Data 624 Homework 3

Steven Gonzalez

2/23/2025

Load Packages

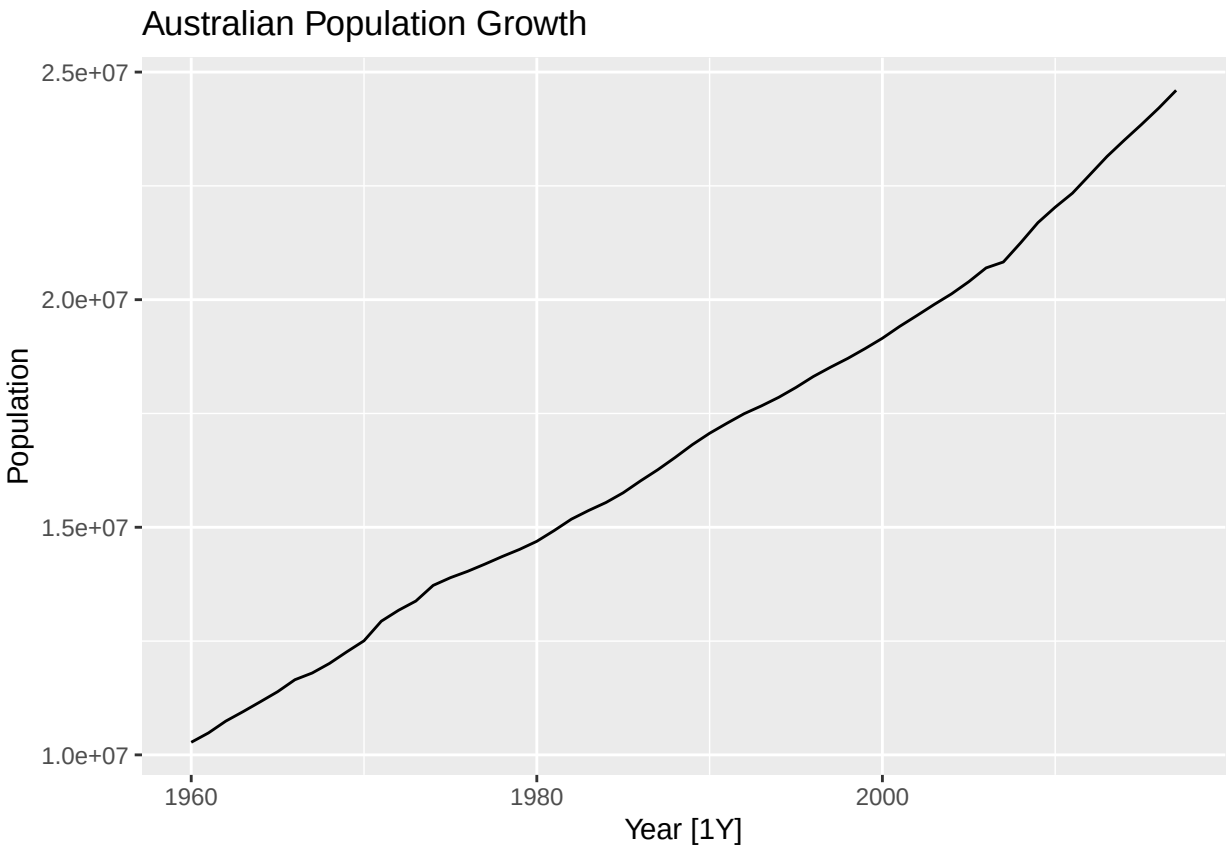
```
library(fpp3)
library(seasonal)
library(USgas)
```

Exercise 1

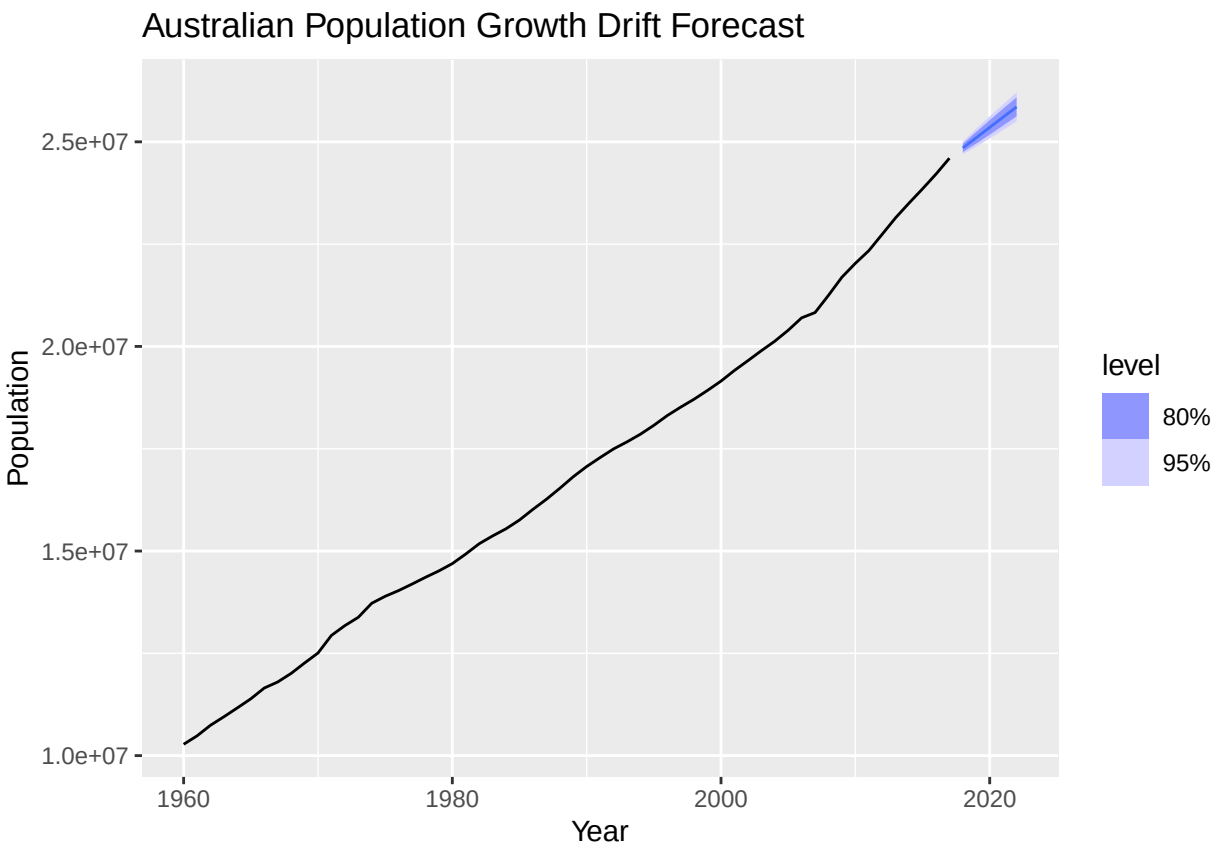
Produce forecasts for the following series using whichever of NAIVE(y), SNAIVE(y) or RW(y ~ drift()) is more appropriate in each case:

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")
```

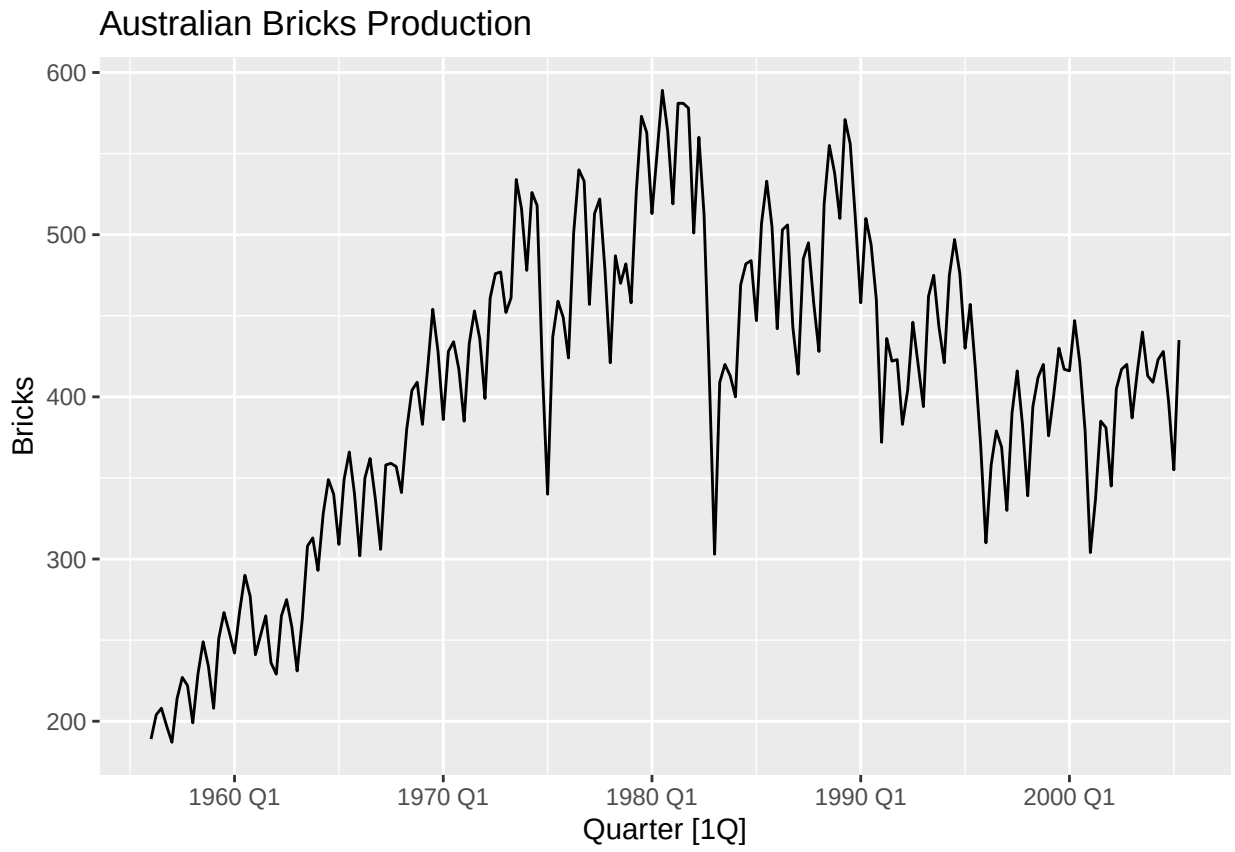



```
aus_pop %>%
  model(Drift = RW(Population ~ drift())) %>%
  forecast(h = "5 years") %>%
  autoplot(aus_pop) +
  labs(title = "Australian Population Growth Drift Forecast")
```

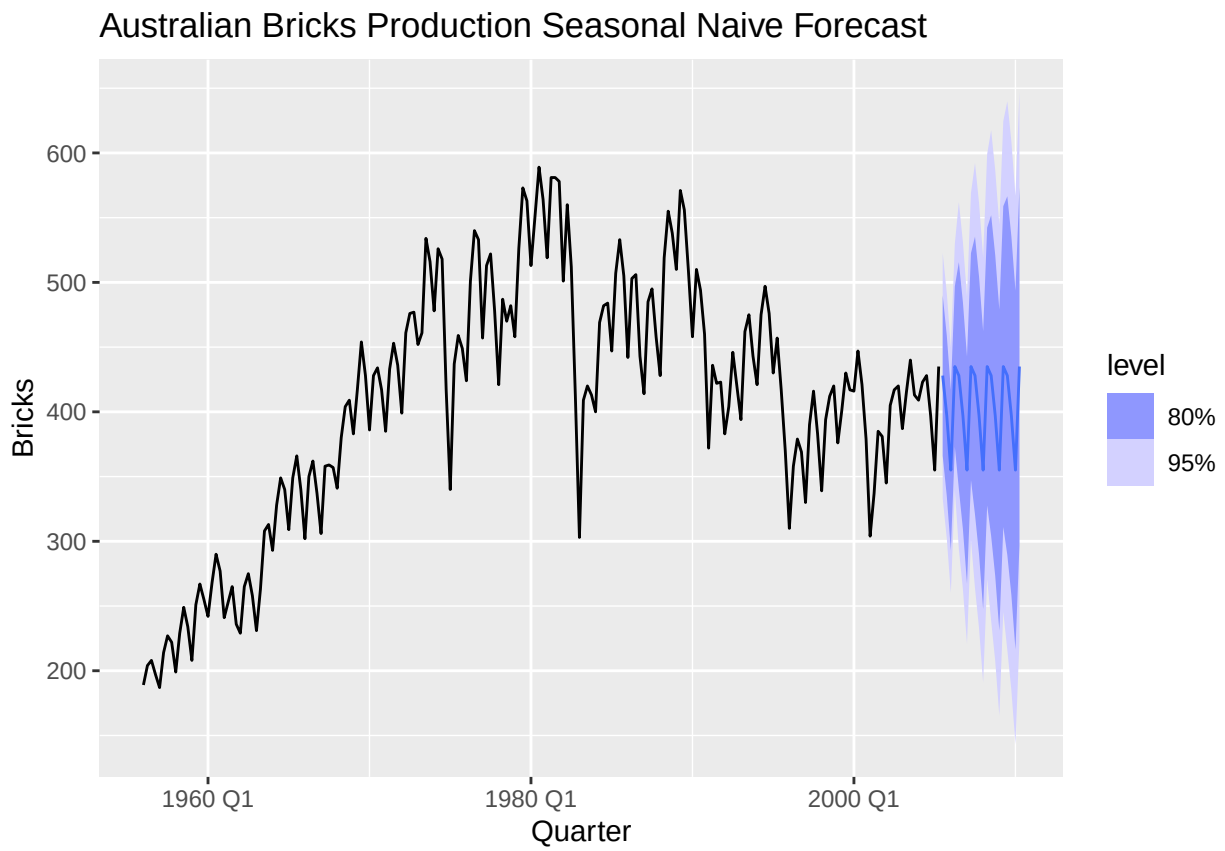


The above forecast for Australian Population from `global_economy` was created using `RW(y ~ drift())`. The reason for using this technique over `NAIVE(y)` and `SNAIVE(y)` was due to the series' already existing, and consistent, positive trend and lack of seasonality rendering other techniques useless.

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

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

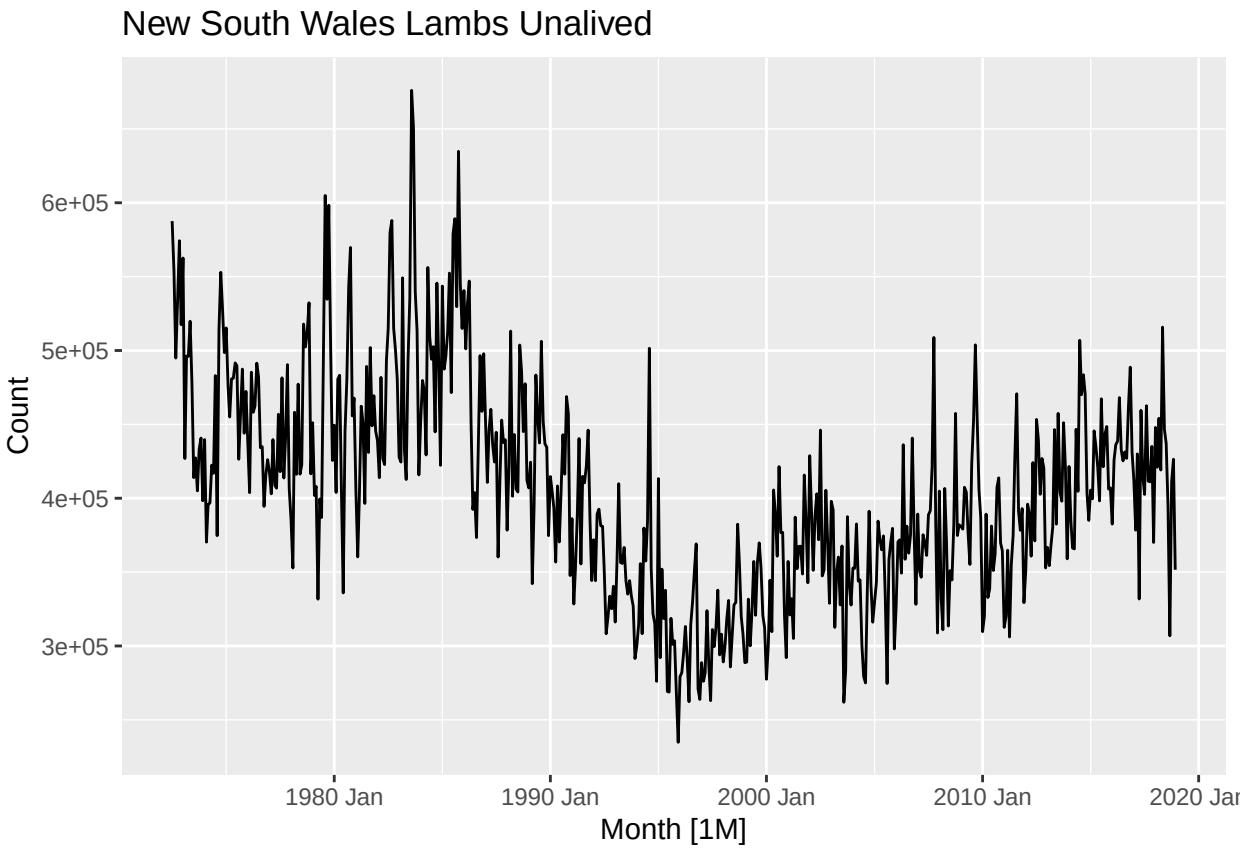



```
bricks_pro %>%
  model(Snaive = SNAIVE(Bricks)) %>%
  forecast(h = "5 years") %>%
  autoplot(bricks_pro) +
  labs(title = "Australian Bricks Production Seasonal Naive Forecast")
```

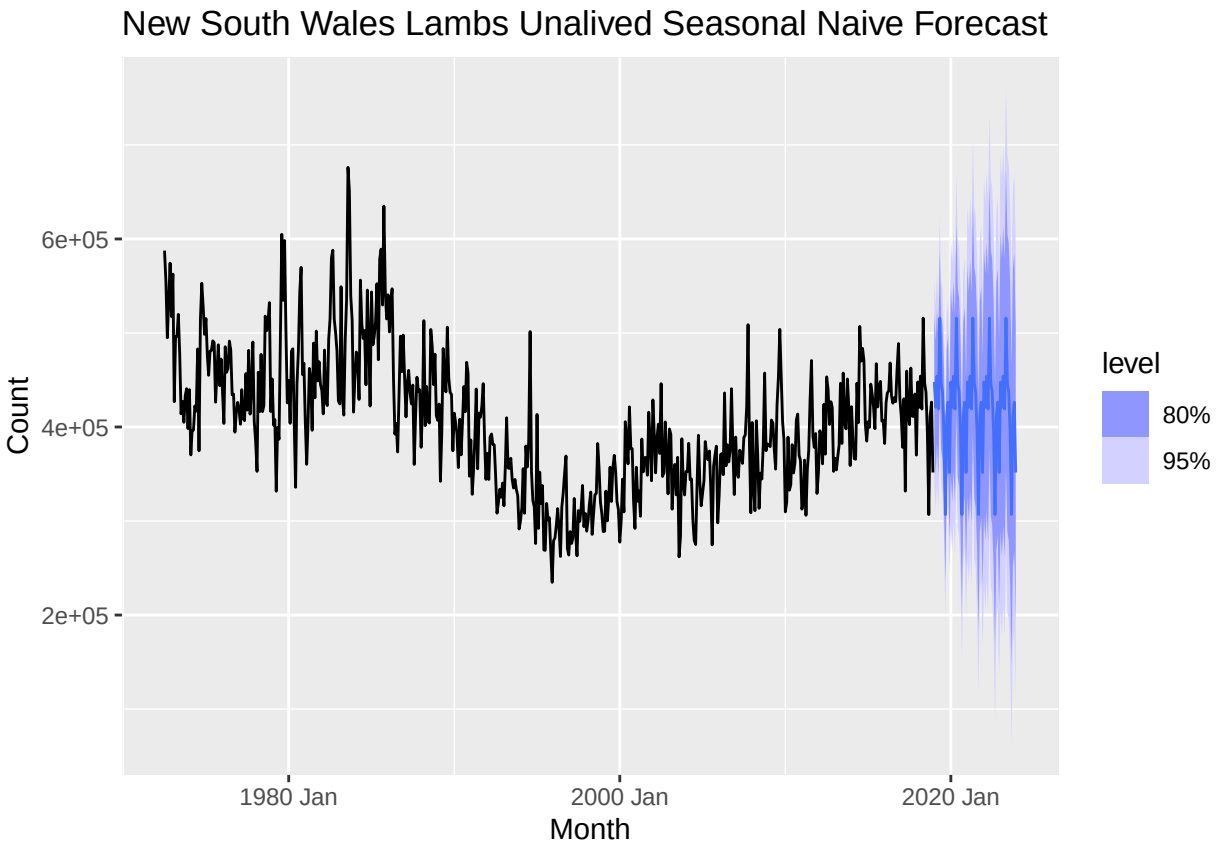


The above forecast for Bricks from `aus_production` was created using `SNAIVE(y)`. Highly repetitive and seasonal data such as the one found in this series is best predicted with seasonality in mind. In addition, the confidence interval seen in the graph can also account for any upward or downward fluctuation that can take place in series' trend.

```
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")
```

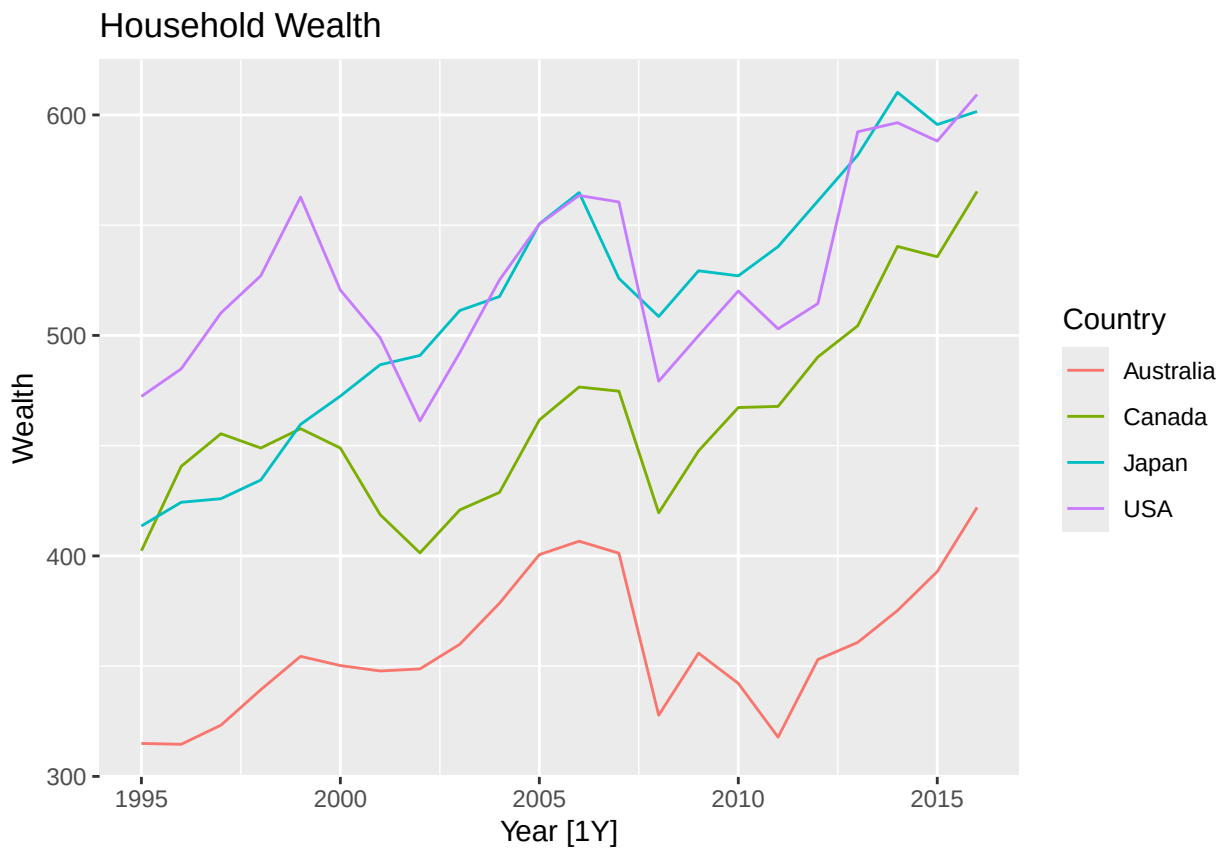



```
nsw_lambs %>%
  model(Snaive = SNAIVE(Count)) %>%
  forecast(h = "5 years") %>%
  autoplot(nsw_lambs) +
  labs(title = "New South Wales Lambs Unalived Seasonal Naive Forecast")
```

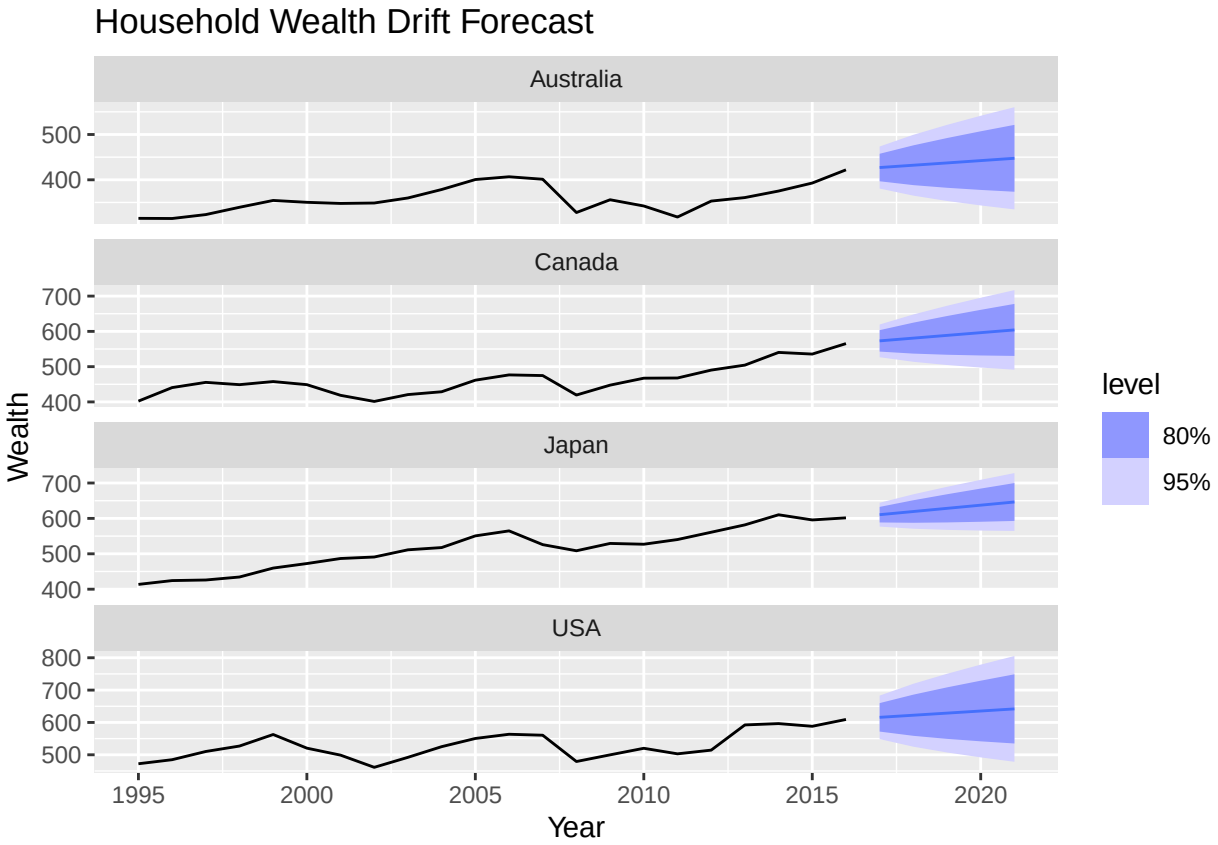


The above forecast for NSW Lambs from `aus_livestock` was created using `SNAIVE(y)`. Just like the series before this data is best predicted with seasonality in mind. The confidence interval can also account for any upward or downward fluctuation just like before.

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

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

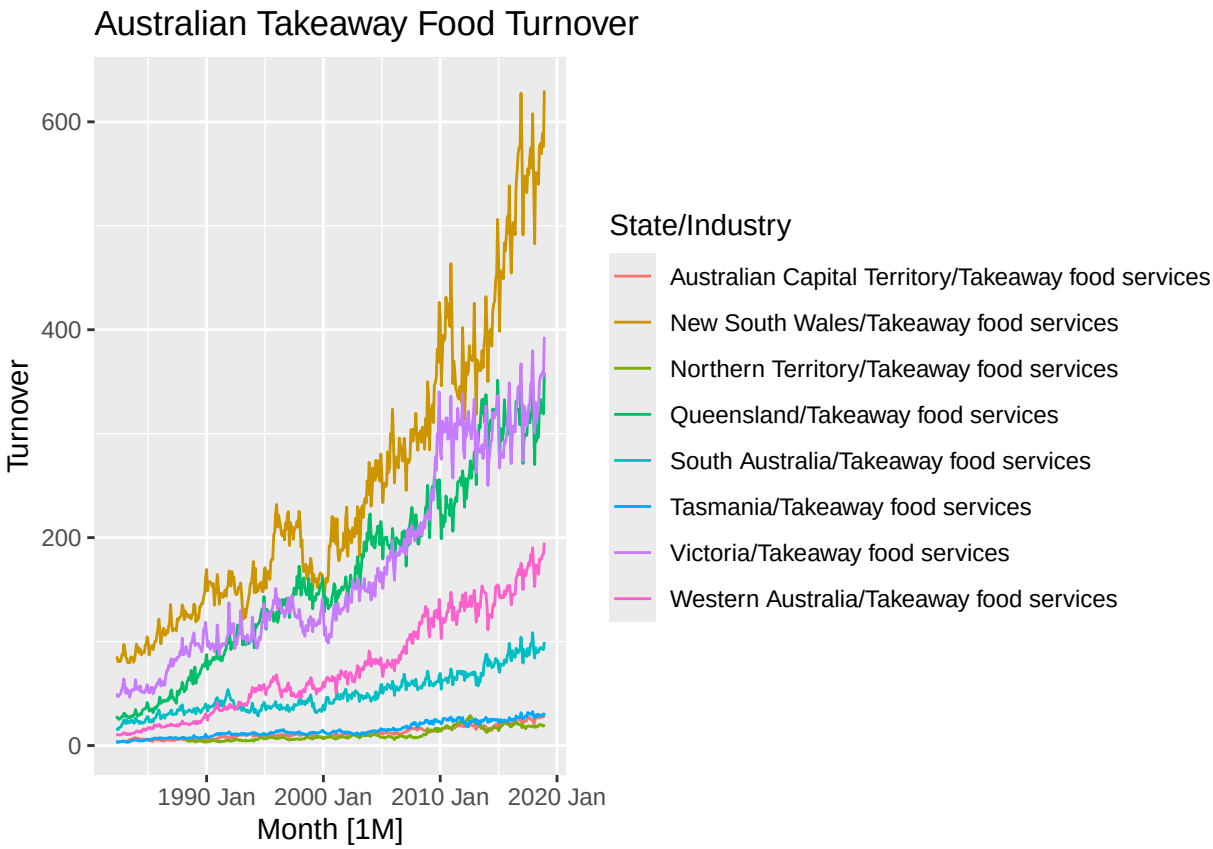



```
hh_wealth %>%
  model(Drift = RW(Wealth ~ drift())) %>%
  forecast(h = "5 years") %>%
  autoplot(hh_wealth) +
  labs(title = "Household Wealth Drift Forecast")
```

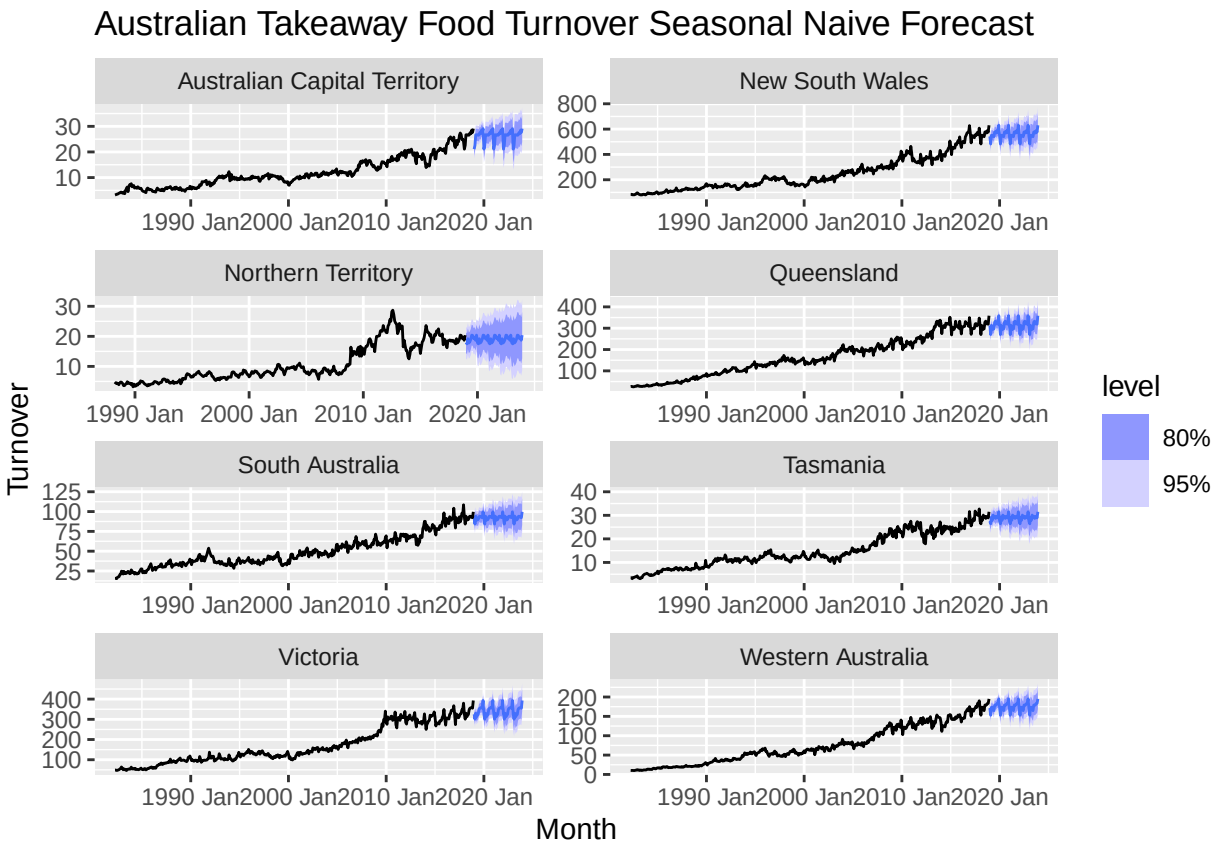


The above forecast for Household wealth from `hh_budget` was created using `RW(y ~ drift())`. The series shows little to no seasonality and each country has a clear positive trend regardless of some fluctuation between the years 1999-2002 and 2007-2008.

```
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)
```



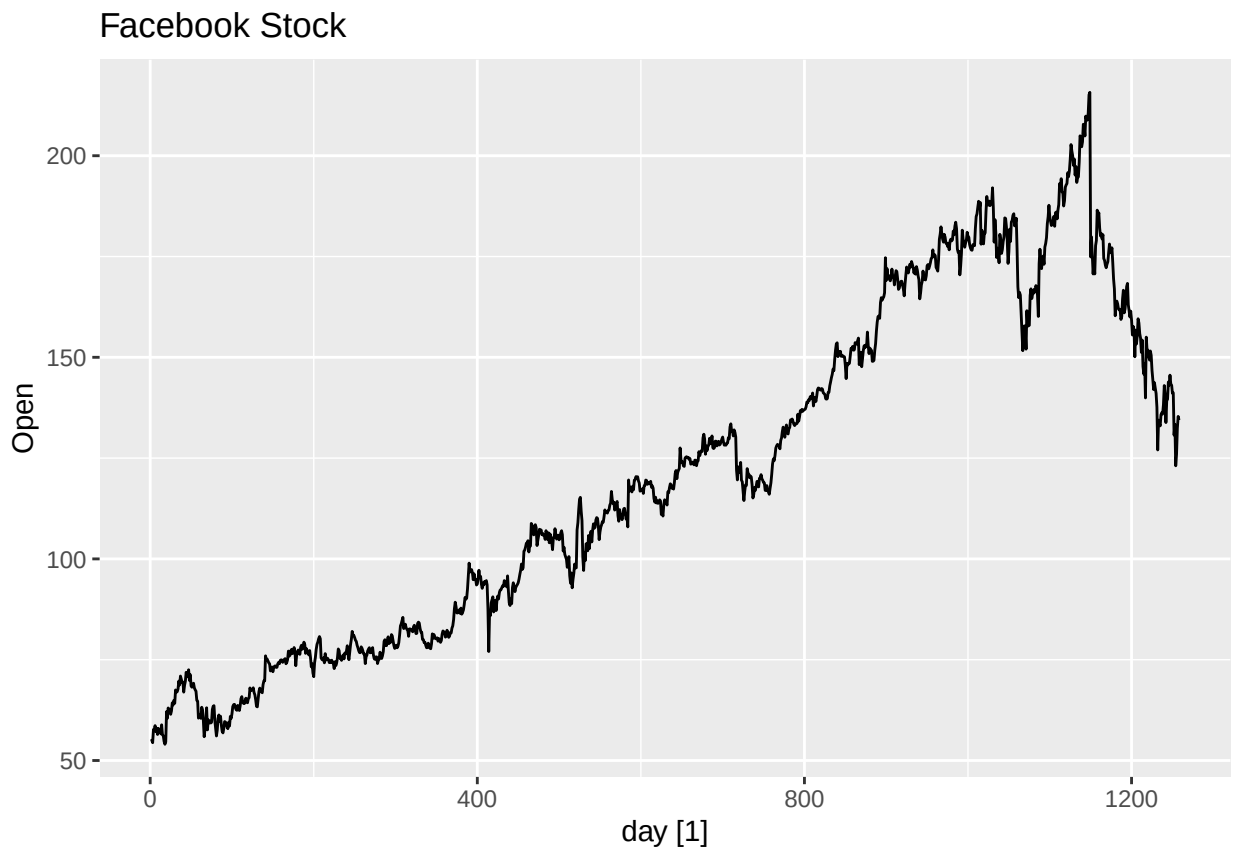
The above forecast for Australian takeaway food turnover from `aus_retail` was created using `SNAIVE(y)`. The turnover data for each region of Australia shows clear seasonality even with a slightly upward trend this can easily be accounted for through the confidence intervals for each respective graph.

Exercise 2

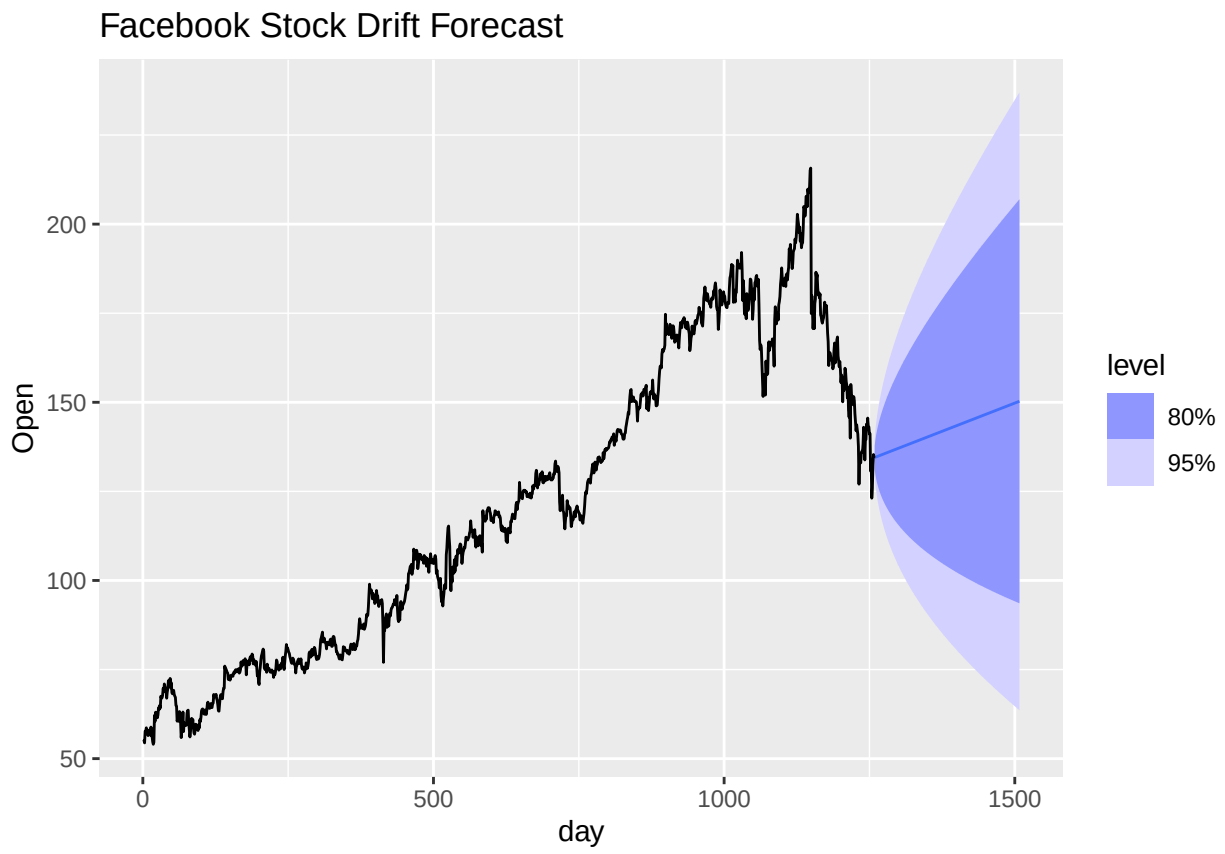
Use the Facebook stock price (data set gafa_stock) to do the following:

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?

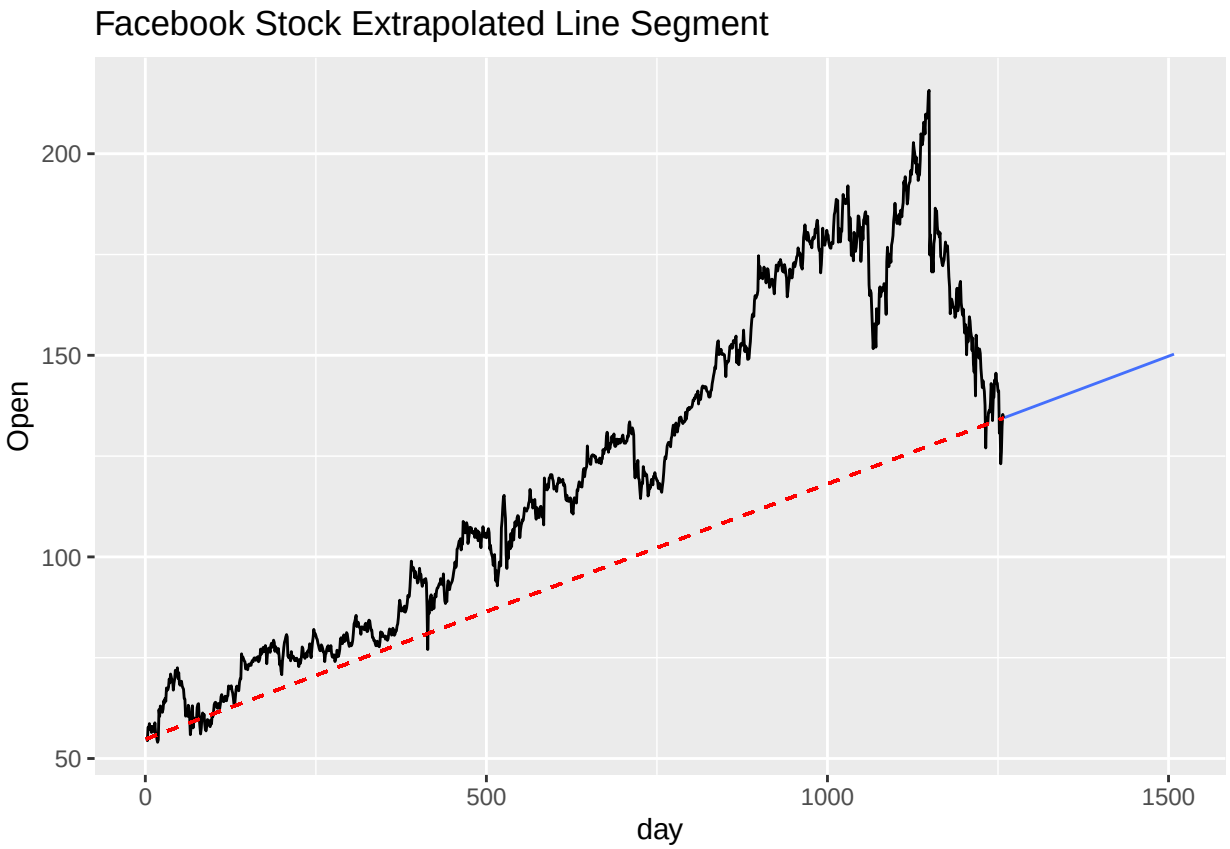
```
fb_stock <- gafa_stock %>%
  filter(Symbol == "FB") %>%
  mutate(day = row_number()) %>%
  update_tsibble(index = day, regular = TRUE)

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




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

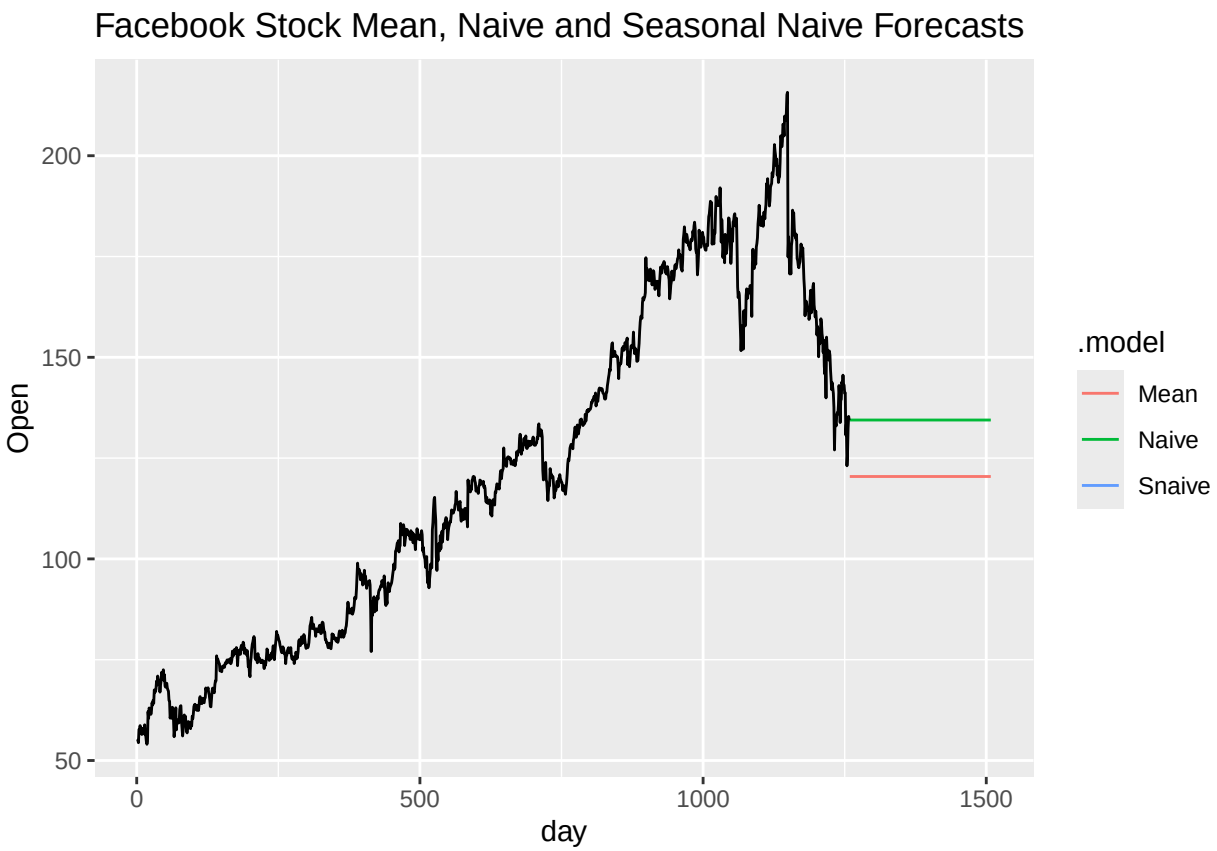



```
fb_stock %>%
  model(Drift = RW(Open ~ drift())) %>%
  forecast(h = 250) %>%
  autoplot(fb_stock, level = NULL) +
  geom_segment(aes(x = min(day), y = Open[which.min(day)],
                   xend = max(day), yend = Open[which.max(day)]),
               color = "red", linetype = "dashed") +
  labs(title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



In order to show that the drift forecast is identical to extending the line drawn between the first and last observation, a second plot was created using `geom_segment` that drew a visible dashed line between the first and last points of the graph. As can be seen, the drift forecast is a clear continuation of this line.

```
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")
```

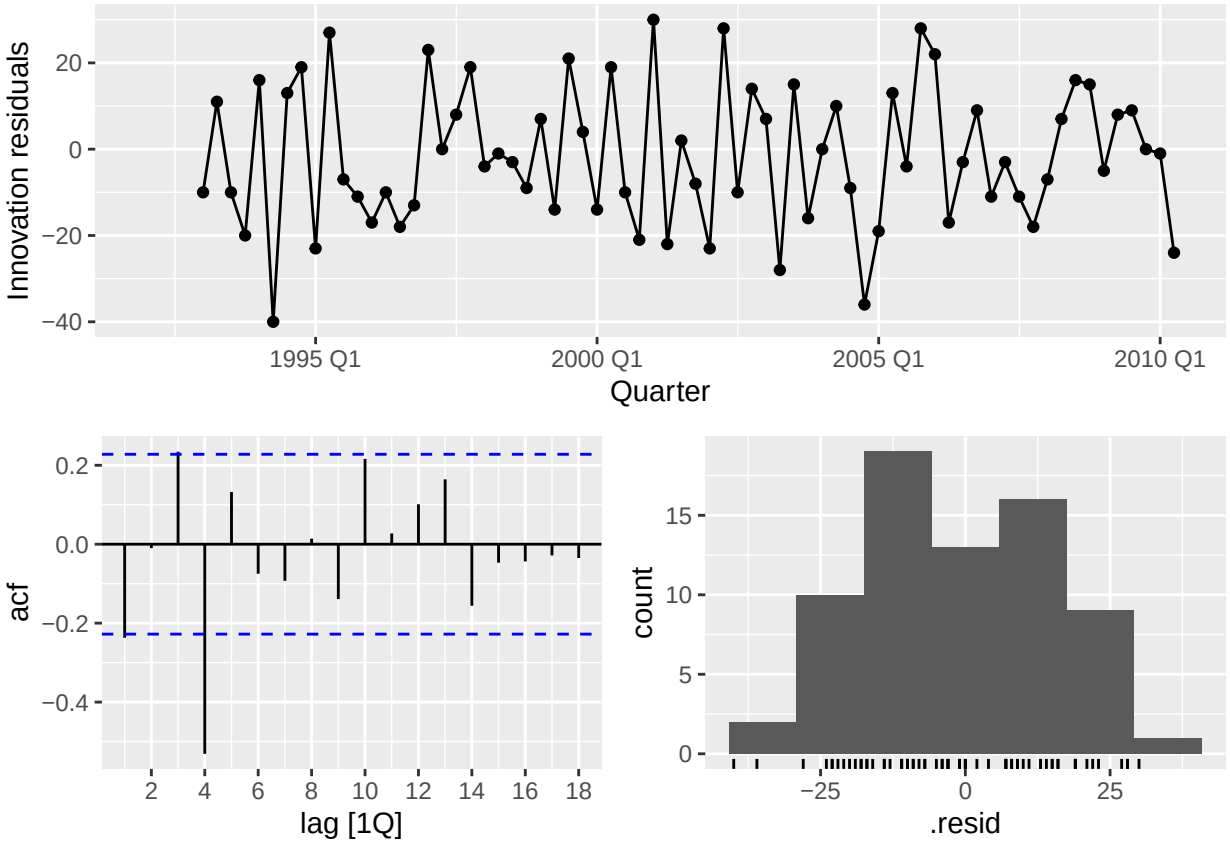


When using some of the other benchmark functions to forecast the same data set, the naive technique seems to have had the best results. This statement, of course, should be taken with a grain of salt since the forecast extends all the way to 250 from the last data point. With this in mind, the naive forecast can hold true for a good few days but will then begin to deviate from the true value. Still, a naive forecast does perform better when compared to mean and seasonal naive forecasts.

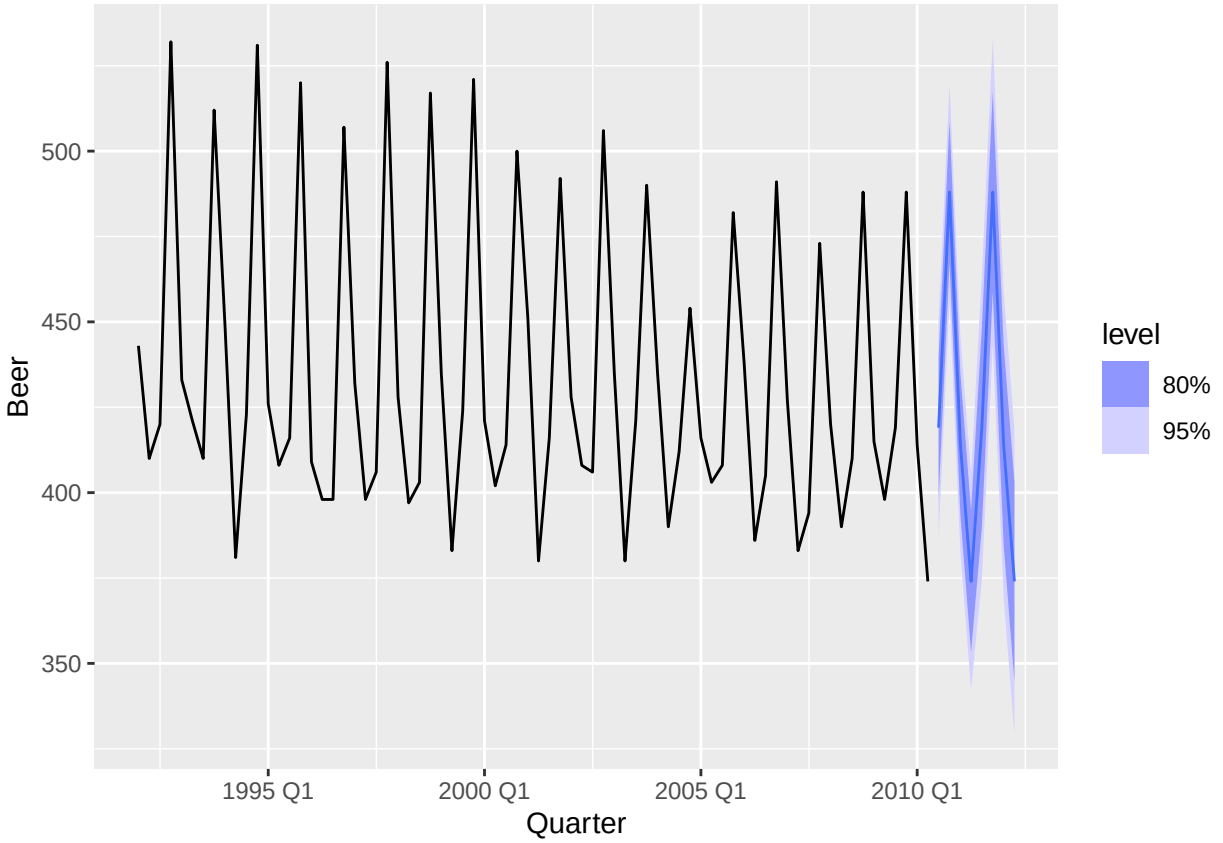
Exercise 3

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




```
# Check for significance
augment(fit) %>%
  features(.resid, ljung_box, lag=8)
```

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

What do you conclude?

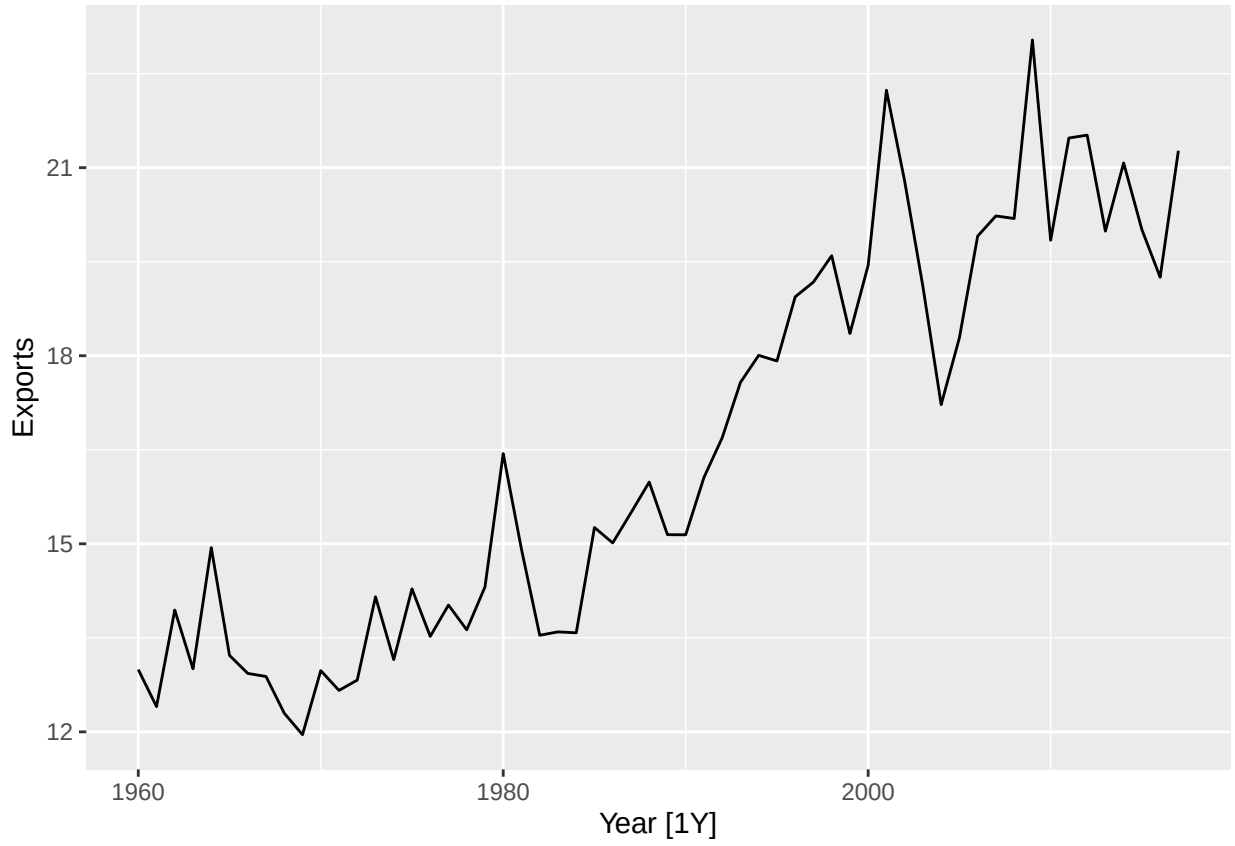
As can be seen from the visuals, there is a consistent up and down pattern in the innovation residuals chart, there are 4 lines on the lag chart extending past or near the boundary, and the distribution seems to be skewed slightly to the right. If this wasn't enough to entail room for improvement in the forecast, a Ljung-Box test was conducted which led to a large Q* value and significant p-value, indicating that this is not just white noise.

Exercise 4

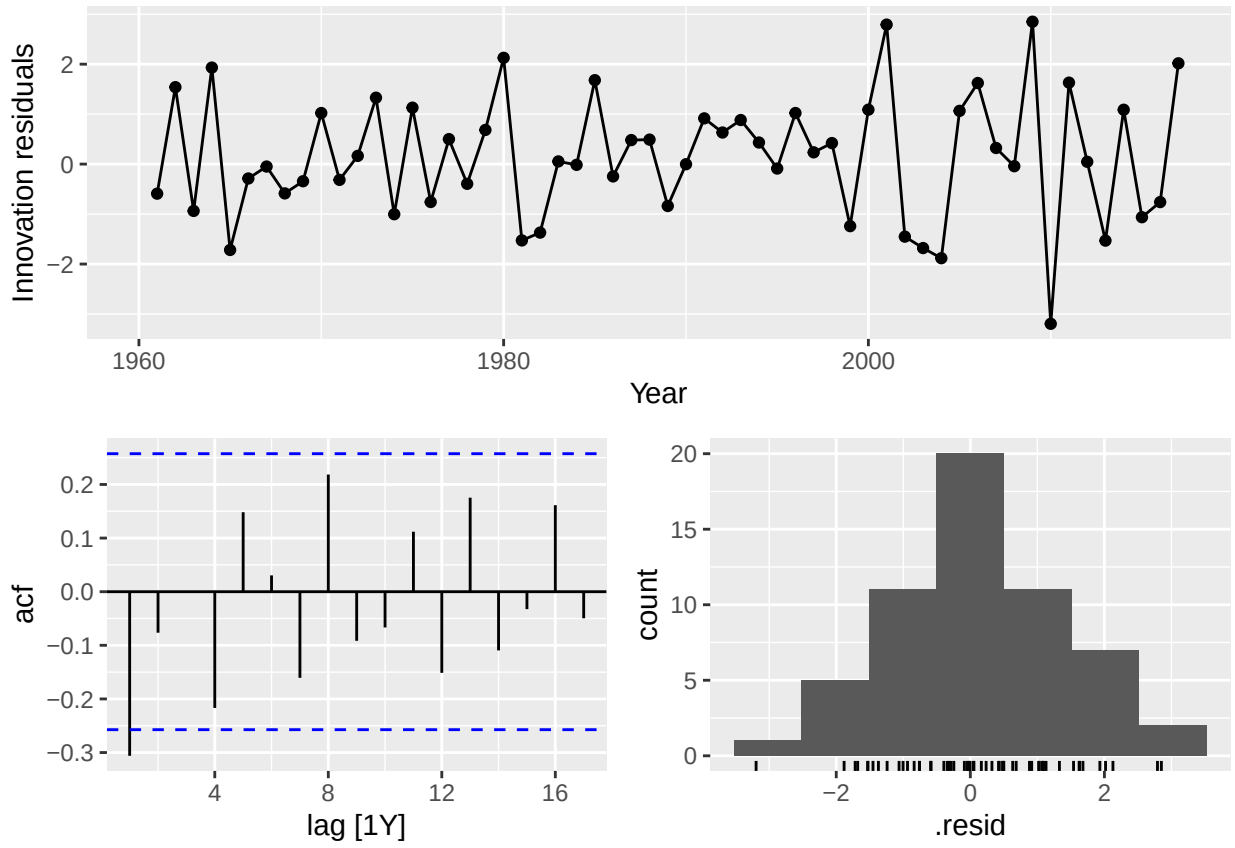
Repeat the previous exercise using the Australian Exports series from global_economy and the Bricks series from aus_production. Use whichever of NAIVE() or SNAIVE() is more appropriate in each case.

```
aus_exports <- global_economy %>%
  filter(!is.na(Exports)) %>%
```

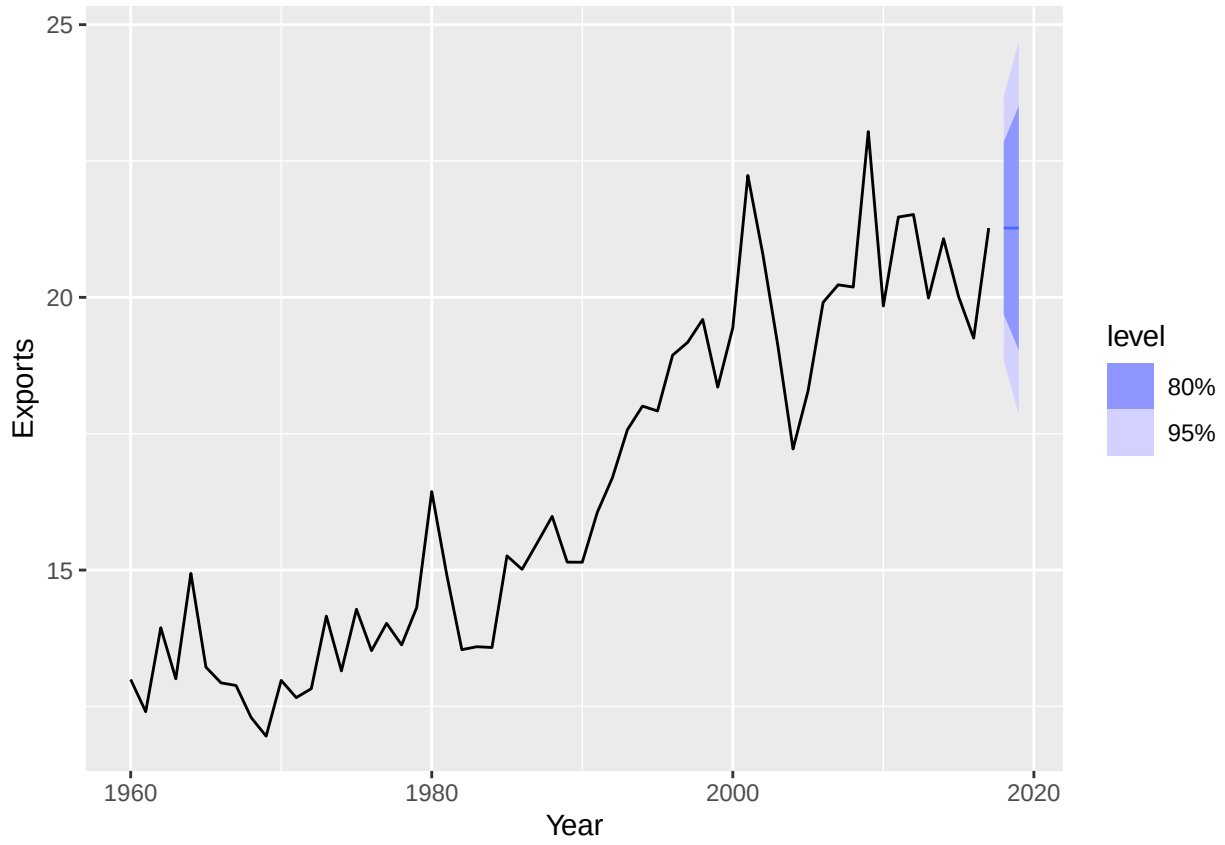
```
  filter(Country == "Australia") %>%
  select(Exports)
autoplot(aus_exports)
```




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

afit %>%
  gg_tsresiduals()
```




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

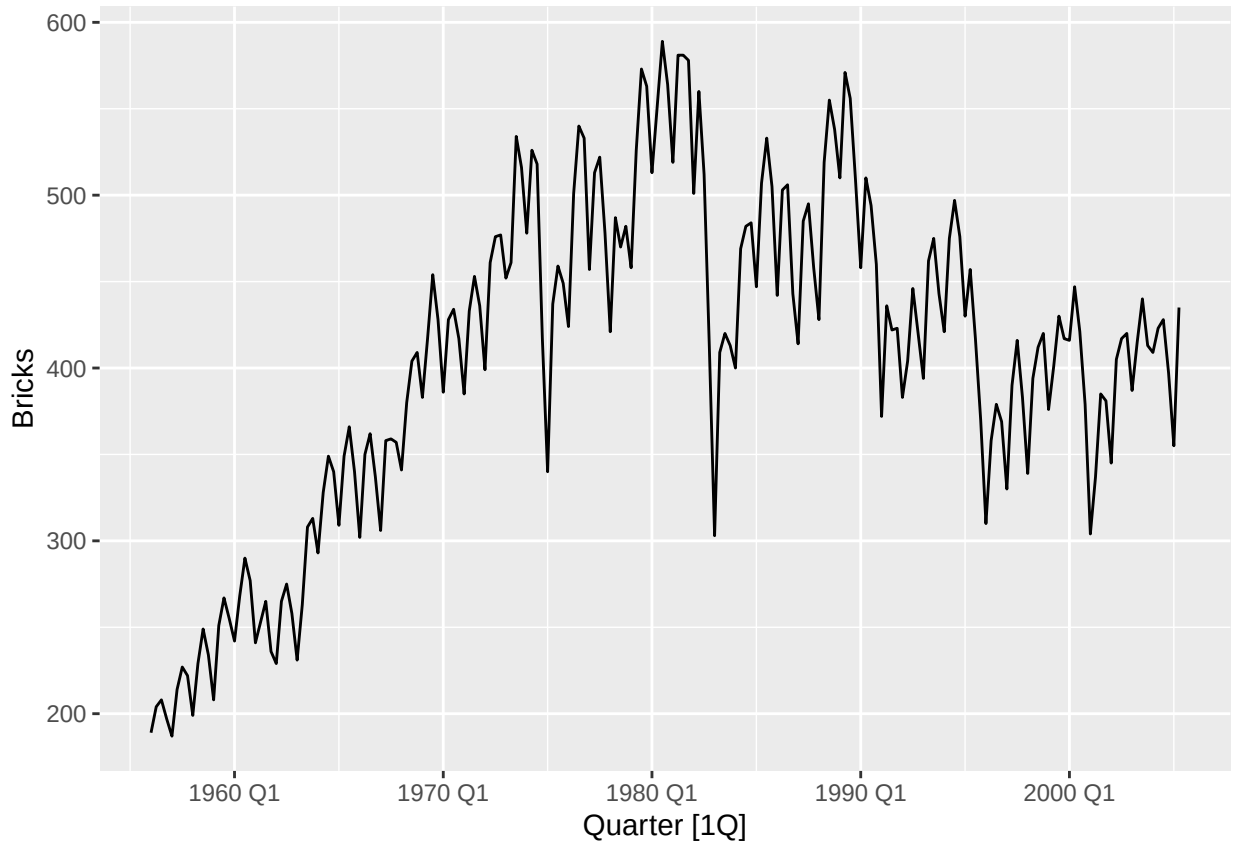



```
augment(afit) %>%
  features(.resid, ljung_box, lag=10)
```

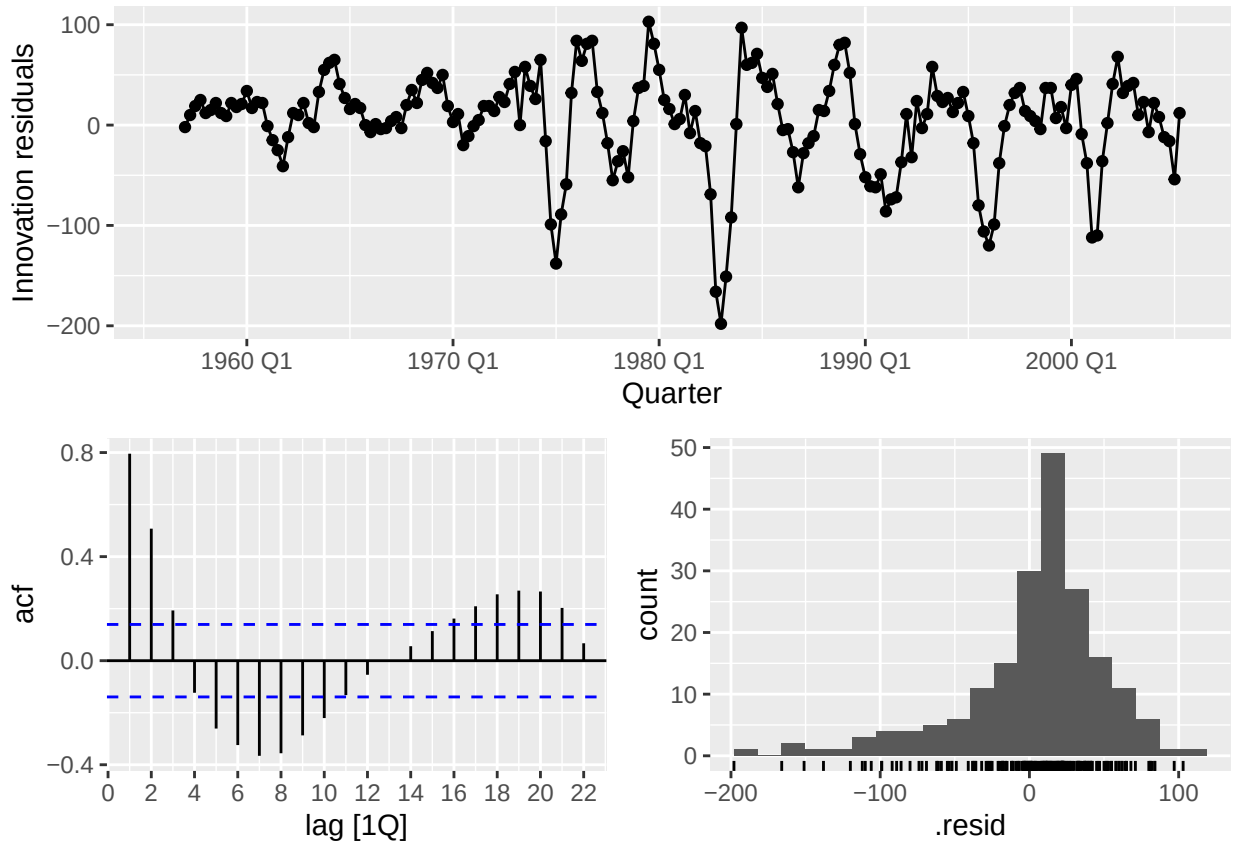
```
## # A tibble: 1 x 3
##   .model         lb_stat lb_pvalue
##   <chr>            <dbl>     <dbl>
## 1 NAIVE(Exports)    16.4    0.0896
```

The above visuals show us that the innovation residuals data for aus_exports from `global_economy` has a slight pattern but increases in volatility towards the end, there is just one line in the lags chart out of bounds, and it is more or less normally distributed. Add on top of that that the Q* value is low and the p-value is insignificant and we can conclude that what is remaining from the forecast is mostly white noise.

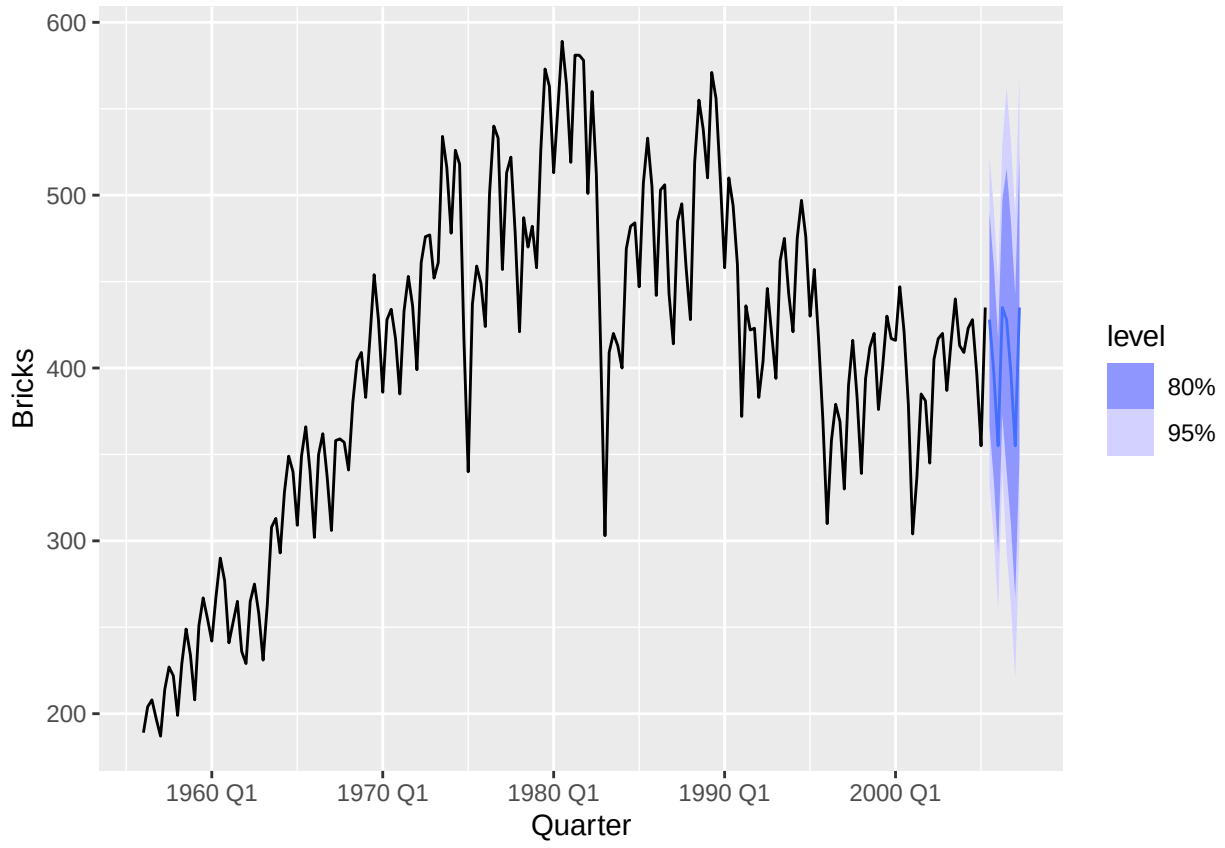
```
bricks <- aus_production %>%
  filter(!is.na(Bricks)) %>%
  select(Bricks)
autoplot(bricks)
```




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

bfit %>%
  gg_tsresiduals()
```




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




```
augment(bfit) %>%
  features(.resid, ljung_box, lag=8)
```

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

The above visuals show us that the innovation residuals data for bricks from `aus_production` has an obvious pattern with an increase in volatility towards the middle, the majority of the lag segments surpass the boundaries, and its distribution is skewed to the left. In addition, the Q* value is very large and the p-value is very significant which helps us conclude that what is remaining from the forecast is not just white noise and can be improved to take what is remaining into account.

Exercise 7

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

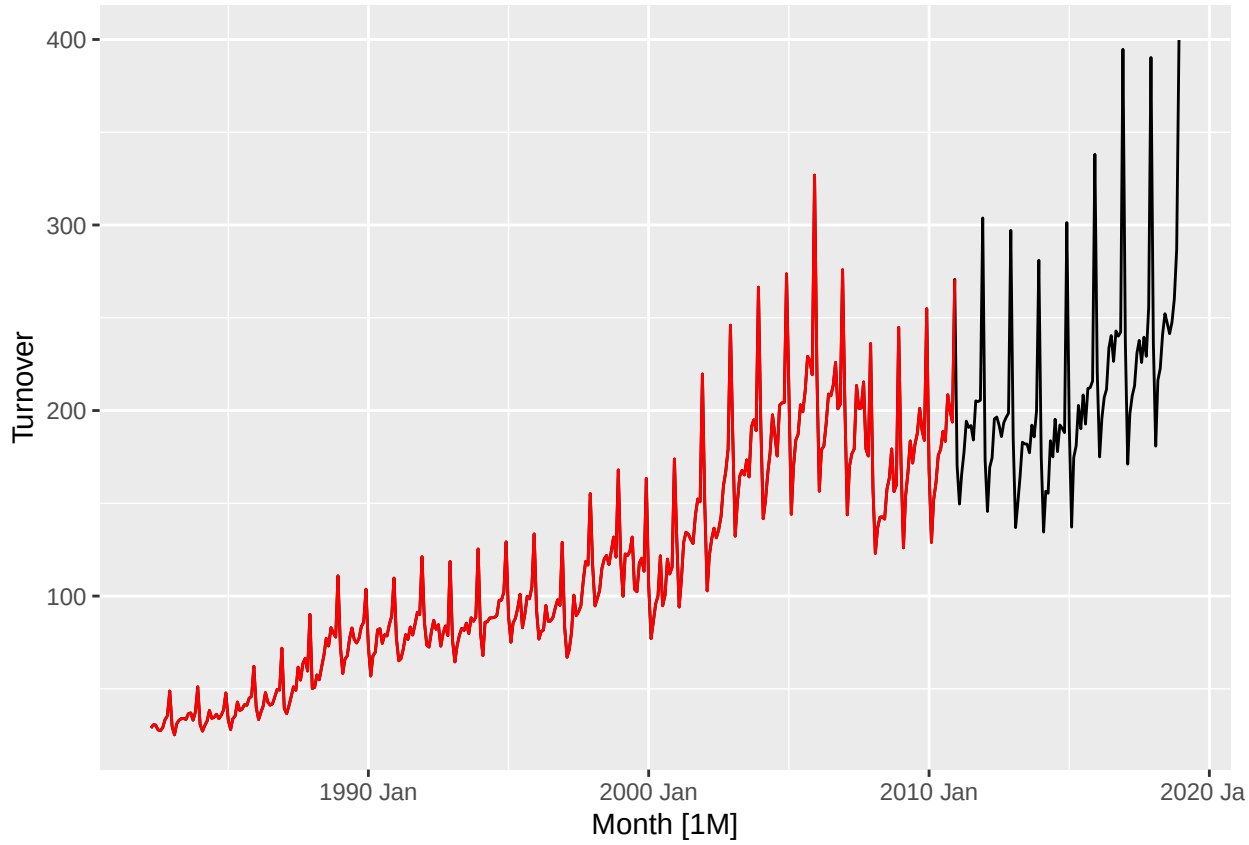
```
set.seed(1)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

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")
```

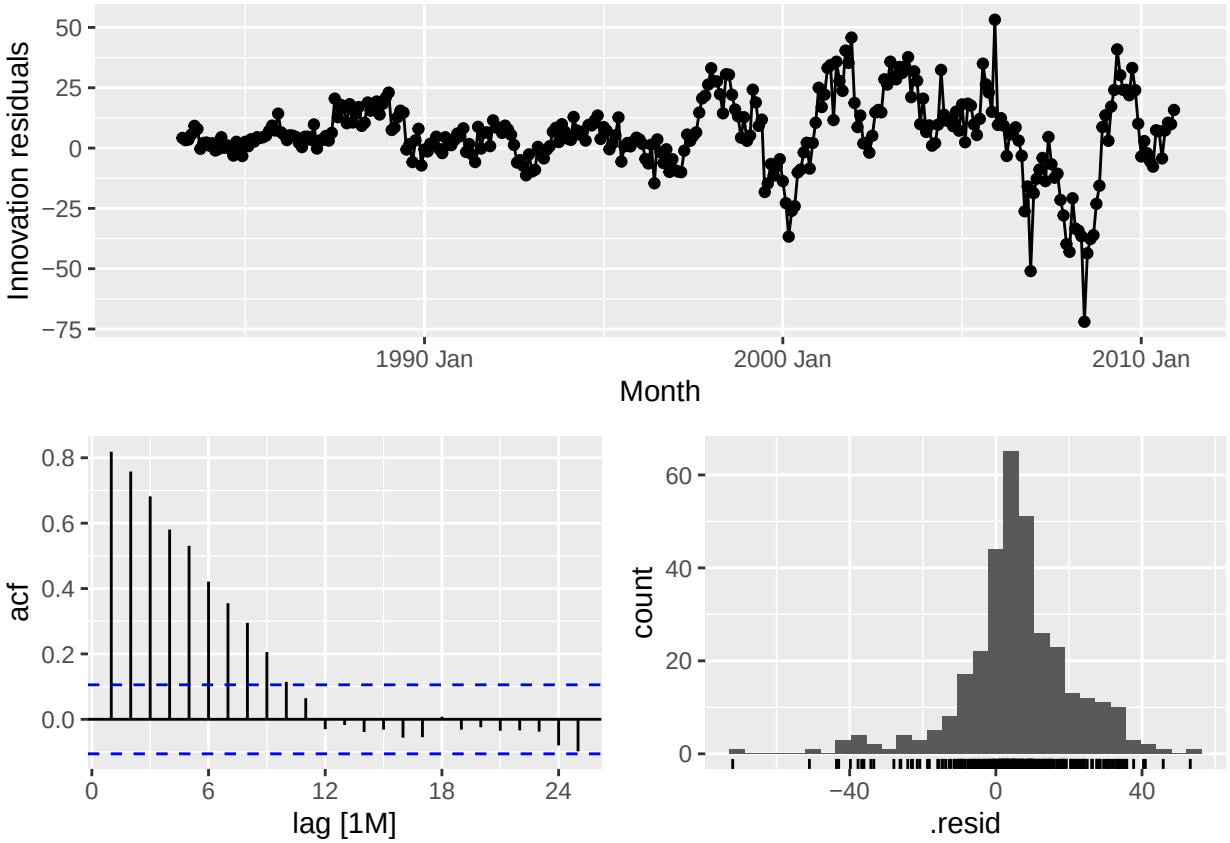


Fit a seasonal naïve model using SNAIVE() applied to your training data (myseries_train).

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

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

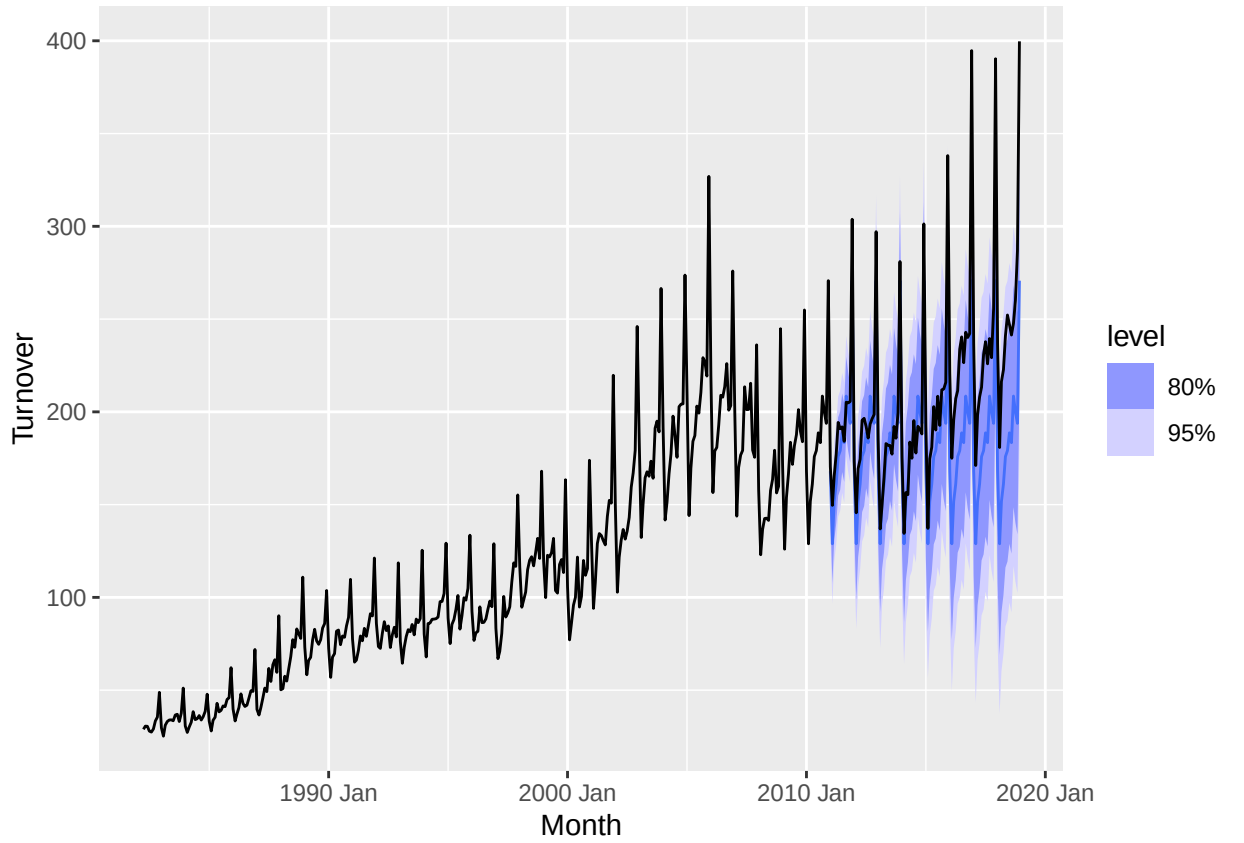
```
mfit %>%
  gg_tsresiduals()
```



As can be seen above, the residuals do not appear to be uncorrelated due to the innovation residuals chart presenting high dependency after the year 1995, almost half of the segments on the lag chart surpassing the boundaries, and their distribution being narrow and skewed to the left.

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?

The accuracy measures seem to be very sensitive to the amount of training data used. As can be seen in the results above, the measures for the test data are drastically different from the measures for the training data. This is most likely due to the fact that the training data takes into account absolutely everything before what is going to be tested irrespective of any sudden changes or fluctuations that could have taken place towards the end of the data. Meanwhile, the test data, depending on the very last seasonal period in this case, uses the pattern at the very end and repeats it over and over. Given the drastic change in trend between the years 2005-2010, this would lead to the forecast having the errors we see above.