Data 624_Exercise 9.11_HW6

Jiaxin Zheng

2025-04-04

9.11 Exercises:

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.4.2

## Registered S3 method overwritten by 'tsibble':
##   method               from
##   as_tibble.grouped_df dplyr

## -- Attaching packages -------------------------------------------- fpp3 1.0.1 --

## v tibble      3.2.1     v tsibble     1.1.6
## v dplyr       1.1.4     v tsibbledata 0.4.1
## v tidyr       1.3.1     v feasts      0.4.1
## v lubridate   1.9.4     v fable       0.4.1
## v ggplot2     3.5.1

## Warning: package 'dplyr' was built under R version 4.4.3

## Warning: package 'ggplot2' was built under R version 4.4.2

## Warning: package 'tsibbledata' was built under R version 4.4.2

## Warning: package 'feasts' was built under R version 4.4.2

## Warning: package 'fabletools' was built under R version 4.4.2

## Warning: package 'fable' was built under R version 4.4.2

## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```r
library(dplyr)
```

1. Figure 9.32 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.

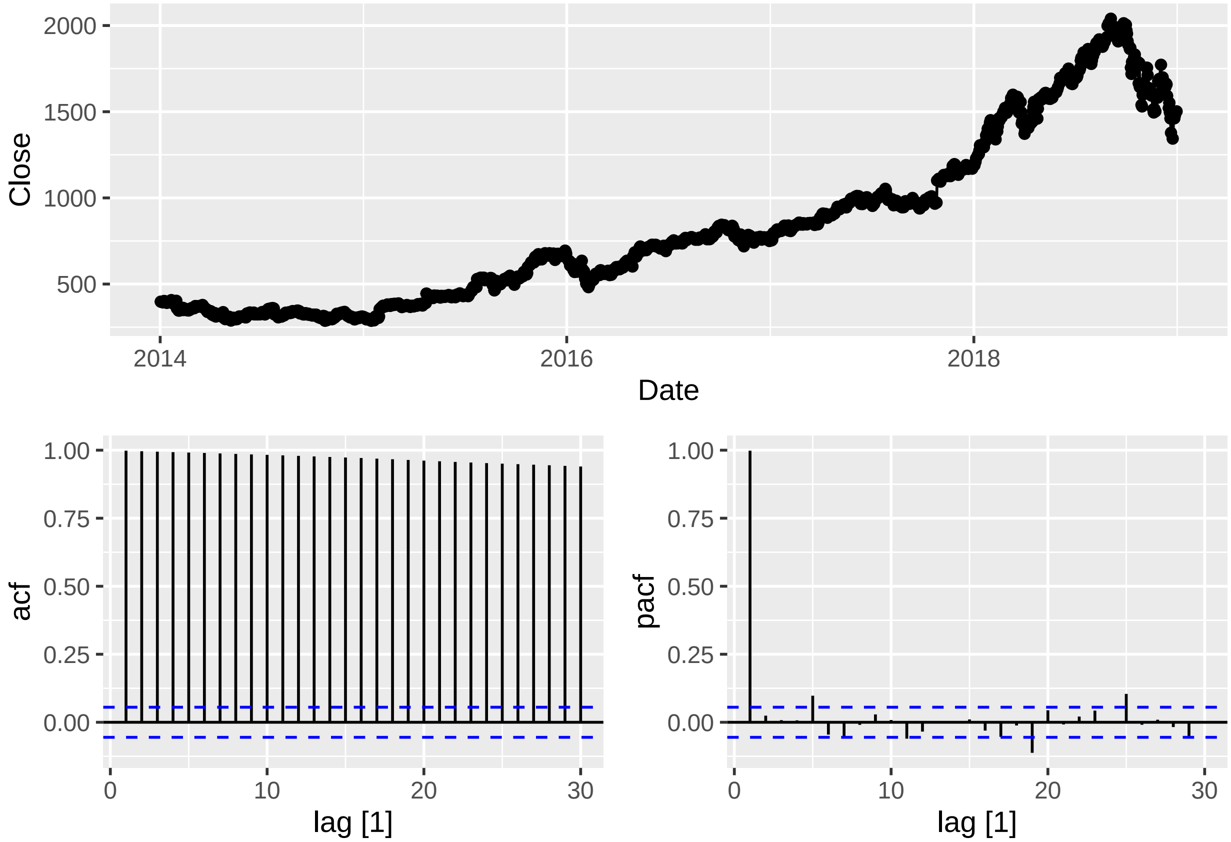
- a.Explain the differences among these figures. Do they all indicate that the data are white noise?
- Answer: It looks like all the plots, the spikes are all within the blueline. ALl of the ACF plots indicate the data is white noise.
- b.Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?
- Answer: As the sample size increase the critical values are more precise. The sample size progresses from 36, 360 to 1,000, the critical values approach to zero.

2. A classic example of a non-stationary series are stock prices. Plot the daily closing prices for Amazon stock (contained in gafa_stock), along with the ACF and PACF. Explain how each plot shows that the series is non-stationary and should be differenced.

- Answer:The data is not stationary and plenty of variation. I make the difference plot for the daily closing prices for Amazon. Both ACF and PACF shoes many spikes, and the data is not white noise.

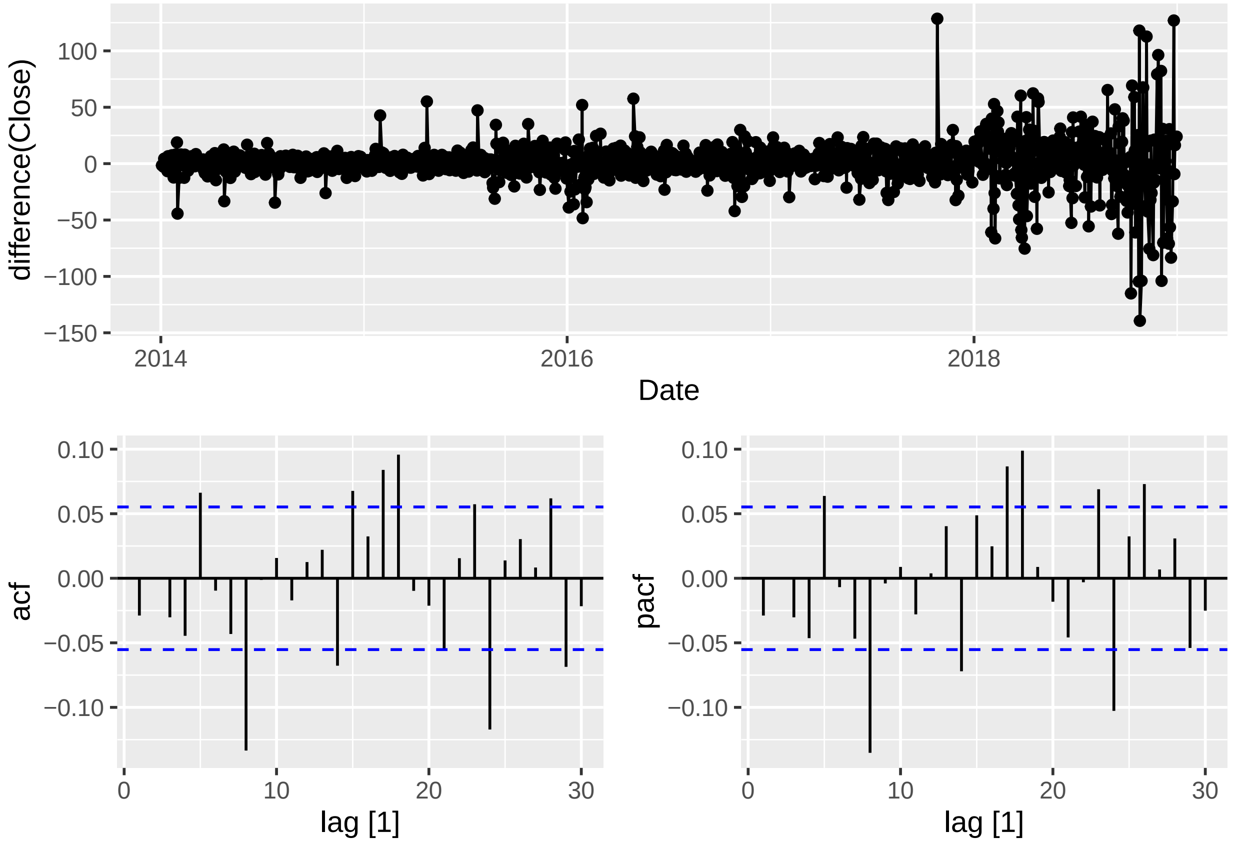
```r
amazon <- gafa_stock %>%
  filter(Symbol == "AMZN")

amazon %>%
  gg_tsdisplay(Close, plot_type = 'partial')
```

```
## Warning: Provided data has an irregular interval, results should be treated with caution. Computing A
## Provided data has an irregular interval, results should be treated with caution. Computing ACF by obs
```



```r
amazon <- gafa_stock %>%
  filter(Symbol == "AMZN")

amazon %>%
  gg_tsdisplay(difference(Close), plot_type = 'partial')
```

```
## Warning: Provided data has an irregular interval, results should be treated with caution. Computing A
## Provided data has an irregular interval, results should be treated with caution. Computing ACF by obs

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



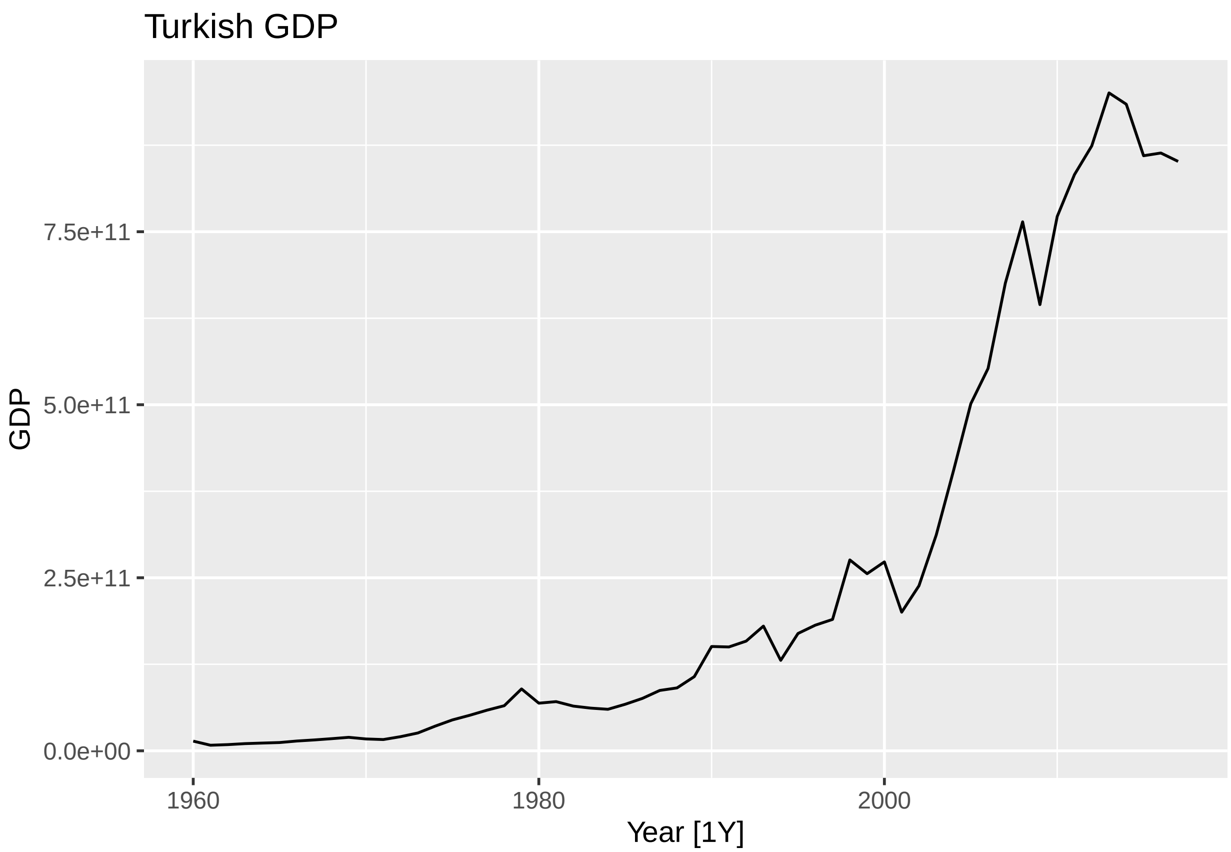
3. For the following series, find an appropriate Box-Cox transformation and order of differencing in order to obtain stationary data.

- a.Turkish GDP from global_economy.

```
head(global_economy)
```

```
## # A tsibble: 6 x 9 [1Y]
## # Key:       Country [1]
##   Country     Code   Year        GDP Growth   CPI Imports Exports Population
##   <fct>       <fct> <dbl>      <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1 Afghanistan AFG    1960  537777811.     NA    NA    7.02    4.13    8996351
## 2 Afghanistan AFG    1961  548888896.     NA    NA    8.10    4.45    9166764
## 3 Afghanistan AFG    1962  546666678.     NA    NA    9.35    4.88    9345868
## 4 Afghanistan AFG    1963  751111191.     NA    NA   16.9     9.17    9533954
## 5 Afghanistan AFG    1964  800000044.     NA    NA   18.1     8.89    9731361
## 6 Afghanistan AFG    1965 1006666638.     NA    NA   21.4    11.3     9938414
```

```
turkish_GDP <- global_economy %>%
  filter(Country == 'Turkey')

turkish_GDP %>%
  autoplot(GDP) +
  labs(title = "Turkish GDP")
```



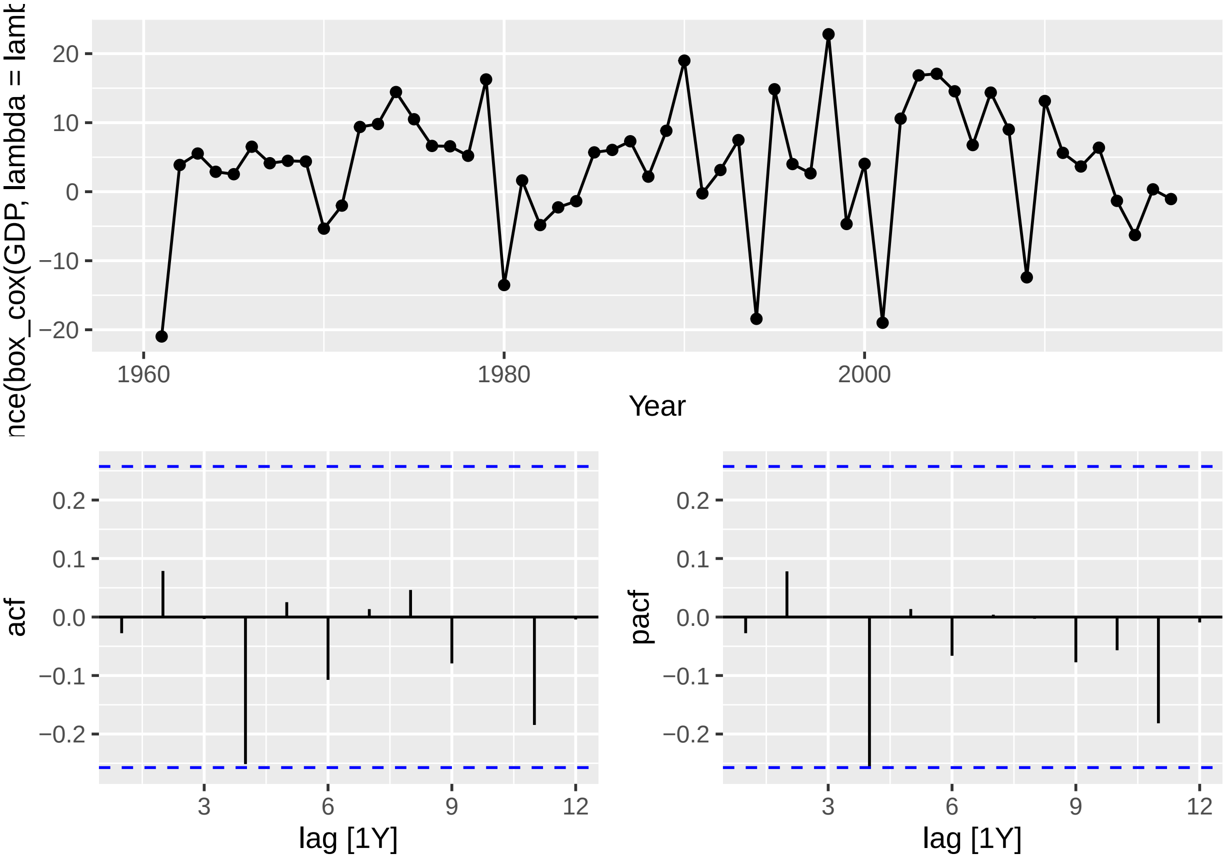
```r
#Box-cox transform
#find lambda value
lambda_turkey <- turkish_GDP %>%
  features(GDP, features = guerrero) %>%
  pull(lambda_guerrero)

#find ndiffs
turkish_GDP  %>%
  mutate(GDP = box_cox(GDP, lambda_turkey)) %>%
  features(GDP, unitroot_ndiffs)
```

```
## # A tibble: 1 x 2
##   Country ndiffs
##   <fct>    <int>
## 1 Turkey       1
```

```r
turkish_GDP %>%
  gg_tsdisplay(difference(box_cox(GDP, lambda = lambda_turkey)),plot_type = 'partial', lag = 12)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

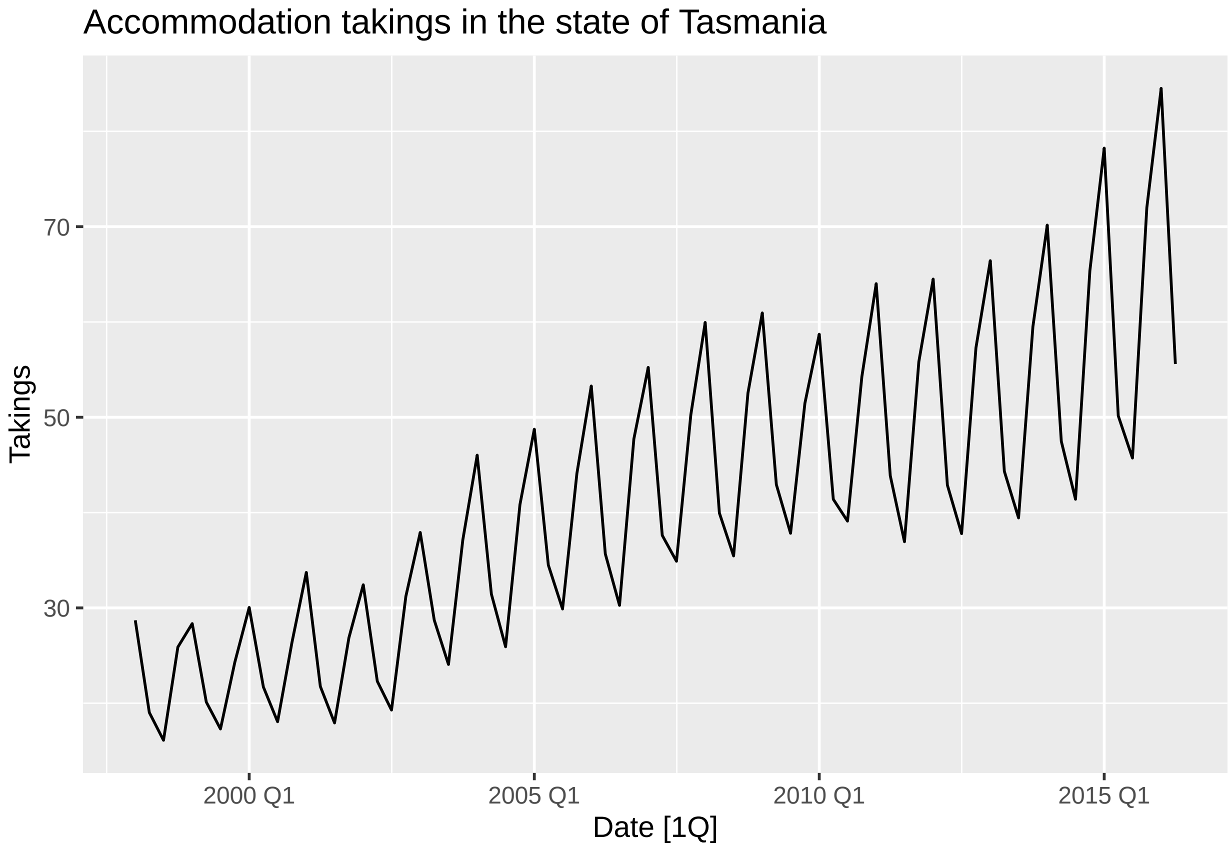


- b.Accommodation takings in the state of Tasmania from aus_accommodation.

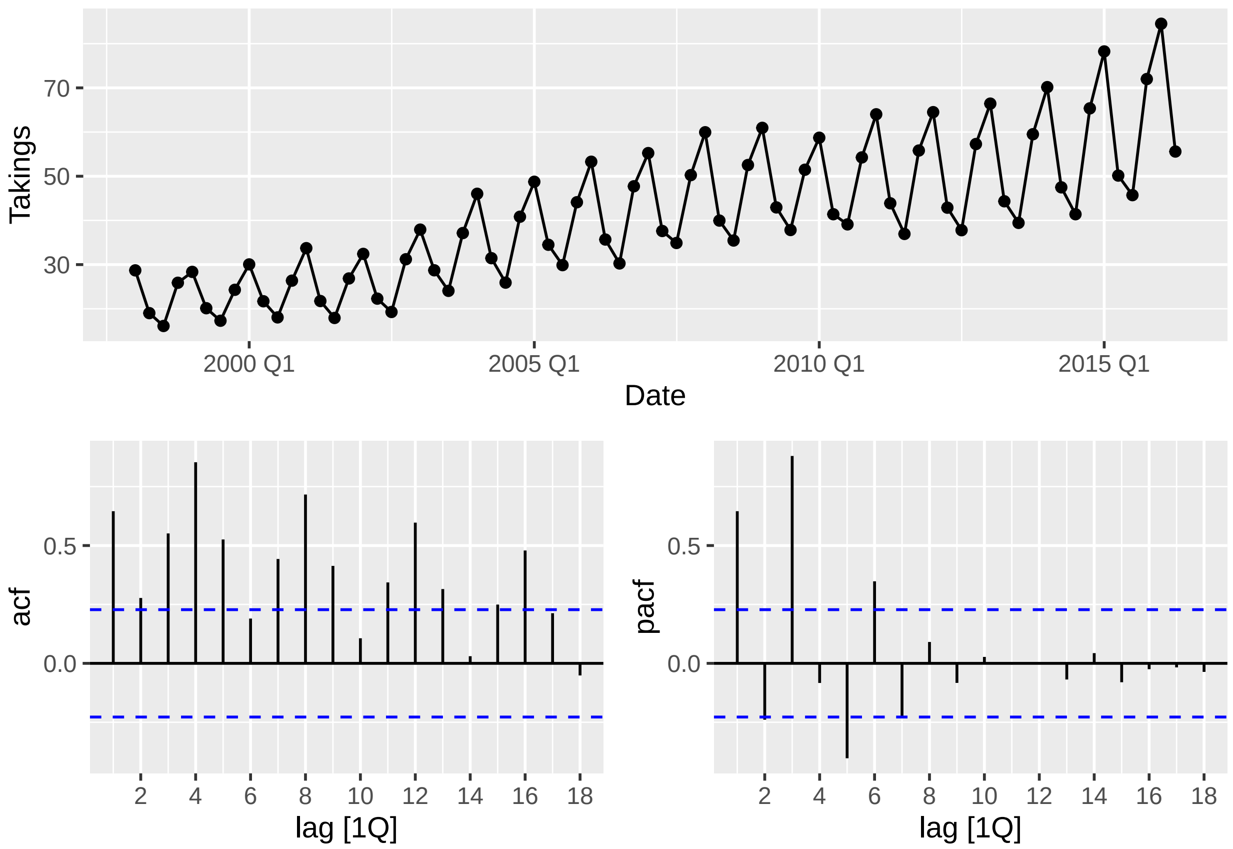
```
head(aus_accommodation)
```

```
## # A tsibble: 6 x 5 [1Q]
## # Key:       State [1]
##      Date State                        Takings Occupancy   CPI
##     <qtr> <chr>                          <dbl>     <dbl> <dbl>
## 1 1998 Q1 Australian Capital Territory    24.3        65  67
## 2 1998 Q2 Australian Capital Territory    22.3        59  67.4
## 3 1998 Q3 Australian Capital Territory    22.5        58  67.5
## 4 1998 Q4 Australian Capital Territory    24.4        59  67.8
## 5 1999 Q1 Australian Capital Territory    23.7        58  67.8
## 6 1999 Q2 Australian Capital Territory    25.4        61  68.1
```

```
tasmania <- aus_accommodation %>%
  filter(State == 'Tasmania')

tasmania %>%
  autoplot(Takings) +
  labs(title = "Accommodation takings in the state of Tasmania")
```



```
tasmania %>%
  gg_tsdisplay(Takings, plot_type = 'partial')
```

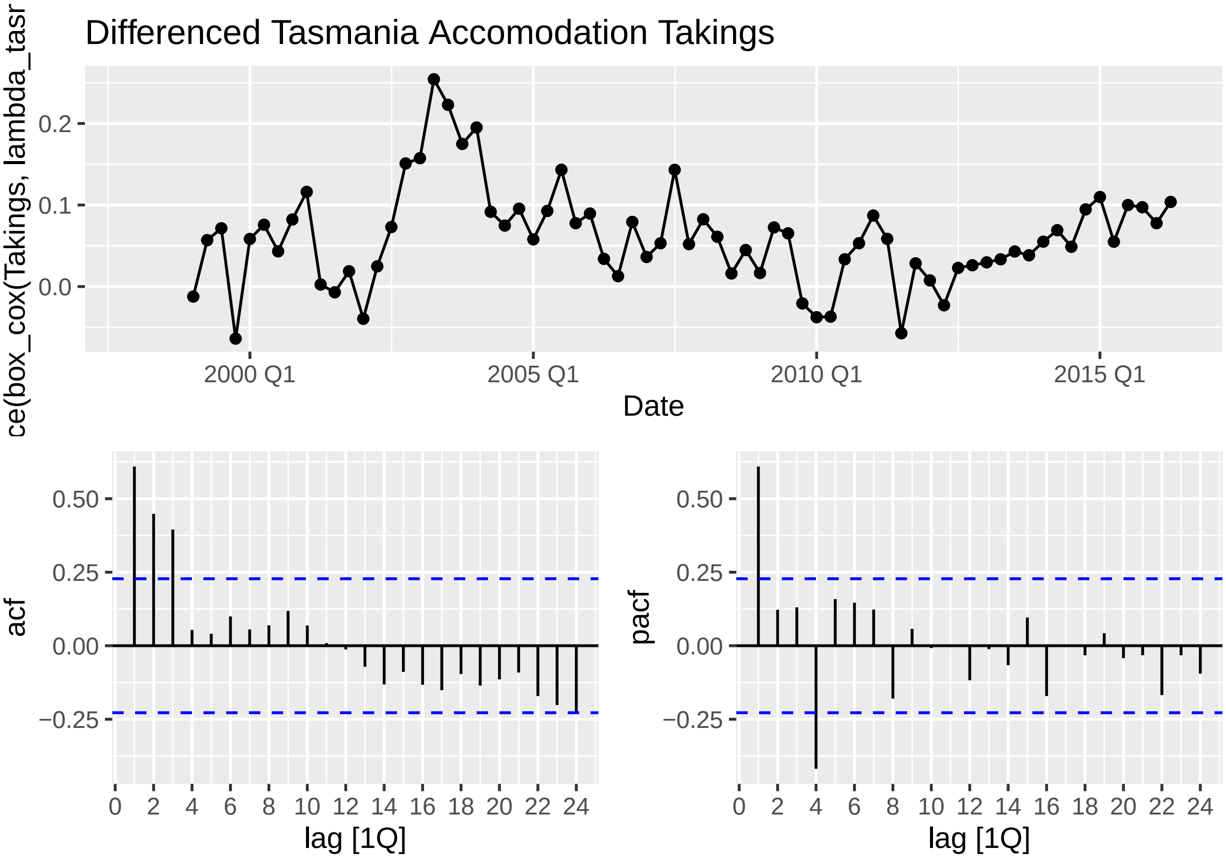


```r
#Box-cox transform
#find lambda value
lambda_tasmania <- tasmania %>%
  features(Takings, features = guerrero) %>%
  pull(lambda_guerrero)

#find ndiffs
tasmania  %>%
  mutate(Takings = box_cox(Takings, lambda_tasmania)) %>%
  features(Takings, unitroot_ndiffs)
```

```
## # A tibble: 1 x 2
##   State    ndiffs
##   <chr>     <int>
## 1 Tasmania      1
```

```r
tasmania %>%
  gg_tsdisplay(difference(box_cox(Takings, lambda_tasmania), 4),plot_type = 'partial', lag = 24) +
  labs(title="Differenced Tasmania Accomodation Takings")
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

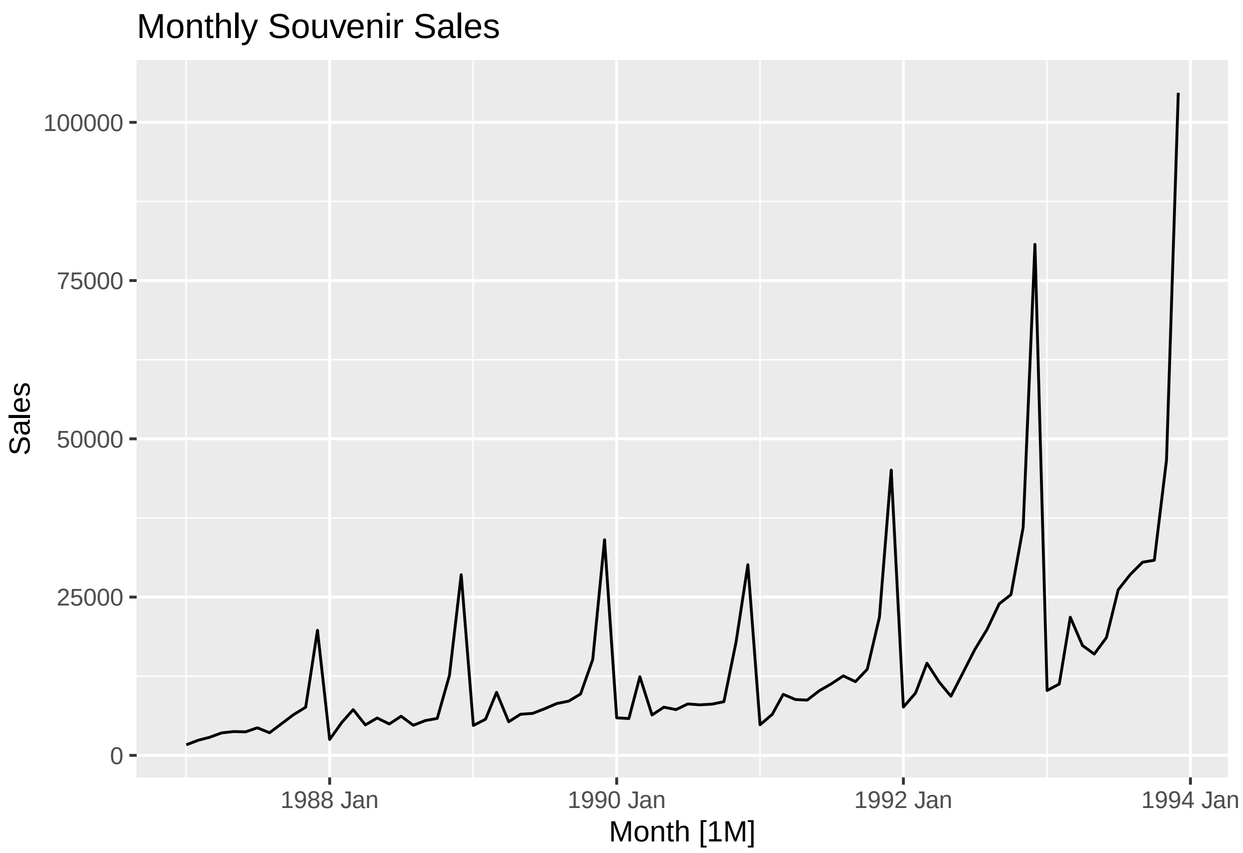


- c.Monthly sales from souvenirs.

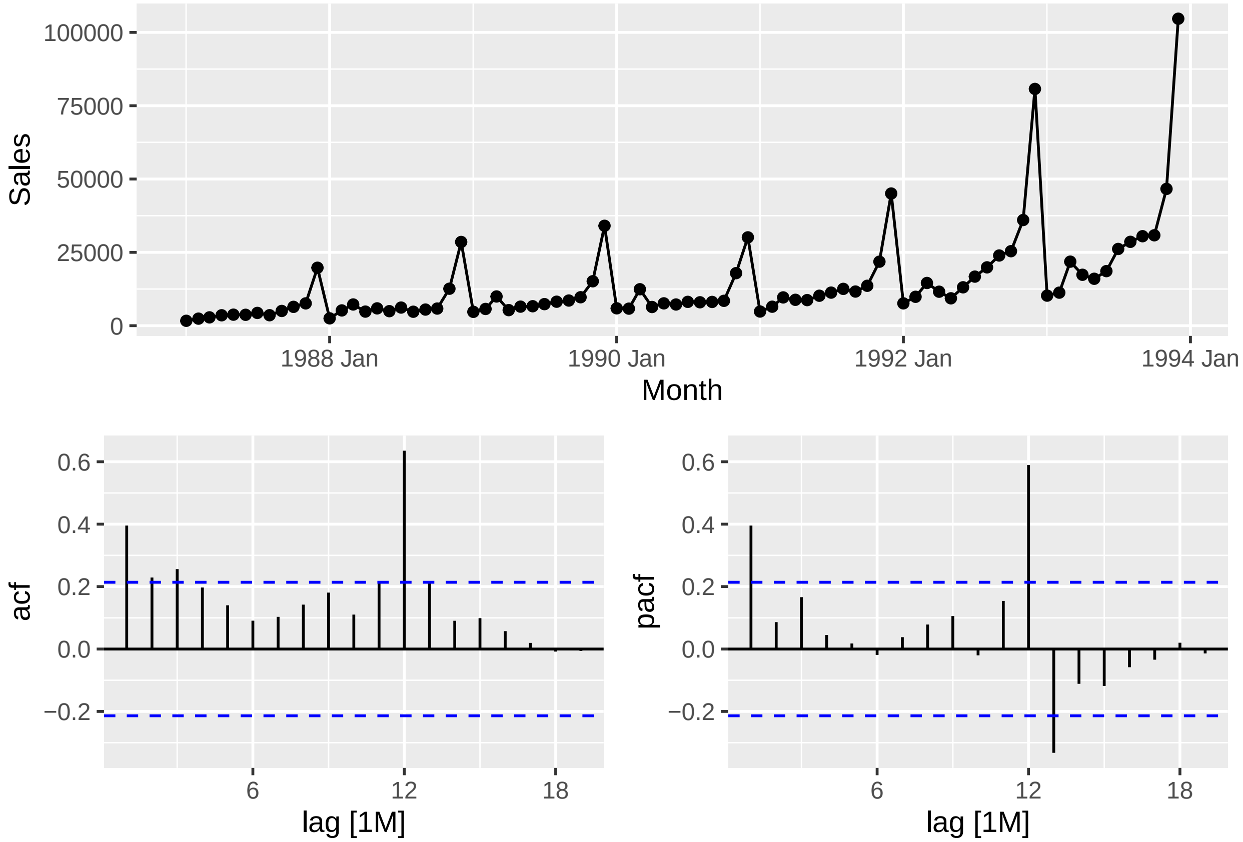
```
head(souvenirs)
```

```
## # A tsibble: 6 x 2 [1M]
##      Month Sales
##      <mth> <dbl>
## 1 1987 Jan 1665.
## 2 1987 Feb 2398.
## 3 1987 Mar 2841.
## 4 1987 Apr 3547.
## 5 1987 May 3753.
## 6 1987 Jun 3715.
```

```
souvenirs %>%
  autoplot(Sales) +
  labs(title = "Monthly Souvenir Sales")
```



```
souvenirs %>%
  gg_tsdisplay(Sales, plot_type = 'partial')
```



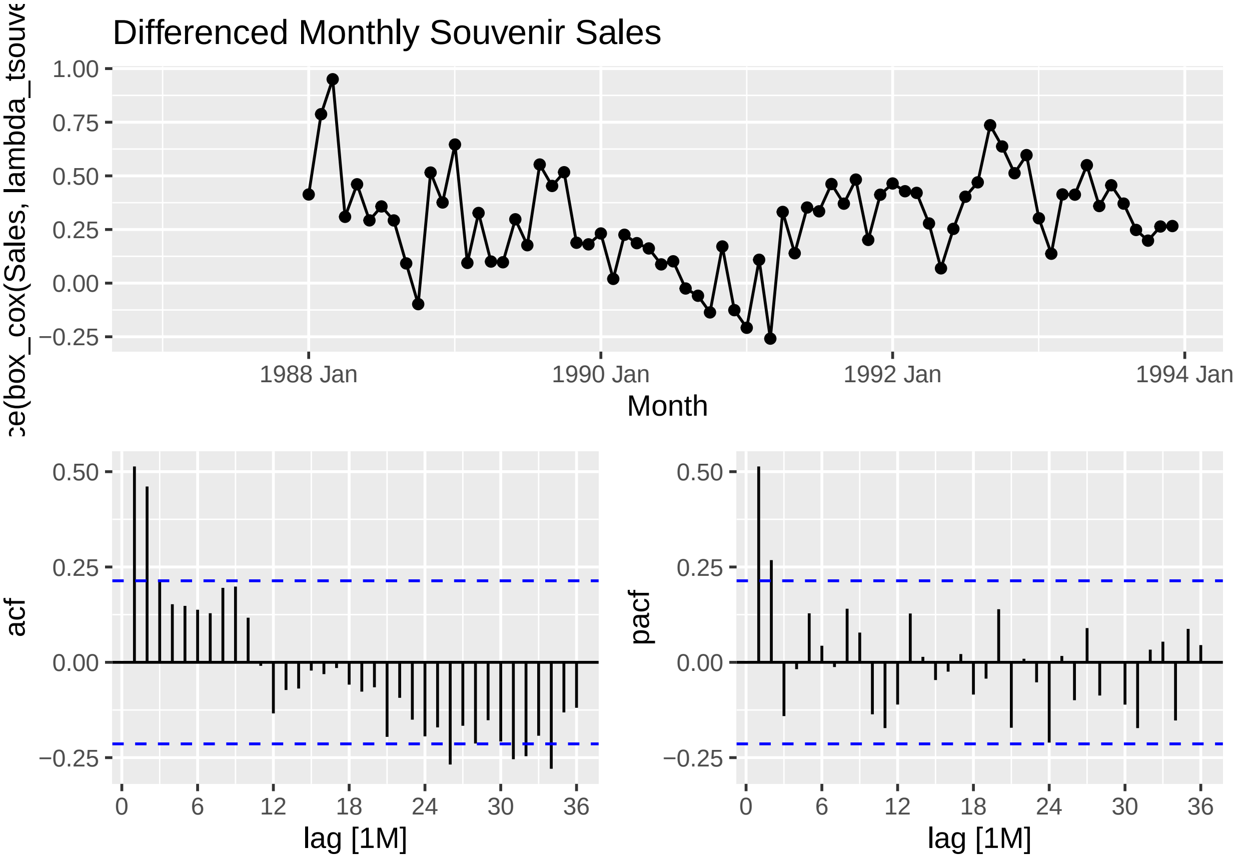
```r
#Box-cox transform
#find lambda value
lambda_tsouvenirs <- souvenirs %>%
  features(Sales, features = guerrero) %>%
  pull(lambda_guerrero)

#find ndiffs
souvenirs  %>%
  mutate(Sales = box_cox(Sales, lambda_tasmania)) %>%
  features(Sales, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

```r
souvenirs %>%
  gg_tsdisplay(difference(box_cox(Sales, lambda_tsouvenirs), 12),plot_type = 'partial', lag = 36) +
  labs(title="Differenced Monthly Souvenir Sales")
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

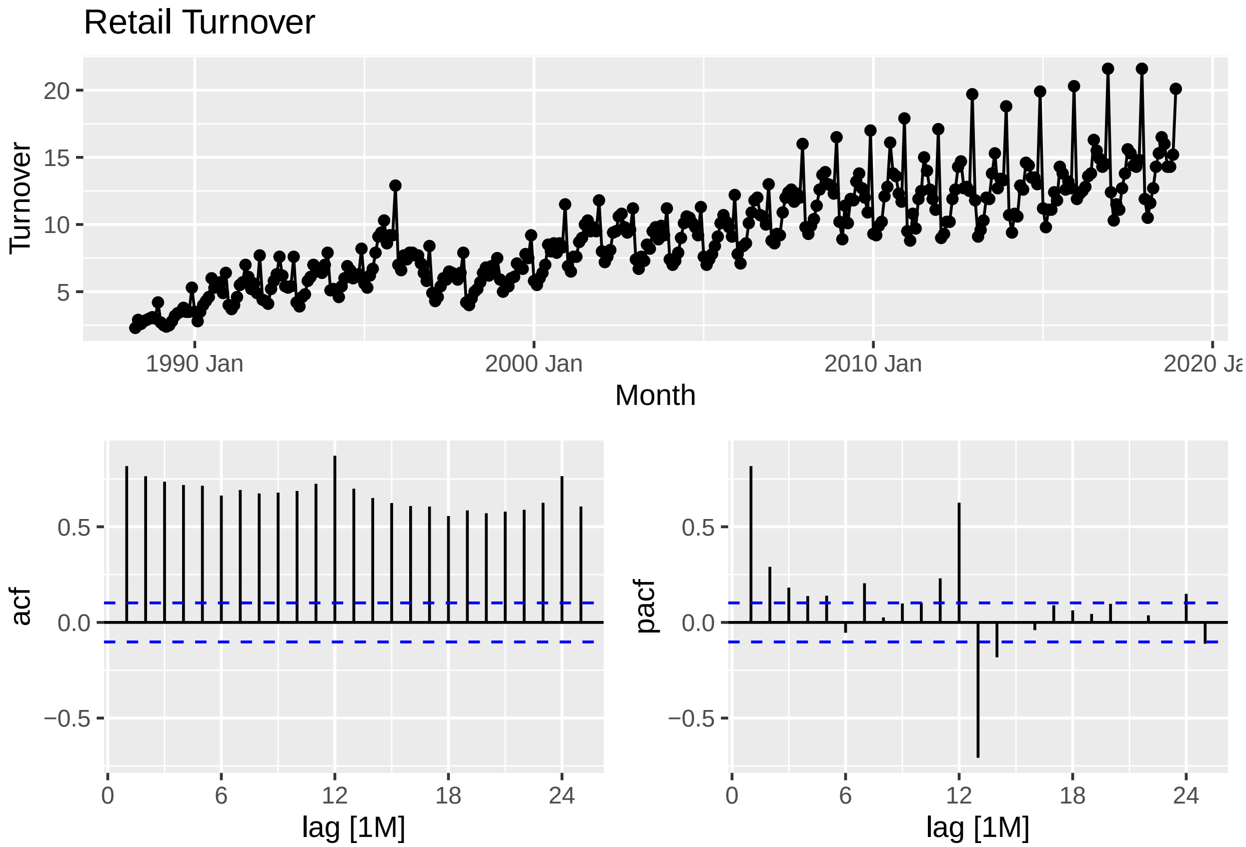


5.For your retail data (from Exercise 7 in Section 2.10), find the appropriate order of differencing (after transformation if necessary) to obtain stationary data.

```
set.seed(12345678)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

- The plots shows no pastern, many spikes in acf plots, and now we have use box-cox to transfer.

```
myseries %>%
  gg_tsdisplay(Turnover, plot_type = 'partial')+
  labs (title = "Retail Turnover")
```



- box- cox transfer the data

```
#find lambad
lambad_myseries <- myseries %>%
  features(Turnover, features = guerrero) %>%
  pull (lambda_guerrero)

#find ndiffs
myseries  %>%
  mutate(Turnover = box_cox(Turnover, lambad_myseries)) %>%
  features(Turnover, unitroot_ndiffs)
```

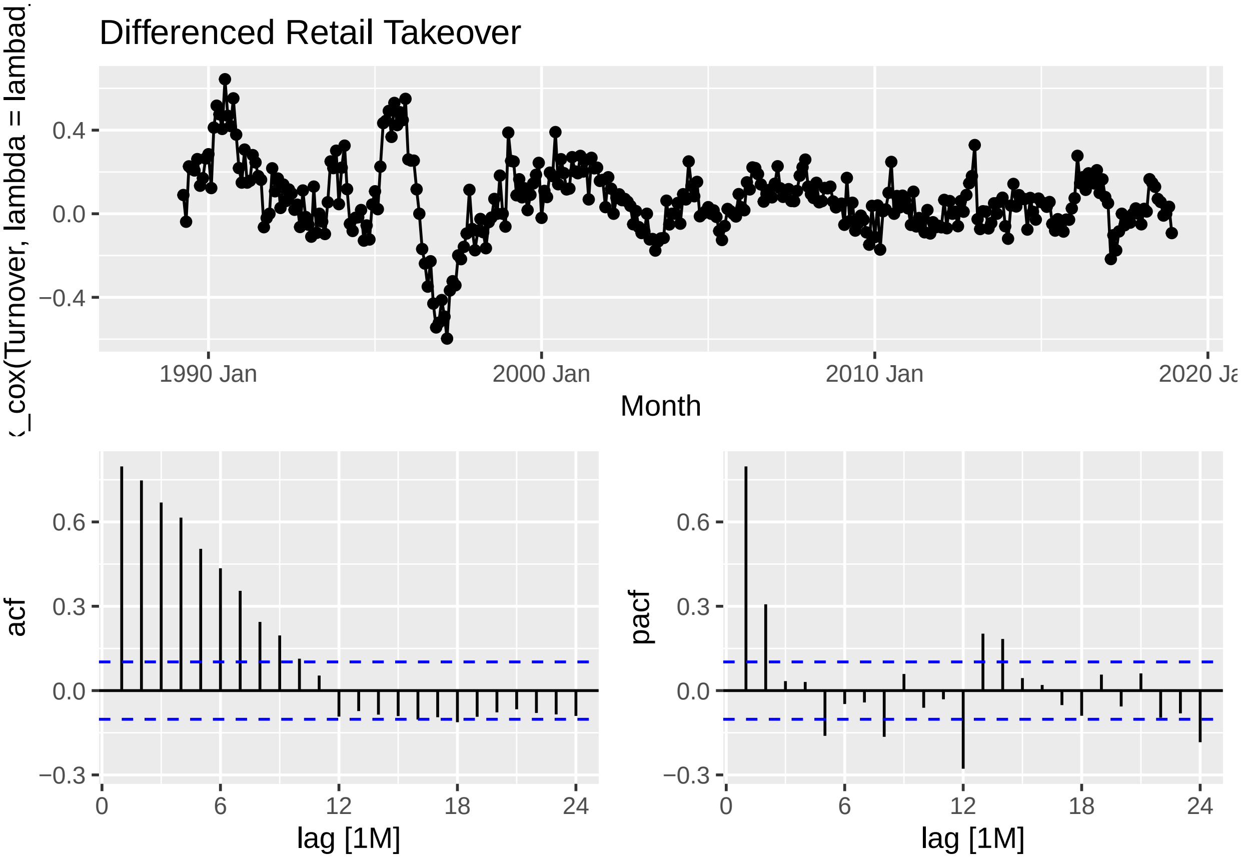
```
## # A tibble: 1 x 3
##   State              Industry                                             ndiffs
##   <chr>              <chr>                                                 <int>
## 1 Northern Territory Clothing, footwear and personal accessory retailing      1
```

- We can see in the acf still have many spikes, and clearly non- stationary, so we need to use double differenced.

```
myseries %>%
  gg_tsdisplay(difference(box_cox(Turnover,lambda = lambad_myseries), 12), plot_type='partial', lag = 2
  labs(title = paste("Differenced Retail Takeover"))
```

```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

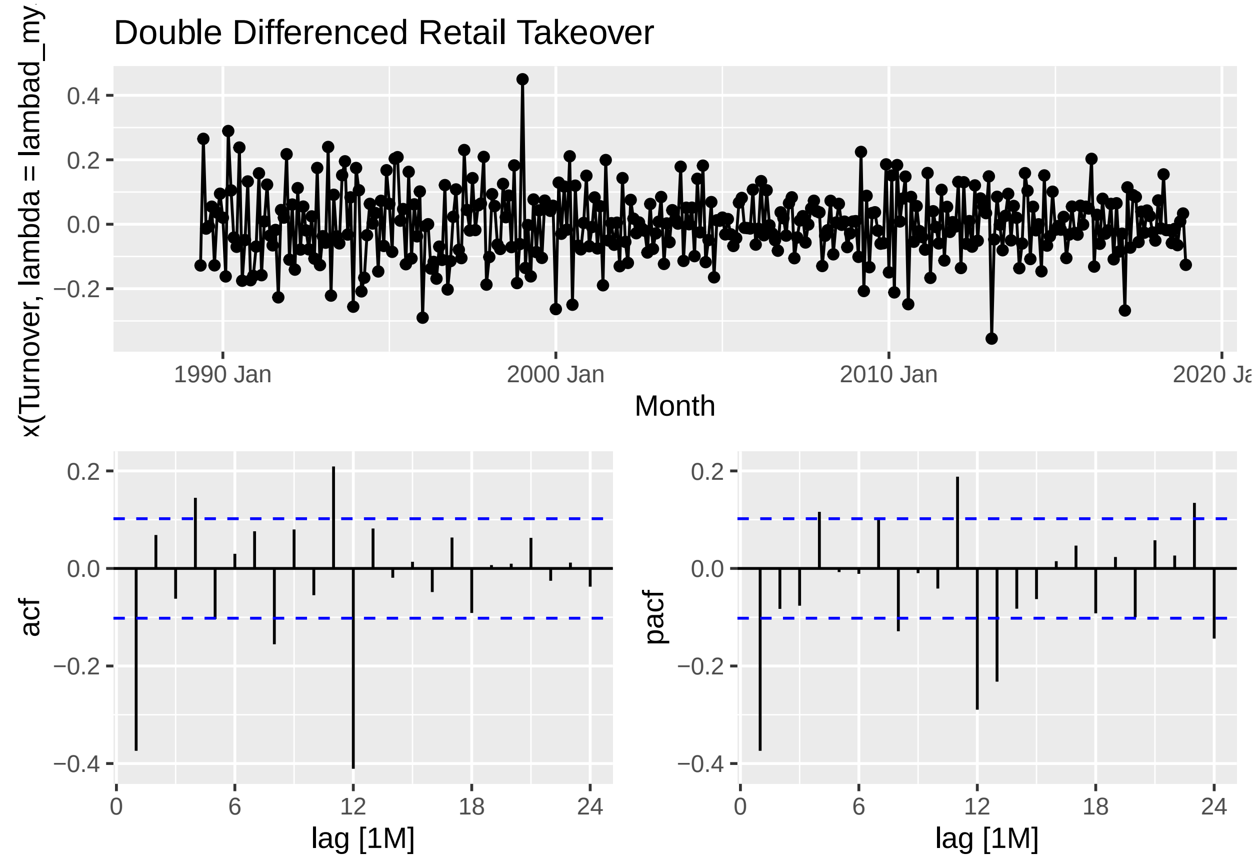
```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
# Double differenced
myseries %>%
  gg_tsdisplay(difference(box_cox(Turnover,lambda = lambad_myseries), 12) %>% difference(), plot_type='
  labs(title = paste("Double Differenced Retail Takeover"))
```

```
## Warning: Removed 13 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 13 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



6.Simulate and plot some data from simple ARIMA models.

- a.Use the following R code to generate data from an AR(1) model with 1=0.6 and 2=1. The process starts with y1=0.

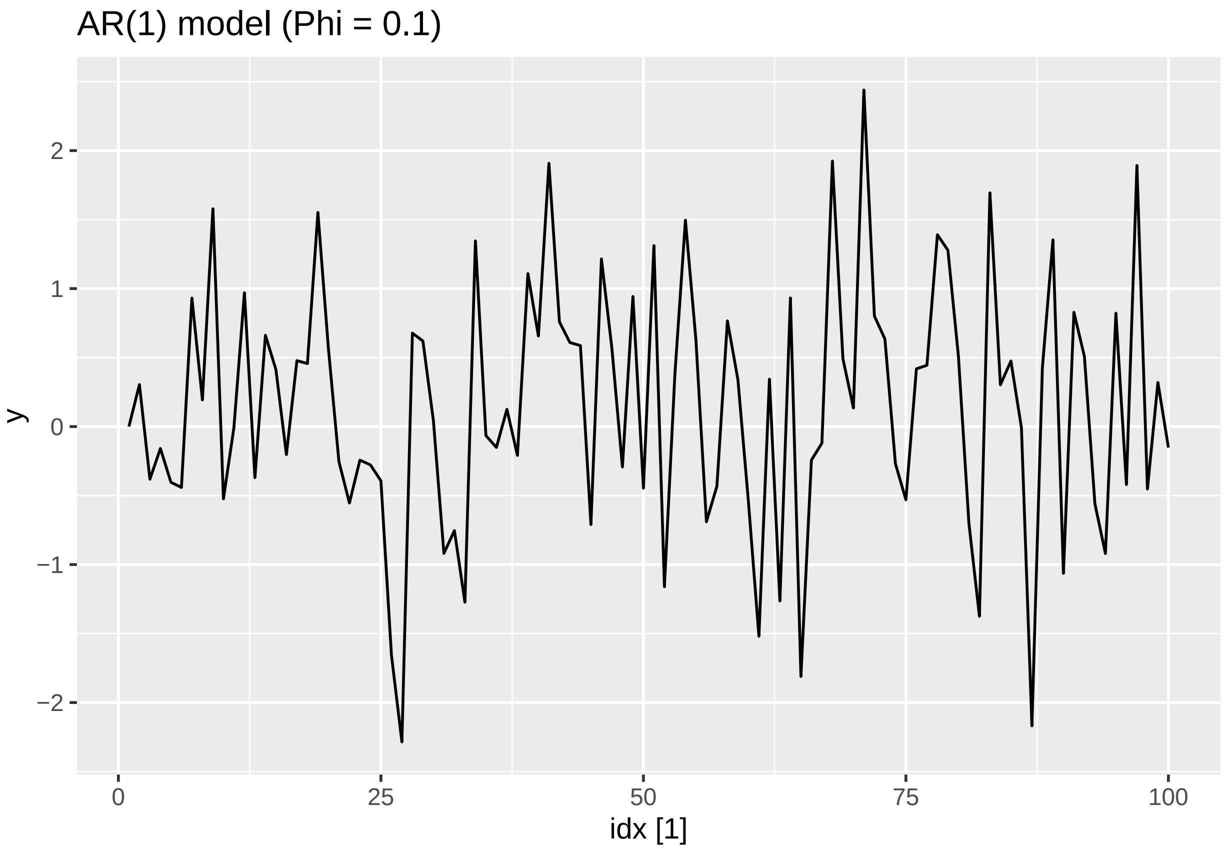
```
y <- numeric(100)

e <- rnorm(100)

for(i in 2:100)
  y[i] <- 0.6*y[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

- b.Produce a time plot for the series. How does the plot change as you change 1?
- Answer: As 1 increases or decreases, the pattern of the time series changes. Specifically, when 1 is positive and closer to 1, the series shows smoother and more persistent trends, with longer "waves". When 1 is negative, the series tends is rapidly, resulting in shorter wavelengths and more frequent changes in direction.

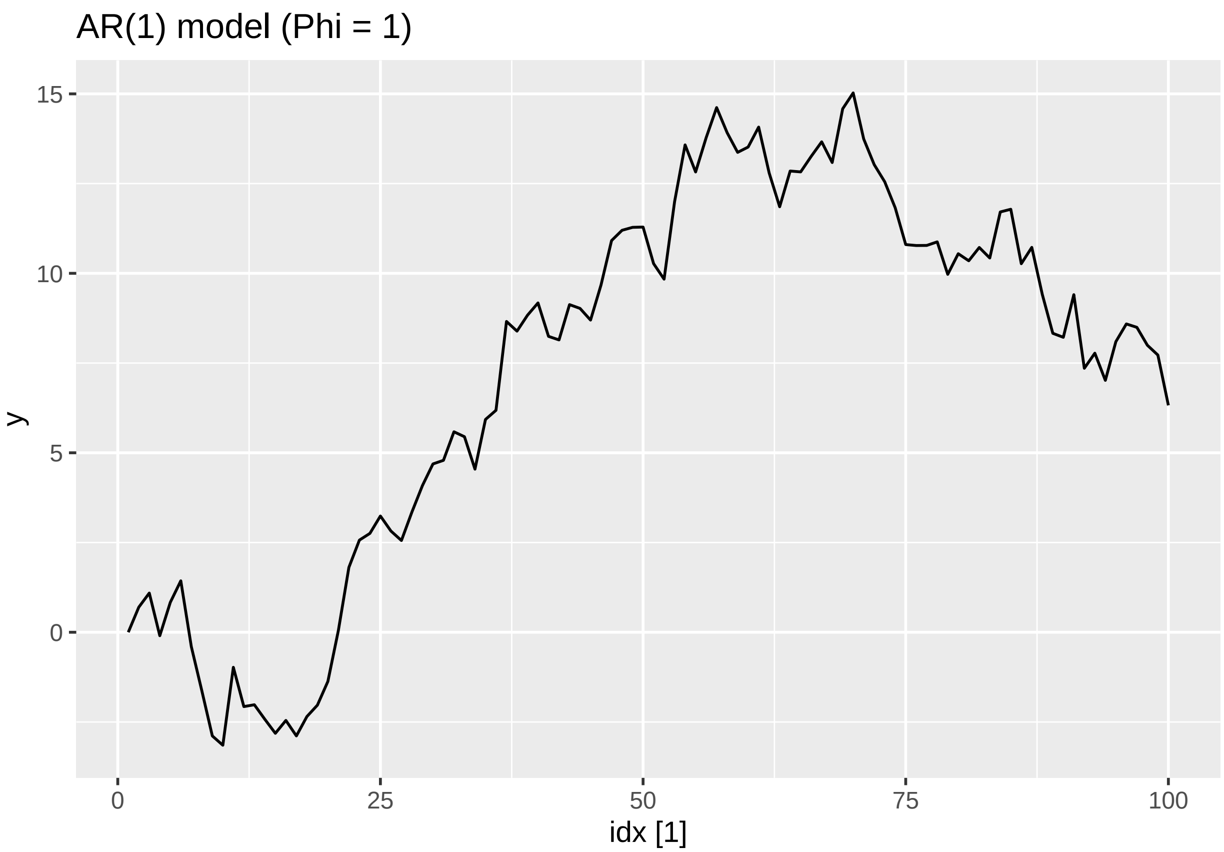
```
y <- numeric(100)

e <- rnorm(100)
```

```
for(i in 2:100)
  y[i] <- 0.1*y[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim %>%
  autoplot(y) +
  labs(title = "AR(1) model (Phi = 0.1)")
```



```
y <- numeric(100)

e <- rnorm(100)

for(i in 2:100)
  y[i] <- 1*y[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)

sim %>%
  autoplot(y) +
  labs(title = "AR(1) model (Phi = 1)")
```

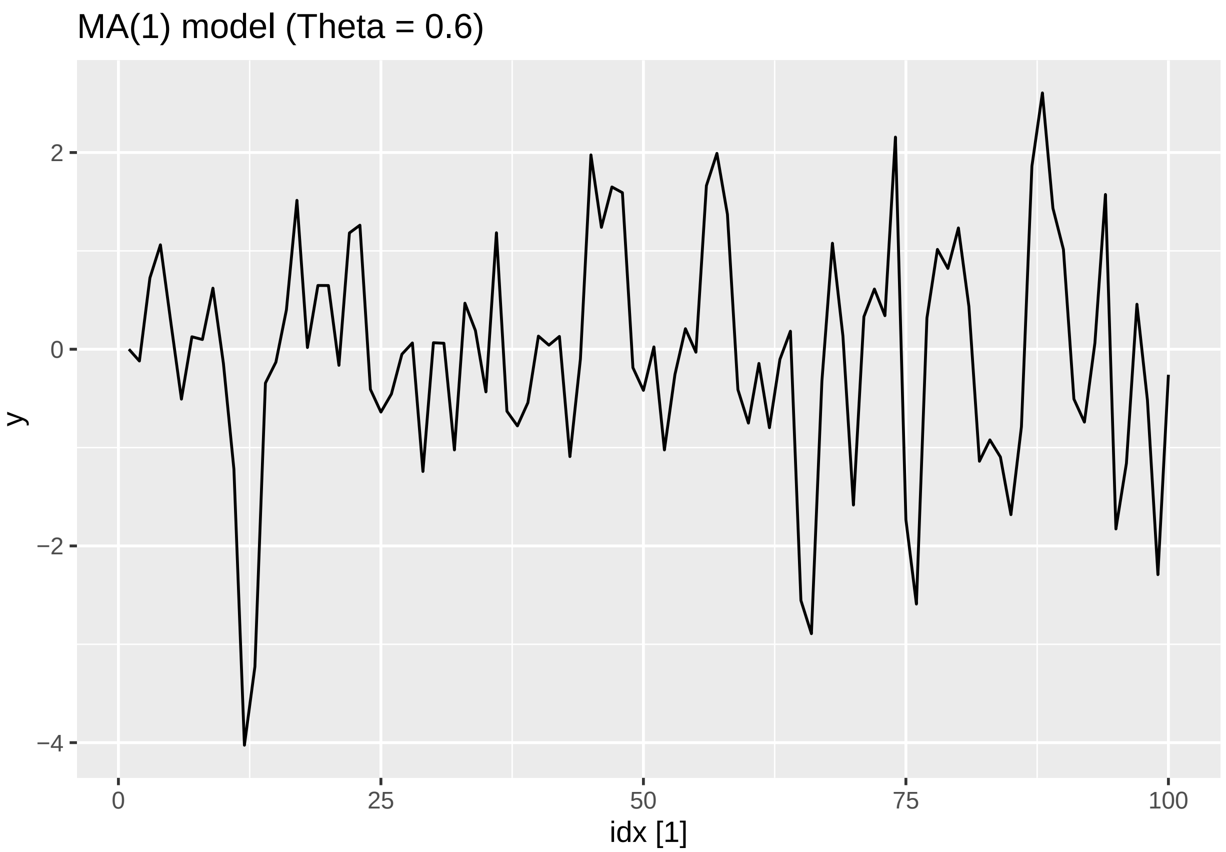


- c.Write your own code to generate data from an MA(1) model with 1=0.6 and 2=1.

```
y <- numeric(100)

e <- rnorm(100)

for(i in 2:100)
  y[i] <- 0.6*e[i-1] + e[i]
sim_ma <- tsibble(idx = seq_len(100), y = y, index = idx)
```

- d.Produce a time plot for the series. How does the plot change as you change 1?
- Answer: The idea is the same as Phi= 0.6.

```
sim_ma %>%
  autoplot(y) +
  labs(title = "MA(1) model (Theta = 0.6)")
```



- e.Generate data from an ARMA(1,1) model with 1=0.6, 1=0.6 and 2=1.

```
y <- numeric(100)

e <- rnorm(100)

for (i in 2:100){
    y[i] <- (0.6 * y[i-1]) + (0.6 * e[i-1]) + e[i]
}

sim1 <- tsibble(idx = seq_len(100), y = y, index = idx)
```

- f.Generate data from an AR(2) model with 1=−0.8, 2=0.3 and 2=1. (Note that these parameters will give a non-stationary series.)

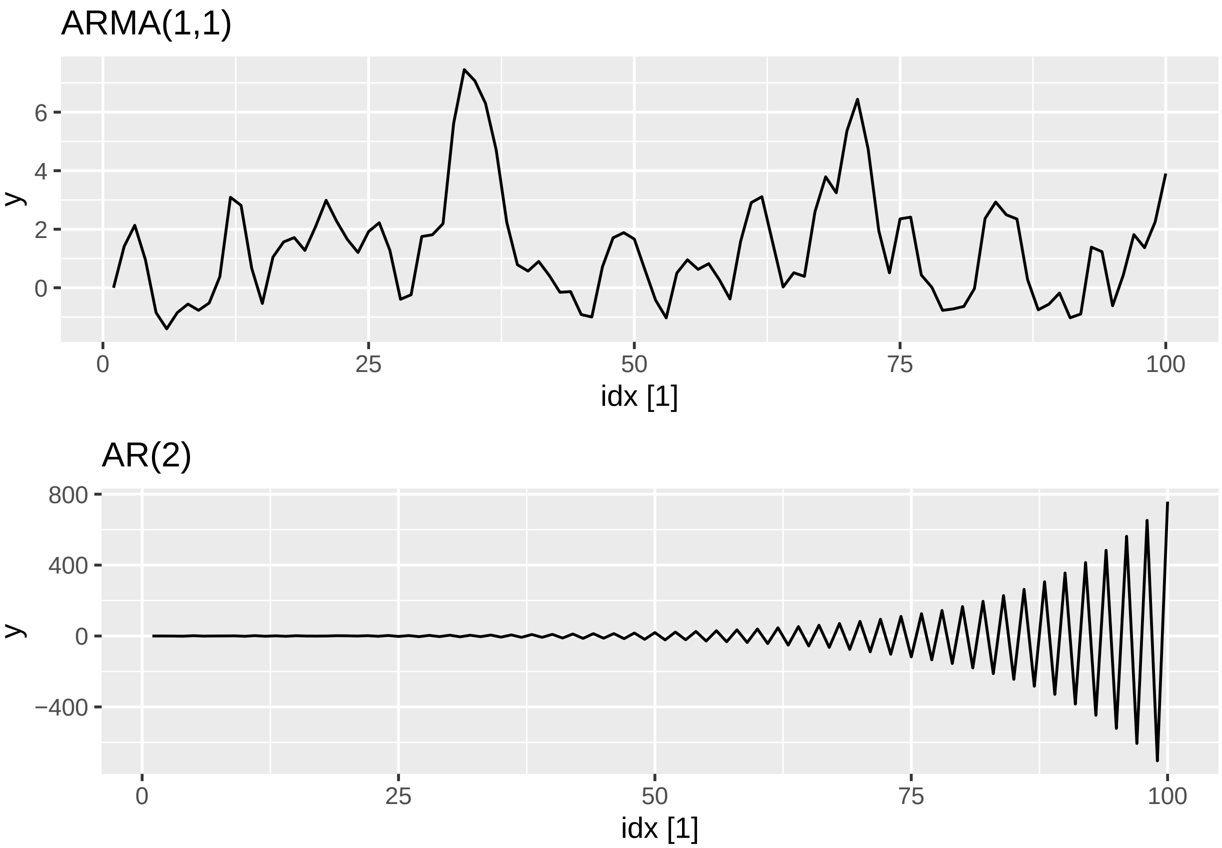
```
y <- numeric(100)

e <- rnorm(100)

for (i in 3:100){
    y[i] <- (-0.8 * y[i-1]) + (0.3 * y[i-2]) + e[i]
}

sim2 <- tsibble(idx = seq_len(100), y = y, index = idx)
```

- g.Graph the latter two series and compare them.

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
plot1 <- sim1 %>%
  autoplot(y) +
  labs(title = "ARMA(1,1)")

plot2 <- sim2 %>%
  autoplot(y) +
  labs(title = "AR(2)")

plot_grid(plot1, plot2, ncol=1)
```



- AR(2) appears increases over time.

7.Consider aus_airpassengers, the total number of passengers (in millions) from Australian air carriers for the period 1970-2011.

- a.Use ARIMA() to find an appropriate ARIMA model. What model was selected. Check that the residuals look like white noise. Plot forecasts for the next 10 periods.
- Answer: ARIMA(0,2,1) has been select.

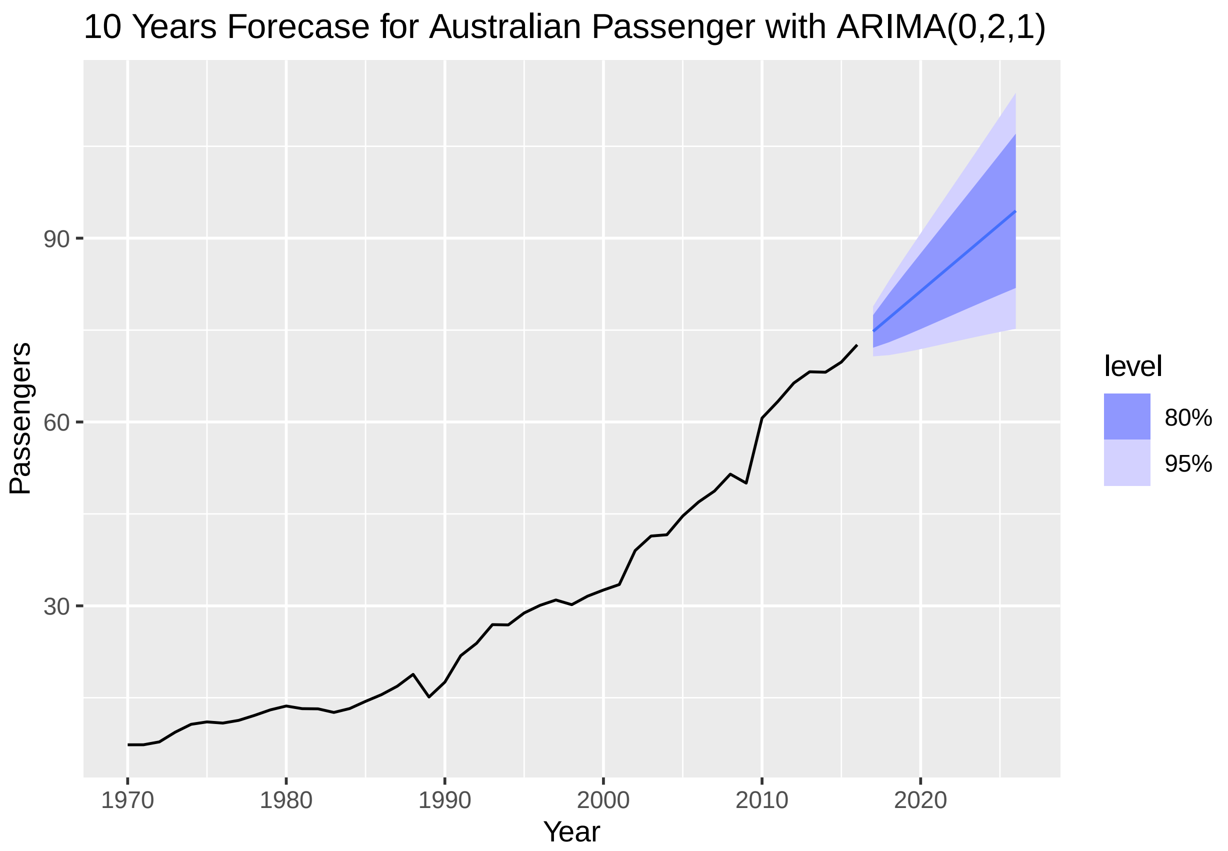
```
head(aus_airpassengers)
```

```
## # A tsibble: 6 x 2 [1Y]
##    Year Passengers
##   <dbl>      <dbl>
## 1  1970       7.32
## 2  1971       7.33
## 3  1972       7.80
## 4  1973       9.38
## 5  1974      10.7
## 6  1975      11.1
```

```
fit <- aus_airpassengers |>
  model(ARIMA(Passengers))

report(fit)
```

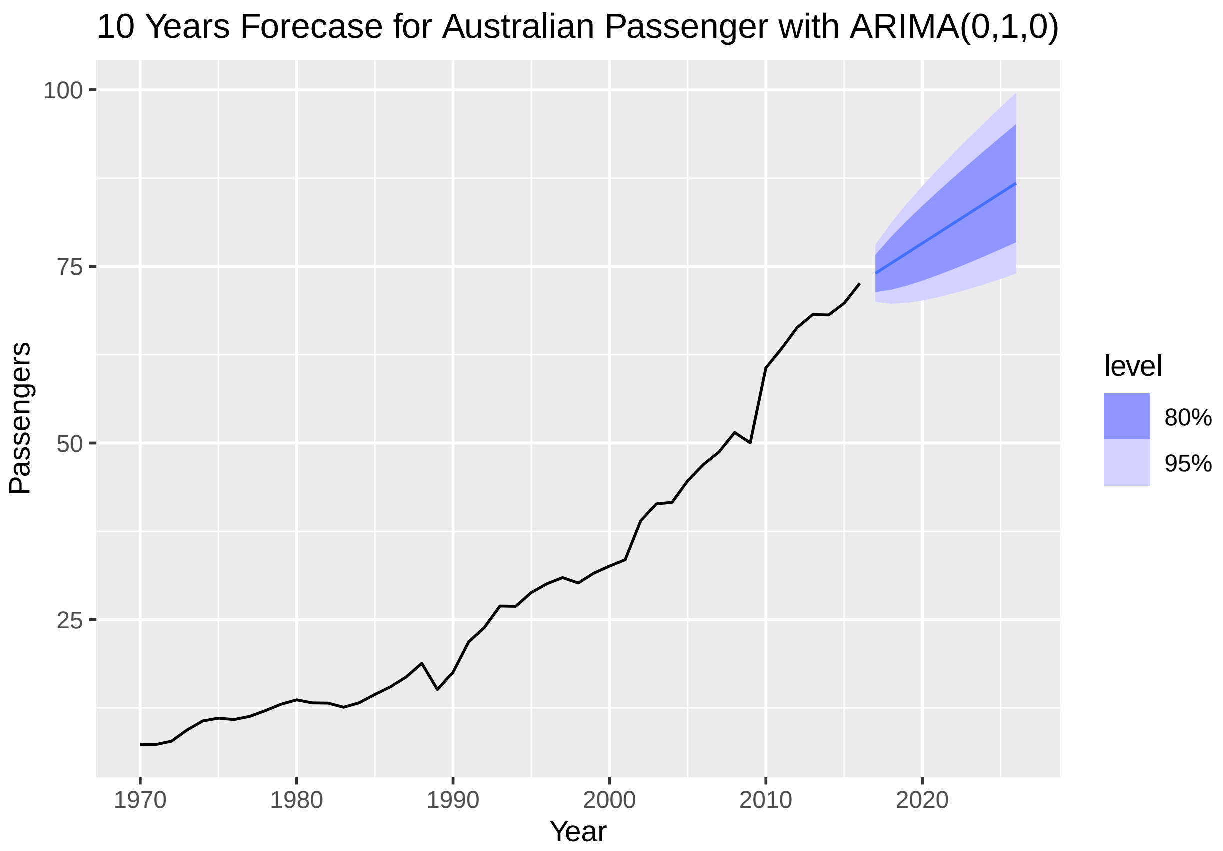
```
## Series: Passengers
## Model: ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.8963
## s.e.   0.0594
##
## sigma^2 estimated as 4.308:  log likelihood=-97.02
## AIC=198.04   AICc=198.32   BIC=201.65
```

```
# Plot for forecast for 10 years
fit %>%
  forecast(h="10 years") %>%
  autoplot(aus_airpassengers) +
  labs(title = "10 Years Forecase for Australian Passenger with ARIMA(0,2,1)")
```



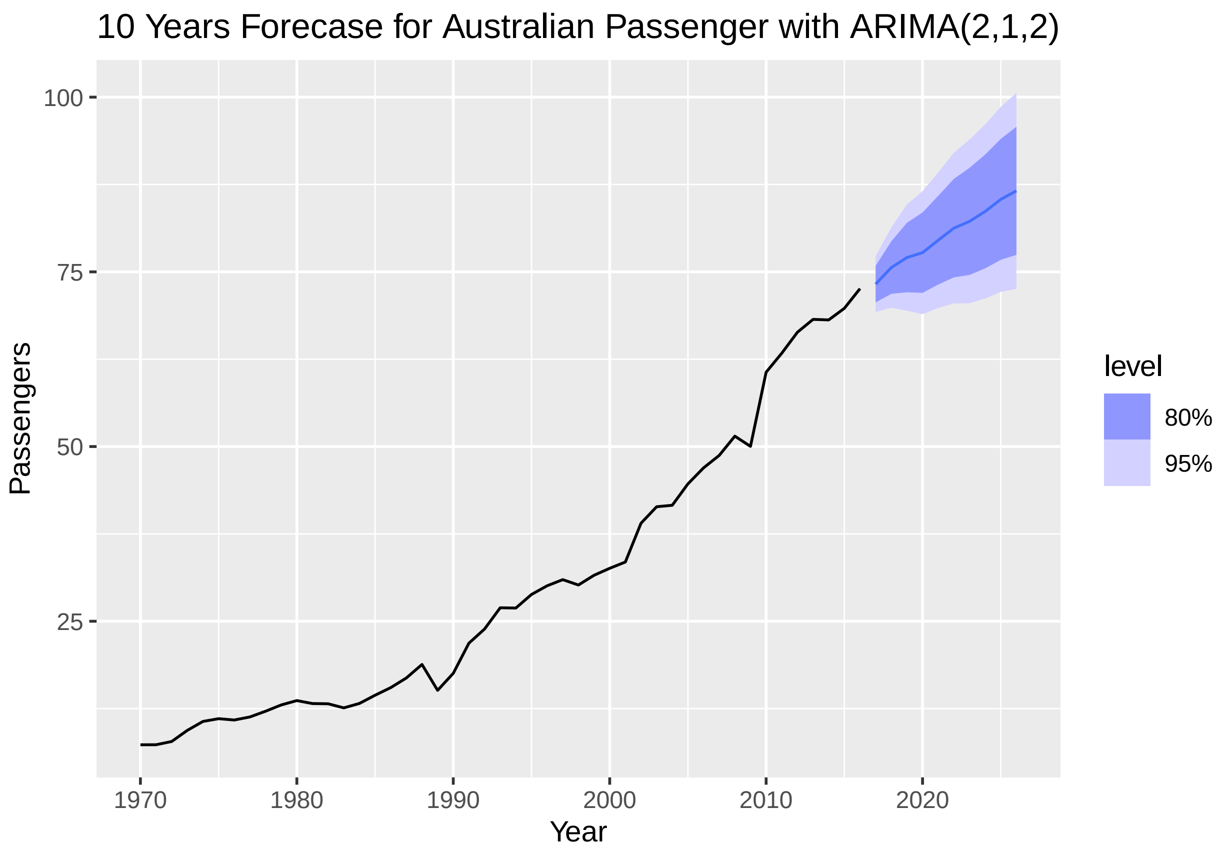
- b.Write the model in terms of the backshift operator.
- Answer: (1−B)^2yt=(1+ 1B)) t
- c.Plot forecasts from an ARIMA(0,1,0) model with drift and compare these to part a.
- Answer: Can’t really see the difference, the two forecasts are very similar with an increasing trend over the time.

```r
fit2 <-aus_airpassengers %>%
  model(ARIMA(Passengers ~ 1 + pdq(0,1,0)))

fit2 %>%
  forecast(h=10) %>%
  autoplot(aus_airpassengers) +
  labs(title = "10 Years Forecase for Australian Passenger with ARIMA(0,1,0)")
```



- d.Plot forecasts from an ARIMA(2,1,2) model with drift and compare these to parts a and c. Remove the constant and see what happens.
- Answer: The result is similar. ARIMA(2,1,2) model can see the "wave" at the forecast part, ARIMA(2,1,2) is unable to forecast the non-stationary data.

```
fit3 <-aus_airpassengers %>%
  model(ARIMA(Passengers ~ 1 + pdq(2,1,2)))

fit3 %>%
  forecast(h=10) %>%
  autoplot(aus_airpassengers) +
  labs(title = "10 Years Forecase for Australian Passenger with ARIMA(2,1,2)")
```

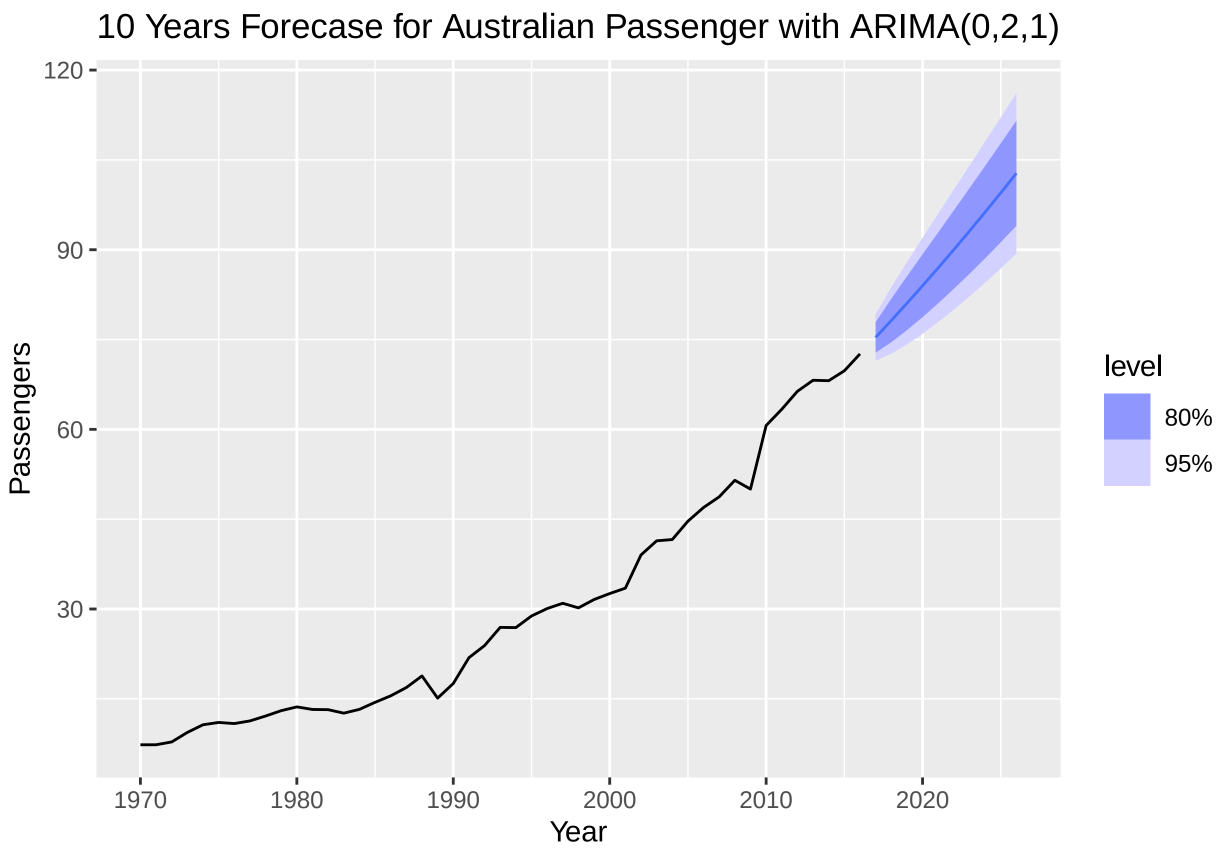


- e.Plot forecasts from an ARIMA(0,2,1) model with a constant. What happens?
- Answer: This model forecast is with a drastic increase. This is generally discouraged.

```
fit4 <-aus_airpassengers %>%
  model(ARIMA(Passengers ~ 1 + pdq(0,2,1)))
```

```
## Warning: Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
```

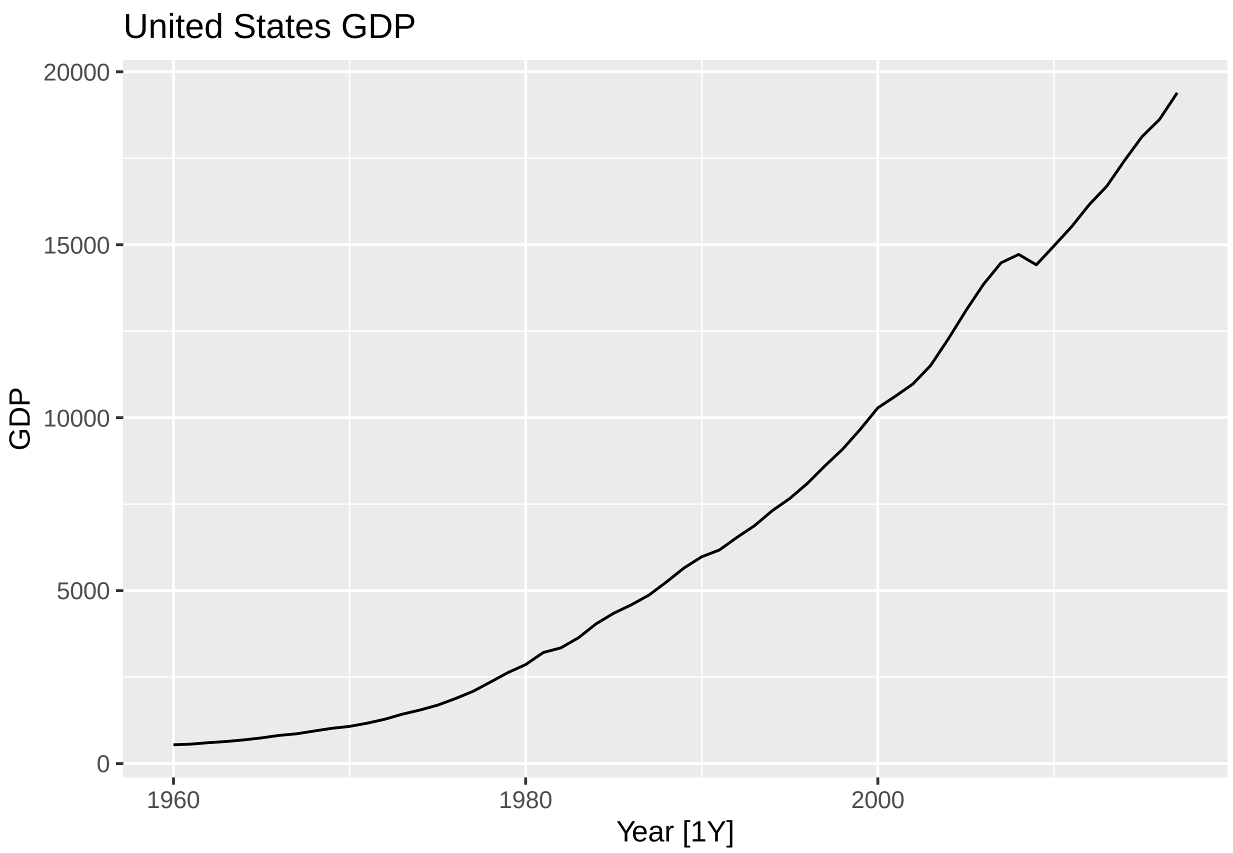
```
fit4 %>%
  forecast(h=10) %>%
  autoplot(aus_airpassengers) +
  labs(title = "10 Years Forecase for Australian Passenger with ARIMA(0,2,1)")
```



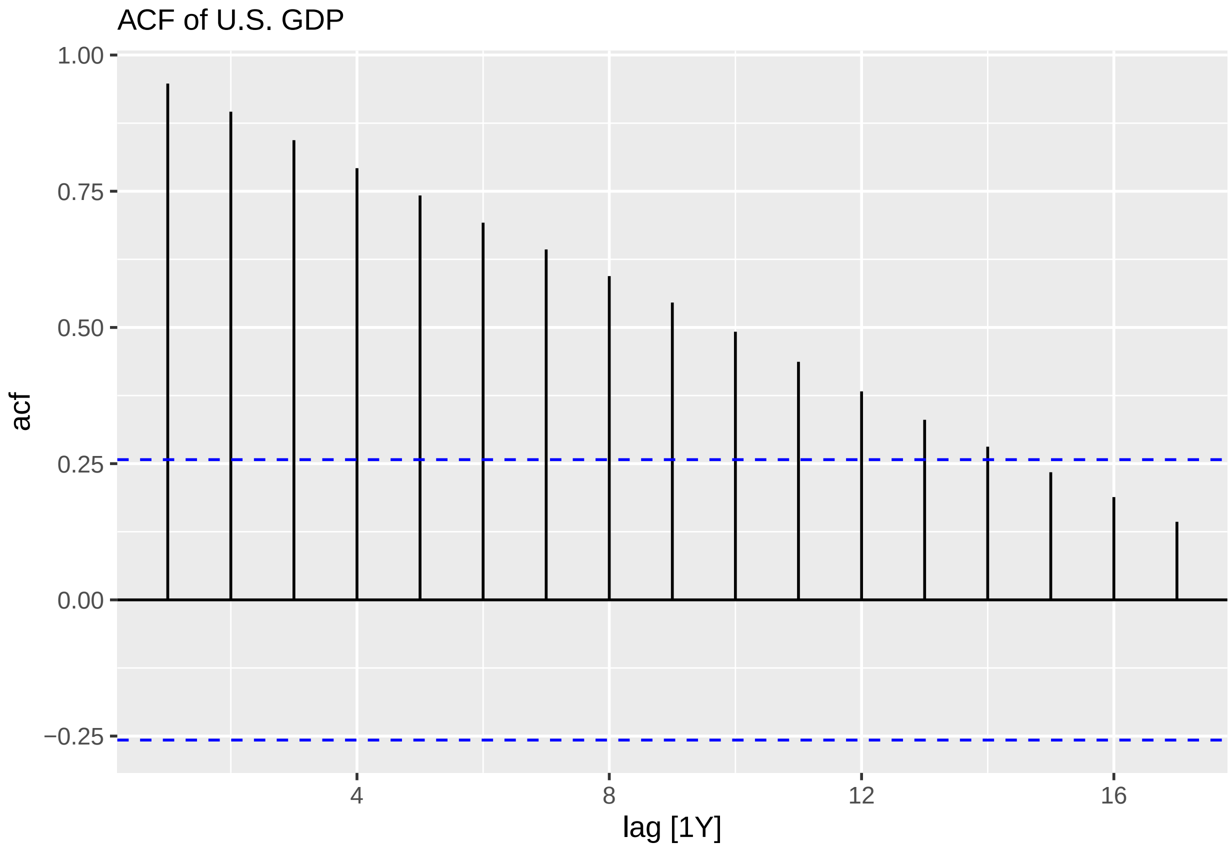
8.For the United States GDP series (from global_economy):

- a.if necessary, find a suitable Box-Cox transformation for the data;
- Answer: The P- value is 0 for this model, I think we need to use box-cox transformation the data.

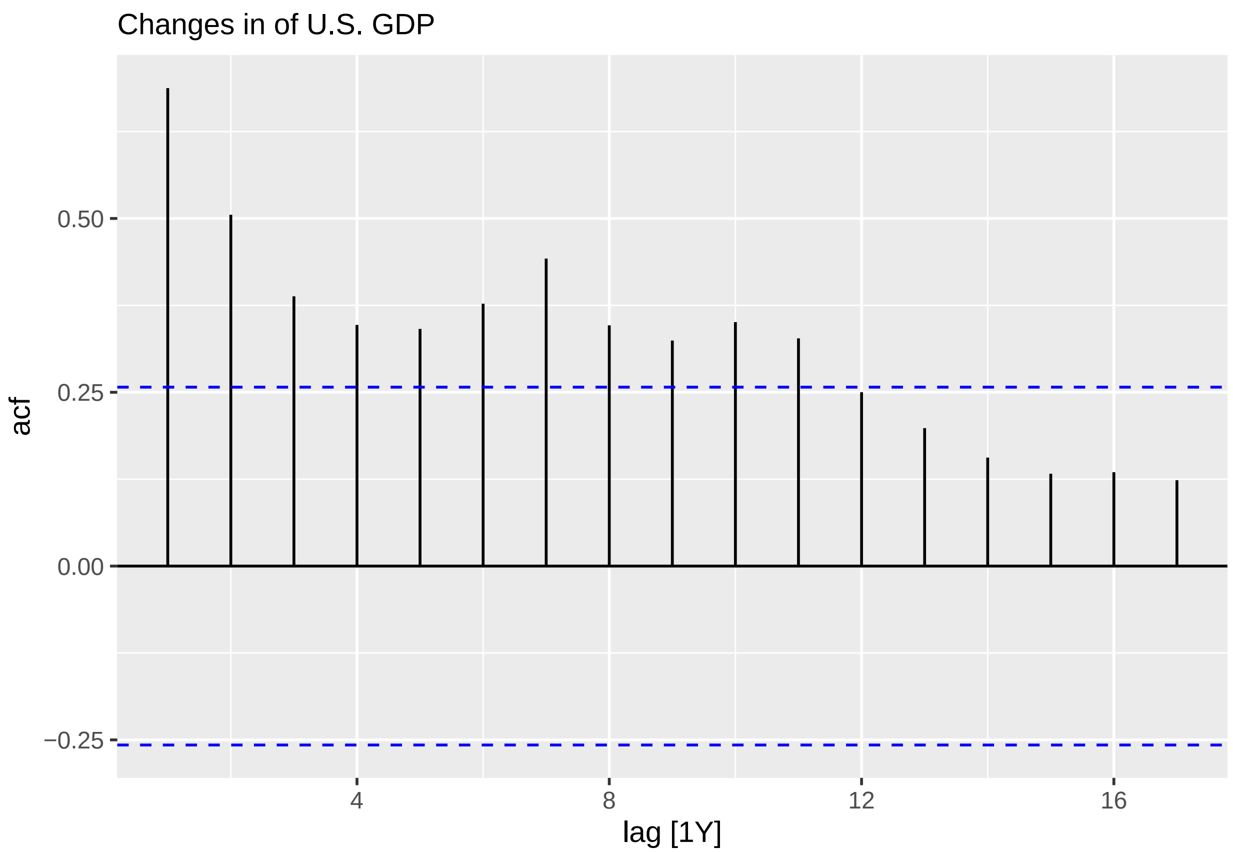
```
us <- global_economy %>%
  filter(Country == 'United States')%>%
  summarise(GDP= sum(GDP)/1e9)

us %>% autoplot(GDP) +
  labs(title = 'United States GDP')
```



```
us %>%  ACF(GDP) %>%
  autoplot() + labs(subtitle = "ACF of U.S. GDP")
```



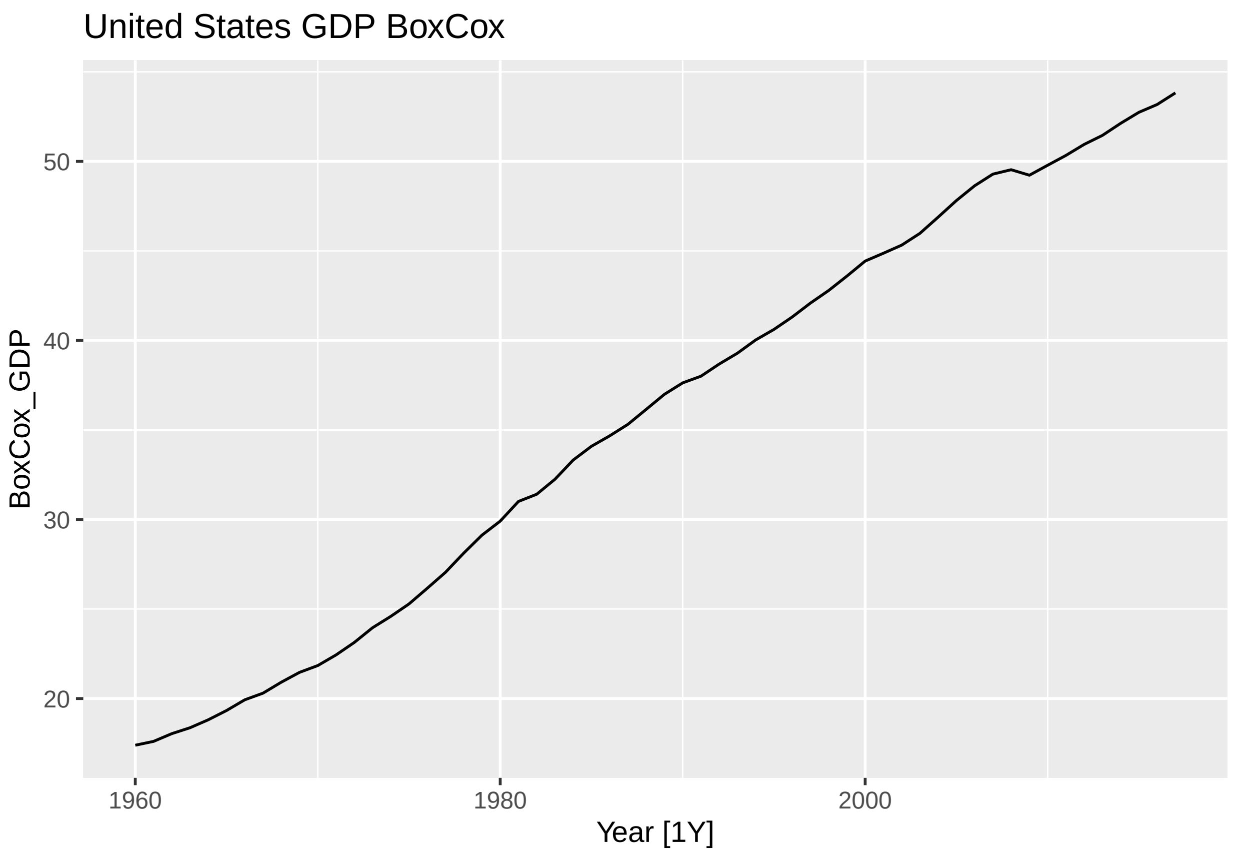
```
us %>%  ACF(difference(GDP)) %>%
  autoplot() + labs(subtitle = "Changes in of U.S. GDP")
```



```
us |>
  mutate(GDP = difference(GDP)) |>
  features(GDP, ljung_box, lag = 10)
```

```
## # A tibble: 1 x 2
##   lb_stat lb_pvalue
##     <dbl>     <dbl>
## 1    115.         0
```

```
#find lambad
lambad_us <- us %>%
  features(GDP, features = guerrero) %>%
  pull (lambda_guerrero)

us <- us %>%
  mutate(BoxCox_GDP = box_cox(GDP, lambad_us))

us_plot2 <- us |>
  autoplot(BoxCox_GDP) +
  labs(title = "United States GDP BoxCox")

us_plot2
```



- b.fit a suitable ARIMA model to the transformed data using ARIMA();

```
us_fit <- us|>
  model(ARIMA(box_cox(GDP, lambad_us)))

report(us_fit)
```

```
## Series: GDP
## Model: ARIMA(1,1,0) w/ drift
## Transformation: box_cox(GDP, lambad_us)
##
## Coefficients:
##          ar1  constant
##       0.4586    0.3428
## s.e.  0.1198    0.0276
##
## sigma^2 estimated as 0.0461:  log likelihood=7.72
## AIC=-9.43   AICc=-8.98   BIC=-3.3
```

- c.try some other plausible models by experimenting with the orders chosen;

```
us_fit2 <- us %>%
  model(ARIMA(box_cox(GDP, lambad_us) ~ pdq(2,1,1)))

report(us_fit2)
```

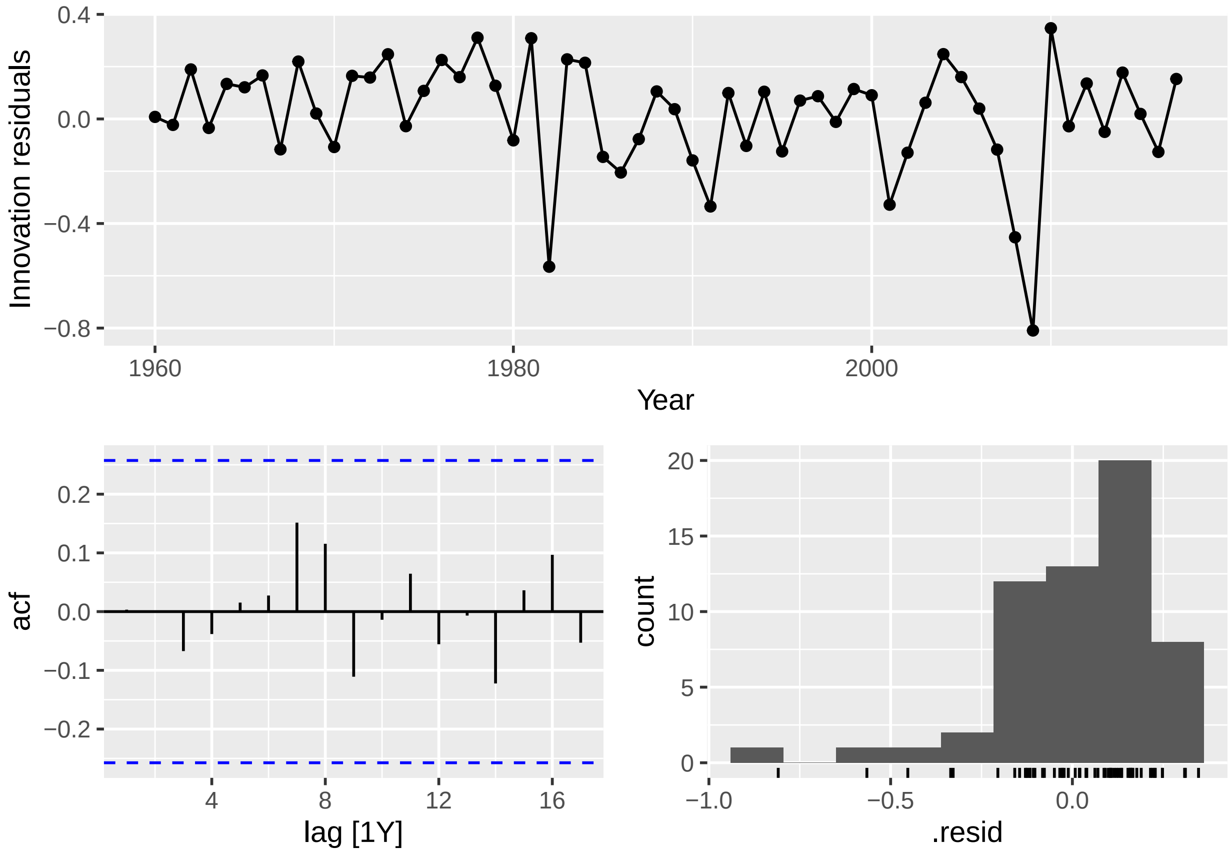
```
## Series: GDP
## Model: ARIMA(2,1,1) w/ drift
## Transformation: box_cox(GDP, lambad_us)
##
## Coefficients:
##          ar1      ar2      ma1  constant
##       1.1662  -0.2792  -0.7357    0.0706
## s.e.  0.3418   0.2108   0.3077    0.0074
##
## sigma^2 estimated as 0.04751:  log likelihood=7.9
## AIC=-5.79   AICc=-4.62   BIC=4.42
```

```
us_fit3 <- us %>%
  model(ARIMA(box_cox(GDP, lambad_us) ~ pdq(0,2,2)))

report(us_fit3)
```

```
## Series: GDP
## Model: ARIMA(0,2,2)
## Transformation: box_cox(GDP, lambad_us)
##
## Coefficients:
##           ma1      ma2
##       -0.5020  -0.2419
## s.e.   0.1303   0.1270
##
## sigma^2 estimated as 0.04832:  log likelihood=6.05
## AIC=-6.1   AICc=-5.64   BIC=-0.03
```

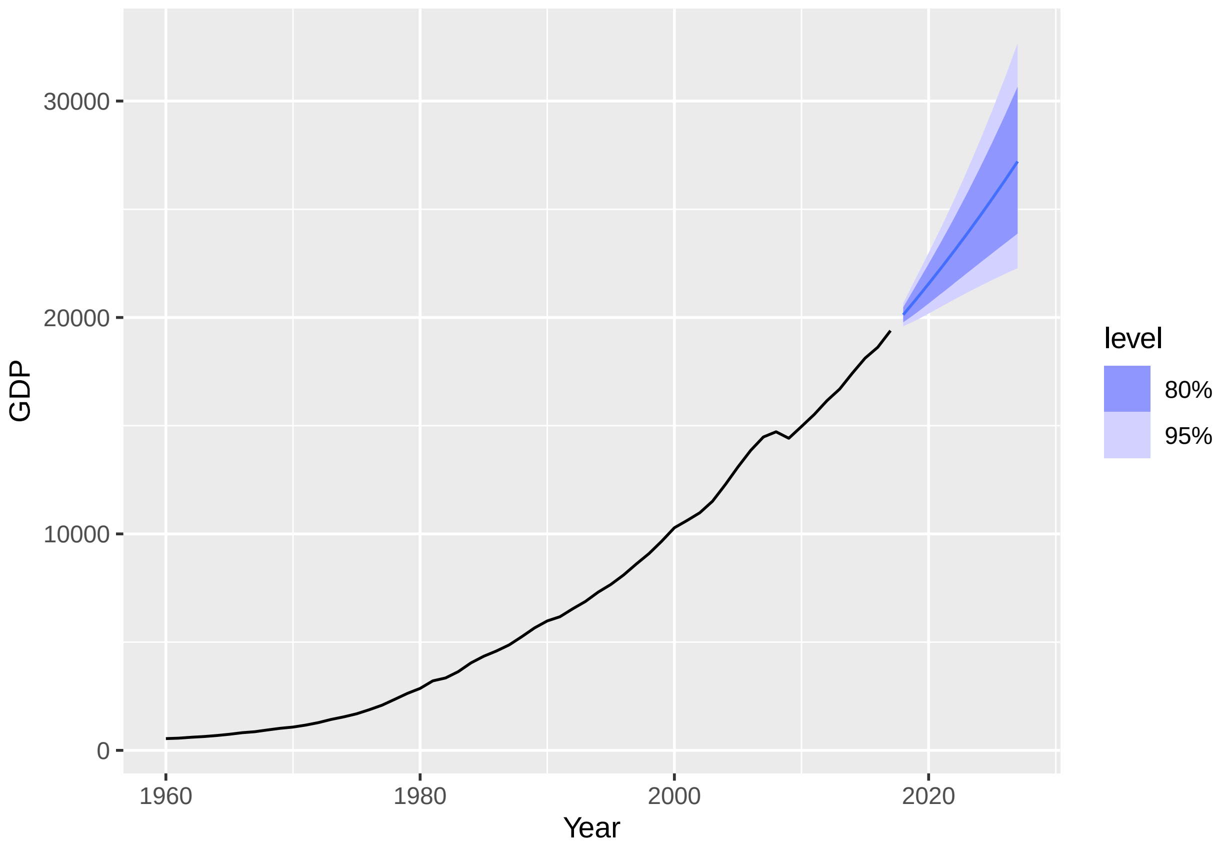
- d.choose what you think is the best model and check the residual diagnostics;

```
us_fit3 %>%
  gg_tsresiduals()
```



- e.produce forecasts of your fitted model. Do the forecasts look reasonable?
- It does look like reasonable, the trend going upward.

```
fc <- us_fit3 %>%
  forecast(h="10 years")

fc |>
  autoplot(us) +
  labs("10 Year United States GDP Prediction")
```

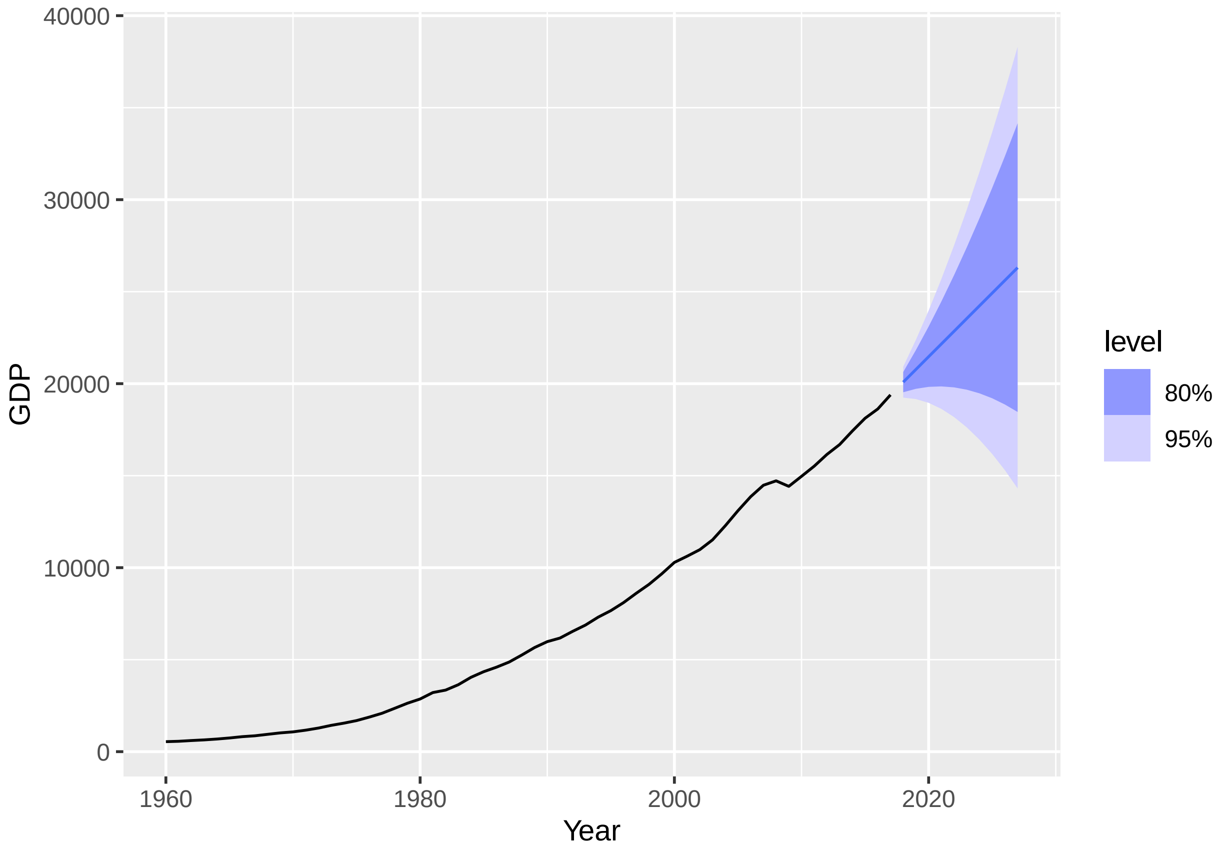


- f.compare the results with what you would obtain using ETS() (with no transformation).
- The ARIMA model with box-cox transformation has lower RMSE value than others, that means the model is better than others..

```
us_fit <- us |>
  model(ETS(GDP))

report(us_fit)
```

```
## Series: GDP
## Model: ETS(M,A,N)
##   Smoothing parameters:
##     alpha = 0.999899
##     beta  = 0.6151203
##
##   Initial states:
##      l[0]     b[0]
##  516.8849 26.39527
##
##   sigma^2:  5e-04
##
##      AIC     AICc      BIC
## 763.6422 764.7960 773.9444
```

```r
fc <- us_fit |>
  forecast(h="10 years")

fc |>
  autoplot(us)
```



```r
accuracy(us_fit)
```

```
## # A tibble: 1 x 10
##   .model   .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ETS(GDP) Training  18.6  167.  103. 0.585  1.67 0.302 0.410 0.0843
```

```r
accuracy(us_fit2)
```

```
## # A tibble: 1 x 10
##   .model                 .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
##   <chr>                  <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ARIMA(box_cox(GDP, la~ Trai~ -1.82  149.  87.1 0.0399  1.49 0.255 0.366 0.0624
```

```r
accuracy(us_fit3)
```

```
## # A tibble: 1 x 10
##   .model                 .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>                  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ARIMA(box_cox(GDP, lam~ Trai~ -5.93  156.  90.8 0.305  1.53 0.266 0.382 0.0701
```