Data624 - Homework6

Amit Kapoor

3/16/2021

Contents

Exercise 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?

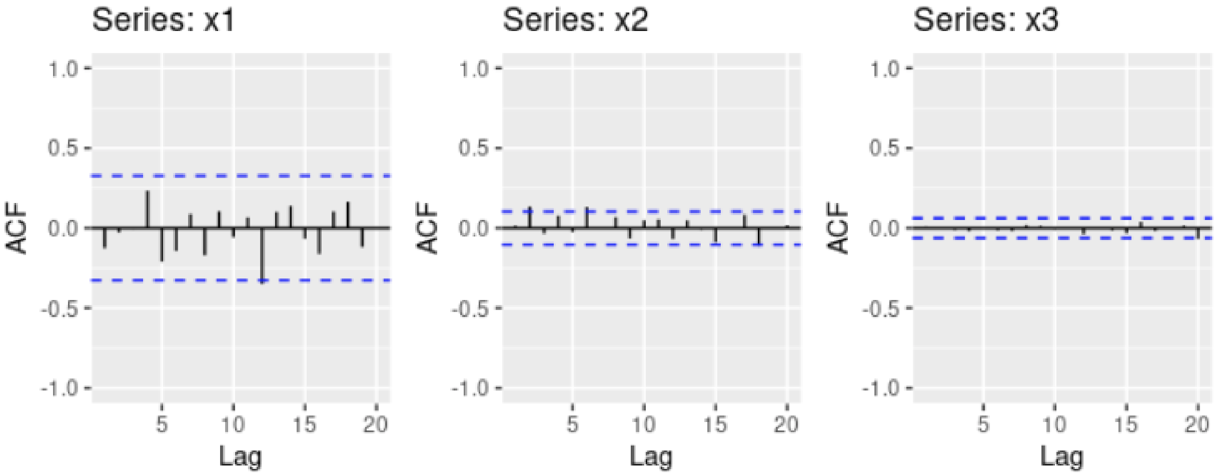


Figure 8.31: Left: ACF for a white noise series of 36 numbers. Middle: ACF for a white noise series of 360 numbers. Right: ACF for a white noise series of 1,000 numbers.

As sample size increases (from 36 random number to 360 random numbers and then 1,000 random numbers), the correlation tends to 0.The ACF bands keeps on getting narrower and random numbers size increases. If the number of spikes are more than 5% that are outside the bounds than series is not white noise. In each of these ACF plots, that is not the case as all the bars are close to 0 so all are of white noise.

b

Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?

The critical values are at different distances from the mean of zero since critical values for white noise are supposed to lie within $\pm\frac{1.96}{\sqrt{T}}$ where T is length of time series. In this case, as T gets bigger, range gets smaller. Hence the autocorrelations different in each figure.

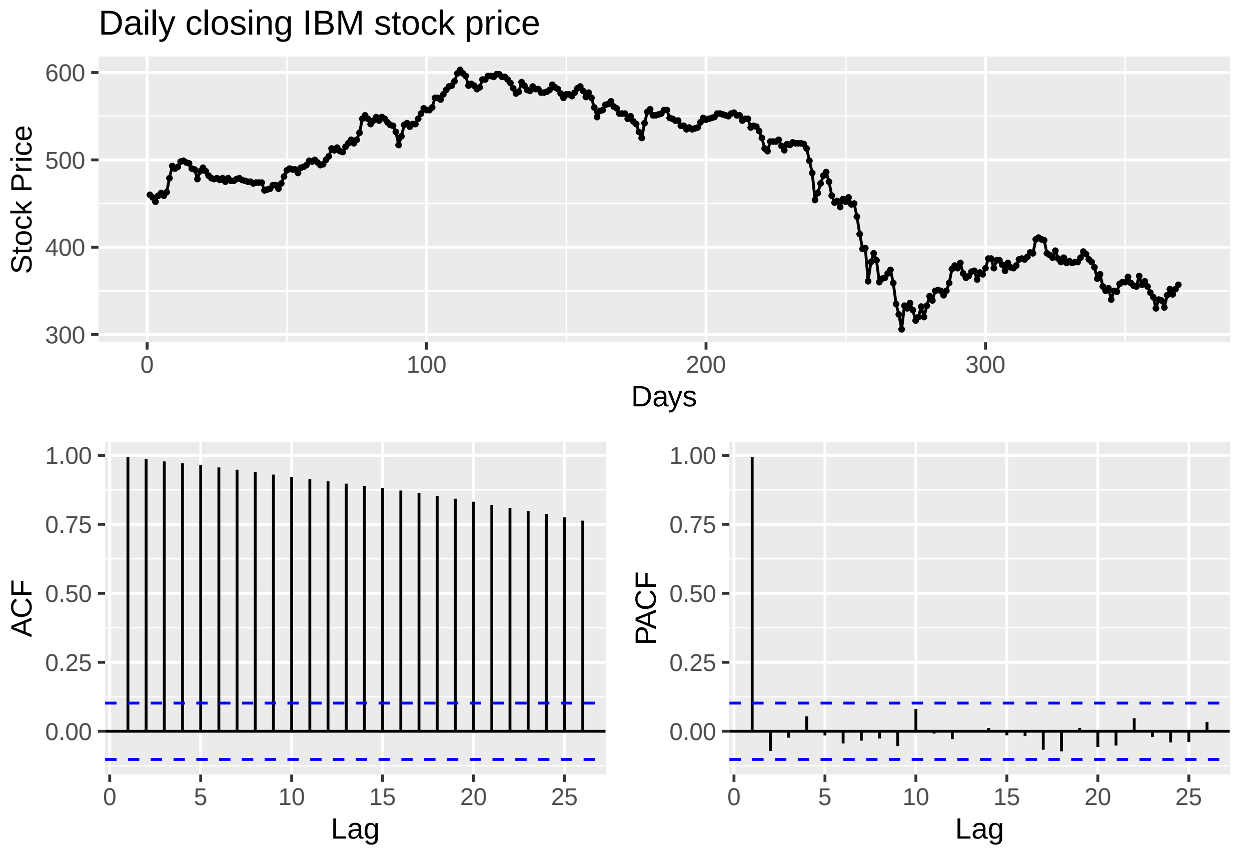
Exercise 8.2

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

```
head(ibmclose)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1] 460 457 452 459 462 459
```

```
ggtsdisplay(ibmclose, main="Daily closing IBM stock price", ylab="Stock Price", xlab="Days")
```



This time series does show the trend e.g. from 210 to 270 it shows the downward trend. the ACF plot is useful for identifying non-stationary time series. For a non stationary time series, the ACF plot decreases slowly. Here the ACF plot does show a slow decrease as the lag increases.The PACF plot is a plot of the partial correlation coefficients between the series and lags of itself. Here PACF plot shows the first lag is close to 1 and all the other PACF is close to 0. Thus we can conclude it is a non-stationary time series and should be differenced to make it stationary.

Exercise 8.3

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

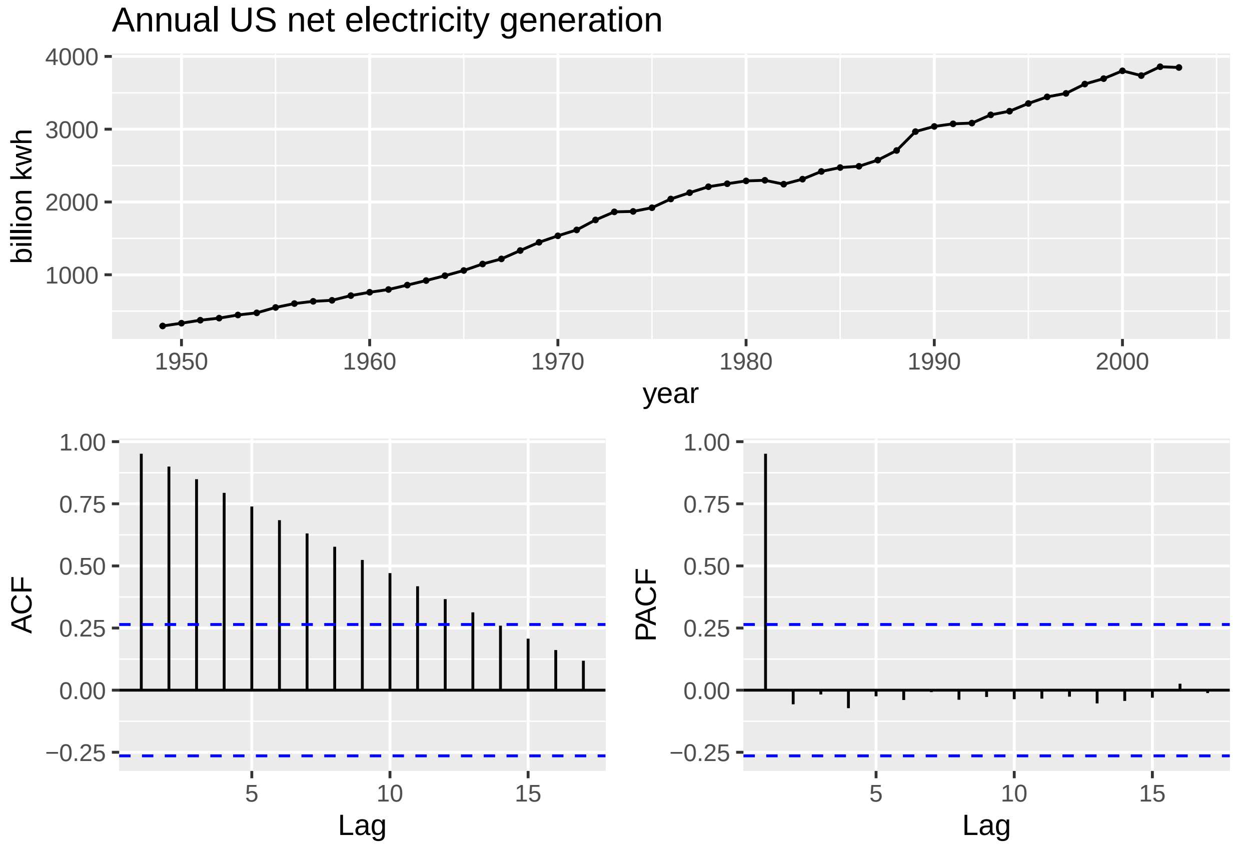
a

usnetelec

```
#Annual US net electricity generation (billion kwh) for 1949-2003
head(usnetelec)
```

```
## Time Series:
## Start = 1949
## End = 1954
## Frequency = 1
## [1] 296.1 334.1 375.3 403.8 447.0 476.3
```

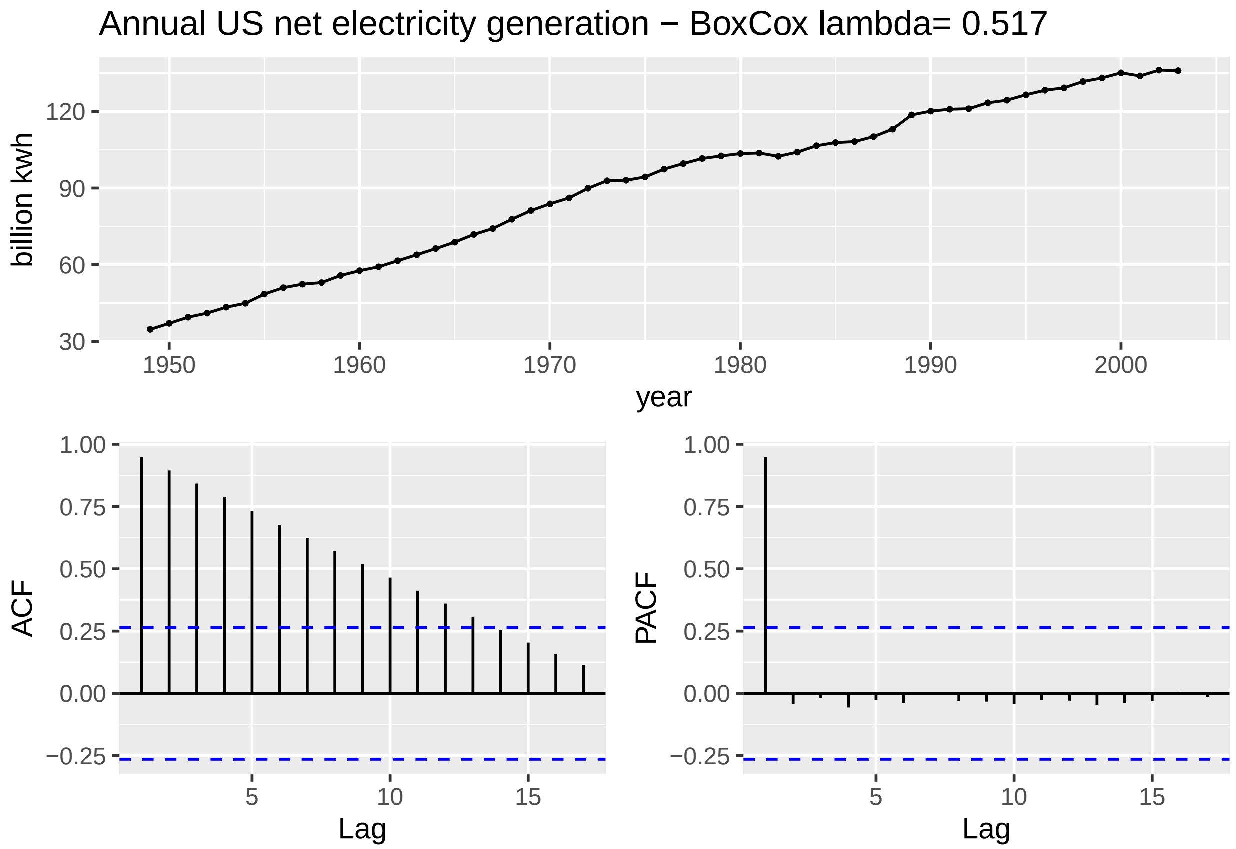
```
ggtsdisplay(usnetelec,
            main="Annual US net electricity generation",
            ylab="billion kwh",
            xlab="year")
```



The graph and ACF plot show an upward trend for this time series. PACF shows all the lags close to 0 except the 1st one which is close to 1. It confirms this a non stationary time series.

Now lets do the BoxCox transformation and see the results.

```
bc_trans <- BoxCox(usnetelec, BoxCox.lambda(usnetelec))
ggtsdisplay(bc_trans,
            main=paste("Annual US net electricity generation - BoxCox lambda=", round(BoxCox.lambda(usne
            ylab="billion kwh",
            xlab="year")
```



After BoxCox transformation of given time series, we dont see noticeable change here which could be due to non seasonalilty in time series. Next is to use kpss test in which the null hypothesis is that the data are stationary, and we look for evidence that the null hypothesis is false. Consequently, small p-values (e.g., less than 0.05) suggest that differencing is required.

```
bc_trans %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 1.4583
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected and the boxcox transformed data is not stationary. We will now use ndiffs() function to determine the order of differencing.

```
ndiffs(bc_trans)
```

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

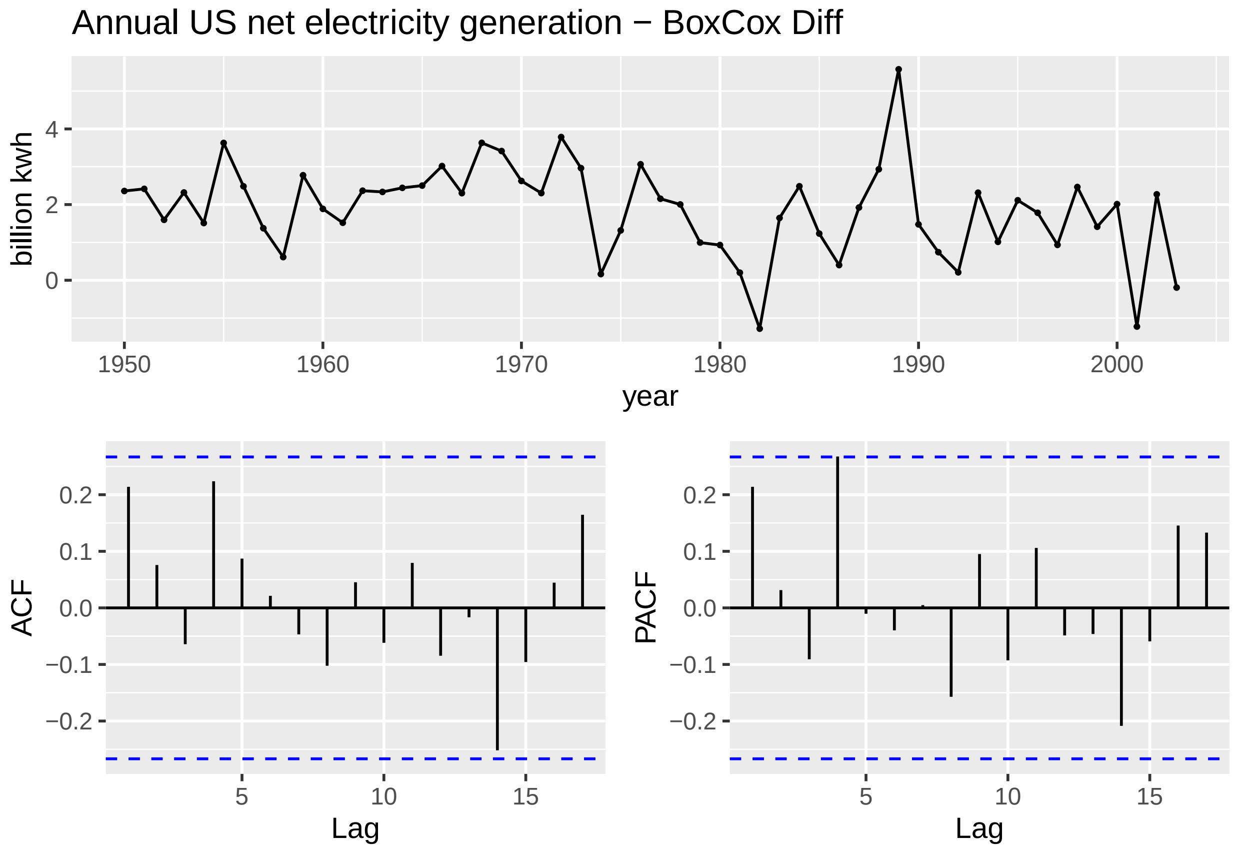
It shows number of differences required is 2 for boxcox transformed data. Lets first apply the differences as 1 and see the results.

```r
bct.diff <- bc_trans %>% diff()
bct.diff %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.4315
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can see the diff of order 1 makes the test statistic small and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

```r
ggtsdisplay(bct.diff,
            main="Annual US net electricity generation - BoxCox Diff",
            ylab="billion kwh",
            xlab="year")
```



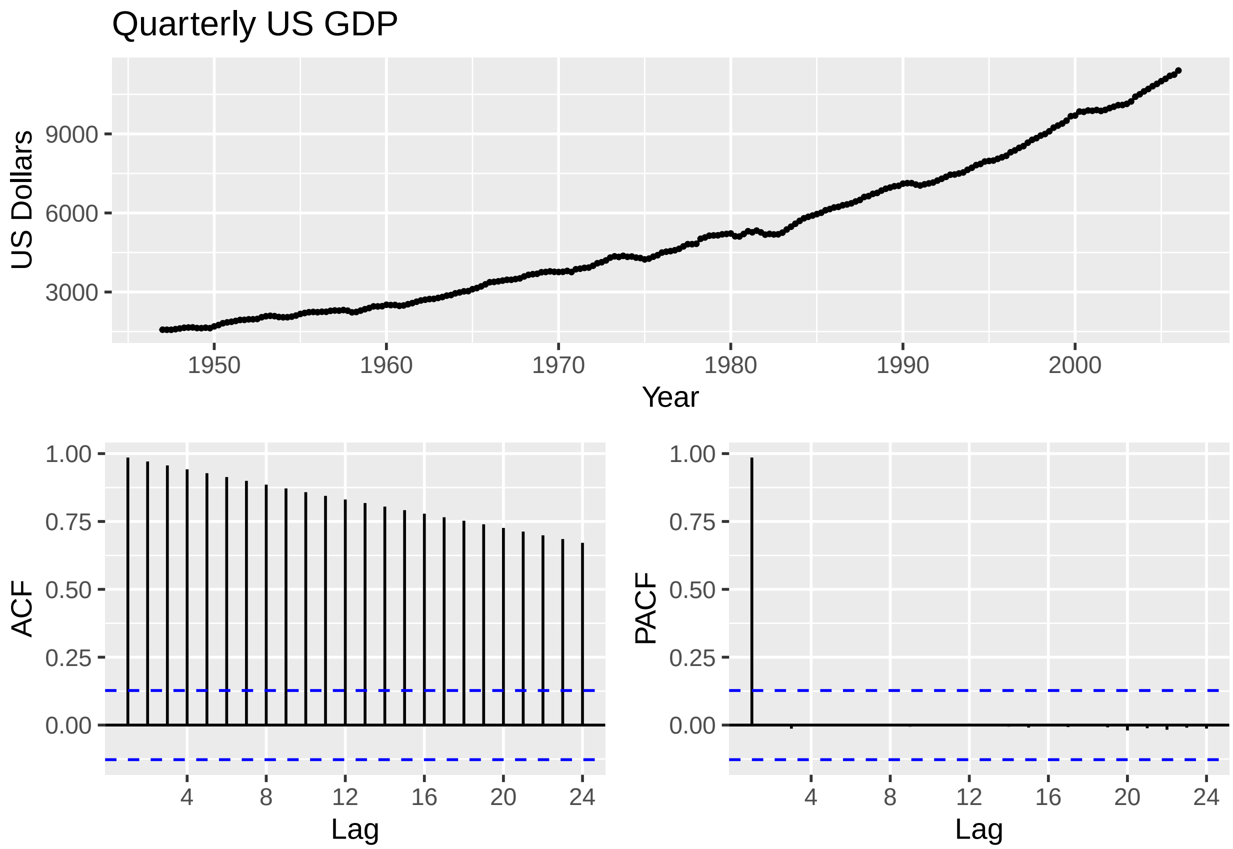
b

`usgdp`

```
# Quarterly US GDP. 1947:1 - 2006.1.
head(usgdp)
```

```
##         Qtr1   Qtr2   Qtr3   Qtr4
## 1947 1570.5 1568.7 1568.0 1590.9
## 1948 1616.1 1644.6
```

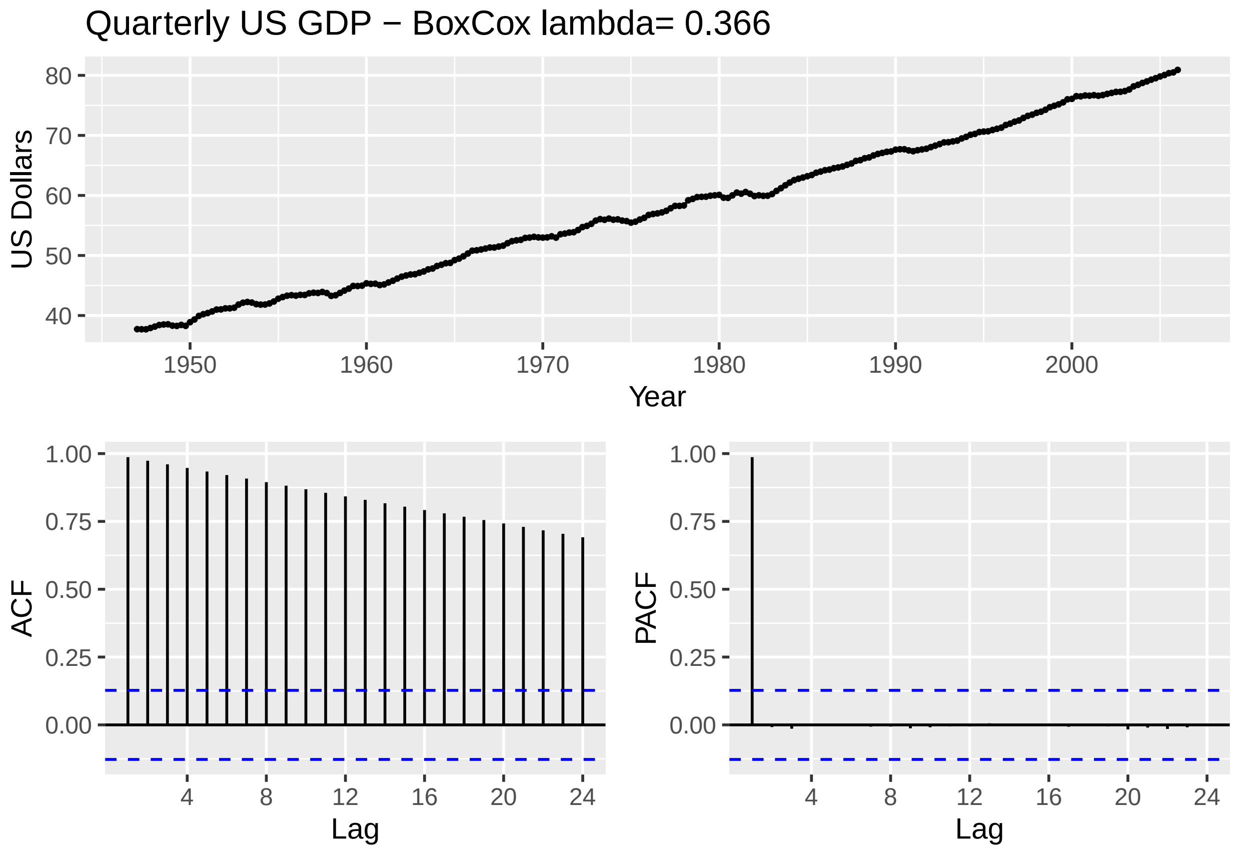
```
ggtsdisplay(usgdp, main="Quarterly US GDP",xlab="Year",ylab="US Dollars")
```



The graph and ACF plot show an upward trend for this time series. PACF shows all the lags close to 0 except the 1st one which is close to 1. It confirms this a non stationary time series.

Now lets do the BoxCox transformation and see the results.

```
usgdp.bc_trans <- BoxCox(usgdp, BoxCox.lambda(usgdp))
ggtsdisplay(usgdp.bc_trans,
            main=paste("Quarterly US GDP - BoxCox lambda=", round(BoxCox.lambda(usgdp), 3)),
            xlab="Year",
            ylab="US Dollars")
```



It is evident here that Box-Cox transformation, with lambda 0.366, has removed the curvature in the original data. Next is to use kpss test in which the null hypothesis is that the data are stationary,

```
usgdp.bc_trans %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 4.8114
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected and the boxcox transformed data is not stationary. We will now use ndiffs() function to determine the order of differencing.

```
ndiffs(usgdp.bc_trans)
```

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

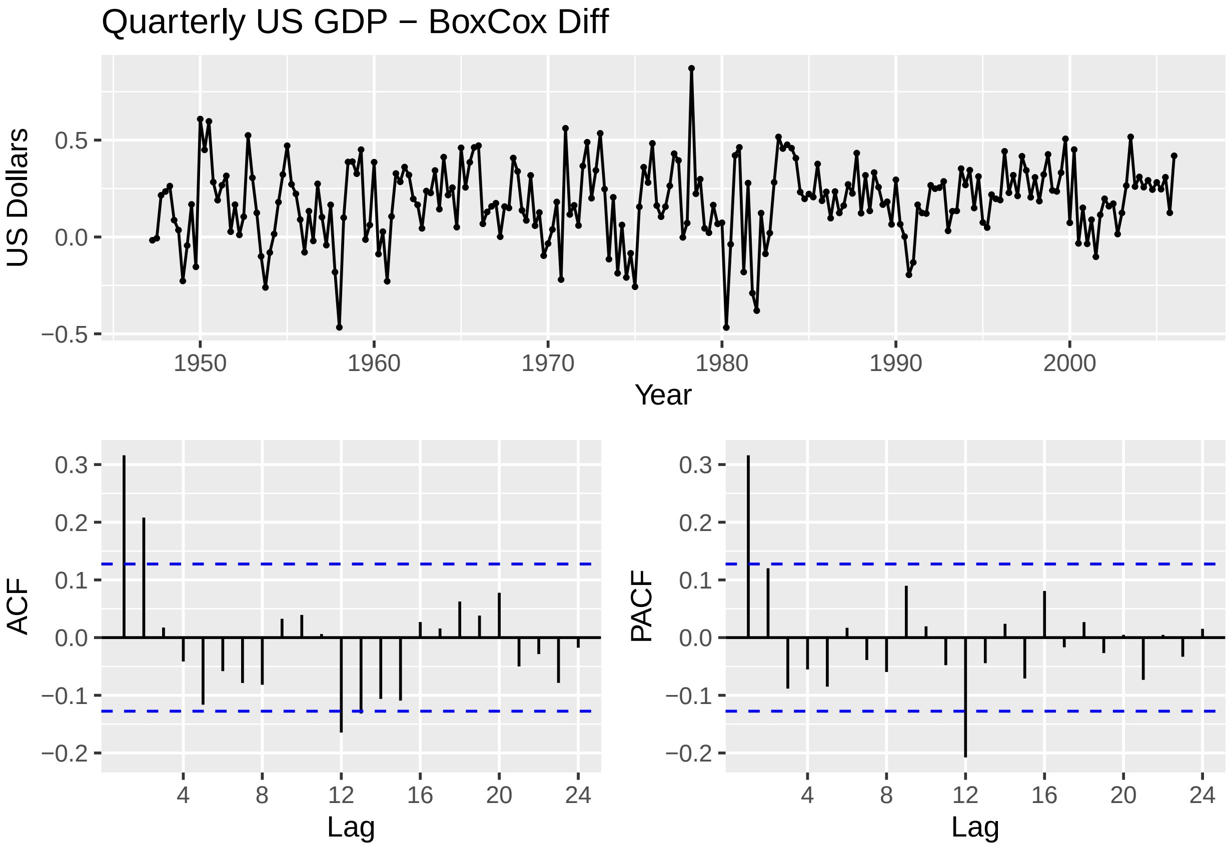
It shows number of differences required is 1 for boxcox transformed data.

```
usgdp.bct.diff <- usgdp.bc_trans %>% diff()
usgdp.bct.diff %>% ur.kpss() %>% summary()
```

```
## 
## ####################### 
## # KPSS Unit Root Test # 
## ####################### 
## 
## Test is of type: mu with 4 lags. 
## 
## Value of test-statistic is: 0.2013 
## 
## Critical value for a significance level of: 
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can see the diff of order 1 makes the test statistic small and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

```
ggtsdisplay(usgdp.bct.diff,
            main="Quarterly US GDP - BoxCox Diff",
            xlab="Year",
            ylab="US Dollars")
```



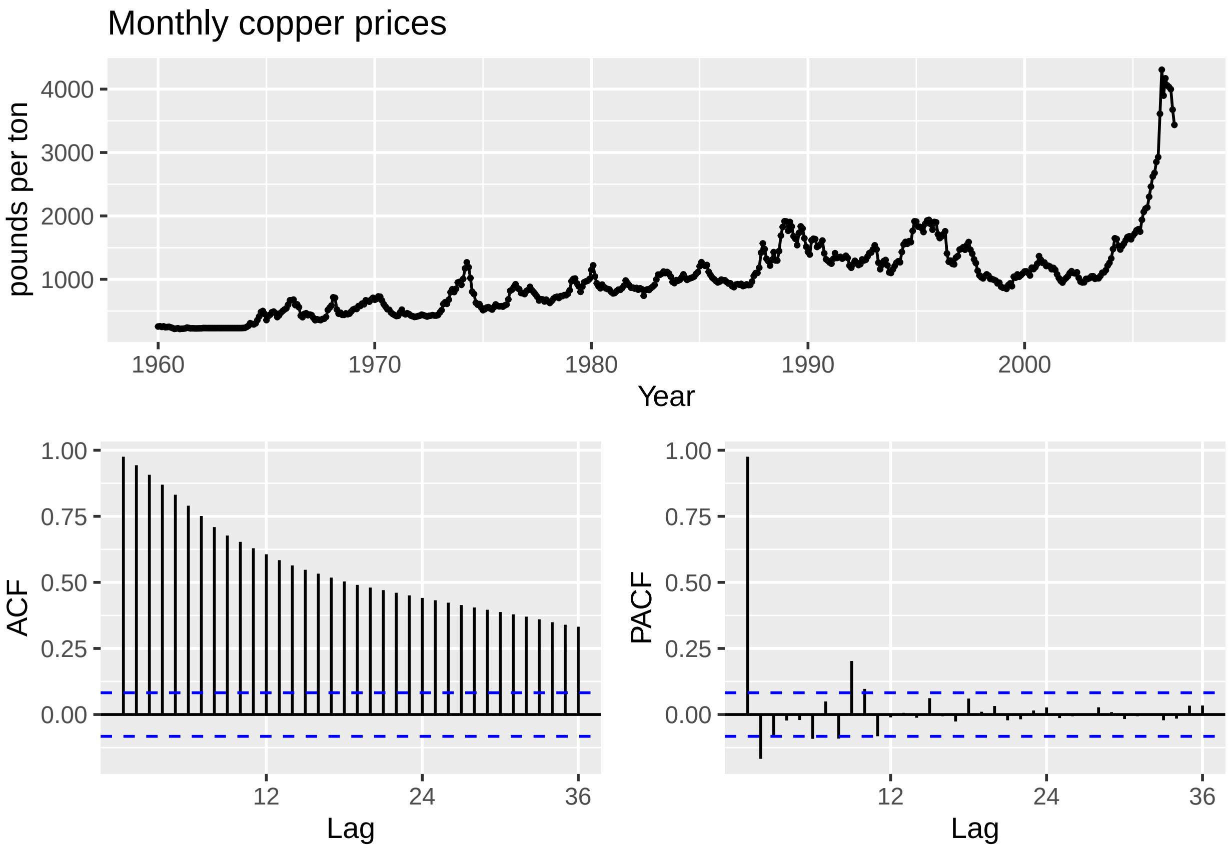
c

`mcopper`

```
head(mcopper)
```

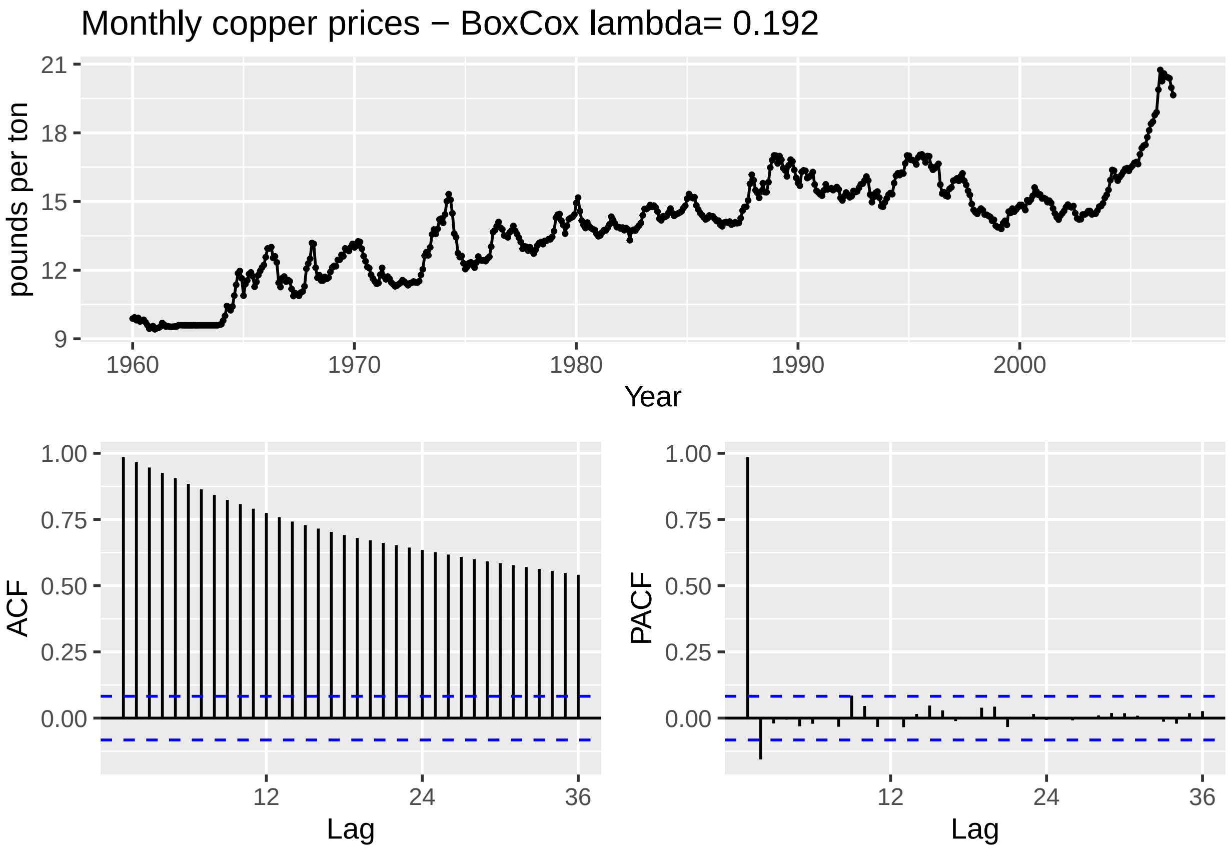
```
##        Jan   Feb   Mar   Apr   May   Jun
## 1960 255.2 259.7 249.3 258.0 244.3 246.8
```

```
# Monthly copper prices
ggtsdisplay(mcopper, main="Monthly copper prices", ylab="pounds per ton", xlab="Year")
```



The time series shows a slight seasonality and periods of both downward and upward trend. We can also see a sudden spike in 2000s. These facts confirm that it is a non-stationary time series. Now lets do the BoxCox transformation and see the results.

```
mcop.bc_trans <- BoxCox(mcopper, BoxCox.lambda(mcopper))
ggtsdisplay(mcop.bc_trans,
            main=paste("Monthly copper prices - BoxCox lambda=", round(BoxCox.lambda(mcopper), 3)),
            ylab="pounds per ton",
            xlab="Year")
```



It is evident here that Box-Cox transformation, with lambda 0.192, shows the seasonality. Next is to use kpss test in which the null hypothesis is that the data are stationary.

```
mcop.bc_trans %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 6.2659
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected and the boxcox transformed data is not stationary. We will now use ndiffs() function to determine the order of differencing.

```
ndiffs(mcop.bc_trans)
```

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

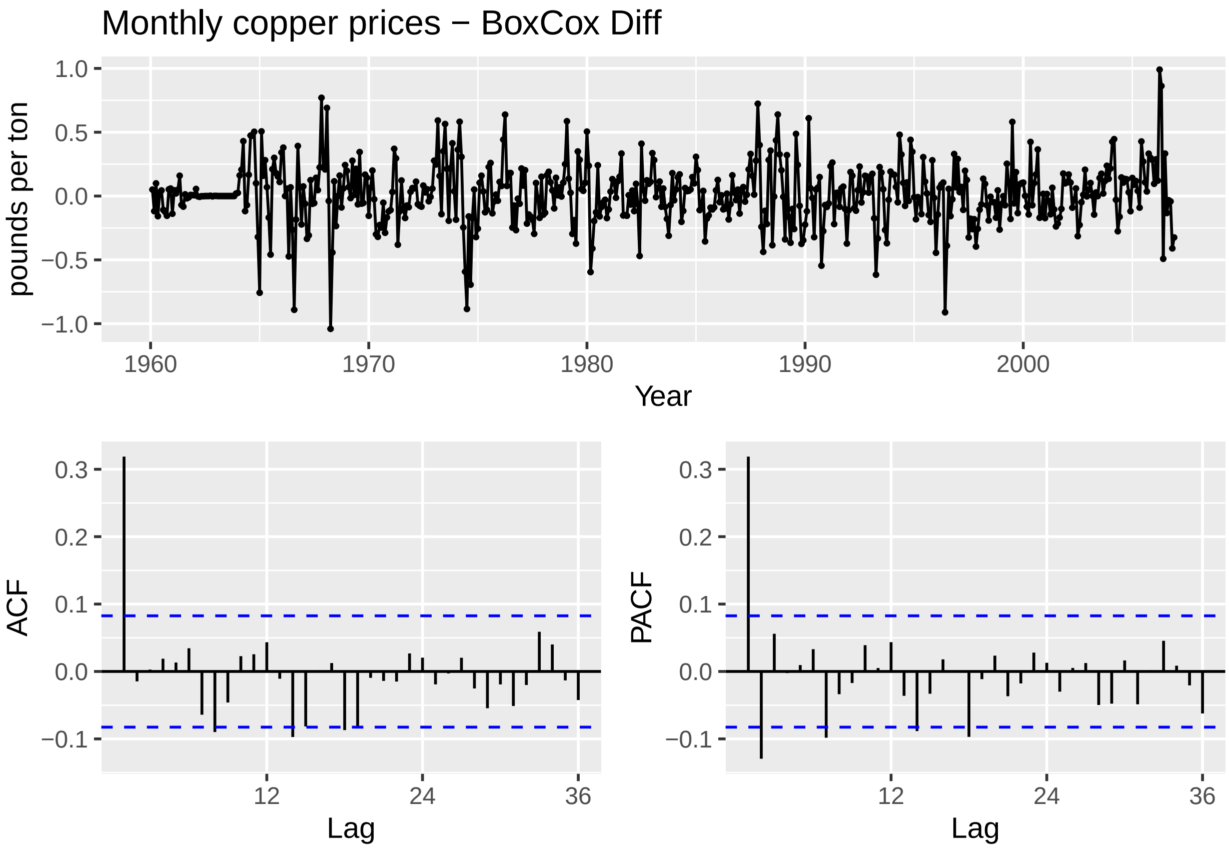
It shows number of differences required is 1 for boxcox transformed data.

```
mcop.bct.diff <- mcop.bc_trans %>% diff()
mcop.bct.diff %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 6 lags.
##
## Value of test-statistic is: 0.0573
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can see the diff of order 1 makes the test statistic small and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

```
ggtsdisplay(mcop.bct.diff,
            main="Monthly copper prices - BoxCox Diff",
            ylab="pounds per ton",
            xlab="Year")
```



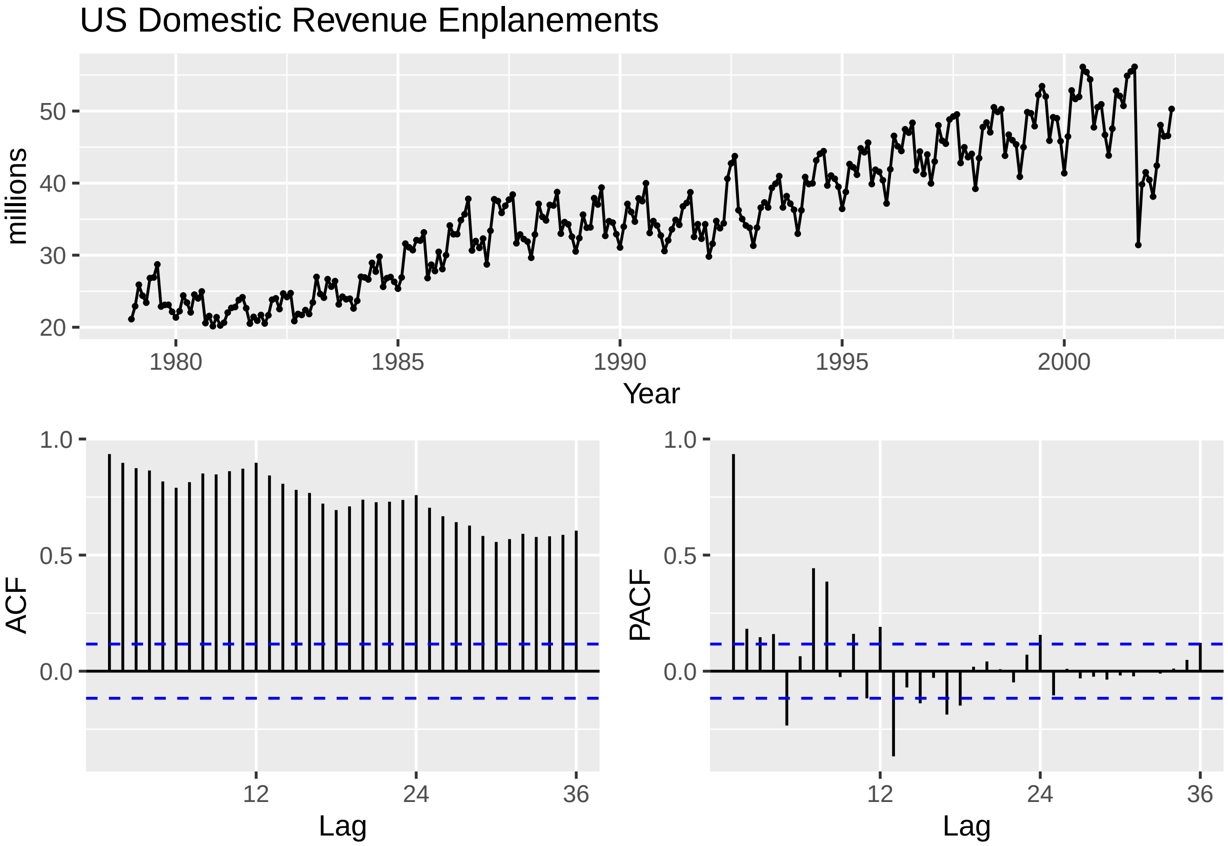
d

```
enplanements
```

```
# Monthly US domestic enplanements
head(enplanements)
```

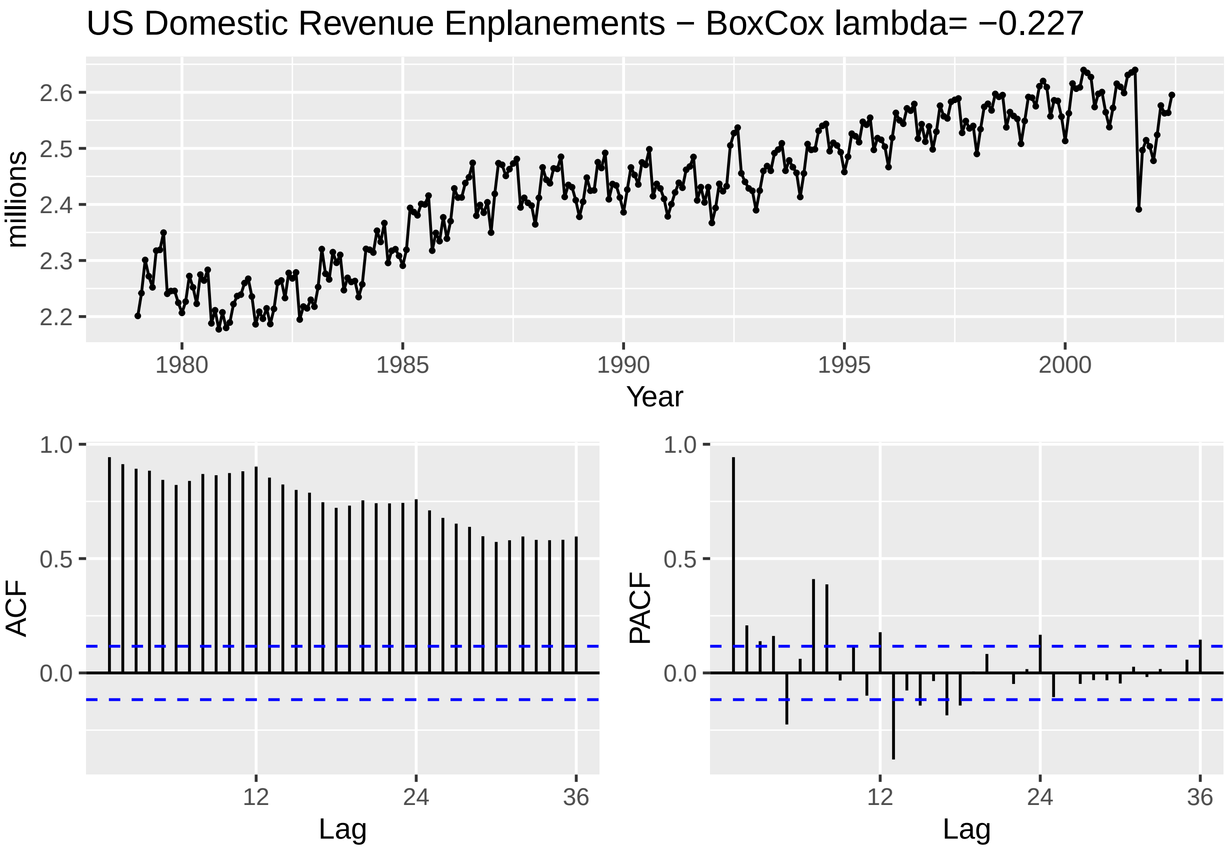
```
##          Jan    Feb    Mar    Apr    May    Jun
## 1979 21.12 22.92 25.90 24.38 23.41 26.82
```

```
ggtsdisplay(enplanements, main="US Domestic Revenue Enplanements", ylab="millions", xlab="Year")
```



This time series has upward trend and seasonality. Also we see a sudden drop in year 2002. Thus this series is non stationary. Now lets do the BoxCox transformation and see the results.

```
enpl.bc_trans <- BoxCox(enplanements, BoxCox.lambda(enplanements))
ggtsdisplay(enpl.bc_trans,
            main=paste("US Domestic Revenue Enplanements - BoxCox lambda=", round(BoxCox.lambda(enplanem
            ylab="millions",
            xlab="Year")
```



It is evident here that Box-Cox transformation, with lambda -0.227, shows the seasonality. Next is to use kpss test in which the null hypothesis is that the data are stationary.

```
enpl.bc_trans %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 4.3785
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected and the boxcox transformed data is not stationary. We will now use ndiffs() function to determine the order of differencing.

```
ndiffs(enpl.bc_trans)
```

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

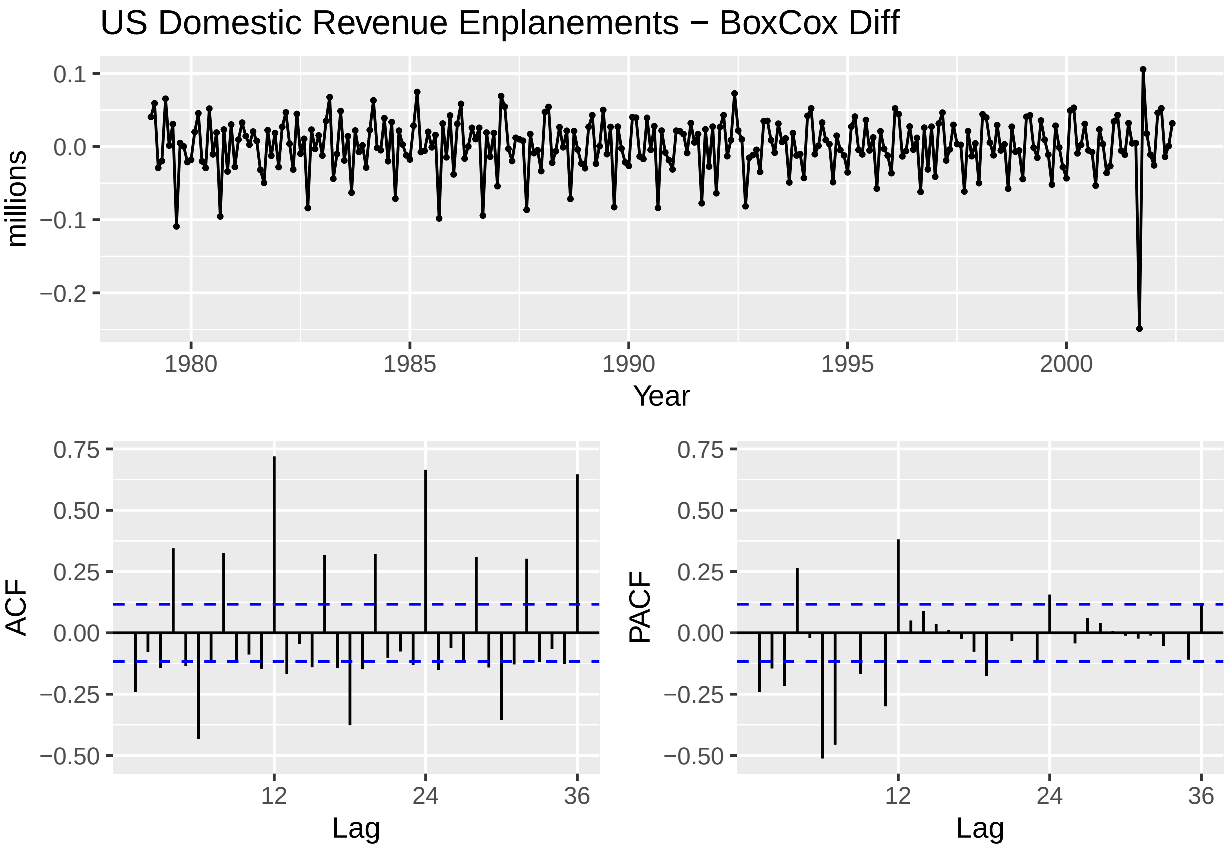
It shows number of differences required is 1 for boxcox transformed data.

```
enpl.bct.diff <- enpl.bc_trans %>% diff()
enpl.bct.diff %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0151
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can see the diff of order 1 makes the test statistic small and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

```
ggtsdisplay(enpl.bct.diff,
            main="US Domestic Revenue Enplanements - BoxCox Diff",
            ylab="millions",
            xlab="Year")
```



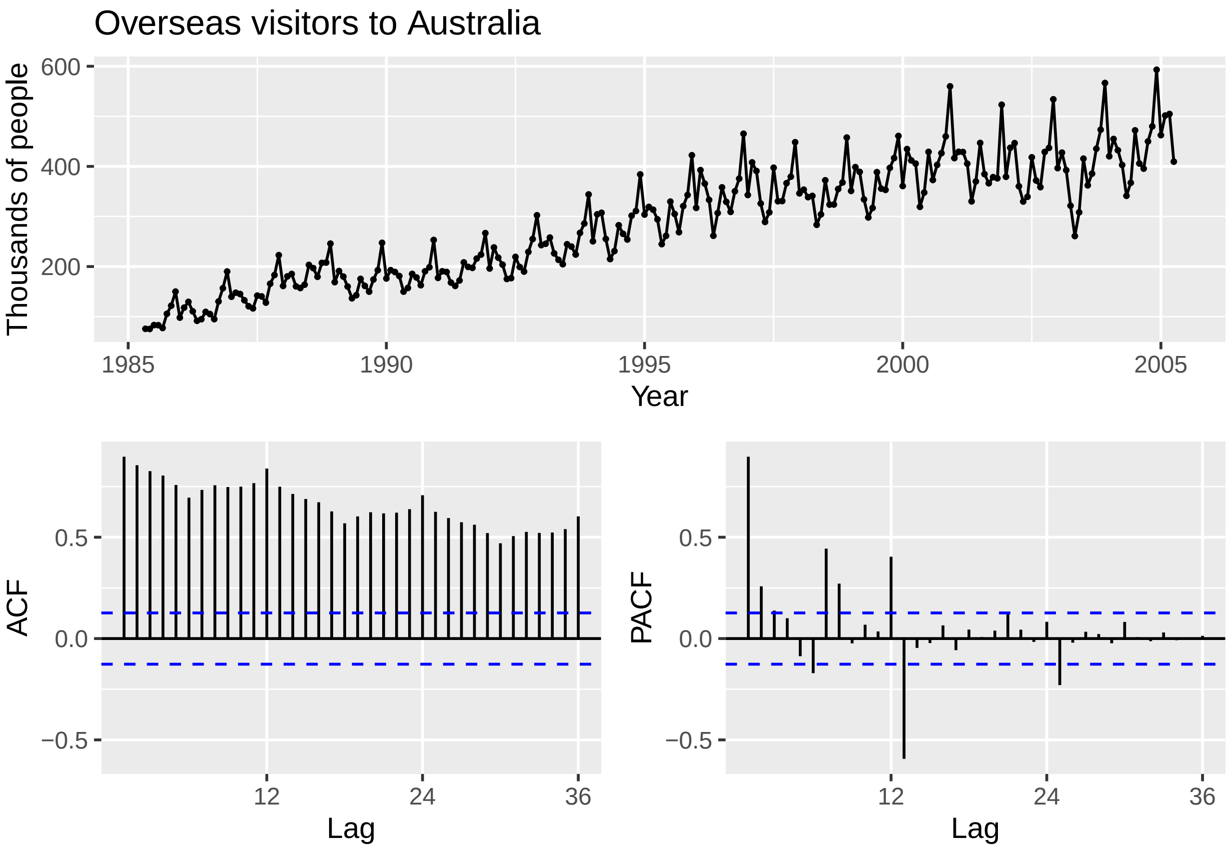
e

visitors

```
# Monthly Australian short-term overseas vistors. May 1985-April 2005
head(visitors)
```

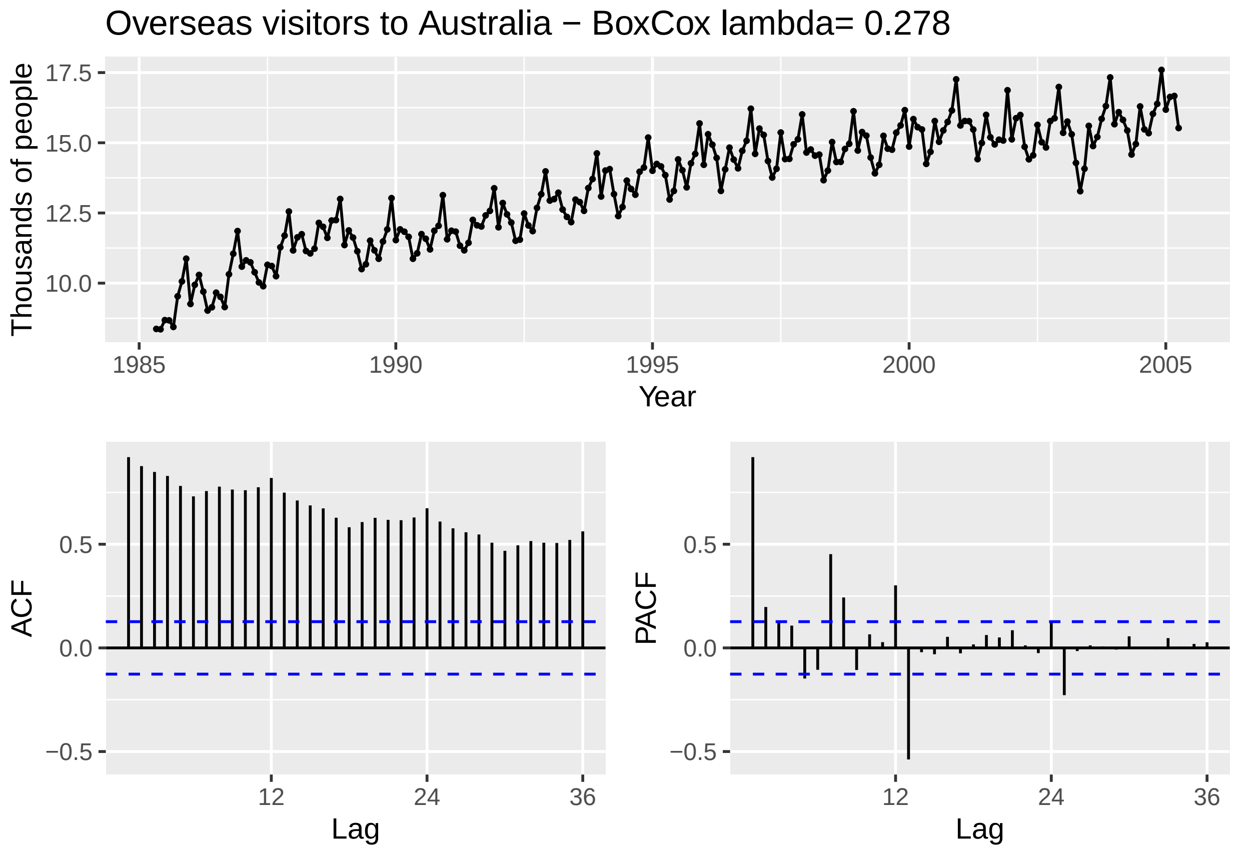
```
##         May   Jun   Jul   Aug   Sep   Oct
## 1985  75.7  75.4  83.1  82.9  77.3 105.7
```

```
ggtsdisplay(visitors,
            main="Overseas visitors to Australia",
            ylab="Thousands of people",
            xlab="Year")
```



This time series has upward trend and seasonality. So this series is non stationary. Now lets do the BoxCox transformation and see the results.

```
visi.bc_trans <- BoxCox(visitors, BoxCox.lambda(visitors))
ggtsdisplay(visi.bc_trans,
            main=paste("Overseas visitors to Australia - BoxCox lambda=", round(BoxCox.lambda(visitors)
            ylab="Thousands of people",
            xlab="Year")
```



It is evident here that Box-Cox transformation, with lambda 0.278 does help for variation. Next is to use kpss test in which the null hypothesis is that the data are stationary.

```
visi.bc_trans %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 4.5233
##
## Critical value for a significance level of:
##                  10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected and the boxcox transformed data is not stationary. We will now use ndiffs() function to determine the order of differencing.

```
ndiffs(visi.bc_trans)
```

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

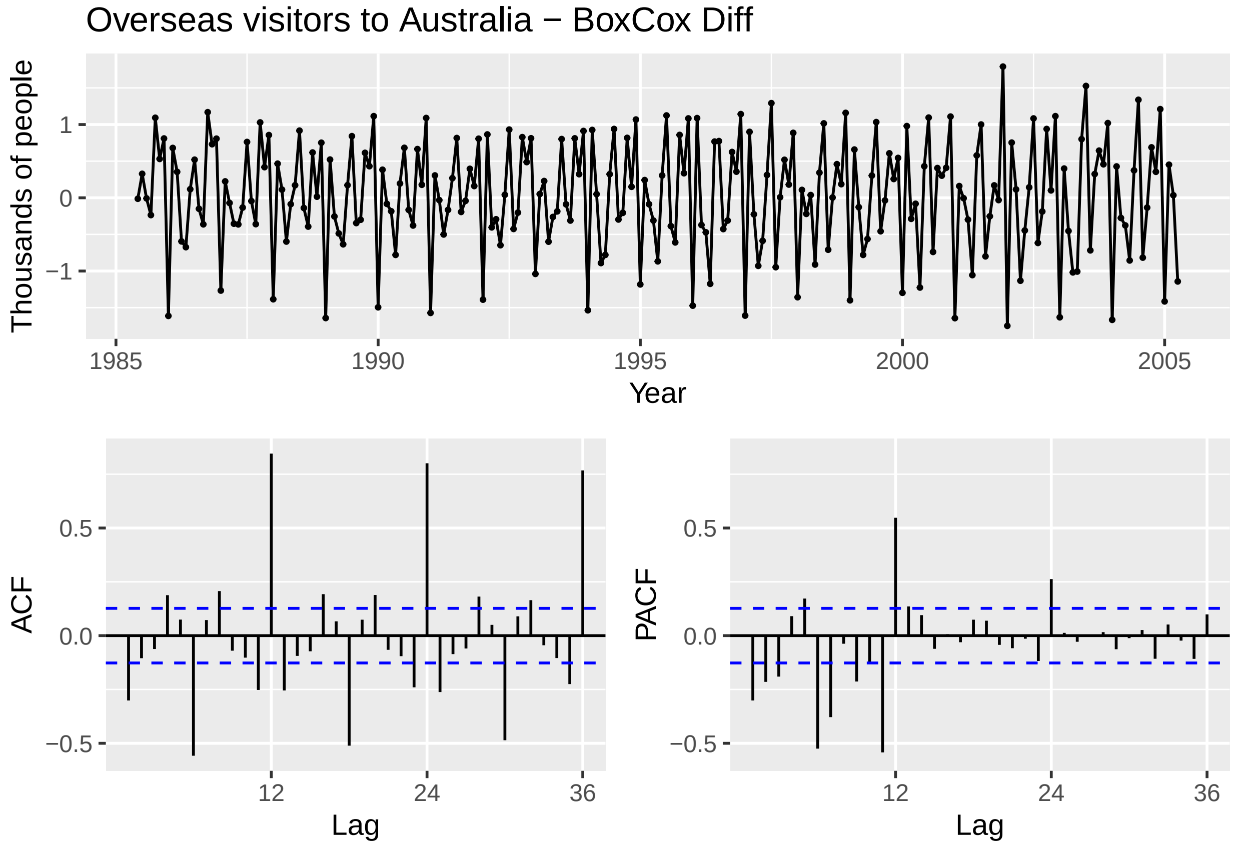
It shows number of differences required is 1 for boxcox transformed data.

```
visi.bct.diff <- visi.bc_trans %>% diff()
visi.bct.diff %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 0.0519
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can see the diff of order 1 makes the test statistic small and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

```
ggtsdisplay(visi.bct.diff,
            main="Overseas visitors to Australia - BoxCox Diff",
            ylab="Thousands of people",
            xlab="Year")
```

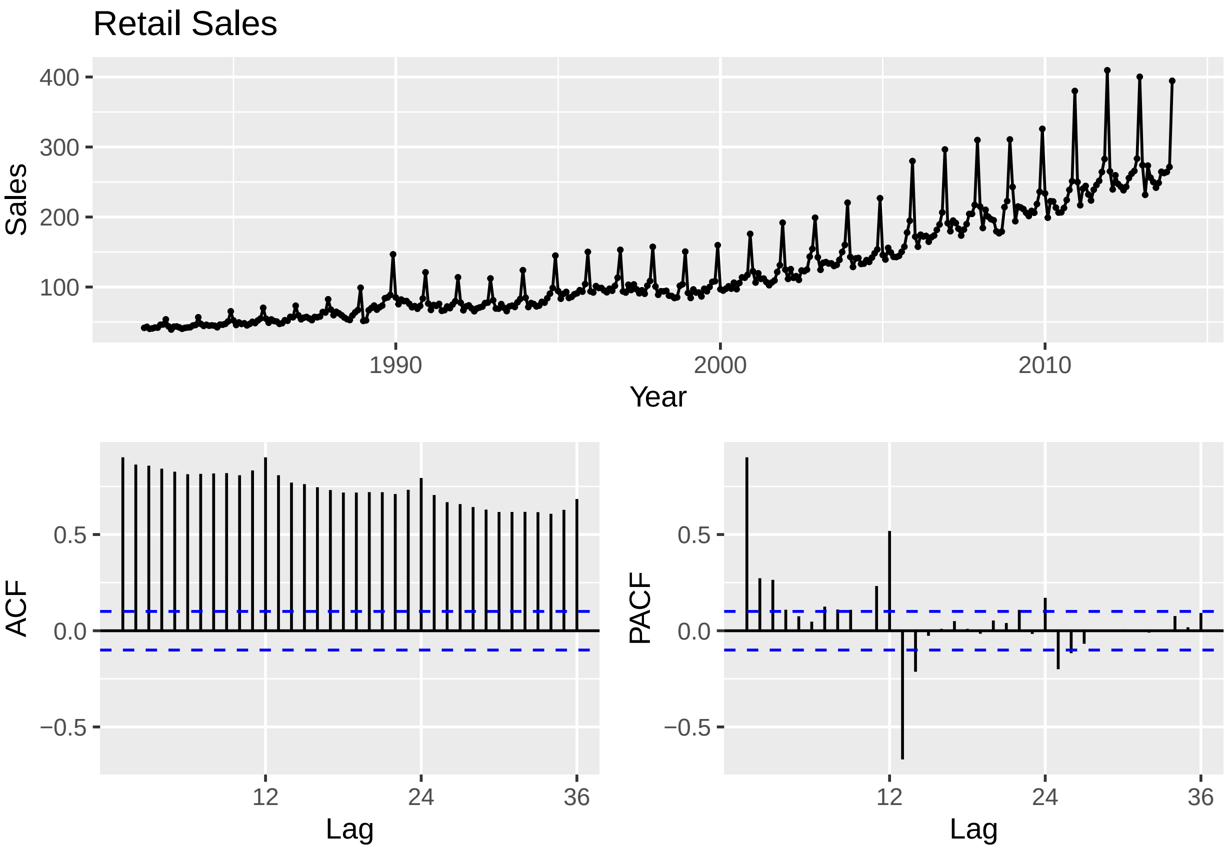


Exercise 8.5

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.

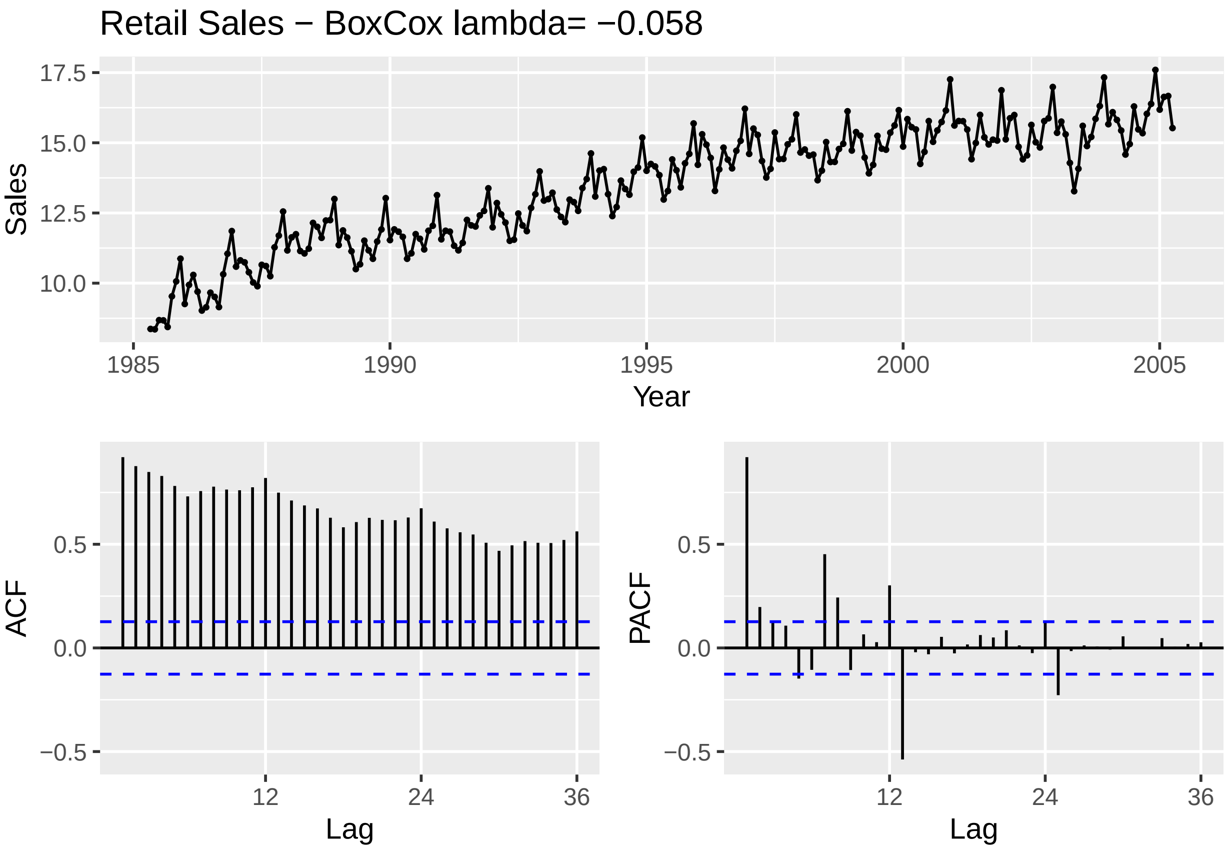
```
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
myts <- ts(retaildata[,"A3349627V"], frequency=12, start=c(1982,4))
```

```
ggtsdisplay(myts,
            main="Retail Sales",
            ylab="Sales",
            xlab="Year")
```



There is a clear annual seasonality increase in retail sales from October to December. I see a consistent upward trend and dont see cyclicity. Now lets do the BoxCox transformation and see the results.

```
sale.bc_trans <- BoxCox(myts, BoxCox.lambda(myts))
ggtsdisplay(visi.bc_trans,
            main=paste("Retail Sales - BoxCox lambda=", round(BoxCox.lambda(myts), 3)),
            ylab="Sales",
            xlab="Year")
```



t is evident here that Box-Cox transformation, with lambda -0.058 does help for seasonal variation. Next is to use kpss test in which the null hypothesis is that the data are stationary.

```
sale.bc_trans %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 6.2172
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected and the boxcox transformed data is not stationary. We will now use ndiffs() function to determine the order of differencing.

```
ndiffs(sale.bc_trans)
```

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

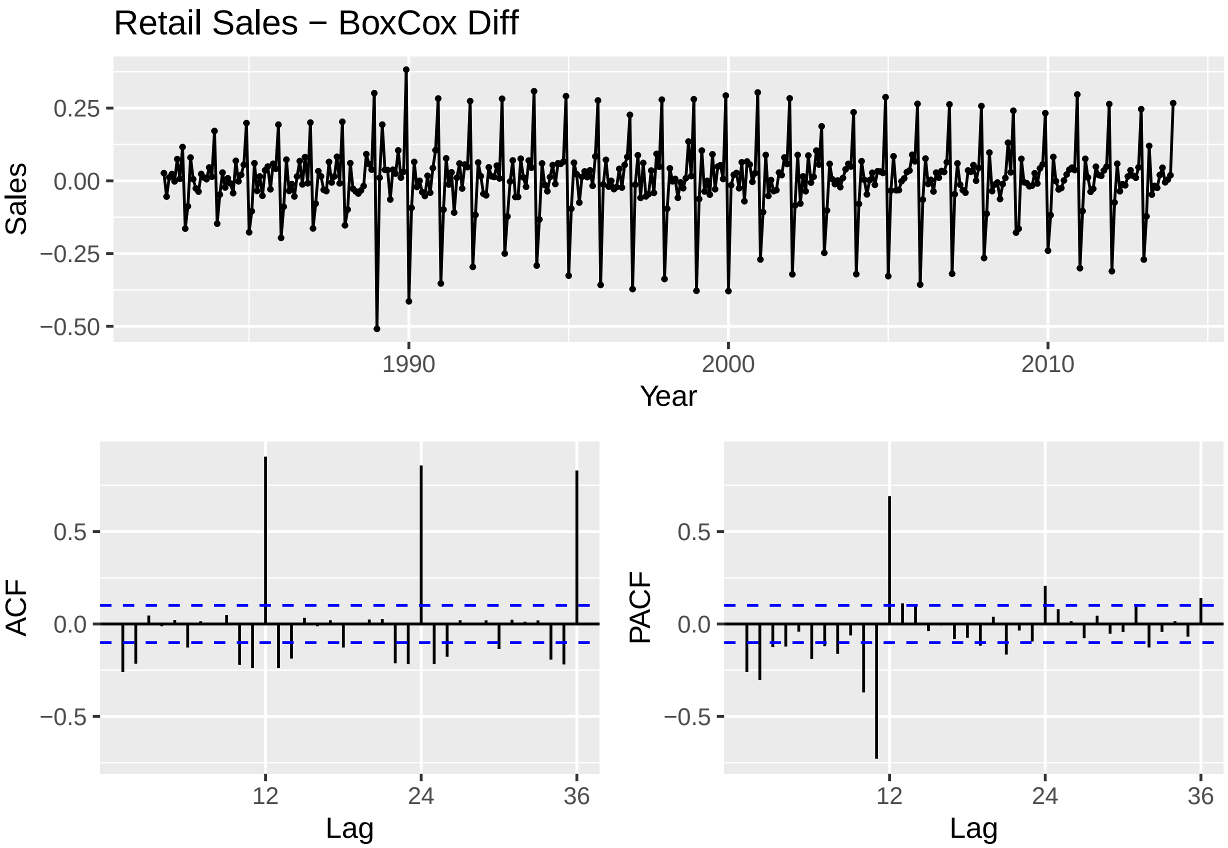
It shows number of differences required is 1 for boxcox transformed data.

```
sale.bct.diff <- sale.bc_trans %>% diff()
sale.bct.diff %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0175
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can see the diff of order 1 makes the test statistic small and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

```
ggtsdisplay(sale.bct.diff,
            main="Retail Sales - BoxCox Diff",
            ylab="Sales",
            xlab="Year")
```



Exercise 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 $y_1 = 0$.

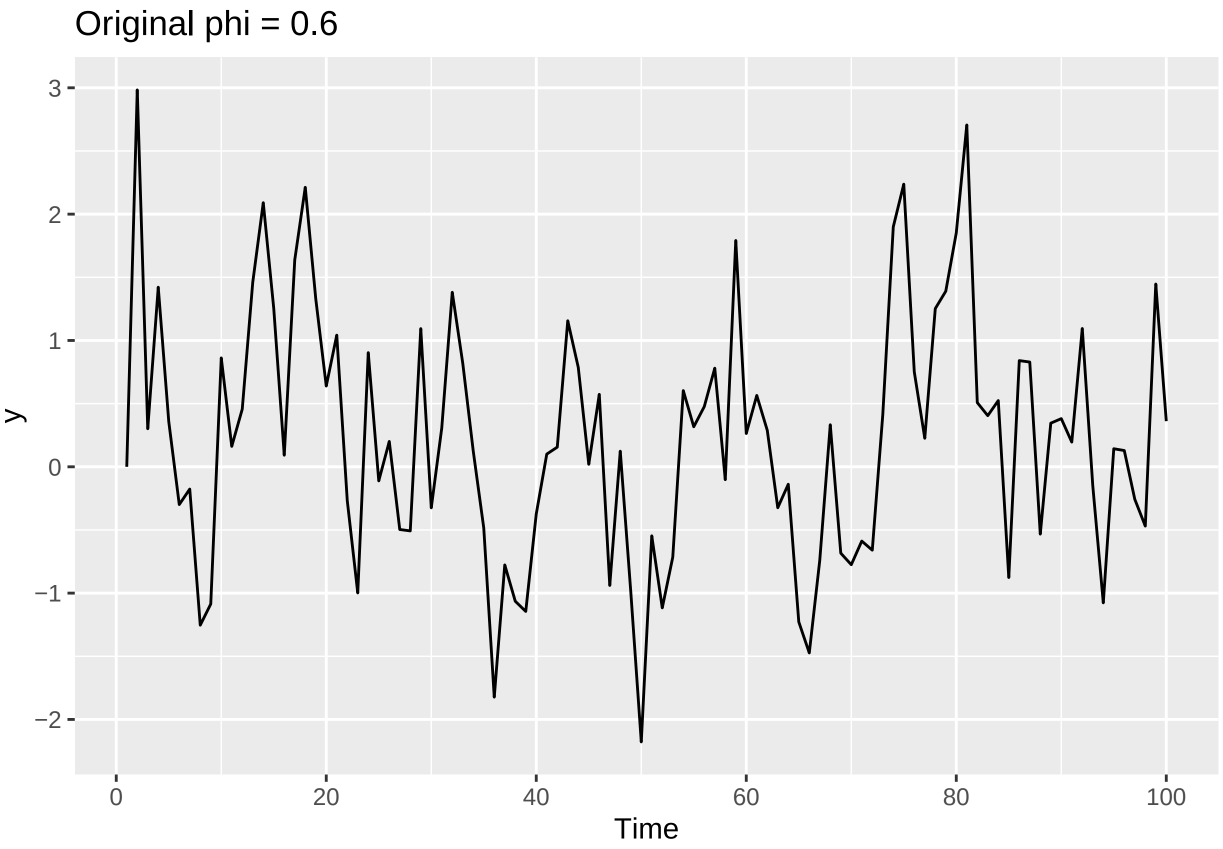
```
y <- ts(numeric(100))
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.6*y[i-1] + e[i]
```

b

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

Lets first draw the plot of original y.

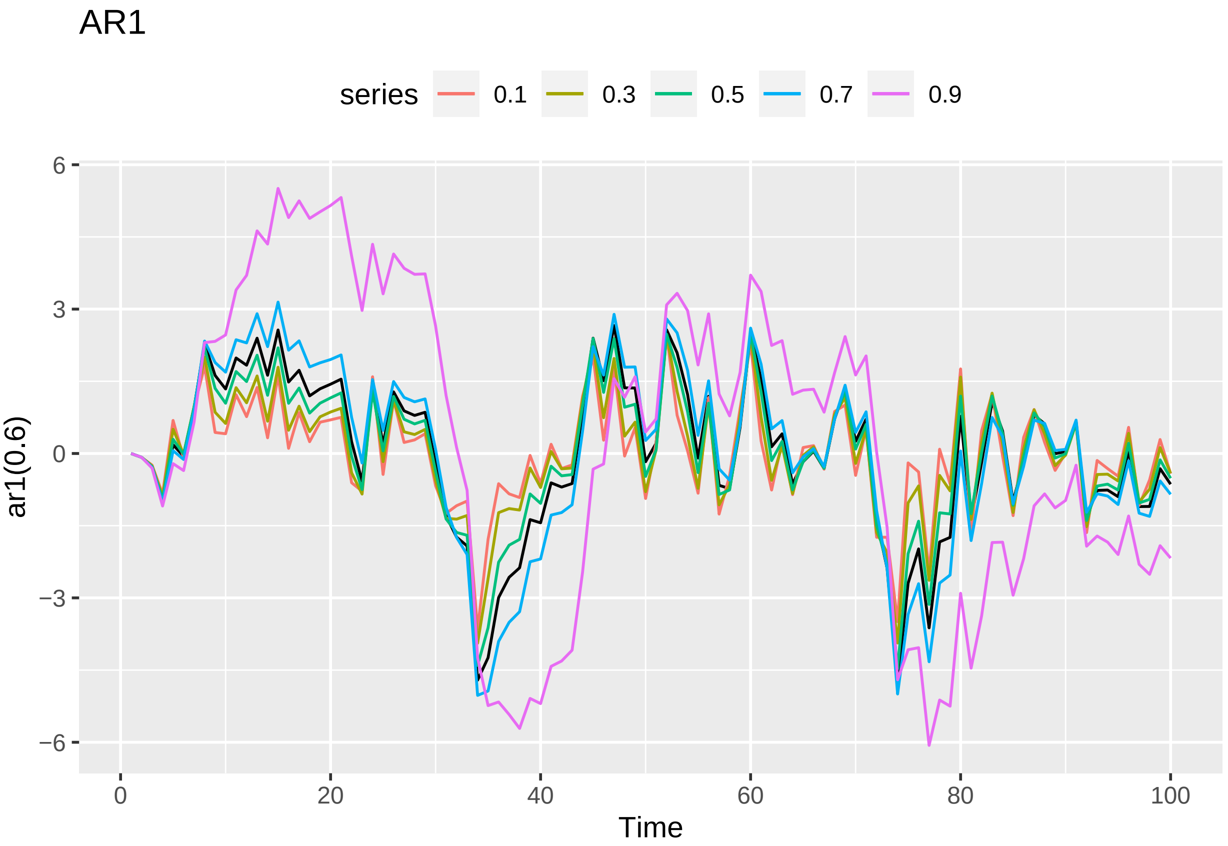
```
autoplot(y) + labs(title=paste("Original phi =", 0.6))
```



We will now create a function which will accept phi and return the series. Next step is to get the series for different values of phi and and then plot them. Here we will have phi values 0.1, 0.3, 0.5, 0.7, 0.9 to plot different series.

```
ar1 <- function(phi) {
  set.seed(17)
  y <- ts(numeric(100))
  e <- rnorm(100)

  for(i in 2:100)
    y[i] <- phi*y[i-1] + e[i]
  return(y)
}
```

```
# initialize p
p <- autoplot(ar1(0.6))
# create series
for(phi in seq(0.1,1, 0.2)) {
  p <- p + autolayer(ar1(phi), series=paste(phi))
}
# plot all series
p +
  theme(legend.position = "top") +
  labs(title = "AR1")
```



It is apparent that as the value of ϕ_1 increases, the distance from 0 increases and that causes higher correlation.

c

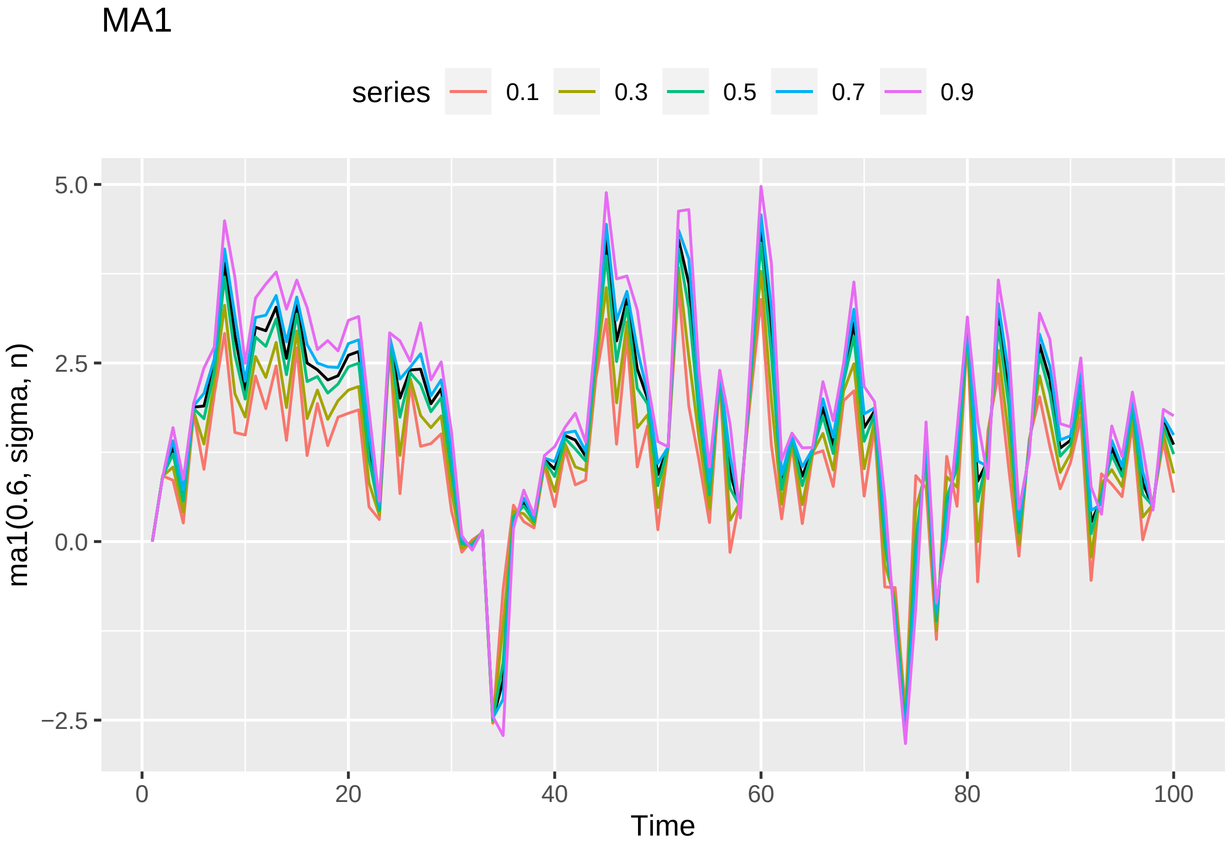
Write your own code to generate data from an MA(1) model with $\theta_1 = 0.6$ and $\sigma^2 = 1$.

```
ma1 <- function(theta, sigma, n) {
  set.seed(17)
  y <- ts(numeric(n))
  e <- rnorm(n, sigma)

  for(i in 2:n)
    y[i] <- theta*e[i-1] + e[i]
  return(y)
}
```

d

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

```
sigma <-1
n <- 100

# initialize p
p <- autoplot(ma1(0.6,sigma,n))

# create series
for(theta in seq(0.1,1, 0.2)) {
  p <- p + autolayer( ma1(theta,sigma,n), series=paste(theta))
}

# plot all series
p +
  theme(legend.position = "top")  +
  labs(title = "MA1")
```

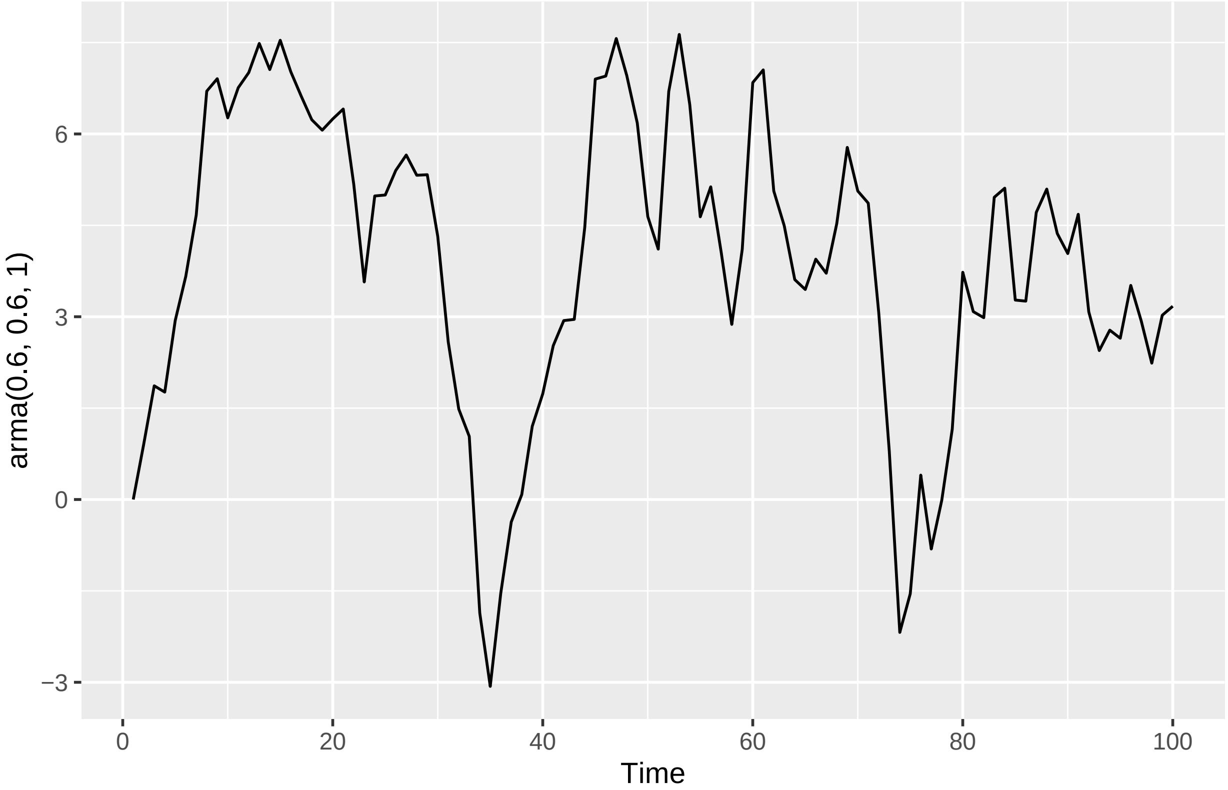


Similar to AR(1) model, MA(1) changes as well with θ_1 changes. As θ_1 increases, it shows higher correlation.

e

Generate data from an ARMA(1,1) model with $\phi_1 = 0.6$, $\theta_1 = 0.6$ and $\sigma^2 = 1$.

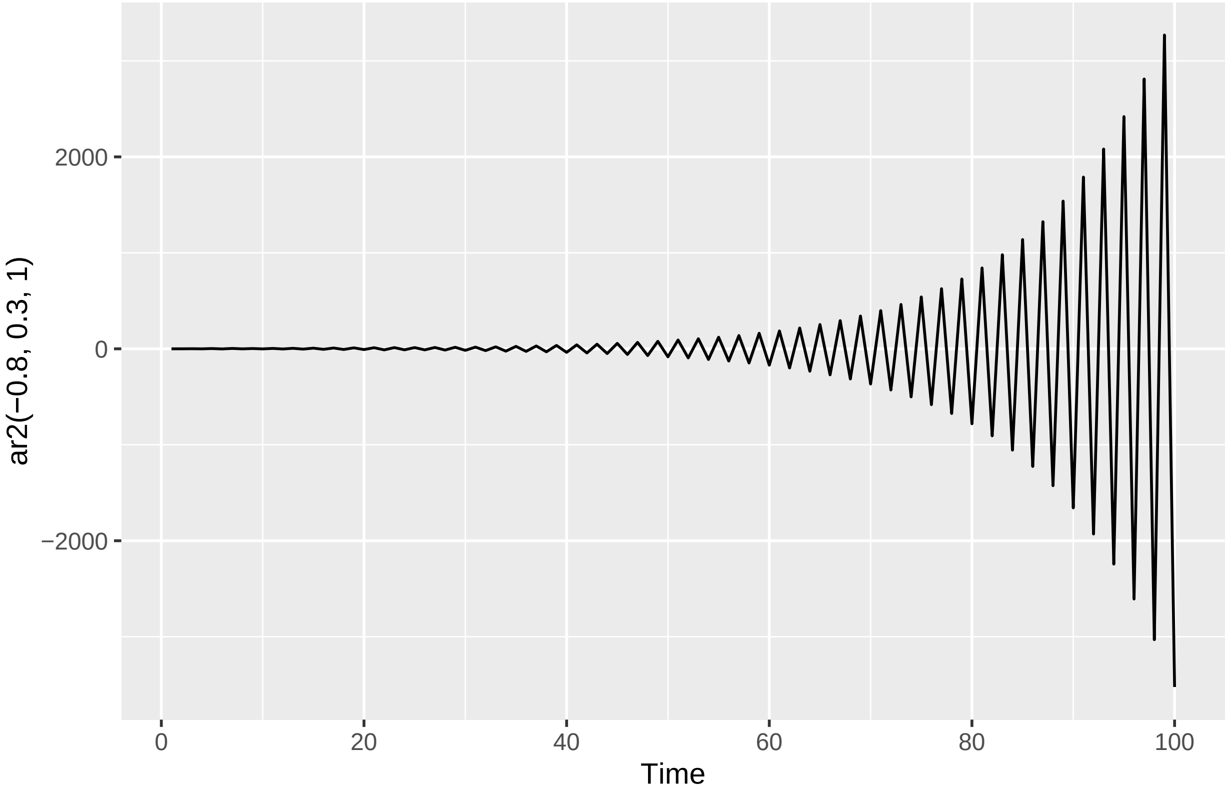
```
# function to return data for ARMA
arma <- function(phi, theta, sigma) {

  set.seed(17)
  y <- ts(numeric(100))
  e <- rnorm(100, sigma)
  for(i in 2:100)
    y[i] <- phi*y[i-1] + theta*e[i-1] + e[i]

  return(y)
}

# plot the data with phi=0.6, theta=0.6 and sigma=1
autoplot(arma(0.6, 0.6, 1))
```



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.)

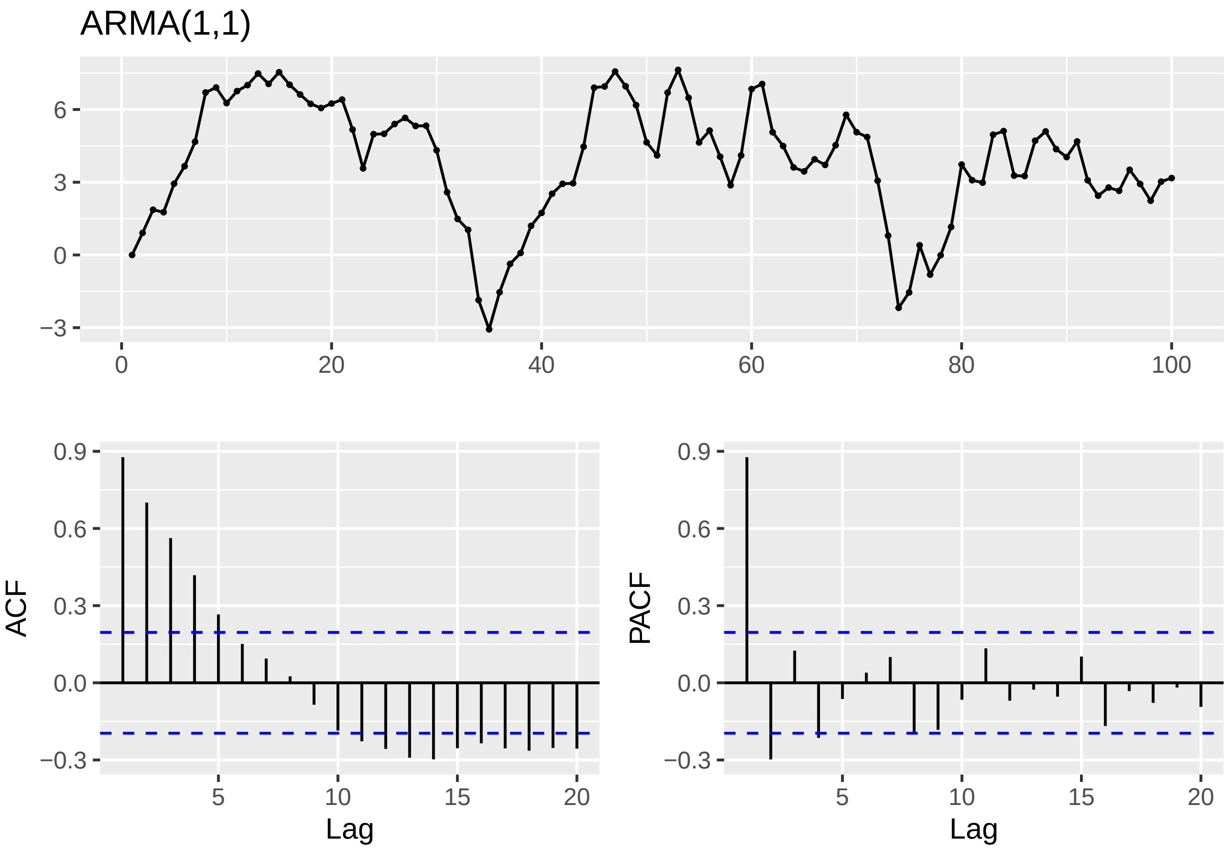
```
# function to return data for ARMA
ar2 <- function(phi1, phi2, sigma) {

  set.seed(17)
  y <- ts(numeric(100))
  e <- rnorm(100, sigma)
  for(i in 3:100)
    y[i] <- phi1*y[i-1] + phi2*y[i-2] + e[i]
  return(y)
}

# plot the data with phi1=-0.8, phi2=0.3 and sigma=1
autoplot(ar2(-0.8, 0.3, 1))
```



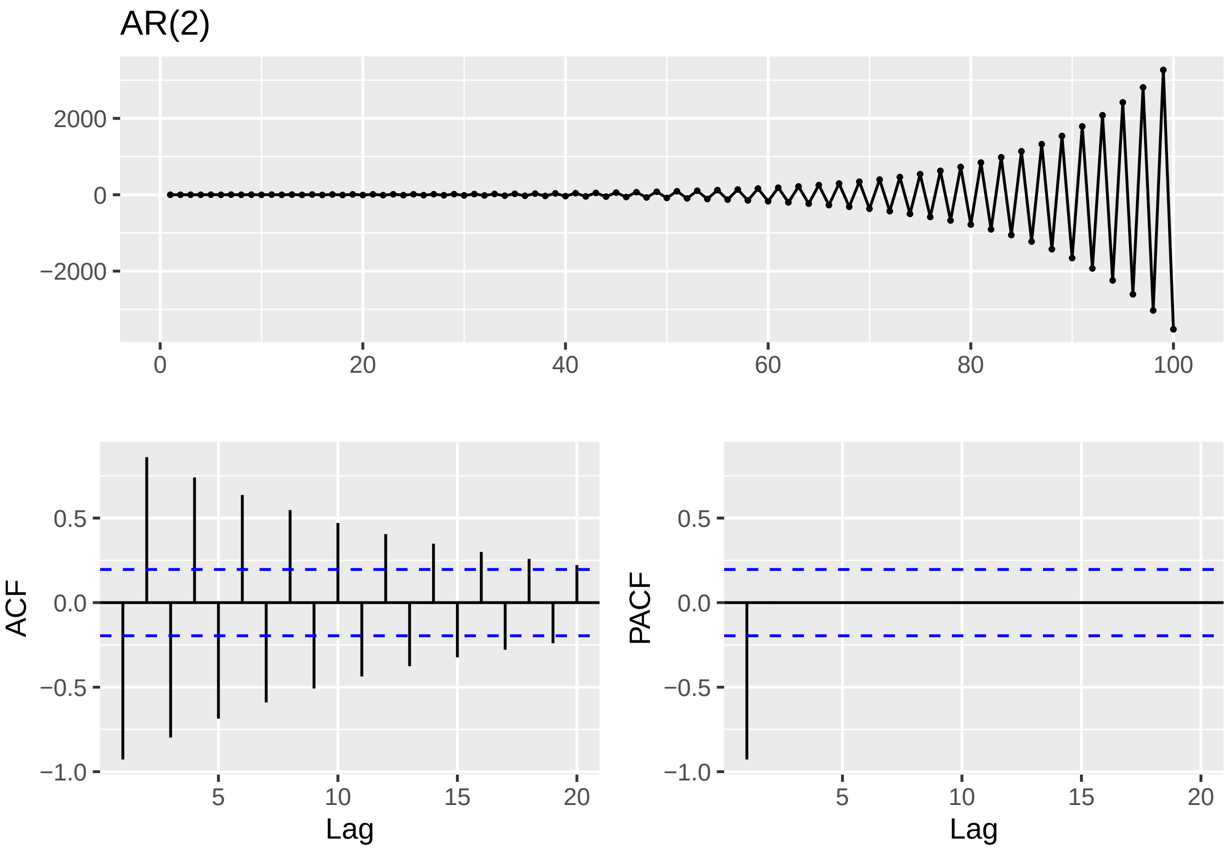
g

Graph the latter two series and compare them.

```
ggtsdisplay(arma(0.6, 0.6, 1), , main="ARMA(1,1)")
```



```
ggtsdisplay(ar2(-0.8, 0.3, 1), main="AR(2)")
```



`ARMA(1,1)` seems like stationary and almost white noise. On the other hand `AR(2)` starts like a line and gradually started to grow and eventually increases exponentially over the time.

Exercise 8.7

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

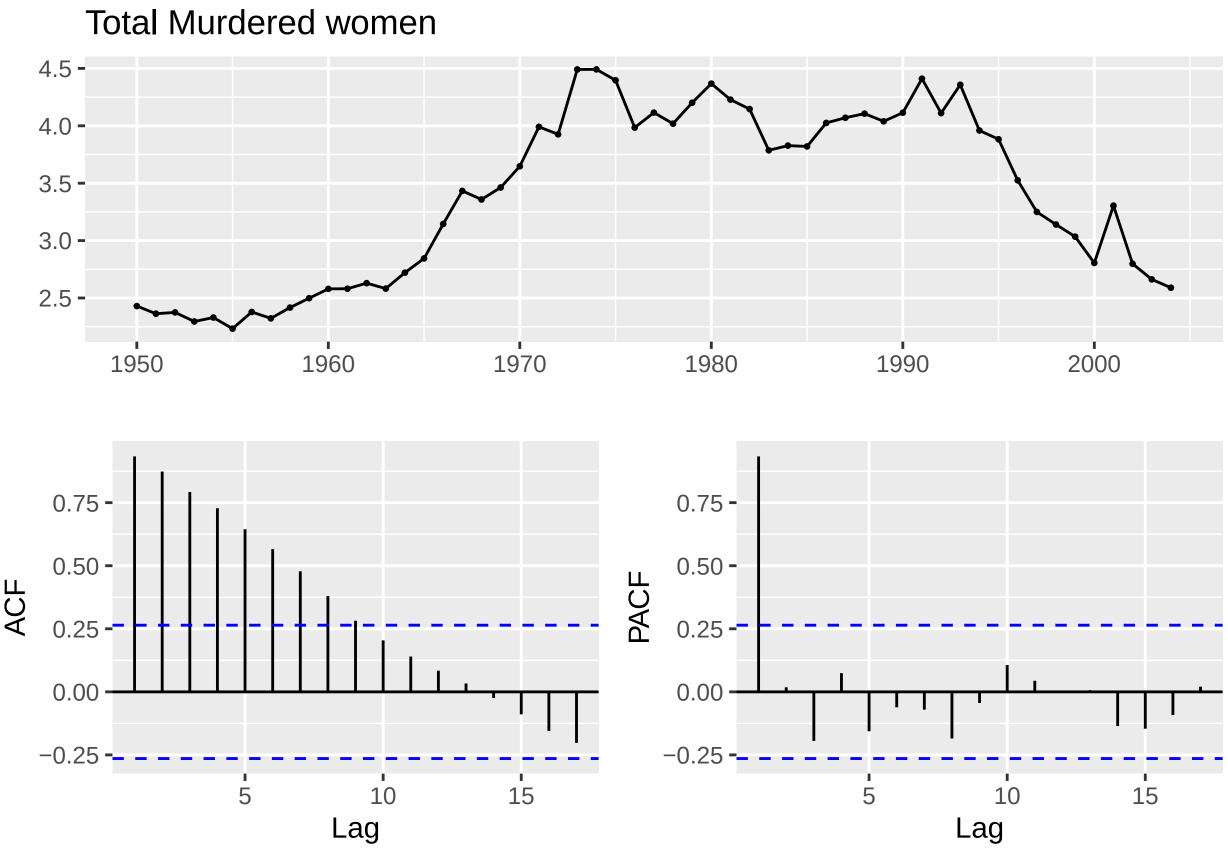
a

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

```
# Total Murdered women, per 100 000 standard population.
head(wmurders)
```

```
## Time Series:
## Start = 1950
## End = 1955
## Frequency = 1
## [1] 2.429415 2.363364 2.374305 2.295520 2.329716 2.233017
```

```
ggtsdisplay(wmurders, main = "Total Murdered women")
```



The data initially shows an upward trend followed by oscillation and then declines. We dont see any seasonality variation so BoxCox transformation is not needed here. Lets find out if differencing required to make this time series stationary.

```
ndiffs(wmurders)
```

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

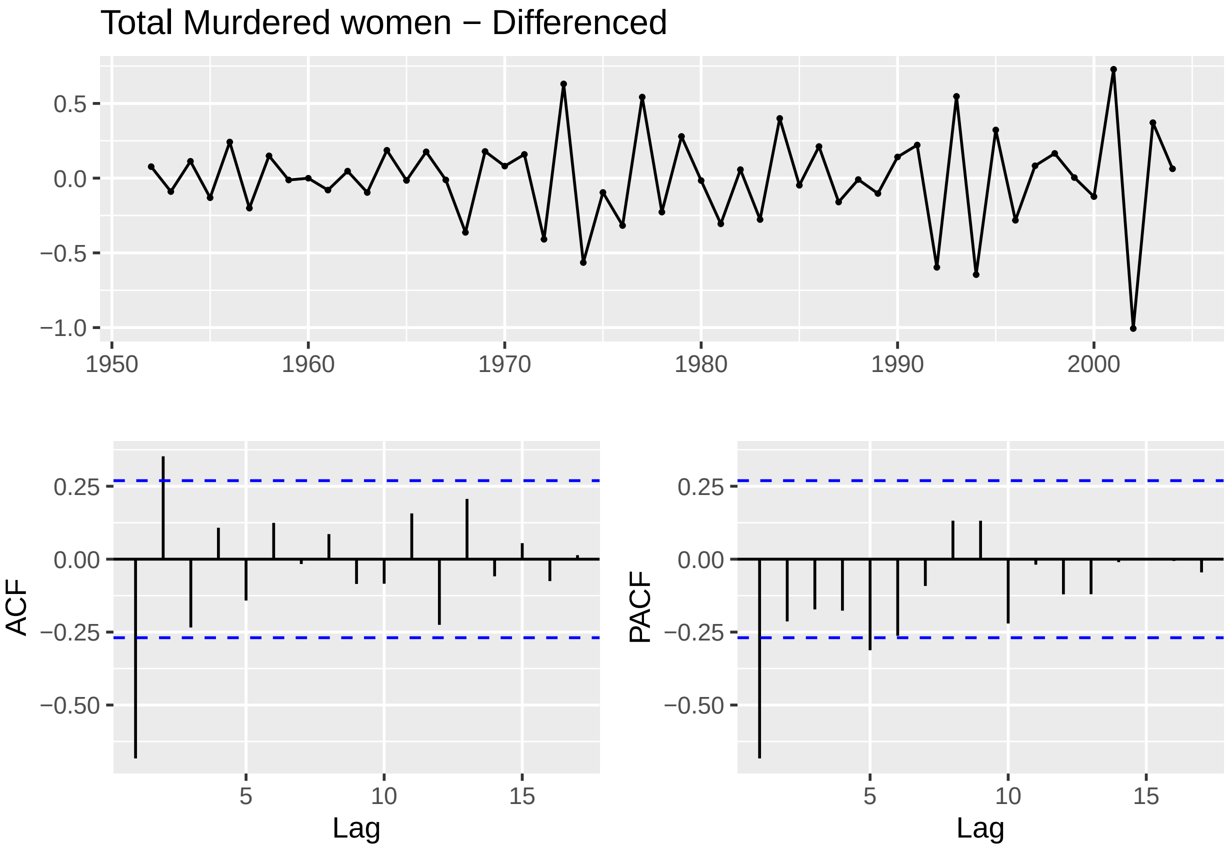
Next step is to perform differencing and apply kpss unit root test.

```
wmurders.diff <- wmurders %>% diff(differences=2)
wmurders.diff %>% ur.kpss() %>% summary()
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0458
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

We can see the diff of order 1 makes the test statistic small and well within the range we would expect for stationary data. So we can conclude that the differenced data are stationary.

```
ggtsdisplay(wmurders.diff, main = "Total Murdered women - Differenced")
```



ARIMA(p,d,q) model, p= order of the autoregressive part, d=degree of first differencing involved and q= order of the moving average part. The first lag of PACF shows the only spike which suggests p as 1, ndiff suggests difference is of order 2 so d will 2. ACF plot shows significant spike at 1 so q=2. Hence **ARIMA(1,2,1)** seems appropriate here.

b

Should you include a constant in the model? Explain.

As mentioned here, If c=0 and d=2, the long-term forecasts will follow a straight line. If c $\neq$ 0 and d = 2, the long-term forecasts will follow a quadratic trend which is not recommended. In this case d=2, so we will not include constant in the model.

c

Write this model in terms of the backshift operator

$$(1 - \phi_1 B)(1 - B)^2 y_t = (1 + \theta_1 B)\epsilon_t$$

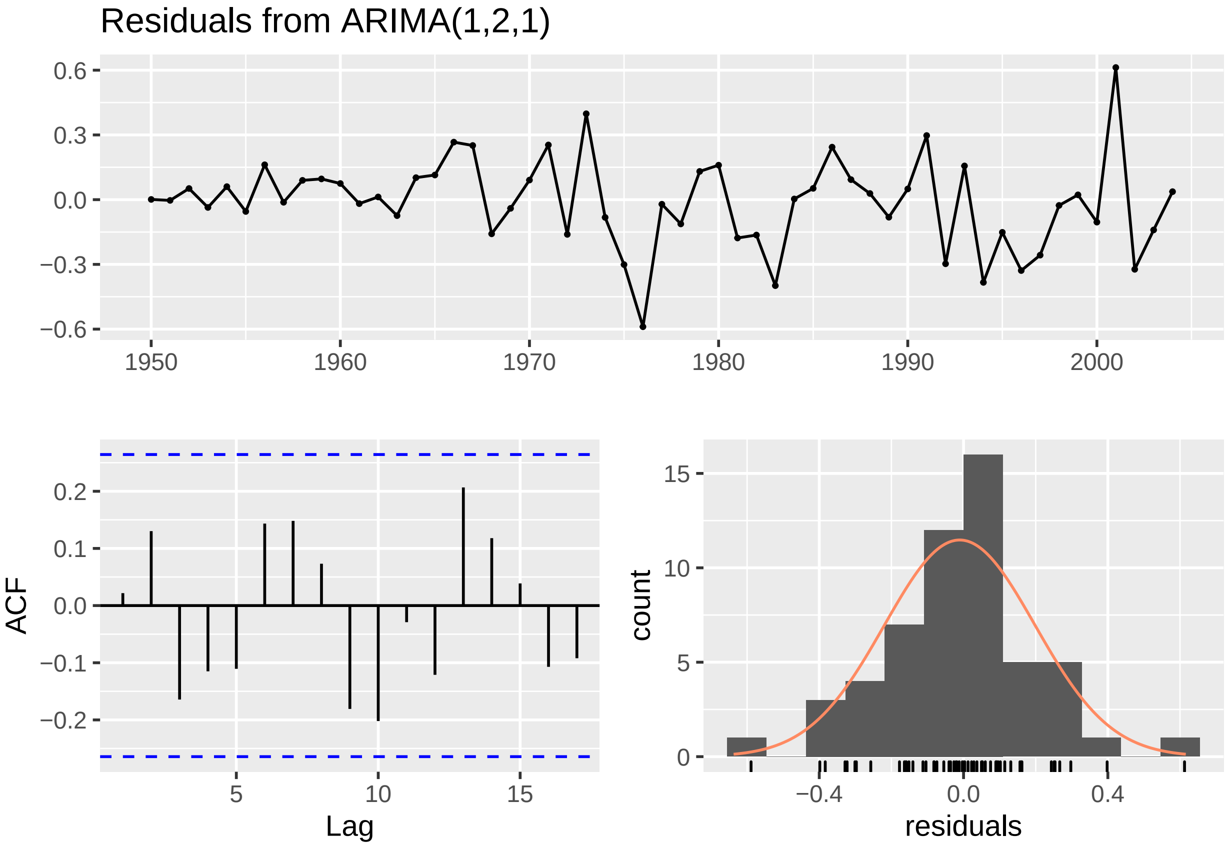
d

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

```
(fit <- Arima(wmurders, order = 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(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
```

The ACF plot from ARIMA(1,2,1) shows all autocorrelations are within acceptable limit, indicating residuals are white noise. The histogram doesnt appear highly skewed as mean appears newar to 0. The p-value from Ljung-Box test indicates the residuals are white noise. Hence the model will be

$$(1+0.24B)(1-B)^2y_t = (1-0.83B)\epsilon_t$$

e

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

```
forecast(fit, h=3)
```

```
##      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
```

By hand, the forecast three times ahead is calculated below:

$$(1-\phi_1 B)(1-B)^2 y_t = (1+\theta_1 B)\epsilon_t \tag{1}$$

After transpose and factoring, the equation could be written as:

$$y_t = (2+\phi_1)y_{t-1} - (1+2\phi_1)y_{t-2} + \phi_1 y_{t-3} + \epsilon_t + \theta_1\epsilon_{t-1}$$

Lets replace values as $\phi_1 = -0.2434$, $\theta_1 = -0.8261$ and $\epsilon_t = 0$ and do the calculations

```
t <- length(wmurders)
e <- fit$residuals
fc_1 <- round( ( (2-0.2434)*wmurders[t] - (1-2*0.2434)*wmurders[t-1] - 0.2434*wmurders[t-2] - 0.8261*e[t
fc_2 <- round( ( (2-0.2434)*fc_1 - (1-2*0.2434)*wmurders[t] - 0.2434*wmurders[t-1] - 0.8261*0 ), 3)
fc_3 <- round( ( (2-0.2434)*fc_2 - (1-2*0.2434)*fc_1 - 0.2434*wmurders[t] - 0.8261*0  ), 3)

df <- data.frame(forecast=c(1,2,3),
                 values=c(fc_1,fc_2,fc_3))
df
```

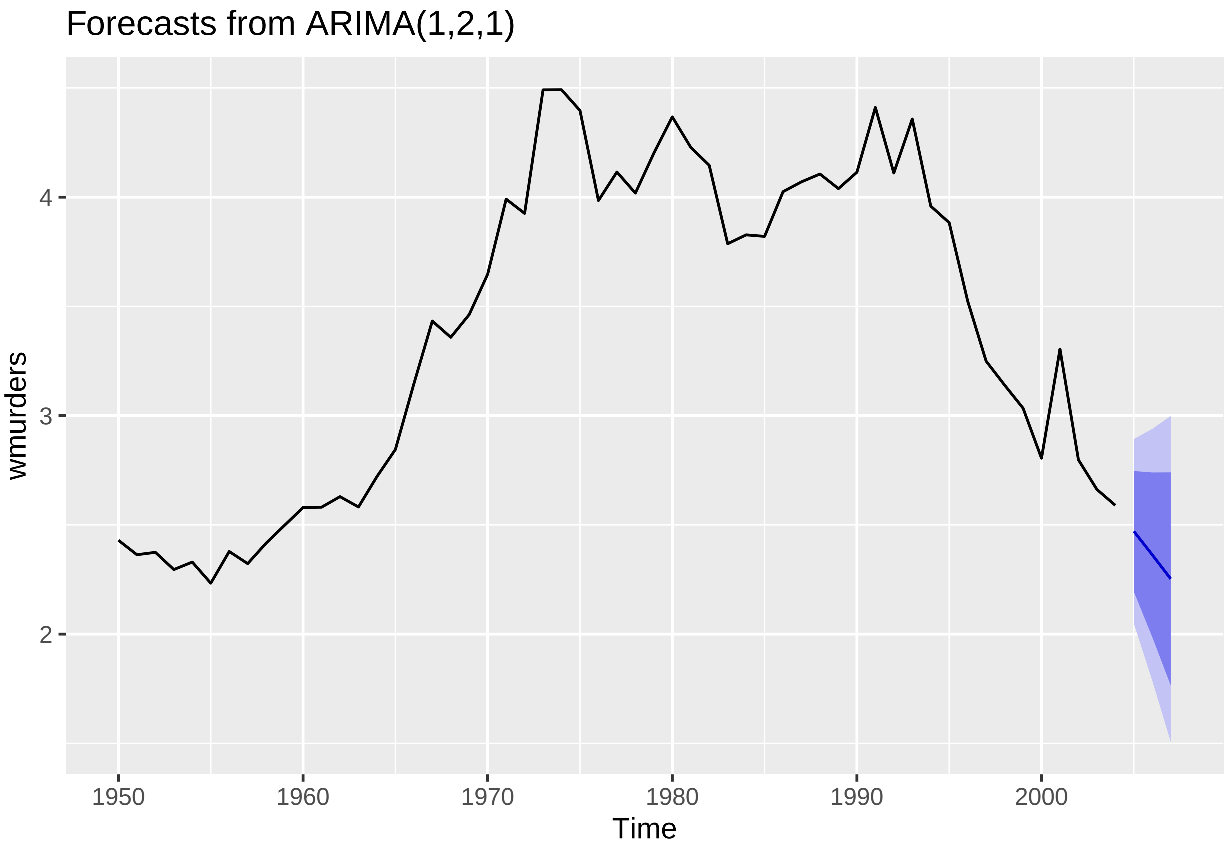
```
##   forecast values
## 1        1  2.471
## 2        2  2.364
## 3        3  2.254
```

We see the above values are very close to one calculated by R.

f

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

```
autoplot(forecast(fit, h=3))
```



The plot shows decreasing trend for forecast in the next three periods.

g

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

```
auto.arima(wmurders)
```

```
## 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
```

auto.arima() produces the same **ARIMA(1,2,1)** model that we have chosen earlier at solution **a**.