Homework 3

Heleine Fouda

2024-09-21

From Rob J Hyndman and George Athanasopoulos' book: Forecasting: Principles and Practice.

Set up the environment: Load required libraries.

Exercise 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:

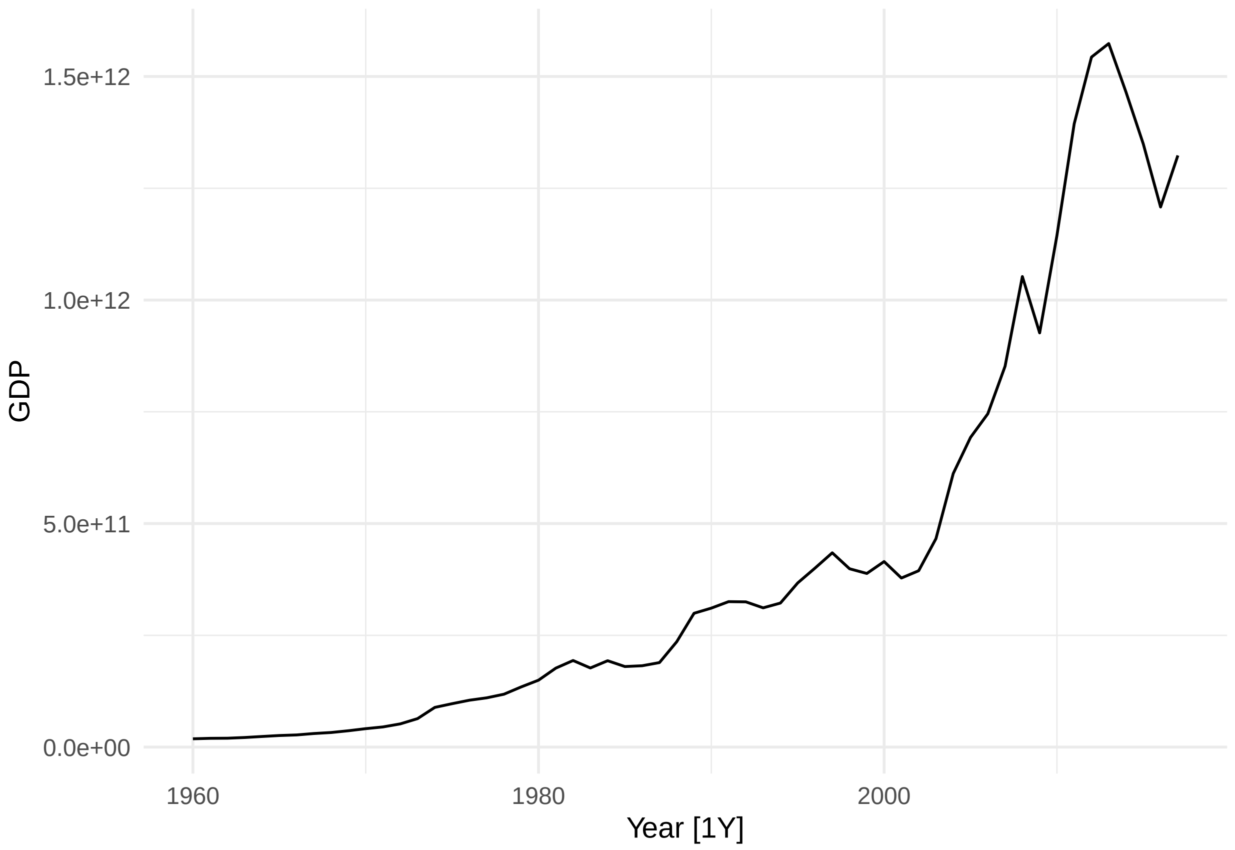
Australian Population (global_economy) Bricks (aus_production) NSW Lambs (aus_livestock) Household wealth (hh_budget). Australian takeaway food turnover (aus_retail).

Australian Population (global_economy)

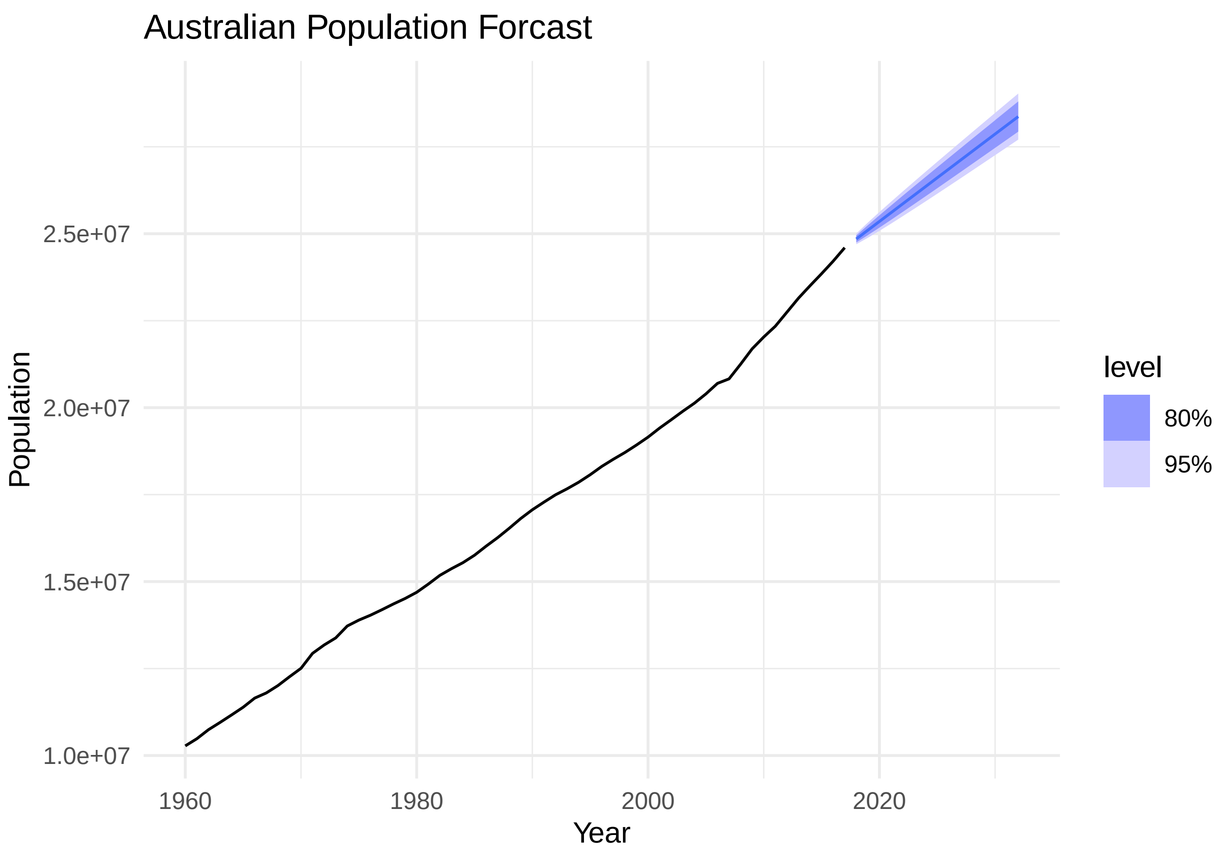
The population series typically shows a trend over time, making the random walk with drift method suitable.This method allows the forecasts to increase or decrease over time with the drift reflecting the average change seen in the historical data.

```r
# Plot the time series
aus_economy <- global_economy %>%
    filter(Country == "Australia")
 autoplot(aus_economy)+
   theme_minimal()
```

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



```
# Apply the appropriate method
aus_economy <- global_economy %>%
    filter(Country == "Australia")

aus_economy %>%
  model(Drift = RW(Population ~ drift())) %>%
  forecast(h = 15) %>%
  autoplot(aus_economy) +
    labs(title = "Australian Population Forcast")+
  theme_minimal()
```



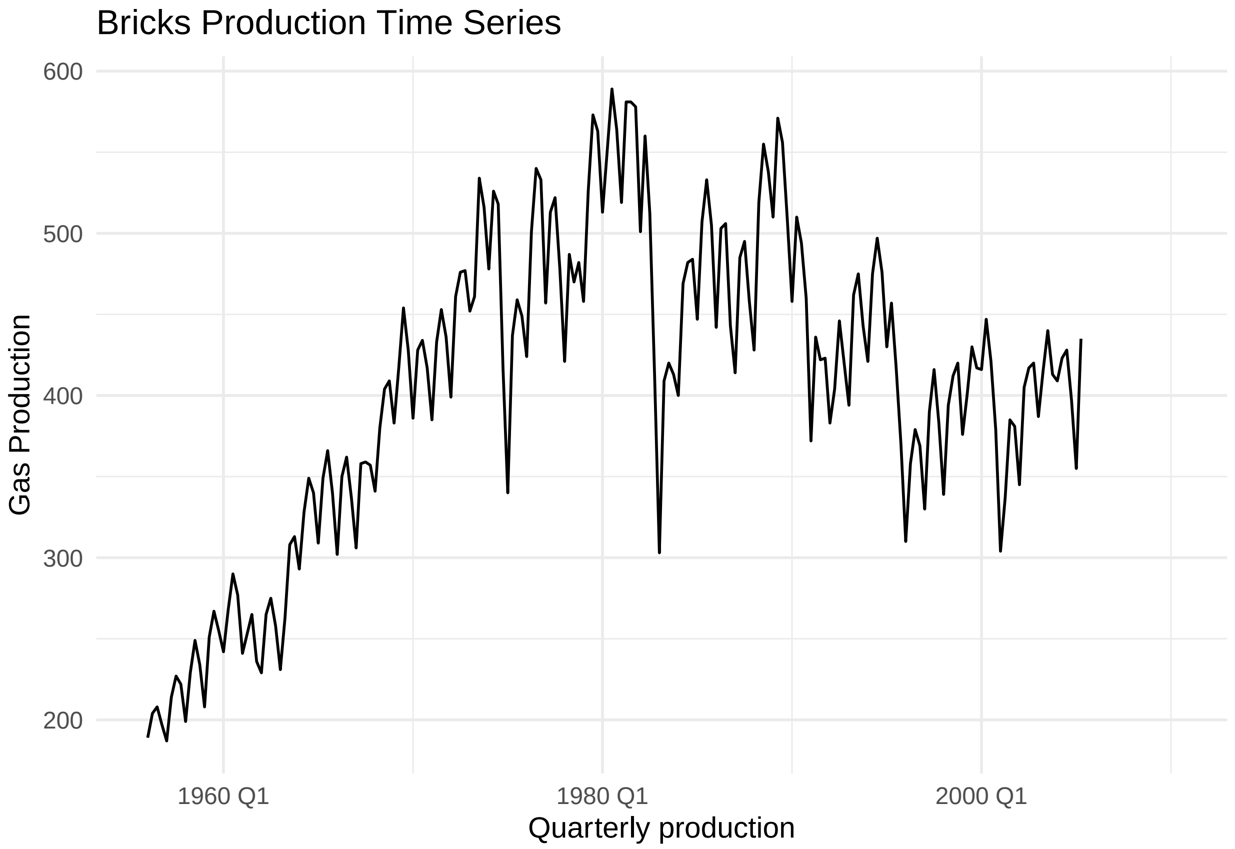
Bricks (aus_production)

```
library(tsibble)
# A glimpse of the data set
aus_production <-tsibbledata::aus_production
glimpse(aus_production)
```

```
## Rows: 218
## Columns: 7
## $ Quarter     <qtr> 1956 Q1, 1956 Q2, 1956 Q3, 1956 Q4, 1957 Q1, 1957 Q2, 1957~
## $ Beer        <dbl> 284, 213, 227, 308, 262, 228, 236, 320, 272, 233, 237, 313~
## $ Tobacco     <dbl> 5225, 5178, 5297, 5681, 5577, 5651, 5317, 6152, 5758, 5641~
## $ Bricks      <dbl> 189, 204, 208, 197, 187, 214, 227, 222, 199, 229, 249, 234~
## $ Cement      <dbl> 465, 532, 561, 570, 529, 604, 603, 582, 554, 620, 646, 637~
## $ Electricity <dbl> 3923, 4436, 4806, 4418, 4339, 4811, 5259, 4735, 4608, 5196~
## $ Gas         <dbl> 5, 6, 7, 6, 5, 7, 7, 6, 5, 7, 8, 6, 5, 7, 8, 6, 6, 8, 8, 7~
```

```
bricks <- aus_production |>
 dplyr:: select("Bricks", "Quarter")

# Plot the time series
bricks |>
  autoplot(Bricks) +
  labs(title = "Bricks Production Time Series",
       x = "Quarterly production",
       y = "Gas Production")+
  theme_minimal()
```

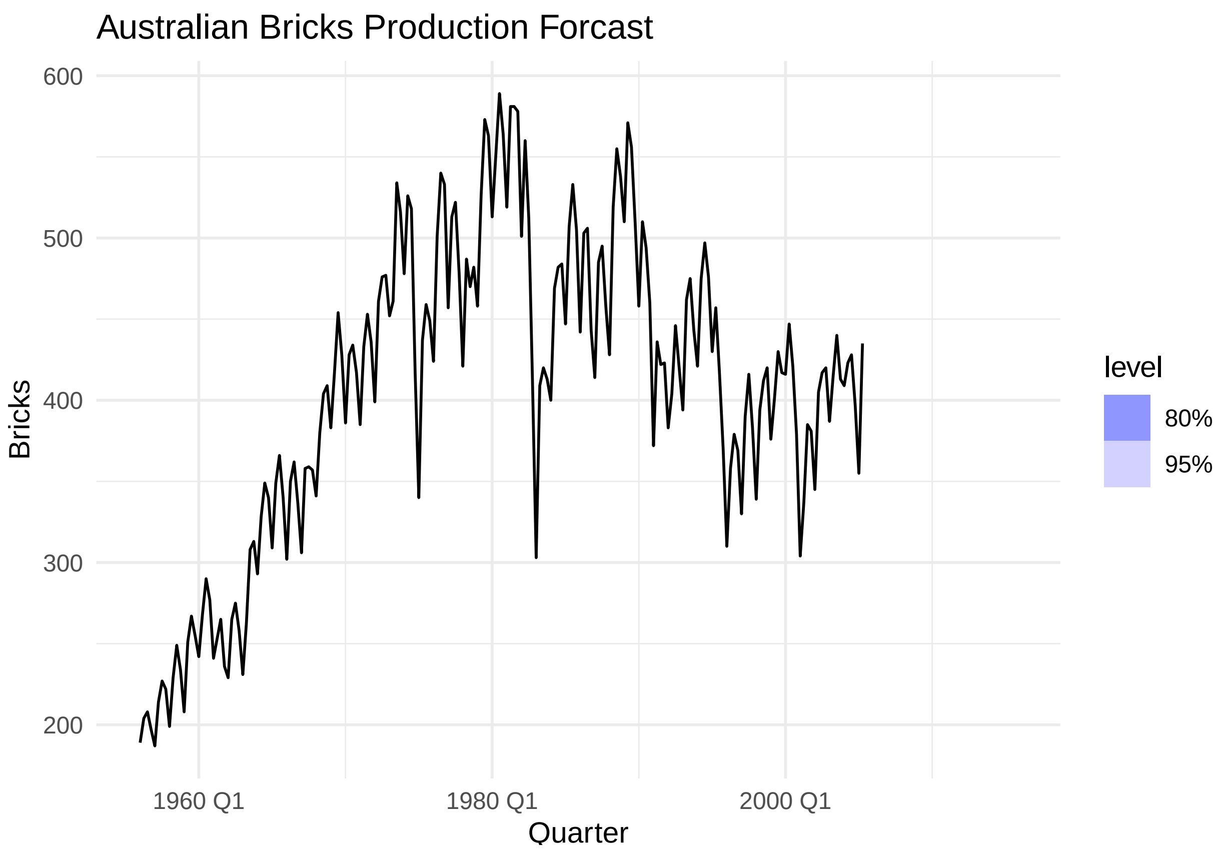
```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



This series exhibits strong seasonal patterns expressed in Quarters. The seasonal naive method will therefore be the appropriate method.

```
# Apply the appropriate method
bricks %>%
  filter(!is.na(Bricks)) %>%
  model(SNAIVE(Bricks ~lag("1956"))) %>%
 forecast(h = 42) %>%
  autoplot(aus_production) +
    labs(title = "Australian Bricks Production Forcast") +
  theme_minimal()
```

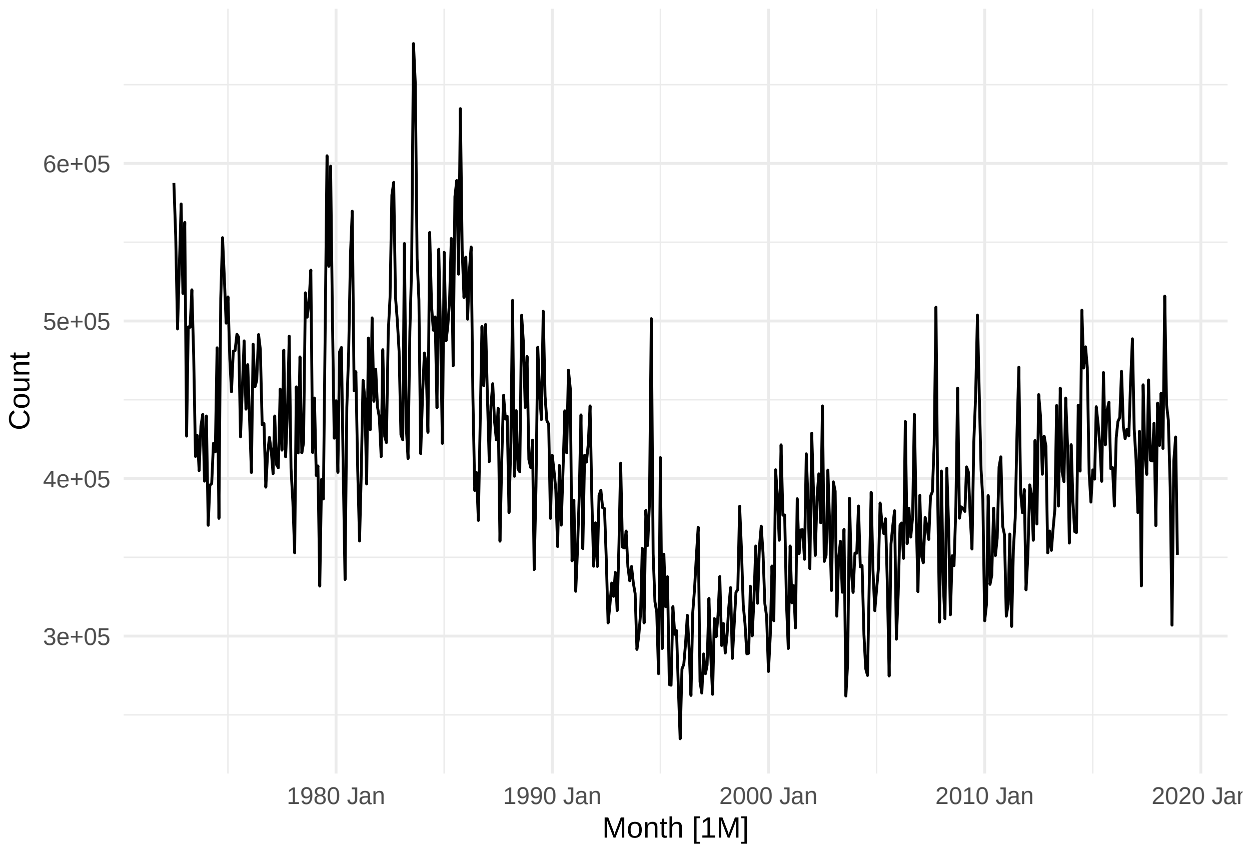
```
## Warning: 1 error encountered for SNAIVE(Bricks ~ lag("1956"))
## [1] Unknown period: 1956

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## Warning: Removed 42 rows containing missing values or values outside the scale range
## (`geom_line()`).

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



### NSW Lambs (aus_livestock)

```r
# Plot the time series
nsw_livestock <- aus_livestock %>%
  filter(State == "New South Wales",
         Animal == "Lambs")
autoplot(nsw_livestock)+
  theme_minimal()
```

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

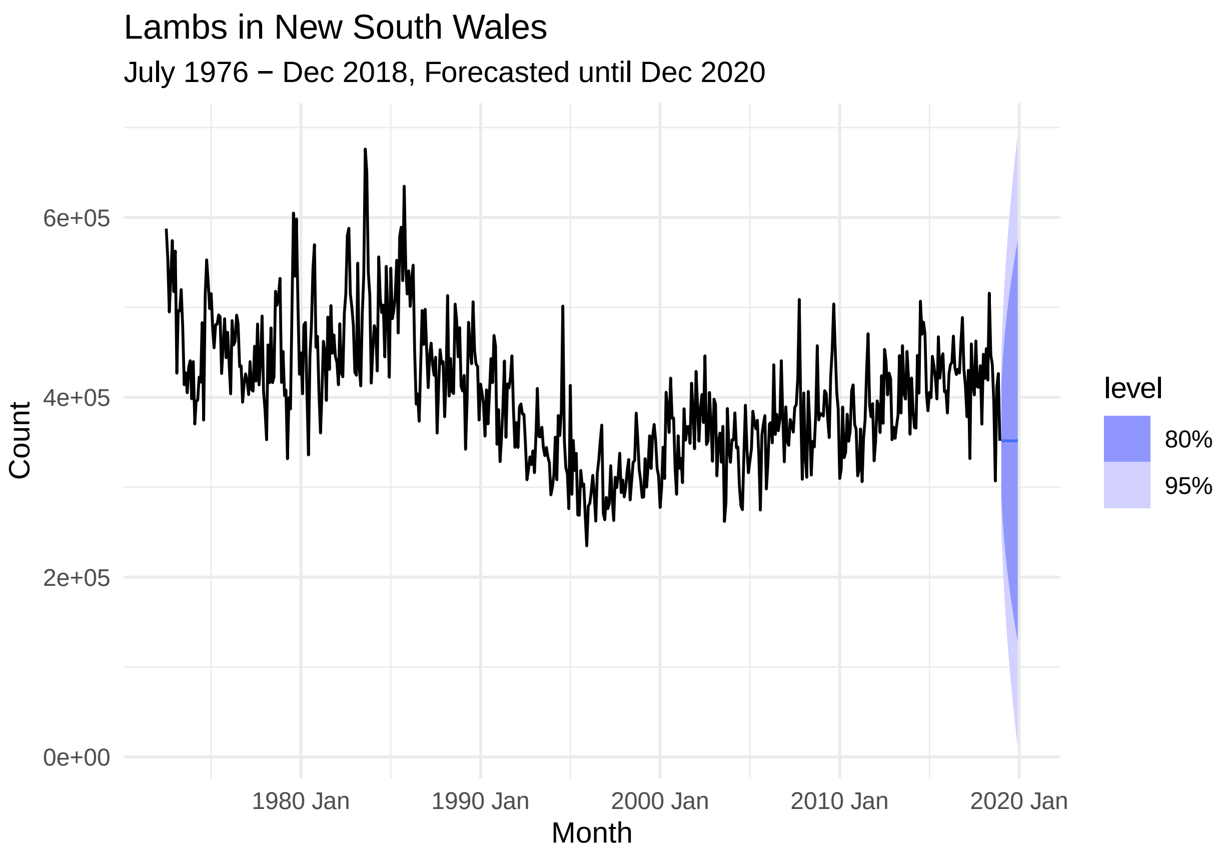


aus_livestock is a monthly tsibble that presents seasonal variations. The seasonal naive method seems a good choice for this dataset

```
unique(aus_livestock$Animal)
```

```
## [1] Bulls, bullocks and steers Calves
## [3] Cattle (excl. calves)      Cows and heifers
## [5] Lambs                      Pigs
## [7] Sheep
## 7 Levels: Bulls, bullocks and steers Calves ... Sheep
```

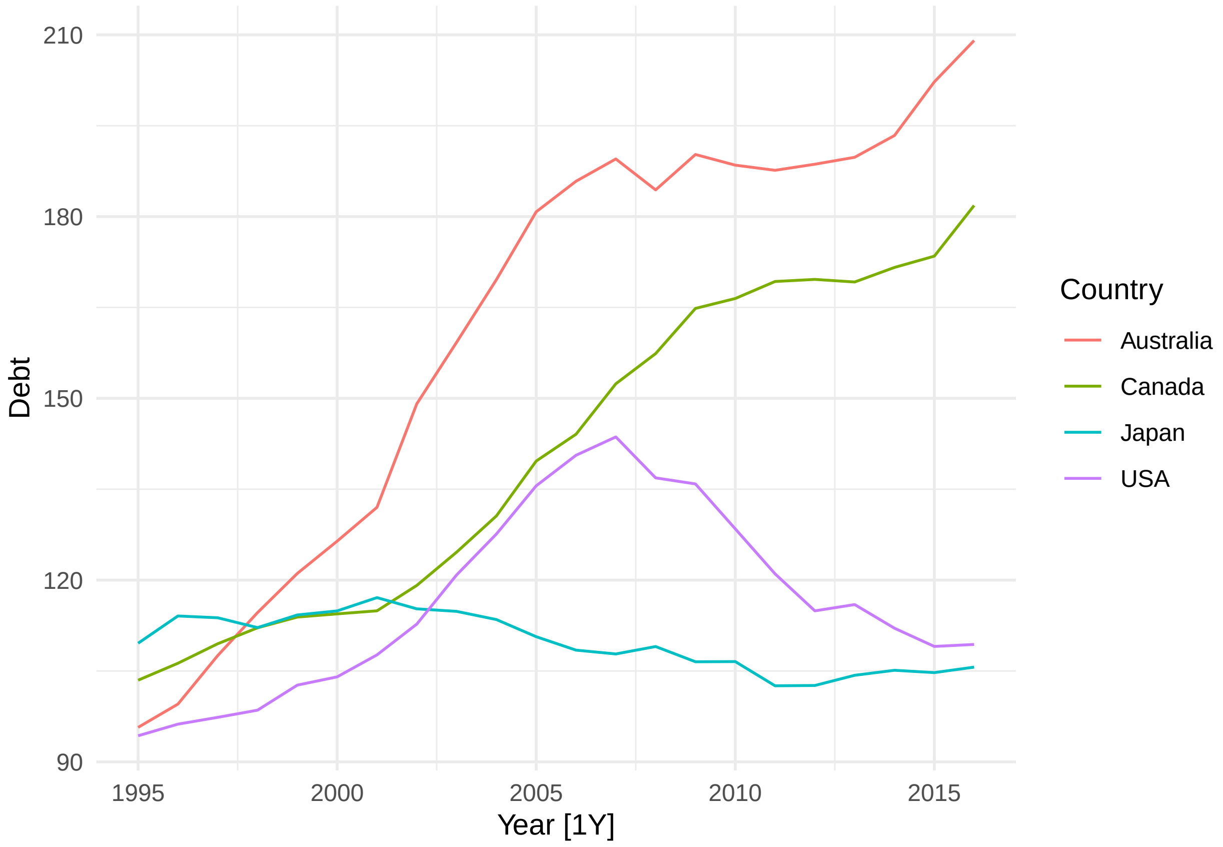
```
# Apply the appropriate method
aus_livestock %>%
  filter(State == "New South Wales",
         Animal == "Lambs") %>%
  model(NAIVE(Count)) %>%
  forecast(h = 12) %>%
  autoplot(aus_livestock) +
  labs(title = "Lambs in New South Wales",     # Update title accordingly
       subtitle = "July 1976 - Dec 2018, Forecasted until Dec 2020") +
  theme_minimal()
```



Household wealth (hh_budget).

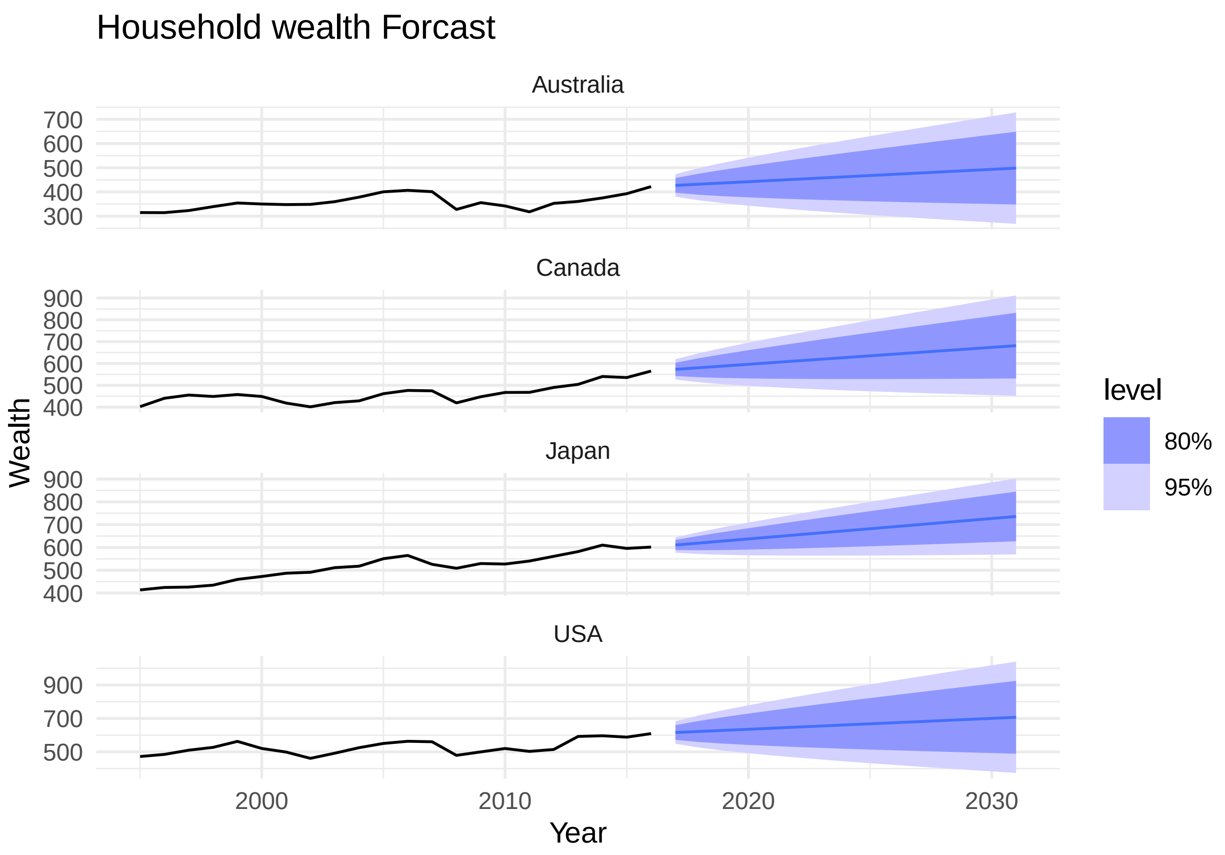
```
# Plot the time series
autoplot(hh_budget)+
  theme_minimal()
```

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



Household wealth shows a long-term trend rather than seasonal variations, so using random walk with drift (RW(y ~ drift())) would be a better fit.

```r
# Apply the appropriate method

hh_budget %>%
  model(Drift = RW(Wealth ~ drift())) %>%
  forecast(h = 15) %>%
  autoplot(hh_budget) +
    labs(title = "Household wealth Forcast")+
  theme_minimal()
```



Australian takeaway food turnover (aus_retail).

Retail turnover often follows seasonal trends. Thus, the seasonal naive method (SNAIVE(y)) would be appropriate here.

```
unique(aus_retail$State)
```

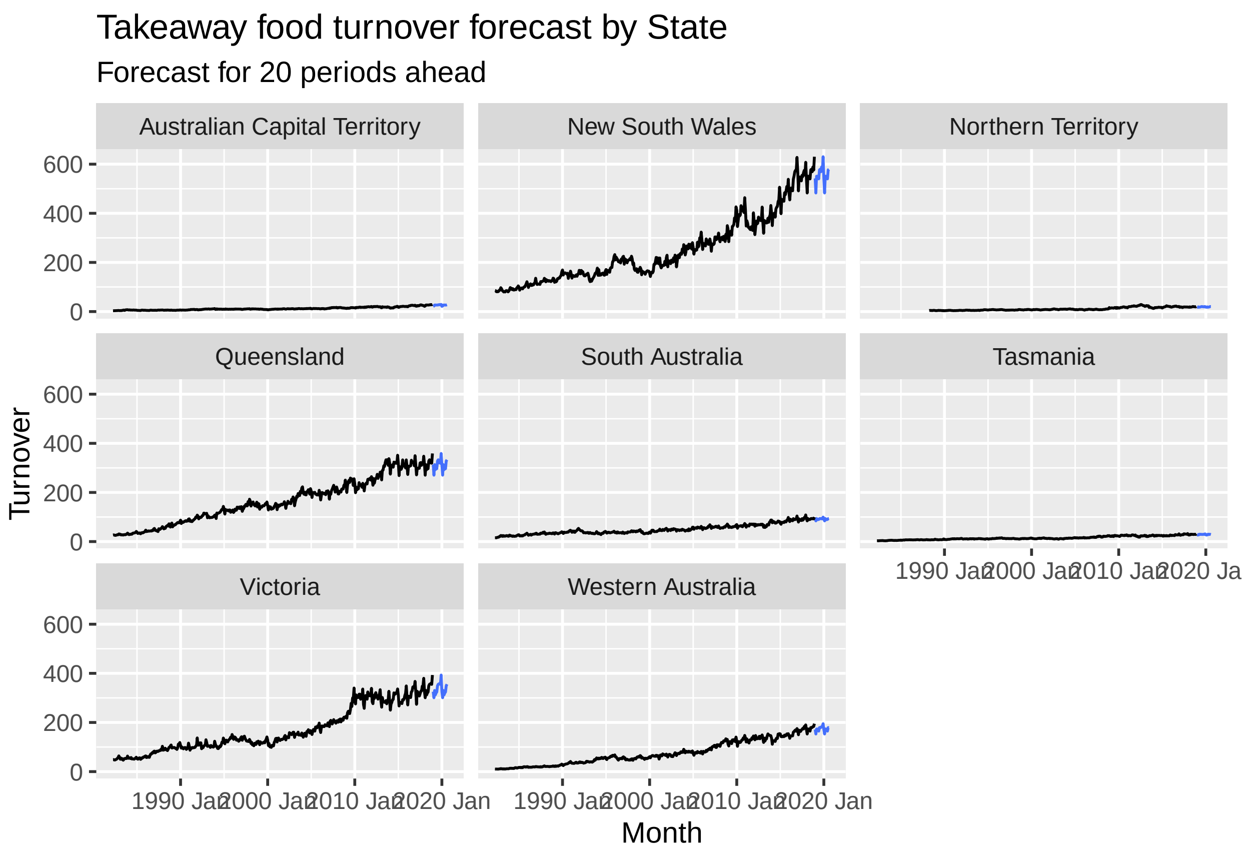
```
## [1] "Australian Capital Territory" "New South Wales"
## [3] "Northern Territory"           "Queensland"
## [5] "South Australia"              "Tasmania"
## [7] "Victoria"                     "Western Australia"
```

```
# Plot the time series f

autoplot(aus_retail)+
  theme_minimal()
```

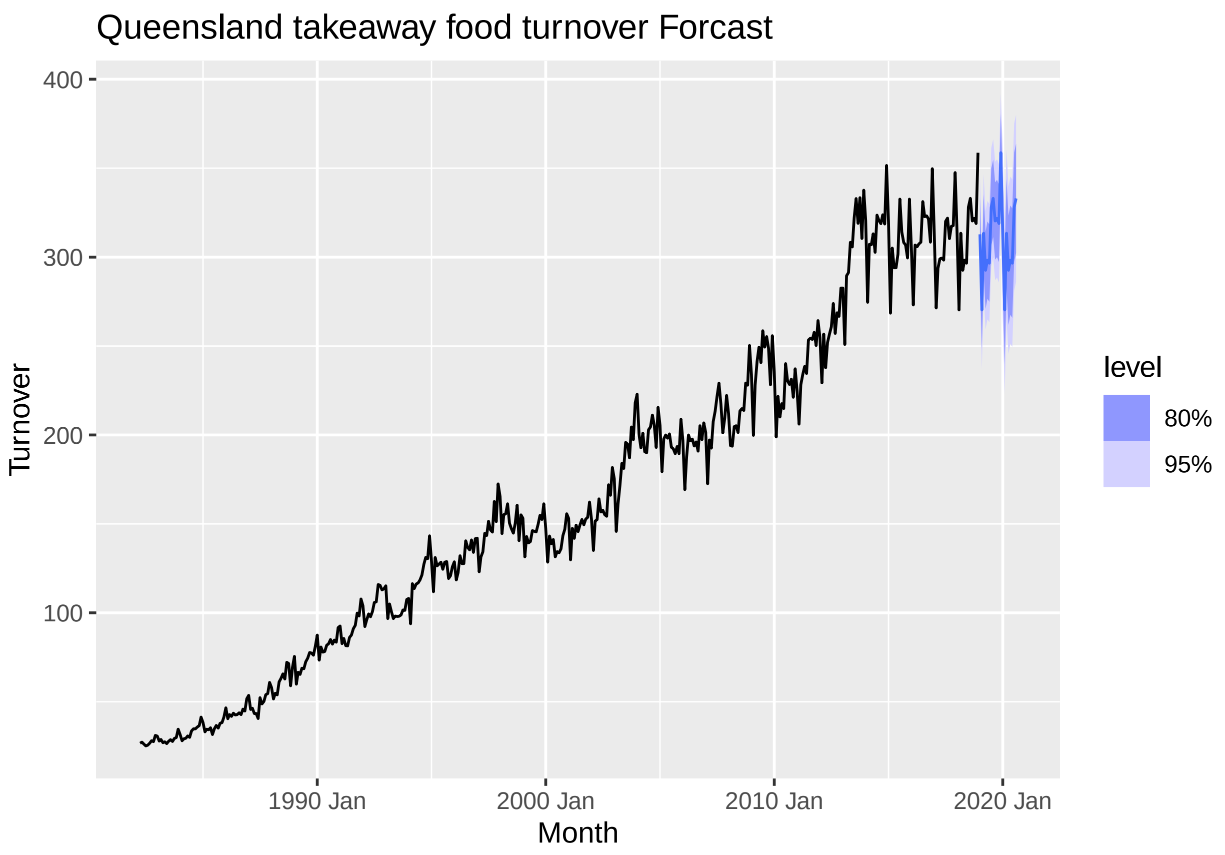
```
## Plot variable not specified, automatically selected `.vars = Turnover`
```

ısland/Footwear and other personal accessory retailing
Australia/Footwear and other personal accessory retailing
.nia/Footwear and other personal accessory retailing
a/Footwear and other personal accessory retailing
rn Australia/Footwear and other personal accessory retailing
lian Capital Territory/Furniture, floor coverings, houseware and textile goods retailing
outh Wales/Furniture, floor coverings, houseware and textile goods retailing
ərn Territory/Furniture, floor coverings, houseware and textile goods retailing
ısland/Furniture, floor coverings, houseware and textile goods retailing
Australia/Furniture, floor coverings, houseware and textile goods retailing
.nia/Furniture, floor coverings, houseware and textile goods retailing
a/Furniture, floor coverings, houseware and textile goods retailing
rn Australia/Furniture, floor coverings, houseware and textile goods retailing
lian Capital Territory/Hardware, building and garden supplies retailing
outh Wales/Hardware, building and garden supplies retailing
ərn Territory/Hardware, building and garden supplies retailing
ısland/Hardware, building and garden supplies retailing
Australia/Hardware, building and garden supplies retailing
.nia/Hardware, building and garden supplies retailing
Victoria/Hardware, building
Western Australia/Hardwa
Australian Capital Territory
New South Wales/Househ
Northern Territory/Househ
Queensland/Household gc
South Australia/Household
Tasmania/Household good
Victoria/Household goods
Western Australia/Househ
Australian Capital Territory
New South Wales/Liquor r
Queensland/Liquor retailin
South Australia/Liquor reta
Tasmania/Liquor retailing
Victoria/Liquor retailing
Western Australia/Liquor r
Australian Capital Territory
New South Wales/Newspa

```r
#Apply the appropriate method and group by each Australian State
aus_retail %>%
  filter(Industry == 'Takeaway food services') %>%
  group_by(State) %>%
  model(SNAIVE(Turnover ~ lag("year"))) %>%
  forecast(h = 20) %>%
  autoplot(aus_retail, level = NULL) +  # Visualize each state's forecast
    labs(title = "Takeaway food turnover forecast by State",
         subtitle = "Forecast for 20 periods ahead") +
    facet_wrap(~ State)  # Creates a separate plot for each state
```



```
# zooming on Queensland
aus_retail %>%
  filter(State == "Queensland",
         Industry == 'Takeaway food services') %>%
  model(SNAIVE(Turnover ~ lag("year"))) %>%
  forecast(h = 20) %>%
  autoplot(aus_retail) +
    labs(title = "Queensland takeaway food turnover Forcast")
```



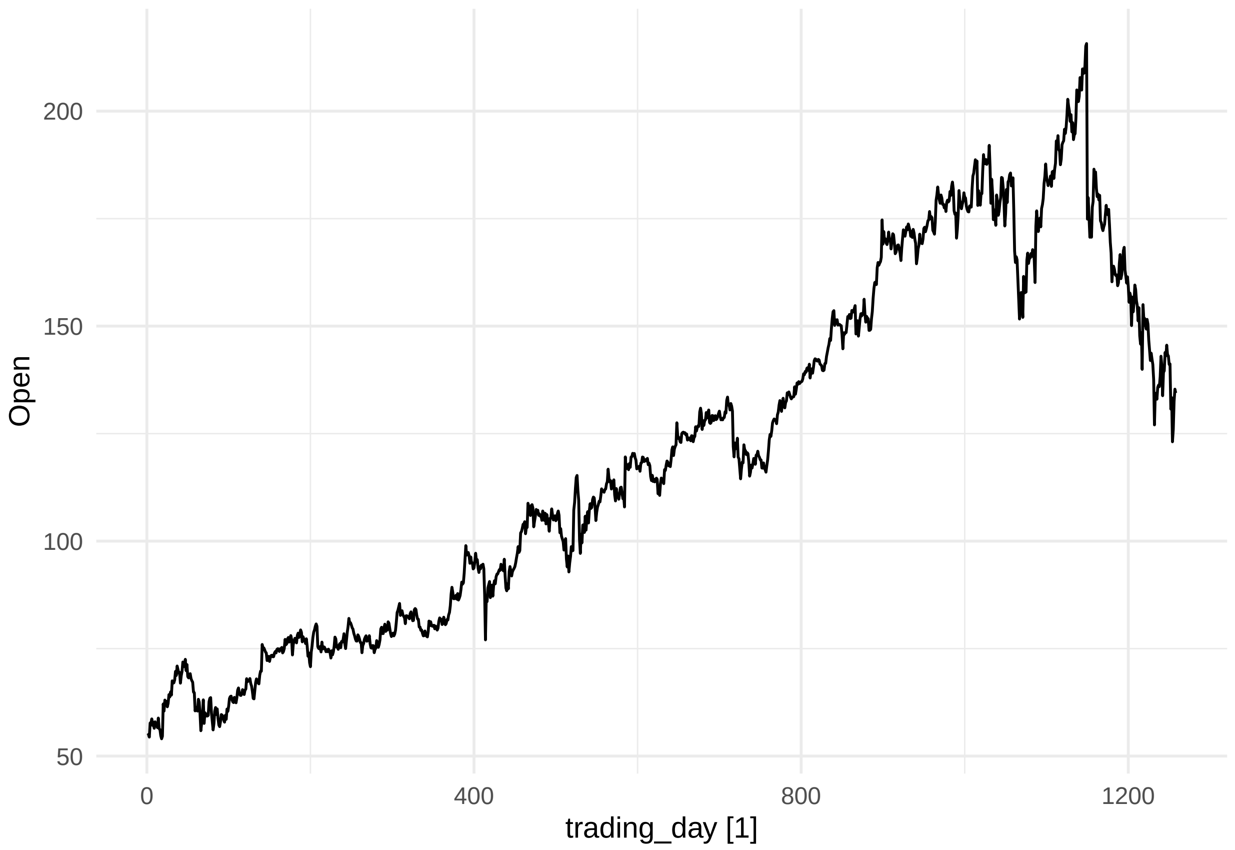
Exercise 5.2

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

Produce a time plot of the series.

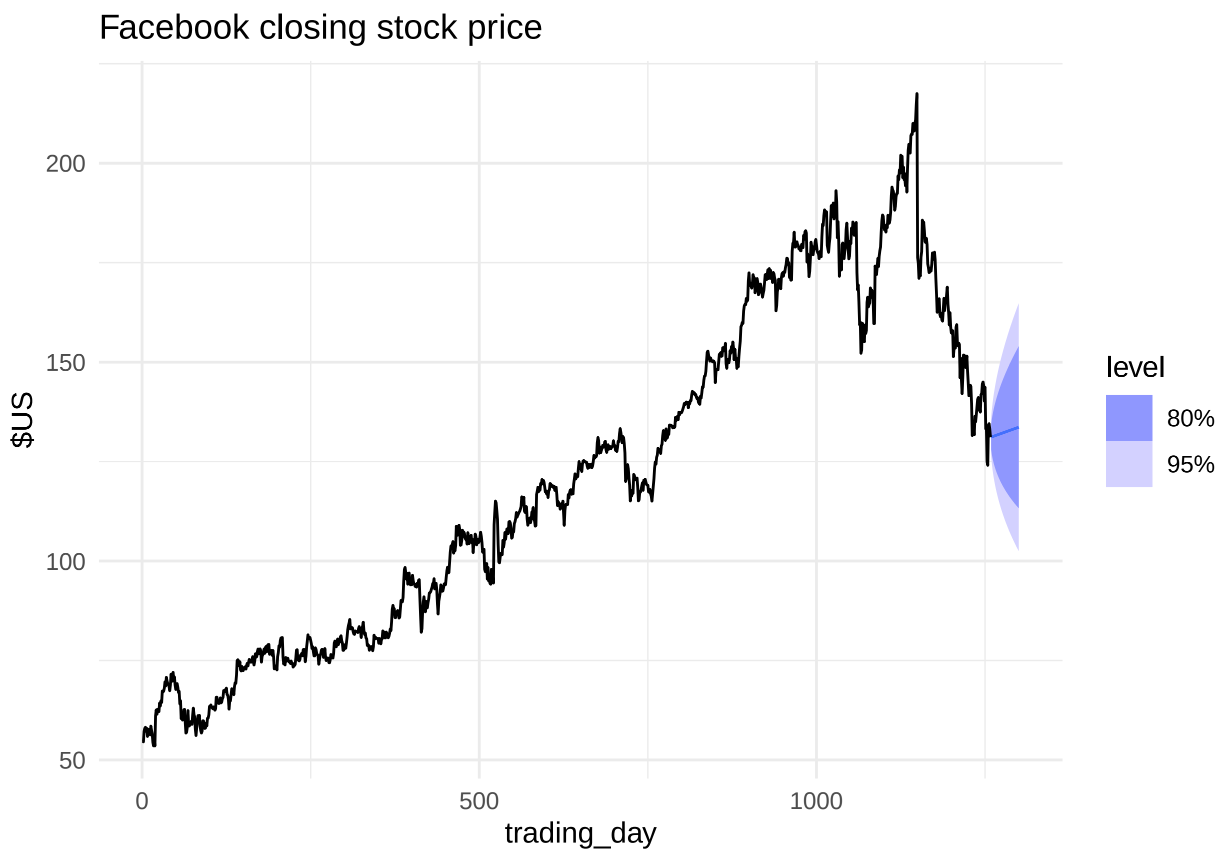
```r
# Extract training data
fb_stock <-gafa_stock %>%
  filter(Symbol == "FB") %>%
 dplyr:: mutate(trading_day = row_number()) %>% update_tsibble(index = trading_day, regular = TRUE)
autoplot(fb_stock)+
  theme_minimal()
```

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



Produce forecasts using the drift method and plot them.

```r
# Specify, estimate and forecast.
fb_stock %>%
  model(Drift = RW(Close~drift())
  ) %>%
  forecast(h = 42) %>%
  autoplot(fb_stock) +

  labs(title = "Facebook closing stock price", y = "$US") +
  theme_minimal()
```

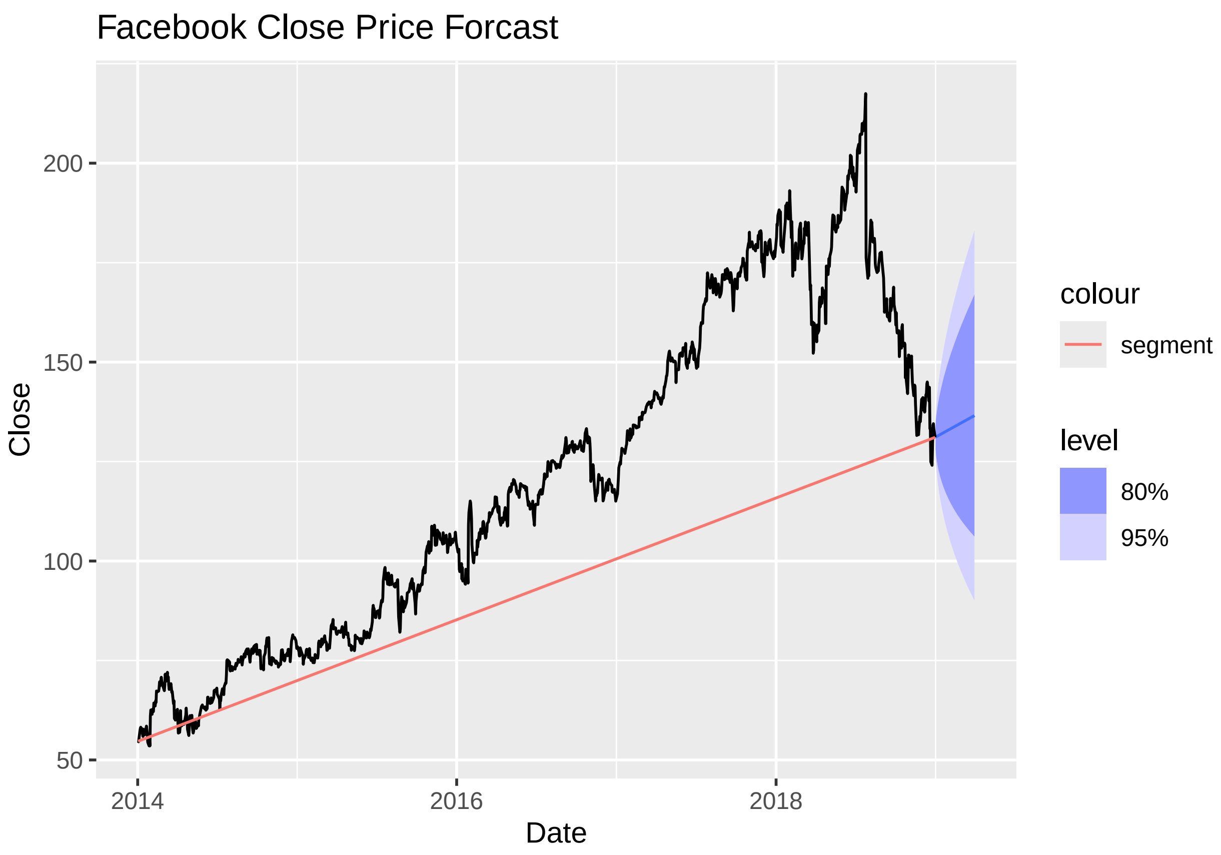


```
  guides(colour = guide_legend(title = "Forecast"))
```

```
## <Guides[1] ggproto object>
##
## colour : <GuideLegend>
```

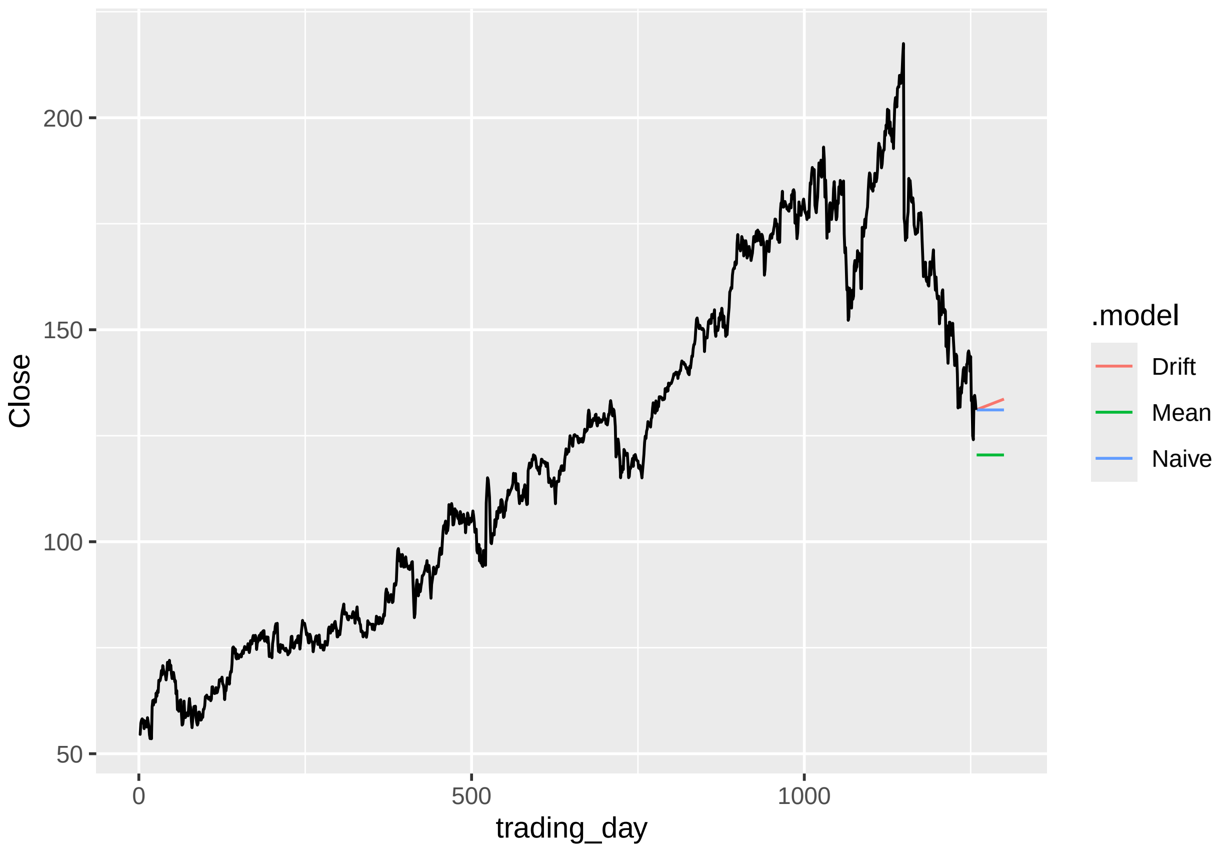
Show that the forecasts are identical to extending the line drawn between the first and last observations.

```
fb_data <- gafa_stock %>%
  filter(Symbol == "FB")

fb_stock <- as_tsibble(fb_data, key = "Symbol", index = "Date", regular = TRUE) %>% fill_gaps()


df <- data.frame(x1 = as.Date('2014-01-02'), x2 = as.Date('2018-12-31'), y1 = 54.71, y2 = 131.09)

fb_stock %>%
model(Drift = RW(Close ~ drift())) %>%
  forecast(h = 90) %>%
  autoplot(fb_data) +
  labs(title = "Facebook Close Price Forcast") +
  geom_segment(aes(x = x1, y = y1, xend = x2, yend = y2, colour = "segment"), data = df)
```



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

```
# Extract training data
fb_stock <-gafa_stock %>%
  filter(Symbol == "FB") %>%
 dplyr:: mutate(trading_day = row_number()) %>% update_tsibble(index = trading_day, regular = TRUE)

# Specify, estimate and forecast
fb_stock %>%
  model(
    Mean = MEAN(Close),
    Naive = NAIVE(Close),
    Drift = RW(Close~drift())
  ) %>%
  forecast(h = 42) %>%
  autoplot(fb_stock, level = NULL)
```



```
  labs(title = "Facebook closing stock price", y = "$US")
```

```
## $y
## [1] "$US"
##
## $title
## [1] "Facebook closing stock price"
##
## attr(,"class")
## [1] "labels"
```

```
  guides(colour = guide_legend(title = "Forecast"))
```

```
## <Guides[1] ggproto object>
##
## colour : <GuideLegend>
```

Which do you think is best? Why?

ANSWER: The drift and naive methods are the best as they start off where the most recent observation is.

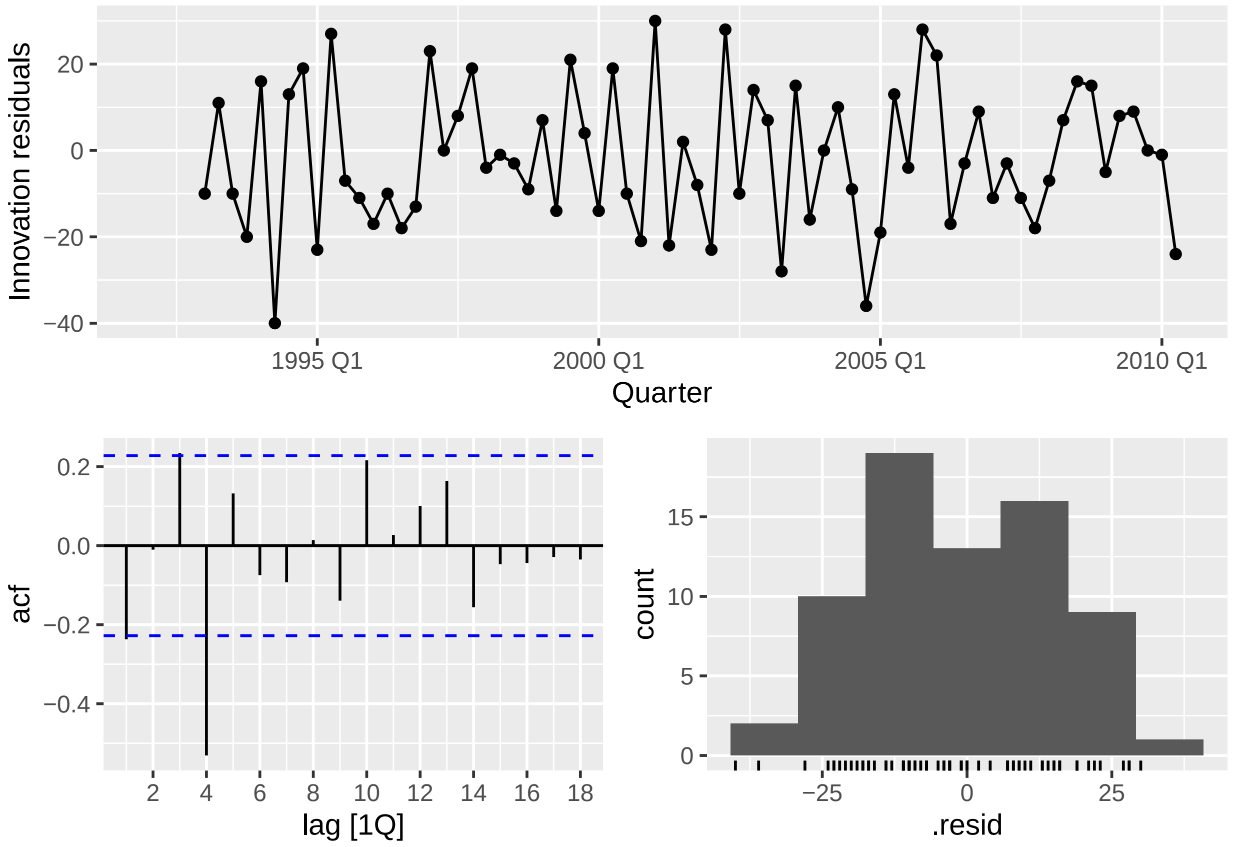
Exercise 5.3

Apply a seasonal NAIVE method to the quarterly Australian beer production data from 1992. Check if the residuals look like white noise, and plot the forecasts.

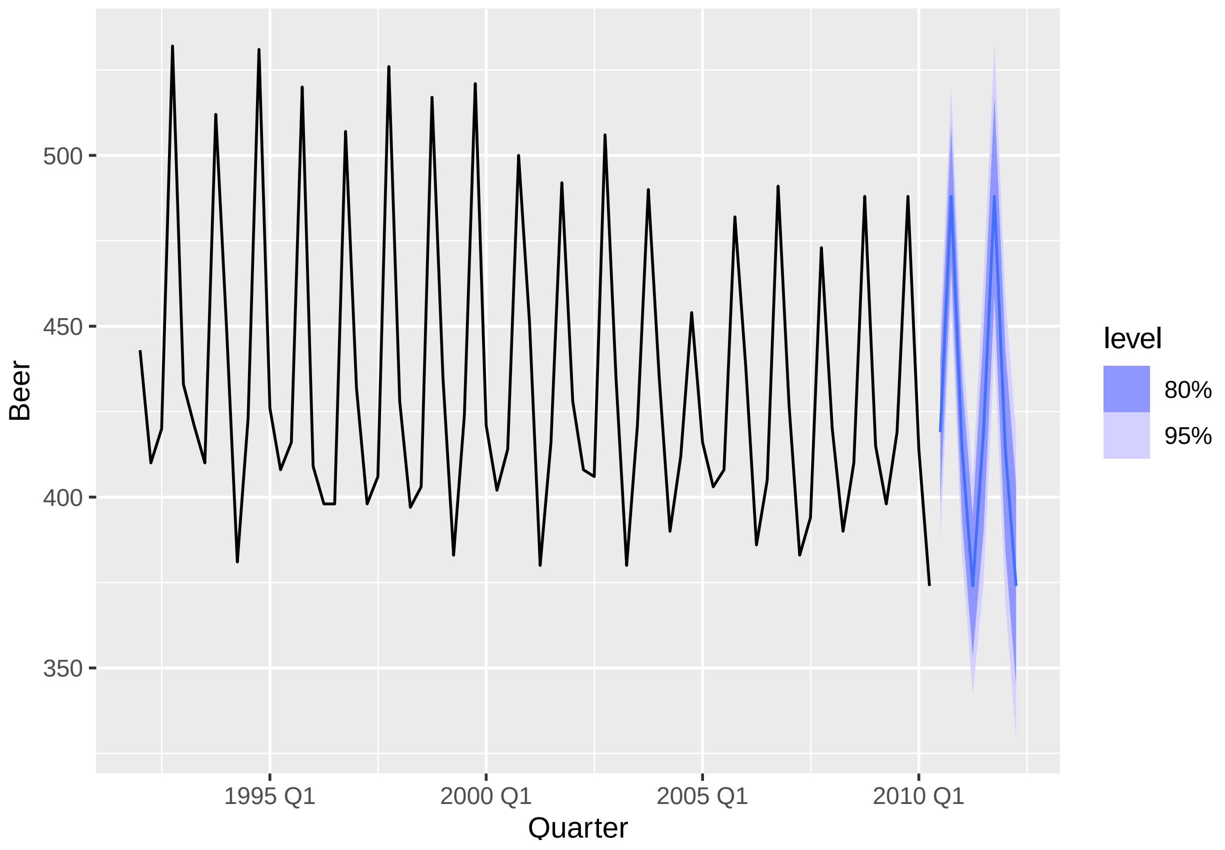
```
 # Extract data of interest
recent_production <- aus_production |>
  filter(year(Quarter) >= 1992)
```

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



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



What do you conclude?

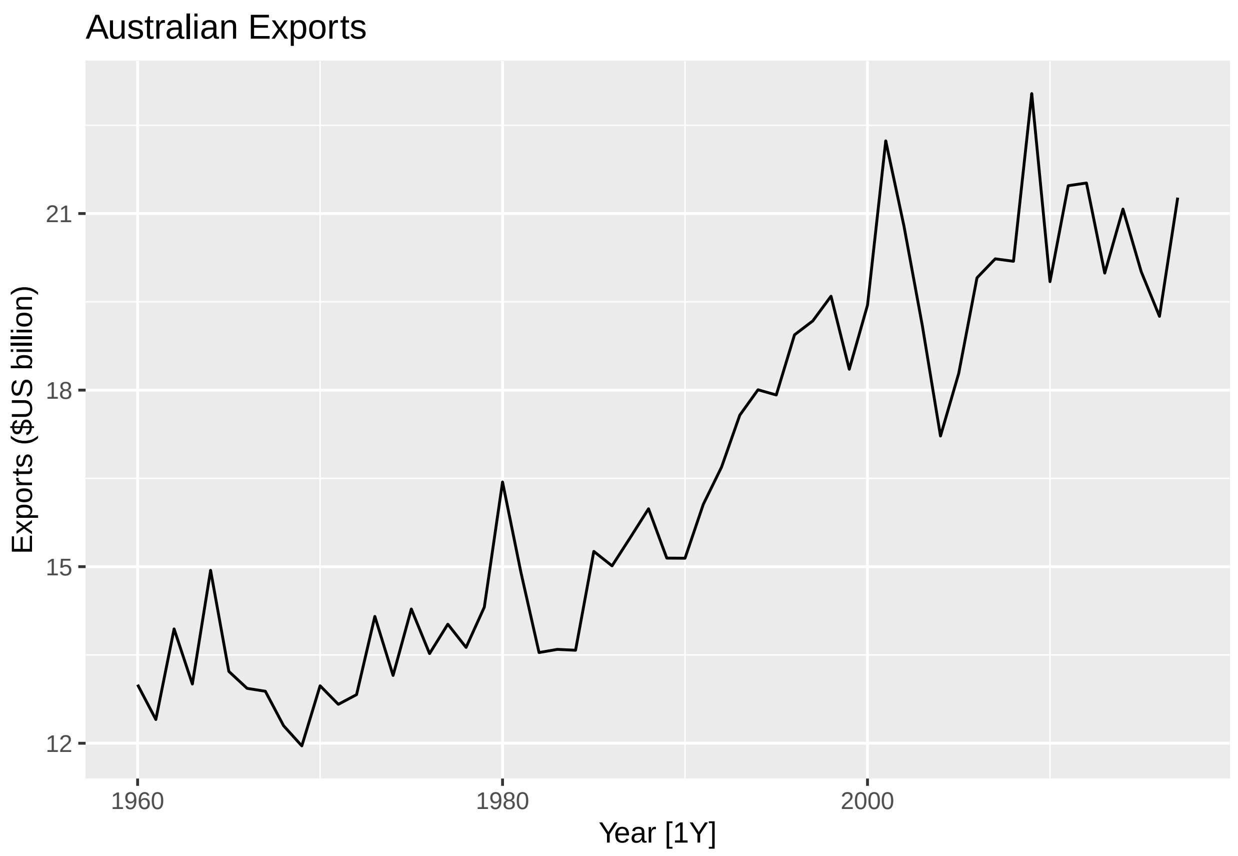
The plots provide the evidence that beer production in Australia has consistent seasonal patterns, thus confirming the seasonal assumption of the SNAIVE model. The residual analysis seems to validate this assumption by showing randomly distributed residuals with no significant patterns. The histogram plot also shows normally distributed values. The forecast plots show how well the SNAIVE model performs. In this scenario, the future values closely follow past seasonal patterns,indicating that the model will be effective for short-term forecasting. The ACF plot shows that lag 4 is larger than the others which can be attributed to peaks occurring every 4 quarters in Q4.

Exercise 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.

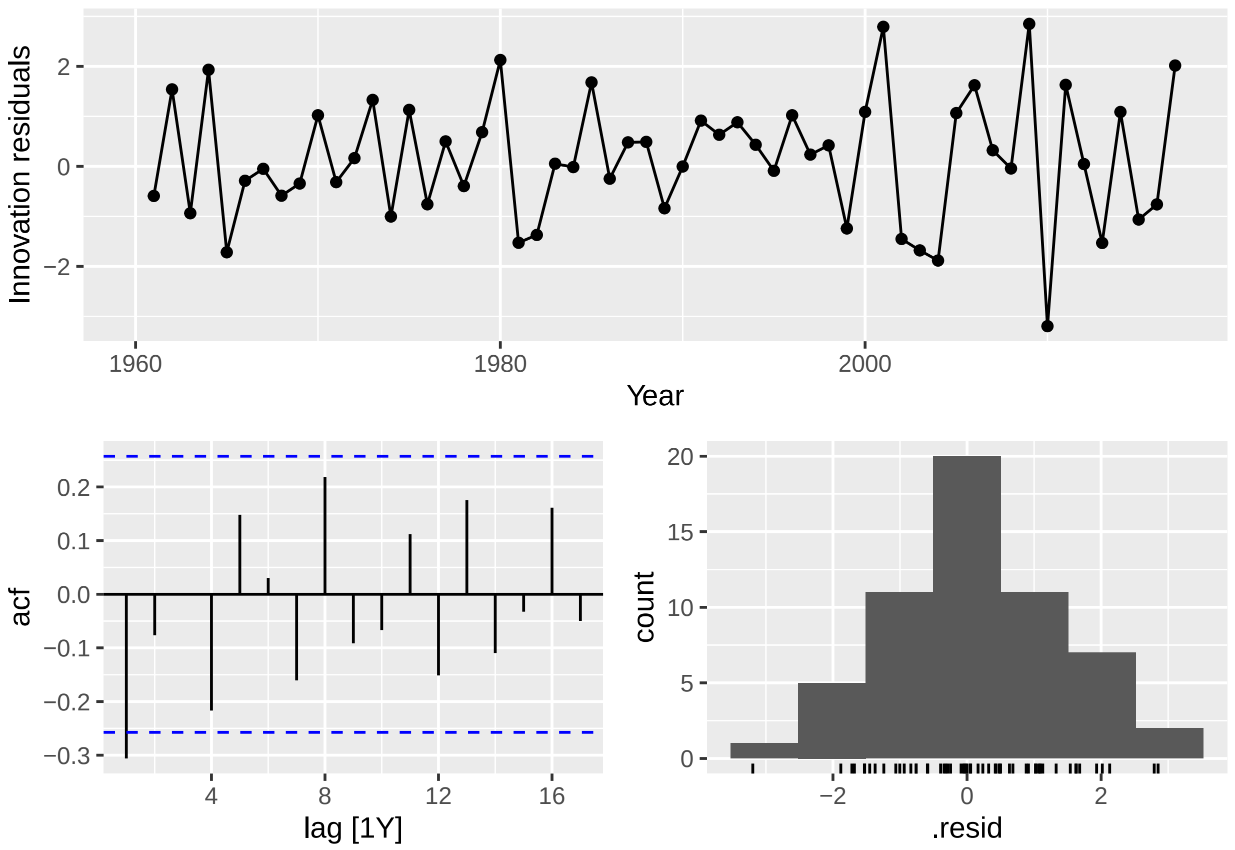
Australian Exports Series (global_economy)

```r
# Extract Australian exports data and remove missing data
australian_exports <- global_economy |>
  dplyr::filter(Country == "Australia") |>
  dplyr::select(Year, Exports) |>
  drop_na(Exports)  # Remove rows with missing Export values

# Visualize the Australian exports data
autoplot(australian_exports, Exports) +
  labs(title = "Australian Exports", y = "Exports ($US billion)")
```

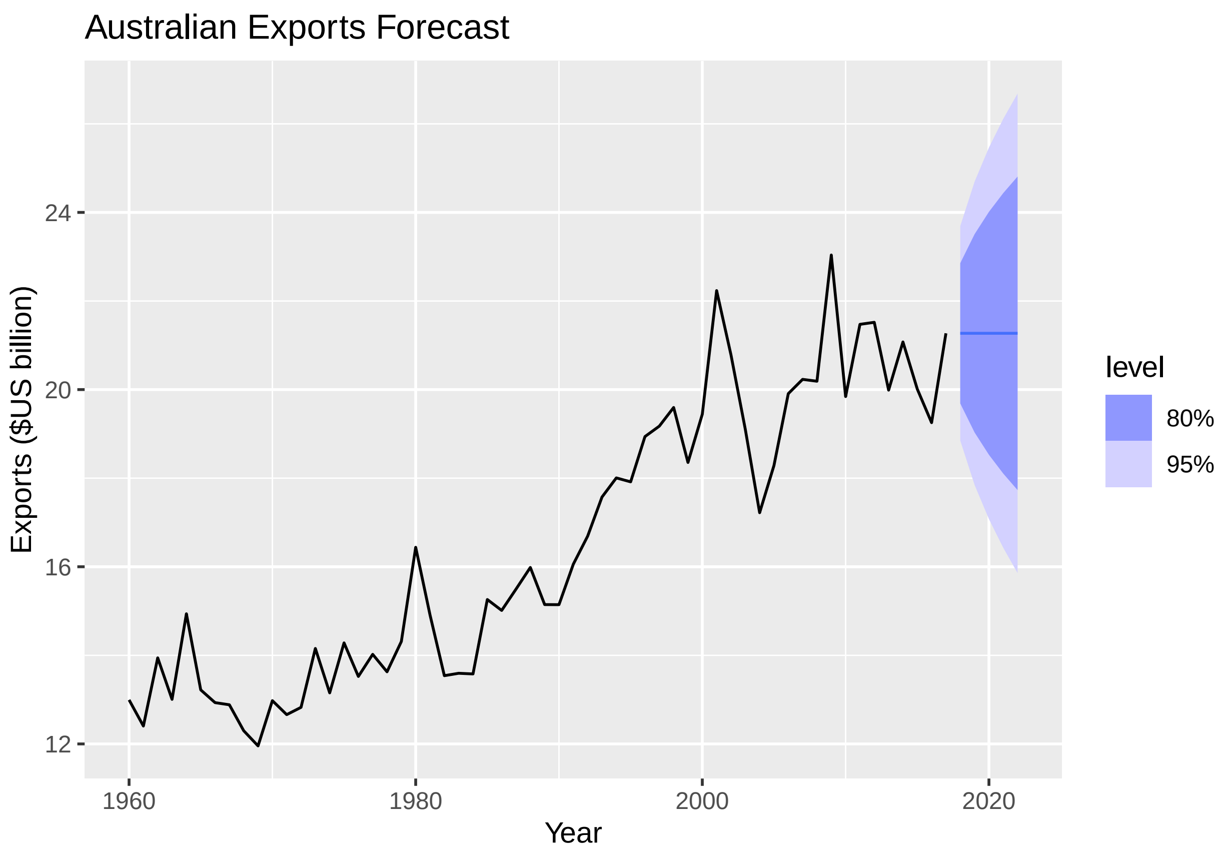


```
# Define and estimate the model - NAIVE
fit_exports <- australian_exports |> model(NAIVE(Exports))

# Look at the residuals
fit_exports |> 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()`).
```



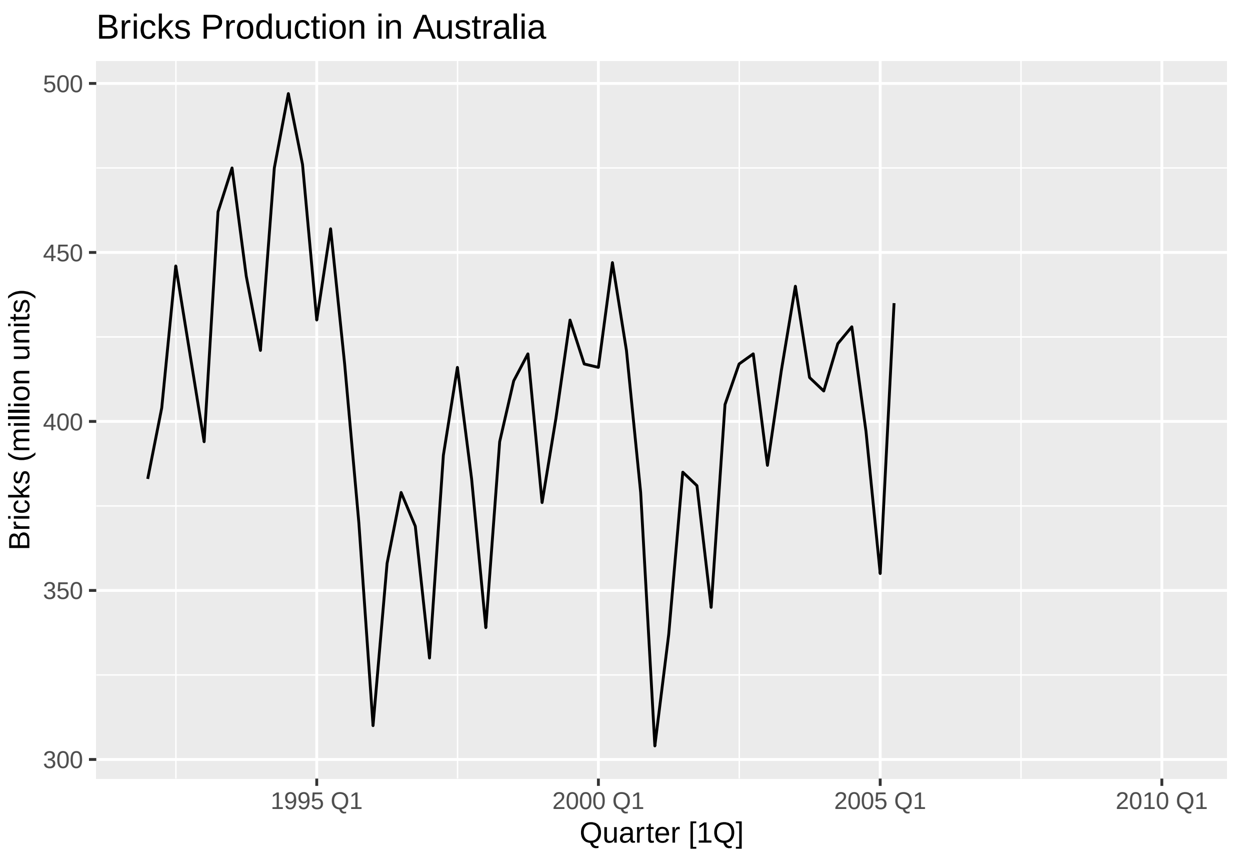
```r
# Forecast and visualize the results
fit_exports |> forecast(h = "5 years") |> autoplot(australian_exports) +
  labs(title = "Australian Exports Forecast", y = "Exports ($US billion)")
```



Bricks Series (aus_production)

```
# Extract the bricks production data
recent_bricks_production <- aus_production |>
 dplyr:: filter(year(Quarter) >= 1992) |>
 dplyr:: select(Quarter, Bricks)

# Visualize the bricks production data
autoplot(recent_bricks_production, Bricks) +
  labs(title = "Bricks Production in Australia", y = "Bricks (million units)")
```

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

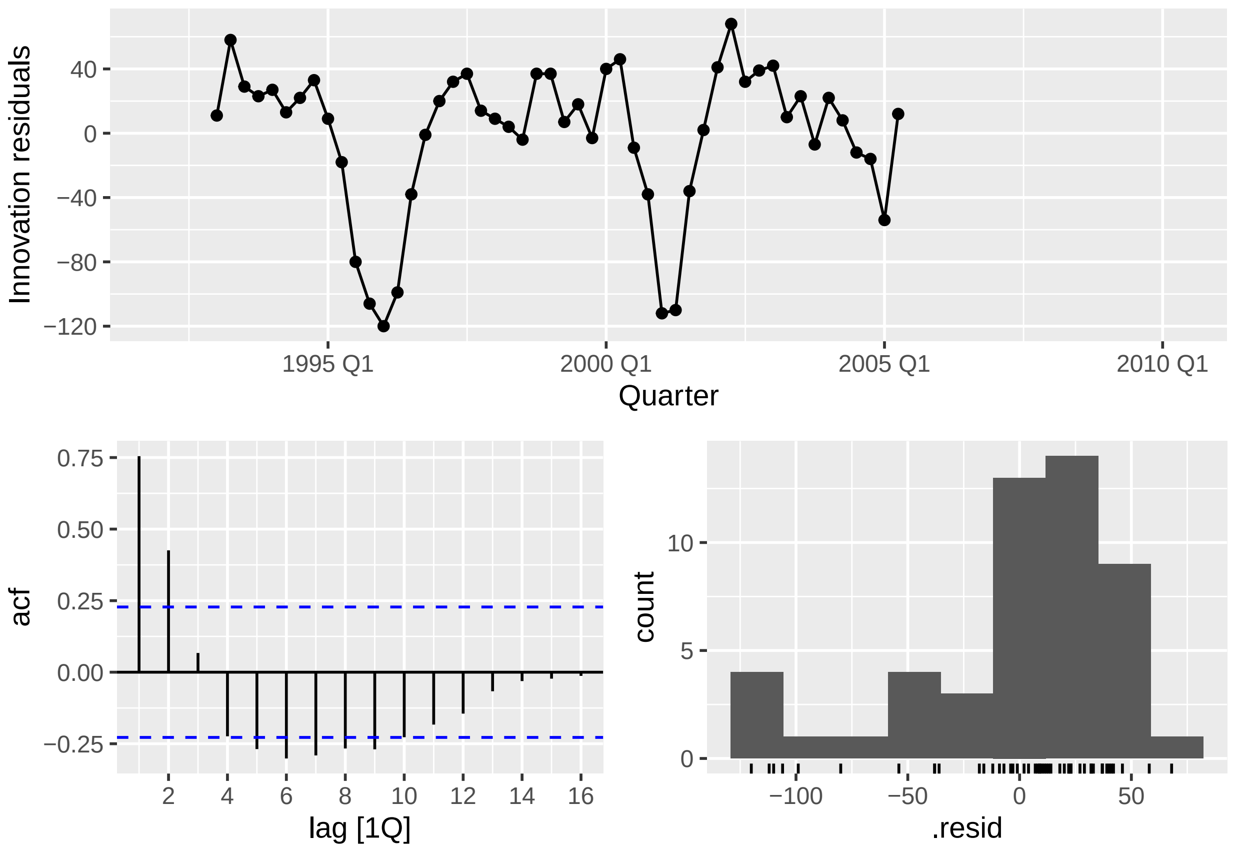


```r
# Define and estimate the model - SNAIVE
fit_bricks <- recent_bricks_production |> model(SNAIVE(Bricks))

# Look at the residuals
fit_bricks |> gg_tsresiduals()
```

```
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

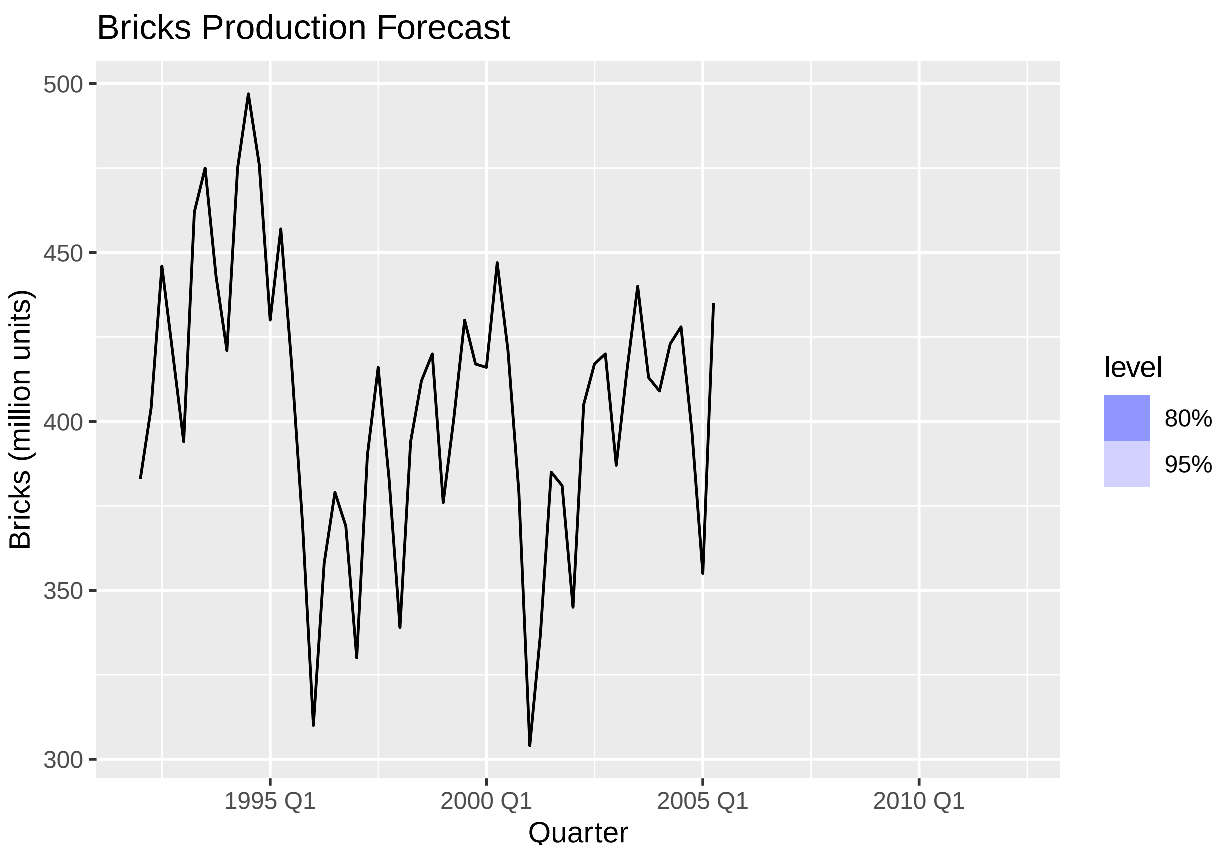
```
## Warning: Removed 24 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 24 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



```r
# Forecast and visualize the results
fit_bricks |> forecast(h = "2 years") |> autoplot(recent_bricks_production) +
  labs(title = "Bricks Production Forecast", y = "Bricks (million units)")
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_line()`).

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



Exercise 5.7

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

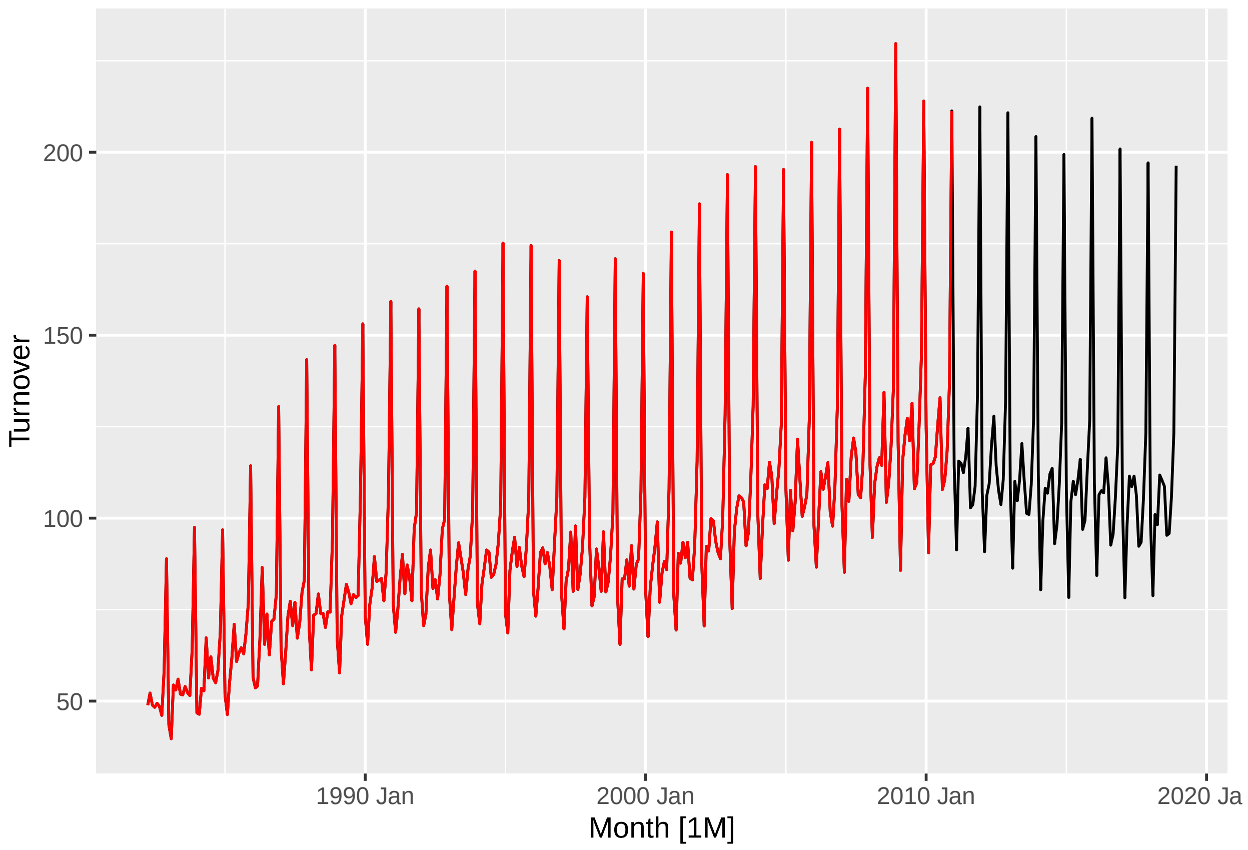
Create a training dataset consisting of observations before 2011 using

```r
set.seed(13101917)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

```r
# Create a training dataset consisting of observations before 2011 using
myseries_train <- myseries |>
  filter(year(Month) < 2011)
```

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

```r
# Check that your data have been split appropriately by producing the following plot.
autoplot(myseries, Turnover) +
  autolayer(myseries_train, Turnover, colour = "red")
```

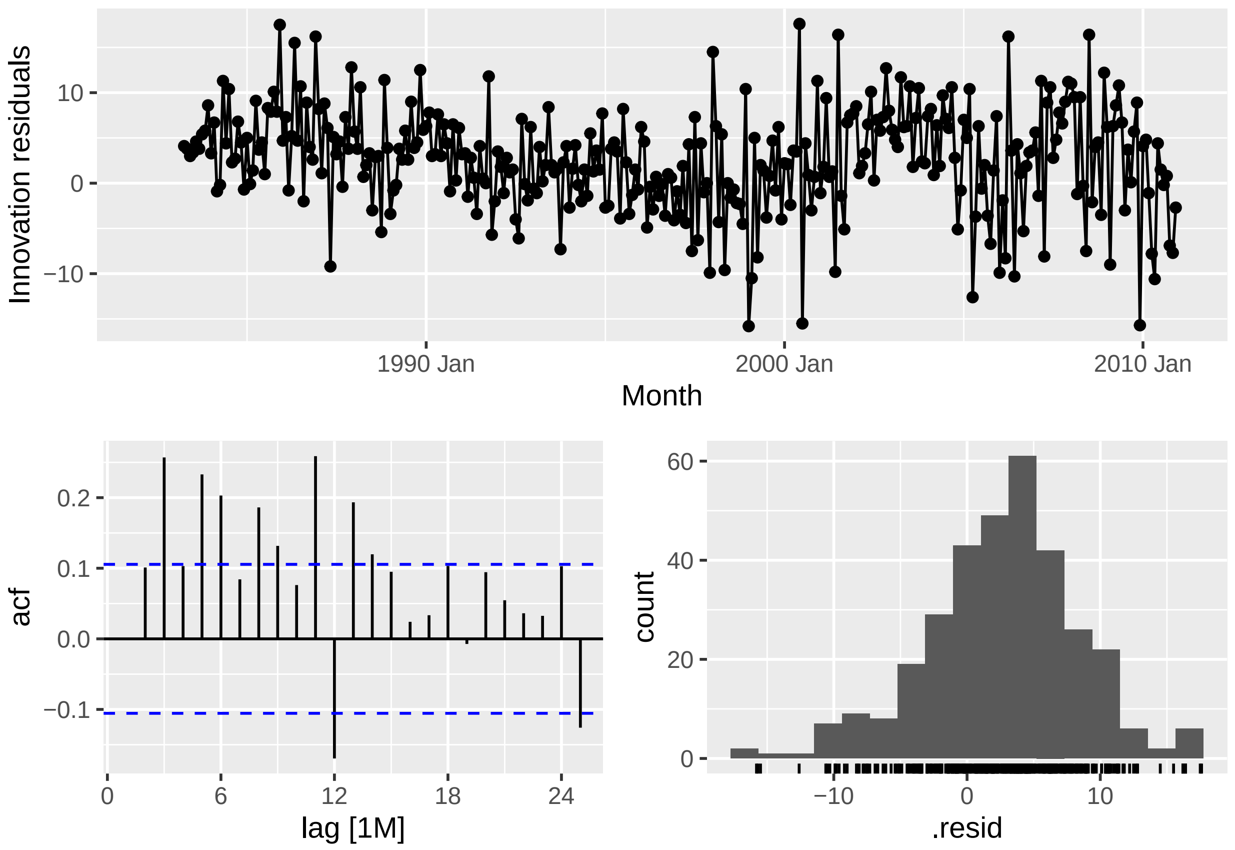


```
# Fit a seasonal naïve model using SNAIVE() applied to your training data (myseries_train).

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

Check the residuals.

```
# Check the 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()`).
```



Do the residuals appear to be uncorrelated and normally distributed?

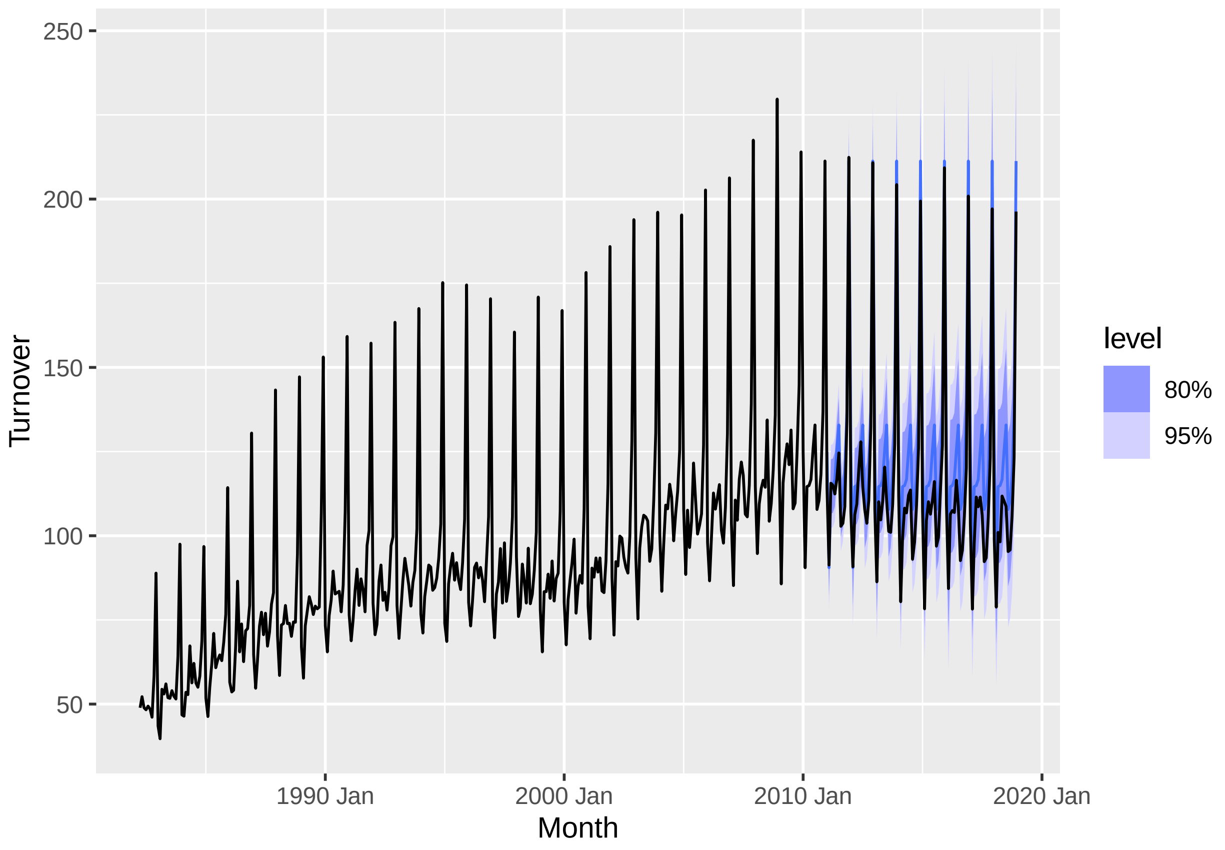
ANSWER: No. The histogram shows right skewed data and the residuals plot reveals move values above zero than below zero. All this suggests that the model may not be the optimal one.

Produce forecasts for the test data

```
# Check the normality of the residuals distribution
fc <- fit |>
  forecast(new_data = anti_join(myseries, myseries_train))
```

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

```
fc |> autoplot(myseries)
```



Compare the accuracy of your forecasts against the actual values.

```r
# Compare the accuracy of your forecasts against the actual values.
fit |> accuracy()
```

```
## # 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 South~ Departm~ SNAIV~ Trai~  2.64  6.30  5.03  2.91  5.47     1     1 1.19e-4
```

```r
fc |> accuracy(myseries)
```

```
## # 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(T~ Sout~ Departm~ Test  -10.5  12.2  10.7 -9.77  9.92  2.13  1.95 0.230
```

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

ANSWER: Very little.The SNAIVE and NAIVE models are less sensitive to training data size because they rely on historical values or seasonal patterns. As a result, they can perform reasonably well with smaller datas sets.