STA237-hw2

Collin & Qianhui

10/15/2021

Problem 9:

```r
spots_data = read_csv("spots.dat",col_names = F) %>%
  rename("spots" = "X1")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double()
## )
```

```r
spots_data
```

```
## # A tibble: 100 x 1
##    spots
##    <dbl>
##  1   101
##  2    82
##  3    66
##  4    35
##  5    31
##  6     7
##  7    20
##  8    92
##  9   154
## 10   125
## # ... with 90 more rows
```

```r
#make a time series
spots_ts = ts(data = spots_data,start = 1770, end = 1869, frequency = 1)





#transform data to make it stationary
spots_data = spots_data %>%
  mutate(log_diff_spots = log(spots)-log(lag(spots))) %>%
  mutate(grow_rate_spots = (spots - lag(spots))/lag(spots)) %>%
  mutate(log_spots = log(spots)) %>%
  mutate(diff_spots = spots - lag(spots)) %>%
  filter(spots != 0, spots != 1) %>%
  drop_na()
```

```r
spots_data
```

```
## # A tibble: 97 x 5
##    spots log_diff_spots grow_rate_spots log_spots diff_spots
##    <dbl>          <dbl>           <dbl>     <dbl>      <dbl>
##  1    82         -0.208          -0.188      4.41        -19
##  2    66         -0.217          -0.195      4.19        -16
##  3    35         -0.634          -0.470      3.56        -31
##  4    31         -0.121          -0.114      3.43         -4
##  5     7         -1.49           -0.774      1.95        -24
##  6    20          1.05            1.86       3.00         13
##  7    92          1.53            3.6        4.52         72
##  8   154          0.515           0.674      5.04         62
##  9   125         -0.209          -0.188      4.83        -29
## 10    85         -0.386          -0.32       4.44        -40
## # ... with 87 more rows
```

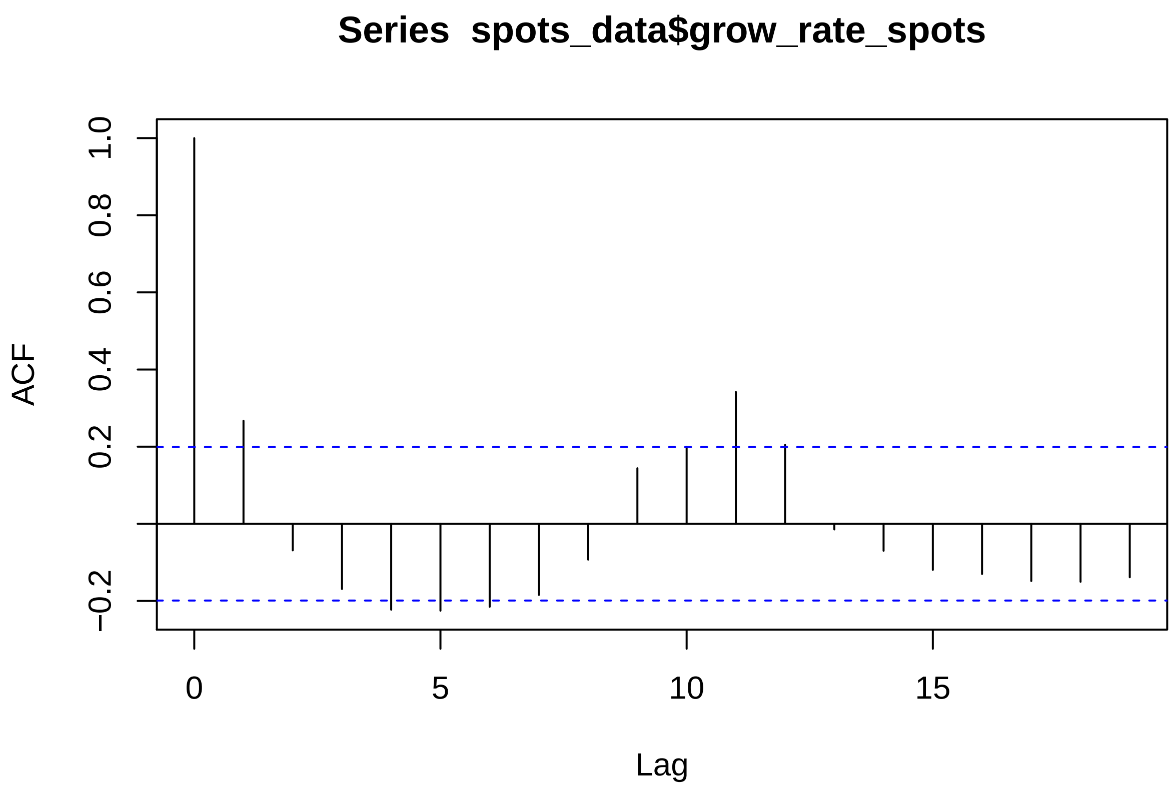
```r
transformed_spots_ts = ts(data = spots_data, start = 1770, end = 1869, frequency = 1)
```

part 1: preliminary inspection of the data:

```r
# acf(spots_data$spots)
# adf.test(spots_data$spots) #potentially unit root, try differencing transformation
#
#
#
# acf(spots_data$diff_spots) #doesnt look stationary?
# adf.test(spots_data$diff_spots) #not unit root, conclude stationarity assumption is met
# pacf(spots_data$diff_spots)
#
# spots_data$grow_rate_spots
spots_data$grow_rate_spots
```

```
##  [1] -0.188118812 -0.195121951 -0.469696970 -0.114285714 -0.774193548
##  [6]  1.857142857  3.600000000  0.673913043 -0.188311688 -0.320000000
## [11] -0.200000000 -0.441176471 -0.394736842 -0.565217391  1.400000000
## [16]  2.458333333  0.590361446 -0.007575758 -0.099236641 -0.237288136
## [21] -0.255555556 -0.104477612 -0.216666667 -0.127659574 -0.487804878
## [26] -0.238095238 -0.625000000 -0.333333333  0.750000000  1.000000000
## [31]  1.428571429  0.323529412 -0.044444444  0.116279070 -0.125000000
## [36] -0.333333333 -0.642857143 -0.200000000 -0.750000000  4.000000000
## [41]  1.400000000  0.166666667  1.500000000  0.314285714 -0.108695652
## [46] -0.268292683 -0.200000000 -0.333333333 -0.562500000 -0.428571429
## [51] -0.500000000  3.000000000  1.125000000  1.117647059  0.388888889
## [56]  0.240000000  0.080645161  0.059701493 -0.323943662 -0.416666667
## [61] -0.714285714  0.625000000  3.384615385  1.140350877  0.131147541
## [66] -0.253623188 -0.165048544 -0.267441860 -0.412698413 -0.351351351
## [71] -0.541666667  0.363636364  1.666666667  0.550000000  0.580645161
## [76]  0.265306122 -0.225806452 -0.312500000 -0.030303030 -0.156250000
## [81] -0.277777778 -0.461538462 -0.666666667 -0.428571429  4.750000000
## [86]  1.391304348  0.709090909  0.021276596 -0.197916667 -0.233766234
## [91] -0.254237288  0.068181818 -0.361702128 -0.466666667 -0.562500000
## [96]  4.285714286  1.000000000
```

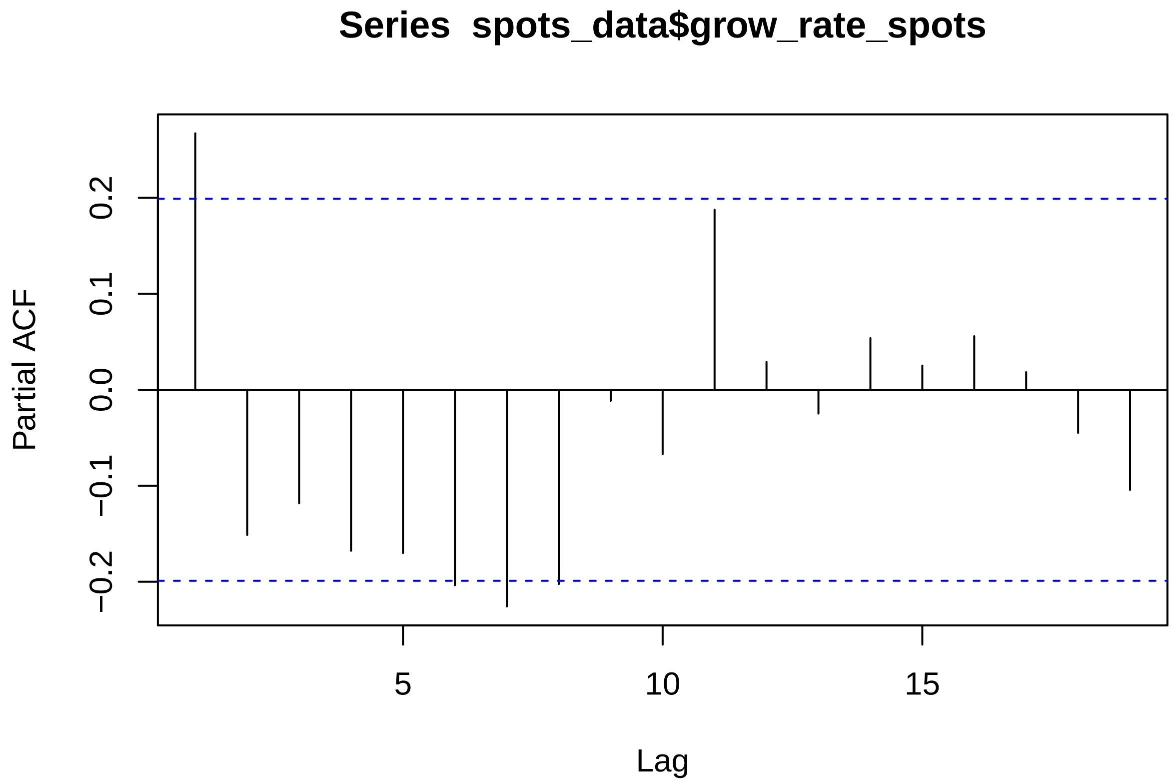
```r
acf(spots_data$grow_rate_spots) #this might be the one
```




```r
adf.test(spots_data$grow_rate_spots) #yessss not unit root, conclude stationary
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag   ADF p.value
## [1,]   0 -6.98    0.01
## [2,]   1 -6.11    0.01
## [3,]   2 -5.46    0.01
## [4,]   3 -5.20    0.01
## Type 2: with drift no trend
##      lag   ADF p.value
## [1,]   0 -7.36    0.01
## [2,]   1 -6.62    0.01
## [3,]   2 -6.13    0.01
## [4,]   3 -6.10    0.01
## Type 3: with drift and trend
##      lag   ADF p.value
## [1,]   0 -7.33    0.01
## [2,]   1 -6.59    0.01
## [3,]   2 -6.10    0.01
## [4,]   3 -6.06    0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
pacf(spots_data$grow_rate_spots) #Ar1
```



After trying several transformations of the sunspots data, I find that a growth rate transformation appears to result in the stationarity assumption being met, based on the ACF plot. Conducting the Augmented-Dickey Fuller Test confirms this, as a p-value $< .01$ leads us to reject the null hypothesis of unit root and conclude that the time series is indeed stationary.

Looking at the PACF, it appears that there is 1 highly statistically significant lag. As a result, I would suggest an AR(1) model to be sufficient for modelling sunspots.

part 2.

```r
ar_model = ar.ols(spots_data$grow_rate_spots,order.max = 1)

ar_model
```

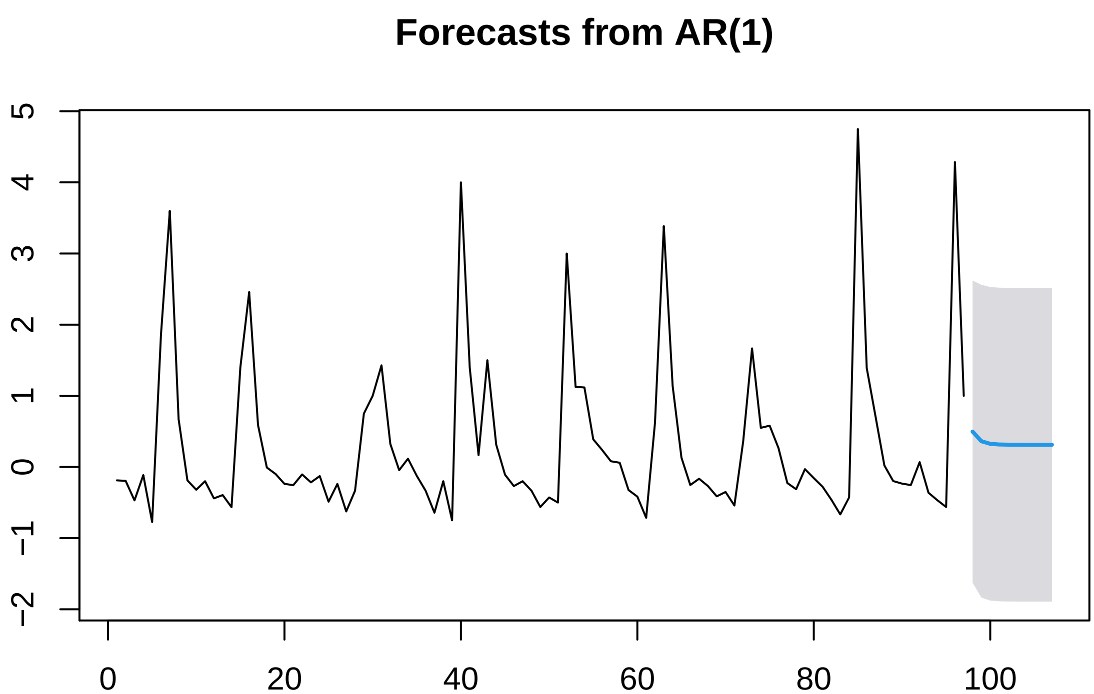
```
##
## Call:
## ar.ols(x = spots_data$grow_rate_spots, order.max = 1)
##
## Coefficients:
##      1
## 0.2683
##
## Intercept: 0.007062 (0.1106)
##
## Order selected 1  sigma^2 estimated as  1.173
```

part 3.

```
fcst = forecast(ar_model,h = 10, level = .95)
fcst
```

```
##     Point Forecast     Lo 95    Hi 95
##  98      0.4968503 -1.626323 2.620023
##  99      0.3618447 -1.836430 2.560119
## 100      0.3256199 -1.877963 2.529203
## 101      0.3159001 -1.888065 2.519865
## 102      0.3132920 -1.890700 2.517284
## 103      0.3125922 -1.891402 2.516586
## 104      0.3124045 -1.891590 2.516399
## 105      0.3123541 -1.891640 2.516348
## 106      0.3123405 -1.891654 2.516335
## 107      0.3123369 -1.891657 2.516331
```

```
plot(fcst)
```



Problem 10:

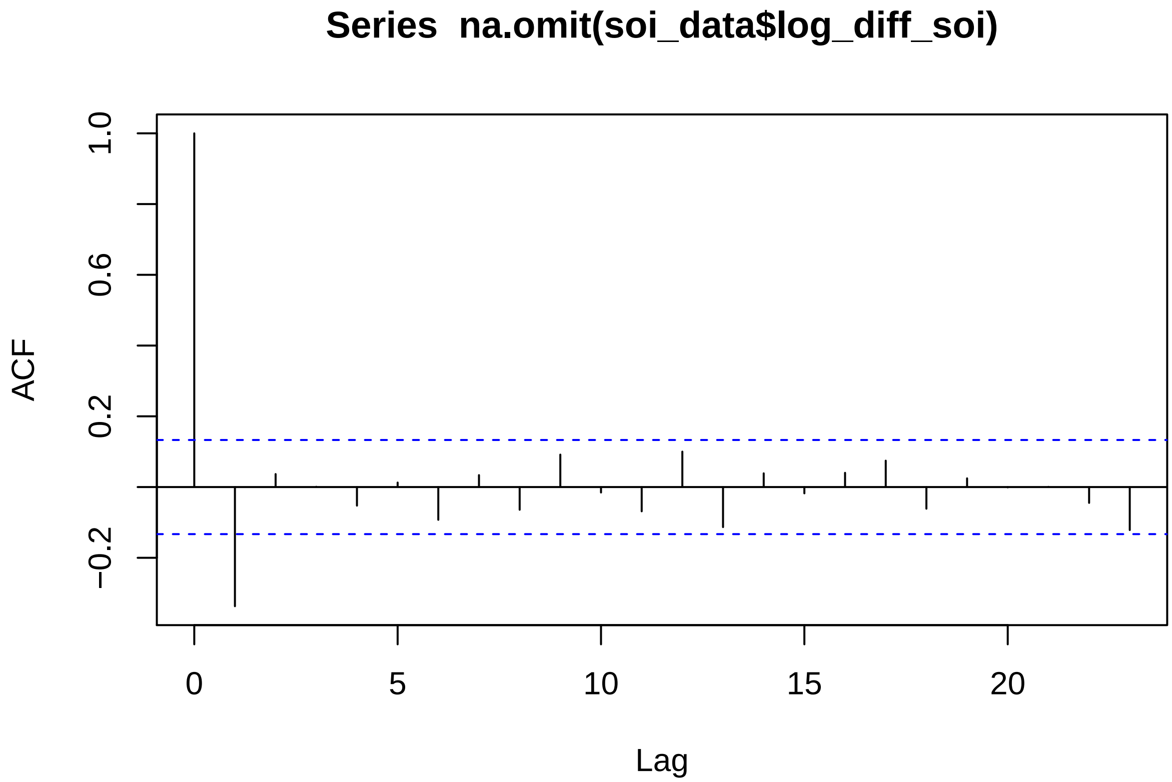
part 1:

```
soi_data = read_csv("soi.dat",col_names = FALSE) %>%
  rename("soi" = "X1") %>%
  mutate(log_diff_soi = log(soi) - log(lag(soi))) %>%
  mutate(growth_rate_soi = (soi - lag(soi))/lag(soi))
```

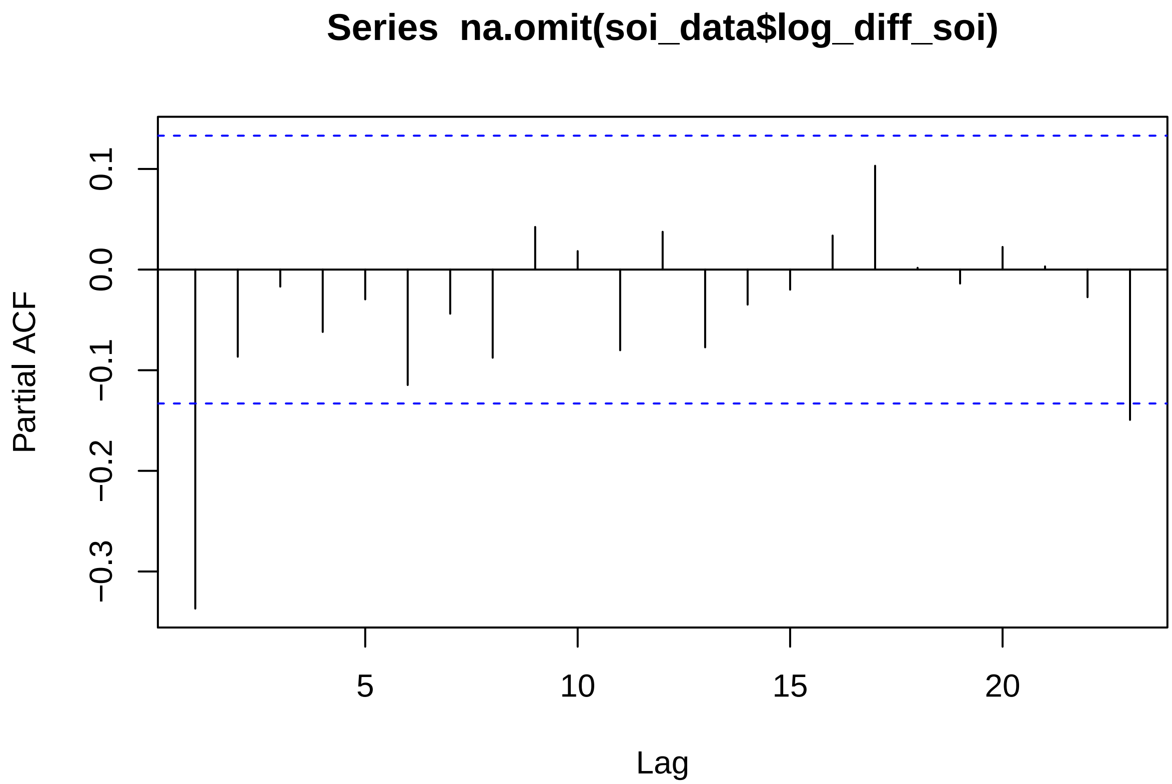
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double()
```

```
## )

## Warning in log(soi): NaNs produced

## Warning in log(lag(soi)): NaNs produced
```

```r
#assuming this is monthly data based on what I found online
soi_ts = ts(soi_data,start = 1, end = 453,frequency = 12 )

acf(na.omit(soi_data$log_diff_soi))
```




```r
pacf(na.omit(soi_data$log_diff_soi))
```



After noticing considerable autocorrelation in the acf plot of the unmodified data, I applied a log-difference transformation and then re-checked the ACF plot. The autocorrelation appears to have been alleviated, and the PACF has one statistically significant lag, suggesting that again an AR(1) model may be sufficient to model the southern oscillation index.

part 2.

```
#creating AR(1) model
soi_ar_model = ar.ols(x = na.omit(soi_data$log_diff_soi,order.max = 1))
soi_ar_model
```

```
##
## Call:
## ar.ols(x = na.omit(soi_data$log_diff_soi, order.max = 1))
##
## Coefficients:
##      1
## -0.34
##
## Intercept: 0.004644 (0.07782)
##
## Order selected 1  sigma^2 estimated as  1.308
```

part 3.

```
soi_forecast = forecast(soi_ar_model,h = 10, level = .95)
#1 - 10 step ahead forecasts along with lower and upper bounds for the predictions intervals:
soi_forecast
```

```
##     Point Forecast     Lo 95    Hi 95
## 218   5.808210e-01 -1.660844 2.822486
## 219  -1.940327e-01 -2.561705 2.173640
## 220   6.939847e-02 -2.312409 2.451206
## 221  -2.016163e-02 -2.403597 2.363274
## 222   1.028660e-02 -2.373337 2.393910
## 223  -6.504605e-05 -2.383710 2.383580
## 224   3.454259e-03 -2.380194 2.387102
## 225   2.257782e-03 -2.381390 2.385906
## 226   2.664554e-03 -2.380984 2.386313
## 227   2.526262e-03 -2.381122 2.386175
```