Homework 3

Benjamin Noland

1. (Cryer & Chan, Exercise 3.6)

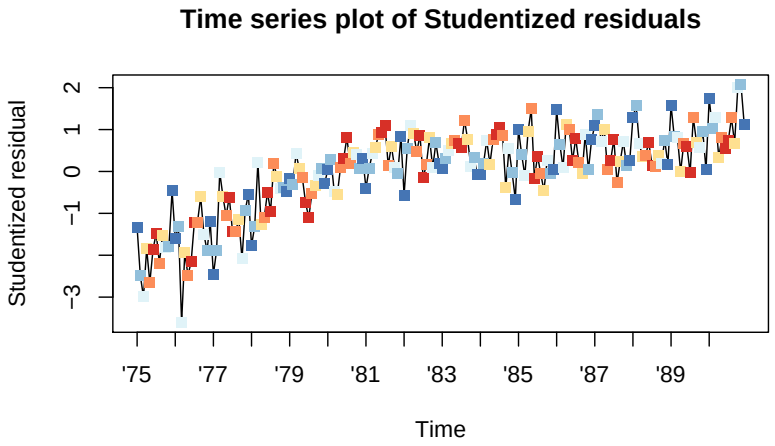
   d. The following R code plots the time series plot of the Studentized residuals from the model constructed in part (c):

```
month. <- season(beersales)
model <- lm(beersales ~ month. - 1)
resid <- rstudent(model)

plot(
  y = resid,
  x = time(beersales),
  type = "l",
  main = "Time series plot of Studentized residuals",
  xlab = "Time",
  ylab = "Studentized residual",
  xaxt = "n"  # Suppress default x axis.
)

temp_color = c(rev(brewer.pal(6, 'RdYlBu')), brewer.pal(6, 'RdYlBu'))
points(
  y = resid,
  x = time(beersales),
  col = temp_color,
  pch = 15
)

axis(1, at = 1975:1990, labels = paste0("'", 75:90))
```



The plot of the residuals seems to indicate that the residuals have non-constant mean and non-constant variance over time. In particular, the plot reflects the general increase in average monthly beer sales from approximately 1975 to 1980. However, the seasonal sales patterns visible in the original data (high sales in warm months, low sales in cold months) appear to be gone in this plot. Thus the model seems to have captured these patterns in the data.

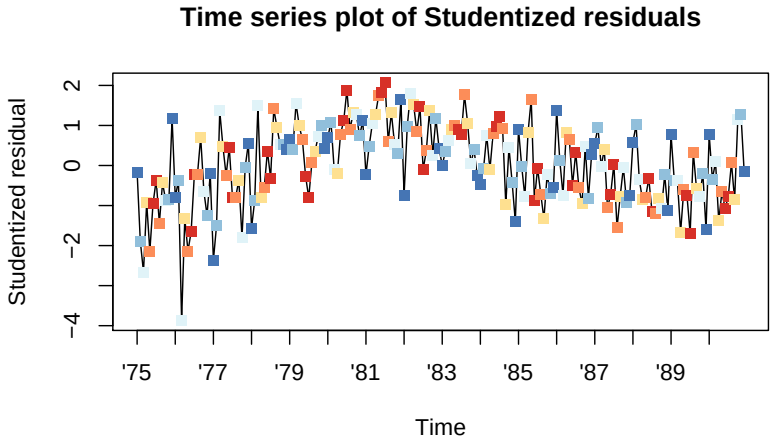
f. The following R code plots the time series plot of the Studentized residuals from the model constructed in part (e):

```
month. <- season(beersales)
time <- time(beersales)
model <- lm(beersales ~ month. + I(time^2) - 1)
resid <- rstudent(model)

plot(
  y = resid,
  x = time(beersales),
  type = "l",
  main = "Time series plot of Studentized residuals",
  xlab = "Time",
  ylab = "Studentized residual",
  xaxt = "n"  # Suppress default x axis.
)

temp_color = c(rev(brewer.pal(6, 'RdYlBu')), brewer.pal(6, 'RdYlBu'))
points(
  y = resid,
  x = time(beersales),
  col = temp_color,
  pch = 15
)

axis(1, at = 1975:1990, labels = paste0("'", 75:90))
```



The plot of the residuals seems to indicate that the residuals have non-constant mean and non-constant variance over time. In particular, as in the plot from part (d), the plot reflects the general increase in average monthly beer sales from approximately 1975 to 1980, but it is much less pronounced this time around. Thus it appears that the model has partly captured this pattern in the data. Moreover, the observations from 1975 to 1980 seem to be much more variable than those from 1980 onwards. However, the seasonal sales patterns visible in the original data (high sales in warm months, low sales in cold months) appear to be gone in this plot. Thus the model seems to have captured these patterns in the data.

2. (Cryer & Chan, Exercise 3.12)

   a. The following R code obtains the Studentized residuals from the seasonal-means plus quadratic time trend model:

```
month. <- season(beersales)
time <- time(beersales)
model <- lm(beersales ~ month. + I(time^2) - 1)
resid <- rstudent(model)
```

   b. The following R code performs a runs test on the residuals, assuming the residuals have a true median of zero:

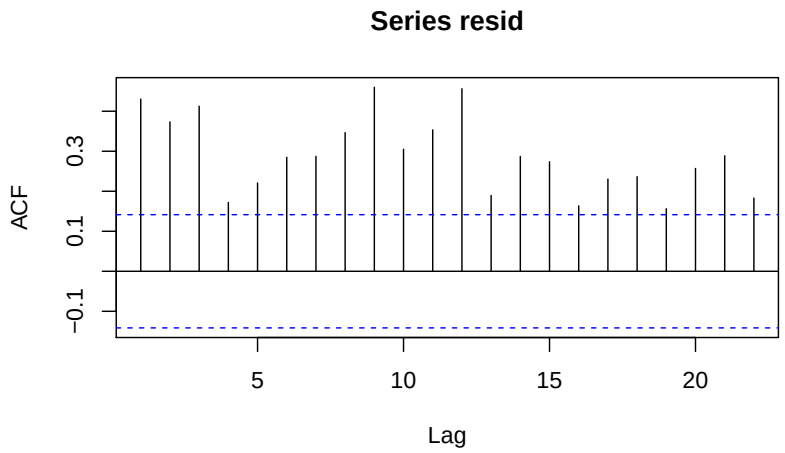
```
runs(resid)
```

```
## $pvalue
## [1] 4.32e-06
##
## $observed.runs
## [1] 65
##
## $expected.runs
## [1] 96.95833
##
## $n1
## [1] 98
##
## $n2
## [1] 94
##
## $k
## [1] 0
```

   The small p-value from the test indicates that we can safely reject the null hypothesis of independence of the residuals. In particular, we obtain a small number of runs compared to the expected numbers of runs, indicating that the observations tend to hang together over time.

c. The following R code plots a correlogram for the Studentized residuals:

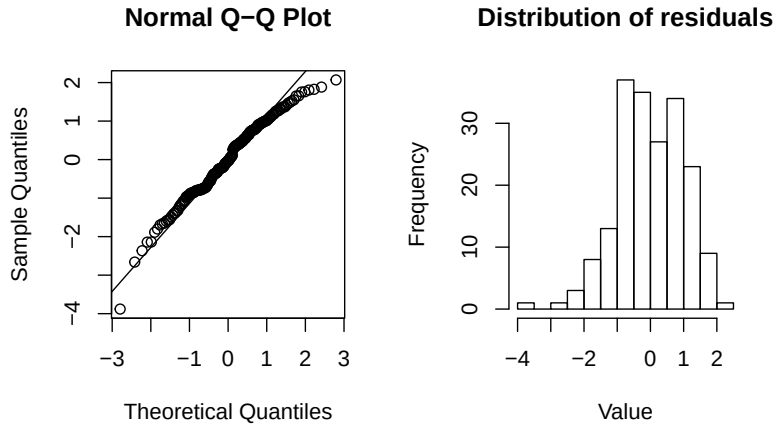
```
acf(resid)
```



Assuming stationarity of the time series represented by the residuals, the correlogram plots estimates (namely the sample autocorrelation function r_k) of the autocorrelation function ρ_k against lag k. Under independence of the residuals, we would have $\rho_k = 0$ for every k, and so we would expect the estimates r_k to all lie within two approximate standard errors of the sample autocorrelations, indicated by the horizontal dashed lines in the above plot. Clearly this is not the case here, indicating that the residuals are probably not independent.

d. The following R code displays a normal Q-Q plot of the Studentized residuals and a histogram displaying their distribution:

```
old_par <- par(mfrow = c(1, 2))
qqnorm(resid)
qqline(resid)
hist(resid, main = "Distribution of residuals", xlab = "Value")
par(old_par)
```



The points on the normal Q-Q plot deviate somewhat from the straight line, and so the plot residuals appear to be non-normal. This is further substantiated by

the histogram, which indicates clear asymmetry in the distribution.

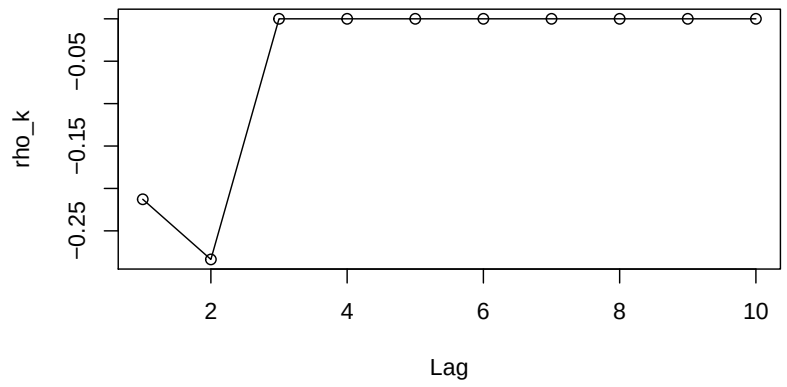
3. (Cryer & Chan, Exercise 4.2) The following R functions respectively compute the autocorrelation function for a given MA(2) process and plot it against lag:

```r
rho <- function(k, theta_1, theta_2) {
  if (k == 1) {
    (-theta_1 + theta_1 * theta_2) / (1 + theta_1^2 + theta_2^2)
  } else if (k == 2) {
    -theta_2 / (1 + theta_1^2 + theta_2^2)
  } else if (k >= 3) {
    0
  }
}

plot_rho <- function(k_vals, theta_1, theta_2) {
  rho_vals <- sapply(k_vals, rho, theta_1, theta_2)
  plot(x = k_vals, y = rho_vals, type = "o", xlab = "Lag", ylab = "rho_k")
}
```

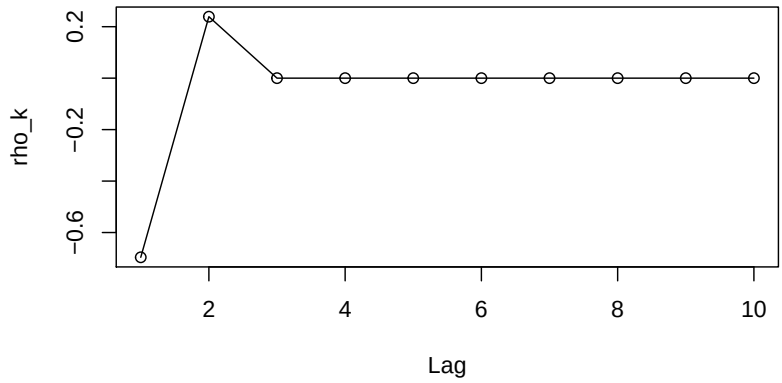
   a. Plot for $\theta_1 = 0.5$, $\theta_2 = 0.4$:

```r
plot_rho(1:10, 0.5, 0.4)
```



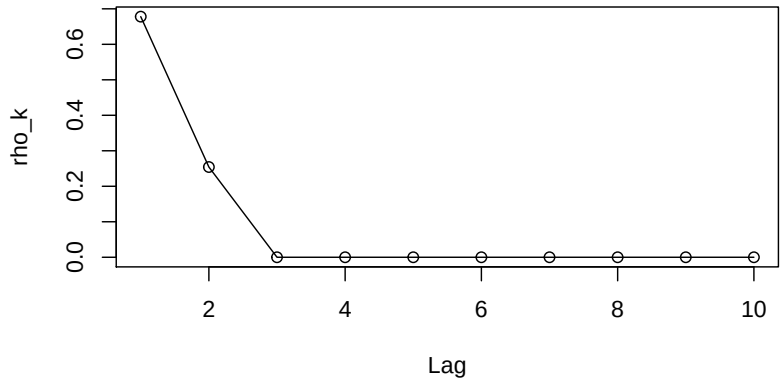
   b. Plot for $\theta_1 = 1.2$, $\theta_2 = -0.7$:

```r
plot_rho(1:10, 1.2, -0.7)
```



c. Plot for $\theta_1 = -1$, $\theta_2 = -0.6$:

```
plot_rho(1:10, -1, -0.6)
```



4. (Cryer & Chan, Exercise 4.3) We have the following expression for ρ_1 in terms of the parameter θ:

$$\rho_1 = \rho_1(\theta) = \frac{-\theta}{1+\theta^2}.$$

We can find the critical points of ρ_1 by setting

$$0 = \frac{d}{d\theta}\rho_1(\theta) = \frac{\theta^2 - 1}{(1+\theta^2)^2}.$$

The values of θ that satisfy this equation are $\theta = -1$ and $\theta = 1$, and since the domain of optimization has no boundary points, these are the only critical points. Since we have $\rho_1(-1) = 1/2$ and $\rho_1(1) = -1/2$, we can verify formally using the second derivative test that

$$\min_{\theta\in\mathbb{R}} \rho_1(\theta) = -\frac{1}{2} \quad \text{and} \quad \max_{\theta\in\mathbb{R}} \rho_1(\theta) = \frac{1}{2}.$$

5. (Cryer & Chan, Exercise 4.19) The process in question can be written

$$\begin{aligned} Y_t &= e_t - 0.5e_{t-1} + 0.25e_{t-2} - 0.125e_{t-3} + 0.0625e_{t-4} - 0.03125e_{t-5} + 0.015625e_{t-6} \\ &= e^t + \sum_{k=1}^{6} \left(-\frac{1}{2}\right)^k e_{t-k}. \end{aligned}$$

Thus the process can be expressed as a general linear process whose coefficients $\Psi_k = (-1/2)^k$ form an exponentially decaying sequence.

6. i. Assume that for each of the processes in question, e_t is independent of $Y_{t-1}, Y_{t-2}, Y_{t-3}, \ldots$. Then to prove stationarity it suffices to show that each of the solutions x to the AR characteristic equation

$$1 - \phi_1 x - \phi_2 x^2 = 0$$

satisfy $|x| > 1$ (i.e., have modulus exceeding 1). We get the following for each process:

   a. The solutions are $x_1 \approx -3.08167$ and $x_2 \approx 1.08167$, each of which has modulus exceeding 1. Thus the process is stationary.
   b. The solutions are $x_1 \approx -1.06969$ and $x_2 \approx 1.86969$, each of which has modulus exceeding 1. Thus the process is stationary.
   c. The solutions are $x_1 \approx 0.75 - 0.829156i$ and $x_1 \approx 0.75 + 0.829156i$, each of which have modulus $\approx 1.11803 > 1$. Thus the process is stationary.
   d. The solutions are $x_1 \approx -0.833 - 0.986013i$ and $x_2 \approx -0.833 + 0.986013i$, each of which have modulus $\approx 1.29078 > 1$. Thus the process is stationary.

   ii. The coefficients Ψ_j satisfy the recurrence

$$\begin{aligned} \Psi_0 &= 1 \\ \Psi_1 &= \phi_1 \\ \Psi_j &= \phi_1 \Psi_{j-1} + \phi_2 \Psi_{j-2} \quad (j \geq 2). \end{aligned}$$

   The following R function uses this recurrence to compute Φ_j for any $j \geq 0$:

```
psi <- function(j, phi_1, phi_2) {
  if (j == 0) {
    1
  } else if (j == 1) {
    phi_1
  } else if (j >= 2) {
    phi_1 * psi(j - 1, phi_1, phi_2) + phi_2 * psi(j - 2, phi_1, phi_2)
  }
}
```

   We can now compute Ψ_j $(j = 0, 1, 2, 3, 4)$ for each process:

   a.
```
sapply(0:4, psi, 0.6, 0.3)
```
```
## [1] 1.0000 0.6000 0.6600 0.5760 0.5436
```

   b.
```
sapply(0:4, psi, -0.4, 0.5)
```
```
## [1]  1.0000 -0.4000  0.6600 -0.4640  0.5156
```

   c.
```
sapply(0:4, psi, 1.2, -0.7)
```

```
## [1]  1.0000  1.2000  0.7400  0.0480 -0.4604
```

d.
```
sapply(0:4, psi, -1, -0.6)
```

```
## [1]  1.00 -1.00  0.40  0.20 -0.44
```

iii. The Yule-Walker equations state that

$$\rho_1 = \phi_1 + \phi_2\rho_1$$
$$\rho_2 = \phi_1\rho_1 + \phi_2\rho_2,$$

so that

$$\rho_1 = \frac{\phi_1}{1-\phi_2}$$
$$\rho_2 = \frac{\phi_2(1-\phi_2)+\phi_1^2}{1-\phi_2}.$$

Using these expressions for ρ_1 and ρ_2, the fact that $\rho_0 = 1$, as well as the relation

$$\rho_k = \phi_1\rho_{k-1} + \phi_2\rho_{k-2} \quad (k \geq 1),$$

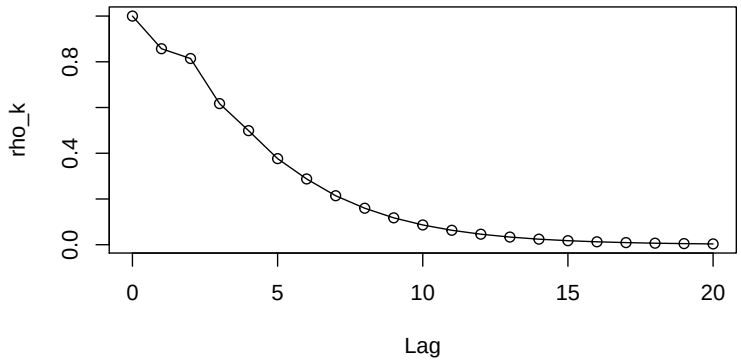
we can compute the autocorrelation function ρ_k for any chosen lag $k \geq 0$. The first of the following R functions computes ρ_k in this way; the second plots the results against lag:

```
rho <- function(k, phi_1, phi_2) {
  if (k == 0) {
    1
  } else if (k == 1) {
    phi_1 / (1 - phi_2)
  } else if (k == 2) {
    (phi_2 * (1 - phi_2) + phi_1^2) / (1 - phi_2)
  } else if (k >= 3) {
    phi_1 * rho(k - 1, phi_1, phi_2) + phi_2 * rho(k - 2, phi_2, phi_2)
  }
}

plot_rho <- function(k_vals, phi_1, phi_2) {
  rho_vals <- sapply(k_vals, rho, phi_1, phi_2)
  plot(x = k_vals, y = rho_vals, type = "o", xlab = "Lag", ylab = "rho_k")
}
```

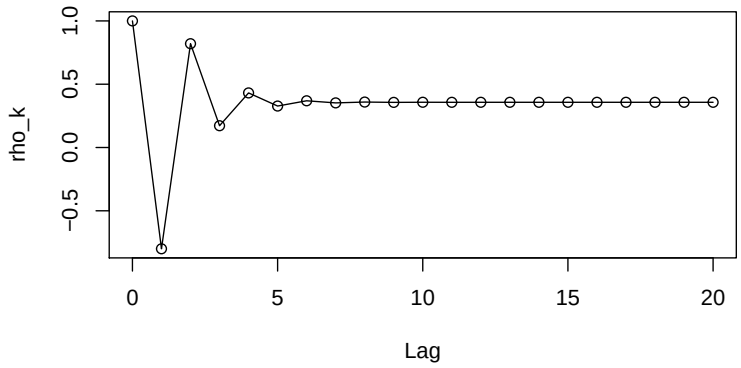
a. Plot for $\phi_1 = 0.6$, $\phi_2 = 0.3$:

```
plot_rho(0:20, 0.6, 0.3)
```



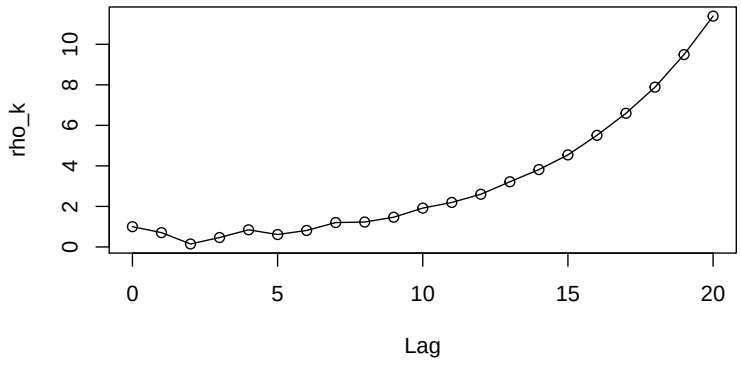
b. Plot for $\phi_1 = -0.4$, $\phi_2 = 0.5$:

```
plot_rho(0:20, -0.4, 0.5)
```



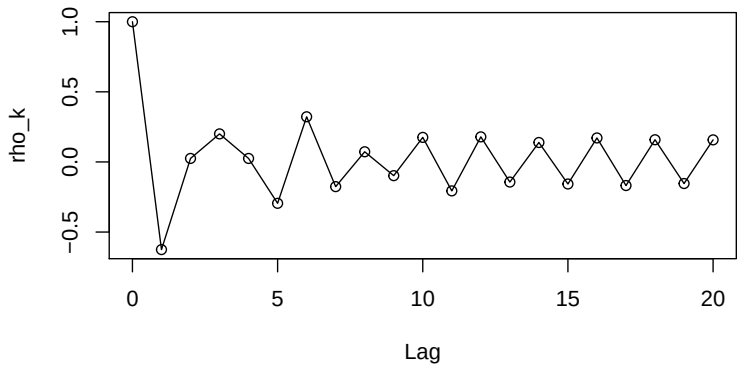
c. Plot for $\phi_1 = 1.2$, $\phi_2 = -0.7$:

```
plot_rho(0:20, 1.2, -0.7)
```



d. Plot for $\phi_1 = -1$, $\phi_2 = -0.6$:

```
plot_rho(0:20, -1, -0.6)
```


rho_k
1.0
0.5
0.0
−0.5
0
5
10
15
20
Lag