STAT 2321
Orly Olbum

Chapter 4 HW

4.2 $\{w_t; t=0,1,\dots\}$ is white noise with σ_w^2.
$|\phi|<1$ is a constant
$x_0 = w_0$

$$x_t = \phi x_{t-1} + w_t, \quad t=1,2,\dots \qquad (AR(1))$$

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

for $t=1$: $x_1 = \sum_{j=0}^{1} \phi^j w_{t-j} = \phi^0 w_1 + \phi^1 w_0 = \phi x_0 + w_1$

from x_t above, $x_t = \phi x_{t-1} + w_t$

and $x_1 = \phi x_0 + w_1$, so the equation holds.

Assuming it holds true for $t-1$, we can then say that

$$\sum_{j=0}^{t-1} \phi^j w_{t-j} + [\phi x_0 + w_1] = \sum_{j=0}^{t} \phi^j w_{t-j}$$

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

ⓑ Find $E(x_t)$

$$E(x_t) = E(\phi x_{t-1} + w_t) = \phi E(x_{t-1}) + \underbrace{E(w_t)}_{0}$$
$$= \phi E(x_{t-1}) = 0$$

ⓒ Show that for $t=0,1,\dots$

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

we know $var(w_t) = var(w_{t-1}) = \dots = \sigma_w^2$

And, for $|\phi|>1$, $var(x_t) = \frac{\sigma_w^2}{1-\phi^2}$

$\sum_{j=0}^{k} \phi^k = (1-\phi^{k+1})/(1-\phi)$

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

$$var(x_t) = var\left[\sum_{j=0}^{\infty} \phi^j w_{t-j}\right] = \left[\sum_{j=0}^{\infty} \phi^j\right]^2 \underbrace{var(w_{t-j})}_{\sigma_w^2}$$
$$= \sigma_w^2 \cdot \left[\sum_{j=0}^{\infty} \phi^{2j}\right] = \sigma_w^2 \cdot \left[(1-\phi^{2(t+1)})/(1-\phi^2)\right]$$
$$= \frac{\sigma_w^2}{1-\phi^2}(1-\phi^{2(t+1)})$$

ⓓ Show that for $h \ge 0$,

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

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

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

→

4.2, continued

ⓔ Is x_t stationary?

No. The variance of x_t depends on time t, and since $\gamma(s,t)$ of x_t depends on t and not only on the time difference, x_t is <u>not stationary</u>.

ⓕ As $t \to \infty$, x_t becomes stationary. (x_t is "asymptotically stationary")

we saw that

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

Since $|\phi|<1$, as $t \to \infty$, the term $(1-\phi^{2(t+1)})$ will approach $(1-0)=1$, making

$var(x_t) \to \frac{\sigma_w^2}{1-\phi^2}$, and since it no longer depends on time t, x_t becomes stationary as $t \to \infty$ ("asymptotically" stationary).

ⓖ If we generated n observations of x_t, we could get rid of the beginning observations (say it's "warming up"!) and we have a Gaussian AR(1) model, and since $|\phi|<1$, $\overset{iid}{\sim} N(0,1)$.

ⓗ Suppose $x_0 = \frac{w_0}{\sqrt{1-\phi^2}}$ > is this stationary?

For $x_t = \phi x_{t-1} + w_t$

at $t=1$: $x_1 = \phi x_0 + w_1 = \phi \frac{w_0}{\sqrt{1-\phi^2}} + w_1$

at $t=2$: $x_2 = \phi x_1 + w_2 = \phi^2 \frac{w_0}{\sqrt{1-\phi^2}} + \phi w_1 + w_2$

which gives us:

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

$$E(x_t) = E\left[\frac{\phi^t w_0}{\sqrt{1-\phi^2}} + \sum_{j=0}^{t-1}\phi^j w_{t-j}\right] = \frac{\phi^t}{\sqrt{1-\phi^2}} \overset{\to 0}{E(w_0)} + \sum_{j=0}^{t-1}\phi^j \overset{\to 0}{E(w_{t-j})} = 0$$

$$var(x_t) = cov\left(\frac{\phi^t w_0}{\sqrt{1-\phi^2}} + \sum_{j=0}^{t-1}\phi^j w_{t-j}, \frac{\phi^t w_0}{\sqrt{1-\phi^2}} + \sum_{j=0}^{t-1}\phi^j w_{t-j}\right)$$

$$= cov\left(\frac{\phi^t w_0}{\sqrt{1+\phi^2}}, \frac{\phi^t w_0}{\sqrt{1-\phi^2}}\right) + cov\left(\frac{\phi^t w_0}{\sqrt{1-\phi^2}}, \sum_{j=0}^{t-1}\phi^j w_{t-j}\right) + cov\left(\sum_{j=0}^{t-1}\phi^j w_{t-j}, \frac{\phi^t w_0}{\sqrt{1-\phi^2}}\right)$$

$$+ cov\left(\sum_{j=0}^{t-1}\phi^j w_{t-j}, \sum_{k=0}^{t-1}\phi^k w_{t-k}\right)$$

(see next page) →

4.2 (h), continued

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

Since $E(x_t) = 0$ (constant) and the variance $var(x_t)$ does not depend on time t, the process <u>is</u> stationary

4.3

Consider: (i) xt = .80xt-1 - .15xt-2 + wt - .30wt-1 (ii) xt = xt-1 - .50xt-2 + wt - wt-1

(a) Using Example 4.10 as a guide, check the models for parameter redundancy. If a model has redundancy, find the reduced form of the model.

First, determine the roots of each model. If there are common factors, the model can be reduced.

```
# (i)
ari = c(1, -.8, .15)
mai = c(1, -.3)
Mod(polyroot(ari))
```

```
## [1] 2.000000 3.333333
```

```
Mod(polyroot(mai))
```

```
## [1] 3.333333
```

We see that for (i), there is a common factor and the model can be reduced. The reduced model is xt = .5xt-1 + wt (which is AR(1)) (see below for calculation).

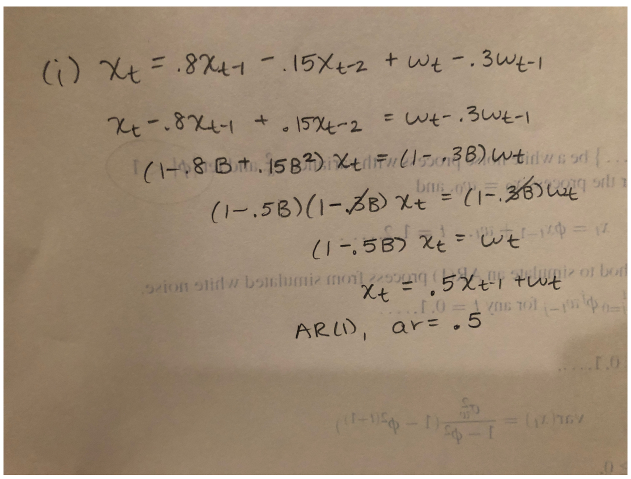


Figure 1: Reduced Model (i)

```
# (ii)
arii = c(1, -1, .5)
maii = c(1, -1)
Mod(polyroot(arii))
```

```
## [1] 1.414214 1.414214
```

```
Mod(polyroot(maii))
```

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

For (ii), there are no common factors so the model is in its reduced form (no redundancies).

(b) A way to tell if an ARMA model is causal is to examine the roots of AR term phi(B) to see if there are no roots less than or equal to one in magnitude. Likewise, to determine invertibility of a model, the roots of the MA term theta(B) must not be less than or equal to one in magnitude. Use Example 4.11 as a guide to determine if the reduced (if appropriate) models (i) and (ii), are causal and/or invertible.

```r
# reduced model (i)
ari_red = .5
mai_red = 0

Mod(polyroot(ari_red))
```

```
## numeric(0)
```

```r
Mod(polyroot(mai_red))
```

```
## numeric(0)
```

```r
# (ii)
Mod(polyroot(arii))
```

```
## [1] 1.414214 1.414214
```

```r
Mod(polyroot(maii))
```

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

For reduced model (i), roots are zero so the model is both causal and invertible. For model ii, both roots of phi(B) are greater than one so it is not causal, but the root of theta(B) is equal to 1, so it is invertible.

(c) In Example 4.3 and Example 4.12, we used ARMAtoMA and ARMAtoAR to exhibit some of the coefficients of the causal [MA(inf)] and invertible [AR(inf)] representations of a model. If the model is in fact causal or invertible, the coefficients must converge to zero fast. For each of the reduced (if appropriate) models (i) and (ii), find the first 50 coefficients and comment.

```r
# reduced model (i)
round(ARMAtoMA(ar = .5, ma = 0, 50), 3)
```

```
##  [1] 0.500 0.250 0.125 0.062 0.031 0.016 0.008 0.004 0.002 0.001 0.000 0.000
## [13] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [25] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [37] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [49] 0.000 0.000
```

```r
round(ARMAtoAR(ar = .5, ma = 0, 50), 3)
```

```
##  [1] -0.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
## [16]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
## [31]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
## [46]  0.0  0.0  0.0  0.0  0.0
```

```r
# (ii)
round(ARMAtoMA(ar = c(1, -.5), ma = -1, 50), 3)
```

```
##  [1]  0.000 -0.500 -0.500 -0.250  0.000  0.125  0.125  0.062  0.000 -0.031
## [11] -0.031 -0.016  0.000  0.008  0.008  0.004  0.000 -0.002 -0.002 -0.001
## [21]  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## [31]  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## [41]  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
```

```r
round(ARMAtoAR(ar = c(1, -.5), ma = -1, 50), 3)
```

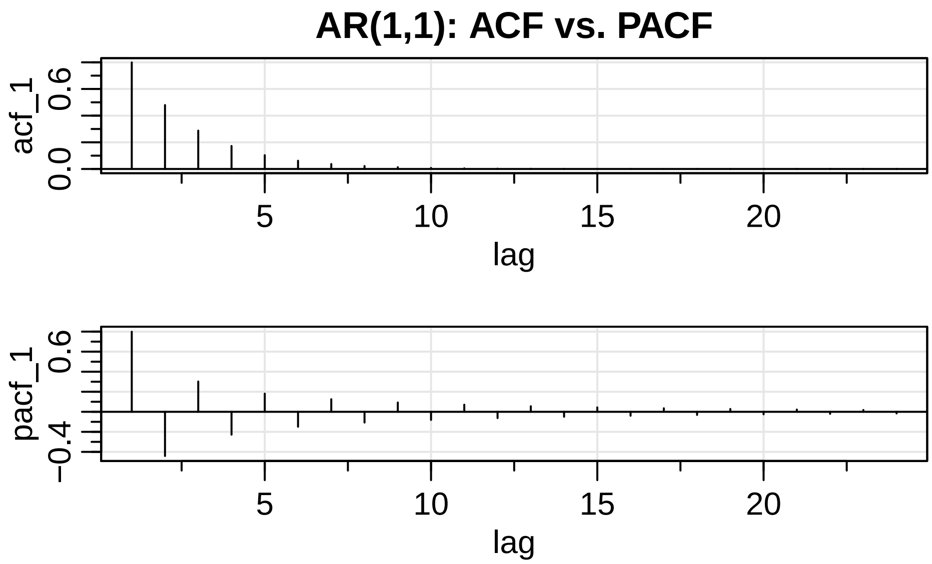
```
##  [1] 0.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
## [20] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
## [39] 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
```

Model (i) is causal and invertible, because from the coefficients produced we can see both the ARMAtoMA and ARMAtoAR go to zero. Model (ii) is not, since the ARMAtoAR coefficients do not go to zero and instead repeat .5.

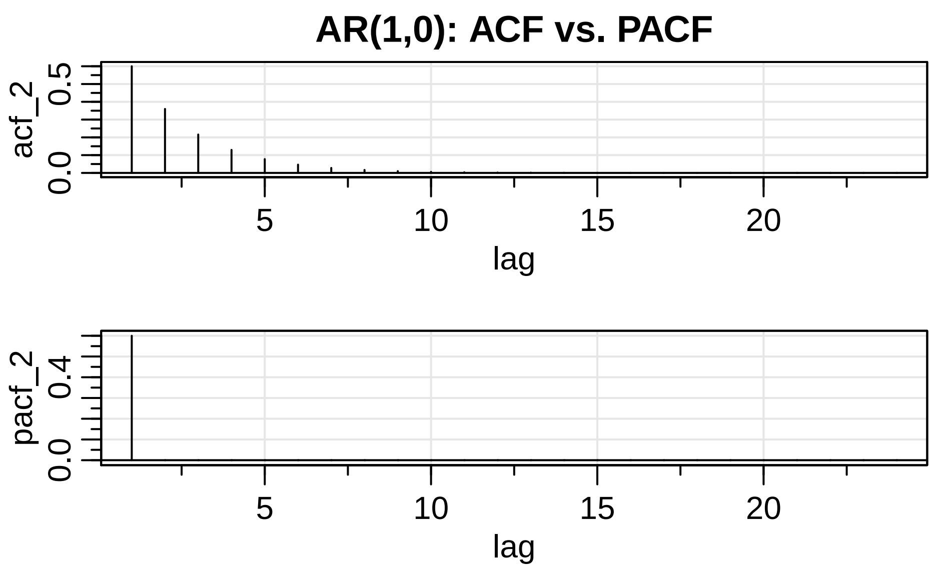
4.4

(a) Compare the theoretical ACF and PACF of an ARMA(1, 1), an ARMA(1, 0), and an ARMA(0, 1) series by plotting the ACFs and PACFs of the three series for phi = .6, theta = .9. Comment on the capability of the ACF and PACF to determine the order of the models. Hint: See the code for Example 4.18.

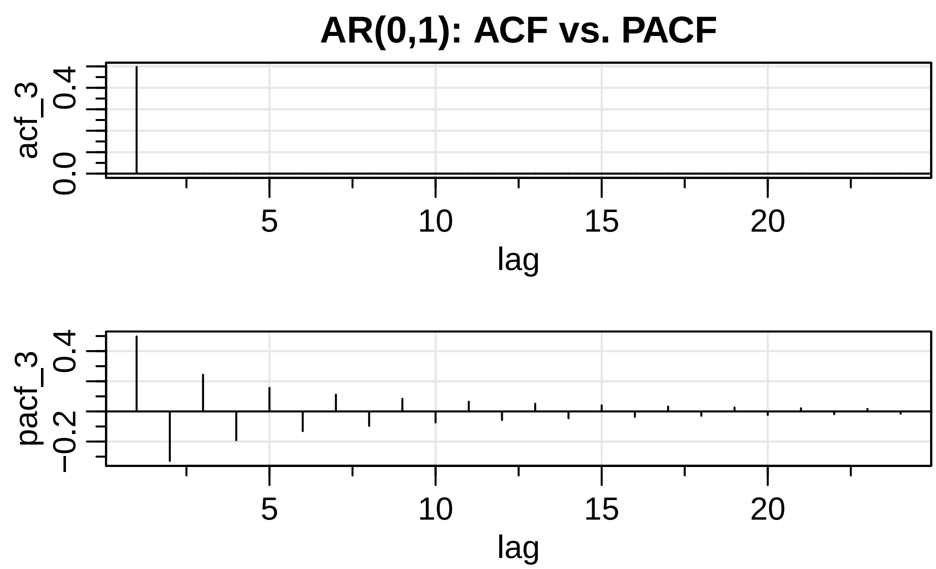
```r
acf_1 = ARMAacf(ar = .6, ma = .9, 24)[-1]
pacf_1 = ARMAacf(ar = .6, ma = .9, 24, pacf = TRUE)
acf_2 = ARMAacf(ar = .6, ma = 0, 24)[-1]
pacf_2 = ARMAacf(ar = .6, ma = 0, 24, pacf = TRUE)
acf_3 = ARMAacf(ar = 0, ma = .9, 24)[-1]
pacf_3 = ARMAacf(ar = 0, ma = .9, 24, pacf = TRUE)

par(mfrow = 2:1)
tsplot(acf_1, type = "h", xlab = "lag", main = "AR(1,1): ACF vs. PACF")
abline(h = 0)
tsplot(pacf_1, type = "h", xlab = "lag")
abline(h = 0)
```




```r
tsplot(acf_2, type = "h", xlab = "lag", main = "AR(1,0): ACF vs. PACF")
abline(h = 0)
tsplot(pacf_2, type = "h", xlab = "lag")
abline(h = 0)
```



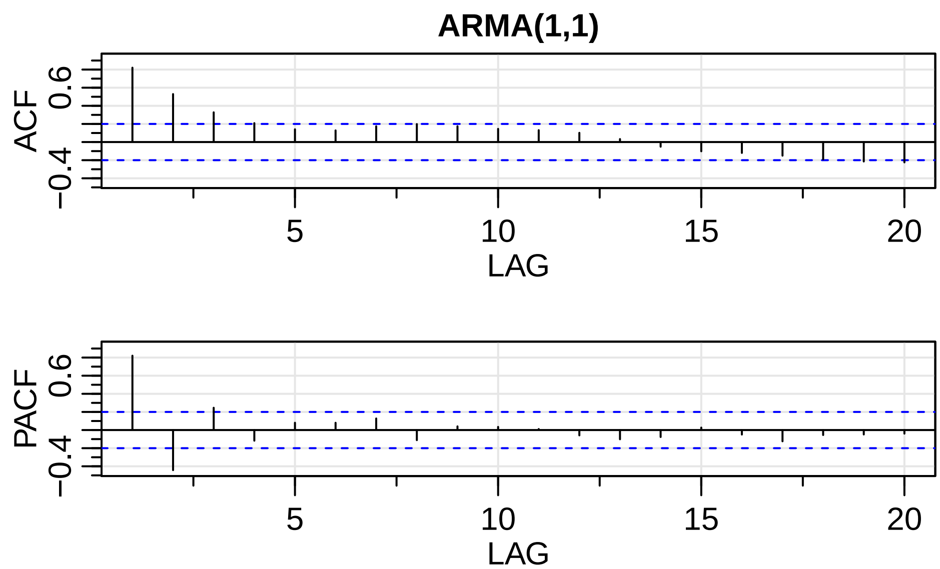

```r
tsplot(acf_3, type = "h", xlab = "lag", main = "AR(0,1): ACF vs. PACF")
abline(h = 0)
tsplot(pacf_3, type = "h", xlab = "lag")
abline(h = 0)
```



The ACF and PACF plots are very telling - for ARMA(1,1), we see that both the ACF and PACF taper off somewhat together, while for AR(1) and MA(1) the PACF and ACF, respectively, abruptly end at the specified lag. More specifically, for ARMA(1,0) = AR(1), the PACF exists at lag 1 and the correlation is 0 from there on out. For ARMA(0,1) = MA(1), the ACF this time drops to 0 after lag 1.

(b) Use arima.sim to generate n = 100 observations from each of the three models discussed in (a). Compute the sample ACFs and PACFs for each model and compare it to the theoretical values. How do the results compare with the general results given in Table 4.1?

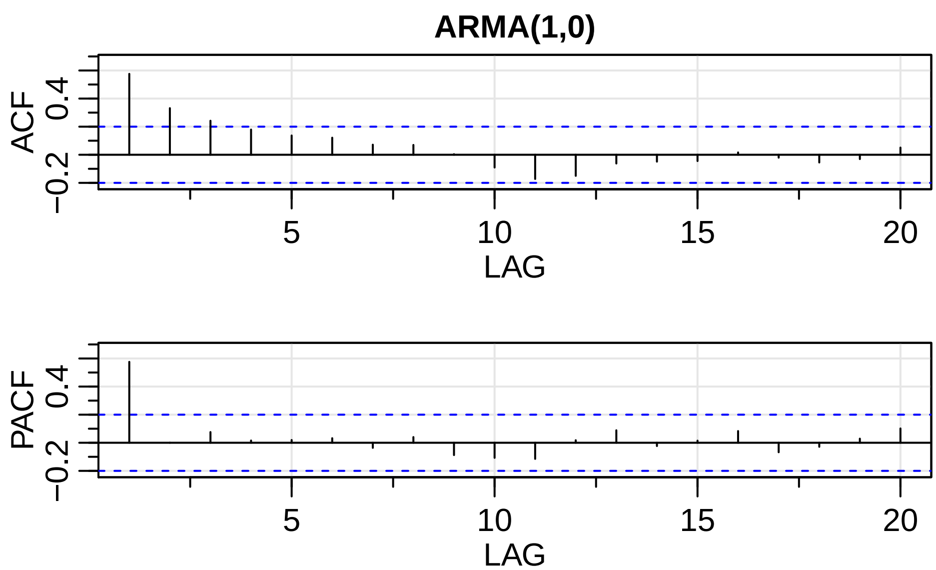
```r
arma1 = arima.sim(list(order = c(1, 0, 1), ar = .6, ma = .9), n = 100)
acf2(arma1, main = "ARMA(1,1)")
```




```
##      [,1]  [,2] [,3]  [,4] [,5] [,6] [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.82  0.53 0.33  0.21 0.14 0.13 0.17  0.20 0.17  0.15  0.13  0.10  0.03
## PACF 0.82 -0.44 0.25 -0.12 0.08 0.08 0.13 -0.11 0.04  0.03  0.01 -0.06 -0.10
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20]
```

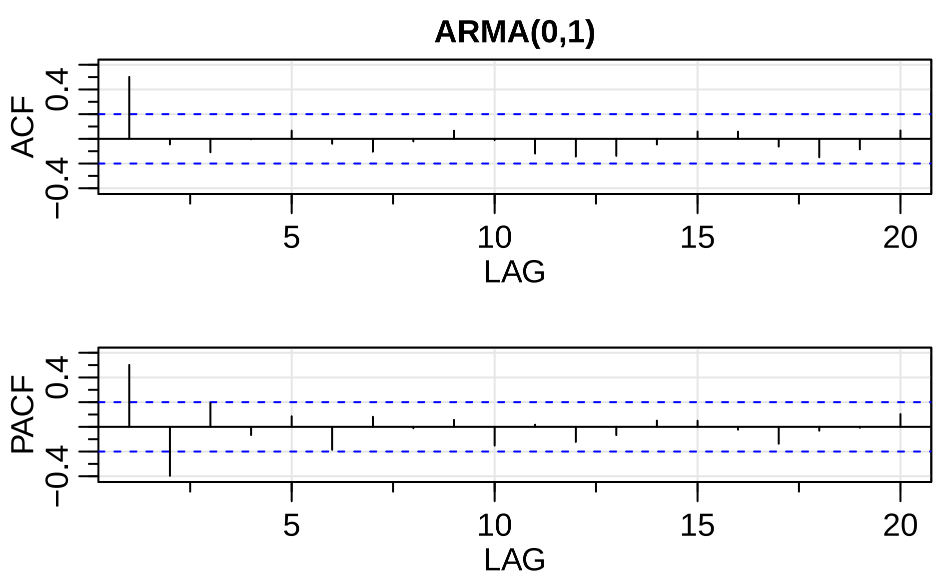
```
## ACF  -0.05 -0.10 -0.12 -0.15 -0.19 -0.21 -0.22
## PACF -0.08  0.03 -0.05 -0.12 -0.05 -0.05 -0.04
```

```r
arma2 = arima.sim(list(order = c(1, 0, 0), ar = .6), n = 100)
acf2(arma2, main = "ARMA(1,0)")
```




```
##      [,1] [,2] [,3] [,4] [,5] [,6]  [,7] [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.58 0.33 0.24 0.18 0.14 0.12  0.07 0.07  0.00 -0.09 -0.17 -0.15 -0.06
## PACF 0.58 0.00 0.08 0.02 0.02 0.03 -0.04 0.04 -0.09 -0.11 -0.11  0.02  0.09
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## ACF  -0.05 -0.04  0.02 -0.02 -0.06 -0.03  0.05
## PACF -0.02  0.02  0.08 -0.07 -0.03  0.03  0.10
```

```r
arma3 = arima.sim(list(order = c(0, 0, 1), ma = .9), n = 100)
acf2(arma3, main = "ARMA(0,1)")
```

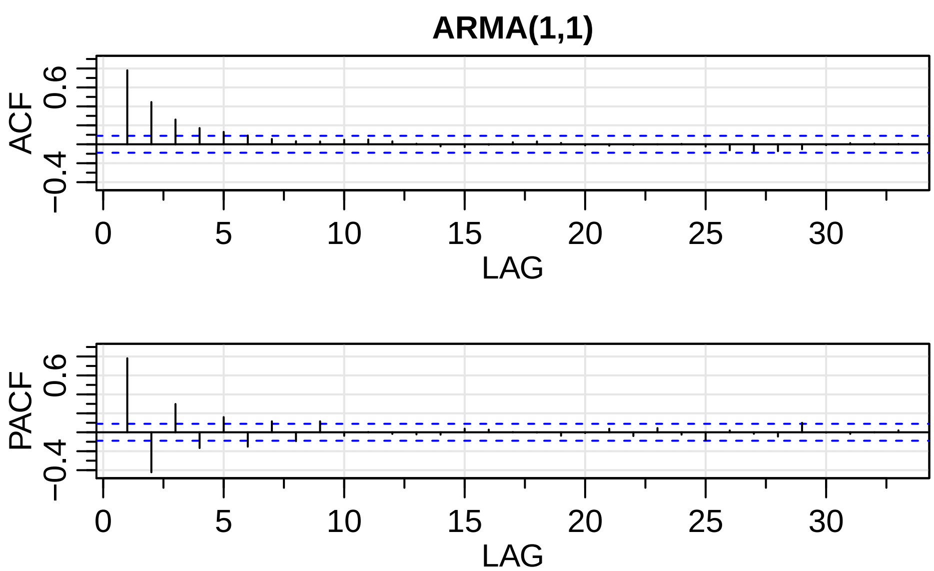


```
##       [,1]  [,2]  [,3]  [,4] [,5]  [,6]  [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF   0.5 -0.05 -0.11  0.00 0.07 -0.04 -0.10 -0.02 0.07 -0.01 -0.12 -0.14 -0.14
## PACF  0.5 -0.40  0.19 -0.07 0.09 -0.19  0.08 -0.01 0.06 -0.15  0.02 -0.12 -0.07
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## ACF  -0.05  0.06  0.06 -0.06 -0.15 -0.08  0.07
## PACF  0.05  0.05 -0.02 -0.14 -0.03 -0.01  0.10
```

The simulated models above track with what we did in (a), save for a bit of noise. For ARMA(1,1), we see that both ACF and PACF slowly trail off together. AR(1) has a steadily decreasing ACF and an abrupt tail off for PACF, but not exactly 0 like we saw above. MA(1) shows the same, with ACF dropping right after lag 1 (but some noise later on?) and a slower-tailing PACF, which for the most part is what we expect from Table 4.1.

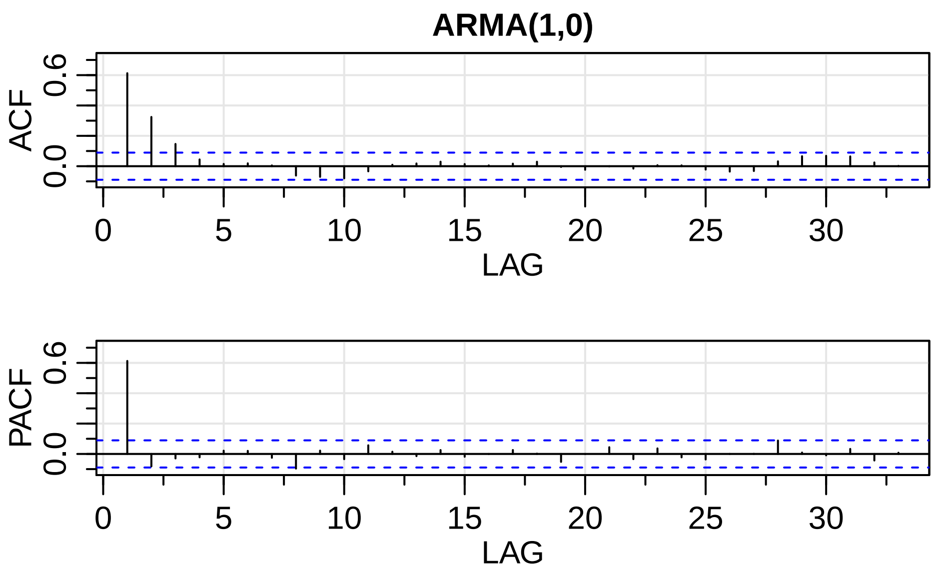
(c) Repeat (b) but with $n = 500$. Comment.

```r
arma1c = arima.sim(list(order = c(1, 0, 1), ar = .6, ma = .9), n = 500)
acf2(arma1c, main = "ARMA(1,1)")
```



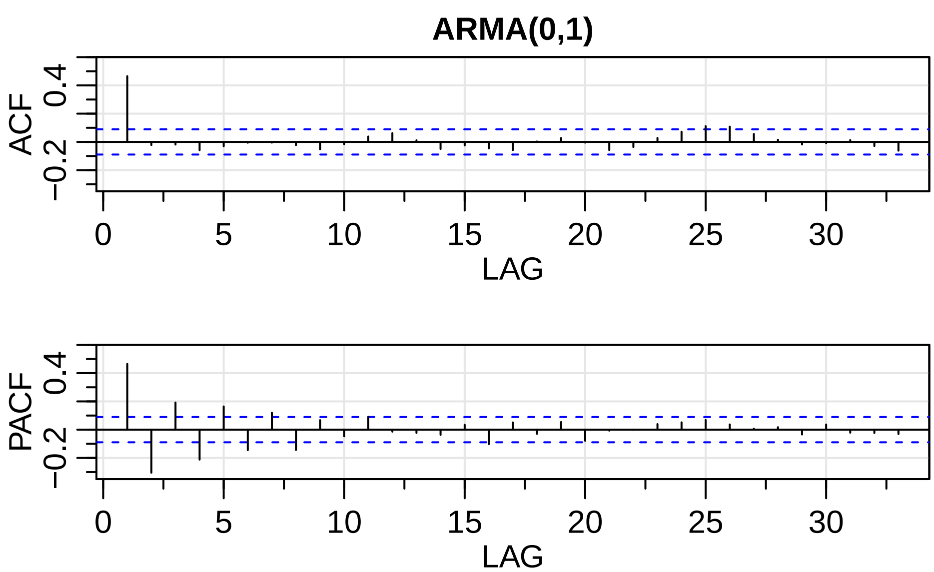

```
##      [,1]  [,2] [,3]  [,4] [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.78  0.45 0.26  0.17 0.13  0.09 0.06  0.03 0.03  0.05  0.05  0.03  0.01
## PACF 0.78 -0.42 0.30 -0.17 0.16 -0.15 0.12 -0.09 0.12 -0.04  0.00 -0.02 -0.02
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF  -0.02 -0.03 -0.01  0.02  0.03  0.01 -0.01 -0.01 -0.01  0.00  0.00 -0.02
## PACF -0.03  0.04  0.03  0.00  0.00 -0.04 -0.01  0.04 -0.04  0.04 -0.03 -0.08
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33]
## ACF  -0.06 -0.07 -0.07 -0.05 -0.01  0.01  0.01  0.00
## PACF  0.02 -0.02 -0.05  0.10  0.00 -0.02  0.00  0.02
```

```r
arma2c = arima.sim(list(order = c(1, 0, 0), ar = .6), n = 500)
acf2(arma2c, main = "ARMA(1,0)")
```




```
##       [,1]  [,2]  [,3]  [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.61  0.32  0.15  0.04 0.01 0.02  0.01 -0.06 -0.07 -0.08 -0.03  0.01  0.02
## PACF 0.61 -0.08 -0.03 -0.02 0.02 0.02 -0.03 -0.10  0.02 -0.03  0.06  0.01 -0.01
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.03  0.01  0.01  0.02  0.03 -0.01 -0.02  0.00 -0.02  0.01  0.01 -0.02
## PACF  0.03 -0.02  0.00  0.03  0.00 -0.05  0.00  0.04 -0.03  0.04 -0.02 -0.04
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33]
## ACF  -0.04 -0.03  0.03  0.07  0.07  0.06  0.02  0.00
## PACF  0.00  0.00  0.09  0.01 -0.01  0.03 -0.04  0.01
```

```
arma3c = arima.sim(list(order = c(0, 0, 1), ma = .9), n = 500)
acf2(arma3c, main = "ARMA(0,1)")
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
## ACF  0.47 -0.02 -0.02 -0.06 -0.03 -0.01 -0.01 -0.02 -0.05 -0.02  0.04  0.06
```

```
## PACF 0.47 -0.30  0.19 -0.21  0.16 -0.15  0.12 -0.14  0.07 -0.05  0.09 -0.01
##       [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF    0.01 -0.05 -0.02 -0.05 -0.06  0.00  0.03 -0.01 -0.06 -0.04  0.03  0.07
## PACF -0.02 -0.04  0.03 -0.10  0.05 -0.03  0.05 -0.08 -0.01  0.00  0.04  0.05
##       [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33]
## ACF    0.11  0.11  0.06  0.02 -0.02 -0.01  0.01 -0.03 -0.06
## PACF  0.07  0.04  0.01  0.02 -0.03  0.04 -0.02 -0.02 -0.03
```

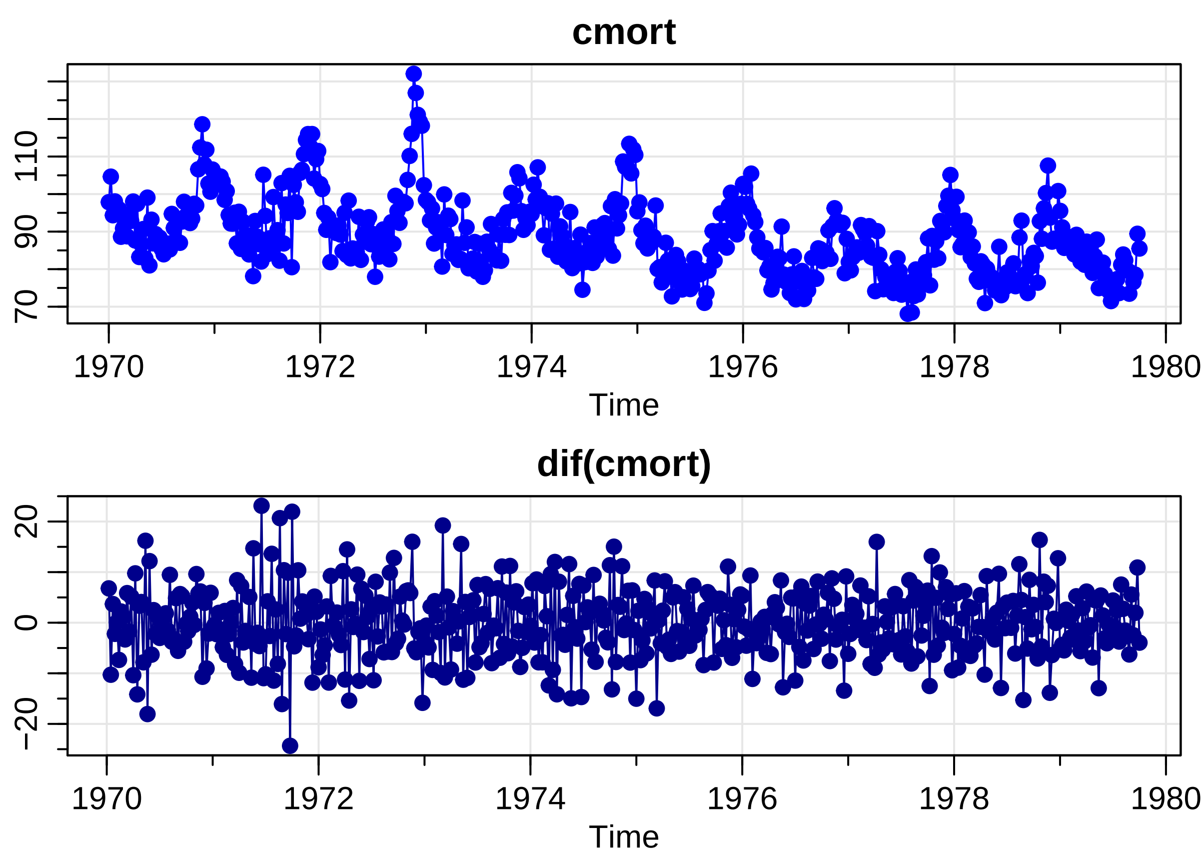
With 500 observations rather than 100, the ACF and PACF graphs of the simulated models more closely resemble what we did in (a), and what we expect to see from Table 4.1. ARMA(1,1) shows an even tail off from both ACF and PACF. The PACF for AR(1) and ACF for MA(1) show more abrupt drops to 0 after lag 1 than did the simulation with only 100 observations.

4.5

Let ct be the cardiovascular mortality series (cmort) discussed in Example 3.5 and let xt = dif(ct) be the differenced data.

(a) Plot xt and compare it to the actual data plotted in Figure 3.2. Why does differencing seem reasonable in this case?

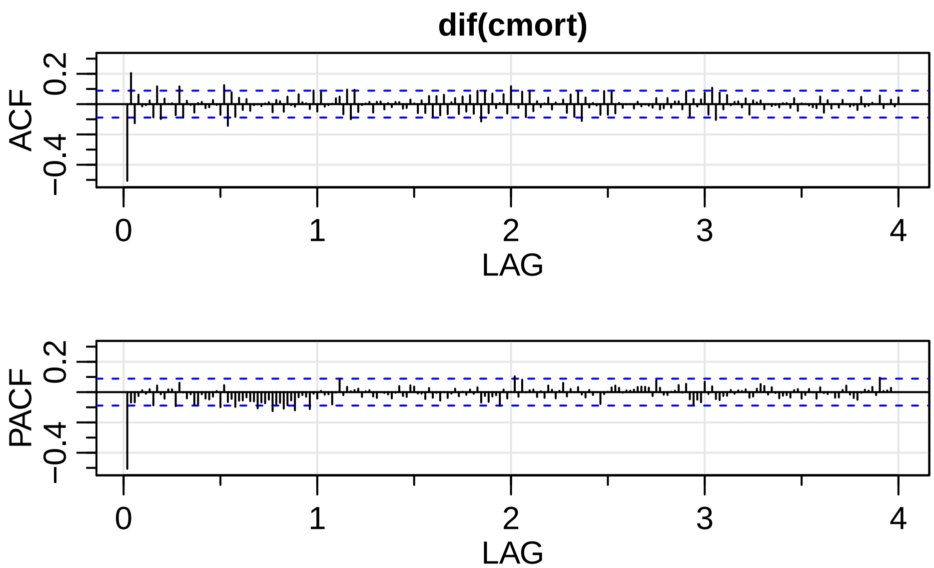
```
ct = cmort
xt = diff(ct)
par(mfrow = c(2, 1))
tsplot(ct, main = "cmort", col = 4, type = "o", pch = 19, ylab = "")
tsplot(xt, main = "dif(cmort)", col = "darkblue", type = "o", pch = 19, ylab = "")
```



Compared to the plotted cmort data, the differenced data looked much more stationary. In the nature of the data being a moving average, differencing will yield a stationary process when the original process is not stationary.

(b) Calculate and plot the sample ACF and PACF of xt and using Table 4.1, argue that an AR(1) is appropriate for xt.

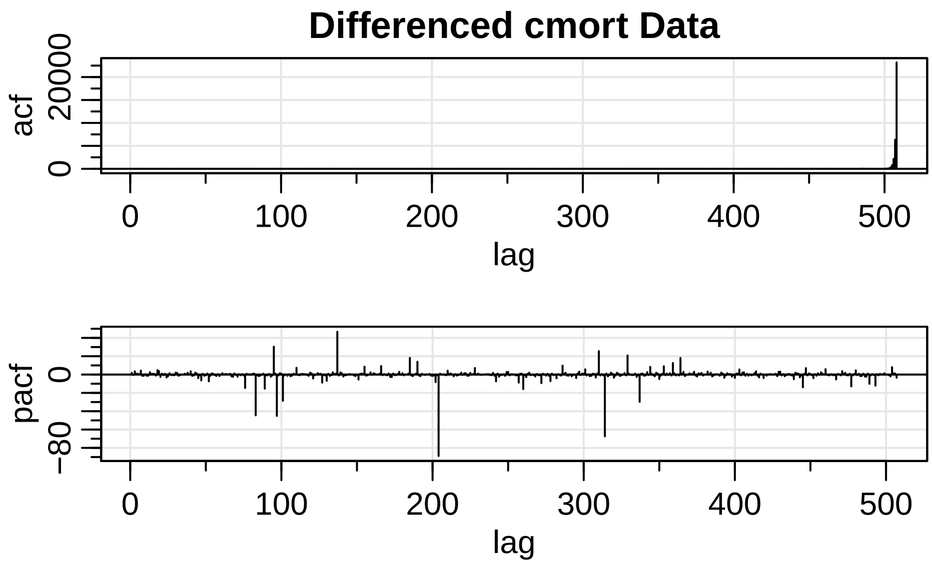
```
acf2(xt, main = "dif(cmort)")
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6] [,7]  [,8] [,9] [,10] [,11] [,12]
## ACF  -0.51  0.21 -0.13  0.06 -0.02 -0.01 0.02 -0.09 0.12 -0.10  0.04  0.00
## PACF -0.51 -0.07 -0.07 -0.03  0.01 -0.01 0.02 -0.09 0.04 -0.01 -0.04  0.02
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.01 -0.07  0.12 -0.09  0.02 -0.01 -0.06  0.01  0.01 -0.03 -0.02  0.03
## PACF  0.02 -0.09  0.06  0.00 -0.04 -0.01 -0.08 -0.08 -0.01 -0.04 -0.05 -0.03
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF  -0.01 -0.07  0.13 -0.14  0.08 -0.08  0.04 -0.04  0.03 -0.04 -0.01  0.00
## PACF  0.01 -0.10  0.05 -0.07 -0.04 -0.10 -0.06 -0.05 -0.03 -0.06 -0.06 -0.11
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF  -0.01  0.00  0.01 -0.05  0.03  0.02 -0.05  0.05 -0.01 -0.02  0.06  0.01
## PACF -0.07 -0.07 -0.05 -0.13 -0.08 -0.07 -0.11 -0.09 -0.05 -0.12 -0.03 -0.02
##      [,49] [,50] [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60]
## ACF   0.00 -0.03  0.09 -0.05  0.08 -0.02 -0.01  0.00  0.04  0.05 -0.07  0.10
## PACF -0.03 -0.11 -0.01 -0.04  0.01 -0.02 -0.01 -0.08  0.00  0.08 -0.02  0.04
##      [,61] [,62] [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72]
## ACF  -0.10  0.09 -0.05 -0.01  0.00  0.01 -0.06  0.02  0.02 -0.03  0.01 -0.02
## PACF  0.01  0.01  0.02 -0.03  0.01  0.01 -0.03 -0.04  0.00 -0.01 -0.02 -0.04
##      [,73] [,74] [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84]
## ACF   0.01  0.01 -0.03 -0.03  0.03  0.00 -0.06  0.02 -0.06  0.05 -0.08  0.05
## PACF  0.00  0.04 -0.03 -0.03  0.04  0.04 -0.01 -0.01 -0.05  0.03 -0.04  0.00
##      [,85] [,86] [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96]
## ACF  -0.07  0.06 -0.07  0.01  0.04 -0.07  0.05 -0.05  0.06 -0.06  0.09 -0.11
## PACF -0.06  0.00 -0.04 -0.01  0.02 -0.03  0.00 -0.02  0.02 -0.01  0.03 -0.07
##      [,97] [,98] [,99] [,100] [,101] [,102] [,103] [,104] [,105] [,106] [,107]
## ACF   0.09 -0.06  0.07   -0.03    0.01    0.07   -0.06    0.12     0.0   -0.03    0.08
```

```
## PACF -0.03 -0.06 -0.03  -0.02  -0.09   0.02  -0.04   0.00    0.1  -0.03   0.08
##      [,108] [,109] [,110] [,111] [,112] [,113] [,114] [,115] [,116] [,117]
## ACF   -0.09   0.09  -0.05   0.02  -0.02   0.00   0.04  -0.04   0.01   0.00
## PACF   0.00   0.01   0.02  -0.03   0.01  -0.04   0.04   0.02  -0.04   0.01
##      [,118] [,119] [,120] [,121] [,122] [,123] [,124] [,125] [,126] [,127]
## ACF    0.03  -0.06   0.06  -0.08   0.09  -0.11   0.04  -0.02   0.01  -0.01
## PACF   0.06  -0.03   0.02   0.00   0.03  -0.02  -0.04   0.01  -0.02   0.00
##      [,128] [,129] [,130] [,131] [,132] [,133] [,134] [,135] [,136] [,137]
## ACF   -0.07   0.09  -0.07   0.08  -0.06   0.01  -0.03   0.00   0.00  -0.03
## PACF  -0.08  -0.01   0.00   0.03   0.04   0.03   0.00   0.01   0.01   0.02
##      [,138] [,139] [,140] [,141] [,142] [,143] [,144] [,145] [,146] [,147]
## ACF    0.02  -0.01   0.00  -0.01  -0.02   0.04  -0.04  -0.03   0.04  -0.02
## PACF   0.03   0.04   0.03   0.03  -0.03   0.08   0.03  -0.02  -0.02   0.00
##      [,148] [,149] [,150] [,151] [,152] [,153] [,154] [,155] [,156] [,157]
## ACF    0.02   0.02  -0.03   0.08  -0.08   0.03  -0.02   0.03   0.07  -0.07
## PACF   0.01   0.05   0.00   0.05  -0.05  -0.08  -0.05  -0.07   0.07  -0.01
##      [,158] [,159] [,160] [,161] [,162] [,163] [,164] [,165] [,166] [,167]
## ACF    0.11  -0.10   0.08  -0.03   0.06  -0.01   0.01   0.02  -0.02   0.04
## PACF   0.04  -0.05  -0.05  -0.03  -0.02   0.01  -0.01   0.01   0.01   0.02
##      [,168] [,169] [,170] [,171] [,172] [,173] [,174] [,175] [,176] [,177]
## ACF   -0.07   0.02   0.01   0.02  -0.03   0.00  -0.01  -0.01  -0.02   0.01
## PACF  -0.04  -0.03   0.02   0.05   0.04  -0.02   0.03  -0.01  -0.04  -0.02
##      [,178] [,179] [,180] [,181] [,182] [,183] [,184] [,185] [,186] [,187]
## ACF    0.01  -0.02   0.04  -0.04   0.00   0.00  -0.01  -0.02  -0.03   0.05
## PACF  -0.02  -0.04   0.01   0.02  -0.04  -0.02   0.02   0.00  -0.04   0.03
##      [,188] [,189] [,190] [,191] [,192] [,193] [,194] [,195] [,196] [,197]
## ACF   -0.06   0.03  -0.03   0.00  -0.02   0.03   0.00  -0.02  -0.01  -0.04
## PACF  -0.01  -0.01   0.00  -0.04  -0.04   0.01   0.04  -0.02  -0.04  -0.05
##      [,198] [,199] [,200] [,201] [,202] [,203] [,204] [,205] [,206] [,207]
## ACF    0.05  -0.02  -0.01   0.01   0.00   0.06  -0.03   0.00   0.03  -0.02
## PACF   0.00   0.02   0.01   0.03  -0.02   0.09   0.01   0.01   0.03   0.00
##      [,208]
## ACF    0.04
## PACF   0.00
```

```r
acf = ARMAacf(xt, 24)
pacf = ARMAacf(xt, 24, pacf = TRUE)

par(mfrow = 2:1)
tsplot(acf, type = "h", xlab = "lag", main = "Differenced cmort Data")
abline(h = 0)
tsplot(pacf, type = "h", xlab = "lag")
abline(h = 0)
```

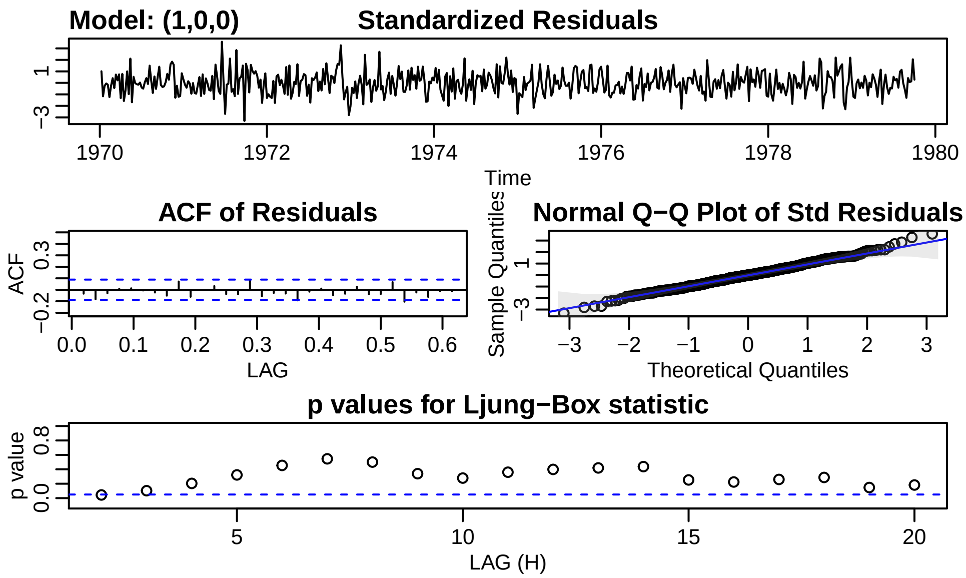


The graph shows that the ACF drops off somewhat steadily, while the PACF drops abruptly at lag 1, which (according to Table 4.1) indicates that AR(1) is an appropriate model for xt.

(c) Fit an AR(1) to xt using maximum likelihood (basically unconditional least squares) as in Section 4.3. The easiest way to do this is to use sarima from astsa. Comment on the significance of the regression parameter estimates of the model. What is the estimate of the white noise variance?

```r
sarima(xt, p = 1, d = 0, q = 0)
```

```
## initial  value 1.908706
## iter   2 value 1.760342
## iter   3 value 1.760341
## iter   4 value 1.760341
## iter   4 value 1.760341
## iter   4 value 1.760341
## final  value 1.760341
## converged
## initial  value 1.760656
## iter   2 value 1.760656
## iter   3 value 1.760656
## iter   3 value 1.760656
## iter   3 value 1.760656
## final  value 1.760656
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, transform.pars = trans,
##     fixed = fixed, optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1    xmean
##       -0.5064  -0.0263
## s.e.   0.0383   0.1715
##
## sigma^2 estimated as 33.81:  log likelihood = -1612.05,  aic = 3230.11
##
## $degrees_of_freedom
## [1] 505
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1    -0.5064 0.0383 -13.2233  0.0000
## xmean  -0.0263 0.1715  -0.1533  0.8782
##
## $AIC
## [1] 6.371023
##
## $AICc
## [1] 6.37107
##
## $BIC
## [1] 6.396044
```

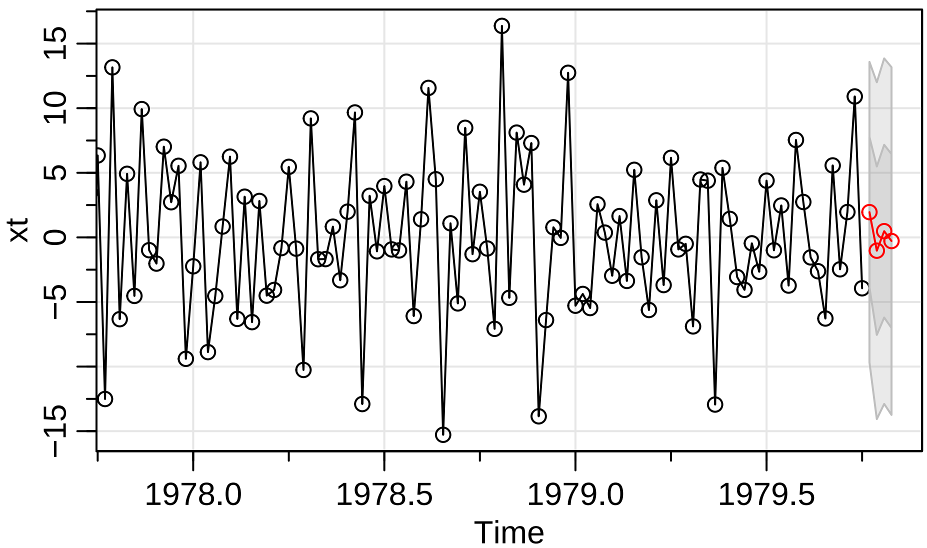
The fitted model gives us an estimate for phi as -0.5604, and a p-value of 0. The white noise variance is estimated to be 33.81.

(d) Examine the residuals and comment on whether or not you think the residuals are white.

From the plot produced from sarima above, we can see that the residuals behave like white noise, which is reinforced by the normal q-q plot, where the errors are almost all on the normal line, and the ACF plot where there are no significant correlations.

(e) Assuming the fitted model is the true model, find the forecasts over a four week horizon, xn n+m, for m = 1, 2, 3, 4, and the corresponding 95% prediction intervals; n = 508 here. The easiest way to do this is to use sarima.for from astsa.

```
par(mfrow = c(1, 1))
forecast = sarima.for(xt, n.ahead = 4, p = 1, d = 0, q = 0)
```




```
forecast
```

```
## $pred
## Time Series:
## Start = c(1979, 41)
## End = c(1979, 44)
## Frequency = 52
## [1]  1.9555424 -1.0298842  0.4818973 -0.2836494
##
## $se
## Time Series:
## Start = c(1979, 41)
## End = c(1979, 44)
## Frequency = 52
## [1] 5.814551 6.517559 6.685932 6.728429
```

```
cbind(
  forecast$pred,
  forecast$pred - 1.96*forecast$se,
  forecast$pred + 1.96*forecast$se
)
```

```
## Time Series:
```

```
## Start = c(1979, 41)
## End = c(1979, 44)
## Frequency = 52
##          forecast$pred forecast$pred - 1.96 * forecast$se
## 1979.769     1.9555424                          -9.440977
## 1979.788    -1.0298842                         -13.804299
## 1979.808     0.4818973                         -12.622530
## 1979.827    -0.2836494                         -13.471371
##          forecast$pred + 1.96 * forecast$se
## 1979.769                           13.35206
## 1979.788                           11.74453
## 1979.808                           13.58632
## 1979.827                           12.90407
```

(f) Show how the values obtained in part (e) were calculated.

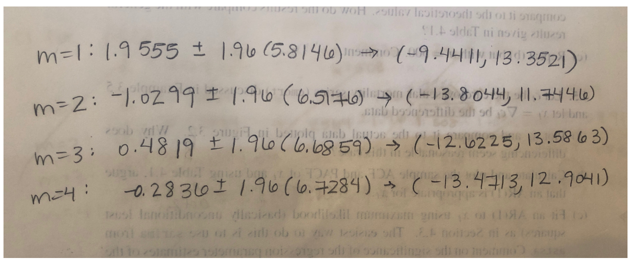
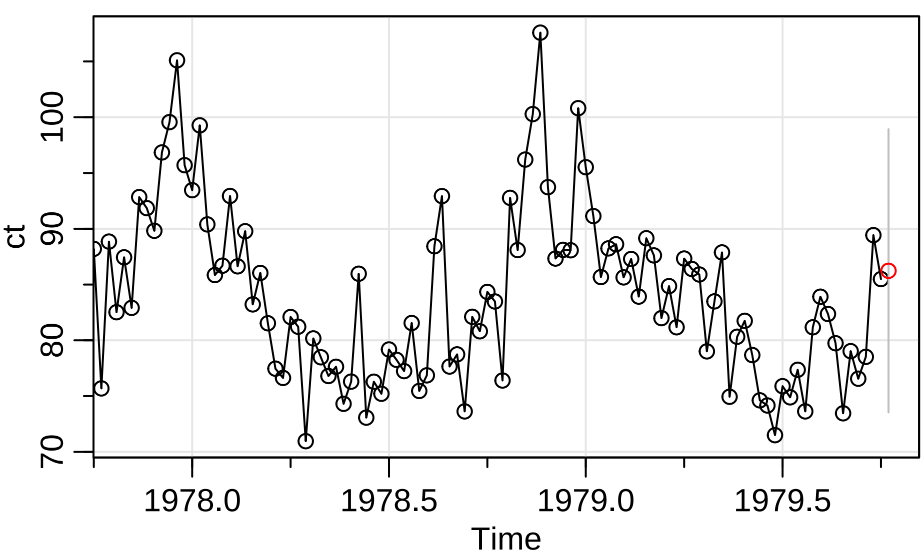

m=1: 1.9555 ± 1.96(5.8146) → (-9.4411, 13.3521)
m=2: -1.0299 ± 1.96 (6.5746) → (-13.8044, 11.7446)
m=3: 0.4819 ± 1.96(6.6859) → (-12.6225, 13.5863)
m=4: -0.2836 ± 1.96(6.7284) → (-13.4713, 12.9041)

Figure 2: Confidence Intervals

Using the output from the sarima in (e), the intervals use the prediction values at each m step ahead (m = 1, 2, 3, 4), alpha of .95 (consequently z* = 1.96), and the se values for each m step ahead predictions.

(g) What is the one-step-ahead forecast of the actual value of cardiovascular mortality; i.e., what is cnn +1?

```
forecast.c = sarima.for(ct, n.ahead = 1, p = 1, d = 0, q = 0)
```



```r
forecast.c
```

```
## $pred
## Time Series:
## Start = c(1979, 41)
## End = c(1979, 41)
## Frequency = 52
## [1] 86.23254
##
## $se
## Time Series:
## Start = c(1979, 41)
## End = c(1979, 41)
## Frequency = 52
## [1] 6.345681
```

```r
forecast.c$pred
```

```
## Time Series:
## Start = c(1979, 41)
## End = c(1979, 41)
## Frequency = 52
## [1] 86.23254
```

The one-step-ahead prediction for the actual cmort data is 86.23254.