The Forecasters Toolbox- HW3

Umer Farooq

2024-02-14

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).

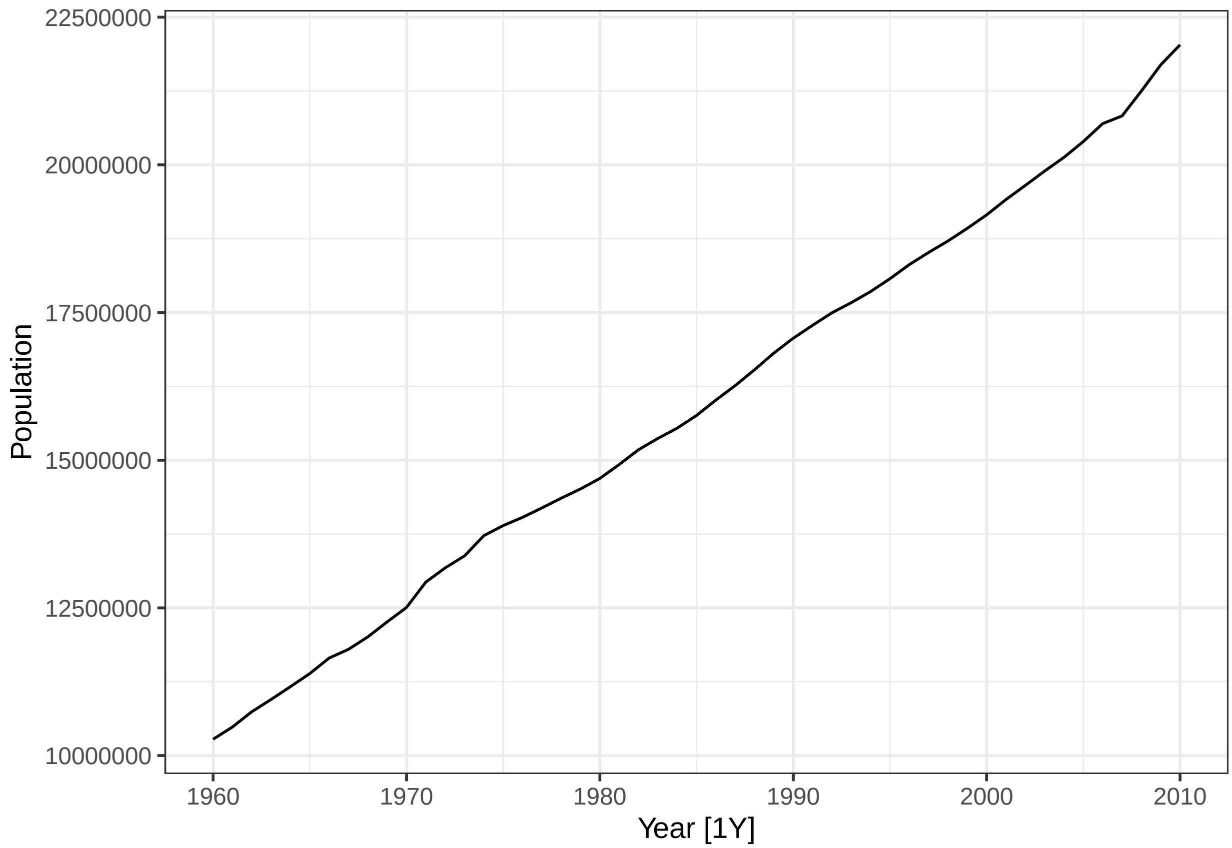
Answer:

Australian Population (global_economy):

Let's check out the time series first

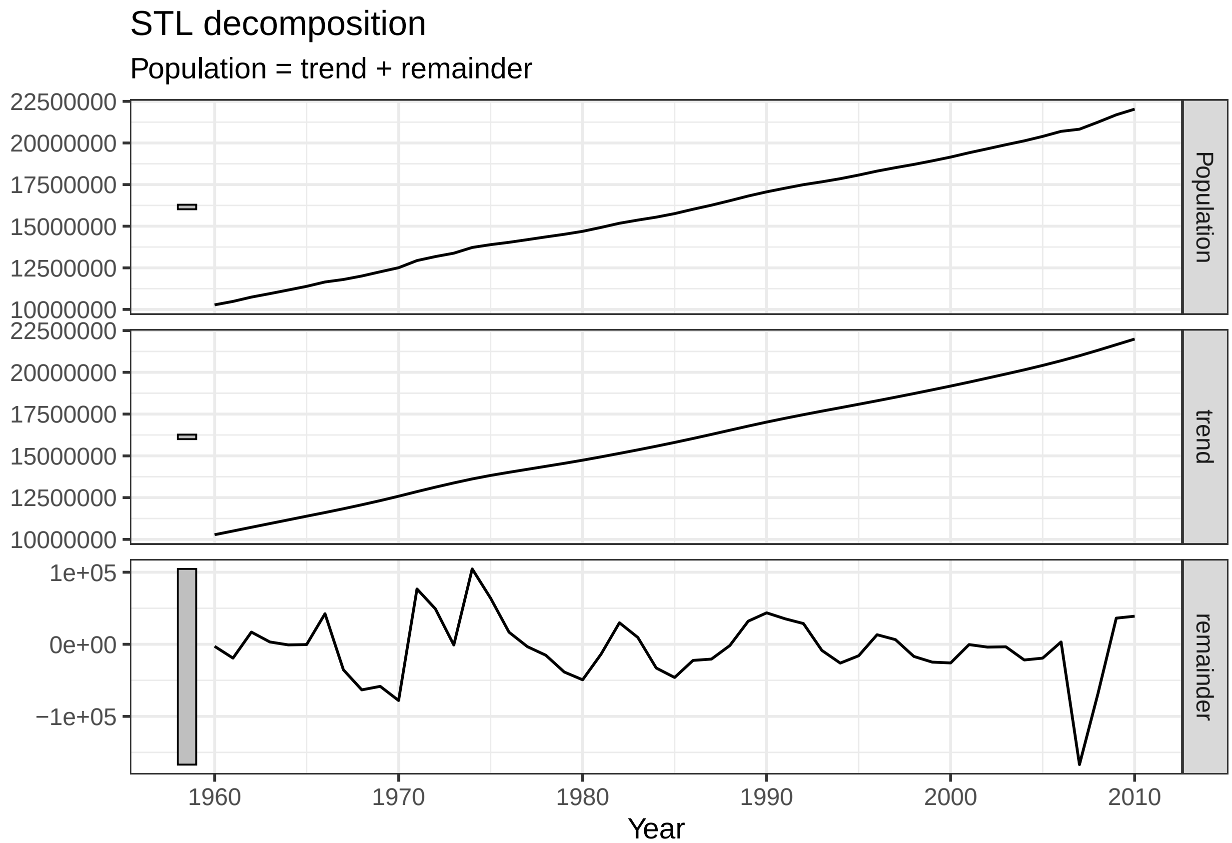
```
aus_pop <- global_economy|>
  filter(Country == 'Australia')|>
  select(Country, Year, Population)
aus_pop_2010 <- global_economy|>
  filter(Country == 'Australia', Year <= 2010)|>
  select(Country, Year, Population)
```

```
aus_pop_2010|>
  autoplot(Population)+theme_bw()
```



As we can see that there is no seasonality or cyclic behavior but rather just an increasing trend we can further make sure by decomposing the time series using STL decomposition

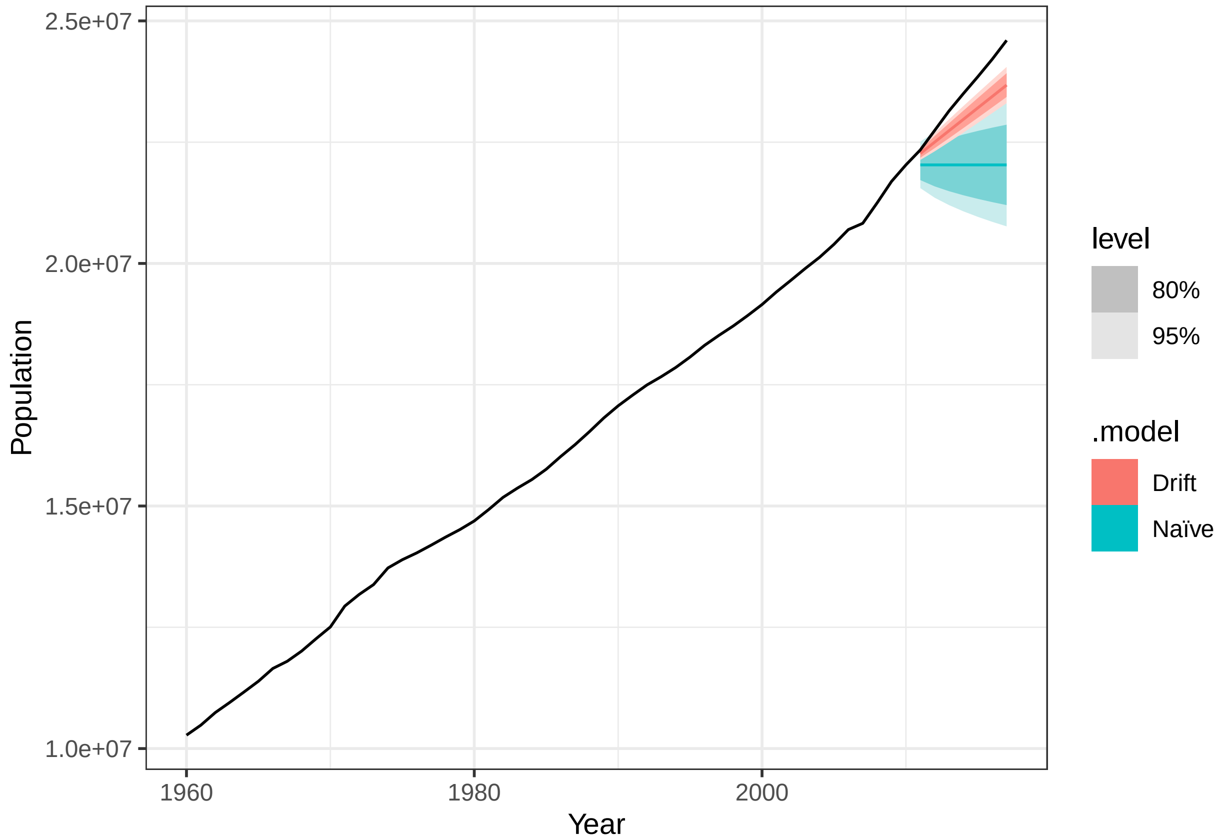
```
aus_pop_2010 |>
  model(
    STL(Population ~ trend(window = 7) +
                   season(window = "periodic"),
    robust = TRUE)) |>
  components() |>
  autoplot()+theme_bw()
```



As we confirmed that ther is only an increasing trend in the series with no seasonality so I believe that tha drift technique would performed better than any other technique. So lets create model for Drift and Naive.

```
aus_fit_2010 <- aus_pop_2010 |>
  model(    `Naïve` = NAIVE(Population),
    Drift = RW(Population ~ drift())
  )
```

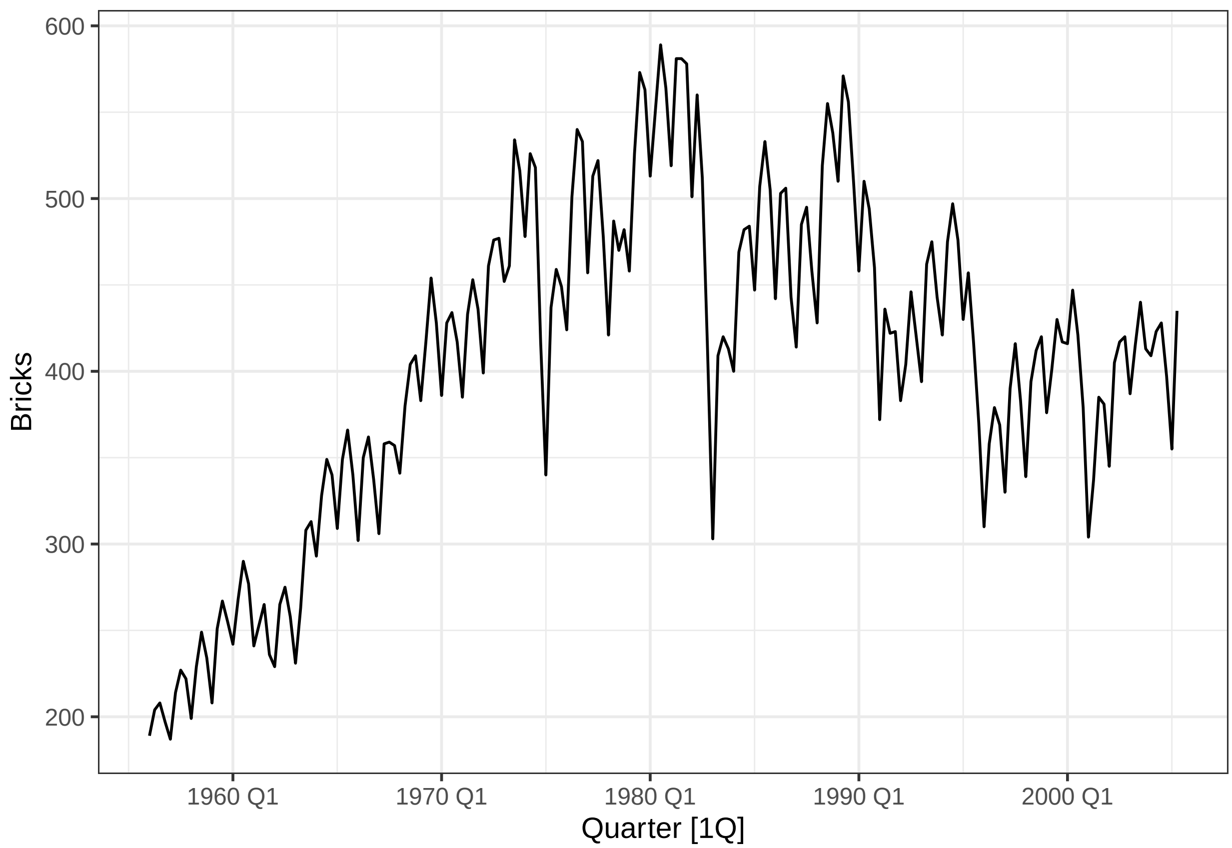
```
aus_fit_2010|>
  forecast(h = 7)|>
  autoplot(aus_pop)+theme_bw()
```



Bricks (aus_production):

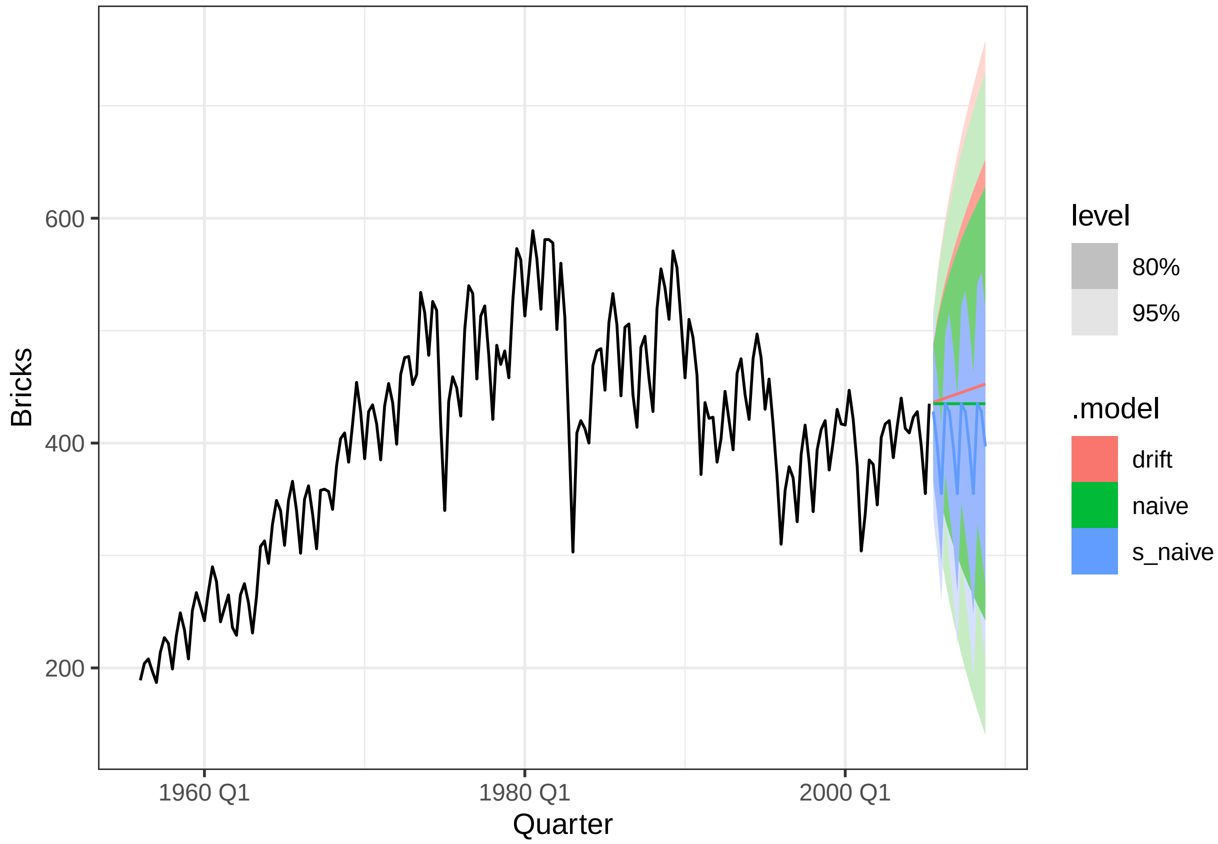
```
aus_bricks <- aus_production|>
  filter(!is.na(Bricks))|>
  select(Quarter, Bricks)
```

```
aus_bricks|>
  autoplot(Bricks)+theme_bw()
```




```
aus_bricks_model <- aus_bricks|>
  model(s_naive = SNAIVE(Bricks),
        `naive` = NAIVE(Bricks),
        drift = RW(Bricks~drift()))
```

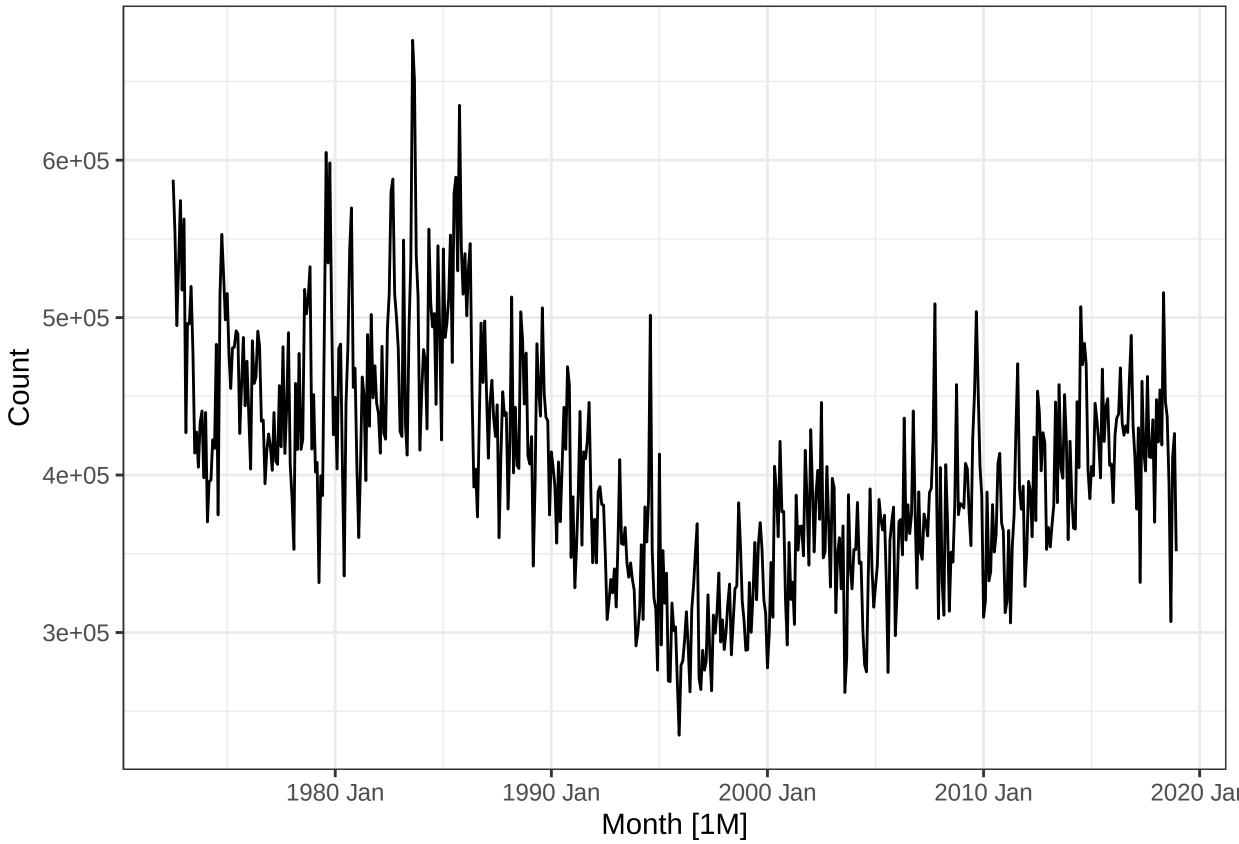
```
aus_bricks_model|>
  forecast(h = 14)|>
  autoplot(aus_bricks)+theme_bw()
```



NSW Lambs (aus_livestock):

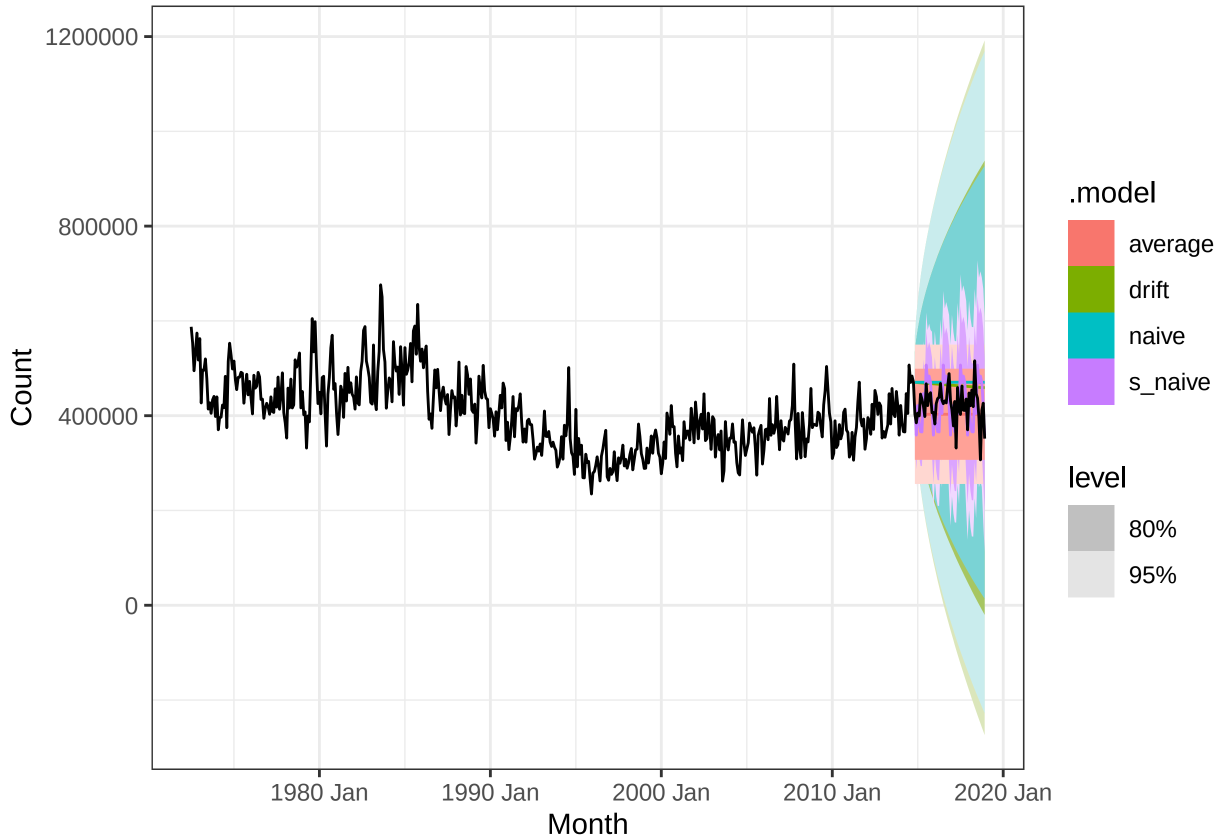
```
aus_livestock <- mutate(aus_livestock, Month = yearmonth(Month))

# Now let's redo the filtering
lambs_nsw <- aus_livestock %>%
  mutate(ind = yearmonth(Month))|>
  filter(Animal == 'Lambs', State == 'New South Wales')

filtered_lambs <- lambs_nsw %>%
  filter(Animal == 'Lambs', State == 'New South Wales') %>%
  filter(ind <= yearmonth("Oct 2014"))
```

```
lambs_nsw|>
  autoplot(Count)+theme(legend.position = 'none')+theme_bw()
```



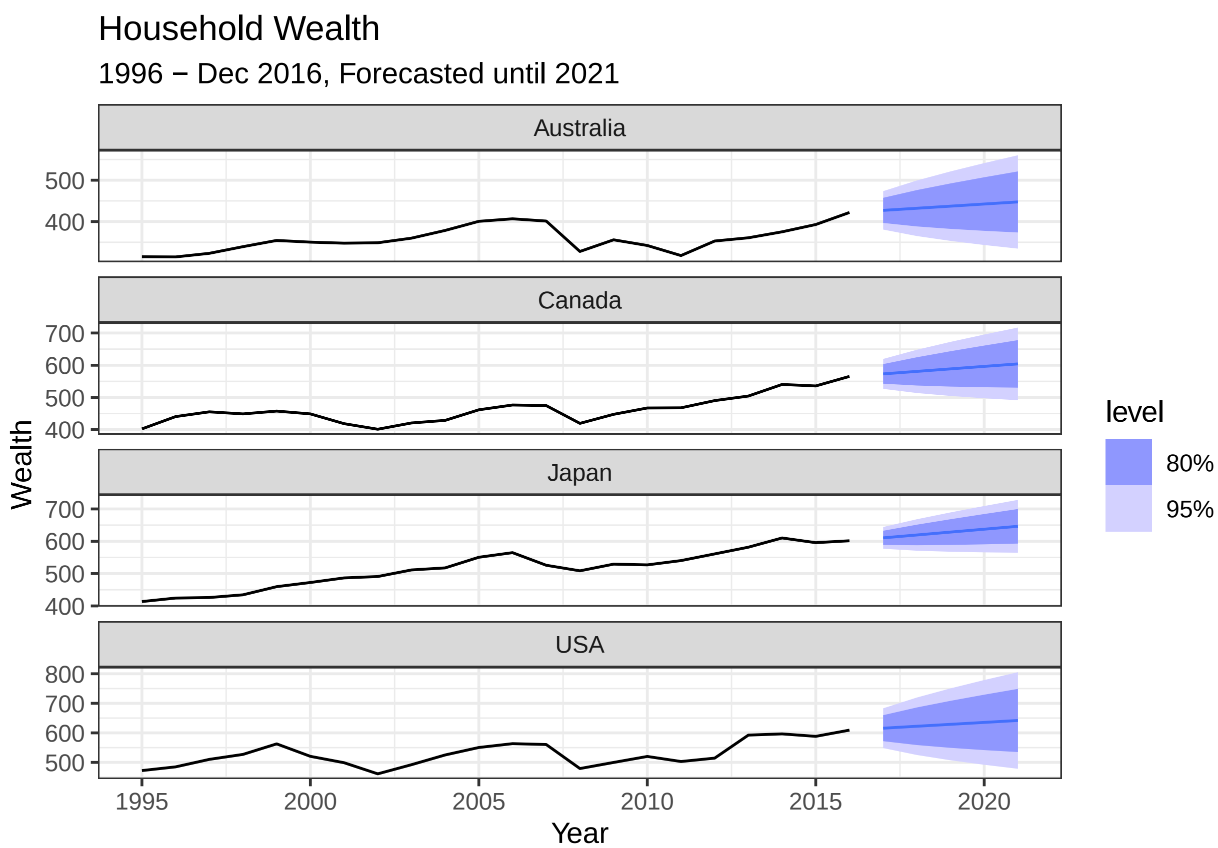

```
lambs_nsw_model <-  filtered_lambs|>
  model ( average = MEAN(Count),
          `naive` = NAIVE(Count),
          s_naive = SNAIVE(Count),
          drift = RW(Count~drift()))
```

```
lambs_nsw_model|>
  forecast(h = 50)|>
  autoplot(lambs_nsw)+theme_bw()
```



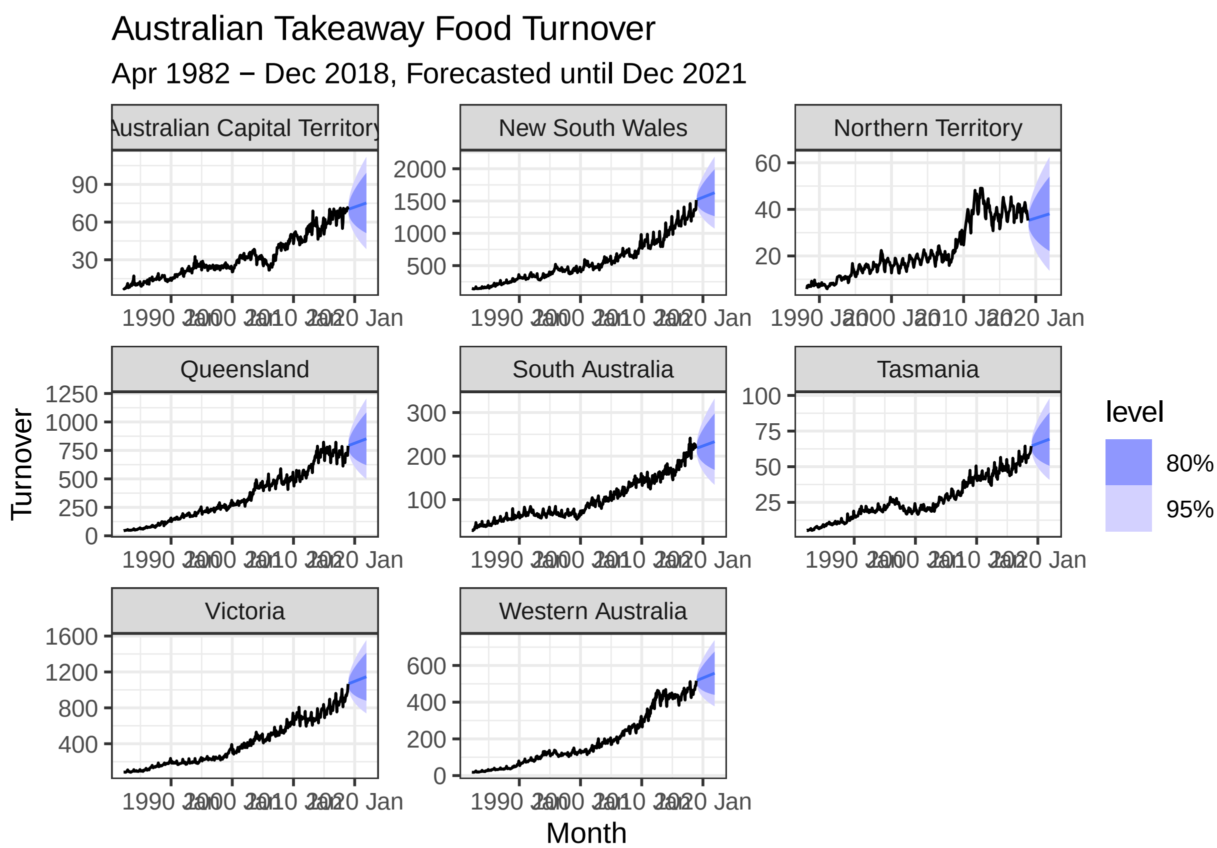
Household wealth (hh_budget).

```
hh_budget %>%
  model(RW(Wealth ~ drift())) %>%
  forecast(h = 5) %>%
  autoplot(hh_budget) +
  labs(title = "Household Wealth",
       subtitle = "1996 - Dec 2016, Forecasted until 2021")+theme_bw()
```



Australian takeaway food turnover (aus_retail):

```
aus_retail %>%
  filter(Industry == "Cafes, restaurants and takeaway food services") %>%
  model(RW(Turnover ~ drift())) %>%
  forecast(h = 36) %>%
  autoplot(aus_retail) +
  labs(title = "Australian Takeaway Food Turnover",
       subtitle = "Apr 1982 - Dec 2018, Forecasted until Dec 2021") +
  facet_wrap(~State, scales = "free")+theme_bw()
```



---

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?

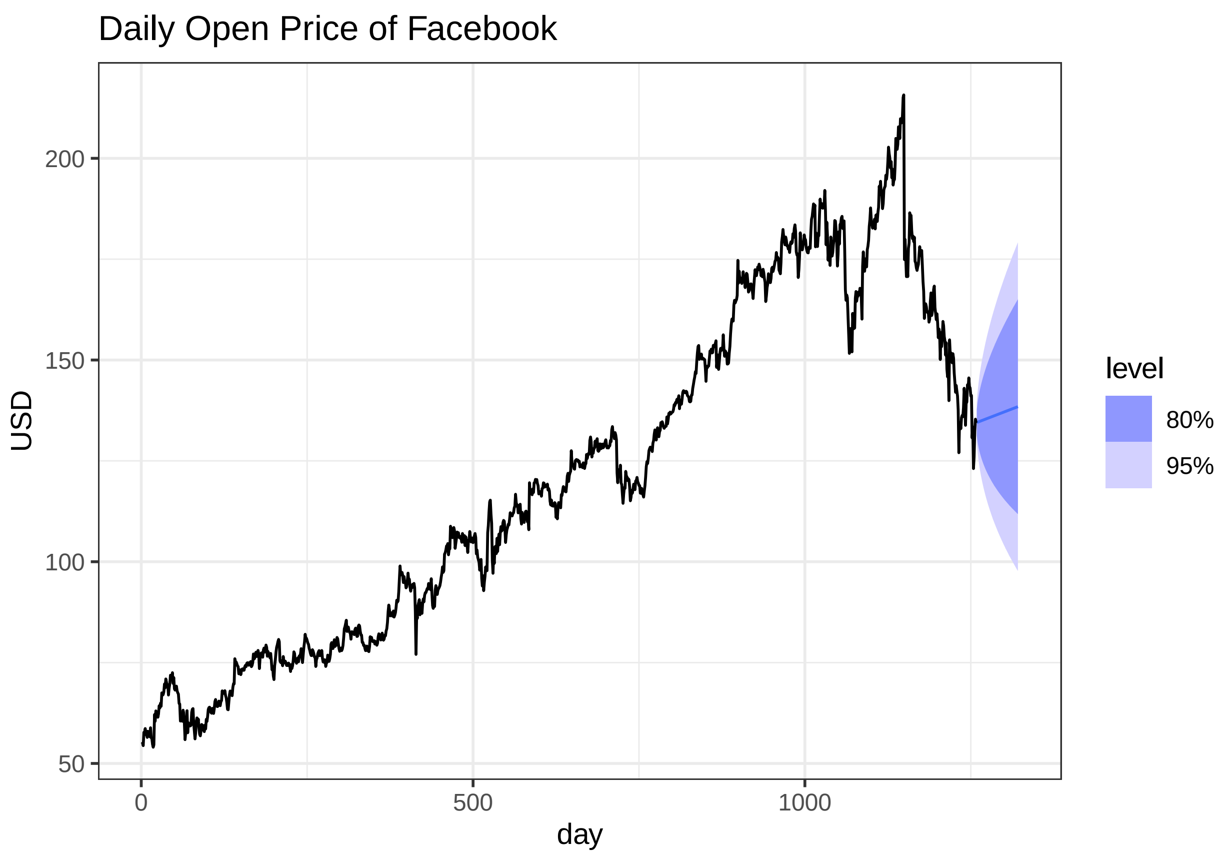
Answer:

a. Produce a time plot of the series:

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

fb_stock%>%
  autoplot(Open) +
  labs(title= "Daily Open Price of Facebook", y = "USD")+theme_bw()
```

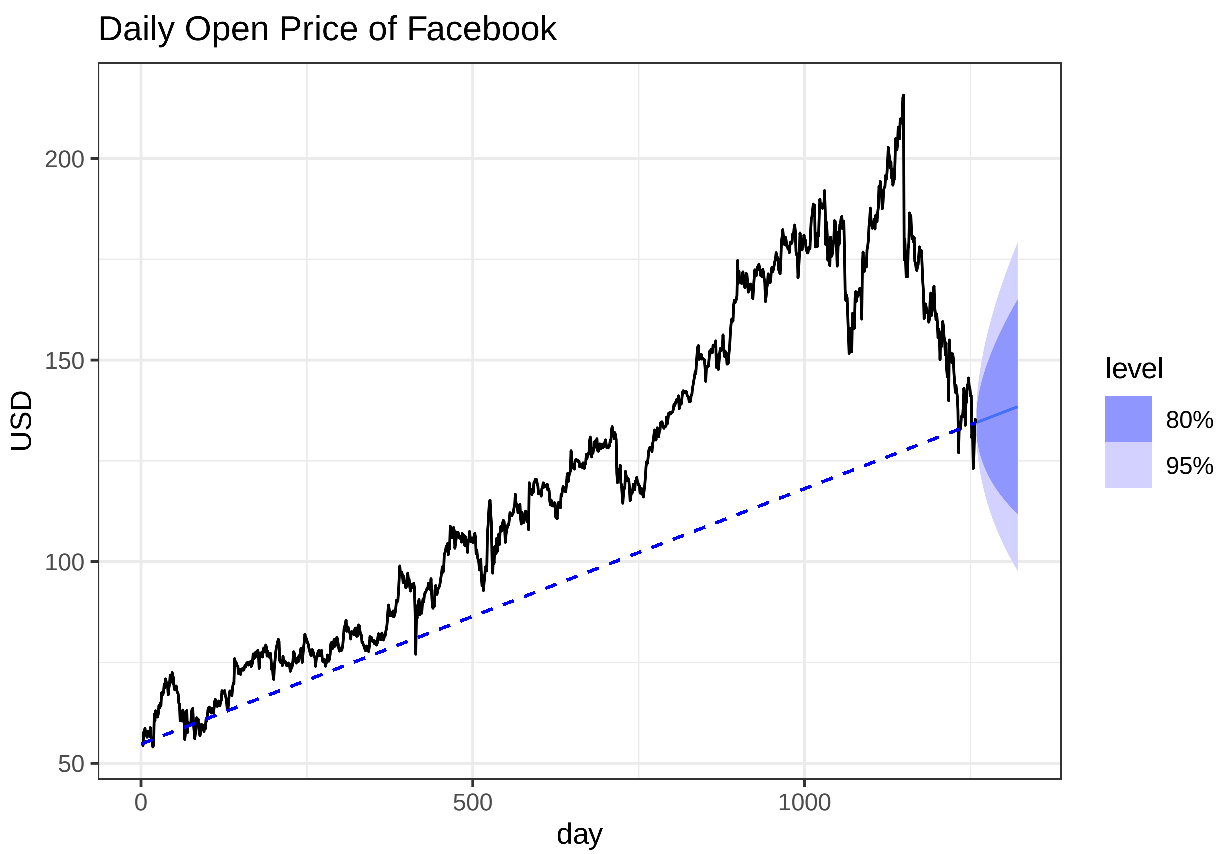
b. Produce forecasts using the drift method and plot them:

```
fb_stock %>%
  model(RW(Open ~ drift())) %>%
  forecast(h = 63) %>%
  autoplot(fb_stock) +
  labs(title = "Daily Open Price of Facebook", y = "USD")+theme_bw()
```



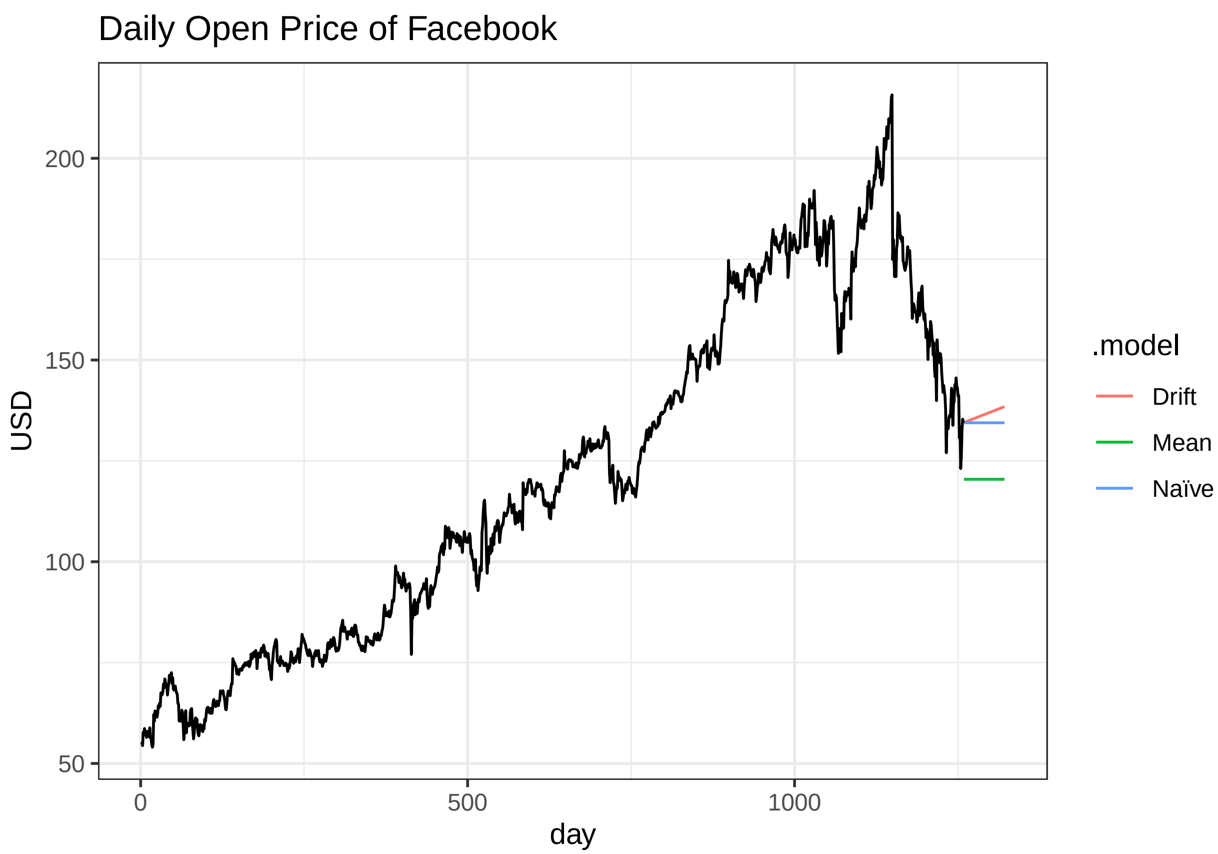
c. Show that the forecasts are identical to extending the line drawn between the first and last observations:

```r
fb_stock %>%
  model(RW(Open ~ drift())) %>%
  forecast(h = 63) %>%
  autoplot(fb_stock) +
  labs(title = "Daily Open Price of Facebook", y = "USD") +
  geom_segment(aes(x = 1, y = 54.83, xend = 1258, yend = 134.45),
               colour = "blue", linetype = "dashed")+theme_bw()
```



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

```
fb_stock %>%
  model(Mean = MEAN(Open),
        `Naïve` = NAIVE(Open),
        Drift = NAIVE(Open ~ drift())) %>%
  forecast(h = 63) %>%
  autoplot(fb_stock, level = NULL) +
  labs(title = "Daily Open Price of Facebook", y = "USD")+theme_bw()
```

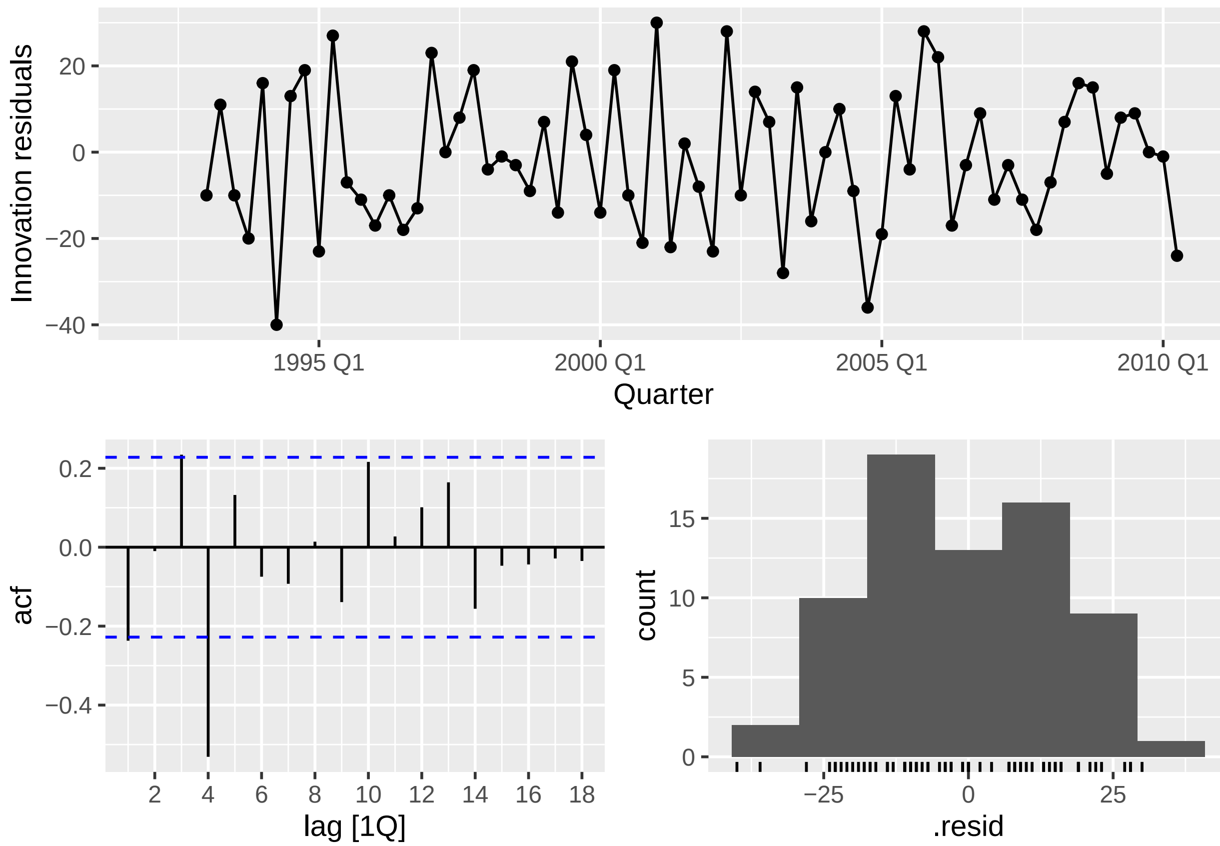


---

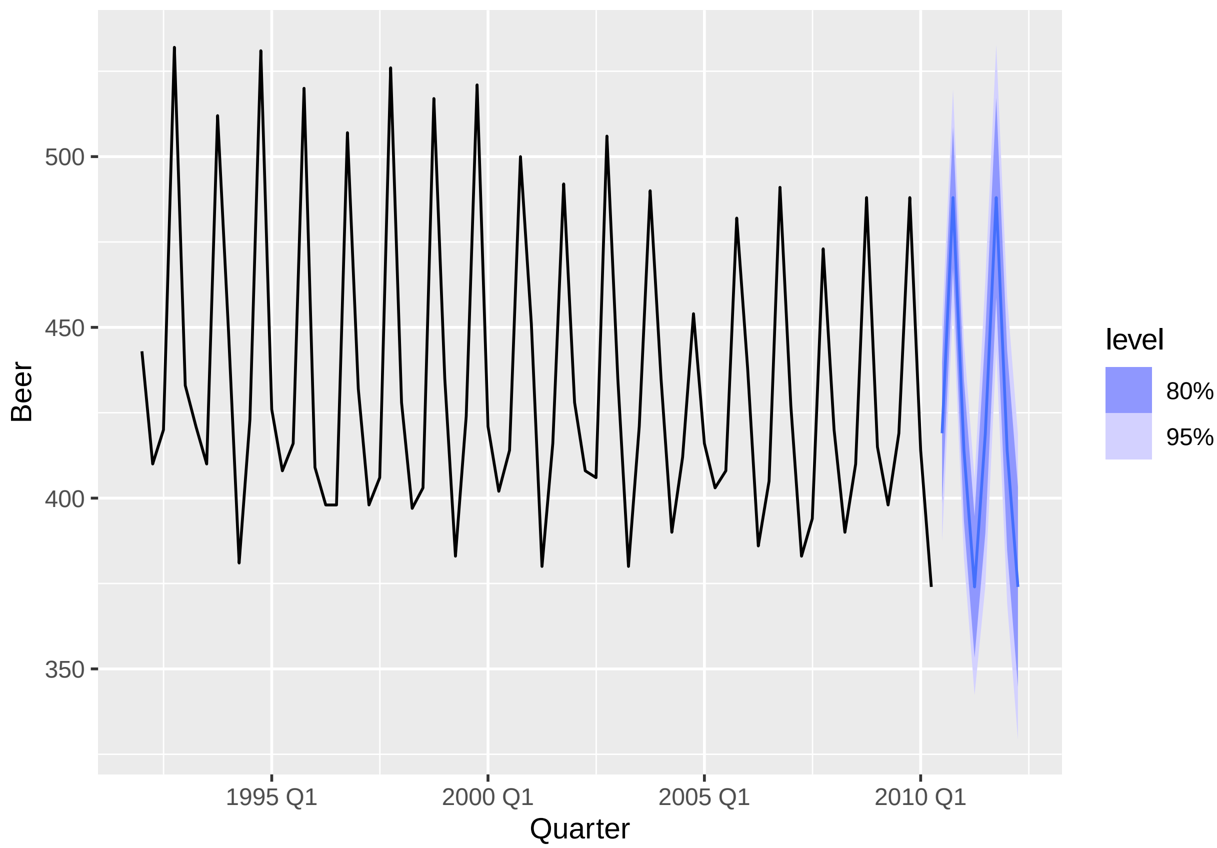
3. Apply a seasonal naïve method to the quarterly Australian beer production data from 1992. Check if the residuals look like white noise, and plot the forecasts

Answer:

```
# 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)
```




```r
fit |>
  augment() |>
  features(.innov, box_pierce, lag = 8, dof = 0)
```

```
## # A tibble: 1 x 3
##   .model       bp_stat bp_pvalue
##   <chr>          <dbl>     <dbl>
## 1 SNAIVE(Beer)    29.7  0.000234
```

```r
fit %>%
  augment()%>% features(.innov, ljung_box, lag = 8, dof = 0)
```

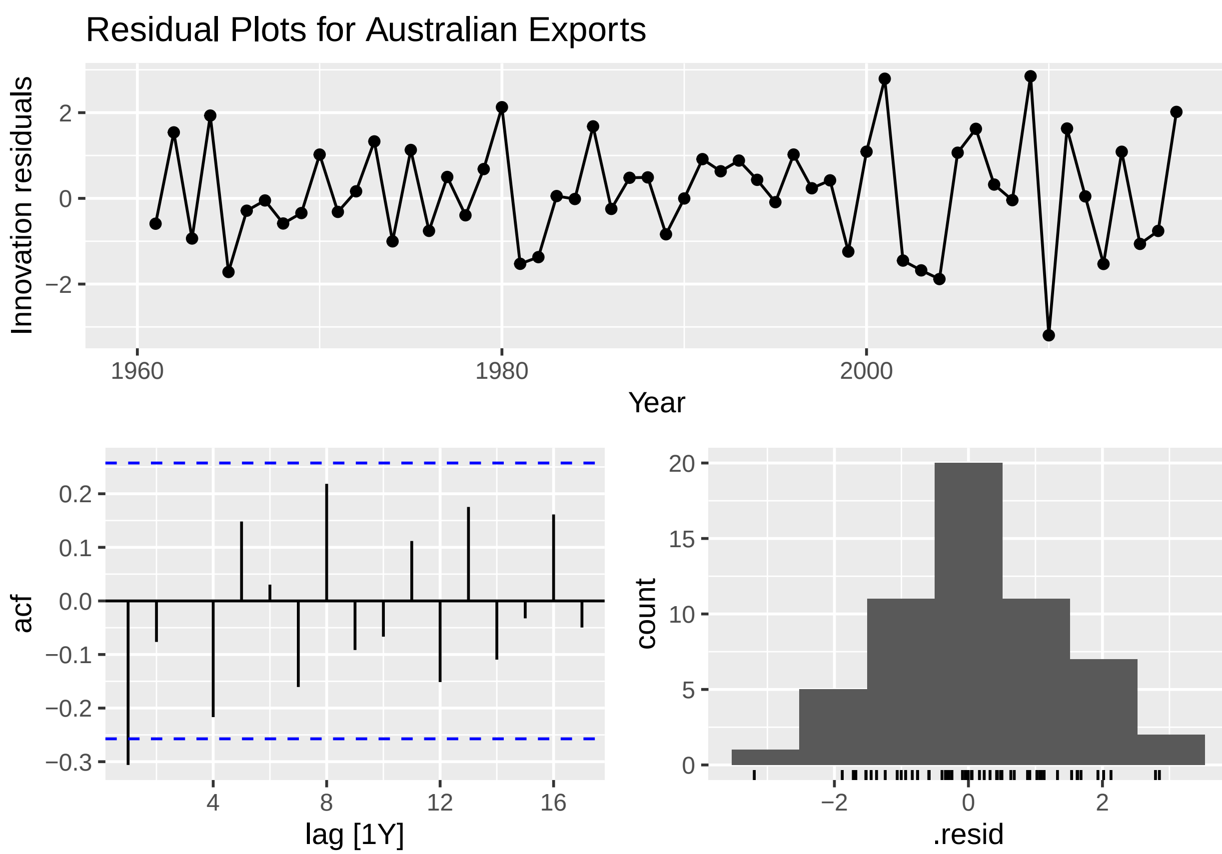
```
## # A tibble: 1 x 3
##   .model       lb_stat lb_pvalue
##   <chr>          <dbl>     <dbl>
## 1 SNAIVE(Beer)    32.3 0.0000834
```

The tests show that the results are distinguishable from a white noise series since the p-values are relatively small. The results are not white noise, as the residuals seem to be centered around zero and follow a constant variance. The ACF plot shows that lag 4 is larger than the others which can be attributed to peaks occurring every 4 quarters in Q4, and troughs occurring every Q2

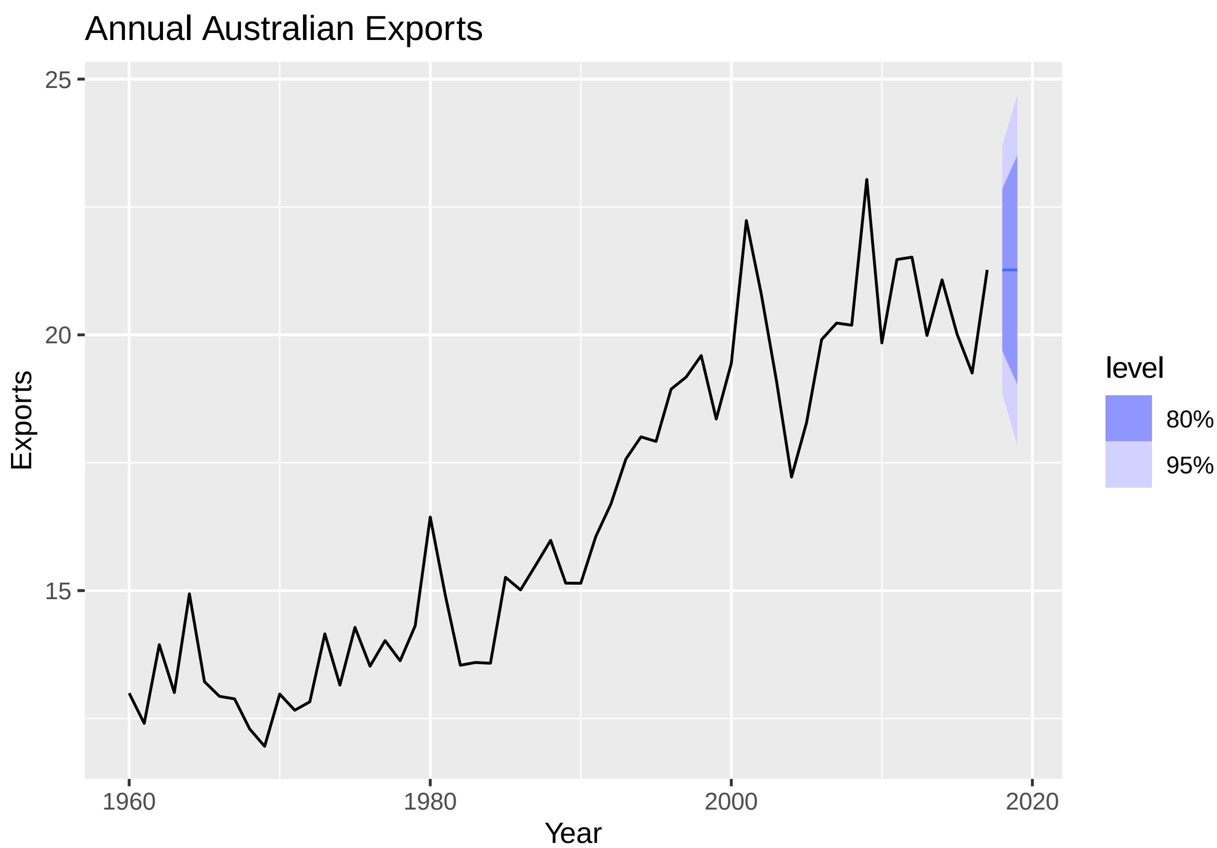
4. Repeat the previous exercise using the Australian Exports series from global_economy and the Bricks series from aus_production. Use whichever of NAIVE() or SNAIVE() is more appropriate in each case.

```
# Extract data of interest
aus_exports <- global_economy %>%
  filter(Country == "Australia")

# Define and estimate a model
fit <- aus_exports %>% model(NAIVE(Exports))

# Look at the residuals
fit %>% gg_tsresiduals() +
  ggtitle("Residual Plots for Australian Exports")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).

## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```




```
# Look at some forecasts
fit %>% forecast() %>% autoplot(aus_exports) +
  ggtitle("Annual Australian Exports")
```




```r
#Box-Pierce test,  =10 for non-seasonal data
fit %>%
  augment() %>%
  features(.innov, box_pierce, lag = 10, dof = 0)
```

```
## # A tibble: 1 x 4
##   Country   .model          bp_stat bp_pvalue
##   <fct>     <chr>             <dbl>     <dbl>
## 1 Australia NAIVE(Exports)     14.6     0.148
```

```r
fit %>%
  augment()%>% features(.innov, ljung_box, lag = 10, dof = 0)
```

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

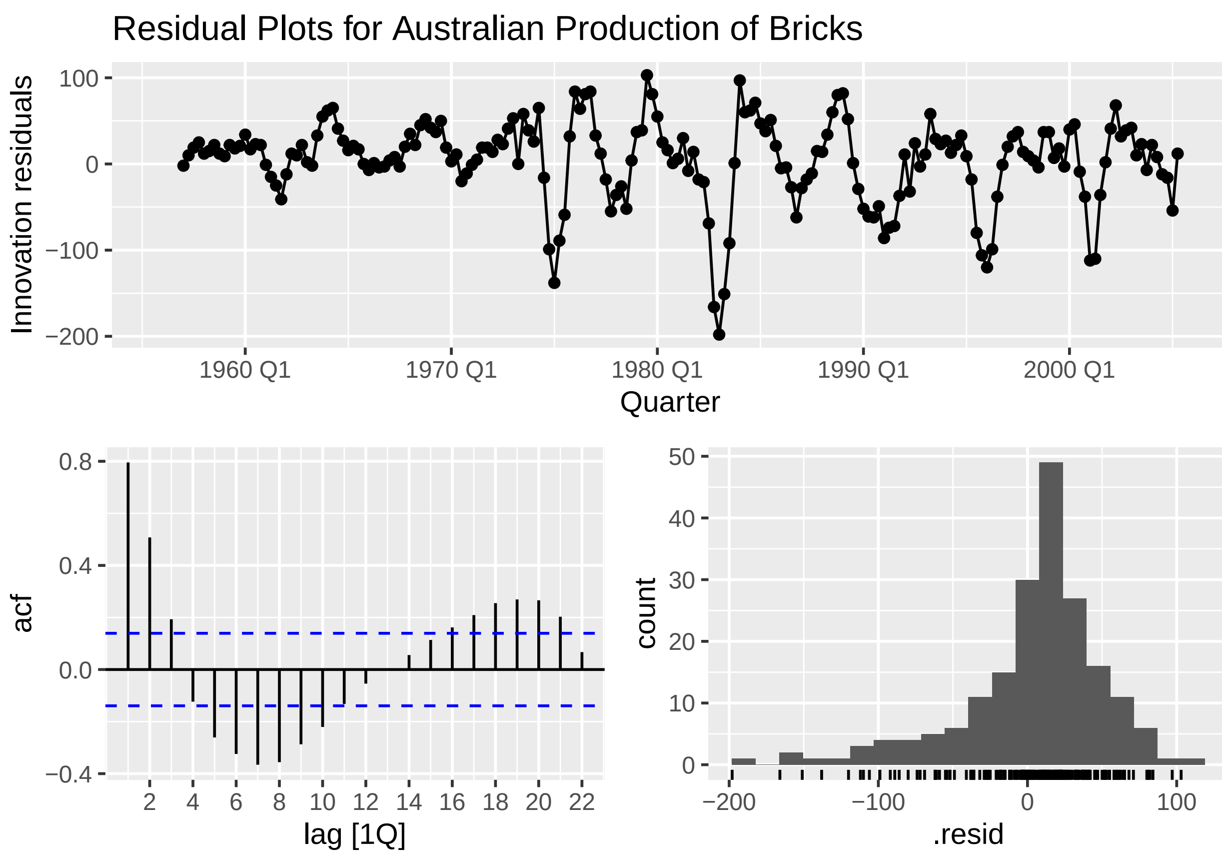
Since it is yearly data, it would be best to use the NAIVE() method. The mean of the residuals is close to zero and they seem to have constant variation except from 2000 to 2010. The ACF plot shows there is some autocorrelation at lag 1. The Box-Pierce and Ljung-Box tests further show that the results are not significant at a significance level of p=0.05. This shows that the residuals are not distinguishable from white noise.

```r
# Define and estimate a model
fit <- aus_production %>%
  filter(!is.na(Bricks)) %>%
  model(SNAIVE(Bricks))

# Look at the residuals
fit %>% gg_tsresiduals() +
  ggtitle("Residual Plots for Australian Production of Bricks")
```

```
## Warning: Removed 4 rows containing missing values (`geom_line()`).
```

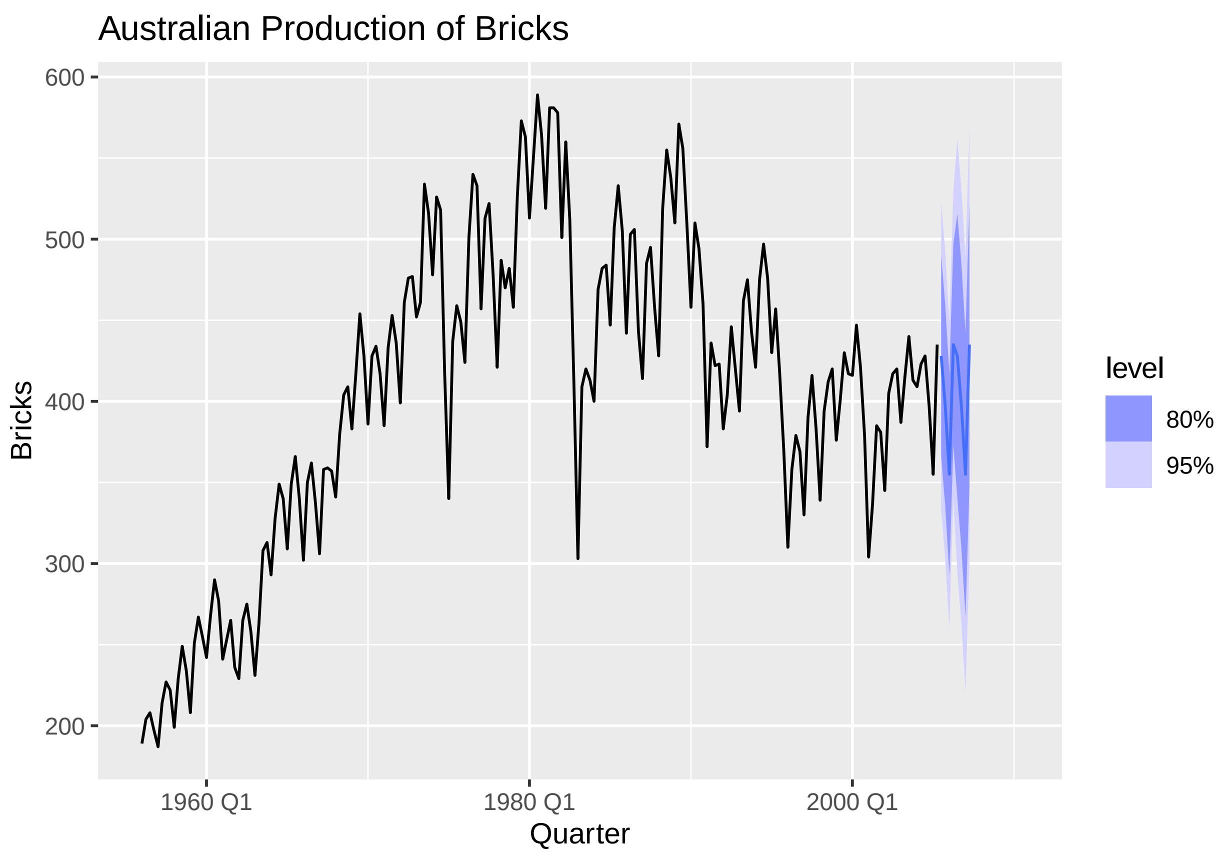
```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).
```




```r
# Look at some forecasts
fit %>% forecast() %>% autoplot(aus_production) +
  ggtitle("Australian Production of Bricks")
```

```
## Warning: Removed 20 rows containing missing values (`geom_line()`).
```




```r
#Box-Pierce test,  =2m for seasonal data, m=4
fit %>%
  augment() %>%
  features(.innov, box_pierce, lag = 8, dof = 0)
```

```
## # A tibble: 1 x 3
##   .model          bp_stat bp_pvalue
##   <chr>             <dbl>     <dbl>
## 1 SNAIVE(Bricks)    267.          0
```

```r
fit %>%
  augment()%>% features(.innov, ljung_box, lag = 8, dof = 0)
```

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

There is a seasonal pattern in the manufacturing production of bricks, so it is best to use the SNAIVE() method. The results from the autocorrelation tests are significant, which shows that the residuals are distinguishable from a white noise series. Furthermore, the residuals do not follow a normal distribution as it not centered around 0 and left skewed. The ACF is also interesting as there seems to be waves.

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

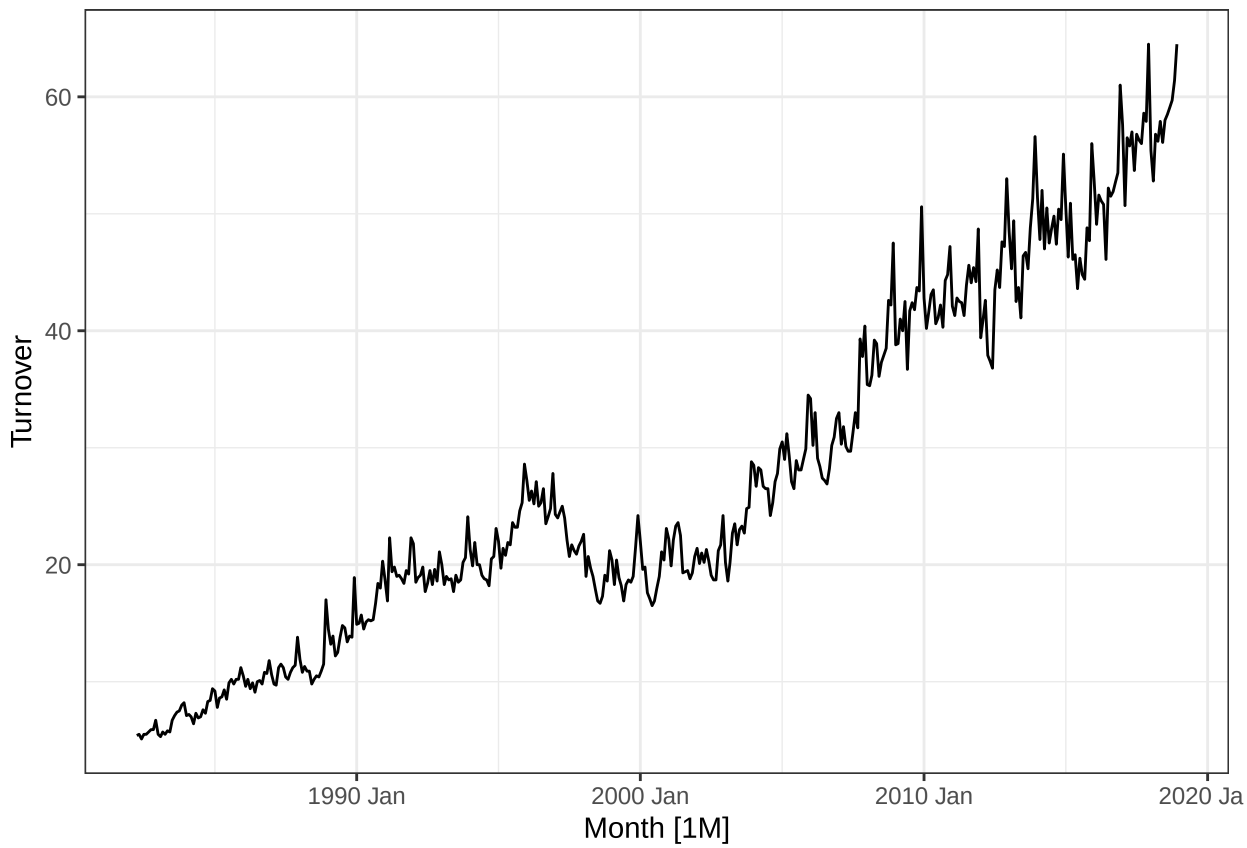
a. Create a training dataset consisting of observations before 2011 using

Here is the time series from Section 2.10 Exercise 7.

```
set.seed(1234)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

Let's check it out visually

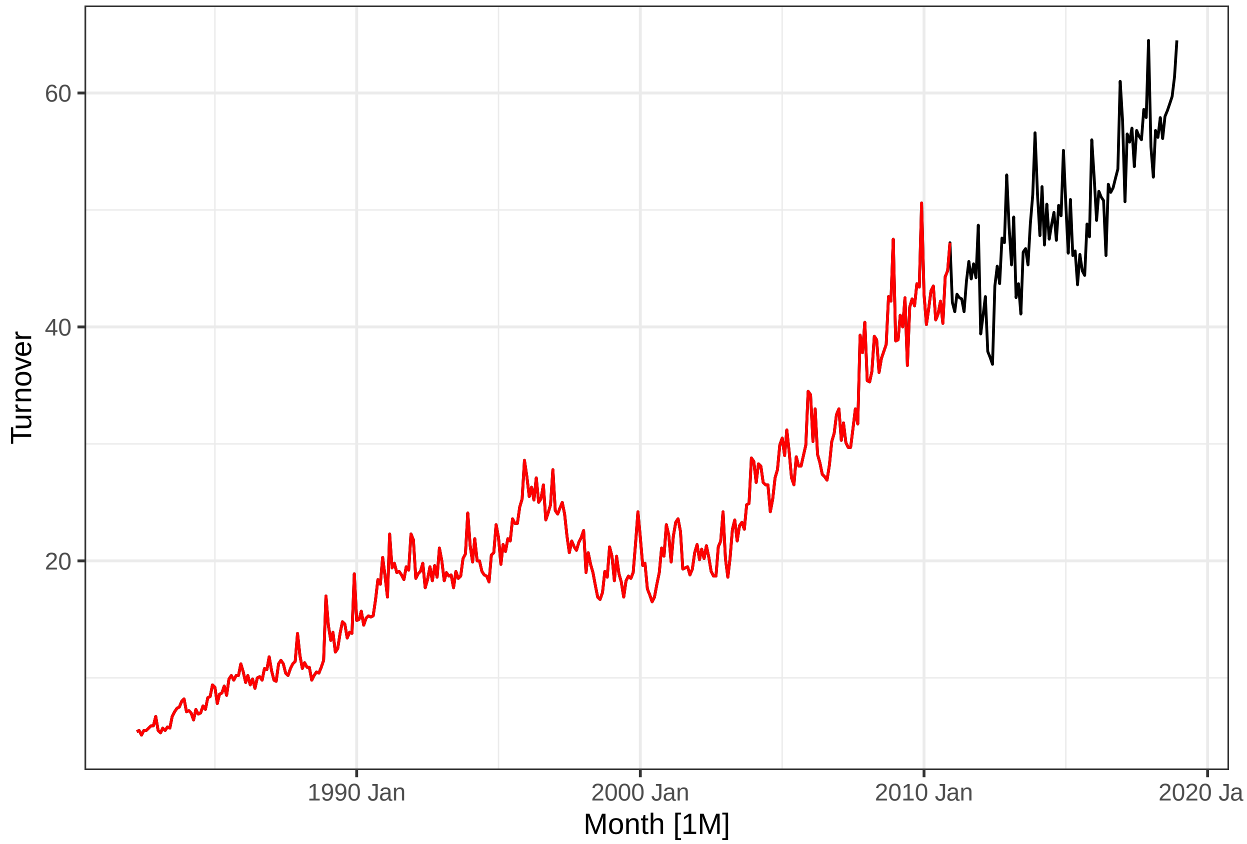
```
autoplot(myseries,.vars=Turnover)+theme_bw()
```




```
myseries_train <- myseries %>%
  filter(year(Month) < 2011, !is.na(Turnover))
```

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

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



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

```
fit <- myseries_train |>
  model(SNAIVE(Turnover ~ lag(12)))
```

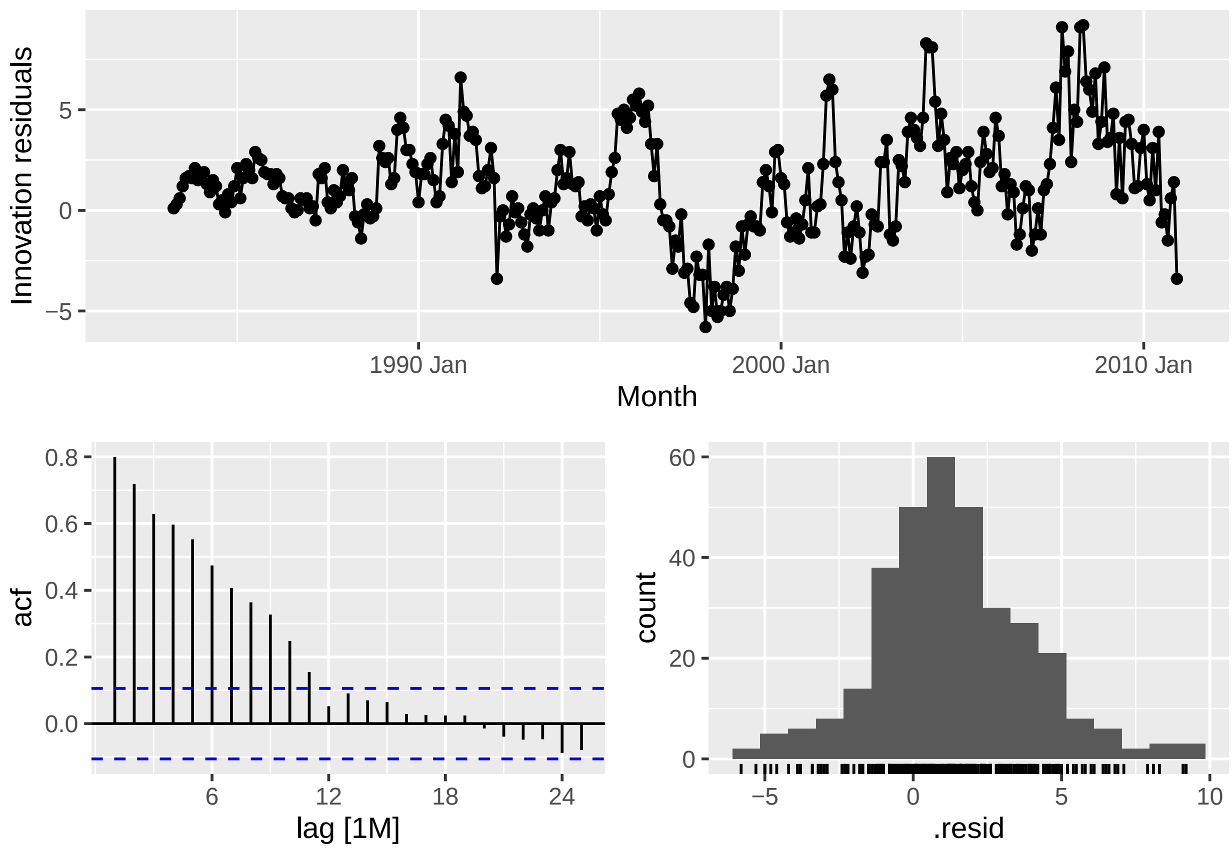
d. Check the residuals

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

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 12 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 12 rows containing non-finite values (`stat_bin()`).
```



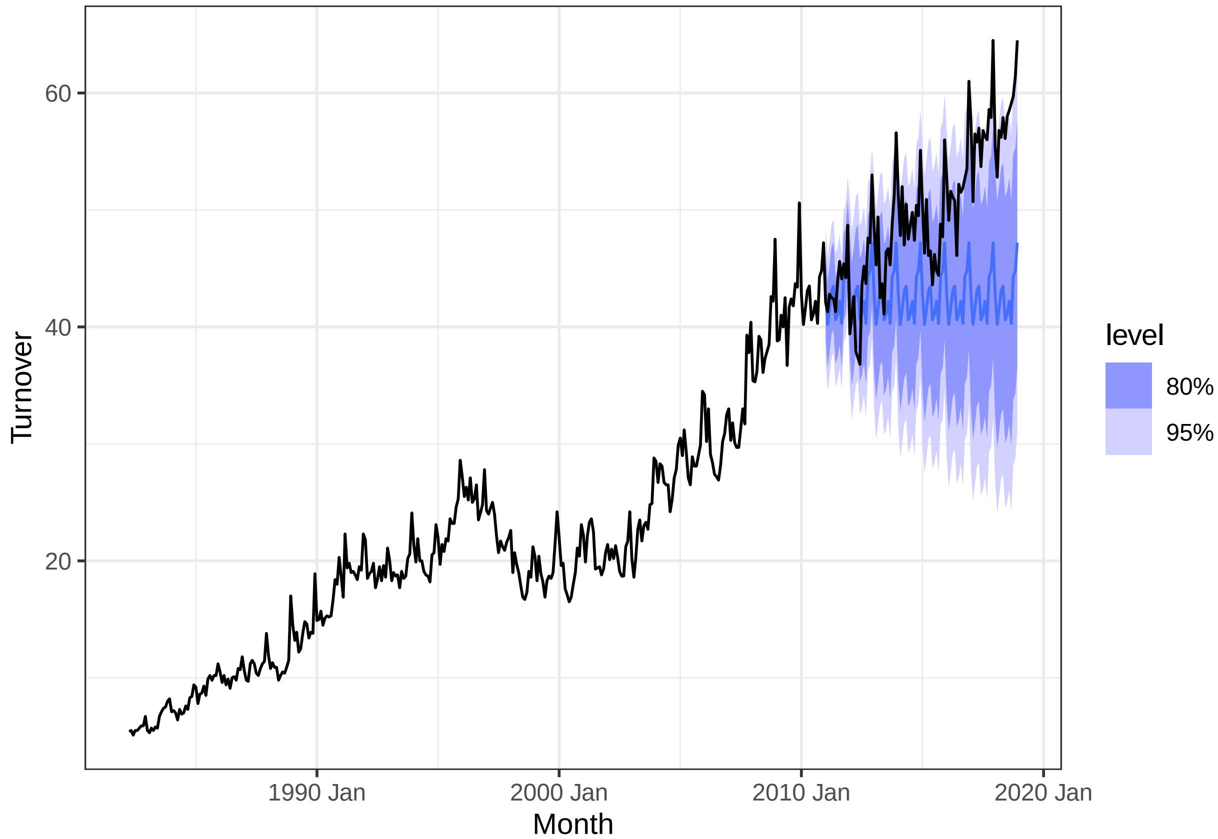
The ACF plot shows that there is some autocorrelation in the data. The residuals are not centered around 0 and seems to be right skewed. They also do not have constant variation. The residuals do not appear to be uncorrelated and normally distributed

e. Produce forecasts for the test 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)+theme_bw()
```



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

```
fit |> fabletools::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 Tasmania Cafes, ~ SNAIV~ Trai~  1.33  2.90  2.22  6.31  10.7     1     1 0.800
```

```
fc |> fabletools::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~ Tasm~ Cafes, ~ Test   7.12  9.13  7.58  13.2  14.4  3.42  3.15 0.863
```

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

The accuracy measures are highly sensitive to the amount of training data used, which can also depend on how you split the data you used. Including more or less data in training will change the forecast, and in turn change the accuracy measurements.