\omega)$.

Many times, the visual impact of a spectral density plot will be improved by plotting the logarithm of the spectrum instead of the spectrum (the log transformation is the variance stabilizing transformation in this situation). This phenomenon can occur when regions of the spectrum exist with peaks of interest much smaller than some of the main power components. Taking logs in (4.58), we obtain an interval for the logged spectrum given by

$$\left[\log \bar{f}(\omega) - a_L, \log \bar{f}(\omega) + b_L \right] \quad (4.59)$$

where

^{4.9} There are many definitions of bandwidth and an excellent discussion may be found in Percival and Walden (1993, §6.7). The bandwidth value used in R for `spec.pgram` is based on Grenander (1951). The basic idea is that bandwidth can be related to the standard deviation of the weighting distribution. For the uniform distribution on the frequency range $-m/n$ to m/n , the standard deviation is $L/n\sqrt{12}$ (using a continuity correction). Consequently, in the case of (4.54), R will report a bandwidth of $L/n\sqrt{12}$, which amounts to dividing our definition by $\sqrt{12}$. Note that in the extreme case $L = n$, we would have $B = 1$ indicating that everything was used in the estimation. In this case, R would report a bandwidth of $1/\sqrt{12} \approx .29$, which seems to miss the point.

$$a_L = -\log 2L + \log \chi^2_{2L}(1 - \alpha/2) \quad \text{and} \quad b_L = \log 2L - \log \chi^2_{2L}(\alpha/2)$$

do not depend on ω .

If zeros are appended before computing the spectral estimators, we need to adjust the degrees of freedom (because you do not get more information by padding) and an approximation is to replace $2L$ by $2Ln/n'$. Hence, we define the *adjusted degrees of freedom* as

$$df = \frac{2Ln}{n'} \quad (4.60)$$

and use it instead of $2L$ in the confidence intervals (4.58) and (4.59). For example, (4.58) becomes

$$\frac{df\bar{f}(\omega)}{\chi^2_{df}(1 - \alpha/2)} \leq f(\omega) \leq \frac{df\bar{f}(\omega)}{\chi^2_{df}(\alpha/2)}. \quad (4.61)$$

A number of assumptions are made in computing the approximate confidence intervals given above, which may not hold in practice. In such cases, it may be reasonable to employ resampling techniques such as one of the parametric bootstraps proposed by Hurvich and Zeger (1987) or a nonparametric *local bootstrap* proposed by Paparoditis and Politis (1999). To develop the bootstrap distributions, we assume that the contiguous DFTs in a frequency band of the form (4.52) all came from a time series with identical spectrum $f(\omega)$. This, in fact, is exactly the same assumption made in deriving the large-sample theory. We may then simply resample the L DFTs in the band, with replacement, calculating a spectral estimate from each bootstrap sample. The sampling distribution of the bootstrap estimators approximates the distribution of the nonparametric spectral estimator. For further details, including the theoretical properties of such estimators, see Paparoditis and Politis (1999).

Before proceeding further, we consider computing the average periodograms for the SOI and Recruitment series.

Example 4.14 Averaged Periodogram for SOI and Recruitment

Generally, it is a good idea to try several bandwidths that seem to be compatible with the general overall shape of the spectrum, as suggested by the periodogram. We will discuss this problem in more detail after the example. The SOI and Recruitment series periodograms, previously computed in Figure 4.5, suggest the power in the lower El Niño frequency needs smoothing to identify the predominant overall period. Trying values of L leads to the choice $L = 9$ as a reasonable value, and the result is displayed in Figure 4.7.

The smoothed spectra shown provide a sensible compromise between the noisy version, shown in Figure 4.5, and a more heavily smoothed spectrum, which might lose some of the peaks. An undesirable effect of averaging can be noticed at the yearly cycle, $\omega = 1\Delta$, where the narrow band peaks that appeared in the periodograms in Figure 4.5 have been flattened and spread out to nearby frequencies. We also notice, and have marked, the appearance of *harmonics* of the yearly cycle, that is, frequencies of the form $\omega = k\Delta$ for $k = 1, 2, \dots$. Harmonics typically occur when a periodic non-sinusoidal component is present; see Example 4.15.

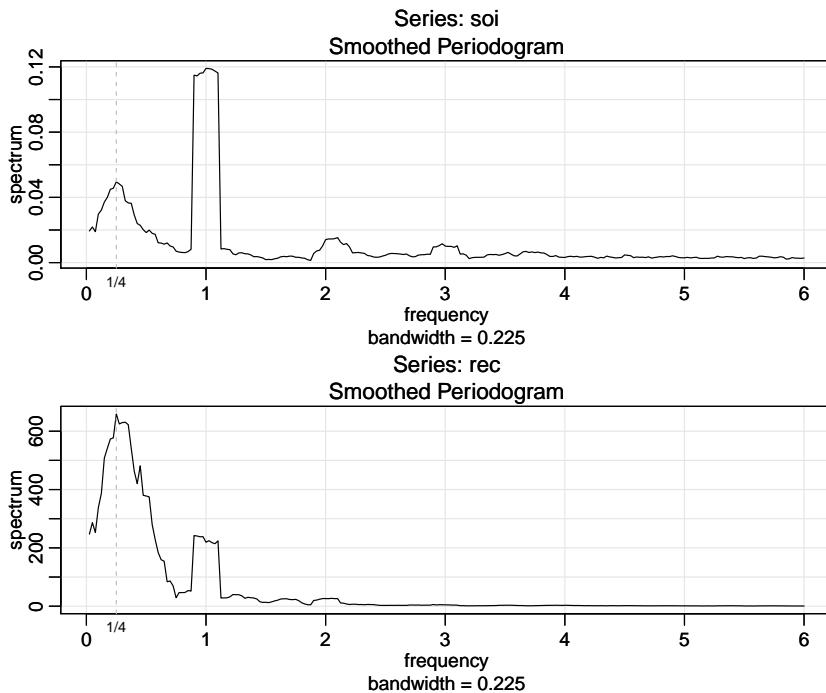


Fig. 4.7. The averaged periodogram of the SOI and Recruitment series $n = 453$, $n' = 480$, $L = 9$, $df = 17$, showing common peaks at the four year period, $\omega = \frac{1}{4}\Delta = 1/48$ cycles/month, the yearly period, $\omega = 1\Delta = 1/12$ cycles/month and some of its harmonics $\omega = k\Delta$ for $k = 2, 3$.

Figure 4.7 can be reproduced in R using the following commands. To compute averaged periodograms, use the Daniell kernel, and specify m , where $L = 2m + 1$ ($L = 9$ and $m = 4$ in this example). We will explain the kernel concept later in this section, specifically just prior to Example 4.16.

```
soi.ave = mvspec(soi, kernel('daniell',4)), log='no')
abline(v=c(.25,1,2,3), lty=2)
soi.ave$bandwidth      # = 0.225
# Repeat above lines using rec in place of soi on line 3
```

The displayed bandwidth (.225) is adjusted for the fact that the frequency scale of the plot is in terms of cycles per year instead of cycles per month. Using (4.56), the bandwidth in terms of months is $9/480 = .01875$; the displayed value is simply converted to years, $.01875 \times 12 = .225$.

The adjusted degrees of freedom are $df = 2(9)(453)/480 \approx 17$. We can use this value for the 95% confidence intervals, with $\chi^2_{df}(.025) = 7.56$ and $\chi^2_{df}(.975) = 30.17$. Substituting into (4.61) gives the intervals in Table 4.1 for the two frequency bands identified as having the maximum power. To examine the two peak power possibilities, we may look at the 95% confidence intervals and see whether the lower limits are substantially larger than adjacent baseline spectral levels. For example,

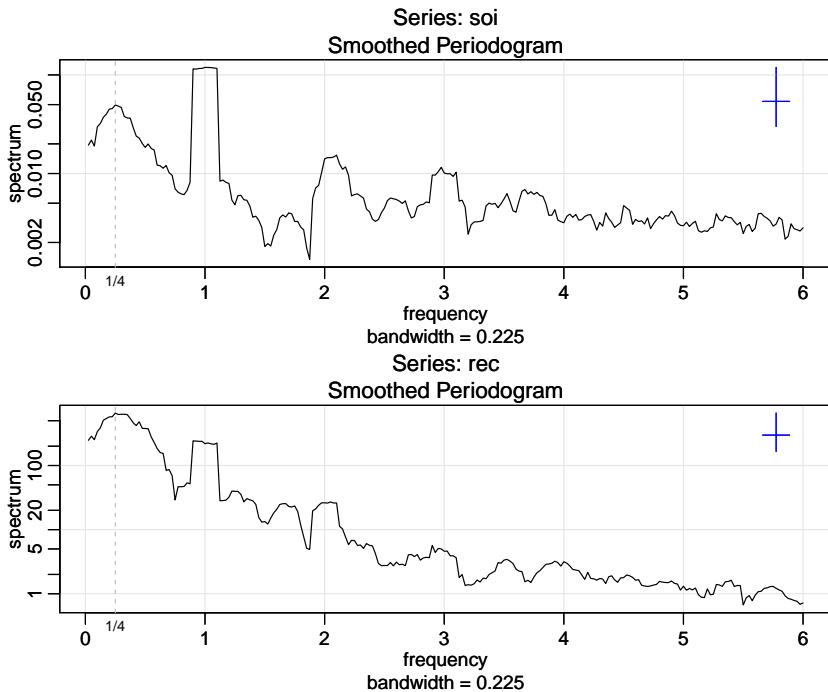


Fig. 4.8. Figure 4.7 with the average periodogram ordinates plotted on a \log_{10} scale. The display in the upper right-hand corner represents a generic 95% confidence interval where the middle tick mark is the width of the bandwidth.

the El Niño frequency of 48 months has lower limits that exceed the values the spectrum would have if there were simply a smooth underlying spectral function without the peaks. The relative distribution of power over frequencies is different, with the SOI having less power at the lower frequency, relative to the seasonal periods, and the Recruitment series having more power at the lower or El Niño frequency.

The entries in Table 4.1 for SOI can be obtained in R as follows:

```
df = soi.ave$df      # df = 16.9875 (returned values)
U = qchisq(.025, df) # U = 7.555916
L = qchisq(.975, df) # L = 30.17425
soi.ave$spec[10]      # 0.0495202
soi.ave$spec[40]      # 0.1190800
# intervals
df*soi.ave$spec[10]/L # 0.0278789
df*soi.ave$spec[10]/U # 0.1113333
df*soi.ave$spec[40]/L # 0.0670396
df*soi.ave$spec[40]/U # 0.2677201
# repeat above commands with soi replaced by rec
```

Finally, Figure 4.8 shows the averaged periodograms in Figure 4.7 plotted on a \log_{10} scale. This is the default can be obtained by removing the statement `log="no"`.

Table 4.1. Confidence Intervals for the Spectra of the SOI and Recruitment Series

Series	ω	Period	Power	Lower	Upper
SOI	1/48	4 years	.05	.03	.11
	1/12	1 year	.12	.07	.27
Recruits $\times 10^2$	1/48	4 years	6.59	3.71	14.82
	1/12	1 year	2.19	1.24	4.93

Notice that the default plot also shows a generic confidence interval of the form (4.59) (with log replaced by \log_{10}) in the upper right-hand corner. To use it, imagine placing the middle tick mark (the width of which is the bandwidth) on the averaged periodogram ordinate of interest; the resulting bar then constitutes an approximate 95% confidence interval for the spectrum at that frequency. We note that displaying the estimates on a log scale tends to emphasize the harmonic components.

Example 4.15 Harmonics

In the previous example, we saw that the spectra of the annual signals displayed minor peaks at the harmonics; that is, the signal spectra had a large peak at $\omega = 1\Delta = 1/12$ cycles/month (the one-year cycle) and minor peaks at its harmonics $\omega = k\Delta$ for $k = 2, 3, \dots$ (two-, three-, and so on, cycles per year). This will often be the case because most signals are not perfect sinusoids (or perfectly cyclic). In this case, the harmonics are needed to capture the non-sinusoidal behavior of the signal. As an example, consider the signal formed in Figure 4.9 from a (fundamental) sinusoid oscillating at two cycles per unit time along with the second through sixth harmonics at decreasing amplitudes. In particular, the signal was formed as

$$\begin{aligned} x_t = & \sin(2\pi 2t) + .5 \sin(2\pi 4t) + .4 \sin(2\pi 6t) \\ & + .3 \sin(2\pi 8t) + .2 \sin(2\pi 10t) + .1 \sin(2\pi 12t) \end{aligned} \quad (4.62)$$

for $0 \leq t \leq 1$. Notice that the signal is non-sinusoidal in appearance and rises quickly then falls slowly.

A figure similar to Figure 4.9 can be generated in R as follows.

```
t = seq(0, 1, by=1/200)
amps = c(1, .5, .4, .3, .2, .1)
x = matrix(0, 201, 6)
for (j in 1:6){ x[,j] = amps[j]*sin(2*pi*t*2^2*j) }
x = ts(cbind(x, rowSums(x)), start=0, deltat=1/200)
ts.plot(x, lty=c(1:6, 1), lwd=c(rep(1,6), 2), ylab="Sinusoids")
names = c("Fundamental", "2nd Harmonic", "3rd Harmonic", "4th Harmonic", "5th
Harmonic", "6th Harmonic", "Formed Signal")
legend("topright", names, lty=c(1:6, 1), lwd=c(rep(1,6), 2))
```

Example 4.14 points out the necessity for having some relatively systematic procedure for deciding whether peaks are significant. The question of deciding whether a single peak is significant usually rests on establishing what we might think of as a baseline level for the spectrum, defined rather loosely as the shape that one would

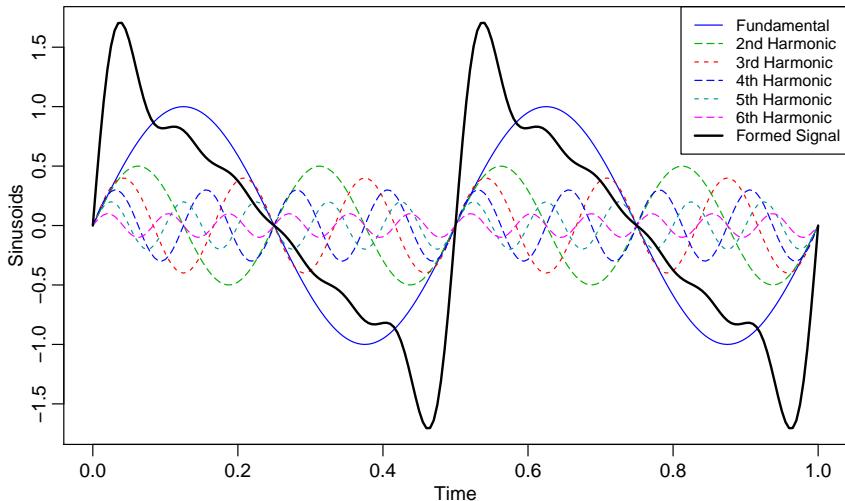


Fig. 4.9. A signal (thick solid line) formed by a fundamental sinusoid (thin solid line) oscillating at two cycles per unit time and its harmonics as specified in (4.62).

expect to see if no spectral peaks were present. This profile can usually be guessed by looking at the overall shape of the spectrum that includes the peaks; usually, a kind of baseline level will be apparent, with the peaks seeming to emerge from this baseline level. If the lower confidence limit for the spectral value is still greater than the baseline level at some predetermined level of significance, we may claim that frequency value as a statistically significant peak. To be consistent with our stated indifference to the upper limits, we might use a one-sided confidence interval.

An important aspect of interpreting the significance of confidence intervals and tests involving spectra is that typically, more than one frequency will be of interest, so that we will potentially be interested in *simultaneous statements* about a whole collection of frequencies. For example, it would be unfair to claim in Table 4.1 the two frequencies of interest as being statistically significant and all other potential candidates as nonsignificant at the overall level of $\alpha = .05$. In this case, we follow the usual statistical approach, noting that if K statements S_1, S_2, \dots, S_k are made at significance level α , i.e., $P\{S_k\} = 1 - \alpha$, then the overall probability all statements are true satisfies the *Bonferroni inequality*

$$P\{\text{all } S_k \text{ true}\} \geq 1 - K\alpha. \quad (4.63)$$

For this reason, it is desirable to set the significance level for testing each frequency at α/K if there are K potential frequencies of interest. If, a priori, potentially $K = 10$ frequencies are of interest, setting $\alpha = .01$ would give an overall significance level of bound of .10.

The use of the confidence intervals and the necessity for smoothing requires that we make a decision about the bandwidth B over which the spectrum will be essentially

constant. Taking too broad a band will tend to smooth out valid peaks in the data when the constant variance assumption is not met over the band. Taking too narrow a band will lead to confidence intervals so wide that peaks are no longer statistically significant. Thus, we note that there is a conflict here between variance properties or *bandwidth stability*, which can be improved by increasing B and *resolution*, which can be improved by decreasing B . A common approach is to try a number of different bandwidths and to look qualitatively at the spectral estimators for each case.

To address the problem of resolution, it should be evident that the flattening of the peaks in [Figure 4.7](#) and [Figure 4.8](#) was due to the fact that simple averaging was used in computing $\bar{f}(\omega)$ defined in [\(4.54\)](#). There is no particular reason to use simple averaging, and we might improve the estimator by employing a weighted average, say

$$\hat{f}(\omega) = \sum_{k=-m}^m h_k I(\omega_j + k/n), \quad (4.64)$$

using the same definitions as in [\(4.54\)](#) but where the weights $h_k > 0$ satisfy

$$\sum_{k=-m}^m h_k = 1.$$

In particular, it seems reasonable that the resolution of the estimator will improve if we use weights that decrease as distance from the center weight h_0 increases; we will return to this idea shortly. To obtain the averaged periodogram, $\bar{f}(\omega)$, in [\(4.64\)](#), set $h_k = L^{-1}$, for all k , where $L = 2m + 1$. The asymptotic theory established for $\bar{f}(\omega)$ still holds for $\hat{f}(\omega)$ provided that the weights satisfy the additional condition that if $m \rightarrow \infty$ as $n \rightarrow \infty$ but $m/n \rightarrow 0$, then

$$\sum_{k=-m}^m h_k^2 \rightarrow 0.$$

Under these conditions, as $n \rightarrow \infty$,

- (i) $E(\hat{f}(\omega)) \rightarrow f(\omega)$
- (ii) $\left(\sum_{k=-m}^m h_k^2\right)^{-1} \text{cov}(\hat{f}(\omega), \hat{f}(\lambda)) \rightarrow f^2(\omega) \quad \text{for } \omega = \lambda \neq 0, 1/2.$

In (ii), replace $f^2(\omega)$ by 0 if $\omega \neq \lambda$ and by $2f^2(\omega)$ if $\omega = \lambda = 0$ or $1/2$.

We have already seen these results in the case of $\bar{f}(\omega)$, where the weights are constant, $h_k = L^{-1}$, in which case $\sum_{k=-m}^m h_k^2 = L^{-1}$. The distributional properties of [\(4.64\)](#) are more difficult now because $\hat{f}(\omega)$ is a weighted linear combination of asymptotically independent χ^2 random variables. An approximation that seems to work well is to replace L by $\left(\sum_{k=-m}^m h_k^2\right)^{-1}$. That is, define

$$L_h = \left(\sum_{k=-m}^m h_k^2 \right)^{-1} \quad (4.65)$$

and use the approximation^{4.10}

$$\frac{2L_h \hat{f}(\omega)}{f(\omega)} \stackrel{\sim}{\sim} \chi^2_{2L_h}. \quad (4.66)$$

In analogy to (4.56), we will define the bandwidth in this case to be

$$B = \frac{L_h}{n}. \quad (4.67)$$

Using the approximation (4.66) we obtain an approximate $100(1 - \alpha)\%$ confidence interval of the form

$$\frac{2L_h \hat{f}(\omega)}{\chi^2_{2L_h}(1 - \alpha/2)} \leq f(\omega) \leq \frac{2L_h \hat{f}(\omega)}{\chi^2_{2L_h}(\alpha/2)} \quad (4.68)$$

for the true spectrum, $f(\omega)$. If the data are padded to n' , then replace $2L_h$ in (4.68) with $df = 2L_h n / n'$ as in (4.60).

An easy way to generate the weights in R is by repeated use of the *Daniell kernel*. For example, with $m = 1$ and $L = 2m + 1 = 3$, the Daniell kernel has weights $\{h_k\} = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$; applying this kernel to a sequence of numbers, $\{u_t\}$, produces

$$\hat{u}_t = \frac{1}{3}u_{t-1} + \frac{1}{3}u_t + \frac{1}{3}u_{t+1}.$$

We can apply the same kernel again to the \hat{u}_t ,

$$\hat{\hat{u}}_t = \frac{1}{3}\hat{u}_{t-1} + \frac{1}{3}\hat{u}_t + \frac{1}{3}\hat{u}_{t+1},$$

which simplifies to

$$\hat{\hat{u}}_t = \frac{1}{9}u_{t-2} + \frac{2}{9}u_{t-1} + \frac{3}{9}u_t + \frac{2}{9}u_{t+1} + \frac{1}{9}u_{t+2}.$$

The *modified Daniell kernel* puts half weights at the end points, so with $m = 1$ the weights are $\{h_k\} = \{\frac{1}{4}, \frac{2}{4}, \frac{1}{4}\}$ and

$$\hat{u}_t = \frac{1}{4}u_{t-1} + \frac{1}{2}u_t + \frac{1}{4}u_{t+1}.$$

Applying the same kernel again to \hat{u}_t yields

$$\hat{\hat{u}}_t = \frac{1}{16}u_{t-2} + \frac{4}{16}u_{t-1} + \frac{6}{16}u_t + \frac{4}{16}u_{t+1} + \frac{1}{16}u_{t+2}.$$

These coefficients can be obtained in R by issuing the `kernel` command. For example, `kernel("modified.daniell", c(1,1))` would produce the coefficients of the last example. The other kernels that are currently available in R are the Dirichlet kernel and the Fejér kernel, which we will discuss shortly.

It is interesting to note that these kernel weights form a probability distribution. If X and Y are independent discrete uniforms on the integers $\{-1, 0, 1\}$ each with probability $\frac{1}{3}$, then the convolution $X + Y$ is discrete on the integers $\{-2, -1, 0, 1, 2\}$ with corresponding probabilities $\{\frac{1}{9}, \frac{2}{9}, \frac{3}{9}, \frac{2}{9}, \frac{1}{9}\}$.

^{4.10} The approximation proceeds as follows: If $\hat{f} \stackrel{\sim}{\sim} c\chi^2_\nu$, where c is a constant, then $E\hat{f} \approx cv$ and $\text{var}\hat{f} \approx f^2 \sum_k h_k^2 \approx c^2 2\nu$. Solving, $c \approx f \sum_k h_k^2 / 2 = f / 2L_h$ and $\nu \approx 2 \left(\sum_k h_k^2 \right)^{-1} = 2L_h$.

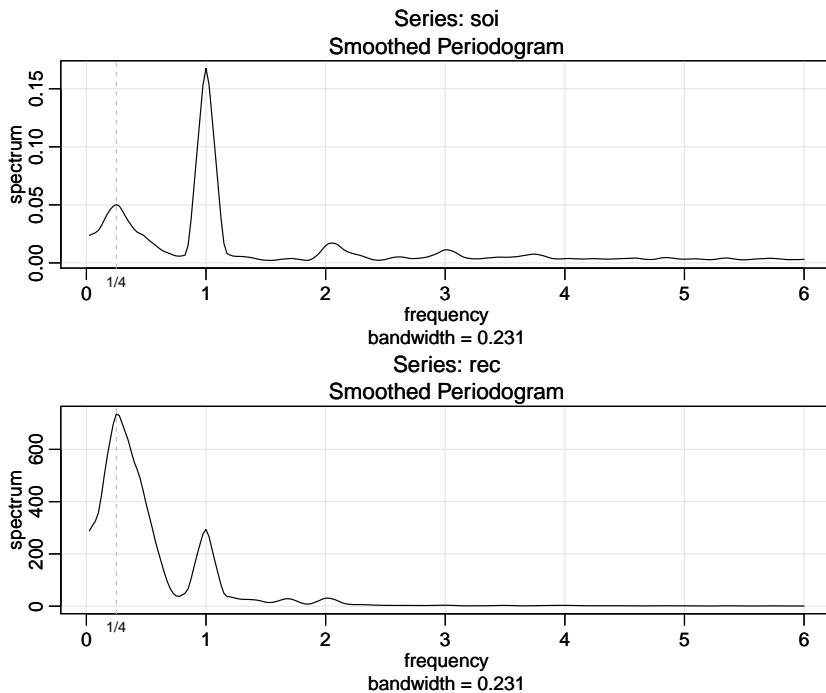


Fig. 4.10. Smoothed (tapered) spectral estimates of the SOI and Recruitment series; see Example 4.16 for details.

Example 4.16 Smoothed Periodogram for SOI and Recruitment

In this example, we estimate the spectra of the SOI and Recruitment series using the smoothed periodogram estimate in (4.64). We used a modified Daniell kernel twice, with $m = 3$ both times. This yields $L_h = 1/\sum_{k=-m}^m h_k^2 = 9.232$, which is close to the value of $L = 9$ used in Example 4.14. In this case, the bandwidth is $B = 9.232/480 = .019$ and the modified degrees of freedom is $df = 2L_h453/480 = 17.43$. The weights, h_k , can be obtained and graphed in R as follows:

```
kernel("modified.daniell", c(3,3))
  coef[-6] = 0.006944 = coef[ 6]
  coef[-5] = 0.027778 = coef[ 5]
  coef[-4] = 0.055556 = coef[ 4]
  coef[-3] = 0.083333 = coef[ 3]
  coef[-2] = 0.111111 = coef[ 2]
  coef[-1] = 0.138889 = coef[ 1]
  coef[ 0] = 0.152778
plot(kernel("modified.daniell", c(3,3))) # not shown
```

The resulting spectral estimates can be viewed in Figure 4.10 and we notice that the estimates more appealing than those in Figure 4.7. Figure 4.10 was generated in R as follows; we also show how to obtain the associated bandwidth and degrees of freedom.

```

k = kernel("modified.daniell", c(3,3))
soi.smo = mvspec(soi, kernel=k, taper=.1, log="no")
abline(v=c(.25,1), lty=2)
## Repeat above lines with rec replacing soi in line 3
df = soi.smo$df      # df = 17.42618
soi.smo$bandwidth     # B = 0.2308103

```

Note that a *taper* was applied in the estimation process; we discuss tapering in the next part. Reissuing the `mvspec` commands with `log="no"` removed will result in a figure similar to [Figure 4.8](#). Finally, we mention that the modified Daniell kernel is used by default and an easier way to obtain `soi.smo` is to issue the command:

```
soi.smo = mvspec(soi, taper=.1, spans=c(7,7))
```

Notice that `spans` is a vector of odd integers, given in terms of $L = 2m + 1$ instead of m .

There have been many attempts at dealing with the problem of smoothing the periodogram in a automatic way; an early reference is Wahba (1980). It is apparent from [Example 4.16](#) that the smoothing bandwidth for the broadband El Niño behavior (near the 4 year cycle), should be much larger than the bandwidth for the annual cycle (the 1 year cycle). Consequently, it is perhaps better to perform automatic adaptive smoothing for estimating the spectrum. We refer interested readers to Fan and Kreutzberger (1998) and the numerous references within.

TAPERING

We are now ready to introduce the concept of *tapering*; a more detailed discussion may be found in Bloomfield (2000, §9.5). Suppose x_t is a mean-zero, stationary process with spectral density $f_x(\omega)$. If we replace the original series by the tapered series

$$y_t = h_t x_t, \quad (4.69)$$

for $t = 1, 2, \dots, n$, use the modified DFT

$$d_y(\omega_j) = n^{-1/2} \sum_{t=1}^n h_t x_t e^{-2\pi i \omega_j t}, \quad (4.70)$$

and let $I_y(\omega_j) = |d_y(\omega_j)|^2$, we obtain (see [Problem 4.17](#))

$$E[I_y(\omega_j)] = \int_{-\frac{1}{2}}^{\frac{1}{2}} W_n(\omega_j - \omega) f_x(\omega) d\omega \quad (4.71)$$

where

$$W_n(\omega) = |H_n(\omega)|^2 \quad (4.72)$$

and

$$H_n(\omega) = n^{-1/2} \sum_{t=1}^n h_t e^{-2\pi i \omega t}. \quad (4.73)$$

The value $W_n(\omega)$ is called a *spectral window* because, in view of (4.71), it is determining which part of the spectral density $f_x(\omega)$ is being “seen” by the estimator

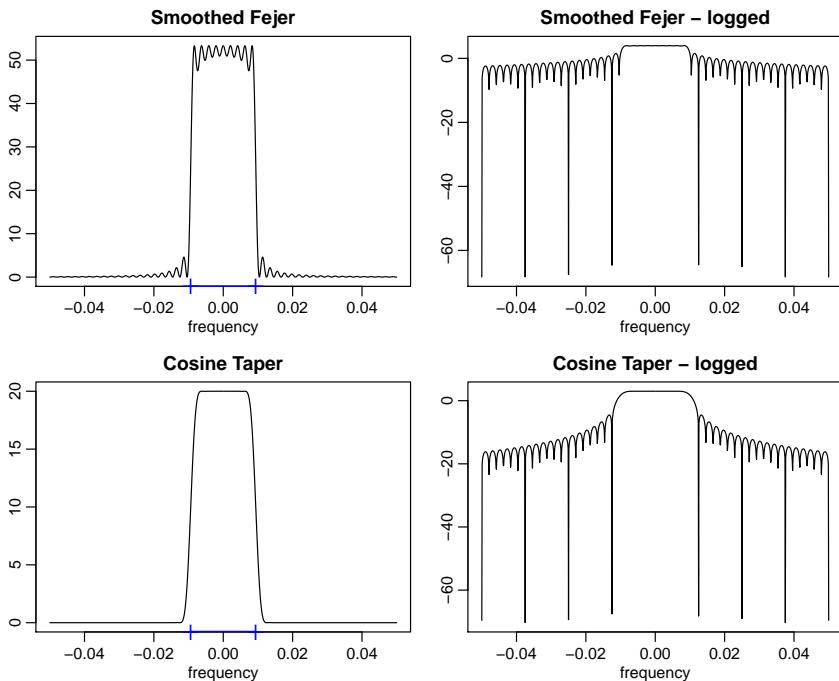


Fig. 4.11. Averaged Fejér window (top row) and the corresponding cosine taper window (bottom row) for $L = 9$, $n = 480$. The extra tic marks on the horizontal axis of the left-hand plots exhibit the predicted bandwidth, $B = 9/480 = .01875$.

$I_y(\omega_j)$ on average. In the case that $h_t = 1$ for all t , $I_y(\omega_j) = I_x(\omega_j)$ is simply the periodogram of the data and the window is

$$W_n(\omega) = \frac{\sin^2(n\pi\omega)}{n \sin^2(\pi\omega)} \quad (4.74)$$

with $W_n(0) = n$, which is known as the Fejér or modified Bartlett kernel. If we consider the averaged periodogram in (4.54), namely

$$\tilde{f}_x(\omega) = \frac{1}{L} \sum_{k=-m}^m I_x(\omega_j + k/n),$$

the window, $W_n(\omega)$, in (4.71) will take the form

$$W_n(\omega) = \frac{1}{nL} \sum_{k=-m}^m \frac{\sin^2[n\pi(\omega + k/n)]}{\sin^2[\pi(\omega + k/n)]}. \quad (4.75)$$

Tapers generally have a shape that enhances the center of the data relative to the extremities, such as a cosine bell of the form

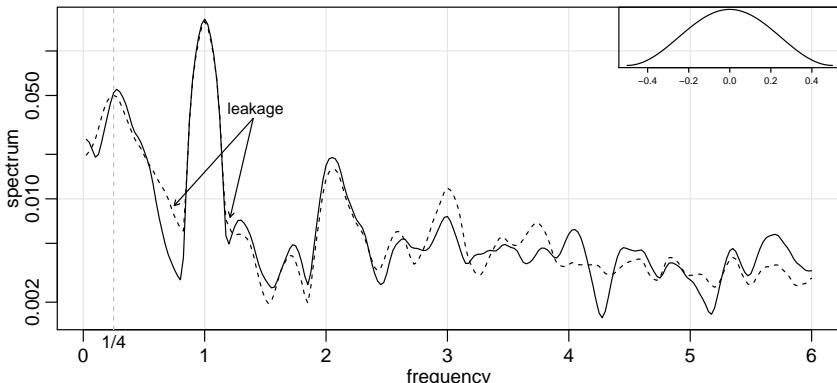


Fig. 4.12. Smoothed spectral estimates of the SOI without tapering (dashed line) and with full tapering (solid line); see [Example 4.17](#). The insert shows a full cosine bell taper, (4.76), with horizontal axis $(t - \bar{t})/n$, for $t = 1, \dots, n$.

$$h_t = .5 \left[1 + \cos\left(\frac{2\pi(t - \bar{t})}{n}\right) \right], \quad (4.76)$$

where $\bar{t} = (n + 1)/2$, favored by Blackman and Tukey (1959). The shape of this taper is shown in the insert to [Figure 4.12](#). In [Figure 4.11](#), we have plotted the shapes of two windows, $W_n(\omega)$, for $n = 480$ and $L = 9$, when (i) $h_t \equiv 1$, in which case, (4.75) applies, and (ii) h_t is the cosine taper in (4.76). In both cases the predicted bandwidth should be $B = 9/480 = .01875$ cycles per point, which corresponds to the “width” of the windows shown in [Figure 4.11](#). Both windows produce an integrated average spectrum over this band but the untapered window in the top panels shows considerable ripples over the band and outside the band. The ripples outside the band are called sidelobes and tend to introduce frequencies from outside the interval that may contaminate the desired spectral estimate within the band. For example, a large dynamic range for the values in the spectrum introduces spectra in contiguous frequency intervals several orders of magnitude greater than the value in the interval of interest. This effect is sometimes called *leakage*. [Figure 4.11](#) emphasizes the suppression of the sidelobes in the Fejér kernel when a cosine taper is used.

Example 4.17 The Effect of Tapering the SOI Series

The estimates in [Example 4.16](#) were obtained by tapering the upper and lower 10% of the data. In this example, we examine the effect of tapering on the estimate of the spectrum of the SOI series (the results for the Recruitment series are similar). [Figure 4.12](#) shows two spectral estimates plotted on a log scale. The dashed line in [Figure 4.12](#) shows the estimate without any tapering. The solid line shows the result with full tapering. Notice that the tapered spectrum does a better job in separating the yearly cycle ($\omega = 1$) and the El Niño cycle ($\omega = 1/4$).

The following R session was used to generate [Figure 4.12](#). We note that, by default, `mvspec` does not taper. For full tapering, we use the argument `taper=.5` to

instruct `mvspec` to taper 50% of each end of the data; any value between 0 and .5 is acceptable. In Example 4.16, we used `taper=.1`.

```
s0 = mvspec(soi, spans=c(7,7), plot=FALSE)           # no taper
s50 = mvspec(soi, spans=c(7,7), taper=.5, plot=FALSE)  # full taper
plot(s50$freq, s50$spec, log="y", type="l", ylab="spectrum",
      xlab="frequency")                                # solid line
lines(s0$freq, s0$spec, lty=2)                         # dashed line
```

We close this section with a brief discussion of *lag window* estimators. First, consider the periodogram, $I(\omega_j)$, which was shown in (4.30) to be

$$I(\omega_j) = \sum_{|h|<n} \hat{\gamma}(h) e^{-2\pi i \omega_j h}.$$

Thus, (4.64) can be written as

$$\begin{aligned} \hat{f}(\omega) &= \sum_{|k| \leq m} h_k I(\omega_j + k/n) = \sum_{|k| \leq m} h_k \sum_{|h|<n} \hat{\gamma}(h) e^{-2\pi i (\omega_j + k/n) h} \\ &= \sum_{|h|<n} g\left(\frac{h}{n}\right) \hat{\gamma}(h) e^{-2\pi i \omega_j h}. \end{aligned} \quad (4.77)$$

where $g\left(\frac{h}{n}\right) = \sum_{|k| \leq m} h_k \exp(-2\pi i kh/n)$. Equation (4.77) suggests estimators of the form

$$\tilde{f}(\omega) = \sum_{|h| \leq r} w\left(\frac{h}{r}\right) \hat{\gamma}(h) e^{-2\pi i \omega h} \quad (4.78)$$

where $w(\cdot)$ is a weight function, called the lag window, that satisfies

- (i) $w(0) = 1$
- (ii) $|w(x)| \leq 1$ and $w(x) = 0$ for $|x| > 1$,
- (iii) $w(x) = w(-x)$.

Note that if $w(x) = 1$ for $|x| < 1$ and $r = n$, then $\tilde{f}(\omega_j) = I(\omega_j)$, the periodogram. This result indicates the problem with the periodogram as an estimator of the spectral density is that it gives too much weight to the values of $\hat{\gamma}(h)$ when h is large, and hence is unreliable [e.g., there is only one pair of observations used in the estimate $\hat{\gamma}(n-1)$, and so on]. The smoothing window is defined to be

$$W(\omega) = \sum_{h=-r}^r w\left(\frac{h}{r}\right) e^{-2\pi i \omega h}, \quad (4.79)$$

and it determines which part of the periodogram will be used to form the estimate of $f(\omega)$. The asymptotic theory for $\hat{f}(\omega)$ holds for $\tilde{f}(\omega)$ under the same conditions and provided $r \rightarrow \infty$ as $n \rightarrow \infty$ but with $r/n \rightarrow 0$. That is,

$$E\{\tilde{f}(\omega)\} \rightarrow f(\omega), \quad (4.80)$$

$$\frac{n}{r} \text{cov}(\tilde{f}(\omega), \tilde{f}(\lambda)) \rightarrow f^2(\omega) \int_{-1}^1 w^2(x) dx \quad \omega = \lambda \neq 0, 1/2. \quad (4.81)$$

In (4.81), replace $f^2(\omega)$ by 0 if $\omega \neq \lambda$ and by $2f^2(\omega)$ if $\omega = \lambda = 0$ or $1/2$.

Many authors have developed various windows and Brillinger (2001, Ch 3) and Brockwell and Davis (1991, Ch 10) are good sources of detailed information on this topic.

4.5 Parametric Spectral Estimation

The methods of the previous section lead to what is generally referred to as *non-parametric spectral estimators* because no assumption is made about the parametric form of the spectral density. In [Property 4.4](#), we exhibited the spectrum of an ARMA process and we might consider basing a spectral estimator on this function, substituting the parameter estimates from an ARMA(p, q) fit on the data into the formula for the spectral density $f_x(\omega)$ given in (4.23). Such an estimator is called a parametric spectral estimator. For convenience, a parametric spectral estimator is obtained by fitting an AR(p) to the data, where the order p is determined by one of the model selection criteria, such as AIC, AICc, and BIC, defined in (2.15)–(2.17). Parametric autoregressive spectral estimators will often have superior resolution in problems when several closely spaced narrow spectral peaks are present and are preferred by engineers for a broad variety of problems (see Kay, 1988). The development of autoregressive spectral estimators has been summarized by Parzen (1983).

If $\hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_p$ and $\hat{\sigma}_w^2$ are the estimates from an AR(p) fit to x_t , then based on [Property 4.4](#), a parametric spectral estimate of $f_x(\omega)$ is attained by substituting these estimates into (4.23), that is,

$$\hat{f}_x(\omega) = \frac{\hat{\sigma}_w^2}{|\hat{\phi}(e^{-2\pi i \omega})|^2}, \quad (4.82)$$

where

$$\hat{\phi}(z) = 1 - \hat{\phi}_1 z - \hat{\phi}_2 z^2 - \cdots - \hat{\phi}_p z^p. \quad (4.83)$$

The asymptotic distribution of the autoregressive spectral estimator has been obtained by Berk (1974) under the conditions $p \rightarrow \infty$, $p^3/n \rightarrow 0$ as $p, n \rightarrow \infty$, which may be too severe for most applications. The limiting results imply a confidence interval of the form

$$\frac{\hat{f}_x(\omega)}{(1 + Cz_{\alpha/2})} \leq f_x(\omega) \leq \frac{\hat{f}_x(\omega)}{(1 - Cz_{\alpha/2})}, \quad (4.84)$$

where $C = \sqrt{2p/n}$ and $z_{\alpha/2}$ is the ordinate corresponding to the upper $\alpha/2$ probability of the standard normal distribution. If the sampling distribution is to be checked, we suggest applying the bootstrap estimator to get the sampling distribution of $\hat{f}_x(\omega)$ using a procedure similar to the one used for $p = 1$ in [Example 3.36](#). An alternative for higher order autoregressive series is to put the AR(p) in state-space form and use the bootstrap procedure discussed in [Section 6.7](#).

An interesting fact about rational spectra of the form (4.23) is that any spectral density can be approximated, arbitrarily close, by the spectrum of an AR process.

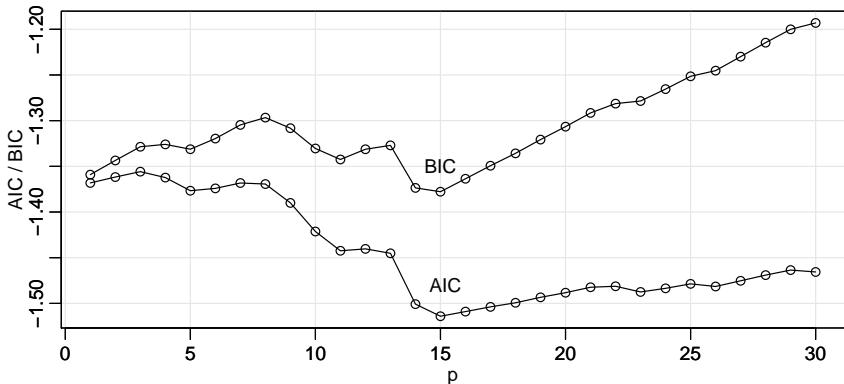


Fig. 4.13. Model selection criteria AIC and BIC as a function of order p for autoregressive models fitted to the SOI series.

Property 4.7 AR Spectral Approximation

Let $g(\omega)$ be the spectral density of a stationary process. Then, given $\epsilon > 0$, there is a time series with the representation

$$x_t = \sum_{k=1}^p \phi_k x_{t-k} + w_t$$

where w_t is white noise with variance σ_w^2 , such that

$$|f_x(\omega) - g(\omega)| < \epsilon \quad \text{for all } \omega \in [-1/2, 1/2].$$

Moreover, p is finite and the roots of $\phi(z) = 1 - \sum_{k=1}^p \phi_k z^k$ are outside the unit circle.

One drawback of the property is that it does not tell us how large p must be before the approximation is reasonable; in some situations p may be extremely large. Property 4.7 also holds for MA and for ARMA processes in general, and a proof of the result may be found in Section C.6. We demonstrate the technique in the following example.

Example 4.18 Autoregressive Spectral Estimator for SOI

Consider obtaining results comparable to the nonparametric estimators shown in Figure 4.7 for the SOI series. Fitting successively higher order AR(p) models for $p = 1, 2, \dots, 30$ yields a minimum BIC and a minimum AIC at $p = 15$, as shown in Figure 4.13. We can see from Figure 4.13 that BIC is very definite about which model it chooses; that is, the minimum BIC is very distinct. On the other hand, it is not clear what is going to happen with AIC; that is, the minimum is not so clear, and there is some concern that AIC will start decreasing after $p = 30$. Minimum AICc selects the $p = 15$ model, but suffers from the same uncertainty as AIC. The spectrum is shown in Figure 4.14, and we note the strong peaks near the four year

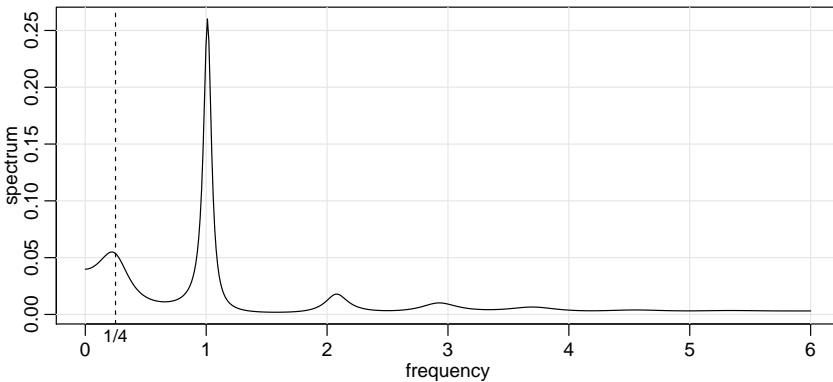


Fig. 4.14. Autoregressive spectral estimator for the SOI series using the AR(15) model selected by AIC, AICC, and BIC.

and one year cycles as in the nonparametric estimates obtained in Section 4.4. In addition, the harmonics of the yearly period are evident in the estimated spectrum.

To perform a similar analysis in R, the command `spec.ar` can be used to fit the best model via AIC and plot the resulting spectrum. A quick way to obtain the AIC values is to run the `ar` command as follows.

```
spaic = spec.ar(soi, log="no")           # min AIC spec
abline(v=frequency(soi)*1/52, lty=3)     # El Nino peak
(soi.ar = ar(soi, order.max=30))         # estimates and AICs
dev.new()
plot(1:30, soi.ar$aic[-1], type="o")      # plot AICs
```

No likelihood is calculated here, so the use of the term AIC is loose. To generate Figure 4.13 we used the following code to (loosely) obtain AIC, AICC, and BIC. Because AIC and AICC are nearly identical in this example, we only graphed AIC and BIC+1; we added 1 to the BIC to reduce white space in the graphic.

```
n = length(soi)
AIC = rep(0, 30) -> AICc -> BIC
for (k in 1:30){
  sigma2 = ar(soi, order=k, aic=FALSE)$var.pred
  BIC[k] = log(sigma2) + (k*log(n)/n)
  AICc[k] = log(sigma2) + ((n+k)/(n-k-2))
  AIC[k] = log(sigma2) + ((n+2*k)/n)
}
IC = cbind(AIC, BIC+1)
ts.plot(IC, type="o", xlab="p", ylab="AIC / BIC")
```

Finally, it should be mentioned that any parametric spectrum, say $f(\omega; \theta)$, depending on the vector parameter θ can be estimated via the Whittle likelihood (Whittle, 1961), using the approximate properties of the discrete Fourier transform derived in Appendix C. We have that the DFTs, $d(\omega_j)$, are approximately complex normally distributed with mean zero and variance $f(\omega_j; \theta)$ and are approximately independent for $\omega_j \neq \omega_k$. This implies that an approximate log likelihood can be written in the form

$$\ln L(x; \theta) \approx - \sum_{0 < \omega_j < 1/2} \left(\ln f_x(\omega_j; \theta) + \frac{|d(\omega_j)|^2}{f_x(\omega_j; \theta)} \right), \quad (4.85)$$

where the sum is sometimes expanded to include the frequencies $\omega_j = 0, 1/2$. If the form with the two additional frequencies is used, the multiplier of the sum will be unity, except for the purely real points at $\omega_j = 0, 1/2$ for which the multiplier is 1/2. For a discussion of applying the Whittle approximation to the problem of estimating parameters in an ARMA spectrum, see Anderson (1978). The Whittle likelihood is especially useful for fitting long memory models that will be discussed in [Chapter 5](#).

4.6 Multiple Series and Cross-Spectra

The notion of analyzing frequency fluctuations using classical statistical ideas extends to the case in which there are several jointly stationary series, for example, x_t and y_t . In this case, we can introduce the idea of a correlation indexed by frequency, called the *coherence*. The results in [Section C.2](#) imply the covariance function

$$\gamma_{xy}(h) = E[(x_{t+h} - \mu_x)(y_t - \mu_y)]$$

has the representation

$$\gamma_{xy}(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} f_{xy}(\omega) e^{2\pi i \omega h} d\omega \quad h = 0, \pm 1, \pm 2, \dots, \quad (4.86)$$

where the *cross-spectrum* is defined as the Fourier transform

$$f_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h) e^{-2\pi i \omega h} \quad -1/2 \leq \omega \leq 1/2, \quad (4.87)$$

assuming that the cross-covariance function is absolutely summable, as was the case for the autocovariance. The cross-spectrum is generally a complex-valued function, and it is often written as

$$f_{xy}(\omega) = c_{xy}(\omega) - iq_{xy}(\omega), \quad (4.88)$$

where

$$c_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h) \cos(2\pi\omega h) \quad (4.89)$$

and

$$q_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h) \sin(2\pi\omega h) \quad (4.90)$$

are defined as the *cosppectrum* and *quadspectrum*, respectively. Because of the relationship $\gamma_{yx}(h) = \gamma_{xy}(-h)$, it follows, by substituting into (4.87) and rearranging, that

$$f_{yx}(\omega) = f_{xy}^*(\omega), \quad (4.91)$$

with $*$ denoting conjugation. This result, in turn, implies that the cospectrum and quadsspectrum satisfy

$$c_{yx}(\omega) = c_{xy}(\omega) \quad (4.92)$$

and

$$q_{yx}(\omega) = -q_{xy}(\omega). \quad (4.93)$$

An important example of the application of the cross-spectrum is to the problem of predicting an output series y_t from some input series x_t through a linear filter relation such as the three-point moving average considered below. A measure of the strength of such a relation is the *squared coherence* function, defined as

$$\rho_{y,x}^2(\omega) = \frac{|f_{yx}(\omega)|^2}{f_{xx}(\omega)f_{yy}(\omega)}, \quad (4.94)$$

where $f_{xx}(\omega)$ and $f_{yy}(\omega)$ are the individual spectra of the x_t and y_t series, respectively. Although we consider a more general form of this that applies to multiple inputs later, it is instructive to display the single input case as (4.94) to emphasize the analogy with conventional squared correlation, which takes the form

$$\rho_{yx}^2 = \frac{\sigma_{yx}^2}{\sigma_x^2 \sigma_y^2},$$

for random variables with variances σ_x^2 and σ_y^2 and covariance $\sigma_{yx} = \sigma_{xy}$. This motivates the interpretation of squared coherence and the squared correlation between two time series at frequency ω .

Example 4.19 Three-Point Moving Average

As a simple example, we compute the cross-spectrum between x_t and the three-point moving average $y_t = (x_{t-1} + x_t + x_{t+1})/3$, where x_t is a stationary input process with spectral density $f_{xx}(\omega)$. First,

$$\begin{aligned} \gamma_{xy}(h) &= \text{cov}(x_{t+h}, y_t) = \frac{1}{3} \text{cov}(x_{t+h}, x_{t-1} + x_t + x_{t+1}) \\ &= \frac{1}{3} \left[\gamma_{xx}(h+1) + \gamma_{xx}(h) + \gamma_{xx}(h-1) \right] \\ &= \frac{1}{3} \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(e^{2\pi i \omega} + 1 + e^{-2\pi i \omega} \right) e^{2\pi i \omega h} f_{xx}(\omega) d\omega \\ &= \frac{1}{3} \int_{-\frac{1}{2}}^{\frac{1}{2}} \left[1 + 2 \cos(2\pi \omega) \right] f_{xx}(\omega) e^{2\pi i \omega h} d\omega, \end{aligned}$$

where we have used (4.16). Using the uniqueness of the Fourier transform, we argue from the spectral representation (4.86) that

$$f_{xy}(\omega) = \frac{1}{3} \left[1 + 2 \cos(2\pi \omega) \right] f_{xx}(\omega)$$

so that the cross-spectrum is real in this case. Using [Property 4.3](#), the spectral density of y_t is

$$f_{yy}(\omega) = \frac{1}{9} |e^{2\pi i \omega} + 1 + e^{-2\pi i \omega}|^2 f_{xx}(\omega) = \frac{1}{9} [1 + 2 \cos(2\pi \omega)]^2 f_{xx}(\omega).$$

Substituting into [\(4.94\)](#) yields,

$$\rho_{y \cdot x}^2(\omega) = \frac{\left| \frac{1}{3} [1 + 2 \cos(2\pi \omega)] f_{xx}(\omega) \right|^2}{f_{xx}(\omega) \cdot \frac{1}{9} [1 + 2 \cos(2\pi \omega)]^2 f_{xx}(\omega)} = 1;$$

that is, the squared coherence between x_t and y_t is unity over all frequencies. This is a characteristic inherited by more general linear filters; see [Problem 4.30](#). However, if some noise is added to the three-point moving average, the coherence is not unity; these kinds of models will be considered in detail later.

Property 4.8 Spectral Representation of a Vector Stationary Process

If $x_t = (x_{t1}, x_{t2}, \dots, x_{tp})'$ is a $p \times 1$ stationary process with autocovariance matrix $\Gamma(h) = E[(x_{t+h} - \mu)(x_t - \mu)'] = \{\gamma_{jk}(h)\}$ satisfying

$$\sum_{h=-\infty}^{\infty} |\gamma_{jk}(h)| < \infty \quad (4.95)$$

for all $j, k = 1, \dots, p$, then $\Gamma(h)$ has the representation

$$\Gamma(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} f(\omega) d\omega \quad h = 0, \pm 1, \pm 2, \dots, \quad (4.96)$$

as the inverse transform of the spectral density matrix, $f(\omega) = \{f_{jk}(\omega)\}$, for $j, k = 1, \dots, p$. The matrix $f(\omega)$ has the representation

$$f(\omega) = \sum_{h=-\infty}^{\infty} \Gamma(h) e^{-2\pi i \omega h} \quad -1/2 \leq \omega \leq 1/2. \quad (4.97)$$

The spectral matrix $f(\omega)$ is Hermitian, $f(\omega) = f^*(\omega)$, where $*$ means to conjugate and transpose.

Example 4.20 Spectral Matrix of a Bivariate Process

Consider a jointly stationary bivariate process (x_t, y_t) . We arrange the autocovariances in the matrix

$$\Gamma(h) = \begin{pmatrix} \gamma_{xx}(h) & \gamma_{xy}(h) \\ \gamma_{yx}(h) & \gamma_{yy}(h) \end{pmatrix}.$$

The spectral matrix would be given by

$$f(\omega) = \begin{pmatrix} f_{xx}(\omega) & f_{xy}(\omega) \\ f_{yx}(\omega) & f_{yy}(\omega) \end{pmatrix},$$

where the Fourier transform [\(4.96\)](#) and [\(4.97\)](#) relate the autocovariance and spectral matrices.

The extension of spectral estimation to vector series is fairly obvious. For the vector series $x_t = (x_{t1}, x_{t2}, \dots, x_{tp})'$, we may use the vector of DFTs, say $d(\omega_j) = (d_1(\omega_j), d_2(\omega_j), \dots, d_p(\omega_j))'$, and estimate the spectral matrix by

$$\bar{f}(\omega) = L^{-1} \sum_{k=-m}^m I(\omega_j + k/n) \quad (4.98)$$

where now

$$I(\omega_j) = d(\omega_j) d^*(\omega_j) \quad (4.99)$$

is a $p \times p$ complex matrix. The series may be tapered before the DFT is taken in (4.98) and we can use weighted estimation,

$$\hat{f}(\omega) = \sum_{k=-m}^m h_k I(\omega_j + k/n) \quad (4.100)$$

where $\{h_k\}$ are weights as defined in (4.64). The estimate of squared coherence between two series, y_t and x_t is

$$\hat{\rho}_{y \cdot x}^2(\omega) = \frac{|\hat{f}_{yx}(\omega)|^2}{\hat{f}_{xx}(\omega) \hat{f}_{yy}(\omega)}. \quad (4.101)$$

If the spectral estimates in (4.101) are obtained using equal weights, we will write $\bar{\rho}_{y \cdot x}^2(\omega)$ for the estimate.

Under general conditions, if $\rho_{y \cdot x}^2(\omega) > 0$ then

$$|\hat{\rho}_{y \cdot x}(\omega)| \sim AN \left(|\rho_{y \cdot x}(\omega)|, (1 - \rho_{y \cdot x}^2(\omega))^2 / 2L_h \right) \quad (4.102)$$

where L_h is defined in (4.65); the details of this result may be found in Brockwell and Davis (1991, Ch 11). We may use (4.102) to obtain approximate confidence intervals for the squared coherence, $\rho_{y \cdot x}^2(\omega)$.

We may also test the null hypothesis that $\rho_{y \cdot x}^2(\omega) = 0$ if we use $\bar{\rho}_{y \cdot x}^2(\omega)$ for the estimate with $L > 1$,^{4.11} that is,

$$\bar{\rho}_{y \cdot x}^2(\omega) = \frac{|\bar{f}_{yx}(\omega)|^2}{\bar{f}_{xx}(\omega) \bar{f}_{yy}(\omega)}. \quad (4.103)$$

In this case, under the null hypothesis, the statistic

$$F = \frac{\bar{\rho}_{y \cdot x}^2(\omega)}{(1 - \bar{\rho}_{y \cdot x}^2(\omega))} (L - 1) \quad (4.104)$$

has an approximate F -distribution with 2 and $2L - 2$ degrees of freedom. When the series have been extended to length n' , we replace $2L - 2$ by $df - 2$, where df is defined in (4.60). Solving (4.104) for a particular significance level α leads to

^{4.11} If $L = 1$ then $\bar{\rho}_{y \cdot x}^2(\omega) \equiv 1$.

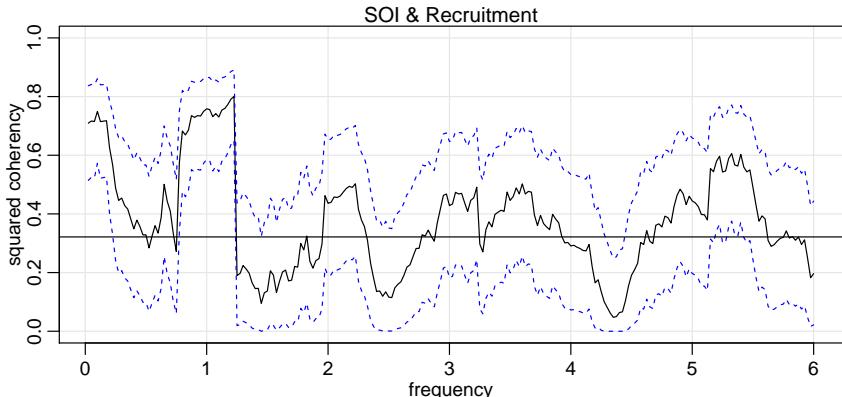


Fig. 4.15. Squared coherency between the SOI and Recruitment series; $L = 19$, $n = 453$, $n' = 480$, and $\alpha = .001$. The horizontal line is $C_{.001}$.

$$C_\alpha = \frac{F_{2,2L-2}(\alpha)}{L - 1 + F_{2,2L-2}(\alpha)} \quad (4.105)$$

as the approximate value that must be exceeded for the original squared coherence to be able to reject $\rho_{y,x}^2(\omega) = 0$ at an a priori specified frequency.

Example 4.21 Coherence Between SOI and Recruitment

Figure 4.15 shows the squared coherence between the SOI and Recruitment series over a wider band than was used for the spectrum. In this case, we used $L = 19$, $df = 2(19)(453/480) \approx 36$ and $F_{2,df-2}(.001) \approx 8.53$ at the significance level $\alpha = .001$. Hence, we may reject the hypothesis of no coherence for values of $\rho_{y,x}^2(\omega)$ that exceed $C_{.001} = .32$. We emphasize that this method is crude because, in addition to the fact that the F -statistic is approximate, we are examining the squared coherence across all frequencies with the Bonferroni inequality, (4.63), in mind. Figure 4.15 also exhibits confidence bands as part of the R plotting routine. We emphasize that these bands are only valid for ω where $\rho_{y,x}^2(\omega) > 0$.

In this case, the two series are obviously strongly coherent at the annual seasonal frequency. The series are also strongly coherent at lower frequencies that may be attributed to the El Niño cycle, which we claimed had a 3 to 7 year period. The peak in the coherency, however, occurs closer to the 9 year cycle. Other frequencies are also coherent, although the strong coherence is less impressive because the underlying power spectrum at these higher frequencies is fairly small. Finally, we note that the coherence is persistent at the seasonal harmonic frequencies.

This example may be reproduced using the following R commands.

```
sr = mvspec(cbind(soi,rec), kernel="daniell",9), plot=FALSE)
sr$df # df = 35.8625
f = qf(.999, 2, sr$df-2) # = 8.529792
C = f/(18+f) # = 0.321517
plot(sr, plot.type = "coh", ci.lty = 2)
abline(h = C)
```

4.7 Linear Filters

Some of the examples of the previous sections have hinted at the possibility the distribution of power or variance in a time series can be modified by making a linear transformation. In this section, we explore that notion further by showing how linear filters can be used to extract signals from a time series. These filters modify the spectral characteristics of a time series in a predictable way, and the systematic development of methods for taking advantage of the special properties of linear filters is an important topic in time series analysis.

Recall [Property 4.3](#) that stated if

$$y_t = \sum_{j=-\infty}^{\infty} a_j x_{t-j}, \quad \sum_{j=-\infty}^{\infty} |a_j| < \infty,$$

and x_t has spectrum $f_{xx}(\omega)$, then y_t has spectrum

$$f_{yy}(\omega) = |A_{yx}(\omega)|^2 f_{xx}(\omega),$$

where

$$A_{yx}(\omega) = \sum_{j=-\infty}^{\infty} a_j e^{-2\pi i \omega j}$$

is the *frequency response function*. This result shows that the filtering effect can be characterized as a frequency-by-frequency multiplication by the squared magnitude of the frequency response function.

Example 4.22 First Difference and Moving Average Filters

We illustrate the effect of filtering with two common examples, the first difference filter

$$y_t = \nabla x_t = x_t - x_{t-1}$$

and the annual symmetric moving average filter,

$$y_t = \frac{1}{24} (x_{t-6} + x_{t+6}) + \frac{1}{12} \sum_{r=-5}^5 x_{t-r},$$

which is a modified Daniell kernel with $m = 6$. The results of filtering the SOI series using the two filters are shown in the middle and bottom panels of [Figure 4.16](#). Notice that the effect of differencing is to roughen the series because it tends to retain the higher or faster frequencies. The centered moving average smoothes the series because it retains the lower frequencies and tends to attenuate the higher frequencies. In general, differencing is an example of a *high-pass filter* because it retains or passes the higher frequencies, whereas the moving average is a *low-pass filter* because it passes the lower or slower frequencies.

Notice that the slower periods are enhanced in the symmetric moving average and the seasonal or yearly frequencies are attenuated. The filtered series makes

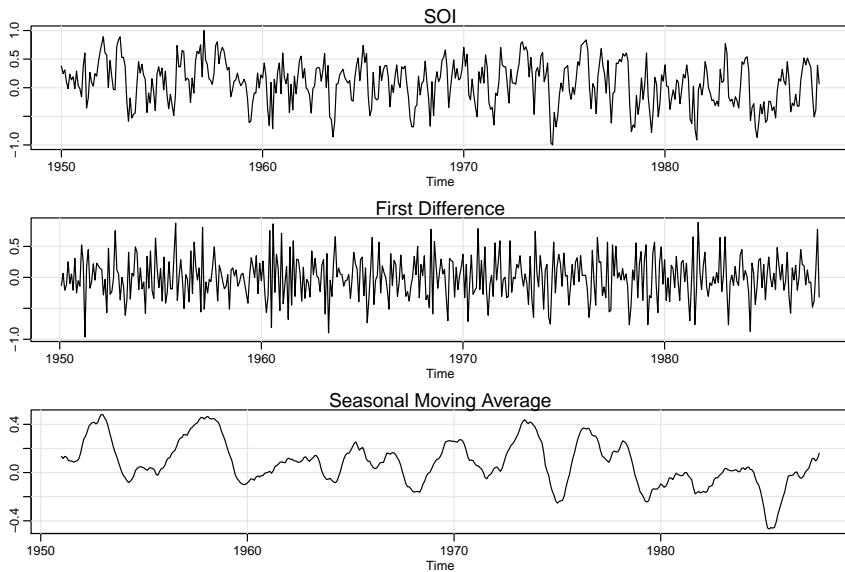


Fig. 4.16. SOI series (top) compared with the differenced SOI (middle) and a centered 12-month moving average (bottom).

about 9 cycles in the length of the data (about one cycle every 52 months) and the moving average filter tends to enhance or *extract* the El Niño signal. Moreover, by low-pass filtering the data, we get a better sense of the El Niño effect and its irregularity.

Now, having done the filtering, it is essential to determine the exact way in which the filters change the input spectrum. We shall use (4.21) and (4.22) for this purpose. The first difference filter can be written in the form (4.20) by letting $a_0 = 1$, $a_1 = -1$, and $a_r = 0$ otherwise. This implies that

$$A_{yx}(\omega) = 1 - e^{-2\pi i\omega},$$

and the squared frequency response becomes

$$|A_{yx}(\omega)|^2 = (1 - e^{-2\pi i\omega})(1 - e^{2\pi i\omega}) = 2[1 - \cos(2\pi\omega)]. \quad (4.106)$$

The top panel of Figure 4.17 shows that the first difference filter will attenuate the lower frequencies and enhance the higher frequencies because the multiplier of the spectrum, $|A_{yx}(\omega)|^2$, is large for the higher frequencies and small for the lower frequencies. Generally, the slow rise of this kind of filter does not particularly recommend it as a procedure for retaining only the high frequencies.

For the centered 12-month moving average, we can take $a_{-6} = a_6 = 1/24$, $a_k = 1/12$ for $-5 \leq k \leq 5$ and $a_k = 0$ elsewhere. Substituting and recognizing the cosine terms gives

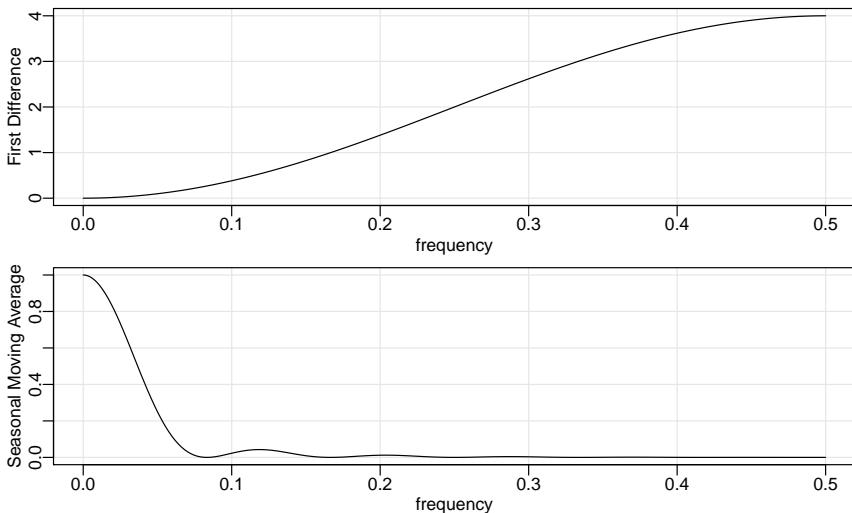


Fig. 4.17. Squared frequency response functions of the first difference (top) and twelve-month moving average (bottom) filters.

$$A_{yx}(\omega) = \frac{1}{12} \left[1 + \cos(12\pi\omega) + 2 \sum_{k=1}^5 \cos(2\pi\omega k) \right]. \quad (4.107)$$

Plotting the squared frequency response of this function as in the bottom of Figure 4.17 shows that we can expect this filter to cut most of the frequency content above .05 cycles per point, and nearly all of the frequency content above $1/12 \approx .083$. In particular, this drives down the yearly components with periods of 12 months and enhances the El Niño frequency, which is somewhat lower. The filter is not completely efficient at attenuating high frequencies; some power contributions are left at higher frequencies, as shown in the function $|A_{yx}(\omega)|^2$.

The following R session shows how to filter the data, perform the spectral analysis of a filtered series, and plot the squared frequency response curves of the difference and moving average filters.

```
par(mfrow=c(3,1), mar=c(3,3,1,1), mgp=c(1.6,.6,0))
plot(soi) # plot data
plot(diff(soi)) # plot first difference
k = kernel("modified.daniell", 6) # filter weights
plot(soif <- kernapply(soi, k)) # plot 12 month filter
dev.new()
spectrum(soif, spans=9, log="no") # spectral analysis (not shown)
abline(v=12/52, lty="dashed")
dev.new()
##-- frequency responses --#
par(mfrow=c(2,1), mar=c(3,3,1,1), mgp=c(1.6,.6,0))
w = seq(0, .5, by=.01)
FRdiff = abs(1-exp(2i*pi*w))^2
plot(w, FRdiff, type='l', xlab='frequency')
```

```

u = cos(2*pi*w)+cos(4*pi*w)+cos(6*pi*w)+cos(8*pi*w)+cos(10*pi*w)
FRma = ((1 + cos(12*pi*w) + 2*u)/12)^2
plot(w, FRma, type='l', xlab='frequency')

```

The two filters discussed in the previous example were different in that the frequency response function of the first difference was complex-valued, whereas the frequency response of the moving average was purely real. A short derivation similar to that used to verify (4.22) shows, when x_t and y_t are related by the linear filter relation (4.20), the cross-spectrum satisfies

$$f_{yx}(\omega) = A_{yx}(\omega)f_{xx}(\omega),$$

so the frequency response is of the form

$$A_{yx}(\omega) = \frac{f_{yx}(\omega)}{f_{xx}(\omega)} \quad (4.108)$$

$$= \frac{c_{yx}(\omega)}{f_{xx}(\omega)} - i \frac{q_{yx}(\omega)}{f_{xx}(\omega)}, \quad (4.109)$$

where we have used (4.88) to get the last form. Then, we may write (4.109) in polar coordinates as

$$A_{yx}(\omega) = |A_{yx}(\omega)| \exp\{-i \phi_{yx}(\omega)\}, \quad (4.110)$$

where the *amplitude* and *phase* of the filter are defined by

$$|A_{yx}(\omega)| = \frac{\sqrt{c_{yx}^2(\omega) + q_{yx}^2(\omega)}}{f_{xx}(\omega)} \quad (4.111)$$

and

$$\phi_{yx}(\omega) = \tan^{-1} \left(-\frac{q_{yx}(\omega)}{c_{yx}(\omega)} \right). \quad (4.112)$$

A simple interpretation of the phase of a linear filter is that it exhibits time delays as a function of frequency in the same way as the spectrum represents the variance as a function of frequency. Additional insight can be gained by considering the simple delaying filter

$$y_t = Ax_{t-D},$$

where the series gets replaced by a version, amplified by multiplying by A and delayed by D points. For this case,

$$f_{yx}(\omega) = Ae^{-2\pi i \omega D} f_{xx}(\omega),$$

and the amplitude is $|A|$, and the phase is

$$\phi_{yx}(\omega) = -2\pi\omega D,$$

or just a linear function of frequency ω . For this case, applying a simple time delay causes phase delays that depend on the frequency of the periodic component being delayed. Interpretation is further enhanced by setting

$$x_t = \cos(2\pi\omega t),$$

in which case

$$y_t = A \cos(2\pi\omega t - 2\pi\omega D).$$

Thus, the output series, y_t , has the same period as the input series, x_t , but the amplitude of the output has increased by a factor of $|A|$ and the phase has been changed by a factor of $-2\pi\omega D$.

Example 4.23 Difference and Moving Average Filters

We consider calculating the amplitude and phase of the two filters discussed in [Example 4.22](#). The case for the moving average is easy because $A_{yx}(\omega)$ given in (4.107) is purely real. So, the amplitude is just $|A_{yx}(\omega)|$ and the phase is $\phi_{yx}(\omega) = 0$. In general, symmetric ($a_j = a_{-j}$) filters have zero phase. The first difference, however, changes this, as we might expect from the example above involving the time delay filter. In this case, the squared amplitude is given in (4.106). To compute the phase, we write

$$\begin{aligned} A_{yx}(\omega) &= 1 - e^{-2\pi i\omega} = e^{-i\pi\omega}(e^{i\pi\omega} - e^{-i\pi\omega}) \\ &= 2ie^{-i\pi\omega} \sin(\pi\omega) = 2 \sin^2(\pi\omega) + 2i \cos(\pi\omega) \sin(\pi\omega) \\ &= \frac{c_{yx}(\omega)}{f_{xx}(\omega)} - i \frac{q_{yx}(\omega)}{f_{xx}(\omega)}, \end{aligned}$$

so

$$\phi_{yx}(\omega) = \tan^{-1}\left(-\frac{q_{yx}(\omega)}{c_{yx}(\omega)}\right) = \tan^{-1}\left(\frac{\cos(\pi\omega)}{\sin(\pi\omega)}\right).$$

Noting that

$$\cos(\pi\omega) = \sin(-\pi\omega + \pi/2)$$

and that

$$\sin(\pi\omega) = \cos(-\pi\omega + \pi/2),$$

we get

$$\phi_{yx}(\omega) = -\pi\omega + \pi/2,$$

and the phase is again a linear function of frequency.

The above tendency of the frequencies to arrive at different times in the filtered version of the series remains as one of two annoying features of the difference type filters. The other weakness is the gentle increase in the frequency response function. If low frequencies are really unimportant and high frequencies are to be preserved, we would like to have a somewhat sharper response than is obvious in [Figure 4.17](#). Similarly, if low frequencies are important and high frequencies are not, the moving average filters are also not very efficient at passing the low frequencies and attenuating the high frequencies. Improvement is possible by designing better and longer filters, but we do not discuss this here.

We will occasionally use results for multivariate series $x_t = (x_{t1}, \dots, x_{tp})'$ that are comparable to the simple property shown in [\(4.22\)](#). Consider the *matrix filter*

$$y_t = \sum_{j=-\infty}^{\infty} A_j x_{t-j}, \quad (4.113)$$

where $\{A_j\}$ denotes a sequence of $q \times p$ matrices such that $\sum_{j=-\infty}^{\infty} \|A_j\| < \infty$ and $\|\cdot\|$ denotes any matrix norm, $x_t = (x_{t1}, \dots, x_{tp})'$ is a $p \times 1$ stationary vector process with mean vector μ_x and $p \times p$, matrix covariance function $\Gamma_{xx}(h)$ and spectral matrix $f_{xx}(\omega)$, and y_t is the $q \times 1$ vector output process. Then, we can obtain the following property.

Property 4.9 Output Spectral Matrix of Filtered Vector Series

The spectral matrix of the filtered output y_t in (4.113) is related to the spectrum of the input x_t by

$$f_{yy}(\omega) = \mathcal{A}(\omega) f_{xx}(\omega) \mathcal{A}^*(\omega), \quad (4.114)$$

where the matrix frequency response function $\mathcal{A}(\omega)$ is defined by

$$\mathcal{A}(\omega) = \sum_{j=-\infty}^{\infty} A_j \exp(-2\pi i \omega j). \quad (4.115)$$

4.8 Lagged Regression Models

One of the intriguing possibilities offered by the coherence analysis of the relation between the SOI and Recruitment series discussed in [Example 4.21](#) would be extending classical regression to the analysis of lagged regression models of the form

$$y_t = \sum_{r=-\infty}^{\infty} \beta_r x_{t-r} + v_t, \quad (4.116)$$

where v_t is a stationary noise process, x_t is the observed input series, and y_t is the observed output series. We are interested in estimating the filter coefficients β_r relating the adjacent lagged values of x_t to the output series y_t .

In the case of SOI and Recruitment series, we might identify the El Niño driving series, SOI, as the input, x_t , and y_t , the Recruitment series, as the output. In general, there will be more than a single possible input series and we may envision a $q \times 1$ vector of driving series. This multivariate input situation is covered in [Chapter 7](#). The model given by (4.116) is useful under several different scenarios, corresponding to different assumptions that can be made about the components.

We assume that the inputs and outputs have zero means and are jointly stationary with the 2×1 vector process $(x_t, y_t)'$ having a spectral matrix of the form

$$f(\omega) = \begin{pmatrix} f_{xx}(\omega) & f_{xy}(\omega) \\ f_{yx}(\omega) & f_{yy}(\omega) \end{pmatrix}. \quad (4.117)$$

Here, $f_{xy}(\omega)$ is the cross-spectrum relating the input x_t to the output y_t , and $f_{xx}(\omega)$ and $f_{yy}(\omega)$ are the spectra of the input and output series, respectively. Generally, we

observe two series, regarded as input and output and search for regression functions $\{\beta_t\}$ relating the inputs to the outputs. We assume all autocovariance functions satisfy the absolute summability conditions of the form (4.38).

Then, minimizing the mean squared error

$$MSE = E \left(y_t - \sum_{r=-\infty}^{\infty} \beta_r x_{t-r} \right)^2 \quad (4.118)$$

leads to the usual orthogonality conditions

$$E \left[\left(y_t - \sum_{r=-\infty}^{\infty} \beta_r x_{t-r} \right) x_{t-s} \right] = 0 \quad (4.119)$$

for all $s = 0, \pm 1, \pm 2, \dots$. Taking the expectations inside leads to the normal equations

$$\sum_{r=-\infty}^{\infty} \beta_r \gamma_{xx}(s-r) = \gamma_{yx}(s) \quad (4.120)$$

for $s = 0, \pm 1, \pm 2, \dots$. These equations might be solved, with some effort, if the covariance functions were known exactly. If data (x_t, y_t) for $t = 1, \dots, n$ are available, we might use a finite approximation to the above equations with $\hat{\gamma}_{xx}(h)$ and $\hat{\gamma}_{yx}(h)$ substituted into (4.120). If the regression vectors are essentially zero for $|s| \geq M/2$, and $M < n$, the system (4.120) would be of full rank and the solution would involve inverting an $(M-1) \times (M-1)$ matrix.

A frequency domain approximate solution is easier in this case for two reasons. First, the computations depend on spectra and cross-spectra that can be estimated from sample data using the techniques of Section 4.5. In addition, no matrices will have to be inverted, although the frequency domain ratio will have to be computed for each frequency. In order to develop the frequency domain solution, substitute the representation (4.96) into the normal equations, using the convention defined in (4.117). The left side of (4.120) can then be written in the form

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{r=-\infty}^{\infty} \beta_r e^{2\pi i \omega(s-r)} f_{xx}(\omega) d\omega = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega s} B(\omega) f_{xx}(\omega) d\omega,$$

where

$$B(\omega) = \sum_{r=-\infty}^{\infty} \beta_r e^{-2\pi i \omega r} \quad (4.121)$$

is the Fourier transform of the regression coefficients β_t . Now, because $\gamma_{yx}(s)$ is the inverse transform of the cross-spectrum $f_{yx}(\omega)$, we might write the system of equations in the frequency domain, using the uniqueness of the Fourier transform, as

$$B(\omega) f_{xx}(\omega) = f_{yx}(\omega), \quad (4.122)$$

which then become the analogs of the usual normal equations. Then, we may take

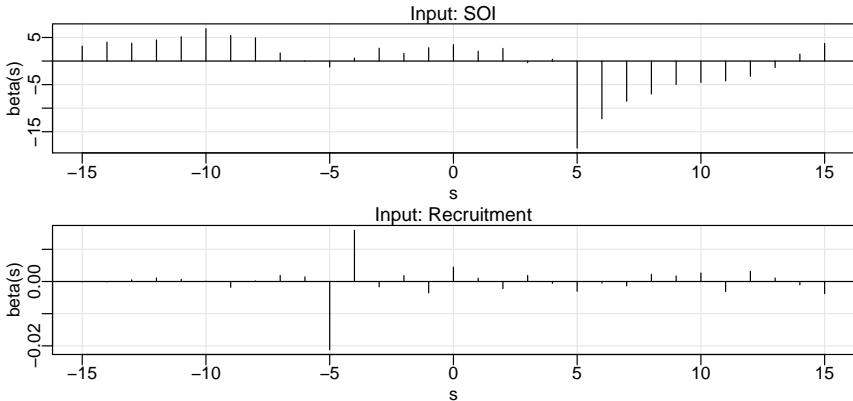


Fig. 4.18. Estimated impulse response functions relating SOI to Recruitment (top) and Recruitment to SOI (bottom) $L = 15, M = 32$.

$$\hat{B}(\omega_k) = \frac{\hat{f}_{yx}(\omega_k)}{\hat{f}_{xx}(\omega_k)} \quad (4.123)$$

as the estimator for the Fourier transform of the regression coefficients, evaluated at some subset of fundamental frequencies $\omega_k = k/M$ with $M \ll n$. Generally, we assume smoothness of $B(\cdot)$ over intervals of the form $\{\omega_k + \ell/n; \ell = -m, \dots, 0, \dots, m\}$, with $L = 2m + 1$. The inverse transform of the function $\hat{B}(\omega)$ would give $\hat{\beta}_t$, and we note that the discrete time approximation can be taken as

$$\hat{\beta}_t = M^{-1} \sum_{k=0}^{M-1} \hat{B}(\omega_k) e^{2\pi i \omega_k t} \quad (4.124)$$

for $t = 0, \pm 1, \pm 2, \dots, \pm(M/2-1)$. If we were to use (4.124) to define $\hat{\beta}_t$ for $|t| \geq M/2$, we would end up with a sequence of coefficients that is periodic with a period of M . In practice we define $\hat{\beta}_t = 0$ for $|t| \geq M/2$ instead. **Problem 4.32** explores the error resulting from this approximation.

Example 4.24 Lagged Regression for SOI and Recruitment

The high coherence between the SOI and Recruitment series noted in [Example 4.21](#) suggests a lagged regression relation between the two series. A natural direction for the implication in this situation is implied because we feel that the sea surface temperature or SOI should be the input and the Recruitment series should be the output. With this in mind, let x_t be the SOI series and y_t the Recruitment series.

Although we think naturally of the SOI as the input and the Recruitment as the output, two input-output configurations are of interest. With SOI as the input, the model is

$$y_t = \sum_{r=-\infty}^{\infty} a_r x_{t-r} + w_t$$

whereas a model that reverses the two roles would be

$$x_t = \sum_{r=-\infty}^{\infty} b_r y_{t-r} + v_t,$$

where w_t and v_t are white noise processes. Even though there is no plausible environmental explanation for the second of these two models, displaying both possibilities helps to settle on a parsimonious transfer function model.

Based on the script `LagReg` in `astsa`, the estimated regression or impulse response function for SOI, with $M = 32$ and $L = 15$ is

```
LagReg(soi, rec, L=15, M=32, threshold=6)
```

lag s	beta(s)
[1,]	5 -18.479306
[2,]	6 -12.263296
[3,]	7 -8.539368
[4,]	8 -6.984553

The prediction equation is

```
rec(t) = alpha + sum_s[ beta(s)*soi(t-s) ], where alpha = 65.97
```

```
MSE = 414.08
```

Note the negative peak at a lag of five points in the top of Figure 4.18; in this case, SOI is the input series. The fall-off after lag five seems to be approximately exponential and a possible model is

$$y_t = 66 - 18.5x_{t-5} - 12.3x_{t-6} - 8.5x_{t-7} - 7x_{t-8} + w_t.$$

If we examine the inverse relation, namely, a regression model with the Recruitment series y_t as the input, the bottom of Figure 4.18 implies a much simpler model,

```
LagReg(rec, soi, L=15, M=32, inverse=TRUE, threshold=.01)
```

lag s	beta(s)
[1,]	4 0.01593167
[2,]	5 -0.02120013

The prediction equation is

```
soi(t) = alpha + sum_s[ beta(s)*rec(t+s) ], where alpha = 0.41
```

```
MSE = 0.07
```

depending on only two coefficients, namely,

$$x_t = .41 + .016y_{t+4} - .02y_{t+5} + v_t.$$

Multiplying both sides by $50B^5$ and rearranging, we have

$$(1 - .8B)y_t = 20.5 - 50B^5x_t + \epsilon_t.$$

Finally, we check whether the noise, ϵ_t , is white. In addition, at this point, it simplifies matters if we rerun the regression with autocorrelated errors and reestimate the coefficients. The model is referred to as an ARMAX model (the X stands for exogenous; see Section 5.6 and Section 6.6.1):

```
fish = ts.intersect(R=rec, RL1=lag(rec,-1), SL5=lag(soi,-5))
(u = lm(fish[,1]~fish[,2:3], na.action=NULL))
acf2(resid(u)) # suggests ar1
sarima(fish[,1], 1, 0, 0, xreg=fish[,2:3]) # armax model
```

```
Coefficients:
ar1 intercept RL1 SL5
0.4487 12.3323 0.8005 -21.0307
s.e. 0.0503 1.5746 0.0234 1.0915
sigma^2 estimated as 49.93
```

Our final parsimonious fitted model is (with rounding)

$$y_t = 12 + .8y_{t-1} - 21x_{t-5} + \epsilon_t, \quad \text{and} \quad \epsilon_t = .45\epsilon_{t-1} + w_t,$$

where w_t is white noise with $\sigma_w^2 = 50$. This example is also examined in [Chapter 5](#) and the fitted values for the final model can be viewed [Figure 5.12](#).

The example shows we can get a clean estimator for the transfer functions relating the two series if the coherence $\hat{\rho}_{xy}^2(\omega)$ is large. The reason is that we can write the minimized mean squared error (4.118) as

$$MSE = E \left[\left(y_t - \sum_{r=-\infty}^{\infty} \beta_r x_{t-r} \right) y_t \right] = \gamma_{yy}(0) - \sum_{r=-\infty}^{\infty} \beta_r \gamma_{xy}(-r),$$

using the result about the orthogonality of the data and error term in the Projection theorem. Then, substituting the spectral representations of the autocovariance and cross-covariance functions and identifying the Fourier transform (4.121) in the result leads to

$$\begin{aligned} MSE &= \int_{-\frac{1}{2}}^{\frac{1}{2}} [f_{yy}(\omega) - B(\omega)f_{xy}(\omega)] d\omega \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} f_{yy}(\omega)[1 - \rho_{yx}^2(\omega)] d\omega, \end{aligned} \quad (4.125)$$

where $\rho_{yx}^2(\omega)$ is just the squared coherence given by (4.94). The similarity of (4.125) to the usual mean square error that results from predicting y from x is obvious. In that case, we would have

$$E(y - \beta x)^2 = \sigma_y^2(1 - \rho_{xy}^2)$$

for jointly distributed random variables x and y with zero means, variances σ_x^2 and σ_y^2 , and covariance $\sigma_{xy} = \rho_{xy}\sigma_x\sigma_y$. Because the mean squared error in (4.125) satisfies $MSE \geq 0$ with $f_{yy}(\omega)$ a non-negative function, it follows that the coherence satisfies

$$0 \leq \rho_{xy}^2(\omega) \leq 1$$

for all ω . Furthermore, [Problem 4.33](#) shows the squared coherence is one when the output are linearly related by the filter relation (4.116), and there is no noise, i.e., $v_t = 0$. Hence, the multiple coherence gives a measure of the association or correlation between the input and output series as a function of frequency.

The matter of verifying that the F -distribution claimed for (4.104) will hold when the sample coherence values are substituted for theoretical values still remains. Again,

the form of the F -statistic is exactly analogous to the usual t -test for no correlation in a regression context. We give an argument leading to this conclusion later using the results in [Section C.3](#). Another question that has not been resolved in this section is the extension to the case of multiple inputs $x_{t1}, x_{t2}, \dots, x_{tq}$. Often, more than just a single input series is present that can possibly form a lagged predictor of the output series y_t . An example is the cardiovascular mortality series that depended on possibly a number of pollution series and temperature. We discuss this particular extension as a part of the multivariate time series techniques considered in [Chapter 7](#).

4.9 Signal Extraction and Optimum Filtering

A model closely related to regression can be developed by assuming again that

$$y_t = \sum_{r=-\infty}^{\infty} \beta_r x_{t-r} + v_t, \quad (4.126)$$

but where the β s are known and x_t is some unknown random *signal* that is uncorrelated with the *noise* process v_t . In this case, we observe only y_t and are interested in an estimator for the signal x_t of the form

$$\hat{x}_t = \sum_{r=-\infty}^{\infty} a_r y_{t-r}. \quad (4.127)$$

In the frequency domain, it is convenient to make the additional assumptions that the series x_t and v_t are both mean-zero stationary series with spectra $f_{xx}(\omega)$ and $f_{vv}(\omega)$, often referred to as the *signal spectrum* and *noise spectrum*, respectively. Often, the special case $\beta_t = \delta_t$, in which δ_t is the Kronecker delta, is of interest because (4.126) reduces to the simple *signal plus noise* model

$$y_t = x_t + v_t \quad (4.128)$$

in that case. In general, we seek the set of filter coefficients a_t that minimize the mean squared error of estimation, say,

$$MSE = E \left[\left(x_t - \sum_{r=-\infty}^{\infty} a_r y_{t-r} \right)^2 \right]. \quad (4.129)$$

This problem was originally solved by Kolmogorov (1941) and by Wiener (1949), who derived the result in 1941 and published it in classified reports during World War II.

We can apply the orthogonality principle to write

$$E \left[\left(x_t - \sum_{r=-\infty}^{\infty} a_r y_{t-r} \right) y_{t-s} \right] = 0$$

for $s = 0, \pm 1, \pm 2, \dots$, which leads to

$$\sum_{r=-\infty}^{\infty} a_r \gamma_{yy}(s-r) = \gamma_{xy}(s),$$

to be solved for the filter coefficients. Substituting the spectral representations for the autocovariance functions into the above and identifying the spectral densities through the uniqueness of the Fourier transform produces

$$A(\omega) f_{yy}(\omega) = f_{xy}(\omega), \quad (4.130)$$

where $A(\omega)$ and the optimal filter a_t are Fourier transform pairs for $B(\omega)$ and β_t . Now, a special consequence of the model is that (see [Problem 4.30](#))

$$f_{xy}(\omega) = B^*(\omega) f_{xx}(\omega) \quad (4.131)$$

and

$$f_{yy}(\omega) = |B(\omega)|^2 f_{xx}(\omega) + f_{vv}(\omega), \quad (4.132)$$

implying the optimal filter would be Fourier transform of

$$A(\omega) = \frac{B^*(\omega)}{\left(|B(\omega)|^2 + \frac{f_{vv}(\omega)}{f_{xx}(\omega)} \right)}, \quad (4.133)$$

where the second term in the denominator is just the inverse of the *signal to noise ratio*, say,

$$\text{SNR}(\omega) = \frac{f_{xx}(\omega)}{f_{vv}(\omega)}. \quad (4.134)$$

The result shows the optimum filters can be computed for this model if the signal and noise spectra are both known or if we can assume knowledge of the signal-to-noise ratio $\text{SNR}(\omega)$ as function of frequency. In [Chapter 7](#), we show some methods for estimating these two parameters in conjunction with random effects analysis of variance models, but we assume here that it is possible to specify the signal-to-noise ratio *a priori*. If the signal-to-noise ratio is known, the optimal filter can be computed by the inverse transform of the function $A(\omega)$. It is more likely that the inverse transform will be intractable and a finite filter approximation like that used in the previous section can be applied to the data. In this case, we will have

$$a_t^M = M^{-1} \sum_{k=0}^{M-1} A(\omega_k) e^{2\pi i \omega_k t} \quad (4.135)$$

as the estimated filter function. It will often be the case that the form of the specified frequency response will have some rather sharp transitions between regions where the signal-to-noise ratio is high and regions where there is little signal. In these cases, the shape of the frequency response function will have ripples that can introduce

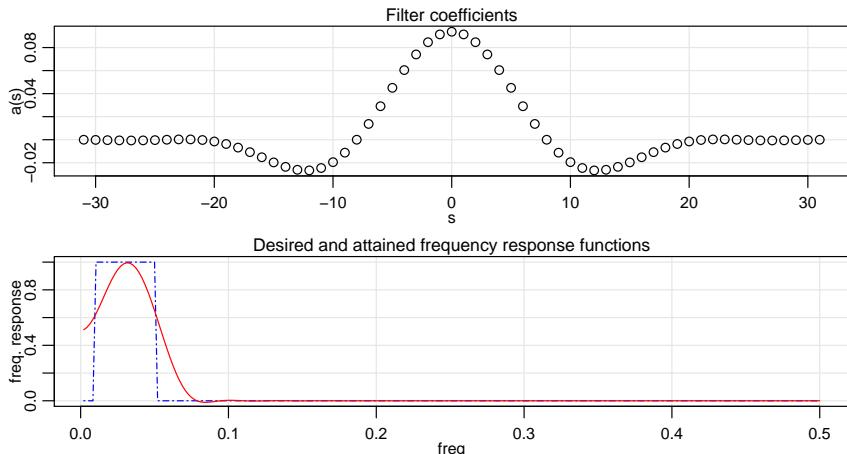


Fig. 4.19. Filter coefficients (top) and frequency response functions (bottom) for designed SOI filters.

frequencies at different amplitudes. An aesthetic solution to this problem is to introduce tapering as was done with spectral estimation in (4.69)–(4.76). We use below the tapered filter $\tilde{a}_t = h_t a_t$ where h_t is the cosine taper given in (4.76). The squared frequency response of the resulting filter will be $|\tilde{A}(\omega)|^2$, where

$$\tilde{A}(\omega) = \sum_{t=-\infty}^{\infty} a_t h_t e^{-2\pi i \omega t}. \quad (4.136)$$

The results are illustrated in the following example that extracts the El Niño component of the sea surface temperature series.

Example 4.25 Estimating the El Niño Signal via Optimal Filters

Figure 4.7 shows the spectrum of the SOI series, and we note that essentially two components have power, the El Niño frequency of about .02 cycles per month (the four-year cycle) and a yearly frequency of about .08 cycles per month (the annual cycle). We assume, for this example, that we wish to preserve the lower frequency as signal and to eliminate the higher order frequencies, and in particular, the annual cycle. In this case, we assume the simple signal plus noise model

$$y_t = x_t + v_t,$$

so that there is no convolving function β_t . Furthermore, the signal-to-noise ratio is assumed to be high to about .06 cycles per month and zero thereafter. The optimal frequency response was assumed to be unity to .05 cycles per point and then to decay linearly to zero in several steps. Figure 4.19 shows the coefficients as specified by (4.135) with $M = 64$, as well as the frequency response function given by (4.136), of the cosine tapered coefficients; recall Figure 4.11, where we demonstrated the

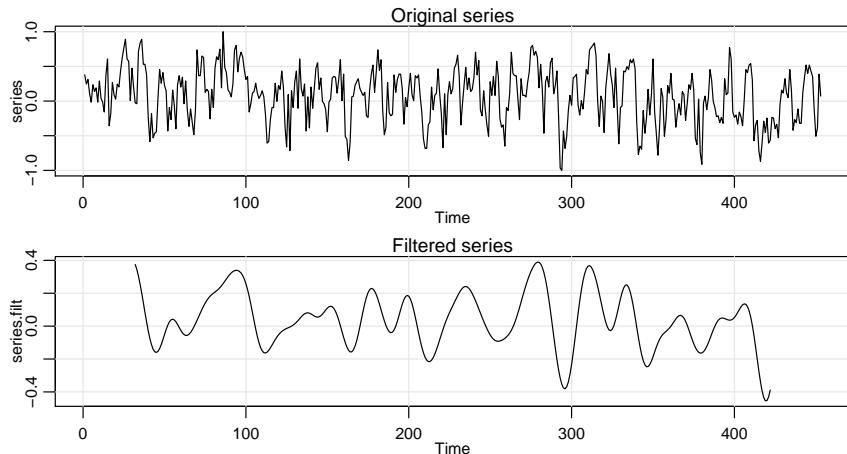


Fig. 4.20. Original SOI series (top) compared to filtered version showing the estimated El Niño temperature signal (bottom).

need for tapering to avoid severe ripples in the window. The constructed response function is compared to the ideal window in [Figure 4.19](#).

[Figure 4.20](#) shows the original and filtered SOI index, and we see a smooth extracted signal that conveys the essence of the underlying El Niño signal. The frequency response of the designed filter can be compared with that of the symmetric 12-month moving average applied to the same series in [Example 4.22](#). The filtered series, shown in [Figure 4.16](#), shows a good deal of higher frequency chatter riding on the smoothed version, which has been introduced by the higher frequencies that leak through in the squared frequency response, as in [Figure 4.17](#).

The analysis can be replicated using the script [SigExtract](#).

```
SigExtract(soi, L=9, M=64, max.freq=.05)
```

The design of finite filters with a specified frequency response requires some experimentation with various target frequency response functions and we have only touched on the methodology here. The filter designed here, sometimes called a low-pass filter reduces the high frequencies and keeps or passes the low frequencies. Alternately, we could design a high-pass filter to keep high frequencies if that is where the signal is located. An example of a simple high-pass filter is the first difference with a frequency response that is shown in [Figure 4.17](#). We can also design band-pass filters that keep frequencies in specified bands. For example, seasonal adjustment filters are often used in economics to reject seasonal frequencies while keeping both high frequencies, lower frequencies, and trend (see, for example, Grether and Nerlove, 1970).

The filters we have discussed here are all symmetric two-sided filters, because the designed frequency response functions were purely real. Alternatively, we may design recursive filters to produce a desired response. An example of a recursive filter is one that replaces the input x_t by the filtered output

$$y_t = \sum_{k=1}^p \phi_k y_{t-k} + x_t - \sum_{k=1}^q \theta_k x_{t-k}. \quad (4.137)$$

Note the similarity between (4.137) and the ARMA(p, q) model, in which the white noise component is replaced by the input. Transposing the terms involving y_t and using the basic linear filter result in [Property 4.3](#) leads to

$$f_y(\omega) = \frac{|\theta(e^{-2\pi i \omega})|^2}{|\phi(e^{-2\pi i \omega})|^2} f_x(\omega), \quad (4.138)$$

where

$$\phi(e^{-2\pi i \omega}) = 1 - \sum_{k=1}^p \phi_k e^{-2\pi i k \omega}$$

and

$$\theta(e^{-2\pi i \omega}) = 1 - \sum_{k=1}^q \theta_k e^{-2\pi i k \omega}.$$

Recursive filters such as those given by (4.138) distort the phases of arriving frequencies, and we do not consider the problem of designing such filters in any detail.

4.10 Spectral Analysis of Multidimensional Series

Multidimensional series of the form x_s , where $s = (s_1, s_2, \dots, s_r)'$ is an r -dimensional vector of spatial coordinates or a combination of space and time coordinates, were introduced in [Section 1.6](#). The example given there, shown in [Figure 1.18](#), was a collection of temperature measurements taking on a rectangular field. These data would form a two-dimensional process, indexed by row and column in space. In that section, the multidimensional autocovariance function of an r -dimensional stationary series was given as $\gamma_x(h) = E[x_{s+h}x_s]$, where the multidimensional lag vector is $h = (h_1, h_2, \dots, h_r)'$.

The multidimensional *wavenumber spectrum* is given as the Fourier transform of the autocovariance, namely,

$$f_x(\omega) = \sum_h \cdots \sum_h \gamma_x(h) e^{-2\pi i \omega' h}. \quad (4.139)$$

Again, the inverse result

$$\gamma_x(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \cdots \int_{-\frac{1}{2}}^{\frac{1}{2}} f_x(\omega) e^{2\pi i \omega' h} d\omega \quad (4.140)$$

holds, where the integral is over the multidimensional range of the vector ω . The wavenumber argument is exactly analogous to the frequency argument, and we have the corresponding intuitive interpretation as the cycling rate ω_i per distance traveled s_i in the i -th direction.

Two-dimensional processes occur often in practical applications, and the representations above reduce to

$$f_x(\omega_1, \omega_2) = \sum_{h_1=-\infty}^{\infty} \sum_{h_2=-\infty}^{\infty} \gamma_x(h_1, h_2) e^{-2\pi i(\omega_1 h_1 + \omega_2 h_2)} \quad (4.141)$$

and

$$\gamma_x(h_1, h_2) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} f_x(\omega_1, \omega_2) e^{2\pi i(\omega_1 h_1 + \omega_2 h_2)} d\omega_1 d\omega_2 \quad (4.142)$$

in the case $r = 2$. The notion of linear filtering generalizes easily to the two-dimensional case by defining the impulse response function a_{s_1, s_2} and the spatial filter output as

$$y_{s_1, s_2} = \sum_{u_1} \sum_{u_2} a_{u_1, u_2} x_{s_1 - u_1, s_2 - u_2}. \quad (4.143)$$

The spectrum of the output of this filter can be derived as

$$f_y(\omega_1, \omega_2) = |A(\omega_1, \omega_2)|^2 f_x(\omega_1, \omega_2), \quad (4.144)$$

where

$$A(\omega_1, \omega_2) = \sum_{u_1} \sum_{u_2} a_{u_1, u_2} e^{-2\pi i(\omega_1 u_1 + \omega_2 u_2)}. \quad (4.145)$$

These results are analogous to those in the one-dimensional case, described by [Property 4.3](#).

The multidimensional DFT is also a straightforward generalization of the univariate expression. In the two-dimensional case with data on a rectangular grid, $\{x_{s_1, s_2}; s_1 = 1, \dots, n_1, s_2 = 1, \dots, n_2\}$, we will write, for $-1/2 \leq \omega_1, \omega_2 \leq 1/2$,

$$d(\omega_1, \omega_2) = (n_1 n_2)^{-1/2} \sum_{s_1=1}^{n_1} \sum_{s_2=1}^{n_2} x_{s_1, s_2} e^{-2\pi i(\omega_1 s_1 + \omega_2 s_2)} \quad (4.146)$$

as the two-dimensional DFT, where the frequencies ω_1, ω_2 are evaluated at multiples of $(1/n_1, 1/n_2)$ on the spatial frequency scale. The two-dimensional wavenumber spectrum can be estimated by the smoothed *sample wavenumber spectrum*

$$\tilde{f}_x(\omega_1, \omega_2) = (L_1 L_2)^{-1} \sum_{\ell_1, \ell_2} |d(\omega_1 + \ell_1/n_1, \omega_2 + \ell_2/n_2)|^2, \quad (4.147)$$

where the sum is taken over the grid $\{-m_j \leq \ell_j \leq m_j; j = 1, 2\}$, where $L_1 = 2m_1 + 1$ and $L_2 = 2m_2 + 1$. The statistic

$$\frac{2L_1 L_2 \tilde{f}_x(\omega_1, \omega_2)}{f_x(\omega_1, \omega_2)} \sim \chi^2_{2L_1 L_2} \quad (4.148)$$

can be used to set confidence intervals or make approximate tests against a fixed assumed spectrum $f_0(\omega_1, \omega_2)$.

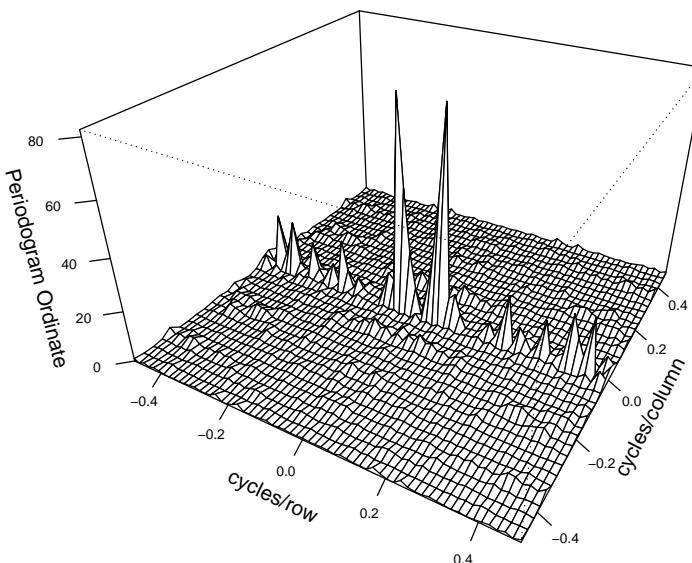


Fig. 4.21. Two-dimensional periodogram of soil temperature profile showing peak at .0625 cycles/row. The period is 16 rows, and this corresponds to $16 \times 17 \text{ ft} = 272 \text{ ft}$.

Example 4.26 Soil Surface Temperatures

As an example, consider the periodogram of the two-dimensional temperature series shown in Figure 1.18 and analyzed by Bazza et al. (1988). We recall the spatial coordinates in this case will be (s_1, s_2) , which define the spatial coordinates rows and columns so that the frequencies in the two directions will be expressed as cycles per row and cycles per column. Figure 4.21 shows the periodogram of the two-dimensional temperature series, and we note the ridge of strong spectral peaks running over rows at a column frequency of zero. An obvious periodic component appears at frequencies of .0625 and -.0625 cycles per row, which corresponds to 16 rows or about 272 ft. On further investigation of previous irrigation patterns over this field, treatment levels of salt varied periodically over columns. This analysis is extended in Problem 4.24, where we recover the salt treatment profile over rows and compare it to a signal, computed by averaging over columns.

Figure 4.21 may be reproduced in R as follows. In the code for this example, the periodogram is computed in one step as `per`; the rest of the code is simply manipulation to obtain a nice graphic.

```

per = Mod(fft(soiltemp-mean(soiltemp))/sqrt(64*36))^2
per2 = cbind(per[1:32,18:2], per[1:32,1:18])
per3 = rbind(per2[32:2,],per2)
par(mar=c(1,2.5,0,0)+.1)
persp(-31:31/64, -17:17/36, per3, phi=30, theta=30, expand=.6,
      ticktype="detailed", xlab="cycles/row", ylab="cycles/column",
      zlab="Periodogram Ordinate")

```

Another application of two-dimensional spectral analysis of agricultural field trials is given in McBratney and Webster (1981), who used it to detect ridge and furrow patterns in yields. The requirement for regular, equally spaced samples on fairly large grids has tended to limit enthusiasm for strict two-dimensional spectral analysis. An exception is when a propagating signal from a given velocity and azimuth is present so predicting the wavenumber spectrum as a function of velocity and azimuth becomes feasible (see Shumway et al., 1999).

Problems

Section 4.1

4.1 Verify that for any positive integer n and $j, k = 0, 1, \dots, \lfloor n/2 \rfloor$, where $\lfloor \cdot \rfloor$ denotes the greatest integer function:

- (a) Except for $j = 0$ or $j = n/2$,^{4.12}

$$\sum_{t=1}^n \cos^2(2\pi t j/n) = \sum_{t=1}^n \sin^2(2\pi t j/n) = n/2.$$

- (b) When $j = 0$ or $j = n/2$,

$$\sum_{t=1}^n \cos^2(2\pi t j/n) = n \text{ but } \sum_{t=1}^n \sin^2(2\pi t j/n) = 0.$$

- (c) For $j \neq k$,

$$\sum_{t=1}^n \cos(2\pi t j/n) \cos(2\pi t k/n) = \sum_{t=1}^n \sin(2\pi t j/n) \sin(2\pi t k/n) = 0.$$

Also, for any j and k ,

$$\sum_{t=1}^n \cos(2\pi t j/n) \sin(2\pi t k/n) = 0.$$

^{4.12} Hint: We'll do part of the problem.

$$\begin{aligned} \sum_{t=1}^n \cos^2(2\pi t j/n) &= \frac{1}{4} \sum_{t=1}^n (e^{2\pi i t j/n} + e^{-2\pi i t j/n})(e^{2\pi i t j/n} + e^{-2\pi i t j/n}) \\ &= \frac{1}{4} \sum_{t=1}^n (e^{4\pi i t j/n} + 1 + 1 + e^{-4\pi i t j/n}) = \frac{n}{2}. \end{aligned}$$

4.2 Repeat the simulations and analyses in [Example 4.1](#) and [Example 4.2](#) with the following changes:

- (a) Change the sample size to $n = 128$ and generate and plot the same series as in [Example 4.1](#):

$$\begin{aligned}x_{t1} &= 2 \cos(2\pi .06 t) + 3 \sin(2\pi .06 t), \\x_{t2} &= 4 \cos(2\pi .10 t) + 5 \sin(2\pi .10 t), \\x_{t3} &= 6 \cos(2\pi .40 t) + 7 \sin(2\pi .40 t), \\x_t &= x_{t1} + x_{t2} + x_{t3}.\end{aligned}$$

What is the major difference between these series and the series generated in [Example 4.1](#)? (Hint: The answer is *fundamental*. But if your answer is the series are longer, you may be punished severely.)

- (b) As in [Example 4.2](#), compute and plot the periodogram of the series, x_t , generated in (a) and comment.
(c) Repeat the analyses of (a) and (b) but with $n = 100$ (as in [Example 4.1](#)), and adding noise to x_t ; that is

$$x_t = x_{t1} + x_{t2} + x_{t3} + w_t$$

where $w_t \sim \text{iid } N(0, 25)$. That is, you should simulate and plot the data, and then plot the periodogram of x_t and comment.

4.3 With reference to equations [\(4.1\)](#) and [\(4.2\)](#), let $Z_1 = U_1$ and $Z_2 = -U_2$ be independent, standard normal variables. Consider the polar coordinates of the point (Z_1, Z_2) , that is,

$$A^2 = Z_1^2 + Z_2^2 \quad \text{and} \quad \phi = \tan^{-1}(Z_2/Z_1).$$

- (a) Find the joint density of A^2 and ϕ , and from the result, conclude that A^2 and ϕ are independent random variables, where A^2 is a chi-squared random variable with 2 df, and ϕ is uniformly distributed on $(-\pi, \pi)$.
(b) Going in reverse from polar coordinates to rectangular coordinates, suppose we assume that A^2 and ϕ are independent random variables, where A^2 is chi-squared with 2 df, and ϕ is uniformly distributed on $(-\pi, \pi)$. With $Z_1 = A \cos(\phi)$ and $Z_2 = A \sin(\phi)$, where A is the positive square root of A^2 , show that Z_1 and Z_2 are independent, standard normal random variables.

4.4 Verify [\(4.5\)](#).

Section 4.2

4.5 A time series was generated by first drawing the white noise series w_t from a normal distribution with mean zero and variance one. The observed series x_t was generated from

$$x_t = w_t - \theta w_{t-1}, \quad t = 0, \pm 1, \pm 2, \dots,$$

where θ is a parameter.

- (a) Derive the theoretical mean value and autocovariance functions for the series x_t and w_t . Are the series x_t and w_t stationary? Give your reasons.
 (b) Give a formula for the power spectrum of x_t , expressed in terms of θ and ω .

4.6 A first-order autoregressive model is generated from the white noise series w_t using the generating equations

$$x_t = \phi x_{t-1} + w_t,$$

where ϕ , for $|\phi| < 1$, is a parameter and the w_t are independent random variables with mean zero and variance σ_w^2 .

- (a) Show that the power spectrum of x_t is given by

$$f_x(\omega) = \frac{\sigma_w^2}{1 + \phi^2 - 2\phi \cos(2\pi\omega)}.$$

- (b) Verify the autocovariance function of this process is

$$\gamma_x(h) = \frac{\sigma_w^2 \phi^{|h|}}{1 - \phi^2},$$

$h = 0, \pm 1, \pm 2, \dots$, by showing that the inverse transform of $\gamma_x(h)$ is the spectrum derived in part (a).

4.7 In applications, we will often observe series containing a signal that has been delayed by some unknown time D , i.e.,

$$x_t = s_t + As_{t-D} + n_t,$$

where s_t and n_t are stationary and independent with zero means and spectral densities $f_s(\omega)$ and $f_n(\omega)$, respectively. The delayed signal is multiplied by some unknown constant A . Show that

$$f_x(\omega) = [1 + A^2 + 2A \cos(2\pi\omega D)]f_s(\omega) + f_n(\omega).$$

4.8 Suppose x_t and y_t are stationary zero-mean time series with x_t independent of y_s for all s and t . Consider the product series

$$z_t = x_t y_t.$$

Prove the spectral density for z_t can be written as

$$f_z(\omega) = \int_{-\frac{1}{2}}^{\frac{1}{2}} f_x(\omega - \nu) f_y(\nu) d\nu.$$

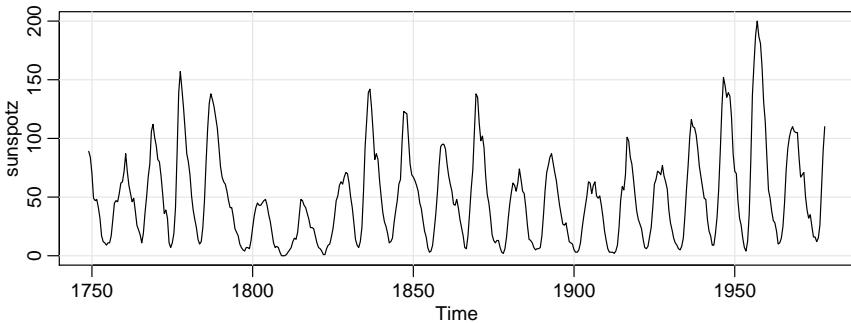


Fig. 4.22. Smoothed 12-month sunspot numbers (`sunspotz`) sampled twice per year.

Section 4.3

4.9 Figure 4.22 shows the biyearly smoothed (12-month moving average) number of sunspots from June 1749 to December 1978 with $n = 459$ points that were taken twice per year; the data are contained in `sunspotz`. With Example 4.13 as a guide, perform a periodogram analysis identifying the predominant periods and obtaining confidence intervals for the identified periods. Interpret your findings.

4.10 The levels of salt concentration known to have occurred over rows, corresponding to the average temperature levels for the soil science data considered in Figure 1.18 and Figure 1.19, are in `salt` and `saltemp`. Plot the series and then identify the dominant frequencies by performing separate spectral analyses on the two series. Include confidence intervals for the dominant frequencies and interpret your findings.

4.11 Let the observed series x_t be composed of a periodic signal and noise so it can be written as

$$x_t = \beta_1 \cos(2\pi\omega_k t) + \beta_2 \sin(2\pi\omega_k t) + w_t,$$

where w_t is a white noise process with variance σ_w^2 . The frequency ω_k is assumed to be known and of the form k/n in this problem. Suppose we consider estimating β_1 , β_2 and σ_w^2 by least squares, or equivalently, by maximum likelihood if the w_t are assumed to be Gaussian.

(a) Prove, for a fixed ω_k , the minimum squared error is attained by

$$\begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix} = 2n^{-1/2} \begin{pmatrix} d_c(\omega_k) \\ d_s(\omega_k) \end{pmatrix},$$

where the cosine and sine transforms (4.31) and (4.32) appear on the right-hand side.

(b) Prove that the error sum of squares can be written as

$$\text{SSE} = \sum_{t=1}^n x_t^2 - 2I_x(\omega_k)$$

so that the value of ω_k that minimizes squared error is the same as the value that maximizes the periodogram $I_x(\omega_k)$ estimator (4.28).

- (c) Under the Gaussian assumption and fixed ω_k , show that the F -test of no regression leads to an F -statistic that is a monotone function of $I_x(\omega_k)$.

4.12 Prove the convolution property of the DFT, namely,

$$\sum_{s=1}^n a_s x_{t-s} = \sum_{k=0}^{n-1} d_A(\omega_k) d_x(\omega_k) \exp\{2\pi\omega_k t\},$$

for $t = 1, 2, \dots, n$, where $d_A(\omega_k)$ and $d_x(\omega_k)$ are the discrete Fourier transforms of a_t and x_t , respectively, and we assume that $x_t = x_{t+n}$ is periodic.

Section 4.4

4.13 Analyze the chicken price data ([chicken](#)) using a nonparametric spectral estimation procedure. Aside from the obvious annual cycle discovered in [Example 2.5](#), what other interesting cycles are revealed?

4.14 Repeat [Problem 4.9](#) using a nonparametric spectral estimation procedure. In addition to discussing your findings in detail, comment on your choice of a spectral estimate with regard to smoothing and tapering.

4.15 Repeat [Problem 4.10](#) using a nonparametric spectral estimation procedure. In addition to discussing your findings in detail, comment on your choice of a spectral estimate with regard to smoothing and tapering.

4.16 Cepstral Analysis. The periodic behavior of a time series induced by echoes can also be observed in the spectrum of the series; this fact can be seen from the results stated in [Problem 4.7](#). Using the notation of that problem, suppose we observe $x_t = s_t + As_{t-D} + n_t$, which implies the spectra satisfy $f_x(\omega) = [1 + A^2 + 2A \cos(2\pi\omega D)]f_s(\omega) + f_n(\omega)$. If the noise is negligible ($f_n(\omega) \approx 0$) then $\log f_x(\omega)$ is approximately the sum of a periodic component, $\log[1 + A^2 + 2A \cos(2\pi\omega D)]$, and $\log f_s(\omega)$. Bogart et al. (1962) proposed treating the detrended log spectrum as a pseudo time series and calculating its spectrum, or *cepstrum*, which should show a peak at a *quefrency* corresponding to $1/D$. The cepstrum can be plotted as a function of quefrency, from which the delay D can be estimated.

For the speech series presented in [Example 1.3](#), estimate the pitch period using cepstral analysis as follows. The data are in [speech](#).

- (a) Calculate and display the log-periodogram of the data. Is the periodogram periodic, as predicted?
- (b) Perform a cepstral (spectral) analysis on the detrended logged periodogram, and use the results to estimate the delay D . How does your answer compare with the analysis of [Example 1.27](#), which was based on the ACF?

4.17 Use Property 4.2 to verify (4.71). Then verify (4.74) and (4.75).

4.18 Consider two time series

$$x_t = w_t - w_{t-1},$$

$$y_t = \frac{1}{2}(w_t + w_{t-1}),$$

formed from the white noise series w_t with variance $\sigma_w^2 = 1$.

- (a) Are x_t and y_t jointly stationary? Recall the cross-covariance function must also be a function only of the lag h and cannot depend on time.
- (b) Compute the spectra $f_y(\omega)$ and $f_x(\omega)$, and comment on the difference between the two results.
- (c) Suppose sample spectral estimators $\bar{f}_y(.10)$ are computed for the series using $L = 3$. Find a and b such that

$$P\left\{a \leq \bar{f}_y(.10) \leq b\right\} = .90.$$

This expression gives two points that will contain 90% of the sample spectral values. Put 5% of the area in each tail.

Section 4.5

4.19 Often, the periodicities in the sunspot series are investigated by fitting an autoregressive spectrum of sufficiently high order. The main periodicity is often stated to be in the neighborhood of 11 years. Fit an autoregressive spectral estimator to the sunspot data using a model selection method of your choice. Compare the result with a conventional nonparametric spectral estimator found in Problem 4.9.

4.20 Analyze the chicken price data ([chicken](#)) using a parametric spectral estimation procedure. Compare the results to Problem 4.13.

4.21 Fit an autoregressive spectral estimator to the Recruitment series and compare it to the results of Example 4.16.

4.22 Suppose a sample time series with $n = 256$ points is available from the first-order autoregressive model. Furthermore, suppose a sample spectrum computed with $L = 3$ yields the estimated value $\bar{f}_x(1/8) = 2.25$. Is this sample value consistent with $\sigma_w^2 = 1, \phi = .5$? Repeat using $L = 11$ if we just happen to obtain the same sample value.

4.23 Suppose we wish to test the noise alone hypothesis $H_0 : x_t = n_t$ against the signal-plus-noise hypothesis $H_1 : x_t = s_t + n_t$, where s_t and n_t are uncorrelated zero-mean stationary processes with spectra $f_s(\omega)$ and $f_n(\omega)$. Suppose that we want the test over a band of $L = 2m+1$ frequencies of the form $\omega_{j:n} + k/n$, for $k = 0, \pm 1, \pm 2, \dots, \pm m$ near some fixed frequency ω . Assume that both the signal and noise spectra are approximately constant over the interval.

- (a) Prove the approximate likelihood-based test statistic for testing H_0 against H_1 is proportional to

$$T = \sum_k |d_x(\omega_{j:n} + k/n)|^2 \left(\frac{1}{f_n(\omega)} - \frac{1}{f_s(\omega) + f_n(\omega)} \right).$$

- (b) Find the approximate distributions of T under H_0 and H_1 .
(c) Define the false alarm and signal detection probabilities as $P_F = P\{T > K|H_0\}$ and $P_d = P\{T > k|H_1\}$, respectively. Express these probabilities in terms of the signal-to-noise ratio $f_s(\omega)/f_n(\omega)$ and appropriate chi-squared integrals.

Section 4.6

4.24 Analyze the coherency between the temperature and salt data discussed in [Problem 4.10](#). Discuss your findings.

4.25 Consider two processes

$$x_t = w_t \quad \text{and} \quad y_t = \phi x_{t-D} + v_t$$

where w_t and v_t are independent white noise processes with common variance σ^2 , ϕ is a constant, and D is a fixed integer delay.

- (a) Compute the coherency between x_t and y_t .
(b) Simulate $n = 1024$ normal observations from x_t and y_t for $\phi = .9$, $\sigma^2 = 1$, and $D = 0$. Then estimate and plot the coherency between the simulated series for the following values of L and comment:
(i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, and (iv) $L = 101$.

Section 4.7

4.26 For the processes in [Problem 4.25](#):

- (a) Compute the phase between x_t and y_t .
(b) Simulate $n = 1024$ observations from x_t and y_t for $\phi = .9$, $\sigma^2 = 1$, and $D = 1$. Then estimate and plot the phase between the simulated series for the following values of L and comment:
(i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, and (iv) $L = 101$.

4.27 Consider the bivariate time series records containing monthly U.S. production ([prod](#)) as measured by the Federal Reserve Board Production Index and the monthly unemployment series ([unemp](#)).

- (a) Compute the spectrum and the log spectrum for each series, and identify statistically significant peaks. Explain what might be generating the peaks. Compute the coherence, and explain what is meant when a high coherence is observed at a particular frequency.

- (b) What would be the effect of applying the filter

$$u_t = x_t - x_{t-1} \quad \text{followed by} \quad v_t = u_t - u_{t-12}$$

to the series given above? Plot the predicted frequency responses of the simple difference filter and of the seasonal difference of the first difference.

- (c) Apply the filters successively to one of the two series and plot the output. Examine the output after taking a first difference and comment on whether stationarity is a reasonable assumption. Why or why not? Plot after taking the seasonal difference of the first difference. What can be noticed about the output that is consistent with what you have predicted from the frequency response? Verify by computing the spectrum of the output after filtering.

- 4.28** Determine the theoretical power spectrum of the series formed by combining the white noise series w_t to form

$$y_t = w_{t-2} + 4w_{t-1} + 6w_t + 4w_{t+1} + w_{t+2}.$$

Determine which frequencies are present by plotting the power spectrum.

- 4.29** Let $x_t = \cos(2\pi\omega t)$, and consider the output

$$y_t = \sum_{k=-\infty}^{\infty} a_k x_{t-k},$$

where $\sum_k |a_k| < \infty$. Show

$$y_t = |A(\omega)| \cos(2\pi\omega t + \phi(\omega)),$$

where $|A(\omega)|$ and $\phi(\omega)$ are the amplitude and phase of the filter, respectively. Interpret the result in terms of the relationship between the input series, x_t , and the output series, y_t .

- 4.30** Suppose x_t is a stationary series, and we apply two filtering operations in succession, say,

$$y_t = \sum_r a_r x_{t-r} \quad \text{then} \quad z_t = \sum_s b_s y_{t-s}.$$

- (a) Show the spectrum of the output is

$$f_z(\omega) = |A(\omega)|^2 |B(\omega)|^2 f_x(\omega),$$

where $A(\omega)$ and $B(\omega)$ are the Fourier transforms of the filter sequences a_t and b_t , respectively.

- (b) What would be the effect of applying the filter

$$u_t = x_t - x_{t-1} \quad \text{followed by} \quad v_t = u_t - u_{t-12}$$

to a time series?

- (c) Plot the predicted frequency responses of the simple difference filter and of the seasonal difference of the first difference. Filters like these are called seasonal adjustment filters in economics because they tend to attenuate frequencies at multiples of the monthly periods. The difference filter tends to attenuate low-frequency trends.

4.31 Suppose we are given a stationary zero-mean series x_t with spectrum $f_x(\omega)$ and then construct the derived series

$$y_t = ay_{t-1} + x_t, \quad t = \pm 1, \pm 2, \dots.$$

- (a) Show how the theoretical $f_y(\omega)$ is related to $f_x(\omega)$.
 (b) Plot the function that multiplies $f_x(\omega)$ in part (a) for $a = .1$ and for $a = .8$. This filter is called a recursive filter.

Section 4.8

4.32 Consider the problem of approximating the filter output

$$y_t = \sum_{k=-\infty}^{\infty} a_k x_{t-k}, \quad \sum_{-\infty}^{\infty} |a_k| < \infty,$$

by

$$y_t^M = \sum_{|k| < M/2} a_k^M x_{t-k}$$

for $t = M/2 - 1, M/2, \dots, n - M/2$, where x_t is available for $t = 1, \dots, n$ and

$$a_t^M = M^{-1} \sum_{k=0}^{M-1} A(\omega_k) \exp\{2\pi i \omega_k t\}$$

with $\omega_k = k/M$. Prove

$$E\{(y_t - y_t^M)^2\} \leq 4\gamma_x(0) \left(\sum_{|k| \geq M/2} |a_k| \right)^2.$$

4.33 Prove the squared coherence $\rho_{y,x}^2(\omega) = 1$ for all ω when

$$y_t = \sum_{r=-\infty}^{\infty} a_r x_{t-r},$$

that is, when x_t and y_t can be related exactly by a linear filter.

4.34 The data set `climhyd`, contains 454 months of measured values for six climatic variables: (i) air temperature [`Temp`], (ii) dew point [`DewPt`], (iii) cloud cover [`CldCvr`], (iv) wind speed [`WndSpd`], (v) precipitation [`Precip`], and (vi) inflow [`Inflow`], at Lake Shasta in California; the data are displayed in Figure 7.3. We would like to look at possible relations among the weather factors and between the weather factors and the inflow to Lake Shasta.

- (a) First transform the inflow and precipitation series as follows: $I_t = \log i_t$, where i_t is inflow, and $P_t = \sqrt{p_t}$, where p_t is precipitation. Then, compute the square coherencies between all the weather variables and transformed inflow and argue that the strongest determinant of the inflow series is (transformed) precipitation. [Tip: If \mathbf{x} contains multiple time series, then the easiest way to display all the squared coherencies is to plot the coherencies suppressing the confidence intervals, e.g., `mvspec(x, spans=c(7,7), taper=.5, plot.type="coh", ci=-1)`.

- (b) Fit a lagged regression model of the form

$$I_t = \beta_0 + \sum_{j=0}^{\infty} \beta_j P_{t-j} + w_t,$$

using thresholding, and then comment of the predictive ability of precipitation for inflow.

Section 4.9

- 4.35** Consider the *signal plus noise* model

$$y_t = \sum_{r=-\infty}^{\infty} \beta_r x_{t-r} + v_t,$$

where the signal and noise series, x_t and v_t are both stationary with spectra $f_x(\omega)$ and $f_v(\omega)$, respectively. Assuming that x_t and v_t are independent of each other for all t , verify (4.131) and (4.132).

- 4.36** Consider the model

$$y_t = x_t + v_t,$$

where

$$x_t = \phi x_{t-1} + w_t,$$

such that v_t is Gaussian white noise and independent of x_t with $\text{var}(v_t) = \sigma_v^2$, and w_t is Gaussian white noise and independent of v_t , with $\text{var}(w_t) = \sigma_w^2$, and $|\phi| < 1$ and $\text{Ex}_0 = 0$. Prove that the spectrum of the observed series y_t is

$$f_y(\omega) = \sigma^2 \frac{|1 - \theta e^{-2\pi i \omega}|^2}{|1 - \phi e^{-2\pi i \omega}|^2},$$

where

$$\theta = \frac{c \pm \sqrt{c^2 - 4}}{2}, \quad \sigma^2 = \frac{\sigma_v^2 \phi}{\theta},$$

and

$$c = \frac{\sigma_w^2 + \sigma_v^2(1 + \phi^2)}{\sigma_v^2 \phi}.$$

- 4.37** Consider the same model as in the preceding problem.

(a) Prove the optimal smoothed estimator of the form

$$\hat{x}_t = \sum_{s=-\infty}^{\infty} a_s y_{t-s}$$

has

$$a_s = \frac{\sigma_v^2}{\sigma^2} \frac{\theta^{|s|}}{1 - \theta^2}.$$

(b) Show the mean square error is given by

$$E\{(x_t - \hat{x}_t)^2\} = \frac{\sigma_v^2 \sigma_w^2}{\sigma^2 (1 - \theta^2)}.$$

(c) Compare mean square error of the estimator in part (b) with that of the optimal finite estimator of the form

$$\hat{x}_t = a_1 y_{t-1} + a_2 y_{t-2}$$

when $\sigma_v^2 = .053$, $\sigma_w^2 = .172$, and $\phi_1 = .9$.

Section 4.10

4.38 Consider the two-dimensional linear filter given as the output (4.143).

- (a) Express the two-dimensional autocovariance function of the output, say, $\gamma_y(h_1, h_2)$, in terms of an infinite sum involving the autocovariance function of x_s and the filter coefficients a_{s_1, s_2} .
- (b) Use the expression derived in (a), combined with (4.142) and (4.145) to derive the spectrum of the filtered output (4.144).

The following problems require supplemental material from Appendix C

4.39 Let w_t be a Gaussian white noise series with variance σ_w^2 . Prove that the results of **Theorem C.4** hold without error for the DFT of w_t .

4.40 Show that condition (4.48) implies (C.19) by showing

$$n^{-1/2} \sum_{h \geq 0} h |\gamma(h)| \leq \sigma_w^2 \sum_{k \geq 0} |\psi_k| \sum_{j \geq 0} \sqrt{j} |\psi_j|.$$

4.41 Prove **Lemma C.4**.

4.42 Finish the proof of **Theorem C.5**.

4.43 For the zero-mean complex random vector $z = x_c - ix_s$, with $\text{cov}(z) = \Sigma = C - iQ$, with $\Sigma = \Sigma^*$, define

$$w = 2\text{Re}(a^* z),$$

where $a = a_c - ia_s$ is an arbitrary non-zero complex vector. Prove

$$\text{cov}(w) = 2a^* \Sigma a.$$

Recall $*$ denotes the complex conjugate transpose.

Chapter 5

Additional Time Domain Topics

In this chapter, we present material that may be considered special or advanced topics in the time domain. Chapter 6 is devoted to one of the most useful and interesting time domain topics, state-space models. Consequently, we do not cover state-space models or related topics—of which there are many—in this chapter. This chapter contains sections of independent topics that may be read in any order. Most of the sections depend on a basic knowledge of ARMA models, forecasting and estimation, which is the material that is covered in Chapter 3. A few sections, for example the section on long memory models, require some knowledge of spectral analysis and related topics covered in Chapter 4. In addition to long memory, we discuss unit root testing, GARCH models, threshold models , lagged regression or transfer functions, and selected topics in multivariate ARMAX models.

5.1 Long Memory ARMA and Fractional Differencing

The conventional ARMA(p, q) process is often referred to as a short-memory process because the coefficients in the representation

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j},$$

obtained by solving

$$\phi(z)\psi(z) = \theta(z),$$

are dominated by exponential decay. As pointed out in Section 3.2 and Section 3.3, this result implies the ACF of the short memory process satisfies $\rho(h) \rightarrow 0$ exponentially fast as $h \rightarrow \infty$. When the sample ACF of a time series decays slowly, the advice given in Chapter 3 has been to difference the series until it seems stationary. Following this advice with the glacial varve series first presented in Example 3.33 leads to the first difference of the logarithms of the data being represented as a first-order moving average. In Example 3.41, further analysis of the residuals leads to fitting an ARIMA(1, 1, 1) model,

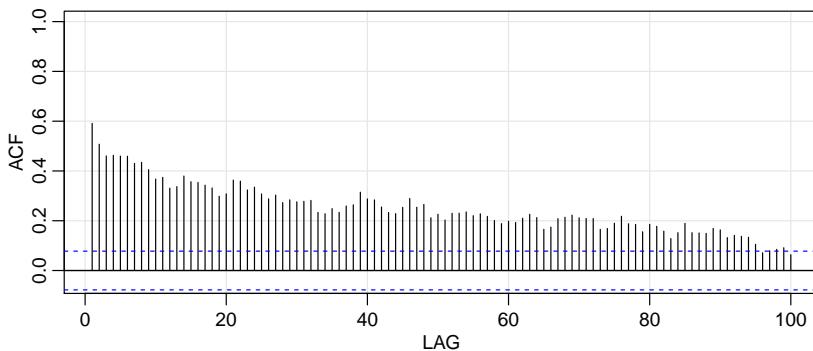


Fig. 5.1. Sample ACF of the log transformed varve series.

$$\nabla x_t = \phi \nabla x_{t-1} + w_t + \theta w_{t-1},$$

where we understand x_t is the log-transformed varve series. In particular, the estimates of the parameters (and the standard errors) were $\hat{\phi} = .23(.05)$, $\hat{\theta} = -.89(.03)$, and $\hat{\sigma}_w^2 = .23$.

The use of the first difference $\nabla x_t = (1 - B)x_t$, however, can sometimes be too severe a modification in the sense that the nonstationary model might represent an overdifferencing of the original process. Long memory (or persistent) time series were considered in Hosking (1981) and Granger and Joyeux (1980) as intermediate compromises between the short memory ARMA type models and the fully integrated nonstationary processes in the Box–Jenkins class. The easiest way to generate a long memory series is to think of using the difference operator $(1 - B)^d$ for fractional values of d , say, $0 < d < .5$, so a basic long memory series gets generated as

$$(1 - B)^d x_t = w_t, \quad (5.1)$$

where w_t still denotes white noise with variance σ_w^2 . The fractionally differenced series (5.1), for $|d| < .5$, is often called *fractional noise* (except when d is zero). Now, d becomes a parameter to be estimated along with σ_w^2 . Differencing the original process, as in the Box–Jenkins approach, may be thought of as simply assigning a value of $d = 1$. This idea has been extended to the class of fractionally integrated ARMA, or ARFIMA models, where $-.5 < d < .5$; when d is negative, the term antipersistent is used. Long memory processes occur in hydrology (see Hurst, 1951, and McLeod and Hipel, 1978) and in environmental series, such as the varve data we have previously analyzed, to mention a few examples. Long memory time series data tend to exhibit sample autocorrelations that are not necessarily large (as in the case of $d = 1$), but persist for a long time. Figure 5.1 shows the sample ACF, to lag 100, of the log-transformed varve series, which exhibits classic long memory behavior:

```
acf(log(varve), 100)
acf(cumsum(rnorm(1000)), 100) # compare to ACF of random walk (not shown)
```

Figure 5.1 can be contrasted with the ACF of the original GNP series shown in **Figure 3.13**, which is also persistent and decays linearly, but the values of the ACF are large.

To investigate its properties, we can use the binomial expansion ($d > -1$) to write

$$w_t = (1 - B)^d x_t = \sum_{j=0}^{\infty} \pi_j B^j x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} \quad (5.2)$$

where

$$\pi_j = \frac{\Gamma(j-d)}{\Gamma(j+1)\Gamma(-d)} \quad (5.3)$$

with $\Gamma(x+1) = x\Gamma(x)$ being the gamma function. Similarly ($d < 1$), we can write

$$x_t = (1 - B)^{-d} w_t = \sum_{j=0}^{\infty} \psi_j B^j w_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} \quad (5.4)$$

where

$$\psi_j = \frac{\Gamma(j+d)}{\Gamma(j+1)\Gamma(d)}. \quad (5.5)$$

When $|d| < .5$, the processes (5.2) and (5.4) are well-defined stationary processes (see Brockwell and Davis, 1991, for details). In the case of fractional differencing, however, the coefficients satisfy $\sum \pi_j^2 < \infty$ and $\sum \psi_j^2 < \infty$ as opposed to the absolute summability of the coefficients in ARMA processes.

Using the representation (5.4)–(5.5), and after some nontrivial manipulations, it can be shown that the ACF of x_t is

$$\rho(h) = \frac{\Gamma(h+d)\Gamma(1-d)}{\Gamma(h-d+1)\Gamma(d)} \sim h^{2d-1} \quad (5.6)$$

for large h . From this we see that for $0 < d < .5$

$$\sum_{h=-\infty}^{\infty} |\rho(h)| = \infty$$

and hence the term *long memory*.

In order to examine a series such as the varve series for a possible long memory pattern, it is convenient to look at ways of estimating d . Using (5.3) it is easy to derive the recursions

$$\pi_{j+1}(d) = \frac{(j-d)\pi_j(d)}{(j+1)}, \quad (5.7)$$

for $j = 0, 1, \dots$, with $\pi_0(d) = 1$. Maximizing the joint likelihood of the errors under normality, say, $w_t(d)$, will involve minimizing the sum of squared errors

$$Q(d) = \sum w_t^2(d).$$

The usual Gauss–Newton method, described in **Section 3.5**, leads to the expansion

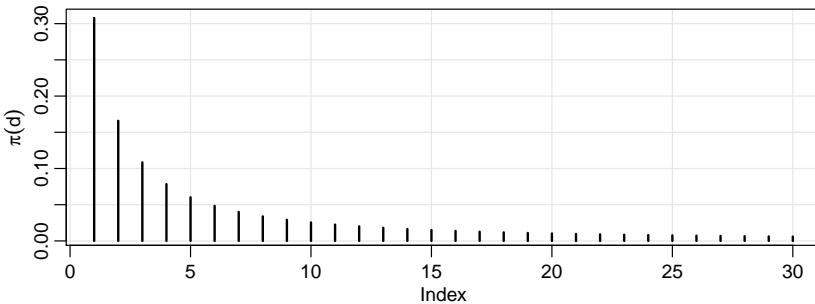


Fig. 5.2. Coefficients $\pi_j(.384)$, $j = 1, 2, \dots, 30$ in the representation (5.7).

$$w_t(d) = w_t(d_0) + w'_t(d_0)(d - d_0),$$

where

$$w'_t(d_0) = \left. \frac{\partial w_t}{\partial d} \right|_{d=d_0}$$

and d_0 is an initial estimate (guess) at to the value of d . Setting up the usual regression leads to

$$d = d_0 - \frac{\sum_t w'_t(d_0)w_t(d_0)}{\sum_t w'_t(d_0)^2}. \quad (5.8)$$

The derivatives are computed recursively by differentiating (5.7) successively with respect to d :

$$\pi'_{j+1}(d) = \frac{(j-d)\pi'_j(d) - \pi_j(d)}{j+1},$$

where $\pi'_0(d) = 0$. The errors are computed from an approximation to (5.2), namely,

$$w_t(d) = \sum_{j=0}^t \pi_j(d)x_{t-j}. \quad (5.9)$$

It is advisable to omit a number of initial terms from the computation and start the sum, (5.8), at some fairly large value of t to have a reasonable approximation.

Example 5.1 Long Memory Fitting of the Glacial Varve Series

We consider analyzing the glacial varve series discussed in various examples and first presented in Example 2.7 . Figure 2.7 shows the original and log-transformed series (which we denote by x_t). In Example 3.41, we noted that x_t could be modeled as an ARIMA(1, 1, 1) process. We fit the fractionally differenced model, (5.1), to the mean-adjusted series, $x_t - \bar{x}$. Applying the Gauss–Newton iterative procedure previously described, starting with $d = .1$ and omitting the first 30 points from the computation, leads to a final value of $d = .384$, which implies the set of coefficients $\pi_j(.384)$, as given in Figure 5.2 with $\pi_0(.384) = 1$. We can compare roughly the performance of the fractional difference operator with the ARIMA model by

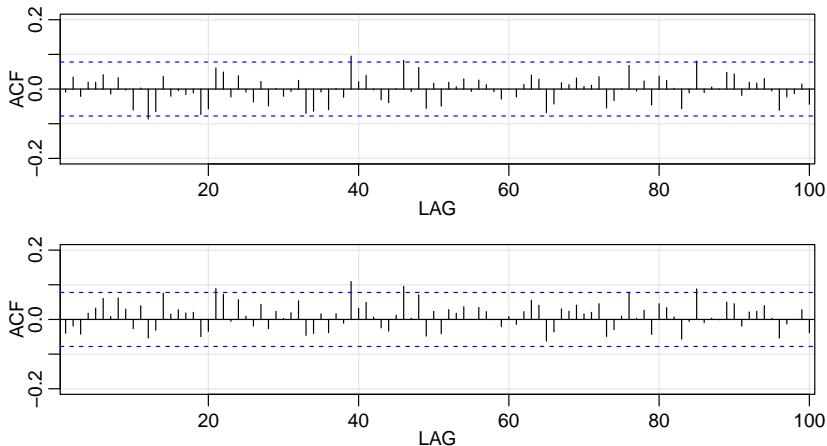


Fig. 5.3. ACF of residuals from the ARIMA(1,1,1) fit to the logged varve series (top) and of the residuals from the long memory model fit, $(1 - B)^d x_t = w_t$, with $d = .384$ (bottom).

examining the autocorrelation functions of the two residual series as shown in Figure 5.3. The ACFs of the two residual series are roughly comparable with the white noise model.

To perform this analysis in R, first download and install the `fracdiff` package. Then use

```
library(fracdiff)
lvarve = log(varve)-mean(log(varve))
varve.fd = fracdiff(lvarve, nar=0, nma=0, M=30)
varve.fd$d          # = 0.3841688
varve.fd$stderror.dpq    # = 4.589514e-06 (questionable result!!)
p = rep(1,31)
for (k in 1:30){ p[k+1] = (k-varve.fd$d)*p[k]/(k+1) }
plot(1:30, p[-1], ylab=expression(pi(d)), xlab="Index", type="h")
res.fd = diffseries(log(varve), varve.fd$d)           # frac diff resids
res.arima = resid(arima(log(varve), order=c(1,1,1))) # arima resids
par(mfrow=c(2,1))
acf(res.arima, 100, xlim=c(4,97), ylim=c(-.2,.2), main="")
acf(res.fd, 100, xlim=c(4,97), ylim=c(-.2,.2), main="")
```

The R package uses a truncated maximum likelihood procedure that was discussed in Haslett and Raftery (1989), which is a little more elaborate than simply zeroing out initial values. The default truncation value in R is $M = 100$. In the default case, the estimate is $\hat{d} = .37$ with approximately the same *questionable* standard error. The standard error is (supposedly) obtained from the Hessian as described in Example 3.30. A more believable standard error is given in Example 5.2.

Forecasting long memory processes is similar to forecasting ARIMA models. That is, (5.2) and (5.7) can be used to obtain the truncated forecasts

$$\tilde{x}_{n+m}^n = - \sum_{j=1}^n \pi_j(\hat{d}) \tilde{x}_{n+m-j}^n, \quad (5.10)$$

for $m = 1, 2, \dots$. Error bounds can be approximated by using

$$P_{n+m}^n = \hat{\sigma}_w^2 \left(\sum_{j=0}^{m-1} \psi_j^2(\hat{d}) \right) \quad (5.11)$$

where, as in (5.7),

$$\psi_j(\hat{d}) = \frac{(j + \hat{d})\psi_j(\hat{d})}{(j + 1)}, \quad (5.12)$$

with $\psi_0(\hat{d}) = 1$.

No obvious short memory ARMA-type component can be seen in the ACF of the residuals from the fractionally differenced varve series shown in Figure 5.3. It is natural, however, that cases will exist in which substantial short memory-type components will also be present in data that exhibits long memory. Hence, it is natural to define the general ARFIMA(p, d, q), $-.5 < d < .5$ process as

$$\phi(B)\nabla^d(x_t - \mu) = \theta(B)w_t, \quad (5.13)$$

where $\phi(B)$ and $\theta(B)$ are as given in Chapter 3. Writing the model in the form

$$\phi(B)\pi_d(B)(x_t - \mu) = \theta(B)w_t \quad (5.14)$$

makes it clear how we go about estimating the parameters for the more general model. Forecasting for the ARFIMA(p, d, q) series can be easily done, noting that we may equate coefficients in

$$\phi(z)\psi(z) = (1 - z)^{-d}\theta(z) \quad (5.15)$$

and

$$\theta(z)\pi(z) = (1 - z)^d\phi(z) \quad (5.16)$$

to obtain the representations

$$x_t = \mu + \sum_{j=0}^{\infty} \psi_j w_{t-j} \quad \text{and} \quad w_t = \sum_{j=0}^{\infty} \pi_j(x_{t-j} - \mu).$$

We then can proceed as discussed in (5.10) and (5.11).

Comprehensive treatments of long memory time series models are given in the texts by Beran (1994), Palma (2007), and Robinson (2003), and it should be noted that several other techniques for estimating the parameters, especially, the long memory parameter, can be developed in the frequency domain. In this case, we may think of the equations as generated by an infinite order autoregressive series with coefficients π_j given by (5.7). Using the same approach as before, we obtain

$$\begin{aligned} f_x(\omega) &= \frac{\sigma_w^2}{|\sum_{k=0}^{\infty} \pi_k e^{-2\pi i k \omega}|^2} \\ &= \sigma_w^2 |1 - e^{-2\pi i \omega}|^{-2d} = [4 \sin^2(\pi \omega)]^{-d} \sigma_w^2 \end{aligned} \quad (5.17)$$

as equivalent representations of the spectrum of a long memory process. The long memory spectrum approaches infinity as the frequency $\omega \rightarrow 0$.

The main reason for defining the Whittle approximation to the log likelihood is to propose its use for estimating the parameter d in the long memory case as an alternative to the time domain method previously mentioned. The time domain approach is useful because of its simplicity and easily computed standard errors. One may also use an exact likelihood approach by developing an innovations form of the likelihood as in Brockwell and Davis (1991).

For the approximate approach using the Whittle likelihood (4.85), we consider using the approach of Fox and Taqqu (1986) who showed that maximizing the Whittle log likelihood leads to a consistent estimator with the usual asymptotic normal distribution that would be obtained by treating (4.85) as a conventional log likelihood (see also Dahlhaus, 1989; Robinson, 1995; Hurvich et al., 1998). Unfortunately, the periodogram ordinates are not asymptotically independent (Hurvich and Beltrao, 1993), although a quasi-likelihood in the form of the Whittle approximation works well and has good asymptotic properties.

To see how this would work for the purely long memory case, write the long memory spectrum as

$$f_x(\omega_k; d, \sigma_w^2) = \sigma_w^2 g_k^{-d}, \quad (5.18)$$

where

$$g_k = 4 \sin^2(\pi \omega_k). \quad (5.19)$$

Then, differentiating the log likelihood, say,

$$\ln L(x; d, \sigma_w^2) \approx -m \ln \sigma_w^2 + d \sum_{k=1}^m \ln g_k - \frac{1}{\sigma_w^2} \sum_{k=1}^m g_k^d I(\omega_k) \quad (5.20)$$

at $m = n/2 - 1$ frequencies and solving for σ_w^2 yields

$$\sigma_w^2(d) = \frac{1}{m} \sum_{k=1}^m g_k^d I(\omega_k) \quad (5.21)$$

as the approximate maximum likelihood estimator for the variance parameter. To estimate d , we can use a grid search of the concentrated log likelihood

$$\ln L(x; d) \approx -m \ln \sigma_w^2(d) + d \sum_{k=1}^m \ln g_k - m \quad (5.22)$$

over the interval $(0, .5)$, followed by a Newton–Raphson procedure to convergence.

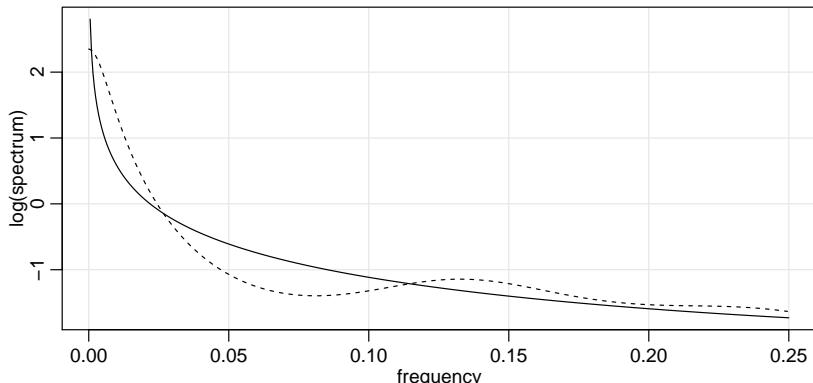


Fig. 5.4. Long Memory ($d = .380$) [solid line] and autoregressive AR(8) [dashed line] spectral estimators for the paleoclimatic glacial varve series.

Example 5.2 Long Memory Spectra for the Varve Series

In Example 5.1, we fit a long memory model to the glacial varve data via time domain methods. Fitting the same model using frequency domain methods and the Whittle approximation above gives $\hat{d} = .380$, with an estimated standard error of .028. The earlier time domain method gave $\hat{d} = .384$ with $M = 30$ and $\hat{d} = .370$ with $M = 100$. Both estimates obtained via time domain methods had a standard error of about 4.6×10^{-6} , which seems implausible. The error variance estimate in this case is $\hat{\sigma}_w^2 = .2293$; in Example 5.1, we could have used `var(res.fd)` as an estimate, in which case we obtain .2298. The R code to perform this analysis is

```
series = log(varve)      # specify series to be analyzed
d0 = .1                  # initial value of d
n.per = nextn(length(series))
m = (n.per)/2 - 1
per = Mod(fft(series-mean(series))[-1])^2 # remove 0 freq and
per = per/n.per           # scale the periodogram
g = 4*(sin(pi*((1:m)/n.per))^2)
# Function to calculate -log.likelihood
whit.like = function(d){
  g.d=g^d
  sig2 = (sum(g.d*per[1:m])/m)
  log.like = m*log(sig2) - d*sum(log(g)) + m
  return(log.like)
}
# Estimation (output not shown)
est = optim(d0, whit.like, gr=NULL, method="L-BFGS-B", hessian=TRUE,
            lower=-.5, upper=.5, control=list(trace=1,REPORT=1))
##-- Results: d.hat = .380, se(dhat) = .028, and sig2hat = .229 --##
cat("d.hat =", est$par, "se(dhat) = ", 1/sqrt(est$hessian), "\n")
g.dhat = g^est$par; sig2 = sum(g.dhat*per[1:m])/m
cat("sig2hat =", sig2, "\n")
```

One might also consider fitting an autoregressive model to these data using a procedure similar to that used in Example 4.18. Following this approach gave an autoregressive model with $p = 8$ and $\hat{\phi}_{1:8} = \{.34, .11, .04, .09, .08, .02, .09\}$,

with $\hat{\sigma}_w^2 = .23$ as the error variance. The two log spectra are plotted in [Figure 5.4](#) for $\omega > 0$, and we note that long memory spectrum will eventually become infinite, whereas the AR(8) spectrum is finite at $\omega = 0$. The R code used for this part of the example (assuming the previous values have been retained) is

```
u = spec.ar(log(varve), plot=FALSE) # produces AR(8)
g = 4*(sin(pi*((1:500)/2000))^2)
fhat = sig2*g^{est$par} # long memory spectral estimate
plot(1:500/2000, log(fhat), type="l", ylab="log(spectrum)", xlab="frequency")
lines(u$freq[1:250], log(u$spec[1:250]), lty="dashed")
ar.mle(log(varve)) # to get AR(8) estimates
```

Often, time series are not purely long memory. A common situation has the long memory component multiplied by a short memory component, leading to an alternate version of [\(5.18\)](#) of the form

$$f_x(\omega_k; d, \theta) = g_k^{-d} f_0(\omega_k; \theta), \quad (5.23)$$

where $f_0(\omega_k; \theta)$ might be the spectrum of an autoregressive moving average process with vector parameter θ , or it might be unspecified. If the spectrum has a parametric form, the Whittle likelihood can be used. However, there is a substantial amount of semiparametric literature that develops the estimators when the underlying spectrum $f_0(\omega; \theta)$ is unknown. A class of *Gaussian semi-parametric* estimators simply uses the same Whittle likelihood [\(5.22\)](#), evaluated over a sub-band of low frequencies, say $m' = \sqrt{n}$. There is some latitude in selecting a band that is relatively free from low frequency interference due to the short memory component in [\(5.23\)](#). If the spectrum is highly parameterized, one might estimate using the Whittle log likelihood [\(5.19\)](#) under [\(5.23\)](#) and jointly estimate the parameters d and θ using the Newton–Raphson method. If we are interested in a nonparametric estimator, using the conventional smoothed spectral estimator for the periodogram, adjusted for the long memory component, say $g_k^d I(\omega_k)$ might be a possible approach.

Geweke and Porter–Hudak (1983) developed an approximate method for estimating d based on a regression model, derived from [\(5.22\)](#). Note that we may write a simple equation for the logarithm of the spectrum as

$$\ln f_x(\omega_k; d) = \ln f_0(\omega_k; \theta) - d \ln[4 \sin^2(\pi \omega_k)], \quad (5.24)$$

with the frequencies $\omega_k = k/n$ restricted to a range $k = 1, 2, \dots, m$ near the zero frequency with $m = \sqrt{n}$ as the recommended value. Relationship [\(5.24\)](#) suggests using a simple linear regression model of the form,

$$\ln I(\omega_k) = \beta_0 - d \ln[4 \sin^2(\pi \omega_k)] + e_k \quad (5.25)$$

for the periodogram to estimate the parameters σ_w^2 and d . In this case, one performs least squares using $\ln I(\omega_k)$ as the dependent variable, and $\ln[4 \sin^2(\pi \omega_k)]$ as the independent variable for $k = 1, 2, \dots, m$. The resulting slope estimate is then used as an estimate of $-d$. For a good discussion of various alternative methods for selecting m , see Hurvich and Deo (1999). The R package `fracdiff` also provides this method via the command `fdGPH()`; see the help file for further information. Here is a quick example using the logged varve data.

```
library(fracdiff)
fdGPH(log(varve), bandw=.9) # m = n^bandw
dhat = 0.383 se(dhat) = 0.041
```

5.2 Unit Root Testing

As discussed in the previous section, the use of the first difference $\nabla x_t = (1 - B)x_t$ can be too severe a modification in the sense that the nonstationary model might represent an overdifferencing of the original process. For example, consider a causal AR(1) process (we assume throughout this section that the noise is Gaussian),

$$x_t = \phi x_{t-1} + w_t. \quad (5.26)$$

Applying $(1 - B)$ to both sides shows that differencing, $\nabla x_t = \phi \nabla x_{t-1} + \nabla w_t$, or

$$y_t = \phi y_{t-1} + w_t - w_{t-1},$$

where $y_t = \nabla x_t$, introduces extraneous correlation and invertibility problems. That is, while x_t is a causal AR(1) process, working with the differenced process y_t will be problematic because it is a non-invertible ARMA(1, 1).

A unit root test provides a way to test whether (5.26) is a random walk (the null case) as opposed to a causal process (the alternative). That is, it provides a procedure for testing

$$H_0: \phi = 1 \quad \text{versus} \quad H_1: |\phi| < 1.$$

An obvious test statistic would be to consider $(\hat{\phi} - 1)$, appropriately normalized, in the hope to develop an asymptotically normal test statistic, where $\hat{\phi}$ is one of the optimal estimators discussed in Chapter 3. Unfortunately, the theory of Section 3.5 will not work in the null case because the process is nonstationary. Moreover, as seen in Example 3.36, estimation near the boundary of stationarity produces highly skewed sample distributions (see Figure 3.12) and this is a good indication that the problem will be atypical.

To examine the behavior of $(\hat{\phi} - 1)$ under the null hypothesis that $\phi = 1$, or more precisely that the model is a random walk, $x_t = \sum_{j=1}^t w_j$, or $x_t = x_{t-1} + w_t$ with $x_0 = 0$, consider the least squares estimator of ϕ . Noting that $\mu_x = 0$, the least squares estimator can be written as

$$\hat{\phi} = \frac{\sum_{t=1}^n x_t x_{t-1}}{\sum_{t=1}^n x_{t-1}^2} = 1 + \frac{\frac{1}{n} \sum_{t=1}^n w_t x_{t-1}}{\frac{1}{n} \sum_{t=1}^n x_{t-1}^2}, \quad (5.27)$$

where we have written $x_t = x_{t-1} + w_t$ in the numerator; recall that $x_0 = 0$ and in the least squares setting, we are regressing x_t on x_{t-1} for $t = 1, \dots, n$. Hence, under H_0 , we have that

$$\hat{\phi} - 1 = \frac{\frac{1}{n\sigma_w^2} \sum_{t=1}^n w_t x_{t-1}}{\frac{1}{n\sigma_w^2} \sum_{t=1}^n x_{t-1}^2}. \quad (5.28)$$

Consider the numerator of (5.28). Note first that by squaring both sides of $x_t = x_{t-1} + w_t$, we obtain $x_t^2 = x_{t-1}^2 + 2x_{t-1}w_t + w_t^2$ so that

$$x_{t-1}w_t = \frac{1}{2}(x_t^2 - x_{t-1}^2 - w_t^2),$$

and summing,

$$\frac{1}{n\sigma_w^2} \sum_{t=1}^n x_{t-1}w_t = \frac{1}{2} \left(\frac{x_n^2}{n\sigma_w^2} - \frac{\sum_{t=1}^n w_t^2}{n\sigma_w^2} \right).$$

Because $x_n = \sum_1^n w_t$, we have that $x_n \sim N(0, n\sigma_w^2)$, so that $\chi_1^2 = \frac{1}{n\sigma_w^2} x_n^2$ has a chi-squared distribution with one degree of freedom. Moreover, because w_t is white Gaussian noise, $\frac{1}{n} \sum_1^n w_t^2 \rightarrow_p \sigma_w^2$, or $\frac{1}{n\sigma_w^2} \sum_1^n w_t^2 \rightarrow_p 1$. Consequently ($n \rightarrow \infty$),

$$\frac{1}{n\sigma_w^2} \sum_{t=1}^n x_{t-1}w_t \xrightarrow{d} \frac{1}{2}(\chi_1^2 - 1). \quad (5.29)$$

Next we focus on the denominator of (5.28). First, we introduce standard Brownian motion.

Definition 5.1 A continuous time process $\{W(t); t \geq 0\}$ is called **standard Brownian motion** if it satisfies the following conditions:

- (i) $W(0) = 0$;
- (ii) $\{W(t_2) - W(t_1), W(t_3) - W(t_2), \dots, W(t_n) - W(t_{n-1})\}$ are independent for any collection of points, $0 \leq t_1 < t_2 \dots < t_n$, and integer $n > 2$;
- (iii) $W(t + \Delta t) - W(t) \sim N(0, \Delta t)$ for $\Delta t > 0$.

In addition to (i)–(iii), it is assumed that almost all sample paths of $W(t)$ are continuous in t . The result for the denominator uses the functional central limit theorem, which can be found in Billingsley (1999, §2.8). In particular, if ξ_1, \dots, ξ_n is a sequence of iid random variables with mean 0 and variance 1, then, for $0 \leq t \leq 1$, the continuous time process^{5.1}

$$S_n(t) = \frac{1}{\sqrt{n}} \sum_{j=1}^{\lfloor nt \rfloor} \xi_j \xrightarrow{d} W(t), \quad (5.30)$$

as $n \rightarrow \infty$, where $\lfloor \cdot \rfloor$ is the greatest integer function and $W(t)$ is standard Brownian motion on $[0, 1]$. Note the under the null hypothesis, $x_s = w_1 + \dots + w_s \sim N(0, s\sigma_w^2)$, and based on (5.30), we have $\frac{x_s}{\sigma_w \sqrt{n}} \rightarrow_d W(s)$. From this fact, we can show that ($n \rightarrow \infty$)

$$\sum_{t=1}^n \left(\frac{x_{t-1}}{\sigma_w \sqrt{n}} \right)^2 \frac{1}{n} \xrightarrow{d} \int_0^1 W^2(t) dt. \quad (5.31)$$

The denominator in (5.28) is off from the left side of (5.31) by a factor of n^{-1} , and we adjust accordingly to finally obtain ($n \rightarrow \infty$),

^{5.1} The intuition here is, for $k = \lfloor nt \rfloor$ and fixed t , the central limit theorem has $\sqrt{t} \frac{1}{\sqrt{k}} \sum_{j=1}^k \xi_j \sim AN(0, t)$ with $n \rightarrow \infty$.

$$n(\hat{\phi} - 1) = \frac{\frac{1}{n\sigma_w^2} \sum_{t=1}^n w_t x_{t-1}}{\frac{1}{n^2\sigma_w^2} \sum_{t=1}^n x_{t-1}^2} \xrightarrow{d} \frac{\frac{1}{2}(\chi_1^2 - 1)}{\int_0^1 W^2(t) dt}. \quad (5.32)$$

The test statistic $n(\hat{\phi} - 1)$ is known as the unit root or Dickey-Fuller (DF) statistic (see Fuller, 1976 or 1996), although the actual DF test statistic is normalized a little differently. Related derivations were discussed in Rao (1978; Correction 1980) and in Evans & Savin (1981). Because the distribution of the test statistic does not have a closed form, quantiles of the distribution must be computed by numerical approximation or by simulation. The R package [tseries](#) provides this test along with more general tests that we mention briefly.

Toward a more general model, we note that the DF test was established by noting that if $x_t = \phi x_{t-1} + w_t$, then $\nabla x_t = (\phi - 1)x_{t-1} + w_t = \gamma x_{t-1} + w_t$, and one could test $H_0 : \gamma = 0$ by regressing ∇x_t on x_{t-1} . They formed a Wald statistic and derived its limiting distribution [the previous derivation based on Brownian motion is due to Phillips (1987)]. The test was extended to accommodate AR(p) models, $x_t = \sum_{j=1}^p \phi_j x_{t-j} + w_t$, as follows. Subtract x_{t-1} from both sides to obtain

$$\nabla x_t = \gamma x_{t-1} + \sum_{j=1}^{p-1} \psi_j \nabla x_{t-j} + w_t, \quad (5.33)$$

where $\gamma = \sum_{j=1}^p \phi_j - 1$ and $\psi_j = -\sum_{i=j}^p \phi_i$ for $j = 2, \dots, p$. For a quick check of (5.33) when $p = 2$, note that $x_t = (\phi_1 + \phi_2)x_{t-1} - \phi_2(x_{t-1} - x_{t-2}) + w_t$; now subtract x_{t-1} from both sides. To test the hypothesis that the process has a unit root at 1 (i.e., the AR polynoimial $\phi(z) = 0$ when $z = 1$), we can test $H_0 : \gamma = 0$ by estimating γ in the regression of ∇x_t on $x_{t-1}, \nabla x_{t-1}, \dots, \nabla x_{t-p+1}$, and forming a Wald test based on $t_\gamma = \hat{\gamma}/\text{se}(\hat{\gamma})$. This test leads to the so-called augmented Dickey-Fuller test (ADF). While the calculations for obtaining the asymptotic null distribution change, the basic ideas and machinery remain the same as in the simple case. The choice of p is crucial, and we will discuss some suggestions in the example. For ARMA(p, q) models, the ADF test can be used by assuming p is large enough to capture the essential correlation structure; another alternative is the Phillips-Perron (PP) test, which differs from the ADF tests mainly in how they deal with serial correlation and heteroskedasticity in the errors.

One can extend the model to include a constant, or even non-stochastic trend. For example, consider the model

$$x_t = \beta_0 + \beta_1 t + \phi x_{t-1} + w_t.$$

If we assume $\beta_1 = 0$, then under the null hypothesis, $\phi = 1$, the process is a random walk with drift β_0 . Under the alternate hypothesis, the process is a causal AR(1) with mean $\mu_x = \beta_0(1 - \phi)$. If we cannot assume $\beta_1 = 0$, then the interest here is testing the null that $(\beta_1, \phi) = (0, 1)$, simultaneously, versus the alternative that $\beta_1 \neq 0$ and $|\phi| < 1$. In this case, the null hypothesis is that the process is a random walk with drift, versus the alternative hypothesis that the process is trend stationary such as might be considered for the chicken price series in [Example 2.1](#).

Example 5.3 Testing Unit Roots in the Glacial Varve Series

In this example we use the R package `tseries` to test the null hypothesis that the log of the glacial varve series has a unit root, versus the alternate hypothesis that the process is stationary. We test the null hypothesis using the available DF, ADF and PP tests; note that in each case, the general regression equation incorporates a constant and a linear trend. In the ADF test, the default number of AR components included in the model, say k , is $\lceil (n - 1)^{\frac{1}{3}} \rceil$, which corresponds to the suggested upper bound on the rate at which the number of lags, k , should be made to grow with the sample size for the general ARMA(p, q) setup. For the PP test, the default value of k is $\lceil 0.04n^{\frac{1}{4}} \rceil$.

```
library(tseries)
adf.test(log(varve), k=0)          # DF test
  Dickey-Fuller = -12.8572, Lag order = 0, p-value < 0.01
  alternative hypothesis: stationary
adf.test(log(varve))               # ADF test
  Dickey-Fuller = -3.5166, Lag order = 8, p-value = 0.04071
  alternative hypothesis: stationary
pp.test(log(varve))                # PP test
  Dickey-Fuller Z(alpha) = -304.5376,
  Truncation lag parameter = 6, p-value < 0.01
  alternative hypothesis: stationary
```

In each test, we reject the null hypothesis that the logged varve series has a unit root. The conclusion of these tests supports the conclusion of the previous section that the logged varve series is long memory rather than integrated.

5.3 GARCH Models

Various problems such as option pricing in finance have motivated the study of the *volatility*, or variability, of a time series. ARMA models were used to model the conditional mean of a process when the conditional variance was constant. Using an AR(1) as an example, we assumed

$$E(x_t | x_{t-1}, x_{t-2}, \dots) = \phi x_{t-1}, \quad \text{and} \quad \text{var}(x_t | x_{t-1}, x_{t-2}, \dots) = \text{var}(w_t) = \sigma_w^2.$$

In many problems, however, the assumption of a constant conditional variance will be violated. Models such as the *autoregressive conditionally heteroscedastic* or ARCH model, first introduced by Engle (1982), were developed to model changes in volatility. These models were later extended to generalized ARCH, or GARCH models by Bollerslev (1986).

In these problems, we are concerned with modeling the return or growth rate of a series. For example, if x_t is the value of an asset at time t , then the return or relative gain, r_t , of the asset at time t is

$$r_t = \frac{x_t - x_{t-1}}{x_{t-1}}. \tag{5.34}$$

Definition (5.34) implies that $x_t = (1 + r_t)x_{t-1}$. Thus, based on the discussion in Section 3.7, if the return represents a small (in magnitude) percentage change then

$$\nabla \log(x_t) \approx r_t. \quad (5.35)$$

Either value, $\nabla \log(x_t)$ or $(x_t - x_{t-1})/x_{t-1}$, will be called the *return*,^{5.2} and will be denoted by r_t . An alternative to the GARCH model is the *stochastic volatility model*; we will discuss these models in [Chapter 6](#) because they are state-space models.

Typically, for financial series, the return r_t , does not have a constant conditional variance, and highly volatile periods tend to be clustered together. In other words, there is a strong dependence of sudden bursts of variability in a return on the series own past. For example, [Figure 1.4](#) shows the daily returns of the Dow Jones Industrial Average (DJIA) from April 20, 2006 to April 20, 2016. In this case, as is typical, the return r_t is fairly stable, except for short-term bursts of high volatility.

The simplest ARCH model, the ARCH(1), models the return as

$$r_t = \sigma_t \epsilon_t \quad (5.36)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2, \quad (5.37)$$

where ϵ_t is standard Gaussian white noise, $\epsilon_t \sim \text{iid } N(0, 1)$. The normal assumption may be relaxed; we will discuss this later. As with ARMA models, we must impose some constraints on the model parameters to obtain desirable properties. An obvious constraint is that $\alpha_0, \alpha_1 \geq 0$ because σ_t^2 is a variance.

As we shall see, the ARCH(1) models return as a white noise process with non-constant conditional variance, and that conditional variance depends on the previous return. First, notice that the conditional distribution of r_t given r_{t-1} is Gaussian:

$$r_t \mid r_{t-1} \sim N(0, \alpha_0 + \alpha_1 r_{t-1}^2). \quad (5.38)$$

In addition, it is possible to write the ARCH(1) model as a non-Gaussian AR(1) model in the square of the returns r_t^2 . First, rewrite [\(5.36\)](#)–[\(5.37\)](#) as

$$\begin{aligned} r_t^2 &= \sigma_t^2 \epsilon_t^2 \\ \alpha_0 + \alpha_1 r_{t-1}^2 &= \sigma_t^2, \end{aligned}$$

and subtract the two equations to obtain

$$r_t^2 - (\alpha_0 + \alpha_1 r_{t-1}^2) = \sigma_t^2 \epsilon_t^2 - \sigma_t^2.$$

Now, write this equation as

$$r_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + v_t, \quad (5.39)$$

where $v_t = \sigma_t^2(\epsilon_t^2 - 1)$. Because ϵ_t^2 is the square of a $N(0, 1)$ random variable, $\epsilon_t^2 - 1$ is a shifted (to have mean-zero), χ_1^2 random variable.

To explore the properties of ARCH, we define $\mathcal{R}_s = \{r_s, r_{s-1}, \dots\}$. Then, using [\(5.38\)](#), we immediately see that r_t has a zero mean:

^{5.2} Recall from [Footnote 1.2](#) that if $r_t = (x_t - x_{t-1})/x_{t-1}$ is a small percentage, then $\log(1 + r_t) \approx r_t$. It is easier to program $\nabla \log x_t$, so this is often used instead of calculating r_t directly. Although it is a misnomer, $\nabla \log x_t$ is often called the *log-return*; but the returns are not being logged.

$$\text{E}(r_t) = \text{EE}(r_t \mid \mathcal{R}_{t-1}) = \text{EE}(r_t \mid r_{t-1}) = 0. \quad (5.40)$$

Because $\text{E}(r_t \mid \mathcal{R}_{t-1}) = 0$, the process r_t is said to be a *martingale difference*.

Because r_t is a martingale difference, it is also an uncorrelated sequence. For example, with $h > 0$,

$$\begin{aligned} \text{cov}(r_{t+h}, r_t) &= \text{E}(r_t r_{t+h}) = \text{EE}(r_t r_{t+h} \mid \mathcal{R}_{t+h-1}) \\ &= \text{E}\{r_t \text{E}(r_{t+h} \mid \mathcal{R}_{t+h-1})\} = 0. \end{aligned} \quad (5.41)$$

The last line of (5.41) follows because r_t belongs to the information set \mathcal{R}_{t+h-1} for $h > 0$, and, $\text{E}(r_{t+h} \mid \mathcal{R}_{t+h-1}) = 0$, as determined in (5.40).

An argument similar to (5.40) and (5.41) will establish the fact that the error process v_t in (5.39) is also a martingale difference and, consequently, an uncorrelated sequence. If the variance of v_t is finite and constant with respect to time, and $0 \leq \alpha_1 < 1$, then based on [Property 3.1](#), (5.39) specifies a causal AR(1) process for r_t^2 . Therefore, $\text{E}(r_t^2)$ and $\text{var}(r_t^2)$ must be constant with respect to time t . This, implies that

$$\text{E}(r_t^2) = \text{var}(r_t) = \frac{\alpha_0}{1 - \alpha_1} \quad (5.42)$$

and, after some manipulations,

$$\text{E}(r_t^4) = \frac{3\alpha_0^2}{(1 - \alpha_1)^2} \frac{1 - \alpha_1^2}{1 - 3\alpha_1^2}, \quad (5.43)$$

provided $3\alpha_1^2 < 1$. Note that

$$\text{var}(r_t^2) = \text{E}(r_t^4) - [\text{E}(r_t^2)]^2,$$

which exists only if $0 < \alpha_1 < 1/\sqrt{3} \approx .58$. In addition, these results imply that the kurtosis, κ , of r_t is

$$\kappa = \frac{\text{E}(r_t^4)}{[\text{E}(r_t^2)]^2} = 3 \frac{1 - \alpha_1^2}{1 - 3\alpha_1^2}, \quad (5.44)$$

which is never smaller than 3, the kurtosis of the normal distribution. Thus, the marginal distribution of the returns, r_t , is leptokurtic, or has “fat tails.” Summarizing, if $0 \leq \alpha_1 < 1$, the process r_t itself is white noise and its unconditional distribution is symmetrically distributed around zero; this distribution is leptokurtic. If, in addition, $3\alpha_1^2 < 1$, the square of the process, r_t^2 , follows a causal AR(1) model with ACF given by $\rho_{y^2}(h) = \alpha_1^h \geq 0$, for all $h > 0$. If $3\alpha_1 \geq 1$, but $\alpha_1 < 1$, it can be shown that r_t^2 is strictly stationary with infinite variance (see Douc, et al., 2014).

Estimation of the parameters α_0 and α_1 of the ARCH(1) model is typically accomplished by conditional MLE. The conditional likelihood of the data r_2, \dots, r_n given r_1 , is given by

$$L(\alpha_0, \alpha_1 \mid r_1) = \prod_{t=2}^n f_{\alpha_0, \alpha_1}(r_t \mid r_{t-1}), \quad (5.45)$$

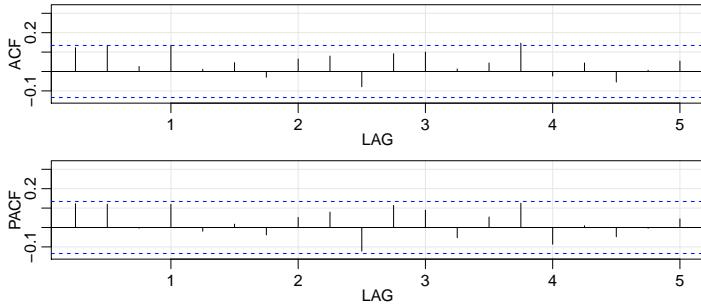


Fig. 5.5. ACF and PACF of the squares of the residuals from the AR(1) fit on U.S. GNP.

where the density $f_{\alpha_0, \alpha_1}(r_t \mid r_{t-1})$ is the normal density specified in (5.38). Hence, the criterion function to be minimized, $l(\alpha_0, \alpha_1) \propto -\ln L(\alpha_0, \alpha_1 \mid r_1)$ is given by

$$l(\alpha_0, \alpha_1) = \frac{1}{2} \sum_{t=2}^n \ln(\alpha_0 + \alpha_1 r_{t-1}^2) + \frac{1}{2} \sum_{t=2}^n \left(\frac{r_t^2}{\alpha_0 + \alpha_1 r_{t-1}^2} \right). \quad (5.46)$$

Estimation is accomplished by numerical methods, as described in [Section 3.5](#). In this case, analytic expressions for the gradient vector, $l^{(1)}(\alpha_0, \alpha_1)$, and Hessian matrix, $l^{(2)}(\alpha_0, \alpha_1)$, as described in [Example 3.30](#), can be obtained by straight-forward calculations. For example, the 2×1 gradient vector, $l^{(1)}(\alpha_0, \alpha_1)$, is given by

$$\begin{pmatrix} \partial l / \partial \alpha_0 \\ \partial l / \partial \alpha_1 \end{pmatrix} = \sum_{t=2}^n \begin{pmatrix} 1 \\ r_{t-1}^2 \end{pmatrix} \times \frac{\alpha_0 + \alpha_1 r_{t-1}^2 - r_t^2}{2 (\alpha_0 + \alpha_1 r_{t-1}^2)^2}.$$

The calculation of the Hessian matrix is left as an exercise ([Problem 5.8](#)). The likelihood of the ARCH model tends to be flat unless n is very large. A discussion of this problem can be found in Shephard (1996).

It is also possible to combine a regression or an ARMA model for the mean with an ARCH model for the errors. For example, a regression with ARCH(1) errors model would have the observations x_t as linear function of p regressors, $z_t = (z_{t1}, \dots, z_{tp})'$, and ARCH(1) noise y_t , say,

$$x_t = \beta' z_t + y_t,$$

where y_t satisfies (5.36)–(5.37), but, in this case, is unobserved. Similarly, for example, an AR(1) model for data x_t exhibiting ARCH(1) errors would be

$$x_t = \phi_0 + \phi_1 x_{t-1} + y_t.$$

These types of models were explored by Weiss (1984).

Example 5.4 Analysis of U.S. GNP

In Example 3.39, we fit an MA(2) model and an AR(1) model to the U.S. GNP series and we concluded that the residuals from both fits appeared to behave like a white noise process. In Example 3.43 we concluded that the AR(1) is probably the better model in this case. It has been suggested that the U.S. GNP series has ARCH errors, and in this example, we will investigate this claim. If the GNP noise term is ARCH, the squares of the residuals from the fit should behave like a non-Gaussian AR(1) process, as pointed out in (5.39). Figure 5.5 shows the ACF and PACF of the squared residuals it appears that there may be some dependence, albeit small, left in the residuals. The figure was generated in R as follows.

```
u = sarima(diff(log(gnp)), 1, 0, 0)
acf2(resid(u$fit)^2, 20)
```

We used the R package `fGarch` to fit an AR(1)-ARCH(1) model to the U.S. GNP returns with the following results. A partial output is shown; we note that `garch(1,0)` specifies an ARCH(1) in the code below (details later).

```
library(fGarch)
summary(garchFit(~arma(1,0)+garch(1,0), diff(log(gnp))))

```

	Estimate	Std.Error	t.value	p.value
mu	0.005	0.001	5.867	0.000
ar1	0.367	0.075	4.878	0.000
omega	0.000	0.000	8.135	0.000
alpha1	0.194	0.096	2.035	0.042
--				
Standardised Residuals Tests:	Statistic	p-Value		
Jarque-Bera Test	R	Chi^2	9.118	0.010
Shapiro-Wilk Test	R	W	0.984	0.014
Ljung-Box Test	R	Q(20)	23.414	0.269
Ljung-Box Test	R^2	Q(20)	37.743	0.010

Note that the p-values given in the estimation paragraph are two-sided, so they should be halved when considering the ARCH parameters. In this example, we obtain $\hat{\phi}_0 = .005$ (called `mu` in the output) and $\hat{\phi}_1 = .367$ (called `ar1`) for the AR(1) parameter estimates; in Example 3.39 the values were .005 and .347, respectively. The ARCH(1) parameter estimates are $\hat{\alpha}_0 = 0$ (called `omega`) for the constant and $\hat{\alpha}_1 = .194$, which is significant with a p-value of about .02. There are a number of tests that are performed on the residuals [R] or the squared residuals [R^2]. For example, the Jarque–Bera statistic tests the residuals of the fit for normality based on the observed skewness and kurtosis, and it appears that the residuals have some non-normal skewness and kurtosis. The Shapiro–Wilk statistic tests the residuals of the fit for normality based on the empirical order statistics. The other tests, primarily based on the Q-statistic, are used on the residuals and their squares.

The ARCH(1) model can be extended to the general ARCH(p) model in an obvious way. That is, (5.36), $r_t = \sigma_t \epsilon_t$, is retained, but (5.37) is extended to

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \cdots + \alpha_p r_{t-p}^2. \quad (5.47)$$

Estimation for ARCH(p) also follows in an obvious way from the discussion of estimation for ARCH(1) models. That is, the conditional likelihood of the data r_{p+1}, \dots, r_n given r_1, \dots, r_p , is given by

$$L(\alpha \mid r_1, \dots, r_p) = \prod_{t=p+1}^n f_\alpha(r_t \mid r_{t-1}, \dots, r_{t-m}), \quad (5.48)$$

where $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_p)$ and, under the assumption of normality, the conditional densities $f_\alpha(\cdot | \cdot)$ in (5.48) are, for $t > p$, given by

$$r_t \mid r_{t-1}, \dots, r_{t-p} \sim N(0, \alpha_0 + \alpha_1 r_{t-1}^2 + \dots + \alpha_m r_{t-p}^2).$$

Another extension of ARCH is the generalized ARCH or GARCH model developed by Bollerslev (1986). For example, a GARCH(1, 1) model retains (5.36), $r_t = \sigma_t \epsilon_t$, but extends (5.37) as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1}^2. \quad (5.49)$$

Under the condition that $\alpha_1 + \beta_1 < 1$, using similar manipulations as in (5.39), the GARCH(1, 1) model, (5.36) and (5.49), admits a non-Gaussian ARMA(1, 1) model for the squared process

$$r_t^2 = \alpha_0 + (\alpha_1 + \beta_1)r_{t-1}^2 + v_t - \beta_1 v_{t-1}, \quad (5.50)$$

where v_t is as defined in (5.39). Representation (5.50) follows by writing (5.36) as

$$\begin{aligned} r_t^2 - \sigma_t^2 &= \sigma_t^2(\epsilon_t^2 - 1) \\ \beta_1(r_{t-1}^2 - \sigma_{t-1}^2) &= \beta_1 \sigma_{t-1}^2(\epsilon_{t-1}^2 - 1), \end{aligned}$$

subtracting the second equation from the first, and using the fact that, from (5.49), $\sigma_t^2 - \beta_1 \sigma_{t-1}^2 = \alpha_0 + \alpha_1 r_{t-1}^2$, on the left-hand side of the result. The GARCH(p, q) model retains (5.36) and extends (5.49) to

$$\sigma_t^2 = \alpha_0 + \sum_{j=1}^p \alpha_j r_{t-j}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2. \quad (5.51)$$

Conditional maximum likelihood estimation of the GARCH(m, r) model parameters is similar to the ARCH(m) case, wherein the conditional likelihood, (5.48), is the product of $N(0, \sigma_t^2)$ densities with σ_t^2 given by (5.51) and where the conditioning is on the first $\max(m, r)$ observations, with $\sigma_1^2 = \dots = \sigma_r^2 = 0$. Once the parameter estimates are obtained, the model can be used to obtain *one-step-ahead forecasts* of the volatility, say $\hat{\sigma}_{t+1}^2$, given by

$$\hat{\sigma}_{t+1}^2 = \hat{\alpha}_0 + \sum_{j=1}^p \hat{\alpha}_j r_{t+1-j}^2 + \sum_{j=1}^q \hat{\beta}_j \hat{\sigma}_{t+1-j}^2. \quad (5.52)$$

We explore these concepts in the following example.

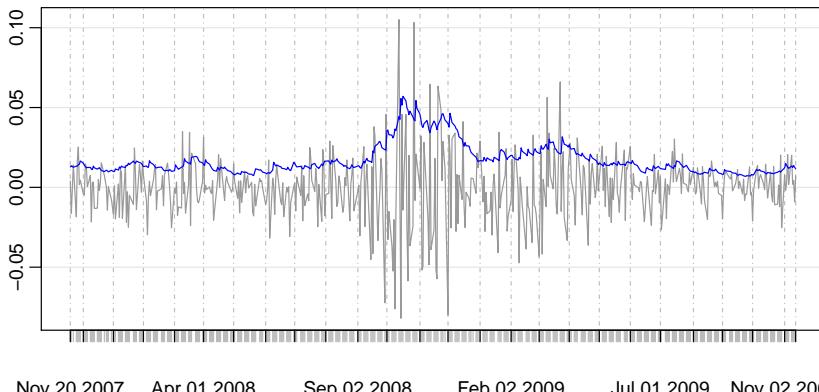


Fig. 5.6. GARCH one-step-ahead predictions of the DJIA volatility, $\hat{\sigma}_t$, superimposed on part of the data including the financial crisis of 2008.

Example 5.5 GARCH Analysis of the DJIA Returns

As previously mentioned, the daily returns of the DJIA shown in Figure 1.4 exhibit classic GARCH features. In addition, there is some low level autocorrelation in the series itself, and to include this behavior, we used the R `fGarch` package to fit an AR(1)-GARCH(1, 1) model to the series using t errors:

```
library(xts)
djiar = diff(log(djia$Close))[-1]
acf2(djiar) # exhibits some autocorrelation (not shown)
acf2(djiar^2) # oozes autocorrelation (not shown)
library(fGarch)
summary(djia.g <- garchFit(~arma(1,0)+garch(1,1), data=djiar,
                           cond.dist='std'))
plot(djia.g) # to see all plot options
      Estimate Std.Error t.value p.value
mu     8.585e-04 1.470e-04  5.842 5.16e-09
ar1    -5.531e-02 2.023e-02 -2.735 0.006239
omega   1.610e-06 4.459e-07  3.611 0.000305
alpha1  1.244e-01 1.660e-02   7.497 6.55e-14
beta1   8.700e-01 1.526e-02  57.022 < 2e-16
shape   5.979e+00 7.917e-01   7.552 4.31e-14
---
Standardised Residuals Tests:
      Statistic p-Value
Ljung-Box Test   R Q(10) 16.81507 0.0785575
Ljung-Box Test   R^2 Q(10) 15.39137 0.1184312
```

To explore the GARCH predictions of volatility, we calculated and plotted part of the data surrounding the financial crises of 2008 along with the one-step-ahead predictions of the corresponding volatility, σ_t^2 as a solid line in Figure 5.6.

Another model that we mention briefly is the *asymmetric power ARCH* model. The model retains (5.36), $r_t = \sigma_t \epsilon_t$, but the conditional variance is modeled as

$$\sigma_t^\delta = \alpha_0 + \sum_{j=1}^p \alpha_j (|r_{t-j}| - \gamma_j r_{t-j})^\delta + \sum_{j=1}^q \beta_j \sigma_{t-j}^\delta. \quad (5.53)$$

Note that the model is GARCH when $\delta = 2$ and $\gamma_j = 0$, for $j \in \{1, \dots, p\}$. The parameters γ_j ($|\gamma_j| \leq 1$) are the *leverage* parameters, which are a measure of asymmetry, and $\delta > 0$ is the parameter for the power term. A positive [negative] value of γ_j 's means that past negative [positive] shocks have a deeper impact on current conditional volatility than past positive [negative] shocks. This model couples the flexibility of a varying exponent with the asymmetry coefficient to take the *leverage effect* into account. Further, to guarantee that $\sigma_t > 0$, we assume that $\alpha_0 > 0$, $\alpha_j \geq 0$ with at least one $\alpha_j > 0$, and $\beta_j \geq 0$.

We continue the analysis of the DJIA returns in the following example.

Example 5.6 APARCH Analysis of the DJIA Returns

The R package `fGarch` was used to fit an AR-APARCH model to the DJIA returns discussed in Example 5.5. As in the previous example, we include an AR(1) in the model to account for the conditional mean. In this case, we may think of the model as $r_t = \mu_t + y_t$ where μ_t is an AR(1), and y_t is APARCH noise with conditional variance modeled as (5.53) with t-errors. A partial output of the analysis is given below. We do not include displays, but we show how to obtain them. The predicted volatility is, of course, different than the values shown in Figure 5.6, but appear similar when graphed.

```
library(xts)
library(fGarch)
summary(djia.ap <- garchFit(~arma(1,0)+aparch(1,1), data=djiar,
  cond.dist='std'))
plot(djia.ap) # to see all plot options (none shown)
      Estimate Std. Error   t value   p.value
mu      5.234e-04  1.525e-04   3.432  0.000598
ar1     -4.818e-02  1.934e-02  -2.491  0.012727
omega    1.798e-04  3.443e-05   5.222  1.77e-07
alpha1   9.809e-02  1.030e-02   9.525  < 2e-16
gamma1  1.000e+00  1.045e-02  95.731  < 2e-16
beta1    8.945e-01  1.049e-02  85.280  < 2e-16
delta    1.070e+00  1.350e-01   7.928  2.22e-15
shape    7.286e+00  1.123e+00   6.489  8.61e-11
---
Standardised Residuals Tests:
                               Statistic p-Value
Ljung-Box Test      R      Q(10) 15.71403 0.108116
Ljung-Box Test      R^2     Q(10) 16.87473 0.077182
```

In most applications, the distribution of the noise, ϵ_t in (5.36), is rarely normal. The R package `fGarch` allows for various distributions to be fit to the data; see the help file for information. Some drawbacks of GARCH and related models are as follows. (i) The GARCH model assumes positive and negative returns have the same effect because volatility depends on squared returns; the asymmetric models help alleviate this problem. (ii) These models are often restrictive because of the tight constraints on the model parameters (e.g., for an ARCH(1), $0 \leq \alpha_1^2 < \frac{1}{3}$). (iii) The likelihood is

flat unless n is very large. (iv) The models tend to overpredict volatility because they respond slowly to large isolated returns.

Various extensions to the original model have been proposed to overcome some of the shortcomings we have just mentioned. For example, we have already discussed the fact that `fGarch` allows for asymmetric return dynamics. In the case of persistence in volatility, the integrated GARCH (IGARCH) model may be used. Recall (5.50) where we showed the GARCH(1, 1) model can be written as

$$r_t^2 = \alpha_0 + (\alpha_1 + \beta_1)r_{t-1}^2 + v_t - \beta_1 v_{t-1}$$

and r_t^2 is stationary if $\alpha_1 + \beta_1 < 1$. The IGARCH model sets $\alpha_1 + \beta_1 = 1$, in which case the IGARCH(1, 1) model is

$$r_t = \sigma_t \epsilon_t \quad \text{and} \quad \sigma_t^2 = \alpha_0 + (1 - \beta_1)r_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

There are many different extensions to the basic ARCH model that were developed to handle the various situations noticed in practice. Interested readers might find the general discussions in Engle et al. (1994) and Shephard (1996) worthwhile reading. Also, Gouriéroux (1997) gives a detailed presentation of ARCH and related models with financial applications and contains an extensive bibliography. Two excellent texts on financial time series analysis are Chan (2002) and Tsay (2002).

Finally, we briefly discuss *stochastic volatility models*; a detailed treatment of these models is given in Chapter 6. The volatility component, σ_t^2 , in GARCH and related models are conditionally nonstochastic. For example, in the ARCH(1) model, any time the previous return is valued at, say c , i.e., $r_{t-1} = c$, it must be the case that $\sigma_t^2 = \alpha_0 + \alpha_1 c^2$. This assumption seems a bit unrealistic. The stochastic volatility model adds a stochastic component to the volatility in the following way. In the GARCH model, a return, say r_t , is

$$r_t = \sigma_t \epsilon_t \quad \Rightarrow \quad \log r_t^2 = \log \sigma_t^2 + \log \epsilon_t^2. \quad (5.54)$$

Thus, the observations $\log r_t^2$ are generated by two components, the unobserved volatility, $\log \sigma_t^2$, and the unobserved noise, $\log \epsilon_t^2$. While, for example, GARCH(1, 1) models volatility without error, $\sigma_{t+1}^2 = \alpha_0 + \alpha_1 r_t^2 + \beta_1 \sigma_t^2$, the basic stochastic volatility model assumes the logged latent variable is an autoregressive process,

$$\log \sigma_{t+1}^2 = \phi_0 + \phi_1 \log \sigma_t^2 + w_t \quad (5.55)$$

where $w_t \sim \text{iid } N(0, \sigma_w^2)$. The introduction of the noise term w_t makes the latent volatility process stochastic. Together (5.54) and (5.55) comprise the stochastic volatility model. Given n observations, the goals are to estimate the parameters ϕ_0 , ϕ_1 and σ_w^2 , and then predict future volatility. Details are provided in Section 6.11.

5.4 Threshold Models

In Section 3.4 we discussed the fact that, for a stationary time series, best linear prediction forward in time is the same as best linear prediction backward in time.

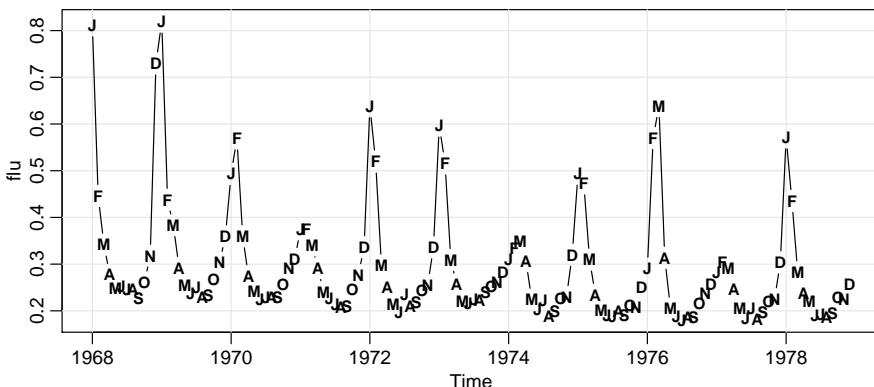


Fig. 5.7. U.S. monthly pneumonia and influenza deaths per 10,000.

This result followed from the fact that the variance–covariance matrix of $x_{1:n} = \{x_1, x_2, \dots, x_n\}$, say, $\Gamma = \{\gamma(i-j)\}_{i,j=1}^n$, is the same as the variance–covariance matrix of $x_{n:1} = \{x_n, x_{n-1}, \dots, x_1\}$. In addition, if the process is Gaussian, the distributions of $x_{1:n}$ and $x_{n:1}$ are identical. In this case, a time plot of $x_{1:n}$ (that is, the data plotted forward in time) should look similar to a time plot of $x_{n:1}$ (that is, the data plotted backward in time).

There are, however, many series that do not fit into this category. For example, Figure 5.7 shows a plot of monthly pneumonia and influenza deaths per 10,000 in the U.S. for 11 years, 1968 to 1978. Typically, the number of deaths tends to increase faster than it decreases ($\uparrow\downarrow$), especially during epidemics. Thus, if the data were plotted backward in time, that series would tend to increase slower than it decreases. Also, if monthly pneumonia and influenza deaths followed a linear Gaussian process, we would not expect to see such large bursts of positive and negative changes that occur periodically in this series. Moreover, although the number of deaths is typically largest during the winter months, the data are not perfectly seasonal. That is, although the peak of the series often occurs in January, in other years, the peak occurs in February or in March. Hence, seasonal ARMA models would not capture this behavior.

Many approaches to modeling nonlinear series exist that could be used (see Priestley, 1988); here, we focus on the class of *threshold models* (TARMA) presented in Tong (1983, 1990). The basic idea of these models is that of fitting local linear ARMA models, and their appeal is that we can use the intuition from fitting global linear ARMA models. For example, a k -regimes self-exciting threshold (SETARMA) model has the form

$$x_t = \begin{cases} \phi_0^{(1)} + \sum_{i=1}^{p_1} \phi_i^{(1)} x_{t-i} + w_t^{(1)} + \sum_{j=1}^{q_1} \theta_j^{(1)} w_{t-j}^{(1)} & \text{if } x_{t-d} \leq r_1, \\ \phi_0^{(2)} + \sum_{i=1}^{p_2} \phi_i^{(2)} x_{t-i} + w_t^{(2)} + \sum_{j=1}^{q_2} \theta_j^{(2)} w_{t-j}^{(2)} & \text{if } r_1 < x_{t-d} \leq r_2, \\ \vdots & \vdots \\ \phi_0^{(k)} + \sum_{i=1}^{p_k} \phi_i^{(k)} x_{t-i} + w_t^{(k)} + \sum_{j=1}^{q_k} \theta_j^{(k)} w_{t-j}^{(k)} & \text{if } r_{k-1} < x_{t-d}, \end{cases} \quad (5.56)$$

where $w_t^{(j)} \sim \text{iid } N(0, \sigma_j^2)$, for $j = 1, \dots, k$, the positive integer d is a specified em delay, and $-\infty < r_1 < \dots < r_{k-1} < \infty$ is a partition of \mathbb{R} .

These models allow for changes in the ARMA coefficients over time, and those changes are determined by comparing previous values (back-shifted by a time lag equal to d) to fixed threshold values. Each different ARMA model is referred to as a *regime*. In the definition above, the values (p_j, q_j) of the order of ARMA models can differ in each regime, although in many applications, they are equal. Stationarity and invertibility are obvious concerns when fitting time series models. For the threshold time series models, such as TAR, TMA and TARMA models, however, the stationary and invertible conditions in the literature are less well-known in general and often restricted models of order one.

The model can be generalized to include the possibility that the regimes depend on a collection of the past values of the process, or that the regimes depend on an exogenous variable (in which case the model is not self-exciting) such in predator-prey cases. For example, Canadian lynx have been thoroughly studied (see the R data set [lynx](#)) and the series is typically used to demonstrate the fitting of threshold models. The lynx prey varies from small rodents to deer, with the Snowshoe Hare being its overwhelmingly favored prey. In fact, in certain areas the lynx is so closely tied to the Snowshoe that its population rises and falls with that of the hare, even though other food sources may be abundant. In this case, it seems reasonable to replace x_{t-d} in (5.56) with say y_{t-d} , where y_t is the size of the Snowshoe Hare population.

The popularity of TAR models is due to their being relatively simple to specify, estimate, and interpret as compared to many other nonlinear time series models. In addition, despite its apparent simplicity, the class of TAR models can reproduce many nonlinear phenomena. In the following example, we use these methods to fit a threshold model to monthly pneumonia and influenza deaths series previously mentioned.

Example 5.7 Threshold Modeling of the Influenza Series

As previously discussed, examination of [Figure 5.7](#) leads us to believe that the monthly pneumonia and influenza deaths time series, say flu_t , is not linear. It is also evident from [Figure 5.7](#) that there is a slight negative trend in the data. We have found that the most convenient way to fit a threshold model to these data, while removing the trend, is to work with the first differences. The differenced data, say

$$x_t = \text{flu}_t - \text{flu}_{t-1}$$

is exhibited in [Figure 5.9](#) as points (+) representing the observations.

The nonlinearity of the data is more pronounced in the plot of the first differences, x_t . Clearly x_t slowly rises for some months and, then, sometime in the winter, has a possibility of jumping to a large number once x_t exceeds about .05. If the process does make a large jump, then a subsequent significant decrease occurs in x_t . Another telling graphic is the lag plot of x_t versus x_{t-1} shown in [Figure 5.8](#), which suggests the possibility of two linear regimes based on whether or not x_{t-1} exceeds .05.

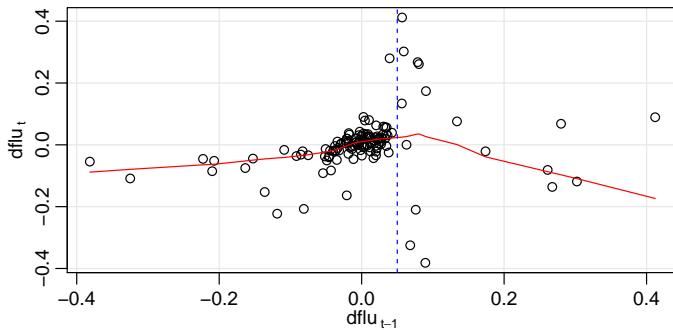


Fig. 5.8. Scatterplot of $d\text{flu}_t = \text{flu}_t - \text{flu}_{t-1}$ versus $d\text{flu}_{t-1}$ with a lowess fit superimposed (line). A vertical dashed line indicates $d\text{flu}_{t-1} = .05$.

As an initial analysis, we fit the following threshold model

$$\begin{aligned} x_t &= \alpha^{(1)} + \sum_{j=1}^p \phi_j^{(1)} x_{t-j} + w_t^{(1)}, & x_{t-1} < .05; \\ x_t &= \alpha^{(2)} + \sum_{j=1}^p \phi_j^{(2)} x_{t-j} + w_t^{(2)}, & x_{t-1} \geq .05, \end{aligned} \quad (5.57)$$

with $p = 6$, assuming this would be larger than necessary. Model (5.57) is easy to fit using two linear regression runs, one when $x_{t-1} < .05$ and the other when $x_{t-1} \geq .05$. Details are provided in the R code at the end of this example.

An order $p = 4$ was finally selected and the fit was

$$\begin{aligned} \hat{x}_t &= 0 + .51_{(.08)} x_{t-1} - .20_{(.06)} x_{t-2} + .12_{(.05)} x_{t-3} \\ &\quad - .11_{(.05)} x_{t-4} + \hat{w}_t^{(1)}, \quad \text{for } x_{t-1} < .05; \\ \hat{x}_t &= .40 - .75_{(.17)} x_{t-1} - 1.03_{(.21)} x_{t-2} - 2.05_{(.105)} x_{t-3} \\ &\quad - 6.71_{(.125)} x_{t-4} + \hat{w}_t^{(2)}, \quad \text{for } x_{t-1} \geq .05, \end{aligned}$$

where $\hat{\sigma}_1 = .05$ and $\hat{\sigma}_2 = .07$. The threshold of $.05$ was exceeded 17 times.

Using the final model, one-month-ahead predictions can be made, and these are shown in Figure 5.9 as a line. The model does extremely well at predicting a flu epidemic; the peak at 1976, however, was missed by this model. When we fit a model with a smaller threshold of $.04$, flu epidemics were somewhat underestimated, but the flu epidemic in the eighth year was predicted one month early. We chose the model with a threshold of $.05$ because the residual diagnostics showed no obvious departure from the model assumption (except for one outlier at 1976); the model with a threshold of $.04$ still had some correlation left in the residuals and there was more than one outlier. Finally, prediction beyond one-month-ahead for this model is complicated, but some approximate techniques exist (see Tong, 1983). The following commands can be used to perform this analysis in R.

```

# Plot data with month initials as points
plot(flu, type="c")
Months = c("J","F","M","A","M","J","J","A","S","O","N","D")
points(flu, pch=Months, cex=.8, font=2)
# Start analysis
dflu = diff(flu)
lag1.plot(dflu, corr=FALSE) # scatterplot with lowess fit
thrsh = .05 # threshold
Z = ts.intersect(dflu, lag(dflu,-1), lag(dflu, 2), lag(dflu, -3),
                 lag(dflu,-4) )
ind1 = ifelse(Z[,2] < thrsh, 1, NA) # indicator < thrsh
ind2 = ifelse(Z[,2] < thrsh, NA, 1) # indicator >= thrsh
X1 = Z[,1]*ind1
X2 = Z[,1]*ind2
summary(fit1 <- lm(X1~ Z[,2:5])) # case 1
summary(fit2 <- lm(X2~ Z[,2:5])) # case 2
D = cbind(rep(1, nrow(Z)), Z[,2:5]) # design matrix
p1 = D %*% coef(fit1) # get predictions
p2 = D %*% coef(fit2)
prd = ifelse(Z[,2] < thrsh, p1, p2)
plot(dflu, ylim=c(-.5,.5), type='p', pch=3)
lines(prd)
prde1 = sqrt(sum(resid(fit1)^2)/df.residual(fit1) )
prde2 = sqrt(sum(resid(fit2)^2)/df.residual(fit2) )
prde = ifelse(Z[,2] < thrsh, prde1, prde2)
tx = time(dflu)[-1:4]
xx = c(tx, rev(tx))
yy = c(prd-2*prde, rev(prd+2*prde))
polygon(xx, yy, border=8, col=gray(.6, alpha=.25) )
abline(h=.05, col=4, lty=6)

```

Finally, we note that there is an R package called `tsDyn` that can be used to fit these models; we assume `dflu` already exists.

```

library(tsDyn) # load package - install it if you don't have it
# vignette("tsDyn") # for package details
(u = setar(dflu, m=4, thDelay=0, th=.05)) # fit model and view results
(u = setar(dflu, m=4, thDelay=0)) # let program fit threshold (= .036)
BIC(u); AIC(u) # if you want to try other models; m=3 works well too
plot(u) # graphics - ?plot.setar for information

```

The threshold found here is .036, which includes a few more observations than using .04, but suffers from the same drawbacks previously noted.

5.5 Lagged Regression and Transfer Function Modeling

In [Section 4.8](#), we considered lagged regression in a frequency domain approach based on coherency. For example, consider the SOI and Recruitment series that were analyzed in [Example 4.24](#); the series are displayed in [Figure 1.5](#). In that example, the interest was in predicting the output Recruitment series, say, y_t , from the input SOI, say x_t . We considered the lagged regression model

$$y_t = \sum_{j=0}^{\infty} \alpha_j x_{t-j} + \eta_t = \alpha(B)x_t + \eta_t, \quad (5.58)$$

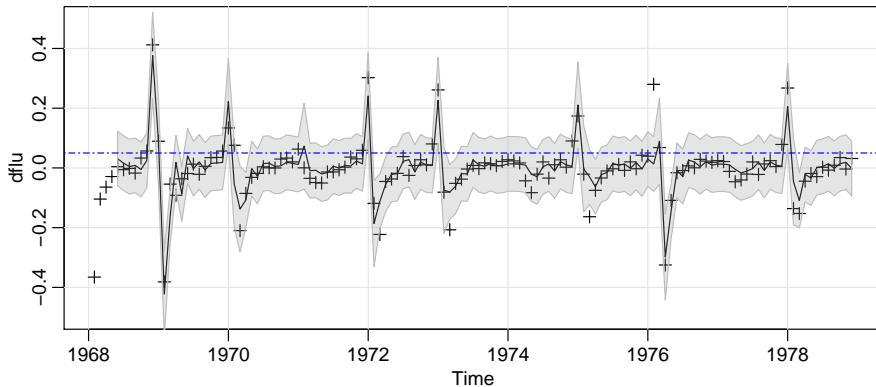


Fig. 5.9. First differenced U.S. monthly pneumonia and influenza deaths (+); one-month-ahead predictions (solid line) with ± 2 prediction error bounds . The horizontal line is the threshold.

where $\sum_j |\alpha_j| < \infty$. We assume the input process x_t and noise process η_t in (5.58) are both stationary and mutually independent. The coefficients $\alpha_0, \alpha_1, \dots$ describe the weights assigned to past values of x_t used in predicting y_t and we have used the notation

$$\alpha(B) = \sum_{j=0}^{\infty} \alpha_j B^j. \quad (5.59)$$

In the Box and Jenkins (1970) formulation, we assign ARIMA models, say, ARIMA(p, d, q) and ARIMA(p_η, d_η, q_η), to the series x_t and η_t , respectively. In Section 4.8, we assumed the noise, η_t , was white. The components of (5.58) in backshift notation, for the case of simple ARMA(p, q) modeling of the input and noise, would have the representation

$$\phi(B)x_t = \theta(B)w_t \quad (5.60)$$

and

$$\phi_\eta(B)\eta_t = \theta_\eta(B)z_t, \quad (5.61)$$

where w_t and z_t are independent white noise processes with variances σ_w^2 and σ_z^2 , respectively. Box and Jenkins (1970) proposed that systematic patterns often observed in the coefficients α_j , for $j = 1, 2, \dots$, could often be expressed as a ratio of polynomials involving a small number of coefficients, along with a specified delay, d , so

$$\alpha(B) = \frac{\delta(B)B^d}{\omega(B)}, \quad (5.62)$$

where

$$\omega(B) = 1 - \omega_1 B - \omega_2 B^2 - \cdots - \omega_r B^r \quad (5.63)$$

and

$$\delta(B) = \delta_0 + \delta_1 B + \cdots + \delta_s B^s \quad (5.64)$$

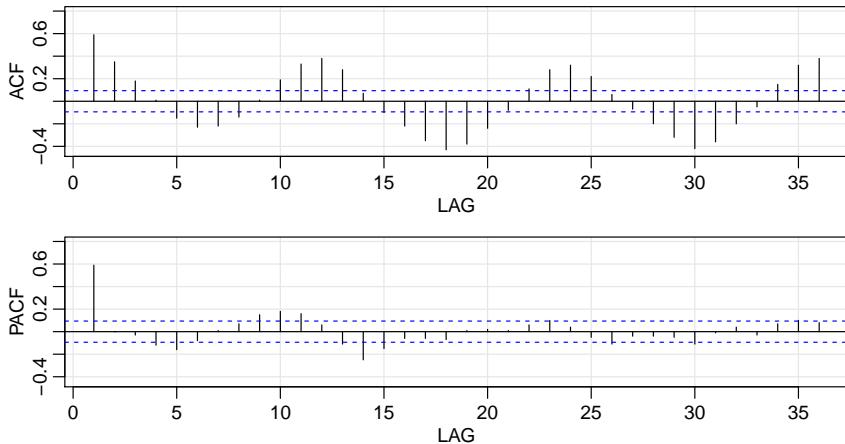


Fig. 5.10. Sample ACF and PACF of detrended SOI.

are the indicated operators; in this section, we find it convenient to represent the inverse of an operator, say, $\omega(B)^{-1}$, as $1/\omega(B)$.

Determining a parsimonious model involving a simple form for $\alpha(B)$ and estimating all of the parameters in the above model are the main tasks in the transfer function methodology. Because of the large number of parameters, it is necessary to develop a sequential methodology. Suppose we focus first on finding the ARIMA model for the input x_t and apply this operator to both sides of (5.58), obtaining the new model

$$\tilde{y}_t = \frac{\phi(B)}{\theta(B)} y_t = \alpha(B) \frac{\phi(B)}{\theta(B)} x_t + \frac{\phi(B)}{\theta(B)} \eta_t = \alpha(B) w_t + \tilde{\eta}_t,$$

where w_t and the transformed noise $\tilde{\eta}_t$ are independent.

The series w_t is a *prewhitened* version of the input series, and its cross-correlation with the transformed output series \tilde{y}_t will be just

$$\gamma_{\tilde{y}w}(h) = E[\tilde{y}_{t+h} w_t] = E \left[\sum_{j=0}^{\infty} \alpha_j w_{t+h-j} w_t \right] = \sigma_w^2 \alpha_h, \quad (5.65)$$

because the autocovariance function of white noise will be zero except when $j = h$ in (5.65). Hence, by computing the cross-correlation between the prewhitened input series and the transformed output series should yield a rough estimate of the behavior of $\alpha(B)$.

Example 5.8 Relating the Prewhitened SOI to the Transformed Recruitment Series

We give a simple example of the suggested procedure for the SOI and the Recruitment series. Figure 5.10 shows the sample ACF and PACF of the detrended SOI, and it is clear, from the PACF, that an autoregressive series with $p = 1$ will do a

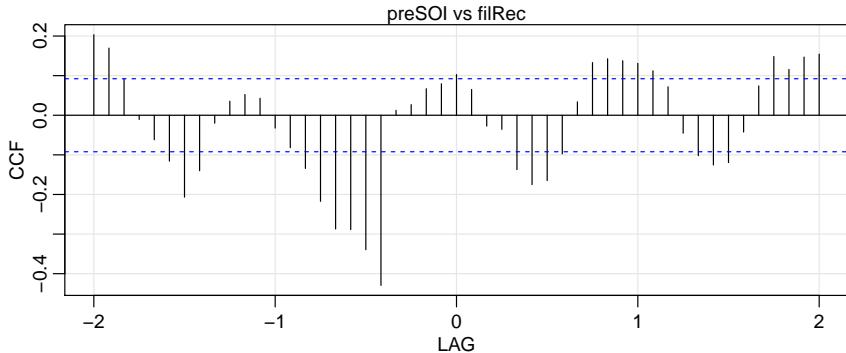


Fig. 5.11. Sample CCF of the prewhitened, detrended SOI and the similarly transformed Recruitment series; negative lags indicate that SOI leads Recruitment.

reasonable job. Fitting the series gave $\hat{\phi} = .588$ with $\hat{\sigma}_w^2 = .092$, and we applied the operator $(1 - .588B)$ to both x_t and y_t and computed the cross-correlation function, which is shown in Figure 5.11. Noting the apparent shift of $d = 5$ months and the decrease thereafter, it seems plausible to hypothesize a model of the form

$$\alpha(B) = \delta_0 B^5 (1 + \omega_1 B + \omega_1^2 B^2 + \dots) = \frac{\delta_0 B^5}{1 - \omega_1 B}$$

for the transfer function. In this case, we would expect ω_1 to be negative. The following R code was used for this example.

```
soi.d = resid(lm(soi~time(soi), na.action=NULL)) # detrended SOI
acf2(soi.d)
fit = arima(soi.d, order=c(1,0,0))
ar1 = as.numeric(coef(fit)[1])      # = 0.5875
soi.pw = resid(fit)
rec.fil = filter(rec, filter=c(1, -ar1), sides=1)
ccf(soi.pw, rec.fil, ylab="CCF", na.action=na.omit, panel.first=grid())
```

In the code above, `soi.pw` is the prewhitened detrended SOI series, and `rec.fil` is the filtered Recruitment series.

In some cases, we may postulate the form of the separate components $\delta(B)$ and $\omega(B)$, so we might write the equation

$$y_t = \frac{\delta(B)B^d}{\omega(B)}x_t + \eta_t$$

as

$$\omega(B)y_t = \delta(B)B^d x_t + \omega(B)\eta_t,$$

or in regression form

$$y_t = \sum_{k=1}^r \omega_k y_{t-k} + \sum_{k=0}^s \delta_k x_{t-d-k} + u_t, \quad (5.66)$$

where

$$u_t = \omega(B)\eta_t. \quad (5.67)$$

Once we have (5.66), it will be easy to fit the model if we forget about η_t and allow u_t to have any ARMA behavior. We illustrate this technique in the following example.

Example 5.9 Transfer Function Model for SOI and Recruitment

We illustrate the procedure for fitting a lagged regression model of the form suggested in Example 5.8 to the detrended SOI series (x_t) and the Recruitment series (y_t). The results reported here are practically the same as the the results obtained from the frequency domain approach used in Example 4.24.

Based on Example 5.8, we have determined that

$$y_t = \alpha + \omega_1 y_{t-1} + \delta_0 x_{t-5} + u_t$$

is a reasonable model. At this point, we simply run the regression allowing for autocorrelated errors based on the techniques discussed in Section 3.8. Based on these techniques, the fitted model is the same as the one obtained in Example 4.24, namely,

$$y_t = 12 + .8y_{t-1} - 21x_{t-5} + u_t, \quad \text{and} \quad u_t = .45u_{t-1} + w_t,$$

where w_t is white noise with $\sigma_w^2 = 50$.

Figure 5.12 displays the ACF and PACF of the estimated noise u_t , showing that an AR(1) is appropriate. In addition, the figure displays the Recruitment series and the one-step-ahead predictions based on the final model. The following R code was used for this example.

```
soi.d = resid(lm(soi~time(soi), na.action=NULL))
fish = ts.intersect(rec, RL1=lag(rec,-1), SL5=lag(soi.d,-5))
(u = lm(fish[,1]~fish[,2:3], na.action=NULL))
acf2(resid(u)) # suggests ar1
(arx = sarima(fish[,1], 1, 0, 0, xreg=fish[,2:3])) # final model
Coefficients:
ar1 intercept RL1 SL5
0.4487 12.3323 0.8005 -21.0307
s.e. 0.0503 1.5746 0.0234 1.0915
sigma^2 estimated as 49.93
pred = rec + resid(arx$fit) # 1-step-ahead predictions
ts.plot(pred, rec, col=c('gray90',1), lwd=c(7,1))
```

For completeness, we finish the discussion of the more complicated Box-Jenkins method for fitting transfer function models. We note, however, that the method has no recognizable overall optimality, and is not generally better or worse than the method previously discussed.

The form of (5.66) suggests doing a regression on the lagged versions of both the input and output series to obtain $\hat{\beta}$, the estimate of the $(r+s+1) \times 1$ regression vector

$$\beta = (\omega_1, \dots, \omega_r, \delta_0, \delta_1, \dots, \delta_s)'.$$

The residuals from the regression, say,

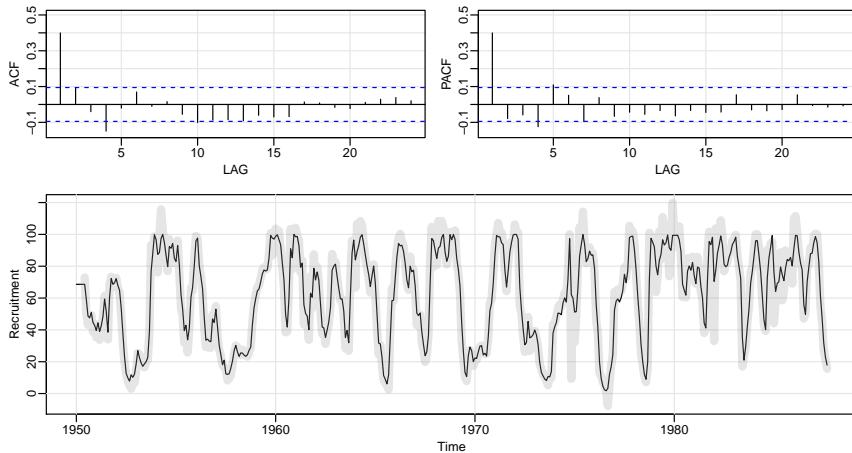


Fig. 5.12. Top: ACF and PACF of the estimated noise u_t . Bottom: The recruitment series (line) and the one-step-ahead predictions (gray swatch) based on the final transfer function model.

$$\hat{u}_t = y_t - \hat{\beta}' z_t,$$

where

$$z_t = (y_{t-1}, \dots, y_{t-r}, x_{t-d}, \dots, x_{t-d-s})'$$

denotes the usual vector of independent variables, could be used to approximate the best ARMA model for the noise process η_t , because we can compute an estimator for that process from (5.67), using \hat{u}_t and $\hat{\omega}(B)$ and applying the moving average operator to get $\hat{\eta}_t$. Fitting an ARMA(p_η, q_η) model to the this estimated noise then completes the specification. The preceding suggests the following sequential procedure for fitting the transfer function model to data.

- (i) Fit an ARMA model to the input series x_t to estimate the parameters ϕ_1, \dots, ϕ_p , $\theta_1, \dots, \theta_q$, σ_w^2 in the specification (5.60). Retain ARMA coefficients for use in step (ii) and the fitted residuals \hat{w}_t for use in Step (iii).
- (ii) Apply the operator determined in step (i), that is,

$$\hat{\phi}(B)y_t = \hat{\theta}(B)\tilde{y}_t,$$

to determine the transformed output series \tilde{y}_t .

- (iii) Use the cross-correlation function between \tilde{y}_t and \hat{w}_t in steps (i) and (ii) to suggest a form for the components of the polynomial

$$\alpha(B) = \frac{\delta(B)B^d}{\omega(B)}$$

and the estimated time delay d .

- (iv) Obtain $\hat{\beta} = (\hat{\omega}_1, \dots, \hat{\omega}_r, \hat{\delta}_0, \hat{\delta}_1, \dots, \hat{\delta}_s)$ by fitting a linear regression of the form (5.66). Retain the residuals \hat{u}_t for use in step (v).

- (v) Apply the moving average transformation (5.67) to the residuals \hat{u}_t to find the noise series $\hat{\eta}_t$, and fit an ARMA model to the noise, obtaining the estimated coefficients in $\hat{\phi}_\eta(B)$ and $\hat{\theta}_\eta(B)$.

The above procedure is fairly reasonable, but as previously mentioned, is not optimal in any sense. Simultaneous least squares estimation, based on the observed x_t and y_t , can be accomplished by noting that the transfer function model can be written as

$$y_t = \frac{\delta(B)B^d}{\omega(B)}x_t + \frac{\theta_\eta(B)}{\phi_\eta(B)}z_t,$$

which can be put in the form

$$\omega(B)\phi_\eta(B)y_t = \phi_\eta(B)\delta(B)B^d x_t + \omega(B)\theta_\eta(B)z_t, \quad (5.68)$$

and it is clear that we may use least squares to minimize $\sum_t z_t^2$, as in earlier sections. In Example 5.9, we simply allowed $u_t \frac{\theta_\eta(B)}{\phi_\eta(B)}z_t$ in (5.68) to have any ARMA structure. Finally, we mention that we may also express the transfer function in state-space form as an ARMAX model; see Section 5.6 and Section 6.6.1.

5.6 Multivariate ARMAX Models

To understand multivariate time series models and their capabilities, we first present an introduction to multivariate time series regression techniques. Since all processes are vector processes, we suspend the use of boldface for vectors. A useful extension of the basic univariate regression model presented in Section 2.1 is the case in which we have more than one output series, that is, *multivariate regression analysis*. Suppose, instead of a single output variable y_t , a collection of k output variables $y_{t1}, y_{t2}, \dots, y_{tk}$ exist that are related to the inputs as

$$y_{ti} = \beta_{i1}z_{t1} + \beta_{i2}z_{t2} + \dots + \beta_{ir}z_{tr} + w_{ti} \quad (5.69)$$

for each of the $i = 1, 2, \dots, k$ output variables. We assume the w_{ti} variables are correlated over the variable identifier i , but are still independent over time. Formally, we assume $\text{cov}\{w_{si}, w_{tj}\} = \sigma_{ij}$ for $s = t$ and is zero otherwise. Then, writing (5.69) in matrix notation, with $y_t = (y_{t1}, y_{t2}, \dots, y_{tk})'$ being the vector of outputs, and $\mathcal{B} = \{\beta_{ij}\}, i = 1, \dots, k, j = 1, \dots, r$ being a $k \times r$ matrix containing the regression coefficients, leads to the simple looking form

$$y_t = \mathcal{B}z_t + w_t. \quad (5.70)$$

Here, the $k \times 1$ vector process w_t is assumed to be a collection of independent vectors with common covariance matrix $E\{w_t w_t'\} = \Sigma_w$, the $k \times k$ matrix containing the covariances σ_{ij} . Under the assumption of normality, the maximum likelihood estimator for the regression matrix is

$$\hat{\mathcal{B}} = \left(\sum_{t=1}^n y_t z_t' \right) \left(\sum_{t=1}^n z_t z_t' \right)^{-1}. \quad (5.71)$$

The error covariance matrix Σ_w is estimated by

$$\hat{\Sigma}_w = \frac{1}{n-r} \sum_{t=1}^n (y_t - \hat{\mathcal{B}} z_t)(y_t - \hat{\mathcal{B}} z_t)'. \quad (5.72)$$

The uncertainty in the estimators can be evaluated from

$$\text{se}(\hat{\beta}_{ij}) = \sqrt{c_{ii} \hat{\sigma}_{jj}}, \quad (5.73)$$

for $i = 1, \dots, r$, $j = 1, \dots, k$, where se denotes estimated standard error, $\hat{\sigma}_{jj}$ is the j -th diagonal element of $\hat{\Sigma}_w$, and c_{ii} is the i -th diagonal element of $(\sum_{t=1}^n z_t z_t')^{-1}$.

Also, the information theoretic criterion changes to

$$\text{AIC} = \ln |\hat{\Sigma}_w| + \frac{2}{n} \left(kr + \frac{k(k+1)}{2} \right). \quad (5.74)$$

and BIC replaces the second term in (5.74) by $K \ln n/n$ where $K = kr + k(k+1)/2$. Bedrick and Tsai (1994) have given a corrected form for AIC in the multivariate case as

$$\text{AICc} = \ln |\hat{\Sigma}_w| + \frac{k(r+n)}{n-k-r-1}. \quad (5.75)$$

Many data sets involve more than one time series, and we are often interested in the possible dynamics relating all series. In this situation, we are interested in modeling and forecasting $k \times 1$ vector-valued time series $x_t = (x_{t1}, \dots, x_{tk})'$, $t = 0, \pm 1, \pm 2, \dots$. Unfortunately, extending univariate ARMA models to the multivariate case is not so simple. The multivariate autoregressive model, however, is a straight-forward extension of the univariate AR model.

For the first-order *vector autoregressive model*, VAR(1), we take

$$x_t = \alpha + \Phi x_{t-1} + w_t, \quad (5.76)$$

where Φ is a $k \times k$ *transition matrix* that expresses the dependence of x_t on x_{t-1} . The *vector white noise* process w_t is assumed to be multivariate normal with mean-zero and covariance matrix

$$\text{E}(w_t w_t') = \Sigma_w. \quad (5.77)$$

The vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)'$ appears as the constant in the regression setting. If $E(x_t) = \mu$, then $\alpha = (I - \Phi)\mu$.

Note the similarity between the VAR model and the multivariate linear regression model (5.70). The regression formulas carry over, and we can, on observing x_1, \dots, x_n , set up the model (5.76) with $y_t = x_t$, $\mathcal{B} = (\alpha, \Phi)$ and $z_t = (1, x'_{t-1})'$. Then, write the solution as (5.71) with the conditional maximum likelihood estimator for the covariance matrix given by

$$\hat{\Sigma}_w = (n-1)^{-1} \sum_{t=2}^n (x_t - \hat{\alpha} - \hat{\Phi}x_{t-1})(x_t - \hat{\alpha} - \hat{\Phi}x_{t-1})'. \quad (5.78)$$

The special form assumed for the constant component, α , of the vector AR model in (5.76) can be generalized to include a fixed $r \times 1$ vector of inputs, u_t . That is, we could have proposed the *vector ARX model*,

$$x_t = \Gamma u_t + \sum_{j=1}^p \Phi_j x_{t-j} + w_t, \quad (5.79)$$

where Γ is a $p \times r$ parameter matrix. The X in ARX refers to the exogenous vector process we have denoted here by u_t . The introduction of exogenous variables through replacing α by Γu_t does not present any special problems in making inferences and we will often drop the X for being superfluous.

Example 5.10 Pollution, Weather, and Mortality

For example, for the three-dimensional series composed of cardiovascular mortality x_{t1} , temperature x_{t2} , and particulate levels x_{t3} , introduced in Example 2.2, take $x_t = (x_{t1}, x_{t2}, x_{t3})'$ as a vector of dimension $k = 3$. We might envision dynamic relations among the three series defined as the first order relation,

$$x_{t1} = \alpha_1 + \beta_1 t + \phi_{11} x_{t-1,1} + \phi_{12} x_{t-1,2} + \phi_{13} x_{t-1,3} + w_{t1},$$

which expresses the current value of mortality as a linear combination of trend and its immediate past value and the past values of temperature and particulate levels. Similarly,

$$x_{t2} = \alpha_2 + \beta_2 t + \phi_{21} x_{t-1,1} + \phi_{22} x_{t-1,2} + \phi_{23} x_{t-1,3} + w_{t2}$$

and

$$x_{t3} = \alpha_3 + \beta_3 t + \phi_{31} x_{t-1,1} + \phi_{32} x_{t-1,2} + \phi_{33} x_{t-1,3} + w_{t3}$$

express the dependence of temperature and particulate levels on the other series. Of course, methods for the preliminary identification of these models exist, and we will discuss these methods shortly. The model in the form of (5.79) is

$$x_t = \Gamma u_t + \Phi x_{t-1} + w_t,$$

where, in obvious notation, $\Gamma = [\alpha | \beta]$ is 3×2 and $u_t = (1, t)'$ is 2×1 .

Throughout much of this section we will use the R package `vars` to fit vector AR models via least squares. For this particular example, we have (partial output shown):

```
library(vars)
x = cbind(cmort, temp, part)
summary(VAR(x, p=1, type='both'))      # 'both' fits constant + trend
Estimation results for equation cmort: # other equations not shown
cmort = cmort.l1 + temp.l1 + part.l1 + const + trend
    Estimate Std. Error t value p.value
```

```

cmort.11  0.464824  0.036729  12.656 < 2e-16
tempr.11 -0.360888  0.032188 -11.212 < 2e-16
part.11   0.099415  0.019178   5.184 3.16e-07
const     73.227292  4.834004  15.148 < 2e-16
trend    -0.014459  0.001978  -7.308 1.07e-12
--
Residual standard error: 5.583 on 502 degrees of freedom
Multiple R-Squared: 0.6908, Adjusted R-squared: 0.6883
F-statistic: 280.3 on 4 and 502 DF, p-value: < 2.2e-16

Covariance matrix of residuals: Correlation matrix of residuals:
      cmort  tempr  part      cmort  tempr  part
cmort  31.172  5.975  16.65  cmort  1.0000  0.1672  0.2484
tempr   5.975  40.965  42.32  tempr  0.1672  1.0000  0.5506
part    16.654  42.323 144.26  part  0.2484  0.5506  1.0000

```

For this particular case, we obtain

$$\hat{\alpha} = (73.23, 67.59, 67.46)', \quad \hat{\beta} = (-0.014, -0.007, -0.005)',$$

$$\hat{\Phi} = \begin{pmatrix} .46(.04) & -.36(.03) & .10(.02) \\ -.24(.04) & .49(.04) & -.13(.02) \\ -.12(.08) & -.48(.07) & .58(.04) \end{pmatrix}, \quad \hat{\Sigma}_w = \begin{pmatrix} 31.17 & 5.98 & 16.65 \\ 5.98 & 40.965 & 42.32 \\ 16.65 & 42.32 & 144.26 \end{pmatrix}$$

where the standard errors, computed as in (5.73), are given in parentheses.

For the vector $(x_{t1}, x_{t2}, x_{t3}) = (M_t, T_t, P_t)$, with M_t , T_t and P_t denoting mortality, temperature, and particulate level, respectively, we obtain the prediction equation for mortality,

$$\hat{M}_t = 73.23 - .014t + .46M_{t-1} - .36T_{t-1} + .10P_{t-1}.$$

Comparing observed and predicted mortality with this model leads to an R^2 of about .69.

It is easy to extend the VAR(1) process to higher orders, VAR(p). To do this, we use the notation of (5.70) and write the vector of regressors as

$$z_t = (1, x'_{t-1}, x'_{t-2}, \dots x'_{t-p})'$$

and the regression matrix as $\mathcal{B} = (\alpha, \Phi_1, \Phi_2, \dots, \Phi_p)$. Then, this regression model can be written as

$$x_t = \alpha + \sum_{j=1}^p \Phi_j x_{t-j} + w_t \tag{5.80}$$

for $t = p+1, \dots, n$. The $k \times k$ error sum of products matrix becomes

$$\text{SSE} = \sum_{t=p+1}^n (x_t - \mathcal{B}z_t)(x_t - \mathcal{B}z_t)', \tag{5.81}$$

so that the conditional maximum likelihood estimator for the *error covariance matrix* Σ_w is

$$\hat{\Sigma}_w = \text{SSE}/(n - p), \quad (5.82)$$

as in the multivariate regression case, except now only $n - p$ residuals exist in (5.81). For the multivariate case, we have found that the Schwarz criterion

$$\text{BIC} = \log |\hat{\Sigma}_w| + k^2 p \ln n/n, \quad (5.83)$$

gives more reasonable classifications than either AIC or corrected version AICc. The result is consistent with those reported in simulations by Lütkepohl (1985). Of course, estimation via Yule-Walker, unconditional least squares and MLE follow directly from the univariate counterparts.

Example 5.11 Pollution, Weather, and Mortality (cont)

We used the R package first to select a VAR(p) model and then fit the model. The selection criteria used in the package are AIC, Hannan-Quinn (HQ; Hannan & Quinn, 1979), BIC (SC), and Final Prediction Error (FPE). The Hannan-Quinn procedure is similar to BIC, but with $\ln n$ replaced by $2 \ln(\ln(n))$ in the penalty term. FPE finds the model that minimizes the approximate mean squared one-step-ahead prediction error (see Akaike, 1969 for details); it is rarely used.

```
VARselect(x, lag.max=10, type="both")
$selection
  AIC(n)  HQ(n)  SC(n)  FPE(n)
    9      5      2      9
$criteria
  1     2     3     4     5     6     7     8     9     10
AIC(n) 11.738 11.302 11.268 11.230 11.176 11.153 11.152 11.129 11.119 11.120
HQ(n)   11.788 11.381 11.377 11.370 11.346 11.352 11.381 11.388 11.408 11.439
SC(n)   11.865 11.505 11.547 11.585 11.608 11.660 11.736 11.788 11.855 11.932
```

Note that BIC picks the order $p = 2$ model while AIC and FPE pick an order $p = 9$ model and Hannan-Quinn selects an order $p = 5$ model.

Fitting the model selected by BIC we obtain

$$\hat{\alpha} = (56.1, 49.9, 59.6)', \quad \hat{\beta} = (-0.011, -0.005, -0.008)',$$

$$\begin{aligned}\hat{\Phi}_1 &= \begin{pmatrix} .30(.04) & -.20(.04) & .04(.02) \\ -.11(.05) & .26(.05) & -.05(.03) \\ .08(.09) & -.39(.09) & .39(.05) \end{pmatrix}, \\ \hat{\Phi}_2 &= \begin{pmatrix} .28(.04) & -.08(.03) & .07(.03) \\ -.04(.05) & .36(.05) & -.10(.03) \\ -.33(.09) & .05(.09) & .38(.05) \end{pmatrix},\end{aligned}$$

where the standard errors are given in parentheses. The estimate of Σ_w is

$$\hat{\Sigma}_w = \begin{pmatrix} 28.03 & 7.08 & 16.33 \\ 7.08 & 37.63 & 40.88 \\ 16.33 & 40.88 & 123.45 \end{pmatrix}.$$

To fit the model using the `vars` package use the following:

```

summary(fit <- VAR(x, p=2, type="both")) # partial results displayed
cmort = cmort.l1 + tempr.l1 + part.l1 + cmort.l2 + tempr.l2 + part.l2 +
       const + trend

  Estimate Std. Error t value p.value
cmort.l1  0.297059  0.043734  6.792 3.15e-11
tempr.l1 -0.199510  0.044274 -4.506 8.23e-06
part.l1   0.042523  0.024034  1.769 0.07745
cmort.l2  0.276194  0.041938  6.586 1.15e-10
tempr.l2 -0.079337  0.044679 -1.776 0.07639
part.l2   0.068082  0.025286  2.692 0.00733
const     56.098652 5.916618  9.482 < 2e-16
trend    -0.011042  0.001992 -5.543 4.84e-08

Covariance matrix of residuals:
            cmort  tempr  part
cmort  28.034  7.076 16.33
tempr   7.076 37.627 40.88
part   16.325 40.880 123.45

```

Using the notation of the previous example, the prediction model for cardiovascular mortality is estimated to be

$$\hat{M}_t = 56 - .01t + .3M_{t-1} - .2T_{t-1} + .04P_{t-1} + .28M_{t-2} - .08T_{t-2} + .07P_{t-2}.$$

To examine the residuals, we can plot the cross-correlations of the residuals and examine the multivariate version of the Q-test as follows:

```

acf(resid(fit), 52)
serial.test(fit, lags.pt=12, type="PT.adjusted")
  Portmanteau Test (adjusted)
  data: Residuals of VAR object fit
  Chi-squared = 162.3502, df = 90, p-value = 4.602e-06

```

The cross-correlation matrix is shown in [Figure 5.13](#). The figure shows the ACFs of the individual residual series along the diagonal. For example, the first diagonal graph is the ACF of $M_t - \hat{M}_t$, and so on. The off diagonals display the CCFs between pairs of residual series. If the title of the off-diagonal plot is x & y , then y leads in the graphic; that is, on the upper-diagonal, the plot shows `corr[x(t+Lag), y(t)]` whereas in the lower-diagonal, if the title is x & y , you get a plot of `corr[x(t+Lag), y(t)]` (yes, it is the same thing, but the lags are negative in the lower diagonal). The graphic is labeled in a strange way, just remember the second named series is the one that leads. In [Figure 5.13](#) we notice that most of the correlations in the residual series are negligible, however, the zero-order correlation of mortality with temperature residuals is about .22 and mortality with particulate residuals is about .28 (type `acf(resid(fit), 52)$acf`) to see the actual values. This means that the AR model is not capturing the concurrent effect of temperature and pollution on mortality (recall the data evolves over a week). It is possible to fit simultaneous models; see Reinsel (1997) for further details. Thus, not unexpectedly, the Q-test rejects the null hypothesis that the noise is white. The Q-test statistic is given by

$$Q = n^2 \sum_{h=1}^H \frac{1}{n-h} \text{tr} \left[\hat{\Gamma}_w(h) \hat{\Gamma}_w(0)^{-1} \hat{\Gamma}_w(h) \hat{\Gamma}_w(0)^{-1} \right], \quad (5.84)$$

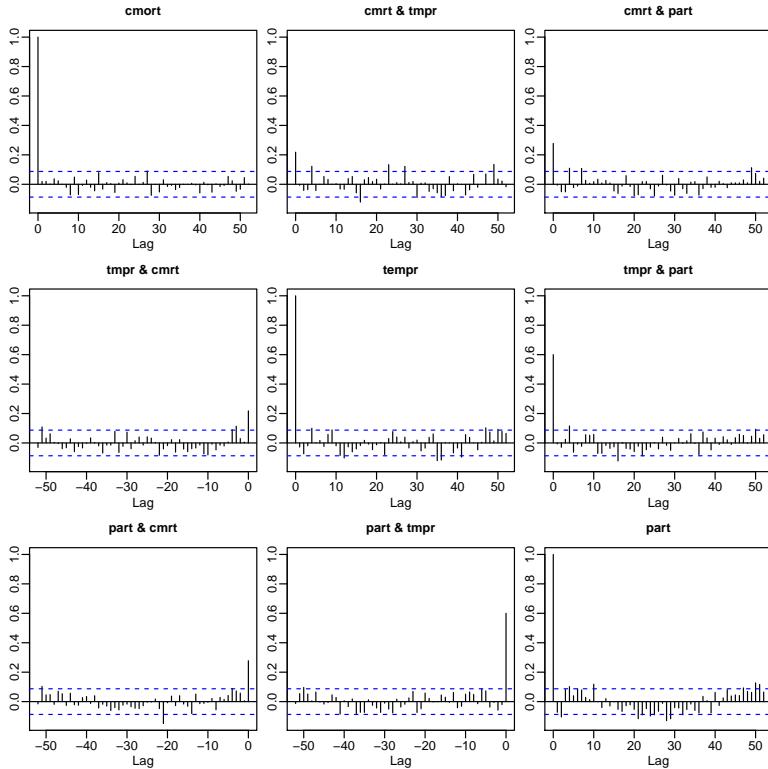


Fig. 5.13. ACFs (diagonals) and CCFs (off-diagonals) for the residuals of the three-dimensional VAR(2) fit to the LA mortality – pollution data set. On the off-diagonals, the second-named series is the one that leads.

where

$$\hat{\Gamma}_w(h) = n^{-1} \sum_{t=1}^{n-h} \hat{w}_{t+h} \hat{w}'_t,$$

and \hat{w}_t is the residual process. Under the null that w_t is white noise, (5.84) has an asymptotic χ^2 distribution with $k^2(H - p)$ degrees of freedom.

Finally, prediction follows in a straight forward manner from the univariate case. Using the R package `vars`, use the `predict` command and the `fanchart` command, which produces a nice graphic:

```
(fit.pr = predict(fit, n.ahead = 24, ci = 0.95)) # 4 weeks ahead
fanchart(fit.pr) # plot prediction + error
```

The results are displayed in Figure 5.14; we note that the package stripped time when plotting the fanchart and the horizontal axis is labeled 1, 2, 3,

For pure $\text{VAR}(p)$ models, the autocovariance structure leads to the multivariate version of the Yule–Walker equations:

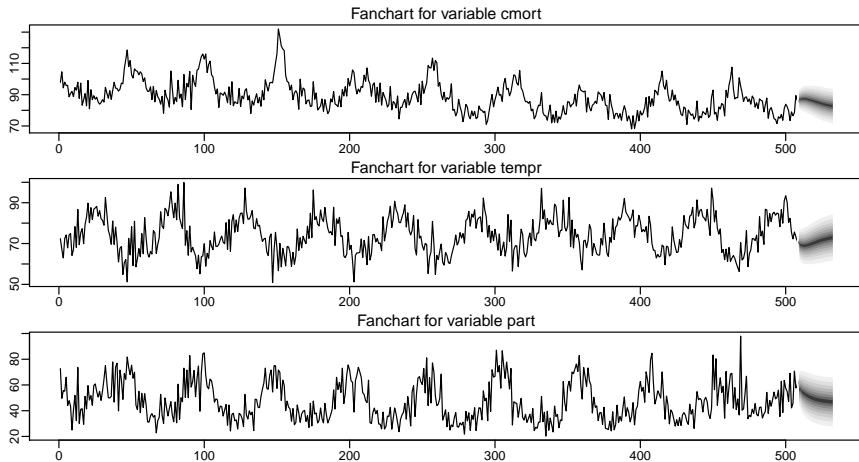


Fig. 5.14. Predictions from a VAR(2) fit to the LA mortality – pollution data.

$$\Gamma(h) = \sum_{j=1}^p \Phi_j \Gamma(h-j), \quad h = 1, 2, \dots, \quad (5.85)$$

$$\Gamma(0) = \sum_{j=1}^p \Phi_j \Gamma(-j) + \Sigma_w. \quad (5.86)$$

where $\Gamma(h) = \text{cov}(x_{t+h}, x_t)$ is a $k \times k$ matrix and $\Gamma(-h) = \Gamma(h)'$.

Estimation of the autocovariance matrix is similar to the univariate case, that is, with $\bar{x} = n^{-1} \sum_{t=1}^n x_t$, as an estimate of $\mu = Ex_t$,

$$\hat{\Gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x})', \quad h = 0, 1, 2, \dots, n-1, \quad (5.87)$$

and $\hat{\Gamma}(-h) = \hat{\Gamma}(h)'$. If $\hat{\gamma}_{i,j}(h)$ denotes the element in the i -th row and j -th column of $\hat{\Gamma}(h)$, the cross-correlation functions (CCF), as discussed in (1.35), are estimated by

$$\hat{\rho}_{i,j}(h) = \frac{\hat{\gamma}_{i,j}(h)}{\sqrt{\hat{\gamma}_{i,i}(0)} \sqrt{\hat{\gamma}_{j,j}(0)}} \quad h = 0, 1, 2, \dots, n-1. \quad (5.88)$$

When $i = j$ in (5.88), we get the estimated autocorrelation function (ACF) of the individual series.

Although least squares estimation was used in [Example 5.10](#) and [Example 5.11](#), we could have also used Yule-Walker estimation, conditional or unconditional maximum likelihood estimation. As in the univariate case, the Yule–Walker estimators, the maximum likelihood estimators, and the least squares estimators are asymptotically equivalent. To exhibit the asymptotic distribution of the autoregression parameter estimators, we write

$$\phi = \text{vec}(\Phi_1, \dots, \Phi_p),$$

where the *vec operator* stacks the columns of a matrix into a vector. For example, for a bivariate AR(2) model,

$$\phi = \text{vec}(\Phi_1, \Phi_2) = (\Phi_{111}, \Phi_{121}, \Phi_{112}, \Phi_{122}, \Phi_{211}, \Phi_{221}, \Phi_{212}, \Phi_{222})',$$

where $\Phi_{\ell ij}$ is the ij -th element of Φ_ℓ , $\ell = 1, 2$. Because (Φ_1, \dots, Φ_p) is a $k \times kp$ matrix, ϕ is a $k^2 p \times 1$ vector. We now state the following property.

Property 5.1 Large-Sample Distribution of VAR Estimators

Let $\hat{\phi}$ denote the vector of parameter estimators (obtained via Yule–Walker, least squares, or maximum likelihood) for a k -dimensional AR(p) model. Then,

$$\sqrt{n}(\hat{\phi} - \phi) \sim AN(0, \Sigma_w \otimes \Gamma_{pp}^{-1}), \quad (5.89)$$

where $\Gamma_{pp} = \{\Gamma(i-j)\}_{i,j=1}^p$ is a $kp \times kp$ matrix and $\Sigma_w \otimes \Gamma_{pp}^{-1} = \{\sigma_{ij}\Gamma_{pp}^{-1}\}_{i,j=1}^k$ is a $k^2 p \times k^2 p$ matrix with σ_{ij} denoting the ij -th element of Σ_w .

The variance–covariance matrix of the estimator $\hat{\phi}$ is approximated by replacing Σ_w by $\hat{\Sigma}_w$, and replacing $\Gamma(h)$ by $\hat{\Gamma}(h)$ in Γ_{pp} . The square root of the diagonal elements of $\hat{\Sigma}_w \otimes \hat{\Gamma}_{pp}^{-1}$ divided by \sqrt{n} gives the individual standard errors. For the mortality data example, the estimated standard errors for the VAR(2) fit are listed in [Example 5.11](#); although those standard errors were taken from a regression run, they could have also been calculated using [Property 5.1](#).

A $k \times 1$ vector-valued time series x_t , for $t = 0, \pm 1, \pm 2, \dots$, is said to be VARMA(p, q) if x_t is stationary and

$$x_t = \alpha + \Phi_1 x_{t-1} + \cdots + \Phi_p x_{t-p} + w_t + \Theta_1 w_{t-1} + \cdots + \Theta_q w_{t-q}, \quad (5.90)$$

with $\Phi_p \neq 0$, $\Theta_q \neq 0$, and $\Sigma_w > 0$ (that is, Σ_w is positive definite). The coefficient matrices Φ_j ; $j = 1, \dots, p$ and Θ_j ; $j = 1, \dots, q$ are, of course, $k \times k$ matrices. If x_t has mean μ then $\alpha = (I - \Phi_1 - \cdots - \Phi_p)\mu$. As in the univariate case, we will have to place a number of conditions on the multivariate ARMA model to ensure the model is unique and has desirable properties such as causality. These conditions will be discussed shortly.

As in the VAR model, the special form assumed for the constant component can be generalized to include a fixed $r \times 1$ vector of inputs, u_t . That is, we could have proposed the *vector ARMAX model*,

$$x_t = \Gamma u_t + \sum_{j=1}^p \Phi_j x_{t-j} + \sum_{k=1}^q \Theta_k w_{t-k} + w_t, \quad (5.91)$$

where Γ is a $p \times r$ parameter matrix.

While extending univariate AR (or pure MA) models to the vector case is fairly easy, extending univariate ARMA models to the multivariate case is not a simple

matter. Our discussion will be brief, but interested readers can get more details in Lütkepohl (1993), Reinsel (1997), and Tiao and Tsay (1989).

In the multivariate case, the *autoregressive operator* is

$$\Phi(B) = I - \Phi_1 B - \cdots - \Phi_p B^p, \quad (5.92)$$

and the *moving average operator* is

$$\Theta(B) = I + \Theta_1 B + \cdots + \Theta_q B^q, \quad (5.93)$$

The zero-mean VARMA(p, q) model is then written in the concise form as

$$\Phi(B)x_t = \Theta(B)w_t. \quad (5.94)$$

The model is said to be *causal* if the roots of $|\Phi(z)|$ (where $|\cdot|$ denotes determinant) are outside the unit circle, $|z| > 1$; that is, $|\Phi(z)| \neq 0$ for any value z such that $|z| \leq 1$. In this case, we can write

$$x_t = \Psi(B)w_t,$$

where $\Psi(B) = \sum_{j=0}^{\infty} \Psi_j B^j$, $\Psi_0 = I$, and $\sum_{j=0}^{\infty} \|\Psi_j\| < \infty$. The model is said to be *invertible* if the roots of $|\Theta(z)|$ lie outside the unit circle. Then, we can write

$$w_t = \Pi(B)x_t,$$

where $\Pi(B) = \sum_{j=0}^{\infty} \Pi_j B^j$, $\Pi_0 = I$, and $\sum_{j=0}^{\infty} \|\Pi_j\| < \infty$. Analogous to the univariate case, we can determine the matrices Ψ_j by solving $\Psi(z) = \Phi(z)^{-1}\Theta(z)$, $|z| \leq 1$, and the matrices Π_j by solving $\Pi(z) = \Theta(z)^{-1}\Phi(z)$, $|z| \leq 1$.

For a causal model, we can write $x_t = \Psi(B)w_t$ so the general autocovariance structure of an ARMA(p, q) model is ($h \geq 0$)

$$\Gamma(h) = \text{cov}(x_{t+h}, x_t) = \sum_{j=0}^{\infty} \Psi_{j+h} \Sigma_w \Psi_j'. \quad (5.95)$$

and $\Gamma(-h) = \Gamma(h)'$. For pure MA(q) processes, (5.95) becomes

$$\Gamma(h) = \sum_{j=0}^{q-h} \Theta_{j+h} \Sigma_w \Theta_j', \quad (5.96)$$

where $\Theta_0 = I$. Of course, (5.96) implies $\Gamma(h) = 0$ for $h > q$.

As in the univariate case, we will need conditions for model uniqueness. These conditions are similar to the condition in the univariate case that the autoregressive and moving average polynomials have no common factors. To explore the uniqueness problems that we encounter with multivariate ARMA models, consider a bivariate AR(1) process, $x_t = (x_{t,1}, x_{t,2})'$, given by

$$x_{t,1} = \phi x_{t-1,2} + w_{t,1},$$

$$x_{t,2} = w_{t,2},$$

where $w_{t,1}$ and $w_{t,2}$ are independent white noise processes and $|\phi| < 1$. Both processes, $x_{t,1}$ and $x_{t,2}$ are causal and invertible. Moreover, the processes are jointly stationary because $\text{cov}(x_{t+h,1}, x_{t,2}) = \phi \text{cov}(x_{t+h-1,2}, x_{t,2}) \equiv \phi \gamma_{2,2}(h-1) = \phi \sigma_{w_2}^2 \delta_1^h$ does not depend on t ; note, $\delta_1^h = 1$ when $h = 1$, otherwise, $\delta_1^h = 0$. In matrix notation, we can write this model as

$$x_t = \Phi x_{t-1} + w_t, \quad \text{where } \Phi = \begin{bmatrix} 0 & \phi \\ 0 & 0 \end{bmatrix}. \quad (5.97)$$

We can write (5.97) in operator notation as

$$\Phi(B)x_t = w_t \quad \text{where } \Phi(z) = \begin{bmatrix} 1 & -\phi z \\ 0 & 1 \end{bmatrix}.$$

In addition, model (5.97) can be written as a bivariate ARMA(1,1) model

$$x_t = \Phi_1 x_{t-1} + \Theta_1 w_{t-1} + w_t, \quad (5.98)$$

where

$$\Phi_1 = \begin{bmatrix} 0 & \phi + \theta \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \Theta_1 = \begin{bmatrix} 0 & -\theta \\ 0 & 0 \end{bmatrix},$$

and θ is arbitrary. To verify this, we write (5.98), as $\Phi_1(B)x_t = \Theta_1(B)w_t$, or

$$\Theta_1(B)^{-1}\Phi_1(B)x_t = w_t,$$

where

$$\Phi_1(z) = \begin{bmatrix} 1 & -(\phi + \theta)z \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \Theta_1(z) = \begin{bmatrix} 1 & -\theta z \\ 0 & 1 \end{bmatrix}.$$

Then,

$$\Theta_1(z)^{-1}\Phi_1(z) = \begin{bmatrix} 1 & \theta z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -(\phi + \theta)z \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -\phi z \\ 0 & 1 \end{bmatrix} = \Phi(z),$$

where $\Phi(z)$ is the polynomial associated with the bivariate AR(1) model in (5.97). Because θ is arbitrary, the parameters of the ARMA(1,1) model given in (5.98) are not identifiable. No problem exists, however, in fitting the AR(1) model given in (5.97).

The problem in the previous discussion was caused by the fact that both $\Theta(B)$ and $\Theta(B)^{-1}$ are finite; such a matrix operator is called *unimodular*. If $U(B)$ is unimodular, $|U(z)|$ is constant. It is also possible for two seemingly different multivariate ARMA(p, q) models, say, $\Phi(B)x_t = \Theta(B)w_t$ and $\Phi_*(B)x_t = \Theta_*(B)w_t$, to be related through a unimodular operator, $U(B)$ as $\Phi_*(B) = U(B)\Phi(B)$ and $\Theta_*(B) = U(B)\Theta(B)$, in such a way that the orders of $\Phi(B)$ and $\Theta(B)$ are the same as the orders of $\Phi_*(B)$ and $\Theta_*(B)$, respectively. For example, consider the bivariate ARMA(1,1) models given by

$$\Phi x_t \equiv \begin{bmatrix} 1 & -\phi B \\ 0 & 1 \end{bmatrix} x_t = \begin{bmatrix} 1 & \theta B \\ 0 & 1 \end{bmatrix} w_t \equiv \Theta w_t$$

and

$$\Phi_*(B)x_t \equiv \begin{bmatrix} 1 & (\alpha - \phi)B \\ 0 & 1 \end{bmatrix} x_t = \begin{bmatrix} 1 & (\alpha + \theta)B \\ 0 & 1 \end{bmatrix} w_t \equiv \Theta_*(B)w_t,$$

where α , ϕ , and θ are arbitrary constants. Note,

$$\Phi_*(B) \equiv \begin{bmatrix} 1 & (\alpha - \phi)B \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \alpha B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\phi B \\ 0 & 1 \end{bmatrix} \equiv U(B)\Phi(B)$$

and

$$\Theta_*(B) \equiv \begin{bmatrix} 1 & (\alpha + \theta)B \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \alpha B \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \theta B \\ 0 & 1 \end{bmatrix} \equiv U(B)\Theta(B).$$

In this case, both models have the same infinite MA representation $x_t = \Psi(B)w_t$, where

$$\Psi(B) = \Phi(B)^{-1}\Theta(B) = \Phi(B)^{-1}U(B)^{-1}U(B)\Theta(B) = \Phi_*(B)^{-1}\Theta_*(B).$$

This result implies the two models have the same autocovariance function $\Gamma(h)$. Two such ARMA(p, q) models are said to be *observationally equivalent*.

As previously mentioned, in addition to requiring causality and invertibility, we will need some additional assumptions in the multivariate case to make sure that the model is unique. To ensure the *identifiability* of the parameters of the multivariate ARMA(p, q) model, we need the following additional two conditions: (i) the matrix operators $\Phi(B)$ and $\Theta(B)$ have no common left factors other than unimodular ones [that is, if $\Phi(B) = U(B)\Phi_*(B)$ and $\Theta(B) = U(B)\Theta_*(B)$, the common factor must be unimodular] and (ii) with q as small as possible and p as small as possible for that q , the matrix $[\Phi_p, \Theta_q]$ must be full rank, k . One suggestion for avoiding most of the aforementioned problems is to fit only vector AR(p) models in multivariate situations. Although this suggestion might be reasonable for many situations, this philosophy is not in accordance with the law of parsimony because we might have to fit a large number of parameters to describe the dynamics of a process.

Asymptotic inference for the general case of vector ARMA models is more complicated than pure AR models; details can be found in Reinsel (1997) or Lütkepohl (1993), for example. We also note that estimation for VARMA models can be recast into the problem of estimation for state-space models that will be discussed in Chapter 6.

Example 5.12 The Spliid Algorithm for Fitting Vector ARMA

A simple algorithm for fitting vector ARMA models from Spliid (1983) is worth mentioning because it repeatedly uses the multivariate regression equations. Consider a general ARMA(p, q) model for a time series with a nonzero mean

$$x_t = \alpha + \Phi_1 x_{t-1} + \cdots + \Phi_p x_{t-p} + w_t + \Theta_1 w_{t-1} + \cdots + \Theta_q w_{t-q}. \quad (5.99)$$

If $\mu = E x_t$, then $\alpha = (I - \Phi_1 - \cdots - \Phi_p)\mu$. If w_{t-1}, \dots, w_{t-q} were observed, we could rearrange (5.99) as a multivariate regression model

$$x_t = \mathcal{B}z_t + w_t, \quad (5.100)$$

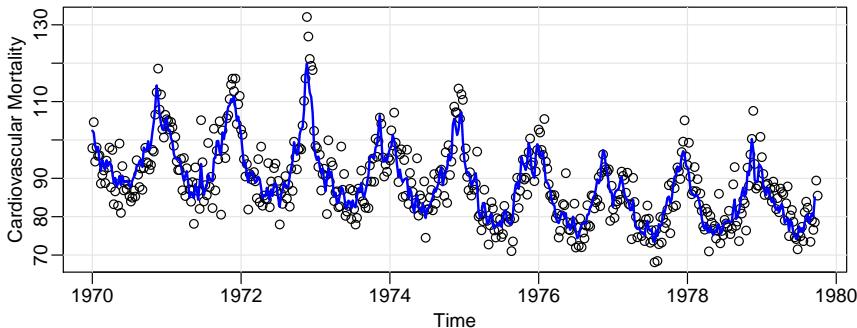


Fig. 5.15. Predictions (line) from a VARMA(2,1) fit to the LA mortality (points) data using Spliid's algorithm.

with

$$z_t = (1, x'_{t-1}, \dots, x'_{t-p}, w'_{t-1}, \dots, w'_{t-q})' \quad (5.101)$$

and

$$\mathcal{B} = [\alpha, \Phi_1, \dots, \Phi_p, \Theta_1, \dots, \Theta_q], \quad (5.102)$$

for $t = p + 1, \dots, n$. Given an initial estimator \mathcal{B}_0 , of \mathcal{B} , we can reconstruct $\{w_{t-1}, \dots, w_{t-q}\}$ by setting

$$w_{t-j} = x_{t-j} - \mathcal{B}_0 z_{t-j}, \quad t = p + 1, \dots, n, \quad j = 1, \dots, q, \quad (5.103)$$

where, if $q > p$, we put $w_{t-j} = 0$ for $t - j \leq 0$. The new values of $\{w_{t-1}, \dots, w_{t-q}\}$ are then put into the regressors z_t and a new estimate, say, \mathcal{B}_1 , is obtained. The initial value, \mathcal{B}_0 , can be computed by fitting a pure autoregression of order p or higher, and taking $\Theta_1 = \dots = \Theta_q = 0$. The procedure is then iterated until the parameter estimates stabilize. The algorithm often converges, but not to the maximum likelihood estimators. Experience suggests the estimators can be reasonably close to the maximum likelihood estimators. The algorithm can be considered as a quick and easy way to fit an initial VARMA model as a starting point to using maximum likelihood estimation, which is best done via state-space models covered in the next chapter.

We used the R package `marima` to fit a vector ARMA(2, 1) to the mortality–pollution data set and part of the output is displayed. We note that mortality is detrended prior to the analysis. The one-step-ahead predictions for mortality are displayed in Figure 5.15.

```
library(marima)
model = define.model(kvar=3, ar=c(1,2), ma=c(1))
arp = model$ar.pattern; map = model$ma.pattern
cmort.d = resid(detr <- lm(cmort~ time(cmort), na.action=NULL))
xdata = matrix(cbind(cmort.d, temp, part), ncol=3) # strip ts attributes
fit = marima(xdata, ar.pattern=arp, ma.pattern=map, means=c(0,1,1),
             penalty=1)
# resid analysis (not displayed)
```

```

innov = t(resid(fit)); plot.ts(innov); acf(innov, na.action=na.pass)
# fitted values for cmort
pred = ts(t(fitted(fit))[,1], start=start(cmort), freq=frequency(cmort)) +
      detr$coef[1] + detr$coef[2]*time(cmort)
plot(pred, ylab="Cardiovascular Mortality", lwd=2, col=4); points(cmort)
# print estimates and corresponding t^2-statistic
short.form(fit$ar.estimates, leading=FALSE)
short.form(fit$ar.fvalues, leading=FALSE)
short.form(fit$ma.estimates, leading=FALSE)
short.form(fit$ma.fvalues, leading=FALSE)
  parameter estimate      t^2 statistic
AR1
-0.311  0.000 -0.114   51.21   0.0    7.9
  0.000 -0.656  0.048    0.00   41.7   3.1
-0.109  0.000 -0.861    1.57   0.0 113.3
AR2:
-0.333  0.133 -0.047   67.24 11.89  2.52
  0.000 -0.200  0.055    0.00   8.10  2.90
  0.179 -0.102 -0.151    4.86   1.77  6.48
MA1:
  0.000 -0.187 -0.106    0.00 14.51  4.75
-0.114 -0.446  0.000    4.68 16.38  0.00
  0.000 -0.278 -0.673    0.00  8.08 47.56
fit$resid.cov  # estimate of noise cov matrix
  27.3   6.5  13.8
  6.5  36.2  38.1
 13.8  38.1 109.2

```

Problems

Section 5.1

5.1 The data set `arf` is 1000 simulated observations from an ARFIMA(1, 1, 0) model with $\phi = .75$ and $d = .4$.

- (a) Plot the data and comment.
- (b) Plot the ACF and PACF of the data and comment.
- (c) Estimate the parameters and test for the significance of the estimates $\hat{\phi}$ and \hat{d} .
- (d) Explain why, using the results of parts (a) and (b), it would seem reasonable to difference the data prior to the analysis. That is, if x_t represents the data, explain why we might choose to fit an ARMA model to ∇x_t .
- (e) Plot the ACF and PACF of ∇x_t and comment.
- (f) Fit an ARMA model to ∇x_t and comment.

5.2 Compute the sample ACF of the absolute values of the NYSE returns displayed in Figure 1.4 up to lag 200, and comment on whether the ACF indicates long memory. Fit an ARFIMA model to the absolute values and comment.

Section 5.2

5.3 Plot the global temperature series, `globtemp`, and then test whether there is a unit root versus the alternative that the process is stationary using the three tests, DF, ADF, and PP, discussed in [Example 5.3](#). Comment.

5.4 Plot the GNP series, `gnp`, and then test for a unit root against the alternative that the process is explosive. State your conclusion.

5.5 Verify (5.33).

Section 5.3

5.6 Weekly crude oil spot prices in dollars per barrel are in `oil`; see Problem [Problem 2.10](#) and Appendix R for more details. Investigate whether the growth rate of the weekly oil price exhibits GARCH behavior. If so, fit an appropriate model to the growth rate.

5.7 The `stats` package of R contains the daily closing prices of four major European stock indices; type `help(EuStockMarkets)` for details. Fit a GARCH model to the returns of one of these series and discuss your findings. (Note: The data set contains actual values, and not returns. Hence, the data must be transformed prior to the model fitting.)

5.8 The 2×1 gradient vector, $l^{(1)}(\alpha_0, \alpha_1)$, given for an ARCH(1) model was displayed in (5.47). Verify (5.47) and then use the result to calculate the 2×2 Hessian matrix

$$l^{(2)}(\alpha_0, \alpha_1) = \begin{pmatrix} \partial^2 l / \partial \alpha_0^2 & \partial^2 l / \partial \alpha_0 \partial \alpha_1 \\ \partial^2 l / \partial \alpha_0 \partial \alpha_1 & \partial^2 l / \partial \alpha_1^2 \end{pmatrix}.$$

Section 5.4

5.9 The sunspot data (`sunspotz`) are plotted in Chapter 4, [Figure 4.22](#). From a time plot of the data, discuss why it is reasonable to fit a threshold model to the data, and then fit a threshold model.

Section 5.5

5.10 The data in `climhyd` have 454 months of measured values for the climatic variables air temperature, dew point, cloud cover, wind speed, precipitation (p_t), and inflow (i_t), at Lake Shasta; the data are displayed in [Figure 7.3](#). We would like to look at possible relations between the weather factors and the inflow to Lake Shasta.

- (a) Fit ARIMA(0, 0, 0) \times (0, 1, 1)₁₂ models to (i) transformed precipitation $P_t = \sqrt{p_t}$ and (ii) transformed inflow $I_t = \log i_t$.

- (b) Apply the ARIMA model fitted in part (a) for transformed precipitation to the flow series to generate the prewhitened flow residuals assuming the precipitation model. Compute the cross-correlation between the flow residuals using the precipitation ARIMA model and the precipitation residuals using the precipitation model and interpret. Use the coefficients from the ARIMA model to construct the transformed flow residuals.

5.11 For the `climhyd` data set, consider predicting the transformed flows $I_t = \log i_t$ from transformed precipitation values $P_t = \sqrt{p_t}$ using a transfer function model of the form

$$(1 - B^{12})I_t = \alpha(B)(1 - B^{12})P_t + n_t,$$

where we assume that seasonal differencing is a reasonable thing to do. You may think of it as fitting

$$y_t = \alpha(B)x_t + n_t,$$

where y_t and x_t are the seasonally differenced transformed flows and precipitations.

- (a) Argue that x_t can be fitted by a first-order seasonal moving average, and use the transformation obtained to prewhiten the series x_t .
- (b) Apply the transformation applied in (a) to the series y_t , and compute the cross-correlation function relating the prewhitened series to the transformed series. Argue for a transfer function of the form

$$\alpha(B) = \frac{\delta_0}{1 - \omega_1 B}.$$

- (c) Write the overall model obtained in regression form to estimate δ_0 and ω_1 . Note that you will be minimizing the sums of squared residuals for the transformed noise series $(1 - \hat{\omega}_1 B)n_t$. Retain the residuals for further modeling involving the noise n_t . The observed residual is $u_t = (1 - \hat{\omega}_1 B)n_t$.
- (d) Fit the noise residuals obtained in (c) with an ARMA model, and give the final form suggested by your analysis in the previous parts.
- (e) Discuss the problem of forecasting y_{t+m} using the infinite past of y_t and the present and infinite past of x_t . Determine the predicted value and the forecast variance.

Section 5.6

5.12 Consider the data set `econ5` containing quarterly U.S. unemployment, GNP, consumption, and government and private investment from 1948-III to 1988-II. The seasonal component has been removed from the data. Concentrating on unemployment (U_t), GNP (G_t), and consumption (C_t), fit a vector ARMA model to the data after first logging each series, and then removing the linear trend. That is, fit a vector ARMA model to $x_t = (x_{1t}, x_{2t}, x_{3t})'$, where, for example, $x_{1t} = \log(U_t) - \hat{\beta}_0 - \hat{\beta}_1 t$, where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the least squares estimates for the regression of $\log(U_t)$ on time, t . Run a complete set of diagnostics on the residuals.

Chapter 6

State Space Models

A very general model that subsumes a whole class of special cases of interest in much the same way that linear regression does is the state-space model or the dynamic linear model, which was introduced in Kalman (1960) and Kalman and Bucy (1961). The model arose in the space tracking setting, where the state equation defines the motion equations for the position or state of a spacecraft with location x_t and the data y_t reflect information that can be observed from a tracking device such as velocity and azimuth. Although introduced as a method primarily for use in aerospace-related research, the model has been applied to modeling data from economics (Harrison and Stevens, 1976; Harvey and Pierse, 1984; Harvey and Todd, 1983; Kitagawa and Gersch 1984, Shumway and Stoffer, 1982), medicine (Jones, 1984) and the soil sciences (Shumway, 1988, §3.4.5). An excellent treatment of time series analysis based on the state space model is the text by Durbin and Koopman (2001). A modern treatment of nonlinear state space models can be found in Douc, Moulines and Stoffer (2014).

In this chapter, we focus primarily on linear Gaussian state space models. We present various forms of the model, introduce the concepts of prediction, filtering and smoothing state space models and include their derivations. We explain how to perform maximum likelihood estimation using various techniques, and include methods for handling missing data. In addition, we present several special topics such as hidden Markov models (HMM), switching autoregressions, smoothing splines, ARMAX models, bootstrapping, stochastic volatility, and state space models with switching. Finally, we discuss a Bayesian approach to fitting state space models using Markov chain Monte Carlo (MCMC) techniques. The essential material is supplied in Sections 6.1, 6.2, and 6.3. After that, the other sections may be read in any order with some occasional backtracking.

In general, the state space model is characterized by two principles. First, there is a hidden or latent process x_t called the state process. The state process is assumed to be a Markov process; this means that the future $\{x_s; s > t\}$, and past $\{x_s; s < t\}$, are independent conditional on the present, x_t . The second condition is that the observations, y_t are independent given the states x_t . This means that the dependence among the observations is generated by states. The principles are displayed in Figure 6.1.

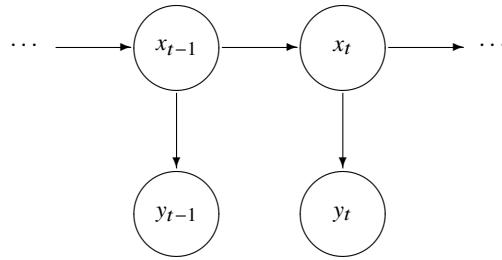


Fig. 6.1. Diagram of a state space model.

6.1 Linear Gaussian Model

The linear Gaussian state space model or dynamic linear model (DLM), in its basic form, employs an order one, p -dimensional vector autoregression as the *state equation*,

$$x_t = \Phi x_{t-1} + w_t. \quad (6.1)$$

The w_t are $p \times 1$ independent and identically distributed, zero-mean normal vectors with covariance matrix Q ; we write this as $w_t \sim \text{iid } N_p(0, Q)$. In the DLM, we assume the process starts with a normal vector x_0 , such that $x_0 \sim N_p(\mu_0, \Sigma_0)$.

We do not observe the state vector x_t directly, but only a linear transformed version of it with noise added, say

$$y_t = A_t x_t + v_t, \quad (6.2)$$

where A_t is a $q \times p$ *measurement* or *observation matrix*; (6.2) is called the *observation equation*. The observed data vector, y_t , is q -dimensional, which can be larger than or smaller than p , the state dimension. The additive observation noise is $v_t \sim \text{iid } N_q(0, R)$. In addition, we initially assume, for simplicity, x_0 , $\{w_t\}$ and $\{v_t\}$ are uncorrelated; this assumption is not necessary, but it helps in the explanation of first concepts. The case of correlated errors is discussed in Section 6.6.

As in the ARMAX model of Section 5.6, exogenous variables, or fixed inputs, may enter into the states or into the observations. In this case, we suppose we have an $r \times 1$ vector of inputs u_t , and write the model as

$$x_t = \Phi x_{t-1} + \Upsilon u_t + w_t \quad (6.3)$$

$$y_t = A_t x_t + \Gamma u_t + v_t \quad (6.4)$$

where Υ is $p \times r$ and Γ is $q \times r$; either of these matrices may be the zero matrix.

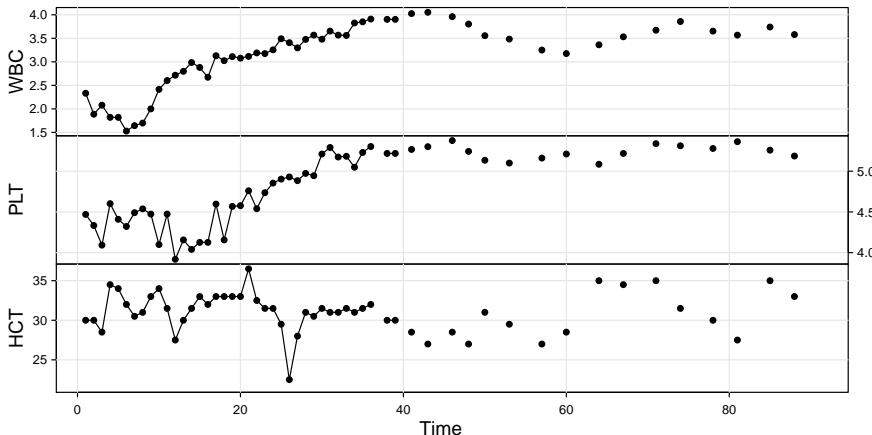


Fig. 6.2. Longitudinal series of monitored blood parameters, log (white blood count) [WBC], log (platelet) [PLT], and hematocrit [HCT], after a bone marrow transplant ($n = 91$ days).

Example 6.1 A Biomedical Example

Suppose we consider the problem of monitoring the level of several biomedical markers after a cancer patient undergoes a bone marrow transplant. The data in [Figure 6.2](#), used by Jones (1984), are measurements made for 91 days on three variables, log(white blood count) [WBC], log(platelet) [PLT], and hematocrit [HCT], denoted $y_t = (y_{t1}, y_{t2}, y_{t3})'$. Approximately 40% of the values are missing, with missing values occurring primarily after the 35th day. The main objectives are to model the three variables using the state-space approach, and to estimate the missing values. According to Jones, “Platelet count at about 100 days post transplant has previously been shown to be a good indicator of subsequent long term survival.” For this particular situation, we model the three variables in terms of the state equation (6.1); that is,

$$\begin{pmatrix} x_{t1} \\ x_{t2} \\ x_{t3} \end{pmatrix} = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{pmatrix} \begin{pmatrix} x_{t-1,1} \\ x_{t-1,2} \\ x_{t-1,3} \end{pmatrix} + \begin{pmatrix} w_{t1} \\ w_{t2} \\ w_{t3} \end{pmatrix}. \quad (6.5)$$

The observation equations would be $y_t = A_t x_t + v_t$, where the 3×3 observation matrix, A_t , is either the identity matrix or the zero matrix depending on whether a blood sample was taken on that day. The covariance matrices R and Q are each 3×3 matrices. A plot similar to [Figure 6.2](#) can be produced as follows.

```
plot(blood, type='o', pch=19, xlab='day', main='')
```

As we progress through the chapter, it will become apparent that, while the model seems simplistic, it is quite general. For example, if the state process is VAR(2), we may write the state equation as a $2p$ -dimensional process,

$$\begin{pmatrix} x_t \\ x_{t-1} \end{pmatrix}_{2p \times 1} = \begin{pmatrix} \Phi_1 & \Phi_2 \\ I & 0 \end{pmatrix}_{2p \times 2p} \begin{pmatrix} x_{t-1} \\ x_{t-2} \end{pmatrix}_{2p \times 1} + \begin{pmatrix} w_t \\ 0 \end{pmatrix}_{2p \times 1}, \quad (6.6)$$

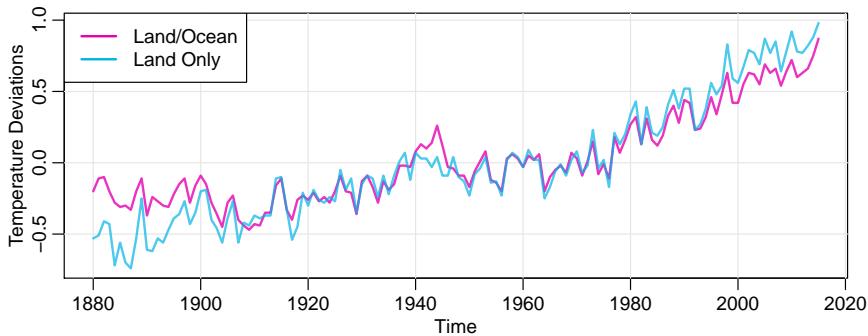


Fig. 6.3. Annual global temperature deviation series, measured in degrees centigrade, 1880–2015. The series differ by whether or not ocean data is included.

and the observation equation as the q -dimensional process,

$$\mathbf{y}_t = \begin{bmatrix} A_t & | & 0 \end{bmatrix}_{q \times 2p} \begin{pmatrix} x_t \\ x_{t-1} \end{pmatrix}_{2p \times 1} + v_t. \quad (6.7)$$

The real advantages of the state space formulation, however, do not really come through in the simple example given above. The special forms that can be developed for various versions of the matrix A_t and for the transition scheme defined by the matrix Φ allow fitting more parsimonious structures with fewer parameters needed to describe a multivariate time series. We will see numerous examples throughout the chapter; [Section 6.5 on structural models](#) is a good example of the model flexibility. The simple example shown below is instructive.

Example 6.2 Global Warming

[Figure 6.3](#) shows two different estimators for the global temperature series from 1880 to 2015. One is `globtemp`, which was considered in the first chapter, and are the global mean land-ocean temperature index data. The second series, `globtempl`, are the surface air temperature index data using only meteorological station data. Conceptually, both series should be measuring the same underlying climatic signal, and we may consider the problem of extracting this underlying signal. The R code to generate the figure is

```
ts.plot(globtemp, globtempl, col=c(6,4), ylab='Temperature Deviations')
```

We suppose both series are observing the same signal with different noises; that is,

$$y_{t1} = x_t + v_{t1} \quad \text{and} \quad y_{t2} = x_t + v_{t2},$$

or more compactly as

$$\begin{pmatrix} y_{t1} \\ y_{t2} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} x_t + \begin{pmatrix} v_{t1} \\ v_{t2} \end{pmatrix}, \quad (6.8)$$

where

$$R = \text{var} \begin{pmatrix} v_{t1} \\ v_{t2} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}.$$

It is reasonable to suppose that the unknown common signal, x_t , can be modeled as a random walk with drift of the form

$$x_t = \delta + x_{t-1} + w_t, \quad (6.9)$$

with $Q = \text{var}(w_t)$. In terms of the model (6.3)–(6.4), this example has, $p = 1$, $q = 2$, $\Phi = 1$, and $\Upsilon = \delta$ with $u_t \equiv 1$.

The introduction of the state-space approach as a tool for modeling data in the social and biological sciences requires model identification and parameter estimation because there is rarely a well-defined differential equation describing the state transition. The questions of general interest for the dynamic linear model (6.3) and (6.4) relate to estimating the unknown parameters contained in Φ , Υ , Q , Γ , A_t , and R , that define the particular model, and estimating or forecasting values of the underlying unobserved process x_t . The advantages of the state-space formulation are in the ease with which we can treat various missing data configurations and in the incredible array of models that can be generated from (6.3) and (6.4). The analogy between the observation matrix A_t and the design matrix in the usual regression and analysis of variance setting is a useful one. We can generate fixed and random effect structures that are either constant or vary over time simply by making appropriate choices for the matrix A_t and the transition structure Φ .

Before continuing our investigation of the general model, it is instructive to consider a simple univariate state-space model wherein an AR(1) process is observed using a noisy instrument.

Example 6.3 An AR(1) Process with Observational Noise

Consider a univariate state-space model where the observations are noisy,

$$y_t = x_t + v_t, \quad (6.10)$$

and the signal (state) is an AR(1) process,

$$x_t = \phi x_{t-1} + w_t, \quad (6.11)$$

where $v_t \sim \text{iid } N(0, \sigma_v^2)$, $w_t \sim \text{iid } N(0, \sigma_w^2)$, and $x_0 \sim N(0, \frac{\sigma_w^2}{1-\phi^2})$; $\{v_t\}$, $\{w_t\}$, and x_0 are independent, and $t = 1, 2, \dots$.

In Chapter 3, we investigated the properties of the state, x_t , because it is a stationary AR(1) process (recall Problem 3.2). For example, we know the autocovariance function of x_t is

$$\gamma_x(h) = \frac{\sigma_w^2}{1-\phi^2} \phi^h, \quad h = 0, 1, 2, \dots. \quad (6.12)$$

But here, we must investigate how the addition of observation noise affects the dynamics. Although it is not a necessary assumption, we have assumed in this

example that x_t is stationary. In this case, the observations are also stationary because y_t is the sum of two independent stationary components x_t and v_t . We have

$$\gamma_y(0) = \text{var}(y_t) = \text{var}(x_t + v_t) = \frac{\sigma_w^2}{1 - \phi^2} + \sigma_v^2, \quad (6.13)$$

and, when $h \geq 1$,

$$\gamma_y(h) = \text{cov}(y_t, y_{t-h}) = \text{cov}(x_t + v_t, x_{t-h} + v_{t-h}) = \gamma_x(h). \quad (6.14)$$

Consequently, for $h \geq 1$, the ACF of the observations is

$$\rho_y(h) = \frac{\gamma_y(h)}{\gamma_y(0)} = \left(1 + \frac{\sigma_v^2}{\sigma_w^2}(1 - \phi^2)\right)^{-1} \phi^h. \quad (6.15)$$

It should be clear from the correlation structure given by (6.15) that the observations, y_t , are not AR(1) unless $\sigma_v^2 = 0$. In addition, the autocorrelation structure of y_t is identical to the autocorrelation structure of an ARMA(1,1) process, as presented in [Example 3.14](#). Thus, the observations can also be written in an ARMA(1,1) form,

$$y_t = \phi y_{t-1} + \theta u_{t-1} + u_t,$$

where u_t is Gaussian white noise with variance σ_u^2 , and with θ and σ_u^2 suitably chosen. We leave the specifics of this problem alone for now and defer the discussion to [Section 6.6](#); in particular, see [Example 6.11](#).

Although an equivalence exists between stationary ARMA models and stationary state-space models (see [Section 6.6](#)), it is sometimes easier to work with one form than another. As previously mentioned, in the case of missing data, complex multivariate systems, mixed effects, and certain types of nonstationarity, it is easier to work in the framework of state-space models.

6.2 Filtering, Smoothing, and Forecasting

From a practical view, a primary aim of any analysis involving the state space model, (6.3)–(6.4), would be to produce estimators for the underlying unobserved signal x_t , given the data $y_{1:s} = \{y_1, \dots, y_s\}$, to time s . As will be seen, state estimation is an essential component of parameter estimation. When $s < t$, the problem is called *forecasting* or *prediction*. When $s = t$, the problem is called *filtering*, and when $s > t$, the problem is called *smoothing*. In addition to these estimates, we would also want to measure their precision. The solution to these problems is accomplished via the *Kalman filter and smoother* and is the focus of this section.

Throughout this chapter, we will use the following definitions:

$$x_t^s = \mathbb{E}(x_t \mid y_{1:s}) \quad (6.16)$$

and

$$P_{t_1, t_2}^s = E \{ (x_{t_1} - x_{t_1}^s)(x_{t_2} - x_{t_2}^s)' \}. \quad (6.17)$$

When $t_1 = t_2$ ($= t$ say) in (6.17), we will write P_t^s for convenience.

In obtaining the filtering and smoothing equations, we will rely heavily on the Gaussian assumption. Some knowledge of the material covered in [Appendix B](#) will be helpful in understanding the details of this section (although these details may be skipped on a casual reading of the material). Even in the non-Gaussian case, the estimators we obtain are the minimum mean-squared error estimators within the class of linear estimators. That is, we can think of E in (6.16) as the projection operator in the sense of [Section B.1](#) rather than expectation and $y_{1:s}$ as the space of linear combinations of $\{y_1, \dots, y_s\}$; in this case, P_t^s is the corresponding mean-squared error. Since the processes are Gaussian, (6.17) is also the conditional error covariance; that is,

$$P_{t_1, t_2}^s = E \{ (x_{t_1} - x_{t_1}^s)(x_{t_2} - x_{t_2}^s)' \mid y_{1:s} \}.$$

This fact can be seen, for example, by noting the covariance matrix between $(x_t - x_t^s)$ and $y_{1:s}$, for any t and s , is zero; we could say they are orthogonal in the sense of [Section B.1](#). This result implies that $(x_t - x_t^s)$ and $y_{1:s}$ are independent (because of the normality), and hence, the conditional distribution of $(x_t - x_t^s)$ given $y_{1:s}$ is the unconditional distribution of $(x_t - x_t^s)$. Derivations of the filtering and smoothing equations from a Bayesian perspective are given in Meinhold and Singpurwalla (1983); more traditional approaches based on the concept of projection and on multivariate normal distribution theory are given in Jazwinski (1970) and Anderson and Moore (1979).

First, we present the Kalman filter, which gives the filtering and forecasting equations. The name filter comes from the fact that x_t^t is a linear filter of the observations $y_{1:t}$; that is, $x_t^t = \sum_{s=1}^t B_s y_s$ for suitably chosen $p \times q$ matrices B_s . The advantage of the Kalman filter is that it specifies how to update the filter from x_{t-1}^{t-1} to x_t^t once a new observation y_t is obtained, without having to reprocess the entire data set $y_{1:t}$.

Property 6.1 The Kalman Filter

For the state-space model specified in (6.3) and (6.4), with initial conditions $x_0^0 = \mu_0$ and $P_0^0 = \Sigma_0$, for $t = 1, \dots, n$,

$$x_t^{t-1} = \Phi x_{t-1}^{t-1} + \Gamma u_t, \quad (6.18)$$

$$P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi' + Q, \quad (6.19)$$

with

$$x_t^t = x_t^{t-1} + K_t(y_t - A_t x_t^{t-1} - \Gamma u_t), \quad (6.20)$$

$$P_t^t = [I - K_t A_t] P_t^{t-1}, \quad (6.21)$$

where

$$K_t = P_t^{t-1} A_t' [A_t P_t^{t-1} A_t' + R]^{-1} \quad (6.22)$$

is called the Kalman gain. Prediction for $t > n$ is accomplished via (6.18) and (6.19) with initial conditions x_n^n and P_n^n . Important byproducts of the filter are the innovations (prediction errors)

$$\epsilon_t = y_t - \mathbb{E}(y_t \mid y_{1:t-1}) = y_t - A_t x_t^{t-1} - \Gamma u_t, \quad (6.23)$$

and the corresponding variance-covariance matrices

$$\Sigma_t \stackrel{\text{def}}{=} \text{var}(\epsilon_t) = \text{var}[A_t(x_t - x_t^{t-1}) + v_t] = A_t P_t^{t-1} A_t' + R \quad (6.24)$$

for $t = 1, \dots, n$. We assume that $\Sigma_t > 0$ (is positive definite), which is guaranteed, for example, if $R > 0$. This assumption is not necessary and may be relaxed.

Proof: The derivations of (6.18) and (6.19) follow from straight forward calculations, because from (6.3) we have

$$x_t^{t-1} = \mathbb{E}(x_t \mid y_{1:t-1}) = \mathbb{E}(\Phi x_{t-1} + \Upsilon u_t + w_t \mid y_{1:t-1}) = \Phi x_{t-1}^{t-1} + \Upsilon u_t,$$

and thus

$$\begin{aligned} P_t^{t-1} &= \mathbb{E}\{(x_t - x_t^{t-1})(x_t - x_t^{t-1})'\} \\ &= \mathbb{E}\left\{\left[\Phi(x_{t-1} - x_{t-1}^{t-1}) + w_t\right]\left[\Phi(x_{t-1} - x_{t-1}^{t-1}) + w_t\right]'\right\} \\ &= \Phi P_{t-1}^{t-1} \Phi' + Q. \end{aligned}$$

To derive (6.20), we note that $\text{cov}(\epsilon_t, y_s) = 0$ for $s < t$, which in view of the fact the innovation sequence is a Gaussian process, implies that the innovations are independent of the past observations. Furthermore, the conditional covariance between x_t and ϵ_t given $y_{1:t-1}$ is

$$\begin{aligned} \text{cov}(x_t, \epsilon_t \mid y_{1:t-1}) &= \text{cov}(x_t, y_t - A_t x_t^{t-1} - \Gamma u_t \mid y_{1:t-1}) \\ &= \text{cov}(x_t - x_t^{t-1}, y_t - A_t x_t^{t-1} - \Gamma u_t \mid y_{1:t-1}) \\ &= \text{cov}[x_t - x_t^{t-1}, A_t(x_t - x_t^{t-1}) + v_t] \\ &= P_t^{t-1} A_t'. \end{aligned} \quad (6.25)$$

Using these results we have that the joint conditional distribution of x_t and ϵ_t given $y_{1:t-1}$ is normal

$$\begin{pmatrix} x_t \\ \epsilon_t \end{pmatrix} \Big| y_{1:t-1} \sim N \left(\begin{bmatrix} x_t^{t-1} \\ 0 \end{bmatrix}, \begin{bmatrix} P_t^{t-1} & P_t^{t-1} A_t' \\ A_t P_t^{t-1} & \Sigma_t \end{bmatrix} \right). \quad (6.26)$$

Thus, using (B.9) of Appendix B, we can write

$$x_t^t = \mathbb{E}(x_t \mid y_{1:t}) = \mathbb{E}(x_t \mid y_{1:t-1}, \epsilon_t) = x_t^{t-1} + K_t \epsilon_t, \quad (6.27)$$

where

$$K_t = P_t^{t-1} A_t' \Sigma_t^{-1} = P_t^{t-1} A_t' (A_t P_t^{t-1} A_t' + R)^{-1}.$$

The evaluation of P_t^t is easily computed from (6.26) [see (B.10)] as

$$P_t^t = \text{cov}(x_t \mid y_{1:t-1}, \epsilon_t) = P_t^{t-1} - P_t^{t-1} A_t' \Sigma_t^{-1} A_t P_t^{t-1},$$

which simplifies to (6.21). \square

Nothing in the proof of Property 6.1 precludes the cases where some or all of the parameters vary with time, or where the observation dimension changes with time, which leads to the following corollary.

Corollary 6.1 Kalman Filter: The Time-Varying Case

If, in (6.3) and (6.4), any or all of the parameters are time dependent, $\Phi = \Phi_t$, $\Upsilon = \Upsilon_t$, $Q = Q_t$ in the state equation or $\Gamma = \Gamma_t$, $R = R_t$ in the observation equation, or the dimension of the observational equation is time dependent, $q = q_t$, [Property 6.1](#) holds with the appropriate substitutions.

Next, we explore the model, prediction, and filtering from a density point of view. To ease the notation, we will drop the inputs from the model. There are two key ingredients to the state space model. Letting $p_{\Theta}(\cdot)$ denote a generic density function with parameters represented by Θ , we have the state process is Markovian:

$$p_{\Theta}(x_t \mid x_{t-1}, x_{t-2}, \dots, x_0) = p_{\Theta}(x_t \mid x_{t-1}), \quad (6.28)$$

and the observations are conditionally independent given the states:

$$p_{\Theta}(y_{1:n} \mid x_{1:n}) = \prod_{t=1}^n p_{\Theta}(y_t \mid x_t), \quad (6.29)$$

Since we are focusing on the linear Gaussian model, if we let $g(x; \mu, \Sigma)$ denote a multivariate normal density with mean μ and covariance matrix Σ as given in (1.33), then

$$p_{\Theta}(x_t \mid x_{t-1}) = g(x_t; \Phi x_{t-1}, Q) \quad \text{and} \quad p_{\Theta}(y_t \mid x_t) = g(y_t; A_t x_t, R).$$

with initial condition $p_{\Theta}(x_0) = g(x_0; \mu_0, \Sigma_0)$.

In terms of densities, the Kalman filter can be seen as a simple updating scheme, where, to determine the forecast densities, we have,

$$\begin{aligned} p_{\Theta}(x_t \mid y_{1:t-1}) &= \int_{\mathbb{R}^P} p_{\Theta}(x_t, x_{t-1} \mid y_{1:t-1}) dx_{t-1} \\ &= \int_{\mathbb{R}^P} p_{\Theta}(x_t \mid x_{t-1}) p_{\Theta}(x_{t-1} \mid y_{1:t-1}) dx_{t-1} \\ &= \int_{\mathbb{R}^P} g(x_t; \Phi x_{t-1}, Q) g(x_{t-1}; x_{t-1}^{t-1}, P_{t-1}^{t-1}) dx_{t-1} \\ &= g(x_t; x_t^{t-1}, P_t^{t-1}), \end{aligned} \quad (6.30)$$

where the values of x_t^{t-1} and P_t^{t-1} are given in (6.18) and (6.19). These values are obtained upon evaluating the integral using the usual trick of completing the square; see [Example 6.4](#). Since we were seeking an iterative procedure, we introduced x_{t-1} in (6.30) because we have (presumably) previously evaluated the filter density $p_{\Theta}(x_{t-1} \mid y_{1:t-1})$. Once we have the predictor, the filter density is obtained as

$$\begin{aligned} p_{\Theta}(x_t \mid y_{1:t}) &= p_{\Theta}(x_t \mid y_t, y_{1:t-1}) \propto p_{\Theta}(y_t \mid x_t) p_{\Theta}(x_t \mid y_{1:t-1}), \\ &= g(y_t; A_t x_t, R) g(x_t; x_t^{t-1}, P_t^{t-1}), \end{aligned} \quad (6.31)$$

from which we deduce is $g(x_t; x_t^t, P_t^t)$ where x_t^t and P_t^t are given in (6.20) and (6.21). The following example illustrates these ideas for a simple univariate case.

Example 6.4 Local Level Model

In this example, we suppose that we observe a univariate series y_t that consists of a trend component, μ_t , and a noise component, v_t , where

$$y_t = \mu_t + v_t \quad (6.32)$$

and $v_t \sim \text{iid } N(0, \sigma_v^2)$. In particular, we assume the trend is a random walk given by

$$\mu_t = \mu_{t-1} + w_t \quad (6.33)$$

where $w_t \sim \text{iid } N(0, \sigma_w^2)$ is independent of $\{v_t\}$. Recall Example 6.2, where we suggested this type of trend model for the global temperature series.

The model is, of course, a state-space model with (6.32) being the observation equation, and (6.33) being the state equation. We will use the following notation introduced in Blight (1974). Let

$$\{x; \mu, \sigma^2\} = \exp \left\{ -\frac{1}{2\sigma^2}(x - \mu)^2 \right\}, \quad (6.34)$$

then simple manipulation shows

$$\{x; \mu, \sigma^2\} = \{\mu; x, \sigma^2\} \quad (6.35)$$

and by completing the square,

$$\begin{aligned} \{x; \mu_1, \sigma_1^2\} \{x; \mu_2, \sigma_2^2\} &= \left\{ x; \frac{\mu_1/\sigma_1^2 + \mu_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2}, (1/\sigma_1^2 + 1/\sigma_2^2)^{-1} \right\} \\ &\quad \times \{\mu_1; \mu_2, \sigma_1^2 + \sigma_2^2\}. \end{aligned} \quad (6.36)$$

Thus, using (6.30), (6.35) and (6.36) we have

$$\begin{aligned} p(\mu_t | y_{1:t-1}) &\propto \int \{\mu_t; \mu_{t-1}, \sigma_w^2\} \{\mu_{t-1}; \mu_{t-1}^{t-1}, P_{t-1}^{t-1}\} d\mu_{t-1} \\ &= \int \{\mu_{t-1}; \mu_t, \sigma_w^2\} \{\mu_{t-1}; \mu_{t-1}^{t-1}, P_{t-1}^{t-1}\} d\mu_{t-1} \\ &= \{\mu_t; \mu_{t-1}^{t-1}, P_{t-1}^{t-1} + \sigma_w^2\}. \end{aligned} \quad (6.37)$$

From (6.37) we conclude that

$$\mu_t | y_{1:t-1} \sim N(\mu_{t-1}^{t-1}, P_t^{t-1}) \quad (6.38)$$

where

$$\mu_t^{t-1} = \mu_{t-1}^{t-1} \quad \text{and} \quad P_t^{t-1} = P_{t-1}^{t-1} + \sigma_w^2 \quad (6.39)$$

which agrees with the first part of Property 6.1. To derive the filter density using (6.31) and (6.35) we have

$$\begin{aligned} p(\mu_t \mid y_{1:t}) &\propto \{y_t; \mu_t, \sigma_v^2\} \{\mu_t; \mu_t^{t-1}, P_t^{t-1}\} \\ &= \{\mu_t; y_t, \sigma_v^2\} \{\mu_t; \mu_t^{t-1}, P_t^{t-1}\}. \end{aligned} \quad (6.40)$$

An application of (6.36) gives

$$\mu_t \mid y_{1:t} \sim N(\mu_t^t, P_t^t) \quad (6.41)$$

with

$$\mu_t^t = \frac{\sigma_v^2 \mu_t^{t-1}}{P_t^{t-1} + \sigma_v^2} + \frac{P_t^{t-1} y_t}{P_t^{t-1} + \sigma_v^2} = \mu_t^{t-1} + K_t(y_t - \mu_t^{t-1}), \quad (6.42)$$

where we have defined

$$K_t = \frac{P_t^{t-1}}{P_t^{t-1} + \sigma_v^2}, \quad (6.43)$$

and

$$P_t^t = \left(\frac{1}{\sigma_v^2} + \frac{1}{P_t^{t-1}} \right)^{-1} = \frac{\sigma_v^2 P_t^{t-1}}{P_t^{t-1} + \sigma_v^2} = (1 - K_t) P_t^{t-1}. \quad (6.44)$$

The filter for this specific case, of course, agrees with [Property 6.1](#).

Next, we consider the problem of obtaining estimators for x_t based on the entire data sample y_1, \dots, y_n , where $t \leq n$, namely, x_t^n . These estimators are called smoothers because a time plot of the sequence $\{x_t^n; t = 1, \dots, n\}$ is typically smoother than the forecasts $\{x_t^{t-1}; t = 1, \dots, n\}$ or the filters $\{x_t^t; t = 1, \dots, n\}$. As is obvious from the above remarks, smoothing implies that each estimated value is a function of the present, future, and past, whereas the filtered estimator depends on the present and past. The forecast depends only on the past, as usual.

Property 6.2 The Kalman Smoother

For the state-space model specified in (6.3) and (6.4), with initial conditions x_n^n and P_n^n obtained via [Property 6.1](#), for $t = n, n-1, \dots, 1$,

$$x_{t-1}^n = x_{t-1}^{t-1} + J_{t-1} \left(x_t^n - x_t^{t-1} \right), \quad (6.45)$$

$$P_{t-1}^n = P_{t-1}^{t-1} + J_{t-1} \left(P_t^n - P_t^{t-1} \right) J'_{t-1}, \quad (6.46)$$

where

$$J_{t-1} = P_{t-1}^{t-1} \Phi' \left[P_t^{t-1} \right]^{-1}. \quad (6.47)$$

Proof: The smoother can be derived in many ways. Here we provide a proof that was given in Ansley and Kohn (1982). First, for $1 \leq t \leq n$, define

$$y_{1:t-1} = \{y_1, \dots, y_{t-1}\} \quad \text{and} \quad \eta_t = \{v_t, \dots, v_n, w_{t+1}, \dots, w_n\},$$

with $y_{1:0}$ being empty, and let

$$m_{t-1} = E\{x_{t-1} \mid y_{1:t-1}, x_t - x_t^{t-1}, \eta_t\}.$$

Then, because $y_{1:t-1}$, $\{x_t - x_t^{t-1}\}$, and η_t are mutually independent, and x_{t-1} and η_t are independent, using (B.9) we have

$$m_{t-1} = x_{t-1}^{t-1} + J_{t-1}(x_t - x_t^{t-1}), \quad (6.48)$$

where

$$J_{t-1} = \text{cov}(x_{t-1}, x_t - x_t^{t-1})[P_t^{t-1}]^{-1} = P_{t-1}^{t-1} \Phi' [P_t^{t-1}]^{-1}.$$

Finally, because $y_{1:t-1}$, $x_t - x_t^{t-1}$, and η_t generate $y_{1:n} = \{y_1, \dots, y_n\}$,

$$x_{t-1}^n = E\{x_{t-1} \mid y_{1:n}\} = E\{m_{t-1} \mid y_{1:n}\} = x_{t-1}^{t-1} + J_{t-1}(x_t^n - x_t^{t-1}),$$

which establishes (6.45).

The recursion for the error covariance, P_{t-1}^n , is obtained by straight-forward calculation. Using (6.45) we obtain

$$x_{t-1} - x_{t-1}^n = x_{t-1} - x_{t-1}^{t-1} - J_{t-1} \left(x_t^n - \Phi x_{t-1}^{t-1} \right),$$

or

$$(x_{t-1} - x_{t-1}^n) + J_{t-1} x_t^n = \left(x_{t-1} - x_{t-1}^{t-1} \right) + J_{t-1} \Phi x_{t-1}^{t-1}. \quad (6.49)$$

Multiplying each side of (6.49) by the transpose of itself and taking expectation, we have

$$P_{t-1}^n + J_{t-1} E(x_t^n x_t^{n'}) J_{t-1}' = P_{t-1}^{t-1} + J_{t-1} \Phi E(x_{t-1}^{t-1} x_{t-1}^{t-1'}) \Phi' J_{t-1}', \quad (6.50)$$

using the fact the cross-product terms are zero. But,

$$E(x_t^n x_t^{n'}) = E(x_t x_t') - P_t^n = \Phi E(x_{t-1} x_{t-1}') \Phi' + Q - P_t^n,$$

and

$$E(x_{t-1}^{t-1} x_{t-1}^{t-1'}) = E(x_{t-1} x_{t-1}') - P_{t-1}^{t-1},$$

so (6.50) simplifies to (6.46). \square

Example 6.5 Prediction, Filtering and Smoothing for the Local Level Model

For this example, we simulated $n = 50$ observations from the local level trend model discussed in Example 6.4. We generated a random walk

$$\mu_t = \mu_{t-1} + w_t \quad (6.51)$$

with $w_t \sim \text{iid } N(0, 1)$ and $\mu_0 \sim N(0, 1)$. We then supposed that we observe a univariate series y_t consisting of the trend component, μ_t , and a noise component, $v_t \sim \text{iid } N(0, 1)$, where

$$y_t = \mu_t + v_t. \quad (6.52)$$

The sequences $\{w_t\}$, $\{v_t\}$ and μ_0 were generated independently. We then ran the Kalman filter and smoother, Property 6.1 and Property 6.2, using the actual parameters. The top panel of Figure 6.4 shows the actual values of μ_t as points, and

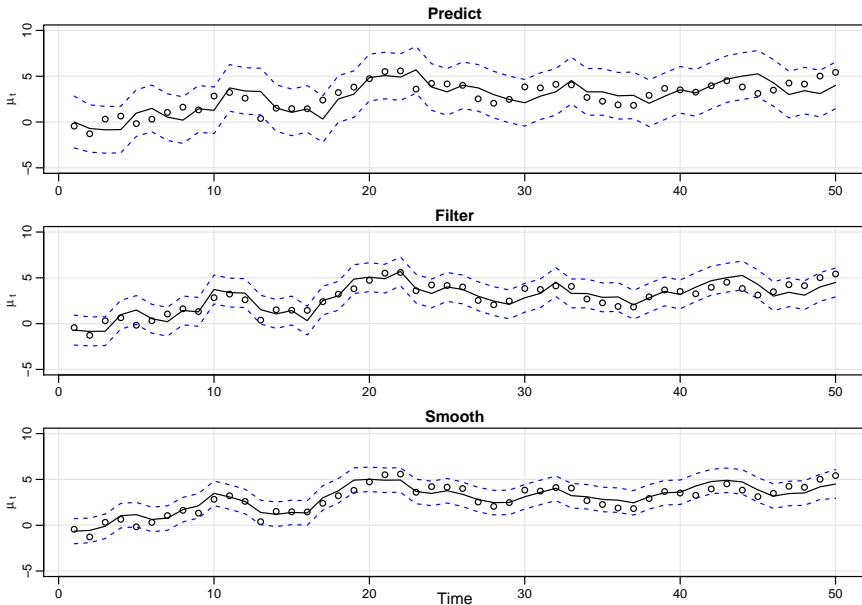


Fig. 6.4. Displays for Example 6.5. The simulated values of μ_t , for $t = 1, \dots, 50$, given by (6.51) are shown as points. The top shows the predictions μ_t^{t-1} as a line with $\pm 2\sqrt{P_t^{t-1}}$ error bounds as dashed lines. The middle is similar, showing $\mu_t^t \pm 2\sqrt{P_t^t}$. The bottom shows $\mu_t^n \pm 2\sqrt{P_t^n}$.

the predictions μ_t^{t-1} , for $t = 1, 2, \dots, 50$, superimposed on the graph as a line. In addition, we display $\mu_t^{t-1} \pm 2\sqrt{P_t^{t-1}}$ as dashed lines on the plot. The middle panel displays the filter, μ_t^t , for $t = 1, \dots, 50$, as a line with $\mu_t^t \pm 2\sqrt{P_t^t}$ as dashed lines. The bottom panel of Figure 6.4 shows a similar plot for the smoother μ_t^n .

Table 6.1 shows the first 10 observations as well as the corresponding state values, the predictions, filters and smoothers. Note that one-step-ahead prediction is more uncertain than the corresponding filtered value, which, in turn, is more uncertain than the corresponding smoother value (that is $P_t^{t-1} \geq P_t^t \geq P_t^n$). Also, in each case, the error variances stabilize quickly.

The R code for this example is as follows. In the example we use `Ksmooth0`, which calls `Kfilter0` for the filtering part. In the returned values from `Ksmooth0`, the letters `p`, `f`, `s` denote prediction, filter, and smooth, respectively (e.g., `xp` is x_t^{t-1} , `xf` is x_t^t , `xs` is x_t^n , and so on). These scripts use a Cholesky-type decomposition^{6.1} of Q and R ; they are denoted by `cQ` and `cR`. Practically, the scripts only require that `Q` or `R` may be reconstructed as `t(cQ)%*%(cQ)` or `t(cR)%*%(cR)`, respectively, which allows more flexibility. For example, the model (6.6) - (6.7) does not pose a problem even though the state noise covariance matrix is not positive definite.

^{6.1} Given a positive definite matrix A , its Cholesky decomposition is an upper triangular matrix U with strictly positive diagonal entries such that $A = U'U$. In R, use `chol(A)`. For the univariate case, it is simply the positive square root of A .

Table 6.1. First 10 Observations of Example 6.5

t	y_t	μ_t	μ_t^{t-1}	P_t^{t-1}	μ_t^t	P_t^t	μ_t^n	P_t^n
0	—	-.63	—	—	.00	1.00	-.32	.62
1	-1.05	-.44	.00	2.00	-.70	.67	-.65	.47
2	-.94	-1.28	-.70	1.67	-.85	.63	-.57	.45
3	-.81	.32	-.85	1.63	-.83	.62	-.11	.45
4	2.08	.65	-.83	1.62	.97	.62	1.04	.45
5	1.81	-.17	.97	1.62	1.49	.62	1.16	.45
6	-.05	.31	1.49	1.62	.53	.62	.63	.45
7	.01	1.05	.53	1.62	.21	.62	.78	.45
8	2.20	1.63	.21	1.62	1.44	.62	1.70	.45
9	1.19	1.32	1.44	1.62	1.28	.62	2.12	.45
10	5.24	2.83	1.28	1.62	3.73	.62	3.48	.45

```
# generate data
set.seed(1); num = 50
w = rnorm(num+1, 0, 1); v = rnorm(num, 0, 1)
mu = cumsum(w) # state: mu[0], mu[1], ..., mu[50]
y = mu[-1] + v # obs: y[1], ..., y[50]
# filter and smooth (Ksmooth0 does both)
ks = Ksmooth0(num, y, A=1, mu0=0, Sigma0=1, Phi=1, cQ=1, cR=1)
# start figure
par(mfrow=c(3,1)); Time = 1:num
plot(Time, mu[-1], main='Predict', ylim=c(-5,10))
lines(ks$xp)
lines(ks$xp+2*sqrt(ks$Pp), lty=2, col=4)
lines(ks$xp-2*sqrt(ks$Pp), lty=2, col=4)
plot(Time, mu[-1], main='Filter', ylim=c(-5,10))
lines(ks$xf)
lines(ks$xf+2*sqrt(ks$Pf), lty=2, col=4)
lines(ks$xf-2*sqrt(ks$Pf), lty=2, col=4)
plot(Time, mu[-1], main='Smooth', ylim=c(-5,10))
lines(ks$xs)
lines(ks$xs+2*sqrt(ks$Ps), lty=2, col=4)
lines(ks$xs-2*sqrt(ks$Ps), lty=2, col=4)
mu[1]; ks$x0n; sqrt(ks$P0n) # initial value info
```

When we discuss maximum likelihood estimation via the EM algorithm in the next section, we will need a set of recursions for obtaining $P_{t,t-1}^n$, as defined in (6.17). We give the necessary recursions in the following property.

Property 6.3 The Lag-One Covariance Smoother

For the state-space model specified in (6.3) and (6.4), with K_t , J_t ($t = 1, \dots, n$), and P_n^n obtained from *Property 6.1* and *Property 6.2*, and with initial condition

$$P_{n,n-1}^n = (I - K_n A_n) \Phi P_{n-1}^{n-1}, \quad (6.53)$$

for $t = n, n-1, \dots, 2$,

$$P_{t-1,t-2}^n = P_{t-1}^{t-1} J'_{t-2} + J_{t-1} \left(P_{t,t-1}^n - \Phi P_{t-1}^{t-1} \right) J'_{t-2}. \quad (6.54)$$

Proof: Because we are computing covariances, we may assume $u_t \equiv 0$ without loss of generality. To derive the initial term (6.53), we first define

$$\tilde{x}_t^s = x_t - x_t^s.$$

Then, using (6.20) and (6.45), we write

$$\begin{aligned} P_{t,t-1}^t &= E\left(\tilde{x}_t^t \tilde{x}_{t-1}^{t'}\right) \\ &= E\left\{[\tilde{x}_t^{t-1} - K_t(y_t - A_t x_t^{t-1})][\tilde{x}_{t-1}^{t-1} - J_{t-1} K_t(y_t - A_t x_t^{t-1})]'\right\} \\ &= E\left\{[\tilde{x}_t^{t-1} - K_t(A_t \tilde{x}_t^{t-1} + v_t)][\tilde{x}_{t-1}^{t-1} - J_{t-1} K_t(A_t \tilde{x}_t^{t-1} + v_t)]'\right\}. \end{aligned}$$

Expanding terms and taking expectation, we arrive at

$$P_{t,t-1}^t = P_{t,t-1}^{t-1} - P_t^{t-1} A_t' K_t' J_{t-1}' - K_t A_t P_{t,t-1}^{t-1} + K_t (A_t P_t^{t-1} A_t' + R) K_t' J_{t-1}',$$

noting $E(\tilde{x}_t^{t-1} v_t') = 0$. The final simplification occurs by realizing that $K_t (A_t P_t^{t-1} A_t' + R) = P_t^{t-1} A_t'$, and $P_{t,t-1}^{t-1} = \Phi P_{t-1}^{t-1}$. These relationships hold for any $t = 1, \dots, n$, and (6.53) is the case $t = n$.

We give the basic steps in the derivation of (6.54). The first step is to use (6.45) to write

$$\tilde{x}_{t-1}^n + J_{t-1} x_t^n = \tilde{x}_{t-1}^{t-1} + J_{t-1} \Phi x_{t-1}^{t-1} \quad (6.55)$$

and

$$\tilde{x}_{t-2}^n + J_{t-2} x_{t-1}^n = \tilde{x}_{t-2}^{t-2} + J_{t-2} \Phi x_{t-2}^{t-2}. \quad (6.56)$$

Next, multiply the left-hand side of (6.55) by the transpose of the left-hand side of (6.56), and equate that to the corresponding result of the right-hand sides of (6.55) and (6.56). Then, taking expectation of both sides, the left-hand side result reduces to

$$P_{t-1,t-2}^n + J_{t-1} E(x_t^n x_{t-1}^{n'}) J_{t-2}' \quad (6.57)$$

and the right-hand side result reduces to

$$\begin{aligned} P_{t-1,t-2}^{t-2} - K_{t-1} A_{t-1} P_{t-1,t-2}^{t-2} + J_{t-1} \Phi K_{t-1} A_{t-1} P_{t-1,t-2}^{t-2} \\ + J_{t-1} \Phi E(x_{t-1}^{t-1} x_{t-2}^{t-2'}) \Phi' J_{t-2}'. \end{aligned} \quad (6.58)$$

In (6.57), write

$$E(x_t^n x_{t-1}^{n'}) = E(x_t x_{t-1}') - P_{t,t-1}^n = \Phi E(x_{t-1} x_{t-2}') \Phi' + \Phi Q - P_{t,t-1}^n,$$

and in (6.58), write

$$E(x_{t-1}^{t-1} x_{t-2}^{t-2'}) = E(x_{t-1}^{t-2} x_{t-2}^{t-2'}) = E(x_{t-1} x_{t-2}') - P_{t-1,t-2}^{t-2}.$$

Equating (6.57) to (6.58) using these relationships and simplifying the result leads to (6.54). \square

6.3 Maximum Likelihood Estimation

Estimation of the parameters that specify the state space model, (6.3) and (6.4), is quite involved. We use Θ to represent the vector of unknown parameters in the initial mean and covariance μ_0 and Σ_0 , the transition matrix Φ , and the state and observation covariance matrices Q and R and the input coefficient matrices, Υ and Γ . We use maximum likelihood under the assumption that the initial state is normal, $x_0 \sim N_p(\mu_0, \Sigma_0)$, and the errors are normal, $w_t \sim \text{iid } N_p(0, Q)$ and $v_t \sim \text{iid } N_q(0, R)$. We continue to assume, for simplicity, $\{w_t\}$ and $\{v_t\}$ are uncorrelated.

The likelihood is computed using the *innovations* $\epsilon_1, \epsilon_2, \dots, \epsilon_n$, defined by (6.23),

$$\epsilon_t = y_t - A_t x_t^{t-1} - \Gamma u_t.$$

The innovations form of the likelihood of the data $y_{1:n}$, which was first given by Scheppe (1965), is obtained using an argument similar to the one leading to (3.117) and proceeds by noting the innovations are independent Gaussian random vectors with zero means and, as shown in (6.24), covariance matrices

$$\Sigma_t = A_t P_t^{t-1} A_t' + R. \quad (6.59)$$

Hence, ignoring a constant, we may write the likelihood, $L_Y(\Theta)$, as

$$-\ln L_Y(\Theta) = \frac{1}{2} \sum_{t=1}^n \ln |\Sigma_t(\Theta)| + \frac{1}{2} \sum_{t=1}^n \epsilon_t(\Theta)' \Sigma_t(\Theta)^{-1} \epsilon_t(\Theta), \quad (6.60)$$

where we have emphasized the dependence of the innovations on the parameters Θ . Of course, (6.60) is a highly nonlinear and complicated function of the unknown parameters. The usual procedure is to fix x_0 and then develop a set of recursions for the log likelihood function and its first two derivatives (for example, Gupta and Mehra, 1974). Then, a Newton–Raphson algorithm (see 3.30) can be used successively to update the parameter values until the negative of the log likelihood is minimized. This approach is advocated, for example, by Jones (1980), who developed ARMA estimation by putting the ARMA model in state-space form. For the univariate case, (6.60) is identical, in form, to the likelihood for the ARMA model given in (3.117).

The steps involved in performing a Newton–Raphson estimation procedure are as follows.

- (i) Select initial values for the parameters, say, $\Theta^{(0)}$.
- (ii) Run the Kalman filter, [Property 6.1](#), using the initial parameter values, $\Theta^{(0)}$, to obtain a set of innovations and error covariances, say, $\{\epsilon_t^{(0)}; t = 1, \dots, n\}$ and $\{\Sigma_t^{(0)}; t = 1, \dots, n\}$.
- (iii) Run one iteration of a Newton–Raphson procedure with $-\ln L_Y(\Theta)$ as the criterion function (refer to [Example 3.30](#) for details), to obtain a new set of estimates, say $\Theta^{(1)}$.
- (iv) At iteration j , ($j = 1, 2, \dots$), repeat step 2 using $\Theta^{(j)}$ in place of $\Theta^{(j-1)}$ to obtain a new set of innovation values $\{\epsilon_t^{(j)}; t = 1, \dots, n\}$ and $\{\Sigma_t^{(j)}; t = 1, \dots, n\}$.

Then repeat step 3 to obtain a new estimate $\boldsymbol{\theta}^{(j+1)}$. Stop when the estimates or the likelihood stabilize; for example, stop when the values of $\boldsymbol{\theta}^{(j+1)}$ differ from $\boldsymbol{\theta}^{(j)}$, or when $L_Y(\boldsymbol{\theta}^{(j+1)})$ differs from $L_Y(\boldsymbol{\theta}^{(j)})$, by some predetermined, but small amount.

Example 6.6 Newton–Raphson for Example 6.3

In this example, we generated $n = 100$ observations, $y_{1:100}$, from the AR with noise model given in [Example 6.3](#), to perform a Newton–Raphson estimation of the parameters ϕ , σ_w^2 , and σ_v^2 . In the notation of [Section 6.2](#), we would have $\Phi = \phi$, $Q = \sigma_w^2$ and $R = \sigma_v^2$. The actual values of the parameters are $\phi = .8$, $\sigma_w^2 = \sigma_v^2 = 1$.

In the simple case of an AR(1) with observational noise, initial estimation can be accomplished using the results of [Example 6.3](#). For example, using (6.15), we set

$$\phi^{(0)} = \hat{\rho}_y(2)/\hat{\rho}_y(1).$$

Similarly, from (6.14), $\gamma_x(1) = \gamma_y(1) = \phi\sigma_w^2/(1 - \phi^2)$, so that, initially, we set

$$\sigma_w^{2(0)} = (1 - \phi^{(0)^2})\hat{\gamma}_y(1)/\phi^{(0)}.$$

Finally, using (6.13) we obtain an initial estimate of σ_v^2 , namely,

$$\sigma_v^{2(0)} = \hat{\gamma}_y(0) - [\sigma_w^{2(0)} / (1 - \phi^{(0)^2})].$$

Newton–Raphson estimation was accomplished using the R program [optim](#). The code used for this example is given below. In that program, we must provide an evaluation of the function to be minimized, namely, $-\ln L_Y(\boldsymbol{\theta})$. In this case, the function call combines steps 2 and 3, using the current values of the parameters, $\boldsymbol{\theta}^{(j-1)}$, to obtain first the filtered values, then the innovation values, and then calculating the criterion function, $-\ln L_Y(\boldsymbol{\theta}^{(j-1)})$, to be minimized. We can also provide analytic forms of the gradient or *score vector*, $-\partial \ln L_Y(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$, and the *Hessian matrix*, $-\partial^2 \ln L_Y(\boldsymbol{\theta})/\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'$, in the optimization routine, or allow the program to calculate these values numerically. In this example, we let the program proceed numerically and we note the need to be cautious when calculating gradients numerically. It is suggested in Press et al. (1993, Ch. 10) that it is better to use numerical methods for the derivatives, at least for the Hessian, along with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. Details on the gradient and Hessian are provided in [Problem 6.9](#) and [Problem 6.10](#); see Gupta and Mehra (1974).

```
# Generate Data
set.seed(999); num = 100
x = arima.sim(n=num+1, list(ar=.8), sd=1)
y = ts(x[-1] + rnorm(num, 0, 1))
# Initial Estimates
u = ts.intersect(y, lag(y,-1), lag(y,-2))
varu = var(u); coru = cor(u)
phi = coru[1,3]/coru[1,2]
q = (1-phi^2)*varu[1,2]/phi
r = varu[1,1] - q/(1-phi^2)
```

```
(init.par = c(phi, sqrt(q), sqrt(r))) # = .91, .51, 1.03
# Function to evaluate the likelihood
Linn = function(para){
  phi = para[1]; sigw = para[2]; sigv = para[3]
  Sigma0 = (sigw^2)/(1-phi^2); Sigma0[Sigma0<0]=0
  kf = Kfilter0(num, y, 1, mu0=0, Sigma0, phi, sigw, sigv)
  return(kf$like) }
# Estimation (partial output shown)
(est = optim(init.par, Linn, gr=NULL, method='BFGS', hessian=TRUE,
             control=list(trace=1, REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
cbind(estimate=c(phi=est$par[1], sigw=est$par[2], sigv=est$par[3]), SE)
  estimate      SE
  phi    0.814  0.081
  sigw   0.851  0.175
  sigv   0.874  0.143
```

As seen from the output, the final estimates, along with their standard errors (in parentheses), are $\hat{\phi} = .81 (.08)$, $\hat{\sigma}_w = .85 (.18)$, $\hat{\sigma}_v = .87 (.14)$. The report from `optim` yielded the following results of the estimation procedure:

```
initial  value 81.313627
iter    2 value 80.169051
iter    3 value 79.866131
iter    4 value 79.222846
iter    5 value 79.021504
iter    6 value 79.014723
iter    7 value 79.014453
iter    7 value 79.014452
iter    7 value 79.014452
final   value 79.014452
converged
```

Note that the algorithm converged in seven steps with the final value of the negative of the log likelihood being 79.014452. The standard errors are a byproduct of the estimation procedure, and we will discuss their evaluation later in this section, after [Property 6.4](#).

Example 6.7 Newton–Raphson for the Global Temperature Deviations

In [Example 6.2](#), we considered two different global temperature series of $n = 136$ observations each, and they are plotted in [Figure 6.3](#). In that example, we argued that both series should be measuring the same underlying climatic signal, x_t , which we model as a random walk with drift,

$$x_t = \delta + x_{t-1} + w_t.$$

Recall that the observation equation was written as

$$\begin{pmatrix} y_{t1} \\ y_{t2} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} x_t + \begin{pmatrix} v_{t1} \\ v_{t2} \end{pmatrix},$$

and the model covariance matrices are given by $Q = q_{11}$ and

$$R = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix}.$$

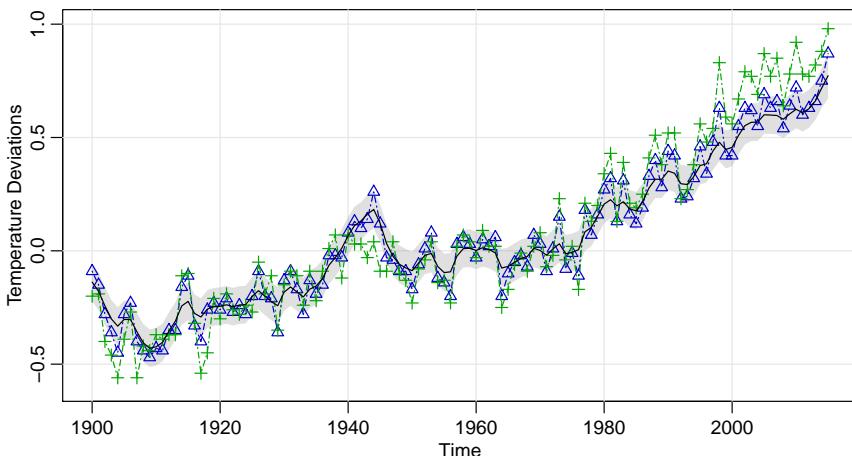


Fig. 6.5. Plot for Example 6.7. The dashed lines with points (+ and Δ) are the two average global temperature deviations shown in Figure 6.3. The solid line is the estimated smoother \hat{x}_t^n , and the corresponding two root mean square error bound is the gray swatch. Only the values later than 1900 are shown.

Hence, there are five parameters to estimate, δ , the drift, and the variance components, $q_{11}, r_{11}, r_{12}, r_{22}$, noting that $r_{21} = r_{12}$. We hold the initial state parameters fixed in this example at $\mu_0 = -.35$ and $\Sigma_0 = 1$, which is large relative to the data. The final estimates were (the R matrix is reassembled in the code).

	estimate	SE
sigw	0.055	0.011
cR11	0.074	0.010
cR22	0.127	0.015
cR12	0.129	0.038
drift	0.006	0.005

The observations and the smoothed estimate of the signal, $\hat{x}_t^n \pm 2\sqrt{\hat{P}_t^n}$, are displayed in Figure 6.5. The code, which uses `Kfilter1` and `Ksmooth1`, is as follows.

```
# Setup
y  = cbind(globtemp, globtempl); num = nrow(y); input = rep(1,num)
A  = array(rep(1,2), dim=c(2,1,num))
mu0 = -.35; Sigma0 = 1; Phi = 1
# Function to Calculate Likelihood
Linn = function(para){
  cQ  = para[1] # sigma_w
  cR1 = para[2] # 11 element of chol(R)
  cR2 = para[3] # 22 element of chol(R)
  cR12 = para[4] # 12 element of chol(R)
  cR  = matrix(c(cR1,0,cR12,cR2),2) # put the matrix together
  drift = para[5]
  kf  = Kfilter1(num,y,A,mu0,Sigma0,Phi,drift,0,cQ,cR,input)
  return(kf$like) }
# Estimation
init.par = c(.1,.1,.1,0,.05) # initial values of parameters
```

```

est = optim(init.par, Linn, NULL, method='BFGS', hessian=TRUE,
            control=list(trace=1,REPORT=1)) # output not shown
SE = sqrt(diag(solve(est$hessian)))
# Display estimates
u = cbind(estimate=est$par, SE)
rownames(u)=c('sigw', 'cR11', 'cR22', 'cR12', 'drift'); u
# Smooth (first set parameters to their final estimates)
cQ   = est$par[1]
cR1  = est$par[2]
cR2  = est$par[3]
cR12 = est$par[4]
cR   = matrix(c(cR1,0,cR12,cR2), 2)
(R   = t(cR)%*%cR)    # to view the estimated R matrix
drift = est$par[5]
ks   = Ksmooth1(num,y,A,mu0,Sigma0,Phi,drift,0,cQ,cR,input)
# Plot
xsm = ts(as.vector(ks$xs), start=1880)
rmse = ts(sqrt(as.vector(ks$Ps)), start=1880)
plot(xsm, ylim=c(-.6, 1), ylab='Temperature Deviations')
  xx = c(time(xsm), rev(time(xsm)))
  yy = c(xsm-2*rmse, rev(xsm+2*rmse))
  polygon(xx, yy, border=NA, col=gray(.6, alpha=.25))
lines(globtemp, type='o', pch=2, col=4, lty=6)
lines(globtempl, type='o', pch=3, col=3, lty=6)

```

In addition to Newton–Raphson, Shumway and Stoffer (1982) presented a conceptually simpler estimation procedure based on the Baum–Welch algorithm (Baum et al., 1970), also known as the EM (*expectation–maximization*) algorithm (Dempster et al., 1977). For the sake of brevity, we ignore the inputs and consider the model in the form of (6.1) and (6.2). The basic idea is that if we could observe the states, $x_{0:n} = \{x_0, x_1, \dots, x_n\}$, in addition to the observations $y_{1:n} = \{y_1, \dots, y_n\}$, then we would consider $\{x_{0:n}, y_{1:n}\}$ as the *complete data*, with joint density

$$p_{\Theta}(x_{0:n}, y_{1:n}) = p_{\mu_0, \Sigma_0}(x_0) \prod_{t=1}^n p_{\Phi, Q}(x_t | x_{t-1}) \prod_{t=1}^n p_R(y_t | x_t). \quad (6.61)$$

Under the Gaussian assumption and ignoring constants, the complete data likelihood, (6.61), can be written as

$$\begin{aligned} -2 \ln L_{X,Y}(\Theta) &= \ln |\Sigma_0| + (x_0 - \mu_0)' \Sigma_0^{-1} (x_0 - \mu_0) \\ &\quad + n \ln |Q| + \sum_{t=1}^n (x_t - \Phi x_{t-1})' Q^{-1} (x_t - \Phi x_{t-1}) \\ &\quad + n \ln |R| + \sum_{t=1}^n (y_t - A_t x_t)' R^{-1} (y_t - A_t x_t). \end{aligned} \quad (6.62)$$

Thus, in view of (6.62), if we did have the complete data, we could then use the results from multivariate normal theory to easily obtain the MLEs of Θ . Although we do not have the complete data, the EM algorithm gives us an iterative method for finding the MLEs of Θ based on the *incomplete data*, $y_{1:n}$, by successively maximizing

the conditional expectation of the complete data likelihood. To implement the EM algorithm, we write, at iteration j , ($j = 1, 2, \dots$),

$$\mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j-1)}) = E \left\{ -2 \ln L_{X,Y}(\boldsymbol{\theta}) \mid y_{1:n}, \boldsymbol{\theta}^{(j-1)} \right\}. \quad (6.63)$$

Calculation of (6.63) is the *expectation step*. Of course, given the current value of the parameters, $\boldsymbol{\theta}^{(j-1)}$, we can use [Property 6.2](#) to obtain the desired conditional expectations as smoothers. This property yields

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\theta} | \boldsymbol{\theta}^{(j-1)}) &= \ln |\Sigma_0| + \text{tr} \left\{ \Sigma_0^{-1} [P_0^n + (x_0^n - \mu_0)(x_0^n - \mu_0)'] \right\} \\ &\quad + n \ln |Q| + \text{tr} \left\{ Q^{-1} [S_{11} - S_{10}\Phi' - \Phi S_{10}' + \Phi S_{00}\Phi'] \right\} \\ &\quad + n \ln |R| + \text{tr} \left\{ R^{-1} \sum_{t=1}^n [(y_t - A_t x_t^n)(y_t - A_t x_t^n)' + A_t P_t^n A_t'] \right\}, \end{aligned} \quad (6.64)$$

where

$$S_{11} = \sum_{t=1}^n (x_t^n x_t^n)' + P_t^n, \quad (6.65)$$

$$S_{10} = \sum_{t=1}^n (x_t^n x_{t-1}^n)' + P_{t,t-1}^n, \quad (6.66)$$

and

$$S_{00} = \sum_{t=1}^n (x_{t-1}^n x_{t-1}^n)' + P_{t-1}^n. \quad (6.67)$$

In (6.64)–(6.67), the smoothers are calculated under the current value of the parameters $\boldsymbol{\theta}^{(j-1)}$; for simplicity, we have not explicitly displayed this fact. In obtaining $\mathcal{Q}(\cdot \mid \cdot)$, we made repeated use of fact $E(x_s x_t' \mid y_{1:n}) = x_s^n x_t^n' + P_{s,t}^n$; it is important to note that one does not simply replace x_t with x_t^n in the likelihood.

Minimizing (6.64) with respect to the parameters, at iteration j , constitutes the *maximization step*, and is analogous to the usual multivariate regression approach, which yields the updated estimates

$$\Phi^{(j)} = S_{10} S_{00}^{-1}, \quad (6.68)$$

$$Q^{(j)} = n^{-1} \left(S_{11} - S_{10} S_{00}^{-1} S_{10}' \right), \quad (6.69)$$

and

$$R^{(j)} = n^{-1} \sum_{t=1}^n [(y_t - A_t x_t^n)(y_t - A_t x_t^n)' + A_t P_t^n A_t']. \quad (6.70)$$

The updates for the initial mean and variance–covariance matrix are

$$\mu_0^{(j)} = x_0^n \quad \text{and} \quad \Sigma_0^{(j)} = P_0^n \quad (6.71)$$

obtained from minimizing (6.64).

The overall procedure can be regarded as simply alternating between the Kalman filtering and smoothing recursions and the multivariate normal maximum likelihood estimators, as given by (6.68)–(6.71). Convergence results for the EM algorithm under general conditions can be found in Wu (1983). A thorough discussion of the convergence of the EM algorithm and related methods may be found in Douc et al. (2014, Appendix D). We summarize the iterative procedure as follows.

- (i) Initialize by choosing starting values for the parameters in $\{\mu_0, \Sigma_0, \Phi, Q, R\}$, say $\Theta^{(0)}$, and compute the incomplete-data likelihood, $-\ln L_Y(\Theta^{(0)})$; see (6.60).

On iteration j , ($j = 1, 2, \dots$):

- (ii) Perform the E-Step: Using the parameters $\Theta^{(j-1)}$, use Properties 6.1, 6.2, and 6.3 to obtain the smoothed values x_t^n, P_t^n and $P_{t,t-1}^n$, $t = 1, \dots, n$, and calculate S_{11}, S_{10}, S_{00} given in (6.65)–(6.67).
- (iii) Perform the M-Step: Update the estimates in $\{\mu_0, \Sigma_0, \Phi, Q, R\}$ using (6.68)–(6.71), obtaining $\Theta^{(j)}$.
- (iv) Compute the incomplete-data likelihood, $-\ln L_Y(\Theta^{(j)})$.
- (v) Repeat Steps (ii) – (iv) to convergence.

Example 6.8 EM Algorithm for Example 6.3

Using the same data generated in Example 6.6, we performed an EM algorithm estimation of the parameters ϕ , σ_w^2 and σ_v^2 as well as the initial parameters μ_0 and Σ_0 using the script `EM0`. The convergence rate of the EM algorithm compared with the Newton–Raphson procedure is slow. In this example, with convergence being claimed when the relative change in the log likelihood is less than .00001; convergence was attained after 59 iterations. The final estimates, along with their standard errors are listed below and the results are close those in Example 6.6.

	estimate	SE
phi	0.810	0.078
sigw	0.853	0.164
sigv	0.864	0.136
mu0	-1.981	NA
Sigma0	0.022	NA

Evaluation of the standard errors used a call to `fdHess` in the `nlme` R package to evaluate the Hessian at the final estimates. The `nlme` package must be loaded prior to the call to `fdHess`.

```
library(nlme)    # loads package nlme
# Generate data (same as Example 6.6)
set.seed(999); num = 100
x = arima.sim(n=num+1, list(ar = .8, sd=1))
y = ts(x[-1] + rnorm(num, 0, 1))
# Initial Estimates (same as Example 6.6)
u = ts.intersect(y, lag(y,-1), lag(y,-2))
varu = var(u); coru = cor(u)
phi = coru[1,3]/coru[1,2]
q = (1-phi^2)*varu[1,2]/phi
r = varu[1,1] - q/(1-phi^2)
# EM procedure - output not shown
```

```

(em = EM0(num, y, A=1, mu0=0, Sigma0=2.8, Phi=phi, cQ=sqrt(q), cR=sqrt(r),
           max.iter=75, tol=.00001))
# Standard Errors (this uses nlme)
phi = em$Phi; cq = sqrt(em$Q); cr = sqrt(em$R)
mu0 = em$mu0; Sigma0 = em$Sigma0
para = c(phi, cq, cr)
Linn = function(para){ # to evaluate likelihood at estimates
  kf = Kfilter0(num, y, 1, mu0, Sigma0, para[1], para[2], para[3])
  return(kf$like)
}
emhess = fdHess(para, function(para) Linn(para))
SE = sqrt(diag(solve(emhess$Hessian)))
# Display Summary of Estimation
estimate = c(para, em$mu0, em$Sigma0); SE = c(SE, NA, NA)
u = cbind(estimate, SE)
rownames(u) = c('phi', 'sigw', 'sigv', 'mu0', 'Sigma0'); u

```

STEADY STATE AND ASYMPTOTIC DISTRIBUTION OF THE MLEs

The asymptotic distribution of estimators of the model parameters, say, $\widehat{\theta}_n$, is studied in very general terms in Douc, Moulines, and Stoffer (2014, Chapter 13). Earlier treatments can be found in Caines (1988, Chapters 7 and 8), and in Hannan and Deistler (1988, Chapter 4). In these references, the consistency and asymptotic normality of the estimators are established under general conditions. An essential condition is the stability of the filter. Stability of the filter assures that, for large t , the innovations ϵ_t are basically copies of each other with a stable covariance matrix Σ that does not depend on t and that, asymptotically, the innovations contain all of the information about the unknown parameters. Although it is not necessary, for simplicity, we shall assume here that $A_t \equiv A$ for all t . Details on departures from this assumption can be found in Jazwinski (1970, Sections 7.6 and 7.8). We also drop the inputs and use the model in the form of (6.1) and (6.2).

For stability of the filter, we assume the eigenvalues of Φ are less than one in absolute value; this assumption can be weakened (for example, see Harvey, 1991, Section 4.3), but we retain it for simplicity. This assumption is enough to ensure the stability of the filter in that, as $t \rightarrow \infty$, the filter error covariance matrix P_t^* converges to P , the steady-state error covariance matrix, and the gain matrix K_t converges to K , the steady-state gain matrix. From these facts, it follows that the innovation covariance matrix Σ_t converges to Σ , the steady-state covariance matrix of the stable innovations; details can be found in Jazwinski (1970, Sections 7.6 and 7.8) and Anderson and Moore (1979, Section 4.4). In particular, the steady-state filter error covariance matrix, P , satisfies the Riccati equation:

$$P = \Phi[P - PA'(APA' + R)^{-1}AP]\Phi' + Q;$$

the steady-state gain matrix satisfies $K = PA'[APA' + R]^{-1}$. In [Example 6.5](#) (see [Table 6.1](#)), for all practical purposes, stability was reached by the third observation.

When the process is in steady-state, we may consider x_{t+1}^t as the steady-state predictor and interpret it as $x_{t+1}^t = E(x_{t+1} | y_t, y_{t-1}, \dots)$. As can be seen from (6.18) and (6.20), the steady-state predictor can be written as

$$x_{t+1}^t = \Phi[I - KA]x_t^{t-1} + \Phi Ky_t = \Phi x_t^{t-1} + \Phi K\epsilon_t, \quad (6.72)$$

where ϵ_t is the steady-state innovation process given by

$$\epsilon_t = y_t - E(y_t \mid y_{t-1}, y_{t-2}, \dots).$$

In the Gaussian case, $\epsilon_t \sim \text{iid } N(0, \Sigma)$, where $\Sigma = APA' + R$. In steady-state, the observations can be written as

$$y_t = Ax_t^{t-1} + \epsilon_t. \quad (6.73)$$

Together, (6.72) and (6.73) make up the *steady-state innovations form* of the dynamic linear model.

In the following property, we assume the Gaussian state space model (6.1) and (6.2), is time invariant, i.e., $A_t \equiv A$, the eigenvalues of Φ are within the unit circle and the model has the smallest possible dimension (see Hannan and Deistler, 1988, Section 2.3 for details). We denote the true parameters by Θ_0 , and we assume the dimension of Θ_0 is the dimension of the parameter space. Although it is not necessary to assume w_t and v_t are Gaussian, certain additional conditions would have to apply and adjustments to the asymptotic covariance matrix would have to be made; see Douc et al. (2014, Chapter 13).

Property 6.4 Asymptotic Distribution of the Estimators

Under general conditions, let $\widehat{\Theta}_n$ be the estimator of Θ_0 obtained by maximizing the innovations likelihood, $L_Y(\Theta)$, as given in (6.60). Then, as $n \rightarrow \infty$,

$$\sqrt{n} (\widehat{\Theta}_n - \Theta_0) \xrightarrow{d} N [0, \mathcal{I}(\Theta_0)^{-1}],$$

where $\mathcal{I}(\Theta)$ is the asymptotic information matrix given by

$$\mathcal{I}(\Theta) = \lim_{n \rightarrow \infty} n^{-1} E [-\partial^2 \ln L_Y(\Theta) / \partial \Theta \partial \Theta'].$$

For a Newton procedure, the Hessian matrix (as described in Example 6.6) at the time of convergence can be used as an estimate of $n\mathcal{I}(\Theta_0)$ to obtain estimates of the standard errors. In the case of the EM algorithm, no derivatives are calculated, but we may include a numerical evaluation of the Hessian matrix at the time of convergence to obtain estimated standard errors. Also, extensions of the EM algorithm exist, such as the SEM algorithm (Meng and Rubin, 1991), that include a procedure for the estimation of standard errors. In the examples of this section, the estimated standard errors were obtained from the numerical Hessian matrix of $-\ln L_Y(\widehat{\Theta})$, where $\widehat{\Theta}$ is the vector of parameters estimates at the time of convergence.

6.4 Missing Data Modifications

An attractive feature available within the state space framework is its ability to treat time series that have been observed irregularly over time. For example, Jones (1980) used the state-space representation to fit ARMA models to series with missing observations, and Palma and Chan (1997) used the model for estimation and forecasting of

ARFIMA series with missing observations. Shumway and Stoffer (1982) described the modifications necessary to fit multivariate state-space models via the EM algorithm when data are missing. We will discuss the procedure in detail in this section. Throughout this section, for notational simplicity, we assume the model is of the form (6.1) and (6.2).

Suppose, at a given time t , we define the partition of the $q \times 1$ observation vector into two parts, $y_t^{(1)}$, the $q_{1t} \times 1$ component of observed values, and $y_t^{(2)}$, the $q_{2t} \times 1$ component of unobserved values, where $q_{1t} + q_{2t} = q$. Then, write the partitioned observation equation

$$\begin{pmatrix} y_t^{(1)} \\ y_t^{(2)} \end{pmatrix} = \begin{bmatrix} A_t^{(1)} \\ A_t^{(2)} \end{bmatrix} x_t + \begin{pmatrix} v_t^{(1)} \\ v_t^{(2)} \end{pmatrix}, \quad (6.74)$$

where $A_t^{(1)}$ and $A_t^{(2)}$ are, respectively, the $q_{1t} \times p$ and $q_{2t} \times p$ partitioned observation matrices, and

$$\text{cov} \begin{pmatrix} v_t^{(1)} \\ v_t^{(2)} \end{pmatrix} = \begin{bmatrix} R_{11t} & R_{12t} \\ R_{21t} & R_{22t} \end{bmatrix} \quad (6.75)$$

denotes the covariance matrix of the measurement errors between the observed and unobserved parts.

In the missing data case where $y_t^{(2)}$ is not observed, we may modify the observation equation in the DLM, (6.1)–(6.2), so that the model is

$$x_t = \Phi x_{t-1} + w_t \quad \text{and} \quad y_t^{(1)} = A_t^{(1)} x_t + v_t^{(1)}, \quad (6.76)$$

where now, the observation equation is q_{1t} -dimensional at time t . In this case, it follows directly from Corollary 6.1 that the filter equations hold with the appropriate notational substitutions. If there are no observations at time t , then set the gain matrix, K_t , to the $p \times q$ zero matrix in Property 6.1, in which case $x_t^t = x_t^{t-1}$ and $P_t^t = P_t^{t-1}$.

Rather than deal with varying observational dimensions, it is computationally easier to modify the model by zeroing out certain components and retaining a q -dimensional observation equation throughout. In particular, Corollary 6.1 holds for the missing data case if, at update t , we substitute

$$y_{(t)} = \begin{pmatrix} y_t^{(1)} \\ 0 \end{pmatrix}, \quad A_{(t)} = \begin{bmatrix} A_t^{(1)} \\ 0 \end{bmatrix}, \quad R_{(t)} = \begin{bmatrix} R_{11t} & 0 \\ 0 & I_{22t} \end{bmatrix}, \quad (6.77)$$

for y_t , A_t , and R , respectively, in (6.20)–(6.22), where I_{22t} is the $q_{2t} \times q_{2t}$ identity matrix. With the substitutions (6.77), the innovation values (6.23) and (6.24) will now be of the form

$$\epsilon_{(t)} = \begin{pmatrix} \epsilon_t^{(1)} \\ 0 \end{pmatrix}, \quad \Sigma_{(t)} = \begin{bmatrix} A_t^{(1)} P_t^{t-1} A_t^{(1)'} + R_{11t} & 0 \\ 0 & I_{22t} \end{bmatrix}, \quad (6.78)$$

so that the innovations form of the likelihood given in (6.60) is correct for this case. Hence, with the substitutions in (6.77), maximum likelihood estimation via the innovations likelihood can proceed as in the complete data case.

Once the missing data filtered values have been obtained, Stoffer (1982) also established the smoother values can be processed using [Property 6.2](#) and [Property 6.3](#) with the values obtained from the missing data-filtered values. In the missing data case, the state estimators are denoted

$$x_t^{(s)} = E \left(x_t \mid y_1^{(1)}, \dots, y_s^{(1)} \right), \quad (6.79)$$

with error variance–covariance matrix

$$P_t^{(s)} = E \left\{ \left(x_t - x_t^{(s)} \right) \left(x_t - x_t^{(s)} \right)' \right\}. \quad (6.80)$$

The missing data lag-one smoother covariances will be denoted by $P_{t,t-1}^{(n)}$.

The maximum likelihood estimators in the EM procedure require further modifications for the case of missing data. Now, we consider

$$y_{1:n}^{(1)} = \{y_1^{(1)}, \dots, y_n^{(1)}\} \quad (6.81)$$

as the incomplete data, and $\{x_{0:n}, y_{1:n}\}$, as defined in [\(6.61\)](#), as the complete data. In this case, the complete data likelihood, [\(6.61\)](#), or equivalently [\(6.62\)](#), is the same, but to implement the E-step, at iteration j , we must calculate

$$\begin{aligned} Q(\Theta \mid \Theta^{(j-1)}) &= E \left\{ -2 \ln L_{X,Y}(\Theta) \mid y_{1:n}^{(1)}, \Theta^{(j-1)} \right\} \\ &= E_* \left\{ \ln |\Sigma_0| + \text{tr } \Sigma_0^{-1} (x_0 - \mu_0)(x_0 - \mu_0)' \mid y_{1:n}^{(1)} \right\} \\ &\quad + E_* \left\{ n \ln |Q| + \sum_{t=1}^n \text{tr} [Q^{-1} (x_t - \Phi x_{t-1})(x_t - \Phi x_{t-1})'] \mid y_{1:n}^{(1)} \right\} \\ &\quad + E_* \left\{ n \ln |R| + \sum_{t=1}^n \text{tr} [R^{-1} (y_t - A_t x_t)(y_t - A_t x_t)'] \mid y_{1:n}^{(1)} \right\}, \end{aligned} \quad (6.82)$$

where E_* denotes the conditional expectation under $\Theta^{(j-1)}$ and tr denotes trace. The first two terms in [\(6.82\)](#) will be like the first two terms of [\(6.64\)](#) with the smoothers x_t^n , P_t^n , and $P_{t,t-1}^n$ replaced by their missing data counterparts, $x_t^{(n)}$, $P_t^{(n)}$, and $P_{t,t-1}^{(n)}$. In the third term of [\(6.82\)](#), we must additionally evaluate $E_*(y_t^{(2)} \mid y_{1:n}^{(1)})$ and $E_*(y_t^{(2)} y_t^{(2)'} \mid y_{1:n}^{(1)})$. In Stoffer (1982), it is shown that

$$\begin{aligned} E_* \left\{ (y_t - A_t x_t)(y_t - A_t x_t)' \mid y_{1:n}^{(1)} \right\} \\ = \begin{pmatrix} y_t^{(1)} - A_t^{(1)} x_t^{(n)} \\ R_{*21t} R_{*11t}^{-1} (y_t^{(1)} - A_t^{(1)} x_t^{(n)}) \end{pmatrix} \begin{pmatrix} y_t^{(1)} - A_t^{(1)} x_t^{(n)} \\ R_{*21t} R_{*11t}^{-1} (y_t^{(1)} - A_t^{(1)} x_t^{(n)}) \end{pmatrix}' \\ + \begin{pmatrix} A_t^{(1)} \\ R_{*21t} R_{*11t}^{-1} A_t^{(1)} \end{pmatrix} P_t^{(n)} \begin{pmatrix} A_t^{(1)} \\ R_{*21t} R_{*11t}^{-1} A_t^{(1)} \end{pmatrix}' \\ + \begin{pmatrix} 0 & 0 \\ 0 & R_{*22t} - R_{*21t} R_{*11t}^{-1} R_{*12t} \end{pmatrix}. \end{aligned} \quad (6.83)$$

In (6.83), the values of R_{*ikt} , for $i, k = 1, 2$, are the current values specified by $\Theta^{(j-1)}$. In addition, $x_t^{(n)}$ and $P_t^{(n)}$ are the values obtained by running the smoother under the current parameter estimates specified by $\Theta^{(j-1)}$.

In the case in which observed and unobserved components have uncorrelated errors, that is, R_{*12t} is the zero matrix, (6.83) can be simplified to

$$\begin{aligned} \text{E}_* \{ & (y_t - A_t x_t)(y_t - A_t x_t)' \mid y_{1:n}^{(1)} \} \\ &= (y_{(t)} - A_{(t)} x_t^{(n)}) (y_{(t)} - A_{(t)} x_t^{(n)})' + A_{(t)} P_t^{(n)} A'_{(t)} + \begin{pmatrix} 0 & 0 \\ 0 & R_{*22t} \end{pmatrix}, \quad (6.84) \end{aligned}$$

where $y_{(t)}$ and $A_{(t)}$ are defined in (6.77).

In this simplified case, the missing data M-step looks like the M-step given in (6.65)-(6.71). That is, with

$$S_{(11)} = \sum_{t=1}^n (x_t^{(n)} x_t^{(n)'} + P_t^{(n)}), \quad (6.85)$$

$$S_{(10)} = \sum_{t=1}^n (x_t^{(n)} x_{t-1}^{(n)'} + P_{t,t-1}^{(n)}), \quad (6.86)$$

and

$$S_{(00)} = \sum_{t=1}^n (x_{t-1}^{(n)} x_{t-1}^{(n)'} + P_{t-1}^{(n)}), \quad (6.87)$$

where the smoothers are calculated under the present value of the parameters $\Theta^{(j-1)}$ using the missing data modifications, at iteration j , the *maximization step* is

$$\Phi^{(j)} = S_{(10)} S_{(00)}^{-1}, \quad (6.88)$$

$$Q^{(j)} = n^{-1} \left(S_{(11)} - S_{(10)} S_{(00)}^{-1} S'_{(10)} \right), \quad (6.89)$$

and

$$\begin{aligned} R^{(j)} = n^{-1} \sum_{t=1}^n D_t & \left\{ \left(y_{(t)} - A_{(t)} x_t^{(n)} \right) \left(y_{(t)} - A_{(t)} x_t^{(n)} \right)' \right. \\ & \left. + A_{(t)} P_t^{(n)} A'_{(t)} + \begin{pmatrix} 0 & 0 \\ 0 & R_{22t}^{(j-1)} \end{pmatrix} \right\} D'_t, \quad (6.90) \end{aligned}$$

where D_t is a permutation matrix that reorders the variables at time t in their original order and $y_{(t)}$ and $A_{(t)}$ are defined in (6.77). For example, suppose $q = 3$ and at time t , y_{t2} is missing. Then,

$$y_{(t)} = \begin{pmatrix} y_{t1} \\ y_{t3} \\ 0 \end{pmatrix}, \quad A_{(t)} = \begin{bmatrix} A_{t1} \\ A_{t3} \\ 0' \end{bmatrix}, \quad \text{and} \quad D_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

where A_{ti} is the i th row of A_t and $0'$ is a $1 \times p$ vector of zeros. In (6.90), only R_{11t} gets updated, and R_{22t} at iteration j is simply set to its value from the previous iteration, $j - 1$. Of course, if we cannot assume $R_{12t} = 0$, (6.90) must be changed accordingly using (6.83), but (6.88) and (6.89) remain the same. As before, the parameter estimates for the initial state are updated as

$$\mu_0^{(j)} = x_0^{(n)} \quad \text{and} \quad \Sigma_0^{(j)} = P_0^{(n)}. \quad (6.91)$$

Example 6.9 Longitudinal Biomedical Data

We consider the biomedical data in Example 6.1, which have portions of the three-dimensional vector missing after the 40th day. The maximum likelihood procedure yielded the estimators (code at the end of the example):

```
$Phi
      [,1]   [,2]   [,3]
[1,]  0.984 -0.041  0.009
[2,]  0.061  0.921  0.007
[3,] -1.495  2.289  0.794

$Q
      [,1]   [,2]   [,3]
[1,]  0.014 -0.002  0.012
[2,] -0.002  0.003  0.018
[3,]  0.012  0.018  3.494

$R
      [,1]   [,2]   [,3]
[1,]  0.007  0.000  0.000
[2,]  0.000  0.017  0.000
[3,]  0.000  0.000  1.147
```

for the transition, state error covariance and observation error covariance matrices, respectively. The coupling between the first and second series is relatively weak, whereas the third series HCT is strongly related to the first two; that is,

$$\hat{x}_{t3} = -1.495x_{t-1,1} + 2.289x_{t-1,2} + .794x_{t-1,3}.$$

Hence, the HCT is negatively correlated with white blood count (WBC) and positively correlated with platelet count (PLT). Byproducts of the procedure are estimated trajectories for all three longitudinal series and their respective prediction intervals. In particular, Figure 6.6 shows the data as points, the estimated smoothed values $\hat{x}_t^{(n)}$ as solid lines, and error bounds, $\pm 2\sqrt{\hat{P}_t^{(n)}}$ as a gray swatch.

In the following R code we use the script EM1. In this case the observation matrices A_t are either the identity or zero matrix because all the series are either observed or not observed.

```
y = cbind(WBC, PLT, HCT); num = nrow(y)
# make array of obs matrices
A = array(0, dim=c(3,3,num))
for(k in 1:num) { if (y[k,1] > 0) A[, ,k] = diag(1,3) }
# Initial values
mu0 = matrix(0, 3, 1); Sigma0 = diag(c(.1, .1, 1), 3)
Phi = diag(1, 3); cQ = diag(c(.1, .1, 1), 3); cR = diag(c(.1, .1, 1), 3)
# EM procedure - some output previously shown
```

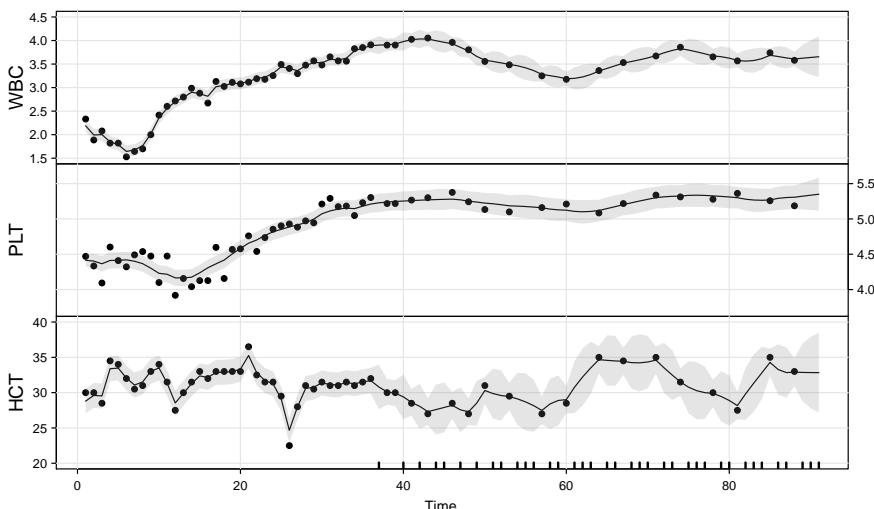


Fig. 6.6. Smoothed values for various components in the blood parameter tracking problem. The actual data are shown as points, the smoothed values are shown as solid lines, and ± 2 standard error bounds are shown as a gray swatch; tick marks indicate days with no observation.

```
(em = EM1(num, y, A, mu0, Sigma0, Phi, cQ, cR, 100, .001))
# Graph smoother
ks = Ksmooth1(num, y, A, em$mu0, em$Sigma0, em$Phi, 0, 0, chol(em$Q),
               chol(em$R), 0)
y1s = ks$xs[1,,]; y2s = ks$xs[2,,]; y3s = ks$xs[3,,]
p1 = 2*sqrt(ks$Ps[1,1]); p2 = 2*sqrt(ks$Ps[2,2]); p3 = 2*sqrt(ks$Ps[3,3])
par(mfrow=c(3,1))
plot(WBC, type='p', pch=19, ylim=c(1,5), xlab='day')
lines(y1s); lines(y1s+p1, lty=2, col=4); lines(y1s-p1, lty=2, col=4)
plot(PLT, type='p', ylim=c(3,6), pch=19, xlab='day')
lines(y2s); lines(y2s+p2, lty=2, col=4); lines(y2s-p2, lty=2, col=4)
plot(HCT, type='p', pch=19, ylim=c(20,40), xlab='day')
lines(y3s); lines(y3s+p3, lty=2, col=4); lines(y3s-p3, lty=2, col=4)
```

6.5 Structural Models: Signal Extraction and Forecasting

Structural models are component models in which each component may be thought of as explaining a specific type of behavior. The models are often some version of the classical time series decomposition of data into trend, seasonal, and irregular components. Consequently, each component has a direct interpretation as to the nature of the variation in the data. Furthermore, the model fits into the state space framework quite easily. To illustrate these ideas , we consider an example that shows how to fit a sum of trend, seasonal, and irregular components to the quarterly earnings data that we have considered before.

Example 6.10 Johnson & Johnson Quarterly Earnings

Here, we focus on the quarterly earnings series from the U.S. company Johnson & Johnson as displayed in [Figure 1.1](#). The series is highly nonstationary, and there is both a trend signal that is gradually increasing over time and a seasonal component that cycles every four quarters or once per year. The seasonal component is getting larger over time as well. Transforming into logarithms or even taking the n th root does not seem to make the series trend stationary, however, such a transformation does help with stabilizing the variance over time; this is explored in [Problem 6.13](#). Suppose, for now, we consider the series to be the sum of a trend component, a seasonal component, and a white noise. That is, let the observed series be expressed as

$$y_t = T_t + S_t + v_t, \quad (6.92)$$

where T_t is trend and S_t is the seasonal component. Suppose we allow the trend to increase exponentially; that is,

$$T_t = \phi T_{t-1} + w_{t1}, \quad (6.93)$$

where the coefficient $\phi > 1$ characterizes the increase. Let the seasonal component be modeled as

$$S_t + S_{t-1} + S_{t-2} + S_{t-3} = w_{t2}, \quad (6.94)$$

which corresponds to assuming the component is expected to sum to zero over a complete period or four quarters. To express this model in state-space form, let $x_t = (T_t, S_t, S_{t-1}, S_{t-2})'$ be the state vector so the observation equation [\(6.2\)](#) can be written as

$$y_t = (1 \ 1 \ 0 \ 0) \begin{pmatrix} T_t \\ S_t \\ S_{t-1} \\ S_{t-2} \end{pmatrix} + v_t,$$

with the state equation written as

$$\begin{pmatrix} T_t \\ S_t \\ S_{t-1} \\ S_{t-2} \end{pmatrix} = \begin{pmatrix} \phi & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} T_{t-1} \\ S_{t-1} \\ S_{t-2} \\ S_{t-3} \end{pmatrix} + \begin{pmatrix} w_{t1} \\ w_{t2} \\ 0 \\ 0 \end{pmatrix},$$

where $R = r_{11}$ and

$$Q = \begin{pmatrix} q_{11} & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The model reduces to state-space form, [\(6.1\)](#) and [\(6.2\)](#), with $p = 4$ and $q = 1$. The parameters to be estimated are r_{11} , the noise variance in the measurement equations, q_{11} and q_{22} , the model variances corresponding to the trend and seasonal components and ϕ , the transition parameter that models the growth rate. Growth

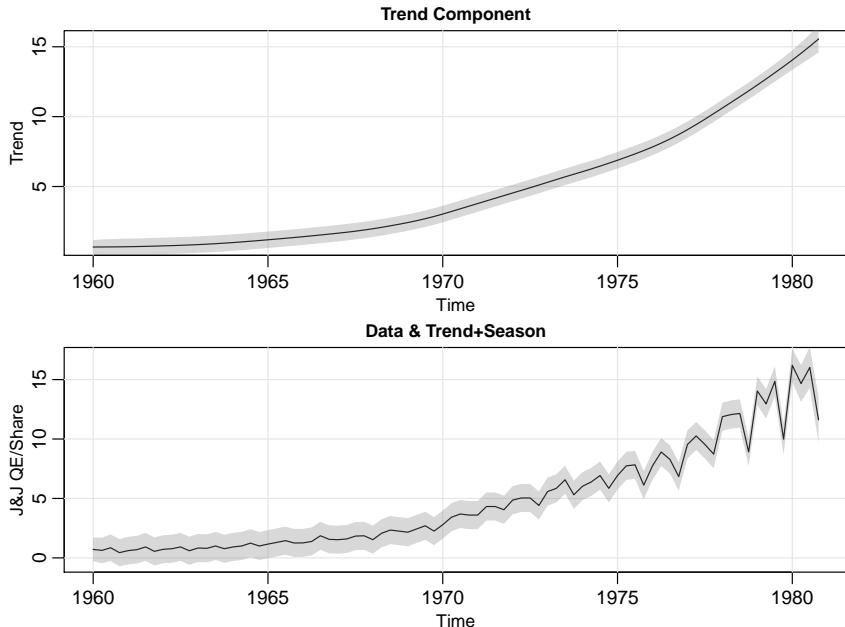


Fig. 6.7. Estimated trend component, T_t^n , and seasonal component, S_t^n , of the Johnson and Johnson quarterly earnings series. Gray areas are three root MSE bounds.

is about 3% per year, and we began with $\phi = 1.03$. The initial mean was fixed at $\mu_0 = (.7, 0, 0, 0)'$, with uncertainty modeled by the diagonal covariance matrix with $\Sigma_{0ii} = .04$, for $i = 1, \dots, 4$. Initial state covariance values were taken as $q_{11} = .01$, $q_{22} = .01$. The measurement error covariance was started at $r_{11} = .25$.

After about 20 iterations of a Newton–Raphson, the transition parameter estimate was $\hat{\phi} = 1.035$, corresponding to exponential growth with inflation at about 3.5% per year. The measurement uncertainty was small at $\sqrt{r_{11}} = .0005$, compared with the model uncertainties $\sqrt{\hat{q}_{11}} = .1397$ and $\sqrt{\hat{q}_{22}} = .2209$. Figure 6.7 shows the smoothed trend estimate and the exponentially increasing seasonal components. We may also consider forecasting the Johnson & Johnson series, and the result of a 12-quarter forecast is shown in Figure 6.8 as basically an extension of the latter part of the observed data.

This example uses the `Kfilter0` and `Ksmooth0` scripts as follows.

```

num = length(jj)
A = cbind(1,1,0,0)
# Function to Calculate Likelihood
Linn=function(para){
  Phi = diag(0,4); Phi[1,1] = para[1]
  Phi[2,]=c(0,-1,-1,-1); Phi[3,]=c(0,1,0,0); Phi[4,]=c(0,0,1,0)
  cQ1 = para[2]; cQ2 = para[3]      # sqrt q11 and q22
  cQ  = diag(0,4); cQ[1,1]=cQ1; cQ[2,2]=cQ2
  cR  = para[4]                      # sqrt r11
  kf = Kfilter0(num, jj, A, mu0, Sigma0, Phi, cQ, cR)
}

```

```

    return(kf$like)  }
# Initial Parameters
mu0 = c(.7,0,0,0); Sigma0 = diag(.04,4)
init.par = c(1.03,.1,.1,.5)           # Phi[1,1], the 2 cQs and cR
# Estimation and Results
est = optim(init.par, Linn,NULL, method='BFGS', hessian=TRUE,
            control=list(trace=1,REPORT=1))
SE = sqrt(diag(solve(est$hessian)))
u = cbind(estimate=est$par, SE)
rownames(u)=c('Phi11','sigw1','sigw2','sigv'); u
# Smooth
Phi = diag(0,4); Phi[1,1] = est$par[1]
Phi[2,]=c(0,-1,-1,-1); Phi[3,]=c(0,1,0,0); Phi[4,]=c(0,0,1,0)
cQ1 = est$par[2]; cQ2 = est$par[3]
cQ = diag(1,4); cQ[1,1]=cQ1; cQ[2,2]=cQ2
cR = est$par[4]
ks = Ksmooth0(num,jj,A,mu0,Sigma0,Phi,cQ,cR)
# Plots
Tsm = ts(ks$xs[,], start=1960, freq=4)
Ssm = ts(ks$xs[,], start=1960, freq=4)
p1 = 3*sqrt(ks$Ps[1,1,]); p2 = 3*sqrt(ks$Ps[2,2,])
par(mfrow=c(2,1))
plot(Tsm, main='Trend Component', ylab='Trend')
xx = c(time(jj), rev(time(jj)))
yy = c(Tsm-p1, rev(Tsm+p1))
polygon(xx, yy, border=NA, col=gray(.5, alpha = .3))
plot(jj, main='Data & Trend+Season', ylab='J&J QE/Share', ylim=c(-.5,17))
xx = c(time(jj), rev(time(jj)))
yy = c((Tsm+Ssm)-(p1+p2), rev((Tsm+Ssm)+(p1+p2)))
polygon(xx, yy, border=NA, col=gray(.5, alpha = .3))
# Forecast
n.ahead = 12;
y = ts(append(jj, rep(0,n.ahead)), start=1960, freq=4)
rmspe = rep(0,n.ahead); x00 = ks$xf[,num]; P00 = ks$Pf[,num]
Q = t(cQ)%*%cQ; R = t(cR)%*%(cR)
for (m in 1:n.ahead){
  xp = Phi%*%x00; Pp = Phi%*%P00%*%t(Phi)+Q
  sig = A%*%Pp%*%t(A)+R; K = Pp%*%t(A)%*%(1/sig)
  x00 = xp; P00 = Pp-K%*%A%*%Pp
  y[num+m] = A%*%xp; rmspe[m] = sqrt(sig)  }
plot(y, type='o', main='', ylab='J&J QE/Share', ylim=c(5,30),
      xlim=c(1975,1984))
upp = ts(y[(num+1):(num+n.ahead)]+2*rmspe, start=1981, freq=4)
low = ts(y[(num+1):(num+n.ahead)]-2*rmspe, start=1981, freq=4)
xx = c(time(low), rev(time(upp)))
yy = c(low, rev(upp))
polygon(xx, yy, border=8, col=gray(.5, alpha = .3))
abline(v=1981, lty=3)

```

Note that the Cholesky decomposition of `Q` does not exist here, however, the diagonal form allows us to use standard deviations for the first two diagonal elements of `cQ`. This technicality can be avoided using a form of the model that we present in the next section.

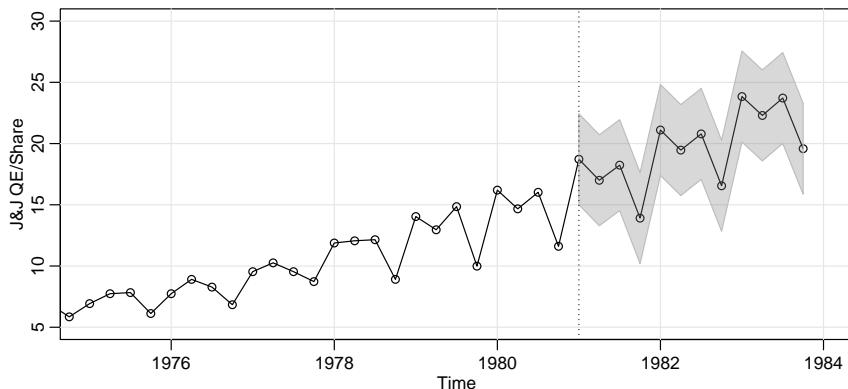


Fig. 6.8. A 12-quarter forecast for the Johnson & Johnson quarterly earnings series. The forecasts are shown as a continuation of the data (points connected by a solid line). The gray area represents two root MSPE bounds.

6.6 State-Space Models with Correlated Errors

Sometimes it is advantageous to write the state-space model in a slightly different way, as is done by numerous authors; for example, Anderson and Moore (1979) and Hannan and Deistler (1988). Here, we write the state-space model as

$$x_{t+1} = \Phi x_t + \Upsilon u_{t+1} + \Theta w_t \quad t = 0, 1, \dots, n \quad (6.95)$$

$$y_t = A_t x_t + \Gamma u_t + v_t \quad t = 1, \dots, n \quad (6.96)$$

where, in the state equation, $x_0 \sim N_p(\mu_0, \Sigma_0)$, Φ is $p \times p$, and Υ is $p \times r$, Θ is $p \times m$ and $w_t \sim \text{iid } N_m(0, Q)$. In the observation equation, A_t is $q \times p$ and Γ is $q \times r$, and $v_t \sim \text{iid } N_q(0, R)$. In this model, while w_t and v_t are still white noise series (both independent of x_0), we also allow the state noise and observation noise to be correlated at time t ; that is,

$$\text{cov}(w_s, v_t) = S \delta_s^t, \quad (6.97)$$

where δ_s^t is Kronecker's delta; note that S is an $m \times q$ matrix. The major difference between this form of the model and the one specified by (6.3)–(6.4) is that this model starts the state noise process at $t = 0$ in order to ease the notation related to the concurrent covariance between w_t and v_t . Also, the inclusion of the matrix Θ allows us to avoid using a singular state noise process as was done in Example 6.10.

To obtain the innovations, $\epsilon_t = y_t - A_t x_t^{t-1} - \Gamma u_t$, and the innovation variance $\Sigma_t = A_t P_t^{t-1} A_t' + R$, in this case, we need the one-step-ahead state predictions. Of course, the filtered estimates will also be of interest, and they will be needed for smoothing. **Property 6.2** (the smoother) as displayed in Section 6.2 still holds. The following property generates the predictor x_{t+1}^t from the past predictor x_t^{t-1} when the noise terms are correlated and exhibits the filter update.

Property 6.5 The Kalman Filter with Correlated Noise

For the state-space model specified in (6.95) and (6.96), with initial conditions x_1^0 and P_1^0 , for $t = 1, \dots, n$,

$$x_{t+1}^t = \Phi x_t^{t-1} + \Upsilon u_{t+1} + K_t \epsilon_t \quad (6.98)$$

$$P_{t+1}^t = \Phi P_t^{t-1} \Phi' + \Theta Q \Theta' - K_t \Sigma_t K_t' \quad (6.99)$$

where $\epsilon_t = y_t - A_t x_t^{t-1} - \Gamma u_t$ and the gain matrix is given by

$$K_t = [\Phi P_t^{t-1} A_t' + \Theta S][A_t P_t^{t-1} A_t' + R]^{-1}. \quad (6.100)$$

The filter values are given by

$$x_t^t = x_t^{t-1} + P_t^{t-1} A_t' [A_t P_t^{t-1} A_t' + R]^{-1} \epsilon_t, \quad (6.101)$$

$$P_t^t = P_t^{t-1} - P_t^{t-1} A_{t+1}' [A_t P_t^{t-1} A_t' + R]^{-1} A_t P_t^{t-1}. \quad (6.102)$$

The derivation of Property 6.5 is similar to the derivation of the Kalman filter in Property 6.1 (Problem 6.17); we note that the gain matrix K_t differs in the two properties. The filter values, (6.101)–(6.102), are symbolically identical to (6.18) and (6.19). To initialize the filter, we note that

$$x_1^0 = E(x_1) = \Phi \mu_0 + \Upsilon u_1, \quad \text{and} \quad P_1^0 = \text{var}(x_1) = \Phi \Sigma_0 \Phi' + \Theta Q \Theta'.$$

In the next two subsections, we show how to use the model (6.95)–(6.96) for fitting ARMAX models and for fitting (multivariate) regression models with autocorrelated errors. To put it succinctly, for ARMAX models, the inputs enter in the state equation and for regression with autocorrelated errors, the inputs enter in the observation equation. It is, of course, possible to combine the two models and we give an example of this at the end of the section.

6.6.1 ARMAX Models

Consider a k -dimensional ARMAX model given by

$$y_t = \Upsilon u_t + \sum_{j=1}^p \Phi_j y_{t-j} + \sum_{k=1}^q \Theta_k v_{t-k} + v_t. \quad (6.103)$$

The observations y_t are a k -dimensional vector process, the Φ s and Θ s are $k \times k$ matrices, Υ is $k \times r$, u_t is the $r \times 1$ input, and v_t is a $k \times 1$ white noise process; in fact, (6.103) and (5.91) are identical models, but here, we have written the observations as y_t . We now have the following property.

Property 6.6 A State-Space Form of ARMAX

For $p \geq q$, let

$$F = \begin{bmatrix} \Phi_1 & I & 0 & \cdots & 0 \\ \Phi_2 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_{p-1} & 0 & 0 & \cdots & I \\ \Phi_p & 0 & 0 & \cdots & 0 \end{bmatrix} \quad G = \begin{bmatrix} \Theta_1 + \Phi_1 \\ \vdots \\ \Theta_q + \Phi_q \\ \Phi_{q+1} \\ \vdots \\ \Phi_p \end{bmatrix} \quad H = \begin{bmatrix} \gamma \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (6.104)$$

where F is $kp \times kp$, G is $kp \times k$, and H is $kp \times r$. Then, the state-space model given by

$$x_{t+1} = Fx_t + Hu_{t+1} + Gv_t, \quad (6.105)$$

$$y_t = Ax_t + v_t, \quad (6.106)$$

where $A = [I, 0, \dots, 0]$ is $k \times pk$ and I is the $k \times k$ identity matrix, implies the ARMAX model (6.103). If $p < q$, set $\Phi_{p+1} = \dots = \Phi_q = 0$, in which case $p = q$ and (6.105)–(6.106) still apply. Note that the state process is kp -dimensional, whereas the observations are k -dimensional.

We do not prove Property 6.6 directly, but the following example should suggest how to establish the general result.

Example 6.11 Univariate ARMAX(1, 1) in State-Space Form

Consider the univariate ARMAX(1, 1) model

$$y_t = \alpha_t + \phi y_{t-1} + \theta v_{t-1} + v_t,$$

where $\alpha_t = \gamma u_t$ to ease the notation. For a simple example, if $\gamma = (\beta_0, \beta_1)$ and $u_t = (1, t)'$, the model for y_t would be ARMA(1,1) with linear trend, $y_t = \beta_0 + \beta_1 t + \phi y_{t-1} + \theta v_{t-1} + v_t$. Using Property 6.6, we can write the model as

$$x_{t+1} = \phi x_t + \alpha_{t+1} + (\theta + \phi)v_t, \quad (6.107)$$

and

$$y_t = x_t + v_t. \quad (6.108)$$

In this case, (6.107) is the state equation with $w_t \equiv v_t$ and (6.108) is the observation equation. Consequently, $\text{cov}(w_t, v_t) = \text{var}(v_t) = R$, and $\text{cov}(w_t, v_s) = 0$ when $s \neq t$, so Property 6.5 would apply. To verify (6.107) and (6.108) specify an ARMAX(1, 1) model, we have

$$\begin{aligned} y_t &= x_t + v_t && \text{from (6.108)} \\ &= \phi x_{t-1} + \alpha_t + (\theta + \phi)v_{t-1} + v_t && \text{from (6.107)} \\ &= \alpha_t + \phi(x_{t-1} + v_{t-1}) + \theta v_{t-1} + v_t && \text{rearrange terms} \\ &= \alpha_t + \phi y_{t-1} + \theta v_{t-1} + v_t, && \text{from (6.108).} \end{aligned}$$

Together, [Property 6.5](#) and [Property 6.6](#) can be used to accomplish maximum likelihood estimation as described in [Section 6.3](#) for ARMAX models. The ARMAX model is only a special case of the model (6.95)–(6.96), which is quite rich, as will be discovered in the next subsection.

6.6.2 Multivariate Regression with Autocorrelated Errors

In regression with autocorrelated errors, we are interested in fitting the regression model

$$y_t = \Gamma u_t + \varepsilon_t \quad (6.109)$$

to a $k \times 1$ vector process, y_t , with r regressors $u_t = (u_{t1}, \dots, u_{tr})'$ where ε_t is vector ARMA(p, q) and Γ is a $k \times r$ matrix of regression parameters. We note that the regressors do not have to vary with time (e.g., $u_{t1} \equiv 1$ includes a constant in the regression) and that the case $k = 1$ was treated in [Section 3.8](#).

To put the model in state-space form, we simply notice that $\varepsilon_t = y_t - \Gamma u_t$ is a k -dimensional ARMA(p, q) process. Thus, if we set $H = 0$ in (6.105), and include Γu_t in (6.106), we obtain

$$x_{t+1} = F x_t + G v_t, \quad (6.110)$$

$$y_t = \Gamma u_t + A x_t + v_t, \quad (6.111)$$

where the model matrices A , F , and G are defined in [Property 6.6](#). The fact that (6.110)–(6.111) is multivariate regression with autocorrelated errors follows directly from [Property 6.6](#) by noticing that together, $x_{t+1} = F x_t + G v_t$ and $\varepsilon_t = A x_t + v_t$ imply $\varepsilon_t = y_t - \Gamma u_t$ is vector ARMA(p, q).

As in the case of ARMAX models, regression with autocorrelated errors is a special case of the state-space model, and the results of [Property 6.5](#) can be used to obtain the innovations form of the likelihood for parameter estimation.

Example 6.12 Mortality, Temperature and Pollution

This example combines both techniques of [Section 6.6.1](#) and [Section 6.6.2](#). We will fit an ARMAX model to the detrended mortality series `cmort`. The detrending part of the example constitutes the regression with autocorrelated errors.

Here, we let M_t denote the weekly cardiovascular mortality series, T_t as the corresponding temperature series `temp`, and P_t as the corresponding particulate series. A preliminary analysis suggests the following considerations (no output is shown):

- An AR(2) model fits well to detrended M_t :

```
fit1 = sarima(cmort, 2,0,0, xreg=time(cmort))
```

- The CCF between the mortality residuals, the temperature series and the particulates series, shows a strong correlation with temperature lagged one week (T_{t-1}), concurrent particulate level (P_t) and the particulate level about one month prior (P_{t-4}).


```
acf(cbind(dmort <- resid(fit1$fit), temp, part))
```

```
lag2.plot(temp, dmort, 8)
```

```
lag2.plot(part, dmort, 8)
```

From these results, we decided to fit the ARMAX model

$$\tilde{M}_t = \phi_1 \tilde{M}_{t-1} + \phi_2 \tilde{M}_{t-2} + \beta_1 T_{t-1} + \beta_2 P_t + \beta_3 P_{t-4} + v_t \quad (6.112)$$

to the detrended mortality series, $\tilde{M}_t = M_t - (\alpha + \beta_4 t)$, where $v_t \sim \text{iid } N(0, \sigma_v^2)$. To write the model in state-space form using [Property 6.6](#), let

$$x_{t+1} = \Phi x_t + \Upsilon u_{t+1} + \Theta v_t \quad t = 0, 1, \dots, n$$

$$y_t = \alpha + Ax_t + \Gamma u_t + v_t \quad t = 1, \dots, n$$

with

$$\Phi = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \quad \Upsilon = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \Theta = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

$A = [1 \ 0]$, $\Gamma = [0 \ 0 \ 0 \ \beta_4 \ \alpha]$, $u_t = (T_{t-1}, P_t, P_{t-4}, t, 1)'$, $y_t = M_t$. Note that the state process is bivariate and the observation process is univariate.

Some additional data analysis notes are: (1) Time is centered as $t - \bar{t}$. In this case, α should be close to the average value of M_t . (2) P_t and P_{t-4} are highly correlated, so orthogonalizing these two inputs would be advantageous (although we did not do it here), perhaps by partialling out P_{t-4} from P_t using simple linear regression. (3) T_t and T_t^2 , as in [Chapter 2](#), are not needed in the model when T_{t-1} is included. (4) Initial values of the parameters are taken from a preliminary investigation that we discuss now.

A quick and dirty method for fitting the model is to first detrend `cmort` and then fit (6.112) using `lm` on the detrended series. Rather than use `lm` in the second phase, we use `sarima` because it also provides a thorough analysis of the residuals. The code for this run is quite simple; the residual analysis (not displayed) supports the model.

```
trend = time(cmort) - mean(time(cmort)) # center time
dcmort = resid(fit2 <- lm(cmort~trend, na.action=NULL)); fit2
  (Intercept) trend
    88.699   -1.625
u = ts.intersect(dM=dcmort, dM1=lag(dcmort,-1), dM2=lag(dcmort,-2),
                 T1=lag(temp,-1), P=part, P4=lag(part,-4))
# lm(dM~, data=u, na.action=NULL) # and then analyze residuals ... or
sarima(u[,1], 0,0,0, xreg=u[,2:6]) # get residual analysis as a byproduct
Coefficients:
      intercept      dM1      dM2       T1        P       P4
      5.9884  0.3164  0.2989 -0.1826  0.1107  0.0495
      s.e.     2.6401  0.0370  0.0395  0.0309  0.0177  0.0195
      sigma^2 estimated as 25.42
```

We can now use Newton–Raphson and the Kalman filter to fit all the parameters simultaneously because the quick method has given us reasonable starting values. The results are close to the quick and dirty method:

	estimate	SE	
phi1	0.315	0.037	# $\hat{\phi}_1$
phi2	0.318	0.041	# $\hat{\phi}_2$
sigv	5.061	0.161	# $\hat{\sigma}_v$
T1	-0.119	0.031	# $\hat{\beta}_1$

P	0.119	0.018	# $\hat{\beta}_2$
P4	0.067	0.019	# $\hat{\beta}_3$
trend	-1.340	0.220	# $\hat{\beta}_4$
constant	88.752	7.015	# $\hat{\alpha}$

The R code for the complete analysis is as follows:

```

trend = time(cmort) - mean(time(cmort)) # center time
const = time(cmort)/time(cmort) # appropriate time series of ls
ded = ts.intersect(M=cmort, T1=lag(temp, -1), P=part, P4=lag(part, -4),
                    trend, const)
y = ded[,1]
input = ded[,2:6]
num = length(y)
A = array(c(1,0), dim = c(1,2,num))
# Function to Calculate Likelihood
Linn=function(para){
  phi1=para[1]; phi2=para[2]; cR=para[3]; b1=para[4]
  b2=para[5]; b3=para[6]; b4=para[7]; alf=para[8]
  mu0 = matrix(c(0,0), 2, 1)
  Sigma0 = diag(100, 2)
  Phi = matrix(c(phi1, phi2, 1, 0), 2)
  Theta = matrix(c(phi1, phi2), 2)
  Ups = matrix(c(b1, 0, b2, 0, b3, 0, 0, 0, 0, 0, 2, 5))
  Gam = matrix(c(0, 0, 0, b4, alf), 1, 5); cQ = cR; S = cR^2
  kf = Kfilter2(num, y, A, mu0, Sigma0, Phi, Ups, Gam, Theta, cQ, cR, S,
                input)
  return(kf$like) }
# Estimation
init.par = c(phi1=.3, phi2=.3, cR=5, b1=-.2, b2=.1, b3=.05, b4=-.16,
            alf=mean(cmort)) # initial parameters
L = c( 0, 0, 1, -1, 0, 0, -2, 70) # lower bound on parameters
U = c(.5, .5, 10, 0, .5, .5, 0, 90) # upper bound - used in optim
est = optim(init.par, Linn, NULL, method='L-BFGS-B', lower=L, upper=U,
            hessian=TRUE, control=list(trace=1, REPORT=1, factr=10^8))
SE = sqrt(diag(solve(est$hessian)))
round(cbind(estimate=est$par, SE), 3) # results

```

The residual analysis involves running the Kalman filter with the final estimated values and then investigating the resulting innovations. We do not display the results, but the analysis supports the model.

```

# Residual Analysis (not shown)
phi1 = est$par[1]; phi2 = est$par[2]
cR = est$par[3]; b1 = est$par[4]
b2 = est$par[5]; b3 = est$par[6]
b4 = est$par[7]; alf = est$par[8]
mu0 = matrix(c(0,0), 2, 1); Sigma0 = diag(100, 2)
Phi = matrix(c(phi1, phi2, 1, 0), 2)
Theta = matrix(c(phi1, phi2), 2)
Ups = matrix(c(b1, 0, b2, 0, b3, 0, 0, 0, 0, 0, 2, 5))
Gam = matrix(c(0, 0, 0, b4, alf), 1, 5)
cQ = cR
S = cR^2
kf = Kfilter2(num, y, A, mu0, Sigma0, Phi, Ups, Gam, Theta, cQ, cR, S,
              input)
res = ts(as.vector(kf$innov), start=start(cmort), freq=frequency(cmort))
sarima(res, 0,0,0, no.constant=TRUE) # gives a full residual analysis

```

Finally, a similar and simpler analysis can be fit using a complete ARMAX model. In this case the model would be

$$M_t = \alpha + \phi_1 M_{t-1} + \phi_2 M_{t-2} + \beta_1 T_{t-1} + \beta_2 P_t + \beta_3 P_{t-4} + \beta_4 t + v_t \quad (6.113)$$

where $v_t \sim \text{iid } N(0, \sigma_v^2)$. This model is different from (6.112) in that the mortality process is not detrended, but trend appears as an exogenous variable. In this case, we may use `sarima` to easily perform the regression and get the residual analysis as a byproduct.

```
trend = time(cmort) - mean(time(cmort))
u      = ts.intersect(M=cmort, M1=lag(cmort,-1), M2=lag(cmort,-2),
                      T1=lag(temp, -1), P=part, P4=lag(part, -4), trend)
sarima(u[,1], 0, 0, 0, xreg=u[,2:7]) # could use lm, but it's more work
Coefficients:
intercept      M1      M2      T1      P      P4      trend
40.3838  0.315  0.2971 -0.1845  0.1113  0.0513 -0.5214
s.e.       4.5982  0.037  0.0394  0.0309  0.0177  0.0195  0.0956
sigma^2 estimated as 25.32
```

We note that the residuals look fine, and the model fit is similar to the fit of (6.112).

6.7 Bootstrapping State Space Models

Although in [Section 6.3](#) we discussed the fact that under general conditions (which we assume to hold in this section) the MLEs of the parameters of a DLM are consistent and asymptotically normal, time series data are often of short or moderate length. Several researchers have found evidence that samples must be fairly large before asymptotic results are applicable (Dent and Min, 1978; Ansley and Newbold, 1980). Moreover, as we discussed in [Example 3.36](#), problems occur if the parameters are near the boundary of the parameter space. In this section, we discuss an algorithm for bootstrapping state space models; this algorithm and its justification, including the non-Gaussian case, along with numerous examples, can be found in Stoffer and Wall (1991) and in Stoffer and Wall (2004). In view of [Section 6.6](#), anything we do or say here about DLMs applies equally to ARMAX models.

Using the DLM given by (6.95)–(6.97) and [Property 6.5](#), we write the *innovations form of the filter* as

$$\epsilon_t = y_t - A_t x_t^{t-1} - \Gamma u_t, \quad (6.114)$$

$$\Sigma_t = A_t P_t^{t-1} A_t' + R, \quad (6.115)$$

$$K_t = [\Phi P_t^{t-1} A_t' + \Theta S] \Sigma_t^{-1}, \quad (6.116)$$

$$x_{t+1}^t = \Phi x_t^{t-1} + \Upsilon u_{t+1} + K_t \epsilon_t, \quad (6.117)$$

$$P_{t+1}^t = \Phi P_t^{t-1} \Phi' + \Theta Q \Theta' - K_t \Sigma_t K_t'. \quad (6.118)$$

This form of the filter is just a rearrangement of the filter given in [Property 6.5](#).

In addition, we can rewrite the model to obtain its innovations form,

$$x_{t+1}^t = \Phi x_t^{t-1} + \Upsilon u_{t+1} + K_t \epsilon_t, \quad (6.119)$$

$$y_t = A_t x_t^{t-1} + \Gamma u_t + \epsilon_t. \quad (6.120)$$

This form of the model is a rewriting of (6.114) and (6.117), and it accommodates the bootstrapping algorithm.

As discussed in [Example 6.5](#), although the innovations ϵ_t are uncorrelated, initially, Σ_t can be vastly different for different time points t . Thus, in a resampling procedure, we can either ignore the first few values of ϵ_t until Σ_t stabilizes or we can work with the *standardized innovations*

$$e_t = \Sigma_t^{-1/2} \epsilon_t, \quad (6.121)$$

so we are guaranteed these innovations have, at least, the same first two moments. In (6.121), $\Sigma_t^{1/2}$ denotes the unique square root matrix of Σ_t defined by $\Sigma_t^{1/2} \Sigma_t^{1/2} = \Sigma_t$. In what follows, we base the bootstrap procedure on the standardized innovations, but we stress the fact that, even in this case, ignoring startup values might be necessary, as noted by Stoffer & Wall (1991).

The model coefficients and the correlation structure of the model are uniquely parameterized by a $k \times 1$ parameter vector Θ_0 ; that is, $\Phi = \Phi(\Theta_0)$, $\Upsilon = \Upsilon(\Theta_0)$, $Q = Q(\Theta_0)$, $A_t = A_t(\Theta_0)$, $\Gamma = \Gamma(\Theta_0)$, and $R = R(\Theta_0)$. Recall the innovations form of the Gaussian likelihood (ignoring a constant) is

$$\begin{aligned} -2 \ln L_Y(\Theta) &= \sum_{t=1}^n [\ln |\Sigma_t(\Theta)| + \epsilon_t(\Theta)' \Sigma_t(\Theta)^{-1} \epsilon_t(\Theta)] \\ &= \sum_{t=1}^n [\ln |\Sigma_t(\Theta)| + e_t(\Theta)' e_t(\Theta)]. \end{aligned} \quad (6.122)$$

We stress the fact that it is not necessary for the model to be Gaussian to consider (6.122) as the criterion function to be used for parameter estimation.

Let $\hat{\Theta}$ denote the MLE of Θ_0 , that is, $\hat{\Theta} = \operatorname{argmax}_{\Theta} L_Y(\Theta)$, obtained by the methods discussed in [Section 6.3](#). Let $\epsilon_t(\hat{\Theta})$ and $\Sigma_t(\hat{\Theta})$ be the innovation values obtained by running the filter, (6.114)–(6.118), under $\hat{\Theta}$. Once this has been done, the nonparametric^{6.2} bootstrap procedure is accomplished by the following steps.

- (i) Construct the standardized innovations

$$e_t(\hat{\Theta}) = \Sigma_t^{-1/2}(\hat{\Theta}) \epsilon_t(\hat{\Theta}).$$

- (ii) Sample, with replacement, n times from the set $\{e_1(\hat{\Theta}), \dots, e_n(\hat{\Theta})\}$ to obtain $\{e_1^*(\hat{\Theta}), \dots, e_n^*(\hat{\Theta})\}$, a bootstrap sample of standardized innovations.
- (iii) Construct a bootstrap data set $\{y_1^*, \dots, y_n^*\}$ as follows. Define the $(p+q) \times 1$ vector $\xi_t = (x_{t+1}'^t, y_t')'$. Stacking (6.119) and (6.120) results in a vector first-order equation for ξ_t given by

^{6.2} Nonparametric refers to the fact that we use the empirical distribution of the innovations rather than assuming they have a parametric form.

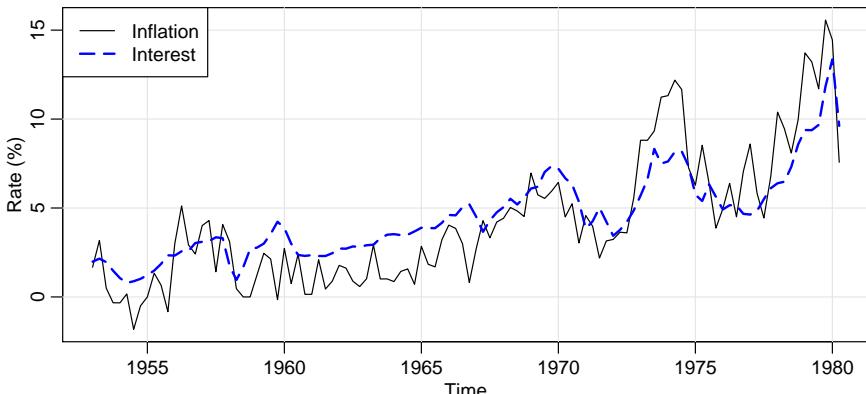


Fig. 6.9. Quarterly interest rate for Treasury bills (dashed line) and quarterly inflation rate (solid line) in the Consumer Price Index.

$$\xi_t = F_t \xi_{t-1} + Gu_t + H_t e_t, \quad (6.123)$$

where

$$F_t = \begin{bmatrix} \Phi & 0 \\ A_t & 0 \end{bmatrix}, \quad G = \begin{bmatrix} \Gamma \\ \Gamma \end{bmatrix}, \quad H_t = \begin{bmatrix} K_t \Sigma_t^{1/2} \\ \Sigma_t^{1/2} \end{bmatrix}.$$

Thus, to construct the bootstrap data set, solve (6.123) using $e_t^*(\hat{\Theta})$ in place of e_t . The exogenous variables u_t and the initial conditions of the Kalman filter remain fixed at their given values, and the parameter vector is held fixed at $\hat{\Theta}$.

- (iv) Using the bootstrap data set $y_{1:n}^*$, construct a likelihood, $L_{Y^*}(\Theta)$, and obtain the MLE of Θ , say, $\hat{\Theta}^*$.
- (v) Repeat steps 2 through 4, a large number, B , of times, obtaining a bootstrapped set of parameter estimates $\{\hat{\Theta}_b^*; b = 1, \dots, B\}$. The finite sample distribution of $\hat{\Theta} - \Theta_0$ may be approximated by the distribution of $\hat{\Theta}_b^* - \hat{\Theta}$, $b = 1, \dots, B$.

In the next example, we discuss the case of a linear regression model, but where the regression coefficients are stochastic and allowed to vary with time. The state space model provides a convenient setting for the analysis of such models.

Example 6.13 Stochastic Regression

Figure 6.9 shows the quarterly inflation rate (solid line), y_t , in the Consumer Price Index and the quarterly interest rate recorded for Treasury bills (dashed line), z_t , from the first quarter of 1953 through the second quarter of 1980, $n = 110$ observations. These data are taken from Newbold and Bos (1985).

In this example, we consider one analysis that was discussed in Newbold and Bos (1985, pp. 61–73), that focused on the first 50 observations and where quarterly inflation was modeled as being stochastically related to quarterly interest rate,

$$y_t = \alpha + \beta_t z_t + v_t,$$

Table 6.2. Comparison of Standard Errors

Parameter	MLE	Asymptotic Standard Error	Bootstrap Standard Error
ϕ	.865	.223	.463
α	-.686	.487	.557
b	.788	.226	.821
σ_w	.115	.107	.216
σ_v	1.135	.147	.340

where α is a fixed constant, β_t is a stochastic regression coefficient, and v_t is white noise with variance σ_v^2 . The stochastic regression term, which comprises the state variable, is specified by a first-order autoregression,

$$(\beta_t - b) = \phi(\beta_{t-1} - b) + w_t,$$

where b is a constant, and w_t is white noise with variance σ_w^2 . The noise processes, v_t and w_t , are assumed to be uncorrelated.

Using the notation of the state-space model (6.95) and (6.96), we have in the state equation, $x_t = \beta_t$, $\Phi = \phi$, $u_t \equiv 1$, $\Upsilon = (1 - \phi)b$, $Q = \sigma_w^2$, and in the observation equation, $A_t = z_t$, $\Gamma = \alpha$, $R = \sigma_v^2$, and $S = 0$. The parameter vector is $\Theta = (\phi, \alpha, b, \sigma_w, \sigma_v)'$. The results of the Newton–Raphson estimation procedure are listed in Table 6.2. Also shown in the Table 6.2 are the corresponding standard errors obtained from $B = 500$ runs of the bootstrap. These standard errors are simply the standard deviations of the bootstrapped estimates, that is, the square root of $\sum_{b=1}^B (\hat{\Theta}_{ib}^* - \hat{\Theta}_i)^2 / (B - 1)$, where $\hat{\Theta}_i$ represents the MLE of the i th parameter, Θ_i , for $i = 1, \dots, 5$,

The asymptotic standard errors listed in Table 6.2 are typically much smaller than those obtained from the bootstrap. For most of the cases, the bootstrapped standard errors are at least 50% larger than the corresponding asymptotic value. Also, asymptotic theory prescribes the use of normal theory when dealing with the parameter estimates. The bootstrap, however, allows us to investigate the small sample distribution of the estimators and, hence, provides more insight into the data analysis.

For example, Figure 6.10 shows the bootstrap distribution of the estimator of ϕ in the upper left-hand corner. This distribution is highly skewed with values concentrated around .8, but with a long tail to the left. Some quantiles are $-.09$ (5%), $.11$ (10%), $.34$ (25%), $.73$ (50%), $.86$ (75%), $.96$ (90%), $.98$ (95%), and they can be used to obtain confidence intervals. For example, a 90% confidence interval for ϕ would be approximated by $(-.09, .96)$. This interval is ridiculously wide and includes 0 as a plausible value of ϕ ; we will interpret this after we discuss the results of the estimation of σ_w .

Figure 6.10 shows the bootstrap distribution of $\hat{\sigma}_w$ in the lower right-hand corner. The distribution is concentrated at two locations, one at approximately $\hat{\sigma}_w = .25$ (which is the median of the distribution of values away from 0) and

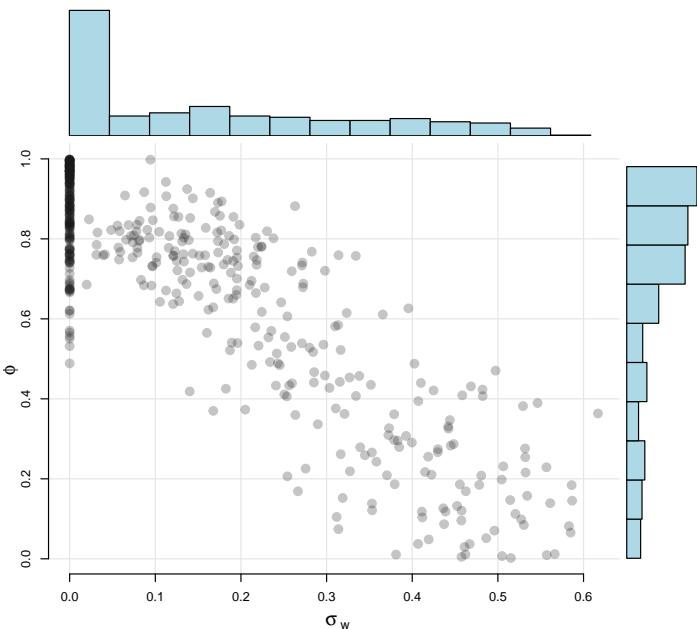


Fig. 6.10. Joint and marginal bootstrap distributions, $B = 500$, of $\hat{\phi}$ and $\hat{\sigma}_w$. Only the values corresponding to $\hat{\phi}^* \geq 0$ are shown.

the other at $\hat{\sigma}_w = 0$. The cases in which $\hat{\sigma}_w \approx 0$ correspond to deterministic state dynamics. When $\sigma_w = 0$ and $|\phi| < 1$, then $\beta_t \approx b$ for large t , so the approximately 25% of the cases in which $\hat{\sigma}_w \approx 0$ suggest a fixed state, or constant coefficient model. The cases in which $\hat{\sigma}_w$ is away from zero would suggest a truly stochastic regression parameter. To investigate this matter further, the off-diagonals of Figure 6.10 show the joint bootstrapped estimates, $(\hat{\phi}, \hat{\sigma}_w)$, for positive values of $\hat{\phi}^*$. The joint distribution suggests $\hat{\sigma}_w > 0$ corresponds to $\hat{\phi} \approx 0$. When $\phi = 0$, the state dynamics are given by $\beta_t = b + w_t$. If, in addition, σ_w is small relative to b , the system is nearly deterministic; that is, $\beta_t \approx b$. Considering these results, the bootstrap analysis leads us to conclude the dynamics of the data are best described in terms of a fixed regression effect.

The following R code was used for this example. We note that the first few lines of the code set the relative tolerance for determining convergence of the numerical optimization and the number of bootstrap replications. *Using the current settings may result in a long run time of the algorithm* and we suggest the tolerance and the number of bootstrap replicates be decreased on slower machines or for demonstration purposes. For example, setting `tol=.001` and `nboot=200` yields reasonable results. In this example, we fixed the first three values of the data for the resampling scheme.

```
library(plyr)                      # used for displaying progress
tol   = sqrt(.Machine$double.eps)  # determines convergence of optimizer
```

```

nboot = 500                      # number of bootstrap replicates
y    = window(qinfl, c(1953,1), c(1965,2)) # inflation
z    = window(qintr, c(1953,1), c(1965,2)) # interest
num  = length(y)
A    = array(z, dim=c(1,1,num))
input = matrix(1,num,1)
# Function to Calculate Likelihood
Linn = function(para, y.data){ # pass data also
  phi = para[1]; alpha = para[2]
  b   = para[3]; Ups  = (1-phi)*b
  cQ  = para[4]; cR   = para[5]
  kf  = Kfilter2(num,y.data,A,mu0,Sigma0,phi,Ups,alpha,1,cQ,cR,0,input)
  return(kf$like) }
# Parameter Estimation
mu0 = 1; Sigma0 = .01
init.par = c(phi=.84, alpha=-.77, b=.85, cQ=.12, cR=1.1) # initial values
est = optim(init.par, Linn, NULL, y.data=y, method="BFGS", hessian=TRUE,
            control=list(trace=1, REPORT=1, reltol=tol))
SE = sqrt(diag(solve(est$hessian)))
phi = est$par[1]; alpha = est$par[2]
b   = est$par[3]; Ups  = (1-phi)*b
cQ  = est$par[4]; cR   = est$par[5]
round(cbind(estimate=est$par, SE), 3)
      estimate     SE
phi      0.865 0.223
alpha    -0.686 0.487
b        0.788 0.226
cQ       0.115 0.107
cR       1.135 0.147
# BEGIN BOOTSTRAP
# Run the filter at the estimates
kf = Kfilter2(num,y,A,mu0,Sigma0,phi,Ups,alpha,1,cQ,cR,0,input)
# Pull out necessary values from the filter and initialize
xp     = kf$xp
innov  = kf$innov
sig    = kf$sig
K      = kf$K
e      = innov/sqrt(sig)
e.star = e                         # initialize values
y.star = y
xp.star = xp
k      = 4:50                      # hold first 3 observations fixed
para.star = matrix(0, nboot, 5) # to store estimates
init.par = c(.84, -.77, .85, .12, 1.1)
pr <- progress_text()           # displays progress
pr$init(nboot)
for (i in 1:nboot){
  pr$step()
  e.star[k] = sample(e[k], replace=TRUE)
  for (j in k){ xp.star[j] = phi*xp.star[j-1] +
    Ups+K[j]*sqrt(sig[j])*e.star[j] }
  y.star[k] = z[k]*xp.star[k] + alpha + sqrt(sig[k])*e.star[k]
  est.star = optim(init.par, Linn, NULL, y.data=y.star, method="BFGS",
                  control=list(reltol=tol))
  para.star[i,] = cbind(est.star$par[1], est.star$par[2], est.star$par[3],
                        abs(est.star$par[4]), abs(est.star$par[5])) }

```

```

# Some summary statistics
rmse = rep(NA,5)           # SEs from the bootstrap
for(i in 1:5){rmse[i]=sqrt(sum((para.star[,i]-est$par[i])^2)/nboot)
  cat(i, rmse[i],"\n") }
# Plot phi and sigw
phi = para.star[,1]
sigw = abs(para.star[,4])
phi = ifelse(phi<0, NA, phi)    # any phi < 0 not plotted
library(psych)                 # load psych package for scatter.hist
scatter.hist(sigw, phi, ylab=expression(phi), xlab=expression(sigma[~w]),
             smooth=FALSE, correl=FALSE, density=FALSE, ellipse=FALSE,
             title='', pch=19, col=gray(.1,alpha=.33),
             panel.first=grid(lty=2), cex.lab=1.2)

```

6.8 Smoothing Splines and the Kalman Smoother

There is a connection between smoothing splines, e.g., Eubank (1993), Green (1993), or Wahba (1990) and state space models. The basic idea of smoothing splines (recall Example 2.14) in discrete time is we suppose that data y_t are generated by $y_t = \mu_t + \epsilon_t$ for $t = 1, \dots, n$, where μ_t is a smooth function of t , and ϵ_t is white noise. In cubic smoothing with knots at the time points t , μ_t is estimated by minimizing

$$\sum_{t=1}^n [y_t - \mu_t]^2 + \lambda \sum_{t=1}^n (\nabla^2 \mu_t)^2 \quad (6.124)$$

with respect to μ_t , where $\lambda > 0$ is a smoothing parameter. The parameter λ controls the degree of smoothness, with larger values yielding smoother estimates. For example, if $\lambda = 0$, then the minimizer is the data itself $\hat{\mu}_t = y_t$; consequently, the estimate will not be smooth. If $\lambda = \infty$, then the only way to minimize (6.124) is to choose the second term to be zero, i.e., $\nabla^2 \mu_t = 0$, in which case it is of the form $\mu_t = \alpha + \beta t$, and we are in the setting of linear regression.^{6.3} Hence, the choice of $\lambda > 0$ is seen as a trade-off between fitting a line that goes through all the data points and linear regression.

Now, consider the model given by

$$\nabla^2 \mu_t = w_t \quad \text{and} \quad y_t = \mu_t + v_t, \quad (6.125)$$

where w_t and v_t are independent white noise processes with $\text{var}(w_t) = \sigma_w^2$ and $\text{var}(v_t) = \sigma_v^2$. Rewrite (6.125) as

$$\begin{pmatrix} \mu_t \\ \mu_{t-1} \end{pmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \mu_{t-1} \\ \mu_{t-2} \end{pmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w_t \quad \text{and} \quad y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{pmatrix} \mu_t \\ \mu_{t-1} \end{pmatrix} + v_t, \quad (6.126)$$

so that the state vector is $x_t = (\mu_t, \mu_{t-1})'$. It is clear then that (6.125) specifies a state space model.

^{6.3} That the unique general solution to $\nabla^2 \mu_t = 0$ is of the form $\mu_t = \alpha + \beta t$ follows from difference equation theory; e.g., see Mickens (1990).

Note that the model is similar to the local level model discussed in [Example 6.5](#). In particular, the state process could be written as $\mu_t = \mu_{t-1} + \eta_t$, where $\eta_t = \eta_{t-1} + w_t$. An example of such a trajectory can be seen in [Figure 6.11](#); note that the generated data in [Figure 6.11](#) look like the global temperature data in [Figure 1.2](#).

Next, we examine the problem of estimating the states, x_t , when the model parameters, $\theta = \{\sigma_w^2, \sigma_v^2\}$, are specified. For ease, we assume x_0 is fixed. Then using the notation surrounding equations [\(6.61\)](#)–[\(6.62\)](#), the goal is to find the MLE of $x_{1:n} = \{x_1, \dots, x_n\}$ given $y_{1:n} = \{y_1, \dots, y_n\}$; i.e., to maximize $\log p_\theta(x_{1:n} | y_{1:n})$ with respect to the states. Because of the Gaussianity, the maximum (or mode) of the distribution is when the states are estimated by x_t^n , the conditional means. These values are, of course, the smoothers obtained via [Property 6.2](#).

But $\log p_\theta(x_{1:n} | y_{1:n}) = \log p_\theta(x_{1:n}, y_{1:n}) - \log p_\theta(y_{1:n})$, so maximizing the complete data likelihood, $\log p_\theta(x_{1:n}, y_{1:n})$ with respect to $x_{1:n}$, is an equivalent problem. Writing [\(6.62\)](#) in the notation of [\(6.125\)](#), we have,

$$-2 \log p_\theta(x_{1:n}, y_{1:n}) \propto \sigma_w^{-2} \sum_{t=1}^n (\nabla^2 \mu_t)^2 + \sigma_v^{-2} \sum_{t=1}^n (y_t - \mu_t)^2, \quad (6.127)$$

where we have kept only the terms involving the states, μ_t . If we set $\lambda = \sigma_v^2 / \sigma_w^2$, we can write

$$-2 \log p_\theta(x_{1:n}, y_{1:n}) \propto \lambda \sum_{t=1}^n (\nabla^2 \mu_t)^2 + \sum_{t=1}^n (y_t - \mu_t)^2, \quad (6.128)$$

so that maximizing $\log p_\theta(x_{1:n}, y_{1:n})$ with respect to the states is equivalent to minimizing [\(6.128\)](#), which is the original problem stated in [\(6.124\)](#).

In the general state space setting, we would estimate σ_w^2 and σ_v^2 via maximum likelihood as described in [Section 6.3](#), and then obtain the smoothed state values by running [Property 6.2](#) with the estimated variances, say $\hat{\sigma}_w^2$ and $\hat{\sigma}_v^2$. In this case, the estimated value of the smoothing parameter would be given by $\hat{\lambda} = \hat{\sigma}_v^2 / \hat{\sigma}_w^2$.

Example 6.14 Smoothing Splines

In this example, we generated the signal, or state process, μ_t and observations y_t from the model [\(6.125\)](#) with $n = 50$, $\sigma_w = .1$ and $\sigma_v = 1$. The state is displayed in [Figure 6.11](#) as a thick solid line, and the observations are displayed as points. We then estimated σ_w and σ_v using Newton-Raphson techniques and obtained $\hat{\sigma}_w = .08$ and $\hat{\sigma}_v = .94$. We then used [Property 6.2](#) to generate the estimated smoothers, say, $\hat{\mu}_t^n$, and those values are displayed in [Figure 6.11](#) as a thick dashed line along with a corresponding 95% (pointwise) confidence band as thin dashed lines. Finally, we used the R function `smooth.spline` to fit a smoothing spline to the data based on the method of generalized cross-validation (gcv). The fitted spline is displayed in [Figure 6.11](#) as a thin solid line, which is close to $\hat{\mu}_t^n$.

The R code to reproduce [Figure 6.11](#) is given below.

```
set.seed(123)
num = 50
w = rnorm(num, 0, .1)
x = cumsum(cumsum(w))
```

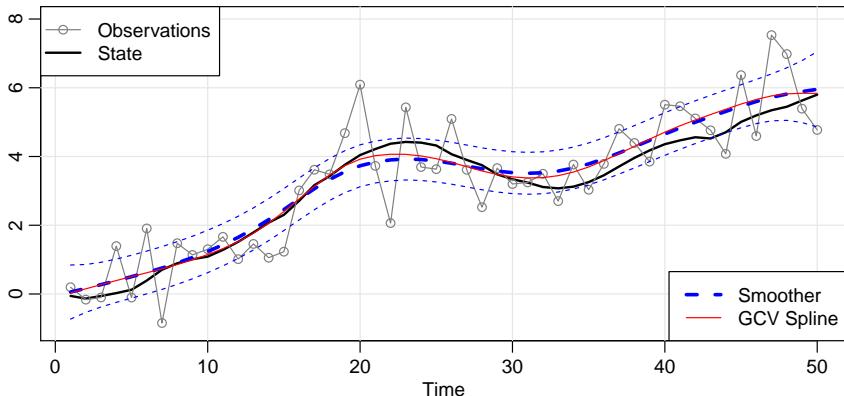


Fig. 6.11. Display for Example 6.14: Simulated state process, μ_t and observations y_t from the model (6.125) with $n = 50$, $\sigma_w = .1$ and $\sigma_v = 1$. Estimated smoother (dashed lines): $\hat{\mu}_{t|n}$ and corresponding 95% confidence band. GCV smoothing spline (thin solid line).

```

y = x + rnorm(num,0,1)
plot.ts(x, ylab="", lwd=2, ylim=c(-1,8))
lines(y, type='o', col=8)
## State Space ##
Phi = matrix(c(2,1,-1,0),2); A = matrix(c(1,0),1)
mu0 = matrix(0,2); Sigma0 = diag(1,2)
Linn = function(para){
  sigw = para[1]; sigv = para[2]
  cQ = diag(c(sigw,0))
  kf = Kfilter0(num, y, A, mu0, Sigma0, Phi, cQ, sigv)
  return(kf$like) }
## Estimation ##
init.par = c(.1, 1)
est = optim(init.par, Linn, NULL, method="BFGS", hessian=TRUE,
            control=list(trace=1,REPORT=1))
SE = sqrt(diag(solve(est$hessian)))
# Summary of estimation
estimate = est$par; u = cbind(estimate, SE)
rownames(u) = c("sigw","sigv"); u
# Smooth
sigw = est$par[1]
cQ = diag(c(sigw,0))
sigv = est$par[2]
ks = Ksmooth0(num, y, A, mu0, Sigma0, Phi, cQ, sigv)
xsmoo = ts(ks$xs[,1,]); psmoo = ts(ks$Ps[,1,])
upp = xsmoo+2*sqrt(psmoo); low = xsmoo-2*sqrt(psmoo)
lines(xsmoo, col=4, lty=2, lwd=3)
lines(upp, col=4, lty=2); lines(low, col=4, lty=2)
lines(smooth.spline(y), lty=1, col=2)
legend("topleft", c("Observations","State"), pch=c(1,-1), lty=1, lwd=c(1,2),
       col=c(8,1))
legend("bottomright", c("Smoothen", "GCV Spline"), lty=c(2,1), lwd=c(3,1),
       col=c(4,2))

```

6.9 Hidden Markov Models and Switching Autoregression

In the introduction to this chapter, we mentioned that the state space model is characterized by two principles. First, there is a hidden state process, $\{x_t; t = 0, 1, \dots\}$, that is assumed to be Markovian. Second, the observations, $\{y_t; t = 1, 2, \dots\}$, are independent given the states. The principles were displayed in Figure 6.1 and written in terms of densities in (6.28) and (6.29).

We have been focusing primarily on linear Gaussian state space models, but there is an entire area that has developed around the case where the states x_t are a discrete-valued Markov chain, and that will be the focus in this section. The basic idea is that the value of the state at time t specifies the distribution of the observation at time t . These models were developed in Goldfeld and Quandt (1973) and Lindgren (1978). Changes can also be modeled in the classical regression setting by allowing the value of the state to determine the design matrix, as in Quandt (1972). An early application to speech recognition was considered by Juang and Rabiner (1985). An application of the idea of switching to the tracking of multiple targets was considered in Bar-Shalom (1978), who obtained approximations to Kalman filtering in terms of weighted averages of the innovations. As another example, some authors (for example, Hamilton, 1989, or McCulloch and Tsay, 1993) have explored the possibility that the dynamics of a country's economy might be different during expansion than during contraction.

In the Markov chain approach, we declare the dynamics of the system at time t are generated by one of m possible regimes evolving according to a Markov chain over time. The case in which the particular regime is unknown to the observer comes under the heading of *hidden Markov models* (HMM), and the techniques related to analyzing these models are summarized in Rabiner and Juang (1986). Although the model satisfies the conditions for being a state space model, HMMs were developed in parallel. If the state process is discrete-valued, one typically uses the term “hidden Markov model” and if the state process is continuous-valued, one uses the term “state space model” or one of its variants. Texts that cover the theory and methods in whole or in part are Cappé, Moulines, & Rydén (2009) and Douc, Moulines, & Stoffer (2014). A recent introductory text that uses R is Zucchini & MacDonald (2009).

Here, we assume the states, x_t , are a Markov chain taking values in a finite state space $\{1, \dots, m\}$, with stationary distribution

$$\pi_j = \Pr(x_t = j), \quad (6.129)$$

and stationary transition probabilities

$$\pi_{ij} = \Pr(x_{t+1} = j \mid x_t = i), \quad (6.130)$$

for $t = 0, 1, 2, \dots$, and $i, j = 1, \dots, m$. Since the second component of the model is that the observations are conditionally independent, we need to specify the distributions, and we denote them by

$$p_j(y_t) = p(y_t \mid x_t = j). \quad (6.131)$$

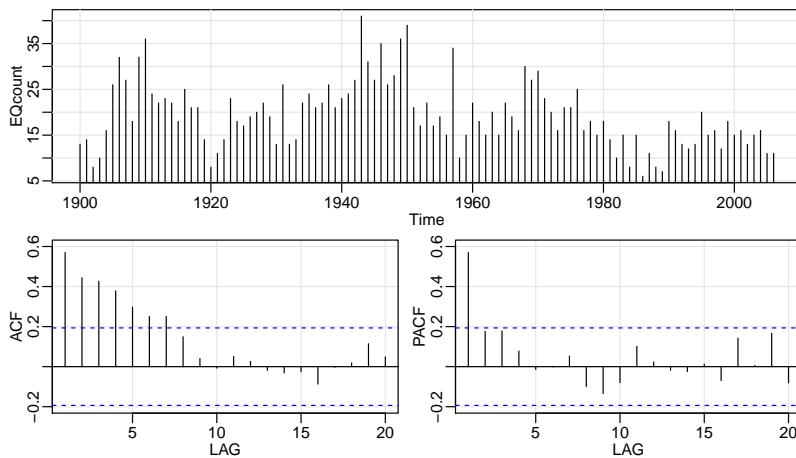


Fig. 6.12. Top: Series of annual counts of major earthquakes (magnitude 7 and above) in the world between 1900–2006. Bottom: Sample ACF and PACF of the counts.

Example 6.15 Poisson HMM – Number of Major Earthquakes

Consider the time series of annual counts of major earthquakes displayed in Figure 6.12 that were discussed in Zucchini & MacDonald (2009). A natural model for unbounded count data is a Poisson distribution, in which case the mean and variance are equal. However, the sample mean and variance of the data are $\bar{x} = 19.4$ and $s^2 = 51.6$, so this model is clearly inappropriate. It would be possible to take into account the overdispersion by using other distributions for counts such as the negative binomial distribution or a mixture of Poisson distributions. This approach, however, ignores the sample ACF and PACF displayed Figure 6.12, which indicate the observations are serially correlated, and further suggest an AR(1)-type correlation structure.

A simple and convenient way to capture both the marginal distribution and the serial dependence is to consider a Poisson-HMM model. Let y_t denote the number of major earthquakes in year t , and consider the state, or latent variable, x_t to be a stationary two-state Markov chain taking values in $\{1, 2\}$. Using the notation in (6.129) and (6.130), we have $\pi_{12} = 1 - \pi_{11}$ and $\pi_{21} = 1 - \pi_{22}$. The stationary distribution of this Markov chain is given by^{6.4}

$$\pi_1 = \frac{\pi_{21}}{\pi_{12} + \pi_{21}}, \quad \text{and} \quad \pi_2 = \frac{\pi_{12}}{\pi_{12} + \pi_{21}}.$$

For $j \in \{1, 2\}$, denote $\lambda_j > 0$ as the parameter of a Poisson distribution,

$$p_j(y) = \frac{\lambda_j^y e^{-\lambda_j}}{y!}, \quad y = 0, 1, \dots.$$

^{6.4} The stationary distribution must satisfy $\pi_j = \sum_i \pi_i \pi_{ij}$.

Since the states are stationary, the marginal distribution of y_t is stationary and a mixture of Poissons,

$$p_{\Theta}(y_t) = \pi_1 p_1(y_t) + \pi_2 p_2(y_t)$$

with $\Theta = \{\lambda_1, \lambda_2\}$. The mean of the stationary distribution is

$$E(y_t) = \pi_1 \lambda_1 + \pi_2 \lambda_2 \quad (6.132)$$

and the variance^{6.5} is

$$\text{var}(y_t) = E(y_t) + \pi_1 \pi_2 (\lambda_2 - \lambda_1)^2 \geq E(y_t), \quad (6.133)$$

implying that the two-state Poisson HMM is overdispersed. Similar calculations (see [Problem 6.21](#)) show that the autocovariance function of y_t is given by

$$\gamma_y(h) = \sum_{i=1}^2 \sum_{j=1}^2 \pi_i (\pi_{ij}^h - \pi_j) \lambda_i \lambda_j = \pi_1 \pi_2 (\lambda_2 - \lambda_1)^2 (1 - \pi_{12} - \pi_{21})^h. \quad (6.134)$$

Thus, a two-state Poisson-HMM has an exponentially decaying autocorrelation function, and this is consistent with the sample ACF seen in [Figure 6.12](#). It is worthwhile to note that if we increase the number of states, more complex dependence structures may be obtained.

As in the linear Gaussian case, we need filters and smoothers of the state in their own right, and additionally for estimation and prediction. We then write

$$\pi_j(t \mid s) = \Pr(x_t = j \mid y_{1:s}). \quad (6.135)$$

Straight forward calculations (see [Problem 6.22](#)) give the filter equations as:

Property 6.7 HMM Filter

For $t = 1, \dots, n$,

$$\pi_j(t \mid t-1) = \sum_{i=1}^m \pi_i(t-1 \mid t-1) \pi_{ij}, \quad (6.136)$$

$$\pi_j(t \mid t) = \frac{\pi_j(t) p_j(y_t)}{\sum_{i=1}^m \pi_i(t) p_i(y_t)}, \quad (6.137)$$

with initial condition $\pi_j(1 \mid 0) = \pi_j$.

Let Θ denote the parameters of interest. Given data $y_{1:n}$, the likelihood is given by

$$L_Y(\Theta) = \prod_{t=1}^n p_{\Theta}(y_t \mid y_{1:t-1}).$$

But, by the conditional independence,

^{6.5} Recall $\text{var}(U) = E[\text{var}(U \mid V)] + \text{var}[E(U \mid V)]$.

$$\begin{aligned} p_{\Theta}(y_t \mid y_{1:t-1}) &= \sum_{j=1}^m \Pr(x_t = j \mid y_{1:t-1}) p_{\Theta}(y_j \mid x_t = j, y_{1:t-1}) \\ &= \sum_{j=1}^m \pi_j(t \mid t-1) p_j(y_t). \end{aligned}$$

Consequently,

$$\ln L_Y(\Theta) = \sum_{t=1}^n \ln \left(\sum_{j=1}^m \pi_j(t \mid t-1) p_j(y_t) \right). \quad (6.138)$$

Maximum likelihood can then proceed as in the linear Gaussian case discussed in [Section 6.3](#).

In addition, the Baum-Welch (or EM) algorithm discussed in [Section 6.3](#) applies here as well. First, the general complete data likelihood still has the form of [\(6.61\)](#), that is,

$$\ln p_{\Theta}(x_{0:n}, y_{1:n}) = \ln p_{\Theta}(x_0) + \sum_{t=1}^n \ln p_{\Theta}(x_t \mid x_{t-1}) + \sum_{t=1}^n \ln p_{\Theta}(y_t \mid x_t).$$

It is more useful to define $I_j(t) = 1$ if $x_t = j$ and 0 otherwise, and $I_{ij}(t) = 1$ if $(x_{t-1}, x_t) = (i, j)$ and 0 otherwise, for $i, j = 1, \dots, m$. Recall $\Pr[I_j(t) = 1] = \pi_j$ and $\Pr[I_{ij}(t) = 1] = \pi_{ij} \pi_i$. Then the complete data likelihood can be written as (we drop Θ from some of the notation for convenience)

$$\begin{aligned} \ln p_{\Theta}(x_{0:n}, y_{1:n}) &= \sum_{j=1}^m I_j(0) \ln \pi_j + \sum_{t=1}^n \sum_{i=1}^m \sum_{j=1}^m I_{ij}(t) \ln \pi_{ij}(t) \\ &\quad + \sum_{t=1}^n \sum_{j=1}^m I_j(t) \ln p_j(y_t), \end{aligned} \quad (6.139)$$

and, as before, we need to maximize $Q(\Theta \mid \Theta') = E[\ln p_{\Theta}(x_{0:n}, y_{1:n}) \mid y_{1:n}, \Theta']$. In this case, it should be clear that in addition to the filter, [\(6.137\)](#), we will need

$$\pi_j(t \mid n) = E(I_j(t) \mid y_{1:n}) = \Pr(x_t = j \mid y_{1:n}) \quad (6.140)$$

for the first and third terms, and

$$\pi_{ij}(t \mid n) = E(I_{ij}(t) \mid y_{1:n}) = \Pr(x_t = i, x_{t+1} = j \mid y_{1:n}). \quad (6.141)$$

for the second term. In the evaluation of the second term, as will be seen, we must also evaluate

$$\varphi_j(t) = p(y_{t+1:n} \mid x_t = j). \quad (6.142)$$

Property 6.8 HMM Smoother

For $t = n - 1, \dots, 0$,

$$\pi_j(t | n) = \frac{\pi_j(t | t)\varphi_j(t)}{\sum_{j=1}^m \pi_j(t | t)\varphi_j(t)}, \quad (6.143)$$

$$\pi_{ij}(t | n) = \pi_i(t | n)\pi_{ij}\text{p}_j(y_{t+1})\varphi_j(t+1)/\varphi_i(t), \quad (6.144)$$

$$\varphi_i(t) = \sum_{j=1}^m \pi_{ij}\text{p}_j(y_{t+1})\varphi_j(t+1), \quad (6.145)$$

where $\varphi_j(n) = 1$ for $j = 1, \dots, m$.

Proof: We leave the proof of (6.143) to the reader; see [Problem 6.22](#). To verify (6.145), note that

$$\begin{aligned} \varphi_i(t) &= \sum_{j=1}^m \text{p}(y_{t+1:n}, x_{t+1} = j | x_t = i) \\ &= \sum_{j=1}^m \Pr(x_{t+1} = j | x_t = i) \text{p}(y_{t+1} | x_{t+1} = j) \text{p}(y_{t+2:n} | x_{t+1} = j) \\ &= \sum_{j=1}^m \pi_{ij} \text{p}_j(y_{t+1})\varphi_j(t+1). \end{aligned}$$

To verify (6.144), we have

$$\begin{aligned} \pi_{ij}(t | n) &\propto \Pr(x_t = i, x_{t+1} = j, y_{t+1}, y_{t+2:n} | y_{1:t}) \\ &= \Pr(x_t = i | y_{1:t}) \Pr(x_{t+1} = j | x_t = i) \\ &\quad \times \text{p}(y_{t+1} | x_{t+1} = j) \text{p}(y_{t+2:n} | x_{t+1} = j) \\ &= \pi_i(t | t) \pi_{ij} \text{p}_j(y_{t+1}) \varphi_j(t+1). \end{aligned}$$

Finally, to find the constant of proportionality, say C_t , if we sum over j on both sides we get, $\sum_{j=1}^m \pi_{ij}(t | n) = \pi_i(t | n)$ and $\sum_{j=1}^m \pi_{ij} \text{p}_j(y_{t+1}) \varphi_j(t+1) = \varphi_i(t)$. This means that $\pi_i(t | n) = C_t \pi_i(t | t) \varphi_i(t)$, and (6.144) follows. \square

For the Baum-Welch (or EM) algorithm, given the current value of the parameters, say Θ' , run the filter [Property 6.7](#) and smoother [Property 6.8](#), and then, as is evident from (6.139), update the first two estimates as

$$\hat{\pi}_j = \pi'_j(0 | n) \quad \text{and} \quad \hat{\pi}_{ij} = \frac{\sum_{t=1}^n \pi'_{ij}(t | n)}{\sum_{t=1}^n \sum_{k=1}^m \pi'_{ik}(t | n)}. \quad (6.146)$$

Of course, the prime indicates that values have been obtain under Θ' and the hat denotes the update. Although not the MLE, it has been suggested by Lindgren (1978) that a natural estimate of the stationary distribution of the chain would be

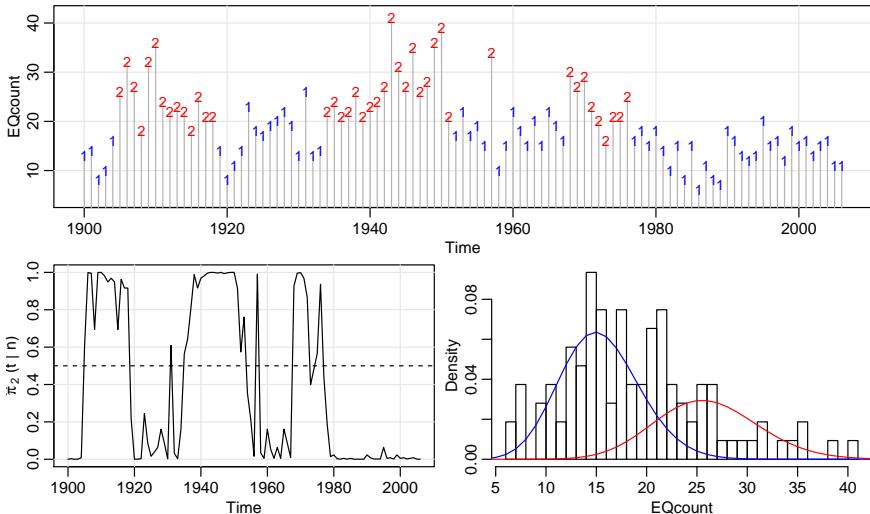


Fig. 6.13. Top: Earthquake count data and estimated states. Bottom left: Smoothing probabilities. Bottom right: Histogram of the data with the two estimated Poisson densities superimposed (solid lines).

$$\hat{\pi}_j = n^{-1} \sum_{t=1}^n \pi'_j(t|n),$$

rather than the value given in (6.146). Finally, the third term in (6.139) will require knowing the distribution of $p_j(y_t)$, and this will depend on the particular model. We will discuss the Poisson distribution in Example 6.15 and the normal distribution in Example 6.17

Example 6.16 Poisson HMM – Number of Major Earthquakes (cont)

To run the EM algorithm in this case, we still need to maximize the conditional expectation of the third term of (6.139). The conditional expectation of the third term at the current parameter value Θ' is

$$\sum_{t=1}^n \sum_{j=1}^m \pi'_j(t|t-1) \ln p_j(y_t),$$

where

$$\log p_j(y_t) \propto y_t \log \lambda_j - \lambda_j.$$

Consequently, maximization with respect to λ_j yields

$$\hat{\lambda}_j = \frac{\sum_{t=1}^n \pi'_j(t|n) y_t}{\sum_{t=1}^n \pi'_j(t|n)}, \quad j = 1, \dots, m.$$

We fit the model to the time series of earthquake counts using the R package `depmixS4`. The package, which uses the EM algorithm, does not provide standard errors, so we obtained them by a parametric bootstrap procedure; see Remillard (2011) for justification. The MLEs of the intensities, along with their standard errors, were $(\hat{\lambda}_1, \hat{\lambda}_2) = (15.4_{(.7)}, 26.0_{(1.1)})$. The MLE of the transition matrix was $\hat{\pi}_{11}, \hat{\pi}_{12}, \hat{\pi}_{21}, \hat{\pi}_{22} = [.93_{(.04)}, .07_{(.04)}, .12_{(.09)}, .88_{(.09)}]$. Figure 6.13 displays the counts, the estimated state (displayed as points) and the smoothing distribution for the earthquakes data, modeled as a two-state Poisson HMM model with parameters fitted using the MLEs. Finally, a histogram of the data is displayed along with the two estimated Poisson densities superimposed as solid lines.

The R code for this example is as follows.

```
library(depmixS4)
model <- depmix(EQcount ~1, nstates=2, data=data.frame(EQcount),
                 family=poisson())
set.seed(90210)
summary(fm <- fit(model)) # estimation results
##-- Get Parameters --##
u = as.vector(getpars(fm)) # ensure state 1 has smaller lambda
if (u[7] <= u[8]) { para.mle = c(u[3:6], exp(u[7]), exp(u[8])) }
else { para.mle = c(u[6:3], exp(u[8]), exp(u[7])) }
mtrans = matrix(para.mle[1:4], byrow=TRUE, nrow=2)
lams = para.mle[5:6]
pi1 = mtrans[2,1]/(2 - mtrans[1,1] - mtrans[2,2]); pi2 = 1-pi1
##-- Graphics --##
layout(matrix(c(1,2,1,3), 2))
par(mar = c(3,3,1,1), mgp = c(1.6,.6,0))
# data and states
plot(EQcount, main="", ylab='EQcount', type='h', col=gray(.7))
text(EQcount, col=6*posterior(fm)[,1]-2, labels=posterior(fm)[,1], cex=.9)
# prob of state 2
plot(ts(posterior(fm)[,3], start=1900), ylab =
      expression(hat(pi)[~2]*(t|n'))); abline(h=.5, lty=2)
# histogram
hist(EQcount, breaks=30, prob=TRUE, main="")
xvals = seq(1,45)
u1 = pi1*dpois(xvals, lams[1])
u2 = pi2*dpois(xvals, lams[2])
lines(xvals, u1, col=4); lines(xvals, u2, col=2)
##-- Bootstrap --##
# function to generate data
pois.HMM.generate_sample = function(n,m,lambda,Mtrans,StatDist=NULL){
  # n = data length, m = number of states, Mtrans = transition matrix,
  # StatDist = stationary distn
  if(is.null(StatDist)) StatDist = solve(t(diag(m)-Mtrans +1),rep(1,m))
  mvect = 1:m
  state = numeric(n)
  state[1] = sample(mvect ,1, prob=StatDist)
  for (i in 2:n)
    state[i] = sample(mvect ,1,prob=Mtrans[state[i-1] ,])
  y = rpois(n,lambda=lambda[state ])
  list(y=y, state=state) }
# start it up
set.seed(10101101)
```

```

nboot = 100
nobs = length(EQcount)
para.star = matrix(NA, nrow=nboot, ncol = 6)
for (j in 1:nboot){
  x.star = pois.HMM.generate_sample(n=nobs, m=2, lambda=lams, Mtrans=mtrans)$y
  model <- depmix(x.star ~1, nstates=2, data=data.frame(x.star),
    family=poisson())
  u = as.vector(getpars(fit(model, verbose=0)))
  # make sure state 1 is the one with the smaller intensity parameter
  if (u[7] <= u[8]) { para.star[j,] = c(u[3:6], exp(u[7]), exp(u[8])) }
  else { para.star[j,] = c(u[6:3], exp(u[8]), exp(u[7])) } }
# bootstrapped std errors
SE = sqrt(apply(para.star,2,var) +
  (apply(para.star,2,mean)-para.mle)^2)[c(1,4:6)]
names(SE)=c('seM11/M12', 'seM21/M22', 'seLam1', 'seLam2'); SE

```

Next, we present an example using a mixture of normal distributions.

Example 6.17 Normal HMM – S&P500 Weekly Returns

Estimation in the Gaussian case is similar to the Poisson case given in Example 6.16, except that now, $p_j(y_t)$ is the normal density; i.e., $(y_t \mid x_t = j) \sim N(\mu_j, \sigma_j^2)$ for $j = 1, \dots, m$. Then, dealing with the third term in (6.139) in this case yields

$$\hat{\mu}_j = \frac{\sum_{t=1}^n \pi'_j(t|n) y_t}{\sum_{t=1}^n \pi'_j(t|n)}, \quad \hat{\sigma}_j^2 = \frac{\sum_{t=1}^n \pi'_j(t|n) y_t^2}{\sum_{t=1}^n \pi'_j(t|n)} - \hat{\mu}_j^2.$$

In this example, we fit a normal HMM using the R package `depmixS4` to the weekly S&P 500 returns displayed in Figure 6.14. We chose a three-state model and we leave it to the reader to investigate a two-state model (see Problem 6.24). Standard errors (shown in parentheses below) were obtained via a parametric bootstrap based on a simulation script provided with the package.

If we let $P = \{\pi_{ij}\}$ denote the 3×3 matrix of transition probabilities, the fitted transition matrix was

$$\hat{P} = \begin{bmatrix} .945_{(.074)} & .055_{(.074)} & .000_{(.000)} \\ .739_{(.275)} & .000_{(.000)} & .261_{(.275)} \\ .032_{(.122)} & .027_{(.057)} & .942_{(.147)} \end{bmatrix},$$

and the three fitted normals were $N(\hat{\mu}_1 = .004_{(.173)}, \hat{\sigma}_1 = .014_{(.968)})$, $N(\hat{\mu}_2 = -.034_{(.909)}, \hat{\sigma}_2 = .009_{(.777)})$, and $N(\hat{\mu}_3 = -.003_{(.317)}, \hat{\sigma}_3 = .044_{(.910)})$. The data, along with the predicted state (based on the smoothing distribution), are plotted in Figure 6.14.

Note that regime 2 appears to represent a somewhat large-in-magnitude negative return, and may be a lone dip, or the start or end of a highly volatile period. States 1 and 3 represent clusters of regular or high volatility, respectively. Note that there is a large amount of uncertainty in the fitted normals, and in the transition matrix involving transitions from state 2 to states 1 or 3. The R code for this example is:

```

library(depmixS4)
y = ts(sp500w, start=2003, freq=52) # make data depmix friendly

```

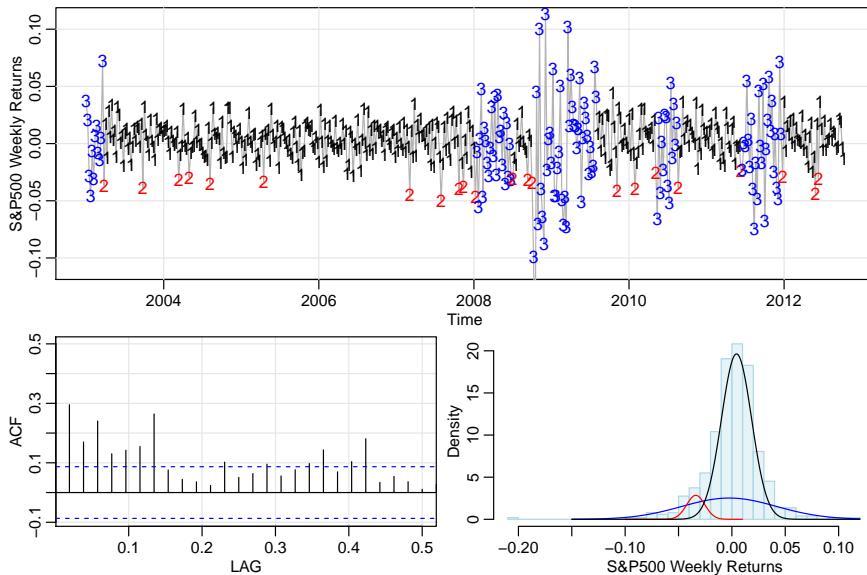


Fig. 6.14. Top: S&P 500 weekly returns with estimated regimes labeled as a number, 1, 2, or 3. The minimum value of -20% during the financial crisis has been truncated to improve the graphics. Bottom left: Sample ACF of the squared returns. Bottom right: Histogram of the data with the three estimated normal densities superimposed.

```

mod3 <- depmix(y~1, nstates=3, data=data.frame(y))
set.seed(2)
summary(fm3 <- fit(mod3))
##-- Graphics --##
layout(matrix(c(1,2, 1,3), 2), heights=c(1,.75))
par(mar=c(2.5,2.5,.5,.5), mgp=c(1.6,.6,0))
plot(y, main="", ylab='S&P500 Weekly Returns', col=gray(.7),
      ylim=c(-.11,.11))
culer = 4-posterior(fm3)[,1]; culer[culer==3]=4 # switch labels 1 and 3
text(y, col=culer, labels=4-posterior(fm3)[,1])
##-- MLEs --##
para.mle = as.vector(getpars(fm3)[-1:3])
permu = matrix(c(0,0,1,0,1,0,1,0,0), 3,3) # for the label switch
(mtrans.mle = permu%*%round(t(matrix(para.mle[1:9], 3,3)),3)%*%permu)
(norms.mle = round(matrix(para.mle[10:15],2,3),3)%*%permu)
acf(y^2, xlim=c(.02,.5), ylim=c(-.09,.5), panel.first=grid(lty=2) )
hist(y, 25, prob=TRUE, main='')
culer=c(1,2,4); pi.hat = colSums(posterior(fm3)[-1,2:4])/length(y)
for (i in 1:3) { mu=norms.mle[1,i]; sig = norms.mle[2,i]
x = seq(-.15,.12, by=.001)
lines(x, pi.hat[4-i]*dnorm(x, mean=mu, sd=sig), col=culer[i]) }
##-- Bootstrap --##
set.seed(666); n.obs = length(y); n.boot = 100
para.star = matrix(NA, nrow=n.boot, ncol = 15)
respst <- para.mle[10:15]; trst <- para.mle[1:9]
for ( nb in 1:n.boot ) {

```

```

mod <- simulate(mod3)
y.star = as.vector(mod@response[[1]][[1]]@y)
dfy = data.frame(y.star)
mod.star <- depmix(y.star~1, data=dfy, respst=respst, trst=trst, nst=3)
fm.star = fit(mod.star, emcontrol=em.control(tol = 1e-5), verbose=FALSE)
para.star[nb,] = as.vector(getpars(fm.star)[-c(1:3)])
para.star[nb,] = as.vector(getpars(fm.star)[-c(1:3)])
# bootstrap stdn errors
SE = sqrt(apply(para.star, 2, var) + (apply(para.star, 2, mean)-para.mle)^2)
(SE.mtrns.mle = permute%*%round(t(matrix(SE[1:9], 3, 3)), 3)%*%permute)
(SE.norms.mle = round(matrix(SE[10:15], 2, 3), 3)%*%permute)

```

It is worth mentioning that *switching regressions* also fits into this framework. In this case, we would change μ_j in the model in [Example 6.17](#) to depend on independent inputs, say z_{t1}, \dots, z_{tr} , so that

$$\mu_j = \beta_0^{(j)} + \sum_{i=1}^r \beta_i^{(j)} z_{ti}.$$

This type of model is easily handled using the `depmixS4` R package.

By conditioning on the first few observations, it is also possible to include simple switching linear autoregression into this framework. In this case, we model the observations as being an AR(p), with parameters depending on the state; that is,

$$y_t = \phi_0^{(x_t)} + \sum_{i=1}^p \phi_i^{(x_t)} y_{t-i} + \sigma^{(x_t)} v_t, \quad (6.147)$$

and $v_t \sim \text{iid } N(0, 1)$. The model is similar to the threshold model discussed in [Section 5.4](#), however, the process is not self-exciting or influenced by an observed exogenous process. In (6.147), we are saying that the parameters are random, and the regimes are changing due to a latent Markov process. In a similar fashion to (6.131), we write the conditional distribution of the observations as

$$p_j(y_t) = p(y_t \mid x_t = j, y_{t-1:t-p}), \quad (6.148)$$

and we note that for $t > p$, $p_j(y_t)$ is the normal density (g),

$$p_j(y_t) = g\left(y_t; \phi_0^{(j)} + \sum_{i=1}^p \phi_i^{(j)} y_{t-i}, \sigma^{2(j)}\right). \quad (6.149)$$

As in (6.138), the conditional likelihood is given by

$$\ln L_Y(\Theta \mid y_{1:p}) = \sum_{t=p+1}^n \ln \left(\sum_{j=1}^m \pi_j(t \mid t-1) p_j(y_t) \right).$$

where [Property 6.7](#) still applies, but with the updated evaluation of $p_j(y_t)$ given in (6.149). In addition, the EM algorithm may be used analogously by assessing the smoothers. The smoothers in this case are symbolically the same as given in [Property 6.8](#) with the appropriate definition changes, $p_j(y_t)$ as given in (6.148) and with $\varphi_j(t) = p(y_{t+1:n} \mid x_t = j, y_{t+1-p:t})$ for $t > p$.

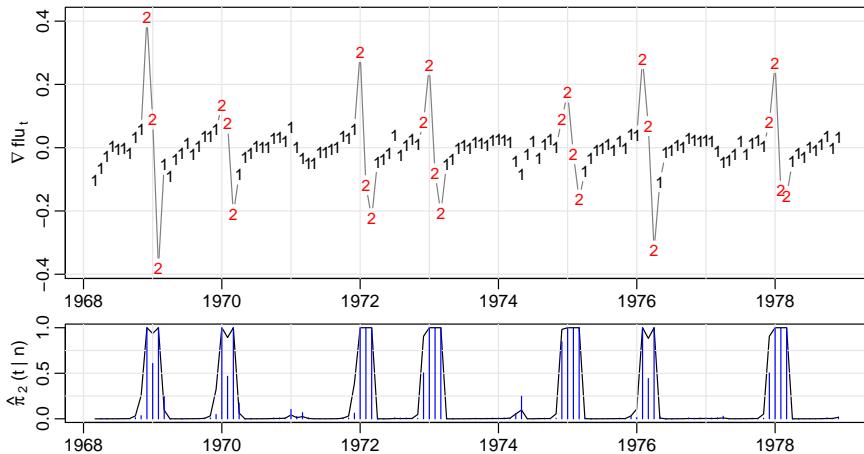


Fig. 6.15. The differenced flu mortality data along with the estimated states (displayed as points). The smoothed state 2 probabilities are displayed in the bottom of the figure as a straight line. The filtered state 2 probabilities are displayed as vertical lines.

Example 6.18 Switching AR – Influenza Mortality

In Example 5.7, we discussed the monthly pneumonia and influenza mortality series shown in Figure 5.7. We pointed out the non-reversibility of the series, which rules out the possibility that the data are generated by a linear Gaussian process. In addition, note that the series is irregular, and while mortality is highest during the winter, the peak does not occur in the same month each year. Moreover, some seasons have very large peaks, indicating flu epidemics, whereas other seasons are mild. In addition, it can be seen from Figure 5.7 that there is a slight negative trend in the data set, indicating that flu prevention is getting better over the eleven year period.

As in Example 5.7, we focus on the differenced data, which removes the trend. In this case, we denote $y_t = \nabla \text{flu}_t$, where flu_t represents the data displayed in Figure 5.7. Since we already fit a threshold model to y_t , we might also consider a switching autoregressive model where there are two hidden regimes, one for epidemic periods and one for more mild periods. In this case, the model is given by

$$y_t = \begin{cases} \phi_0^{(1)} + \sum_{j=1}^p \phi_j^{(1)} y_{t-j} + \sigma^{(1)} v_t, & \text{for } x_t = 1, \\ \phi_0^{(2)} + \sum_{j=1}^p \phi_j^{(2)} y_{t-j} + \sigma^{(2)} v_t, & \text{for } x_t = 2, \end{cases} \quad (6.150)$$

where $v_t \sim \text{iid } N(0, 1)$, and x_t is a latent, two-state Markov chain.

We used the R package `MSwM` to fit the model specified in (6.150), with $p = 2$. The results were

$$\hat{y}_t = \begin{cases} .006_{(.003)} + .293_{(.039)} y_{t-1} + .097_{(.031)} y_{t-2} + .024 v_t, & \text{for } x_t = 1, \\ .199_{(.063)} - .313_{(.281)} y_{t-1} - 1.604_{(.276)} y_{t-2} + .112 v_t, & \text{for } x_t = 2, \end{cases}$$

with estimated transition matrix

$$\hat{P} = \begin{bmatrix} .93 & .07 \\ .30 & .70 \end{bmatrix}.$$

Figure 6.15 displays the data $y_t = \nabla \text{flu}_t$ along with the estimated states (displayed as points labeled 1 or 2). The smoothed state 2 probabilities are displayed in the bottom of the figure as a straight line. The filtered state 2 probabilities are displayed in the same graph as vertical lines. The code for this example is as follows.

```
library(MSwM)
set.seed(90210)
dflu = diff(flu)
model = lm(dflu~ 1)
mod = msmFit(model, k=2, p=2, sw=rep(TRUE, 4)) # 2 regimes, AR(2)s
summary(mod)
plotProb(mod, which=3)
```

6.10 Dynamic Linear Models with Switching

In this section, we extend the hidden Markov model discussed in [Section 6.9](#) to more general problems. As previously indicated, the problem of modeling changes in regimes for time series has been of interest in many different fields, and we have explored these ideas in [Section 5.4](#) as well as in [Section 6.9](#).

Generalizations of the state space model to include the possibility of changes occurring over time have been approached by allowing changes in the error covariances (Harrison and Stevens, 1976, Gordon and Smith, 1988, 1990) or by assigning mixture distributions to the observation errors v_t (Peña and Guttman, 1988). Approximations to filtering were derived in all of the aforementioned articles. An application to monitoring renal transplants was described in Smith and West (1983) and in Gordon and Smith (1990). Gerlach et al. (2000) considered an extension of the switching AR model to allow for level shifts and outliers in both the observations and innovations. An application of the idea of switching to the tracking of multiple targets has been considered in Bar-Shalom (1978), who obtained approximations to Kalman filtering in terms of weighted averages of the innovations. For a thorough coverage of these and related techniques, see Cappé, Moulines, & Rydén (2009) and Douc, Moulines, & Stoffer (2014).

In this section, we will concentrate on the method presented in Shumway and Stoffer (1991). One way of modeling change in an evolving time series is by assuming the dynamics of some underlying model changes discontinuously at certain undetermined points in time. Our starting point is the DLM given by [\(6.1\)](#) and [\(6.2\)](#), namely,

$$x_t = \Phi x_{t-1} + w_t, \tag{6.151}$$

to describe the $p \times 1$ state dynamics, and

$$y_t = A_t x_t + v_t \tag{6.152}$$

to describe the $q \times 1$ observation dynamics. Recall w_t and v_t are Gaussian white noise sequences with $\text{var}(w_t) = Q$, $\text{var}(v_t) = R$, and $\text{cov}(w_t, v_s) = 0$ for all s and t .

Example 6.19 Tracking Multiple Targets

The approach of Shumway and Stoffer (1991) was motivated primarily by the problem of tracking a large number of moving targets using a vector y_t of sensors. In this problem, we do not know at any given point in time which target any given sensor has detected. Hence, it is the structure of the measurement matrix A_t in (6.152) that is changing, and not the dynamics of the signal x_t or the noises, w_t or v_t . As an example, consider a 3×1 vector of satellite measurements $y_t = (y_{t1}, y_{t2}, y_{t3})'$ that are observations on some combination of a 3×1 vector of targets or signals, $x_t = (x_{t1}, x_{t2}, x_{t3})'$. For the measurement matrix

$$A_t = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

for example, the first sensor, y_{t1} , observes the second target, x_{t2} ; the second sensor, y_{t2} , observes the first target, x_{t1} ; and the third sensor, y_{t3} , observes the third target, x_{t3} . All possible detection configurations will define a set of possible values for A_t , say, $\{M_1, M_2, \dots, M_m\}$, as a collection of plausible measurement matrices.

Example 6.20 Modeling Economic Change

As another example of the switching model presented in this section, consider the case in which the dynamics of the linear model changes suddenly over the history of a given realization. For example, Lam (1990) has given the following generalization of Hamilton's (1989) model for detecting positive and negative growth periods in the economy. Suppose the data are generated by

$$y_t = z_t + n_t, \quad (6.153)$$

where z_t is an autoregressive series and n_t is a random walk with a drift that switches between two values α_0 and $\alpha_0 + \alpha_1$. That is,

$$n_t = n_{t-1} + \alpha_0 + \alpha_1 S_t, \quad (6.154)$$

with $S_t = 0$ or 1 , depending on whether the system is in state 1 or state 2. For the purpose of illustration, suppose

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + w_t \quad (6.155)$$

is an AR(2) series with $\text{var}(w_t) = \sigma_w^2$. Lam (1990) wrote (6.153) in a differenced form

$$\nabla y_t = z_t - z_{t-1} + \alpha_0 + \alpha_1 S_t, \quad (6.156)$$

which we may take as the observation equation (6.152) with state vector

$$x_t = (z_t, z_{t-1}, \alpha_0, \alpha_1)' \quad (6.157)$$

and

$$M_1 = [1, -1, 1, 0] \quad \text{and} \quad M_2 = [1, -1, 1, 1] \quad (6.158)$$

determining the two possible economic conditions. The state equation, (6.151), is of the form

$$\begin{pmatrix} z_t \\ z_{t-1} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} = \begin{bmatrix} \phi_1 & \phi_2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} z_{t-1} \\ z_{t-2} \\ \alpha_0 \\ \alpha_1 \end{pmatrix} + \begin{pmatrix} w_t \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (6.159)$$

The observation equation, (6.156), can be written as

$$\nabla y_t = A_t x_t + v_t, \quad (6.160)$$

where we have included the possibility of observational noise, and where $\Pr(A_t = M_1) = 1 - \Pr(A_t = M_2)$, with M_1 and M_2 given in (6.158).

To incorporate a reasonable switching structure for the measurement matrix into the DLM that is compatible with both practical situations previously described, we assume that the m possible configurations are states in a nonstationary, independent process defined by the time-varying probabilities

$$\pi_j(t) = \Pr(A_t = M_j), \quad (6.161)$$

for $j = 1, \dots, m$ and $t = 1, 2, \dots, n$. Important information about the current state of the measurement process is given by the filtered probabilities of being in state j , defined as the conditional probabilities

$$\pi_j(t | t) = \Pr(A_t = M_j | y_{1:t}), \quad (6.162)$$

which also vary as a function of time. Recall that $y_{s':s} = \{y_{s'}, \dots, y_s\}$. The filtered probabilities (6.162) give the time-varying estimates of the probability of being in state j given the data to time t .

It will be important for us to obtain estimators of the configuration probabilities, $\pi_j(t | t)$, the predicted and filtered state estimators, x_t^{t-1} and x_t^t , and the corresponding error covariance matrices P_t^{t-1} and P_t^t . Of course, the predictor and filter estimators will depend on the parameters, Θ , of the DLM. In many situations, the parameters will be unknown and we will have to estimate them. Our focus will be on maximum likelihood estimation, but other authors have taken a Bayesian approach that assigns priors to the parameters, and then seeks posterior distributions of the model parameters; see, for example, Gordon and Smith (1990), Peña and Guttman (1988), or McCulloch and Tsay (1993).

We now establish the recursions for the filters associated with the state x_t and the switching process, A_t . As discussed in Section 6.3, the filters are also an essential part of the maximum likelihood procedure. The predictors, $x_t^{t-1} = E(x_t | y_{1:t-1})$, and filters, $x_t^t = E(x_t | y_{1:t})$, and their associated error variance–covariance matrices, P_t^{t-1} and P_t^t , are given by

$$x_t^{t-1} = \Phi x_{t-1}^{t-1}, \quad (6.163)$$

$$P_t^{t-1} = \Phi P_{t-1}^{t-1} \Phi' + Q, \quad (6.164)$$

$$x_t^t = x_t^{t-1} + \sum_{j=1}^m \pi_j(t|t) K_{tj} \epsilon_{tj}, \quad (6.165)$$

$$P_t^t = \sum_{j=1}^m \pi_j(t|t) (I - K_{tj} M_j) P_t^{t-1}, \quad (6.166)$$

$$K_{tj} = P_t^{t-1} M_j' \Sigma_{tj}^{-1}, \quad (6.167)$$

where the innovation values in (6.165) and (6.167) are

$$\epsilon_{tj} = y_t - M_j x_t^{t-1}, \quad (6.168)$$

$$\Sigma_{tj} = M_j P_t^{t-1} M_j' + R, \quad (6.169)$$

for $j = 1, \dots, m$.

Equations (6.163)–(6.167) exhibit the filter values as weighted linear combinations of the m innovation values, (6.168)–(6.169), corresponding to each of the possible measurement matrices. The equations are similar to the approximations introduced by Bar-Shalom and Tse (1975), by Gordon and Smith (1990), and Peña and Guttman (1988).

To verify (6.165), let the indicator $I(A_t = M_j) = 1$ when $A_t = M_j$, and zero otherwise. Then, using (6.20),

$$\begin{aligned} x_t^t &= E(x_t \mid y_{1:t}) = E[E(x_t \mid y_{1:t}, A_t) \mid y_{1:t}] \\ &= E\left\{\sum_{j=1}^m E(x_t \mid y_{1:t}, A_t = M_j) I(A_t = M_j) \mid y_{1:t}\right\} \\ &= E\left\{\sum_{j=1}^m [x_t^{t-1} + K_{tj}(y_t - M_j x_t^{t-1})] I(A_t = M_j) \mid y_{1:t}\right\} \\ &= \sum_{j=1}^m \pi_j(t \mid t) [x_t^{t-1} + K_{tj}(y_t - M_j x_t^{t-1})], \end{aligned}$$

where K_{tj} is given by (6.167). Equation (6.166) is derived in a similar fashion; the other relationships, (6.163), (6.164), and (6.167), follow from straightforward applications of the Kalman filter results given in [Property 6.1](#).

Next, we derive the filters $\pi_j(t|t)$. Let $p_j(t \mid t-1)$ denote the conditional density of y_t given the past $y_{1:t-1}$, and $A_t = M_j$, for $j = 1, \dots, m$. Then,

$$\pi_j(t \mid t) = \frac{\pi_j(t) p_j(t \mid t-1)}{\sum_{k=1}^m \pi_k(t) p_k(t \mid t-1)}, \quad (6.170)$$

where we assume the distribution $\pi_j(t)$, for $j = 1, \dots, m$ has been specified before observing $y_{1:t}$ (details follow as in [Example 6.21](#) below). If the investigator has

no reason to prefer one state over another at time t , the choice of uniform priors, $\pi_j(t) = m^{-1}$, for $j = 1, \dots, m$, will suffice. Smoothness can be introduced by letting

$$\pi_j(t) = \sum_{i=1}^m \pi_i(t-1 \mid t-1) \pi_{ij}, \quad (6.171)$$

where the non-negative weights π_{ij} are chosen so $\sum_{i=1}^m \pi_{ij} = 1$. If the A_t process was Markov with transition probabilities π_{ij} , then (6.171) would be the update for the filter probability, as shown in the next example.

Example 6.21 Hidden Markov Chain Model

If $\{A_t\}$ is a hidden Markov chain with stationary transition probabilities $\pi_{ij} = \Pr(A_t = M_j \mid A_{t-1} = M_i)$, for $i, j = 1, \dots, m$, we have

$$\begin{aligned} \pi_j(t \mid t) &= \frac{p(A_t = M_j, y_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})} \\ &= \frac{\Pr(A_t = M_j \mid y_{1:t-1}) p(y_t \mid A_t = M_j, y_{1:t-1})}{p(y_t \mid y_{1:t-1})} \\ &= \frac{\pi_j(t \mid t-1) p_j(t \mid t-1)}{\sum_{k=1}^m \pi_k(t \mid t-1) p_k(t \mid t-1)}. \end{aligned} \quad (6.172)$$

In the Markov case, the conditional probabilities

$$\pi_j(t \mid t-1) = \Pr(A_t = M_j \mid y_{1:t-1})$$

in (6.172) replace the unconditional probabilities, $\pi_j(t) = \Pr(A_t = M_j)$, in (6.170).

To evaluate (6.172), we must be able to calculate $\pi_j(t \mid t-1)$ and $p_j(t \mid t-1)$. We will discuss the calculation of $p_j(t \mid t-1)$ after this example. To derive $\pi_j(t \mid t-1)$, note,

$$\begin{aligned} \pi_j(t \mid t-1) &= \Pr(A_t = M_j \mid y_{1:t-1}) \\ &= \sum_{i=1}^m \Pr(A_t = M_j, A_{t-1} = M_i \mid y_{1:t-1}) \\ &= \sum_{i=1}^m \Pr(A_t = M_j \mid A_{t-1} = M_i) \Pr(A_{t-1} = M_i \mid y_{1:t-1}) \\ &= \sum_{i=1}^m \pi_{ij} \pi_i(t-1 \mid t-1). \end{aligned} \quad (6.173)$$

Expression (6.171) comes from equation (6.173), where, as previously noted, we replace $\pi_j(t \mid t-1)$ by $\pi_j(t)$.

The difficulty in extending the approach here to the Markov case is the dependence among the y_t , which makes it necessary to enumerate over all possible histories to derive the filtering equations. This problem will be evident when we derive the

conditional density $p_j(t \mid t - 1)$. Equation (6.171) has $\pi_j(t)$ as a function of the past observations, $y_{1:t-1}$, which is inconsistent with our model assumption. Nevertheless, this seems to be a reasonable compromise that allows the data to modify the probabilities $\pi_j(t)$, without having to develop a highly computer-intensive technique.

As previously suggested, the computation of $p_j(t \mid t - 1)$, without some approximations, is highly computer-intensive. To evaluate $p_j(t \mid t - 1)$, consider the event

$$\{A_1 = M_{j_1}, \dots, A_{t-1} = M_{j_{t-1}}\}, \quad (6.174)$$

for $j_i = 1, \dots, m$, and $i = 1, \dots, t - 1$, which specifies a specific set of measurement matrices through the past; we will write this event as $A_{(t-1)} = M_{(\ell)}$. Because m^{t-1} possible outcomes exist for A_1, \dots, A_{t-1} , the index ℓ runs through $\ell = 1, \dots, m^{t-1}$. Using this notation, we may write

$$\begin{aligned} p_j(t \mid t - 1) &= \sum_{\ell=1}^{m^{t-1}} \Pr\{A_{(t-1)} = M_{(\ell)} \mid y_{1:t-1}\} p(y_t \mid y_{1:t-1}, A_t = M_j, A_{(t-1)} = M_{(\ell)}) \\ &\stackrel{\text{def}}{=} \sum_{\ell=1}^{m^{t-1}} \alpha(\ell) g(y_t; \mu_{tj}(\ell), \Sigma_{tj}(\ell)), \quad j = 1, \dots, m, \end{aligned} \quad (6.175)$$

where $g(\cdot; \mu, \Sigma)$ represents the normal density with mean vector μ and variance-covariance matrix Σ . Thus, $p_j(t \mid t - 1)$ is a mixture of normals with non-negative weights $\alpha(\ell) = \Pr\{A_{(t-1)} = M_{(\ell)} \mid y_{1:t-1}\}$ such that $\sum_{\ell} \alpha(\ell) = 1$, and with each normal distribution having mean vector

$$\mu_{tj}(\ell) = M_j x_t^{t-1}(\ell) = M_j E[x_t \mid y_{1:t-1}, A_{(t-1)} = M_{(\ell)}] \quad (6.176)$$

and covariance matrix

$$\Sigma_{tj}(\ell) = M_j P_t^{t-1}(\ell) M_j' + R. \quad (6.177)$$

This result follows because the conditional distribution of y_t in (6.175) is identical to the fixed measurement matrix case presented in Section 6.2. The values in (6.176) and (6.177), and hence the densities, $p_j(t \mid t - 1)$, for $j = 1, \dots, m$, can be obtained directly from the Kalman filter, Property 6.1, with the measurement matrices $A_{(t-1)}$ fixed at $M_{(\ell)}$.

Although $p_j(t \mid t - 1)$ is given explicitly in (6.175), its evaluation is highly computer intensive. For example, with $m = 2$ states and $n = 20$ observations, we have to filter over $2+2^2+\dots+2^{20}$ possible sample paths ($2^{20} = 1,048,576$). There are a few remedies to this problem. An algorithm that makes it possible to efficiently compute the most likely sequence of states given the data is known as the *Viterbi algorithm*, which is based on the well-known dynamic programming principle. Details may be found in Douc et al. (2014, §9.2). Another remedy is to trim (remove), at each t , highly improbable sample paths; that is, remove events in (6.174) with extremely small probability of occurring, and then evaluate $p_j(t \mid t - 1)$ as if the trimmed sample paths could not have occurred. Another rather simple alternative, as suggested by Gordon

and Smith (1990) and Shumway and Stoffer (1991), is to approximate $p_j(t | t - 1)$ using the closest (in the sense of Kulback–Leibler distance) normal distribution. In this case, the approximation leads to choosing normal distribution with the same mean and variance associated with $p_j(t | t - 1)$; that is, we approximate $p_j(t | t - 1)$ by a normal with mean $M_j x_t^{t-1}$ and variance Σ_{tj} given in (6.169).

To develop a procedure for maximum likelihood estimation, the joint density of the data is

$$\begin{aligned} f(y_1, \dots, y_n) &= \prod_{t=1}^n f(y_t | y_{1:t-1}) \\ &= \prod_{t=1}^n \sum_{j=1}^m \Pr(A_t = M_j | y_{1:t-1}) p(y_t | A_t = M_j, y_{1:t-1}), \end{aligned}$$

and hence, the likelihood can be written as

$$\ln L_Y(\Theta) = \sum_{t=1}^n \ln \left(\sum_{j=1}^m \pi_j(t) p_j(t | t - 1) \right). \quad (6.178)$$

For the hidden Markov model, $\pi_j(t)$ would be replaced by $\pi_j(t | t - 1)$. In (6.178), we will use the normal approximation to $p_j(t | t - 1)$. That is, henceforth, we will consider $p_j(t | t - 1)$ as the normal, $N(M_j x_t^{t-1}, \Sigma_{tj})$, density, where x_t^{t-1} is given in (6.163) and Σ_{tj} is given in (6.169). We may consider maximizing (6.178) directly as a function of the parameters Θ in $\{\mu_0, \Phi, Q, R\}$ using a Newton method, or we may consider applying the EM algorithm to the complete data likelihood.

To apply the EM algorithm as in Section 6.3, we call $x_{0:n}$, $A_{1:n}$, and $y_{1:n}$, the complete data, with likelihood given by

$$\begin{aligned} -2 \ln L_{X,A,Y}(\Theta) &= \ln |\Sigma_0| + (x_0 - \mu_0)' \Sigma_0^{-1} (x_0 - \mu_0) \\ &\quad + n \ln |Q| + \sum_{t=1}^n (x_t - \Phi x_{t-1})' Q^{-1} (x_t - \Phi x_{t-1}) \\ &\quad - 2 \sum_{t=1}^n \sum_{j=1}^m I(A_t = M_j) \ln \pi_j(t) + n \ln |R| \\ &\quad + \sum_{t=1}^n \sum_{j=1}^m I(A_t = M_j) (y_t - A_t x_t)' R^{-1} (y_t - A_t x_t). \end{aligned} \quad (6.179)$$

As discussed in Section 6.3, we require the minimization of the conditional expectation

$$Q(\Theta | \Theta^{(k-1)}) = E \left\{ -2 \ln L_{X,A,Y}(\Theta) \mid y_{1:n}, \Theta^{(k-1)} \right\}, \quad (6.180)$$

with respect to Θ at each iteration, $k = 1, 2, \dots$. The calculation and maximization of (6.180) is similar to the case of (6.63). In particular, with

$$\pi_j(t | n) = E[I(A_t = M_j) \mid y_{1:n}], \quad (6.181)$$

we obtain on iteration k ,

$$\pi_j^{(k)}(t) = \pi_j(t | n), \quad (6.182)$$

$$\mu_0^{(k)} = x_0^n, \quad (6.183)$$

$$\Phi^{(k)} = S_{10}S_{00}^{-1}, \quad (6.184)$$

$$Q^{(k)} = n^{-1} \left(S_{11} - S_{10}S_{00}^{-1}S'_{10} \right), \quad (6.185)$$

and

$$R^{(k)} = n^{-1} \sum_{t=1}^n \sum_{j=1}^m \pi_j(t|n) \left[(y_t - M_j x_t^n)(y_t - M_j x_t^n)' + M_j P_t^n M_j' \right]. \quad (6.186)$$

where S_{11}, S_{10}, S_{00} are given in (6.65)–(6.67). As before, at iteration k , the filters and the smoothers are calculated using the current values of the parameters, $\Theta^{(k-1)}$, and Σ_0 is held fixed. Filtering is accomplished by using (6.163)–(6.167). Smoothing is derived in a similar manner to the derivation of the filter, and one is led to the smoother given in [Property 6.2](#) and [Property 6.3](#), with one exception, the initial smoother covariance, (6.53), is now

$$P_{n,n-1}^n = \sum_{j=1}^m \pi_j(n|n) (I - K_{tj} M_j) \Phi P_{n-1}^{n-1}. \quad (6.187)$$

Unfortunately, the computation of $\pi_j(t | n)$ is excessively complicated, and requires integrating over mixtures of normal distributions. Shumway and Stoffer (1991) suggest approximating the smoother $\pi_j(t | n)$ by the filter $\pi_j(t | t)$, and find the approximation works well.

Example 6.22 Analysis of the Influenza Data

We use the results of this section to analyze the U.S. monthly pneumonia and influenza mortality data plotted in [Figure 5.7](#). Letting y_t denote the observations at month t , we model y_t in terms of a structural component model coupled with a hidden Markov process that determines whether a flu epidemic exists.

The model consists of three structural components. The first component, x_{t1} , is an AR(2) process chosen to represent the periodic (seasonal) component of the data,

$$x_{t1} = \alpha_1 x_{t-1,1} + \alpha_2 x_{t-2,1} + w_{t1}, \quad (6.188)$$

where w_{t1} is white noise, with $\text{var}(w_{t1}) = \sigma_1^2$. The second component, x_{t2} , is an AR(1) process with a nonzero constant term, which is chosen to represent the sharp rise in the data during an epidemic,

$$x_{t2} = \beta_0 + \beta_1 x_{t-1,2} + w_{t2}, \quad (6.189)$$

where w_{t2} is white noise, with $\text{var}(w_{t2}) = \sigma_2^2$. The third component, x_{t3} , is a fixed trend component given by,

$$x_{t3} = x_{t-1,3} + w_{t3}, \quad (6.190)$$

where $\text{var}(w_{t3}) = 0$. The case in which $\text{var}(w_{t3}) > 0$, which corresponds to a stochastic trend (random walk), was tried here, but the estimation became unstable, and lead to us fitting a fixed, rather than stochastic, trend. Thus, in the final model, the trend component satisfies $\nabla x_{t3} = 0$; recall in [Example 6.18](#) the data were also differenced once before fitting the model.

Throughout the years, periods of normal influenza mortality (state 1) are modeled as

$$y_t = x_{t1} + x_{t3} + v_t, \quad (6.191)$$

where the measurement error, v_t , is white noise with $\text{var}(v_t) = \sigma_v^2$. When an epidemic occurs (state 2), mortality is modeled as

$$y_t = x_{t1} + x_{t2} + x_{t3} + v_t. \quad (6.192)$$

The model specified in [\(6.188\)](#)–[\(6.192\)](#) can be written in the general state-space form. The state equation is

$$\begin{pmatrix} x_{t1} \\ x_{t-1,1} \\ x_{t2} \\ x_{t3} \end{pmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \beta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{t-1,1} \\ x_{t-2,1} \\ x_{t-1,2} \\ x_{t-1,3} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \beta_0 \\ 0 \end{pmatrix} + \begin{pmatrix} w_{t1} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \quad (6.193)$$

Of course, [\(6.193\)](#) can be written in the standard state-equation form as

$$x_t = \Phi x_{t-1} + \Upsilon u_t + w_t, \quad (6.194)$$

where $x_t = (x_{t1}, x_{t-1,1}, x_{t2}, x_{t3})'$, $\Upsilon = (0, 0, \beta_0, 0)'$, $u_t \equiv 1$, and Φ is a 4×4 matrix with σ_1^2 as the (1,1)-element, σ_2^2 as the (3,3)-element, and the remaining elements set equal to zero. The observation equation is

$$y_t = A_t x_t + v_t, \quad (6.195)$$

where A_t is 1×4 , and v_t is white noise with $\text{var}(v_t) = R = \sigma_v^2$. We assume all components of variance w_{t1} , w_{t2} , and v_t are uncorrelated.

As discussed in [\(6.191\)](#) and [\(6.192\)](#), A_t can take one of two possible forms

$$\begin{aligned} A_t &= M_1 = [1, 0, 0, 1] && \text{no epidemic,} \\ A_t &= M_2 = [1, 0, 1, 1] && \text{epidemic,} \end{aligned}$$

corresponding to the two possible states of (1) no flu epidemic and (2) flu epidemic, such that $\Pr(A_t = M_1) = 1 - \Pr(A_t = M_2)$. In this example, we will assume A_t is a hidden Markov chain, and hence we use the updating equations given in [Example 6.21](#), [\(6.172\)](#) and [\(6.173\)](#), with transition probabilities $\pi_{11} = \pi_{22} = .75$ (and, thus, $\pi_{12} = \pi_{21} = .25$).

Parameter estimation was accomplished using a quasi-Newton–Raphson procedure to maximize the approximate log likelihood given in [\(6.178\)](#), with initial

Table 6.3. Estimation Results for Influenza Data

Parameter	Initial Model Estimates	Final Model Estimates
α_1	1.422 (.100)	1.406 (.079)
α_2	-.634 (.089)	-.622 (.069)
β_0	.276 (.056)	.210 (.025)
β_1	-.312 (.218)	—
σ_1	.023 (.003)	.023 (.005)
σ_2	.108 (.017)	.112 (.017)
σ_v	.002 (.009)	—

Estimated standard errors in parentheses

values of $\pi_1(1 | 0) = \pi_2(1 | 0) = .5$. **Table 6.3** shows the results of the estimation procedure. On the initial fit, two estimates are not significant, namely, $\hat{\beta}_1$ and $\hat{\sigma}_v$. When $\sigma_v^2 = 0$, there is no measurement error, and the variability in data is explained solely by the variance components of the state system, namely, σ_1^2 and σ_2^2 . The case in which $\beta_1 = 0$ corresponds to a simple level shift during a flu epidemic. In the final model, with β_1 and σ_v^2 removed, the estimated level shift ($\hat{\beta}_0$) corresponds to an increase in mortality by about .2 per 1000 during a flu epidemic. The estimates for the final model are also listed in **Table 6.3**.

Figure 6.16(a) shows a plot of the data, y_t , for the ten-year period of 1969–1978 as well as an indicator that takes the value of 1 if $\hat{\pi}_1(t | t - 1) \geq .5$, or 2 if $\hat{\pi}_2(t | t - 1) > .5$. The estimated prediction probabilities do a reasonable job of predicting a flu epidemic, although the peak in 1972 is missed.

Figure 6.16(b) shows the estimated filtered values (that is, filtering is done using the parameter estimates) of the three components of the model, x_{t1}^t , x_{t2}^t , and x_{t3}^t . Except for initial instability (which is not shown), \hat{x}_{t1}^t represents the seasonal (cyclic) aspect of the data, \hat{x}_{t2}^t represents the spikes during a flu epidemic, and \hat{x}_{t3}^t represents the slow decline in flu mortality over the ten-year period of 1969–1978.

One-month-ahead prediction, say, \hat{y}_t^{t-1} , is obtained as

$$\hat{y}_t^{t-1} = M_1 \hat{x}_t^{t-1} \quad \text{if } \hat{\pi}_1(t | t - 1) > \hat{\pi}_2(t | t - 1),$$

$$\hat{y}_t^{t-1} = M_2 \hat{x}_t^{t-1} \quad \text{if } \hat{\pi}_1(t | t - 1) \leq \hat{\pi}_2(t | t - 1).$$

Of course, \hat{x}_t^{t-1} is the estimated state prediction, obtained via the filter presented in (6.163)–(6.167) (with the addition of the constant term in the model) using the estimated parameters. The results are shown in **Figure 6.16(c)**. The precision of the forecasts can be measured by the innovation variances, Σ_{t1} when no epidemic is predicted, and Σ_{t2} when an epidemic is predicted. These values become stable quickly, and when no epidemic is predicted, the estimated standard prediction error is approximately .02 (this is the square root of Σ_{t1} for t large); when a flu epidemic is predicted, the estimated standard prediction error is approximately .11.

The results of this analysis are impressive given the small number of parameters and the degree of approximation that was made to obtain computationally

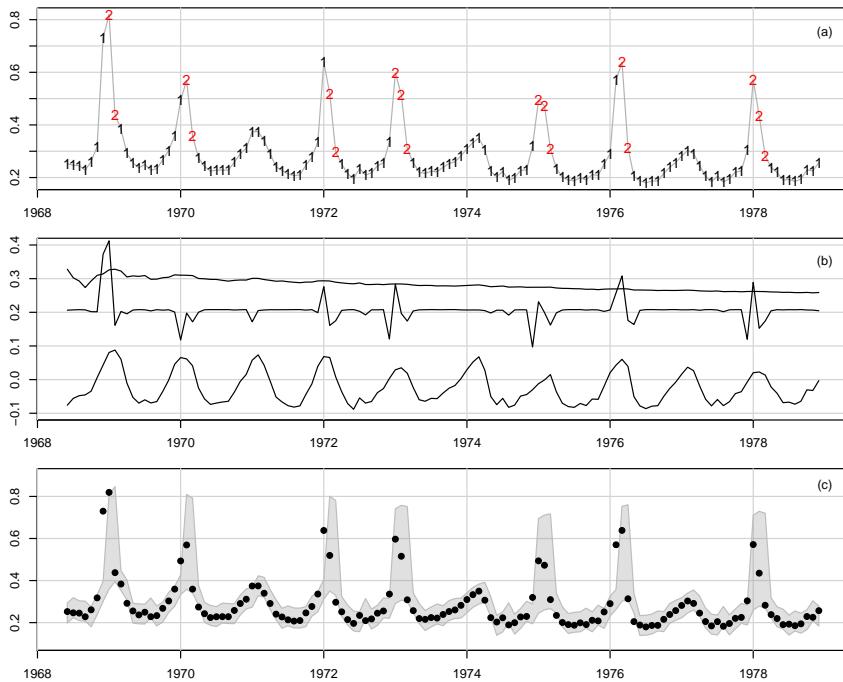


Fig. 6.16. (a) Influenza data, y_t , (line—points) and a prediction indicator (1 or 2) that an epidemic occurs in month t given the data up to month $t - 1$ (dashed line). (b) The three filtered structural components of influenza mortality: \hat{x}_{t1}^t (cyclic trace), \hat{x}_{t2}^t (spiked trace), and \hat{x}_{t3}^t (negative linear trace). (c) One-month-ahead predictions shown as upper and lower limits $\hat{y}_t^{t-1} \pm 2\sqrt{\hat{P}_t^{t-1}}$ (gray swatch), of the number of pneumonia and influenza deaths, and y_t (points).

simple method for fitting a complex model. Further evidence of the strength of this technique can be found in the example given in Shumway and Stoffer (1991).

The R code for the final model estimation is as follows.

```

y = as.matrix(flu); num = length(y); nstate = 4;
M1 = as.matrix(cbind(1,0,0,1)) # obs matrix normal
M2 = as.matrix(cbind(1,0,1,1)) # obs matrix flu epi
prob = matrix(0,num,1); yp = y # to store pi2(t/t-1) & y(t/t-1)
xfilter = array(0, dim=c(nstate,1,num)) # to store x(t/t)
# Function to Calculate Likelihood
Linn = function(para){
  alpha1 = para[1]; alpha2 = para[2]; beta0 = para[3]
  sQ1 = para[4]; sQ2 = para[5]; like=0
  xf = matrix(0, nstate, 1) # x filter
  xp = matrix(0, nstate, 1) # x pred
  Pf = diag(.1, nstate) # filter cov
  Pp = diag(.1, nstate) # pred cov
  pi11 <- .75 -> pi22; pi12 <- .25 -> pi21; pif1 <- .5 -> pif2
  phi = matrix(0,nstate,nstate)
}
```

```

phi[1,1] = alpha1; phi[1,2] = alpha2; phi[2,1]=1; phi[4,4]=1
Ups = as.matrix(rbind(0,0,beta0,0))
Q = matrix(0,nstate,nstate)
Q[1,1] = sQ1^2; Q[3,3] = sQ2^2; R=0 # R=0 in final model
# begin filtering #
for(i in 1:num){
  xp = phi%*%xf + Ups; Pp = phi%*%Pf%*%t(phi) + Q
  sig1 = as.numeric(M1%*%Pp%*%t(M1) + R)
  sig2 = as.numeric(M2%*%Pp%*%t(M2) + R)
  k1 = Pp%*%t(M1)/sig1; k2 = Pp%*%t(M2)/sig2
  e1 = y[i]-M1%*%xp; e2 = y[i]-M2%*%xp
  pip1 = pif1*pi11 + pif2*pi21; pip2 = pif1*pi12 + pif2*pi22
  den1 = (1/sqrt(sig1))*exp(-.5*e1^2/sig1)
  den2 = (1/sqrt(sig2))*exp(-.5*e2^2/sig2)
  denm = pip1*den1 + pip2*den2
  pif1 = pip1*den1/denm; pif2 = pip2*den2/denm
  pif1 = as.numeric(pif1); pif2 = as.numeric(pif2)
  e1 = as.numeric(e1); e2=as.numeric(e2)
  xf = xp + pif1*k1*e1 + pif2*k2*e2
  eye = diag(1, nstate)
  Pf = pif1*(eye-k1%*%M1)%*%Pp + pif2*(eye-k2%*%M2)%*%Pp
  like = like - log(pip1*den1 + pip2*den2)
  prob[i]<-pip2; xfilter[,i]<-xf; innov.sig<-c(sig1,sig2)
  yp[i]<-ifelse(pip1 > pip2, M1%*%xp, M2%*%xp) }
return(like) }
# Estimation
alpha1 = 1.4; alpha2 = -.5; beta0 = .3; sQ1 = .1; sQ2 = .1
init.par = c(alpha1, alpha2, beta0, sQ1, sQ2)
(est = optim(init.par, Linn, NULL, method='BFGS', hessian=TRUE,
            control=list(trace=1, REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
u = cbind(estimate=est$par, SE)
rownames(u)=c('alpha1','alpha2','beta0','sQ1','sQ2'); u
      estimate       SE
alpha1  1.40570967 0.078587727
alpha2 -0.62198715 0.068733109
beta0   0.21049042 0.024625302
sQ1     0.02310306 0.001635291
sQ2     0.11217287 0.016684663
# Graphics
predepi = ifelse(prob<.5,0,1); k = 6:length(y)
Time = time(flu)[k]
regime = predepi[k]+1
par(mfrow=c(3,1), mar=c(2,3,1,1)+.1)
plot(Time, y[k], type="n", ylab="")
grid(lty=2); lines(Time, y[k], col=gray(.7))
text(Time, y[k], col=regime, labels=regime, cex=1.1)
text(1979,.95,"(a)")
plot(Time, xfilter[1,,k], type="n", ylim=c(-.1,.4), ylab="")
grid(lty=2); lines(Time, xfilter[1,,k])
lines(Time, xfilter[3,,k]); lines(Time, xfilter[4,,k])
text(1979,.35,"(b)")
plot(Time, y[k], type="n", ylim=c(.1,.9), ylab="")
grid(lty=2); points(Time, y[k], pch=19)
prde1 = 2*sqrt(innov.sig[1]); prde2 = 2*sqrt(innov.sig[2])
prde = ifelse(predepi[k]<.5, prde1,prde2)

```

```

xx = c(Time, rev(Time))
yy = c(yp[k]-prde, rev(yp[k]+prde))
polygon(xx, yy, border=8, col=gray(.6, alpha=.3))
text(1979, .85, "(c)")

```

6.11 Stochastic Volatility

Stochastic volatility (SV) models are an alternative to GARCH-type models that were presented in [Chapter 5](#). Throughout this section, we let r_t denote the returns of some financial asset. Most models for return data used in practice are of a multiplicative form that we have seen in [Section 5.3](#),

$$r_t = \sigma_t \varepsilon_t, \quad (6.196)$$

where ε_t is an iid sequence and the *volatility process*, σ_t , is a non-negative stochastic process such that ε_t is independent of σ_s for all $s \leq t$. It is often assumed that ε_t has zero mean and unit variance.

In SV models, the volatility is a nonlinear transform of a hidden linear autoregressive process where the hidden volatility process, $x_t = \log \sigma_t^2$, follows a first order autoregression,

$$x_t = \phi x_{t-1} + w_t, \quad (6.197a)$$

$$r_t = \beta \exp(x_t/2) \varepsilon_t, \quad (6.197b)$$

where $w_t \sim \text{iid } N(0, \sigma_w^2)$ and ε_t is iid noise having finite moments. The error processes w_t and ε_t are assumed to be mutually independent and $|\phi| < 1$. As w_t is normally distributed, x_t is also normally distributed. All moments of ε_t exist, so that all moments of r_t in (6.197) exist as well. Assuming that $x_0 \sim N(0, \sigma_w^2/(1 - \phi^2))$ [the stationary distribution] the kurtosis^{6.6} of r_t is given by

$$\kappa_4(r_t) = \kappa_4(\varepsilon_t) \exp(\sigma_x^2), \quad (6.198)$$

where $\sigma_x^2 = \sigma_w^2/(1 - \phi^2)$ is the (stationary) variance of x_t . Thus $\kappa_4(r_t) > \kappa_4(\varepsilon_t)$, so that if $\varepsilon_t \sim \text{iid } N(0, 1)$, the distribution of r_t is leptokurtic. The autocorrelation function of $\{r_t^{2m}; t = 1, 2, \dots\}$ for any integer m is given by (see [Problem 6.29](#))

$$\text{corr}(r_{t+h}^{2m}, r_t^{2m}) = \frac{\exp(m^2 \sigma_x^2 \phi^h) - 1}{\kappa_{4m}(\varepsilon_t) \exp(m^2 \sigma_x^2) - 1}. \quad (6.199)$$

The decay rate of the autocorrelation function is faster than exponential at small time lags and then stabilizes to ϕ for large lags.

Sometimes it is easier to work with the linear form of the model where we define

$$y_t = \log r_t^2 \quad \text{and} \quad v_t = \log \varepsilon_t^2,$$

^{6.6} For an integer m and a random variable U , $\kappa_m(U) := E[|U|^m]/(E[|U|^2])^{m/2}$. Typically, κ_3 is called *skewness* and κ_4 is called *kurtosis*.

in which case we may write

$$y_t = \alpha + x_t + v_t. \quad (6.200)$$

A constant is usually needed in either the state equation or the observation equation (but not typically both), so we write the state equation as

$$x_t = \phi_0 + \phi_1 x_{t-1} + w_t, \quad (6.201)$$

where w_t is white Gaussian noise with variance σ_w^2 . The constant ϕ_0 is sometimes referred to as the *leverage effect*. Together, (6.200) and (6.201) make up the stochastic volatility model due to Taylor (1982).

If ε_t^2 had a log-normal distribution, (6.200)–(6.201) would form a Gaussian state-space model, and we could then use standard DLM results to fit the model to data. Unfortunately, that assumption does not seem to work well. Instead, one often keeps the ARCH normality assumption on $\varepsilon_t \sim \text{iid } N(0, 1)$, in which case, $v_t = \log \varepsilon_t^2$ is distributed as the log of a chi-squared random variable with one degree of freedom. This density is given by

$$f(v) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(e^v - v)\right\} \quad -\infty < v < \infty. \quad (6.202)$$

The mean of the distribution is $-(\gamma + \log 2)$, where $\gamma \approx 0.5772$ is Euler's constant, and the variance of the distribution is $\pi^2/2$. It is a highly skewed density (see Figure 6.18) but it is not flexible because there are no free parameters to be estimated.

Various approaches to the fitting of stochastic volatility models have been examined; these methods include a wide range of assumptions on the observational noise process. A good summary of the proposed techniques, both Bayesian (via MCMC) and non-Bayesian approaches (such as quasi-maximum likelihood estimation and the EM algorithm), can be found in Jacquier et al. (1994), and Shephard (1996). Simulation methods for classical inference applied to stochastic volatility models are discussed in Danielson (1994) and Sandmann and Koopman (1998).

Kim, Shephard and Chib (1998) proposed modeling the log of a chi-squared random variable by a mixture of seven normals to approximate the first four moments of the observational error distribution; the mixture is fixed and no additional model parameters are added by using this technique. The basic model assumption that ε_t is Gaussian is unrealistic for most applications. In an effort to keep matters simple but more general (in that we allow the observational error dynamics to depend on parameters that will be fitted), our method of fitting stochastic volatility models is to retain the Gaussian state equation (6.201), but to write the observation equation, as

$$y_t = \alpha + x_t + \eta_t, \quad (6.203)$$

where η_t is white noise, whose distribution is a mixture of two normals, one centered at zero. In particular, we write

$$\eta_t = I_t z_{t0} + (1 - I_t) z_{t1}, \quad (6.204)$$

where I_t is an iid Bernoulli process, $\Pr\{I_t = 0\} = \pi_0$, $\Pr\{I_t = 1\} = \pi_1$ ($\pi_0 + \pi_1 = 1$), $z_{t0} \sim \text{iid } N(0, \sigma_0^2)$, and $z_{t1} \sim \text{iid } N(\mu_1, \sigma_1^2)$.

The advantage to this model is that it is easy to fit because it uses normality. In fact, the model equations (6.201) and (6.203)-(6.204) are similar to those presented in Peña and Guttman (1988), who used the idea to obtain a robust Kalman filter, and, as previously mentioned, in Kim et al. (1998). The material presented in Section 6.10 applies here, and in particular, the filtering equations for this model are

$$x_{t+1}^t = \phi_0 + \phi_1 x_t^{t-1} + \sum_{j=0}^1 \pi_{tj} K_{tj} \epsilon_{tj}, \quad (6.205)$$

$$P_{t+1}^t = \phi_1^2 P_t^{t-1} + \sigma_w^2 - \sum_{j=0}^1 \pi_{tj} K_{tj}^2 \Sigma_{tj}, \quad (6.206)$$

$$\epsilon_{t0} = y_t - \alpha - x_t^{t-1}, \quad \epsilon_{t1} = y_t - \alpha - x_t^{t-1} - \mu_1, \quad (6.207)$$

$$\Sigma_{t0} = P_t^{t-1} + \sigma_0^2, \quad \Sigma_{t1} = P_t^{t-1} + \sigma_1^2, \quad (6.208)$$

$$K_{t0} = \phi_1 P_t^{t-1} / \Sigma_{t0}, \quad K_{t1} = \phi_1 P_t^{t-1} / \Sigma_{t1}. \quad (6.209)$$

To complete the filtering, we must be able to assess the probabilities $\pi_{t1} = \Pr(I_t = 1 | y_{1:t})$, for $t = 1, \dots, n$; of course, $\pi_{t0} = 1 - \pi_{t1}$. Let $p_j(t | t-1)$ denote the conditional density of y_t given the past $y_{1:t-1}$, and $I_t = j$ for $j = 0, 1$. Then,

$$\pi_{t1} = \frac{\pi_1 p_1(t | t-1)}{\pi_0 p_0(t | t-1) + \pi_1 p_1(t | t-1)}, \quad (6.210)$$

where we assume the distribution π_j , for $j = 0, 1$ has been specified *a priori*. If the investigator has no reason to prefer one state over another the choice of uniform priors, $\pi_1 = 1/2$, will suffice. Unfortunately, it is computationally difficult to obtain the exact values of $p_j(t | t-1)$; although we can give an explicit expression of $p_j(t | t-1)$, the actual computation of the conditional density is prohibitive. A viable approximation, however, is to choose $p_j(t | t-1)$ to be the normal density, $N(x_t^{t-1} + \mu_j, \Sigma_{tj})$, for $j = 0, 1$ and $\mu_0 = 0$; see Section 6.10 for details.

The innovations filter given in (6.205)–(6.210) can be derived from the Kalman filter by a simple conditioning argument; e.g., to derive (6.205), write

$$\begin{aligned} E(x_{t+1} | y_{1:t}) &= \sum_{j=0}^1 E(x_{t+1} | y_{1:t}, I_t = j) \Pr(I_t = j | y_{1:t}) \\ &= \sum_{j=0}^1 \left(\phi_0 + \phi_1 x_t^{t-1} + K_{tj} \epsilon_{tj} \right) \pi_{tj} \\ &= \phi_0 + \phi_1 x_t^{t-1} + \sum_{j=0}^1 \pi_{tj} K_{tj} \epsilon_{tj}. \end{aligned}$$

Estimation of the parameters, $\Theta = (\phi_0, \phi_1, \sigma_0^2, \mu_1, \sigma_1^2, \sigma_w^2)'$, is accomplished via MLE based on the likelihood given by

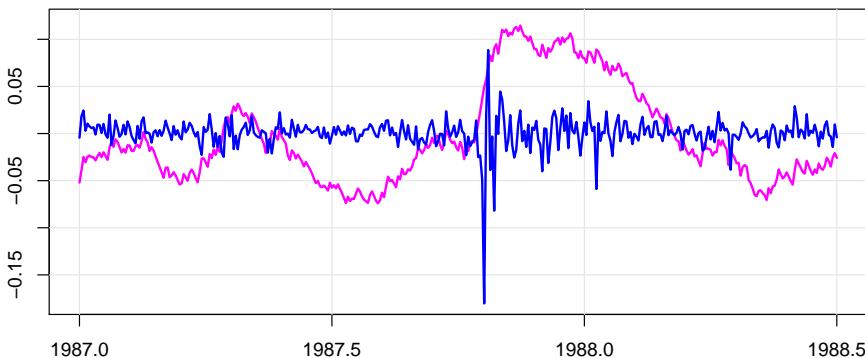


Fig. 6.17. Approximately four hundred observations of r_t , the daily returns of the NYSE surrounding the crash of October 19, 1987. Also displayed is the corresponding one-step-ahead predicted log volatility, \hat{x}_t^{t-1} where $x_t = \log \sigma_t^2$, scaled by .1 to fit on the plot.

$$\ln L_Y(\Theta) = \sum_{t=1}^n \ln \left(\sum_{j=0}^1 \pi_j f_j(t \mid t-1) \right), \quad (6.211)$$

where the density $p_j(t \mid t-1)$ is approximated by the normal density, $N(x_t^{t-1} + \mu_j, \sigma_j^2)$, previously mentioned. We may consider maximizing (6.211) directly as a function of the parameters Θ using a Newton method, or we may consider applying the EM algorithm to the complete data likelihood.

Example 6.23 Analysis of the New York Stock Exchange Returns

Figure 6.17 shows the returns, r_t , for about 400 of the 2000 trading days of the NYSE. Model (6.201) and (6.203)–(6.204), with π_1 fixed at .5, was fit to the data using a quasi-Newton–Raphson method to maximize (6.211). The results are given in Table 6.4. Figure 6.18 compares the density of the log of a χ_1^2 with the fitted normal mixture; we note the data indicate a substantial amount of probability in the upper tail that the $\log\chi_1^2$ distribution misses.

Finally, Figure 6.17 also displays the one-step-ahead predicted log volatility, \hat{x}_t^{t-1} where $x_t = \log \sigma_t^2$, surrounding the crash of October 19, 1987. The analysis indicates that ϕ_0 is not needed. The R code when ϕ_0 is included in the model is as follows.

```

y      = log(nyse^2)
num   = length(y)
# Initial Parameters
phi0 = 0; phi1 = .95; sQ  = .2; alpha = mean(y)
sR0  = 1; mu1  = -3; sR1 = 2
init.par = c(phi0, phi1, sQ, alpha, sR0, mu1, sR1)
# Innovations Likelihood
Linn = function(para){
  phi0 = para[1]; phi1 = para[2]; sQ  = para[3]; alpha = para[4]
  sR0  = para[5]; mu1  = para[6]; sR1 = para[7]
  sv = SVfilter(num, y, phi0, phi1, sQ, alpha, sR0, mu1, sR1)
}

```

Table 6.4. Estimation Results for the NYSE Fit

Parameter	Estimate	Standard Error	Estimated
ϕ_0	-.006	.016 [†]	
ϕ_1	.988	.007	
σ_w	.091	.027	
α	-9.613	1.269	
σ_0	1.220	.065	
μ_1	-2.292	.205	
σ_1	2.683	.105	

[†] not significant

```

    return(sv$like)    }
# Estimation
est = optim(init.par, Linn, NULL, method='BFGS', hessian=TRUE,
            control=list(trace=1,REPORT=1)))
SE = sqrt(diag(solve(est$hessian)))
u = cbind(estimate=est$par, SE)
rownames(u)=c('phi0','phi1','sQ','alpha','sigv0','mu1','sigv1'); u
# Graphics (need filters at the estimated parameters)
phi0 = est$par[1]; phi1 = est$par[2]; sQ = est$par[3]; alpha = est$par[4]
sR0 = est$par[5]; mu1 = est$par[6]; sR1 = est$par[7]
sv = SVfilter(num,y,phi0,phi1,sQ,alpha,sR0,mu1,sR1)
# densities plot (f is chi-sq, fm is fitted mixture)
x = seq(-15,6,by=.01)
f = exp(-.5*(exp(x)-x))/(sqrt(2*pi))
f0 = exp(-.5*(x^2)/sR0^2)/(sR0*sqrt(2*pi))
f1 = exp(-.5*(x-mu1)^2/sR1^2)/(sR1*sqrt(2*pi))
fm = (f0+f1)/2
plot(x, f, type='l'); lines(x, fm, lty=2, lwd=2)
dev.new(); Time=701:1100
plot (Time, nyse[Time], type='l', col=4, lwd=2, ylab='', xlab='',
      ylim=c(-.18,.12))
lines(Time, sv$xp[Time]/10, lwd=2, col=6)

```

It is possible to use the bootstrap procedure described in Section 6.7 for the stochastic volatility model, with some minor changes. The following procedure was described in Stoffer and Wall (2004). We develop a vector first-order equation, as was done in (6.123). First, using (6.207), and noting that $y_t = \pi_{t0}y_t + \pi_{t1}y_t$, we may write

$$y_t = \alpha + x_t^{t-1} + \pi_{t0}\epsilon_{t0} + \pi_{t1}(\epsilon_{t1} + \mu_1). \quad (6.212)$$

Consider the standardized innovations

$$e_{tj} = \Sigma_{tj}^{-1/2} \epsilon_{tj}, \quad j = 0, 1, \quad (6.213)$$

and define the 2×1 vector

$$e_t = \begin{bmatrix} e_{t0} \\ e_{t1} \end{bmatrix}.$$

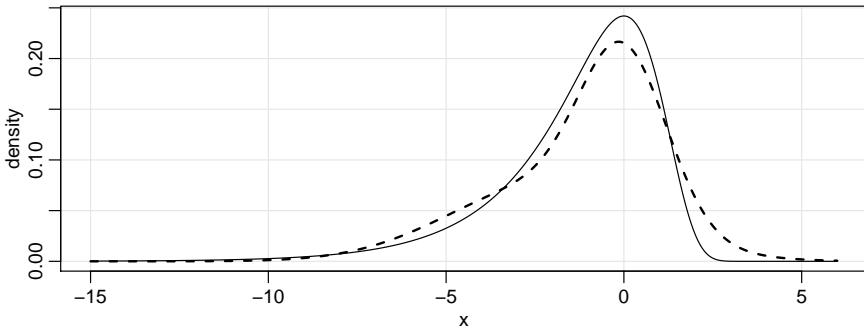


Fig. 6.18. Density of the log of a χ^2_1 as given by (6.202) (solid line) and the fitted normal mixture (dashed line) from Example 6.23.

Also, define the 2×1 vector

$$\xi_t = \begin{bmatrix} x_{t+1}^t \\ y_t \end{bmatrix}.$$

Combining (6.205) and (6.212) results in a vector first-order equation for ξ_t given by

$$\xi_t = F\xi_{t-1} + G_t + H_t e_t, \quad (6.214)$$

where

$$F = \begin{bmatrix} \phi_1 & 0 \\ 1 & 0 \end{bmatrix}, \quad G_t = \begin{bmatrix} \phi_0 \\ \alpha + \pi_{t1}\mu_1 \end{bmatrix}, \quad H_t = \begin{bmatrix} \pi_{t0}K_{t0}\Sigma_{t0}^{1/2} & \pi_{t1}K_{t1}\Sigma_{t1}^{1/2} \\ \pi_{t0}\Sigma_{t0}^{1/2} & \pi_{t1}\Sigma_{t1}^{1/2} \end{bmatrix}.$$

Hence, the steps in bootstrapping for this case are the same as steps (i) through (v) described in Section 6.7, but with (6.123) replaced by the following first-order equation:

$$\xi_t^* = F(\hat{\Theta})\xi_{t-1}^* + G_t(\hat{\Theta}; \hat{\pi}_{t1}) + H_t(\hat{\Theta}; \hat{\pi}_{t1})e_t^*, \quad (6.215)$$

where $\hat{\Theta} = \{\hat{\phi}_0, \hat{\phi}_1, \hat{\sigma}_0^2, \hat{\alpha}, \hat{\mu}_1, \hat{\sigma}_1^2, \hat{\sigma}_w^2\}$ is the MLE of Θ , and $\hat{\pi}_{t1}$ is estimated via (6.210), replacing $p_1(t | t-1)$ and $p_0(t | t-1)$ by their respective estimated normal densities ($\hat{\pi}_{t0} = 1 - \hat{\pi}_{t1}$).

Example 6.24 Analysis of the U.S. GNP Growth Rate

In Example 5.4, we fit an ARCH model to the U.S. GNP growth rate. In this example, we will fit a stochastic volatility model to the residuals from the AR(1) fit on the growth rate (see Example 3.39). Figure 6.19 shows the log of the squared residuals, say y_t , from the fit on the U.S. GNP series. The stochastic volatility model (6.200)–(6.204) was then fit to y_t . Table 6.5 shows the MLEs of the model parameters along with their asymptotic SEs assuming the model is correct. Also displayed in Table 6.5 are the SEs of $B = 500$ bootstrapped samples. There is little agreement between most of the asymptotic values and the bootstrapped values. The interest here, however, is not so much in the SEs, but in the actual sampling distribution

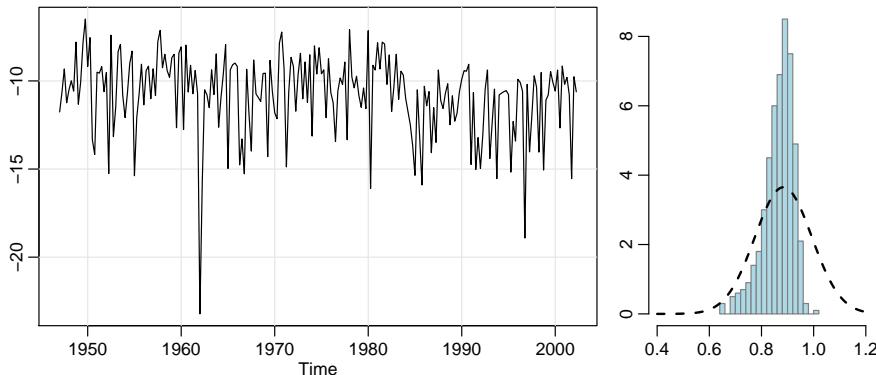


Fig. 6.19. Results for Example 6.24: Log of the squared residuals from an AR(1) fit on GNP growth rate. Bootstrap histogram and asymptotic distribution of $\hat{\phi}_1$.

of the estimates. For example, Figure 6.19 compares the bootstrap histogram and asymptotic normal distribution of $\hat{\phi}_1$. In this case, the bootstrap distribution exhibits positive kurtosis and skewness which is missed by the assumption of asymptotic normality.

The R code for this example is as follows. We held ϕ_0 at 0 for this analysis because it was not significantly different from 0 in an initial analysis.

```

n.boot = 500 # number of bootstrap replicates
tol = sqrt(.Machine$double.eps) # convergence tolerance
gnpgr = diff(log(gnp))
fit = arima(gnpgr, order=c(1,0,0))
y = as.matrix(log(resid(fit)^2))
num = length(y)
plot.ts(y, ylab='')
# Initial Parameters
phi1 = .9; sQ = .5; alpha = mean(y); sR0 = 1; mu1 = -3; sR1 = 2.5
init.par = c(phi1, sQ, alpha, sR0, mu1, sR1)
# Innovations Likelihood
Linn = function(para, y.data){
  phi1 = para[1]; sQ = para[2]; alpha = para[3]
  sR0 = para[4]; mu1 = para[5]; sR1 = para[6]
  sv = SVfilter(num, y.data, 0, phi1, sQ, alpha, sR0, mu1, sR1)
  return(sv$like)}
# Estimation
est = optim(init.par, Linn, NULL, y.data=y, method='BFGS', hessian=TRUE,
            control=list(trace=1,REPORT=1))
SE = sqrt(diag(solve(est$hessian)))
u = rbind(estimate=est$par, SE)
colnames(u)=c('phi1','sQ','alpha','sig0','mu1','sig1')
phi1      sQ   alpha   sig0   mu1   sig1
estimates 0.884 0.381 -9.654 0.835 -2.350 2.453
SE        0.109 0.221  0.343 0.204  0.495 0.293
# Bootstrap
para.star = matrix(0, n.boot, 6) # to store parameter estimates
for (jb in 1:n.boot){

```

Table 6.5. Estimates and Standard Errors for GNP Example

Parameter	MLE	Asymptotic SE	Bootstrap† SE
ϕ_1	0.884	0.109	0.057
σ_w	0.381	0.221	0.324
α	-9.654	0.343	1.529
σ_0	0.835	0.204	0.527
μ_1	-2.350	0.495	0.410
σ_1	2.453	0.293	0.375

† Based on 500 bootstrapped samples.

```

cat('iteration:', jb, '\n')
phi1 = est$par[1]; sQ = est$par[2]; alpha = est$par[3]
sR0 = est$par[4]; mu1 = est$par[5]; sR1 = est$par[6]
Q = sQ^2; R0 = sR0^2; R1 = sR1^2
sv = SVfilter(num, y, 0, phi1, sQ, alpha, sR0, mu1, sR1)
sig0 = sv$Pp+R0; sig1 = sv$Pp+R1;
K0 = sv$Pp/sig0; K1 = sv$Pp/sig1
inn0 = y-sv$xp-alpha; inn1 = y-sv$xp-mu1-alpha
den1 = (1/sqrt(sig1))*exp(-.5*inn1^2/sig1)
den0 = (1/sqrt(sig0))*exp(-.5*inn0^2/sig0)
fpi1 = den1/(den0+den1)
# start resampling at t=4
e0 = inn0/sqrt(sig0); e1 = inn1/sqrt(sig1)
indx = sample(4:num, replace=TRUE)
sinn = cbind(c(e0[1:3], e0[indx]), c(e1[1:3], e1[indx]))
eF = matrix(c(phi1, 1, 0, 0), 2, 2)
xi = cbind(sv$xp,y) # initialize
for (i in 4:num){ # generate boot sample
  G = matrix(c(0, alpha+fpi1[i]*mu1), 2, 1)
  h21 = (1-fpi1[i])*sqrt(sig0[i]); h11 = h21*K0[i]
  h22 = fpi1[i]*sqrt(sig1[i]); h12 = h22*K1[i]
  H = matrix(c(h11,h21,h12,h22),2,2)
  xi[i,] = t(eF%*%as.matrix(xi[i-1,],2) + G + H%*%as.matrix(sinn[i,],2))}
# Estimates from boot data
y.star = xi[,2]
phi1=.9; sQ=.5; alpha=mean(y.star); sR0=1; mu1=-3; sR1=2.5
init.par = c(phi1, sQ, alpha, sR0, mu1, sR1) # same as for data
est.star = optim(init.par, Linn, NULL, y.data=y.star, method='BFGS',
                 control=list(reltol=tol))
para.star[jb,] = cbind(est.star$par[1], abs(est.star$par[2]),
                       est.star$par[3], abs(est.star$par[4]), est.star$par[5],
                       abs(est.star$par[6])) }

# Some summary statistics and graphics
rmse = rep(NA,6) # SEs from the bootstrap
for(i in 1:6){
  rmse[i] = sqrt(sum((para.star[,i]-est$par[i])^2)/n.boot)
  cat(i, rmse[i], '\n') }
dev.new(); phi = para.star[,1]
hist(phi, 15, prob=TRUE, main='', xlim=c(.4,1.2), xlab='')
xx = seq(.4, 1.2, by=.01)
lines(xx, dnorm(xx, mean=u[1,1], sd=u[2,1]), lty='dashed', lwd=2)

```

6.12 Bayesian Analysis of State Space Models

We now consider some Bayesian approaches to fitting linear Gaussian state space models via Markov chain Monte Carlo (MCMC) methods. We assume that the model is given by (6.1)–(6.2); inputs are allowed in the model, but we do not display them for the sake of brevity. In this case, Frühwirth-Schnatter (1994) and Carter and Kohn (1994) established the MCMC procedure that we will discuss here. A comprehensive text that we highly recommend for this case is Petris et al. (2009) and the corresponding R package `dlm`. For nonlinear and non-Gaussian models, the reader is referred to Douc, Moulines, & Stoffer (2014). As in previous sections, we have n observations denoted by $y_{1:n} = \{y_1, \dots, y_n\}$, whereas the states are denoted as $x_{0:n} = \{x_0, x_1, \dots, x_n\}$, with x_0 being the initial state.

MCMC methods refer to Monte Carlo integration methods that use a Markovian updating scheme to sample from intractable posterior distributions. The most common MCMC method is the Gibbs sampler, which is essentially a modification of the Metropolis algorithm (Metropolis et al., 1953) developed by Hastings (1970) in the statistical setting and by Geman and Geman (1984) in the context of image restoration. Later, Tanner and Wong (1987) used the ideas in their substitution sampling approach, and Gelfand and Smith (1990) developed the Gibbs sampler for a wide class of parametric models. The basic strategy is to use conditional distributions to set up a Markov chain to obtain samples from a joint distribution. The following simple case demonstrates this idea.

Example 6.25 Gibbs Sampling for the Bivariate Normal

Suppose we wish to obtain samples from a bivariate normal distribution,

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right],$$

where $|\rho| < 1$, but we can only generate samples from a univariate normal.

- The univariate conditionals are [see (B.9)–(B.10)]

$$(X \mid Y = y) \sim N(\rho y, 1 - \rho^2) \quad \text{and} \quad (Y \mid X = x) \sim N(\rho x, 1 - \rho^2),$$

and we can simulate from these distributions.

- Construct a Markov chain: Pick $X^{(0)} = x_0$, and then iterate the process $X^{(0)} = x_0 \mapsto Y^{(0)} \mapsto X^{(1)} \mapsto Y^{(1)} \mapsto \dots \mapsto X^{(k)} \mapsto Y^{(k)} \mapsto \dots$, where

$$\begin{aligned} (Y^{(k)} \mid X^{(k)} = x_k) &\sim N(\rho x_k, 1 - \rho^2) \\ (X^{(k)} \mid Y^{(k-1)} = y_{k-1}) &\sim N(\rho y_{k-1}, 1 - \rho^2). \end{aligned}$$

- The joint distribution of $(X^{(k)}, Y^{(k)})$ is (see Problem 3.2)

$$\begin{pmatrix} X^{(k)} \\ Y^{(k)} \end{pmatrix} \sim N \left[\begin{pmatrix} \rho^{2k} x_0 \\ \rho^{2k+1} x_0 \end{pmatrix}, \begin{pmatrix} 1 - \rho^{4k} & \rho(1 - \rho^{4k}) \\ \rho(1 - \rho^{4k}) & 1 - \rho^{4k+2} \end{pmatrix} \right].$$

- Thus, for any starting value, x_0 , $(X^{(k)}, Y^{(k)}) \rightarrow_d (X, Y)$ as $k \rightarrow \infty$; the speed depends on ρ . Then one would run the chain and throw away the initial n_0 sampled values (burnin) and retain the rest.

For state space models, the main objective is to obtain the posterior density of the parameters $p(\Theta | y_{1:n})$ or $p(x_{0:n} | y_{1:n})$ if the states are meaningful. For example, the states do not have any meaning for an ARMA model, but they are important for a stochastic volatility model. It is generally easier to get samples from the full posterior $p(\Theta, x_{0:n} | y_{1:n})$ and then marginalize (“average”) to obtain $p(\Theta | y_{1:n})$ or $p(x_{0:n} | y_{1:n})$. As previously mentioned, the most popular method is to run a full Gibbs sampler, alternating between sampling model parameters and latent state sequences from their respective full conditional distributions.

Procedure 6.1 Gibbs Sampler for State Space Models

- (i) Draw $\Theta' \sim p(\Theta | x_{0:n}, y_{1:n})$
- (ii) Draw $x'_{0:n} \sim p(x_{0:n} | \Theta', y_{1:n})$

Procedure 6.1-(i) is generally much easier because it conditions on the complete data $\{x_{0:n}, y_{1:n}\}$, which we saw in [Section 6.3](#) can simplify the problem. **Procedure 6.1-(ii)** amounts to sampling from the joint smoothing distribution of the latent state sequence and is generally difficult. For linear Gaussian models, however, both parts of **Procedure 6.1** are relatively easy to perform.

To accomplish **Procedure 6.1-(i)**, note that

$$p(\Theta | x_{0:n}, y_{1:n}) \propto \pi(\Theta) p(x_0 | \Theta) \prod_{t=1}^n p(x_t | x_{t-1}, \Theta) p(y_t | x_t, \Theta) \quad (6.216)$$

where $\pi(\Theta)$ is the prior on the parameters. The prior often depends on “hyperparameters” that add another level to the hierarchy. For simplicity, these hyperparameters are assumed to be known. The parameters are typically conditionally independent with distributions from standard parametric families (at least as long as the prior distribution is conjugate relative to the Bayesian model specification). For non-conjugate models, one option is to replace **Procedure 6.1-(i)** with a Metropolis-Hastings step, which is feasible since the complete data density $p(\Theta, x_{0:n}, y_{1:n})$ can be evaluated pointwise.

For example, in the univariate model

$$x_t = \phi x_{t-1} + w_t \quad \text{and} \quad y_t = x_t + v_t$$

where $w_t \sim \text{iid } N(0, \sigma_w^2)$ independent of $v_t \sim \text{iid } N(0, \sigma_v^2)$, we can use the normal and inverse gamma (IG) distributions for priors. In this case, the priors on the variance components are chosen from a conjugate family, that is, $\sigma_w^2 \sim \text{IG}(a_0/2, b_0/2)$ independent of $\sigma_v^2 \sim \text{IG}(c_0/2, d_0/2)$, where IG denotes the inverse (reciprocal) gamma distribution. Then, for example, if the prior on ϕ is Gaussian, $\phi \sim N(\mu_\phi, \sigma_\phi^2)$, then $\phi | \sigma_w, x_{0:n}, y_{1:n} \sim N(Bb, B)$, where

$$B^{-1} = \frac{1}{\sigma_\phi^2} + \frac{1}{\sigma_w^2} \sum_{t=1}^n x_{t-1}^2, \quad b = \frac{\mu_\phi}{\sigma_\phi^2} + \frac{1}{\sigma_w^2} \sum_{t=1}^n x_t x_{t-1}.$$

and

$$\begin{aligned}\sigma_w^2 | \phi, x_{0:n}, y_{1:n} &\sim \text{IG}\left(\frac{1}{2}(a_0 + n), \frac{1}{2}\{b_0 + \sum_{t=1}^n [x_t - \phi x_{t-1}]^2\}\right); \\ \sigma_v^2 | x_{0:n}, y_{1:n} &\sim \text{IG}\left(\frac{1}{2}(c_0 + n), \frac{1}{2}\{c_0 + \sum_{t=1}^n [y_t - x_t]^2\}\right).\end{aligned}$$

For **Procedure 6.1-(ii)**, the goal is to sample the entire set of state vectors, $x_{0:n}$, from the posterior density $p(x_{0:n} | \Theta, y_{1:n})$, where Θ is a fixed set of parameters obtained from the previous step. We will write the posterior as $p_\Theta(x_{0:n} | y_{1:n})$ to save space. Because of the Markov structure, we can write,

$$p_\Theta(x_{0:n} | y_{1:n}) = p_\Theta(x_n | y_{1:n}) p_\Theta(x_{n-1} | x_n, y_{1:n-1}) \cdots p_\Theta(x_0 | x_1). \quad (6.217)$$

In view of (6.217), it is possible to sample the entire set of state vectors, $x_{0:n}$, by sequentially simulating the individual states backward. This process yields a simulation method that Frühwirth-Schnatter (1994) called the forward-filtering, backward-sampling (FFBS) algorithm. From (6.217), we see that we must obtain the densities

$$p_\Theta(x_t | x_{t+1}, y_{1:t}) \propto p_\Theta(x_t | y_{1:t}) p_\Theta(x_{t+1} | x_t).$$

In particular, we know that $x_t | y_{1:t} \sim N_p^\Theta(x_t^t, P_t^t)$ and $x_{t+1} | x_t \sim N_p^\Theta(\Phi x_t, Q)$. And because the processes are Gaussian, we need only obtain the conditional means and variances, say, $m_t = E_\Theta(x_t | y_{1:t}, x_{t+1})$ and $V_t = \text{var}_\Theta(x_t | y_{1:t}, x_{t+1})$. In particular,

$$m_t = x_t^t + J_t(x_{t+1} - x_{t+1}^t) \quad \text{and} \quad V_t = P_t^t - J_t P_{t+1}^t J_t', \quad (6.218)$$

for $t = n-1, n-2, \dots, 0$, where J_t is defined in (6.47). We note that m_t has already been derived in (6.48). To derive m_t and V_t using standard normal theory, use a strategy similar to the derivation of the filter in **Property 6.1**. That is,

$$\begin{pmatrix} x_t \\ x_{t+1} \end{pmatrix} | y_{1:t} \sim N \left(\begin{bmatrix} x_t^t \\ x_{t+1}^t \end{bmatrix}, \begin{bmatrix} P_t^t & P_t^t \Phi' \\ \Phi P_t^t & P_{t+1}^t \end{bmatrix} \right);$$

now use (B.9), (B.10), and the definition of J_t in (6.47). Also, recall the proof of **Property 6.3** wherein we noted the off-diagonal $P_{t+1,t}^t = \Phi P_t^t$.

Hence, given Θ , the algorithm is to first sample x_n from a $N_p^\Theta(x_n^n, P_n^n)$, where x_n^n and P_n^n are obtained from the Kalman filter, **Property 6.1**, and then sample x_t from a $N_p^\Theta(m_t, V_t)$, for $t = n-1, n-2, \dots, 0$, where the conditioning value of x_{t+1} is the value previously sampled.

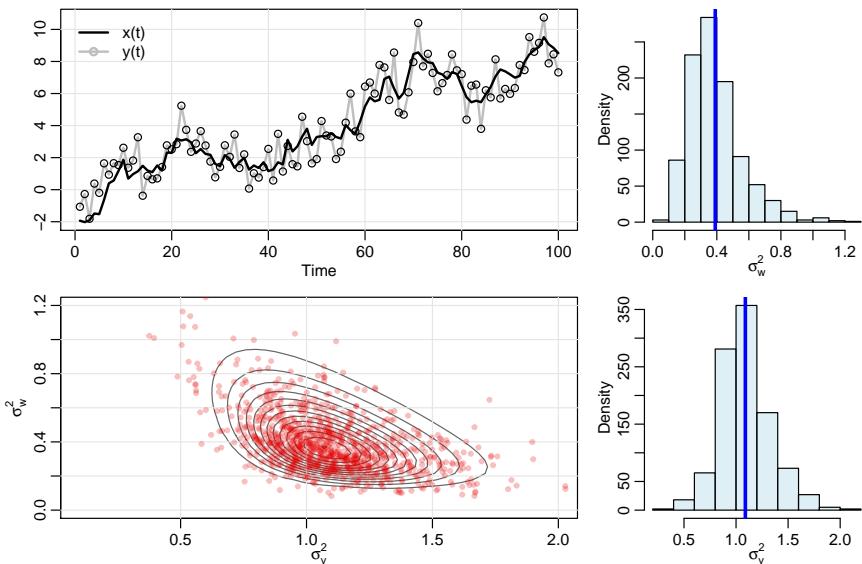


Fig. 6.20. Display for [Example 6.26](#): Left: Generated states, x_t and data y_t . Contours of the likelihood (solid line) of the data and sampled posterior values as points. Right : Marginal sampled posteriors and posterior means (vertical lines) of each variance component. The true values are $\sigma_w^2 = .5$ and $\sigma_v^2 = 1$.

Example 6.26 Local Level Model

In this example, we consider the local level model previously discussed in [Example 6.4](#). Here, we consider the model

$$y_t = x_t + v_t \quad \text{and} \quad x_t = x_{t-1} + w_t$$

where $v_t \sim \text{iid } N(0, \sigma_v^2 = 1)$ independent of $w_t \sim \text{iid } N(0, \sigma_w^2 = .5)$. This is the univariate model we just discussed, but where $\phi = 1$. In this case, we used IG priors for each of the variance components.

For the prior distributions, all parameters (a_0, b_0, c_0, d_0) were set to .02. We generated 1010 samples, using the first 10 as burn-in. [Figure 6.20](#) displays the simulated data and states, the contours of the likelihood of the data, the sampled posterior values as points, and the marginal sampled posteriors of each variance component along with the posterior means. [Figure 6.21](#) compares the actual smoother x_t^n with the posterior mean of the sampled smoothed values. In addition, a pointwise 95% credible interval is displayed as a filled area.

The following code was used in this example.

```
##-- Notation --##
#           y(t) = x(t) + v(t);    v(t) ~ iid N(0, V)
#           x(t) = x(t-1) + w(t);  w(t) ~ iid N(0, W)
#  priors: x(0) ~ N(m0, C0);  V ~ IG(a, b);  W ~ IG(c, d)
#  FFBS:   x(t|t) ~ N(m, C);  x(t|n) ~ N(mm, CC); x(t|t+1) ~ N(a, R)
##--
```

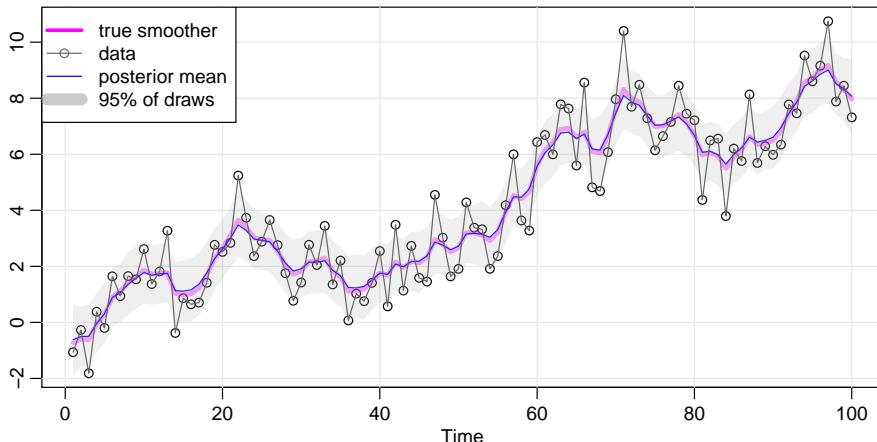


Fig. 6.21. Display for Example 6.26: True smoother, x_t^n , the data y_t , and the posterior mean of the sampled smoother values; the filled in area shows 2.5% to 97.5%-tiles of the draws.

```

ffbs = function(y,V,W,m0,C0){
  n = length(y); a = rep(0,n); R = rep(0,n)
  m = rep(0,n); C = rep(0,n); B = rep(0,n-1)
  H = rep(0,n-1); mm = rep(0,n); CC = rep(0,n)
  x = rep(0,n); llike = 0.0
  for (t in 1:n){
    if(t==1){a[1] = m0; R[1] = C0 + W
    }else{ a[t] = m[t-1]; R[t] = C[t-1] + W }
    f = a[t]
    Q = R[t] + V
    A = R[t]/Q
    m[t] = a[t]+A*(y[t]-f)
    C[t] = R[t]-Q*A^2
    B[t-1] = C[t-1]/R[t]
    H[t-1] = C[t-1]-R[t]*B[t-1]**2
    llike = llike + dnorm(y[t],f,sqrt(Q),log=TRUE) }
    mm[n] = m[n]; CC[n] = C[n]
    x[n] = rnorm(1,m[n],sqrt(C[n]))
    for (t in (n-1):1){
      mm[t] = m[t] + C[t]/R[t+1]*(mm[t+1]-a[t+1])
      CC[t] = C[t] - (C[t]^2)/(R[t+1]^2)*(R[t+1]-CC[t+1])
      x[t] = rnorm(1,m[t]+B[t]*(x[t+1]-a[t+1]),sqrt(H[t])) }
  return(list(x=x,m=m,C=C,mm=mm,CC=CC,llike=llike)) }
# Simulate states and data
set.seed(1); W = 0.5; V = 1.0
n = 100; m0 = 0.0; C0 = 10.0; x0 = 0
w = rnorm(n,0,sqrt(W))
v = rnorm(n,0,sqrt(V))
x = y = rep(0,n)
x[1] = x0 + w[1]
y[1] = x[1] + v[1]
for (t in 2:n){
  x[t] = x[t-1] + w[t]
  y[t] = x[t] + v[t]
}
  
```

```

y[t] = x[t] + v[t] }
# actual smoother (for plotting)
ks = Ksmooth0(num=n, y, A=1, m0, C0, Phi=1, cQ=sqrt(W), cR=sqrt(V))
xsmooth = as.vector(ks$xs)
#
run = ffbs(y,V,W,m0,C0)
m = run$m; C = run$C; mm = run$mm
CC = run$CC; L1 = m-2*C; U1 = m+2*C
L2 = mm-2*CC; U2 = mm+2*CC
N = 50
Vs = seq(0.1,2,length=N)
Ws = seq(0.1,2,length=N)
likes = matrix(0,N,N)
for (i in 1:N){
  for (j in 1:N){
    V = Vs[i]
    W = Ws[j]
    run = ffbs(y,V,W,m0,C0)
    likes[i,j] = run$llike  }  }
# Hyperparameters
a = 0.01; b = 0.01; c = 0.01; d = 0.01
# MCMC step
set.seed(90210)
burn = 10; M = 1000
niter = burn + M
V1 = V; W1 = W
draws = NULL
all_draws = NULL
for (iter in 1:niter){
  run = ffbs(y,V1,W1,m0,C0)
  x = run$x
  V1 = 1/rgamma(1,a+n/2,b+sum((y-x)^2)/2)
  W1 = 1/rgamma(1,c+(n-1)/2,d+sum(diff(x)^2)/2)
  draws = rbind(draws,c(V1,W1,x))  }
all_draws = draws[,1:2]
q025 = function(x){quantile(x,0.025)}
q975 = function(x){quantile(x,0.975)}
draws = draws[(burn+1):(niter),]
xs = draws[,3:(n+2)]
lx = apply(xs,2,q025)
mx = apply(xs,2,mean)
ux = apply(xs,2,q975)
## plot of the data
par(mfrow=c(2,2), mgp=c(1.6,.6,0), mar=c(3,3.2,1,1))
ts.plot(ts(x), ts(y), ylab='', col=c(1,8), lwd=2)
points(y)
legend(0, 11, legend=c("x(t)","y(t)"), lty=1, col=c(1,8), lwd=2, bty="n",
       pch=c(-1,1))
contour(Vs, Ws, exp(likes), xlab=expression(sigma[v]^2),
        ylab=expression(sigma[w]^2), drawlabels=FALSE, ylim=c(0,1.2))
points(draws[,1:2], pch=16, col=rgb(.9,0,0,.3), cex=.7)
hist(draws[,1], ylab="Density",main="", xlab=expression(sigma[v]^2))
abline(v=mean(draws[,1]), col=3, lwd=3)
hist(draws[,2],main="", ylab="Density", xlab=expression(sigma[w]^2))
abline(v=mean(draws[,2]), col=3, lwd=3)
## plot states

```

```

par(mgp=c(1.6,.6,0), mar=c(2,1,.5,0)+.5)
plot(ts(mx), ylab='', type='n', ylim=c(min(y),max(y)))
grid(lty=2); points(y)
lines(xsmooth, lwd=4, col=rgb(1,0,1,alpha=.4))
lines(mx, col=4)
xx=c(1:100, 100:1)
yy=c(lx, rev(ux))
polygon(xx, yy, border=NA, col= gray(.6,alpha=.2))
lines(yy, col=gray(.4))
legend('topleft', c('true smoother', 'data', 'posterior mean', '95% of
                     draws'), lty=1, lwd=c(3,1,1,10), pch=c(-1,1,-1,-1), col=c(6,
                     gray(.4) ,4, gray(.6, alpha=.5)), bg='white' )

```

Next, we consider a more complicated model.

Example 6.27 Structural Model

Consider the Johnson & Johnson quarterly earnings per share series that was discussed in [Example 6.10](#). Recall that the model is

$$y_t = (1 \ 1 \ 0 \ 0) x_t + v_t,$$

$$x_t = \begin{pmatrix} T_t \\ S_t \\ S_{t-1} \\ S_{t-2} \end{pmatrix} = \begin{pmatrix} \phi & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} T_{t-1} \\ S_{t-1} \\ S_{t-2} \\ S_{t-3} \end{pmatrix} + \begin{pmatrix} w_{t1} \\ w_{t2} \\ 0 \\ 0 \end{pmatrix}$$

where $R = \sigma_v^2$ and

$$Q = \begin{pmatrix} \sigma_{w,11}^2 & 0 & 0 & 0 \\ 0 & \sigma_{w,22}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The parameters to be estimated are the transition parameter associated with the growth rate, $\phi > 1$, the observation noise variance, σ_v^2 , and the state noise variances associated with the trend and the seasonal components, $\sigma_{w,11}^2$ and $\sigma_{w,22}^2$, respectively.

In this case, sampling from $p(x_{0:n} | \Theta, y_{1:n})$ follows directly from [\(6.217\)](#)–[\(6.218\)](#). Next, we discuss how to sample from $p(\Theta | x_{0:n}, y_{1:n})$. For the transition parameter, write $\phi = 1 + \beta$, where $0 < \beta \ll 1$; recall that in [Example 6.10](#), ϕ was estimated to be 1.035, which indicated a growth rate, β , of 3.5%. Note that the trend component may be rewritten as

$$\nabla T_t = T_t - T_{t-1} = \beta T_{t-1} + w_{t1}.$$

Consequently, conditional on the states, the parameter β is the slope in the linear regression (through the origin) of ∇T_t on T_{t-1} , for $t = 1, \dots, n$, and w_{t1} is the error. As is typical, we put a Normal–Inverse Gamma (IG) prior on $(\beta, \sigma_{w,11}^2)$, i.e., $\beta | \sigma_{w,11}^2 \sim N(b_0, \sigma_{w,11}^2 B_0)$ and $\sigma_{w,11}^2 \sim IG(n_0/2, n_0 s_0^2/2)$, with known hyperparameters b_0, B_0, n_0, s_0^2 .

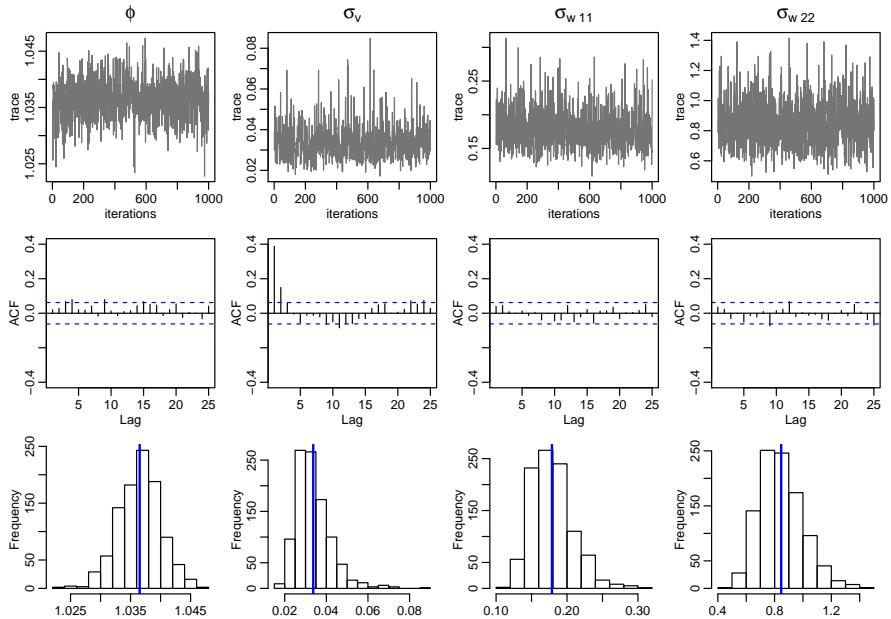


Fig. 6.22. Parameter estimation results for Example 6.27. The top row displays the traces of 1000 draws after burn-in. The middle row displays the ACF of the traces. The sampled posteriors are displayed in the last row (the mean is marked by a solid vertical line).

We also used IG priors for the other two variance components, σ_v^2 and $\sigma_{w,22}^2$. In this case, if the prior $\sigma_v^2 \sim \text{IG}(n_0/2, n_0 s_0^2/2)$, then the posterior is

$$\sigma_v^2 \mid x_{0:n}, y_{1:n} \sim \text{IG}(n_v/2, n_v s_v^2/2),$$

where $n_v = n_0 + n$, and $n_v s_v^2 = n_0 s_0^2 + \sum_{t=1}^n (Y_t - T_t - S_t)^2$. Similarly, if the prior $\sigma_{w,22}^2 \sim \text{IG}(n_0/2, n_0 s_0^2/2)$, then the posterior is

$$\sigma_{w,22}^2 \mid x_{0:n}, y_{1:n} \sim \text{IG}(n_w/2, n_w s_w^2/2),$$

where $n_w = n_0 + (n - 3)$, and $n_w s_w^2 = n_0 s_0^2 + \sum_{t=1}^{n-3} (S_t - S_{t-1} - S_{t-2} - S_{t-3})^2$.

Figure 6.22 displays the results of the posterior estimates of the parameters. The top row of the figure displays the traces of 1000 draws, after a burn-in of 100, with a step size of 10 (i.e., every 10th sampled value is retained). The middle row of the figure displays the ACF of the traces, and the sampled posteriors are displayed in the last row of the figure. The results of this analysis are comparable to the results obtained in Example 6.10; the posterior mean and median for ϕ indicates a 3.7% growth rate in the Johnson & Johnson quarterly earnings over this time period.

Figure 6.23 displays the smoothers of trend (T_t) and season ($T_t + S_t$) along with 99% credible intervals. Again, these results are comparable to the results obtained in Example 6.10. The R code for this example is as follows:

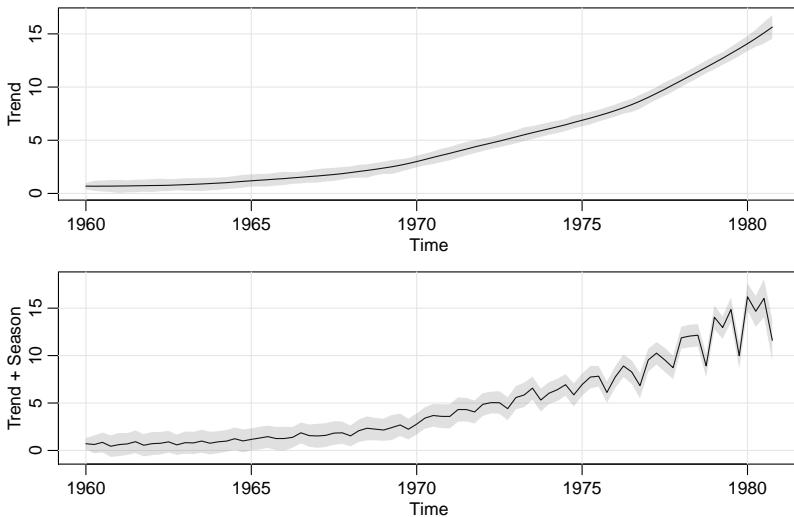


Fig. 6.23. Example 6.27 smoother estimates of trend (T_t) and trend plus season ($T_t + S_t$) along with corresponding 99% credible intervals.

```
library(plyr)    # used to view progress (install it if you don't have it)
y = jj
### setup - model and initial parameters
set.seed(90210)
n = length(y)
F = c(1,1,0,0)      # this is A
G = diag(0,4)        # G is Phi
  G[1,1] = 1.03
  G[2,] = c(0,-1,-1,-1); G[3,]=c(0,1,0,0); G[4,]=c(0,0,1,0)
a1 = rbind(.7,0,0,0) # this is mu0
R1 = diag(.04,4)      # this is Sigma0
V = .1
W11 = .1
W22 = .1
##-- FFBS --#
ffbs = function(y,F,G,V,W11,W22,a1,R1){
  n = length(y)
  Ws = diag(c(W11,W22,1,1)) # this is Q with 1s as a device only
  iW = diag(1/diag(Ws),4)
  a = matrix(0,n,4)          # this is m_t
  R = array(0,c(n,4,4))     # this is V_t
  m = matrix(0,n,4)
  C = array(0,c(n,4,4))
  a[1,] = a1[,1]
  R[1,,] = R1
  f = t(F)%*%a[1,]
  Q = t(F)%*%R[1,,]%*%F + V
  A = R[1,,]%*%F/Q[1,1]
  m[1,] = a[1,]+A%*%(y[1]-f)
  C[1,,] = R[1,,]-A%*%t(A)*Q[1,1]
```

```

for (t in 2:n){
  a[t,] = G%*%m[t-1,]
  R[t,,] = G%*%C[t-1,,]*%t(G) + Ws
  f      = t(F)%*%a[t,]
  Q      = t(F)%*%R[t,,]*%F + V
  A      = R[t,,]*%F/Q[1,1]
  m[t,] = a[t,] + A%*%(y[t]-f)
  C[t,,] = R[t,,] - A%*%t(A)*Q[1,1]      }
  xb    = matrix(0,n,4)
  xb[n,] = m[n,] + t(chol(C[n,,]))%*%rnorm(4)
  for (t in (n-1):1){
    iC = solve(C[t,,])
    CCC = solve(t(G)%*%iW%*%G + iC)
    mmm = CCC%*%(t(G)%*%iW%*%xb[t+1,] + iC%*%m[t,])
    xb[t,] = mmm + t(chol(CCC))%*%rnorm(4)  }
  return(xb)  }

##-- Prior hyperparameters --##
# b0 = 0      # mean for beta = phi -1
# B0 = Inf    # var for beta (non-informative => use OLS for sampling beta)
n0 = 10     # use same for all- the prior is 1/Gamma(n0/2, n0*s20_/2)
s20v = .001  # for V
s20w = .05   # for Ws
##-- MCMC scheme --##
set.seed(90210)
burnin = 100
step   = 10
M      = 1000
niter  = burnin+step*M
pars   = matrix(0,niter,4)
xbs   = array(0,c(niter,n,4))
pr <- progress_text()           # displays progress
pr$init(niter)
for (iter in 1:niter){
  xb = ffbs(y,F,G,V,W11,W22,a1,R1)
  u = xb[,1]
  yu = diff(u); xu = u[-n]      # for phihat and se(phihat)
  regu = lm(yu~0+xu)            # est of beta = phi-1
  phies = as.vector(coef(summary(regu)))[1:2] + c(1,0) # phi estimate and SE
  dft = df.residual(regu)
  G[1,1] = phies[1] + rt(1,dft)*phies[2] # use a t
  V = 1/rgamma(1, (n0+n)/2, (n0*s20v/2) + sum((y-xb[,1]-xb[,2])^2)/2)
  W11 = 1/rgamma(1, (n0+n-1)/2, (n0*s20w/2) +
    sum((xb[-1,1]-phies[1]*xb[-n,1])^2)/2)
  W22 = 1/rgamma(1, (n0+n-3)/2, (n0*s20w/2) + sum((xb[4:n,2] +
    xb[3:(n-1),2]+ xb[2:(n-2),2] +xb[1:(n-3),2])^2)/2)
  xbs[iter,,] = xb
  pars[iter,] = c(G[1,1], sqrt(V), sqrt(W11), sqrt(W22))
  pr$step()          }

# Plot results
ind = seq(burnin+1,niter,by=step)
names= c(expression(phi), expression(sigma[v]), expression(sigma[w-11]),
       expression(sigma[w-22]))
dev.new(height=5)
par(mfcol=c(3,4), mar=c(2,2,.25,0)+.75, mgp=c(1.6,.6,0), oma=c(0,0,1,0))
for (i in 1:4){
  plot.ts(pars[ind,i],xlab="iterations", ylab="trace", main=""")

```

```

mtext(names[i], side=3, line=.5, cex=1)
acf(pars[ind,i],main="", lag.max=25, xlim=c(1,25), ylim=c(-.4,.4))
hist(pars[ind,i],main="", xlab="")
abline(v=mean(pars[ind,i]), lwd=2, col=3) }
par(mfrow=c(2,1), mar=c(2,2,0,0)+.7, mgp=c(1.6,.6,0))
  mxb = cbind(apply(xbs[ind,,1],2,mean), apply(xbs[,,2],2,mean))
  lxb = cbind(apply(xbs[ind,,1],2,quantile,0.005),
               apply(xbs[ind,,2],2,quantile,0.005))
  uxb = cbind(apply(xbs[ind,,1],2,quantile,0.995),
               apply(xbs[ind,,2],2,quantile,0.995))
  mxb = ts(cbind(mxb, rowSums(mxb)), start = tsp(jj)[1], freq=4)
  lxb = ts(cbind(lxb, rowSums(lxb)), start = tsp(jj)[1], freq=4)
  uxb = ts(cbind(uxb, rowSums(uxb)), start = tsp(jj)[1], freq=4)
  names=c('Trend', 'Season', 'Trend + Season')
  L = min(lxb[,1])- .01; U = max(uxb[,1]) + .01
plot(mxb[,1], ylab=names[1], ylim=c(L,U), type='n')
  grid(lty=2); lines(mxb[,1])
  xx=c(time(jj), rev(time(jj)))
  yy=c(lxb[,1], rev(uxb[,1]))
  polygon(xx, yy, border=NA, col=gray(.4, alpha = .2))
  L = min(lxb[,3])- .01; U = max(uxb[,3]) + .01
plot(mxb[,3], ylab=names[3], ylim=c(L,U), type='n')
  grid(lty=2); lines(mxb[,3])
  xx=c(time(jj), rev(time(jj)))
  yy=c(lxb[,3], rev(uxb[,3]))
  polygon(xx, yy, border=NA, col=gray(.4, alpha = .2))

```

Problems

Section 6.1

6.1 Consider a system process given by

$$x_t = -0.9x_{t-2} + w_t \quad t = 1, \dots, n$$

where $x_0 \sim N(0, \sigma_0^2)$, $x_{-1} \sim N(0, \sigma_1^2)$, and w_t is Gaussian white noise with variance σ_w^2 . The system process is observed with noise, say,

$$y_t = x_t + v_t,$$

where v_t is Gaussian white noise with variance σ_v^2 . Further, suppose x_0 , x_{-1} , $\{w_t\}$ and $\{v_t\}$ are independent.

- (a) Write the system and observation equations in the form of a state space model.
- (b) Find the values of σ_0^2 and σ_1^2 that make the observations, y_t , stationary.
- (c) Generate $n = 100$ observations with $\sigma_w = 1$, $\sigma_v = 1$ and using the values of σ_0^2 and σ_1^2 found in (b). Do a time plot of x_t and of y_t and compare the two processes.
Also, compare the sample ACF and PACF of x_t and of y_t .
- (d) Repeat (c), but with $\sigma_v = 10$.

6.2 Consider the state-space model presented in [Example 6.3](#). Let $x_t^{t-1} = E(x_t \mid y_{t-1}, \dots, y_1)$ and let $P_t^{t-1} = E(x_t - x_t^{t-1})^2$. The innovation sequence or residuals are $\epsilon_t = y_t - y_t^{t-1}$, where $y_t^{t-1} = E(y_t \mid y_{t-1}, \dots, y_1)$. Find $\text{cov}(\epsilon_s, \epsilon_t)$ in terms of x_t^{t-1} and P_t^{t-1} for (i) $s \neq t$ and (ii) $s = t$.

Section 6.2

6.3 Simulate $n = 100$ observations from the following state-space model:

$$x_t = .8x_{t-1} + w_t \quad \text{and} \quad y_t = x_t + v_t$$

where $x_0 \sim N(0, 2.78)$, $w_t \sim \text{iid } N(0, 1)$, and $v_t \sim \text{iid } N(0, 1)$ are all mutually independent. Compute and plot the data, y_t , the one-step-ahead predictors, y_t^{t-1} along with the root mean square prediction errors, $E^{1/2}(y_t - y_t^{t-1})^2$ using [Example 6.5](#) as a guide.

6.4 Suppose the vector $z = (x', y')'$, where $x (p \times 1)$ and $y (q \times 1)$ are jointly distributed with mean vectors μ_x and μ_y and with covariance matrix

$$\text{cov}(z) = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}.$$

Consider projecting x on $\mathcal{M} = \overline{\text{sp}}\{1, y\}$, say, $\hat{x} = b + By$.

(a) Show the orthogonality conditions can be written as

$$E(x - b - By) = 0,$$

$$E[(x - b - By)y'] = 0,$$

leading to the solutions

$$b = \mu_x - B\mu_y \quad \text{and} \quad B = \Sigma_{xy}\Sigma_{yy}^{-1}.$$

(b) Prove the mean square error matrix is

$$MSE = E[(x - b - By)x'] = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}.$$

(c) How can these results be used to justify the claim that, in the absence of normality, [Property 6.1](#) yields the best linear estimate of the state x_t given the data Y_t , namely, x_t^t , and its corresponding MSE, namely, P_t^t ?

6.5 *Projection Theorem Derivation of Property 6.2.* Throughout this problem, we use the notation of [Property 6.2](#) and of the Projection Theorem given in [Appendix B](#), where \mathcal{H} is L^2 . If $\mathcal{L}_{k+1} = \overline{\text{sp}}\{y_1, \dots, y_{k+1}\}$, and $\mathcal{V}_{k+1} = \overline{\text{sp}}\{y_{k+1} - y_{k+1}^k\}$, for $k = 0, 1, \dots, n-1$, where y_{k+1}^k is the projection of y_{k+1} on \mathcal{L}_k , then, $\mathcal{L}_{k+1} = \mathcal{L}_k \oplus \mathcal{V}_{k+1}$. We assume $P_0^0 > 0$ and $R > 0$.

- (a) Show the projection of x_k on \mathcal{L}_{k+1} , that is, x_k^{k+1} , is given by

$$x_k^{k+1} = x_k^k + H_{k+1}(y_{k+1} - y_{k+1}^k),$$

where H_{k+1} can be determined by the orthogonality property

$$\mathbb{E} \left\{ \left(x_k - H_{k+1}(y_{k+1} - y_{k+1}^k) \right) \left(y_{k+1} - y_{k+1}^k \right)' \right\} = 0.$$

Show

$$H_{k+1} = P_k^k \Phi' A'_{k+1} [A_{k+1} P_{k+1}^k A'_{k+1} + R]^{-1}.$$

- (b) Define $J_k = P_k^k \Phi' [P_{k+1}^k]^{-1}$, and show

$$x_k^{k+1} = x_k^k + J_k(x_{k+1}^{k+1} - x_{k+1}^k).$$

- (c) Repeating the process, show

$$x_k^{k+2} = x_k^k + J_k(x_{k+1}^{k+2} - x_{k+1}^k) + H_{k+2}(y_{k+2} - y_{k+2}^{k+1}),$$

solving for H_{k+2} . Simplify and show

$$x_k^{k+2} = x_k^k + J_k(x_{k+1}^{k+2} - x_{k+1}^k).$$

- (d) Using induction, conclude

$$x_k^n = x_k^k + J_k(x_{k+1}^n - x_{k+1}^k),$$

which yields the smoother with $k = t - 1$.

Section 6.3

- 6.6** Consider the univariate state-space model given by state conditions $x_0 = w_0$, $x_t = x_{t-1} + w_t$ and observations $y_t = x_t + v_t$, $t = 1, 2, \dots$, where w_t and v_t are independent, Gaussian, white noise processes with $\text{var}(w_t) = \sigma_w^2$ and $\text{var}(v_t) = \sigma_v^2$.

- (a) Show that y_t follows an IMA(1,1) model, that is, ∇y_t follows an MA(1) model.
 (b) Fit the model specified in part (a) to the logarithm of the glacial varve series and compare the results to those presented in Example 3.33.

- 6.7** Consider the model

$$y_t = x_t + v_t,$$

where v_t is Gaussian white noise with variance σ_v^2 , x_t are independent Gaussian random variables with mean zero and $\text{var}(x_t) = r_t \sigma_x^2$ with x_t independent of v_t , and r_1, \dots, r_n are known constants. Show that applying the EM algorithm to the problem of estimating σ_x^2 and σ_v^2 leads to updates (represented by hats)

$$\hat{\sigma}_x^2 = \frac{1}{n} \sum_{t=1}^n \frac{\sigma_t^2 + \mu_t^2}{r_t} \quad \text{and} \quad \hat{\sigma}_v^2 = \frac{1}{n} \sum_{t=1}^n [(y_t - \mu_t)^2 + \sigma_t^2],$$

where, based on the current estimates (represented by tildes),

$$\mu_t = \frac{r_t \tilde{\sigma}_x^2}{r_t \tilde{\sigma}_x^2 + \tilde{\sigma}_v^2} y_t \quad \text{and} \quad \sigma_t^2 = \frac{r_t \tilde{\sigma}_x^2 \tilde{\sigma}_v^2}{r_t \tilde{\sigma}_x^2 + \tilde{\sigma}_v^2}.$$

6.8 To explore the stability of the filter, consider a univariate state-space model. That is, for $t = 1, 2, \dots$, the observations are $y_t = x_t + v_t$ and the state equation is $x_t = \phi x_{t-1} + w_t$, where $\sigma_w = \sigma_v = 1$ and $|\phi| < 1$. The initial state, x_0 , has zero mean and variance one.

- (a) Exhibit the recursion for P_t^{t-1} in [Property 6.1](#) in terms of P_{t-1}^{t-2} .
- (b) Use the result of (a) to verify P_t^{t-1} approaches a limit ($t \rightarrow \infty$) P that is the positive solution of $P^2 - \phi^2 P - 1 = 0$.
- (c) With $K = \lim_{t \rightarrow \infty} K_t$ as given in [Property 6.1](#), show $|1 - K| < 1$.
- (d) Show, in steady-state, the one-step-ahead predictor, $y_{n+1}^n = E(y_{n+1} \mid y_n, y_{n-1}, \dots)$, of a future observation satisfies

$$y_{n+1}^n = \sum_{j=0}^{\infty} \phi^j K (1 - K)^{j-1} y_{n+1-j}.$$

6.9 In [Section 6.3](#), we discussed that it is possible to obtain a recursion for the gradient vector, $-\partial \ln L_Y(\Theta) / \partial \Theta$. Assume the model is given by [\(6.1\)](#) and [\(6.2\)](#) and A_t is a known design matrix that does not depend on Θ , in which case [Property 6.1](#) applies. For the gradient vector, show

$$\begin{aligned} \partial \ln L_Y(\Theta) / \partial \Theta_i = & \sum_{t=1}^n \left\{ \epsilon_t' \Sigma_t^{-1} \frac{\partial \epsilon_t}{\partial \Theta_i} - \frac{1}{2} \epsilon_t' \Sigma_t^{-1} \frac{\partial \Sigma_t}{\partial \Theta_i} \Sigma_t^{-1} \epsilon_t \right. \\ & \left. + \frac{1}{2} \text{tr} \left(\Sigma_t^{-1} \frac{\partial \Sigma_t}{\partial \Theta_i} \right) \right\}, \end{aligned}$$

where the dependence of the innovation values on Θ is understood. In addition, with the general definition $\partial_i g = \partial g(\Theta) / \partial \Theta_i$, show the following recursions, for $t = 2, \dots, n$ apply:

- (i) $\partial_i \epsilon_t = -A_t \partial_i x_t^{t-1}$,
- (ii) $\partial_i x_t^{t-1} = \partial_i \Phi x_{t-1}^{t-2} + \Phi \partial_i x_{t-1}^{t-2} + \partial_i K_{t-1} \epsilon_{t-1} + K_{t-1} \partial_i \epsilon_{t-1}$,
- (iii) $\partial_i \Sigma_t = A_t \partial_i P_t^{t-1} A_t' + \partial_i R$,
- (iv) $\partial_i K_t = [\partial_i \Phi P_t^{t-1} A_t' + \Phi \partial_i P_t^{t-1} A_t' - K_t \partial_i \Sigma_t] \Sigma_t^{-1}$,
- (v) $\partial_i P_t^{t-1} = \partial_i \Phi P_{t-1}^{t-2} \Phi' + \Phi \partial_i P_{t-1}^{t-2} \Phi' + \Phi P_{t-1}^{t-2} \partial_i \Phi' + \partial_i Q$,
 $- \partial_i K_{t-1} \Sigma_t K_{t-1}' - K_{t-1} \partial_i \Sigma_t K_{t-1}' - K_{t-1} \Sigma_t \partial_i K_{t-1}'$,

using the fact that $P_t^{t-1} = \Phi P_{t-1}^{t-2} \Phi' + Q - K_{t-1} \Sigma_t K_{t-1}'$.

6.10 Continuing with the previous problem, consider the evaluation of the Hessian matrix and the numerical evaluation of the asymptotic variance–covariance matrix of the parameter estimates. The information matrix satisfies

$$E \left\{ -\frac{\partial^2 \ln L_Y(\Theta)}{\partial \Theta \partial \Theta'} \right\} = E \left\{ \left(\frac{\partial \ln L_Y(\Theta)}{\partial \Theta} \right) \left(\frac{\partial \ln L_Y(\Theta)}{\partial \Theta} \right)' \right\};$$

see Anderson (1984, Section 4.4), for example. Show the (i, j) -th element of the information matrix, say, $\mathcal{I}_{ij}(\Theta) = E \{-\partial^2 \ln L_Y(\Theta) / \partial \Theta_i \partial \Theta_j\}$, is

$$\begin{aligned}\mathcal{I}_{ij}(\boldsymbol{\theta}) = \sum_{t=1}^n E\Big\{ & \partial_i \epsilon'_t \Sigma_t^{-1} \partial_j \epsilon_t + \frac{1}{2} \text{tr}(\Sigma_t^{-1} \partial_i \Sigma_t \Sigma_t^{-1} \partial_j \Sigma_t) \\ & + \frac{1}{4} \text{tr}(\Sigma_t^{-1} \partial_i \Sigma_t) \text{tr}(\Sigma_t^{-1} \partial_j \Sigma_t)\Big\}.\end{aligned}$$

Consequently, an approximate Hessian matrix can be obtained from the sample by dropping the expectation, E, in the above result and using only the recursions needed to calculate the gradient vector.

Section 6.4

6.11 As an example of the way the state-space model handles the missing data problem, suppose the first-order autoregressive process

$$x_t = \phi x_{t-1} + w_t$$

has an observation missing at $t = m$, leading to the observations $y_t = A_t x_t$, where $A_t = 1$ for all t , except $t = m$ wherein $A_t = 0$. Assume $x_0 = 0$ with variance $\sigma_w^2/(1 - \phi^2)$, where the variance of w_t is σ_w^2 . Show the Kalman smoother estimators in this case are

$$x_t^n = \begin{cases} \phi y_1 & t = 0, \\ \frac{\phi}{1+\phi^2}(y_{m-1} + y_{m+1}) & t = m, \\ y, & t \neq 0, m, \end{cases}$$

with mean square covariances determined by

$$P_t^n = \begin{cases} \sigma_w^2 & t = 0, \\ \sigma_w^2/(1 + \phi^2) & t = m, \\ 0 & t \neq 0, m. \end{cases}$$

6.12 The data set `ar1miss` is $n = 100$ observations generated from an AR(1) process, $x_t = \phi x_{t-1} + w_t$, with $\phi = .9$ and $\sigma_w = 1$, where 10% of the data have been deleted at random (replaced with `NA`). Use the results of Problem 6.11 to estimate the parameters of the model, ϕ and σ_w , using the EM algorithm, and then estimate the missing values.

Section 6.5

6.13 Redo Example 6.10 on the *logged* Johnson & Johnson quarterly earnings per share.

6.14 Fit a structural model to quarterly unemployment as follows. Use the data in `unemp`, which are monthly. The series can be made quarterly by aggregating and averaging: `y = aggregate(unemp, nfrequency=4, FUN=mean)`, so that `y` is the quarterly average unemployment. Use Example 6.10 as a guide.

Section 6.6

- 6.15** (a) Fit an AR(2) to the recruitment series, R_t in `rec`, and consider a lag-plot of the residuals from the fit versus the SOI series, S_t in `soi`, at various lags, S_{t-h} , for $h = 0, 1, \dots$. Use the lag-plot to argue that S_{t-5} is reasonable to include as an exogenous variable.
 (b) Fit an ARX(2) to R_t using S_{t-5} as an exogenous variable and comment on the results; include an examination of the innovations.

6.16 Use Property 6.6 to complete the following exercises.

- (a) Write a univariate AR(1) model, $y_t = \phi y_{t-1} + v_t$, in state-space form. Verify your answer is indeed an AR(1).
 (b) Repeat (a) for an MA(1) model, $y_t = v_t + \theta v_{t-1}$.
 (c) Write an IMA(1,1) model, $y_t = y_{t-1} + v_t + \theta v_{t-1}$, in state-space form.

6.17 Verify Property 6.5.

6.18 Verify Property 6.6.

Section 6.7

- 6.19** Repeat the bootstrap analysis of Example 6.13 on the entire three-month Treasury bills and rate of inflation data set of 110 observations. Do the conclusions of Example 6.13—that the dynamics of the data are best described in terms of a fixed, rather than stochastic, regression—still hold?

Section 6.8

6.20 Let y_t represent the global temperature series (`globtemp`) shown in Figure 1.2.

- (a) Fit a smoothing spline using gcv (the default) to y_t and plot the result superimposed on the data. Repeat the fit using `spar=.7`; the gcv method yields `spar=.5` approximately. (Example 2.14 on page 70 may help. Also in R, see the help file `?smooth.spline`.)
 (b) Write the model $y_t = x_t + v_t$ with $\nabla^2 x_t = w_t$, in state-space form. Fit this state-space model to y_t , and exhibit a time plot the estimated smoother, \hat{x}_t^n and the corresponding error limits, $\hat{x}_t^n \pm 2\sqrt{\hat{P}_t^n}$ superimposed on the data.
 (c) Superimpose all the fits from parts (a) and (b) [include the error bounds] on the data and briefly compare and contrast the results.

Section 6.9

6.21 Verify (6.132), (6.133), and (6.134).

6.22 Prove Property 6.7 and verify (6.143).

6.23 Fit a Poisson-HMM to the dataset `polio` from the `gamlss.data` package. The data are reported polio cases in the U.S. for the years 1970 to 1983. To get started, install the package and then type

```
library(gamlss.data)      # load package
plot(polio, type='s')    # view the data
```

6.24 Fit a two-state HMM model to the weekly S&P 500 returns that were analyzed in [Example 6.17](#) and compare the results.

Section 6.10

6.25 Fit the switching model described in [Example 6.20](#) to the growth rate of GNP. The data are in `gnp` and, in the notation of the example, y_t is log-GNP and ∇y_t is the growth rate. Use the code in [Example 6.22](#) as a guide.

6.26 Argue that a switching model is reasonable in explaining the behavior of the number of sunspots (see [Figure 4.22](#)) and then fit a switching model to the sunspot data.

Section 6.11

6.27 Fit a stochastic volatility model to the returns of one (or more) of the four financial time series available in the R datasets package as `EuStockMarkets`.

6.28 Fit a stochastic volatility model to the residuals of the GNP (`gnp`) returns analyzed in [Example 3.39](#).

6.29 We consider the stochastic volatility model (6.197).

(a) Show that for any integer m ,

$$\mathbb{E}[r_t^{2m}] = \beta^{2m} \mathbb{E}[r_t^{2m}] \exp(m^2 \sigma_x^2 / 2),$$

where $\sigma_x^2 = \sigma^2 / (1 - \phi^2)$.

(b) Show (6.198).

(c) Show that for any positive integer h , $\text{var}(X_t + X_{t+h}) = 2\sigma_X^2(1 + \phi^h)$.

(d) Show that

$$\text{cov}(r_t^{2m}, r_{t+h}^{2m}) = \beta^{4m} \left(\mathbb{E}[r_t^{2m}] \right)^2 \left(\exp(m^2 \sigma_x^2 (1 + \phi^h)) - \exp(m^2 \sigma_x^2) \right).$$

(e) Establish (6.199).

Section 6.12

6.30 Verify the distributional statements made in [Example 6.25](#).

6.31 Repeat [Example 6.27](#) on the log of the Johnson & Johnson data.

6.32 Fit an AR(1) to the returns of the US GNP (`gnp`) using a Bayesian approach via MCMC.

Chapter 7

Statistical Methods in the Frequency Domain

In previous chapters, we saw many applied time series problems that involved relating series to each other or to evaluating the effects of treatments or design parameters that arise when time-varying phenomena are subjected to periodic stimuli. In many cases, the nature of the physical or biological phenomena under study are best described by their Fourier components rather than by the difference equations involved in ARIMA or state-space models. The fundamental tools we use in studying periodic phenomena are the discrete Fourier transforms (DFTs) of the processes and their statistical properties. Hence, in [Section 7.2](#), we review the properties of the DFT of a multivariate time series and discuss various approximations to the likelihood function based on the large-sample properties and the properties of the complex multivariate normal distribution. This enables extension of the classical techniques such as ANOVA and principal component analysis to the multivariate time series case, which is the focus of this chapter.

7.1 Introduction

An extremely important class of problems in classical statistics develops when we are interested in relating a collection of input series to some output series. For example, in [Chapter 2](#), we have previously considered relating temperature and various pollutant levels to daily mortality, but have not investigated the frequencies that appear to be driving the relation and have not looked at the possibility of leading or lagging effects. In [Chapter 4](#), we isolated a definite lag structure that could be used to relate sea surface temperature to the number of new recruits. In [Problem 5.10](#), the possible driving processes that could be used to explain inflow to Lake Shasta were hypothesized in terms of the possible inputs precipitation, cloud cover, temperature, and other variables. Identifying the combination of input factors that produce the best prediction for inflow is an example of multiple regression in the frequency domain, with the models treated theoretically by considering the regression, conditional on the random input processes.

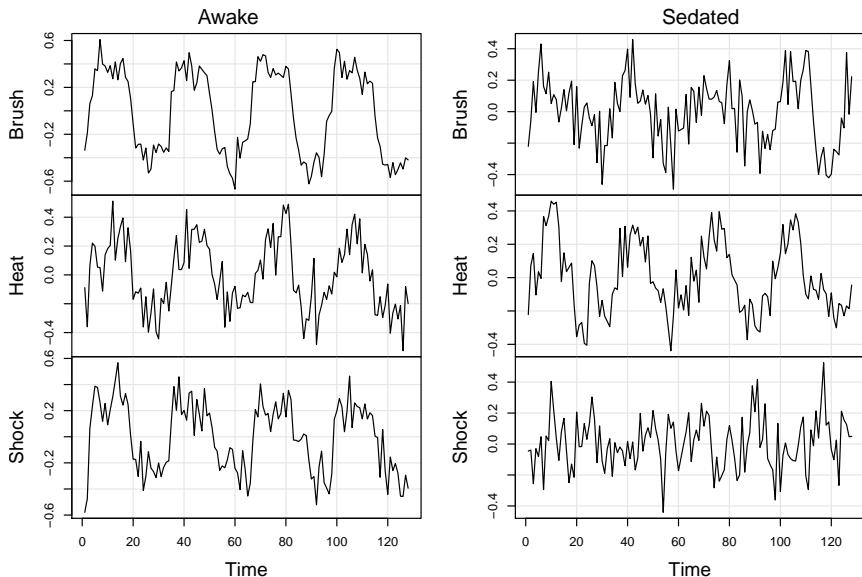


Fig. 7.1. Mean response of subjects to various combinations of periodic stimuli measured at the cortex (primary somatosensory, contralateral). In the first column, the subjects are awake, in the second column the subjects are under mild anesthesia. In the first row, the stimulus is a brush on the hand, the second row involves the application of heat, and the third row involves a low level shock.

A situation somewhat different from that above would be one in which the input series are regarded as fixed and known. In this case, we have a model analogous to that occurring in *analysis of variance*, in which the analysis now can be performed on a frequency by frequency basis. This analysis works especially well when the inputs are dummy variables, depending on some configuration of treatment and other design effects and when effects are largely dependent on periodic stimuli. As an example, we will look at a designed experiment measuring the fMRI brain responses of a number of awake and mildly anesthetized subjects to several levels of periodic brushing, heat, and shock effects. Some limited data from this experiment have been discussed previously in [Example 1.6](#). [Figure 7.1](#) shows mean responses to various levels of periodic heat, brushing, and shock stimuli for subjects awake and subjects under mild anesthesia. The stimuli were periodic in nature, applied alternately for 32 seconds (16 points) and then stopped for 32 seconds. The periodic input signal comes through under all three design conditions when the subjects are awake, but is somewhat attenuated under anesthesia. The mean shock level response hardly shows on the input signal; shock levels were designed to simulate surgical incision without inflicting tissue damage. The means in [Figure 7.1](#) are from a single location. Actually, for each individual, some nine series were recorded at various locations in the brain. It is natural to consider testing the effects of brushing, heat, and shock under the two

levels of consciousness, using a time series generalization of analysis of variance. The R code used to generate Figure 7.1 is:

```
x = matrix(0, 128, 6)
for (i in 1:6) { x[,i] = rowMeans(fmri[[i]]) }
colnames(x) = c("Brush", "Heat", "Shock", "Brush", "Heat", "Shock")
plot.ts(x, main="")
mtext("Awake", side=3, line=1.2, adj=.05, cex=1.2)
mtext("Sedated", side=3, line=1.2, adj=.85, cex=1.2)
```

A generalization to random coefficient regression is also considered, paralleling the univariate approach to signal extraction and detection presented in Section 4.9. This method enables a treatment of multivariate ridge-type regressions and *inversion problems*. Also, the usual random effects analysis of variance in the frequency domain becomes a special case of the random coefficient model.

The extension of frequency domain methodology to more classical approaches to multivariate discrimination and clustering is of interest in the frequency dependent case. Many time series differ in their means and in their autocovariance functions, making the use of both the mean function and the spectral density matrices relevant. As an example of such data, consider the bivariate series consisting of the P and S components derived from several earthquakes and explosions, such as those shown in Figure 7.2, where the P and S components, representing different arrivals have been separated from the first and second halves, respectively, of waveforms like those shown originally in Figure 1.7.

Two earthquakes and two explosions from a set of eight earthquakes and explosions are shown in Figure 7.2 and some essential differences exist that might be used to characterize the two classes of events. Also, the frequency content of the two components of the earthquakes appears to be lower than those of the explosions, and relative amplitudes of the two classes appear to differ. For example, the ratio of the S to P amplitudes in the earthquake group is much higher for this restricted subset. Spectral differences were also noticed in Chapter 4, where the explosion processes had a stronger high-frequency component relative to the low-frequency contributions. Examples like these are typical of applications in which the essential differences between multivariate time series can be expressed by the behavior of either the frequency-dependent mean value functions or the spectral matrix. In *discriminant analysis*, these types of differences are exploited to develop combinations of linear and quadratic classification criteria. Such functions can then be used to classify events of unknown origin, such as the Novaya Zemlya event shown in Figure 7.2, which tends to bear a visual resemblance to the explosion group. The R code used to produce Figure 7.2 is:

```
attach(eqexp) # so you can use the names of the series
P = 1:1024; S = P+1024
x = cbind(EQ5[P], EQ6[P], EX5[P], EX6[P], NZ[P], EQ5[S], EQ6[S], EX5[S],
           EX6[S], NZ[S])
x.name = c("EQ5", "EQ6", "EX5", "EX6", "NZ")
colnames(x) = c(x.name, x.name)
plot.ts(x, main="")
mtext("P waves", side=3, line=1.2, adj=.05, cex=1.2)
mtext("S waves", side=3, line=1.2, adj=.85, cex=1.2)
```

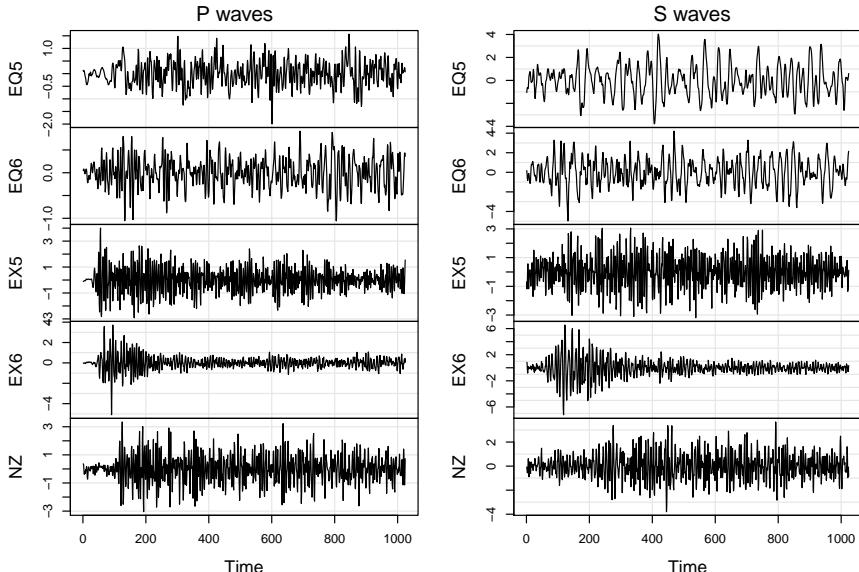


Fig. 7.2. Various bivariate earthquakes (EQ) and explosions (EX) recorded at 40 pts/sec compared with an event NZ (Novaya Zemlya) of unknown origin. Compressional waves, also known as primary or P waves, travel fastest in the Earth's crust and are first to arrive. Shear waves propagate more slowly through the Earth and arrive second, hence they are called secondary or S waves.

Finally, for multivariate processes, the structure of the spectral matrix is also of great interest. We might reduce the dimension of the underlying process to a smaller set of input processes that explain most of the variability in the cross-spectral matrix as a function of frequency. *Principal component analysis* can be used to decompose the spectral matrix into a smaller subset of component factors that explain decreasing amounts of power. For example, the hydrological data might be explained in terms of a component process that weights heavily on precipitation and inflow and one that weights heavily on temperature and cloud cover. Perhaps these two components could explain most of the power in the spectral matrix at a given frequency. The ideas behind principal component analysis can also be generalized to include an optimal scaling methodology for categorical data called the *spectral envelope* (see Stoffer et al., 1993).

7.2 Spectral Matrices and Likelihood Functions

We have previously argued for an approximation to the log likelihood based on the joint distribution of the DFTs in (4.85), where we used approximation as an aid in estimating parameters for certain parameterized spectra. In this chapter, we make heavy use of the fact that the sine and cosine transforms of the $p \times 1$ vector process

$x_t = (x_{t1}, x_{t2}, \dots, x_{tp})'$ with mean $\mathbf{E}x_t = \mu_t$, say, with DFT^{7.1}

$$X(\omega_k) = n^{-1/2} \sum_{t=1}^n x_t e^{-2\pi i \omega_k t} = X_c(\omega_k) - iX_s(\omega_k) \quad (7.1)$$

and mean

$$M(\omega_k) = n^{-1/2} \sum_{t=1}^n \mu_t e^{-2\pi i \omega_k t} = M_c(\omega_k) - iM_s(\omega_k) \quad (7.2)$$

will be approximately uncorrelated, where we evaluate at the usual Fourier frequencies $\{\omega_k = k/n, 0 < |\omega_k| < 1/2\}$. By [Theorem C.6](#), the approximate $2p \times 2p$ covariance matrix of the cosine and sine transforms, say, $X(\omega_k) = (X_c(\omega_k)', X_s(\omega_k)')'$, is

$$\Sigma(\omega_k) = \frac{1}{2} \begin{pmatrix} C(\omega_k) & -Q(\omega_k) \\ Q(\omega_k) & C(\omega_k) \end{pmatrix}, \quad (7.3)$$

and the real and imaginary parts are jointly normal. This result implies, by the results stated in [Appendix C](#), the density function of the vector DFT, say, $X(\omega_k)$, can be approximated as

$$p(\omega_k) \approx |f(\omega_k)|^{-1} \exp\{-\overline{(X(\omega_k) - M(\omega_k))}^* f^{-1}(\omega_k) (X(\omega_k) - M(\omega_k))\},$$

where the spectral matrix is the usual

$$f(\omega_k) = C(\omega_k) - iQ(\omega_k). \quad (7.4)$$

Certain computations that we do in the section on discriminant analysis will involve approximating the joint likelihood by the product of densities like the one given above over subsets of the frequency band $0 < \omega_k < 1/2$.

To use the likelihood function for estimating the spectral matrix, for example, we appeal to the limiting result implied by [Theorem C.7](#) and again choose L frequencies in the neighborhood of some target frequency ω , say, $X(\omega_k \pm k/n)$, for $k = 1, \dots, m$ and $L = 2m + 1$. Then, let X_ℓ denote the indexed values, and note the DFTs of the mean adjusted vector process are approximately jointly normal with mean zero and complex covariance matrix $f = f(\omega)$. Then, write the log likelihood over the L sub-frequencies as

$$\ln L_X(f(\omega_k)) \approx -L \ln |f(\omega_k)| - \sum_{\ell=-m}^m (X_\ell - M_\ell)^* f(\omega_k)^{-1} (X_\ell - M_\ell). \quad (7.5)$$

The use of spectral approximations to the likelihood has been fairly standard, beginning with the work of Whittle (1961) and continuing in Brillinger (1981) and Hannan

^{7.1} In previous chapters, the DFT of a process x_t was denoted by $d_x(\omega_k)$. In this chapter, we will consider the Fourier transforms of many different processes and so, to avoid the overuse of subscripts and to ease the notation, we use a capital letter, e.g., $X(\omega_k)$, to denote the DFT of x_t . This notation is standard in the digital signal processing (DSP) literature.

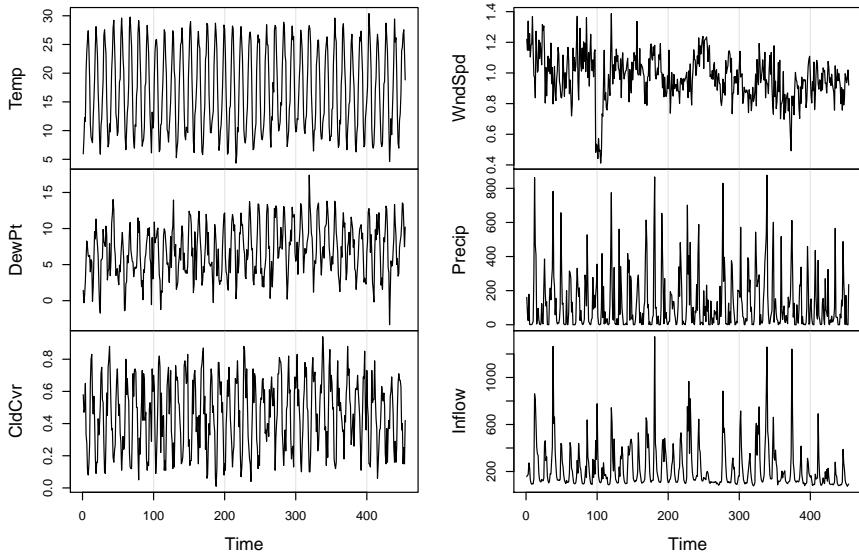


Fig. 7.3. Monthly values of weather and inflow at Lake Shasta ([climhyd](#)).

(1970). Assuming the mean adjusted series are available, i.e., M_ℓ is known, we obtain the maximum likelihood estimator for f , namely,

$$\hat{f}(\omega_k) = L^{-1} \sum_{\ell=-m}^m (X_\ell - M_\ell)(X_\ell - M_\ell)^*; \quad (7.6)$$

see [Problem 7.2](#).

7.3 Regression for Jointly Stationary Series

In [Section 4.7](#), we considered a model of the form

$$y_t = \sum_{r=-\infty}^{\infty} \beta_{1r} x_{t-r,1} + v_t, \quad (7.7)$$

where x_{t1} is a single observed input series and y_t is the observed output series, and we are interested in estimating the filter coefficients β_{1r} relating the adjacent lagged values of x_{t1} to the output series y_t . In the case of the SOI and Recruitment series, we identified the El Niño driving series as x_{t1} , the input and y_t , the Recruitment series, as the output. In general, more than a single plausible input series may exist. For example, the Lake Shasta inflow hydrological data ([climhyd](#)) shown in [Figure 7.3](#) suggests there may be at least five possible series driving the inflow; see [Example 7.1](#) for more details. Hence, we may envision a $q \times 1$ input vector of driving series,

say, $x_t = (x_{t1}, x_{t2}, \dots, x_{tq})'$, and a set of $q \times 1$ vector of regression functions $\beta_r = (\beta_{1r}, \beta_{2r}, \dots, \beta_{qr})'$, which are related as

$$y_t = \sum_{r=-\infty}^{\infty} \beta'_r x_{t-r} + v_t = \sum_{j=1}^q \sum_{r=-\infty}^{\infty} \beta_{jr} x_{t-r,j} + v_t, \quad (7.8)$$

which shows that the output is a sum of linearly filtered versions of the input processes and a stationary noise process v_t , assumed to be uncorrelated with x_t . Each filtered component in the sum over j gives the contribution of lagged values of the j -th input series to the output series. We assume the regression functions β_{jr} are fixed and unknown.

The model given by (7.8) is useful under several different scenarios, corresponding to a number of different assumptions that can be made about the components. Assuming the input and output processes are jointly stationary with zero means leads to the conventional regression analysis given in this section. The analysis depends on theory that assumes we observe the output process y_t conditional on fixed values of the input vector x_t ; this is the same as the assumptions made in conventional regression analysis. Assumptions considered later involve letting the coefficient vector β_t be a random unknown signal vector that can be estimated by Bayesian arguments, using the conditional expectation given the data. The answers to this approach, given in [Section 7.5](#), allow signal extraction and deconvolution problems to be handled. Assuming the inputs are fixed allows various experimental designs and analysis of variance to be done for both fixed and random effects models. Estimation of the frequency-dependent random effects variance components in the analysis of variance model is also considered in [Section 7.5](#).

For the approach in this section, assume the inputs and outputs have zero means and are jointly stationary with the $(q+1) \times 1$ vector process $(x'_t, y_t)'$ of inputs x_t and outputs y_t assumed to have a spectral matrix of the form

$$f(\omega) = \begin{pmatrix} f_{xx}(\omega) & f_{xy}(\omega) \\ f_{yx}(\omega) & f_{yy}(\omega) \end{pmatrix}, \quad (7.9)$$

where $f_{yx}(\omega) = (f_{yx_1}(\omega), f_{yx_2}(\omega), \dots, f_{yx_q}(\omega))$ is the $1 \times q$ vector of cross-spectra relating the q inputs to the output and $f_{xx}(\omega)$ is the $q \times q$ spectral matrix of the inputs. Generally, we observe the inputs and search for the vector of regression functions β_t relating the inputs to the outputs. We assume all autocovariance functions satisfy the absolute summability conditions of the form

$$\sum_{h=-\infty}^{\infty} |h| |\gamma_{jk}(h)| < \infty. \quad (7.10)$$

$(j, k = 1, \dots, q+1)$, where $\gamma_{jk}(h)$ is the autocovariance corresponding to the cross-spectrum $f_{jk}(\omega)$ in (7.9). We also need to assume a linear process of the form (C.35) as a condition for using [Theorem C.7](#) on the joint distribution of the discrete Fourier transforms in the neighborhood of some fixed frequency.

ESTIMATION OF THE REGRESSION FUNCTION

In order to estimate the regression function β_r , the Projection Theorem (Appendix B) applied to minimizing

$$\text{MSE} = E \left[(y_t - \sum_{r=-\infty}^{\infty} \beta'_r x_{t-r})^2 \right] \quad (7.11)$$

leads to the orthogonality conditions

$$E \left[(y_t - \sum_{r=-\infty}^{\infty} \beta'_r x_{t-r}) x'_{t-s} \right] = 0' \quad (7.12)$$

for all $s = 0, \pm 1, \pm 2, \dots$, where $0'$ denotes the $1 \times q$ zero vector. Taking the expectations inside and substituting for the definitions of the autocovariance functions appearing and leads to the normal equations

$$\sum_{r=-\infty}^{\infty} \beta'_r \Gamma_{xx}(s-r) = \gamma'_{yx}(s), \quad (7.13)$$

for $s = 0, \pm 1, \pm 2, \dots$, where $\Gamma_{xx}(s)$ denotes the $q \times q$ autocovariance matrix of the vector series x_t at lag s and $\gamma_{yx}(s) = (\gamma_{yx_1}(s), \dots, \gamma_{yx_q}(s))$ is a $1 \times q$ vector containing the lagged covariances between y_t and x_t . Again, a frequency domain approximate solution is easier in this case because the computations can be done frequency by frequency using cross-spectra that can be estimated from sample data using the DFT. In order to develop the frequency domain solution, substitute the representation into the normal equations, using the same approach as used in the simple case derived in [Section 4.7](#). This approach yields

$$\int_{-1/2}^{1/2} \sum_{r=-\infty}^{\infty} \beta'_r e^{2\pi i \omega(s-r)} f_{xx}(\omega) d\omega = \gamma'_{yx}(s).$$

Now, because $\gamma'_{yx}(s)$ is the Fourier transform of the cross-spectral vector $f_{yx}(\omega) = f_{xy}^*(\omega)$, we might write the system of equations in the frequency domain, using the uniqueness of the Fourier transform, as

$$B'(\omega) f_{xx}(\omega) = f_{xy}^*(\omega), \quad (7.14)$$

where $f_{xx}(\omega)$ is the $q \times q$ spectral matrix of the inputs and $B(\omega)$ is the $q \times 1$ vector Fourier transform of β_t . Multiplying (7.14) on the right by $f_{xx}^{-1}(\omega)$, assuming $f_{xx}(\omega)$ is nonsingular at ω , leads to the *frequency domain estimator*

$$B'(\omega) = f_{xy}^*(\omega) f_{xx}^{-1}(\omega). \quad (7.15)$$

Note, (7.15) implies the regression function would take the form

$$\beta_t = \int_{-1/2}^{1/2} B(\omega) e^{2\pi i \omega t} d\omega. \quad (7.16)$$

As before, it is conventional to introduce the DFT as the approximate estimator for the integral (7.16) and write

$$\beta_t^M = M^{-1} \sum_{k=0}^{M-1} B(\omega_k) e^{2\pi i \omega_k t}, \quad (7.17)$$

where $\omega_k = k/M$, $M \ll n$. The approximation was shown in [Problem 4.35](#) to hold exactly as long as $\beta_t = 0$ for $|t| \geq M/2$ and to have a mean-squared-error bounded by a function of the zero-lag autocovariance and the absolute sum of the neglected coefficients.

The mean-squared-error (7.11) can be written using the orthogonality principle, giving

$$\text{MSE} = \int_{-1/2}^{1/2} f_{y \cdot x}(\omega) d\omega, \quad (7.18)$$

where

$$f_{y \cdot x}(\omega) = f_{yy}(\omega) - f_{xy}^*(\omega) f_{xx}^{-1}(\omega) f_{xy}(\omega) \quad (7.19)$$

denotes the residual or error spectrum. The resemblance of (7.19) to the usual equations in regression analysis is striking. It is useful to pursue the multiple regression analogy further by noting a *squared multiple coherence* can be defined as

$$\rho_{y \cdot x}^2(\omega) = \frac{f_{xy}^*(\omega) f_{xx}^{-1}(\omega) f_{xy}(\omega)}{f_{yy}(\omega)}. \quad (7.20)$$

This expression leads to the mean squared error in the form

$$\text{MSE} = \int_{-1/2}^{1/2} f_{yy}(\omega) [1 - \rho_{y \cdot x}^2(\omega)] d\omega, \quad (7.21)$$

and we have an interpretation of $\rho_{y \cdot x}^2(\omega)$ as the *proportion of power* accounted for by the lagged regression on x_t at frequency ω . If $\rho_{y \cdot x}^2(\omega) = 0$ for all ω , we have

$$\text{MSE} = \int_{-1/2}^{1/2} f_{yy}(\omega) d\omega = E[y_t^2],$$

which is the mean squared error when no predictive power exists. As long as $f_{xx}(\omega)$ is positive definite at all frequencies, $\text{MSE} \geq 0$, and we will have

$$0 \leq \rho_{y \cdot x}^2(\omega) \leq 1 \quad (7.22)$$

for all ω . If the multiple coherence is unity for all frequencies, the mean squared error in (7.21) is zero and the output series is perfectly predicted by a linearly filtered combination of the inputs. [Problem 7.3](#) shows the ordinary squared coherence between the series y_t and the linearly filtered combinations of the inputs appearing in (7.11) is exactly (7.20).

ESTIMATION USING SAMPLED DATA

Clearly, the matrices of spectra and cross-spectra will not ordinarily be known, so the regression computations need to be based on sampled data. We assume, therefore, the inputs $x_{t1}, x_{t2}, \dots, x_{tq}$ and output y_t series are available at the time points $t = 1, 2, \dots, n$, as in Chapter 4. In order to develop reasonable estimates for the spectral quantities, some replication must be assumed. Often, only one replication of each of the inputs and the output will exist, so it is necessary to assume a band exists over which the spectra and cross-spectra are approximately equal to $f_{xx}(\omega)$ and $f_{xy}(\omega)$, respectively. Then, let $Y(\omega_k + \ell/n)$ and $X(\omega_k + \ell/n)$ be the DFTs of y_t and x_t over the band, say, at frequencies of the form

$$\omega_k \pm \ell/n, \quad \ell = 1, \dots, m,$$

where $L = 2m + 1$ as before. Then, simply substitute the sample spectral matrix

$$\hat{f}_{xx}(\omega) = L^{-1} \sum_{\ell=-m}^m X(\omega_k + \ell/n) X^*(\omega_k + \ell/n) \quad (7.23)$$

and the vector of sample cross-spectra

$$\hat{f}_{xy}(\omega) = L^{-1} \sum_{\ell=-m}^m X(\omega_k + \ell/n) \overline{Y(\omega_k + \ell/n)} \quad (7.24)$$

for the respective terms in (7.15) to get the regression estimator $\hat{B}(\omega)$. For the regression estimator (7.17), we may use

$$\hat{\beta}_t^M = \frac{1}{M} \sum_{k=0}^{M-1} \hat{f}_{xy}^*(\omega_k) \hat{f}_{xx}^{-1}(\omega_k) e^{2\pi i \omega_k t} \quad (7.25)$$

for $t = 0, \pm 1, \pm 2, \dots, \pm(M/2 - 1)$, as the estimated regression function.

TESTS OF HYPOTHESES

The estimated squared multiple coherence, corresponding to the theoretical coherence (7.20), becomes

$$\hat{\rho}_{y \cdot x}^2(\omega) = \frac{\hat{f}_{xy}^*(\omega) \hat{f}_{xx}^{-1}(\omega) \hat{f}_{xy}(\omega)}{\hat{f}_{yy}(\omega)}. \quad (7.26)$$

We may obtain a distributional result for the multiple coherence function analogous to that obtained in the univariate case by writing the multiple regression model in the frequency domain, as was done in Section 4.5. We obtain the statistic

$$F_{2q, 2(L-q)} = \frac{(L-q)}{q} \frac{\hat{\rho}_{y \cdot x}^2(\omega)}{[1 - \hat{\rho}_{y \cdot x}^2(\omega)]}, \quad (7.27)$$

Table 7.1. ANOPOW for the Partitioned Regression Model

Source	Power	Degrees of Freedom
$x_{t,q_1+1}, \dots, x_{t,q_1+q_2}$	$\text{SSR}(\omega)$ (7.34)	$2q_2$
Error	$\text{SSE}(\omega)$ (7.35)	$2(L - q_1 - q_2)$
Total	$L\hat{f}_{y,1}(\omega)$	$2(L - q_1)$

which has an F -distribution with $2q$ and $2(L - q)$ degrees of freedom under the null hypothesis that $\rho_{y,x}^2(\omega) = 0$, or equivalently, that $B(\omega) = 0$, in the model

$$Y(\omega_k + \ell/n) = B'(\omega)X(\omega_k + \ell/n) + V(\omega_k + \ell/n), \quad (7.28)$$

where the spectral density of the error $V(\omega_k + \ell/n)$ is $f_{y,x}(\omega)$. **Problem 7.4** sketches a derivation of this result.

A second kind of hypothesis of interest is one that might be used to test whether a full model with q inputs is significantly better than some submodel with $q_1 < q$ components. In the time domain, this hypothesis implies, for a partition of the vector of inputs into q_1 and q_2 components ($q_1 + q_2 = q$), say, $x_t = (x'_{t1}, x'_{t2})'$, and the similarly partitioned vector of regression functions $\beta_t = (\beta'_{1t}, \beta'_{2t})'$, we would be interested in testing whether $\beta_{2t} = 0$ in the partitioned regression model

$$y_t = \sum_{r=-\infty}^{\infty} \beta'_{1r} x_{t-r,1} + \sum_{r=-\infty}^{\infty} \beta'_{2r} x_{t-r,2} + v_t. \quad (7.29)$$

Rewriting the regression model (7.29) in the frequency domain in a form that is similar to (7.28) establishes that, under the partitions of the spectral matrix into its $q_i \times q_j$ ($i, j = 1, 2$) submatrices, say,

$$\hat{f}_{xx}(\omega) = \begin{pmatrix} \hat{f}_{11}(\omega) & \hat{f}_{12}(\omega) \\ \hat{f}_{21}(\omega) & \hat{f}_{22}(\omega) \end{pmatrix}, \quad (7.30)$$

and the cross-spectral vector into its $q_i \times 1$ ($i = 1, 2$) subvectors,

$$\hat{f}_{xy}(\omega) = \begin{pmatrix} \hat{f}_{1y}(\omega) \\ \hat{f}_{2y}(\omega) \end{pmatrix}, \quad (7.31)$$

we may test the hypothesis $\beta_{2t} = 0$ at frequency ω by comparing the estimated residual power

$$\hat{f}_{y,x}(\omega) = \hat{f}_{yy}(\omega) - \hat{f}_{xy}^*(\omega)\hat{f}_{xx}^{-1}(\omega)\hat{f}_{xy}(\omega) \quad (7.32)$$

under the full model with that under the reduced model, given by

$$\hat{f}_{y,1}(\omega) = \hat{f}_{yy}(\omega) - \hat{f}_{1y}^*(\omega)\hat{f}_{11}^{-1}(\omega)\hat{f}_{1y}(\omega). \quad (7.33)$$

The power due to regression can be written as

$$\text{SSR}(\omega) = L[\hat{f}_{y,1}(\omega) - \hat{f}_{y,x}(\omega)], \quad (7.34)$$

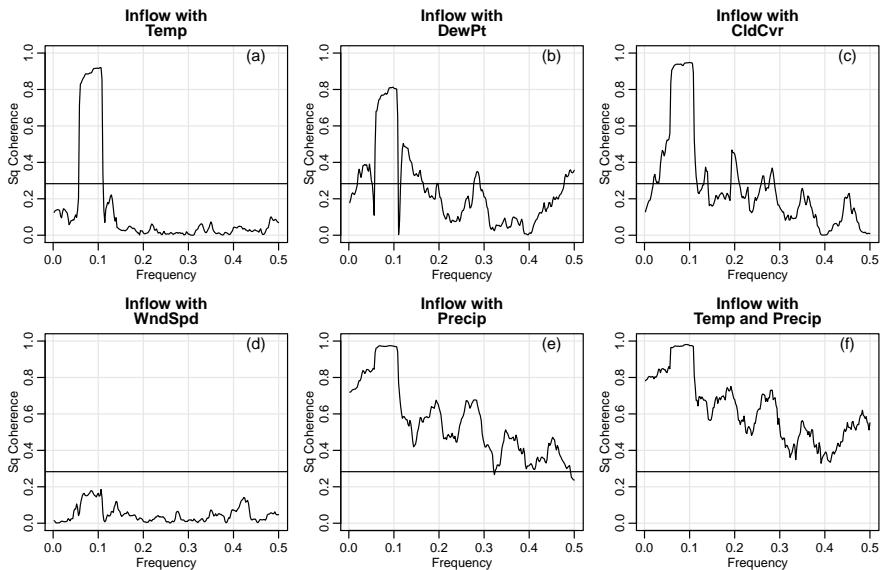


Fig. 7.4. Squared coherency between Lake Shasta inflow and (a) temperature; (b) dew point; (c) cloud cover; (d) wind speed; (e) precipitation. The multiple coherency between inflow and temperature – precipitation jointly is displayed in (f). In each case, the .001 threshold is exhibited as a horizontal line.

with the usual error power given by

$$\text{SSE}(\omega) = L \hat{f}_{y \cdot x}(\omega). \quad (7.35)$$

The test of no regression proceeds using the F -statistic

$$F_{2q_2, 2(L-q)} = \frac{(L-q)}{q_2} \frac{\text{SSR}(\omega)}{\text{SSE}(\omega)}. \quad (7.36)$$

The distribution of this F -statistic with $2q_2$ numerator degrees of freedom and $2(L-q)$ denominator degrees of freedom follows from an argument paralleling that given in Chapter 4 for the case of a single input. The test results can be summarized in an *Analysis of Power* (ANOPOW) table that parallels the usual analysis of variance (ANOVA) table. **Table 7.1** shows the components of power for testing $\beta_{2l} = 0$ at a particular frequency ω . The ratio of the two components divided by their respective degrees of freedom just yields the F -statistic (7.36) used for testing whether the q_2 add significantly to the predictive power of the regression on the q_1 series.

Example 7.1 Predicting Lake Shasta Inflow

We illustrate some of the preceding ideas by considering the problem of predicting the transformed (logged) inflow series shown in [Figure 7.3](#) from some combination of the inputs. First, look for the best single input predictor using the squared coherence function (7.26). The results, exhibited in [Figure 7.4\(a\)-\(e\)](#), show transformed

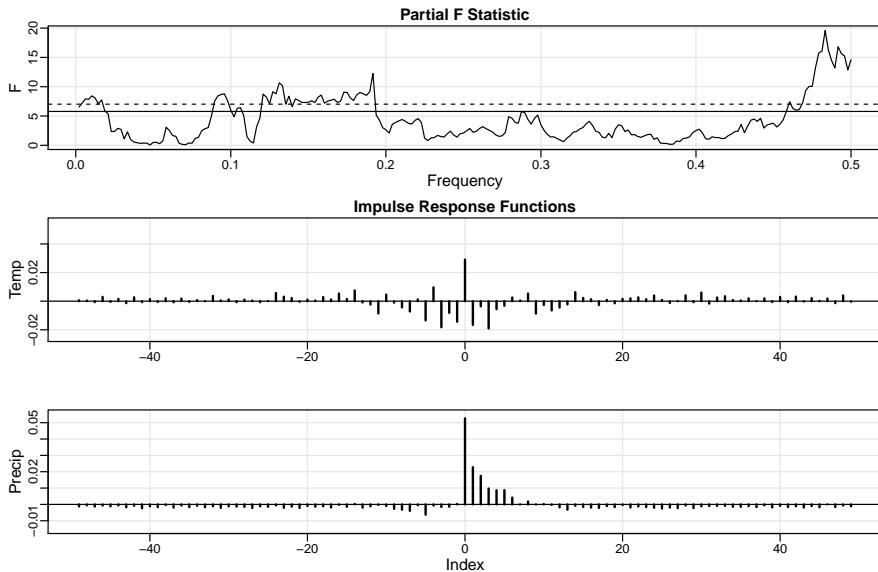


Fig. 7.5. Partial F -statistics [top] for testing whether temperature adds to the ability to predict Lake Shasta inflow when precipitation is included in the model. The dashed line indicates the .001 FDR level and the solid line represents the corresponding quantile of the null F distribution. Multiple impulse response functions for the regression relations of temperature [middle] and precipitation [bottom].

(square root) precipitation produces the most consistently high squared coherence values at all frequencies ($L = 25$), with the seasonal period contributing most significantly. Other inputs, with the exception of wind speed, also appear to be plausible contributors. Figure 7.4(a)-(e) shows a .001 threshold corresponding to the F -statistic, separately, for each possible predictor of inflow.

Next, we focus on the analysis with two predictor series, temperature and transformed precipitation. The additional contribution of temperature to the model seems somewhat marginal because the multiple coherence (7.26), shown in the top panel of Figure 7.4(f) seems only slightly better than the univariate coherence with precipitation shown in Figure 7.4(e). It is, however, instructive to produce the multiple regression functions, using (7.25) to see if a simple model for inflow exists that would involve some regression combination of inputs temperature and precipitation that would be useful for predicting inflow to Shasta Lake. The top of Figure 7.5 shows the partial F -statistic, (7.36), for testing if temperature is predictive of inflow when precipitation is in the model. In addition, threshold values corresponding to a false discovery rate (FDR) of .001 (see Benjamini & Hochberg, 1995) and the corresponding null F quantile are displayed in that figure.

Although the contribution of temperature is marginal, it is instructive to produce the multiple regression functions, using (7.25), to see if a simple model for inflow exists that would involve some regression combination of inputs temperature and

precipitation that would be useful for predicting inflow to Lake Shasta. With this in mind, denoting the possible inputs P_t for transformed precipitation and T_t for transformed temperature, the regression function has been plotted in the lower two panels of Figure 7.5 using a value of $M = 100$ for each of the two inputs. In that figure, the time index runs over both positive and negative values and are centered at time $t = 0$. Hence, the relation with temperature seems to be instantaneous and positive and an exponentially decaying relation to precipitation exists that has been noticed previously in the analysis in Problem 4.37. The plots suggest a transfer function model of the general form fitted to the Recruitment and SOI series in Example 5.8. We might propose fitting the inflow output, say, I_t , using the model

$$I_t = \alpha_0 + \frac{\delta_0}{(1 - \omega_1 B)} P_t + \alpha_2 T_t + \eta_t,$$

which is the transfer function model, without the temperature component, considered in that section. The R code for this example is as follows.

```
plot.ts(climhyd)      # Figure 7.3
Y = climhyd          # Y holds the transformed series
Y[,6] = log(Y[,6])   # log inflow
Y[,5] = sqrt(Y[,5])  # sqrt precipitation
L = 25; M = 100; alpha = .001; fdr = .001
nq = 2                # number of inputs (Temp and Precip)
# Spectral Matrix
Yspec = mvspec(Y, spans=L, kernel="daniell", detrend=TRUE, demean=FALSE,
               taper=.1)
n = Yspec$used        # effective sample size
Fr = Yspec$freq        # fundamental freqs
n.freq = length(Fr)    # number of frequencies
Yspec$bandwidth*sqrt(12) # = 0.050 - the bandwidth
# Coherencies
Fq = qf(1-alpha, 2, L-2)
cn = Fq/(L-1+Fq)
plt.name = c("(a)", "(b)", "(c)", "(d)", "(e)", "(f)")
dev.new(); par(mfrow=c(2,3), cex.lab=1.2)
# The coherencies are listed as 1,2,...,15=choose(6,2)
for (i in 11:15){
  plot(Fr, Yspec$coh[,i], type="l", ylab="Sq Coherence", xlab="Frequency",
       ylim=c(0,1), main=c("Inflow with", names(climhyd[i-10])))
  abline(h = cn); text(.45, .98, plt.name[i-10], cex=1.2) }
# Multiple Coherency
coh.15 = stoch.reg(Y, cols.full = c(1,5), cols.red = NULL, alpha, L, M,
                    plot.which = "coh")
text(.45, .98, plt.name[6], cex=1.2)
title(main = c("Inflow with", "Temp and Precip"))
# Partial F (called eF; avoid use of F alone)
numer.df = 2*nq; denom.df = Yspec$df-2*nq
dev.new()
par(mfrow=c(3,1), mar=c(3,3,2,1)+.5, mgp = c(1.5,0.4,0), cex.lab=1.2)
out.15 = stoch.reg(Y, cols.full = c(1,5), cols.red = 5, alpha, L, M,
                    plot.which = "F.stat")
eF = out.15$eF
pvals = pf(eF, numer.df, denom.df, lower.tail = FALSE)
pID = FDR(pvals, fdr); abline(h=c(eF$pID), lty=2)
title(main = "Partial F Statistic")
```

```
# Regression Coefficients
S = seq(from = -M/2+1, to = M/2 - 1, length = M-1)
plot(S, coh.15$Betahat[,1], type = "h", xlab = "", ylab = names(climhyd[1]),
      ylim = c(-.025, .055), lwd=2)
abline(h=0); title(main = "Impulse Response Functions")
plot(S, coh.15$Betahat[,2], type = "h", xlab = "Index", ylab =
      names(climhyd[5]), ylim = c(-.015, .055), lwd=2)
abline(h=0)
```

7.4 Regression with Deterministic Inputs

The previous section considered the case in which the input and output series were jointly stationary, but there are many circumstances in which we might want to assume that the input functions are fixed and have a known functional form. This happens in the analysis of data from designed experiments. For example, we may want to take a collection of earthquakes and explosions such as are shown in Figure 7.2 and test whether the mean functions are the same for either the P or S components or, perhaps, for them jointly. In certain other signal detection problems using arrays, the inputs are used as dummy variables to express lags corresponding to the arrival times of the signal at various elements, under a model corresponding to that of a plane wave from a fixed source propagating across the array. In Figure 7.1, we plotted the mean responses of the cortex as a function of various underlying design configurations corresponding to various stimuli applied to awake and mildly anesthetized subjects.

It is necessary to introduce a replicated version of the underlying model to handle even the univariate situation, and we replace (7.8) by

$$y_{jt} = \sum_{r=-\infty}^{\infty} \beta'_r z_{j,t-r} + v_{jt} \quad (7.37)$$

for $j = 1, 2, \dots, N$ series, where we assume the vector of known deterministic inputs, $z_{jt} = (z_{jt1}, \dots, z_{jtq})'$, satisfies

$$\sum_{t=-\infty}^{\infty} |t| |z_{jtk}| < \infty$$

for $j = 1, \dots, N$ replicates of an underlying process involving $k = 1, \dots, q$ regression functions. The model can also be treated under the assumption that the deterministic function satisfy Grenander's conditions, as in Hannan (1970), but we do not need those conditions here and simply follow the approach in Shumway (1983, 1988).

It will sometimes be convenient in what follows to represent the model in matrix notation, writing (7.37) as

$$y_t = \sum_{r=-\infty}^{\infty} z_{t-r} \beta_r + v_t, \quad (7.38)$$

where $z_t = (z_{1t}, \dots, z_{Nt})'$ are the $N \times q$ matrices of independent inputs and y_t and v_t are the $N \times 1$ output and error vectors. The error vector $v_t = (v_{1t}, \dots, v_{Nt})'$ is

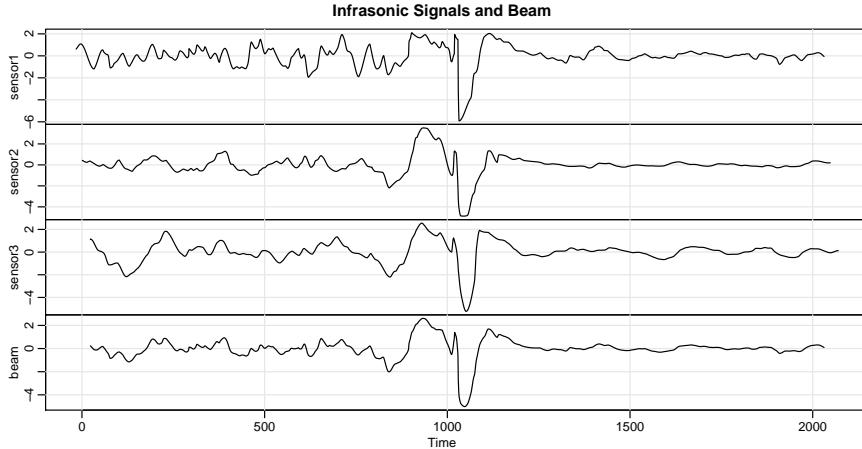


Fig. 7.6. Three series for a nuclear explosion detonated 25 km south of Christmas Island and the delayed average or beam. The time scale is 10 points per second.

assumed to be a multivariate, zero-mean, stationary, normal process with spectral matrix $f_v(\omega)I_N$ that is proportional to the $N \times N$ identity matrix. That is, we assume the error series v_{jt} are independently and identically distributed with spectral densities $f_v(\omega)$.

Example 7.2 An Infrasonic Signal from a Nuclear Explosion

Often, we will observe a common signal, say, β_t on an array of sensors, with the response at the j -th sensor denoted by y_{jt} , $j = 1, \dots, N$. For example, Figure 7.6 shows an infrasonic or low-frequency acoustic signal from a nuclear explosion, as observed on a small triangular array of $N = 3$ acoustic sensors. These signals appear at slightly different times. Because of the way signals propagate, a plane wave signal of this kind, from a given source, traveling at a given velocity, will arrive at elements in the array at predictable time delays. In the case of the infrasonic signal in Figure 7.6, the delays were approximated by computing the cross-correlation between elements and simply reading off the time delay corresponding to the maximum. For a detailed discussion of the statistical analysis of array signals, see Shumway et al. (1999).

A simple additive signal-plus-noise model of the form

$$y_{jt} = \beta_{t-\tau_j} + v_{jt} \quad (7.39)$$

can be assumed, where τ_j , $j = 1, 2, \dots, N$ are the time delays that determine the start point of the signal at each element of the array. The model (7.39) is written in the form (7.37) by letting $z_{jt} = \delta_{t-\tau_j}$, where $\delta_t = 1$ when $t = 0$ and is zero otherwise. In this case, we are interested in both the problem of detecting the presence of the signal and in estimating its waveform β_t . In this case, a plausible estimator of the waveform would be the unbiased *beam*, say,

$$\hat{\beta}_t = \frac{\sum_{j=1}^N y_{j,t+\tau_j}}{N}, \quad (7.40)$$

where time delays in this case were measured as $\tau_1 = 17$, $\tau_2 = 0$, and $\tau_3 = -22$ from the cross-correlation function. The bottom panel of Figure 7.6 shows the computed beam in this case, and the noise in the individual channels has been reduced and the essential characteristics of the common signal are retained in the average. The R code for this example is

```
attach(beamd)
tau    = rep(0,3)
u     = ccf(sensor1, sensor2, plot=FALSE)
tau[1] = u$lag[which.max(u$acf)] # 17
u     = ccf(sensor3, sensor2, plot=FALSE)
tau[3] = u$lag[which.max(u$acf)] # -22
Y = ts.union(lag(sensor1,tau[1]), lag(sensor2, tau[2]), lag(sensor3, tau[3]))
Y = ts.union(Y, rowMeans(Y))
colnames(Y) = c('sensor1', 'sensor2', 'sensor3', 'beam')
plot.ts(Y)
```

The above discussion and example serve to motivate a more detailed look at the estimation and detection problems in the case in which the input series z_{jt} are fixed and known. We consider the modifications needed for this case in the following sections.

ESTIMATION OF THE REGRESSION RELATION

Because the regression model (7.37) involves fixed functions, we may parallel the usual approach using the Gauss–Markov theorem to search for linear-filtered estimators of the form

$$\hat{\beta}_t = \sum_{j=1}^N \sum_{r=-\infty}^{\infty} h_{jr} y_{j,t-r}, \quad (7.41)$$

where $h_{jt} = (h_{jt1}, \dots, h_{jtq})'$ is a vector of filter coefficients, determined so the estimators are unbiased and have minimum variance. The equivalent matrix form is

$$\hat{\beta}_t = \sum_{r=-\infty}^{\infty} h_r y_{t-r}, \quad (7.42)$$

where $h_t = (h_{1t}, \dots, h_{Nt})$ is a $q \times N$ matrix of filter functions. The matrix form resembles the usual classical regression case and is more convenient for extending the Gauss–Markov Theorem to lagged regression. The unbiased condition is considered in Problem 7.6. It can be shown (see Shumway and Dean, 1968) that h_{js} can be taken as the Fourier transform of

$$H_j(\omega) = S_z^{-1}(\omega) \overline{Z_j(\omega)}, \quad (7.43)$$

where

$$Z_j(\omega) = \sum_{t=-\infty}^{\infty} z_{jt} e^{-2\pi i \omega t} \quad (7.44)$$

is the infinite Fourier transform of z_{jt} . The matrix

$$S_z(\omega) = \sum_{j=1}^N \overline{Z_j(\omega)} Z'_j(\omega) \quad (7.45)$$

can be written in the form

$$S_z(\omega) = Z^*(\omega) Z(\omega), \quad (7.46)$$

where the $N \times q$ matrix $Z(\omega)$ is defined by $Z(\omega) = (Z_1(\omega), \dots, Z_N(\omega))'$. In matrix notation, the Fourier transform of the optimal filter becomes

$$H(\omega) = S_z^{-1}(\omega) Z^*(\omega), \quad (7.47)$$

where $H(\omega) = (H_1(\omega), \dots, H_N(\omega))$ is the $q \times N$ matrix of frequency response functions. The optimal filter then becomes the Fourier transform

$$h_t = \int_{-1/2}^{1/2} H(\omega) e^{2\pi i \omega t} d\omega. \quad (7.48)$$

If the transform is not tractable to compute, an approximation analogous to (7.25) may be used.

Example 7.3 Estimation of the Infrasonic Signal in Example 7.2

We consider the problem of producing a best linearly filtered unbiased estimator for the infrasonic signal in Example 7.2. In this case, $q = 1$ and (7.44) becomes

$$Z_j(\omega) = \sum_{t=-\infty}^{\infty} \delta_{t-\tau_j} e^{-2\pi i \omega t} = e^{-2\pi i \omega \tau_j}$$

and $S_z(\omega) = N$. Hence, we have

$$H_j(\omega) = \frac{1}{N} e^{2\pi i \omega \tau_j}.$$

Using (7.48), we obtain $h_{jt} = \frac{1}{N} \delta(t + \tau_j)$. Substituting in (7.41), we obtain the best linear unbiased estimator as the beam, computed as in (7.40).

TESTS OF HYPOTHESES

We consider first testing the hypothesis that the complete vector β_t is zero, i.e., that the vector signal is absent. We develop a test at each frequency ω by taking single adjacent frequencies of the form $\omega_k = k/n$, as in the initial section. We may approximate the DFT of the observed vector in the model (7.37) using a representation of the form

$$Y_j(\omega_k) = B'(\omega_k) Z_j(\omega_k) + V_j(\omega_k) \quad (7.49)$$

for $j = 1, \dots, N$, where the error terms will be uncorrelated with common variance $f(\omega_k)$, the spectral density of the error term. The independent variables $Z_j(\omega_k)$ can

either be the infinite Fourier transform, or they can be approximated by the DFT. Hence, we can obtain the matrix version of a complex regression model, written in the form

$$Y(\omega_k) = Z(\omega_k)B(\omega_k) + V(\omega_k), \quad (7.50)$$

where the $N \times q$ matrix $Z(\omega_k)$ has been defined previously below (7.46) and $Y(\omega_k)$ and $V(\omega_k)$ are $N \times 1$ vectors with the error vector $V(\omega_k)$ having mean zero, with covariance matrix $f(\omega_k)I_N$. The usual regression arguments show that the maximum likelihood estimator for the regression coefficient will be

$$\hat{B}(\omega_k) = S_z^{-1}(\omega_k)s_{zy}(\omega_k), \quad (7.51)$$

where $S_z(\omega_k)$ is given by (7.46) and

$$s_{zy}(\omega_k) = Z^*(\omega_k)Y(\omega_k) = \sum_{j=1}^N \overline{Z_j(\omega_k)}Y_j(\omega_k). \quad (7.52)$$

Also, the maximum likelihood estimator for the error spectral matrix is proportional to

$$\begin{aligned} s_{y-z}^2(\omega_k) &= \sum_{j=1}^N |Y_j(\omega_k) - \hat{B}(\omega_k)'Z_j(\omega_k)|^2 \\ &= Y^*(\omega_k)Y(\omega_k) - Y^*(\omega_k)Z(\omega_k)[Z^*(\omega_k)Z(\omega_k)]^{-1}Z^*(\omega_k)Y(\omega_k) \\ &= s_y^2(\omega_k) - s_{zy}^*(\omega_k)S_z^{-1}(\omega_k)s_{zy}(\omega_k), \end{aligned} \quad (7.53)$$

where

$$s_y^2(\omega_k) = \sum_{j=1}^N |Y_j(\omega_k)|^2. \quad (7.54)$$

Under the null hypothesis that the regression coefficient $B(\omega_k) = 0$, the estimator for the error power is just $s_y^2(\omega_k)$. If smoothing is needed, we may replace the (7.53) and (7.54) by smoothed components over the frequencies $\omega_k + \ell/n$, for $\ell = -m, \dots, m$ and $L = 2m + 1$, close to ω . In that case, we obtain the regression and error spectral components as

$$\text{SSR}(\omega) = \sum_{\ell=-m}^m s_{zy}^*(\omega_k + \ell/n)S_z^{-1}(\omega_k + \ell/n)s_{zy}(\omega_k + \ell/n) \quad (7.55)$$

and

$$\text{SSE}(\omega) = \sum_{\ell=-m}^m s_{y-z}^2(\omega_k + \ell/n). \quad (7.56)$$

The F -statistic for testing no regression relation is

$$F_{2Lq, 2L(N-q)} = \frac{N-q}{q} \frac{\text{SSR}(\omega)}{\text{SSE}(\omega)}. \quad (7.57)$$

Table 7.2. Analysis of Power (ANOPOW) for Testing No Contribution from the Independent Series at Frequency ω in the Fixed Input Case

Source	Power	Degrees of Freedom
Regression	$\text{SSR}(\omega)$ (7.55)	$2Lq$
Error	$\text{SSE}(\omega)$ (7.56)	$2L(N - q)$
Total	$\text{SST}(\omega)$	$2LN$

The analysis of power pertaining to this situation appears in [Table 7.2](#).

In the fixed regression case, the partitioned hypothesis that is the analog of $\beta_{2t} = 0$ in [\(7.27\)](#) with x_{t1}, x_{t2} replaced by z_{t1}, z_{t2} . Here, we partition $S_z(\omega)$ into $q_i \times q_j$ ($i, j = 1, 2$) submatrices, say,

$$S_z(\omega_k) = \begin{pmatrix} S_{11}(\omega_k) & S_{12}(\omega_k) \\ S_{21}(\omega_k) & S_{22}(\omega_k) \end{pmatrix}, \quad (7.58)$$

and the cross-spectral vector into its $q_i \times 1$, for $i = 1, 2$, subvectors

$$s_{zy}(\omega_k) = \begin{pmatrix} s_{1y}(\omega_k) \\ s_{2y}(\omega_k) \end{pmatrix}. \quad (7.59)$$

Here, we test the hypothesis $\beta_{2t} = 0$ at frequency ω by comparing the residual power [\(7.53\)](#) under the full model with the residual power under the reduced model, given by

$$s_{y \cdot 1}^2(\omega_k) = s_y^2(\omega_k) - s_{1y}^*(\omega_k)S_{11}^{-1}(\omega_k)s_{1y}(\omega_k). \quad (7.60)$$

Again, it is desirable to add over adjacent frequencies with roughly comparable spectra so the regression and error power components can be taken as

$$\text{SSR}(\omega) = \sum_{\ell=-m}^m \left[s_{y \cdot 1}^2(\omega_k + \ell/n) - s_{y \cdot z}^2(\omega_k + \ell/n) \right] \quad (7.61)$$

and

$$\text{SSE}(\omega) = \sum_{\ell=-m}^m s_{y \cdot z}^2(\omega_k + \ell/n). \quad (7.62)$$

The information can again be summarized as in [Table 7.3](#), where the ratio of mean power regression and error components leads to the F -statistic

$$F_{2Lq_2, 2L(N-q)} = \frac{(N - q)}{q_2} \frac{\text{SSR}(\omega)}{\text{SSE}(\omega)}. \quad (7.63)$$

We illustrate the analysis of power procedure using the infrasonic signal detection procedure of [Example 7.2](#).

Table 7.3. Analysis of Power (ANOPOW) for Testing No Contribution from the Last q_2 Inputs in the Fixed Input Case

Source	Power	Degrees of Freedom
Regression	$\text{SSR}(\omega)$ (7.61)	$2Lq_2$
Error	$\text{SSE}(\omega)$ (7.62)	$2L(N - q)$
Total	$\text{SST}(\omega)$	$2L(N - q_1)$

Example 7.4 Detecting the Infrasonic Signal Using ANOPOW

We consider the problem of detecting the common signal for the three infrasonic series observing the common signal, as shown in Figure 7.4. The presence of the signal is obvious in the waveforms shown, so the test here mainly confirms the statistical significance and isolates the frequencies containing the strongest signal components. Each series contained $n = 2048$ points, sampled at 10 points per second. We use the model in (7.39) so $Z_j(\omega) = e^{-2\pi i \omega \tau_j}$ and $S_z(\omega) = N$ as in Example 7.3, with $s_{zy}(\omega_k)$ given as

$$s_{zy}(\omega_k) = \sum_{j=1}^N e^{2\pi i \omega \tau_j} Y_j(\omega_k),$$

using (7.45) and (7.52). The above expression can be interpreted as being proportional to the weighted mean or *beam*, computed in frequency, and we introduce the notation

$$B_w(\omega_k) = \frac{1}{N} \sum_{j=1}^N e^{2\pi i \omega \tau_j} Y_j(\omega_k) \quad (7.64)$$

for that term. Substituting for the power components in Table 7.3 yields

$$s_{zy}^*(\omega_k) S_z^{-1}(\omega_k) s_{zy}(\omega_k) = N |B_w(\omega_k)|^2$$

and

$$s_{y-z}^2(\omega_k) = \sum_{j=1}^N |Y_j(\omega_k) - B_w(\omega_k)|^2 = \sum_{j=1}^N |Y_j(\omega_k)|^2 - N |B_w(\omega_k)|^2$$

for the regression signal and error components, respectively. Because only three elements in the array and a reasonable number of points in time exist, it seems advisable to employ some smoothing over frequency to obtain additional degrees of freedom. In this case, $L = 9$, yielding $2(9) = 18$ and $2(9)(3 - 1) = 36$ degrees of freedom for the numerator and denominator of the F -statistic (7.57). The top of Figure 7.7 shows the analysis of power components due to error and the total power. The power is maximum at about .002 cycles per point or about .02 cycles per second. The F -statistic is compared with the .001 FDR and the corresponding null significance in the bottom panel and has the strongest detection at about .02 cycles

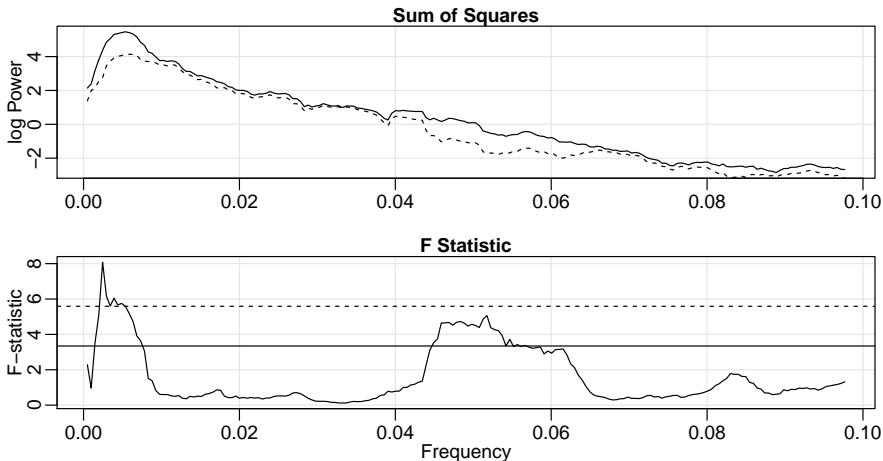


Fig. 7.7. Analysis of power for infrasound array on a log scale (top panel) with $SST(\omega)$ shown as a solid line and $SSE(\omega)$ as a dashed line. The F -statistics (bottom panel) showing detections with the dashed line based on an FDR level of .001 and the solid line corresponding null F quantile.

per second. Little power of consequence appears to exist elsewhere, however, there is some marginally significant signal power near the .5 cycles per second frequency band.

The R code for this example is as follows.

```

attach(beamd)
L      = 9; fdr = .001; N = 3
Y      = cbind(beamd, beam=rowMeans(beamd) )
n      = nextn(nrow(Y))
Y.fft = mvfft(as.ts(Y))/sqrt(n)
Df    = Y.fft[,1:3] # fft of the data
Bf    = Y.fft[,4]   # beam fft
ssr   = N*Re(Bf*Conj(Bf))           # raw signal spectrum
sse   = Re(rowSums(Df*Conj(Df))) - ssr # raw error spectrum
# Smooth
SSE   = filter(sse, sides=2, filter=rep(1/L,L), circular=TRUE)
SSR   = filter(ssr, sides=2, filter=rep(1/L,L), circular=TRUE)
SST   = SSE + SSR
par(mfrow=c(2,1), mar=c(4,4,2,1)+.1)
Fr   = 0:(n-1)/n      # the fundamental frequencies
nFr  = 1:200          # number of freqs to plot
plot(Fr[nFr], SST[nFr], type="l", ylab="log Power", xlab="", main="Sum of
                 Squares", log="y")
lines(Fr[nFr], SSE[nFr], type="l", lty=2)
eF   = (N-1)*SSR/SSE; df1 = 2*L; df2 = 2*L*(N-1)
pvals = pf(eF, df1, df2, lower=FALSE) # p values for FDR
pID  = FDR(pvals, fdr); Fq = qf(1-fdr, df1, df2)
plot(Fr[nFr], eF[nFr], type="l", ylab="F-statistic", xlab="Frequency",
     main="F Statistic")
abline(h=c(Fq, eF[pID]), lty=1:2)

```

Although there are examples of detecting multiple regression functions of the general type considered above (see, for example, Shumway, 1983), we do not consider additional examples of partitioning in the fixed input case here. The reason is that several examples exist in the section on designed experiments that illustrate the partitioned approach.

7.5 Random Coefficient Regression

The lagged regression models considered so far have assumed the input process is either stochastic or fixed and the components of the vector of regression function β_t are fixed and unknown parameters to be estimated. There are many cases in time series analysis in which it is more natural to regard the regression vector as an unknown stochastic signal. For example, we have studied the state-space model in [Chapter 6](#), where the state equation can be considered as involving a random parameter vector that is essentially a multivariate autoregressive process. In [Section 4.8](#), we considered estimating the univariate regression function β_t as a signal extraction problem.

In this section, we consider a *random coefficient regression model* of (7.38) in the equivalent form

$$y_t = \sum_{r=-\infty}^{\infty} z_{t-r} \beta_r + v_t, \quad (7.65)$$

where $y_t = (y_{1t}, \dots, y_{Nt})'$ is the $N \times 1$ response vector and $z_t = (z_{1t}, \dots, z_{Nt})'$ are the $N \times q$ matrices containing the fixed input processes. Here, the components of the $q \times 1$ regression vector β_t are zero-mean, uncorrelated, stationary series with common spectral matrix $f_\beta(\omega)I_q$ and the error series v_t have zero-means and spectral matrix $f_v(\omega)I_N$, where I_N is the $N \times N$ identity matrix. Then, defining the $N \times q$ matrix $Z(\omega) = (Z_1(\omega), Z_2(\omega), \dots, Z_N(\omega))'$ of Fourier transforms of z_t , as in (7.44), it is easy to show the spectral matrix of the response vector y_t is given by

$$f_y(\omega) = f_\beta(\omega)Z(\omega)Z^*(\omega) + f_v(\omega)I_N. \quad (7.66)$$

The regression model with a stochastic stationary signal component is a general version of the simple additive noise model

$$y_t = \beta_t + v_t,$$

considered by Wiener (1949) and Kolmogorov (1941), who derived the minimum mean squared error estimators for β_t , as in [Section 4.8](#). The more general multivariate version (7.65) represents the series as a convolution of the signal vector β_t and a known set of vector input series contained in the matrix z_t . Restricting the covariance matrices of signal and noise to diagonal form is consistent with what is done in statistics using random effects models, which we consider here in a later section. The problem of estimating the regression function β_t is often called *deconvolution* in the engineering and geophysical literature.

ESTIMATION OF THE REGRESSION RELATION

The regression function β_t can be estimated by a general filter of the form (7.42), where we write that estimator in matrix form

$$\hat{\beta}_t = \sum_{r=-\infty}^{\infty} h_r y_{t-r}, \quad (7.67)$$

where $h_t = (h_{1t}, \dots, h_{Nt})$, and apply the orthogonality principle, as in Section 4.8. A generalization of the argument in that section (see Problem 7.7) leads to the estimator

$$H(\omega) = [S_z(\omega) + \theta(\omega)I_q]^{-1}Z^*(\omega) \quad (7.68)$$

for the Fourier transform of the minimum mean-squared error filter, where the parameter

$$\theta(\omega) = \frac{f_v(\omega)}{f_\beta(\omega)} \quad (7.69)$$

is the inverse of the signal-to-noise ratio. It is clear from the frequency domain version of the linear model (7.50), the comparable version of the estimator (7.51) can be written as

$$\hat{B}(\omega) = [S_z(\omega) + \theta(\omega)I_q]^{-1}s_{zy}(\omega). \quad (7.70)$$

This version exhibits the estimator in the stochastic regressor case as the usual estimator, with a *ridge correction*, $\theta(\omega)$, that is proportional to the inverse of the signal-to-noise ratio.

The mean-squared covariance of the estimator is shown to be

$$E[(\hat{B} - B)(\hat{B} - B)^*] = f_v(\omega)[S_z(\omega) + \theta(\omega)I_q]^{-1}, \quad (7.71)$$

which again exhibits the close connection between this case and the variance of the estimator (7.51), which can be shown to be $f_v(\omega)S_z^{-1}(\omega)$.

Example 7.5 Estimating the Random Infrasonic Signal

In Example 7.4, we have already determined the components needed in (7.68) and (7.69) to obtain the estimators for the random signal. The Fourier transform of the optimum filter at series j has the form

$$H_j(\omega) = \frac{e^{2\pi i \omega \tau_j}}{N + \theta(\omega)} \quad (7.72)$$

with the mean-squared error given by $f_v(\omega)/[N + \theta(\omega)]$ from (7.71). The net effect of applying the filters will be the same as filtering the beam with the frequency response function

$$H_0(\omega) = \frac{N}{N + \theta(\omega)} = \frac{Nf_\beta(\omega)}{f_v(\omega) + Nf_\beta(\omega)}, \quad (7.73)$$

where the last form is more convenient in cases in which portions of the signal spectrum are essentially zero.

The optimal filters h_t have frequency response functions that depend on the signal spectrum $f_\beta(\omega)$ and noise spectrum $f_v(\omega)$, so we will need estimators for these parameters to apply the optimal filters. Sometimes, there will be values, suggested from experience, for the signal-to-noise ratio $1/\theta(\omega)$ as a function of frequency. The analogy between the model here and the usual variance components model in statistics, however, suggests we try an approach along those lines as in the next section.

DETECTION AND PARAMETER ESTIMATION

The analogy to the usual variance components situation suggests looking at the regression and error components of Table 7.2 under the stochastic signal assumptions. We consider the components of (7.55) and (7.56) at a single frequency ω_k . In order to estimate the spectral components $f_\beta(\omega)$ and $f_v(\omega)$, we reconsider the linear model (7.50) under the assumption that $B(\omega_k)$ is a random process with spectral matrix $f_\beta(\omega_k)I_q$. Then, the spectral matrix of the observed process is (7.66), evaluated at frequency ω_k .

Consider first the component of the regression power, defined as

$$\begin{aligned}\text{SSR}(\omega_k) &= s_{zy}^*(\omega_k)S_z^{-1}(\omega_k)s_{zy}(\omega_k) \\ &= Y^*(\omega_k)Z(\omega_k)S_z^{-1}(\omega_k)Z^*(\omega_k)Y(\omega_k).\end{aligned}$$

A computation shows

$$E[\text{SSR}(\omega_k)] = f_\beta(\omega_k) \text{tr}\{S_z(\omega_k)\} + qf_v(\omega_k),$$

where tr denotes the trace of a matrix. If we can find a set of frequencies of the form $\omega_k + \ell/n$, where the spectra and the Fourier transforms $S_z(\omega_k + \ell/n) \approx S_z(\omega)$ are relatively constant, the expectation of the averaged values in (7.55) yields

$$E[\text{SSR}(\omega)] = Lf_\beta(\omega)\text{tr}[S_z(\omega)] + Lqf_v(\omega). \quad (7.74)$$

A similar computation establishes

$$E[\text{SSE}(\omega)] = L(N - q)f_v(\omega). \quad (7.75)$$

We may obtain an approximately unbiased estimator for the spectra $f_v(\omega)$ and $f_\beta(\omega)$ by replacing the expected power components by their values and solving (7.74) and (7.75).

7.6 Analysis of Designed Experiments

An important special case (see Brillinger, 1973, 1980) of the regression model (7.49) occurs when the regression (7.38) is of the form

$$y_t = z\beta_t + v_t, \quad (7.76)$$

where $z = (z_1, z_2, \dots, z_N)'$ is a matrix that determines what is observed by the j -th series; i.e.,

$$y_{jt} = z_j' \beta_t + v_{jt}. \quad (7.77)$$

In this case, the the matrix z of independent variables is constant and we will have the frequency domain model.

$$Y(\omega_k) = ZB(\omega_k) + V(\omega_k) \quad (7.78)$$

corresponding to (7.50), where the matrix $Z(\omega_k)$ was a function of frequency ω_k . The matrix is purely real, in this case, but the equations (7.51)–(7.57) can be applied with $Z(\omega_k)$ replaced by the constant matrix Z .

EQUALITY OF MEANS

A typical general problem that we encounter in analyzing real data is a simple *equality of means test* in which there might be a collection of time series y_{ijt} , $i = 1, \dots, I$, $j = 1, \dots, N_i$, belonging to I possible groups, with N_i series in group i . To test equality of means, we may write the regression model in the form

$$y_{ijt} = \mu_t + \alpha_{it} + v_{ijt}, \quad (7.79)$$

where μ_t denotes the overall mean and α_{it} denotes the effect of the i -th group at time t and we require that $\sum_i \alpha_{it} = 0$ for all t . In this case, the full model can be written in the general regression notation as

$$y_{ijt} = z_{ij}' \beta_t + v_{ijt}$$

where

$$\beta_t = (\mu_t, \alpha_{1t}, \alpha_{2t}, \dots, \alpha_{I-1,t})'$$

denotes the regression vector, subject to the constraint. The reduced model becomes

$$y_{ijt} = \mu_t + v_{ijt} \quad (7.80)$$

under the assumption that the group means are equal. In the full model, there are I possible values for the $I \times 1$ design vectors z_{ij} ; the first component is always one for the mean, and the rest have a one in the i -th position for $i = 1, \dots, I - 1$ and zeros elsewhere. The vectors for the last group take the value -1 for $i = 2, 3, \dots, I - 1$. Under the reduced model, each z_{ij} is a single column of ones. The rest of the analysis follows the approach summarized in (7.51)–(7.57). In this particular case, the power components in Table 7.3 (before smoothing) simplify to

$$\text{SSR}(\omega_k) = \sum_{i=1}^I \sum_{j=1}^{N_i} |Y_{i\cdot}(\omega_k) - Y_{\cdot\cdot}(\omega_k)|^2 \quad (7.81)$$

and

$$\text{SSE}(\omega_k) = \sum_{i=1}^I \sum_{j=1}^{N_i} |Y_{ij}(\omega_k) - Y_{i\cdot}(\omega_k)|^2, \quad (7.82)$$

which are analogous to the usual sums of squares in analysis of variance. Note that a dot (\cdot) stands for a mean, taken over the appropriate subscript, so the regression power component $\text{SSR}(\omega_k)$ is basically the power in the residuals of the group means from the overall mean and the error power component $\text{SSE}(\omega_k)$ reflects the departures of the group means from the original data values. Smoothing each component over L frequencies leads to the usual F -statistic (7.63) with $2L(I - 1)$ and $2L(\sum_i N_i - I)$ degrees of freedom at each frequency ω of interest.

Example 7.6 Means Test for the fMRI Data

Figure 7.1 showed the mean responses of subjects to various levels of periodic stimulation while awake and while under anesthesia, as collected in a pain perception experiment of Antognini et al. (1997). Three types of periodic stimuli were presented to awake and anesthetized subjects, namely, brushing, heat, and shock. The periodicity was introduced by applying the stimuli, brushing, heat, and shocks in on-off sequences lasting 32 seconds each and the sampling rate was one point every two seconds. The blood oxygenation level (BOLD) signal intensity (Ogawa et al., 1990) was measured at nine locations in the brain. Areas of activation were determined using a technique first described by Bandettini et al. (1993). The specific locations of the brain where the signal was measured were Cortex 1: Primary Somatosensory, Contralateral, Cortex 2: Primary Somatosensory, Ipsilateral, Cortex 3: Secondary Somatosensory, Contralateral, Cortex 4: Secondary Somatosensory, Ipsilateral, Caudate, Thalamus 1: Contralateral, Thalamus 2: Ipsilateral, Cerebellum 1: Contralateral and Cerebellum 2: Ipsilateral. Figure 7.1 shows the mean response of subjects at Cortex 1 for each of the six treatment combinations, 1: Awake-Brush (5 subjects), 2: Awake-Heat (4 subjects), 3: Awake-Shock (5 subjects), 4: Low-Brush (3 subjects), 5: Low-Heat (5 subjects), and 6: Low-Shock (4 subjects). The objective of this first analysis is to test equality of these six group means, paying special attention to the 64-second period band (1/64 cycles per second) expected from the periodic driving stimuli. Because a test of equality is needed at each of the nine brain locations, we took $\alpha = .001$ to control for the overall error rate. Figure 7.8 shows F -statistics, computed from (7.63), with $L = 3$, and we see substantial signals for the four cortex locations and for the second cerebellum trace, but the effects are nonsignificant in the caudate and thalamus regions. Hence, we will retain the four cortex locations and the second cerebellum location for further analysis.

The R code for this example is as follows.

```

n      = 128          # length of series
n.freq = 1 + n/2     # number of frequencies
Fr    = (0:(n.freq-1))/n # the frequencies
N     = c(5,4,5,3,5,4) # number of series for each cell
n.subject = sum(N)    # number of subjects (26)
n.trt   = 6           # number of treatments
L      = 3           # for smoothing
num.df = 2*L*(n.trt-1) # df for F test

```

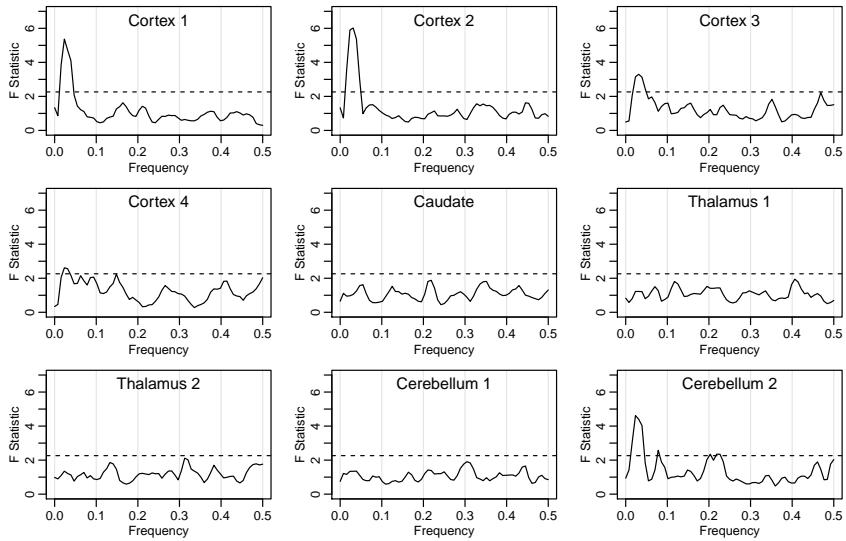


Fig. 7.8. Frequency-dependent equality of means tests for fMRI data at 9 brain locations. $L = 3$ and critical value $F_{.001}(30, 120) = 2.26$.

```

den.df      = 2*L*(n.subject-n.trt)
# Design Matrix (Z):
Z1  = outer(rep(1,N[1]), c(1,1,0,0,0,0))
Z2  = outer(rep(1,N[2]), c(1,0,1,0,0,0))
Z3  = outer(rep(1,N[3]), c(1,0,0,1,0,0))
Z4  = outer(rep(1,N[4]), c(1,0,0,0,1,0))
Z5  = outer(rep(1,N[5]), c(1,0,0,0,0,1))
Z6  = outer(rep(1,N[6]), c(1,-1,-1,-1,-1,-1))
Z  = rbind(Z1, Z2, Z3, Z4, Z5, Z6)
ZZ  = t(Z)%*%Z
SSEF <- rep(NA, n) -> SSER
HatF = Z%*%solve(ZZ, t(Z))
HatR = Z[,1]%*%t(Z[,1])/ZZ[1,1]
par(mfrow=c(3,3), mar=c(3.5,4,0,0), oma=c(0,0,2,2), mgp = c(1.6,.6,0))
loc.name = c("Cortex 1","Cortex 2","Cortex 3","Cortex 4","Caudate","Thalamus
    1","Thalamus 2","Cerebellum 1","Cerebellum 2")
for(Loc in 1:9) {
  i = n.trt*(Loc-1)
  Y = cbind(fmri[[i+1]], fmri[[i+2]], fmri[[i+3]], fmri[[i+4]], fmri[[i+5]],
            fmri[[i+6]])
  Y = mvfft(spec.taper(Y, p=.5))/sqrt(n)
  Y = t(Y)          # Y is now 26 x 128 FFTs
# Calculation of Error Spectra
for (k in 1:n) {
  SSY    = Re(Conj(t(Y[,k]))%*%Y[,k])
  SSReg = Re(Conj(t(Y[,k]))%*%HatF%*%Y[,k])
  SSEF[k] = SSY - SSReg
  SSReg = Re(Conj(t(Y[,k]))%*%HatR%*%Y[,k])
}

```

```

SSER[k] = SSY - SSReg }
# Smooth
sSSEF   = filter(SSEF, rep(1/L, L), circular = TRUE)
sSSER   = filter(SSER, rep(1/L, L), circular = TRUE)
eF      = (den.df/num.df)*(sSSER-sSSEF)/sSSEF
plot(Fr, eF[1:n.freq], type="l", xlab="Frequency", ylab="F Statistic",
      ylim=c(0,7))
abline(h=qf(.999, num.df, den.df), lty=2)
text(.25, 6.5, loc.name[Loc], cex=1.2) }

```

AN ANALYSIS OF VARIANCE MODEL

The arrangement of treatments for the fMRI data in Figure 7.1 suggests more information might be available than was obtained from the simple equality of means test. Separate effects caused by state of consciousness as well as the separate treatments brush, heat, and shock might exist. The reduced signal present in the low shock mean suggests a possible interaction between the treatments and level of consciousness. The arrangement in the classical two-way table suggests looking at the analog of the two factor analysis of variance as a function of frequency. In this case, we would obtain a different version of the regression model (7.79) of the form

$$y_{ijkt} = \mu_t + \alpha_{it} + \beta_{jt} + \gamma_{ijt} + v_{ijkt} \quad (7.83)$$

for the k -th individual undergoing the i -th level of some factor A and the j -th level of some other factor B, $i = 1, \dots, I$, $j = 1, \dots, J$, $k = 1, \dots, n_{ij}$. The number of individuals in each cell can be different, as for the fMRI data in the next example. In the above model, we assume the response can be modeled as the sum of a mean, μ_t , a *row effect* (type of stimulus), α_{it} , a *column effect* (level of consciousness), β_{jt} and an *interaction*, γ_{ijt} , with the usual restrictions

$$\sum_i \alpha_{it} = \sum_j \beta_{jt} = \sum_i \gamma_{ijt} = \sum_j \gamma_{ijt} = 0$$

required for a full rank design matrix Z in the overall regression model (7.78). If the number of observations in each cell were the same, the usual simple analogous version of the power components (7.81) and (7.82) would exist for testing various hypotheses. In the case of (7.83), we are interested in testing hypotheses obtained by dropping one set of terms at a time out of (7.83), so an A factor (testing $\alpha_{it} = 0$), a B factor ($\beta_{jt} = 0$), and an interaction term ($\gamma_{ijt} = 0$) will appear as components in the analysis of power. Because of the unequal numbers of observations in each cell, we often put the model in the form of the regression model (7.76)–(7.78).

Example 7.7 Analysis of Power Tests for the fMRI Series

For the fMRI data given as the means in Figure 7.1, a model of the form (7.83) is plausible and will yield more detailed information than the simple equality of means test described earlier. The results of that test, shown in Figure 7.8, were that the means were different for the four cortex locations and for the second cerebellum

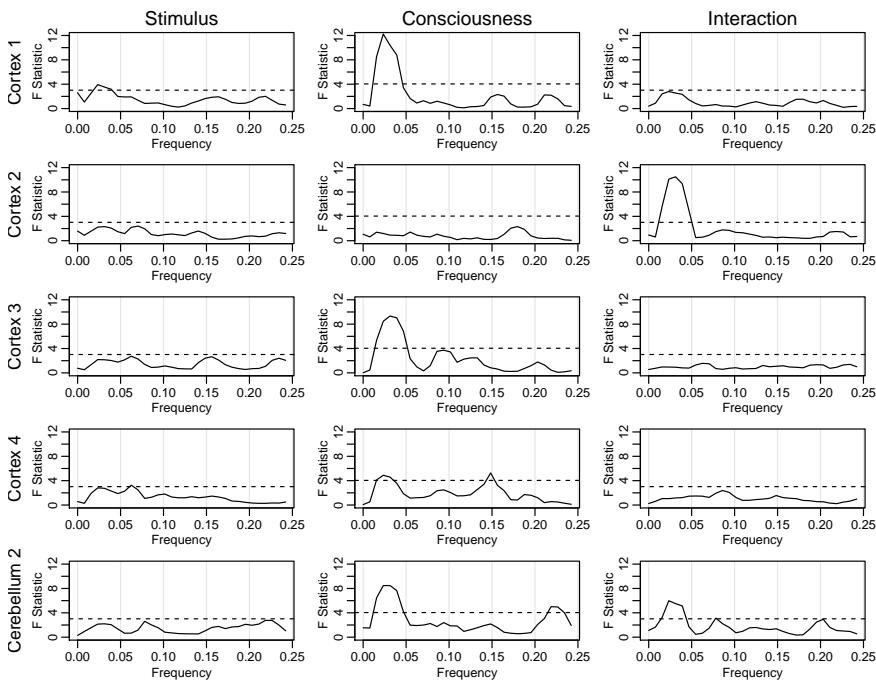


Fig. 7.9. Analysis of power for fMRI data at five locations, $L = 3$ and critical values $F_{0.001}(6, 120) = 4.04$ for stimulus and $F_{0.001}(12, 120) = 3.02$ for consciousness and interaction.

location. We may examine these differences further by testing whether the mean differences are because of the nature of the stimulus or the consciousness level, or perhaps due to an interaction between the two factors. Unequal numbers of observations exist in the cells that contributed the means in Figure 7.1. For the regression vector,

$$(\mu_t, \alpha_{1t}, \alpha_{2t}, \beta_{1t}, \gamma_{11t}, \gamma_{21t})',$$

the rows of the design matrix are as specified in Table 7.4. Note the restrictions given above for the parameters.

The results of testing the three hypotheses are shown in Figure 7.9 for the four cortex locations and the cerebellum, the components that showed some significant differences in the means in Figure 7.8. Again, the regression power components were smoothed over $L = 3$ frequencies. Appealing to the ANOPOW results summarized in Table 7.3 for each of the subhypotheses, $q_2 = 1$ when the stimulus effect is dropped, and $q_2 = 2$ when either the consciousness effect or the interaction terms are dropped. Hence, $2Lq_2 = 6, 12$ for the two cases, with $N = \sum_{ij} n_{ij} = 26$ total observations. Here, the state of consciousness (Awake, Sedated) has the major effect at the signal frequency. The level of stimulus was less significant at the signal frequency. A significant interaction occurred, however, at the ipsilateral component of the primary somatosensory cortex location.

The R code for this example is similar to [Example 7.6](#).

```

n      = 128
n.freq = 1 + n/2
Fr     = (0:(n.freq-1))/n
nFr   = 1:(n.freq/2)
N      = c(5,4,5,3,5,4)
n.subject = sum(N)
n.para  = 6           # number of parameters
L      = 3           # for smoothing
df.stm = 2*L*(3-1)    # stimulus (3 levels: Brush,Heat,Shock)
df.con = 2*L*(2-1)    # conscious (2 levels: Awake,Sedated)
df.int = 2*L*(3-1)*(2-1) # interaction
den.df = 2*L*(n.subject-n.para) # df for full model
# Design Matrix:          mu  a1  a2  b  g1  g2
Z1  = outer(rep(1,N[1]), c(1, 1, 0, 1, 1, 0))
Z2  = outer(rep(1,N[2]), c(1, 0, 1, 1, 0, 1))
Z3  = outer(rep(1,N[3]), c(1, -1, -1, 1, -1, -1))
Z4  = outer(rep(1,N[4]), c(1, 1, 0, -1, -1, 0))
Z5  = outer(rep(1,N[5]), c(1, 0, 1, -1, 0, -1))
Z6  = outer(rep(1,N[6]), c(1, -1, -1, -1, 1, 1))
Z  = rbind(Z1, Z2, Z3, Z4, Z5, Z6)
ZZ = t(Z)%*%Z
rep(NA, n)-> SSEF-> SSE.stm-> SSE.con-> SSE.int
HatF  = ZZ%*%solve(ZZ,t(Z))
Hat.stm = Z[,-(2:3)]%*%solve(ZZ[-(2:3),-(2:3)], t(Z[,-(2:3)]))
Hat.con = Z[,-4]%*%solve(ZZ[-4,-4], t(Z[,-4]))
Hat.int = Z[,-(5:6)]%*%solve(ZZ[-(5:6),-(5:6)], t(Z[,-(5:6)]))
par(mfrow=c(5,3), mar=c(3.5,4,0,0), oma=c(0,0,2,2), mgp = c(1.6,.6,0))
loc.name = c("Cortex 1","Cortex 2","Cortex 3","Cortex 4","Caudate", "Thalamus 1","Thalamus 2","Cerebellum 1","Cerebellum 2")
for(Loc in c(1:4,9)) { # only Loc 1 to 4 and 9 used
  i = 6*(Loc-1)
  Y = cbind(fmri[[i+1]], fmri[[i+2]], fmri[[i+3]], fmri[[i+4]], fmri[[i+5]], fmri[[i+6]])
  Y = mvfft(spec.taper(Y, p=.5))/sqrt(n); Y = t(Y)
  for (k in 1:n) {
    SSY      = Re(Conj(t(Y[,k]))%*%Y[,k])
    SSReg    = Re(Conj(t(Y[,k]))%*%HatF%*%Y[,k])
    SSEF[k]  = SSY - SSReg
    SSReg    = Re(Conj(t(Y[,k]))%*%Hat.stm%*%Y[,k])
    SSE.stm[k] = SSY-SSReg
    SSReg    = Re(Conj(t(Y[,k]))%*%Hat.con%*%Y[,k])
    SSE.con[k] = SSY-SSReg
    SSReg    = Re(Conj(t(Y[,k]))%*%Hat.int%*%Y[,k])
    SSE.int[k] = SSY-SSReg
  }
  # Smooth
  sSSEF  = filter(SSEF, rep(1/L, L), circular = TRUE)
  sSSE.stm = filter(SSE.stm, rep(1/L, L), circular = TRUE)
  sSSE.con = filter(SSE.con, rep(1/L, L), circular = TRUE)
  sSSE.int = filter(SSE.int, rep(1/L, L), circular = TRUE)
  eF.stm = (den.df/df.stm)*(sSSE.stm-sSSEF)/sSSEF
  eF.con = (den.df/df.con)*(sSSE.con-sSSEF)/sSSEF
  eF.int = (den.df/df.int)*(sSSE.int-sSSEF)/sSSEF
  plot(Fr[nFr],eF.stm[nFr], type="l", xlab="Frequency", ylab="F Statistic",
       ylim=c(0,12))
  abline(h=qf(.999, df.stm, den.df), lty=2)
}

```

Table 7.4. Rows of the Design Matrix for *Example 7.7*

	Awake						Low Anesthesia							
Brush	1	1	0	1	1	0	(5)	1	1	0	-1	-1	0	(3)
Heat	1	0	1	1	0	1	(4)	1	0	1	-1	0	-1	(5)
Shock	1	-1	-1	1	-1	-1	(5)	1	-1	-1	-1	1	1	(4)

Number of Observations per Cell in Parentheses

```

if(Loc==1) mtext("Stimulus", side=3, line=.3, cex=1)
mtext(loc.name[Loc], side=2, line=3, cex=.9)
plot(Fr[nFr], eF.con[nFr], type="l", xlab="Frequency", ylab="F Statistic",
      ylim=c(0,12))
abline(h=qf(.999, df.con, den.df), lty=2)
if(Loc==1) mtext("Consciousness", side=3, line=.3, cex=1)
plot(Fr[nFr], eF.int[nFr], type="l", xlab="Frequency", ylab="F Statistic",
      ylim=c(0,12))
abline(h=qf(.999, df.int, den.df), lty=2)
if(Loc==1) mtext("Interaction", side=3, line= .3, cex=1)    }

```

SIMULTANEOUS INFERENCE

In the previous examples involving the fMRI data, it would be helpful to focus on the components that contributed most to the rejection of the equal means hypothesis. One way to accomplish this is to develop a test for the significance of an arbitrary *linear compound* of the form

$$\Psi(\omega_k) = A^*(\omega_k)B(\omega_k), \quad (7.84)$$

where the components of the vector $A(\omega_k) = (A_1(\omega_k), A_2(\omega_k), \dots, A_q(\omega_k))'$ are chosen in such a way as to isolate particular linear functions of parameters in the regression vector $B(\omega_k)$ in the regression model (7.78). This argument suggests developing a test of the hypothesis $\Psi(\omega_k) = 0$ for *all possible* values of the linear coefficients in the compound (7.84) as is done in the conventional analysis of variance approach (see, for example, Scheffé, 1959).

Recalling the material involving the regression models of the form (7.50), the linear compound (7.84) can be estimated by

$$\hat{\Psi}(\omega_k) = A^*(\omega_k)\hat{B}(\omega_k), \quad (7.85)$$

where $\hat{B}(\omega_k)$ is the estimated vector of regression coefficients given by (7.51) and independent of the error spectrum $s_{y\cdot z}^2(\omega_k)$ in (7.53). It is possible to show the maximum of the ratio

$$F(A) = \frac{N - q}{q} \frac{|\hat{\Psi}(\omega_k) - \Psi(\omega_k)|^2}{s_{y\cdot z}^2(\omega_k)Q(A)}, \quad (7.86)$$

where

$$Q(A) = A^*(\omega_k)S_z^{-1}(\omega_k)A(\omega_k) \quad (7.87)$$

is bounded by a statistic that has an F -distribution with $2q$ and $2(N - q)$ degrees of freedom. Testing the hypothesis that the compound has a particular value, usually $\Psi(\omega_k) = 0$, then proceeds naturally, by comparing the statistic (7.86) evaluated at the hypothesized value with the α level point on an $F_{2q, 2(N-q)}$ distribution. We can choose an infinite number of compounds of the form (7.84) and the test will still be valid at level α . As before, arguing the error spectrum is relatively constant over a band enables us to smooth the numerator and denominator of (7.86) separately over L frequencies so distribution involving the smooth components is $F_{2Lq, 2L(N-q)}$.

Example 7.8 Simultaneous Inference for the fMRI Series

As an example, consider the previous tests for significance of the fMRI factors, in which we have indicated the primary effects are among the stimuli but have not investigated which of the stimuli, heat, brushing, or shock, had the most effect. To analyze this further, consider the means model (7.79) and a 6×1 contrast vector of the form

$$\hat{\Psi} = A^*(\omega_k) \hat{B}(\omega_k) = \sum_{i=1}^6 A_i^*(\omega_k) Y_i(\omega_k), \quad (7.88)$$

where the means are easily shown to be the regression coefficients in this particular case. In this case, the means are ordered by columns; the first three means are the three levels of stimuli for the awake state, and the last three means are the levels for the anesthetized state. In this special case, the denominator terms are

$$Q = \sum_{i=1}^6 \frac{|A_i(\omega_k)|^2}{N_i}, \quad (7.89)$$

with $SSE(\omega_k)$ available in (7.82). In order to evaluate the effect of a particular stimulus, like brushing over the two levels of consciousness, we may take $A_1(\omega_k) = A_4(\omega_k) = 1$ for the two brush levels and $A(\omega_k) = 0$ zero otherwise. From Figure 7.10, we see that, at the first and third cortex locations, brush and heat are both significant, whereas the fourth cortex shows only brush and the second cerebellum shows only heat. Shock appears to be transmitted relatively weakly, when averaged over the awake and mildly anesthetized states.

The R code for this example is as follows.

```
n = 128; n.freq = 1 + n/2
Fr = (0:(n.freq-1))/n; nFr = 1:(n.freq/2)
N = c(5,4,5,3,5,4); n.subject = sum(N); L = 3
# Design Matrix
Z1 = outer(rep(1,N[1]), c(1,0,0,0,0,0))
Z2 = outer(rep(1,N[2]), c(0,1,0,0,0,0))
Z3 = outer(rep(1,N[3]), c(0,0,1,0,0,0))
Z4 = outer(rep(1,N[4]), c(0,0,0,1,0,0))
Z5 = outer(rep(1,N[5]), c(0,0,0,0,1,0))
Z6 = outer(rep(1,N[6]), c(0,0,0,0,0,1))
Z = rbind(Z1, Z2, Z3, Z4, Z5, Z6); ZZ = t(Z)%*%Z
# Contrasts: 6 by 3
A = rbind(diag(1,3), diag(1,3))
nq = nrow(A); num.df = 2*L*nq; den.df = 2*L*(n.subject-nq)
```

```

HatF = Z%*%solve(ZZ, t(Z)) # full model
rep(NA, n)-> SSEF; eF = matrix(0, n, 3)
par(mfrow=c(5, 3), mar=c(3.5, 4, 0, 0), oma=c(0, 0, 2, 2), mgp = c(1.6, .6, 0))
loc.name = c("Cortex 1", "Cortex 2", "Cortex 3", "Cortex 4", "Caudate", "
    Thalamus 1", "Thalamus 2", "Cerebellum 1", "Cerebellum 2")
cond.name = c("Brush", "Heat", "Shock")
for(Loc in c(1:4, 9)) {
  i = 6*(Loc-1)
  Y = cbind(fmri[[i+1]], fmri[[i+2]], fmri[[i+3]], fmri[[i+4]], fmri[[i+5]],
    fmri[[i+6]])
  Y = mvfft(spec.taper(Y, p=.5))/sqrt(n); Y = t(Y)
  for (cond in 1:3){
    Q = t(A[,cond])%*%solve(ZZ, A[,cond])
    HR = A[,cond]%*%solve(ZZ, t(Z))
    for (k in 1:n){
      SSY     = Re(Conj(t(Y[,k]))%*%Y[,k])
      SSReg  = Re(Conj(t(Y[,k]))%*%HatF%*%Y[,k])
      SSEF[k] = (SSY-SSReg)*Q
      SSReg  = HR%*%Y[,k]
      SSER[k] = Re(SSReg*Conj(SSReg))  }
    # Smooth
    sSSEF = filter(SSEF, rep(1/L, L), circular = TRUE)
    sSSER = filter(SSER, rep(1/L, L), circular = TRUE)
    eF[,cond] = (den.df/num.df)*(sSSER/sSSEF)  }
  plot(Fr[nFr], eF[nFr, 1], type="l", xlab="Frequency", ylab="F Statistic",
    ylim=c(0, 5))
  abline(h=qf(.999, num.df, den.df), lty=2)
  if(Loc==1) mtext("Brush", side=3, line=.3, cex=1)
  mtext(loc.name[Loc], side=2, line=3, cex=.9)
  plot(Fr[nFr], eF[nFr, 2], type="l", xlab="Frequency", ylab="F Statistic",
    ylim=c(0, 5))
  abline(h=qf(.999, num.df, den.df), lty=2)
  if(Loc==1) mtext("Heat", side=3, line=.3, cex=1)
  plot(Fr[nFr], eF[nFr, 3], type="l", xlab="Frequency", ylab="F Statistic",
    ylim=c(0, 5))
  abline(h = qf(.999, num.df, den.df) ,lty=2)
  if(Loc==1) mtext("Shock", side=3, line=.3, cex=1)  }

```

MULTIVARIATE TESTS

Although it is possible to develop multivariate regression along lines analogous to the usual real valued case, we will only look at tests involving equality of group means and spectral matrices, because these tests appear to be used most often in applications. For these results, consider the p -variate time series $y_{ijt} = (y_{ijt1}, \dots, y_{ijtp})'$ to have arisen from observations on $j = 1, \dots, N_i$ individuals in group i , all having mean μ_{it} and stationary autocovariance matrix $\Gamma_i(h)$. Denote the DFTs of the group mean vectors as $Y_i(\omega_k)$ and the $p \times p$ spectral matrices as $\hat{f}_i(\omega_k)$ for the $i = 1, 2, \dots, I$ groups. Assume the same general properties as for the vector series considered in Section 7.3.

In the multivariate case, we obtain the analogous versions of (7.81) and (7.82) as the *between cross-power* and *within cross-power* matrices

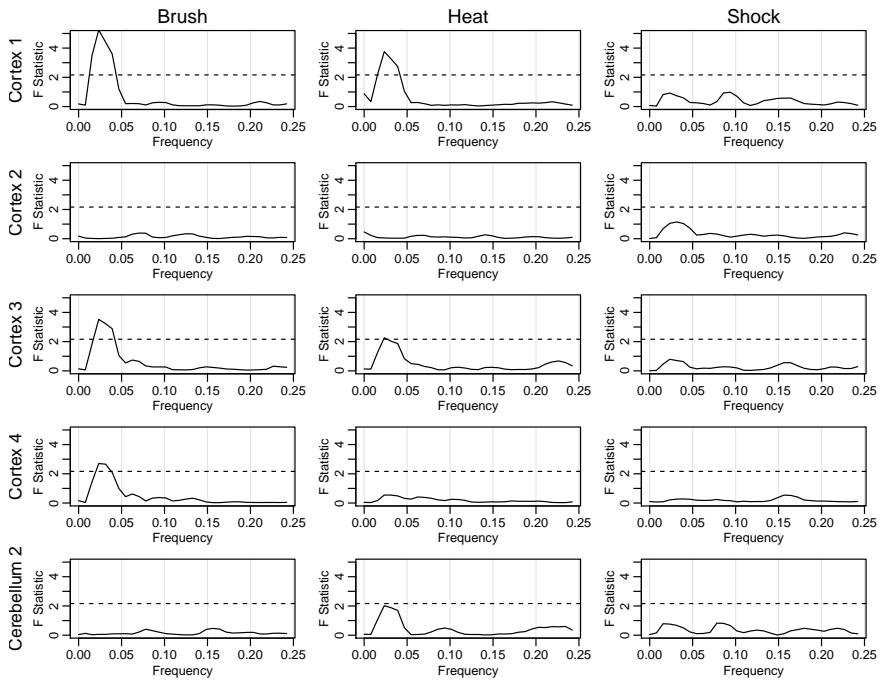


Fig. 7.10. Power in simultaneous linear compounds at five locations, enhancing brush, heat, and shock effects, $L = 3$, $F_{.001}(36, 120) = 2.16$.

$$\text{SPR}(\omega_k) = \sum_{i=1}^I \sum_{j=1}^{N_i} (Y_{i\cdot}(\omega_k) - Y_{..}(\omega_k))(Y_{i\cdot}(\omega_k) - Y_{..}(\omega_k))^* \quad (7.90)$$

and

$$\text{SPE}(\omega_k) = \sum_{i=1}^I \sum_{j=1}^{N_i} (Y_{ij}(\omega_k) - Y_{i\cdot}(\omega_k))(Y_{ij}(\omega_k) - Y_{i\cdot}(\omega_k))^*. \quad (7.91)$$

The equality of means test is rejected using the fact that the likelihood ratio test yields a monotone function of

$$\Lambda(\omega_k) = \frac{|\text{SPE}(\omega_k)|}{|\text{SPE}(\omega_k) + \text{SPR}(\omega_k)|}. \quad (7.92)$$

Khatri (1965) and Hannan (1970) give the approximate distribution of the statistic

$$\chi^2_{2(I-1)p} = -2 \left(\sum N_i - I - p - 1 \right) \log \Lambda(\omega_k) \quad (7.93)$$

as chi-squared with $2(I-1)p$ degrees of freedom when the group means are equal.

The case of $I = 2$ groups reduces to Hotelling's T^2 , as has been shown by Giri (1965), where

$$T^2 = \frac{N_1 N_2}{(N_1 + N_2)} [Y_1(\omega_k) - Y_2(\omega_k)]^* \hat{f}_v^{-1}(\omega_k) [Y_1(\omega_k) - Y_2(\omega_k)], \quad (7.94)$$

where

$$\hat{f}_v(\omega_k) = \frac{\text{SPE}(\omega_k)}{\sum_i N_i - I} \quad (7.95)$$

is the pooled error spectrum given in (7.91), with $I = 2$. The test statistic, in this case, is

$$F_{2p, 2(N_1 + N_2 - p - 1)} = \frac{(N_1 + N_2 - 2)p}{(N_1 + N_2 - p - 1)} T^2, \quad (7.96)$$

which was shown by Giri (1965) to have the indicated limiting F -distribution with $2p$ and $2(N_1 + N_2 - p - 1)$ degrees of freedom when the means are the same. The classical t -test for inequality of two univariate means will be just (7.95) and (7.96) with $p = 1$.

Testing equality of the spectral matrices is also of interest, not only for discrimination and pattern recognition, as considered in the next section, but also as a test indicating whether the equality of means test, which assumes equal spectral matrices, is valid. The test evolves from the likelihood ratio criterion, which compares the single group spectral matrices

$$\hat{f}_i(\omega_k) = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (Y_{ij}(\omega_k) - Y_{i\cdot}(\omega_k)) (Y_{ij}(\omega_k) - Y_{i\cdot}(\omega_k))^* \quad (7.97)$$

with the pooled spectral matrix (7.95). A modification of the likelihood ratio test, which incorporates the degrees of freedom $M_i = N_i - 1$ and $M = \sum M_i$ rather than the sample sizes into the likelihood ratio statistic, uses

$$L'(\omega_k) = \frac{M^{Mp}}{\prod_{i=1}^I M_i^{M_i p}} \frac{\prod |M_i \hat{f}_i(\omega_k)|^{M_i}}{|M \hat{f}_v(\omega_k)|^M}. \quad (7.98)$$

Krishnaiah et al. (1976) have given the moments of $L'(\omega_k)$ and calculated 95% critical points for $p = 3, 4$ using a Pearson Type I approximation. For reasonably large samples involving smoothed spectral estimators, the approximation involving the first term of the usual chi-squared series will suffice and Shumway (1982) has given

$$\chi_{(I-1)p^2}^2 = -2r \log L'(\omega_k), \quad (7.99)$$

where

$$1 - r = \frac{(p+1)(p-1)}{6p(I-1)} \left(\sum_i M_i^{-1} - M^{-1} \right), \quad (7.100)$$

with an approximate chi-squared distribution with $(I-1)p^2$ degrees of freedom when the spectral matrices are equal. Introduction of smoothing over L frequencies leads to replacing M_j and M by LM_j and LM in the equations above.

Of course, it is often of great interest to use the above result for testing equality of two univariate spectra, and it is obvious from the material in Chapter 4,

$$F_{2LM_1,2LM_2} = \frac{\hat{f}_1(\omega)}{\hat{f}_2(\omega)} \quad (7.101)$$

will have the requisite F -distribution with $2LM_1$ and $2LM_2$ degrees of freedom when spectra are smoothed over L frequencies.

Example 7.9 Equality of Means and Spectral Matrices

An interesting problem arises when attempting to develop a methodology for discriminating between waveforms originating from explosions and those that came from the more commonly occurring earthquakes. Figure 7.2 shows a small subset of a larger population of bivariate series consisting of two phases from each of eight earthquakes and eight explosions. If the large-sample approximations to normality hold for the DFTs of these series, it is of interest to know whether the differences between the two classes are better represented by the mean functions or by the spectral matrices. The tests described above can be applied to look at these two questions. The upper left panel of Figure 7.11 shows the test statistic (7.96) with the straight line denoting the critical level for $\alpha = .001$, i.e., $F_{.001}(4, 26) = 7.36$, for equal means using $L = 1$, and the test statistic remains well below its critical value at all frequencies, implying that the means of the two classes of series are not significantly different. Checking Figure 7.2 shows little reason exists to suspect that either the earthquakes or explosions have a nonzero mean signal. Checking the equality of the spectra and the spectral matrices, however, leads to a different conclusion. Some smoothing ($L = 21$) is useful here, and univariate tests on both the P and S components using (7.101) and $N_1 = N_2 = 8$ lead to strong rejections of the equal spectra hypotheses. The rejection seems stronger for the S component and we might tentatively identify that component as being dominant. Testing equality of the spectral matrices using (7.99) and $\chi^2_{.001}(4) = 18.47$ shows a similar strong rejection of the equality of spectral matrices. We use these results to suggest optimal discriminant functions based on spectral differences in the next section.

The R code for this example is as follows. We make use of the recycling feature of R and the fact that the data are bivariate to produce simple code specific to this problem in order to avoid having to use multiple arrays.

```
P = 1:1024; S = P+1024; N = 8; n = 1024; p.dim = 2; m = 10; L = 2*m+1
eq.P = as.ts(eqexp[P,1:8]); eq.S = as.ts(eqexp[S,1:8])
eq.m = cbind(rowMeans(eq.P), rowMeans(eq.S))
ex.P = as.ts(eqexp[P,9:16]); ex.S = as.ts(eqexp[S,9:16])
ex.m = cbind(rowMeans(ex.P), rowMeans(ex.S))
m.diff = mvfft(eq.m - ex.m)/sqrt(n)
eq.Pf = mvfft(eq.P-eq.m[,1])/sqrt(n)
eq.Sf = mvfft(eq.S-eq.m[,2])/sqrt(n)
ex.Pf = mvfft(ex.P-ex.m[,1])/sqrt(n)
ex.Sf = mvfft(ex.S-ex.m[,2])/sqrt(n)
fv11 = rowSums(eq.Pf*Conj(eq.Pf))+rowSums(ex.Pf*Conj(ex.Pf))/(2*(N-1))
fv12 = rowSums(eq.Pf*Conj(eq.Sf))+rowSums(ex.Pf*Conj(ex.Sf))/(2*(N-1))
fv22 = rowSums(eq.Sf*Conj(eq.Sf))+rowSums(ex.Sf*Conj(ex.Sf))/(2*(N-1))
fv21 = Conj(fv12)
# Equal Means
T2 = rep(NA, 512)
for (k in 1:512){
```

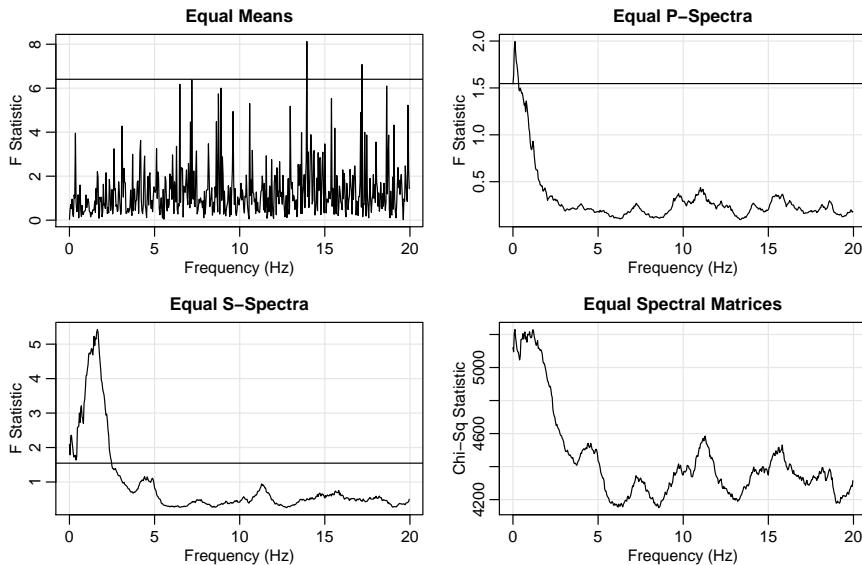


Fig. 7.11. Tests for equality of means, spectra, and spectral matrices for the earthquake and explosion data $p = 2, L = 21, n = 1024$ points at 40 points per second.

```

fvk   = matrix(c(fv11[k], fv21[k], fv12[k], fv22[k]), 2, 2)
dk   = as.matrix(m.diff[,])
T2[k] = Re((N/2)*Conj(t(dk))%*%solve(fvk,dk)) }
eF = T2*(2*p.dim*(N-1))/(2*N*p.dim-1)
par(mfrow=c(2,2), mar=c(3,3,2,1), mgp = c(1.6,.6,0), cex.main=1.1)
freq = 40*(0:511)/n # Hz
plot(freq, eF, type="l", xlab="Frequency (Hz)", ylab="F Statistic",
     main="Equal Means")
abline(h = qf(.999, 2*p.dim, 2*(2*N-p.dim-1)))
# Equal P
kd   = kernel("daniell",m);
u   = Re(rowSums(eq.Pf*Conj(eq.Pf))/(N-1))
feq.P = kernapply(u, kd, circular=TRUE)
u   = Re(rowSums(ex.Pf*Conj(ex.Pf))/(N-1))
fex.P = kernapply(u, kd, circular=TRUE)
plot(freq, feq.P[1:512]/fex.P[1:512], type="l", xlab="Frequency (Hz)",
     ylab="F Statistic", main="Equal P-Spectra")
abline(h=qf(.999, 2*L*(N-1), 2*L*(N-1)))
# Equal S
u   = Re(rowSums(eq.Sf*Conj(eq.Sf))/(N-1))
feq.S = kernapply(u, kd, circular=TRUE)
u   = Re(rowSums(ex.Sf*Conj(ex.Sf))/(N-1))
fex.S = kernapply(u, kd, circular=TRUE)
plot(freq, feq.S[1:512]/fex.S[1:512], type="l", xlab="Frequency (Hz)",
     ylab="F Statistic", main="Equal S-Spectra")
abline(h=qf(.999, 2*L*(N-1), 2*L*(N-1)))
# Equal Spectra
u   = rowSums(eq.Pf*Conj(eq.Sf))/(N-1)

```

```

freq.PS = kernapply(u, kd, circular=TRUE)
u = rowSums(ex.Pf*Conj(ex.Sf)/(N-1))
fex.PS = kernapply(u, kd, circular=TRUE)
fv11 = kernapply(fv11, kd, circular=TRUE)
fv22 = kernapply(fv22, kd, circular=TRUE)
fv12 = kernapply(fv12, kd, circular=TRUE)
Mi = L*(N-1); M = 2*Mi
TS = rep(NA, 512)
for (k in 1:512){
  det.freq.k = Re(freq.P[k]*freq.S[k] - freq.PS[k]*Conj(freq.PS[k]))
  det.fex.k = Re(fex.P[k]*fex.S[k] - fex.PS[k]*Conj(fex.PS[k]))
  det.fv.k = Re(fv11[k]*fv22[k] - fv12[k]*Conj(fv12[k]))
  log.n1 = log(M)*(M*p.dim); log.d1 = log(Mi)*(2*Mi*p.dim)
  log.n2 = log(Mi)^2 + log(det.freq.k)*Mi + log(det.fex.k)*Mi
  log.d2 = (log(M)+log(det.fv.k))*M
  r = 1 - ((p.dim-1)*(p.dim-1)/6*p.dim*(2-1))*(2/Mi - 1/M)
  TS[k] = -2*r*(log.n1+log.n2-log.d1-log.d2) }
plot(freq, TS, type="l", xlab="Frequency (Hz)", ylab="Chi-Sq Statistic",
      main="Equal Spectral Matrices")
abline(h = qchisq(.9999, p.dim^2))

```

7.7 Discriminant and Cluster Analysis

The extension of classical pattern-recognition techniques to experimental time series is a problem of great practical interest. A series of observations indexed in time often produces a pattern that may form a basis for discriminating between different classes of events. As an example, consider Figure 7.2, which shows regional (100-2000 km) recordings of several typical Scandinavian earthquakes and mining explosions measured by stations in Scandinavia. A listing of the events is given in Kakizawa et al. (1998). The problem of discriminating between mining explosions and earthquakes is a reasonable proxy for the problem of discriminating between nuclear explosions and earthquakes. This latter problem is one of critical importance for monitoring a comprehensive test-ban treaty. Time series classification problems are not restricted to geophysical applications, but occur under many and varied circumstances in other fields. Traditionally, the detection of a signal embedded in a noise series has been analyzed in the engineering literature by statistical pattern recognition techniques (see Problem 7.10 and Problem 7.11).

The historical approaches to the problem of discriminating among different classes of time series can be divided into two distinct categories. The *optimality* approach, as found in the engineering and statistics literature, makes specific Gaussian assumptions about the probability density functions of the separate groups and then develops solutions that satisfy well-defined minimum error criteria. Typically, in the time series case, we might assume the difference between classes is expressed through differences in the theoretical mean and covariance functions and use likelihood methods to develop an optimal classification function. A second class of techniques, which might be described as a *feature extraction* approach, proceeds more heuristically by looking at quantities that tend to be good visual discriminators for well-separated populations and have some basis in physical theory or intuition. Less

attention is paid to finding functions that are approximations to some well-defined optimality criterion.

As in the case of regression, both time domain and frequency domain approaches to discrimination will exist. For relatively short univariate series, a time domain approach that follows conventional multivariate discriminant analysis as described in conventional multivariate texts, such as Anderson (1984) or Johnson and Wichern (1992) may be preferable. We might even characterize differences by the autocovariance functions generated by different ARMA or state-space models. For longer multivariate time series that can be regarded as stationary after the common mean has been subtracted, the frequency domain approach will be easier computationally because the np dimensional vector in the time domain, represented here as $x = (x'_1, x'_t, \dots, x'_n)'$, with $x_t = (x_{t1}, \dots, x_{tp})'$, will be reduced to separate computations made on the p -dimensional DFTs. This happens because of the approximate independence of the DFTs, $X(\omega_k), 0 \leq \omega_k \leq 1$, a property that we have often used in preceding chapters.

Finally, the grouping properties of measures like the discrimination information and likelihood-based statistics can be used to develop measures of *disparity* for clustering multivariate time series. In this section, we define a measure of disparity between two multivariate times series by the spectral matrices of the two processes and then apply hierarchical clustering and partitioning techniques to identify natural groupings within the bivariate earthquake and explosion populations.

THE GENERAL DISCRIMINATION PROBLEM

The general problem of classifying a vector time series x occurs in the following way. We observe a time series x known to belong to one of g populations, denoted by $\Pi_1, \Pi_2, \dots, \Pi_g$. The general problem is to assign or *classify* this observation into one of the g groups in some optimal fashion. An example might be the $g = 2$ populations of earthquakes and explosions shown in Figure 7.2. We would like to classify the unknown event, shown as NZ in the bottom two panels, as belonging to either the earthquake (Π_1) or explosion (Π_2) populations. To solve this problem, we need an optimality criterion that leads to a statistic $T(x)$ that can be used to assign the NZ event to either the earthquake or explosion populations. To measure the success of the classification, we need to evaluate errors that can be expected in the future relating to the number of earthquakes classified as explosions (false alarms) and the number of explosions classified as earthquakes (missed signals).

The problem can be formulated by assuming the observed series x has a probability density $p_i(x)$ when the observed series is from population Π_i for $i = 1, \dots, g$. Then, partition the space spanned by the np -dimensional process x into g mutually exclusive regions R_1, R_2, \dots, R_g such that, if x falls in R_i , we assign x to population Π_i . The *misclassification probability* is defined as the probability of classifying the observation into population Π_j when it belongs to Π_i , for $j \neq i$ and would be given by the expression

$$P(j | i) = \int_{R_j} p_i(x) dx. \quad (7.102)$$

The overall *total error probability* depends also on the *prior probabilities*, say, $\pi_1, \pi_2, \dots, \pi_g$, of belonging to one of the g groups. For example, the probability that an observation x originates from Π_i and is then classified into Π_j is obviously $\pi_i P(j | i)$, and the total error probability becomes

$$P_e = \sum_{i=1}^g \pi_i \sum_{j \neq i} P(j | i). \quad (7.103)$$

Although costs have not been incorporated into (7.103), it is easy to do so by multiplying $P(j | i)$ by $C(j | i)$, the cost of assigning a series from population Π_i to Π_j .

The overall error P_e is minimized by classifying x into Π_i if

$$\frac{p_i(x)}{p_j(x)} > \frac{\pi_j}{\pi_i} \quad (7.104)$$

for all $j \neq i$ (see, for example, Anderson, 1984). A quantity of interest, from the Bayesian perspective, is the *posterior probability* an observation belongs to population Π_i , conditional on observing x , say,

$$P(\Pi_i | x) = \frac{\pi_i p_i(x)}{\sum_j \pi_j(x) p_j(x)}. \quad (7.105)$$

The procedure that classifies x into the population Π_i for which the posterior probability is largest is equivalent to that implied by using the criterion (7.104). The posterior probabilities give an intuitive idea of the relative odds of belonging to each of the plausible populations.

Many situations occur, such as in the classification of earthquakes and explosions, in which there are only $g = 2$ populations of interest. For two populations, the *Neyman–Pearson lemma* implies, in the absence of prior probabilities, classifying an observation into Π_1 when

$$\frac{p_1(x)}{p_2(x)} > K \quad (7.106)$$

minimizes each of the error probabilities for a fixed value of the other. The rule is identical to the Bayes rule (7.104) when $K = \pi_2/\pi_1$.

The theory given above takes a simple form when the vector x has a p -variate normal distribution with mean vectors μ_j and covariance matrices Σ_j under Π_j for $j = 1, 2, \dots, g$. In this case, simply use

$$p_j(x) = (2\pi)^{-p/2} |\Sigma_j|^{-1/2} \exp\left\{-\frac{1}{2}(x - \mu_j)' \Sigma_j^{-1} (x - \mu_j)\right\}. \quad (7.107)$$

The classification functions are conveniently expressed by quantities that are proportional to the logarithms of the densities, say,

$$g_j(x) = -\frac{1}{2} \ln |\Sigma_j| - \frac{1}{2} x' \Sigma_j^{-1} x + \mu_j' \Sigma_j^{-1} x - \frac{1}{2} \mu_j' \Sigma_j^{-1} \mu_j + \ln \pi_j. \quad (7.108)$$

In expressions involving the log likelihood, we will generally ignore terms involving the constant $-\ln 2\pi$. For this case, we may assign an observation x to population Π_i whenever

$$g_i(x) > g_j(x) \quad (7.109)$$

for $j \neq i, j = 1, \dots, g$ and the posterior probability (7.105) has the form

$$P(\Pi_i|x) = \frac{\exp\{g_i(x)\}}{\sum_j \exp\{g_j(x)\}}.$$

A common situation occurring in applications involves classification for $g = 2$ groups under the assumption of multivariate normality and equal covariance matrices; i.e., $\Sigma_1 = \Sigma_2 = \Sigma$. Then, the criterion (7.109) can be expressed in terms of the *linear discriminant function*

$$\begin{aligned} d_l(x) &= g_1(x) - g_2(x) \\ &= (\mu_1 - \mu_2)' \Sigma^{-1} x - \frac{1}{2} (\mu_1 - \mu_2)' \Sigma^{-1} (\mu_1 + \mu_2) + \ln \frac{\pi_1}{\pi_2}, \end{aligned} \quad (7.110)$$

where we classify into Π_1 or Π_2 according to whether $d_l(x) \geq 0$ or $d_l(x) < 0$. The linear discriminant function is clearly a combination of normal variables and, for the case $\pi_1 = \pi_2 = .5$, will have mean $D^2/2$ under Π_1 and mean $-D^2/2$ under Π_2 , with variances given by D^2 under both hypotheses, where

$$D^2 = (\mu_1 - \mu_2)' \Sigma^{-1} (\mu_1 - \mu_2) \quad (7.111)$$

is the *Mahalanobis distance* between the mean vectors μ_1 and μ_2 . In this case, the two misclassification probabilities (7.1) are

$$P(1|2) = P(2|1) = \Phi\left(-\frac{D}{2}\right), \quad (7.112)$$

and the performance is directly related to the Mahalanobis distance (7.111).

For the case in which the covariance matrices cannot be assumed to be the same, the discriminant function takes a different form, with the difference $g_1(x) - g_2(x)$ taking the form

$$\begin{aligned} d_q(x) &= -\frac{1}{2} \ln \frac{|\Sigma_1|}{|\Sigma_2|} - \frac{1}{2} x' (\Sigma_1^{-1} - \Sigma_2^{-1}) x \\ &\quad + (\mu_1' \Sigma_1^{-1} - \mu_2' \Sigma_2^{-1}) x + \ln \frac{\pi_1}{\pi_2} \end{aligned} \quad (7.113)$$

for $g = 2$ groups. This discriminant function differs from the equal covariance case in the linear term and in a nonlinear quadratic term involving the differing covariance matrices. The distribution theory is not tractable for the quadratic case so no convenient expression like (7.112) is available for the error probabilities for the quadratic discriminant function.

A difficulty in applying the above theory to real data is that the group mean vectors μ_j and covariance matrices Σ_j are seldom known. Some engineering problems, such

as the detection of a signal in white noise, assume the means and covariance parameters are known exactly, and this can lead to an optimal solution (see Problems 7.14 and 7.15). In the classical multivariate situation, it is possible to collect a sample of N_i training vectors from group Π_i , say, x_{ij} , for $j = 1, \dots, N_i$, and use them to estimate the mean vectors and covariance matrices for each of the groups $i = 1, 2, \dots, g$; i.e., simply choose $x_{i\cdot}$ and

$$S_i = (N_i - 1)^{-1} \sum_{j=1}^{N_i} (x_{ij} - x_{i\cdot})(x_{ij} - x_{i\cdot})' \quad (7.114)$$

as the estimators for μ_i and Σ_i , respectively. In the case in which the covariance matrices are assumed to be equal, simply use the pooled estimator

$$S = \left(\sum_i N_i - g \right)^{-1} \sum_i (N_i - 1) S_i. \quad (7.115)$$

For the case of a linear discriminant function, we may use

$$\hat{g}_i(x) = x_{i\cdot}' S^{-1} x - \frac{1}{2} x_{i\cdot}' S^{-1} x_{i\cdot} + \log \pi_i \quad (7.116)$$

as a simple estimator for $g_i(x)$. For large samples, $x_{i\cdot}$ and S converge to μ_i and Σ in probability so $\hat{g}_i(x)$ converges in distribution to $g_i(x)$ in that case. The procedure works reasonably well for the case in which $N_i, i = 1, \dots, g$ are large, relative to the length of the series n , a case that is relatively rare in time series analysis. For this reason, we will resort to using spectral approximations for the case in which data are given as long time series.

The performance of sample discriminant functions can be evaluated in several different ways. If the population parameters are known, (7.111) and (7.112) can be evaluated directly. If the parameters are estimated, the estimated Mahalanobis distance \hat{D}^2 can be substituted for the theoretical value in very large samples. Another approach is to calculate the *apparent error rates* using the result of applying the classification procedure to the training samples. If n_{ij} denotes the number of observations from population Π_j classified into Π_i , the sample error rates can be estimated by the ratio

$$\hat{P}(i \mid j) = \frac{n_{ij}}{\sum_i n_{ij}} \quad (7.117)$$

for $i \neq j$. If the training samples are not large, this procedure may be biased and a resampling option like cross-validation or the bootstrap can be employed. A simple version of cross-validation is the jackknife procedure proposed by Lachenbruch and Mickey (1968), which holds out the observation to be classified, deriving the classification function from the remaining observations. Repeating this procedure for each of the members of the training sample and computing (7.117) for the *holdout* samples leads to better estimators of the error rates.

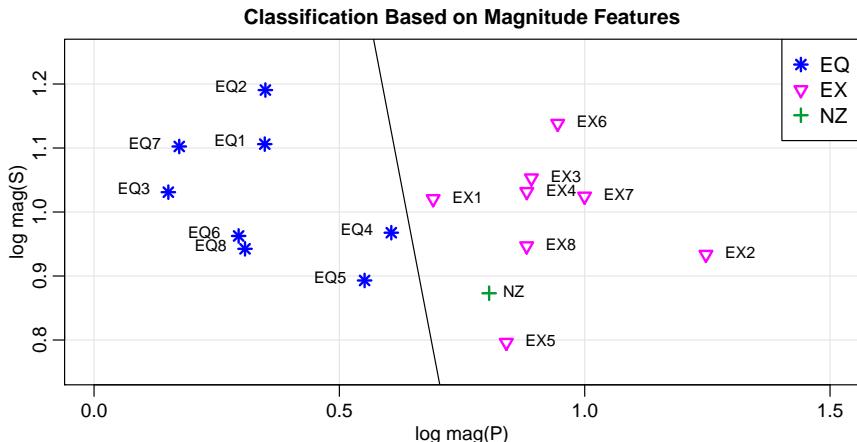


Fig. 7.12. Classification of earthquakes and explosions based on linear discriminant analysis using the magnitude features.

Example 7.10 Discriminant Analysis Using Amplitudes

We can give a simple example of applying the above procedures to the logarithms of the amplitudes of the separate P and S components of the original earthquake and explosion traces. The logarithms (base 10) of the maximum peak-to-peak amplitudes of the P and S components, denoted by $\log_{10} P$ and $\log_{10} S$, can be considered as two-dimensional feature vectors, say, $x = (x_1, x_2)' = (\log_{10} P, \log_{10} S)'$, from a bivariate normal population with differing means and covariances. The original data, from Kakizawa et al. (1998), are shown in Figure 7.12. The figure includes the Novaya Zemlya (NZ) event of unknown origin. The tendency of the earthquakes to have higher values for $\log_{10} S$, relative to $\log_{10} P$ has been noted by many and the use of the logarithm of the ratio, i.e., $\log_{10} P - \log_{10} S$ in some references (see Lay, 1997, pp. 40-41) is a tacit indicator that a linear function of the two parameters will be a useful discriminant.

The sample means $x_1. = (.346, 1.024)'$ and $x_2. = (.922, .993)'$, and covariance matrices

$$S_1 = \begin{pmatrix} .026 & -.007 \\ -.007 & .010 \end{pmatrix} \quad \text{and} \quad S_2 = \begin{pmatrix} .025 & -.001 \\ -.001 & .010 \end{pmatrix}$$

are immediate from (7.114), with the pooled covariance matrix given by

$$S = \begin{pmatrix} .026 & -.004 \\ -.004 & .010 \end{pmatrix}$$

from (7.115). Although the covariance matrices are not equal, we try the linear discriminant function anyway, which yields (with equal prior probabilities $\pi_1 = \pi_2 = .5$) the sample discriminant functions

$$\hat{g}_1(x) = 30.668x_1 + 111.411x_2 - 62.401$$

and

$$\hat{g}_2(x) = 54.048x_1 + 117.255x_2 - 83.142$$

from (7.116), with the estimated linear discriminant function (7.110) as

$$\hat{d}_l(x) = -23.380x_1 - 5.843x_2 + 20.740.$$

The jackknifed posterior probabilities of being an earthquake for the earthquake group ranged from .621 to 1.000, whereas the explosion probabilities for the explosion group ranged from .717 to 1.000. The unknown event, NZ, was classified as an explosion, with posterior probability .960.

The R code for this example is as follows.

```
P = 1:1024; S = P+1024
mag.P = log10(apply(eqexp[P,], 2, max) - apply(eqexp[P,], 2, min))
mag.S = log10(apply(eqexp[S,], 2, max) - apply(eqexp[S,], 2, min))
eq.P = mag.P[1:8]; eq.S = mag.S[1:8]
ex.P = mag.P[9:16]; ex.S = mag.S[9:16]
NZ.P = mag.P[17]; NZ.S = mag.S[17]
# Compute linear discriminant function
cov.eq = var(cbind(eq.P, eq.S))
cov.ex = var(cbind(ex.P, ex.S))
cov.pooled = (cov.ex + cov.eq)/2
means.eq = colMeans(cbind(eq.P, eq.S))
means.ex = colMeans(cbind(ex.P, ex.S))
slopes.eq = solve(cov.pooled, means.eq)
inter.eq = -sum(slopes.eq*means.eq)/2
slopes.ex = solve(cov.pooled, means.ex)
inter.ex = -sum(slopes.ex*means.ex)/2
d.slopes = slopes.eq - slopes.ex
d.inter = inter.eq - inter.ex
# Classify new observation
new.data = cbind(NZ.P, NZ.S)
d = sum(d.slopes*new.data) + d.inter
post.eq = exp(d)/(1+exp(d))
# Print (disc function, posteriors) and plot results
cat(d.slopes[1], "mag.P+", d.slopes[2], "mag.S+", d.inter, "\n")
cat("P(EQ|data) =", post.eq, " P(EX|data) =", 1-post.eq, "\n")
plot(eq.P, eq.S, xlim=c(0,1.5), ylim=c(.75,1.25), xlab="log mag(P)", ylab="log mag(S)", pch = 8, cex=1.1, lwd=2, main="Classification Based on Magnitude Features")
points(ex.P, ex.S, pch = 6, cex=1.1, lwd=2)
points(new.data, pch = 3, cex=1.1, lwd=2)
abline(a = -d.inter/d.slopes[2], b = -d.slopes[1]/d.slopes[2])
text(eq.P-.07,eq.S+.005, label=names(eqexp[1:8]), cex=.8)
text(ex.P+.07,ex.S+.003, label=names(eqexp[9:16]), cex=.8)
text(NZ.P+.05,NZ.S+.003, label=names(eqexp[17])), cex=.8)
legend("topright",c("EQ","EX","NZ"),pch=c(8,6,3),pt.lwd=2,cex=1.1)
# Cross-validation
all.data = rbind(cbind(eq.P, eq.S), cbind(ex.P, ex.S))
post.eq <- rep(NA, 8) -> post.ex
for(j in 1:16) {
  if (j <= 8){samp.eq = all.data[-c(j, 9:16),]
  samp.ex = all.data[9:16,]}
  if (j > 8){samp.eq = all.data[1:8,]
  samp.ex = all.data[-c(j, 1:8),]  }
}
```

```

df.eq      = nrow(samp.eq)-1; df.ex = nrow(samp.ex)-1
mean.eq    = colMeans(samp.eq); mean.ex = colMeans(samp.ex)
cov.eq = var(samp.eq); cov.ex = var(samp.ex)
cov.pooled = (df.eq*cov.eq + df.ex*cov.ex)/(df.eq + df.ex)
slopes.eq = solve(cov.pooled, mean.eq)
inter.eq   = -sum(slopes.eq*mean.eq)/2
slopes.ex  = solve(cov.pooled, mean.ex)
inter.ex   = -sum(slopes.ex*mean.ex)/2
d.slopes   = slopes.eq - slopes.ex
d.inter    = inter.eq - inter.ex
d          = sum(d.slopes*all.data[,]) + d.inter
if (j <= 8) post.eq[j] = exp(d)/(1+exp(d))
if (j > 8) post.ex[j-8] = 1/(1+exp(d)) }
Posterior = cbind(1:8, post.eq, 1:8, post.ex)
colnames(Posterior) = c("EQ","P(EQ|data)","EX","P(EX|data)")
round(Posterior,3) # Results from Cross-validation (not shown)

```

FREQUENCY DOMAIN DISCRIMINATION

The feature extraction approach often works well for discriminating between classes of univariate or multivariate series when there is a simple low-dimensional vector that seems to capture the essence of the differences between the classes. It still seems sensible, however, to develop optimal methods for classification that exploit the differences between the multivariate means and covariance matrices in the time series case. Such methods can be based on the Whittle approximation to the log likelihood given in Section 7.2. In this case, the vector DFTs, say, $X(\omega_k)$, are assumed to be approximately normal, with means $M_j(\omega_k)$ and spectral matrices $f_j(\omega_k)$ for population Π_j at frequencies $\omega_k = k/n$, for $k = 0, 1, \dots, [n/2]$, and are approximately uncorrelated at different frequencies, say, ω_k and ω_ℓ for $k \neq \ell$. Then, writing the complex normal densities as in Section 7.2 leads to a criterion similar to (7.108); namely,

$$g_j(X) = \ln \pi_j - \sum_{0<\omega_k<1/2} \left[\ln |f_j(\omega_k)| + X^*(\omega_k) f_j^{-1}(\omega_k) X(\omega_k) - 2M_j^*(\omega_k) f_j^{-1}(\omega_k) X(\omega_k) + M_j^*(k) f_j^{-1}(\omega_k) M_j(\omega_k) \right], \quad (7.118)$$

where the sum goes over frequencies for which $|f_j(\omega_k)| \neq 0$. The periodicity of the spectral density matrix and DFT allows adding over $0 < k < 1/2$. The classification rule is as in (7.109).

In the time series case, it is more likely the discriminant analysis involves assuming the covariance matrices are different and the means are equal. For example, the tests, shown in Figure 7.11, imply, for the earthquakes and explosions, the primary differences are in the bivariate spectral matrices and the means are essentially the same. For this case, it will be convenient to write the Whittle approximation to the log likelihood in the form

$$\ln p_j(X) = \sum_{0<\omega_k<1/2} \left[-\ln |f_j(\omega_k)| - X^*(\omega_k) f_j^{-1}(\omega_k) X(\omega_k) \right], \quad (7.119)$$

where we have omitted the prior probabilities from the equation. The quadratic detector in this case can be written in the form

$$\ln p_j(X) = \sum_{0 < \omega_k < 1/2} \left[-\ln |f_j(\omega_k)| - \text{tr} \{ I(\omega_k) f_j^{-1}(\omega_k) \} \right], \quad (7.120)$$

where

$$I(\omega_k) = X(\omega_k)X^*(\omega_k) \quad (7.121)$$

denotes the *periodogram matrix*. For equal prior probabilities, we may assign an observation x into population Π_i whenever

$$\ln p_i(X) > \ln p_j(X) \quad (7.122)$$

for $j \neq i, j = 1, 2, \dots, g$.

Numerous authors have considered various versions of discriminant analysis in the frequency domain. Shumway and Unger (1974) considered (7.118) for $p = 1$ and equal covariance matrices, so the criterion reduces to a simple linear one. They apply the criterion to discriminating between earthquakes and explosions using teleseismic P wave data in which the means over the two groups might be considered as fixed. Alagón (1989) and Dargahi-Noubary and Laycock (1981) considered discriminant functions of the form (7.118) in the univariate case when the means are zero and the spectra for the two groups are different. Taniguchi et al. (1994) adopted (7.119) as a criterion and discussed its *non-Gaussian robustness*. Shumway (1982) reviews general discriminant functions in both the univariate and multivariate time series cases.

MEASURES OF DISPARITY

Before proceeding to examples of discriminant and cluster analysis, it is useful to consider the relation to the Kullback–Leibler (K-L) *discrimination information*, as defined in [Problem 2.4](#). Using the spectral approximation and noting the periodogram matrix has the approximate expectation

$$E_j I(\omega_k) = f_j(\omega_k)$$

under the assumption that the data come from population Π_j , and approximating the ratio of the densities by

$$\ln \frac{p_1(X)}{p_2(X)} = \sum_{0 < \omega_k < 1/2} \left[-\ln \frac{|f_1(\omega_k)|}{|f_2(\omega_k)|} - \text{tr} \{ (f_2^{-1}(\omega_k) - f_1^{-1}(\omega_k)) I(\omega_k) \} \right],$$

we may write the approximate discrimination information as

$$\begin{aligned} I(f_1; f_2) &= \frac{1}{n} E_1 \ln \frac{p_1(X)}{p_2(X)} \\ &= \frac{1}{n} \sum_{0 < \omega_k < 1/2} \left[\text{tr} \{ f_1(\omega_k) f_2^{-1}(\omega_k) \} - \ln \frac{|f_1(\omega_k)|}{|f_2(\omega_k)|} - p \right]. \end{aligned} \quad (7.123)$$

The approximation may be carefully justified by noting the multivariate normal time series $x = (x'_1, x'_2, \dots, x'_n)$ with zero means and $np \times np$ stationary covariance matrices Γ_1 and Γ_2 will have $p, n \times n$ blocks, with elements of the form $\gamma_{ij}^{(l)}(s-t)$, $s, t = 1, \dots, n$, $i, j = 1, \dots, p$ for population Π_ℓ , $\ell = 1, 2$. The discrimination information, under these conditions, becomes

$$I(1; 2 : x) = \frac{1}{n} E_1 \ln \frac{p_1(x)}{p_2(x)} = \frac{1}{n} \left[\text{tr} \{ \Gamma_1 \Gamma_2^{-1} \} - \ln \frac{|\Gamma_1|}{|\Gamma_2|} - np \right]. \quad (7.124)$$

The limiting result

$$\lim_{n \rightarrow \infty} I(1; 2 : x) = \frac{1}{2} \int_{-1/2}^{1/2} \left[\text{tr} \{ f_1(\omega) f_2^{-1}(\omega) \} - \ln \frac{|f_1(\omega)|}{|f_2(\omega)|} - p \right] d\omega$$

has been shown, in various forms, by Pinsker (1964), Hannan (1970), and Kazakos and Papantoni-Kazakos (1980). The discrete version of (7.123) is just the approximation to the integral of the limiting form. The K-L measure of disparity is not a true distance, but it can be shown that $I(1; 2) \geq 0$, with equality if and only if $f_1(\omega) = f_2(\omega)$ almost everywhere. This result makes it potentially suitable as a measure of disparity between the two densities.

A connection exists, of course, between the discrimination information number, which is just the expectation of the likelihood criterion and the likelihood itself. For example, we may measure the disparity between the sample and the process defined by the theoretical spectrum $f_j(\omega_k)$ corresponding to population Π_j in the sense of Kullback (1958), as $I(\hat{f}; f_j)$, where

$$\hat{f}(\omega_k) = \sum_{\ell=-m}^m h_\ell I(\omega_k + \ell/n) \quad (7.125)$$

denotes the smoothed spectral matrix with weights $\{h_\ell\}$. The likelihood ratio criterion can be thought of as measuring the disparity between the periodogram and the theoretical spectrum for each of the populations. To make the discrimination information finite, we replace the periodogram implied by the log likelihood by the sample spectrum. In this case, the classification procedure can be regarded as finding the population closest, in the sense of minimizing disparity between the sample and theoretical spectral matrices. The classification in this case proceeds by simply choosing the population Π_j that minimizes $I(\hat{f}; f_j)$, i.e., assigning x to population Π_i whenever

$$I(\hat{f}; f_i) < I(\hat{f}; f_j) \quad (7.126)$$

for $j \neq i$, $j = 1, 2, \dots, g$.

Kakizawa et al. (1998) proposed using the *Chernoff (CH) information measure* (Chernoff, 1952, Renyi, 1961), defined as

$$B_\alpha(1; 2) = -\ln E_2 \left\{ \left(\frac{p_2(x)}{p_1(x)} \right)^\alpha \right\}, \quad (7.127)$$

where the measure is indexed by a *regularizing parameter* α , for $0 < \alpha < 1$. When $\alpha = .5$, the Chernoff measure is the *symmetric divergence* proposed by Bhattacharya (1943). For the multivariate normal case,

$$B_\alpha(1; 2 : x) = \frac{1}{n} \left[\ln \frac{|\alpha \Gamma_1 + (1 - \alpha) \Gamma_2|}{|\Gamma_2|} - \alpha \ln \frac{|\Gamma_1|}{|\Gamma_2|} \right]. \quad (7.128)$$

The large sample spectral approximation to the Chernoff information measure is analogous to that for the discrimination information, namely,

$$B_\alpha(f_1; f_2) = \frac{1}{2n} \sum_{0 < \omega_k < 1/2} \left[\ln \frac{|\alpha f_1(\omega_k) + (1 - \alpha) f_2(\omega_k)|}{|f_2(\omega_k)|} - \alpha \ln \frac{|f_1(\omega_k)|}{|f_2(\omega_k)|} \right]. \quad (7.129)$$

The Chernoff measure, when divided by $\alpha(1 - \alpha)$, behaves like the discrimination information in the limit in the sense that it converges to $I(1; 2 : x)$ for $\alpha \rightarrow 0$ and to $I(2; 1 : x)$ for $\alpha \rightarrow 1$. Hence, near the boundaries of the parameter α , it tends to behave like discrimination information and for other values represents a compromise between the two information measures. The classification rule for the Chernoff measure reduces to assigning x to population Π_i whenever

$$B_\alpha(\hat{f}; f_i) < B_\alpha(\hat{f}; f_j) \quad (7.130)$$

for $j \neq i, j = 1, 2, \dots, g$.

Although the classification rules above are well defined if the group spectral matrices are known, this will not be the case in general. If there are g training samples, $x_{ij}, j = 1, \dots, N_i, i = 1, \dots, g$, with N_i vector observations available in each group, the natural estimator for the spectral matrix of the group i is just the average spectral matrix (7.97), namely, with $\hat{f}_{ij}(\omega_k)$ denoting the estimated spectral matrix of series j from the i -th population,

$$\hat{f}_i(\omega_k) = \frac{1}{N_i} \sum_{j=1}^{N_i} \hat{f}_{ij}(\omega_k). \quad (7.131)$$

A second consideration is the choice of the regularization parameter α for the Chernoff criterion, (7.129). For the case of $g = 2$ groups, it should be chosen to maximize the disparity between the two group spectra, as defined in (7.129). Kakizawa et al. (1998) simply plot (7.129) as a function of α , using the estimated group spectra in (7.131), choosing the value that gives the maximum disparity between the two groups.

Example 7.11 Discriminant Analysis on Seismic Data

The simplest approaches to discriminating between the earthquake and explosion groups have been based on either the relative amplitudes of the P and S phases, as in Figure 7.5 or on relative power components in various frequency bands. Considerable effort has been expended on using various spectral ratios involving

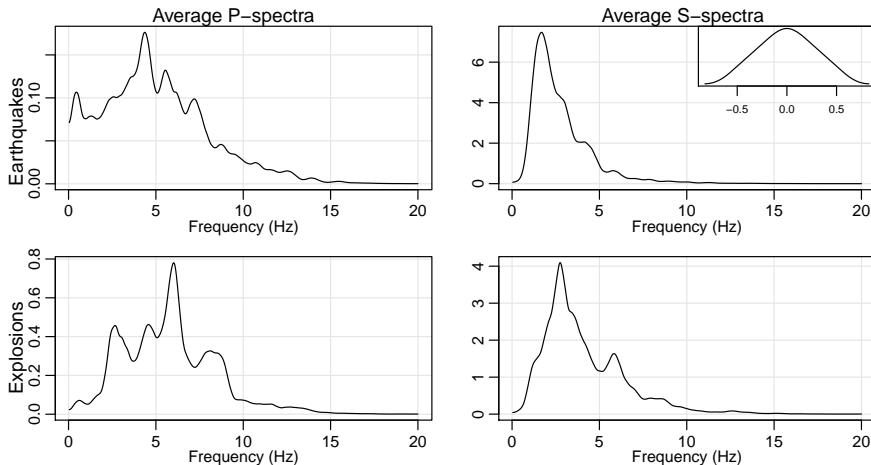


Fig. 7.13. Average P-spectra and S-spectra of the earthquake and explosion series. The insert on the upper right shows the smoothing kernel used; the resulting bandwidth is about .75 Hz.

the bivariate P and S phases as discrimination features. Kakizawa et al. (1998) mention a number of measures that have been used in the seismological literature as features. These features include ratios of power for the two phases and ratios of power components in high- and low-frequency bands. The use of such features of the spectrum suggests an optimal procedure based on discriminating between the spectral matrices of two stationary processes would be reasonable. The fact that the hypothesis that the spectral matrices were equal, tested in [Example 7.9](#), was also soundly rejected suggests the use of a discriminant function based on spectral differences. Recall the sampling rate is 40 points per second, leading to a folding frequency of 20 Hz.

[Figure 7.13](#) displays the diagonal elements of the average spectral matrices for each group. The maximum value of the estimated Chernoff disparity $B_\alpha(\hat{f}_1; \hat{f}_2)$ occurs for $\alpha = .4$, and we use that value in the discriminant criterion [\(7.129\)](#). [Figure 7.14](#) shows the results of using the Chernoff differences along with the Kullback-Leibler differences for classification. The differences are the measures for earthquakes minus explosions, so negative values of the differences indicate earthquake and positive values indicate explosion. Hence, points in the first quadrant of [Figure 7.14](#) are classified an explosion and points in the third quadrant are classified as earthquakes. We note that Explosion 6 is misclassified as an earthquake. Also, Earthquake 1, which falls in the fourth quadrant has an uncertain classification, the Chernoff distance classifies it as an earthquake, however, the Kullback-Leibler difference classifies it as an explosion.

The NZ event of unknown origin was also classified using these distance measures, and, as in [Example 7.10](#), it is classified as an explosion. The Russians have asserted no mine blasting or nuclear testing occurred in the area in question, so the event remains as somewhat of a mystery. The fact that it was relatively removed

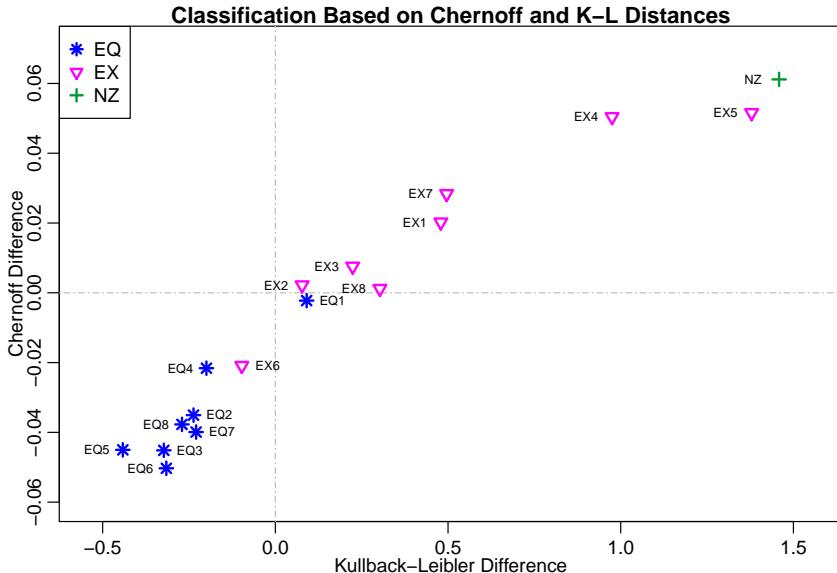


Fig. 7.14. Classification (by quadrant) of earthquakes and explosions using the Chernoff and Kullback-Leibler differences.

geographically from the test set may also have introduced some uncertainties into the procedure. The R code for this example is as follows.

```
P = 1:1024; S = P+1024; p.dim = 2; n =1024
eq = as.ts(eqexp[, 1:8])
ex = as.ts(eqexp[, 9:16])
nz = as.ts(eqexp[, 17])
f.eq <- array(dim=c(8, 2, 2, 512)) -> f.ex
f.NZ = array(dim=c(2, 2, 512))
# below calculates determinant for 2x2 Hermitian matrix
det.c <- function(mat){return(Re(mat[1,1]*mat[2,2]-mat[1,2]*mat[2,1]))}
L = c(15,13,5)      # for smoothing
for (i in 1:8){      # compute spectral matrices
  f.eq[i,,,] = mvspec(cbind(eq[P,i], eq[S,i]), spans=L, taper=.5)$ffx
  f.ex[i,,,] = mvspec(cbind(ex[P,i], ex[S,i]), spans=L, taper=.5)$ffx
  u = mvspec(cbind(nz[P], nz[S]), spans=L, taper=.5)
  f.NZ = u$ffx
bndwidth = u$bandwidth*sqrt(12)*40  # about .75 Hz
fhat.eq = apply(f.eq, 2:4, mean)    # average spectra
fhat.ex = apply(f.ex, 2:4, mean)
# plot the average spectra
par(mfrow=c(2,2), mar=c(3,3,2,1), mgp = c(1.6,.6,0))
Fr = 40*(1:512)/n
plot(Fr,Re(fhat.eq[1,1,]),type="l",xlab="Frequency (Hz)",ylab="")
plot(Fr,Re(fhat.eq[2,2,]),type="l",xlab="Frequency (Hz)",ylab="")
plot(Fr,Re(fhat.ex[1,1,]),type="l",xlab="Frequency (Hz)",ylab="")
plot(Fr,Re(fhat.ex[2,2,]),type="l",xlab="Frequency (Hz)",ylab="")
mtext("Average P-spectra", side=3, line=-1.5, adj=.2, outer=TRUE)
mtext("Earthquakes", side=2, line=-1, adj=.8, outer=TRUE)
```

```

mtext("Average S-spectra", side=3, line=-1.5, adj=.82, outer=TRUE)
mtext("Explosions", side=2, line=-1, adj=.2, outer=TRUE)
par(fig = c(.75, 1, .75, 1), new = TRUE)
ker = kernel("modified.daniell", L)$coef; ker = c(rev(ker),ker[-1])
plot((-33:33)/40, ker, type="l", ylab="", xlab="", cex.axis=.7,
      yaxp=c(0,.04,2))
# Choose alpha
Balpha = rep(0,19)
for (i in 1:19){ alf=i/20
for (k in 1:256) {
  Balpha[i] = Balpha[i] + Re(log(det.c(alf*fhat.ex[,,k] +
    (1-alf)*fhat.eq[,,k])/det.c(fhat.eq[,,k])) -
    alf*log(det.c(fhat.ex[,,k])/det.c(fhat.eq[,,k])))} }
alf = which.max(Balpha)/20 # alpha = .4
# Calculate Information Criteria
rep(0,17) -> KLDiff -> BDiff -> KLeq -> KLex -> Beq -> Bex
for (i in 1:17){
  if (i <= 8) f0 = f.eq[i,,,]
  if (i > 8 & i <= 16) f0 = f.ex[i-8,,,]
  if (i == 17) f0 = f.NZ
  for (k in 1:256) { # only use freqs out to .25
    tr = Re(sum(diag(solve(fhat.eq[,,k],f0[,,k]))))
    KLeq[i] = KLeq[i] + tr + log(det.c(fhat.eq[,,k])) - log(det.c(f0[,,k]))
    Beq[i] = Beq[i] +
      Re(log(det.c(alf*f0[,,k]+(1-alf)*fhat.eq[,,k])/det.c(fhat.eq[,,k])) -
        alf*log(det.c(f0[,,k])/det.c(fhat.eq[,,k])))
    tr = Re(sum(diag(solve(fhat.ex[,,k],f0[,,k]))))
    KLex[i] = KLex[i] + tr + log(det.c(fhat.ex[,,k])) - log(det.c(f0[,,k]))
    Bex[i] = Bex[i] +
      Re(log(det.c(alf*f0[,,k]+(1-alf)*fhat.ex[,,k])/det.c(fhat.ex[,,k])) -
        alf*log(det.c(f0[,,k])/det.c(fhat.ex[,,k])))}
  KLDiff[i] = (KLeq[i] - KLex[i])/n
  BDiff[i] = (Beq[i] - Bex[i])/(2*n) }
x.b = max(KLDiff)+.1; x.a = min(KLDiff)-.1
y.b = max(BDiff)+.01; y.a = min(BDiff)-.01
dev.new()
plot(KLDiff[9:16], BDiff[9:16], type="p", xlim=c(x.a,x.b), ylim=c(y.a,y.b),
      cex=1.1,lwd=2, xlab="Kullback-Leibler Difference",ylab="Chernoff
Difference", main="Classification Based on Chernoff and K-L
Distances", pch=6)
points(KLDiff[1:8], BDiff[1:8], pch=8, cex=1.1, lwd=2)
points(KLDiff[17], BDiff[17], pch=3, cex=1.1, lwd=2)
legend("topleft", legend=c("EQ", "EX", "NZ"), pch=c(8,6,3), pt.lwd=2)
abline(h=0, v=0, lty=2, col="gray")
text(KLDiff[-c(1,2,3,7,14)]-.075, BDiff[-c(1,2,3,7,14)],
      label=names(eqexp[-c(1,2,3,7,14)]), cex=.7)
text(KLDiff[c(1,2,3,7,14)]+.075, BDiff[c(1,2,3,7,14)],
      label=names(eqexp[c(1,2,3,7,14)]), cex=.7)

```

CLUSTER ANALYSIS

For the purpose of clustering, it may be more useful to consider a *symmetric disparity measures* and we introduce the *J-Divergence* measure

$$J(f_1; f_2) = I(f_1; f_2) + I(f_2; f_1) \quad (7.132)$$

and the symmetric Chernoff number

$$JB_\alpha(f_1; f_2) = B_\alpha(f_1; f_2) + B_\alpha(f_2; f_1) \quad (7.133)$$

for that purpose. In this case, we define the disparity between the sample spectral matrix of a single vector, x , and the population Π_j as

$$J(\hat{f}; f_j) = I(\hat{f}; f_j) + I(f_j; \hat{f}) \quad (7.134)$$

and

$$JB_\alpha(\hat{f}; f_j) = B_\alpha(\hat{f}; f_j) + B_\alpha(f_j; \hat{f}), \quad (7.135)$$

respectively and use these as quasi-distances between the vector and population Π_j .

The measures of disparity can be used to cluster multivariate time series. The symmetric measures of disparity, as defined above ensure that the disparity between f_i and f_j is the same as the disparity between f_j and f_i . Hence, we will consider the symmetric forms (7.134) and (7.135) as quasi-distances for the purpose of defining a distance matrix for input into one of the standard clustering procedures (see Johnson and Wichern, 1992). In general, we may consider either *hierarchical* or *partitioned* clustering methods using the quasi-distance matrix as an input.

For purposes of illustration, we may use the symmetric divergence (7.134), which implies the quasi-distance between sample series with estimated spectral matrices \hat{f}_i and \hat{f}_j would be (7.134); i.e.,

$$J(\hat{f}_i; \hat{f}_j) = \frac{1}{n} \sum_{0 < \omega_k < 1/2} \left[\text{tr} \{ \hat{f}_i(\omega_k) \hat{f}_j^{-1}(\omega_k) \} + \text{tr} \{ \hat{f}_j(\omega_k) \hat{f}_i^{-1}(\omega_k) \} - 2p \right], \quad (7.136)$$

for $i \neq j$. We can also use the comparable form for the Chernoff divergence, but we may not want to make an assumption for the regularization parameter α .

For hierarchical clustering, we begin by clustering the two members of the population that minimize the disparity measure (7.136). Then, these two items form a cluster, and we can compute distances between unclustered items as before. The distance between unclustered items and a current cluster is defined here as the average of the distances to elements in the cluster. Again, we combine objects that are closest together. We may also compute the distance between the unclustered items and clustered items as the closest distance, rather than the average. Once a series is in a cluster, it stays there. At each stage, we have a fixed number of clusters, depending on the merging stage.

Alternatively, we may think of clustering as a partitioning of the sample into a prespecified number of groups. MacQueen (1967) has proposed this using *k-means clustering*, using the Mahalanobis distance between an observation and the group mean vectors. At each stage, a reassignment of an observation into its closest affinity group is possible. To see how this procedure applies in the current context, consider a preliminary partition into a fixed number of groups and define the disparity between the spectral matrix of the observation, say, \hat{f} , and the average spectral matrix of the group, say, \hat{f}_i , as $J(\hat{f}; \hat{f}_i)$, where the group spectral matrix can be estimated by (7.131). At any pass, a single series is reassigned to the group for which its disparity is minimized. The reassignment procedure is repeated until all observations stay in

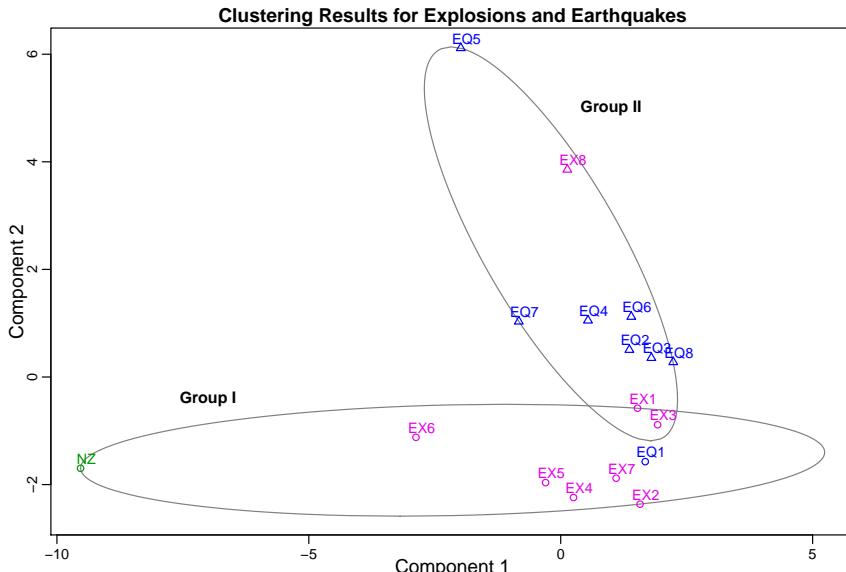


Fig. 7.15. Clustering results for the earthquake and explosion series based on symmetric divergence using a robust version of k-means clustering with two groups. Circles indicate Group I classification, triangles indicate Group II classification.

their current groups. Of course, the number of groups must be specified for each repetition of the partitioning algorithm and a starting partition must be chosen. This assignment can either be random or chosen from a preliminary hierarchical clustering, as described above.

Example 7.12 Cluster Analysis for Earthquakes and Explosions

It is instructive to try a clustering procedure on the population of known earthquakes and explosions. Figure 7.15 shows the results of applying the Partitioning Around Medoids (PAM) clustering algorithm, which is essentially a robustification of the k-means procedure (see Kaufman & Rousseeuw, 1990, Ch. 2), under the assumption that two groups are appropriate. The two-group partition tends to produce a final partition that agrees closely with the known configuration with earthquake 1 (EQ1) and explosion 8 (EX8) being misclassified; as in previous examples, the NZ event is classified as an explosion.

The R code for this example uses the `cluster` package and our `mvspec` script for estimating spectral matrices.

```
library(cluster)
P = 1:1024; S = P+1024; p.dim = 2; n = 1024
eq = as.ts(eqexp[, 1:8])
ex = as.ts(eqexp[, 9:16])
nz = as.ts(eqexp[, 17])
f = array(dim=c(17, 2, 2, 512))
L = c(15, 15)      # for smoothing
for (i in 1:8){    # compute spectral matrices
```

```

f[i,,,] = mvspec(cbind(eq[P,i], eq[S,i]), spans=L, taper=.5)$fxx
f[i+8,,,] = mvspec(cbind(ex[P,i], ex[S,i]), spans=L, taper=.5)$fxx }
f[17,,,] = mvspec(cbind(nz[P], nz[S]), spans=L, taper=.5)$fxx
JD = matrix(0, 17, 17)
# Calculate Symmetric Information Criteria
for (i in 1:16){
  for (j in (i+1):17){
    for (k in 1:256) { # only use freqs out to .25
      tr1 = Re(sum(diag(solve(f[i,,,k], f[j,,,k]))))
      tr2 = Re(sum(diag(solve(f[j,,,k], f[i,,,k]))))
      JD[i,j] = JD[i,j] + (tr1 + tr2 - 2*p.dim)}}
  JD = (JD + t(JD))/n
colnames(JD) = c(colnames(eq), colnames(ex), "NZ")
rownames(JD) = colnames(JD)
cluster.2 = pam(JD, k = 2, diss = TRUE)
summary(cluster.2) # print results
par(mgp = c(1.6,.6,0), cex=3/4, cex.lab=4/3, cex.main=4/3)
clusplot(JD, cluster.2$cluster, col.clus=1, labels=3, lines=0, col.p=1,
         main="Clustering Results for Explosions and Earthquakes")
text(-7,-.5, "Group I", cex=1.1, font=2)
text(1, 5, "Group II", cex=1.1, font=2)

```

7.8 Principal Components and Factor Analysis

In this section, we introduce the related topics of spectral domain principal components analysis and factor analysis for time series. The topics of principal components and canonical analysis in the frequency domain are rigorously presented in Brillinger (1981, Chapters 9 and 10) and many of the details concerning these concepts can be found there.

The techniques presented here are related to each other in that they focus on extracting pertinent information from spectral matrices. This information is important because dealing directly with a high-dimensional spectral matrix $f(\omega)$ itself is somewhat cumbersome because it is a function into the set of complex, nonnegative-definite, Hermitian matrices. We can view these techniques as easily understood, parsimonious tools for exploring the behavior of vector-valued time series in the frequency domain with minimal loss of information. Because our focus is on spectral matrices, we assume for convenience that the time series of interest have zero means; the techniques are easily adjusted in the case of nonzero means.

In this and subsequent sections, it will be convenient to work occasionally with *complex-valued time series*. A $p \times 1$ complex-valued time series can be represented as $x_t = x_{1t} - ix_{2t}$, where x_{1t} is the real part and x_{2t} is the imaginary part of x_t . The process is said to be stationary if $E(x_t)$ and $E(x_{t+h}x_t^*)$ exist and are independent of time t . The $p \times p$ autocovariance function,

$$\Gamma_{xx}(h) = E(x_{t+h}x_t^*) - E(x_{t+h})E(x_t^*),$$

of x_t satisfies conditions similar to those of the real-valued case. Writing $\Gamma_{xx}(h) = \{\gamma_{ij}(h)\}$, for $i, j = 1, \dots, p$, we have (i) $\gamma_{ii}(0) \geq 0$ is real, (ii) $|\gamma_{ij}(h)|^2 \leq \gamma_{ii}(0)\gamma_{jj}(0)$

for all integers h , and (iii) $\Gamma_{xx}(h)$ is a non-negative definite function. The spectral theory of complex-valued vector time series is analogous to the real-valued case. For example, if $\sum_h \|\Gamma_{xx}(h)\| < \infty$, the spectral density matrix of the complex series x_t is given by

$$f_{xx}(\omega) = \sum_{h=-\infty}^{\infty} \Gamma_{xx}(h) \exp(-2\pi i h\omega).$$

PRINCIPAL COMPONENTS

Classical principal component analysis (PCA) is concerned with explaining the variance–covariance structure among p variables, $x = (x_1, \dots, x_p)'$, through a few linear combinations of the components of x . Suppose we wish to find a linear combination

$$y = c'x = c_1 x_1 + \dots + c_p x_p \quad (7.137)$$

of the components of x such that $\text{var}(y)$ is as large as possible. Because $\text{var}(y)$ can be increased by simply multiplying c by a constant, it is common to restrict c to be of unit length; that is, $c'c = 1$. Noting that $\text{var}(y) = c'\Sigma_{xx}c$, where Σ_{xx} is the $p \times p$ variance–covariance matrix of x , another way of stating the problem is to find c such that

$$\max_{c \neq 0} \frac{c'\Sigma_{xx}c}{c'c}. \quad (7.138)$$

Denote the *eigenvalue–eigenvector pairs* of Σ_{xx} by $\{(\lambda_1, e_1), \dots, (\lambda_p, e_p)\}$, where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$, and the eigenvectors are of unit length. The solution to (7.138) is to choose $c = e_1$, in which case the linear combination $y_1 = e_1'x$ has maximum variance, $\text{var}(y_1) = \lambda_1$. In other words,

$$\max_{c \neq 0} \frac{c'\Sigma_{xx}c}{c'c} = \frac{e_1'\Sigma_{xx}e_1}{e_1'e_1} = \lambda_1. \quad (7.139)$$

The linear combination, $y_1 = e_1'x$, is called the *first principal component*. Because the eigenvalues of Σ_{xx} are not necessarily unique, the first principal component is not necessarily unique.

The *second principal component* is defined to be the linear combination $y_2 = c'x$ that maximizes $\text{var}(y_2)$ subject to $c'c = 1$ and such that $\text{cov}(y_1, y_2) = 0$. The solution is to choose $c = e_2$, in which case, $\text{var}(y_2) = \lambda_2$. In general, the k -th principal component, for $k = 1, 2, \dots, p$, is the linear combination $y_k = c'x$ that maximizes $\text{var}(y_k)$ subject to $c'c = 1$ and such that $\text{cov}(y_k, y_j) = 0$, for $j = 1, 2, \dots, k - 1$. The solution is to choose $c = e_k$, in which case $\text{var}(y_k) = \lambda_k$.

One measure of the importance of a principal component is to assess the proportion of the total variance attributed to that principal component. The *total variance* of x is defined to be the sum of the variances of the individual components; that is, $\text{var}(x_1) + \dots + \text{var}(x_p) = \sigma_{11} + \dots + \sigma_{pp}$, where σ_{jj} is the j -th diagonal element of Σ_{xx} . This sum is also denoted as $\text{tr}(\Sigma_{xx})$, or the *trace* of Σ_{xx} . Because $\text{tr}(\Sigma_{xx}) = \lambda_1 + \dots + \lambda_p$, the *proportion of the total variance attributed to the k -th principal component* is given simply by $\text{var}(y_k) / \text{tr}(\Sigma_{xx}) = \lambda_k / \sum_{j=1}^p \lambda_j$.

Given a random sample x_1, \dots, x_n , the *sample principal components* are defined as above, but with Σ_{xx} replaced by the sample variance–covariance matrix, $S_{xx} = (n-1)^{-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})'$. Further details can be found in the introduction to classical principal component analysis in Johnson and Wichern (1992, Chapter 9).

For the case of time series, suppose we have a zero mean, $p \times 1$, stationary vector process x_t that has a $p \times p$ spectral density matrix given by $f_{xx}(\omega)$. Recall $f_{xx}(\omega)$ is a complex-valued, nonnegative-definite, Hermitian matrix. Using the analogy of classical principal components, and in particular (7.137) and (7.138), suppose, for a fixed value of ω , we want to find a complex-valued univariate process $y_t(\omega) = c(\omega)^* x_t$, where $c(\omega)$ is complex, such that the spectral density of $y_t(\omega)$ is maximized at frequency ω , and $c(\omega)$ is of unit length, $c(\omega)^* c(\omega) = 1$. Because, at frequency ω , the spectral density of $y_t(\omega)$ is $f_y(\omega) = c(\omega)^* f_{xx}(\omega) c(\omega)$, the problem can be restated as: Find complex vector $c(\omega)$ such that

$$\max_{c(\omega) \neq 0} \frac{c(\omega)^* f_{xx}(\omega) c(\omega)}{c(\omega)^* c(\omega)}. \quad (7.140)$$

Let $\{(\lambda_1(\omega), e_1(\omega)), \dots, (\lambda_p(\omega), e_p(\omega))\}$ denote the eigenvalue–eigenvector pairs of $f_{xx}(\omega)$, where $\lambda_1(\omega) \geq \lambda_2(\omega) \geq \dots \geq \lambda_p(\omega) \geq 0$, and the eigenvectors are of unit length. We note that the eigenvalues of a Hermitian matrix are real. The solution to (7.140) is to choose $c(\omega) = e_1(\omega)$; in which case the desired linear combination is $y_t(\omega) = e_1(\omega)^* x_t$. For this choice,

$$\max_{c(\omega) \neq 0} \frac{c(\omega)^* f_{xx}(\omega) c(\omega)}{c(\omega)^* c(\omega)} = \frac{e_1(\omega)^* f_x(\omega) e_1(\omega)}{e_1(\omega)^* e_1(\omega)} = \lambda_1(\omega). \quad (7.141)$$

This process may be repeated for any frequency ω , and the complex-valued process, $y_{t1}(\omega) = e_1(\omega)^* x_t$, is called the *first principal component at frequency ω* . The k -th principal component at frequency ω , for $k = 1, 2, \dots, p$, is the complex-valued time series $y_{tk}(\omega) = e_k(\omega)^* x_t$, in analogy to the classical case. In this case, the spectral density of $y_{tk}(\omega)$ at frequency ω is $f_{y_k}(\omega) = e_k(\omega)^* f_{xx}(\omega) e_k(\omega) = \lambda_k(\omega)$.

The previous development of spectral domain principal components is related to the *spectral envelope* methodology first discussed in Stoffer et al. (1993). We will present the spectral envelope in the next section, where we motivate the use of principal components as it is presented above. Another way to motivate the use of principal components in the frequency domain was given in Brillinger (1981, Chapter 9). Although this technique leads to the same analysis, the motivation may be more satisfactory to the reader at this point. In this case, we suppose we have a stationary, p -dimensional, vector-valued process x_t and we are only able to keep a univariate process y_t such that, when needed, we may reconstruct the vector-valued process, x_t , according to an optimality criterion.

Specifically, we suppose we want to approximate a mean-zero, stationary, vector-valued time series, x_t , with spectral matrix $f_{xx}(\omega)$, by a univariate process y_t defined by

$$y_t = \sum_{j=-\infty}^{\infty} c_{t-j}^* x_j, \quad (7.142)$$

where $\{c_j\}$ is a $p \times 1$ vector-valued filter, such that $\{c_j\}$ is absolutely summable; that is, $\sum_{j=-\infty}^{\infty} |c_j| < \infty$. The approximation is accomplished so the reconstruction of x_t from y_t , say,

$$\hat{x}_t = \sum_{j=-\infty}^{\infty} b_{t-j} y_j, \quad (7.143)$$

where $\{b_j\}$ is an absolutely summable $p \times 1$ filter, is such that the mean square approximation error

$$E\{(x_t - \hat{x}_t)^*(x_t - \hat{x}_t)\} \quad (7.144)$$

is minimized.

Let $b(\omega)$ and $c(\omega)$ be the transforms of $\{b_j\}$ and $\{c_j\}$, respectively. For example,

$$c(\omega) = \sum_{j=-\infty}^{\infty} c_j \exp(-2\pi i j \omega), \quad (7.145)$$

and, consequently,

$$c_j = \int_{-1/2}^{1/2} c(\omega) \exp(2\pi i j \omega) d\omega. \quad (7.146)$$

Brillinger (1981, Theorem 9.3.1) shows the solution to the problem is to choose $c(\omega)$ to satisfy (7.140) and to set $b(\omega) = \overline{c(\omega)}$. This is precisely the previous problem, with the solution given by (7.141). That is, we choose $c(\omega) = e_1(\omega)$ and $b(\omega) = \overline{e_1(\omega)}$; the filter values can be obtained via the inversion formula given by (7.146). Using these results, in view of (7.142), we may form the *first principal component series*, say y_{t1} .

This technique may be extended by requesting another series, say, y_{t2} , for approximating x_t with respect to minimum mean square error, but where the coherency between y_{t2} and y_{t1} is zero. In this case, we choose $c(\omega) = e_2(\omega)$. Continuing this way, we can obtain the first $q \leq p$ principal components series, say, $y_t = (y_{t1}, \dots, y_{tq})'$, having spectral density $f_q(\omega) = \text{diag}\{\lambda_1(\omega), \dots, \lambda_q(\omega)\}$. The series y_{tk} is the k -th principal component series.

As in the classical case, given observations, x_1, x_2, \dots, x_n , from the process x_t , we can form an estimate $\hat{f}_{xx}(\omega)$ of $f_{xx}(\omega)$ and define the *sample principal component series* by replacing $f_{xx}(\omega)$ with $\hat{f}_{xx}(\omega)$ in the previous discussion. Precise details pertaining to the asymptotic ($n \rightarrow \infty$) behavior of the principal component series and their spectra can be found in Brillinger (1981, Chapter 9). To give a basic idea of what we can expect, we focus on the first principal component series and on the spectral estimator obtained by smoothing the periodogram matrix, $I_n(\omega)$; that is

$$\hat{f}_{xx}(\omega_j) = \sum_{\ell=-m}^m h_\ell I_n(\omega_j + \ell/n), \quad (7.147)$$

where $L = 2m + 1$ is odd and the weights are chosen so $h_\ell = h_{-\ell}$ are positive and $\sum_\ell h_\ell = 1$. Under the conditions for which $\hat{f}_{xx}(\omega_j)$ is a well-behaved estimator of $f_{xx}(\omega_j)$, and for which the largest eigenvalue of $f_{xx}(\omega_j)$ is unique,

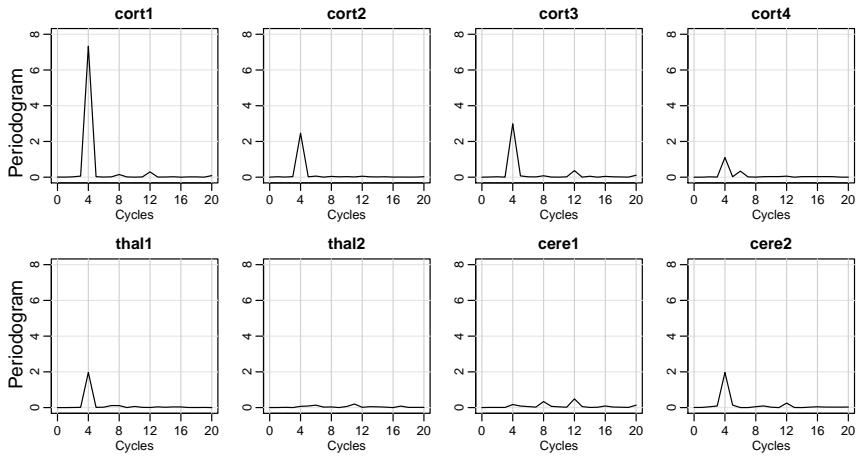


Fig. 7.16. The individual periodograms of x_{tk} , for $k = 1, \dots, 8$, in Example 7.13.

$$\left\{ \eta_n \frac{\hat{\lambda}_1(\omega_j) - \lambda_1(\omega_j)}{\lambda_1(\omega_j)}; \eta_n [\hat{e}_1(\omega_j) - e_1(\omega_j)] ; j = 1, \dots, J \right\} \quad (7.148)$$

converges ($n \rightarrow \infty$) jointly in distribution to independent, zero-mean normal distributions, the first of which is standard normal. In (7.148), $\eta_n^{-2} = \sum_{\ell=-m}^m h_\ell^2$, noting we must have $L \rightarrow \infty$ and $\eta_n \rightarrow \infty$, but $L/n \rightarrow 0$ as $n \rightarrow \infty$. The asymptotic variance–covariance matrix of $\hat{e}_1(\omega)$, say, $\Sigma_{e_1}(\omega)$, is given by

$$\Sigma_{e_1}(\omega) = \eta_n^{-2} \lambda_1(\omega) \sum_{\ell=2}^p \lambda_\ell(\omega) \{ \lambda_1(\omega) - \lambda_\ell(\omega) \}^{-2} e_\ell(\omega) e_\ell^*(\omega). \quad (7.149)$$

The distribution of $\hat{e}_1(\omega)$ depends on the other latent roots and vectors of $f_x(\omega)$. Writing $\hat{e}_1(\omega) = (\hat{e}_{11}(\omega), \hat{e}_{12}(\omega), \dots, \hat{e}_{1p}(\omega))'$, we may use this result to form confidence regions for the components of \hat{e}_1 by approximating the distribution of

$$\frac{2 |\hat{e}_{1,j}(\omega) - e_{1,j}(\omega)|^2}{s_j^2(\omega)}, \quad (7.150)$$

for $j = 1, \dots, p$, by a χ^2 distribution with two degrees of freedom. In (7.150), $s_j^2(\omega)$ is the j -th diagonal element of $\hat{\Sigma}_{e_1}(\omega)$, the estimate of $\Sigma_{e_1}(\omega)$. We can use (7.150) to check whether the value of zero is in the confidence region by comparing $2|\hat{e}_{1,j}(\omega)|^2/s_j^2(\omega)$ with $\chi_2^2(1 - \alpha)$, the $1 - \alpha$ upper tail cutoff of the χ_2^2 distribution.

Example 7.13 Principal Component Analysis of the fMRI Data

Recall Example 1.6 where the vector time series $x_t = (x_{t1}, \dots, x_{t8})'$, $t = 1, \dots, 128$, represents consecutive measures of average blood oxygenation level dependent (BOLD) signal intensity, which measures areas of activation in the brain. Recall

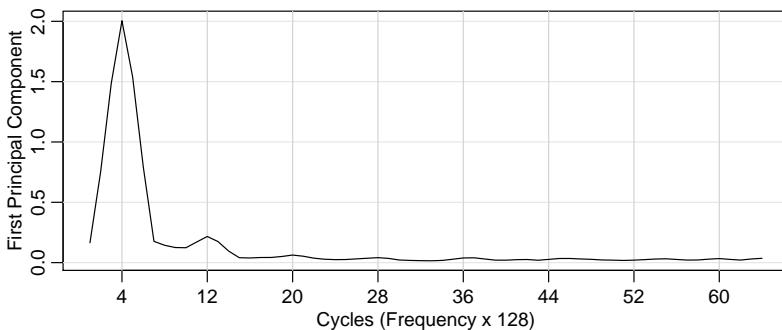


Fig. 7.17. The estimated spectral density, $\hat{\lambda}_1(j/128)$, of the first principal component series in Example 7.13.

subjects were given a non-painful brush on the hand and the stimulus was applied for 32 seconds and then stopped for 32 seconds; thus, the signal period is 64 seconds (the sampling rate was one observation every two seconds for 256 seconds). The series x_{tk} for $k = 1, 2, 3, 4$ represent locations in cortex, series x_{t5} and x_{t6} represent locations in the thalamus, and x_{t7} and x_{t8} represent locations in the cerebellum.

As is evident from Figure 1.6, different areas of the brain are responding differently, and a principal component analysis may help in indicating which locations are responding with the most spectral power, and which locations do not contribute to the spectral power at the stimulus signal period. In this analysis, we will focus primarily on the signal period of 64 seconds, which translates to four cycles in 256 seconds or $\omega = 4/128$ cycles per time point.

Figure 7.16 shows individual periodograms of the series x_{tk} for $k = 1, \dots, 8$. As was evident from Figure 1.6, a strong response to the brush stimulus occurred in areas of the cortex. To estimate the spectral density of x_t , we used (7.147) with $L = 5$ and $\{h_0 = 3/9, h_{\pm 1} = 2/9, h_{\pm 2} = 1/9\}$; this is a Daniell kernel with $m = 1$ passed twice. Calling the estimated spectrum $\hat{f}_{xx}(j/128)$, for $j = 0, 1, \dots, 64$, we can obtain the estimated spectrum of the first principal component series y_{t1} by calculating the largest eigenvalue, $\hat{\lambda}_1(j/128)$, of $\hat{f}_{xx}(j/128)$ for each $j = 0, 1, \dots, 64$. The result, $\hat{\lambda}_1(j/128)$, is shown in Figure 7.17. As expected, there is a large peak at the stimulus frequency $4/128$, wherein $\hat{\lambda}_1(4/128) = 2$. The total power at the stimulus frequency is $\text{tr}(\hat{f}_{xx}(4/128)) = 2.05$, so the proportion of the power at frequency $4/128$ attributed to the first principal component series is about $2/2.05$ or roughly 98%. Because the first principal component explains nearly all of the total power at the stimulus frequency, there is no need to explore the other principal component series at this frequency.

The estimated first principal component series at frequency $4/128$ is given by $\hat{y}_{t1}(4/128) = \hat{e}_1^*(4/128)x_t$, and the components of $\hat{e}_1(4/128)$ can give insight as to which locations of the brain are responding to the brush stimulus. Table 7.5 shows the magnitudes of $\hat{e}_1(4/128)$. In addition, an approximate 99% confidence interval was obtained for each component using (7.150). As expected, the analysis indicates

Table 7.5. Magnitudes of the PC Vector at the Stimulus Frequency

Location	1	2	3	4	5	6	7	8
$\left \hat{e}_1 \left(\frac{4}{128} \right) \right $.64	.36	.36	.22	.32	.05*	.13	.39

*Zero is in an approximate 99% confidence region for this component.

that location 6 is not contributing to the power at this frequency, but surprisingly, the analysis suggests location 5 (cerebellum 1) is responding to the stimulus.

The R code for this example is as follows.

```
n = 128; Per = abs(mvfft(fmri1[,-1]))^2/n
par(mfrow=c(2,4), mar=c(3,2,2,1), mgp = c(1.6,.6,0), oma=c(0,1,0,0))
for (i in 1:8){ plot(0:20, Per[1:21,i], type="l", ylim=c(0,8),
  main=colnames(fmri1)[i+1], xlab="Cycles", ylab="", xaxp=c(0,20,5))
mtext("Periodogram", side=2, line=-.3, outer=TRUE, adj=c(.2,.8))
dev.new()
fxx = mvspec(fmri1[,-1], kernel="daniell", c(1,1)), taper=.5, plot=FALSE)$fxx
l.val = rep(NA,64)
for (k in 1:64) {
u = eigen(fxx[,k], symmetric=TRUE, only.values = TRUE)
l.val[k] = u$values[1]} # largest e-value
plot(l.val, type="n", xaxt="n", xlab="Cycles (Frequency x 128)", ylab="First
  Principal Component")
axis(1, seq(4,60,by=8)); grid(lty=2, nx=NA, ny=NULL)
abline(v=seq(4,60,by=8), col='lightgray', lty=2); lines(l.val)
# At freq 4/128
u = eigen(fxx[,4], symmetric=TRUE)
lam=u$values; evec=u$vectors
lam[1]/sum(lam) # % of variance explained
sig.e1 = matrix(0,8,8)
for (l in 2:5){ # last 3 evs are 0
  sig.e1 = sig.e1 + lam[l]*evec[,l] %*% Conj(t(evec[,l]))/(lam[1]-lam[l])^2
  sig.e1 = Re(sig.e1)*lam[1]*sum(kernel("daniell", c(1,1))$coef^2)
p.val = round(pchisq(2*abs(evec[,1])^2/diag(sig.e1), 2, lower.tail=FALSE), 3)
cbind(colnames(fmri1)[-1], abs(evec[,1]), p.val) # table values
```

FACTOR ANALYSIS

Classical factor analysis is similar to classical principal component analysis. Suppose x is a mean-zero, $p \times 1$, random vector with variance–covariance matrix Σ_{xx} . The factor model proposes that x is dependent on a few unobserved common factors, z_1, \dots, z_q , plus error. In this model, one hopes that q will be much smaller than p . The *factor model* is given by

$$x = \mathcal{B}z + \epsilon, \quad (7.151)$$

where \mathcal{B} is a $p \times q$ matrix of *factor loadings*, $z = (z_1, \dots, z_q)'$ is a random $q \times 1$ vector of *factors* such that $E(z) = 0$ and $E(zz') = I_q$, the $q \times q$ identity matrix. The $p \times 1$ unobserved error vector ϵ is assumed to be independent of the factors, with zero mean and diagonal variance-covariance matrix $D = \text{diag}\{\delta_1^2, \dots, \delta_p^2\}$. Note, (7.151)

differs from the multivariate regression model in [Section 5.6](#) because the factors, z , are unobserved. Equivalently, the factor model, [\(7.151\)](#), can be written in terms of the covariance structure of x ,

$$\Sigma_{xx} = \mathcal{B}\mathcal{B}' + D; \quad (7.152)$$

i.e., the variance-covariance matrix of x is the sum of a symmetric, nonnegative-definite rank $q \leq p$ matrix and a nonnegative-definite diagonal matrix. If $q = p$, then Σ_{xx} can be reproduced exactly as $\mathcal{B}\mathcal{B}'$, using the fact that $\Sigma_{xx} = \lambda_1 e_1 e_1' + \dots + \lambda_p e_p e_p'$, where (λ_i, e_i) are the eigenvalue–eigenvector pairs of Σ_{xx} . As previously indicated, however, we hope q will be much smaller than p . Unfortunately, most covariance matrices cannot be factored as [\(7.152\)](#) when q is much smaller than p .

To motivate factor analysis, suppose the components of x can be grouped into meaningful groups. Within each group, the components are highly correlated, but the correlation between variables that are not in the same group is small. A group is supposedly formed by a single construct, represented as an unobservable factor, responsible for the high correlations within a group. For example, a person competing in a decathlon performs $p = 10$ athletic events, and we may represent the outcome of the decathlon as a 10×1 vector of scores. The events in a decathlon involve running, jumping, or throwing, and it is conceivable the 10×1 vector of scores might be able to be factored into $q = 4$ factors, (1) arm strength, (2) leg strength, (3) running speed, and (4) running endurance. The model [\(7.151\)](#) specifies that $\text{cov}(x, z) = \mathcal{B}$, or $\text{cov}(x_i, z_j) = b_{ij}$ where b_{ij} is the ij -th component of the *factor loading matrix* \mathcal{B} , for $i = 1, \dots, p$ and $j = 1, \dots, q$. Thus, the elements of \mathcal{B} are used to identify which hypothetical factors the components of x belong to, or load on.

At this point, some ambiguity is still associated with the factor model. Let Q be a $q \times q$ orthogonal matrix; that is $Q'Q = QQ' = I_q$. Let $\mathcal{B}_* = \mathcal{B}Q$ and $z_* = Q'z$ so [\(7.151\)](#) can be written as

$$x = \mathcal{B}z + \epsilon = \mathcal{B}QQ'z + \epsilon = \mathcal{B}_*z_* + \epsilon. \quad (7.153)$$

The model in terms of \mathcal{B}_* and z_* fulfills all of the factor model requirements, for example, $\text{cov}(z_*) = Q'\text{cov}(z)Q = QQ' = I_q$, so

$$\Sigma_{xx} = \mathcal{B}_*\text{cov}(z_*)\mathcal{B}_* + D = \mathcal{B}QQ'\mathcal{B}' + D = \mathcal{B}\mathcal{B}' + D. \quad (7.154)$$

Hence, on the basis of observations on x , we cannot distinguish between the loadings \mathcal{B} and the rotated loadings $\mathcal{B}_* = \mathcal{B}Q$. Typically, Q is chosen so the matrix \mathcal{B} is easy to interpret, and this is the basis of what is called *factor rotation*.

Given a sample x_1, \dots, x_n , a number of methods are used to estimate the parameters of the factor model, and we discuss two of them here. The first method is the *principal component method*. Let S_{xx} denote the sample variance–covariance matrix, and let $(\hat{\lambda}_i, \hat{e}_i)$ be the eigenvalue–eigenvector pairs of S_{xx} . The $p \times q$ matrix of estimated factor loadings is found by setting

$$\hat{\mathcal{B}} = \left[\hat{\lambda}_1^{1/2} \hat{e}_1 \mid \hat{\lambda}_2^{1/2} \hat{e}_2 \mid \dots \mid \hat{\lambda}_q^{1/2} \hat{e}_q \right]. \quad (7.155)$$

The argument here is that if q factors exist, then

$$S_{xx} \approx \hat{\lambda}_1 \hat{e}_1 \hat{e}'_1 + \cdots + \hat{\lambda}_q \hat{e}_q \hat{e}'_q = \hat{\mathcal{B}} \hat{\mathcal{B}}', \quad (7.156)$$

because the remaining eigenvalues, $\hat{\lambda}_{q+1}, \dots, \hat{\lambda}_p$, will be negligible. The estimated diagonal matrix of error variances is then obtained by setting $\hat{D} = \text{diag}\{\hat{\delta}_1^2, \dots, \hat{\delta}_p^2\}$, where $\hat{\delta}_j^2$ is the j -th diagonal element of $S_{xx} - \hat{\mathcal{B}} \hat{\mathcal{B}}'$.

The second method, which can give answers that are considerably different from the principal component method is maximum likelihood. Upon further assumption that in (7.151), z and ϵ are multivariate normal, the log likelihood of \mathcal{B} and D ignoring a constant is

$$-2 \ln L(\mathcal{B}, D) = n \ln |\Sigma_{xx}| + \sum_{j=1}^n x'_j \Sigma_{xx}^{-1} x_j. \quad (7.157)$$

The likelihood depends on \mathcal{B} and D through (7.152), $\Sigma_{xx} = \mathcal{B} \mathcal{B}' + D$. As discussed in (7.153)-(7.154), the likelihood is not well defined because \mathcal{B} can be rotated. Typically, restricting $\mathcal{B} D^{-1} \mathcal{B}'$ to be a diagonal matrix is a computationally convenient uniqueness condition. The actual maximization of the likelihood is accomplished using numerical methods.

One obvious method of performing maximum likelihood for the Gaussian factor model is the EM algorithm. For example, suppose the factor vector z is known. Then, the factor model is simply the multivariate regression model given in Section 5.6, that is, write $X' = [x_1, x_2, \dots, x_n]$ and $Z' = [z_1, z_2, \dots, z_n]$, and note that X is $n \times p$ and Z is $n \times q$. Then, the MLE of \mathcal{B} is

$$\hat{\mathcal{B}} = X' Z (Z' Z)^{-1} = \left(n^{-1} \sum_{j=1}^n x_j z'_j \right) \left(n^{-1} \sum_{j=1}^n z_j z'_j \right)^{-1} \stackrel{\text{def}}{=} C_{xz} C_{zz}^{-1} \quad (7.158)$$

and the MLE of D is

$$\hat{D} = \text{diag} \left\{ n^{-1} \sum_{j=1}^n (x_j - \hat{\mathcal{B}} z_j) (x_j - \hat{\mathcal{B}} z_j)' \right\}; \quad (7.159)$$

that is, only the diagonal elements of the right-hand side of (7.159) are used. The bracketed quantity in (7.159) reduces to

$$C_{xx} - C_{xz} C_{zz}^{-1} C'_{xz}, \quad (7.160)$$

where $C_{xx} = n^{-1} \sum_{j=1}^n x_j x'_j$.

Based on the derivation of the EM algorithm for the state-space model, (4.66)–(4.75), we conclude that, to employ the EM algorithm here, given the current parameter estimates, in C_{xz} , we replace $x_j z'_j$ by $x_j \tilde{z}'_j$, where $\tilde{z}_j = E(z_j | x_j)$, and in C_{zz} , we replace $z_j z'_j$ by $P_z + \tilde{z}_j \tilde{z}'_j$, where $P_z = \text{var}(z_j | x_j)$. Using the fact that the $(p+q) \times 1$ vector $(x'_j, z'_j)'$ is multivariate normal with mean-zero, and variance–covariance matrix given by

$$\begin{pmatrix} \mathcal{B}\mathcal{B}' + D & \mathcal{B} \\ \mathcal{B}' & I_q \end{pmatrix}, \quad (7.161)$$

we have

$$\tilde{z}_j \equiv E(z_j \mid x_j) = \mathcal{B}'(\mathcal{B}'\mathcal{B} + D)^{-1}x_j \quad (7.162)$$

and

$$P_z \equiv \text{var}(z_j \mid x_j) = I_q - \mathcal{B}'(\mathcal{B}'\mathcal{B} + D)^{-1}\mathcal{B}. \quad (7.163)$$

For time series, suppose x_t is a stationary $p \times 1$ process with $p \times p$ spectral matrix $f_{xx}(\omega)$. Analogous to the classical model displayed in (7.152), we may postulate that at a given frequency of interest, ω , the spectral matrix of x_t satisfies

$$f_{xx}(\omega) = \mathcal{B}(\omega)\mathcal{B}(\omega)^* + D(\omega), \quad (7.164)$$

where $\mathcal{B}(\omega)$ is a complex-valued $p \times q$ matrix with $\text{rank}(\mathcal{B}(\omega)) = q \leq p$ and $D(\omega)$ is a real, nonnegative-definite, diagonal matrix. Typically, we expect q will be much smaller than p .

As an example of a model that gives rise to (7.164), let $x_t = (x_{t1}, \dots, x_{tp})'$, and suppose

$$x_{tj} = c_j s_{t-\tau_j} + \epsilon_{tj}, \quad j = 1, \dots, p, \quad (7.165)$$

where $c_j \geq 0$ are individual amplitudes and s_t is a common unobserved signal (factor) with spectral density $f_{ss}(\omega)$. The values τ_j are the individual phase shifts. Assume s_t is independent of $\epsilon_t = (\epsilon_{t1}, \dots, \epsilon_{tp})'$ and the spectral matrix of ϵ_t , $D_{\epsilon\epsilon}(\omega)$, is diagonal. The DFT of x_{tj} is given by

$$X_j(\omega) = n^{-1/2} \sum_{t=1}^n x_{tj} \exp(-2\pi i t \omega)$$

and, in terms of the model (7.165),

$$X_j(\omega) = a_j(\omega)X_s(\omega) + X_{\epsilon_j}(\omega), \quad (7.166)$$

where $a_j(\omega) = c_j \exp(-2\pi i \tau_j \omega)$, and $X_s(\omega)$ and $X_{\epsilon_j}(\omega)$ are the respective DFTs of the signal s_t and the noise ϵ_{tj} . Stacking the individual elements of (7.166), we obtain a complex version of the classical factor model with one factor,

$$\begin{pmatrix} X_1(\omega) \\ \vdots \\ X_p(\omega) \end{pmatrix} = \begin{pmatrix} a_1(\omega) \\ \vdots \\ a_p(\omega) \end{pmatrix} X_s(\omega) + \begin{pmatrix} X_{\epsilon_1}(\omega) \\ \vdots \\ X_{\epsilon_p}(\omega) \end{pmatrix},$$

or more succinctly,

$$X(\omega) = a(\omega)X_s(\omega) + X_{\epsilon}(\omega). \quad (7.167)$$

From (7.167), we can identify the spectral components of the model; that is,

$$f_{xx}(\omega) = b(\omega)b(\omega)^* + D_{\epsilon\epsilon}(\omega), \quad (7.168)$$

where $b(\omega)$ is a $p \times 1$ complex-valued vector, $b(\omega)b(\omega)^* = a(\omega)f_{ss}(\omega)a(\omega)^*$. Model (7.168) could be considered the one-factor model for time series. This model can be extended to more than one factor by adding other independent signals into the original model (7.165). More details regarding this and related models can be found in Stoffer (1999).

Example 7.14 Single Factor Analysis of the fMRI Data

The fMRI data analyzed in Example 7.13 is well suited for a single factor analysis using the model (7.165), or, equivalently, the complex-valued, single factor model (7.167). In terms of (7.165), we can think of the signal s_t as representing the brush stimulus signal. As before, the frequency of interest is $\omega = 4/128$, which corresponds to a period of 32 time points, or 64 seconds.

A simple way to estimate the components $b(\omega)$ and $D_{\epsilon\epsilon}(\omega)$, as specified in (7.168), is to use the principal components method. Let $\hat{f}_{xx}(\omega)$ denote the estimate of the spectral density of $x_t = (x_{t1}, \dots, x_{t8})'$ obtained in Example 7.13. Then, analogous to (7.155) and (7.156), we set

$$\hat{b}(\omega) = \sqrt{\hat{\lambda}_1(\omega)} \hat{e}_1(\omega),$$

where $(\hat{\lambda}_1(\omega), \hat{e}_1(\omega))$ is the first eigenvalue–eigenvector pair of $\hat{f}_{xx}(\omega)$. The diagonal elements of $\hat{D}_{\epsilon\epsilon}(\omega)$ are obtained from the diagonal elements of $\hat{f}_{xx}(\omega) - \hat{b}(\omega)\hat{b}(\omega)^*$. The appropriateness of the model can be assessed by checking the elements of the residual matrix, $\hat{f}_{xx}(\omega) - [\hat{b}(\omega)\hat{b}(\omega)^* + \hat{D}_{\epsilon\epsilon}(\omega)]$, are negligible in magnitude.

Concentrating on the stimulus frequency, recall $\hat{\lambda}_1(4/128) = 2$. The magnitudes of $\hat{e}_1(4/128)$ are displayed in Table 7.5, indicating all locations load on the stimulus factor except for location 6, and location 7 could be considered borderline. The diagonal elements of $\hat{f}_{xx}(\omega) - \hat{b}(\omega)\hat{b}(\omega)^*$ yield

$$\hat{D}_{\epsilon\epsilon}(4/128) = 0.001 \times \text{diag}\{1.36, 2.04, 6.22, 11.30, 0.73, 13.26, 6.93, 5.88\}.$$

The magnitudes of the elements of the residual matrix at $\omega = 4/128$ are

$$0.001 \times \begin{bmatrix} 0.00 & 1.73 & 3.88 & 3.61 & 0.88 & 2.04 & 1.60 & 2.81 \\ 2.41 & 0.00 & 1.17 & 3.77 & 1.49 & 5.58 & 3.68 & 4.21 \\ 8.49 & 5.34 & 0.00 & 2.94 & 7.58 & 10.91 & 8.36 & 10.64 \\ 12.65 & 11.84 & 6.12 & 0.00 & 12.56 & 14.64 & 13.34 & 16.10 \\ 0.32 & 0.29 & 2.10 & 2.01 & 0.00 & 1.18 & 2.01 & 1.18 \\ 10.34 & 16.69 & 17.09 & 15.94 & 13.49 & 0.00 & 5.78 & 14.74 \\ 5.71 & 8.51 & 8.94 & 10.18 & 7.56 & 0.97 & 0.00 & 8.66 \\ 6.25 & 8.00 & 10.31 & 10.69 & 5.95 & 8.69 & 7.64 & 0.00 \end{bmatrix},$$

indicating the model fit is good. Assuming the results of the previous example are available, use the following R code.

```
bhat = sqrt(lam[1])*evec[,1]
Dhat = Re(diag(fxx[,4] - bhat%*%Conj(t(bhat))))
res = Mod(fxx[,4] - Dhat - bhat%*%Conj(t(bhat)))
```

A number of authors have considered factor analysis in the spectral domain, for example Priestley et al. (1974); Priestley and Subba Rao (1975); Geweke (1977), and Geweke and Singleton (1981), to mention a few. An obvious extension of simple model (7.165) is the factor model

$$x_t = \sum_{j=-\infty}^{\infty} \Lambda_j s_{t-j} + \epsilon_t, \quad (7.169)$$

where $\{\Lambda_j\}$ is a real-valued $p \times q$ filter, s_t is a $q \times 1$ stationary, unobserved signal, with independent components, and ϵ_t is white noise. We assume the signal and noise process are independent, s_t has $q \times q$ real, diagonal spectral matrix $f_{ss}(\omega) = \text{diag}\{f_{s1}(\omega), \dots, f_{sq}(\omega)\}$, and ϵ_t has a real, diagonal, $p \times p$ spectral matrix given by $D_{\epsilon\epsilon}(\omega) = \text{diag}\{f_{\epsilon 1}(\omega), \dots, f_{\epsilon p}(\omega)\}$. If, in addition, $\sum \|\Lambda_j\| < \infty$, the spectral matrix of x_t can be written as

$$f_{xx}(\omega) = \Lambda(\omega) f_{ss}(\omega) \Lambda(\omega)^* + D_{\epsilon\epsilon}(\omega) = \mathcal{B}(\omega) \mathcal{B}(\omega)^* + D_{\epsilon\epsilon}(\omega), \quad (7.170)$$

where

$$\Lambda(\omega) = \sum_{t=-\infty}^{\infty} \Lambda_t \exp(-2\pi i t \omega) \quad (7.171)$$

and $\mathcal{B}(\omega) = \Lambda(\omega) f_{ss}^{1/2}(\omega)$. Thus, by (7.170), the model (7.169) is seen to satisfy the basic requirement of the spectral domain factor analysis model; that is, the $p \times p$ spectral density matrix of the process of interest, $f_{xx}(\omega)$, is the sum of a rank $q \leq p$ matrix, $\mathcal{B}(\omega) \mathcal{B}(\omega)^*$, and a real, diagonal matrix, $D_{\epsilon\epsilon}(\omega)$. For the purpose of identifiability we set $f_{ss}(\omega) = I_q$ for all ω ; in which case, $\mathcal{B}(\omega) = \Lambda(\omega)$. As in the classical case [see (7.154)], the model is specified only up to rotations; for details, see Bloomfield and Davis (1994).

Parameter estimation for the model (7.169), or equivalently (7.170), can be accomplished using the principal component method. Let $\hat{f}_{xx}(\omega)$ be an estimate of $f_{xx}(\omega)$, and let $(\hat{\lambda}_j(\omega), \hat{e}_j(\omega))$, for $j = 1, \dots, p$, be the eigenvalue–eigenvector pairs, in the usual order, of $\hat{f}_{xx}(\omega)$. Then, as in the classical case, the $p \times q$ matrix \mathcal{B} is estimated by

$$\hat{\mathcal{B}}(\omega) = \left[\hat{\lambda}_1(\omega)^{1/2} \hat{e}_1(\omega) \mid \hat{\lambda}_2(\omega)^{1/2} \hat{e}_2(\omega) \mid \cdots \mid \hat{\lambda}_q(\omega)^{1/2} \hat{e}_q(\omega) \right]. \quad (7.172)$$

The estimated diagonal spectral density matrix of errors is then obtained by setting $\hat{D}_{\epsilon\epsilon}(\omega) = \text{diag}\{\hat{f}_{\epsilon 1}(\omega), \dots, \hat{f}_{\epsilon p}(\omega)\}$, where $\hat{f}_{\epsilon j}(\omega)$ is the j -th diagonal element of $\hat{f}_{xx}(\omega) - \hat{\mathcal{B}}(\omega) \hat{\mathcal{B}}(\omega)^*$.

Alternatively, we can estimate the parameters by approximate likelihood methods. As in (7.167), let $X(\omega_j)$ denote the DFT of the data x_1, \dots, x_n at frequency $\omega_j = j/n$. Similarly, let $X_s(\omega_j)$ and $X_\epsilon(\omega_j)$ be the DFTs of the signal and of the noise processes, respectively. Then, under certain conditions (see Pawitan and Shumway, 1989), for $\ell = 0, \pm 1, \dots, \pm m$,

$$X(\omega_j + \ell/n) = \Lambda(\omega_j) X_s(\omega_j + \ell/n) + X_\epsilon(\omega_j + \ell/n) + o_{as}(n^{-\alpha}), \quad (7.173)$$

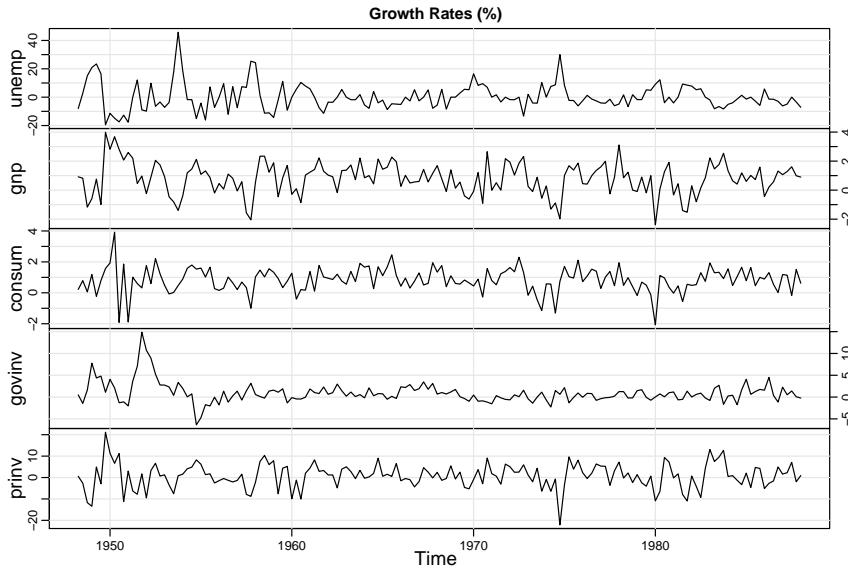


Fig. 7.18. The seasonally adjusted, quarterly growth rate (as percentages) of five macroeconomic series, unemployment, GNP, consumption, government investment, and private investment in the United States between 1948 and 1988, $n = 160$ values.

where $\Lambda(\omega_j)$ is given by (7.171) and $o_{as}(n^{-\alpha}) \rightarrow 0$ almost surely for some $0 \leq \alpha < 1/2$ as $n \rightarrow \infty$. In (7.173), the $X(\omega_j + \ell/n)$ are the DFTs of the data at the L odd frequencies $\{\omega_j + \ell/n; \ell = 0, \pm 1, \dots, \pm m\}$ surrounding the central frequency of interest $\omega_j = j/n$.

Under appropriate conditions $\{X(\omega_j + \ell/n); \ell = 0, \pm 1, \dots, \pm m\}$ in (7.173) are approximately ($n \rightarrow \infty$) independent, complex Gaussian random vectors with variance-covariance matrix $f_{xx}(\omega_j)$. The approximate likelihood is given by

$$\begin{aligned} -2 \ln L(\mathcal{B}(\omega_j), D_{\epsilon\epsilon}(\omega_j)) \\ = n \ln |f_{xx}(\omega_j)| + \sum_{\ell=-m}^m X^*(\omega_j + \ell/n) f_{xx}^{-1}(\omega_j) X(\omega_j + \ell/n), \end{aligned} \quad (7.174)$$

with the constraint $f_{xx}(\omega_j) = \mathcal{B}(\omega_j)\mathcal{B}(\omega_j)^* + D_{\epsilon\epsilon}(\omega_j)$. As in the classical case, we can use various numerical methods to maximize $L(\mathcal{B}(\omega_j), D_{\epsilon\epsilon}(\omega_j))$ at every frequency, ω_j , of interest. For example, the EM algorithm discussed for the classical case, (7.158)–(7.163), can easily be extended to this case.

Assuming $f_{ss}(\omega) = I_q$, the estimate of $\mathcal{B}(\omega_j)$ is also the estimate of $\Lambda(\omega_j)$. Calling this estimate $\hat{\Lambda}(\omega_j)$, the time domain filter can be estimated by

$$\hat{\Lambda}_t^M = M^{-1} \sum_{j=0}^{M-1} \hat{\Lambda}(\omega_j) \exp(2\pi i jt/n), \quad (7.175)$$

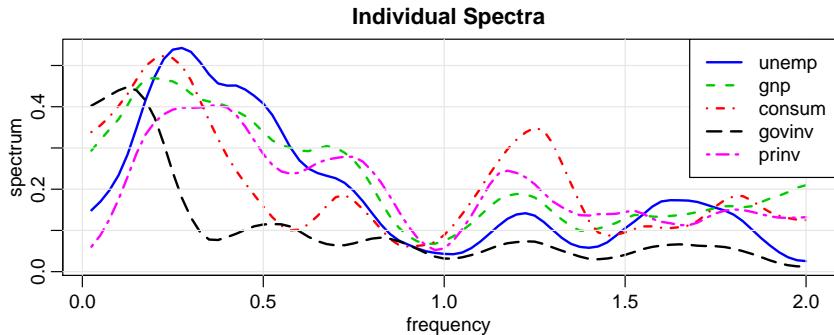


Fig. 7.19. The individual estimated spectra (scaled by 1000) of each series show in Figure 7.18 in terms of the number of cycles in 160 quarters.

for some $0 < M \leq n$, which is the discrete and finite version of the inversion formula given by

$$\Lambda_t = \int_{-1/2}^{1/2} \Lambda(\omega) \exp(2\pi i \omega t) d\omega. \quad (7.176)$$

Note that we have used this approximation earlier in Chapter 4, (4.124), for estimating the time response of a frequency response function defined over a finite number of frequencies.

Example 7.15 Government Spending, Private Investment, and Unemployment

Figure 7.18 shows the seasonally adjusted, quarterly growth rate (as percentages) of five macroeconomic series, unemployment, GNP, consumption, government investment, and private investment in the United States between 1948 and 1988, $n = 160$ values. These data are analyzed in the time domain by Young and Pedregal (1998), who were investigating how government spending and private capital investment influenced the rate of unemployment.

Spectral estimation was performed on the detrended, standardized, and tapered growth rate values; see the R code at the end of this example for details. Figure 7.19 shows the individual estimated spectra of each series. We focus on three interesting frequencies. First, we note the lack of spectral power near the annual cycle ($\omega = 1$, or one cycle every four quarters), indicating the data have been seasonally adjusted. In addition, because of the seasonal adjustment, some spectral power appears near the seasonal frequency; this is a distortion apparently caused by the method of seasonally adjusting the data. Next, we note the spectral power near $\omega = .25$, or one cycle every four years, in unemployment, GNP, consumption, and, to lesser degree, in private investment. Finally, spectral power appears near $\omega = .125$, or one cycle every eight years in government investment, and perhaps to lesser degrees in unemployment, GNP, and consumption.

Figure 7.20 shows the coherences among various series. At the frequencies of interest, $\omega = .125$ and $.25$, pairwise, GNP, Unemployment, Consumption, and Private Investment (except for Unemployment and Private Investment) are coherent.

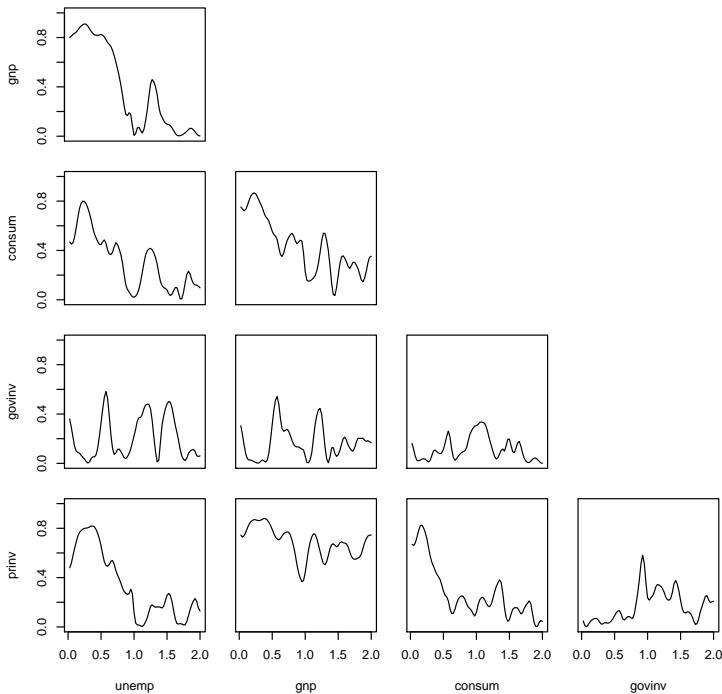


Fig. 7.20. The squared coherencies between the various series displayed in Figure 7.18.

Government Investment is either not coherent or minimally coherent with the other series.

Figure 7.21 shows $\hat{\lambda}_1(\omega)$ and $\hat{\lambda}_2(\omega)$, the first and second eigenvalues of the estimated spectral matrix $\hat{f}_{xx}(\omega)$. These eigenvalues suggest the first factor is identified by the frequency of one cycle every four years, whereas the second factor is identified by the frequency of one cycle every eight years. The modulus of the corresponding eigenvectors at the frequencies of interest, $\hat{e}_1(10/160)$ and $\hat{e}_2(5/160)$, are shown in Table 7.6. These values confirm Unemployment, GNP, Consumption, and Private Investment load on the first factor, and Government Investment loads on the second factor. The remainder of the details involving the factor analysis of these data is left as an exercise.

The following code was used to perform the analysis in R.

```
gr = diff(log(ts(econ5, start=1948, frequency=4))) # growth rate
plot(100*gr, main="Growth Rates (%)")
# scale each series to have variance 1
gr = ts(apply(gr, 2, scale), freq=4) # scaling strips ts attributes
L = c(7,7) # degree of smoothing
gr.spec = mvspec(gr, spans=L, demean=FALSE, detrend=FALSE, taper=.25)
dev.new()
plot(kernel("modified.daniell", L)) # view the kernel - not shown
dev.new()
```

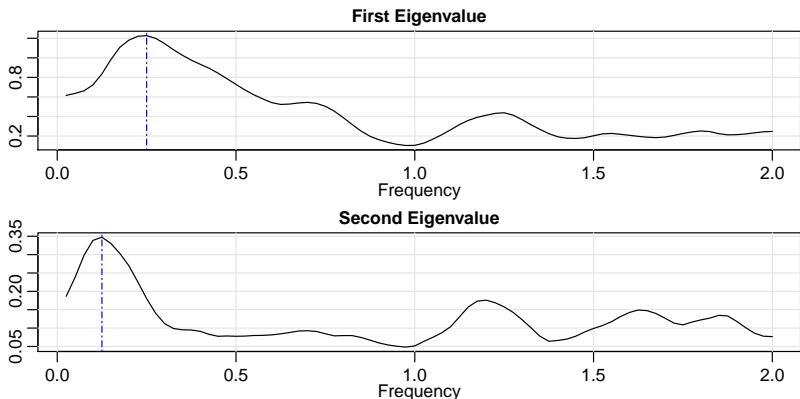


Fig. 7.21. The first, $\hat{\lambda}_1(\omega)$, and second, $\hat{\lambda}_2(\omega)$, eigenvalues of the estimated spectral matrix $\hat{f}_{xx}(\omega)$. The vertical dashed lines at the peaks are $\omega = .25$ and $.125$, respectively.

Table 7.6. Magnitudes of the Eigenvectors in [Example 7.15](#)

	Unemp	GNP	Cons	G. Inv.	P. Inv.
$ \hat{e}_1(\frac{10}{160}) $	0.53	0.50	0.51	0.06	0.44
$ \hat{e}_2(\frac{5}{160}) $	0.19	0.14	0.23	0.93	0.16

```

plot(gr.spec, log="no", main="Individual Spectra", lty=1:5, lwd=2)
legend("topright", colnames(econ5), lty=1:5, lwd=2)
dev.new()
plot.spec.coherency(gr.spec, ci=NA, main="Squared Coherencies")
# PCs
n.freq = length(gr.spec$freq)
lam = matrix(0,n.freq,5)
for (k in 1:n.freq) lam[,k] = eigen(gr.spec$fx[,k], symmetric=TRUE,
only.values=TRUE)$values
dev.new()
par(mfrow=c(2,1), mar=c(4,2,2,1), mgp=c(1.6,.6,0))
plot(gr.spec$freq, lam[,1], type="l", ylab="", xlab="Frequency", main="First
Eigenvalue")
abline(v=.25, lty=2)
plot(gr.spec$freq, lam[,2], type="l", ylab="", xlab="Frequency",
main="Second Eigenvalue")
abline(v=.125, lty=2)
e.vec1 = eigen(gr.spec$fx[,10], symmetric=TRUE)$vectors[,1]
e.vec2 = eigen(gr.spec$fx[,5], symmetric=TRUE)$vectors[,2]
round(Mod(e.vec1), 2); round(Mod(e.vec2), 3)

```

Table 7.7. Per Minute Infant EEG Sleep States
(read down and across)

REM	NR2	NR4	NR2	NR1	NR2	NR3	NR4	NR1	NR1	REM
REM	REM	NR4	NR1	NR1	NR2	NR4	NR4	NR1	NR1	REM
REM	REM	NR4	NR1	NR1	REM	NR4	NR4	NR1	NR1	REM
REM	NR3	NR4	NR1	REM	REM	NR4	NR4	NR1	NR1	REM
REM	NR4	NR4	NR1	REM	REM	NR4	NR4	NR1	NR1	REM
REM	NR4	NR4	NR1	REM	REM	NR4	NR4	NR1	NR1	REM
REM	NR4	NR4	NR2	REM	NR2	NR4	NR4	NR1	NR1	NR2
REM	NR4	NR4	REM	REM	NR2	NR4	NR4	NR1	REM	
NR2	NR4	NR4	NR1	REM	NR2	NR4	NR4	NR1	REM	
REM	NR2	NR4	NR1	REM	NR3	NR4	NR2	NR1	REM	

7.9 The Spectral Envelope

The concept of spectral envelope for the spectral analysis and *scaling* of categorical time series was first introduced in Stoffer et al. (1993). Since then, the idea has been extended in various directions (not only restricted to categorical time series), and we will explore these problems as well. First, we give a brief introduction to the concept of scaling time series.

The spectral envelope was motivated by collaborations with researchers who collected categorical-valued time series with an interest in the cyclic behavior of the data. For example Table 7.7 shows the per minute sleep-state of an infant taken from a study on the effects of prenatal exposure to alcohol. Details can be found in Stoffer et al. (1988), but briefly, an electroencephalographic (EEG) sleep recording of approximately two hours is obtained on a full term infant 24 to 36 hours after birth, and the recording is scored by a pediatric neurologist for sleep state. There are two main types of sleep, Non-Rapid Eye Movement (non-REM), also known as *quiet sleep* and Rapid Eye Movement (REM), also known as *active sleep*. In addition, there are four stages of non-REM (NR1-NR4), with NR1 being the “most active” of the four states, and finally awake (AW), which naturally occurs briefly through the night. This particular infant was never awake during the study.

It is not too difficult to notice a pattern in the data if one concentrates on REM versus non-REM sleep states. But, it would be difficult to try to assess patterns in a longer sequence, or if there were more categories, without some graphical aid. One simple method would be to *scale* the data, that is, *assign numerical values to the categories* and then draw a time plot of the scales. Because the states have an order, one obvious scaling is

$$\text{NR4} = 1, \quad \text{NR3} = 2, \quad \text{NR2} = 3, \quad \text{NR1} = 4, \quad \text{REM} = 5, \quad \text{AW} = 6, \quad (7.177)$$

and Figure 7.22 shows the time plot using this scaling. Another interesting scaling might be to combine the quiet states and the active states:

$$\text{NR4} = \text{NR3} = \text{NR2} = \text{NR1} = 0, \quad \text{REM} = 1, \quad \text{AW} = 2. \quad (7.178)$$

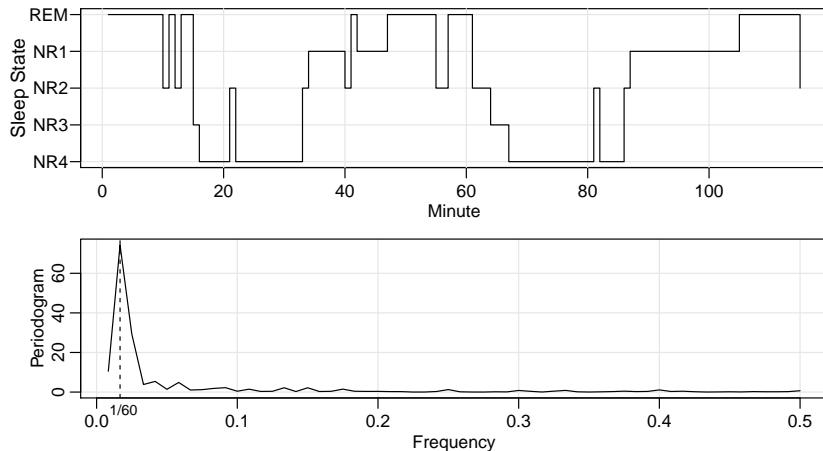


Fig. 7.22. [Top] Time plot of the EEG sleep state data in Table 7.7 using the scaling in (7.177). [Bottom] Periodogram of the EEG sleep state data in Figure 7.22 based on the scaling in (7.177). The peak corresponds to a frequency of approximately one cycle every 60 minutes.

The time plot using (7.178) would be similar to Figure 7.22 as far as the cyclic (in and out of quiet sleep) behavior of this infant's sleep pattern. Figure 7.22 shows the periodogram of the sleep data using the scaling in (7.177). A large peak exists at the frequency corresponding to one cycle every 60 minutes. As we might imagine, the general appearance of the periodogram using the scaling (7.178) (not shown) is similar to Figure 7.22. Most of us would feel comfortable with this analysis even though we made an arbitrary and ad hoc choice about the particular scaling. It is evident from the data (without any scaling) that if the interest is in infant sleep cycling, this particular sleep study indicates an infant cycles between active and quiet sleep at a rate of about one cycle per hour.

The intuition used in the previous example is lost when we consider a long DNA sequence. Briefly, a DNA strand can be viewed as a long string of linked nucleotides. Each nucleotide is composed of a nitrogenous base, a five carbon sugar, and a phosphate group. There are four different bases, and they can be grouped by size; the pyrimidines, thymine (T) and cytosine (C), and the purines, adenine (A) and guanine (G). The nucleotides are linked together by a backbone of alternating sugar and phosphate groups with the 5' carbon of one sugar linked to the 3' carbon of the next, giving the string direction. DNA molecules occur naturally as a double helix composed of polynucleotide strands with the bases facing inwards. The two strands are complementary, so it is sufficient to represent a DNA molecule by a sequence of bases on a single strand. Thus, a strand of DNA can be represented as a sequence of letters, termed base pairs (bp), from the finite alphabet {A, C, G, T}. The order of the nucleotides contains the genetic information specific to the organism. Expression of information stored in these molecules is a complex multistage process. One important task is to translate the information stored in the protein-coding sequences (CDS) of

Table 7.8. Part of the Epstein–Barr Virus DNA Sequence
(read across and down)

```

AGAATTCGTC TTGCTCTATT CACCCTTACT TTTCTTCTTG CCCGTTCTCT TTCTTAGTAT
GAATCCAGTA TGCCTGCTG TAATTTGTC GCCCTACCTC TTTGGCTGG CGGCTATIGC
CGCCTCGTGT TTCACGGCCT CAGTTAGTAC CGTTGTGACC GCCACCGGCT TGGCCCTCTC
ACTTCTACTC TTGGCAGCAG TGCCGACTC ATATGCCGT GCACAAAGGA AACTGCTGAC
ACCGGTGACA GTGCTTACTG CGGGTGTAC TTGTAAGTAC ACACGACCA TTTACAATGC
ATGATGTTG TGAGATTTGAT CTGCTCTAA CAGTTCACTT CCTCTGCTTT TCTCCTCAGT
CTTGCAATT TGCCTAACAT GGAGGATTGA GGACCCACCT TTTAATTCTC TTCTGTTGC
ATTGCTGGCC GCAGCTGGCG GACTACAAGG CATTACGGT TAGTGTGCCT CTGTTATGAA
ATGCAGGTTT GACTTCATAT GTATGCCCTG GCATGACGTC AACTTACTT TTATTTCACT
TCTGGTATG CTTGCTCC TGATACTAGC GTACAGAAGG AGATGGCGCC GTTGACTGT
TTGTCGGCGC ATCATGTTT TGGCATGTGT ACTTGTCTC ATCGTCGACG CTGTTTGCA
GCTGAGTCCC CTCCCTGGAG CTGTAACCTG GTTTCCATG ACGCTGCTGC TACTGGCTTT
CGTCCTCTGG CTCTCTCGC CAGGGGGCT AGGTACTCTT GGTGCAGCCC TTTAAACATT
GGCAGCAGGT AAGCCACACG TGTGACATTG CTTGGCTTT TGCCACATGT TTCTGGACA
CAGGACTAAC CATGCCATCT CTGATTATAG CTCTGGACT GCTAGCGTCA CTGATTTGG
GCACACTTAA CTTGACTACA ATGTTCTTC TCATGCTCT ATGGACACTT GGTAAAGTTT
CCCTCCCTTAA AACTCATTAC TTGTTCTTTT GTAATCGAG CTCTAACTTG GCATCTCTT
TACAGTGGTTT CTCTGATTG GCTCTCGTCTCTCATGT CCACTGAGCA AGATCCCTT

```

the DNA. A common problem in analyzing long DNA sequence data is in identifying CDS dispersed throughout the sequence and separated by regions of noncoding (which makes up most of the DNA). Table 7.8 shows part of the Epstein–Barr virus (EBV) DNA sequence. The entire EBV DNA sequence consists of approximately 172,000 bp.

We could try scaling according to the purine–pyrimidine alphabet, that is $A = G = 0$ and $C = T = 1$, but this is not necessarily of interest for every CDS of EBV. Numerous possible alphabets of interest exist. For example, we might focus on the strong–weak hydrogen-bonding alphabet $C = G = 0$ and $A = T = 1$. Although model calculations as well as experimental data strongly agree that some kind of periodic signal exists in certain DNA sequences, a large disagreement about the exact type of periodicity exists. In addition, a disagreement exists about which nucleotide alphabets are involved in the signals.

If we consider the naive approach of arbitrarily assigning numerical values (scales) to the categories and then proceeding with a spectral analysis, the result will depend on the particular assignment of numerical values. For example, consider the artificial sequence ACGTACGTACGT... . Then, setting $A = G = 0$ and $C = T = 1$ yields the numerical sequence 010101010101..., or one cycle every two base pairs. Another interesting scaling is $A = 1, C = 2, G = 3$, and $T = 4$, which results in the sequence 123412341234..., or one cycle every four bp. In this example, both scalings (that is, $\{A, C, G, T\} = \{0, 1, 0, 1\}$ and $\{A, C, G, T\} = \{1, 2, 3, 4\}$) of the nucleotides are interesting and bring out different properties of the sequence. Hence, we do not want to focus on only one scaling. Instead, the focus should be on finding all possible scalings that bring out all of the interesting features in the data. Rather than choose values arbitrarily, the spectral envelope approach selects scales that help emphasize any periodic feature that exists in a categorical time series of virtually any length in

a quick and automated fashion. In addition, the technique can help in determining whether a sequence is merely a random assignment of categories.

THE SPECTRAL ENVELOPE FOR CATEGORICAL TIME SERIES

As a general description, the spectral envelope is a frequency-based principal components technique applied to a multivariate time series. First, we will focus on the basic concept and its use in the analysis of categorical time series. Technical details can be found in Stoffer et al. (1993).

Briefly, in establishing the spectral envelope for categorical time series, the basic question of how to efficiently discover periodic components in categorical time series was addressed. This, was accomplished via nonparametric spectral analysis as follows. Let x_t , $t = 0, \pm 1, \pm 2, \dots$, be a categorical-valued time series with finite state-space $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$. Assume x_t is stationary and $p_j = \Pr\{x_t = c_j\} > 0$ for $j = 1, 2, \dots, k$. For $\beta = (\beta_1, \beta_2, \dots, \beta_k)' \in \mathbb{R}^k$, denote by $x_t(\beta)$ the real-valued stationary time series corresponding to the scaling that assigns the category c_j the numerical value β_j , $j = 1, 2, \dots, k$. The spectral density of $x_t(\beta)$ will be denoted by $f_{xx}(\omega; \beta)$. The goal is to find scalings β , so the spectral density is in some sense interesting, and to summarize the spectral information by what is called the spectral envelope.

In particular, β is chosen to maximize the power at each frequency, ω , of interest, relative to the total power $\sigma^2(\beta) = \text{var}\{x_t(\beta)\}$. That is, we chose $\beta(\omega)$, at each ω of interest, so

$$\lambda(\omega) = \max_{\beta} \left\{ \frac{f_{xx}(\omega; \beta)}{\sigma^2(\beta)} \right\}, \quad (7.179)$$

over all β not proportional to 1_k , the $k \times 1$ vector of ones. Note, $\lambda(\omega)$ is not defined if $\beta = a1_k$ for $a \in \mathbb{R}$ because such a scaling corresponds to assigning each category the same value a ; in this case, $f_{xx}(\omega; \beta) \equiv 0$ and $\sigma^2(\beta) = 0$. The optimality criterion $\lambda(\omega)$ possesses the desirable property of being invariant under location and scale changes of β .

As in most scaling problems for categorical data, it is useful to represent the categories in terms of the unit vectors u_1, u_2, \dots, u_k , where u_j represents the $k \times 1$ vector with a one in the j -th row, and zeros elsewhere. We then define a k -dimensional stationary time series y_t by $y_t = u_j$ when $x_t = c_j$. The time series $x_t(\beta)$ can be obtained from the y_t time series by the relationship $x_t(\beta) = \beta'y_t$. Assume the vector process y_t has a continuous spectral density denoted by $f_{yy}(\omega)$. For each ω , $f_{yy}(\omega)$ is, of course, a $k \times k$ complex-valued Hermitian matrix. The relationship $x_t(\beta) = \beta'y_t$ implies $f_{xx}(\omega; \beta) = \beta'f_{yy}(\omega)\beta = \beta'f_{yy}^{re}(\omega)\beta$, where $f_{yy}^{re}(\omega)$ denotes the real part^{7.2} of $f_{yy}(\omega)$. The imaginary part disappears from the expression because it is skew-symmetric, that is, $f_{yy}^{im}(\omega)' = -f_{yy}^{im}(\omega)$. The optimality criterion can thus be expressed as

$$\lambda(\omega) = \max_{\beta} \left\{ \frac{\beta'f_{yy}^{re}(\omega)\beta}{\beta'V\beta} \right\}, \quad (7.180)$$

^{7.2} In this section, it is more convenient to write complex values in the form $z = z^{re} + iz^{im}$, which represents a change from the notation used previously.

where V is the variance–covariance matrix of y_t . The resulting scaling $\beta(\omega)$ is called the optimal scaling.

The y_t process is a multivariate point process, and any particular component of y_t is the individual point process for the corresponding state (for example, the first component of y_t indicates whether the process is in state c_1 at time t). For any fixed t , y_t represents a single observation from a simple multinomial sampling scheme. It readily follows that $V = D - p p'$, where $p = (p_1, \dots, p_k)'$, and D is the $k \times k$ diagonal matrix $D = \text{diag}\{p_1, \dots, p_k\}$. Because, by assumption, $p_j > 0$ for $j = 1, 2, \dots, k$, it follows that $\text{rank}(V) = k - 1$ with the null space of V being spanned by 1_k . For any $k \times (k - 1)$ full rank matrix Q whose columns are linearly independent of 1_k , $Q'VQ$ is a $(k - 1) \times (k - 1)$ positive-definite symmetric matrix.

With the matrix Q as previously defined, define $\lambda(\omega)$ to be the largest eigenvalue of the determinantal equation

$$|Q' f_{yy}^{re}(\omega) Q - \lambda(\omega) Q' V Q| = 0,$$

and let $b(\omega) \in \mathbb{R}^{k-1}$ be any corresponding eigenvector, that is,

$$Q' f_{yy}^{re}(\omega) Q b(\omega) = \lambda(\omega) Q' V Q b(\omega).$$

The eigenvalue $\lambda(\omega) \geq 0$ does not depend on the choice of Q . Although the eigenvector $b(\omega)$ depends on the particular choice of Q , the equivalence class of scalings associated with $\beta(\omega) = Qb(\omega)$ does not depend on Q . A convenient choice of Q is $Q = [I_{k-1} \mid 0]'$, where I_{k-1} is the $(k - 1) \times (k - 1)$ identity matrix and 0 is the $(k - 1) \times 1$ vector of zeros. For this choice, $Q' f_{yy}^{re}(\omega) Q$ and $Q' V Q$ are the upper $(k - 1) \times (k - 1)$ blocks of $f_{yy}^{re}(\omega)$ and V , respectively. This choice corresponds to setting the last component of $\beta(\omega)$ to zero.

The value $\lambda(\omega)$ itself has a useful interpretation; specifically, $\lambda(\omega)d\omega$ represents the largest proportion of the total power that can be attributed to the frequencies $(\omega, \omega + d\omega)$ for any particular scaled process $x_t(\beta)$, with the maximum being achieved by the scaling $\beta(\omega)$. Because of its central role, $\lambda(\omega)$ is defined to be the *spectral envelope of a stationary categorical time series*.

The name spectral envelope is appropriate since $\lambda(\omega)$ envelopes the standardized spectrum of any scaled process. That is, given any β normalized so that $x_t(\beta)$ has total power one, $f_{xx}(\omega ; \beta) \leq \lambda(\omega)$ with equality if and only if β is proportional to $\beta(\omega)$.

Given observations x_t , for $t = 1, \dots, n$, on a categorical time series, we form the multinomial point process y_t , for $t = 1, \dots, n$. Then, the theory for estimating the spectral density of a multivariate, real-valued time series can be applied to estimating $f_{yy}(\omega)$, the $k \times k$ spectral density of y_t . Given an estimate $\hat{f}_{yy}(\omega)$ of $f_{yy}(\omega)$, estimates $\hat{\lambda}(\omega)$ and $\hat{\beta}(\omega)$ of the spectral envelope, $\lambda(\omega)$, and the corresponding scalings, $\beta(\omega)$, can then be obtained. Details on estimation and inference for the sample spectral envelope and the optimal scalings can be found in Stoffer et al. (1993), but the main result of that paper is as follows: If $\hat{f}_{yy}(\omega)$ is a consistent spectral estimator and if for each $j = 1, \dots, J$, the largest root of $f_{yy}^{re}(\omega_j)$ is distinct, then

$$\{\eta_n[\hat{\lambda}(\omega_j) - \lambda(\omega_j)]/\lambda(\omega_j), \eta_n[\hat{\beta}(\omega_j) - \beta(\omega_j)]; j = 1, \dots, J\} \quad (7.181)$$

converges ($n \rightarrow \infty$) jointly in distribution to independent zero-mean, normal, distributions, the first of which is standard normal; the asymptotic covariance structure of $\hat{\beta}(\omega_j)$ is discussed in Stoffer et al. (1993). Result (7.181) is similar to (7.148), but in this case, $\beta(\omega)$ and $\hat{\beta}(\omega)$ are real. The term η_n is the same as in (7.181), and its value depends on the type of estimator being used. Based on these results, asymptotic normal confidence intervals and tests for $\lambda(\omega)$ can be readily constructed. Similarly, for $\beta(\omega)$, asymptotic confidence ellipsoids and chi-square tests can be constructed; details can be found in Stoffer et al. (1993, Theorems 3.1 – 3.3).

Peak searching for the smoothed spectral envelope estimate can be aided using the following approximations. Using a first-order Taylor expansion, we have

$$\log \hat{\lambda}(\omega) \approx \log \lambda(\omega) + \frac{\hat{\lambda}(\omega) - \lambda(\omega)}{\lambda(\omega)}, \quad (7.182)$$

so $\eta_n[\log \hat{\lambda}(\omega) - \log \lambda(\omega)]$ is approximately standard normal. It follows that $E[\log \hat{\lambda}(\omega)] \approx \log \lambda(\omega)$ and $\text{var}[\log \hat{\lambda}(\omega)] \approx \eta_n^{-2}$. If no signal is present in a sequence of length n , we expect $\lambda(j/n) \approx 2/n$ for $1 < j < n/2$, and hence approximately $(1 - \alpha) \times 100\%$ of the time, $\log \hat{\lambda}(\omega)$ will be less than $\log(2/n) + (z_\alpha/\eta_n)$, where z_α is the $(1 - \alpha)$ upper tail cutoff of the standard normal distribution. Exponentiating, the α critical value for $\hat{\lambda}(\omega)$ becomes $(2/n) \exp(z_\alpha/\eta_n)$. Useful values of z_α are $z_{.001} = 3.09$, $z_{.0001} = 3.71$, and $z_{.00001} = 4.26$, and from our experience, thresholding at these levels works well.

Example 7.16 Spectral Analysis of DNA Sequences

To help understand the methodology, we give explicit instructions for the calculations involved in estimating the spectral envelope of a DNA sequence, x_t , for $t = 1, \dots, n$, using the nucleotide alphabet.

- (i) In this example, we hold the scale for T fixed at zero. In this case, we form the 3×1 data vectors y_t :

$$\begin{aligned} y_t = (1, 0, 0)' &\text{ if } x_t = A; & y_t = (0, 1, 0)' &\text{ if } x_t = C; \\ y_t = (0, 0, 1)' &\text{ if } x_t = G; & y_t = (0, 0, 0)' &\text{ if } x_t = T. \end{aligned}$$

The scaling vector is $\beta = (\beta_1, \beta_2, \beta_3)'$, and the scaled process is $x_t(\beta) = \beta'y_t$.

- (ii) Calculate the DFT of the data

$$Y(j/n) = n^{-1/2} \sum_{t=1}^n y_t \exp(-2\pi i t j/n).$$

Note $Y(j/n)$ is a 3×1 complex-valued vector. Calculate the periodogram, $I(j/n) = Y(j/n)Y^*(j/n)$, for $j = 1, \dots, [n/2]$, and retain only the real part, say, $I^{re}(j/n)$.

- (iii) Smooth the $I^{re}(j/n)$ to obtain an estimate of $f_{yy}^{re}(j/n)$. Let $\{h_k; k = 0, \pm 1, \dots, \pm m\}$ be weights as described in (4.64). Calculate

$$\hat{f}_{yy}^{re}(j/n) = \sum_{k=-m}^m h_k I^{re}(j/n + k/n).$$

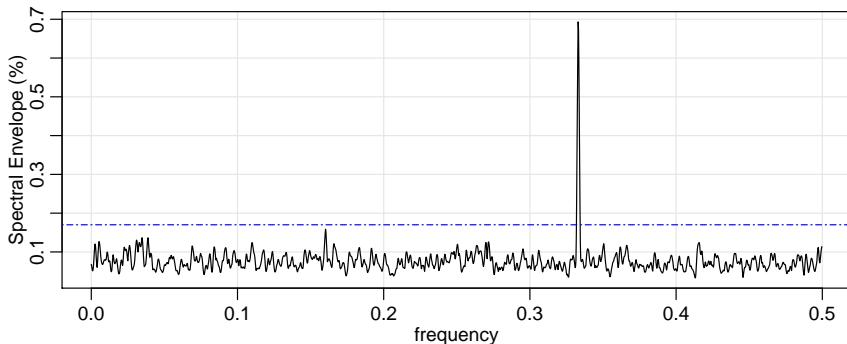


Fig. 7.23. Smoothed sample spectral envelope of the BNRF1 gene from the Epstein–Barr virus.

(iv) Calculate the 3×3 sample variance–covariance matrix,

$$S_{yy} = n^{-1} \sum_{t=1}^n (y_t - \bar{y})(y_t - \bar{y})'$$

where $\bar{y} = n^{-1} \sum_{t=1}^n y_t$ is the sample mean of the data.

- (v) For each $\omega_j = j/n$, $j = 0, 1, \dots, [n/2]$, determine the largest eigenvalue and the corresponding eigenvector of the matrix $2n^{-1} S_{yy}^{-1/2} \hat{f}_{yy}^{re}(\omega_j) S_{yy}^{-1/2}$. Note, $S_{yy}^{1/2}$ is the unique square root matrix of S_{yy} .
- (vi) The sample spectral envelope $\hat{\lambda}(\omega_j)$ is the eigenvalue obtained in the previous step. If $b(\omega_j)$ denotes the eigenvector obtained in the previous step, the optimal sample scaling is $\hat{\beta}(\omega_j) = S_{yy}^{-1/2} b(\omega_j)$; this will result in three values, the value corresponding to the fourth category, T, being held fixed at zero.

Example 7.17 Analysis of an Epstein–Barr Virus Gene

In this example, we focus on a dynamic (or sliding-window) analysis of the gene labeled BNRF1 (bp 1736–5689) of Epstein–Barr. Figure 7.23 shows the spectral envelope estimate of the entire coding sequence (3954 bp long). The figure also shows a strong signal at frequency 1/3; the corresponding optimal scaling was $A = .10$, $C = .61$, $G = .78$, $T = 0$, which indicates the signal is in the strong–weak bonding alphabet, $S = \{C, G\}$ and $W = \{A, T\}$.

Figure 7.24 shows the result of computing the spectral envelope over three nonoverlapping 1000-bp windows and one window of 954 bp, across the CDS, namely, the first, second, third, and fourth quarters of BNRF1. An approximate 0.0001 significance threshold is .69%. The first three quarters contain the signal at the frequency 1/3 (Figure 7.24a–c); the corresponding sample optimal scalings for the first three windows were (a) $A = .01$, $C = .71$, $G = .71$, $T = 0$; (b) $A = .08$, $C = .71$, $G = .70$, $T = 0$; (c) $A = .20$, $C = .58$, $G = .79$, $T = 0$. The first two windows are consistent with the overall analysis. The third section, however, shows some minor

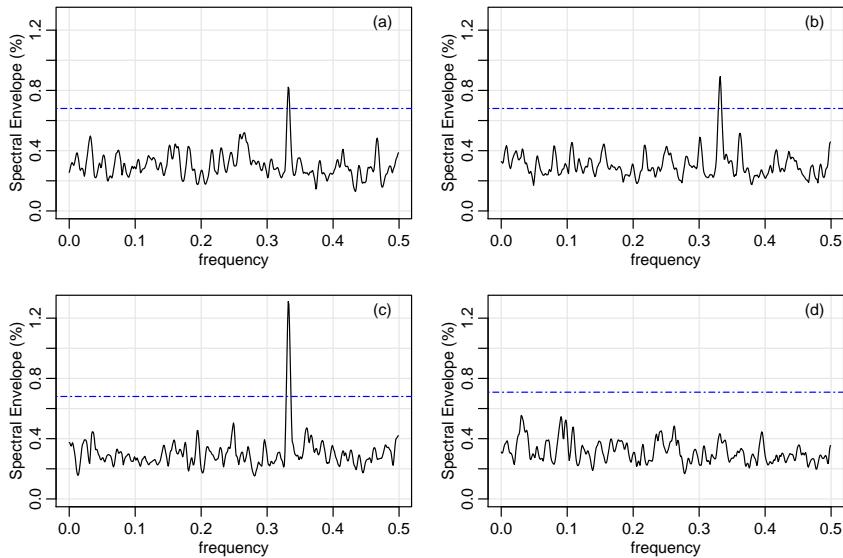


Fig. 7.24. Smoothed sample spectral envelope of the BNRF1 gene from the Epstein–Barr virus: (a) first 1000 bp, (b) second 1000 bp, (c) third 1000 bp, and (d) last 954 bp.

departure from the strong-weak bonding alphabet. The most interesting outcome is that the fourth window shows that no signal is present. This leads to the conjecture that the fourth quarter of BNRF1 of Epstein–Barr is actually noncoding.

The R code for the first part of the example is as follows.

```
u = factor(bnrf1ebv) # first, input the data as factors and then
x = model.matrix(~u-1)[,1:3] # make an indicator matrix
# x = x[1:1000,] # select subsequence if desired
Var = var(x) # var-cov matrix
xspec = mvspec(x, spans=c(7,7), plot=FALSE)
fxxr = Re(xspec$fxx) # fxxr is real(fxx)
# compute Q = Var^-1/2
ev = eigen(Var)
Q = ev$vectors %*% diag(1/sqrt(ev$values)) %*% t(ev$vectors)
# compute spec envelope and scale vectors
num = xspec$n.used # sample size used for FFT
nfreq = length(xspec$freq) # number of freqs used
specenv = matrix(0,nfreq,1) # initialize the spec envelope
beta = matrix(0,nfreq,3) # initialize the scale vectors
for (k in 1:nfreq){
  ev = eigen(2*Q %*% fxxr[,k] %*% Q/num, symmetric=TRUE)
  specenv[k] = ev$values[1] # spec env at freq k/n is max eval
  b = Q %*% ev$vectors[,1] # beta at freq k/n
  beta[,k] = b/sqrt(sum(b^2)) } # helps to normalize beta
# output and graphics
frequency = xspec$freq
plot(frequency, 100*specenv, type="l", ylab="Spectral Envelope (%)")
# add significance threshold to plot
```

```

m = xspec$kernel$m
etainv = sqrt(sum(xspec$kernel[-m:m]^2))
thresh=100*(2/num)*exp(qnorm(.9999)*etainv)
abline(h=thresh, lty=6, col=4)
# details
output = cbind(frequency, specenv, beta)
colnames(output) = c("freq", "specenv", "A", "C", "G")
round(output, 3)

```

THE SPECTRAL ENVELOPE FOR REAL-VALUED TIME SERIES

The concept of the spectral envelope for categorical time series was extended to real-valued time series, $\{x_t; t = 0, \pm 1, \pm 2, \dots\}$, in McDougall et al. (1997). The process x_t can be vector-valued, but here we will concentrate on the univariate case. Further details can be found in McDougall et al. (1997). The concept is similar to projection pursuit (Friedman and Stuetzle, 1981). Let \mathcal{G} denote a k -dimensional vector space of continuous real-valued transformations with $\{g_1, \dots, g_k\}$ being a set of basis functions satisfying $E[g_i(x_t)^2] < \infty$, $i = 1, \dots, k$. Analogous to the categorical time series case, define the scaled time series with respect to the set \mathcal{G} to be the real-valued process

$$x_t(\beta) = \beta' y_t = \beta_1 g_1(x_t) + \dots + \beta_k g_k(x_t)$$

obtained from the vector process

$$y_t = (g_1(X_t), \dots, g_k(X_t))'$$

where $\beta = (\beta_1, \dots, \beta_k)' \in \mathbb{R}^k$. If the vector process, y_t , is assumed to have a continuous spectral density, say, $f_{yy}(\omega)$, then $x_t(\beta)$ will have a continuous spectral density $f_{xx}(\omega; \beta)$ for all $\beta \neq 0$. Noting, $f_{xx}(\omega; \beta) = \beta' f_{yy}(\omega) \beta = \beta' f_{yy}^{re}(\omega) \beta$, and $\sigma^2(\beta) = \text{var}[x_t(\beta)] = \beta' V \beta$, where $V = \text{var}(y_t)$ is assumed to be positive definite, the optimality criterion

$$\lambda(\omega) = \sup_{\beta \neq 0} \left\{ \frac{\beta' f_{yy}^{re}(\omega) \beta}{\beta' V \beta} \right\}, \quad (7.183)$$

is well defined and represents the largest proportion of the total power that can be attributed to the frequency ω for any particular scaled process $x_t(\beta)$. This interpretation of $\lambda(\omega)$ is consistent with the notion of the spectral envelope introduced in the previous section and provides the following working definition: *The spectral envelope of a time series with respect to the space \mathcal{G} is defined to be $\lambda(\omega)$.*

The solution to this problem, as in the categorical case, is attained by finding the largest scalar $\lambda(\omega)$ such that

$$f_{yy}^{re}(\omega) \beta(\omega) = \lambda(\omega) V \beta(\omega) \quad (7.184)$$

for $\beta(\omega) \neq 0$. That is, $\lambda(\omega)$ is the largest eigenvalue of $f_{yy}^{re}(\omega)$ in the metric of V , and the optimal scaling, $\beta(\omega)$, is the corresponding eigenvector.

If x_t is a categorical time series taking values in the finite state-space $\mathcal{S} = \{c_1, c_2, \dots, c_k\}$, where c_j represents a particular category, an appropriate choice for

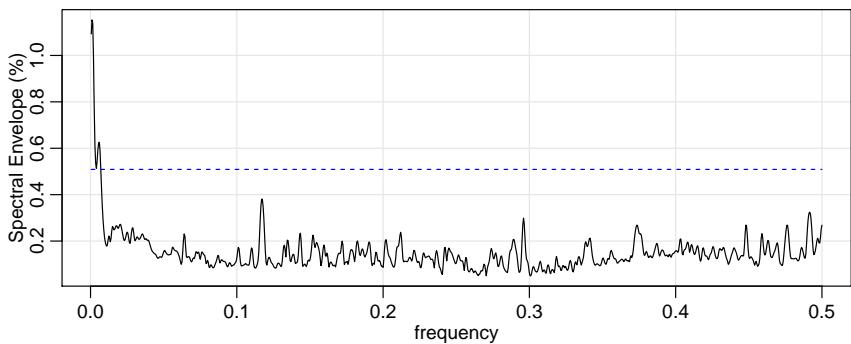


Fig. 7.25. Spectral envelope with respect to $\mathcal{G} = \{x, |x|, x^2\}$ for the NYSE returns.

\mathcal{G} is the set of indicator functions $g_j(x_t) = I(x_t = c_j)$. Hence, this is a natural generalization of the categorical case. In the categorical case, \mathcal{G} does not consist of linearly independent g 's, but it was easy to overcome this problem by reducing the dimension by one. In the vector-valued case, $x_t = (x_{1t}, \dots, x_{pt})'$, we consider \mathcal{G} to be the class of transformations from \mathbb{R}^p into \mathbb{R} such that the spectral density of $g(x_t)$ exists. One class of transformations of interest are linear combinations of x_t . In Tiao et al. (1993), for example, linear transformations of this type are used in a time domain approach to investigate contemporaneous relationships among the components of multivariate time series. Estimation and inference for the real-valued case are analogous to the methods described in the previous section for the categorical case. We consider an example here; numerous other examples can be found in McDougall et al. (1997).

Example 7.18 Optimal Transformations for Financial Data: NYSE Returns

In many financial applications, one typically addresses the analysis of the squared returns, such as was done in Section 5.3 and Section 6.11. However, there may be other transformations that supply more information than simply squaring the data. For example, Ding et al. (1993) who applied transformations of the form $|x_t|^d$, for $d \in (0, 3]$, to the S&P 500 stock market series. They found that power transformation of the absolute return has quite high autocorrelation for long lags, and this property is strongest when d is around 1. They concluded that the “result appears to argue against ARCH type specifications based upon squared returns.”

In this example, we examine the NYSE returns ([nyse](#)). We used with the generating set $\mathcal{G} = \{x, |x|, x^2\}$ —which seems natural for this analysis—to estimate the spectral envelope for the data, and the result is plotted in Figure 7.25. Although the data are white noise, they are clearly not iid, and considerable power is present at the low frequencies. The presence of spectral power at very low frequencies in detrended economic series has been frequently reported and is typically associated with long-range dependence. The estimated optimal transformation near the zero frequency, $\omega = .001$, was $\hat{\beta}(0.001) = (-1, 921, -2596)'$, which leads to the transformation

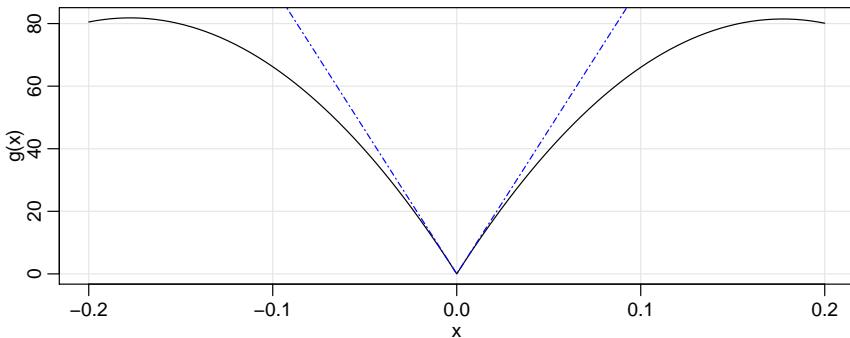


Fig. 7.26. Estimated optimal transformation, (7.185), for the NYSE returns at $\omega = .001$. The dashed line indicates the pure absolute value transformation.

$$g(x) = -x + 921|x| - 2596x^2. \quad (7.185)$$

This transformation is plotted in Figure 7.26. The transformation given in (7.185) is basically the absolute value (with some slight curvature and asymmetry) for most of the values, but the effect of extremes is damped.

The following R code was used in this example.

```

u      = astsa::nyse          # accept no substitutes
x      = cbind(u, abs(u), u^2)
Var   = var(x)                # var-cov matrix
xspec = mvspec(x, spans=c(5,3), taper=.5, plot=FALSE)
fxxr  = Re(xspec$ffx)         # fxxr is real(fxx)
# compute Q = Var^-1/2
ev    = eigen(Var)
Q     = ev$vectors%*%diag(1/sqrt(ev$values))%*%t(ev$vectors)
# compute spec env and scale vectors
num   = xspec$n.used          # sample size used for FFT
nfreq = length(xspec$freq)     # number of freqs used
specenv = matrix(0,nfreq,1)    # initialize the spec envelope
beta  = matrix(0,nfreq,3)      # initialize the scale vectors
for (k in 1:nfreq){
  ev = eigen(2*Q%*%fxxr[, , k] %*% Q/num) # get evalues of normalized spectral
  # matrix at freq k/n
  specenv[k] = ev$values[1]                  # spec env at freq k/n is max evalue
  b = Q%*%ev$vectors[, 1]                   # beta at freq k/n
  beta[k, ] = b/b[1]                         # first coef is always 1
# output and graphics
par(mar=c(2.5,2.75,.5,.5), mgp=c(1.5,.6,0))
frequency = xspec$freq
plot(frequency, 100*specenv, type="l", ylab="Spectral Envelope (%)")
m      = xspec$kernel$m
etainv = sqrt(sum(xspec$kernel[-m:m]^2))
thresh = 100*(2/num)*exp(qnorm(.9999)*etainv)*matrix(1,nfreq,1)
lines(frequency, thresh, lty=2, col=4)
# details
b = sign(b[2])*output[2,3:5]      # sign of |x| positive for beauty
output = cbind(frequency, specenv, beta)
colnames(output)=c("freq","specenv","x", "|x|", "x^2"); round(output, 4)

```

```

dev.new(); par(mar=c(2.5,2.5,.5,.5), mgp=c(1.5,.6,0))
# plot transform
g = function(x) { b[1]*x+b[2]*abs(x)+b[3]*x^2 }
curve(g, -.2, .2, panel.first=grid(lty=2))
g2 = function(x) { b[2]*abs(x) } # corresponding |x|
curve(g2, -.2,.2, add=TRUE, lty=6, col=4)

```

Problems

Section 7.2

7.1 Consider the complex Gaussian distribution for the random variable $X = X_c - iX_s$, as defined in (7.1)-(7.3), where the argument ω_k has been suppressed. Now, the $2p \times 1$ real random variable $Z = (X'_c, X'_s)'$ has a multivariate normal distribution with density

$$p(Z) = (2\pi)^{-P} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(Z - \mu)' \Sigma^{-1} (Z - \mu)\right\},$$

where $\mu = (M'_c, M'_s)'$ is the mean vector. Prove

$$|\Sigma| = \left(\frac{1}{2}\right)^{2p} |C - iQ|^2,$$

using the result that the eigenvectors and eigenvalues of Σ occur in pairs, i.e., $(v'_c, v'_s)'$ and $(v'_s, -v'_c)'$, where $v_c - iv_s$ denotes the eigenvector of f_{xx} . Show that

$$\frac{1}{2}(Z - \mu)' \Sigma^{-1} (Z - \mu)) = (X - M)^* f^{-1} (X - M)$$

so $p(X) = p(Z)$ and we can identify the density of the complex multivariate normal variable X with that of the real multivariate normal Z .

7.2 Prove \hat{f} in (7.6) maximizes the log likelihood (7.5) by minimizing the negative of the log likelihood

$$L \ln |f| + L \operatorname{tr}\{\hat{f}f^{-1}\}$$

in the form

$$L \sum_i (\lambda_i - \ln \lambda_i - 1) + Lp + L \ln |\hat{f}|,$$

where the λ_i values correspond to the eigenvalues in a simultaneous diagonalization of the matrices f and \hat{f} ; i.e., there exists a matrix P such that $P^* f P = I$ and $P^* \hat{f} P = \operatorname{diag}(\lambda_1, \dots, \lambda_p) = \Lambda$. Note, $\lambda_i - \ln \lambda_i - 1 \geq 0$ with equality if and only if $\lambda_i = 1$, implying $\Lambda = I$ maximizes the log likelihood and $f = \hat{f}$ is the maximizing value.

Section 7.3

7.3 Verify (7.18) and (7.19) for the mean-squared prediction error MSE in (7.11). Use the orthogonality principle, which implies

$$MSE = E \left[(y_t - \sum_{r=-\infty}^{\infty} \beta'_r x_{t-r}) y_t \right]$$

and gives a set of equations involving the autocovariance functions. Then, use the spectral representations and Fourier transform results to get the final result. Next, consider the predicted series

$$\hat{y}_t = \sum_{r=-\infty}^{\infty} \beta'_r x_{t-r},$$

where β_r satisfies (7.13). Show the ordinary coherence between y_t and \hat{y}_t is exactly the multiple coherence (7.20).

7.4 Consider the complex regression model (7.28) in the form

$$Y = XB + V,$$

where $Y = (Y_1, Y_2, \dots, Y_L)'$ denotes the observed DFTs after they have been re-indexed and $X = (X_1, X_2, \dots, X_L)'$ is a matrix containing the reindexed input vectors. The model is a complex regression model with $Y = Y_c - iY_s$, $X = X_c - iX_s$, $B = B_c - iB_s$, and $V = V_c - iV_s$ denoting the representation in terms of the usual cosine and sine transforms. Show the partitioned real regression model involving the $2L \times 1$ vector of cosine and sine transforms, say,

$$\begin{pmatrix} Y_c \\ Y_s \end{pmatrix} = \begin{pmatrix} X_c & -X_s \\ X_s & X_c \end{pmatrix} \begin{pmatrix} B_c \\ B_s \end{pmatrix} + \begin{pmatrix} V_c \\ V_s \end{pmatrix},$$

is *isomorphic* to the complex regression regression model in the sense that the real and imaginary parts of the complex model appear as components of the vectors in the real regression model. Use the usual regression theory to verify (7.27) holds. For example, writing the real regression model as

$$y = xb + v,$$

the isomorphism would imply

$$\begin{aligned} L(\hat{f}_{yy} - \hat{f}_{xy}^* \hat{f}_{xx}^{-1} \hat{f}_{xy}) &= Y^* Y - Y^* X (X^* X)^{-1} X^* Y \\ &= y'y - y'x(x'x)^{-1}x'y. \end{aligned}$$

Section 7.4

7.5 Consider estimating the function

$$\psi_t = \sum_{r=-\infty}^{\infty} a'_r \beta_{t-r}$$

by a linear filter estimator of the form

$$\hat{\psi}_t = \sum_{r=-\infty}^{\infty} a'_r \hat{\beta}_{t-r},$$

where $\hat{\beta}_t$ is defined by (7.42). Show a sufficient condition for $\hat{\psi}_t$ to be an unbiased estimator; i.e., $E \hat{\psi}_t = \psi_t$, is

$$H(\omega)Z(\omega) = I$$

for all ω . Similarly, show any other unbiased estimator satisfying the above condition has minimum variance (see Shumway and Dean, 1968), so the estimator given is a best linear unbiased (BLUE) estimator.

7.6 Consider a linear model with mean value function μ_t and a signal α_t delayed by an amount τ_j on each sensor, i.e.,

$$y_{jt} = \mu_t + \alpha_{t-\tau_j} + v_{jt}.$$

Show the estimators (7.42) for the mean and the signal are the Fourier transforms of

$$\hat{M}(\omega) = \frac{Y(\omega) - \overline{\phi(\omega)}B_w(\omega)}{1 - |\phi(\omega)|^2}$$

and

$$\hat{A}(\omega) = \frac{B_w(\omega) - \phi(\omega)Y(\omega)}{1 - |\phi(\omega)|^2},$$

where

$$\phi(\omega) = \frac{1}{N} \sum_{j=1}^N e^{2\pi i \omega \tau_j}$$

and $B_w(\omega)$ is defined in (7.64).

Section 7.5

7.7 Consider the estimator (7.67) as applied in the context of the random coefficient model (7.65). Prove the filter coefficients for the minimum mean square estimator can be determined from (7.68) and the mean square covariance is given by (7.71).

7.8 For the random coefficient model, verify the expected mean square of the regression power component is

$$\begin{aligned} E[SSR(\omega_k)] &= E[Y^*(\omega_k)Z(\omega_k)S_z^{-1}(\omega_k)Z^*(\omega_k)Y(\omega_k)] \\ &= Lf_\beta(\omega_k)\text{tr}\{S_z(\omega_k)\} + Lqf_v(\omega_k). \end{aligned}$$

Recall, the underlying frequency domain model is

$$Y(\omega_k) = Z(\omega_k)B(\omega_k) + V(\omega_k),$$

where $B(\omega_k)$ has spectrum $f_\beta(\omega_k)I_q$ and $V(\omega_k)$ has spectrum $f_v(\omega_k)I_N$ and the two processes are uncorrelated.

Section 7.6

7.9 Suppose we have $I = 2$ groups and the models

$$y_{1jt} = \mu_t + \alpha_{1t} + v_{1jt}$$

for the $j = 1, \dots, N$ observations in group 1 and

$$y_{2jt} = \mu_t + \alpha_{2t} + v_{2jt}$$

for the $j = 1, \dots, N$ observations in group 2, with $\alpha_{1t} + \alpha_{2t} = 0$. Suppose we want to test equality of the two group means; i.e.,

$$y_{ijt} = \mu_t + v_{ijt}, \quad i = 1, 2.$$

- (a) Derive the residual and error power components corresponding to (7.81) and (7.82) for this particular case.
- (b) Verify the forms of the linear compounds involving the mean given in (7.88) and (7.89), using (7.86) and (7.87).
- (c) Show the ratio of the two smoothed spectra in (7.101) has the indicated F -distribution when $f_1(\omega) = f_2(\omega)$. When the spectra are not equal, show the variable is proportional to an F -distribution, where the proportionality constant depends on the ratio of the spectra.

Section 7.7

7.10 The problem of detecting a signal in noise can be considered using the model

$$x_t = s_t + w_t, \quad t = 1, \dots, n,$$

for $p_1(x)$ when a signal is present and the model

$$x_t = w_t, \quad t = 1, \dots, n,$$

for $p_2(x)$ when no signal is present. Under multivariate normality, we might specialize even further by assuming the vector $w = (w_1, \dots, w_n)'$ has a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\Sigma = \sigma_w^2 I_n$, corresponding to white noise. Assuming the signal vector $s = (s_1, \dots, s_n)'$ is fixed and known, show the discriminant function (7.110) becomes the *matched filter*

$$\frac{1}{\sigma_w^2} \sum_{t=1}^n s_t x_t - \frac{1}{2} \left(\frac{S}{N} \right) + \ln \frac{\pi_1}{\pi_2},$$

where

$$\left(\frac{S}{N} \right) = \frac{\sum_{t=1}^n s_t^2}{\sigma_w^2}$$

denotes the *signal-to-noise ratio*. Give the decision criterion if the prior probabilities are assumed to be the same. Express the false alarm and missed signal probabilities in terms of the normal cdf and the signal-to-noise ratio.

7.11 Assume the same additive signal plus noise representations as in the previous problem, except, the signal is now a random process with a zero mean and covariance matrix $\sigma_s^2 I$. Derive the comparable version of (7.113) as a *quadratic detector*, and characterize its performance under both hypotheses in terms of constant multiples of the chi-squared distribution.

Section 7.8

7.12 Perform principal component analyses on the stimulus conditions (i) awake-heat and (ii) awake-shock, and compare your results to the results of Example 7.13. Use the data in `fmri` and average across subjects.

7.13 For this problem, consider the first three earthquake series (EQ1, EQ2, EQ3) listed in `eqexp`.

- (a) Estimate and compare the spectral density of the P component and then of the S component for each individual earthquake.
- (b) Estimate and compare the squared coherency between the P and S components of each individual earthquake. Comment on the strength of the coherence.
- (c) Let x_{ti} be the P component of earthquake $i = 1, 2, 3$, and let $x_t = (x_{t1}, x_{t2}, x_{t3})'$ be the 3×1 vector of P components. Estimate the spectral density, $\lambda_1(\omega)$, of the first principal component series of x_t . Compare this to the corresponding spectra calculated in (a).
- (d) Analogous to part (c), let y_t denote the 3×1 vector series of S components of the first three earthquakes. Repeat the analysis of part (c) on y_t .

7.14 In the factor analysis model (7.152), let $p = 3$, $q = 1$, and

$$\Sigma_{xx} = \begin{bmatrix} 1 & .4 & .9 \\ .4 & 1 & .7 \\ .9 & .7 & 1 \end{bmatrix}.$$

Show there is a unique choice for \mathcal{B} and D , but $\delta_3^2 < 0$, so the choice is not valid.

7.15 Extend the EM algorithm for classical factor analysis, (7.158)-(7.163), to the time series case of maximizing $\ln L(\mathcal{B}(\omega_j), D_{\epsilon\epsilon}(\omega_j))$ in (7.174). Then, for the data used in Example 7.15, find the approximate maximum likelihood estimates of $\mathcal{B}(\omega_j)$ and $D_{\epsilon\epsilon}(\omega_j)$, and, consequently, Λ_t .

Section 7.9

7.16 Verify, as stated in (7.179), the imaginary part of a $k \times k$ spectral matrix, $f^{im}(\omega)$, is skew symmetric, and then show $\beta' f_{yy}^{im}(\omega) \beta = 0$ for a real $k \times 1$ vector, β .

7.17 Repeat the analysis of Example 7.17 on BNRF1 of herpesvirus saimiri (the data file is [bnrf1hvs](#)), and compare the results with the results obtained for Epstein–Barr.

7.18 For the S&P500 weekly returns, say, r_t , analyzed in Example 6.17

- Estimate the spectrum of the r_t . Does the spectral estimate appear to support the hypothesis that the returns are white?
- Examine the possibility of spectral power near the zero frequency for a transformation of the returns, say, $g(r_t)$, using the spectral envelope with Example 7.18 as your guide. Compare the optimal transformation near or at the zero frequency with the usual transformation $y_t = r_t^2$.

Appendix A

Large Sample Theory

A.1 Convergence Modes

The study of the optimality properties of various estimators (such as the sample autocorrelation function) depends, in part, on being able to assess the large-sample behavior of these estimators. We summarize briefly here the kinds of convergence useful in this setting, namely, *mean square convergence*, *convergence in probability*, and *convergence in distribution*.

We consider first a particular class of random variables that plays an important role in the study of *second-order time series*, namely, the class of random variables belonging to the space L^2 , satisfying $E|x|^2 < \infty$. In proving certain properties of the class L^2 we will often use, for random variables $x, y \in L^2$, the *Cauchy–Schwarz inequality*,

$$|E(xy)|^2 \leq E(|x|^2)E(|y|^2), \quad (\text{A.1})$$

and the *Tchebycheff inequality*,

$$\Pr\{|x| \geq a\} \leq \frac{E(|x|^2)}{a^2}, \quad (\text{A.2})$$

for $a > 0$.

Next, we investigate the properties of *mean square convergence* of random variables in L^2 .

Definition A.1 A sequence of L^2 random variables $\{x_n\}$ is said to converge in **mean square** to a random variable $x \in L^2$, denoted by

$$x_n \xrightarrow{\text{ms}} x, \quad (\text{A.3})$$

if and only if

$$E|x_n - x|^2 \rightarrow 0 \quad (\text{A.4})$$

as $n \rightarrow \infty$.

Example A.1 Mean Square Convergence of the Sample Mean

Consider the white noise sequence w_t and the *signal plus noise* series

$$x_t = \mu + w_t.$$

Then, because

$$\mathbb{E}|\bar{x}_n - \mu|^2 = \frac{\sigma_w^2}{n} \rightarrow 0$$

as $n \rightarrow \infty$, where $\bar{x}_n = n^{-1} \sum_{t=1}^n x_t$ is the sample mean, we have $\bar{x}_n \xrightarrow{ms} \mu$.

We summarize some of the properties of mean square convergence as follows. If $x_n \xrightarrow{ms} x$, and $y_n \xrightarrow{ms} y$, then, as $n \rightarrow \infty$,

$$\mathbb{E}(x_n) \rightarrow \mathbb{E}(x); \quad (\text{A.5})$$

$$\mathbb{E}(|x_n|^2) \rightarrow \mathbb{E}(|x|^2); \quad (\text{A.6})$$

$$\mathbb{E}(x_n y_n) \rightarrow \mathbb{E}(xy). \quad (\text{A.7})$$

We also note the L^2 completeness theorem known as the *Riesz–Fischer Theorem*.

Theorem A.1 Let $\{x_n\}$ be a sequence in L^2 . Then, there exists a x in L^2 such that $x_n \xrightarrow{ms} x$ if and only if

$$\lim_{m \rightarrow \infty} \sup_{n \geq m} \mathbb{E}|x_n - x_m|^2 = 0. \quad (\text{A.8})$$

Often the condition of **Theorem A.1** is easier to verify to establish that a mean square limit x exists without knowing what it is. Sequences that satisfy (A.8) are said to be *Cauchy sequences* in L^2 and (A.8) is also known as the *Cauchy criterion* for L^2 .

Example A.2 Time Invariant Linear Filter

As an important example of the use of the Riesz–Fisher Theorem and the properties of mean square convergent series given in (A.5)–(A.7), a *time-invariant linear filter* is defined as a convolution of the form

$$y_t = \sum_{j=-\infty}^{\infty} a_j x_{t-j} \quad (\text{A.9})$$

for each $t = 0, \pm 1, \pm 2, \dots$, where x_t is a weakly stationary input series with mean μ_x and autocovariance function $\gamma_x(h)$, and a_j , for $j = 0, \pm 1, \pm 2, \dots$ are constants satisfying

$$\sum_{j=-\infty}^{\infty} |a_j| < \infty. \quad (\text{A.10})$$

The output series y_t defines a *filtering* or *smoothing* of the input series that changes the character of the time series in a predictable way. We need to know the conditions under which the outputs y_t in (A.9) and the linear process (1.31) exist.

Considering the sequence

$$y_t^n = \sum_{j=-n}^n a_j x_{t-j}, \quad (\text{A.11})$$

$n = 1, 2, \dots$, we need to show first that y_t^n has a mean square limit. By [Theorem A.1](#), it is enough to show that

$$\mathbb{E} |y_t^n - y_t^m|^2 \rightarrow 0$$

as $m, n \rightarrow \infty$. For $n > m > 0$,

$$\begin{aligned} \mathbb{E} |y_t^n - y_t^m|^2 &= \mathbb{E} \left| \sum_{m < |j| \leq n} a_j x_{t-j} \right|^2 \\ &= \sum_{m < |j| \leq n} \sum_{m \leq |k| \leq n} a_j a_k \mathbb{E}(x_{t-j} x_{t-k}) \\ &\leq \sum_{m < |j| \leq n} \sum_{m \leq |k| \leq n} |a_j| |a_k| |\mathbb{E}(x_{t-j} x_{t-k})| \\ &\leq \sum_{m < |j| \leq n} \sum_{m \leq |k| \leq n} |a_j| |a_k| (\mathbb{E}|x_{t-j}|^2)^{1/2} (\mathbb{E}|x_{t-k}|^2)^{1/2} \\ &= [\gamma_x(0) + \mu_x^2] \left(\sum_{m \leq |j| \leq n} |a_j| \right)^2 \rightarrow 0 \end{aligned}$$

as $m, n \rightarrow \infty$, because $\gamma_x(0)$ is a constant and $\{a_j\}$ is absolutely summable (the second inequality follows from the Cauchy–Schwarz inequality).

Although we know that the sequence $\{y_t^n\}$ given by (A.11) converges in mean square, we have not established its mean square limit. If S denotes the mean square limit of y_t^n , then using Fatou's Lemma, $\mathbb{E}|S - y_t|^2 = \mathbb{E} \liminf_{n \rightarrow \infty} |S - y_t^n|^2 \leq \liminf_{n \rightarrow \infty} \mathbb{E}|S - y_t^n|^2 = 0$, which establishes that y_t is the mean square limit of y_t^n .

Finally, we may use (A.5) and (A.7) to establish the mean, μ_y and autocovariance function, $\gamma_y(h)$ of y_t . In particular we have,

$$\mu_y = \mu_x \sum_{j=-\infty}^{\infty} a_j, \quad (\text{A.12})$$

and

$$\begin{aligned} \gamma_y(h) &= \mathbb{E} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a_j (x_{t+h-j} - \mu_x) a_k (x_{t-k} - \mu_x) \\ &= \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a_j \gamma_x(h-j+k) a_k. \end{aligned} \quad (\text{A.13})$$

A second important kind of convergence is *convergence in probability*.

Definition A.2 The sequence $\{x_n\}$, for $n = 1, 2, \dots$, converges in probability to a random variable x , denoted by

$$x_n \xrightarrow{P} x, \quad (\text{A.14})$$

if and only if

$$\Pr\{|x_n - x| > \epsilon\} \rightarrow 0 \quad (\text{A.15})$$

for all $\epsilon > 0$, as $n \rightarrow \infty$.

An immediate consequence of the Tchebycheff inequality, (A.2), is that

$$\Pr\{|x_n - x| \geq \epsilon\} \leq \frac{\text{E}(|x_n - x|^2)}{\epsilon^2},$$

so convergence in mean square implies convergence in probability, i.e.,

$$x_n \xrightarrow{ms} x \Rightarrow x_n \xrightarrow{P} x. \quad (\text{A.16})$$

This result implies, for example, that the filter (A.9) exists as a limit in probability because it converges in mean square [it is also easily established that (A.9) exists with probability one]. We mention, at this point, the useful *Weak Law of Large Numbers* which states that, for an independent identically distributed sequence x_n of random variables with mean μ , we have

$$\bar{x}_n \xrightarrow{P} \mu \quad (\text{A.17})$$

as $n \rightarrow \infty$, where $\bar{x}_n = n^{-1} \sum_{t=1}^n x_t$ is the usual sample mean.

We also will make use of the following concepts.

Definition A.3 For order in probability we write

$$x_n = o_p(a_n) \quad (\text{A.18})$$

if and only if

$$\frac{x_n}{a_n} \xrightarrow{P} 0. \quad (\text{A.19})$$

The term **boundedness in probability**, written $x_n = O_p(a_n)$, means that for every $\epsilon > 0$, there exists a $\delta(\epsilon) > 0$ such that

$$\Pr\left\{\left|\frac{x_n}{a_n}\right| > \delta(\epsilon)\right\} \leq \epsilon \quad (\text{A.20})$$

for all n .

Under this convention, e.g., the notation for $x_n \xrightarrow{P} x$ becomes $x_n - x = o_p(1)$. The definitions can be compared with their nonrandom counterparts, namely, for a fixed sequence $x_n = o(1)$ if $x_n \rightarrow 0$ and $x_n = O(1)$ if x_n , for $n = 1, 2, \dots$ is bounded. Some handy properties of $o_p(\cdot)$ and $O_p(\cdot)$ are as follows.

- (i) If $x_n = o_p(a_n)$ and $y_n = o_p(b_n)$, then $x_n y_n = o_p(a_n b_n)$ and $x_n + y_n = o_p(\max(a_n, b_n))$.

- (ii) If $x_n = o_p(a_n)$ and $y_n = O_p(b_n)$, then $x_n y_n = o_p(a_n b_n)$.
 (iii) Statement (i) is true if $O_p(\cdot)$ replaces $o_p(\cdot)$.

Example A.3 Convergence and Order in Probability for the Sample Mean

For the sample mean, \bar{x}_n , of iid random variables with mean μ and variance σ^2 , by the Tchebycheff inequality,

$$\begin{aligned}\Pr\{|\bar{x}_n - \mu| > \epsilon\} &\leq \frac{\mathbb{E}[(\bar{x}_n - \mu)^2]}{\epsilon^2} \\ &= \frac{\sigma^2}{n\epsilon^2} \rightarrow 0,\end{aligned}$$

as $n \rightarrow \infty$. It follows that $\bar{x}_n \xrightarrow{P} \mu$, or $\bar{x}_n - \mu = o_p(1)$. To find the rate, it follows that, for $\delta(\epsilon) > 0$,

$$\Pr\{\sqrt{n} |\bar{x}_n - \mu| > \delta(\epsilon)\} \leq \frac{\sigma^2/n}{\delta^2(\epsilon)/n} = \frac{\sigma^2}{\delta^2(\epsilon)}$$

by Tchebycheff's inequality, so taking $\epsilon = \sigma^2/\delta^2(\epsilon)$ shows that $\delta(\epsilon) = \sigma/\sqrt{\epsilon}$ does the job and

$$\bar{x}_n - \mu = O_p(n^{-1/2}).$$

For $k \times 1$ random vectors x_n , convergence in probability, written $x_n \xrightarrow{P} x$ or $x_n - x = o_p(1)$ is defined as element-by-element convergence in probability, or equivalently, as convergence in terms of the Euclidean distance

$$\|x_n - x\| \xrightarrow{P} 0, \quad (\text{A.21})$$

where $\|a\| = \sum_j a_j^2$ for any vector a . In this context, we note the result that if $x_n \xrightarrow{P} x$ and $g(x_n)$ is a continuous mapping,

$$g(x_n) \xrightarrow{P} g(x). \quad (\text{A.22})$$

Furthermore, if $x_n - a = O_p(\delta_n)$ with $\delta_n \rightarrow 0$ and $g(\cdot)$ is a function with continuous first derivatives continuous in a neighborhood of $a = (a_1, a_2, \dots, a_k)'$, we have the *Taylor series expansion in probability*

$$g(x_n) = g(a) + \left. \frac{\partial g(x)}{\partial x} \right|'_{x=a} (x_n - a) + O_p(\delta_n), \quad (\text{A.23})$$

where

$$\left. \frac{\partial g(x)}{\partial x} \right|_{x=a} = \left(\left. \frac{\partial g(x)}{\partial x_1} \right|_{x=a}, \dots, \left. \frac{\partial g(x)}{\partial x_k} \right|_{x=a} \right)'$$

denotes the vector of partial derivatives with respect to x_1, x_2, \dots, x_k , evaluated at a . This result remains true if $O_p(\delta_n)$ is replaced everywhere by $o_p(\delta_n)$.

Example A.4 Expansion for the Logarithm of the Sample Mean

With the same conditions as [Example A.3](#), consider $g(\bar{x}_n) = \log \bar{x}_n$, which has a derivative at μ , for $\mu > 0$. Then, because $\bar{x}_n - \mu = O_p(n^{-1/2})$ from [Example A.3](#), the conditions for the Taylor expansion in probability, ([A.23](#)), are satisfied and we have

$$\log \bar{x}_n = \log \mu + \mu^{-1}(\bar{x}_n - \mu) + O_p(n^{-1/2}).$$

The large sample distributions of sample mean and sample autocorrelation functions defined earlier can be developed using the notion of convergence in distribution.

Definition A.4 A sequence of $k \times 1$ random vectors $\{x_n\}$ is said to **converge in distribution**, written

$$x_n \xrightarrow{d} x \quad (\text{A.24})$$

if and only if

$$F_n(x) \rightarrow F(x) \quad (\text{A.25})$$

at the continuity points of distribution function $F(\cdot)$.

Example A.5 Convergence in Distribution

Consider a sequence $\{x_n\}$ of iid normal random variables with mean zero and variance $1/n$. Using the standard normal cdf, $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp\left\{-\frac{1}{2}u^2\right\} du$, we have $F_n(z) = \Phi(\sqrt{nz})$, so

$$F_n(z) \rightarrow \begin{cases} 0 & z < 0, \\ 1/2 & z = 0 \\ 1 & z > 0 \end{cases}$$

and we may take

$$F(z) = \begin{cases} 0 & z < 0, \\ 1 & z \geq 0, \end{cases}$$

because the point where the two functions differ is not a continuity point of $F(z)$.

The distribution function relates uniquely to the *characteristic function* through the Fourier transform, defined as a function with vector argument $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)'$, say

$$\phi(\lambda) = E(\exp\{i\lambda' x\}) = \int \exp\{i\lambda' x\} dF(x). \quad (\text{A.26})$$

Hence, for a sequence $\{x_n\}$ we may characterize convergence in distribution of $F_n(\cdot)$ in terms of convergence of the sequence of characteristic functions $\phi_n(\cdot)$, i.e.,

$$\phi_n(\lambda) \rightarrow \phi(\lambda) \Leftrightarrow F_n(x) \xrightarrow{d} F(x), \quad (\text{A.27})$$

where \Leftrightarrow means that the implication goes both directions. In this connection, we have

Proposition A.1 The Cramér–Wold device. Let $\{x_n\}$ be a sequence of $k \times 1$ random vectors. Then, for every $c = (c_1, c_2, \dots, c_k)' \in \mathbb{R}^k$

$$c' x_n \xrightarrow{d} c' x \Leftrightarrow x_n \xrightarrow{d} x. \quad (\text{A.28})$$

Proposition A.1 can be useful because sometimes it easier to show the convergence in distribution of $c' x_n$ than x_n directly.

Convergence in probability implies convergence in distribution, namely,

$$x_n \xrightarrow{P} x \Rightarrow x_n \xrightarrow{d} x, \quad (\text{A.29})$$

but the converse is only true when $x_n \xrightarrow{d} c$, where c is a constant vector. If $x_n \xrightarrow{d} x$ and $y_n \xrightarrow{d} c$ are two sequences of random vectors and c is a constant vector,

$$x_n + y_n \xrightarrow{d} x + c \quad \text{and} \quad y_n' x_n \xrightarrow{d} c' x. \quad (\text{A.30})$$

For a continuous mapping $h(x)$,

$$x_n \xrightarrow{d} x \Rightarrow h(x_n) \xrightarrow{d} h(x). \quad (\text{A.31})$$

A number of results in time series depend on making a series of approximations to prove convergence in distribution. For example, we have that if $x_n \xrightarrow{d} x$ can be approximated by the sequence y_n in the sense that

$$y_n - x_n = o_p(1), \quad (\text{A.32})$$

then we have that $y_n \xrightarrow{d} x$, so the approximating sequence y_n has the same limiting distribution as x . We present the following *Basic Approximation Theorem (BAT)* that will be used later to derive asymptotic distributions for the sample mean and ACF.

Theorem A.2 [Basic Approximation Theorem (BAT)] Let x_n for $n = 1, 2, \dots$, and y_{mn} for $m = 1, 2, \dots$, be random $k \times 1$ vectors such that

- (i) $y_{mn} \xrightarrow{d} y_m$ as $n \rightarrow \infty$ for each m ;
- (ii) $y_m \xrightarrow{d} y$ as $m \rightarrow \infty$;
- (iii) $\lim_{m \rightarrow \infty} \limsup_{n \rightarrow \infty} \Pr\{|x_n - y_{mn}| > \epsilon\} = 0$ for every $\epsilon > 0$.

Then, $x_n \xrightarrow{d} y$.

As a practical matter, the BAT condition (iii) is implied by the Tchebycheff inequality if

$$(iii') \quad E\{|x_n - y_{mn}|^2\} \rightarrow 0 \quad (\text{A.33})$$

as $m, n \rightarrow \infty$, and (iii') is often much easier to establish than (iii).

The theorem allows approximation of the underlying sequence in two steps, through the intermediary sequence y_{mn} , depending on two arguments. In the time series case, n is generally the sample length and m is generally the number of terms in an approximation to the linear process of the form (A.11).

Proof: The proof of the theorem is a simple exercise in using the characteristic functions and appealing to (A.27). We need to show

$$|\phi_{x_n} - \phi_y| \rightarrow 0,$$

where we use the shorthand notation $\phi \equiv \phi(\lambda)$ for ease. First,

$$|\phi_{x_n} - \phi_y| \leq |\phi_{x_n} - \phi_{y_{mn}}| + |\phi_{y_{mn}} - \phi_{y_m}| + |\phi_{y_m} - \phi_y|. \quad (\text{A.34})$$

By the condition (ii) and (A.27), the last term converges to zero, and by condition (i) and (A.27), the second term converges to zero and we only need consider the first term in (A.34). Now, write

$$\begin{aligned} |\phi_{x_n} - \phi_{y_{mn}}| &= \left| \mathbb{E}(e^{i\lambda' x_n} - e^{i\lambda' y_{mn}}) \right| \\ &\leq \mathbb{E} \left| e^{i\lambda' x_n} (1 - e^{i\lambda' (y_{mn} - x_n)}) \right| \\ &= \mathbb{E} \left| 1 - e^{i\lambda' (y_{mn} - x_n)} \right| \\ &= \mathbb{E} \left\{ \left| 1 - e^{i\lambda' (y_{mn} - x_n)} \right| I\{|y_{mn} - x_n| < \delta\} \right\} \\ &\quad + \mathbb{E} \left\{ \left| 1 - e^{i\lambda' (y_{mn} - x_n)} \right| I\{|y_{mn} - x_n| \geq \delta\} \right\}, \end{aligned}$$

where $\delta > 0$ and $I\{A\}$ denotes the indicator function of the set A . Then, given λ and $\epsilon > 0$, choose $\delta(\epsilon) > 0$ such that

$$\left| 1 - e^{i\lambda' (y_{mn} - x_n)} \right| < \epsilon$$

if $|y_{mn} - x_n| < \delta$, and the first term is less than ϵ , an arbitrarily small constant. For the second term, note that

$$\left| 1 - e^{i\lambda' (y_{mn} - x_n)} \right| \leq 2$$

and we have

$$\mathbb{E} \left\{ \left| 1 - e^{i\lambda' (y_{mn} - x_n)} \right| I\{|y_{mn} - x_n| \geq \delta\} \right\} \leq 2 \Pr\{|y_{mn} - x_n| \geq \delta\},$$

which converges to zero as $n \rightarrow \infty$ by property (iii). \square

A.2 Central Limit Theorems

We will generally be concerned with the large-sample properties of estimators that turn out to be normally distributed as $n \rightarrow \infty$.

Definition A.5 A sequence of random variables $\{x_n\}$ is said to be **asymptotically normal** with mean μ_n and variance σ_n^2 if, as $n \rightarrow \infty$,

$$\sigma_n^{-1}(x_n - \mu_n) \xrightarrow{d} z,$$

where z has the standard normal distribution. We shall abbreviate this as

$$x_n \sim \text{AN}(\mu_n, \sigma_n^2), \quad (\text{A.35})$$

where \sim will denote is distributed as.

We state the important *Central Limit Theorem*, as follows.

Theorem A.3 Let x_1, \dots, x_n be independent and identically distributed with mean μ and variance σ^2 . If $\bar{x}_n = (x_1 + \dots + x_n)/n$ denotes the sample mean, then

$$\bar{x}_n \sim \text{AN}(\mu, \sigma^2/n). \quad (\text{A.36})$$

Often, we will be concerned with a sequence of $k \times 1$ vectors $\{x_n\}$. The following property is motivated by the Cramér–Wold device, [Proposition A.1](#).

Proposition A.2 A sequence of random vectors is asymptotically normal, i.e.,

$$x_n \sim \text{AN}(\mu_n, \Sigma_n), \quad (\text{A.37})$$

if and only if

$$c' x_n \sim \text{AN}(c' \mu_n, c' \Sigma_n c) \quad (\text{A.38})$$

for all $c \in \mathbb{R}^k$ and Σ_n is positive definite.

In order to begin to consider what happens for dependent data in the limiting case, it is necessary to define, first of all, a particular kind of dependence known as M -dependence. We say that a time series x_t is M -dependent if the set of values $x_s, s \leq t$ is independent of the set of values $x_s, s \geq t + M + 1$, so time points separated by more than M units are independent. A central limit theorem for such dependent processes, used in conjunction with the Basic Approximation Theorem, will allow us to develop large-sample distributional results for the sample mean \bar{x} and the sample ACF $\hat{\rho}_x(h)$ in the stationary case.

In the arguments that follow, we often make use of the formula for the variance of \bar{x}_n in the stationary case, namely,

$$\text{var } \bar{x}_n = n^{-1} \sum_{u=-(n-1)}^{(n-1)} \left(1 - \frac{|u|}{n}\right) \gamma(u), \quad (\text{A.39})$$

which was established in (1.35) on page 27. We shall also use the fact that, for

$$\sum_{u=-\infty}^{\infty} |\gamma(u)| < \infty,$$

we would have, by dominated convergence,^{A.1}

$$n \operatorname{var} \bar{x}_n \rightarrow \sum_{u=-\infty}^{\infty} \gamma(u), \quad (\text{A.40})$$

because $|(1 - |u|/n)\gamma(u)| \leq |\gamma(u)|$ and $(1 - |u|/n)\gamma(u) \rightarrow \gamma(u)$. We may now state the *M-Dependent Central Limit Theorem* as follows.

Theorem A.4 *If x_t is a strictly stationary M-dependent sequence of random variables with mean zero and autocovariance function $\gamma(\cdot)$ and if*

$$V_M = \sum_{u=-M}^M \gamma(u), \quad (\text{A.41})$$

where $V_M \neq 0$,

$$\bar{x}_n \sim \text{AN}(0, V_M/n). \quad (\text{A.42})$$

Proof: To prove the theorem, using [Theorem A.2](#), the Basic Approximation Theorem, we may construct a sequence of variables y_{mn} approximating

$$n^{1/2} \bar{x}_n = n^{-1/2} \sum_{t=1}^n x_t$$

in the dependent case and then simply verify conditions (i), (ii), and (iii) of [Theorem A.2](#). For $m > 2M$, we may first consider the approximation

$$\begin{aligned} y_{mn} &= n^{-1/2} [(x_1 + \cdots + x_{m-M}) + (x_{m+1} + \cdots + x_{2m-M}) \\ &\quad + (x_{2m+1} + \cdots + x_{3m-M}) + \cdots + (x_{(r-1)m+1} + \cdots + x_{rm-M})] \\ &= n^{-1/2} (z_1 + z_2 + \cdots + z_r), \end{aligned}$$

where $r = [n/m]$, with $[n/m]$ denoting the greatest integer less than or equal to n/m . This approximation contains only part of $n^{1/2} \bar{x}_n$, but the random variables z_1, z_2, \dots, z_r are independent because they are separated by more than M time points, e.g., $m+1-(m-M) = M+1$ points separate z_1 and z_2 . Because of strict stationarity, z_1, z_2, \dots, z_r are identically distributed with zero means and variances

$$S_{m-M} = \sum_{|u| \leq M} (m - M - |u|) \gamma(u)$$

by a computation similar to that producing [\(A.39\)](#). We now verify the conditions of the Basic Approximation Theorem hold.

^{A.1} Dominated convergence technically relates to convergent sequences (with respect to a sigma-additive measure μ) of measurable functions $f_n \rightarrow f$ bounded by an integrable function g , $\int g d\mu < \infty$. For such a sequence,

$$\int f_n d\mu \rightarrow \int f d\mu.$$

For the case in point, take $f_n(u) = (1 - |u|/n)\gamma(u)$ for $|u| < n$ and as zero for $|u| \geq n$. Take $\mu(u) = 1, u = \pm 1, \pm 2, \dots$ to be counting measure.

(i) Applying the Central Limit Theorem to the sum y_{mn} gives

$$y_{mn} = n^{-1/2} \sum_{i=1}^r z_i = (n/r)^{-1/2} r^{-1/2} \sum_{i=1}^r z_i.$$

Because $(n/r)^{-1/2} \rightarrow m^{1/2}$ and

$$r^{-1/2} \sum_{i=1}^r z_i \xrightarrow{d} N(0, S_{m-M}),$$

it follows from (A.30) that

$$y_{mn} \xrightarrow{d} y_m \sim N(0, S_{m-M}/m).$$

as $n \rightarrow \infty$, for a fixed m .

(ii) Note that as $m \rightarrow \infty$, $S_{m-M}/m \rightarrow V_M$ using dominated convergence, where V_M is defined in (A.41). Hence, the characteristic function of y_m , say,

$$\phi_m(\lambda) = \exp\left\{-\frac{1}{2}\lambda^2 \frac{S_{m-M}}{m}\right\} \rightarrow \exp\left\{-\frac{1}{2}\lambda^2 V_M\right\},$$

as $m \rightarrow \infty$, which is the characteristic function of a random variable $y \sim N(0, V_M)$ and the result follows because of (A.27).

(iii) To verify the last condition of the BAT theorem,

$$\begin{aligned} n^{1/2} \bar{x}_n - y_{mn} &= n^{-1/2} [(x_{m-M+1} + \cdots + x_m) \\ &\quad + (x_{2m-M+1} + \cdots + x_{2m}) \\ &\quad + (x_{(r-1)m-M+1} + \cdots + x_{(r-1)m}) \\ &\quad \vdots \\ &\quad + (x_{rm-M+1} + \cdots + x_n)] \\ &= n^{-1/2} (w_1 + w_2 + \cdots + w_r), \end{aligned}$$

so the error is expressed as a scaled sum of iid variables with variance S_M for the first $r - 1$ variables and

$$\begin{aligned} \text{var}(w_r) &= \sum_{|u| \leq m-M} \left(n - [n/m]m + M - |u| \right) \gamma(u) \\ &\leq \sum_{|u| \leq m-M} (m + M - |u|) \gamma(u). \end{aligned}$$

Hence,

$$\text{var}[n^{1/2} \bar{x} - y_{mn}] = n^{-1} [(r-1)S_M + \text{var } w_r],$$

which converges to $m^{-1} S_M$ as $n \rightarrow \infty$. Because $m^{-1} S_M \rightarrow 0$ as $m \rightarrow \infty$, the condition of (iii) holds by the Tchebycheff inequality. □

A.3 The Mean and Autocorrelation Functions

The background material in the previous two sections can be used to develop the asymptotic properties of the sample mean and ACF used to evaluate statistical significance. In particular, we are interested in verifying [Property 1.2](#).

We begin with the distribution of the sample mean \bar{x}_n , noting that [\(A.40\)](#) suggests a form for the limiting variance. In all of the asymptotics, we will use the assumption that x_t is a linear process, as defined in [Definition 1.12](#), but with the added condition that $\{w_t\}$ is iid. That is, throughout this section we assume

$$x_t = \mu_x + \sum_{j=-\infty}^{\infty} \psi_j w_{t-j} \quad (\text{A.43})$$

where $w_t \sim \text{iid}(0, \sigma_w^2)$, and the coefficients satisfy

$$\sum_{j=-\infty}^{\infty} |\psi_j| < \infty. \quad (\text{A.44})$$

Before proceeding further, we should note that the exact sampling distribution of \bar{x}_n is available if the distribution of the underlying vector $x = (x_1, x_2, \dots, x_n)'$ is multivariate normal. Then, \bar{x}_n is just a linear combination of jointly normal variables that will have the normal distribution

$$\bar{x}_n \sim N\left(\mu_x, n^{-1} \sum_{|u|< n} \left(1 - \frac{|u|}{n}\right) \gamma_x(u)\right), \quad (\text{A.45})$$

by [\(A.39\)](#). In the case where x_t are not jointly normally distributed, we have the following theorem.

Theorem A.5 *If x_t is a linear process of the form [\(A.43\)](#) and $\sum_j \psi_j \neq 0$, then*

$$\bar{x}_n \sim \text{AN}(\mu_x, n^{-1} V), \quad (\text{A.46})$$

where

$$V = \sum_{h=-\infty}^{\infty} \gamma_x(h) = \sigma_w^2 \left(\sum_{j=-\infty}^{\infty} \psi_j \right)^2 \quad (\text{A.47})$$

and $\gamma_x(\cdot)$ is the autocovariance function of x_t .

Proof: To prove the above, we can again use the Basic Approximation Theorem, [Theorem A.2](#), by first defining the strictly stationary $2m$ -dependent linear process with finite limits

$$x_t^m = \sum_{j=-m}^m \psi_j w_{t-j}$$

as an approximation to x_t to use in the approximating mean

$$\bar{x}_{n,m} = n^{-1} \sum_{t=1}^n x_t^m.$$

Then, take

$$y_{mn} = n^{1/2}(\bar{x}_{n,m} - \mu_x)$$

as an approximation to $n^{1/2}(\bar{x}_n - \mu_x)$.

(i) Applying [Theorem A.4](#), we have

$$y_{mn} \xrightarrow{d} y_m \sim N(0, V_m),$$

as $n \rightarrow \infty$, where

$$V_m = \sum_{h=-2m}^{2m} \gamma_x(h) = \sigma_w^2 \left(\sum_{j=-m}^m \psi_j \right)^2.$$

To verify the above, we note that for the general linear process with infinite limits, [\(1.32\)](#) implies that

$$\sum_{h=-\infty}^{\infty} \gamma_x(h) = \sigma_w^2 \sum_{h=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \psi_{j+h} \psi_j = \sigma_w^2 \left(\sum_{j=-\infty}^{\infty} \psi_j \right)^2,$$

so taking the special case $\psi_j = 0$, for $|j| > m$, we obtain V_m .

(ii) Because $V_m \rightarrow V$ in [\(A.47\)](#) as $m \rightarrow \infty$, we may use the same characteristic function argument as under (ii) in the proof of [Theorem A.4](#) to note that

$$y_m \xrightarrow{d} y \sim N(0, V),$$

where V is given by [\(A.47\)](#).

(iii) Finally,

$$\begin{aligned} \text{var} \left[n^{1/2}(\bar{x}_n - \mu_x) - y_{mn} \right] &= n \text{var} \left[n^{-1} \sum_{t=1}^n \sum_{|j|>m} \psi_j w_{t-j} \right] \\ &= \sigma_w^2 \left(\sum_{|j|>m} \psi_j \right)^2 \rightarrow 0 \end{aligned}$$

as $m \rightarrow \infty$. □

In order to develop the sampling distribution of the sample autocovariance function, $\hat{\gamma}_x(h)$, and the sample autocorrelation function, $\hat{\rho}_x(h)$, we need to develop some idea as to the mean and variance of $\hat{\gamma}_x(h)$ under some reasonable assumptions. These computations for $\hat{\gamma}_x(h)$ are messy, and we consider a comparable quantity

$$\tilde{\gamma}_x(h) = n^{-1} \sum_{t=1}^n (x_{t+h} - \mu_x)(x_t - \mu_x) \quad (\text{A.48})$$

as an approximation. By [Problem 1.30](#),

$$n^{1/2}[\tilde{\gamma}_x(h) - \hat{\gamma}_x(h)] = o_p(1),$$

so that limiting distributional results proved for $n^{1/2}\tilde{\gamma}_x(h)$ will hold for $n^{1/2}\hat{\gamma}_x(h)$ by [\(A.32\)](#).

We begin by proving formulas for the variance and for the limiting variance of $\tilde{\gamma}_x(h)$ under the assumptions that x_t is a linear process of the form [\(A.43\)](#), satisfying [\(A.44\)](#) with the white noise variates w_t having variance σ_w^2 as before, but also required to have fourth moments satisfying

$$\mathbb{E}(w_t^4) = \eta\sigma_w^4 < \infty, \quad (\text{A.49})$$

where η is some constant. We seek results comparable with [\(A.39\)](#) and [\(A.40\)](#) for $\tilde{\gamma}_x(h)$. To ease the notation, we will henceforth drop the subscript x from the notation.

Using [\(A.48\)](#), $\mathbb{E}[\tilde{\gamma}(h)] = \gamma(h)$. Under the above assumptions, we show now that, for $p, q = 0, 1, 2, \dots$,

$$\text{cov} [\tilde{\gamma}(p), \tilde{\gamma}(q)] = n^{-1} \sum_{u=-n+1}^{n-1} \left(1 - \frac{|u|}{n}\right) V_u, \quad (\text{A.50})$$

where

$$\begin{aligned} V_u &= \gamma(u)\gamma(u+p-q) + \gamma(u+p)\gamma(u-q) \\ &\quad + (\eta-3)\sigma_w^4 \sum_i \psi_{i+u+q}\psi_{i+u}\psi_{i+p}\psi_i. \end{aligned} \quad (\text{A.51})$$

The absolute summability of the ψ_j can then be shown to imply the absolute summability of the V_u .^{A.2} Thus, the dominated convergence theorem implies

$$\begin{aligned} n \text{cov} [\tilde{\gamma}(p), \tilde{\gamma}(q)] &\rightarrow \sum_{u=-\infty}^{\infty} V_u \\ &= (\eta-3)\gamma(p)\gamma(q) \\ &\quad + \sum_{u=-\infty}^{\infty} \left[\gamma(u)\gamma(u+p-q) + \gamma(u+p)\gamma(u-q) \right]. \end{aligned} \quad (\text{A.52})$$

To verify [\(A.50\)](#) is somewhat tedious, so we only go partially through the calculations, leaving the repetitive details to the reader. First, rewrite [\(A.43\)](#) as

$$x_t = \mu + \sum_{i=-\infty}^{\infty} \psi_{t-i} w_i,$$

^{A.2} Note: $\sum_{j=-\infty}^{\infty} |a_j| < \infty$ and $\sum_{j=-\infty}^{\infty} |b_j| < \infty$ implies $\sum_{j=-\infty}^{\infty} |a_j b_j| < \infty$.

so that

$$\text{E}[\tilde{\gamma}(p)\tilde{\gamma}(q)] = n^{-2} \sum_{s,t} \sum_{i,j,k,\ell} \psi_{s+p-i}\psi_{s-j}\psi_{t+q-k}\psi_{t-\ell} \text{E}(w_i w_j w_k w_\ell).$$

Then, evaluate, using the easily verified properties of the w_t series

$$\text{E}(w_i w_j w_k w_\ell) = \begin{cases} \eta \sigma_w^4 & \text{if } i = j = k = \ell \\ \sigma_w^4 & \text{if } i = j \neq k = \ell \\ 0 & \text{if } i \neq j, i \neq k \text{ and } i \neq \ell. \end{cases}$$

To apply the rules, we break the sum over the subscripts i, j, k, ℓ into four terms, namely,

$$\sum_{i,j,k,\ell} = \sum_{i=j=k=\ell} + \sum_{i=j \neq k=\ell} + \sum_{i=k \neq j=\ell} + \sum_{i=\ell \neq j=k} = S_1 + S_2 + S_3 + S_4.$$

Now,

$$S_1 = \eta \sigma_w^4 \sum_i \psi_{s+p-i}\psi_{s-i}\psi_{t+q-i}\psi_{t-i} = \eta \sigma_w^4 \sum_i \psi_{i+s-t+p}\psi_{i+s-t}\psi_{i+q}\psi_i,$$

where we have let $i' = t - i$ to get the final form. For the second term,

$$\begin{aligned} S_2 &= \sum_{i=j \neq k=\ell} \psi_{s+p-i}\psi_{s-j}\psi_{t+q-k}\psi_{t-\ell} \text{E}(w_i w_j w_k w_\ell) \\ &= \sum_{i \neq k} \psi_{s+p-i}\psi_{s-i}\psi_{t+q-k}\psi_{t-k} \text{E}(w_i^2) \text{E}(w_k^2). \end{aligned}$$

Then, using the fact that

$$\sum_{i \neq k} = \sum_{i,k} - \sum_{i=k},$$

we have

$$\begin{aligned} S_2 &= \sigma_w^4 \sum_{i,k} \psi_{s+p-i}\psi_{s-i}\psi_{t+q-k}\psi_{t-k} - \sigma_w^4 \sum_i \psi_{s+p-i}\psi_{s-i}\psi_{t+q-i}\psi_{t-i} \\ &= \gamma(p)\gamma(q) - \sigma_w^4 \sum_i \psi_{i+s-t+p}\psi_{i+s-t}\psi_{i+q}\psi_i, \end{aligned}$$

letting $i' = s - i$, $k' = t - k$ in the first term and $i' = s - i$ in the second term. Repeating the argument for S_3 and S_4 and substituting into the covariance expression yields

$$\begin{aligned} \text{E}[\tilde{\gamma}(p)\tilde{\gamma}(q)] &= n^{-2} \sum_{s,t} \left[\gamma(p)\gamma(q) + \gamma(s-t)\gamma(s-t+p-q) \right. \\ &\quad + \gamma(s-t+p)\gamma(s-t-q) \\ &\quad \left. + (\eta-3)\sigma_w^4 \sum_i \psi_{i+s-t+p}\psi_{i+s-t}\psi_{i+q}\psi_i \right]. \end{aligned}$$

Then, letting $u = s - t$ and subtracting $E[\tilde{\gamma}(p)]E[\tilde{\gamma}(q)] = \gamma(p)\gamma(q)$ from the summation leads to the result (A.51). Summing (A.51) over u and applying dominated convergence leads to (A.52).

The above results for the variances and covariances of the approximating statistics $\tilde{\gamma}(\cdot)$ enable proving the following central limit theorem for the autocovariance functions $\hat{\gamma}(\cdot)$.

Theorem A.6 *If x_t is a stationary linear process of the form (A.43) satisfying the fourth moment condition (A.49), then, for fixed K ,*

$$\begin{pmatrix} \hat{\gamma}(0) \\ \hat{\gamma}(1) \\ \vdots \\ \hat{\gamma}(K) \end{pmatrix} \sim \text{AN} \left(\begin{pmatrix} \gamma(0) \\ \gamma(1) \\ \vdots \\ \gamma(K) \end{pmatrix}, n^{-1}V \right),$$

where V is the matrix with elements given by

$$\begin{aligned} v_{pq} &= (\eta - 3)\gamma(p)\gamma(q) \\ &+ \sum_{u=-\infty}^{\infty} \left[\gamma(u)\gamma(u - p + q) + \gamma(u + q)\gamma(u - p) \right]. \end{aligned} \quad (\text{A.53})$$

Proof: It suffices to show the result for the approximate autocovariance (A.48) for $\tilde{\gamma}(\cdot)$ by the remark given below it (see also Problem 1.30). First, define the strictly stationary $(2m + K)$ -dependent $(K + 1) \times 1$ vector

$$y_t^m = \begin{pmatrix} (x_t^m - \mu)^2 \\ (x_{t+1}^m - \mu)(x_t^m - \mu) \\ \vdots \\ (x_{t+K}^m - \mu)(x_t^m - \mu) \end{pmatrix},$$

where

$$x_t^m = \mu + \sum_{j=-m}^m \psi_j w_{t-j}$$

is the usual approximation. The sample mean of the above vector is

$$\bar{y}_{mn} = n^{-1} \sum_{t=1}^n y_t^m = \begin{pmatrix} \bar{\gamma}^{mn}(0) \\ \bar{\gamma}^{mn}(1) \\ \vdots \\ \bar{\gamma}^{mn}(K) \end{pmatrix},$$

where

$$\bar{\gamma}^{mn}(h) = n^{-1} \sum_{t=1}^n (x_{t+h}^m - \mu)(x_t^m - \mu)$$

denotes the sample autocovariance of the approximating series. Also,

$$\mathbb{E}y_t^m = \begin{pmatrix} \gamma^m(0) \\ \gamma^m(1) \\ \vdots \\ \gamma^m(K) \end{pmatrix},$$

where $\gamma^m(h)$ is the theoretical covariance function of the series x_t^m . Then, consider the vector

$$y_{mn} = n^{1/2}[\bar{y}_{mn} - \mathbb{E}(\bar{y}_{mn})]$$

as an approximation to

$$y_n = n^{1/2} \left[\begin{pmatrix} \tilde{\gamma}(0) \\ \tilde{\gamma}(1) \\ \vdots \\ \tilde{\gamma}(K) \end{pmatrix} - \begin{pmatrix} \gamma(0) \\ \gamma(1) \\ \vdots \\ \gamma(K) \end{pmatrix} \right],$$

where $\mathbb{E}(\bar{y}_{mn})$ is the same as $\mathbb{E}(y_t^m)$ given above. The elements of the vector approximation y_{mn} are clearly $n^{1/2}(\tilde{\gamma}^{mn}(h) - \gamma^m(h))$. Note that the elements of y_n are based on the linear process x_t , whereas the elements of y_{mn} are based on the m -dependent linear process x_t^m . To obtain a limiting distribution for y_n , we apply the Basic Approximation Theorem, [Theorem A.2](#), using y_{mn} as our approximation. We now verify (i), (ii), and (iii) of [Theorem A.2](#).

- (i) First, let c be a $(K+1) \times 1$ vector of constants, and apply the central limit theorem to the $(2m+K)$ -dependent series $c'y_{mn}$ using the Cramér–Wold device [\(A.28\)](#). We obtain

$$c'y_{mn} = n^{1/2} c' [\bar{y}_{mn} - \mathbb{E}(\bar{y}_{mn})] \xrightarrow{d} c'y_m \sim N(0, c'V_m c),$$

as $n \rightarrow \infty$, where V_m is a matrix containing the finite analogs of the elements v_{pq} defined in [\(A.53\)](#).

- (ii) Note that, since $V_m \rightarrow V$ as $m \rightarrow \infty$, it follows that

$$c'y_m \xrightarrow{d} c'y \sim N(0, c'Vc),$$

so, by the Cramér–Wold device, the limiting $(K+1) \times 1$ multivariate normal variable is $N(0, V)$.

- (iii) For this condition, we can focus on the element-by-element components of

$$\Pr\{|y_n - y_{mn}| > \epsilon\}.$$

For example, using the Tchebycheff inequality, the h -th element of the probability statement can be bounded by

$$\begin{aligned} n\epsilon^{-2}\text{var}(\tilde{\gamma}(h) - \gamma^m(h)) \\ = \epsilon^{-2} \{n \text{var} \tilde{\gamma}(h) + n \text{var} \gamma^m(h) - 2n \text{cov}[\tilde{\gamma}(h), \gamma^m(h)]\}. \end{aligned}$$

Using the results that led to (A.52), we see that the preceding expression approaches

$$(v_{hh} + v_{hh} - 2v_{hh})/\epsilon^2 = 0,$$

as $m, n \rightarrow \infty$.

□

To obtain a result comparable to [Theorem A.6](#) for the autocorrelation function ACF, we note the following theorem.

Theorem A.7 *If x_t is a stationary linear process of the form (1.31) satisfying the fourth moment condition (A.49), then for fixed K ,*

$$\begin{pmatrix} \widehat{\rho}(1) \\ \vdots \\ \widehat{\rho}(K) \end{pmatrix} \sim \text{AN} \left[\begin{pmatrix} \rho(1) \\ \vdots \\ \rho(K) \end{pmatrix}, n^{-1}W \right],$$

where W is the matrix with elements given by

$$\begin{aligned} w_{pq} &= \sum_{u=-\infty}^{\infty} \left[\rho(u+p)\rho(u+q) + \rho(u-p)\rho(u+q) + 2\rho(p)\rho(q)\rho^2(u) \right. \\ &\quad \left. - 2\rho(p)\rho(u)\rho(u+q) - 2\rho(q)\rho(u)\rho(u+p) \right] \\ &= \sum_{u=1}^{\infty} [\rho(u+p) + \rho(u-p) - 2\rho(p)\rho(u)] \\ &\quad \times [\rho(u+q) + \rho(u-q) - 2\rho(q)\rho(u)], \end{aligned} \tag{A.54}$$

where the last form is more convenient.

Proof: To prove the theorem, we use the delta method^{A.3} for the limiting distribution of a function of the form

$$g(x_0, x_1, \dots, x_K) = (x_1/x_0, \dots, x_K/x_0)',$$

where $x_h = \widehat{\gamma}(h)$, for $h = 0, 1, \dots, K$. Hence, using the delta method and [Theorem A.6](#),

$$g(\widehat{\gamma}(0), \widehat{\gamma}(1), \dots, \widehat{\gamma}(K)) = (\widehat{\rho}(1), \dots, \widehat{\rho}(K))'$$

is asymptotically normal with mean vector $(\rho(1), \dots, \rho(K))'$ and covariance matrix

$$n^{-1}W = n^{-1}DVD',$$

where V is defined by (A.53) and D is the $(K+1) \times K$ matrix of partial derivatives

^{A.3} The *delta method* states that if a k -dimensional vector sequence $x_n \sim \text{AN}(\mu, a_n^2 \Sigma)$, with $a_n \rightarrow 0$, and $g(x)$ is an $r \times 1$ continuously differentiable vector function of x , then $g(x_n) \sim \text{AN}(g(\mu), a_n^2 D \Sigma D')$ where D is the $r \times k$ matrix with elements $d_{ij} = \frac{\partial g_i(x)}{\partial x_j}|_{\mu}$.

$$D = \frac{1}{x_0^2} \begin{pmatrix} -x_1 & x_0 & 0 & \dots & 0 \\ -x_2 & 0 & x_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -x_K & 0 & 0 & \dots & x_0 \end{pmatrix}$$

Substituting $\gamma(h)$ for x_h , we note that D can be written as the patterned matrix

$$D = \frac{1}{\gamma(0)} (-\rho I_K),$$

where $\rho = (\rho(1), \rho(2), \dots, \rho(K))'$ is the $K \times 1$ matrix of autocorrelations and I_K is the $K \times K$ identity matrix. Then, it follows from writing the matrix V in the partitioned form

$$V = \begin{pmatrix} v_{00} & v_1' \\ v_1 & V_{22} \end{pmatrix}$$

that

$$W = \gamma^{-2}(0) [v_{00}\rho\rho' - \rho v_1' - v_1\rho' + V_{22}],$$

where $v_1 = (v_{10}, v_{20}, \dots, v_{K0})'$ and $V_{22} = \{v_{pq}; p, q = 1, \dots, K\}$. Hence,

$$\begin{aligned} w_{pq} &= \gamma^{-2}(0) [v_{pq} - \rho(p)v_{0q} - \rho(q)v_{p0} + \rho(p)\rho(q)v_{00}] \\ &= \sum_{u=-\infty}^{\infty} \left[\rho(u)\rho(u-p+q) + \rho(u-p)\rho(u+q) + 2\rho(p)\rho(q)\rho^2(u) \right. \\ &\quad \left. - 2\rho(p)\rho(u)\rho(u+q) - 2\rho(q)\rho(u)\rho(u-p) \right]. \end{aligned}$$

Interchanging the summations, we get the w_{pq} specified in the statement of the theorem, finishing the proof. \square

Specializing the theorem to the case of interest in this chapter, we note that if $\{x_t\}$ is iid with finite fourth moment, then $w_{pq} = 1$ for $p = q$ and is zero otherwise. In this case, for $h = 1, \dots, K$, the $\widehat{\rho}(h)$ are asymptotically independent and jointly normal with

$$\widehat{\rho}(h) \sim \text{AN}(0, n^{-1}). \quad (\text{A.55})$$

This justifies the use of (1.38) and the discussion below it as a method for testing whether a series is white noise.

For the cross-correlation, it has been noted that the same kind of approximation holds and we quote the following theorem for the bivariate case, which can be proved using similar arguments (see Brockwell and Davis, 1991, p. 410).

Theorem A.8 If

$$x_t = \sum_{j=-\infty}^{\infty} \alpha_j w_{t-j,1}$$

and

$$y_t = \sum_{j=-\infty}^{\infty} \beta_j w_{t-j,2}$$

are two linear processes with absolutely summable coefficients and the two white noise sequences are iid and independent of each other with variances σ_1^2 and σ_2^2 , then for $h \geq 0$,

$$\hat{\rho}_{xy}(h) \sim \text{AN}\left(\rho_{xy}(h), n^{-1} \sum_j \rho_x(j)\rho_y(j)\right) \quad (\text{A.56})$$

and the joint distribution of $(\hat{\rho}_{xy}(h), \hat{\rho}_{xy}(k))'$ is asymptotically normal with mean vector zero and

$$\text{cov}(\hat{\rho}_{xy}(h), \hat{\rho}_{xy}(k)) = n^{-1} \sum_j \rho_x(j)\rho_y(j+k-h). \quad (\text{A.57})$$

Again, specializing to the case of interest in this chapter, as long as at least one of the two series is white (iid) noise, we obtain

$$\hat{\rho}_{xy}(h) \sim \text{AN}\left(0, n^{-1}\right), \quad (\text{A.58})$$

which justifies [Property 1.3](#).

Appendix B

Time Domain Theory

B.1 Hilbert Spaces and the Projection Theorem

Most of the material on mean square estimation and regression can be embedded in a more general setting involving an inner product space that is also complete (that is, satisfies the Cauchy condition). Two examples of inner products are $E(xy^*)$, where the elements are random variables, and $\sum x_i y_i^*$, where the elements are sequences. These examples include the possibility of complex elements, in which case, $*$ denotes the conjugation. We denote an inner product, in general, by the notation $\langle x, y \rangle$. Now, define an *inner product space* by its properties, namely,

- (i) $\langle x, y \rangle = \langle y, x \rangle^*$
- (ii) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$
- (iii) $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$
- (iv) $\langle x, x \rangle = \|x\|^2 \geq 0$
- (v) $\langle x, x \rangle = 0$ iff $x = 0$.

We introduced the notation $\|\cdot\|$ for the *norm* or distance in property (iv). The norm satisfies the *triangle inequality*

$$\|x + y\| \leq \|x\| + \|y\| \quad (\text{B.1})$$

and the *Cauchy–Schwarz inequality*

$$|\langle x, y \rangle|^2 \leq \|x\|^2 \|y\|^2, \quad (\text{B.2})$$

which we have seen before for random variables in (A.35). Now, a *Hilbert space*, \mathcal{H} , is defined as an inner product space with the Cauchy property. In other words, \mathcal{H} is a *complete inner product space*. This means that every Cauchy sequence converges in norm; that is, $x_n \rightarrow x \in \mathcal{H}$ if and only if $\|x_n - x_m\| \rightarrow 0$ as $m, n \rightarrow \infty$. This is just the L^2 completeness [Theorem A.1](#) for random variables.

For a broad overview of Hilbert space techniques that are useful in statistical inference and in probability, see Small and McLeish (1994). Also, Brockwell and Davis (1991, Chapter 2) is a nice summary of Hilbert space techniques that are useful

in time series analysis. In our discussions, we mainly use the *projection theorem* (**Theorem B.1**) and the associated *orthogonality principle* as a means for solving various kinds of linear estimation problems.

Theorem B.1 (Projection Theorem) *Let \mathcal{M} be a closed subspace of the Hilbert space \mathcal{H} and let y be an element in \mathcal{H} . Then, y can be uniquely represented as*

$$y = \hat{y} + z, \quad (\text{B.3})$$

where \hat{y} belongs to \mathcal{M} and z is orthogonal to \mathcal{M} ; that is, $\langle z, w \rangle = 0$ for all w in \mathcal{M} . Furthermore, the point \hat{y} is closest to y in the sense that, for any w in \mathcal{M} , $\|y - w\| \geq \|y - \hat{y}\|$, where equality holds if and only if $w = \hat{y}$.

We note that (B.3) and the statement following it yield the *orthogonality property*

$$\langle y - \hat{y}, w \rangle = 0 \quad (\text{B.4})$$

for any w belonging to \mathcal{M} , which can sometimes be used easily to find an expression for the projection. The norm of the error can be written as

$$\begin{aligned} \|y - \hat{y}\|^2 &= \langle y - \hat{y}, y - \hat{y} \rangle \\ &= \langle y - \hat{y}, y \rangle - \langle y - \hat{y}, \hat{y} \rangle \\ &= \langle y - \hat{y}, y \rangle \end{aligned} \quad (\text{B.5})$$

because of orthogonality.

Using the notation of **Theorem B.1**, we call the mapping $P_{\mathcal{M}}y = \hat{y}$, for $y \in \mathcal{H}$, the *projection mapping of \mathcal{H} onto \mathcal{M}* . In addition, the *closed span* of a finite set $\{x_1, \dots, x_n\}$ of elements in a Hilbert space, \mathcal{H} , is defined to be the set of all linear combinations $w = a_1x_1 + \dots + a_nx_n$, where a_1, \dots, a_n are scalars. This subspace of \mathcal{H} is denoted by $\mathcal{M} = \overline{\text{sp}}\{x_1, \dots, x_n\}$. By the projection theorem, the projection of $y \in \mathcal{H}$ onto \mathcal{M} is unique and given by

$$P_{\mathcal{M}}y = a_1x_1 + \dots + a_nx_n,$$

where $\{a_1, \dots, a_n\}$ are found using the orthogonality principle

$$\langle y - P_{\mathcal{M}}y, x_j \rangle = 0 \quad j = 1, \dots, n.$$

Evidently, $\{a_1, \dots, a_n\}$ can be obtained by solving

$$\sum_{i=1}^n a_i \langle x_i, x_j \rangle = \langle y, x_j \rangle \quad j = 1, \dots, n. \quad (\text{B.6})$$

When the elements of \mathcal{H} are vectors, this problem is the linear regression problem.

Example B.1 Linear Regression Analysis

For the regression model introduced in [Section 2.1](#), we want to find the regression coefficients β_i that minimize the residual sum of squares. Consider the vectors $y = (y_1, \dots, y_n)'$ and $z_i = (z_{1i}, \dots, z_{ni})'$, for $i = 1, \dots, q$ and the inner product

$$\langle z_i, y \rangle = \sum_{t=1}^n z_{ti} y_t = z_i' y.$$

We solve the problem of finding a projection of the observed y on the linear space spanned by $\beta_1 z_1 + \dots + \beta_q z_q$, that is, linear combinations of the z_i . The orthogonality principle gives

$$\left\langle y - \sum_{i=1}^q \beta_i z_i, z_j \right\rangle = 0$$

for $j = 1, \dots, q$. Writing the orthogonality condition, as in [\(B.6\)](#), in vector form gives

$$y' z_j = \sum_{i=1}^q \beta_i z_i' z_j \quad j = 1, \dots, q, \tag{B.7}$$

which can be written in the usual matrix form by letting $Z = (z_1, \dots, z_q)$, which is assumed to be full rank. That is, [\(B.7\)](#) can be written as

$$y' Z = \beta' (Z' Z), \tag{B.8}$$

where $\beta = (\beta_1, \dots, \beta_q)'$. Transposing both sides of [\(B.8\)](#) provides the solution for the coefficients,

$$\hat{\beta} = (Z' Z)^{-1} Z' y.$$

The mean-square error in this case would be

$$\left\| y - \sum_{i=1}^q \hat{\beta}_i z_i \right\|^2 = \left\langle y - \sum_{i=1}^q \hat{\beta}_i z_i, y \right\rangle = \langle y, y \rangle - \sum_{i=1}^q \hat{\beta}_i \langle z_i, y \rangle = y' y - \hat{\beta}' Z' y,$$

which is in agreement with [Section 2.1](#).

The extra generality in the above approach hardly seems necessary in the finite dimensional case, where differentiation works perfectly well. It is convenient, however, in many cases to regard the elements of \mathcal{H} as infinite dimensional, so that the orthogonality principle becomes of use. For example, the projection of the process $\{x_t; t = 0 \pm 1, \pm 2, \dots\}$ on the linear manifold spanned by all filtered convolutions of the form

$$\hat{x}_t = \sum_{k=-\infty}^{\infty} a_k x_{t-k}$$

would be in this form.

There are some useful results, which we state without proof, pertaining to projection mappings.

Theorem B.2 Under established notation and conditions:

- (i) $P_{\mathcal{M}}(ax + by) = aP_{\mathcal{M}}x + bP_{\mathcal{M}}y$, for $x, y \in \mathcal{H}$, where a and b are scalars.
- (ii) If $\|y_n - y\| \rightarrow 0$, then $P_{\mathcal{M}}y_n \rightarrow P_{\mathcal{M}}y$, as $n \rightarrow \infty$.
- (iii) $w \in \mathcal{M}$ if and only if $P_{\mathcal{M}}w = w$. Consequently, a projection mapping can be characterized by the property that $P_{\mathcal{M}}^2 = P_{\mathcal{M}}$, in the sense that, for any $y \in \mathcal{H}$, $P_{\mathcal{M}}(P_{\mathcal{M}}y) = P_{\mathcal{M}}y$.
- (iv) Let \mathcal{M}_1 and \mathcal{M}_2 be closed subspaces of \mathcal{H} . Then, $\mathcal{M}_1 \subseteq \mathcal{M}_2$ if and only if $P_{\mathcal{M}_1}(P_{\mathcal{M}_2}y) = P_{\mathcal{M}_1}y$ for all $y \in \mathcal{H}$.
- (v) Let \mathcal{M} be a closed subspace of \mathcal{H} and let \mathcal{M}_{\perp} denote the orthogonal complement of \mathcal{M} . Then, \mathcal{M}_{\perp} is also a closed subspace of \mathcal{H} , and for any $y \in \mathcal{H}$, $y = P_{\mathcal{M}}y + P_{\mathcal{M}_{\perp}}y$.

Part (iii) of [Theorem B.2](#) leads to the well-known result, often used in linear models, that a square matrix M is a projection matrix if and only if it is symmetric and idempotent (that is, $M^2 = M$). For example, using the notation of [Example B.1](#) for linear regression, the projection of y onto $\overline{\text{sp}}\{z_1, \dots, z_q\}$, the space generated by the columns of Z , is $P_Z(y) = Z\hat{\beta} = Z(Z'Z)^{-1}Z'y$. The matrix $M = Z(Z'Z)^{-1}Z'$ is an $n \times n$, symmetric and idempotent matrix of rank q (which is the dimension of the space that M projects y onto). Parts (iv) and (v) of [Theorem B.2](#) are useful for establishing recursive solutions for estimation and prediction.

By imposing extra structure, *conditional expectation* can be defined as a projection mapping for random variables in L^2 with the equivalence relation that, for $x, y \in L^2$, $x = y$ if $\Pr(x = y) = 1$. In particular, for $y \in L^2$, if \mathcal{M} is a closed subspace of L^2 containing 1, the conditional expectation of y given \mathcal{M} is defined to be the projection of y onto \mathcal{M} , namely, $E_{\mathcal{M}}y = P_{\mathcal{M}}y$. This means that conditional expectation, $E_{\mathcal{M}}$, must satisfy the orthogonality principle of the Projection Theorem and that the results of [Theorem B.2](#) remain valid (the most ly used tool in this case is item (iv) of the theorem). If we let $\mathcal{M}(x)$ denote the closed subspace of all random variables in L^2 that can be written as a (measurable) function of x , then we may define, for $x, y \in L^2$, the *conditional expectation of y given x* as $E(y | x) = E_{\mathcal{M}(x)}y$. This idea may be generalized in an obvious way to define the conditional expectation of y given $x_{1:n} = (x_1, \dots, x_n)$; that is $E(y | x) = E_{\mathcal{M}(x)}y$. Of particular interest to us is the following result which states that, in the Gaussian case, conditional expectation and linear prediction are equivalent.

Theorem B.3 Under established notation and conditions, if (y, x_1, \dots, x_n) is multivariate normal, then

$$E(y | x_{1:n}) = P_{\overline{\text{sp}}\{1, x_1, \dots, x_n\}}y.$$

Proof: First, by the projection theorem, the conditional expectation of y given $x_{1:n}$ is the unique element $E_{\mathcal{M}(x)}y$ that satisfies the orthogonality principle,

$$E\{(y - E_{\mathcal{M}(x)}y) w\} = 0 \quad \text{for all } w \in \mathcal{M}(x).$$

We will show that $\hat{y} = P_{\overline{\text{sp}}\{1, x_1, \dots, x_n\}}y$ is that element. In fact, by the projection theorem, \hat{y} satisfies

$$\langle y - \hat{y}, x_i \rangle = 0 \quad \text{for } i = 0, 1, \dots, n,$$

where we have set $x_0 = 1$. But $\langle y - \hat{y}, x_i \rangle = \text{cov}(y - \hat{y}, x_i) = 0$, implying that $y - \hat{y}$ and (x_1, \dots, x_n) are independent because the vector $(y - \hat{y}, x_1, \dots, x_n)'$ is multivariate normal. Thus, if $w \in \mathcal{M}(x)$, then w and $y - \hat{y}$ are independent and, hence, $\langle y - \hat{y}, w \rangle = E\{(y - \hat{y})w\} = E(y - \hat{y})E(w) = 0$, recalling that $0 = \langle y - \hat{y}, 1 \rangle = E(y - \hat{y})$. \square

In the Gaussian case, conditional expectation has an explicit form. Let $y = (y_1, \dots, y_m)'$, $x = (x_1, \dots, x_n)'$, and suppose the x and y are jointly normal:

$$\begin{pmatrix} y \\ x \end{pmatrix} \sim N_{m+n} \left[\begin{pmatrix} \mu_y \\ \mu_x \end{pmatrix}, \begin{pmatrix} \Sigma_{yy} & \Sigma_{yx} \\ \Sigma_{xy} & \Sigma_{xx} \end{pmatrix} \right],$$

then $y | x$ is normal with

$$\mu_{y|x} = \mu_y + \Sigma_{yx} \Sigma_{xx}^{-1} (x - \mu_x) \quad (\text{B.9})$$

$$\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy}, \quad (\text{B.10})$$

where Σ_{xx} is assumed to be nonsingular.

B.2 Causal Conditions for ARMA Models

In this section, we prove **Property 3.1** of **Section 3.1** pertaining to the causality of ARMA models. The proof of **Property 3.2**, which pertains to invertibility of ARMA models, is similar.

Proof of Property 3.1. Suppose first that the roots of $\phi(z)$, say, z_1, \dots, z_p , lie outside the unit circle. We write the roots in the following order, $1 < |z_1| \leq |z_2| \leq \dots \leq |z_p|$, noting that z_1, \dots, z_p are not necessarily unique, and put $|z_1| = 1 + \epsilon$, for some $\epsilon > 0$. Thus, $\phi(z) \neq 0$ as long as $|z| < |z_1| = 1 + \epsilon$ and, hence, $\phi^{-1}(z)$ exists and has a power series expansion,

$$\frac{1}{\phi(z)} = \sum_{j=0}^{\infty} a_j z^j, \quad |z| < 1 + \epsilon.$$

Now, choose a value δ such that $0 < \delta < \epsilon$, and set $z = 1 + \delta$, which is inside the radius of convergence. It then follows that

$$\phi^{-1}(1 + \delta) = \sum_{j=0}^{\infty} a_j (1 + \delta)^j < \infty. \quad (\text{B.11})$$

Thus, we can bound each of the terms in the sum in (B.11) by a constant, say, $|a_j(1 + \delta)^j| < c$, for $c > 0$. In turn, $|a_j| < c(1 + \delta)^{-j}$, from which it follows that

$$\sum_{j=0}^{\infty} |a_j| < \infty. \quad (\text{B.12})$$

Hence, $\phi^{-1}(B)$ exists and we may apply it to both sides of the ARMA model, $\phi(B)x_t = \theta(B)w_t$, to obtain

$$x_t = \phi^{-1}(B)\phi(B)x_t = \phi^{-1}(B)\theta(B)w_t.$$

Thus, putting $\psi(B) = \phi^{-1}(B)\theta(B)$, we have

$$x_t = \psi(B)w_t = \sum_{j=0}^{\infty} \psi_j w_{t-j},$$

where the ψ -weights, which are absolutely summable, can be evaluated by $\psi(z) = \phi^{-1}(z)\theta(z)$, for $|z| \leq 1$.

Now, suppose x_t is a causal process; that is, it has the representation

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}, \quad \sum_{j=0}^{\infty} |\psi_j| < \infty.$$

In this case, we write

$$x_t = \psi(B)w_t,$$

and premultiplying by $\phi(B)$ yields

$$\phi(B)x_t = \phi(B)\psi(B)w_t. \quad (\text{B.13})$$

In addition to (B.13), the model is ARMA, and can be written as

$$\phi(B)x_t = \theta(B)w_t. \quad (\text{B.14})$$

From (B.13) and (B.14), we see that

$$\phi(B)\psi(B)w_t = \theta(B)w_t. \quad (\text{B.15})$$

Now, let

$$a(z) = \phi(z)\psi(z) = \sum_{j=0}^{\infty} a_j z^j \quad |z| \leq 1$$

and, hence, we can write (B.15) as

$$\sum_{j=0}^{\infty} a_j w_{t-j} = \sum_{j=0}^q \theta_j w_{t-j}. \quad (\text{B.16})$$

Next, multiply both sides of (B.16) by w_{t-h} , for $h = 0, 1, 2, \dots$, and take expectation. In doing this, we obtain

$$\begin{aligned} a_h &= \theta_h, \quad h = 0, 1, \dots, q \\ a_h &= 0, \quad h > q. \end{aligned} \quad (\text{B.17})$$

From (B.17), we conclude that

$$\phi(z)\psi(z) = a(z) = \theta(z), \quad |z| \leq 1. \quad (\text{B.18})$$

If there is a complex number in the unit circle, say z_0 , for which $\phi(z_0) = 0$, then by (B.18), $\theta(z_0) = 0$. But, if there is such a z_0 , then $\phi(z)$ and $\theta(z)$ have a common factor which is not allowed. Thus, we may write $\psi(z) = \theta(z)/\phi(z)$. In addition, by hypothesis, we have that $|\psi(z)| < \infty$ for $|z| \leq 1$, and hence

$$|\psi(z)| = \left| \frac{\theta(z)}{\phi(z)} \right| < \infty, \quad \text{for } |z| \leq 1. \quad (\text{B.19})$$

Finally, (B.19) implies $\phi(z) \neq 0$ for $|z| \leq 1$; that is, the roots of $\phi(z)$ lie outside the unit circle. \square

B.3 Large Sample Distribution of the AR Conditional Least Squares Estimators

In Section 3.5 we discussed the conditional least squares procedure for estimating the parameters $\phi_1, \phi_2, \dots, \phi_p$ and σ_w^2 in the AR(p) model

$$x_t = \sum_{k=1}^p \phi_k x_{t-k} + w_t,$$

where we assume $\mu = 0$, for convenience. Write the model as

$$x_t = \phi' x_{t-1} + w_t, \quad (\text{B.20})$$

where $x_{t-1} = (x_{t-1}, x_{t-2}, \dots, x_{t-p})'$ is a $p \times 1$ vector of lagged values, and $\phi = (\phi_1, \phi_2, \dots, \phi_p)'$ is the $p \times 1$ vector of regression coefficients. Assuming observations are available at x_1, \dots, x_n , the conditional least squares procedure is to minimize

$$S_c(\phi) = \sum_{t=p+1}^n (x_t - \phi' x_{t-1})^2$$

with respect to ϕ . The solution is

$$\hat{\phi} = \left(\sum_{t=p+1}^n x_{t-1} x_{t-1}' \right)^{-1} \sum_{t=p+1}^n x_{t-1} x_t \quad (\text{B.21})$$

for the regression vector ϕ ; the conditional least squares estimate of σ_w^2 is

$$\hat{\sigma}_w^2 = \frac{1}{n-p} \sum_{t=p+1}^n (x_t - \hat{\phi}' x_{t-1})^2. \quad (\text{B.22})$$

As pointed out following (3.116), Yule–Walker estimators and least squares estimators are approximately the same in that the estimators differ only by inclusion or

exclusion of terms involving the endpoints of the data. Hence, it is easy to show the asymptotic equivalence of the two estimators; this is why, for AR(p) models, (3.103) and (3.132), are equivalent. Details on the asymptotic equivalence can be found in Brockwell and Davis (1991, Chapter 8).

Here, we use the same approach as in [Appendix A](#), replacing the lower limits of the sums in (B.21) and (B.22) by one and noting the asymptotic equivalence of the estimators

$$\tilde{\phi} = \left(\sum_{t=1}^n x_{t-1} x'_{t-1} \right)^{-1} \sum_{t=1}^n x_{t-1} x_t \quad (\text{B.23})$$

and

$$\tilde{\sigma}_w^2 = \frac{1}{n} \sum_{t=1}^n (x_t - \tilde{\phi}' x_{t-1})^2 \quad (\text{B.24})$$

to those two estimators. In (B.23) and (B.24), we are acting as if we are able to observe x_{1-p}, \dots, x_0 in addition to x_1, \dots, x_n . The asymptotic equivalence is then seen by arguing that for n sufficiently large, it makes no difference whether or not we observe x_{1-p}, \dots, x_0 . In the case of (B.23) and (B.24), we obtain the following theorem.

Theorem B.4 *Let x_t be a causal AR(p) series with white (iid) noise w_t satisfying $E(w_t^4) = \eta \sigma_w^4$. Then,*

$$\tilde{\phi} \sim \text{AN}\left(\phi, n^{-1} \sigma_w^2 \Gamma_p^{-1}\right), \quad (\text{B.25})$$

where $\Gamma_p = \{\gamma(i-j)\}_{i,j=1}^p$ is the $p \times p$ autocovariance matrix of the vector x_{t-1} . We also have, as $n \rightarrow \infty$,

$$n^{-1} \sum_{t=1}^n x_{t-1} x'_{t-1} \xrightarrow{P} \Gamma_p \quad \text{and} \quad \tilde{\sigma}_w^2 \xrightarrow{P} \sigma_w^2. \quad (\text{B.26})$$

Proof: First, (B.26) follows from the fact that $E(x_{t-1} x'_{t-1}) = \Gamma_p$, recalling that from [Theorem A.6](#), second-order sample moments converge in probability to their population moments for linear processes in which w_t has a finite fourth moment. To show (B.25), we can write

$$\begin{aligned} \tilde{\phi} &= \left(\sum_{t=1}^n x_{t-1} x'_{t-1} \right)^{-1} \sum_{t=1}^n x_{t-1} (x'_{t-1} \phi + w_t) \\ &= \phi + \left(\sum_{t=1}^n x_{t-1} x'_{t-1} \right)^{-1} \sum_{t=1}^n x_{t-1} w_t, \end{aligned}$$

so that

$$n^{1/2}(\tilde{\phi} - \phi) = \left(n^{-1} \sum_{t=1}^n x_{t-1} x'_{t-1} \right)^{-1} n^{-1/2} \sum_{t=1}^n x_{t-1} w_t$$

$$= \left(n^{-1} \sum_{t=1}^n x_{t-1} x'_{t-1} \right)^{-1} n^{-1/2} \sum_{t=1}^n u_t,$$

where $u_t = x_{t-1} w_t$. We use the fact that w_t and x_{t-1} are independent to write $Eu_t = E(x_{t-1})E(w_t) = 0$, because the errors have zero means. Also,

$$Eu_t u'_t = Ex_{t-1} w_t w_t x'_{t-1} = Ex_{t-1} x'_{t-1} Ew_t^2 = \sigma_w^2 \Gamma_p.$$

In addition, we have, for $h > 0$,

$$Eu_{t+h} u'_t = Ex_{t+h-1} w_{t+h} w_t x'_{t-1} = Ex_{t+h-1} w_t x'_{t-1} Ew_{t+h} = 0.$$

A similar computation works for $h < 0$.

Next, consider the mean square convergent approximation

$$x_t^m = \sum_{j=0}^m \psi_j w_{t-j}$$

for x_t , and define the $(m+p)$ -dependent process $u_t^m = w_t(x_{t-1}^m, x_{t-2}^m, \dots, x_{t-p}^m)'$. Note that we need only look at a central limit theorem for the sum

$$y_{nm} = n^{-1/2} \sum_{t=1}^n \lambda' u_t^m,$$

for arbitrary vectors $\lambda = (\lambda_1, \dots, \lambda_p)'$, where y_{nm} is used as an approximation to

$$S_n = n^{-1/2} \sum_{t=1}^n \lambda' u_t.$$

First, apply the m -dependent central limit theorem to y_{nm} as $n \rightarrow \infty$ for fixed m to establish (i) of [Theorem A.2](#). This result shows $y_{nm} \xrightarrow{d} y_m$, where y_m is asymptotically normal with covariance $\lambda' \Gamma_p^{(m)} \lambda$, where $\Gamma_p^{(m)}$ is the covariance matrix of u_t^m . Then, we have $\Gamma_p^{(m)} \rightarrow \Gamma_p$, so that y_m converges in distribution to a normal random variable with mean zero and variance $\lambda' \Gamma_p \lambda$ and we have verified part (ii) of [Theorem A.2](#). We verify part (iii) of [Theorem A.2](#) by noting that

$$E[(S_n - y_{nm})^2] = n^{-1} \sum_{t=1}^n \lambda' E[(u_t - u_t^m)(u_t - u_t^m)'] \lambda$$

clearly converges to zero as $n, m \rightarrow \infty$ because

$$x_t - x_t^m = \sum_{j=m+1}^{\infty} \psi_j w_{t-j}$$

form the components of $u_t - u_t^m$.

Now, the form for $\sqrt{n}(\tilde{\phi} - \phi)$ contains the premultiplying matrix

$$\left(n^{-1} \sum_{t=1}^n x_{t-1} x'_{t-1} \right)^{-1} \xrightarrow{P} \Gamma_p^{-1},$$

because (A.22) can be applied to the function that defines the inverse of the matrix. Then, applying (A.30), shows that

$$n^{1/2} (\tilde{\phi} - \phi) \xrightarrow{d} N\left(0, \sigma_w^2 \Gamma_p^{-1} \Gamma_p \Gamma_p^{-1}\right),$$

so we may regard it as being multivariate normal with mean zero and covariance matrix $\sigma_w^2 \Gamma_p^{-1}$.

To investigate $\tilde{\sigma}_w^2$, note

$$\begin{aligned} \tilde{\sigma}_w^2 &= n^{-1} \sum_{t=1}^n (x_t - \tilde{\phi}' x_{t-1})^2 \\ &= n^{-1} \sum_{t=1}^n x_t^2 - n^{-1} \sum_{t=1}^n x'_{t-1} x_t \left(n^{-1} \sum_{t=1}^n x_{t-1} x'_{t-1} \right)^{-1} n^{-1} \sum_{t=1}^n x_{t-1} x_t \\ &\xrightarrow{P} \gamma(0) - \gamma'_p \Gamma_p^{-1} \gamma_p \\ &= \sigma_w^2, \end{aligned}$$

and we have that the sample estimator converges in probability to σ_w^2 , which is written in the form of (3.66). \square

The arguments above imply that, for sufficiently large n , we may consider the estimator $\hat{\phi}$ in (B.21) as being approximately multivariate normal with mean ϕ and variance–covariance matrix $\sigma_w^2 \Gamma_p^{-1}/n$. Inferences about the parameter ϕ are obtained by replacing the σ_w^2 and Γ_p by their estimates given by (B.22) and

$$\hat{\Gamma}_p = n^{-1} \sum_{t=p+1}^n x_{t-1} x'_{t-1},$$

respectively. In the case of a nonzero mean, the data x_t are replaced by $x_t - \bar{x}$ in the estimates and the results of Theorem A.2 remain valid.

B.4 The Wold Decomposition

The ARMA approach to modeling time series is generally implied by the assumption that the dependence between adjacent values in time is best explained in terms of a regression of the current values on the past values. This assumption is partially justified, in theory, by the Wold decomposition.

In this section we assume that $\{x_t; t = 0, \pm 1, \pm 2, \dots\}$ is a stationary, mean-zero process. Using the notation of Section B.1, we define

$$\mathcal{M}_n^x = \overline{\text{sp}}\{x_t, -\infty < t \leq n\}, \quad \text{with} \quad \mathcal{M}_{-\infty}^x = \bigcap_{n=-\infty}^{\infty} \mathcal{M}_n^x,$$

and

$$\sigma_x^2 = E(x_{n+1} - P_{\mathcal{M}_n^x} x_{n+1})^2.$$

We say that x_t is a *deterministic process* if and only if $\sigma_x^2 = 0$. That is, a deterministic process is one in which its future is perfectly predictable from its past; a simple example is the process given in (4.1). We are now ready to present the decomposition.

Theorem B.5 (The Wold Decomposition) *Under the conditions and notation of this section, if $\sigma_x^2 > 0$, then x_t can be expressed as*

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} + v_t$$

where

- (i) $\sum_{j=0}^{\infty} \psi_j^2 < \infty$ ($\psi_0 = 1$)
- (ii) $\{w_t\}$ is white noise with variance σ_w^2
- (iii) $w_t \in \mathcal{M}_t^x$
- (iv) $\text{cov}(w_s, v_t) = 0$ for all $s, t = 0, \pm 1, \pm 2, \dots$
- (v) $v_t \in \mathcal{M}_{-\infty}^x$
- (vi) $\{v_t\}$ is deterministic.

The proof of the decomposition follows from the theory of Section B.1 by defining the unique sequences:

$$\begin{aligned} w_t &= x_t - P_{\mathcal{M}_{t-1}^x} x_t, \\ \psi_j &= \sigma_w^{-2} \langle x_t, w_{t-j} \rangle = \sigma_w^{-2} E(x_t w_{t-j}), \\ v_t &= x_t - \sum_{j=0}^{\infty} \psi_j w_{t-j}. \end{aligned}$$

Although every stationary process can be represented by the Wold decomposition, it does not mean that the decomposition is the best way to describe the process. In addition, there may be some dependence structure among the $\{w_t\}$; we are only guaranteed that the sequence is an uncorrelated sequence. The theorem, in its generality, falls short of our needs because we would prefer the noise process, $\{w_t\}$, to be white independent noise. But, the decomposition does give us the confidence that we will not be completely off the mark by fitting ARMA models to time series data.

Appendix C

Spectral Domain Theory

C.1 Spectral Representation Theorems

In this section, we present a spectral representation for the process x_t itself, which allows us to think of a stationary process as a random sum of sines and cosines as described in (4.4). In addition, we present results that justify representing the autocovariance function of a weakly stationary process in terms of a spectral distribution function.

First, we consider developing a representation for the autocovariance function of a stationary, possibly complex, series x_t with zero mean and autocovariance function $\gamma_x(h) = E(x_{t+h}x_t^*)$. An autocovariance function, $\gamma(h)$, is non-negative definite in that, for any set of complex constants, $\{a_t \in \mathbb{C}; t = 1, \dots, n\}$, and any integer $n > 0$,

$$\sum_{s=1}^n \sum_{t=1}^n a_s^* \gamma(s-t) a_t \geq 0.$$

Likewise, any non-negative definite function, say $\gamma(h)$, on the integers is an autocovariance of some stationary process. To see this, let $\Gamma_n = \{\gamma(t_i - t_j)\}_{i,j=1}^n$ be the $n \times n$ matrix with i, j th equal to $\gamma(t_i - t_j)$. Then choose $\{x_t\}$ such that $(x_{t_1}, \dots, x_{t_n}) \sim N_n(0, \Gamma_n)$.

We now establish the relationship of such functions to a spectral distribution function; Riemann-Stieljes integration is explained in [Section C.4.1](#).

Theorem C.1 *A function $\gamma(h)$, for $h = 0, \pm 1, \pm 2, \dots$, is non-negative definite if and only if it can be expressed as*

$$\gamma(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \exp\{2\pi i \omega h\} dF(\omega), \quad (\text{C.1})$$

where $F(\cdot)$ is nondecreasing. The function $F(\cdot)$ is right continuous, bounded and uniquely determined by the conditions $F(\omega) = F(-1/2) = 0$ for $\omega \leq -1/2$ and $F(\omega) = F(1/2) = \gamma(0)$ for $\omega \geq 1/2$.

Proof: If $\gamma(h)$ has the representation (C.1), then

$$\begin{aligned} \sum_{s=1}^n \sum_{t=1}^n a_s^* \gamma(s-t) a_t &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{s=1}^n \sum_{t=1}^n a_s^* a_t e^{2\pi i \omega(s-t)} dF(\omega) \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left| \sum_{t=1}^n a_t e^{-2\pi i \omega t} \right|^2 dF(\omega) \geq 0 \end{aligned}$$

and $\gamma(h)$ is non-negative definite.

Conversely, suppose $\gamma(h)$ is a non-negative definite function. Define the non-negative function

$$\begin{aligned} f_n(\omega) &= n^{-1} \sum_{s=1}^n \sum_{t=1}^n e^{-2\pi i \omega s} \gamma(s-t) e^{2\pi i \omega t} \\ &= n^{-1} \sum_{h=-(n-1)}^{(n-1)} (n - |h|) e^{-2\pi i \omega h} \gamma(h) \geq 0 \end{aligned} \tag{C.2}$$

Now, let $F_n(\omega)$ be the distribution function corresponding to $f_n(\omega)I_{(-1/2, 1/2]}$, where $I_{(\cdot)}$ denotes the indicator function of the interval in the subscript. Note that $F_n(\omega) = 0, \omega \leq -1/2$ and $F_n(\omega) = F_n(1/2)$ for $\omega \geq 1/2$. Then,

$$\begin{aligned} \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF_n(\omega) &= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} f_n(\omega) d\omega \\ &= \begin{cases} (1 - |h|/n) \gamma(h), & |h| < n \\ 0, & \text{elsewhere.} \end{cases} \end{aligned}$$

We also have

$$\begin{aligned} F_n(1/2) &= \int_{-\frac{1}{2}}^{\frac{1}{2}} f_n(\omega) d\omega \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} \sum_{|h|< n} (1 - |h|/n) \gamma(h) e^{-2\pi i \omega h} d\omega = \gamma(0). \end{aligned}$$

Now, by Helly's first convergence theorem (Bhat, 1985, p. 157), there exists a subsequence F_{n_k} converging to F , and by the Helly-Bray Lemma (see Bhat, p. 157), this implies

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF_{n_k}(\omega) \rightarrow \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF(\omega)$$

and, from the right-hand side of the earlier equation,

$$(1 - |h|/n_k) \gamma(h) \rightarrow \gamma(h)$$

as $n_k \rightarrow \infty$, and the required result follows. \square

Next, we present the version of the spectral representation theorem of a mean-zero, stationary process, x_t in terms of an orthogonal increment process. This version allows us to think of a stationary process as being generated (approximately) by a random sum of sines and cosines such as described in (4.4). We refer the reader to Hannan (1970, §2.3) for details.

Theorem C.2 *If x_t is a mean-zero stationary process, with spectral distribution $F(\omega)$ as given in Theorem C.1, then there exists a complex-valued stochastic process $Z(\omega)$, on the interval $\omega \in [-1/2, 1/2]$, having stationary uncorrelated increments, such that x_t can be written as the stochastic integral*

$$x_t = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega t} dZ(\omega),$$

where, for $-1/2 \leq \omega_1 \leq \omega_2 \leq 1/2$,

$$\text{var}\{Z(\omega_2) - Z(\omega_1)\} = F(\omega_2) - F(\omega_1).$$

The theorem uses stochastic integration and orthogonal increment processes, which are described in further detail in Section C.4.2.

In general, the spectral distribution function can be a mixture of discrete and continuous distributions. The special case of greatest interest is the absolutely continuous case, namely, when $dF(\omega) = f(\omega)d\omega$, and the resulting function is the spectral density considered in Section 4.2. What made the proof of Theorem C.1 difficult was that, after we defined

$$f_n(\omega) = \sum_{h=-(n-1)}^{(n-1)} \left(1 - \frac{|h|}{n}\right) \gamma(h) e^{-2\pi i \omega h}$$

in (C.2), we could not simply allow $n \rightarrow \infty$ because $\gamma(h)$ may not be absolutely summable. If, however, $\gamma(h)$ is absolutely summable we may define $f(\omega) = \lim_{n \rightarrow \infty} f_n(\omega)$, and we have the following result.

Theorem C.3 *If $\gamma(h)$ is the autocovariance function of a stationary process, x_t , with*

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty, \tag{C.3}$$

then the spectral density of x_t is given by

$$f(\omega) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega h}. \tag{C.4}$$

We may extend the representation to the vector case $x_t = (x_{t1}, \dots, x_{tp})'$ by considering linear combinations of the form

$$y_t = \sum_{j=1}^p a_j^* x_{tj},$$

which will be stationary with autocovariance functions of the form

$$\gamma_y(h) = \sum_{j=1}^p \sum_{k=1}^p a_j^* \gamma_{jk}(h) a_k,$$

where $\gamma_{jk}(h)$ is the usual cross-covariance function between x_{tj} and x_{tk} . To develop the spectral representation of $\gamma_{jk}(h)$ from the representations of the univariate series, consider the linear combinations

$$y_{t1} = x_{tj} + x_{tk} \quad \text{and} \quad y_{t2} = x_{tj} + i x_{tk},$$

which are both stationary series with respective covariance functions

$$\begin{aligned} \gamma_1(h) &= \gamma_{jj}(h) + \gamma_{jk}(h) + \gamma_{kj}(h) + \gamma_{kk}(h) \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dG_1(\omega), \end{aligned}$$

$$\begin{aligned} \gamma_2(h) &= \gamma_{jj}(h) + i\gamma_{kj}(h) - i\gamma_{jk}(h) + \gamma_{kk}(h) \\ &= \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dG_2(\omega). \end{aligned}$$

Introducing the spectral representations for $\gamma_{jj}(h)$ and $\gamma_{kk}(h)$ yields

$$\gamma_{jk}(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF_{jk}(\omega),$$

with

$$F_{jk}(\omega) = \frac{1}{2} \left[G_1(\omega) + iG_2(\omega) - (1+i)(F_{jj}(\omega) + F_{kk}(\omega)) \right].$$

Now, under the summability condition

$$\sum_{h=-\infty}^{\infty} |\gamma_{jk}(h)| < \infty,$$

we have the representation

$$\gamma_{jk}(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} f_{jk}(\omega) d\omega,$$

where the cross-spectral density function has the inverse Fourier representation

$$f_{jk}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{jk}(h) e^{-2\pi i \omega h}.$$

The cross-covariance function satisfies $\gamma_{jk}(h) = \gamma_{kj}(-h)$, which implies $f_{jk}(\omega) = f_{kj}(-\omega)$ using the above representation.

Then, defining the autocovariance function of the general vector process x_t as the $p \times p$ matrix

$$\Gamma(h) = E[(x_{t+h} - \mu_x)(x_t - \mu_x)'],$$

and the $p \times p$ spectral matrix as $f(\omega) = \{f_{jk}(\omega); j, k = 1, \dots, p\}$, we have the representation in matrix form, written as

$$\Gamma(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} f(\omega) d\omega, \quad (\text{C.5})$$

and the inverse result

$$f(\omega) = \sum_{h=-\infty}^{\infty} \Gamma(h) e^{-2\pi i \omega h}. \quad (\text{C.6})$$

which appears as [Property 4.8 in Section 4.5](#). [Theorem C.2](#) can also be extended to the multivariate case.

C.2 Large Sample Distribution of the Smoothed Periodogram

We have previously introduced the DFT, for the stationary zero-mean process x_t , observed at $t = 1, \dots, n$ as

$$d(\omega) = n^{-1/2} \sum_{t=1}^n x_t e^{-2\pi i \omega t}, \quad (\text{C.7})$$

as the result of matching sines and cosines of frequency ω against the series x_t . We will suppose now that x_t has an absolutely continuous spectrum $f(\omega)$ corresponding to the absolutely summable autocovariance function $\gamma(h)$. Our purpose in this section is to examine the statistical properties of the complex random variables $d(\omega_k)$, for $\omega_k = k/n$, $k = 0, 1, \dots, n-1$ in providing a basis for the estimation of $f(\omega)$. To develop the statistical properties, we examine the behavior of

$$\begin{aligned} S_n(\omega, \omega) &= E |d(\omega)|^2 = n^{-1} E \left[\sum_{s=1}^n x_s e^{-2\pi i \omega s} \sum_{t=1}^n x_t e^{2\pi i \omega t} \right] \\ &= n^{-1} \sum_{s=1}^n \sum_{t=1}^n e^{-2\pi i \omega s} e^{2\pi i \omega t} \gamma(s-t) \\ &= \sum_{h=-(n-1)}^{n-1} (1 - |h|/n) \gamma(h) e^{-2\pi i \omega h}, \end{aligned} \quad (\text{C.8})$$

where we have let $h = s - t$. Using dominated convergence,

$$S_n(\omega, \omega) \rightarrow \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega h} = f(\omega),$$

as $n \rightarrow \infty$, making the large sample variance of the Fourier transform equal to the spectrum evaluated at ω . We have already seen this result in [Theorem C.3](#). For exact bounds it is also convenient to add an absolute summability assumption for the autocovariance function, namely,

$$\theta = \sum_{h=-\infty}^{\infty} |h| |\gamma(h)| < \infty. \quad (\text{C.9})$$

Example C.1 Condition (C.9) Verified for ARMA Models

For pure MA(q) models [ARMA(0, q)], $\gamma(h) = 0$ for $|h| > q$, so the condition holds trivially. In [Section 3.3](#), we showed that when $p > 0$, the autocovariance function $\gamma(h)$ behaves like the inverse of the roots of the AR polynomial to the power h . Recalling [\(3.50\)](#), we can write

$$\gamma(h) \sim |h|^k \xi^h,$$

for large h , where $\xi = |z|^{-1} \in (0, 1)$, z is a root of the AR polynomial, and $0 \leq k \leq p - 1$ is some integer depending on the multiplicity of the root.

We show that $\sum_{h \geq 0} h \xi^h$ is finite, the other cases follow in a similar manner. Note the $\sum_{h \geq 0} \xi^h = 1/(1 - \xi)$ because it is a geometric sum. Taking derivatives, we have $\sum_{h \geq 0} h \xi^{h-1} = 1/(1 - \xi)^2$ and multiplying through by ξ , we have $\sum_{h \geq 0} h \xi^h = \xi/(1 - \xi)^2$. For other values of k , follow the recipe but take k th derivatives.

To elaborate further, we derive two approximation lemmas.

Lemma C.1 *For $S_n(\omega, \omega)$ as defined in (C.8) and θ in (C.9) finite, we have*

$$|S_n(\omega, \omega) - f(\omega)| \leq \frac{\theta}{n} \quad (\text{C.10})$$

or

$$S_n(\omega, \omega) = f(\omega) + O(n^{-1}). \quad (\text{C.11})$$

Proof: To prove the lemma, write

$$\begin{aligned} n|S_n(\omega, \omega) - f(\omega)| &= \left| \sum_{|u|< n} (n - |u|) \gamma(u) e^{-2\pi i \omega u} - n \sum_{u=-\infty}^{\infty} \gamma(u) e^{-2\pi i \omega u} \right| \\ &= \left| -n \sum_{|u| \geq n} \gamma(u) e^{-2\pi i \omega u} - \sum_{|u|< n} |u| \gamma(u) e^{-2\pi i \omega u} \right| \\ &\leq \sum_{|u| \geq n} |u| |\gamma(u)| + \sum_{|u|< n} |u| |\gamma(u)| \\ &= \theta, \end{aligned}$$

which establishes the lemma. \square

Lemma C.2 For $\omega_k = k/n$, $\omega_\ell = \ell/n$, $\omega_k - \omega_\ell \neq 0, \pm 1, \pm 2, \pm 3, \dots$, and θ in (C.9), we have

$$|S_n(\omega_k, \omega_\ell)| \leq \frac{\theta}{n} = O(n^{-1}), \quad (\text{C.12})$$

where

$$S_n(\omega_k, \omega_\ell) = \mathbb{E}\{d(\omega_k)d^*(\omega_\ell)\}. \quad (\text{C.13})$$

Proof: Write

$$\begin{aligned} n|S_n(\omega_k, \omega_\ell)| &= \sum_{u=-(n-1)}^{-1} \gamma(u) \sum_{v=-(u-1)}^n e^{-2\pi i(\omega_k - \omega_\ell)v} e^{-2\pi i\omega_k u} \\ &\quad + \sum_{u=0}^{n-1} \gamma(u) \sum_{v=1}^{n-u} e^{-2\pi i(\omega_k - \omega_\ell)v} e^{-2\pi i\omega_k u}. \end{aligned}$$

Now, for the first term, with $u < 0$,

$$\begin{aligned} \sum_{v=-(u-1)}^n e^{-2\pi i(\omega_k - \omega_\ell)v} &= \left(\sum_{v=1}^n - \sum_{v=1}^{-u} \right) e^{-2\pi i(\omega_k - \omega_\ell)v} \\ &= 0 - \sum_{v=1}^{-u} e^{-2\pi i(\omega_k - \omega_\ell)v}. \end{aligned}$$

For the second term with $u \geq 0$,

$$\begin{aligned} \sum_{v=1}^{n-u} e^{-2\pi i(\omega_k - \omega_\ell)v} &= \left(\sum_{v=1}^n - \sum_{v=n-u+1}^n \right) e^{-2\pi i(\omega_k - \omega_\ell)v} \\ &= 0 - \sum_{v=n-u+1}^n e^{-2\pi i(\omega_k - \omega_\ell)v}. \end{aligned}$$

Consequently,

$$\begin{aligned} n|S_n(\omega_k, \omega_\ell)| &= \left| - \sum_{u=-(n-1)}^{-1} \gamma(u) \sum_{v=1}^{-u} e^{-2\pi i(\omega_k - \omega_\ell)v} e^{-2\pi i\omega_k u} \right. \\ &\quad \left. - \sum_{u=1}^{n-1} \gamma(u) \sum_{v=n-u+1}^n e^{-2\pi i(\omega_k - \omega_\ell)v} e^{-2\pi i\omega_k u} \right| \\ &\leq \sum_{u=-(n-1)}^0 (-u)|\gamma(u)| + \sum_{u=1}^{n-1} u|\gamma(u)| \\ &= \sum_{u=-(n-1)}^{(n-1)} |u| |\gamma(u)|. \end{aligned}$$

Hence, we have

$$S_n(\omega_k, \omega_\ell) \leq \frac{\theta}{n},$$

and the asserted relations of the lemma follow. \square

Because the DFTs are approximately uncorrelated, say, of order $1/n$, when the frequencies are of the form $\omega_k = k/n$, we shall compute at those frequencies. The behavior of $f(\omega)$ at neighboring frequencies will often be of interest and we shall use **Lemma C.3** below to handle such cases.

Lemma C.3 *For $|\omega_k - \omega| \leq L/2n$ and θ in (C.9), we have*

$$|f(\omega_k) - f(\omega)| \leq \frac{\pi\theta L}{n} \quad (\text{C.14})$$

or

$$f(\omega_k) - f(\omega) = O(L/n). \quad (\text{C.15})$$

Proof: Write the difference

$$\begin{aligned} |f(\omega_k) - f(\omega)| &= \left| \sum_{h=-\infty}^{\infty} \gamma(h) \left(e^{-2\pi i \omega_k h} - e^{-2\pi i \omega h} \right) \right| \\ &\leq \sum_{h=-\infty}^{\infty} |\gamma(h)| \left| e^{-\pi i (\omega_k - \omega) h} - e^{\pi i (\omega_k - \omega) h} \right| \\ &= 2 \sum_{h=-\infty}^{\infty} |\gamma(h)| \left| \sin[\pi(\omega_k - \omega)h] \right| \\ &\leq 2\pi |\omega_k - \omega| \sum_{h=-\infty}^{\infty} |h| |\gamma(h)| \\ &\leq \frac{\pi\theta L}{n} \end{aligned}$$

because $|\sin x| \leq |x|$. \square

The main use of the properties described by **Lemma C.1** and **Lemma C.2** is in identifying the covariance structure of the DFT, say,

$$d(\omega_k) = n^{-1/2} \sum_{t=1}^n x_t e^{-2\pi i \omega_k t} = d_c(\omega_k) - i d_s(\omega_k),$$

where

$$d_c(\omega_k) = n^{-1/2} \sum_{t=1}^n x_t \cos(2\pi \omega_k t)$$

and

$$d_s(\omega_k) = n^{-1/2} \sum_{t=1}^n x_t \sin(2\pi \omega_k t)$$

are the cosine and sine transforms, respectively, of the observed series, defined previously in (4.31) and (4.32). For example, assuming zero means for convenience, we will have

$$\begin{aligned} \mathbb{E}[d_c(\omega_k)d_c(\omega_\ell)] &= \frac{1}{4}n^{-1} \sum_{s=1}^n \sum_{t=1}^n \gamma(s-t)(e^{2\pi i \omega_k s} + e^{-2\pi i \omega_k s})(e^{2\pi i \omega_\ell t} + e^{-2\pi i \omega_\ell t}) \\ &= \frac{1}{4} [S_n(-\omega_k, \omega_\ell) + S_n(\omega_k, \omega_\ell) + S_n(\omega_\ell, \omega_k) + S_n(\omega_k, -\omega_\ell)]. \end{aligned}$$

Lemma C.1 and **Lemma C.2** imply, for $k = \ell$,

$$\begin{aligned} \mathbb{E}[d_c(\omega_k)d_c(\omega_\ell)] &= \frac{1}{4} [O(n^{-1}) + f(\omega_k) + O(n^{-1}) \\ &\quad + f(\omega_k) + O(n^{-1}) + O(n^{-1})] \\ &= \frac{1}{2}f(\omega_k) + O(n^{-1}). \end{aligned} \tag{C.16}$$

For $k \neq \ell$, all terms are $O(n^{-1})$. Hence, we have

$$\mathbb{E}[d_c(\omega_k)d_c(\omega_\ell)] = \begin{cases} \frac{1}{2}f(\omega_k) + O(n^{-1}), & k = \ell \\ O(n^{-1}), & k \neq \ell. \end{cases} \tag{C.17}$$

A similar argument gives

$$\mathbb{E}[d_s(\omega_k)d_s(\omega_\ell)] = \begin{cases} \frac{1}{2}f(\omega_k) + O(n^{-1}), & k = \ell, \\ O(n^{-1}), & k \neq \ell \end{cases} \tag{C.18}$$

and we also have $\mathbb{E}[d_s(\omega_k)d_c(\omega_\ell)] = O(n^{-1})$ for all k, ℓ . We may summarize the results of **Lemma C.1–Lemma C.3** as follows.

Theorem C.4 *For a stationary mean zero process with autocovariance function satisfying (C.9) and frequencies $\omega_{k:n}$, such that $|\omega_{k:n} - \omega| < 1/n$, are close to some target frequency ω , the cosine and sine transforms (4.31) and (4.32) are approximately uncorrelated with variances equal to $(1/2)f(\omega)$, and the error in the approximation can be uniformly bounded by $\pi\theta L/n$.*

Now, consider estimating the spectrum in a neighborhood of some target frequency ω , using the periodogram estimator

$$I(\omega_{k:n}) = |d(\omega_{k:n})|^2 = d_c^2(\omega_{k:n}) + d_s^2(\omega_{k:n}),$$

where we take $|\omega_{k:n} - \omega| \leq n^{-1}$ for each n . In case the series x_t is Gaussian with zero mean,

$$\begin{pmatrix} d_c(\omega_{k:n}) \\ d_s(\omega_{k:n}) \end{pmatrix} \xrightarrow{d} N \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \frac{1}{2} \begin{pmatrix} f(\omega) & 0 \\ 0 & f(\omega) \end{pmatrix} \right\},$$

and we have that

$$\frac{2 I(\omega_{k:n})}{f(\omega)} \xrightarrow{d} \chi_2^2,$$

where χ_v^2 denotes a chi-squared random variable with v degrees of freedom, as usual. Unfortunately, the distribution does not become more concentrated as $n \rightarrow \infty$, because the variance of the periodogram estimator does not go to zero.

We develop a fix for the deficiencies mentioned above by considering the average of the periodogram over a set of frequencies in the neighborhood of ω . For example, we can always find a set of $L = 2m + 1$ frequencies of the form $\{\omega_{j:n} + k/n; k = 0, \pm 1, \pm 2, \dots, m\}$, for which

$$f(\omega_{j:n} + k/n) = f(\omega) + O(Ln^{-1})$$

by Lemma C.3. As n increases, the values of the separate frequencies change.

Now, we can consider the smoothed periodogram estimator, $\hat{f}(\omega)$, given in (4.64); this case includes the averaged periodogram, $\bar{f}(\omega)$. First, we note that (C.9), $\theta = \sum_{h=-\infty}^{\infty} |h| |\gamma(h)| < \infty$, is a crucial condition in the estimation of spectra. In investigating local averages of the periodogram, we will require a condition on the rate of (C.9), namely

$$\sum_{h=-n}^n |h| |\gamma(h)| = O(n^{-1/2}). \quad (\text{C.19})$$

One can show that a sufficient condition for (C.19) is that the time series is the linear process given by,

$$x_t = \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}, \quad \sum_{j=0}^{\infty} \sqrt{j} |\psi_j| < \infty \quad (\text{C.20})$$

where $w_t \sim \text{iid}(0, \sigma_w^2)$ and w_t has finite fourth moment,

$$E(w_t^4) = \eta \sigma_w^4 < \infty.$$

We leave it to the reader (see Problem 4.40 for more details) to show (C.20) implies (C.19). If $w_t \sim \text{wn}(0, \sigma_w^2)$, then (C.20) implies (C.19), but we will require the noise to be iid in the following lemma.

Lemma C.4 Suppose x_t is the linear process given by (C.20), and let $I(\omega_j)$ be the periodogram of the data $\{x_1, \dots, x_n\}$. Then

$$\text{cov}(I(\omega_j), I(\omega_k)) = \begin{cases} 2f^2(\omega_j) + o(1) & \omega_j = \omega_k = 0, 1/2 \\ f^2(\omega_j) + o(1) & \omega_j = \omega_k \neq 0, 1/2 \\ O(n^{-1}) & \omega_j \neq \omega_k. \end{cases}$$

The proof of Lemma C.4 is straightforward but tedious, and details may be found in Fuller (1976, Theorem 7.2.1) or in Brockwell and Davis (1991, Theorem 10.3.2). For demonstration purposes, we present the proof of the lemma for the pure white noise case; i.e., $x_t = w_t$, in which case $f(\omega) \equiv \sigma_w^2$. By definition, the periodogram in this case is

$$I(\omega_j) = n^{-1} \sum_{s=1}^n \sum_{t=1}^n w_s w_t e^{2\pi i \omega_j(t-s)},$$

where $\omega_j = j/n$, and hence

$$\mathbb{E}\{I(\omega_j)I(\omega_k)\} = n^{-2} \sum_{s=1}^n \sum_{t=1}^n \sum_{u=1}^n \sum_{v=1}^n \mathbb{E}(w_s w_t w_u w_v) e^{2\pi i \omega_j(t-s)} e^{2\pi i \omega_k(u-v)}.$$

Now when all the subscripts match, $\mathbb{E}(w_s w_t w_u w_v) = \eta \sigma_w^4$, when the subscripts match in pairs (e.g., $s = t \neq u = v$), $\mathbb{E}(w_s w_t w_u w_v) = \sigma_w^4$, otherwise, $\mathbb{E}(w_s w_t w_u w_v) = 0$. Thus,

$$\mathbb{E}\{I(\omega_j)I(\omega_k)\} = n^{-1}(\eta - 3)\sigma_w^4 + \sigma_w^4 \left(1 + n^{-2}[A(\omega_j + \omega_k) + A(\omega_k - \omega_j)]\right),$$

where

$$A(\lambda) = \left| \sum_{t=1}^n e^{2\pi i \lambda t} \right|^2.$$

Noting that $\mathbb{E}I(\omega_j) = n^{-1} \sum_{t=1}^n \mathbb{E}(w_t^2) = \sigma_w^2$, we have

$$\begin{aligned} \text{cov}\{I(\omega_j), I(\omega_k)\} &= \mathbb{E}\{I(\omega_j)I(\omega_k)\} - \sigma_w^4 \\ &= n^{-1}(\eta - 3)\sigma_w^4 + n^{-2}\sigma_w^4[A(\omega_j + \omega_k) + A(\omega_k - \omega_j)]. \end{aligned}$$

Thus we conclude that

$$\begin{aligned} \text{var}\{I(\omega_j)\} &= n^{-1}(\eta - 3)\sigma_w^4 + \sigma_w^4 && \text{for } \omega_j \neq 0, 1/2 \\ \text{var}\{I(\omega_j)\} &= n^{-1}(\eta - 3)\sigma_w^4 + 2\sigma_w^4 && \text{for } \omega_j = 0, 1/2 \\ \text{cov}\{I(\omega_j), I(\omega_k)\} &= n^{-1}(\eta - 3)\sigma_w^4 && \text{for } \omega_j \neq \omega_k, \end{aligned}$$

which establishes the result in this case. We also note that if w_t is Gaussian, then $\eta = 3$ and the periodogram ordinates are independent. Using [Lemma C.4](#), we may establish the following fundamental result.

Theorem C.5 Suppose x_t is the linear process given by [\(C.20\)](#). Then, with $\hat{f}(\omega)$ defined in [\(4.64\)](#) and corresponding conditions on the weights h_k , we have, as $n \rightarrow \infty$,

- (i) $\mathbb{E}(\hat{f}(\omega)) \rightarrow f(\omega)$
- (ii) $\left(\sum_{k=-m}^m h_k^2\right)^{-1} \text{cov}(\hat{f}(\omega), \hat{f}(\lambda)) \rightarrow f^2(\omega) \quad \text{for } \omega = \lambda \neq 0, 1/2.$

In (ii), replace $f^2(\omega)$ by 0 if $\omega \neq \lambda$ and by $2f^2(\omega)$ if $\omega = \lambda = 0$ or $1/2$.

Proof: (i): First, recall [\(4.36\)](#)

$$\mathbb{E}[I(\omega_{j:n})] = \sum_{h=-(n-1)}^{n-1} \left(\frac{n-|h|}{n}\right) \gamma(h) e^{-2\pi i \omega_{j:n} h} \stackrel{\text{def}}{=} f_n(\omega_{j:n}).$$

But since $f_n(\omega_{j:n}) \rightarrow f(\omega)$ uniformly, and $|f(\omega_{j:n}) - f(\omega_{j:n} + k/n)| \rightarrow 0$ by the continuity of f , we have

$$\begin{aligned}\text{E}\hat{f}(\omega) &= \sum_{k=-m}^m h_k \text{EI}(\omega_{j:n} + k/n) = \sum_{k=-m}^m h_k f_n(\omega_{j:n} + k/n) \\ &= \sum_{k=-m}^m h_k [f(\omega) + o(1)] \rightarrow f(\omega),\end{aligned}$$

because $\sum_{k=-m}^m h_k = 1$.

(ii): First, suppose we have $\omega_{j:n} \rightarrow \omega_1$ and $\omega_{\ell:n} \rightarrow \omega_2$, and $\omega_1 \neq \omega_2$. Then, for n large enough to separate the bands, using [Lemma C.4](#), we have

$$\begin{aligned}|\text{cov}(\hat{f}(\omega_1), \hat{f}(\omega_2))| &= \left| \sum_{|k| \leq m} \sum_{|r| \leq m} h_k h_r \text{cov}[I(\omega_{j:n} + k/n), I(\omega_{\ell:n} + r/n)] \right| \\ &= \left| \sum_{|k| \leq m} \sum_{|r| \leq m} h_k h_r O(n^{-1}) \right| \\ &\leq \frac{c}{n} \left(\sum_{|k| \leq m} h_k \right)^2 \quad (\text{where } c \text{ is a constant}) \\ &\leq \frac{cL}{n} \left(\sum_{|k| \leq m} h_k^2 \right),\end{aligned}$$

which establishes (ii) for the case of different frequencies. The case of the same frequencies, i.e., $\omega = \lambda$, is established in a similar manner to the above arguments. \square

[Theorem C.5](#) justifies the distributional properties used throughout [Section 4.4](#) and [Chapter 7](#). We may extend the results of this section to vector series of the form $x_t = (x_{t1}, \dots, x_{tp})'$, when the cross-spectrum is given by

$$f_{ij}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{ij}(h) e^{-2\pi i \omega h} = c_{ij}(\omega) - i q_{ij}(\omega), \quad (\text{C.21})$$

where

$$c_{ij}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{ij}(h) \cos(2\pi \omega h) \quad (\text{C.22})$$

and

$$q_{ij}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{ij}(h) \sin(2\pi \omega h) \quad (\text{C.23})$$

denote the cospectrum and quadspectrum, respectively. We denote the DFT of the series x_{tj} by

$$\begin{aligned} d_j(\omega_k) &= n^{-1/2} \sum_{t=1}^n x_{tj} e^{-2\pi i \omega_k t} \\ &= d_{cj}(\omega_k) - i d_{sj}(\omega_k), \end{aligned}$$

where d_{cj} and d_{sj} are the cosine and sine transforms of x_{tj} , for $j = 1, 2, \dots, p$. We bound the covariance structure as before and summarize the results as follows.

Theorem C.6 *The covariance structure of the multivariate cosine and sine transforms, subject to*

$$\theta_{ij} = \sum_{h=-\infty}^{\infty} |h| |\gamma_{ij}(h)| < \infty, \quad (\text{C.24})$$

is given by

$$\mathbb{E}[d_{ci}(\omega_k) d_{cj}(\omega_\ell)] = \begin{cases} \frac{1}{2} c_{ij}(\omega_k) + O(n^{-1}), & k = \ell \\ O(n^{-1}), & k \neq \ell. \end{cases} \quad (\text{C.25})$$

$$\mathbb{E}[d_{ci}(\omega_k) d_{sj}(\omega_\ell)] = \begin{cases} -\frac{1}{2} q_{ij}(\omega_k) + O(n^{-1}), & k = \ell \\ O(n^{-1}), & k \neq \ell \end{cases} \quad (\text{C.26})$$

$$\mathbb{E}[d_{si}(\omega_k) d_{cj}(\omega_\ell)] = \begin{cases} \frac{1}{2} q_{ij}(\omega_k) + O(n^{-1}), & k = \ell \\ O(n^{-1}), & k \neq \ell \end{cases} \quad (\text{C.27})$$

$$\mathbb{E}[d_{si}(\omega_k) d_{sj}(\omega_\ell)] = \begin{cases} \frac{1}{2} c_{ij}(\omega_k) + O(n^{-1}), & k = \ell \\ O(n^{-1}), & k \neq \ell. \end{cases} \quad (\text{C.28})$$

Proof: We define

$$S_n^{ij}(\omega_k, \omega_\ell) = \sum_{s=1}^n \sum_{t=1}^n \gamma_{ij}(s-t) e^{-2\pi i \omega_k s} e^{2\pi i \omega_\ell t}. \quad (\text{C.29})$$

Then, we may verify the theorem with manipulations like

$$\begin{aligned} \mathbb{E}[d_{ci}(\omega_k) d_{sj}(\omega_k)] &= \frac{1}{4i} \sum_{s=1}^n \sum_{t=1}^n \gamma_{ij}(s-t) (e^{2\pi i \omega_k s} + e^{-2\pi i \omega_k s})(e^{2\pi i \omega_k t} - e^{-2\pi i \omega_k t}) \\ &= \frac{1}{4i} \left[S_n^{ij}(-\omega_k, \omega_k) + S_n^{ij}(\omega_k, \omega_k) - S_n^{ij}(\omega_k, \omega_k) - S_n^{ij}(\omega_k, -\omega_k) \right] \\ &= \frac{1}{4i} \left[c_{ij}(\omega_k) - iq_{ij}(\omega_k) - (c_{ij}(\omega_k) + iq_{ij}(\omega_k)) + O(n^{-1}) \right] \\ &= -\frac{1}{2} q_{ij}(\omega_k) + O(n^{-1}), \end{aligned}$$

where we have used the fact that the properties given in Lemma C.1–Lemma C.3 can be verified for the cross-spectral density functions $f_{ij}(\omega)$, $i, j = 1, \dots, p$. \square

Now, if the underlying multivariate time series x_t is a normal process, it is clear that the DFTs will be jointly normal and we may define the vector DFT, $d(\omega_k) = (d_1(\omega_k), \dots, d_p(\omega_k))'$ as

$$d(\omega_k) = n^{-1/2} \sum_{t=1}^n x_t e^{-2\pi i \omega_k t} = d_c(\omega_k) - i d_s(\omega_k), \quad (\text{C.30})$$

where

$$d_c(\omega_k) = n^{-1/2} \sum_{t=1}^n x_t \cos(2\pi \omega_k t) \quad (\text{C.31})$$

and

$$d_s(\omega_k) = n^{-1/2} \sum_{t=1}^n x_t \sin(2\pi \omega_k t) \quad (\text{C.32})$$

are the cosine and sine transforms, respectively, of the observed vector series x_t . Then, constructing the vector of real and imaginary parts $(d'_c(\omega_k), d'_s(\omega_k))'$, we may note it has mean zero and $2p \times 2p$ covariance matrix

$$\Sigma(\omega_k) = \frac{1}{2} \begin{pmatrix} C(\omega_k) & -Q(\omega_k) \\ Q(\omega_k) & C(\omega_k) \end{pmatrix} \quad (\text{C.33})$$

to order n^{-1} as long as $\omega_k - \omega = O(n^{-1})$. We have introduced the $p \times p$ matrices $C(\omega_k) = \{c_{ij}(\omega_k)\}$ and $Q = \{q_{ij}(\omega_k)\}$. The complex random variable $d(\omega_k)$ has covariance

$$\begin{aligned} S(\omega_k) &= E[d(\omega_k)d^*(\omega_k)] \\ &= E\left[(d_c(\omega_k) - i d_s(\omega_k))(d_c(\omega_k) - i d_s(\omega_k))^*\right] \\ &= E[d_c(\omega_k)d_c(\omega_k)'] + E[d_s(\omega_k)d_s(\omega_k)'] \\ &\quad - i(E[d_s(\omega_k)d_c(\omega_k)'] - E[d_c(\omega_k)d_s(\omega_k)']) \\ &= C(\omega_k) - iQ(\omega_k). \end{aligned} \quad (\text{C.34})$$

If the process x_t has a multivariate normal distribution, the complex vector $d(\omega_k)$ has approximately the *complex multivariate normal distribution* with mean zero and covariance matrix $S(\omega_k) = C(\omega_k) - iQ(\omega_k)$ if the real and imaginary parts have the covariance structure as specified above. In the next section, we work further with this distribution and show how it adapts to the real case. If we wish to estimate the spectral matrix $S(\omega)$, it is natural to take a band of frequencies of the form $\omega_{k:n} + \ell/n$, for $\ell = -m, \dots, m$ as before, so that the estimator becomes (4.98) of Section 4.5. A discussion of further properties of the multivariate complex normal distribution is deferred.

It is also of interest to develop a large sample theory for cases in which the underlying distribution is not necessarily normal. If x_t is not necessarily a normal process, some additional conditions are needed to get asymptotic normality. In particular, introduce the notion of a *generalized linear process*

$$y_t = \sum_{r=-\infty}^{\infty} A_r w_{t-r}, \quad (\text{C.35})$$

where w_t is a $p \times 1$ vector white noise process with $p \times p$ covariance $E[w_t w_t'] = G$ and the $p \times p$ matrices of filter coefficients A_t satisfy

$$\sum_{t=-\infty}^{\infty} \text{tr}\{A_t A_t'\} = \sum_{t=-\infty}^{\infty} \|A_t\|^2 < \infty. \quad (\text{C.36})$$

In particular, stable vector ARMA processes satisfy these conditions. For generalized linear processes, we state the following general result from Hannan (1970, p.224).

Theorem C.7 *If x_t is generated by a generalized linear process with a continuous spectrum that is not zero at ω and $\omega_{k,n} + \ell/n$ are a set of frequencies within L/n of ω , the joint density of the cosine and sine transforms (C.31) and (C.32) converges to that of L independent $2p \times 1$ normal vectors with covariance matrix $\Sigma(\omega)$ with structure given by (C.33). At $\omega = 0$ or $\omega = 1/2$, the distribution is real with covariance matrix $2\Sigma(\omega)$.*

The above result provides the basis for inference involving the Fourier transforms of stationary series because it justifies approximations to the likelihood function based on multivariate normal theory. We make extensive use of this result in Chapter 7, but will still need a simple form to justify the distributional result for the sample coherence given in (4.104). The next section gives an elementary introduction to the complex normal distribution.

C.3 The Complex Multivariate Normal Distribution

The multivariate normal distribution will be the fundamental tool for expressing the likelihood function and determining approximate maximum likelihood estimators and their large sample probability distributions. A detailed treatment of the multivariate normal distribution can be found in standard texts such as Anderson (1984). We will use the multivariate normal distribution of the $p \times 1$ vector $x = (x_1, x_2, \dots, x_p)'$, as defined by its density function

$$p(x) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(x - \mu)' \Sigma^{-1} (x - \mu)\right\}, \quad (\text{C.37})$$

which has mean vector $E[x] = \mu = (\mu_1, \dots, \mu_p)'$ and covariance matrix

$$\Sigma = E[(x - \mu)(x - \mu)']. \quad (\text{C.38})$$

We use the notation $x \sim N_p(\mu, \Sigma)$ for densities of the form (C.37) and note that linearly transformed multivariate normal variables of the form $y = Ax$, with A a $q \times p$ matrix $q \leq p$, will also be multivariate normal with distribution

$$y \sim N_q(A\mu, A\Sigma A'). \quad (\text{C.39})$$

Often, the partitioned multivariate normal, based on the vector $x = (x'_1, x'_2)'$, split into two $p_1 \times 1$ and $p_2 \times 1$ components x_1 and x_2 , respectively, will be used where $p = p_1 + p_2$. If the mean vector $\mu = (\mu'_1, \mu'_2)'$ and covariance matrices

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \quad (\text{C.40})$$

are also compatibly partitioned, the marginal distribution of any subset of components is multivariate normal, say,

$$x_1 \sim N_{p_1}\{\mu_1, \Sigma_{11}\},$$

and that the conditional distribution x_2 given x_1 is normal with mean

$$\mathbb{E}[x_2 | x_1] = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1 - \mu_1) \quad (\text{C.41})$$

and conditional covariance

$$\text{cov}[x_2 | x_1] = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \quad (\text{C.42})$$

In the previous section, the real and imaginary parts of the DFT had a partitioned covariance matrix as given in (C.33), and we use this result to say the complex $p \times 1$ vector

$$z = x_1 - ix_2 \quad (\text{C.43})$$

has a *complex multivariate normal distribution*, with mean vector $\mu_z = \mu_1 - i\mu_2$ and $p \times p$ covariance matrix

$$\Sigma_z = C - iQ \quad (\text{C.44})$$

if the real multivariate $2p \times 1$ normal vector $x = (x'_1, x'_2)'$ has a real multivariate normal distribution with mean vector $\mu = (\mu'_1, \mu'_2)'$ and covariance matrix

$$\Sigma = \frac{1}{2} \begin{pmatrix} C & -Q \\ Q & C \end{pmatrix}. \quad (\text{C.45})$$

The restrictions $C' = C$ and $Q' = -Q$ are necessary for the matrix Σ to be a covariance matrix, and these conditions then imply $\Sigma_z = \Sigma_z^*$ is Hermitian. The probability density function of the complex multivariate normal vector z can be expressed in the concise form

$$p_z(z) = \pi^{-p} |\Sigma_z|^{-1} \exp\{-(z - \mu_z)^* \Sigma_z^{-1} (z - \mu_z)\}, \quad (\text{C.46})$$

and this is the form that we will often use in the likelihood. The result follows from showing that $p_x(x_1, x_2) = p_z(z)$ exactly, using the fact that the quadratic and Hermitian forms in the exponent are equal and that $|\Sigma_x| = |\Sigma_z|^2$. The second assertion follows directly from the fact that the matrix Σ_x has repeated eigenvalues, $\lambda_1, \lambda_2, \dots, \lambda_p$ corresponding to eigenvectors $(\alpha'_1, \alpha'_2)'$ and the same set, $\lambda_1, \lambda_2, \dots, \lambda_p$ corresponding to $(\alpha'_2, -\alpha'_1)'$. Hence

$$|\Sigma_x| = \prod_{i=1}^p \lambda_i^2 = |\Sigma_z|^2.$$

For further material relating to the complex multivariate normal distribution, see Goodman (1963), Giri (1965), or Khatri (1965).

Example C.2 A Complex Normal Random Variable

To fix ideas, consider a very simple complex random variable

$$z = \Re(z) - i\Im(z) = z_1 - iz_2,$$

where $z_1 \sim N(0, \frac{1}{2}\sigma^2)$ independent of $z_2 \sim N(0, \frac{1}{2}\sigma^2)$. Then the joint density of (z_1, z_2) is

$$p(z_1, z_2) \propto \sigma^{-1} \exp\left(-\frac{z_1^2}{\sigma^2}\right) \times \sigma^{-1} \exp\left(-\frac{z_2^2}{\sigma^2}\right) = \sigma^{-2} \exp\left\{-\left(\frac{z_1^2 + z_2^2}{\sigma^2}\right)\right\}.$$

More succinctly, we write $z \sim N_c(0, \sigma^2)$, and

$$p(z) \propto \sigma^{-2} \exp\left(-\frac{z^* z}{\sigma^2}\right).$$

In Fourier analysis, z_1 would be the cosine transform of the data at a fundamental frequency (excluding the end points) and z_2 the corresponding sine transform. If the process is Gaussian, z_1 and z_2 are independent normals with zero means and variances that are half of the spectral density at the particular frequency. Consequently, the definition of the complex normal distribution is natural in the context of spectral analysis.

Example C.3 A Bivariate Complex Normal Distribution

Consider the joint distribution of the complex random variables $u_1 = x_1 - ix_2$ and $u_2 = y_1 - iy_2$, where the partitioned vector $(x_1, x_2, y_1, y_2)'$ has a real multivariate normal distribution with mean $(0, 0, 0, 0)'$ and covariance matrix

$$\Sigma = \frac{1}{2} \begin{pmatrix} c_{xx} & 0 & c_{xy} & -q_{xy} \\ 0 & c_{xx} & q_{xy} & c_{xy} \\ c_{xy} & q_{xy} & c_{yy} & 0 \\ -q_{xy} & c_{yx} & 0 & c_{yy} \end{pmatrix}. \quad (\text{C.47})$$

Now, consider the conditional distribution of $y = (y_1, y_2)'$, given $x = (x_1, x_2)'$. Using (C.41), we obtain

$$E(y | x) = \begin{pmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad (\text{C.48})$$

where

$$(b_1, b_2) = \left(\frac{c_{yx}}{c_{xx}}, \frac{q_{yx}}{c_{xx}} \right). \quad (\text{C.49})$$

It is natural to identify the cross-spectrum

$$f_{xy} = c_{xy} - iq_{xy}, \quad (\text{C.50})$$

so that the complex variable identified with the pair is just

$$b = b_1 - ib_2 = \frac{c_{yx} - iq_{yx}}{c_{xx}} = \frac{f_{yx}}{f_{xx}},$$

and we identify it as the complex regression coefficient. The conditional covariance follows from (C.42) and simplifies to

$$\text{cov}(y \mid x) = \frac{1}{2} f_{y \cdot x} I_2, \quad (\text{C.51})$$

where I_2 denotes the 2×2 identity matrix and

$$f_{y \cdot x} = c_{yy} - \frac{c_{xy}^2 + q_{xy}^2}{c_{xx}} = f_{yy} - \frac{|f_{xy}|^2}{f_{xx}} \quad (\text{C.52})$$

Example C.3 leads to an approach for justifying the distributional results for the function coherence given in (4.104). That equation suggests that the result can be derived using the regression results that lead to the F-statistics in [Section 2.1](#). Suppose that we consider L values of the sine and cosine transforms of the input x_t and output y_t , which we will denote by $d_{x,c}(\omega_k + \ell/n)$, $d_{x,s}(\omega_k + \ell/n)$, $d_{y,c}(\omega_k + \ell/n)$, $d_{y,s}(\omega_k + \ell/n)$, sampled at $L = 2m + 1$ frequencies, $\ell = -m, \dots, m$, in the neighborhood of some target frequency ω . Suppose these cosine and sine transforms are re-indexed and denoted by $d_{x,cj}$, $d_{x,sj}$, $d_{y,cj}$, $d_{y,sj}$, for $j = 1, 2, \dots, L$, producing $2L$ real random variables with a large sample normal distribution that have limiting covariance matrices of the form (C.47) for each j . Then, the conditional normal distribution of the 2×1 vector $d_{y,cj}$, $d_{y,sj}$ given $d_{x,cj}$, $d_{x,sj}$, given in [Example C.3](#), shows that we may write, approximately, the regression model

$$\begin{pmatrix} d_{y,cj} \\ d_{y,sj} \end{pmatrix} = \begin{pmatrix} d_{x,cj} & -d_{x,sj} \\ d_{x,sj} & d_{x,cj} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} V_{cj} \\ V_{sj} \end{pmatrix},$$

where V_{cj} , V_{sj} are approximately uncorrelated with approximate variances

$$\text{E}[V_{cj}^2] = \text{E}[V_{sj}^2] = (1/2)f_{y \cdot x}.$$

Now, construct, by stacking, the $2L \times 1$ vectors $y_c = (d_{y,c1}, \dots, d_{y,cL})'$, $y_s = (d_{y,s1}, \dots, d_{y,sL})'$, $x_c = (d_{x,c1}, \dots, d_{x,cL})'$ and $x_s = (d_{x,s1}, \dots, d_{x,sL})'$, and rewrite the regression model as

$$\begin{pmatrix} y_c \\ y_s \end{pmatrix} = \begin{pmatrix} x_c & -x_s \\ x_s & x_c \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} + \begin{pmatrix} v_c \\ v_s \end{pmatrix}$$

where v_c and v_s are the error stacks. Finally, write the overall model as the regression model in Chapter 2, namely,

$$y = Zb + v,$$

making the obvious identifications in the previous equation. Conditional on Z , the model becomes exactly the regression model considered in [Chapter 2](#) where there are $q = 2$ regression coefficients and $2L$ observations in the observation vector y . To test the hypothesis of no regression for that model, we use an F-Statistic that depends on the difference between the residual sum of squares for the full model, say,

$$SSE = y'y - y'Z(Z'Z)^{-1}Z'y \quad (\text{C.53})$$

and the residual sum of squares for the reduced model, $SSE_0 = y'y$. Then,

$$F_{2,2L-2} = (L-1) \frac{SSE_0 - SSE}{SSE} \quad (\text{C.54})$$

has the F-distribution with 2 and $2L - 2$ degrees of freedom. Also, it follows by substitution for y that

$$SSE_0 = y'y = y'_c y_c + y'_s y_s = \sum_{j=1}^L (d_{y,cj}^2 + d_{y,sj}^2) = L \hat{f}_y(\omega),$$

which is just the sample spectrum of the output series. Similarly,

$$Z'Z = \begin{pmatrix} L\hat{f}_x & 0 \\ 0 & L\hat{f}_x \end{pmatrix}$$

and

$$\begin{aligned} Z'y &= \begin{pmatrix} (x'_c y_c + x'_s y_s) \\ (x'_c y_s - x'_s y_c) \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^L (d_{x,cj} d_{y,cj} + d_{x,sj} d_{y,sj}) \\ \sum_{j=1}^L (d_{x,cj} d_{y,sj} - d_{x,sj} d_{y,cj}) \end{pmatrix} \\ &= \begin{pmatrix} L\hat{c}_{yx} \\ L\hat{q}_{yx} \end{pmatrix}. \end{aligned}$$

together imply that

$$y'Z(Z'Z)^{-1}Z'y = L |\hat{f}_{xy}|^2 / \hat{f}_x.$$

Substituting into [\(C.54\)](#) gives

$$F_{2,2L-2} = (L-1) \frac{|\hat{f}_{xy}|^2 / \hat{f}_x}{\left(\hat{f}_y - |\hat{f}_{xy}|^2 / \hat{f}_x \right)},$$

which converts directly into the F-statistic [\(4.104\)](#), using the sample coherence defined in [\(4.103\)](#).

C.4 Integration

In [Chapter 4](#) and in this appendix, we use Riemann–Stieltjes integration and stochastic integration. We now give a cursory introduction to these concepts for readers unfamiliar with the techniques.

C.4.1 Riemann–Stieltjes Integration

Rather than work in complete generality, we focus on the meaning of [\(4.14\)](#),

$$\gamma(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF(\omega).$$

Here, we are concerned with the integration of a bounded, continuous (complex-valued) function $g(\omega) = e^{2\pi i \omega h}$ with respect to a monotonically increasing, right continuous (real-valued) function $F(\omega)$.

Let $\mathcal{Q} = \{-\frac{1}{2} = \omega_0, \omega_1, \dots, \omega_n = \frac{1}{2}\}$ be a partition of the interval, and define the sum

$$S_{\mathcal{Q}}(g, F) = \sum_{j=1}^n g(u_j)[F(\omega_j) - F(\omega_{j-1})] \quad (\text{C.55})$$

where $u_j \in [\omega_{j-1}, \omega_j]$. In our case, there is a unique number, say $\mathcal{I}(g, F)$ such that for any $\epsilon > 0$, there is a $\delta > 0$ for which

$$|S_{\mathcal{Q}}(g, F) - \mathcal{I}(g, F)| < \epsilon$$

for any partition \mathcal{Q} with $\max_j |\omega_j - \omega_{j-1}| < \delta$ and any $u_j \in [\omega_{j-1}, \omega_j]$ for $j = 1, \dots, n$. In this case, we define

$$\mathcal{I}(g, F) = \int_{-\frac{1}{2}}^{\frac{1}{2}} g(\omega) dF(\omega). \quad (\text{C.56})$$

In the *absolutely continuous case*, such as in [Property 4.2](#), $dF(\omega) = f(\omega)d\omega$ and, as stated in the property,

$$\gamma(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF(\omega) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} f(\omega) d\omega.$$

Another case that we discussed was the *discrete case* such as in [Example 4.4](#) where the spectral distribution $F(\omega)$ makes jumps at specific values of ω . First, consider the case where $F(\omega)$ has only one jump of size $c > 0$ at $\omega^* \in (-\frac{1}{2}, \frac{1}{2})$, so that $F(\omega) = 0$ if $\omega < \omega^*$ and $F(\omega) = c$ if $\omega \geq \omega^*$. Then considering $S_{\mathcal{Q}}(g, F)$ in [\(C.55\)](#), note that $F(\omega_j) - F(\omega_{j-1}) = 0$ for all intervals that do not include ω^* . Now suppose in some k th interval of the partition, $\omega^* \in (\omega_{k-1}, \omega_k]$ for a $k \in \{1, \dots, n\}$. Then

$$S_{\mathcal{Q}}(g, F) = \sum_{j=1}^n g(u_j)[F(\omega_j) - F(\omega_{j-1})] = g(u_k)c,$$

where $u_k \in [\omega_{k-1}, \omega_k]$. Thus,

$$|S_{\mathcal{Q}}(g, F) - g(\omega^*) c| = c |g(u_k) - g(\omega^*)|.$$

Since g is continuous, given $\epsilon > 0$, there is a $\delta > 0$ such that $|g(u_k) - g(\omega^*)| < \epsilon/c$ when $|u_k - \omega^*| < \delta$. Hence, for any partition \mathcal{Q} with $\max_j |\omega_j - \omega_{j-1}| < \delta$, we have $|S_{\mathcal{Q}}(g, F) - g(\omega^*) c| < \epsilon$, and consequently,

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} g(\omega) dF(\omega) = g(\omega^*) c.$$

This result may be extended in an obvious way to the case where F makes jumps at more than one value as was the case in [Example 4.4](#).

Example C.4 Complex Harmonic Process

Recall [\(4.4\)](#) where we considered a mix of periodic components. In that example, the process was real, but it is possible to consider a complex-valued process in a similar way. In this case, we define

$$x_t = \sum_{j=1}^q Z_j e^{2\pi i t \omega_j}, \quad -\frac{1}{2} < \omega_1 < \dots < \omega_q < \frac{1}{2}, \quad (\text{C.57})$$

where the Z_j are uncorrelated complex-valued random variables such that $E[Z_j] = 0$ and $E[|Z_j|^2] = \sigma_j^2 > 0$. As discussed in [Example 4.9](#), the case where x_t is real-valued is a special case of [\(C.57\)](#). Extending [Example 4.4](#) to the case of [\(C.57\)](#), we have

$$F(\omega) = \begin{cases} 0 & -\frac{1}{2} \leq \omega < \omega_1, \\ \sigma_1^2 & \omega_1 \leq \omega < \omega_2, \\ \sigma_1^2 + \sigma_2^2 & \omega_2 \leq \omega < \omega_3, \\ \sigma_1^2 + \sigma_2^2 + \sigma_3^2 & \omega_3 \leq \omega < \omega_4, \\ \vdots & \vdots \\ \sigma_1^2 + \sigma_2^2 + \dots + \sigma_q^2 & \omega_q \leq \omega \leq \frac{1}{2}. \end{cases} \quad (\text{C.58})$$

Thus, for the process in this example,

$$\gamma_x(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF(\omega) = \sum_{j=1}^q \sigma_j^2 e^{2\pi i h \omega_j}.$$

Note that $\gamma_x(h)$ is complex, but satisfies the properties of an autocovariance function: (i) $\gamma_x(h)$ is a Hermitian function, $\gamma_x(h) = \gamma_x^*(-h)$; (ii) $0 \leq |\gamma_x(h)| \leq \gamma_x(0)$, and (iii) $\gamma_x(h)$ is non-negative definite. As in the real case, the total variance of the process is the sum of the variances of the individual components, $\text{var}(x_t) = \gamma_x(0) = \sum_{j=1}^q \sigma_j^2$.

C.4.2 Stochastic Integration

We first used stochastic integration in [Example 4.9](#), although it was not necessary for that particular example. There is an analogy of stochastic integration to Riemann-Stieltjes integration defined in the previous subsection, however, we will have to deal with convergence of random processes rather than convergence of numbers. We focus on the case of interest to us; namely the stochastic integral in [Theorem C.2](#),

$$x_t = \int_{-\frac{1}{2}}^{\frac{1}{2}} g(\omega) dZ(\omega),$$

where $Z(\omega)$ is a complex-valued *orthogonal increment process* and $g(\omega) = e^{2\pi i \omega t}$. For $\{Z(\omega); \omega \in [-\frac{1}{2}, \frac{1}{2}]\}$ and $-\frac{1}{2} \leq \omega_1 < \omega_2 < \omega_3 < \omega_4 \leq \frac{1}{2}$, we have

- $Z(-\frac{1}{2}) = 0$,
- $E[Z(\omega)] = 0$,
- $\text{var}[Z(\omega)] = E[|Z(\omega)|^2] = E[Z(\omega) Z^*(\omega)] < \infty$,
- $E\{|Z(\omega_4) - Z(\omega_3)| |Z(\omega_2) - Z(\omega_1)|^*\} = 0$.

As an example, recall Brownian motion in [Definition 5.1](#).

We say $\{Z(\omega)\}$ is *mean square (m.s.) right continuous* if $E|Z(\omega+\delta) - Z(\omega)|^2 \rightarrow 0$ as $\delta \downarrow 0$. An important result is that such a process admits a spectral distribution.

Theorem C.8 *If $\{Z(\omega); \omega \in [-\frac{1}{2}, \frac{1}{2}]\}$ is an orthogonal increment process that is m.s. right continuous, then there is a unique spectral distribution function F such that*

- (1) $F(\omega) = 0$ if $\omega \leq -\frac{1}{2}$.
- (2) $F(\omega) = F(\frac{1}{2})$ if $\omega \geq \frac{1}{2}$.
- (3) $F(\omega_2) - F(\omega_1) = E|Z(\omega_2) - Z(\omega_1)|^2$ if $-\frac{1}{2} \leq \omega_1 \leq \omega_2 \leq \frac{1}{2}$.

Proof: Define $F(\omega) = E|Z(\omega)|^2$ for $\omega \in [-\frac{1}{2}, \frac{1}{2}]$, with $F(\omega) = 0$ for $\omega \leq -\frac{1}{2}$ and $F(\omega) = F(\frac{1}{2})$ for $\omega \geq \frac{1}{2}$. It is immediate from the assumptions that F is right continuous and satisfies (1)–(3). To show that F is monotonically increasing, note that for $\omega_2 \geq \omega_1$,

$$\begin{aligned} F(\omega_2) &= E|Z(\omega_2) - Z(\omega_1) + Z(\omega_1) - Z(-\frac{1}{2})|^2 \\ &= E|Z(\omega_2) - Z(\omega_1)|^2 + E|Z(\omega_1)|^2 \\ &\geq F(\omega_1), \end{aligned}$$

since $[-\frac{1}{2}, \omega_1]$ and $[\omega_1, \omega_2]$ are non-overlapping intervals. \square

Similar to the previous subsection, let $\mathcal{Q} = \{-\frac{1}{2} = \omega_0, \omega_1, \dots, \omega_n = \frac{1}{2}\}$ be a partition of the interval, and define the random sum

$$S_{\mathcal{Q}}(g, Z) = \sum_{j=1}^n g(u_j)[Z(\omega_j) - Z(\omega_{j-1})] \tag{C.59}$$

where $u_j \in [\omega_{j-1}, \omega_j]$. We emphasize the fact that $S_Q(g, Z)$ is a complex-valued random variable with mean and variance given by

$$\mathbb{E}[S_Q(g, Z)] = 0 \quad \text{and} \quad \mathbb{E}[|S_Q(g, Z)|^2] = \sum_{j=1}^n g(u_j)[F(\omega_j) - F(\omega_{j-1})]$$

where F is defined in [Theorem C.8](#). In our case, there is a unique (except on a set of probability zero) complex-valued random variable, say $\mathcal{I}(g, Z)$ such that for any $\epsilon > 0$, there is a $\delta > 0$ for which

$$\mathbb{E}|S_Q(g, Z) - \mathcal{I}(g, Z)|^2 < \epsilon$$

for any partition Q with $\Delta_Q = \max_j |\omega_j - \omega_{j-1}| < \delta$ and any $u_j \in [\omega_{j-1}, \omega_j]$ for $j = 1, \dots, n$. In this case, define

$$\mathcal{I}(g, Z) = \int_{-\frac{1}{2}}^{\frac{1}{2}} g(\omega) dZ(\omega). \quad (\text{C.60})$$

We see that the stochastic integral is the mean-square limit of the random sum as $n \rightarrow \infty$ ($\Delta_Q \rightarrow 0$).

Recalling [Example 4.9](#), as in the deterministic case, it is easy to show that, if $Z(\omega)$ is an orthogonal increment process that makes uncorrelated jumps at $-\omega_0$ and ω_0 with mean-zero and variance $\sigma^2/2$, then

$$x_t = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega t} dZ(\omega) = Z(-\omega_0) e^{-2\pi i \omega_0 t} + Z(\omega_0) e^{2\pi i \omega_0 t}.$$

In this case, the spectral distribution is (recall [Example 4.4](#))

$$F(\omega) = \begin{cases} 0 & \omega < -\omega_0, \\ \sigma^2/2 & -\omega_0 \leq \omega < \omega_0, \\ \sigma^2 & \omega \geq \omega_0, \end{cases}$$

and the autocovariance function is

$$\gamma_x(h) = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i \omega h} dF(\omega) = \frac{\sigma^2}{2} e^{-2\pi i \omega_0 h} + \frac{\sigma^2}{2} e^{2\pi i \omega_0 h} = \sigma^2 \cos(2\pi \omega_0 h).$$

C.5 Spectral Analysis as Principal Component Analysis

In [Chapter 4](#), we presented many different ways to view the spectral density. In this section, we show that the spectral density may be thought of as the approximate eigenvalues of the covariance matrix of a stationary process. Suppose $X = (x_1, \dots, x_n)$ are n values of a real, mean-zero, time series, x_t with spectral density $f_x(\omega)$. Then

$$\text{cov}(X) = \Gamma_n = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(n-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(n-1) & \gamma(n-2) & \cdots & \gamma(0) \end{bmatrix}$$

is a non-negative definite, symmetric Toeplitz matrix. Hence, there is an $n \times n$ orthogonal matrix M , such that $M'\Gamma_n M = \text{diag}(\lambda_0, \dots, \lambda_{n-1})$, where $\lambda_j \geq 0$ for $j = 0, \dots, n-1$ are the latent roots of Γ_n . In this section, we will show that, for n sufficiently large,

$$\lambda_j \approx f_x(\omega_j), \quad j = 0, 1, \dots, n-1,$$

where $\omega_j = j/n$ are the Fourier frequencies.

To start the approximation, we introduce a circulant matrix defined as

$$\Gamma_c = \begin{bmatrix} c(0) & c(1) & \cdots & c(n-2) & c(n-1) \\ c(n-1) & c(0) & \cdots & c(n-3) & c(n-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c(2) & c(3) & \cdots & c(0) & c(1) \\ c(1) & c(2) & \cdots & c(n-1) & c(0) \end{bmatrix};$$

the matrix has $c(0)$ on the diagonal, then continue to the right $c(1), c(2), \dots$, and wrap the sequence around to the first column after the last column is reached. Using direct substitution, it can be shown that the latent roots and vectors of Γ_c are

$$\lambda_j = \sum_{h=0}^{n-1} c(h) e^{-2\pi i h j / n},$$

and

$$g_j^* = \frac{1}{\sqrt{n}} \left(e^{-2\pi i 0 \frac{j}{n}}, e^{-2\pi i 1 \frac{j}{n}}, \dots, e^{-2\pi i (n-1) \frac{j}{n}} \right),$$

for $j = 0, 1, \dots, n-1$.

If Γ_c is symmetric [$c(j) = c(n-j)$], call it Γ_s and let $c(h) = c(-h)$. Noting that $e^{-2\pi i h j / n} = e^{-2\pi i (n-h) j / n}$, we have for n odd,

$$\lambda_j = \sum_{|h| \leq \frac{n-1}{2}} c(h) e^{-2\pi i h j / n} = \sum_{|h| \leq \frac{n-1}{2}} c(h) \cos(2\pi h j / n)$$

for $j = 0, 1, \dots, n-1$. If n is even, the sum would include one extra term for $j/n = 1/2$.

We see that λ_0 is a distinct root, and $\lambda_j = \lambda_{n-j}$ are repeated roots for $j = 1, \dots, \frac{n-1}{2}$. For each repeated root, we can find a pair of eigenvectors corresponding to λ_j , namely

$$v'_j = \frac{1}{\sqrt{2}} (g_j^* + g_{n-j}^*) = \frac{\sqrt{2}}{\sqrt{n}} \left(1, \cos(2\pi j / n), \dots, \cos(2\pi (n-1) j / n) \right);$$

$$u'_j = \frac{1}{\sqrt{2}} i(g_j^* - g_{n-j}^*) = \frac{\sqrt{2}}{\sqrt{n}} \left(0, \sin(2\pi j / n), \dots, \sin(2\pi (n-1) j / n) \right).$$

For λ_0 , the corresponding eigenvector is $v'_0 = g_0^* = \frac{1}{\sqrt{n}}(1, 1, \dots, 1) = \frac{\sqrt{2}}{\sqrt{n}}(\frac{1}{\sqrt{2}}, \dots, \frac{1}{\sqrt{2}})$. Now define the matrix Q as

$$Q = \begin{bmatrix} v'_0 \\ v'_1 \\ u'_1 \\ \vdots \\ v'_{\frac{n-1}{2}} \\ u'_{\frac{n-1}{2}} \end{bmatrix} = \frac{\sqrt{2}}{\sqrt{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ 1 & \cos(2\pi \frac{1}{n}) & \cdots & \cos(2\pi \frac{n-1}{n}) \\ 0 & \sin(2\pi \frac{1}{n}) & \cdots & \sin(2\pi \frac{n-1}{n}) \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \cos(2\pi \frac{n-1}{2} \frac{1}{n}) & \cdots & \cos(2\pi \frac{n-1}{2} \frac{n-1}{n}) \\ 0 & \sin(2\pi \frac{n-1}{2} \frac{1}{n}) & \cdots & \sin(2\pi \frac{n-1}{2} \frac{n-1}{n}) \end{bmatrix}. \quad (\text{C.61})$$

Thus, with $m = \frac{n-1}{2}$,

$$Q\Gamma_s Q' = \text{diag}(\lambda_0, \lambda_1, \lambda_1, \lambda_2, \lambda_2, \dots, \lambda_m, \lambda_m)$$

where $\lambda_j = \sum_{|h| \leq m} c(h) \cos(2\pi h j / n)$ for $j = 0, 1, \dots, m$.

Theorem C.9 Let Γ_n be the covariance matrix of n (odd) realizations from a stationary process $\{x_t\}$ with spectral density $f_x(\omega)$. Let Q be as defined in (C.61) and let $D_n = \text{diag}\{d_0, d_1, \dots, d_{n-1}\}$ be the diagonal matrix with entries $d_0 = f_x(0) = \sum_{-\infty}^{\infty} \gamma(h)$ and

$$d_{2j-1} = d_{2j} = f_x(\omega_j) = \sum_{-\infty}^{\infty} \gamma(h) e^{-2\pi i h j / n},$$

for $j = 1, \dots, \frac{n-1}{2}$ and $\omega_j = j/n$. Then

$$Q\Gamma_n Q - D_n \rightarrow 0 \quad \text{uniformly as } n \rightarrow \infty.$$

Proof: Although Γ_n is symmetric, it is not circulant (or the proof would be done). Let $\Gamma_{n,s}$ be the symmetric circulant matrix with elements $c(h) = \gamma(h)$, and latent roots, $\lambda_j = \sum_{|h| \leq \frac{n-1}{2}} \gamma(h) e^{-2\pi i h j / n}$. Note that

$$|\lambda_j - f_x(\omega_j)| \leq \sum_{|h| > \frac{n-1}{2}} |\gamma(h)| \rightarrow 0$$

as $n \rightarrow \infty$. Hence, we must show that $Q\Gamma_{n,s} Q' - Q\Gamma_n Q' \rightarrow 0$ as $n \rightarrow \infty$.

The ij th element of the difference of the two matrices is

$$\{\Gamma_{n,s} - \Gamma_n\}_{ij} = \begin{cases} 0 & \text{if } |i - j| \leq \frac{n-1}{2} \\ \gamma(n - |i - j|) - \gamma(|i - j|) & \text{if } |i - j| > \frac{n-1}{2} \end{cases}.$$

Put $n - m = |i - j|$, so that the second case is

$$\gamma(m) - \gamma(n - m) \quad \text{for } 1 \leq m \leq \frac{n-1}{2}.$$

Let q_j be the j th column of Q , then

$$\begin{aligned}
& |q_i'(\Gamma_{n,s} - \Gamma_n)q_j| \\
&= \left| \sum_{m=1}^{\frac{n-1}{2}} \sum_{k=1}^m q_{ik}[\gamma(m) + \gamma(n-m)]q_{j,n-m+k} + q_{i,n-m+k}[\gamma(m) + \gamma(n-m)]q_{jk} \right| \\
&= \left| \sum_{m=1}^{\frac{n-1}{2}} [\gamma(m) + \gamma(n-m)] + \sum_{k=1}^m q_{ik}q_{j,n-m+k} + q_{i,n-m+k}q_{jk} \right| \\
&\stackrel{(1)}{\leq} \frac{4}{n} \sum_{m=1}^{\frac{n-1}{2}} m|\gamma(m)| + \frac{4}{n} \sum_{m=1}^{\frac{n-1}{2}} m|\gamma(n-m)| \\
&\stackrel{(2)}{\leq} \frac{4}{n} \sum_{m=1}^{\frac{n-1}{2}} m|\gamma(m)| + \frac{4}{n} \sum_{k=\frac{n-1}{2}+1}^n \frac{n-1}{2} |\gamma(k)| \\
&\xrightarrow{n \rightarrow \infty} \underbrace{0}_{(3)} + \underbrace{0}_{(4)}.
\end{aligned}$$

Inequality (1) follows because $|q_{ij}|^2 \leq 2/n$. In the second summation of inequality (2), put $k = n - m$ and use the fact that $m \leq \frac{n-1}{2}$ in the sum. Result (3) follows from Kronecker's Lemma C.1 and (4) follows from the fact that we are summing the tail end of an absolutely summable sequence [and $(n-1)/n \sim 1$]. \square

The results of this section may be summarized as follows. If we transform the data vector, say $X = (x_1, \dots, x_n)$ by $Y = QX$, the components of Y are nearly uncorrelated with $\text{cov}(Y) \approx D_n$. The components of Y are

$$\frac{2}{\sqrt{n}} \sum_{t=1}^n x_t \cos(2\pi t j/n) \quad \text{and} \quad \frac{2}{\sqrt{n}} \sum_{t=1}^n x_t \sin(2\pi t j/n)$$

for $j = 0, 1, \dots, \frac{n-1}{2}$. If we let G be the complex matrix with columns g_j , then the complex transform $Y = G^*X$ has elements that are the DFTs,

$$y_j = \frac{1}{\sqrt{n}} \sum_{t=1}^n x_t e^{-2\pi i t j / n}$$

for $j = 0, 1, \dots, n-1$. In this case, the elements of Y are asymptotically uncorrelated complex random variables, with mean-zero and variance $f(\omega_j)$. Also, X may be recovered as $X = GY$, so that $x_t = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} y_j e^{2\pi i t j / n}$.

In this section, we focused on the case where n is odd. For the n even case, everything follows through as in the odd case, but with the addition of one more term when $\frac{n-1}{2}$ becomes $\frac{n}{2} - 1$, and with the addition of one more row in Q or G , and all in a manner that is so obvious, it would be too simple to be a good homework question.

^{C.1} Kronecker's Lemma: If $\sum_{j=0}^{\infty} |a_j| < \infty$, then $\sum_{j=0}^n \frac{j}{n} |a_j| \rightarrow 0$ as $n \rightarrow \infty$.

C.6 Parametric Spectral Estimation

In this section we prove **Property 4.7**. The basic idea of the result is that a spectral density can be approximated arbitrarily close by the spectrum of an AR(p) process.

Proof of Property 4.7. If $g(\omega) \equiv 0$, then put $p = 0$ and $\sigma_w = 0$. When $g(\omega) > 0$ over some $\omega \in [-\frac{1}{2}, \frac{1}{2}]$, let $\epsilon > 0$ and define

$$d(\omega) = \begin{cases} g^{-1}(\omega) & \text{if } g(\omega) > \epsilon/2, \\ 2/\epsilon & \text{if } g(\omega) \leq \epsilon/2, \end{cases}$$

so that $d^{-1}(\omega) = \max\{g(\omega), \epsilon/2\}$. Define $G = \max_{\omega} \{g(\omega)\}$ and let $0 < \delta < \epsilon[G(2G + \epsilon)]^{-1}$. Define the sum

$$S_n[d(\omega)] = \sum_{|j| \leq n} \langle d, e_j \rangle e_j(\omega)$$

where $e_j(\omega) = e^{2\pi i j \omega}$ and $\langle d, e_j \rangle = \int_{-\frac{1}{2}}^{\frac{1}{2}} d(\omega) e^{-2\pi i j \omega} d\omega$. Now define the Cesaro sum

$$C_m(\omega) = \frac{1}{m} \sum_{n=0}^{m-1} S_n[d(\omega)],$$

which is a cumulative average of $S_n[\cdot]$. In this case, $C_m(\omega) = \sum_{|j| \leq m} c_j e^{-2\pi i j \omega}$ where $c_j = (1 - \frac{|j|}{m}) \langle d, e_j \rangle$. The Cesaro sum converges uniformly on $[-\frac{1}{2}, \frac{1}{2}]$ for $d \in L^2$, consequently there is a finite p such that

$$\left| \sum_{|j| \leq p} c_j e^{-2\pi i j \omega} - d(\omega) \right| < \delta \quad \text{for all } \omega \in [-\frac{1}{2}, \frac{1}{2}].$$

Note that $C_p(\omega)$ is a spectral density. In fact, it is the spectral density of an MA(p) process with $\gamma(h) = c_h$ for $|h| \leq p$ and $\gamma(h) = 0$ for $|h| > p$; it is easy to check that $\gamma(h)$ defined this way is non-negative definite. Hence, the is an invertible MA(p) process, say

$$y_t = u_t + \alpha_1 u_{t-1} + \cdots + \alpha_p u_{t-p}$$

where $u_t \sim w(n(0, \sigma_u^2))$ and $\alpha(z)$ has roots outside the unit circle. Thus,

$$C_p(\omega) = \sum_{|j| \leq p} c_j e^{-2\pi i j \omega} = \sigma_u^2 |\alpha(e^{-2\pi i \omega})|^2,$$

and

$$\left| \sigma_u^2 |\alpha(e^{-2\pi i \omega})|^2 - d(\omega) \right| < \delta < \epsilon[G(2G + \epsilon)]^{-1} \stackrel{\text{def}}{=} \epsilon^*.$$

Now define $f_x(\omega) = [\sigma_u^2 |\alpha(e^{-2\pi i \omega})|^2]^{-1}$. We will show that $|f_x(\omega) - g(\omega)| < \epsilon$, in which case the result follows with $\alpha_1, \dots, \alpha_p$ being the required AR(p) coefficients, and $\sigma_w^2 = \sigma_u^{-2}$ being the noise variance. Consider that

$$|f_x(\omega) - g(\omega)| \leq |f_x(\omega) - d^{-1}(\omega)| + |d^{-1}(\omega) - g(\omega)| < |f_x(\omega) - d^{-1}(\omega)| + \epsilon/2.$$

Also,

$$\begin{aligned} |f_x(\omega) - d^{-1}(\omega)| &= \left| \sigma_w^2 |\alpha(e^{-2\pi i \omega})|^{-2} - d^{-1}(\omega) \right| \\ &= \left| \sigma_w^{-2} |\alpha(e^{-2\pi i \omega})|^2 - d(\omega) \right| \cdot \left[\sigma_w^2 |\alpha(e^{-2\pi i \omega})|^{-2} d^{-1}(\omega) \right] \\ &< \delta \sigma_w^2 |\alpha(e^{-2\pi i \omega})|^{-2} G. \end{aligned}$$

But $\epsilon^* - d(\omega) < \sigma_w^{-2} |\alpha(e^{-2\pi i \omega})|^2 < \epsilon^* + d(\omega)$, so that

$$\sigma_w^2 |\alpha(e^{-2\pi i \omega})|^{-2} < \frac{1}{\epsilon^* - d(\omega)} < \frac{1}{\epsilon^* - G^{-1}} = \frac{1}{\epsilon [G(2G + \epsilon)]^{-1} - G^{-1}} = G + \epsilon/2.$$

We now have that

$$|f_x(\omega) - d^{-1}(\omega)| < \epsilon [G(2G + \epsilon)]^{-1} \cdot G + \epsilon/2 \cdot G = \epsilon/2.$$

Finally,

$$|f_x(\omega) - g(\omega)| < \epsilon/2 + \epsilon/2 = \epsilon,$$

as was to be shown. \square

It should be obvious from the proof of the result, that the property holds if AR(p) is replaced by MA(q) or even ARMA(p, q). As a practical point, it is easier to fit autoregressions of successively increasing order to data, and this is why the property is stated for an AR, even though the MA case is easier to establish.

Appendix R

R Supplement

R.1 First Things First

If you do not already have R, point your browser to the Comprehensive R Archive Network (CRAN), <http://cran.r-project.org/> and download and install it. The installation includes help files and some user manuals. You can find helpful tutorials by following CRAN's link to *Contributed Documentation*. If you are a novice, then RStudio (<https://www.rstudio.com/>) will make using R easier.

R.2 astsa

There is an R package for the text called `astsa` (*Applied Statistical Time Series Analysis*), which was the name of the software distributed with the first and second editions of Shumway & Stoffer (2000), and the original version, Shumway (1988). The package can be obtained from CRAN and its mirrors in the usual way. To download and install `astsa`, start R and type

```
install.packages("astsa")
```

You will be asked to choose the closest CRAN mirror to you. As with all packages, you have to load `astsa` before you use it by issuing the command

```
library(astsa)
```

All the data are loaded when the package is loaded. If you create a `.First` function as follows,

```
.First <- function(){library(astsa)}
```

and save the workspace when you quit, `astsa` will be loaded at every start until you change `.First`.

R is not consistent with help files across different operating systems. The best help system is the html help, which can be started issuing the command `help.start()` and then following the *Packages* link to `astsa`. A useful command to see all the data files available to you, including those loaded with `astsa`, is

```
data()
```

R.3 Getting Started

The convention throughout the text is that R code is in **blue**, output is **purple** and comments are **# green**. Get comfortable, then start her up and try some simple tasks.

```
2+2          # addition
[1] 5
5*5 + 2    # multiplication and addition
[1] 27
5/5 - 3    # division and subtraction
[1] -2
log(exp(pi)) # log, exponential, pi
[1] 3.141593
sin(pi/2)   # sinusoids
[1] 1
exp(1)^(-2) # power
[1] 0.1353353
sqrt(8)     # square root
[1] 2.828427
1:5         # sequences
[1] 1 2 3 4 5
seq(1, 10, by=2) # sequences
[1] 1 3 5 7 9
rep(2, 3)      # repeat 2 three times
[1] 2 2 2
```

Next, we'll use *assignment* to make some *objects*:

```
x <- 1 + 2 # put 1 + 2 in object x
x = 1 + 2   # same as above with fewer keystrokes
1 + 2 -> x # same
x          # view object x
[1] 3
(y = 9 * 3) # put 9 times 3 in y and view the result
[1] 27
(z = rnorm(5)) # put 5 standard normals into z and print z
[1] 0.96607946 1.98135811 -0.06064527 0.31028473 0.02046853
```

In general, `<-` and `=` are not the same; `<-` can be used anywhere, whereas the use of `=` is restricted. But when they are the same, we prefer to code using the least number of keystrokes.

It is worth pointing out R's *recycling rule* for doing arithmetic. In the code below, `c()` [concatenation] is used to create a vector. Note the use of the semicolon for multiple commands on one line.

```
x = c(1, 2, 3, 4); y = 2*x; z = c(10, 20); w = c(8, 3, 2)
x * y      # 1*2, 2*4, 3*6, 4*8
[1] 2 8 18 32
x + z      # 1+10, 2+20, 3+10, 4+20
[1] 11 22 13 24
x + w      # what happened here?
[1] 9 5 5 12
Warning message:
In y + w : longer object length is not a multiple of
shorter object length
```

To work your objects, use the following commands:

```
ls()           # list all objects
"dummy" "mydata" "x" "y" "z"
ls(pattern = "my") # list every object that contains "my"
"dummy" "mydata"
rm(dummy)      # remove object "dummy"
rm(list=ls())  # remove almost everything (use with caution)
help.start()   # html help and documentation
data()         # list of available data sets
help(exp)     # specific help (?exp is the same)
getwd()        # get working directory
setwd()        # change working directory
q()           # end the session (keep reading)
```

When you quit, R will prompt you to save an image of your current workspace. Answering *yes* will save the work you have done so far, and load it when you next start R. We have never regretted selecting *yes*, but we have regretted answering *no*.

To create your own data set inside R, you can make a data vector as follows:

```
mydata = c(1,2,3,2,1)
```

Now you have an object called `mydata` that contains five elements. R calls these objects *vectors* even though they have no dimensions (no rows, no columns); they do have order and length:

```
mydata       # display the data
[1] 1 2 3 2 1
mydata[3]    # the third element
[1] 3
mydata[3:5]  # elements three through five
[1] 3 2 1
mydata[-(1:2)] # everything except the first two elements
[1] 3 2 1
length(mydata) # number of elements
[1] 5
dim(mydata)    # no dimensions
NULL
mydata = as.matrix(mydata) # make it a matrix
dim(mydata)    # now it has dimensions
[1] 5 1
```

If you have an external data set, you can use `scan` or `read.table` (or some variant) to input the data. For example, suppose you have an ASCII (text) data file called `dummy.txt` in your working directory, and the file looks like this:

1 2 3 2 1
9 0 2 1 0

```
(dummy = scan("dummy.txt"))          # scan and view it
Read 10 items
[1] 1 2 3 2 1 9 0 2 1 0
(dummy = read.table("dummy.txt"))    # read and view it
V1 V2 V3 V4 V5
1 2 3 2 1
9 0 2 1 0
```

There is a difference between `scan` and `read.table`. The former produced a data vector of 10 items while the latter produced a *data frame* with names `V1` to `V5` and two

observations per variate. In this case, if you want to list (or use) the second variate, `V2`, you would use

```
dummy$V2
[1] 2 0
```

and so on. You might want to look at the help files `?scan` and `?read.table` now. Data frames (`?data.frame`) are “used as the fundamental data structure by most of R’s modeling software.” Notice that R gave the columns of `dummy` generic names, `V1`, `...`, `V5`. You can provide your own names and then use the names to access the data without the use of `$` as above.

```
colnames(dummy) = c("Dog", "Cat", "Rat", "Pig", "Man")
attach(dummy)
Cat
[1] 2 0
Rat*(Pig - Man) # animal arithmetic
[1] 3 2
head(dummy) # view the first few lines of a data file
detach(dummy) # clean up (if desired)
```

R is case sensitive, thus `cat` and `Cat` are different. Also, `cat` is a reserved name (`?cat`) in R, so using `"cat"` instead of `"Cat"` may cause problems later. You may also include a `header` in the data file to avoid `colnames()`. For example, if you have a *comma separated values* file `dummy.csv` that looks like this,

Dog,Cat,Rat,Pig,Man
1,2,3,2,1
9,0,2,1,0

then use the following command to read the data.

```
(dummy = read.csv("dummy.csv"))
  Dog Cat Rat Pig Man
1   1   2   3   2   1
2   9   0   2   1   0
```

The default for `.csv` files is `header=TRUE`; type `?read.table` for further information on similar types of files.

Some commands that are used frequently to manipulate data are `c()` for *concatenation*, `cbind()` for *column binding*, and `rbind()` for *row binding*.

```
x = 1:3; y = 4:6
(u = c(x, y))      # an R vector
[1] 1 2 3 4 5 6
(u1 = cbind(x, y)) # a 3 by 2 matrix
     x y
[1,] 1 4
[2,] 2 5
[3,] 3 6
(u2 = rbind(x ,y)) # a 2 by 3 matrix
[,1] [,2] [,3]
x    1    2    3
y    4    5    6
```

For example, `u1[,2]` is the second column of the matrix `u1`, whereas `u2[1,]` is the first row of `u2`.

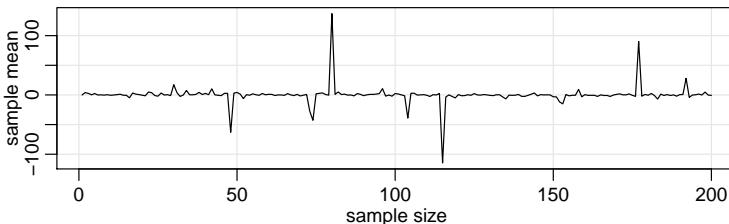


Fig. R.1. Crazy example.

Summary statistics are fairly easy to obtain. We will simulate 25 normals with $\mu = 10$ and $\sigma = 4$ and then perform some basic analyses. The first line of the code is `set.seed`, which fixes the seed for the generation of pseudorandom numbers. Using the same seed yields the same results; to expect anything else would be insanity.

```
set.seed(90210)      # so you can reproduce these results
x = rnorm(25, 10, 4) # generate the data
c( mean(x), median(x), var(x), sd(x) ) # guess
[1] 9.473883 9.448511 13.926701 3.731850
c( min(x), max(x) ) # smallest and largest values
[1] 2.678173 17.326089
which.max(x) # index of the max (x[25] in this case)
[1] 25
summary(x) # a five number summary with six numbers
  Min. 1st Qu. Median Mean 3rd Qu. Max.
  2.678   7.824   9.449  9.474  11.180  17.330
boxplot(x); hist(x); stem(x) # visual summaries (not shown)
```

It can't hurt to learn a little about programming in R because you will see some of it along the way. Consider a simple program that we will call `crazy` to produce a graph of a sequence of sample means of increasing sample sizes from a Cauchy distribution with location parameter zero.

```
1 crazy <- function(num) {
2   x <- c()
3   for (n in 1:num) { x[n] <- mean(rcauchy(n)) }
4   plot(x, type="l", xlab="sample size", ylab="sample mean")
5 }
```

The first line creates the function `crazy` and gives it one argument, `num`, that is the sample size that will end the sequence. Line 2 makes an empty vector, `x`, that will be used to store the sample means. Line 3 generates `n` random Cauchy variates [`rcauchy(n)`], finds the mean of those values, and puts the result into `x[n]`, the n -th value of `x`. The process is repeated in a “do loop” `num` times so that `x[1]` is the sample mean from a sample of size one, `x[2]` is the sample mean from a sample of size two, and so on, until finally, `x[num]` is the sample mean from a sample of size `num`. After the do loop is complete, the fourth line generates a graphic (see Figure R.1). The fifth line closes the function. To use `crazy` ending with sample of size of 200, type

`crazy(200)`

and you will get a graphic that looks like Figure R.1 .

Finally, a word of caution: `TRUE` and `FALSE` are reserved words, whereas `T` and `F` are initially set to these. Get in the habit of using the words rather than the letters `T` or `F` because you may get into trouble if you do something like

```
F = qf(p=.01, df1=3, df2=9)
```

so that `F` is no longer `FALSE`, but a quantile of the specified F -distribution.

R.4 Time Series Primer

In this section, we give a brief introduction on using R for time series. *We assume that `astsa` has been loaded.* To create a time series object, use the command `ts`. Related commands are `as.ts` to coerce an object to a time series and `is.ts` to test whether an object is a time series. First, make a small data set:

```
(mydata = c(1,2,3,2,1) ) # make it and view it
[1] 1 2 3 2 1
```

Now make it a time series:

```
(mydata = as.ts(mydata) )
Time Series:
Start = 1
End = 5
Frequency = 1
[1] 1 2 3 2 1
```

Make it an annual time series that starts in 1950:

```
(mydata = ts(mydata, start=1950) )
Time Series:
Start = 1950
End = 1954
Frequency = 1
[1] 1 2 3 2 1
```

Now make it a quarterly time series that starts in 1950-III:

```
(mydata = ts(mydata, start=c(1950,3), frequency=4) )
      Qtr1 Qtr2 Qtr3 Qtr4
1950          1    2
1951          3    2    1
time(mydata) # view the sampled times
      Qtr1   Qtr2   Qtr3   Qtr4
1950      1950.50 1950.75
1951 1951.00 1951.25 1951.50
```

To use part of a time series object, use `window()`:

```
(x = window(mydata, start=c(1951,1), end=c(1951,3) ))
      Qtr1 Qtr2 Qtr3
1951    3    2    1
```

Next, we'll look at lagging and differencing. First make a simple series, x_t :

```
x = ts(1:5)
```

Now, column bind (`cbind`) lagged values of x_t and you will notice that `lag(x)` is *forward* lag, whereas `lag(x, -1)` is *backward* lag.

```
cbind(x, lag(x), lag(x,-1))
  x lag(x) lag(x, -1)
0  NA     1     NA
1  1     2     NA
2  2     3     1
3  3     4     2 <- in this row, for example, x is 3,
4  4     5     3   lag(x) is ahead at 4, and
5  5     NA    4   lag(x,-1) is behind at 2
6  NA    NA    5
```

Compare `cbind` and `ts.intersect`:

```
ts.intersect(x, lag(x,1), lag(x,-1))
Time Series: Start = 2 End = 4 Frequency = 1
  x lag(x, 1) lag(x, -1)
2 2     3     1
3 3     4     2
4 4     5     3
```

To difference a series, $\nabla x_t = x_t - x_{t-1}$, use

`diff(x)`

but note that

`diff(x, 2)`

is *not* second order differencing, it is $x_t - x_{t-2}$. For second order differencing, that is, $\nabla^2 x_t$, do one of these:

```
diff(diff(x))
diff(x, diff=2) # same thing
```

and so on for higher order differencing.

We will also make use of regression via `lm()`. First, suppose we want to fit a simple linear regression, $y = \alpha + \beta x + \epsilon$. In R, the formula is written as `y~x`:

```
set.seed(1999)
x = rnorm(10)
y = x + rnorm(10)
summary(fit <- lm(y~x) )
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.2576    0.1892   1.362   0.2104  
x           0.4577    0.2016   2.270   0.0529  
-- 
Residual standard error: 0.58 on 8 degrees of freedom
Multiple R-squared: 0.3918,      Adjusted R-squared: 0.3157 
F-statistic: 5.153 on 1 and 8 DF, p-value: 0.05289
plot(x, y) # draw a scatterplot of the data (not shown)
abline(fit) # add the fitted line to the plot (not shown)
```

All sorts of information can be extracted from the `lm` object, which we called `fit`.

For example,

```
resid(fit) # will display the residuals (not shown)
fitted(fit) # will display the fitted values (not shown)
lm(y ~ 0 + x) # will exclude the intercept (not shown)
```

You have to be careful if you use `lm()` for lagged values of a time series. If you use `lm()`, then what you have to do is align the series using `ts.intersect`. Please read the warning *Using time series* in the `lm()` help file [`help(lm)`]. Here is an example regressing `astsa` data, weekly cardiovascular mortality (`cmort`) on

particulate pollution (`part`) at the present value and lagged four weeks (`part4`). First, we create `ded`, which consists of the intersection of the three series:

```
ded = ts.intersect(cmort, part, part4=lag(part, -4))
```

Now the series are all aligned and the regression will work.

```
summary(fit <- lm(cmort~part+part4, data=ded, na.action=NULL) )
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 69.01020    1.37498  50.190 < 2e-16
part         0.15140    0.02898   5.225 2.56e-07
part4        0.26297    0.02899   9.071 < 2e-16
---
Residual standard error: 8.323 on 501 degrees of freedom
Multiple R-squared:  0.3091,    Adjusted R-squared:  0.3063
F-statistic: 112.1 on 2 and 501 DF,  p-value: < 2.2e-16
```

There was no need to rename `lag(part, -4)` to `part4`, it's just an example of what you can do.

An alternative to the above is the package `dynlm`, which has to be installed. After the package is installed, the previous example may be run as follows:

```
library(dynlm)           # load the package
fit = dynlm(cmort~part + L(part,4))    # no new data file needed
summary(fit)
```

The output is identical to the `lm` output. To fit another model, for example, add the temperature series `temp`, the advantage of `dynlm` is that a new data file does not have to be created. We could just run

```
summary(dynlm(cmort~ temp + part + L(part,4)))
```

In Problem 2.1, you are asked to fit a regression model

$$x_t = \beta t + \alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t) + w_t$$

where x_t is logged Johnson & Johnson quarterly earnings ($n = 84$), and $Q_i(t)$ is the indicator of quarter $i = 1, 2, 3, 4$. The indicators can be made using `factor`.

```
trend = time(jj) - 1970      # helps to 'center' time
Q     = factor(cycle(jj) )    # make (Q)quarter factors
reg   = lm(log(jj)~trend + Q, na.action=NULL) # no intercept
model.matrix(reg)            # view the model design matrix
  trend Q1 Q2 Q3 Q4
1  -10.00  1  0  0  0
2   -9.75  0  1  0  0
3   -9.50  0  0  1  0
4   -9.25  0  0  0  1
.    .
.    .
.    .
summary(reg)                # view the results (not shown)
```

The workhorse for ARIMA simulations is `arima.sim`. Here are some examples; no output is shown here so you're on your own.

```
x = arima.sim(list(order=c(1,0,0), ar=.9), n=100) + 50    # AR(1) w/mean 50
x = arima.sim(list(order=c(2,0,0), ar=c(1,-.9)), n=100)    # AR(2)
x = arima.sim(list(order=c(1,1,1), ar=.9 ,ma=-.5), n=200) # ARIMA(1,1,1)
```

An easy way to fit ARIMA models is to use `sarima` from `astsa`. The script is used in Chapter 3 and is introduced in Section 3.7.

R.4.1 Graphics

We introduced some graphics without saying much about it. Many people use the graphics package `ggplot2`, but for quick and easy graphing of time series, the R base graphics does fine and is what we discuss here. As seen in Chapter 1, a time series may be plotted in a few lines, such as

```
plot(speech)
```

in Example 1.3, or the multifigure plot

```
plot.ts(cbind(soi, rec))
```

which we made little fancier in Example 1.5:

```
par(mfrow = c(2,1))
plot(soi, ylab='', xlab='', main='Southern Oscillation Index')
plot(rec, ylab='', xlab='', main='Recruitment')
```

But, if you compare the results of the above to what is displayed in the text, there is a slight difference because we improved the aesthetics by adding a grid and cutting down on the margins. This is how we actually produced Figure 1.3:

```
1 dev.new(width=7, height=4)           # default is 7 x 7 inches
2 par(mar=c(3,3,1,1), mgp=c(1.6,.6,0)) # change the margins (?par)
3 plot(speech, type='n')
4 grid(lty=1, col=gray(.9)); lines(speech)
```

In line 1, the dimensions are in inches. Line 2 adjusts the margins; see `help(par)` for a complete list of settings. In line 3, the `type='n'` means to set up the graph, but don't actually plot anything yet. Line 4 adds a grid and then plots the lines. The reason for using `type='n'` is to avoid having the grid lines on top of the data plot. You can print the graphic directly to a pdf, for example, by replacing line 1 with something like

```
pdf(file="speech.pdf", width=7, height=4)
```

but you have to turn the device off to complete the file save:

```
dev.off()
```

Here is the code we used to plot two series individually in Figure 1.5:

```
dev.new(width=7, height=6)
par(mfrow = c(2,1), mar=c(2,2,1,0)+.5, mgp=c(1.6,.6,0))
plot(soi, ylab='', xlab='', main='Southern Oscillation Index', type='n')
grid(lty=1, col=gray(.9)); lines(soi)
plot(rec, ylab='', main='Recruitment', type='n')
grid(lty=1, col=gray(.9)); lines(rec)
```

For plotting many time series, `plot.ts` and `ts.plot` are available. If the series are all on the same scale, it might be useful to do the following:

```
ts.plot(cmort, tempr, part, col=1:3)
legend('topright', legend=c('M','T','P'), lty=1, col=1:3)
```

This produces a plot of all three series on the same axes with different colors, and then adds a legend. We are not restricted to using basic colors; an internet search on ‘R colors’ is helpful. The following code gives separate plots of each different series (with a limit of 10):

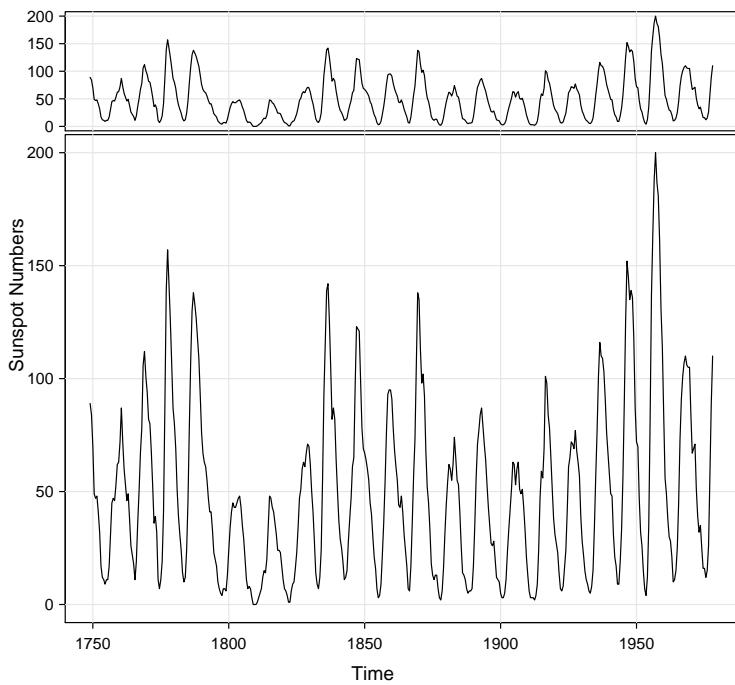


Fig. R.2. The sunspot numbers plotted in different-sized boxes, demonstrating that the dimensions of the graphic matters when displaying time series data.

```
plot.ts(cbind(cmort, tempr, part) )
plot.ts(eqexp)                                # you will get a warning
plot.ts(eqexp[,9:16], main='Explosions') # but this works
```

Finally, we mention that size matters when plotting time series. Figure R.2 shows the sunspot numbers discussed in Problem 4.9 plotted with varying dimension size as follows.

```
layout(matrix(c(1:2, 1:2), ncol=2), height=c(.2,.8))
par(mar=c(.2,3.5,0,.5), oma=c(3.5,0,.5,0), mgp=c(2,.6,0), tcl=-.3, las=1)
plot(sunspotz, type='n', xaxt='no', ylab='')
  grid(lty=1, col=gray(.9))
  lines(sunspotz)
plot(sunspotz, type='n', ylab='')
  grid(lty=1, col=gray(.9))
  lines(sunspotz)
title(xlab="Time", outer=TRUE, cex.lab=1.2)
mtext(side=2, "Sunspot Numbers", line=2, las=0, adj=.75)
```

The result is shown in Figure R.2. The top plot is wide and narrow, revealing the fact that the series rises quickly ↑ and falls slowly ↓. The bottom plot, which is more square, obscures this fact. You will notice that in the main part of the text, we never plotted a series in a square box. The ideal shape for plotting time series, in most instances, is when the time axis is much wider than the value axis.

References

- Akaike, H. (1969). Fitting autoregressive models for prediction. *Ann. Inst. Stat. Math.*, 21, 243-247.
- Akaike, H. (1973). Information theory and an extension of the maximum likelihood principal. In *2nd Int. Symp. Inform. Theory*, 267-281. B.N. Petrov and F. Csake, eds. Budapest: Akademia Kiado.
- Akaike, H. (1974). A new look at statistical model identification. *IEEE Trans. Automat. Contr.*, AC-19, 716-723.
- Alagón, J. (1989). Spectral discrimination for two groups of time series. *J. Time Series Anal.*, 10, 203-214.
- Alspach, D.L. and H.W. Sorenson (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. Automat. Contr.*, AC-17, 439-447.
- Anderson, B.D.O. and J.B. Moore (1979). *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall.
- Anderson, T.W. (1978). Estimation for autoregressive moving average models in the time and frequency domain. *Ann. Stat.*, 5, 842-865.
- Anderson, T.W. (1984). *An Introduction to Multivariate Statistical Analysis*, 2nd ed. New York: Wiley.
- Ansley, C.F. and P. Newbold (1980). Finite sample properties of estimators for autoregressive moving average processes. *J. Econ.*, 13, 159-183.
- Ansley, C.F. and R. Kohn (1982). A geometrical derivation of the fixed interval smoothing algorithm. *Biometrika*, 69 , 486-487.
- Antognini, J.F., M.H. Buonocore, E.A. Disbrow, and E. Carstens (1997). Isoflurane anesthesia blunts cerebral responses to noxious and innocuous stimuli: a fMRI study. *Life Sci.*, 61, PL349-PL354.
- Bandettini, A., A. Jesmanowicz, E.C. Wong, and J.S. Hyde (1993). Processing strategies for time-course data sets in functional MRI of the human brain. *Magnetic Res. Med.*, 30, 161-173.
- Bar-Shalom, Y. (1978). Tracking methods in a multi-target environment. *IEEE Trans. Automat. Contr.*, AC-23, 618-626.

- Bar-Shalom, Y. and E. Tse (1975). Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11, 4451-4460.
- Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, 41, 164-171.
- Bazza, M., R.H. Shumway, and D.R. Nielsen (1988). Two-dimensional spectral analysis of soil surface temperatures. *Hilgardia*, 56, 1-28.
- Bedrick, E.J. and C.-L. Tsai (1994). Model selection for multivariate regression in small samples. *Biometrics*, 50, 226-231.
- Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, 289-300.
- Beran, J. (1994). *Statistics for Long Memory Processes*. New York: Chapman and Hall.
- Berk, K.N. (1974). Consistent autoregressive spectral estimates. *Ann. Stat.*, 2, 489-502.
- Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. R. Stat. Soc. B*, 36, 192-236.
- Bhat, R.R. (1985). *Modern Probability Theory*, 2nd ed. New York: Wiley.
- Bhattacharya, A. (1943). On a measure of divergence between two statistical populations. *Bull. Calcutta Math. Soc.*, 35, 99-109.
- Billingsley, P. (1999). *Convergence of Probability Measures*, (2nd edition). New York: Wiley.
- Blackman, R.B. and J.W. Tukey (1959). *The Measurement of Power Spectra from the Point of View of Communications Engineering*. New York: Dover.
- Blight, B.J.N. (1974). Recursive solutions for the estimation of a stochastic parameter. *J. Am. Stat. Assoc.*, 69, 477-481
- Bloomfield, P. (1976). *Fourier Analysis of Time Series: An Introduction*. New York: Wiley.
- Bloomfield, P. (2000). *Fourier Analysis of Time Series: An Introduction*, 2nd ed. New York: Wiley.
- Bloomfield, P. and J.M. Davis (1994). Orthogonal rotation of complex principal components. *Int. J. Climatol.*, 14, 759-775.
- Bogart, B. P., M. J. R. Healy, and J.W. Tukey (1962). The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking. In *Proc. of the Symposium on Time Series Analysis*, pp. 209-243, Brown University, Providence, USA.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroscedasticity. *J. Econ.*, 31, 307- 327.
- Box, G.E.P. and D.A. Pierce (1970). Distributions of residual autocorrelations in autoregressive integrated moving average models. *J. Am. Stat. Assoc.*, 72, 397-402.
- Box, G.E.P. and G.M. Jenkins (1970). *Time Series Analysis, Forecasting, and Control*. Oakland, CA: Holden-Day.
- Box, G.E.P. and G.C. Tiao (1973). *Bayesian Inference in Statistical Analysis*. New York: Wiley.
- Box, G.E.P., G.M. Jenkins and G.C. Reinsel (1994). *Time Series Analysis, Forecasting, and Control*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall.
- Breiman, L. and J. Friedman (1985). Estimating optimal transformations for multiple regression and correlation (with discussion). *J. Am. Stat. Assoc.*, 80, 580-619.

- Brillinger, D.R. (1973). The analysis of time series collected in an experimental design. In *Multivariate Analysis-III.*, pp. 241-256. P.R. Krishnaiah ed. New York: Academic Press.
- Brillinger, D.R. (1975). *Time Series: Data Analysis and Theory*. New York: Holt, Rinehart & Winston Inc.
- Brillinger, D.R. (1980). Analysis of variance and problems under time series models. In *Handbook of Statistics*, Vol I, pp. 237-278. P.R. Krishnaiah and D.R. Brillinger, eds. Amsterdam: North Holland.
- Brillinger, D.R. (1981, 2001). *Time Series: Data Analysis and Theory*, 2nd ed. San Francisco: Holden-Day. Republished in 2001 by the Society for Industrial and Applied Mathematics, Philadelphia.
- Brockwell, P.J. and R.A. Davis (1991). *Time Series: Theory and Methods*, 2nd ed. New York: Springer-Verlag.
- Bruce, A. and H-Y. Gao (1996). *Applied Wavelet Analysis with S-PLUS*. New York: Springer-Verlag.
- Cappé, O., Moulines, E., & Rydén, T. (2009). *Inference in Hidden Markov Models*. New York: Springer.
- Caines, P.E. (1988). *Linear Stochastic Systems*. New York: Wiley.
- Carlin, B.P., N.G. Polson, and D.S. Stoffer (1992). A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *J. Am. Stat. Assoc.*, 87, 493-500.
- Carter, C. K. and R. Kohn (1994). On Gibbs sampling for state space models. *Biometrika*, 81, 541-553.
- Chan, N.H. (2002). *Time Series: Applications to Finance*. New York: Wiley.
- Chernoff, H. (1952). A measure of asymptotic efficiency for tests of a hypothesis based on the sum of the observations. *Ann. Math. Stat.*, 25, 573-578.
- Cleveland, W.S. (1979). Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, 74, 829-836.
- Cochrane, D. and G.H. Orcutt (1949). Applications of least squares regression to relationships containing autocorrelated errors. *J. Am. Stat. Assoc.*, 44, 32-61.
- Cooley, J.W. and J.W. Tukey (1965). An algorithm for the machine computation of complex Fourier series. *Math. Comput.*, 19, 297-301.
- Cressie, N.A.C. (1993). *Statistics for Spatial Data*. New York: Wiley.
- Dahlhaus, R. (1989). Efficient parameter estimation for self-similar processes. *Ann. Stat.*, 17, 1749-1766.
- Dargahi-Noubary, G.R. and P.J. Laycock (1981). Spectral ratio discriminants and information theory. *J. Time Series Anal.*, 16, 201-219.
- Danielson, J. (1994). Stochastic volatility in asset prices: Estimation with simulated maximum likelihood. *J. Econometrics*, 61, 375-400.
- Daubechies, I. (1992). *Ten Lectures on Wavelets*. Philadelphia: CBMS-NSF Regional Conference Series in Applied Mathematics.
- Davies, N., C.M. Triggs, and P. Newbold (1977). Significance levels of the Box-Pierce portmanteau statistic in finite samples. *Biometrika*, 64, 517-522.
- Dent, W. and A.-S. Min. (1978). A Monte Carlo study of autoregressive-integrated-moving average processes. *J. Econ.*, 7, 23-55.

- Dempster, A.P., N.M. Laird and D.B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B*, 39, 1-38.
- Ding, Z., C.W.J. Granger, and R.F. Engle (1993). A long memory property of stock market returns and a new model. *J. Empirical Finance*, 1, 83-106.
- Donoho, D.L. and I.M. Johnstone (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81, 425-455.
- Donoho, D.L. and I.M. Johnstone (1995). Adapting to unknown smoothness via wavelet shrinkage. *J. of Am. Stat. Assoc.*, 90, 1200-1224.
- Douc, R., E. Moulines, and D.S. Stoffer (2014). *Nonlinear Time Series: Theory, Methods, and Applications with R Examples*. Boca Raton: CRC Press.
- Durbin, J. (1960). Estimation of parameters in time series regression models. *J. R. Stat. Soc. B*, 22, 139-153.
- Durbin, J. and S.J. Koopman (2001). *Time Series Analysis by State Space Methods* Oxford: Oxford University Press.
- Efron, B. and R. Tibshirani (1994). *An Introduction to the Bootstrap*. New York: Chapman and Hall.
- Engle, R.F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50, 987-1007.
- Engle, R.F., D. Nelson, and T. Bollerslev (1994). ARCH Models. In *Handbook of Econometrics*, Vol IV, pp. 2959-3038. R. Engle and D. McFadden, eds. Amsterdam: North Holland.
- Evans, G. B. A. and Savin, N. E. (1981) The Calculation of the Limiting Distribution of the Least Squares Estimator of the Parameter in a Random Walk Model. *Ann. Statist.*, 1114-1118. <http://projecteuclid.org/euclid-aos/1176345591>.
- Fan, J., & Kreutzberger, E. (1998). Automatic local smoothing for spectral density estimation. *Scand. J. Statist.*, 25, 359-369.
- Fox, R. and M.S. Taqqu (1986). Large sample properties of parameter estimates for strongly dependent stationary Gaussian time series. *Ann. Stat.*, 14, 517-532.
- Friedman, J.H. (1984). A Variable Span Smoother. Tech. Rep. No. 5, Lab. for Computational Statistics, Dept. Statistics, Stanford Univ., California.
- Friedman, J.H. and W. Stuetzle. (1981). Projection pursuit regression. *J. Am. Stat. Assoc.*, 76, 817-823.
- Frühwirth-Schnatter, S. (1994). Data Augmentation and Dynamic Linear Models. *J. Time Series Anal.*, 15, 183-202.
- Fuller, W.A. (1976). *Introduction to Statistical Time Series*. New York: Wiley.
- Fuller, W.A. (1996). *Introduction to Statistical Time Series, 2nd ed.* New York: Wiley.
- Gelfand, A.E. and A.F.M. Smith (1990). Sampling-based approaches to calculating marginal densities. *J. Am. Stat. Assoc.*, 85, 398-409.
- Gelman, A., J. Carlin, H. Stern, and D. Rubin (1995). *Bayesian Data Analysis*. London: Chapman and Hall.
- Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6, 721-741.
- Gerlach, R., Carter, C., and Kohn, R. (2000). Efficient Bayesian inference for dynamic mixture models. *J. Amer. Statist. Assoc.*, 95, 819-828.

- Geweke, J.F. (1977). The dynamic factor analysis of economic time series models. In *Latent Variables in Socio-Economic Models*, pp 365-383. D. Aigner and A. Goldberger, eds. Amsterdam: North Holland.
- Geweke, J.F. and K.J. Singleton (1981). Latent variable models for time series: A frequency domain approach with an application to the Permanent Income Hypothesis. *J. Econ.*, 17, 287-304.
- Geweke, J.F. and S. Porter-Hudak (1983). The estimation and application of long-memory time series models. *J. Time Series Anal.*, 4, 221-238.
- Gilks, W.R., S. Richardson, and D.J. Spiegelhalter (eds.) (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.
- Giri, N. (1965). On complex analogues of T^2 and R^2 tests. *Ann. Math. Stat.*, 36, 664-670.
- Goldfeld, S.M. and R.E. Quandt (1973). A Markov model for switching regressions. *J. Econ.*, 1, 3-16.
- Goodman, N.R. (1963). Statistical analysis based on a certain multivariate complex Gaussian distribution. *Ann. Math. Stat.*, 34, 152-177.
- Gordon, K. and A.F.M. Smith (1988). Modeling and monitoring discontinuous changes in time series. In *Bayesian Analysis of Time Series and Dynamic Models*, 359-392.
- Gordon, K. and A.F.M. Smith (1990). Modeling and monitoring biomedical time series. *J. Am. Stat. Assoc.*, 85, 328-337.
- Gouriéroux, C. (1997). *ARCH Models and Financial Applications*. New York: Springer-Verlag.
- Granger, C.W. and R. Joyeux (1980). An introduction to long-memory time series models and fractional differencing. *J. Time Series Anal.*, 1, 15-29.
- Grenander, U. (1951). On empirical spectral analysis of stochastic processes. *Arkiv for Matematik*, 1, 503-531.
- Grenander, U. and M. Rosenblatt (1957). *Statistical Analysis of Stationary Time Series*. New York: Wiley.
- Grether, D.M. and M. Nerlove (1970). Some properties of optimal seasonal adjustment. *Econometrica*, 38, 682-703.
- Gupta, N.K. and R.K. Mehra (1974). Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Trans. Automat. Contr.*, AC-19, 774-783.
- Hamilton, J.D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57, 357-384.
- Hannan, E.J. (1970). *Multiple Time Series*. New York: Wiley.
- Hannan, E. J. and B. G. Quinn (1979). The determination of the order of an autoregression. *J. Royal Statistical Society, B*, 41, 190-195.
- Hannan, E.J. and M. Deistler (1988). *The Statistical Theory of Linear Systems*. New York: Wiley.
- Hansen, J., M. Sato, R. Ruedy, K. Lo, D.W. Lea, and M. Medina-Elizade (2006). Global temperature change. *Proc. Natl. Acad. Sci.*, 103, 14288-14293.
- Harrison, P.J. and C.F. Stevens (1976). Bayesian forecasting (with discussion). *J. R. Stat. Soc. B*, 38, 205-247.

- Harvey, A.C. and P.H.J. Todd (1983). Forecasting economic time series with structural and Box-Jenkins models: A case study. *J. Bus. Econ. Stat.*, 1, 299-307.
- Harvey, A.C. and R.G. Pierse (1984). Estimating missing observations in economic time series. *J. Am. Stat. Assoc.*, 79, 125-131.
- Harvey, A.C. (1991). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge University Press.
- Harvey, A.C. (1993). *Time Series Models*. Cambridge, MA: MIT Press.
- Harvey A.C., E. Ruiz and N. Shephard (1994). Multivariate stochastic volatility models. *Rev. Economic Studies*, 61, 247-264.
- Haslett, J. and A.E. Raftery (1989) Space-time modelling with long-memory dependence: Assessing Ireland's wind power resource (C/R: 89V38 p21-50) *Applied Statistics*, 38, 1-21
- Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97-109.
- Hosking, J.R.M. (1981). Fractional differencing. *Biometrika*, 68, 165-176.
- Hurst, H. (1951). Long term storage capacity of reservoirs. *Trans. Am. Soc. Civil Eng.*, 116, 778-808.
- Hurvich, C.M. and S. Zeger (1987). Frequency domain bootstrap methods for time series. *Tech. Report 87-115*, Department of Statistics and Operations Research, Stern School of Business, New York University.
- Hurvich, C.M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika*, 76, 297-307.
- Hurvich, C.M. and K.I. Beltrao (1993). Asymptotics for the low-frequency ordinates of the periodogram for a long-memory time series. *J. Time Series Anal.*, 14, 455-472.
- Hurvich, C.M., R.S. Deo and J. Brodsky (1998). The mean squared error of Geweke and Porter-Hudak's estimator of the memory parameter of a long-memory time series. *J. Time Series Anal.*, 19, 19-46.
- Hurvich, C.M. and R.S. Deo (1999). Plug-in selection of the number of frequencies in regression estimates of the memory parameter of a long-memory time series. *J. Time Series Anal.*, 20, 331-341.
- Jacquier, E., N.G. Polson, and P.E. Rossi (1994). Bayesian analysis of stochastic volatility models. *J. Bus. Econ. Stat.*, 12, 371-417.
- Jazwinski, A.H. (1970). *Stochastic Processes and Filtering Theory*. New York: Academic Press.
- Jenkins, G.M. and D.G. Watts. (1968). *Spectral Analysis and Its Applications*. San Francisco: Holden-Day.
- Johnson, R.A. and D.W. Wichern (1992). *Applied Multivariate Statistical Analysis*, 3rd ed.. Englewood Cliffs, NJ: Prentice-Hall.
- Jones, P.D. (1994). Hemispheric surface air temperature variations: A reanalysis and an update to 1993. *J. Clim.*, 7, 1794-1802.
- Jones, R.H. (1980). Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics*, 22, 389-395.
- Jones, R.H. (1984). Fitting multivariate models to unequally spaced data. In *Time Series Analysis of Irregularly Observed Data*, pp. 158-188. E. Parzen, ed. Lecture Notes in Statistics, 25, New York: Springer-Verlag.

- Jones, R.H. (1993). *Longitudinal Data With Serial Correlation : A State-Space Approach*. London: Chapman and Hall.
- Journel, A.G. and C.H. Huijbregts (1978). *Mining Geostatistics*. New York: Academic Press.
- Juang, B.H. and L.R. Rabiner (1985). Mixture autoregressive hidden Markov models for speech signals, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-33, 1404-1413.
- Kakizawa, Y., R. H. Shumway, and M. Taniguchi (1998). Discrimination and clustering for multivariate time series. *J. Am. Stat. Assoc.*, 93, 328-340.
- Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Trans ASME J. Basic Eng.*, 82, 35-45.
- Kalman, R.E. and R.S. Bucy (1961). New results in filtering and prediction theory. *Trans. ASME J. Basic Eng.*, 83, 95-108.
- Kaufman, L. and P.J. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.
- Kay, S.M. (1988). *Modern Spectral Analysis: Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall.
- Kazakos, D. and P. Papantoni-Kazakos (1980). Spectral distance measuring between Gaussian processes. *IEEE Trans. Automat. Contr.*, AC-25, 950-959.
- Khatri, C.G. (1965). Classical statistical analysis based on a certain multivariate complex Gaussian distribution. *Ann. Math. Stat.*, 36, 115-119.
- Kim S., N. Shephard and S. Chib (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. *Rev. Economic Studies*, 65, p.361-393.
- Kitagawa, G. and W. Gersch (1984). A smoothness priors modeling of time series with trend and seasonality. *J. Am. Stat. Assoc.*, 79, 378-389.
- Kitagawa, G. (1987). Non-Gaussian state-space modeling of nonstationary time series (with discussion). *J. Am. Stat. Assoc.*, 82, 1032-1041, (C/R: p1041-1063; C/R: V83 p1231).
- Kitagawa, G. and W. Gersch (1996). *Smoothness Priors Analysis of Time Series*. New York: Springer-Verlag.
- Kolmogorov, A.N. (1941). Interpolation und extrapolation von stationären zufälligen Folgen. *Bull. Acad. Sci. U.R.S.S.*, 5, 3-14.
- Krishnaiah, P.R., J.C. Lee, and T.C. Chang (1976). The distribution of likelihood ratio statistics for tests of certain covariance structures of complex multivariate normal populations. *Biometrika*, 63, 543-549.
- Kullback, S. and R.A. Leibler (1951). On information and sufficiency. *Ann. Math. Stat.*, 22, 79-86.
- Kullback, S. (1958). *Information Theory and Statistics*. Gloucester, MA: Peter Smith.
- Lachenbruch, P.A. and M.R. Mickey (1968). Estimation of error rates in discriminant analysis. *Technometrics*, 10, 1-11.
- Lam, P.S. (1990). The Hamilton model with a general autoregressive component: Estimation and comparison with other models of economic time series. *J. Monetary Econ.*, 26, 409-432.
- Lay, T. (1997). Research required to support comprehensive nuclear test ban treaty monitoring. *National Research Council Report, National Academy Press*, 2101 Constitution Ave., Washington, DC 20055.

- Levinson, N. (1947). The Wiener (root mean square) error criterion in filter design and prediction. *J. Math. Phys.*, 25, 262-278.
- Lindgren, G. (1978). Markov regime models for mixed distributions and switching regressions. *Scand. J. Stat.*, 5, 81-91.
- Ljung, G.M. and G.E.P. Box (1978). On a measure of lack of fit in time series models. *Biometrika*, 65, 297-303.
- Lütkepohl, H. (1985). Comparison of criteria for estimating the order of a vector autoregressive process. *J. Time Series Anal.*, 6, 35-52.
- Lütkepohl, H. (1993). *Introduction to Multiple Time Series Analysis*, 2nd ed. Berlin: Springer-Verlag.
- MacQueen, J.B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley: University of California Press, 1:281-297
- Mallows, C.L. (1973). Some comments on C_p . *Technometrics*, 15, 661-675.
- McBratney, A.B. and R. Webster (1981). Detection of ridge and furrow pattern by spectral analysis of crop yield. *Int. Stat. Rev.*, 49, 45-52.
- McCulloch, R.E. and R.S. Tsay (1993). Bayesian inference and prediction for mean and variance shifts in autoregressive time series. *J. Am. Stat. Assoc.*, 88, 968-978.
- McDougall, A. J., D.S. Stoffer and D.E. Tyler (1997). Optimal transformations and the spectral envelope for real-valued time series. *J. Stat. Plan. Infer.*, 57, 195-214.
- McLeod A.I. (1978). On the distribution of residual autocorrelations in Box-Jenkins models. *J. R. Stat. Soc. B*, 40, 296-302.
- McLeod, A.I. and K.W. Hipel (1978). Preservation of the rescaled adjusted range, I. A reassessment of the Hurst phenomenon. *Water Resour. Res.*, 14, 491-508.
- McQuarrie, A.D.R. and C-L. Tsai (1998). *Regression and Time Series Model Selection*, Singapore: World Scientific.
- Meinhold, R.J. and N.D. Singpurwalla (1983). Understanding the Kalman filter. *Am. Stat.*, 37, 123-127.
- Meinhold, R.J. and N.D. Singpurwalla (1989). Robustification of Kalman filter models. *J. Am. Stat. Assoc.*, 84, 479-486.
- Meng X.L. and Rubin, D.B. (1991). Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm. *J. Am. Stat. Assoc.*, 86, 899-909.
- Metropolis N., A.W. Rosenbluth, M.N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21, 1087-1091.
- Mickens, R.E. (1990). *Difference Equations: Theory and Applications* (2nd ed). New York: Springer.
- Nason, G.P. (2008). *Wavelet Methods in Statistics with R*. New York: Springer.
- Newbold, P. and T. Bos (1985). *Stochastic Parameter Regression Models*. Beverly Hills: Sage.
- Ogawa, S., T.M. Lee, A. Nayak and P. Glynn (1990). Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magn. Reson. Med.*, 14, 68-78.
- Palma, W. (2007). *Long-Memory Time Series: Theory and Methods*. New York: Wiley.

- Palma, W. and N.H. Chan (1997). Estimation and forecasting of long-memory time series with missing values. *J. Forecast.*, 16, 395-410.
- Paparoditis, E. and Politis, D.N. (1999). The local bootstrap for periodogram statistics. *J. Time Series Anal.*, 20, 193-222.
- Parzen, E. (1962). On estimation of a probability density and mode. *Ann. Math. Stat.*, 35, 1065-1076.
- Parzen, E. (1983). Autoregressive spectral estimation. In *Time Series in the Frequency Domain, Handbook of Statistics*, Vol. 3, pp. 211-243. D.R. Brillinger and P.R. Krishnaiah eds. Amsterdam: North Holland.
- Pawitan, Y. and R.H. Shumway (1989). Spectral estimation and deconvolution for a linear time series model. *J. Time Series Anal.*, 10, 115-129.
- Peña, D. and I. Guttman (1988). A Bayesian approach to robustifying the Kalman filter. In *Bayesian Analysis of Time Series and Dynamic Linear Models*, pp. 227-254. J.C. Spall, ed. New York: Marcel Dekker.
- Percival, D.B. and A.T. Walden (1993). *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques* Cambridge: Cambridge University Press.
- Percival, D.B. and A.T. Walden (2000). *Wavelet Methods for Time Series Analysis*. Cambridge: Cambridge University Press.
- Petris G, Petrone S, Campagnoli P (2009). Dynamic Linear Models with R. New York: Springer.
- Phillips, P.C.B. (1987). Time series regression with a unit root. *Econometrica*, 55, 227-301.
- Phillips, P.C.B. and P. Perron (1988). Testing for unit roots in time series regression. *Biometrika*, 75, 335-346.
- Pinsker, M.S. (1964). *Information and Information Stability of Random Variables and Processes*, San Francisco: Holden Day.
- Pole, P.J. and M. West (1988). Nonnormal and nonlinear dynamic Bayesian modeling. In *Bayesian Analysis of Time Series and Dynamic Linear Models*, pp. 167-198. J.C. Spall, ed. New York: Marcel Dekker.
- Press, W.H., S.A. Teukolsky, W. T. Vetterling, and B.P. Flannery (1993). *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge: Cambridge University Press.
- Priestley, M.B., T. Subba-Rao and H. Tong (1974). Applications of principal components analysis and factor analysis in the identification of multi-variable systems. *IEEE Trans. Automat. Contr.*, AC-19, 730-734.
- Priestley, M.B. and T. Subba-Rao (1975). The estimation of factor scores and Kalman filtering for discrete parameter stationary processes. *Int. J. Contr.*, 21, 971-975.
- Priestley, M.B. (1981). *Spectral Analysis and Time Series*. Vol. 1: Univariate Series; Vol 2: Multivariate Series, Prediction and Control. New York: Academic Press.
- Priestley, M.B. (1988). *Nonlinear and Nonstationary Time Series Analysis*. London: Academic Press.
- Quandt, R.E. (1972). A new approach to estimating switching regressions. *J. Am. Stat. Assoc.*, 67, 306-310.
- Rabiner, L.R. and B.H. Juang (1986). An introduction to hidden Markov models, *IEEE Acoust., Speech, Signal Process.*, ASSP-34, 4-16.
- Rao, C.R. (1973). *Linear Statistical Inference and Its Applications*. New York: Wiley.

- Rao, M. M. (1978). Asymptotic Distribution of an Estimator of the Boundary Parameter of an Unstable Process. *Ann. Statist.* 185–190.
- Rauch, H.E., F. Tung, and C.T. Striebel (1965). Maximum likelihood estimation of linear dynamic systems. *J. AIAA*, 3, 1445-1450.
- Reinsel, G.C. (1997). *Elements of Multivariate Time Series Analysis*, 2nd ed. New York: Springer-Verlag.
- Remillard, B. (2011). Validity of the parametric bootstrap for goodness-of-fit testing in dynamic models. Available at SSRN 1966476.
- Renyi, A. (1961). On measures of entropy and information. In *Proceedings of 4th Berkeley Symp. Math. Stat. and Probability*, pp. 547-561, Berkeley: Univ. of California Press.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14, 465-471.
- Robinson, P.M. (1995). Gaussian semiparametric estimation of long range dependence. *Ann. Stat.*, 23, 1630-1661.
- Robinson, P.M. (2003). *Time Series With Long Memory*. Oxford: Oxford University Press.
- Rosenblatt, M. (1956a). A central limit theorem and a strong mixing condition. *Proc. Nat. Acad. Sci.*, 42, 43-47.
- Rosenblatt, M. (1956b). Remarks on some nonparametric estimates of a density functions. *Ann. Math. Stat.*, 27 642-669.
- Royston, P. (1982). An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics*, 31, 115-124.
- Said E. and D.A. Dickey (1984). Testing for unit roots in autoregressive moving average models of unknown order. *Biometrika*, 71, 599–607.
- Sandmann, G. and S.J. Koopman (1998). Estimation of stochastic volatility models via Monte Carlo maximum likelihood. *J. Econometrics*, 87 , 271-301.
- Sargan, J.D. (1964). Wages and prices in the United Kingdom: A study in econometric methodology. In *Econometric Analysis for National Economic Planning*, eds. P. E. Hart, G. Mills and J. K. Whitaker. London: Butterworths. reprinted in *Quantitative Economics and Econometric Analysis*, pp. 275-314, eds. K. F. Wallis and D. F. Hendry (1984). Oxford: Basil Blackwell.
- Scheffé, H. (1959). *The Analysis of Variance*. New York: Wiley.
- Schuster, A. (1898). On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. *Terrestrial Magnetism*, III, 11-41.
- Schuster, A. (1906). On the periodicities of sunspots. *Phil. Trans. R. Soc., Ser. A*, 206, 69-100.
- Schwarz, F. (1978). Estimating the dimension of a model. *Ann. Stat.*, 6, 461-464.
- Schweppe, F.C. (1965). Evaluation of likelihood functions for Gaussian signals. *IEEE Trans. Inform. Theory*, IT-4, 294-305.
- Shephard, N. (1996). Statistical aspects of ARCH and stochastic volatility. In *Time Series Models in Econometrics, Finance and Other Fields* , pp 1-100. D.R. Cox, D.V. Hinkley, and O.E. Barndorff-Nielson eds. London: Chapman and Hall.
- Shumway, R.H. and W.C. Dean (1968). Best linear unbiased estimation for multivariate stationary processes. *Technometrics*, 10, 523-534.
- Shumway, R.H. (1970). Applied regression and analysis of variance for stationary time series. *J. Am. Stat. Assoc.*, 65, 1527-1546.

- Shumway, R.H. (1971). On detecting a signal in N stationarily correlated noise series. *Technometrics*, 10, 523-534.
- Shumway, R.H. and A.N. Unger (1974). Linear discriminant functions for stationary time series. *J. Am. Stat. Assoc.*, 69, 948-956.
- Shumway, R.H. (1982). Discriminant analysis for time series. In *Classification, Pattern Recognition and Reduction of Dimensionality, Handbook of Statistics Vol. 2*, pp. 1-46. P.R. Krishnaiah and L.N. Kanal, eds. Amsterdam: North Holland.
- Shumway, R.H. and D.S. Stoffer (1982). An approach to time series smoothing and forecasting using the EM algorithm. *J. Time Series Anal.*, 3, 253-264.
- Shumway, R.H. (1983). Replicated time series regression: An approach to signal estimation and detection. In *Time Series in the Frequency Domain, Handbook of Statistics Vol. 3*, pp. 383-408. D.R. Brillinger and P.R. Krishnaiah, eds. Amsterdam: North Holland.
- Shumway, R.H. (1988). *Applied Statistical Time Series Analysis*. Englewood Cliffs, NJ: Prentice-Hall.
- Shumway, R.H., R.S. Azari, and Y. Pawitan (1988). Modeling mortality fluctuations in Los Angeles as functions of pollution and weather effects. *Environ. Res.*, 45, 224-241.
- Shumway, R.H. and D.S. Stoffer (1991). Dynamic linear models with switching. *J. Am. Stat. Assoc.*, 86, 763-769, (Correction: V87 p. 913).
- Shumway, R.H. and K.L. Verosub (1992). State space modeling of paleoclimatic time series. In *Pro. 5th Int. Meeting Stat. Climatol.* Toronto, pp. 22-26, June, 1992.
- Shumway, R.H., S.E. Kim and R.R. Blandford (1999). Nonlinear estimation for time series observed on arrays. Chapter 7, S. Ghosh, ed. *Asymptotics, Nonparametrics and Time Series*, pp. 227-258. New York: Marcel Dekker.
- Small, C.G. and D.L. McLeish (1994). *Hilbert Space Methods in Probability and Statistical Inference*. New York: Wiley.
- Smith, A.F.M. and M. West (1983). Monitoring renal transplants: An application of the multiprocess Kalman filter. *Biometrics*, 39, 867-878.
- Spliid, H. (1983). A fast estimation method for the vector autoregressive moving average model with exogenous variables. *J. Am. Stat. Assoc.*, 78, 843-849.
- Stoffer, D.S. (1982). Estimation of Parameters in a Linear Dynamic System with Missing Observations. Ph.D. Dissertation. Univ. California, Davis.
- Stoffer, D.S., M. Scher, G. Richardson, N. Day, and P. Coble (1988). A Walsh-Fourier analysis of the effects of moderate maternal alcohol consumption on neonatal sleep-state cycling. *J. Am. Stat. Assoc.*, 83, 954-963.
- Stoffer, D.S. and K.D. Wall (1991). Bootstrapping state space models: Gaussian maximum likelihood estimation and the Kalman filter. *J. Am. Stat. Assoc.*, 86, 1024-1033.
- Stoffer, D.S., D.E. Tyler, and A.J. McDougall (1993). Spectral analysis for categorical time series: Scaling and the spectral envelope. *Biometrika*, 80, 611-622.
- Stoffer, D.S. (1999). Detecting common signals in multiple time series using the spectral envelope. *J. Am. Stat. Assoc.*, 94, 1341-1356.
- Stoffer, D.S. and K.D. Wall (2004). Resampling in State Space Models. In *State Space and Unobserved Component Models Theory and Applications*, Chapter 9, pp. 227-258. Andrew Harvey, Siem Jan Koopman, and Neil Shephard, eds. Cambridge: Cambridge University Press.

- Sugiura, N. (1978). Further analysis of the data by Akaike's information criterion and the finite corrections. *Commun. Statist. A, Theory Methods*, 7, 13-26.
- Taniguchi, M., M.L. Puri, and M. Kondo (1994). Nonparametric approach for non-Gaussian vector stationary processes. *J. Mult. Anal.*, 56, 259-283.
- Tanner, M. and W.H. Wong (1987). The calculation of posterior distributions by data augmentation (with discussion). *J. Am. Stat. Assoc.*, 82, 528-554.
- Taylor, S. J. (1982). Financial returns modelled by the product of two stochastic processes – A study of daily sugar prices, 1961-79. In Anderson, O. D., editor, *Time Series Analysis: Theory and Practice*, Volume 1, pages 203–226. New York: Elsevier/North-Holland.
- Tiao, G.C. and R.S. Tsay (1989). Model specification in multivariate time series (with discussion). *J. Roy. Statist. Soc. B*, 51, 157-213.
- Tiao, G. C. and R.S. Tsay (1994). Some advances in nonlinear and adaptive modeling in time series analysis. *J. Forecast.*, 13, 109-131.
- Tiao, G.C., R.S. Tsay and T .Wang (1993). Usefulness of linear transformations in multivariate time series analysis. *Empir. Econ.*, 18, 567-593.
- Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Ann. Stat.*, 22, 1701-1728.
- Tong, H. (1983). *Threshold Models in Nonlinear Time Series Analysis*. Springer Lecture Notes in Statistics, 21. New York: Springer-Verlag.
- Tong, H. (1990). *Nonlinear Time Series: A Dynamical System Approach*. Oxford: Oxford Univ. Press.
- Tsay, Ruey S. (2002). *Analysis of Financial Time Series*. New York: Wiley.
- Venables, W.N. and B.D. Ripley (1994). *Modern Applied Statistics with S-Plus*. New York: Springer-Verlag.
- Wahba, G. (1980). Automatic smoothing of the log periodogram. *J. Am. Stat. Assoc.*, 75, 122-132.
- Watson, G.S. (1966). Smooth regression analysis. *Sankhya*, 26, 359-378.
- Weiss, A.A. (1984). ARMA models with ARCH errors. *J. Time Series Anal.*, 5, 129-143.
- West, M. and J. Harrison (1997). *Bayesian Forecasting and Dynamic Models 2nd ed.* New York: Springer-Verlag.
- Whittle, P. (1951). *Hypothesis Testing in Time Series Analysis*. Uppsala: Almqvist & Wiksell.
- Whittle, P. (1961). Gaussian estimation in stationary time series. *Bull. Int. Stat. Inst.*, 33, 1-26.
- Wiener, N. (1949). *The Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. New York: Wiley.
- Wu, C.F. (1983). On the convergence properties of the EM algorithm. *Ann. Stat.*, 11, 95-103.
- Young, P.C. and D.J. Pedregal (1998). Macro-economic relativity: Government spending, private investment and unemployment in the USA. Centre for Research on Environmental Systems and Statistics, Lancaster University, U.K.
- Yule, G.U. (1927). On a method of investigating periodicities in disturbed series with special reference to Wolfer's Sunspot Numbers. *Phil. Trans. R. Soc. Lond.*, A226, 267-298.
- Zucchini, W., & MacDonald, I. L. (2009). *Hidden Markov Models for Time Series: An Introduction using R*. Boca Raton: CRC Press.

Index

- ACF, 18, 21
large sample distribution, 28, 488
multidimensional, 36
of an AR(p), 97
of an AR(1), 80
of an AR(2), 93
of an ARMA(1,1), 97
of an MA(q), 96
sample, 27
- AIC, 51, 144, 206
multivariate case, 272
- AICc, 51, 144
multivariate case, 272
- Aliasing, 9, 169
- Amplitude, 168
of a filter, 216
- Analysis of Power, *see* ANOPOW
- ANOPOW, 394, 402, 403
designed experiments, 407
- APARCH, 259
- AR model, 11, 78
conditional sum of squares, 119
bootstrap, 130
conditional likelihood, 119
estimation
large sample distribution, 116, 498
likelihood, 119
maximum likelihood estimation, 118
missing data, 379
operator, 79
polynomial, 87
spectral density, 178
unconditional sum of squares, 119
- vector, *see* VAR
with observational noise, 291
- ARCH model
ARCH(p), 257
ARCH(1), 254
Asymmetric power, 259
estimation, 255
GARCH, 258
- ARFIMA model, 242, 246
- ARIMA model, 134
fractionally integrated, 246
multiplicative seasonal models, 152
multivariate, 271
- ARMA model, 85
 ψ -weights, 95
conditional least squares, 121
pure seasonal models
behavior of ACF and PACF, 150
unconditional least squares, 121
backcasts, 113
behavior of ACF and PACF, 101
bootstrap, 326
causality of, 87
conditional least squares, 123
forecasts, 109
mean square prediction error, 110
based on infinite past, 109
prediction intervals, 112
truncated prediction, 111
Gauss–Newton, 123
in state-space form, 321
invertibility of, 88

- large sample distribution of estimators, 127
- likelihood, 120
- MLE, 121
- multiplicative seasonal model, 150
- pure seasonal model, 148
- unconditional least squares, 123
- vector, *see* VARMA model
- ARMAX model, 221, 279, 320
 - bootstrap, 326
 - in state-space form, 321
- ARX model, 273
- Autocorrelation function, *see* ACF
- Autocovariance
 - calculation, 17
- Autocovariance function, 16, 21, 79
 - multidimensional, 35
 - random sum of sines and cosines, 169
 - sample, 27
- Autocovariance matrix, 33
 - sample, 34
- Autoregressive Integrated Moving Average Model, *see* ARIMA model
- Autoregressive models, *see* AR model
- Autoregressive Moving Average Models, *see* ARMA model
- Backcasting, 113
- Backshift operator, 58
- Bandwidth, 192
- Bartlett kernel, 202
- Beam, 398
- Best linear predictor, *see* BLP
- BIC, 52, 144, 206
 - multivariate case, 272, 275
- BLP, 103
 - m*-step-ahead prediction, 107
 - mean square prediction error, 107
 - one-step-ahead prediction, 104
 - definition, 103
 - one-step-ahead prediction
 - mean square prediction error, 104
 - stationary processes, 103
- Bone marrow transplant series, 289, 314
- Bonferroni inequality, 197
- Bootstrap, 130, 193, 205, 326
 - stochastic volatility, 361
- Bounded in probability O_p , 474
- Brownian motion, 251
- Cauchy sequence, 491
- Cauchy–Schwarz inequality, 471, 491
- Causal, 81, 87, 495
 - conditions for an AR(2), 89
 - vector model, 280
- CCF, 18, 23
 - large sample distribution, 30
 - sample, 30
- Central Limit Theorem, 479
 - M-dependent, 480
- Cepstral analysis, 234
- Characteristic function, 476
- Chernoff information, 430
- Chicken prices, 56
- Cluster analysis, 434
- Coherence, 209
 - estimation, 211
 - hypothesis test, 211, 521
 - multiple, 391
- Completeness of L^2 , 472
- Complex normal distribution, 517
- Complex roots, 94
- Conditional least squares, 121
- Convergence in distribution, 476
 - Basic Approximation Theorem, 477
- Convergence in probability, 473
- Convolution, 177
- Cosine transform
 - large sample distribution, 507
 - of a vector process, 387
 - properties, 186
- Cospectrum, 208
 - of a vector process, 387
- Cramér–Wold device, 477
- Cross-correlation function, *see* CCF
- Cross-covariance function, 18
 - sample, 30
- Cross-spectrum, 208
- Cycle, 168
- Daniell kernel, 199, 200
 - modified, 199, 200
- Deconvolution, 405
- Density function, 15
- Designed experiments, *see* ANOPOW
- Deterministic process, 501
- Detrending, 47
- DFT, 171
 - inverse, 182

- large sample distribution, 507
- multidimensional, 228
- of a vector process, 387
 - likelihood, 387
- Differencing, 57–59
- Discriminant analysis, 422
- DJIA, *see* Dow Jones Industrial Average, *see also* Dow Jones Industrial Average
- DLM, 288, 319
 - Bayesian approach, 365
 - bootstrap, 326
 - innovations form, 325
 - maximum likelihood estimation
 - large sample distribution, 310
 - via EM algorithm, 308, 313
 - via Newton-Raphson, 302
 - MCMC methods, 371
 - observation equation, 288
 - state equation, 288
 - steady-state, 310
 - with switching, 345
 - EM algorithm, 351
 - maximum likelihood estimation, 351
- DNA series, 455, 459
- Dow Jones Industrial Average, 4
- Durbin–Levinson algorithm, 105
- Earthquake series, 7, 385, 419, 426, 431, 436
- EM algorithm, 306
 - complete data likelihood, 306
 - DLM with missing observations, 313
 - expectation step, 307
 - maximization step, 307
- Explosion series, 7, 385, 419, 426, 431, 436
- Exponentially Weighted Moving Averages, 136
- Factor analysis, 443
 - EM algorithm, 445
- Fejér kernel, 202
- FFT, 172
- Filter, 59
 - amplitude, 216, 217
 - band-pass, 226
 - design, 226
 - high-pass, 213, 226
 - linear, 213
 - low-pass, 213, 226
 - matrix, 217, 218
- optimum, 224
- phase, 216, 217
- recursive, 226
- seasonal adjustment, 226
- spatial, 228
- time-invariant, 472
- fMRI, *see* Functional magnetic resonance imaging series
- Folding frequency, 169, 172
- Fourier frequency, 172, 182
- Fractional difference, 60, 242
 - fractional noise, 242
- Frequency bands, 176, 191
- Frequency response function, 177
 - of a first difference filter, 213
 - of a moving average filter, 213
- Functional magnetic resonance imaging series, 6, 384, 409, 411, 415, 441, 447
- Fundamental frequency, 171, 172, 182
- Glacial varve series, 61, 125, 143, 244, 253
- Global temperature series, 2, 60, 290
- Gradient vector, 303, 378
- Growth rate, 137, 253
- Harmonics, 196
- Hessian matrix, 303, 378
- Hidden Markov Model, *see* HMM
- Hidden Markov model, 334, 349
 - estimation, 351
- Hilbert space, 491
 - closed span, 492
 - conditional expectation, 494
 - projection mapping, 492
 - regression, 493
- HMM, 345
 - Poisson, 335, 339
- Homogeneous difference equation
 - first order, 91
 - general solution, 92
 - second order, 91
 - solution, 92
- Impulse response function, 177
- Influenza series, 263, 352
- Infrasound series, 398, 400, 403, 406
- Inner product space, 491
- Innovations, 140, 302
 - standardized, 140

- steady-state, 310
- Innovations algorithm, 108
- Integrated models, 133, 136, 152
 - forecasting, 135
- Interest rate and inflation rate series, 327
- Invertible, 85
 - vector model, 280
- J-divergence measure, 434
- Johnson & Johnson quarterly earnings series, 2, 316
- Joint distribution function, 14
- Kalman filter, 293
 - correlated noise, 320
 - innovations form, 325
 - Riccati equation, 309
 - stability, 309
 - with missing observations, 312
 - with switching, 347
 - with time-varying parameters, 295
- Kalman smoother, 297, 376
 - as a smoothing spline, 331
 - for the lag-one covariance, 300
 - spline smoothing, 332
 - with missing observations, 312
- Kronecker's Lemma, 528
- Kullback-Leibler information, 73, 429
- Kurtosis, 357
- LA Pollution – Mortality Study, 52, 71, 146, 273, 275, 322
- Lag, 17, 24
- Lag window estimator, 204
- Lagged regression model, 265
- Lake Shasta series, 383, 388, 394
- Lead, 24
- Leakage, 203
 - sidelobe, 203
- Least squares estimation, *see* LSE
- Likelihood
 - AR(1) model, 119
 - conditional, 119
 - innovations form, 120, 302
- Linear filter, *see* Filter
- Linear process, 25, 87
- Ljung–Box–Pierce statistic, 141
 - multivariate, 276
- Local level model, 296, 298, 368
- Long memory, 60, 242
 - estimation, 243
 - estimation of d , 248
 - spectral density, 247
- LSE
 - conditional sum of squares, 119
 - Gauss–Newton, 122
 - unconditional, 119
- MA model, 9, 83
 - autocovariance function, 17, 96
 - Gauss–Newton, 124
 - mean function, 15
 - operator, 83
 - polynomial, 87
 - spectral density, 178
- Maximum likelihood estimation, *see* MLE
- Mean function, 15
- Mean square convergence, 471
- Method of moments estimators, *see* Yule–Walker
- Minimum mean square error predictor, 102
- Missing data, 313
- MLE
 - ARMA model, 121
 - conditional likelihood, 119
 - DLM, 302
 - state-space model, 302
 - via EM algorithm, 306
 - via Newton–Raphson, 121, 302
 - via scoring, 121
- Moving average model, *see* MA model
- New York Stock Exchange, 360
- Newton–Raphson, 121
- Non-negative definite, 22
- Normal distribution
 - marginal density, 15
 - multivariate, 25, 517
- NYSE, 462
- Order in probability o_p , 474
- Ordinary Least Squares, 48
- Orthogonality property, 492
- PACF, 99
 - of an MA(1), 101
 - iterative solution, 106
 - large sample results, 116

- of an AR(p), 100
- of an AR(1), 100
- of an MA(q), 101
- Parameter redundancy, 86
- Partial autocorrelation function, *see* PACF
- Period, 167
- Periodogram, 172, 182
 - distribution, 188
 - matrix, 429
- Phase, 168
 - of a filter, 216
- Pitch period, 4
- Prediction equations, 103
- Prewhiten, 32, 267
- Principal components, 438
- Projection Theorem, 492
- Q-test, *see* Ljung–Box–Pierce statistic
- Quadspectrum, 208
 - of a vector process, 387
- Random sum of sines and cosines, 169, 503, 505
- Random walk, 11, 15, 20, 135
 - autocovariance function, 18
- Recruitment series, 5, 31, 62, 101, 112, 189, 193, 200, 212, 220, 267
- Regression
 - ANOVA table, 50
 - autocorrelated errors, 145, 322
 - Cochrane–Orcutt procedure, 146
 - coefficient of determination, 51
 - for jointly stationary series, 388
 - ANOPOW table, 394
 - Hilbert space, 493
 - lagged, 218
 - model, 47
 - multivariate, 271, 322
 - normal equations, 49
 - random coefficients, 405
 - spectral domain, 388
 - stochastic, 327, 405
 - ridge correction, 406
 - with deterministic inputs, 397
- Return, 4, 137, 253, 254
 - log-, 254
- Riesz–Fischer Theorem, 472
- Scatterplot matrix, 54, 62
- Scatterplot smoothers
 - kernel, 68
 - lowess, 69, 71
 - nearest neighbors, 69
 - splines, 70
- Score vector, 303
- SIC, 52
- Signal plus noise, 12, 14, 223, 398
 - mean function, 16
- Signal-to-noise ratio, 13, 224
- Sine transform
 - large sample distribution, 507
 - of a vector process, 387
 - properties, 186
- Smoothing splines, 70, 331
- Soil surface temperature series, 35, 36, 229
- Southern Oscillation Index, 5, 31, 62, 189, 193, 200, 203, 206, 212, 213, 220, 225, 267
- Spectral density, 175
 - autoregression, 206, 529
 - estimation, 191
 - adjusted degrees of freedom, 193
 - bandwidth stability, 198
 - confidence interval, 192
 - degrees of freedom, 192
 - large sample distribution, 192
 - nonparametric, 205
 - parametric, 205
 - resolution, 198
 - matrix, 210
 - linear filter, 218
 - of a filtered series, 177
 - of a moving average, 178
 - of an AR(2), 178
 - of white noise, 176
 - wavenumber, 227
- Spectral distribution function, 175
- Spectral envelope, 453
 - categorical time series, 456
 - real-valued time series, 461
- Spectral Representation Theorem, 175, 181, 503, 505
 - vector process, 210, 505
- Speech series, 3, 29
- Spline smoothing, 332
- State-space model
 - Bayesian approach, 365
 - linear, *see* DLM

- Stationary
 - Gaussian series, 25
 - jointly, 23
 - strictly, 19
 - weakly, 20
- Stochastic process, 8
 - realization, 8
- Stochastic regression, 327
- Stochastic trend, 134
- Stochastic volatility model, 358
 - bootstrap, 361
 - estimation, 360
- Structural component model, 72, 315, 352
- Taper, 201, 203
 - cosine bell, 202
- Taylor series expansion in probability, 475
- Tchebycheff inequality, 471
- Time series, 8
 - categorical, 456
 - complex-valued, 437
 - multidimensional, 34, 227
 - multivariate, 19, 33
 - two-dimensional, 228
- Toepeliz Matrix, 526
- Transfer function model, 265
- Transformation
 - Box-Cox, 60
- Trend stationarity, 22
- Triangle inequality, 491
- U.S. GNP series, 138, 141, 144, 257
- U.S. macroeconomic series, 450
- U.S. population series, 144
- Unconditional least squares, 121
- Unit root tests, 250
 - Augmented Dickey-Fuller test, 252
- Dickey-Fuller test, 252
- Phillips-Perron test, 252
- VAR model, 272, 274
 - estimation
 - large sample distribution, 279
 - operator, 280
- Variogram, 37, 43
- VARMA model, 279
 - autocovariance function, 280
 - estimation
 - Spliid algorithm, 282
 - identifiability of, 282
- Varve series, 248
- Viterbi algorithm, 350
- VMA model, 280
 - operator, 280
- Volatility, 4, 253
- Wavenumber spectrum, 227
 - estimation, 228
- Weak law of large numbers, 474
- White noise, 9
 - autocovariance function, 16
 - Gaussian, 9
 - vector, 272
- Whittle likelihood, 207, 428
- Wold Decomposition, 501
- Yule–Walker
 - equations, 115
 - vector model, 277
- estimators, 115
 - AR(2), 116
 - MA(1), 117
- large sample results, 116

Time Series Analysis

Computer Lab B: ARIMA models-4

Model selection, Forecasting

Tohid Ardesthiri

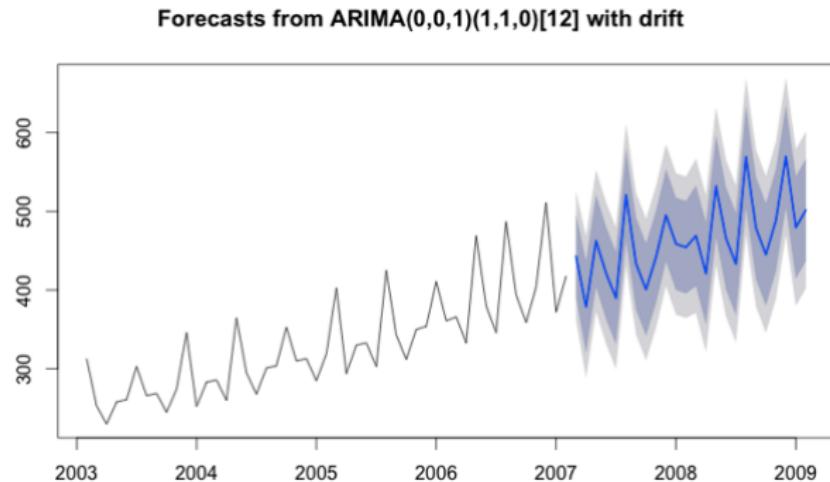
Linköping University
Division of Statistics and Machine Learning

September 23, 2019



Forecasting

- We have our series $x_1 \dots x_n$
- Use series to predict m steps ahead: x_{n+m}^n
- The prediction should be based on our observed data
 $x_{n+m}^n = g(x_1, \dots, x_n)$



Forecasting

- Assume $g(x_1, \dots, x_n) = \alpha_0 + \sum_{k=1}^n \alpha_k x_k$
 - ▶ Best linear predictors
- How to find α 's?

$$\min E[(x_{n+m} - g(x_1, \dots, x_n))^2]$$

- Prediction equations
 - ▶ Find α 's by solving ($x_0 = 1$)
$$E[(x_{n+m} - x_{n+m}^n)x_k] = 0, k = 0, \dots, n$$
- **Note:** $n+1$ equations, $n+1$ unknowns

One-step-ahead

- Denote $x_{n+1}^n = \phi_{n1}x_n + \dots + \phi_{nn}x_1$
- Prediction equations give

$$\Gamma_n \phi_n = \gamma_n$$

$$\Gamma_n = \begin{pmatrix} \gamma(1-1) & \gamma(2-1) & \dots & \gamma(n-1) \\ \gamma(2-1) & \gamma(2-2) & \dots & \gamma(n-2) \\ \dots & \dots & \dots & \dots \\ \gamma(n-1) & \gamma(n-2) & \dots & \gamma(n-n) \end{pmatrix}$$

$$\phi_n = \begin{pmatrix} \phi_{n1} \\ \dots \\ \phi_{nn} \end{pmatrix} \quad \gamma_n = \begin{pmatrix} \gamma_1 \\ \dots \\ \gamma_n \end{pmatrix}$$

- **Note:** for ARMA models Γ_n is positive def \rightarrow unique solution

One-step-ahead

- Causal AR(p): for $n \geq p$ best linear prediction is

$$x_{n+1}^n = \phi_1 x_n + \dots + \phi_p x_{n-p+1}$$

- In general, solve system of equations $\rightarrow O(n^3)$ operations
- Much faster algorithms exist
 - ▶ Durbin-Levinson algorithm
 - ▶ Innovations algorithm
- **Property:** PACF of a stationary process can be obtained as ϕ_{nn} by solving $\Gamma_n \phi_n = \gamma_n$

One-step-ahead

- Mean square prediction error (MSPE)

$$P_{n+1}^n = E[(x_{n+1} - x_{n+1}^n)^2] = \gamma(0) - \gamma_n' \Gamma_n^{-1} \gamma_n$$

- Confidence intervals for x_{n+1}

$$x_{n+1}^n \pm \alpha \sqrt{P_{n+1}^n}$$

- m-step ahead in general? Prediction equations
 - ▶ Difficult in general

m-step-ahead for ARMA

- Assume causal and invertible ARMA(p,q)
- Finite past prediction

$$x_{n+1}^n = E(x_{n+1}|x_n, \dots, x_1)$$

- Infinite past prediction

$$\tilde{x}_{n+m}^n = E(x_{n+m}|x_n, \dots, x_1, x_0, x_{-1}, \dots)$$

- m-step-ahead forecast for infinite past

- ▶ Compute recursively

$$\tilde{x}_{n+m} = - \sum_{j=1}^{m-1} \pi_j \hat{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j \tilde{x}_{n+m-j}, \quad m = 1, 2, \dots$$

- m-step ahead prediction error: $P_{n+m}^n = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2$

Long-range forecasts

- What if $m \rightarrow \infty$?

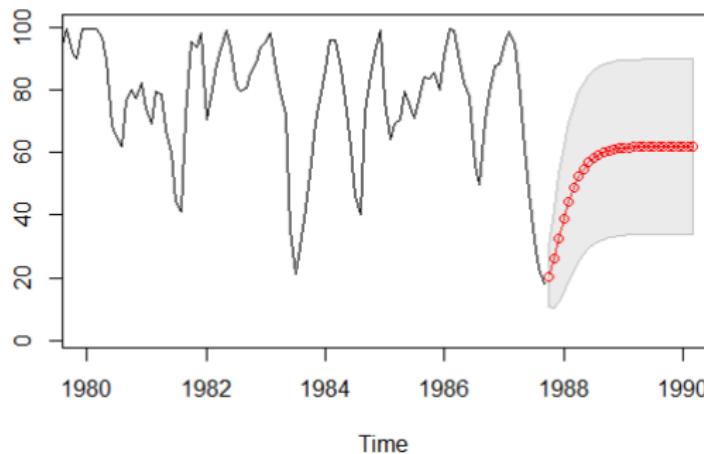
$$\tilde{x}_{n+m} \rightarrow 0(\text{or } \mu)$$

$$P_{n+m}^n \rightarrow \sigma_x^2$$

m-step-ahead

- Recruitment, AR(2)

$$x_{n+m}^n \pm 2\sqrt{P_{n+m}^n}$$



Truncated prediction

- Ignore non-positive j in x_j

$$\tilde{x}_{n+m} = - \sum_{j=1}^{m-1} \pi_j \tilde{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j x_{n+m-j}, m = 1, 2, \dots$$

- For ARMA, truncated prediction formula:
 - Recursive computation, explicit

$$\tilde{x}_{n+m}^n = \phi_1 \tilde{x}_{n+m-1}^n + \dots + \phi_p \tilde{x}_{n+m-p}^n + \theta_1 \tilde{w}_{n+m-1}^n + \dots + \theta_q \tilde{w}_{n+m-q}^n$$

$$\tilde{w}_t^n = \tilde{x}_t^n - \phi_1 \tilde{x}_{t-1}^n - \dots - \phi_p \tilde{x}_{t-p}^n - \theta_1 \tilde{w}_{t-1}^n - \dots - \theta_q \tilde{w}_{t-q}^n$$

- Boundary conditions: $\tilde{x}_t^n = x_n, 1 \leq t \leq n, \tilde{x}_t^n = 0, t \leq 0$

$$\tilde{w}_t^n = 0, t \leq 0 \quad \text{or } t > n$$

Model selection

- ARIMA models

$$\phi(B)(1 - B)^d x_t = \theta(B) w_t$$

- What is p, d, q in ARIMA(p,d,q)?

Step 1: Check ACF, PACF and EACF to define a few tentative models

Model selection

- Step 2: Fit the tentative models, compare them
 - ▶ Analytical measures: AIC, BIC
 - ★ Penalize models with many parameters → simpler models
 - ▶ Residual analysis
- Akaike Information Criterion (AIC)

$$AIC = -2 \log(L) + 2k$$

$$k = p + q \text{ or } k = p + q - 1 \text{ (intercept)}$$

- Corrected Akaike Information Criterion (AICc)

$$AIC_c = AIC + \frac{2(k+1)(k+2)}{n-k-2}$$

- Bayesian information criterion (BIC)

$$BIC = -2 \log(L) + k \log(n)$$

Model selection

- **Example:** GNP data
 - ▶ Fitting ARIMA(1,1,0) to log(gmp)
 - ▶ Write down equation of the model

Coefficients:

	ar1	constant
s.e.	0.3467	0.0083
s.e.	0.0627	0.0010

sigma^2 estimated as 9.03e-05: log likelihood = 718.61, aic = -1431.22

\$degrees_of_freedom
[1] 221

\$ttable

	Estimate	SE	t.value	p.value
ar1	0.3467	0.0627	5.5255	0
constant	0.0083	0.0010	8.5398	0

\$AIC
[1] -8.294483

\$AICC
[1] -8.285023

\$BIC
[1] -9.263925

Model selection

- **Example:** GNP data
 - ▶ Fitting ARIMA(0,1,2) to log(gmp)
 - ▶ Write down equation of the model

```
Coefficients:
            ma1      ma2  constant
0.3028  0.2035  0.0083
s.e.    0.0654  0.0644  0.0010

sigma^2 estimated as 8.919e-05:  log likelihood = 719.96,  aic = -1431.93

$degrees_of_freedom
[1] 220

$ttable
        Estimate      SE t.value p.value
ma1     0.3028  0.0654  4.6272  0.0000
ma2     0.2035  0.0644  3.1593  0.0018
constant 0.0083  0.0010  8.7177  0.0000

$AIC
[1] -8.297814

$AICC
[1] -8.288023

$BIC
[1] -9.251978
```

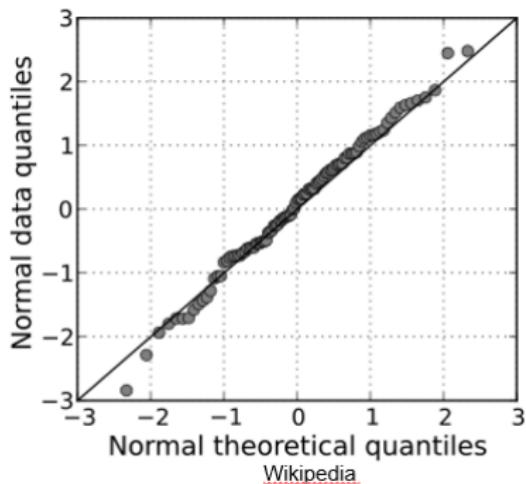
Which model is optimal according to AIC, AICc and BIC?

Residual analysis

- Residuals $e_t = x_t - \hat{x}_t^{t-1}$? they are innovations
 - ▶ Note: computed from one-step-ahead predictions!
 - ▶ Measures predictive quality of the model (compare OLS)
- Residual analysis
 - ▶ Visual inspection: stationary? Patterns?
 - ▶ Histograms, Q-Q plots
 - ▶ ACF, PACF
 - ▶ Runs test
 - ▶ Box-Ljung test

Q-Q plots

- ① Sort data
- ② For each x_t , compute $f_k = \frac{\#(x_i \leq x_k)}{n}$
- ③ For each x_k , compute $g_k = p_N^{-1}(f_k)$
- ④ Plot (g_k, x_k)



If ECDF reminds normal, quantiles should coincide? straight line

Runs test

- Used to test independence
- H_0 : x_t values are i.i.d.
- H_a : x_t values are not i.i.d.
- Idea:
 - ▶ Count amount of segments (runs) where $x_t > \text{median}(x_t)$
 - ▶ If the amount of segments large \rightarrow negative dependence
 - ▶ If the amount of segments small \rightarrow positive dependence
 - ▶ Medium? \rightarrow independence

Box-Ljung test

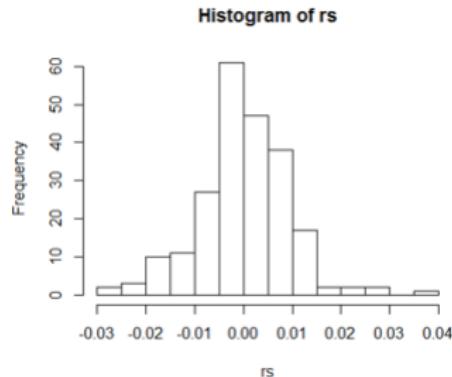
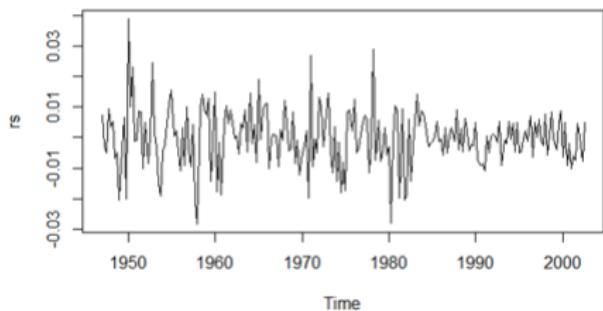
- Do we observe a white noise if each ACF value below threshold?
 - ▶ Many of them just below threshold?
- H_0 : data are independent
- H_a : data are not independent

$$Q = n(n+2) \sum_{h=1}^H \frac{\rho_e^2(h)}{n-h}$$

- Test with different $H \rightarrow$ almost all Q-values are large when reject

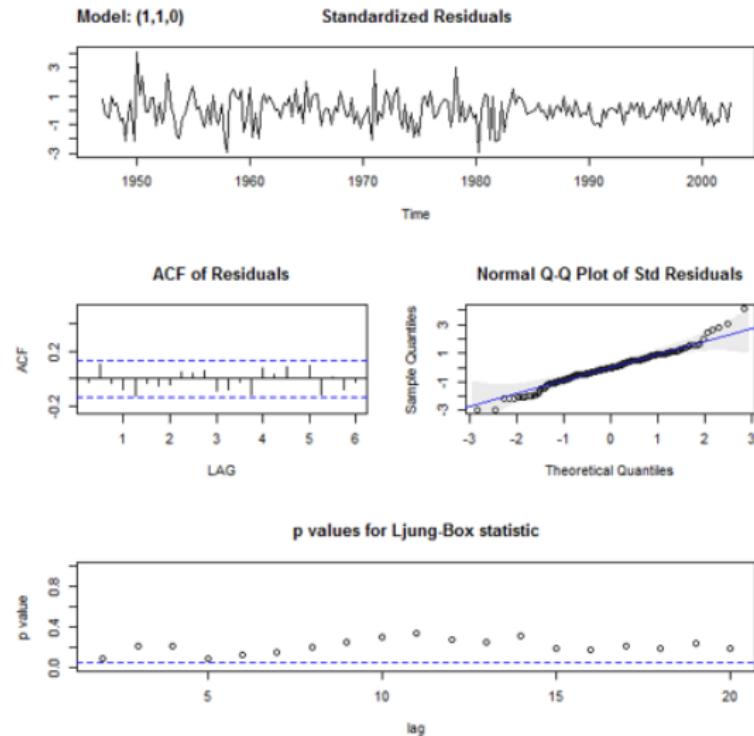
Residual analysis

- **Example:** GNP data
 - ▶ Fitting ARIMA(1,1,0) to log(gmp)
 - ▶ Histogram and visual inspection



Residual analysis

- Example: GNP data

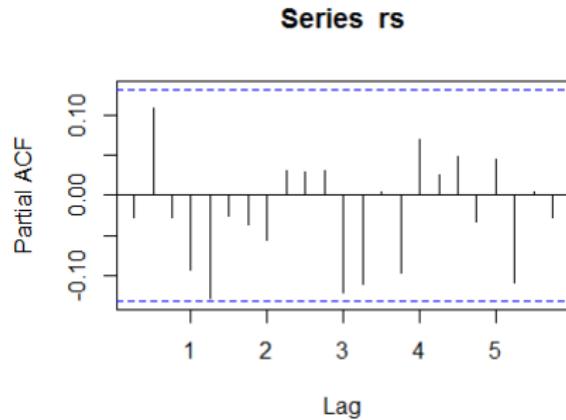


Conclusions?

Residual analysis

- Example: GNP data

Conclusions?



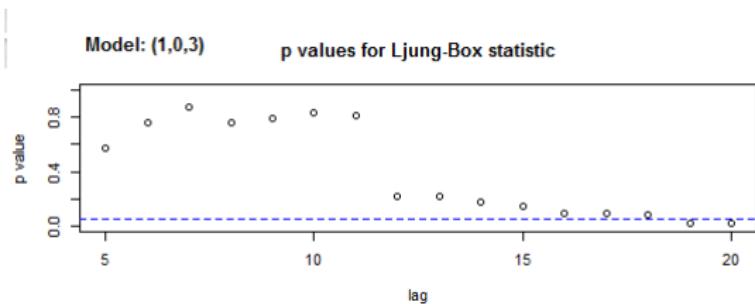
```
> TSA::runs(rs)  
$pvalue  
[1] 0.416
```

Overfitting

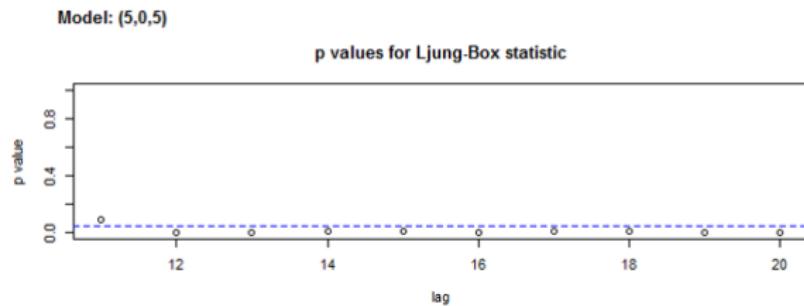
- Occams razor: among equally good models, choose the simplest one
- Overfitting: taking too complex models leads to bad predictions
- If ARIMA(p, d, q) has almost the same predictive quality as ARIMA(p', d', q') , take the one with less parameters

Overfitting

- Example: Recruitment series
 - ▶ Fit ARIMA(1,0,3) and ARIMA(5,0,5)



- Conclusions?



Read home

- Ch 3.7-3.9
- R code: sarima, sarima.for, runs

Course information for 732A62 – Time Series Analysis

Learning activities

The course consists of

- 10 lectures,
- 6 computer lab sessions,
- 3 teaching sessions and,
- a final examination which is computer-based but pen and paper derivations will be necessary, and these derivations are graded along with the code.

Lectures

Lectures will consist of both presentations on screen as well as whiteboard derivations. Presentation files will be uploaded to LISAM to the folder *Course Documents* in pdf format few days before each lecture. Lecture titles are roughly as listed below:

1. Introduction
2. Exploratory analysis and Time Series Regression
3. Introduction to ARIMA
4. ARIMA models 1: Difference equations, Forecasting
5. ARIMA models 2: Forecasting, Estimation, ARIMA, Model selection
6. ARIMA models 3: Model selection, Seasonal models
7. State Space Models: Filtering and smoothing
8. State Space Models: Learning Linear state space models, stochastic volatility
9. Deep learning methods: RNN
10. Summary

Computer labs

The students are suggested to do the computer labs in groups of 2. Students are recommended to work independently on each assignment and discuss the solutions with their partner after they have tried on their own. Finally, students should compile a report jointly. Students are welcome to have discussions within a group during the laboratory. Attendance at the lab sessions is not mandatory but it might be difficult to complete the lab without supervision. Only one of the group members is expected to submit the lab report via the functionality “Submit” of the respective computer lab in LISAM/Submissions. **Attention: there is a deadline for each computer lab report!**

The file should be named *Group X.pdf* (where X is your group number)

The document should clearly state the names of the students that participated in its compilation and a short description of how each student contributed to the report.

**Passed computer lab reports will earn each student 1 ECTS credits with grade pass/fail.
Computer labs may be done using the programming language R. Other programming languages are not supported by the teaching staff.**

Teaching sessions

At the teaching session, the teacher presents solutions of some exercises on the whiteboard. Similar exercises are given to the students for self-studies. Selected exercises are required to be solved by each student and handed in via LISAM. **Passed hand-in assignments will earn each student 1 ECTS credit.** The exam problems will not be far from such problems. Attendance of the teaching sessions is not mandatory.

Missing submission deadline

Missing deadlines without a reasonable cause is not recommended. The course examiner may defer correction of late submissions to future examination rounds.

Office hours

Course teacher, Dr. Tohid Ardestiri, will be available every Thursday morning 09:00-12:00 at his office room (RUM 3E:485) to answer your possible questions in person. Otherwise, you can send an email to tohid.ardeshiri@liu.se.

Examination

The completed hand-in assignments and passed computer lab report will earn each student 2 ECTS credits where the grading is pass/fail.

The final computer-based examination is graded A-F and earns a successful student 4 ECTS credits. In total the course offers a maximum of 6 ECTS credits.

Students may bring any hand-written and printed material to the computer-based examination. The total number of pages of such aid material may not exceed 2000 pages. No online aid or resource such as phones and tablets may be used during the examination.

To succeed in the exam, a student should have

- read the course theory,
- solved the assignments given as take-home,
- completed all computer lab tasks,
- and be able to interpret and use the printouts of the software R.

Time Series Analysis

Lecture 1: Introduction

Tohid Ardeshtiri

Linköping University
Division of Statistics and Machine Learning

September 2, 2019



Course teacher

Tohid Ardeshiri

PhD in 2015 in Bayesian inference



LINKÖPING
UNIVERSITY



UNIVERSITY OF
CAMBRIDGE

Senior Data Scientist at
Qamcom Research & Technology AB



Linköping Studies in Science and Technology. Dissertations.
No. 1710

Analytical
Approximations for
Bayesian Inference

Tohid Ardeshiri



LINKÖPING
UNIVERSITY

Bayesian Inference

Bayesian inference is a means of combining prior beliefs with the data (evidence) to obtain posterior beliefs.

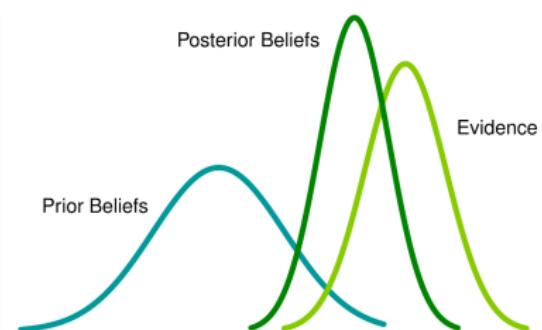
Example: Parameter learning

$$f(\theta|x) \propto f(x|\theta)f(\theta)$$

Probability Calculus

$$f(\theta, x) = f(x|\theta)f(\theta)$$

$$f(\theta, x) = f(\theta|x)f(x)$$



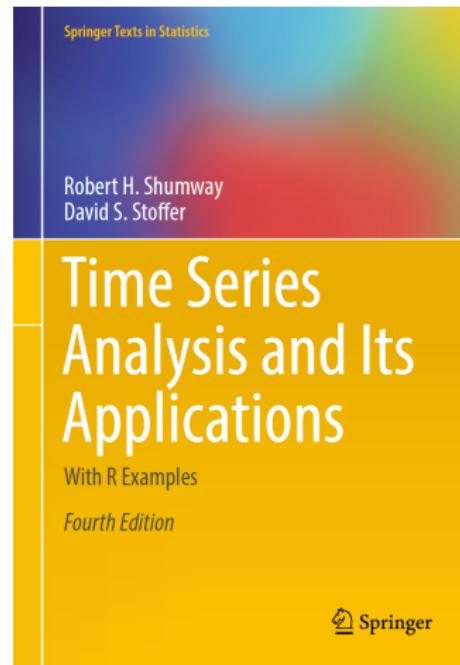
Course literature and software

Course literature:

Time series Analysis and its Applications
Can be downloaded freely here:

<https://www.stat.pitt.edu/stoffer/tsa4/tsa4.pdf>

Software for computer labs is R:



Sequential data



Sequential data: Motion of a ball



Sequential data: A sentence

This is a sequential data type.

Sequential data: A sentence

This is a sequential data type.

This is a sequential data type .

Sequential data: A sentence or a word

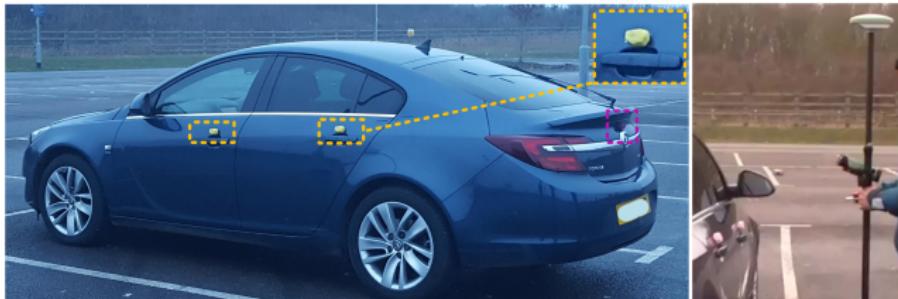
This is a sequential data type.

This is a sequential data type .

s e q u e n t i a l

A look at real data

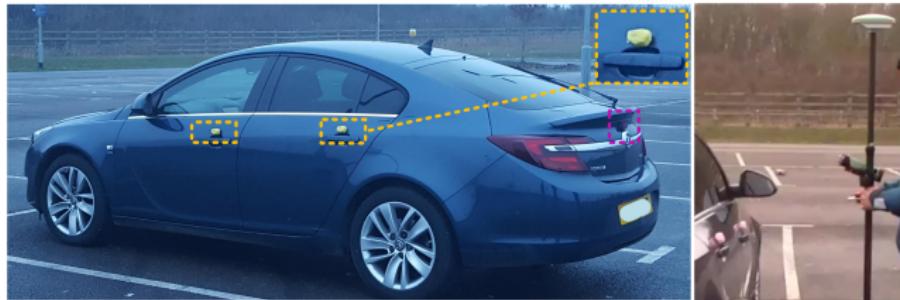
Received signal strength indicator (RSSI) is a common observation (data).



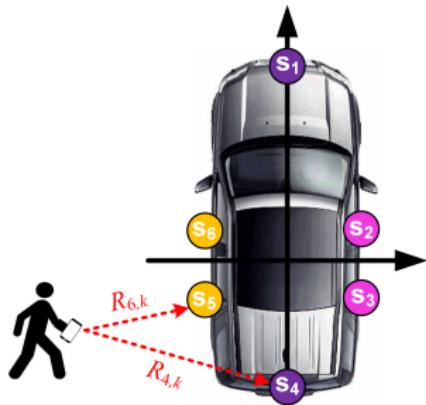
Where is the driver?

A look at real data

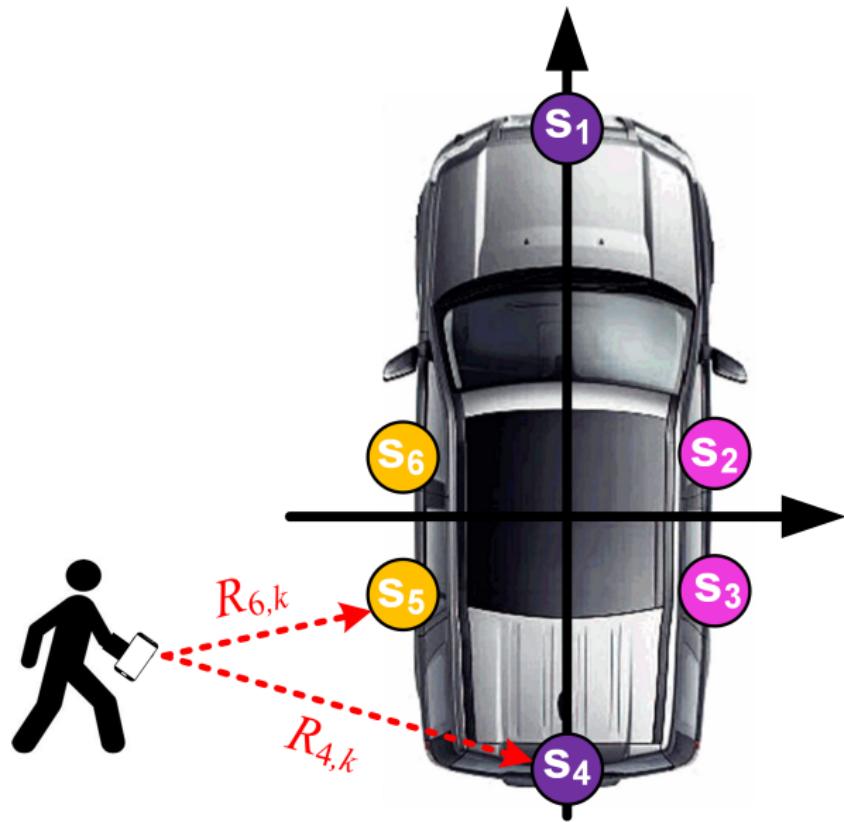
Received signal strength indicator (RSSI) is a common observation (data).



Where is the driver?



Where is the driver?



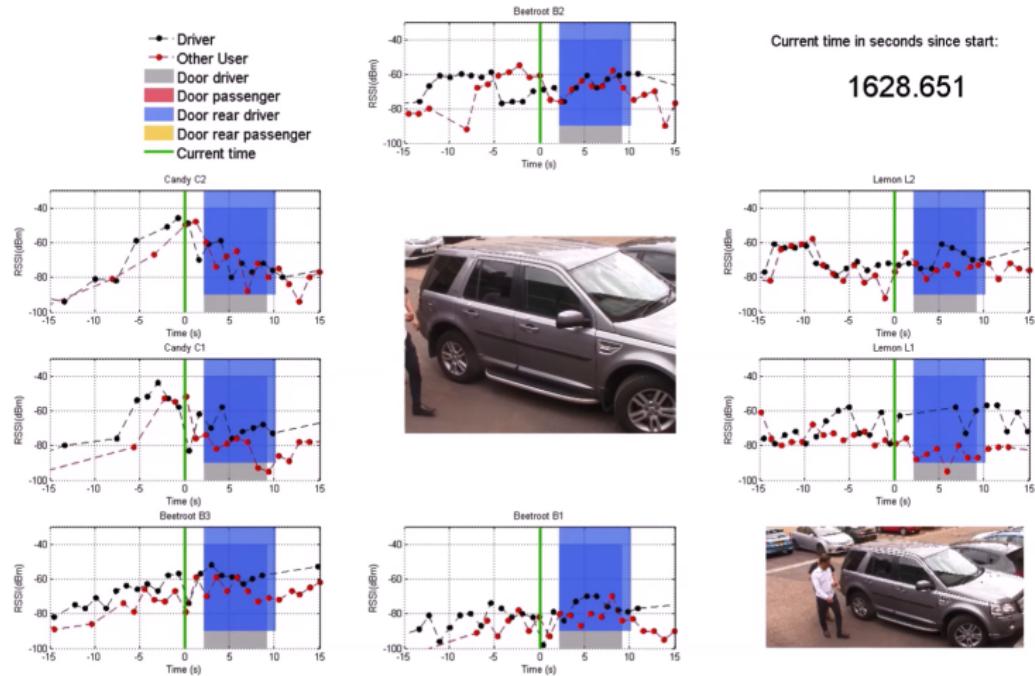
Where is the driver?

Video of data collection



Where is the driver?

Animation of the of signals



Current time in seconds since start:

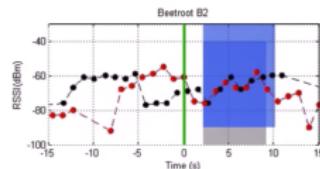
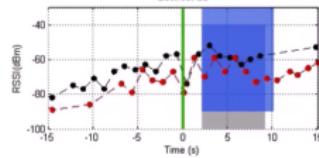
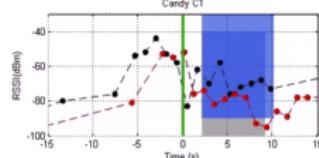
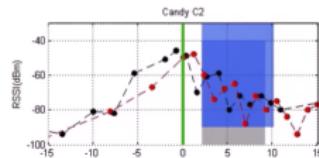
1628.651

Video is Proprietary to Cambridge/Tohid Ardestiri

Time Series Analysis

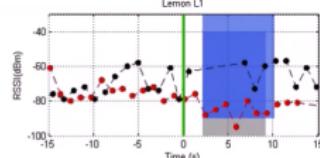
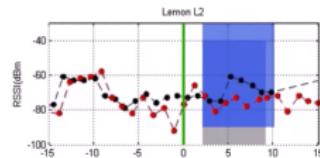
What is a Time Series?

- A sequential data where observations are collected over time
- Observations are typically **correlated!**



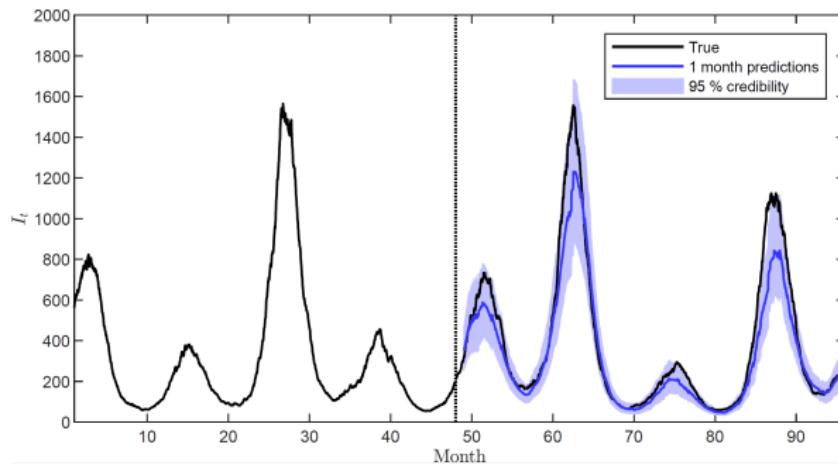
Current time in seconds since start:

1628.651



Time Series Analysis

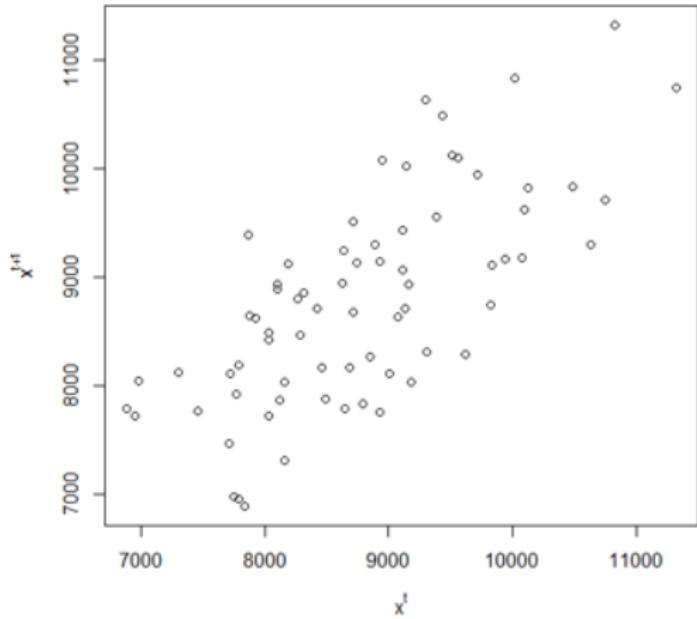
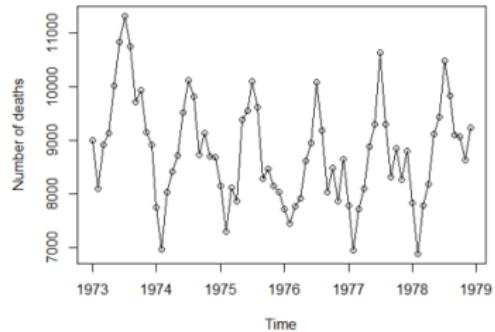
- Understand the properties of the underlying process
- Be able to predict (forecast) possible future values
- Reason about the **uncertainties** in the predictions
requires statistical methods!



Time Series Analysis

Usual regression analysis: observations are often **iid.**

Time Series Analysis: observations are **correlated!**

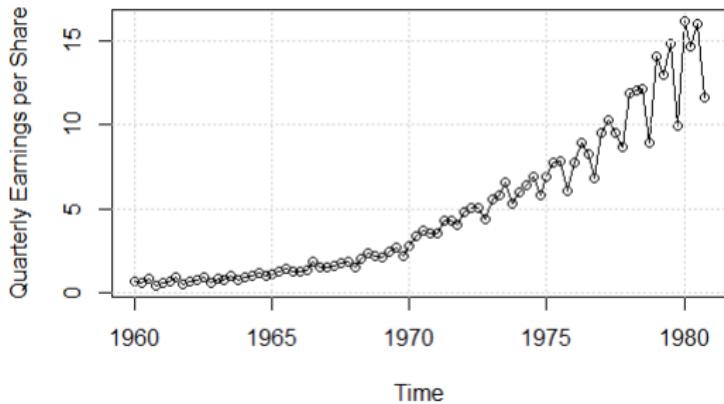


Ex) See connection
between x_t and x_{t+1}

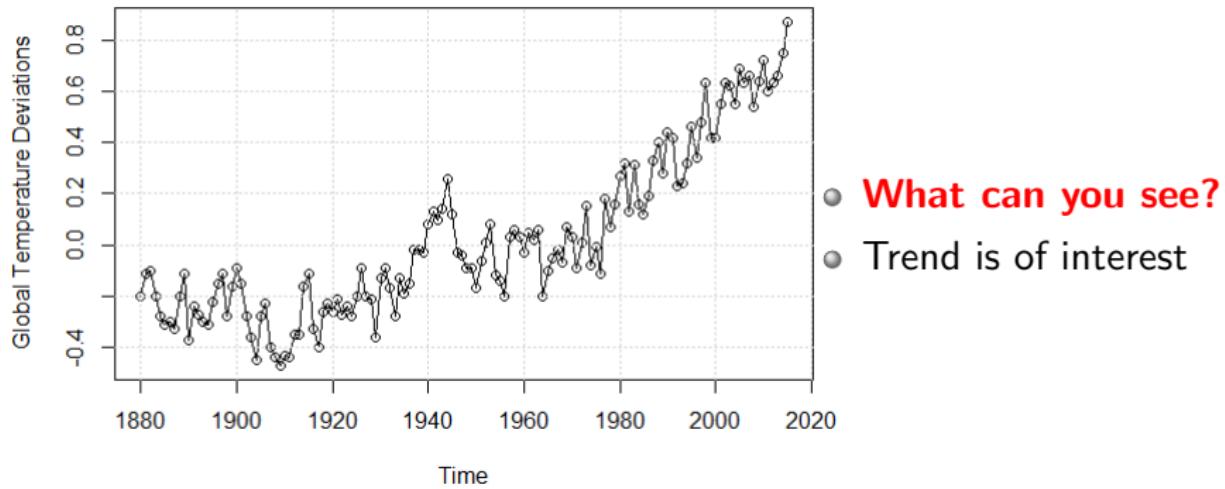
Ex 1: Johnson & Johnson quarterly earnings

- **What can you see?**

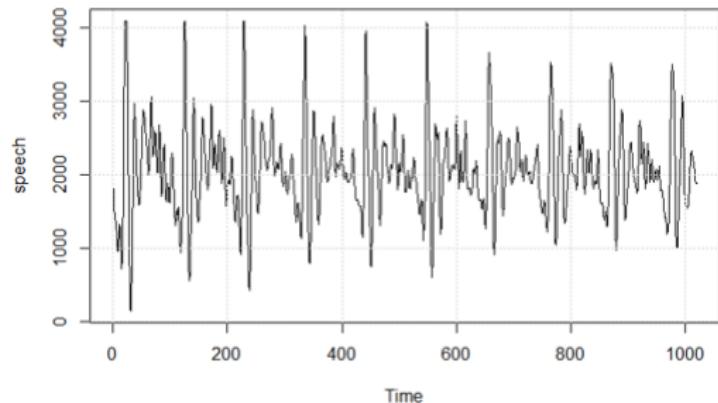
- ▶ Trend?
 - ★ Constant
 - ★ Linear
 - ★ Other
- ▶ Variation?
- ▶ Seasonality?
- ▶ Outliers?



Ex 2: Global warming



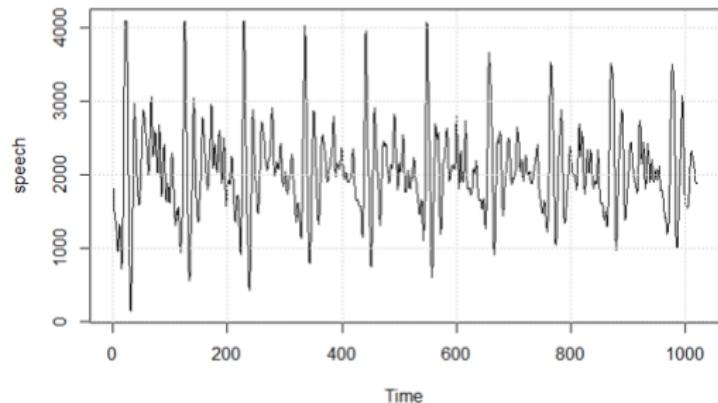
Ex 3: Speech data



- **What can you see?**

Pattern of periodicity is of interest → decompose signal into different frequencies

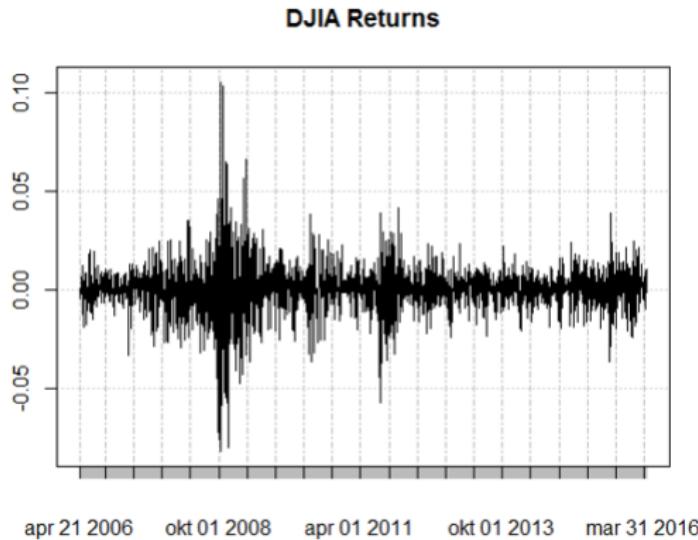
Ex 3: Speech data



- **What can you see?**

Pattern of periodicity is of interest → decompose signal into different frequencies
not covered in this course!

Ex 4: Dow Jones Industrial Average

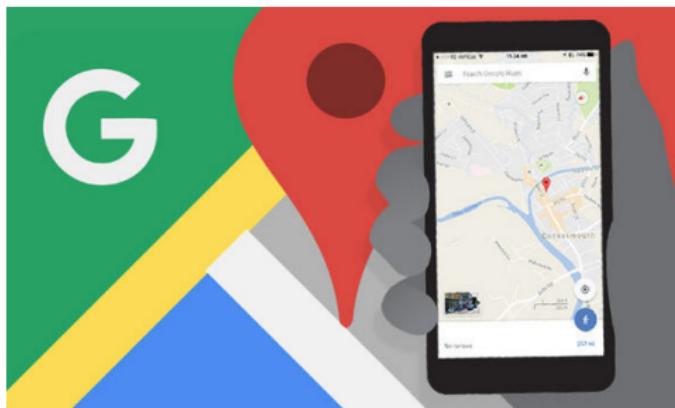
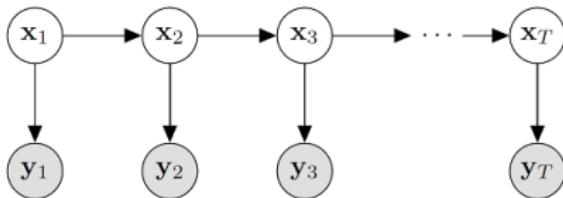


- What can you see here?

Pattern of periodicity is of interest → Stochastic volatility

Ex 5: Dynamical systems

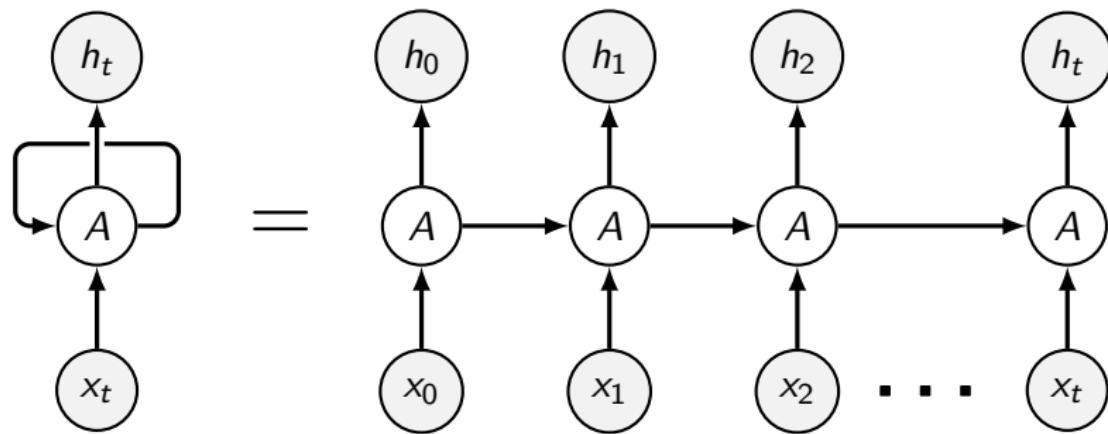
Linear and Gaussian state-space models for tracking objects



Ex 6: Recurrent neural networks

Natural Language Processing

This is a sequential data type .

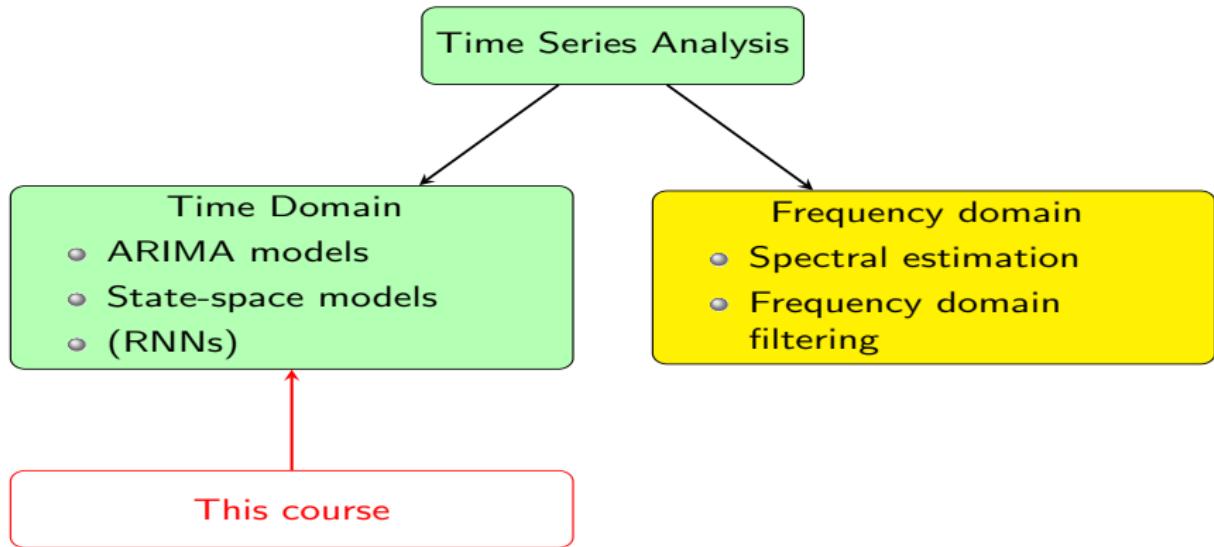


Time Series Analysis

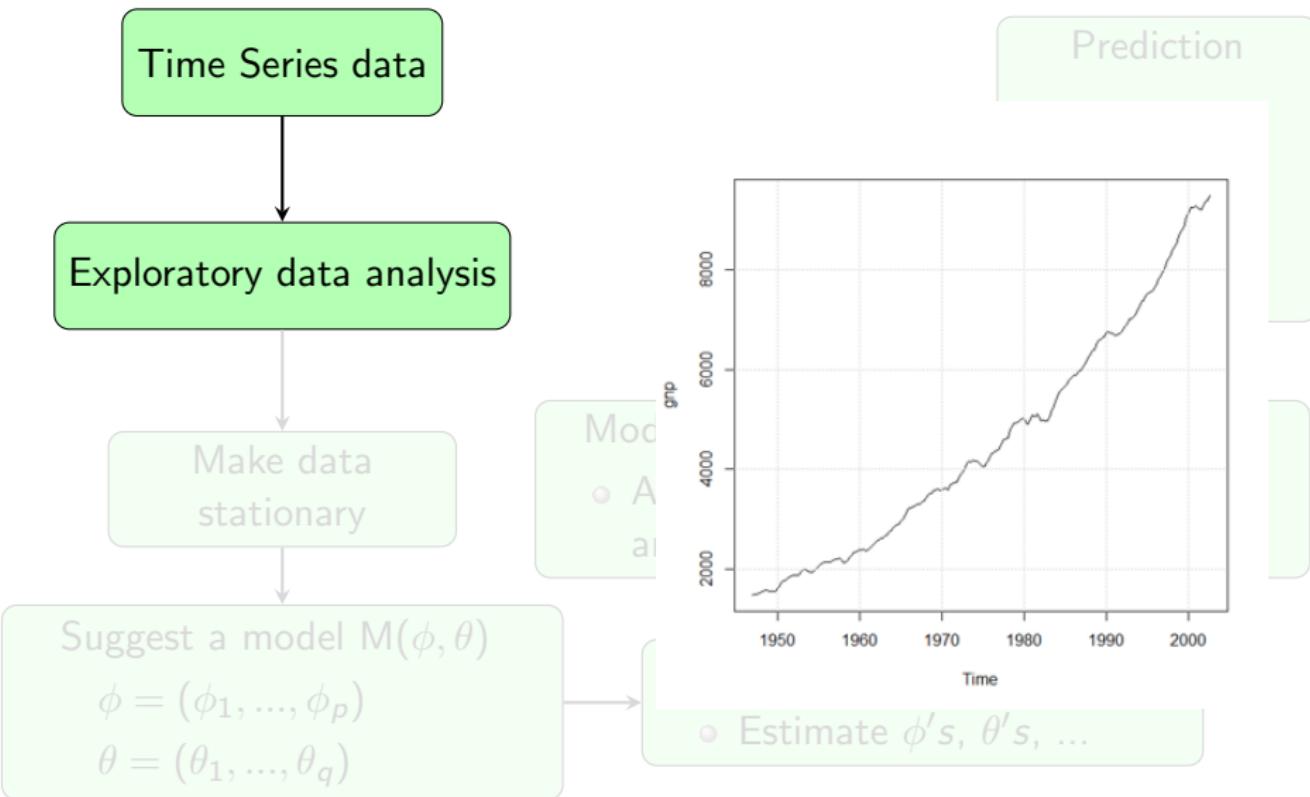
Application areas

- Natural sciences
- Climatology
- Robotics/autonomous systems
- Social sciences
- Medicine
- Economics
- Telecommunications
- ...

The Big Picture



Time domain: The Big Picture



Time domain: The Big Picture

Time Series data

$$Y_t = \nabla(\log(X_t))$$

Prediction

Exploratory data analysis

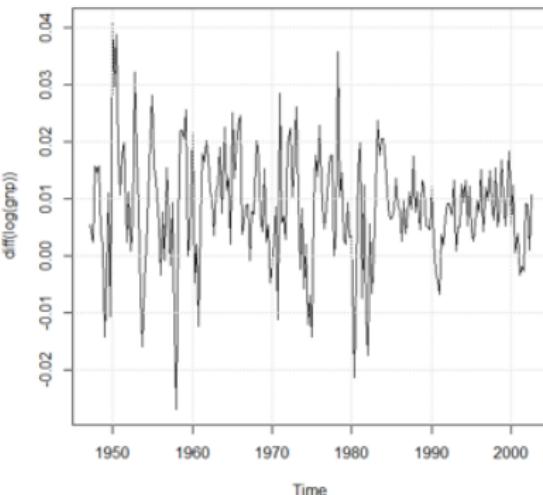
Make data stationary

Suggest a model $M(\phi, \theta)$

$$\phi = (\phi_1, \dots, \phi_p)$$

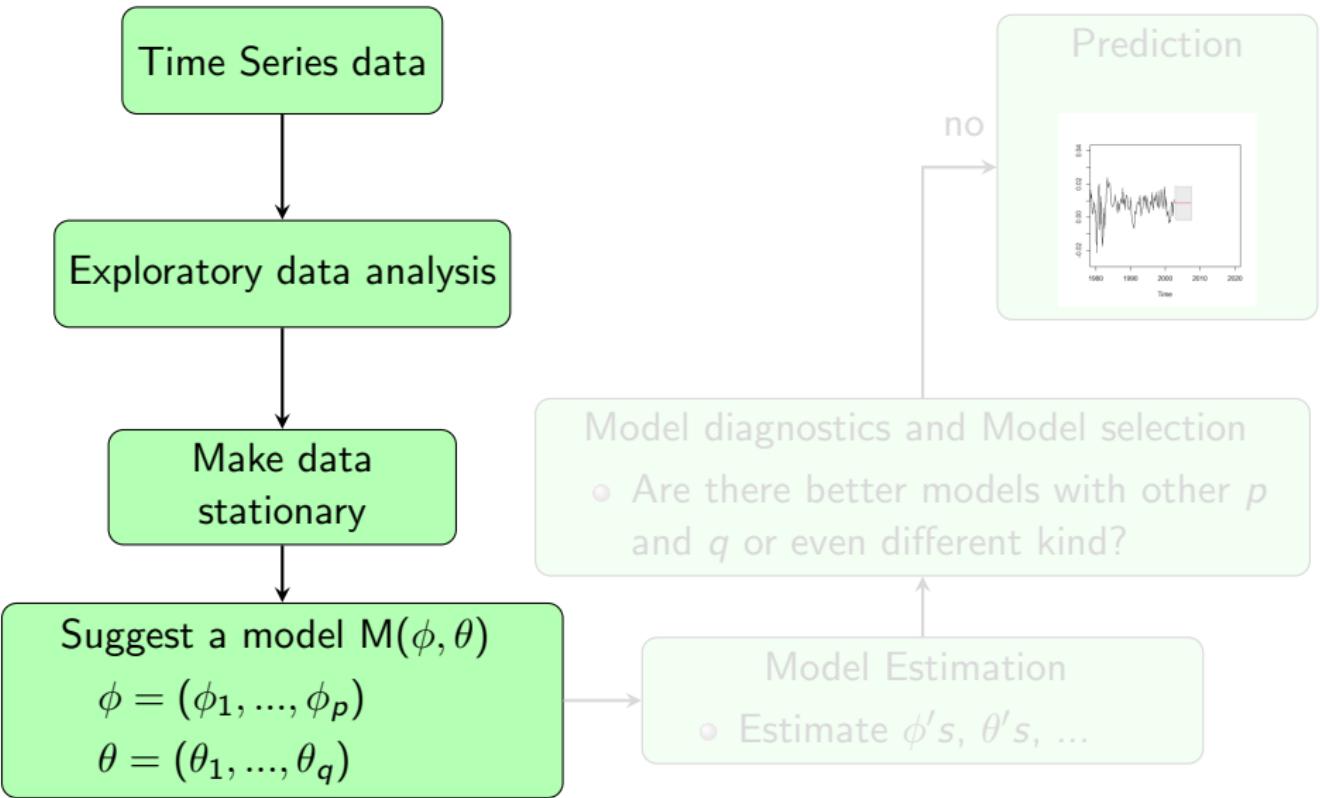
$$\theta = (\theta_1, \dots, \theta_q)$$

Model
A
ai

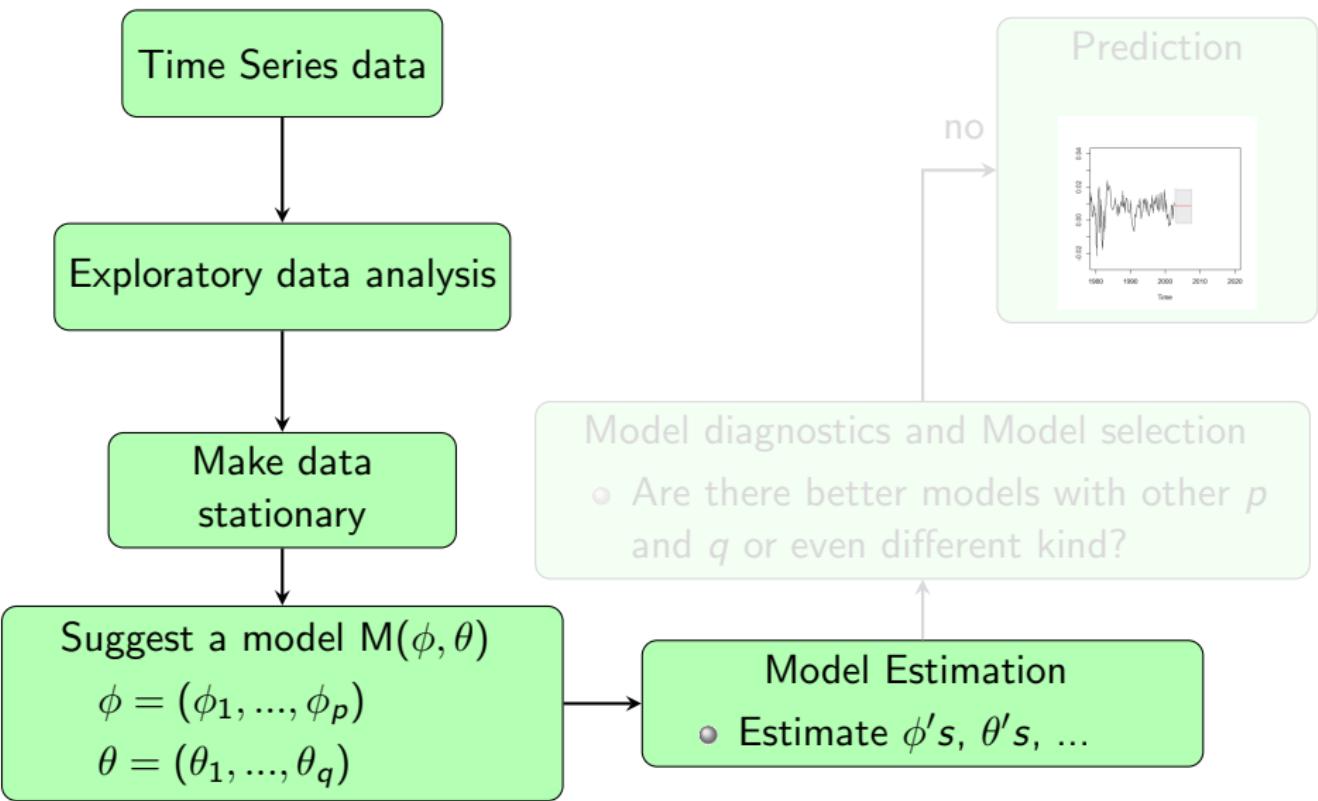


Estimate ϕ 's, θ 's, ...

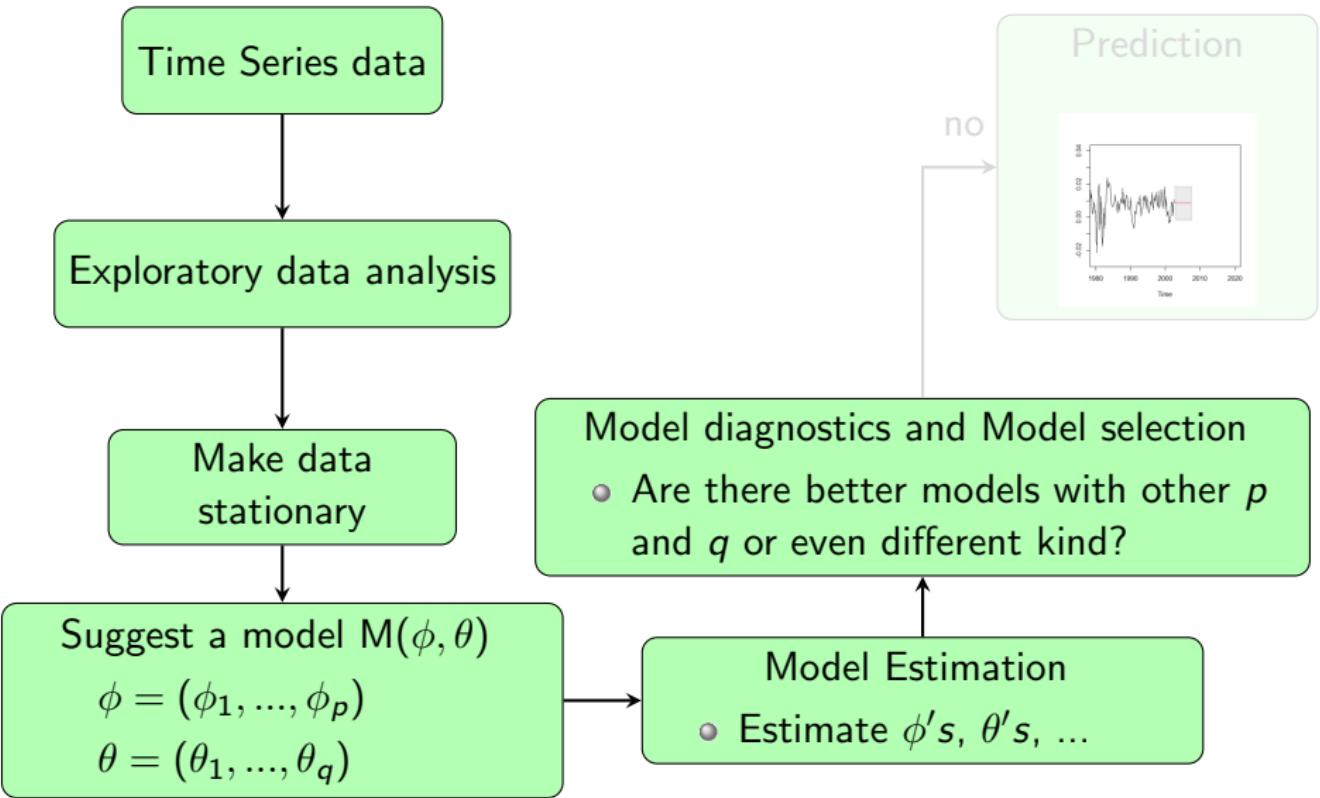
Time domain: The Big Picture



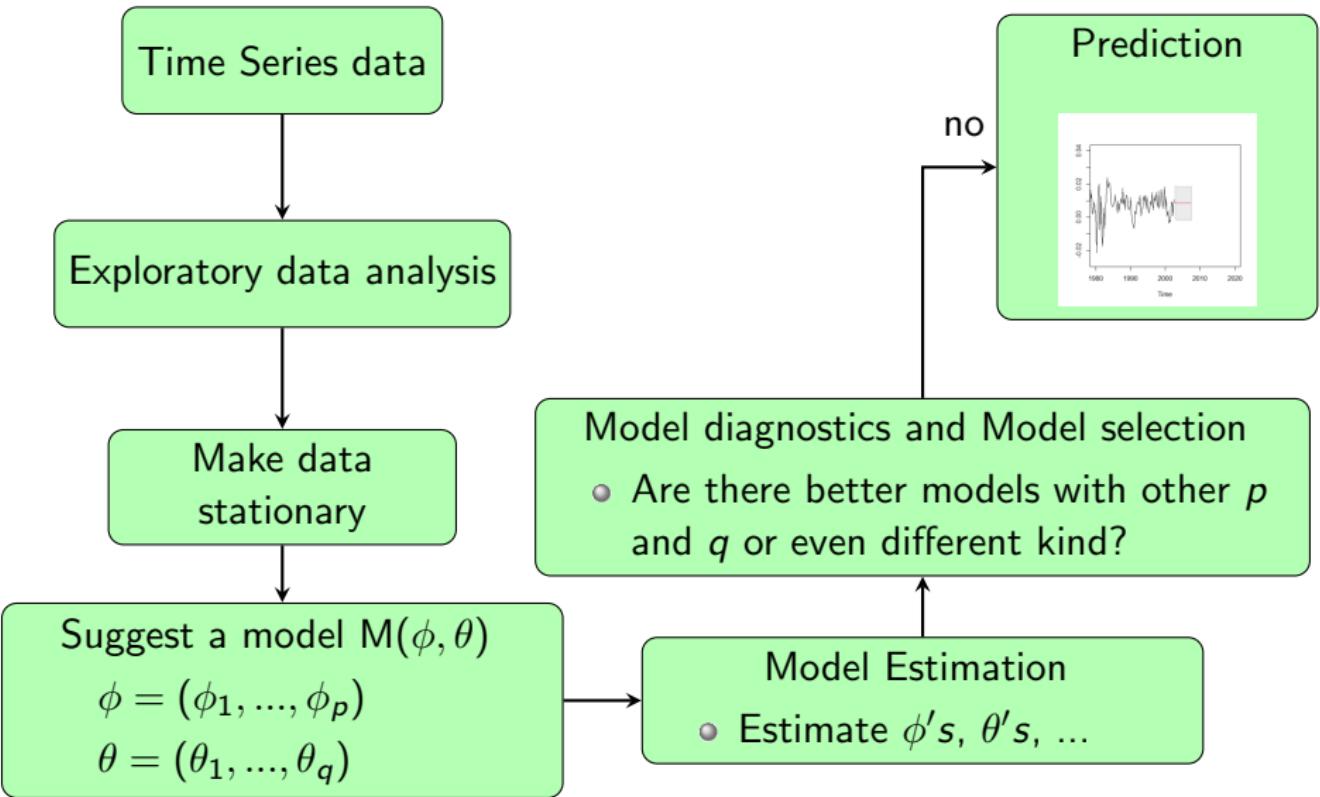
Time domain: The Big Picture



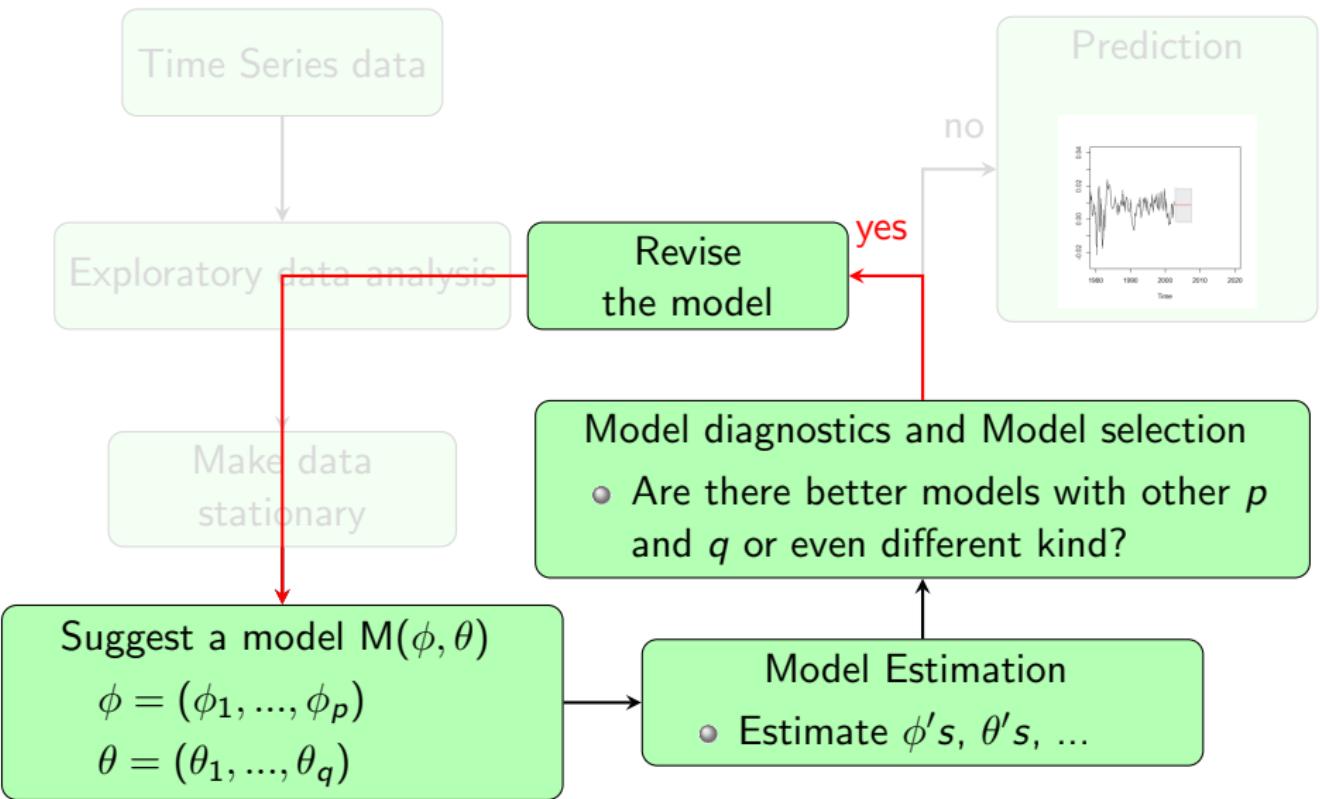
Time domain: The Big Picture



Time domain: The Big Picture



Time domain: The Big Picture



Course topics

- Time series regression and explorative analysis
- ARIMA models
 - ▶ AR, MA, ARMA, ARIMA, seasonal ARIMA
 - ▶ Model selection
 - ▶ Estimation
 - ▶ Forecasting
- State space models
 - ▶ Linear and Gaussian state space models
 - ▶ Kalman filtering and smoothing
- Recurrent Neural Networks (RNNs)

Course organization

- Lectures
 - ▶ Available at LISAM
- Teaching sessions
- Computer labs
 - ▶ Available at LISAM, under Submissions
 - ▶ Work in pairs
 - ▶ Send your report via LISAM
 - ▶ Deadlines
- Written assignments
 - ▶ Submissions needed - keys are given for some assignments
- Examination
 - ▶ Computer based exam
 - ▶ Submission of lab reports and written assignments

Course organization

- Software: R
 - ▶ <https://www.r-project.org/>
 - ▶ <https://www.rstudio.com/>



- Define your groups (2 persons) this week:
 - ▶ <https://docs.google.com/spreadsheets/d/1tzG35WSDWRhHWFA0cNOL1WUzoYqdn0GhZUwvXz3HhII/edit?usp=sharing>
 - ▶ **Difficult to find a group? Put your name in some cell.. I will merge you to someone**

Course organization

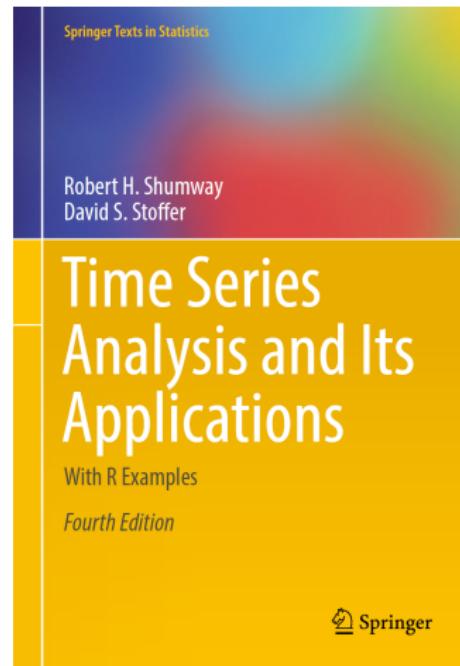
Course literature:

Time series Analysis and its Applications, Fourth Edition (2017), ISBN 978-3-319-52451-1

Can be downloaded freely here:

<https://www.stat.pitt.edu/stoffer/tsa4/tsa4.pdf>

- Do not skip examples when you read!
- First 2 chapters are easy, but don't relax!



Time Series models

- Time series x_t : random variable
 - ▶ A collection of $x_t =$ stochastic process
 - ▶ $t = 0, \pm 1, \pm 2, \dots$
- (probably) Simplest series: white noise
 - ▶ w_t uncorrelated (white: all possible periodic oscillations are present at equal strength)

$$w_t \sim wn(0, \sigma_w^2)$$

- ▶ w_t independent and identically distributed (white independent noise)

$$w_t \sim iid(0, \sigma_w^2)$$

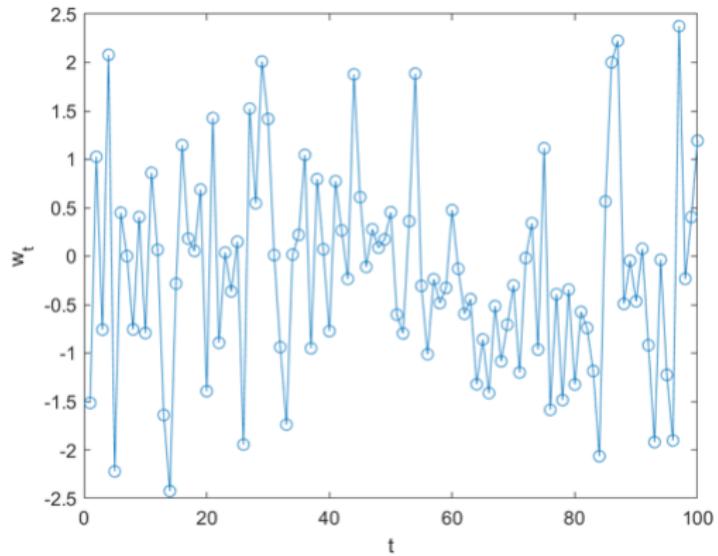
- Reminder:

$$\text{uncorrelated} \iff E(XY) = EX.EY$$

$$\text{independent} \iff f_{X,Y}(x,y) = f_X(x).f_Y(y)$$

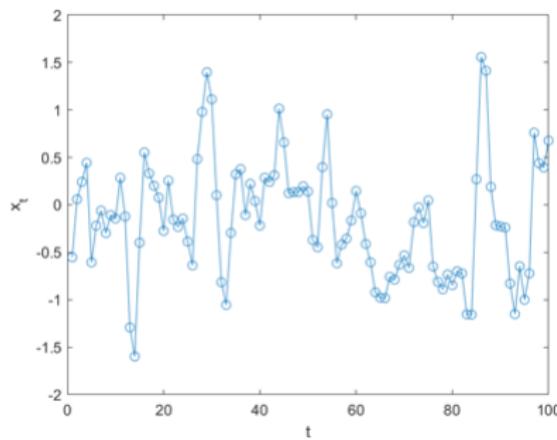
White noise

- Example: $w_t \sim iidN(0, 1)$



Moving average

Example: $x_t = 0.2w_{t-1} + 0.5w_t + 0.2w_{t+1}$



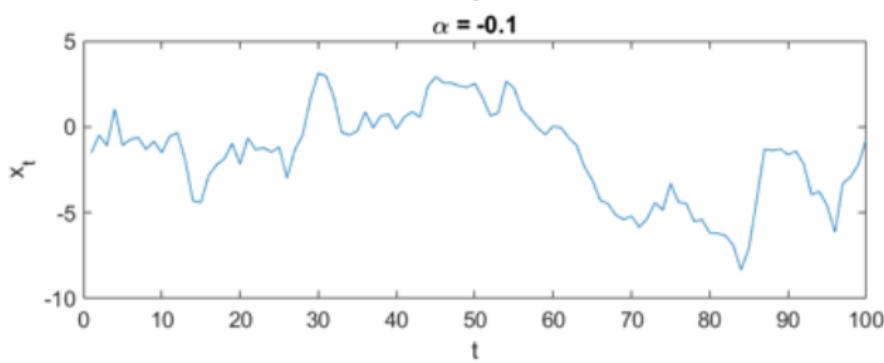
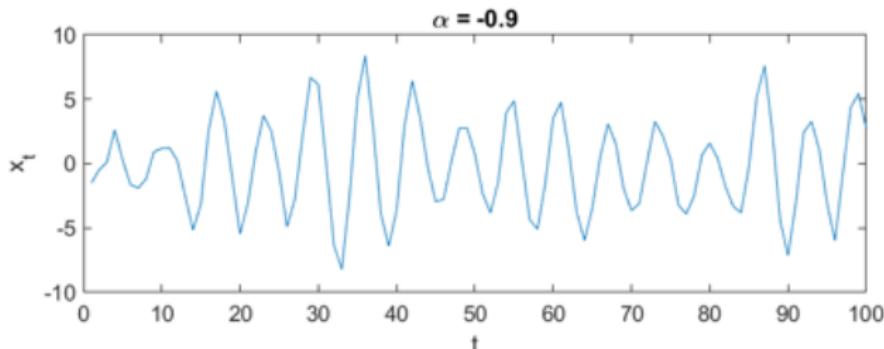
Very Interesting Fact: most stationary processes can be represented as a sum of lagged white noise:

$$x_t = \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}$$

Autoregressive model

Example: AR(2) process (Assume $x_0 = 0, x_{-1} = 0$)

$$x_t = x_{t-1} + \alpha x_{t-2} + w_t$$



Random walk with drift

A simple model for a "drifting" time series

$$x_t = \delta + x_{t-1} + w_t$$

- δ is the drift
- $\delta = 0 \Rightarrow$ random walk

Note: if we assume $x_0 = 0$,

$$x_t = \delta t + \sum_{j=1}^t w_j$$

Random walk with drift

A simple model for a "drifting" time series

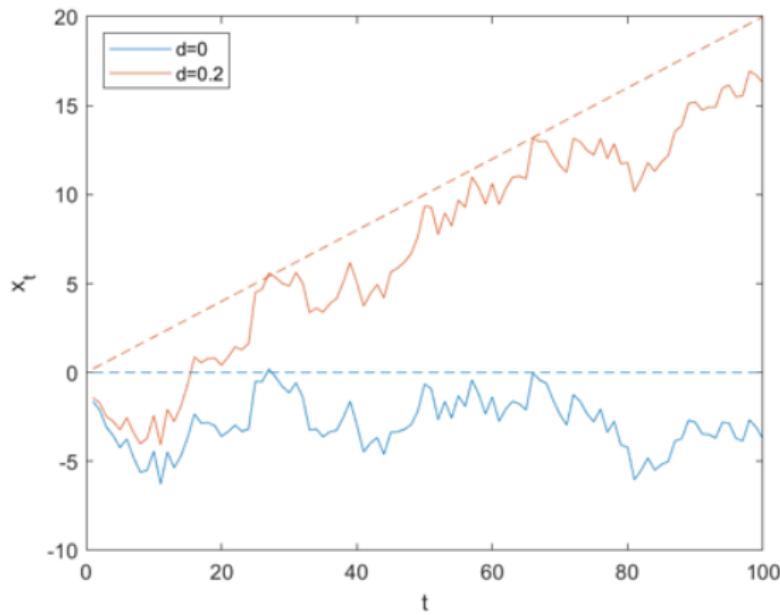
$\delta=0$ and $\delta=0.2$

$$x_t = \delta + x_{t-1} + w_t$$

- δ is the drift
- $\delta = 0 \Rightarrow$ random walk

Note: if we assume $x_0 = 0$,

$$x_t = \delta t + \sum_{j=1}^t w_j$$



Basic statistics - reminder

- Probability density function for x : $f(x)$
- Marginal density $f_i(x_i) = \int f(x) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_p$
- Expected (mean) value $Ex = \int xf(x)dx$
- Covariance $\text{cov}(x, y) = E\{(x - Ex)(y - Ey)\}$
- Variance $\text{var}(x) = E\{(x - Ex)^2\} = \text{cov}(x, x)$
- Relationships (a is a constant)
 - ▶ $E(x + a) = Ex + a$, $E(ax) = aEx$
 - ▶ $E(x + y) = Ex + Ey$
 - ▶ $\text{cov}(x + a, y) = \text{cov}(x, y)$
 - ▶ $\text{cov}(x + z, y) = \text{cov}(x, y) + \text{cov}(z, y)$
 - ▶ $\text{var}(ax) = a^2 \text{var}(x)$

Statistical representation of a time series

Which measures of dependence exist for time series?

- Theoretical?
- Practical?

Given time series x_1, \dots, x_n measured at fixed t_1, \dots, t_n

- Joint pdf

$$f_{t_1, \dots, t_n}(x_{t_1}, \dots, x_{t_n})$$

- Marginal pdf

$$f_t(x_t)$$

Statistical representation of a time series on whiteboard

Mean function at time t

$$\mu_t = E(x_t) = \int_{-\infty}^{\infty} xf_t(x)dx$$

Examples: Compute mean function for

- Moving average $x_t = 0.2w_{t-1} + 0.5w_t + 0.2w_{t+1}$
- Random walk $x_t = \delta t + \sum_{j=1}^t w_j$

Autocovariance and ACF

How do we measure linear dependence between two variables? → Covariance or Correlation

How do we measure linear dependence between two time-lags in a time series? In the same way!

- Autocovariance function

$$\gamma(s, t) = \text{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)]$$

Note $\text{var}(x_t) = \gamma(t, t)$

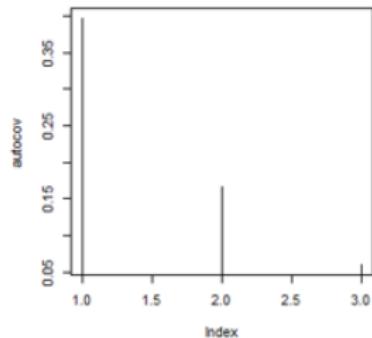
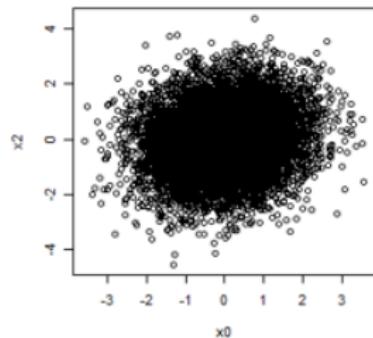
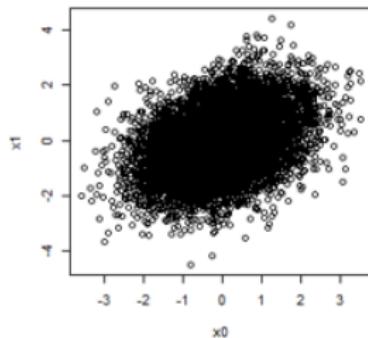
- Autocorrelation function (ACF)

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}}$$

Autocovariance and ACF

Generate x_0, x_1, x_2 from $x_t = 0.4x_{t-1} + w_t$

- Consider $\gamma(0, 1), \gamma(0, 2)$



Autocovariance and ACF

Useful fact: If $U = \sum_{j=1}^m a_j x_j$ and

$$V = \sum_{k=1}^r b_k y_k$$

$$\text{cov}(U, V) = \sum_{j=1}^m \sum_{k=1}^r a_j b_k \text{cov}(x_j, y_k)$$

Examples: Autocovariance and ACF of on whiteboard

- White noise
- Random walk $x_t = \delta t + \sum_{j=1}^t w_j$
- Moving average $x_t = 0.2w_{t-1} + 0.5w_t + 0.2w_{t+1}$

Home reading

- Shumway and Stoffer, chapters 1.1-1.3
- TS functions in R: ts, plot.ts, acf, ts.intersect, filter, ts.plot

Time Series Analysis

Lecture 2: Exploratory analysis and Time Series Regression

Tohid Ardestiri

Linköping University
Division of Statistics and Machine Learning

September 4, 2019



LINKÖPING
UNIVERSITY

Summary of Lecture 1

- Time series
 - ▶ White noise
 - ▶ Random walk
 - ▶ Moving average filter
- Autocovariance and autocorrelation functions:

$$\gamma(s, t) = \text{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)]$$

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}}$$

Autocovariance and ACF

Examples: Autocovariance and ACF of on whiteboard

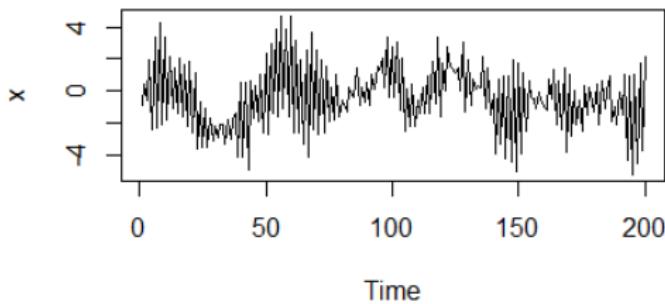
- White noise ✓
- Random walk $x_t = \delta t + \sum_{j=1}^t w_j$
- Moving average $x_t = 0.2w_{t-1} + 0.5w_t + 0.2w_{t+1}$

Autocovariance

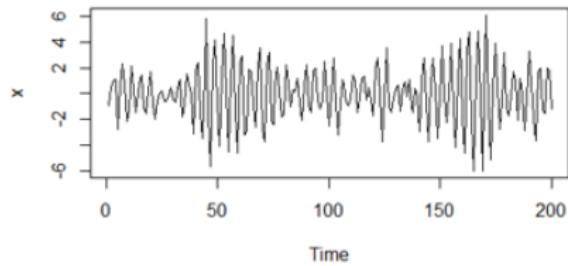
- Intuition:

$$x_t = \phi x_{t-1} + w_t$$

$$\alpha = 0.9$$



$$\alpha = -0.9$$



- when $x_0 = 0$ and $w_t \sim wn(0, 1)$:

$$\text{cov}(x_t, x_{t-1}) = \phi$$

Autocovariance (read at home)

$$x_t = \phi x_{t-1} + w_t$$

Mean function:

$$Ex_t = \phi Ex_{t-1} + Ew_t = \phi Ex_{t-1} = \phi(\phi Ex_{t-2}) = \dots = \phi^t Ex_0$$

for $Ex_0 = 0$, $Ex_t = 0$ for all t .

Variance $\text{var}(x_t)$ when $Ex_0 = 0$ and w_t is uncorrelated with x_0 for all t :

$$\begin{aligned}\text{var}(x_t) &= E\{(x_t - 0)^2\} = E\{\phi^2 x_{t-1}^2 + 2\phi x_{t-1} w_t + w_t^2\} = \\ \phi^2 \text{var}(x_{t-1}) + 2\phi \text{cov}(x_{t-1}, w_t) + \text{var}(w_t) &= \phi^2 \text{var}(x_{t-1}) + \text{var}(w_t) = \\ \phi^2 \text{var}(x_{t-1}) + \sigma_w^2 &= \phi^2(\phi^2 \text{var}(x_{t-2}) + \sigma_w^2) + \sigma_w^2 = \\ \phi^{2t} \text{var}(x_0) + \sigma_w^2 \sum_{k=0}^{t-1} (\phi^{2k}) &= \phi^{2t} \text{var}(x_0) + \frac{\sigma_w^2(1-\phi^{2t})}{1-\phi^2}\end{aligned}$$

When $\text{var}(x_0) = \frac{\sigma_w^2}{1-\phi^2}$ then $\text{var}(x_t) = \frac{\sigma_w^2}{1-\phi^2}$ and time independent.

Autocovariance (read at home)

$$x_t = \phi x_{t-1} + w_t$$

$$x_t = \phi(\phi x_{t-2} + w_{t-1}) + w_t = \dots = \phi^h x_{t-h} + \sum_{j=0}^{h-1} \phi^j w_{t-j}$$

$$\begin{aligned}\gamma(x_t, x_{t-h}) &= \text{cov}(x_t, x_{t-h}) = E(x_t x_{t-h}) = \\ E\{(\phi^h x_{t-h} + \sum_{j=0}^{h-1} \phi^j w_{t-j}) x_{t-h}\} &= \phi^h \text{var}(x_{t-h}) = \frac{\phi^h \sigma_w^2}{1-\phi^2}\end{aligned}$$

Hence,

$$\gamma(h) = \frac{\phi^h \sigma_w^2}{1 - \phi^2}$$

Also,

$$\rho(h) = \phi^h$$

Stationarity

Fact: sometimes $\rho(s, t)$ depends on lag $|s - t|$ only

Time series is **strictly stationary** if distributions of $\{x_{t1}, \dots, x_{tn}\}$ and $\{x_{t1+h}, \dots, x_{tn+h}\}$ are identical for any $\{t_1, \dots, t_n\}$ and all lags $h = 0, \pm 1, \pm 2, \dots$

$$P(x_{t1} \leq c_1, \dots, x_{tn} \leq c_n) = P(x_{t1+h} \leq c_1, \dots, x_{tn+h} \leq c_n)$$

Note: This means

- Mean function $\mu_t = E x_t = \text{const.}$
- Autocovariance $\gamma(t, t + h) = \text{function only of lag } h$

Stationarity

Strict stationarity is often too strong!

- Time series x_t is **weakly stationary (stationary)** if
 - ▶ $E x_t = \text{const}$
 - ▶ $\gamma(s, t) = \gamma(|s - t|)$
 - ▶ $\text{var}(x_t) < \infty$
- $\gamma(t, t + h) = \gamma(|t + h - t|) = \gamma(h)$
 - ▶ Autocovariance depends on lag only!
- Autocovariance for stationary process $\gamma(h) = \text{cov}(x_t, x_{t+h})$
- ACF for stationary process $\rho(h) = \frac{\gamma(h)}{\gamma(0)}$

Stationarity

Properties of stationary process:

$$\gamma(h) = \gamma(-h) \quad \rho(h) = \rho(-h)$$

$$|\gamma(h)| \leq \gamma(0) \quad \rho(h) \leq 1, \rho(0) = 1$$

Reflect: Are these processes stationary?

- White noise
- Moving average, $x_t = 0.2w_{t-1} + 0.5w_t + 0.2w_{t+1}$
- Random walk, $x_t = \delta t + \sum_{j=1}^t w_j$

Sample autocovariance and ACF

Dependence measures for samples?

- Idea: replace mean and covariance with sample estimates

If x_t is stationary,

- Sample mean

$$Ex \approx \bar{x} = \frac{1}{n} \sum_{t=1}^n x_t$$

- Sample autocovariance function

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x})$$

Sample autocovariance and ACF

Example: n=6, h=2

		X1	X2	X3	X4	X5	X6
X1	X2	X3	X4	X5	x6		

Sample autocorrelation function (sample ACF)

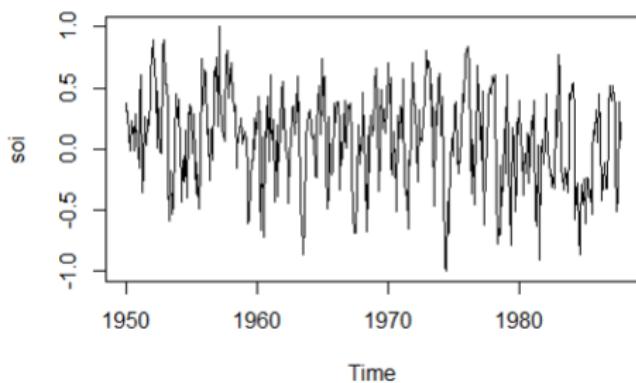
$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}$$

Sample ACF

In R: `acf()`

Example: southern oscillation index (SOI)

- `rho=acf(soi, 5, type="correlation", plot=T)`



```
> print(rho)
```

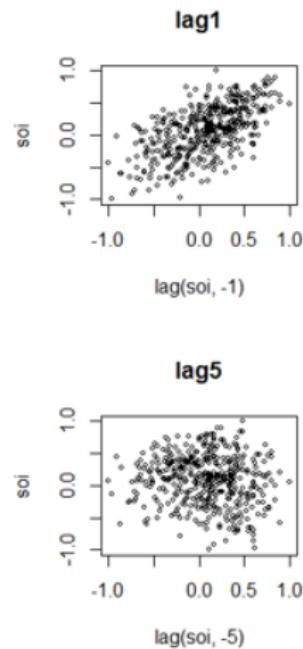
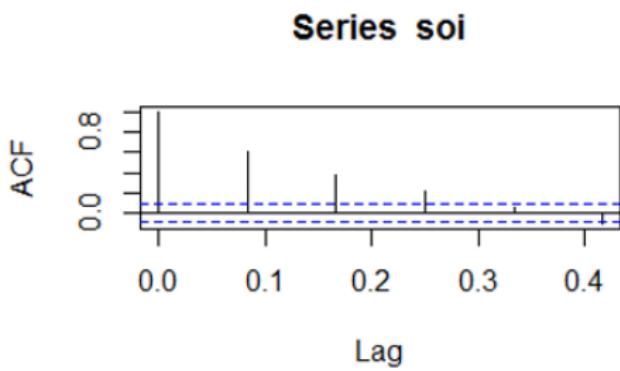
```
Autocorrelations of series 'soi', by lag
```

Lag	Autocorrelation
0	0.0000
1	0.0833
2	0.1667
3	0.2500
4	0.3333
5	0.4167

```
0.0000 0.0833 0.1667 0.2500 0.3333 0.4167  
1.0000 0.6040 0.3740 0.2140 0.0500 -0.1070
```

Why is sample ACF '1' for h=0?

Sample ACF



Sample ACF

What are these blue lines?

Theorem: Under weak conditions, if x_t is white noise and $n \rightarrow \infty$ then $\hat{\rho}(h)$ is approximately $N(0, \frac{1}{n})$

Consequence: If some $|\hat{\rho}(h)| > \frac{2}{\sqrt{n}}$ then the time series is not a white noise (with approximately 95 % confidence).

Typical modeling strategy:

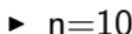
- Fit a model
- Compute residuals
- Check ACF within $\pm \frac{2}{\sqrt{n}}$

Sample ACF vs theoretical

- Moving average $x_t = 0.2w_{t-1} + 0.5w_t + 0.2w_{t+1}$



$$ACF\gamma(h) = \begin{cases} 1 & h = 0 \\ 0.61 & h = 1 \\ 0.12 & h = 2 \\ 0 & other \end{cases}$$



Autocorrelations of series 'y1', by lag

0	1	2	3	4	5
1.000	0.236	-0.399	-0.187	-0.008	-0.118



Autocorrelations of series 'y1', by lag

0	1	2	3	4	5
1.000	0.609	0.129	-0.007	0.001	0.044

⋮

Vector-valued time series

If $x_t = (x_{t1}, x_{t2}, \dots, x_{tp})'$ is stationary,

- mean vector is $\mu = E(x_t)$ and sample mean is its approximation

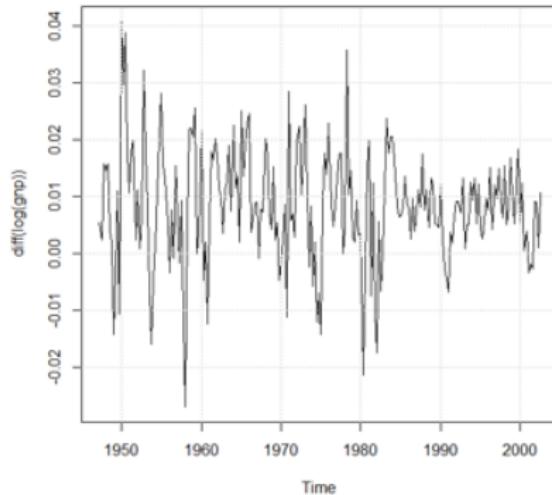
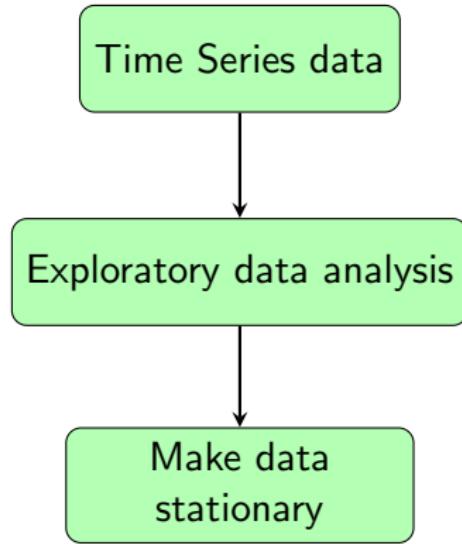
$$\mu = E(x_t) \approx \bar{x} = \frac{1}{n} \sum_{t=1}^n x_t$$

- Autocovariance function is $\Gamma(h) = E[(x_{t+h} - \mu)(x_t - \mu)']$ and sample autocovariance matrix

$$\hat{\Gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x})'$$

Recap: time domain modeling

$$Y_t = \nabla(\log(X_t))$$



Stationarity

- Why do we need stationarity?
 - ▶ Sample ACF becomes consistent
 - ▶ ARIMA models require stationarity

- Tools
 - ▶ Detrending (trend removal)
 - ▶ Differencing
 - ▶ Transformations

whiteboard

- Introduce linear regression/least squares
- Trend removal, simple drift

Trend removal by regression

Regressing on covariates

Given x_t (dependent series) and z_{t1}, \dots, z_{t2} (independent series) we model

$$x_t = \beta_0 + \beta_1 z_{t1} + \dots + \beta_q z_{tq} + w_t$$

where w_t is assumed white noise.

Note: w_t is seldom white noise in practice, used as a tool for detrending!

Trend removal by regression

Still a linear regression in β

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad Z = \begin{pmatrix} 1 & z_{11} & \dots & z_{1q} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z_{n1} & \dots & z_{nq} \end{pmatrix}$$

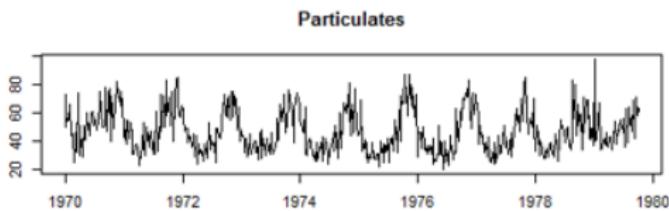
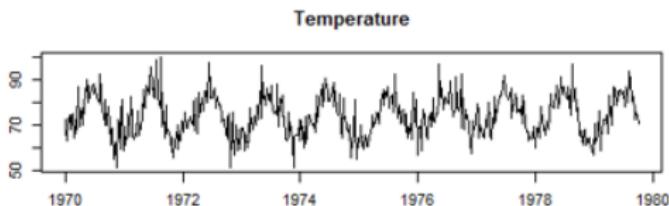
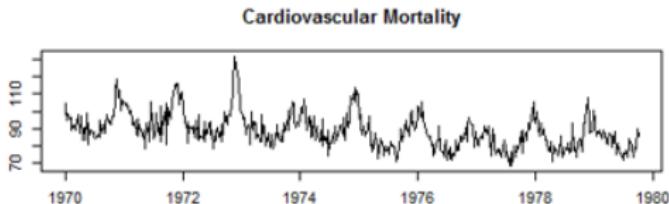
Least squares estimate is computed as

$$\hat{\beta} = (Z^T Z)^{-1} Z^T X$$

Trend removal

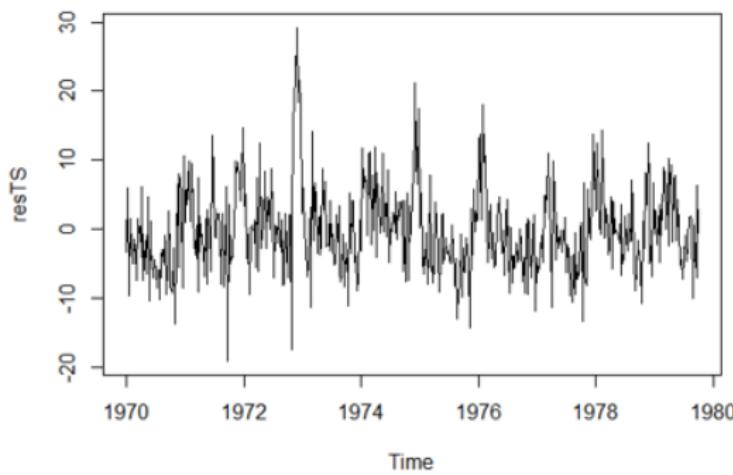
Example: Mortality

- x_t : Cardiovascular mortality
- z_{t1} : Temp (centered)
- z_{t2} : Temp (centered, squared)
- z_{t3} : Time
- z_{t4} : Levels of particles



Trend removal

- Residuals
 - ▶ Stationary?
 - ▶ Independent?
 - ▶ Some additional modeling of the residuals (ARIMA) can be done



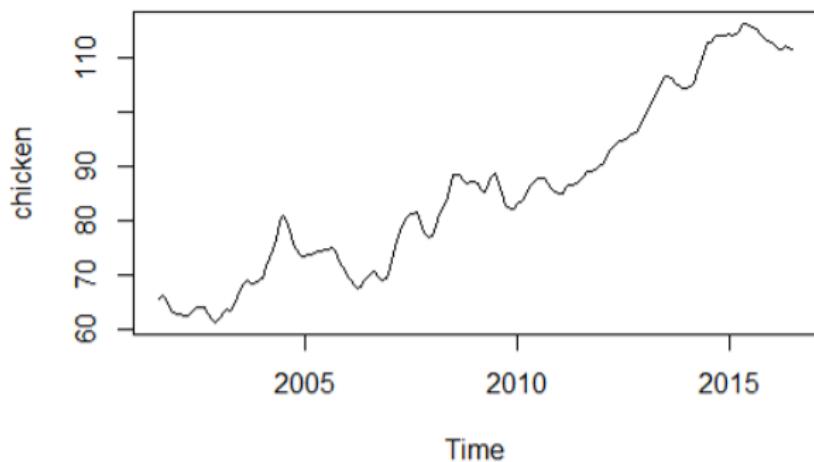
Differencing

Assume $x_t = \mu_t + y_t$, y_t stationary

Differencing gives $z_t = \nabla x_t = x_t - x_{t-1}$

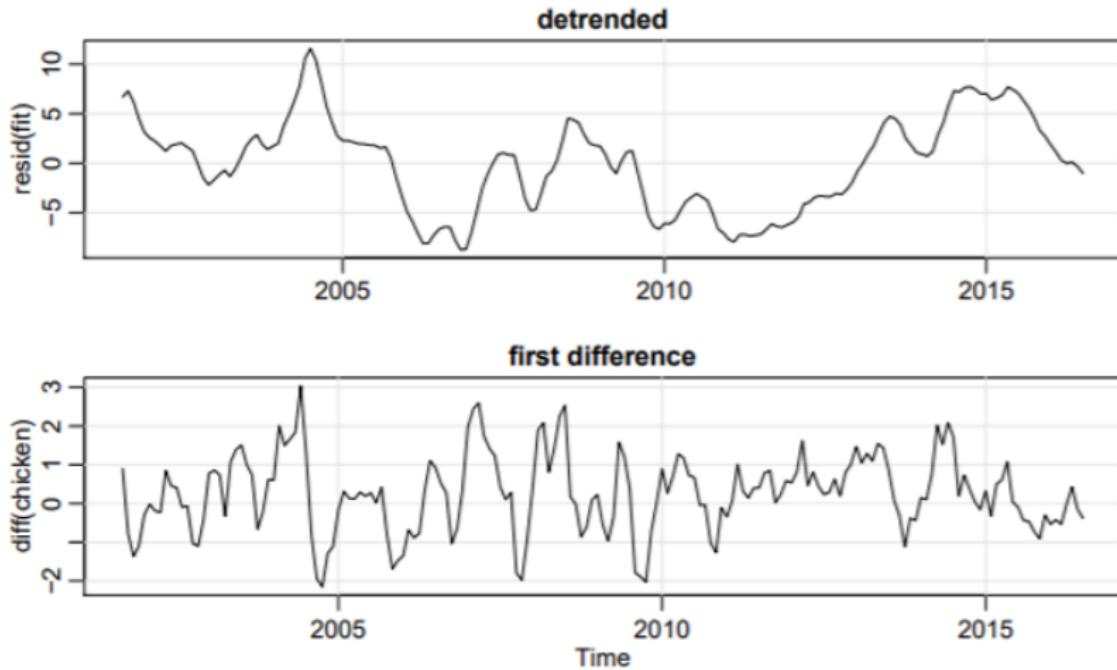
- **Property 1:** If $\mu_t = \alpha_0 + \alpha_1 t$ then z_t is stationary
- **Property 2:** If μ_t is random walk with a drift then z_t is stationary

Example:
Chicken prices

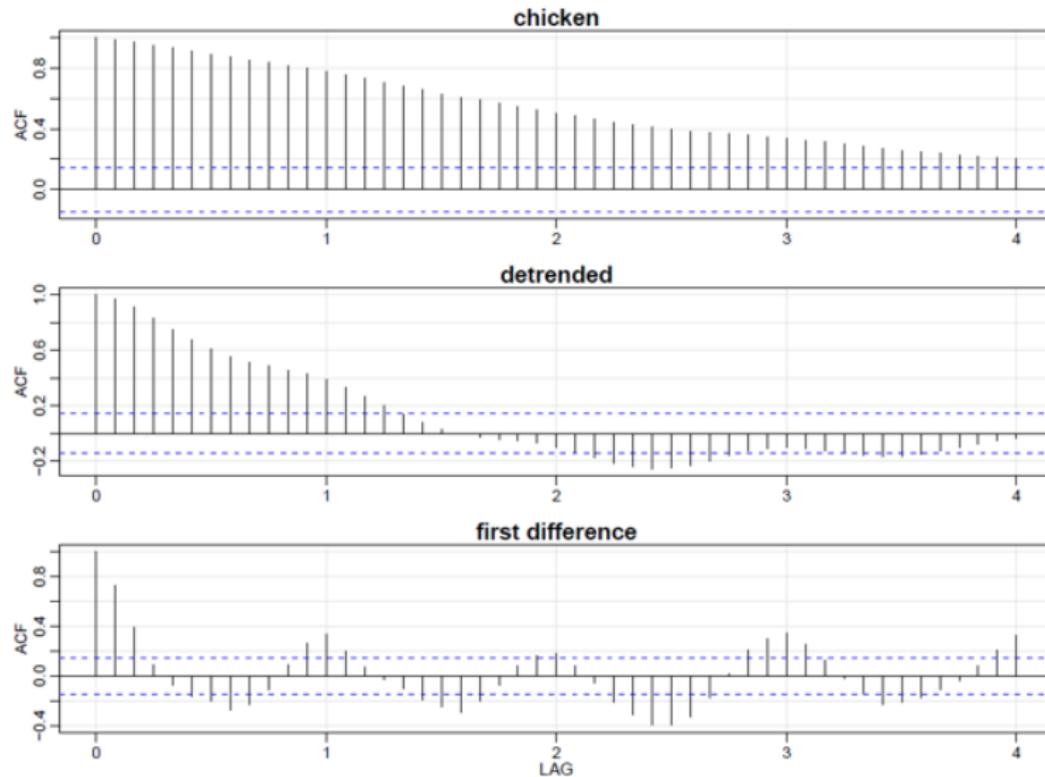


Differencing

Which looks **most random**? Other differences?



Differencing



Detrending vs differencing

- Differencing is more flexible than linear detrending
- Differencing does not require model estimation
- If trend is complex, detrending with a flexible (machine learning) model can be better
- Differencing does not give us the trend

Backshift operator

- Backshift operator $Bx_t = x_{t-1}$, Powers $B^k x_t = x_{t-k}$
- Forward-shift operator $B^{-1}x_t = x_{t+1}$
- Note $BB^{-1}x_t = x_t$ (i.e. $BB^{-1} = 1$)
- Differencing $\nabla x_t = (1 - B)x_t$
- Differences of order d : $\nabla^d = (1 - B)^d$
- Property: Operators can be manipulated as polynomials
- Example Check that $\nabla^2 x_t = x_t - 2x_{t-1} + x_{t-2}$
- Property: Differencing of order p can remove polynomial trend of order p

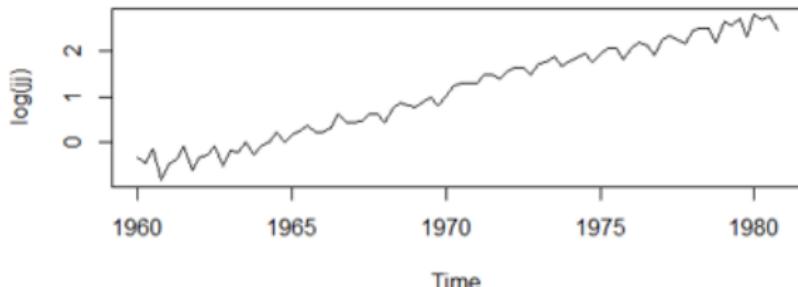
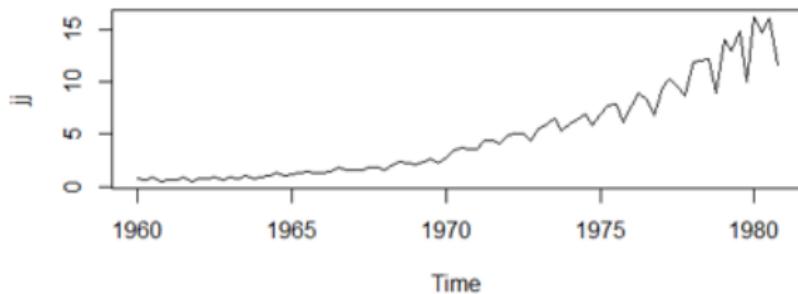
Transformations

- Often used to stabilize variance
 - ▶ If for $\text{ex.var}(x_t) \neq \text{var}(x_s)$ then time series is non-stationary ...
- Sometimes makes data more similar to normal distr.
- Common transforms:
 - ▶ $z_t = \log(x_t)$
 - ▶ Power transformation

$$z_t = \begin{cases} \frac{(x_t^\lambda - 1)}{\lambda} & \lambda \neq 0 \\ \log(x_t) & \lambda = 0 \end{cases}$$

Transformations

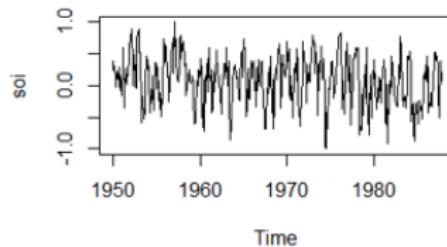
- Johnson & Johnson quarterly earnings



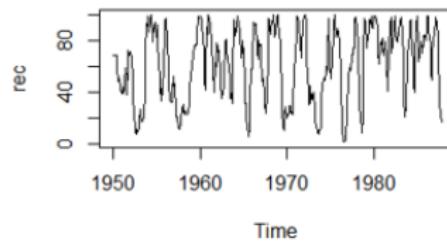
Scatterplots

- Plot x_t vs z_{t_i} or z_{t_i} vs z_{t_j}
- Exploratory tool: indicates which relationship to model

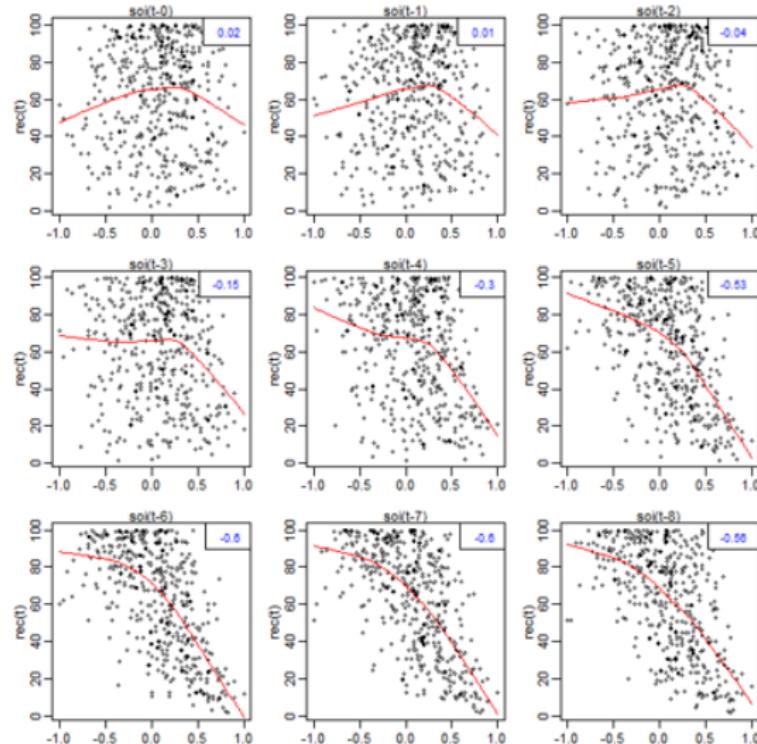
$$x_t = f(z_{t_1}, z_{t_2}, \dots, z_{t_q}) + w_t$$



- Example: SOI and Recruitment



Scatterplots



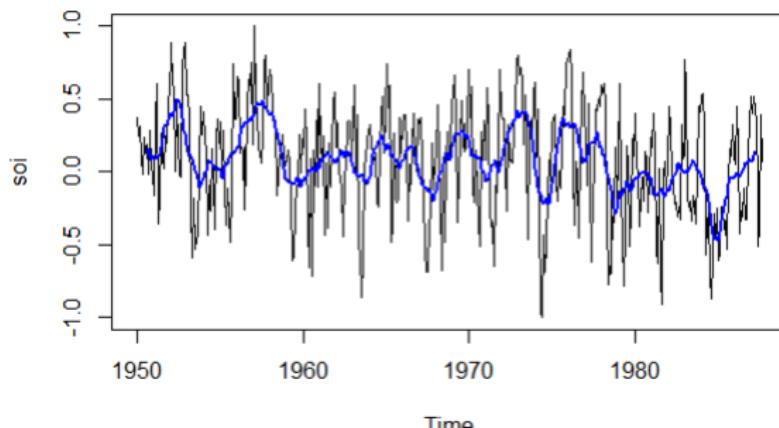
- Which relationships are nonlinear?
- Conclusion:
include dummy variables
 $I(\text{soil}(t - j) > 0)$ in the linear model

Smoothing

- Moving average smoother

$$m_t = \sum_{j=-k}^{j=k} a_j x_{t-j}$$

- Where $\sum_{j=-k}^{j=k} a_j = 1$ and $a_j = a_{-j} \geq 0$,
- Example: SOI data Disadvantage?



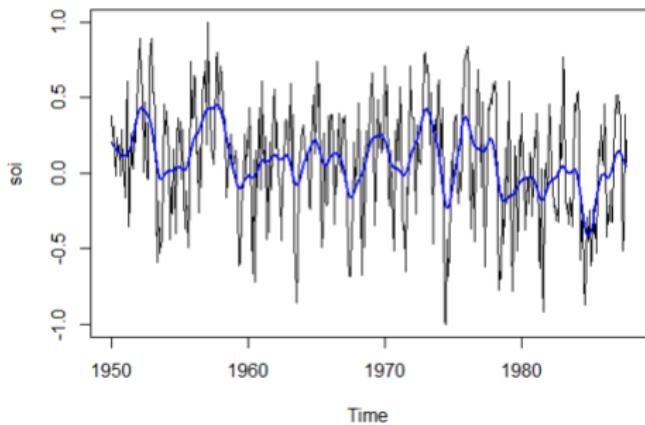
Smoothing

More flexible models?

- Splines
- Kernel smoothers
- Gaussian Process
- Neural networks
- ...

Welcome to ML courses!!

Example: kernel smoothers



Home reading

- Shumway and Stoffer, sections 1.4-1.6 and chapter 2
- TS functions: lag, ksmooth, lm, diff, lag1.plot, lag2.plot

Time Series Analysis

Lecture 3: Introduction to ARIMA

Tohid Ardesthiri

Linköping University
Division of Statistics and Machine Learning

September 6, 2019



Recap

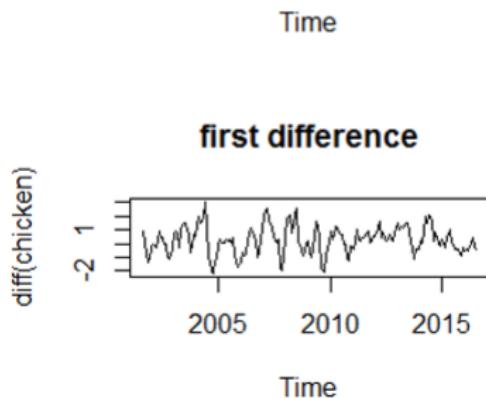
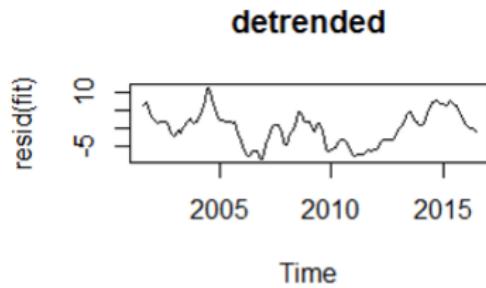
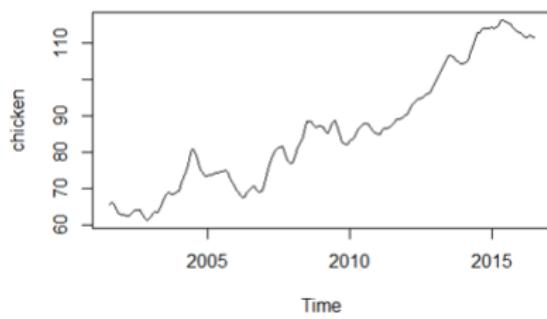
How to make data stationary?

- Transformations (log, other)
- Detrending
 - ▶ Differencing
 - ▶ Linear regression
 - ▶ Kernel smoother
 - ▶ ...

How shall we model the data after detrending and transformations
(residuals)? →?ARIMA models!

ARIMA models

- Why ARIMA models?
 - ▶ Removing trend is not sufficient



Moving average models

- Moving average model of order q, MA(q)

$$\begin{aligned}x_t &= w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q} \\&= \sum_{j=0}^q \theta_j w_{t-j}\end{aligned}$$

- ▶ $w_t \sim wn(0, \sigma_w^2)$
- ▶ $\theta_1, \dots, \theta_q$ constants, $\theta_q \neq 0$ and $\theta_0 = 1$

- Moving average operator

$$\theta(B) = \sum_{j=0}^q \theta_j B^j$$

- MA(q): $x_t = \theta(B)w_t$

Linear process

x_t is a **linear process** if

$$x_t = \mu + \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}$$

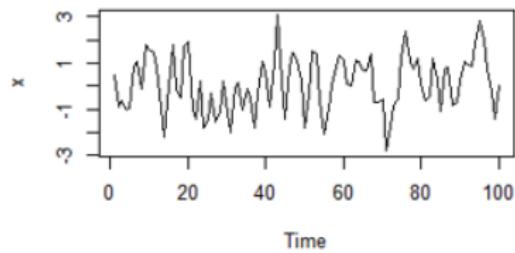
Property: It can be shown that

$$\gamma_x(h) = \sigma_w^2 \sum_{j=-\infty}^{\infty} \psi_{j+h} \psi_j$$

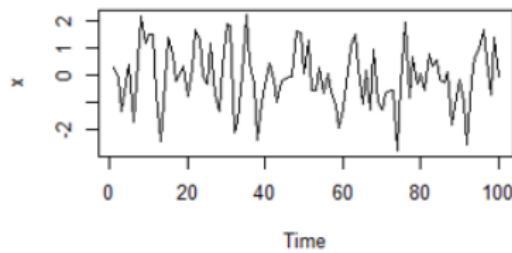
Example: MA(1)

$$x_t = w_t + \theta w_{t-1}$$

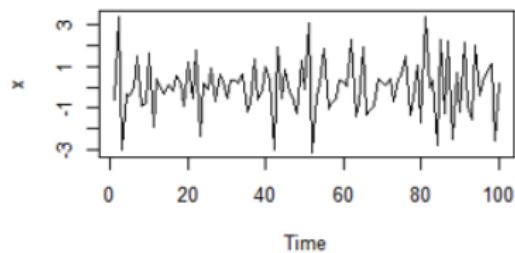
$\theta = 0.9$



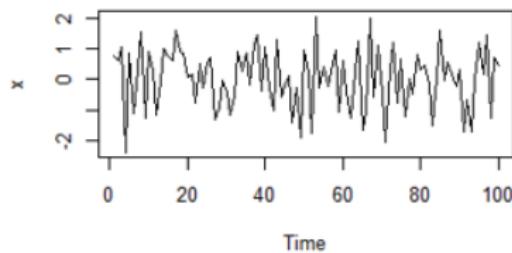
$\theta = 0.2$



$\theta = -0.9$



$\theta = -0.2$



Example: MA(1)

$$x_t = w_t + \theta w_{t-1}$$

- Autocovariance and ACF

$$\gamma(h) = \begin{cases} (1 + \theta^2)\sigma_w^2 & h = 0 \\ \theta\sigma_w^2 & h = 1 \\ 0 & h > 1 \end{cases}$$

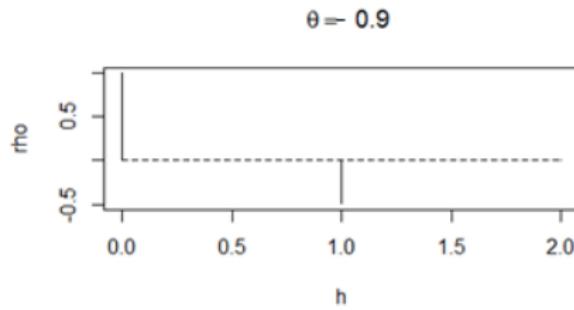
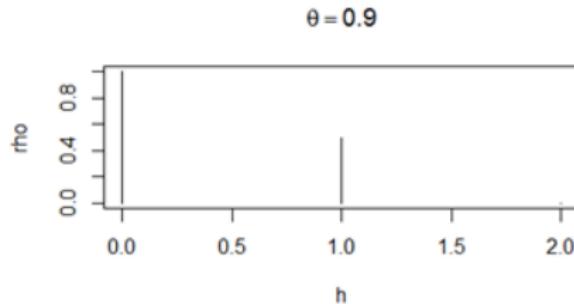
$$\rho(h) = \begin{cases} \frac{\theta}{1+\theta^2} & h = 1 \\ 0 & h > 1 \end{cases}$$

Note: $\rho(0) = 1$ is often not written as it is trivial.

- Process is stationary

Example: MA(1)

- Note: $\rho(0) = 1$ is often not shown \rightarrow only 1 bar



AR models

- Autoregressive model of order p , $AR(p)$

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t$$

- ▶ x_t is stationary if x_0 is sampled from the stationary distribution
 - ▶ $w_t \sim \text{wn}(0, \sigma_w^2)$
 - ▶ ϕ_1, \dots, ϕ_p constants, $\phi_p \neq 0$
 - ▶ $E x_t = 0$
-
- Note: if $E x_t = \mu \neq 0$, model $x'_t = x_t - \mu$

AR models

Another form

- **Autoregressive operator**

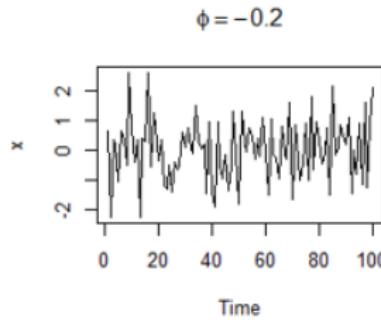
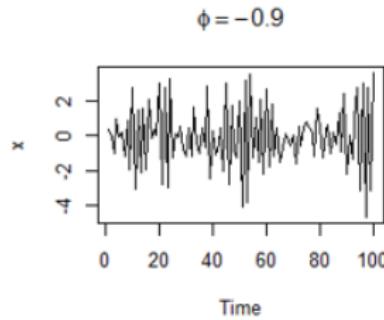
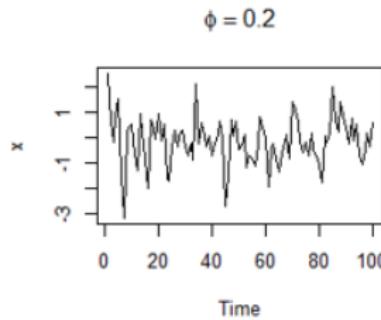
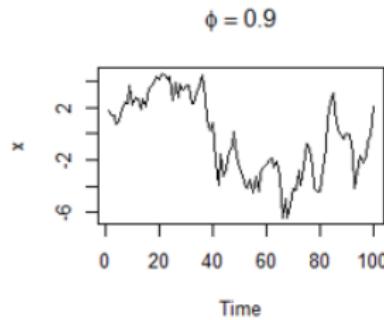
$$\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$$

- AR(p) model

$$\boxed{\phi(B)x_t = w_t}$$

Example: AR(1)

- How do these plots differ? $x_t = \phi x_{t-1} + w_t$



Ar(1) (read at home)

$$x_t = \phi x_{t-1} + w_t$$

Mean function:

$$Ex_t = \phi Ex_{t-1} + Ew_t = \phi Ex_{t-1} = \phi(\phi Ex_{t-2}) = \dots = \phi^t Ex_0$$

for $Ex_0 = 0$, $Ex_t = 0$ for all t .

Variance $\text{var}(x_t)$ when $Ex_0 = 0$ and w_t is uncorrelated with x_0 for all t :

$$\begin{aligned}\text{var}(x_t) &= E\{(x_t - 0)^2\} = E\{\phi^2 x_{t-1}^2 + 2\phi x_{t-1} w_t + w_t^2\} = \\ \phi^2 \text{var}(x_{t-1}) + 2\phi \text{cov}(x_{t-1}, w_t) + \text{var}(w_t) &= \phi^2 \text{var}(x_{t-1}) + \text{var}(w_t) = \\ \phi^2 \text{var}(x_{t-1}) + \sigma_w^2 &= \phi^2(\phi^2 \text{var}(x_{t-2}) + \sigma_w^2) + \sigma_w^2 = \\ \phi^{2t} \text{var}(x_0) + \sigma_w^2 \sum_{k=0}^{t-1} (\phi^{2k}) &= \phi^{2t} \text{var}(x_0) + \frac{\sigma_w^2(1-\phi^{2t})}{1-\phi^2}\end{aligned}$$

When $\text{var}(x_0) = \frac{\sigma_w^2}{1-\phi^2}$ then $\text{var}(x_t) = \frac{\sigma_w^2}{1-\phi^2}$ and time independent.

A(1) (read at home)

$$x_t = \phi x_{t-1} + w_t$$

$$x_t = \phi(\phi x_{t-2} + w_{t-1}) + w_t = \dots = \phi^h x_{t-h} + \sum_{j=0}^{h-1} \phi^j w_{t-j}$$

$$\begin{aligned}\gamma(x_t, x_{t-h}) &= \text{cov}(x_t, x_{t-h}) = E(x_t x_{t-h}) = \\ E\{(\phi^h x_{t-h} + \sum_{j=0}^{h-1} \phi^j w_{t-j}) x_{t-h}\} &= \phi^h \text{var}(x_{t-h}) = \frac{\phi^h \sigma_w^2}{1-\phi^2}\end{aligned}$$

Hence,

$$\gamma(h) = \frac{\phi^h \sigma_w^2}{1 - \phi^2}$$

Also,

$$\rho(h) = \phi^h$$

- **Property:** If $|\phi| < 1$ and $\sup \text{var}(x_t) < \infty$

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j}$$

- Show it by
 - ▶ Substitution
 - ▶ Taylor expansion
 - ▶ Coefficient matching
- Autocovariance and ACF

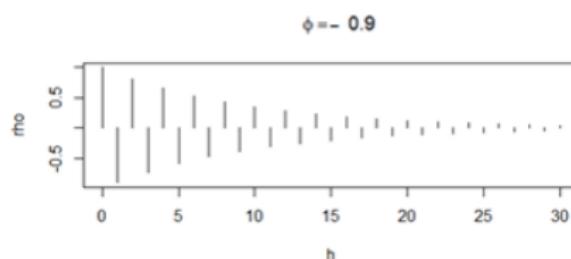
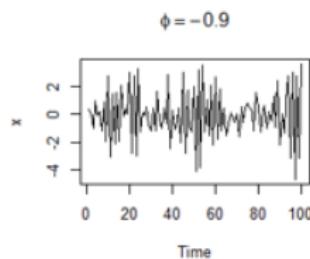
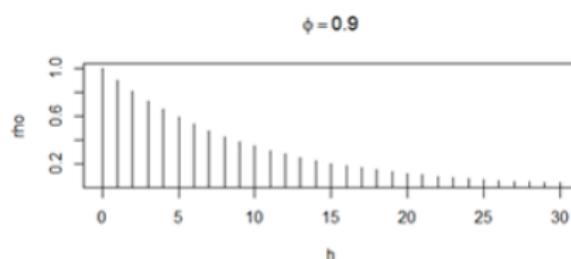
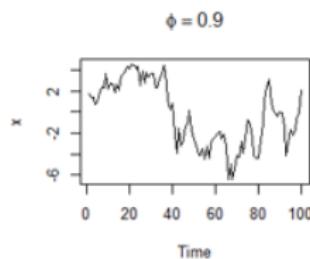
$$\gamma(h) = \frac{\sigma_w^2 \phi^h}{1 - \phi^2} \quad \rho(h) = \phi^h$$

for $h \geq 0$.

Example: AR(1)

Autocovariance and ACF (for $h \geq 0$)

$$\gamma(h) = \frac{\sigma_w^2 \phi^h}{1 - \phi^2} \quad \rho(h) = \phi^h$$



Explosive AR models

- **Explosive** =series become arbitrarily large in magnitude
- AR(1): What if $|\phi| > 1$?
 - ▶ $x_t = \phi^p x_{t-p} + \sum_{j=0}^{p-1} \phi^p w_{t-j} \rightarrow$ grows exponentially
 - ▶ **Stationary?** Check variance
- Can we make it stationary?
$$x_t = \phi^{-1} x_{t+1} - \phi^{-1} w_{t+1} = \phi' x_{t+1} + w'_t$$
 - ▶ Stationary, but dependent on the future
 - ▶ $w'_t \sim N(0, \phi^{-2} \sigma_w^2)$
 - ▶ $x_t = - \sum_{j=1}^{\infty} \phi^{-j} w_{t+j}$

Causal process

A stationary process is **causal** if it is only dependent on the past values of the process

Def: A linear process is **nonexplosive** and **causal** if it can be written as a one-sided sum:

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B)w_t$$

where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$ and $\sum_{j=0}^{\infty} |\psi_j| < \infty$.

$$\rho(h) = \begin{cases} \frac{\theta}{1+\theta^2} & h = 1 \\ 0 & h > 1 \end{cases}$$

Note: MA(1) gives equivalent models for $\theta = s$ and $\theta = \frac{1}{s}$

Probabilistic expressions equivalent: ACF identical

→ we can not distinguish between these models

Invertibility of MA

Def: An MA process is **invertible** if it has a causal AR representation,

$$w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$$

Example: MA(1) with $\theta = 1/5$ is invertible, $\theta = 5$ not.

ARMA models

- Autoregressive moving average ARMA(p,q)

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$$

- ▶ $\phi_p \neq 0, \theta_q \neq 0$
- ▶ Is stationary
- ▶ $E x_t = 0$
- p -autoregressive order, q -moving average order
- Alternative form

$$\phi(B)x_t = \theta(B)w_t$$

- Note: $x_t = \phi^{-1}(B)\theta(B)w_t = \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}$
 - ▶ But series might be non-convergent

Parameter redundancy

Note: we can multiply both sides with $\eta(B)$

$$\eta(B)\phi(B)x_t = \eta(B)\theta(B)w_t$$

- The resulting model looks different (higher orders)
- Underlying model is actually the same

Example: $x_t = w_t$, white noise. Let $\eta(B) = 1 - 0.5B$.

We get

$$x_t - 0.5x_{t-1} = w_t - 0.5w_{t-1}$$

Looks like ARMA(1,1)!

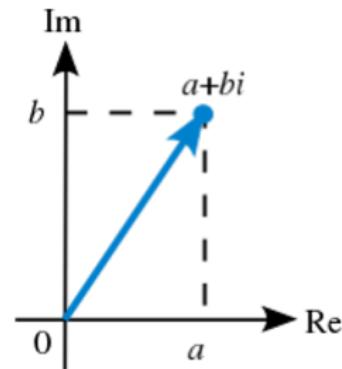
Reminder: complex numbers

- Imaginary unit $i^2 = -1$
- Complex number $z = a + ib$
- Conjugate $\bar{z} = a - ib$

- Absolute value $|z|^2 = z\bar{z} = a^2 + b^2$
- Trigonometric form
$$z = r(\cos(\theta) + i \sin(\theta))$$
- Eulers formula $e^{i\theta} = \cos(\theta) + i \sin(\theta)$

- Therefore

$$\cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2}$$



$$\sin(\theta) = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

Reminder: polynomials

- Any polynomial $P_r(x)$ of degree r can be written as

$$P_r(x) = a(x - z_1)\dots(x - z_r)$$

- where z_i are roots (real or complex)
- If z_i is a root, \bar{z}_i is also a root

Causal ARMA

Def: Linear process is **causal** and **nonexplosive** if

- $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$ (depends on the past only)
- $\sum_{j=0}^{\infty} |\psi_j| < \infty$
- We set $\psi_0 = 1$ by convention.

Property: ARMA(p,q) is **causal** iff roots $\phi(z') = 0$ are outside unit circle,
i.e. $|z'| > 1$

$$\phi(B)x_t = \theta(B)w_t$$

Causal ARMA

Example: Is the ARMA process below causal?

$$x_t = 0.4x_{t-1} + 0.3x_{t-2} + 0.2x_{t-3} + w_t - 0.1w_{t-1}$$
$$\Rightarrow \phi(B) = 1 - 0.4B - 0.3B^2 - 0.2B^3$$

```
> z=c(1, -0.4,-0.3,-0.2)
> polyroot(z)
[1] 1.060419-0.000000i -1.280210+1.753904i -1.280210-1.753904i
>
```

Invertible ARMA

Def: ARMA(p,q) is **invertible** if

- $w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$ (depends on the past only)
- $\sum_{j=0}^{\infty} |\pi_j| < \infty$

Property: ARMA(p,q) is **invertible** iff roots $\theta(z') = 0$ are outside unit circle, i.e. $|z'| > 1$

$$\phi(B)x_t = \theta(B)w_t$$

- $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} \rightarrow x_t = \psi(B)w_t$
- $w_t = \sum_{j=0}^{\infty} \pi_j w_{t-j} \rightarrow w_t = \pi(B)x_t$
- How to find coefficients in ψ and π → coefficient matching

$$\phi(z)\psi(z) = \theta(z) \quad \pi(z)\theta(z) = \phi(z)$$

- Example: $x_t = 0.4x_{t-1} + 0.45x_{t-2} + w_t + w_{t-1} + 0.25w_{t-2}$

```
> ARMAtoMA(ar=.9,ma=0.5, 6)
[1] 1.400000 1.260000 1.134000 1.020600 0.918540 0.826686
```

Home reading

- Shumway and Stoffer, section 3.1
- R code: arima.sim, arima, polyroot, ARMAtoMA, ARMAacf
 - ▶ Check carefully arima() docs to see how ar and ma coefficients are specified in the software

Time Series Analysis

Lecture 4: ARIMA models-1, Estimation

Tohid Ardestiri

Linköping University
Division of Statistics and Machine Learning

September 13, 2019



White noise

Simplest and most random time series: **white noise**

- w_t uncorrelated $E(w_t w_{t-h}) = 0$ for all $h \neq 0$

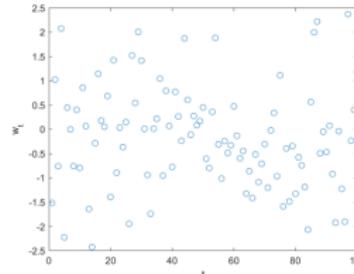
$$w_t \sim wn(0, \sigma_w^2)$$

- w_t white independent noise: independent and identically distributed
independence: $f(w_t, w_{t-h}) = f(w_t)f(w_{t-h})$

$$w_t \sim iid(0, \sigma_w^2)$$

- w_t white normal noise: independent and identically normal distributed

$$w_t \sim iidN(0, \sigma_w^2)$$



Autocovariance and ACF

- Autocovariance function

$$\gamma(s, t) = \text{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)]$$

Note $\text{var}(x_t) = \gamma(t, t)$

- Autocorrelation function (ACF)

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}}$$

Useful fact: If $U = \sum_{j=1}^m a_j x_j$ and

$$V = \sum_{k=1}^r b_k y_k$$

$$\text{cov}(U, V) = \sum_{j=1}^m \sum_{k=1}^r a_j b_k \text{cov}(x_j, y_k)$$

Stationarity

- Time series x_t is **weakly stationary (stationary)** if
 - ▶ $E x_t = \text{const}$
 - ▶ $\gamma(s, t) = \gamma(|s - t|)$
 - ▶ $\text{var}(x_t) < \infty$
- $\gamma(t, t + h) = \gamma(|t + h - t|) = \gamma(h)$
 - ▶ Autocovariance depends on lag only!
- Autocovariance for stationary process $\gamma(h) = \text{cov}(x_t, x_{t+h})$
- ACF for stationary process $\rho(h) = \frac{\gamma(h)}{\gamma(0)}$

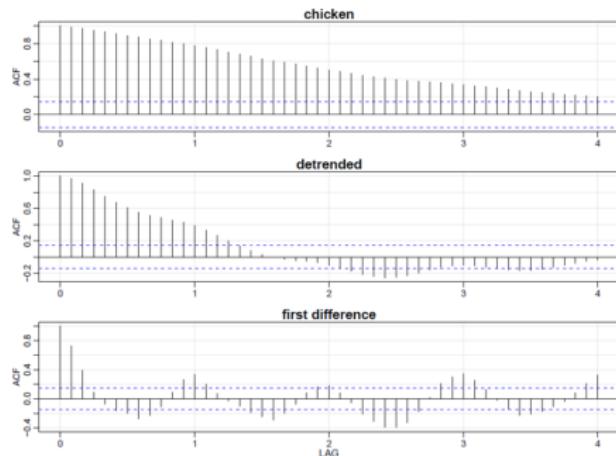
Sample ACF

Theorem: Under weak conditions, if x_t is white noise and $n \rightarrow \infty$ then $\hat{\rho}(h)$ is approximately $N(0, \frac{1}{n})$

Consequence: If some $|\hat{\rho}(h)| > \frac{2}{\sqrt{n}}$ then the time series is not a white noise (with approximately 95 % confidence).

Typical modeling strategy:

- Propose a model
- Fit a model
- Compute residuals
- Check ACF within $\pm \frac{2}{\sqrt{n}}$



Moving average models

- Moving average model of order q, MA(q)

$$1 \quad x_t = 1 w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$$

$$x_t = \sum_{j=0}^q \theta_j w_{t-j}$$

- $w_t \sim wn(0, \sigma_w^2)$
- $\theta_1, \dots, \theta_q$ constants, $\theta_q \neq 0$ and $\theta_0 = 1$

- Moving average operator

$$\theta(B) = \sum_{j=0}^q \theta_j B^j$$

- MA(q):

$$x_t = \theta(B)w_t$$

Autoregressive models

- Autoregressive model of order p , $AR(p)$

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t$$

$$x_t - \sum_{j=1}^p \phi_j x_{t-j} = w_t$$

- ▶ x_t is stationary if x_0 is sampled from the stationary distribution
- ▶ $w_t \sim \text{wn}(0, \sigma_w^2)$
- ▶ ϕ_1, \dots, ϕ_p constants, $\phi_p \neq 0$
- ▶ $E x_t = 0$ if $E x_0 = 0$

- Autoregressive operator

$$\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$$

- AR(p) model

$$\boxed{\phi(B)x_t = w_t}$$

ARMA models

- Autoregressive moving average ARMA(p,q)

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$$

- ▶ $\phi_p \neq 0, \theta_q \neq 0$
- ▶ Is stationary
- ▶ $E x_t = 0$ if $E x_0 = 0$

- p -autoregressive order, q -moving average order
- Alternative form

$$\phi(B)x_t = \theta(B)w_t$$

- Criteria for **causality** and **invertibility**

- ▶ Check roots of the characteristic polynomials $\phi(\cdot)$ and $\theta(\cdot)$

Property: ARMA(p,q) is **causal** iff **ALL** roots $\phi(z') = 0$ are outside unit circle, i.e. $|z'| > 1$

Property: ARMA(p,q) is **invertible** iff **ALL** roots $\theta(z') = 0$ are outside unit circle, i.e. $|z'| > 1$

Linear process

For a **linear process** x_t : $x_t = \mu + \sum_{j=-\infty}^{\infty} \psi_j w_{t-j} = \mu + \psi(B)w_t$
where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$,

$$\gamma_x(h) = \sigma_w^2 \sum_{j=-\infty}^{\infty} \psi_{j+h} \psi_j$$

Note: $x_t = \phi^{-1}(B)\theta(B)w_t = \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}$ But series might be non-convergent

- Coefficient matching **whiteboard**
- How to find coefficients in $\psi(B)$ → **coefficient matching**
- **Example:** $x_t = 0.4x_{t-1} + 0.45x_{t-2} + w_t + w_{t-1} + 0.25w_{t-2}$

```
> ARMAtoMA(ar=.9,ma=0.5, 6)
[1] 1.400000 1.260000 1.134000 1.020600 0.918540 0.826686
```

Differencing

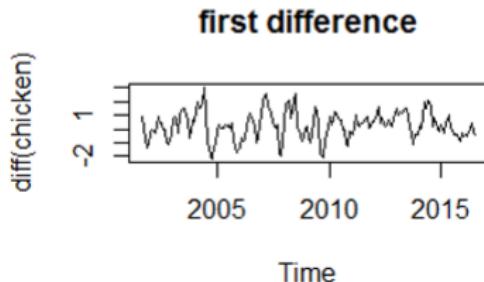
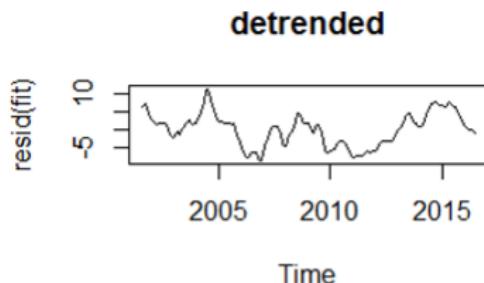
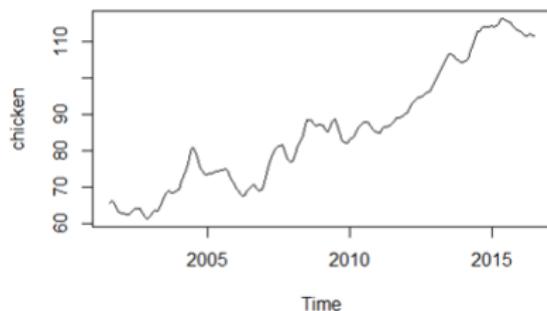
Assume $x_t = \mu_t + y_t$ and y_t stationary

Differencing gives

$$z_t = \nabla x_t = x_t - x_{t-1}$$

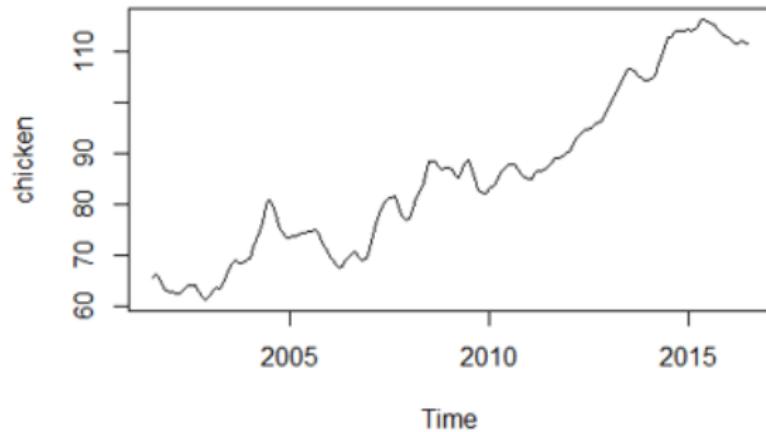
Also,

- $\nabla x_t = (1 - B)x_t$
- $\nabla^d = (1 - B)^d$



ARIMA models

- ARMA for stationary models
 - ▶ What if not stationary?



ARIMA models

- Differencing helps (lecture 2)
 - ▶ $\nabla x_t = x_t - x_{t-1}$ removes linear trend and random walk
 - ▶ $\nabla^d x_t$ removes polynomial of order d and some stochastic trends
 - ▶ → differencing is important modeling instrument!
- **Def:** x_t is ARIMA(p,d,q) if $\nabla^d x_t$ is ARMA(p,q), i.e.

$$\phi(B)(1 - B)^d x_t = \theta(B)w_t$$

- For nonzero mean $E(\nabla^d x_t) = \mu$,

$$\phi(B)(1 - B)^d x_t = \theta(B)w_t + \delta$$

$$\delta = \mu(1 - \phi_1 - \dots - \phi_p)$$

ARIMA models

- Notation: $p=0 \rightarrow \text{IMA}(d,q)$, $q=0 \rightarrow \text{ARI}(p,d)$
- Estimation: Differentiate + fit ARMA
- Forecasting:
 - ▶ Transform data $y_t = \nabla^d x_t$ and forecast ARMA(p,q)
 - ▶ Solve $(1 - B)^d x_t^n = y_t^n$

Estimation

Consider **ARIMA(p,d,q)**

$$\phi(B)(1 - B)^d x_t = \theta(B)w_t$$

- What are the unknowns?
 - ▶ Orders p , d and q
 - ▶ Parameters ϕ_1, \dots, ϕ_p and $\theta_1, \dots, \theta_q$
 - ▶ variance σ_w^2 where $w_t \sim N(0, \sigma_w^2)$
- How to estimate these?
- Assumption: Let us assume for now that we know p , d and q
 - ▶ Maximum likelihood (ML) estimate
 - ▶ Least squares

Maximum likelihood estimation: reminder

Let $x \sim f(x|\alpha)$

- Likelihood of α given observations x_1, \dots, x_t is

$$L(\alpha) = f(x_1, \dots, x_t | \alpha)$$

- Maximum likelihood: Optimal α

$$\hat{\alpha} = \arg \max_{\alpha} L(\alpha)$$

- Independent observations: $x_i \stackrel{iid}{\sim} f(x_i | \alpha)$
- $L(\alpha) = \prod_i f(x_i | \alpha)$
- Negative log-likelihood $I(\alpha) = -\sum_i \log(f(x_i | \alpha))$
- Maximum likelihood α can be obtained from negative log-likelihood

$$\max_{\alpha} L(\alpha) = \min_{\alpha} I(\alpha)$$

Maximum likelihood estimation: reminder

Time series data are NOT independent

- Likelihood of α given observations x_1, \dots, x_t is

$$L(\alpha) = f(x_1, \dots, x_t | \alpha)$$

- Maximum likelihood: Optimal α

$$\hat{\alpha} = \arg \max_{\alpha} L(\alpha)$$

- Dependent data (time series): chain rule

$$L(\alpha) = f(x_1 | \alpha) f(x_2 | \alpha, x_1) f(x_3 | \alpha, x_2, x_1) \dots$$

- Negative log-likelihood $I(\alpha) = - \sum_i \log(f(x_i | \alpha, x_{i-1}, \dots))$
- Maximum likelihood: Optimal α

$$\max_{\alpha} L(\alpha) = \min_{\alpha} I(\alpha)$$

Maximum likelihood estimation: reminder

- Normal distributions: if $x_i \sim N(\mu, \sigma^2)$, iid.

$$L(\theta) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{\sum_i(x_i-\mu)^2}{2\sigma^2}}$$

- Maximum likelihood

$$\hat{\mu} = \bar{x}$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_i (x_i - \bar{x})^2$$

- For ARMA models, assume normality of w_t !
- Negative log-likelihood

$$I(\mu, \phi, \sigma_w^2) = \frac{S(\mu, \phi)}{2\sigma_w^2} + \frac{n}{2} \log(2\pi\sigma_w^2) - \frac{1}{2} \log(1 - \phi^2)$$

$$S(\mu, \phi) = (1 - \phi^2)(x_1 - \mu)^2 + \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2$$

- How to find optimum?

- For σ^2 explicit

$$\hat{\sigma}_w^2 = \frac{1}{n} S(\hat{\mu}, \hat{\phi})$$

- Otherwise numerical optimization (unconstrained optimization)

Optimization methods

- Examples:
 - ▶ Steepest descent
 - ▶ Newtons Methods
 - ▶ Gauss-Newton methods
 - ▶ (least squares)
 - ▶ ...

Least squares

- **Unconditional least squares**

- Estimate by numerical methods or sometimes analytically

$$\min_{\mu, \phi} S(\mu, \phi)$$

- **Conditional least squares:** assume x_1 given (constant)

$$\min \sum_{i=1}^t w_i^2$$

- For AR(1), $\sum_{i=1}^t w_i^2 = S_c(\mu, \phi)$

$$S_c(\mu, \phi) = \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2 = \sum_{t=2}^n [x_t - \alpha - \phi x_{t-1}]^2$$

- **Note:** Minimize by doing regression $Y = x_t, X = \text{lag}(x_t)$

Home reading

- Shumway and Stoffer, parts of sections 3.5, 3.6, 3.7
- R code: arima.sim, arima, polyroot, ARMAtoMA, ARMAacf

Time Series Analysis

Lecture 5: ARIMA models-2

Estimation, PACF, Forecasting

Tohid Ardesthiri

Linköping University
Division of Statistics and Machine Learning

September 16, 2019



Maximum likelihood estimation: reminder

Time series data are NOT independent

- Likelihood of α given observations x_1, \dots, x_t is

$$L(\alpha) = f(x_1, \dots, x_t | \alpha)$$

- Maximum likelihood:** Optimal α

$$\hat{\alpha} = \arg \max_{\alpha} L(\alpha)$$

- Dependent data (time series):** chain rule

$$L(\alpha) = f(x_1 | \alpha) f(x_2 | \alpha, x_1) f(x_3 | \alpha, x_2, x_1) \dots$$

- Negative log-likelihood $I(\alpha) = - \sum_i \log(f(x_i | \alpha, x_{i-1}, \dots))$
- Maximum likelihood:** Optimal α

$$\max_{\alpha} L(\alpha) = \min_{\alpha} I(\alpha)$$

Maximum likelihood estimation: reminder

- Normal distributions: if $x_i \sim N(\mu, \sigma^2)$, iid.

$$L(\theta) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{\sum_i(x_i-\mu)^2}{2\sigma^2}}$$

- Maximum likelihood

$$\hat{\mu} = \bar{x}$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_i (x_i - \bar{x})^2$$

- For ARMA models, assume normality of w_t !
- Negative log-likelihood

$$I(\mu, \phi, \sigma_w^2) = \frac{S(\mu, \phi)}{2\sigma_w^2} + \frac{n}{2} \log(2\pi\sigma_w^2) - \frac{1}{2} \log(1 - \phi^2)$$

$$S(\mu, \phi) = (1 - \phi^2)(x_1 - \mu)^2 + \sum_{t=2}^n [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2$$

- How to find optimum?

- For σ^2 explicit

$$\hat{\sigma}_w^2 = \frac{1}{n} S(\hat{\mu}, \hat{\phi})$$

- Otherwise numerical optimization (unconstrained optimization)

ARMA

- **Autoregressive moving average ARMA(p, q)**

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$$

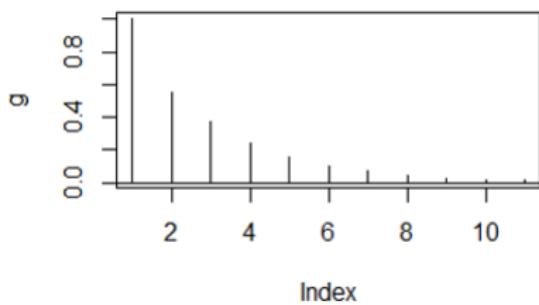
- ▶ $\phi_p \neq 0, \theta_q \neq 0$
- ▶ Is stationary
- ▶ $E x_t = 0$

- ACF for AR(1), MA(1), MA(2)

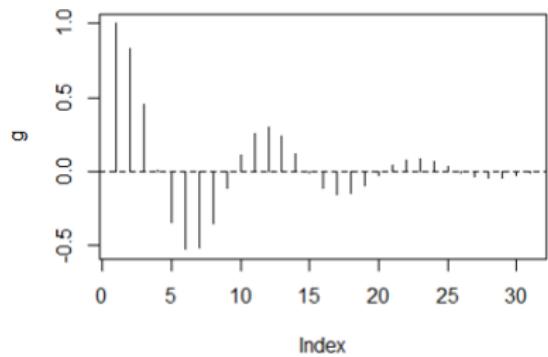
→ how to compute ACF for a general ARMA?

ACF for AR(2)

$$\phi^1 = 0.5 \quad \phi^2 = 0.1$$



$$\phi^1 = 1.5 \quad \phi^2 = -0.8$$



ACF for AR(p), MA(p)

- AR(p): using difference equations
- MA(q): using difference equations

$$\rho(h) = \begin{cases} \frac{\sum_{j=0}^{q-h} \theta_j \theta_{j+h}}{1+\theta^2+\dots+\theta_q^2} & 0 \leq h \leq q \\ 0 & h > q \end{cases}$$

ACF for ARMA(p,q)

- ARMA(p,q):

$$\phi(B)x_t = \theta(B)w_t$$

- Causal ARMA: $x_t = \phi^{-1}(B)\theta(B)w_t = \psi(B)w_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$
 - ▶ How to find ψ_j in practice? Coefficient matching
- **Theorem:** ACF of ARMA(p,q) can be found by solving general homogeneous equations:

$$\gamma(h) - \phi_1\gamma(h-1) - \dots - \phi_p\gamma(h-p) = 0, \quad h \geq \max(p, q+1)$$

- ▶ Initial conditions

$$\gamma(h) - \phi_1\gamma(h-1) - \dots - \phi_p\gamma(h-p) = \sigma_w^2 \sum_{j=h}^q \theta_j \psi_{j-h}, \quad 0 \leq h < \max(p, q+1)$$

ACF for ARMA(1,1)

- Show for ARMA(1,1)

$$\rho(h) = \frac{(1 + \theta\phi)(\phi + \theta)}{1 + 2\theta\phi + \theta^2} \phi^{h-1}, h \geq 1$$

- **Note:** pattern similar to AR(1) → hard to differentiate
- **Note:** ACF is 0 for $h > q$ from MA(q) → MA(q) is identifiable from ACF
- **How to differentiate between AR(p)? ARMA(p)?**

Partial correlation

A Gaussian intuition:

- Conditional density: $f(x, y|z) = \frac{f(x, y, z)}{f(z)}$
- if x , y and z were jointly normal then

$$f(x, y|z) = N\left(\begin{bmatrix} x \\ y \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

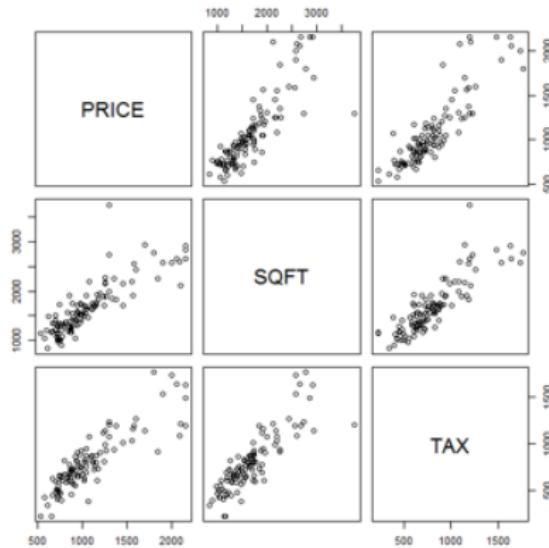
- Also,

$$\rho_{xy|z} = \frac{\text{cov}(x, y|z)}{\sqrt{\text{var}(x|z) \text{var}(y|z)}} = \frac{\Sigma_{12}}{\sqrt{\Sigma_{11}\Sigma_{22}}}$$

- **What if $\Sigma_{12} = 0$?**

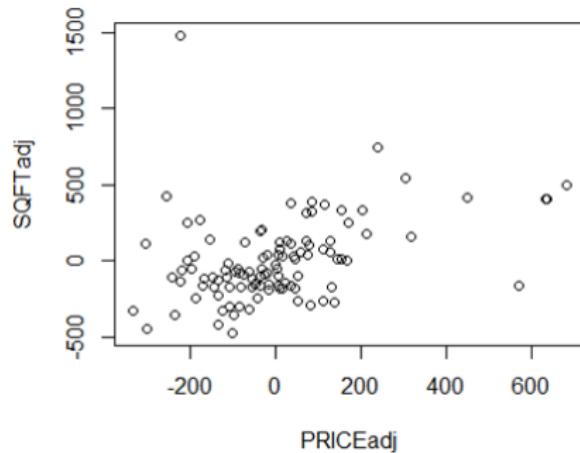
Partial autocorrelation

- **Example:** Albuquerque home prices
 - ▶ What if we remove information stored in TAX from PRICE and SQFT?



Partial autocorrelation

- $\hat{y} = \hat{\alpha}_0 + \hat{\alpha}_1 z$
- $\hat{x} = \hat{\beta}_0 + \hat{\beta}_1 z$
- $x' = x - \hat{x}$
- $y' = y - \hat{y}$
- **Partial autocorrelation**



- If we know α , β and z , we can reduce connection between x and y

```
> corr(cbind(PRICEadj,SQFTadj))  
[1] 0.3675204
```

PACF

- Partial autocorrelation function (PACF) for a stationary process

$$\phi_{11} = \text{corr}(x_{t+1}, x_t)$$

$$\phi_{hh} = \text{corr}(x'_{t+h}, x''_t), \quad h > 1$$

- ▶ where $x'_{t+h} = x_{t+h} - \sum_{j=1}^{h-1} \hat{\beta}_j x_{t+h-j}$
- ▶ and $x''_t = x_t - \sum_{j=1}^{h-1} \hat{\beta}_j x_{t+j}$
- ▶ **Note:** coefficients in x''_{t+h} and x'_{t+h} are the same (stationarity)
- **Example:** AR(1) $\phi_{11} = \phi, \phi_{22} = 0$

PACF for AR(p)

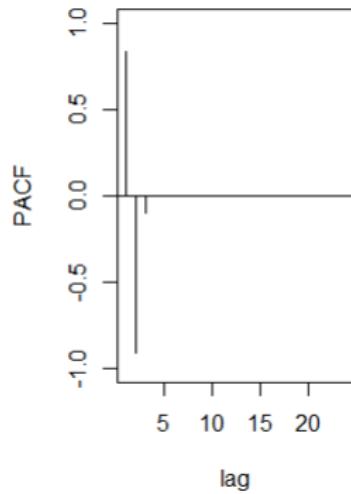
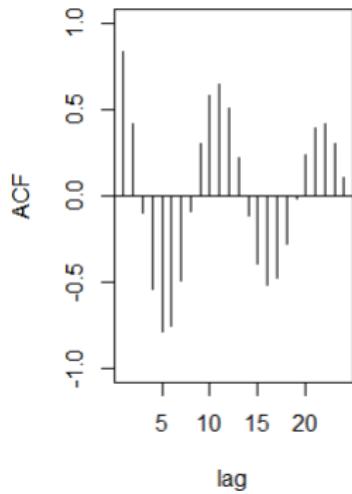
$$x_t = \sum_{j=1}^p \phi_j x_{t-j} + w_t$$

- It can be shown:
 - ▶ $\phi_{pp} = \phi_p$
 - ▶ $\hat{\beta}_1 = \phi_1, \dots, \hat{\beta}_p = \phi_p, \hat{\beta}_{p+1} = 0, \dots, \hat{\beta}_h = 0$ for $h > p$
- It means

$$\begin{aligned}\phi_{hh} &= \text{cov}(x_{t+h} - \sum_{j=1}^p \phi_j x_{t+h-j}, x_t - \sum_{j=1}^p \phi_j x_{t+j}) \\ &= \text{cov}(w_{t+h}, x_t - \sum_{j=1}^p \phi_j x_{t+j}) = 0, \quad \text{when } h > p\end{aligned}$$

PACF for AR(p)

- Example: AR(3) $\phi_1 = 1.5$, $\phi_2 = -0.75$, $\phi_3 = -0.1$

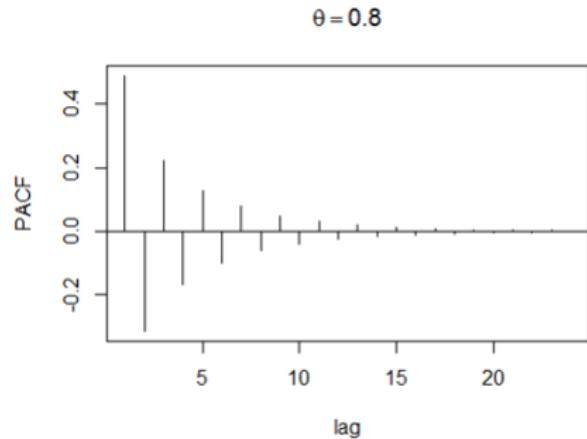


PACF for MA(1)

- If invertible,

$$\phi_{hh} = -\frac{(-\theta)^h(1-\theta^2)}{1-\theta^{2h+2}}, \quad h \geq 1$$

Decreases exponentially with h



ACF and PACF

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

How to differentiate between ARMA(p, q)?

Empirical ACF (EACF)

Idea:

- ARMA(p,q): $x_t = \sum_{j=1}^p \phi_j x_{t-j} + \sum_{j=1}^q \theta_j w_{t-j} + w_t$
- If we can estimate $\phi_j \rightarrow x'_t = x_t - \sum_{j=1}^p \phi_j x_{t-j}$ is linear function in w_t, \dots, w_{t-q}
- If we run regression x'_t against $w_t \dots w_{t-j}$:
 - ▶ Residuals are white noise, $j \geq q \rightarrow$ ACFs not significant
 - ★ Some of the coefficients will be 0
 - ▶ Residuals are not white noise, $j < q \rightarrow$ ACFs significant
 - ▶ Note: w_t s substituted by lagged residuals from a series of regressions
- If $x'_t = x_t - \sum_{j=1}^k \phi_j x_{t-j}, k < p \rightarrow$ white noise will never be achieved
 \rightarrow ACFs are not zero

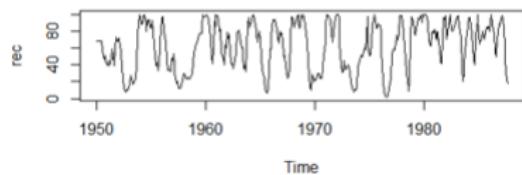
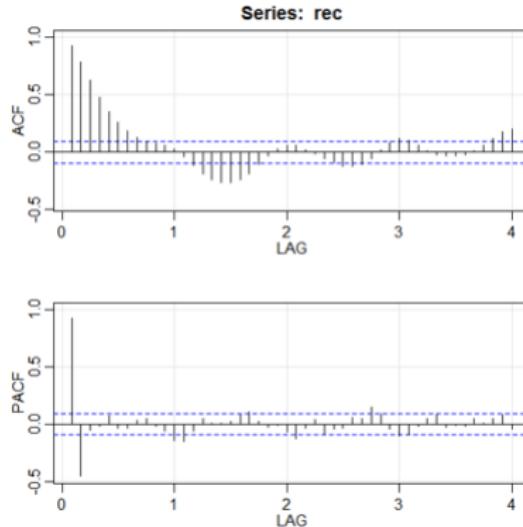
Empirical ACF (EACF)

- $k > p$ General result: ACFs are 0 for $j > q + (k - p)$
 - ▶ Example: ARMA(0,1)
- General conclusion for AR,MA = (k,j):
 - ▶ This is theoretical one! → not exactly the same for the samples

AR/MA	0	1	2
0	X	X	X	X	X	X	X
1	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X
...	X	X	X	X	X	X	X
...	X	X	X	X	X	X	X
...	X	X	0	0	0	0	0
...	X	X	X	0	0	0	0
...	X	X	X	X	0	0	0
...	X	X	X	X	X	0	0

ARMA orders

- Recruitment series



Conclusion?

ARMA orders

- EACF

```
> TSA::eacf(rec)
```

AR/MA

0 1 2 3 4 5 6 7 8 9 10 11 12 13

0 x x x x x x x o o o o o x

1 x x x o o o o o o o o o o o

2 o o x x o o o o o o o o o o o

3 x o o x o o o o o o o o o o o

4 x x o o o o o o o o o o o o o

5 x x x o o o o o o o o o o o o

6 x x x o o o o o o o o o o o o

7 x x o o o o o o o o o o o o x o

ARMA orders

- AR(2) and ARMA(1,3)

- Conclusions?

```
> arima(rec, order=c(2,0,0))

Call:
arima(x = rec, order = c(2, 0, 0))

Coefficients:
          ar1      ar2  intercept
        1.3512 -0.4612    61.8585
  s.e.  0.0416  0.0417     4.0039

sigma^2 estimated as 89.33:  log likelihood = -1661.51,  aic = 3331.02
> arima(rec, order=c(1,0,3))

Call:
arima(x = rec, order = c(1, 0, 3))

Coefficients:
          ar1      ma1      ma2      ma3  intercept
        0.7826  0.5484  0.3239  0.2119    61.8609
  s.e.  0.0390  0.0554  0.0621  0.0530     4.1953

sigma^2 estimated as 88.43:  log likelihood = -1659.24,  aic = 3330.48
> |
```

Model selection

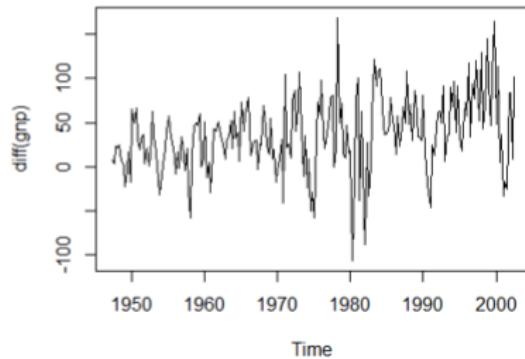
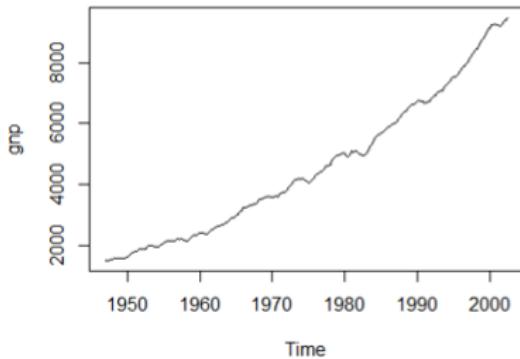
- Which model is suitable?
 - ▶ What is p, d, q is ARIMA(p,d,q)?
 - ▶ d is defined before!
 - ▶
- Step 1: Check ACF, PACF and EACF to define a few tentative models

Model selection

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

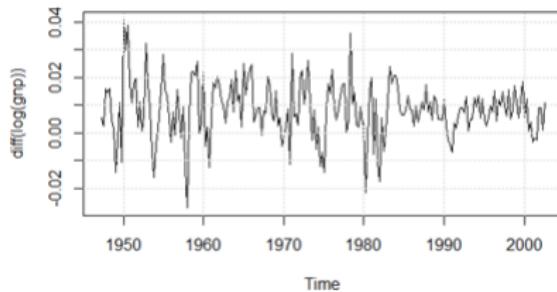
Model selection

- **Example:** GNP data
 - ▶ Trying differencing → non-constant variance and maybe trend? → transformation



Model selection

- Example: GNP data
 - ▶ Taking log and then differencing → still not perfect, but ... keep it as is.



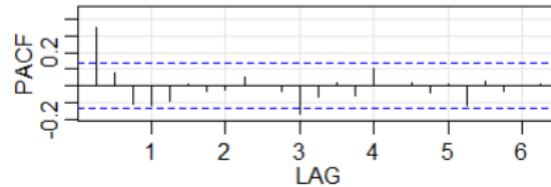
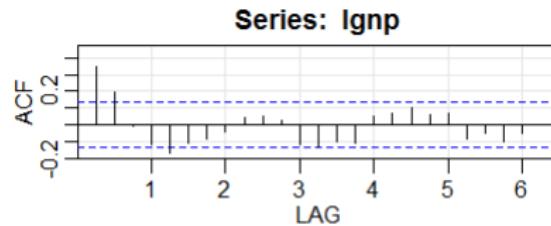
```
> adf.test(lgnp)

Augmented Dickey-Fuller Test

data: lgnp
Dickey-Fuller = -6.1756, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

Model selection

- Example: GNP data
 - ▶ Testing ACF and PACF



Conclusion?

Model selection

- Example: GMP data
 - ▶ Checking EACF

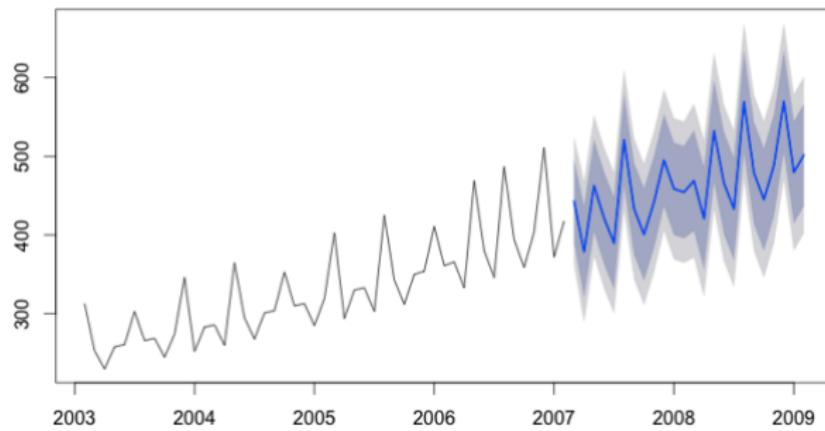
```
> TSA::eacf(lgnp)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x o o x o o o o o o o o o o
1 x x o o o o o o o o o o o o o
2 x x o o o o o o o o o o o o o
3 x o x o o o o o o o o o o o o
4 x o x o o o o o o o o o o o o
5 o x x o x o o o o o o o o o o
6 x x x x x o o o o o o o o o o
7 x x o x o o x o o o o o o o o
```

Conclusion?

Forecasting

- We have our series $x_1 \dots x_n$
- Use series to predict m steps ahead: x_{n+m}^n should be based on our observed data $x_{n+m}^n = g(x_1, \dots, x_n)$

Forecasts from ARIMA(0,0,1)(1,1,0)[12] with drift



Forecasting

- Assume $g(x_1, \dots, x_n) = \alpha_0 + \sum_{k=1}^n \alpha_k x_k$
 - ▶ Best linear predictors
- How to find α 's?

$$\min E[(x_{n+m} - g(x_1, \dots, x_n))^2]$$

- Prediction equations
 - ▶ Find α 's by solving ($x_0 = 1$)
$$E[(x_{n+m} - x_{n+m}^n)x_k] = 0, k = 0, \dots, n$$
- **Note:** $n+1$ equations, $n+1$ unknowns

One-step-ahead

- Denote $x_{n+1}^n = \phi_{n1}x_n + \dots + \phi_{nn}x_1$
- Prediction equations give

$$\Gamma_n \phi_n = \gamma_n$$

$$\Gamma_n = \begin{pmatrix} \gamma(1-1) & \gamma(2-1) & \dots & \gamma(n-1) \\ \gamma(2-1) & \gamma(2-2) & \dots & \gamma(n-2) \\ \dots & \dots & \dots & \dots \\ \gamma(n-1) & \gamma(n-2) & \dots & \gamma(n-n) \end{pmatrix}$$

$$\phi_n = \begin{pmatrix} \phi_{n1} \\ \dots \\ \phi_{nn} \end{pmatrix} \quad \gamma_n = \begin{pmatrix} \gamma_1 \\ \dots \\ \gamma_n \end{pmatrix}$$

- **Note:** for ARMA models Γ_n is positive def \rightarrow unique solution

One-step-ahead

- Causal AR(p): for $n \geq p$ best linear prediction is

$$x_{n+1}^n = \phi_1 x_n + \dots + \phi_p x_{n-p+1}$$

- In general, solve system of equations $\rightarrow O(n^3)$ operations
- Much faster algorithms exist
 - ▶ Durbin-Levinson algorithm
 - ▶ Innovations algorithm
- **Property:** PACF of a stationary process can be obtained as ϕ_{nn} by solving $\Gamma_n \phi_n = \gamma_n$

One-step-ahead

- Mean square prediction error (MSPE)

$$P_{n+1}^n = E[(x_{n+1} - x_{n+1}^n)^2] = \gamma(0) - \gamma_n' \Gamma_n^{-1} \gamma_n$$

- Confidence intervals for x_{n+1}

$$x_{n+1}^n \pm \alpha \sqrt{P_{n+1}^n}$$

- m-step ahead in general? Prediction equations
 - ▶ Difficult in general

Read home

- Ch 3.2-3.4
- Paper "Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation" by Tsay and Tiao
- R code: eacf in TSA package

m-step-ahead for ARMA

- Assume causal and invertible ARMA(p,q)
- Finite past prediction

$$x_{n+1}^n = E(x_{n+1}|x_n, \dots, x_1)$$

- Infinite past prediction

$$\tilde{x}_{n+m}^n = E(x_{n+m}|x_n, \dots, x_1, x_0, x_{-1}, \dots)$$

- m-step-ahead forecast for infinite past

- ▶ Compute recursively

$$\tilde{x}_{n+m} = - \sum_{j=1}^{m-1} \pi_j \hat{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j \tilde{x}_{n+m-j}, \quad m = 1, 2, \dots$$

- m-step ahead prediction error: $P_{n+m}^n = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2$

Long-range forecasts

- What if $m \rightarrow \infty$?

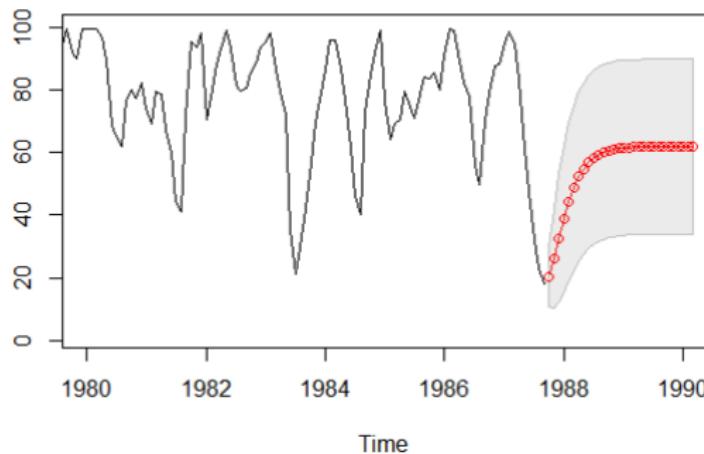
$$\tilde{x}_{n+m} \rightarrow 0(\text{or } \mu)$$

$$P_{n+m}^n \rightarrow \sigma_x^2$$

m-step-ahead

- Recruitment, AR(2)

$$x_{n+m}^n \pm 2\sqrt{P_{n+m}^n}$$



Truncated prediction

- Ignore non-positive j in x_j

$$\tilde{x}_{n+m} = - \sum_{j=1}^{m-1} \pi_j \tilde{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j x_{n+m-j}, m = 1, 2, \dots$$

- For ARMA, truncated prediction formula:
 - Recursive computation, explicit

$$\tilde{x}_{n+m}^n = \phi_1 \tilde{x}_{n+m-1}^n + \dots + \phi_p \tilde{x}_{n+m-p}^n + \theta_1 \tilde{w}_{n+m-1}^n + \dots + \theta_q \tilde{w}_{n+m-q}^n$$

$$\tilde{w}_t^n = \tilde{x}_t^n - \phi_1 \tilde{x}_{t-1}^n - \dots - \phi_p \tilde{x}_{t-p}^n - \theta_1 \tilde{w}_{t-1}^n - \dots - \theta_q \tilde{w}_{t-q}^n$$

- Boundary conditions: $\tilde{x}_t^n = x_n, 1 \leq t \leq n, \tilde{x}_t^n = 0, t \leq 0$

$$\tilde{w}_t^n = 0, t \leq 0 \quad \text{or } t > n$$

Time Series Analysis

Lecture 6: ARIMA models summary

State space models

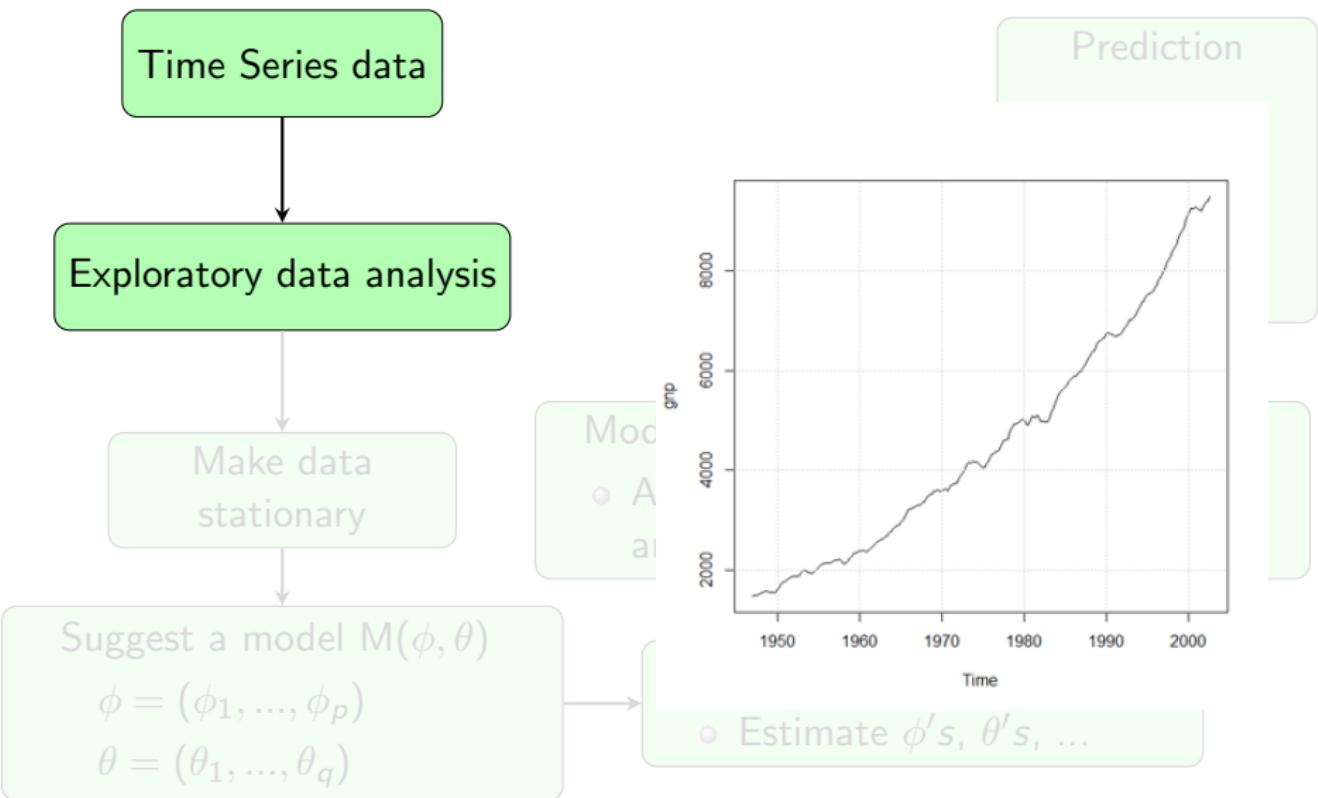
Tohid Ardesthiri

Linköping University
Division of Statistics and Machine Learning

September 27, 2019



Time domain: The Big Picture



Time domain: The Big Picture

Time Series data

$$Y_t = \nabla(\log(X_t))$$

Prediction

Exploratory data analysis

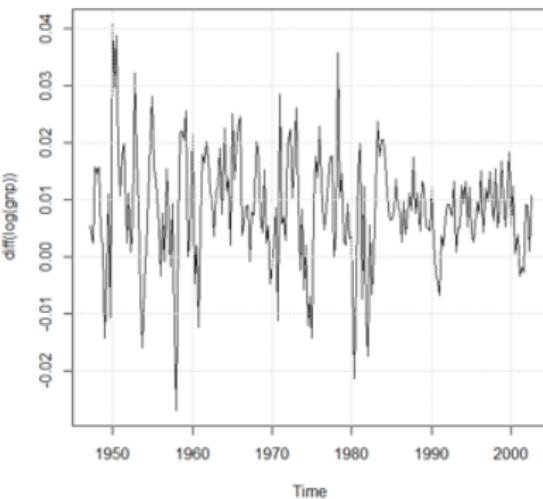
Make data stationary

Suggest a model $M(\phi, \theta)$

$$\phi = (\phi_1, \dots, \phi_p)$$

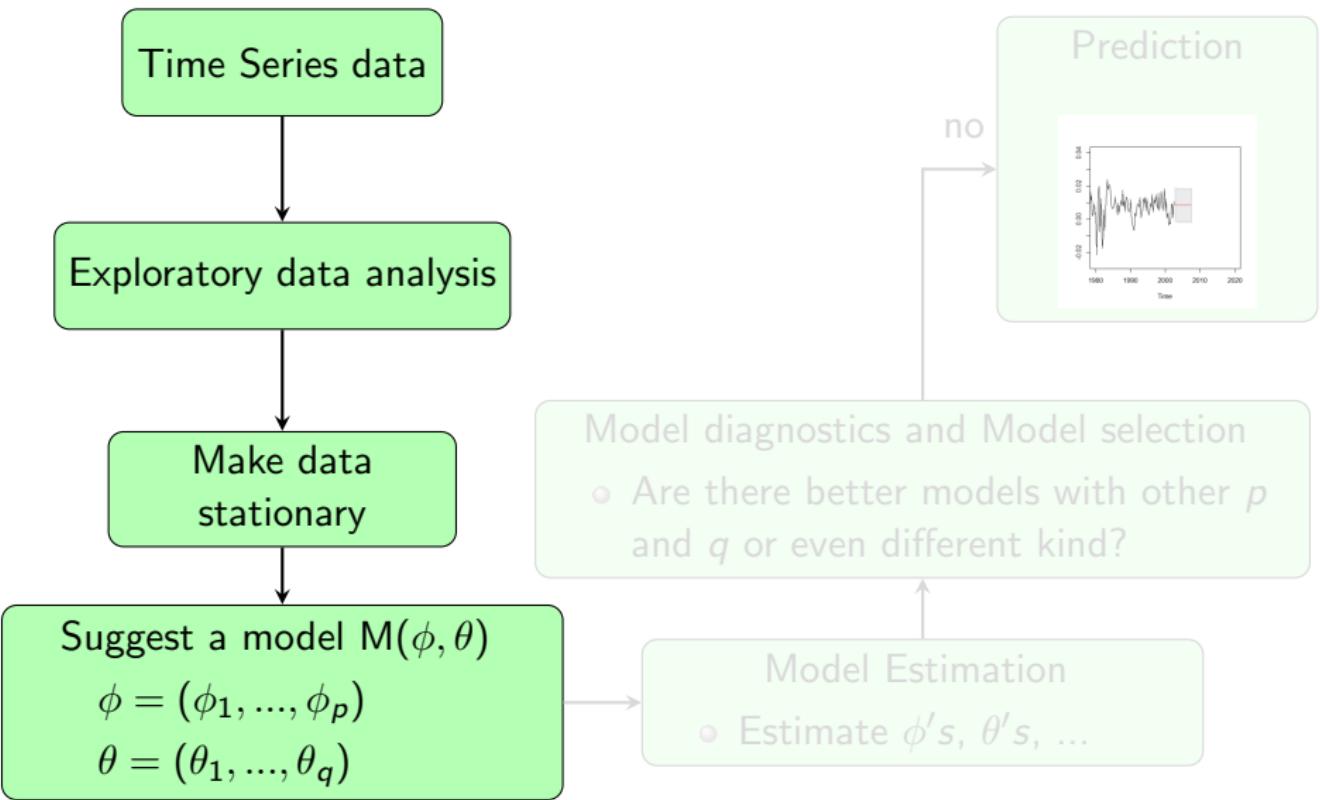
$$\theta = (\theta_1, \dots, \theta_q)$$

Model
A
ai

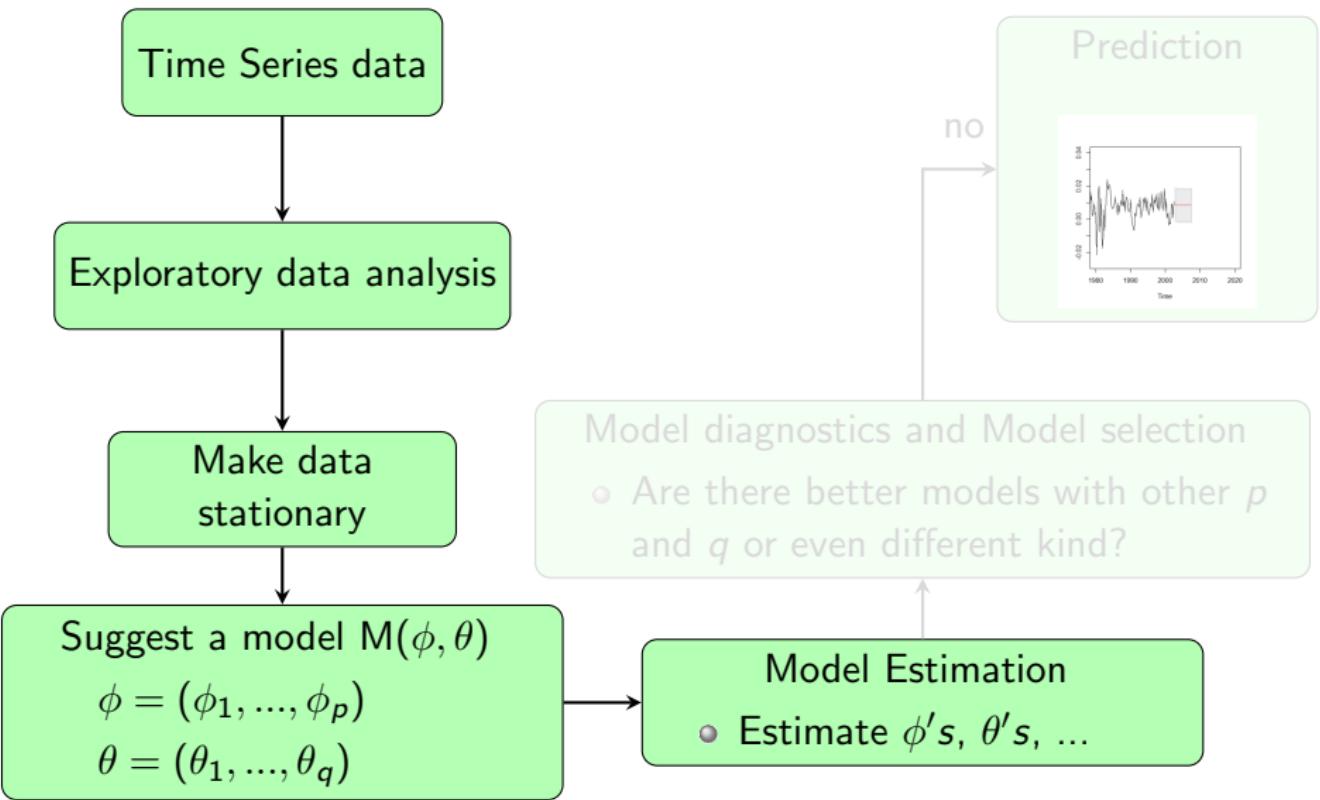


Estimate ϕ 's, θ 's, ...

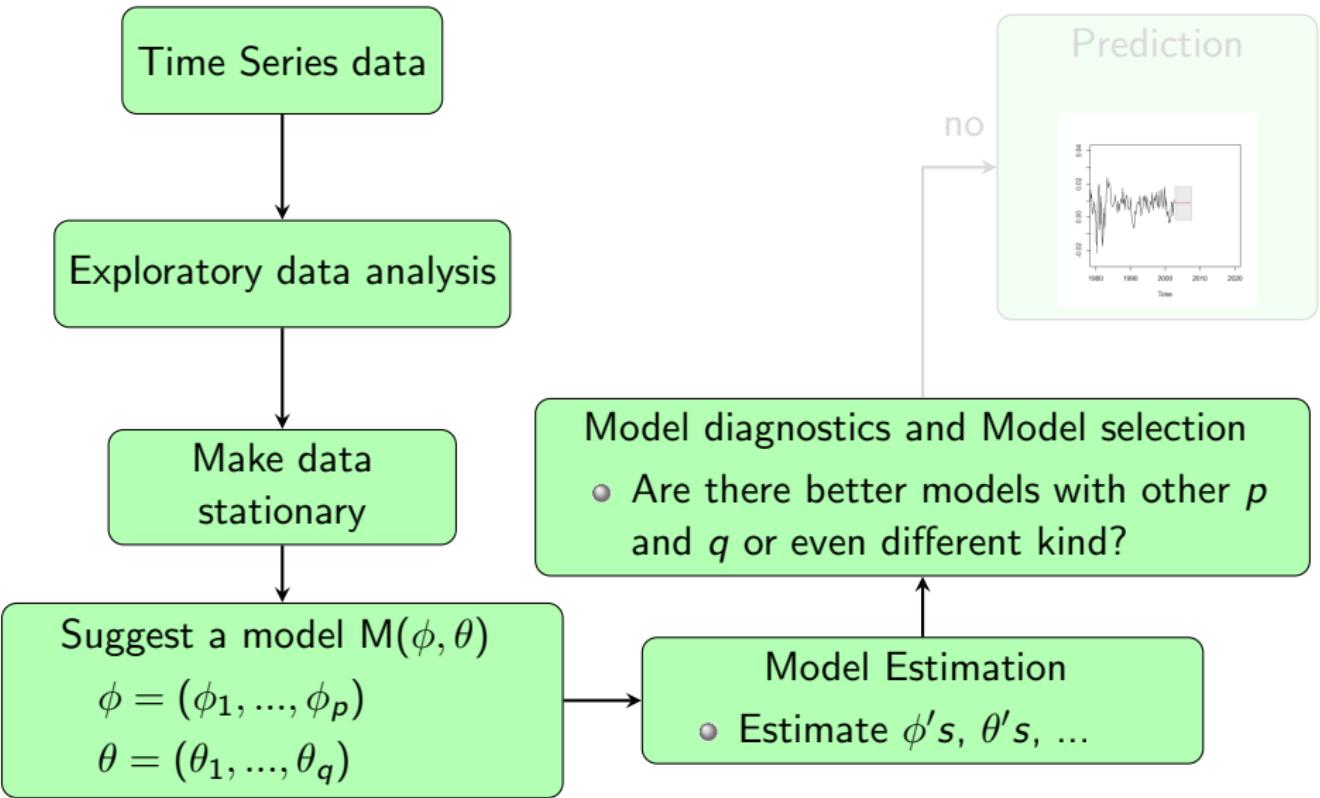
Time domain: The Big Picture



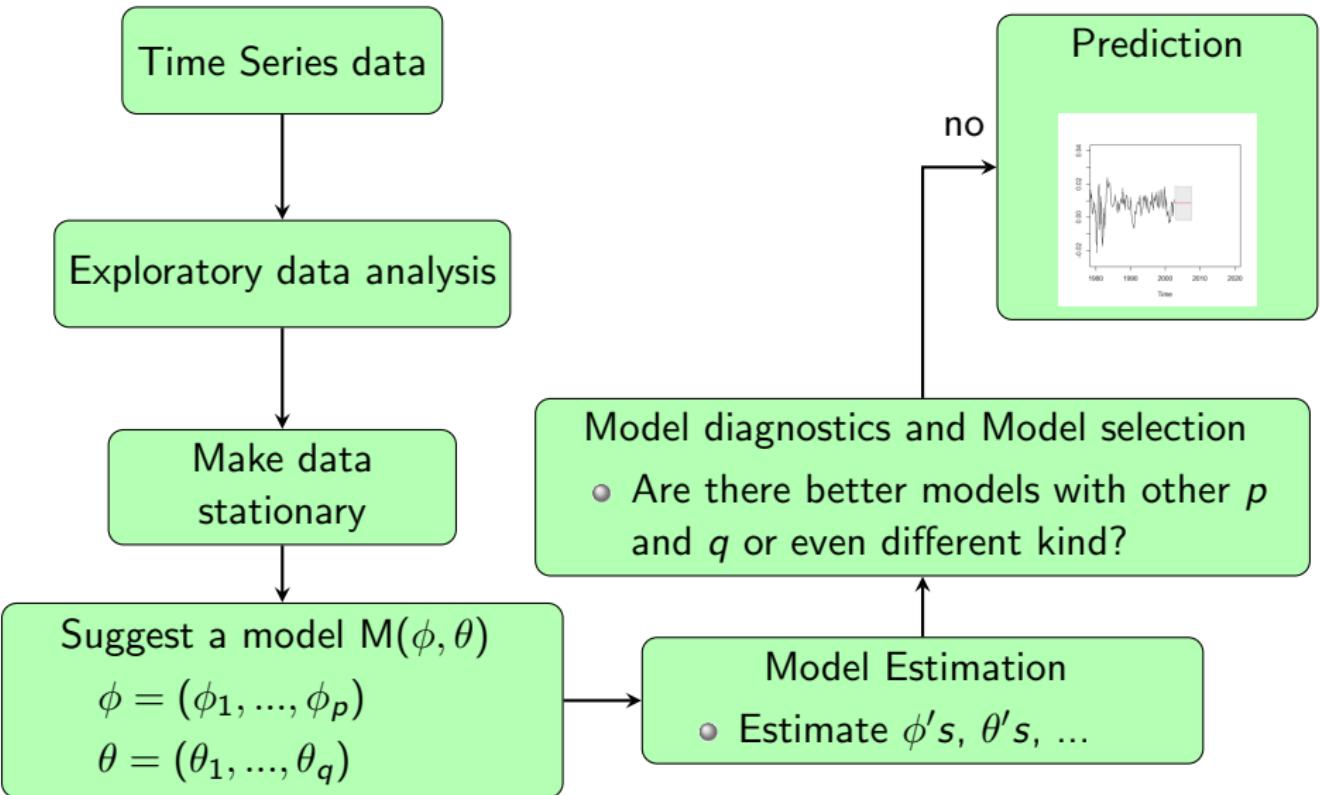
Time domain: The Big Picture



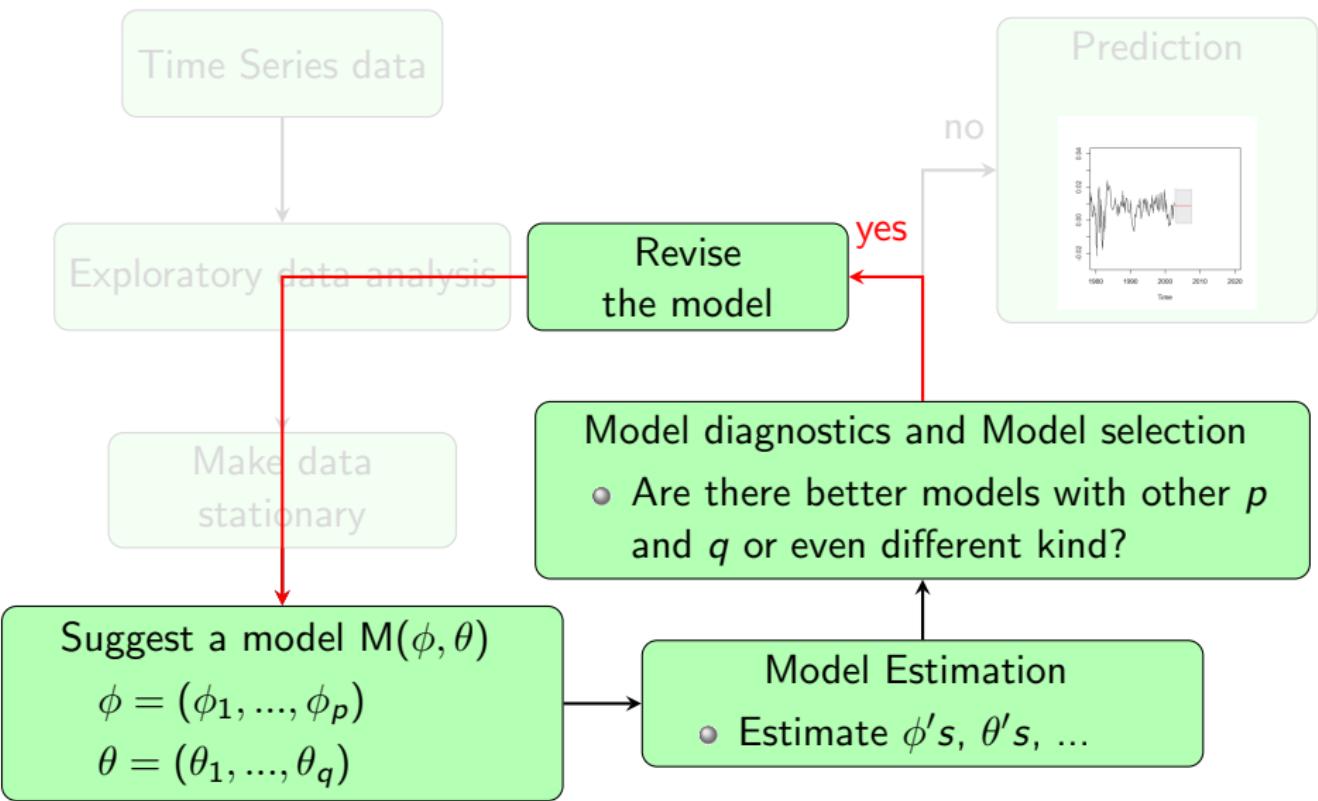
Time domain: The Big Picture



Time domain: The Big Picture



Time domain: The Big Picture



Model selection

Fit the tentative models, compare them

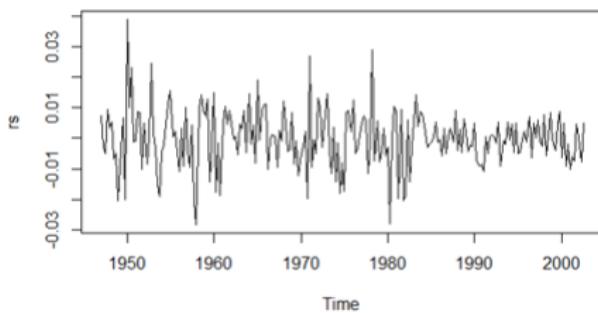
- Analytical measures: AIC, BIC
 - ▶ Penalize models with many parameters → simpler models
- Residual analysis

Residual analysis

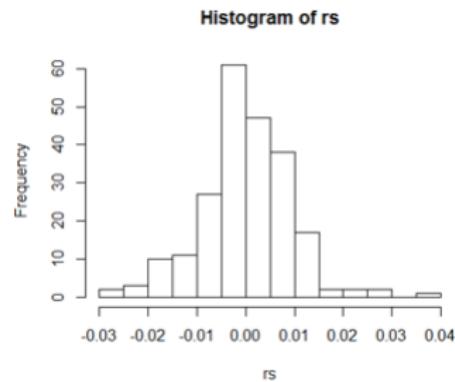
- Residuals $r_t = x_t - \hat{x}_t^{t-1}$? they are innovations
 - ▶ Note: computed from one-step-ahead predictions!
 - ▶ Measures predictive quality of the model (compare OLS)
- Residual analysis
 - ▶ Visual inspection: stationary? Patterns?
 - ▶ Histograms, Q-Q plots
 - ▶ ACF, PACF
 - ▶ Runs test
 - ▶ Box-Ljung test

Residual analysis - Visual inspection

Histogram and visual inspection

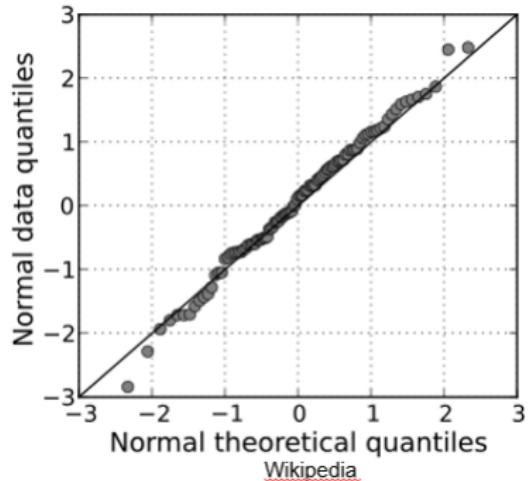
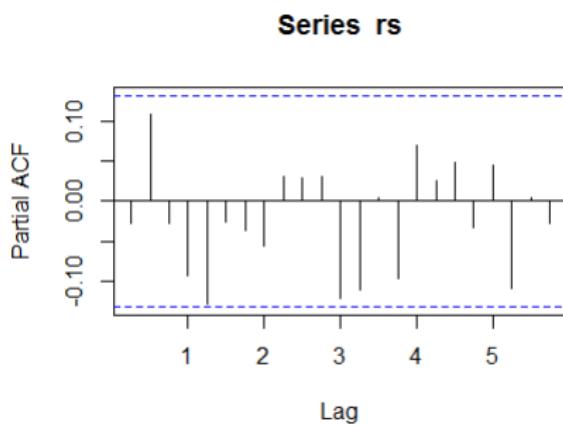


If looks white is good



If looks Normal is good

Residual analysis - ACF /PACF Q-Q plots



If between the blue lines good

If along the diagonal line GOOD

Statistical tests

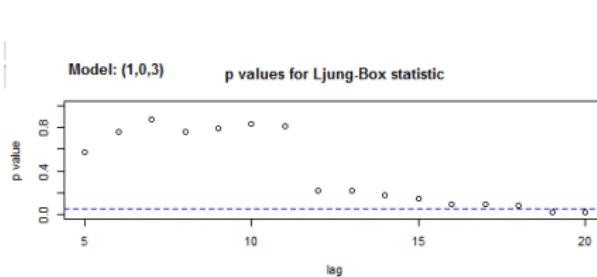
Tests are used to test independence

Runs test

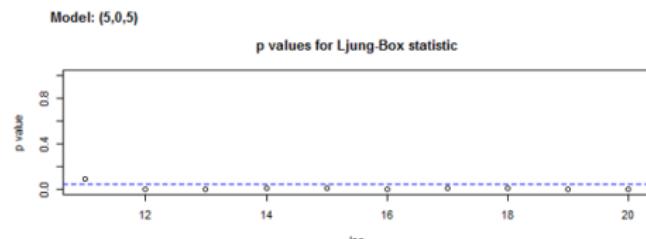
- H_0 : x_t values are i.i.d. **p-value NOT small**
- H_a : x_t values are not i.i.d. **p-value small**

Box-Ljung test

- H_0 : data are independent **p-value NOT small**
- H_a : data are not independent **p-value small**



GOOD



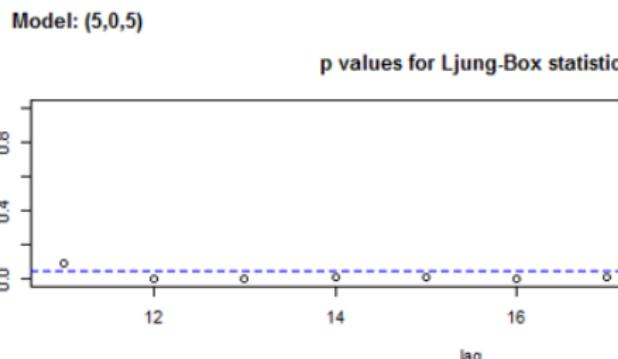
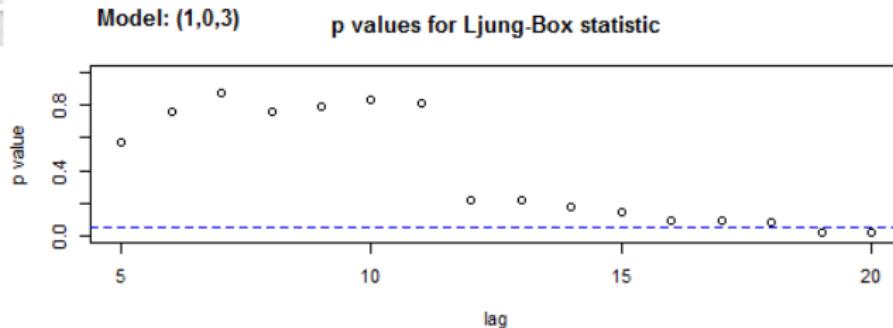
BAD

Overfitting

- Occams razor: among equally good models, choose the simplest one
- Overfitting: taking too complex models leads to bad predictions
- If ARIMA(p, d, q) has almost the same predictive quality as ARIMA(p', d', q') , take the one with less parameters

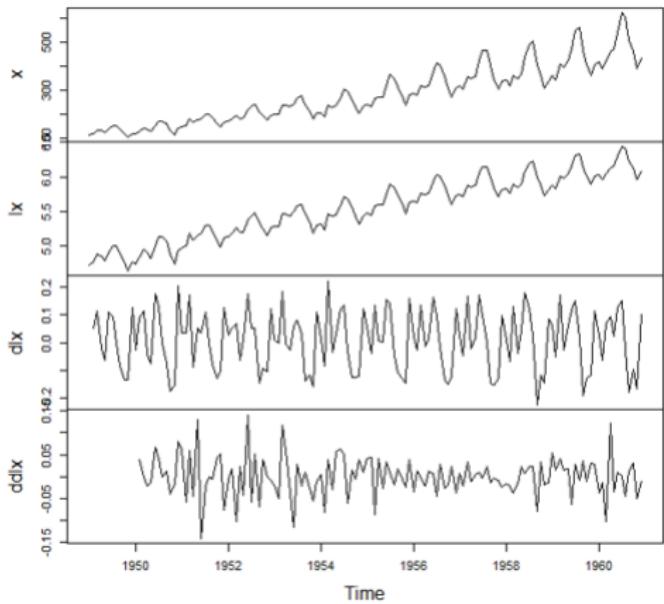
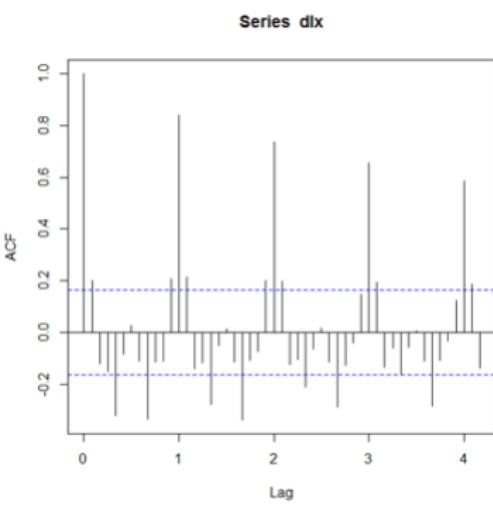
Overfitting

- Example: Recruitment series
 - Fit ARIMA(1,0,3) and ARIMA(5,0,5)



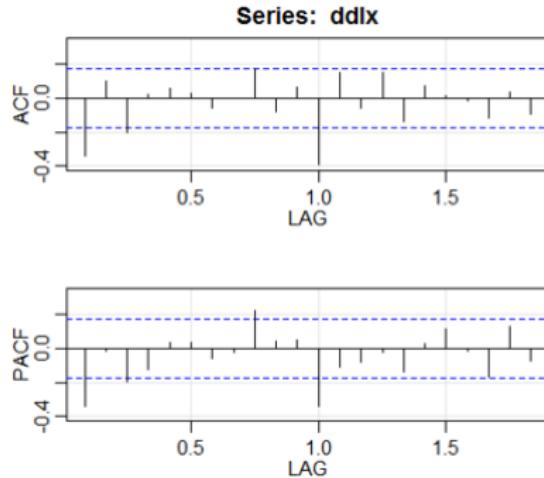
SARIMA - Air passangers

- Example: Air passangers



SARIMA - Air passangers

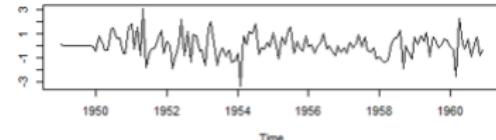
- Example: Air passangers



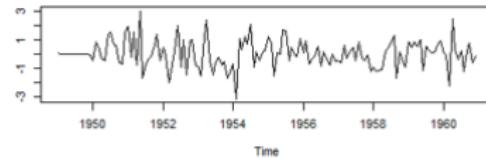
ARIMA(0, 1, 1)₁₂ or
ARIMA(1, 1, 0)₁₂

SARIMA - Air passengers

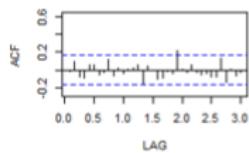
Model: (1,1,1) (0,1,1) Standardized Residuals



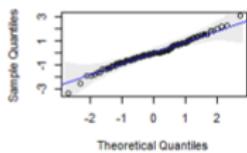
Model: (1,1,1) (1,1,0) Standardized Residuals



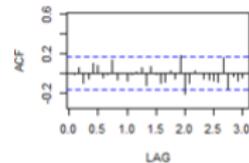
ACF of Residuals



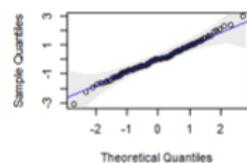
Normal Q-Q Plot of Std Residuals



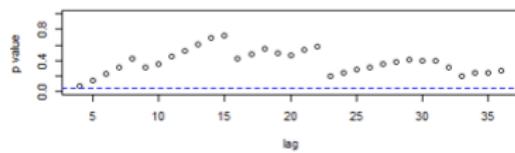
ACF of Residuals



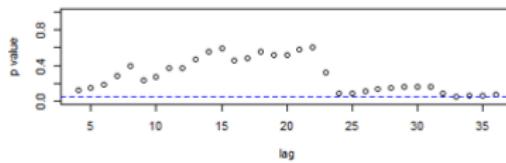
Normal Q-Q Plot of Std Residuals



p values for Ljung-Box statistic

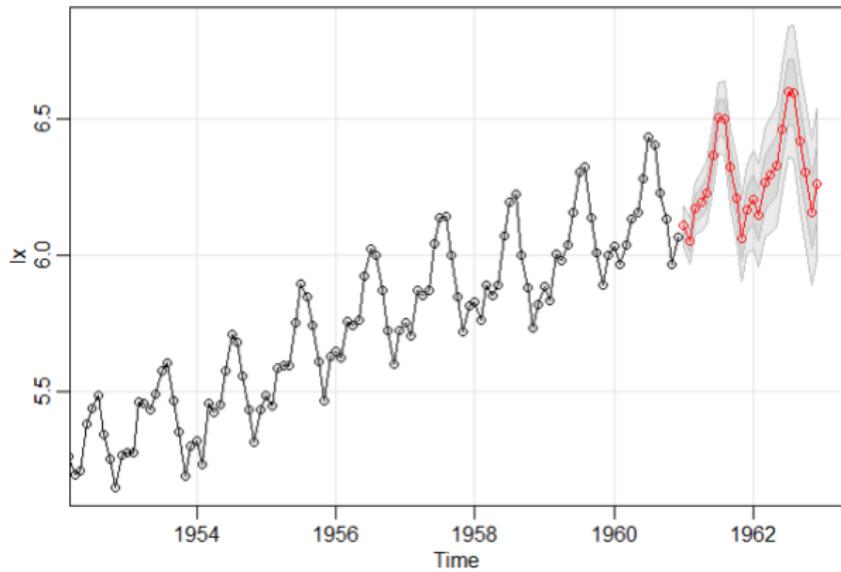


p values for Ljung-Box statistic



SARIMA

- Forecasting



Read home

- Shumway and Stoffer, Chapter 1, 2 and 3

ARIMA models

Time series models so far

$$\phi^P(B)x_t = \theta^q(B)w_t$$

Model	Concise form
AR(p)	$\phi^P(B)x_t = w_t$
MA(q)	$x_t = \theta^q(B)w_t$
ARMA(p, q)	$\phi^P(B)x_t = \theta^q(B)w_t$
ARIMA(p, d, q)	$\phi^P(B)(1 - B)^d x_t = \theta^q(B)w_t$
ARMA($P, Q)_s$	$\Phi^P(B^s)x_t = \Theta^Q(s)w_t$
ARIMA($P, D, Q)_s$	$\Phi^P(B^s)(1 - B^s)^D x_t = \Theta^Q(B^s)w_t$
ARMA($p, q) \times (P, Q)_s$	$\Phi^P(B^s)\phi^P(B)x_t = \Theta^Q(B^s)\theta^q(B)w_t$
ARIMA($p, d, q) \times (P, D, Q)_s$	$\Phi^P(B^s)\phi^P(B)(1 - B^s)^D(1 - B)^d x_t = \Theta^Q(B^s)\theta^q(B)w_t$

* The notation used in this slide deviates from the notation used in the course literature so far.

Consider an AR(2) model

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

Let $\mathbf{z}_t = \begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix}$ and $e_t = \begin{bmatrix} w_t \\ 0 \end{bmatrix}$.

Show that we rewrite the AR(2) model in the state space form:

$$\begin{aligned}\mathbf{z}_t &= \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \mathbf{z}_{t-1} + e_t \\ x_t &= [1 \ 0] \mathbf{z}_t,\end{aligned}$$

$$\phi^P(B)x_t = \theta^q(B)w_t$$

Can we rewrite any model of this form as a state space model?

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + e_t,$$

$$\mathbf{x}_t = C\mathbf{z}_t + \nu_t,$$

$$\phi^p(B)x_t = \theta^q(B)w_t$$

Outline of the solution:

Let $r = \max(p, q + 1)$,

$$\phi^r(B) = 1 - \phi_1 B - \cdots - \phi_r B^r,$$

$$\theta^r(B) = 1 + \theta_1 B + \cdots + \theta_{r-1} B^{r-1},$$

$\phi^r(B)(\theta^r(B))^{-1}x_t = w_t$. Hence, for $z_t = (\theta^r(B))^{-1}x_t$ we can have

$$\phi^r(B)z_t = w_t$$

$$z_t = \begin{bmatrix} z_t \\ z_{t-1} \\ z_{t-2} \\ \vdots \\ z_{t-r+1} \end{bmatrix} \text{ and } z_t = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_r \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} z_{t-1} + \begin{bmatrix} w_t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

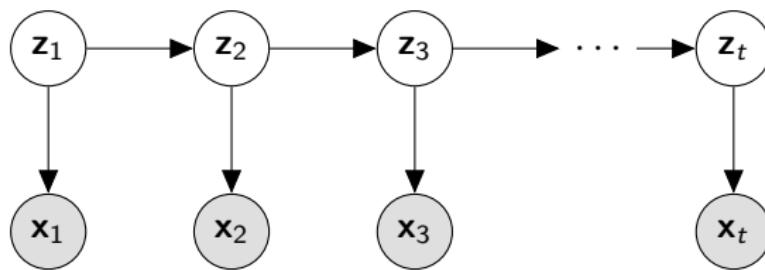
$$x_t = [1 \ \theta_1 \ \theta_2 \ \cdots \ \theta_r] z_t$$

State Space models - graphical models

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + e_t, \quad e_t \sim f_e(\cdot)$$

$$\mathbf{x}_t = C\mathbf{z}_t + \nu_t, \quad \nu_t \sim f_\nu(\cdot)$$

A probabilistic graphical model for stochastic dynamical system with latent state \mathbf{z}_k and observations \mathbf{x}_k

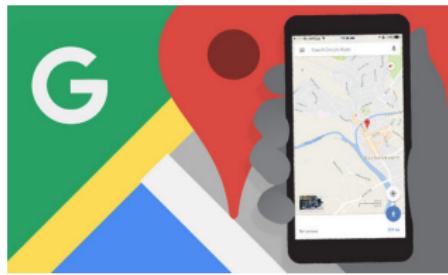
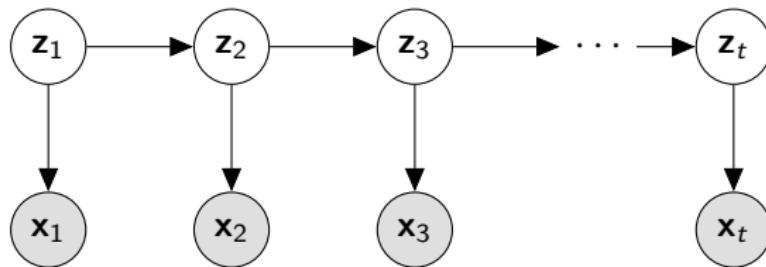


The main tool here is the probability Calculus; Bayes rule and marginalization.

Dynamical systems - more general case

$$\mathbf{z}_t = \mathcal{F}(\mathbf{z}_{t-1}) + e_t, \quad e_t \sim f_e(\cdot)$$

$$\mathbf{x}_t = \mathcal{C}(\mathbf{z}_t) + \nu_t, \quad \nu_t \sim f_\nu(\cdot)$$

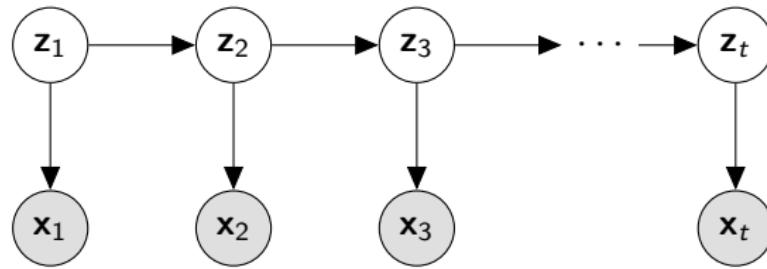


State Space models - Linear and Gaussian

Our main focus will be on linear and Gaussian models:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + e_t, \quad e_t \sim N(0, Q)$$

$$\mathbf{x}_t = C\mathbf{z}_t + \nu_t, \quad \nu_t \sim N(0, R)$$



Bayesian Inference

Bayesian inference is a means of combining prior beliefs with the data (evidence) to obtain posterior beliefs.

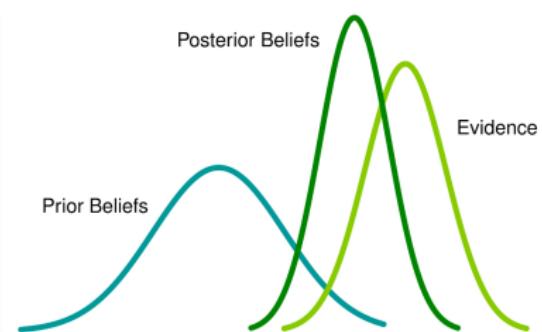
Example: likelihood update

$$f(z|x) \propto f(x|z)f(z)$$

Probability Calculus

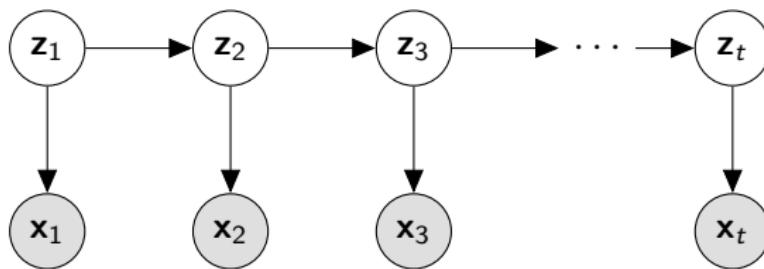
$$f(z, x) = f(z|x)f(x)$$

$$f(z, x) = f(x|z)f(z)$$



Online recursive algorithms

Consider a stochastic dynamical system represented by the following recursion



$$z_1 \sim f(z_1), \quad (1a)$$

$$x_k \sim f(x_k | z_k), \quad (1b)$$

$$z_{k+1} \sim f(z_{k+1} | z_k). \quad (1c)$$

The Bayesian filtering recursion corresponds to computing the posterior distributions $f(z_k | x_{1:k})$;

$$f(z_k | x_{1:k}) = \frac{f(z_k | x_{1:k-1}) f(x_k | z_k)}{\int f(z_k | x_{1:k-1}) f(x_k | z_k) dz_k}. \quad (2)$$

The density $f(z_k | x_{1:k-1})$ in the numerator of (2) which is called the predicted density of z_k and is obtained by integration as in

$$f(z_k | x_{1:k-1}) = \int f(z_k | z_{k-1}) f(z_{k-1} | x_{1:k-1}) dz_{k-1}. \quad (3)$$

Properties of the Normal density function

Property 1: $f(\mathbf{z})f(\mathbf{x}|\mathbf{z}) = f(\mathbf{z}, \mathbf{x})$

$$N(\mathbf{z}; \mu, \Sigma)N(\mathbf{x}; C\mathbf{z}, R) = N\left(\begin{bmatrix}\mathbf{z} \\ \mathbf{x}\end{bmatrix}; \begin{bmatrix}\mu \\ C\mu\end{bmatrix}, \begin{bmatrix}\Sigma & \Sigma C^T \\ C\Sigma & C\Sigma C^T + R\end{bmatrix}\right)$$

Property 2: marginalization and conditioning

If x, y were jointly normal:

$$f(x, y) = N\left(\begin{bmatrix}x \\ y\end{bmatrix}; \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

then

$$f(x) = N(x; \mu_1, \Sigma_{11})$$

$$f(y) = N(y; \mu_2, \Sigma_{22})$$

$$f(x|y) = N(x; \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(y - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$$

$$f(y|x) = N(y; \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})$$

The Kalman Filter's Foundation

Let \mathbf{z} have a normal prior distribution with mean μ and covariance Σ , i.e., $\mathbf{z} \sim N(\mathbf{z}; \mu, \Sigma)$.

An observation \mathbf{x} with the likelihood function $f(\mathbf{x}|\mathbf{z}) = N(\mathbf{x}; C\mathbf{z}, R)$ is in hand where C is a matrix with proper dimensions and R is a covariance matrix. The posterior distribution of \mathbf{z} can be obtained using the Bayes' rule

$$f(\mathbf{z}|\mathbf{x}) = \frac{f(\mathbf{z})f(\mathbf{x}|\mathbf{z})}{\int f(\mathbf{z})f(\mathbf{x}|\mathbf{z}) d\mathbf{z}} \quad (4)$$

$$= \frac{N(\mathbf{z}; \mu, \Sigma)N(\mathbf{x}; C\mathbf{z}, R)}{\int N(\mathbf{z}; \mu, \Sigma)N(\mathbf{x}; C\mathbf{z}, R) d\mathbf{z}}. \quad (5)$$

The posterior distribution $f(\mathbf{z}|\mathbf{x})$ has an analytical solution and turns out to be the normal distribution $N(\mathbf{z}; \mu', \Sigma')$ where

$$\mu' = \mu + K(\mathbf{x} - C\mu), \quad (6a)$$

$$\Sigma' = \Sigma - KC\Sigma, \quad (6b)$$

where

$$K = \Sigma C^T (C\Sigma C^T + R)^{-1}. \quad (7)$$

Time Series Analysis

Lecture 7: State Space Model - Estimation

Tohid Ardestiri

Linköping University
Division of Statistics and Machine Learning

October 2, 2019



Kalman filter

Kalman filter is an algorithm that uses time series data, **containing statistical noise and unknown innovations**, and produces estimates of latent (hidden) process that tend to be more accurate than those based on a single observations using a probabilistic framework.

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \mathbf{e}_t,$$

$$\mathbf{x}_t = C\mathbf{z}_t + \nu_t,$$

Kalman filtering output is

$$f(\mathbf{z}_t | \mathbf{x}_{1:t}).$$

That is, it computes the the posterior density of \mathbf{z}_t using the observations up to time t .

Kalman filtering recursion

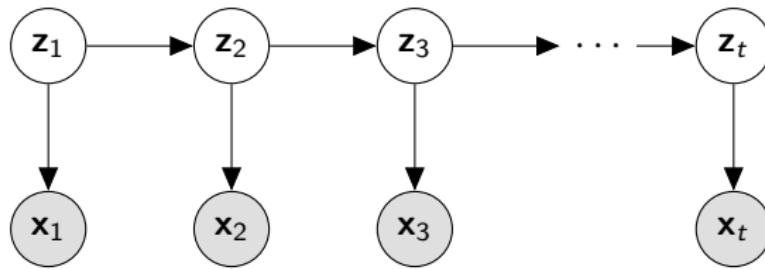
- ① initial estimate at $t = 1 \rightarrow N(\mathbf{z}_1; m_0, P_0)$
- ② observation update using \mathbf{x}_t and $\mathbf{x}_t = C\mathbf{z}_t + \nu_t \rightarrow N(\mathbf{z}_t; m_{t|t}, P_{t|t})$
- ③ prediction using $\mathbf{z}_{t+1} = A\mathbf{z}_t + e_{t+1} \rightarrow N(\mathbf{z}_t; m_{t+1|t}, P_{t+1|t})$
- ④ $t \leftarrow t + 1$
- ⑤ go to 2

State Space models - Time varying

State space models can be time-varying

$$\mathbf{z}_t = A_t \mathbf{z}_{t-1} + e_t, \quad e_t \sim N(0, Q_t)$$

$$\mathbf{x}_t = C_t \mathbf{z}_t + \nu_t, \quad \nu_t \sim N(0, R_t)$$



State space models with known deterministic input

State space model with
input \mathbf{u} .

$$\begin{aligned}\mathbf{z}_t &= A\mathbf{z}_{t-1} + B\mathbf{u}_{t-1} + \mathbf{e}_t, \\ \mathbf{x}_t &= C\mathbf{z}_t + \nu_t,\end{aligned}$$

Initialization:

$$f(\mathbf{z}_1) = N(\mathbf{z}_1; m_{1|0}, P_{1|0})$$

```
1: Inputs:  $A, B, C, Q, R, \mathbf{u}_{1:T},$ 
    $\mathbf{x}_{1:T}, m_{1|0}, P_{1|0}$ 
2: for  $t = 1$  to  $T$  do
   Kalman filter observation update step
3:    $K_t \leftarrow P_{t|t-1} C^T (C P_{t|t-1} C^T + R)^{-1}$ 
4:    $m_{t|t} \leftarrow m_{t|t-1} + K_t (\mathbf{x}_t - C m_{t|t-1})$ 
5:    $P_{t|t} \leftarrow P_{t|t-1} - K_t C P_{t|t-1}$ 
   Kalman filter prediction step
6:    $m_{t+1|t} \leftarrow A m_{t|t} + B \mathbf{u}_t$ 
7:    $P_{t+1|t} \leftarrow A P_{t|t} A^T + Q$ 
8: end for
9: Outputs:  $m_{t|t}$  and  $P_{t|t}$  for  $t = 1 : T$ 
```

Kalman Smoothing

The purpose of Kalman smoothing is to compute the marginal posterior distribution of \mathbf{z}_t at time t after receiving observations up to time T where $T > t$:

$$f(\mathbf{z}_t | \mathbf{x}_{1:T}) = N(\mathbf{z}_t; m_{t|T}, P_{t|T})$$

The RTS smoother uses a Kalman filter in its forward path. In its backwards path it updates the densities using the relation

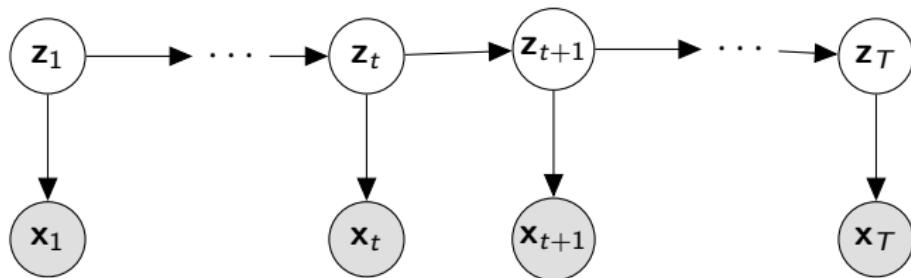
$$\mathbf{z}_t = A_{t-1} \mathbf{z}_{t-1} + \mathbf{e}_t$$

RTS Smoother's derivation

Assume $f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T})$ is available as in

$$f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) = N(\mathbf{z}_{t+1}; \mathbf{m}_{t+1|T}, \mathbf{P}_{t+1|T})$$

For example $f(\mathbf{z}_T | \mathbf{x}_{1:T})$ which is the filtering density of \mathbf{z}_T is available after filtering.



The objective is to compute $f(\mathbf{z}_t, \mathbf{z}_{t+1} | \mathbf{x}_{1:T})$.

RTS Smoother's derivation

The joint posterior $f(\mathbf{z}_t, \mathbf{z}_{t+1} | \mathbf{x}_{1:t})$ can be written as

$$\begin{aligned} f(\mathbf{z}_t, \mathbf{z}_{t+1} | \mathbf{x}_{1:t}) &= N(\mathbf{z}_t; m_{t|t}, P_{t|t}) N(\mathbf{z}_{t+1}; A\mathbf{z}_t, Q) \\ &= N\left(\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t+1} \end{bmatrix}, \begin{bmatrix} m_{t|t} \\ Am_{t|t} \end{bmatrix}, \begin{bmatrix} P_{t|t} & P_{t|t}A^T \\ AP_{t|t} & AP_{t|t}A^T + Q \end{bmatrix}\right) \end{aligned}$$

Using the conditioning property of the multivariate normal distribution $f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t})$ can be computed as a normal density as given in the following:

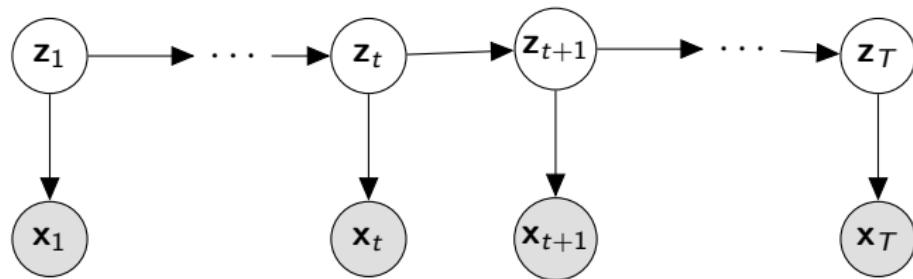
$$f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t}) = N(\mathbf{z}_t; \tilde{m}_t, \tilde{P}_t)$$

where \tilde{m}_t is a function of \mathbf{z}_{t+1} .

RTS Smoother's derivation

Note the Markov property

$$f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:T}) = f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t})$$



Assume $f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T})$ is available as in

$$f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) = N(\mathbf{z}_{t+1}; m_{t+1|T}, P_{t+1|T})$$

Recall that

$$\begin{aligned} f(\mathbf{z}_{t+1}, \mathbf{z}_t | \mathbf{x}_{1:T}) &= f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:T}) \\ &= f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t}) \\ &= N(\mathbf{z}_{t+1}; m_{t+1|T}, P_{t+1|T}) N(\mathbf{z}_t; \tilde{m}_t, \tilde{P}_t) \end{aligned}$$

RTS Smoother's derivation **Whiteboard**

where

$$G_t = P_{t|t} A_t^T (A P_{t|t} A^T + Q)^{-1} = P_{t|t} A_t^T P_{t+1|t}^{-1}$$

$$\tilde{m}_t = m_{t|t} + G_t (\mathbf{z}_{t+1} - A m_{t|t})$$

$$\tilde{P}_t = P_{t|t} - G_t (A P_{t|t} A^T + Q) G_t^T = P_{t|t} - G_t P_{t+1|t} G_t^T$$

Hence,

$$f(\mathbf{z}_{t+1}, \mathbf{z}_t | \mathbf{x}_{1:T}) = N(\mathbf{z}_{t+1}; m_{t+1|T}, P_{t+1|T}) N(\mathbf{z}_t; \tilde{m}_t, \tilde{P}_t)$$

$$= N \left(\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t+1} \end{bmatrix}, \begin{bmatrix} m_{t|t} + G_t (m_{t+1|T} - A m_{t|t}) \\ m_{t+1|T} \end{bmatrix}, \begin{bmatrix} G_t P_{t+1|T} G_t^T + \tilde{P}_t & G_t P_{t+1|T} \\ P_{t+1|T} G_t^T & P_{t+1|T} \end{bmatrix} \right)$$

RTS Smoother's derivation **Whiteboard**

The smoothing density's parameters is given by

$$G_t = P_{t|t} A_t^T (A P_{t|t} A^T + Q)^{-1} = P_{t|t} A_t^T P_{t+1|t}^{-1}$$

$$m_{t|T} = m_{t|t} + G_t(m_{t+1|T} - A m_{t|t})$$

$$\begin{aligned} P_{t|T} &= \tilde{P}_t + G_t P_{t+1|T} G_t^T = P_{t|t} - G_t P_{t+1|t} G_t^T + G_t P_{t+1|T} G_t^T \\ &= P_{t|t} + G_t(P_{t+1|T} - P_{t+1|t}) G_t^T \end{aligned}$$

RTS smoother's backwards recursion

Prove the backwards recursion of the RTS smoother for following state space model with initial prior on the state $f(\mathbf{z}_1) = N(\mathbf{z}_1; \mathbf{m}_0, \mathbf{P}_0)$

$$\mathbf{z}_t = A_{t-1}\mathbf{z}_{t-1} + e_t, \quad e_t \sim N(0, Q_t)$$

$$\mathbf{x}_t = C_t\mathbf{z}_t + \nu_t, \quad \nu_t \sim N(0, R_t)$$

-
- 1: **Inputs:** $A_t, Q_t, m_{t|t}, P_{t|t}, m_{t+1|t}, P_{t+1|t}$ for $1 \leq t \leq T$
initialization
 - 2: **for** $t = T-1$ down to 1 **do**
 - 3: $G_t \leftarrow P_{t|t}A_t^T P_{t+1|t}^{-1}$
 - 4: $m_{t|T} \leftarrow m_{t|t} + G_t(m_{t+1|T} - A_t m_{t|t})$
 - 5: $P_{t|T} \leftarrow P_{t|t} + G_t(P_{t+1|T} - P_{t+1|t})G_t^T$
 - 6: **end for**
 - 7: **Outputs:** $m_{t|T}, P_{t|T}$
-

State Space models - Estimation

We consider three approaches.

① (Variational Bayes)

T. Ardestiri, E. Özkan, U. Orguner and F. Gustafsson, "Approximate Bayesian Smoothing with Unknown Process and Measurement Noise Covariances," in IEEE Signal Processing Letters, vol. 22, no. 12, pp. 2450-2454, Dec. 2015.

② Direct maximum likelihood estimate

③ Expectation maximization (EM)

Variational Bayes smoothing with unknown time varying R_t and Q_t

Consider a Linear and Gaussian state space model with parameters

$$A_k = \text{Diag}(a, a),$$

$$a = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix},$$

$$R_k^{\text{True}} = \left(2 - \cos\left(\frac{4\pi k}{K}\right) \right) R_0,$$

$$Q_k^{\text{True}} = \left(\frac{2}{3} + \frac{1}{3} \cos\left(\frac{4\pi k}{K}\right) \right) Q_0,$$

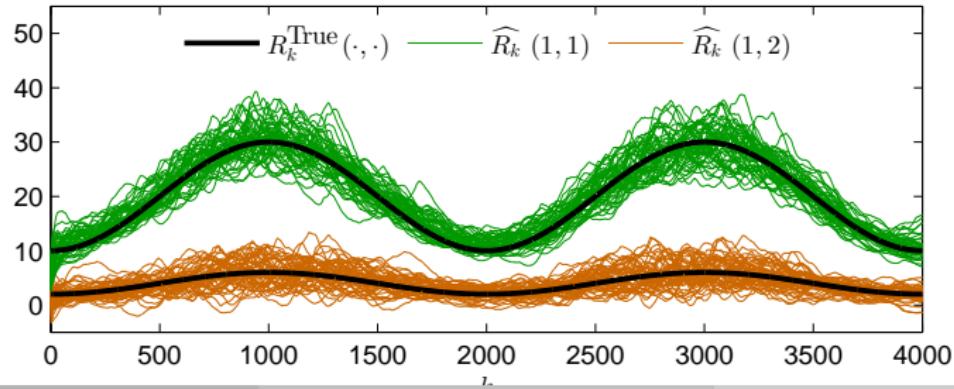
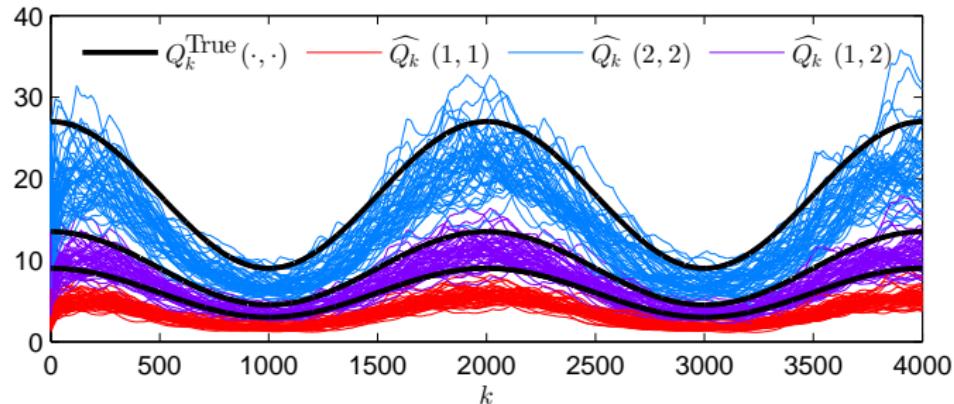
$$Q_0 = \text{Diag}(q, q),$$

$$q = \sigma_{\nu}^2 \begin{bmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{bmatrix},$$

$$R_0 = \sigma_e^2 \begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix},$$

$$C_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Variational Bayes smoothing with unknown time varying R_t and Q_t



Maximum likelihood methods

Whiteboard

Let $\theta = \{A, C, R, Q, m_0, P_0\}$ denote the unknown state space parameters

$$f(\mathbf{x}_{1:T}|\theta) = f(\mathbf{x}_1|\theta)f(\mathbf{x}_2|\mathbf{x}_1, \theta)f(\mathbf{x}_3|\mathbf{x}_{1:2}, \theta) \cdots f(\mathbf{x}_T|\mathbf{x}_{1:T-1}, \theta)$$

where

$$f(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \theta) = \int f(\mathbf{x}_{t+1}|\mathbf{z}_{t+1}, \mathbf{x}_{1:t}, \theta)f(\mathbf{z}_{t+1}|\mathbf{x}_{1:t}, \theta) d\mathbf{z}_{t+1}$$

This can be computed using the Kalman filter

$$\begin{aligned} f(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \theta) &= \int f(\mathbf{x}_{t+1}|\mathbf{z}_{t+1}, \mathbf{x}_{1:t}, \theta)f(\mathbf{z}_{t+1}|\mathbf{x}_{1:t}, \theta) d\mathbf{z}_{t+1} \\ &= \int N(\mathbf{x}_{t+1}; C\mathbf{z}_{t+1}, R)N(\mathbf{z}_{t+1}; m_{t+1|t}, P_{t+1|t}) d\mathbf{z}_{t+1} \\ &= N(\mathbf{x}_{t+1}; Cm_{t+1|t}, CP_{t+1|t}C^T + R) \end{aligned}$$

The negative logarithm of the likelihood becomes

$$\begin{aligned} I(\theta) &= - \sum_{t=1}^T \log f(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \theta) \\ &= - \sum_{t=1}^T \log N(\mathbf{z}_{t+1}; Cm_{t+1|t}, CP_{t+1|t}C^T + R) \\ &= \frac{1}{2} \sum_{t=1}^T \log |CP_{t+1|t}C^T + R| \\ &\quad + \frac{1}{2} \sum_{t=1}^T (\mathbf{x}_t - Cm_{t+1|t})(CP_{t+1|t}C^T + R)^{-1}(\mathbf{x}_t - Cm_{t+1|t})^T \end{aligned}$$

which can be solved using for example Newton-Raphson method.

Maximum likelihood methods

The first two derivatives of the negative log-likelihood is computed with respect to the θ .

Then in the iterations of the Newton-Raphson method

- ① An initial value for θ is selected, say $\theta^{(0)}$.
- ② A Kalman filter is run to compute the quantities for the first two derivatives of $I(\theta)$.
- ③ A new set of parameters are obtained from a Newton-Raphson procedure.
- ④ Iterations are repeated until convergence.

Expectation Maximization

Whiteboard

- Expectation-maximization (EM) method can be used to compute the maximum likelihood (ML) estimate of the state space parameters.
- In the E (Expectation) step of the EM algorithm the conditional expectation of the joint log-likelihood is computed using the last estimates of the unknown parameters as in

$$\mathcal{Q} = E \left[\log f(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) \mid \mathbf{x}_{1:T}, \theta^{(i)} \right] \quad (1)$$

where

$$\begin{aligned} \log f(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) &= \log N(\mathbf{z}_1; m_0, P_0) - \frac{T+1}{2} \log |R| \\ &\quad - \frac{1}{2} \sum_{t=1}^T \text{Tr} (R^{-1}(\mathbf{x}_t - C\mathbf{z}_t)(\mathbf{x}_t - C\mathbf{z}_t)^T) - \frac{T}{2} \log |Q| \\ &\quad - \frac{1}{2} \sum_{t=1}^{T-1} \text{Tr} (Q^{-1}(\mathbf{z}_{t+1} - A\mathbf{z}_t)(\mathbf{z}_{t+1} - A\mathbf{z}_t)^T) + c. \end{aligned} \quad (2)$$

Therefore,

$$\begin{aligned} \mathcal{Q} = & -\frac{1}{2} E[(\mathbf{z}_0 - m_0) P_0^{-1} (\mathbf{z}_0 - m_0)^T + \log |P_0|] \\ & - \frac{T+1}{2} \log |R| - \frac{1}{2} \text{Tr} \left(R^{-1} \sum_{t=0}^T E[(\mathbf{x}_t - C\mathbf{z}_t)(\mathbf{x}_t - C\mathbf{z}_t)^T | \mathbf{x}_{1:T}] \right) \\ & - \frac{T}{2} \log |Q| - \frac{1}{2} \text{Tr} \left(Q^{-1} \sum_{t=0}^{T-1} E[(\mathbf{z}_{t+1} - A\mathbf{z}_t)(\mathbf{z}_{t+1} - A\mathbf{z}_t)^T | \mathbf{x}_{1:T}] \right) + c, \end{aligned} \tag{3}$$

In order to compute the expectations the RTS smoother's posterior $f(\mathbf{z}_t | \mathbf{z}_{1:T})$ is used.

Then in the iterations of the EM method

- ① An initial value for θ is selected, say $\theta^{(0)}$.
- ② A Kalman smoother is run using $\theta^{(i)}$
- ③ In the expectation step Q function as a function of θ is derived.
- ④ A new set of parameters $\theta^{(i+1)}$ are obtained from maximization of the Q function.
- ⑤ Iterations are repeated until convergence.

Read home

- Shumway and Stoffer, Chapter 6.3

Time Series Analysis

Lecture 8: State Space Model

Stochastic Volatility

Tohid Ardestiri

Linköping University
Division of Statistics and Machine Learning

October 4, 2019



Remaining Course topics

- ARIMA models
- State space models (2 lectures, 1 teaching session with hand-in, 1 computer lab with short report)
 - ▶ Linear and Gaussian state space models (Chapter 6.1)
 - ▶ Kalman filtering, Kalman smoothing and Forecasting (Chapter 6.2)
 - ▶ Maximum likelihood estimate of the state space models (Chapter 6.3)
 - ▶ Stochastic volatility (Chapter 6.11)
- Recurrent Neural Networks (RNNs) (1 lecture and 1 Computer lab No examination)
- Summary lecture

Why Stochastic volatility

$$\begin{aligned}\mathbf{z}_t &= A\mathbf{z}_{t-1} + \mathbf{e}_t, & \mathbf{e}_t &\sim N(0, Q) \\ \mathbf{x}_t &= C\mathbf{z}_t + \nu_t, & \nu_t &\sim N(0, R)\end{aligned}$$

- **Filtering:** Kalman filtering, $f(\mathbf{z}_t | \mathbf{x}_{1:t})$
- **Smoothing:** Kalman smoothing, $f(\mathbf{z}_t | \mathbf{x}_{1:T})$
- **Modelling:** Maximum likelihood and EM, $\hat{\theta} = \arg \max_{\theta} f(\mathbf{x}_{1:T} | \theta)$
- Case study on **Stochastic volatility** via a generalization of the above tools

Stochastic Volatility

In finance, **return** is a profit on an investment. It comprises any change in value of the investment, and/or cash flows which the investor receives from the investment, such as interest payments or dividends.

Stochastic volatility models are those in which the variance of a stochastic process is itself randomly distributed.

In the following:

- r_t denote the **return** of some financial asset. A common model for the return is

$$r_t = \beta \sigma_t \epsilon_t$$

- σ_t is the **volatility process** and
- ϵ_t is an **iid sequence** and $\epsilon_t \sim iid(0, 1)$ and ϵ_t is independent of past σ_s ($s \leq t$)

Stochastic Volatility

In the following:

- r_t denote the **return** of some financial asset. A common model for the return is

$$r_t = \beta \sigma_t \epsilon_t$$

- σ_t is the **volatility process** and
- ϵ_t is an **iid sequence** and $\epsilon_t \sim iid(0, 1)$ and ϵ_t is independent of past σ_s ($s \leq t$)
- Let $z_t = \log \sigma_t^2$ and consider the hidden autoregressive model

$$z_t = \phi z_{t-1} + w_t$$

$$r_t = \beta \exp(z_t/2) \epsilon_t$$

In this model $w_t \sim iidN(0, \sigma_w^2)$ and ϵ_t is iid noise with finite moments.

$$\mathbf{z}_t = \phi \mathbf{z}_{t-1} + w_t$$

$$r_t = \beta \exp(\mathbf{z}_t/2) \epsilon_t$$

Furthermore, let $\mathbf{x}_t = \log r_t^2$ and $\nu_t = \log \epsilon_t^2$. We obtain

$$\mathbf{x}_t = \alpha + \mathbf{z}_t + \nu_t$$

We can move the α to the state equation and rewrite it as

$$\mathbf{z}_t = \phi_0 + \phi_1 \mathbf{z}_{t-1} + w_t$$

$$\mathbf{x}_t = \mathbf{z}_t + \nu_t$$

where the ϕ_0 is called the leverage effect.

Stochastic Volatility

The distribution of ν_t is not Gaussian because

$$\begin{aligned}\nu_t &= \log \epsilon_t^2 \text{ and} \\ \epsilon_t &\sim iidN(0, 1)\end{aligned}$$

Hence, ν is distributed as a log of a chi-squared distribution with degree of freedom 1 with density

$$f(\nu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(e^\nu - \nu)\right\} \quad -\infty < \nu < \infty$$

Stochastic Volatility - Gaussian mixture approximation

Instead let us approximate $f(\nu)$ by a Gaussian mixture

$$f(\eta) = \pi_0 N(\eta; 0, \sigma_0^2) + \pi_1 N(\eta; \mu_1, \sigma_1^2)$$

That is,

$$\eta_t = I_t n_{t0} + (1 - I_t) n_{t1}$$

where I_t is an iid Bernoulli process where $Pr\{I = 0\} = \pi_0$ and $Pr\{I = 1\} = \pi_1$, $\pi_0 + \pi_1 = 1$. Also,

$$n_{t0} \sim N(0, \sigma_0^2)$$

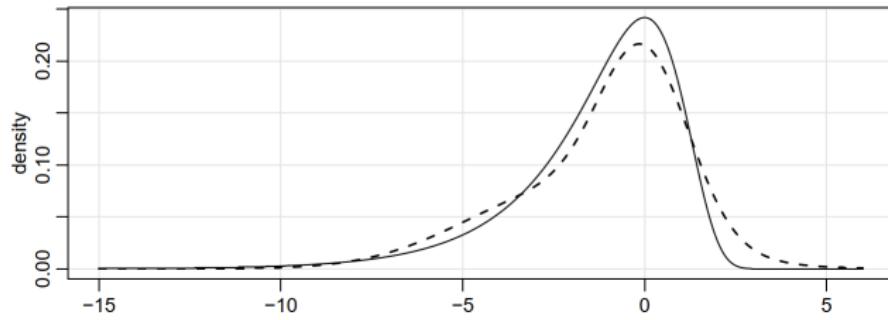
$$n_{t1} \sim N(\mu_1, \sigma_1^2)$$

Stochastic Volatility - Gaussian sum approximation

$$f(\eta) = \pi_0 N(\eta; 0, \sigma_0^2) + \pi_1 N(\eta; \mu_1, \sigma_1^2) \quad -\infty < \eta < \infty$$

$$f(\nu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(e^\nu - \nu)\right\} \quad -\infty < \nu < \infty$$

$f(\nu)$ and $f(\eta)$ are plotted for comparison. The dashed line is the Gaussian sum approximation, $f(\eta)$.



Stochastic Volatility - Gaussian sum formulation

The problem is finding the filtering distribution of $\mathbf{z}_t | \mathbf{x}_{1:t}$ when

$$\mathbf{z}_t = \phi_0 + \phi_1 \mathbf{z}_{t-1} + w_t$$

$$\mathbf{x}_t = \mathbf{z}_t + \eta_t$$

and

$$w_t \sim iidN(0, \sigma_w^2)$$

$$\eta_t \sim \pi_0 N(0, \sigma_0^2) + \pi_1 N(\mu_1, \sigma_1^2)$$

where $\pi_0 + \pi_1 = 1$

The problem is finding the filtering distribution of $\mathbf{z}_t | \mathbf{x}_{1:t}$ when

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + w_t$$

$$\mathbf{x}_t = C\mathbf{z}_t + \eta_t$$

and

$$w_t \sim iidN(0, Q)$$

$$\eta_t \sim \pi_0 N(\mu_0, R_1) + \pi_1 N(\mu_1, R_2)$$

where $\pi_0 + \pi_1 = 1$

Read home

- Shumway and Stoffer, Chapter 6.11

Time Series Analysis

Lecture X: Summary Questions and Answers

Tohid Ardesthiri

Linköping University
Division of Statistics and Machine Learning

October 16, 2019



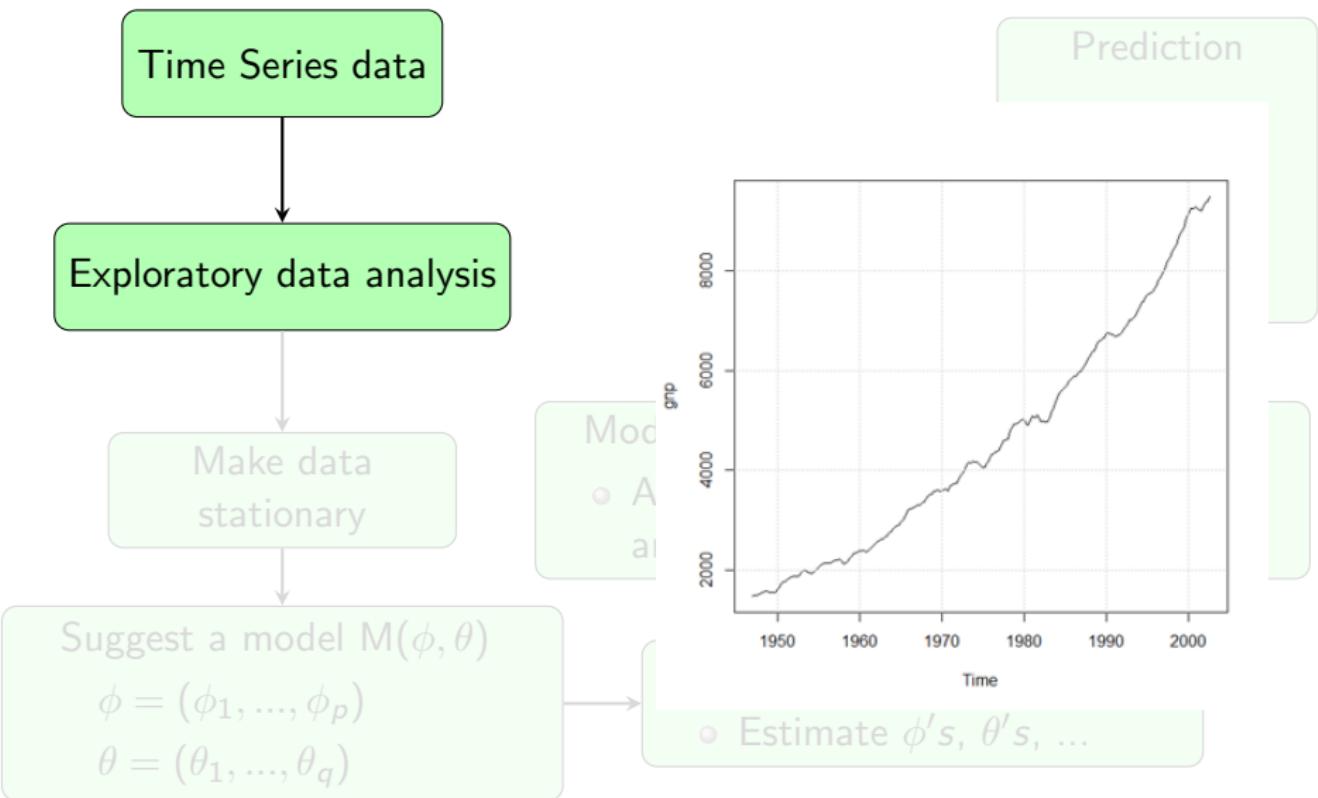
Course topics

- Time series, time series regression and exploratory analysis
 - ▶ Autocovariance, ACF
 - ▶ Sample ACF
 - ▶ Stationarity, detrending, differencing,
 - ▶ transformation and smoothing
- ARIMA models
 - ▶ AR, MA, ARMA, ARIMA, seasonal ARIMA
 - ▶ PACF
 - ▶ Model selection
 - ▶ Estimation
 - ▶ Forecasting
- State space models
 - ▶ Linear and Gaussian state space models
 - ▶ Kalman filtering, Kalman smoothing and Forecasting
 - ▶ Maximum likelihood estimate of the state space models
 - ▶ Stochastic volatility
- Recurrent Neural Networks (RNNs)

Stationarity

- Time series x_t is **weakly stationary (stationary)** if
 - ▶ $E x_t = \text{const}$
 - ▶ $\gamma(s, t) = \gamma(|s - t|)$
 - ▶ $\text{var}(x_t) < \infty$
- $\gamma(t, t + h) = \gamma(|t + h - t|) = \gamma(h)$
 - ▶ Autocovariance depends on lag only!
- Autocovariance for stationary process $\gamma(h) = \text{cov}(x_t, x_{t+h})$
- ACF for stationary process $\rho(h) = \frac{\gamma(h)}{\gamma(0)}$

Time domain: The Big Picture



Time domain: The Big Picture

Time Series data

$$Y_t = \nabla(\log(X_t))$$

Prediction

Exploratory data analysis

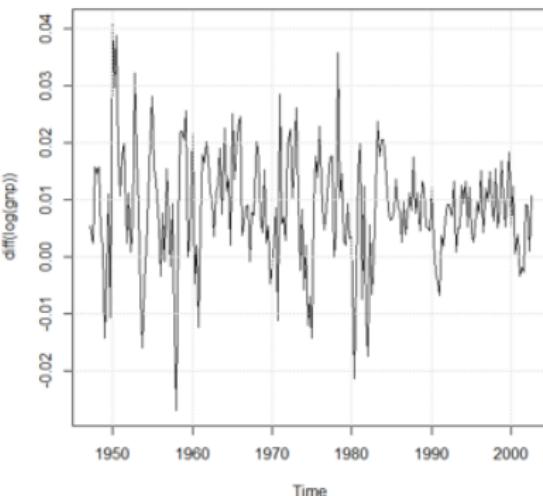
Make data stationary

Suggest a model $M(\phi, \theta)$

$$\phi = (\phi_1, \dots, \phi_p)$$

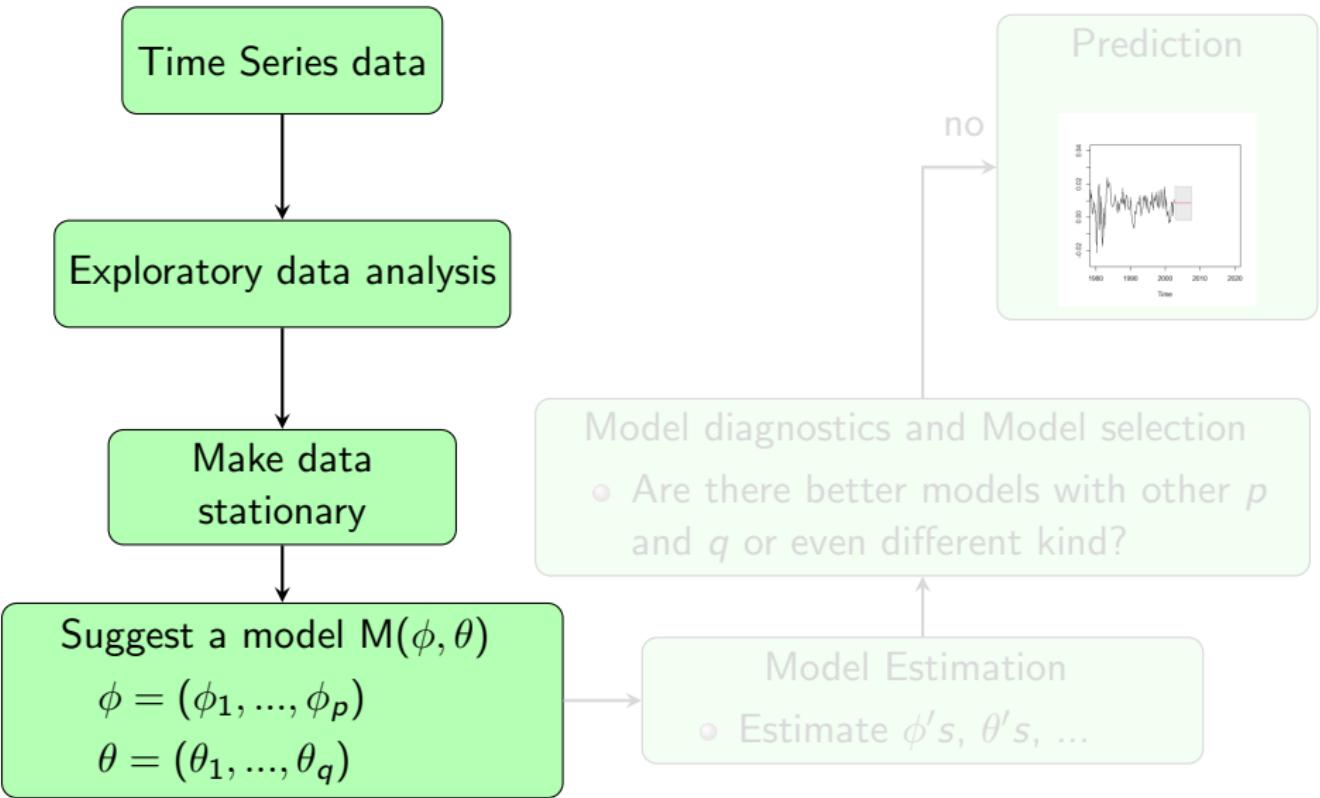
$$\theta = (\theta_1, \dots, \theta_q)$$

Model
A
ai

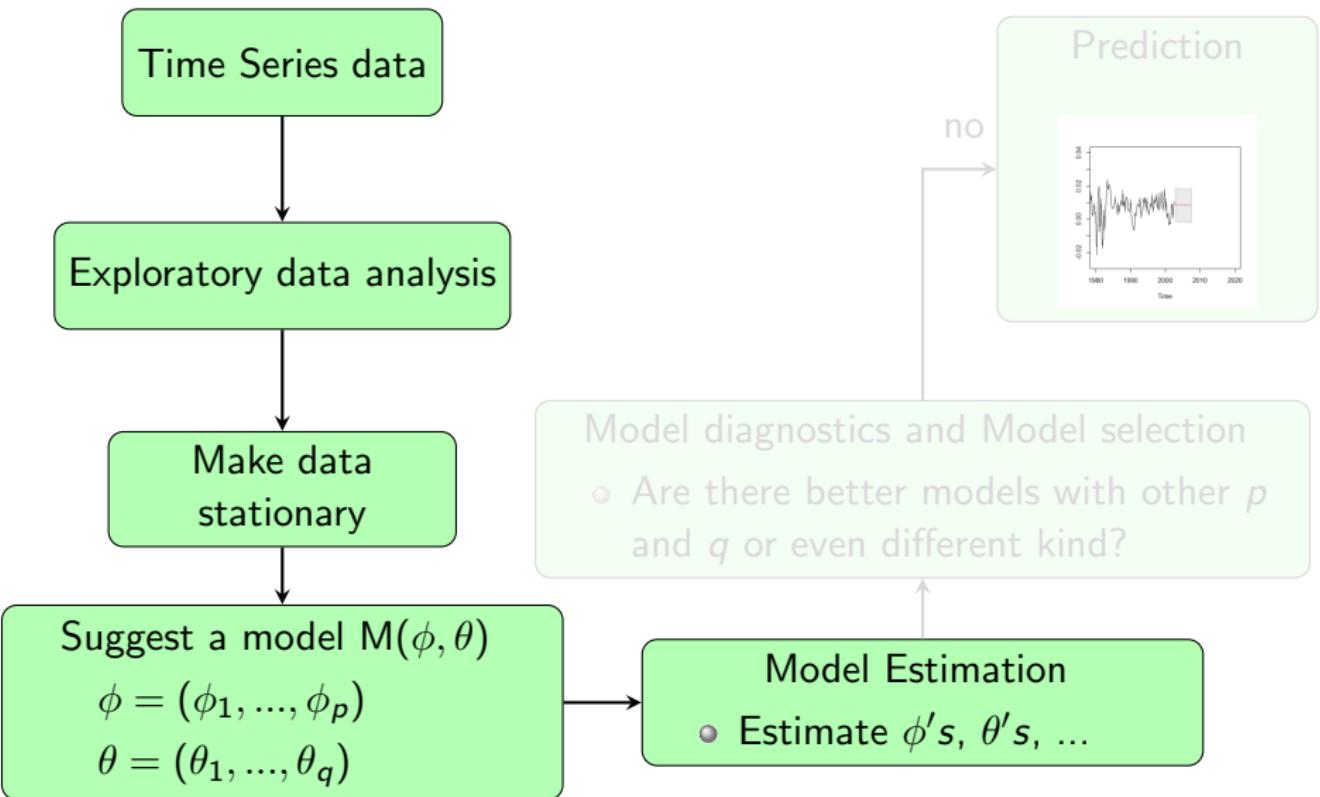


Estimate ϕ 's, θ 's, ...

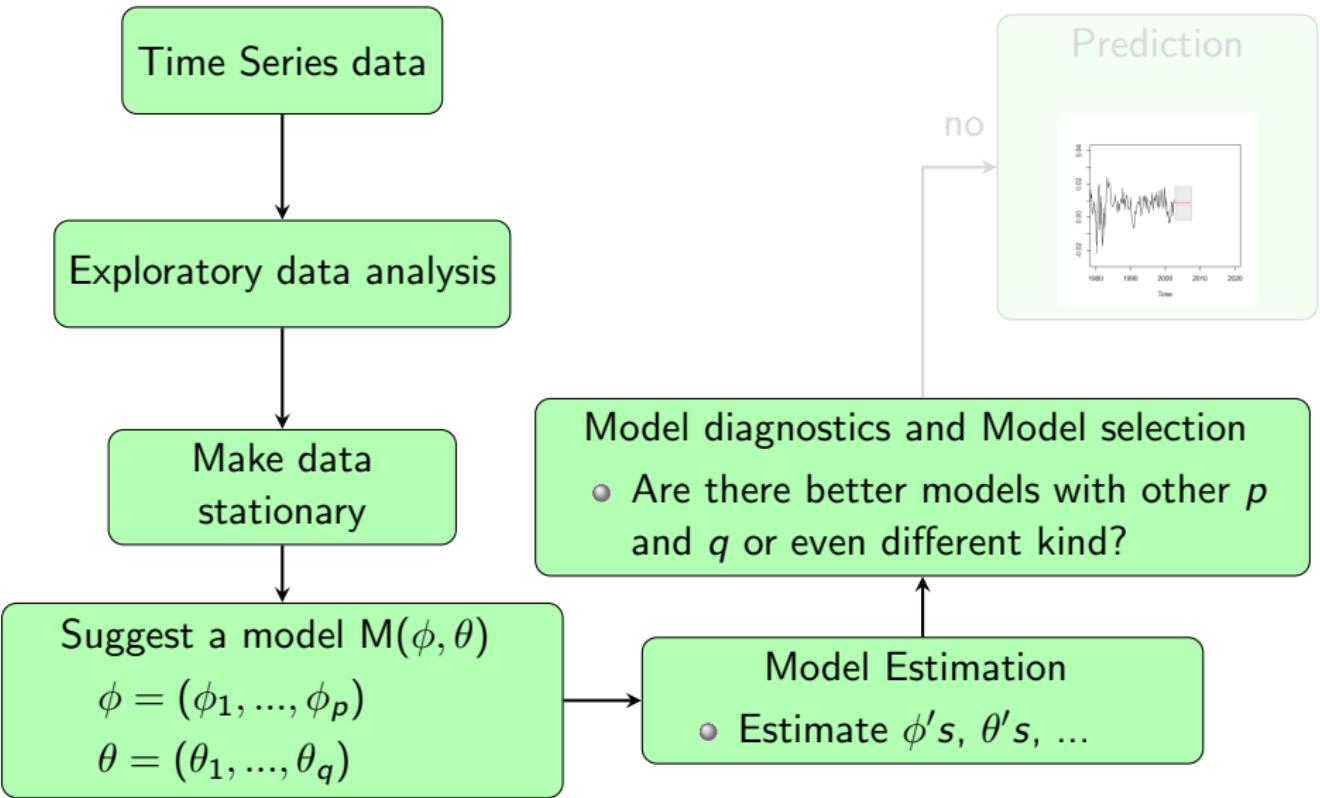
Time domain: The Big Picture



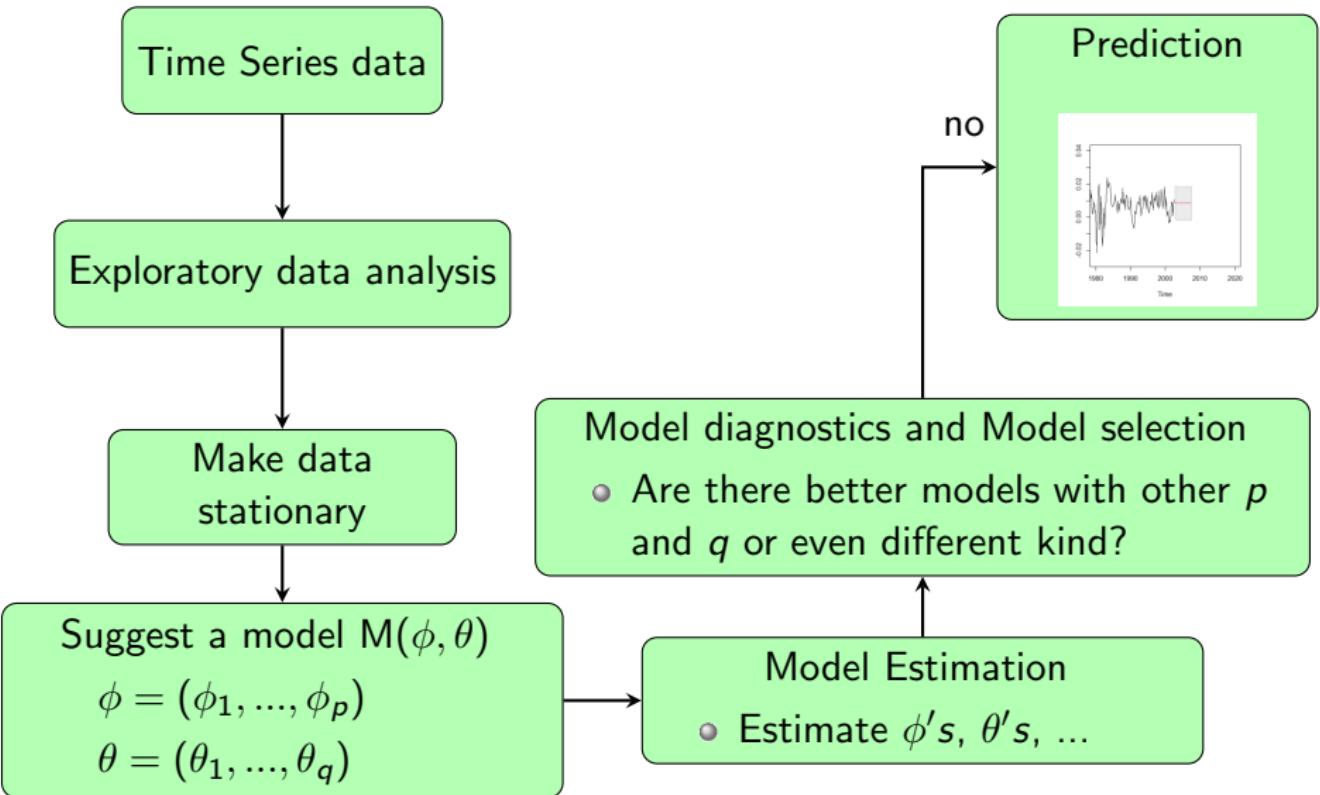
Time domain: The Big Picture



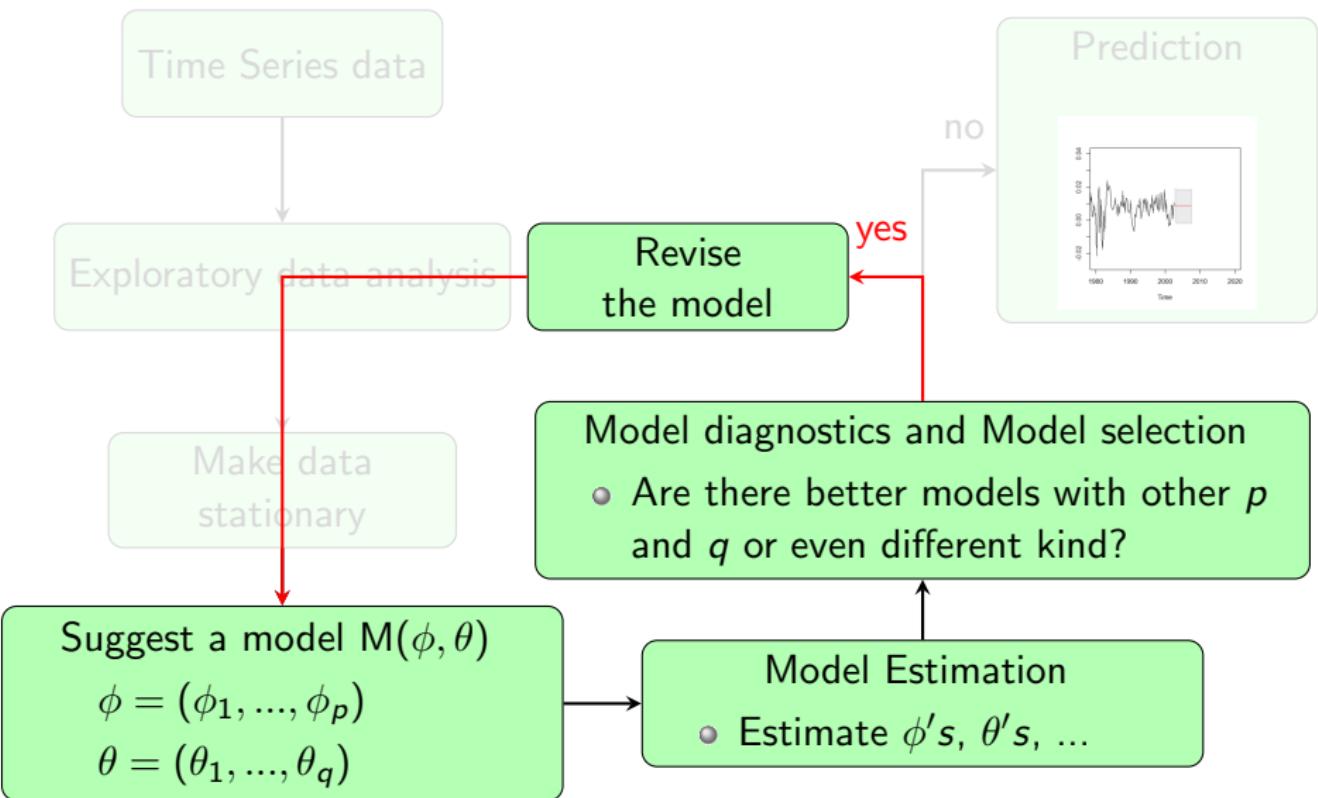
Time domain: The Big Picture



Time domain: The Big Picture



Time domain: The Big Picture



ARIMA modelling

- ARIMA models
 - ▶ AR, MA, ARMA, ARIMA, seasonal ARIMA
 - ▶ PACF
 - ▶ Model selection
 - ▶ Estimation
 - ▶ Forecasting

ARIMA models

Time series models so far

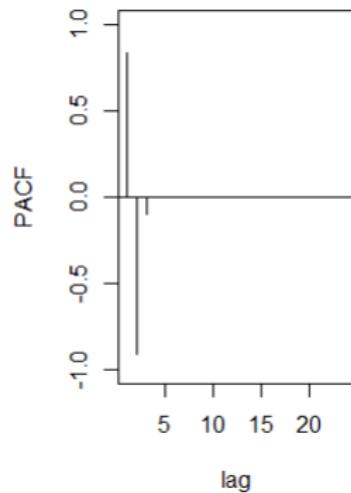
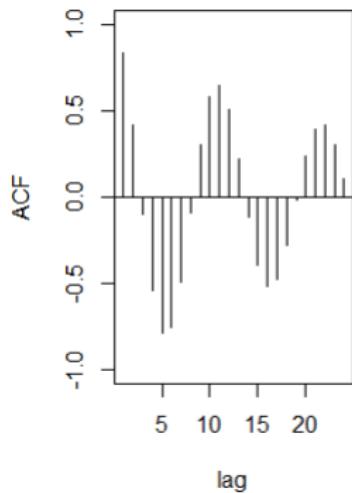
$$\phi^P(B)x_t = \theta^q(B)w_t$$

Model	Concise form
AR(p)	$\phi^P(B)x_t = w_t$
MA(q)	$x_t = \theta^q(B)w_t$
ARMA(p, q)	$\phi^P(B)x_t = \theta^q(B)w_t$
ARIMA(p, d, q)	$\phi^P(B)(1 - B)^d x_t = \theta^q(B)w_t$
ARMA($P, Q)_s$	$\Phi^P(B^s)x_t = \Theta^Q(s)w_t$
ARIMA($P, D, Q)_s$	$\Phi^P(B^s)(1 - B^s)^D x_t = \Theta^Q(B^s)w_t$
ARMA($p, q) \times (P, Q)_s$	$\Phi^P(B^s)\phi^P(B)x_t = \Theta^Q(B^s)\theta^q(B)w_t$
ARIMA($p, d, q) \times (P, D, Q)_s$	$\Phi^P(B^s)\phi^P(B)(1 - B^s)^D(1 - B)^d x_t = \Theta^Q(B^s)\theta^q(B)w_t$

* The notation used in this slide deviates from the notation used in the course literature so far.

PACF for AR(p)

- Example: AR(3) $\phi_1 = 1.5$, $\phi_2 = -0.75$, $\phi_3 = -0.1$



Seasonal?

ACF and PACF

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

How to differentiate between ARMA(p, q)?

Empirical ACF (EACF)

Idea:

- ARMA(p,q): $x_t = \sum_{j=1}^p \phi_j x_{t-j} + \sum_{j=1}^q \theta_j w_{t-j} + w_t$
- If we can estimate $\phi_j \rightarrow x'_t = x_t - \sum_{j=1}^p \phi_j x_{t-j}$ is linear function in w_t, \dots, w_{t-q}
- If we run regression x'_t against $w_t \dots w_{t-j}$:
 - ▶ Residuals are white noise, $j \geq q \rightarrow$ ACFs not significant
 - ★ Some of the coefficients will be 0
 - ▶ Residuals are not white noise, $j < q \rightarrow$ ACFs significant
 - ▶ Note: w_t s substituted by lagged residuals from a series of regressions
- If $x'_t = x_t - \sum_{j=1}^k \phi_j x_{t-j}, k < p \rightarrow$ white noise will never be achieved
 \rightarrow ACFs are not zero

Empirical ACF (EACF)

- $k > p$ General result: ACFs are 0 for $j > q + (k - p)$
 - ▶ Example: ARMA(0,1)
- General conclusion for AR,MA = (k,j):
 - ▶ This is theoretical one! → not exactly the same for the samples

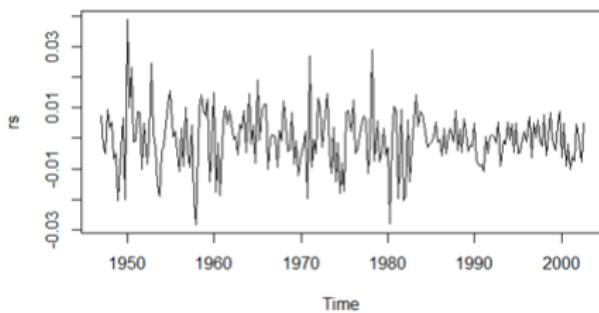
AR/MA	0	1	2
0	X	X	X	X	X	X	X
1	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X
...	X	X	X	X	X	X	X
...	X	X	X	X	X	X	X
...	X	X	0	0	0	0	0
...	X	X	X	0	0	0	0
...	X	X	X	X	0	0	0
...	X	X	X	X	X	0	0

Residual analysis

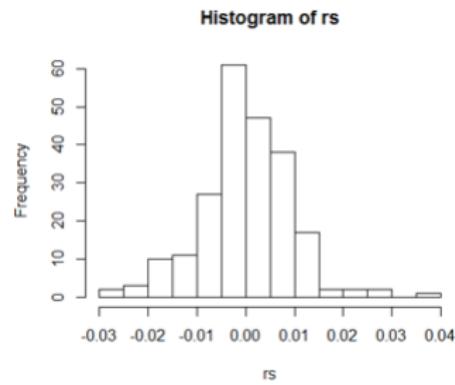
- Residuals $r_t = x_t - \hat{x}_t^{t-1}$? they are innovations
 - ▶ Note: computed from one-step-ahead predictions!
 - ▶ Measures predictive quality of the model (compare OLS)
- Residual analysis
 - ▶ Visual inspection: stationary? Patterns?
 - ▶ Histograms, Q-Q plots
 - ▶ ACF, PACF
 - ▶ Runs test
 - ▶ Box-Ljung test

Residual analysis - Visual inspection

Histogram and visual inspection

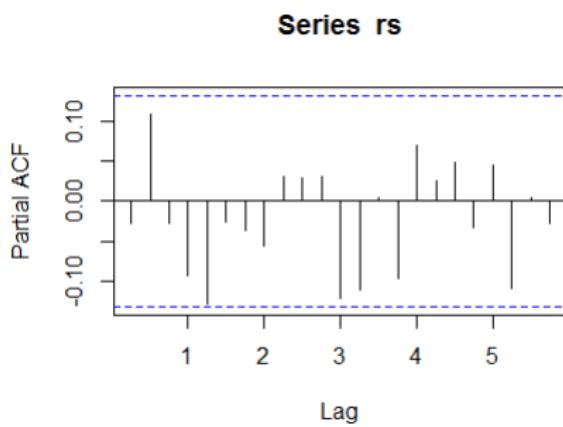


If looks white is good

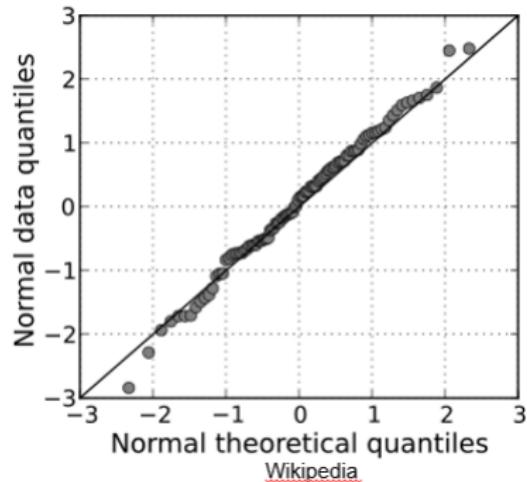


If looks Normal is good

Residual analysis - ACF /PACF Q-Q plots



If between the blue lines good



If along the diagonal line GOOD

Statistical tests

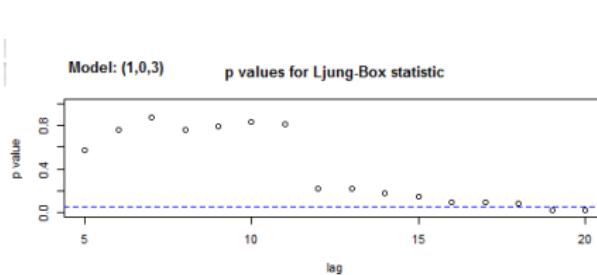
Tests are used to test independence

Runs test

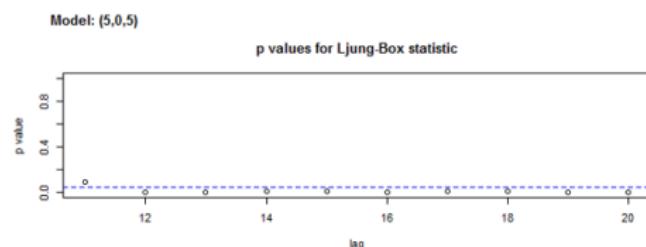
- H_0 : x_t values are i.i.d. **p-value NOT small**
- H_a : x_t values are not i.i.d. **p-value small**

Box-Ljung test

- H_0 : data are independent **p-value NOT small**
- H_a : data are not independent **p-value small**



GOOD



BAD

SARIMA

- Multiplicative seasonal autoregressive integrated moving average model $ARIMA(p, d, q) \times (P, D, Q)_s$

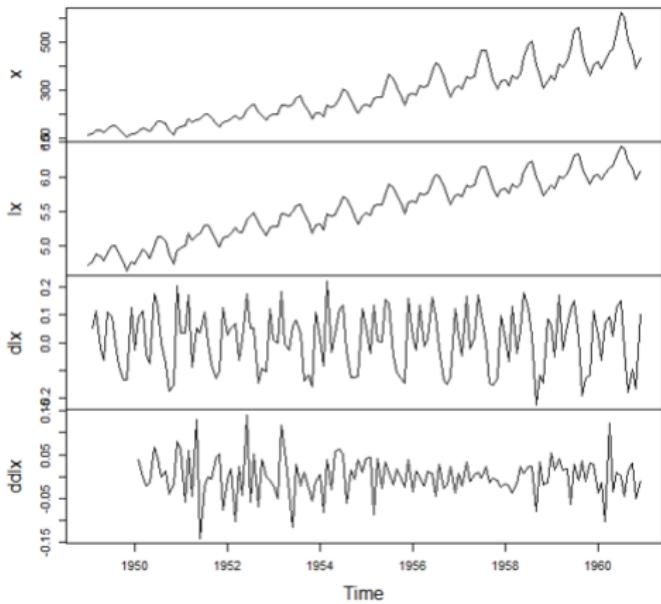
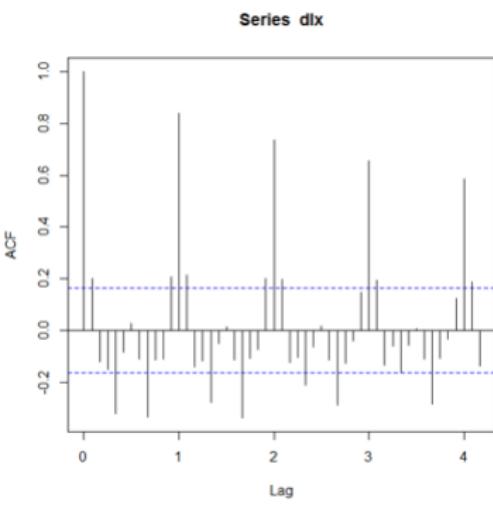
$$\Phi_p(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \delta + \Theta_Q(B^s)\theta(B)w_t$$

$$\nabla_s^D = (1 - B^s)^D$$

- How to identify SARIMA?
 - ① Perform differencing first (trend)
 - ② Investigate ACF → slowly decays at peaks?
 - ① Yes → Additional differencing by ∇_s^D
 - ③ Model non-seasonal part
 - ④ Model seasonal part (check peaks), check ACF and PACF of residuals

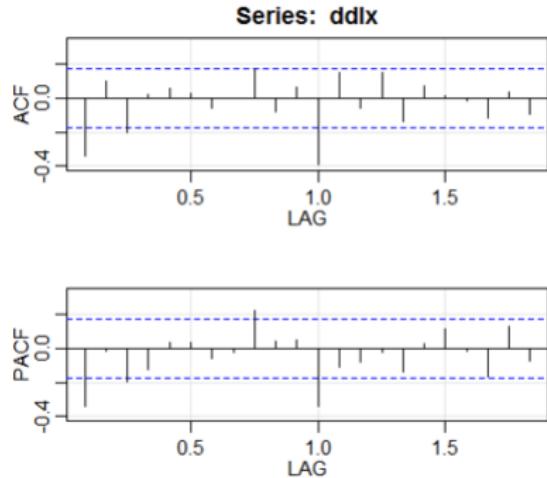
SARIMA

- Example: Air passengers



SARIMA

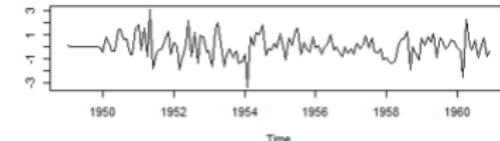
- Example: Air passengers



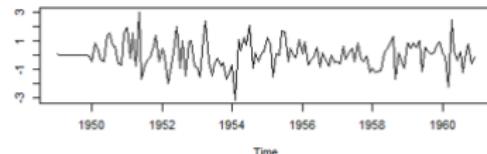
$(0, 1, 1)_{12}$ or $(1, 1, 0)_{12}$

SARIMA

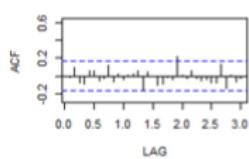
Model: (1,1,1) (0,1,1) Standardized Residuals



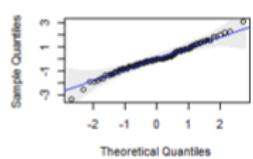
Model: (1,1,1) (1,1,0) Standardized Residuals



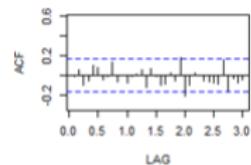
ACF of Residuals



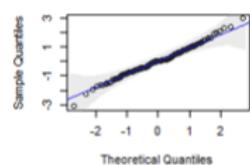
Normal Q-Q Plot of Std Residuals



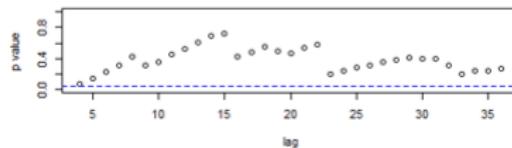
ACF of Residuals



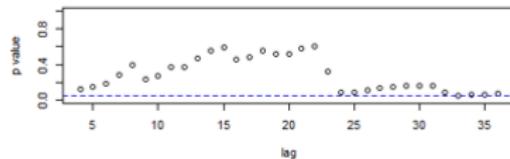
Normal Q-Q Plot of Std Residuals



p values for Ljung-Box statistic



p values for Ljung-Box statistic



SARIMA

- Remove AR term!

Is one model much better than the other one?

```
> m1$fit
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
REPORT = 1, reltol = tol))

Coefficients:
            ar1      ma1      sar1
0.0547   -0.4886  -0.4731
s.e.  0.2161    0.1933   0.0800

sigma2 estimated as 0.001425:  log likelihood = 241.73,  aic = -475.47
> m2$fit
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
REPORT = 1, reltol = tol))

Coefficients:
            ar1      ma1      sma1
0.1960   -0.5784  -0.5643
s.e.  0.2475    0.2132   0.0747

sigma^2 estimated as 0.001341:  log likelihood = 244.95,  aic = -481.9
```

$(1, 1, 1) \times (1, 1, 0)_{12}$

$(1, 1, 1) \times (0, 1, 1)_{12}$

State space modelling

- State space models
 - ▶ Linear and Gaussian state space models
 - ▶ Kalman filtering, Kalman smoothing and Forecasting
 - ▶ Maximum likelihood estimate of the state space models
 - ▶ Stochastic volatility

Consider an AR(2) model

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

Let $\mathbf{z}_t = \begin{bmatrix} x_t \\ x_{t-1} \end{bmatrix}$ and $e_t = \begin{bmatrix} w_t \\ 0 \end{bmatrix}$.

Show that we rewrite the AR(2) model in the state space form:

$$\begin{aligned}\mathbf{z}_t &= \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \mathbf{z}_{t-1} + e_t \\ x_t &= [1 \ 0] \mathbf{z}_t,\end{aligned}$$

$$\phi^p(B)x_t = \theta^q(B)w_t$$

Can we rewrite any model of this form as a state space model?

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + e_t,$$

$$\mathbf{x}_t = C\mathbf{z}_t + \nu_t,$$

$$\phi^p(B)x_t = \theta^q(B)w_t$$

Outline of the solution:

Let $r = \max(p, q + 1)$,

$$\phi^r(B) = 1 - \phi_1 B - \cdots - \phi_r B^r,$$

$$\theta^r(B) = 1 + \theta_1 B + \cdots + \theta_{r-1} B^{r-1},$$

$\phi^r(B)(\theta^r(B))^{-1}x_t = w_t$. Hence, for $z_t = (\theta^r(B))^{-1}x_t$ we can have

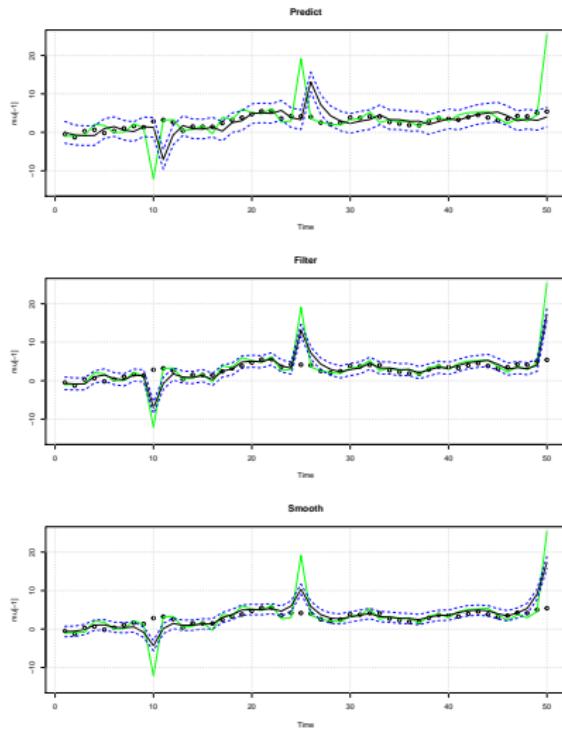
$$\phi^r(B)z_t = w_t$$

$$z_t = \begin{bmatrix} z_t \\ z_{t-1} \\ z_{t-2} \\ \vdots \\ z_{t-r+1} \end{bmatrix} \text{ and } z_t = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_r \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} z_{t-1} + \begin{bmatrix} w_t \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$x_t = [1 \ \theta_1 \ \theta_2 \ \cdots \ \theta_r] z_t$$

Robustness to outliers:filter versus smoother

Live example in Rstudio



Stochastic Volatility : Gaussian sum filter

The problem is finding the filtering distribution of $\mathbf{z}_t | \mathbf{x}_{1:t}$ when

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + w_t$$

$$\mathbf{x}_t = C\mathbf{z}_t + \eta_t$$

and

$$w_t \sim iidN(0, Q)$$

$$\eta_t \sim \pi_0 N(\mu_0, R_1) + \pi_1 N(\mu_1, R_2)$$

where $\pi_0 + \pi_1 = 1$

Examination

- Most of the examination will be your Computer labs and assignments from the teaching sessions with a twist.
- You need to have a deep knowledge of the subjects covered in the lectures to get a B+ score.
- Study them over and over and make sure you have the correct solutions with you on the examination day.

Time Series Analysis – Lecture 9

Recurrent and Temporal Convolutional Networks

Fredrik Lindsten, Linköping University

2019-10-14

Aim and outline

Aim:

- Introduce two popular deep-learning-based methods for time series analysis
- Highlight some formal connections with classical models (state space and auto-regressive) that you have seen in the course.

Outline:

1. Basics of neural network models — the multi-layer perceptron
2. Linear Gaussian state space models on innovation form
3. A nonlinear generalization — Recurrent Neural Networks
4. Nonlinear auto-regressive models
5. Temporal Convolutional Networks

ex) Generating text

Input (human-written) In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Model completion (machine-written)



The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. ‘‘By the time we reached the top of one peak, the water looked blue, with some crystals on top,’’ said Pérez.

<https://openai.com/blog/better-language-models/>

State Space Models \Rightarrow
Recurrent Neural Networks

Linear state space models

Linear state space model:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + e_t,$$

$$x_t = C\mathbf{z}_t + \nu_t.$$

Limitation:

The next state \mathbf{z}_{t+1} as well as the observation x_t depend **linearly** on the current state \mathbf{z}_t .

The model flexibility is limited.

Going nonlinear

Aim: Increase the flexibility of the model by replacing the linear functions by **generic** and **flexible** nonlinear functions.

Linear function: $y = \mathbf{A}\mathbf{z}$, where the matrix \mathbf{A} is the **parameter**.

Nonlinear function: $y = f_{\theta}(\mathbf{z})$. Here, θ is a vector of **parameters** determining the shape of the function $f_{\theta}(\cdot)$.

ex) Let $\theta = (\theta_1, \theta_2, \theta_3)$, and

$$f_{\theta}(z) = \frac{\theta_1}{\theta_2 + z^{\theta_3}}.$$

Neural networks

Recall: We want to use **generic** and **flexible** nonlinear functions.

This is precisely what **neural networks** provide!

Fully connected, 1-layer network:

We **construct** a function $f_{\theta} : \mathbb{R}^p \mapsto \mathbb{R}$ by

$$\begin{aligned}\mathbf{h} &= \sigma(W^{(1)}\mathbf{z} + b^{(1)}) \\ y &= W^{(2)}\mathbf{h} + b^{(2)}.\end{aligned}$$

That is,

$$f_{\theta}(\mathbf{z}) = W^{(2)}\sigma(W^{(1)}\mathbf{z} + b^{(1)}) + b^{(2)}$$

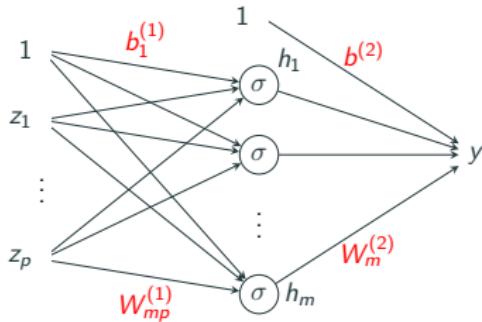
Neural network – graphical illustration

Input variables Hidden units Output

The equations

$$\mathbf{h} = \sigma(W^{(1)}\mathbf{z} + b^{(1)})$$
$$y = W^{(2)}\mathbf{h} + b^{(2)}.$$

can be illustrated graphically.

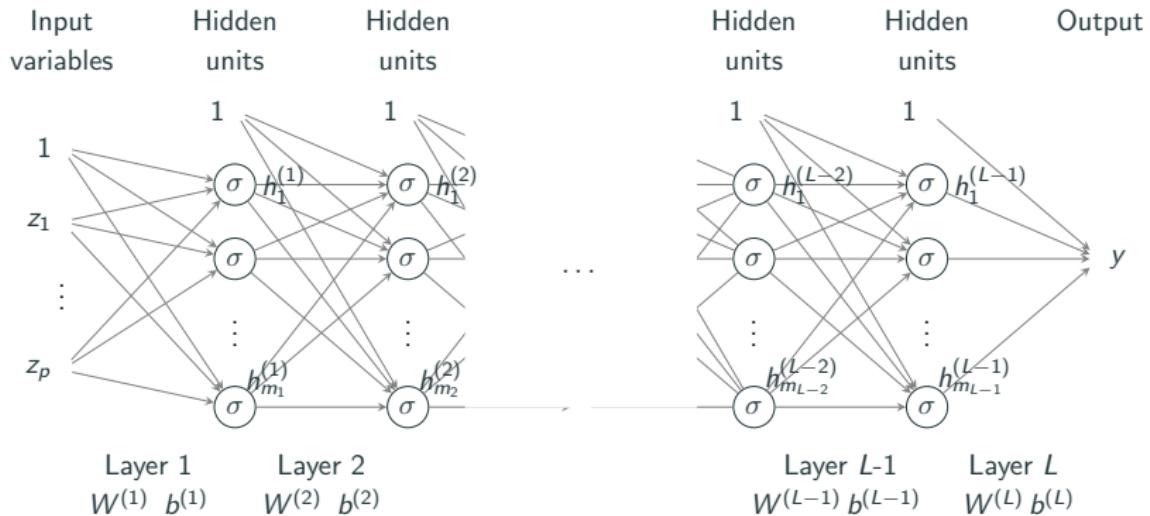


- The variables $\mathbf{h} = (h_1, \dots, h_m)$ are referred to as a **hidden layer**.
- The function $\sigma(\cdot)$ is an element-wise nonlinearity, referred to as an **activation function**. Typical choices are

$$\sigma(x) = \tanh(x) \quad \text{or} \quad \sigma(x) = \text{ReLU}(x) = x \mathbb{1}(x \geq 0)$$

- The model **parameters** are the weight matrices and bias vectors $\theta = \{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}\}$.

Multi-layer perceptron



Innovation form

Linear state space model:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + e_t,$$

$$x_t = C\mathbf{z}_t + \nu_t.$$

Innovation form. There exists an **equivalent** representation given by

$$\mathbf{h}_t = W\mathbf{h}_{t-1} + Ux_{t-1},$$

$$x_t = C\mathbf{h}_t + \nu'_t.$$

(Assuming stationarity for simplicity.)

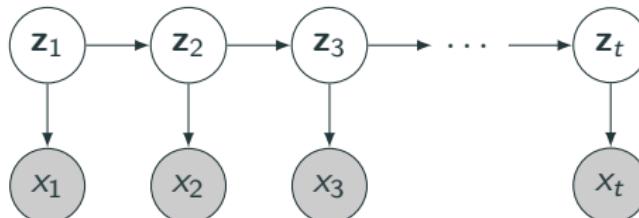
Proof. Let $\mathbf{h}_t = m_{t|t-1}$, the Kalman predictive mean.

Innovation form

Original form:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \mathbf{e}_t,$$

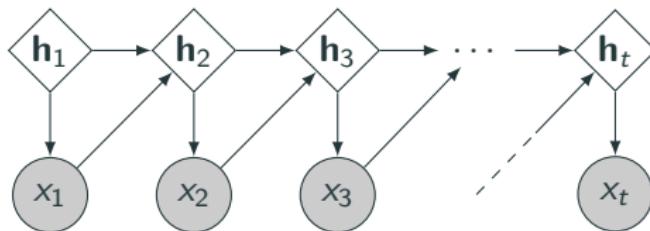
$$\mathbf{x}_t = C\mathbf{z}_t + \mathbf{\nu}_t.$$



Innovation form:

$$\mathbf{h}_t = W\mathbf{h}_{t-1} + U\mathbf{x}_{t-1},$$

$$\mathbf{x}_t = C\mathbf{h}_t + \mathbf{\nu}'_t.$$



The hidden state variables can be **deterministically and recursively computed** from the data.

Going nonlinear

Doesn't this look suspiciously similar to an MLP...?

$$\mathbf{h}_t = W\mathbf{h}_{t-1} + Ux_{t-1},$$

$$x_t = C\mathbf{h}_t + \nu'_t,$$

for some **nonlinear activation function** $\sigma(\cdot)$.

This is a basic **Recurrent Neural Network (RNN)**.

Learning the parameters

The model parameters are the weight matrices and bias vectors:

$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}x_{t-1} + \mathbf{b}),$$

$$x_t = \mathbf{C}\mathbf{h}_t + \mathbf{c} + \nu'_t,$$

with $\theta = \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{C}, \mathbf{c}\}$.

Note:

- The parameters are the same for all time steps (“weight sharing”).
- The fact that there is no state noise means that we can compute

$$p_\theta(x_t | x_{1:t-1}) = N(x_t | \mathbf{C}\mathbf{h}_t + \mathbf{c}, \sigma_{\nu'}^2).$$

Learning the parameters

We can thus learn the parameters θ directly by optimizing the negative log-likelihood,

$$L(\theta; x_{1:T}) = - \sum_{t=1}^T \log p_\theta(x_t | x_{1:t-1}),$$

using gradient-based optimization.

The gradient $\nabla_\theta L(\theta; x_{1:T})$ is computed using the chain rule of differentiation, propagating information from $t = 1$ to $t = T$ and then back again.

⇒ Back-propagation through time.

RNN extensions

- GRU/LSTM
- Non-Gaussian likelihood (e.g., for discrete data)
- Conditioning on context (input)
- Stochastic hidden layers
- Bidirectional connections
- ...

Autoregressive Models \Rightarrow
Temporal Convolutional Nets

Autoregressive models

State space models and RNNs use a latent state vector to model temporal dependencies.

An alternative is to model the dependency of the current data point x_t on the past data points $x_{1:t-1}$ by a **direct functional relationship**.

Auto-regressive model, AR(p):

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t, \quad w_t \sim N(0, \sigma_w^2).$$

The AR model is linear in the parameters:

- ▲ Learning of parameters easy \Leftrightarrow linear regression
- ▼ Flexibility/ability to model complex temporal dependencies is limited
- ▼ Memory/receptive field is just p time steps

Going nonlinear

Nonlinear auto-regressive model, NAR(p):

$$x_t = \sigma(\phi_1 x_{t-1} + \cdots + \phi_p x_{t-p}) + w_t, \quad w_t \sim N(0, \sigma_w^2),$$

for some nonlinear activation function σ .

- ▲ Flexibility increased...
- ▼ ...but only slightly!
- ▼ Memory/receptive field is *still* just p steps

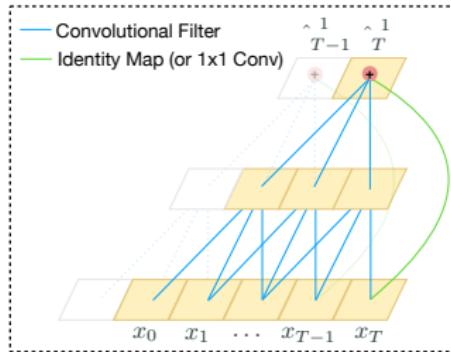
We can address these issues with a **multi-layer network architecture!**

Temporal Convolutional Network

2-layer TCN:

$$h_{t-1} = \sigma(\phi_1^{(1)}x_{t-1} + \cdots + \phi_p^{(1)}x_{t-p}),$$
$$x_t = \sigma(\phi_1^{(2)}h_{t-1} + \cdots + \phi_p^{(2)}h_{t-p}) + w_t,$$

with $w_t \sim N(0, \sigma_w^2)$.

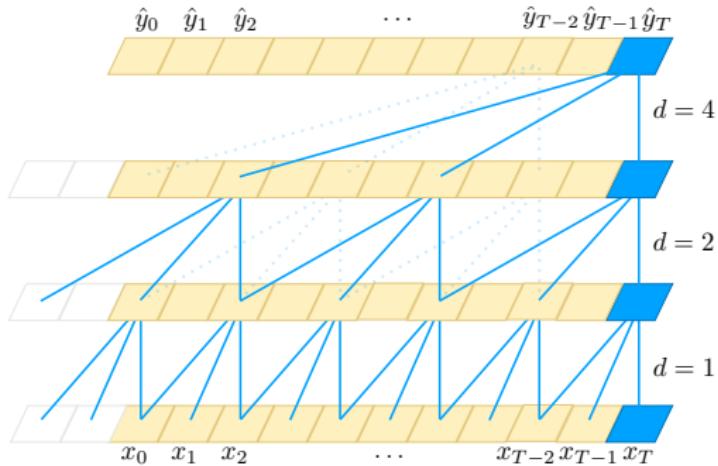


Can extend to multiple hidden layers, $h_t^{(1)}, h_t^{(2)}, \dots$

- ▲ Multiple layers \Rightarrow very flexible models
- ▲ Receptive field increases with depth...
- ▼ ... but only linearly

TCN with dilated convolutions

By using **dilated convolutions** we can increase receptive field **exponentially** with depth.



RNN vs TCN

RNNs are still the *de facto* standard deep learning approach to time series modeling, *but...*

...TCNs have outperformed them on many benchmark problems

 Shaojie Bai, J. Zico Kolter, Vladlen Koltun. **An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.** arXiv.org: 1803.01271, 2018.

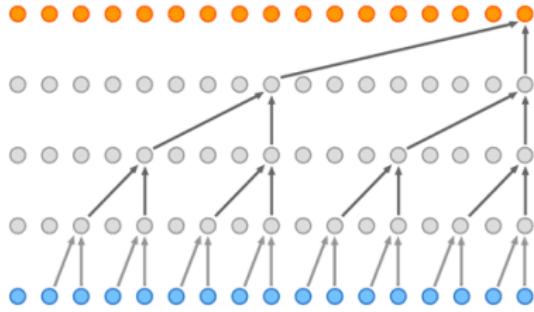
...and have other advantages too: ... but also some disadvantages:

- Easier parallelization
- Better control over receptive field
- Lower memory requirements during training
- ...
- More difficult to reuse model for multiple tasks
- Larger memory requirements after deployment
- ...

ex) WaveNet



WaveNet by DeepMind powers
Google's text-to-speech technology.



Deep Learning for TSA

So is this what we should always do for modeling sequential data?!

No!

- Only makes sense to use something as complex as RNN or TCN when classical methods fail — **Try Simple Things First!**
- Methods based on deep learning work best if we have multiple sequences, or one long sequence that can be split into segments
- For a single univariate time series, classical methods (ARIMA, state space, ...) often work better.

1 Lectures 1-3

- Probability density function for x : $f(x)$
- Marginal density $f_i(x_i) = \int f(x) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_p$
- Expected (mean) value $Ex = \int xf(x)dx$
- Covariance $\text{cov}(x, y) = E\{(x - Ex)(y - Ey)\}$
- Correlation $\rho_{x,y} = \text{corr}(x, y) = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$
- Variance $\text{var}(x) = E\{(x - Ex)^2\} = \text{cov}(x, x)$
- Relationships (a is a constant)
 - $E(x + a) = Ex + a$, $E(ax) = aEx$
 - $E(x + y) = Ex + Ey$
 - $\text{cov}(x + a, y) = \text{cov}(x, y)$
 - $\text{cov}(x + z, y) = \text{cov}(x, y) + \text{cov}(z, y)$
 - $\text{var}(ax) = a^2 \text{var}(x)$

uncorrelated $\iff E(XY) = EX.EY$
 independent $\iff f_{X,Y}(x, y) = f_X(x).f_Y(y)$

- Autocovariance function

$$\gamma(s, t) = \text{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)]$$

Note: $\text{var}(x_t) = \gamma(t, t)$

- Autocorrelation function (ACF)

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{\gamma(s, s)\gamma(t, t)}}$$

Useful fact: If $U = \sum_{j=1}^m a_j x_j$ and

$$V = \sum_{k=1}^r b_k y_k$$

$$\text{cov}(U, V) = \sum_{j=1}^m \sum_{k=1}^r a_j b_k \text{cov}(x_j, y_k)$$

1.1 stationarity

- Time series x_t is weakly stationary (stationary) if
 - $Ex_t = \text{const}$
 - $\gamma(s, t) = \gamma(|s - t|)$
 - $\text{var}(x_t) < \infty$
- $\gamma(t, t + h) = \gamma(|t + h - t|) = \gamma(h)$
 - Autocovariance depends on lag only!
- Autocovariance for stationary process
 $\gamma(h) = \text{cov}(x_t, x_{t+h})$
- ACF for stationary process $\rho(h) = \frac{\gamma(h)}{\gamma(0)}$

Properties of stationary process:

$$\gamma(h) = \gamma(-h) \quad \rho(h) = \rho(-h)$$

$$|\gamma(h)| \leq \gamma(0) \quad \rho(h) \leq 1, \rho(0) = 1$$

If x_t is stationary,

- Sample mean

$$Ex \approx \bar{x} = \frac{1}{n} \sum_{t=1}^n x_t$$

- Sample autocovariance function

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x})$$

Theorem: Under weak conditions,
 if x_t is white noise and $n \rightarrow \infty$
 then $\hat{\rho}(h)$ is approximately $N(0, \frac{1}{n})$

Consequence: If some $|\hat{\rho}(h)| > \frac{2}{\sqrt{n}}$ then the time series is not a white noise (with approximately 95 % confidence).

1.2 Backshift operator

- Backshift operator $Bx_t = x_{t-1}$,
Powers $B^k x_t = x_{t-k}$
- Forward-shift operator $B^{-1}x_t = x_{t+1}$
- Note $BB^{-1}x_t = x_t$ (i.e. $BB^{-1} = 1$)
- Differencing $\nabla x_t = (1 - B)x_t$
- Differences of order d : $\nabla^d = (1 - B)^d$
- Property: Operators can be manipulated as polynomials
- Example Check that $\nabla^2 x_t = x_t - 2x_{t-1} + x_{t-2}$
- Property: Differencing of order p can remove polynomial trend of order p

• Autoregressive operator

$$\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$$

- AR(p) model

$$\boxed{\phi(B)x_t = w_t}$$

• ARMA(p,q)

$$\begin{aligned} x_t = & \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} \\ & + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q} \\ - & \phi_p \neq 0, \theta_q \neq 0 \\ - & \text{Is stationary} \\ - & E x_t = 0 \end{aligned}$$

1.3 MA, AR, ARMA

- Moving average model of order q, MA(q)

$$\begin{aligned} x_t = & w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q} \\ = & \sum_{j=0}^q \theta_j w_{t-j} \end{aligned}$$

- $w_t \sim \text{wn}(0, \sigma_w^2)$
- $\theta_1, \dots, \theta_q$ constants, $\theta_q \neq 0$ and $\theta_0 = 1$

- Moving average operator

$$\theta(B) = \sum_{j=0}^q \theta_j B^j$$

- MA(q):

$$\boxed{x_t = \theta(B)w_t}$$

- Autoregressive model of order p, AR(p)

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t$$

- x_t is stationary if x_0 is sampled from the stationary distribution
- $w_t \sim \text{wn}(0, \sigma_w^2)$
- ϕ_1, \dots, ϕ_p constants, $\phi_p \neq 0$
- $E x_t = 0$

1.4 Causality / invertibility

A stationary process is **causal** if it is only dependent on the past values of the process

Def: A linear process is **nonexplosive** and **causal** if it can be written as a one-sided sum:

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B)w_t$$

where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$ and $\sum_{j=0}^{\infty} |\psi_j| < \infty$.

Def: An MA process is **invertible** if it has a causal AR representation,

$$w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$$

Def: Linear process is **causal** and **nonexplosive** if

- $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$ (depends on the past only)
- $\sum_{j=0}^{\infty} |\psi_j| < \infty$
- We set $\psi_0 = 1$ by convention.

Property: ARMA(p,q) is **causal** iff roots $\phi(z') = 0$ are outside unit circle, i.e. $|z'| > 1$

$$\boxed{\phi(B)x_t = \theta(B)w_t}$$

Def: ARMA(p,q) is **invertible** if

- $w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$ (depends on the past only)
- $\sum_{j=0}^{\infty} |\pi_j| < \infty$

Property: ARMA(p,q) is **invertible** iff roots $\theta(z') = 0$ are outside unit circle, i.e. $|z'| > 1$

$$\boxed{\phi(B)x_t = \theta(B)w_t}$$

Teaching session I

Instructions

The assignments in the first section will be solved by the teacher during the teaching session. You are welcome to ask questions if you cannot follow the derivations. The problems in the second section are take home exercises and the key is given in section 3. The following assignments are hand-in and no solution is given in the key.

Assignment 12

Assignment 18

The hand-in assignment should be solved individually and should be submitted via LISAM in pdf format before the deadline also specified in LISAM.

The solutions are graded pass / insufficient. An insufficient solution can be completed and resubmitted.

1. Assignments solved by the teacher

Assignment 1

Suppose $E(X) = 2$, $\text{var}(X) = 9$, $E(Y) = 0$, $\text{var}(Y) = 4$, and $\text{corr}(X, Y) = 0.25$. Find:

- (a) $\text{var}(X + Y)$.
- (b) $\text{cov}(X, X + Y)$.

Assignment 2

Suppose $y_t = 5 + 2t + x_t$, where $\{x_t\}$ is a zero-mean stationary series with autocovariance function γ_k .

- (a) Find the mean function for $\{y_t\}$.
- (b) Find the autocovariance function for $\{y_t\}$.
- (c) Is $\{y_t\}$ stationary? Why or why not?

Assignment 3

Suppose that $\{x_t\}$ is stationary with autocovariance function γ_k . Show that for any fixed positive integer n and any constants c_1, c_2, \dots, c_n , the process $\{y_t\}$ defined by $y_t = \sum_{i=1}^n c_i x_{t-i+1}$ is stationary.

Assignment 4

Suppose that $x_t = w_t - w_{t-12}$. Show that $\{x_t\}$ is stationary and that, for $k > 0$, its autocorrelation function is nonzero only for lag $k = 12$.

Assignment 5

Suppose $x_t = \mu + w_t - w_{t-1}$. Find $\text{var}(\bar{x})$. Note any unusual results. In particular, compare your answer to what would have been obtained if $x_t = \mu + w_t$.

Assignment 6

Calculate and sketch the autocorrelation functions for AR(1) model with $\phi = 0.6$. Plot for sufficient lags that the autocorrelation function has nearly died out.

Assignment 7

Let $\{x_t\}$ be an AR(2) process $x_t = \phi x_{t-2} + w_t$. Find the range of values of ϕ for which the process is causal.

Assignment 8

For each of the following ARMA models, find the roots of the AR and MA polynomials, identify the values of p and q for which they are ARMA(p,q) (be careful of parameter redundancy), determine whether they are causal, and determine whether they are invertible. In each case, $w_t \sim WN(0,1)$.

- a) $x_t + 0.81x_{t-2} = w_t + 1/3w_{t-1}$
- b) $x_t - x_{t-1} = w_t - 0.5w_{t-1} - 0.5w_{t-2}$

Assignment 9

For those the following model, compute the first four coefficients ψ_0, \dots, ψ_3 in the causal linear process representation $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$

- a) $x_t + 0.81x_{t-2} = w_t + 1/3w_{t-1}$

2. Take home assignments

Assignment 10

Suppose $E(X) = 2$, $Var(X) = 9$, $E(Y) = 0$, $Var(Y) = 4$, and $Corr(X, Y) = 0.25$. Find:

- (a) $Corr(X + Y, X - Y)$.

Assignment 11

Let $\{w_t\}$ be a zero mean white noise process. Suppose that the observed process is $x_t = w_t + \theta w_{t-1}$, where θ is either 3 or 1/3.

- (a) Find the autocorrelation function for $\{x_t\}$ both when $\theta = 3$ and when $\theta = 1/3$.
- (b) You should have discovered that the time series is stationary regardless of the value of θ and that the autocorrelation functions are the same for $\theta = 3$ and $\theta = 1/3$. For simplicity, suppose that the process mean is known to be zero and the variance of y_t is known to be 1. You observe the series $\{y_t\}$ for $t = 1, 2, \dots, n$ and suppose that you can produce good estimates of the autocorrelations ρ_k . Do you think that you could determine which value θ is correct (3 or 1/3) based on the estimate of ρ_k ? Why or why not?

Assignment 12

Let $\{x_t\}$ be a zero-mean, unit-variance stationary process with autocorrelation function ρ_h . Suppose that μ_t is a nonconstant function and that σ_t is a positive-valued nonconstant function. The observed series is formed as $y_t = \mu_t + \sigma_t x_t$.

- (a) Find the mean and covariance function for the $\{y_t\}$ process.
- (b) Show that the autocorrelation function for the $\{y_t\}$ process depends only on the time lag. Is the $\{y_t\}$ process stationary?
- (c) Is it possible to have a time series with a constant mean and with $Corr(y_t, y_{t+h})$ free of t but with $\{y_t\}$ not stationary?

Assignment 13

Suppose that x is a random variable with zero mean. Define a time series by

$$y_t = (-1)^t x$$

- (a) Find the mean function for $\{y_t\}$.
- (b) Find the autocovariance function for $\{y_t\}$.
- (c) Is $\{y_t\}$ stationary?

Assignment 14

Suppose $x_t = \mu + w_t + w_{t-1}$. Find $\text{var}(\bar{x})$. Note any unusual results. In particular, compare your answer to what would have been obtained if $x_t = \mu + w_t$.

Assignment 15

Calculate and sketch the autocorrelation function for MA(2) model with $\theta_1 = 0.5$ and $\theta_2 = 0.4$

Assignment 16

Describe the important characteristics of the autocorrelation function for the following models: (a) MA(1), (b) MA(2), (c) AR(1), (d) AR(2), and (e) ARMA(1,1).

Assignment 17

Suppose that $\{x_t\}$ is an AR(1) process with $-1 < \phi < +1$.

(a) Find the autocovariance function for $y_t = \nabla x_t = x_t - x_{t-1}$ in terms of ϕ and σ_w^2

(b) In particular, show that $\text{var}(y_t) = \frac{2\sigma_w^2}{1+\phi}$

Assignment 18

For each of the following ARMA models, find the roots of the AR and MA polynomials, identify the values of p and q for which they are ARMA(p,q) (be careful of parameter redundancy), determine whether they are causal, and determine whether they are invertible. In each case, $w_t \sim WN(0,1)$.

- c) $x_t - 3x_{t-1} = w_t + 2w_{t-1} - 8w_{t-2}$
- d) $x_t - 2x_{t-1} + 2x_{t-2} = w_t - \frac{8}{9}w_{t-1}$
- e) $x_t - 4x_{t-2} = w_t - w_{t-1} + 0.5w_{t-2}$
- f) $x_t - \frac{9}{4}x_{t-1} - \frac{9}{4}x_{t-2} = w_t$

Assignment 19

For the following models, compute the first four coefficients ψ_0, \dots, ψ_3 in the causal linear process representation $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$

- a) $x_t - 2x_{t-1} + 2x_{t-2} = w_t - \frac{8}{9}w_{t-1}$
- b) $x_t - \frac{9}{4}x_{t-1} - \frac{9}{4}x_{t-2} = w_t - 3w_{t-1} + \frac{1}{9}w_{t-2} - \frac{1}{3}w_{t-3}$

3. Key

Assignment 10

Approximately 0.39

Assignment 11

$\rho(0) = 1, \rho(1) = 0.3, \rho(h) = 0$ otherwise

Assignment 13

- a) 0
- b) $(-1)^h \sigma_x^2$
- c) Yes

Assignment 14

$$var(\bar{x}) = \frac{2(2n-1)}{n^2} \sigma_w^2$$

Assignment 15

$$\rho_1 \approx 0.5, \rho_2 \approx 0.28, \rho_i = 0, i > 2$$

Assignment 16

a) $-\frac{1-\phi}{1+\phi} \phi^{h-1} \sigma_w^2$

Assignment 17

- a) p=1,q=2, neither causal or invertible
- b) p=2,q=1, invertible, but not causal
- c) p=2,q=2, invertible, but not causal
- d) p=2, q=0, invertible, not causal

Assignment 19

a) 1, 10/9, 2/9, -16/9

b) 1, -3/4, 1/9+9/16, -1/12-27/64

Teaching session II

Instructions

The hand-in assignment should be solved individually and should be submitted via LISAM in pdf format before the deadline also specified in LISAM. For the best learning outcome, you are encouraged to solve the problem by pen and paper and take a photo in pdf format and submit. However, non-hand-written solutions are equally accepted by the teacher.

The solutions are graded pass / insufficient. An insufficient solution can be completed and resubmitted.

1. Hand in assignment

Assignment 1

Using the example solved during lecture 6 find the state space representation for a multiplicative seasonal ARIMA model $\text{ARIMA}(p, d, q) \times (P, D, Q)_s$. This process can be written as in the following using the backshift operator.

$$\Phi^P(\mathbf{B}^s)\phi^p(\mathbf{B})(\mathbf{1} - \mathbf{B}^s)^D(\mathbf{1} - \mathbf{B})^d x_t = \Theta^Q(\mathbf{B}^s)\theta^q(\mathbf{B})w_t$$

If the parametric representation imposes prohibitive barriers for you to solve the problem, you may use the following parameters:

$$p = 3, \quad d = 2, \quad q = 1, \quad P = 2, \quad D = 1, \quad Q = 1, \quad s = 5.$$

Teaching session III

Instructions

The hand-in assignment should be solved individually and should be submitted via LISAM in pdf format before the deadline also specified in LISAM. For the best learning outcome, you are encouraged to solve the problem by pen and paper and take a photo in pdf format and submit. However, other formats are equally accepted by the teacher. The solutions are graded pass / insufficient. An insufficient solution can be completed and resubmitted.

Introduction

Useful properties of the normal density function for this assignment are listed here.

Property 1: $f(\mathbf{y}_1)f(\mathbf{y}_2|\mathbf{y}_1) = f(\mathbf{y}_1, \mathbf{y}_2)$

$$N(\mathbf{y}_1; \mu, \Sigma)N(\mathbf{y}_2; B\mathbf{y}_1, R) = N\left(\begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix}; \begin{bmatrix}\mu \\ B\mu\end{bmatrix}, \begin{bmatrix}\Sigma & \Sigma B^T \\ B\Sigma & B\Sigma B^T + R\end{bmatrix}\right)$$

Property 2: marginalization and conditioning

If $\mathbf{y}_1, \mathbf{y}_2$ were jointly normal:

$$f(\mathbf{y}_1, \mathbf{y}_2) = N\left(\begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix}; \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

then

$$\begin{aligned} f(\mathbf{y}_1) &= N(\mathbf{y}_1; \mu_1, \Sigma_{11}) \\ f(\mathbf{y}_2) &= N(\mathbf{y}_2; \mu_2, \Sigma_{22}) \\ f(\mathbf{y}_1|\mathbf{y}_2) &= N(\mathbf{y}_1; \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) \\ f(\mathbf{y}_2|\mathbf{y}_1) &= N(\mathbf{y}_2; \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{y}_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}) \end{aligned}$$

Assignment 1

Prove the Kalman filtering recursion for the following state space model with initial prior on the state $f(\mathbf{z}_1) = N(\mathbf{z}_1; m_0, P_0)$ where $e_t \sim N(0, Q_t)$ and $\nu_t \sim N(0, R_t)$

$$\mathbf{z}_t = A_{t-1}\mathbf{z}_{t-1} + e_t, \tag{1}$$

$$\mathbf{x}_t = C_t\mathbf{z}_t + \nu_t, \tag{2}$$

Particularly, show that given $f(\mathbf{z}_t|\mathbf{x}_{1:t}) = N(\mathbf{z}_t; m_{t|t}, P_{t|t})$, the predicted density $f(\mathbf{z}_{t+1}|\mathbf{x}_{1:t})$ is given by

$$f(\mathbf{z}_{t+1}|\mathbf{x}_{1:t}) = N(\mathbf{z}_{t+1}; A_t m_{t|t}, A_t P_{t|t} A_t^T + Q_{t+1}).$$

Also, show that given $f(\mathbf{z}_t | \mathbf{x}_{1:t-1}) = N(\mathbf{z}_t; m_{t|t-1}, P_{t|t-1})$, the observation updated density $f(\mathbf{z}_t | \mathbf{x}_{1:t})$ is given by

$$f(\mathbf{z}_t | \mathbf{x}_{1:t}) = N(\mathbf{z}_t; m_{t|t}, P_{t|t})$$

where

$$\begin{aligned} m_{t|t} &= m_{t|t-1} + K_t(\mathbf{x}_t - C_t m_{t|t-1}) \\ P_{t|t} &= (I - K_t C_t) P_{t|t-1} \\ K_t &= P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1}. \end{aligned}$$

Table 1: Kalman filtering recursion

-
- 1: **Inputs:** A_t , C_t , Q_t , R_t , m_0 , P_0 and $\mathbf{x}_{1:T}$.
initialization
- 2: $m_{1|0} \leftarrow m_0$, $P_{1|0} \leftarrow P_0$
- 3: **for** $t = 1$ to T **do**
- 4: *observation update step*
- 5: $K_t \leftarrow P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1}$
- 6: $m_{t|t} \leftarrow m_{t|t-1} + K_t(\mathbf{x}_t - C_t m_{t|t-1})$
- 7: $P_{t|t} \leftarrow (I - K_t C_t) P_{t|t-1}$
prediction step
- 8: $m_{t+1|t} \leftarrow A_t m_{t|t}$
- 9: $P_{t+1|t} \leftarrow A_t P_{t|t} A_t^T + Q_{t+1}$
- 9: **end for**
- 10: **Outputs:** $m_{t|t}$, $P_{t|t}$ for $t = 1 : T$
-

Time Series Analysis

Teaching session III : State Space Models, Kalman filtering
Kalman Smoothing

Tohid Ardesthiri

Linköping University
Division of Statistics and Machine Learning

September 30, 2019



Remaining Course topics

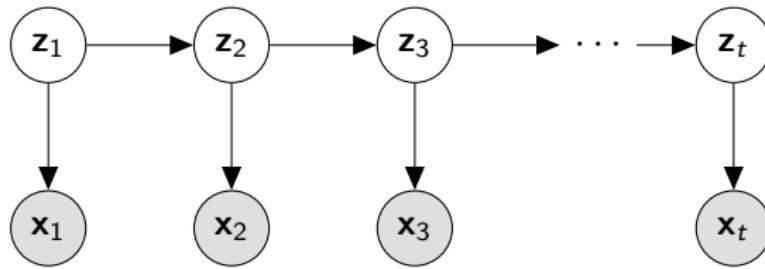
- ARIMA models
- State space models (2 lectures, 1 teaching session with hand-in, 1 computer lab with short report)
 - ▶ Linear and Gaussian state space models (Chapter 6.1)
 - ▶ Kalman filtering, Kalman smoothing and Forecasting (Chapter 6.2)
 - ▶ Maximum likelihood estimate of the state space models (Chapter 6.3)
 - ▶ Stochastic volatility (Chapter 6.11)
- Recurrent Neural Networks (RNNs) (1 lecture and 1 Computer lab No examination)
- Summary lecture

State Space models - Linear and Gaussian

Our main focus will be on linear and Gaussian models:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + e_t, \quad e_t \sim N(0, Q)$$

$$\mathbf{x}_t = C\mathbf{z}_t + \nu_t, \quad \nu_t \sim N(0, R)$$



Bayesian Inference

Bayesian inference is a means of combining prior beliefs with the data (evidence) to obtain posterior beliefs.

Example: likelihood update

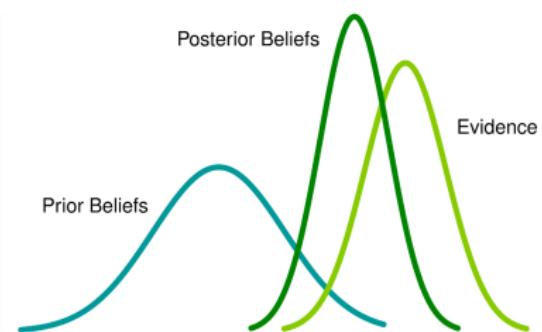
$$f(z|x) \propto f(x|z)f(z)$$

Probability Calculus

$$f(z, x) = f(z|x)f(x)$$

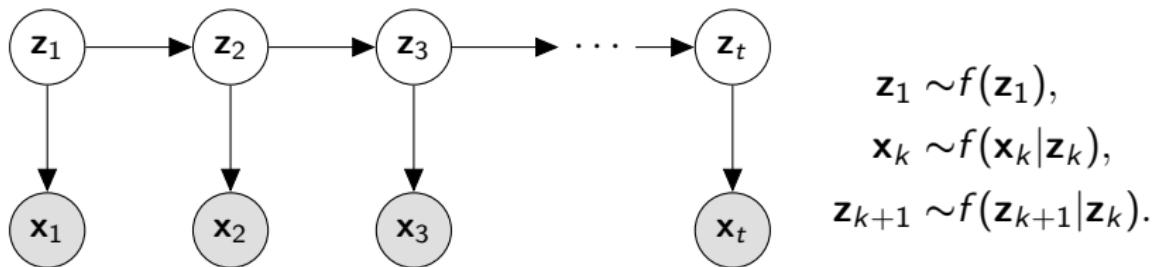
$$f(z, x) = f(x|z)f(z)$$

$$f(z) = \int f(z, x) dx$$



Online recursive algorithms

Consider a stochastic dynamical system represented by the following recursion



The Bayesian filtering recursion corresponds to computing the posterior distributions $f(z_k|x_{1:k})$;

$$f(z_k|x_{1:k}) = \frac{f(z_k|x_{1:k-1})f(x_k|z_k)}{\int f(z_k|x_{1:k-1})f(x_k|z_k) dz_k}$$

The density $f(z_k|x_{1:k-1})$ in the numerator which is called the predicted density of z_k and is obtained by integration as in

$$f(z_k|x_{1:k-1}) = \int f(z_k|z_{k-1})f(z_{k-1}|x_{1:k-1}) dz_{k-1}.$$

Kalman filter

Kalman filter is an algorithm that uses time series data, **containing statistical noise and unknown innovations**, and produces estimates of latent (hidden) process that tend to be more accurate than those based on a single observations using a probabilistic framework.

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \mathbf{e}_t,$$

$$\mathbf{x}_t = C\mathbf{z}_t + \mathbf{\nu}_t,$$



The Kalman Filter's Foundation

Let \mathbf{z} have a normal prior distribution with mean μ and covariance Σ , i.e., $\mathbf{z} \sim N(\mathbf{z}; \mu, \Sigma)$.

An observation \mathbf{x} with the likelihood function $f(\mathbf{x}|\mathbf{z}) = N(\mathbf{x}; C\mathbf{z}, R)$ is in hand where C is a matrix with proper dimensions and R is a covariance matrix. The posterior distribution of \mathbf{z} can be obtained using the Bayes' rule

$$\begin{aligned} f(\mathbf{z}|\mathbf{x}) &= \frac{f(\mathbf{z})f(\mathbf{x}|\mathbf{z})}{\int f(\mathbf{z})f(\mathbf{x}|\mathbf{z}) d\mathbf{z}} \\ &= \frac{N(\mathbf{z}; \mu, \Sigma)N(\mathbf{x}; C\mathbf{z}, R)}{\int N(\mathbf{z}; \mu, \Sigma)N(\mathbf{x}; C\mathbf{z}, R) d\mathbf{z}}. \end{aligned}$$

The posterior distribution $f(\mathbf{z}|\mathbf{x})$ has an analytical solution and turns out to be the normal distribution $N(\mathbf{z}; \mu', \Sigma')$ where

$$\begin{aligned} \mu' &= \mu + K(\mathbf{x} - C\mu), \\ \Sigma' &= \Sigma - KC\Sigma, \end{aligned}$$

where

$$K = \Sigma C^T (C\Sigma C^T + R)^{-1}.$$

Properties of the normal density function

Property 1: $f(\mathbf{y}_1)f(\mathbf{y}_2|\mathbf{y}_1) = f(\mathbf{y}_1, \mathbf{y}_2)$

$$N(\mathbf{y}_1; \mu, \Sigma)N(\mathbf{y}_2; B\mathbf{y}_1, R) = N\left(\begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix}; \begin{bmatrix}\mu \\ B\mu\end{bmatrix}, \begin{bmatrix}\Sigma & \Sigma B^T \\ B\Sigma & B\Sigma B^T + R\end{bmatrix}\right)$$

Property 2: marginalization and conditioning

If $\mathbf{y}_1, \mathbf{y}_2$ were jointly normal:

$$f(\mathbf{y}_1, \mathbf{y}_2) = N\left(\begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix}; \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \begin{bmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{bmatrix}\right)$$

then

$$f(\mathbf{y}_1) = N(\mathbf{y}_1; \mu_1, \Sigma_{11})$$

$$f(\mathbf{y}_2) = N(\mathbf{y}_2; \mu_2, \Sigma_{22})$$

$$f(\mathbf{y}_1|\mathbf{y}_2) = N(\mathbf{y}_1; \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{y}_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$$

$$f(\mathbf{y}_2|\mathbf{y}_1) = N(\mathbf{y}_2; \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{y}_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})$$

Kalman filter's derivation

Consider State space model

$$\begin{aligned}\mathbf{z}_t &= A\mathbf{z}_{t-1} + \mathbf{e}_t, \\ \mathbf{x}_t &= C\mathbf{z}_t + \nu_t.\end{aligned}$$

And initial prior on the state \mathbf{z}_1

$$f(\mathbf{z}_1) = N(\mathbf{z}_1; m_0, P_0)$$

We want to derive a recursive algorithm to compute the posterior filtering density

$$f(\mathbf{z}_t | \mathbf{x}_{1:t}).$$

That is, computing the the posterior density of \mathbf{z}_t using the observations up to time t .

Kalman filter's derivation

Assume that we have

$$f(\mathbf{z}_t | \mathbf{x}_{1:t}) = N(\mathbf{z}_t; m_{t|t}, P_{t|t}).$$

The state transition density $f(\mathbf{z}_{t+1} | \mathbf{z}_t)$ and the likelihood function $f(\mathbf{x}_{t+1} | \mathbf{z}_{t+1})$ can be written as

$$\begin{aligned} f(\mathbf{z}_{t+1} | \mathbf{z}_t) &= N(\mathbf{z}_{t+1}; A\mathbf{z}_t, Q), \\ f(\mathbf{x}_{t+1} | \mathbf{z}_{t+1}) &= N(\mathbf{x}_{t+1}; C\mathbf{z}_{t+1}, R). \end{aligned}$$

Therefore, the joint posterior $f(\mathbf{z}_t, \mathbf{z}_{t+1}, \mathbf{x}_{t+1} | \mathbf{x}_{1:t})$ can be written as

$$\begin{aligned} f(\mathbf{z}_t, \mathbf{z}_{t+1}, \mathbf{x}_{t+1} | \mathbf{x}_{1:t}) &= N(\mathbf{z}_t; m_{t|t}, P_{t|t}) \\ &\quad \times N(\mathbf{z}_{t+1}; A\mathbf{z}_t, Q)N(\mathbf{x}_{t+1}; C\mathbf{z}_{t+1}, R), \end{aligned}$$

Kalman filter's derivation

The $f(\mathbf{z}_t, \mathbf{z}_{t+1}, \mathbf{x}_{t+1} | \mathbf{x}_{1:t})$ can be rewritten in matrix form as

$$f(\mathbf{z}_t, \mathbf{z}_{t+1}, \mathbf{x}_{t+1} | \mathbf{x}_{1:t}) = N([\mathbf{z}_t^T, \mathbf{z}_{t+1}^T, \mathbf{x}_{t+1}^T]^T; \mu_t, \Sigma_t),$$

where

$$\mu_t = \begin{bmatrix} \mu_1 \\ \hline \mu_2 \end{bmatrix} = \begin{bmatrix} m_{t|t} \\ \hline Am_{t|t} \\ \hline CAM_{t|t} \end{bmatrix}$$

and

$$\Sigma_t \triangleq \left[\begin{array}{c|c} \Sigma_{11} & \Sigma_{12} \\ \hline \Sigma_{21} & \Sigma_{22} \end{array} \right] = \left[\begin{array}{cc|c} P_{t|t} & P_{t|t}A^T & (P_{t|t}A^T)C^T \\ AP_{t|t} & AP_{t|t}A^T + Q & (AP_{t|t}A^T + Q)^TC^T \\ \hline C(AP_{t|t}) & C(AP_{t|t}A^T + Q) & C(AP_{t|t}A^T + Q)C^T + R \end{array} \right].$$

Kalman filtering algorithm

Prove the Kalman filtering recursion for the following state space model with initial prior on the state $f(\mathbf{z}_1) = N(\mathbf{z}_1; \mathbf{m}_0, \mathbf{P}_0)$

$$\mathbf{z}_t = \mathbf{A}_{t-1} \mathbf{z}_{t-1} + \mathbf{e}_t, \quad \mathbf{e}_t \sim N(0, \mathbf{Q}_t)$$

$$\mathbf{x}_t = \mathbf{C}_t \mathbf{z}_t + \nu_t, \quad \nu_t \sim N(0, \mathbf{R}_t)$$

1: **Inputs:** \mathbf{A}_t , \mathbf{C}_t , \mathbf{Q}_t , \mathbf{R}_t , \mathbf{m}_0 , \mathbf{P}_0 and $\mathbf{x}_{1:T}$.

initialization

2: $\mathbf{m}_{1|0} \leftarrow \mathbf{m}_0$, $\mathbf{P}_{1|0} \leftarrow \mathbf{P}_0$

3: **for** $t = 1$ to T **do**

observation update step

4: $\mathbf{K}_t \leftarrow \mathbf{P}_{t|t-1} \mathbf{C}_t^T (\mathbf{C}_t \mathbf{P}_{t|t-1} \mathbf{C}_t^T + \mathbf{R}_t)^{-1}$

5: $\mathbf{m}_{t|t} \leftarrow \mathbf{m}_{t|t-1} + \mathbf{K}_t (\mathbf{x}_t - \mathbf{C}_t \mathbf{m}_{t|t-1})$

6: $\mathbf{P}_{t|t} \leftarrow (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \mathbf{P}_{t|t-1}$

prediction step

7: $\mathbf{m}_{t+1|t} \leftarrow \mathbf{A}_t \mathbf{m}_{t|t}$

8: $\mathbf{P}_{t+1|t} \leftarrow \mathbf{A}_t \mathbf{P}_{t|t} \mathbf{A}_t^T + \mathbf{Q}_{t+1}$

9: **end for**

10: **Outputs:** $\mathbf{m}_{t|t}$, $\mathbf{P}_{t|t}$ for $t = 1 : T$

Bayesian Smoothing

The purpose of Bayesian smoothing is to compute the marginal posterior distribution of \mathbf{z}_t at time t after receiving observations up to time T where $T > t$:

$$f(\mathbf{z}_t | \mathbf{x}_{1:T})$$

The Rauch-Tung-Striebel smoother (RTS smoother) which is also called the Kalman smoother is used to compute

$$f(\mathbf{z}_t | \mathbf{x}_{1:T}) = N(\mathbf{z}_t; m_{t|T}, P_{t|T})$$

The RTS smoother uses a Kalman filter in its forward path. In its backwards path it updates the densities using the relation

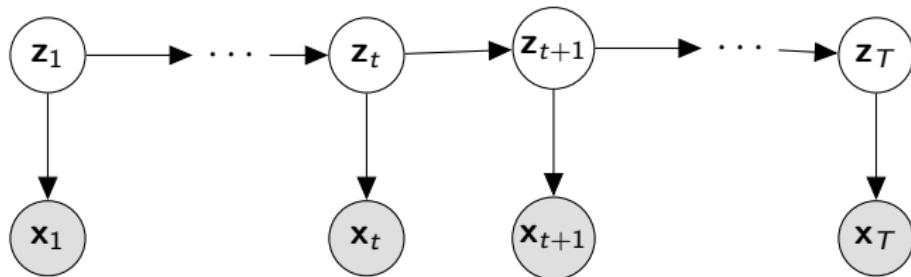
$$\mathbf{z}_t = A_{t-1} \mathbf{z}_{t-1} + e_t$$

RTS Smoother's derivation

Assume $f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T})$ is available as in

$$f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) = N(\mathbf{z}_{t+1}; \mathbf{m}_{t+1|T}, \mathbf{P}_{t+1|T})$$

For example $f(\mathbf{z}_T | \mathbf{x}_{1:T})$ which is the filtering density of \mathbf{z}_T is available after filtering.



The objective is to compute $f(z_t, z_{t+1} | \mathbf{x}_{1:T})$.

RTS Smoother's derivation

The joint posterior $f(\mathbf{z}_t, \mathbf{z}_{t+1} | \mathbf{x}_{1:t})$ can be written as

$$\begin{aligned} f(\mathbf{z}_t, \mathbf{z}_{t+1} | \mathbf{x}_{1:t}) &= N(\mathbf{z}_t; m_{t|t}, P_{t|t}) N(\mathbf{z}_{t+1}; A\mathbf{z}_t, Q) \\ &= N\left(\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t+1} \end{bmatrix}, \begin{bmatrix} m_{t|t} \\ Am_{t|t} \end{bmatrix}, \begin{bmatrix} P_{t|t} & P_{t|t}A^T \\ AP_{t|t} & AP_{t|t}A^T + Q \end{bmatrix}\right) \end{aligned}$$

Using the conditioning property of the multivariate normal distribution $f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t})$ can be computed as a normal density as given in the following:

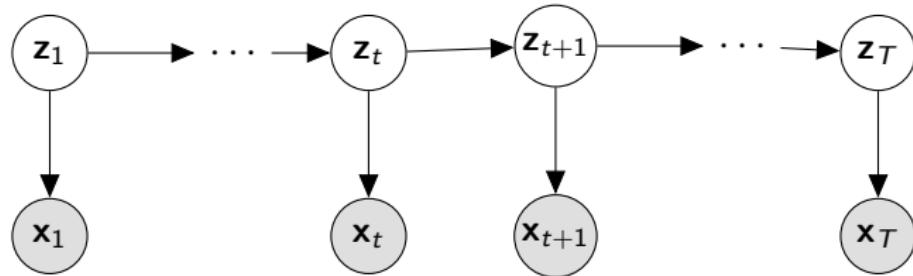
$$f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t}) = N(\mathbf{z}_t; \tilde{m}_t, \tilde{P}_t)$$

where \tilde{m}_t is a function of \mathbf{z}_{t+1} .

RTS Smoother's derivation

Note the Markov property

$$f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:T}) = f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t})$$



Assume $f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T})$ is available as in

$$f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) = N(\mathbf{z}_{t+1}; m_{t+1|T}, P_{t+1|T})$$

Recall that

$$\begin{aligned} f(\mathbf{z}_{t+1}, \mathbf{z}_t | \mathbf{x}_{1:T}) &= f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:T}) \\ &= f(\mathbf{z}_{t+1} | \mathbf{x}_{1:T}) f(\mathbf{z}_t | \mathbf{z}_{t+1}, \mathbf{x}_{1:t}) \\ &= N(\mathbf{z}_{t+1}; m_{t+1|T}, P_{t+1|T}) N(\mathbf{z}_t; \tilde{m}_t, \tilde{P}_t) \end{aligned}$$

RTS Smoother's derivation

where

$$G_t = P_{t|t} A_t^T (A P_{t|t} A^T + Q)^{-1} = P_{t|t} A_t^T P_{t+1|t}^{-1}$$

$$\tilde{m}_t = m_{t|t} + G_t (\mathbf{z}_{t+1} - A m_{t|t})$$

$$\tilde{P}_t = P_{t|t} - G_t (A P_{t|t} A^T + Q) G_t^T = P_{t|t} - G_t P_{t+1|t} G_t^T$$

Hence,

$$\begin{aligned} f(\mathbf{z}_{t+1}, \mathbf{z}_t | \mathbf{x}_{1:T}) &= N(\mathbf{z}_{t+1}; m_{t+1|T}, P_{t+1|T}) N(\mathbf{z}_t; \tilde{m}_t, \tilde{P}_t) \\ &= N\left(\begin{bmatrix} \mathbf{z}_t \\ \mathbf{z}_{t+1} \end{bmatrix}, \begin{bmatrix} \cdot & \cdot \\ m_{t+1|T} & \cdot \\ \cdot & P_{t+1|T} \end{bmatrix}\right) \end{aligned}$$

RTS smoother's backwards recursion

Prove the backwards recursion of the RTS smoother for following state space model with initial prior on the state $f(\mathbf{z}_1) = N(\mathbf{z}_1; \mathbf{m}_0, \mathbf{P}_0)$

$$\mathbf{z}_t = A_{t-1}\mathbf{z}_{t-1} + e_t, \quad e_t \sim N(0, Q_t)$$

$$\mathbf{x}_t = C_t\mathbf{z}_t + \nu_t, \quad \nu_t \sim N(0, R_t)$$

1: **Inputs:** $A_t, Q_t, m_{t|t}, P_{t|t}, m_{t+1|t}, P_{t+1|t}$ for $1 \leq t \leq T$
initialization

2: **for** $t = T-1$ down to 1 **do**

3: $G_t \leftarrow P_{t|t}A_t^T P_{t+1|t}^{-1}$

4: $m_{t|T} \leftarrow m_{t|t} + G_t(m_{t+1|T} - A_t m_{t|t})$

5: $P_{t|T} \leftarrow P_{t|t} + G_t(P_{t+1|T} - P_{t+1|t})G_t^T$

6: **end for**

7: **Outputs:** $m_{t|T}, P_{t|T}$

Read home

- Shumway and Stoffer, Chapters 6.1 and 6.2

Time Series Analysis

Teaching Session II: ARIMA models-3

Seasonal models

Tohid Ardesthiri

Linköping University
Division of Statistics and Machine Learning

September 18, 2019



Seasonal ARMA

- Seasonal patterns
 - ▶ Yearly (ocean temperature)
 - ▶ Daily, weekly (Server workload)
- Strong correlation of x_t and x_{t+s}
 - ▶ $s = 12, 24, \dots$
- Applications
 - ▶ Physics, biology, economics, computer science

Seasonal ARMA

- Pure seasonal $ARMA(P, Q)_s$

$$\Phi_P(B^s)x_t = \theta_Q(B^s)w_t$$

- Seasonal autoregressive operator

$$\Phi_P(B^s) = 1 - \Phi_1(B^{(1\cdot s)}) - \dots - \Phi_P B^{P\cdot s}$$

- Seasonal moving average operator

$$\Theta_Q(B^s) = 1 + \Theta_1(B^{(1\cdot s)}) + \dots + \Theta_Q B^{Q\cdot s}$$

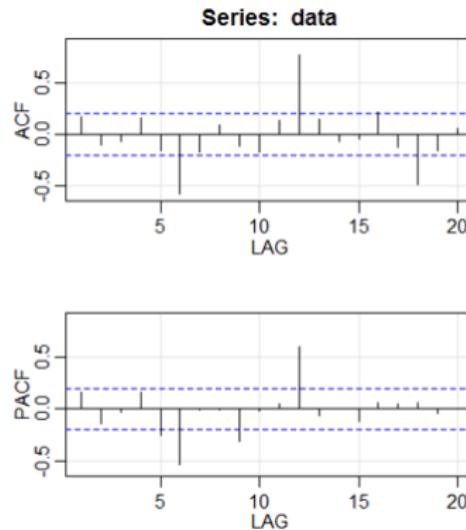
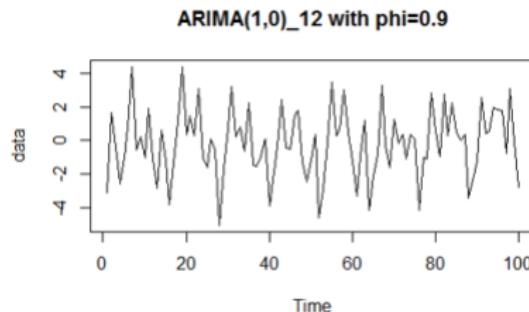
- Same principles for causality and invertibility

- Example: $ARMA(1, 0)_{12}$ and $ARMA(0, 1)_{12}$

- ▶ Autocovariance

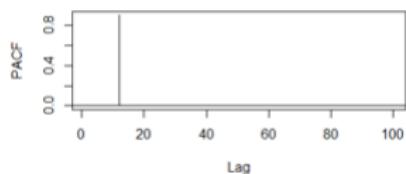
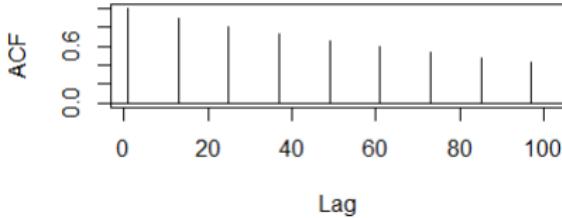
Seasonal ARMA

- Example: Simulated $ARMA(1, 0)_{12}$, $\Phi = 0.9$



Seasonal ARMA

- **Example:** Simulated $ARMA(1, 0)_{12}$, $\Phi = 0.9$
 - ▶ Theoretical ones



Seasonal ARMA

	$\text{AR}(P)_s$	$\text{MA}(Q)_s$	$\text{ARMA}(P, Q)_s$
ACF*	Tails off at lags ks , $k = 1, 2, \dots,$	Cuts off after lag Qs	Tails off at lags ks
PACF*	Cuts off after lag Ps	Tails off at lags ks $k = 1, 2, \dots,$	Tails off at lags ks

*The values at nonseasonal lags $h \neq ks$, for $k = 1, 2, \dots$, are zero.

Multiplicative seasonal ARMA

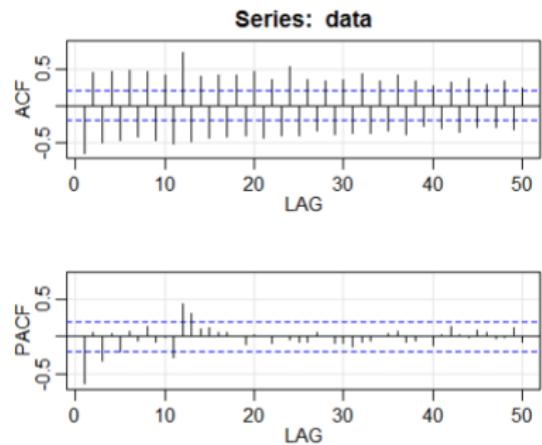
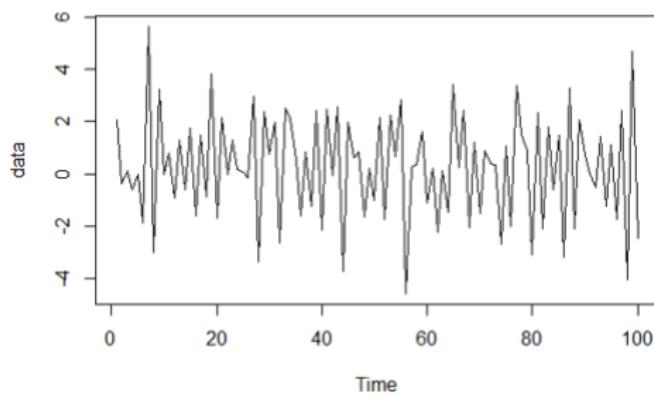
- **Problem:** in real data, it is hard to assume x_t is dependent on x_{t-kh} only...
 - ▶ Combinal seasonal and nonseasonal!
- Multiplicative Seasonal ARMA(p, q) \times (P, Q)_s

$$\Phi_P(B^s)\phi(B)x_t = \Theta_Q(B^s)\theta(B)w_t$$

- **Example** Expression for ARMA(1, 1) \times (1, 0)_s

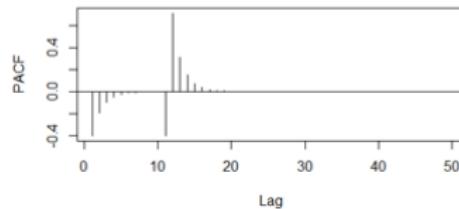
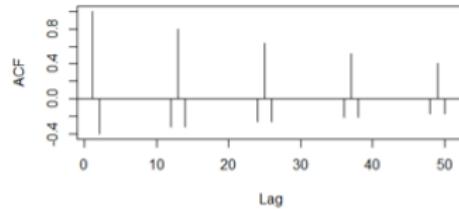
Multiplicative seasonal ARMA

- Example $x_t = 0.8x_{t-12} + w_t - 0.5w_{t-1}$



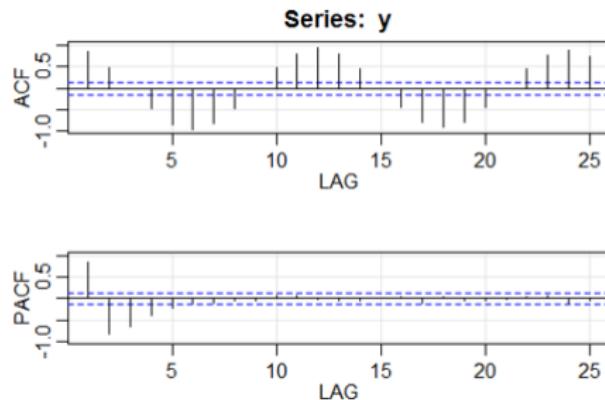
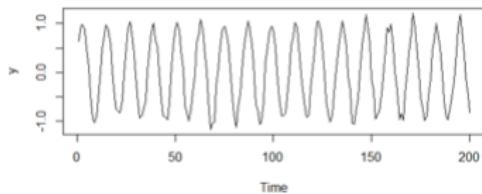
Multiplicative seasonal ARMA

- Example $x_t = 0.8x_{t-12} + w_t - 0.5w_{t-1}$
 - ▶ Theoretical



SARIMA

- What if there is a seasonal pattern which differs a little between the series



Note: ACF almost decays very slowly at peaks 12h

SARIMA

- Multiplicative seasonal autoregressive integrated moving average model $ARIMA(p, d, q) \times (P, D, Q)_s$

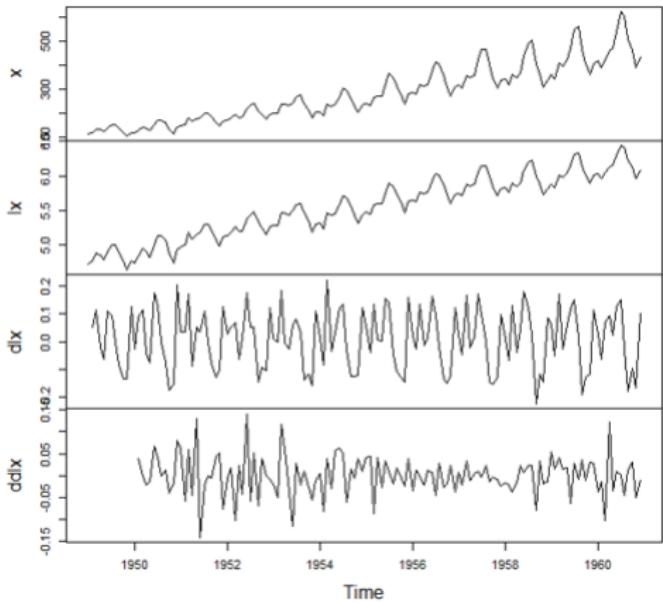
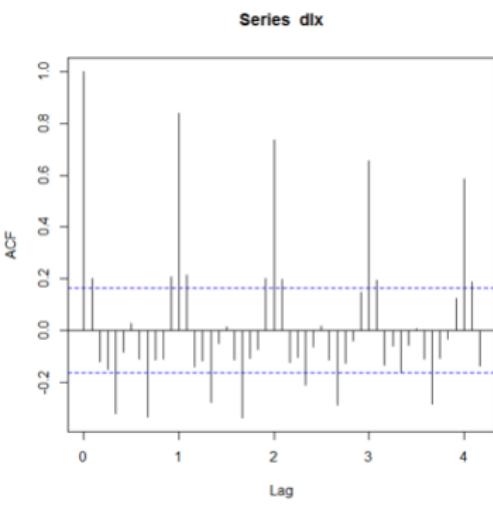
$$\Phi_p(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \delta + \Theta_Q(B^s)\theta(B)w_t$$

$$\nabla_s^D = (1 - B^s)^D$$

- How to identify SARIMA?
 - ① Perform differencing first (trend)
 - ② Investigate ACF → slowly decays at peaks?
 - ① Yes → Additional differencing by ∇_s^D
 - ③ Model non-seasonal part
 - ④ Model seasonal part (check peaks), check ACF and PACF of residuals

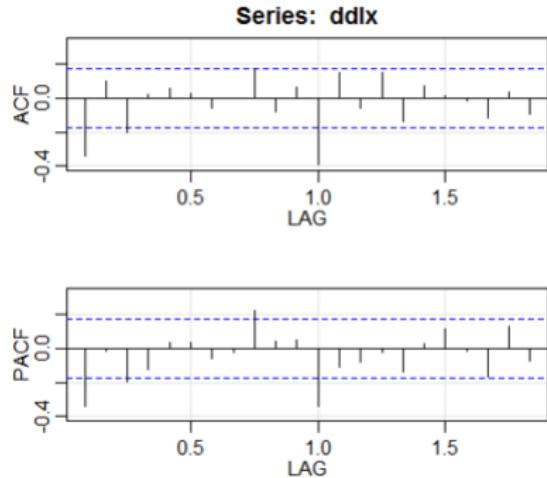
SARIMA

- Example: Air passengers



SARIMA

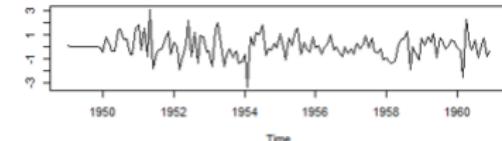
- Example: Air passengers



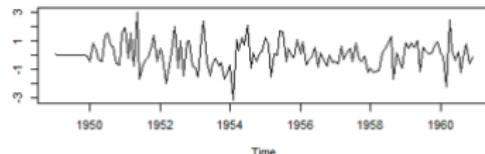
$(0, 1, 1)_{12}$ or $(1, 1, 0)_{12}$

SARIMA

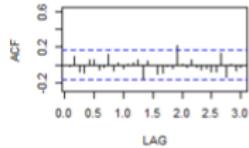
Model: (1,1,1) (0,1,1) Standardized Residuals



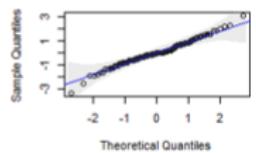
Model: (1,1,1) (1,1,0) Standardized Residuals



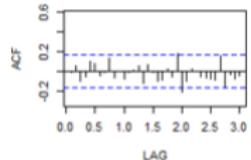
ACF of Residuals



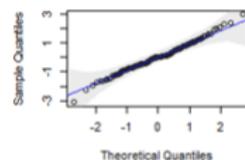
Normal Q-Q Plot of Std Residuals



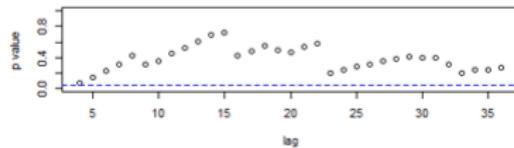
ACF of Residuals



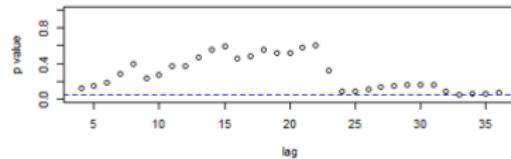
Normal Q-Q Plot of Std Residuals



p values for Ljung-Box statistic



p values for Ljung-Box statistic



SARIMA

- Remove AR term!

Is one model much better than the other one?

```
> m1$fit
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
REPORT = 1, reltol = tol))

Coefficients:
            ar1      ma1      sar1
0.0547   -0.4886  -0.4731
s.e.  0.2161    0.1933   0.0800

sigma2 estimated as 0.001425:  log likelihood = 241.73,  aic = -475.47
> m2$fit
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
Q), period = S), include.mean = !no.constant, optim.control = list(trace = trc,
REPORT = 1, reltol = tol))

Coefficients:
            ar1      ma1      sma1
0.1960   -0.5784  -0.5643
s.e.  0.2475    0.2132   0.0747

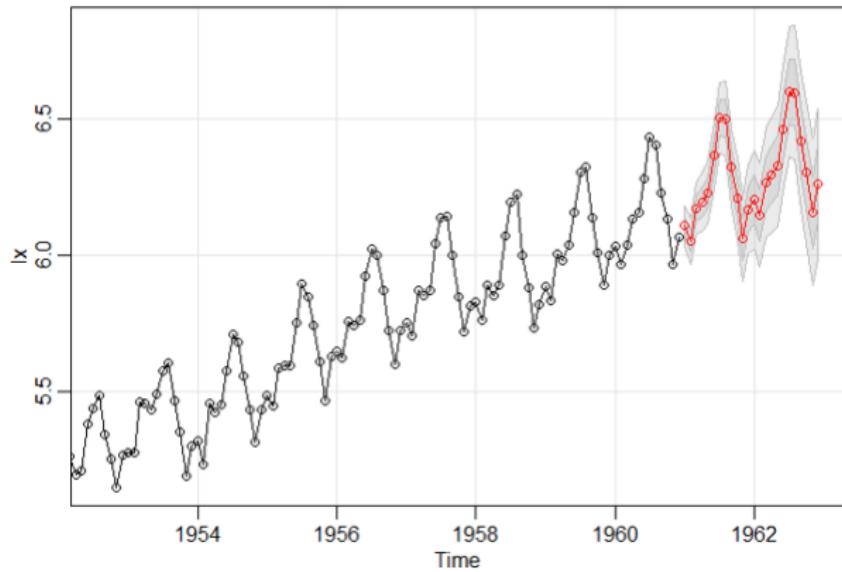
sigma^2 estimated as 0.001341:  log likelihood = 244.95,  aic = -481.9
```

$(1, 1, 1) \times (1, 1, 0)_{12}$

$(1, 1, 1) \times (0, 1, 1)_{12}$

SARIMA

- Forecasting



Read home

- Shumway and Stoffer, section 3.9
- R code: sarima, sarima.for, runs