# Data 624 Homework 3

Steven Gonzalez

2/23/2025

### Load Packages

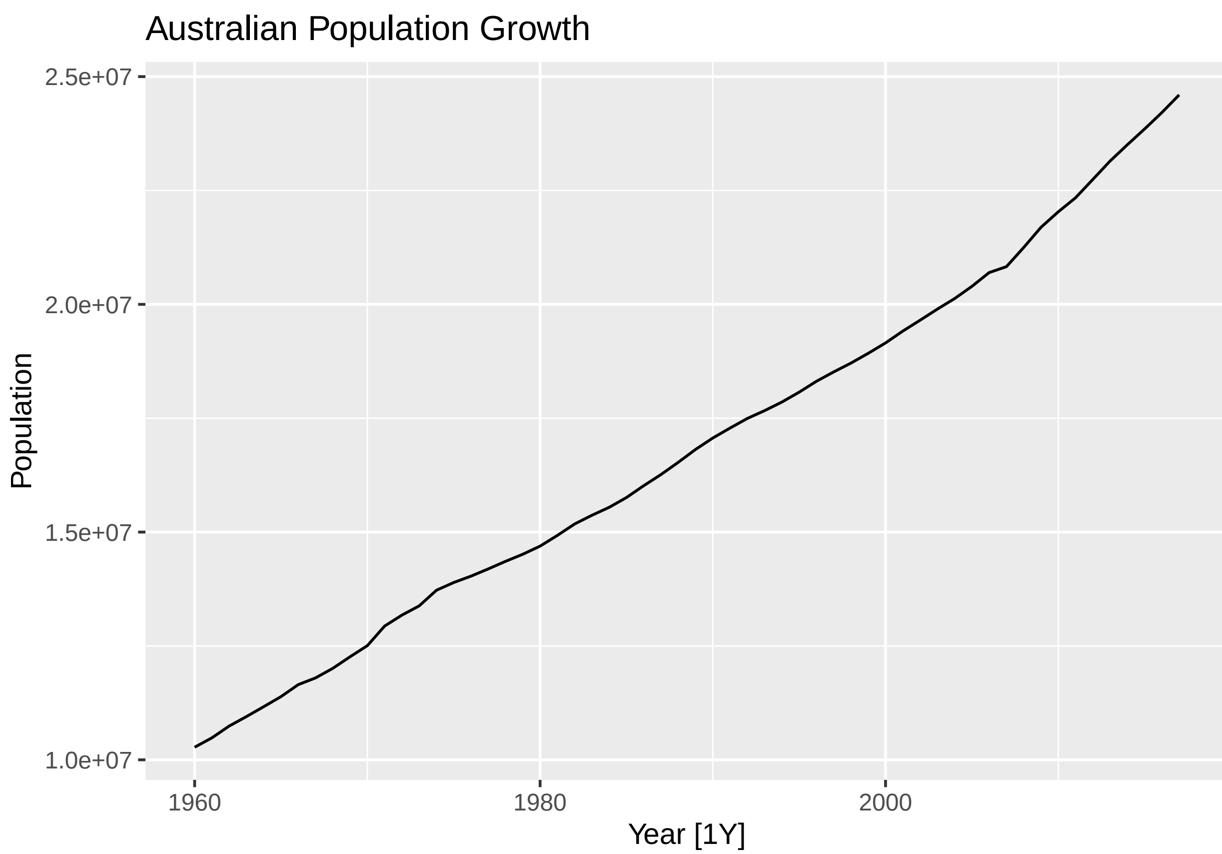
```
library(fpp3)
library(seasonal)
library(USgas)
```

### Exercise 1

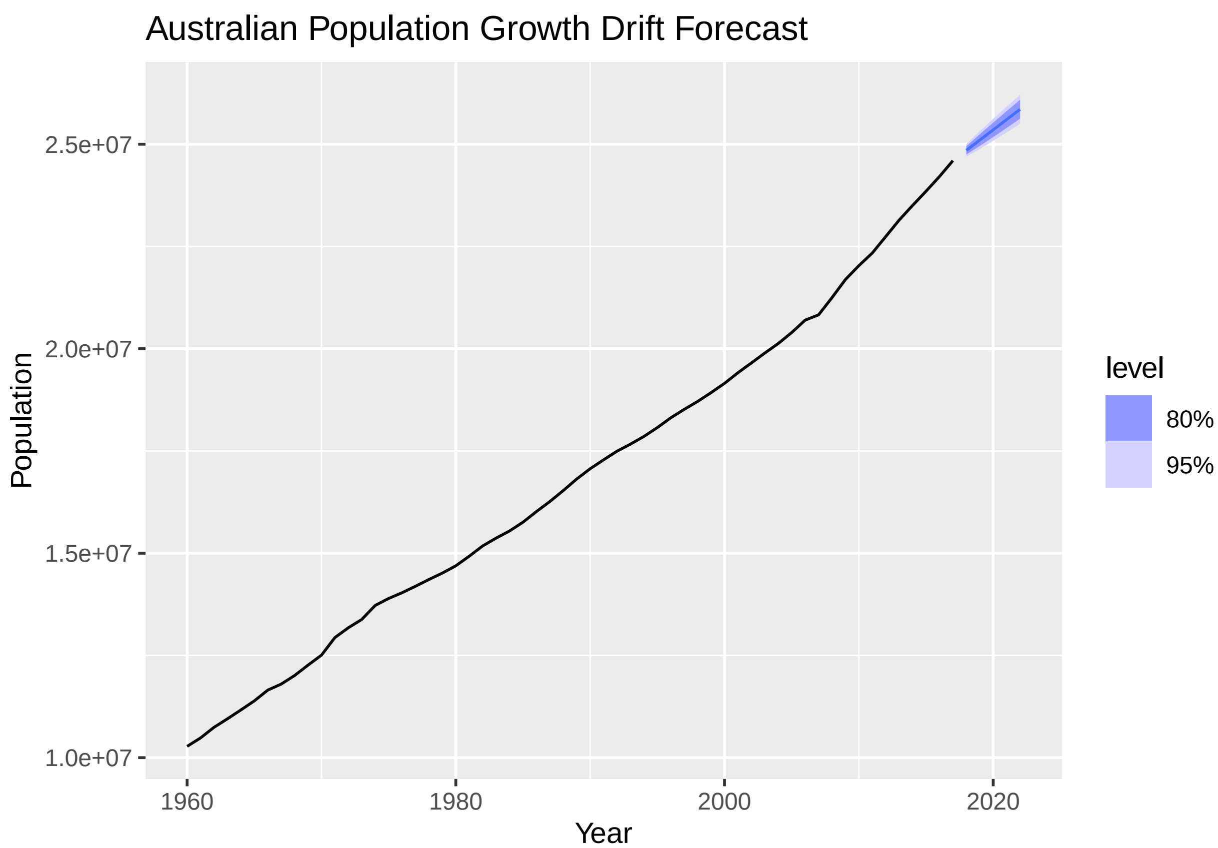
Produce forecasts for the following series using whichever of NAIVE(y), SNAIVE(y) or RW(y ~ drift()) is more appropriate in each case:

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

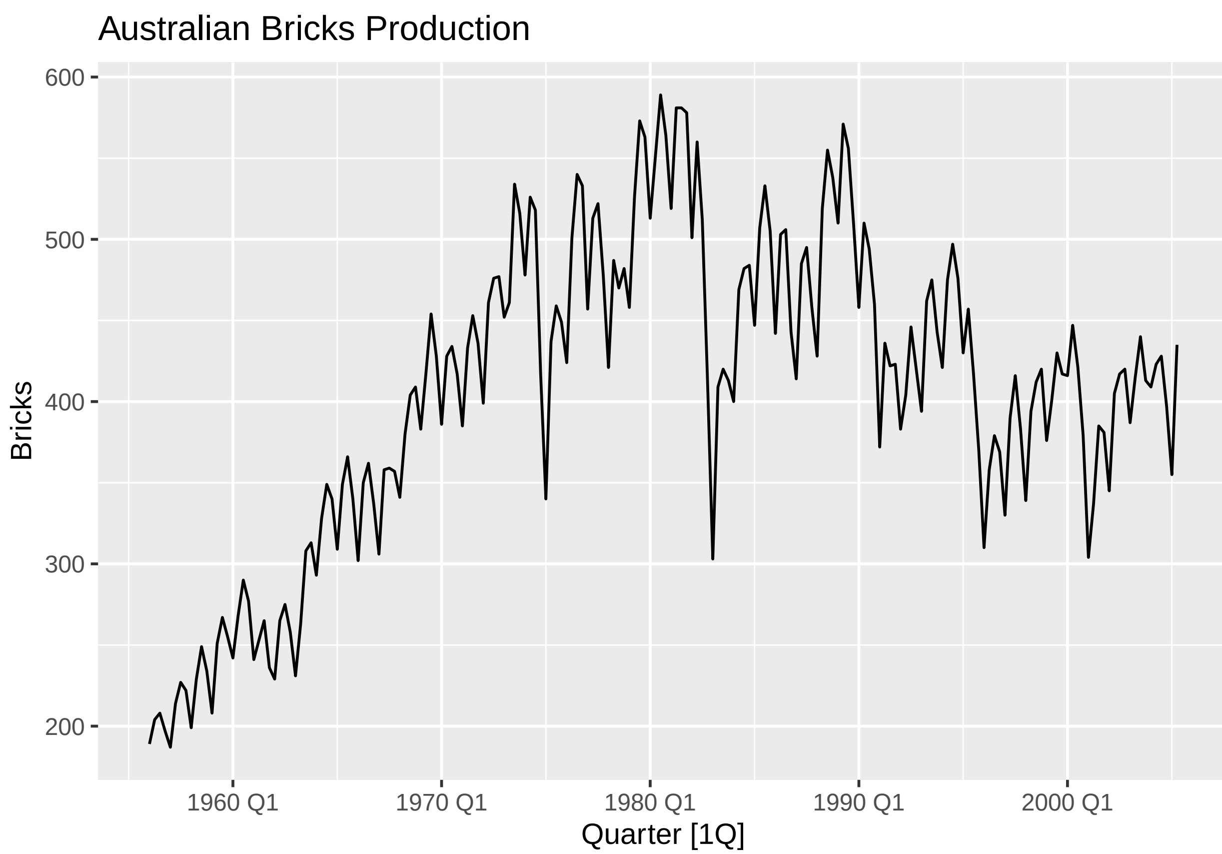
```
aus_pop <- global_economy %>%
  filter(!is.na(Population)) %>%
  filter(Country == "Australia") %>%
  select(Population)

autoplot(aus_pop) +
  labs(title = "Australian Population Growth")
```



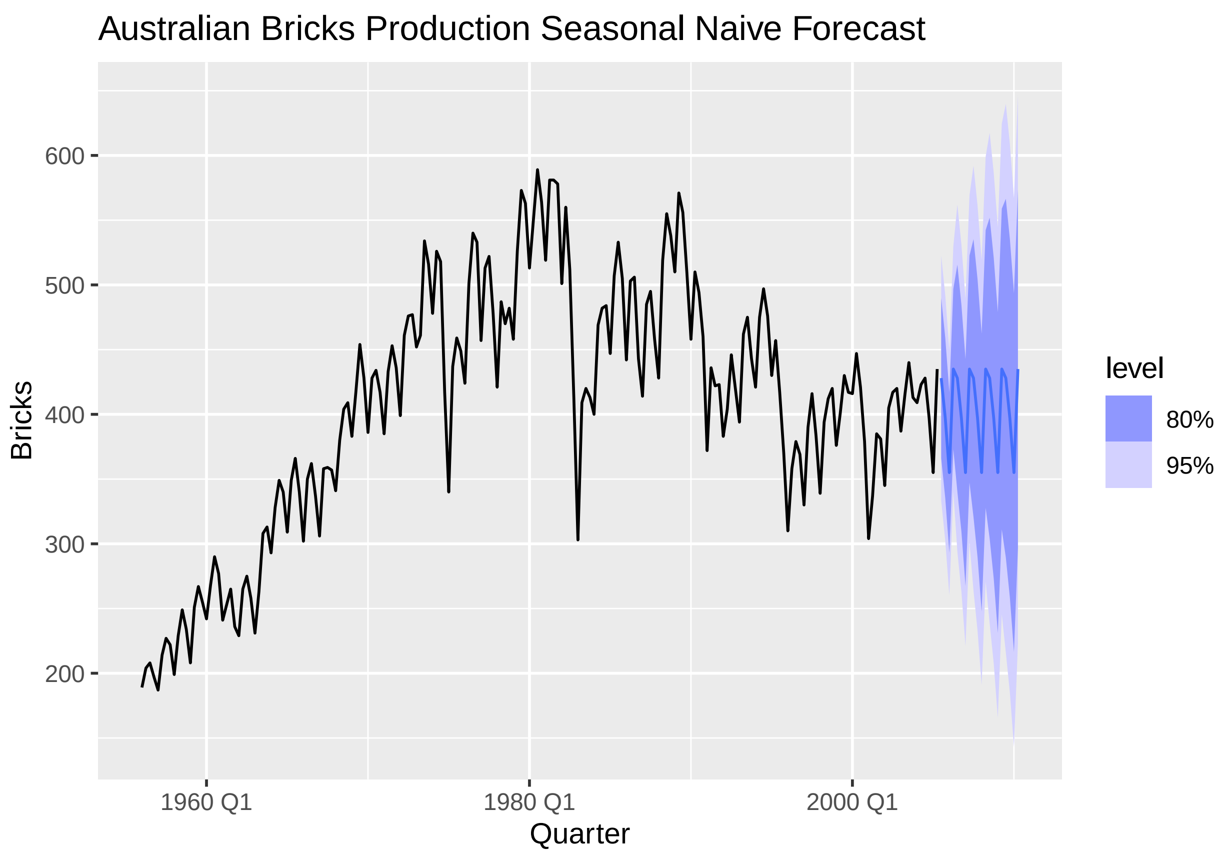

```
aus_pop %>%
  model(Drift = RW(Population ~ drift())) %>%
  forecast(h = "5 years") %>%
  autoplot(aus_pop) +
  labs(title = "Australian Population Growth Drift Forecast")
```



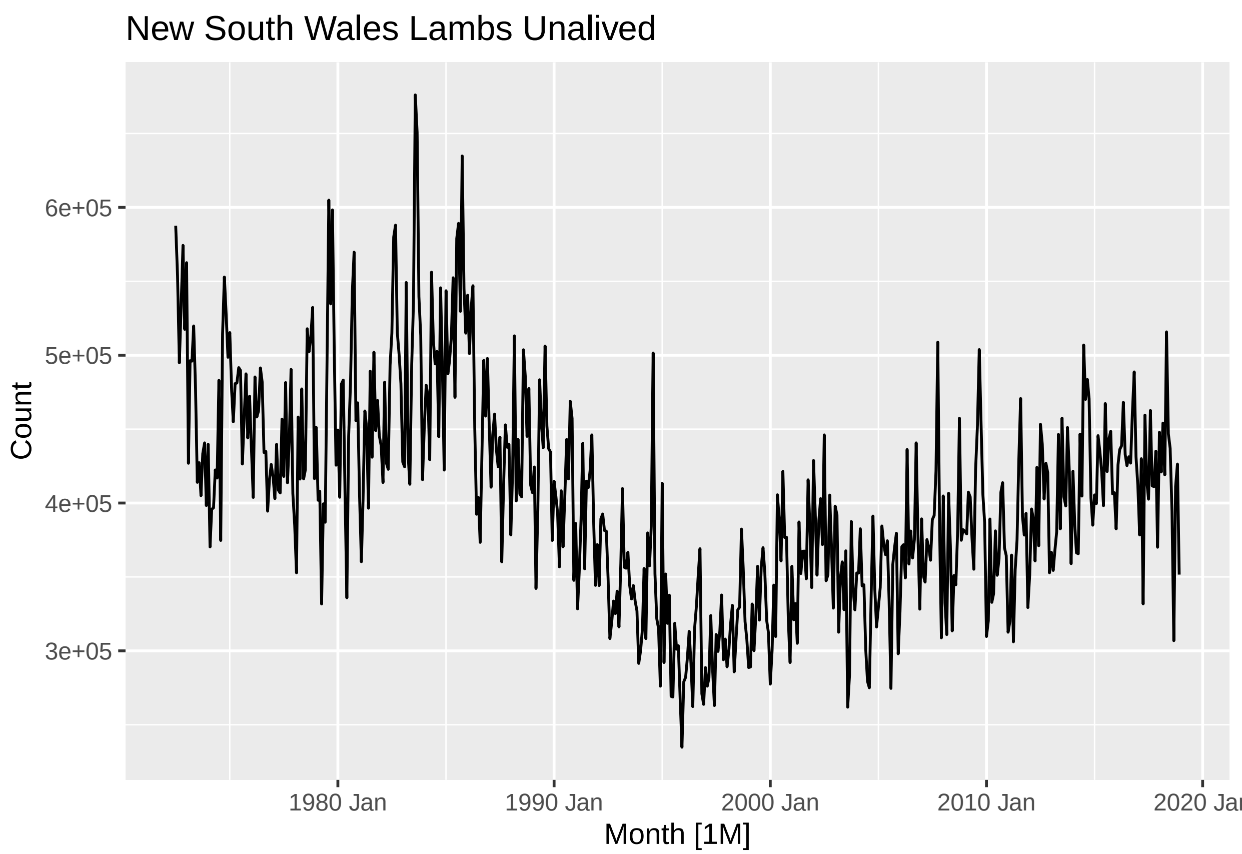

```r
bricks_pro <- aus_production %>%
  filter(!is.na(Bricks)) %>%
  select(Bricks)

autoplot(bricks_pro) +
  labs(title = "Australian Bricks Production")
```



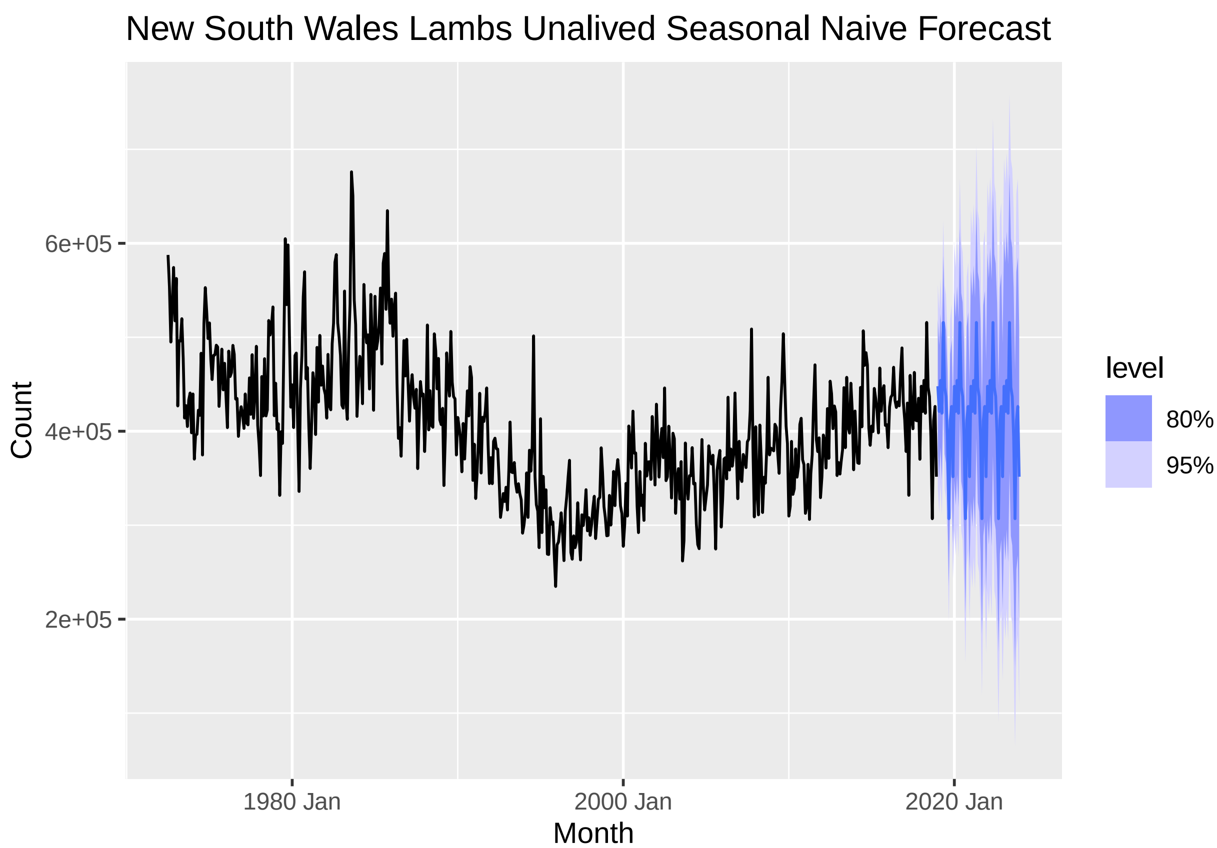

```r
bricks_pro %>%
  model(Snaive = SNAIVE(Bricks)) %>%
  forecast(h = "5 years") %>%
  autoplot(bricks_pro) +
  labs(title = "Australian Bricks Production Seasonal Naive Forecast")
```



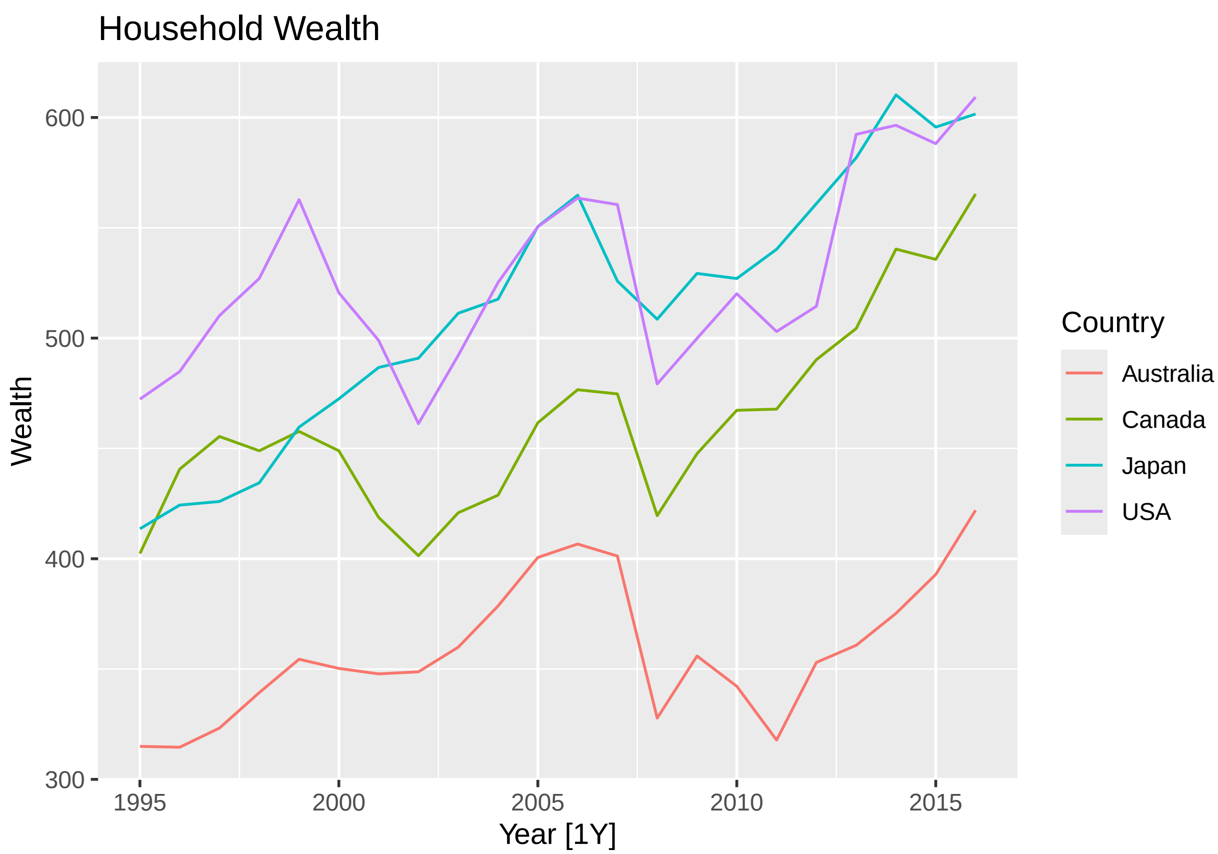

```r
nsw_lambs <- aus_livestock %>%
  filter(!is.na(Count)) %>%
  filter(State == "New South Wales") %>%
  filter(Animal == "Lambs")

autoplot(nsw_lambs) +
  labs(title = "New South Wales Lambs Unalived")
```



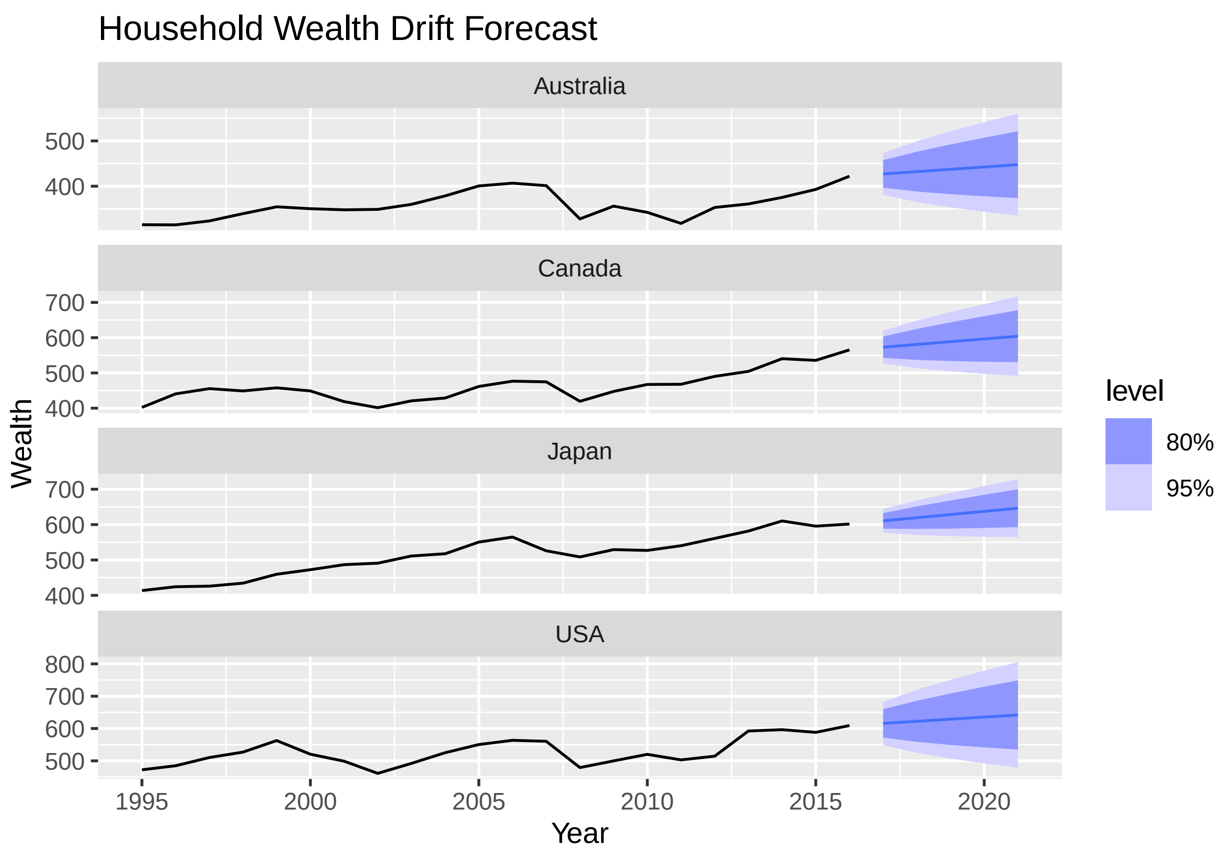

```
nsw_lambs %>%
  model(Snaive = SNAIVE(Count)) %>%
  forecast(h = "5 years") %>%
  autoplot(nsw_lambs) +
  labs(title = "New South Wales Lambs Unalived Seasonal Naive Forecast")
```



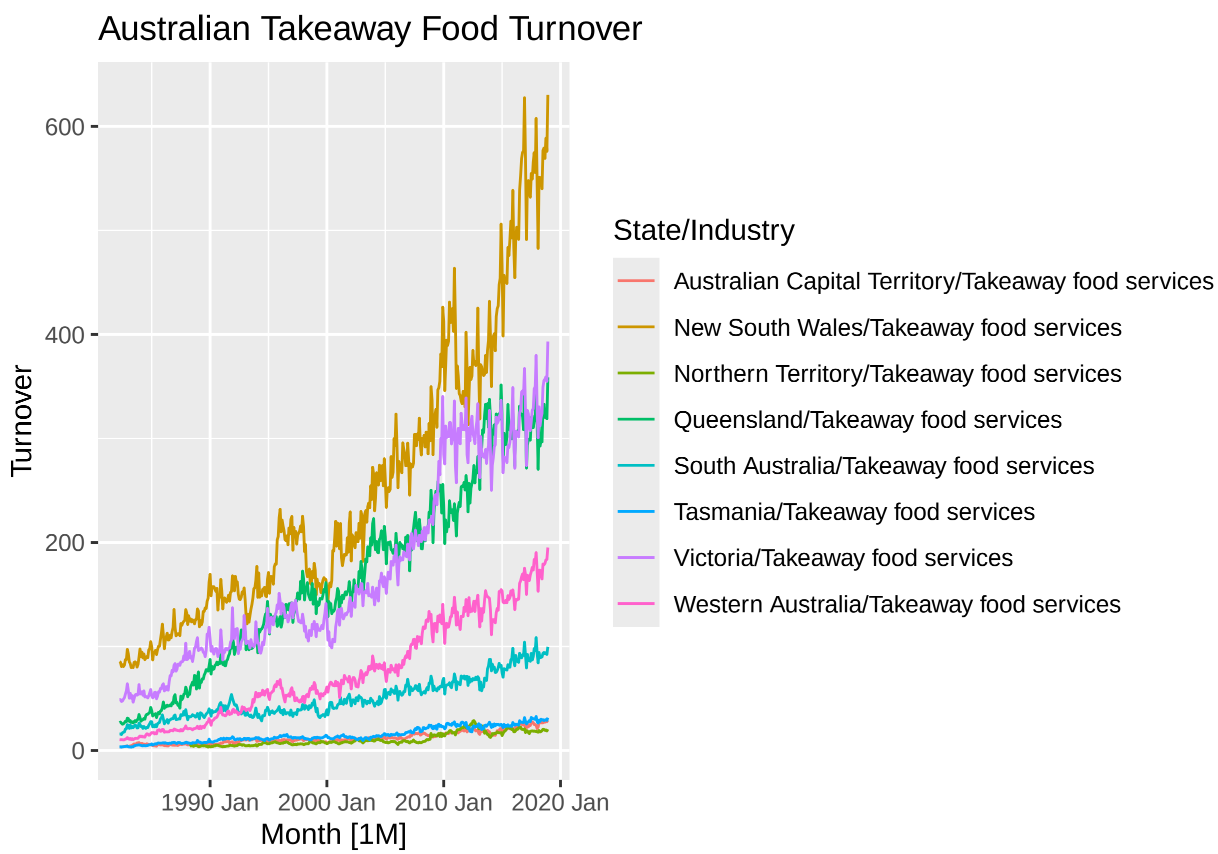

```r
hh_wealth <- hh_budget %>%
  filter(!is.na(Wealth)) %>%
  select(Wealth)

autoplot(hh_wealth) +
  labs(title = "Household Wealth")
```



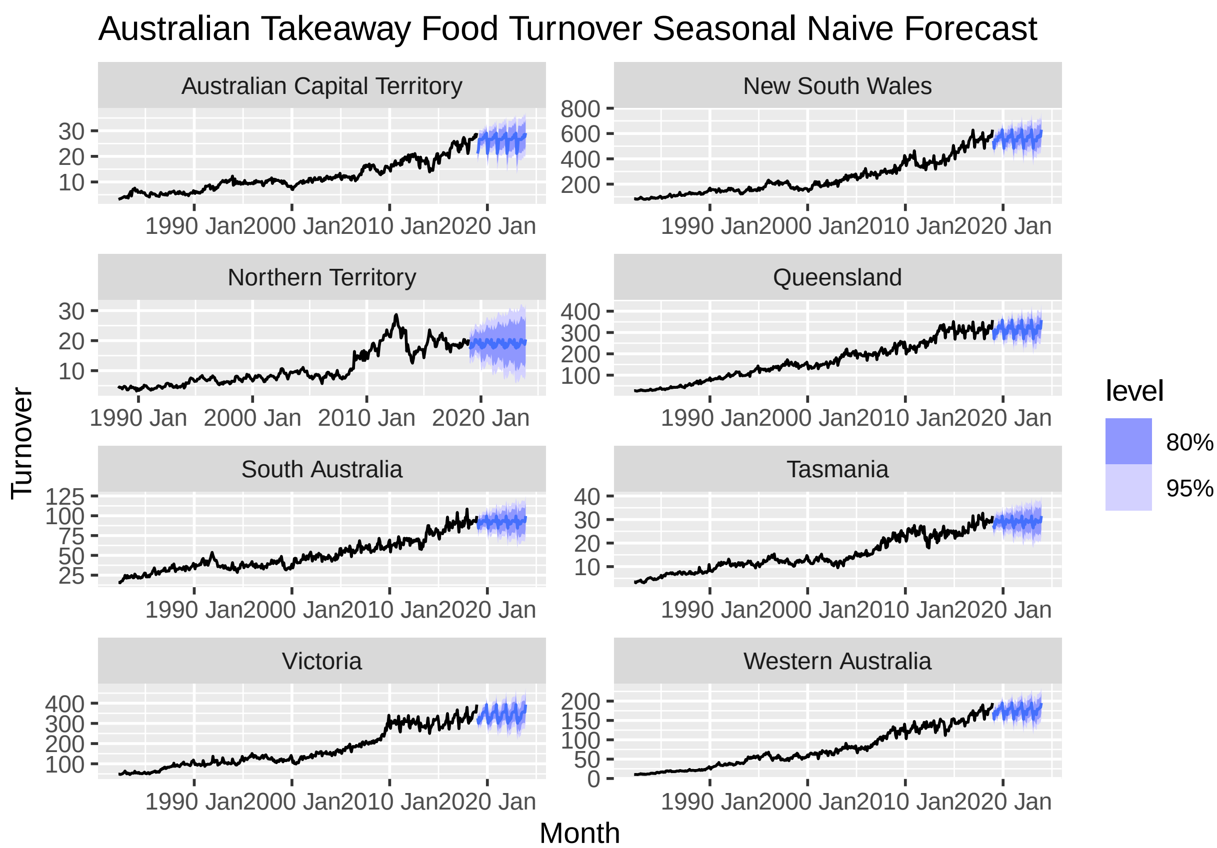

```
hh_wealth %>%
  model(Drift = RW(Wealth ~ drift())) %>%
  forecast(h = "5 years") %>%
  autoplot(hh_wealth) +
  labs(title = "Household Wealth Drift Forecast")
```




```r
takeaway <- aus_retail %>%
  filter(!is.na(Turnover)) %>%
  filter(Industry == "Takeaway food services")

autoplot(takeaway) +
  labs(title = "Australian Takeaway Food Turnover")
```




```
takeaway %>%
  model(Snaive = SNAIVE(Turnover)) %>%
  forecast(h = "5 years") %>%
  autoplot(takeaway) +
  labs(title = "Australian Takeaway Food Turnover Seasonal Naive Forecast") +
  facet_wrap(vars(State), scales = "free", ncol = 2, nrow = 4)
```

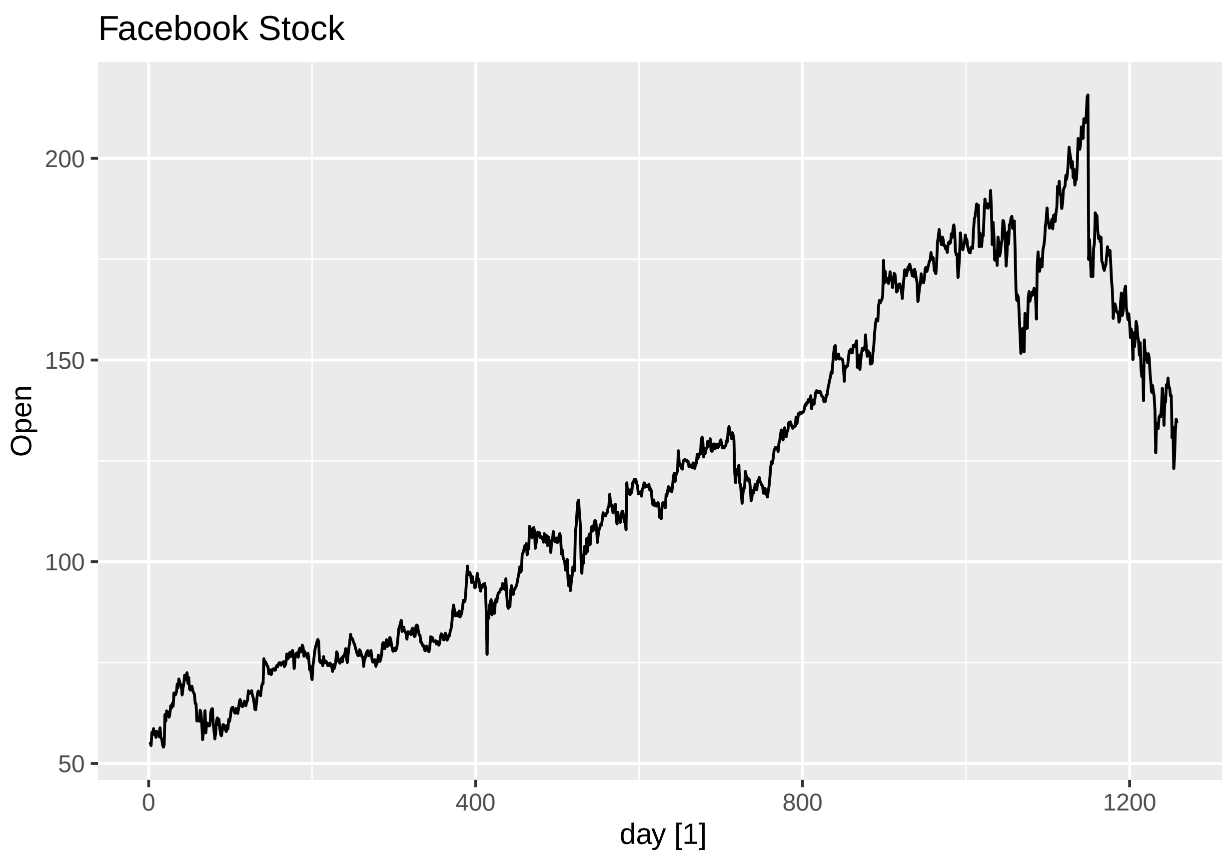



### Exercise 2

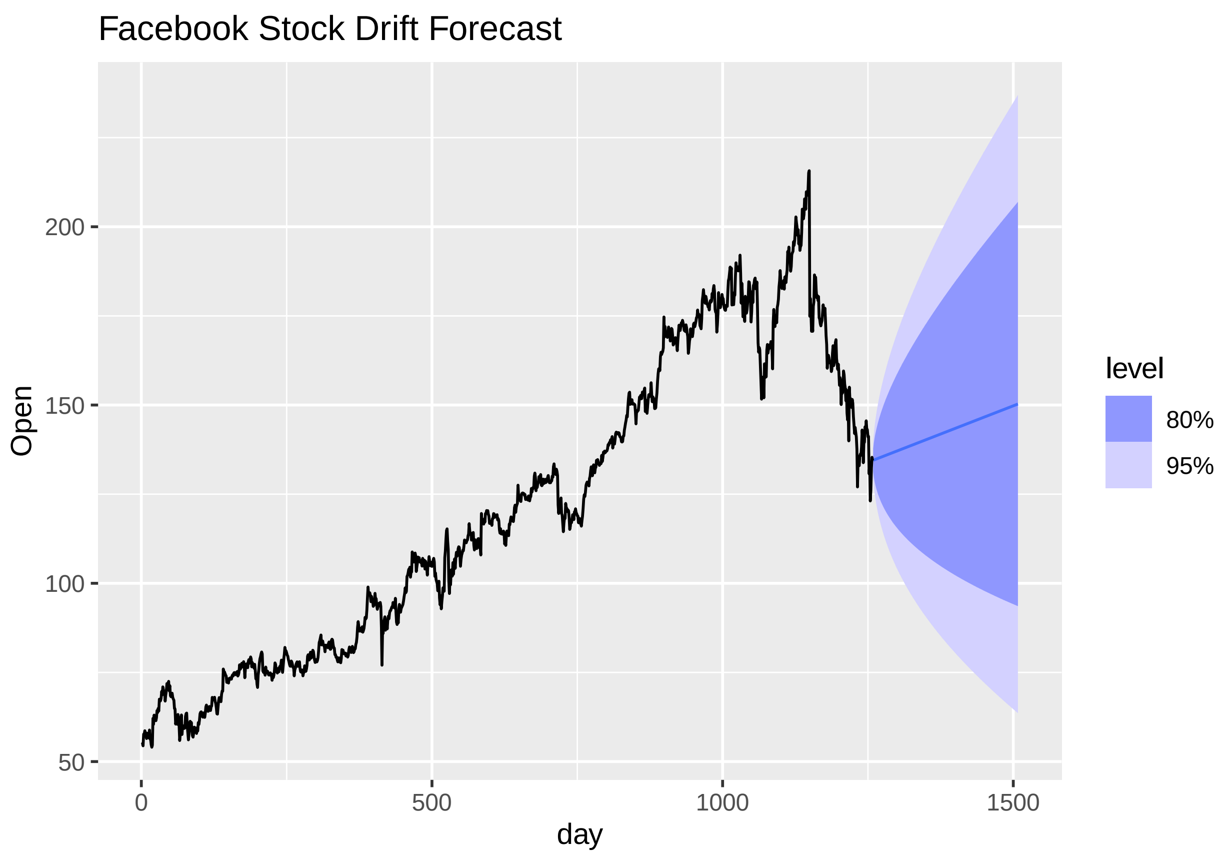
Use the Facebook stock price (data set gafa_stock) to do the following:

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

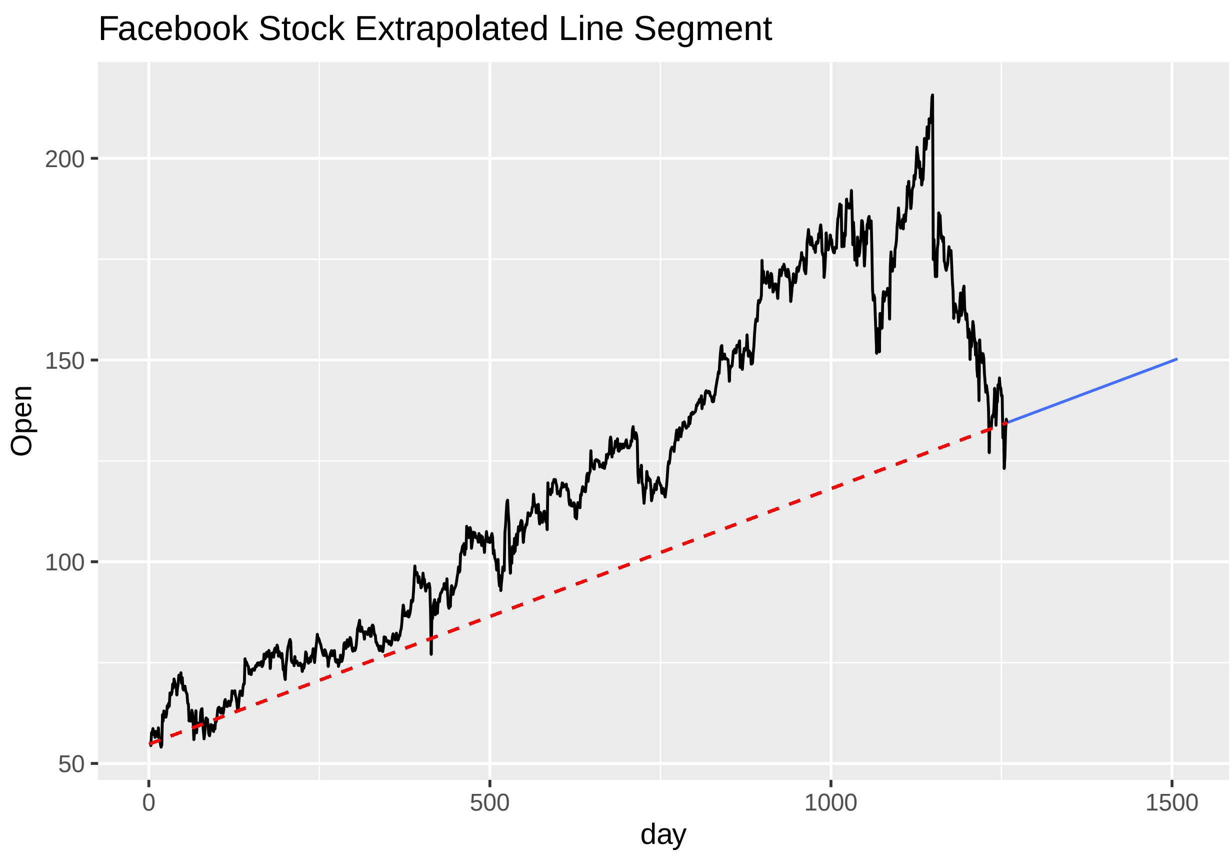
```r
fb_stock <- gafa_stock %>%
  filter(Symbol == "FB") %>%
  mutate(day = row_number()) %>%
  update_tsibble(index = day, regular = TRUE)

autoplot(fb_stock) +
  labs(title = "Facebook Stock")
```



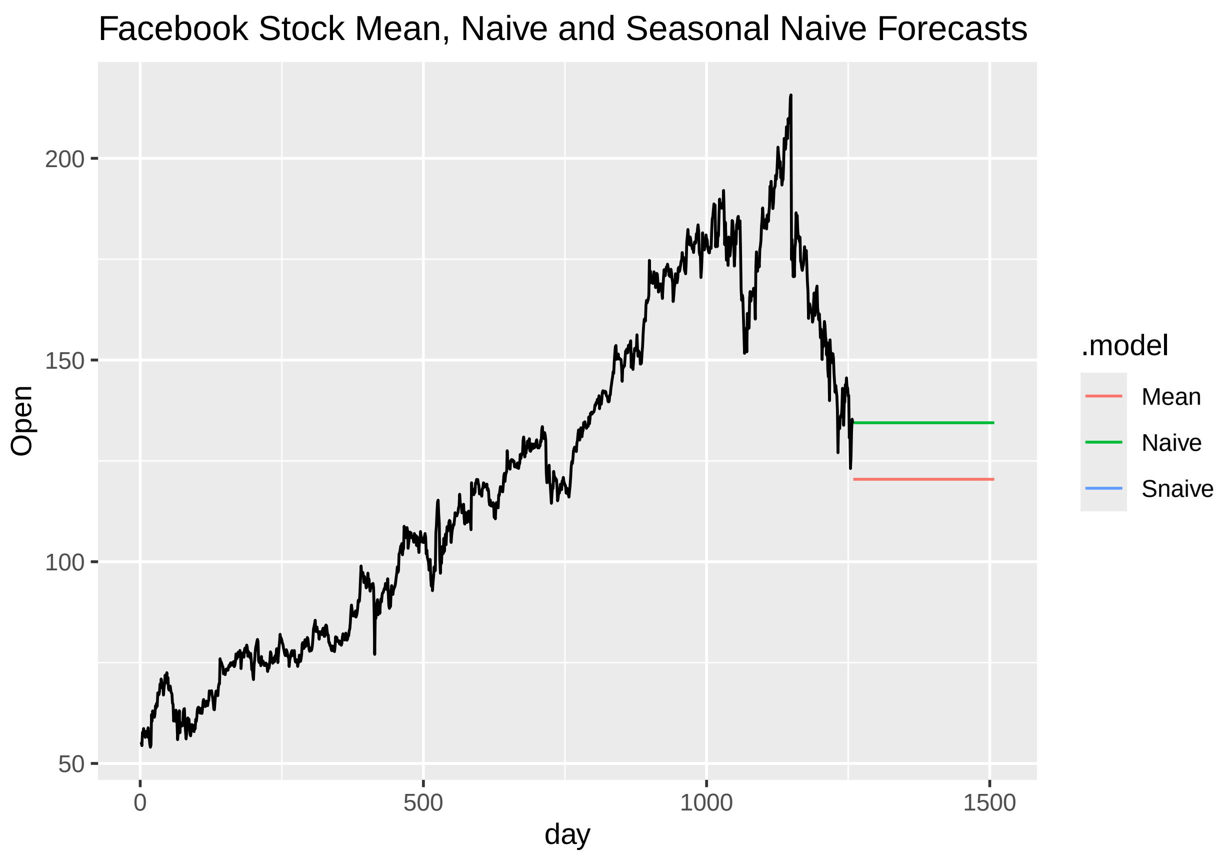

```r
fb_stock %>%
  model(Drift = RW(Open ~ drift())) %>%
  forecast(h = 250) %>%
  autoplot(fb_stock) +
  labs(title = "Facebook Stock Drift Forecast")
```




```
fb_stock %>%
  model(Drift = RW(Open ~ drift())) %>%
  forecast(h = 250) %>%
  autoplot(fb_stock, level = NULL) +
  geom_segment(aes(x = min(day), y = Open[which.min(day)],
                   xend = max(day), yend = Open[which.max(day)]),
               color = "red", linetype = "dashed") +
  labs(title = "Facebook Stock Extrapolated Line Segment")
```



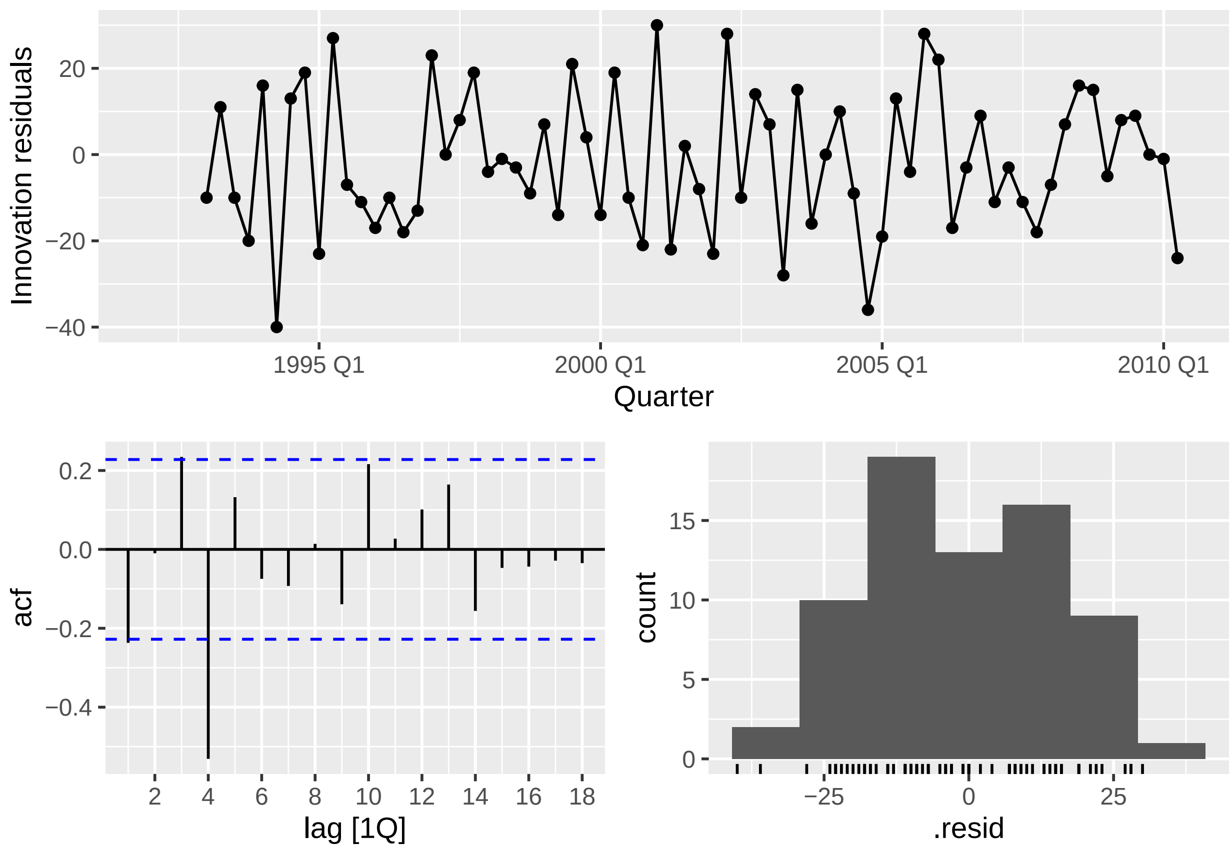

```r
fb_stock %>%
  model(Mean = MEAN(Open),
        Naive = NAIVE(Open),
        Snaive = SNAIVE(Open)) %>%
  forecast(h = 250) %>%
  autoplot(fb_stock, level = NULL) +
  labs(title = "Facebook Stock Mean, Naive and Seasonal Naive Forecasts")
```



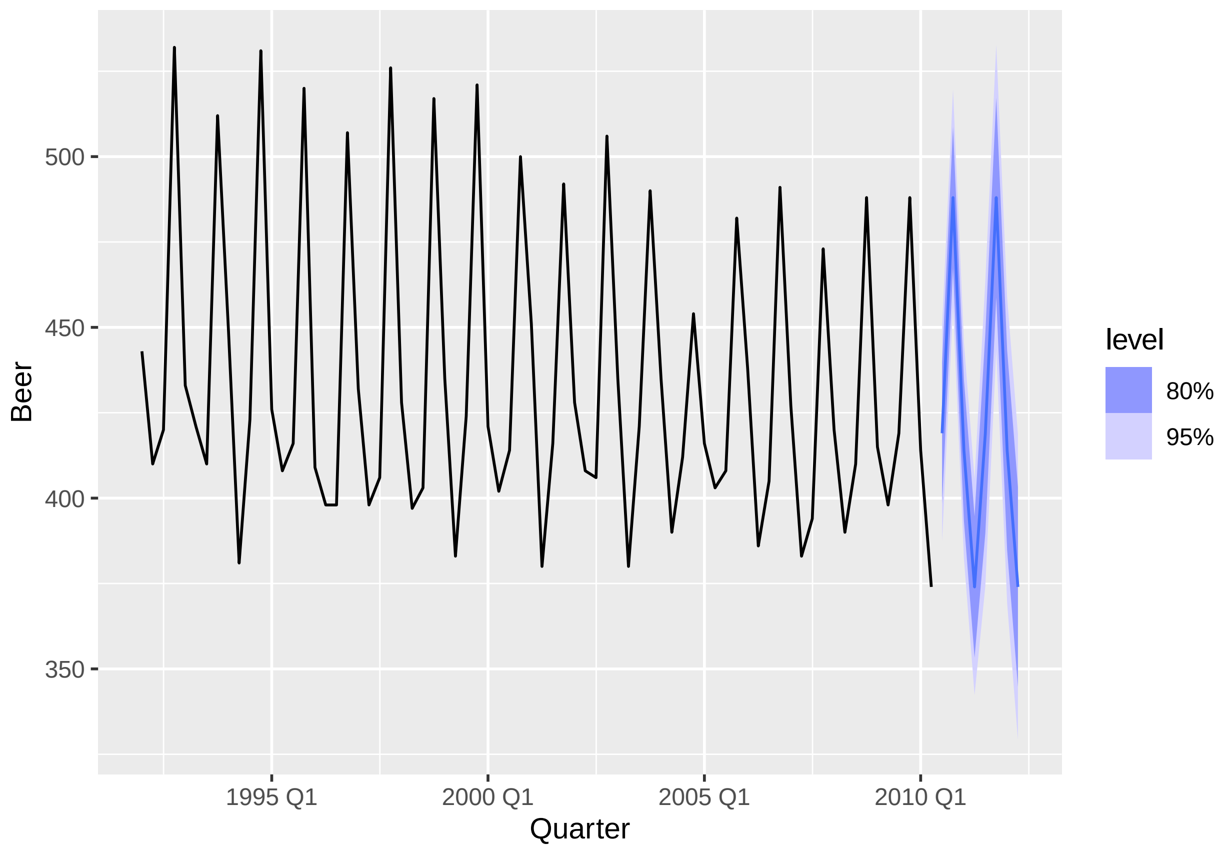

### Exercise 3

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)
# Define and estimate a model
fit <- recent_production |> model(SNAIVE(Beer))
# Look at the residuals
fit |> gg_tsresiduals()
```




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

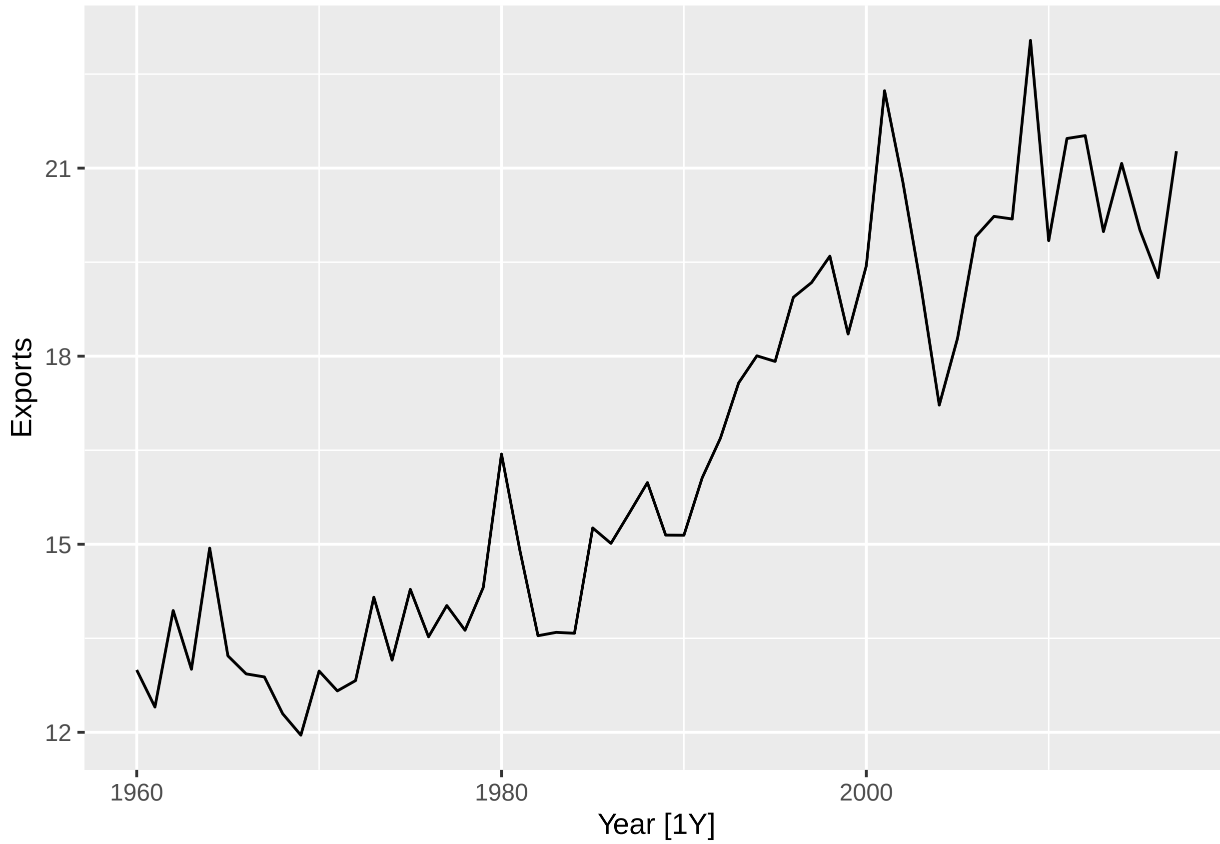



What do you conclude?

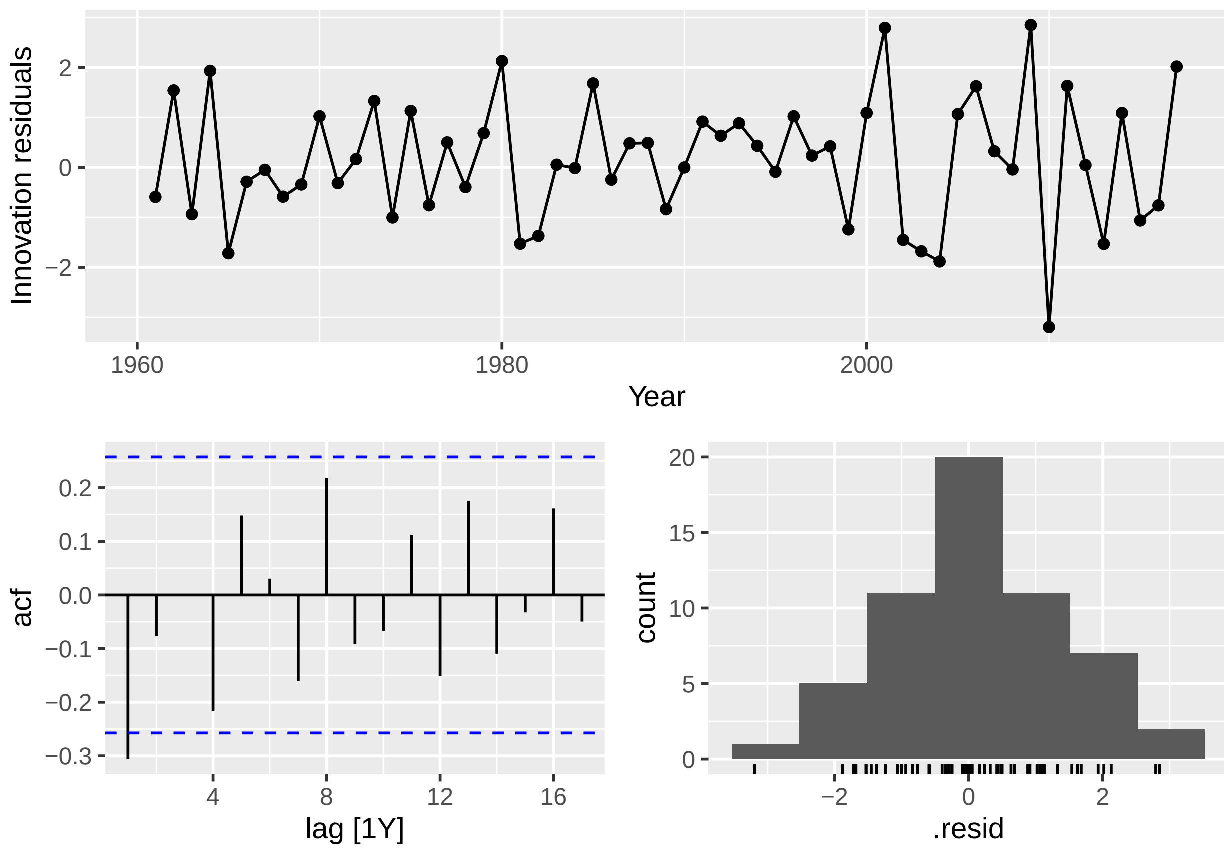
## Exercise 4

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.

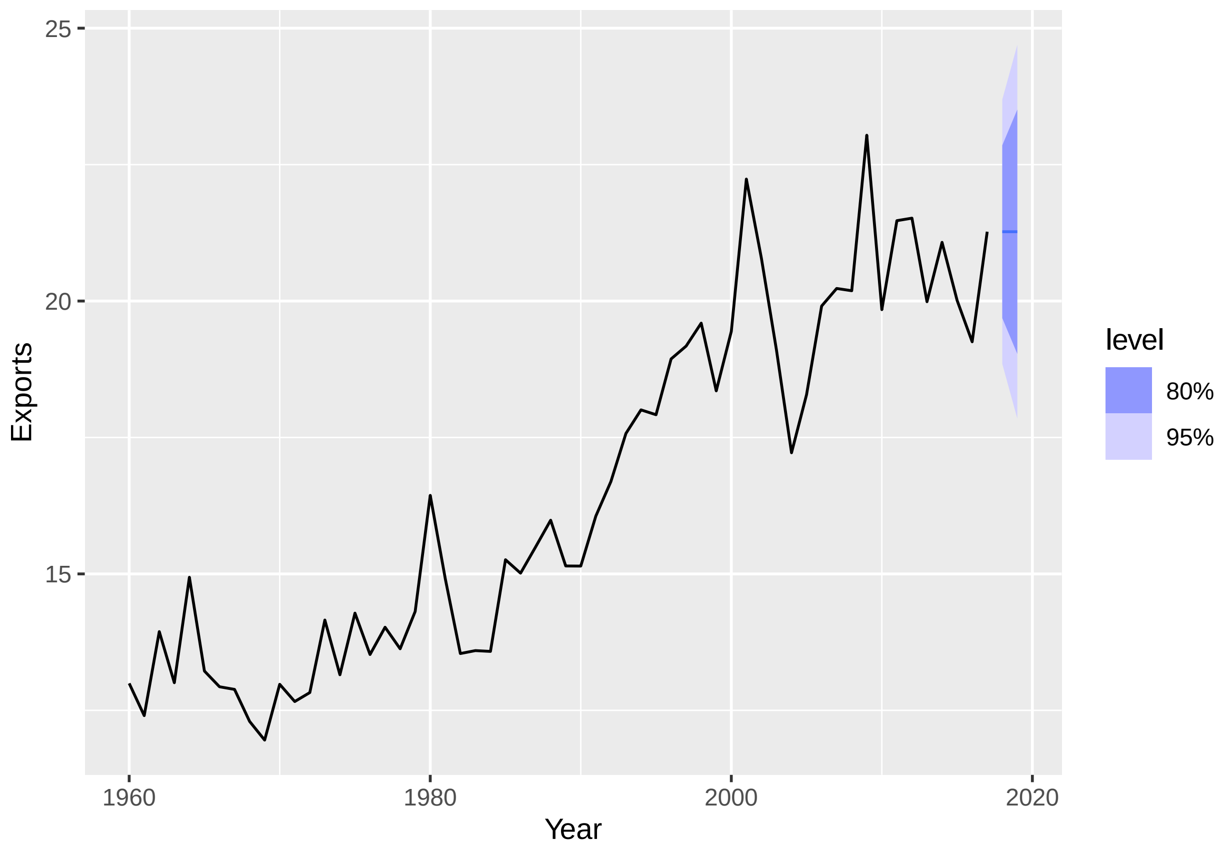
```
aus_exports <- global_economy %>%
  filter(!is.na(Exports)) %>%
  filter(Country == "Australia") %>%
  select(Exports)
autoplot(aus_exports)
```



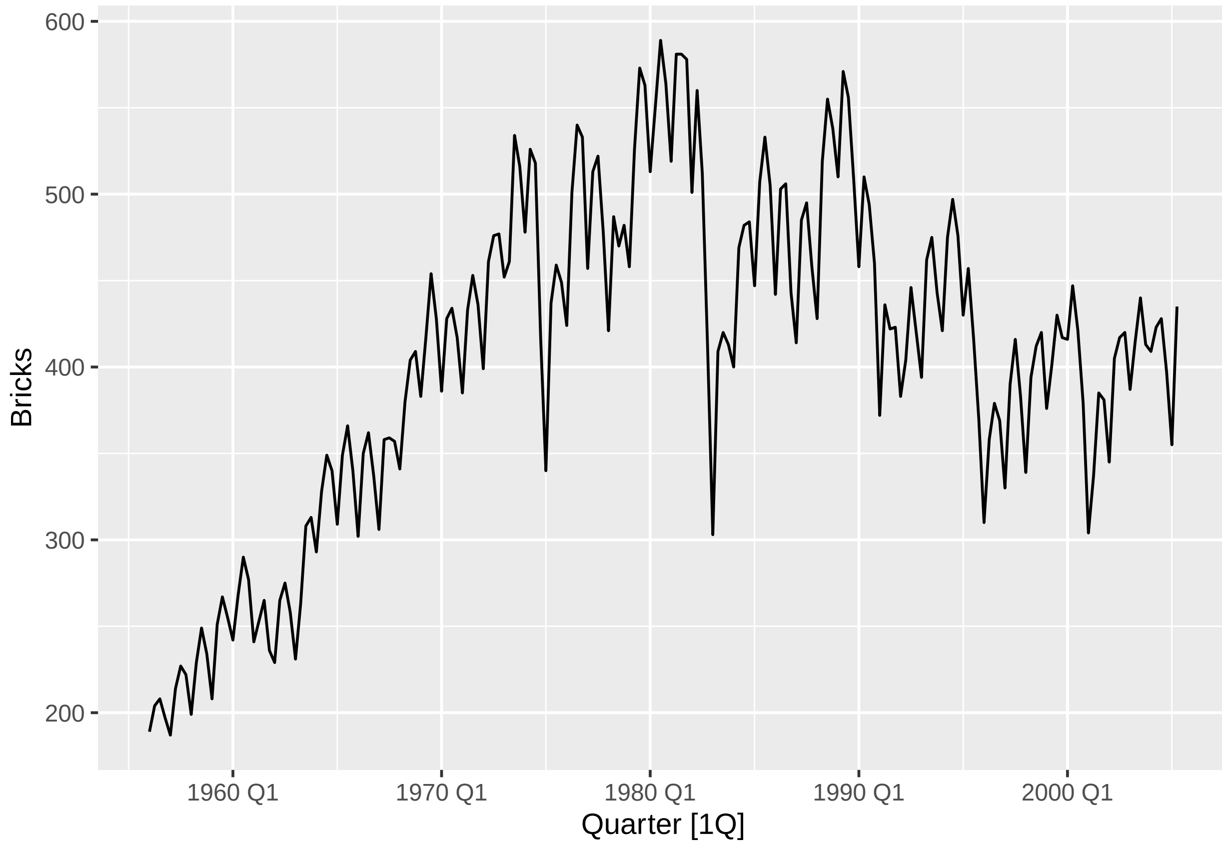

```r
afit <- aus_exports %>%
  model(NAIVE(Exports))

afit %>%
  gg_tsresiduals()
```



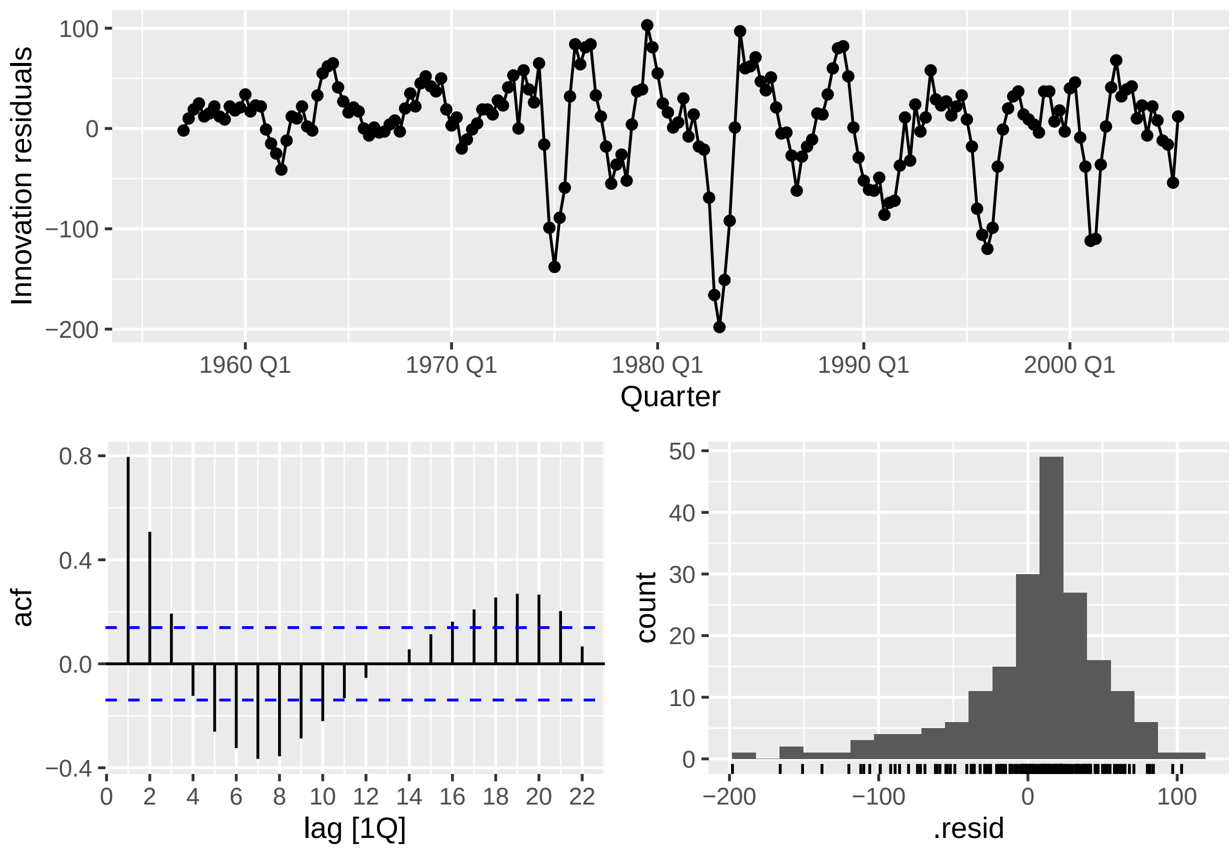

```r
afit %>%
  forecast() %>%
  autoplot(aus_exports)
```



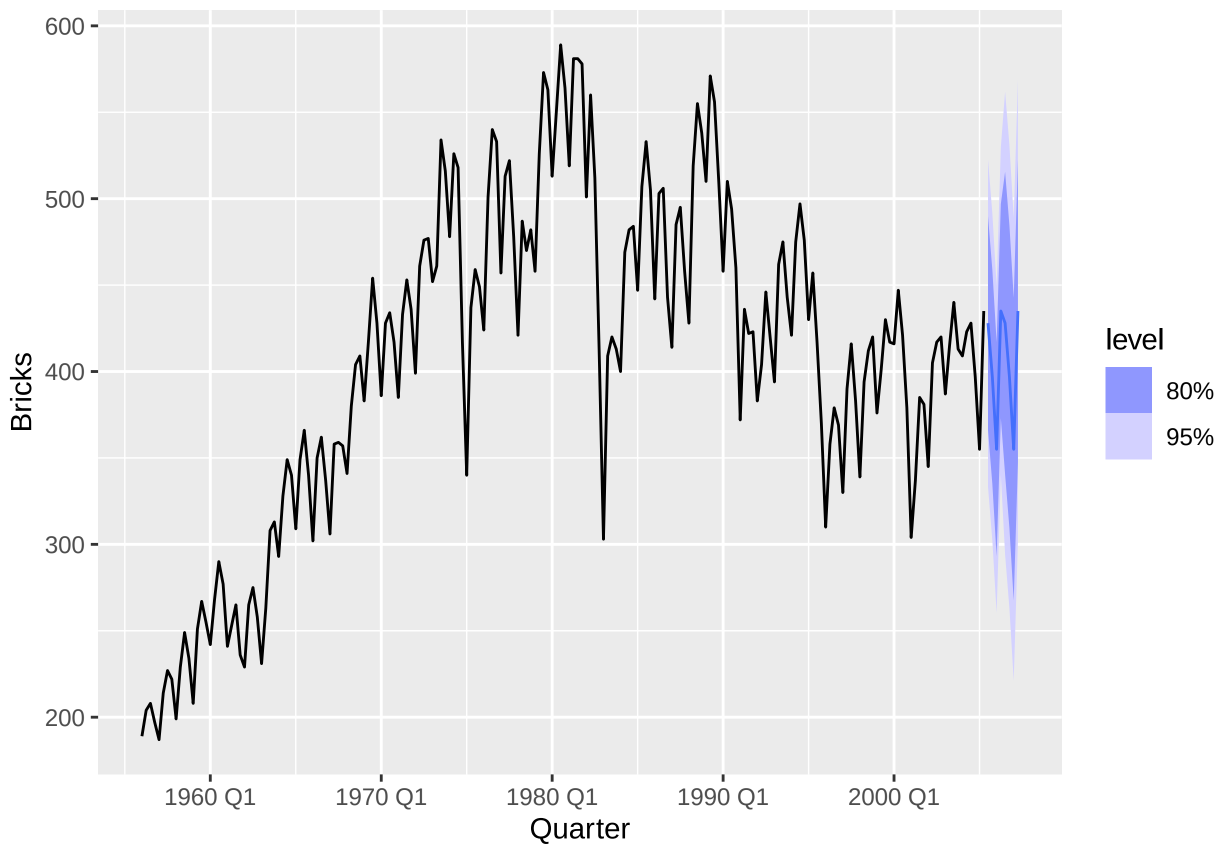

```
bricks <- aus_production %>%
  filter(!is.na(Bricks)) %>%
  select(Bricks)
autoplot(bricks)
```




```
bfit <- bricks %>%
  model(SNAIVE(Bricks))

bfit %>%
  gg_tsresiduals()
```




```r
bfit %>%
  forecast() %>%
  autoplot(bricks)
```




### Exercise 7

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

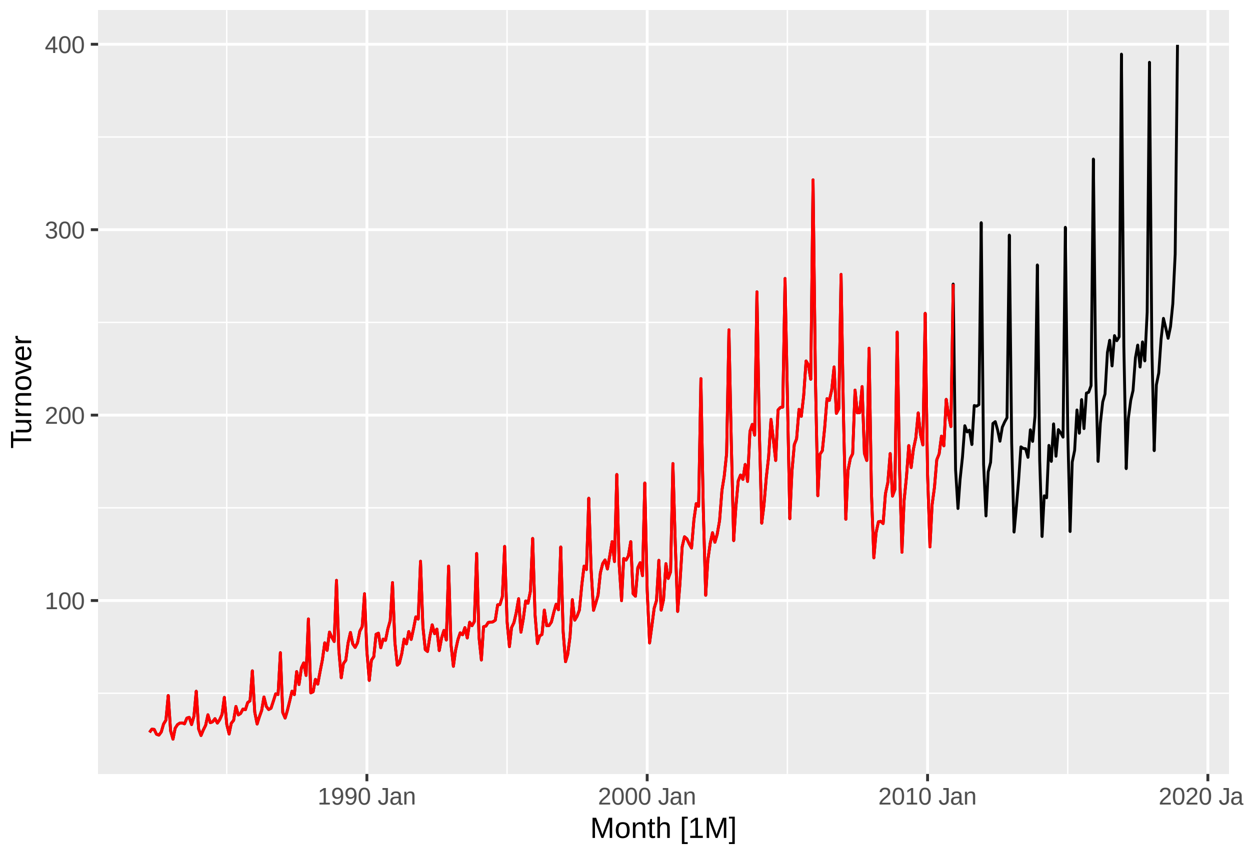
```
set.seed(1)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

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.

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

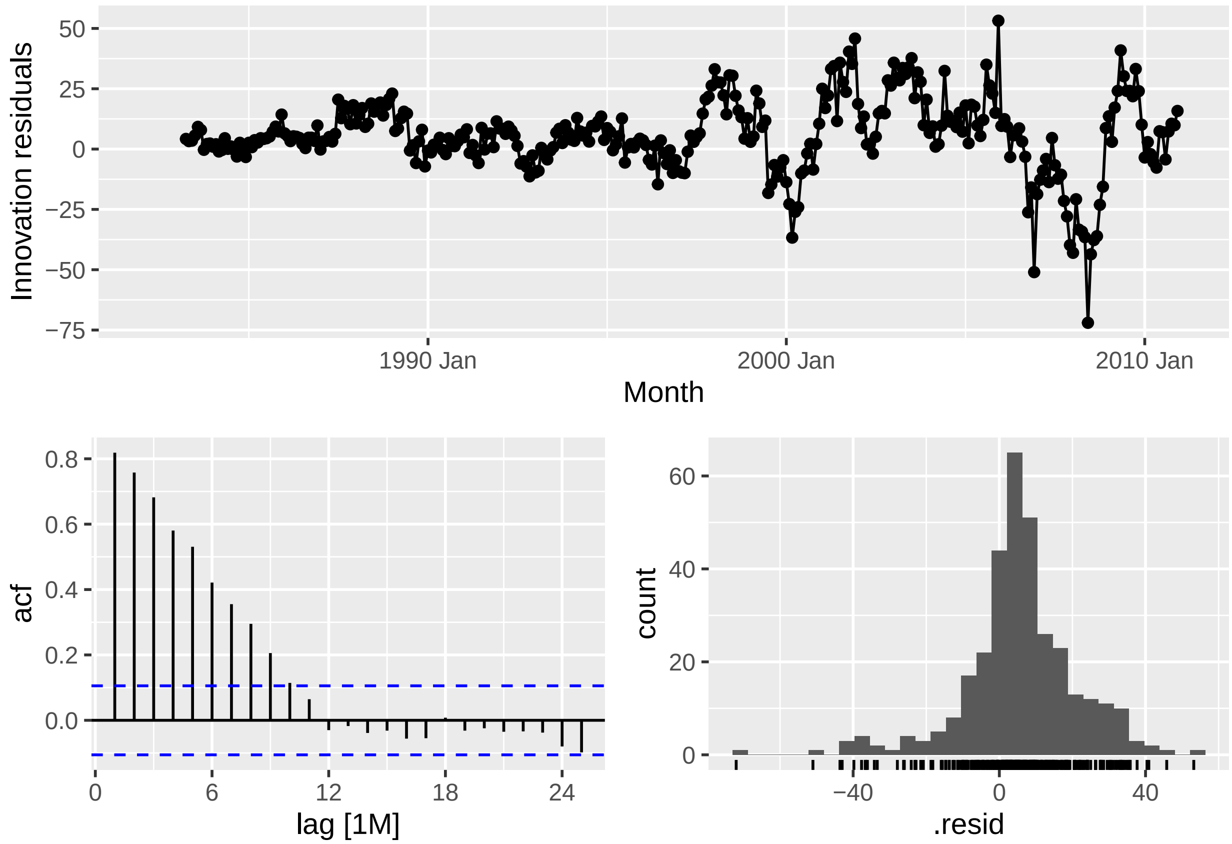



Fit a seasonal naïve model using SNAIVE() applied to your training data (myseries_train).

```
mfit <- myseries_train %>%
  model(SNAIVE(Turnover))
```

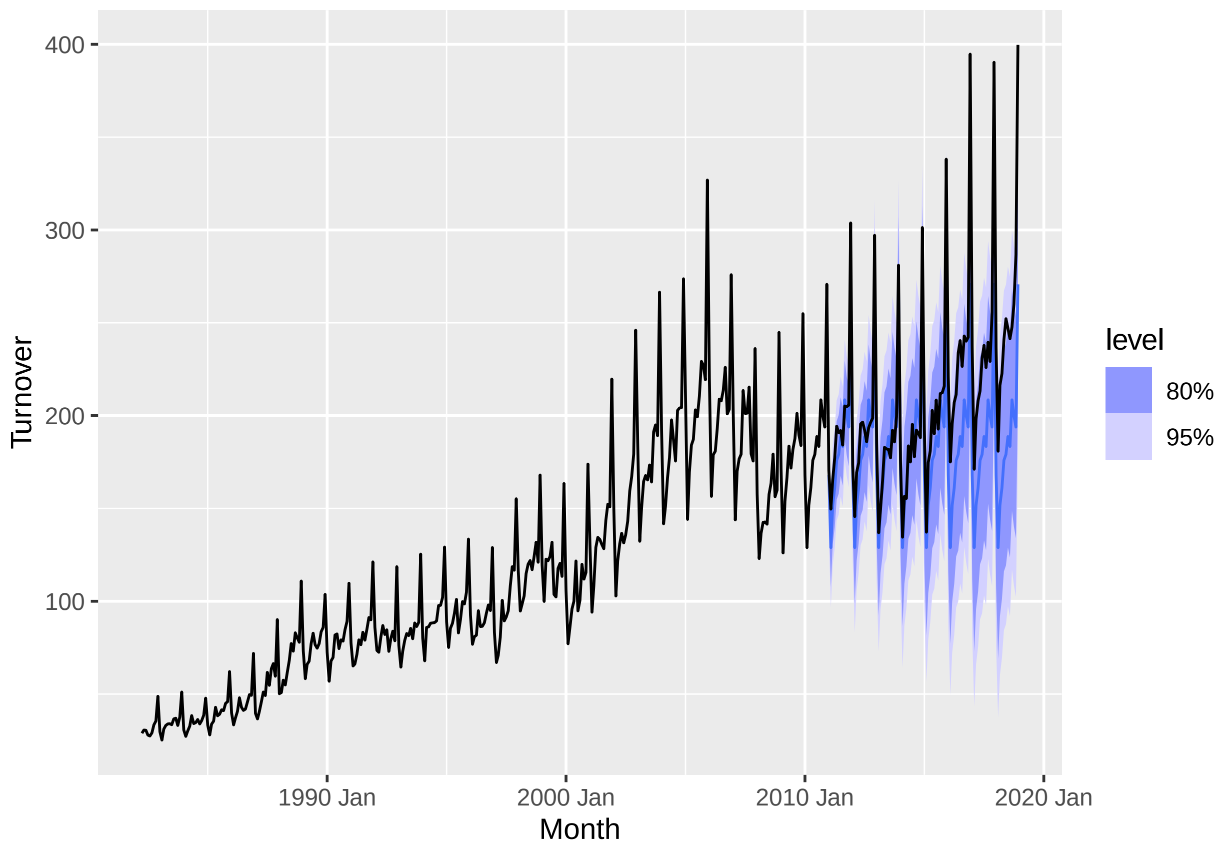
Check the residuals. Do the residuals appear to be uncorrelated and normally distributed?

```
mfit %>%
  gg_tsresiduals()
```




Produce forecasts for the test data

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




Compare the accuracy of your forecasts against the actual values.

```
mfit %>%
  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 Queensl~ Clothin~ SNAIV~ Trai~  5.50  16.5  12.1  5.39  10.6     1     1 0.819
```

```
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~ Quee~ Clothin~ Test   26.9  40.2  29.3  11.3  12.6  2.43  2.44 0.752
```

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