

DATA624-HW6-ARIMA

FPP-Hyndman exercises 8.1, 8.2, 8.3, 8.5, 8.6, 8.7.

Michael Y.

3/22/2020

Contents

Homework 6 - ARIMA	4
8.1 Figure 8.31 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.	5
a) Explain the differences among these figures. Do they all indicate that the data are white noise?	5
b) Why are the critical values at different distances from the mean of zero?	5
8.2 A classic example of a non-stationary series is the daily closing IBM stock price series (data set <code>ibmclose</code>).	6
Use R to plot the daily closing prices for IBM stock and the ACF and PACF.	6
Explain how each plot shows that the series is non-stationary and should be differenced. . . .	6
8.3 For the following series, find an appropriate Box-Cox transformation and order of differencing in order to obtain stationary data.	8
a) <code>usnetelec</code>	8
b) <code>usgdp</code>	17
c) <code>mcopper</code>	32
d) <code>enplanements</code>	44
e) <code>visitors</code>	60
8.5 For your <code>retail</code> data (from Exercise 3 in Section 2.10), find the appropriate order of differencing (after transformation if necessary) to obtain stationary data.	72
8.6 Use R to simulate and plot some data from simple ARIMA models.	79
a) Use the following R code to generate data from an AR(1) model	79
c) Write your own code to generate data from an MA(1) model with $\theta_1 = 0.6$ and $\sigma^2 = 1$. . .	102
d) Produce a time plot for the series. How does the plot change as you change θ_1 ?	103
e) Generate data from an ARMA(1,1) model with	125
f) Generate data from an AR(2) model with $\phi_1 = -0.8$, $\phi_2 = 0.3$, and $\sigma^2 = 1$	126
g) Graph the latter two series and compare them.	127

8.7 Consider <code>wmurders</code> , the number of women murdered each year (per 100,000 standard population) in the United States.	132
a) By studying appropriate graphs of the series in R, find an appropriate <code>ARIMA(p,d,q)</code> model for these data.	132
b) Should you include a constant in the model? Explain.	137
c) Write this model in terms of the backshift operator.	138
d) Fit the model using R and examine the residuals. Is the model satisfactory?	138
e) Forecast three times ahead. Check your forecasts by hand to make sure that you know how they have been calculated.	139
f) Create a plot of the series with forecasts and prediction intervals for the next three periods shown.	139
g) Does <code>auto.arima()</code> give the same model you have chosen? If not, which model do you think is better?	140

```
library(fpp2)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: forecast
```

```
## Registered S3 method overwritten by 'quantmod':
```

```
##   method      from
```

```
## as.zoo.data.frame zoo
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

```
library(tseries)
```

```
library(urca)
```

Homework 6 - ARIMA

Do the exercises 8.1, 8.2, 8.3, 8.5., 8.6, 8.7 in Hyndman.

Please submit both your Rpubs link as well as attach the .rmd file with your code.

8.1 Figure 8.31 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.

a) Explain the differences among these figures. Do they all indicate that the data are white noise?

The figures indicate that the data are white noise because the ACF plots appear to fall inside the critical bands (though for X2, some of the extreme points might be touching the bands.)

b) Why are the critical values at different distances from the mean of zero?

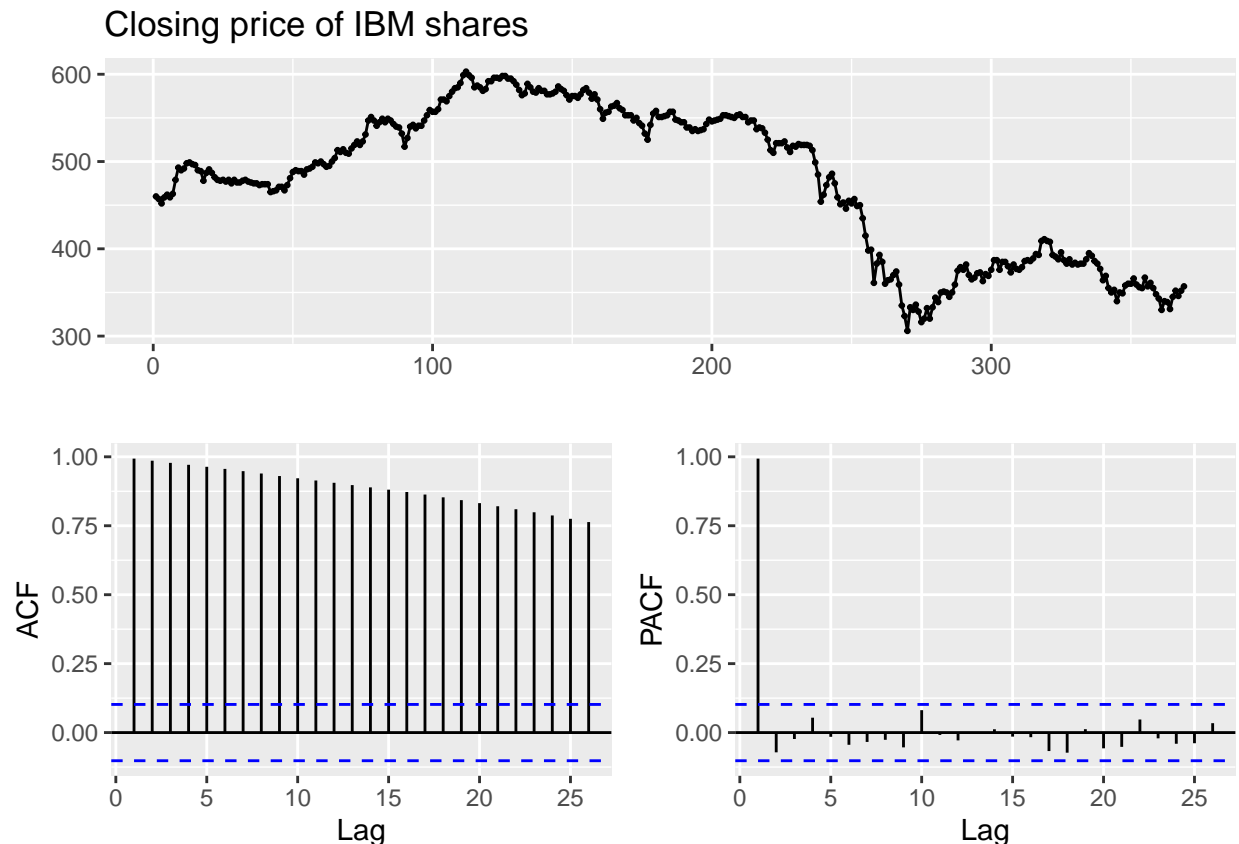
The blue lines represent the 9% confidence interval, which differ depending upon the amount of data. A larger number observations gives a smaller critical value (series X3) while a smaller number of observations gives a larger critical value (X1).

Why are the autocorrelations different in each figure when they each refer to white noise?
Because it is white noise, there should be no discernable pattern. The values on each lag fall below the critical values, which differ based upon the amount of data.

8.2 A classic example of a non-stationary series is the daily closing IBM stock price series (data set `ibmclose`).

Use R to plot the daily closing prices for IBM stock and the ACF and PACF.

```
ggtsdisplay(ibmclose, main="Closing price of IBM shares")
```



Explain how each plot shows that the series is non-stationary and should be differenced.

The ACF plot indicates very high autocorrelation values, well above the critical value. This is because, barring unusual market behavior, the absolute price of a stock tomorrow is somewhat close to the price today (lag 1), and a bit less close on each succeeding day (higher lags.) This is why the ACF values are all high, and slowly declining.

What is interesting in finance is whether the price of the stock will go up or go down from one day to the next. Thus, the net change (which is obtainable by differencing) will result in a series which does not exhibit the strong autocorrelation as the raw price series.

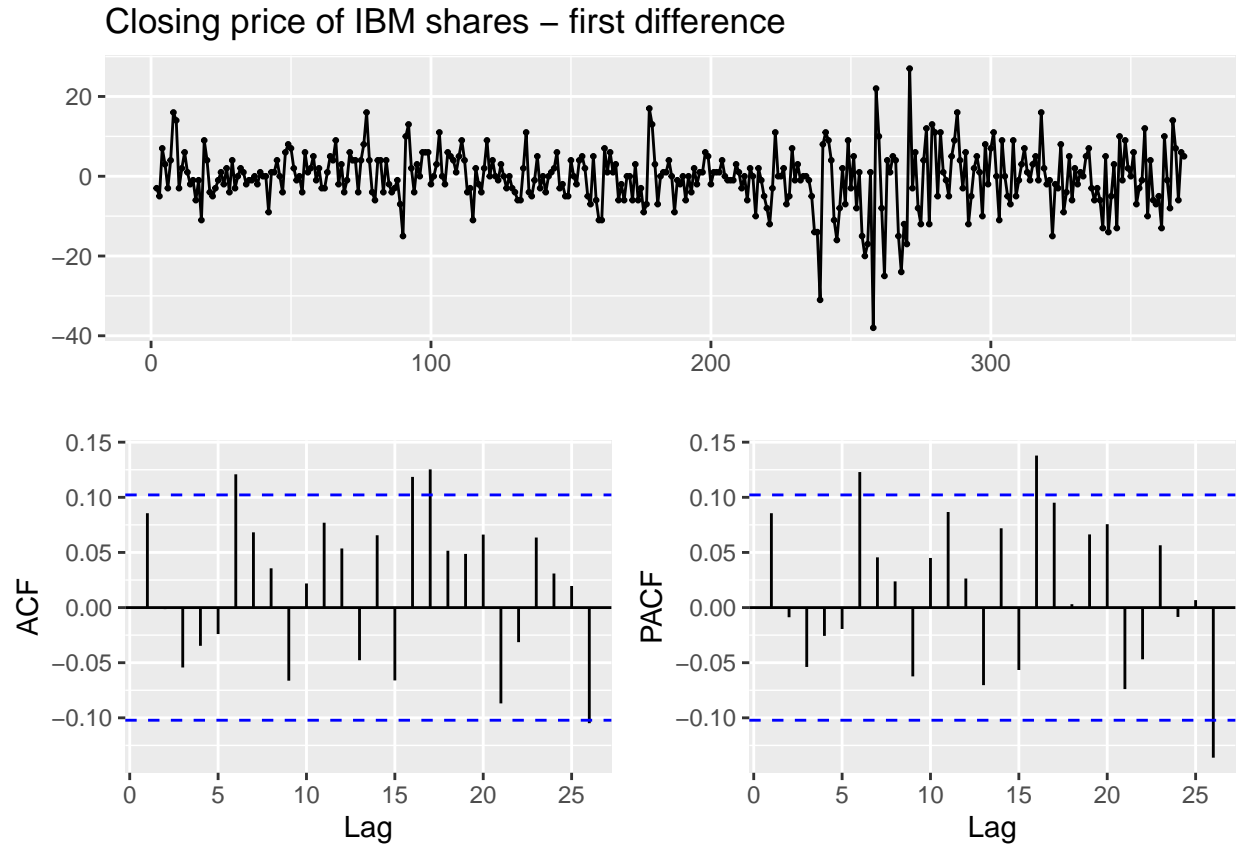
Because the first lag on the PACF chart is nearly 1, this indicates a non-stationary process which should be differenced in order to obtain a stationarity.

```
ibmclose_ndiffs <- ndiffs(ibmclose)
print(paste("Number of differences suggested for ibmclose:", ibmclose_ndiffs))
```

Check estimated number of differences needed:

```
## [1] "Number of differences suggested for ibmclose: 1"
```

```
ggtsdisplay(diff(ibmclose),  
             main="Closing price of IBM shares - first difference")
```



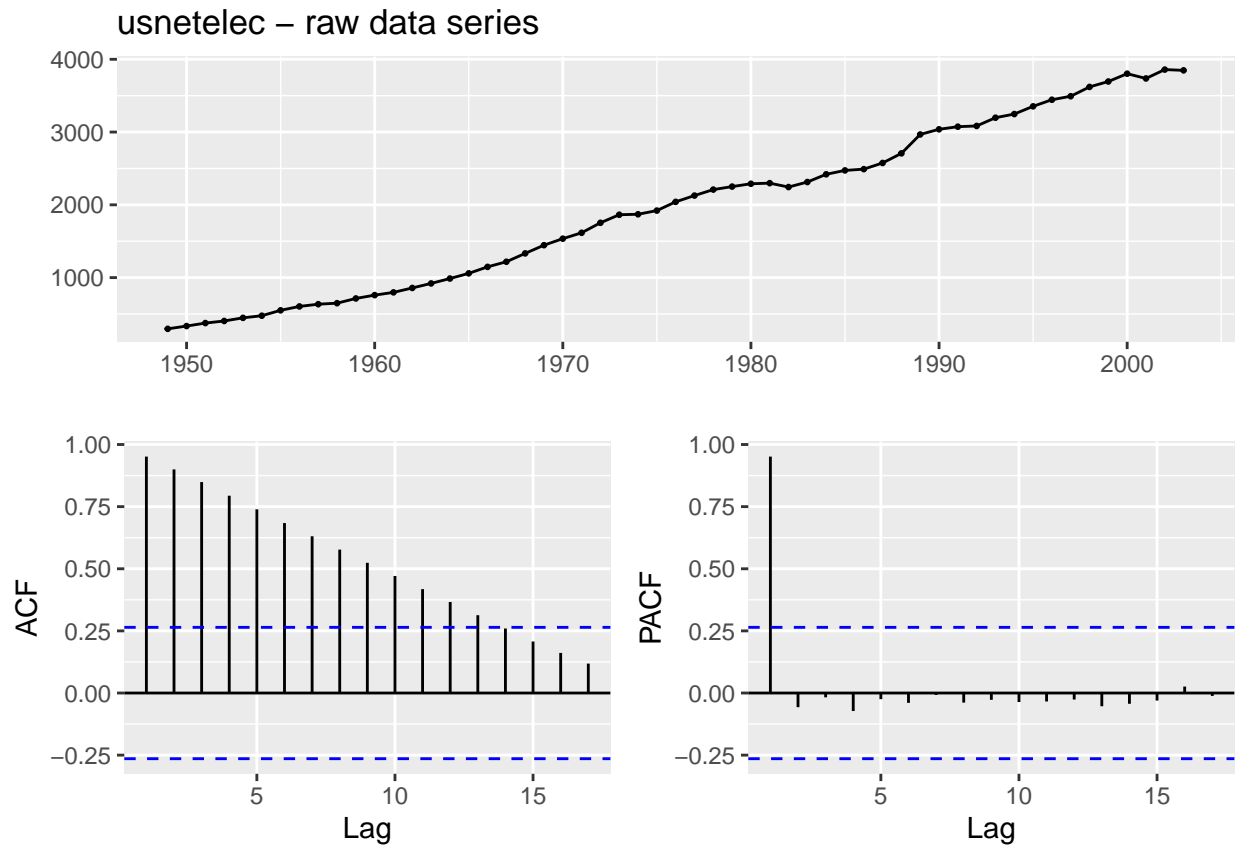
Stationarity can be achieved through such differencing. Sometimes only a single difference is necessary; sometimes additional differencing is required.

We can see that we are now close to achieving stationarity because the lags on the ACF and PACF plots are inside the critical value bands (narrowly exceeding in a few cases.)

8.3 For the following series, find an appropriate Box-Cox transformation and order of differencing in order to obtain stationary data.

a) usnetelec

```
ggtsdisplay(usnetelec, main = "usnetelec - raw data series")
```



```
Box.test(usnetelec, type = "Ljung-Box")
```

Ljung-box test on raw data series:

```
##  
## Box-Ljung test  
##  
## data: usnetelec  
## X-squared = 52.549, df = 1, p-value = 4.196e-13
```

Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

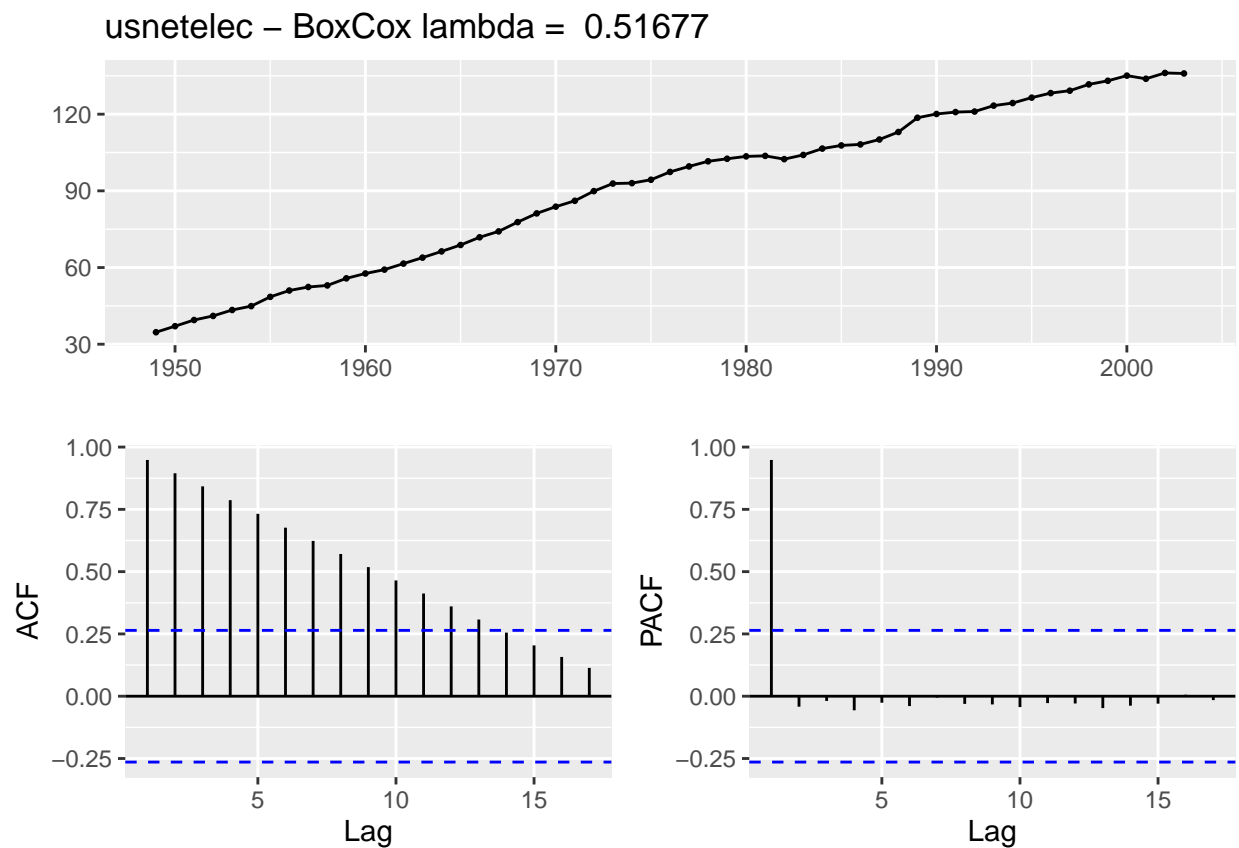

```
usnetelec_lambda <- round(BoxCox.lambda(usnetelec),5)
print(paste("Lambda for usnetelec: ", usnetelec_lambda))
```

Try Box-Cox transformation:

```
## [1] "Lambda for usnetelec: 0.51677"
```

```
usnetelec_BC <- BoxCox(usnetelec, usnetelec_lambda)
```

```
ggtsdisplay(usnetelec_BC, main = paste("usnetelec - BoxCox lambda = ",
                                       usnetelec_lambda))
```



The graph of the Box-Cox transformed data shows strong autocorrelation.

```
Box.test(usnetelec_BC, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: usnetelec_BC
## X-squared = 52.191, df = 1, p-value = 5.035e-13
```

Because the p-value is still low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

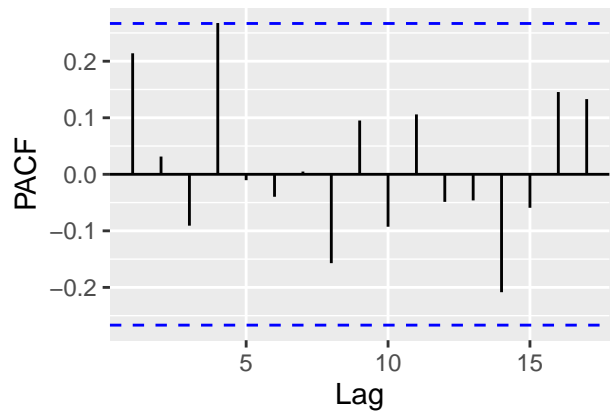
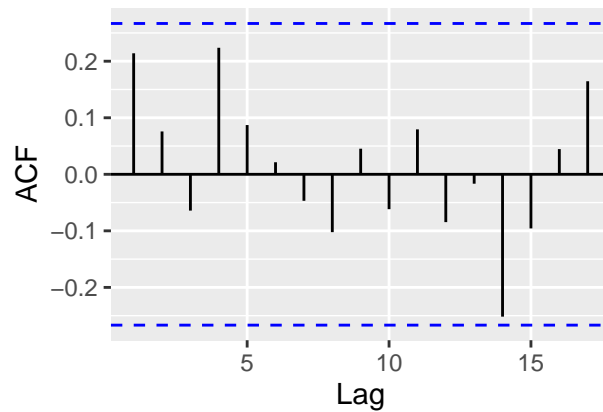
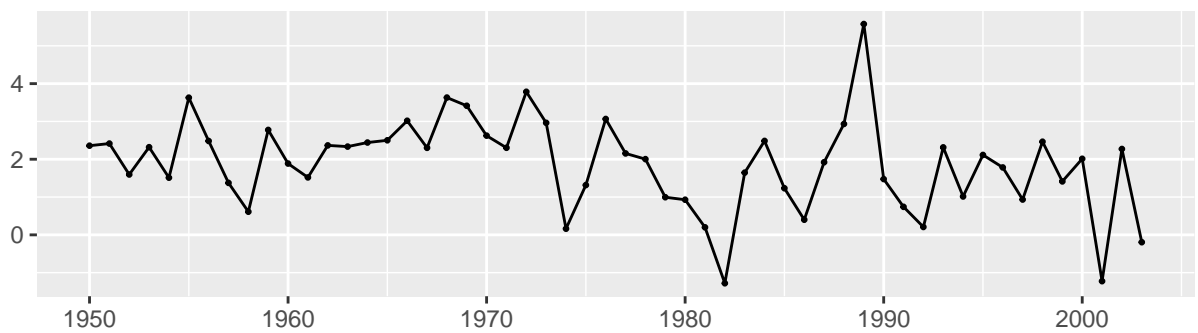
```
usnetelec_BC_ndiffs <- ndiffs(usnetelec_BC)
print(paste("Number of differences suggested for usnetelec_BC:", usnetelec_BC_ndiffs))
```

Check estimated number of differences needed:

```
## [1] "Number of differences suggested for usnetelec_BC: 2"
```

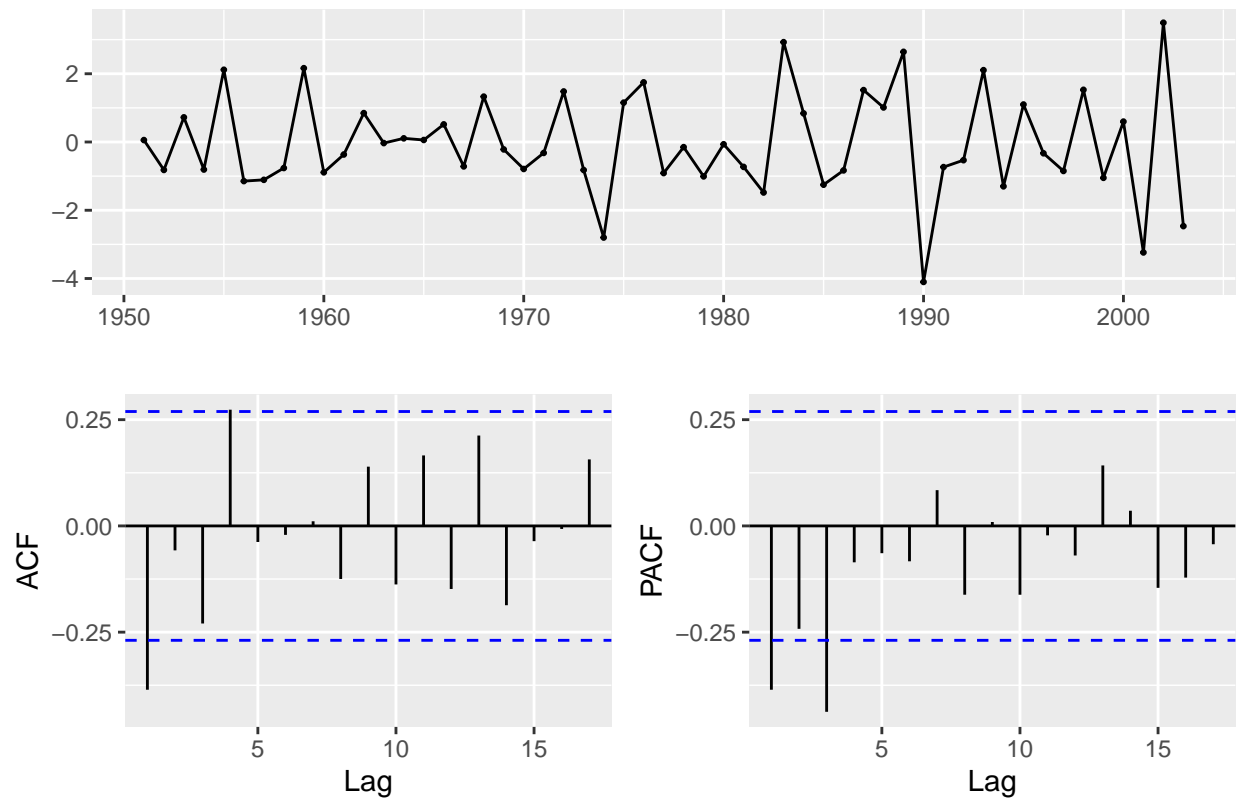
```
ggtsdisplay(diff(usnetelec_BC), main=paste("usnetelec - BoxCox lambda = ",
                                           usnetelec_lambda, " - first difference"))
```

usnetelec - BoxCox lambda = 0.51677 - first difference



```
ggtsdisplay(diff(diff(usnetelec_BC)), main=paste("usnetelec - BoxCox lambda = ",
                                                  usnetelec_lambda, " - second difference"))
```

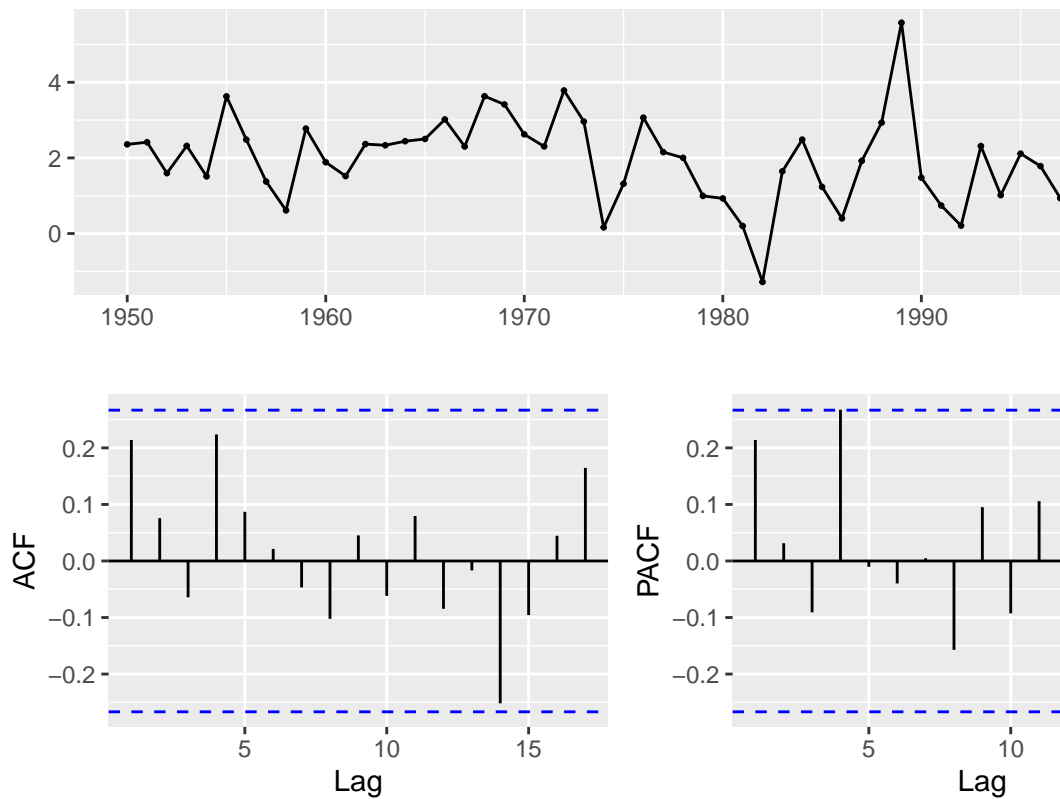
usnetelec – BoxCox lambda = 0.51677 – second difference



Although the result from “ndiffs” estimates that 2 differences are required, the results from a single difference look better.

```
#### Diff data series
usnetelec_BC %>% diff() -> usnetelec_BC_d1
ggtsdisplay(usnetelec_BC_d1, main = paste("usnetelec - BoxCox lambda = ",
                                         usnetelec_lambda, "first difference"))
```

usnetelec – BoxCox lambda = 0.51677 first difference



Let's test first differencing:

```
#### Ljung-box test:
Box.test(usnetelec_BC_d1, type = "Ljung-Box")

##
## Box-Ljung test
##
## data: usnetelec_BC_d1
## X-squared = 2.6132, df = 1, p-value = 0.106
```

Because the p-value is high, we FAIL TO REJECT the null hypothesis, which is that the data are independent (i.e., no serial correlation.) So, first-differenced usnetelec data can be thought of as a white noise series.

```
library(tseries)

#### Test raw data series
kpss.test(usnetelec, "Level", lshort = F)
```

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
##
## KPSS Test for Level Stationarity
```

```
##
## data: usnetelec
## KPSS Level = 0.6153, Truncation lag parameter = 10, p-value = 0.02125
```

```
kpss.test(usnetelec, "Trend", lshort = F)
```

```
## Warning in kpss.test(usnetelec, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: usnetelec
## KPSS Trend = 0.098058, Truncation lag parameter = 10, p-value = 0.1
```

```
usnetelec %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 10 lags.
##
## Value of test-statistic is: 0.6153
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
usnetelec %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 10 lags.
##
## Value of test-statistic is: 0.0981
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```

KPSS Level Test fails on raw data series; Trend test passes

```
kpss.test(usnetelec_BC, "Level", lshort = F)
```

Test Box-Cox transform:

```
##
## KPSS Test for Level Stationarity
##
## data: usnetelec_BC
## KPSS Level = 0.61487, Truncation lag parameter = 10, p-value = 0.02128
```

```
kpss.test(usnetelec_BC, "Trend", lshort = F)
```

```
##
## KPSS Test for Trend Stationarity
##
## data: usnetelec_BC
## KPSS Trend = 0.15177, Truncation lag parameter = 10, p-value = 0.04519
```

```
usnetelec_BC %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 10 lags.
##
## Value of test-statistic is: 0.6149
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
usnetelec_BC %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 10 lags.
##
## Value of test-statistic is: 0.1518
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

Both KPSS tests fail on Box Cox transformed data series.

```
kpss.test(usnetelec_BC_d1, "Level", lshort = F)
```

Test first difference:

```
## Warning in kpss.test(usnetelec_BC_d1, "Level", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: usnetelec_BC_d1
## KPSS Level = 0.33954, Truncation lag parameter = 10, p-value = 0.1
```

```
kpss.test(usnetelec_BC_d1, "Trend", lshort = F)
```

```
## Warning in kpss.test(usnetelec_BC_d1, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: usnetelec_BC_d1
## KPSS Trend = 0.074231, Truncation lag parameter = 10, p-value = 0.1
```

```
usnetelec_BC_d1 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 10 lags.
##
## Value of test-statistic is: 0.3395
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
usnetelec_BC_d1 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

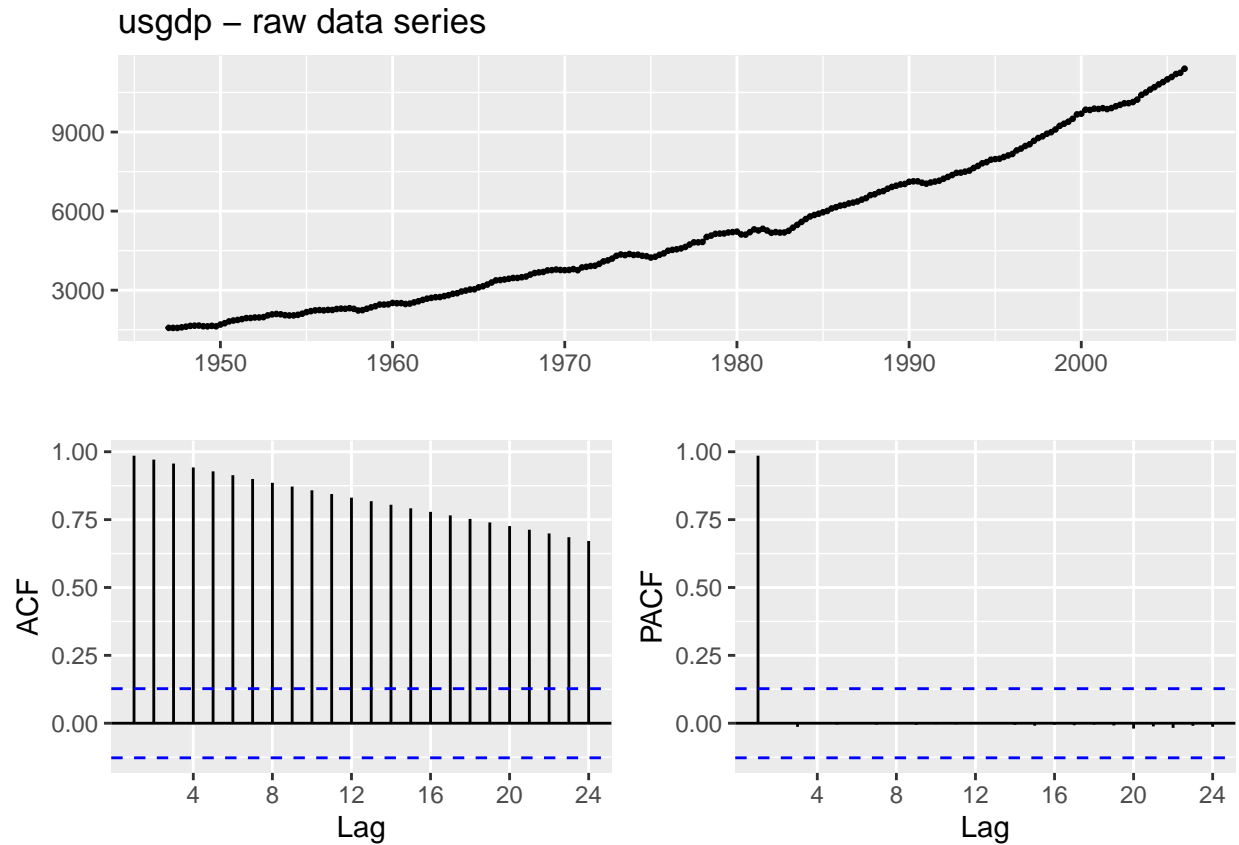
```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 10 lags.
##
## Value of test-statistic is: 0.0742
```

```
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146  0.176 0.216
```

Both KPSS tests pass on first-differenced data series, confirming that second-differencing is not required.

b) usgdp

```
ggtsdisplay(usgdp, main = "usgdp - raw data series")
```



The graph of the raw data shows strong autocorrelation.

```
Box.test(usgdp, type = "Ljung-Box")
```

Ljung-box test on raw data series:

```
##  
## Box-Ljung test  
##  
## data: usgdp  
## X-squared = 233.07, df = 1, p-value < 2.2e-16
```

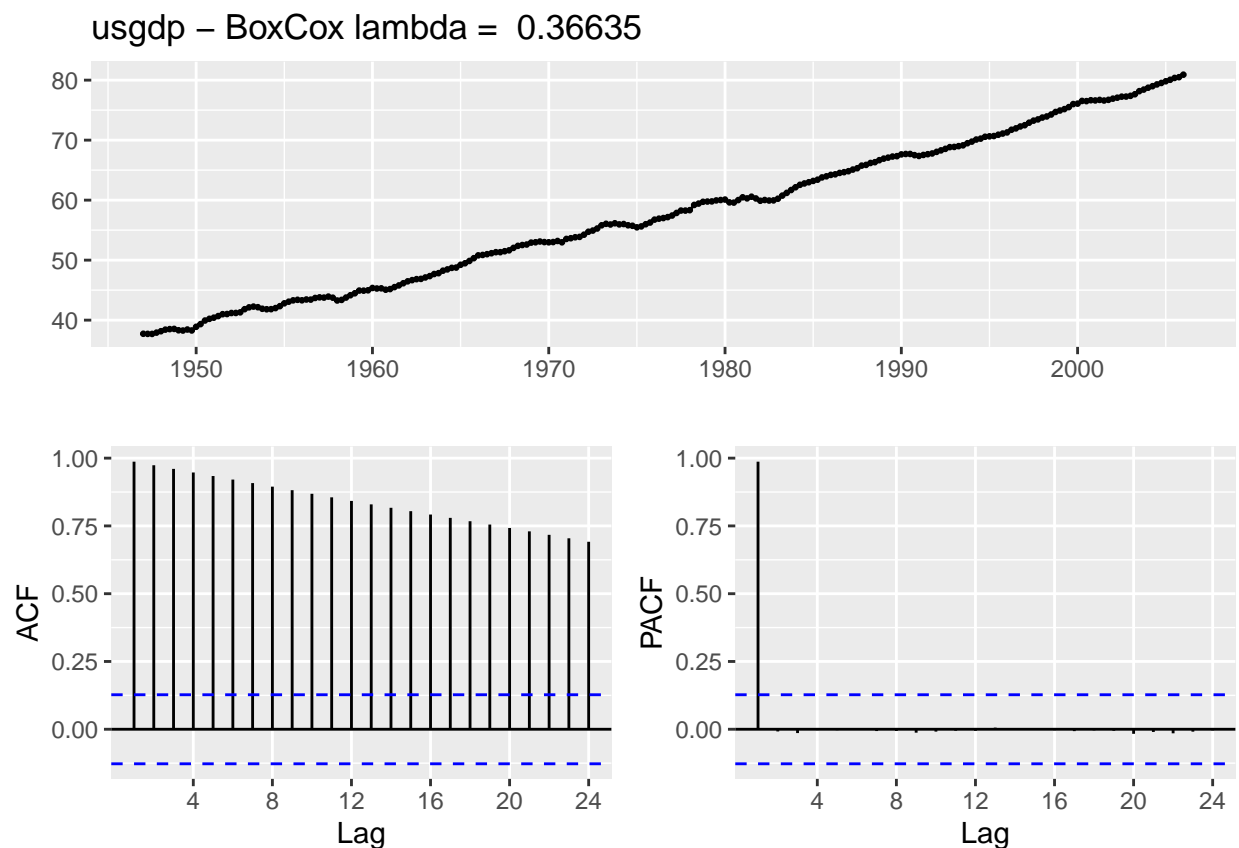
Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

```
usgdp_lambda <- round(BoxCox.lambda(usgdp),5)
print(paste("Lambda for usgdp: ", usgdp_lambda))
```

Try Box-Cox transformation:

```
## [1] "Lambda for usgdp: 0.36635"
```

```
usgdp_BC <- BoxCox(usgdp, usgdp_lambda)
ggtsdisplay(usgdp_BC, main = paste("usgdp - BoxCox lambda = ",usgdp_lambda))
```



The graph of the Box-Cox transformed data shows strong autocorrelation.

```
Box.test(usgdp_BC, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: usgdp_BC
## X-squared = 233.76, df = 1, p-value < 2.2e-16
```

Because the p-value is still low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

```
usgdp_nsdiffs <- nsdiffs(usgdp)
print(paste("Number of SEASONAL differences suggested for usgdp (raw data):", usgdp_nsdiffs))
```

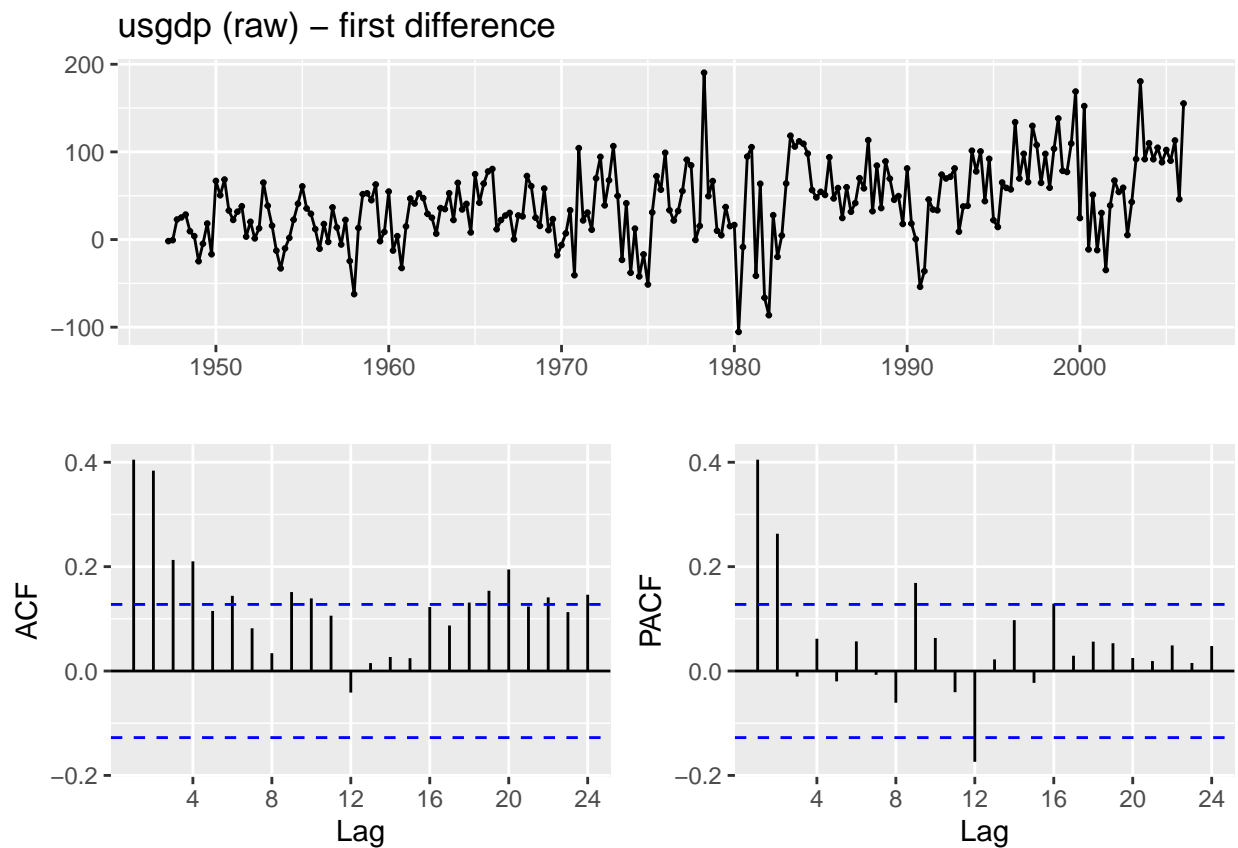
Check estimated number of differences needed:

```
## [1] "Number of SEASONAL differences suggested for usgdp (raw data): 0"
```

```
usgdp_ndiffs <- ndiffs(usgdp)
print(paste("Number of differences suggested for usgdp (raw data):", usgdp_ndiffs))
```

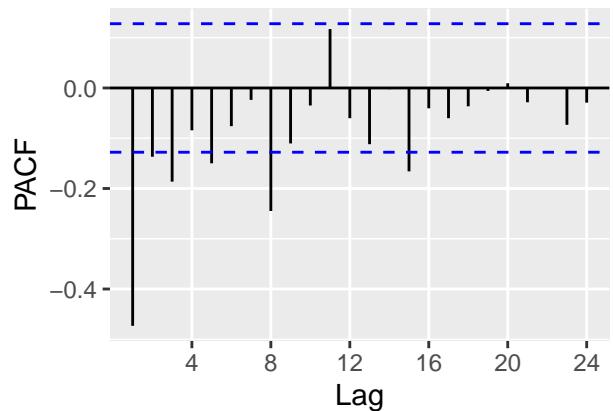
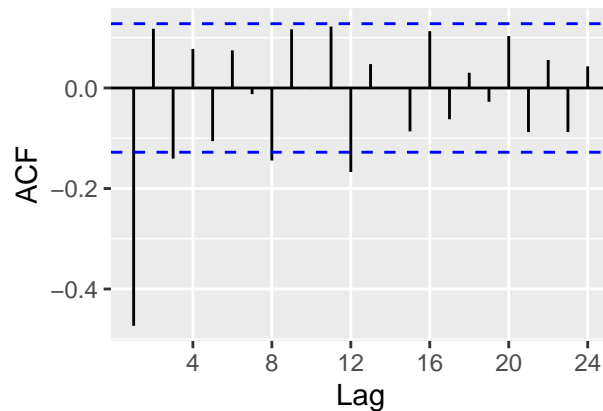
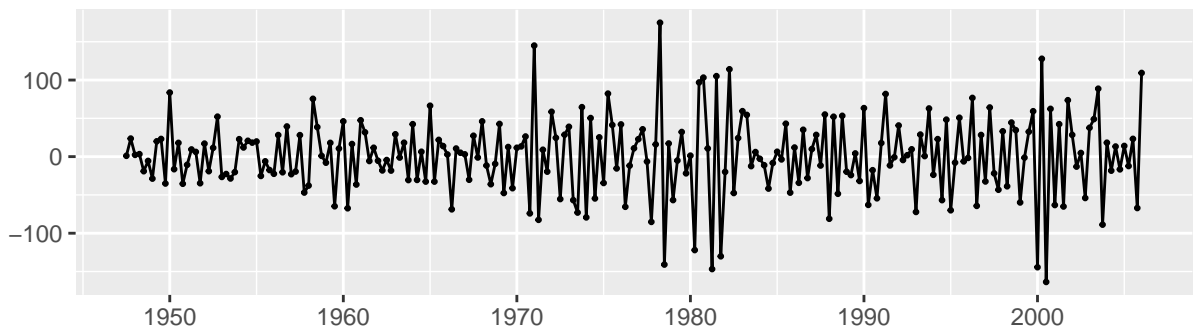
```
## [1] "Number of differences suggested for usgdp (raw data): 2"
```

```
ggtsdisplay(diff(usgdp), main=paste("usgdp (raw) - first difference"))
```



```
ggtsdisplay(diff(diff(usgdp)), main=paste("usgdp (raw) - second difference"))
```

usgdp (raw) – second difference



```
usgdp_BC_nsdiffs <- nsdiffs(usgdp_BC)
print(paste("Number of SEASONAL differences suggested for usgdp_BC:", usgdp_BC_nsdiffs))
```

Check differences estimated for the Box-Cox transformed series:

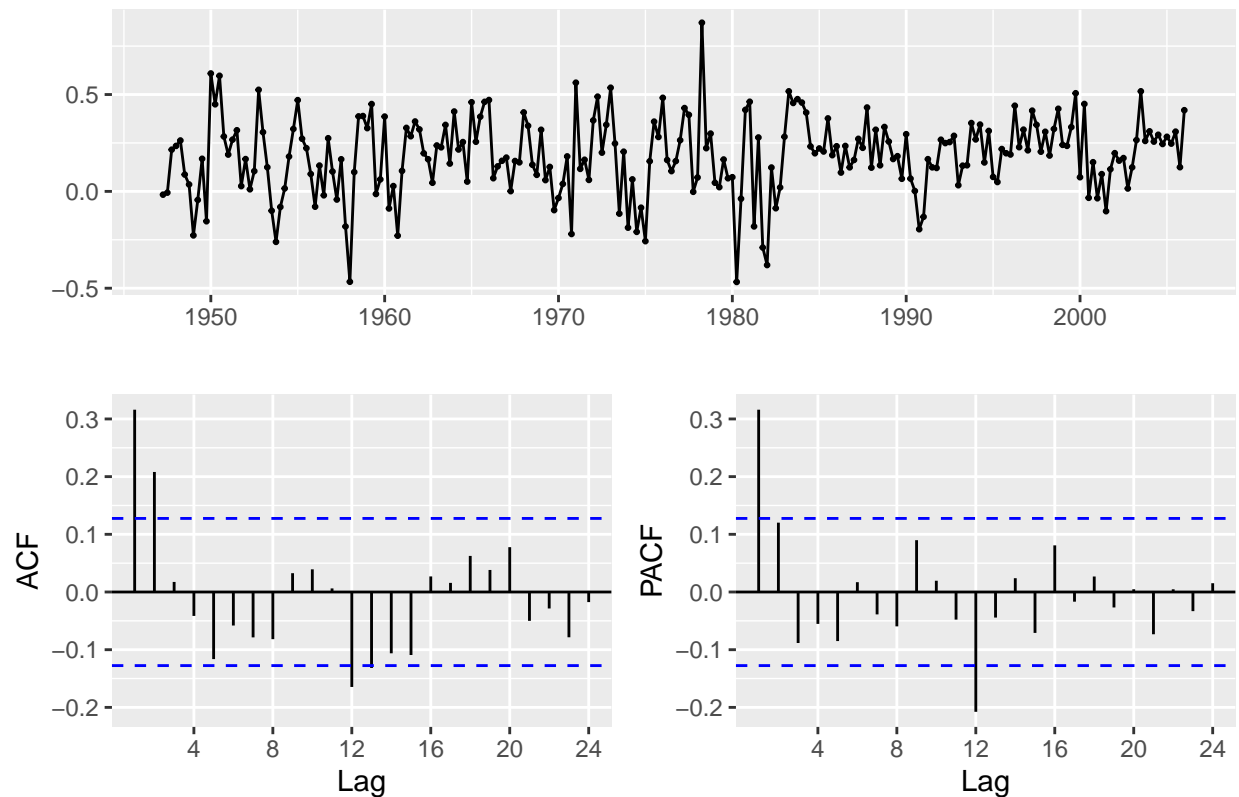
```
## [1] "Number of SEASONAL differences suggested for usgdp_BC: 0"
```

```
usgdp_BC_ndiffs <- ndiffs(usgdp_BC)
print(paste("Number of differences suggested for usgdp_BC:", usgdp_BC_ndiffs))
```

```
## [1] "Number of differences suggested for usgdp_BC: 1"
```

```
ggtssdisplay(diff(usgdp_BC), main=paste("usgdp - BoxCox lambda = ",
                                         usgdp_lambda, " - first difference"))
```

usgdp – BoxCox lambda = 0.36635 – first difference

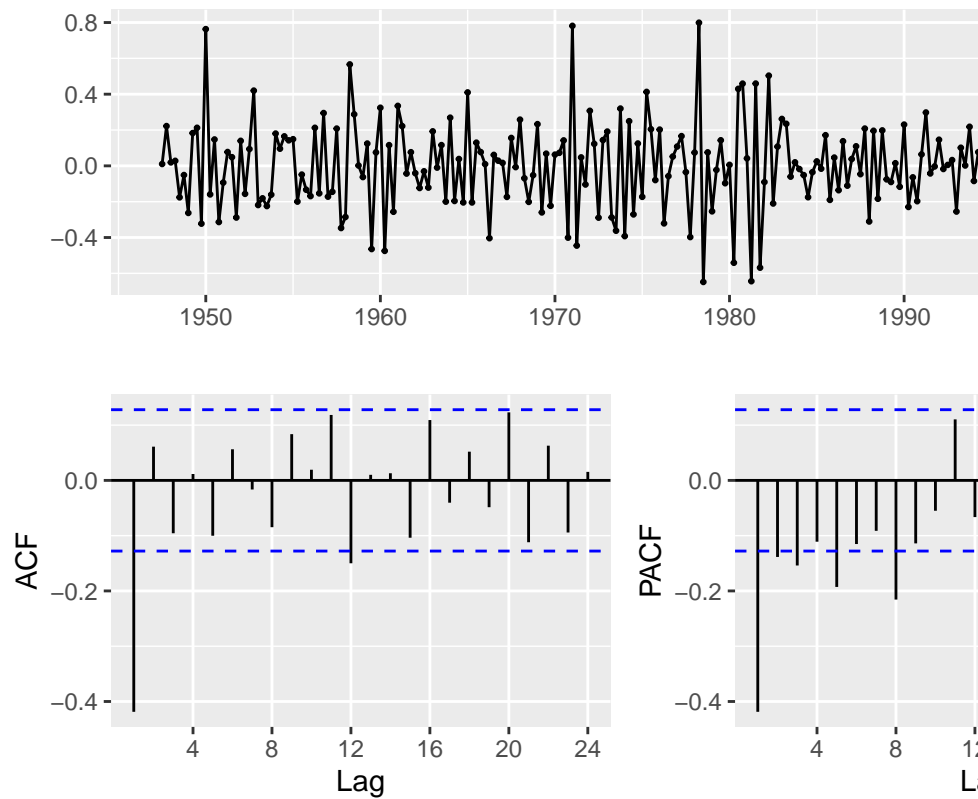


The “nsdiffs” and “ndiffs” functions suggest that no seasonal differencing is required, while the raw data series requires 2 diffs and the Box-Cox transformed series requires 1 diff. However, the plots do not agree, as there still appears to be evidence of autocorrelation.

In particular, there does seem to be autocorrelation at lag 12, suggesting that seasonality does exist.

```
ggtsdisplay(diff(diff(usgdp_BC)), main=paste("usgdp – BoxCox lambda = ",
usgdp_lambda," – second difference"))
```

usgdp – BoxCox lambda = 0.36635 – second difference



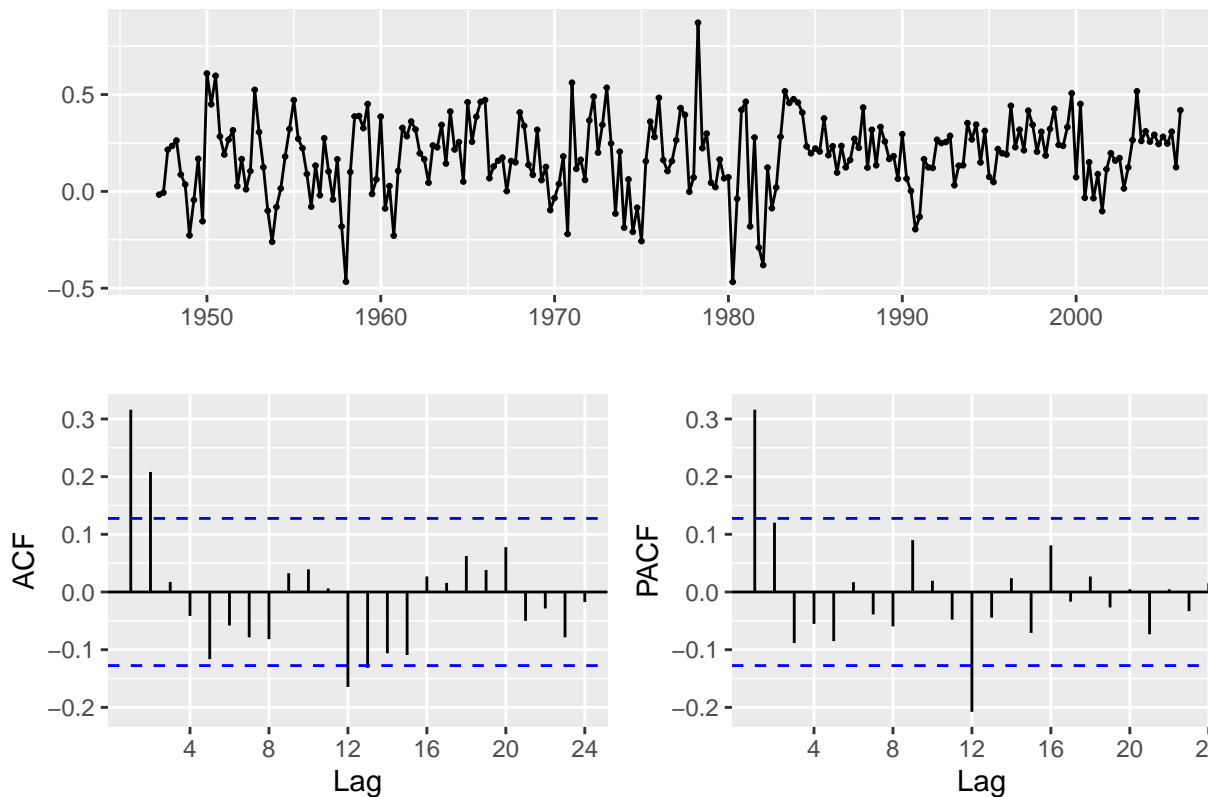
Have a look at second differences:

This shows autocorrelation, too.

Let's evaluate first differencing:

```
usgdp_BC %>% diff() -> usgdp_BC_d1
ggtsdisplay(usgdp_BC_d1, main = paste("usgdp - BoxCox lambda = ",
                                       usgdp_lambda, "first difference"))
```

usgdp – BoxCox lambda = 0.36635 first difference



Diff data series:

The first differenced series shows autocorrelation on lags 1, 2, 12

```
Box.test(usgdp_BC_d1, type = "Ljung-Box")
```

Ljung-box test:

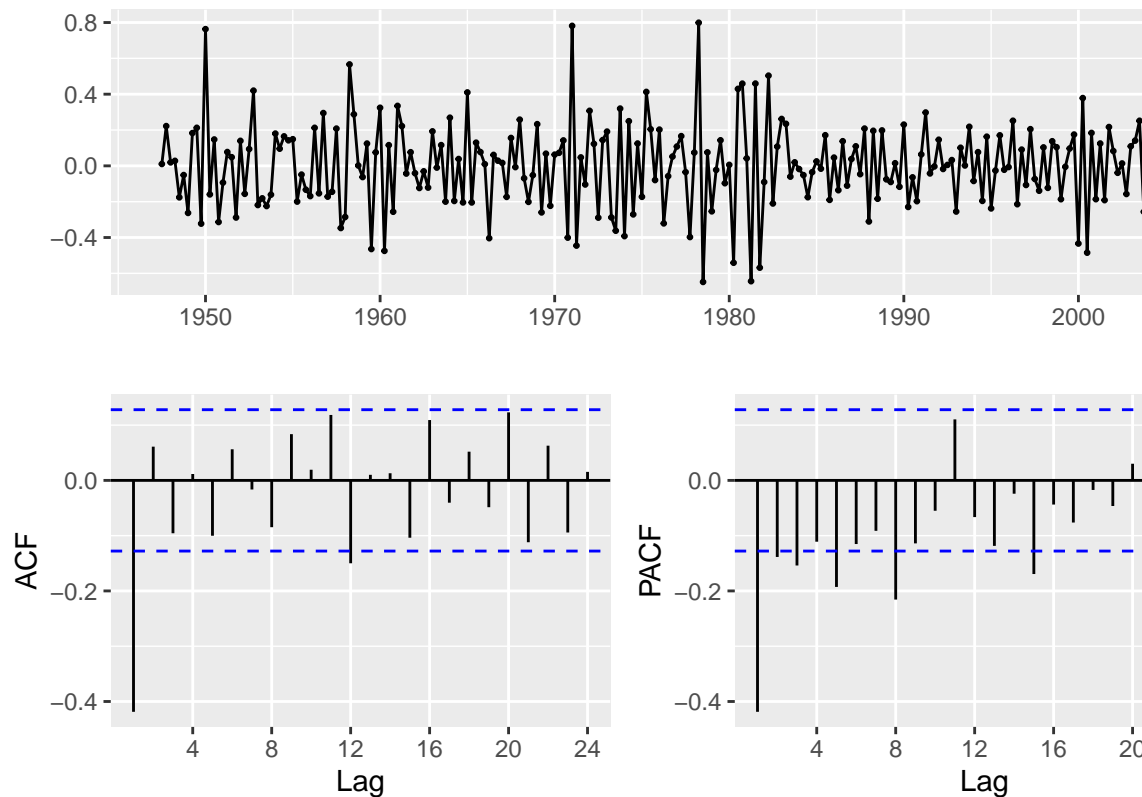
```
##
## Box-Ljung test
##
## data: usgdp_BC_d1
## X-squared = 23.874, df = 1, p-value = 1.029e-06
```

Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

Let's try second differencing:

```
usgdp_BC_d1 %>% diff() -> usgdp_BC_d2
ggtsdisplay(usgdp_BC_d2, main = paste("usgdp - BoxCox lambda = ",
                                     usgdp_lambda, "second difference"))
```

usgdp – BoxCox lambda = 0.36635 second difference



Diff data series again

The second differenced series shows autocorrelation on lags 1, 2, 12

```
Box.test(usgdp_BC_d2, type = "Ljung-Box")
```

Ljung-box test:

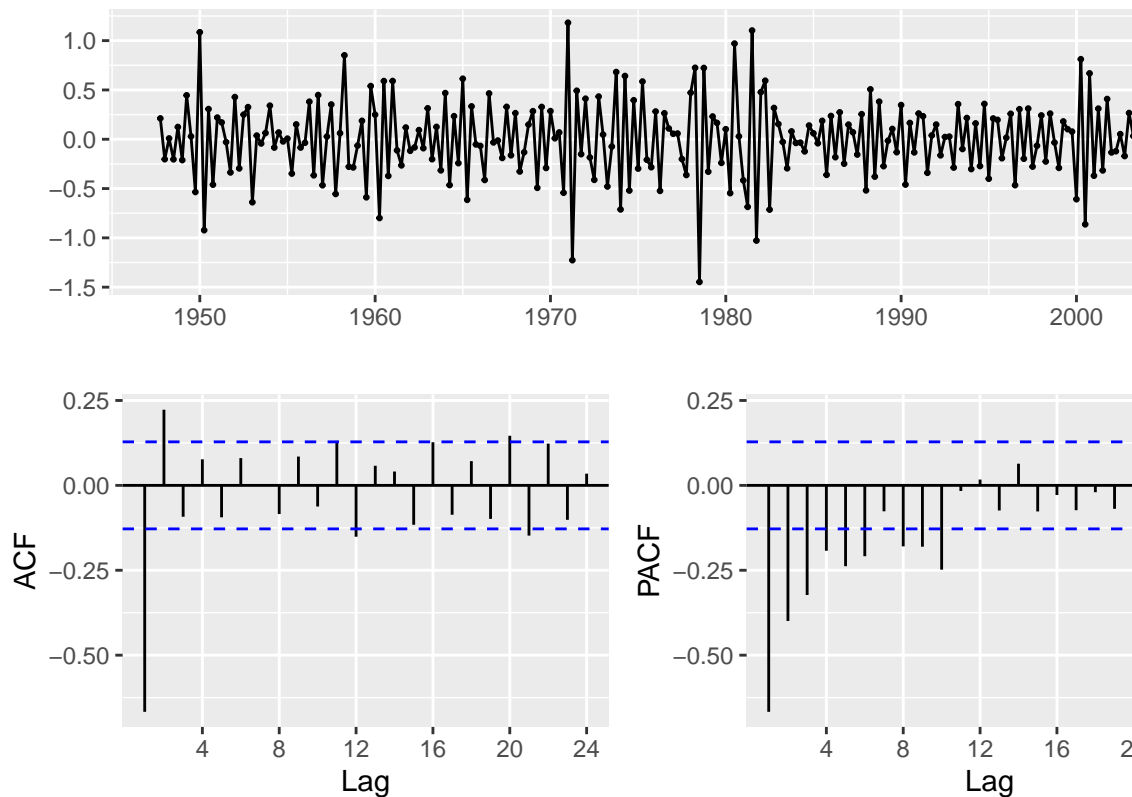
```
##
## Box-Ljung test
##
## data: usgdp_BC_d2
## X-squared = 41.715, df = 1, p-value = 1.056e-10
```

Because the p-value is low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

Let's try third differencing:

```
usgdp_BC_d2 %>% diff() -> usgdp_BC_d3
ggtsdisplay(usgdp_BC_d3, main = paste("usgdp - BoxCox lambda = ",
                                     usgdp_lambda, "third difference"))
```


usgdp – BoxCox lambda = 0.36635 third difference



Diff data series again:

The third differenced series still shows some autocorrelation

```
Box.test(usgdp_BC_d3, type = "Ljung-Box")
```

Ljung-box test:

```
##
## Box-Ljung test
##
## data: usgdp_BC_d3
## X-squared = 105.37, df = 1, p-value < 2.2e-16
```

Because the p-value is low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

```
kpss.test(usgdp, "Level", lshort = F)
```

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
## Warning in kpss.test(usgdp, "Level", lshort = F): p-value smaller than printed
## p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: usgdp
## KPSS Level = 1.6321, Truncation lag parameter = 14, p-value = 0.01

kpss.test(usgdp, "Trend", lshort = F)

## Warning in kpss.test(usgdp, "Trend", lshort = F): p-value smaller than printed
## p-value

##
## KPSS Test for Trend Stationarity
##
## data: usgdp
## KPSS Trend = 0.39501, Truncation lag parameter = 14, p-value = 0.01

usgdp %>% ur.kpss(type="mu", lags="long") %>% summary()

##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 14 lags.
##
## Value of test-statistic is: 1.6321
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

usgdp %>% ur.kpss(type="tau", lags="long") %>% summary()

##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 14 lags.
##
## Value of test-statistic is: 0.395
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216

kpss.test(usgdp_BC, "Level", lshort = F)
```

Both KPSS tests fail on raw data series

```
## Warning in kpss.test(usgdp_BC, "Level", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: usgdp_BC
## KPSS Level = 1.68, Truncation lag parameter = 14, p-value = 0.01
```

```
kpss.test(usgdp_BC, "Trend", lshort = F)
```

```
## Warning in kpss.test(usgdp_BC, "Trend", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: usgdp_BC
## KPSS Trend = 0.24107, Truncation lag parameter = 14, p-value = 0.01
```

```
usgdp_BC %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 14 lags.
##
## Value of test-statistic is: 1.68
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
usgdp_BC %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 14 lags.
##
## Value of test-statistic is: 0.2411
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```

```
kpss.test(usgdp_BC_d1, "Level", lshort = F)
```

Both KPSS tests fail on Box Cox transformed data series

```
## Warning in kpss.test(usgdp_BC_d1, "Level", lshort = F): p-value greater than  
## printed p-value
```

```
##  
## KPSS Test for Level Stationarity  
##  
## data: usgdp_BC_d1  
## KPSS Level = 0.23949, Truncation lag parameter = 14, p-value = 0.1
```

```
kpss.test(usgdp_BC_d1, "Trend", lshort = F)
```

```
## Warning in kpss.test(usgdp_BC_d1, "Trend", lshort = F): p-value greater than  
## printed p-value
```

```
##  
## KPSS Test for Trend Stationarity  
##  
## data: usgdp_BC_d1  
## KPSS Trend = 0.034928, Truncation lag parameter = 14, p-value = 0.1
```

```
usgdp_BC_d1 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: mu with 14 lags.  
##  
## Value of test-statistic is: 0.2395  
##  
## Critical value for a significance level of:  
##          10pct  5pct  2.5pct  1pct  
## critical values 0.347 0.463  0.574 0.739
```

```
usgdp_BC_d1 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: tau with 14 lags.  
##  
## Value of test-statistic is: 0.0349
```

```
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

```
kpss.test(usgdp_BC_d2, "Level", lshort = F)
```

Both KPSS tests pass on first differenced data series

```
## Warning in kpss.test(usgdp_BC_d2, "Level", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: usgdp_BC_d2
## KPSS Level = 0.035084, Truncation lag parameter = 14, p-value = 0.1
```

```
kpss.test(usgdp_BC_d2, "Trend", lshort = F)
```

```
## Warning in kpss.test(usgdp_BC_d2, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: usgdp_BC_d2
## KPSS Trend = 0.034844, Truncation lag parameter = 14, p-value = 0.1
```

```
usgdp_BC_d2 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 14 lags.
##
## Value of test-statistic is: 0.0351
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
usgdp_BC_d2 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 14 lags.
##
## Value of test-statistic is: 0.0348
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

```
kpss.test(usgdp_BC_d3, "Level", lshort = F)
```

Both KPSS tests pass on second differenced data series

```
## Warning in kpss.test(usgdp_BC_d3, "Level", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: usgdp_BC_d3
## KPSS Level = 0.043205, Truncation lag parameter = 14, p-value = 0.1
```

```
kpss.test(usgdp_BC_d3, "Trend", lshort = F)
```

```
## Warning in kpss.test(usgdp_BC_d3, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: usgdp_BC_d3
## KPSS Trend = 0.03361, Truncation lag parameter = 14, p-value = 0.1
```

```
usgdp_BC_d3 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 14 lags.
##
## Value of test-statistic is: 0.0432
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
usgdp_BC_d3 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

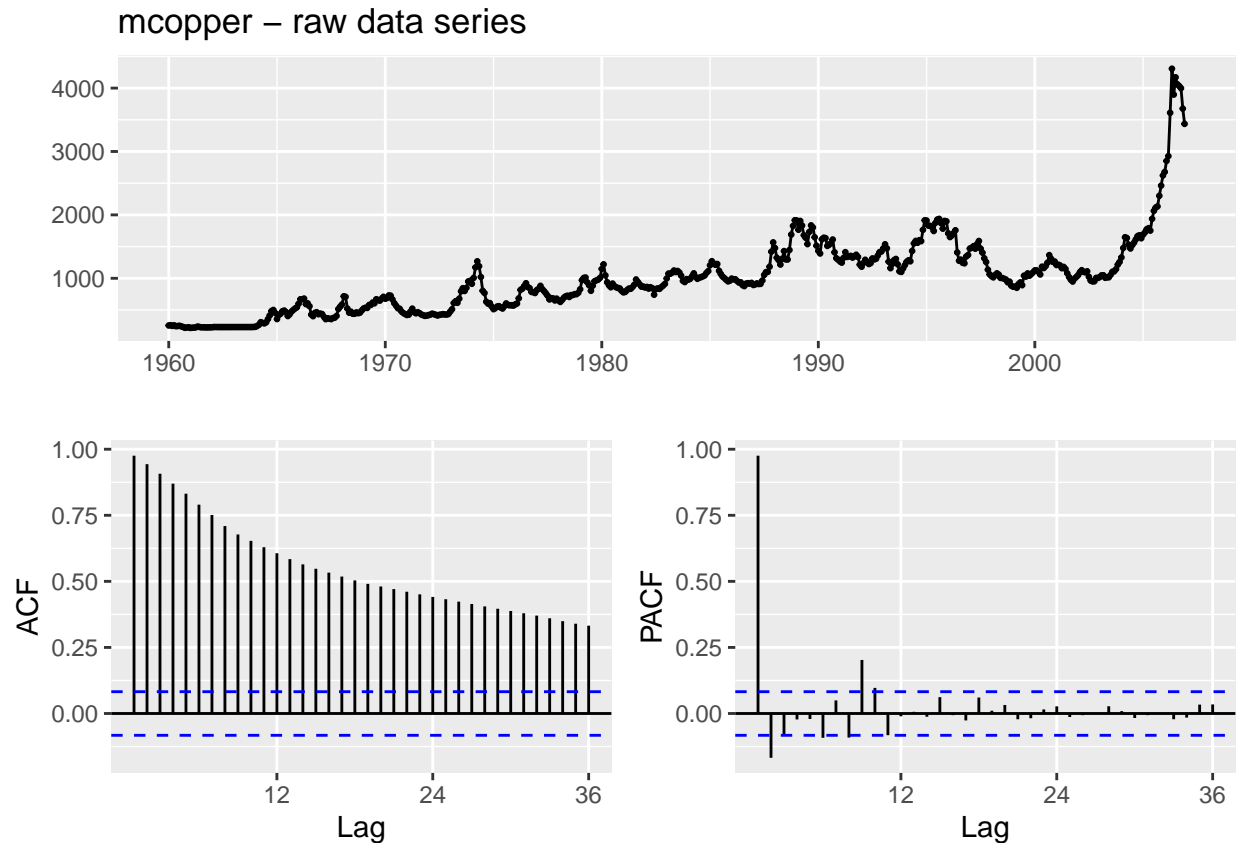
```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: tau with 14 lags.  
##  
## Value of test-statistic is: 0.0336  
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146  0.176 0.216
```

Both KPSS tests pass on third differenced data series

The KPSS test indicates that one difference was sufficient to make the data stationary.

c) mcopper

```
ggtsdisplay(mcopper, main = "mcopper - raw data series")
```



The graph of the raw data shows strong autocorrelation.

```
Box.test(mcopper, type = "Ljung-Box")
```

Ljung-box test on raw data series:

```
##  
## Box-Ljung test  
##  
## data: mcopper  
## X-squared = 539.54, df = 1, p-value < 2.2e-16
```

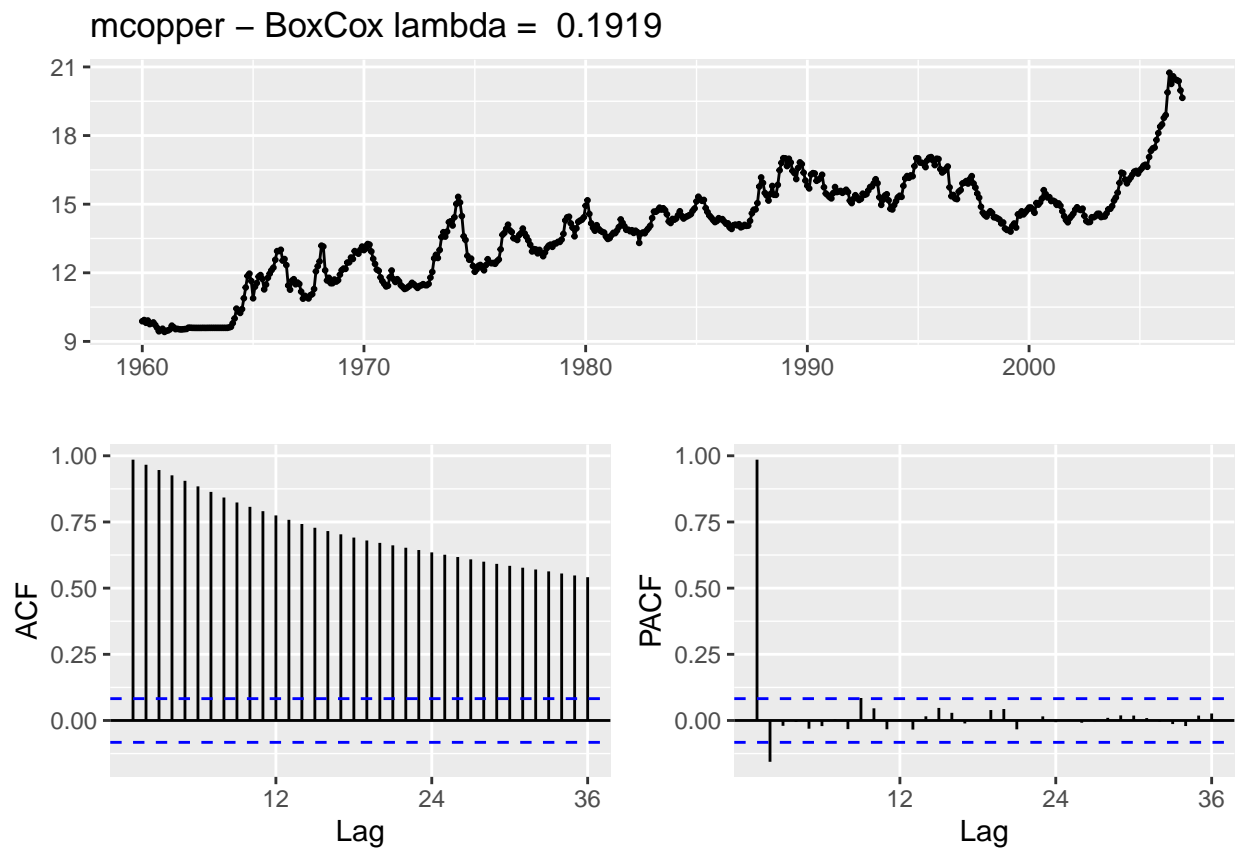
Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)


```
mcopper_lambda <- round(BoxCox.lambda(mcopper),5)
print(paste("Lambda for mcopper: ", mcopper_lambda))
```

Try Box-Cox transformation:

```
## [1] "Lambda for mcopper: 0.1919"
```

```
mcopper_BC <- BoxCox(mcopper, mcopper_lambda)
ggtsdisplay(mcopper_BC, main = paste("mcopper - BoxCox lambda = ",mcopper_lambda))
```



The graph of the Box-Cox transformed data shows strong autocorrelation.

```
Box.test(mcopper_BC, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: mcopper_BC
## X-squared = 550.42, df = 1, p-value < 2.2e-16
```

Because the p-value is still low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

```
mcopper_nsdiffs <- nsdiffs(mccopper)
print(paste("Number of SEASONAL differences suggested for mccopper (raw data):", mcopper_nsdiffs))
```

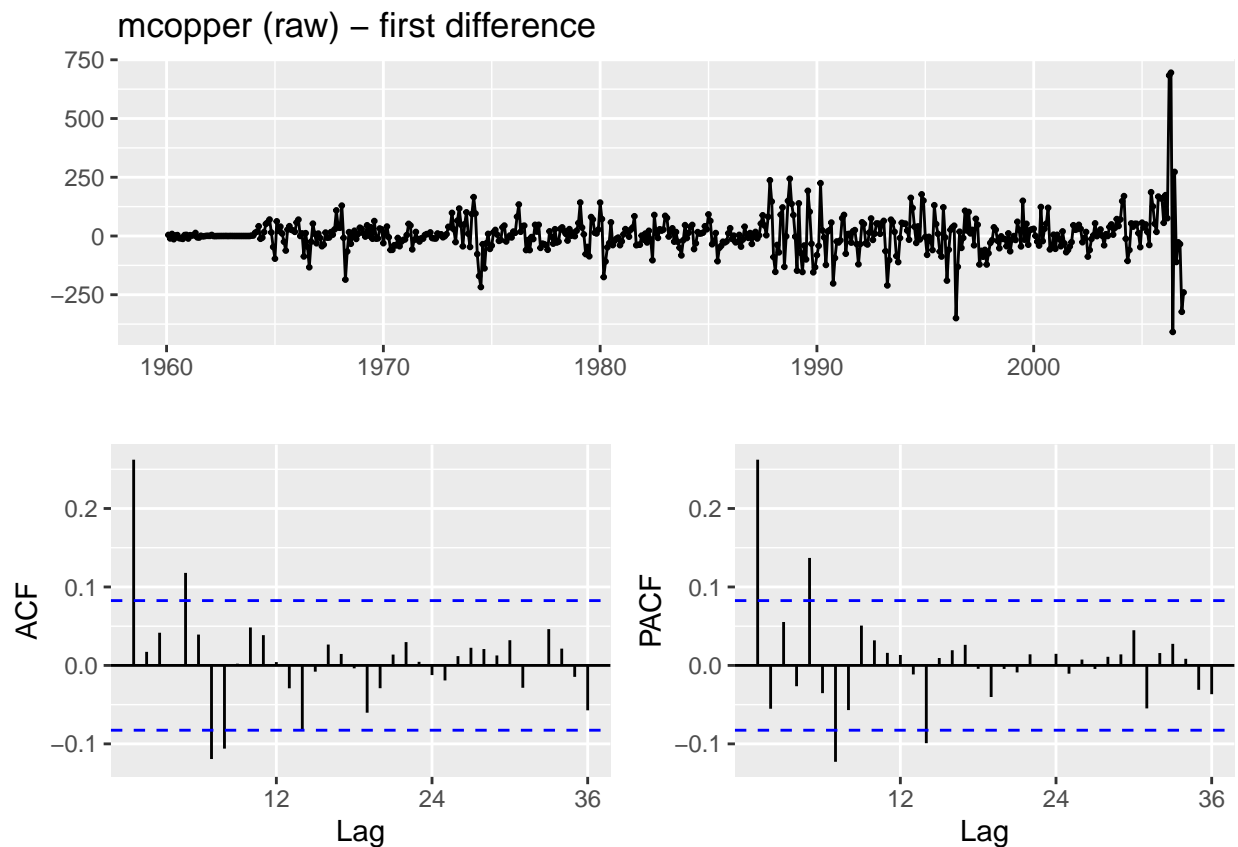
Check estimated number of differences needed:

```
## [1] "Number of SEASONAL differences suggested for mccopper (raw data): 0"
```

```
mcopper_ndiffs <- ndiffs(mccopper)
print(paste("Number of differences suggested for mccopper (raw data):", mcopper_ndiffs))
```

```
## [1] "Number of differences suggested for mccopper (raw data): 1"
```

```
ggtsdisplay(diff(mccopper), main=paste("mccopper (raw) - first difference"))
```



The graphs show that differencing without the Box-Cox transformation is not adequate to achieve stationarity.

```
mcopper_BC_nsdiffs <- nsdiffs(mccopper_BC)
print(paste("Number of SEASONAL differences suggested for mccopper_BC:", mcopper_BC_nsdiffs))
```

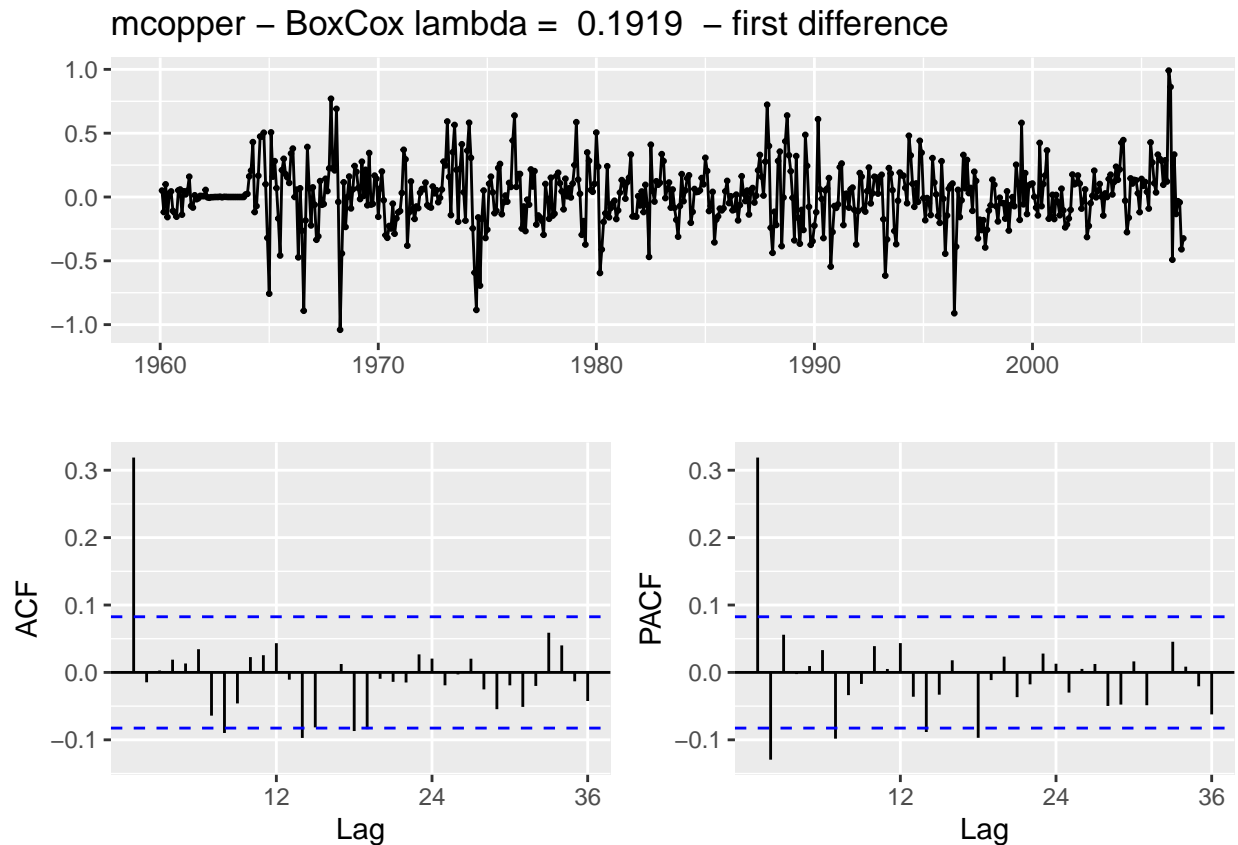
Check differences estimated for the Box-Cox transformed series:

```
## [1] "Number of SEASONAL differences suggested for mcopper_BC: 0"
```

```
mcopper_BC_ndiffs <- ndiffs(mcopper_BC)
print(paste("Number of differences suggested for mcopper_BC:", mcopper_BC_ndiffs))
```

```
## [1] "Number of differences suggested for mcopper_BC: 1"
```

```
ggtsdisplay(diff(mcopper_BC), main=paste("mcopper - BoxCox lambda = ",
                                         mcopper_lambda," - first difference"))
```

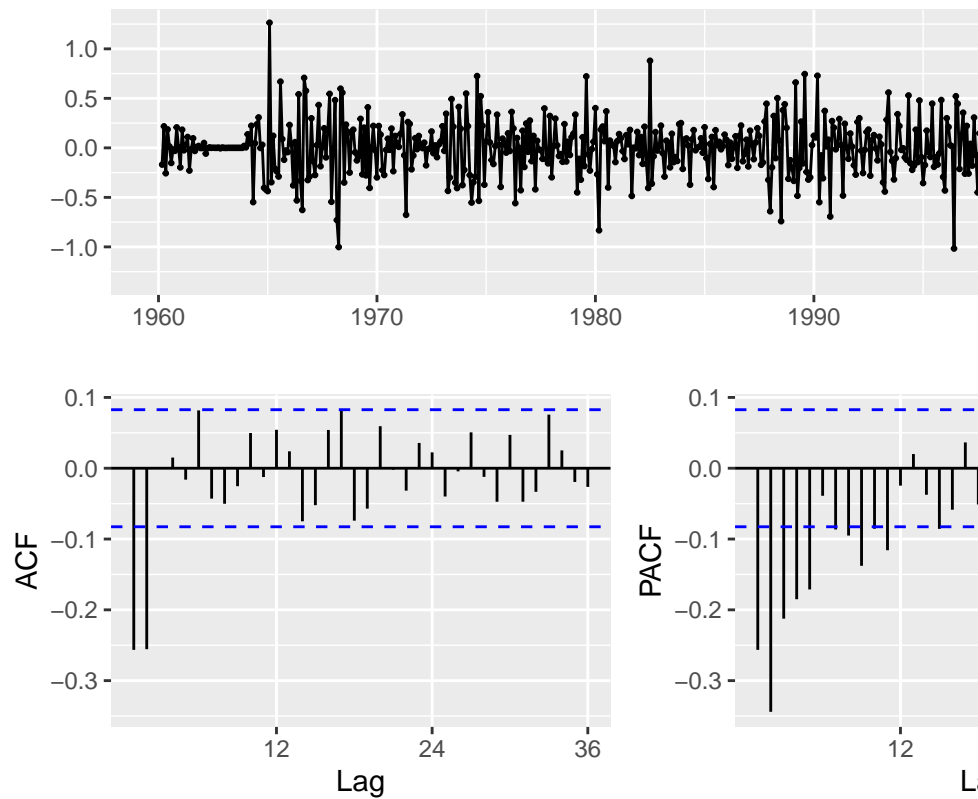


The “nsdiffs” and “ndiffs” functions suggest that no seasonal differencing is required. The raw data series and the Box-Cox transformed series each require 1 diff.

However, the plots do not agree, as there still appears to be evidence of autocorrelation, but the result following the Box-Cox transformation looks better.

```
ggtsdisplay(diff(diff(mcopper_BC)), main=paste("mcopper - BoxCox lambda = ",
                                                mcopper_lambda," - second difference"))
```

mcopper – BoxCox lambda = 0.1919 – second difference

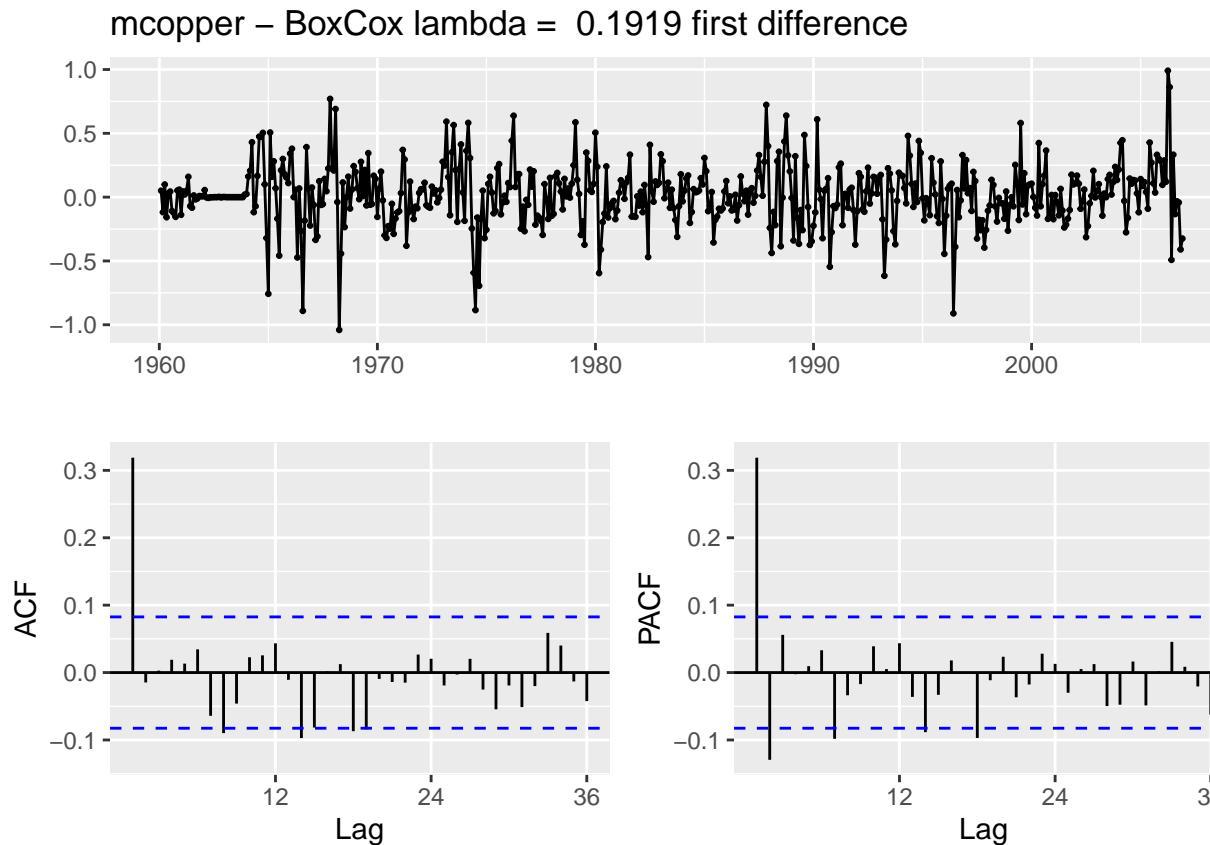


Have a look at second differences:

This looks worse than the result of first differences.

Let's evaluate first differencing:

```
mcopper_BC %>% diff() -> mcopper_BC_d1
ggtsdisplay(mcopper_BC_d1, main = paste("mcopper - BoxCox lambda = ",
                                         mcopper_lambda, "first difference"))
```



Diff data series:

The first differenced series shows autocorrelation on lags 1, and a few of the subsequent lags breach the critical value.

```
Box.test(mcopper_BC_d1, type = "Ljung-Box")
```

Ljung-box test:

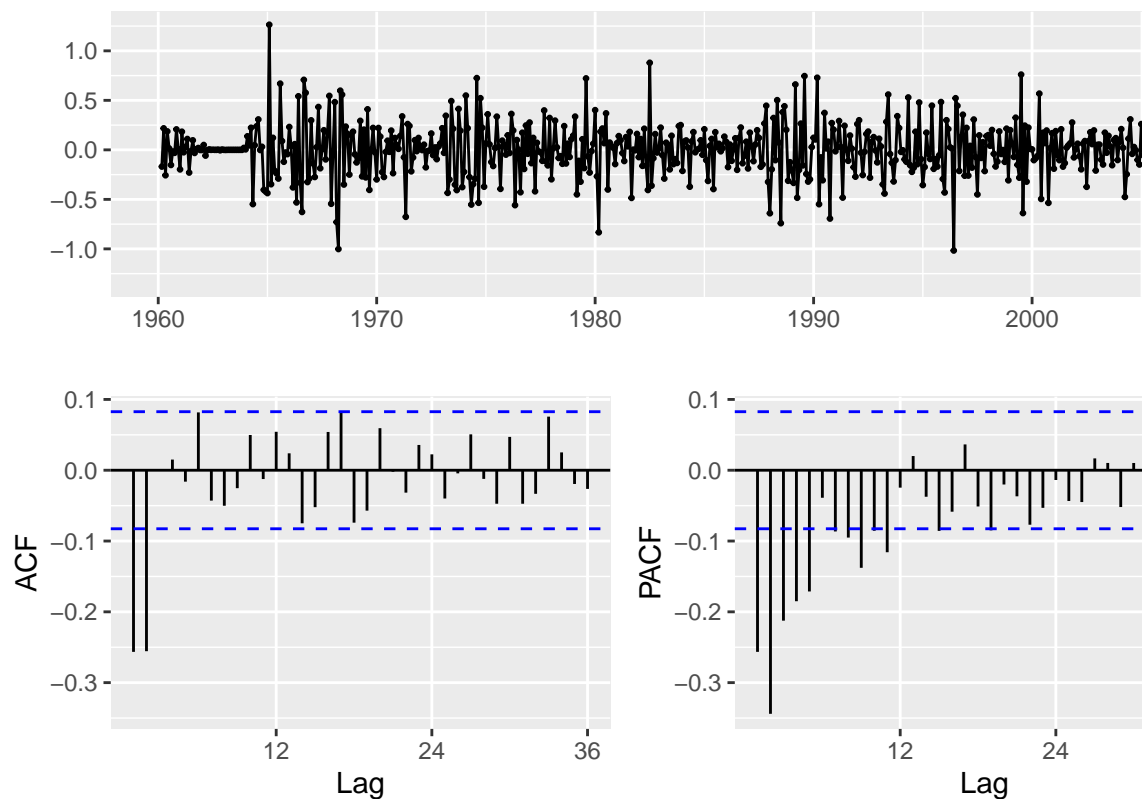
```
##
## Box-Ljung test
##
## data: mcopper_BC_d1
## X-squared = 57.518, df = 1, p-value = 3.353e-14
```

Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

Let's try second differencing:

```
mcopper_BC_d1 %>% diff() -> mcopper_BC_d2
ggtsdisplay(mcopper_BC_d2, main = paste("mcopper – BoxCox lambda = ",
                                         mcopper_lambda, "second difference"))
```

mcopper – BoxCox lambda = 0.1919 second difference



Diff data series again

The second differenced series shows autocorrelation on lags 1 and 2

```
Box.test(mcopper_BC_d2, type = "Ljung-Box")
```

Ljung-box test:

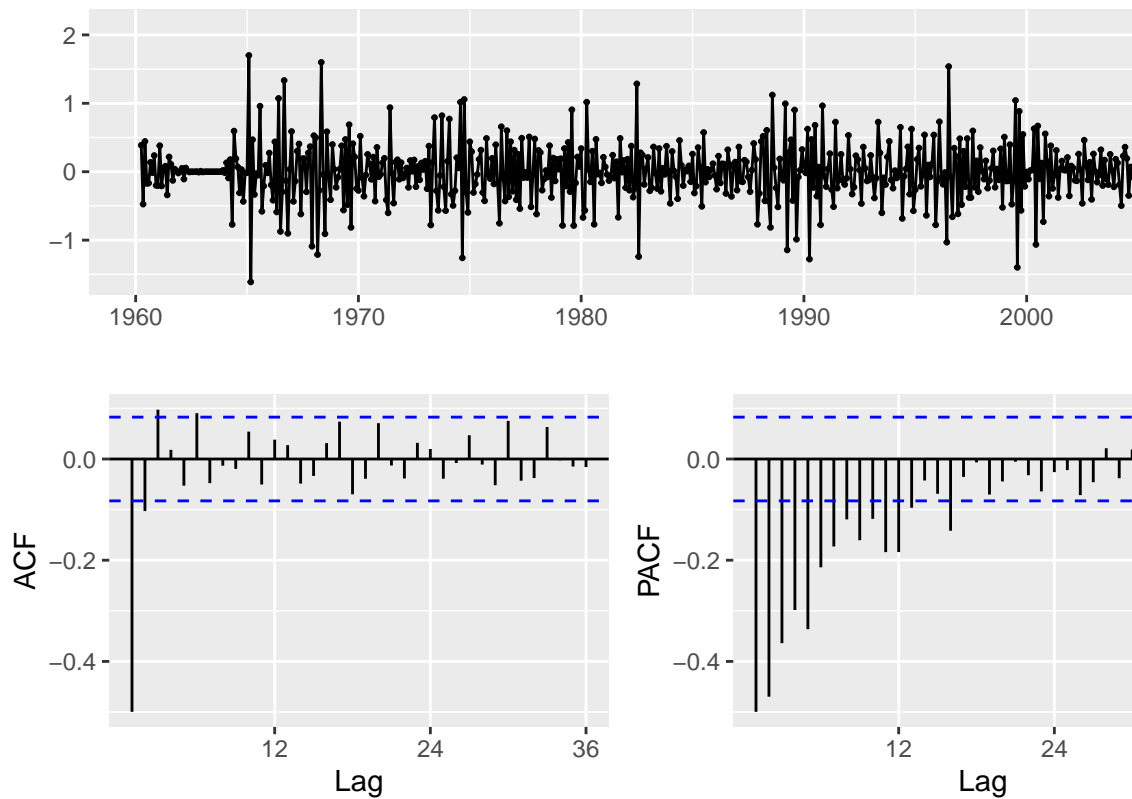
```
##
## Box-Ljung test
##
## data: mcopper_BC_d2
## X-squared = 37.178, df = 1, p-value = 1.078e-09
```

Because the p-value is low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

Let's try third differencing:

```
mcopper_BC_d2 %>% diff() -> mcopper_BC_d3
ggtsdisplay(mcopper_BC_d3, main = paste("mcopper - BoxCox lambda = ",
                                         mcopper_lambda, "third difference"))
```

mcopper – BoxCox lambda = 0.1919 third difference



Diff data series again:

The third differenced series still shows some autocorrelation

```
Box.test(mcopper_BC_d3, type = "Ljung-Box")
```

Ljung-box test:

```
##  
## Box-Ljung test  
##  
## data: mcopper_BC_d3  
## X-squared = 140.77, df = 1, p-value < 2.2e-16
```

Because the p-value is low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

```
kpss.test(mcopper, "Level", lshort = F)
```

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
## Warning in kpss.test(mcopper, "Level", lshort = F): p-value smaller than printed  
## p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: mcopper
## KPSS Level = 2.1648, Truncation lag parameter = 18, p-value = 0.01

kpss.test(mcopper, "Trend", lshort = F)

## Warning in kpss.test(mcopper, "Trend", lshort = F): p-value greater than printed
## p-value

##
## KPSS Test for Trend Stationarity
##
## data: mcopper
## KPSS Trend = 0.068656, Truncation lag parameter = 18, p-value = 0.1

mcopper %>% ur.kpss(type="mu", lags="long") %>% summary()

##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 18 lags.
##
## Value of test-statistic is: 2.1648
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739

mcopper %>% ur.kpss(type="tau", lags="long") %>% summary()

##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 18 lags.
##
## Value of test-statistic is: 0.0687
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216

kpss.test(mcopper_BC, "Level", lshort = F)
```

The KPSS tests on the raw data series fail on Level, but pass on Trend


```
## Warning in kpss.test(mccopper_BC, "Level", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data:  mccopper_BC
## KPSS Level = 2.5113, Truncation lag parameter = 18, p-value = 0.01
```

```
kpss.test(mccopper_BC, "Trend", lshort = F)
```

```
## Warning in kpss.test(mccopper_BC, "Trend", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data:  mccopper_BC
## KPSS Trend = 0.2472, Truncation lag parameter = 18, p-value = 0.01
```

```
mccopper_BC %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 18 lags.
##
## Value of test-statistic is: 2.5113
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
mccopper_BC %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 18 lags.
##
## Value of test-statistic is: 0.2472
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

```
kpss.test(mccopper_BC_d1, "Level", lshort = F)
```

Both KPSS tests fail on Box Cox transformed data series

```
## Warning in kpss.test(mccopper_BC_d1, "Level", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data:  mccopper_BC_d1
## KPSS Level = 0.062912, Truncation lag parameter = 18, p-value = 0.1
```

```
kpss.test(mccopper_BC_d1, "Trend", lshort = F)
```

```
## Warning in kpss.test(mccopper_BC_d1, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data:  mccopper_BC_d1
## KPSS Trend = 0.057247, Truncation lag parameter = 18, p-value = 0.1
```

```
mccopper_BC_d1 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 18 lags.
##
## Value of test-statistic is: 0.0629
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
mccopper_BC_d1 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 18 lags.
##
## Value of test-statistic is: 0.0572
```

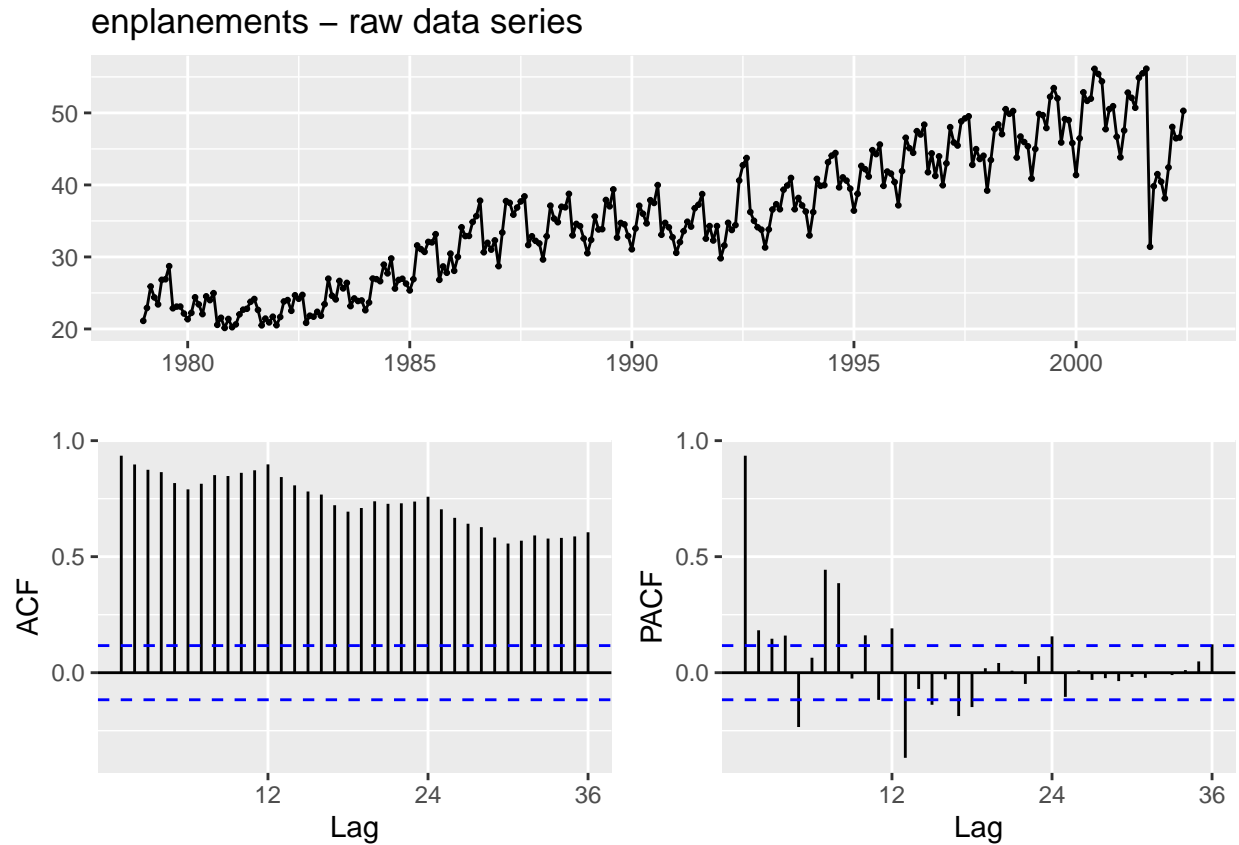
```
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146  0.176 0.216
```

Both KPSS tests pass on first differenced data series

The KPSS test indicates that one difference was sufficient to make the `mcopper` data stationary.

d) enplanements

```
ggtsdisplay(enplanements, main = "enplanements - raw data series")
```



The graph of the raw data shows strong autocorrelation as well as seasonality.

```
Box.test(enplanements, type = "Ljung-Box")
```

Ljung-box test on raw data series:

```
##  
## Box-Ljung test  
##  
## data: enplanements  
## X-squared = 249.22, df = 1, p-value < 2.2e-16
```

Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

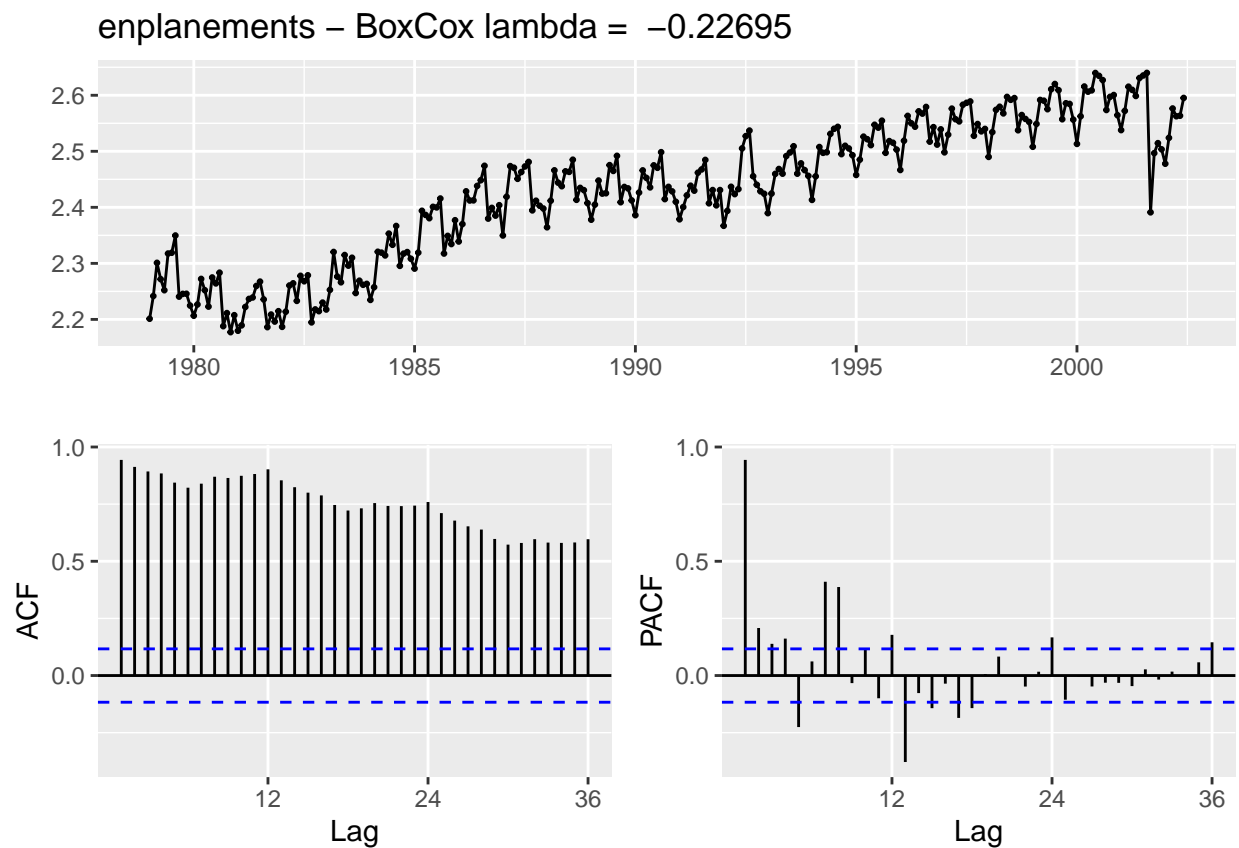
```
enplanements_lambda <- round(BoxCox.lambda(enplanements),5)
print(paste("Lambda for enplanements: ", enplanements_lambda))
```

Try Box-Cox transformation:

```
## [1] "Lambda for enplanements: -0.22695"
```

```
enplanements_BC <- BoxCox(enplanements, enplanements_lambda)

ggtsdisplay(enplanements_BC, main = paste("enplanements - BoxCox lambda = ",
                                          enplanements_lambda))
```



The graph of the Box-Cox transformed data still shows strong autocorrelation and seasonality, but the variance has been dampened.

```
Box.test(enplanements_BC, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: enplanements_BC
## X-squared = 253.7, df = 1, p-value < 2.2e-16
```

Because the p-value is still low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

```
enplanements_nsdiffs <- nsdiffs(enplanements)
print(paste("Number of SEASONAL differences suggested for enplanements (raw data):",
            enplanements_nsdiffs))
```

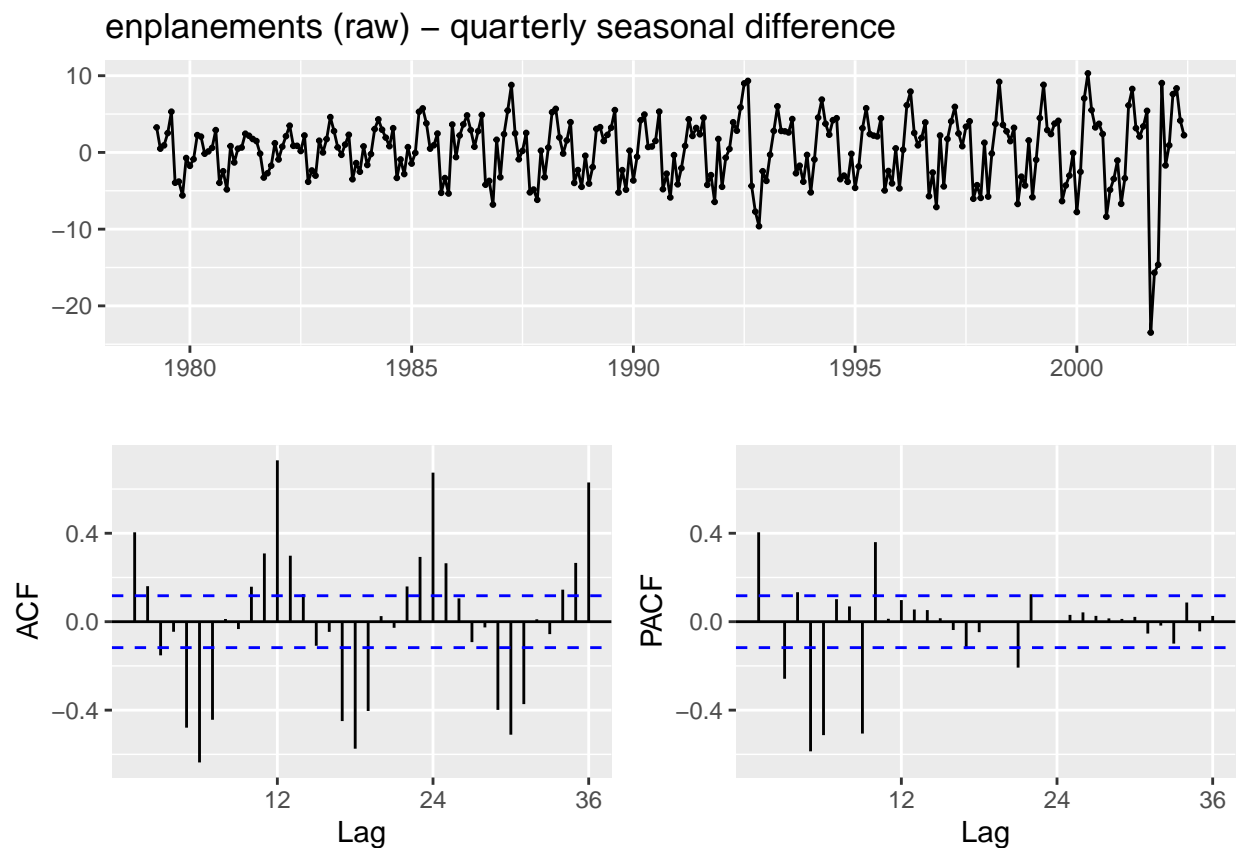
Check estimated number of differences needed:

```
## [1] "Number of SEASONAL differences suggested for enplanements (raw data): 1"
```

```
enplanements_ndiffs <- ndiffs(enplanements)
print(paste("Number of differences suggested for enplanements (raw data):",
            enplanements_ndiffs))
```

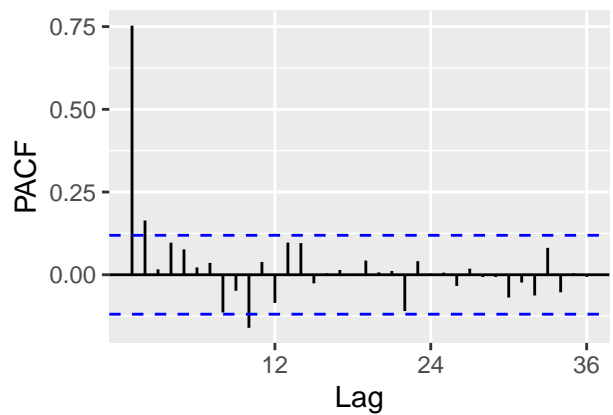
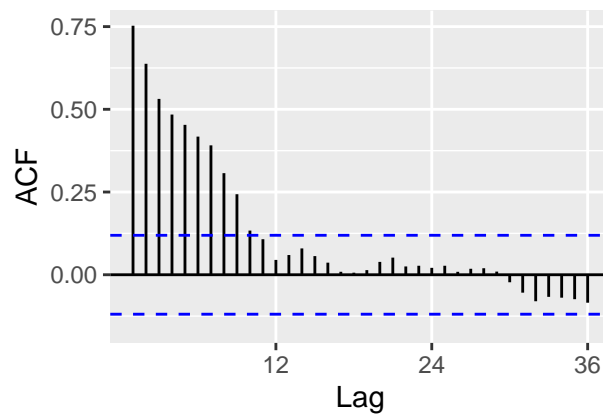
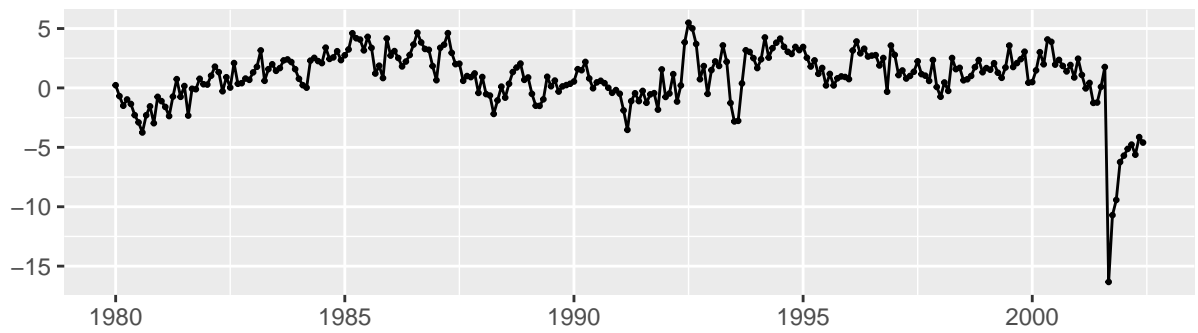
```
## [1] "Number of differences suggested for enplanements (raw data): 1"
```

```
ggtsdisplay(diff(enplanements,lag=3),
            main=paste("enplanements (raw) - quarterly seasonal difference"))
```



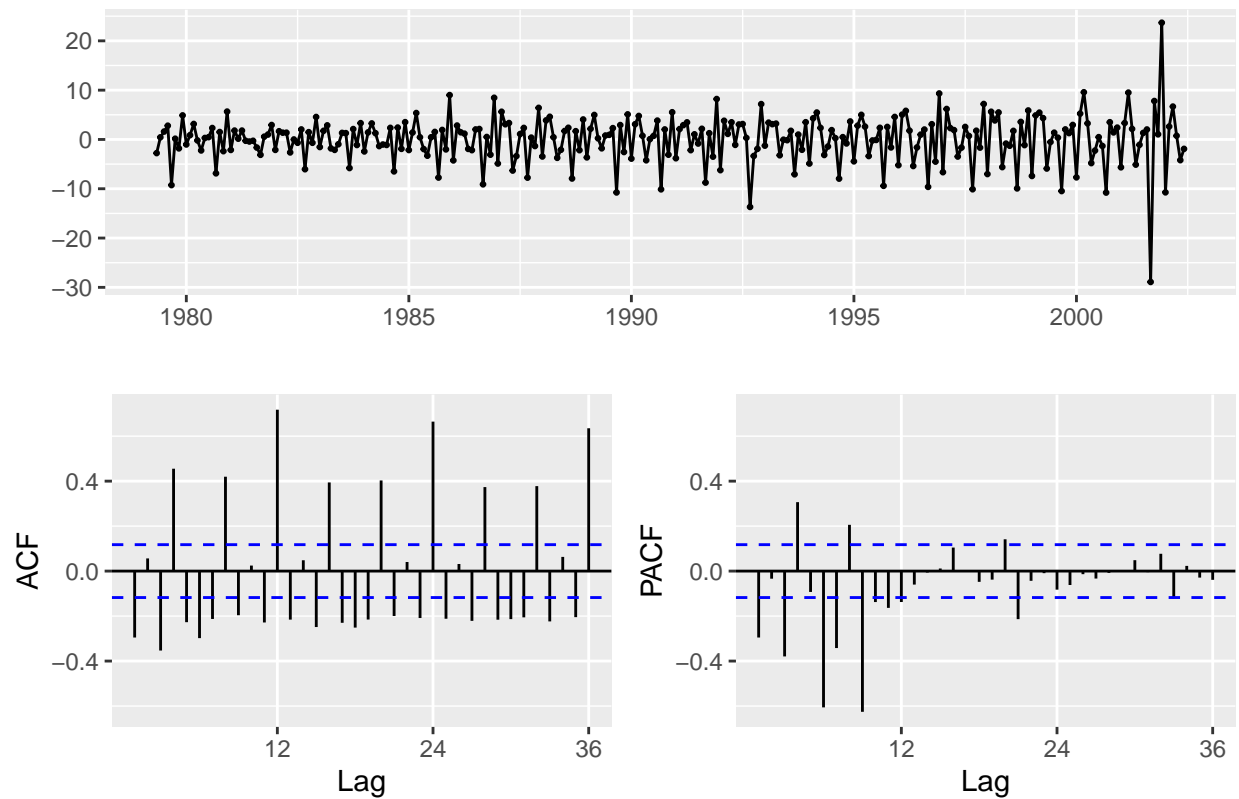
```
ggtsdisplay(diff(enplanements,lag=12),
            main=paste("enplanements (raw) - annual seasonal difference"))
```

enplanements (raw) – annual seasonal difference



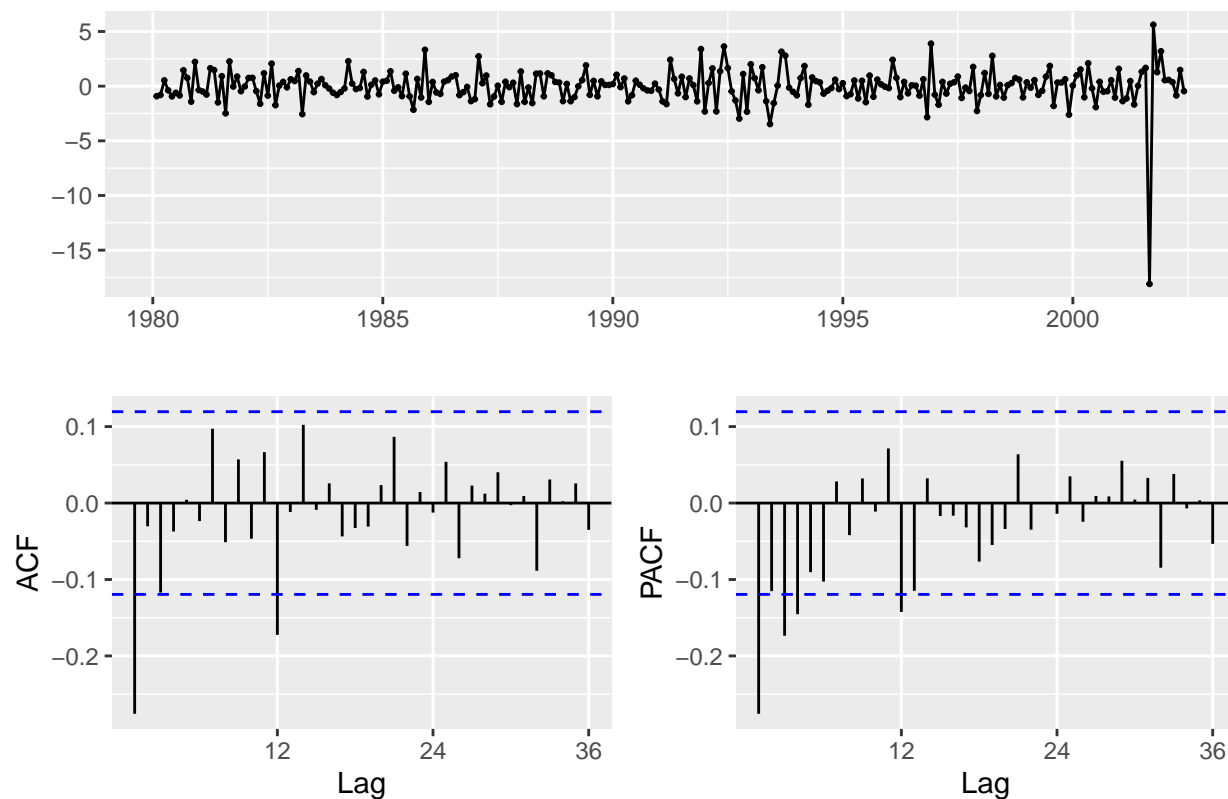
```
ggtsdisplay(diff(diff(enplanements,lag=3),lag=1),
  main=paste("enplanements (raw) - quarterly seasonal difference + first diff"))
```

enplanements (raw) – quarterly seasonal difference + first diff



```
ggtsdisplay(diff(diff(enplanements,lag=12),lag=1),  
            main=paste("enplanements (raw) – annual seasonal difference + first diff"))
```


enplanements (raw) – annual seasonal difference + first diff



The graphs show that differencing without the Box-Cox transformation is not adequate to achieve stationarity

```
enplanements_BC_nsdiffs <- nsdiffs(enplanements_BC)
print(paste("Number of SEASONAL differences suggested for enplanements_BC:", enplanements_BC_nsdiffs))
```

Check differences estimated for the Box-Cox transformed series:

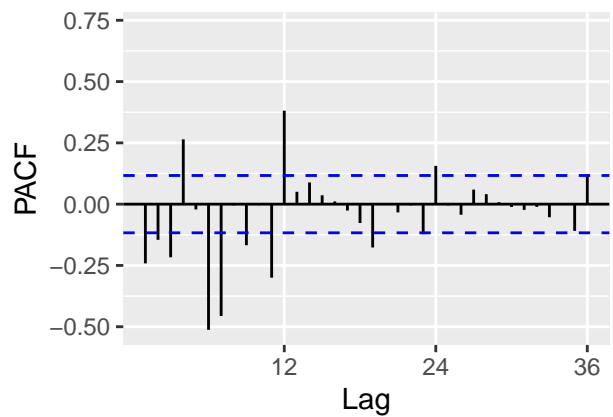
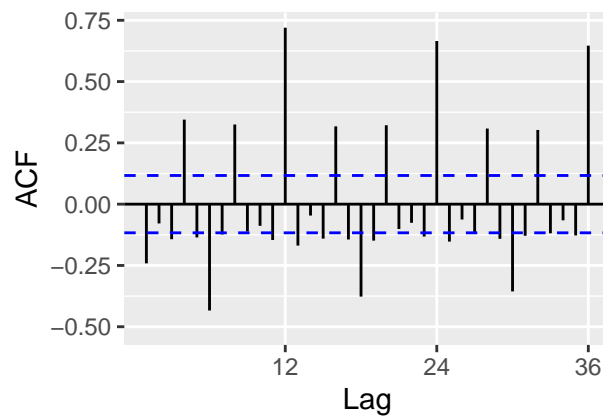
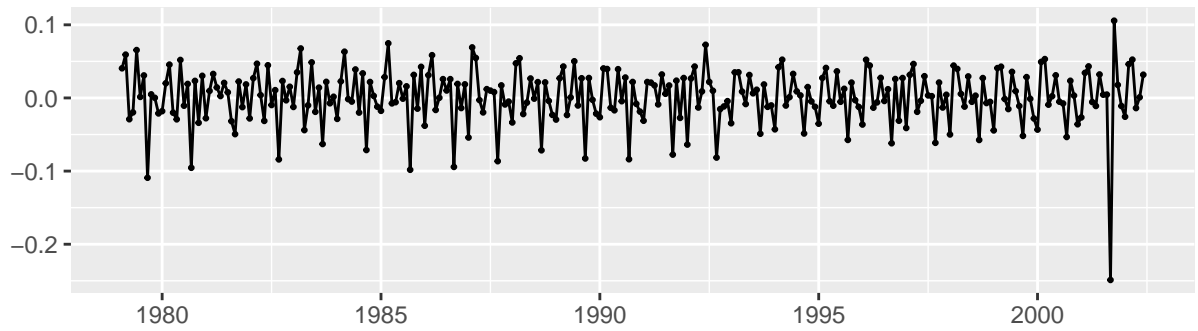
```
## [1] "Number of SEASONAL differences suggested for enplanements_BC: 1"
```

```
enplanements_BC_ndiffs <- ndiffs(enplanements_BC)
print(paste("Number of differences suggested for enplanements_BC:",
  enplanements_BC_ndiffs))
```

```
## [1] "Number of differences suggested for enplanements_BC: 1"
```

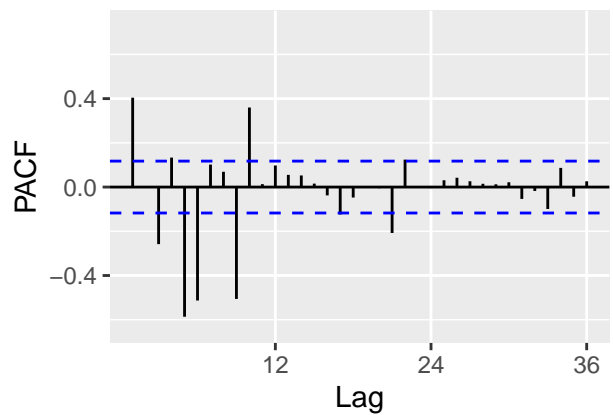
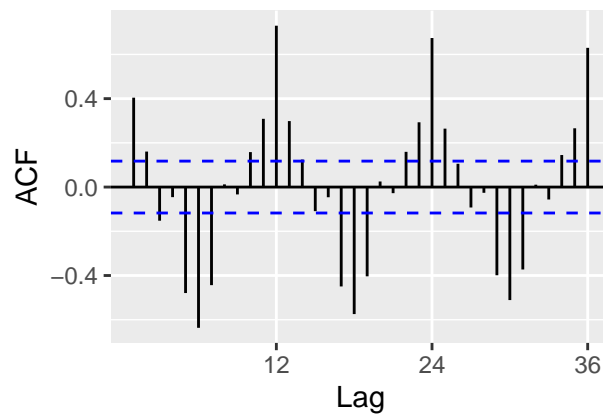
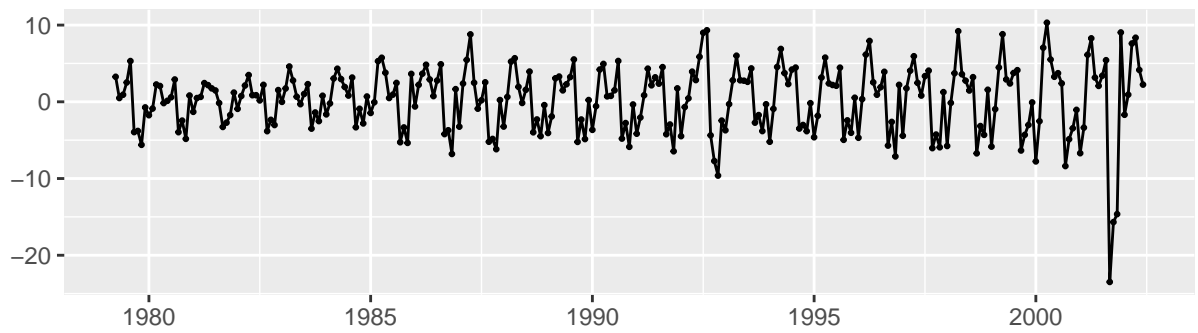
```
ggtstdisplay(diff(enplanements_BC),
  main=paste("enplanements - BoxCox lambda = ",
    enplanements_lambda, " - first difference"))
```

enplanements – BoxCox lambda = -0.22695 – first difference



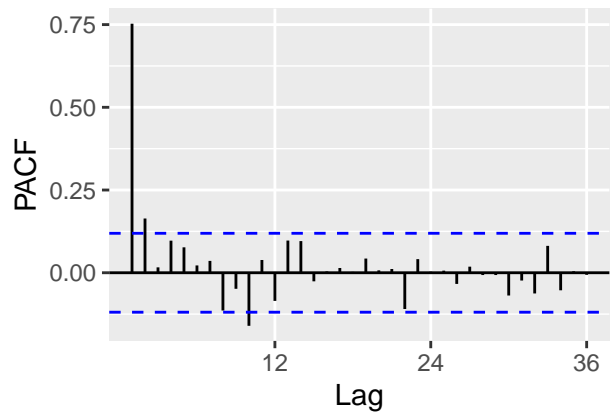
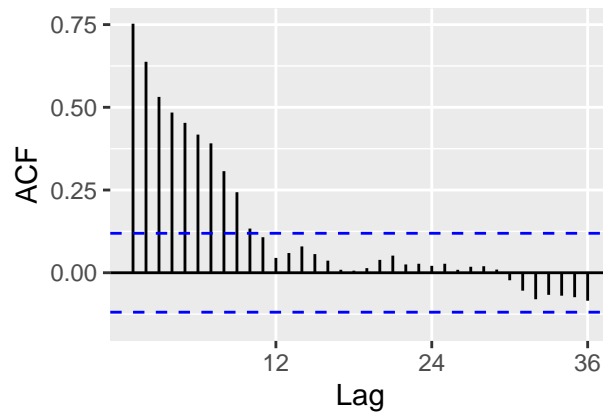
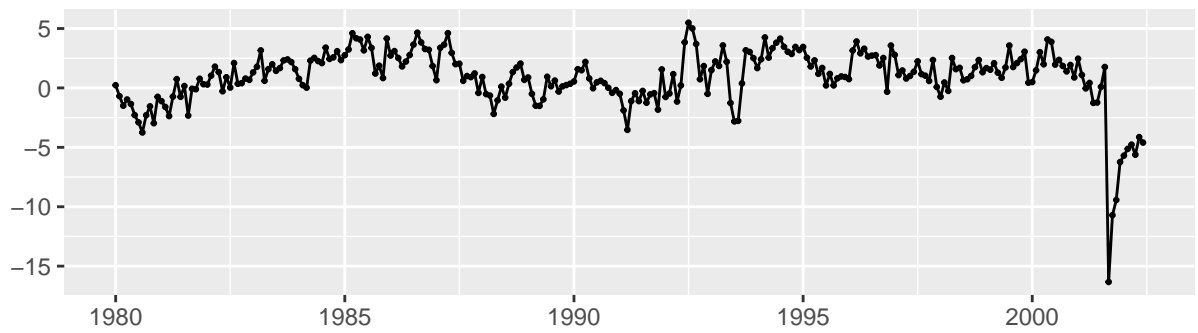
```
ggtsdisplay(diff(enplanements,lag=3),
  main=paste("enplanements - BoxCox lambda = ",
    enplanements_lambda," - quarterly seasonal diff"))
```

enplanements – BoxCox lambda = -0.22695 – quarterly seasonal diff



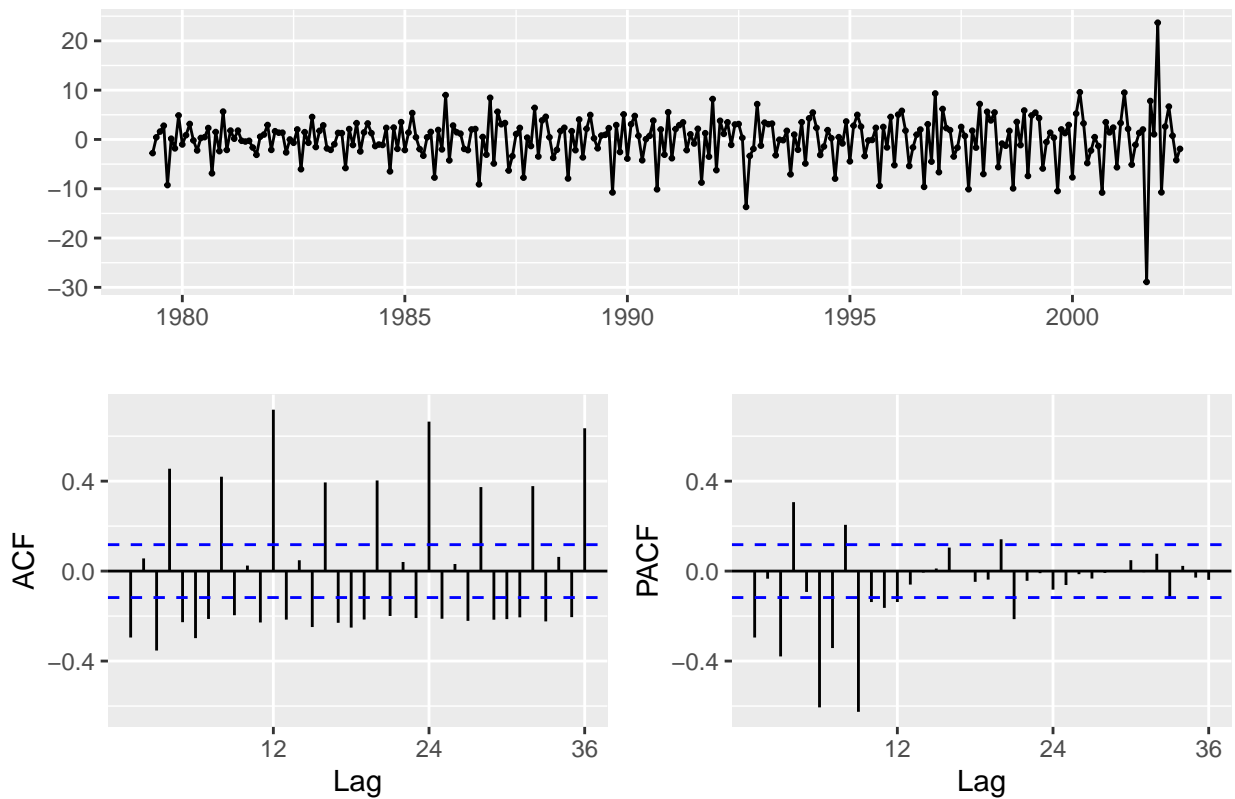
```
ggtsdisplay(diff(enplanements,lag=12),  
  main=paste("enplanements - BoxCox lambda = ",  
    enplanements_lambda," - annual seasonal diff"))
```

enplanements – BoxCox lambda = -0.22695 – annual seasonal diff



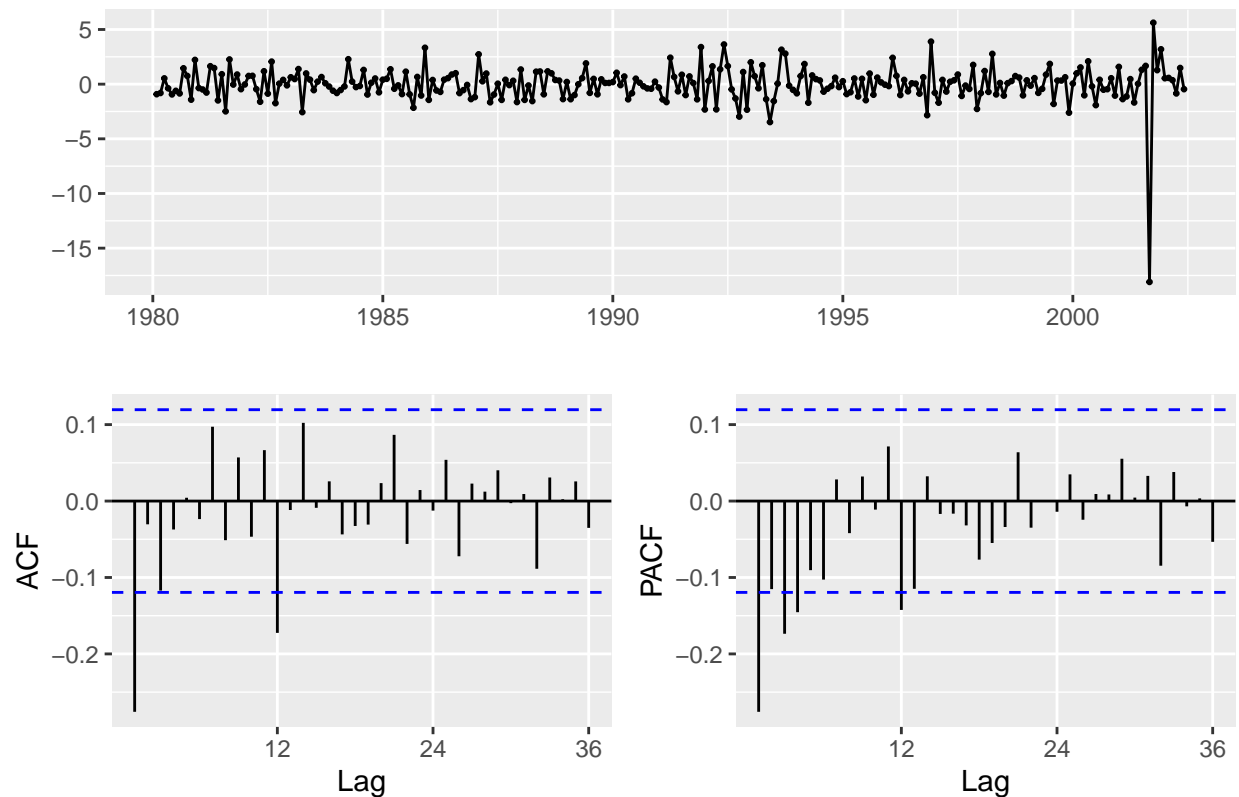
```
ggtsdisplay(diff(diff(enplanements,lag=3),lag=1),  
            main=paste("enplanements - BoxCox lambda = ",  
                        enplanements_lambda," - quarterly seasonal diff + first diff"))
```

enplanements – BoxCox lambda = -0.22695 – quarterly seasonal diff + fi



```
ggtsdisplay(diff(diff(enplanements,lag=12),lag=1),
  main=paste("enplanements - BoxCox lambda = ",
    enplanements_lambda," - annual seasonal diff + first diff"))
```

enplanements – BoxCox lambda = -0.22695 – annual seasonal diff + first



The “nsdiffs” and “ndiffs” functions indicate that seasonal differencing is required. Above, both quarterly and annual seasonality have been examined.

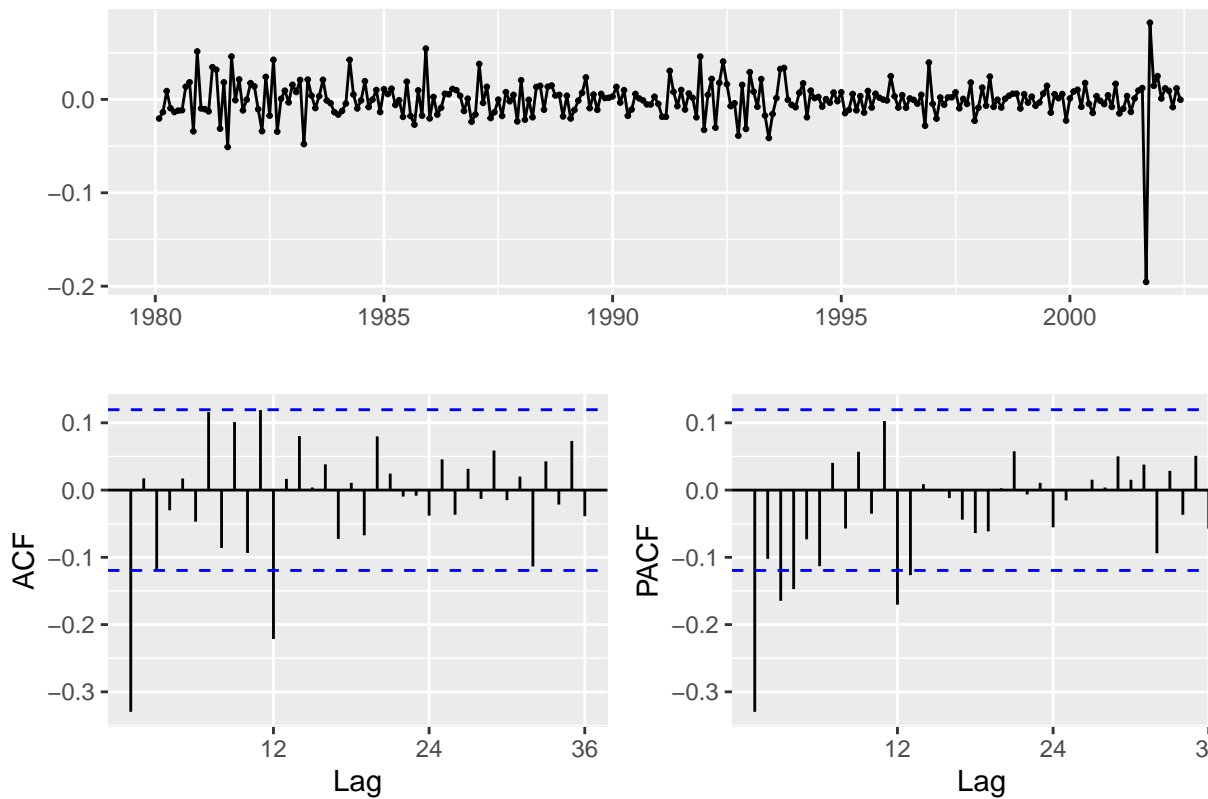
The raw data series and the Box-Cox transformed series each require 1 diff.

The result of Box-Cox transformation, annual seasonal differencing, plus regular differencing, seems to provide the best result.

Let's evaluate first differencing:

```
enplanements_BC %>% diff(lag=12) %>% diff(lag=1) -> enplanements_BC_s1_d1
ggtsdisplay(enplanements_BC_s1_d1,
  main = paste("enplanements - BoxCox lambda = ",
    enplanements_lambda, "annual seasonal diff + first diff"))
```

enplanements – BoxCox lambda = -0.22695 annual seasonal diff + first



Diff data series:

The first differenced series shows autocorrelation on lags 1, and a few of the subsequent lags breach the critical value.

```
Box.test(enplanements_BC_s1_d1, type = "Ljung-Box")
```

Ljung-box test:

```
##
## Box-Ljung test
##
## data: enplanements_BC_s1_d1
## X-squared = 29.562, df = 1, p-value = 5.417e-08
```

Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

```
kpss.test(enplanements, "Level", lshort = F)
```

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
## Warning in kpss.test(enplanements, "Level", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: enplanements
## KPSS Level = 1.7486, Truncation lag parameter = 15, p-value = 0.01
```

```
kpss.test(enplanements, "Trend", lshort = F)
```

```
## Warning in kpss.test(enplanements, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: enplanements
## KPSS Trend = 0.070584, Truncation lag parameter = 15, p-value = 0.1
```

```
enplanements %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 15 lags.
##
## Value of test-statistic is: 1.7486
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
enplanements %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 15 lags.
##
## Value of test-statistic is: 0.0706
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```



```
kpss.test(enplanements_BC, "Level", lshort = F)
```

The KPSS tests on the raw data series fail on Level, but pass on Trend

```
## Warning in kpss.test(enplanements_BC, "Level", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: enplanements_BC
## KPSS Level = 1.714, Truncation lag parameter = 15, p-value = 0.01
```

```
kpss.test(enplanements_BC, "Trend", lshort = F)
```

```
##
## KPSS Test for Trend Stationarity
##
## data: enplanements_BC
## KPSS Trend = 0.18061, Truncation lag parameter = 15, p-value = 0.02327
```

```
enplanements_BC %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 15 lags.
##
## Value of test-statistic is: 1.714
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
enplanements_BC %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 15 lags.
##
## Value of test-statistic is: 0.1806
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```

```
kpss.test(enplanements_BC_s1_d1, "Level", lshort = F)
```

Both KPSS tests fail on Box Cox transformed data series

```
## Warning in kpss.test(enplanements_BC_s1_d1, "Level", lshort = F): p-value  
## greater than printed p-value
```

```
##  
## KPSS Test for Level Stationarity  
##  
## data: enplanements_BC_s1_d1  
## KPSS Level = 0.074682, Truncation lag parameter = 15, p-value = 0.1
```

```
kpss.test(enplanements_BC_s1_d1, "Trend", lshort = F)
```

```
## Warning in kpss.test(enplanements_BC_s1_d1, "Trend", lshort = F): p-value  
## greater than printed p-value
```

```
##  
## KPSS Test for Trend Stationarity  
##  
## data: enplanements_BC_s1_d1  
## KPSS Trend = 0.03852, Truncation lag parameter = 15, p-value = 0.1
```

```
enplanements_BC_s1_d1 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: mu with 15 lags.  
##  
## Value of test-statistic is: 0.0747  
##  
## Critical value for a significance level of:  
##          10pct  5pct  2.5pct  1pct  
## critical values 0.347 0.463  0.574 0.739
```

```
enplanements_BC_s1_d1 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

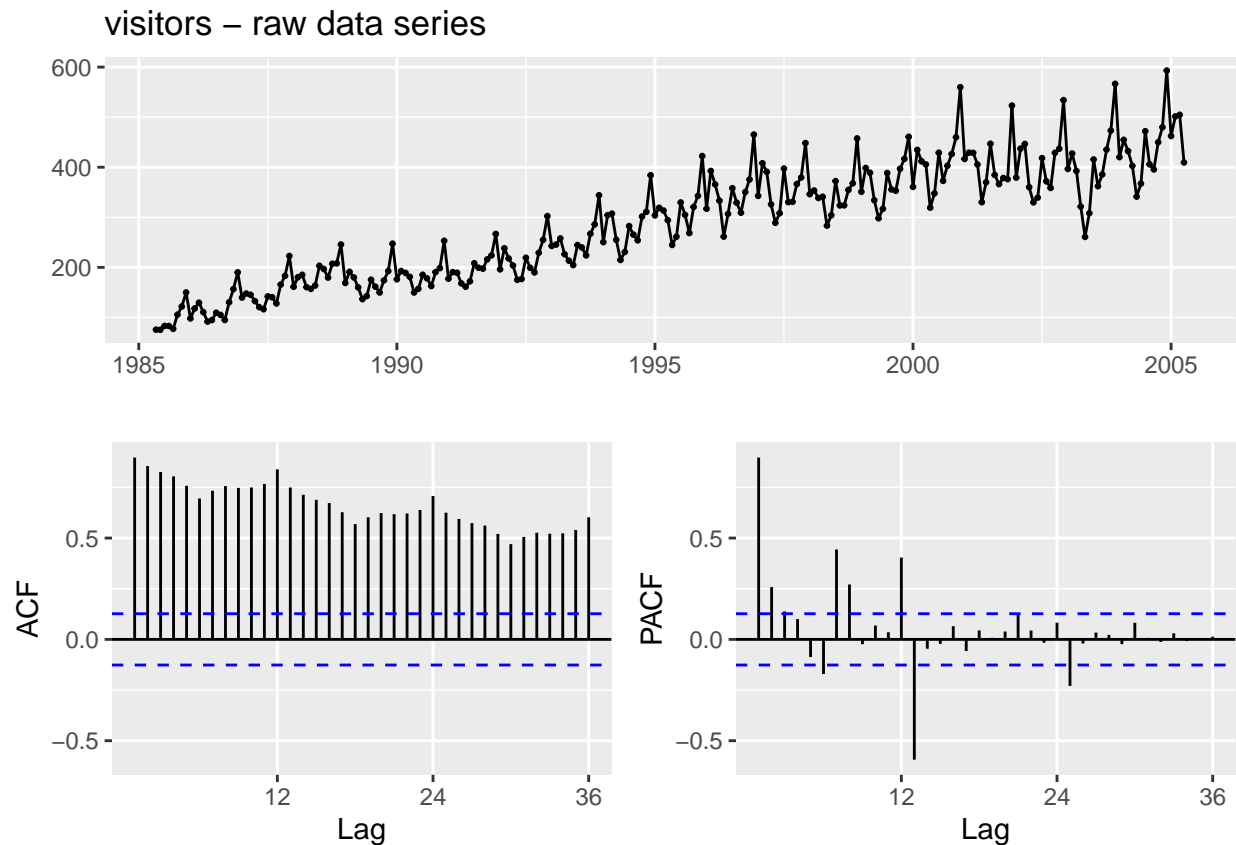
```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: tau with 15 lags.  
##  
## Value of test-statistic is: 0.0385
```

```
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146  0.176 0.216
```

Both KPSS tests pass for the data series which has been transformed by annual seasonality and first differences.

e) visitors

```
ggtsdisplay(visitors, main = "visitors - raw data series")
```



The graph of the raw data shows strong autocorrelation as well as seasonality.

```
Box.test(visitors, type = "Ljung-Box")
```

Ljung-box test on raw data series:

```
##  
## Box-Ljung test  
##  
## data: visitors  
## X-squared = 195.64, df = 1, p-value < 2.2e-16
```

Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

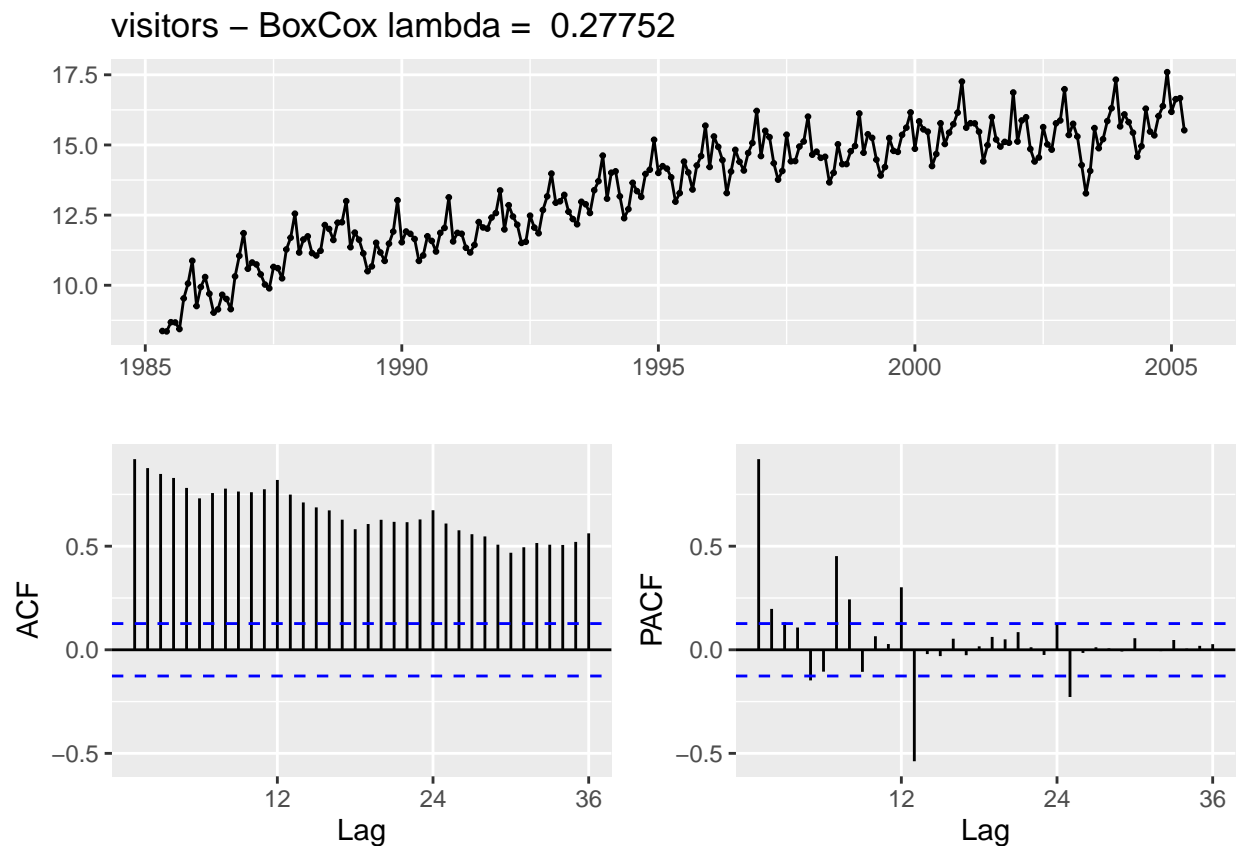
```
visitors_lambda <- round(BoxCox.lambda(visitors),5)
print(paste("Lambda for visitors: ", visitors_lambda))
```

Try Box-Cox transformation:

```
## [1] "Lambda for visitors: 0.27752"
```

```
visitors_BC <- BoxCox(visitors, visitors_lambda)
```

```
ggtsdisplay(visitors_BC, main = paste("visitors - BoxCox lambda = ",
                                     visitors_lambda))
```



The graph of the Box-Cox transformed data still shows strong autocorrelation and seasonality, but the variance has been dampened.

```
Box.test(visitors_BC, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: visitors_BC
## X-squared = 205.74, df = 1, p-value < 2.2e-16
```

Because the p-value is still low, the Null hypothesis is again REJECTED in favor of the alternative (Autocorrelation exists.)

```
visitors_nsdiffs <- nsdiffs(visitors)
print(paste("Number of SEASONAL differences suggested for visitors (raw data):",
            visitors_nsdiffs))
```

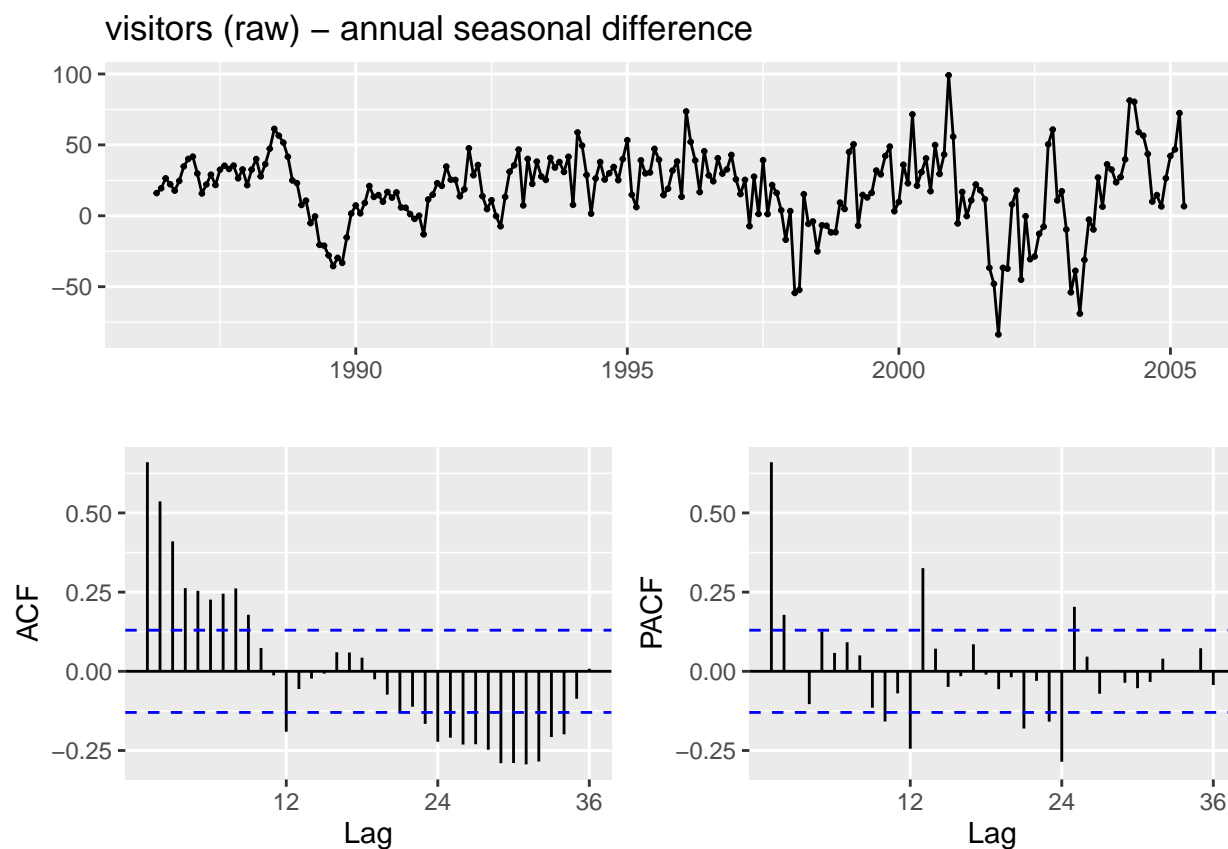
Check estimated number of differences needed:

```
## [1] "Number of SEASONAL differences suggested for visitors (raw data): 1"
```

```
visitors_ndiffs <- ndiffs(visitors)
print(paste("Number of differences suggested for visitors (raw data):",
            visitors_ndiffs))
```

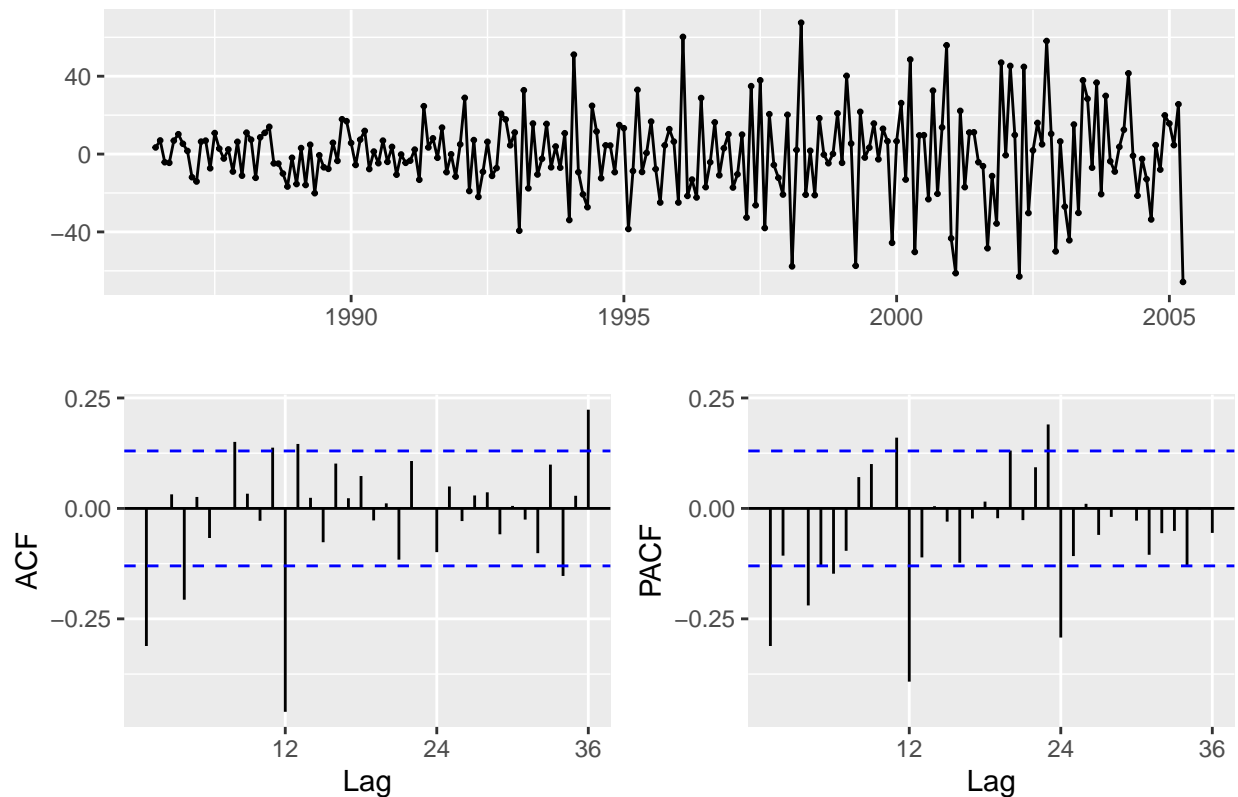
```
## [1] "Number of differences suggested for visitors (raw data): 1"
```

```
ggtsdisplay(diff(visitors,lag=12),
            main=paste("visitors (raw) - annual seasonal difference"))
```



```
ggtsdisplay(diff(diff(visitors,lag=12),lag=1),
            main=paste("visitors (raw) - annual seasonal difference + first diff"))
```

visitors (raw) – annual seasonal difference + first diff



The graphs show that differencing without the Box-Cox transformation is not adequate to achieve stationarity

```
visitors_BC_nsdiffs <- nsdiffs(visitors_BC)
print(paste("Number of SEASONAL differences suggested for visitors_BC:", visitors_BC_nsdiffs))
```

Check differences estimated for the Box-Cox transformed series:

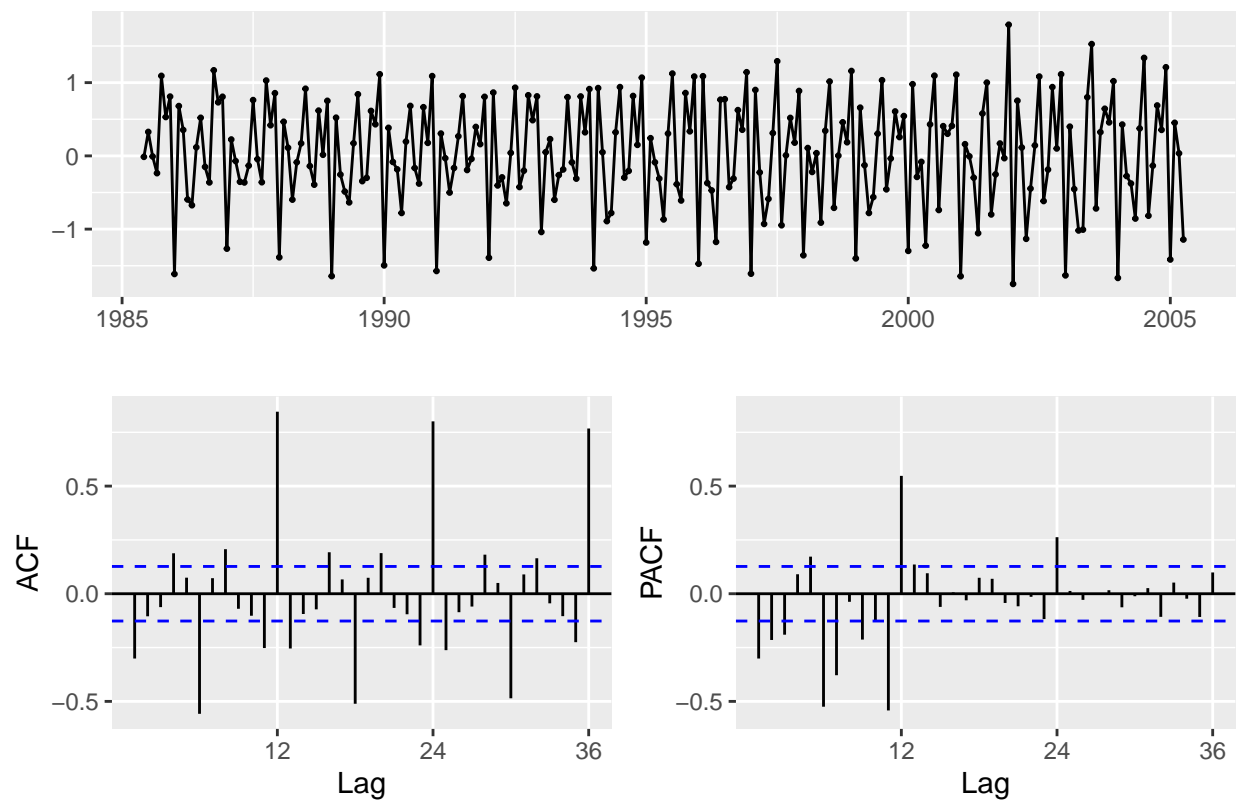
```
## [1] "Number of SEASONAL differences suggested for visitors_BC: 1"
```

```
visitors_BC_ndiffs <- ndiffs(visitors_BC)
print(paste("Number of differences suggested for visitors_BC:",
            visitors_BC_ndiffs))
```

```
## [1] "Number of differences suggested for visitors_BC: 1"
```

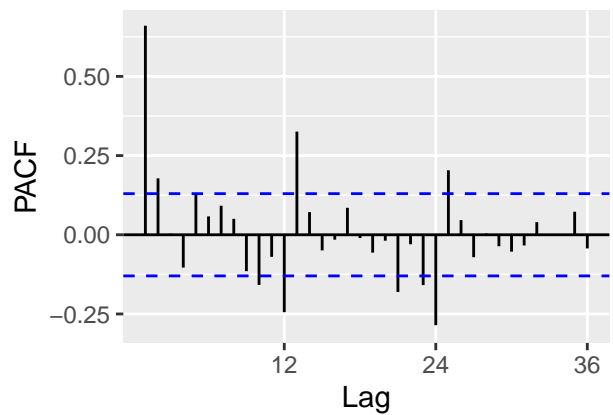
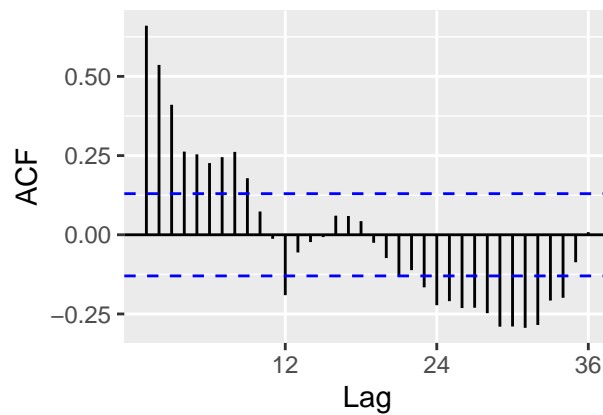
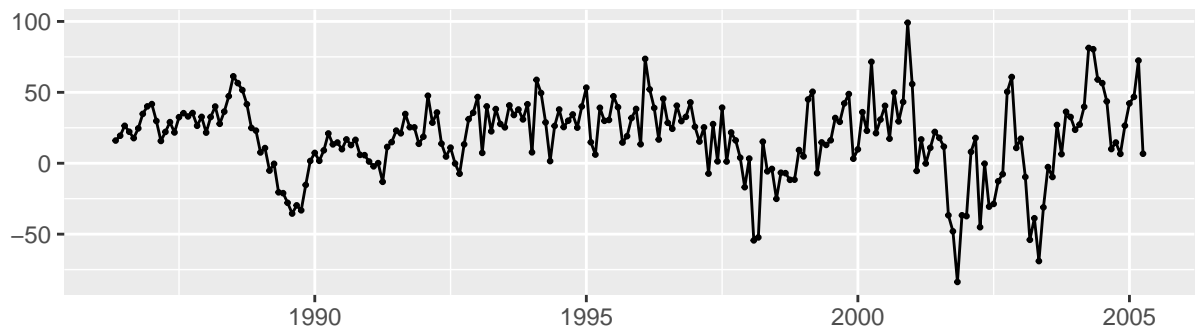
```
ggttsdisplay(diff(visitors_BC),
              main=paste("visitors - BoxCox lambda = ",
                          visitors_lambda, " - first difference"))
```

visitors – BoxCox lambda = 0.27752 – first difference



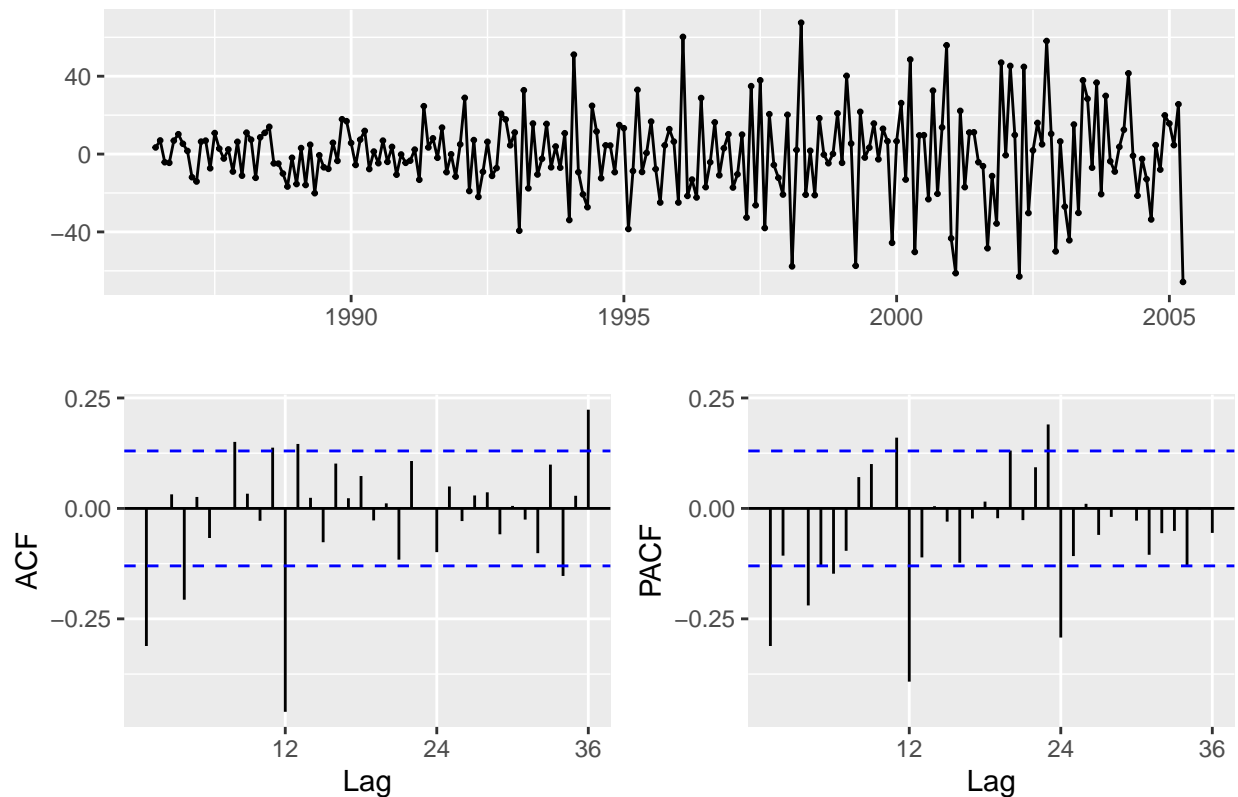
```
ggtsdisplay(diff(visitors,lag=12),  
  main=paste("visitors – BoxCox lambda = ",  
    visitors_lambda," – annual seasonal diff"))
```


visitors – BoxCox lambda = 0.27752 – annual seasonal diff



```
ggtsdisplay(diff(diff(visitors,lag=12),lag=1),  
  main=paste("visitors - BoxCox lambda = ",  
    visitors_lambda," - annual seasonal diff + first diff"))
```

visitors – BoxCox lambda = 0.27752 – annual seasonal diff + first diff



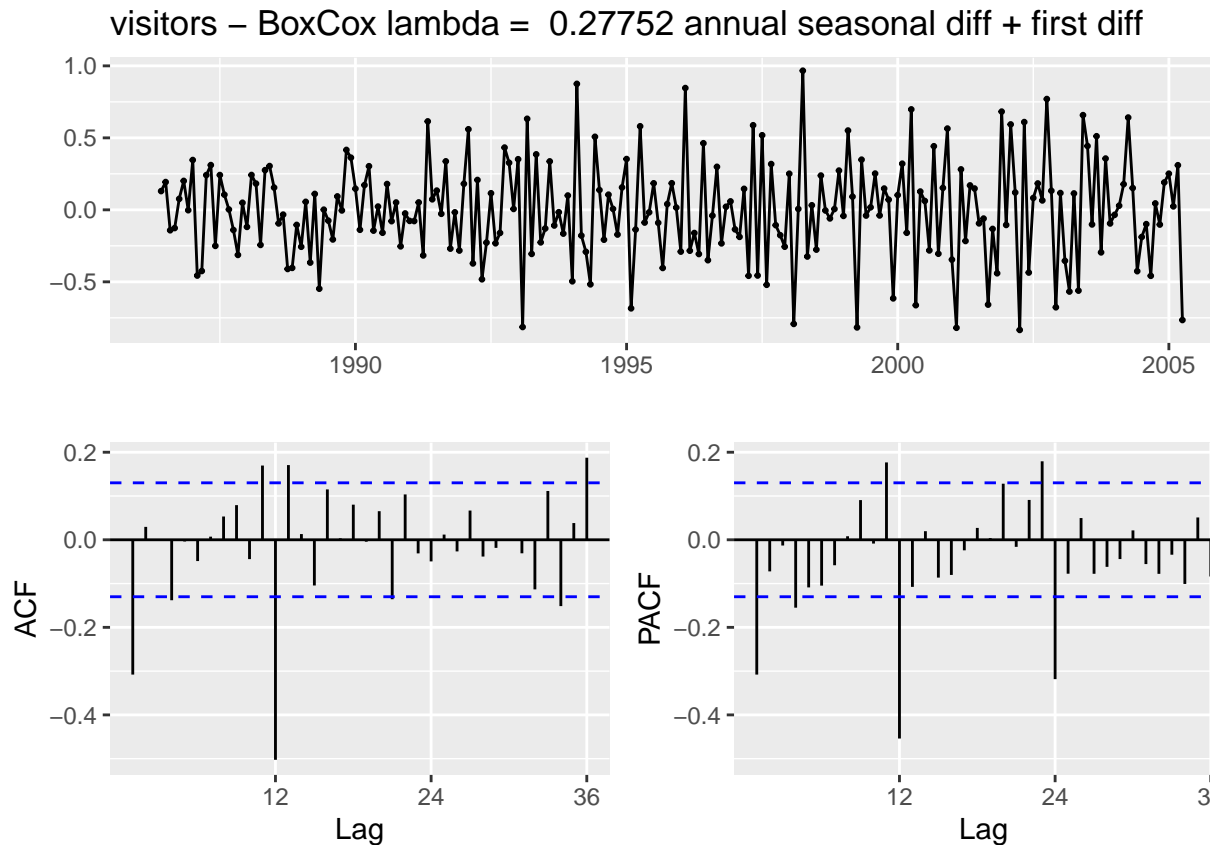
The “nsdiffs” and “ndiffs” functions indicate that seasonal differencing is required.

The raw data series and the Box-Cox transformed series each require 1 diff.

The result of Box-Cox transformation, annual seasonal differencing, plus regular differencing, seems to provide the best result.

Let’s evaluate first differencing:

```
visitors_BC %>% diff(lag=12) %>% diff(lag=1) -> visitors_BC_s1_d1
ggtsdisplay(visitors_BC_s1_d1,
  main = paste("visitors - BoxCox lambda = ",
    visitors_lambda, "annual seasonal diff + first diff"))
```



Diff data series:

The first differenced series shows autocorrelation on lags 1, and a few of the subsequent lags breach the critical value.

```
Box.test(visitors_BC_s1_d1, type = "Ljung-Box")
```

Ljung-box test:

```
##
## Box-Ljung test
##
## data: visitors_BC_s1_d1
## X-squared = 21.804, df = 1, p-value = 3.02e-06
```

Because the p-value is low, the Null hypothesis is REJECTED in favor of the alternative (Autocorrelation exists.)

```
kpss.test(visitors, "Level", lshort = F)
```

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
## Warning in kpss.test(visitors, "Level", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: visitors
## KPSS Level = 1.6864, Truncation lag parameter = 14, p-value = 0.01
```

```
kpss.test(visitors, "Trend", lshort = F)
```

```
##
## KPSS Test for Trend Stationarity
##
## data: visitors
## KPSS Trend = 0.15852, Truncation lag parameter = 14, p-value = 0.03956
```

```
visitors %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 14 lags.
##
## Value of test-statistic is: 1.6864
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
visitors %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 14 lags.
##
## Value of test-statistic is: 0.1585
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```

```
kpss.test(visitors_BC, "Level", lshort = F)
```

Both KPSS tests fail on the raw data series

```
## Warning in kpss.test(visitors_BC, "Level", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: visitors_BC
## KPSS Level = 1.652, Truncation lag parameter = 14, p-value = 0.01
```

```
kpss.test(visitors_BC, "Trend", lshort = F)
```

```
## Warning in kpss.test(visitors_BC, "Trend", lshort = F): p-value smaller than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: visitors_BC
## KPSS Trend = 0.28285, Truncation lag parameter = 14, p-value = 0.01
```

```
visitors_BC %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 14 lags.
##
## Value of test-statistic is: 1.652
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
visitors_BC %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 14 lags.
##
## Value of test-statistic is: 0.2828
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```

```
kpss.test(visitors_BC_s1_d1, "Level", lshort = F)
```

Both KPSS tests fail on Box Cox transformed data series

```
## Warning in kpss.test(visitors_BC_s1_d1, "Level", lshort = F): p-value greater
## than printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: visitors_BC_s1_d1
## KPSS Level = 0.032281, Truncation lag parameter = 14, p-value = 0.1
```

```
kpss.test(visitors_BC_s1_d1, "Trend", lshort = F)
```

```
## Warning in kpss.test(visitors_BC_s1_d1, "Trend", lshort = F): p-value greater
## than printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data: visitors_BC_s1_d1
## KPSS Trend = 0.032061, Truncation lag parameter = 14, p-value = 0.1
```

```
visitors_BC_s1_d1 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 14 lags.
##
## Value of test-statistic is: 0.0323
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
visitors_BC_s1_d1 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

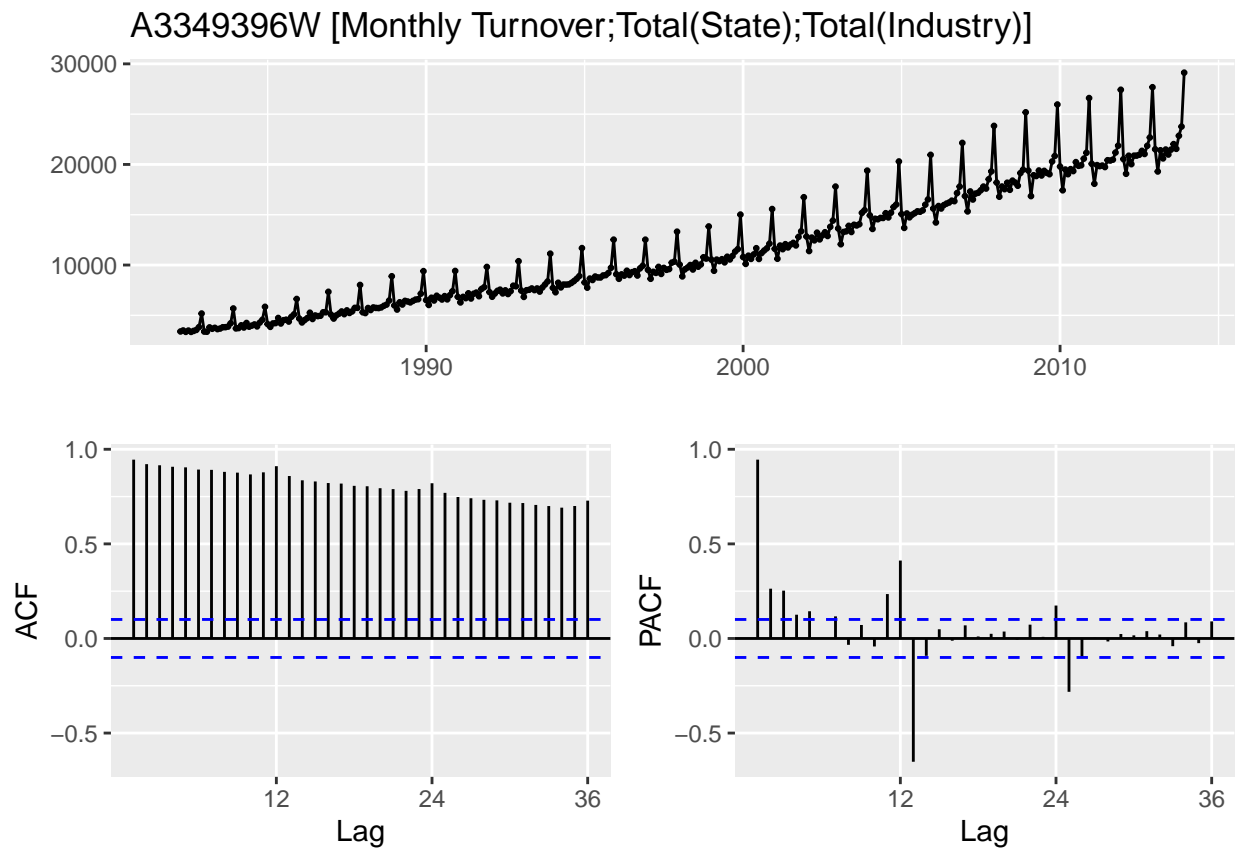
```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 14 lags.
##
## Value of test-statistic is: 0.0321
```

```
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146  0.176 0.216
```

Both KPSS tests pass for the data series which has been transformed by annual seasonality and first differences.

8.5 For your retail data (from Exercise 3 in Section 2.10), find the appropriate order of differencing (after transformation if necessary) to obtain stationary data.

```
mycode <- "A3349396W"
mytitle <- "[Monthly Turnover;Total(State);Total(Industry)]"
mymain <- paste(mycode,mytitle)
myts <- readxl::read_excel("retail.xlsx", skip=1)[,mycode] %>%
  ts(frequency=12, start=c(1982,4))
ggtsdisplay(myts,main=mymain)
```



```
Box.test(myts, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: myts
## X-squared = 343.06, df = 1, p-value < 2.2e-16
```

```
kpss.test(myts, "Level", lshort = F)
```


Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
## Warning in kpss.test(myts, "Level", lshort = F): p-value smaller than printed p-  
## value
```

```
##  
## KPSS Test for Level Stationarity  
##  
## data: myts  
## KPSS Level = 2.2987, Truncation lag parameter = 16, p-value = 0.01
```

```
kpss.test(myts, "Trend", lshort = F)
```

```
## Warning in kpss.test(myts, "Trend", lshort = F): p-value smaller than printed p-  
## value
```

```
##  
## KPSS Test for Trend Stationarity  
##  
## data: myts  
## KPSS Trend = 0.56755, Truncation lag parameter = 16, p-value = 0.01
```

```
myts %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: mu with 16 lags.  
##  
## Value of test-statistic is: 2.2987  
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.347 0.463  0.574 0.739
```

```
myts %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: tau with 16 lags.  
##  
## Value of test-statistic is: 0.5676  
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146  0.176 0.216
```

```
myts_nsdiffs <- nsdiffs(myts)
print(paste("Number of SEASONAL differences suggested for",mymain,":", myts_nsdiffs))
```

The KPSS tests on the raw data series fail on both Level and Trend

```
## [1] "Number of SEASONAL differences suggested for A3349396W [Monthly Turnover;Total(State);Total(Ind
```

```
myts_ndiffs <- ndiffs(myts)
print(paste("Number of SEASONAL differences suggested for",mymain,":", myts_ndiffs))
```

```
## [1] "Number of SEASONAL differences suggested for A3349396W [Monthly Turnover;Total(State);Total(Ind
```

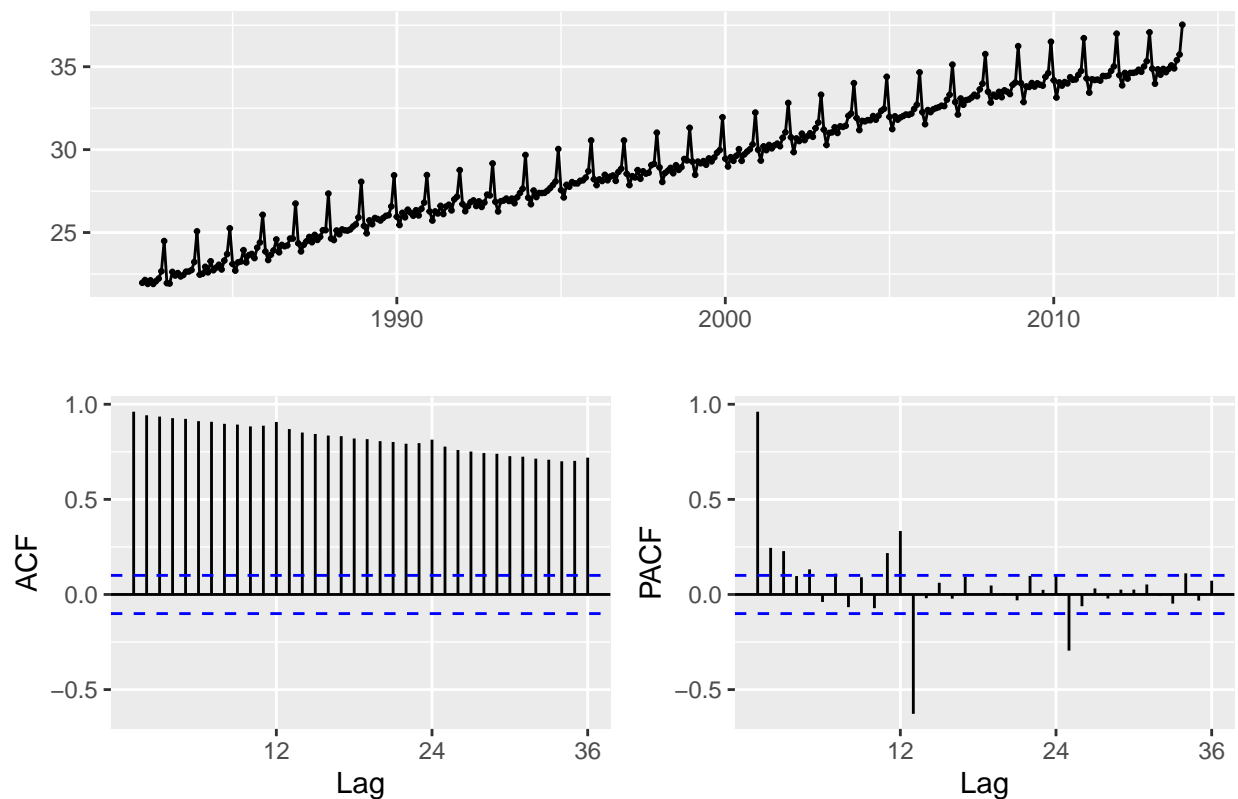
```
myts_lambda <- round(BoxCox.lambda(myts),5)
print(paste("Lambda for",mymain,":", mcopper_lambda))
```

Try Box-Cox transformation:

```
## [1] "Lambda for A3349396W [Monthly Turnover;Total(State);Total(Industry)] : 0.1919"
```

```
myts_BC <- BoxCox(myts, myts_lambda)
mymain_BC <- paste(mycode,"- BoxCox lambda = ",myts_lambda)
ggttsdisplay(myts_BC, main = mymain_BC)
```

A3349396W – BoxCox lambda = 0.21417



```
Box.test(myts_BC, type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: myts_BC
## X-squared = 354.56, df = 1, p-value < 2.2e-16
```

```
kpss.test(myts_BC, "Level", lshort = F)
```

```
## Warning in kpss.test(myts_BC, "Level", lshort = F): p-value smaller than printed
## p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: myts_BC
## KPSS Level = 2.3362, Truncation lag parameter = 16, p-value = 0.01
```

```
kpss.test(myts_BC, "Trend", lshort = F)
```

```
##
## KPSS Test for Trend Stationarity
##
## data: myts_BC
## KPSS Trend = 0.13707, Truncation lag parameter = 16, p-value = 0.06654
```

```
myts_BC %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 16 lags.
##
## Value of test-statistic is: 2.3362
##
## Critical value for a significance level of:
##          10pct  5pct  2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
myts_BC %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 16 lags.
```

```
##
## Value of test-statistic is: 0.1371
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

```
myts_BC_nsdiffs <- nsdiffs(myts_BC)
print(paste("Number of SEASONAL differences suggested for", mymain_BC, ":", myts_BC_nsdiffs))
```

The KPSS test on the Box Cox transformed data series fail on Level but pass on Trend.

```
## [1] "Number of SEASONAL differences suggested for A3349396W - BoxCox lambda = 0.21417 : 1"
```

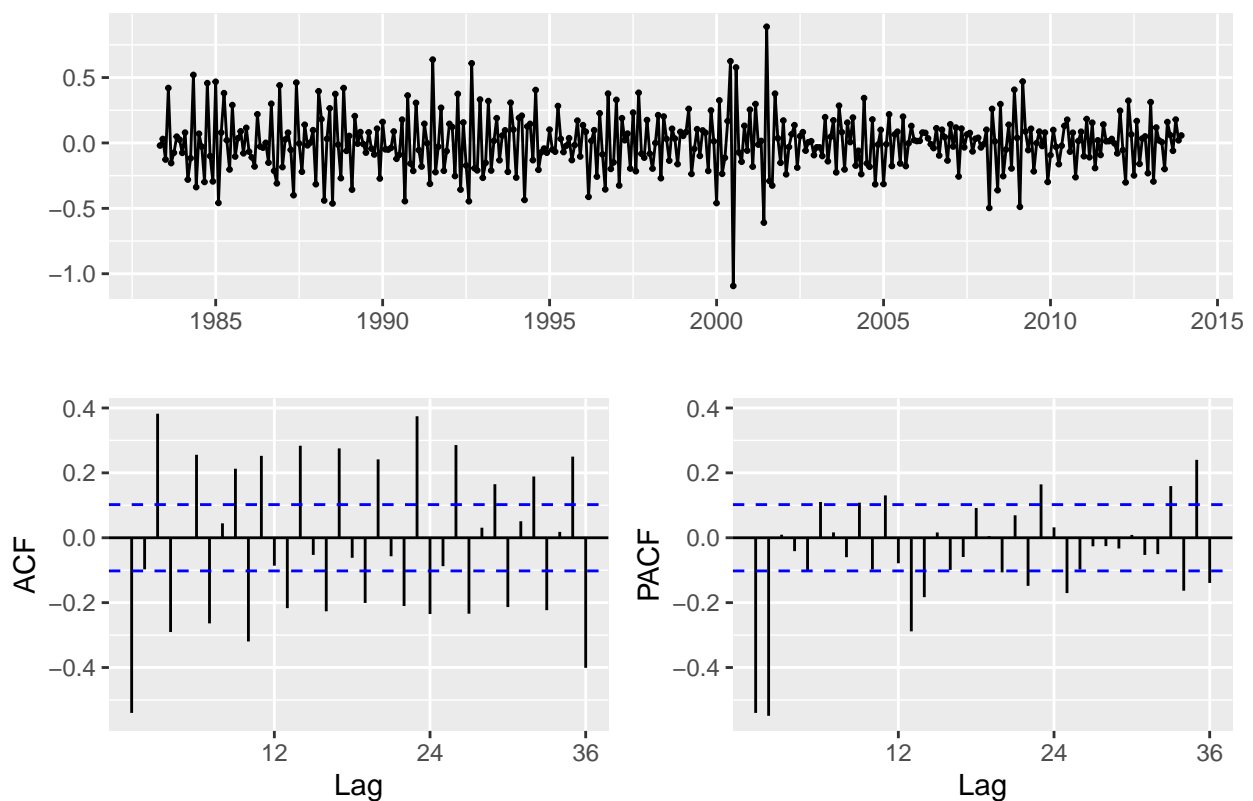
```
myts_BC_ndiffs <- ndiffs(myts_BC)
print(paste("Number of SEASONAL differences suggested for", mymain_BC, ":", myts_BC_ndiffs))
```

```
## [1] "Number of SEASONAL differences suggested for A3349396W - BoxCox lambda = 0.21417 : 1"
```

```
myts_BC_s1_d1 <- myts_BC %>% diff(lag=12) %>% diff(lag=1)

mymain_BC_diffs <- paste(mymain_BC, "Seasonal diff + first diff")
ggttsdisplay(myts_BC_s1_d1,
             main=mymain_BC_diffs)
```

A3349396W – BoxCox lambda = 0.21417 Seasonal diff + first diff



```
kpss.test(myts_BC_s1_d1, "Level", lshort = F)
```

```
## Warning in kpss.test(myts_BC_s1_d1, "Level", lshort = F): p-value greater than  
## printed p-value
```

```
##  
## KPSS Test for Level Stationarity  
##  
## data: myts_BC_s1_d1  
## KPSS Level = 0.028024, Truncation lag parameter = 16, p-value = 0.1
```

```
kpss.test(myts_BC_s1_d1, "Trend", lshort = F)
```

```
## Warning in kpss.test(myts_BC_s1_d1, "Trend", lshort = F): p-value greater than  
## printed p-value
```

```
##  
## KPSS Test for Trend Stationarity  
##  
## data: myts_BC_s1_d1  
## KPSS Trend = 0.027518, Truncation lag parameter = 16, p-value = 0.1
```

```
myts_BC_s1_d1 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: mu with 16 lags.  
##  
## Value of test-statistic is: 0.028  
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.347 0.463 0.574 0.739
```

```
myts_BC_s1_d1 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: tau with 16 lags.  
##  
## Value of test-statistic is: 0.0275  
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146 0.176 0.216
```

Both KPSS tests pass on annual seasonal + first differenced data series

8.6 Use R to simulate and plot some data from simple ARIMA models.

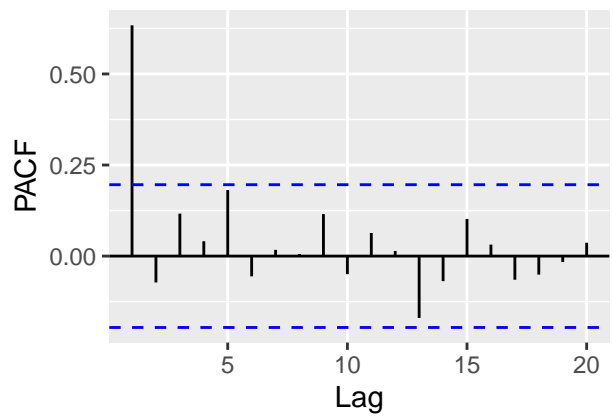
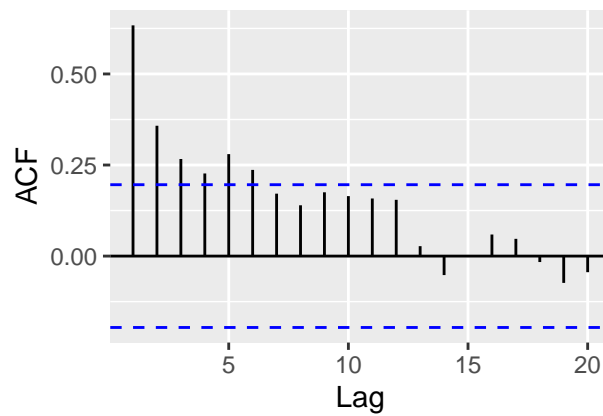
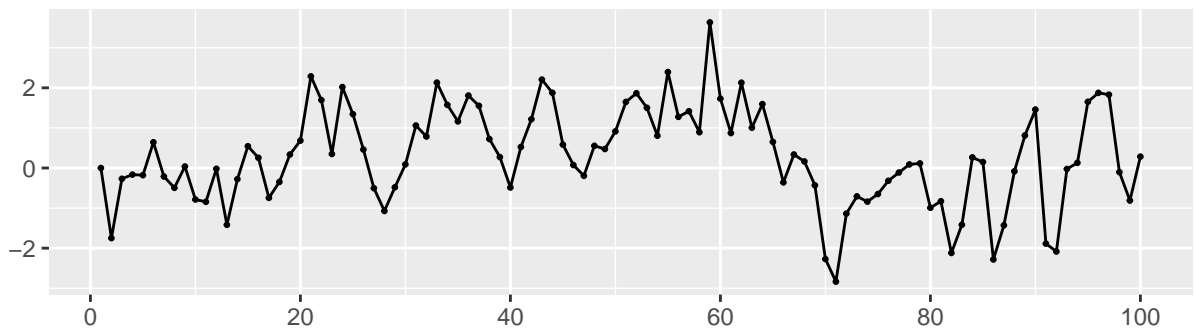
a) Use the following R code to generate data from an AR(1) model

with $\phi_1 = 0.6$ and $\sigma^2 = 1$. The process starts with $y_1 = 0$.

```
AR1 <- function(phi)
{
  y <- ts(numeric(100))
  e <- rnorm(100)
  for(i in 2:100)
    y[i] <- phi*y[i-1] + e[i]
  return(y)
}

phi_1 <- 0.60
ggtsdisplay(AR1(phi_1),main=paste("AR(1) series, phi_1=",phi_1))
```

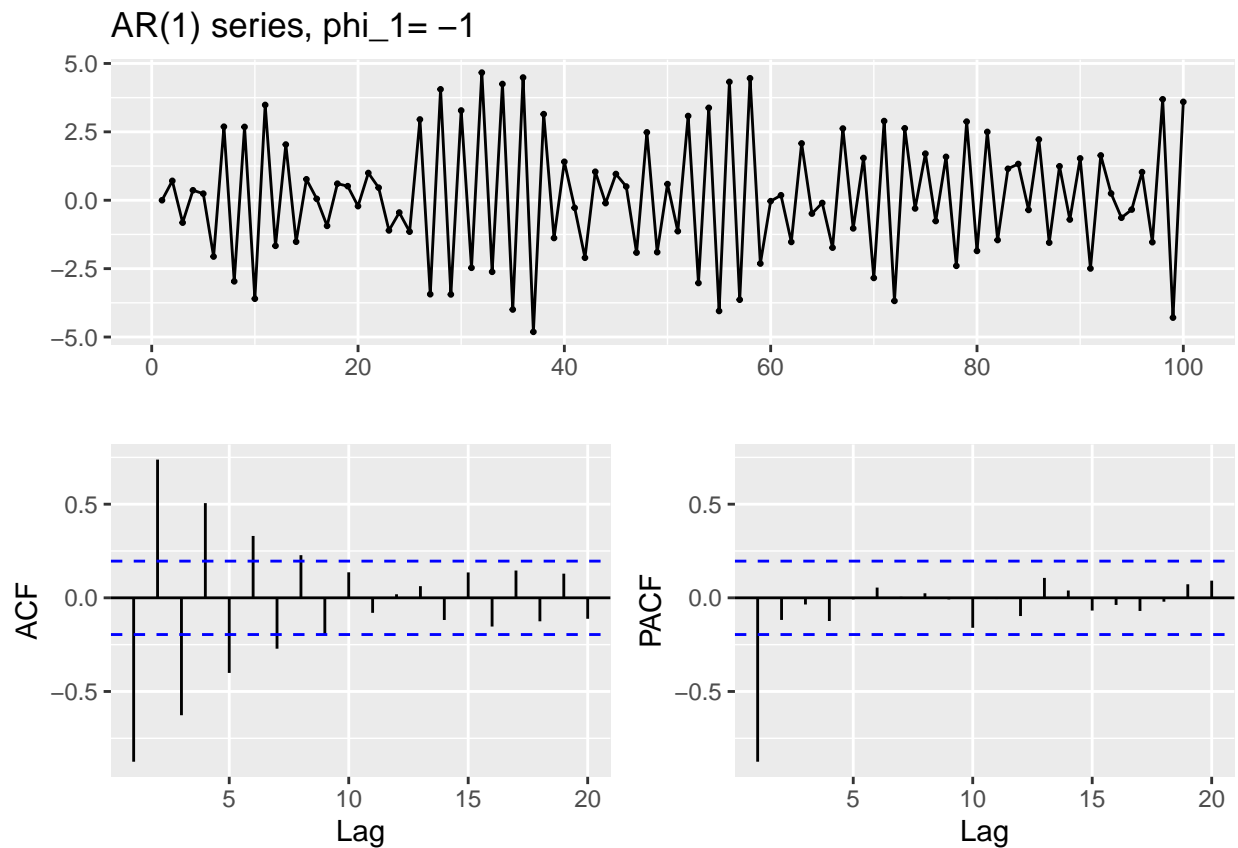
AR(1) series, phi_1= 0.6



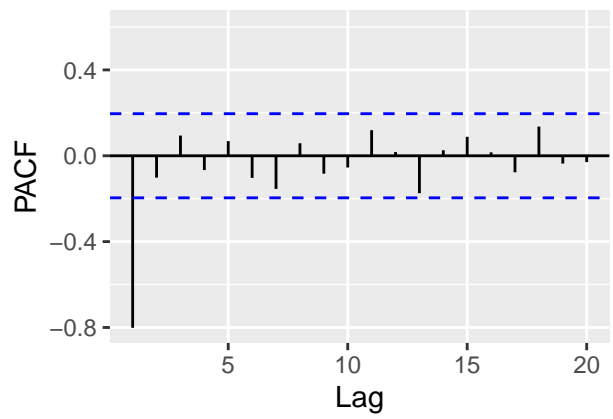
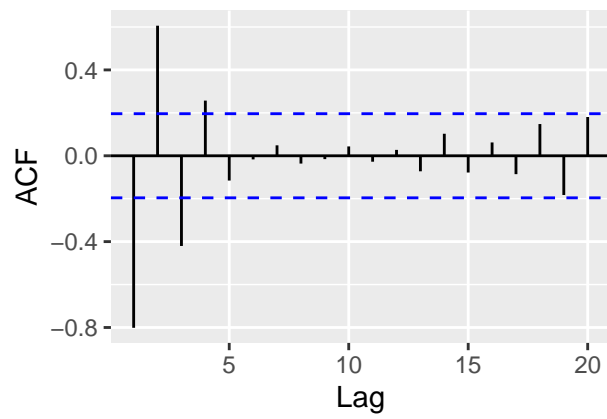
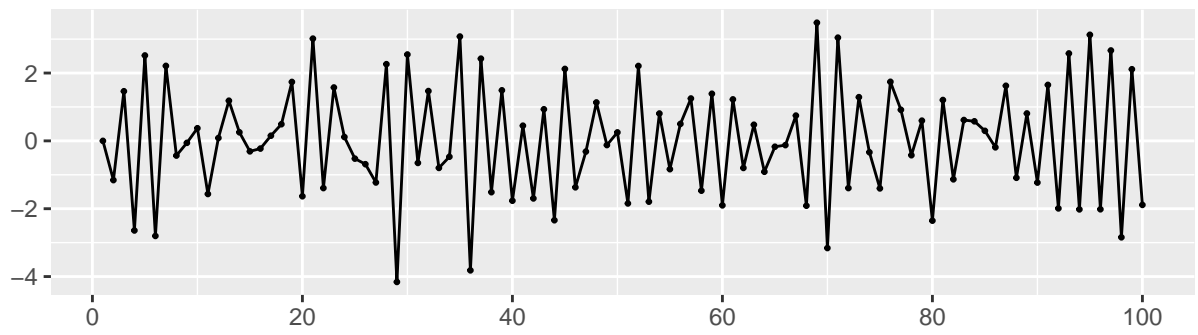
b) Produce a time plot for the series. How does the plot change as you change ϕ_1 ?

```
set.seed(12345)
AR1_stats_results = numeric()
phi_seq <- seq(from=-1,to=1,by = 0.1)

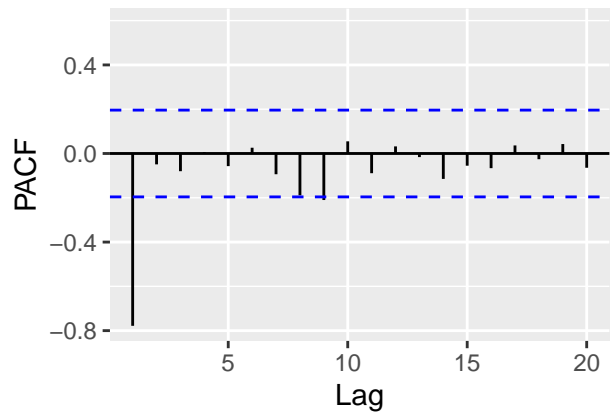
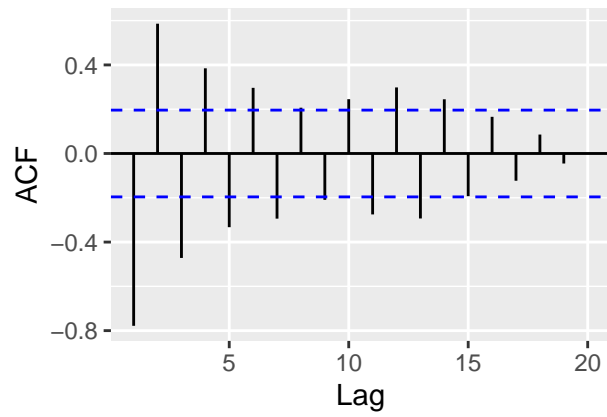
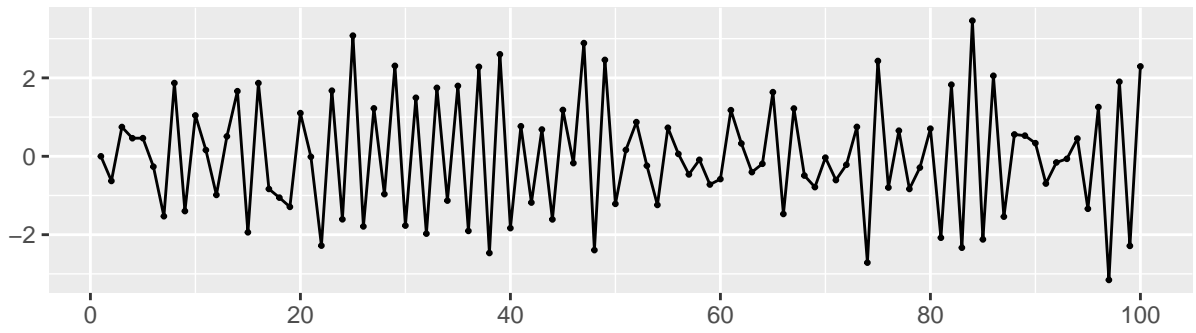
for (phi_1 in phi_seq) {
  AR1_series <- AR1(phi_1)
  AR1_stats <- c(Phi_1 = phi_1, summary(AR1_series),StDev=sd(AR1_series),SdDiff=sd(diff(AR1_series)))
  AR1_main <- paste("AR(1) series, phi_1=",phi_1)
  ggtsdisplay(AR1_series,main=AR1_main)
  #print(AR1_stats)
  AR1_stats_results <- rbind(AR1_stats_results,AR1_stats)
}
```



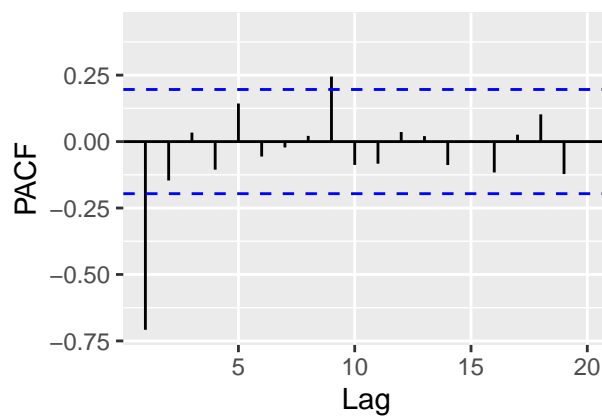
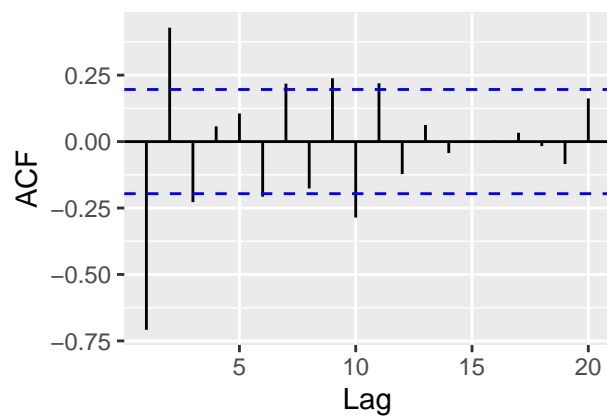
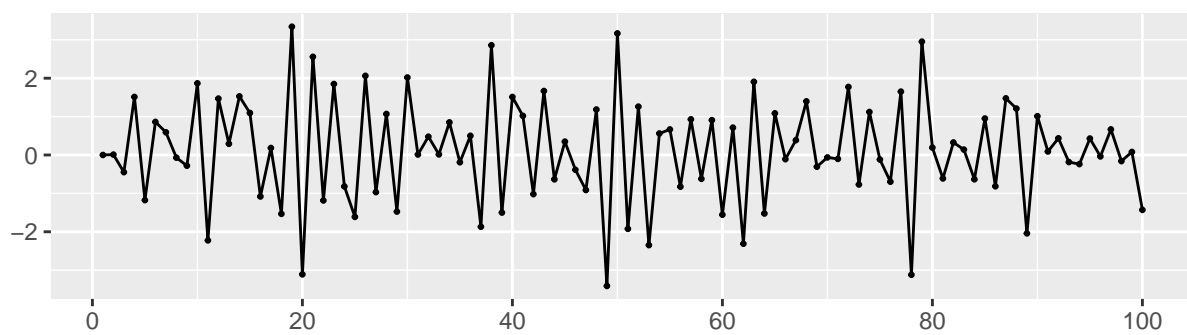
AR(1) series, $\phi_1 = -0.9$



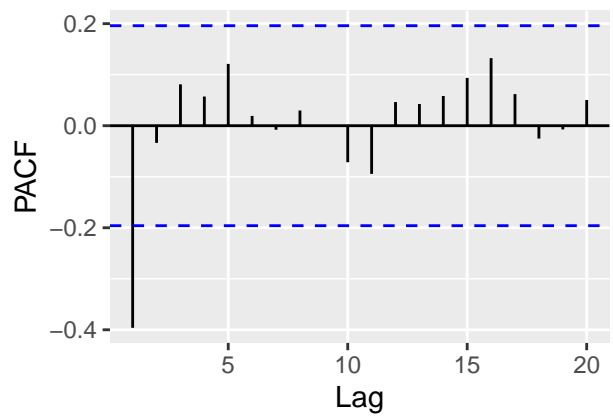
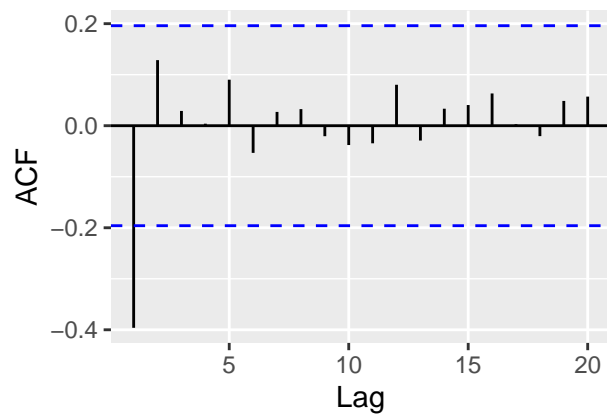
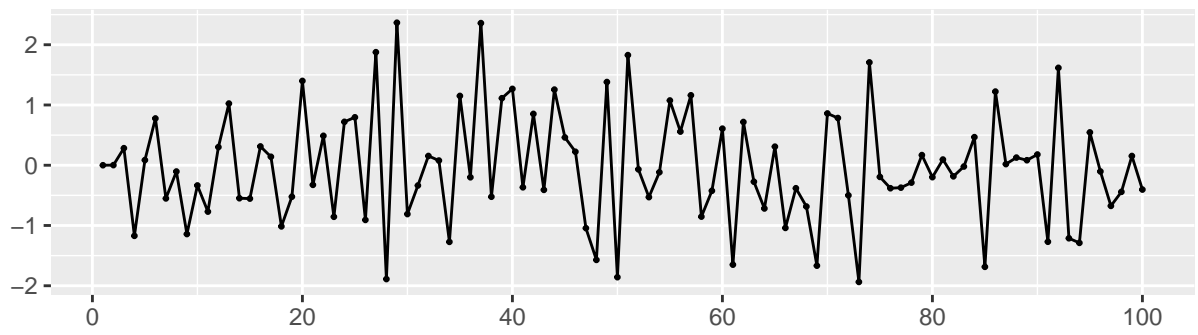
AR(1) series, $\phi_1 = -0.8$



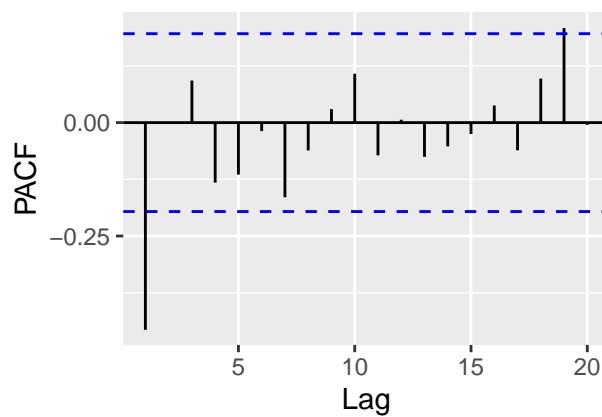
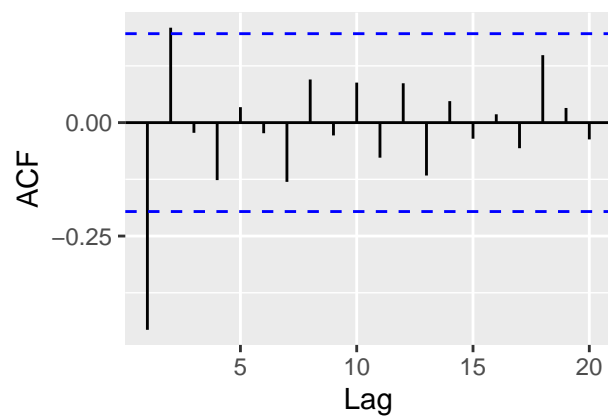
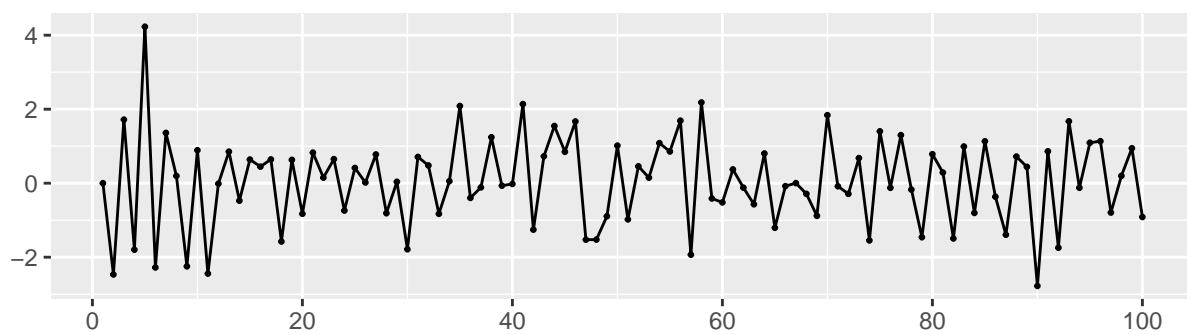
AR(1) series, $\phi_1 = -0.7$



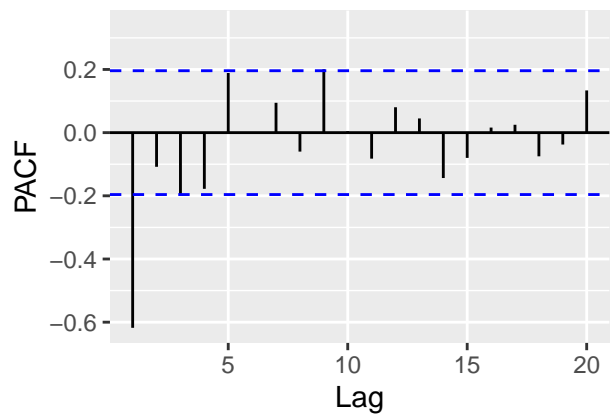
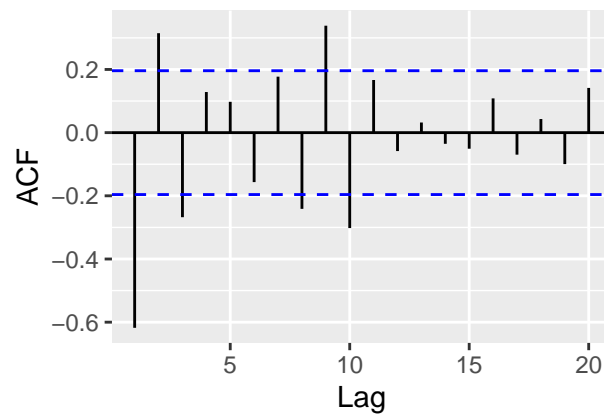
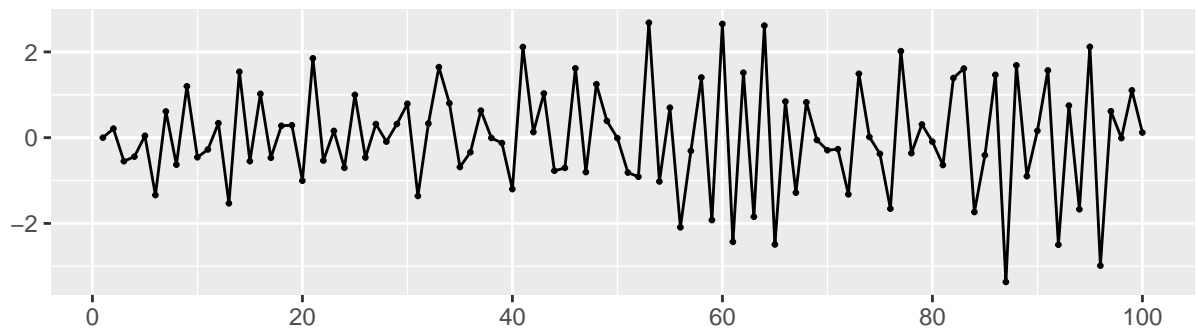
AR(1) series, $\phi_1 = -0.6$



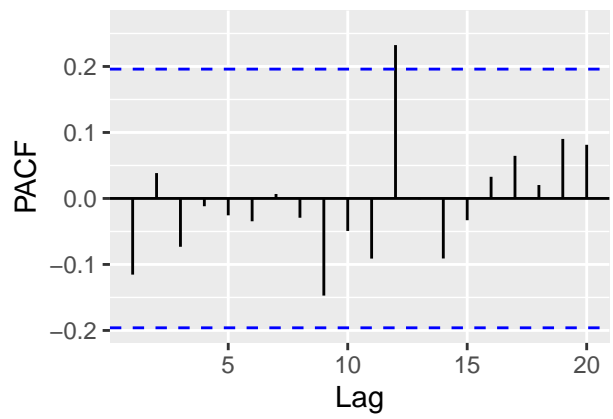
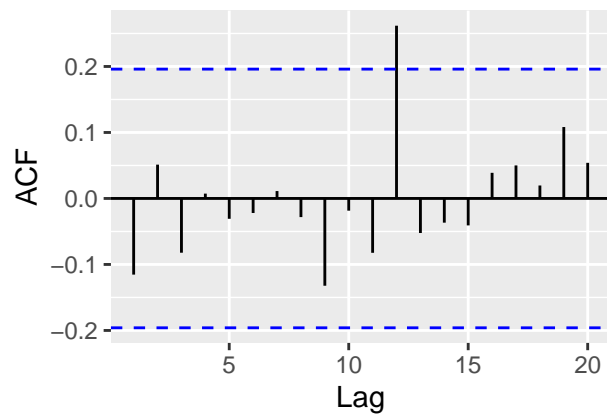
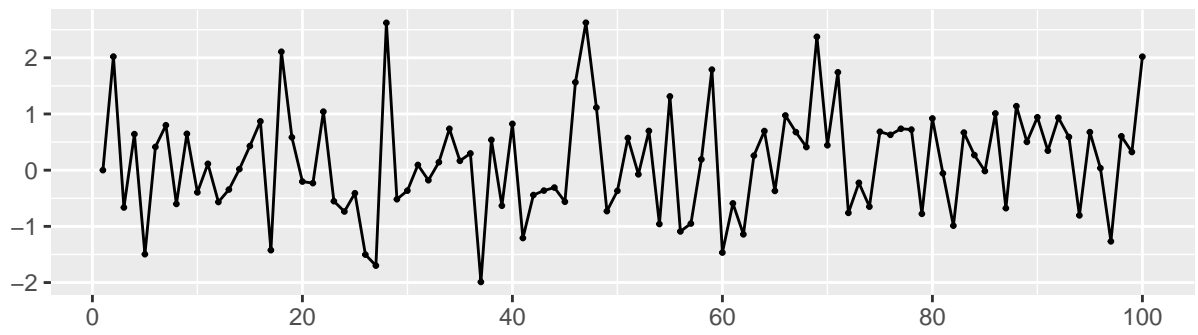
AR(1) series, $\phi_1 = -0.5$



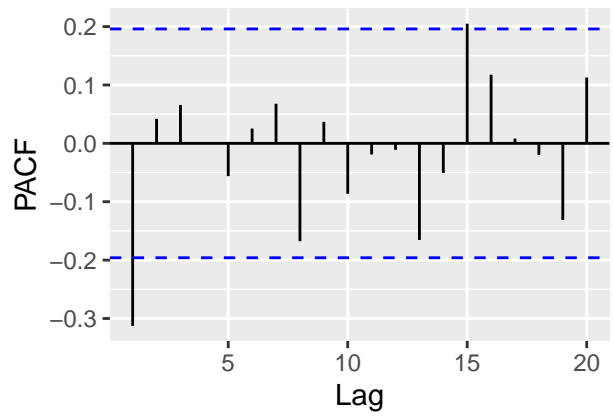
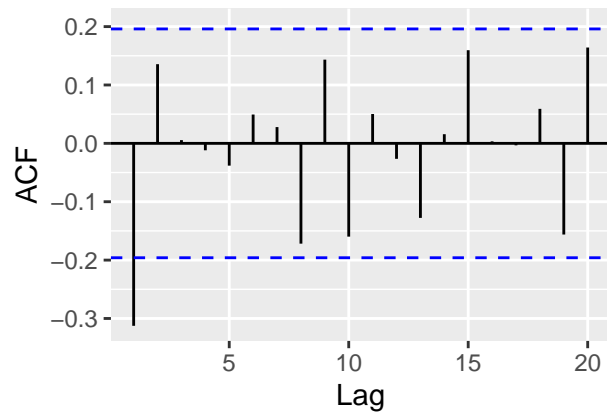
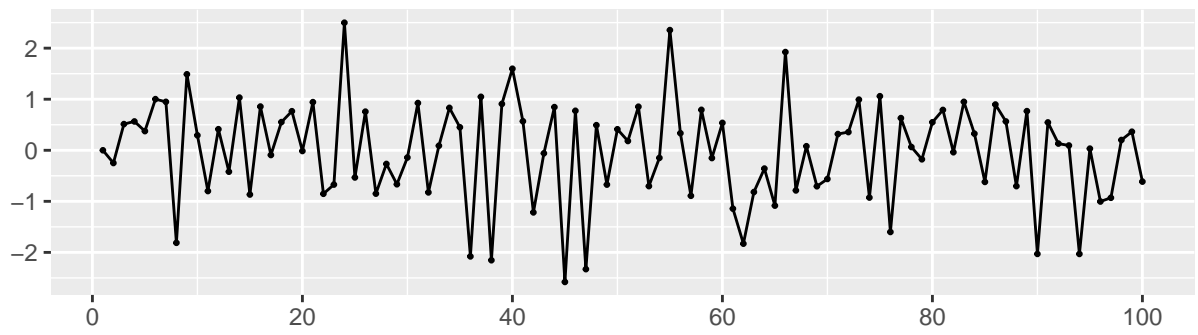
AR(1) series, $\phi_1 = -0.4$



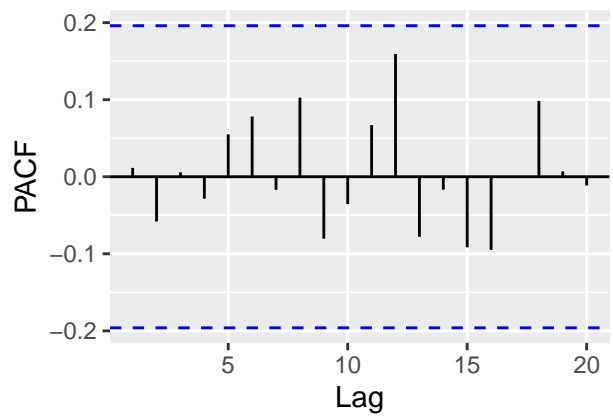
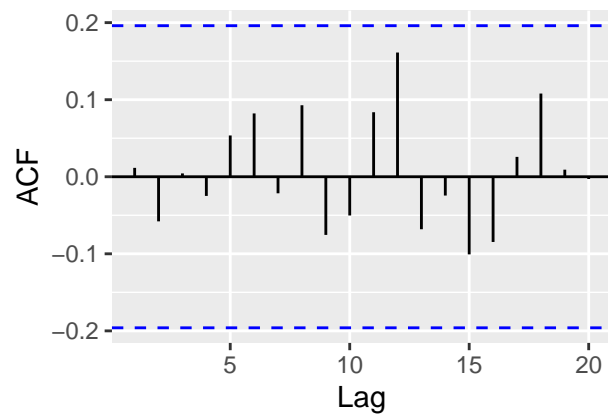
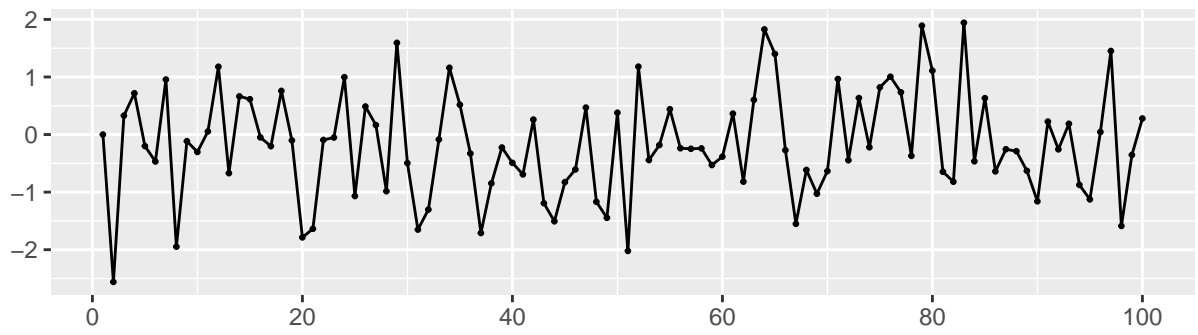
AR(1) series, $\phi_1 = -0.3$



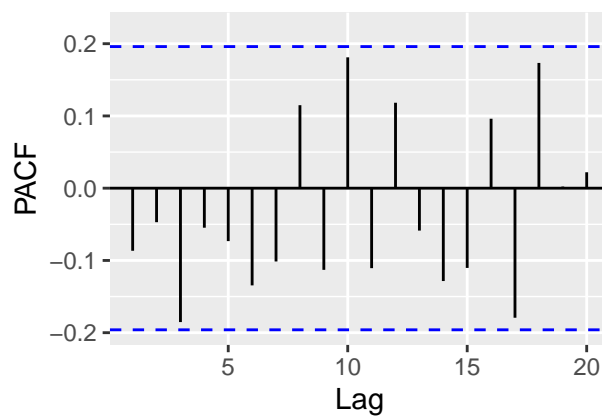
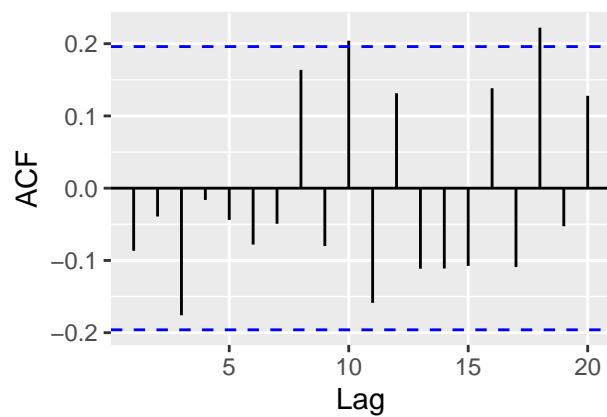
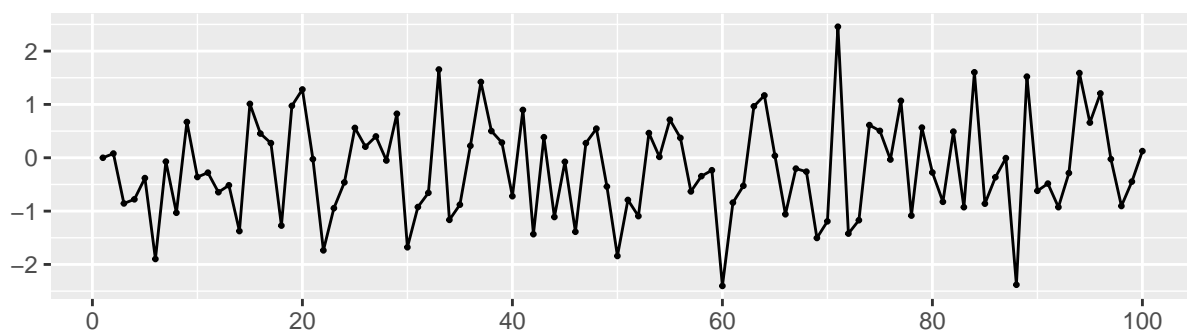
AR(1) series, $\phi_1 = -0.2$



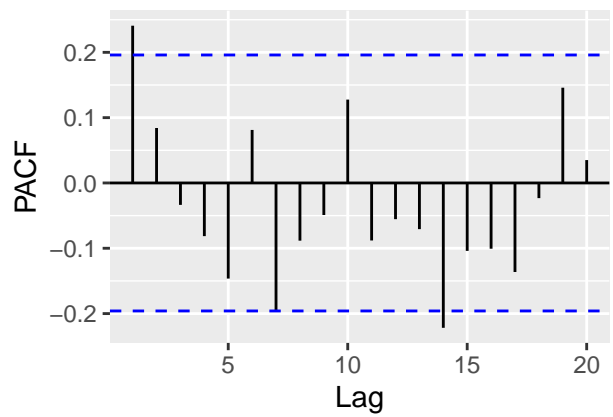
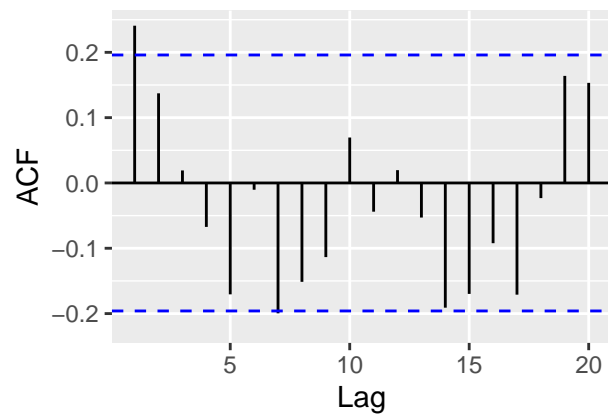
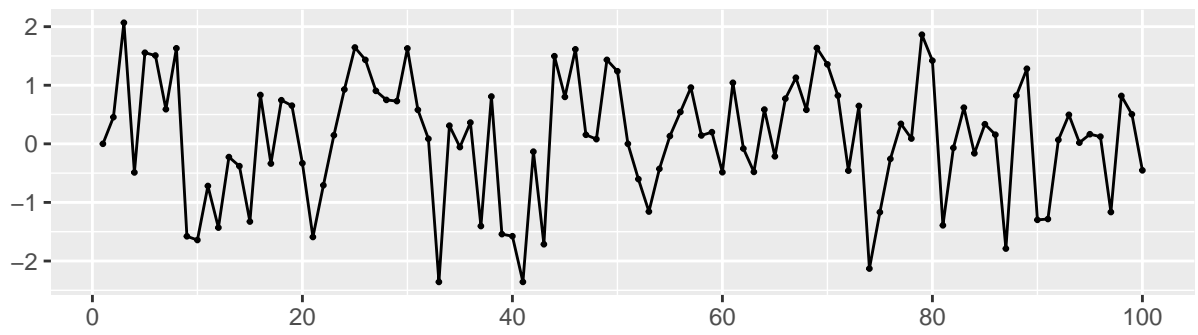
AR(1) series, $\phi_1 = -0.1$



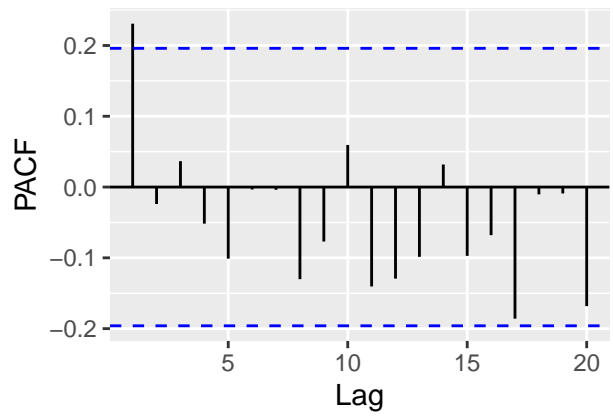
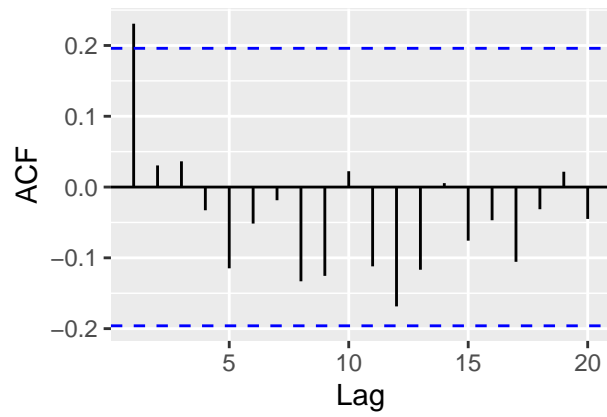
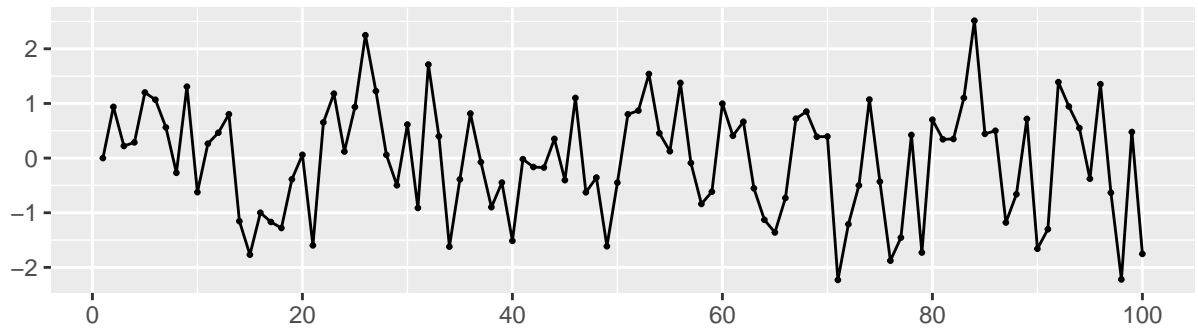
AR(1) series, $\phi_1 = 0$



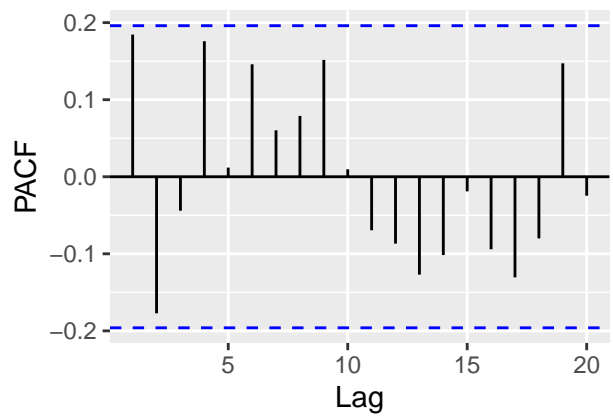
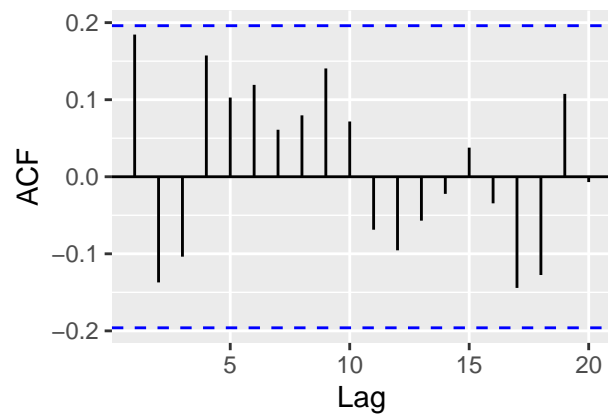
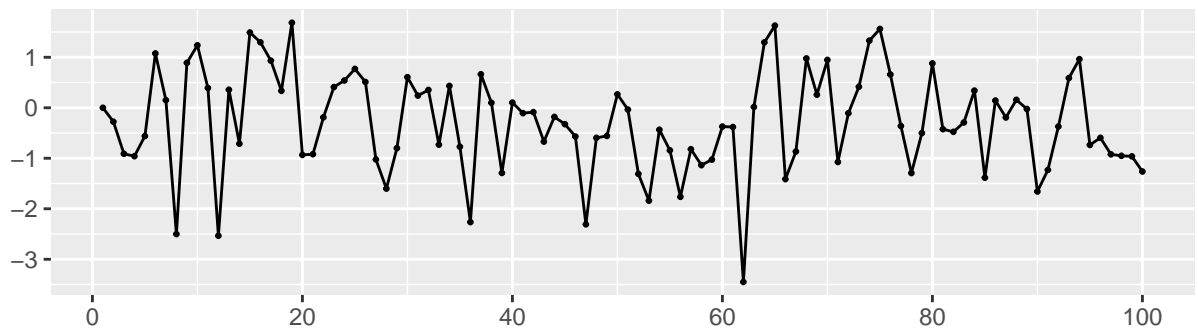
AR(1) series, $\phi_1 = 0.1$



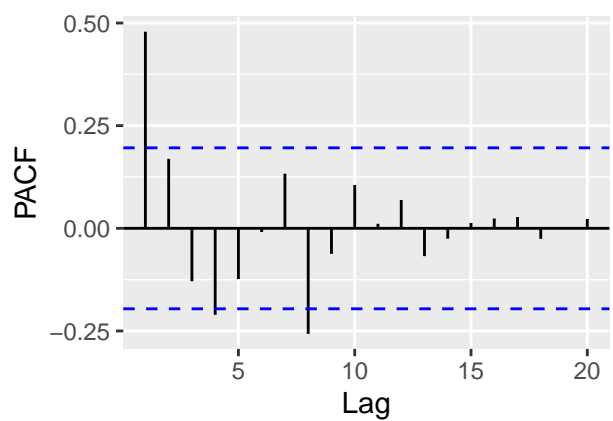
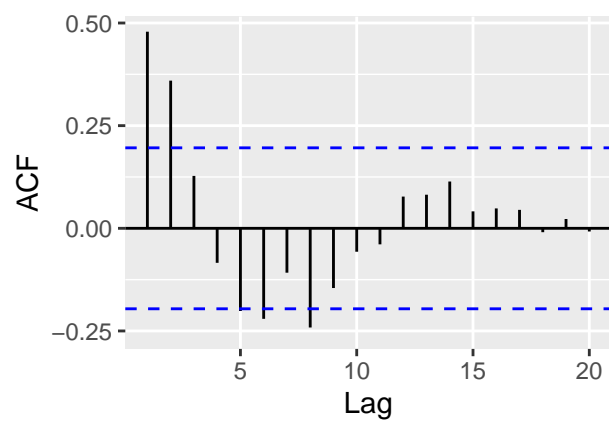
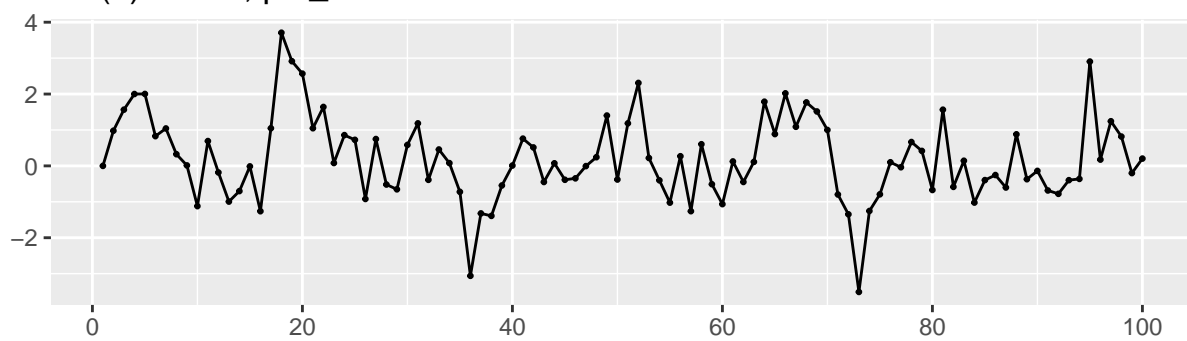
AR(1) series, $\phi_1 = 0.2$



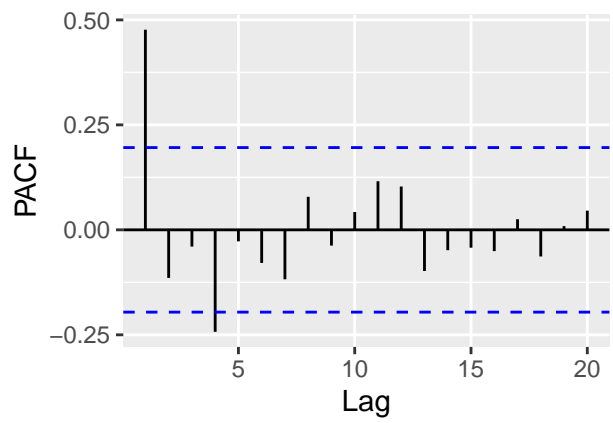
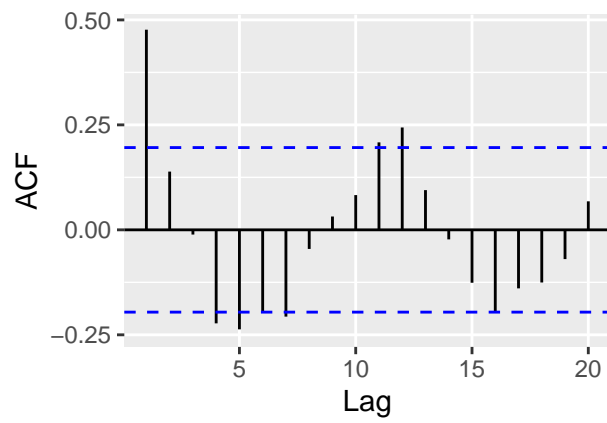
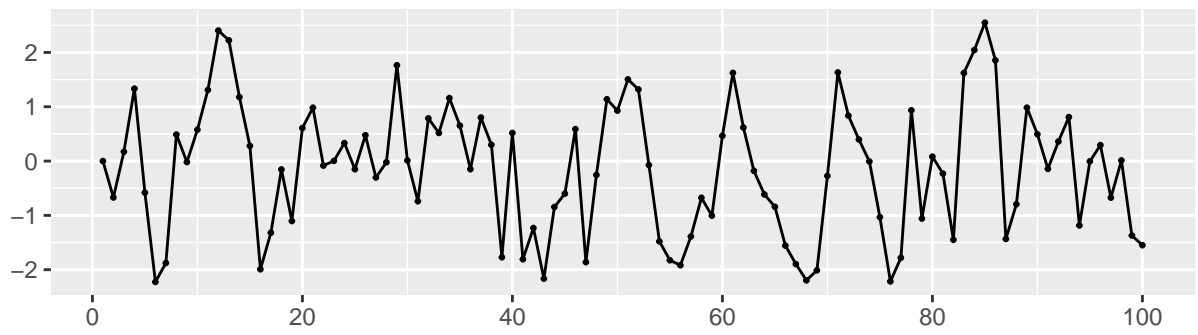
AR(1) series, $\phi_1 = 0.3$



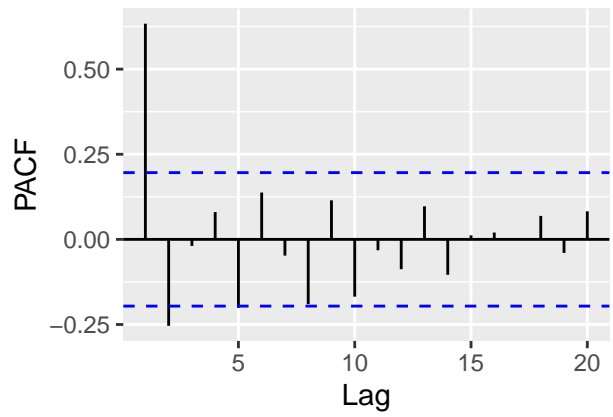
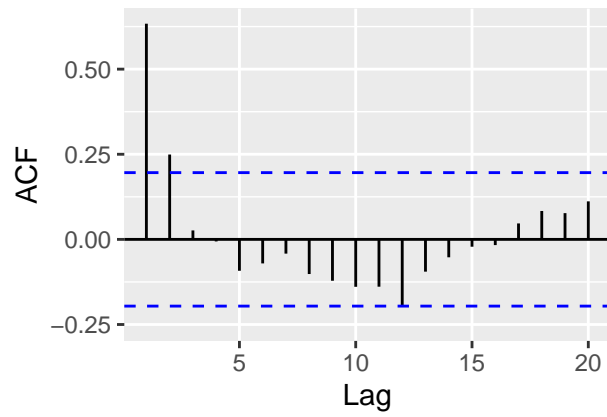
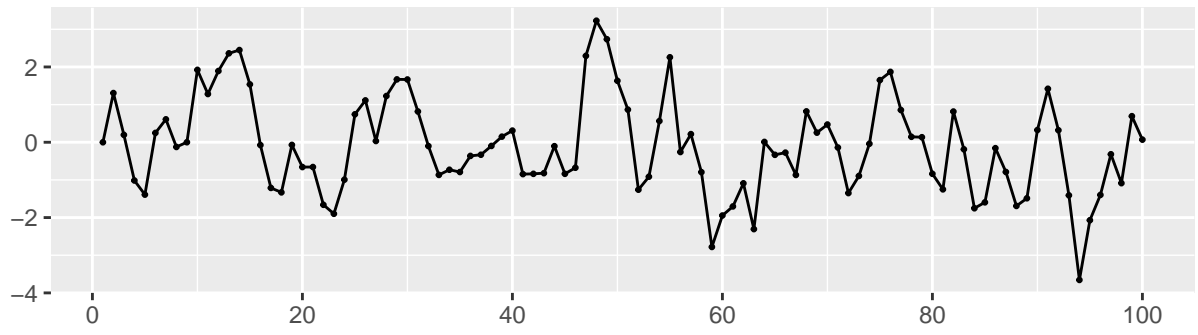
AR(1) series, $\phi_1 = 0.4$



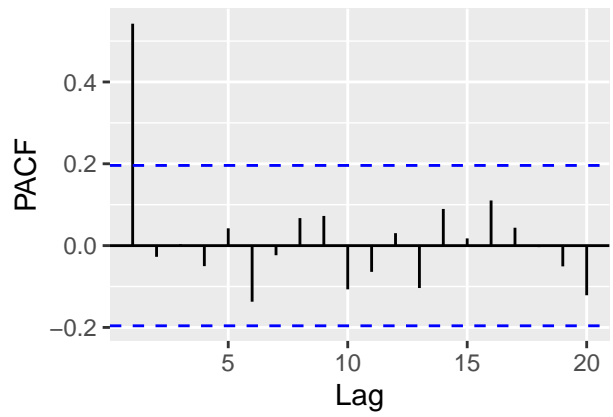
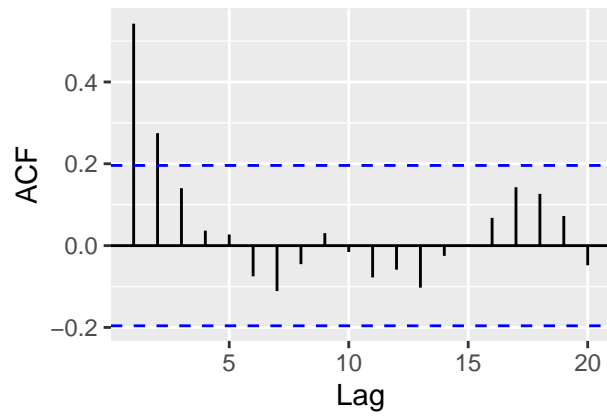
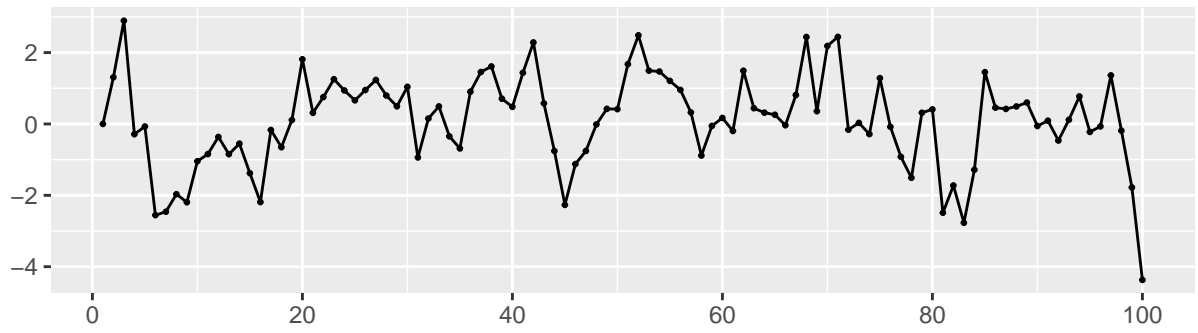
AR(1) series, $\phi_1 = 0.5$



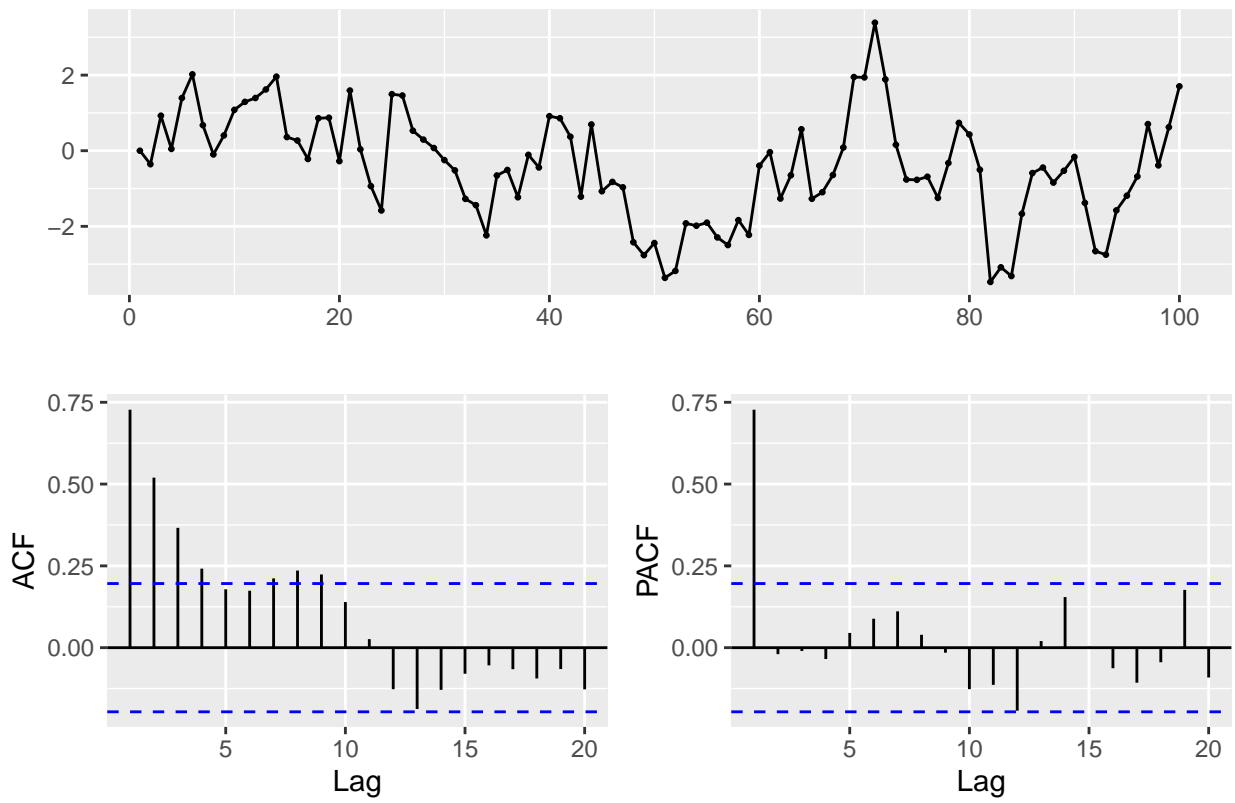
AR(1) series, $\phi_1 = 0.6$



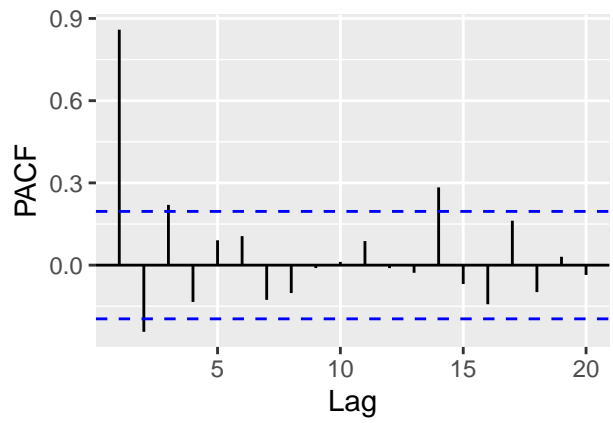
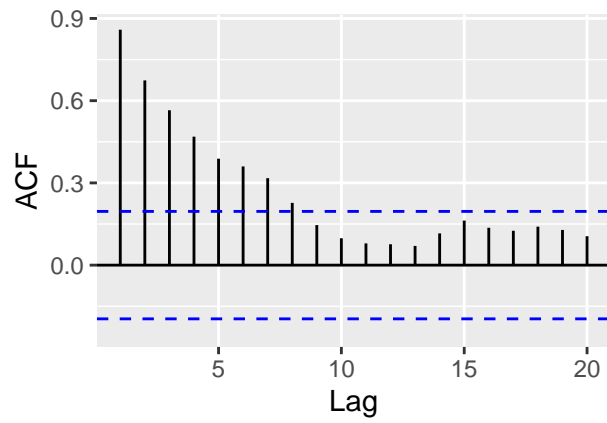
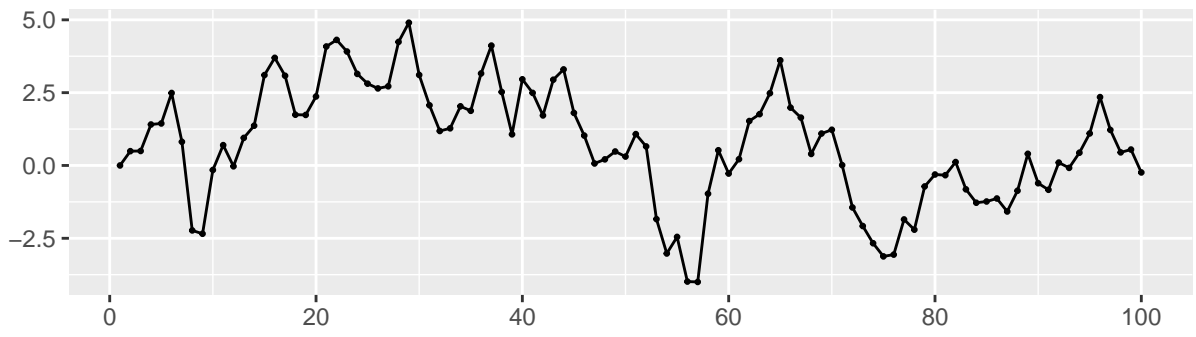
AR(1) series, $\phi_1 = 0.7$

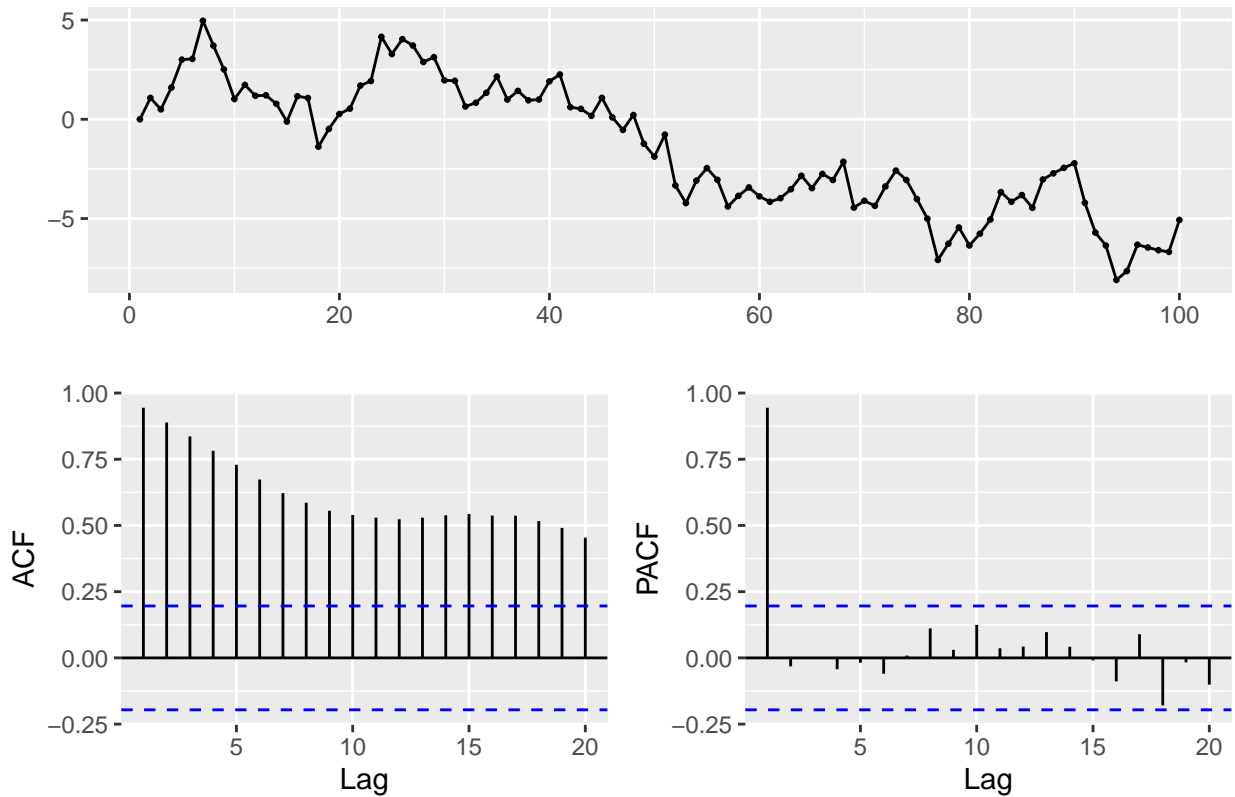


AR(1) series, $\phi_1 = 0.8$



AR(1) series, $\phi_1 = 0.9$



AR(1) series, $\phi_1 = 1$ 

AR1_stats_results

##	Phi_1	Min.	1st Qu.	Median	Mean	3rd Qu.
## AR1_stats	-1.0	-4.810328	-1.5376074	0.025282049	0.137653162	1.7864250
## AR1_stats	-0.9	-4.158772	-1.3747743	-0.028912600	0.013688760	1.2591337
## AR1_stats	-0.8	-3.154382	-1.1887112	-0.077140387	-0.007503376	1.1200676
## AR1_stats	-0.7	-3.410915	-0.7810173	0.084770362	0.117672941	1.0720622
## AR1_stats	-0.6	-1.936819	-0.5516114	-0.103689483	-0.034933100	0.5488060
## AR1_stats	-0.5	-2.776452	-0.8055119	0.046569953	0.049196008	0.8492201
## AR1_stats	-0.4	-3.364833	-0.7039008	-0.008652004	0.007795845	0.8300460
## AR1_stats	-0.3	-1.989033	-0.5633470	0.153928011	0.149565675	0.6968589
## AR1_stats	-0.2	-2.578660	-0.7023718	0.091290686	-0.013764807	0.7590138
## AR1_stats	-0.1	-2.560052	-0.6760593	-0.238899537	-0.171245734	0.4731843
## AR1_stats	0.0	-2.402710	-0.8855039	-0.268306656	-0.201570142	0.4703282
## AR1_stats	0.1	-2.357408	-0.4635547	0.150371587	0.098657753	0.8104388
## AR1_stats	0.2	-2.229606	-0.6793502	0.090601453	-0.028728454	0.7184421
## AR1_stats	0.3	-3.449043	-0.9202869	-0.307662728	-0.278832152	0.3961435
## AR1_stats	0.4	-3.509097	-0.5253735	0.078577201	0.198240741	0.8810261
## AR1_stats	0.5	-2.225483	-1.0705274	-0.020806114	-0.128432926	0.6265709
## AR1_stats	0.6	-3.656013	-0.8997050	-0.114304599	-0.099464438	0.7050210
## AR1_stats	0.7	-4.371388	-0.5770992	0.213402964	0.097913167	0.9130002
## AR1_stats	0.8	-3.470953	-1.2666898	-0.419136558	-0.417538751	0.5820123
## AR1_stats	0.9	-3.998703	-0.3161399	0.754792592	0.754857980	2.1362739
## AR1_stats	1.0	-8.094175	-4.0404369	-1.632484191	-1.463852356	1.1023804
##	Max.	StDev	SdDiff			
## AR1_stats	4.666069	2.3355689	4.532225			

```

## AR1_stats 3.484262 1.6924696 3.223143
## AR1_stats 3.458808 1.4961145 2.826108
## AR1_stats 3.341460 1.3827348 2.563901
## AR1_stats 2.365774 0.9396446 1.577675
## AR1_stats 4.229266 1.2123269 2.077458
## AR1_stats 2.681672 1.2584885 2.275212
## AR1_stats 2.624459 0.9596337 1.427915
## AR1_stats 2.498753 0.9906810 1.612124
## AR1_stats 1.943005 0.9248660 1.306092
## AR1_stats 2.457074 0.9453084 1.400122
## AR1_stats 2.067448 1.0364809 1.282434
## AR1_stats 2.514284 1.0235093 1.263902
## AR1_stats 1.684784 0.9952520 1.273267
## AR1_stats 3.708281 1.1618359 1.192095
## AR1_stats 2.547435 1.1896935 1.214698
## AR1_stats 3.228343 1.2756715 1.097696
## AR1_stats 2.893791 1.2866680 1.150978
## AR1_stats 3.386849 1.4147713 1.026975
## AR1_stats 4.901704 1.9752239 1.046856
## AR1_stats 4.966340 3.2299537 1.005720

```

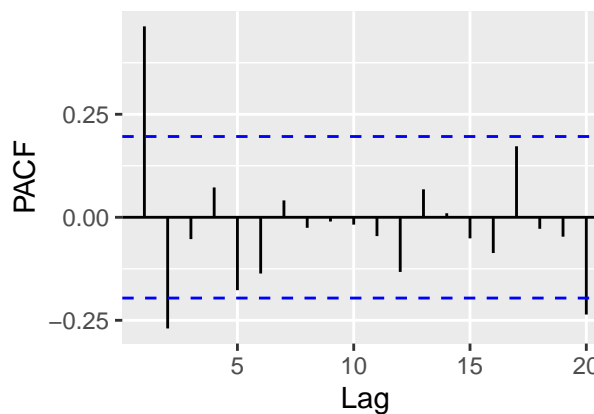
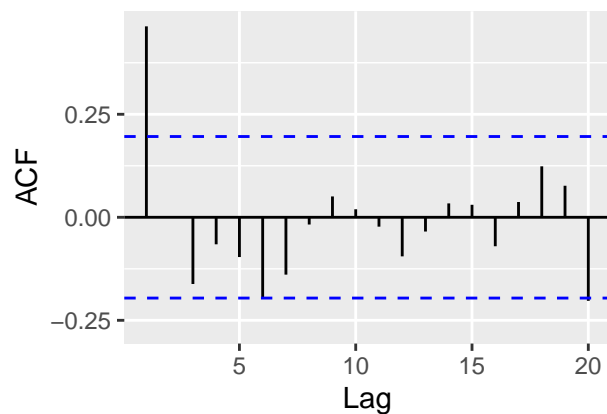
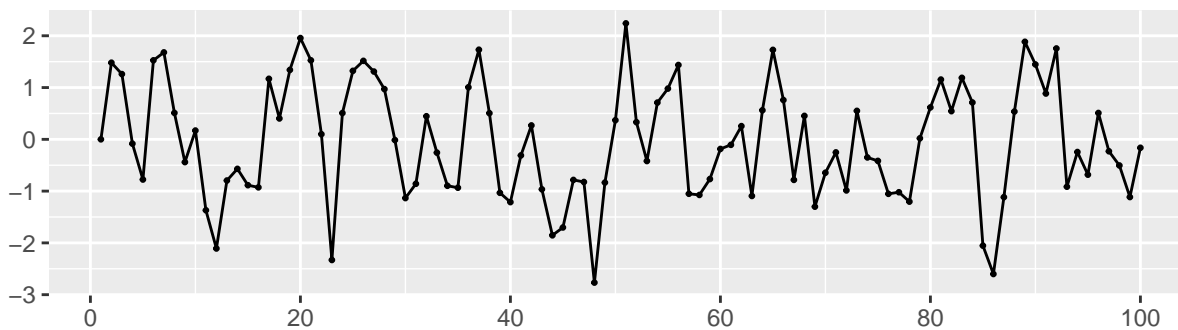
- When $\phi_1 < 0$, y_t tends to oscillate around the mean.
- As ϕ_1 becomes close to -1, the series oscillates more sharply, resulting in a higher standard deviation both of the series and of its changes
- When $\phi_1 = 0$, y_t is equivalent to white noise.
- As ϕ_1 increases toward 1, the standard deviation of the series increases, while the standard deviation of the changes moves closer to 1.
- when $\phi_1 = 1$, y_t is equivalent to a random walk.

c) Write your own code to generate data from an MA(1) model with $\theta_1 = 0.6$ and $\sigma^2 = 1$.

```
MA1 <- function(theta_1)
{
  y <- ts(numeric(100))
  e <- rnorm(100)
  for(i in 2:100)
    y[i] <- e[i] + theta_1*e[i-1]
  return(y)
}

theta_1 <- 0.60
ggtsdisplay(MA1(theta_1),main=paste("MA(1) series, theta_1=",theta_1))
```

MA(1) series, theta_1= 0.6

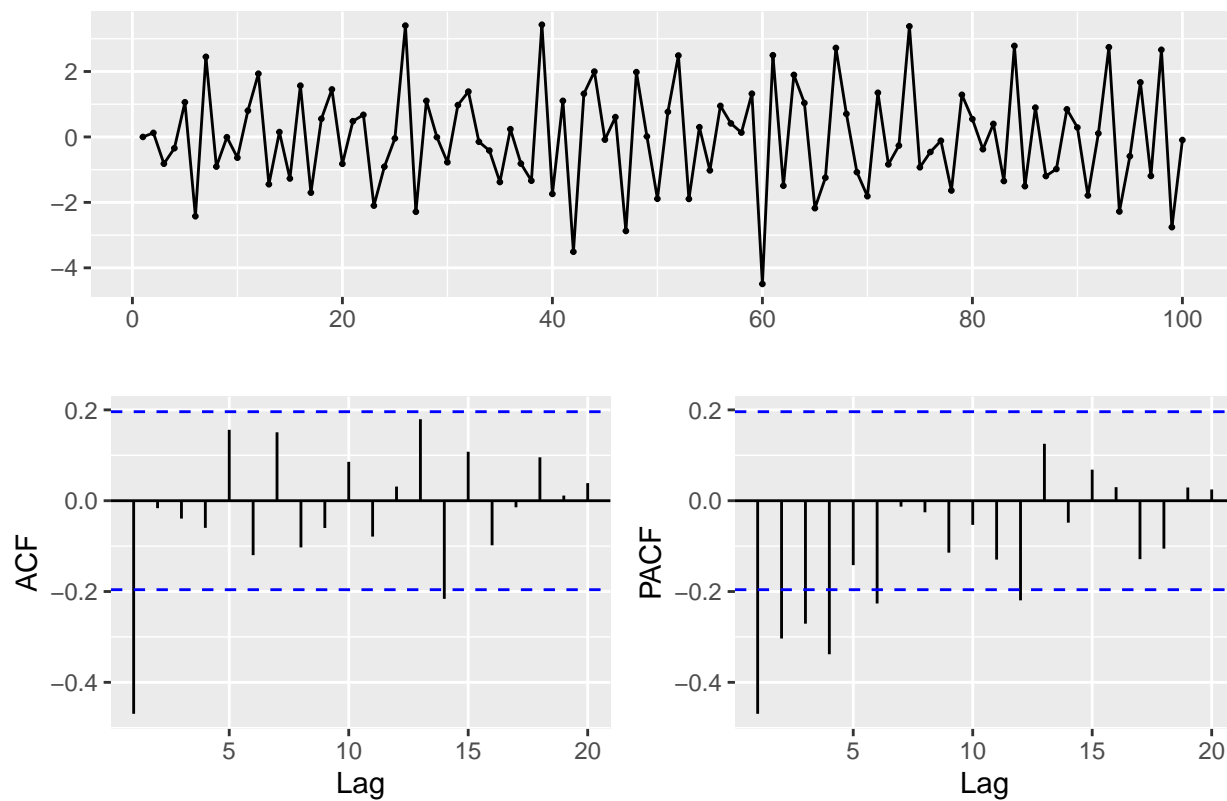


d) Produce a time plot for the series. How does the plot change as you change θ_1 ?

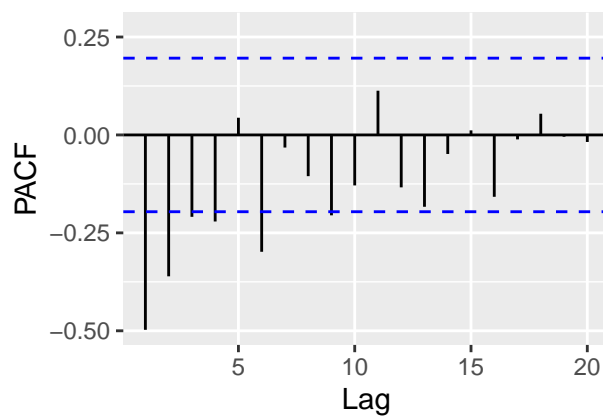
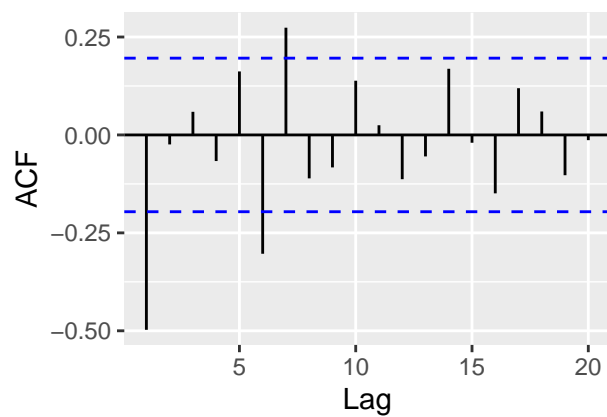
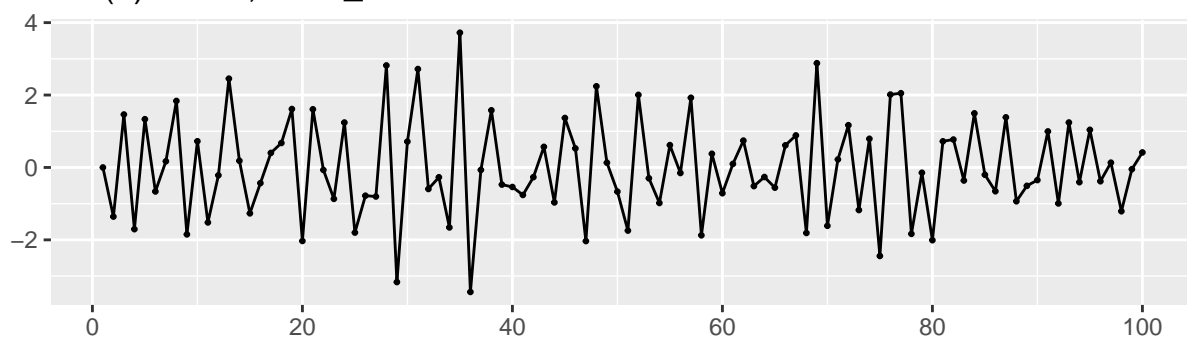
```
set.seed(12345)
MA1_stats_results = numeric()
theta_seq <- seq(from=-1,to=1,by = 0.1)

for (theta_1 in theta_seq) {
  MA1_series <- MA1(theta_1)
  MA1_stats <- c(theta_1 = theta_1, summary(MA1_series), StDev=sd(MA1_series), SdDiff=sd(diff(MA1_series)))
  MA1_main <- paste("MA(1) series, theta_1=", theta_1)
  ggtsdisplay(MA1_series, main=MA1_main)
  #print(MA1_stats)
  MA1_stats_results <- rbind(MA1_stats_results, MA1_stats)
}
```

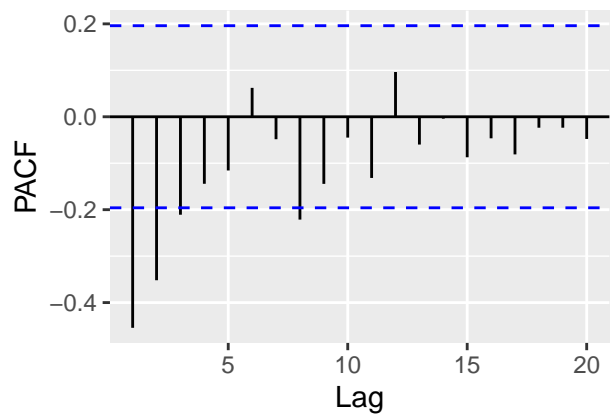
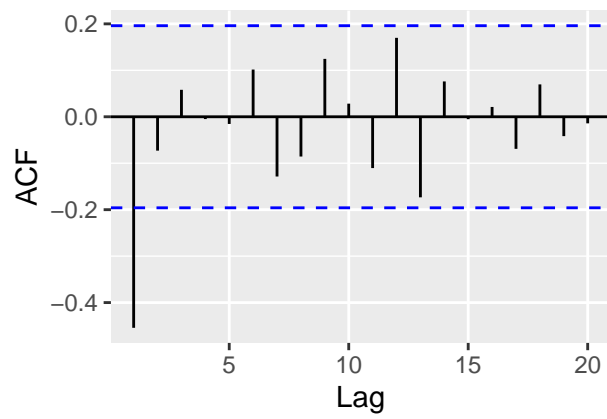
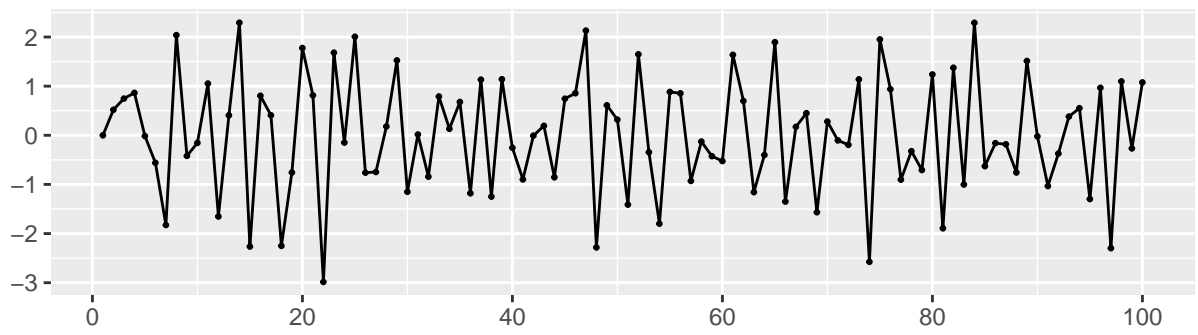
MA(1) series, $\theta_1 = -1$



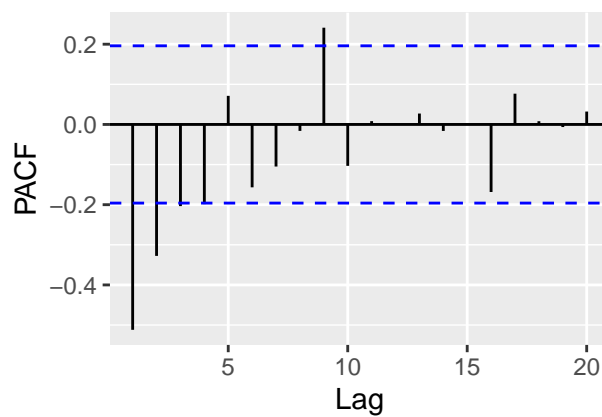
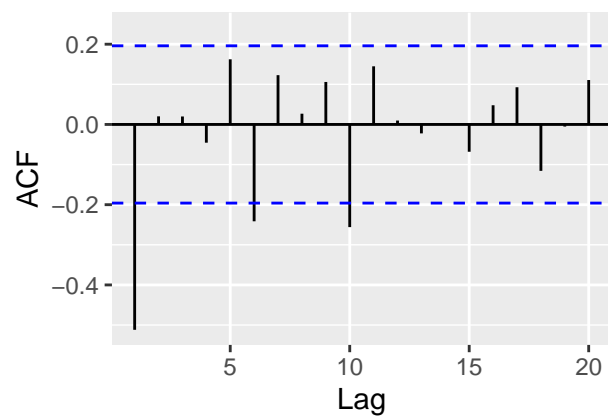
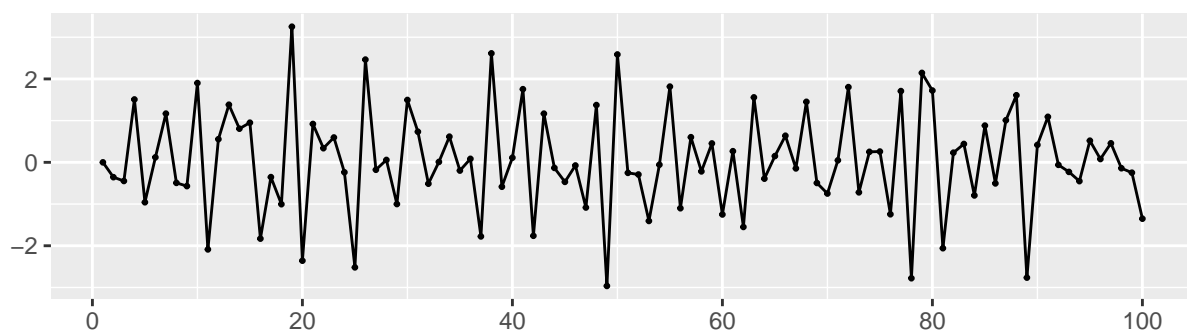
MA(1) series, $\theta_1 = -0.9$



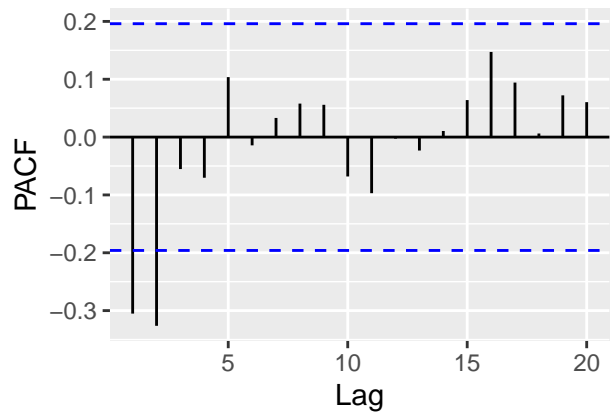
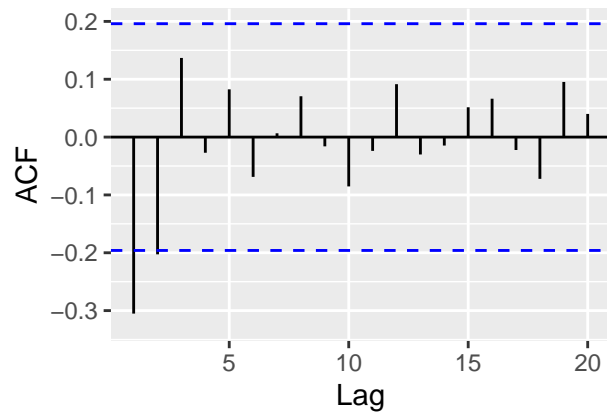
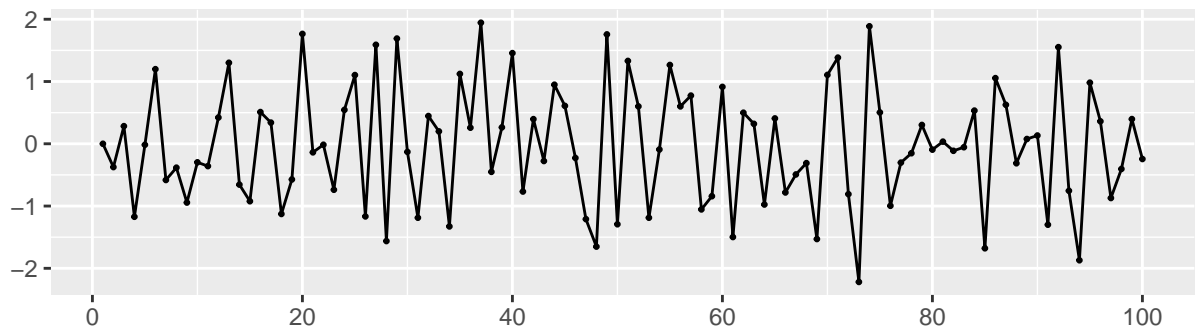
MA(1) series, $\theta_1 = -0.8$



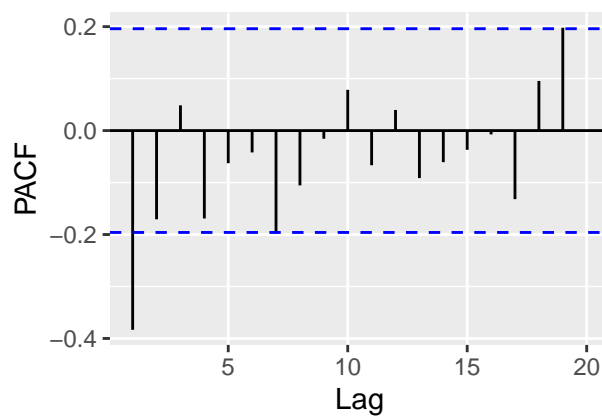
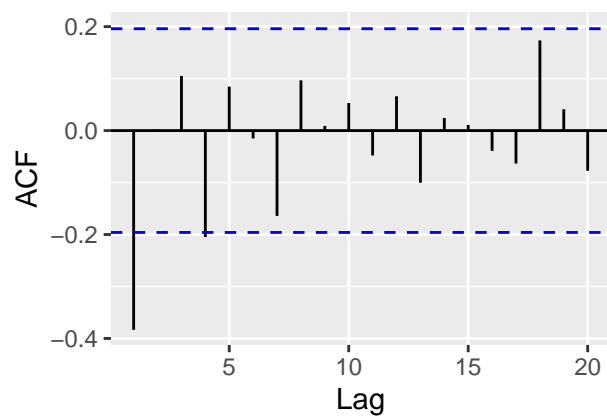
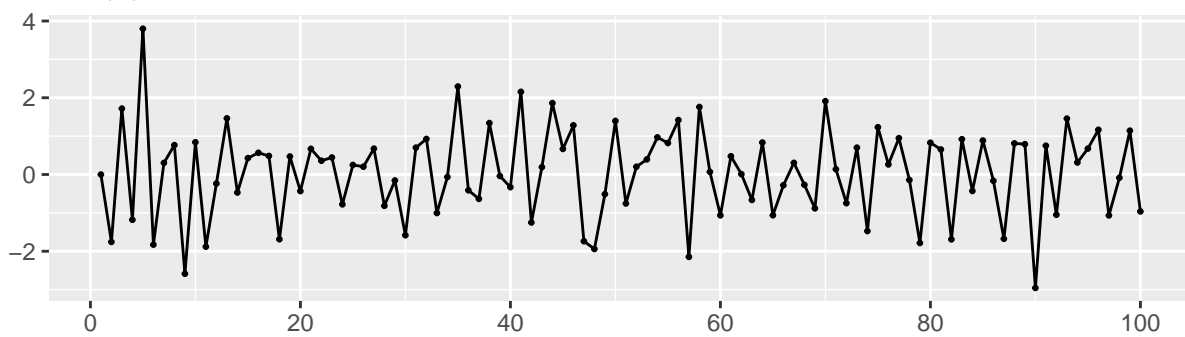
MA(1) series, $\theta_1 = -0.7$



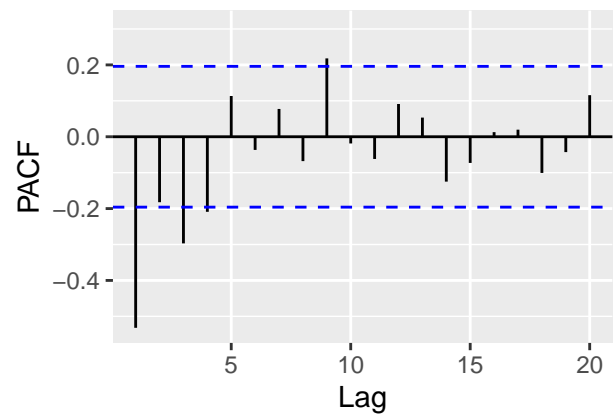
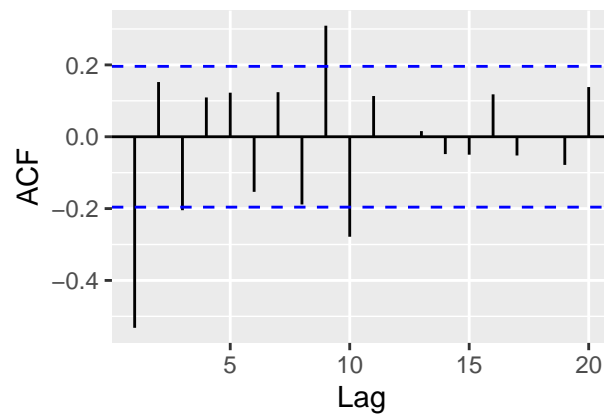
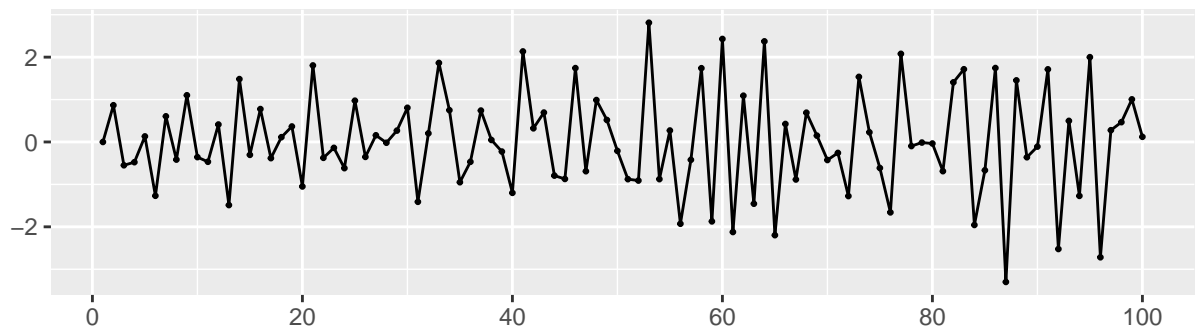
MA(1) series, $\theta_1 = -0.6$



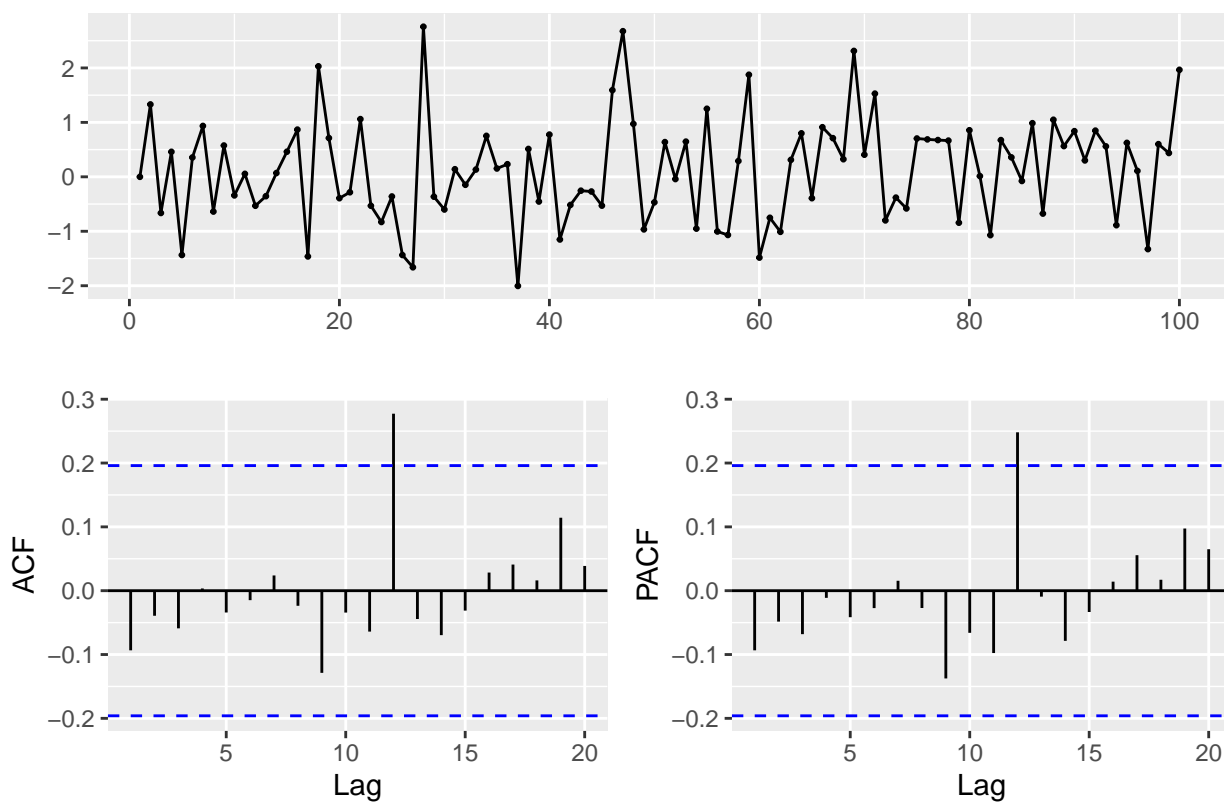
MA(1) series, $\theta_1 = -0.5$



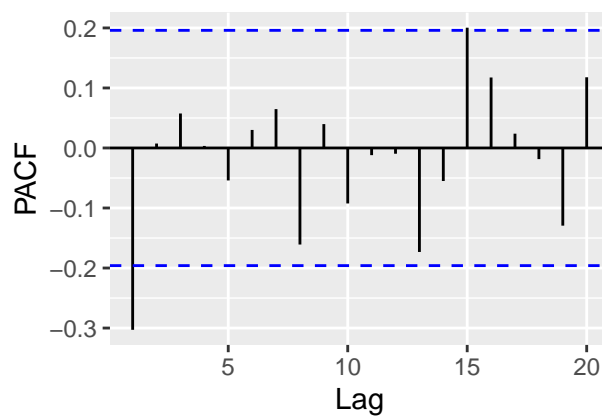
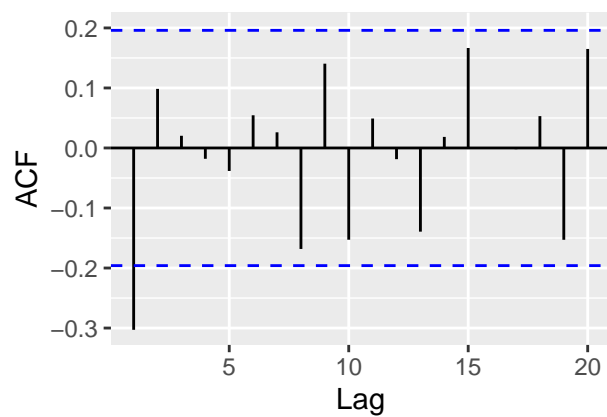
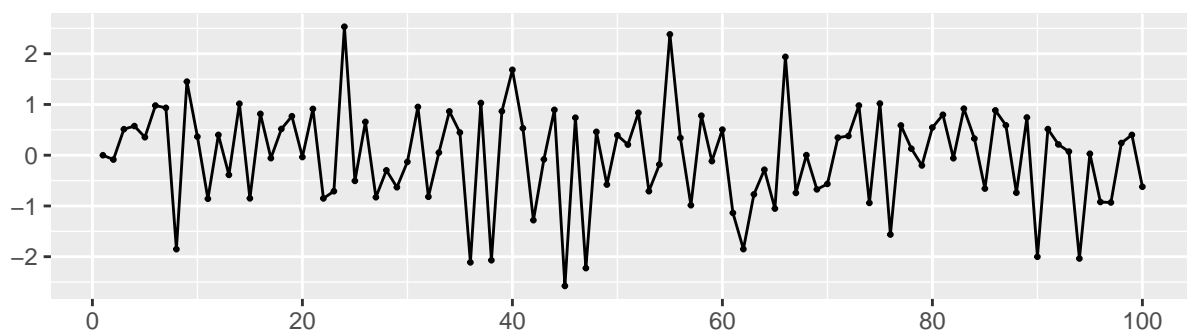
MA(1) series, $\theta_1 = -0.4$



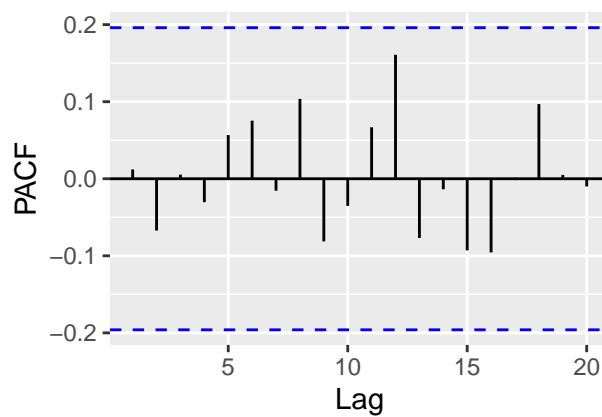
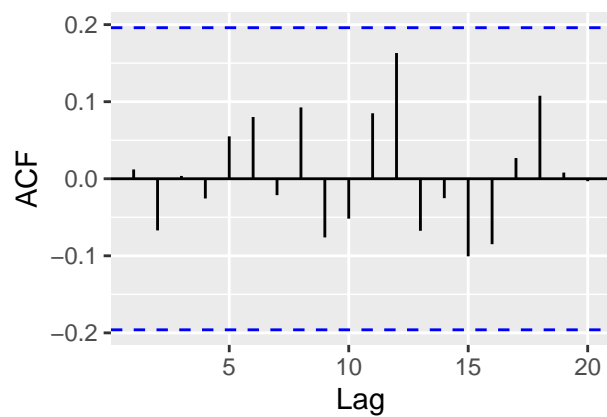
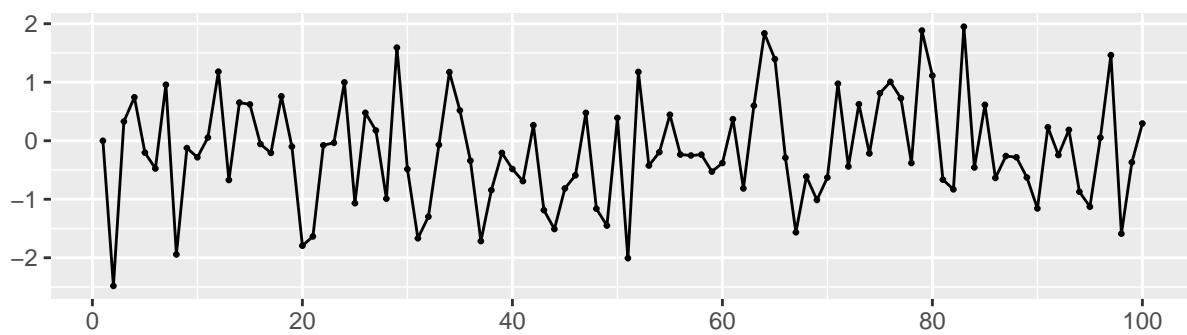
MA(1) series, $\theta_1 = -0.3$



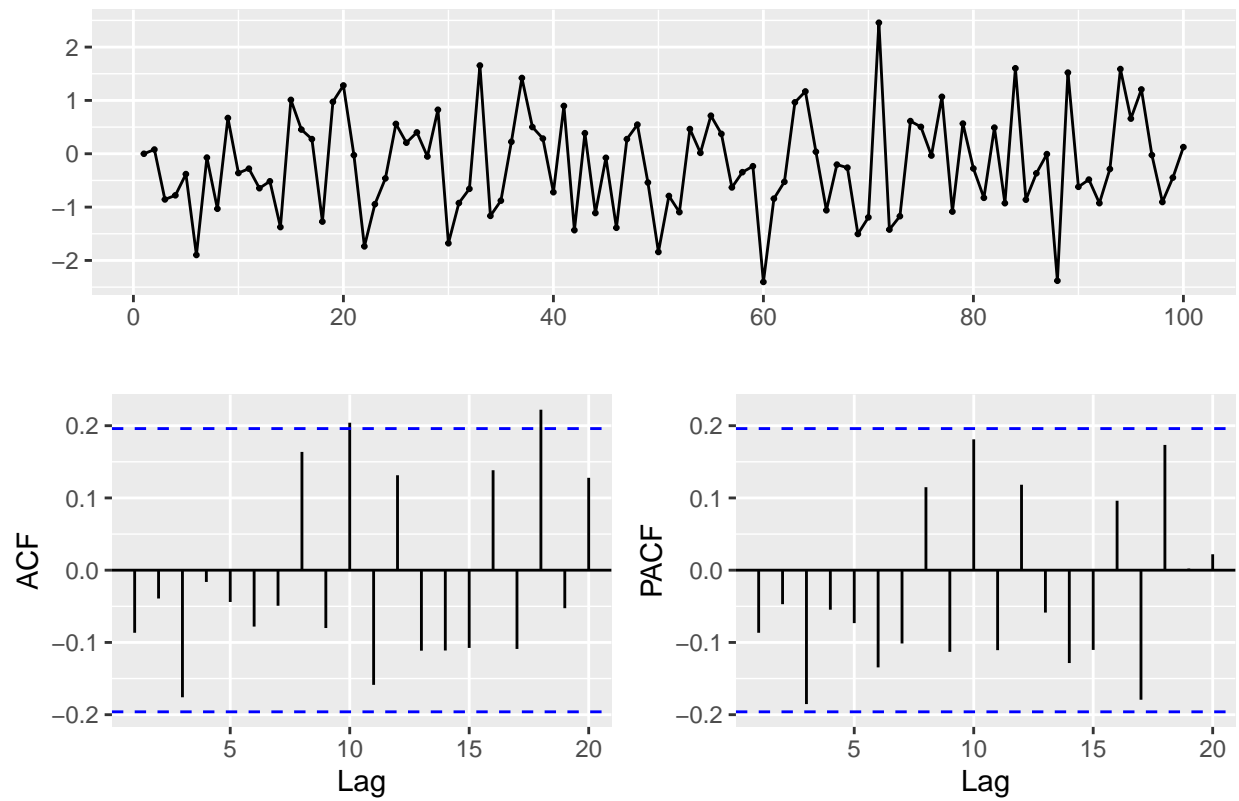
MA(1) series, $\theta_1 = -0.2$



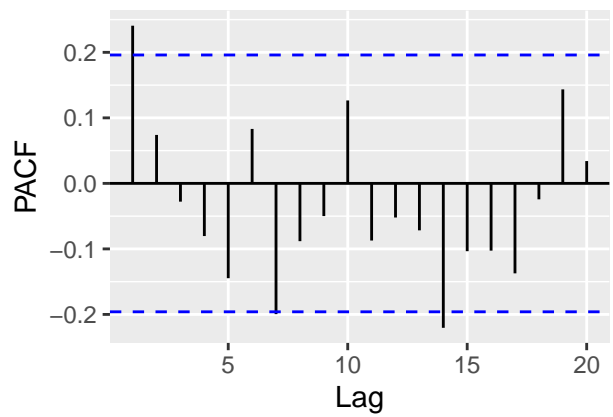
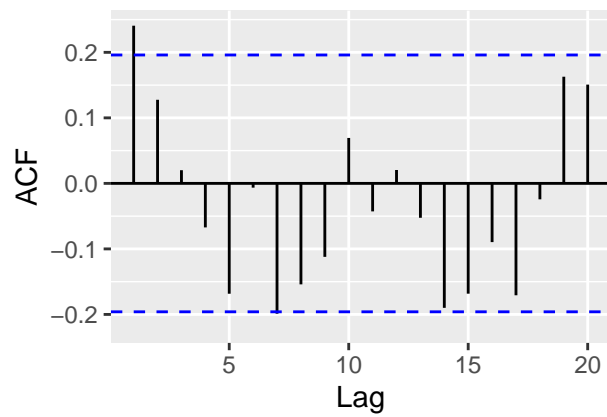
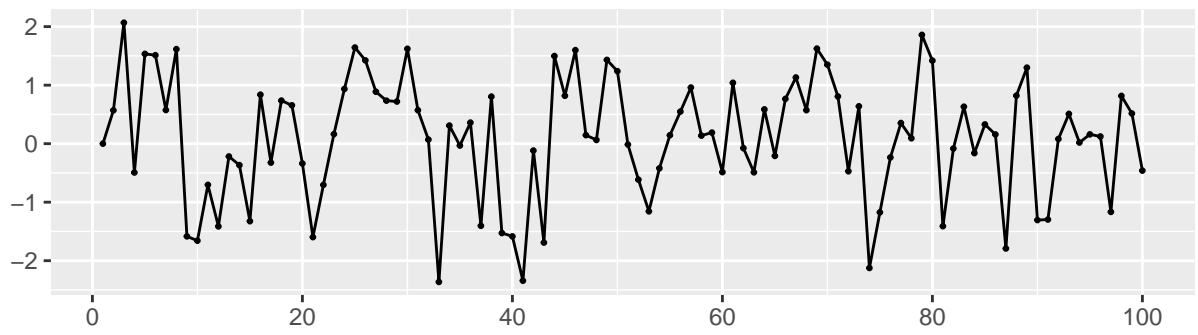
MA(1) series, $\theta_1 = -0.1$



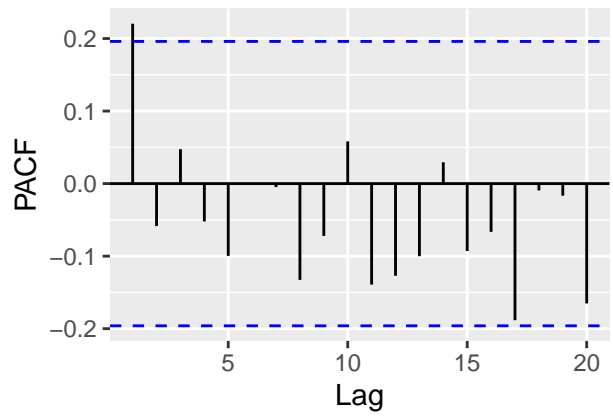
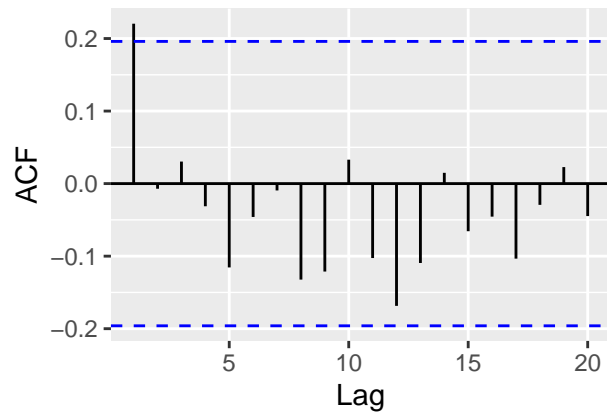
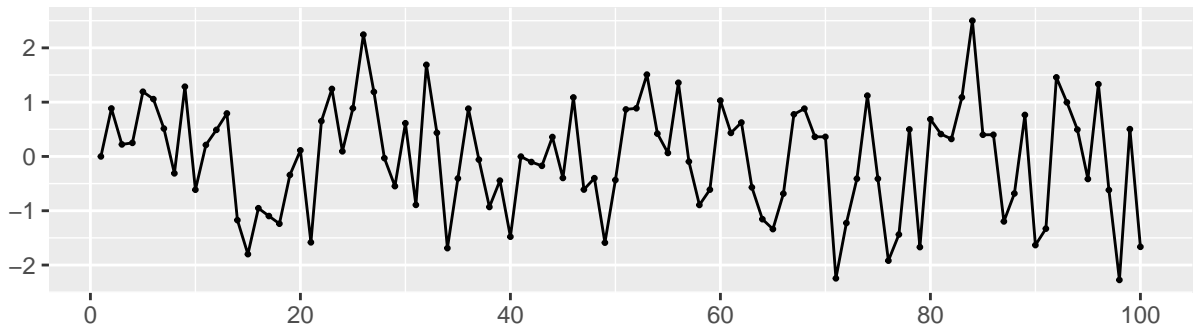
MA(1) series, $\theta_1 = 0$



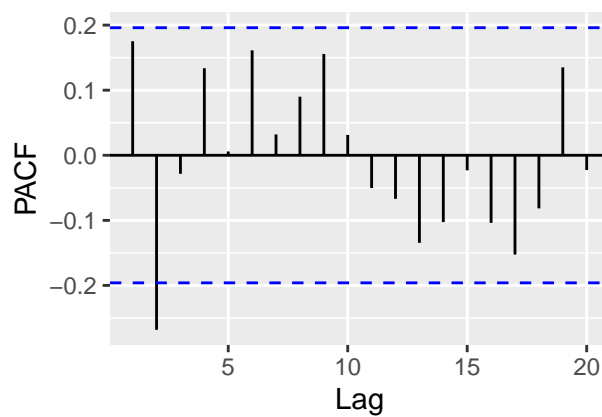
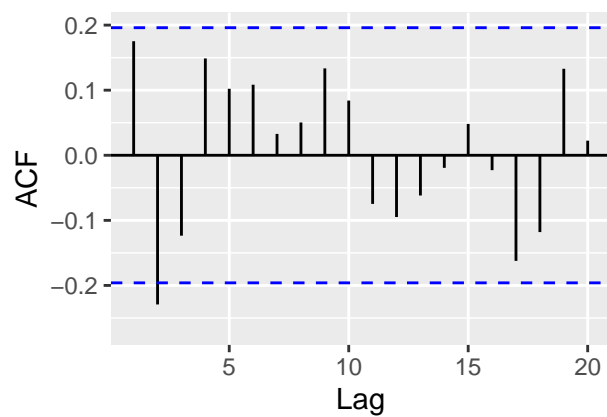
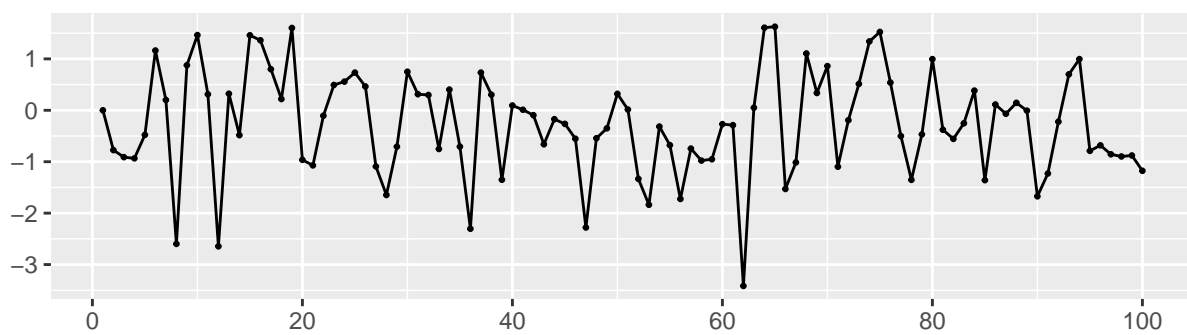
MA(1) series, $\theta_1 = 0.1$



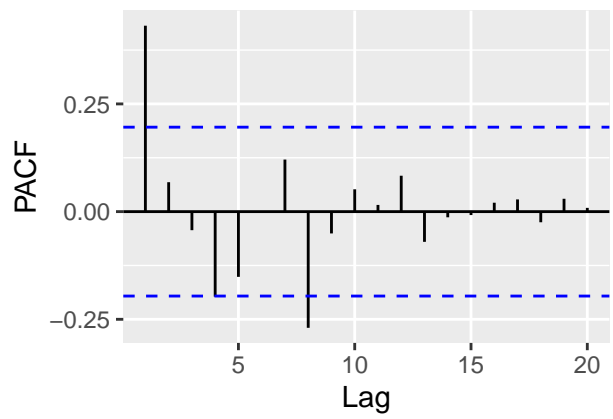
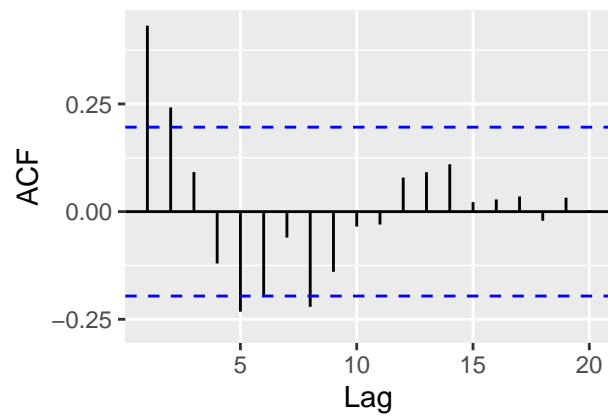
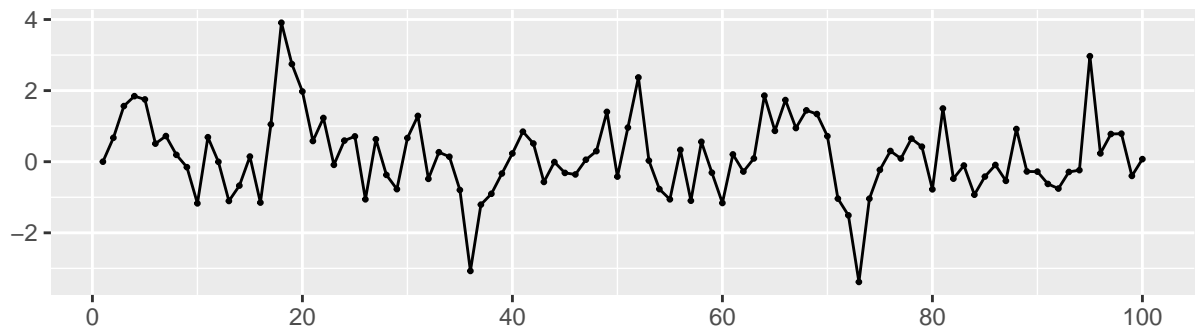
MA(1) series, $\theta_1 = 0.2$



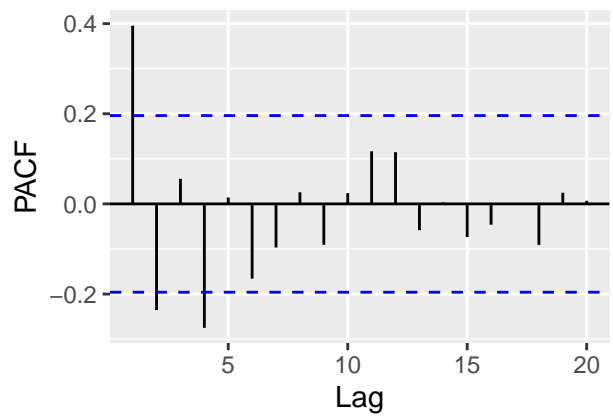
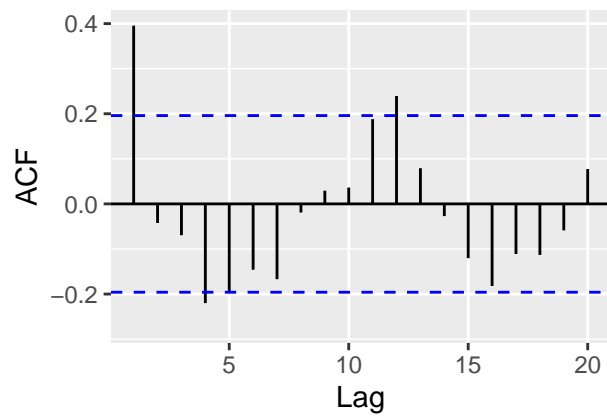
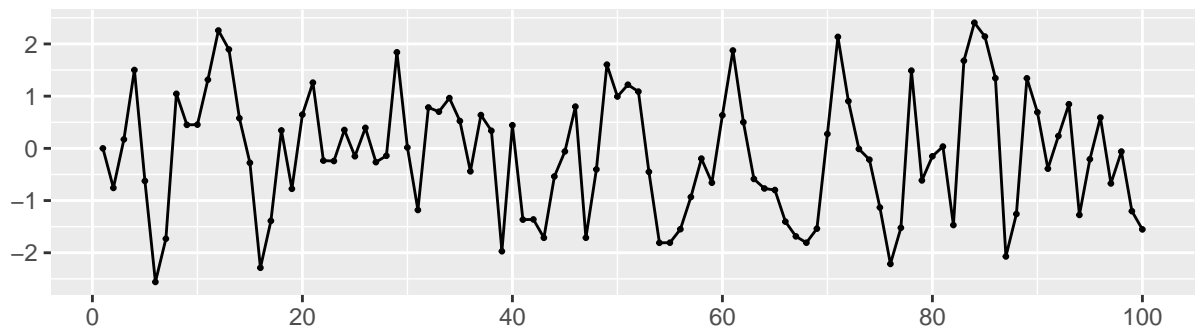
MA(1) series, $\theta_1 = 0.3$



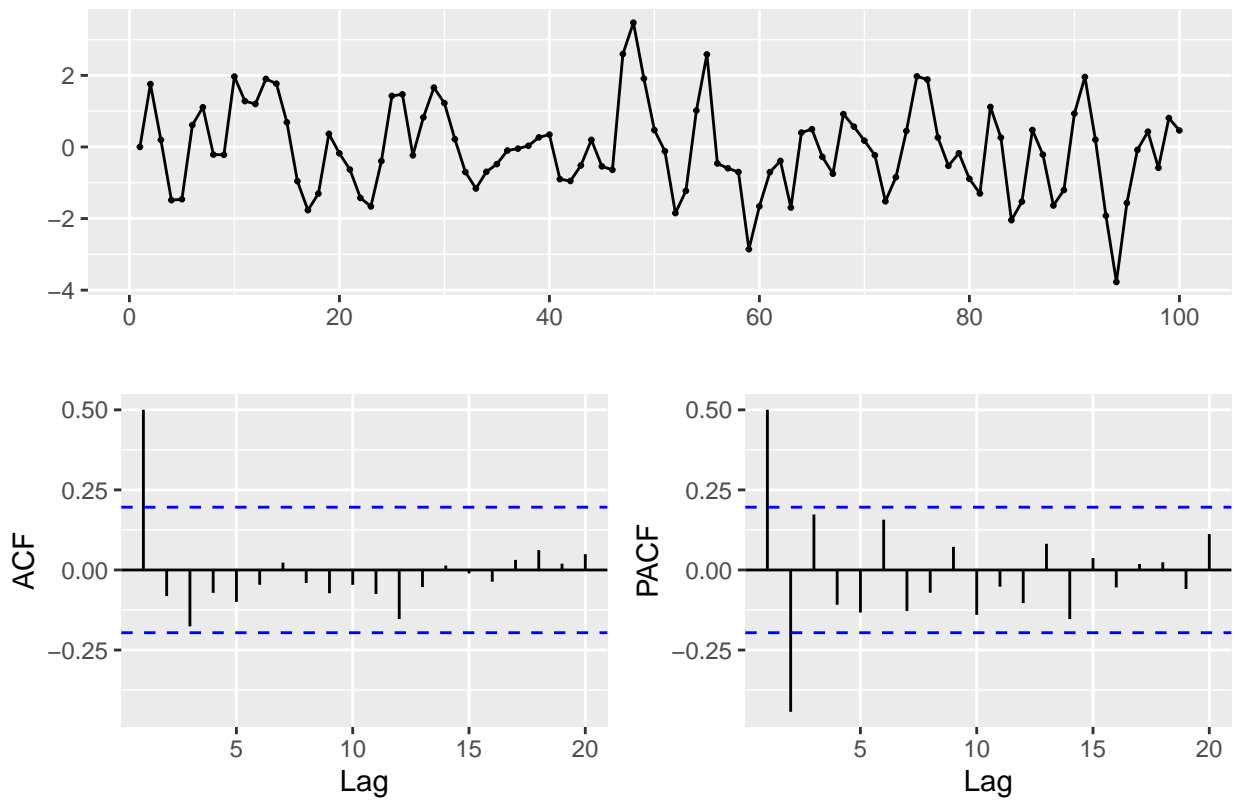
MA(1) series, $\theta_1 = 0.4$



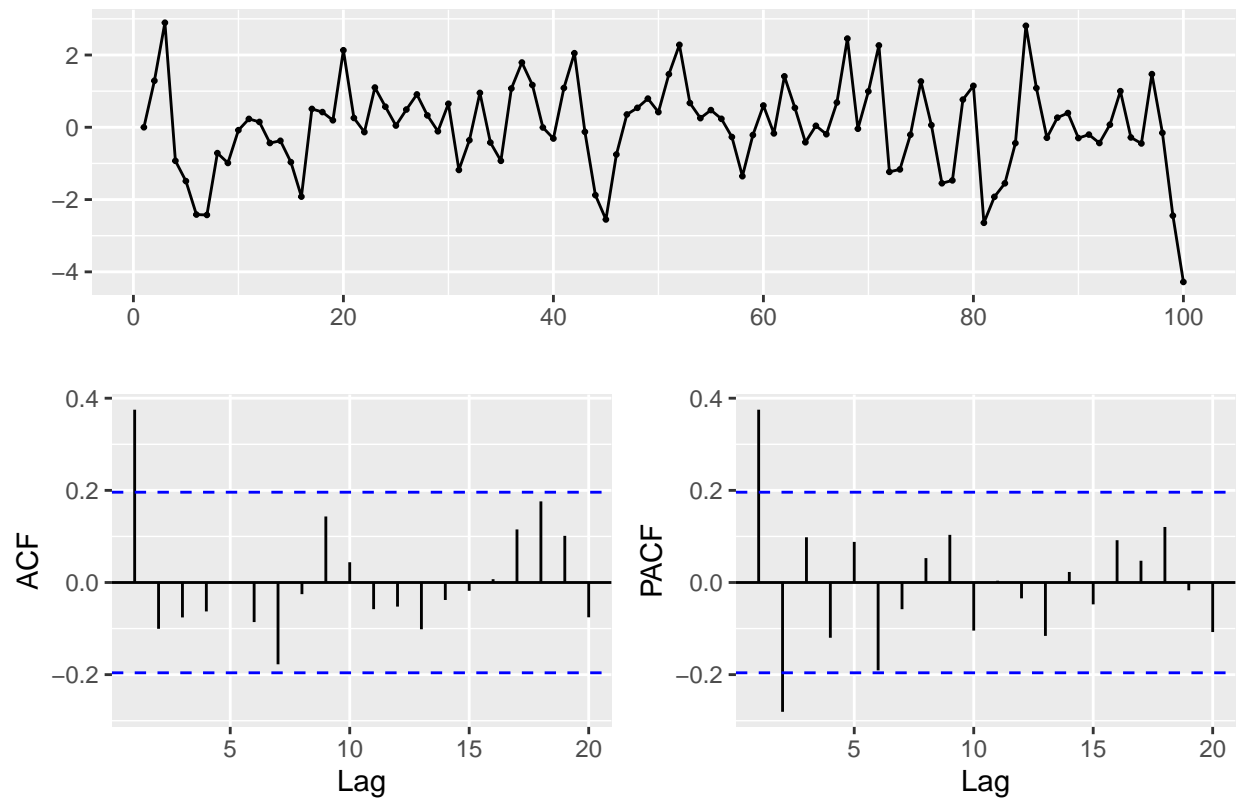
MA(1) series, $\theta_1 = 0.5$



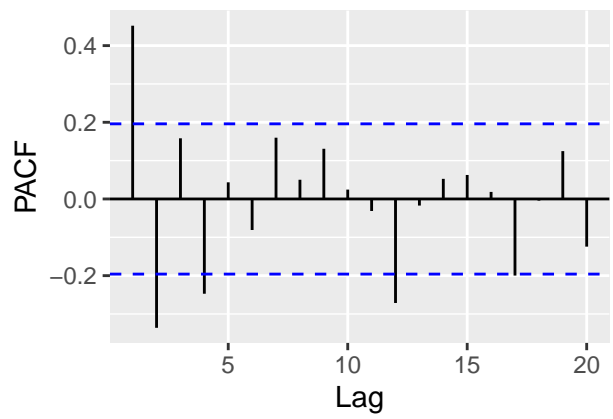
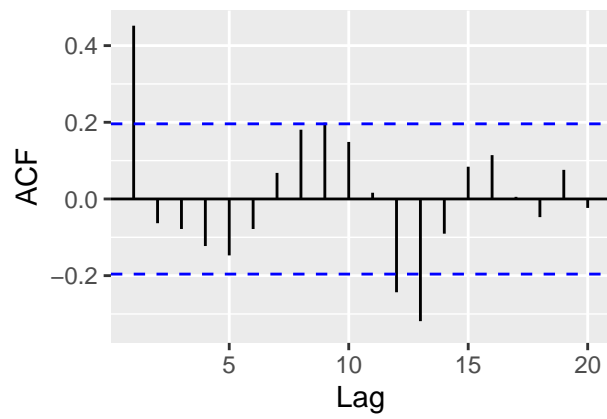
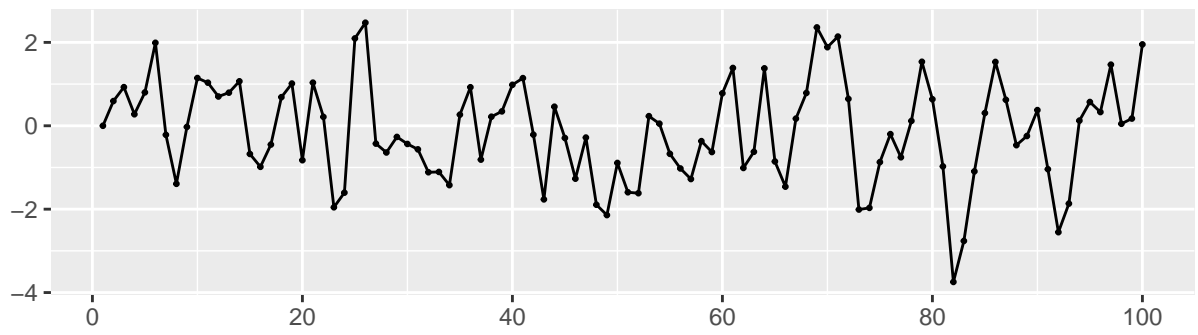
MA(1) series, $\theta_1 = 0.6$



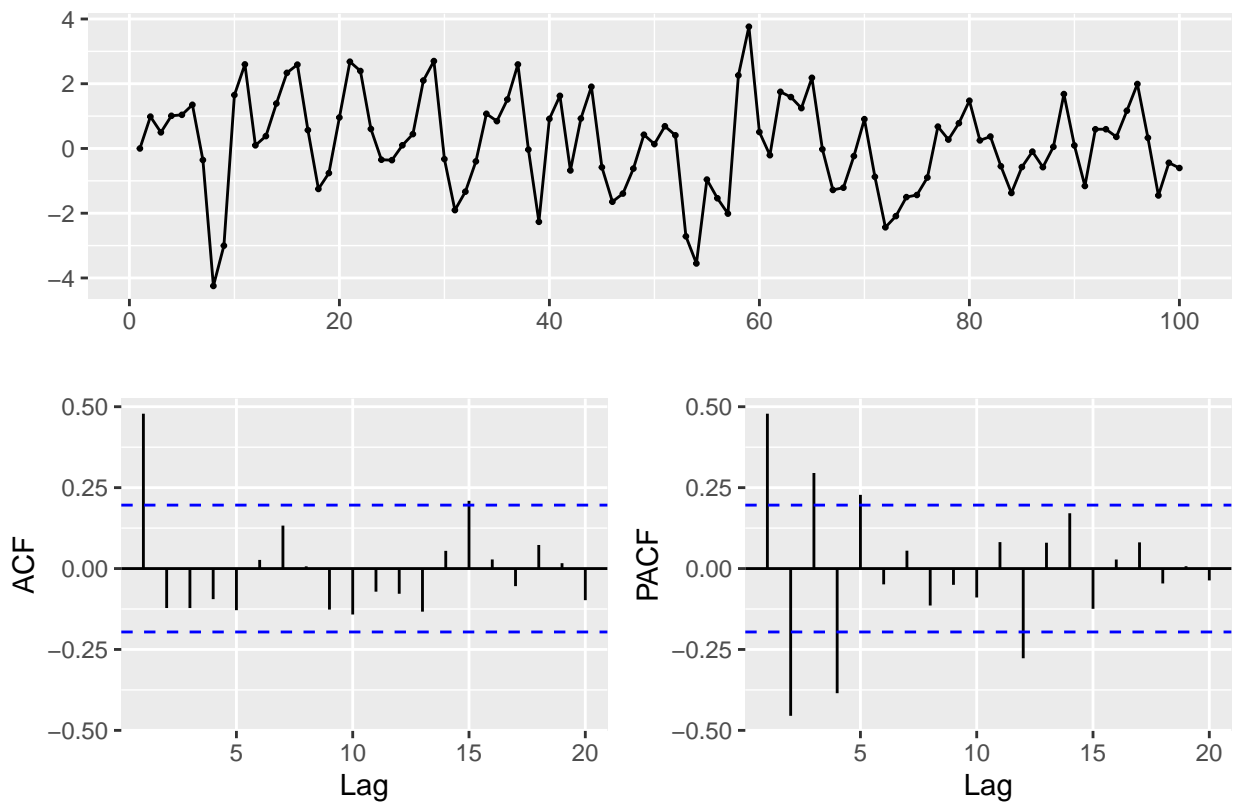
MA(1) series, $\theta_1 = 0.7$



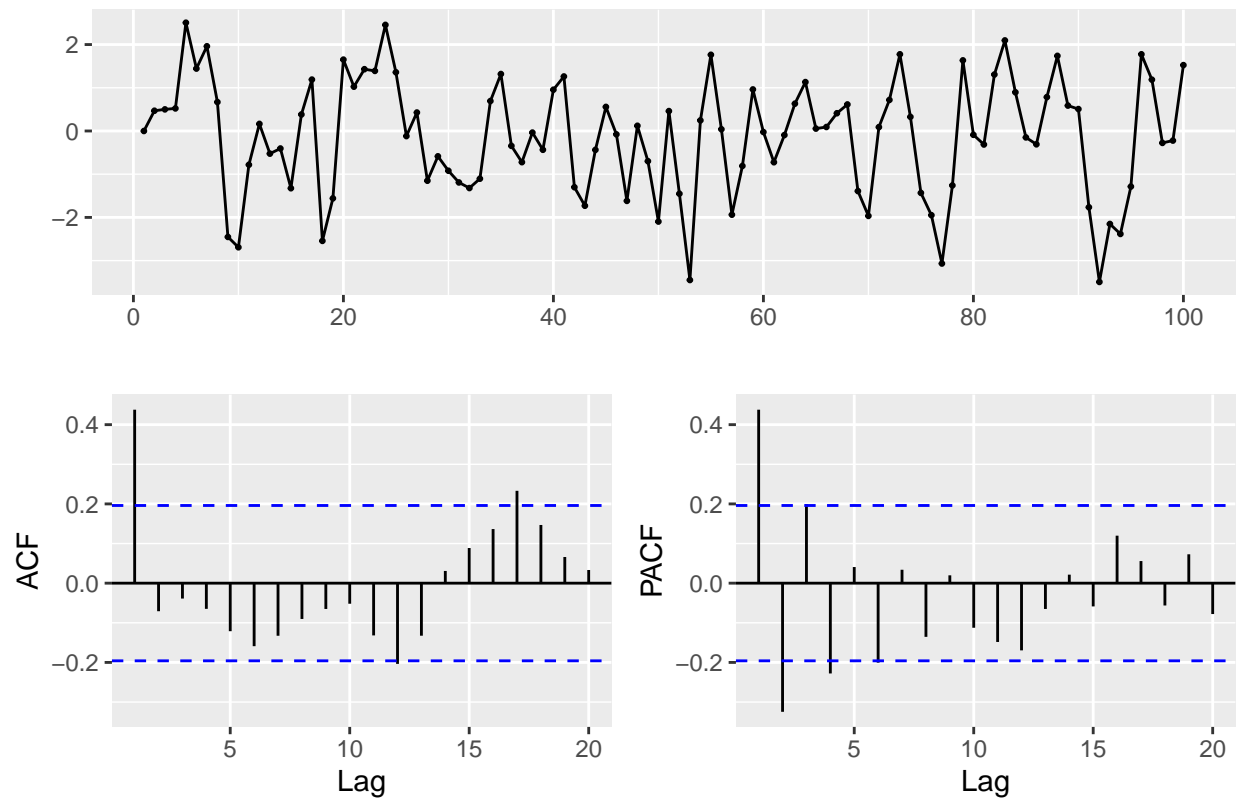
MA(1) series, $\theta_1 = 0.8$



MA(1) series, $\theta_1 = 0.9$



MA(1) series, theta_1= 1



MA1_stats_results

##	theta_1	Min.	1st Qu.	Median	Mean	3rd Qu.
## MA1_stats	-1.0	-4.492009	-1.1930459	-0.008115974	-0.012800755	1.0423410
## MA1_stats	-0.9	-3.438123	-0.8186108	-0.149618065	0.002403803	0.8154866
## MA1_stats	-0.8	-2.984302	-0.7811010	-0.011340950	0.008833616	0.8580352
## MA1_stats	-0.7	-2.963578	-0.5265045	0.004962544	0.049736692	0.8252867
## MA1_stats	-0.6	-2.219001	-0.7707898	-0.073540232	-0.027025561	0.5586403
## MA1_stats	-0.5	-2.956448	-0.7615168	0.202943008	0.044074542	0.8134004
## MA1_stats	-0.4	-3.299380	-0.6885207	-0.015867014	0.015052916	0.7564226
## MA1_stats	-0.3	-2.003957	-0.5314231	0.146578293	0.131302994	0.7055644
## MA1_stats	-0.2	-2.576290	-0.7096066	0.099984501	-0.011679411	0.6775915
## MA1_stats	-0.1	-2.481566	-0.6759008	-0.236799542	-0.168755851	0.4768884
## MA1_stats	0.0	-2.402710	-0.8855039	-0.268306656	-0.201570142	0.4703282
## MA1_stats	0.1	-2.363203	-0.4754717	0.151771071	0.098824380	0.8094831
## MA1_stats	0.2	-2.274544	-0.6830629	0.078845990	-0.028625444	0.7674243
## MA1_stats	0.3	-3.415573	-0.9011877	-0.258412367	-0.260718447	0.3876916
## MA1_stats	0.4	-3.381696	-0.4779428	0.080985451	0.163468653	0.7187999
## MA1_stats	0.5	-2.558856	-1.1446726	-0.146720887	-0.104500453	0.6942808
## MA1_stats	0.6	-3.771141	-0.8557957	-0.146433616	-0.056425331	0.6323674
## MA1_stats	0.7	-4.279108	-0.4368935	0.044567422	0.019592689	0.7057627
## MA1_stats	0.8	-3.745527	-0.9768702	-0.113807449	-0.125918865	0.7216140
## MA1_stats	0.9	-4.245927	-0.6990842	0.261715767	0.150833884	1.0473766
## MA1_stats	1.0	-3.493543	-1.1611192	-0.012198327	-0.123620590	0.8122150
##	Max.	StDev	SdDiff			
## MA1_stats	3.429784	1.5925282	2.743818			

```

## MA1_stats 3.721981 1.3723823 2.386705
## MA1_stats 2.291511 1.1963768 2.047869
## MA1_stats 3.252309 1.2505515 2.180980
## MA1_stats 1.944725 0.9574078 1.554551
## MA1_stats 3.799719 1.1727331 1.957892
## MA1_stats 2.811955 1.2130408 2.134068
## MA1_stats 2.757477 0.9479098 1.396446
## MA1_stats 2.532845 0.9858267 1.598275
## MA1_stats 1.949466 0.9234721 1.303728
## MA1_stats 2.457074 0.9453084 1.400122
## MA1_stats 2.067448 1.0355752 1.281394
## MA1_stats 2.500290 1.0225437 1.272520
## MA1_stats 1.625555 1.0126370 1.303601
## MA1_stats 3.910220 1.1069485 1.186168
## MA1_stats 2.407064 1.1852027 1.301631
## MA1_stats 3.471755 1.2484914 1.253210
## MA1_stats 2.893791 1.2448962 1.329048
## MA1_stats 2.472278 1.2125560 1.258386
## MA1_stats 3.761269 1.4816009 1.518935
## MA1_stats 2.505968 1.3471470 1.426041

```

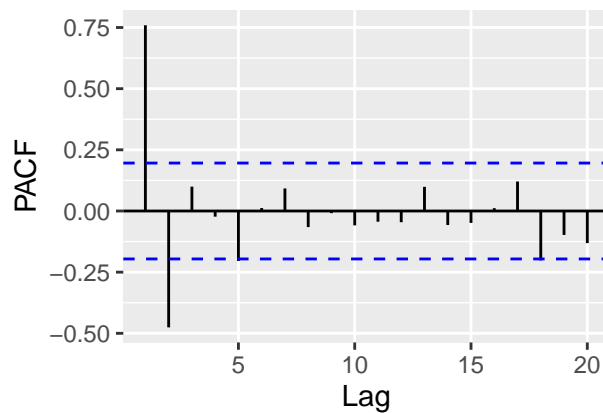
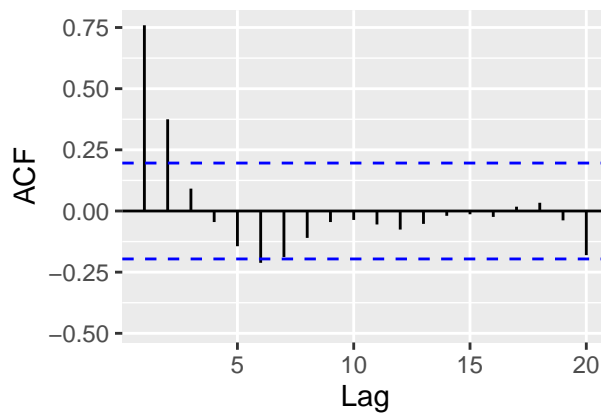
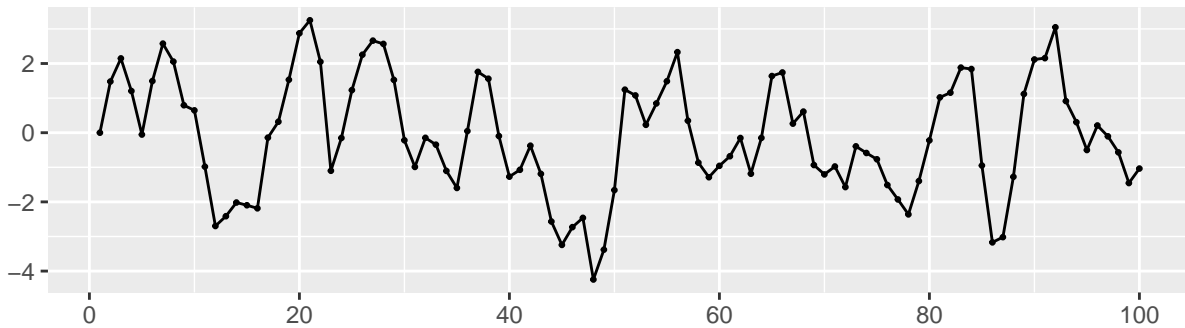
- The ACF for the MA(1) model shows zero correlation after the first lag (i.e., the subsequent lags generally fall within the critical bounds).
- The MA(1) model remains stationary regardless of the value of θ_1 .
- The change in the value of θ_1 determines the importance of the random shock from the previous period relative to that of the current period.

e) Generate data from an ARMA(1,1) model with

$\phi_1 = 0.6$, $\theta_1 = 0.6$, and $\sigma^2 = 1$.

```
ARMA_1_1 <- function(phi_1, theta_1)
{
  y <- ts(numeric(100))
  e <- rnorm(100)
  for(i in 2:100)
    y[i] <- phi_1*y[i-1] + theta_1*e[i-1] + e[i]
  return(y)
}
phi_1 <- 0.60
theta_1 <- 0.60
ARMA_1_1_result <- ARMA_1_1(phi_1,theta_1)
ggttsdisplay(ARMA_1_1_result,main=paste("ARMA(1,1) series, phi_1=", phi_1, ", theta_1=",theta_1))
```

ARMA(1,1) series, phi_1= 0.6 , theta_1= 0.6

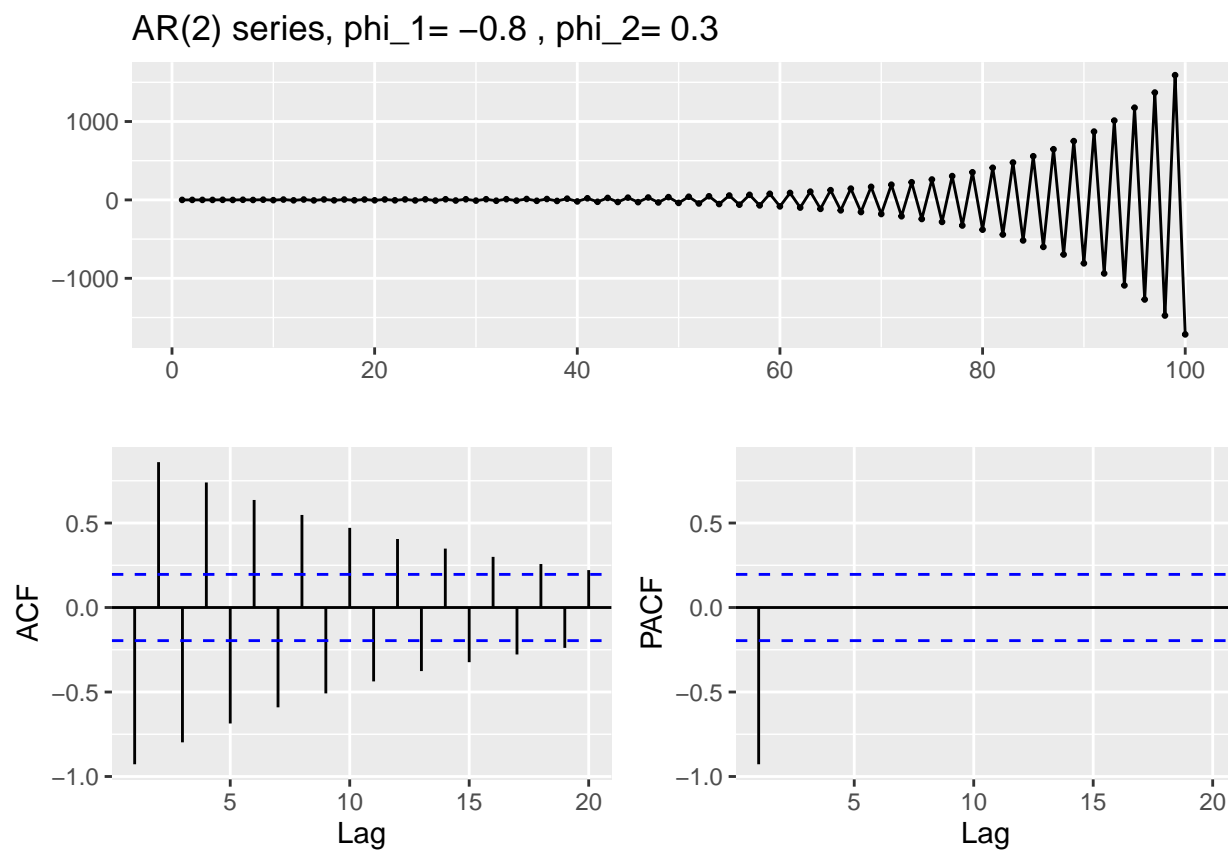


f) Generate data from an AR(2) model with $\phi_1 = -0.8$, $\phi_2 = 0.3$, and $\sigma^2 = 1$.

(Note that these parameters will give a **non-stationary** series.)

```
AR2 <- function(phi_1, phi_2)
{
  y <- ts(numeric(100))
  e <- rnorm(100)
  for(i in 3:100)
    y[i] <- phi_1*y[i-1] + phi_2*y[i-2] + e[i]
  return(y)
}

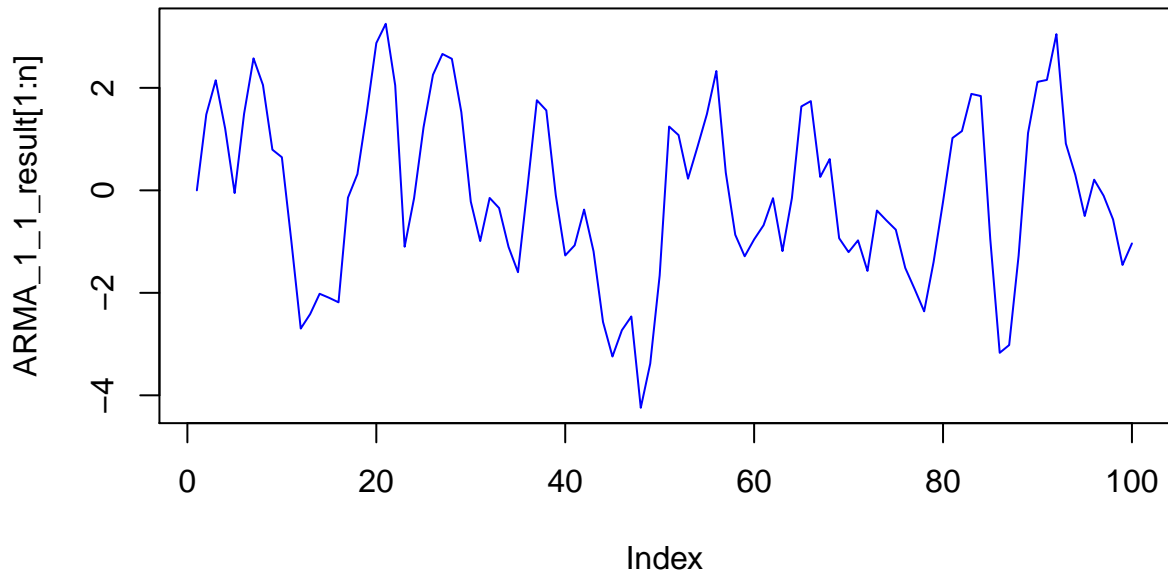
phi_1 <- -0.80
phi_2 <- 0.30
AR2_result <- AR2(phi_1, phi_2)
ggtsdisplay(AR2_result, main=paste("AR(2) series, phi_1=", phi_1, ", phi_2=", phi_2))
```



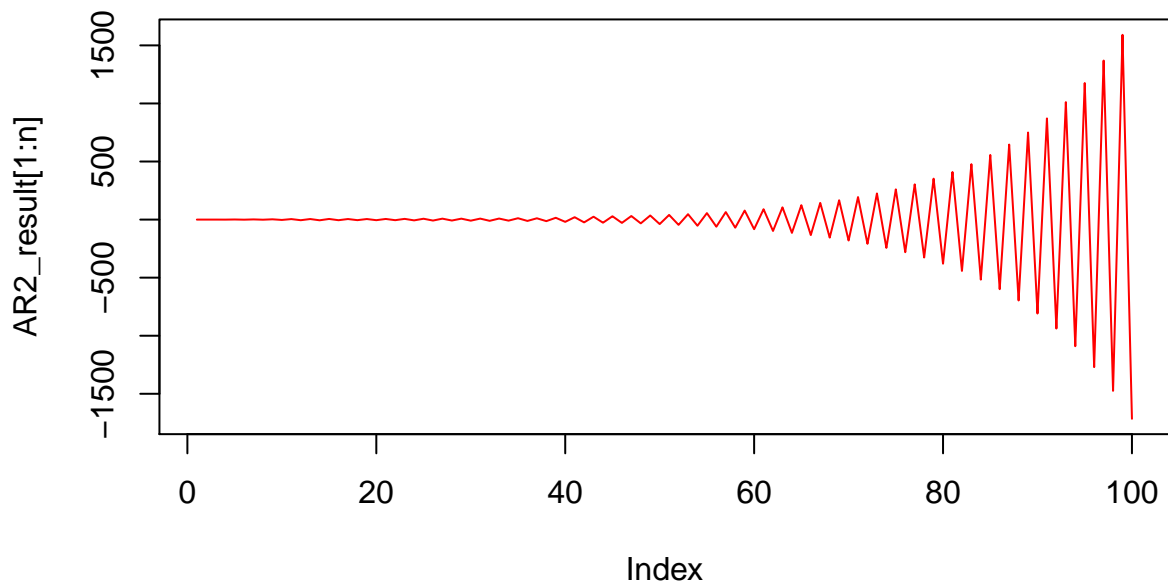
g) Graph the latter two series and compare them.

```
n=100
par(mfrow=c(2,1))
plot(ARMA_1_1_result[1:n], type="l", main="ARMA(1,1) - 100 observations",col="blue")
plot(AR2_result[1:n], type="l", main="AR(2) - 100 observations",col="red")
```

ARMA(1,1) – 100 observations



AR(2) – 100 observations



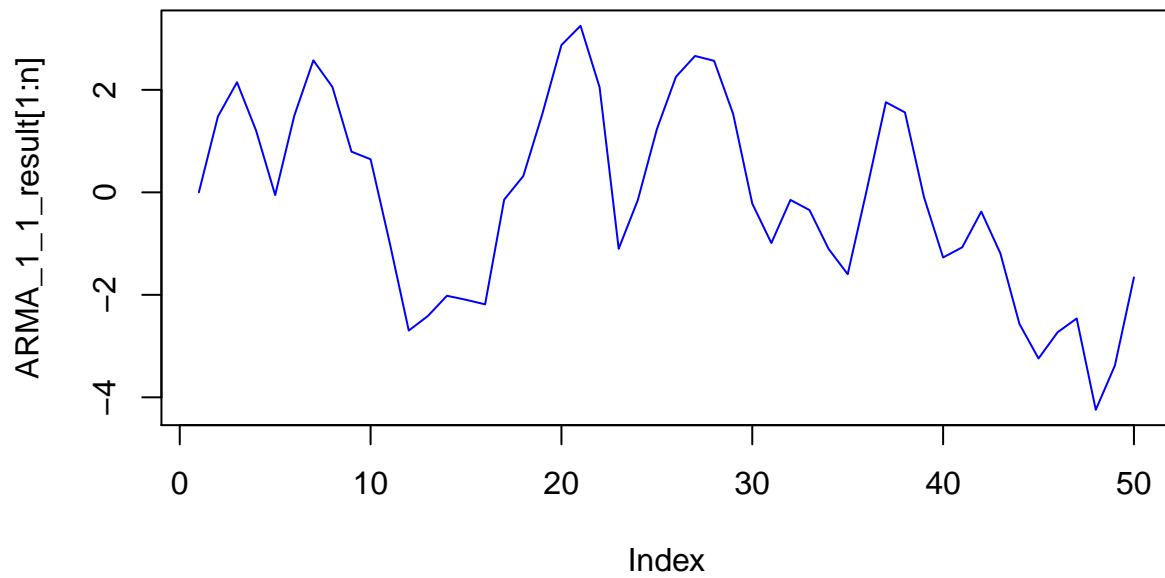
The ARMA(1,1) model reverts to the mean (here, zero) while not wandering more than 2 or 3 units away, while the AR(2) model explodes in an oscillation around zero which grows dramatically.

While it may appear that this series initially remains very close to zero, the difference in the Y-axes masks this behavior for the AR(2) series.

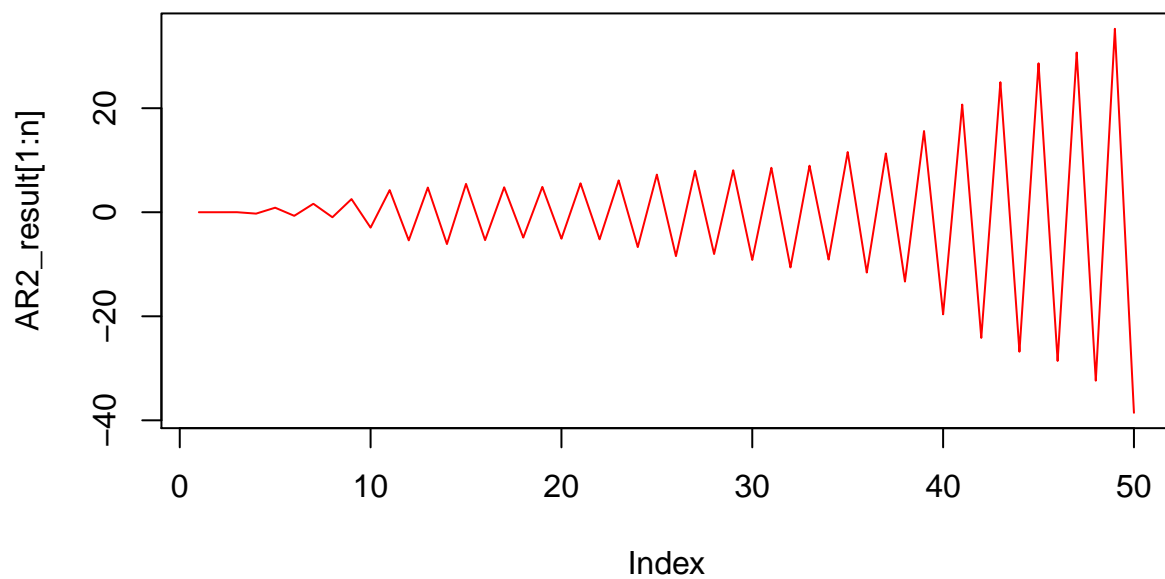
By plotting just the earlier portion of the two series, it is easier to see their behavior:

```
n=50
par(mfrow=c(2,1))
plot(ARMA_1_1_result[1:n], type="l", main="ARMA(1,1) - 50 observations",col="blue")
plot(AR2_result[1:n], type="l", main="AR(2) - 50 observations",col="red")
```

ARMA(1,1) – 50 observations

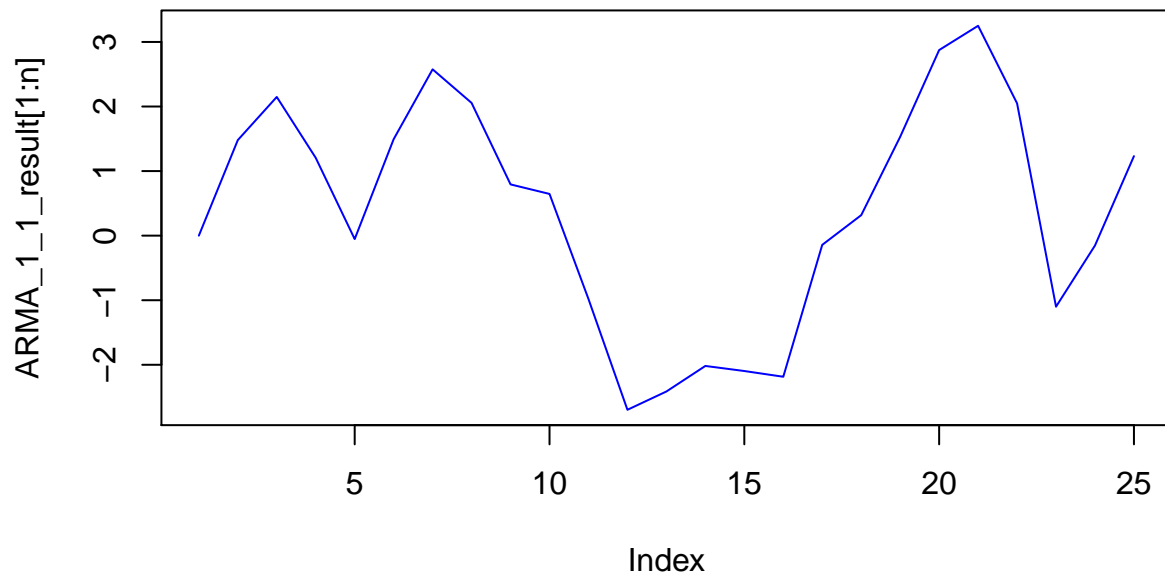


AR(2) – 50 observations

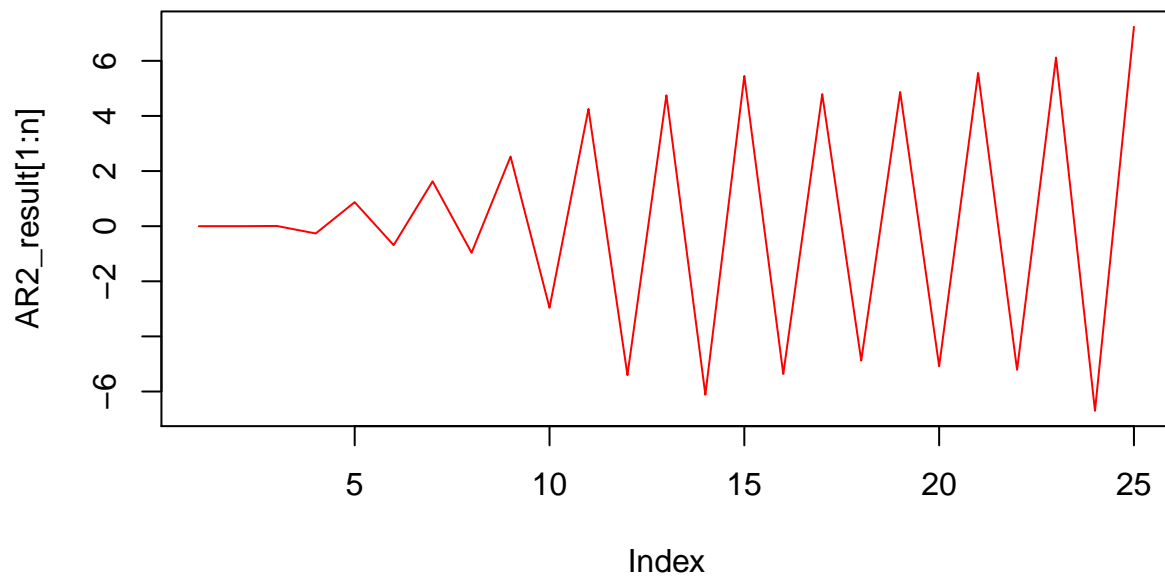


```
n=25
par(mfrow=c(2,1))
plot(ARMA_1_1_result[1:n], type="l", main="ARMA(1,1) - 25 observations",col="blue")
plot(AR2_result[1:n], type="l", main="AR(2) - 25 observations",col="red")
```

ARMA(1,1) – 25 observations



AR(2) – 25 observations

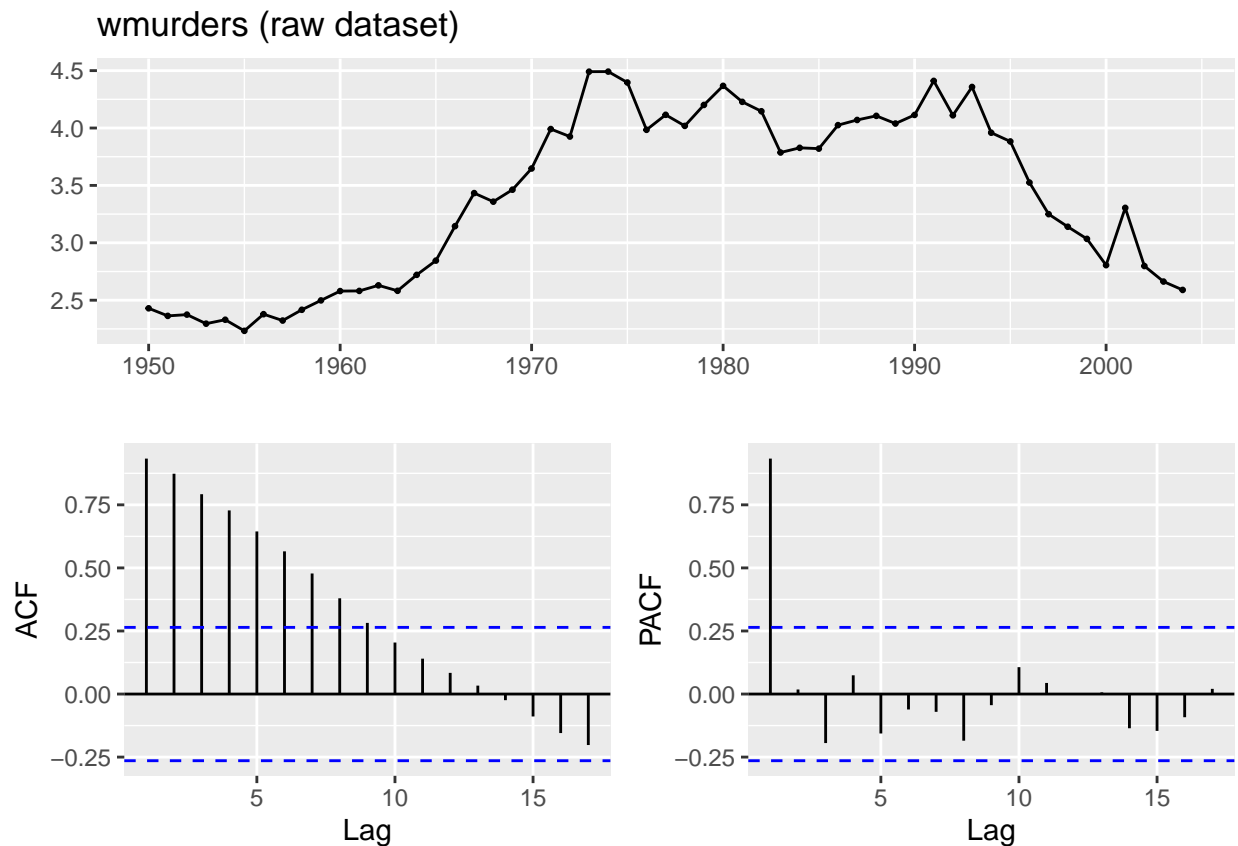


It's around the 25th observation when the AR(2) series begins its explosive oscillations.

8.7 Consider `wmurders`, the number of women murdered each year (per 100,000 standard population) in the United States.

a) By studying appropriate graphs of the series in R, find an appropriate $ARIMA(p,d,q)$ model for these data.

```
wmurders %>% ggtsdisplay(main="wmurders (raw dataset)")
```



The raw data series doesn't exhibit any discernable pattern (e.g., seasonality, which it won't have as the data is annual.)

There is a pattern in the ACF function, and a spike in the PACF function.

Check KPSS:

```
library(tseries)

#### Test raw data series
kpss.test(wmurders, "Level", lshort = F)
```

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test:

```
## Warning in kpss.test(wmurders, "Level", lshort = F): p-value greater than  
## printed p-value
```

```
##  
## KPSS Test for Level Stationarity  
##  
## data: wmurders  
## KPSS Level = 0.28623, Truncation lag parameter = 10, p-value = 0.1
```

```
kpss.test(wmurders, "Trend", lshort = F)
```

```
##  
## KPSS Test for Trend Stationarity  
##  
## data: wmurders  
## KPSS Trend = 0.1591, Truncation lag parameter = 10, p-value = 0.03909
```

```
wmurders %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: mu with 10 lags.  
##  
## Value of test-statistic is: 0.2862  
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.347 0.463  0.574 0.739
```

```
wmurders %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##  
## #####  
## # KPSS Unit Root Test #  
## #####  
##  
## Test is of type: tau with 10 lags.  
##  
## Value of test-statistic is: 0.1591  
##  
## Critical value for a significance level of:  
##          10pct  5pct 2.5pct  1pct  
## critical values 0.119 0.146  0.176 0.216
```

Level passes, but Trend fails.

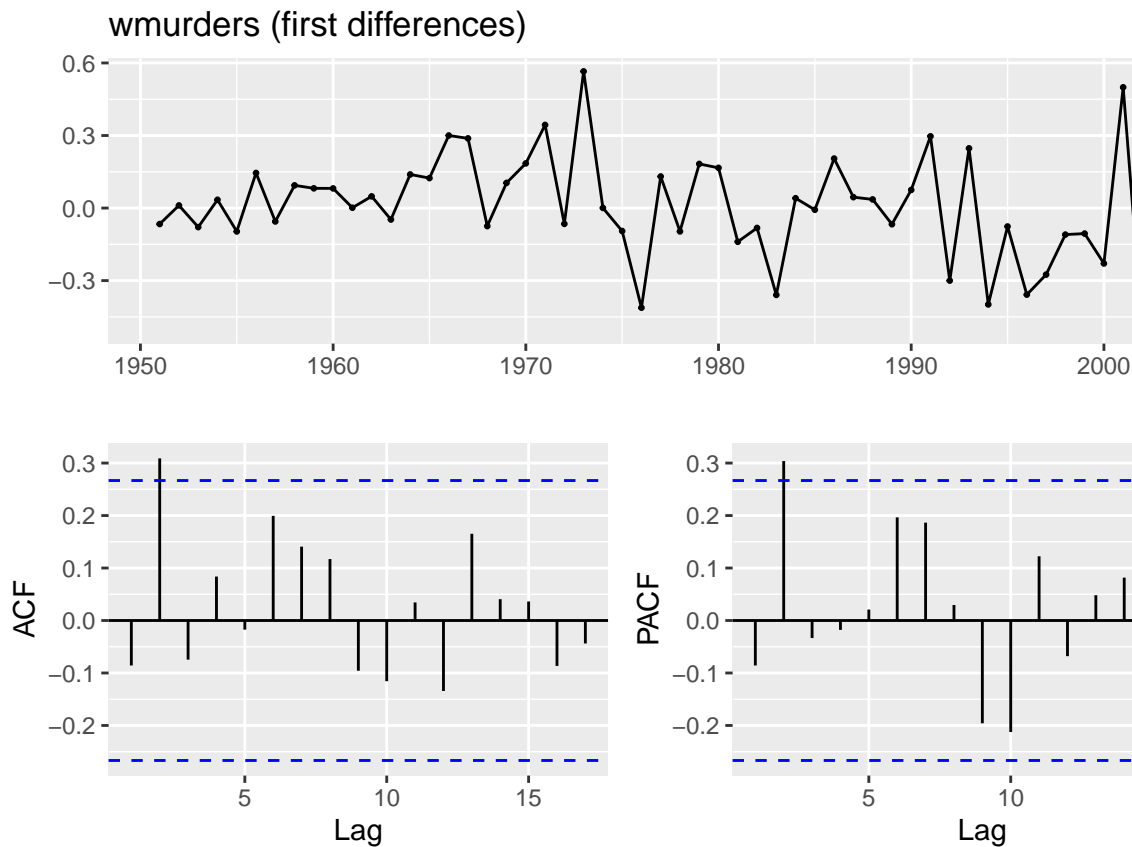
```
ndiffs(wmurders)
```

Check ndiffs:

```
## [1] 2
```

ndiffs indicates that we will need 2 differences.

```
wmurders_d1 <- wmurders %>% diff()  
wmurders_d1 %>% ggtsdisplay(main="wmurders (first differences)")
```



check first differences

There are still spikes at the second lag.

```
#### Test first differences  
kpss.test(wmurders_d1, "Level", lshort = F)
```

Check KPSS:

```
## Warning in kpss.test(wmurders_d1, "Level", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data:  wmurders_d1
## KPSS Level = 0.33134, Truncation lag parameter = 10, p-value = 0.1
```

```
kpss.test(wmurders_d1, "Trend", lshort = F)
```

```
## Warning in kpss.test(wmurders_d1, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data:  wmurders_d1
## KPSS Trend = 0.11631, Truncation lag parameter = 10, p-value = 0.1
```

```
wmurders_d1 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

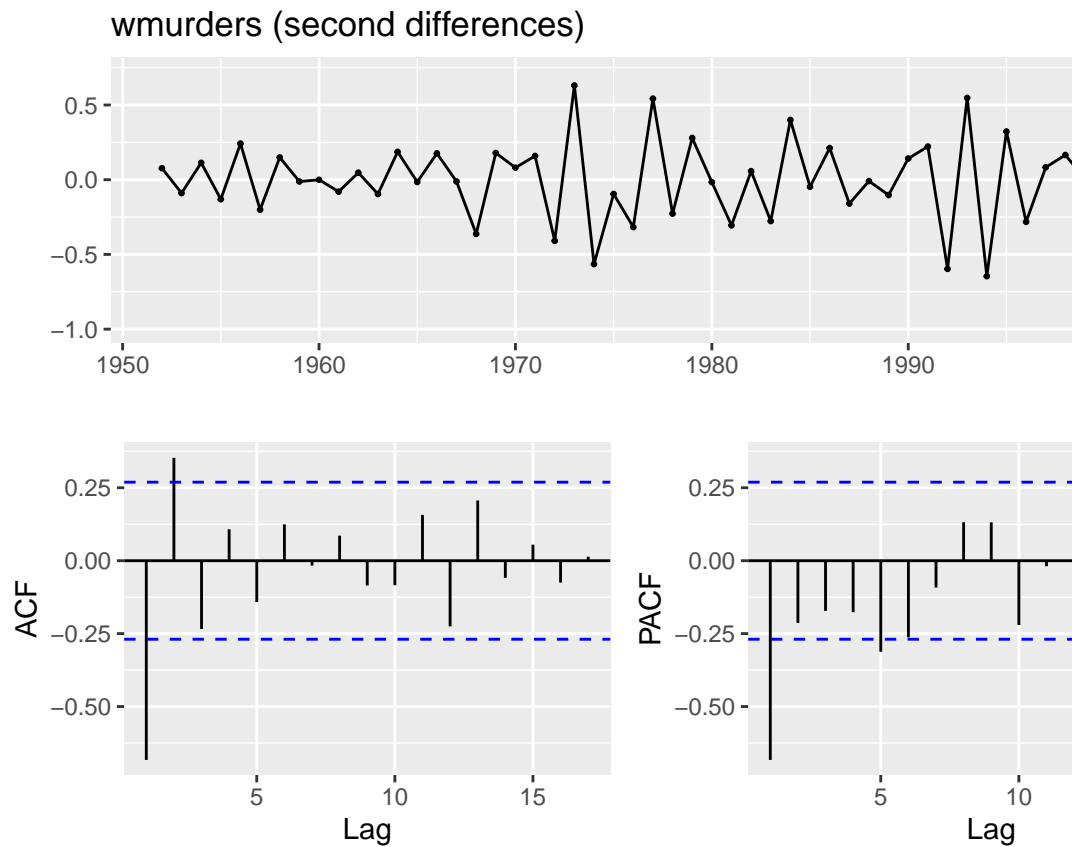
```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 10 lags.
##
## Value of test-statistic is: 0.3313
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
wmurders_d1 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 10 lags.
##
## Value of test-statistic is: 0.1163
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```

The tests pass, but why did ndiffs return 2?

```
wmurders_d2 <- wmurders_d1 %>% diff()
wmurders_d2 %>% ggtsdisplay(main="wmurders (second differences)")
```



Look at second differences

This looks even worse...

```
#### Test second differences
kpss.test(wmurders_d2, "Level", lshort = F)
```

Check KPSS:

```
## Warning in kpss.test(wmurders_d2, "Level", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Level Stationarity
##
## data: wmurders_d2
## KPSS Level = 0.12299, Truncation lag parameter = 10, p-value = 0.1
```



```
kpss.test(wmurders_d2, "Trend", lshort = F)
```

```
## Warning in kpss.test(wmurders_d2, "Trend", lshort = F): p-value greater than
## printed p-value
```

```
##
## KPSS Test for Trend Stationarity
##
## data:  wmurders_d2
## KPSS Trend = 0.10938, Truncation lag parameter = 10, p-value = 0.1
```

```
wmurders_d2 %>% ur.kpss(type="mu", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: mu with 10 lags.
##
## Value of test-statistic is: 0.123
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463 0.574 0.739
```

```
wmurders_d2 %>% ur.kpss(type="tau", lags="long") %>% summary()
```

```
##
## #####
## # KPSS Unit Root Test #
## #####
##
## Test is of type: tau with 10 lags.
##
## Value of test-statistic is: 0.1094
##
## Critical value for a significance level of:
##          10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146 0.176 0.216
```

This passes, too, but it's unclear why we need the second difference.

Let's assume the model is ARIMA(p,2,q).

Because of the spikes in the ACF and the PACF at lag=1, the model may be ARIMA(1,2,0) or ARIMA(0,2,1).

Alternatively, it might be ARIMA(1,2,1), but the text indicates that you can't determine both (p,q) nonzero from looking at the graphs. However, auto.arima indicates that it should be ARIMA(1,2,1), so let's use that.

b) Should you include a constant in the model? Explain.

No, because a constant would cause a drift, and there is no drift visible in the data.

c) Write this model in terms of the backshift operator.

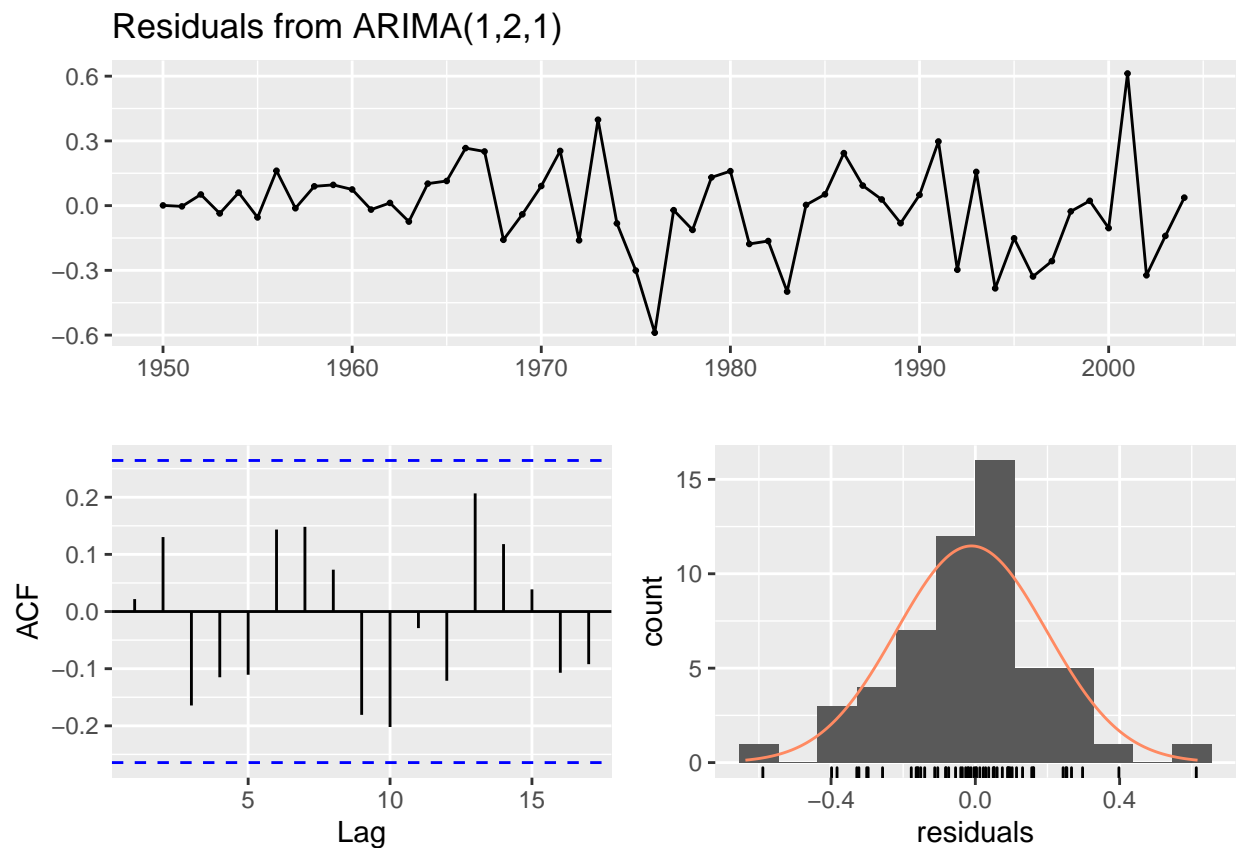
In general, ARIMA(1,2,1): $(1 - \phi_1 B)(1 - B)^2 y_t = c + (1 + \theta_1 B)\varepsilon_t$, but we have decided to omit the constant, so $(1 - \phi_1 B)(1 - B)^2 y_t = (1 + \theta_1 B)\varepsilon_t$

d) Fit the model using R and examine the residuals. Is the model satisfactory?

```
(wmurders_fit <- Arima(wmurders, c(1,2,1)))
```

```
## Series: wmurders
## ARIMA(1,2,1)
##
## Coefficients:
##          ar1          ma1
##       -0.2434   -0.8261
## s.e.    0.1553    0.1143
##
## sigma^2 estimated as 0.04632:  log likelihood=6.44
## AIC=-6.88   AICc=-6.39   BIC=-0.97
```

```
checkresiduals(wmurders_fit)
```



```
##
```

```
## Ljung-Box test
##
## data: Residuals from ARIMA(1,2,1)
## Q* = 12.419, df = 8, p-value = 0.1335
##
## Model df: 2. Total lags used: 10
```

Because the p-value is high, we FAIL TO REJECT the null hypothesis, which is that the data are independent (i.e., no serial correlation.)

Additionally, the ACF bars all fall within the critical bands.

Therefore, the model is satisfactory.

e) Forecast three times ahead. Check your forecasts by hand to make sure that you know how they have been calculated.

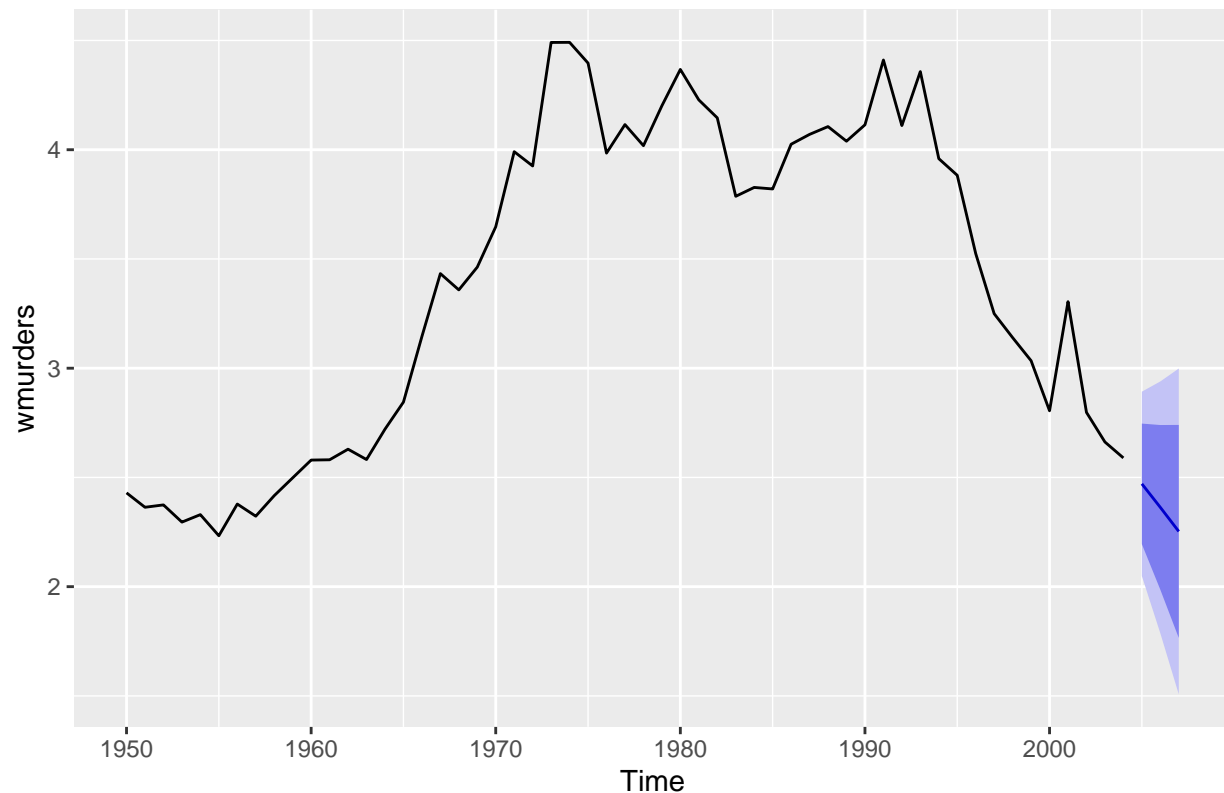
```
wmurders_forecast <- forecast(wmurders_fit, h = 3)
wmurders_forecast
```

```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2005      2.470660 2.194836 2.746484 2.048824 2.892496
## 2006      2.363106 1.986351 2.739862 1.786908 2.939304
## 2007      2.252833 1.765391 2.740276 1.507354 2.998313
```

f) Create a plot of the series with forecasts and prediction intervals for the next three periods shown.

```
autoplot(wmurders_forecast)
```

Forecasts from ARIMA(1,2,1)



g) Does `auto.arima()` give the same model you have chosen? If not, which model do you think is better?

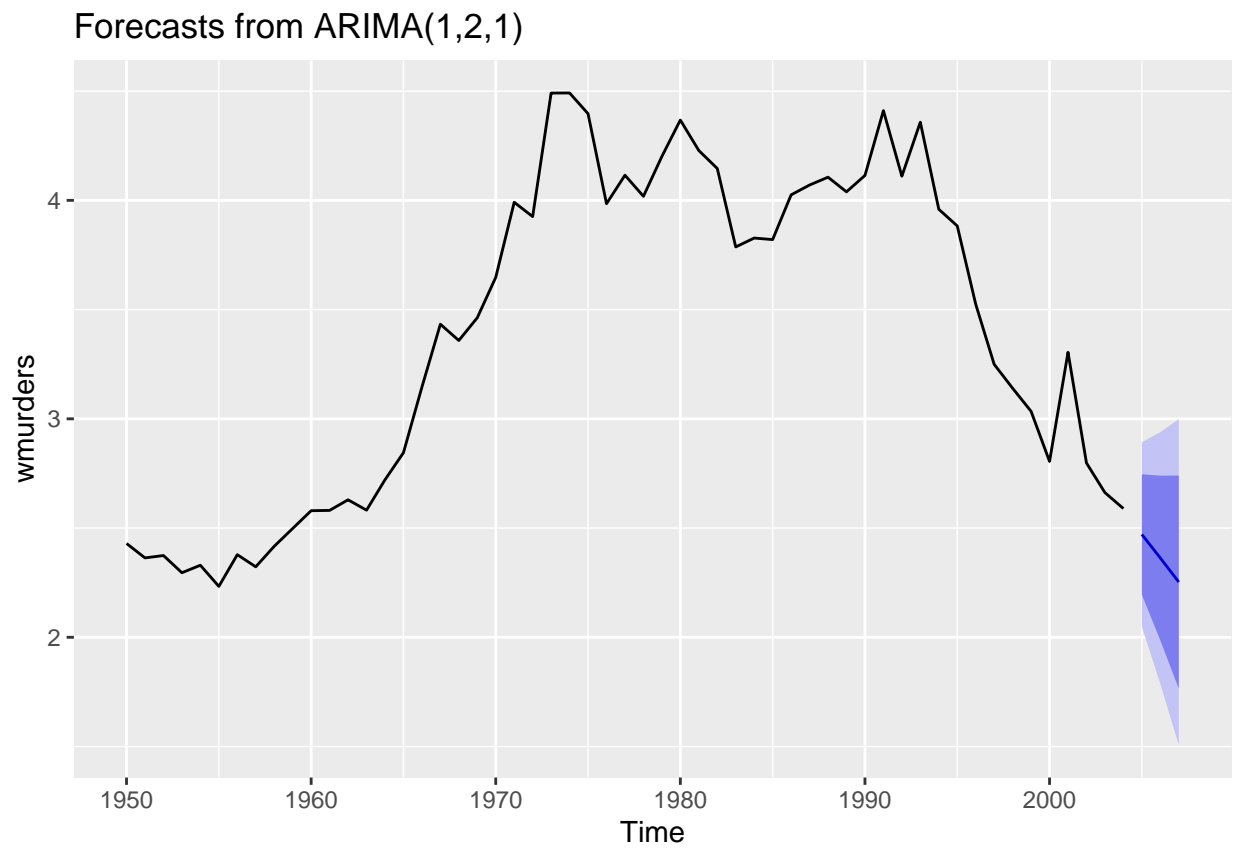
```
wmurders_fit_autoarima <- auto.arima(wmurders)
wmurders_fit_autoarima
```

```
## Series: wmurders
## ARIMA(1,2,1)
##
## Coefficients:
##      ar1      ma1
##    -0.2434 -0.8261
## s.e.   0.1553   0.1143
##
## sigma^2 estimated as 0.04632: log likelihood=6.44
## AIC=-6.88 AICc=-6.39 BIC=-0.97
```

```
wmurders_forecast_autoarima <- forecast(wmurders_fit_autoarima, h = 3)
wmurders_forecast_autoarima
```

```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2005      2.470660 2.194836 2.746484 2.048824 2.892496
## 2006      2.363106 1.986351 2.739862 1.786908 2.939304
## 2007      2.252833 1.765391 2.740276 1.507354 2.998313
```

```
autoplot(wmurders_forecast_autoarima)
```



The model selected is the same.