Homework 6

Aaron Banlao

```
library(pacman)
p_load(fpp3, tidyverse)
```

Section 9.11 Exercises 1

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

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

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:

a) The three figures indicate that the data are white noise because the spikes are within the bounded areas. As the number of numbers increase the ACF plot bounds get smaller, meaning that the auto correlations get closer to zero.

b) As stated above, as the number grows, the bounds get smaller because the denominator would be bigger when calculating the bounds. The larger the number, the less chance of autocorrelation.

Section 9.11 Exercises 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 series in non-stationary because the data has an increasing trend. The data is non-stationary according to the ACF plot because instead of dropping to zero, it is constantly near one. The closing price would need to be differenced to become stationary.

Code and Comments:

```
gafa_stock %>%
  distinct(Symbol)
```

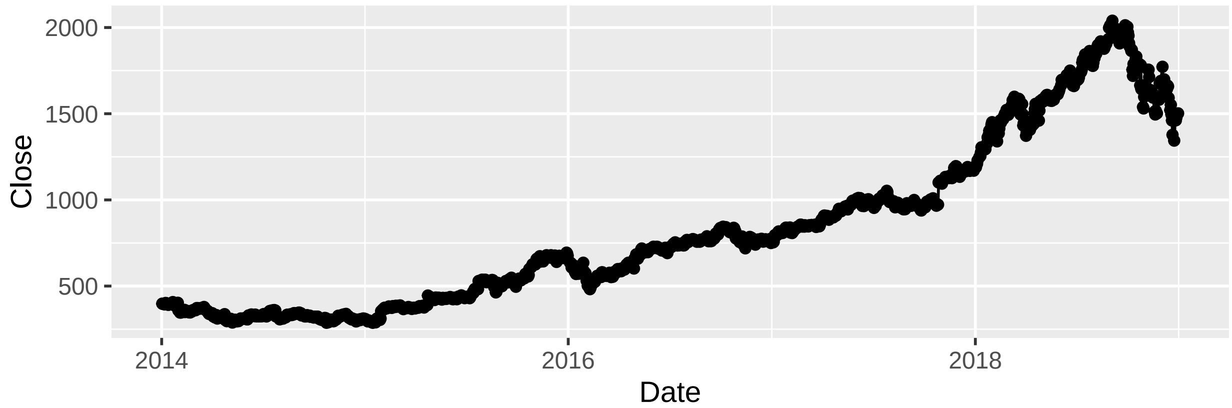
```
## # A tibble: 4 x 1
##   Symbol
##   <chr>
## 1 AAPL
## 2 AMZN
## 3 FB
## 4 GOOG
```

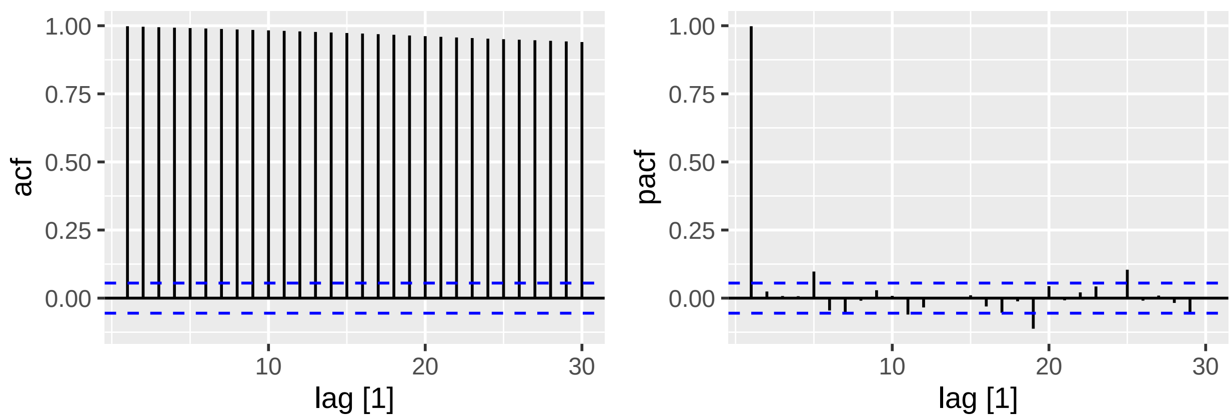
```
head(gafa_stock)
```

```
## # A tsibble: 6 x 8 [!]
## # Key:       Symbol [1]
##   Symbol Date        Open  High   Low Close Adj_Close    Volume
##   <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>     <dbl>
## 1 AAPL   2014-01-02  79.4  79.6  78.9  79.0      67.0  58671200
## 2 AAPL   2014-01-03  79.0  79.1  77.2  77.3      65.5  98116900
## 3 AAPL   2014-01-06  76.8  78.1  76.2  77.7      65.9 103152700
## 4 AAPL   2014-01-07  77.8  78.0  76.8  77.1      65.4  79302300
## 5 AAPL   2014-01-08  77.0  77.9  77.0  77.6      65.8  64632400
## 6 AAPL   2014-01-09  78.1  78.1  76.5  76.6      65.0  69787200
```

```
gafa_stock %>%
  filter(Symbol == "AMZN") %>%
  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
```





Section 9.11 Exercises 6

Answer:

b) As the number gets smaller, there are more random fluctuations.

d) Similar with answer b, the smaller the number gets, there are more random fluctuations.

Code and Comments:

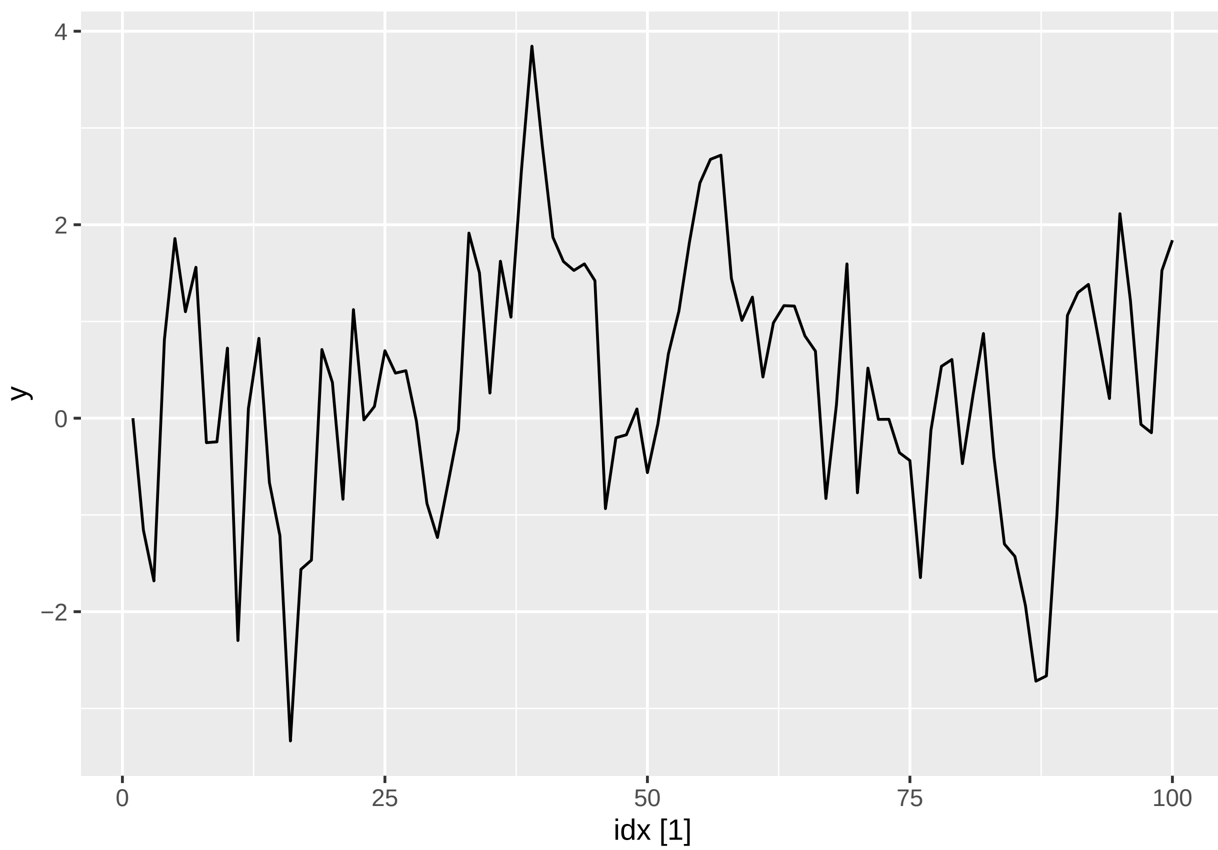
```
y <- numeric(100)
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.6*y[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

a)

```
sim %>%
  autoplot()
```

b)

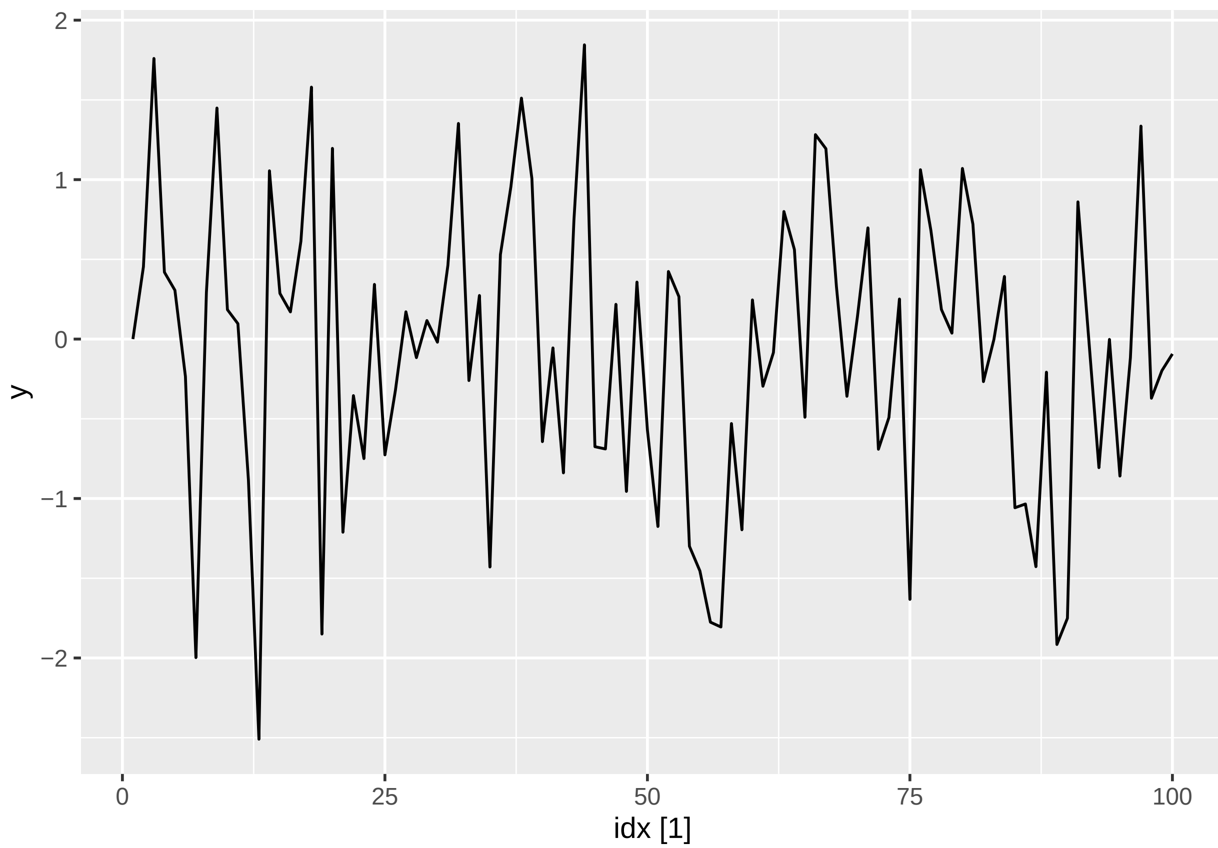
```
## Plot variable not specified, automatically selected `.vars = y`
```




```
y <- numeric(100)
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.06*y[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim %>%
  autoplot()
```

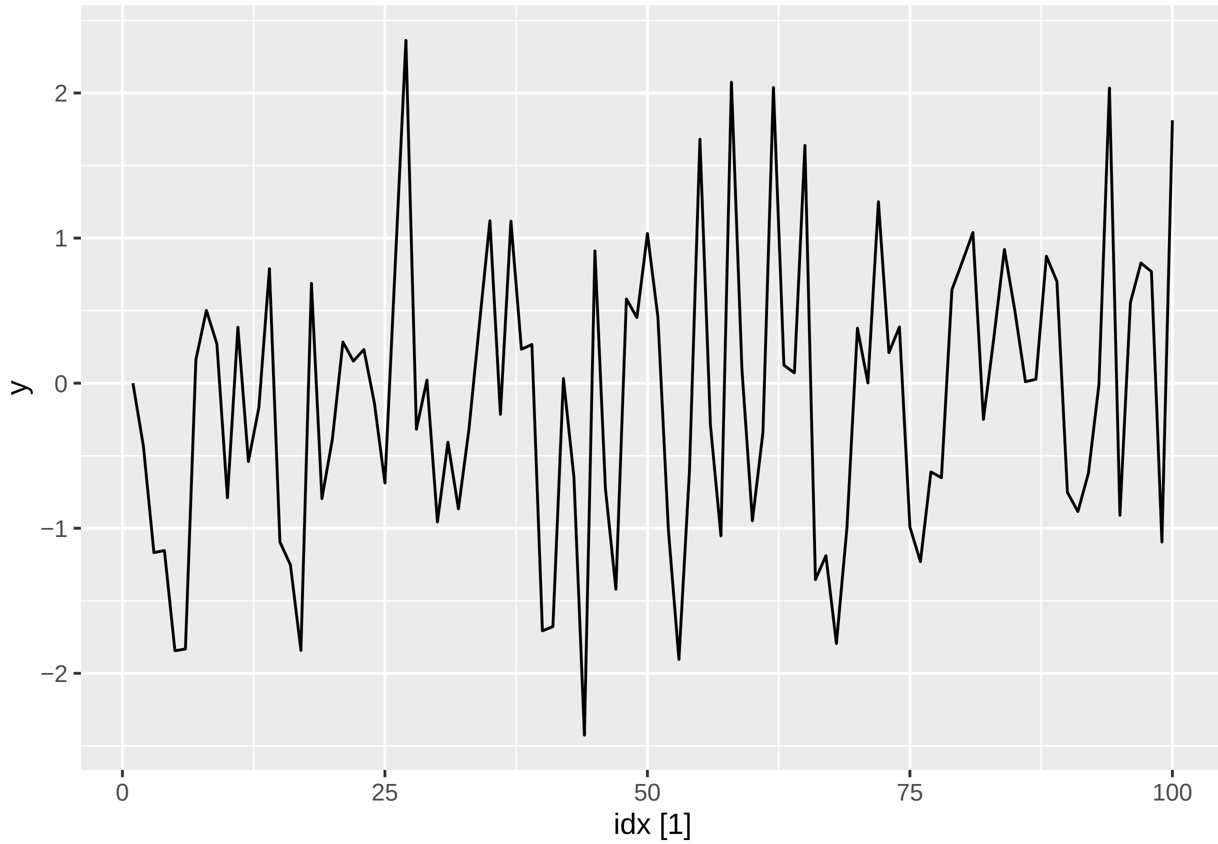
```
## Plot variable not specified, automatically selected `.vars = y`
```




```
y <- numeric(100)
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.00006*y[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim %>%
  autoplot()
```

```
## Plot variable not specified, automatically selected `.vars = y`
```

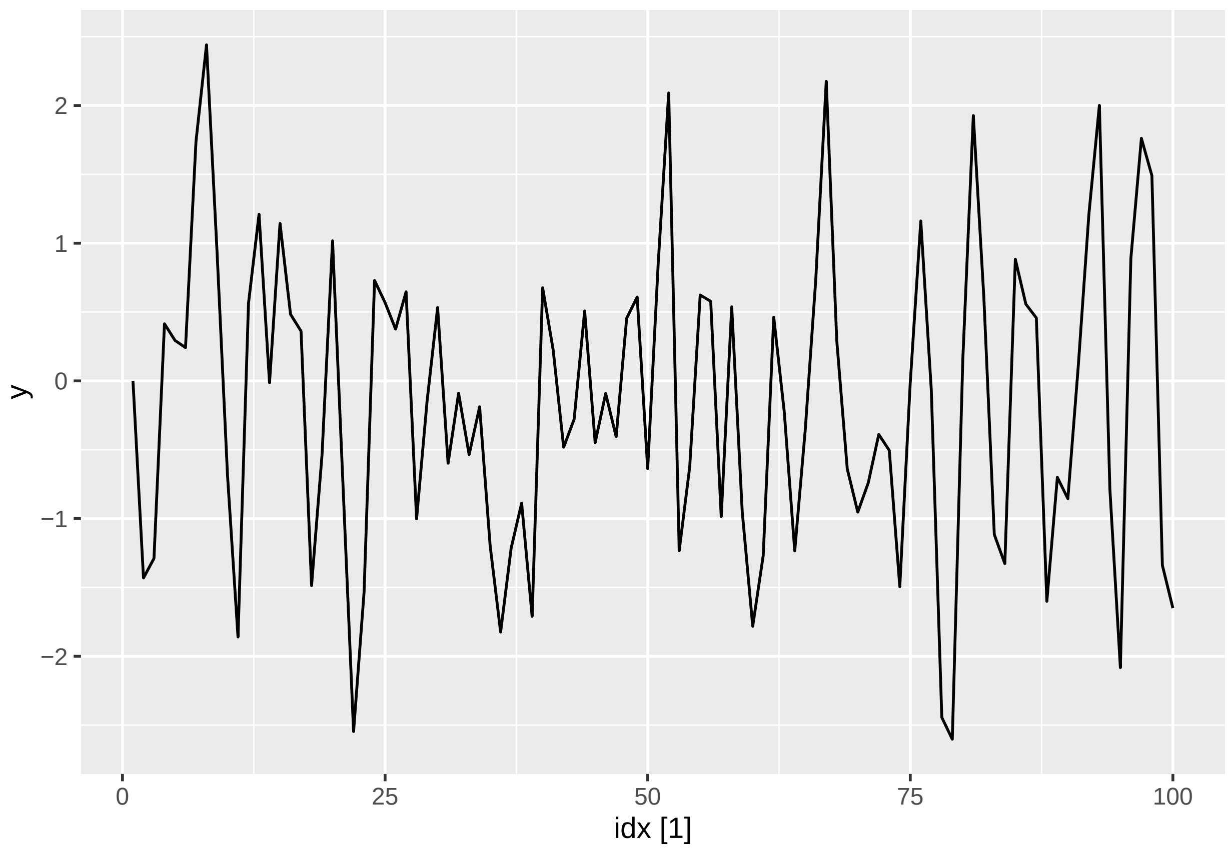



```
y <- numeric(100)
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.6*e[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim %>%
  autoplot()
```

c)

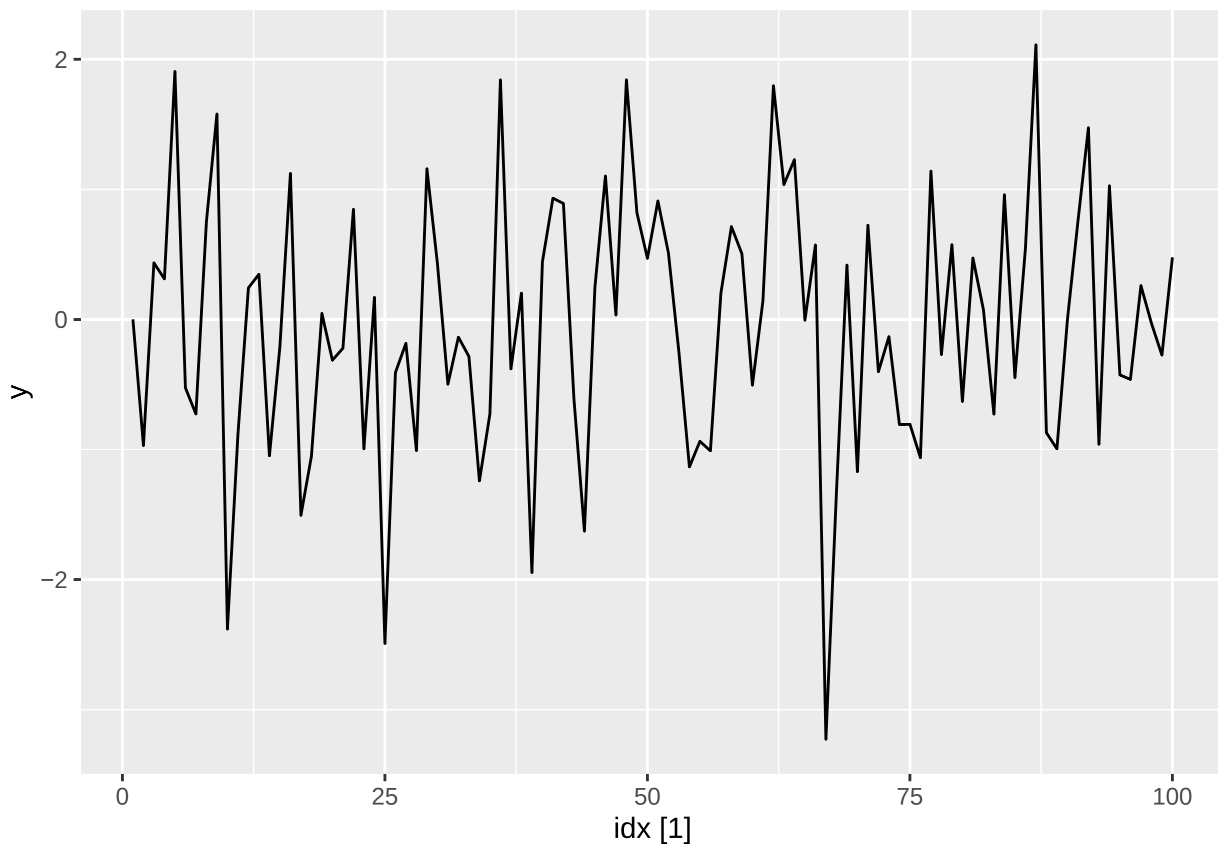
```
## Plot variable not specified, automatically selected ‘.vars = y‘
```




```
y <- numeric(100)
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.06*e[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim %>%
  autoplot()
```

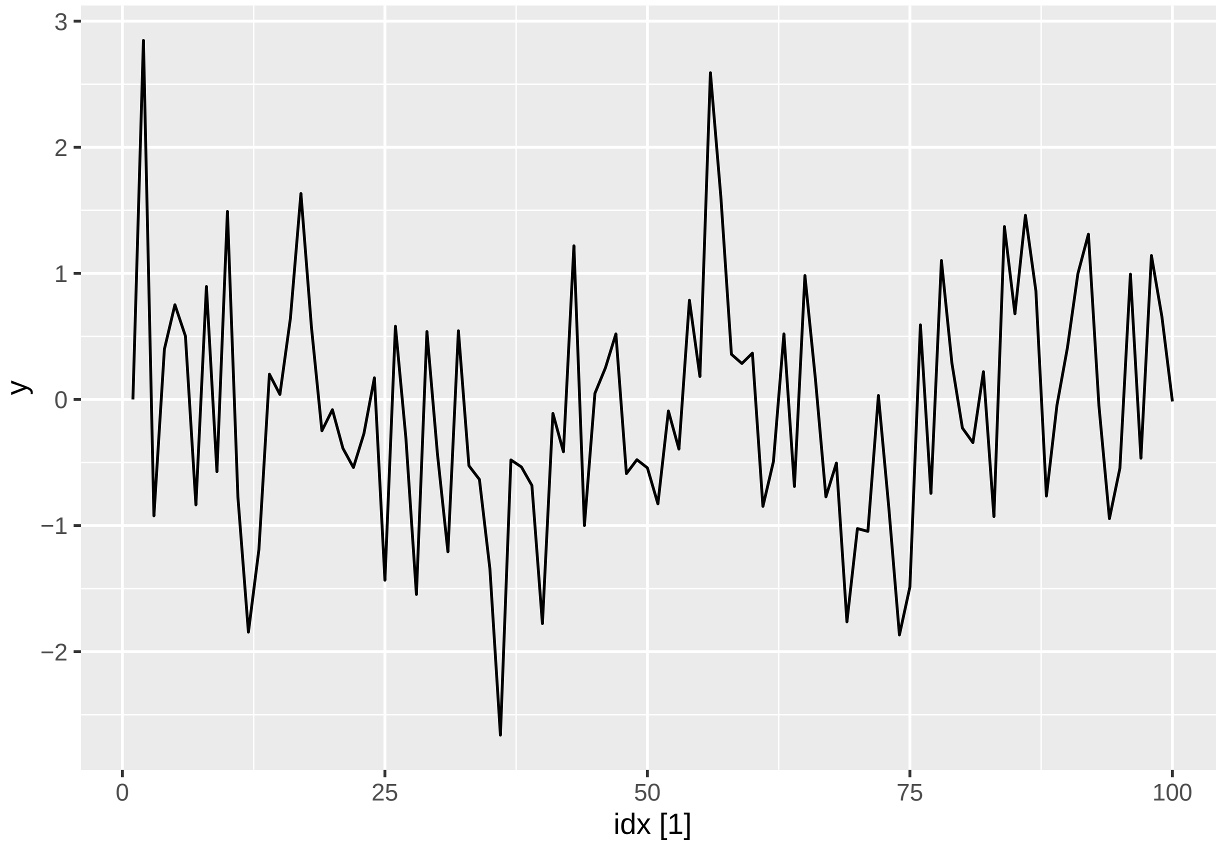
```
## Plot variable not specified, automatically selected `.vars = y`
```




```
y <- numeric(100)
e <- rnorm(100)
for(i in 2:100)
  y[i] <- 0.00006*e[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim %>%
  autoplot()
```

```
## Plot variable not specified, automatically selected `.vars = y`
```

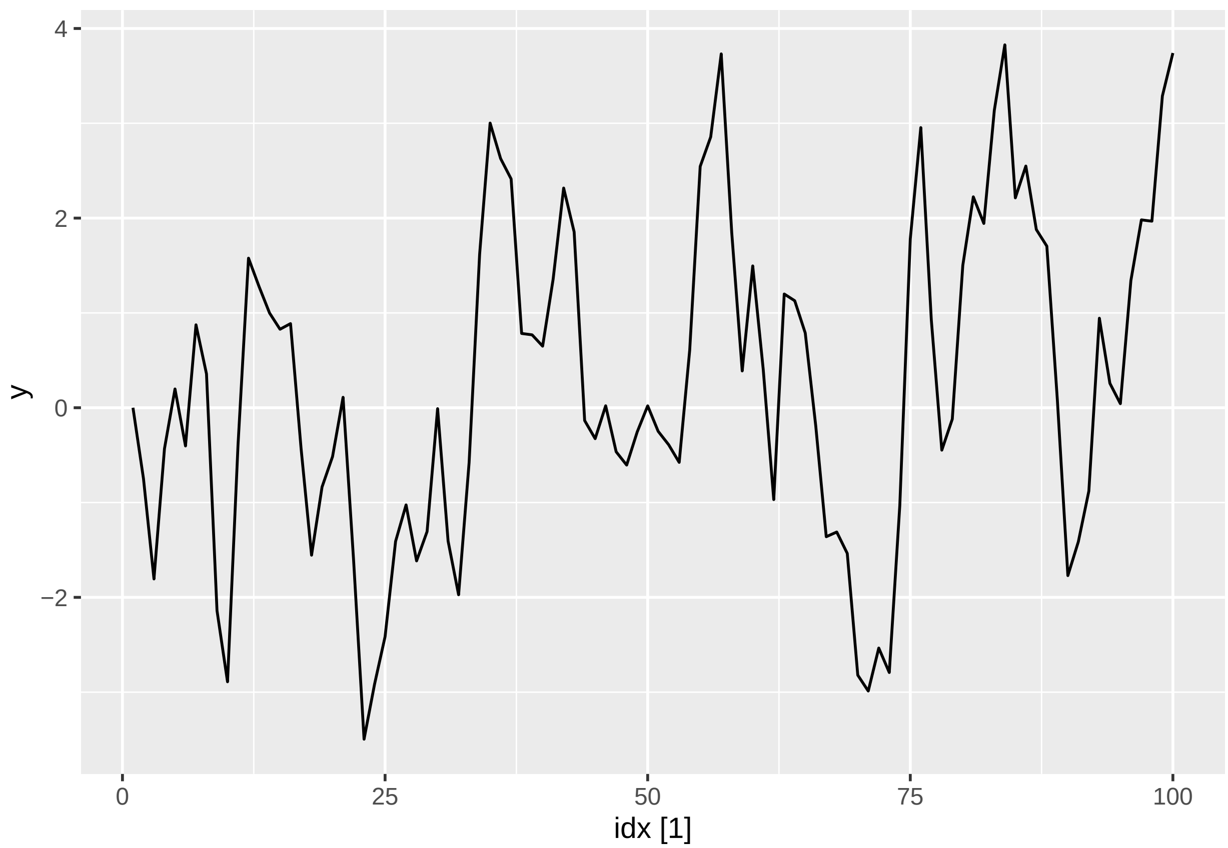



```
y <- numeric(100)
e <- rnorm(100, sd = 1)
for(i in 2:100)
  y[i] <- 0.6*y[i-1] + 0.6*e[i-1] + e[i]
sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim %>%
  autoplot()
```

e)

```
## Plot variable not specified, automatically selected `.vars = y`
```

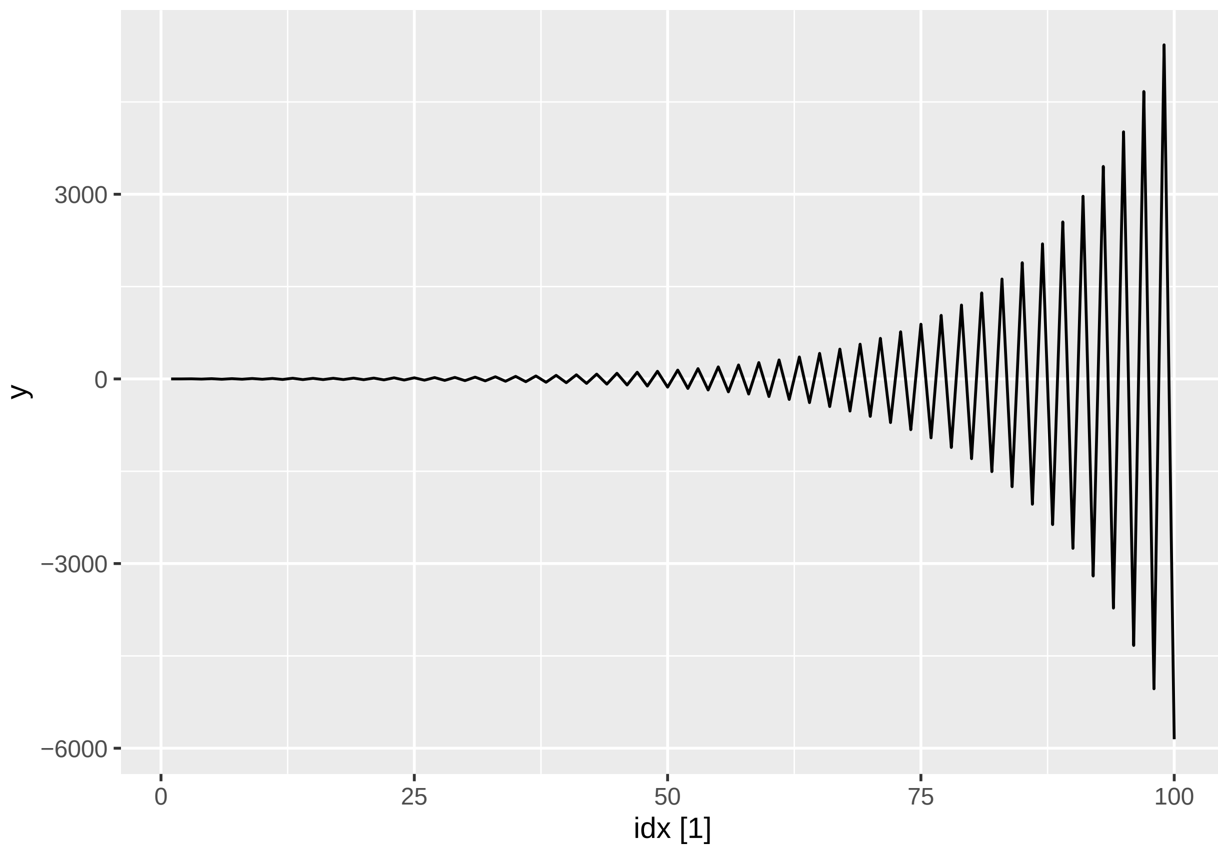



```
y <- numeric(100)
e <- rnorm(100, sd=1)
for(i in 3:100)
  y[i] <- -0.8*y[i-1] + 0.3*y[i-2] + e[i]
sim1 <- tsibble(idx = seq_len(100), y = y, index = idx)
```

```
sim1 %>%
  autoplot()
```

f)

```
## Plot variable not specified, automatically selected `.vars = y`
```



Section 9.11 Exercises 8

For the United States GDP series (from global_economy):

if necessary, find a suitable Box-Cox transformation for the data; fit a suitable ARIMA model to the transformed data using ARIMA(); try some other plausible models by experimenting with the orders chosen; choose what you think is the best model and check the residual diagnostics; produce forecasts of your fitted model. Do the forecasts look reasonable? compare the results with what you would obtain using ETS() (with no transformation).

Answer:

d) The best model that has the lowest AICc is the arima1 model.

e) The forecast made with the model chosen looks reasonable as it follows the trend

f) With the ETS model, there is a wider prediction interval

Code and Comments:

```
head(global_economy)
```

a)

```
## # 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
```

```
global_economy %>%
  distinct(Country)
```

```
## # A tibble: 263 x 1
##    Country
##    <fct>
##  1 Afghanistan
##  2 Albania
##  3 Algeria
##  4 American Samoa
##  5 Andorra
##  6 Angola
##  7 Antigua and Barbuda
##  8 Arab World
##  9 Argentina
## 10 Armenia
## # ... with 253 more rows
```

```
us <- global_economy %>%
  filter(Country == "United States") %>%
  select(Country, GDP)
```

```
lambda <-  us %>%
  features(GDP, features = guerrero) %>%
  pull(lambda_guerrero)
```

```
us <- us %>%
  mutate(GDP = box_cox(GDP, lambda))
```

```
fit <- us %>%
  model(ARIMA(GDP, stepwise = F, approximation = F))

report(fit)
```

b)

```
## Series: GDP
## Model: ARIMA(1,1,0) w/ drift
##
## Coefficients:
##          ar1  constant
##       0.4586  118.1822
## s.e.  0.1198    9.5047
##
## sigma^2 estimated as 5479:  log likelihood=-325.32
## AIC=656.65   AICc=657.1   BIC=662.78
```

```
fit2 <- us %>%
  model(arima1 = ARIMA(GDP ~ pdq(1, 2, 1)),
        arima2 = ARIMA(GDP ~ pdq(1, 1, 1)),
        arima3 = ARIMA(GDP ~ pdq(2, 2, 1)))

report(fit2)
```

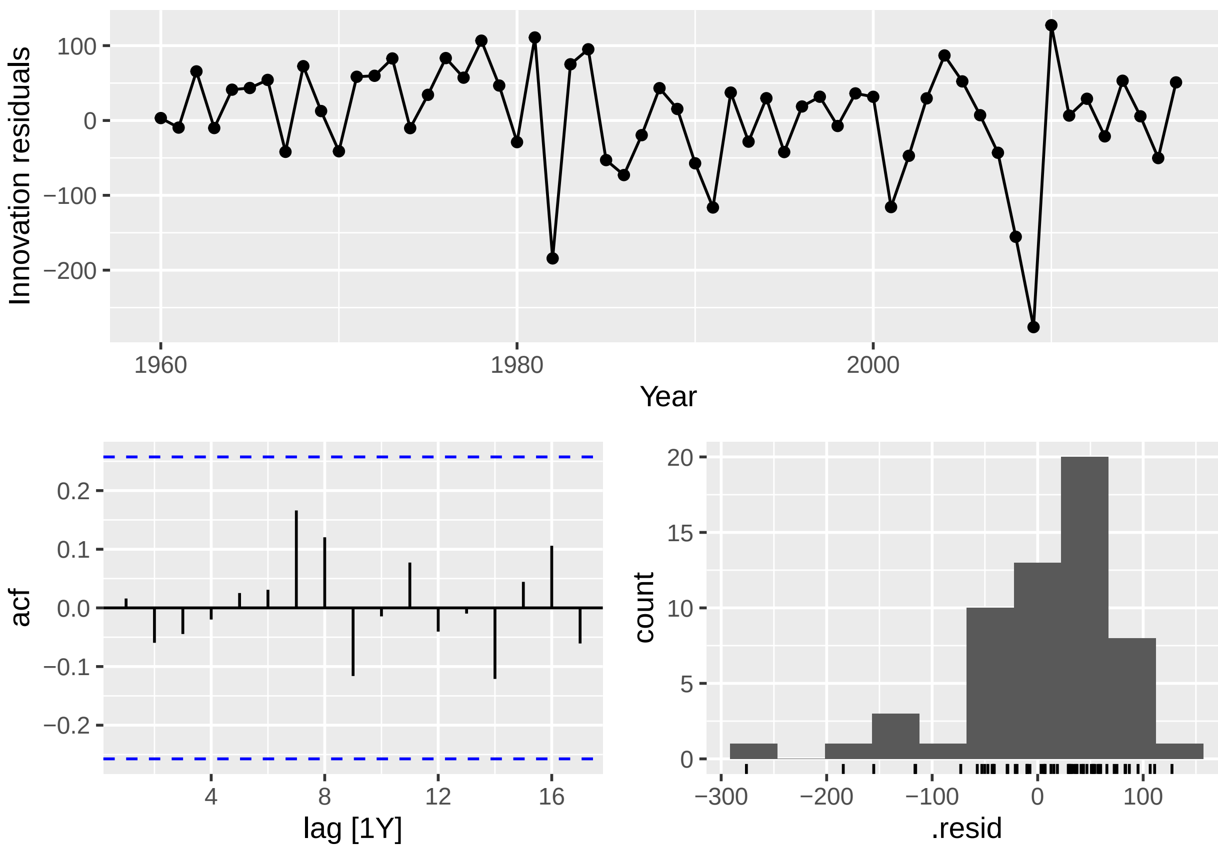
c)

```
## Warning in report.mdl_df(fit2): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 3 x 9
##   Country       .model sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <fct>         <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 United States arima1  5761.   -321.  648.  649.  655. <cpl [1]> <cpl [1]>
## 2 United States arima2  5580.   -325.  659.  659.  667. <cpl [1]> <cpl [1]>
## 3 United States arima3  5834.   -321.  650.  651.  658. <cpl [2]> <cpl [1]>
```

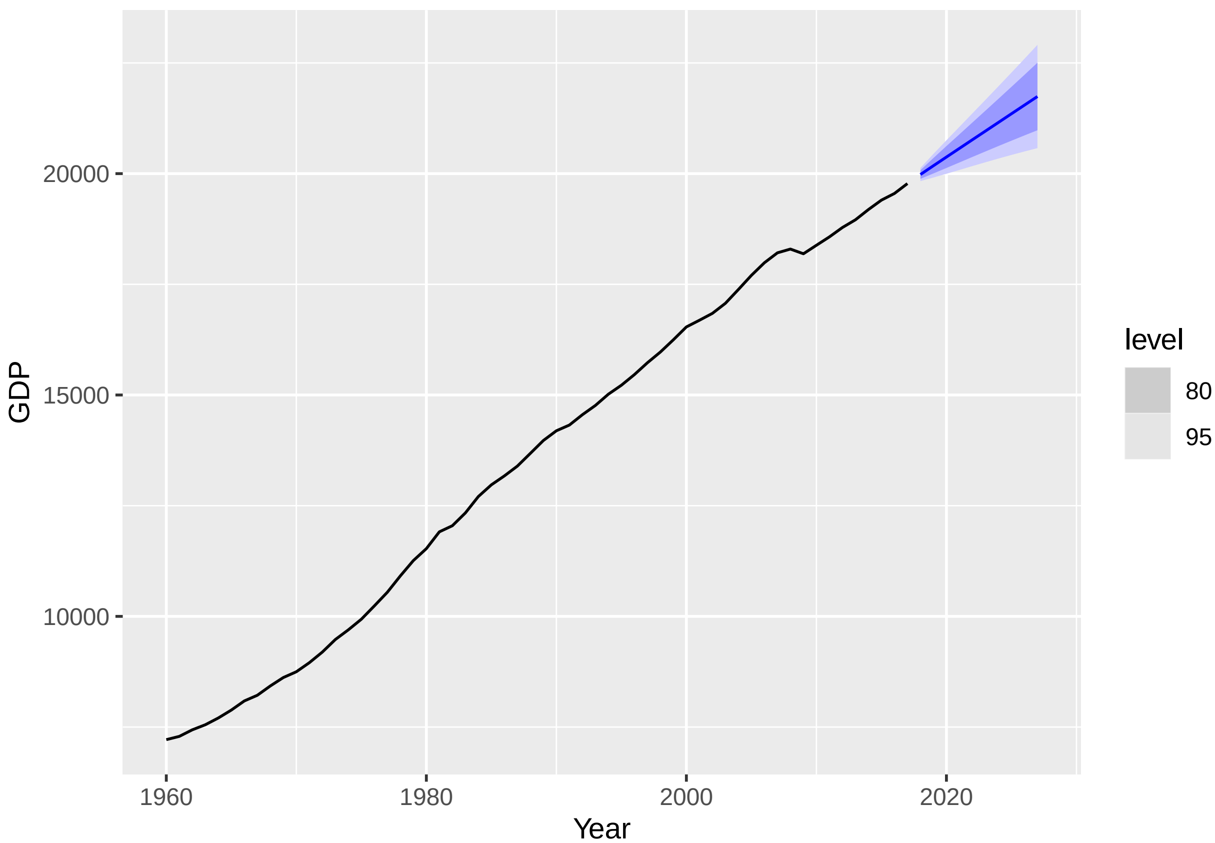
```
fit3 <- us %>%
  model(ARIMA(GDP ~ pdq(1, 2, 1,)))
```

```
fit3 %>%
  gg_tsresiduals()
```



d)

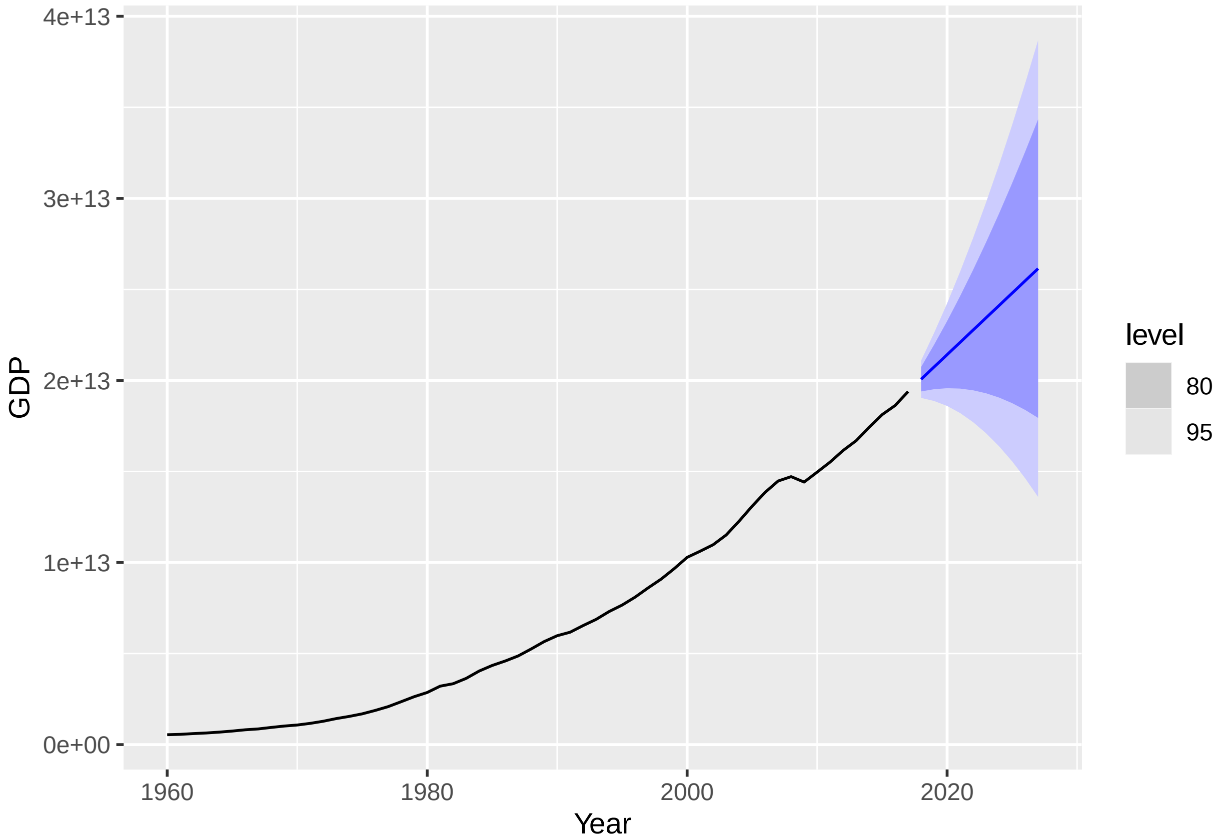
```
fit3 %>%
  forecast(h = 10) %>%
  autoplot(us)
```



e)

```
us2 <- global_economy %>%
  filter(Country == "United States") %>%
  select(Country, GDP)
```

```
us2 %>%
  model(ETS(GDP)) %>%
  forecast(h = 10) %>%
  autoplot(us2)
```



4e+13
3e+13
2e+13
1e+13
0e+00
GDP
1960
1980
2000
2020
Year
level
80
95

f)