Week 4 chapter 5

Chi Hang(Philip) Cheung

2025-02-22

```
library(fpp3)
```

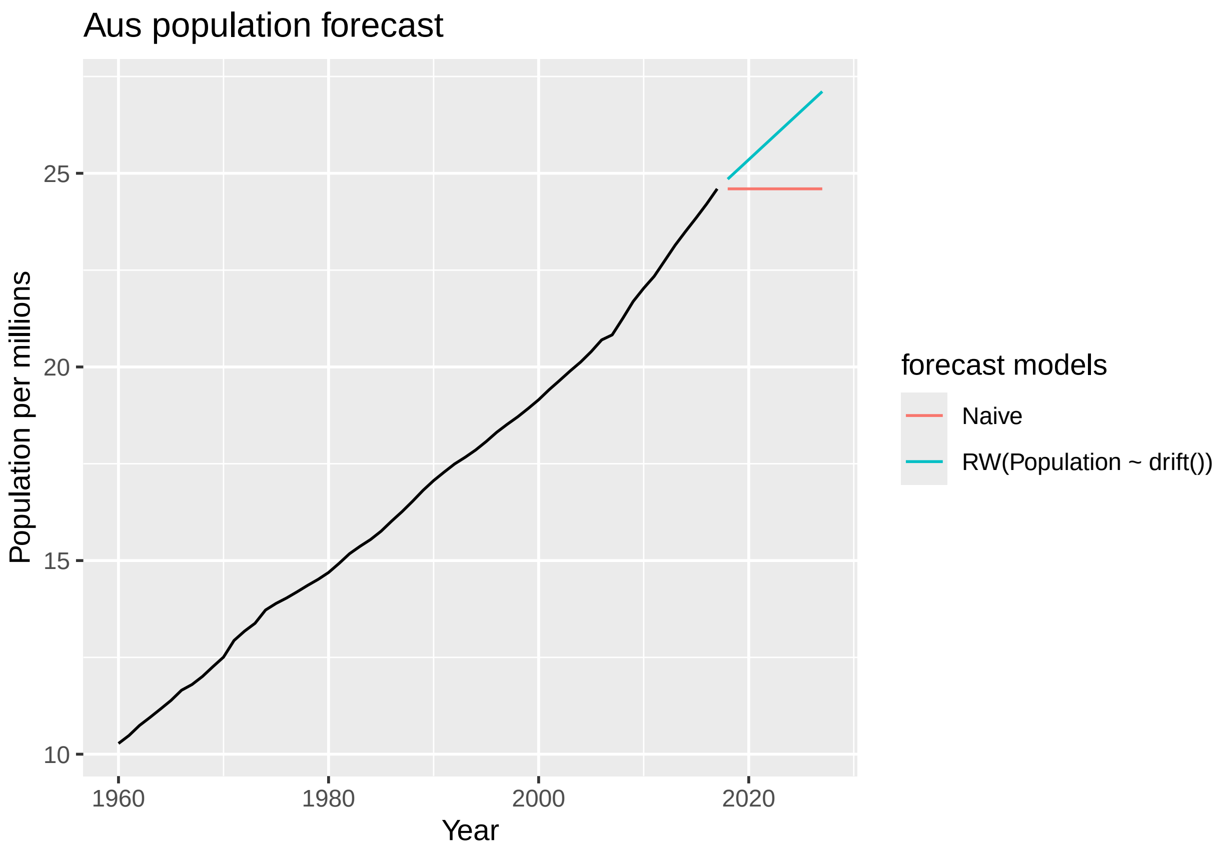
```
## Registered S3 method overwritten by 'tsibble':
##   method               from 
##   as_tibble.grouped_df dplyr

## -- 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

## -- 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()
```

5.1 Produce forecasts for the following series using whichever of NAIVE(y), SNAIVE(y) or RW(y ~ drift()) is more appropriate in each case:

Australian Population (global_economy) Only Naive and RW were used because there is not seasonal pattern in the dataset.

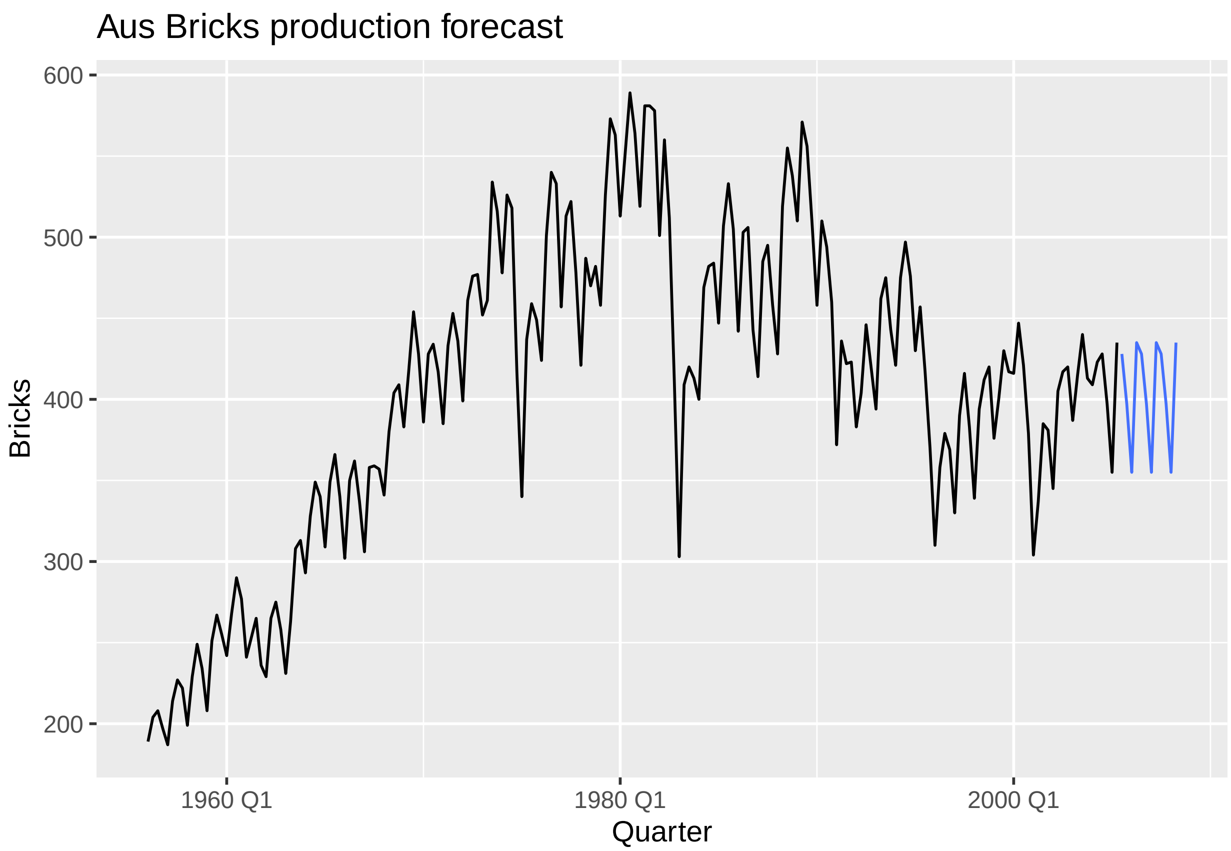
```
aus_pop <- global_economy %>%
  filter(Country == 'Australia') %>%
  select(Year, Population) %>%
  mutate(Population = Population/1e6)

aus_fit<- aus_pop %>% model(
  Naive = NAIVE(Population),
  RW(Population ~ drift())
)
aus_pop_fc<- aus_fit %>%
  forecast(h = '10 year')
```

```
aus_pop_fc %>%
  autoplot(aus_pop, level = NULL)+
  labs(title='Aus population forecast', y='Population per millions')+
  guides(color = guide_legend(title = 'forecast models'))
```



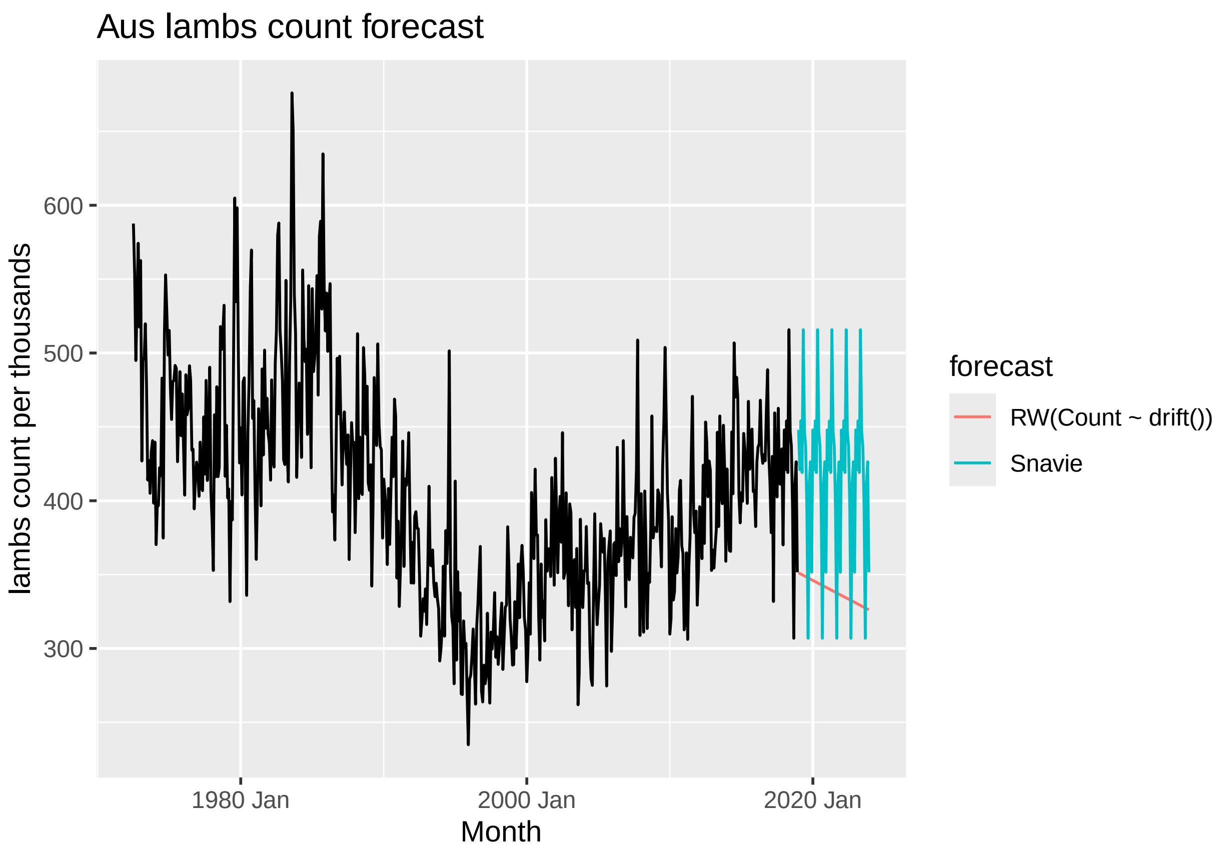
Bricks (aus_production) SNAIVE is used due to seasonality pattern observed.

```
aus_brick<- aus_production %>%
  select(Quarter, Bricks) %>%
  filter(!is.na(Bricks))

aus_brick_fit<- aus_brick %>%
  model(
    Snavie = SNAIVE(Bricks),
    )

aus_brick_fc<- aus_brick_fit %>%
  forecast(h = 12)

aus_brick_fc %>%
  autoplot(aus_brick, level = NULL)+
  labs(title = 'Aus Bricks production forecast')+
  guides(color = guide_legend(title = 'forecast'))
```



NSW Lambs (aus_livestock) SNAIVE and RW are used since there is a seasonality pattern and a downward trend is observed.

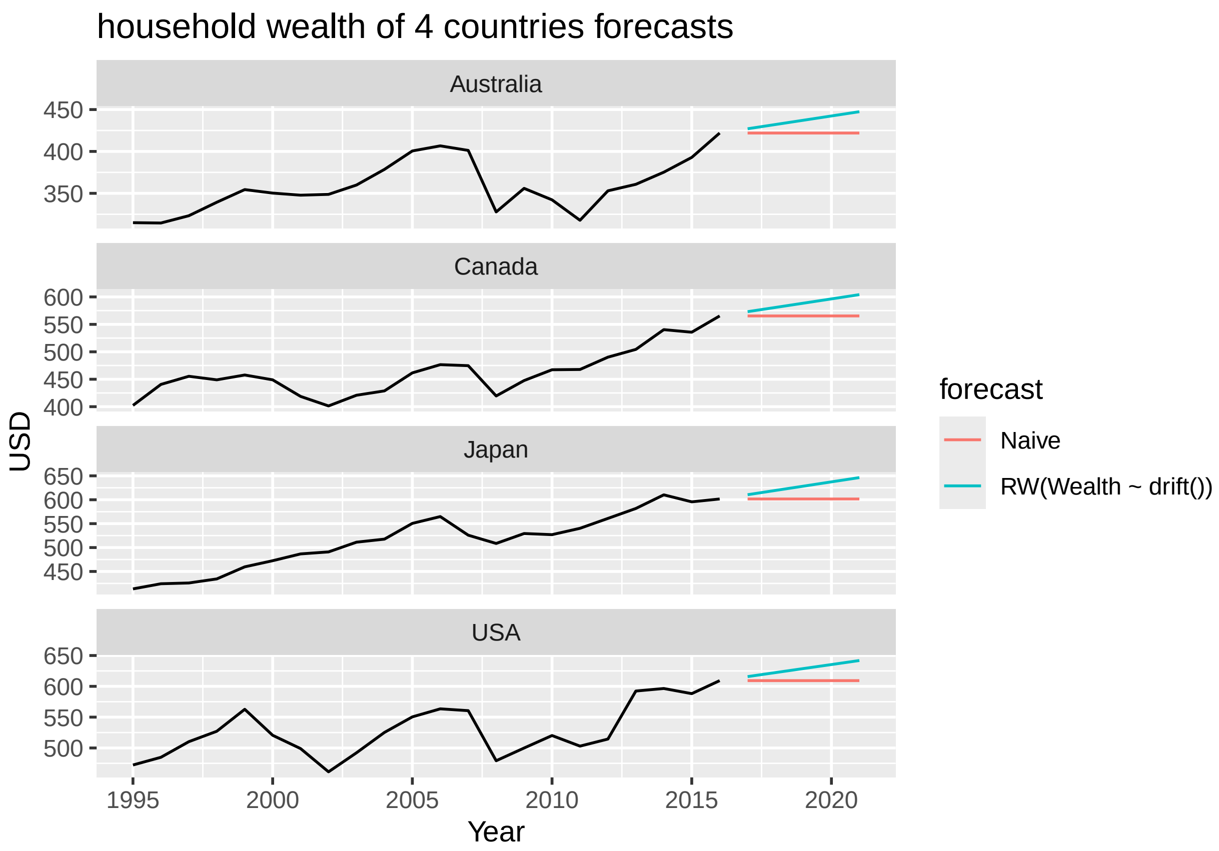
```
lamb<- aus_livestock %>%
  filter(Animal == 'Lambs', State == 'New South Wales') %>%
  mutate(Count = Count/1000)

lambs_fit<- lamb %>%
  model(
    Snavie = SNAIVE(Count),
    RW(Count ~ drift())
  )

lambs_fc<- lambs_fit %>%
  forecast(h = '5 year')

lambs_fc %>%
  autoplot(lamb, level = NULL)+
  labs(title = 'Aus lambs count forecast', y='lambs count per thousands')+
  guides(color = guide_legend(title = 'forecast'))
```



Household wealth (hh_budget) Only Naive and RW are used due to lack of seasonality from the data

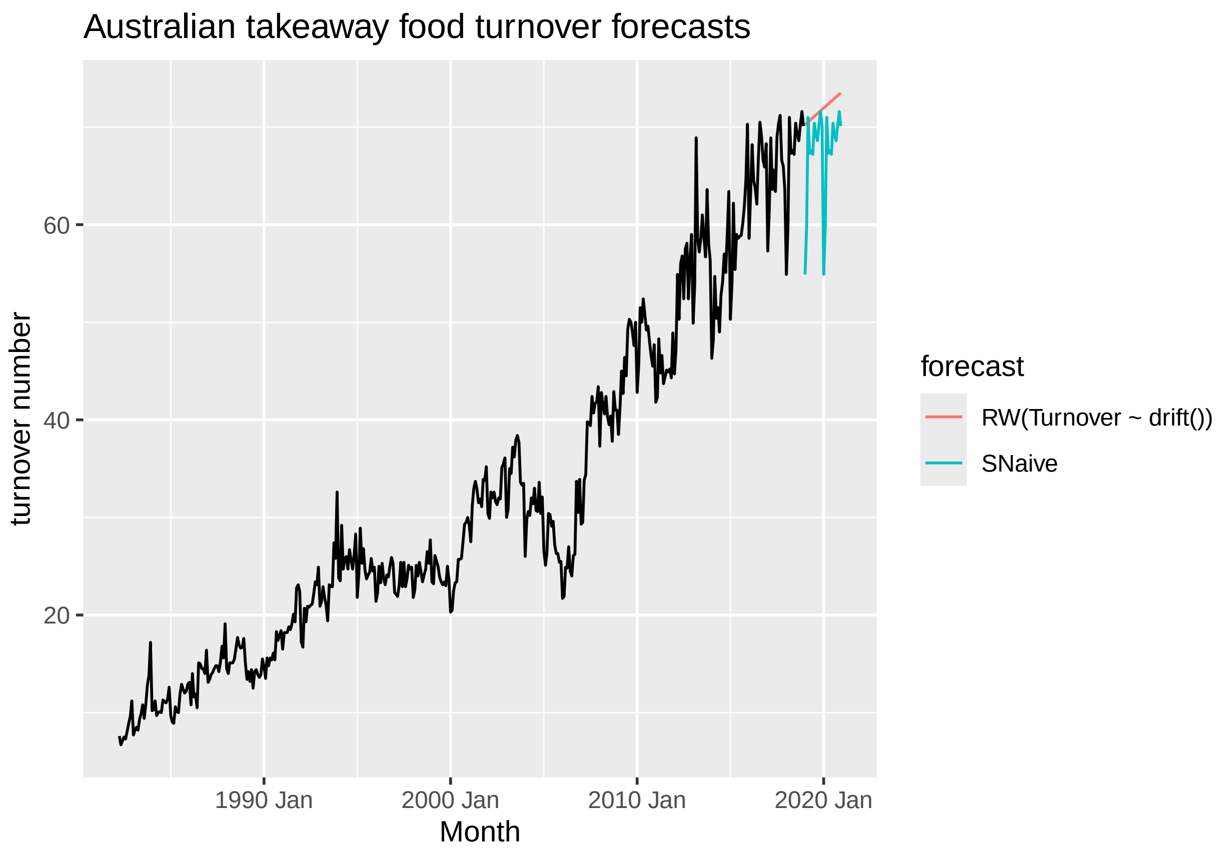
```
wealth<- hh_budget %>%
  select(Year, Wealth)

wealth_fit<- wealth %>%
  model(
    Naive = NAIVE(Wealth),
    RW(Wealth ~ drift())
  )

wealth_fc<- wealth_fit %>%
  forecast(h = '5 year')

wealth_fc %>%
  autoplot(wealth, level = NULL, scale='free_y')+
  labs(title = 'household wealth of 4 countries forecasts', y='USD')+
  guides(color = guide_legend(title = 'forecast'))
```

```
## Warning in geom_line(mapping = without(mapping, "shape"), data =
## unpack_data(object[single_row[["FALSE"]], : Ignoring unknown parameters:
## ‘scale‘
```



Australian takeaway food turnover SNAIVE and RW are used since the data have an upward trend. Using NAIVE might be undermining the data trend.

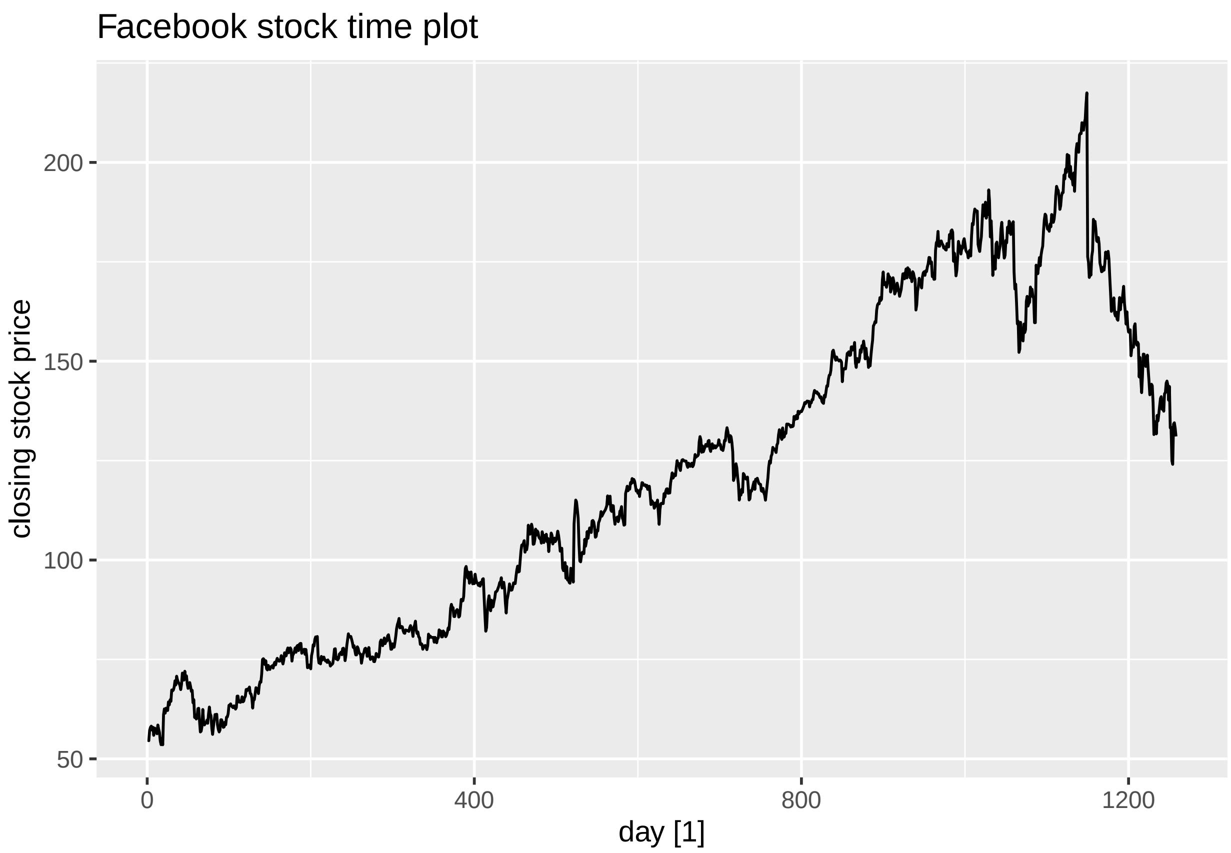
```
turnover<- aus_retail %>%
  filter(Industry == 'Cafes, restaurants and takeaway food services', State== 'Australian Capital Terri
  select(Month, Turnover)

turnover_fit<- turnover %>%
  model(
    SNaive = SNAIVE(Turnover),
    RW(Turnover ~ drift())
  )

turnover_fc<- turnover_fit %>%
  forecast(h = 24)

turnover_fc %>%
  autoplot(turnover, level = NULL)+
  labs(title = 'Australian takeaway food turnover forecasts', y='turnover number')+
  guides(color = guide_legend(title = 'forecast'))
```



5.2 Use the Facebook stock price (data set gafa_stock) to do the following:

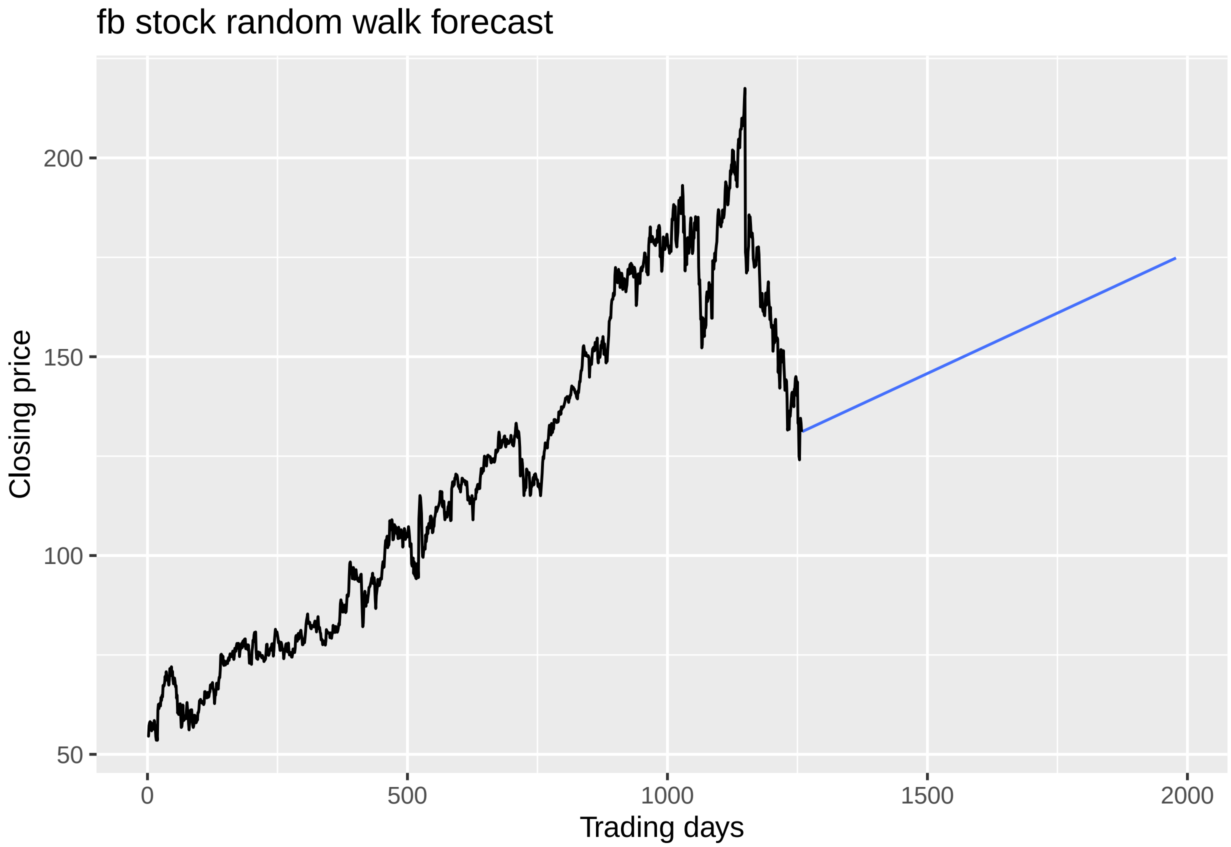
a) Produce a time plot of the series.

```
#Since the trading days are irregular, a modification of the days are needed. Counted the days as the i
fb<- gafa_stock %>%
  filter(Symbol == 'FB') %>%
  mutate(day = row_number()) %>%
  select(day, Close) %>%
  update_tsibble(index=day, regular = TRUE)

fb %>% autoplot(Close)+
  labs(y='closing stock price', title='Facebook stock time plot')
```



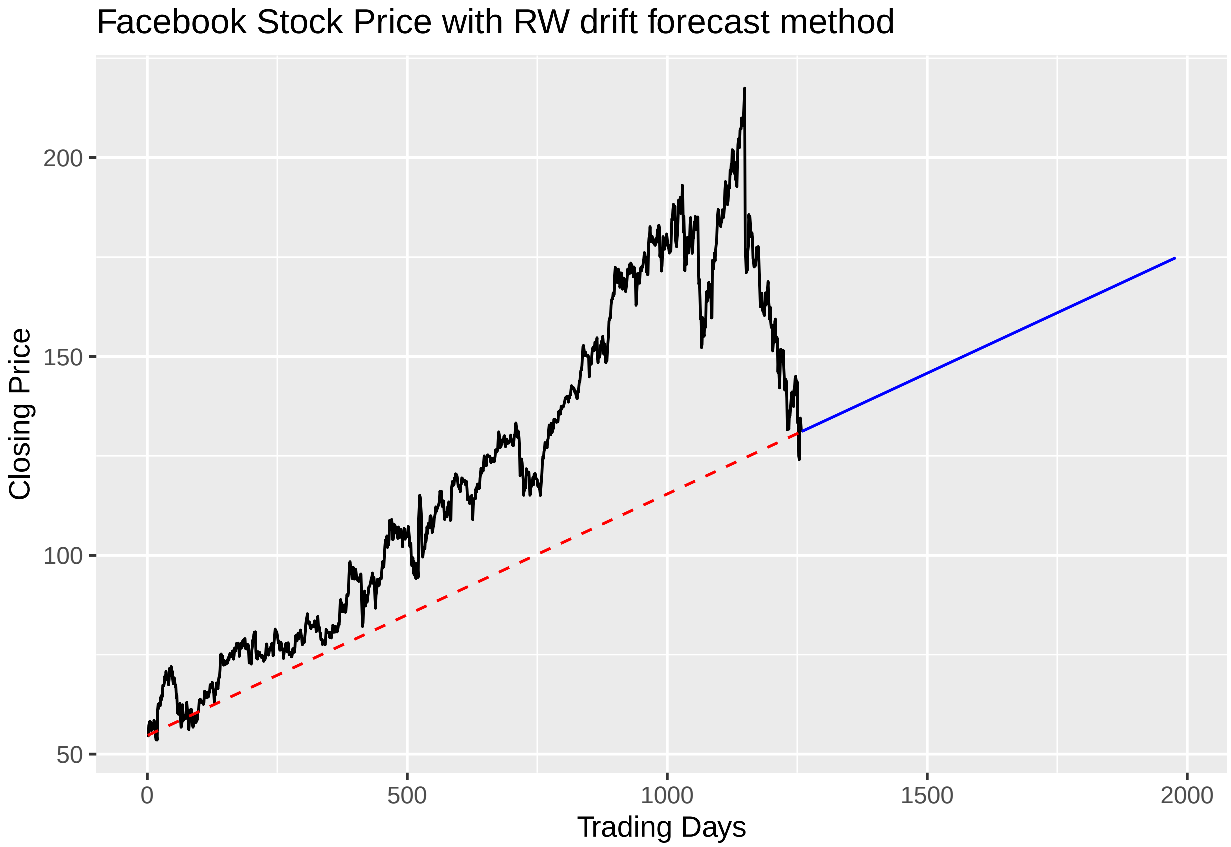
b) Produce forecasts using the drift method and plot them.

```
fb_fit<- fb %>%
  model(RW(Close ~ drift()))

fb_fc<- fb_fit %>%
  forecast(h = 720)

fb_fc %>%
  autoplot(fb, level = NULL) +
  labs(title = 'fb stock random walk forecast', x='Trading days', y='Closing price')+
  guides(color = guide_legend(title = 'RW forecast'))
```



c) Show that the forecasts are identical to extending the line drawn between the first and last observations.

```
first_point<- fb %>% slice(1)
last_point<- fb %>% slice(n())

straight_line<- bind_rows(first_point, last_point)

fb %>%
  ggplot(aes(x = day, y = Close)) +
  geom_line(color = 'black') +  # Plot the historical data
  geom_line(data = straight_line, aes(x = c(1, nrow(fb)), y = c(first_point$Close, last_point$Close)),
            color = 'red', linetype = "dashed") +  # Draw the straight line
  geom_line(data = fb_fc, aes(x=day, y=.mean), color='blue')+ #adding the forecast line
  labs(title = 'Facebook Stock Price with RW drift forecast method',
       x = 'Trading Days', y = 'Closing Price')
```



d) Try using some of the other benchmark functions to forecast the same data set. Which do you think is best? Why?

Ans: SNAIVE could not produce a forecast due to lack of seasonality. I believe the best benchmark tool to use for this data is the RW since it captures the growing trend of the fb stock in a long period of time. the Naive method is only applicable for companies that are stagnant.

```
fb_fit2<- fb %>%
  model(
    naive = NAIVE(Close),
    snaive = SNAIVE(Close)
  )
```

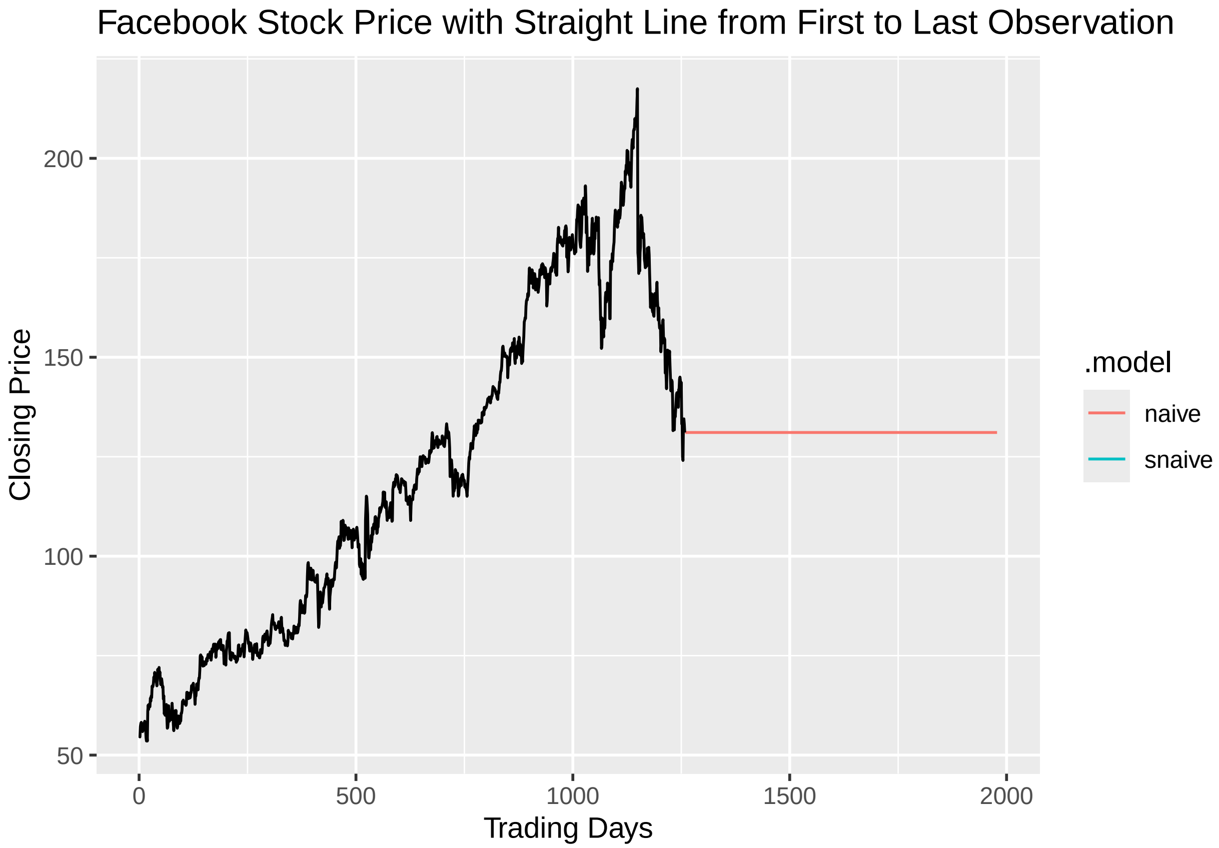
```
## Warning: 1 error encountered for snaive
## [1] Non-seasonal model specification provided, use RW() or provide a different lag specification.
```

```
fb_fit2
```

```
## # A mable: 1 x 2
##      naive       snaive
##    <model>      <model>
## 1 <NAIVE> <NULL model>
```

```
fb_fc2<- fb_fit2 %>% forecast(h=720)

fb_fc2 %>%
  autoplot(fb, level = NULL)+
  labs(title = 'Facebook Stock Price with Straight Line from First to Last Observation',
       x = 'Trading Days', y = 'Closing Price')
```

```
## Warning: Removed 720 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```



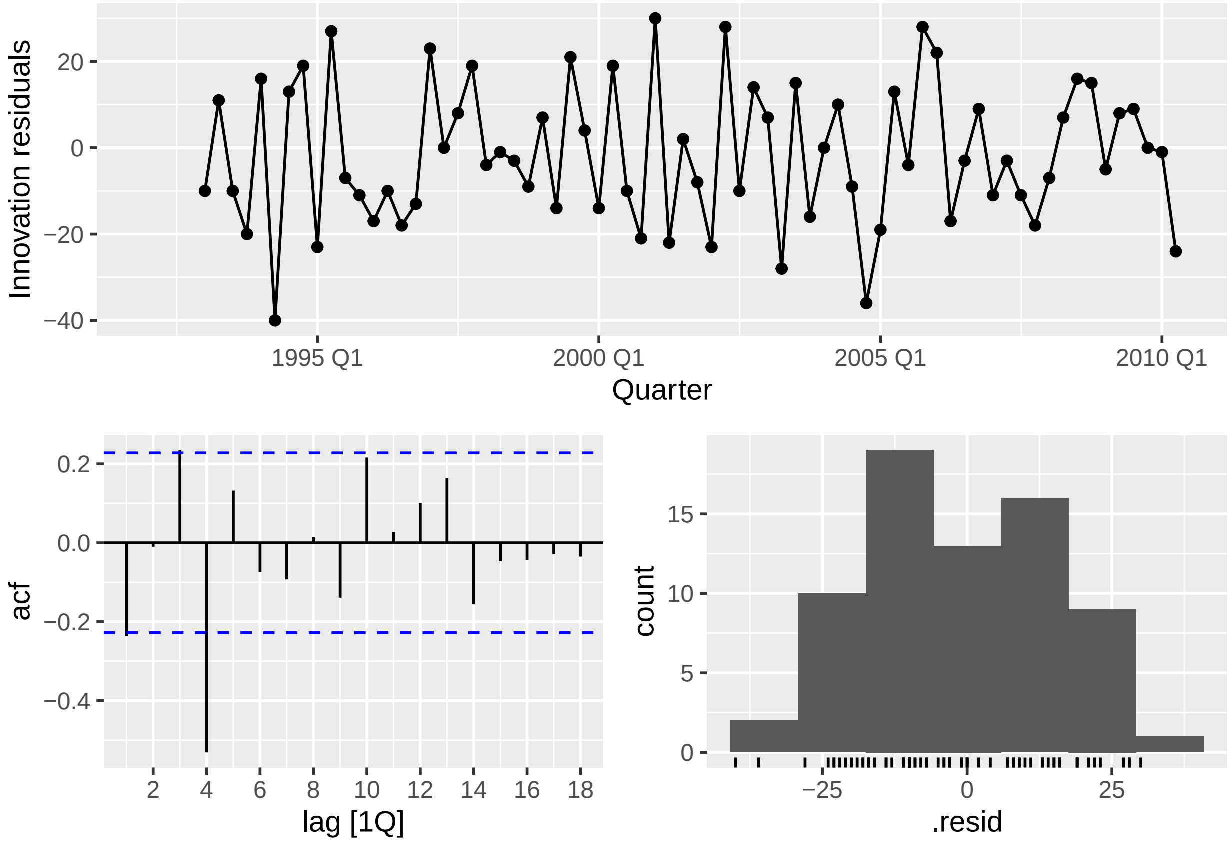
5.3 Apply a seasonal naïve method to the quarterly Australian beer production data from 1992. Check if the residuals look like white noise, and plot the forecasts. The following code will help. What do you conclude?

Ans: The residuals look mostly center around the 0, indicating not much of a sudden peak or trough for outliers. The ACF plot almost confirms majority of the lags are in range with one except at lag 4. The residual histogram also shows a relatively normal distribution. This is fair to conclude that the residuals are most likely white-noise.

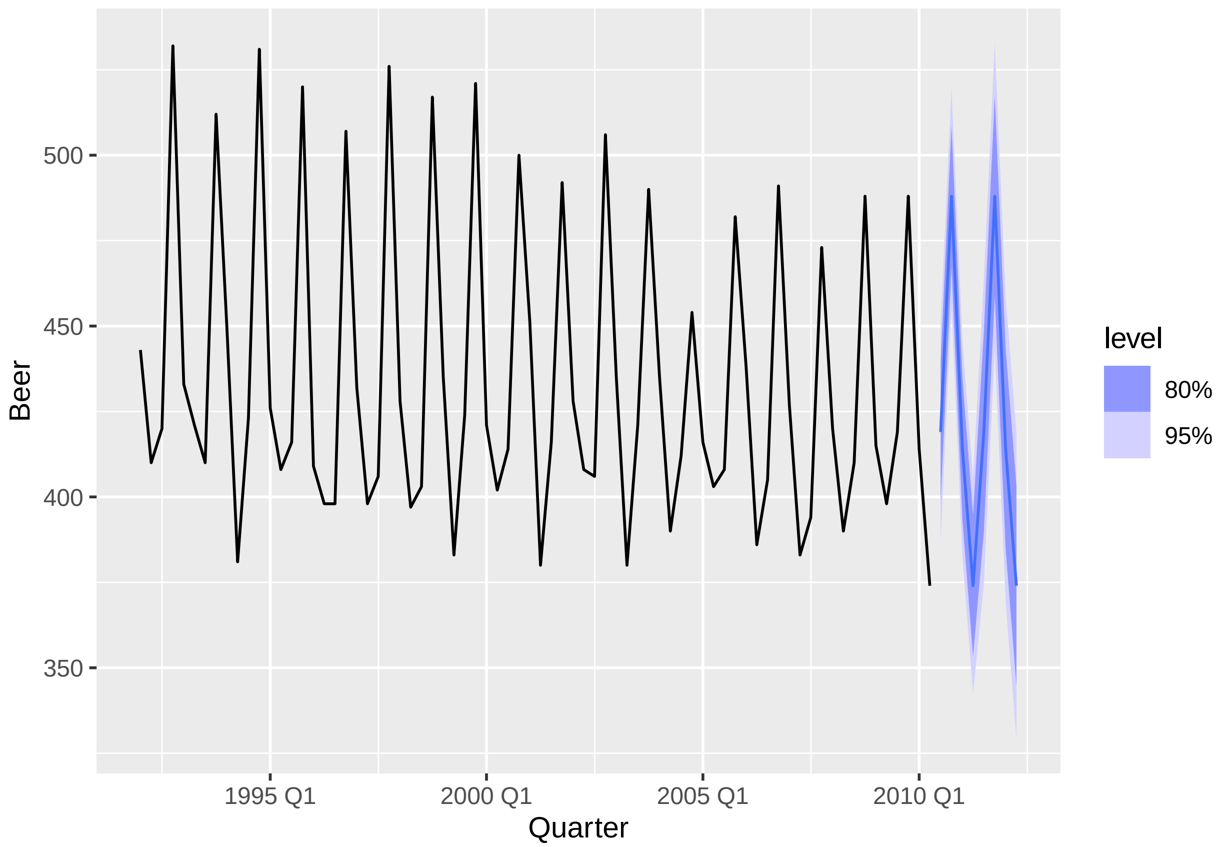
```
recent_production <- aus_production |>
  filter(year(Quarter) >= 1992)
fit <- recent_production |> model(SNAIVE(Beer))
# Look at the residuals
fit |> gg_tsresiduals()
```

```
## 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()`).
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



```
# Look a some forecasts
fit |> forecast() |> autoplot(recent_production)
```



5.4 Repeat the previous exercise using the Australian Exports series from global_economy and the Bricks series from aus_production. Use whichever of NAIVE() or SNAIVE() is more appropriate in each case.

Australian Exports series from global_economy:

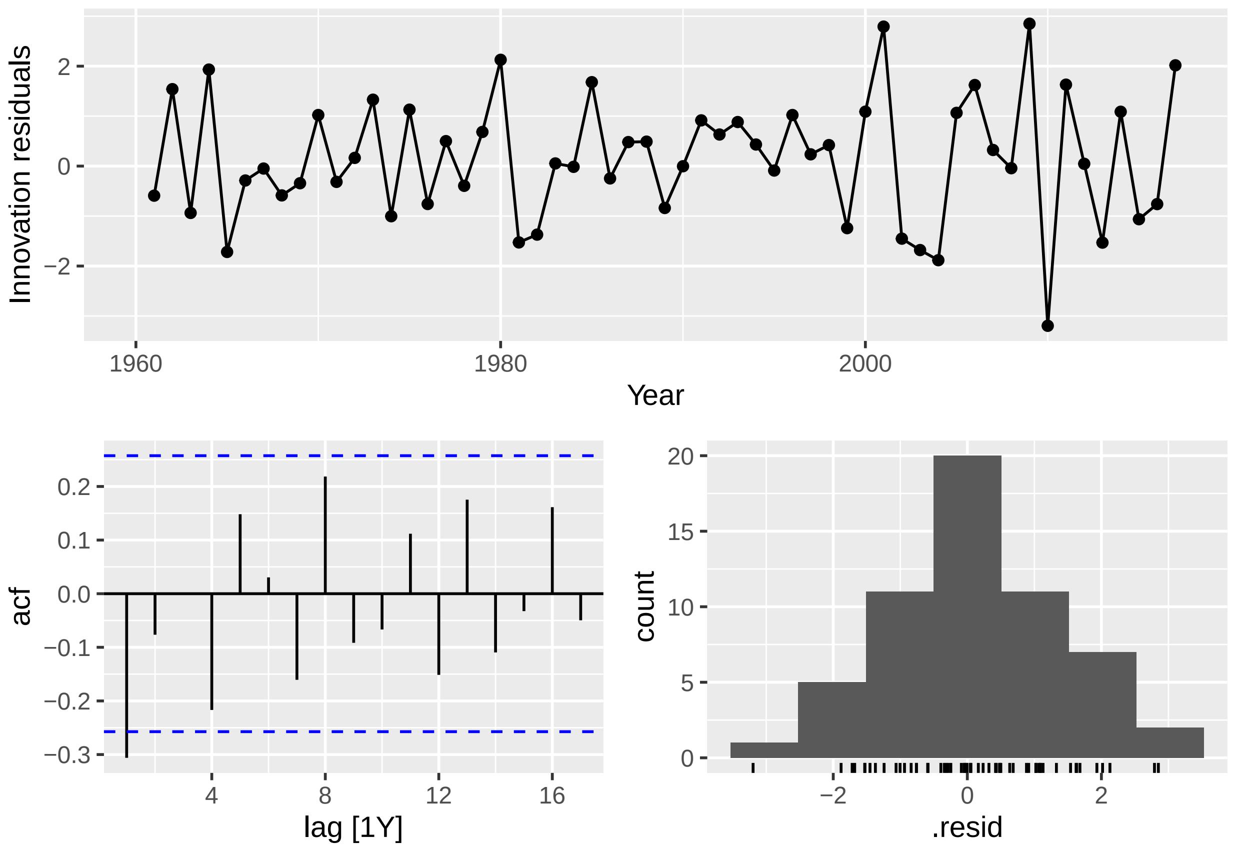
Ans: NAIVE method is used since seasonality is lacking. The residuals are normally distributed and the residual plots revolve around 0. The Ljung_box test also shows p-value > 0.05, indicating no autocorrelation observed. Variations are due to white-noise only.

```r
aus_export<- global_economy %>%
  filter(Country =='Australia') %>%
  select(Year, Exports)

aus_fit<- aus_export %>%
  model(naive=NAIVE(Exports))

aus_fit %>% gg_tsresiduals()
```

```
## 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()‘).
```

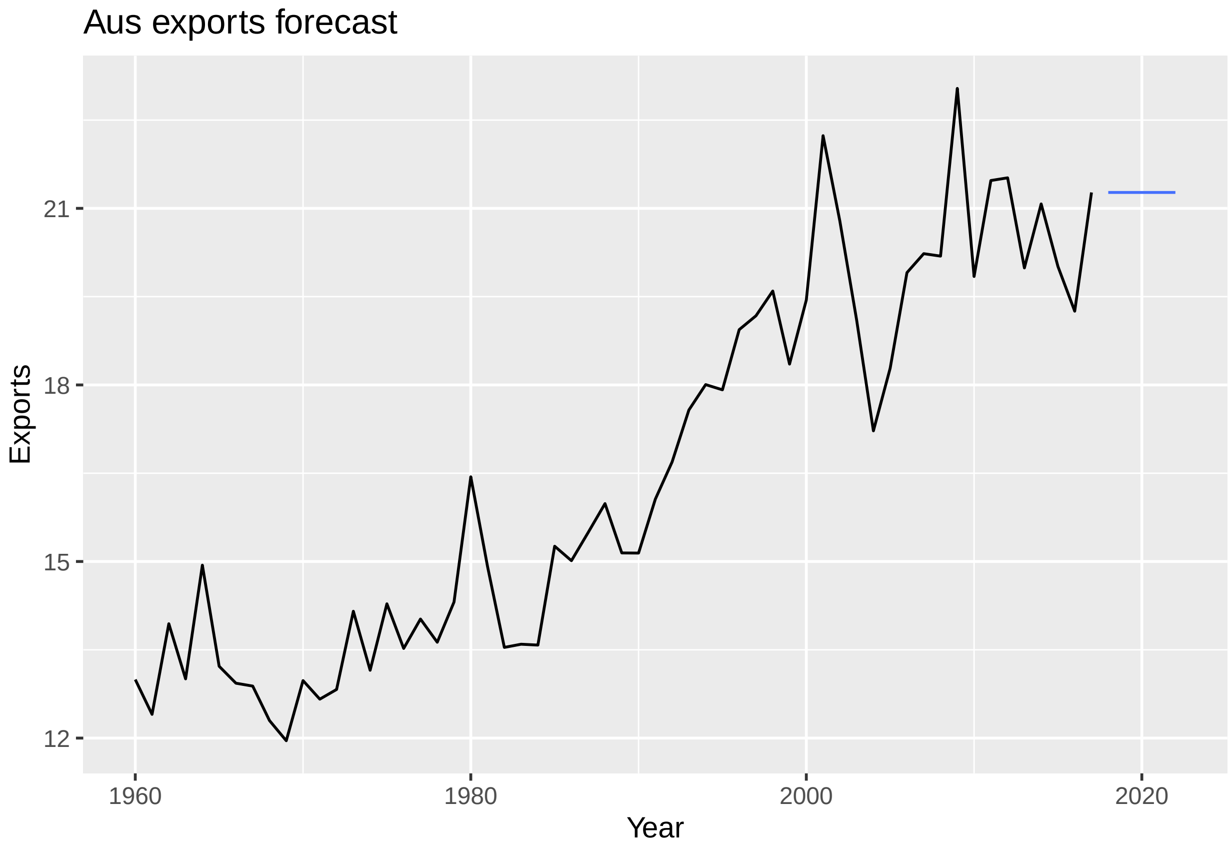
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).
```



```
#check the p-value
augment(aus_fit) %>% features(.innov, ljung_box, lag=10)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 naive     16.4    0.0896
```

```
#plotting the graph
aus_fit %>% forecast(h='5 year') %>%
  autoplot(aus_export, level=NULL)+
  labs(title = 'Aus exports forecast')
```



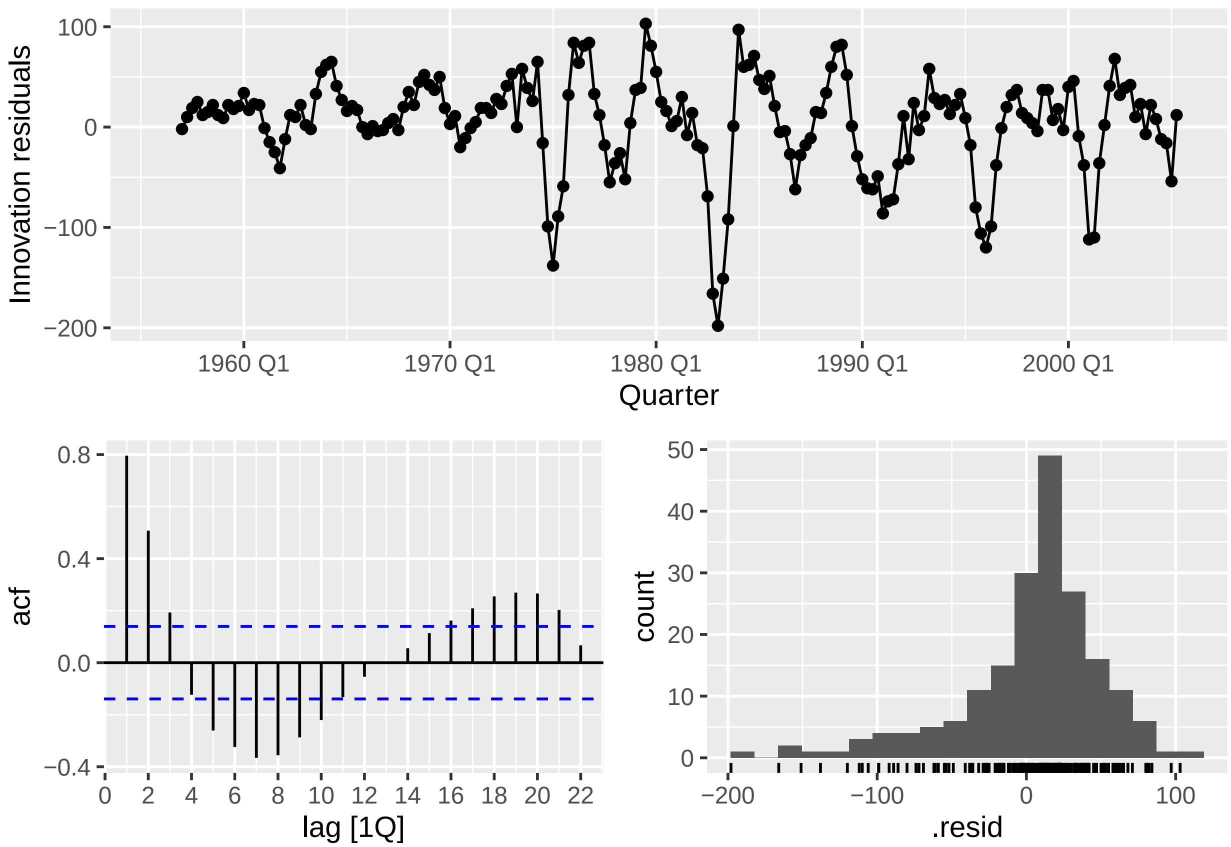
Bricks series from aus_production Ans: SNAIVE is picked due to complex seasonality is seen. The diagnostic plots show that the residuals are left skewed from the histogram. ACF plot shows seasonal pattern of autocorrelation, and the residual plot also shows outliers. The residuals from this fitting are unlikely to be due to white noise. The ljung_box test also confirms with a near or 0 p-value for the residual autocorrelation.

```
aus_brick<- aus_production %>%
  select(Quarter, Bricks) %>%
  filter(!is.na(Bricks))

aus_brick_fit<- aus_brick %>%
  model(Snavie = SNAIVE(Bricks))
#diagnostic plots:
aus_brick_fit %>% gg_tsresiduals()
```

```
## 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()‘).
```

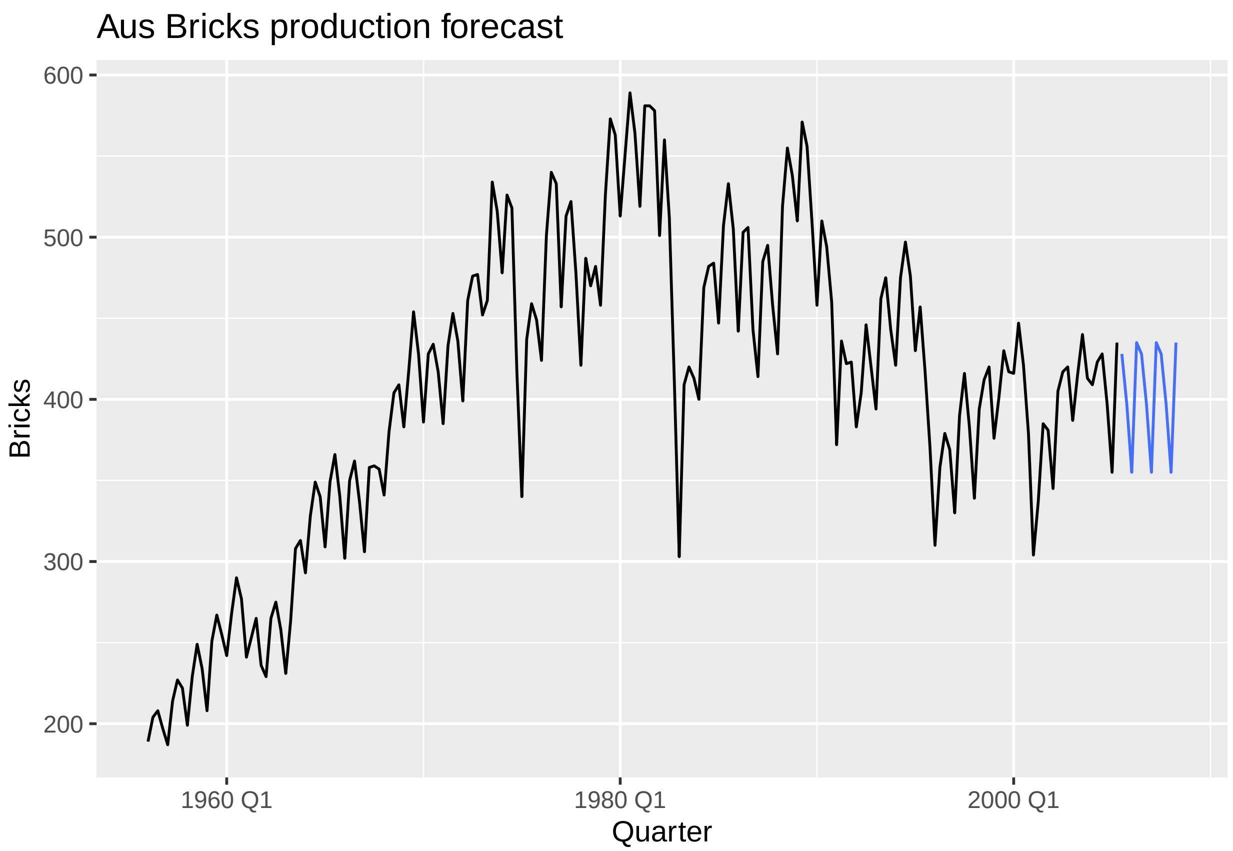
```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



```
#ljung_box test:
augment(aus_brick_fit) %>% features(.innov, ljung_box, lag=10)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 Snavie    301.         0
```

```
#Forecasted another 12 quarters
aus_brick_fc<- aus_brick_fit %>%
  forecast(h = 12)

aus_brick_fc %>%
  autoplot(aus_brick, level = NULL)+
  labs(title = 'Aus Bricks production forecast')+
  guides(color = guide_legend(title = 'forecast'))
```

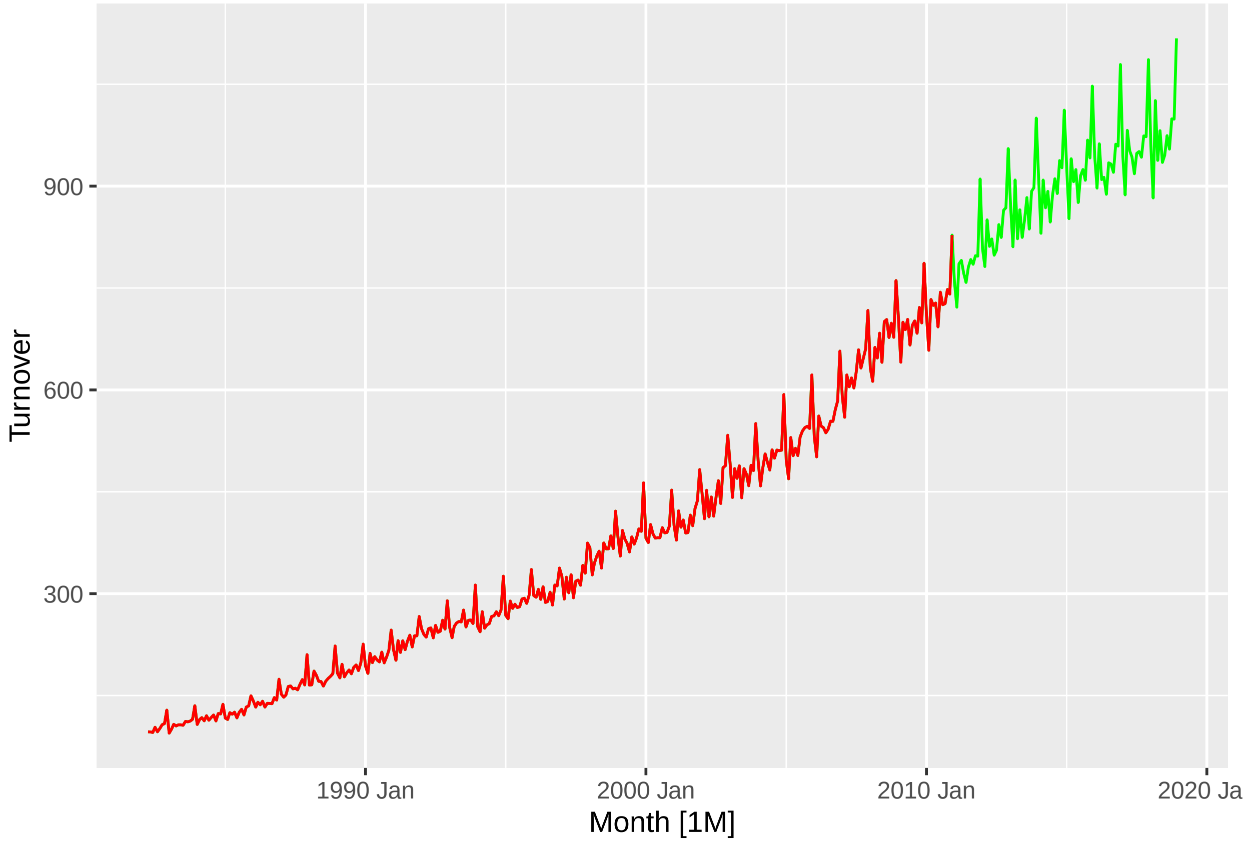


5.7 For your retail time series (from Exercise 7 in Section 2.10): a) Create a training dataset consisting of observations before 2011 using

```
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))

myseries_train <- myseries |>
  filter(year(Month) < 2011)
```

b) Check that your data have been split appropriately by producing the following plot.

```
autoplot(myseries, Turnover, color='green') +
  autolayer(myseries_train, Turnover, color= 'red')
```



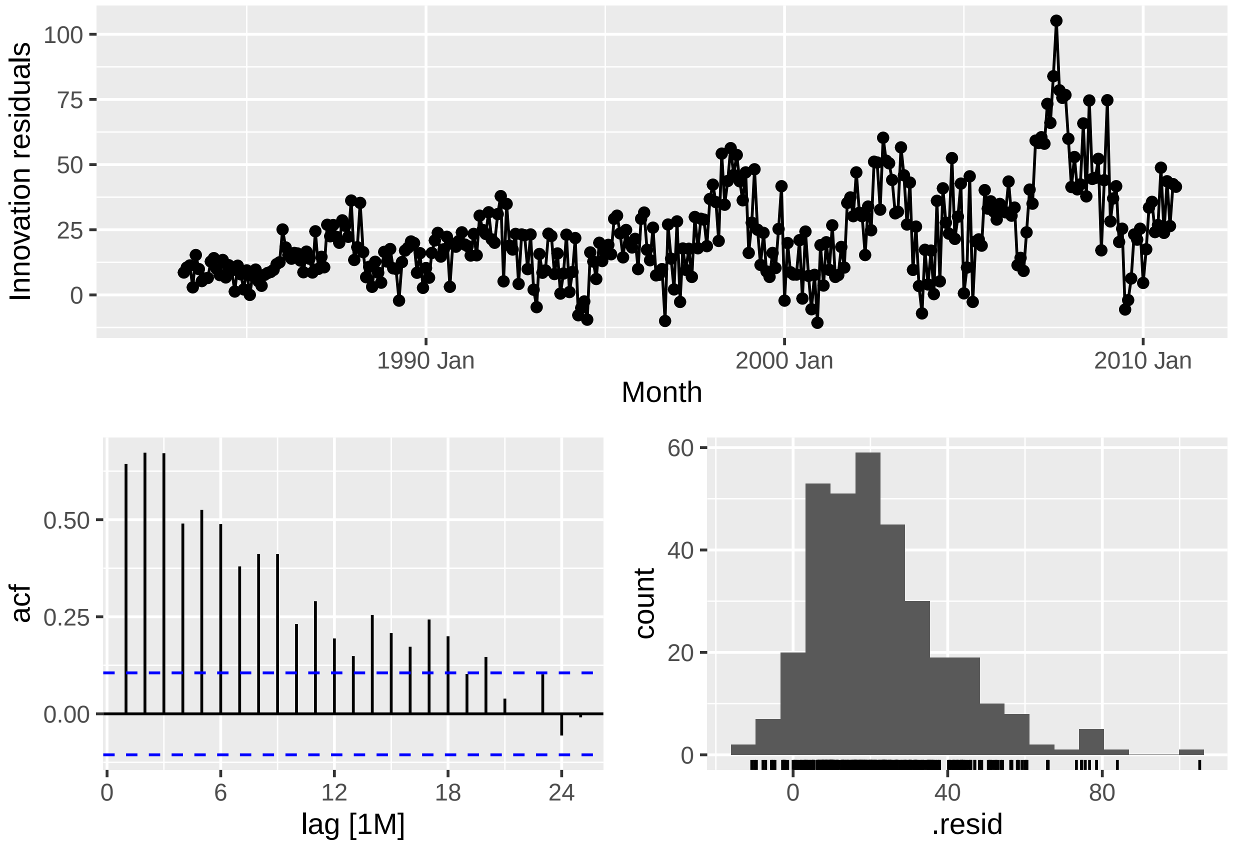
c) Fit a seasonal naïve model using SNAIVE() applied to your training data (myseries_train).

```
fit <- myseries_train |>
  model(snaive = SNAIVE(Turnover))
```

d) Check the residuals. Do the residuals appear to be uncorrelated and normally distributed?

Ans: The residuals look correlated and are not normally distributed.

```
fit %>%
  gg_tsresiduals()
```

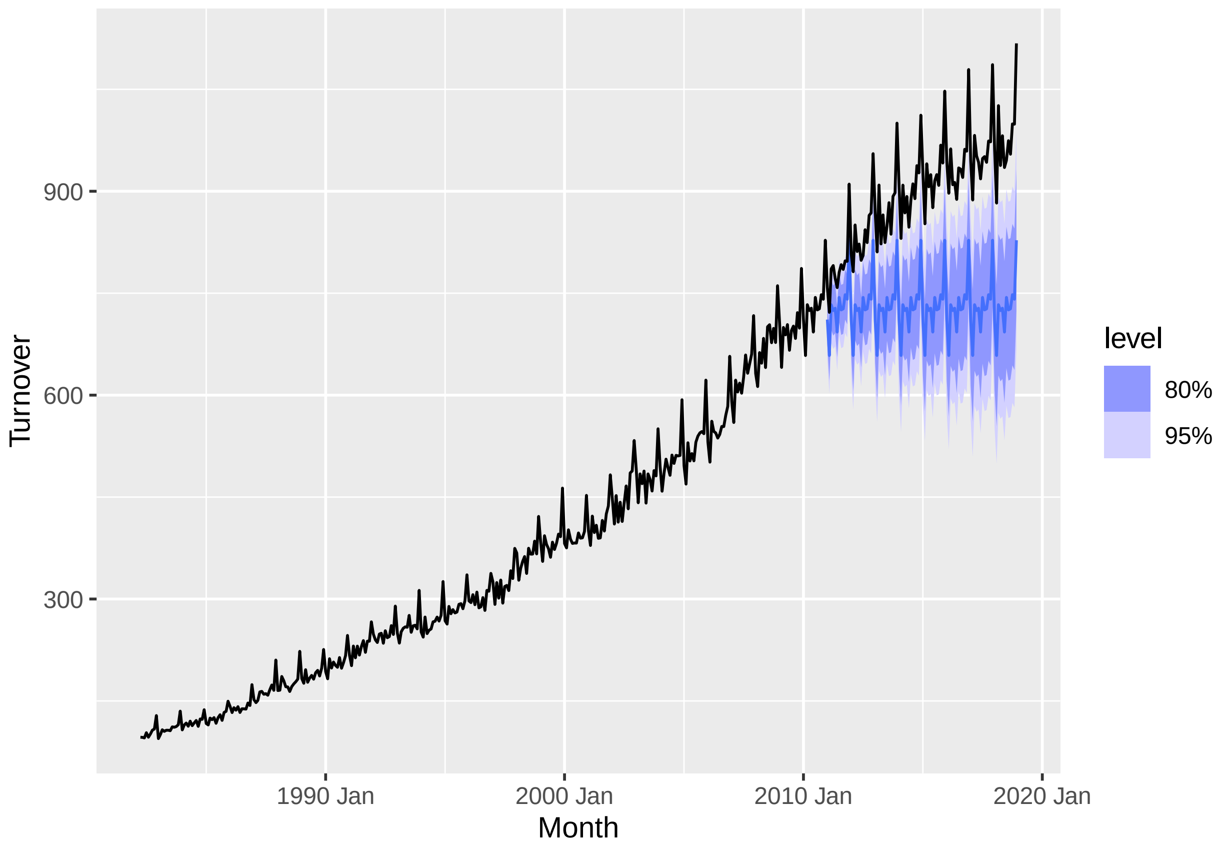


e) Produce forecasts for the test data

```
fc_data<- myseries %>%
  filter(year(Month)>2011)

fc<- fit %>%
  forecast(new_data = anti_join(myseries, myseries_train))
```

```
## Joining with ‘by = join_by(State, Industry, ‘Series ID‘, Month, Turnover)‘
```

```
fc %>% autoplot(myseries)
```



f) Compare the accuracy of your forecasts against the actual values.

Ans: The forecasted data is not very accurate when compared to the actual data. First, the plot shows almost no overlapping region between the observed data(black lines) and the forecasted data(purple). Second, all the forecast measurements are very big numbers when compared to the training data measurements. The SNAIVE method does not fit well with the model presented.

```
fit %>%  accuracy() #training
```

```
## # A tibble: 1 x 12
##   State     Industry .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>     <chr>    <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Western~ Superma~ snaive Trai~  22.6  28.7  23.1  6.73  6.88     1     1 0.644
```

```
fc %>% accuracy(myseries) #forecast value
```

```
## # A tibble: 1 x 12
##   .model State    Industry .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr>    <chr>    <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 snaive Western~ Superma~ Test   170.  182.  170.  18.5  18.5  7.37  6.33 0.874
```

g) How sensitive are the accuracy measures to the amount of training data used?

Ans: The sensitivity of the accuracy measurement is relative important to the amount of training data used. Typically accuracy improves as the amount of training data is available for reference. In this case, the forecasting method used is of benchmark purposes and the amount of training data is slightly independent of the outcome due to how the SNAIVE method takes the last seasonality patterns and replicate them. In real forecasting models, the amount of training data will impact the accuracy of the model more significantly.