Homework 7

Brian Detweiler

March 29, 2017

1. Consider the AR(2) model: $Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$. If we have a series of length 100, with $r_1 = 0.5, r_2 = -0.2, r_3 = 0.1, \overline{y} = 4$ and $s^2 = 6$, use the method of moments estimators to calculate estimates of c, ϕ_1, ϕ_2 and σ_e^2 manually.

From 7.1.2 in the book, we have

$$\begin{aligned}
\hat{\phi}_1 &= \frac{r_1(1-r_2)}{1-r_1^2} \\
&= \frac{0.5(1-(-0.2))}{1-(0.5)^2} \\
&= 0.8 \\
\hat{\phi}_2 &= \frac{r_2 - r_1^2}{1-r_1^2} \\
&= \frac{-0.2-(0.5)^2}{1-(0.5)^2} \\
&= -0.6
\end{aligned}$$

From 7.1.8 in the book, we can calculate the noise variance,

$$\begin{aligned}
\hat{\sigma}_e^2 &= (1 - \hat{\phi}_1 r_1 - \hat{\phi}_2 r_2) s^2 \\
&= (1 - 0.8(0.5) - (-0.6)(-0.2))6 \\
&= 2.88
\end{aligned}$$

With Method of Moments, we let $\hat{\mu} = \overline{y}$, so

$$\begin{aligned}
(Y_t - \mu) &= c + \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + e_t \\
(Y_t - \overline{y}) &= c + \phi_1(Y_{t-1} - \overline{y}) + \phi_2(Y_{t-2} - \overline{y}) + e_t \\
c &= (Y_t - \overline{y}) - \phi_1(Y_{t-1} - \overline{y}) - \phi_2(Y_{t-2} - \overline{y}) - e_t \\
\hat{c} &= (Y_t - 4) - \hat{\phi}_1(Y_{t-1} - 4) - \hat{\phi}_2(Y_{t-2} - 4) - e_t \\
\hat{c} &= (Y_t - 4) - (0.8)(Y_{t-1} - 4) - (-0.6)(Y_{t-2} - 4) - e_t
\end{aligned}$$

■

2. Assume that the following data arise from a stationary process: 8, 7, 6, 9, 7. Calculate method-of-moments estimates of μ, γ_0, and ρ_1.

We only have the 5 data points to go off of, so we'll calculate the sample mean and sample variance and use that to get our estimates of the true paramaters.

```
dat <- c(8, 7, 6, 9, 7)
```

$$
\begin{aligned}
\overline{y} &= 7.4 \\
s_k &= \frac{1}{n}\sum_{i=1}^{n-k}(y_i - \overline{y})(y_{i+k} - \overline{y}) \\
\hat{\gamma_0} &= s_0 = \frac{1}{5}\sum_{i=1}^{5}(y_i - \overline{y})(y_i - \overline{y}) \\
&= \frac{(8-7.4)^2 + (7-7.4)^2 + (6-7.4)^2 + (9-7.4)^2 + (7-7.4)^2}{5} \\
&= \frac{5.2}{5} \\
&= 1.04 \\
s_1 &= \frac{1}{5}\sum_{i=1}^{4}(y_i - \overline{y})(y_{i+1} - \overline{y}) \\
&= \frac{(8-7.4)(7-7.4) + (7-7.4)(6-7.4) + (6-7.4)(9-7.4) + (9-7.4)(7-7.4)}{5} \\
&= \frac{-2.56}{5} \\
&= -0.512 \\
\hat{\rho_1} &= r_1 = \frac{s_1}{s_0} = \frac{-0.512}{1.04} \\
&= -0.4923077
\end{aligned}
$$

■

3. Simulate (use your NUID as the seed) an MA(1) series with $\theta = 0.7$ and $n = 36$. Read pages 443-444 of the text book to learn the R functions that compute the following estimators.

```
set.seed(48548493)
n <- 36
Y_t <- arima.sim(n=n, model=list(ma=c(-0.7)))
```

(a) Find the method of moments estimator of θ. (Hint: the user-created function `estimate.ma1.mom`)

```
estimate.ma1.mom <- function(x) {
  r <- acf(x, plot=F)$acf[1]
  if (abs(r) < 0.5) {
    return(-(-1 + sqrt(1 - 4 * r^2)) / (2 * r))
  } else {
    return(NA)
  }
}

theta1.hat <- estimate.ma1.mom(Y_t)
```

The MoM estimate is $\hat{\theta_1}$ = -0.413039.

(b) Find the (conditional) least squares estimator of θ. (Hint: the `arima` function with `method="CSS"`)

```
a <- arima(x=Y_t, order = c(0,0,1), method="CSS")
theta1.hat <- a$coef[[1]]
```

The conditional sum of squares estimate for θ_1 is -0.5665862.

(c) Find the maximum likelihood estimator of θ. (Hint: the `arima` function with `method="ML"`)

```
a <- arima(x=Y_t, order = c(0,0,1), method="ML")
theta1.hat <- a$coef[[1]]
```

The conditional sum of squares estimate for θ_1 is -0.6080094.

(d) Compare the three estimators, which is closest to the actual θ?

The Maximum Likelihood is the closest, and performs best with smaller sample sizes.

(e) Generate 1000 simulated series and repeat parts (a) - (d). Make three histograms for the three estimators resp ectively. Comment on the results.

```
n <- 1000
Y_t <- arima.sim(n=n, model=list(ma=c(-0.7)))

theta1.hat <- estimate.ma1.mom(Y_t)
```

Method of Moments: $\hat{\theta}$ = -0.8419917

```
a <- arima(x=Y_t, order = c(0,0,1), method="CSS")
theta1.hat <- a$coef[[1]]
```

Conditional Sum of Squares: $\hat{\theta}$ = -0.7306098

```
a <- arima(x=Y_t, order = c(0,0,1), method="ML")
theta1.hat <- a$coef[[1]]
```

Maximum Likelihood: $\hat{\theta}$ = -0.7317521

With more data, the Conditional Sum of Squares performs slightly better than the Maximum Likelihood estimator, but both are fairly close.

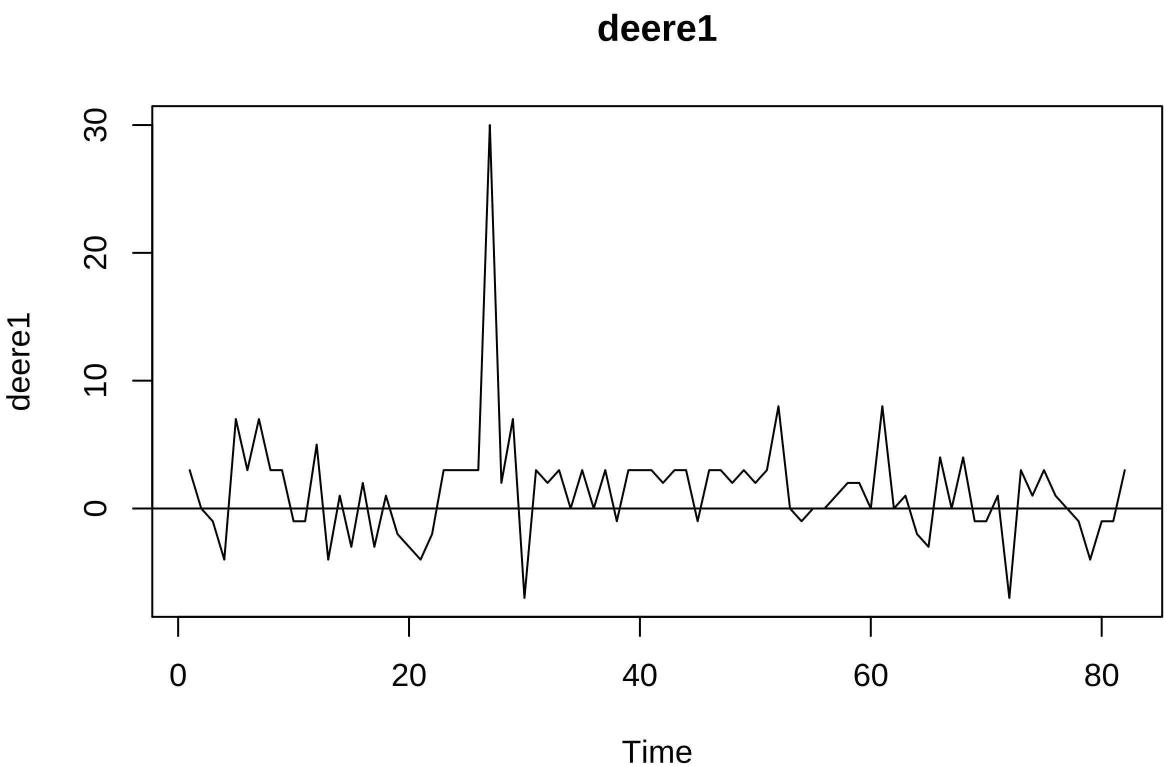
■

4. Three data files `deere1`, `deere2`, `deere3` from the `TSA` package contains different number of consecutive values for the amount of deviation from some specific target values by three machining process. For each of the three data sets, do the following:

```
data(deere1)
data(deere2)
data(deere3)
```

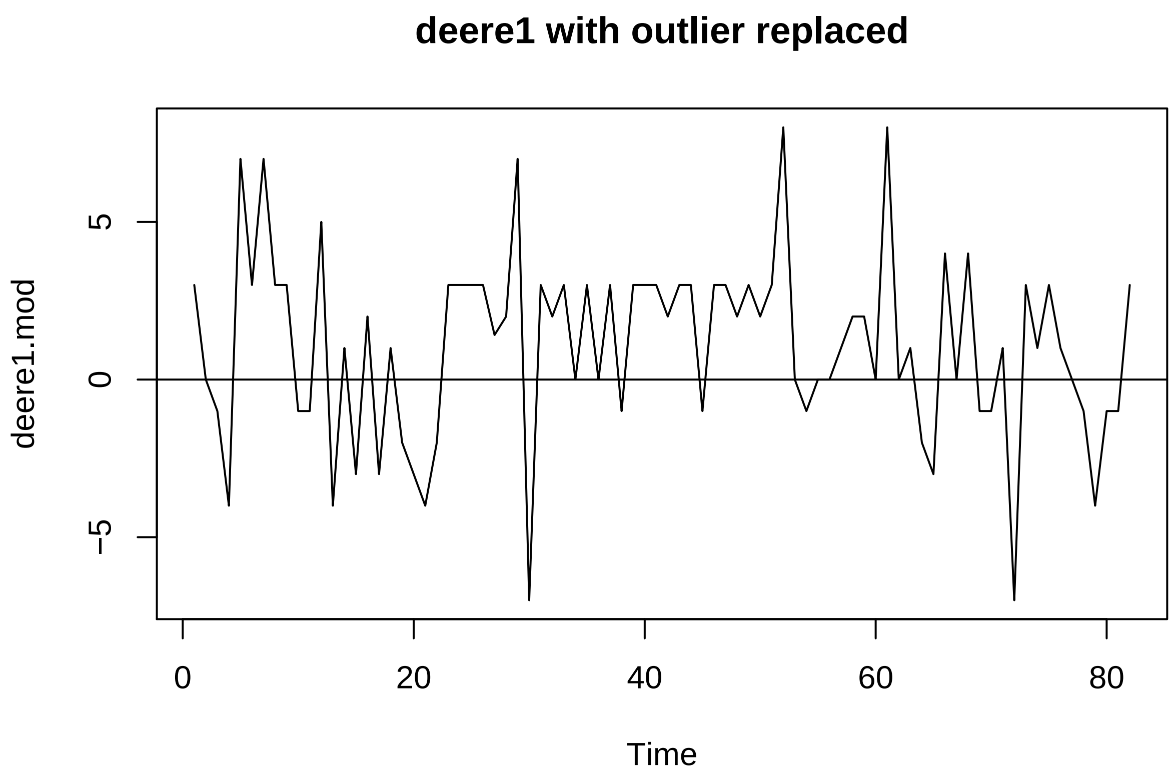
(a) Plot the time series and comment on its appearance. Would a stationary model seem to be appropriate?

```
plot(deere1, main="deere1")
abline(h=0)
```

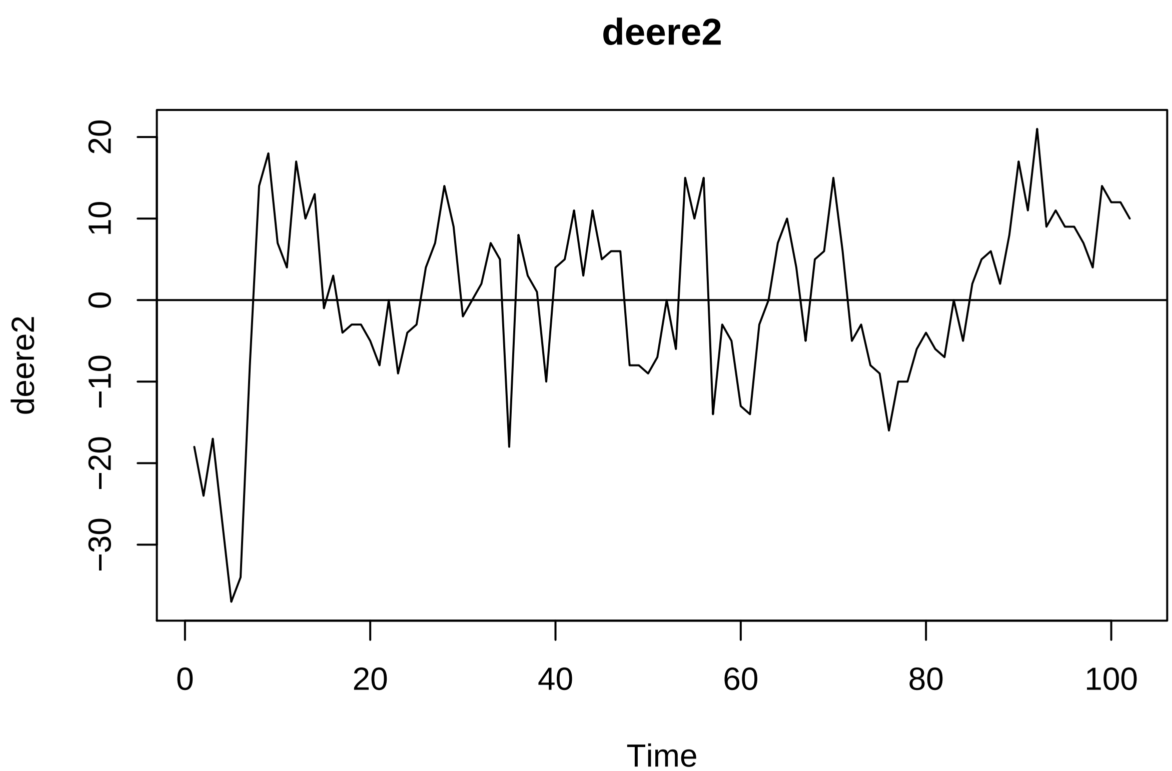


This looks stationary except for the one outlier at time 27. We could replace this with the mean, 1.4146341 and treat it as stationary.

```
deere1.mod <- deere1
deere1.mod[27] <- mean(deere1)
plot(deere1.mod, main="deere1 with outlier replaced")
abline(h=0)
```

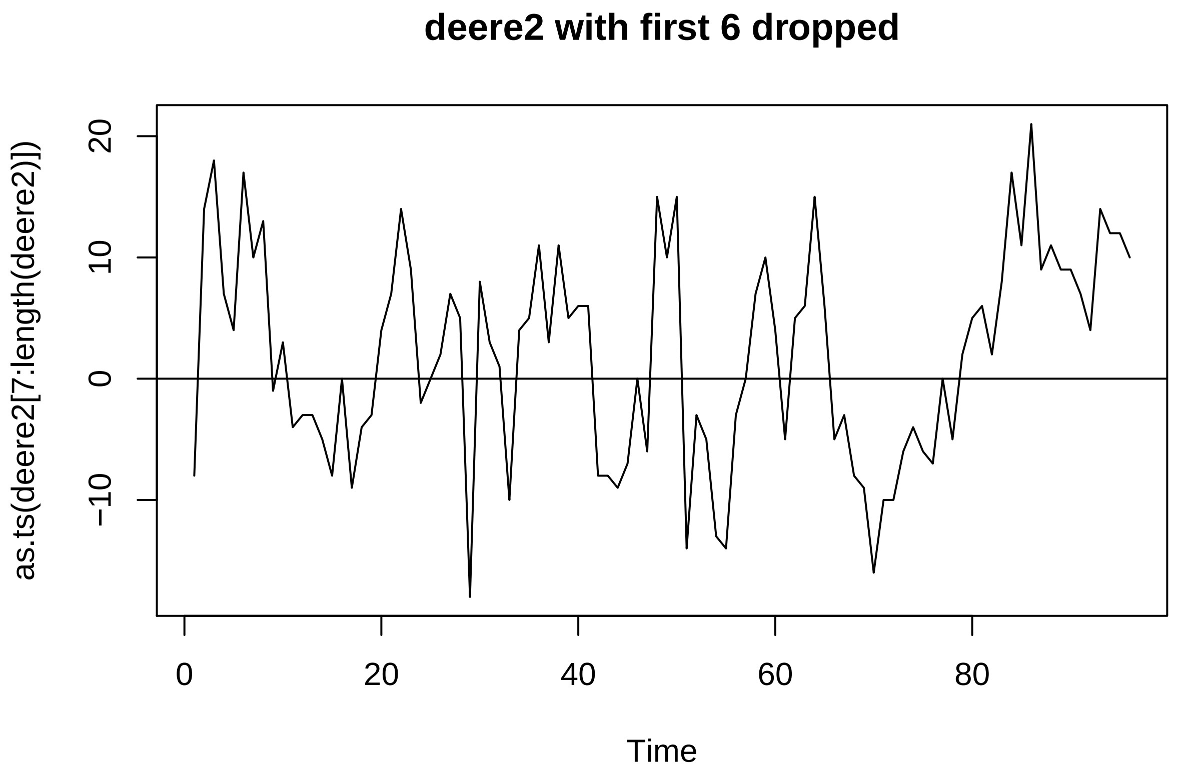



```r
plot(deere2, main="deere2")
abline(h=0)
```

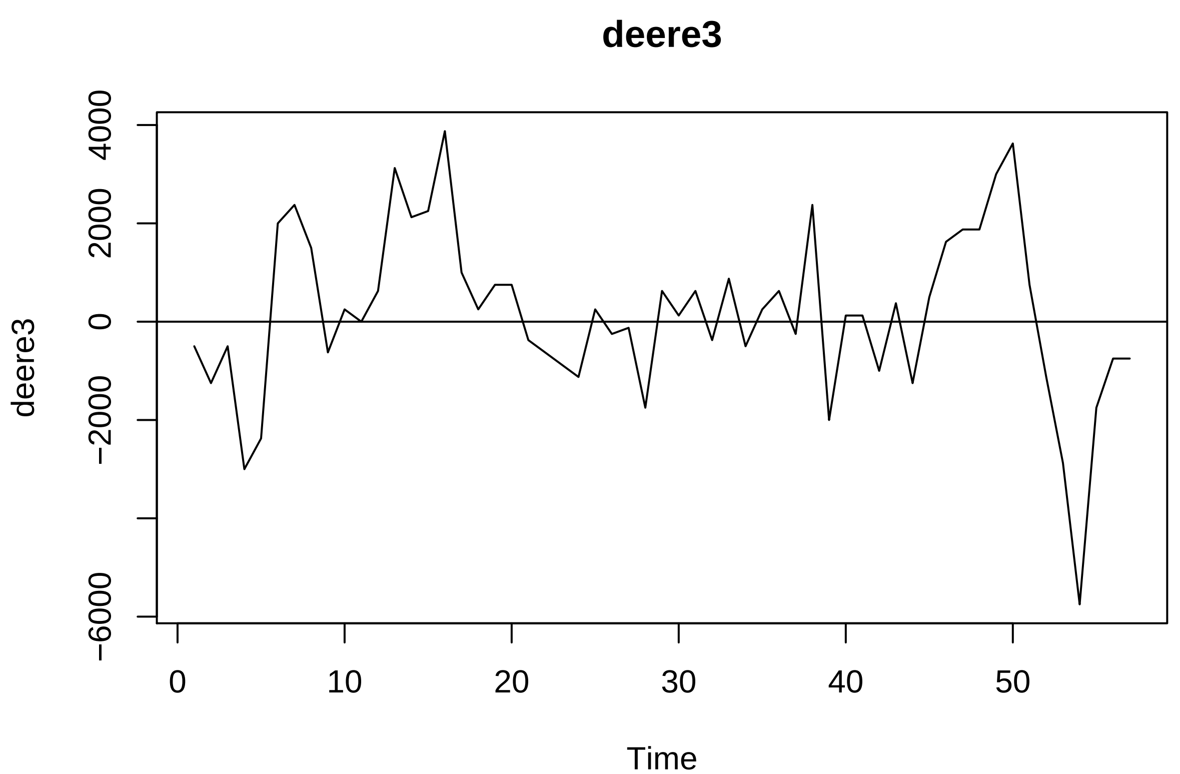


If we drop the first 6 observations, this time series becomes more stationary.

```r
plot(as.ts(deere2[7:length(deere2)]), main="deere2 with first 6 dropped")
abline(h=0)
```

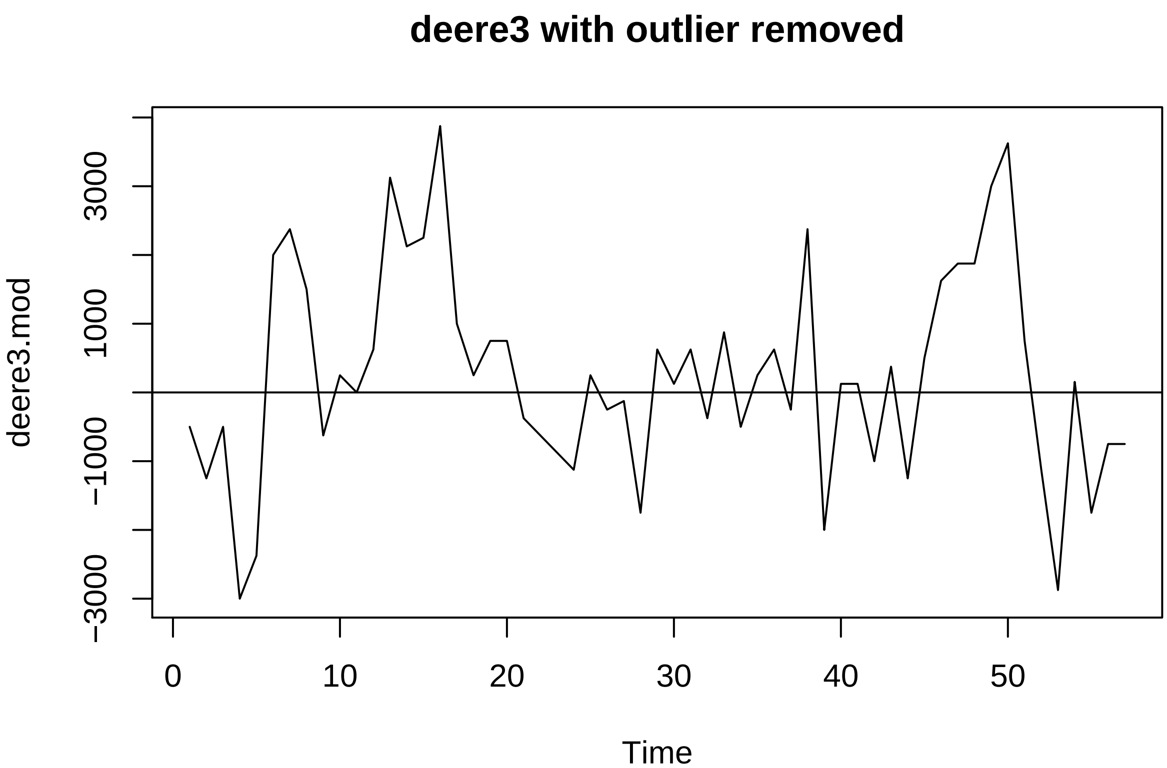



```r
plot(deere3, main="deere3")
abline(h=0)
```



Although there appears to be an outlier at time 54, this appears to be close to stationary.

```
deere3.mod <- deere3
deere3.mod[54] <- mean(deere3)
plot(deere3.mod, main="deere3 with outlier removed")
abline(h=0)
```



(b) Discuss if any log or power transformation should be applied to the series to improve the normality.

Log transformations would not be possible here, because we have negative values. This is even problematic for Box-Cox transformations, since a positive constant must be first added to the negative numbers before applying the transformation. For this reason, transformations are not used.

(c) Perform the (augmented) Dickey-Fuller test on the series. Decide if the series should be differenced to get a stationary model.

```
adf.test(deere1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deere1
## Dickey-Fuller = -3.652, Lag order = 4, p-value = 0.0341
## alternative hypothesis: stationary
```

```
adf.test(deere1.mod)
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  deere1.mod
## Dickey-Fuller = -3.7126, Lag order = 4, p-value = 0.02882
## alternative hypothesis: stationary
```

Performing the augmented Dickey-Fuller test on the first series and it's modified version show that both appear to be stationary. The modified series has a lower p-value, as we would expect.

```r
adf.test(deere2)
```

```
## Warning in adf.test(deere2): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deere2
## Dickey-Fuller = -4.9934, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

With $p << 0.05$, the `deere2` series appears to be stationary.

```r
adf.test(deere3)
```

```
## Warning in adf.test(deere3): p-value smaller than printed p-value
```

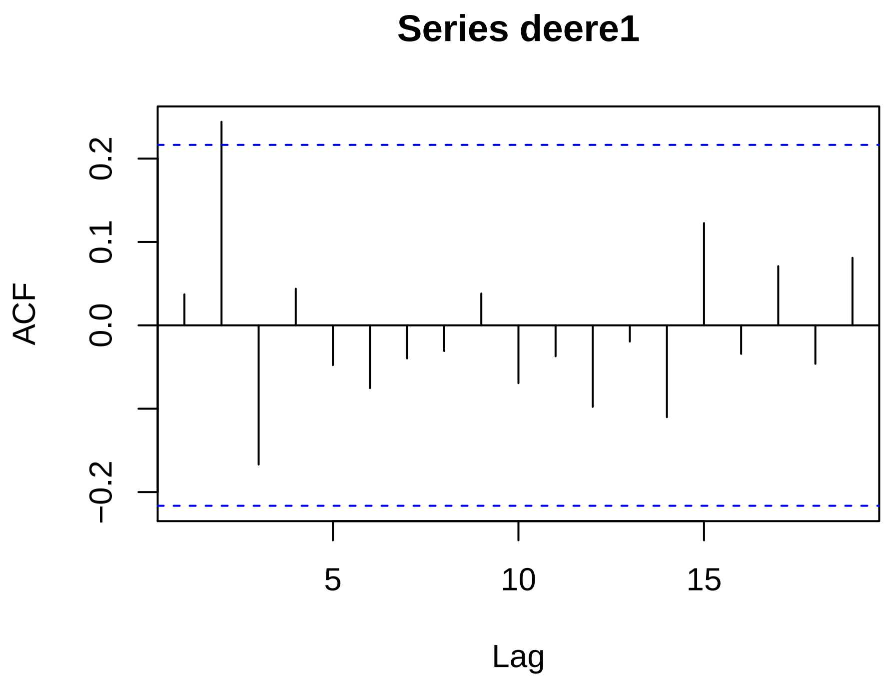
```
##
##  Augmented Dickey-Fuller Test
##
## data:  deere3
## Dickey-Fuller = -4.2521, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

Similarly, $p << 0.05$ for the `deere3` series, so it also appears to be stationary.

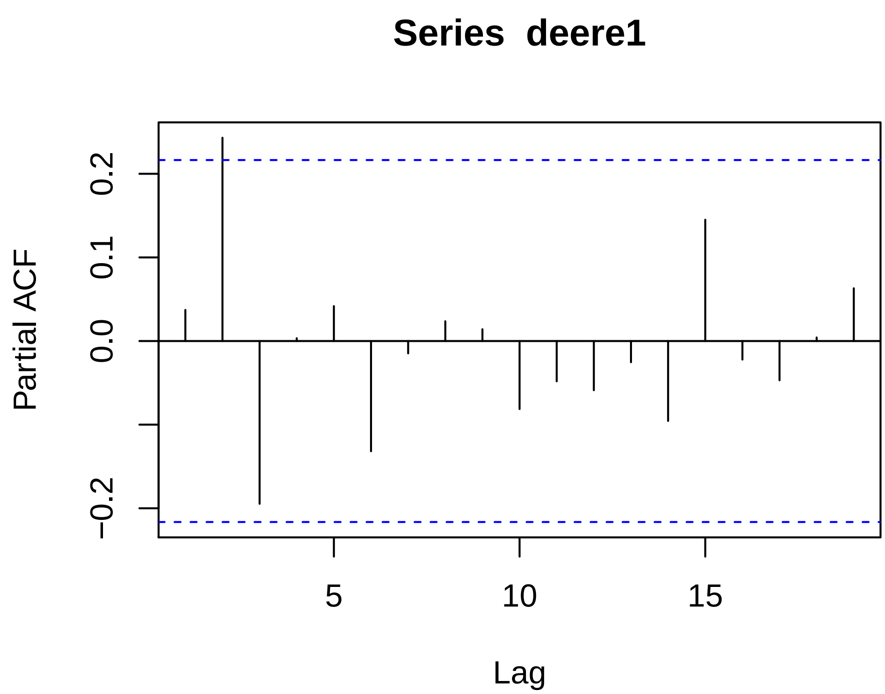
(d) Display the sample `ACF`, `PACF` and `EACF` for the series, and select tentative orders for an ARMA model.

Time series `deere1`

```r
acf(deere1)
```




```r
pacf(deere1)
```



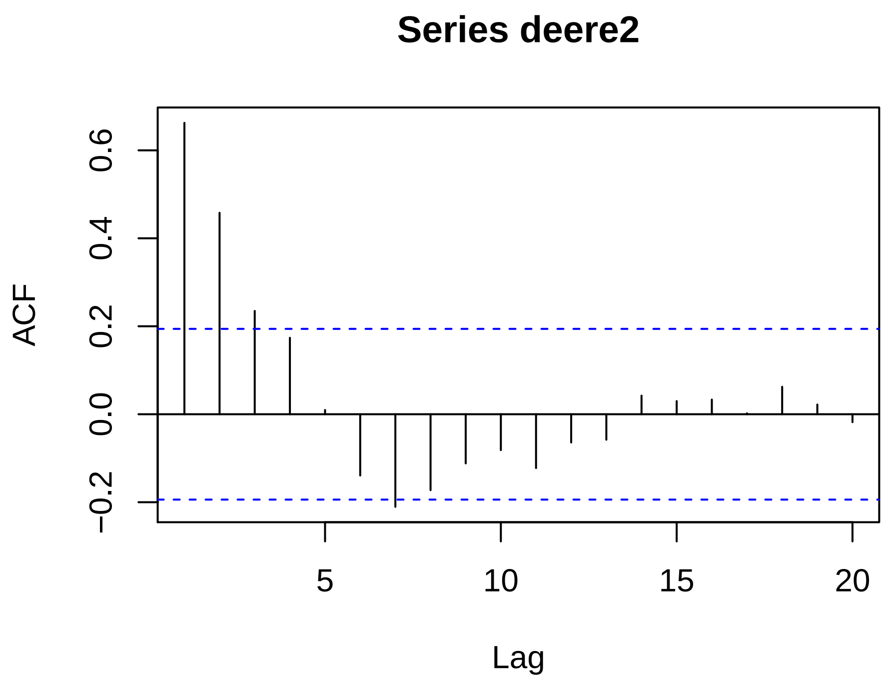

```r
eacf(deere1)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o x o o o o o o o o o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 x x x o o o o o o o o  o  o  o
## 3 o o o o o o o o o o o  o  o  o
## 4 o o x x o o o o o o o  o  o  o
## 5 x o x x o o o o o o o  o  o  o
## 6 o o o x o o o o o o o  o  o  o
## 7 x o o x o o o o o o o  o  o  o
```

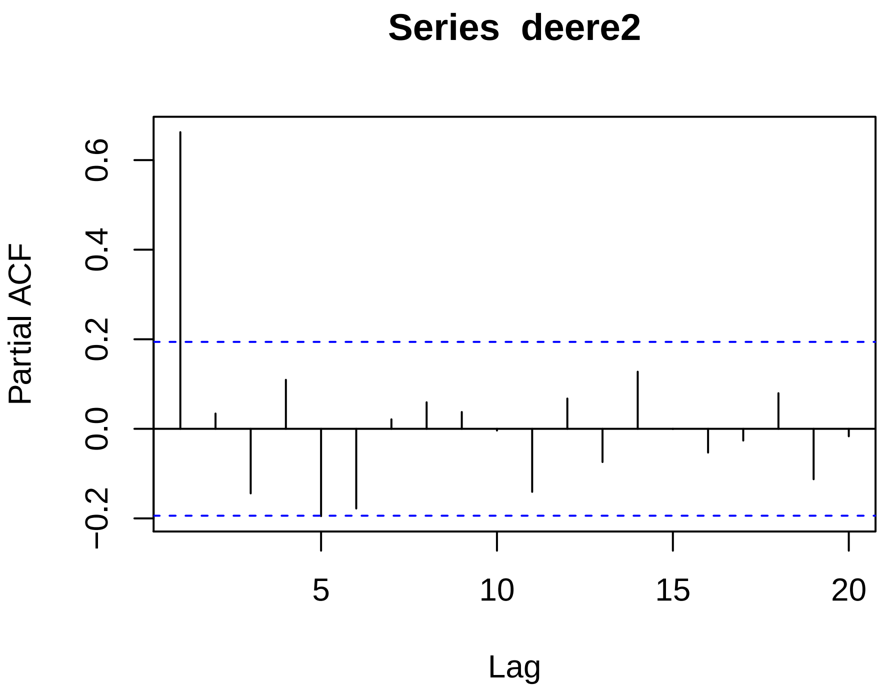
Based on the ACF, PACF, and EACF, I would recommend an ARMA(2, 2) model for this.

Time series `deere2`

```r
acf(deere2)
```




```r
pacf(deere2)
```



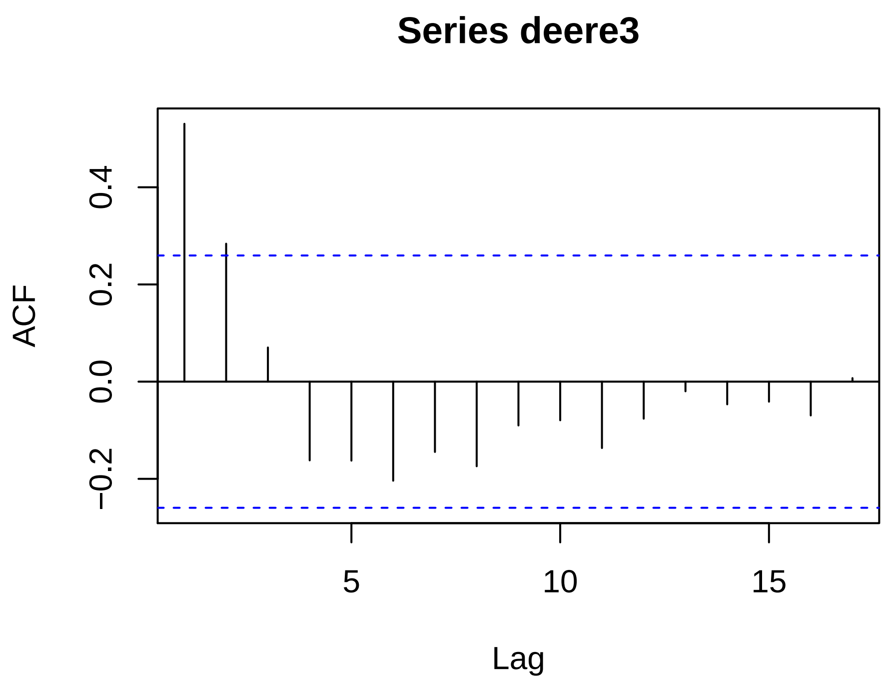

```r
eacf(deere2)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x o o o x o o o o  o  o  o
## 1 o o x o o o o o o o o  o  o  o
## 2 x o o x o o o o o o o  o  o  o
## 3 x o o x o o o o o o o  o  o  o
## 4 x o x x o o o o o o o  o  o  o
## 5 x x o o o o o o o o o  o  o  o
## 6 o o x o o o o o o o o  o  o  o
## 7 x o o o o o o o o o o  o  o  o
```

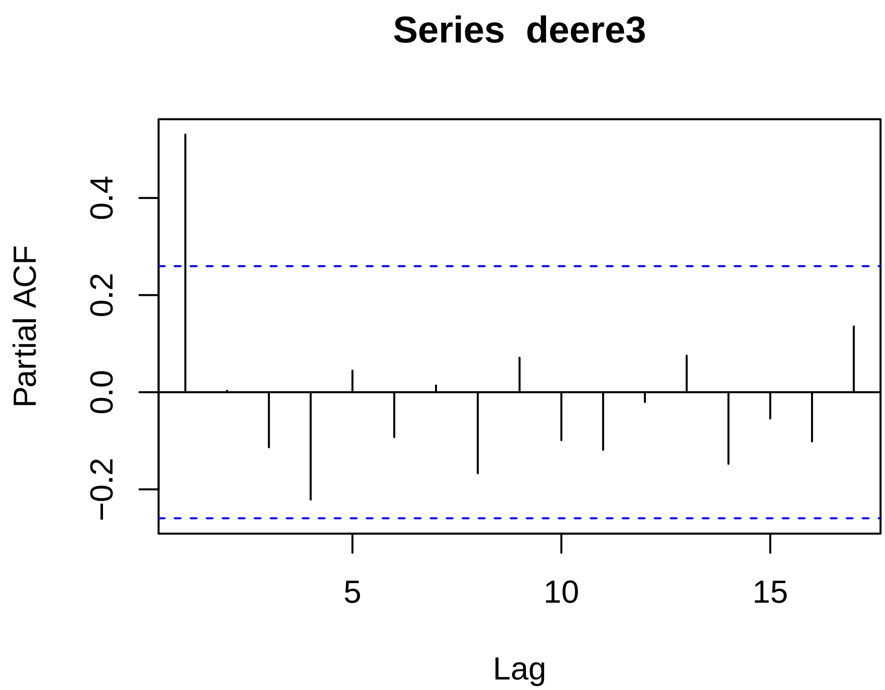
Based on the ACF, PACF, and EACF, I would recommend an ARMA(1, 2) model for this.

Time series `deere3`

```r
acf(deere3)
```




```r
pacf(deere3)
```




```r
eacf(deere3)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x o o o o o o o o o  o  o  o
## 1 o o o o o o o o o o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 x o o o o o o o o o o  o  o  o
## 4 o x o o o o o o o o o  o  o  o
```

```
## 5 o x o x o o o o o o o  o  o  o
## 6 o x o x o o o o o o o  o  o  o
## 7 o x x o o o o o o o o  o  o  o
```

Based on the ACF, PACF, and EACF, I would recommend an ARMA(1, 2) model for this.

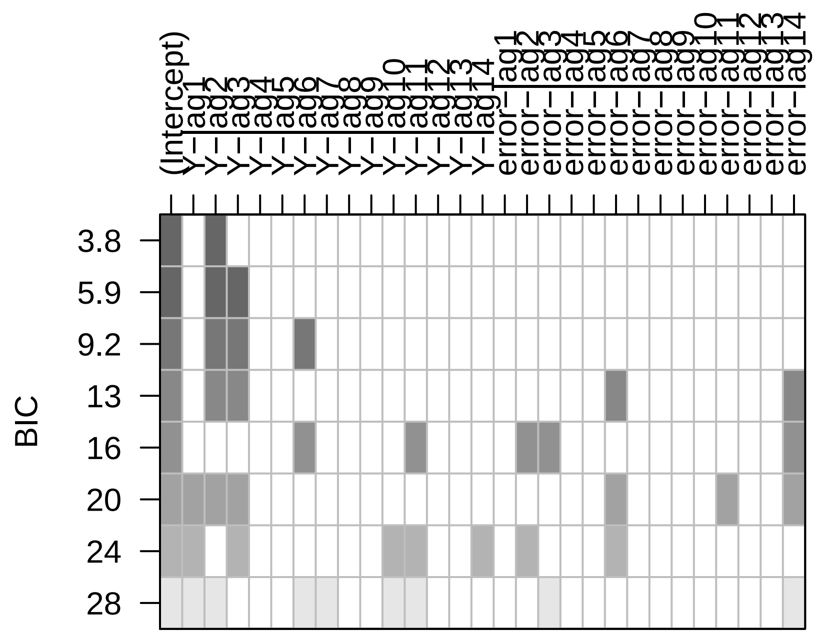
(e) Use the best subsets ARMA approach to specify a model for the series. Compare the result with what you discovered in part (d).

Time series `deere1`

```r
res <- armasubsets(y=deere1, nar=14, nma=14, ar.method='ols')
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 14 linear dependencies found
```

```r
plot(res)
```



This looks to be an AR(2).

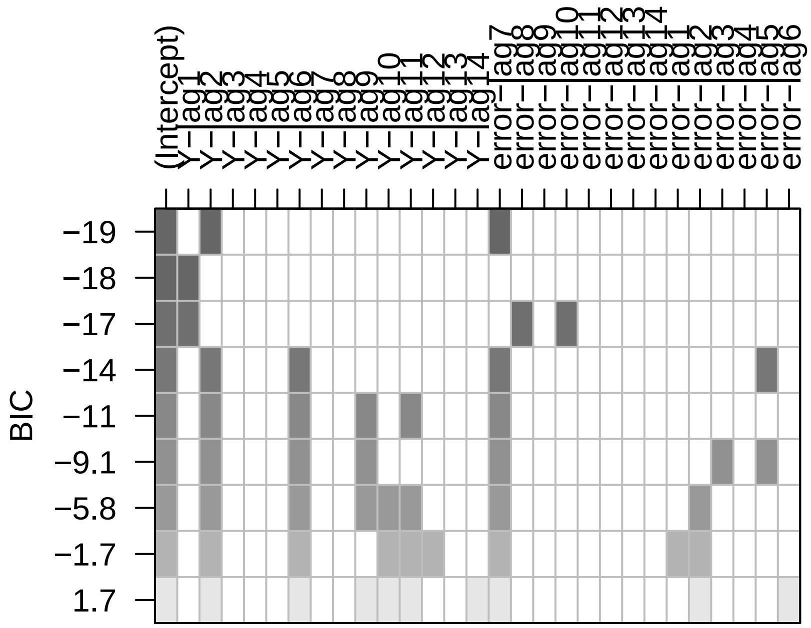
Time series `deere2`

```r
res <- armasubsets(y=deere2, nar=14, nma=14, ar.method='ols')
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 6 linear dependencies found

## Reordering variables and trying again:
```

```r
plot(res)
```



This looks to be an ARMA(1, 1).

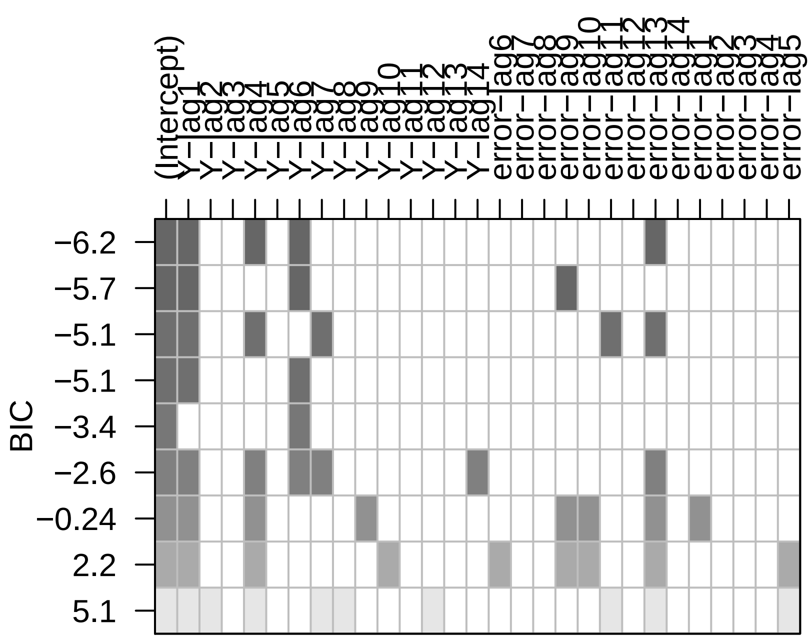
Time series `deere3`

```r
res <- armasubsets(y=deere3, nar=14, nma=14, ar.method='ols')
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 5 linear dependencies found

## Reordering variables and trying again:
```

```r
plot(res)
```



This looks to be an ARMA(2, 1).

■