MATH 8090: Seasonal Time Series Models

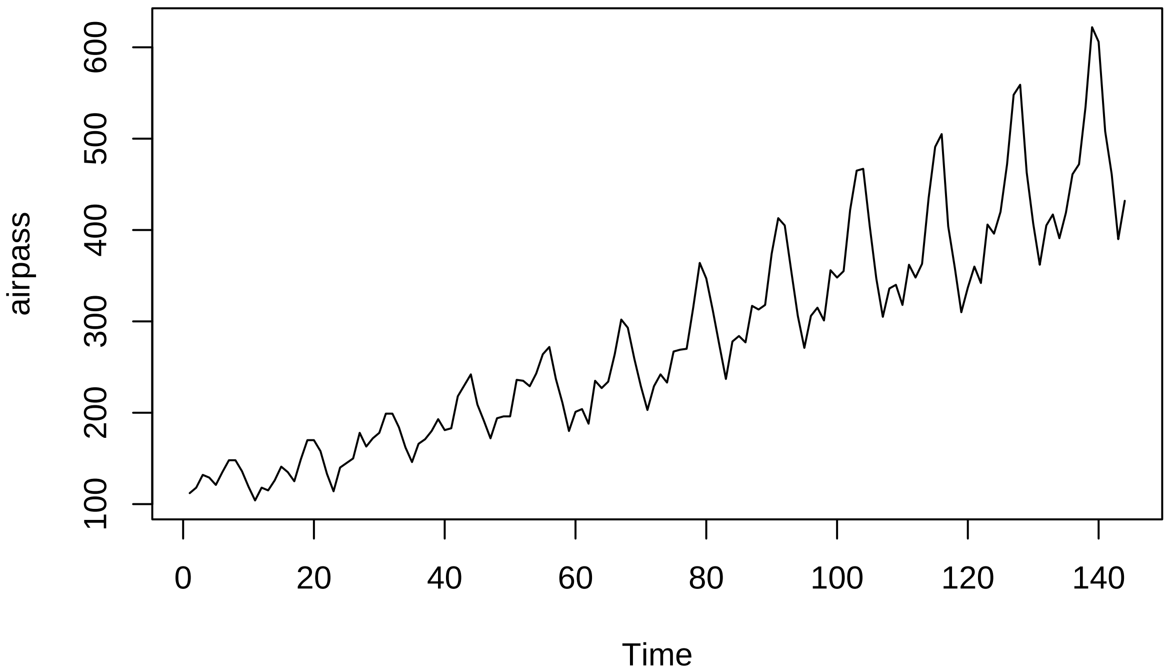
Whitney Huang, Clemson University

10/5-10/7/2021

Contents

Read the data

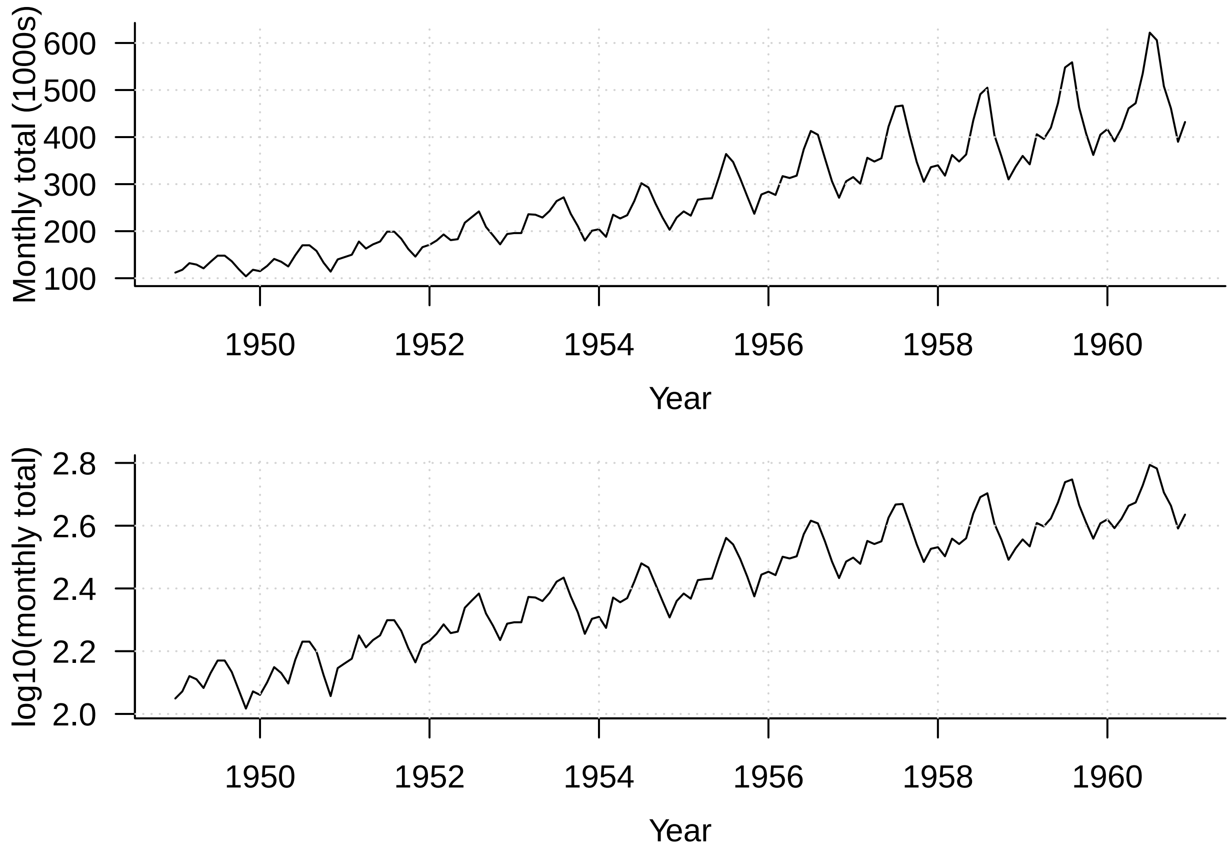
```
url <- "http://www.stat.osu.edu/~pfc/teaching/6550/datasets/airpass.txt"
airpass <- read.table(url)
ts.plot(airpass)
```




```
## create the time variable.
yr <- seq(1949, 1960 + 11 / 12, by = 1 / 12)
```

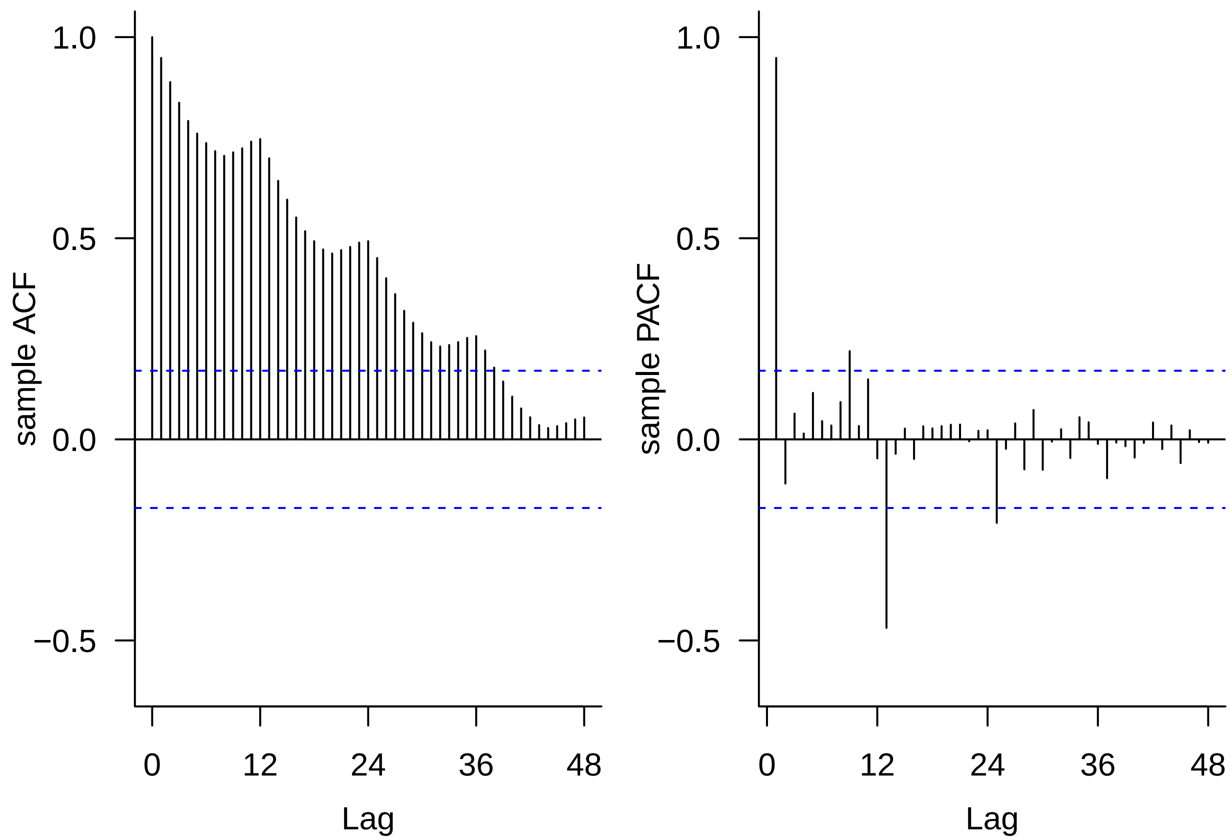
Plot the time series

```
par(bty = "L", mar = c(3.6, 3.5, 0.8, 0.6), mgp = c(2.4, 1, 0), las = 1, mfrow = c(2, 1))
## plot the time series.
plot(yr, airpass$V1, type = "l", xlab = "Year",
     ylab = "Monthly total (1000s)")
grid()
## take a log (to the base 10) of the air passenger data.
log.airpass <- log10(airpass)
plot(yr, log.airpass$V1, type = "l", xlab = "Year", ylab = "log10(monthly total)")
grid()
```



Plot sample ACF/PACF

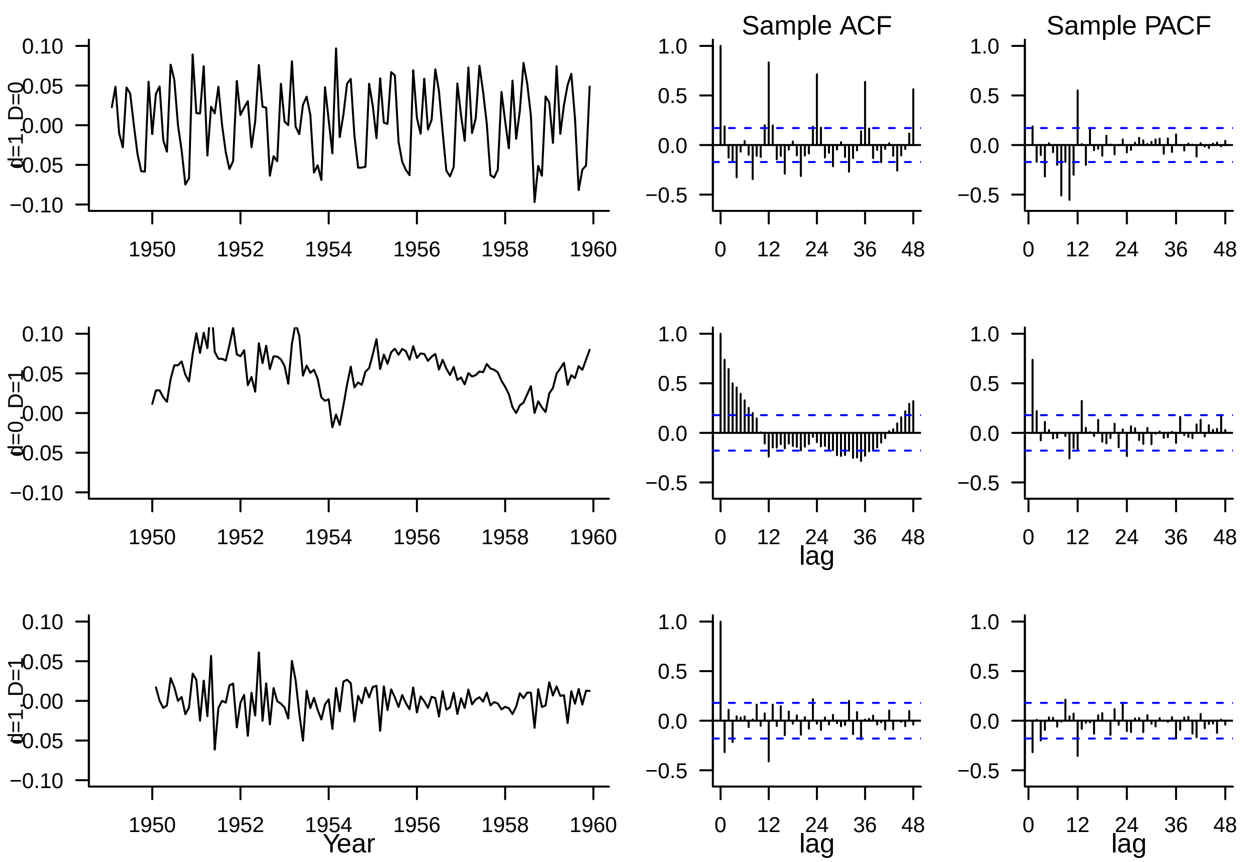
```
log.shortair <- log.airpass$V1[1:132]
shortyears <- yr[1:132]
par(bty = "L", mar = c(3.6, 3.5, 0.8, 0.6), mgp = c(2.4, 1, 0), las = 1, mfrow = c(1, 2))
acf(log.shortair, ylab = "sample ACF", main = "", lag.max = 48, ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
pacf(log.shortair, ylab = "sample PACF", main = "", lag.max = 48, ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
```



Trying Different Orders of Differencing

```
## take the differences Y_t = (1-B) X_t
diff.1.0 <- diff(log.shortair)
## take the seasonal differences Y_t =(1-B^(12)) X_t
diff.0.1 <- diff(log.shortair, lag = 12, diff = 1)
## take the differences Y_t = (1-B^(12)) (1-B) X_t
diff.1.1 <- diff(diff(log.shortair, lag = 12, diff = 1))

par(bty = "L", mar = c(3.6, 3.5, 1, 0.6), mgp = c(2.4, 1, 0), las = 1)
layout.matrix <- matrix(c(1, 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 9), nrow = 3, ncol = 4, byrow = T)
layout(mat = layout.matrix)
plot(shortyears[-1], diff.1.0, xlab = "", ylab = "d=1, D=0",
     type = "l", ylim = c(-0.1, 0.1), xlim = range(shortyears))

acf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("Sample ACF", side = 3, line = 0, cex = 0.8)

pacf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("Sample PACF", side = 3, line = 0, cex = 0.8)


plot(shortyears[-c(1:12)], diff.0.1, xlab = "", ylab = "d=0, D=1",
```

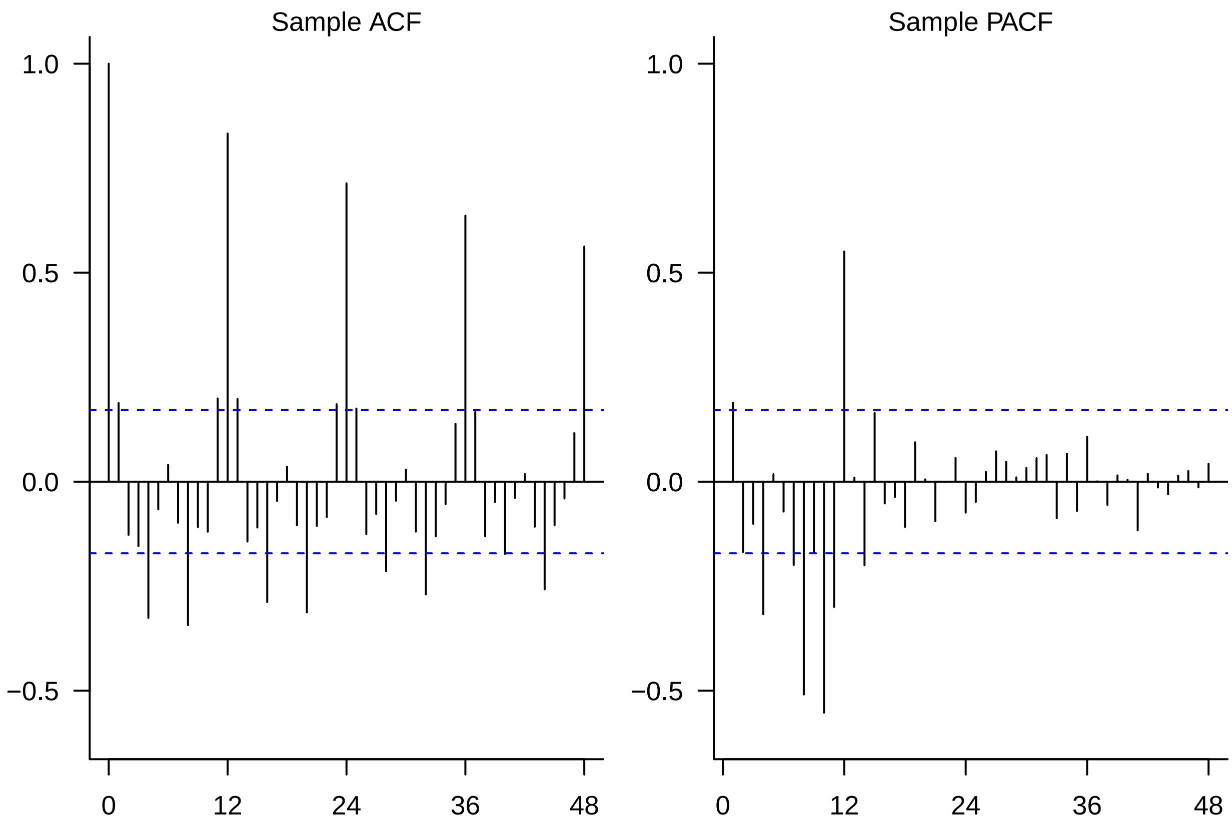
```
     type = "l", ylim = c(-0.1, 0.1), xlim = range(shortyears))

acf(diff.0.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("lag", side = 1, line = 1.8, cex = 0.8)

pacf(diff.0.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))

plot(shortyears[-c(1:13)], diff.1.1, xlab = "", ylab = "d=1, D=1",
     type = "l", ylim = c(-0.1, 0.1), xlim = range(shortyears))
mtext("Year", side = 1, line = 1.8, cex = 0.8)

acf(diff.1.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("lag", side = 1, line = 1.8, cex = 0.8)

pacf(diff.1.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("lag", side = 1, line = 1.8, cex = 0.8)
```



Show the ACF and PACF for the d=1, D=0 case.

```
par(mfrow = c(1, 2), cex = 0.8, bty = "L", mar = c(3.6, 3, 1, 0.6), mgp = c(2.4, 1, 0), las = 1)
acf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
```

```r
axis(side = 1, at = seq(0, 48, 12))
mtext("Sample ACF", side = 3, cex = 0.8)

pacf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("Sample PACF", side = 3, cex = 0.8)
```



A useful function for the model diagnostics (courtesy of Peter Craigmile at OSU)

```r
plot.residuals <- function (x, y = NULL, lag.max = NULL, mean.line = TRUE,
                acf.ylim = c(-0.25, 1), mfrow = c(2, 2),
                            lags = NULL, ...){
  if (!is.null(mfrow))
    par(mfrow = mfrow)
  if (is.null(y)){
    y <- x
    x <- seq(length(y))
  } else {
    x <- as.numeric(x)
    y <- as.numeric(y)
  }

  if (is.null(lag.max)) {
    lag.max <-  floor(10 * log10(length(x)))
  }
  plot(x, y, type = "l", ...)
```

```r
  if (mean.line) abline(h = 0, lty = 2)
  qqnorm(y, main = "", las = 1); qqline(y)
  if (is.null(lags)) {
    acf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
        ylab = "sample ACF", las = 1)

    pacf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
         ylab = "sample PACF", las = 1)
  }
  else {
    acf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
        ylab = "sample ACF", xaxt = "n", las = 1)
    axis(side = 1, at = lags)

    pacf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
         ylab = "sample PACF", xaxt = "n", las = 1)
    axis(side = 1, at = lags)
  }
  Box.test(y, lag.max, type = "Ljung-Box")
}
```

Fitting the SARIMA$(1,1,0) \times (1,0,0)$ model

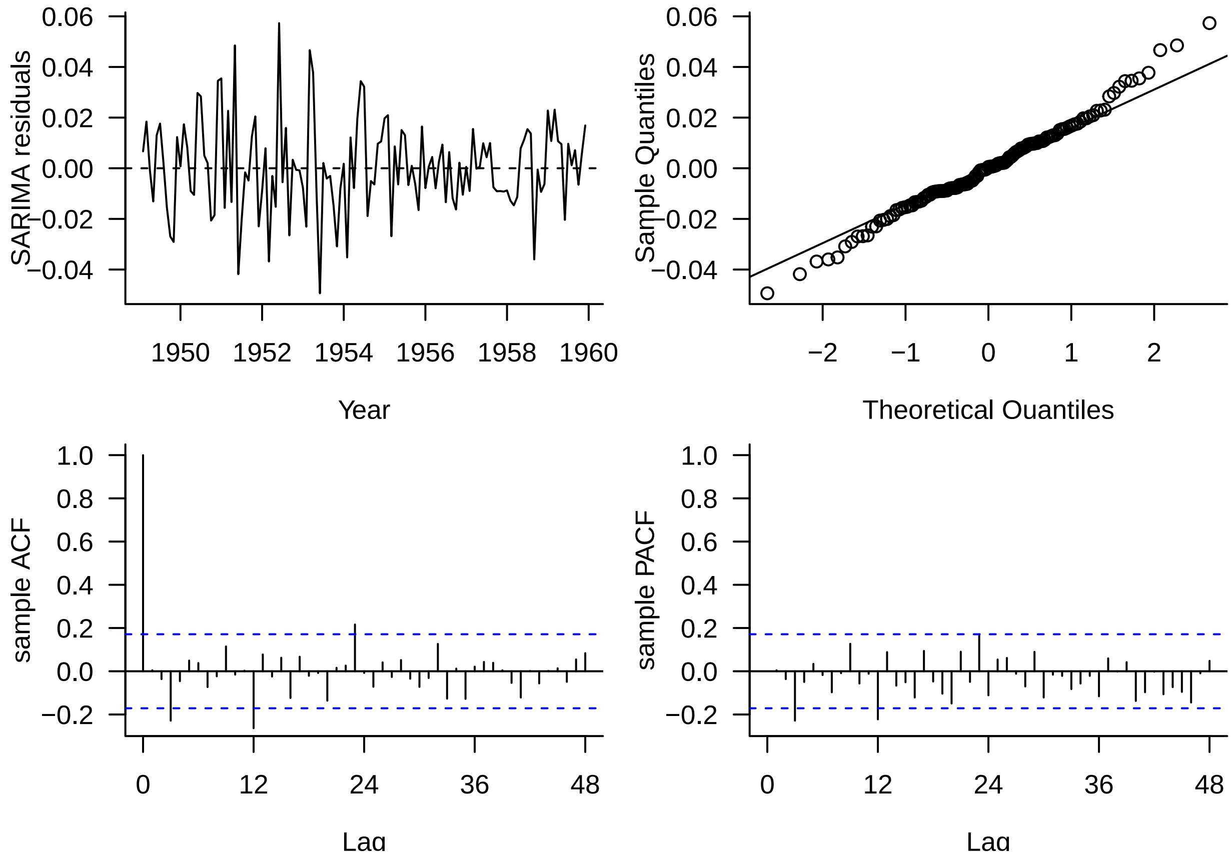
```r
fit1 <- arima(diff.1.0, order = c(1, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12))


fit1
```

```
##
## Call:
## arima(x = diff.1.0, order = c(1, 0, 0), seasonal = list(order = c(1, 0, 0),
##     period = 12))
##
## Coefficients:
##           ar1    sar1  intercept
##       -0.2667  0.9291     0.0039
## s.e.   0.0865  0.0235     0.0096
##
## sigma^2 estimated as 0.0003298:  log likelihood = 327.27,  aic = -646.54
```

```r
Box.test(fit1$residuals, lag = 48, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  fit1$residuals
## X-squared = 55.372, df = 48, p-value = 0.2164
```

```r
par(mfrow = c(2, 2), cex = 0.8, bty = "L", mar = c(3.6, 4, 0.8, 0.6),
    mgp = c(2.8, 1, 0), las = 1)
plot.residuals(shortyears[-1], resid(fit1), lag.max = 48,
               ylab = "SARIMA residuals", xlab = "Year", lags = seq(0, 48, 12))
```




```
##
##  Box-Ljung test
##
## data:  y
## X-squared = 55.372, df = 48, p-value = 0.2164
```

Fitting the SARIMA$(0, 1, 0) \times (1, 0, 0)$ model

```r
(fit2 <- arima(diff.1.0, seasonal = list(order = c(1, 0, 0), period = 12)))
```

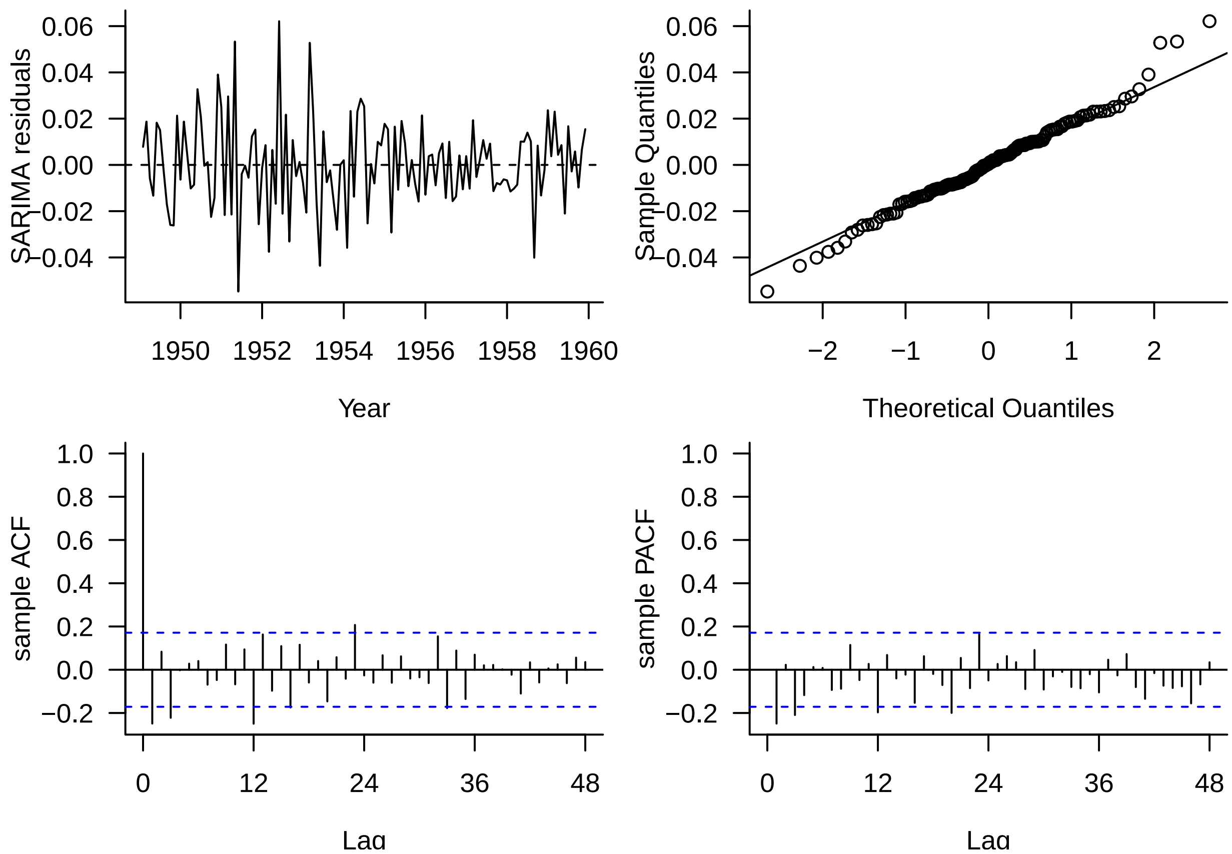
```
##
## Call:
## arima(x = diff.1.0, seasonal = list(order = c(1, 0, 0), period = 12))
##
## Coefficients:
##         sar1  intercept
##       0.9081     0.0040
## s.e.  0.0278     0.0108
##
## sigma^2 estimated as 0.0003616:  log likelihood = 322.75,  aic = -639.51
```

```r
Box.test(fit2$residuals, lag = 48, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
```

```
## data:  fit2$residuals
## X-squared = 80.641, df = 48, p-value = 0.002209
```

```r
par(mfrow = c(2, 2), cex = 0.8, bty = "L", mar = c(3.6, 4, 0.8, 0.6),
    mgp = c(2.8, 1, 0), las = 1)
plot.residuals(shortyears[-1], resid(fit2), lag.max = 48,
               ylab = "SARIMA residuals", xlab = "Year", lags = seq(0, 48, 12))
```




```
##
##  Box-Ljung test
##
## data:  y
## X-squared = 80.641, df = 48, p-value = 0.002209
```

Forecasting 1960 Data

```r
## fit the first full model
fit1 <- arima(log.shortair, order = c(1, 1, 0),
                     seasonal = list(order = c(1, 0, 0), period = 12))
fit1
```

```
##
## Call:
## arima(x = log.shortair, order = c(1, 1, 0), seasonal = list(order = c(1, 0,
##     0), period = 12))
##
## Coefficients:
```

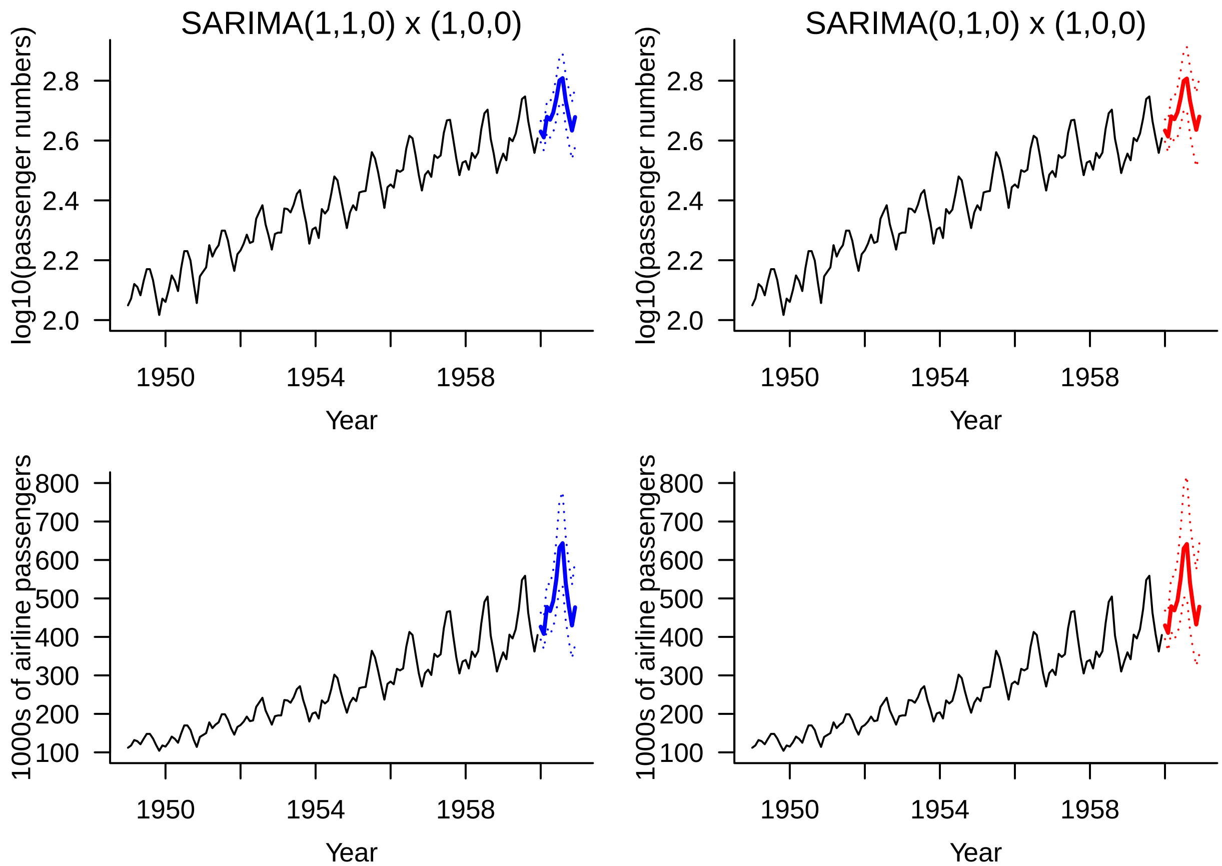
```
##           ar1    sar1
##       -0.2665  0.9298
## s.e.   0.0866  0.0233
##
## sigma^2 estimated as 0.0003299:  log likelihood = 327.19,  aic = -648.38
```

```r
## fit the second full model
fit2 <- arima(log.shortair, order = c(0, 1, 0),
                     seasonal = list(order = c(1, 0, 0), period = 12))
fit2
```

```
##
## Call:
## arima(x = log.shortair, order = c(0, 1, 0), seasonal = list(order = c(1, 0,
##     0), period = 12))
##
## Coefficients:
##         sar1
##       0.9088
## s.e.  0.0276
##
## sigma^2 estimated as 0.0003617:  log likelihood = 322.69,  aic = -641.38
```

```r
## define the forecasting time points
fyears <- yr[133:144]

preds1 <- predict(fit1, 12)
forecast1 <- preds1$pred
flimits1  <- qnorm(0.975) * preds1$se

preds2 <- predict(fit2, 12)
forecast2 <- preds2$pred
flimits2  <- qnorm(0.975) * preds2$se


par(mfrow = c(2, 2), cex = 0.8, bty = "L", mar = c(3.6, 4, 1, 0.6),
    mgp = c(2.4, 1, 0), las = 1)

plot(shortyears, log.shortair, type = "l", xlab = "Year",
     ylab = "log10(passenger numbers)", xlim = range(yr), ylim = c(2, 2.9))
mtext("SARIMA(1,1,0) x (1,0,0)")
## plots the forecasts
lines(fyears, forecast1, lwd = 2, col = "blue")
## plot the 95% prediction intervals.
lines(fyears, forecast1 + flimits1, lty = 3, col = "blue")
lines(fyears, forecast1 - flimits1, lty = 3, col = "blue")

plot(shortyears, log.shortair, type = "l", xlab = "Year",
     ylab = "log10(passenger numbers)", xlim = range(yr), ylim = c(2, 2.9))
mtext("SARIMA(0,1,0) x (1,0,0)")
## plots the forecasts
lines(fyears, forecast2, lwd = 2, col = "red")
## plot the 95% prediction intervals.
lines(fyears, forecast2 + flimits2, lty = 3, col = "red")
lines(fyears, forecast2 - flimits2, lty = 3, col = "red")
```

```r
plot(shortyears, 10^log.shortair, type = "l", xlab = "Year",
     ylab="1000s of airline passengers", xlim = range(yr), ylim = c(100, 800))
lines(fyears, 10^forecast1, lwd = 2, col = "blue")
lines(fyears, 10^(forecast1 + flimits1), lty = 3, col = "blue")
lines(fyears, 10^(forecast1 - flimits1), lty = 3, col = "blue")

plot(shortyears, 10^log.shortair, type = "l", xlab = "Year",
     ylab="1000s of airline passengers", xlim = range(yr), ylim = c(100, 800))
lines(fyears, 10^forecast2, lwd = 2, col = "red")
lines(fyears, 10^(forecast2 + flimits2), lty = 3, col = "red")
lines(fyears, 10^(forecast2 - flimits2), lty = 3, col = "red")
```



Evaluating Forecast Performance

```r
## calculate the root mean square error (RMSE)
sqrt(mean((10^forecast1 - 10^log.airpass$V1[133:144])^2))
```

```
## [1] 30.36384
```

```r
sqrt(mean((10^forecast2 - 10^log.airpass$V1[133:144])^2))
```

```
## [1] 31.32376
```

```r
## calculate the mean relative prediction error.
mean((10^forecast1 - 10^log.airpass$V1[133:144]) / 10^log.airpass$V1[133:144])
```

```
## [1] 0.05671086
```

```r
mean((10^forecast2 - 10^log.airpass$V1[133:144]) / 10^log.airpass$V1[133:144])
```

```
## [1] 0.05951677
```

```r
## calculate the empirical coverage rate
CI_fit1 <- cbind(as.numeric(10^(forecast1 + flimits1)),
                 as.numeric(10^(forecast1 - flimits1)))
sum(CI_fit1 - 10^log.airpass$V1[133:144] < 0) / length(10^log.airpass$V1[133:144])
```

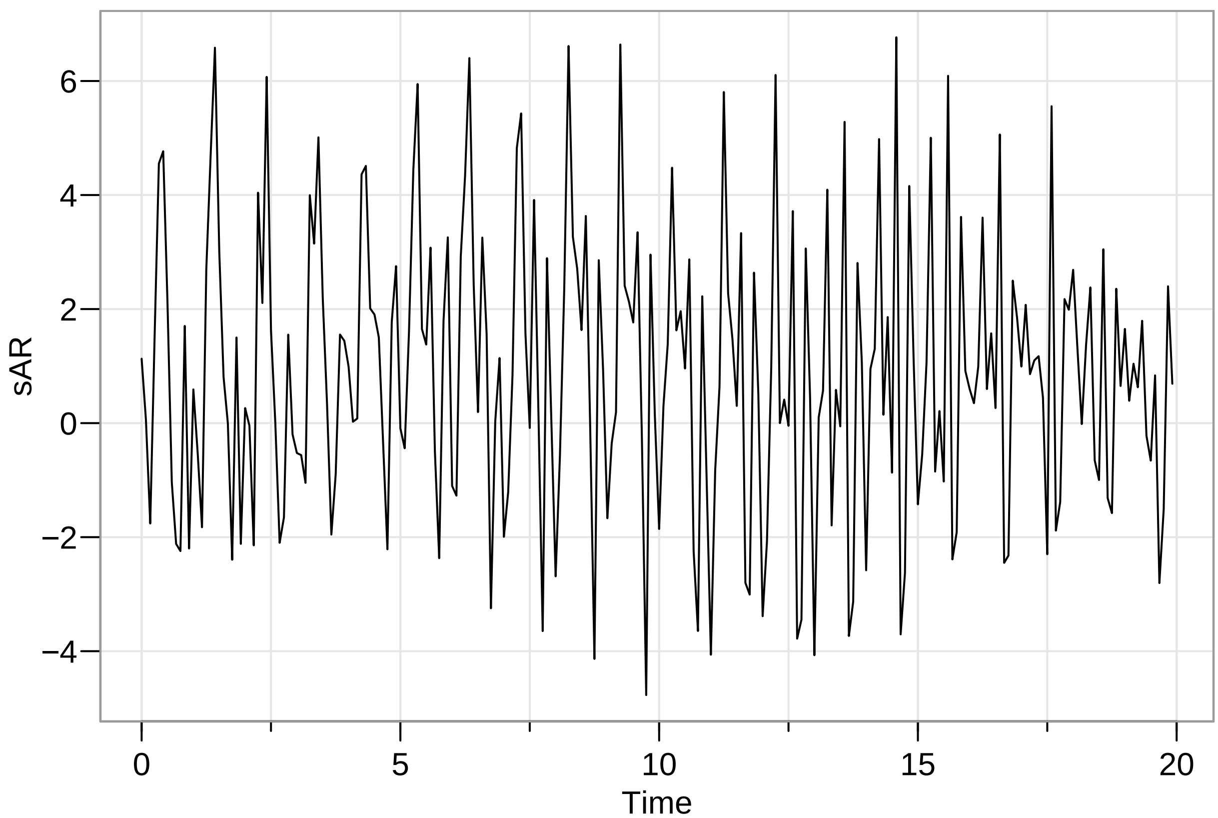
```
## [1] 0.9166667
```

```r
CI_fit2 <- cbind(as.numeric(10^(forecast2 + flimits2)),
                 as.numeric(10^(forecast2 - flimits2)))
sum(CI_fit2 - 10^log.airpass$V1[133:144] < 0) / length(10^log.airpass$V1[133:144])
```

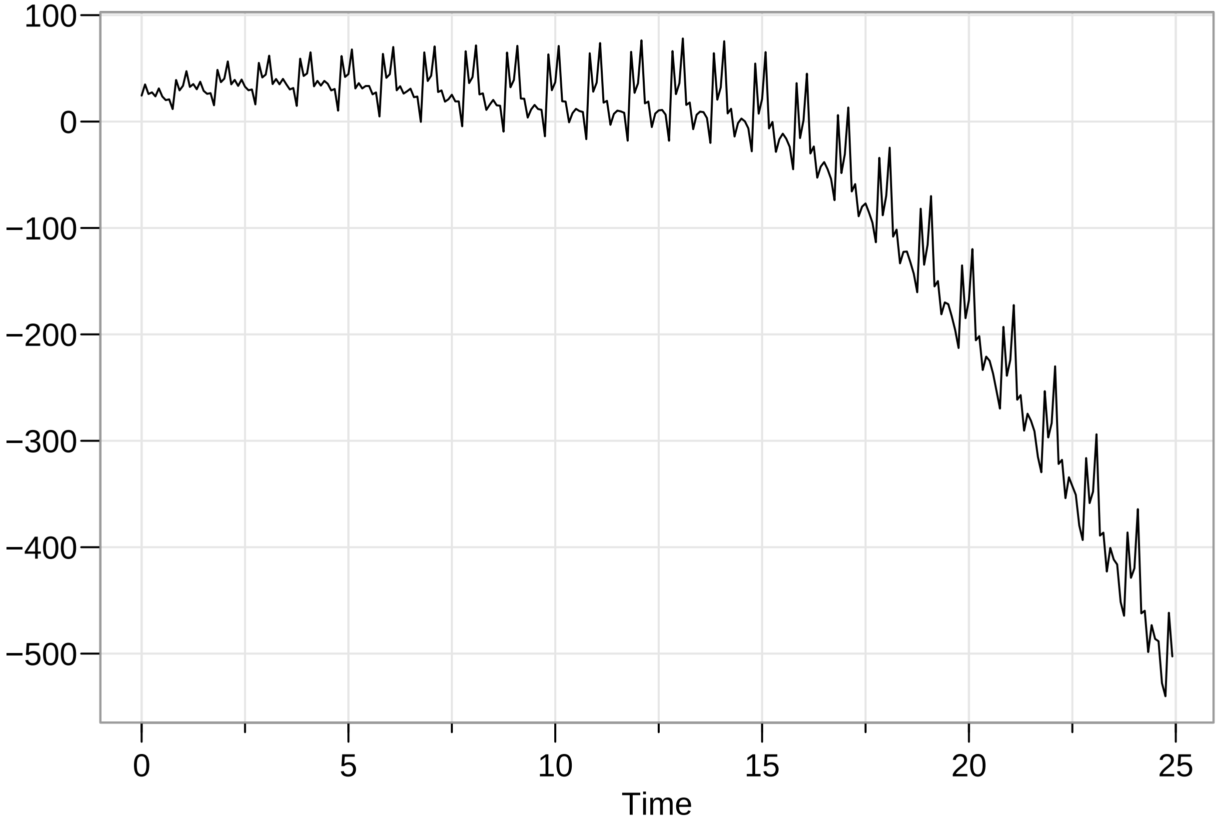
```
## [1] 1
```

SARIMA simulation

```r
library(astsa)
par(las = 1)
sAR = sarima.sim(sar = .9, S = 12, n = 240)
tsplot(sAR)
```

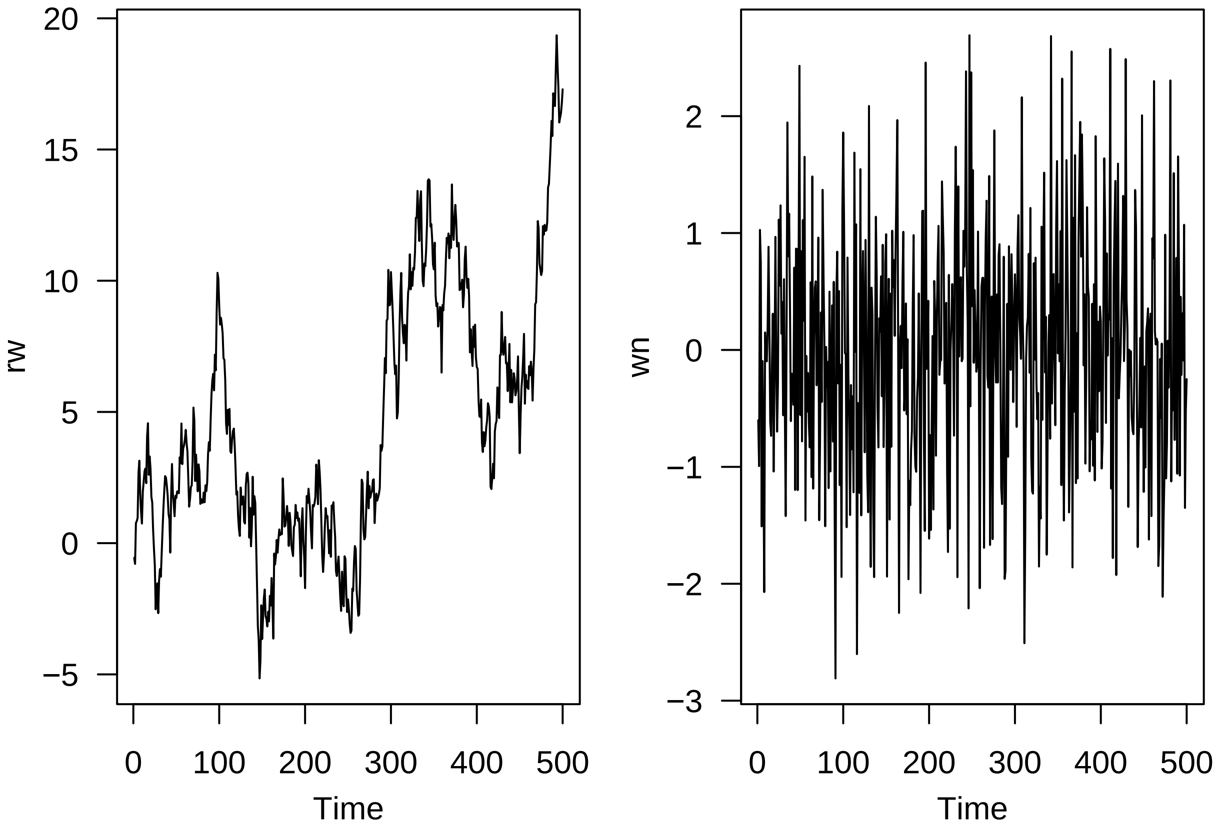



```r
tsplot(sarima.sim(d = 1, ar = -.4, D = 1, sar = .9, S = 12, n = 300),
       ylab = "")
```



Unit root test examples

```r
set.seed(123)
rw <- cumsum(rnorm(500))
wn <- rnorm(500)
par(las = 1, mgp = c(2.2, 1, 0), mar = c(3.6, 3.6, 0.8, 0.6), mfrow = c(1, 2))
ts.plot(rw)
ts.plot(wn)
```




```r
diff.rw <- diff(rw); n <- length(rw)
ys <- diff.rw; xs <- rw[1:(n-1)]
ols.rw <- lm(ys ~ xs); summary(ols.rw)
```

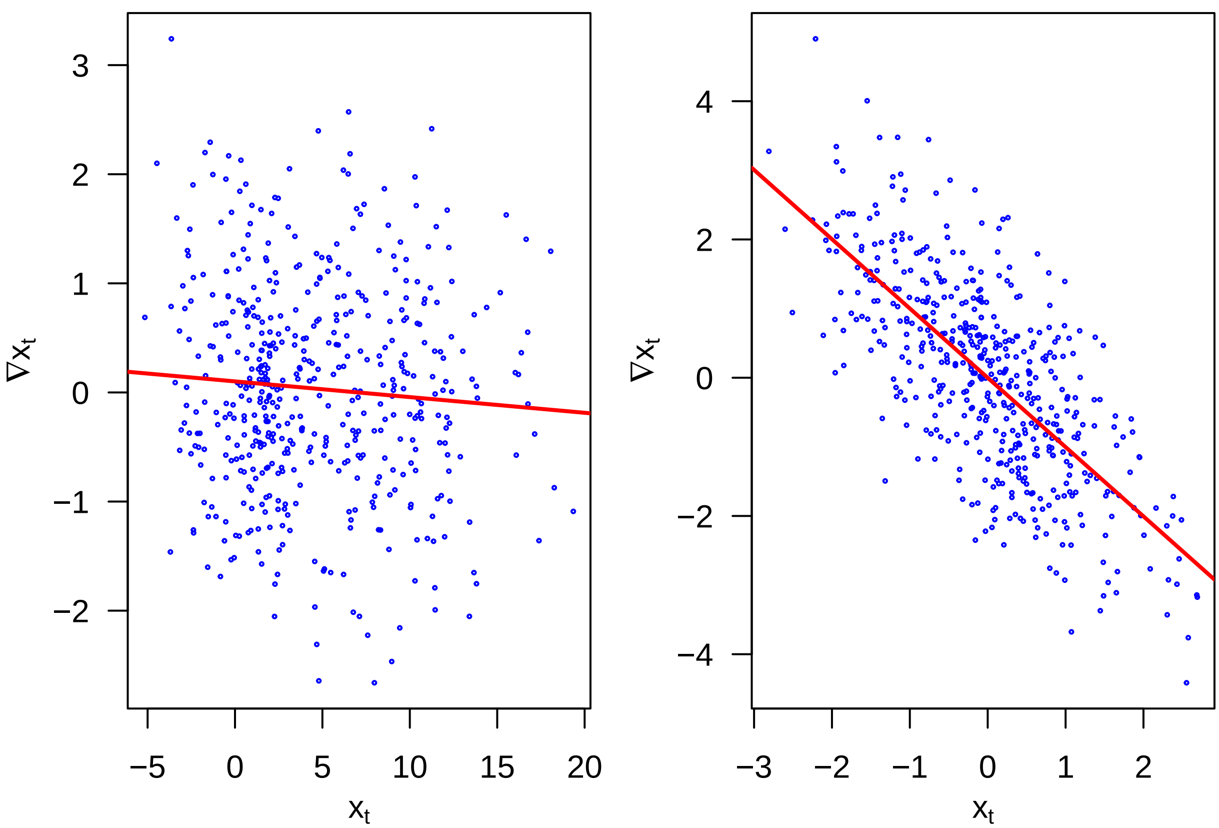
```
##
## Call:
## lm(formula = ys ~ xs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67541 -0.62862 -0.01118  0.63805  3.08747
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.10125    0.05973   1.695   0.0906 .
## xs          -0.01438    0.00899  -1.600   0.1102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9719 on 497 degrees of freedom
## Multiple R-squared:  0.005124,   Adjusted R-squared:  0.003123
## F-statistic:  2.56 on 1 and 497 DF,  p-value: 0.1102
```

```r
diff.wn <- diff(wn)
ys <- diff.wn; xs <- wn[1:(n-1)]
ols.wn <- lm(ys ~ xs); summary(ols.wn)
```

```
##
## Call:
```

```
## lm(formula = ys ~ xs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.81182 -0.69065  0.00075  0.64461  2.68750
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.001138   0.045329  -0.025     0.98
## xs          -1.002420   0.044843 -22.354   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.013 on 497 degrees of freedom
## Multiple R-squared:  0.5014, Adjusted R-squared:  0.5004
## F-statistic: 499.7 on 1 and 497 DF,  p-value: < 2.2e-16
```

```r
par(las = 1, mgp = c(2.2, 1, 0), mar = c(3.6, 3.6, 0.8, 0.6), mfrow = c(1, 2))
plot(rw[1:length(diff.rw)], diff.rw, xlab = expression(x[t]),
     ylab = expression(paste(nabla, x[t])), cex = 0.25, col = "blue")
abline(ols.rw, col = "red", lwd = 2)
plot(wn[1:length(diff.wn)], diff.wn, xlab = expression(x[t]),
     ylab = expression(paste(nabla, x[t])), cex = 0.25, col = "blue")
abline(ols.wn, col = "red", lwd = 2)
```



References