A Solution Manual and Notes for: *Statistics and Data Analysis for Financial Engineering* by David Ruppert

John L. Weatherwax*

November 4, 2014

*wax@alum.mit.edu

To my family.

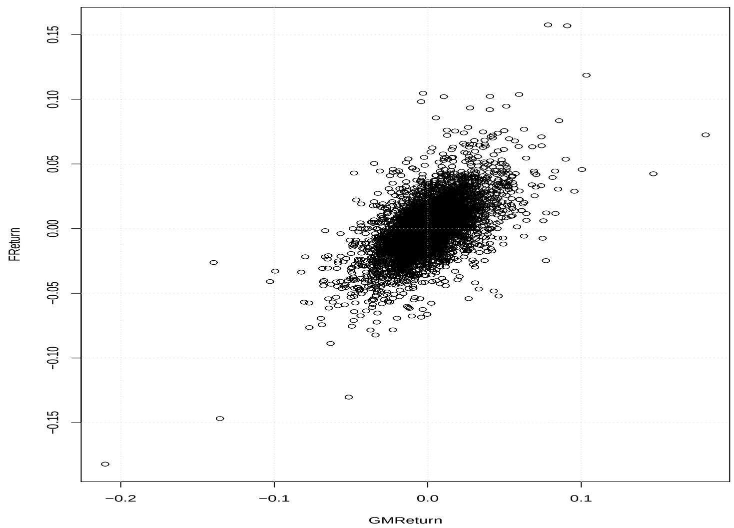

Figure 1: A scatter plot of the returns of Ford as a function of the returns of GM.

Chapter 2 (Returns)

R Lab

See the R script `Rlab.R` where the problem for this chapter are worked.

Problem 1 (GM and Ford returns)

In Figure 1 we show the scatter plot of the returns of Ford as a function of the returns of GM. We notice that these returns do appear to be correlated (they are distributed somewhat symmetrically about a line) and the outliers of each stocks return do appear together.

Problem 2

In the accompanying R code we plot the two returns. The two sets of points lie almost exactly on the line $y = x$. They have a correlation (using the R function `cor`) given by 0.9995408.

Problem 3

I get that with 100% certainty the value of the stock will be below \$950000 at the close of at least one of the 45 trading days.

Problem 4-7

I get that the hedge fund will make a profit with a probability of 0.38775. I get that the hedge fund will suffer a loss with a probability of 0.58844. I get that the hedge funds expected profit is given by 9922.63 but the expected return (in units of days) -0.01783836.

Exercises

Exercise 2.1

Part (a): To have a value less than \$990 means that we must have a log return less than

$$\log\left(\frac{990}{1000}\right) = -0.01005034\,.$$

To find this probability we evaluate

$$\mathrm{pnorm}(-0.01005034, \mathrm{mean} = 0.001, \mathrm{sd} = 0.015)$$

to get the value 0.2306556.

Part (b): In five trading days our log return will be normally distributed with a mean $5(0.001) = 0.005$ and a standard deviation of $\sqrt{5}(0.015) = 0.03354102$. To be less than \$990 we need to have a logarithm less than -0.01005034 (as computed above). Thus in this case we need to evaluate

$$\mathrm{pnorm}(-0.01005034, \mathrm{mean} = 0.005, \mathrm{sd} = 0.03354102)$$

to get the value 0.3268188.

Exercise 2.2

To have a price greater than 110 we must have a log return greater than $\log\left(\frac{110}{100}\right) = 0.09531018$ in one years time. This will happen with a probability of

$$1 - \mathrm{pnorm}(0.09531018, \mathrm{mean} = 0.1, \mathrm{sd} = 0.2) = 0.509354\,.$$

Exercise 2.3

For this problem we will need to recall the definitions of the *net return*

$$R_t(k) = \frac{P_t - P_{t-k}}{P_{t-k}}\,, \tag{1}$$

and the *log return*

$$r_t(k) = \log(1 + R_t(k))\,. \tag{2}$$

Using the above formulas we have that

$$R_3(2) = \frac{P_3 - P_{3-2}}{P_{3-2}} = \frac{P_3 - P_1}{P_1} = \frac{98 - 95}{95} = 0.03061224\,.$$

Then $r_3(2) = \log(1 + 0.03061224) = 0.03015304$.

Exercise 2.4

Part (a): With dividends our single period gross return is given by

$$R_t = \frac{P_t + D_t - P_{t-1}}{P_{t-1}}\,,$$

so that with the numbers for this problem we get

$$R_2 = \frac{P_2 + D_2 - P_1}{P_1} = \frac{54 + 0.2 - 52}{52} = 0.04230769\,.$$

Part (b): Next recall that with dividends the multiperiod gross returns $R_t(k)$ are given by

$$1 + R_t(k) = \left(\frac{P_t + D_t}{P_{t-1}}\right)\left(\frac{P_{t-1} + D_{t-1}}{P_{t-2}}\right)\cdots\left(\frac{P_{t-k+1} + D_{t-k+1}}{P_{t-k}}\right)\,. \tag{3}$$

Using the above we have

$$\begin{aligned}
1 + R_4(3) &= \left(\frac{P_4 + D_4}{P_3}\right)\left(\frac{P_3 + D_3}{P_2}\right)\left(\frac{P_2 + D_3}{P_1}\right)\\
&= \left(\frac{59 + 0.25}{53}\right)\left(\frac{53 + 0.2}{54}\right)\left(\frac{54 + 0.2}{52}\right) = 1.147959\,.
\end{aligned}$$

Thus $R_4(3) = 0.1479588$.

Part (c): For this part we will use

$$r_3 = \log(1 + R_3) = \log\left(\frac{P_3 + D_3}{P_2}\right) = \log\left(\frac{53 + 0.2}{54}\right) = -0.01492565\,.$$

Exercise 2.5

Part (a): The variable $r_t(4)$ would be a normal random variable with a mean $4(0.06) = 0.24$ and a variance of $4(0.47) = 1.88$ (assuming that the number 0.47 quoted is the one period variance and not standard deviation).

Part (b): We would compute this using the `R` command

$$\text{pnorm}(2, \text{mean} = 0.24, \text{sd} = \text{sqrt}(1.88)) = 0.9003611\,.$$

Part (c): For this recall that

$$\begin{aligned} r_1(2) &= r_1 + r_0 \\ r_2(2) &= r_2 + r_1\,, \end{aligned}$$

where each of r_t is i.i.d. from $N(0.06, 0.47)$. Thus we have that

$$\begin{aligned} \text{Cov}(r_1(2), r_2(2)) &= \text{Cov}(r_1 + r_0, r_2 + r_1) \\ &= \text{Cov}(r_1, r_2) + \text{Cov}(r_1, r_1) + \text{Cov}(r_0, r_2) + \text{Cov}(r_0, r_1) \\ &= 0 + \sigma^2 + 0 + 0 = 0.47\,. \end{aligned}$$

Part (d): For this note that

$$r_t(3) = r_t + r_{t-1} + r_{t-2}\,,$$

Thus if we know that r_{t-2} was equal to 0.6 then $r_t(3)$ is made of only two random components (and a known constant) thus $r_t(3)|\{r_{t-2} = 0.6\}$ is a normal random variable with a mean $2(0.06) + 0.6 = 0.72$ and a variance of $2(0.47) = 0.94$.

Exercise 2.6

Part (a): For this we have

$$\begin{aligned} P(X_2 > 1.3X_0) &= P\left(\frac{X_2}{X_0} > 1.3\right) \\ &= P(r_1 + r_2 > \log(1.3)) \\ &= 1 - P(r_1 + r_2 < \log(1.3)) \\ &= 1 - \text{pnorm}(\log(1.3), \text{mean} = 2\mu, \text{sd} = \sqrt{2}\sigma)\,. \end{aligned}$$

Part (b): For this we first recall that A.4 (with X replaced with R) is given by

$$f_Y(y) = f_R(h(y))|h'(y)|\,. \tag{4}$$

For this problem these functions are

$$\begin{aligned} f_R(r) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(r-\mu)^2}{2\sigma^2}\right\} \\ Y &= g(R) = X_0 e^R \\ R &= h(Y) = \log\left(\frac{Y}{X_0}\right) \quad \text{so} \quad h'(Y) = \frac{X_0}{Y}\,. \end{aligned}$$

Then using Equation 4 we have

$$f_Y(y) = \frac{X_0}{\sqrt{2\pi}\sigma y} \exp\left\{-\frac{(\log(Y/X_0) - \mu)^2}{2\sigma^2}\right\},$$

for the density of $Y = X_1$.

Part (c): As we can write $X_k = X_0 e^R$ where R is a normal random variable with mean a $k\mu$ and variance $k\sigma^2$ the probability density function of the random variable X_k is derived just like the one for X_1 above. In fact we have if $Y = X_k$ we have

$$f_Y(y) = \frac{X_0}{\sqrt{2\pi k}\sigma y} \exp\left\{-\frac{(\log(Y/X_0) - k\mu)^2}{2k\sigma^2}\right\}.$$

Then since the transformation from R to X_k is a monotone transformation the quantiles of R transform to the quantiles of X_k using the monotone transformation. Thus finding the 0.9 quantile of R (by using the `qnorm` command in `R` for a normal with a mean $k\mu$ and a variance $k\sigma^2$) say $\mu_{0.9}$ we find the 0.9 quantile of X_k by computing $X_0 e^{\mu_{0.9}}$.

Exercise 2.7

To have the stock price greater than \$100 means that the log return needs to be larger than

$$\log\left(\frac{100}{97}\right) = 0.03045921.$$

In 20 days the log return should be a normal random variable and have a mean value of $20(0.0002) = 0.004$ with a standard deviation of $\sqrt{20}0.03 = 0.1341641$. The probability we have a return larger than the above (and a final price greater than 100) is given by

$$1 - \text{pnorm}(0.03045921, \text{mean} = 0.004, \text{sd} = 0.1341641) = 0.4218295.$$

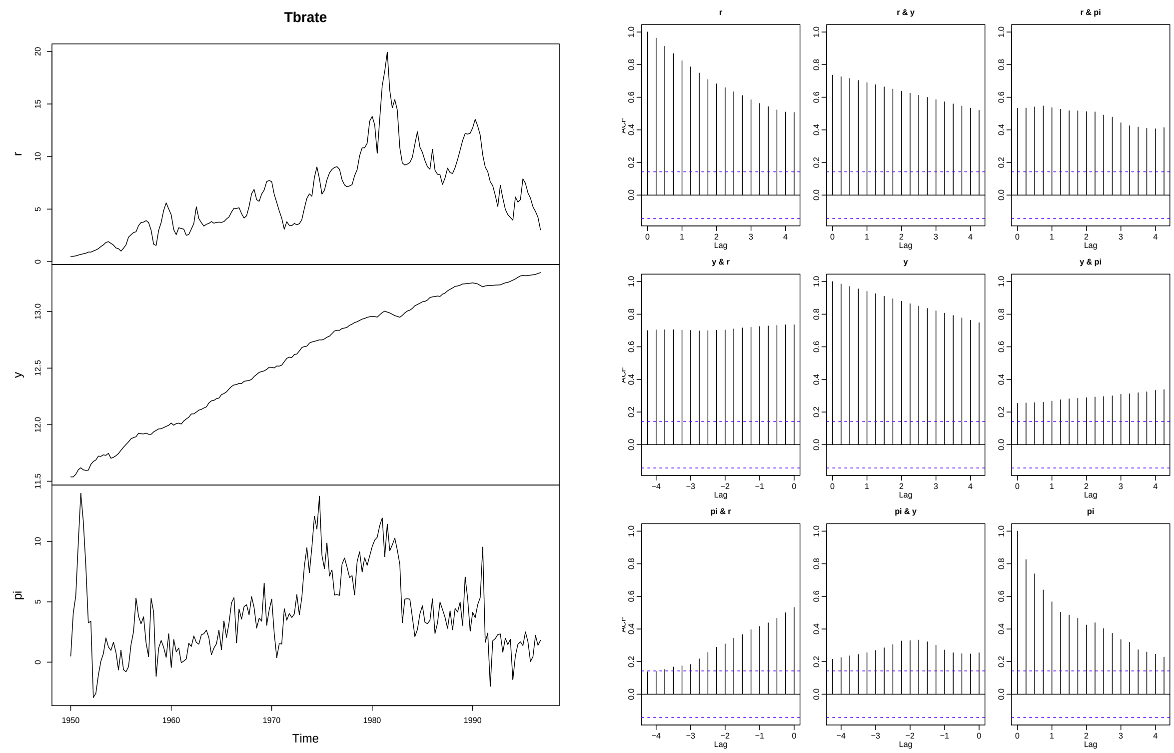

Figure 2: **Left:** The time series plots of the data in the `Tbrate` dataset. **Right:** The autocorrelation plots for the data in the `Tbrate` dataset.

Chapter 9 (Time Series Models: Basics)

R Lab

See the `R` script `Rlab.R` for this chapter.

Problem 1 (stationarity of the raw variables)

Part (a): To start this analysis see Figure 2 (left) for plots of time series for the variables r, y, and π. From that plot, the variables r and y seem to be moving in the same direction for extended periods of time (indicating that they may not be stationary time series). The plot of π seems to be more stationary. Next see Figure 2 (right) for plots of the auto-correlation of the same three variables r, y, and π (on the diagonal) and plots of the cross-correlation functions (on the off-diagonal). The very slow decay of the autocorrelation functions along the diagonal indicate that differencing will probably be needed to obtain stationary time series that we can model with ARMA models.

Part (b): We next run the `adf.test` on each of the given time series before any transformations are taken. Note that the null hypothesis for the `adf.test` is that the time series *has* a unit-root. The alternative hypothesis is that the time series is actually stationary. Thus "large" and negative t-values (or small p-values) indicate that this time series probably *does not* have a unit root. On the other hand "small" t values (or large p-values) indicates that the given time series probably *does* have a unit root. For the variable in this dataframe we find

- For the variable r we get

  ```
  Dickey-Fuller = -1.925, Lag order = 5, p-value = 0.6075
  ```

  Indicating that most likely have a unit-root in this data.

- For the variable y we get

  ```
  Dickey-Fuller = -0.3569, Lag order = 5, p-value = 0.9873
  ```

  Indicating that it is very likely that we have a unit-root in this data.

- For the variable π we get

  ```
  Dickey-Fuller = -2.9499, Lag order = 5, p-value = 0.1788
  ```

  Indicating that there is less chance that we have a unit-root in this data.

Looking back at plots of the time series of the variables using the `plot(Tbrate)` we see that the first two time series seem to be steadily growing while the one for π seems to more stationary. That is what the numbers above state in that y and r seem to very strongly have a unit root while for π there is less evidence of that.

Problem 2 (stationarity of the first differences)

Next we take the first difference of this data and look if the result is more stationary. We first plot the time series of this first difference in Figure 3 (left). There we see that the series are much more mean reverting in this case. Next we plot the ACF functions for these first differences in Figure 3 (right). All of these plots show series that look much more stationary than before.

Next we consider the `adf.test` looking for a unit root for these first differences. For these three time series we get

- For the variable Δr we get

  ```
  Dickey-Fuller = -5.2979, Lag order = 5, p-value = 0.01
  ```

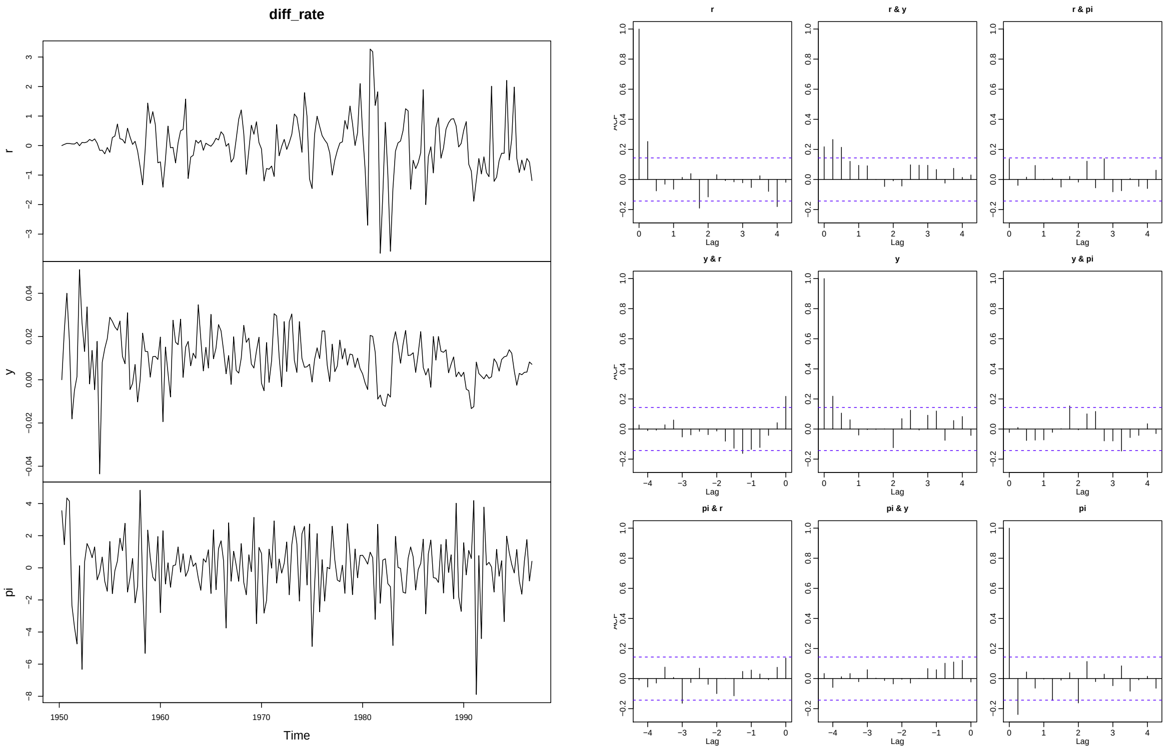

Figure 3: **Left:** The time series plots of the first difference of the data in the `Tbrate` dataset. **Right:** The autocorrelation plots of first difference of the data in the `Tbrate` dataset.

Indicating that there is most likely not a unit-root in this difference.

- For the variable Δy we get

  `Dickey-Fuller = -5.9168, Lag order = 5, p-value = 0.01`

  Also indicating that there is most likely not a unit-root in this difference.

- For the variable $\Delta \pi$ we get

  `Dickey-Fuller = -7.6571, Lag order = 5, p-value = 0.01`

  Again indicating that there is most likely not a unit-root in this difference.

We conclude that all three series appear stationary after taking first differences. Looking at the ACF plots we see that there is evidence that these first difference could be modeled as a MA(1) process.

Problem 3 (seasonality differences)

Next we look to see if the mean level for Δr depends on the quarter. In Figure 4 we construct a box plot of the samples of Δr that fall in each quarter. From that plot it *does not* look like there is much dependence on the mean level with the quarter.

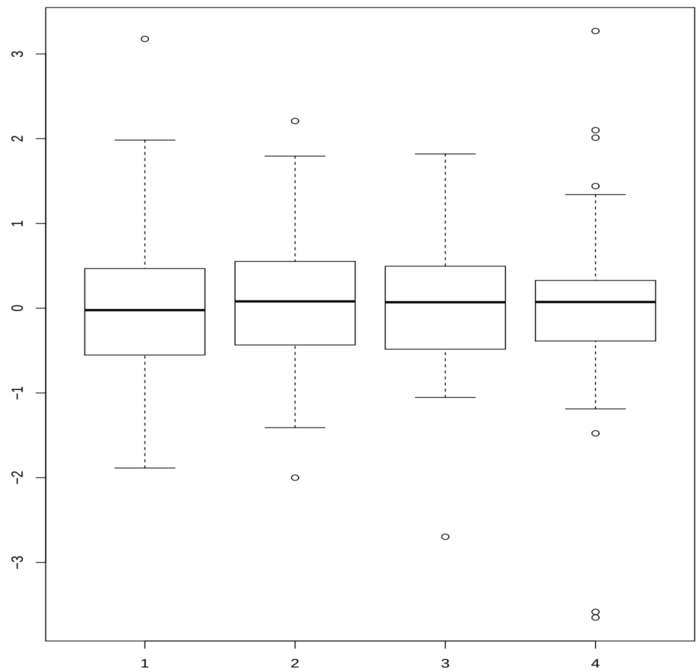

Figure 4: A box plot of Δr grouped by the quarter $1, 2, 3$ or 4 the measurement was taken in.

Problem 4 (fitting an ARIMA model)

The output from the `auto.arima` function call is

```
> auto.arima( Tbrate[,1], max.P=0, max.Q=0, ic="aic" )
Series: Tbrate[, 1]
ARIMA(0,1,1)

Coefficients:
         ma1
      0.3275
s.e.  0.0754

sigma^2 estimated as 0.8096:  log likelihood=-245.65
AIC=495.3   AICc=495.37   BIC=501.76
```

Which indicates that when we use the AIC information criterion we choose to take the first difference $d = 1$ and then to model Δr with a MA(1) model. This is in line with what the ACF plot of Δr above suggested. Changing the `ic` argument in the `auto.arima` call to `bic` does not change the final model selected.

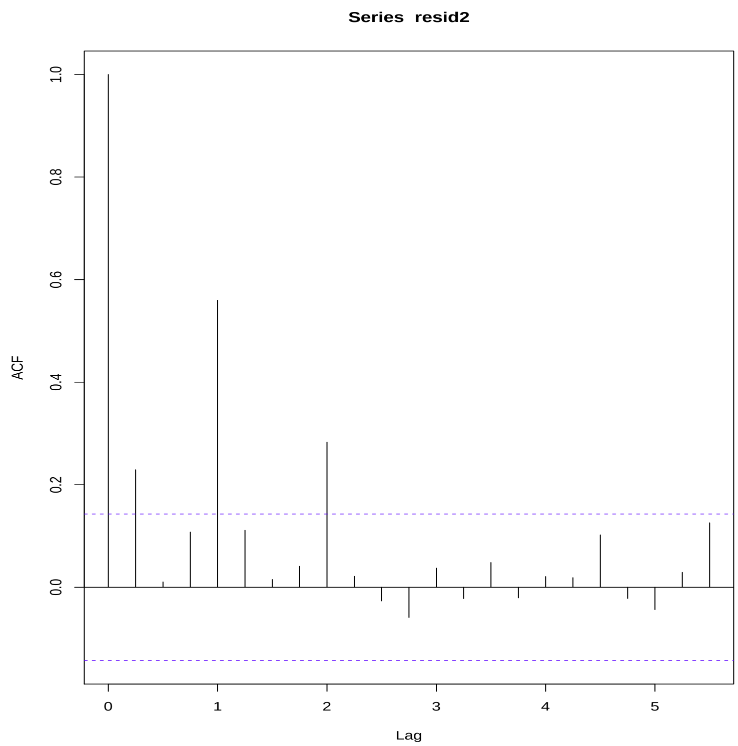

Figure 5: The autocorrelation function of the ARIMA residuals squared. Notice there are several spikes above the 2σ horizontal lines indicating evidence of GARCH effects.

Problem 5 (residual autocorrelation)

The plot of the autocorrelation function of the ARIMA model residual shows no spikes reaching far outside of the confidence intervals plotted on the graph. There are some spikes that are just outside these limits but these are probability due to statistical fluctuations rather than to a real phenomena that we should try to model. The output from the `Box.test` command is

```
X-squared = 13.0169, df = 10, p-value = 0.2227
```

Which indicates that if we accept the hypothesis of *randomly ordered data* there is a probability of around 0.22 of getting results as “extream” as the ones we have. We would like the p-value returned from this test to be as large as possible but at this point we can conclude that our model of the time series r is complete.

Problem 6 (GARCH effects)

The plot of the autocorrelation function for the residuals squares is given in Figure 5. There we see two large spikes above the 2σ horizontal lines. The output from the `Box.test` also

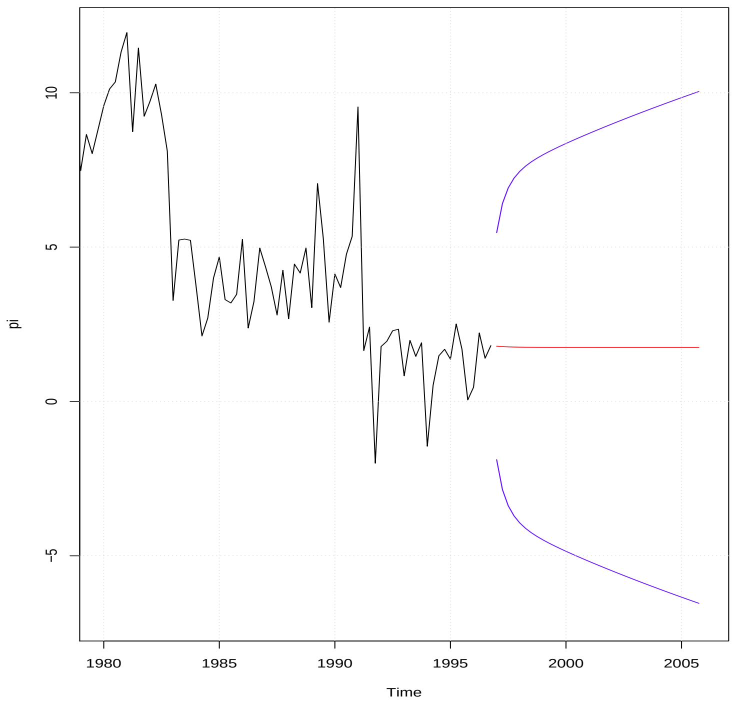

Figure 6: Forecasts of inflation rate as we move forward in time.

indicates the presence of non-randomness

```
X-squared = 92.1004, df = 10, p-value = 1.998e-15
```

A p-value this small means that it is very unlikely that this data is purely random and we see evidence of GARCH effects.

Problem 7 (forecasting)

The output from the `auto.arima` call is

```
> auto.arima( Tbrate[,3], max.P=0, max.Q=0, ic="bic" )
Series: Tbrate[, 3]
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.6749  -0.9078
s.e.  0.0899   0.0501
```

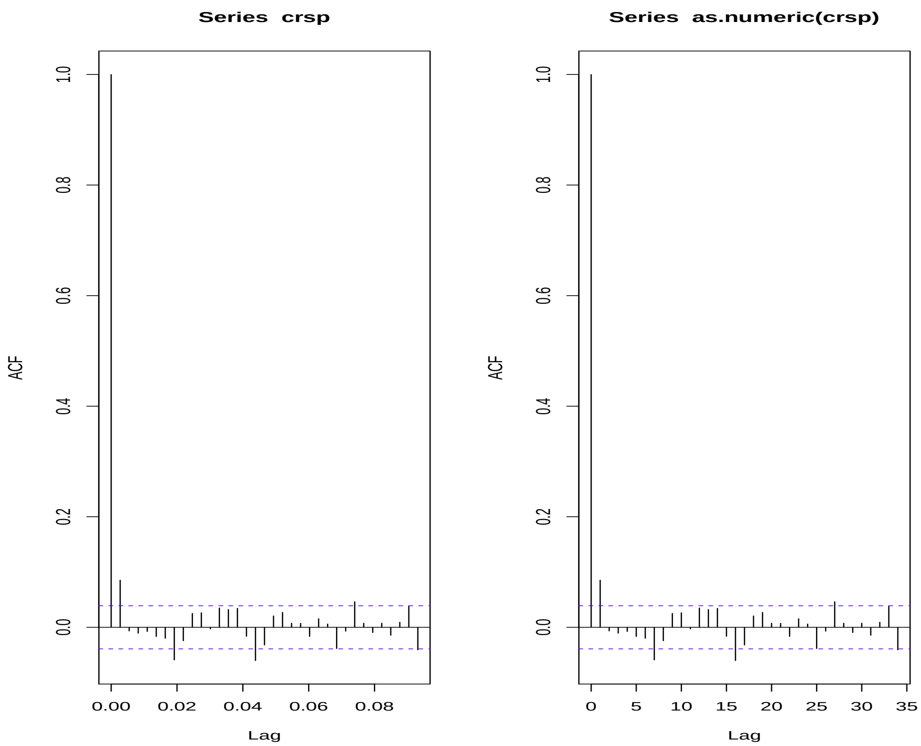

Figure 7: **Left:** The `acf` plot for the `crsp` data (as a time series object). **Right:** The `acf` plot for the `crsp` data (treated as a numeric array).

```
sigma^2 estimated as 3.516:  log likelihood=-383.12
AIC=772.24   AICc=772.37   BIC=781.94
```

Which indicates that we should model this with an ARIMA(1,1,1) model.

We plot the forecast and the standard errors in Figure 6. In a non rigorous way, the prediction intervals widen since at every timestep our ΔY_t build from terms of the form $\epsilon_t + \theta_1 \epsilon_{t-1}$ (along with other terms). These two terms are random and have an associated variance and as we predict further in the future we have to add more and more of these random variables to estimate ΔY_t. The more random variables we add together the larger the variance of the resulting sum.

Exercises

See the `R` code `chap_9.R` where these problems are worked.

Exercise 9.1

Plots of the output from the `acf` function for each of the two calls are given in Figure 7.

Part (a): Note that variable `crsp` is a `ts` (time series) object and as such has some auxiliary data associated with it. If we try to display the variable `crsp` in the command window we get the following (partial output)

```
> crsp
Time Series:
Start = c(1969, 1)
End = c(1975, 338)
Frequency = 365
   [1] -0.007619  0.013016  0.002815  0.003064  0.001633 -0.001991  0.004671
   [8]  0.004027  0.001736  0.001504 -0.001778  0.008950  0.001920 -0.000024
```

When we plot the `acf` of the `ts` object the lag is in units of *time* since the time series object has a notion of how much time is represented between each data point. For the output above we see that the frequency is 365 (corresponding to the number of days in a year) thus a lag of one day corresponds to $\frac{1}{365} = 0.002739726$ yearly time units which is the spacing between the `acf` values in the leftmost plot. If we look at the rightmost `acf` plot in Figure 7 we see a small downward spike at lag seven. In units of *time* this is located at $\frac{7}{365} = 0.01917808$ in the leftmost plot.

Part (b): We see that the three significant autocorrelations (with the smallest lags) are found at lags 1, 7, and 16. I would expect the autocorrelation at lag 1 to be statistically significant but the others are most likely due to change. Autocorrelations with larger lags that seem "significant" (in that they are above the 2σ standard error bars) are also probably due to chance.

Exercise 9.2

Part (a): From the autocorrelation plots for the `crsp` data it looks like a better model to fit would be a MA(1) model rather than any AR(p) model (as suggested in this exercise). In any case, if we are to fit AR(1) and AR(2) model the output from the two `arima` calls is

```
> arima(crsp,order=c(1,0,0))
Series: crsp
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1  intercept
      0.0853      7e-04
```

```
s.e.  0.0198      2e-04

sigma^2 estimated as 5.973e-05:  log likelihood=8706.18
AIC=-17406.37   AICc=-17406.36   BIC=-17388.86
> arima(crsp,order=c(2,0,0))
Series: crsp
ARIMA(2,0,0) with non-zero mean

Coefficients:
         ar1      ar2  intercept
      0.0865  -0.0141      7e-04
s.e.  0.0199   0.0199      2e-04

sigma^2 estimated as 5.972e-05:  log likelihood=8706.43
AIC=-17404.87   AICc=-17404.85   BIC=-17381.53
```

In comparing these two models we would select the one that has the lower value of the Akaike information criterion (aic) or the Bayesian information criterion (bic). For either of these two metrics the AR(1) model is preference to the AR(2) model. We note that we fit a MA(1) model and looked at its AIC (or BIC) it is a smaller value than either of these two AR(p) models and would be the preferred model.

Part (b): Using the data given for the AR(1) fit we have a 95% confidence interval for the value of ϕ given by

```
alpha = 0.05
0.0853 + 0.0198 * qnorm( 1 - 0.5 * alpha ) * c(-1,+1)
[1] 0.04649271 0.12410729
```

Exercise 9.3 (an AR(1) model)

Part (a): An AR(1) model looks like

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \epsilon_t \,.$$

To have a stationary AR(1) model we must have $|\phi| < 1$. Since for this problem we have $\phi = -0.55$ we do have $|\phi| < 1$ and this AR model is stationary.

Part (b): We can write the above expression for an AR(1) model in the form

$$Y_t = (1 - \phi)\mu + \phi Y_{t-1} + \epsilon_t \,.$$

With $\phi = -0.55$ and equating this to the model we were given in this problem we have

$$\mu = \frac{5}{1 - \phi} = \frac{5}{1 + 0.55} = 3.225806 \,.$$

Part (c): From the book we have that

$$\gamma(0) = \text{Var}\,(Y_t) = \frac{\sigma_\epsilon^2}{1-\phi^2} = \frac{1.2}{1-0.55^2} = 1.72043\,.$$

Part (d): From the book we have that

$$\gamma(h) = \text{Cov}(Y_t, Y_{t+h}) = \frac{\sigma_\epsilon^2 \phi^{|h|}}{1-\phi^2} = 1.72043(-0.55)^{|h|}\,.$$

Exercise 9.4 (another AR(1) model)

Part (a): For an AR(1) model we know

$$\text{Var}\,(Y_1) = \frac{\sigma_\epsilon^2}{1-\phi^2} = \frac{1.2}{1-0.5^2} = 1.6\,.$$

Part (b): Again from the discussion in the book on AR(1) models we have

$$\begin{aligned}
\text{Cov}(Y_1, Y_2) &= \frac{\sigma_\epsilon^2 \phi^{|1|}}{1-\phi^2} = \frac{1.2(0.5)}{1-0.5^2} = 0.8 \\
\text{Cov}(Y_1, Y_3) &= \frac{\sigma_\epsilon^2 \phi^{|2|}}{1-\phi^2} = \frac{1.2(0.5)^2}{1-0.5^2} = 0.4\,.
\end{aligned}$$

Part (c): We can compute this expression as follows

$$\begin{aligned}
\text{Var}\left(\frac{Y_1+Y_2+Y_3}{2}\right) &= \frac{1}{4}\text{Var}(Y_1+Y_2+Y_3) \\
&= \frac{1}{4}\left[\sum_{i=1}^{3}\text{Var}\,(Y_i) + 2\sum_{i<j}\text{Cov}(Y_i, Y_j)\right] \\
&= \frac{1}{4}\left[3\text{Var}\,(Y_1) + 2\left(\text{Cov}(Y_1, Y_2) + \text{Cov}(Y_1, Y_3) + \text{Cov}(Y_2, Y_3)\right)\right] \\
&= \frac{1}{4}\left[3(1.6) + 2\left(0.8 + 0.4 + 0.8\right)\right] = 2.2\,.
\end{aligned}$$

Exercise 9.5 (an AR(3) model)

An AR(3) model takes the form

$$Y_t - \mu = \phi_1(Y_{t-1}-\mu) + \phi_2(Y_{t-2}-\mu) + \phi_3(Y_{t-3}-\mu) + \epsilon_t\,. \tag{5}$$

To prediction the value of Y_{t+1} we incrementing the t index by one to get

$$Y_{t+1} - \mu = \phi_1(Y_t-\mu) + \phi_2(Y_{t-1}-\mu) + \phi_3(Y_{t-2}-\mu) + \epsilon_{t+1}\,,$$

and then taking expectations (conditioned data that has arrived by the time t) to get

$$\begin{aligned}\hat{Y}_{t+1} &= \hat{\mu} + \hat{\phi}_1(Y_t - \hat{\mu}) + \hat{\phi}_2(Y_{t-1} - \hat{\mu}) + \hat{\phi}_3(Y_{t-2} - \hat{\mu}) \\ &= 104 + 0.4(99 - 104) + 0.25(103 - 104) + 0.1(102 - 104) = 101.55\,.\end{aligned}$$

For the prediction of Y_{t+2} we use Equation 5 again (evaluated at $t = t + 2$) to get

$$\begin{aligned}\hat{Y}_{t+2} &= \hat{\mu} + \hat{\phi}_1(\hat{Y}_{t+1} - \hat{\mu}) + \hat{\phi}_2(Y_t - \hat{\mu}) + \hat{\phi}_3(Y_{t-1} - \hat{\mu}) \\ &= 104 + 0.4(101.55 - 104) + 0.25(99 - 104) + 0.1(103 - 104) = 101.67\,.\end{aligned}$$

Exercise 9.6 (the autocovariance of a MA(2) model)

For the MA(2) model

$$Y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2}\,,$$

note that

$$E[Y_t] = \mu\,.$$

Thus we can compute

$$\begin{aligned}\gamma(h) = \operatorname{Cov}(Y_t, Y_{t+h}) &= E[(Y_t - \mu)(Y_{t+h} - \mu)] \\ &= E[(\epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2})(\epsilon_{t+h} + \theta_1\epsilon_{t+h-1} + \theta_2\epsilon_{t+h-2})] \\ &= E[\epsilon_t\epsilon_{t+h}] + \theta_1 E[\epsilon_t\epsilon_{t+h-1}] + \theta_2 E[\epsilon_t\epsilon_{t+h-2}] \\ &+ \theta_1 E[\epsilon_{t-1}\epsilon_{t+h}] + \theta_1^2 E[\epsilon_{t-1}\epsilon_{t+h-1}] + \theta_1\theta_2 E[\epsilon_{t-1}\epsilon_{t+h-2}] \\ &+ \theta_2 E[\epsilon_{t-2}\epsilon_{t+h}] + \theta_1\theta_2 E[\epsilon_{t-2}\epsilon_{t+h-1}] + \theta_2^2 E[\epsilon_{t-2}\epsilon_{t+h-2}]\,.\end{aligned}$$

In the above if $h = 0$ then we get

$$\gamma(0) = (1 + \theta_1^2 + \theta_2^2)\sigma_\epsilon^2\,.$$

In the above if $h = \pm 1$ then we get

$$\gamma(1) = (\theta_1 + \theta_1\theta_2)\sigma_\epsilon^2\,.$$

In the above if $h = \pm 2$ then we get

$$\gamma(2) = \theta_2\sigma_\epsilon^2\,.$$

In the above we get $\gamma(h) = 0$ if $|h| > 2$. Using these results we have for the autocorrelation of Y_t

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \begin{cases} 1 & h = 0 \\ \frac{\theta_1 + \theta_1\theta_2}{1 + \theta_1^2 + \theta_2^2} & h = \pm 1 \\ \frac{\theta_2}{1 + \theta_1^2 + \theta_2^2} & h = \pm 2 \\ 0 & |h| > 2 \end{cases}$$

Exercise 9.7

An AR(2) process has the form

$$Y_t - \mu = \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t \,.$$

Part (a): Now given this process we can compute the autocovariance function as

$$\begin{aligned}
\gamma(k) &= E[(Y_t - \mu)(Y_{t-k} - \mu)] \\
&= E[(\phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t)(Y_{t-k} - \mu)] \\
&= \phi_1 E[(Y_{t-1} - \mu)(Y_{t-k} - \mu)] + \phi_2 E[(Y_{t-2} - \mu)(Y_{t-k} - \mu)] + 0 \\
&= \phi_1 \gamma(k-1) + \phi_2 \gamma(k-2) \,.
\end{aligned}$$

Note that the last term in the equation above the last is zero since ϵ_t and Y_{t-k} are independent and $E[\epsilon_t] = 0$. If we now divide both sides by $\gamma(0)$ we get

$$\rho(k) = \phi_1 \rho(k-1) + \phi_2 \rho(k-2) \,,$$

the expression we were trying to show.

Part (b): If we let $k = 1$ in the above equation then we get

$$\rho(1) = \phi_1(1) + \phi_2 \rho(-1) = \phi_1 + \phi_2 \rho(1) \,.$$

If we let $k = 2$ in the above equation we get

$$\rho(2) = \phi_1 \rho(1) + \phi_2(1) \,.$$

Putting these two equations together gives the system

$$\left[\begin{array}{c} \rho(1) \\ \rho(2) \end{array} \right] = \left[\begin{array}{cc} 1 & \rho(1) \\ \rho(1) & 1 \end{array} \right] \left[\begin{array}{c} \phi_1 \\ \phi_2 \end{array} \right] .$$

Thus if we think that a time series is generated from an AR(2) model we could measure $\rho(1)$ and $\rho(2)$ numerically and then invert the above system to compute values for ϕ_1 and ϕ_2 that would be estimates of the parameters in this AR(2) model.

Part (c): These values would give rise to the system

$$\left[\begin{array}{c} 0.4 \\ 0.2 \end{array} \right] = \left[\begin{array}{cc} 1 & 0.4 \\ 0.4 & 1 \end{array} \right] \left[\begin{array}{c} \phi_1 \\ \phi_2 \end{array} \right] ,$$

which if we invert gives $\phi_1 = 0.38095238$ and $\phi_2 = 0.04761905$. Using these parameter estimates we can compute $\rho(3)$ as

$$\rho(3) = \phi_1 \rho(2) + \phi_2 \rho(1) = 0.38095238(0.2) + 0.04761905(0.4) = 0.0952381 \,.$$

Exercise 9.8

The left-hand-side of the book's equation 9.12 is

$$\text{Cov}\left(\sum_{i=0}^{\infty}\epsilon_{t-i}\phi^i, \sum_{j=0}^{\infty}\epsilon_{t+h-j}\phi^j\right) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\phi^i\phi^j\text{Cov}(\epsilon_{t-i}, \epsilon_{t+h-j}) .$$

Now $\text{Cov}(\epsilon_{t-i}, \epsilon_{t+h-j}) = 0$ unless $t - i = t + h - j$ or $j = h + i$. Thus our double summation above simplifies to a single summation to give

$$\sum_{i=0}^{\infty}\phi^i\phi^{h+i}\text{Cov}(\epsilon_{t-i}, \epsilon_{t-i}) = \sigma_\epsilon^2\phi^h\sum_{i=0}^{\infty}\phi^{2i} = \sigma_\epsilon^2\phi^h\left(\frac{1}{1-\phi^2}\right) .$$

Since the autocovariance function is symmetric in h we need to take the absolute value of h in the above to get $\phi^{|h|}$ which gives the desired expression.

Exercise 9.9

The expression for w_t is

$$w_t = w_{t_0} + Y_{t_0} + Y_{t_0+1} + \cdots + Y_t ,$$

so that when we compute Δw_t we get

$$\begin{aligned}\Delta w_t &= w_{t_0} + Y_{t_0} + Y_{t_0+1} + \cdots + Y_t \\ &- (w_{t_0} + Y_{t_0} + Y_{t_0+1} + \cdots + Y_{t-1}) = Y_t ,\end{aligned}$$

as we were to show.

Exercise 9.10

For a large enough length of time maybe. One could get some evidence if an $I(2)$ model would fit a given time series of prices p_t by looking at the results of the Ljung-Box test on the second difference sequence $\Delta^2 p_t$.

Exercise 9.11

Let $t = n + 1$ and from the form of Y_t we see that Y_{n+1} is given by

$$Y_{n+1} = \mu + \epsilon_{n+1} + \theta_1\epsilon_n + \theta_2\epsilon_{n-1} ,$$

so from this we see that our prediction is given by

$$\begin{aligned}\hat{Y}_{n+1} &= \hat{\mu} + 0 + \hat{\theta}_1\hat{\epsilon}_n + \hat{\theta}_2\hat{\epsilon}_{n-1} \\ &= 45 + 0.3(1.5) - 0.15(-4.3) = 46.095 .\end{aligned}$$

For Y_{n+2} from the form of Y_t we see that we have

$$Y_{n+2} = \mu + \epsilon_{n+2} + \theta_1 \epsilon_{n+1} + \theta_2 \epsilon_n \,.$$

Then using this our prediction $\hat{Y}_{n+2}$ is given by

$$\begin{aligned}\hat{Y}_{n+2} &= \hat{\mu} + 0 + 0 + \hat{\theta}_2 \hat{\epsilon}_n \\ &= 45 - 0.15(1.5) = 44.775 \,.\end{aligned}$$

Exercise 9.12

For the given time series model with $t = n + 1$ we have

$$Y_{n+1} = \mu + \phi_1 Y_n + \epsilon_{n+1} + \theta_1 \epsilon_n + \theta_2 \epsilon_{n-1} \,,$$

so our prediction of Y_{n+1} is given by

$$\begin{aligned}\hat{Y}_{n+1} &= \hat{\mu} + \hat{\phi}_1 Y_n + 0 + \hat{\theta}_1 \hat{\epsilon}_n + \hat{\theta}_2 \hat{\epsilon}_{n-1} \\ &= 103 + 0.2(118.3) + 0.4(2.6) - 0.25(-2.3) = 128.275 \,.\end{aligned}$$

Next using the given time series model with $t = n + 2$ we have

$$Y_{n+2} = \mu + \phi_1 Y_{n+1} + \epsilon_{n+2} + \theta_1 \epsilon_{n+1} + \theta_2 \epsilon_n \,,$$

so our prediction of Y_{n+2} is given by

$$\begin{aligned}\hat{Y}_{n+2} &= \hat{\mu} + \hat{\phi}_1 \hat{Y}_{n+1} + 0 + 0 + \hat{\theta}_2 \hat{\epsilon}_n \\ &= 103 + 0.2(128.275) - 0.25(2.6) = 128.005 \,.\end{aligned}$$

Exercise 9.13

The autocorrelation plot of Δy_t shows a function that is effectively zero and thus the simplest model would be to model the time series of y_t as $\Delta y_t = \epsilon_t$ (i.e. $d = 1$).

Exercise 9.14

Part (a): We plot the time series in Figure 8. There we see trending that indicates that we do *not* have a stationary time series. In Figure 9 (left) we plot the autocorrelation function of the raw time series. The very slow decay of the values indicates that we need to take a forward different of this time series to proceed to try to model it with an ARMA model. In Figure 9 (right) we plot the autocorrelation function of the first difference of the time series. There it looks like we can model this first difference with a MA(1) model. Well it looks like there are several spikes in the autocorrelation plot that are above the 2σ horizontal lines and that these seem to be spaced periodically in time. Thus we would probably need a *seasonal* MA model to fully model this process.

Part (b): The output from the `auto.arima` command using the AIC option is

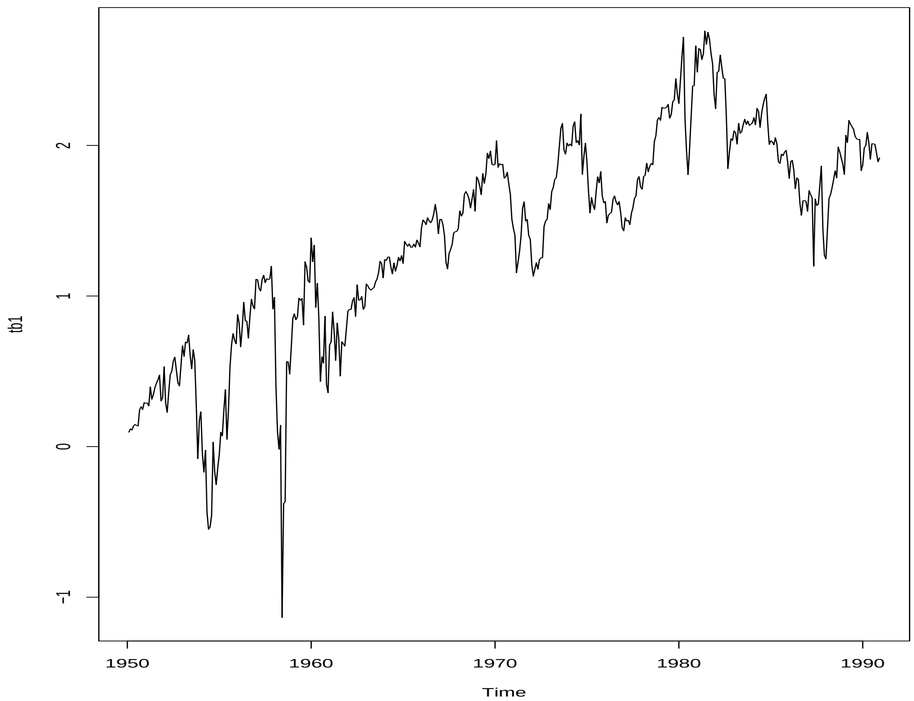

Figure 8: A plot of the tb1 time series.

```
> auto.arima( tb1, max.P=0, max.Q=0, ic="aic" )
Series: tb1
ARIMA(5,1,3)

Coefficients:
         ar1      ar2     ar3     ar4      ar5      ma1     ma2      ma3
      0.7659  -0.7175  0.6176  0.2595  -0.1117  -0.9432  0.9001  -0.8921
s.e.  0.0578   0.0690  0.0697  0.0576   0.0480   0.0387  0.0513   0.0389

sigma^2 estimated as 0.0199:  log likelihood=263.81
AIC=-509.62   AICc=-509.25   BIC=-471.87
```

This does not look like a very parsimonious model (given the large number of parameters that used in fitting it). The output from the `auto.arima` command using the BIC option is much cleaner

```
> auto.arima( tb1, max.P=0, max.Q=0, ic="bic" )
Series: tb1
ARIMA(0,1,1)

Coefficients:
          ma1
```

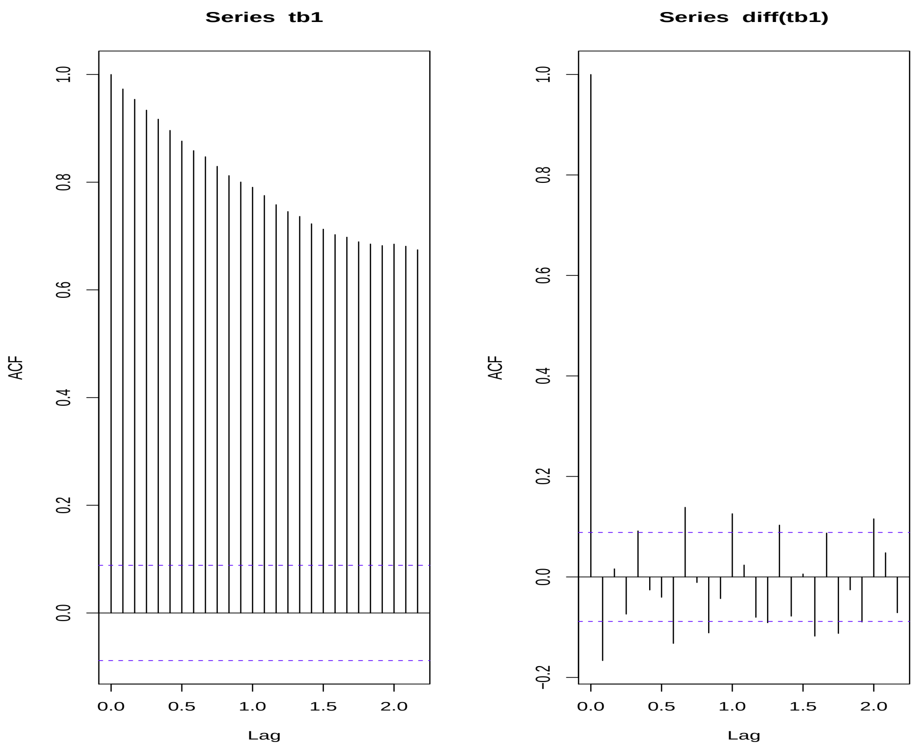

Figure 9: **Left:** A plot of the autocorrelation function for the `tb1` time series (directly). **Right:** A plot of the autocorrelation function of the first difference of the `tb1` time series.

```
      -0.1692
s.e.   0.0448

sigma^2 estimated as 0.02158:  log likelihood=244.48
AIC=-484.95   AICc=-484.93   BIC=-476.57
```

Which is a much simpler model.

Part (c): We plot the autocorrelation function of the residuals of the model fit above using the AIC in Figure 10 (left). There we still see a significant spike at the integer lag 21 which might be problematic. The autocorrelation function of the model fit above using the BIC is given in Figure 10 (right). There we see several spikes that are outside of the 2σ horizontal lines. Both of these results indicate that we have not modeled this series completely and we should probably add a seasonal component.

Exercise 9.15

An AR(2) time series model looks like

$$Y_t = \mu + \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t \,,$$

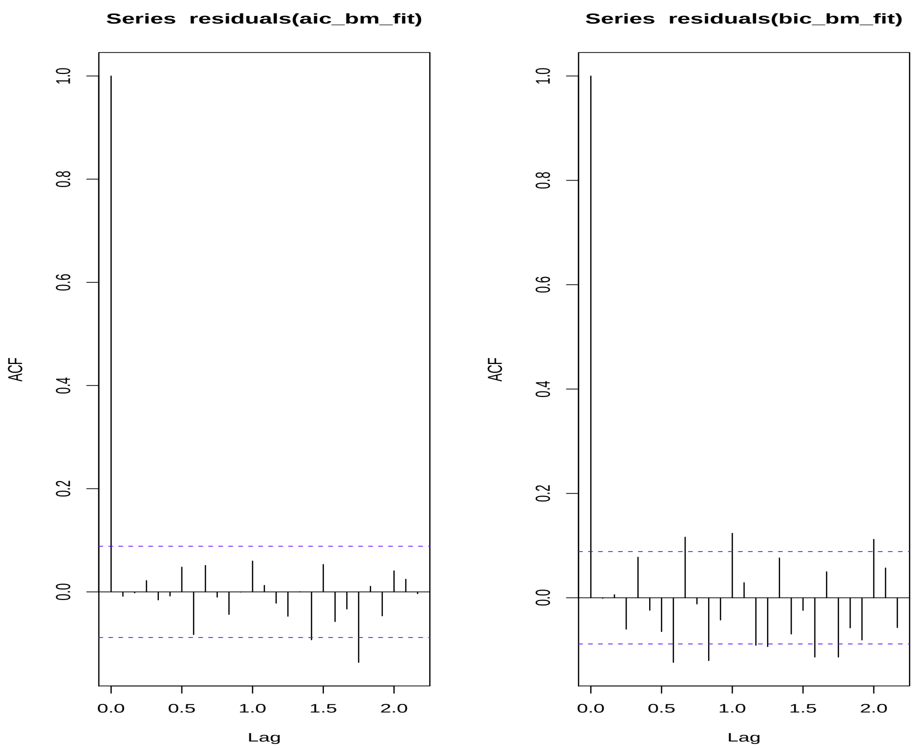

Figure 10: **Left:** A plot of the autocorrelation function for the residuals of the `tb1` time series when we model this as an ARIMA(5,1,3) series. **Right:** A plot of the autocorrelation function for the residuals of the `tb1` time series when we model this as an ARIMA(0,1,1) series.

so letting $t = n + 1$ we get

$$Y_{n+1} = \mu + \phi_1(Y_n - \mu) + \phi_2(Y_{n-1} - \mu) + \epsilon_{n+1}\,.$$

Using this, our prediction of Y_{n+1} is then given by

$$\begin{aligned}\hat{Y}_{n+1} &= \hat{\mu} + \hat{\phi}_1(Y_n - \hat{\mu}) + \hat{\phi}_2(Y_{n-1} - \hat{\mu}) + 0 \\ &= 100.1 + 0.5(102.3 - 100.1) + 0.1(99.5 - 100.1) = 101.14\,.\end{aligned}$$

Next letting $t = n + 2$ in our time series model we get

$$Y_{n+2} = \mu + \phi_1(Y_{n+1} - \mu) + \phi_2(Y_n - \mu) + \epsilon_{n+2}\,.$$

Using this, our prediction of Y_{n+2} is then given by

$$\begin{aligned}\hat{Y}_{n+2} &= \hat{\mu} + \hat{\phi}_1(\hat{Y}_{n+1} - \hat{\mu}) + \hat{\phi}_2(Y_n - \hat{\mu}) + 0 \\ &= 100.1 + 0.5(101.14 - 100.1) + 0.1(102.3 - 100.1) = 100.84\,.\end{aligned}$$

Next letting $t = n + 3$ in our time series model we get

$$Y_{n+3} = \mu + \phi_1(Y_{n+2} - \mu) + \phi_2(Y_{n+1} - \mu) + \epsilon_{n+3}\,.$$

Using this, our prediction of Y_{n+3} is then given by

$$\begin{aligned}\hat{Y}_{n+3} &= \hat{\mu} + \hat{\phi}_1(\hat{Y}_{n+2} - \hat{\mu}) + \hat{\phi}_2(\hat{Y}_{n+1} - \hat{\mu}) + 0 \\ &= 100.1 + 0.5(100.84 - 100.1) + 0.1(101.14 - 100.1) = 100.574\,.\end{aligned}$$

Exercise 9.16

We will just make arguments along these lines. First if Y_t has an mth degree polynomial time trend then we can write $E(Y_t) = \sum_{k=0}^{m} \beta_k t^k$. Note that the expectation of the first difference of Y_t is then given by

$$E(\Delta Y_t) = \sum_{k=0}^{m} \beta_k \Delta t^k\,.$$

Lets looks at what a few forward differences look like. Note that the forward difference of 1, t, and t^2 are given by

$$\begin{aligned}\Delta 1 &= 1 - 1 = 0 \\ \Delta t &= t - (t-1) = 1 \\ \Delta t^2 &= t^2 - (t-1)^2 = t^2 - (t^2 - 2t + 1) = 2t - 1\,.\end{aligned}$$

From these it looks like the first difference of a monomial is a polynomial of one less degree. We can prove this in general using the binomial theorem as follows

$$\begin{aligned}\Delta t^k &= t^k - (t-1)^k = t^k - \sum_{l=0}^{k} \binom{k}{l} t^l (-1)^{k-l} \\ &= t^k - \left(t^k + \sum_{l=0}^{k-1} \binom{k}{l} t^l (-1)^{k-l} \right) = -\sum_{l=0}^{k-1} \binom{k}{l} t^l (-1)^{k-l}\,,\end{aligned}$$

which is a polynomial of degree $k-1$ (one less than we started with). Thus for each forward difference we take we reduce the order of the polynomial time trend by one. If we take d differences we will end with a polynomial time trend of degree $m-d$. If d is larger than m then $E(\Delta^d Y_t) = 0$.

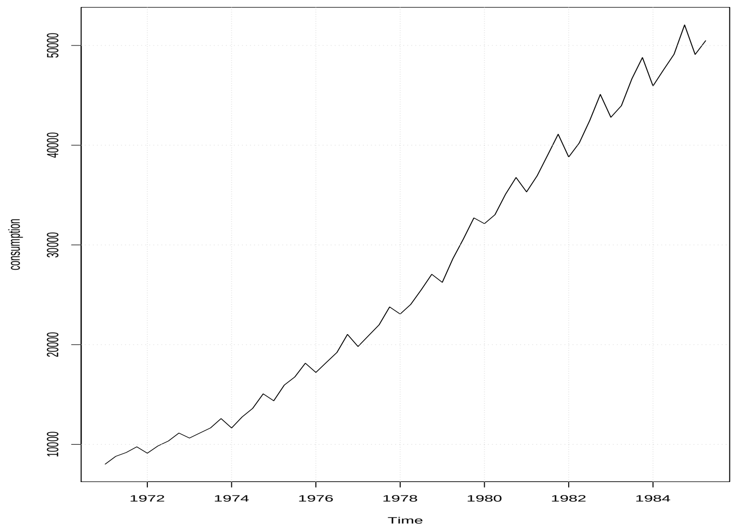

Figure 11: The time series plot of the `consumption` data.

Chapter 10 (Time Series Models: Further Topics)

R Lab

See the `R` script `Rlab.R` where the problem for this chapter are worked.

Problem 1 (seasonal ARIMA models)

From the plot of `consumption` given in Figure 11 we see that the time series is not stationary and will need to take differences to make it stationary. In addition, noticing the periodic "up ticks" that seem to happen in the third quarter of every year imply that we need seasonal differencing to make this time series stationary. This also gives the indication that the order of the seasonal difference should be four.

In Figure 12 we plot the autocorrelation functions (ACF) for various differences of the `consumption` time series data.

- The leftmost plot is the ACF for the untransformed `consumption` data.
- The left-middle plot is the ACF for the first difference of the `consumption` data.
- The right-middle plot is the ACF for the seasonal difference (of order four) of the `consumption` data.

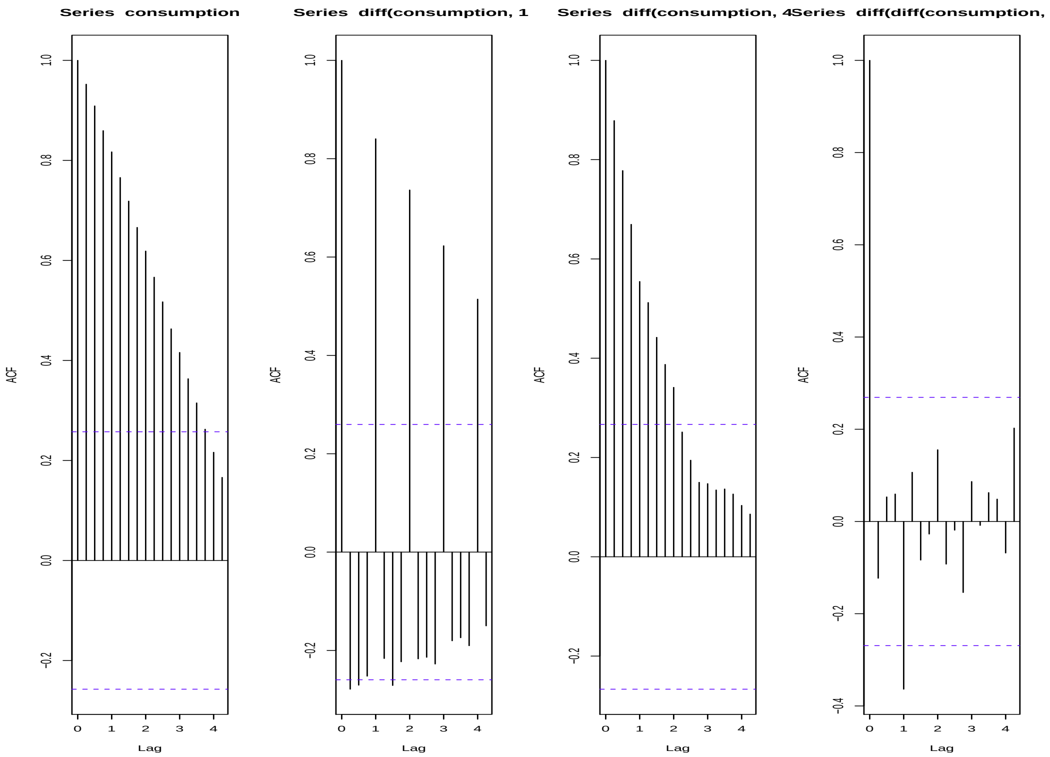

Figure 12: Plots of the autocorrelation functions for the `consumption` time series.

- The rightmost plot is the ACF for the combined nonseasonal and seasonal difference (of order four) of the `consumption` data.

Taking a logarithmic transform of the data gives an ACF that look very much the same as these presented here. In each case it looks like one seasonal difference combined with one nonseasonal difference can be modeled with a seasonal moving average (MA) model with a single coefficient. Fitting this model to both the raw `consumption` time series and the logarithm of `consumption` we will compare which is a better fit by looking at the Ljung-Box test on each models residuals. When we do that we get

```
> consumption_ts_model = arima( consumption, order=c(0,1,0),
                                seasonal=list(order=c(0,1,1), period=4) )
> Box.test( residuals(consumption_ts_model), lag=10, type="Ljung" )
        Box-Ljung test
X-squared = 6.5425, df = 10, p-value = 0.7678

> log_consumption_ts_model = arima( lconsumption, order=c(0,1,0),
                                    seasonal=list(order=c(0,1,1), period=4) )
> Box.test( residuals(log_consumption_ts_model), lag=10, type="Ljung" )
        Box-Ljung test
X-squared = 4.1145, df = 10, p-value = 0.942
```

Intuitively, larger P-values in the `Box.text` indicate that the time series looks like it is a sequence of *independent* events. Since the logarithm of consumption has a larger P-value we might argue that taking the logarithm results in a better fit since the model then produces more independent residuals.

Problem 2 (fitting a seasonal ARIMA model)

Based on the above discussion, we propose a ARIMA$\{(0,1,0) \times (0,1,1)_4\}$ model for the logarithm of consumption. Fitting such a model gives

```
> arima( lconsumption, order=c(0,1,0), seasonal=list(order=c(0,1,1), period=4) )
ARIMA(0,1,0)(0,1,1)[4]

Coefficients:
         sma1
      -0.5348
s.e.   0.1164

sigma^2 estimated as 0.0002923:  log likelihood=139.77
AIC=-275.55   AICc=-275.31   BIC=-271.61
```

Problem 3 (the residuals of the fit)

A plot of the autocorrelation function of the residuals (not shown) for the model above show no statistically significant autocorrelations.

Problem 4 (using `auto.arima`)

The model found by `auto.arima` is the same model as suggested from the `acf` plots. Running `auto.arima` gives

```
> auto.arima( lconsumption, ic="bic" )
ARIMA(0,1,0)(0,1,1)[4]

Coefficients:
         sma1
      -0.5348
s.e.   0.1164

sigma^2 estimated as 0.0002923:  log likelihood=139.77
AIC=-275.55   AICc=-275.31   BIC=-271.61
```

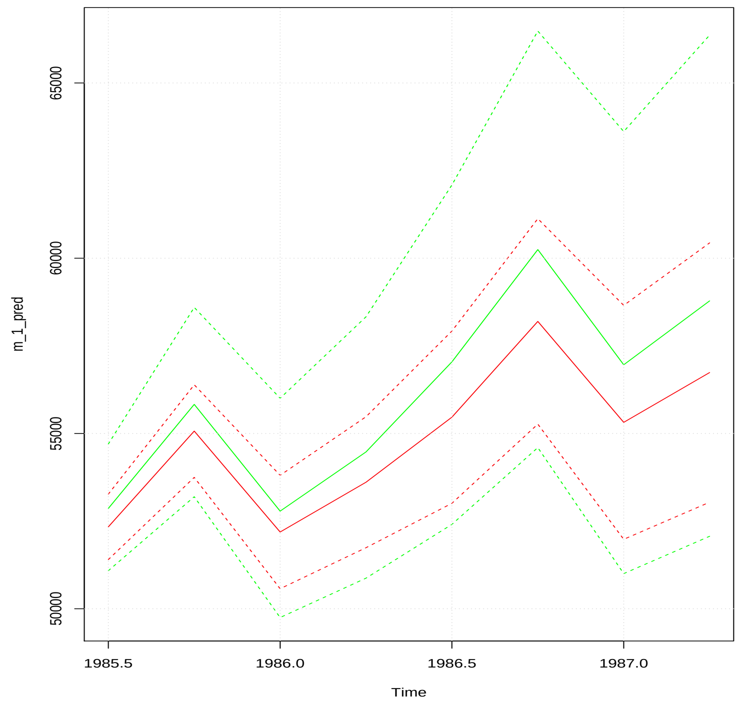

Figure 13: Predictions for eight quarters ahead using two models for the `consumption` data set.

Problem 5 (forecasting)

In Figure 13 I plot the predictions (the solid lines) and two-sigma error bars (the dashed lines) on these predictions for the next eight quarters. The red curves are for the `consumption` data directly and the `green` curves are for the logarithm model. Notice that the log model predicts larger numbers with a larger confidence interval.

Problem 6 (fitting VAR models)

Part (a): For this part we have $p = 1$ and the matrix Φ_1 is given by

```
           r       y         pi
r   0.2139366 17.2841 -0.0304881
y  -0.0001014  0.2206  0.0001002
pi  0.1846333 16.1098 -0.2487303
```

Part (b): The estimate of the covariance of the ϵ_t is given by

```
          r         y        pi
```

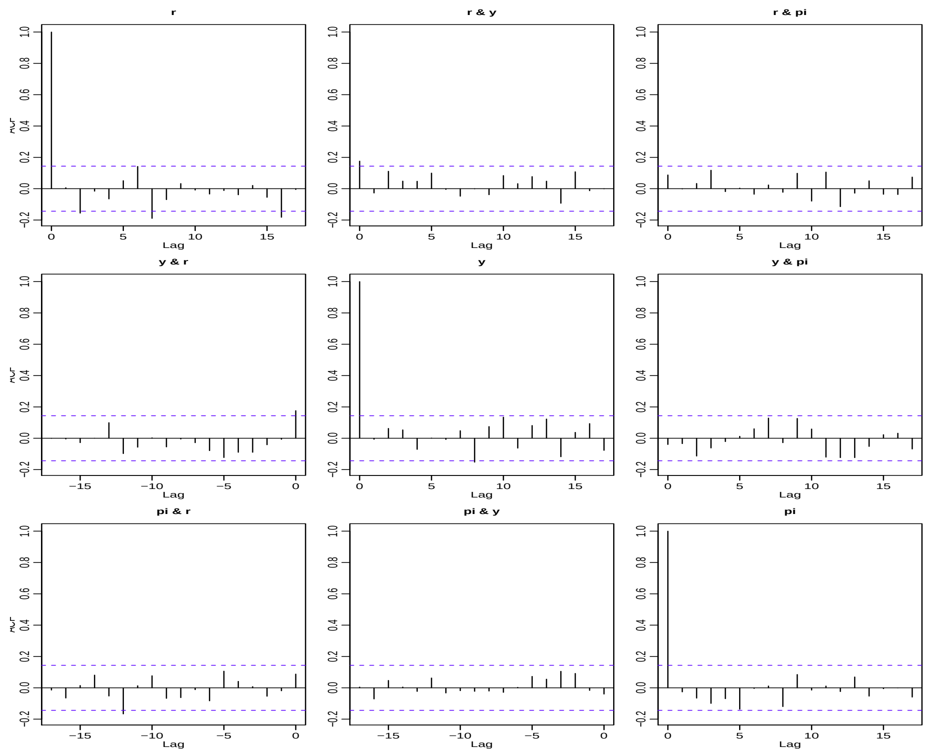

Figure 14: The output from the `acf` on the residuals of the VAR(1) model fit to the time series in the `Tbrate` data set. Note that all auto and cross-correlations are quite small.

```
r  0.807888  0.001858  0.150369
y  0.001858  0.000138 -0.001082
pi 0.150369 -0.001082  3.682619
```

Part (c): The output from the `acf` function is given in Figure 14. From that plot we don't see any spikes above the two sigma standard error lines. We see very few spikes above these lines and the few that are could be due to statistical fluctuations and not real effects.

Problem 7 (predicting with VAR models)

To predict with the model that we extracted from the `ar` function we need to convert the given data into a `ts` object to pass to the `predict` function. I'll assume that the time stamps to associate with the new data occur after the given training data set. Then since the last five samples from the end of the `del_dat` are given by

```
1995 Q4 -0.491  0.002322 -1.64
1996 Q1 -0.834  0.003435  0.41
1996 Q2 -0.438  0.003553  1.76
1996 Q3 -0.561  0.008224 -0.82
```

```
1996 Q4 -1.188  0.007138  0.41
```

Using this the new data we want to predict with can be created using the `ts` command as

```
df = data.frame( r, y, pi )
newdata = ts( df, start=c(1997,1), end=c(1997,3), frequency=4 )
```

Here the vectors `r`, `y`, and `pi` hold the scalar (non vector) time series for each variable given in the text. If we display the resulting `newdata` variable we get

```
> newdata
            r         y    pi
1997 Q1 -1.41 -0.019420  2.31
1997 Q2 -0.48  0.015147 -1.01
1997 Q3  0.66  0.003303  0.31
```

showing that it appends conformally at the end of the training data set. We can then use the `predict` function on the output from the `ar` function and get

```
> predict(var1, newdata=newdata )
$pred
                 r           y          pi
1997 Q4 0.03271604 0.008220368 -0.05135386
```

with an error messages saying that the standard error of predictions for multivariate AR models is not implemented.

Problem 8 (long term memory `DiffSqrtCpi`)

A plot of the time series and the autocorrelation function (ACF) for the suggested transformations of the Consumer Price Index (CPI) data is given in Figure 15. Notice that the ACF decays very slowly and its value is not strictly decreasing (there are periods where it increases as we move from left to right). This is characteristic of perhaps an AR(p) model or the need for fractional differencing (i.e. a long term memory process).

Problem 9 (fractional differencing)

Applying the code `fracdiff` to estimate the amount of fractional differencing we get $d = 0.4104047$ applying this amount of differencing and then plotting the ACF we get Figure 16. Notice that there are no significant spikes above the two sigma horizontal lines indicating that we have found a reasonable fit.

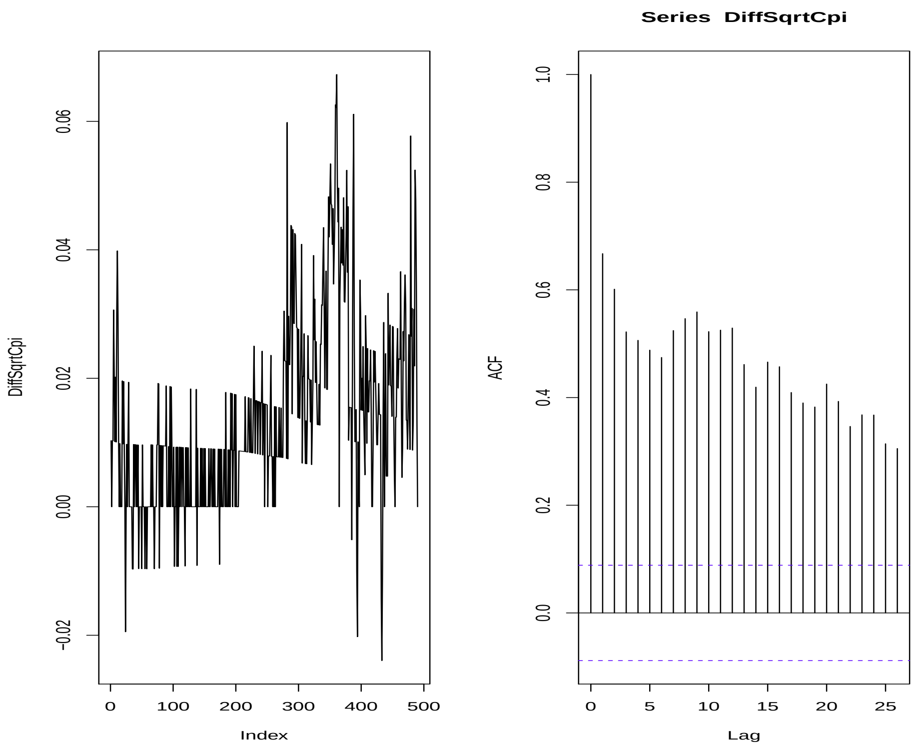

Figure 15: **Left:** A plot of the time series `DiffSqrtCpi`. **Right:** A plot of the autocorrelation function for the time series `DiffSqrtCpi`.

Problem 10 (ARIMA model on `fdiff`)

If we run `auto.arima` on the time series `fdiff` we see that this routine will choose to model this with an ARIMA(0,1,1) model.

```
> auto.arima(fdiff,max.P=0,max.Q=0,ic="aic")
Series: fdiff
ARIMA(0,1,1)

Coefficients:
          ma1
      -0.9857
s.e.   0.0093

sigma^2 estimated as 0.0001029:  log likelihood=1549.4
AIC=-3094.79   AICc=-3094.77   BIC=-3086.41
```

Using BIC does not give a different result. This is a bit strange since looking at the ACF of the raw time series it looks like there are no significant autocorrelations. If we look at the ACF for the first difference of the `fdiff` time series we do indeed see a very significant spike

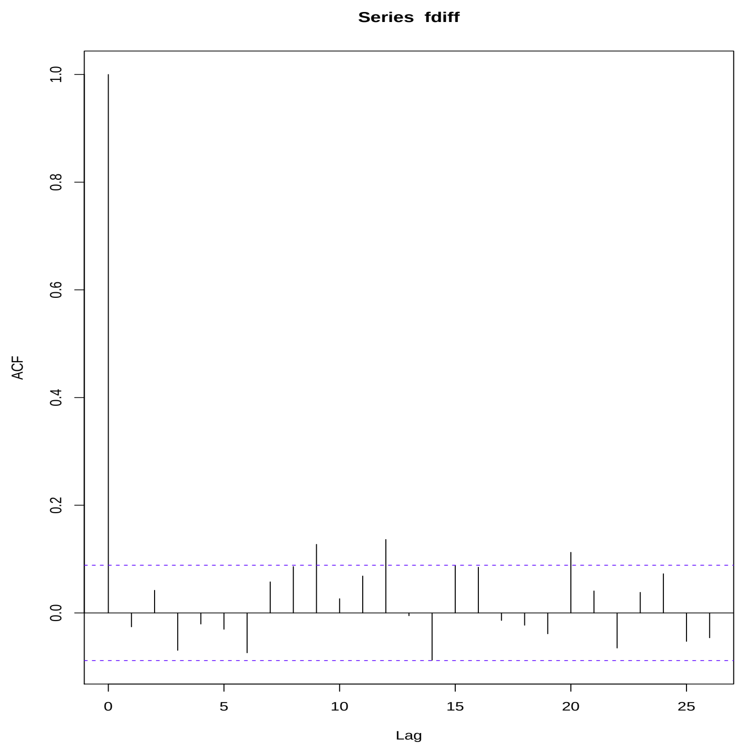

Figure 16: The ACF of the fractionally differenced time series.

at lag one. What is strange is that forcing `auto.arima` to not consider any differences we get a result that has lower AIC and BIC metrics and should be the preferable model.

```
> auto.arima(fdiff,d=0,max.P=0,max.Q=0,ic="aic")
Series: fdiff
ARIMA(0,0,0) with zero mean

sigma^2 estimated as 0.0001028:  log likelihood=1554.56
AIC=-3107.11   AICc=-3107.1   BIC=-3102.92
```

This is a case of spurious differencing (more differencing than is needed). Given these results it seems prudent to model `fdiff` using the simplest model (that of no additional differencing).

Problem 11 (selecting the correct ARIMA model)

Running the given code gives an ARIMA(2,1,0) model (as also stated in the text)

```
> auto.arima(price,ic="bic")
```

```
Series: price
ARIMA(2,1,0)

Coefficients:
         ar1     ar2
      0.2825  0.0570
s.e.  0.0407  0.0408

sigma^2 estimated as 9.989:  log likelihood=-1570.11
AIC=3146.23   AICc=3146.27   BIC=3159.47
```

Running the given bootstrap code in the text we find that of the 10 runs we only get an ARIMA(2,1,0) *once*. Even in that single case the estimate of ϕ_2 is quite different than that used to generate the data (0.1759 vs. 0.0570). Thus it is safe to say that *none* of the bootstrap samples found the correct model.

Problem 12 (estimating the correct parameters)

We next want to determine how well we can estimate the given AR(2) parameters under the assumption that we know the correct model i.e. that our data is coming from an ARIMA(2,1,0) model. When we run the suggested `R` code from the book, for the biases, standard deviations, and MSE we get

```
                  [,1]        [,2]
bias     0.0001361095 0.005447920
std_devs 0.0427685740 0.044955387
mses     0.0027049221 0.002843228
```

The first column has statistics of the estimate for ϕ_1 while the second column holds statistics for the estimates of ϕ_2. We see that the bias in the estimate of ϕ_2 is larger than that in the estimate of ϕ_1. We commented on how different the estimate of ϕ_2 was in the single case where `auto.arima` estimated the correct model in the previous problem. Note that the standard error of the two estimate is about the same. This leads one to conclude that the estimate of ϕ_2 (because the bias is so large) will in general not be correct.

Exercises

See the `R` script `chap_10.R` for the implementation of the exercises for this chapter.

Exercise 10.1

From the given picture is is clear that we need *both* a seasonal and a nonseasonal difference to make this data stationary. Both differences gives autocorrelations with large spikes beyond the two-sigma error lines.

Exercise 10.2

In this case the seasonal difference gives an ACF that has only a few (two) spikes beyond the horizontal two-sigma lines. In this case we could probably fit a seasonal difference of the time series with a MA(2) model.

Exercise 10.3

In this case a single nonseasonal difference gives an ACF that has only a few (two) spikes beyond the two-sigma lines. Thus we could model the first nonseasonal difference using a MA(2) model.

Exercise 10.4

A two-dimensional AR(1) model for the variables ΔCPI and ΔIP is written in the form

$$\left[\begin{array}{c} \Delta\text{CPI}_t \\ \Delta\text{IP}_t \end{array}\right] - \left[\begin{array}{c} \mu_1 \\ \mu_2 \end{array}\right] = \Phi_1 \left(\left[\begin{array}{c} \Delta\text{CPI}_{t-1} \\ \Delta\text{IP}_{t-1} \end{array}\right] - \left[\begin{array}{c} \mu_1 \\ \mu_2 \end{array}\right] \right) + \left[\begin{array}{c} \epsilon_t^1 \\ \epsilon_t^2 \end{array}\right] .$$

Using the given numbers in the book (in gory detail) our prediction of the vector at $t+1$ becomes

$$\left[\begin{array}{c} \Delta\text{CPI}_{t+1} \\ \Delta\text{IP}_{t+1} \end{array}\right] = \left[\begin{array}{c} 0.00518 \\ 0.00215 \end{array}\right] + \left[\begin{array}{cc} 0.767 & 0.0112 \\ -0.33 & 0.3014 \end{array}\right] \left(\left[\begin{array}{c} 0.00173 \\ 0.00591 \end{array}\right] - \left[\begin{array}{c} 0.00518 \\ 0.00215 \end{array}\right] \right) = \left[\begin{array}{c} 0.002575962 \\ 0.004421764 \end{array}\right] .$$

The next prediction is done the same way but using the prediction we just made for the state vector. That is

$$\left[\begin{array}{c} \Delta\text{CPI}_{t+2} \\ \Delta\text{IP}_{t+2} \end{array}\right] = \left[\begin{array}{c} 0.00518 \\ 0.00215 \end{array}\right] + \left[\begin{array}{cc} 0.767 & 0.0112 \\ -0.33 & 0.3014 \end{array}\right] \left(\left[\begin{array}{c} 0.002575962 \\ 0.004421764 \end{array}\right] - \left[\begin{array}{c} 0.00518 \\ 0.00215 \end{array}\right] \right) = \left[\begin{array}{c} 0.003208147 \\ 0.003694042 \end{array}\right] .$$

If we compare this with the results from figure 10.8 in the book we see that the numbers computed here agree with the ones plotted on that graph.

Exercise 10.5

In the `R` code `chap_10.R` we first plot this time series. This plot can be seen in Figure 17. From that plot we see a steady increase in the value of `income`. Next we look at the ACF

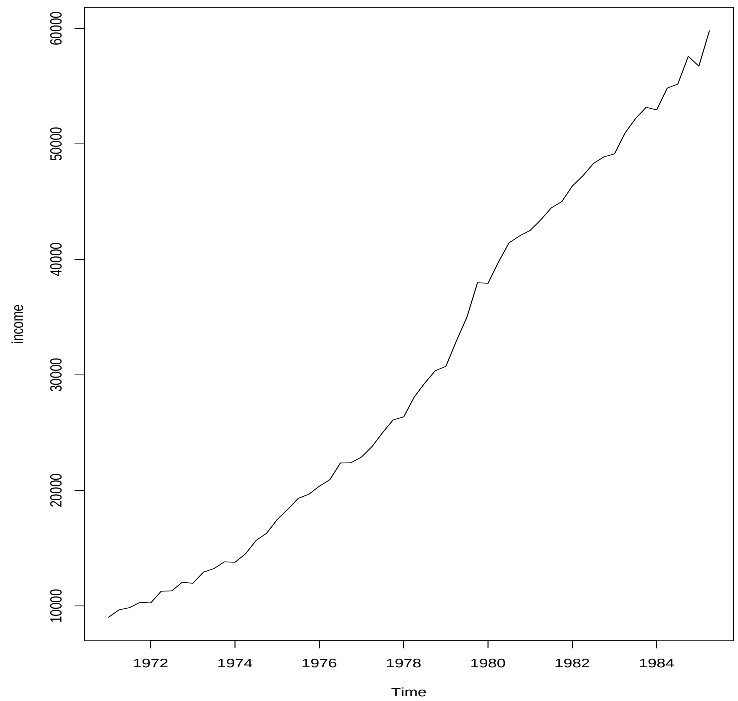

Figure 17: A plot of the `income` time series.

of various differences of this time series. In Figure 18 we plot several of these (for various amounts of differencing). From that plot it looks like either a single seasonal difference or a seasonal difference combined with a single nonseasonal difference would be the two models to consider.

- If we consider the single seasonal difference it looks like the resulting time series would best be modeled with an AR(1) model or at least the ACF has a decaying that looks like it might be exponential (consistent with AR models).
- If we consider the combined seasonal and nonseasonal differencing then the resulting series would best be modeled perhaps with a MA(2) model (or no additional model beyond differencing at all).

From these plots it looks like the most parsimonious model (to me) is given by seasonal differencing followed by nonseasonal differencing. It is interesting to compare this hypothesis with what the `auto.arima` would give. One run of that function gives

```
> auto.arima( income, ic="aic" )
Series: income
ARIMA(2,1,2)(0,1,0)[4]

Coefficients:
```

```
          ar1      ar2     ma1     ma2
      -0.1315  -0.3016  0.0743  0.9129
s.e.   0.2091   0.1571  0.1338  0.1183

sigma^2 estimated as 334423:  log likelihood=-413.54
AIC=837.08   AICc=838.36   BIC=846.93
```

Which arrives at what seems like a relatively complicated model. This model is to be contrasted with the one we get when we take the `ic` option to the `auto.arima` function to the value `bic`. In that case we get

```
> auto.arima( income, ic="bic" )
Series: income
ARIMA(0,1,0)(0,1,0)[4]

sigma^2 estimated as 486420:  log likelihood=-422.22
AIC=846.43   AICc=846.51   BIC=848.4
```

This states that no other modeling (other than taking the differences) is needed. This result seems to have only a marginal increase in the `AIC` and `BIC` metrics but the benefit of being quite a bit simpler. It is also the model we argued for above based on the autocorrelation plots.

Exercise 10.6

Part (a): For this time series each of the differenced autocorrelation function does not look uniformly "flat". This makes modeling this time series difficult since it does not seem to fit nicely into a particular case. Rather than puzzle over the ACF functions in an attempt to figure out what model fits best we will use a data driven approach and let the `auto.arima` function in the `forecast` package give us an estimated model. Running this command on the `unemp` data gives

```
> auto.arima(unemp,ic="aic")
Series: unemp
ARIMA(1,1,0)(0,0,2)[4]

Coefficients:
         ar1     sma1     sma2
      0.6611  -0.4199  -0.2623
s.e.  0.0544   0.0675   0.0660

sigma^2 estimated as 0.07992:  log likelihood=-32.85
AIC=73.69   AICc=73.89   BIC=86.94
```

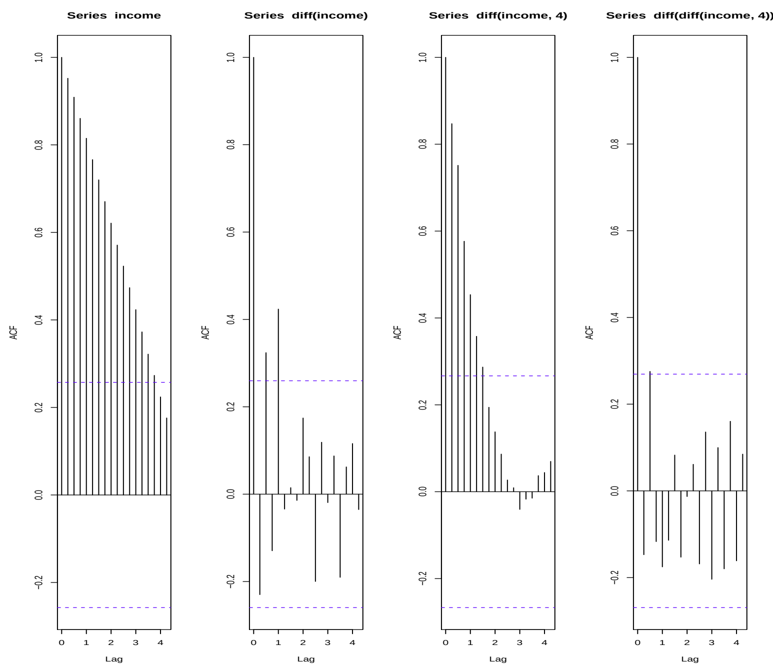

Figure 18: A plot of the ACF for various differences of the `income` time series. From left to right we have (in terms of the amount of differencing): none, nonseasonal, seasonal (4), a seasonal (4) and a nonseasonal difference. Notice that all of the spikes in the fourth plot are within the two sigma horizontal error bars indicating that no additional modeling (other than taking these differences) maybe needed.

As predicted, this gives a somewhat "mixed" model in that it includes effects from several different sources (both seasonal and nonseasonal). We can verify that the residuals show no autocorrelation when using this model by calling the `acf` on the model residuals.

Part (b): For this part we will use the `simulate.Arima` function from the `forecast` library to generate Monte-Carlo samples of the seasonal ARIMA model we fit above. The model we found above (in `R` notation) is given by

```
ARIMA(1,1,0)(0,0,2)[4]
```

Looking at the eight bootstrap samples we see that the correct values of d and D where selected in *none* of the samples. To study this with more Monte-Carlo runs I considered 100 samples (rather than eight). In this case the results were not much better. In all 100 cases the correct model was *never* selected.

Exercise 10.7

For the problem mentioned if we look at Figure 3 (left) we see the ACF plots for all the variables in the `Tbrate` dataframe. From that plot there seem to be a few spikes that look like they might be significant at spaced at seasonal lags. See for example the variable `r` which looks like it might have a spike downwards at around lag "two". Most of these spikes are just on the border between significant and non-significant and as such unless there is strong compelling evidence to include seasonal effects the model is more parsimonious without them. As an example of this in a previous chapter, we forecasted the variable `pi` (the inflation rate) using a nonseasonal ARIMA (see Page 14) where use of `auto.arima` (restricted to nonseasonal models) gave a ARIMA(1,1,1) model with AIC = 772.24 and BIC = 781.94. Running `auto.arima` unrestricted with the two options for `ic` gives (results have been truncated for clarity)

```
> auto.arima( pi, ic="aic" )
ARIMA(1,1,1)(2,0,1)[4]

sigma^2 estimated as 3.432:  log likelihood=-380.97
AIC=773.95   AICc=774.41   BIC=793.33

> auto.arima( pi, ic="bic" )
ARIMA(1,1,1)(2,0,0)[4]

sigma^2 estimated as 3.438:  log likelihood=-381.13
AIC=772.26   AICc=772.59   BIC=788.41
```

In this case the nonseasonal model gives smaller values for AIC and BIC and is the more parsimonious model. Thus it seems prudent to *not* include seasonal effects in modeling `pi`. One could do a similar study for the other two variables.

Chapter 12 (Regression: Basics)

R Lab

See the `R` script `Rlab.R` for this chapter. We plot a pairwise scatter plot of the variables of interest in Figure 19. From that plot we see that it looks like the strongest linear relationship exists between consumption and `dpi` and `unemp`. The variables `cpi` and `government` don't seem to be as linearly related to `consumption`. There seem to be some small outliers in several variables namely: `cpi` (for large values), `government` (large values), and `unemp` (large values). There does not seem to be too much correlation between the variable in that none of the scatter plots seem to look strongly linear and thus there does not look to be collinearity problems.

If we fit a linear model on all four variables we get

```
Call:
lm(formula = consumption ~ dpi + cpi + government + unemp, data = MacroDiff)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.752317   2.520168   5.854 1.97e-08 ***
dpi           0.353044   0.047982   7.358 4.87e-12 ***
cpi           0.726576   0.678754   1.070    0.286
government   -0.002158   0.118142  -0.018    0.985
unemp       -16.304368   3.855214  -4.229 3.58e-05 ***

Residual standard error: 20.39 on 198 degrees of freedom
Multiple R-squared: 0.3385,     Adjusted R-squared: 0.3252
F-statistic: 25.33 on 4 and 198 DF,  p-value: < 2.2e-16
```

The two variables suggested to be the most important above namely `dpi` and `unemp` have the most significant regression coefficients. The `anova` command gives the following

```
> anova(fitLm1)
Analysis of Variance Table

Response: consumption
            Df Sum Sq Mean Sq F value    Pr(>F)
dpi          1  34258   34258 82.4294 < 2.2e-16 ***
cpi          1    253     253  0.6089    0.4361
government   1    171     171  0.4110    0.5222
unemp        1   7434    7434 17.8859 3.582e-05 ***
Residuals  198  82290     416
```

The `anova` table emphasizes the facts that when we add `cpi` and `government` to the regression of `consumption` on `dpi` we don't reduce the regression sum of square significantly enough to make a difference in the modeling. Since two of the variables don't look promising in the modeling of `consumption` we will consider dropping them using `stepAIC` in the `MASS` library. The `stepAIC` suggests that we should first drop `government` and then `cpi` from the regression.

Comparing the AIC for the two models gives that the reduction in AIC is 2.827648 starting with an AIC of 1807.064. This does not seem like a huge change.

The two different `vif` give

```
> vif(fitLm1)
       dpi         cpi  government       unemp
  1.100321    1.005814    1.024822    1.127610
> vif(fitLm2)
     dpi    unemp
1.095699 1.095699
```

Note that after removing the two "noise" variables the variance inflation factors of the remaining two variables decreases (as it should) since now we can determine the coefficients with more precision.

Exercises

Exercise 12.1 (the distributions in regression)

Part (a):

$$Y_i \sim N(1.4 + 1.7, 0.3) = N(3.1, 0.3)\,.$$

To compute $P(Y_i \leq 3|X_i = 1)$ in `R` this would be `pnorm( 3, mean=3.1, sd=sqrt(0.3) )` to find 0.4275661.

Part (b): We can compute the density of $P(Y_i = y)$ as

$$\begin{aligned} P(Y_i = y) &= \int_{-\infty}^{\infty} P(Y_i = y|X)P(X)dX \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left\{-\frac{(y - \beta_0 - \beta_1 x)^2}{2\sigma_1^2}\right\} \left(\frac{1}{\sqrt{2\pi}\sigma_2} \exp\left\{-\frac{x^2}{2\sigma_2^2}\right\}\right) dx \\ &= \frac{1}{\sqrt{2\pi}\sqrt{\sigma_1^2 + \beta_1^2\sigma_2^2}} \exp\left\{-\frac{(y-\beta_0)^2}{2(\sigma_1^2 + \beta_1^2\sigma_2^2)}\right\}, \end{aligned}$$

when we integrate with Mathematica. Here $\sigma_1 = \sqrt{0.3}$ and $\sigma_2 = \sqrt{0.7}$. Thus this density is another normal density and we can evaluate the requested probability using the cumulative normal density function.

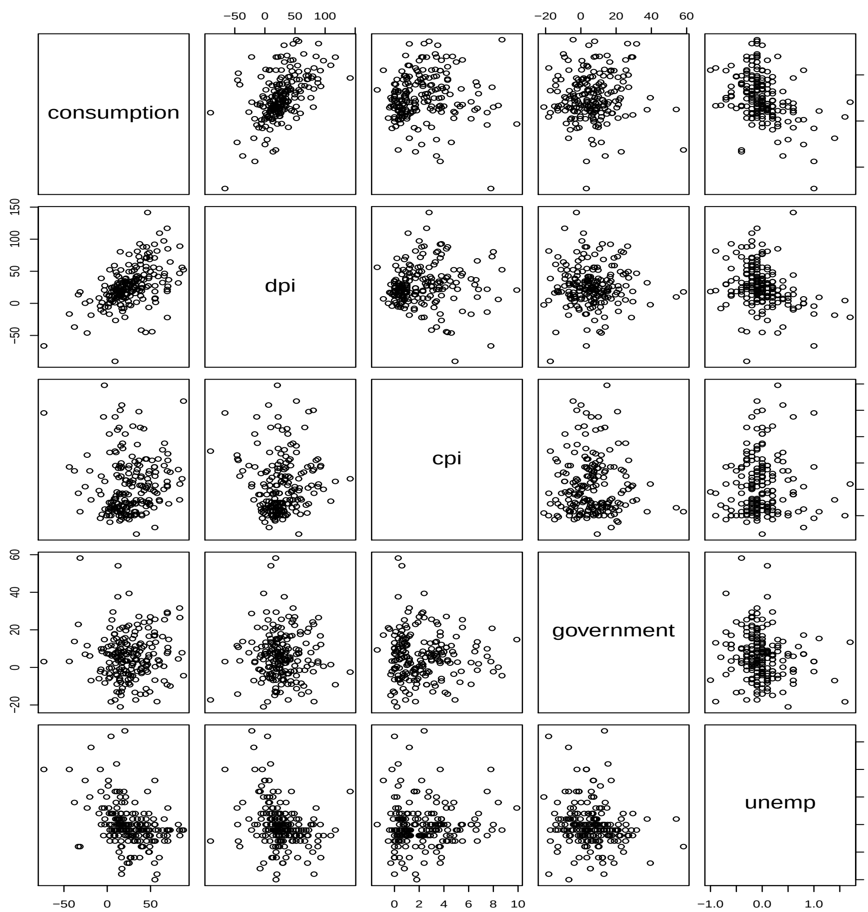

Figure 19: The `pairs` plot for the suggested variables in the `USMacroG` dataset.

Exercise 12.2 (least squares is the same as maximum likelihood)

Maximum likelihood estimation would seek parameters β_0 and β_1 to maximize the log-likelihood of the parameters given the data. For the assumptions in this problem this becomes

$$\begin{aligned}
\text{LL} &= \log\left(\prod_{i=1}^{N} p(Y_i|X_i)\right) = \sum \log p(Y_i|X_i) \\
&= \sum \log\left(\frac{1}{\sqrt{2\pi}\sigma_\epsilon}\exp\left\{-\frac{(y_i-\beta_0-\beta_1 x_i)^2}{2\sigma_\epsilon^2}\right\}\right) \\
&= \text{a constant} - \frac{1}{2\sigma_\epsilon^2}\sum_i (y_i-\beta_0-\beta_1 x_i)^2 \,.
\end{aligned}$$

This later summation expression is what we are minimizing when we perform least-squares minimization.

Exercise 12.4 (the VIF for centered variables)

In the R code chap_12.R we perform the requested experiment and if we denote the variable $X - \bar{X}$ as V we find

```
[1] "cor(X,X^2)= 0.974"
[1] "cor(V,V^2)= 0.000"
[1] "VIF for X and X^2= 9.951"
[1] "VIF for (X-barX) and (X-barX)^2= 1.000"
```

Thus we get a very large reduction in the variance inflation factor when we center our variable.

Exercise 12.5 (the definitions of some terms in linear regression)

In this problem we are told that $n = 40$ and that the empirical correlation $r(Y, \hat{Y}) = 0.65$. Using these facts and the definitions provided in the text we can compute the requested expressions.

Part (a): $R^2 = r^2_{Y\hat{Y}} = (0.65)^2 = 0.4225$

Part (b): From the definition of R^2 we can write

$$R^2 = 1 - \frac{\text{residual error SS}}{\text{total SS}} \,. \tag{6}$$

Since we know the value of R^2 and that the total sum of squares, given by,

$$\text{total SS} = \sum_i (Y_i - \bar{Y})^2 \,,$$

is 100 we can solve Equation 6 for the residual sum of square. We find we have a residual error sum of squares given by 57.75.

Part (c): Since we can decompose the total sum of squares into the regression and residual sum of squares as

$$\text{total SS} = \text{regression SS} + \text{residual SS} \,, \tag{7}$$

and we know the values of the total sum of squares and the residual sum of squares we can solve for the regression sum of squares, in that

$$100 = \text{regression SS} + 57.75 \,.$$

Thus regression SS = 42.25.

Part (d): We can compute s^2 as

$$s^2 = \frac{\text{residual error SS}}{\text{residual degrees of freedom}} = \frac{57.75}{n - 1 - p} = \frac{57.75}{40 - 1 - 3} = 1.604167 \,.$$

Exercise 12.6 (model selection with R^2 and C_p)

For this problem we are told that $n = 66$ and $p = 5$. We will compute several metrics used to select which of the models (the value of the number of predictors or p) one should use in the final regression. The metrics we will consider include

$$R^2 = 1 - \frac{\text{residual error SS}}{\text{total SS}} \tag{8}$$

$$\text{Adjusted } R^2 = 1 - \frac{(n-p-1)^{-1}\text{residual error SS}}{(n-1)^{-1}\text{total SS}} \tag{9}$$

$$C_p = \frac{\text{SSE}(p)}{\hat{\sigma}^2_{\epsilon,M}} - n + 2(p+1) \quad \text{where} \tag{10}$$

$$\text{SSE}(p) = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \quad \text{and}$$

$$\hat{\sigma}^2_{\epsilon,M} = \frac{1}{n - 1 - M} \sum_{i=1}^{M} (Y_i - \hat{Y}_i)^2 \,.$$

Here $\hat{\sigma}^2_{\epsilon,M}$ is the estimated residual variance using *all* of the $M = 5$ predictors, and SSE(p) is computed using values for $\hat{Y}_i$ produced under the model with $p < M$ predictors. From the numbers given we compute it to be 0.1666667. Given the above when we compute the three model selection metrics we find

```
> print( rbind( R2, Adjusted_R2, Cp ) )
                  [,1]      [,2]      [,3]
R2           0.7458333 0.7895833 0.7916667
Adjusted_R2  0.7335349 0.7757855 0.7743056
Cp          15.2000000 4.6000000 6.0000000
```

To use these metrics in model selection we would want to maximize R^2 and the adjusted R^2 and minimize C_p. Thus the R^2 metric would select $p = 5$, the adjusted R^2 metric would select $p = 4$, and the C_p metric would select $p = 4$.

Exercise 12.7 (high p-values)

The p-values reported by `R` are computed under the assumption that the other predictors are still in the model. Thus the large p-values indicate that given X is in the model X^2 does not seem to help much and vice versa. One would need to study the model with either X or X^2 as the predictors. Since X and X^2 are highly correlated one might do better modeling if we subtract the mean of X from all samples i.e. take as predictors $(X - \bar{X})$ and $(X - \bar{X})^2$ rather than X and X^2.

Exercise 12.8 (regression through the origin)

The least square estimator for β_1 is obtained by finding the value of $\hat{\beta}_1$ such that the given $\text{RSS}(\beta_1)$ is minimized. Taking the derivative of the given expression for $\text{RSS}(\hat{\beta}_1)$ with respect to $\hat{\beta}_1$ and setting the resulting expression equal to zero we find

$$\frac{d}{d\hat{\beta}_1}\text{RSS}(\hat{\beta}_1) = 2\sum(Y_i - \hat{\beta}_1 X_i)(-X_i) = 0\,,$$

or

$$-\sum Y_i X_i + \hat{\beta}_1 \sum X_i^2 = 0\,.$$

Solving this expression for $\hat{\beta}_1$ we find

$$\hat{\beta}_1 = \frac{\sum X_i Y_i}{\sum X_i^2}\,. \tag{11}$$

To study the bias introduced by this estimator of β_1 we compute

$$E(\hat{\beta}_1) \;=\; \frac{\sum X_i E(Y_i)}{\sum X_i^2} = \beta_1 \frac{\sum X_i^2}{\sum X_i^2} = \beta_1\,,$$

showing that this estimator is unbiased. To study the variance of this estimator we compute

$$\begin{aligned}
\text{Var}(\hat{\beta}_1) &= \frac{1}{(\sum X_i^2)^2}\sum_i \text{Var}(X_i Y_i) = \frac{1}{(\sum X_i^2)^2}\sum_i X_i^2 \text{Var}(Y_i) \\
&= \frac{\sigma^2}{(\sum X_i^2)^2}\sum_i X_i^2 = \frac{\sigma^2}{\sum_i X_i^2}\,, \qquad (12)
\end{aligned}$$

the requested expression. An estimate of $\hat{\sigma}$ is given by the usual

$$\hat{\sigma}^2 = \frac{\text{RSS}}{n-1} ,$$

which has $n-1$ degrees of freedom.

Exercise 12.9 (filling in the values in an ANOVA table)

To solve this problem we will use the given information to fill in the values for the unknown values. As the total degrees of freedom is 15 the number of points (not really needed) must be one more than this or 16. Since our model has two slopes the degrees of freedom of the regression is 2. Since the degrees of freedom of the regression (2) and the error must add to the total degrees of freedom (15) the degrees of freedom of the error must be $15 - 2 = 13$.

The remaining entries in this table are computed in the `R` code `chap_12.R`.

Exercise 12.10 (least squares with a t-distribution)

For this problem in the `R` code `chap_12.R` we generate data according to a model where y is linearly related to x with an error distribution that is t-distributed (rather than the classical normal distribution). Given this working code we can observe its performance and match the outputs with the outputs given in the problem statement. We find

Part (a): This is the second number in the `mle$par` vector or 1.042.

Part (b): Since the degrees-of-freedom parameter is the fourth one the standard-error of it is given by the fourth number from the output from the `sqrt(diag(FishInfo))` or 0.93492.

Part (c): This would be given by combining the mean and the standard error for the standard deviation estimate or

$$0.152 \pm 1.96(0.01209) = (0.1283036, 0.1756964) .$$

Part (d): Since `mle$convergence` had the value of 0 the optimization converged.

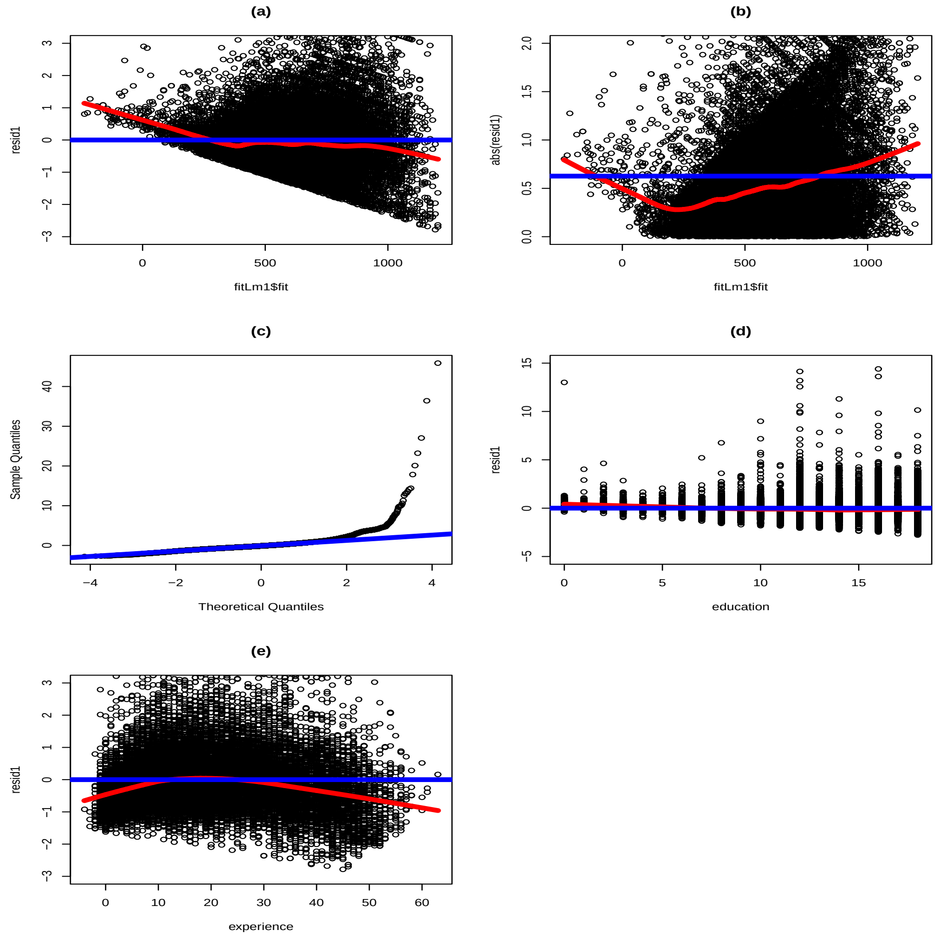

Figure 20: Several regression diagnostics plots for the CPS1988 dataset.

Chapter 13 (Regression: Troubleshooting)

R Lab

See the R script Rlab.R for this chapter. To make the plots more visible I had to change the y limits of the suggested plots. When these limits are changed we get the sequence of plots shown in Figure 20. The plots (in the order in which they are coded to plot) are given by

- The externally studentized residuals as a function of the fitted values which is used to look for heteroscedasticity (non constant variance).
- The *absolute value* of the externally studentized residuals as a function of the fitted values which is used to look for heteroscedasticity.
- The qqplot is used to look for error distributions that are skewed or significantly non-normal. This might suggest applying a log or square root transformation to the response Y to try and make the distribution of residuals more Gaussian.

- Plots of the externally studentized residuals as a function of the variable `education` which can be used to look for nonlinear regression affects in the given variable.
- Plots of the externally studentized residuals as a function of the variable `experience` which can be used to look for nonlinear regression affects in the given variable.

There are a couple of things of note from this plot. The most striking item in the plots presented is in the qqplot. The right limit of the qqplot has a large deviation from a streight line. This indicates that the residuals are not normally distributed and perhaps a transformation of the response will correct this.

We choose to apply a log transformation to the response `wage` and *not* to use `ethnicity` as a predictor (as was done in the previous part of this problem). When we plot the same diagnostic plots as earlier (under this new model) we get the plots shown in Figure 21. The qqplot in this case looks "more" normal (at least both tails of the residual distribution are more symmetric). The distribution of residuals still has heavy tails but certainly not as severe as they were before (without the log transformation). After looking at the plots in Figure 21 we see that there are still non-normal residuals. We also see that it looks like there is a small nonlinear affect in the variable `experience`. We could fit a model that includes this term. We can try a model of `log(wage)` with a quadratic term. When we do that, and then reconsider the diagnostic plots presented so far we get the plots shown in Figure 22. We can then add in the variable `ethnicity` and reproduce the same plots be have been presenting previously. These plots look much like the last ones presented.

Exercises

Exercise 13.1

Some notes on the diagnostic plots are

- From Plot (a) there should be a nonlinear term in x added to the regression.
- From Plot (b) we have some heteroscedasticity in that it looks like we have different values of variance for small and larger values of $\hat{y}$.
- From Plot (c) there might be some heavy tails and or some outliers.
- From Plot (d) it looks like we have autocorrelated errors.
- From Plot (f) we might have some outliers (samples 1 and 100).

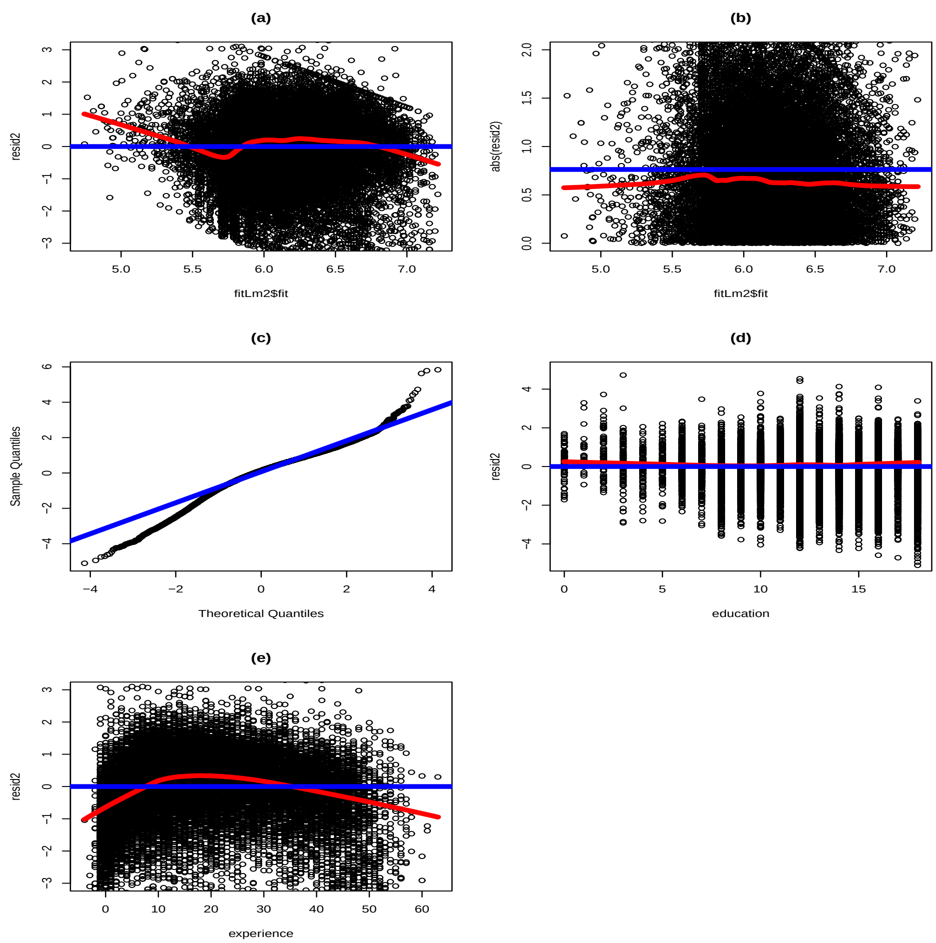

Figure 21: Several regression diagnostic plots for the `CPS1988` dataset where we apply a log transformation to the response.

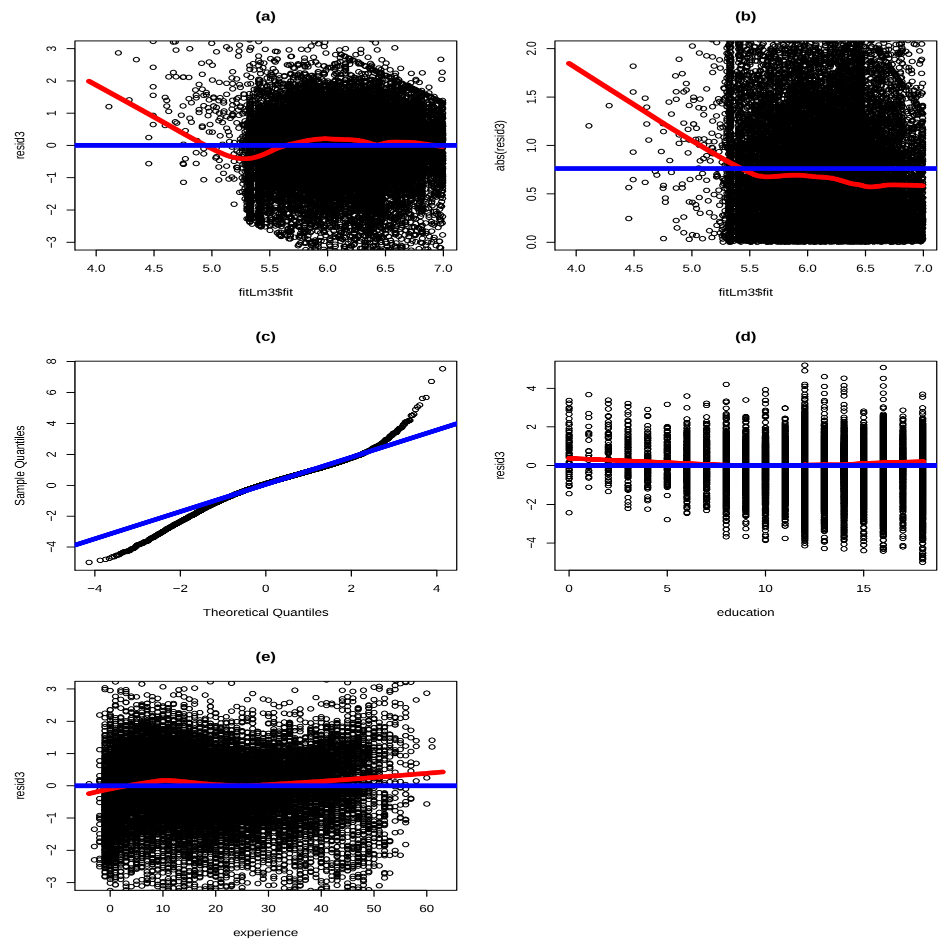

Figure 22: Several regression diagnostic plots for the `CPS1988` dataset where we apply a log transformation to the response and model with a quadratic term in `experience` (as well as a linear term).

Exercise 13.2

Most of these plots seem to emphasis an outlier (the sample with index 58). This sample should be investigated and most likely removed.

Exercise 13.3

Some notes on the diagnostic plots are

- From Plot (a) there should be a nonlinear term in x added to the regression.
- From Plot (b) we don't have much heteroscedasticity i.e. the residual variance looks uniform.
- From Plot (d) it looks like we have autocorrelated residual errors.

Exercise 13.4

Some notes on the diagnostic plots are

- From Plot (a) there is perhaps a small nonlinear term in x that could be added to the regression.
- From Plot (c) we see that the distribution of the residuals have very large tails. Thus we might want to consider taking a logarithmic or a square root transformation of the response Y.
- From Plot (f) it looks like there are two samples (89 and 95) that could be outliers.

Chapter 14 (Regression: Advanced Topics)

Notes on the Text

The maximum likelihood estimation of σ^2

To evaluate what σ is once β has been computed, we take the derivative of L^{GAUSS} with respect to σ, set the result equal to zero, and then solve for the value of σ. For the first derivative of L^{GAUSS} we have

$$\frac{\partial L^{\text{GAUSS}}}{\partial \sigma} = \frac{n}{\sigma} - \sum_{i=1}^{n} \left(\frac{Y_i - x_i^T \beta}{\sigma} \right) \left(\frac{Y_i - x_i^T \beta}{\sigma^2} \right) .$$

Setting this expression equal to zero (and multiply by σ) we get

$$n - \frac{1}{\sigma^2} \sum_{i=1}^{n} (Y_i - x_i^T \beta)^2 = 0 .$$

Solving for σ then gives

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (Y_i - x_i^T \beta)^2 .$$

Notes on the best linear prediction

If we desire to estimate Y with the linear combination $\beta_0 + \beta_1 X$ then to compute β_0 and β_1 we seek to minimize $E((Y - (\beta_0 - \beta_1 X))^2)$. This can be expanded to produce a polynomial in these two variables as

$$\begin{aligned} E((Y - (\beta_0 - \beta_1 X))^2) &= E(Y^2 - 2Y(\beta_0 + \beta_1 X) + (\beta_0 + \beta_1 X)^2) \\ &= E(Y^2 - 2\beta_0 Y - 2\beta_1 XY + \beta_0^2 + 2\beta_0\beta_1 X + \beta_1^2 X^2) \\ &= E(Y^2) - 2\beta_0 E(Y) - 2\beta_1 E(XY) + \beta_0^2 + 2\beta_0\beta_1 E(X) + \beta_1^2 E(X^2) . \end{aligned} \tag{13}$$

Take the β_0 and β_1 derivatives of this result, and then setting them equal to zero gives

$$0 = -2E(Y) + 2\beta_0 + 2\beta_1 E(X) \tag{14}$$
$$0 = -2E(XY) + 2\beta_0 E(X) + 2\beta_1 E(X^2) , \tag{15}$$

as the two equations we must solve for β_0 and β_1 to evaluate the minimum of our expectation. Writing the above system in matrix notation gives

$$\left[\begin{array}{cc} 1 & E(X) \\ E(X) & E(X^2) \end{array} \right] \left[\begin{array}{c} \beta_0 \\ \beta_1 \end{array} \right] = \left[\begin{array}{c} E(Y) \\ E(XY) \end{array} \right] .$$

Using Cramer's rule we find

$$
\begin{aligned}
\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} &= \frac{1}{E(X^2) - E(X)^2} \begin{bmatrix} E(X^2) & -E(X) \\ -E(X) & 1 \end{bmatrix} \begin{bmatrix} E(Y) \\ E(XY) \end{bmatrix} \\
&= \frac{1}{\text{Var}\,(X)} \begin{bmatrix} E(X^2)E(Y) - E(X)E(XY) \\ -E(X)E(Y) + E(XY) \end{bmatrix} \\
&= \frac{1}{\text{Var}\,(X)} \begin{bmatrix} E(X^2)E(Y) - E(X)E(XY) \\ \text{Cov}\,(X,Y) \end{bmatrix}.
\end{aligned}
$$

Thus we see that

$$
\beta_1 = \frac{\text{Cov}\,(X,Y)}{\text{Var}\,(X)} = \frac{\sigma_{XY}}{\sigma_X^2}. \tag{16}
$$

From Equation 14 we have

$$
\beta_0 = E(Y) - \beta_1 E(X) = E(Y) - \left(\frac{\sigma_{XY}}{\sigma_X^2}\right) E(X). \tag{17}
$$

These are the equations presented in the text.

Notes on the error of the best linear prediction

Once we have specified β_0 and β_1 we can evaluate the expected error in using these values for our parameters. With $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$ and the expressions we computed for β_0 and β_1 when we use Equation 13 we have

$$
\begin{aligned}
E((Y - \hat{Y})^2) &= E(Y^2) - 2\left(E(Y) - \frac{\sigma_{XY}}{\sigma_X^2}E(X)\right)E(Y) - 2\left(\frac{\sigma_{XY}}{\sigma_X^2}\right)E(XY) \\
&+ \left(E(Y) - \frac{\sigma_{XY}}{\sigma_X^2}E(X)\right)^2 + 2\left(E(Y) - \frac{\sigma_{XY}}{\sigma_X^2}E(X)\right)\left(\frac{\sigma_{XY}}{\sigma_X^2}\right)E(X) + \left(\frac{\sigma_{XY}}{\sigma_X^2}\right)^2 E(X^2) \\
&= E(Y^2) - 2E(Y)^2 + 2\frac{\sigma_{XY}}{\sigma_X^2}E(X)E(Y) - 2\left(\frac{\sigma_{XY}}{\sigma_X^2}\right)E(XY) \\
&+ E(Y)^2 - 2\frac{\sigma_{XY}}{\sigma_X^2}E(X)E(Y) + \frac{\sigma_{XY}^2}{\sigma_X^4}E(X)^2 \\
&+ 2\frac{\sigma_{XY}}{\sigma_X^2}E(X)E(Y) - 2\left(\frac{\sigma_{XY}}{\sigma_X^2}\right)^2 E(X)^2 + \left(\frac{\sigma_{XY}}{\sigma_X^2}\right)^2 E(X^2) \\
&= E(Y^2) - E(Y)^2 + 2\frac{\sigma_{XY}}{\sigma_X^2}(E(X)E(Y) - E(XY)) - \frac{\sigma_{XY}^2}{\sigma_X^4}E(X)^2 + \frac{\sigma_{XY}^2}{\sigma_X^4}E(X^2) \\
&= \text{Var}\,(Y) - 2\frac{\sigma_{XY}}{\sigma_X^2}\text{Cov}\,(X,Y) + \frac{\sigma_{XY}^2}{\sigma_X^4}(E(X^2) - E(X)^2) \\
&= \text{Var}\,(Y) - 2\frac{\sigma_{XY}}{\sigma_X^2}\text{Cov}\,(X,Y) + \frac{\sigma_{XY}^2}{\sigma_X^4}\sigma_X^2 = \sigma_Y^2 - \frac{\sigma_{XY}}{\sigma_X^2} \\
&= \sigma_Y^2\left(1 - \frac{\sigma_{XY}^2}{\sigma_X^2\sigma_Y^2}\right).
\end{aligned}
$$

Since $\sigma_{XY} = \sigma_X \sigma_Y \rho_{XY}$ we can write the above as

$$E((Y - \hat{Y})^2) = \sigma_Y^2 \left(1 - \rho_{XY}^2\right) , \tag{18}$$

which is the equation presented in the book. Next we evaluate the expectation of $(Y - c)^2$ for a constant c. We find

$$\begin{aligned} E((Y-c)^2) &= E((Y - E(Y) + E(Y) - c)^2) \\ &= E((Y - E(Y))^2 + 2(Y - E(Y))(E(Y) - c) + (E(Y) - c)^2) \\ &= E((Y - E(Y))^2) + 0 + E((E(Y) - c)^2)) \\ &= \mathrm{Var}(Y) + (E(Y) - c)^2 , \end{aligned} \tag{19}$$

since this last term is a constant (independent of the random variable Y).

R Lab

See the `R` script `Rlab.R` for this chapter where these problems are worked.

Regression with ARMA Noise

When the above `R` code is run it computes the two requested AIC values

```
[1] "AIC(arima fit)=     86.852; AIC(lm fit)=    138.882"
```

and also generates Figure 20. Note that both of these diagnostics indicate that the model that considers autocorrelation of residuals is the preferred model i.e. this would mean the model computed using the `R` command `arima`. I was not exactly sure how to compute the BIC directly but since it is related to the AIC (which the output of the `arima` command gives us) I will compute it using

$$\mathrm{BIC} = \mathrm{AIC} + \{\log(n) - 2\}p .$$

Using this we find

```
[1] "BIC(arima fit)=     96.792; BIC(lm fit)=    145.509"
```

which again specifies that the first model is better. We can fit several ARIMA models and compare the BIC of each. When we do that we find

```
[1] "BIC(arima(2,0,0))=     99.532; BIC(arima(1,0,1))=    100.451; "
```

Thus we see that according to the BIC criterion the ARIMA(1,0,0) model is still the best.

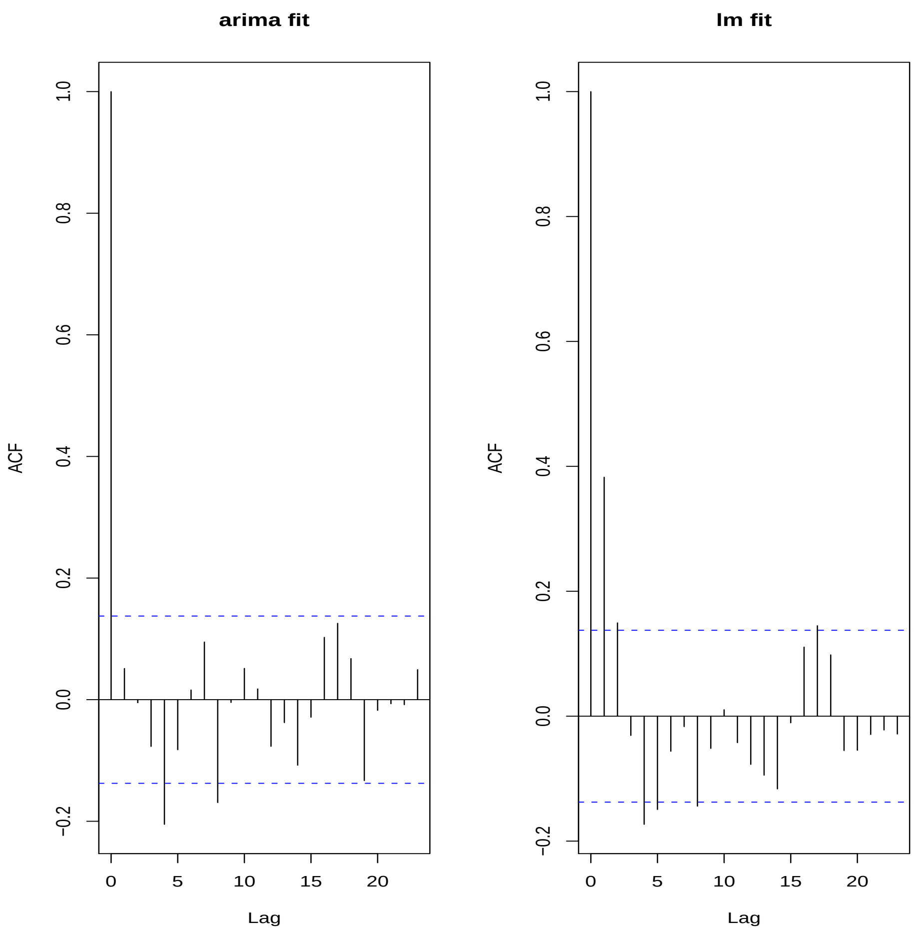

Figure 23: **Left:** Autocorrelation function for the residuals for the `arima` fit of the `MacroDiff` dataset. **Right:** Autocorrelation function for the residuals for the `lm` fit of the `MacroDiff` dataset. Note the significant autocorrelation present at lag one.

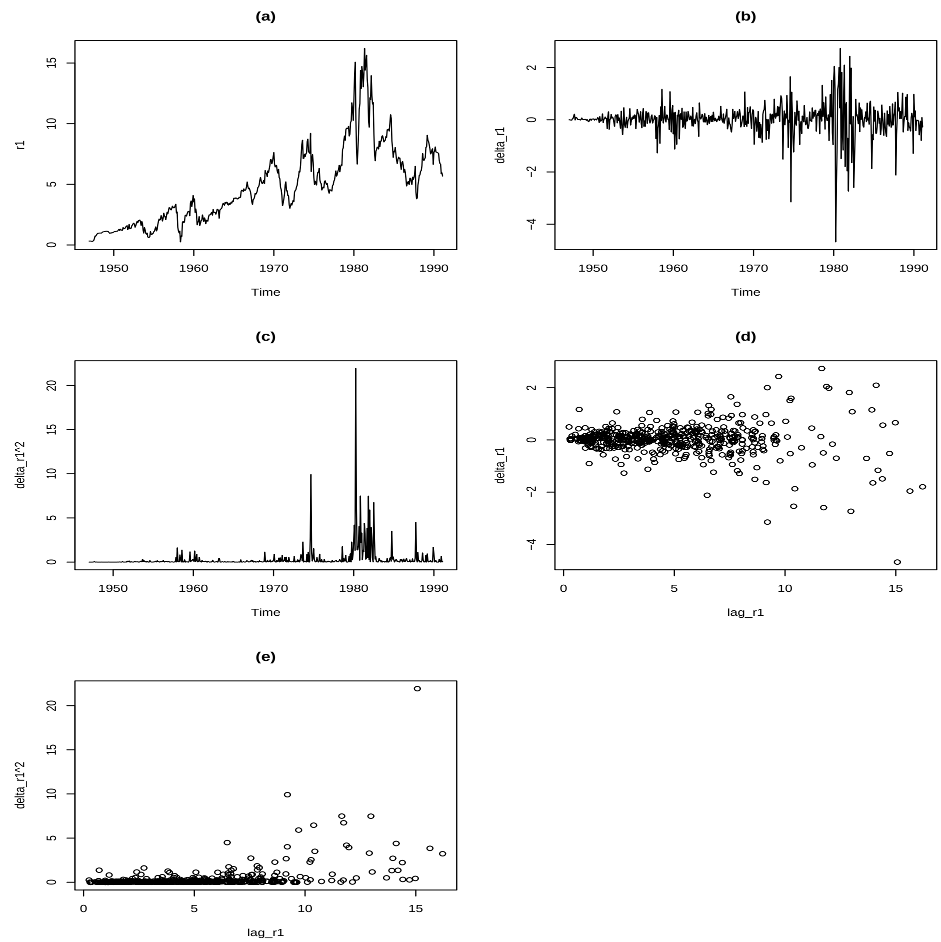

Figure 24: The plots for estimating the short rate models.

Nonlinear Regression

The R command help(Irates) tells us that the r1 column from the Irates dataframe is a ts object of interest rates sampled each month from Dec 1946 until Feb 1991 from the United-States. These rates are expressed as a percentage per year.

When the above R code is run we get the plots shown in Figure 24. These plots are used in building the models for $\mu(t,r)$ and $\sigma(t,r)$. From the plot labeled (d) we see that Δr_t seems (on average) to be relatively constant at least for small values of r_{t-1} i.e. less than 10. For values greater than that we have fewer samples and it is harder to say if a constant would be the best fitting function. From the plot labeled (b) it looks like there are times when Δr_t is larger than others (namely around 1980s). This would perhaps argue for a time dependent μ function. There does not seem to be a strong trend. From the summary command we see that a and θ are estimated as

```
> summary( nlmod_CKLS )
```

```
Formula: delta_r1 ~ a * (theta - lag_r1)

Parameters:
      Estimate Std. Error t value Pr(>|t|)
theta  5.32754    1.33971   3.977 7.96e-05 ***
a      0.01984    0.00822   2.414   0.0161 *
```

Response Transformations

The boxcox function returns x which is a list of the values of α tried and y the value of the loglikelihood for each of these values of α. We want to pick a value of α that maximizes the loglikelihood. Finding the maximum of the loglikelihood we see that it is achieved at a value of $\alpha = 0.1414141$. The new model with Y transformed using the box-cox transform has a much smaller value of the AIC

```
[1] "AIC(fit1)= 12094.187991, AIC(fit3)= 1583.144759"
```

This is a significant reduction in AIC. Plots of the residuals of the box-cox model as a function of the fitted values indicate that there is not a problem of heteroscedasticity. The residuals of this box-cox fit appear to be autocorrelated but since this is not time series data this behavior is probably spurious (not likely to repeat out of sample).

Who Owns an Air Conditioner?

Computing a linear model using *all* of the variables gives that several of the coefficients are not estimated well (given the others in the model). We find

```
> summary(fit1)

Call:
glm(formula = aircon ~ ., family = "binomial", data = HousePrices)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9183  -0.7235  -0.5104   0.6578   3.2650

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)   -3.576e+00  5.967e-01  -5.992 2.07e-09 ***
price          5.450e-05  8.011e-06   6.803 1.02e-11 ***
lotsize       -4.482e-05  6.232e-05  -0.719 0.472060
bedrooms      -6.732e-02  1.746e-01  -0.385 0.699887
```

```
bathrooms     -5.587e-01  2.705e-01  -2.065 0.038907 *
stories        3.155e-01  1.540e-01   2.048 0.040520 *
drivewayyes   -4.089e-01  3.550e-01  -1.152 0.249366
recreationyes  1.052e-01  2.967e-01   0.355 0.722905
fullbaseyes    1.777e-02  2.608e-01   0.068 0.945675
gasheatyes    -3.929e+00  1.121e+00  -3.506 0.000454 ***
garage         6.893e-02  1.374e-01   0.502 0.615841
preferyes     -3.294e-01  2.743e-01  -1.201 0.229886
```

We can use the `stepAIC` in the `MASS` library to sequentially remove predictors. The final step from the `stepAIC` command gives

```
Step:  AIC=539.36
aircon ~ price + bathrooms + stories + gasheat

            Df Deviance    AIC
<none>           529.36 539.36
- bathrooms  1   532.87 540.87
- stories    1   535.46 543.46
- gasheat    1   554.74 562.74
- price      1   615.25 623.25
```

The `summary` command on the resulting linear model gives

```
> summary(fit2)

Call:
glm(formula = aircon ~ price + bathrooms + stories + gasheat,
    family = "binomial", data = HousePrices)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8433  -0.7278  -0.5121   0.6876   3.0753

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.045e+00  4.050e-01  -9.987  < 2e-16 ***
price        4.782e-05  6.008e-06   7.959 1.73e-15 ***
bathrooms   -4.723e-01  2.576e-01  -1.833 0.066786 .
stories      3.224e-01  1.317e-01   2.449 0.014334 *
gasheatyes  -3.657e+00  1.082e+00  -3.378 0.000729 ***
```

Looking at the signs of the coefficients estimated we first see that as `price` and `stories` increases the probability of air conditioning increases which seams reasonable. From the same table, *increasing* `bathrooms` and `gasheatyes` should *decrease* the probability that we have air conditioning. One would not expect that having more bathrooms should decrease our probability of air conditioning. The same might be said for the `gasheatyes` predictor. The difference in AIC between the model suggested and the one when we remove the predictor `bathrooms` is not very large indicating that removing it does not give a very different model. As the sample we are told to look at seems to be the same as the first element in the training set we can just extract that sample and use the `predict` function to evaluate the given model. When we do this (using the first model) we get 0.1191283.

Exercises

Exercise 14.1 (computing β_0 and β_1)

See the notes on Page 54.

Exercise 14.2 (hedging)

The combined portfolio is

$$F_{20}P_{20} - F_{10}P_{10} - F_{30}P_{30}\,.$$

Lets now consider how this portfolio changes as the yield curve changes. From the book we would have that the change in the total portfolio is given by

$$-F_{20}P_{20}\mathrm{DUR}_{20}\Delta y_{20} + F_{10}P_{10}\mathrm{DUR}_{10}\Delta y_{10} + F_{30}P_{30}\mathrm{DUR}_{30}\Delta y_{30}\,.$$

We are told that we have modeled Δy_{20} as

$$\Delta y_{20} = \hat{\beta}_1\Delta y_{10} + \hat{\beta}_2\Delta y_{30}\,.$$

When we put this expression for Δy_{20} into the above (and then group by Δy_{10} and Δy_{30}) we can write the above as

$$(-F_{20}P_{20}\mathrm{DUR}_{20}\hat{\beta}_1 + F_{10}P_{10}\mathrm{DUR}_{10})\Delta y_{10} + (-F_{20}P_{20}\mathrm{DUR}_{20}\hat{\beta}_2 + F_{30}P_{30}\mathrm{DUR}_{30})\Delta y_{30}\,.$$

We will then take F_{10} and F_{30} to be the values that would make the coefficients of Δy_{10} and Δy_{30} both zero. These would be

$$\begin{aligned} F_{10} &= F_{20}\left(\frac{P_{20}\mathrm{DUR}_{20}}{P_{10}\mathrm{DUR}_{10}}\right)\hat{\beta}_1 \\ F_{30} &= F_{20}\left(\frac{P_{20}\mathrm{DUR}_{20}}{P_{30}\mathrm{DUR}_{30}}\right)\hat{\beta}_2\,. \end{aligned}$$

Exercise 14.3 (fitting a yield curve)

We are given the short rate $r(t;\theta)$, which we need to integrate to get the yield $y_t(\theta)$. For the Nelson-Siegel model for $r(t;\theta)$ this integration is presented in the book on page 383. Then given the yield the price is given by

$$P_i = \exp(-T_i y_{T_i}(\theta)) + \epsilon_i \, .$$

I found it hard to fit the model "all at once". In order to fit the model I had to estimate each parameter θ_i in a sequential fashion. See the `R` code `chap_14.R` for the fitting procedure used. When that code is run we get estimate of the four θ parameters given by

```
     theta0      theta1      theta2      theta3
0.009863576 0.049477242 0.002103376 0.056459908
```

When we reconstruct the yield curve with these numbers we get the plot shown in Figure 25.

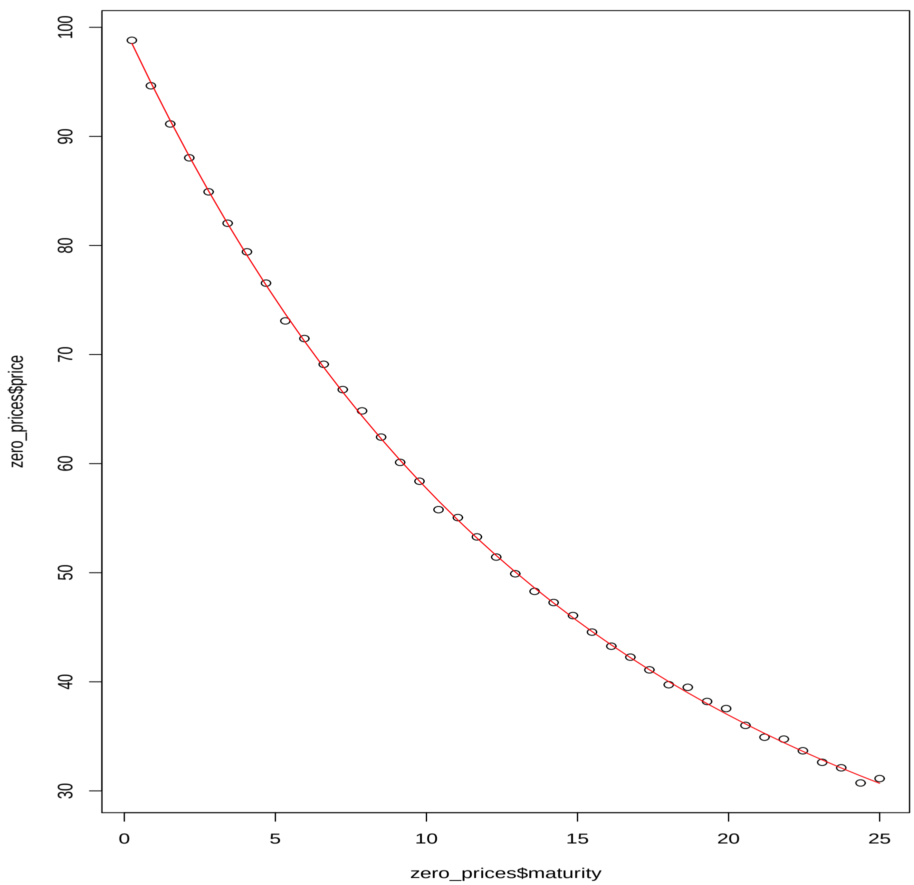

Figure 25: The model (in red) and the discrete bond prices (in black) for Exercise 14.3.

Chapter 15 (Cointegration)

R Lab

See the `R` script `Rlab.R` where the problem for this chapter are worked.

Problem 1

The values of the test statistic output from the `ca.jo` command is given by

```
          test 10pct  5pct 1pct
r <= 9 |  0.60   6.5  8.18 11.7
r <= 8 |  6.31  12.9 14.90 19.2
r <= 7 |  9.20  18.9 21.07 25.8
r <= 6 | 11.79  24.8 27.14 32.1
r <= 5 | 15.32  30.8 33.32 38.8
r <= 4 | 19.89  36.2 39.43 44.6
r <= 3 | 21.65  42.1 44.91 51.3
r <= 2 | 22.98  48.4 51.07 57.1
r <= 1 | 35.62  54.0 57.00 63.4
r = 0  | 58.39  59.0 62.42 68.6
```

To read the output from this command we start at the top and can reject a hypothesis $r \le d$ if the value in the "test" column is larger than the various confidence given in the 10%, 5%, and 1% columns. For this example *none* of the values in the "test" column is larger than the other values in the same row. Thus we can not reject the hypothesis that $r \le d$ for each of the d's given. We are forced to conclude that $r = 0$ or that there are *no* cointegration vectors.

Problem 2

The maturities for the bonds considered is stored in the `mk.maturity` dataframe. Looking at the maturities of the bonds we are considering (in months) for cointegration we find

```
> mk.maturity[2:11,] * 12
 [1]  2  3  4  5  6  7  8  9 10 11
```

Thus these are short-term maturities.

Problem 3

The values of the test statistic output from the `ca.jo` command on this data is given by

```
          test 10pct  5pct 1pct
r <= 9 |  1.43   6.5  8.18 11.7
r <= 8 |  8.03  12.9 14.90 19.2
r <= 7 | 18.48  18.9 21.07 25.8
r <= 6 | 27.87  24.8 27.14 32.1
r <= 5 | 36.12  30.8 33.32 38.8
r <= 4 | 40.96  36.2 39.43 44.6
r <= 3 | 51.87  42.1 44.91 51.3
r <= 2 | 59.25  48.4 51.07 57.1
r <= 1 | 70.71  54.0 57.00 63.4
r = 0  | 90.57  59.0 62.42 68.6
```

Looking at the 1% level we can reject $r \leq 3$ (since $51.87 > 51.3$) but not $r \leq 4$ (since $40.96 \not> 44.6$). Thus we must conclude that $r = 4$. Thus we have four cointegration relationships.

Problem 4-7

Using the simulation written in `Rlab.R` we find that the expected profit is 13238, the probability that we have to liquidate for a loss is given by 0.0966, the expected waiting time is 44.1267 days, and the expected yearly return is 5.267559.

Exercises

Exercise 15.1

Now Eq. 15.4 from the book is given by

$$\Delta(Y_{1,t} - \lambda Y_{2,t}) = (\phi_1 - \lambda\phi_2)(Y_{1,t-1} - \lambda Y_{2,t-1}) + (\epsilon_{1,t} - \lambda\epsilon_{2,t}) \tag{20}$$

When we use the definition of the forward difference Δ we get that the left-hand-side of the above is equal to

$$(Y_{1,t} - \lambda Y_{2,t}) - (Y_{1,t-1} - \lambda Y_{2,t-1}) .$$

When we move $Y_{1,t-1} - \lambda Y_{2,t-1}$ in the above to the right-hand-side of Equation 22 we have the equation

$$Y_{1,t} - \lambda Y_{2,t} = (1 + \phi_1 - \lambda\phi_2)(Y_{1,t-1} - \lambda Y_{2,t-1}) + (\epsilon_{1,t} - \lambda\epsilon_{2,t}) ,$$

which shows that $Y_{1,t-1} - \lambda Y_{2,t-1}$ is an AR(1) process with a coefficient $1 + \phi_1 - \lambda\phi_2$.

Exercise 15.2

Now Eq. 15.2 from the book is given by

$$\Delta Y_{1,t} = \phi_1(Y_{1,t-1} - \lambda Y_{2,t-1}) + \epsilon_{1,t} \,, \tag{21}$$

and Eq. 15.3 from the book is given by

$$\Delta Y_{2,t} = \phi_2(Y_{1,t-1} - \lambda Y_{2,t-1}) + \epsilon_{2,t} \,. \tag{22}$$

If we add constants to them we get the equations

$$\begin{aligned}
\Delta Y_{1,t} &= \phi_1(Y_{1,t-1} - \lambda Y_{2,t-1}) + \mu_1 + \epsilon_{1,t} \\
\Delta Y_{2,t} &= \phi_2(Y_{1,t-1} - \lambda Y_{2,t-1}) + \mu_2 + \epsilon_{2,t} \,.
\end{aligned}$$

From these we find that $\Delta(Y_{1,t} - \lambda Y_{2,t})$ is given by

$$\Delta(Y_{1,t} - \lambda Y_{2,t}) = (\phi_1 - \lambda\phi_2)(Y_{1,t-1} - \lambda Y_{2,t-1}) + (\mu_1 - \lambda\mu_2) + (\epsilon_{1,t} - \lambda\epsilon_{2,t}) \,,$$

or that $Y_{1,t} - \lambda Y_{2,t}$ is given by

$$Y_{1,t} - \lambda Y_{2,t} = (1 + \phi_1 - \lambda\phi_2)(Y_{1,t-1} - \lambda Y_{2,t-1}) + (\mu_1 - \lambda\mu_2) + (\epsilon_{1,t} - \lambda\epsilon_{2,t}) \,,$$

This is an AR(1) model for $Y_{1,t} - \lambda Y_{2,t}$ but it is not yet written in the "standard form" for an AR(1) models (with a mean m) which takes the form

$$Z_t - m = \gamma(Z_{t-1} - m) + \epsilon_t \quad \text{or} \quad Z_t = \gamma Z_{t-1} + m(1-\gamma) + \epsilon_t \,.$$

To find the mean of the AR(1) process for $Y_{1,t} - \lambda Y_{2,t}$ using the second equation for Z_t above we see that the m mean must satisfy

$$m(1-\gamma) = m[1 - (1 + \phi_1 - \lambda\phi_2)] = \mu_1 - \lambda\mu_2 \,.$$

The solution for m is given by

$$m = -\frac{\mu_1 - \lambda\mu_2}{\phi_1 - \lambda\phi_2} \,.$$

The point of this is that the process $Y_{1,t} - \lambda Y_{2,t}$ now has a nonzero mean m (assuming the numerator is nonzero).

Exercise 15.3

In Example 15.2 we were told to take $\phi_1 = 0.5$, $\phi_2 = 0.55$ and $\lambda = 1$. From Exercise 1 above we know that $Y_{1,t} - \lambda Y_{2,t}$ is an AR(1) process with a parameter that has a magnitude of

$$|1 + \phi_1 - \lambda\phi_2| = |1 + 0.5 - 0.55| = 0.95 \,.$$

Since this is less than one we have that $Y_{1,t} - \lambda Y_{2,t}$ is stationary.

Exercise 15.4

Eq. 15.2 and 15.3 together are the vector model

$$\Delta \begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \begin{bmatrix} 1 & -\lambda \end{bmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix} .$$

When we write this as a VAR(1) model for $\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix}$ we get

$$\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{bmatrix} \phi_1 + 1 & -\lambda\phi_1 \\ \phi_2 & -\lambda\phi_2 + 1 \end{bmatrix} \begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} + \begin{bmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{bmatrix} ,$$

To have the vector AR process for Y be stationary means that the coefficient matrix Φ above has all eigenvalues less than one in magnitude. For the numbers we were to use for this problem $\phi_1 = 0.5$, $\phi_2 = 0.55$ and $\lambda = 1$ the coefficient matrix Φ is given by

$$\Phi = \begin{bmatrix} 1.5 & -0.5 \\ 0.55 & 0.45 \end{bmatrix} .$$

This matrix has eigenvalues given by 1 and 0.95. Since on of the eigenvalues has a magnitude of one this process is not technically stationary.

Chapter 16 (The Capital Asset Pricing Model)

R Lab

See the R script Rlab.R where the Rlab problem from this chapter are worked.

Problem 1

We can look at the summary command output for each regression to see if any of the P-values for the constant (in R this is denoted as the Intercept coefficient) of the simple linear regression is sufficiently small. At an intuitive level a very small P-value indicates that a nonzero result is likely to "actually" be nonzero. Scanning the results we find that the *smallest* P-value of the Intercept terms is for F_AC and is the value 0.14. Typical values (for a significant result) need to be at least below 0.05. Thus we can conclude that none of the α terms are nonzero.

Problem 2

Eq. 16.19 from the book is given by

$$R_{j,t} = \mu_{f,t} + \beta_j (R_{M,t} - \mu_{f,t}) + \epsilon_{j,t} \, . \tag{23}$$

Thus the expected excess return of the jth stock is given in terms of the expected excess return of the market by

$$E[R_{j,t} - \mu_{f,t}] = \beta_j E[R_{M,t} - \mu_{f,t}] \, .$$

I took this problem to mean that we will compute the expected excess return of the j stock to be equal to the product of the estimate of β_j (computed earlier) and the excess return of the market (see the previous equation). When we do this we get the following

```
> mean(market) * betas
 GM_AC   F_AC UTX_AC CAT_AC MRK_AC PFE_AC IBM_AC
0.0233 0.0240 0.0190 0.0274 0.0146 0.0184 0.0159
```

These estimates are to be compared against just computing the values for $R_{j,t} - \mu_{f,t}$ and then taking the average of these points. Computing the average in this way gives

```
> apply( stockExRet, 2, mean )
  GM_AC    F_AC  UTX_AC  CAT_AC  MRK_AC  PFE_AC  IBM_AC
-0.0452 -0.0736  0.0485  0.0842  0.0071 -0.0218 -0.0163
```

These results are very different from each other.

Problem 3

Using the `cor` command we get these to be

```
> cor(res)
          GM_AC     F_AC   UTX_AC  CAT_AC  MRK_AC  PFE_AC   IBM_AC
GM_AC   1.00000  0.50911  0.03967  0.0202 -0.0472 -0.0188  0.00785
F_AC    0.50911  1.00000 -0.00714  0.0289  0.0128  0.0114  0.03575
UTX_AC  0.03967 -0.00714  1.00000  0.1498 -0.0154 -0.1110 -0.06949
CAT_AC  0.02023  0.02895  0.14977  1.0000 -0.0757 -0.0650 -0.08342
MRK_AC -0.04715  0.01279 -0.01540 -0.0757  1.0000  0.2833 -0.07817
PFE_AC -0.01877  0.01142 -0.11103 -0.0650  0.2833  1.0000 -0.04606
IBM_AC  0.00785  0.03575 -0.06949 -0.0834 -0.0782 -0.0461  1.00000
```

From the above correlation matrix it looks like some of the tickers have larger correlations with certain other tickers. Some examples are F and GM (both automobile companies) and MRK and PFE (both pharmaceutical companies). The companies UTX and CAT are both in the "industrial goods" sector and have a relatively large correlation also.

Problem 4

Once we have estimated β_j and σ_{ϵ_j} for each security then Eq. 16.19 implies that the covariance matrix for the returns between securities has

$$\sigma_j^2 = \beta_j^2 \sigma_M^2 + \sigma_{\epsilon_j}^2 ,$$

for its diagonal elements and

$$\sigma_{jj'} = \beta_j \beta_{j'} \sigma_M^2 ,$$

for its off diagonal elements. Computing these elements from the given data we get

```
> as.matrix(betas) %*% t(as.matrix(betas)) * sigma2_M + diag( sigmas2 )
       GM_AC  F_AC UTX_AC CAT_AC MRK_AC PFE_AC IBM_AC
GM_AC  5.519 0.686  0.541  0.781  0.416  0.526  0.455
F_AC   0.686 3.677  0.557  0.804  0.428  0.542  0.468
UTX_AC 0.541 0.557  1.231  0.634  0.338  0.427  0.370
CAT_AC 0.781 0.804  0.634  2.443  0.487  0.617  0.534
MRK_AC 0.416 0.428  0.338  0.487  3.334  0.328  0.284
PFE_AC 0.526 0.542  0.427  0.617  0.328  2.007  0.359
IBM_AC 0.455 0.468  0.370  0.534  0.284  0.359  0.994
```

This is to be compared with just computing the covariance matrix directly from the excess returns which gives

```
> cov(stockExRet)
       GM_AC  F_AC UTX_AC CAT_AC MRK_AC PFE_AC IBM_AC
GM_AC  5.512 2.616  0.619  0.836  0.234  0.474  0.469
F_AC   2.616 3.673  0.546  0.866  0.466  0.566  0.519
UTX_AC 0.619 0.546  1.230  0.799  0.314  0.303  0.319
CAT_AC 0.836 0.866  0.799  2.441  0.323  0.516  0.448
MRK_AC 0.234 0.466  0.314  0.323  3.329  0.954  0.171
PFE_AC 0.474 0.566  0.303  0.516  0.954  2.004  0.311
IBM_AC 0.469 0.519  0.319  0.448  0.171  0.311  0.992
```

These two results are quite similar.

Problem 5

This is the R^2 of the CAPM regression performed on the UTX symbol. From the R code for this problem we find a summary of this regression given by

```
> summary_fit_reg[3]
Response UTX_AC :

Call:
lm(formula = UTX_AC ~ market)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.0296     0.0343    0.86     0.39
market        0.9766     0.0506   19.31   <2e-16 ***

Residual standard error: 0.89 on 670 degrees of freedom
Multiple R-squared:  0.357,     Adjusted R-squared:  0.356
F-statistic:  373 on 1 and 670 DF,  p-value: <2e-16
```

Thus we have an R^2 of 0.357 indicating 35.7% of the variance in the returns of UTX explained by the variance in the market.

Problem 6

The expected excess return from each stock will be given by its beta times the expected excess return of the market. Thus we would get

```
> betas * ex_ret_M_predicted
```

```
 GM_AC   F_AC UTX_AC CAT_AC MRK_AC PFE_AC IBM_AC
0.0481 0.0495 0.0391 0.0564 0.0300 0.0380 0.0328
```

For the excess return for our stocks.

Exercises

Exercise 16.1

Using

$$R_{j,t} = \mu_{f,t} + \beta_j(R_{M,t} - \mu_{f,t}) + \epsilon_{j,t} \,,$$

by taking the expectation

Exercise 16.2

Part (a):

Chapter 17 (Factor Models and Principal Components)

Notes on the Book

Notes on Estimating Expectations and Covariances using Factors

Given the expression for $R_{j,t}$ we can evaluate the covariance between two difference asset returns as follows

$$
\begin{aligned}
\operatorname{Cov}(R_{j,t}, R_{j',t}) &= \operatorname{Cov}(\beta_{0,j} + \beta_{1,j}F_{1,t} + \beta_{2,j}F_{2,t} + \epsilon_{j,t}, \beta_{0,j'} + \beta_{1,j'}F_{1,t} + \beta_{2,j'}F_{2,t} + \epsilon_{j',t}) \\
&= \operatorname{Cov}(\beta_{1,j}F_{1,t}, \beta_{0,j'} + \beta_{1,j'}F_{1,t} + \beta_{2,j'}F_{2,t} + \epsilon_{j',t}) \\
&+ \operatorname{Cov}(\beta_{2,j}F_{2,t}, \beta_{0,j'} + \beta_{1,j'}F_{1,t} + \beta_{2,j'}F_{2,t} + \epsilon_{j',t}) \\
&+ \operatorname{Cov}(\epsilon_{j,t}, \beta_{0,j'} + \beta_{1,j'}F_{1,t} + \beta_{2,j'}F_{2,t} + \epsilon_{j',t}) \\
&= \beta_{1,j}\beta_{1,j'}\operatorname{Var}(F_{1,t}) + \beta_{1,j}\beta_{2,j'}\operatorname{Cov}(F_{1,t}, F_{2,t}) \\
&+ \beta_{2,j}\beta_{1,j'}\operatorname{Cov}(F_{2,t}, F_{1,t}) + \beta_{2,j}\beta_{2,j'}\operatorname{Var}(F_{2,t}) \\
&= \beta_{1,j}\beta_{1,j'}\operatorname{Var}(F_{1,t}) + \beta_{2,j}\beta_{2,j'}\operatorname{Var}(F_{2,t}) + (\beta_{1,j}\beta_{2,j'} + \beta_{1,j'}\beta_{2,j})\operatorname{Cov}(F_{1,t}, F_{2,t}) \ ,
\end{aligned}
$$

which is the same as the books equation 17.6.

R Lab

See the `R` script `Rlab.R` for this chapter. We first duplicate the bar plot of the eigenvalues and eigenvectors of the covariance matrix of the dataframe `yielddat`. These are shown in Figure 26.

Problem 1-2 (for fixed maturity are the yields stationary?)

See Figure 27 for a plot of the first four columns of the yield data (the first four maturities). These plots do not look stationary. This is especially true for index values from 1000 to 1400 where all yield curves seem to trend upwards.

As suggested in the book we can also use the augmented Dickey-Fuller test to test for stationarity. When we do this for each possible maturity we get

```
[1] "column index=  1; p_value=   0.924927"
[1] "column index=  2; p_value=   0.543508"
[1] "column index=  3; p_value=   0.410602"
[1] "column index=  4; p_value=   0.382128"
[1] "column index=  5; p_value=   0.382183"
```

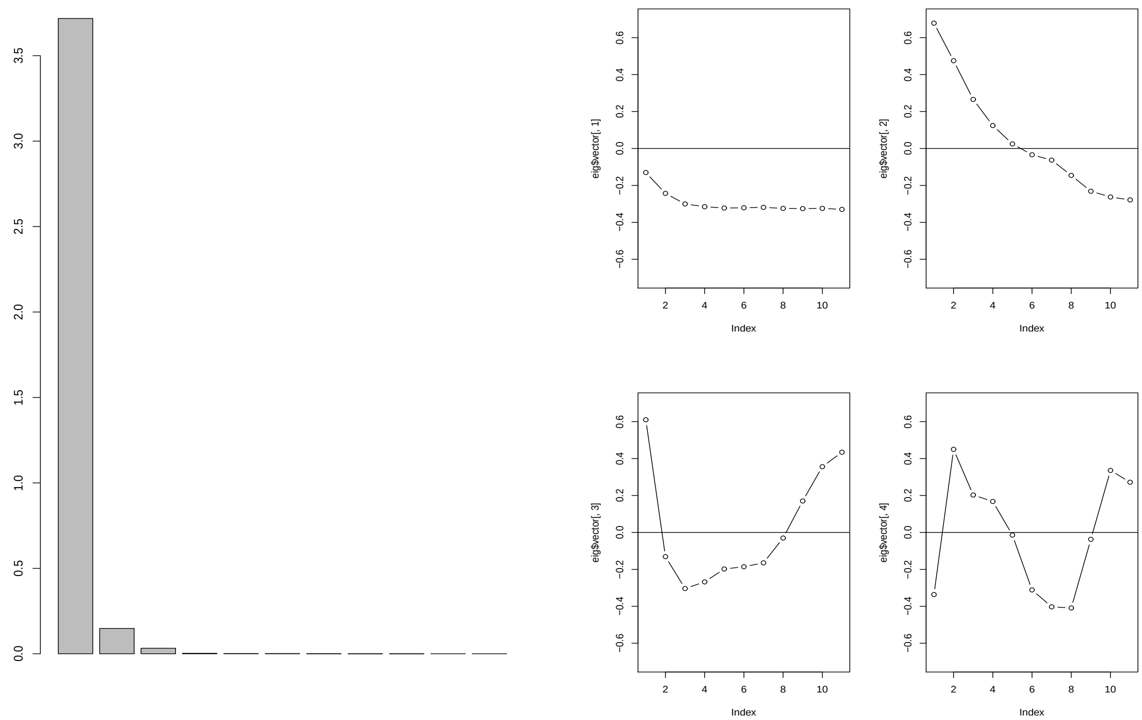

Figure 26: **Left:** The distribution of the eigenvalues of the yield data. **Right:** Plots of the first four eigenvectors of the yield data.

```
[1] "column index=  6; p_value=   0.386320"
[1] "column index=  7; p_value=   0.391729"
[1] "column index=  8; p_value=   0.437045"
[1] "column index=  9; p_value=   0.461692"
[1] "column index= 10; p_value=   0.460651"
[1] "column index= 11; p_value=   0.486028"
```

As all of these p values are “large” (none of them are less than 0.05) we can conclude that the raw yield curve data is *not* stationary.

Problem 3 (for fixed maturity are the *difference* in yields stationary?)

See Figure 28 for a plot of the first difference of each of the four columns of the yield data (the first difference of the first four maturities). These plots now *do* look stationary. Using the augmented Dickey-Fuller test we can show that the time series of yield differences are stationarity. Using the same code as before we get

```
[1] "column index=  1; p_value=   0.010000"
[1] "column index=  2; p_value=   0.010000"
```

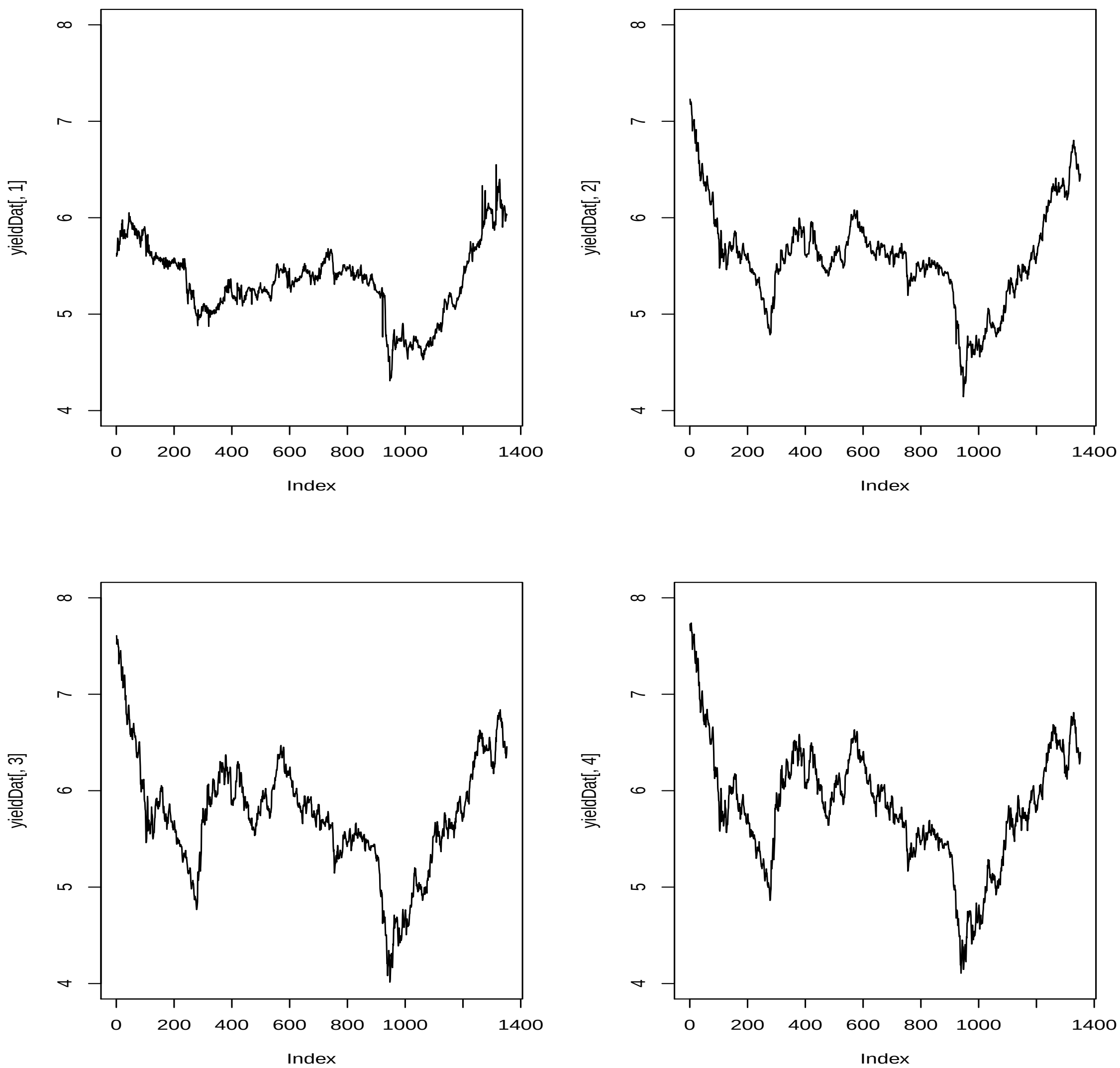

Figure 27: Plots time series of the first four yield maturities.

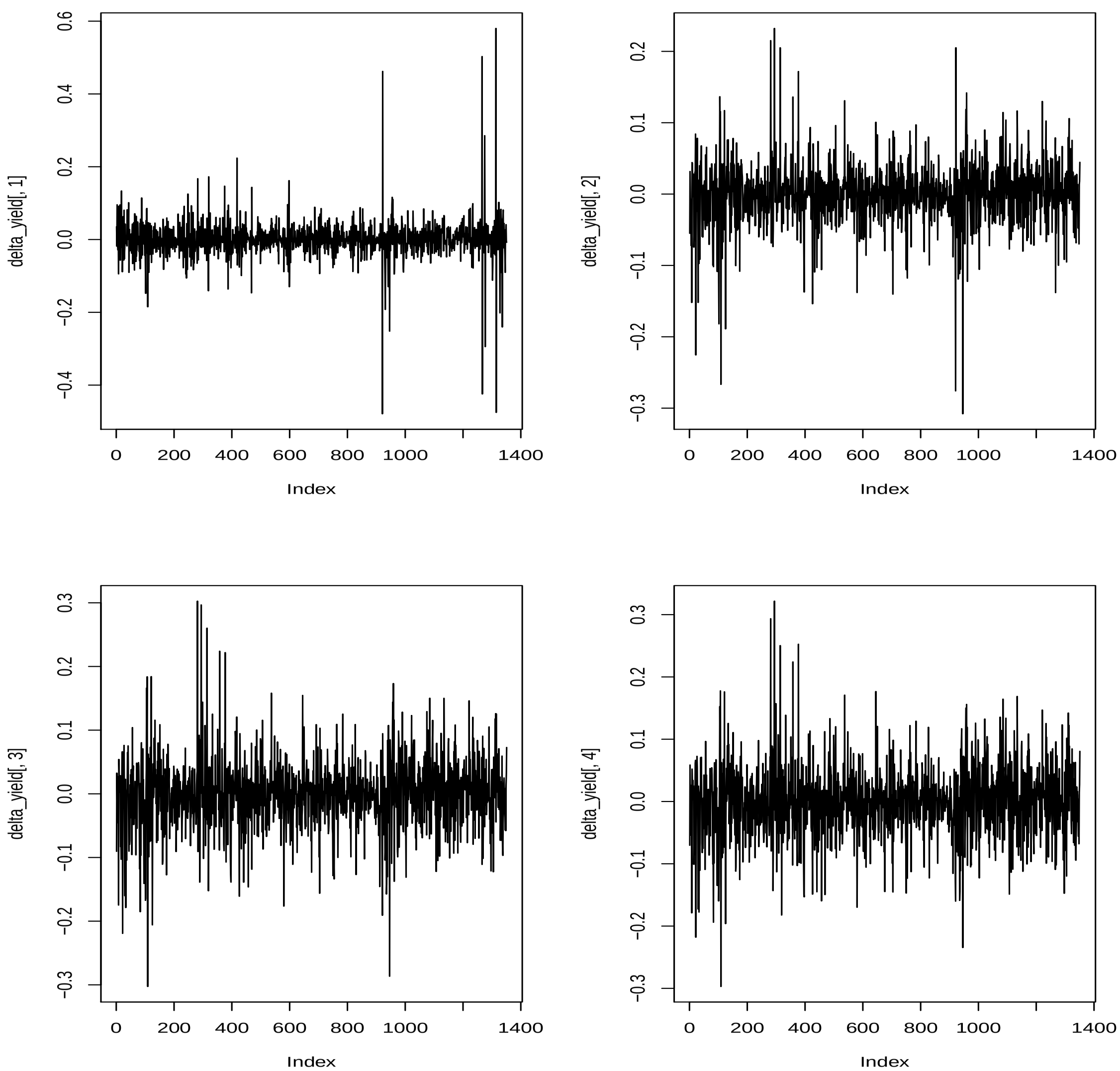

Figure 28: Plots the time series of the first difference of the first four interest rate yield maturities.

```
[1] "column index=  3; p_value=   0.010000"
[1] "column index=  4; p_value=   0.010000"
[1] "column index=  5; p_value=   0.010000"
[1] "column index=  6; p_value=   0.010000"
[1] "column index=  7; p_value=   0.010000"
[1] "column index=  8; p_value=   0.010000"
[1] "column index=  9; p_value=   0.010000"
[1] "column index= 10; p_value=   0.010000"
[1] "column index= 11; p_value=   0.010000"
There were 11 warnings (use warnings() to see them)
```

As all of these p values are "small" (they are all less than 0.01) we can conclude that the first differences of a yield at a fixed maturity *is* stationary. The warnings indicates that the `adf.test` command could not actually compute the correct p-value and that the true p-values are actually smaller than the ones printed above.

Problem 4 (PCA on differences between the yield curves)

Part (a): The variable `sdev` holds the standard deviations of each principal components, these are also the square root of the eigenvalues of the covariance matrix. The variable `loadings` hold the eigenvectors of the covariance matrix. The variable `center` hold the means that were subtracted in each feature dimension in computing the covariance matrix. The variable `scores` holds a matrix of each vector variable projected into all its principle components. We can check that this is so by comparing the two outputs

```
t(as.matrix(pca_del$loadings[,])) %*% t( delta_yield[1,] - pca_del$center )
                    2
Comp.1   0.1905643953
Comp.2   0.0375662026
Comp.3   0.0438591813
Comp.4  -0.0179855611
Comp.5   0.0002473111
Comp.6   0.0002924385
Comp.7   0.0101975886
Comp.8  -0.0093768514
Comp.9  -0.0036798653
Comp.10  0.0004287954
Comp.11 -0.0005602180
```

with

```
pca_del$scores[1,]
       Comp.1        Comp.2        Comp.3        Comp.4        Comp.5
```

```
 0.1905643953  0.0375662026  0.0438591813 -0.0179855611  0.0002473111
       Comp.6        Comp.7        Comp.8        Comp.9       Comp.10
 0.0002924385  0.0101975886 -0.0093768514 -0.0036798653  0.0004287954
      Comp.11
-0.0005602180
```

These two outputs are exactly the same (as they should be).

Part (b): Squaring the first two values of the `sdev` output we get

```
> pca_del$sdev[1:2]^2
     Comp.1      Comp.2
0.031287874 0.002844532
```

Part (c): The eigenvector corresponding to the largest eigenvalue is the first one and has values given by

```
> pca_del$loadings[,1]
      X1mon       X2mon       X3mon       X4mon       X5mon     X5.5mon
-0.06464327 -0.21518811 -0.29722014 -0.32199492 -0.33497517 -0.33411403
    X6.5mon     X7.5mon     X8.5mon     X9.5mon          NA.
-0.33220277 -0.33383143 -0.32985565 -0.32056039 -0.31668346
```

Part (d): Using the output from the `summary(pca_del)` which in a truncated form is given by

```
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4      Comp.5
Standard deviation     0.1768838 0.05333415 0.03200475 0.014442572 0.011029556
Proportion of Variance 0.8762330 0.07966257 0.02868616 0.005841611 0.003406902
Cumulative Proportion  0.8762330 0.95589559 0.98458175 0.990423362 0.993830264
```

we see from the `Cumulative Proportion` row above that to obtain 95% of the variance we must have at least 2 components. Taking three components gives more than 98% of the variance.

Problem 5 (zero intercepts in CAPM?)

The output of the `lm` gives the fitted coefficients and their standard errors, capturing the partial output of the `summary` command we get the following

```
Response GM :
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.0103747  0.0008924 -11.626   <2e-16 ***
FF_data$Mkt.RF  0.0124748  0.0013140   9.494   <2e-16 ***

Response Ford :
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.0099192  0.0007054  -14.06   <2e-16 ***
FF_data$Mkt.RF  0.0131701  0.0010386   12.68   <2e-16 ***

Response UTX :
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.0080626  0.0004199  -19.20   <2e-16 ***
FF_data$Mkt.RF  0.0091681  0.0006183   14.83   <2e-16 ***

Response Merck :
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.0089728  0.0009305  -9.643  < 2e-16 ***
FF_data$Mkt.RF  0.0062294  0.0013702   4.546 6.85e-06 ***
```

Notice that the p-value of all intercepts are smaller than the given value of α i.e. 5%. Thus we *cannot* accept the hypothesis that the coefficient β_0 is zero.

Problem 6

We can use the `cor` command to compute the correlation of the residuals of each of the CAPM models which gives

```
> cor( fit1$residuals )
              GM       Ford        UTX       Merck
GM     1.00000000 0.55410840 0.09020925 -0.04331890
Ford   0.55410840 1.00000000 0.09110409  0.03647845
UTX    0.09020925 0.09110409 1.00000000  0.05171316
Merck -0.04331890 0.03647845 0.05171316  1.00000000
```

The correlation between GM and Ford is quite large. To get confidence intervals for each correlation coefficient we will use the command `cor.test` to compute the 95% confidence intervals. We find

```
[1] "Correlation between  Ford and    GM; (  0.490439,   0.554108,   0.611894)"
[1] "Correlation between   UTX and    GM; (  0.002803,   0.090209,   0.176248)"
[1] "Correlation between   UTX and  Ford; (  0.003705,   0.091104,   0.177122)"
[1] "Correlation between Merck and    GM; ( -0.130254,  -0.043319,   0.044277)"
```

```
[1] "Correlation between Merck and  Ford; ( -0.051113,   0.036478,   0.123513)"
[1] "Correlation between Merck and   UTX; ( -0.035878,   0.051713,   0.138515)"
```

From the above output only the correlations between Merck and GM, Ford, and UTX seem to be zero. The others don't seem to be zero.

Problem 7 (comparing covariances)

The sample covariance or Σ_R can be given by using the `cov` command. Using the factor returns the covariance matrix Σ_R can be written as

$$\Sigma_R = \beta^T \Sigma_F \beta + \Sigma_\epsilon \,, \tag{24}$$

where β is the *row* vector of each stocks CAPM beta value. In the `R` code `Rlab.R` we compute both Σ_R and the right-hand-side of Equation 24 (which we denote as $\hat{\Sigma}$). If we plot these two matrices sequentially we get the following

```
> Sigma_R
                GM         Ford          UTX        Merck
GM    4.705901e-04 2.504410e-04 6.966363e-05 1.781501e-05
Ford  2.504410e-04 3.291703e-04 6.918793e-05 4.982034e-05
UTX   6.966363e-05 6.918793e-05 1.270578e-04 3.645322e-05
Merck 1.781501e-05 4.982034e-05 3.645322e-05 4.515822e-04
> Sigma_R_hat
                GM         Ford          UTX        Merck
GM    4.705901e-04 7.575403e-05 5.273459e-05 3.583123e-05
Ford  7.575403e-05 3.291703e-04 5.567370e-05 3.782825e-05
UTX   5.273459e-05 5.567370e-05 1.270578e-04 2.633334e-05
Merck 3.583123e-05 3.782825e-05 2.633334e-05 4.515822e-04
```

The errors between these two matrices are primarily in the off diagonal elements. We expect the pairs that have their residual correlation non-zero to have the largest discrepancy. If we consider the absolute value of the difference of these two matrices we get

```
> abs( Sigma_R - Sigma_R_hat )
                GM         Ford          UTX        Merck
GM    1.084202e-19 1.746870e-04 1.692904e-05 1.801622e-05
Ford  1.746870e-04 2.168404e-19 1.351424e-05 1.199209e-05
UTX   1.692904e-05 1.351424e-05 1.084202e-19 1.011988e-05
Merck 1.801622e-05 1.199209e-05 1.011988e-05 0.000000e+00
```

The largest difference is between GM and Ford which are also the two stocks that had the largest residual correlations under the CAPM model.

Problem 8 (the beta of `SMB` and `HML`)

If we look at the p-values of the fitted model on each stock we are getting results like the following

```
> sfit2$'Response GM'$coefficients
                   Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)    -0.010607689 0.000892357 -11.887271 7.243666e-29
FF_data$Mkt.RF  0.013862140 0.001565213   8.856390 1.451297e-17
FF_data$SMB    -0.002425130 0.002308093  -1.050708 2.939015e-01
FF_data$HML     0.006373645 0.002727395   2.336899 1.983913e-02
> sfit2$'Response GM'$coefficients
                   Estimate  Std. Error    t value     Pr(>|t|)
(Intercept)    -0.010607689 0.000892357 -11.887271 7.243666e-29
FF_data$Mkt.RF  0.013862140 0.001565213   8.856390 1.451297e-17
FF_data$SMB    -0.002425130 0.002308093  -1.050708 2.939015e-01
FF_data$HML     0.006373645 0.002727395   2.336899 1.983913e-02
> sfit2$'Response Ford'$coefficients
                    Estimate   Std. Error      t value     Pr(>|t|)
(Intercept)    -1.004705e-02 0.0007082909 -14.18492403 1.296101e-38
FF_data$Mkt.RF  1.348451e-02 0.0012423574  10.85396920 8.752040e-25
FF_data$SMB    -7.779018e-05 0.0018320033  -0.04246181 9.661475e-01
FF_data$HML     3.780222e-03 0.0021648160   1.74620926 8.138996e-02
> sfit2$'Response UTX'$coefficients
                    Estimate   Std. Error     t value     Pr(>|t|)
(Intercept)    -0.0080963376 0.0004199014 -19.2815220 2.544599e-62
FF_data$Mkt.RF  0.0102591816 0.0007365160  13.9293389 1.721546e-37
FF_data$SMB    -0.0028475161 0.0010860802  -2.6218286 9.013048e-03
FF_data$HML     0.0003584478 0.0012833841   0.2792989 7.801311e-01
> sfit2$'Response Merck'$coefficients
                   Estimate  Std. Error   t value     Pr(>|t|)
(Intercept)    -0.008694614 0.000926005 -9.389381 2.142386e-19
FF_data$Mkt.RF  0.007065701 0.001624233  4.350178 1.650293e-05
FF_data$SMB    -0.004094797 0.002395124 -1.709639 8.795427e-02
FF_data$HML    -0.009191144 0.002830236 -3.247483 1.242661e-03
```

In the fits above we see that the slope of the SMB and HML for different stocks have significance at the 2% - 8% level. For example, the HML slope for GM is significant at the 1.9% level. Based on this we cannot accept the null hypothesis of zero value for slopes.

Problem 9 (correlation of the residuals in the Fama-French model)

If we look at the 95% confidence interval under the Fama-French model we get

```
[1] "Correlation between  Ford and    GM; (  0.487024,   0.550991,   0.609079)"
[1] "Correlation between   UTX and    GM; ( -0.004525,   0.082936,   0.169138)"
[1] "Correlation between   UTX and  Ford; (  0.002240,   0.089651,   0.175702)"
[1] "Correlation between Merck and    GM; ( -0.119609,  -0.032521,   0.055064)"
[1] "Correlation between Merck and  Ford; ( -0.039887,   0.047708,   0.134575)"
```

```
[1] "Correlation between Merck and   UTX; ( -0.040087,   0.047508,   0.134378)"
```

Now the correlation between UTX and GM is zero (it was not in the CAPM). We still have a significant correlation between Ford and GM and between UTX and Ford (but it is now smaller).

Problem 10 (model fitting)

The AIC and BIC between the two models is given by

```
[1] "AIC(fit1)= -10659.895869; AIC(fit2)= -10688.779045"
[1] "BIC(fit1)= -10651.454689; BIC(fit2)= -10671.896684"
```

The smaller value in each case comes from the second fit or the Fama-French model.

Problem 11 (matching covariance)

The two covariance matrices are now

```
> Sigma_R
                GM         Ford          UTX        Merck
GM    4.705901e-04 2.504410e-04 6.966363e-05 1.781501e-05
Ford  2.504410e-04 3.291703e-04 6.918793e-05 4.982034e-05
UTX   6.966363e-05 6.918793e-05 1.270578e-04 3.645322e-05
Merck 1.781501e-05 4.982034e-05 3.645322e-05 4.515822e-04
> Sigma_R_hat
                GM         Ford          UTX        Merck
GM    4.705901e-04 7.853015e-05 5.432317e-05 3.108052e-05
Ford  7.853015e-05 3.291703e-04 5.602592e-05 3.437406e-05
UTX   5.432317e-05 5.602592e-05 1.270578e-04 2.733456e-05
Merck 3.108052e-05 3.437406e-05 2.733456e-05 4.515822e-04
```

The difference between these two matrices are smaller than in the CAPM model.

Problem 12 (Fama-French betas to excess returns covariance)

We will use the formula

$$\Sigma_R = \beta^T \Sigma_F \beta_j + \Sigma_\epsilon \,. \tag{25}$$

Here we have already calculated the value of Σ_F in the `R` code `Rlab.R`. We had found

```
            Mkt.RF         SMB         HML
Mkt.RF  0.46108683  0.17229574 -0.03480511
SMB     0.17229574  0.21464312 -0.02904749
HML    -0.03480511 -0.02904749  0.11023817
```

This factor covariance matrix will not change if the stock we are considering changes.

Part (a-c): Given the factor loadings for each of the two stocks and their residual variances we can compute the right-hand-side of Equation 25 and find

```
           [,1]       [,2]
[1,] 23.2254396  0.1799701
[2,]  0.1799701 37.2205144
```

Thus we compute that the variance of the excess return of Stock 1 is 23.2254396, the variance of the excess return of Stock 2 is 37.2205144 and the covariance between the excess return of Stock 1 and Stock 2 is 0.1799701.

Problem 13

Using the `factanal` command we see that the factor loadings are given by

```
        Factor1 Factor2
GM_AC    0.874  -0.298
F_AC     0.811
UTX_AC   0.617   0.158
CAT_AC   0.719   0.286
MRK_AC   0.719   0.302
PFE_AC   0.728   0.208
IBM_AC   0.854
MSFT_AC  0.646   0.142
```

The variance of the unique risks for Ford and GM are the values that are found in the "Uniquenesses" list which we found is given by

```
  GM_AC    F_AC  UTX_AC  CAT_AC  MRK_AC  PFE_AC  IBM_AC MSFT_AC
  0.148   0.341   0.594   0.401   0.392   0.427   0.269   0.562
```

Thus the two numbers we are looking for are 0.341 and 0.148.

Problem 14

The p-value for the `factanal` command is very small $1.39\,10^{-64}$ indicating that we should reject the null hypothesis and try a larger number of factors. Using four factors (the largest that we can use with eight inputs) gives a larger p-value 0.00153.

Problem 15

For statistical factor models the covariance between the log returns is given by

$$\Sigma_R = \hat{\beta}^T \hat{\beta} + \hat{\Sigma}_\epsilon \,,$$

where the $\hat{\beta}$ and $\hat{\Sigma}_\epsilon$ are the estimated loadings and uniqueness found using the `factanal` command. When we do that we get an approximate value for Σ_R given by

```
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
[1,] 1.0000002 0.6944136 0.4920453 0.5431030 0.5381280 0.5737600 0.7556789 0.5223550
[2,] 0.6944136 1.0000012 0.5075905 0.5961289 0.5966529 0.5994766 0.6909546 0.5305168
[3,] 0.4920453 0.5075905 0.9999929 0.4892064 0.4915723 0.4821511 0.5222193 0.4215048
[4,] 0.5431030 0.5961289 0.4892064 0.9999983 0.6034673 0.5829600 0.6052811 0.5055927
[5,] 0.5381280 0.5966529 0.4915723 0.6034673 1.0000019 0.5860924 0.6045542 0.5076939
[6,] 0.5737600 0.5994766 0.4821511 0.5829600 0.5860924 1.0000006 0.6150548 0.5000431
[7,] 0.7556789 0.6909546 0.5222193 0.6052811 0.6045542 0.6150548 1.0000003 0.5476733
[8,] 0.5223550 0.5305168 0.4215048 0.5055927 0.5076939 0.5000431 0.5476733 0.9999965
```

As Ford is located at index 2 and IBM is located at index 7 we want to look at the $(2,7)$th or $(7,2)$th element of the above matrix where we find the value 0.6909546.

Exercises

Exercise 17.1-2

These are very similar to the `Rlab` for this chapter.

Exercise 17.3

See the notes on Page 72.

Chapter 21 (Nonparametric Regression and Splines)

Notes on the Text

Notes on polynomial splines

When we force the linear fits on each side of the knot $x = t$ to be continuous we have that $a + bt = c + dt$ and this gives $c = a + (b - d)t$. When we use this fact we can simplify the formula for $s(x)$ when $x > t$ as

$$\begin{aligned} s(x) &= c + dx = a + (b - d)t + dx \\ &= a + bt + d(x - t) = a + bx - bx + bt + d(x - t) \\ &= a + bx - b(x - t) + d(x - t) = a + bx + (d - b)(x - t)\,, \end{aligned}$$

which is the books equation 21.11.

R Lab

See the `R` script `Rlab.R` for this chapter.

R lab: An Additive Model for Wages, Education, and Experience

When we enter and then run the given `R` code we see that the `summary` command gives that

$$\beta_0 = 6.189742 \quad \text{and} \quad \beta_1 = -0.241280\,.$$

Plots from the `fitGam` are duplicated in Figure 29.

R lab: An Extended CKLS model for the Short Rate

We are using 10 knots. The `outer` function here takes the outer difference of the values in `t` with those in `knots`. The statement that computes `X2` then computes the value of the "plus" functions for the various knots i.e. evaluates $(t - k)_+$ where t is the time variable and k is a knot. Then `X3` holds the linear spline basis function i.e. the total spline we are using to predict $\mu(t, r)$ is given by

$$\alpha_0 + \alpha_1 t + \sum_{i=1}^{10} \theta_i (t - k_i)_+\,.$$

Here α_0 and α_1 are the coefficients of the initial linear fit, and θ_i are the jumps in the first derivatives at each of the k_i knots. The first column of `X3` is the a column of all ones (for

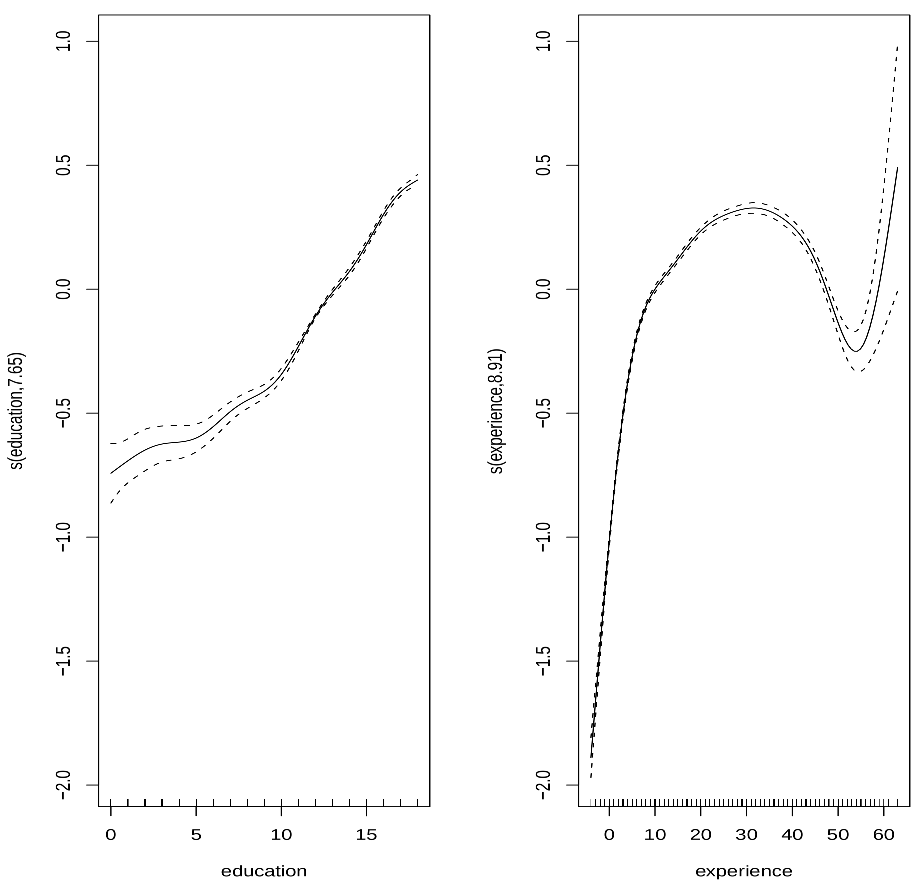

Figure 29: Plots for the splines `CPS1988` dataset.

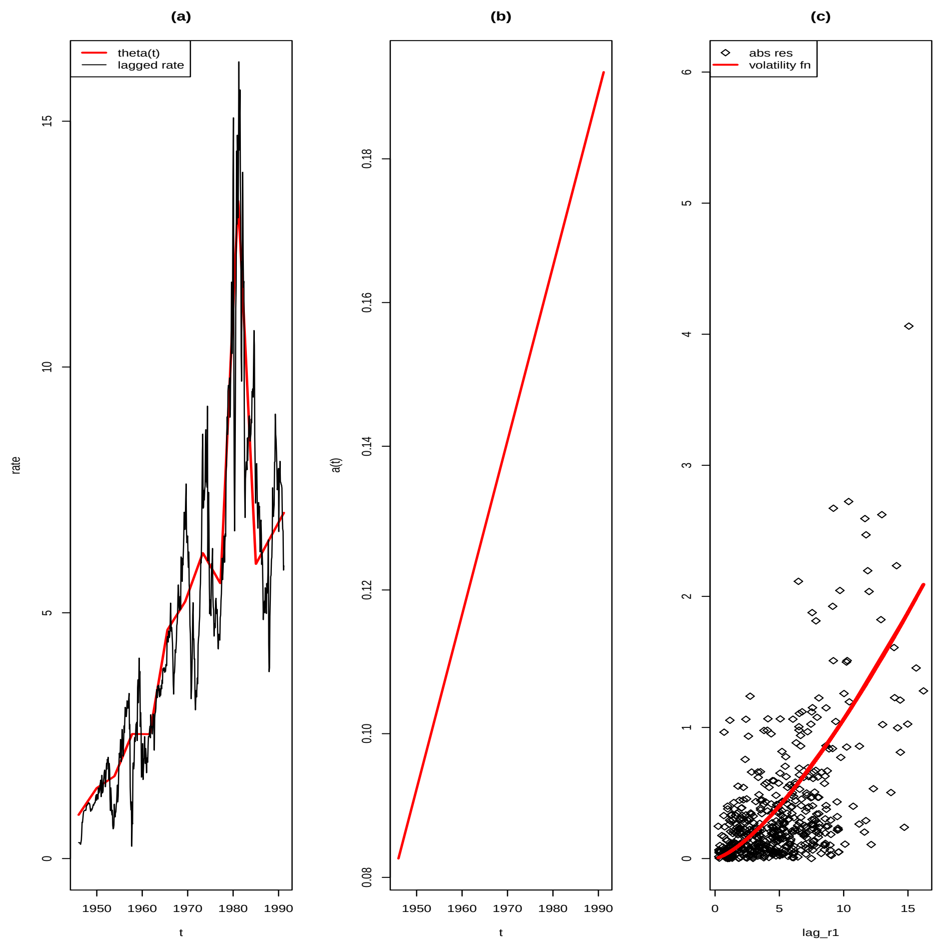

Figure 30: The three plots for the short rate example.

the constant α_0 term), the second column of X3 is a column of time relative to 1946. The rest of the columns of X3 are samples of the spline basis "plus" functions i.e. $(t - k_i)_+$ for $1 \le i \le 10$. When we run the given R code we generate the plot in Figure 30. Note that

```
X3[,1:2]%*%a
```

is a linear function in t but because of the way that X3 is constructed (its last 10 columns)

```
X3%*%theta
```

is the evaluation of a spline. Our estimates of the coefficients of α_0 and α_1 are not significant. A call to summary(nlmod_CKLS_ext)$parameters gives

```
             Estimate  Std. Error      t value      Pr(>|t|)
a1        0.082619604 0.087780955   0.94120192 3.470418e-01
a2        0.002423339 0.002706119   0.89550342 3.709356e-01
```

The large p-values indicate that these coefficients are not well approximated and might not be real effects.

Exercises

Exercise 21.1 (manipulations with splines)

Our expression for $s(t)$ is given by

$$s(t) = \beta_0 + \beta_1 x + b_1(x-1)_+ + b_2(x-2)_+ + b_3(x-3)_+ \,.$$

Where the "plus function" is defined by

$$(x-t)_+ = \left\{ \begin{array}{cc} 0 & x < t \\ x-t & x \geq t \end{array} \right.$$

From the given values of x and $s(x)$ we can compute

$$\begin{aligned}
&s(0) = 1 = \beta_0 \\
&s(1) = 1.3 = 1 + \beta_1 \quad \text{so} \quad \beta_1 = 0.3 \\
&s(2) = 5.5 = 1 + 0.3(2) + b_1(1) \quad \text{so} \quad b_1 = 3.9 \\
&s(4) = 6 = 1 + 0.3(4) + 3.9(3) + b_2(2) + b_3(1) \\
&s(5) = 6 = 1 + 0.3(5) + 3.9(4) + b_2(3) + b_3(2) \,.
\end{aligned}$$

Solving these two equations gives $b_2 = -3.7$ and $b_3 = -0.5$. Thus we have $s(x)$ given by

$$s(x) = 1 + 0.3x + 3.9(x-1)_+ - 3.7(x-2)_+ - 0.5(x-3)_+ \,.$$

Part (a): We would find

$$s(0.5) = 1 + 0.3(0.5) = 1.15 \,.$$

Part (b): We would find

$$s(3) = 1 + 0.3(3) + 3.9(2) - 3.7(1) = 6 \,.$$

Part (c): We would evaluate

$$\int_2^4 s(t)dt = \int_2^4 (1 + 0.3t + 3.9(t-1) - 3.7(t-2))dt - 0.5 \int_3^4 (t-3)dt \,,$$

each term could then be evaluated.

Exercise 21.2

The model 21.1 in the book is

$$\Delta r_t = r_t - r_{t-1} = \mu(r_{t-1}) + \sigma(r_{t-1})\epsilon_t \,. \tag{26}$$

In this problem we are told functional forms for $\mu(r_{t-1})$ and $\sigma(r_{t-1})$.

Part (a): Since ϵ_t has a mean of zero we have that

$$E[r_t | r_{t-1} = 0.04] = r_{t-1} + \mu(r_{t-1}) = 0.04 + 0.1(0.035 - 0.04) = 0.0395 \,.$$

Part (b): Since ϵ_t has a variance of one we have that

$$\mathrm{Var}\,(r_t | r_{t-1} = 0.02) = \sigma^2(r_{t-1}) = 2.3^2 {r_{t-1}}^2 = 2.3^2(0.02)^2 = 0.002116 \,.$$

Exercise 21.4

For the given spline we have

$$\begin{aligned} s'(x) &= 0.65 + 2x + 2(x-1)_+ + 1.2(x-2)_+ \\ s''(x) &= 2 + 2(x-1)_+^0 + 1.2(x-1.2)_+^0 \,. \end{aligned}$$

Part (a): We have

$$s(1.5) = 1 + 0.65(1.5) + 1.5^2 + 0.5^2 = 4.475 \,.$$

and

$$s'(1.5) = 0.65 + 3 + 2(0.5) + 1.2(0) = 4.65 \,.$$

Part (b): We have

$$s''(2.2) = 2 + 2(1) + 1.2(1) = 5.2 \,.$$