DATA624_Homework3

John Ferrara

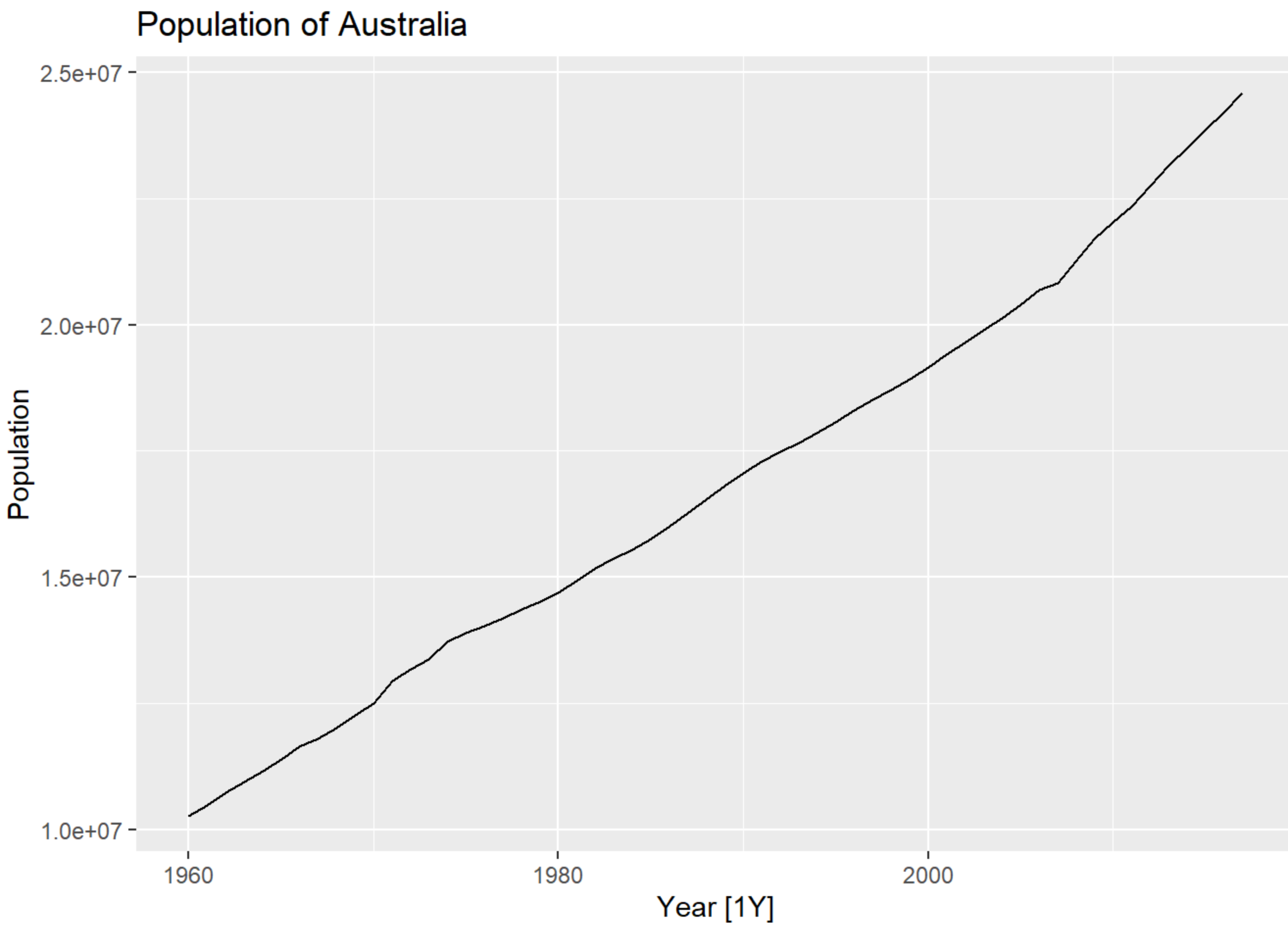
2025-02-22

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

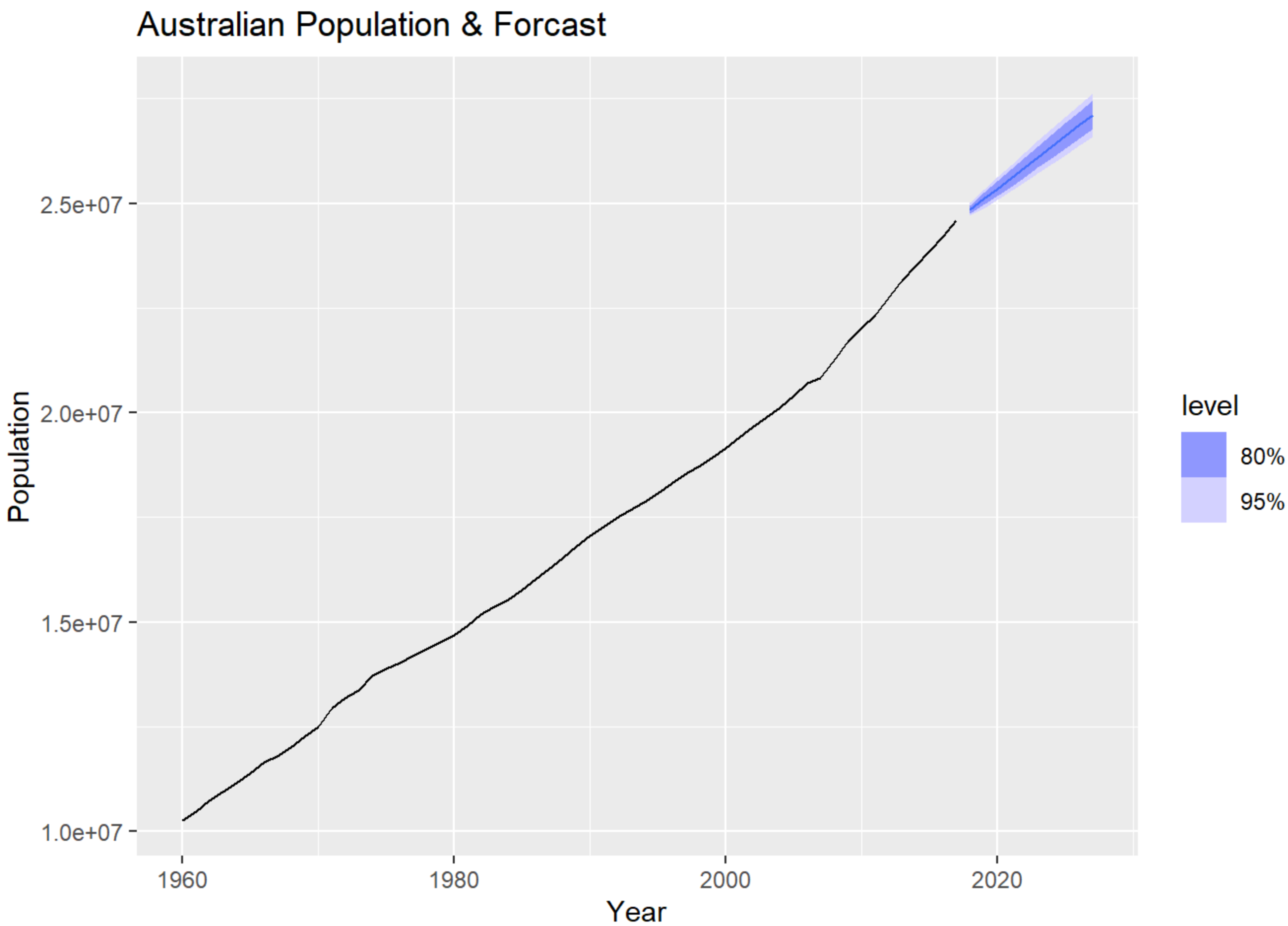
- a. Australian Population (global_economy)
- b. Bricks (aus_production)
- c. NSW Lambs (aus_livestock)
- d. Household wealth (hh_budget).
- e. Australian takeaway food turnover (aus_retail).

Question 1 Answer:

```
## Population of AUS projection with DRIFT method, as its annual data (no seasons) and last o
bserved val considered with avg change over time seems most appropriate.

## Country Lim
aus_pop <- global_economy |> filter(Country == "Australia")

## Plotting
autoplot(aus_pop, Population)+ labs(title="Population of Australia")
```

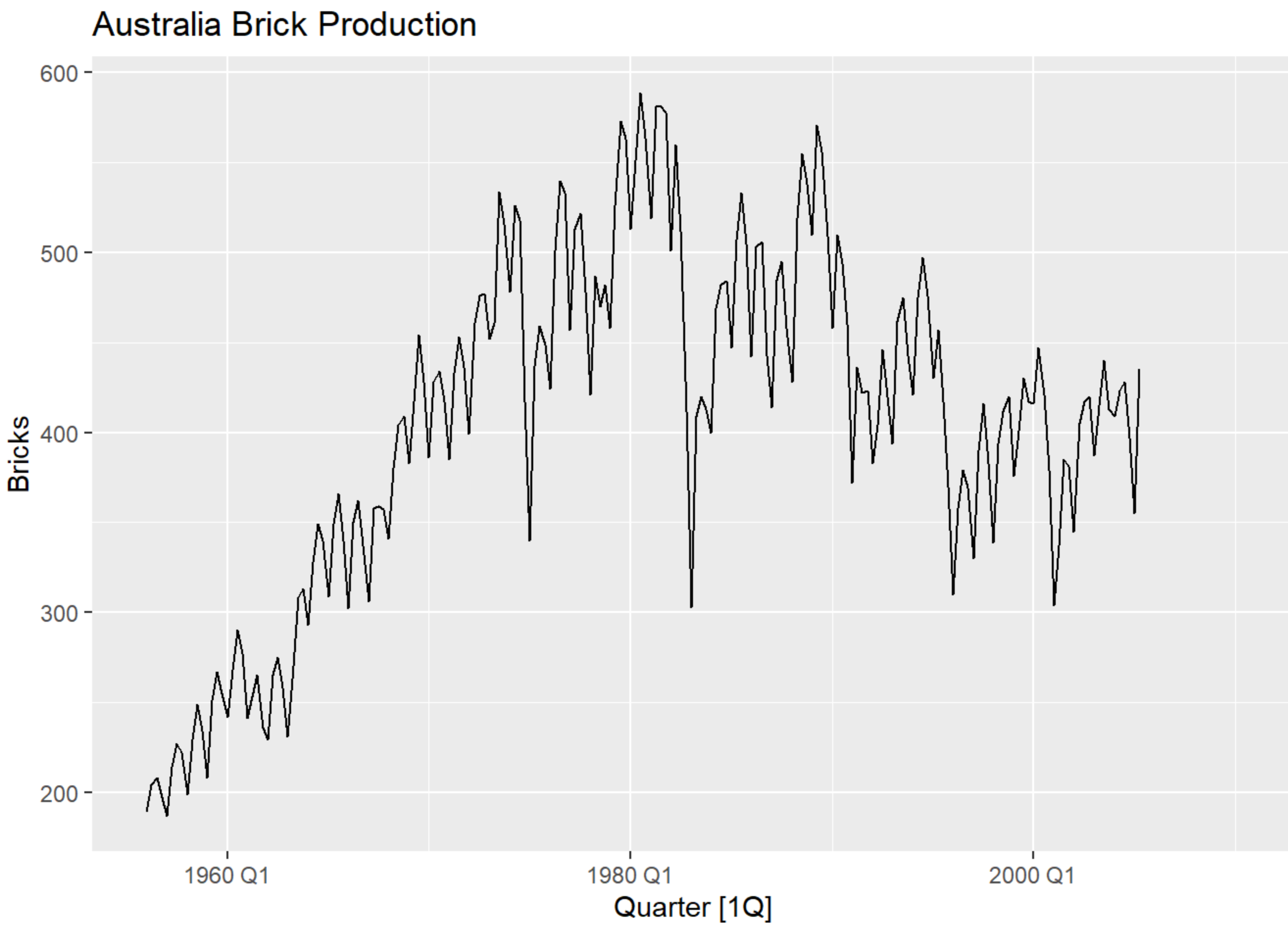



```
## Making mable and fable
aus_pop_pred <- aus_pop |>
  ##Ensuring pop values
  filter(!is.na(Population))|>
  model(Drift = RW(Population ~ drift()))

drift_pop <- aus_pop_pred |> forecast(h = "10 years")

## Plotting the benchmark forcast
drift_pop |>
  autoplot(aus_pop) +
  labs(title="Australian Population & Forcast")
```

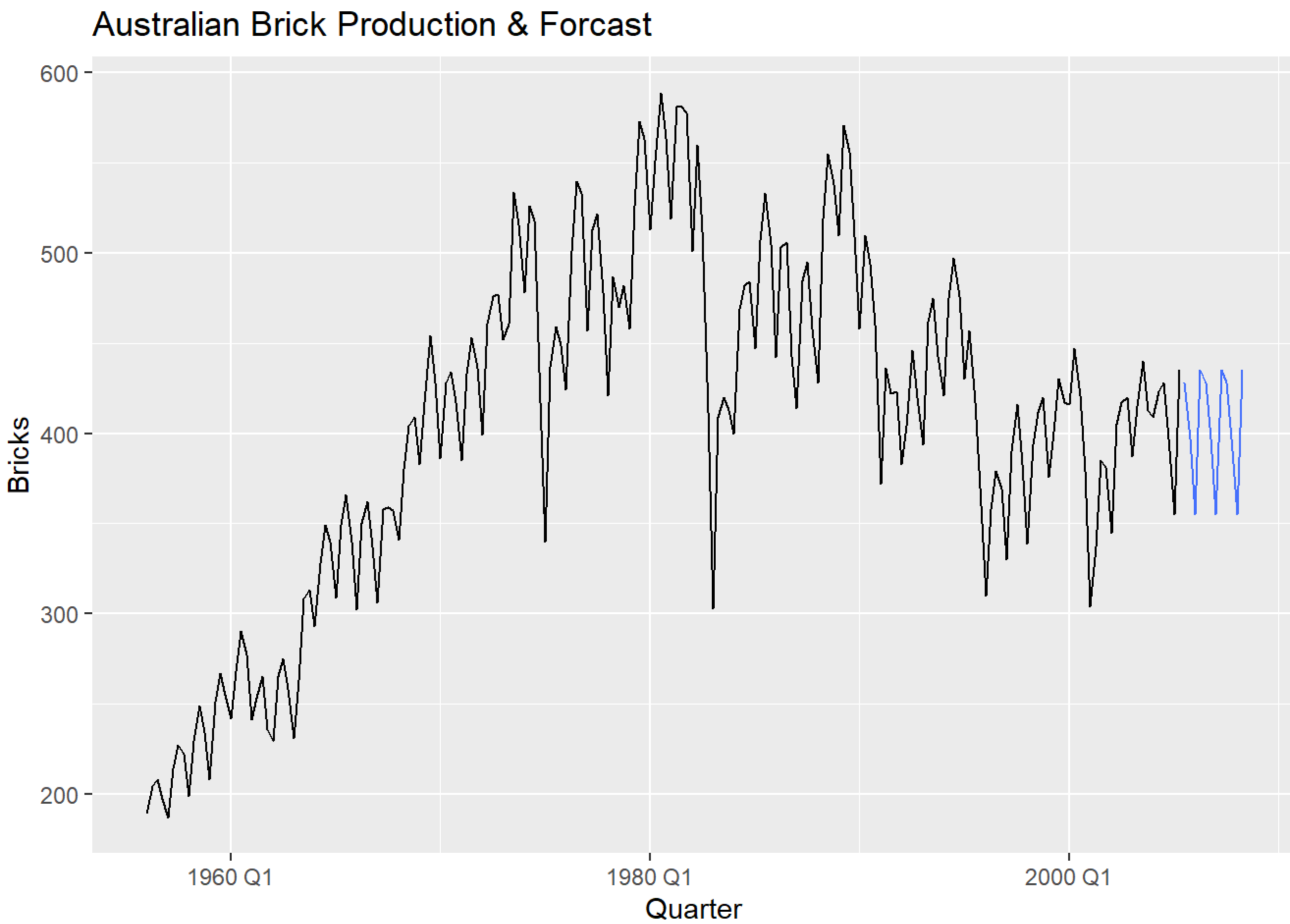



```
##Plotting brick production with SNAIVE method, as the seasonality of data makes it seem most appropriate.

## Pltoting Original Data
autoplot(aus_production, Bricks)+ labs(title="Australia Brick Production ")
```

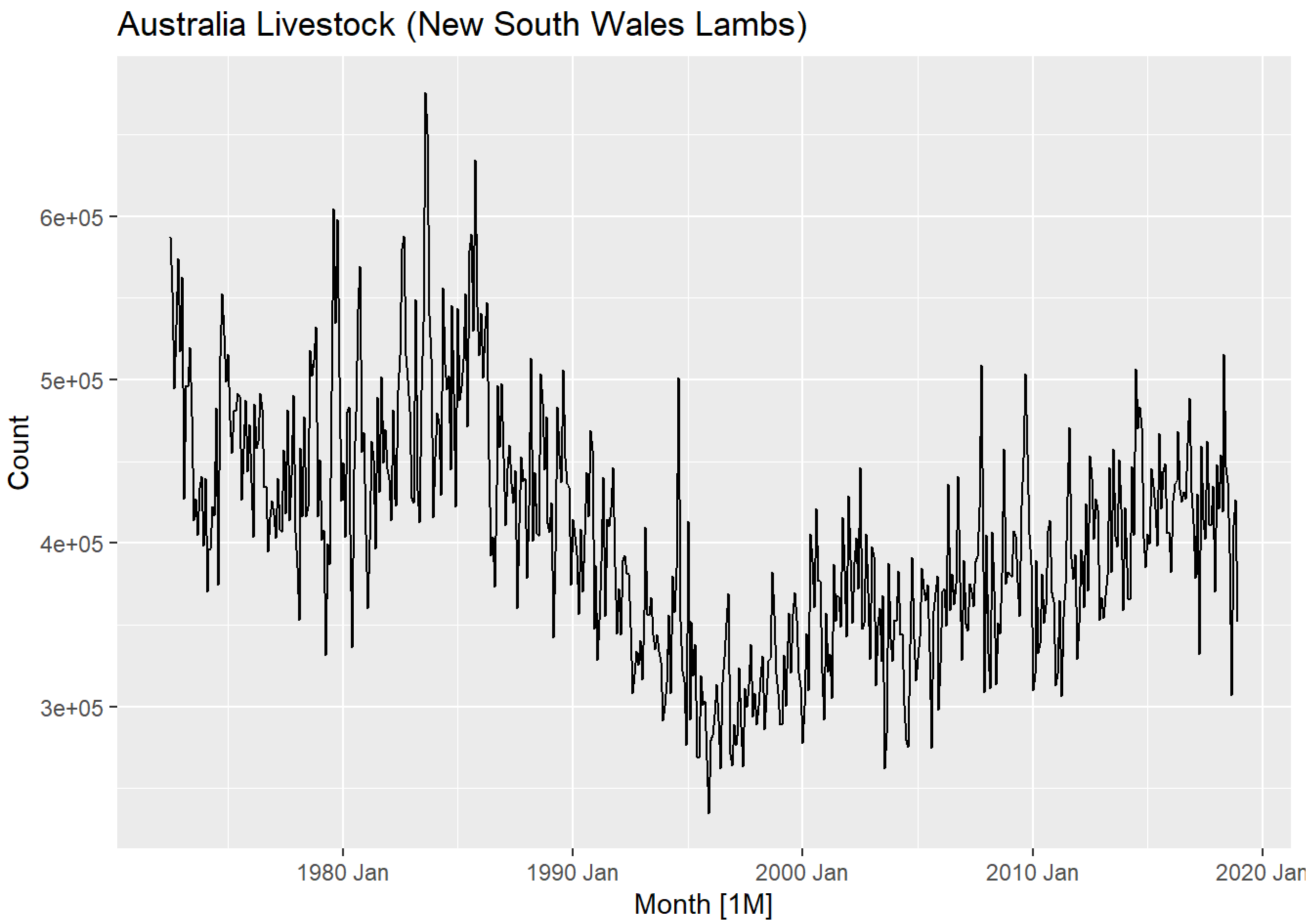
```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



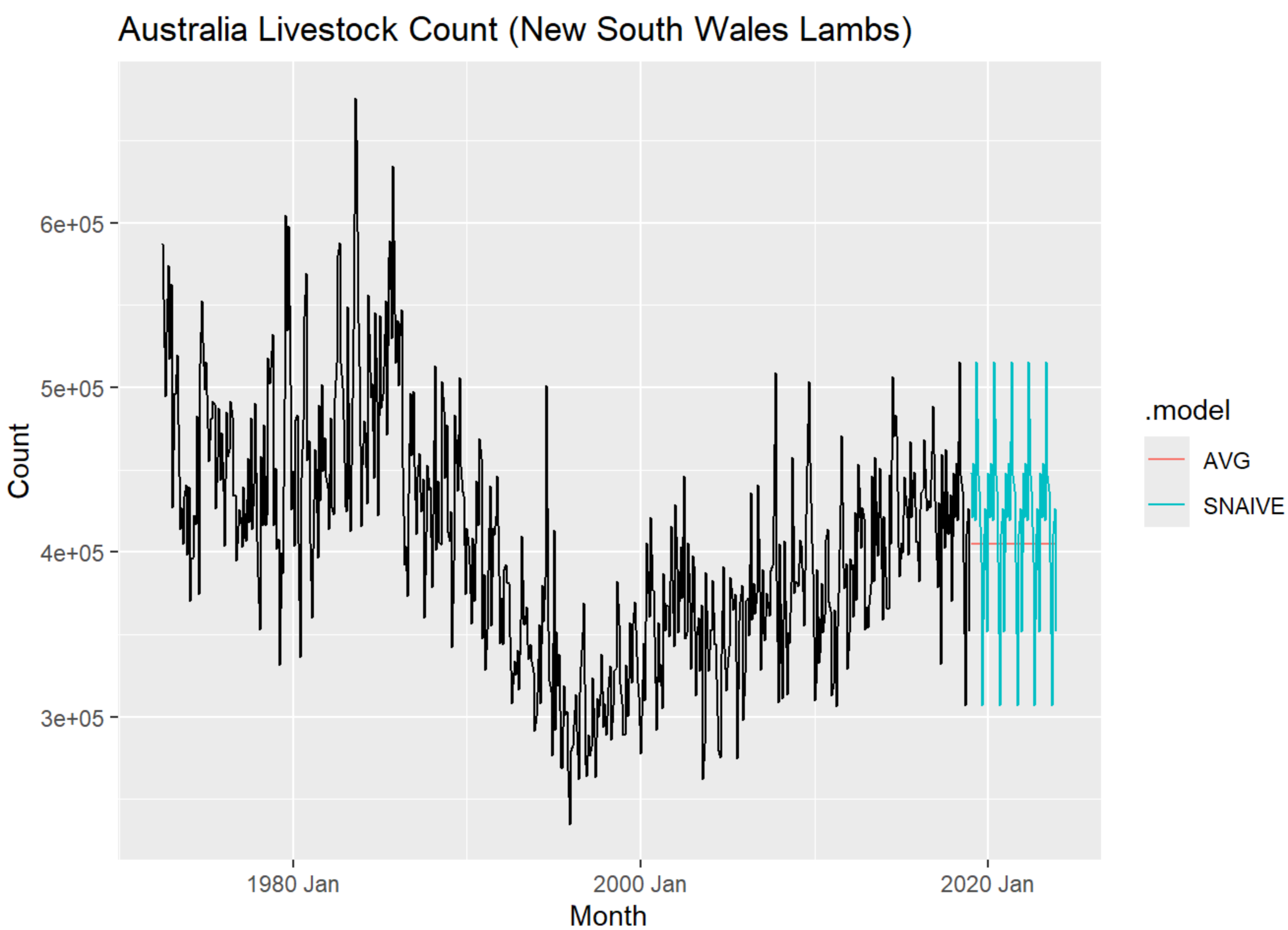

```
## Limiting for values
brick_prod <- aus_production |> filter(!is.na(Bricks))

## Model and forecast
brick_model <- brick_prod |> model(SNAIVE = SNAIVE(Bricks))
brick_fable <- brick_model |> forecast(h = "3 years")

## Plotting the benchmark forcast
brick_fable |>
  autoplot(brick_prod, level=NULL) + ## Removing C.I because it looks messy
  labs(title="Australian Brick Production & Forcast")
```



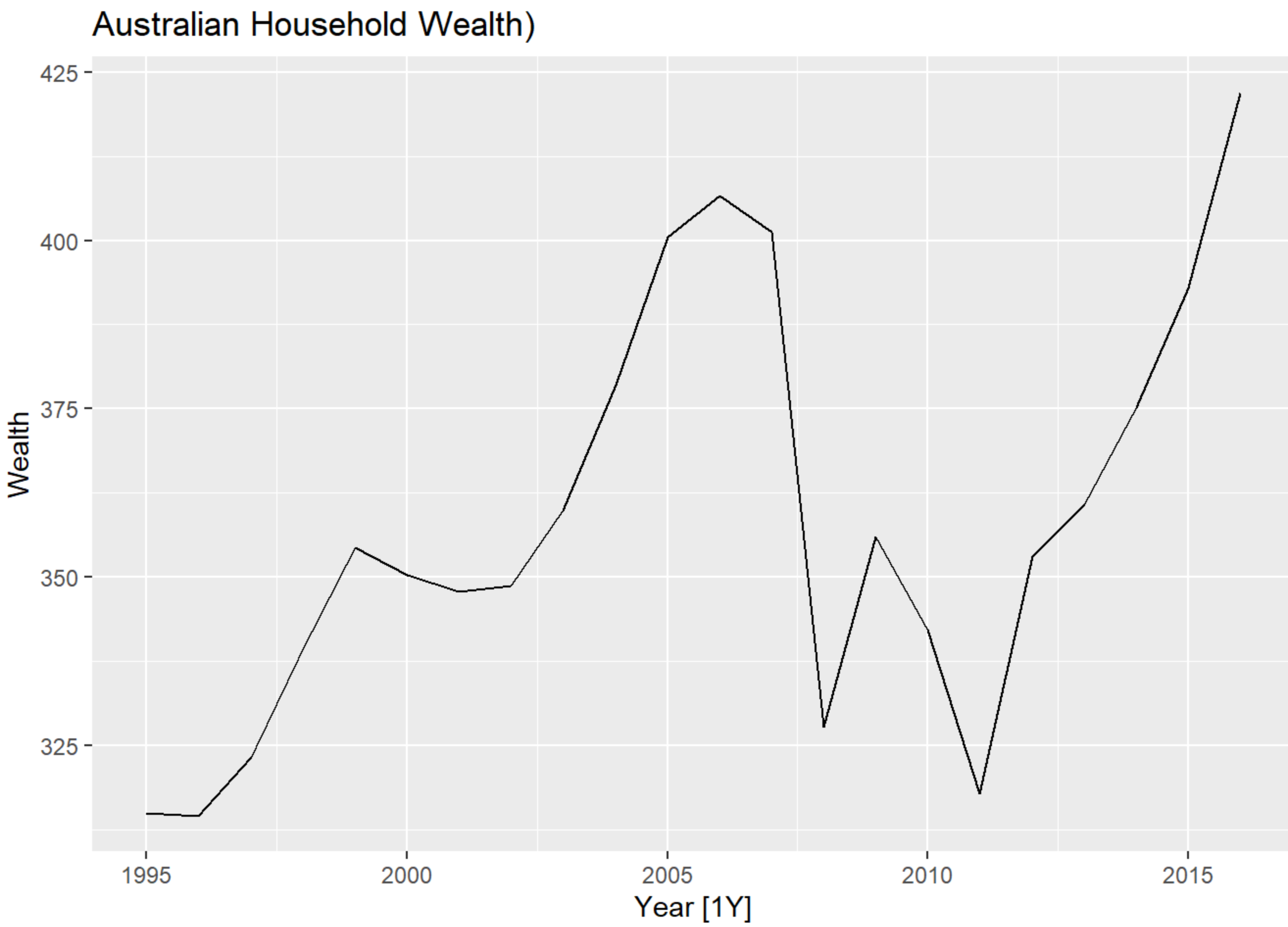

```
## Monthly data so SNAIVE may be the most ideal lets take a look.

##unique(aus_livestock$Animal) #lambs
##unique(aus_livestock$State) #New South Wales

## Limiting to relevant data
nsw_lambs<- aus_livestock |> filter(Animal == "Lambs",
                                State == "New South Wales")
## Plotting original data
autoplot(nsw_lambs, Count)+ labs(title="Australia Livestock (New South Wales Lambs) ")
```



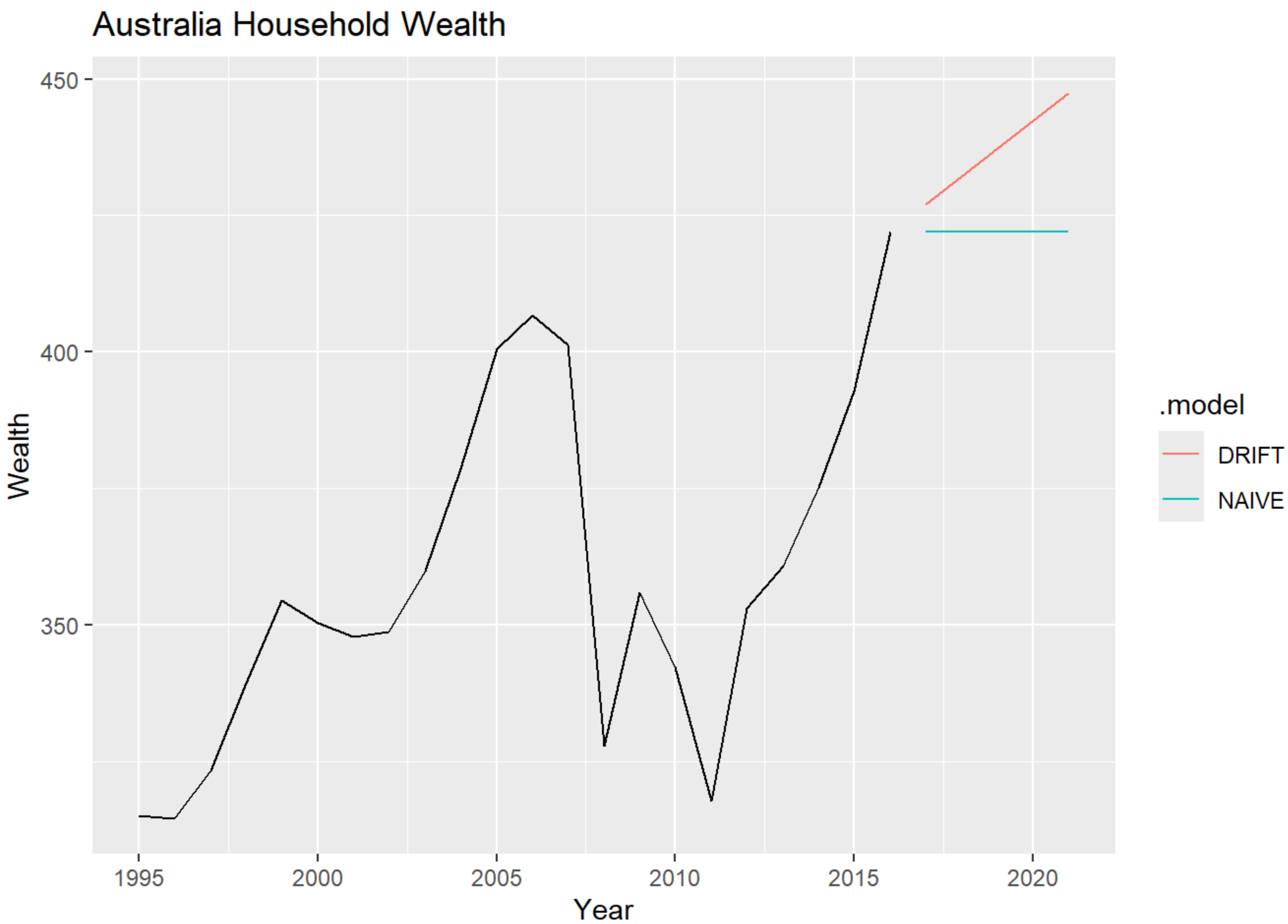

```
## Getting Fitted Model Values. USing Naive and Mean methods.
lamb_fit <- nsw_lambs |> filter(!is.na(Count))|>
  model(AVG = MEAN(Count),  SNAIVE=SNAIVE(Count))

lamb_forecast <- lamb_fit |> forecast(h="5 years")

## Plotting Results
lamb_forecast |>
  autoplot(nsw_lambs, level=NULL) + ## Removing C.I because it looks messy
  labs(title="Australia Livestock Count (New South Wales Lambs)")
```




```
## Household Wealth
lim_budget <- hh_budget |> filter(!is.na(Wealth), Country=="Australia")

## Plotting Original Data
autoplot(lim_budget, Wealth)+ labs(title="Australian Household Wealth) ")
```




```
## No Seasonal Granularity, using NAIVE and DRIFT becuase of the difference in the latest dat
a.

## Getting Fitted Model Values.
wealth_fit <- lim_budget |> model(DRIFT = RW(Wealth ~ drift()),  NAIVE=NAIVE(Wealth))

wealth_forecasts <- wealth_fit |> forecast(h="5 years")

## Plotting Results
wealth_forecasts |>
  autoplot(lim_budget, level=NULL) + ## Removing C.I because it looks messy
  labs(title="Australia Household Wealth")
```



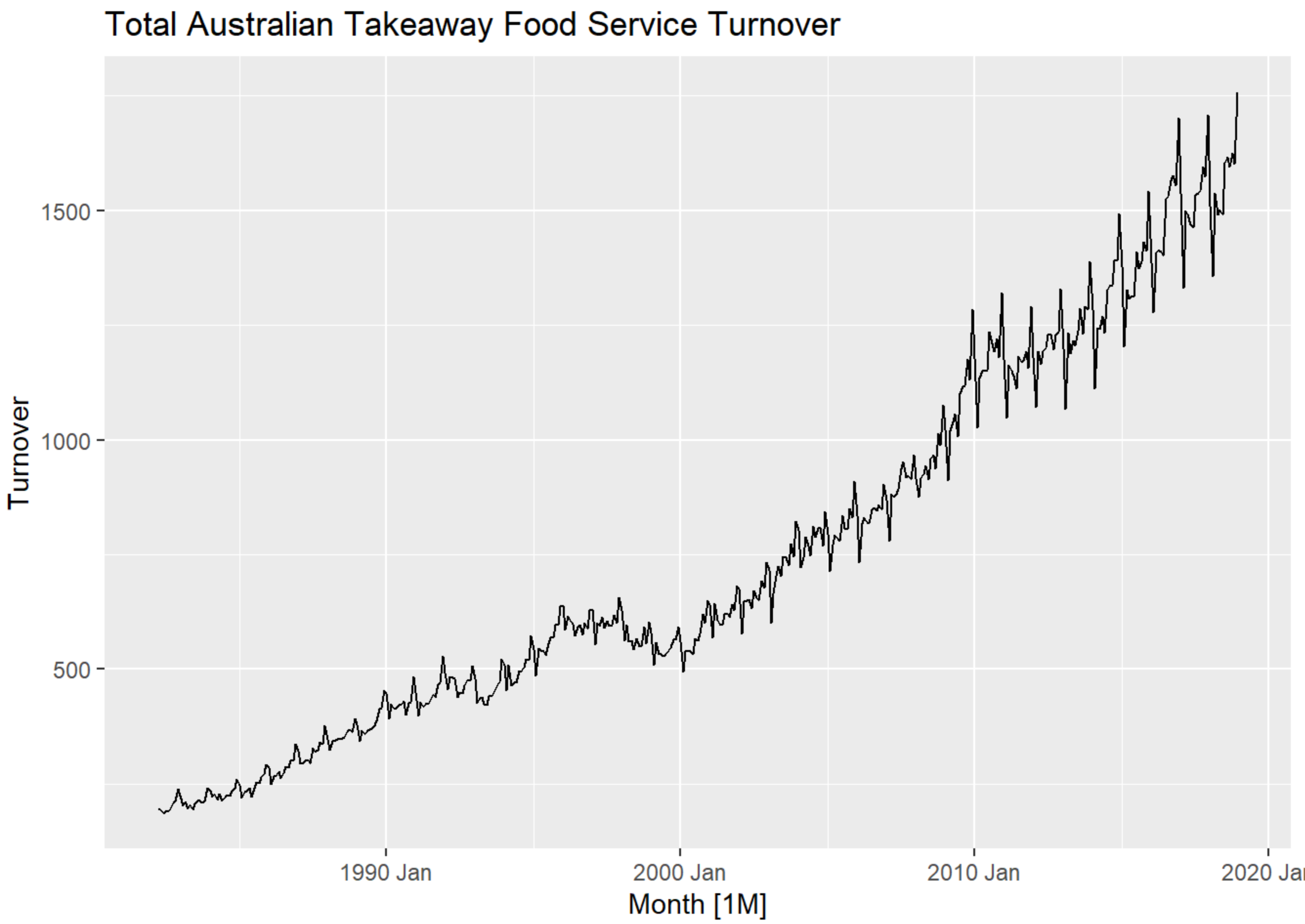

```
### First look is that its monthly Data
unique(aus_retail$Industry)
```

```
##  [1] "Cafes, restaurants and catering services"
##  [2] "Cafes, restaurants and takeaway food services"
##  [3] "Clothing retailing"
##  [4] "Clothing, footwear and personal accessory retailing"
##  [5] "Department stores"
##  [6] "Electrical and electronic goods retailing"
##  [7] "Food retailing"
##  [8] "Footwear and other personal accessory retailing"
##  [9] "Furniture, floor coverings, houseware and textile goods retailing"
## [10] "Hardware, building and garden supplies retailing"
## [11] "Household goods retailing"
## [12] "Liquor retailing"
## [13] "Newspaper and book retailing"
## [14] "Other recreational goods retailing"
## [15] "Other retailing"
## [16] "Other retailing n.e.c."
## [17] "Other specialised food retailing"
## [18] "Pharmaceutical, cosmetic and toiletry goods retailing"
## [19] "Supermarket and grocery stores"
## [20] "Takeaway food services"
```

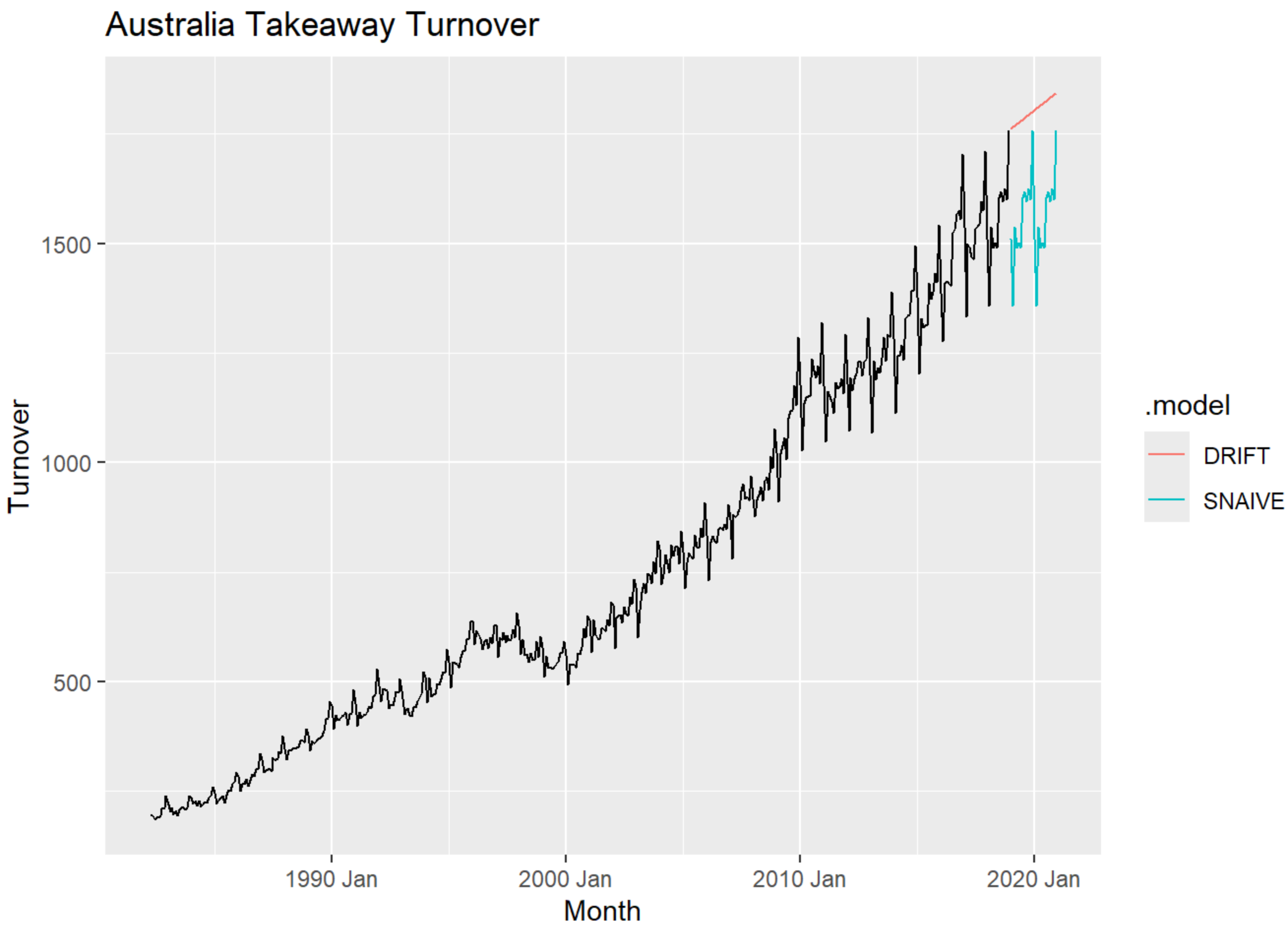
```
aus_retail
```

```
## # A tsibble: 64,532 x 5 [1M]
## # Key:       State, Industry [152]
##    State                        Industry            `Series ID`    Month Turnover
##    <chr>                        <chr>               <chr>          <mth>    <dbl>
##  1 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Apr      4.4
##  2 Australian Capital Territory Cafes, restaurant… A3349849A   1982 May      3.4
##  3 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Jun      3.6
##  4 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Jul      4  
##  5 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Aug      3.6
##  6 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Sep      4.2
##  7 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Oct      4.8
##  8 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Nov      5.4
##  9 Australian Capital Territory Cafes, restaurant… A3349849A   1982 Dec      6.9
## 10 Australian Capital Territory Cafes, restaurant… A3349849A   1983 Jan      3.8
## # ℹ 64,522 more rows
```

```
## Processing to sum up for country over time
lim_aus_retail <- aus_retail |>
  filter(Industry=="Takeaway food services")|>
  select(Month,Turnover)|>
  summarize(Turnover = sum(Turnover))

## Plottign original Data without Forecast - Seasonal Data so SNAIVE, apparent trend so DRIFT
autoplot(lim_aus_retail, Turnover)+ labs(title="Total Australian Takeaway Food Service Turnov
er  ")
```




```
## Getting Fitted Model Values.
retail_fit <- lim_aus_retail |> model(DRIFT = RW(Turnover ~ drift()),  SNAIVE=SNAIVE(Turnove
r))

retail_forecasts <- retail_fit |> forecast(h="2 years")

## Plotting Results
retail_forecasts |>
  autoplot(lim_aus_retail, level=NULL) + ## Removing C.I because it looks messy
  labs(title="Australia Takeaway Turnover")
```

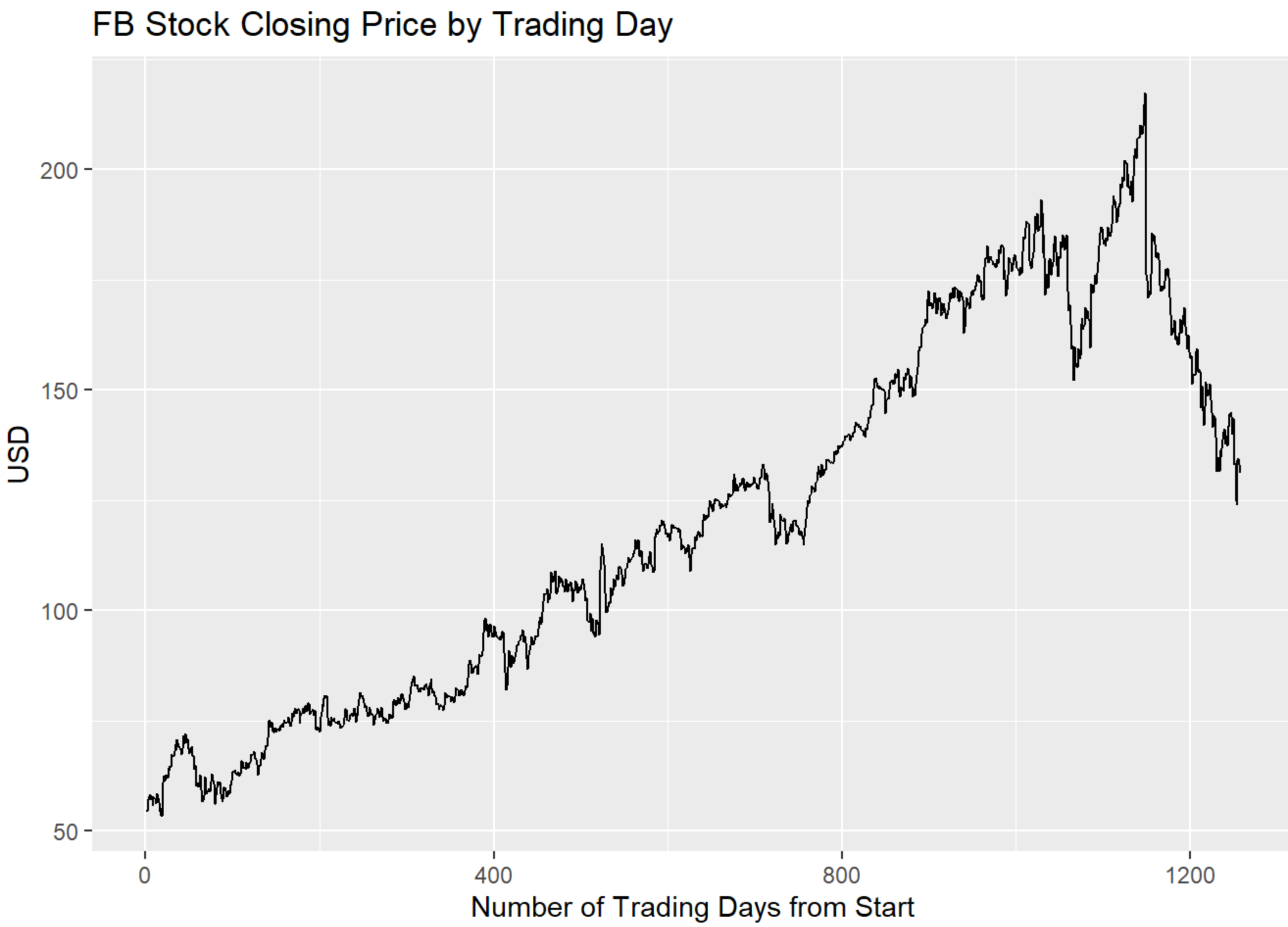


2) Use the Facebook stock price (data set gafa_stock) to do the following:

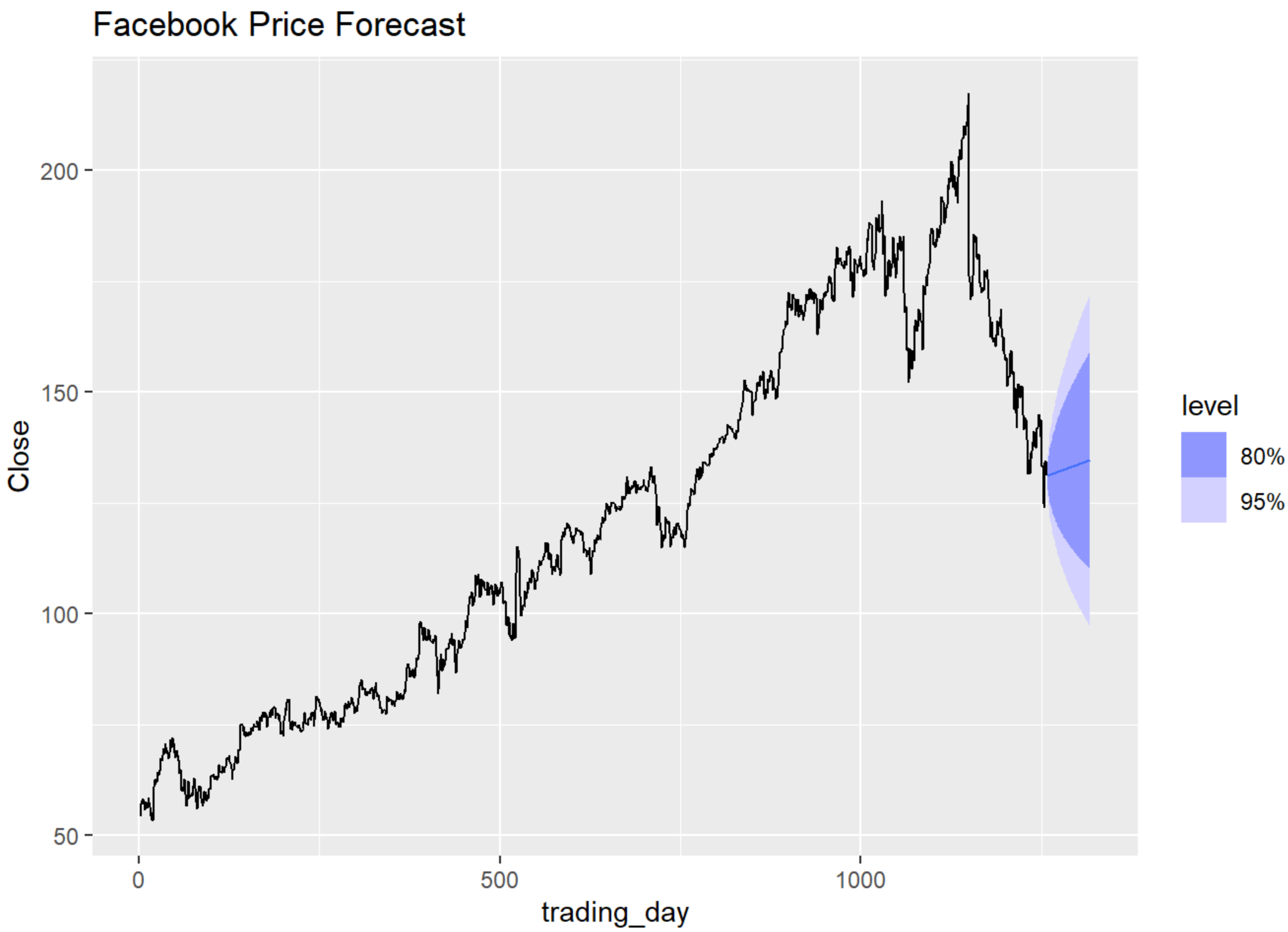
- a. Produce a time plot of the series.
- b. Produce forecasts using the drift method and plot them.
- c. Show that the forecasts are identical to extending the line drawn between the first and last observations.
- d. Try using some of the other benchmark functions to forecast the same data set. Which do you think is best? Why?

Question 2 Answer:

```
#Prepping the Facebook stock price data for what we need.Mimicking the text book here
fb_stock <- gafa_stock |>
  filter(Symbol == "FB") |>
  mutate(trading_day = row_number()) |>
  update_tsibble(index = trading_day, regular = TRUE)

## Initial Time plot of the series
autoplot(fb_stock, Close)+ labs(title="FB Stock Closing Price by Trading Day", y="USD", x="Nu
mber of Trading Days from Start")
```

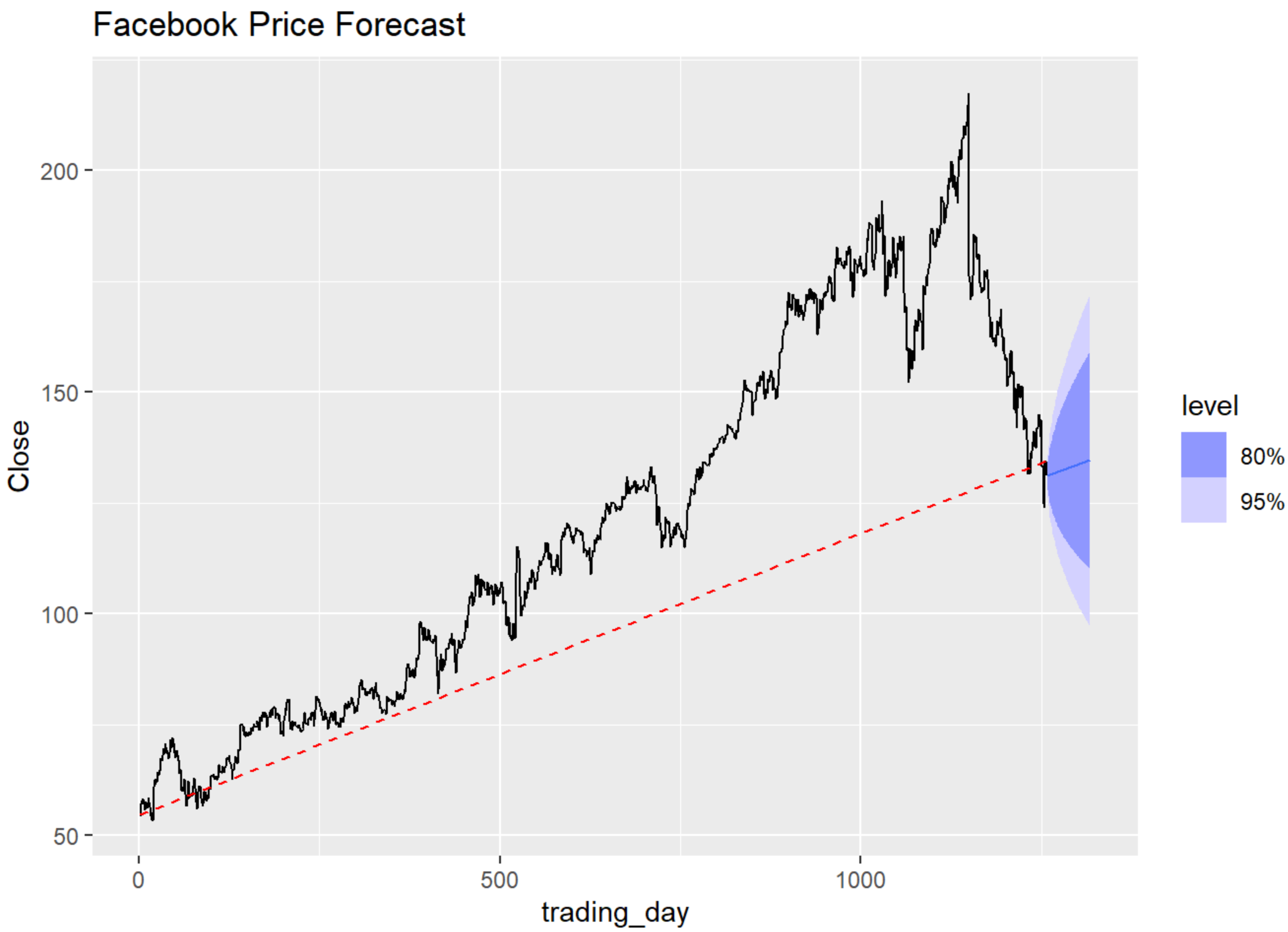



```
## Generating Fit Model Points
drift_fb_fit <- fb_stock |> model(DRIFT = RW(Close ~ drift()))

fb_forecasts <- drift_fb_fit |> forecast(h=60) ## Doing 30 because 30 days

fb_forecasts |>
  autoplot(fb_stock) +
  labs(title="Facebook Price Forecast")
```



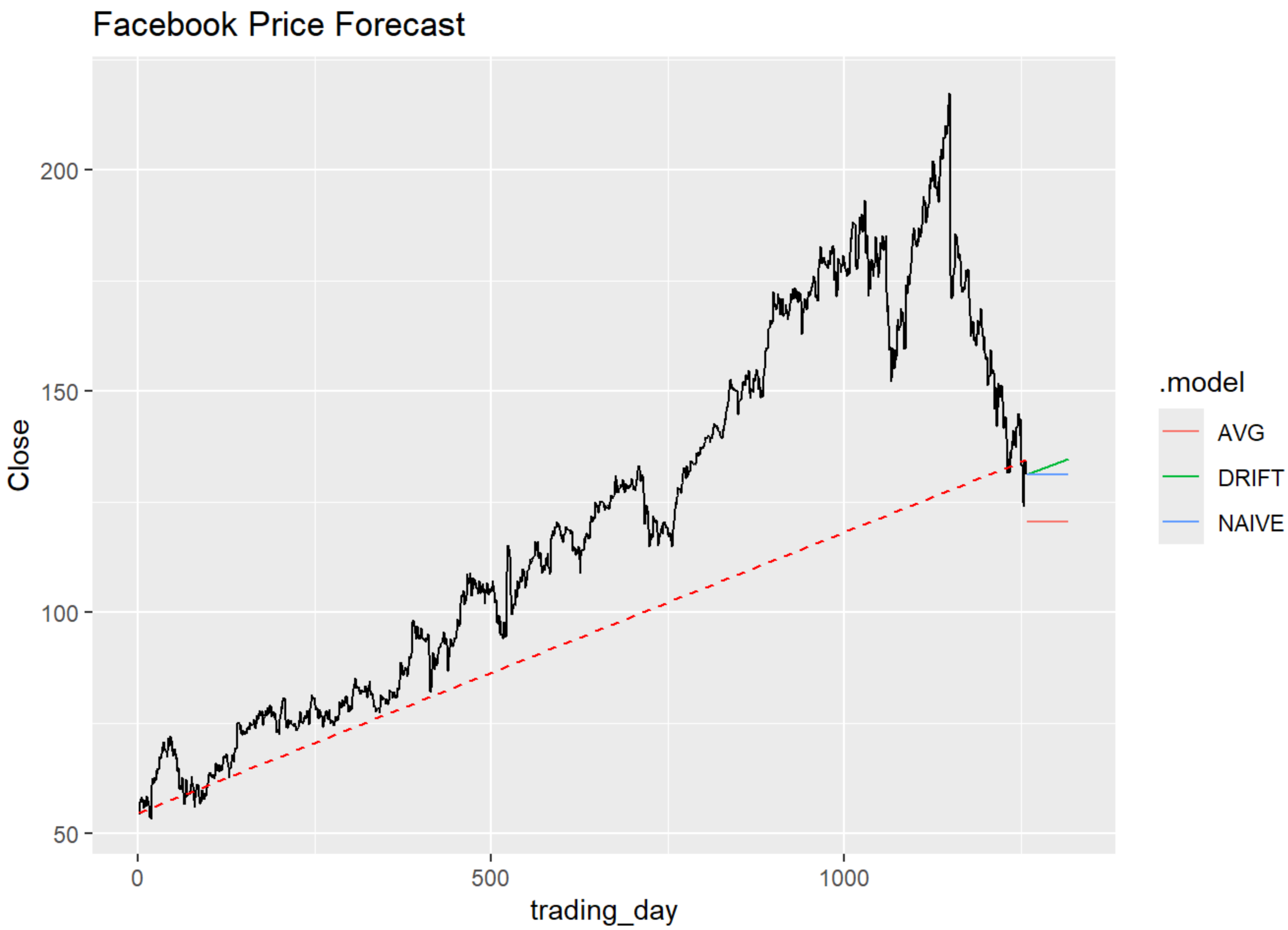

```
## Taking first and last values based on trading day for the begining and end line.
fb_stock_min_max <- fb_stock |>
  filter(trading_day == min(trading_day) | trading_day == max(trading_day))

## Plotting both the forecasts and the line on same chart.
fb_forecasts |>
  autoplot(fb_stock) +
  autolayer(fb_stock_min_max, color = "red", linetype = "dashed") +
  labs(title="Facebook Price Forecast")
```

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




```
### Adding other bench mark functions to the model.
drift_fb_fit <- fb_stock |> model(
  DRIFT = RW(Close ~ drift()),
  NAIVE = NAIVE(Close),
  AVG = MEAN(Close))

fb_forecasts <- drift_fb_fit |> forecast(h=60) ## Doing 30 because 30 days

fb_forecasts |>
  autoplot(fb_stock, level=NULL) +
  autolayer(fb_stock_min_max, color = "red", linetype = "dashed") +
  labs(title="Facebook Price Forecast")
```

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




```
## Tried the other options for bench marks in the data. I think DRIFT is the best as it seems
to capture the longer term trend of increasing price pf tje stoc, while the Avg and the Naive
do not.
```

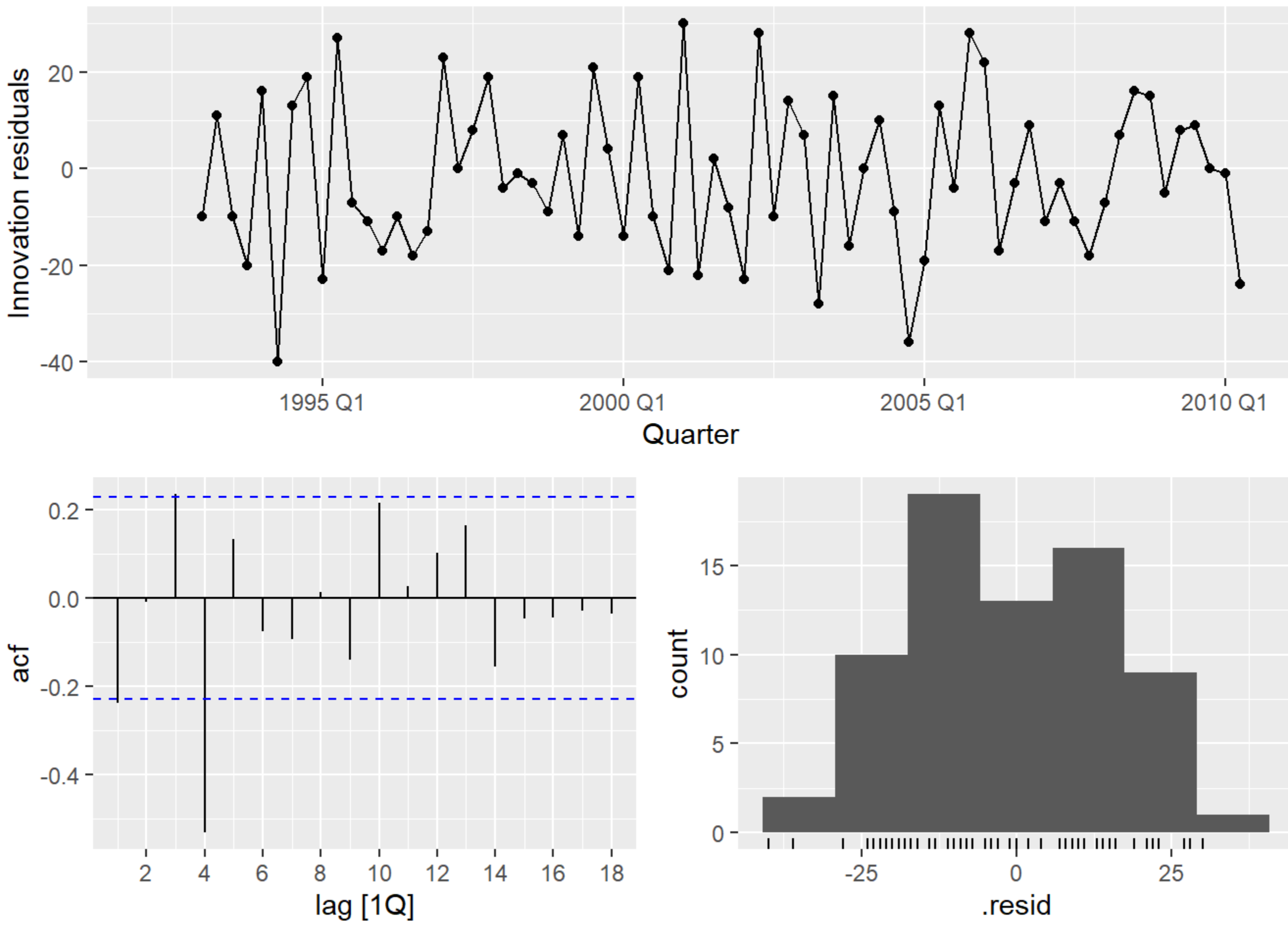
3) Apply a seasonal naïve method to the quarterly Australian beer production data from 1992. Check if the residuals look like white noise, and plot the forecasts. The following code will help.

```
# Extract data of interest
recent_production <- aus_production |>
  filter(year(Quarter) >= 1992,
         ## Limiting no nulls
         !is.na(Beer))
# Define and estimate a model
fit <- recent_production |> model(SNAIVE(Beer))
# Look at the residuals
fit |> gg_tsresiduals()
```

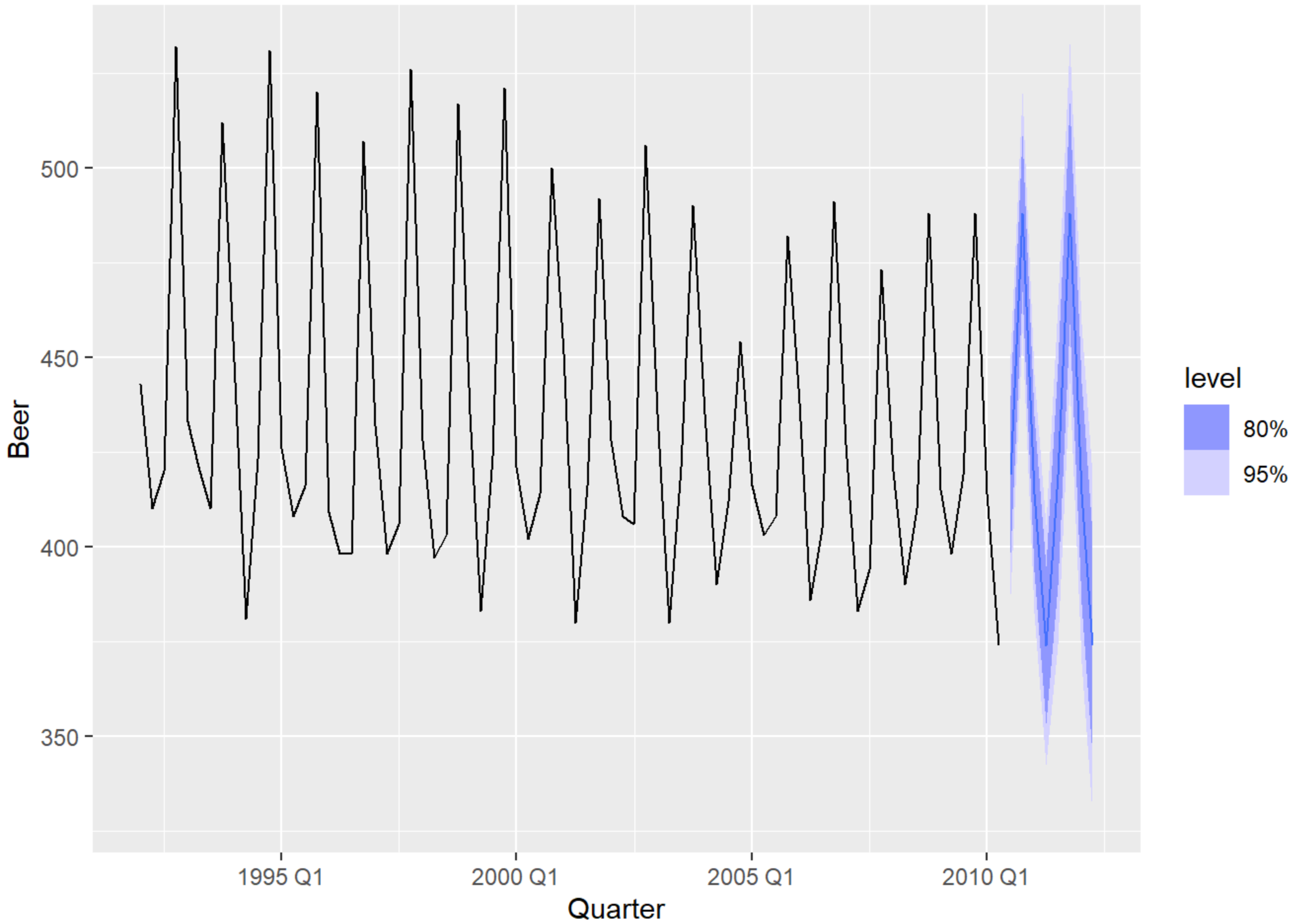
```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## 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)
```



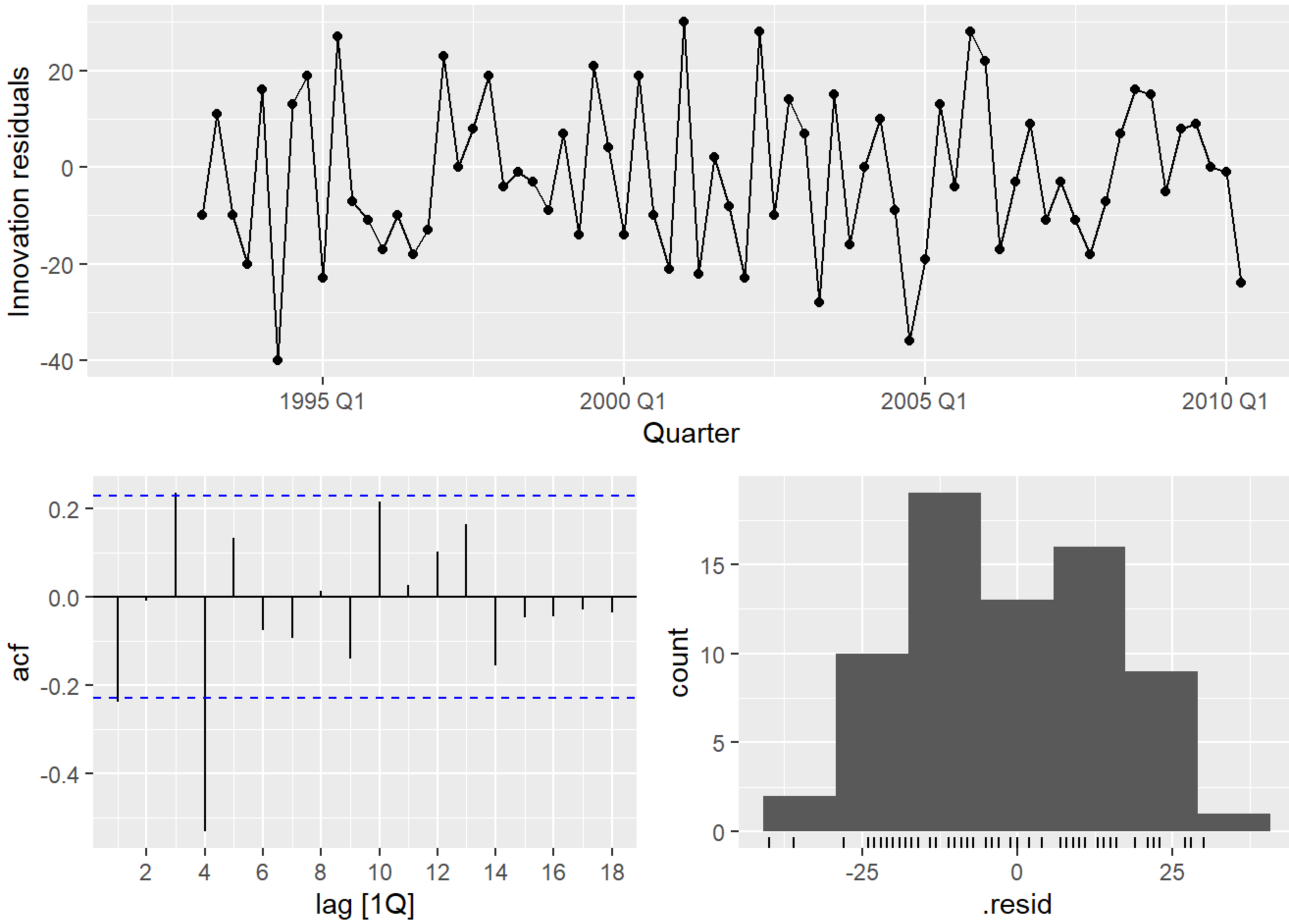
Question 3 Answer:

```
## Checking to see if the residuals  of the previous forecasting look like white noise

## Taking a look at the residual plots for the above.
gg_tsresiduals(fit)
```

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




```
## Visual Notes: The residual distribution in the histogram seems somewhat normal. The ACF ch
art has all but one of the residual values outlise of the limit. Generally good. However, the
first chart on the top portion of the plots does NOT look ok. THe variance does not dseem to
surrounding zero there are many larger spikes in the data.

## Next Step is a Portmanteau Test with Ljung-Box method. Seasonal Data so we want a lag of 8
because ites 2x the seasonal period.
augment(fit) |> features(.resid, ljung_box, lag=8)
```

```
## # A tibble: 1 × 3
##   .model       lb_stat lb_pvalue
##   <chr>          <dbl>     <dbl>
## 1 SNAIVE(Beer)    32.3 0.0000834
```

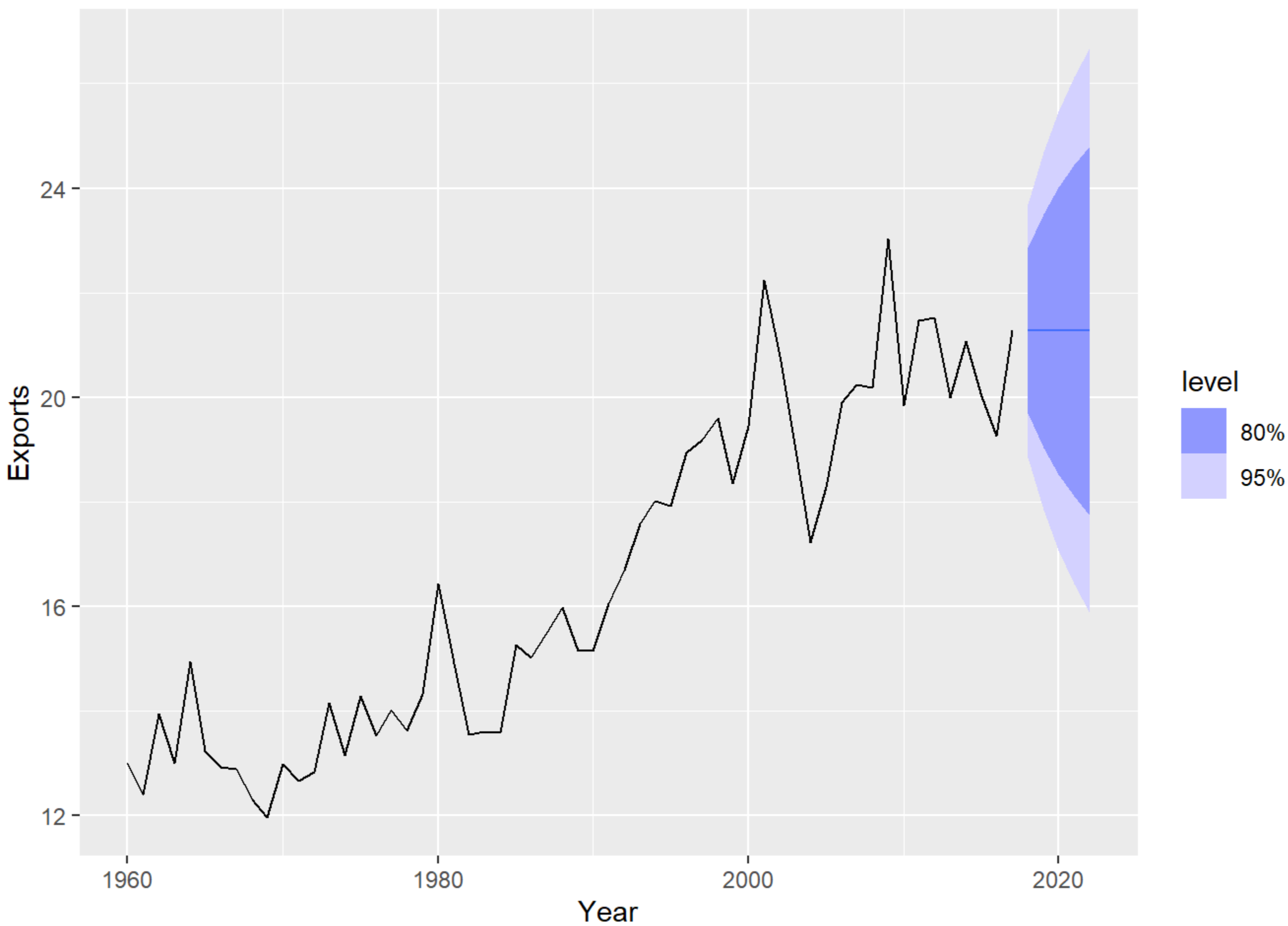
```
## The pvalue is statistically relevant, pair that with the large Q value of 32.2 this means
that this is probably NOT white noise.
```

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.

Question 4 Answer:

```
## Doing the same for the Exports in the global_economy dataset. Non Seasonal Data, so using
NAIVE

## Country Lim
aus_exp <- global_economy |> filter(Country == "Australia", !is.na(Exports))

# Define and estimate a model
fit <- aus_exp |> model(NAIVE(Exports))

# Look a some forecasts
fit |> forecast(h="5 years") |> autoplot(aus_exp)
```

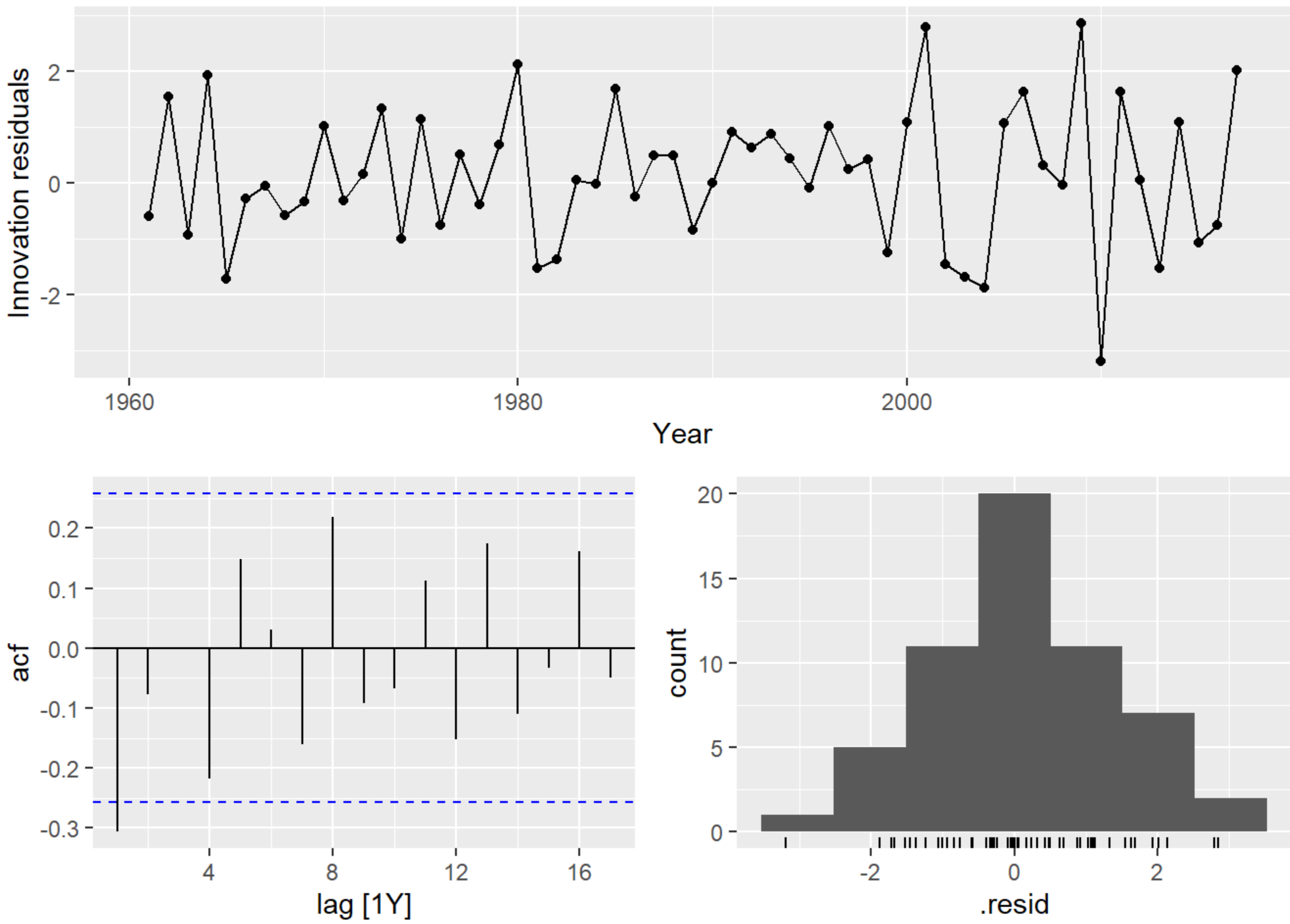



```
### Checking the resids
fit |> gg_tsresiduals()
```

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

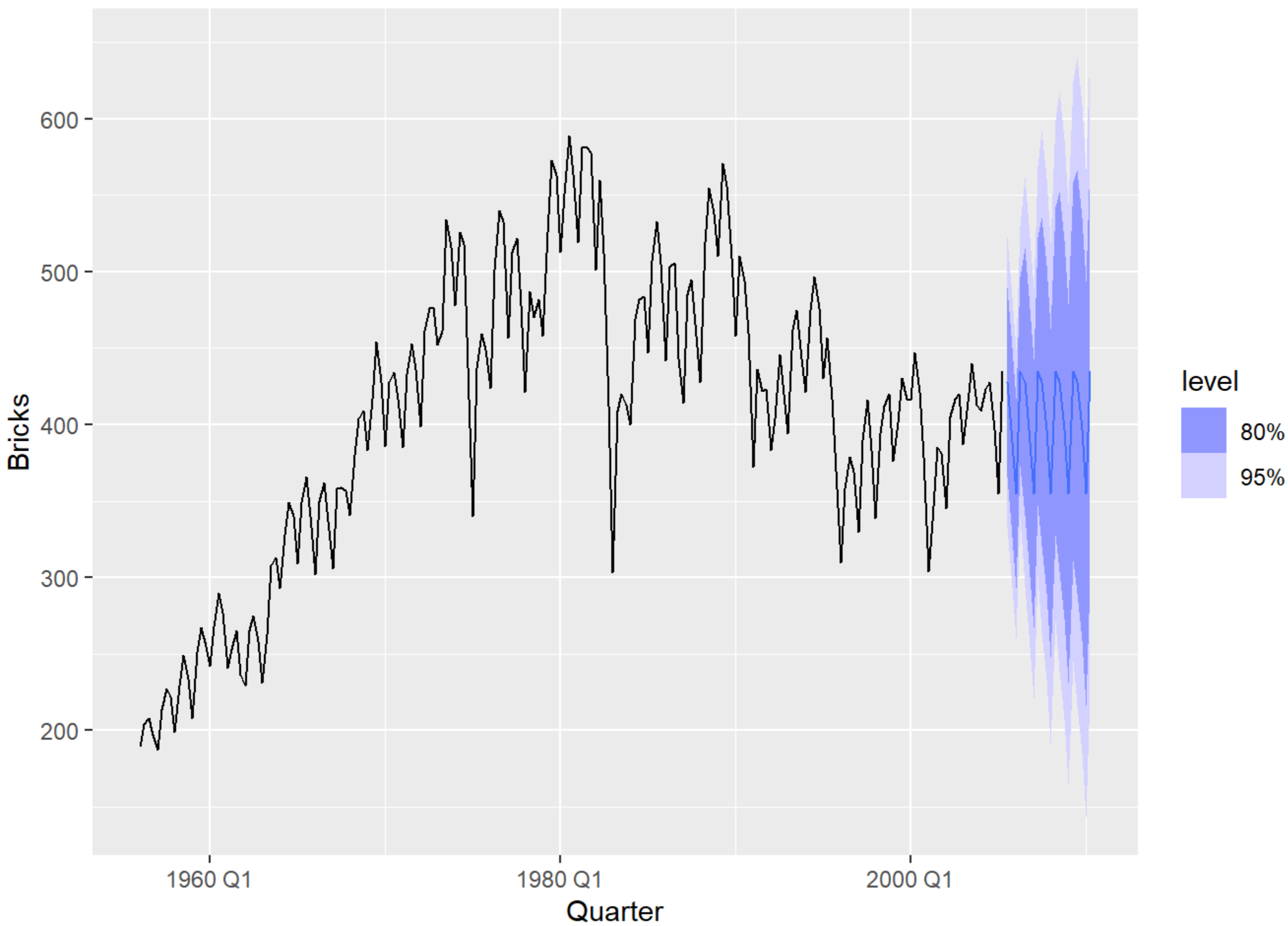



```
### Visual Check: Histogram seems like mostly Normal Distribution. However, first plot does n
ot show residuals staying around zero. There are larger spikes, however this spikes are gener
ally aroudn 2/-2. Lastly, the ACF chart only has on value outside of the limit, which is good
. Moving on to Portmanteau test.

augment(fit) |> features(.resid, ljung_box, lag=10)
```

```
## # A tibble: 1 × 4
##   Country   .model         lb_stat lb_pvalue
##   <fct>     <chr>            <dbl>     <dbl>
## 1 Australia NAIVE(Exports)    16.4    0.0896
```

```
## The pvalue is over 0.05 so that means that the Q result is not statistically significant a
nd we cannot reject the null hypothesis of white noise. So it is white noise.
```

```
## Doing the same for the Bricks in the aus_production dataset. Seasonal Data, so using SNAIV
E

# Limiting for values
brick_prod <- aus_production |> filter(!is.na(Bricks))


# Define and estimate a model
fit <- brick_prod |> model(SNAIVE(Bricks))

# Look a some forecasts
fit |> forecast(h = "5 years") |> autoplot(brick_prod)
```

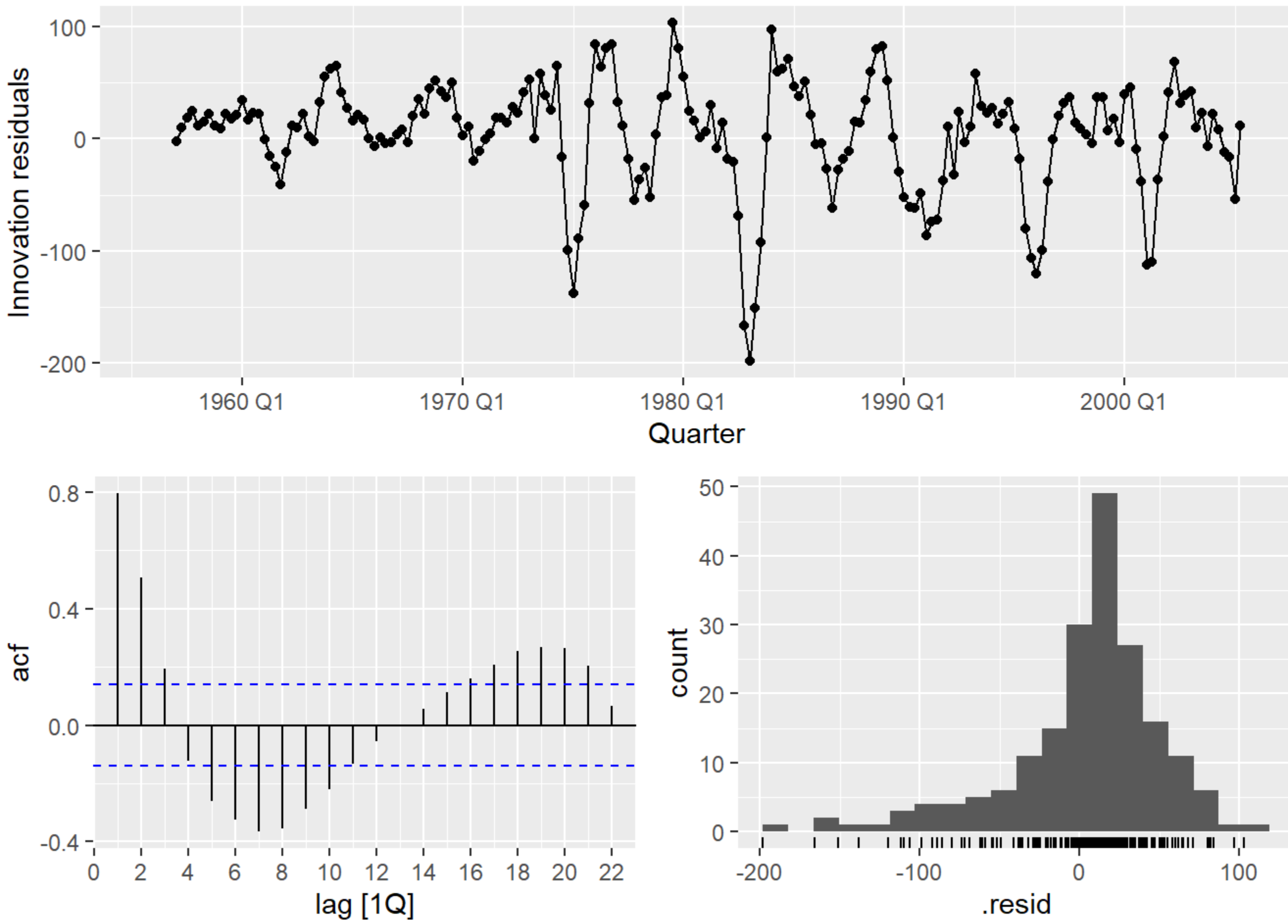



```
### Checking the resid
fit |> gg_tsresiduals()
```

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




```
### Visual Check: Histogram is not ormallydistributed. First plot does not show residuals sta
ying around zero. There are larger spikes up to aroudn 100 and down to -200. Lastly, the ACF
chart only has many values outside the limits. Additionally the is an obvious patterin the th
e residuals here.

augment(fit) |> features(.resid, ljung_box, lag=8) ## Quarterly data so 2m is 8
```

```
## # A tibble: 1 × 3
##   .model         lb_stat lb_pvalue
##   <chr>            <dbl>     <dbl>
## 1 SNAIVE(Bricks)    274.         0
```

```
## P value is 0 with a very large Q score so we reject any possibility of white noise.
```

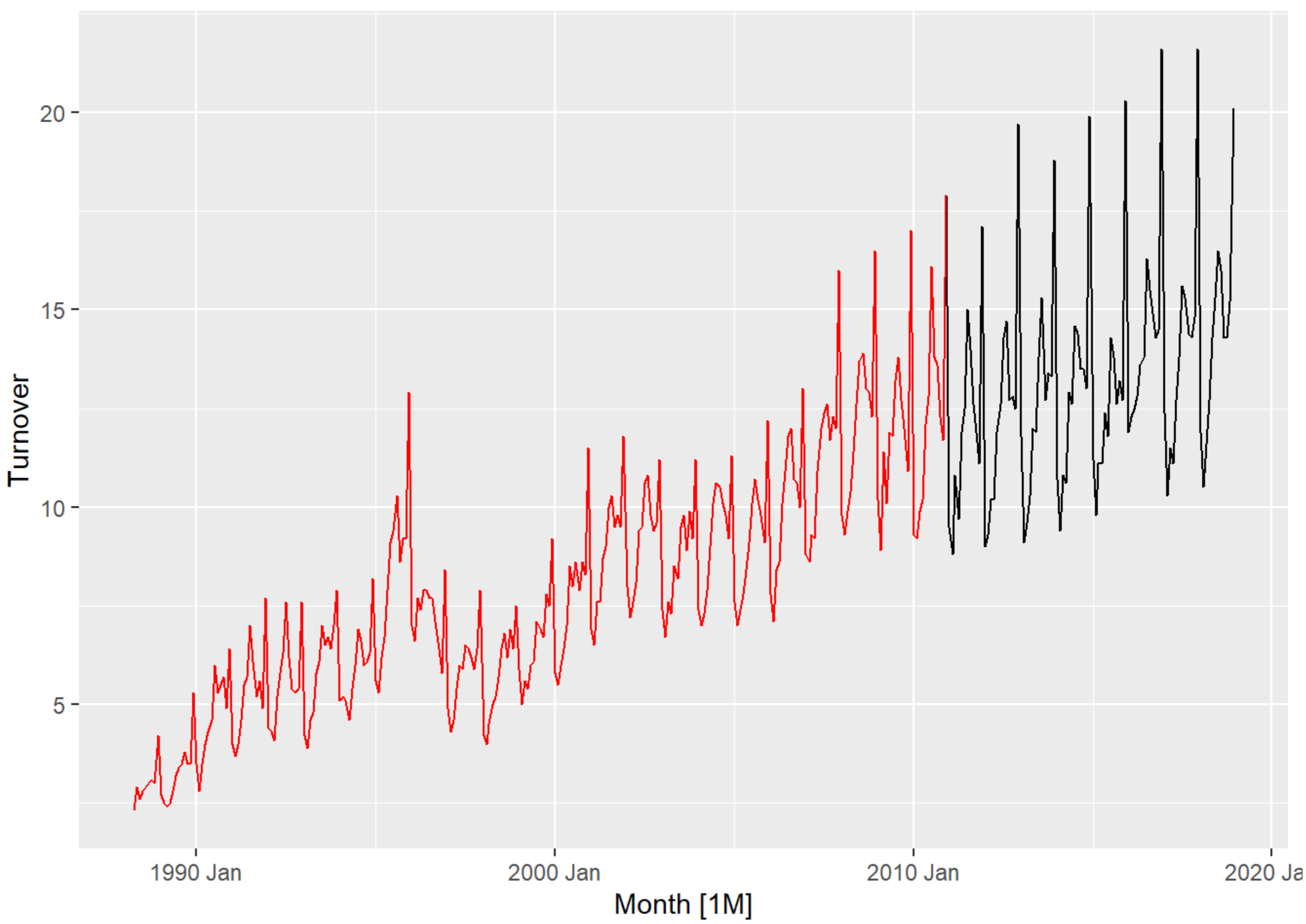
7) For your retail time series (from Exercise 7 in Section 2.10):

a) Create a training dataset consisting of observations before 2011 using.

```
set.seed(12345678)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
myseries_train <- myseries |>
  filter(year(Month) < 2011)

# myseries_test <- myseries |>
#   filter(year(Month) >= 2011)
```

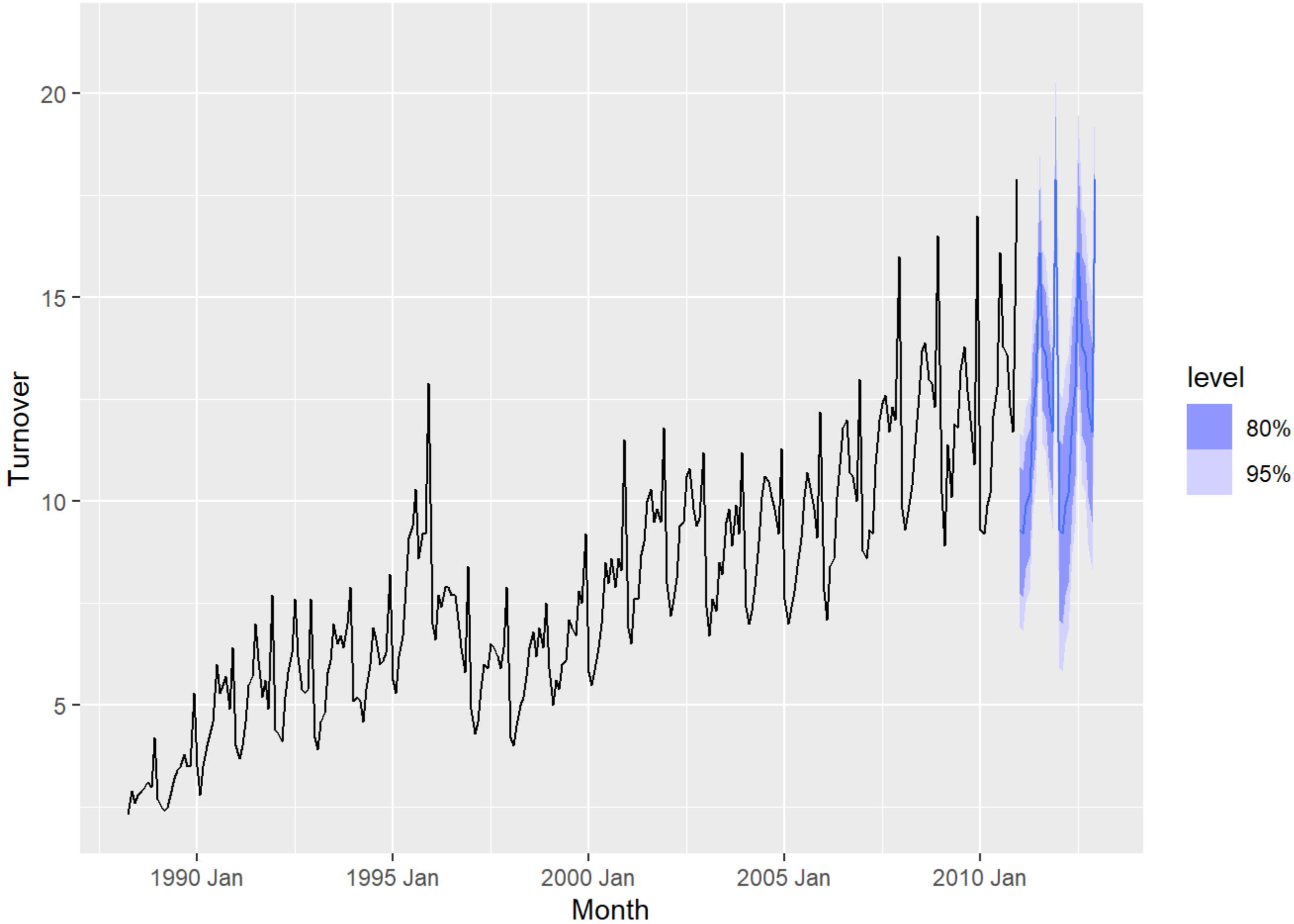
b) Check that your data have been split appropriately by producing the following plot.

```
autoplot(myseries, Turnover) +
  autolayer(myseries_train, Turnover, colour = "red")
```



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

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

## Plotting pre 2011 froecast
fit |> forecast() |> autoplot(myseries_train)
```



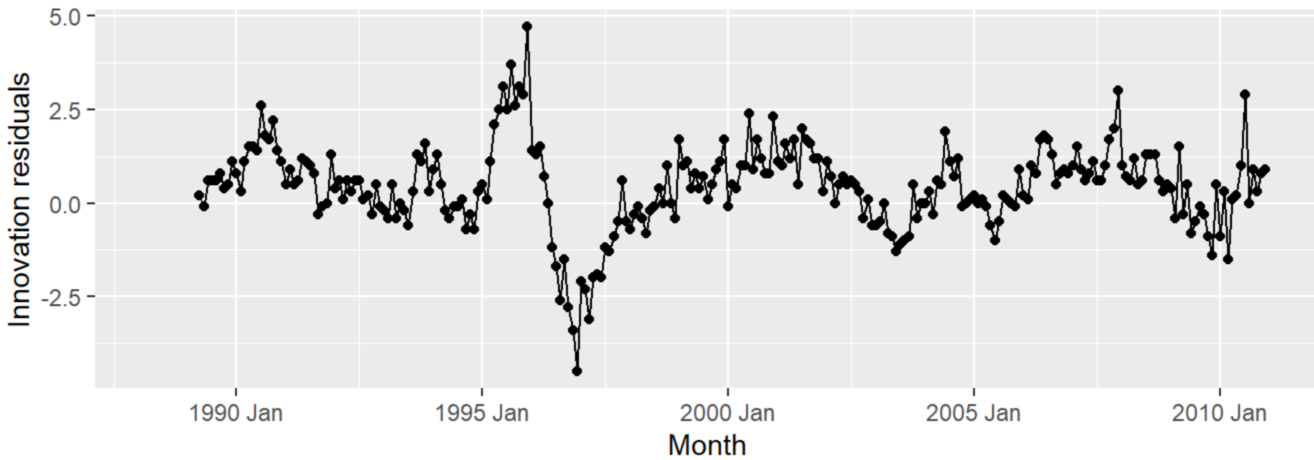
d) Check the residuals.Do the residuals appear to be uncorrelated and normally distributed?

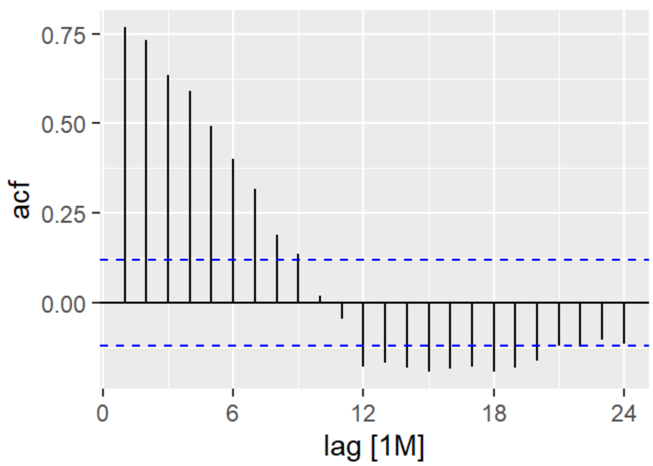
```
## Checking Residuals
fit |> gg_tsresiduals()
```

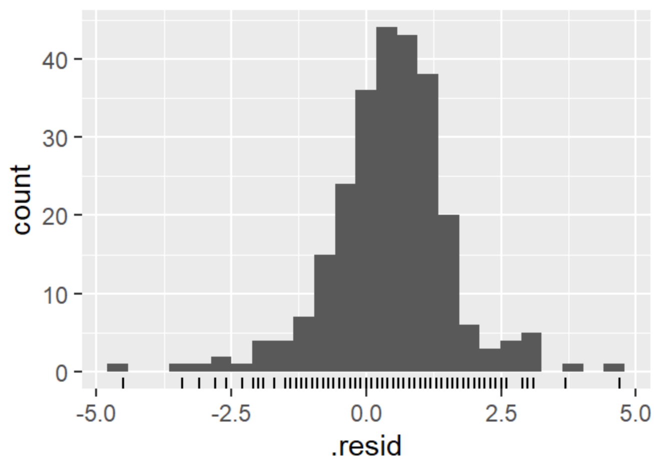
```
## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

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

```
## Warning: Removed 12 rows containing non-finite outside the scale range
## (`stat_bin()`).
```








```
## Visual Checks: There are large variations in the first plot, not just constant around zero
. However, they are relatively large, only going up to 5 and down to 3. The distribution in t
he histogram appears mostly normal, however there are some outliers. Lastly, there are many i
nstances of residuals being outside the limits in the ACF chart. There is also an obvious pat
tern of correlation for the residuals in this chart.

## Port. Test
augment(fit) |> features(.resid, ljung_box, lag=24) ## Monthly Data
```

```
## # A tibble: 1 × 5
##   State              Industry                                  .model lb_stat lb_pvalue
##   <chr>              <chr>                                     <chr>    <dbl>     <dbl>
## 1 Northern Territory Clothing, footwear and personal a… SNAIV…    746.         0
```

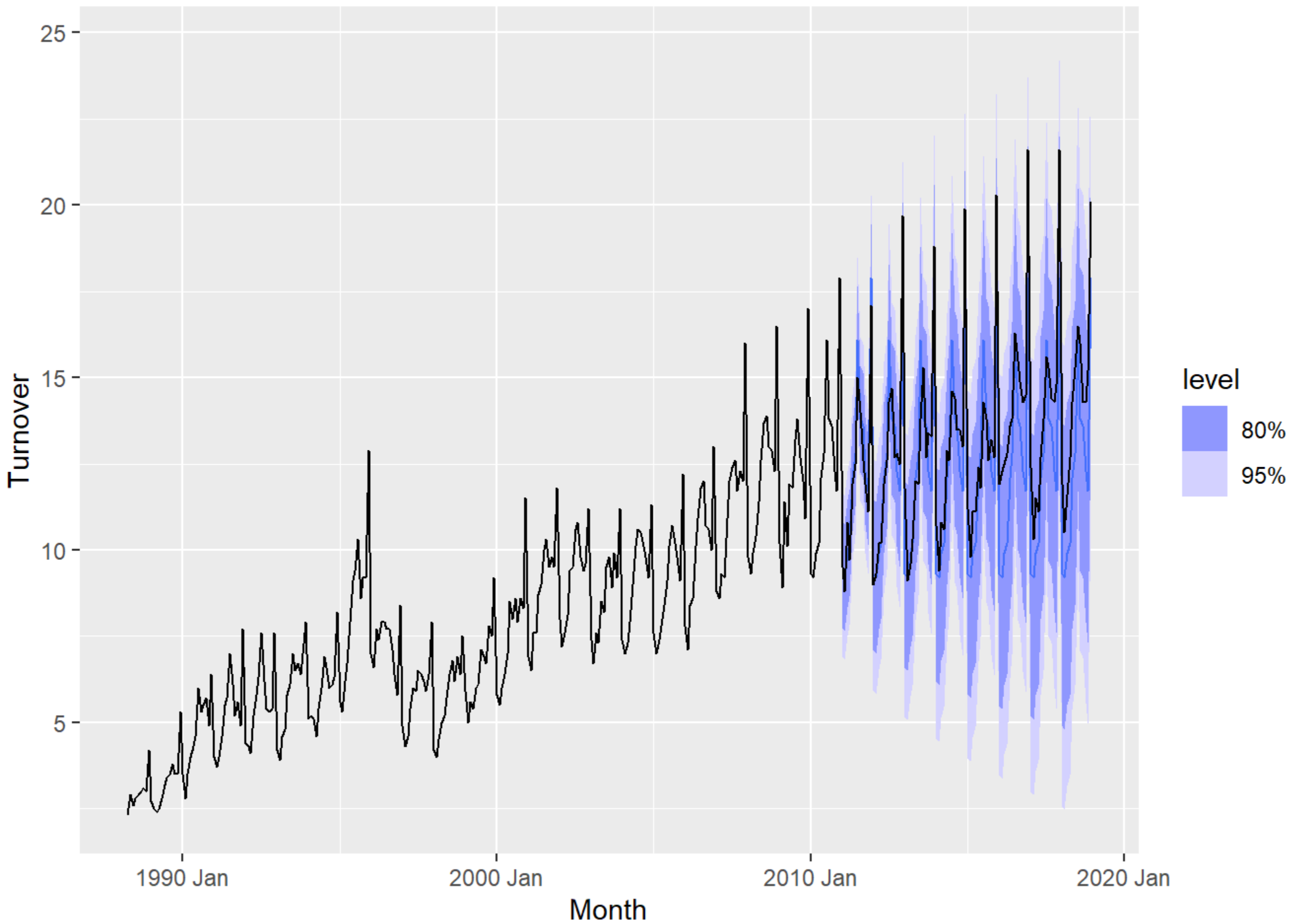
```
### This is NOT white noise, p score is 0 along with very large q score.
```

e) Produce forecasts for the test data

```
## using code from book to forcast all data with training data.

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.

```
##Training Data
print(fit |> accuracy())
```

```
## # A tibble: 1 × 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 Norther… Clothin… SNAIV… Trai… 0.439  1.21 0.915  5.23  12.4     1     1 0.768
```

```
# ME       RMSE        MAE        MPE        MAPE     MASE    RMSSE
# 5.203003  14.39031    10.34054    5.449036    11.49075    1      1



## Test Data
print(fc |> accuracy(myseries))
```

```
## # A tibble: 1 × 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(T… Nort… Clothin… Test  0.836  1.55  1.24  5.94  9.06  1.36  1.28 0.601
```

```
# ME         RMSE       MAE       MPE       MAPE      MASE     RMSSE
# 12.54687  17.71474    14.66354    6.497419    7.782965    1.418063 1.231019


## Overall the first model forecast is better than the second. The level of errors are lower
in the first model most likely because the training data was used on it
```

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

```
###  For each of the following, the sensitivities are:
### ME (MEAN ERROR) - This measure is highly sensitive to the amount of training data useds a
s it is a simple mean, so this metric can be swayed easily.

### MAE (MEAN ABSOLUTE ERROR) - Using the absolute error cancels out the over and under estim
ates in the data with the Absolute fuinction. Less sensitive than ME.

### MSE (Mean Squared Error) - The squaring for the mean error dulls the easiness with which
the measure is influenced by the amount of data. Depending on the outliers this would be less
sensitive than the ME to the amount of data.

### RMSE (Root Mean Square Error) - This would be same sentivity as MSE, as its simply bringi
ng the scale back to the original data scale bu undoing the X^2.

### MAPE (Mean Absolute Percentage Error) - Attempts to normalize the data by taking away the
scale of the errors. Only works if Y doesn't have zero values, or not super close to zero. Th
is would be less sensitive to the amoutn of data trained because the data itself is used to n
ormalize.

### MASE (Mean Absolute Scale Error) - Similar to the MAPE normalization, but uses the "scale
" of the data or the absolute value of the difference of the samples instead of simply the Y
value. The absolue values are averages for this metric. This would be less sensitive to the a
mount of data as the range in the data itself is the denominator for normalization

### RMSSE (Root Mean Squared Scaled Error) - Similar to MASE.

## In conclusion, based on these two sets of results, the ME is the most sensitive. With MAE,
RMSE, and MAPE seeming to be in second for how sensitive if were comparing relational values.
```