Chapter 3

3.1

3.1 For an MA(1), $x_t = w_t + \theta w_{t-1}$, show that $|\rho_x(1)| \leq 1/2$ for any number θ. For which values of θ does $\rho_x(1)$ attain its maximum and minimum?

We know from Problem 2.8,

$$\rho_x(1) = \frac{\theta}{1+\theta^2}$$

We argue by contraction.

Assume $\frac{\theta}{1+\theta^2} > \frac{1}{2}$. The argument for $-\frac{1}{2}$ is similar.

We have three cases: $\theta > 1$, $\theta = 1$ and $\theta < 1$.

Case 1: $\theta > 1$

$$\theta^2 > 1$$
$$1+\theta^2 > 2$$
$$\frac{1}{1+\theta^2} < \frac{1}{2}$$
$$\frac{\theta}{1+\theta^2} < \frac{1}{2}$$

Which contradicts the assumption.

Case 2: $\theta = 1$

It's obvious that

$$\frac{\theta}{1+\theta^2} = \frac{1}{2}$$

Which contradicts the assumption.

Case 3: $\theta < 1$

$$\theta^2 < 1$$
$$\theta^2 + 1 < 2$$
$$\frac{1}{1+\theta^2} < \frac{1}{2}$$
$$\frac{\theta}{1+\theta^2} < \frac{1}{2}$$

Which contradicts the assumption.

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

$\rho_x(1)$ takes a maximum at $\theta = 1$ and a minimum at $\theta = -1$.

3.2

3.2 Let $\{w_t; t = 0, 1, \dots\}$ be a white noise process with variance σ_w^2 and let $|\phi| < 1$ be a constant. Consider the process $x_0 = w_0$, and

$$x_t = \phi x_{t-1} + w_t, \quad t = 1, 2, \dots .$$

We might use this method to simulate an AR(1) process from simulated white noise.

(a) Show that $x_t = \sum_{j=0}^{t} \phi^j w_{t-j}$ for any $t = 0, 1, \dots$.

We can prove by induction.

Base case: $t = 0$

$$x_0 = w_0$$

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

Induction assumption: $$x_t = \sum_{j=0}^{t} \phi^j w_{t-j}$$

$$x_{t+1} = \phi x_t + w_t = \phi \sum_{j=0}^{t} \phi^j w_{t-j} + w_t = \sum_{j=0}^{t} \phi^{j+1} w_{t-j} + w_t$$

$$= \sum_{j=1}^{t+1} \phi^j w_{t+1-j} + \phi^0 w_t = \sum_{j=0}^{t+1} \phi^j w_{t+1-j}$$

(b) Find the $E(x_t)$.

$\mathrm{E}(x_t) = \sum_{j=0}^{t} \phi^j \mathrm{E}(w_{t-j}) = 0$

(c) Show that, for $t = 0, 1, \dots,$

$$\mathrm{var}(x_t) = \frac{\sigma_w^2}{1-\phi^2}(1 - \phi^{2(t+1)})$$

$$var(x_t) = cov(x_t, x_t) = cov(\sum_{j=0}^{t} \phi^j w_{t-j}, \sum_{k=0}^{t} \phi^k w_{t-k})$$

$$= \mathbb{E}[\sum_{j=0}^{t} \phi^j w_{t-j} \sum_{k=0}^{t} \phi^k w_{t-k}] = \sum_{j=0}^{t} \sum_{k=0}^{t} \phi^{t+k} \mathbb{E}[w_{t-j} w_{t-k}]$$

$$= \sum_{j=0}^{t} \phi^{2j} \sigma_w^2 = \sum_{j=1}^{t+1} \phi^{2(j-1)} \sigma_w^2 = \sigma_w^2 \frac{1 - \phi^{2(t+1)}}{1 - \phi^2}$$

Where we have used finite geometric sums and $\mathbb{E}[w_{t-j} w_{t-k}] = 0$ when $j \neq k$

(d) Show that, for $h \geq 0$,

$$\text{cov}(x_{t+h}, x_t) = \phi^h \text{var}(x_t)$$

$$x_{t+h} = \sum_{j=0}^{t+h} \phi^j w_{t+h-j} = \sum_{j=0}^{h-1} \phi^j w_{t+h-j} + \sum_{j=h}^{t+h} \phi^j w_{t+h-j} = \sum_{j=0}^{h-1} \phi^j w_{t+h-j} + \phi^h \sum_{k=0}^{t} \phi^k w_{t-k}$$

$$= \sum_{j=0}^{h-1} \phi^j w_{t+h-j} + \phi^h x_t$$

$$\text{cov}(x_{t+h}, x_t) = \text{cov}\left(\sum_{j=0}^{h-1} \phi^j w_{t+h-j} + \phi^h x_t,\, x_t\right) = \phi^h \text{var}(x_t)$$

(e) Is x_t stationary?

No, because the autocovariance depends on time.

(f) Argue that, as $t \to \infty$, the process becomes stationary, so in a sense, x_t is "asymptotically stationary."

As $t \to \infty$, $\phi^{2(t+1)} \to 0$, so $var(x_t) \to \dfrac{\sigma_w^2}{1-\phi^2}$

Hence, it is independent of time.

Essentially, AR(1) models are only stationary when they have an infinite number of lagged variables. Finite AR(1) models, like the one given in this problem, are not stationary.

(g) Comment on how you could use these results to simulate n observations of a stationary Gaussian AR(1) model from simulated iid N(0,1) values.

Generate more than n, say, $n + n_0$, then discard the first n_0 values.

(h) Now suppose $x_0 = w_0/\sqrt{1-\phi^2}$. Is this process stationary? *Hint*: Show $\text{var}(x_t)$ is constant.

Write:

$$x_t = \phi^t \frac{w_0}{\sqrt{1-\phi^2}} + \sum_{j=0}^{t-1} \phi^j w_{t-j}$$

$$var(x_t) = \mathbb{E}[(\phi^t \frac{w_0}{\sqrt{1-\phi^2}} + \sum_{j=0}^{t-1} \phi^j w_{t-j})(\phi^t \frac{w_0}{\sqrt{1-\phi^2}} + \sum_{k=0}^{t-1} \phi^k w_{t-k})]$$

$$= \mathbb{E}[\phi^{2t} \frac{w_0^2}{1-\phi^2} + \phi^t \frac{w_0}{1-\phi^2} \sum_{j=0}^{t-1} \phi^j w_{t-j} + \phi^t \frac{w_0}{1-\phi^2} \sum_{k=0}^{t-1} \phi^k w_{t-k} + \sum_{j=0}^{t-1} \phi^j w_{t-j} \sum_{k=0}^{t-1} \phi^k w_{t-k}]$$

$$= \phi^{2t} \frac{\sigma_w^2}{1-\phi^2} + \sum_{j=0}^{t-1} \phi^{2j} \sigma_w^2$$

$$= \phi^{2t} \frac{\sigma_w^2}{1-\phi^2} + \sigma_w^2 \frac{1-\phi^{2t}}{1-\phi^2} = \frac{\sigma_w^2}{1-\phi}$$

And, hence, the series is stationary.

3.3

3.3 Verify the calculations made in Example 3.4 as follows.

(a) Let $x_t = \phi x_{t-1} + w_t$ where $|\phi| > 1$ and $w_t \sim$ iid $N(0, \sigma_w^2)$. Show $E(x_t) = 0$ and $\gamma_x(h) = \sigma_w^2 \phi^{-2} \phi^{-h}/(1-\phi^{-2})$ for $h \geq 0$.

The model $x_t = \phi x_{t-1} + w_t$ is not stationary because $|\phi| > 1$. However, we can rewrite the model as a non-causal stationary process, using (3.11):

$$x_t = -\sum_{j=1}^{\infty} \phi^{-j} w_{t+j}$$

Hence,
$\mathbb{E}[x_t] = 0$.

Now, since we are using the non-causal model $x_{t-1} = \phi^{-1} x_t - \phi^{-1} w_t$, the error term is now $w_t \sim N(0, \phi^{-2}\sigma_w^2)$. Otherwise, it would not be an AR(1) process.
Then,

$$cov(x_{t+h}, x_t) = \mathbb{E}[(-\phi^{-1} w_{t+h+1} - \phi^{-2} w_{t+h+2} - \ldots)(-\phi^{-1} w_{t+1} - \ldots - \phi^{-h-1} w_{t+h+1} - \phi^{-h-2} w_{t+h+2} - \ldots)] = \phi^{-1}\phi^{-h-1}\phi^{-2}\sigma_w^2 + \phi^{-2}\phi^{-h-2}\phi^{-2}\sigma_w^2 + \ldots = \sum_{j=1}^{\infty} \phi^{-j}\phi^{-h-j}\phi^{-2}\sigma_w^2$$

$$= \phi^{-2}\phi^{-h}\sigma_w^2 \sum_{j=1}^{\infty} \phi^{-2j}$$

Which gives the result using infinite geometric sums.

(b) Let $y_t = \phi^{-1} y_{t-1} + v_t$ where $v_t \sim$ iid $\mathrm{N}(0, \sigma_w^2 \phi^{-2})$ and ϕ and σ_w are as in part (a). Argue that y_t is causal with the same mean function and autocovariance function as x_t.

The model is causal with the absolute value of the parameter less than 1, and the error term has the same distribution as in part (a). Hence, we can expect the mean and covariance to be the same.

3.4

3.4 Identify the following models as ARMA(p, q) models (watch out for parameter redundancy), and determine whether they are causal and/or invertible:

(a) $x_t = .80 x_{t-1} - .15 x_{t-2} + w_t - .30 w_{t-1}$.

$$x_t - .8x_{t-1} + .15x_{t-2} = w_t - .3w_{t-1}$$
$$(1 - .8B + .15B^2)x_t = (1 - .3B)w_t$$
$$(1 - .3B)(1 - .5B)x_t = (1 - .3B)w_t$$
$$(1 - .5B)x_t = w_t$$
$$x_t = .5x_{t-1} + w_t$$

The model is an AR(1), causal, invertible and stationary because $|\phi| = .5 < 1$.

(b) $x_t = x_{t-1} - .50 x_{t-2} + w_t - w_{t-1}$.

$$x_t - x_{t-1} + .5x_{t-2} = w_t - w_{t-1}$$
$$(1 - B + .5B^2)x_t = (1 - B)w_t$$

The model cannot be reduced further and hence is an ARMA(2,1).

Using Property 3.1, we know that the model is causal if the roots of $\phi(x)$ are greater than 1.

$$\phi(z) = 1 - z_1 + .5z_2 = 0$$
$$z = \frac{1 \pm \sqrt{1 - 4(.5)(1)}}{2(.5)} = 1 \pm i$$
$$|z| = \sqrt{1^2 + 1^2} = \sqrt{2} > 2$$

Given that this effectively represents an AR(2), we can also use

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad |\phi_2| < 1$$

from Example 3.9. Both methods conclude that the model is causal.

Using Property 3.2, we know that the model is invertible if the roots $\theta(z)$ are greater than 1.

$$\theta(z) = 1 - z = 0$$
$$z = 1$$

And hence the model is not invertible.

3.5

3.5 Verify the causal conditions for an AR(2) model given in (3.28). That is, show that an AR(2) is causal if and only if (3.28) holds.

If:
See solutions manual.

Only if:
We have three cases:
$z_1 < 0$ and $z_2 < 0$, $z_1 > 0$ and $z_2 > 0$, $z_1 > 0$ and $z_2 < 0$ w.l.o.g.
Case 1:

$-1 < \frac{1}{z_1} < 0$, $-1 < \frac{1}{z_2} < 0$

$-2 < \frac{1}{z_1} + \frac{1}{z_2} < 0$, $-1 < -\frac{1}{z_1 z_2} < 0$

$-3 < \frac{1}{z_1} + \frac{1}{z_2} - \frac{1}{z_1 z_2} < 0 < 1$

$\phi_1 + \phi_2 < 1$

$0 < -\frac{1}{z_1} < 1$, $0 < -\frac{1}{z_2} < -1$

$-\frac{1}{z_1} - \frac{1}{z_2} < 2$

$-\frac{1}{z_1 z_2} - \frac{1}{z_1} - \frac{1}{z_2} < -\frac{1}{z_1 z_2} + 2 < 2 - 1 = 1$

$\phi_2 - \phi_1 < 1$

Case 2:

$0 < \frac{1}{z_1} < 1$, $0 < \frac{1}{z_2} < 1$

$0 < \frac{1}{z_1 z_2} < 1$

$\frac{1}{z_1} + \frac{1}{z_2} < 2$

$\frac{1}{z_1} + \frac{1}{z_2} - \frac{1}{z_1 z_2} < 2 - 1 = 1$

$\phi_1 + \phi_2 < 1$

$-1 < -\frac{1}{z_1} < 0$, $-1 < -\frac{1}{z_2} < 0$

$$-2 < -\frac{1}{z_1} - \frac{1}{z_2} < 0$$
$$-3 < -\frac{1}{z_1 z_2} - \frac{1}{z_1} - \frac{1}{z_2} < 0$$
$$\phi_2 - \phi_1 < 1$$

Case 3:
Can't figure out the third case.

The third inequality is obvious.

3.6

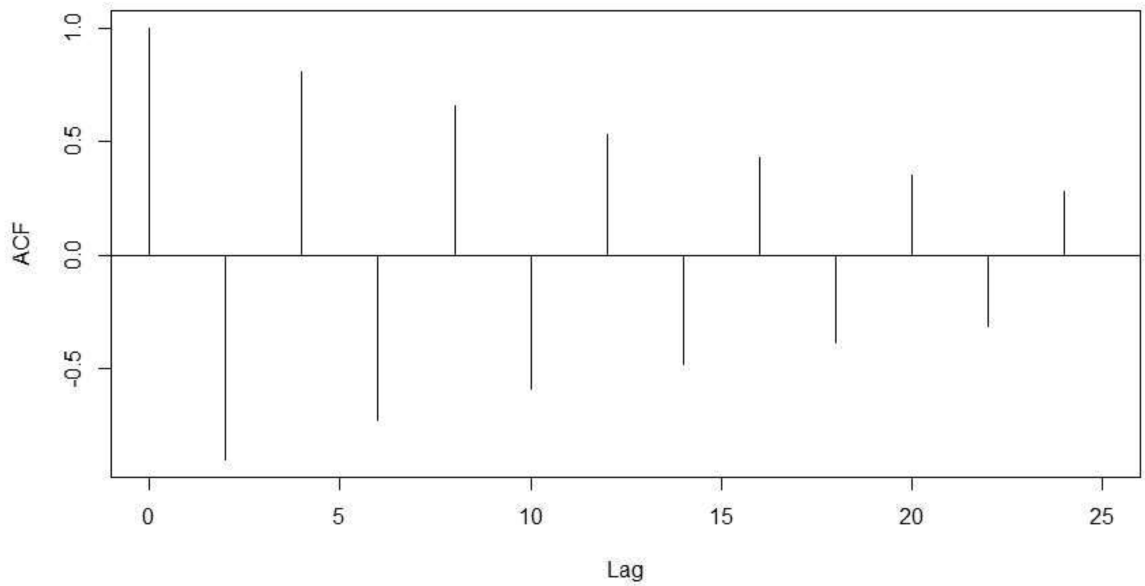
3.6 For the AR(2) model given by $x_t = -.9x_{t-2} + w_t$, find the roots of the autoregressive polynomial, and then plot the ACF, $\rho(h)$.

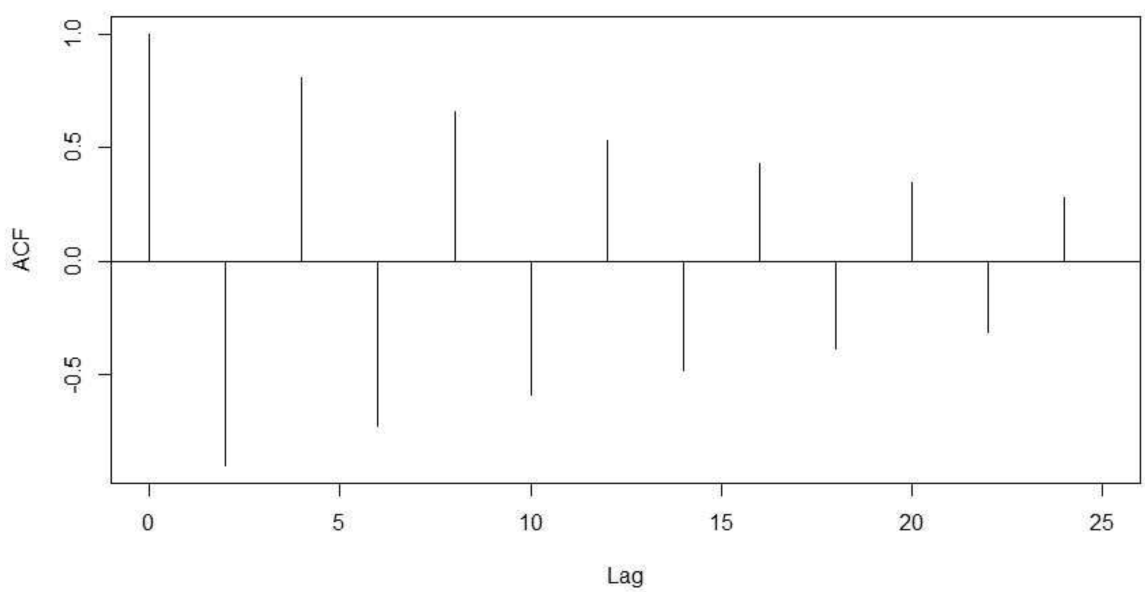
$$\phi(z) = 1 + .9z^2$$
$$z = \frac{\pm\sqrt{-4(.9)(1)}}{2(.9)} - \pm 1.054i$$

The polar coordinate form of the root is $z_1 = 1.054e^{i\frac{\pi}{2}}$
Using the results from Example 3.10,

$$\rho(h) = a1.054^{-h}cos(h\frac{\pi}{2} + b)$$
$$\rho(0) = 1 = acos(b)$$
$$\rho(1) = 0 = \frac{a}{1.054}cos(\frac{\pi}{2} + b)$$
$$\rho(h) = 1.054^{-h}cos(h\frac{\pi}{2})$$

```
par(mfrow=c(2,1))
acf = function(h) {
  lag = (1.054^-h)*cos((pi/2) * h)
  return(lag)
}
x = 0:25
plot(x, acf(x), type="h", xlab="Lag", ylab="ACF")
abline(h=0)
u = ARMAacf(ar=c(0,-.9), lag.max=25)
plot(0:25, u, type="h", xlab="Lag", ylab="ACF")
abline(h=0)
```


ACF
Lag


ACF
Lag

3.7

3.7 For the AR(2) series shown below, use the results of Example 3.10 to determine a set of difference equations that can be used to find the ACF $\rho(h)$, $h = 0, 1, \ldots$; solve for the constants in the ACF using the initial conditions. Then plot the ACF values to lag 10 (use `ARMAacf` as a check on your answers).

(a) $x_t + 1.6x_{t-1} + .64x_{t-2} = w_t$.

$$\phi(z) = 1 + 1.6z + .64z^2 = (1 + .8z)^2$$

$z_0 = -1.25z$

$\rho(h) = -1.25^h(c_1 + c_2 h)$

$\rho(0) = 1 = c_1$

$$\rho(0) = -.8(1 + c_2) = \frac{1.6}{1 + .64} = -.9756$$

$c_2 = .22$

$\rho(h) = -1.25^h(1 + .22h)$

(b) $x_t - .40x_{t-1} - .45x_{t-2} = w_t$.

$$\phi(z) = 1 - .4z - .45z^2 = (1 - .9z)(1 + .5z)$$

$z_1 = 1.1111,\ \ z_2 = -2$

$\rho(h) = c_1 1.1111^{-h} + c_2 2^{-h}$

$\rho(0) = 1 = c_1 + c_2$

$$\rho(1) = \frac{c_1}{1.1111} + \frac{c_2}{2} = \frac{-.4}{1 + .45} = -.2759$$

$c_1 = .16,\ \ c_2 = .84$

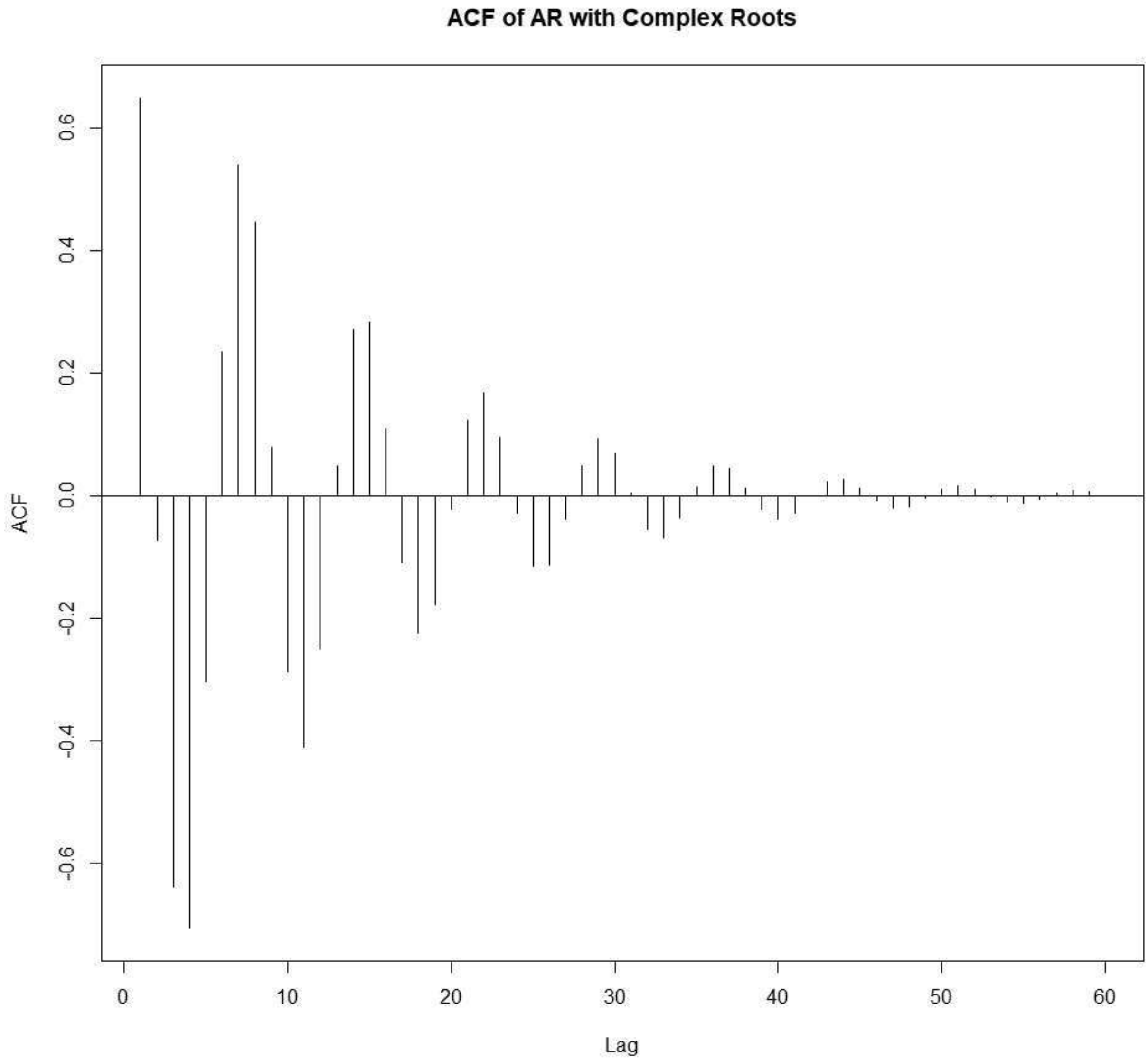
$\rho(h) = (.16)1.1111^{-h} + (.84)2^{-h}$

(c) $x_t - 1.2x_{t-1} + .85x_{t-2} = w_t$.

$$\phi(z) = 1 - 1.2z + .85z^2$$
$$z = -\frac{1.2 \pm \sqrt{1.44 - 4(.85)(1)}}{2(.85)} = .706 \pm .8235i$$
$$|z| = \sqrt{.706^{+}.8235^2} = 1.0847$$
$$\theta = .86206$$
$$z = 1.0847cos.86206 + 1.0847sin.86206i$$
$$\rho(h) = a1.0847^{-h}cos(\theta h + b)$$
$$\rho(0) = acosb$$
$$\rho(1) = \frac{1}{1.0847}cos(\theta + b) = \frac{1.2}{1 + .85} = .64865$$
$$acos(\theta + b) = .703589$$
$$a[cos\theta cosb - sin\theta sinb] = cos\theta - sin\theta tanb = .703589$$
$$tanb = -.06941367$$
$$b = -.06930251$$
$$a = 1.0024$$
$$\rho(h) = (1.0024)1.0847^{-h}cos(.86206h - .06930251)$$

```
acfc = function(h) {
  rho_h = 1.0024*(1.0847^-h)*cos(.86206*h-.06932373)
  return(rho_h)
}

x=1:60
plot(x, acfc(x), type="h", ylab="ACF", xlab="Lag", main="ACF of AR with
Complex Roots")
abline(h=0)
```

AR models with complex roots will appear to be cyclical in nature.



3.8

3.8 Verify the calculations for the autocorrelation function of an ARMA(1, 1) process given in Example 3.14. Compare the form with that of the ACF for the ARMA(1, 0) and the ARMA(0, 1) series. Plot the ACFs of the three series on the same graph for $\phi = .6$, $\theta = .9$, and comment on the diagnostic capabilities of the ACF in this case.

From 3.48,

$\gamma(0) = \phi\gamma(1) + \sigma_w^2\theta_0\psi_0 + \sigma_w^2\theta_1\psi_1$

$\gamma(1) = \psi\gamma(0) + \sigma_w^2\theta$

And from the results from Example 3.12,

$\psi_0 = 1$, $\psi_1 = \theta_1 + \phi_1$

Hence,

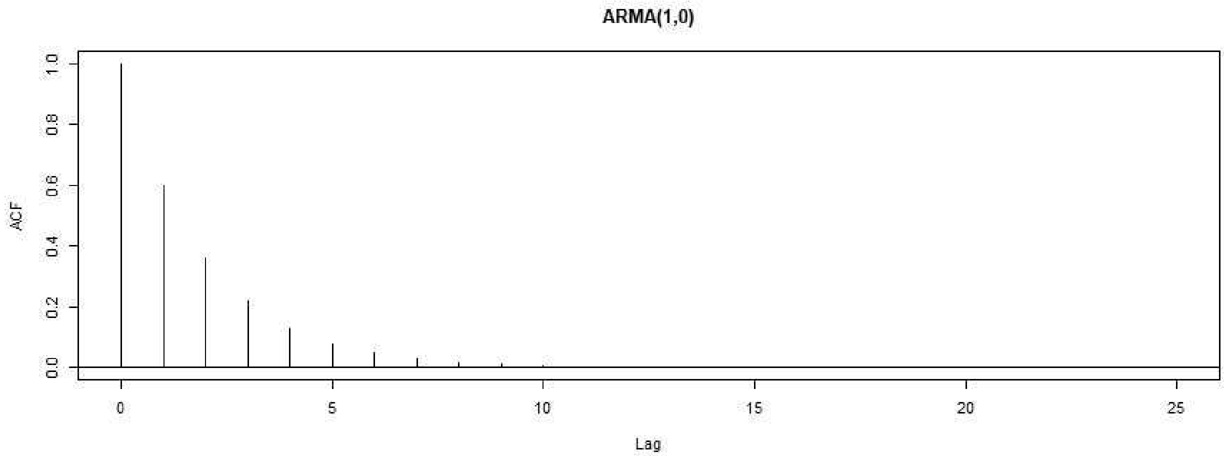
$$\gamma(0) = \psi\gamma(1) + \sigma_w^2[1 + \phi\theta + \theta^2]$$
$$= \phi(\psi\gamma(0) + \sigma_w^2) + \sigma_w^2[1 + \phi\theta + \theta^2]$$
$$= \sigma_w^2 \frac{1 + 2\phi\theta + \theta^2}{1 - \phi^2}$$
$$\gamma(1) = \phi\sigma_w^2 \frac{1 + 2\phi\theta + \theta^2}{1 - \phi^2} + \sigma_w^2\theta$$
$$= \frac{\sigma_w^2(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2}$$

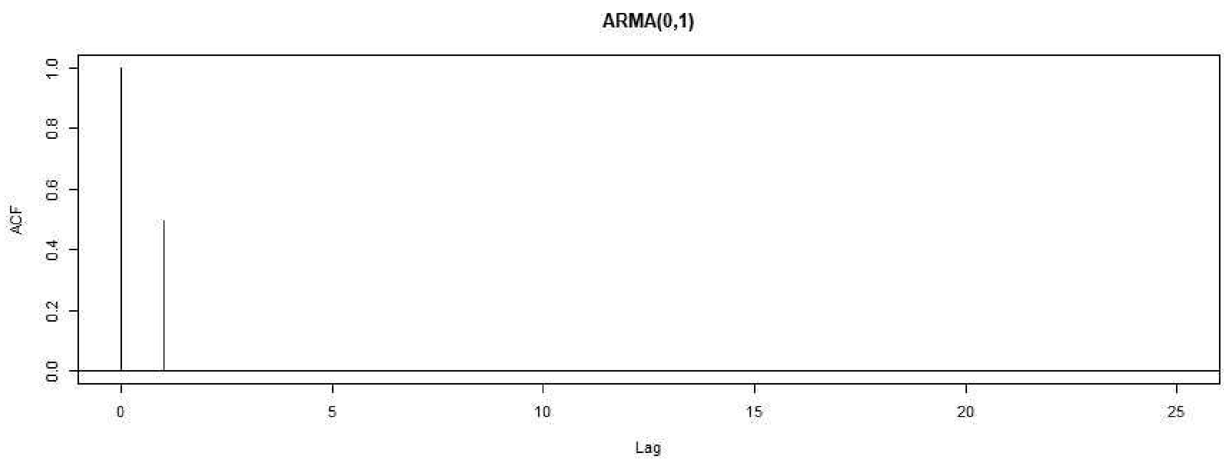
Hence,

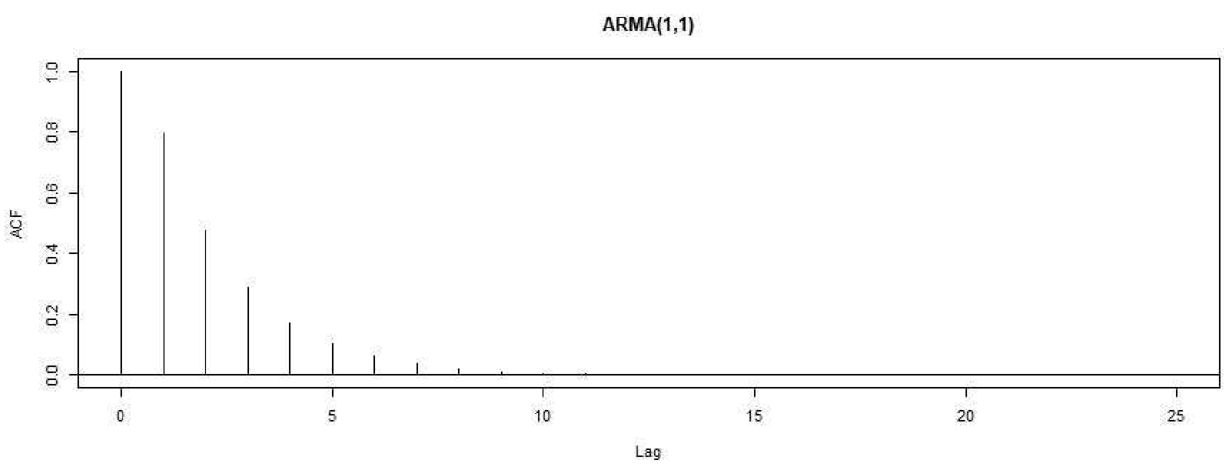
$$\rho(h) = \frac{(1 - \theta\phi)(\phi + \theta)}{1 + 2\theta\phi + \phi^2}\phi^{h-1}, \ h > 1.$$

The ACF of an ARMA(1,0) is $\rho(h) = \phi^h$ and for an ARMA(0,1) it's $\rho(h) = \frac{\theta}{1 + \theta^2}$, $h = 1$, $\rho(h) = 0$, $h > 1$. The ARMA(1, 0) has an ACF of the same form as an ARMA(1,1), while the ARM(0,1) ACF has piecewise constant form, as it is not a function of h.

```
phi = .6; theta = .9
model1 = ARMAacf(ar=c(phi), lag.max=25)
model2 = ARMAacf(ma=c(theta), lag.max=25)
model3 = ARMAacf(ar=c(phi), ma=(theta), lag.max=25)
par(mfrow=c(3,1))
plot(0:25, model1, type="h", ylab="ACF", xlab="Lag", main="ARMA(1,0)")
abline(h=0)
plot(0:25, model2, type="h", ylab="ACF", xlab="Lag", main="ARMA(0,1)")
abline(h=0)
plot(0:25, model3, type="h", ylab="ACF", xlab="Lag", main="ARMA(1,1)")
abline(h=0)
```







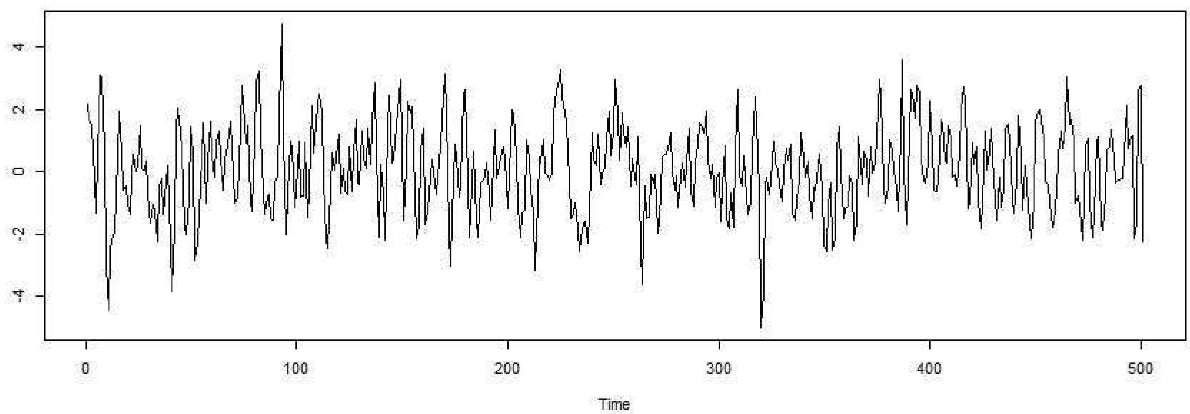
We can see that the ACF of the ARMA(1,0) and the ARMA(1,1) are very similar. They are not exactly the same, as they have different values, but we can observe an exponential reduction in the value of the ACF per lag. This is contrasted with the ARMA(0,1), where the ACF does not decrease exponentially, but rather cuts off at lag 1.

3.9

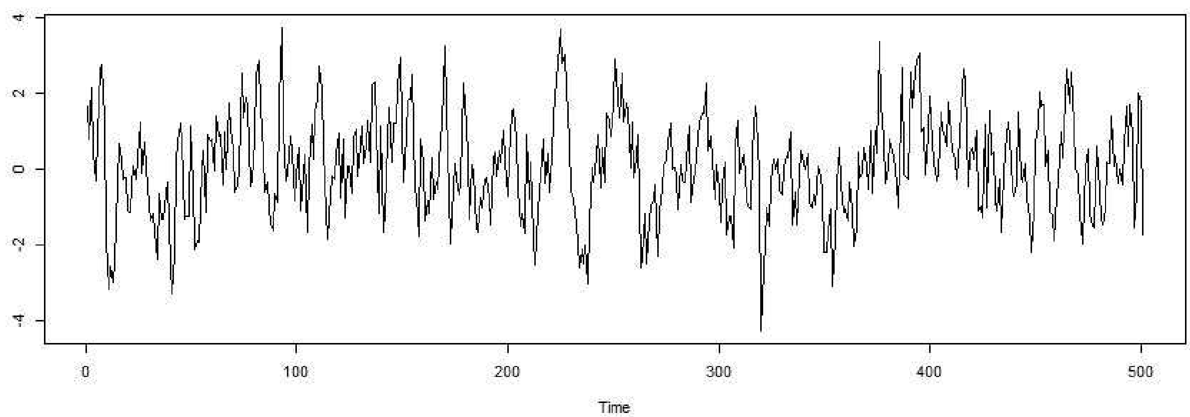
3.9 Generate $n = 100$ observations from each of the three models discussed in Problem 3.8. Compute the sample ACF for each model and compare it to the theoretical values. Compute the sample PACF for each of the generated series and compare the sample ACFs and PACFs with the general results given in Table 3.1.

```
par(mfrow=c(3,1))
phi=.6; theta=.9
n=500; m=n+50
w = rnorm(m)

model1 <- model2 <- model3 <- rep(0, m)
model2[1] = model3[1] = w[1]
for(i in 2:m) {
  model1[i] = theta*w[i-1] + w[i]
  model2[i] = phi*model2[i-1] + w[i]
  model3[i] = phi*model2[i-1] + theta*w[i-1] + w[i]
}

model1 = model1[50:m]; model2 = model2[50:m]; model3 = model3[50:m]

plot.ts(model1, main="MA(1)", ylab='')
plot.ts(model2, main="AR(1)", ylab='')
plot.ts(model3, main="ARMA(1,1)", ylab='')

acf(model1, main="ACF of MA(1)")
acf(model2, main="ACF of AR(1)")
acf(model3, main="ACF of ARMA(1,1)")

acf(model1, main="PACF of MA(1)", type="partial")
acf(model2, main="PACF of AR(1)", type="partial")
acf(model3, main="PACF of ARMA(1,1)", type="partial")
```

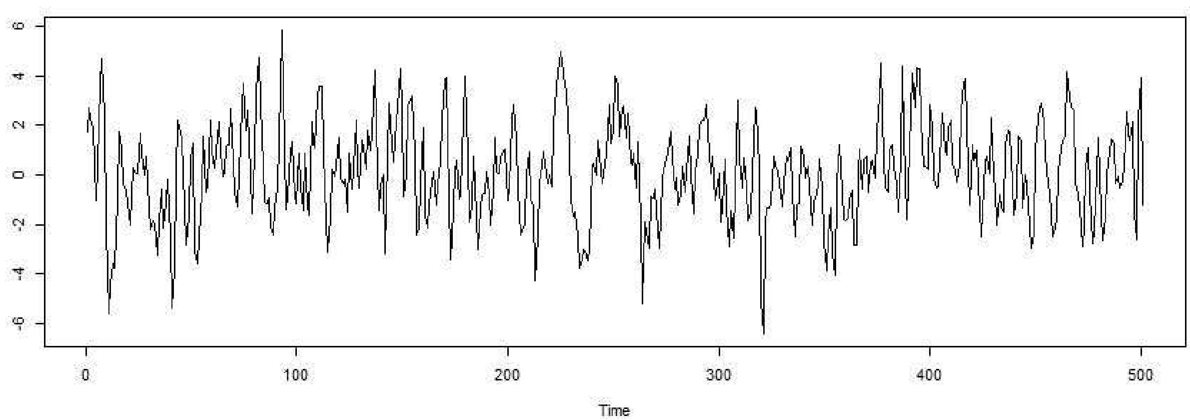
MA(1)

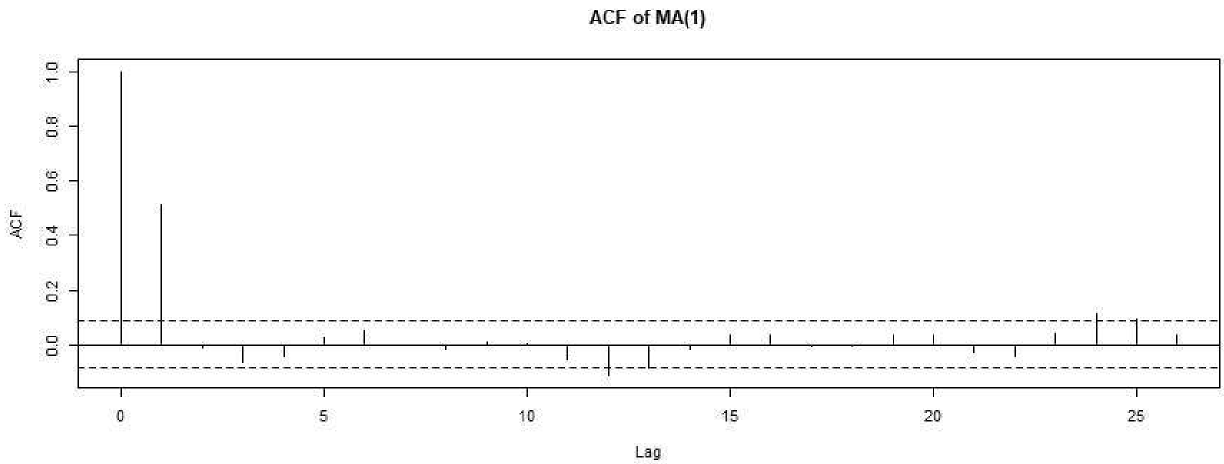


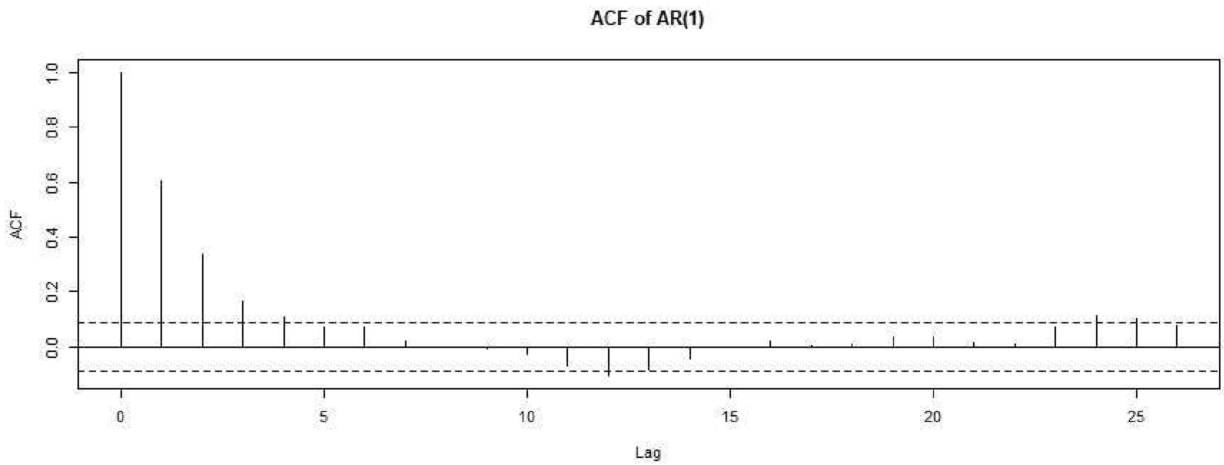
AR(1)

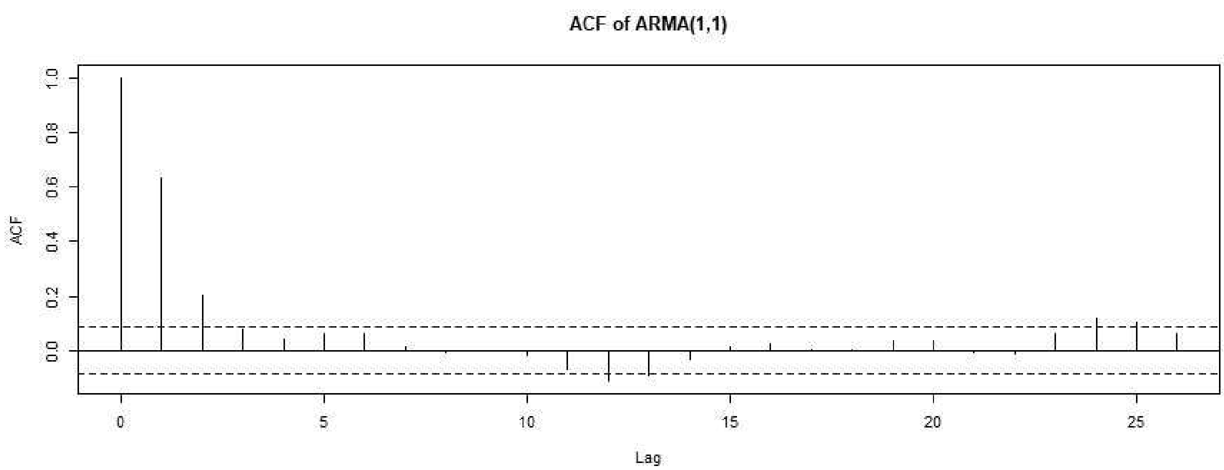


ARMA(1,1)

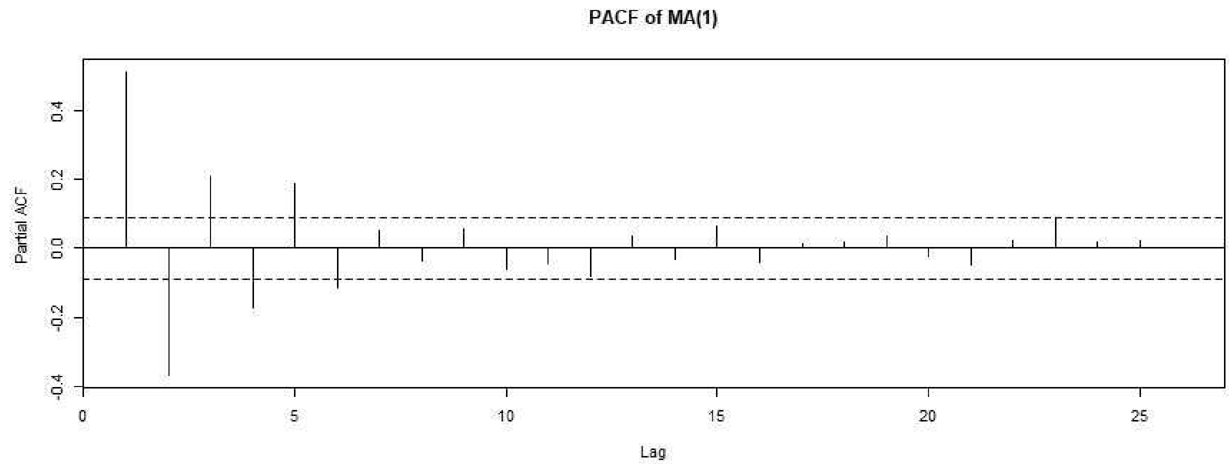


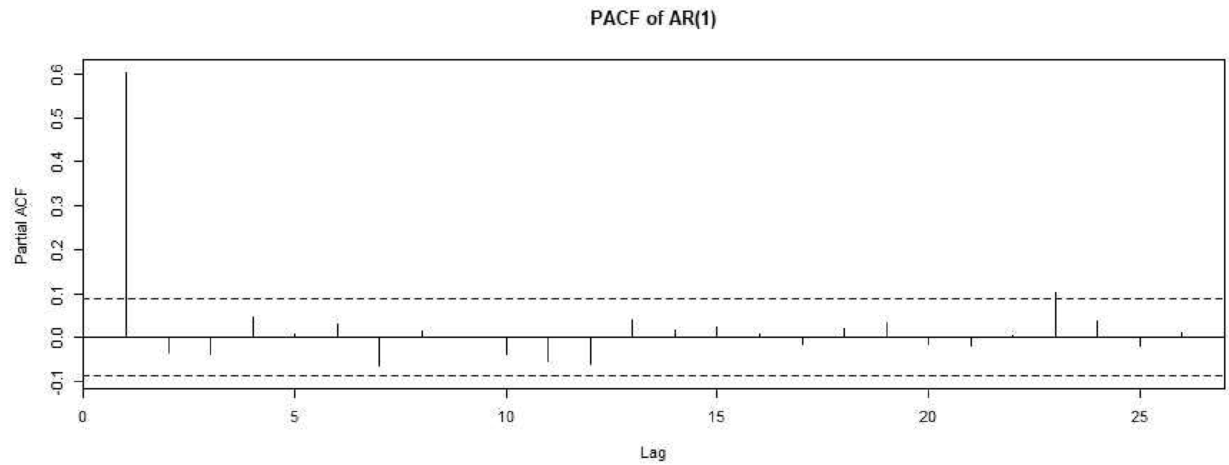


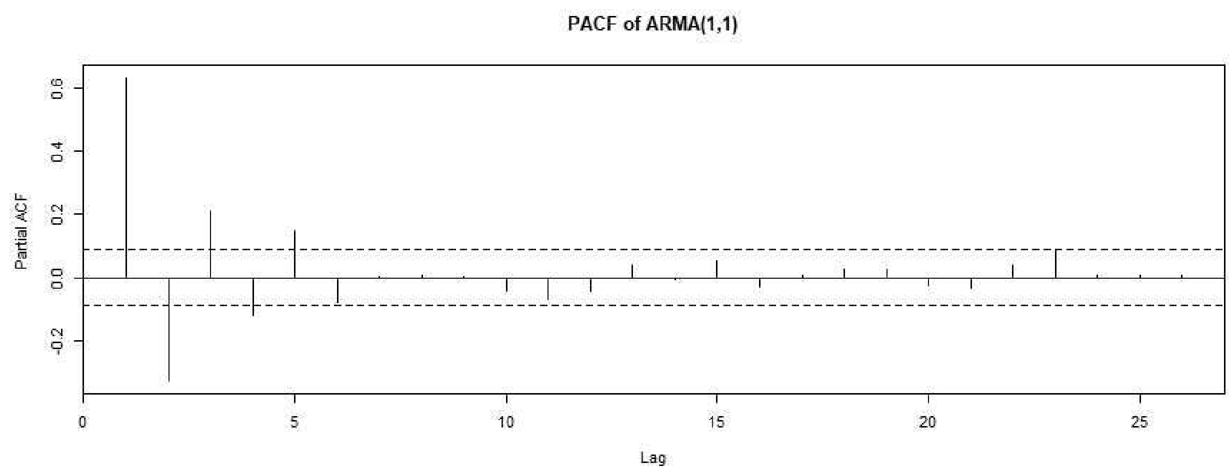




We see that the sample ACF of the ARMA(0,1) model matches closely the theoretical ACF, as it cuts off after lag 1, and is roughly equal to 0.49. The sample ACF of the ARMA(1,0) model is similar to the theoretical one as well, since it is decreasing exponentially. Same conclusion for ARMA(1,1).





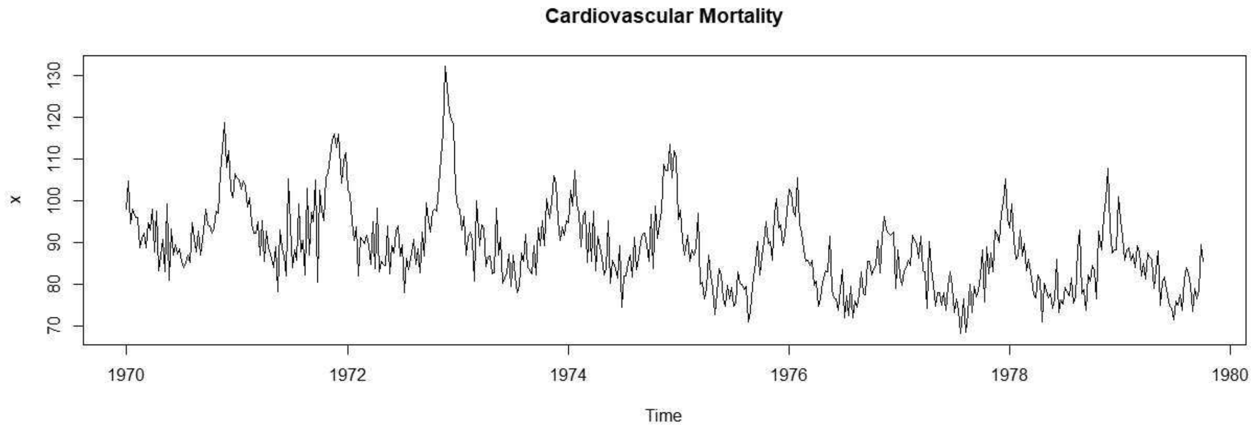


For the MA(1) model, Table 3.1 states that the ACF cuts off after lag q and tails off on the PACF. This is what we observe. Similar conclusion for the AR(1) and ARMA(1,1).

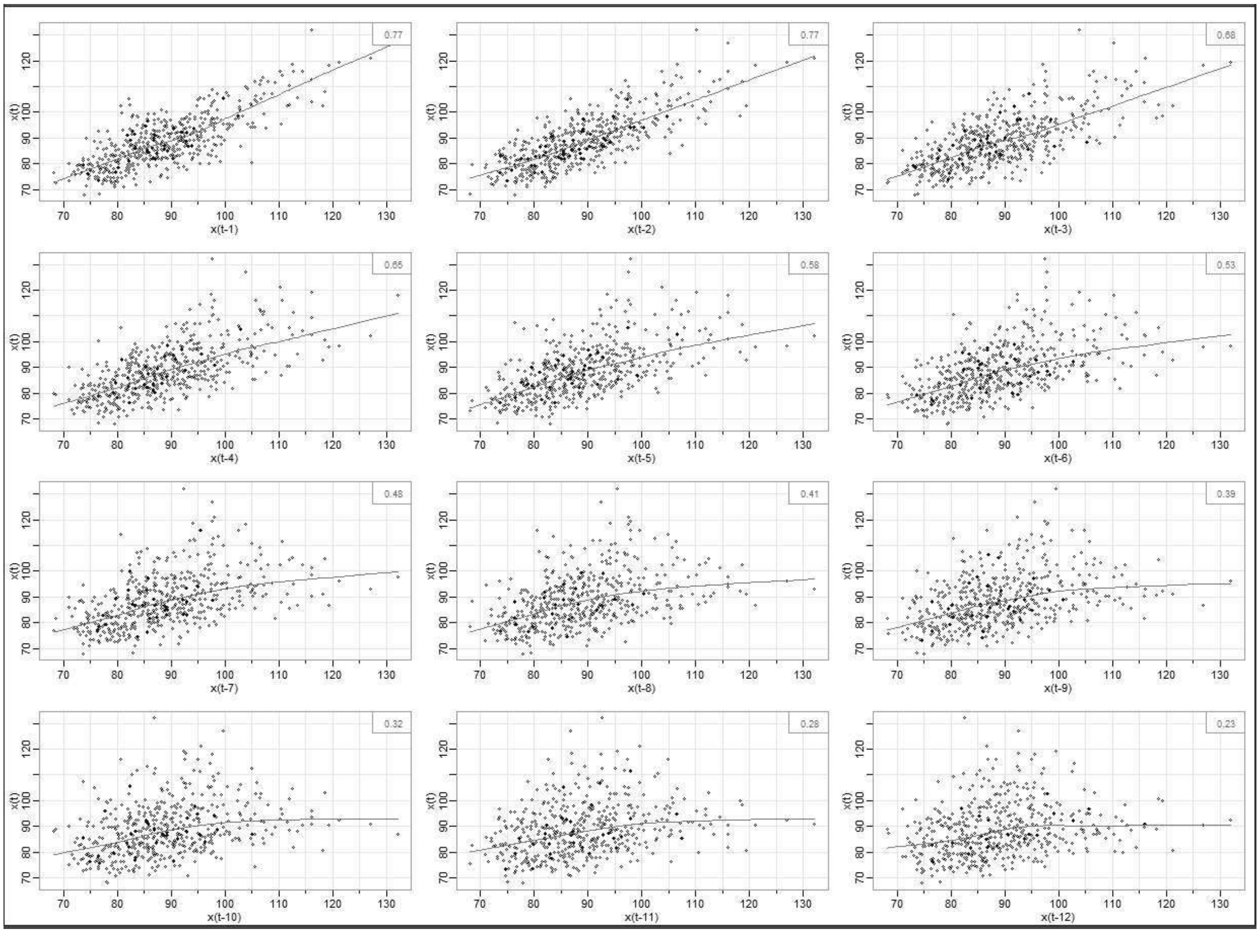
3.10

3.10 Let x_t represent the cardiovascular mortality series (`cmort`) discussed in Example 2.2.

(a) Fit an AR(2) to x_t using linear regression as in Example 3.18.

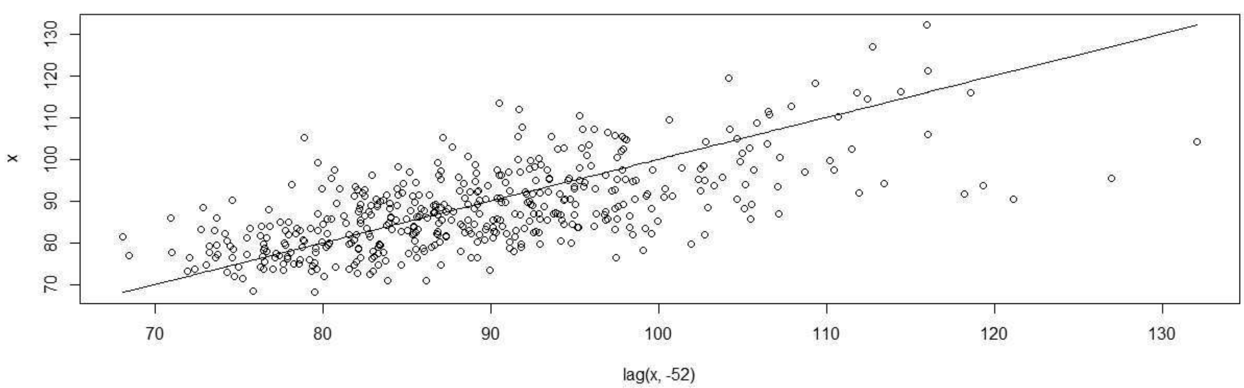


Upon initially plotting data, we can see a slight down trend over the period with seasonal variation occurring yearly. There is an outlier in 1973 but other than that the variance appears relatively consistent.

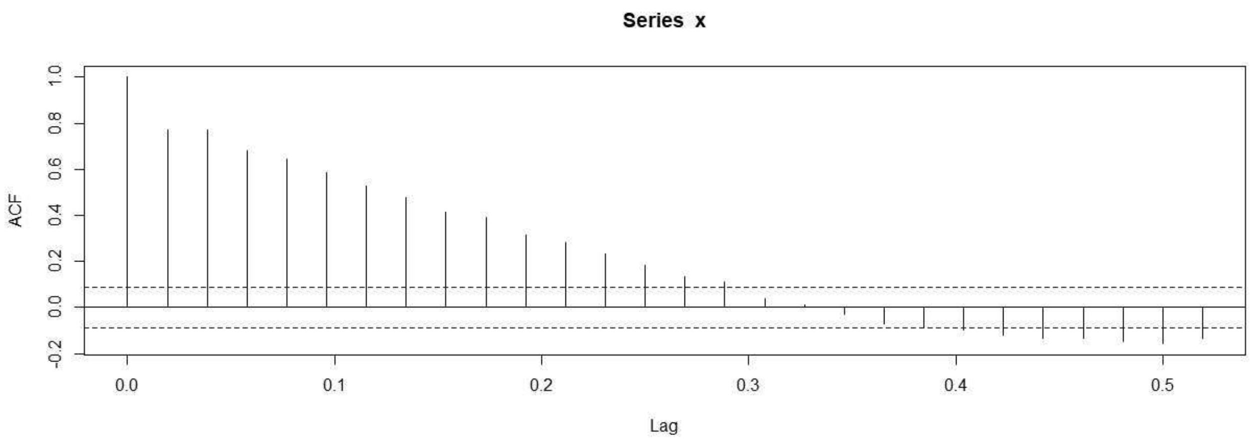


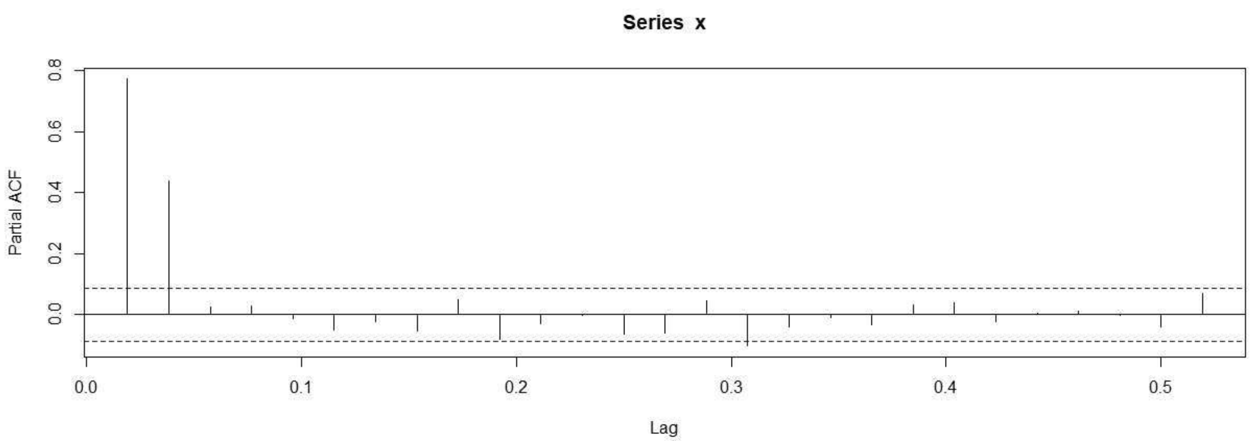
Plotting scatter plots between the first 12 lags, we can see strong linear relationships in the first 3 lags. However, diminishing correlation is observed in subsequent lags.

We also see a moderately strong relationship at lag 52.



We see a gradual tail off in the ACF and a fairly sharp cut off at lag 2 in the PACF. This suggests an AR(2).



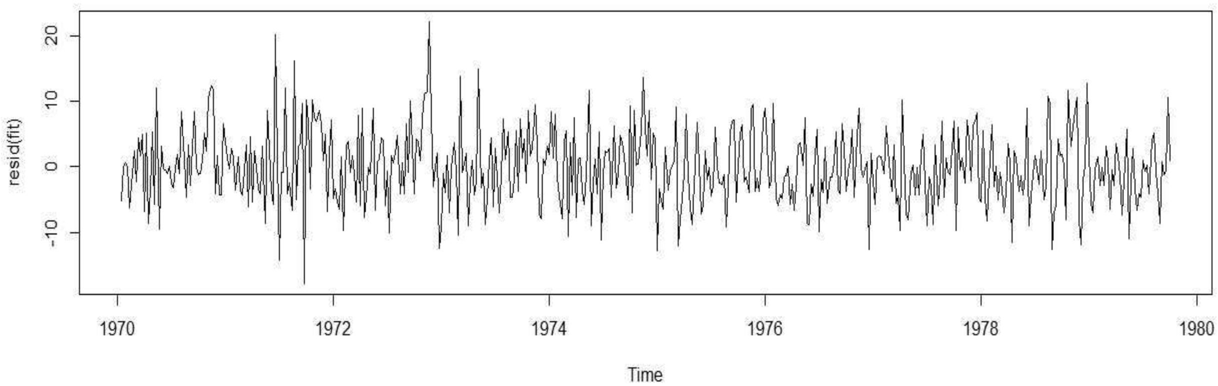


Regressing on the first two lags gives coefficients phi1 = 0.4286 and phi2 = 0.4418 and a mean of 11.45061. All three parameters are significant.

```
lags = ts.intersect(x, xL1=lag(x, -1), xL2=lag(x, -2), dframe=TRUE)
fit = lm(x ~ xL1 + xL2, data=lags, na.action=NULL)

Residuals:
     Min       1Q   Median       3Q      Max
-17.8192  -4.0339  -0.2112   3.4219  22.1840

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.45061    2.40080   4.769 2.42e-06 ***
xL1          0.42859    0.03991  10.738  < 2e-16 ***
xL2          0.44179    0.03988  11.078  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.702 on 503 degrees of freedom
```

```
Multiple R-squared:  0.6752,  Adjusted R-squared:  0.6739
F-statistic: 522.8 on 2 and 503 DF,  p-value: < 2.2e-16
```

Plotting the residuals we can see that the residuals have no discernible trend and appear relatively normal except for a few outliers as per the QQ plot (not shown).



Running a regression on the first three lags provides a statistically insignificant parameter estimate. Hence, despite there being a relatively strong linear relationship between x and lag 3, as demonstrated by the above scatterplot, the PACF confirms that this relationship is completely explained by lag 2.

```
lags = ts.intersect(x, xL1=lag(x, -1), xL2=lag(x, -2), xL3=lag(x, -3),
drame=TRUE)
fit = lm(x ~ xL1 + xL2 + xL3, data=lags, na.action=NULL)

Residuals:
     Min       1Q   Median       3Q      Max
-17.8256  -4.0106  -0.3361   3.4433  22.2910

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.04727    2.45730   4.496 8.62e-06 ***
xL1          0.41958    0.04463   9.402  < 2e-16 ***
xL2          0.42994    0.04435   9.695  < 2e-16 ***
xL3          0.02551    0.04463   0.571    0.568
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.707 on 501 degrees of freedom
Multiple R-squared:  0.6757,  Adjusted R-squared:  0.6738
F-statistic:   348 on 3 and 501 DF,  p-value: < 2.2e-16
```

(b) Assuming the fitted model in (a) is the true model, find the forecasts over a four-week horizon, x^n_{n+m}, for $m = 1, 2, 3, 4$, and the corresponding 95% prediction intervals.

```
predict(regr, n.ahead=4)

$pred
Time Series:
Start = c(1979, 41)
End = c(1979, 44)
Frequency = 52
[1] 87.59986 86.76349 87.33714 87.21350

$se
Time Series:
Start = c(1979, 41)
End = c(1979, 44)
Frequency = 52
[1] 5.684848 6.184973 7.134227 7.593357
```

3.11

3.11 Consider the MA(1) series

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

where w_t is white noise with variance σ^2_w.

(a) Derive the minimum mean-square error one-step forecast based on the infinite past, and determine the mean-square error of this forecast.

We can write an MA(1) as:

$$w_t = \sum_{j=0}^{\infty} (-\theta)^j x_{t-j}$$

Hence,

$$w_{n+1} = \sum_{j=0}^{\infty} (-\theta)^j x_{n+1-j} = x_{n+1} + \sum_{j=1}^{\infty} (-\theta)^j x_{n+1-j}$$

$$x_{n+1} = w_{n+1} - \sum_{j=1}^{\infty} (-\theta)^j x_{n+1-j}$$

Hence,

$$\tilde{x}_{n+1} = -\sum_{j=1}^{\infty}(-\theta)^j x_{n+1-j}$$

Since, the conditional expectation of x_{n+1-j} on $x_n, x_{n-1}, \cdots$ when $t \leq n$ is itself. And the conditional of w_{n+1} on the same is 0.

Then,

$$x_{n+1} - \tilde{x}_{n+1} = w_{n+1} - \sum_{j=1}^{\infty}(-\theta)^j x_{n+1-j} + \sum_{j=1}^{\infty}(-\theta)^j x_{n+1-j} = w_{n+1}$$

Hence,

$$MSE = \mathbb{E}[(x_{n+1} - \tilde{x}_{n+1})^2] = \mathbb{E}[w_{n+1}^2] = \sigma_w^2$$

(b) Let $\tilde{x}_{n+1}^n$ be the truncated one-step-ahead forecast as given in (3.92). Show that

$$\mathrm{E}\left[(x_{n+1} - \tilde{x}_{n+1}^n)^2\right] = \sigma^2(1 + \theta^{2+2n}).$$

Compare the result with (a), and indicate how well the finite approximation works in this case.

Using (3.92),

$$\tilde{x}_{n+1}^n = -\sum_{j=1}^{n+m-1}(-\theta)^j x_{n+1-j}$$

Hence,

$$x_{n+1} - \tilde{x}_{n+1}^n = w_{n+1} - \sum_{n+1}^{(} -\theta)^j x_{n+1-j}$$

$$= w_{n+1} - (-\theta)^{n+1}\sum_{n+1}^{(} -\theta)^j - (n+1)x_{n+1-j}$$

$$= w_{n+1} - (-\theta)^{n+1}\sum_{0}^{(} -\theta)^j x_{-j} = w_{n+1} - (-\theta)^{n+1}w_0$$

Hence,

$$\mathbb{E}[(x_{n+1} - \tilde{x}_{n+1}^n)^2] = \mathbb{E}[(w_{n+1} - (-\theta)^{n+1}w_0)^2] = \sigma_w^2 + \theta^{2(n+1)}\sigma_w^2$$

Which gives the result.

3.12

3.12 In the context of equation (3.63), show that, if $\gamma(0) > 0$ and $\gamma(h) \to 0$ as $h \to \infty$, then Γ_n is positive definite.

We know that Γ_n is non-negative definite as per Problem 1.25.

…

See solutions manual. I don't see why the matrix should have positive eigenvalues.

3.13

3.13 Suppose x_t is stationary with zero mean and recall the definition of the PACF given by (3.55) and (3.56). That is, let

$$\epsilon_t = x_t - \sum_{i=1}^{h-1} a_i x_{t-i} \quad \text{and} \quad \delta_{t-h} = x_{t-h} - \sum_{j=1}^{h-1} b_j x_{t-j}$$

be the two residuals where $\{a_1, \ldots, a_{h-1}\}$ and $\{b_1, \ldots, b_{h-1}\}$ are chosen so that they minimize the mean-squared errors

$$\mathrm{E}[\epsilon_t^2] \quad \text{and} \quad \mathrm{E}[\delta_{t-h}^2].$$

The PACF at lag h was defined as the cross-correlation between ϵ_t and δ_{t-h}; that is,

$$\phi_{hh} = \frac{\mathrm{E}(\epsilon_t \delta_{t-h})}{\sqrt{\mathrm{E}(\epsilon_t^2)\mathrm{E}(\delta_{t-h}^2)}}.$$

Let R_h be the $h \times h$ matrix with elements $\rho(i-j)$ for $i, j = 1, \ldots, h$, and let $\rho_h = (\rho(1), \rho(2), \ldots, \rho(h))'$ be the vector of lagged autocorrelations, $\rho(h) = \mathrm{corr}(x_{t+h}, x_t)$. Let $\tilde{\rho}_h = (\rho(h), \rho(h-1), \ldots, \rho(1))'$ be the reversed vector. In addition, let x_t^h denote the BLP of x_t given $\{x_{t-1}, \ldots, x_{t-h}\}$:

$$x_t^h = \alpha_{h1} x_{t-1} + \cdots + \alpha_{hh} x_{t-h},$$

as described in Property 3.3. Prove

$$\phi_{hh} = \frac{\rho(h) - \tilde{\rho}'_{h-1} R^{-1}_{h-1} \rho_h}{1 - \tilde{\rho}'_{h-1} R^{-1}_{h-1} \tilde{\rho}_{h-1}} = \alpha_{hh}.$$

In particular, this result proves Property 3.4.

Hint: Divide the prediction equations [see (3.63)] by $\gamma(0)$ and write the matrix equation in the partitioned form as

$$\begin{pmatrix} R_{h-1} & \tilde{\rho}_{h-1} \\ \tilde{\rho}'_{h-1} & \rho(0) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_{hh} \end{pmatrix} = \begin{pmatrix} \rho_{h-1} \\ \rho(h) \end{pmatrix},$$

where the $h \times 1$ vector of coefficients $\alpha = (\alpha_{h1}, \dots, \alpha_{hh})'$ is partitioned as $\alpha = (\alpha'_1, \alpha_{hh})'$.

Starting with (3.63), we write in terms of α_{hh}:

$\Gamma_h \alpha_{hh} = \gamma_h$

And then divide both sides by $\gamma(0)$, we have

$R_h \alpha_{hh} = \rho(h)$

$$= \begin{bmatrix} \rho(0) & ... & \rho(1-h) \\ ... & & ... \\ \rho(1-h) & ... & \rho(0) \end{bmatrix} \begin{bmatrix} \alpha_{h1} \\ ... \\ \alpha_{hh} \end{bmatrix} = \begin{bmatrix} \rho(1) \\ ... \\ \rho(h) \end{bmatrix}$$

Which gives the above form. Then we can write

$R_{h-1}\alpha_{h1} + \tilde{\rho}_{h-1}\alpha_{hh} = \rho_{h-1}$

$\tilde{\rho}'_{h-1}\alpha_{h1} + \rho(0)\alpha_{hh} = \rho_h$

Given $\rho(0) = 1$, this gives the desired form for α_{hh}.

To prove the result, we calculate ϕ_{hh} from (3.56) as per the above form.

To derive the prediction equations for x_t, we use (3.62)

$$\mathbb{E}[(x_t - \sum_{j=1}^{h-1} a_j x_{t-j}) x_{t-k}] = 0 \;\; for \;\; k = 1, ..., h-1$$

.

$\Gamma_{h-1} a_{h-1} = \gamma_{h-1}$

And prediction derivatives for x_{t-h}:

$$\mathbb{E}[(x_{t-h} - \sum_{j=1}^{h-1} a_j x_{t-j}) x_{t-k}] = 0 \;\; for \;\; k = 1, , h-1$$

$\Gamma_{h-1}a_{h-1} = \tilde{\gamma}_{h-1}$

Hence, let $x = [x_{t-1}, ..., x_{t-h+1}]'$

$$\epsilon_t = x_t - \gamma'_{h-1}\Gamma^{-1}_{h-1}x$$
$$\delta_{t-h} = x_{t-h} - \tilde{\gamma}'_{h-1}\Gamma^{-1}_{h-1}x$$
$$\mathbb{E}[\epsilon_t\delta_{t-h}] = \gamma(h) - \tilde{\gamma}'_{h-1}\Gamma^{-1}_{h-1}\gamma_{h-1}$$
$$\mathbb{E}[\delta^2_{t-h}] = \gamma(0) - \tilde{\gamma}'_{h-1}\Gamma^{-1}_{h-1}\tilde{\gamma}_{h-1}$$
$$\mathbb{E}[\epsilon^2_t] = \gamma(0) - \tilde{\gamma}'_{h-1}\Gamma^{-1}_{h-1}\tilde{\gamma}_{h-1}$$

Where the calculations are similar to that of (3.66). Where we have the facts:

$$(\Gamma^{-1})' = \Gamma^{-1}$$
$$\mathbb{E}[xx'] = \Gamma_{h-1}$$

3.14

3.14 Suppose we wish to find a prediction function $g(x)$ that minimizes

$$MSE = \mathrm{E}[(y - g(x))^2],$$

where x and y are jointly distributed random variables with density function $f(x, y)$.

(a) Show that MSE is minimized by the choice

$$g(x) = \mathrm{E}(y \mid x).$$

Hint:

$$MSE = \mathrm{E}\mathrm{E}[(y - g(x))^2 \mid x].$$

To find the solution, we can simply minimise the inner expectation.

Hence,

$$\frac{\partial \mathbb{E}[(y - g(x))^2|x]}{\partial g(x)} = -2[\mathbb{E}[y|x] - \mathbb{E}[g(x)|x]]$$
$$= -2[\mathbb{E}[y|x] - g(x)] = 0$$
$$g(x) = E[y|x]$$

(b) Apply the above result to the model

$$y = x^2 + z,$$

where x and z are independent zero-mean normal variables with variance one. Show that $MSE = 1$.

$g(x) = \mathbb{E}[y|x] = \mathbb{E}[x^2 + z|x] = \mathbb{E}[x^2|x] + \mathbb{E}[z|x] = x^2 + \mathbb{E}[z] = x^2$
$MSE = \mathbb{E}[(y - x^2)^2] = \mathbb{E}[(x^2 + z - x^2)^2] = \mathbb{E}[z^2] = 1$

(c) Suppose we restrict our choices for the function $g(x)$ to linear functions of the form

$$g(x) = a + bx$$

and determine a and b to minimize MSE. Show that $a = 1$ and

$$b = \frac{E(xy)}{E(x^2)} = 0$$

and $MSE = 3$. What do you interpret this to mean?

The prediction equations are:

$\mathbb{E}[y - g(x)] = 0$
$\mathbb{E}[(y - g(x))x] = 0$

Hence,

$\mathbb{E}[y] = \mathbb{E}[a + bx]$
$a = 1$
$\mathbb{E}[yx] = \mathbb{E}[(a + bx)x] = \mathbb{E}[ax] + b\mathbb{E}[x^2]$
$b = \frac{\mathbb{E}[yx]}{\mathbb{E}[x^2]}$
$\mathbb{E}[yx] = \mathbb{E}[x^3 + zx] = 0$
$g(x) = 1$

Hence,

$MSE = E[(y - 1)^2] = \mathbb{E}[y^2 - 2y + 1] = \mathbb{E}[x^4 + 2zx^2 + z^2 - 2x^2 - z + 1] = 3$

Where we have used the fact that $\mathbb{E}[x^4] = 3var(x)$.
We interpret this to mean that the BLP has an error three times as large as the of the optimal predictor.

3.15

3.15 For an AR(1) model, determine the general form of the m-step-ahead forecast x_{t+m}^t and show

$$E[(x_{t+m} - x_{t+m}^t)^2] = \sigma_w^2 \frac{1 - \phi^{2m}}{1 - \phi^2}.$$

The general form of an m-step-ahead forecast for an AR model can be written using (3.84)

$\tilde{x}_{n+m} = \sum_{j=0}^{\infty} \phi_j \tilde{w}_{n+m-j}$

If it's an AR(1), then $\phi(z) = 1 - \phi z$ and $\theta(z) = 1$, then using Property 3.3,

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{1}{1-\phi z} = \sum_{j=0}^{\infty} \phi^j z^j$$

Hence $\psi_j = \phi^j$ for AR(1) models and the general form becomes

$$\tilde{x}_{n+m} = \sum_{j=0}^{\infty} \phi^j \tilde{w}_{n+m-j}$$

Which means that we can use (3.86)

$$\mathbb{E}[(x_{n+m} - \tilde{x}_{n+m})^2] = \sigma_w^2 \sum_{j=0}^{m-1} \phi^{2j} = \sigma_w^2 \sum_{j=1}^{m} \phi^{2(j-1)} = \sigma_w^2 \frac{1-\phi^{2m}}{1-\phi^2}$$

3.16

3.16 Consider the ARMA(1,1) model discussed in Example 3.8, equation (3.27); that is, $x_t = .9x_{t-1} + .5w_{t-1} + w_t$. Show that truncated prediction as defined in (3.91) is equivalent to truncated prediction using the recursive formula (3.92).

To write (3.91) for the model, we need to calculate $\pi(z)$.

$$\pi(z) = \frac{\phi(z)}{\theta(z)} = \frac{1-.9z}{1+.5z}$$

$$1 - .9z = (1 + \pi_1 z + \pi_2 z^2 + ...)(1 + .5z) = 1 + (\pi_1 + .5)z + (.5\pi_1 + \pi_2)z^2 + ...$$

$$\pi_1 + .5 = -.9, \quad .5\pi_{j-1} + \pi_j = 0, \quad j > 1$$

$$\pi_j = -1.4(-.5)^{j-1}$$

Hence,

$$w_t = -1.4 \sum_{j=0}^{\infty} (-.5)^{j-1}$$

Since $\pi_0 = 1$ by definition, we have

$$x_t = 1.4 \sum_{j=1}^{\infty} (-.5)^{j-1} + w_t$$

Using (3.91), the truncated one-step-ahead predictor becomes

$$\tilde{x}_{n+1}^n = 1.4 \sum_{j=1}^{n} (-.5)^{j-1}$$

Then, using (3.92), we write

$$\tilde{x}_{n+1}^n = \phi \tilde{x}_{n+1}^n + \theta \tilde{w}_n^n$$

To write (3.92) in its complete form, we need to initialise $\tilde{w}_n^n$ using Property 3.7.

$$\tilde{x}_n^n = (1 - .9B)\tilde{x}_n^n - .5\tilde{w}_{n-1}^n$$
$$\tilde{x}_{n+1}^n = .9\tilde{x}_n^n + .5((1 - .9B)\tilde{x}_n^n - .5\tilde{w}_{n-1}^n)$$
$$= .9x_n + .5(x_n - .9x_{n-1} - .5\tilde{w}_{n-1}^n)$$
$$= 1.4x_n - .9(.5)x_{n-1} - .5\tilde{w}_{n-1}^n$$
$$= 1.4x_n - .9(.5)x_{n-1} - .5^2(x_{n-1} - .9x_{n-2} - .5\tilde{w}_{n-2}^n$$
$$= 1.4x_n - 1.4(.5)x_{n-1} + 1.4(.5^2)x + n - 2 - .9(.5^3)x_{n-3} - .5^4\tilde{w}_{n-3}^n$$
$$= 1.4\sum_{j=1}^{n}(-.5)^{j-1}x_{n+1-j}$$
Where we have used the facts:
$\tilde{x}_0^n = 0$ and $\tilde{w}_0^n = 0$.

To prove this for the m-step-ahead predictor, we use induction, where what we have shown already is the base case, as setting $m = 1$ in (3.91) yields the above form.

Assume:
$$\tilde{x}_{n+m}^n = 1.4\sum_{j=1}^{m-1}(-.5)^{j-1}\tilde{x}_{n+m-j}^n + 1.4\sum_{j=m}^{n+m-1}(-.5)^{j-1}x_{n+m-j} = .9x_{n+m-1}^n$$
Then, for the $m + 1$ case
$$\tilde{x}_{n+m+1}^n = 1.4\sum_{j=1}^{m}(-.5)^{j-1}\tilde{x}_{n+m+1-j}^n + 1.4\sum_{j=m+1}^{n+m}(-.5)^{j-1}x_{n+m+1-j}$$
$$= 1.4\sum_{j=2}^{m}(-.5)^{j-1}\tilde{x}_{n+m+1-j}^n + 1.4(-.5)^0\tilde{x}_{n+m}^n + 1.4\sum_{j=m+1}^{n+m}(-.5)^{j-1}x_{n+m+1-j}$$
$$= 1.4\sum_{j=1}^{m-1}(-.5)^{j}\tilde{x}_{n+m-j}^n + 1.4(-.5)^0\tilde{x}_{n+m}^n + 1.4\sum_{j=m}^{n+m-1}(-.5)^{j}x_{n+m-j}$$
$$= (-.5)1.4\sum_{j=1}^{m-1}(-.5)^{j-1}\tilde{x}_{n+m-j}^n + 1.4\tilde{x}_{n+m}^n + (-.5)1.4\sum_{j=m}^{n+m-1}(-.5)^{j-1}x_{n+m-j}$$
$$= (-.5)\tilde{x}_{n+m}^n + 1.4\tilde{x}_{n+m}^n = .9\tilde{x}_{n+m}^n$$
Which completes the proof.

3.17

3.17 Verify statement (3.87), that for a fixed sample size, the ARMA prediction errors are correlated.

$$x_{n+m} - \tilde{x}_{n+m} = \sum_{j=0}^{m-1} \psi_j w_{n+m-j}$$

$$x_{n+m+k} - \tilde{x}_{n+m+k} = \sum_{j=0}^{m+k-1} \psi_j w_{n+m+k-j}$$

$$\mathbb{E}[(x_{n+m} - \tilde{x}_{n+m})(x_{n+m+k} - \tilde{x}_{n+m+k})] = \mathbb{E}[\sum_{j=0}^{m-1} \psi_j w_{n+m-j} \sum_{l=0}^{m+k-1} \psi_l w_{n+m+k-l}]$$

$$= \mathbb{E}[\sum_{j=0}^{m-1} \sum_{j=0}^{m+k-1} \psi_l w_{n+m-l} \psi_l w_{n+m+k-l}] = \mathbb{E}[\sum_{j=0}^{m-1} \psi_j \psi_{j+k} w_{n+m-j} w_{n+m+k-j}]$$

$$= \sigma_w^2 \sum_{j=0}^{m-1} \psi_j \psi_{j+k}$$

3.18

3.18 Fit an AR(2) model to the cardiovascular mortality series (`cmort`) discussed in Example 2.2. using linear regression and using Yule–Walker.

(a) Compare the parameter estimates obtained by the two methods.
(b) Compare the estimated standard errors of the coefficients obtained by linear regression with their corresponding asymptotic approximations, as given in Property 3.10.

In Problem 3.10 we used linear regression to estimate the parameters for the AR(2). Here, we will use Yule-Walker estimation.

```
X = cmort; x.yw = ar.yw(x, lags=2)
x.yw$x.mean #mean
[1] 88.69888
x.yw$ar # coefficients
[1] 0.4339481 0.4375768
sqrt(diag(x.yw$asy.var.coef)) #standard errors
[1] 0.04001303 0.04001303
```

As we can see, the parameters are similar yet slightly different, within one standard error. The linear regression model produces a mean of 11.45 while YW gives 88.69. The standard errors for both methods are also similar.

3.19

3.19 Suppose $x_1, \dots, x_n$ are observations from an AR(1) process with $\mu = 0$.

(a) Show the backcasts can be written as $x_t^n = \phi^{1-t} x_1$, for $t \le 1$.

We can apply an amended version of Property 3.7 for backcasts, where the general model is $x_t = \phi x_{t+1} + w_t$.

We know that $x_1^n = x_1$, and hence

$x_0^n = \phi x_1,\ x_{-1}^n = \phi x_0^n = \phi^2 x_1$.

And we can derive the general form.

(b) In turn, show, for $t \le 1$, the backcasted errors are

$$\tilde{w}_t(\phi) = x_t^n - \phi x_{t-1}^n = \phi^{1-t}(1 - \phi^2) x_1.$$

Using an amended version of Property 3.7

$\tilde{w}_t^n = x_t^n - \phi x_{t-1}^n = \phi^{1-t} x_1 - \phi \phi^{1-t+1} x_1 = \phi^{1-t}(1 - \phi^2) x_1$

(d) Use the result of (c) to verify the unconditional sum of squares, $S(\phi)$, can be written as $\sum_{t=-\infty}^{n} \tilde{w}_t^2(\phi)$.

Using (3.108), we write

$$S(\phi) = (1 - \phi^2) x_1^2 + \sum_{t=2}^{n} [x_t - \phi x_{t-1}]^2$$

$$= \frac{1}{(1 - \phi^2)} (1 - \phi^2)^2 x_1^2 + \sum_{t=2}^{n} [x_t - \phi x_{t-1}]^2$$

$$= \sum_{t=-\infty}^{1} \phi^{2(1-t)} (1 - \phi^2)^2 x_1^2 + \sum_{t=2}^{n} [x_t - \phi x_{t-1}]^2$$

$$= \sum_{t=-\infty}^{1} \tilde{w}_t^2(\phi) + \sum_{t=2}^{n} [x_t - \phi x_{t-1}]^2$$

$$= \sum_{t=-\infty}^{n} \tilde{w}_t^2(\phi)$$

Where we have used the fact that

$$\sum_{t=-\infty}^{1} \phi^{1-t} = \sum_{t=-1}^{\infty} \phi^{t-1}$$

(e) Find x_t^{t-1} and r_t for $1 \le t \le n$, and show that

$$S(\phi) = \sum_{t=1}^{n} (x_t - x_t^{t-1})^2 \; / \; r_t.$$

We know that $x_t^{t-1} = \phi x_{t-1}$, and also $x_1^0 = \mathbb{E}[x_1] = 0$

hence,

$$S(\phi) = (1-\phi^2)(x_1 - x_1^0)^2 + \sum_{t=2}^{n}[x_t - x_t^{t-1}]^2$$

Now, we know from the equations after (3.116),

$$r_1 = \sum_{j=0}^{\infty} \psi_j^2 = \sum_{j=0}^{\infty} \phi^(2j) = \frac{1}{1-\phi^2}$$

$$r_2 = (1-\phi_{11}^2)r1 = (1-\phi^2)r1 = 1$$

And consequently, $r_t = 1$ for $t \geq 2$, since $\phi_{tt} = 0$ for $t \geq 2$.
Hence,

$$S(\phi) = \frac{x1 - x_1^0}{r_1} + \sum_{t=2}^{n} \frac{x_t - x_t^{t-1}}{r_t}$$

Which gives the required form.

3.20

3.20 Repeat the following numerical exercise three times. Generate $n = 500$ observations from the ARMA model given by

$$x_t = .9x_{t-1} + w_t - .9w_{t-1},$$

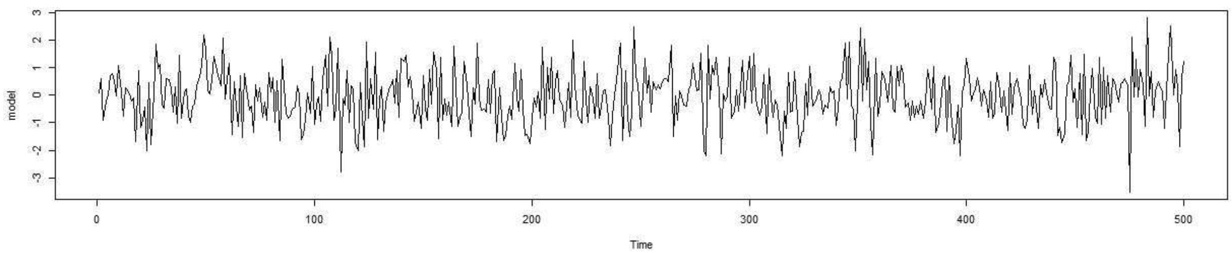
with $w_t \sim$ iid N(0, 1). Plot the simulated data, compute the sample ACF and PACF of the simulated data, and fit an ARMA(1, 1) model to the data. What happened and how do you explain the results?

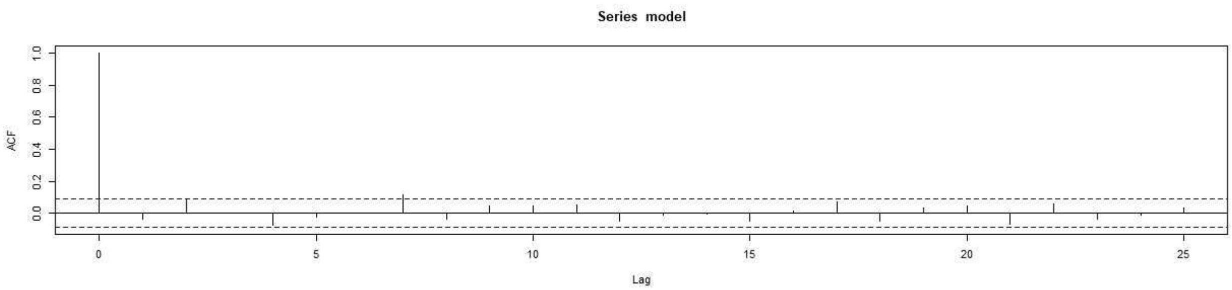
First, we create a simulation of the process $x_t = .9x_{t-1} + w_t - .9w_{t-1}$ using arima.sim, letting n = 500.

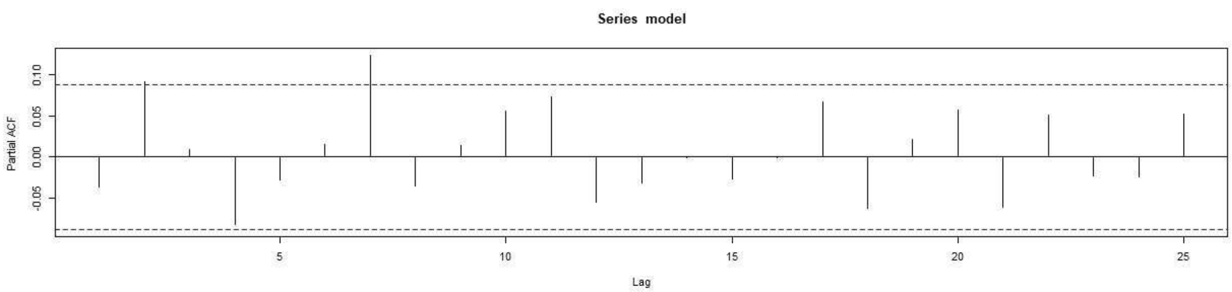
```
n = 500
model1 = arima.sim(list(order=c(1,0,1), ar=.9, ma=-.9), n) #simulated data
par(mfrow=c(3,1))
plot(model1)
acf(model1, lag.max=25)
```

```
pacf(model1, lag.max=25)
```

And we create plots for the data, the ACF and the PACF.





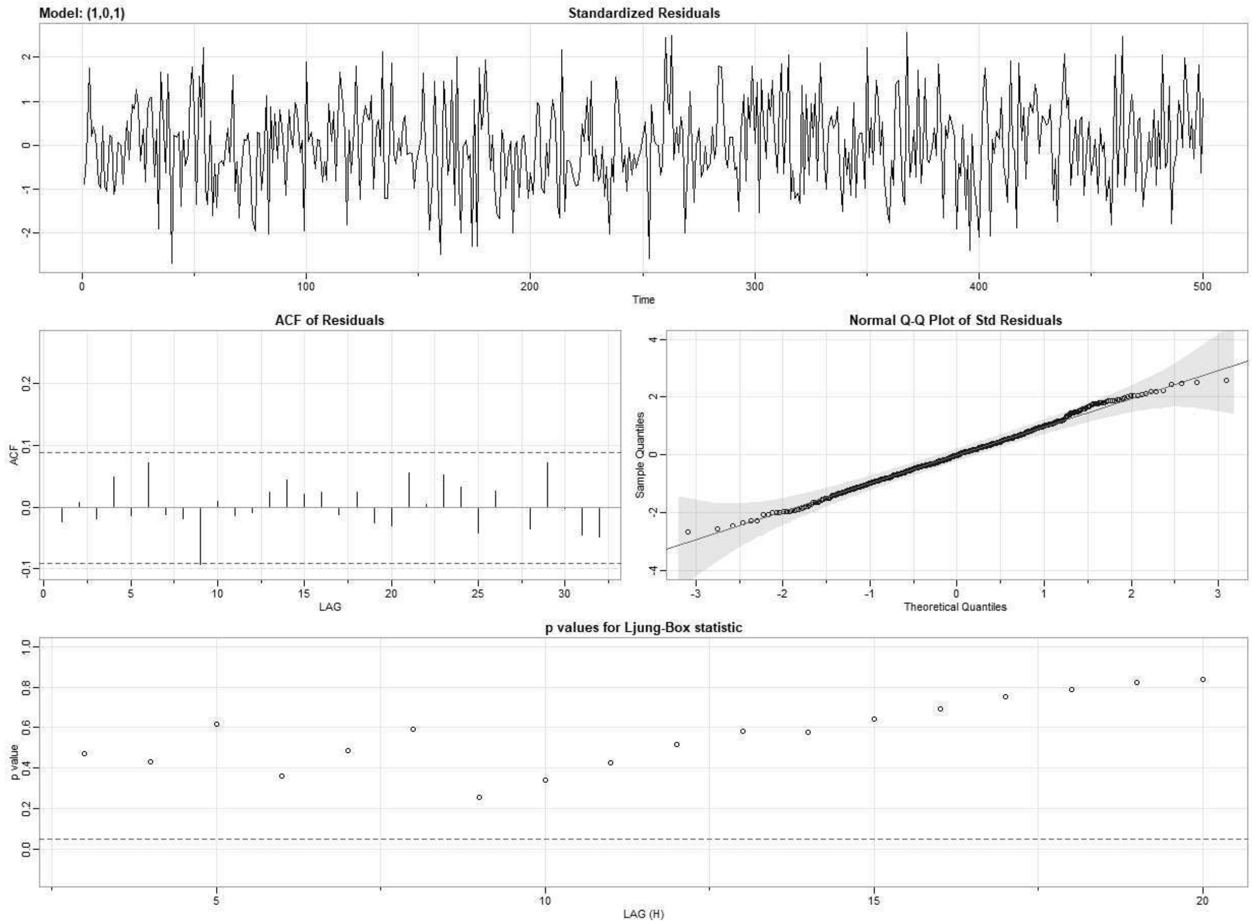


We can see that the ACF cuts off after lag 0, where the correlation is exactly 1. We can also see that the PACF neither cuts off or tails off, but instead is under the significance level for all lags. At lag 7 we have the PACF greater than the significant level, however, I am inclined to believe this is more random than anything interesting. From these observations we can conclude that the underlying process is neither an AR or an MA, but is simply white noise with standard deviation 1.

Fitting an ARMA(1,1) to the simulated data, we have

```
sarima(model1, 1,0,1)
$ttable
      Estimate     SE t.value p.value
ar1     0.8951 0.2026  4.4178  0.0000
ma1    -0.8716 0.2162 -4.0310  0.0001
xmean   0.0338 0.0562  0.6012  0.5480
```

Surprisingly, the fit gave estimates for both the MA and AR components of the model, with fairly uncorrelated residuals.



Interestingly, repeating the above simulation and fitting gives different estimates for the parameters, as we can see from the sarima output. We can also see that the standard errors and p-values have increased for the estimates.

```
model2 = arima.sim(list(order=c(1,0,1), ar=.9, ma=-.9), n) #simulated data
sarima(model2, 1,0,1) #fit using sarima
$ttable
      Estimate     SE t.value p.value
ar1    -0.2759 0.5261 -0.5243  0.6003
ma1     0.2427 0.5291  0.4588  0.6466
xmean   0.0329 0.0415  0.7937  0.4277
```

Repeating the simulation and fitting, we get roughly similar estimates and standard errors

```
model3 = arima.sim(list(order=c(1,0,1), ar=.9, ma=-.9), n) #simulated data
sarima(model3, 1,0,1) #fit using sarima
$ttable
      Estimate     SE t.value p.value
```

```
ar1     -0.3215 0.4967 -0.6473  0.5178
ma1      0.3445 0.4908  0.7018  0.4831
xmean    0.0124 0.0439  0.2834  0.7770
```

Something that I have just noticed is that in the last 2 simulations and fit, the sarima estimates give a negative value for the AR parameter. This is despite the model being generated from data using a positive AR parameter. However, we can observe in each model, the AR and MA parameters are approximately the negatives of each other.

3.21

3.21 Generate 10 realizations of length $n = 200$ each of an ARMA(1,1) process with $\phi = .9, \theta = .5$ and $\sigma^2 = 1$. Find the MLEs of the three parameters in each case and compare the estimators to the true values.

Given an ARMA(1,1) simulation with $\varphi = .9$, $\theta = .5$ and $\sigma_2 = 1$, we will fit a model using MLE. By default, the arima function on R uses MLE by optimising the full likelihood (rather than the conditional). If the model was an AR(p), we could have used ar.mle.

```
set.seed(40)
m = matrix(0, 200, 10) # each column is one simulation
rep(NA, 10) -> phi -> theta -> sigma_2
for (i in 1:10){
  m[,i] = arima.sim(n=200, list(ar=.9, ma=.5))
  fit = arima(m[,i], order=c(1,0,1))
  phi[i] = fit$coef[1]; theta[i] = fit$coef[2]; sigma_2[i] = fit$sigma2
}
> cbind("phi"=phi, "theta"=theta, "sigma_2"=sigma_2)
            phi     theta   sigma_2
 [1,] 0.8803791 0.4713839 1.0027522
 [2,] 0.8971766 0.3985445 1.0388389
 [3,] 0.8815393 0.4882599 0.7526775
 [4,] 0.8445527 0.4162428 1.0346161
 [5,] 0.8139391 0.5182211 0.9465428
 [6,] 0.9140090 0.5346935 1.0330617
 [7,] 0.7860247 0.5014307 1.0785225
 [8,] 0.8763894 0.4703178 1.0868521
 [9,] 0.9124741 0.4643437 0.9352032
[10,] 0.9043393 0.4906726 0.9942676
```

Running 10 simulations we find that the estimations are fairly close to the true parameters.

A more manual approach would be to use nlminb to minimise conditional error sum of squares (SSE_c). First, we can write the conditional error sum of squares as

$$S_c(\beta) = \sum_{t=p+1}^{n} w_t^2(\beta)$$

Where

$$w_t(\beta) = x_t - \sum_{j=1}^{p} \phi_j x_{t-j} - \sum_{k=1}^{q} \theta_k w_{t-k}(\beta)$$

Hence, we create the below function and run the data

```
arma_coef = function(x) {
  l = length(x)
  param=c(mu=0, phi=0, theta=0) # initialise for iteration

  SSE = function(param) {
    mu = param[1]
    phi = param[2]
    theta = param[3]

    res = vector()
    res[1] = 0 # we are condition on p=1
    for(i in 2:l){
      # p = q = 1
      res[i] = (x[i]-mu) - phi*(x[i-1]-mu) - theta*res[i-1]
    }
    return(sum(res*res))
  }
  bla = nlminb(objective=SSE, start=param)
  return(bla)
}
rep(NA, 10) -> phi1 -> theta1 -> sigma_2_1
for (i in 1:10){
  fit = arma_coef(m[,i])
  phi1[i] = fit$par[2]
  theta1[i] = fit$par[3]
  sigma_2_1[i] = fit$objective / length(m[,i])
}
> cbind("phi1"=phi1, "theta1"=theta1, "sigma_2_1"=sigma_2_1)
           phi1    theta1 sigma_2_1
 [1,] 0.8253833 0.4795349 0.9367026
 [2,] 0.9012714 0.3997941 1.0381209
 [3,] 0.8838976 0.4906550 0.7514762
 [4,] 0.8276038 0.4185296 1.0086467
 [5,] 0.8184970 0.5184126 0.9476708
 [6,] 0.8780305 0.5376220 0.9871102
```

```
 [7,] 0.7897150 0.5027594 1.0784332
 [8,] 0.8753721 0.4713160 1.0788653
 [9,] 0.9121669 0.4656902 0.9304635
[10,] 0.9070616 0.4922832 0.9929809
```

We can see that using the SSE_c does not yield as accurate results. Comparing our manual approach to the pure SSE method in arima shows that the estimates are very similar.

```
rep(NA, 10) -> phi2 -> theta2 -> sigma_2_2
for (i in 1:10){
  fit = arima(m[,i], order=c(1,0,1), method="CSS")
  phi2[i] = fit$coef[1]; theta2[i] = fit$coef[2]; sigma_2_2[i] = fit$sigma2
}
> cbind("phi2"=phi2, "theta2"=theta2, "sigma_2_2"=sigma_2_2)
           phi2    theta2 sigma_2_2
 [1,] 0.8253840 0.4795335 0.9414097
 [2,] 0.9012718 0.3997939 1.0433376
 [3,] 0.8838977 0.4906540 0.7552524
 [4,] 0.8276035 0.4185303 1.0137153
 [5,] 0.8184970 0.5184122 0.9524329
 [6,] 0.8780307 0.5376212 0.9920706
 [7,] 0.7897153 0.5027576 1.0838525
 [8,] 0.8753625 0.4713079 1.0842867
 [9,] 0.9121674 0.4656846 0.9351392
[10,] 0.9070616 0.4922831 0.9979708
```

An even more manual approach to the above would be to iterate the Gauss-Newton steps ourselves, which can be seen below. This method yields similar results to the above, not as accurate as using the full conditional with arima, but still reasonable.

```
get_root = function(vector) {
  roots = polyroot(vector)
  if (abs(roots[1]) < 1) {
    return(roots[1])
  } else if (abs(roots[2]) < 1) {
    return(roots[2])
  } else {
    return(NA)
  }
}

# Initiating estimates using method of moments
phi.start = rep(NA, 10)
theta.start = rep(NA, 10)
for (i in 1:10){
  r = acf(m[,i], lag=2, plot=FALSE)$acf
  g = acf(m[,i], lag=1, plot=FALSE, type="covariance")$acf
  corr1 = r[-1][1]
  corr2 = r[-1][2]
  cov0 = g[1]
  cov1 = g[-1]
  phi.start[i] = corr2 / corr1
  alpha = cov1 - phi.start[i]*cov0
  beta = cov0 - phi.start[i]*cov1 - alpha*phi.start[i]
  theta.start[i] = get_root(c(alpha, -beta, alpha))
}

c() -> sigma.final -> phi.final -> theta.final

# Gauss-Newton
for (j in 1:10) {

  niter = 100
  c(0) -> w
  rep(0, niter) -> Sz
  matrix(0, 2, n) -> z
  matrix(0, 2, niter) -> Szw
  matrix(0, 2, niter+1) -> params
  params[,1][1] = phi.start[j]
  params[,1][2] = theta.start[j]
  c() -> Sc

  for (p in 1:niter) {
```

```
    for (i in 2:n) {
      w[i] = m[,j][i] - params[,p][1]*m[,j][i-1] - params[,p][2]*w[i-1]
      # The below were calculated by taking the partial derivative of
      # wt with respect to the parameters
      z[,i][1] = m[,j][i-1] - params[,p][1]*z[,i-1][1]
      z[,i][2] = w[i-1] - params[,p][2]*z[,i-1][2]
    }
    Sc[p] = sum(w^2)
    for (i in 1:n){
      Sz[p] = Sz[p] + sum(z[,i]^2)
    }
    for (i in 1:n){
      Szw[,p] = Szw[,p] + z[,i]*w[i]
    }
    params[,p+1] = params[,p] + Szw[,p]/Sz[p]
  }

  phi.final[j] = params[,niter+1][1]
  theta.final[j] = params[,niter+1][2]
  sigma.final[j] = Sc[niter] / n
}

> cbind("phi.final"=phi.final, "theta.final"=theta.final,
"sigma.final"=sigma.final)
     phi.final theta.final sigma.final
 [1,] 0.8196512   0.4843233   0.9383488
 [2,] 0.9113345   0.3939651   1.0387059
 [3,] 0.8914721   0.4872418   0.7524274
 [4,] 0.8682304   0.4038544   1.0298166
 [5,] 0.8712627   0.4951104   0.9690702
 [6,] 0.8697188   0.5418819   0.9923057
 [7,] 0.7945319   0.5019432   1.0830653
 [8,] 0.8673587   0.4754783   1.0812585
 [9,] 0.9364344   0.4564789   0.9360327
[10,] 0.9300478   0.4875890   1.0082593
```

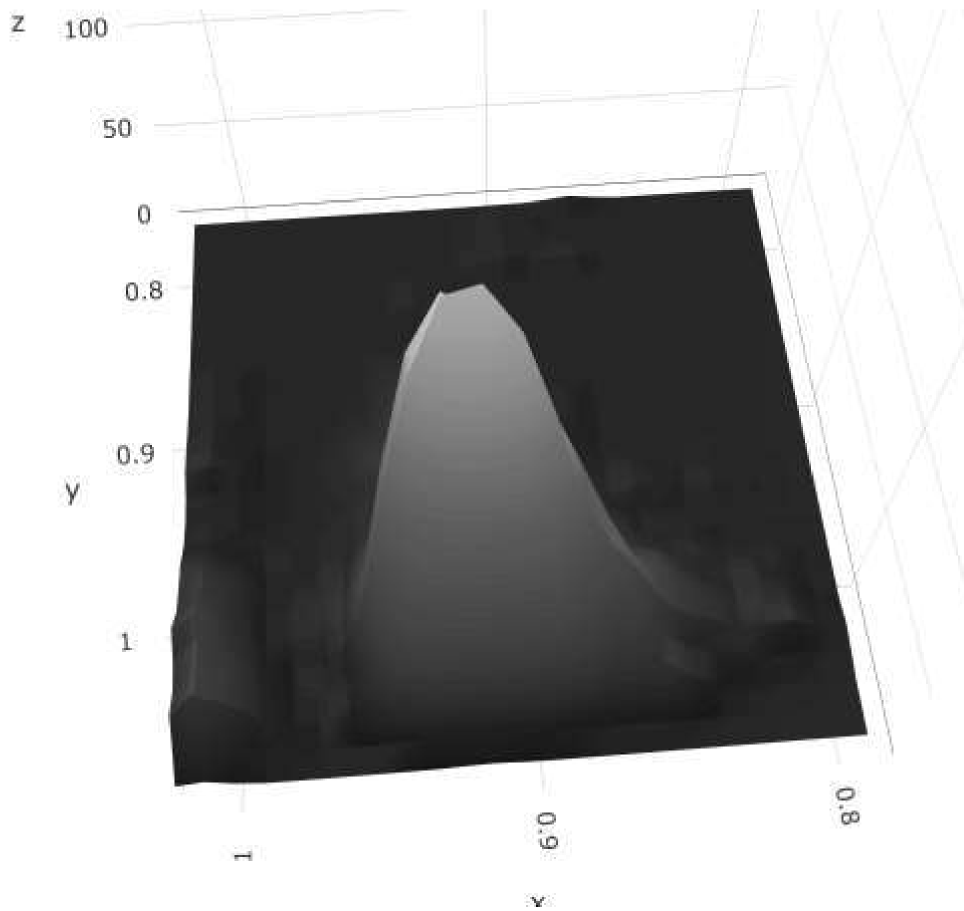
3.22

3.22 Generate $n = 50$ observations from a Gaussian AR(1) model with $\phi = .99$ and $\sigma_w = 1$. Using an estimation technique of your choice, compare the approximate asymptotic distribution of your estimate (the one you would use for inference) with the results of a bootstrap experiment (use $B = 200$).

Here, we are going to run a Gaussian ARMA(1,1) with parameters close to the causal boundary, φ = .99 and θ = .95. Then, we will compare the sampled asymptotic distribution of the parameters to the expected distribution and a bootstrapped one.

```
library(MASS)
library(plotly)
library(matlib)
library(mvtnorm)

set.seed(40)
# Initial simulation and estimation
x = arima.sim(n=200, list(ar=.99, ma=.95))
fit = arima(x, order=c(1,0,1)) # MLE
phi = fit$coef[1]
theta = fit$coef[2]

phi.mle = rep(NA, 100)
theta.mle = rep(NA, 1000)
# arima() tends to have convergence issues when the parameters are so
# close to the boundaries so we’ll use arma_coef from above instead
for (i in 1:1000) {
  sim = arima.sim(n=200, list(ar=.99, ma=.95))
  sim.fit = arma_coef(sim)
  phi.mle[i] = sim.fit$par[2]
  theta.mle[i] = sim.fit$par[3]
}

# Sample Distribution
dens = kde2d(phi.mle, theta.mle)
plot_ly(x = dens$x, y = dens$y, z = dens$z) %>% add_surface()
```

The multivariate density comes out fairly normal with the peak located at (x=0.92,y=1.02).

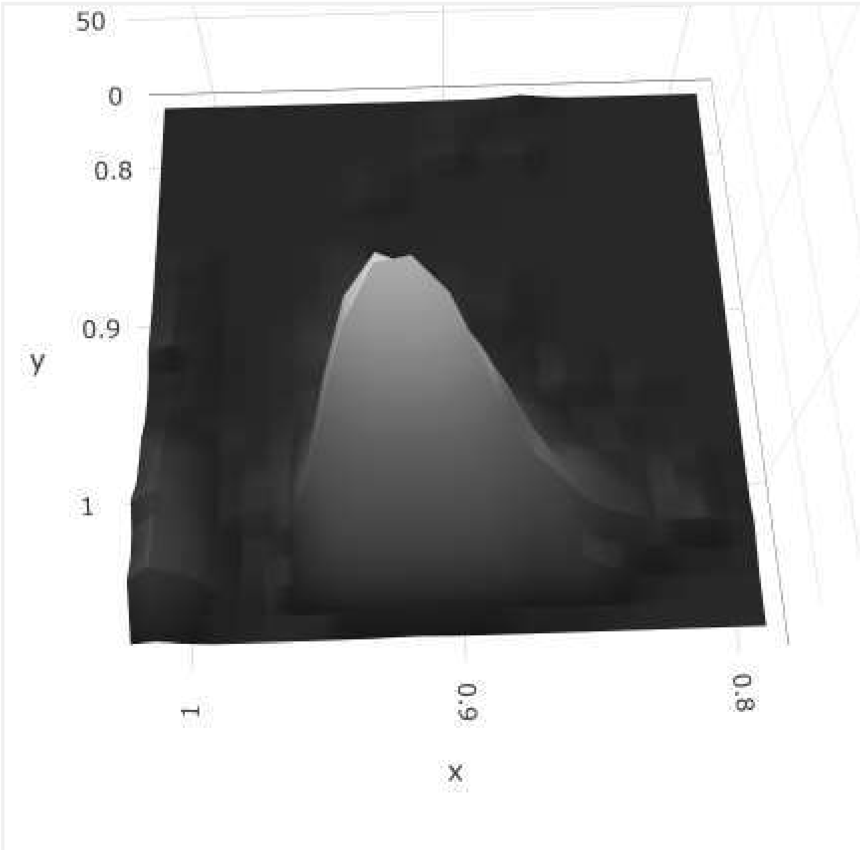


Continuing on, we create the density using the known parameters.

```
# Asymptotic Distribution Parameters
phi.asym = .9
theta.asym = .5
mean.vector = c(phi.asym, theta.asym)
cov.matrix = matrix(0, 2, 2)
cov.matrix[1] = 1 / (1 - phi.asym^2)
cov.matrix[2] = 1 / (1 + phi.asym*theta.asym)
cov.matrix[3] = cov.matrix[2]
cov.matrix[4] = 1 / (1 - theta.asym^2)
cov.matrix.inv = inv(cov.matrix) / 200

y.seq = seq(0.3, 0.7, by=0.001) # length 401
x.seq = seq(0.6, 1, by=0.001) # length 401
seq.matrix = matrix(0, 401, 2)
seq.matrix[,1] = x.seq
seq.matrix[,2] = y.seq

normal.data = rmvnorm(401, mean=mean.vector, sigma=cov.matrix.inv)
normal.dens = kde2d(normal.data[,1], normal.data[,2])
```

```
plot_ly(x=dens$x, y=dens$y, z=dens$z) %>% add_surface()
```

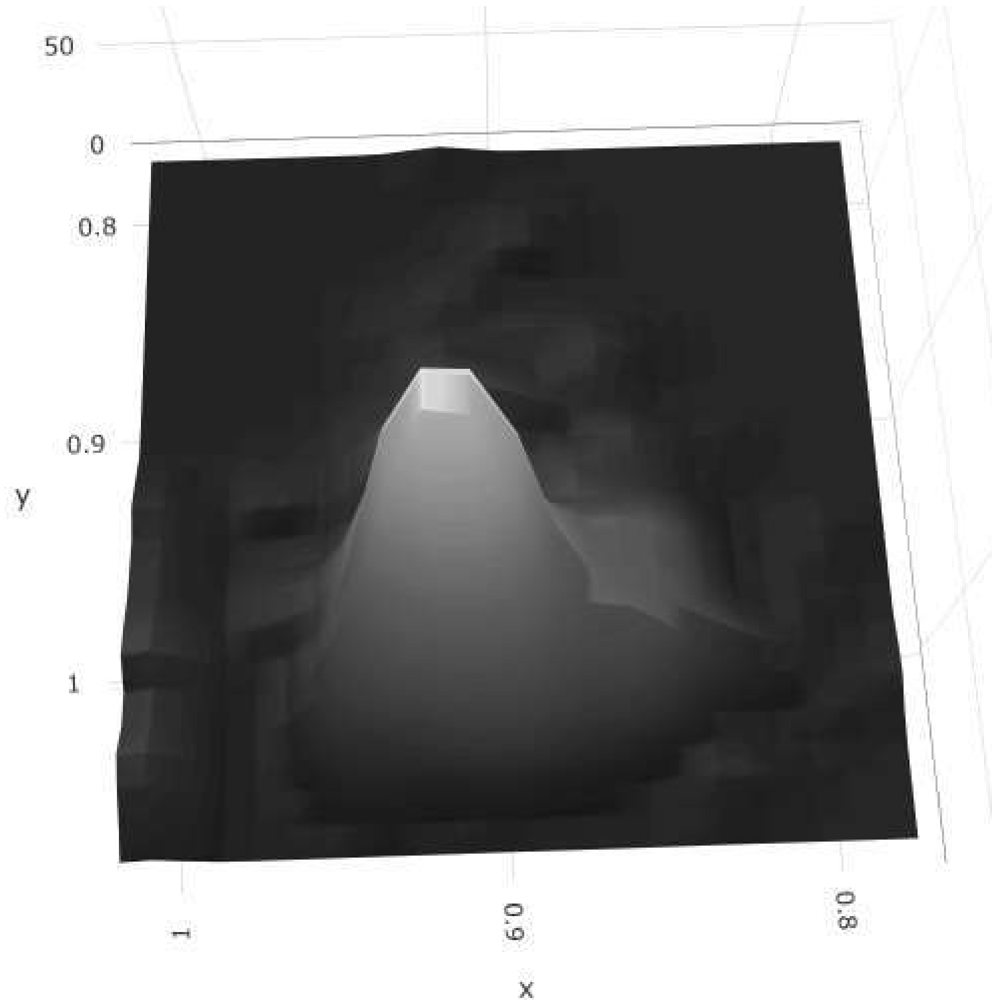
As seen below, the density generated with known parameters is essentially identical as the distribution from sampling. In fact, the peak is the same.




```
# Bootstrap
nboot = 500
resids = fit$residuals[-1]
x.star = x
phi.star.mle = rep(NA, nboot)
theta.star.mle = rep(NA, nboot)

for (i in 1:nboot) {
  resid.star = sample(resids, replace=TRUE)
  for (t in 1:199) {
    x.star[t+1] = phi*x.star[t] + theta*resid.star[t] + resid.star[t+1]
  }
  mle.fit = arma_coef(x.star)
  phi.star.mle[i] = mle.fit$par[2]
  theta.star.mle[i] = mle.fit$par[3]
```

```
}

dens.boot = kde2d(phi.star.mle, theta.star.mle)
plot_ly(x=dens.boot$x, y=dens.boot$y, z=dens.boot$z,) %>% add_surface()
```

As can be seen, running the bootstrap distribution yields fairly reasonable results, however, still less accurate than the sampling distribution, with a peak of (x=0.91,y=0.99).



3.23

3.23 Using Example 3.32 as your guide, find the Gauss–Newton procedure for estimating the autoregressive parameter, ϕ, from the AR(1) model, $x_t = \phi x_{t-1} + w_t$, given data $x_1, \dots, x_n$. Does this procedure produce the unconditional or the conditional estimator? *Hint:* Write the model as $w_t(\phi) = x_t - \phi x_{t-1}$; your solution should work out to be a non-recursive procedure.

$$\frac{\partial w_t(\phi)}{\partial \phi} = \frac{\partial x_t}{\partial \phi} - \frac{\partial \phi x_{t-1}}{\partial \phi} = \frac{\partial x_t}{\partial \phi} - x_{t-1} - \phi \frac{\partial x_{t-1}}{\partial \phi} = -x_{t-1}$$

Hence,

$$z_t(\phi) = -\frac{\partial w_t(\phi)}{\partial \phi} = x_{t-1}$$

And,

$$\phi_{(j+1)} = \phi_{(j)} + \frac{\sum_{t=1}^{n} z_t(\phi_j) w_t(\phi_{(j)})}{\sum_{t=1}^{n} z_t^2(\phi_{(j)})} = \phi_{(j)} + \frac{\sum_{t=1}^{n} x_{t-1}(x_t - \phi x_{t-1})}{\sum_{t=1}^{n} x_{t-1}}$$

$$= \frac{\sum_{t=1}^{n} x_{t-1} x_t}{\sum_{t=1}^{n} x_{t-1}^2}$$

This produces the conditional estimator.

3.24

3.24 Consider the stationary series generated by

$$x_t = \alpha + \phi x_{t-1} + w_t + \theta w_{t-1},$$

where $\mathrm{E}(x_t) = \mu$, $|\theta| < 1$, $|\phi| < 1$ and the w_t are iid random variables with zero mean and variance σ_w^2.

(a) Determine the mean as a function of α for the above model. Find the autocovariance and ACF of the process x_t, and show that the process is weakly stationary. Is the process strictly stationary?

We can write

$\mathbb{E}[x_t] = \alpha + \phi \mathbb{E}[x_{t-1}]$

$\mu = \alpha + \phi\mu, \;\; \alpha = \mu(1-\phi)$

Hence, we can define $y_t = x_t - \mu(1-\phi)$ and write

$yt = \phi y_{t-1} + w_t +_{t-1}$

In which case, the normal results apply that the series is causal given $|\phi| < 1$, and hence stationary, and the general solution of the ACF (3.51) applies. The same technique from Problem 1.17 can be used to show strict stationarity.

(b) Prove the limiting distribution as $n \to \infty$ of the sample mean,

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

is normal, and find its limiting mean and variance in terms of α, ϕ, θ, and σ_w^2. (Note: This part uses results from Appendix A.)

Because of causality, we can write

$$y_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$$

$$x_t = \mu + \sum_{j=0}^{\infty} \psi_j w_{t-j}, \;\; \psi_j = (\phi + \theta)\phi^{j-1}$$

Hence, we can apply Theorem A.5. Note, Theorem A.3 does not apply, because the x_t are not independent in this case.

Hence,

$$\bar{x}_n \sim AN(\mu, n^{-1}, V) = AN(\frac{\alpha}{1-\phi}, n^{-1}, V)$$

Where

$$V = \sigma_w^2 (\sum_{j=0}^{\infty} \psi_j)^2$$

3.25

3.25 A problem of interest in the analysis of geophysical time series involves a simple model for observed data containing a signal and a reflected version of the signal with unknown amplification factor a and unknown time delay δ. For example, the depth of an earthquake is proportional to the time delay δ for the P wave and its reflected form pP on a seismic record. Assume the signal, say s_t, is white and Gaussian with variance σ_s^2, and consider the generating model

$$x_t = s_t + as_{t-\delta}.$$

(a) Prove the process x_t is stationary. If $|a| < 1$, show that

$$s_t = \sum_{j=0}^{\infty} (-a)^j x_{t-\delta j}$$

is a mean square convergent representation for the signal s_t, for $t = 1, \pm 1, \pm 2, \ldots$.

We can easily see that $\mathbb{E}[x_t] = 0$, and

$\gamma(h) = \mathbb{E}[(s_t + as_{t-\delta})(s_{t+h} + ast + h - \delta)]$

Working this out, we can see

$\gamma(0) = \sigma_w^2(1 + a^2)$

$\gamma(\delta) = a\sigma_w^2$

And 0 otherwise.

$s_1 = x_1 - as_{1-\delta}$

$= x_1 - a(x_{1-\delta} - as_{1-2\delta})$

$= x_1 - ax_{1-\delta} + (-a)^2 s_{1-2\delta}$

$= x_1 - ax_{1-\delta} + (-a)^2 x_{1-2\delta} + (-a)^3 s_{1-3\delta}$

$= \sum_{j=0}^{2} (-a)^j x_{t-\delta j} + (-a)^3 s_{1-3\delta}$

Hence, we can write in general

$s_t = \sum_{j=0}^{k-1} (-a)^j x_{t-\delta j} + (-a)^k s_{t-\delta k}$

And

$$s_t - \sum_{j=0}^{k-1}(-a)^j x_{t-\delta j} = (-a)^k s_{t-\delta k}$$

$$\mathbb{E}[(s_t - \sum_{j=0}^{k-1}(-a)^j x_{t-\delta j})^2] = (-a)^k \mathbb{E}[((-a)^k s_{t-\delta k})^2]$$

Hence, as $k \to \infty$,

$$\mathbb{E}[(s_t - \sum_{j=0}^{k-1}(-a)^j x_{t-\delta j})^2] \to 0$$

.

(b) If the time delay δ is assumed to be known, suggest an approximate computational method for estimating the parameters a and σ_s^2 using maximum likelihood and the Gauss–Newton method.

If δ is known, then we have an $MA(\delta)$ model, where $\theta_j = 0$ for $j < \delta$

In which case, the results from Example 3.32 apply.

$$a_{(j+1)} = a_{(j)} + \frac{\sum_{t=1}^{n} z_t(a_{(j)}) s_t(a_{(j)})}{\sum_{t=1}^{n} z_t^2(a_{(j)})}$$

(c) If the time delay δ is an unknown integer, specify how we could estimate the parameters including δ. Generate a $n = 500$ point series with $a = .9$, $\sigma_w^2 = 1$ and $\delta = 5$. Estimate the integer time delay δ by searching over $\delta = 3, 4, \ldots, 7$.

We can use the ACF to detect the order of the MA model, then use Gauss-Newton estimation for the value of a.

```
n=500; a=.9; delta=5
params=rep(0,5); params[delta]=a
data=arima.sim(list(order=c(0,0,delta),ma=params), n)
acf(data, main="ACF")
fit = arima(data, order=c(0,0,delta))
```

3.26

3.26 *Forecasting with estimated parameters*: Let $x_1, x_2, \ldots, x_n$ be a sample of size n from a causal AR(1) process, $x_t = \phi x_{t-1} + w_t$. Let $\hat{\phi}$ be the Yule–Walker estimator of ϕ.

(a) Show $\hat{\phi} - \phi = O_p(n^{-1/2})$. See Appendix A for the definition of $O_p(\cdot)$.

(b) Let x_{n+1}^n be the one-step-ahead forecast of x_{n+1} given the data $x_1, \dots, x_n$, based on the known parameter, ϕ, and let $\hat{x}_{n+1}^n$ be the one-step-ahead forecast when the parameter is replaced by $\hat{\phi}$. Show $x_{n+1}^n - \hat{x}_{n+1}^n = O_p(n^{-1/2})$.

See solutions manual.

3.27

3.27 Suppose

$$y_t = \beta_0 + \beta_1 t + \cdots + \beta_q t^q + x_t, \quad \beta_q \neq 0,$$

where x_t is stationary. First, show that $\nabla^k x_t$ is stationary for any $k = 1, 2, \dots$, and then show that $\nabla^k y_t$ is not stationary for $k < q$, but is stationary for $k \geq q$.

We show by induction that $\nabla^k x_t$ is stationary.

Base case: $k = 1$

$\nabla x_t = x_t - x_{t-1} = \mu_x - \mu_x = 0$

$\gamma_1(h) = \mathbb{E}[(x - x_{t-1})(x_{t+h} - x_{t+h-1)}]$

$$= \mathbb{E}[x_t x_{t+h}] - \mu_x^2 - \mathbb{E}[x_t x_{t+h-1}] + \mu_x^2 - \mathbb{E}[x_{t-1} x_{t+h}] + \mu_x^2 + \mathbb{E}[x_{t-1} x_{t+h-1}] - \mu_x^2$$

$$= 2\gamma_1(h) - \gamma(h-1) - \gamma(h+1)$$

Hence, ∇x_t is stationary.

Induction assumption: $\nabla^k x_t$ is stationary.

$\nabla^{k+1} x_t = \nabla(\nabla^k x_t) = \nabla^k x_t - \nabla^k x_{t-1}$

$\mathbb{E}[\nabla^k x_t] = \mathbb{E}[\nabla^k x_t - \nabla^k x_{t-1}] = \mu_k - \mu_k = 0$

$\gamma_{k+1}(h) = \mathbb{E}[\nabla^k x_t \nabla^k x_{t-1}]$

$$= \mathbb{E}[(\nabla^k x_t - \nabla^k x_{t-1})(\nabla^k x_{t+h} - \nabla^k x_{t+h-1})]$$

$$= 2\gamma_k(h) - \gamma_k(h-1) - \gamma_k(h+1)$$

See solutions manual for remainder of question.

3.28

3.28 Verify that the IMA(1,1) model given in (3.148) can be inverted and written as (3.149).

We can write

$w_t = \lambda w_{t-1} + y_t$

Given $|\lambda| < 1$, we can write

$$w_t = \sum_{j=0}^{\infty} \lambda^j y_{t-j}$$

$$= \sum_{j=0}^{\infty} \lambda^j (x_{t-j} - x_{t-1-j})$$

$$= x_t - x_{t-1} + \lambda x_{t-1} - \lambda x_{t-2} + \lambda^2 x_{t-2} - \lambda^2 x_{t-3} \ldots$$

$$w_t - x_t = \lambda x_{t-1} - x_{t-1} + \lambda^2 x_{t-2} - \lambda x_{t-2} + \ldots$$

$$= \sum_{j=1}^{\infty} (\lambda^j - \lambda^{j-1}) x_{t-j}$$

$$= \sum_{j=1}^{\infty} \lambda^{j-1} (\lambda - 1) x_{t-j}$$

$$x_t = \sum_{j=1}^{\infty} (1 - \lambda) \lambda^{j-1} x_{t-j}$$

3.29

3.29 For the ARIMA(1, 1, 0) model with drift, $(1 - \phi B)(1 - B)x_t = \delta + w_t$, let $y_t = (1 - B)x_t = \nabla x_t$.

(a) Noting that y_t is AR(1), show that, for $j \geq 1$,

$$y_{n+j}^n = \delta\,[1 + \phi + \cdots + \phi^{j-1}] + \phi^j\, y_n.$$

We can only Property 3.7, if the series has a zero mean.

Given that the series y_t is given by:

$y_t = \phi y_{t-1} + w_t + \delta$
Where
$\delta = (1-\phi)\mu$
We can see that if δ is non-zero, then the mean is non-zero and hence we need to define $z_t = y_t - \mu$ to apply Property 3.7.
So, we write

$$z_{n+1}^n = \phi z_n = \phi(y_n - \mu)$$

$$y_{n+1}^n - \mu = \phi y_n - \phi\mu$$

$$y_{n+1}^n = \phi y_n + \mu(1-\phi)$$

$$= \phi y_n + \delta$$

And then

$$z_{n+2}^n = \phi z_{n+1}^n$$

$$y_{n+2}^2 - \mu = \phi(y_{n+1}^n - \mu)$$

$$y_{n+2}^2 = \phi(\phi y_n + \delta - \mu) + \mu$$

$$= \phi^2 y_n + \phi\delta - \phi\mu + \mu$$

$$= \phi^2 y_n + \phi\delta + \mu(1-\phi)$$

$$= \phi^2 y_n + \phi\delta + \delta$$

$$= \phi^2 y_n + \delta(1+\phi)$$

Hence, in general
$y_{n+j}^n = \phi^j y_n + \delta(1 + \phi + ... + \phi^{j-1})$

(b) Use part (a) to show that, for $m = 1, 2, \ldots,$

$$x_{n+m}^n = x_n + \frac{\delta}{1-\phi}\left[m - \frac{\phi(1-\phi^m)}{(1-\phi)}\right] + (x_n - x_{n-1})\frac{\phi(1-\phi^m)}{(1-\phi)}.$$

Hint: From (a), $x_{n+j}^n - x_{n+j-1}^n = \delta\frac{1-\phi^j}{1-\phi} + \phi^j(x_n - x_{n-1})$. Now sum both sides over j from 1 to m.

Given $y_t = x_t - x_{t-1}$, we can immediately write

$$x_{n+j}^n - x_{n+j-1}^n = \phi^j(x_n - x_{n-1}) + \delta\frac{1-\phi^j}{1-\phi}$$

As per the hint:

$$\sum_{j=1}^{m} x_{n+j}^n - x_{n+j-1}^n = \sum_{j=1}^{m} \phi^j(x_n - x_{n-1}) + \delta\frac{1-\phi^j}{1-\phi}$$

Note,

$$\sum_{j=1}^{m} x_{n+j}^n - x_{n+j-1}^n = x_{n+m}^n - x_n$$

Hence,

$$x_{n+m}^n - x_n = \sum_{j=1}^{m} \phi^j(x_n - x_{n-1}) + \sum_{j=1}^{m} \delta\frac{1-\phi^j}{1-\phi}$$

$$= (x_n - x_{n-1})\frac{1-\phi^m}{1-\phi} + \sum_{j=1}^{m} \delta\frac{1-\phi^j}{1-\phi}$$

Now,

$$\sum_{j=1}^{m} \delta\frac{1-\phi^j}{1-\phi} = \sum_{j=1}^{m} \frac{\delta}{1-\phi} - \frac{\phi^j}{1-\phi}$$

$$= \frac{\delta m}{1-\phi} - \frac{\delta}{1-\phi}\sum_{j=1}^{m} \phi^j$$

$$= \frac{\delta m}{1-\phi} - \frac{\delta\phi}{1-\phi}\sum_{j=1}^{m} \phi^{j-1}$$

$$= \frac{\delta m}{1-\phi} - \frac{\delta\phi}{1-\phi}\frac{1-\phi^m}{1-\phi}$$

$$= \frac{\delta}{1-\phi}(m - \frac{\phi(1-\phi^m)}{1-\phi})$$

Hence,

$$x_{n+m} = x_n + (x_n - x_{n-1})\frac{1-\phi^m}{1-\phi} + \frac{\delta}{1-\phi}(m - \frac{\phi(1-\phi^m)}{1-\phi})$$

(c) Use (3.145) to find P_{n+m}^n by first showing that $\psi_0^* = 1$, $\psi_1^* = (1+\phi)$, and $\psi_j^* - (1+\phi)\psi_{j-1}^* + \phi\psi_{j-2}^* = 0$ for $j \geq 2$, in which case $\psi_j^* = \frac{1-\phi^{j+1}}{1-\phi}$, for $j \geq 1$. Note that, as in Example 3.37, equation (3.145) is exact here.

From (3.145), we can write

$$\psi * (z) = \frac{\theta(z)}{\phi(z)(1-z)^d} = \frac{1}{(1-\phi z)(1-z)} = \frac{1}{1-z-\phi z+\phi z^2)}$$

Hence,

$$(1-z-\phi z+\phi z^2)(\psi_0 * + \psi_1 * z + \psi_2 * z^2 + ...) = 1$$

From this we can deduce

$$\psi_0 * = 1, \;\; \psi_1 * = 1+\phi$$
$$\psi_j * - (1+\phi)\psi_{j-1} * + \phi\psi_{j-2} * = 0$$

Hence,

$$\psi_j * = (1+\phi)\psi_{j-1} * - \theta\psi_{j-2} *$$

$$\psi_1 * = 1+\phi = \sum_{j=1}^{2} \phi^{j-1} = \frac{1-\phi^2}{1-\phi}$$

$$\psi_2 * = -\phi + (1+\phi)(1+\phi) = 1+\phi+\phi^2 = \sum_{j=1}^{3} \phi^{j-1} = \frac{1-\phi^3}{1-\phi}$$

Hence, in general

$$\psi_j * = \frac{1-\phi^{j+1}}{1-\phi}$$

Hence, we can write

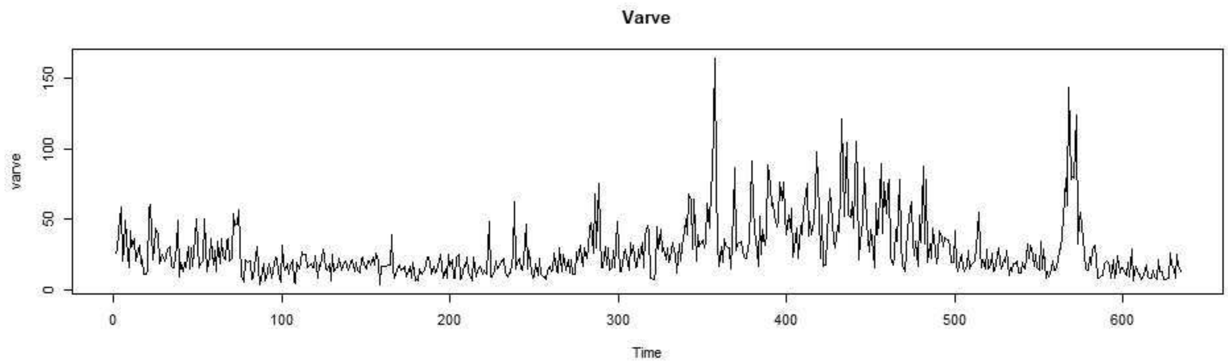
$$P_{n+m}^n = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2 * = \sigma_w^2 + \sum_{j=0}^{m-1} \psi_j^2 *$$

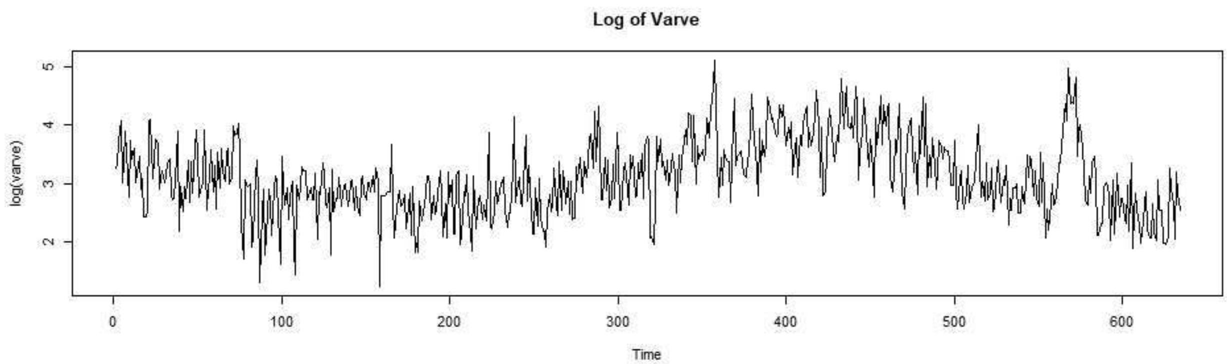
$$= \sigma_w^2 + \frac{1}{(1-\phi)^2} \sum_{j=1}^{m-1} (1-\phi^{j+1})^2$$

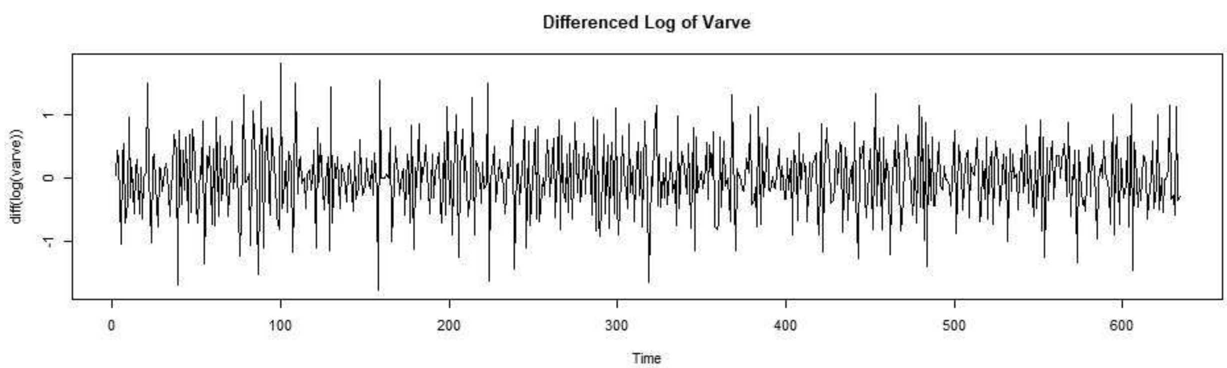
3.30

3.30 For the logarithm of the glacial varve data, say, x_t, presented in Example 3.33, use the first 100 observations and calculate the EWMA, $\tilde{x}_{t+1}^t$, given in (3.151) for $t = 1, \ldots, 100$, using $\lambda = .25, .50$, and $.75$, and plot the EWMAs and the data superimposed on each other. Comment on the results.

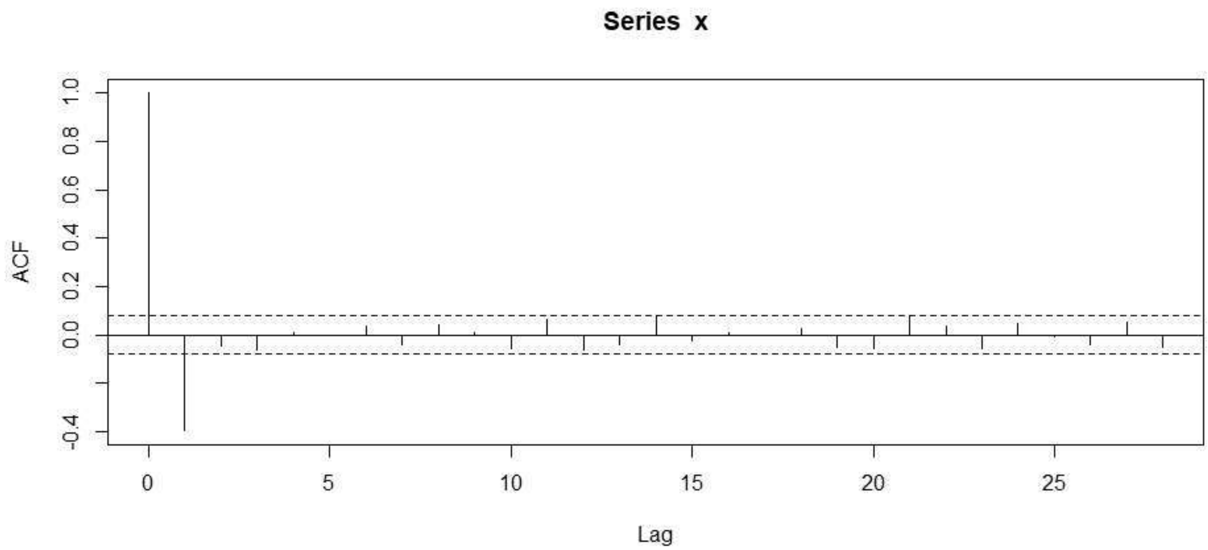
Inspecting the varve series, we can see that the variance is not constant across the data. For this reason, we take the log and observe that the variance is reduced but different means across the series are still apparent. Taking the first difference of the logged data produces a reasonably stationary time series.

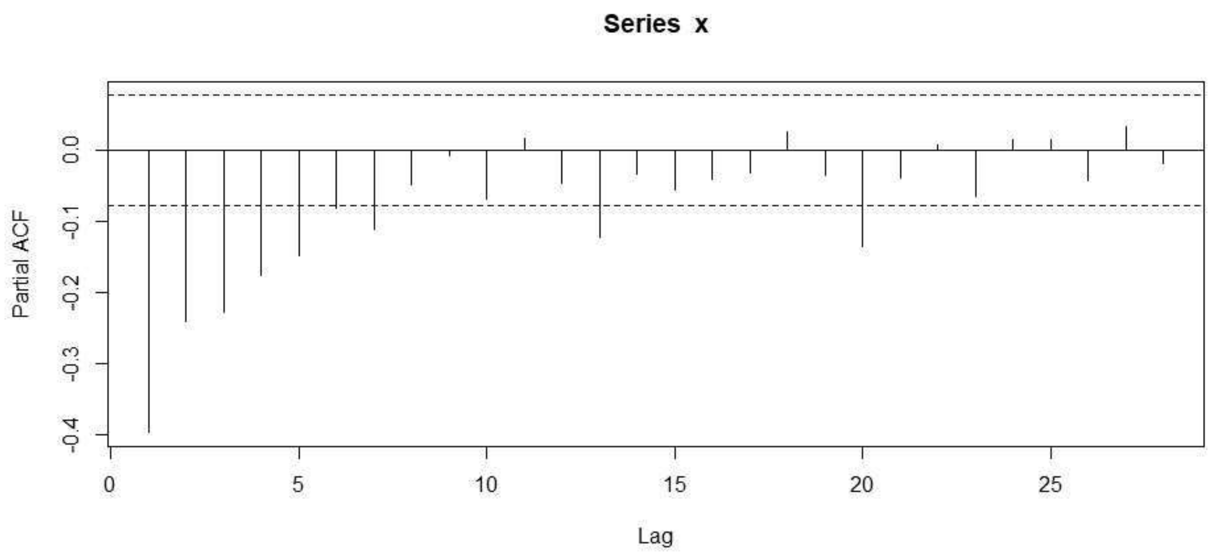






Looking at the ACF and the PACF for the differenced logged data, we can see that the lags on the PACF drop off slowly, while the lags on the ACF cut off quite sharply after lag=1. This tells us that the series is an IMA(1,1).





We can forecast IMA(1,1) using the truncated forecast given below:

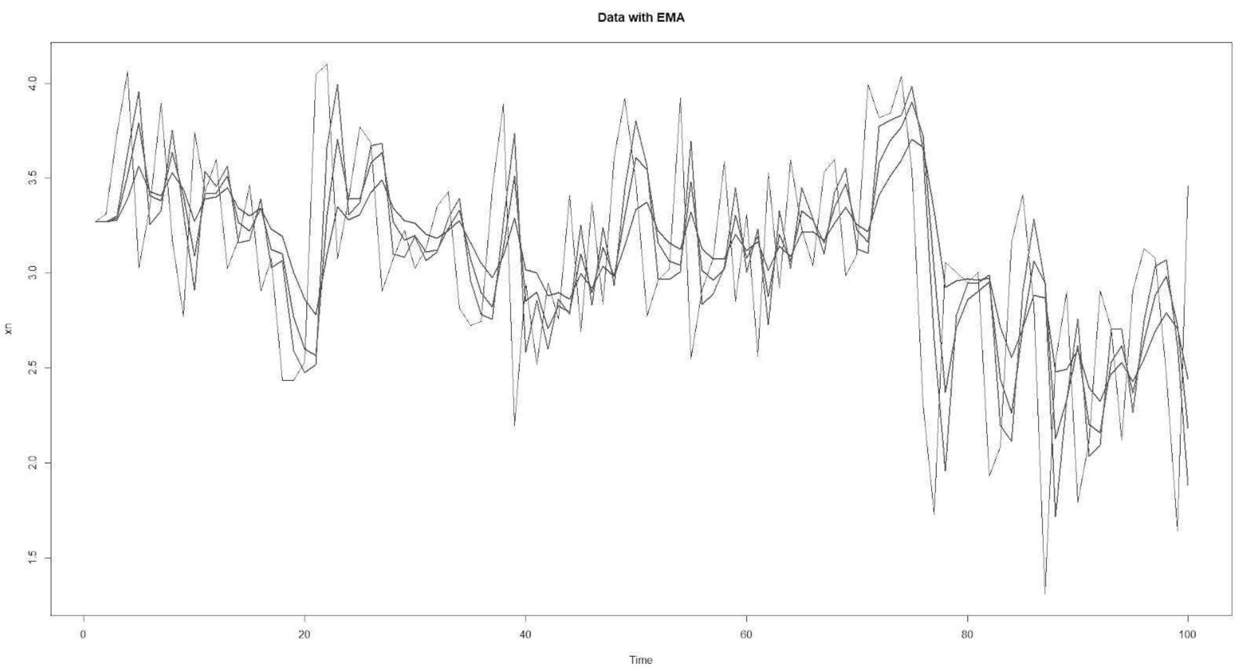
$$\tilde{x}_{n+1}^{n} = (1-\lambda)x_n + \lambda \tilde{x}_n^{n-1}, \quad n \geq 1,$$

We will do so for λ = 0.25, 0.5 and 0.75, using the first 100 observations.

```
xn = log(varve[1:100])
fore.x = matrix(0, 100, 3)
lambdas = c(.25, .5, .75)
xn[1] -> fore.x[1,1] -> fore.x[1,2] -> fore.x[1,3]
for (i in 1:3) {
  lambda = lambdas[i]
```

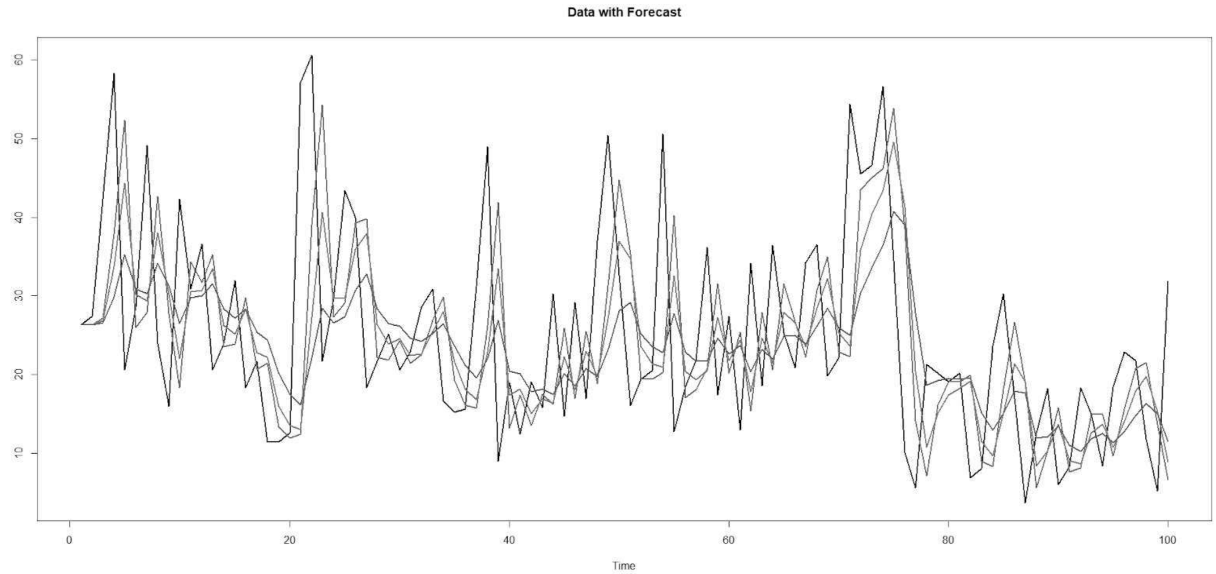
```
  for (j in 2:100) {
    fore.x[j,i] = (1-lambda)*xn[j-1] + lambda*fore.x[j-1,i]
  }
}
plot.ts(xn, main="Data with EMA")
lines(fore.x[,1], col=2, lwd=2)
lines(fore.x[,2], col=4, lwd=2)
lines(fore.x[,3], col=6, lwd=2)
```

Below we can see that for greater values of lambda, the smoother the forecast is, and that all forecasts are within the extremes of the data.



We can make this a smoothing of the original data by taking the exponent of each forecast.

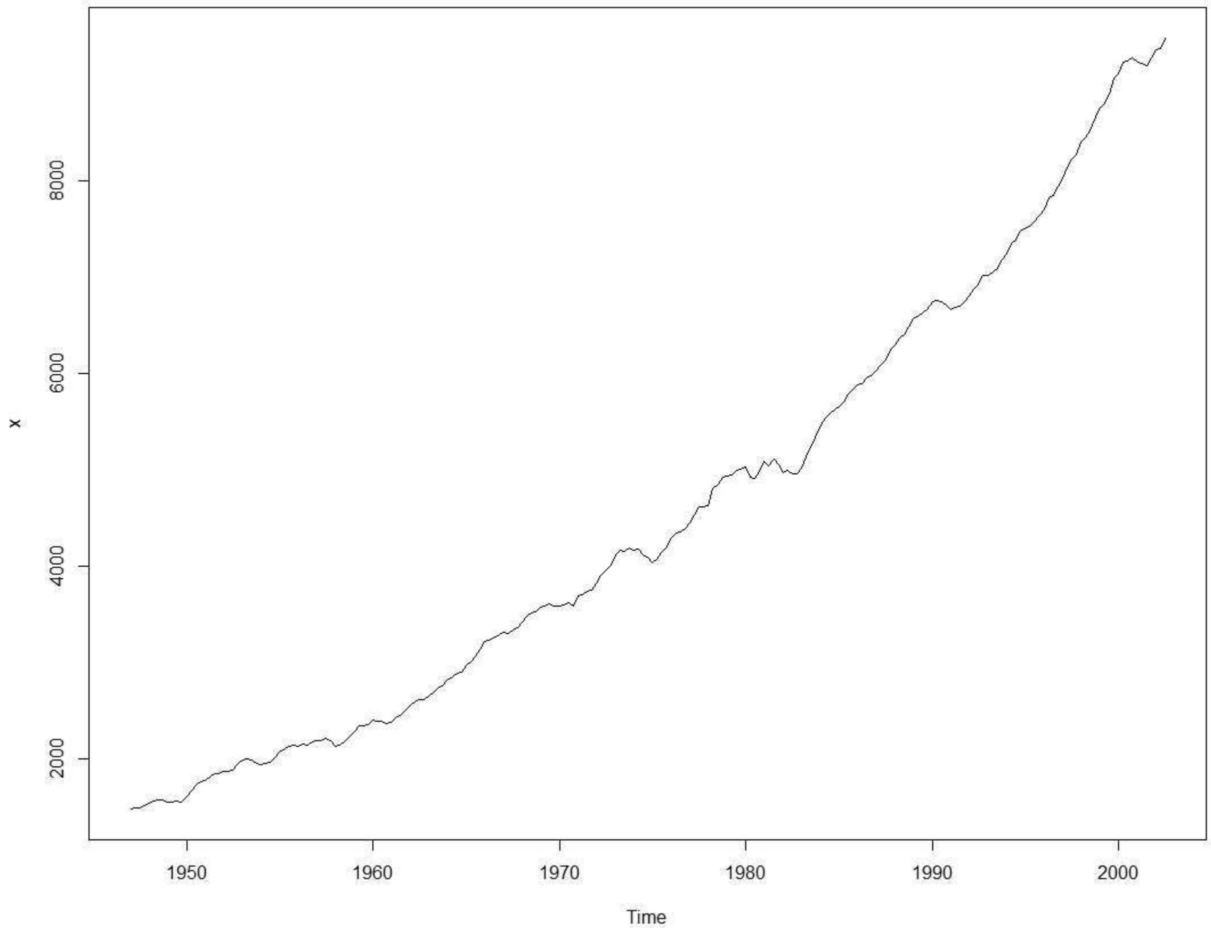
```
plot.ts(varve, main="Data with Forecast", xlim=c(0,100))
fish = ts.intersect(Varve=varve[1:100], fore1=ts(exp(fore.x[,1])),
fore2=ts(exp(fore.x[,2])), fore3=ts(exp(fore.x[,3])), dframe=TRUE)
ts.plot(fish, col=1:4, main="Data with Forecast", lwd=2)
```



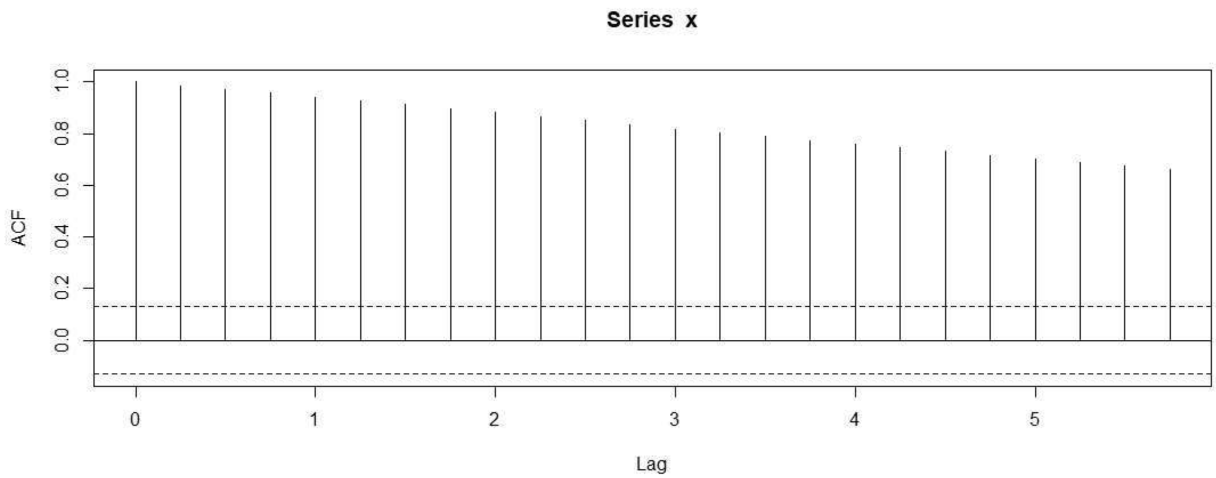
3.31

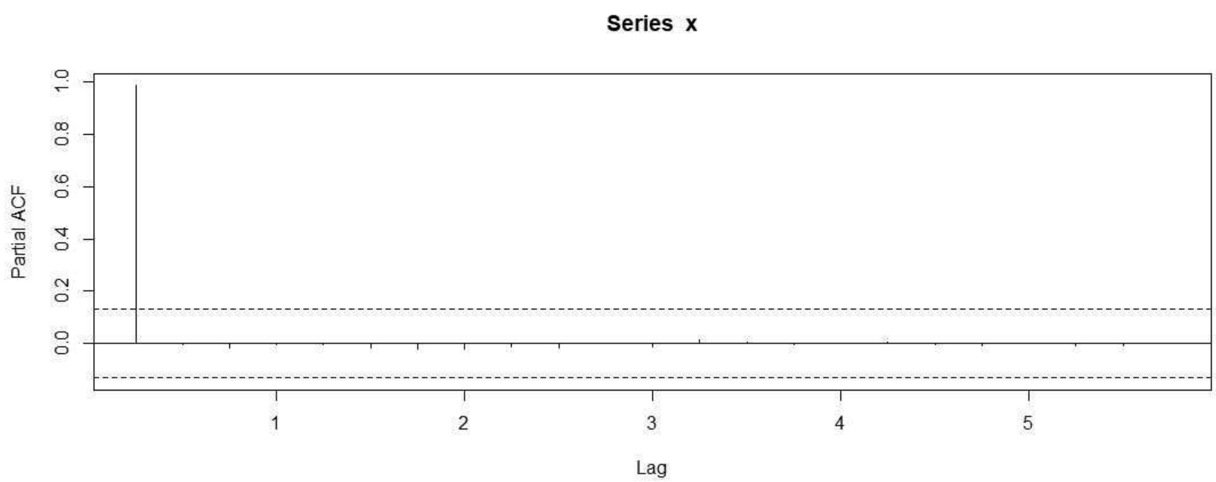
3.31 In Example 3.40, we presented the diagnostics for the MA(2) fit to the GNP growth rate series. Using that example as a guide, complete the diagnostics for the AR(1) fit.

Plotting the GNP data we can see that there is a very strong trend and it is hard to discern any other characteristics, besides some cycles throughout the period.

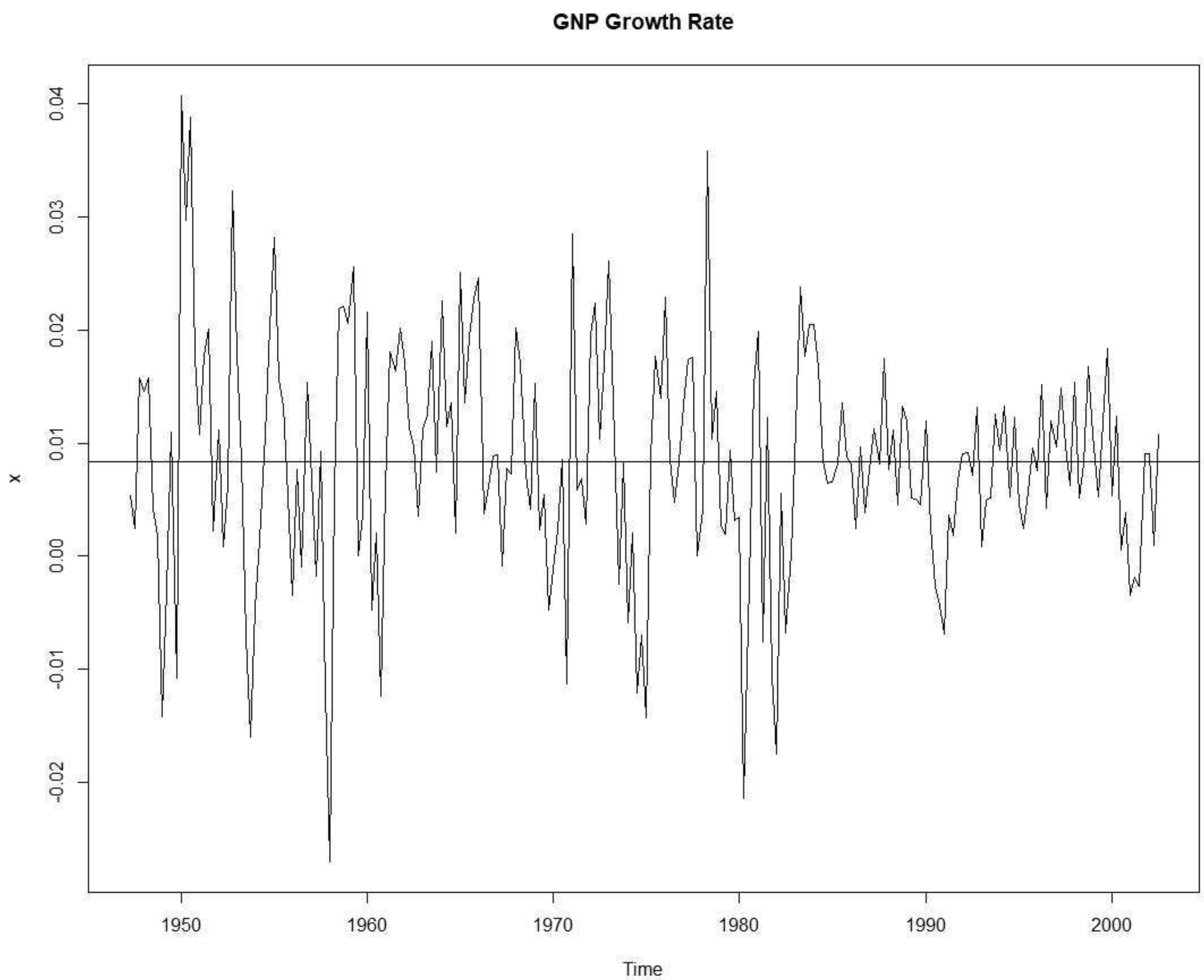


From looking at the ACF and PACF, we see no distinct relationships and obviously conclude that the data is non-stationary.

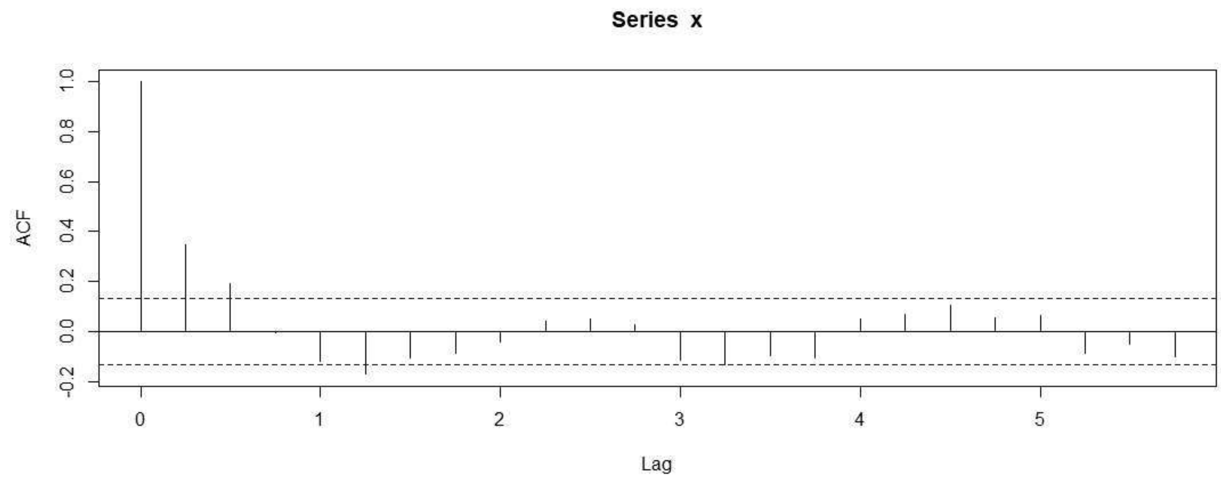


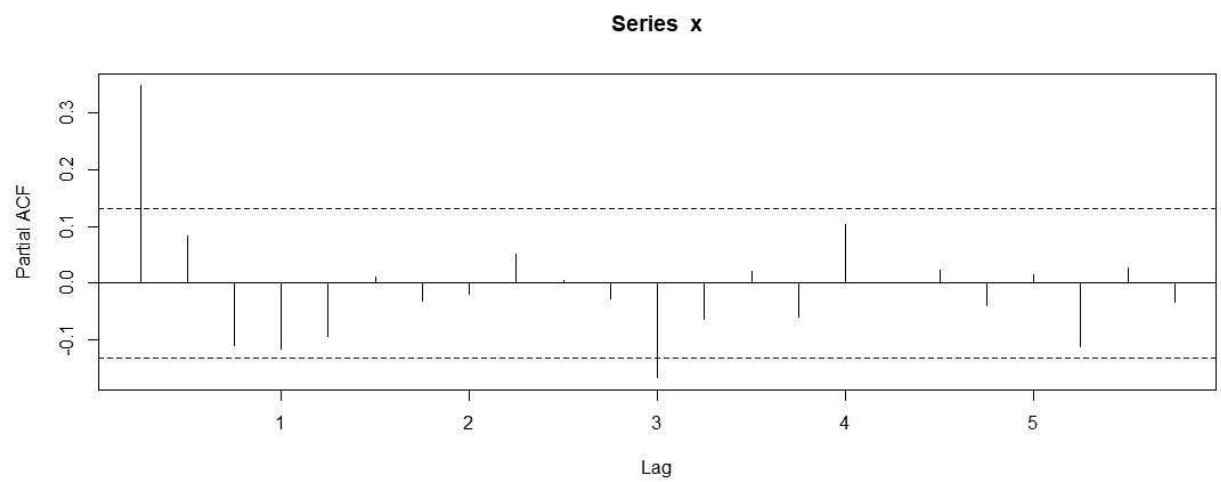


However, looking at the differenced logged data (which is approximately equal to its growth rate), we can see that the series is fairly stationary and stable over time.



Plotting the ACF and PACF we can see that it's not entirely clear whether the plots are tailing off or cutting off. We may argue that the ACF cuts off at lag 2 and the PACF is tailing off, suggesting an ARMA(0,1,2), or MA(2) for the growth rate. Or we may argue that the ACF is tailing off and the PACF is cutting off at lag 1, suggesting an ARMA(1,1,0), or AR(1) for the growth rate.

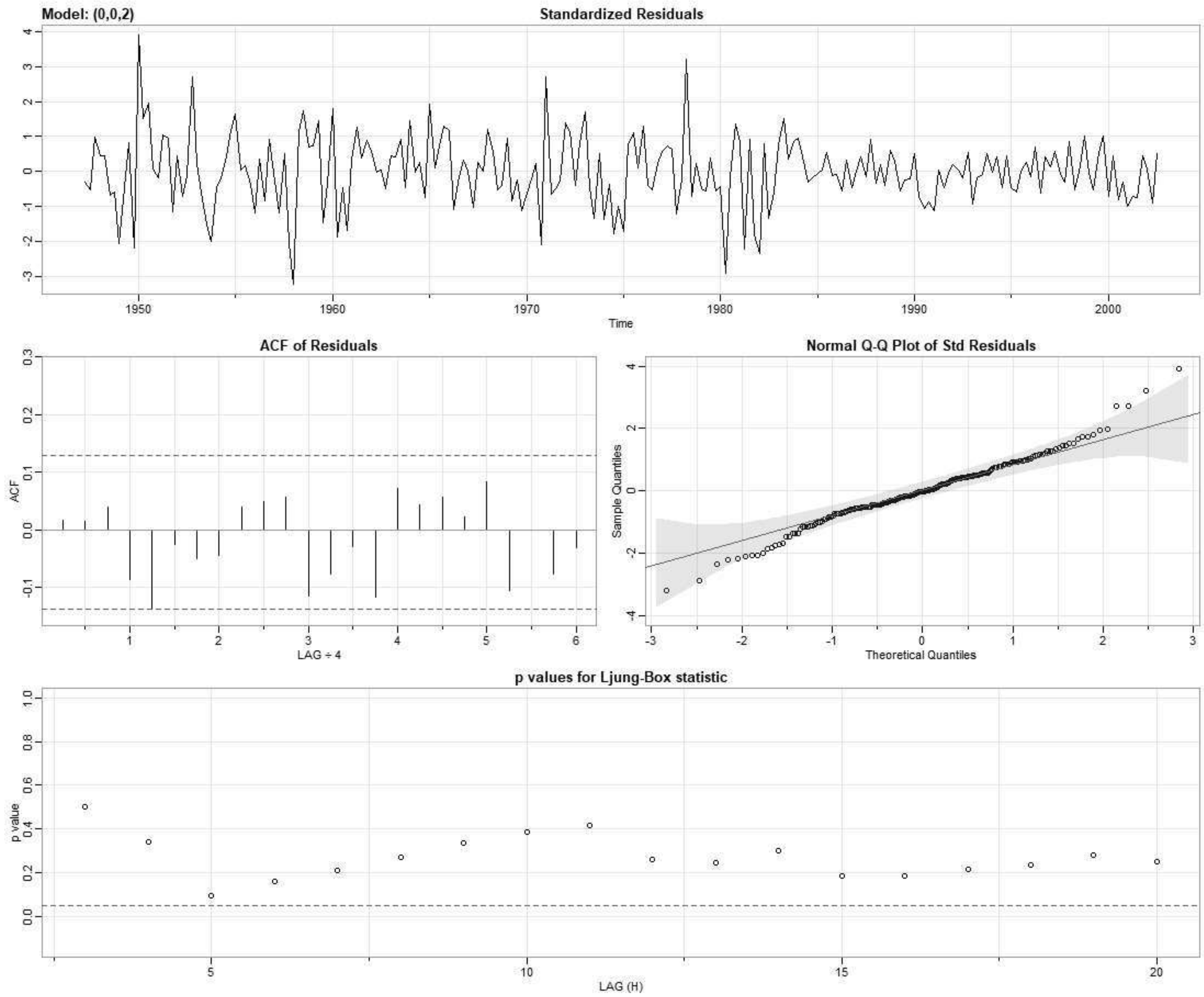




We will do our analysis for both.
Fitting an MA(1) to the growth rate, we find that all coefficients are significant and the residuals are fairly normal. We do see a few outliers in the residuals, with several data exceeding 3 standard deviations.

```
ma.fit = sarima(x,0,0,2)
> ma.fit$ttable
      Estimate     SE t.value p.value
ma1     0.3028 0.0654  4.6272  0.0000
ma2     0.2035 0.0644  3.1594  0.0018
xmean   0.0083 0.0010  8.7178  0.0000
ma.fit$AIC
[1] -6.450133
ma.fit$AICc
[1] -6.449637
$BIC
```

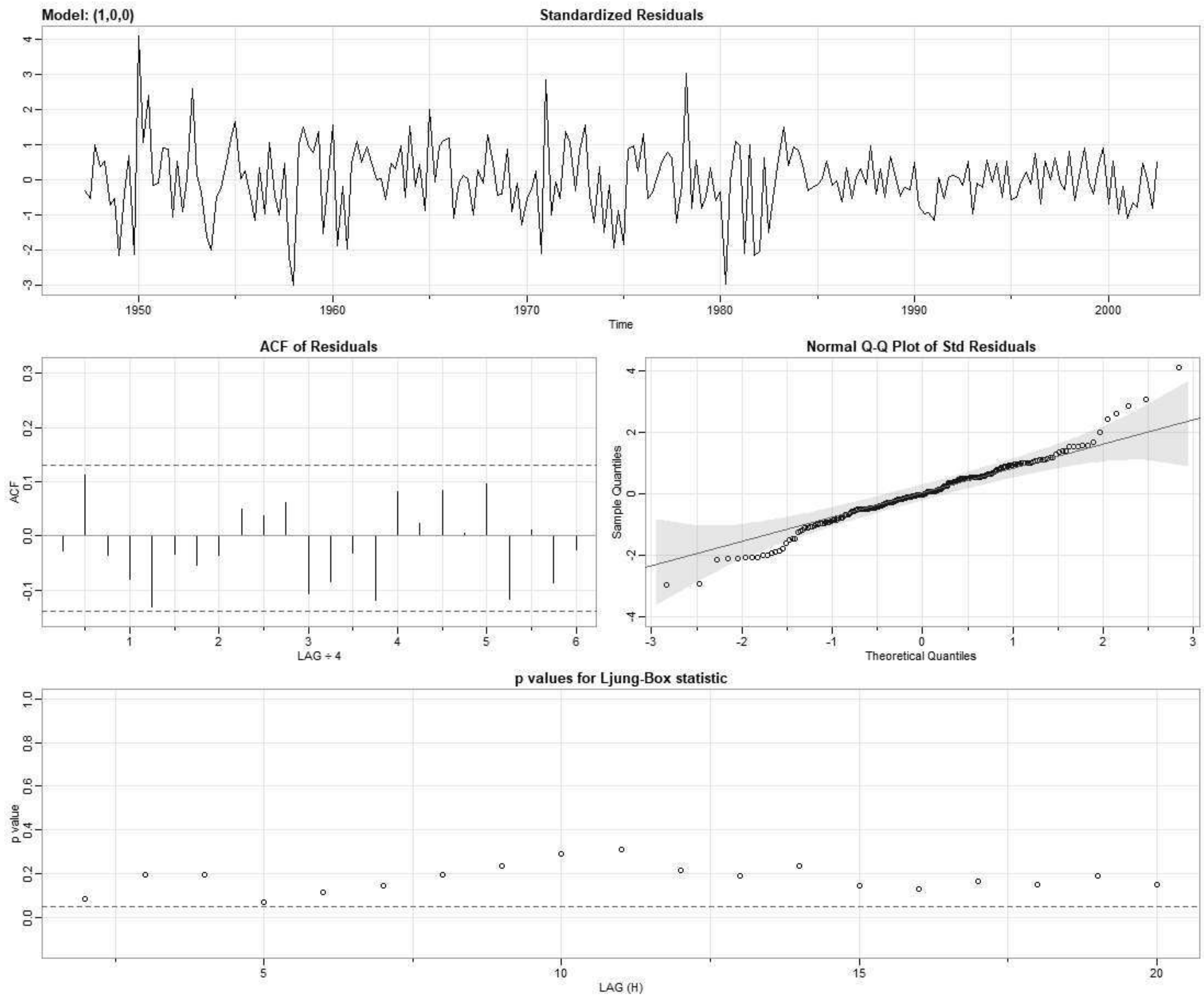
```
[1] -6.388823
```



Fitting an AR(1) to the growth rate, we find the coefficient is significant with a similar mean to the MA(1). The residuals are also fairly normal, with a few outliers similar to the MA(1) fit. The Ljung-Box statistics suggest that the data are independent and identically distributed. The AIC and AICc prefer the AR(1) model, however, the BIC prefers the MA(1).

```
ar.fit = sarima(x,1,0,0)
> ar.fit$ttable
      Estimate     SE  t.value p.value
ar1     0.3467 0.0627  5.5255       0
xmean   0.0083 0.0010  8.5398       0
ar.fit$AIC
[1] -6.44694
ar.fit$AICc
[1] -6.446693
```

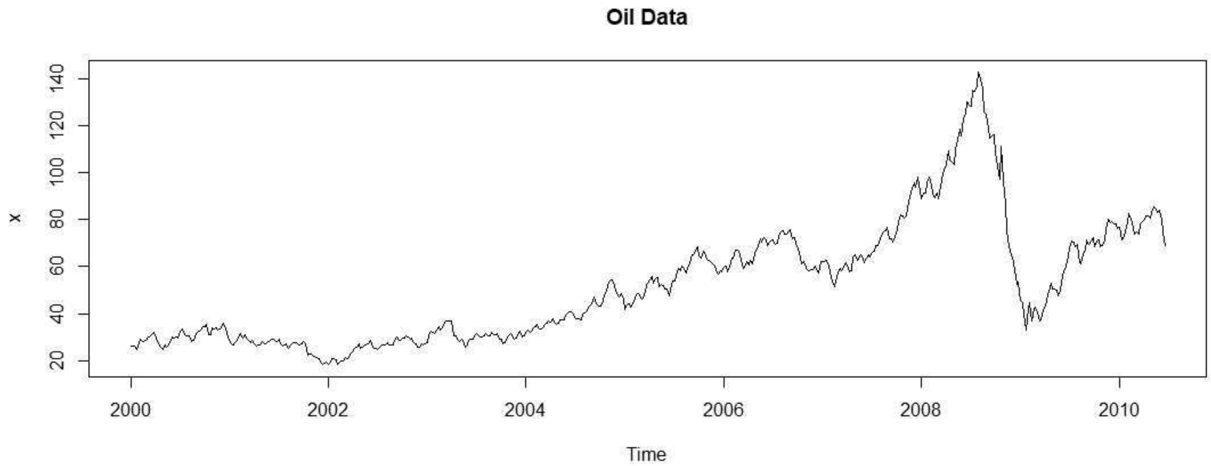
```
ar.fit$BIC
[1] -6.400958
```

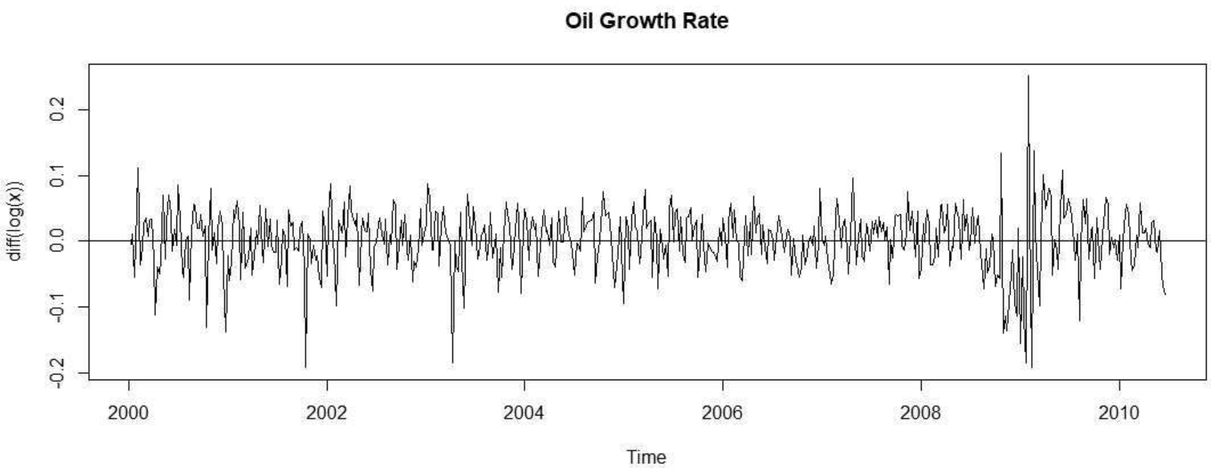


3.32

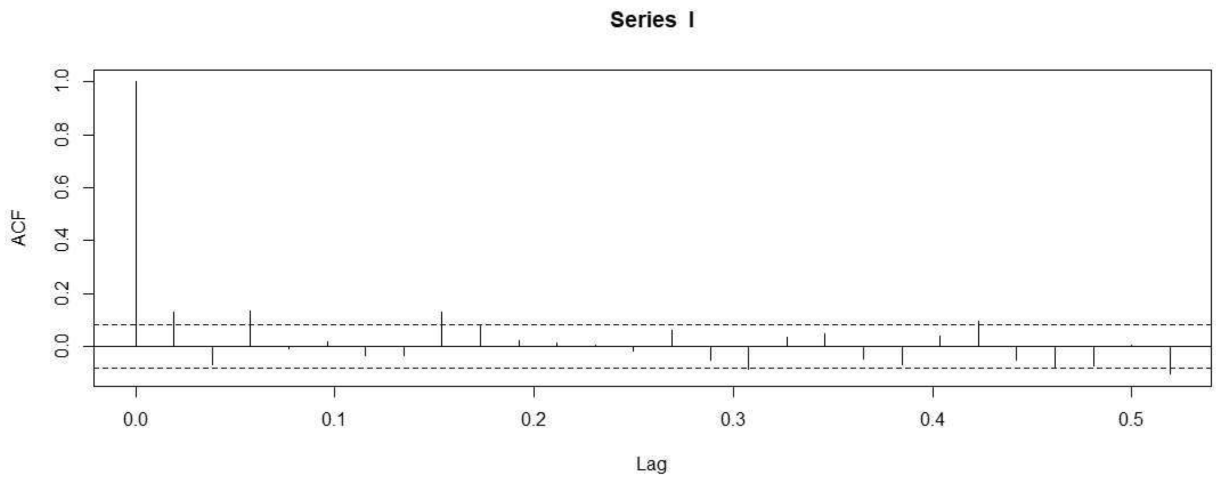
3.32 Crude oil prices in dollars per barrel are in `oil`. Fit an ARIMA(p, d, q) model to the growth rate performing all necessary diagnostics. Comment.

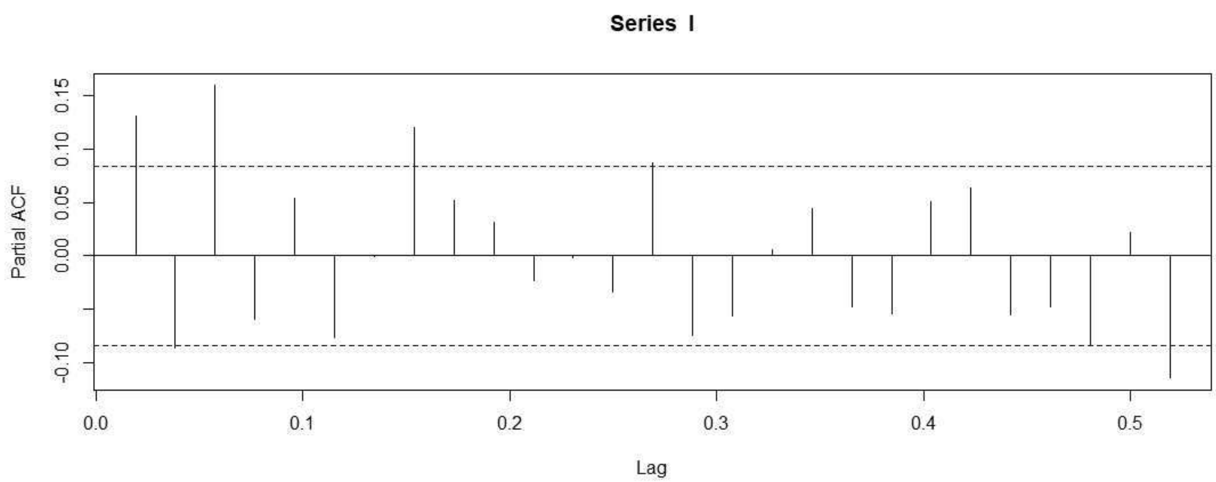
Plotting the data we can see that there is a clear trend through most of the data, though the exact mean is different across the series. The variance also appears to be non-constant. Taking the differenced log of the data gives a fairly stationary series. The average growth rate is close to 0, but not exact.



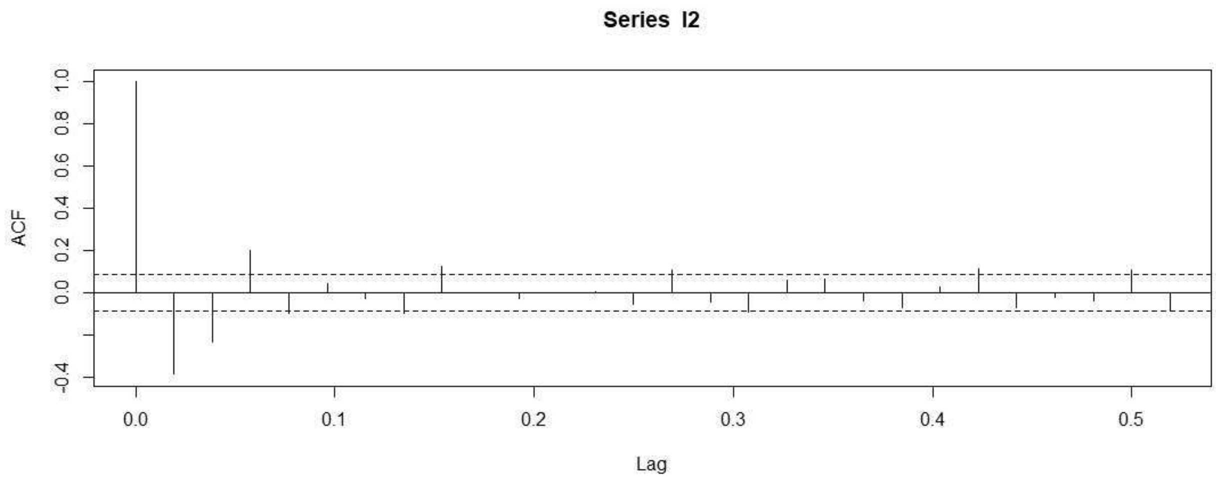


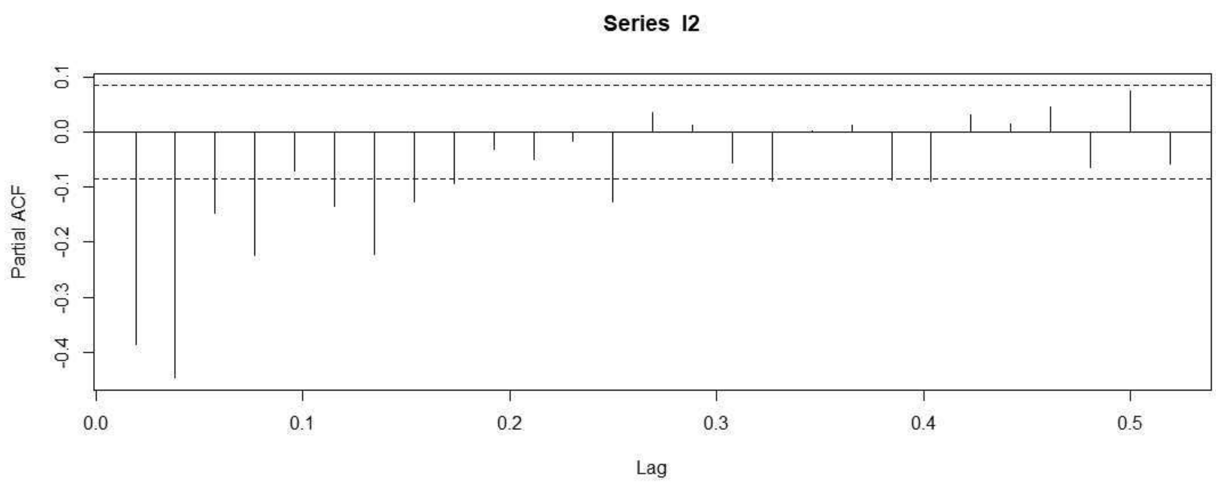
Looking at the ACF and PACF, we see that ACF doesn't seem to have any lags. We could argue that there are significant lags at 1, 2 and 3, but it seems doubtful. The PACF seems to cut off at lag 3 with some possible significance at lag 8. We will try an AR(3) on the growth rate, or ARMA(3,1,0) on the data.





Looking at the differenced growth rate, we find a stationary series (plot not shown), with outliers around the 2008 mark. The ACF for the second difference series could suggest cutting off at lag 3 and the PACF tailing off. Though we could also argue the PACF cutting off at lag 2 and the ACF tailing off. We will try an ARMA(0,2,3) and an ARMA(2,2,0) on the data.





First, we'll consider the various AR models we could run on the growth rate data. We can see that the AR(3) does pan the best. Looking at the diagnostics (not shown), we can see that the residuals are reasonably normal, though there are a number of outliers. Additionally, the Q statistics show that there is some correlation, as does the ACF.

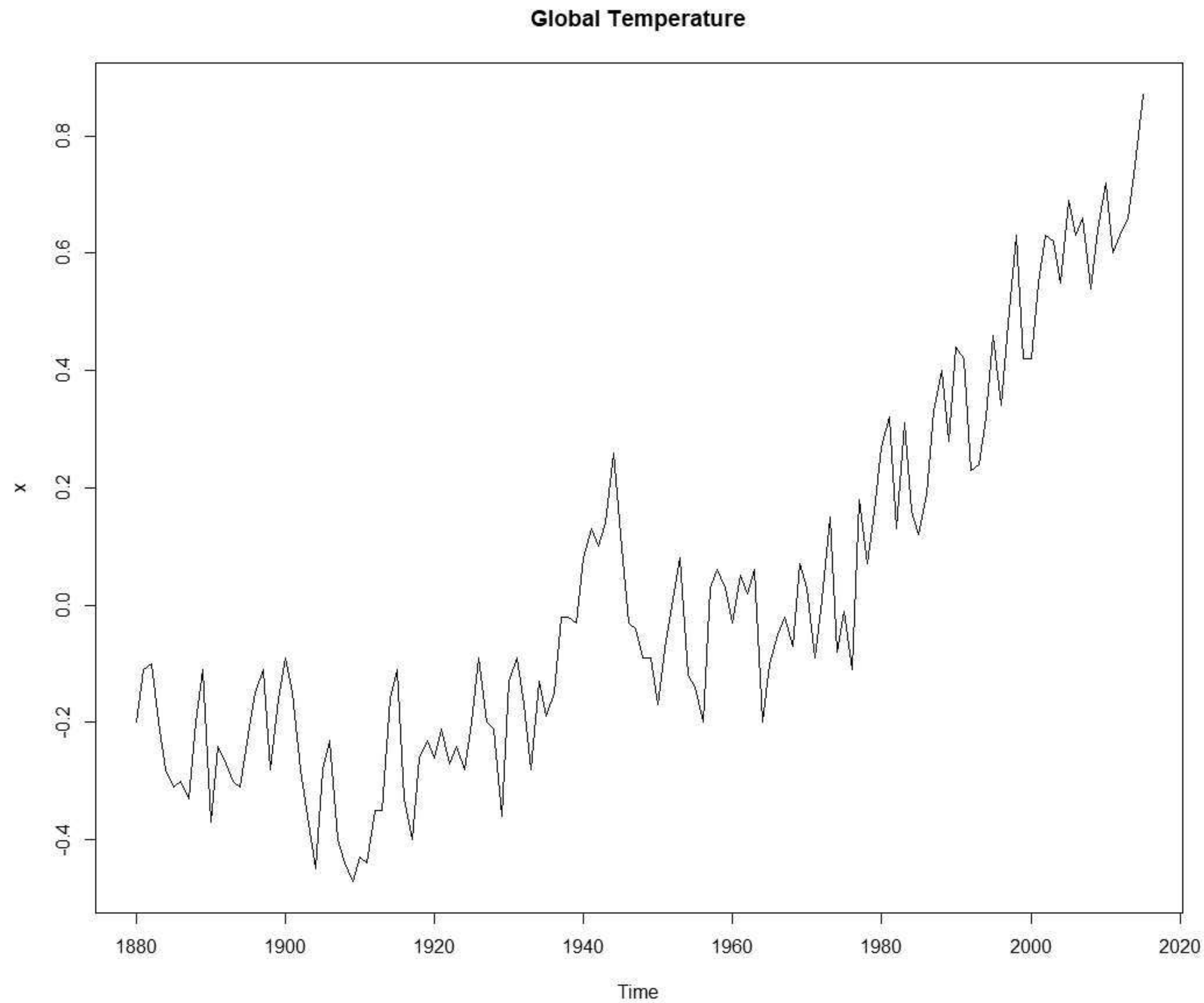
```
rep(0, 3) -> aic -> aicc -> bic
for (i in 1:3) {
  m = sarima(l, i,0,0)
  aic[i] = m$AIC
  aicc[i] = m$AICc
  bic[i] = m$BIC
}
> cbind('AIC'=aic, 'AICc'=aicc, 'BIC'=bic)
           AIC      AICc       BIC
[1,] -3.285544 -3.285504 -3.261837
[2,] -3.289216 -3.289134 -3.257606
[3,] -3.311520 -3.311383 -3.272007
```

Running similar code for the 2nd differenced data for ARMA(0,2,i), we find that the MA(3) fits best as per the information criterion (not shown). The residuals (not shown) are similar to that of the ARMA(3,1,0), though there seems to be more autocorrelation and the Q statistics are slightly significant for all lags. The AIC/AICc prefer the ARMA(3,1,0), but the BIC prefers the ARMA(0,2,3).Running similar code for ARMA(i,2,0) produces similar yet slightly worse results.

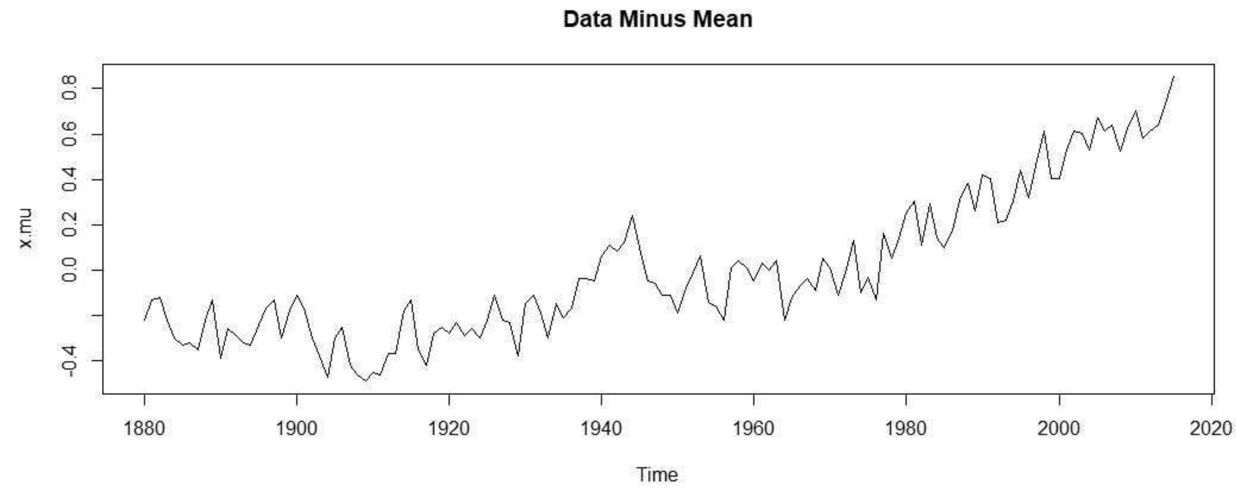
3.33

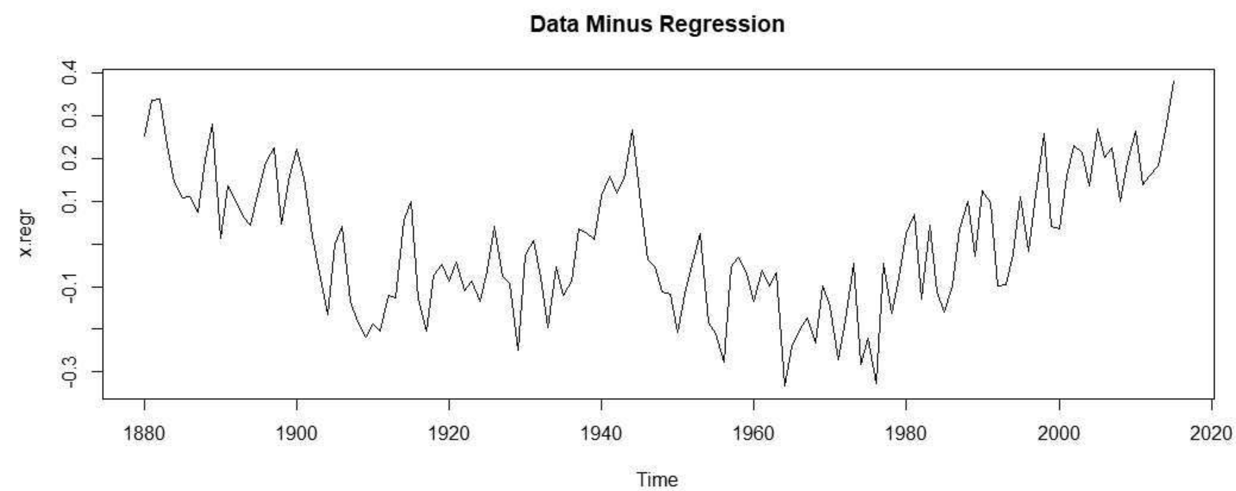
3.33 Fit an ARIMA(p, d, q) model to the global temperature data `globtemp` performing all of the necessary diagnostics. After deciding on an appropriate model, forecast (with limits) the next 10 years. Comment.

Looking at the plot, we can see that there is a clear trend, though the variance remains relatively the same across the series. There is a spike in variation in the middle of the series and the beginning of the series appears to have a different mean than the rest of the data. We can clearly see from the ACF and PACF (not shown) that the data is not stationary.

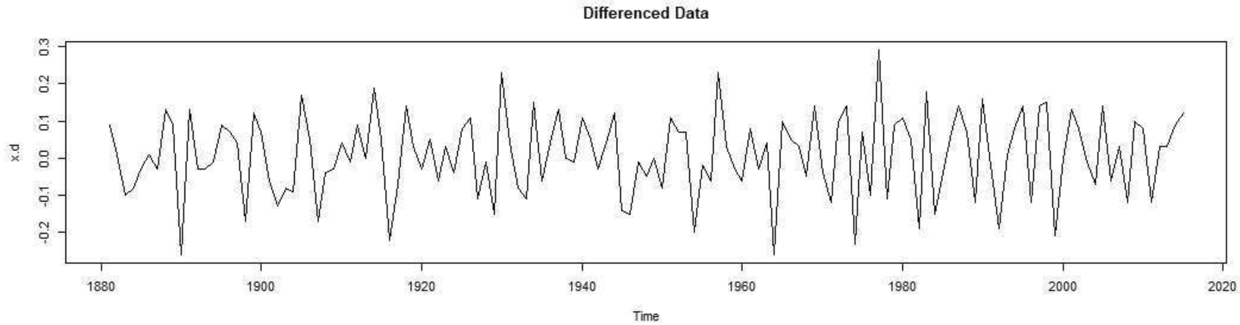


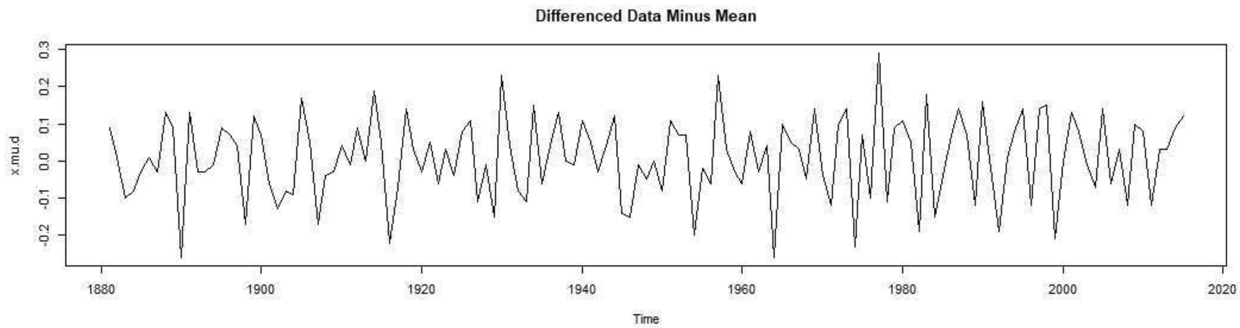
We may consider subtracting a mean constant from the data, either using the sample mean or a regression fit, as shown below. As we can see, subtracting the regression does slightly better at making the data stationary, however, there still remains a difference in mean across the series and slight heteroskedasticity.

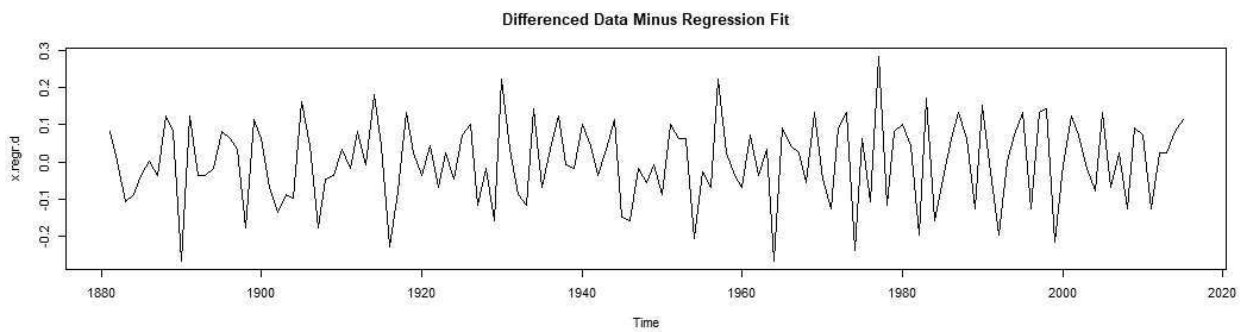




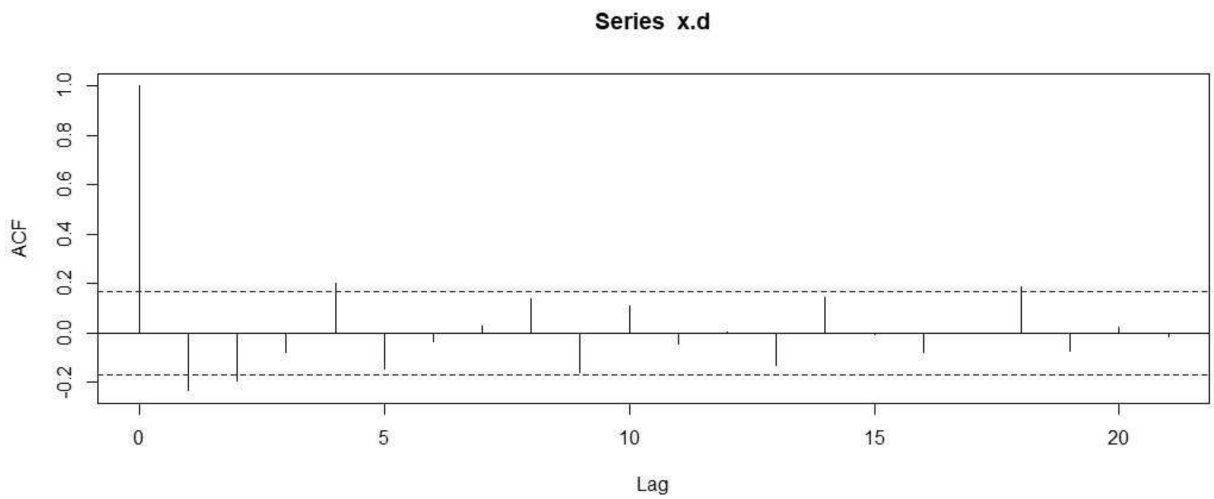
Because the data, and the mean subtraction transformation, contain negative values, we are not able to take the log. However, we can still do differencing. Interestingly, differencing the data, differencing the data minus the sample mean and differencing the data minus the regression fit all produce the same results.

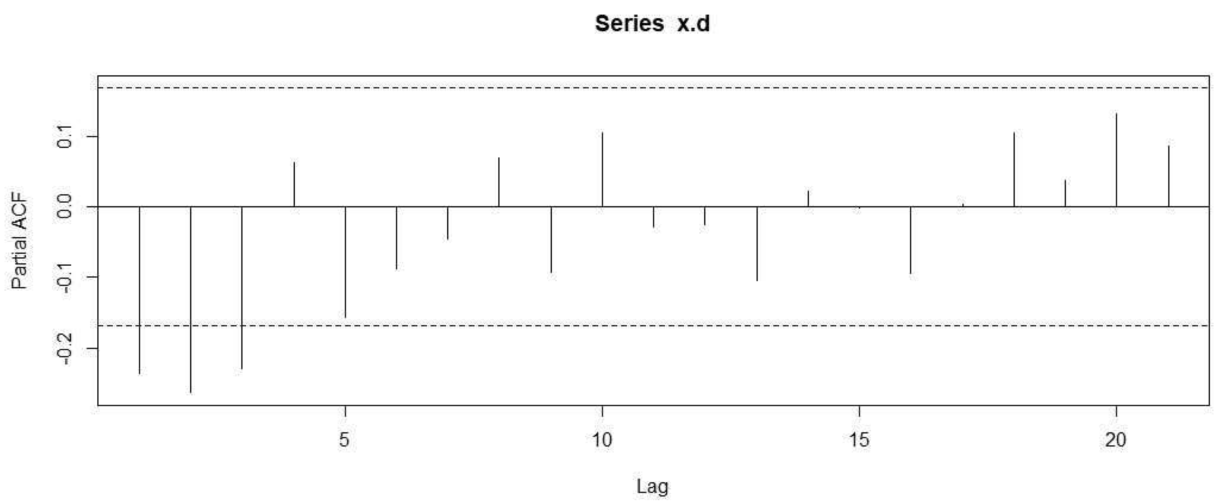






The differenced data by itself looks stationary and so we shall deal with that.
Looking at the ACF and PACF, we can see three fairly decisive lags on the PACF, after which they drop off. On the ACF we can see the first two lags over the significance line, as well as the fourth lag, though the third lag is insignificant. We will try various ARMA models on the differenced data.



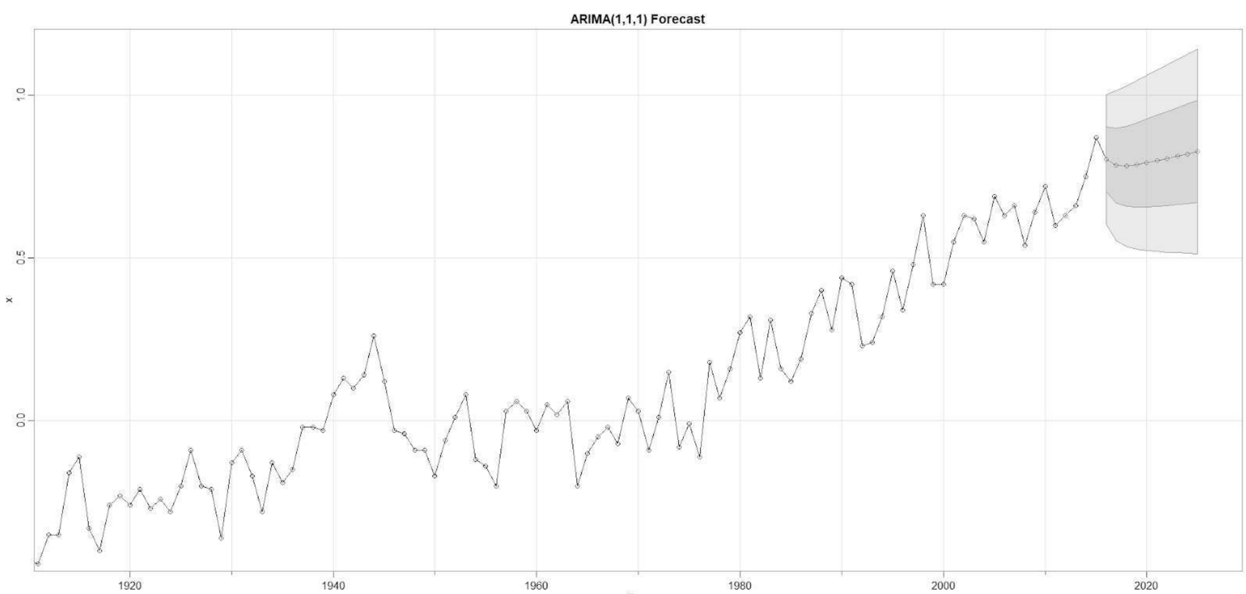


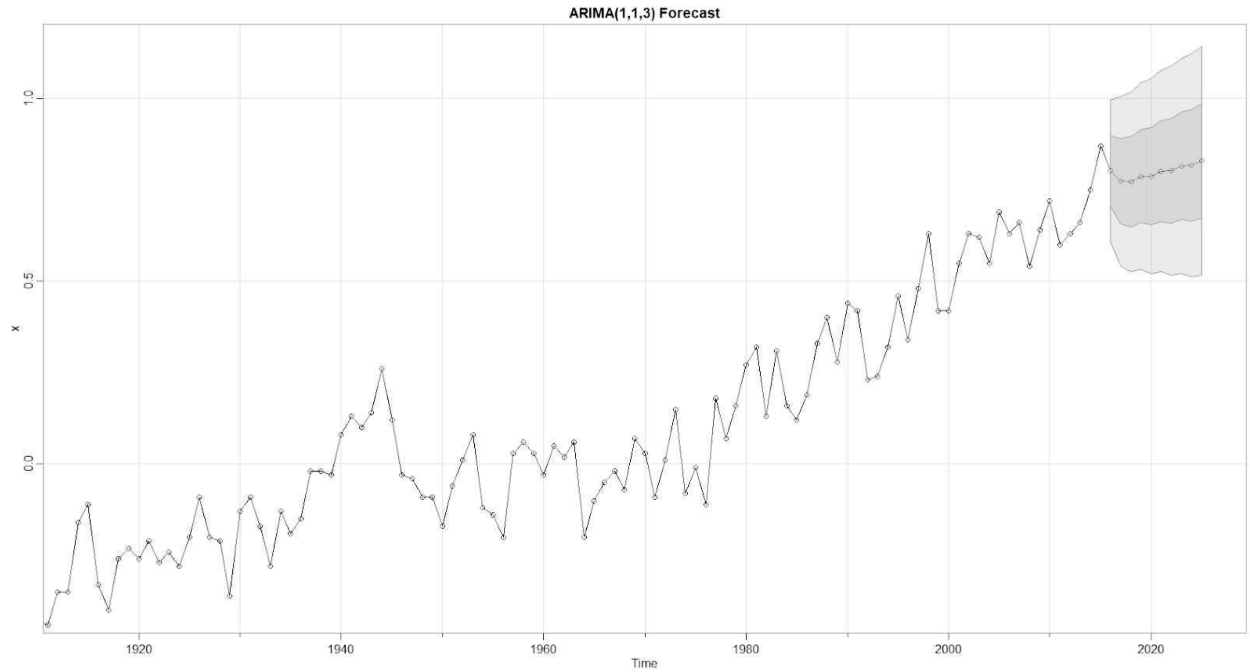
Running the code for various ARMA models, we can see that the BIC prefers the ARMA(1,1), but the AIC and AICc prefer the ARMA(1,3). So we will look at the residuals for these.

```
rep(0, 12) -> aic -> aicc -> bic -> icol -> jcol
k = 1
for (i in 1:3) {
  for (j in 1:4) {
    m = sarima(x.d,i,0,j, details=FALSE)
    aic[k] = m$AIC
    aicc[k] = m$AICc
    bic[k] = m$BIC
    icol[k] = i
    jcol[k] = j
    k = k + 1
  }
}
```

```
> cbind("p"=icol,"q"=jcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
      p q       AIC      AICc       BIC
 [1,] 1 1 -1.716773 -1.715416 -1.630691
 [2,] 1 2 -1.709510 -1.707230 -1.601907
 [3,] 1 3 -1.734246 -1.730800 -1.605122
 [4,] 1 4 -1.720751 -1.715890 -1.570107
 [5,] 2 1 -1.712725 -1.710446 -1.605123
 [6,] 2 2 -1.692799 -1.689353 -1.563675
 [7,] 2 3 -1.713117 -1.708255 -1.562473
 [8,] 2 4 -1.701724 -1.695192 -1.529560
 [9,] 3 1 -1.698221 -1.694776 -1.569098
[10,] 3 2 -1.719356 -1.714495 -1.568712
[11,] 3 3 -1.708561 -1.702029 -1.536397
[12,] 3 4 -1.685284 -1.676819 -1.491599
```

Looking at the diagnostics (not shown) for the ARMA(1,1) model, we can see that the residuals are reasonably normal, though there are a number of outliers violating the tails. No data points go beyond the 95% confidence interval. Every lag is insignificant on the Q statistic and the ACF generally shows no autocorrelation.
The residuals on the ARMA(1,3) are fairly similar, though the outliers are greater. The ACF also shows no autocorrelation and the Q statistics are not significant (even further away from the significance level).
The forecasts are given below using sarima.for.





3.34

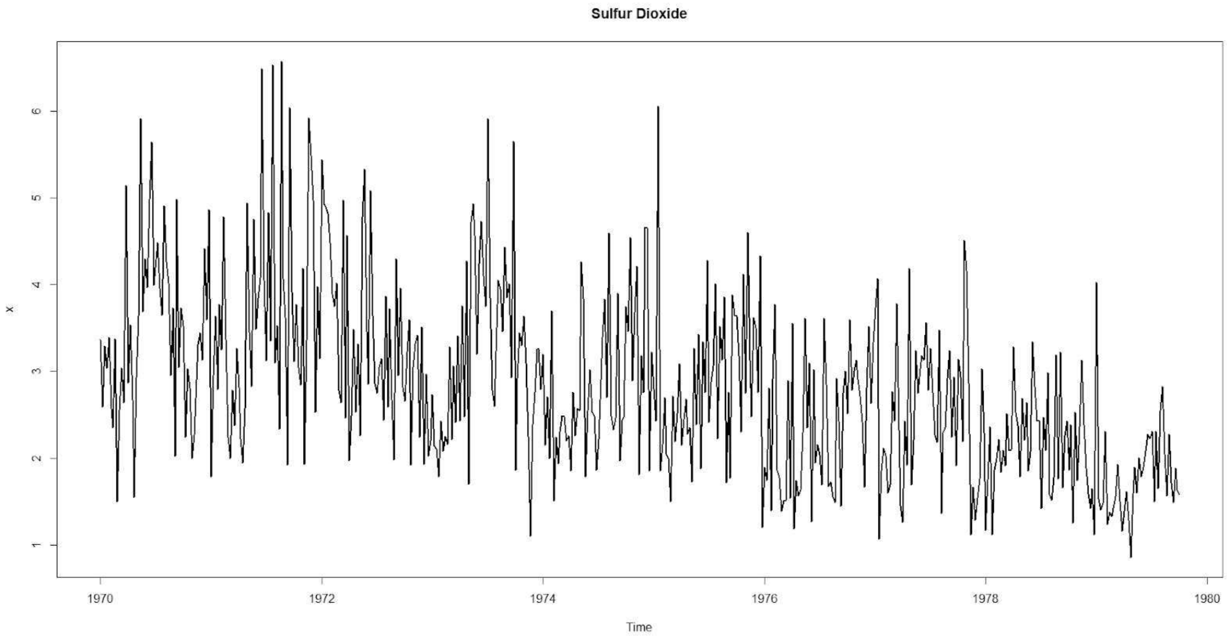
3.34 Fit an ARIMA(p, d, q) model to the sulfur dioxide series, so2, performing all of the necessary diagnostics. After deciding on an appropriate model, forecast the data into the future four time periods ahead (about one month) and calculate 95% prediction intervals for each of the four forecasts. Comment. (Sulfur dioxide is one of the pollutants monitored in the mortality study described in Example 2.2.)

From the below plot, we can see that there is a slight, relatively constant, downwards trend across the series. The variance of the data remains roughly the same except that it seems to be decreasing slightly near the end. Looking at the ACF and PACF (now shown) of the data, we will argue that the data is not stationary given that they both tail off. It may be possible to fit a seasonal model to it, but we won't be doing that.

To transform the data, we look at the logged data and differenced data separately. The logged data has a relatively constant variance but appears to have different means across the series. The differenced data appears to have a constant mean and a relatively stable variance, however, there are a few outliers. The ACF of the differenced data cuts off at lag 1 and tails off from lag 1 on the PACF, suggesting an ARIMA(0,1,1) for so2. The ACF and PACF of the logged data both tail off. The ACF very slowly and the PACF quicker but definitely not cutting off. This suggests that the logged data is still stationary.

We are not able to log the differenced data as it now contains negative values, however, we will take a look at the differenced logged data. The plotted differenced logged data looks much the same as the differenced only data, except that it is scaled down. The ACF and PACF of the differenced logged data are similar to the ACF and PACF of the differenced data as well. For this reason, we will run a model for the differenced data only.

We will look at various potential ARMA models.



As per the below code, surprisingly, the AIC, AICc and the BIC all prefer the ARIMA(4,1,2) model.

```
rep(NA, 16) -> aic -> aicc -> bic -> pcol -> qcol
i = 1
for (p in 1:4) {
  for (q in 1:4) {
    m = sarima(x.d, p,0,q, details=FALSE) #x.d is the differenced data
    aic[i] = m$AIC
    aicc[i] = m$AICc
    bic[i] = m$BIC
    pcol[i] = p
    qcol[i] = q
    i = i + 1
  }
}
> cbind("p"=pcol,"q"=qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
      p q      AIC     AICc      BIC
 [1,] 1 1 2.612059 2.612153 2.645420
 [2,] 1 2 2.606509 2.606666 2.648210
 [3,] 1 3 2.607257 2.607494 2.657299
 [4,] 1 4 2.609475 2.609806 2.667857
 [5,] 2 1 2.609312 2.609469 2.651013
 [6,] 2 2 2.607581 2.607817 2.657622
 [7,] 2 3 2.606446 2.606777 2.664828
```

```
 [8,] 2 4 2.610384 2.610826 2.677106
 [9,] 3 1 2.606445 2.606681 2.656487
[10,] 3 2 2.609937 2.610268 2.668318
[11,] 3 3 2.610383 2.610826 2.677105
[12,] 3 4 2.583744 2.584314 2.658806
[13,] 4 1 2.588275 2.588606 2.646657
[14,] 4 2 2.578262 2.578705 2.644984
[15,] 4 3 2.581918 2.582488 2.656980
[16,] 4 4 2.614533 2.615247 2.697935
```

For the ARIMA(4,1,2), the residuals violate normality at the tails. Some of the outliers go beyond the 95% confidence interval on the QQ plot. Despite this, the p values of the Ljung-Box statistic for all lags shown are insignificant. Also the ACF shows that the errors are uncorrelated and the standardised residuals plot looks fairly normal.
If we find models to the differenced logged data (not shown), we find the ARMA(4,2) and ARMA(4,3) are the best, though the BIC now does not agree with the AIC/AICc. Looking at the diagnostics for these two fits, we see that the outliers are removed and no data points go beyond the 95% confidence interval. Not much else has changed.

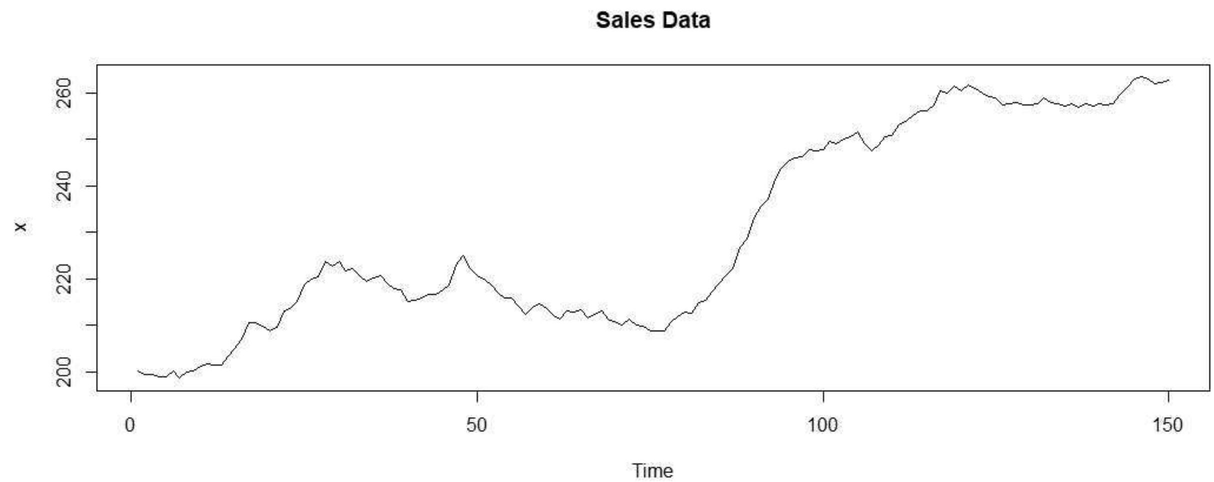
3.35

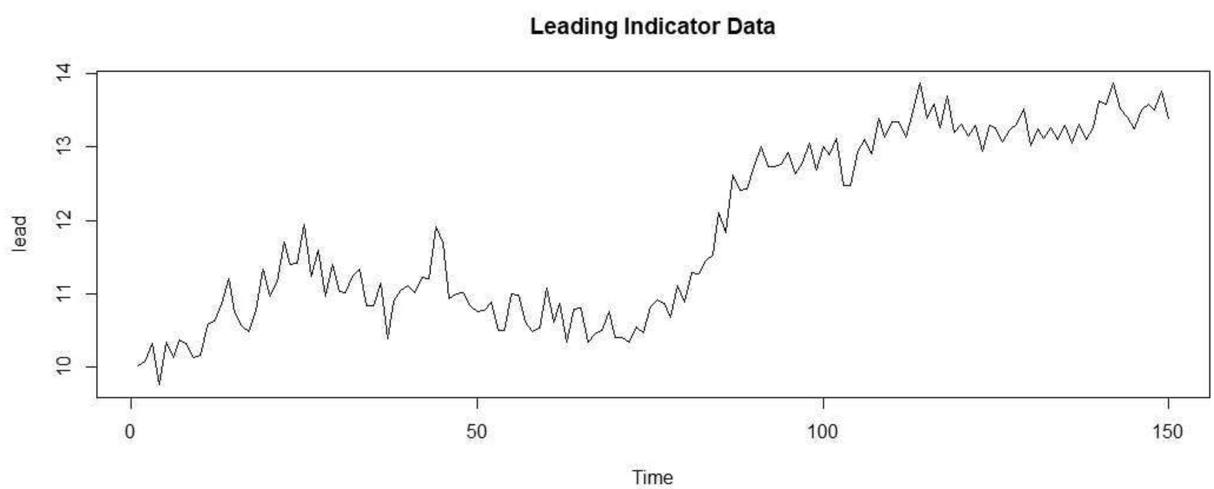
3.35 Let S_t represent the monthly sales data in `sales` ($n = 150$), and let L_t be the leading indicator in `lead`.

(a) Fit an ARIMA model to S_t, the monthly sales data. Discuss your model fitting in a step-by-step fashion, presenting your (A) initial examination of the data, (B) transformations, if necessary, (C) initial identification of the dependence orders and degree of differencing, (D) parameter estimation, (E) residual diagnostics and model choice.

Looking at the plotted data for sales and lead, we can see that there is a fairly steep uptrend in both. The data sets look highly correlated. The sales data is smoother, with the variance being roughly the same across the series except for the large spike in the middle. The lead data is similar in homoscedasticity to the sales data, except that the lead data experiences more shorter term variation.
The data sets have similar ACF and PACF plots (not shown), where the ACF is slowly tailing off and the PACF cuts off at lag 1 for the sales data and possibly lag 2 for the lead data. Although we might think that they are ready for an AR(1) or AR(2), the ACF is clearly showing high levels of autocorrelation. So, we still need to transform the data.
Looking at the logged and differenced transformations separately, we can see that differencing does the best. Both data sets still show a strong trend after being logged, but look reasonably stationary after differencing. The sales data still seem to show different mean levels across the series, so we turn to further transformations on this set. Differencing the logged sales data

improves the stationarity, however, different mean leaves are still apparent. A second differencing on the sales data improves the stationarity the most, so we will take this as our transformed data.





Looking at the ACF and PACF of the transformed data, we can see that the ACFs for both sets cut off at lag 1. The PACF of the transformed sales data is either cutting off at lag 3 or tailing off. The PACF for the differenced lead data looks similar to its sales data counterpart, however, lag 2 and 3 are under the significance level.

First, we will try various ARIMA models on the data. Given the below code, it's fairly unanimous that the ARIMA(1,1,1) is the preferred model. This is despite our analysis arguing that a second differencing on the sales data is necessary to produce stationarity. What we failed to do was look at the ACF and PACF of this first differenced data. If we had, we would have seen that the plots were not showing stationarity. It's important to not over-difference, as it may lead to showing correlation that's not really there.

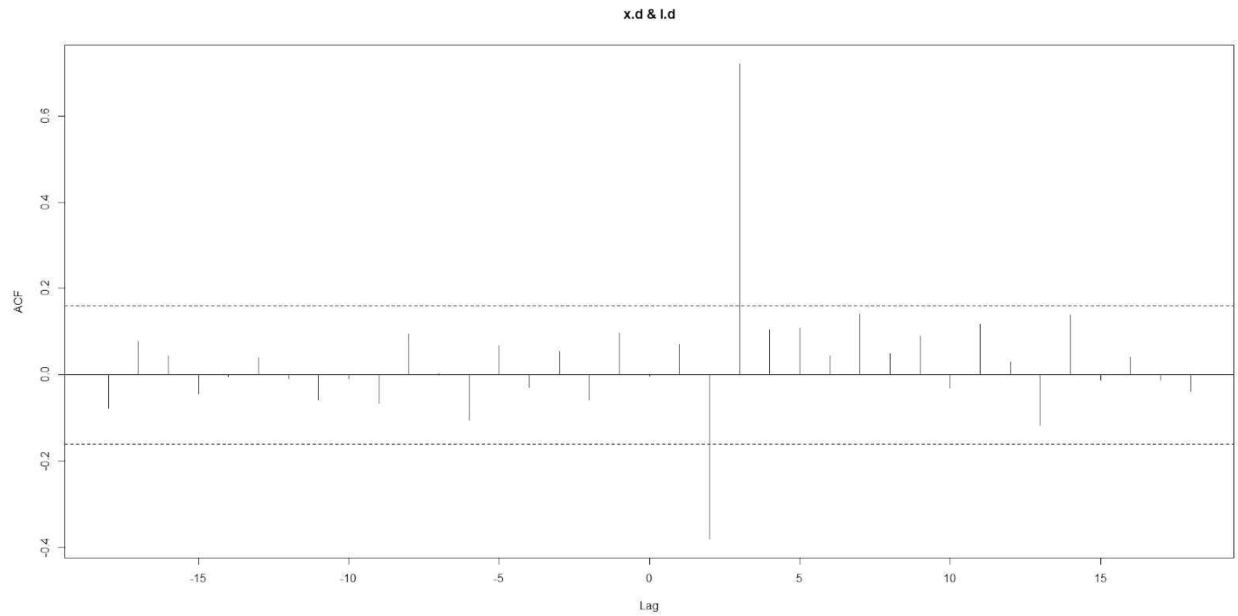
```
rep(NA, 36) -> aic -> aicc -> bic -> pcol -> dcol -> qcol
i = 1
```

```
for (p in 0:3) {
  for (q in 0:2) {
    for (d in 0:2) {
      m = sarima(x, p,d,q, details=FALSE)
      aic[i] = m$AIC
      aicc[i] = m$AICc
      bic[i] = m$BIC
      pcol[i] = p
      qcol[i] = q
      dcol[i] = d
      i = i + 1
    }
  }
}
> cbind("p"=pcol,"d"=dcol,"q"=qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
      p d q      AIC     AICc      BIC
 [1,] 0 0 0 8.992070 8.992250 9.032212
 [2,] 0 1 0 3.592821 3.593004 3.633142
...
[14,] 1 1 1 3.454924 3.456035 3.535567
[15,] 1 2 1 3.506577 3.507136 3.567331
[16,] 1 0 2 3.607752 3.609591 3.708107
[17,] 1 1 2 3.467309 3.469173 3.568113
...
[32,] 3 1 1 3.480270 3.483086 3.601235
[33,] 3 2 1 3.509582 3.511472 3.610839
[34,] 3 0 2 3.537900 3.541816 3.678396
[35,] 3 1 2 3.490013 3.493983 3.631138
[36,] 3 2 2 3.509385 3.512240 3.630893
```

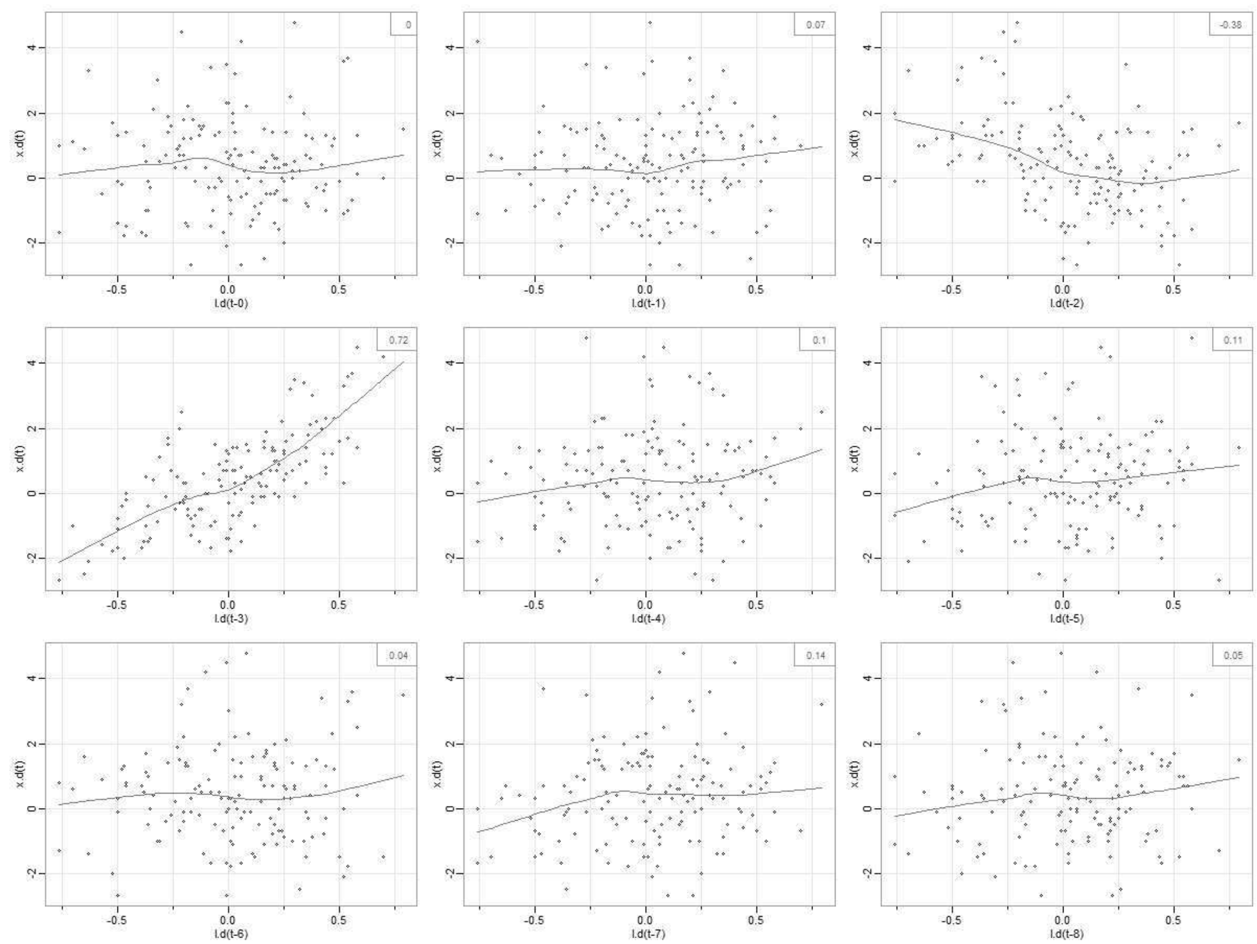
The diagnostics of the ARIMA(1,1,1) model for the second differenced data show the residuals to be reasonably normal. There are a few outliers on the QQ plot, but all are within the 95% confidence interval. The ACF shows low autocorrelation and the p values of the Ljung-Box statistic are all insignificant.

(b) Use the CCF and lag plots between ∇S_t and ∇L_t to argue that a regression of ∇S_t on ∇L_{t-3} is reasonable. [*Note that in* `lag2.plot()`, *the first named series is the one that gets lagged.*]

We can see from the CCF of the differenced data sets that there is a high correlation at lag 3. This suggests that the sales data at time t are positively correlated with the lead data at time t-3.



Also, by looking at lag2.plot(l.d, x.d, 8) we can see again that there is high correlation at lag 3, especially compared to other lags.

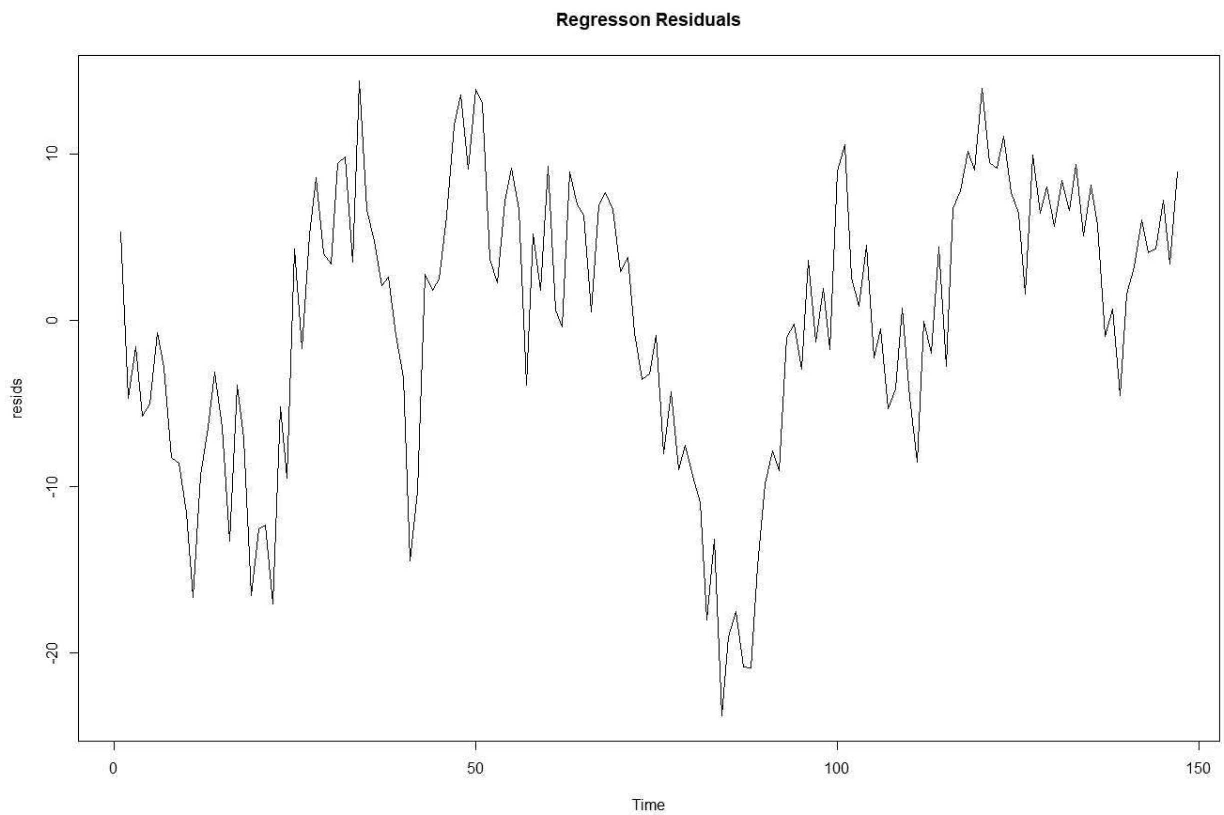


Because of this correlation, we argue that running a regression of sales on lead(t-3) would be reasonable.

(c) Fit the regression model $\nabla S_t = \beta_0 + \beta_1 \nabla L_{t-3} + x_t$, where x_t is an ARMA process (explain how you decided on your model for x_t). Discuss your results. [*See Example 3.45 for help on coding this problem.*]

```
fish = ts.intersect(x, lL=lag(l, 3))
m = lm(x ~ lL, data=fish)
resids = m$residuals
```

Looking at a time series plot of the residuals, we can see that there appears to be some autocorrelation. The ACF (not shown) also shows stationarity, though the PACF cuts off at lag 1. We cannot log the data, as there are negative values, however, differencing the data produces fairly stationary results.
The ACF of the transformed residuals cuts off at lag 1. The PACF cuts off at lag 1 as well, though lags 7 and 10 are above the significance level.



Testing for various ARIMA models on the residuals, similar to the above code for the sales data, yields an ARMA(2,0) as the most preferred. The diagnostics for this model show that the standardised residuals are reasonably normal. The few outliers lie within the 95% confidence interval on the QQ plot and the ACF generally shows low autocorrelation. However, the p values for the Ljung-Box statistics approach the level of significance at some lags.

3.36

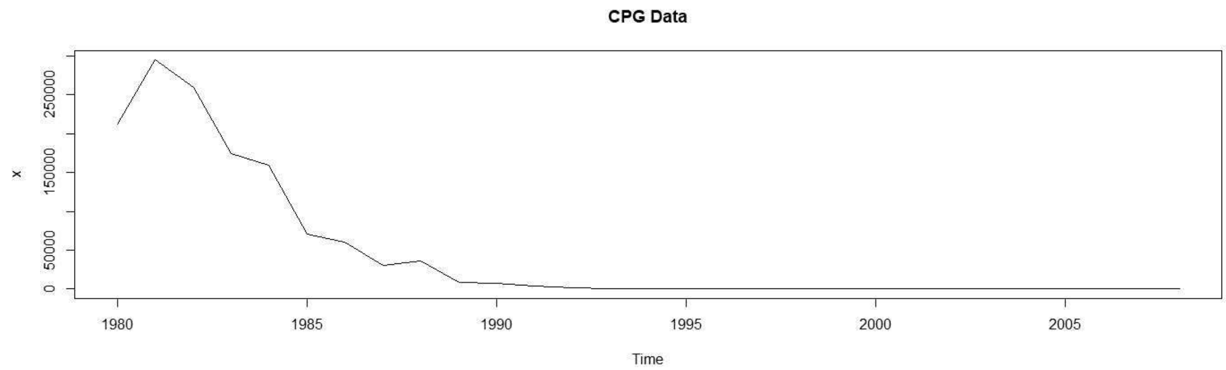
3.36 One of the remarkable technological developments in the computer industry has been the ability to store information densely on a hard drive. In addition, the cost of storage has steadily declined causing problems of *too much data* as opposed to *big data*. The data set for this assignment is `cpg`, which consists of the median annual retail price per GB of hard drives, say c_t, taken from a sample of manufacturers from 1980 to 2008.

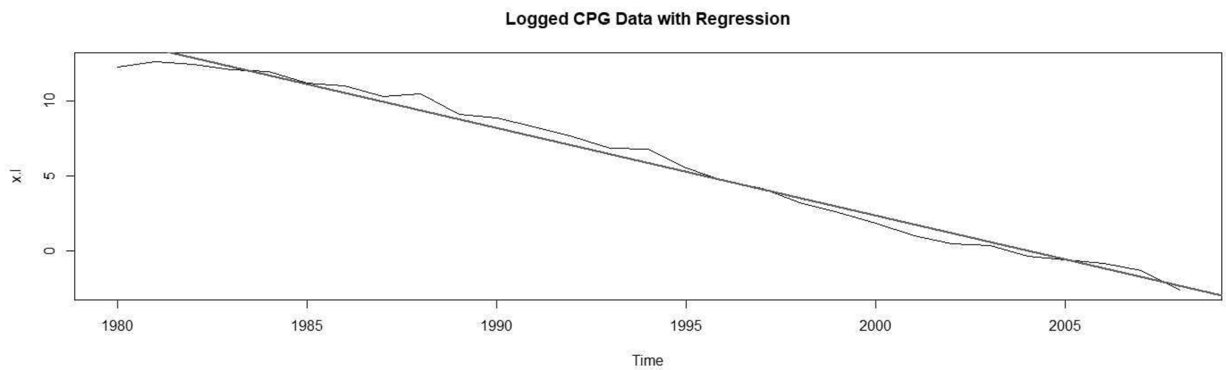
(a) Plot c_t and describe what you see.

We can see below that the CPG data declines rapidly and then asymptotes the time horizon.

(b) Argue that the curve c_t versus t behaves like $c_t \approx \alpha e^{\beta t}$ by fitting a linear regression of $\log c_t$ on t and then plotting the fitted line to compare it to the logged data. Comment.

This is reminiscent of a type of exponential graph. We confirm this by looking at the logged data. There is a fairly strong linear relationship between the logged data and the time horizon. Because of this we will try a linear regression on the logged data. As seen on the same plot, the regression fit is very well suited to the logged data, which strongly suggests that the original data is following a type of exponential function.

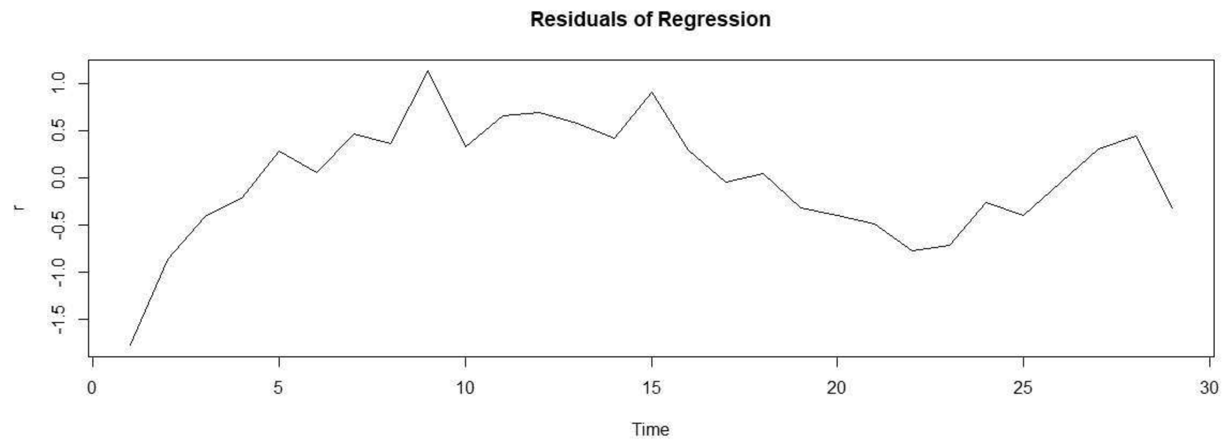


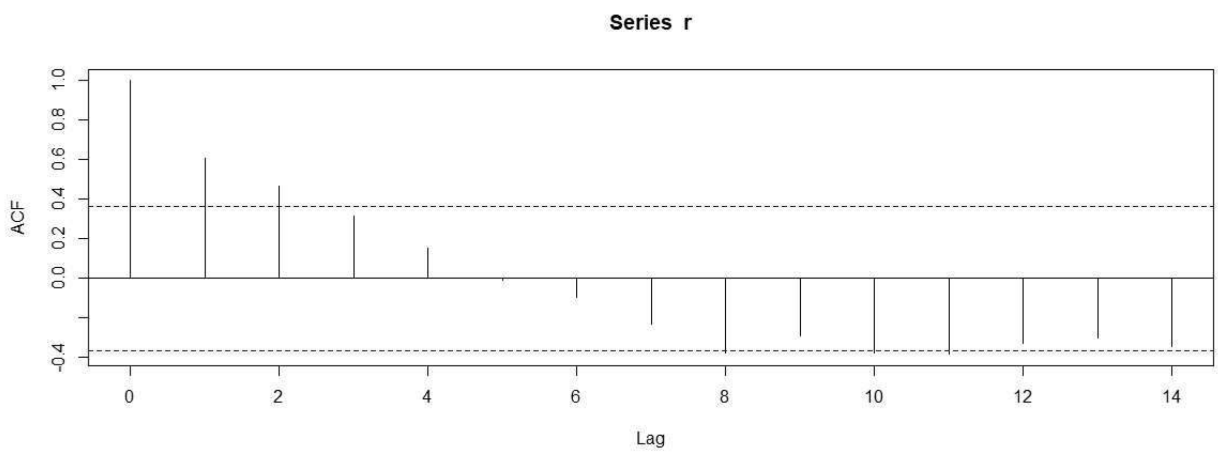



```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 1172.49431   27.57793   42.52   <2e-16 ***
time(x)       -0.58508    0.01383  -42.30   <2e-16 ***
```

(c) Inspect the residuals of the linear regression fit and comment.

Both coefficients are significant. However, the residuals appear to be autocorrelated as per the plot and ACF. Though the ACF is tailing off, the PACF of the residuals (not shown) suggests an AR(1).





Testing various ARMA models on the residuals, we see that the BIC and AICc prefer the AR(1). The AIC prefers the ARMA(3,1) however this is probably an overfit.

```
> cbind("p"=pcol,"q"=qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
      p q      AIC     AICc      BIC
 [1,] 0 0 1.958358 1.963467 2.052655
 [2,] 0 1 1.657886 1.673801 1.799330
 [3,] 0 2 1.526058 1.559162 1.714651
 [4,] 0 3 1.411608 1.469079 1.647349
 [5,] 1 0 1.292033 1.307948 1.433478
 [6,] 1 1 1.352060 1.385164 1.540653
 [7,] 1 2 1.340559 1.398031 1.576300
 [8,] 1 3 1.368566 1.458521 1.651455
 [9,] 2 0 1.345952 1.379056 1.534545
[10,] 2 1 1.393812 1.451284 1.629553
[11,] 2 2 1.282810 1.372765 1.565699
[12,] 2 3 1.437506 1.569167 1.767543
[13,] 3 0 1.371138 1.428609 1.606878
[14,] 3 1 1.262090 1.352045 1.544979
```

```
[15,] 3 2 1.314670 1.446332 1.644707
[16,] 3 3 1.469064 1.652972 1.846249
```

(d) Fit the regression again, but now using the fact that the errors are autocorrelated. Comment.

We rerun the regression on the logged data and incorporate the AR(1) on the residuals.

```
s.r = sarima(x.l, 1,0,0, xreg=time(x))
Coefficients:
         ar1  intercept     xreg
      0.8297  1113.0105  -0.5554
s.e.  0.1190    73.5665   0.0368
```

The diagnostics of this fit are very good. The standardised residuals look like white noise and the ACF shows very little autocorrelation. The data points almost perfectly align on the QQ plot and all lags of the Ljung-Box statistic are insignificant.

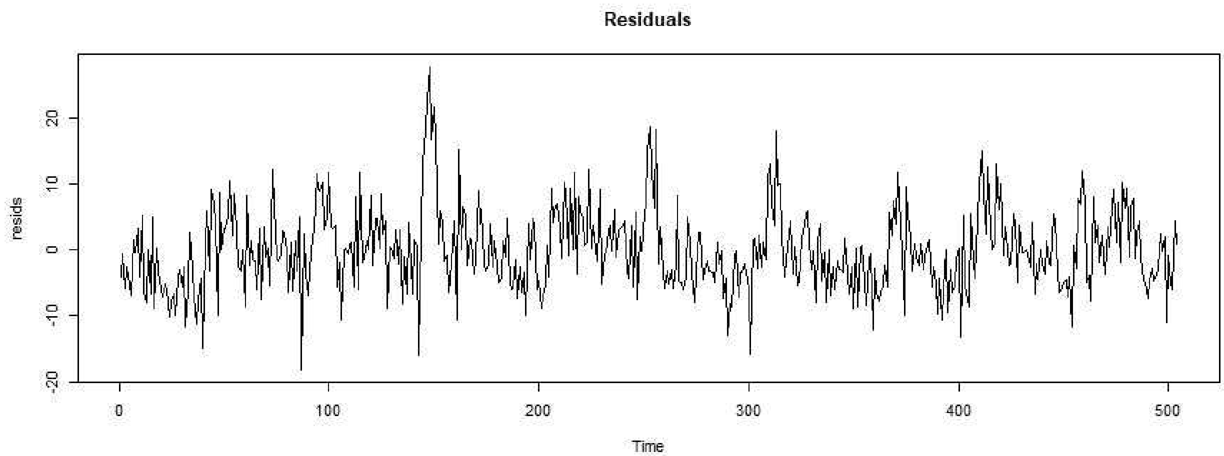
3.37

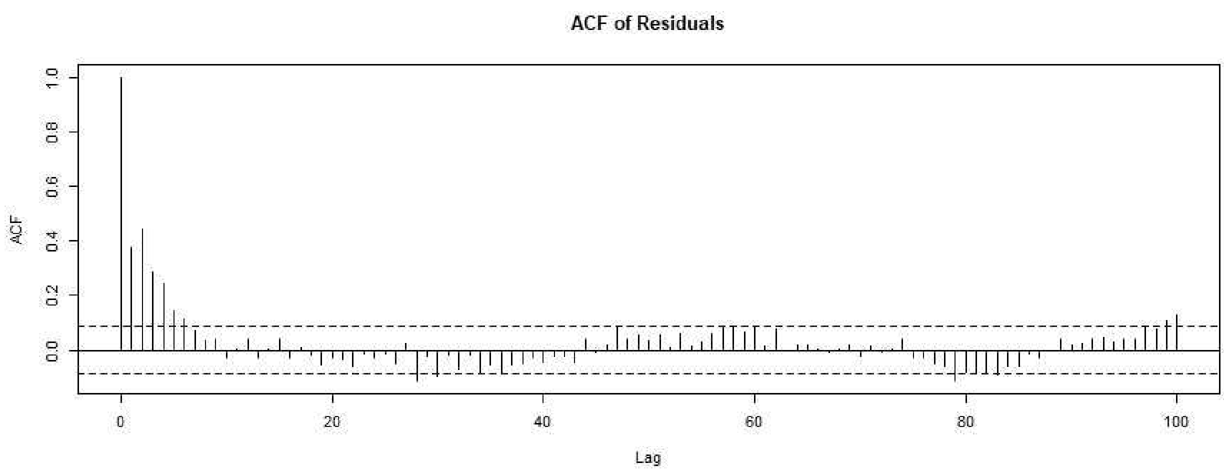
3.37 Redo Problem 2.2 without assuming the error term is white noise.

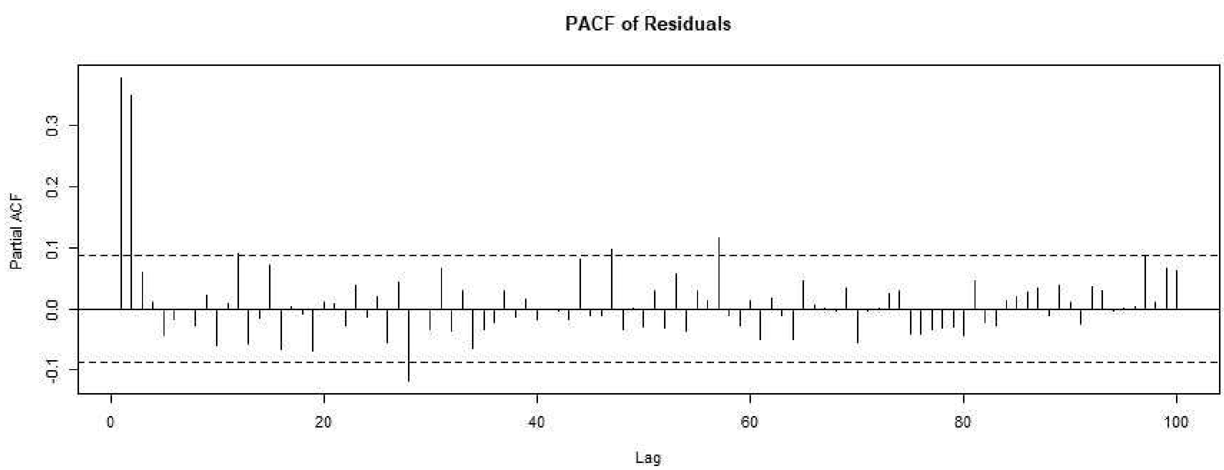
Running the same code and then inspecting the residuals, we see

```
summary(fit)
par(mfrow=c(3,1))
resids = fit$residuals
plot.ts(resids, main="Residuals")
acf(resids, main="ACF of Residuals", lag.max=100)
pacf(resids, main="PACF of Residuals", lag.max=100)
```

We can see that although the residuals series is stationary, there is clear autocorrelation, as per the ACF and PACF. The ACF is decaying exponentially, while the AR appears to cut off after the 1st or 2nd lag. This suggests an AR(2).







Running ARMA on the residuals

```
rep(NA, 9) -> aic -> aicc -> bic -> pcol -> qcol
i = 1
```

```
for (p in 0:2) {
  for (q in 0:2) {
    m = sarima(resids, p, 0, q, details=FALSE)
    aic[i] = m$AIC; aicc[i] = m$AICc; bic = m$BIC
    pcol[i] = p; qcol[i] = q
    i = i + 1
  }
}
cbind("p"=pcol,"q"=qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
      p q      AIC     AICc      BIC
 [1,] 0 0 6.510952 6.510967 6.290281
 [2,] 0 1 6.426930 6.426977 6.290281
 [3,] 0 2 6.307723 6.307818 6.290281
 [4,] 1 0 6.361277 6.361325 6.290281
 [5,] 1 1 6.266197 6.266292 6.290281
 [6,] 1 2 6.236986 6.237145 6.290281
 [7,] 2 0 6.235910 6.236005 6.290281
 [8,] 2 1 6.236403 6.236562 6.290281
 [9,] 2 2 6.240013 6.240252 6.290281
```

Each IC confirms our suggestion that the AR(2) would be most appropriate.

3.38

3.38 Consider the ARIMA model

$$x_t = w_t + \Theta w_{t-2}.$$

(a) Identify the model using the notation ARIMA$(p, d, q) \times (P, D, Q)_s$.

ARIMA(0,0,2)x$(0,0,0)_s$, where s is anything, or ARIMA(0,0,0)x$(0,0,1)_2$.

(b) Show that the series is invertible for $|\Theta| < 1$, and find the coefficients in the representation

$$w_t = \sum_{k=0}^{\infty} \pi_k x_{t-k}.$$

Because we can write this model as an MA(2), where $\theta(z) = 1 + \Theta z^2$, we know that there exists an invertible representation given $|\Theta| < 1$, as per Property 3.2. To find this representation, we note

$$\frac{1}{1-(-\Theta z^2)} = \sum_{j=0}^{\infty}(-^2)^j = \sum_{k=0} \infty \pi_k z^k$$

If we write these out, we can see

$(-\Theta z^2)^0 + (-\Theta z^2)^1 + (-\Theta z^2)^2 + ... = \pi_0 z^0 + \pi_1 z^1 + \pi_2 z^2 + ...$

Hence,

$(-\Theta)^0 = \pi_0, \;\; (-\Theta)^1 = \pi_2, \;\; (-\Theta^2) = \pi_4$

And in general

$\pi_{2j} = (-\Theta)^j, \;\; \pi_{2j+1} = 0$ for $j = 0, 1, 2, ...$.

Hence,

As per Definition 3.8,

$$w_t = \sum_{j=0}^{\infty}(-\Theta)^j x_{t-2j}$$

(c) Develop equations for the m-step ahead forecast, $\tilde{x}_{n+m}$, and its variance based on the infinite past, $x_n, x_{n-1}, \ldots$.

Taking conditional expectations on (3.82), we can write

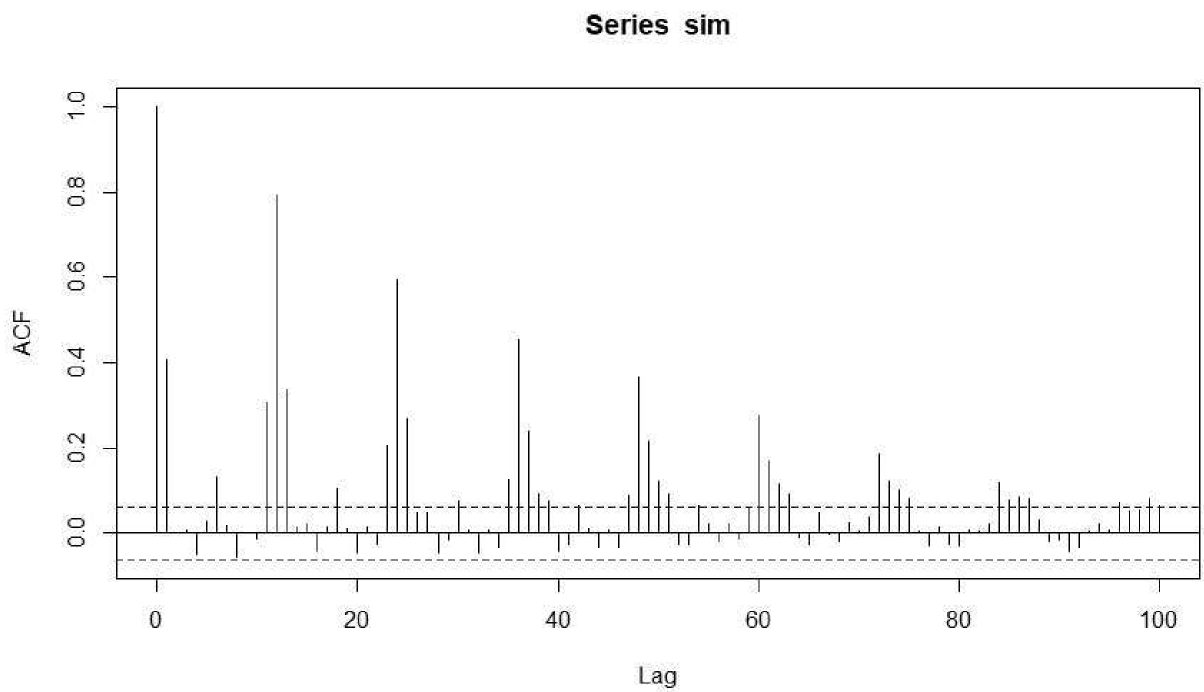
$$\tilde{x}_{n+m} = -\sum_{k=1}^{\infty}(-\Theta)^k \tilde{x}_{n+m-2k}$$

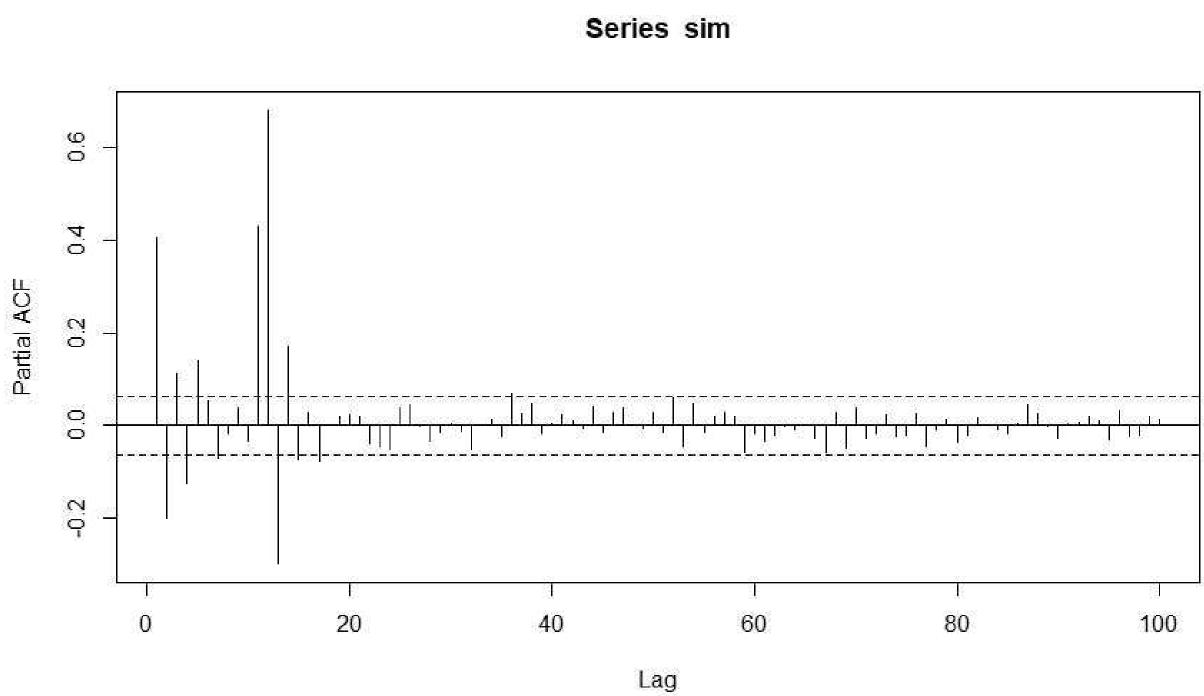
where $\tilde{x}_t = x_t$ for $t \leq n$. For the prediction error, note that $\psi_0 = 1$, $\psi_2 = \Theta$ and $\psi_j = 0$ otherwise. Thus, $P^n_{n+m} = \sigma^2_w$ for $m = 1, 2$; when $m > 2$ we have $P^n_{n+m} = \sigma^2_w(1 + \Theta^2)$.

3.39

3.39 Plot the ACF of the seasonal ARIMA$(0,1) \times (1,0)_{12}$ model with $\Phi = .8$ and $\theta = .5$.

```
phi = c(rep(0,11), .8); theta = .5
n = 1000
sim = arima.sim(list(order=c(12,0,1), ar=phi, ma=theta), n)
acf(sim)
pacf(sim)
```



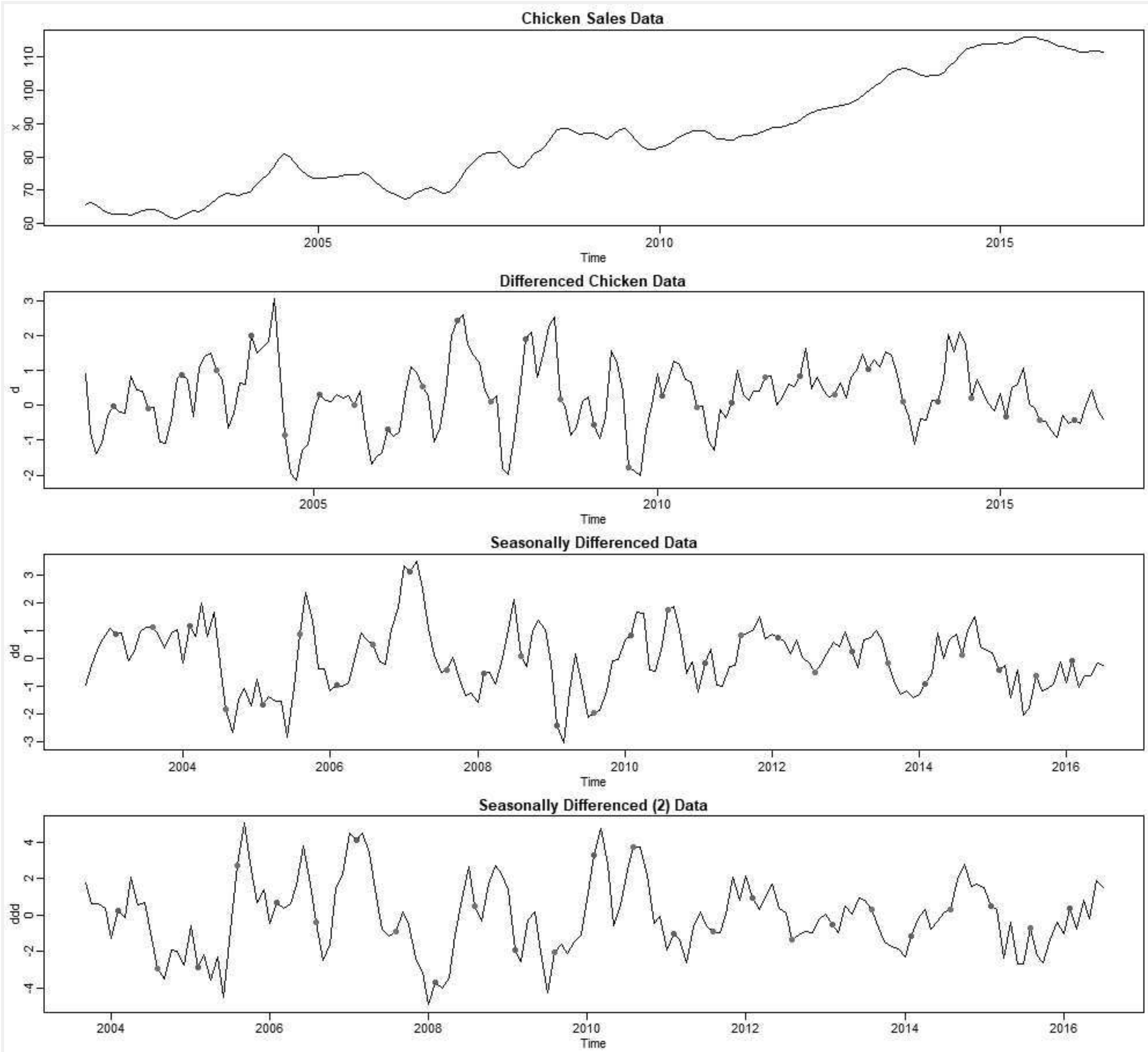


We see seasonality at every 12th lag on the ACF, where it tends to drop off slowly. This suggests an SAR(12). On the PACF, we see a large spike at the 12th lag, with it dropping off slowly, however, no more seasonal lags like this appear. The ACF cuts off at the beginning after the 1st lag and is dropping off slowly on the PACF, suggesting an MA(1) component.

3.40

3.40 Fit a seasonal ARIMA model of your choice to the chicken price data in `chicken`. Use the estimated model to forecast the next 12 months.

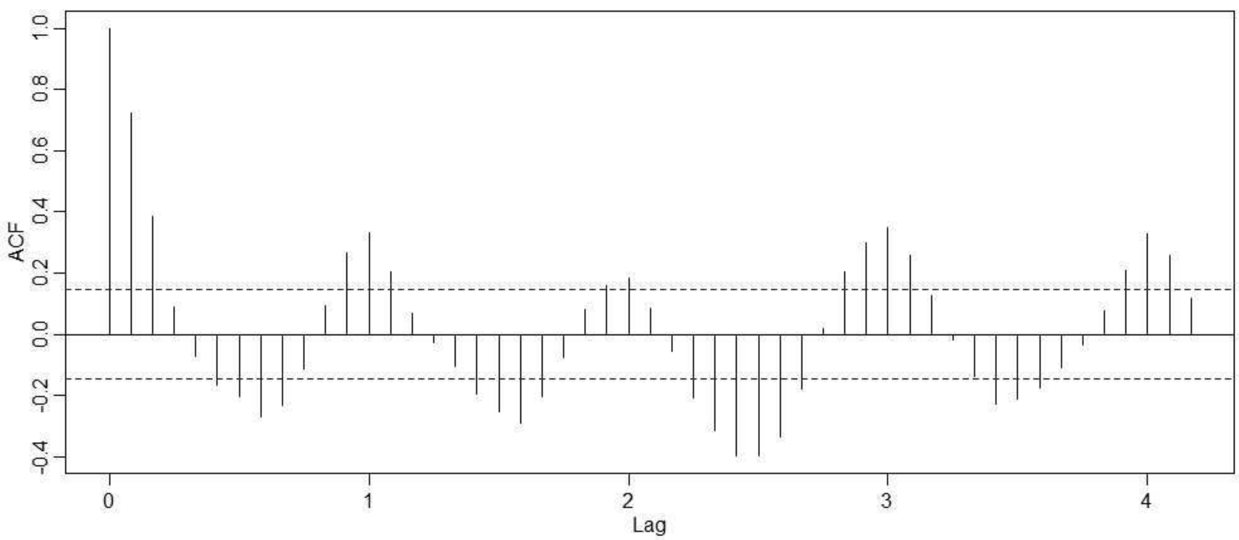
The chichen sales data shows a strong up trend with the variance remaining moderately the same across the series. There appears to be some seasonality in the data, with peaks and troughs more apparent in the beginning. There is a fairly smooth spike in the second half of the series and then some seasonality returns again, but less volatile. While the PACF for the data cuts off at lag 1, the ACF is slowly decaying, which suggests differencing.
Differencing the data, we see that the seasonality is a lot more pronounced. Plotting points every 6th and 12th time period, we see that there is some seasonality every 12th lag. Though there is some seasonality every 6th lag, the variance for this seems to be too great and is probably captured by s=12. Seasonally differencing the data with s=12 produces fairly similar results. The plotted seasonally differenced data look the same as without the seasonal differencing, though there are slight differences. If we seasonally difference again on s=12, we arrive at data that looks similar to that of only non-seasonally differencing once. Because of this, we will stick with the non-seasonally differenced data only.

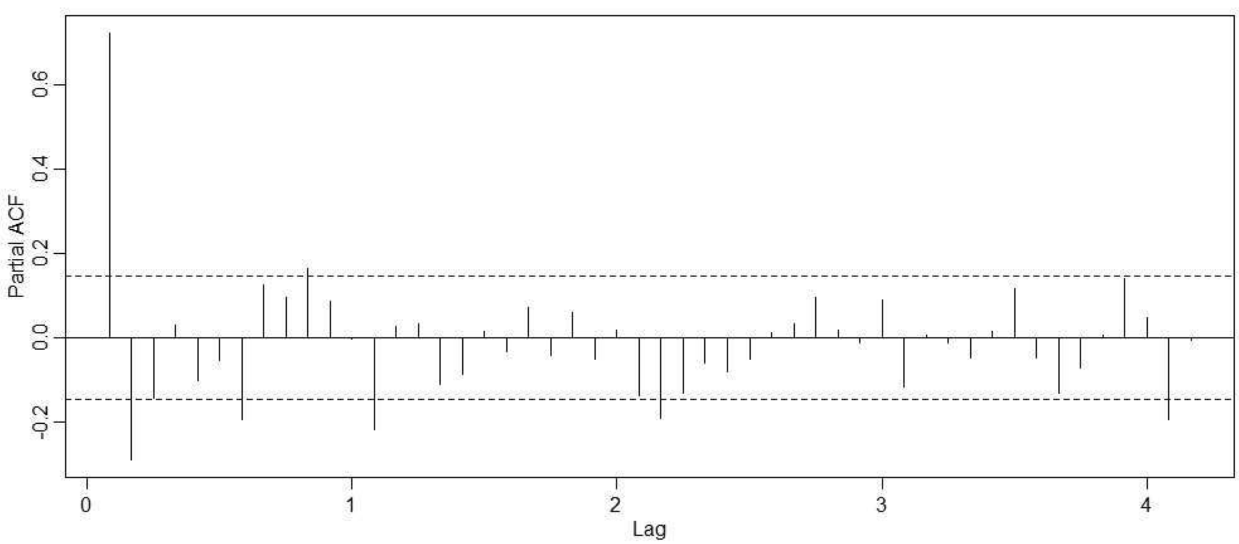



```
x = chicken
par(mfrow=c(2,1))
plot(x, main="Chicken Sales Data")
d = diff(x)
plot(d, main="Differenced Chicken Data")
l = length(d)
# Repeat this for all series
rep(NA, l) -> d6 -> d12 # do not use rep(0,...)
for (i in 1:l) {
  if (i %% 6 == 0) {
    d6[i] = d[i]
```

```
    if (i %% 12 == 0) {
      d12[i] = d[i]
     }
   }
}
fish = ts.intersect(d, d6, d12)
lines(fish[,2], type="p", col=4, pch=19)
lines(fish[,3], type="p", col=3, pch=19)
```

The ACF and PACF for the differenced transformed data are shown below. The PACF appears to cut off at lag 2 with obvious annual cycles in the ACF. The ACF could be said to be cutting off at lag 2 or tailing off. The PACF appears to be mostly cutting off at every annual lag (s=12). We will explore various models



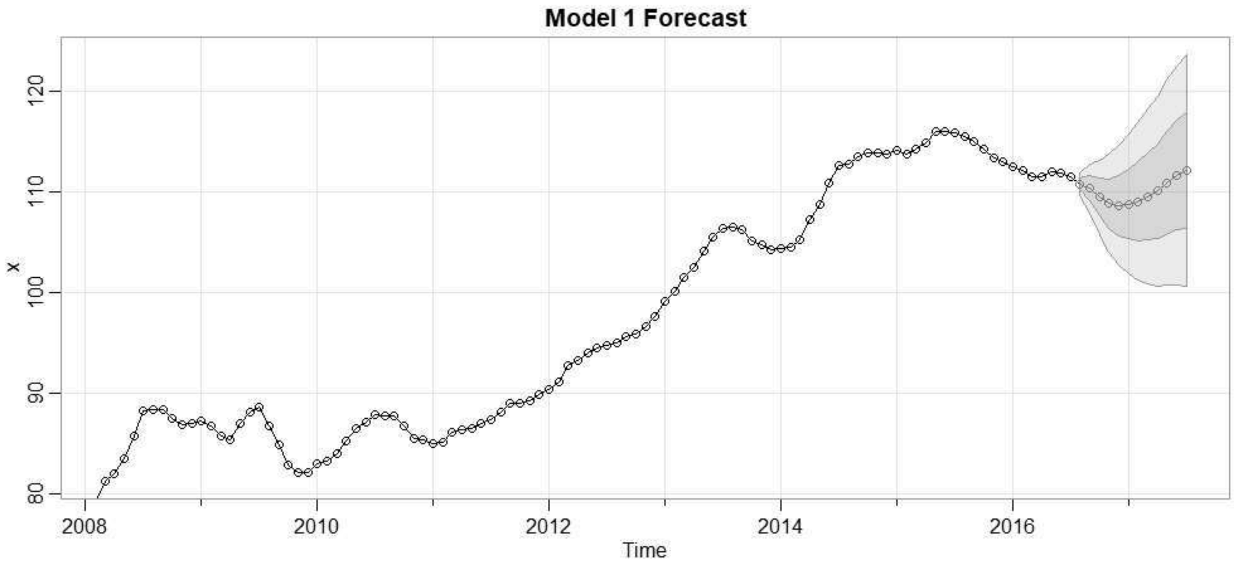


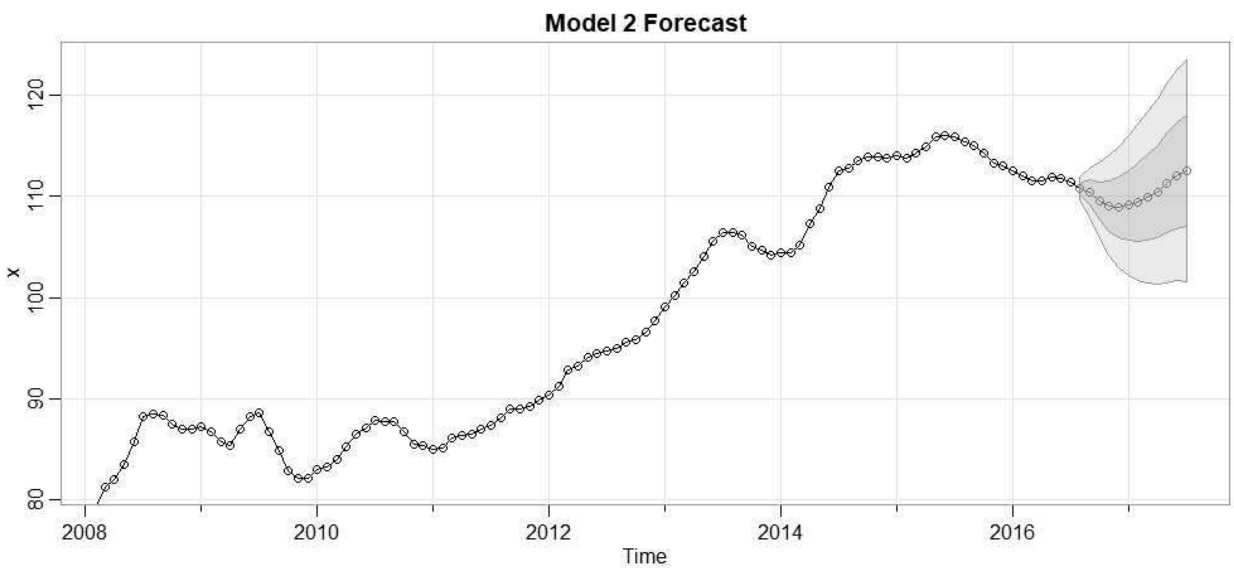
To run various SARIMA's with so many variable parameters, we are likely to run into optimisation errors in R, so we use a tryCatch as per the below.

```
d = diff(chicken)
rep(NA, 81) -> aic -> aicc -> bic -> pcol -> qcol -> Pcol -> Qcol
i = 1
for (p in 0:2) {
  for (q in 0:2) {
    for (P in 0:2) {
      for (Q in 0:2) {
        result = tryCatch({
            m = sarima(d, p,0,q,P,0,Q,12, details=FALSE, no.constant=TRUE)
            aic[i] = m$AIC; aicc[i] = m$AICc; bic[i] = m$BIC
        },
            warning = function(w) {
              print(w) # You don't need these to print
        },
            error = function(e) {
              print(e)
        },
            finally = {
              pcol[i] = p; qcol[i] = q; Pcol[i] = P; Qcol[i] = Q
              i = i + 1
})}}}}
cbind("p"=pcol,"q"=qcol,"P"=Pcol,"Q"=Qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
      p q P Q      AIC     AICc      BIC
 ...
[37,] 1 1 0 0 2.077662 2.078043 2.131082
[38,] 1 1 0 1 1.986134 1.986900 2.057360
[39,] 1 1 0 2 1.981523 1.982807 2.070556
[40,] 1 1 1 0 1.959080 1.959846 2.030306
[41,] 1 1 1 1 1.844985 1.846269 1.934018
[42,] 1 1 1 2 1.855860 1.857797 1.962699
...
[58,] 2 0 1 0 1.942276 1.943043 2.013503
[59,] 2 0 1 1 1.836609 1.837894 1.925642
[60,] 2 0 1 2 1.847199 1.849137 1.954039
...
```

We can see that the SARIMA(1,1,1,1,0,1,12) and the SARIMA(2,1,0,1,0,1,12) are most favourable. The diagnostics on the SARIMA(1,1,1,1,0,1,12) for the data look reasonable.The QQ-plot has a few outliers, but all within the 95% confidence interval. The ACF shows no significant autocorrelation except for one lag. The Ljung-Box statistic does show some significance on a few lags and some lags are close to the significance level.

Slightly better results for the diagnostics of SARIMA(2,1,0,1,0,1,12), where the Ljung-Box statistic has insignificant p-values at all lags shown and the ACF does not show significant autocorrelation.This is the preferred model by the ICs.
We can see that the forecasts for each are fairly similar.

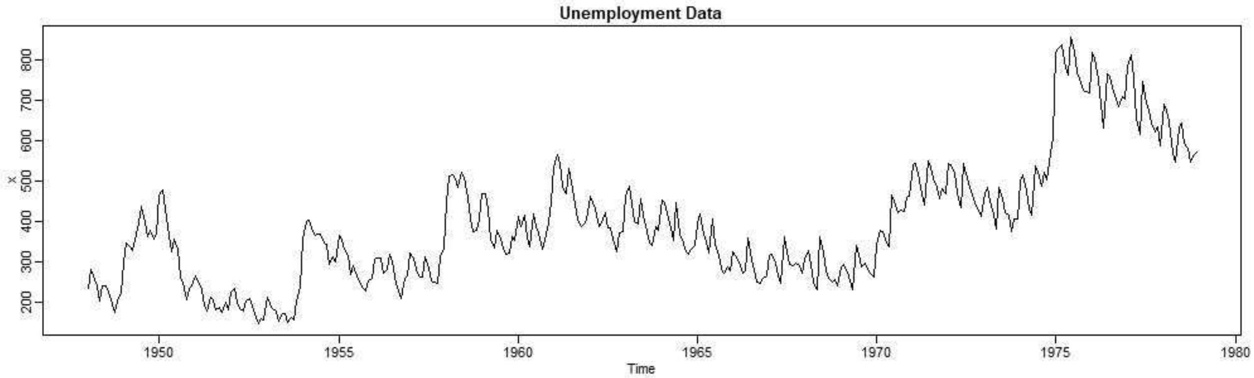


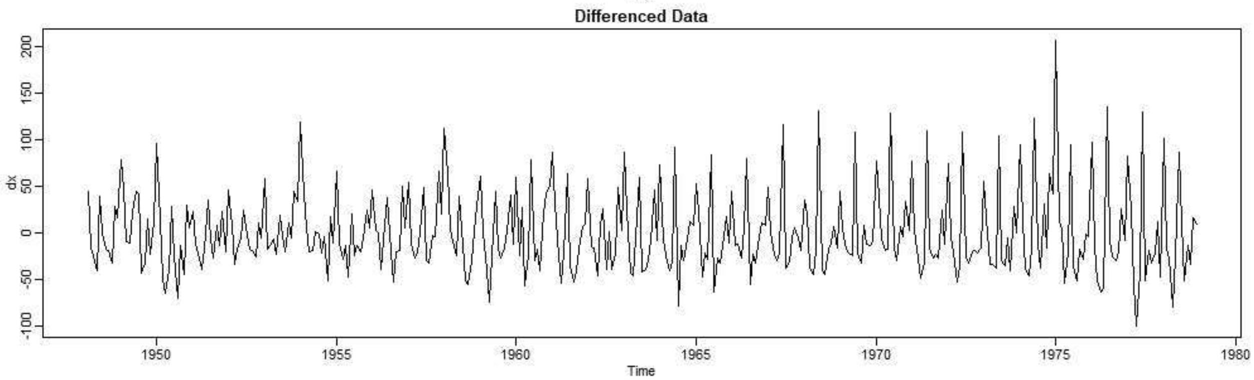


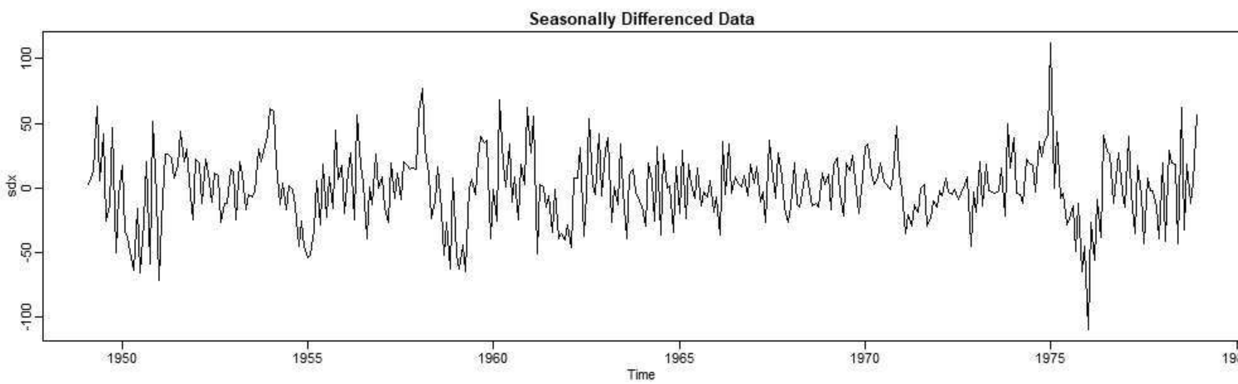
3.41

3.41 Fit a seasonal ARIMA model of your choice to the unemployment data in `unemp`. Use the estimated model to forecast the next 12 months.

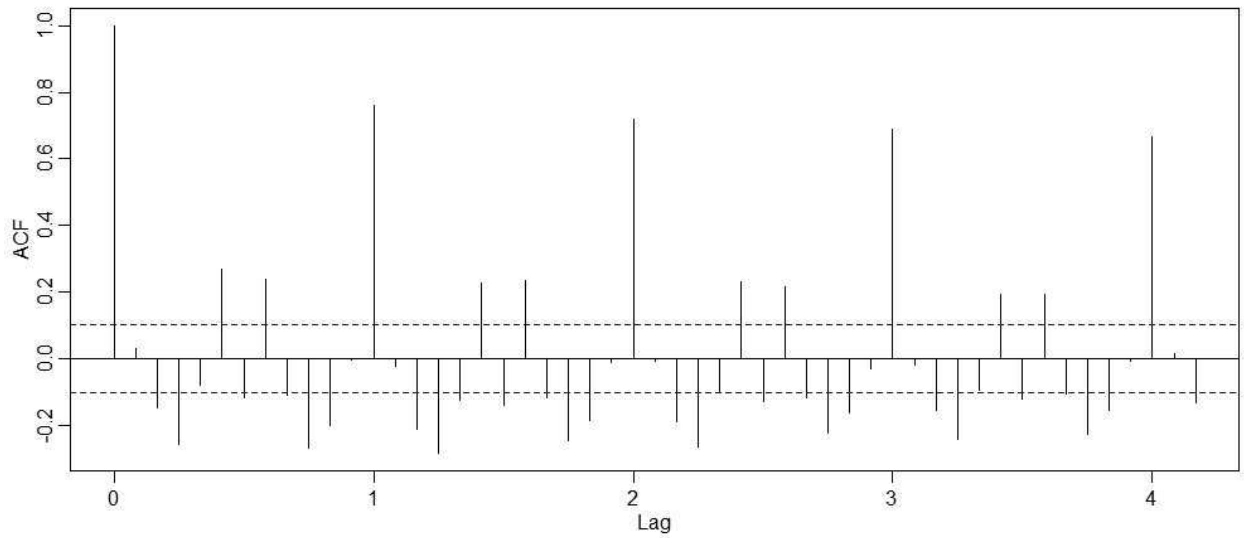
The data for unemp shows an obvious uptrend with some seasonality. Though there are some large spikes in the data, the variance remains relatively constant in between them. Differencing the data provides fairly stationary results. We can see that the mean is relatively constant and same for the variance, though we can see annual seasonality. Seasonally differencing the data, we see that likewise stationarity, though the seasonality is apparently gone. I don't see any benefit in seasonally differencing so we will keep the non-seasonally differenced data.

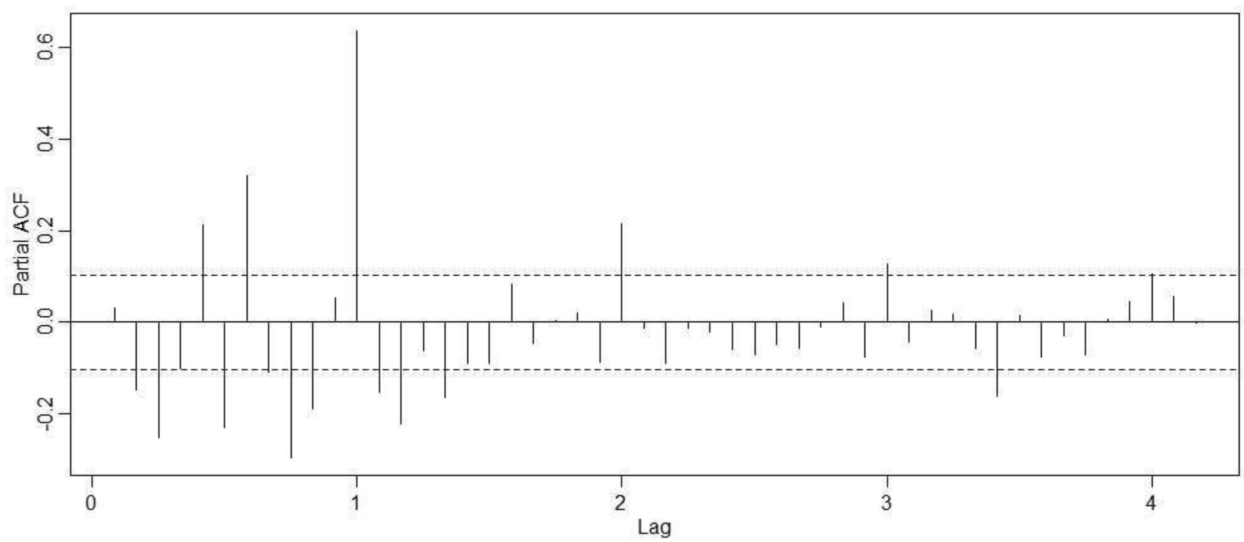






Looking at the ACF and PACF for the differenced data, we see that there is annual seasonality in the ACF with it cutting off every 12th lag. It appears that the PACF may be cutting off at the 3rd lag, while the ACF is tailing off. We will look at various models.





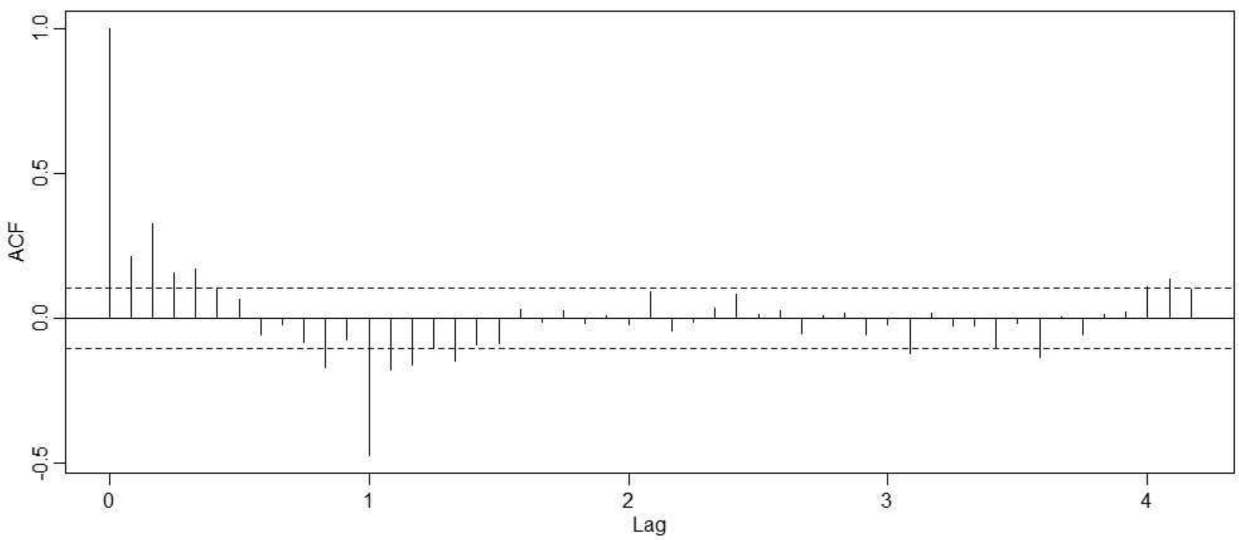
The most favourable models are SARIMA(0,1,0,2,0,1,12) and SARIMA(3,1,2,2,0,1,12).

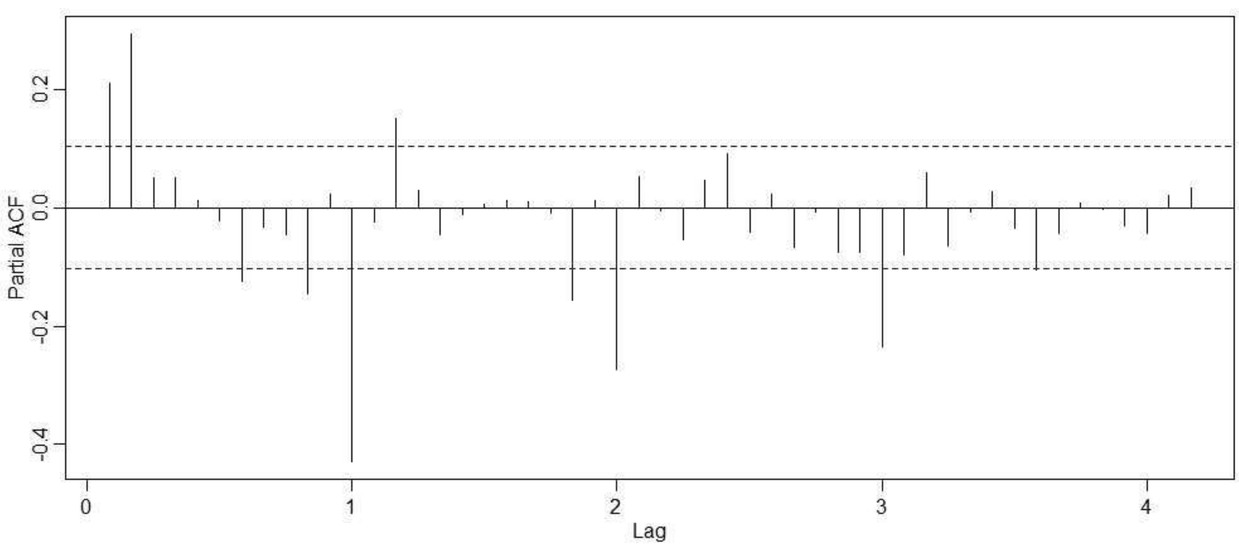
```
cbind("p"=pcol,"q"=qcol,"P"=Pcol,"Q"=Qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
       p q P Q       AIC      AICc       BIC
  [1,] 0 0 0 0 10.366806 10.366806 10.377362
...
  [8,] 0 0 2 1  9.114253  9.114429  9.156476
  [9,] 0 0 2 2        NA        NA        NA
...
[131,] 3 2 1 1        NA        NA        NA
[132,] 3 2 1 2        NA        NA        NA
[133,] 3 2 2 0        NA        NA        NA
[134,] 3 2 2 1  9.054082  9.055154  9.149084
...
```

The diagnostics for the SARIMA(0,1,0,2,0,1,12) show relatively high levels of autocorrelation. None of the p-values for the Ljung-Box statistic are insignificant and the ACF of the residuals show significant autocorrelation for a number of lags. Though the QQ-plot is mostly normal, with one or two outliers, the high autocorrelation can't be ignored. We have better results on the SARIMA(3,1,2,2,0,1,12), where there is no significant autocorrelation on the ACF or the Ljung-Box statistic. The QQ-plot traces normality slightly worse than the first model, however, nothing goes beyond the 95% confidence level and the outliers are reasonably contained. The parameters for ar3, ma1 and sar2 are insignificant, however, trying to remove these results in a non-stationary model and cannot be computed.

Though we said we wouldn't seasonally difference, we are going to investigate the various models anyway.

The ACF and PACF for the seasonally transformed data are shown below. We can see that the ACF appears to tails off at the 12th lag while the PACF cuts off respectively. The PACF also cuts off after lag 2, while the ACF tails.

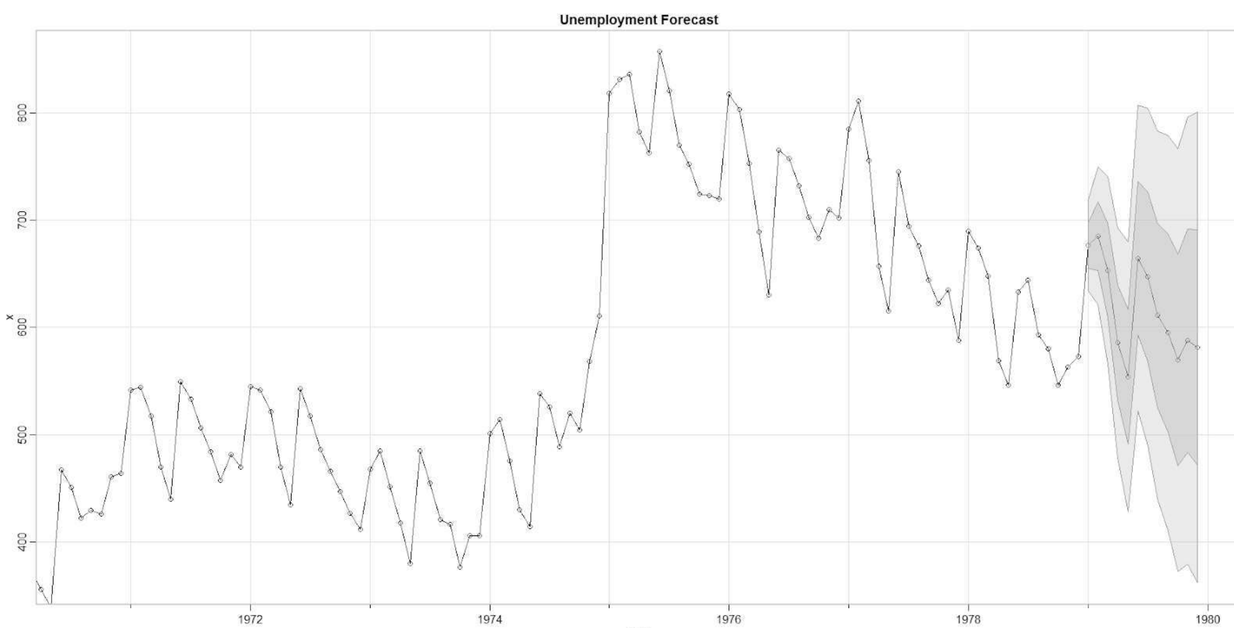




Below, we can see that the preferred models are SARIMA(2,1,0,0,1,1,12) and SARIMA(2,1,0,0,1,2,12), both of which are preferred relative to the models without seasonal differencing.

```
       p q P Q AIC      AICc      BIC
  [1,] 0 0 0 0 9.519671 9.519671 9.530488
  ...
 [74,] 2 0 0 1 8.991114 8.991303 9.034383
 [75,] 2 0 0 2 8.993053 8.993368 9.047139
 [76,] 2 0 1 0 9.133005 9.133193 9.176273
 [77,] 2 0 1 1 8.993700 8.994015 9.047786
...
[109,] 3 0 0 0 9.397528 9.397716 9.440796
[110,] 3 0 0 1 8.996647 8.996962 9.050733
[111,] 3 0 0 2 8.998610 8.999084 9.063513
...
```

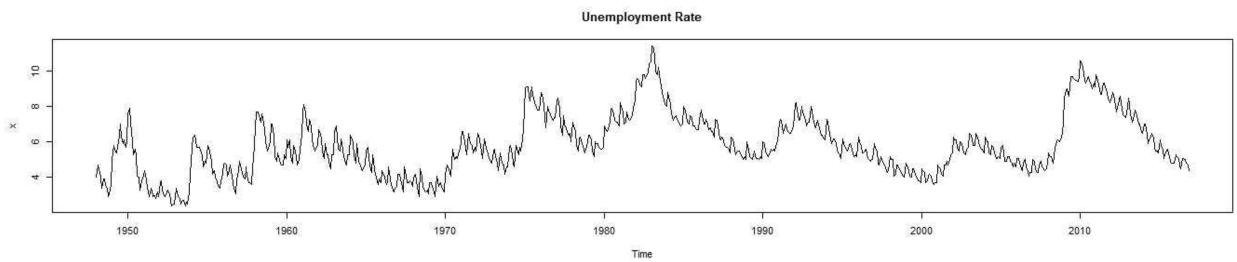
The diagnostics for both models are fairly similar. The p-values of the Ljung-Box statistic of the lags shown are all insignificant and much further away from significance than the non-seasonally differenced models. The ACF for both models show low autocorrelation and the QQ-plot traces normality very well, though the first model traces it slightly better.
The forecast is shown below.

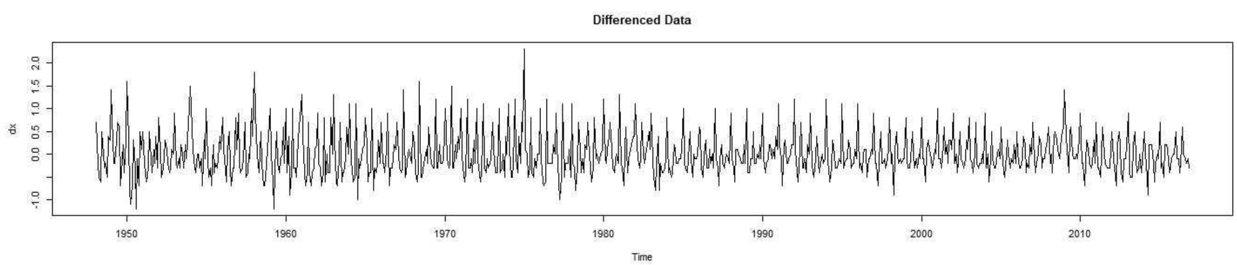


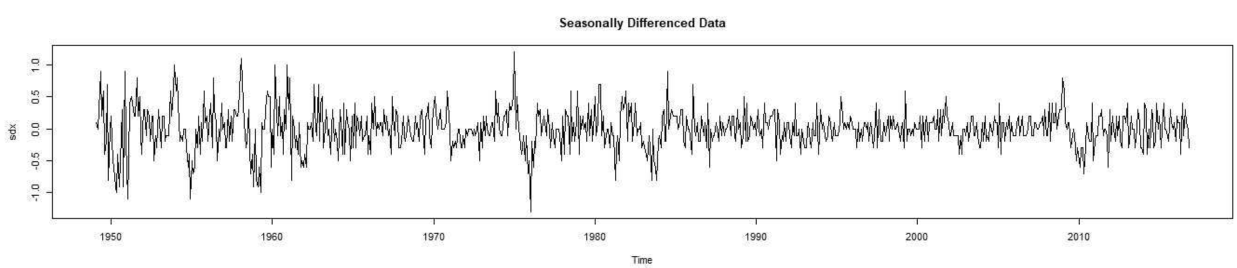
3.42

3.42 Fit a seasonal ARIMA model of your choice to the unemployment data in `UnempRate`. Use the estimated model to forecast the next 12 months.

Looking at the data, we can see that the mean various across the series and the volatility is nonconstant. Differencing the data, we have stationarity, however, there is an apparent annual cycle, so we seasonally difference (s=12). The ACF and PACF of the seasonally differenced data suggest SAR(1), as the PACF is cutting off every 12th lag and the ACF tails off respectively. We will run various seasonal models on the seasonally differenced data.







Running various SARIMA(0,1,0,P,1,Q,12) models we find that the SARIMA(0,1,0,0,0,1,12) is the most preferred by the ICs. Looking at the residuals of these fit, the ACF and PACF suggest an ARMA of possibly of order 4 or 5.
Running various SARIMA models we find that the most preferred is SARIMA(2,1,1,0,1,1,12). The diagnostics show that the autocorrelation is very low, however, there are quite a few outliers beyond the 95% mark on the QQ-plot.
Trying the model selection a different way, we look at the SARIMAs by not breaking it up into two parts. We see that the most favourable models are SARIMA(2,1,1,0,1,1,12), SARIMA(2,1,3,1,1,1,12), SARIMA(3,1,0,0,1,1,12) and SARIMA(3,1,3,2,1,1,12).

```
cbind("p"=pcol,"q"=qcol,"P"=Pcol,"Q"=Qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
       p q P Q            AIC           AICc          BIC
  [1,] 0 0 0 0  0.5261338694  0.5261338694 0.531910233
```

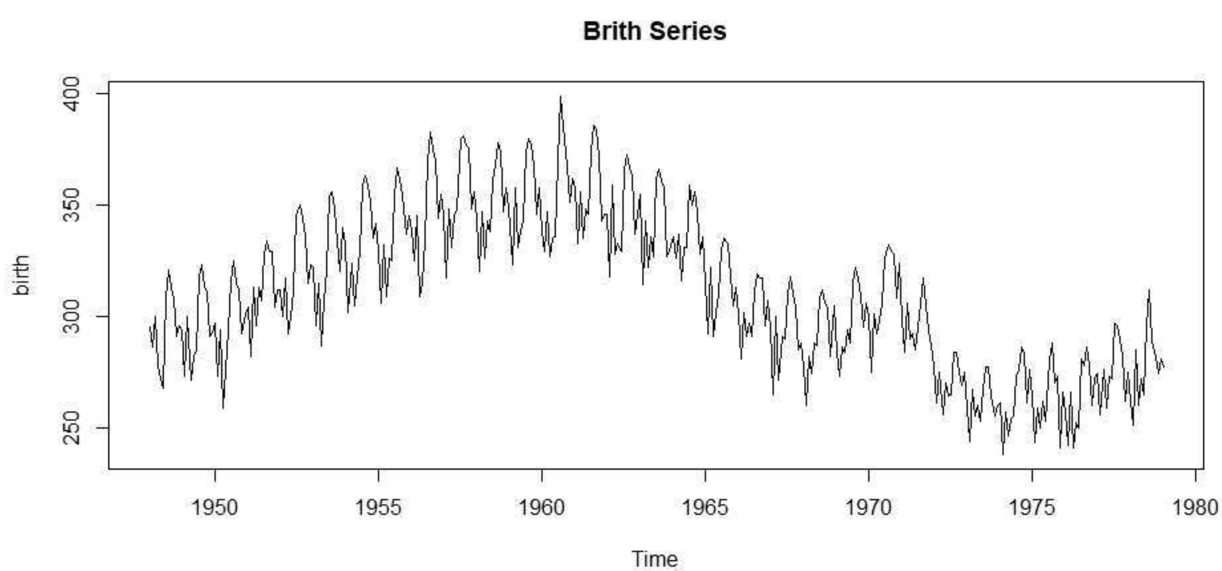
```
...
 [82,] 2 1 0 0  0.4200699476  0.4201063477 0.443175404
 [83,] 2 1 0 1 -0.0262616769 -0.0262009352 0.002620143
 [84,] 2 1 0 2 -0.0238058335 -0.0237146082 0.010852351
...
[103,] 2 3 1 0  0.1188813290  0.1190092027 0.159315878
[104,] 2 3 1 1 -0.0378625470 -0.0376918372 0.008348366
...
[110,] 3 0 0 1 -0.0222109872 -0.0221502455 0.006670833
...
[143,] 3 3 2 1 -0.03444201   -0.03416697   0.02332163
```

Although the AIC and AICc prefer SARIMA(2,1,3,1,1,1,12), SARIMA(2,1,1,0,1,1,12) has the best diagnostics (and is preferred by the BIC). However, all models show considerable outliers on the QQ-plot.

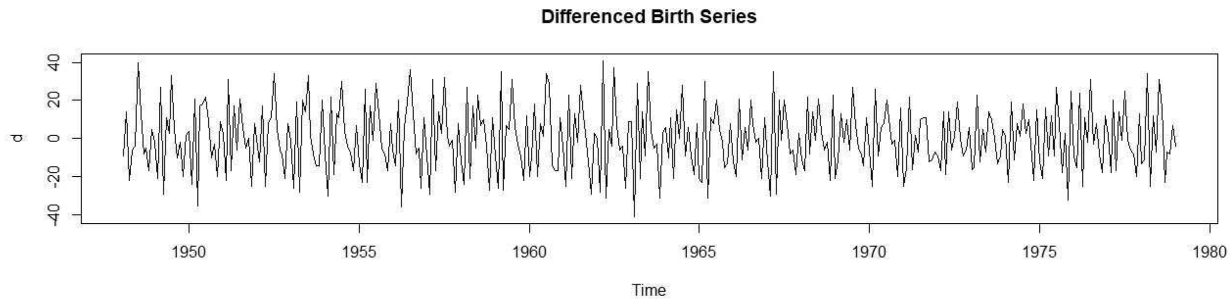
3.43

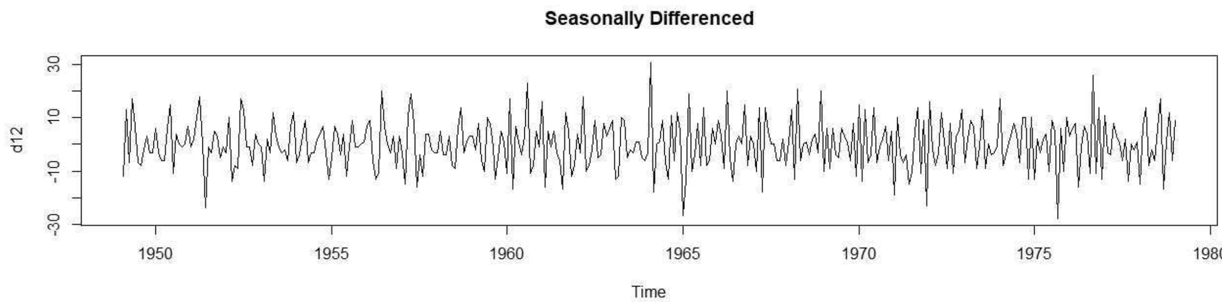
3.43 Fit a seasonal ARIMA model of your choice to the U.S. Live Birth Series (`birth`). Use the estimated model to forecast the next 12 months.

Plotting the data, we see that the mean could be constant over the long term, and the variance remains relatively stable, though we see a strong seasonal component. Over the short term the mean is non-constant.

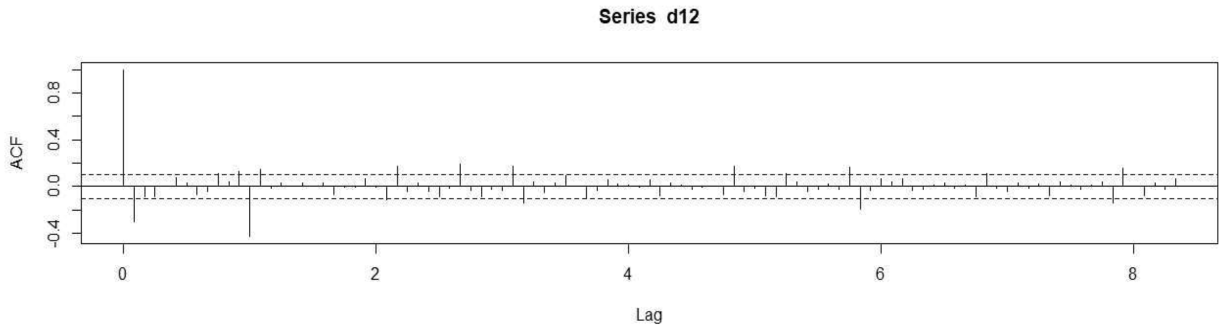


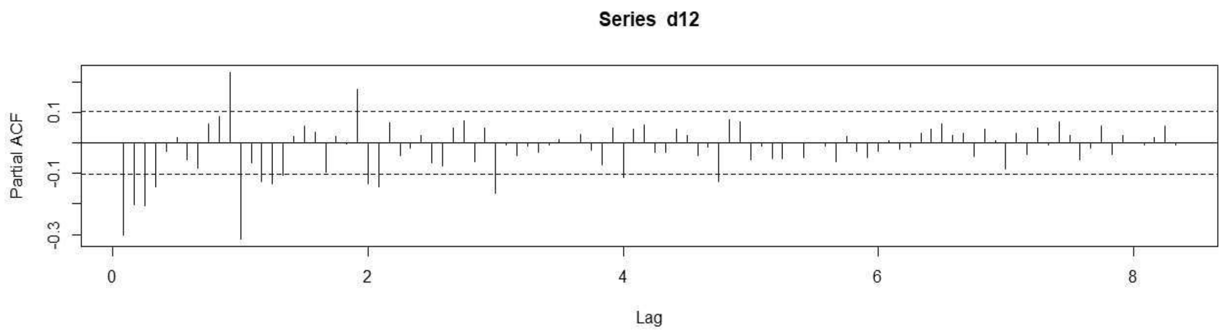
Looking at the ACF and PACF (not shown) of the untransformed data, we see that the ACF is decaying slowly over time, though there is clear seasonality. The PACF has a spike at lag 12, with some significant lags between 0 and 12, and another significant lag around h=12.





Differencing the data, then seasonally differencing it, we see that the series is fairly stationary. The ACF of the differenced data is no longer decaying and we see significant spikes every 12 lags. There are consistently significant lags between the 12s and 12(s+1) lags, with not indication of a slow decay. The PACF is still roughly similar to the untransformed data PACF. The ACF of the seasonally differenced data is showing seasonality, though with less significant peaks, but now in between the peaks there are no significant lags. The PACF is also showing seasonality. For the non-seasonal components, we see that the ACF might be cutting off at the first lag, and the PACF is either tailing off slowly, or possibly cutting off after the 2nd or 3rd lag.






```
cbind("p"=pcol,"q"=qcol,"P"=Pcol,"Q"=Qcol,"AIC"=aic,"AICc"=aicc,"BIC"=bic)
      p q P Q      AIC     AICc      BIC
...
 [81,] 1 1 0 0 7.100564 7.100658 7.132949
 [82,] 1 1 0 1 6.721819 6.722006 6.764998
 ...
 [38,] 2 3 0 2 6.718052 6.718936 6.804410
 [39,] 2 3 0 3 6.721474 6.722614 6.818627
 [40,] 2 3 1 0 6.862252 6.862913 6.937815
 [41,] 2 3 1 1 6.717267 6.718150 6.803624
...
```

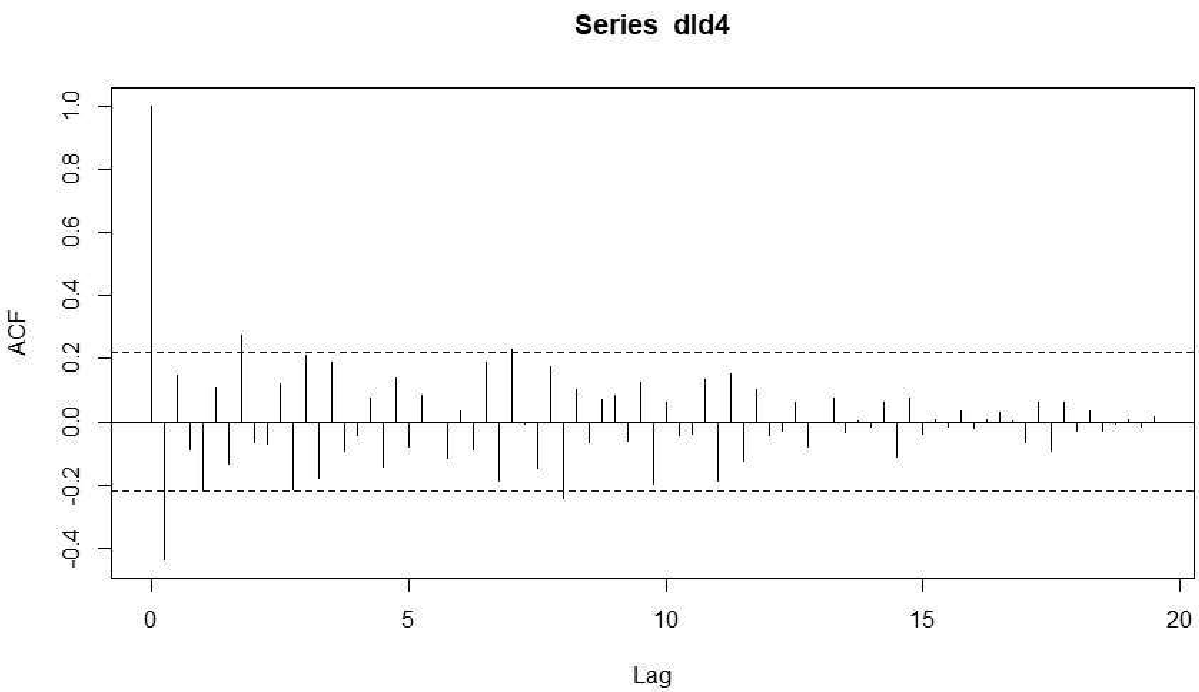
Both the AIC and AICc prefer the SARIMA(2,1,3,1,1,1,12), while the BIC prefers the SARIMA(1,1,1,0,1,1,12). The first model has fairly stationary residuals and generally good diagnostics. The QQ plot does not have any significant outliers, as all data points are within the 95% interval, though a few do diverge near the end. The LB statistic shows insignificance for all points, with only the first few being somewhat close to significance. The second model does not have as good diagnostics, as more data points are closer to significance on the LB and one even crosses the line. The QQ plot shows all data points within 95%, however, there is more divergence and earlier too. The residuals do look stationary however.

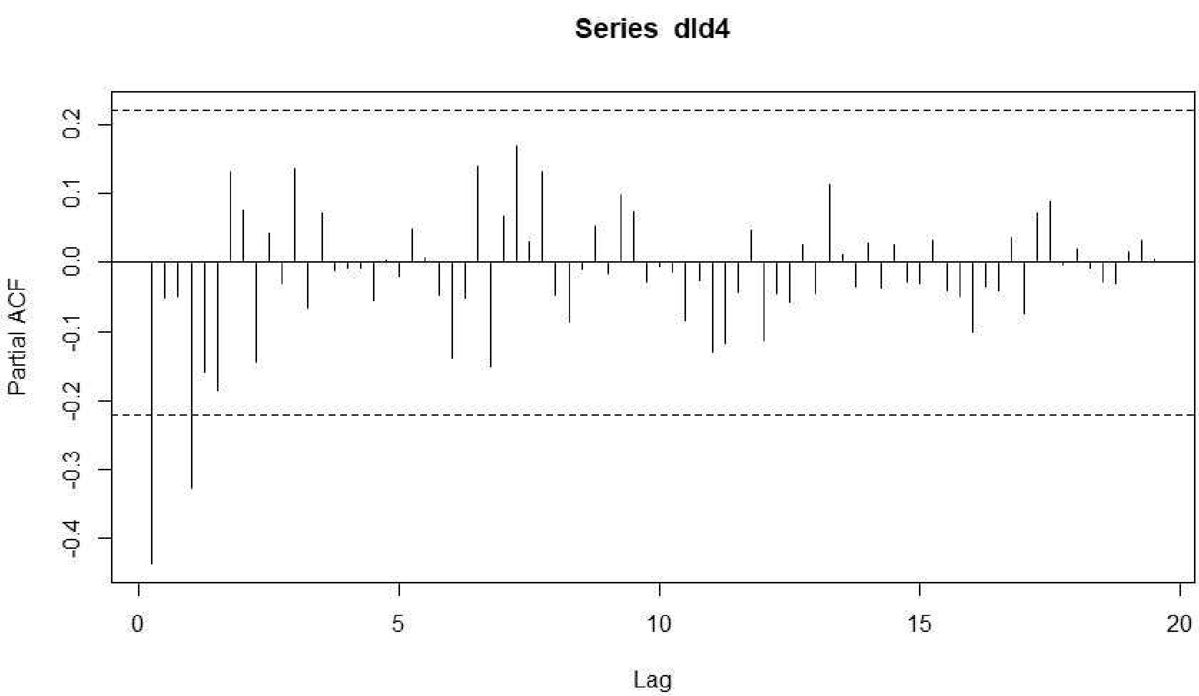
3.44

3.44 Fit an appropriate seasonal ARIMA model to the log-transformed Johnson and Johnson earnings series (`jj`) of Example 1.1. Use the estimated model to forecast the next 4 quarters.

Plotting the untransformed series, we see that there is an increasing mean and increasing variance. If we plot the differenced data, we see a constant mean but still an increasing variance. We can't take the log of this differenced data, because there are negative values. However, if we take the differenced of the logged data, then we see relative stationarity. However, we can still see some seasonality, as the data appears to be highly correlated at every fourth lag. The seasonally differenced series appears the most stationary, as the seasonality as pretty much gone.

The ACF of the untransformed data is slowly decaying over time, even going into the negative values before trailing off to 0. This indicates high correlation in the series. The PACF seems less defined and may be considered tailing off or cutting off after a few lags. The ACF of the logged data looks essentially the same. The PACF is showing less correlation in the first few lags, however, the fourth lag is significant. The ACF of the dl data is now more dispersed, with the sign alternating between positive and negative at every lag. We still see a slow decay and some seasonality at every fourth lag. The PACF of the dl appears to have some seasonality and is possibly cutting off after the fourth lag. The ACF of the seasonally differenced data, dld4, is showing the least significant correlation, though there may be some seasonality. It appears to cut off after the first lag. The PACF is showing more seasonality, but nothing significant though appears to cut off after the fourth lag. We can't have the ACF and PACF both cutting off, so I would be more inclined to say that the PACF is trailing off.






```
bind
       p q P Q       AIC      AICc       BIC
...
[25,] 0 0 4 4       NA        NA         NA
[26,] 0 1 0 0 -1.841950 -1.841292 -1.781964
```

```
[27,] 0 1 0 1 -1.821422 -1.819424 -1.731443
...
[125,] 1 0 1 0 -1.758952 -1.756953 -1.668973
[126,] 1 0 1 1 -1.760579 -1.756529 -1.640607
...
[36,] 2 3 0 0 -1.899923 -1.889519 -1.719965
[37,] 2 3 0 1 -1.878871 -1.864103 -1.668920
[38,] 2 3 0 2 -1.854256 -1.834288 -1.614312
```

The AIC and AICc both prefer the SARIMA(2,1,3,0,1,0,4) model, while the BIC prefers the SARIMA(0,1,1,0,1,0,4). I have also highlighted the SARIMA(1,1,0,1,1,0,4) because the solutions give this model. For the first model, we see fairly stationary residuals and no significant lags in the AC. The LB statistics is close to significance on the first two values, however, it is otherwise insignificant. The QQ plot fits well, with all values within the 95% range and only a small amount of divergence. The second model has a better QQ plot, however, the LB statistic shows consistent significance. The model given by the solutions does better than the second model, however, it diverges more from the QQ plot and has p-value closer to significance on the LB statistic. The forecasts of each model are fairly similar.