Exponential Smoothing HW

Chris Demgen, Partner: Kelsey Brown

2022-03-28

```
rm(list=ls()) #removes all objects from memory
library(wooldridge)
library(stargazer)
library(lmtest)
library(car)
library(tidyverse)
library(tsibble)
library(dplyr)
library(ggplot2)
library(fpp3)
library(lubridate)
library(USgas)
library(latex2exp)
library(seasonal)
library(latexpdf)
```

Question 5 (except part f)

Data set global_economy contains the annual Exports from many countries. Select one country to analyse.

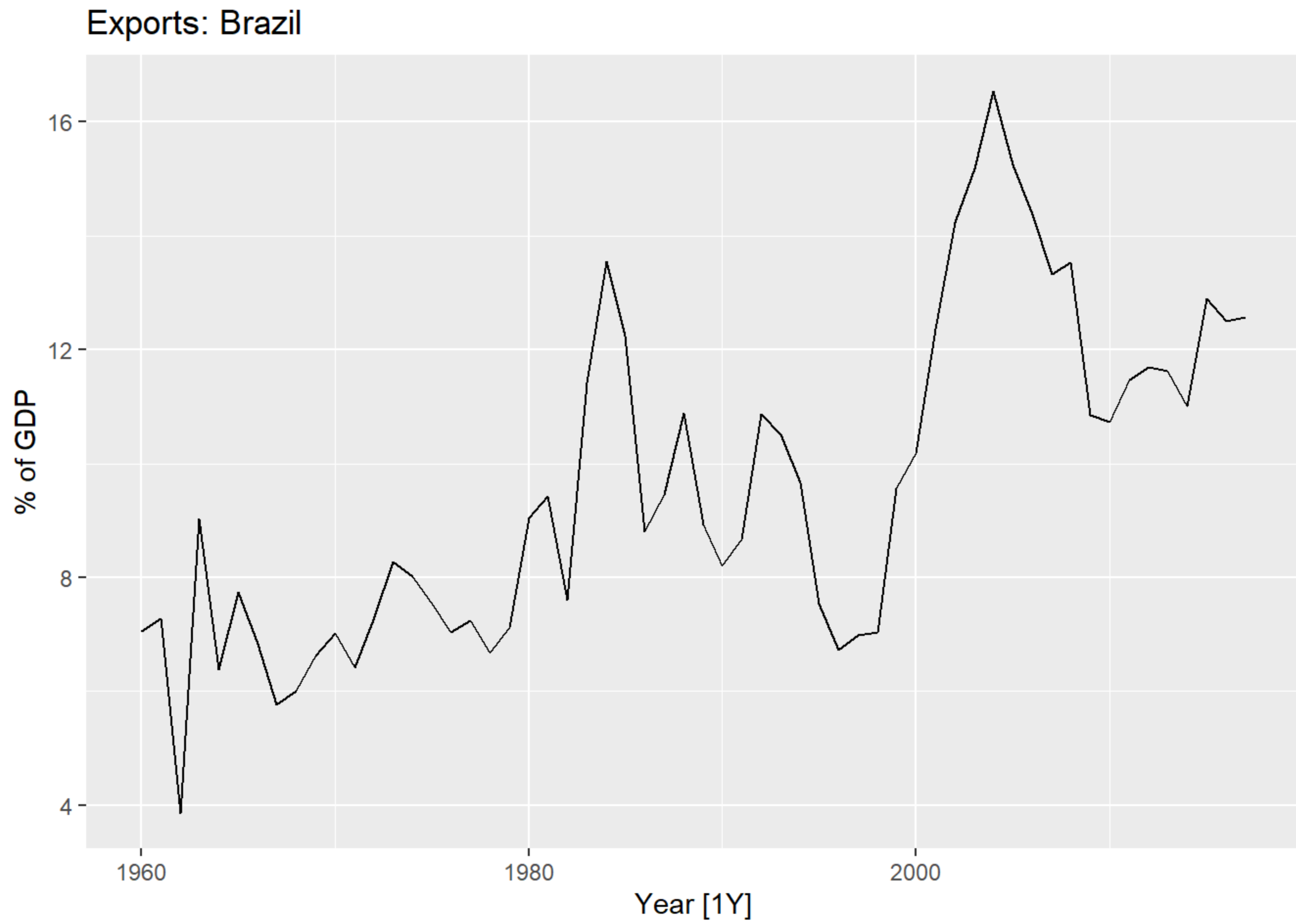
```
help(global_economy)

# Create global to look further into the data
global <- global_economy
head(global)
```

```
# A tsibble: 6 x 9 [1Y]
# Key:       Country [1]
  Country     Code   Year        GDP Growth   CPI Imports Exports Population
  <fct>       <fct> <dbl>      <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
1 Afghanistan AFG    1960  537777811.     NA    NA    7.02    4.13    8996351
2 Afghanistan AFG    1961  548888896.     NA    NA    8.10    4.45    9166764
3 Afghanistan AFG    1962  546666678.     NA    NA    9.35    4.88    9345868
4 Afghanistan AFG    1963  751111191.     NA    NA   16.9     9.17    9533954
5 Afghanistan AFG    1964  800000044.     NA    NA   18.1     8.89    9731361
6 Afghanistan AFG    1965 1006666638.     NA    NA   21.4    11.3     9938414
```

```
# Select Brazil to analyze their annual exports
brazil_economy <- tsibbledata::global_economy %>%
  filter(Country == "Brazil")
```

a. Plot the Exports series and discuss the main features of the data.

```
brazil_economy %>%
  autoplot(Exports) +
  labs(y = "% of GDP", title = "Exports: Brazil")
```



The Export series for Brazil shows an upwards trend. The series does not appear to have seasonality. The exports reach highs and lows, but there is not necessarily a specific cyclic pattern.

b. Use an ETS(A,N,N) model to forecast the series, and plot the forecasts.

```
# Define the training and test data
brazil_train <- brazil_economy %>%
  filter(Year < 2012)

brazil_test <- brazil_economy %>%
  filter(Year >= 2012)
```

```
# Fit the series using ETS(ANN) model
fit <- brazil_train %>%
  model(ANN=ETS(Exports ~ error("A") + trend("N") + season("N")))
report(fit)
```

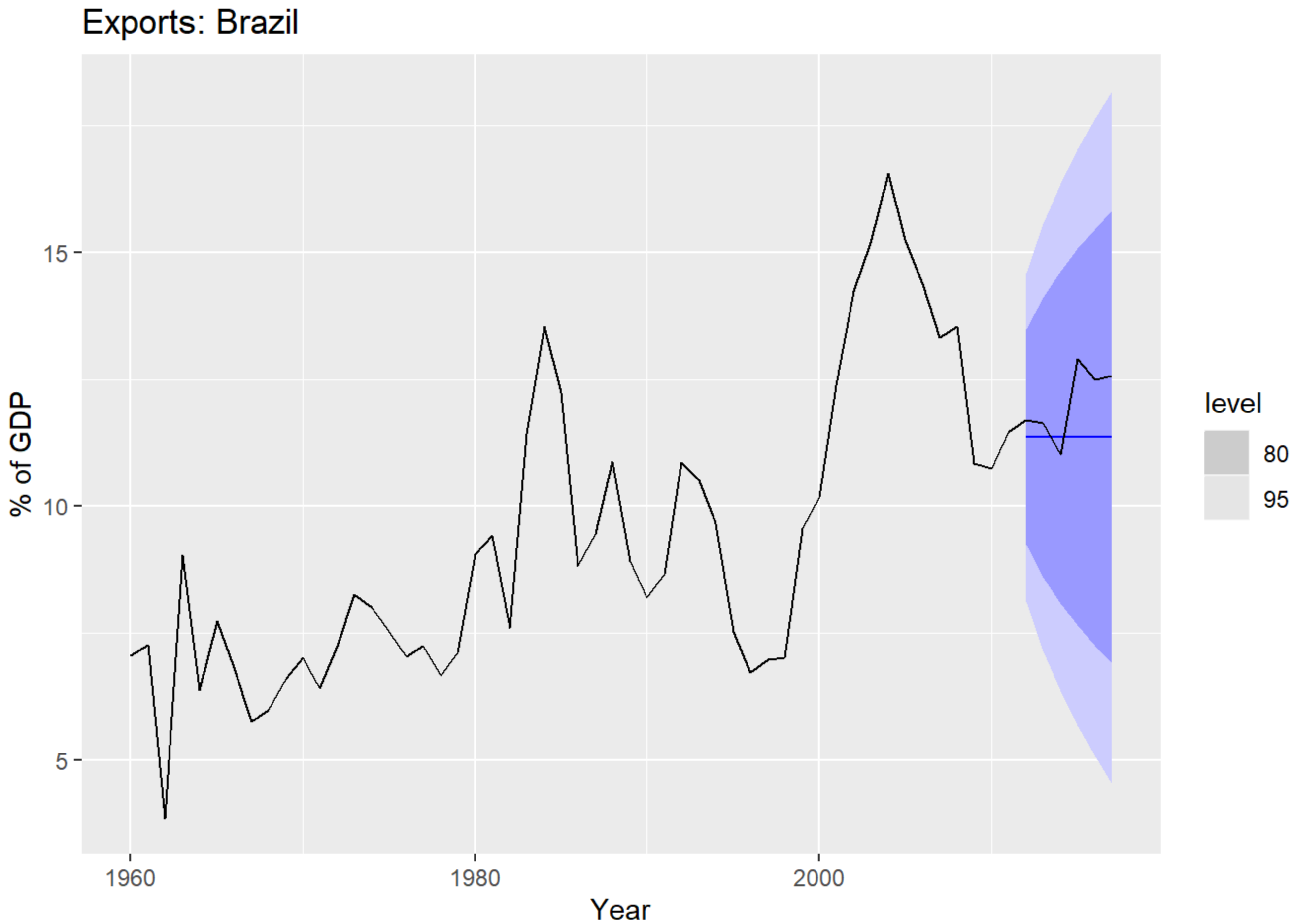
```
Series: Exports
Model: ETS(A,N,N)
  Smoothing parameters:
    alpha = 0.8306747

  Initial states:
     l[0]
 7.020174

  sigma^2:  2.7228

     AIC     AICc      BIC
261.5110 262.0110 267.3647
```

```
components(fit) %>%
  left_join(fitted(fit), by = c("Country", ".model", "Year"))
```

```
# A dable: 53 x 7 [1Y]
# Key:     Country, .model [1]
# :        Exports = lag(level, 1) + remainder
   Country .model  Year Exports level remainder .fitted
   <fct>   <chr>  <dbl>   <dbl> <dbl>     <dbl>   <dbl>
 1 Brazil  ANN     1959   NA     7.02   NA        NA
 2 Brazil  ANN     1960    7.06  7.05    0.0389    7.02
 3 Brazil  ANN     1961    7.28  7.24    0.227     7.05
 4 Brazil  ANN     1962    3.87  4.44   -3.37      7.24
 5 Brazil  ANN     1963    9.04  8.26    4.60      4.44
 6 Brazil  ANN     1964    6.39  6.70   -1.87      8.26
 7 Brazil  ANN     1965    7.74  7.56    1.03      6.70
 8 Brazil  ANN     1966    6.82  6.95   -0.736     7.56
 9 Brazil  ANN     1967    5.77  5.97   -1.18      6.95
10 Brazil  ANN     1968    6.00  6.00    0.0317    5.97
# ... with 43 more rows
```

Alpha = 0.83 indicates that there is a large adjustment in the estimated level, lt, each period. In this case more weight is given to more recent observations.

```
# Forecast and plot the model
fit %>%
  forecast(h=6) %>%
  autoplot(brazil_economy) +
  labs(y = "% of GDP", title = "Exports: Brazil")
```



The simple exponential smoothing model has created a flat forecast.

c. Compute the RMSE values for the training data.

```
# Report RMSE values
accuracy(fit)
```

```
# A tibble: 1 x 11
  Country .model .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE    ACF1
  <fct>   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
1 Brazil  ANN    Training 0.100  1.62  1.22 -1.14  13.9 0.970 0.977 0.00227
```

The RMSE value is 1.62 and the RMSSE value is 0.978.

d. Compare the results to those from an ETS(A,A,N) model. (Remember that the trended model is using one more parameter than the simpler model.) Discuss the merits of the two forecasting methods for this data set.

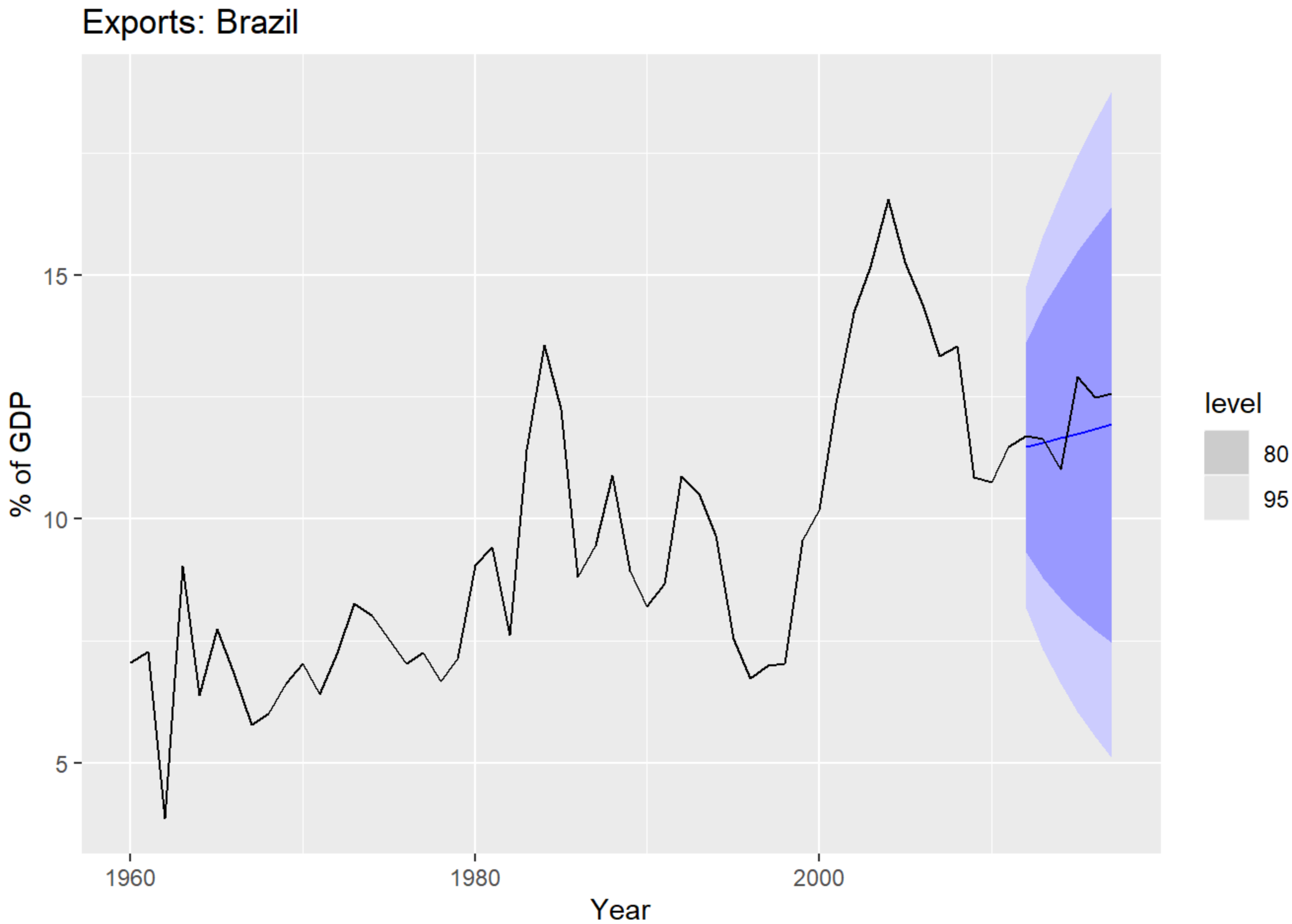
```
# Forecast the ETS(AAN) model
fit2 <- brazil_train %>%
  model(AAN=ETS(Exports ~ error("A") + trend("A") + season("N")))
report(fit2)
```

```
Series: Exports
Model: ETS(A,A,N)
  Smoothing parameters:
    alpha = 0.8141193
    beta  = 0.0001000044

  Initial states:
     l[0]     b[0]
 6.950245 0.092892

  sigma^2:  2.8257

     AIC     AICc      BIC
265.3180 266.6223 275.0742
```

```
components(fit2) %>%
  left_join(fitted(fit2), by = c("Country", ".model", "Year"))
```

```
# A dable: 53 x 8 [1Y]
# Key:     Country, .model [1]
# :        Exports = lag(level, 1) + lag(slope, 1) + remainder
   Country .model  Year Exports level  slope remainder .fitted
   <fct>   <chr>  <dbl>   <dbl> <dbl>  <dbl>     <dbl>   <dbl>
 1 Brazil  AAN     1959   NA     6.95 0.0929   NA        NA
 2 Brazil  AAN     1960    7.06  7.06 0.0929    0.0159    7.04
 3 Brazil  AAN     1961    7.28  7.26 0.0929    0.130     7.15
 4 Brazil  AAN     1962    3.87  4.52 0.0926   -3.48      7.35
 5 Brazil  AAN     1963    9.04  8.21 0.0930    4.43      4.61
 6 Brazil  AAN     1964    6.39  6.74 0.0928   -1.92      8.31
 7 Brazil  AAN     1965    7.74  7.57 0.0929    0.900     6.84
 8 Brazil  AAN     1966    6.82  6.98 0.0928   -0.837     7.66
 9 Brazil  AAN     1967    5.77  6.01 0.0927   -1.30      7.07
10 Brazil  AAN     1968    6.00  6.02 0.0927   -0.104     6.10
# ... with 43 more rows
```

```
fit2 %>%
  forecast(h=6) %>%
  autoplot(brazil_economy) +
  labs(y = "% of GDP", title = "Exports: Brazil")
```




```
# Report RMSE
accuracy(fit2)
```

```
# A tibble: 1 x 11
  Country .model .type         ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
  <fct>   <chr>  <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
1 Brazil  AAN    Training -0.00961  1.62  1.23 -2.44  14.0 0.975 0.976 0.0179
```

AAN is different from ANN because it is adding the parameter beta that is accounting for the trend. The AAN model does a slightly better job of fitting the data by taking account for the trend. However, the RMSE between the two models is only different by roughly 0.003.

e. Compare the forecasts from both methods. Which do you think is best?

```
# Compare the accuracy between forecasts
fc1 <- fit %>%
  forecast(h=6)

fc2 <- fit2 %>%
  forecast(h=6)

accuracy(fc1, brazil_test)
```

```
# A tibble: 1 x 11
  .model Country .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>  <fct>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 ANN    Brazil  Test  0.692 0.952 0.808  5.46  6.51   NaN   NaN 0.120
```

```
accuracy(fc2, brazil_test)
```

```
# A tibble: 1 x 11
  .model Country .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE    ACF1
  <chr>  <fct>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
1 AAN    Brazil  Test  0.353 0.664 0.566  2.69  4.62   NaN   NaN -0.0811
```

The AAN model forecasts th series better. Not only does it forecast the data better visually, but it also has a significantly lower RMSE (.66 < .95) through incorporating the trend.

Question 10

Compute the total domestic overnight trips across Australia from the tourism dataset.

```
help(tourism)

# Dataset with total domestic overnight trips across Australia
tottrips <- tourism %>%
  summarise(Trips = sum(Trips))

head(tottrips)
```

```
# A tsibble: 6 x 2 [1Q]
  Quarter  Trips
    <qtr>  <dbl>
1 1998 Q1 23182.
2 1998 Q2 20323.
3 1998 Q3 19827.
4 1998 Q4 20830.
5 1999 Q1 22087.
6 1999 Q2 21458.
```

```
# Define the training and test data
tottrips_train <- tottrips %>%
   filter(year(Quarter) < 2016)

tottrips_test <- tottrips %>%
filter(year(Quarter) >= 2016)

head(tottrips_test)
```

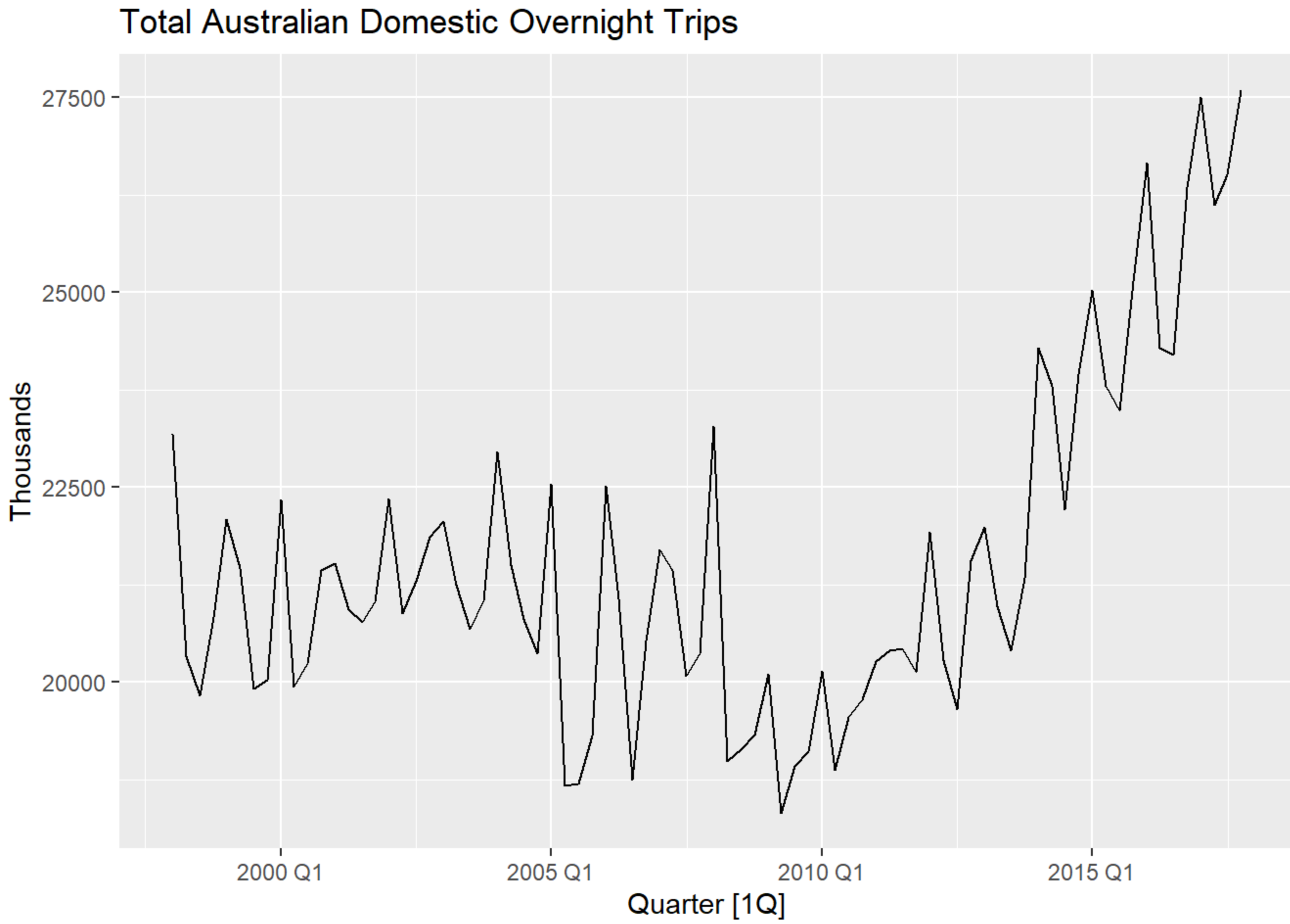
```
# A tsibble: 6 x 2 [1Q]
  Quarter  Trips
    <qtr>  <dbl>
1 2016 Q1 26661.
2 2016 Q2 24285.
3 2016 Q3 24191.
4 2016 Q4 26348.
5 2017 Q1 27496.
6 2017 Q2 26114.
```

```
head(tottrips_train)
```

```
# A tsibble: 6 x 2 [1Q]
  Quarter  Trips
    <qtr>  <dbl>
1 1998 Q1 23182.
2 1998 Q2 20323.
3 1998 Q3 19827.
4 1998 Q4 20830.
5 1999 Q1 22087.
6 1999 Q2 21458.
```

a. Plot the data and describe the main features of the series.

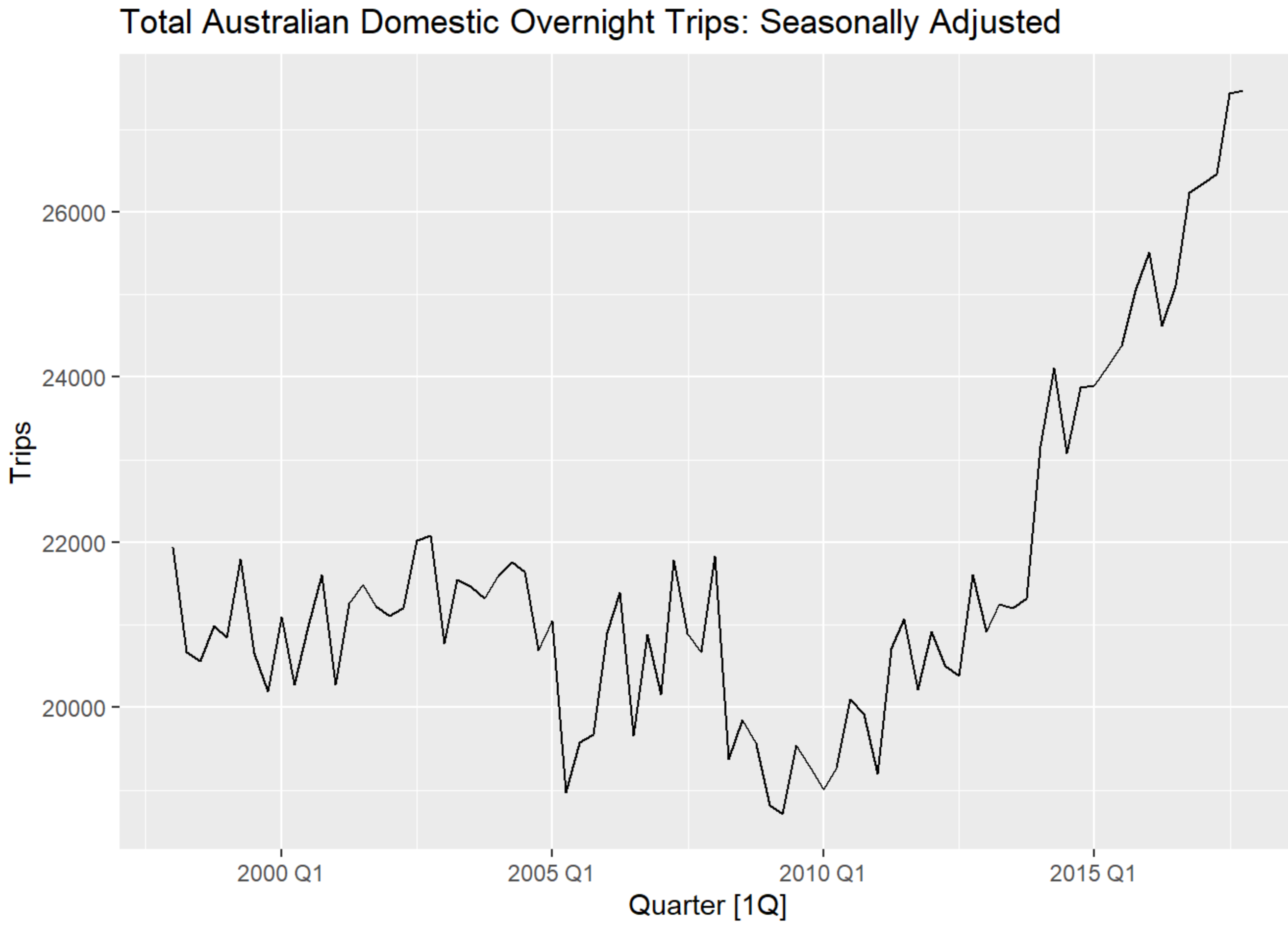
```
# Autoplot the series
autoplot(tottrips) + labs(title = "Total Australian Domestic Overnight Trips", y = "Thousands")
```



There is an upwards trend in total Australian domestic overnight trips. There is strong seasonality. There is no cyclicity present in the data. The Trips seem to decrease from roughly 2008 through 2012.

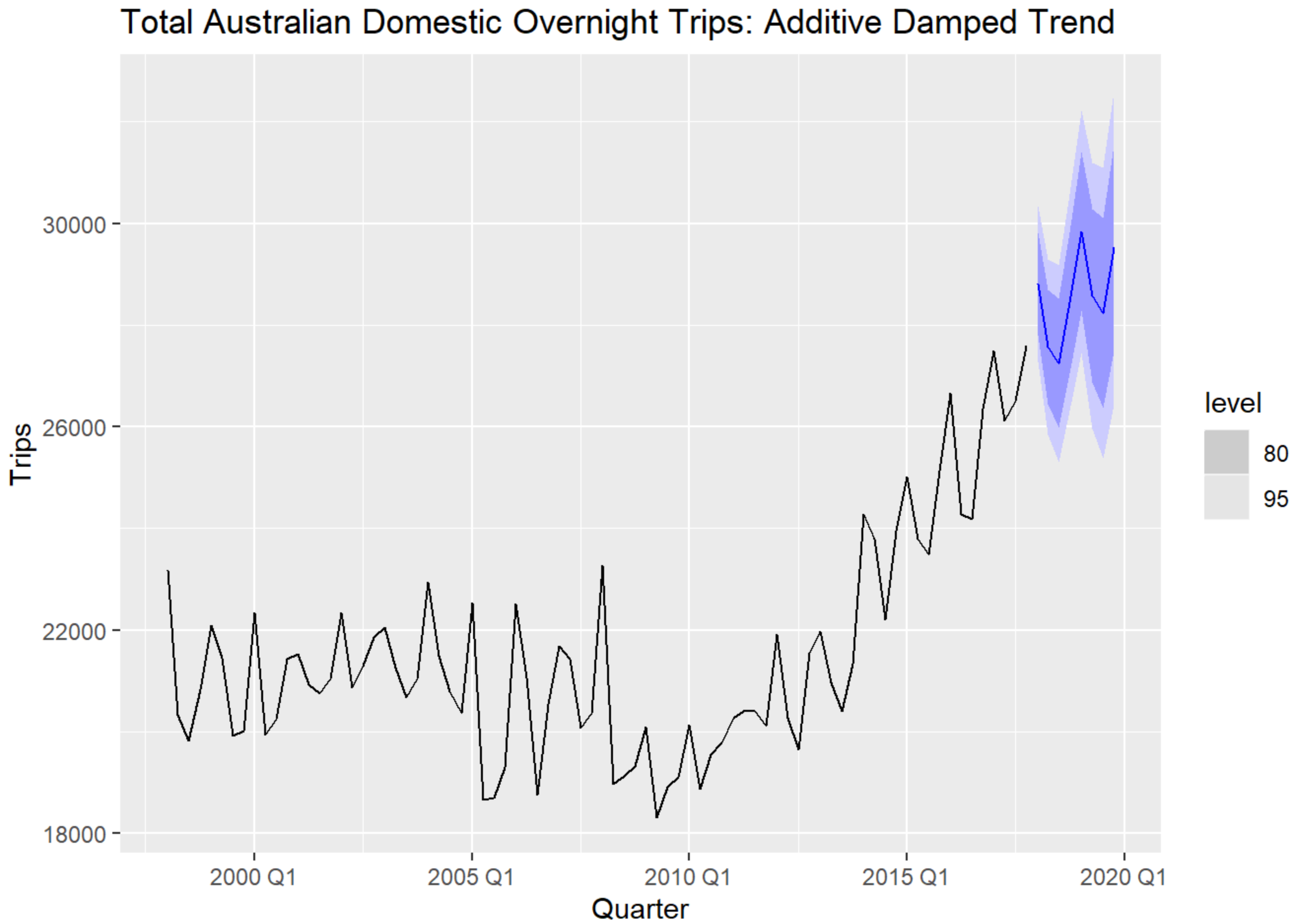
b. Decompose the series using STL and obtain the seasonally adjusted data.

```
# Decompose the series using STL
dcmp <- tottrips %>%
  model(STL(Trips)) %>%
  components()

# obtain the seasonally adjusted data
dcmp %>%
  as_tsibble() %>%
  autoplot(season_adjust) + labs(title = "Total Australian Domestic Overnight Trips: Seasonally
Adjusted", y = "Trips")
```



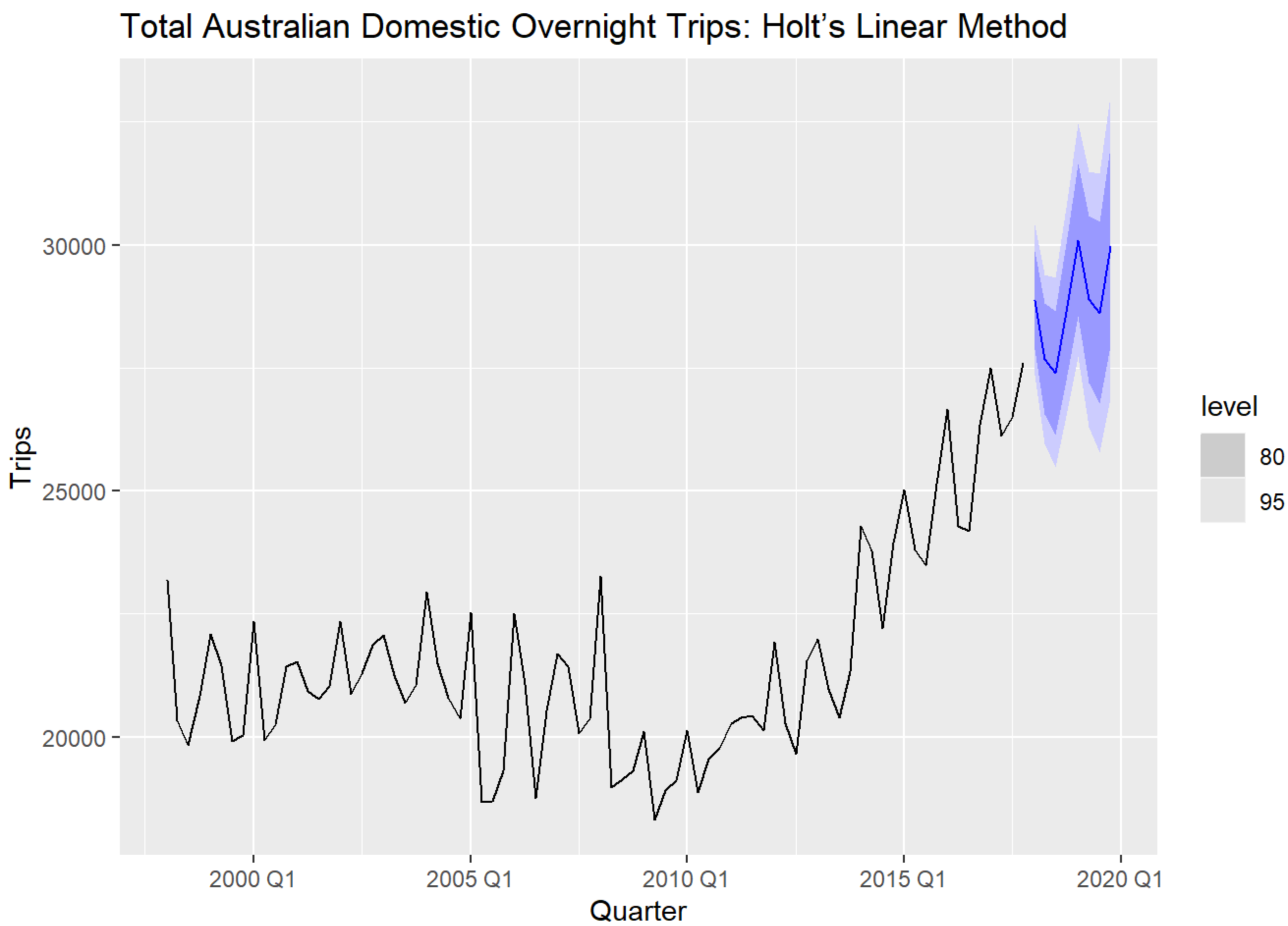
c. Forecast the next two years of the series using an additive damped trend method applied to the seasonally adjusted data. (This can be specified using decomposition_model().)

```
# Forecast 2 years using the additive damped trend method
stletsdamped <- decomposition_model(
  STL(Trips),
  ETS(season_adjust ~ error("A") + trend("Ad") + season("N"))
)
tottrips %>%
  model(dcmp_AAdN = stletsdamped) %>%
  forecast(h = "2 years") %>%
  autoplot(tottrips) + labs(title = "Total Australian Domestic Overnight Trips: Additive Damped
Trend", y = "Trips")
```



d. Forecast the next two years of the series using an appropriate model for Holt's linear method applied to the seasonally adjusted data (as before but without damped trend).

```
# Forecast 2 years using Holt's linear method applied to the seasonally adjusted data
stlets <- decomposition_model(
  STL(Trips),
  holt = ETS(season_adjust ~ error("A") + trend("A") + season("N"))
)
tottrips %>%
  model(dcmp_AAN = stlets) %>%
  forecast(h = "2 years") %>%
  autoplot(tottrips) + labs(title = "Total Australian Domestic Overnight Trips: Holt's Linear Me
thod", y = "Trips")
```

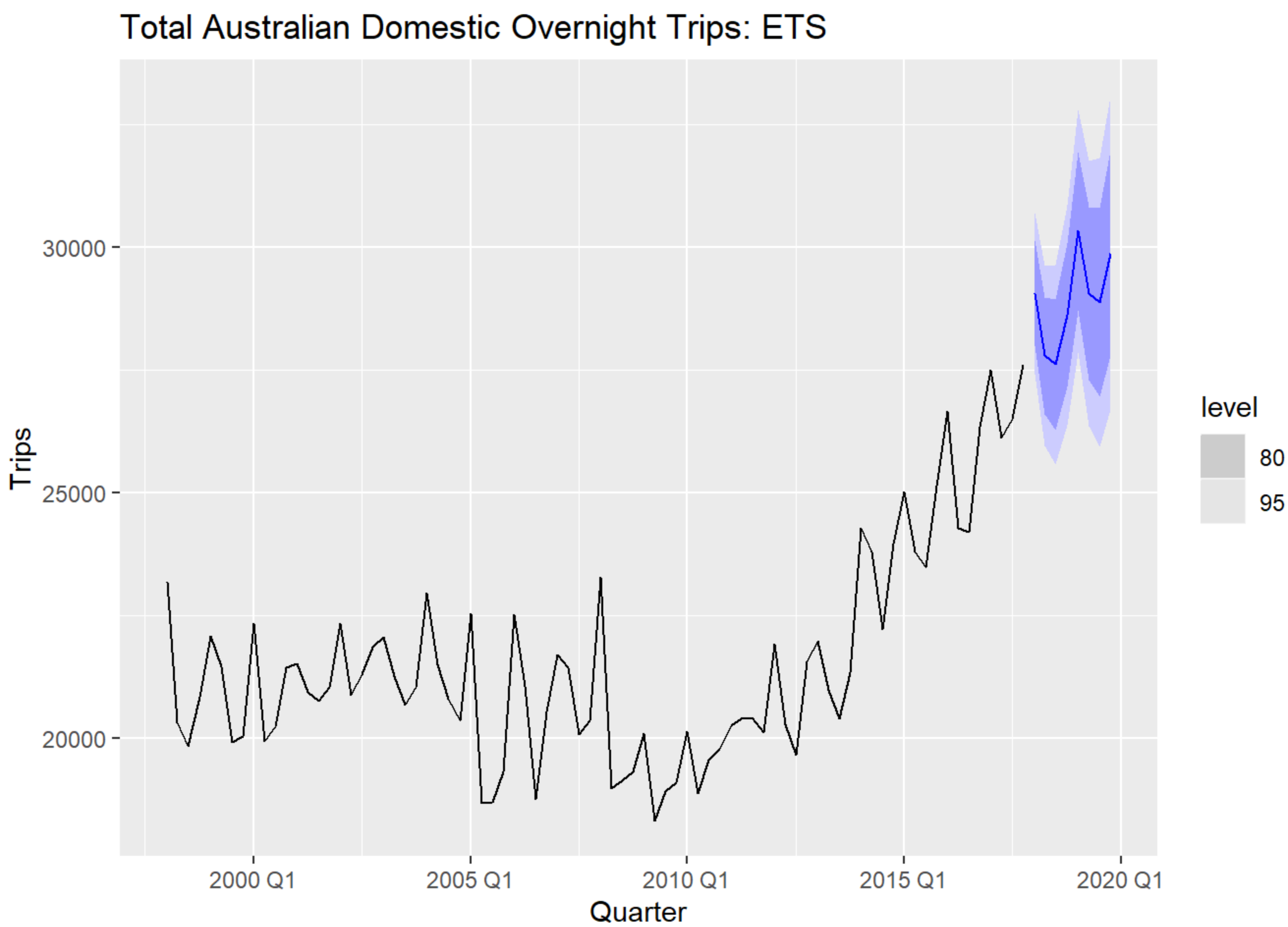


e. Now use ETS() to choose a seasonal model for the data.

```
# Use ETS to choose a seasonal model
tottrips %>%
  model(ETS(Trips)) %>%
  report(tottrips)
```

```
Series: Trips
Model: ETS(A,A,A)
  Smoothing parameters:
    alpha = 0.4495675
    beta  = 0.04450178
    gamma = 0.0001000075

  Initial states:
     l[0]      b[0]      s[0]     s[-1]     s[-2]    s[-3]
 21689.64 -58.46946 -125.8548 -816.3416 -324.5553 1266.752

  sigma^2:  699901.4

     AIC     AICc      BIC
1436.829 1439.400 1458.267
```

```
# Forecast using the ETS model
tottrips %>%
  model(ETS(Trips)) %>%
  forecast(h = "2 years") %>%
  autoplot(tottrips)+ labs(title = "Total Australian Domestic Overnight Trips: ETS", y = "Trip
s")
```



f. Compare the RMSE of the ETS model with the RMSE of the models you obtained using STL decompositions. Which gives the better in-sample fits?

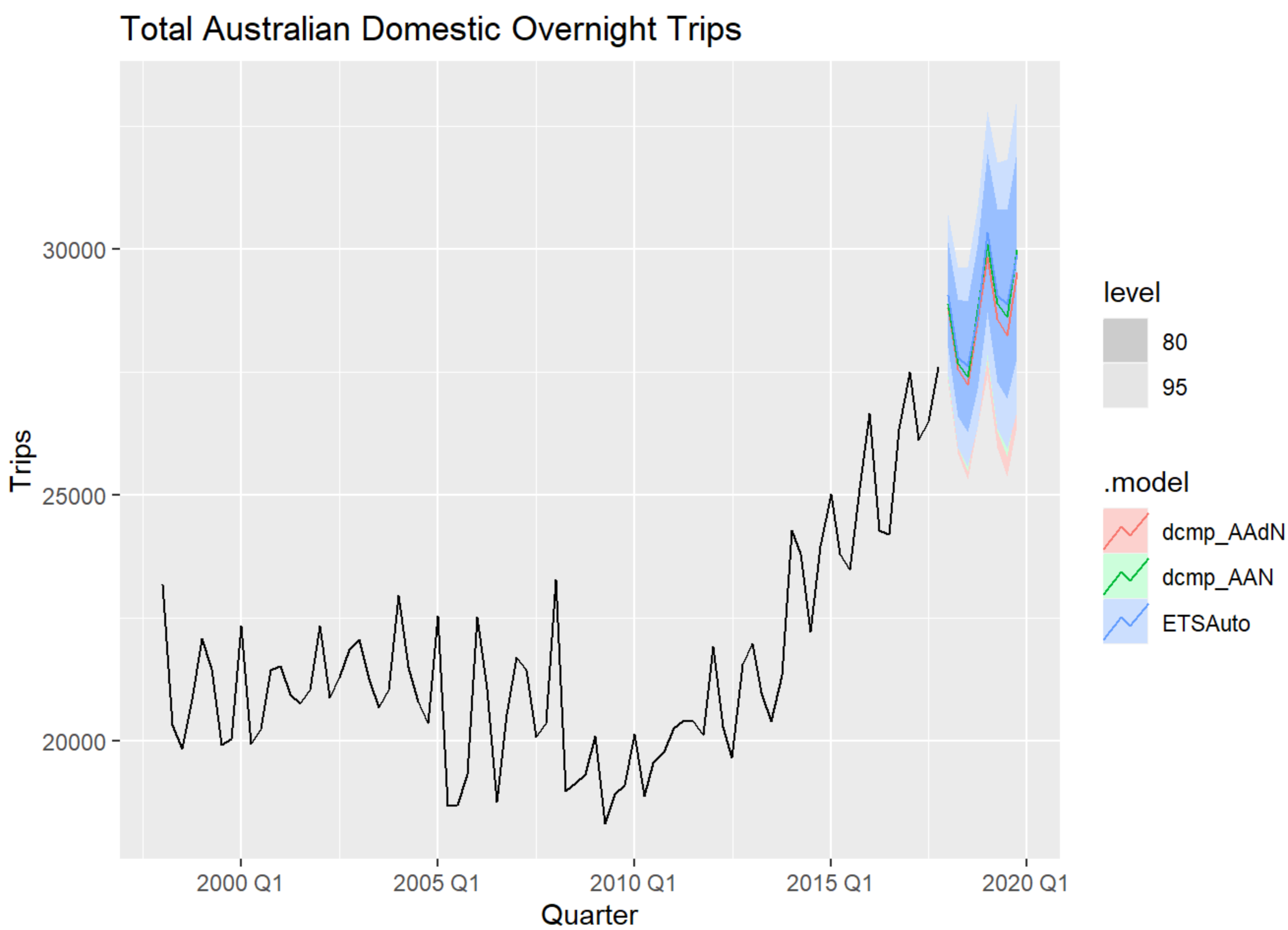
```
# Accuracy of all three fitted models
tottrips %>%
  model(dcmp_AAdN = stletsdamped,
        dcmp_AAN = stlets,
        ETSAuto = ETS(Trips)
        ) %>%
accuracy()
```

```
# A tibble: 3 x 10
  .model    .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
  <chr>     <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
1 dcmp_AAdN Training 103.   763.  576. 0.367  2.72 0.607 0.629 -0.0174
2 dcmp_AAN  Training  99.7  763.  574. 0.359  2.71 0.604 0.628 -0.0182
3 ETSAuto   Training 105.   794.  604. 0.379  2.86 0.636 0.653 -0.00151
```

dcmp_AAN gives the best in-sample fit because it has the lowest values in all of the accuracy measures. The ETS model gives the worst in-sample fit of the models.

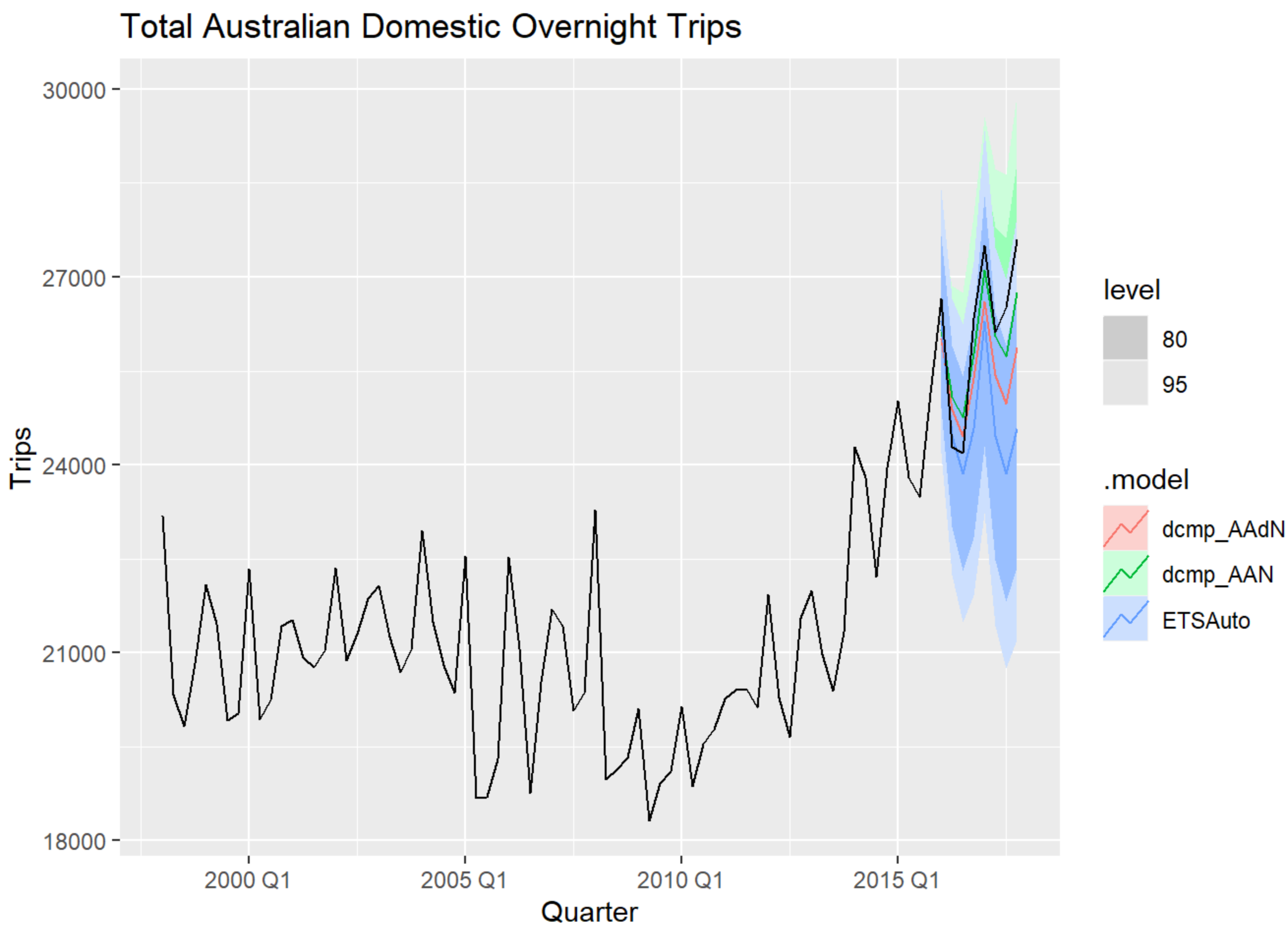
g. Compare the forecasts from the three approaches? Which seems most reasonable?

```
# Plot all three forecasts
tottrips %>%
  model(dcmp_AAdN = stletsdamped,
        dcmp_AAN = stlets,
        ETSAuto = ETS(Trips)
        ) %>%
  forecast(h = "2 years") %>%
  autoplot(tottrips) + labs(title = "Total Australian Domestic Overnight Trips", y = "Trips")
```



The forcasts look very similar, so they all seem somewhat reasonable. Since the forecasts look reasonable, I would go with dcmp_AAN so far because the fitted model it had the smallest RMSE meaning it best fits the overall data. To check if dcmp_AAN is the best model, I compare the accuracy of the forecasts with the last two years of the data.

```
#To check if dcmp_AAN is the best model, I compare the forecast accuracy with the last two years
of the data
dcmp2 <- tottrips_train %>%
  model(STL(Trips)) %>%
  components()

stletsdamped2 <- decomposition_model(
  STL(Trips),
  ETS(season_adjust ~ error("A") + trend("Ad") + season("N"))
)

stlets2 <- decomposition_model(
  STL(Trips),
  holt = ETS(season_adjust ~ error("A") + trend("A") + season("N"))
)


tottrips_fc <- tottrips_train %>%
  model(dcmp_AAdN = stletsdamped2,
        dcmp_AAN = stlets2,
        ETSAuto = ETS(Trips)
        ) %>%
  forecast(h = "2 years")

tottrips_fc %>%
  autoplot(tottrips) + labs(title = "Total Australian Domestic Overnight Trips", y = "Trips")
```



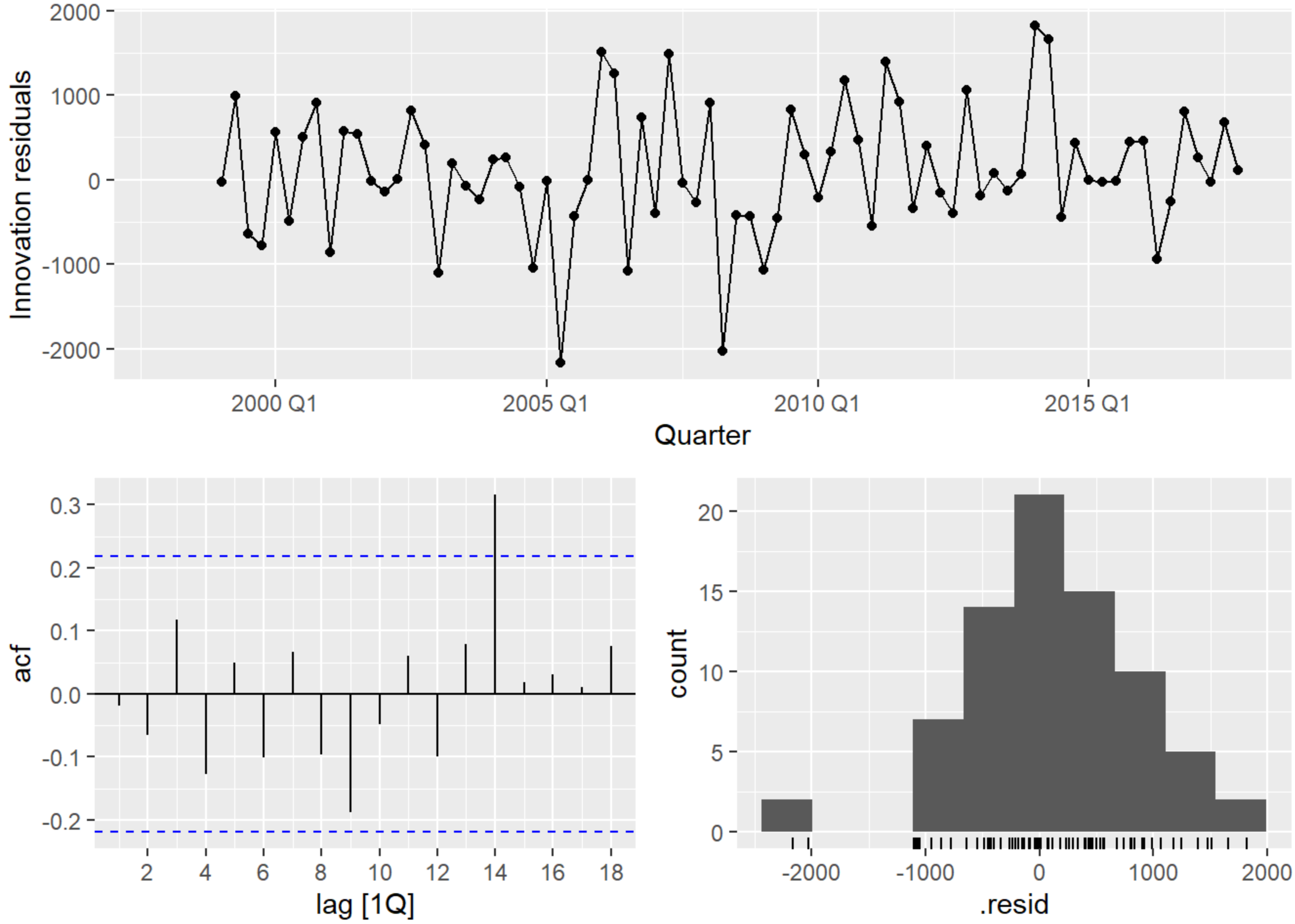

```
accuracy(tottrips_fc, tottrips_test)
```

```
# A tibble: 3 x 10
  .model    .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>     <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 dcmp_AAdN Test   693. 1017.  909. 2.54   3.42   NaN   NaN 0.436
2 dcmp_AAN  Test   223.  614.  564. 0.759  2.16   NaN   NaN 0.199
3 ETSAuto   Test  1353. 1721. 1395. 5.05   5.22   NaN   NaN 0.560
```

This confirms that dcmp_AAN is the best forecast model because it has the lowest RMSE, MAE, MPE and MAPE values.

h. Check the residuals of your preferred model.

```
# Check the residuals
tottrips_AAN <- tottrips %>% model(dcmp_AAN = stlets)
tottrips_AAN %>% gg_tsresiduals()
```



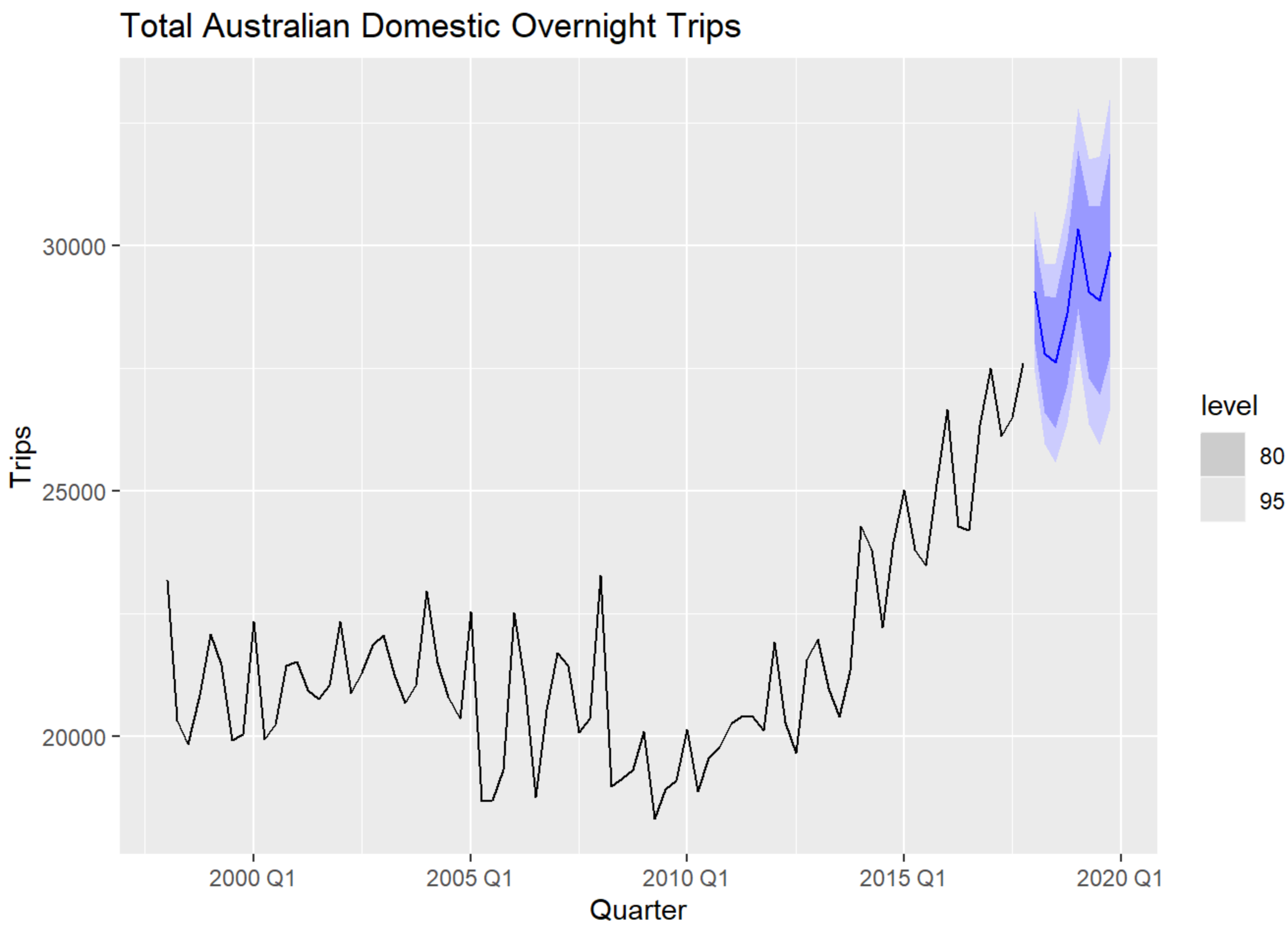
The residuals generally seem to be roughly centered around zero. There is significant positive spike in lag 14 in the ACF demonstrating that the residuals are not perfectly white noise. The residuals look normally distributed.

Question 14

a. Use ETS() to select an appropriate model for the following series: total number of trips across Australia using tourism, the closing prices for the four stocks in gafa_stock, and the lynx series in pelt. Does it always give good forecasts?

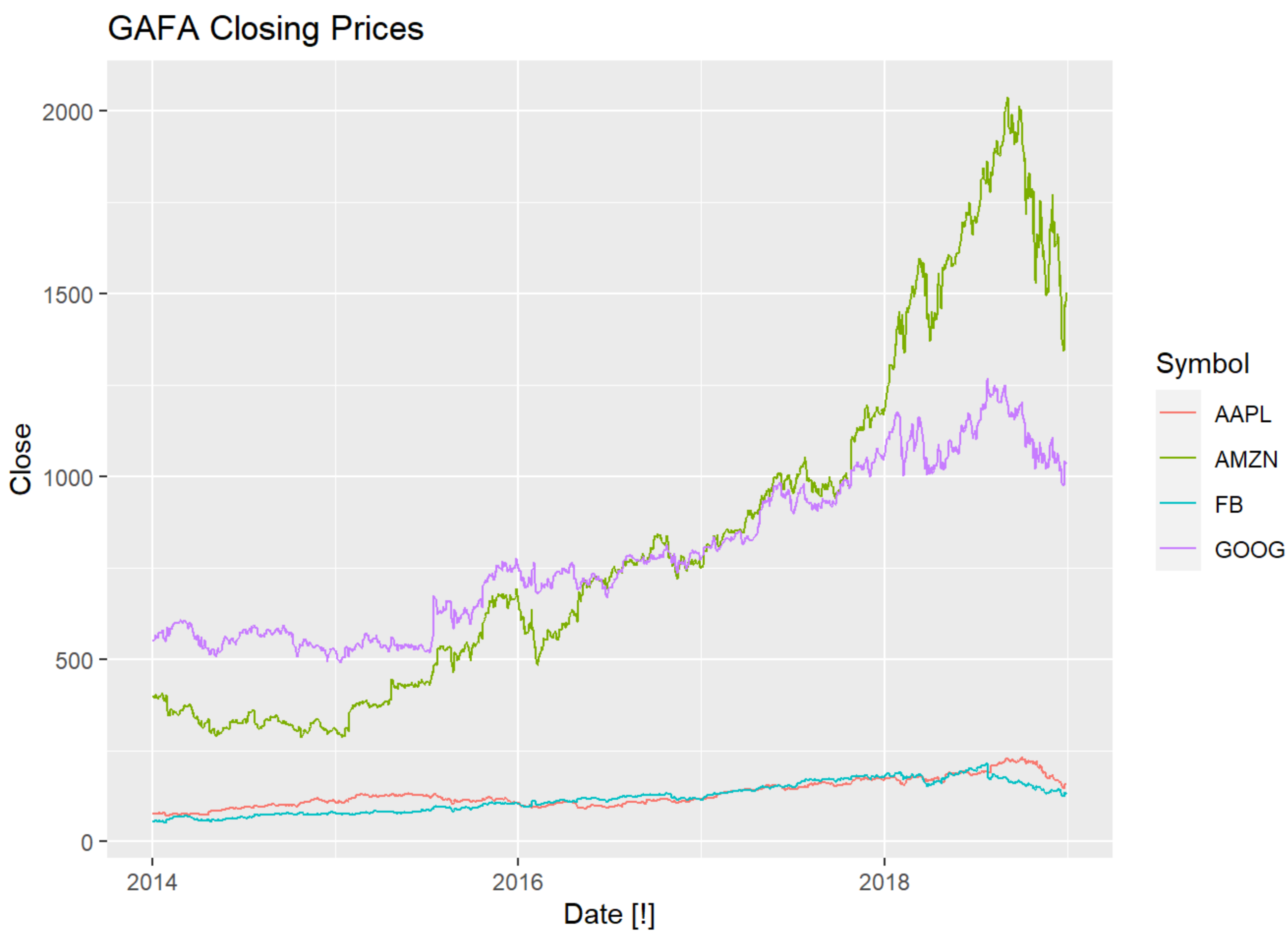
```
# Dataset with total number of trips
aus_trips <- tourism %>%
summarise (Trips = sum(Trips))

# Forecast and plot total trips
aus_trips %>%
model (ETS(Trips)) %>%
report() %>%
forecast () %>%
autoplot (aus_trips) + labs(title = "Total Australian Domestic Overnight Trips", y = "Trips")
```

```
Series: Trips
Model: ETS(A,A,A)
  Smoothing parameters:
    alpha = 0.4495675
    beta  = 0.04450178
    gamma = 0.0001000075

  Initial states:
     l[0]      b[0]      s[0]     s[-1]     s[-2]    s[-3]
 21689.64 -58.46946 -125.8548 -816.3416 -324.5553 1266.752

  sigma^2:  699901.4

     AIC     AICc      BIC
1436.829 1439.400 1458.267
```



Alpha = 0.45 means that there is similar weight being distributed between distant past observations and recent observations. There is slightly more weight being distributed to past observations. The small value for beta (0.045) means slope hardly changes over time. The small value for Gamma (0.0001) means that seasonality practically does not change over time.

```
#Dataset with closing prices for the four stocks in gafa_stock
gafa_regular <- gafa_stock %>%

group_by (Symbol) %>%

mutate (trading_day = row_number ()) %>%

ungroup () %>%

as_tsibble (index =  trading_day, regular = TRUE )

#Plot Dataset
gafa_stock %>% autoplot (Close) +
  labs(title = "GAFA Closing Prices")
```

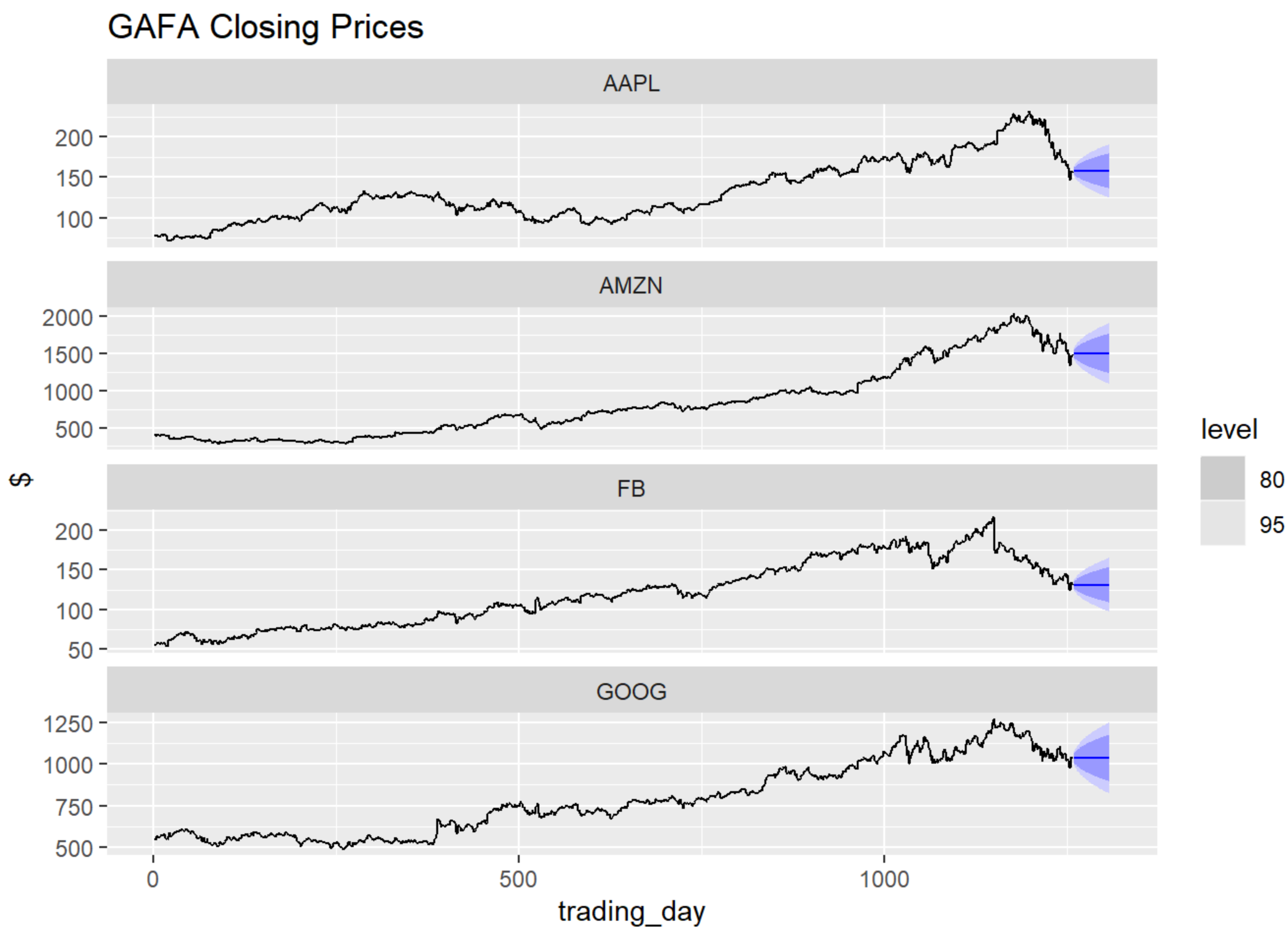


Note for grader

The slice command in the code below was not working for some unknown reason, so I talked to Prof. Kakar. She was not sure why the slice command is not working, so she told me to write a note saying not to mark me off for leaving out the slice command.

gafa_regular %>% model (ETS (Close)) %>% forecast (h = 50) %>% autoplot (gafa_regular %>% group_by_key () %>% slice ((n () - 100):n ()))

```
# Forecast the closing prices
gafa_regular %>%

model (ETS (Close)) %>%

forecast (h = 50 ) %>%

autoplot (gafa_regular %>% group_by_key()) + labs(title = "GAFA Closing Prices", y="$")
```




```
gafa_regular %>%
model (ETS (Close)) %>%
  report()
```

```
# A tibble: 4 x 10
  Symbol .model       sigma2 log_lik    AIC   AICc    BIC    MSE   AMSE    MAE
  <chr>  <chr>         <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 AAPL   ETS(Close) 0.000228  -5299. 10604. 10604. 10620.   4.39   8.96 0.0106
2 AMZN   ETS(Close) 0.000383  -7778. 15561. 15561. 15577. 359.   700.   0.0129
3 FB     ETS(Close) 0.000356  -5442. 10890. 10890. 10906.   5.82  11.3  0.0126
4 GOOG   ETS(Close) 0.000219  -7529. 15064. 15064. 15079. 144.   291.   0.0102
```

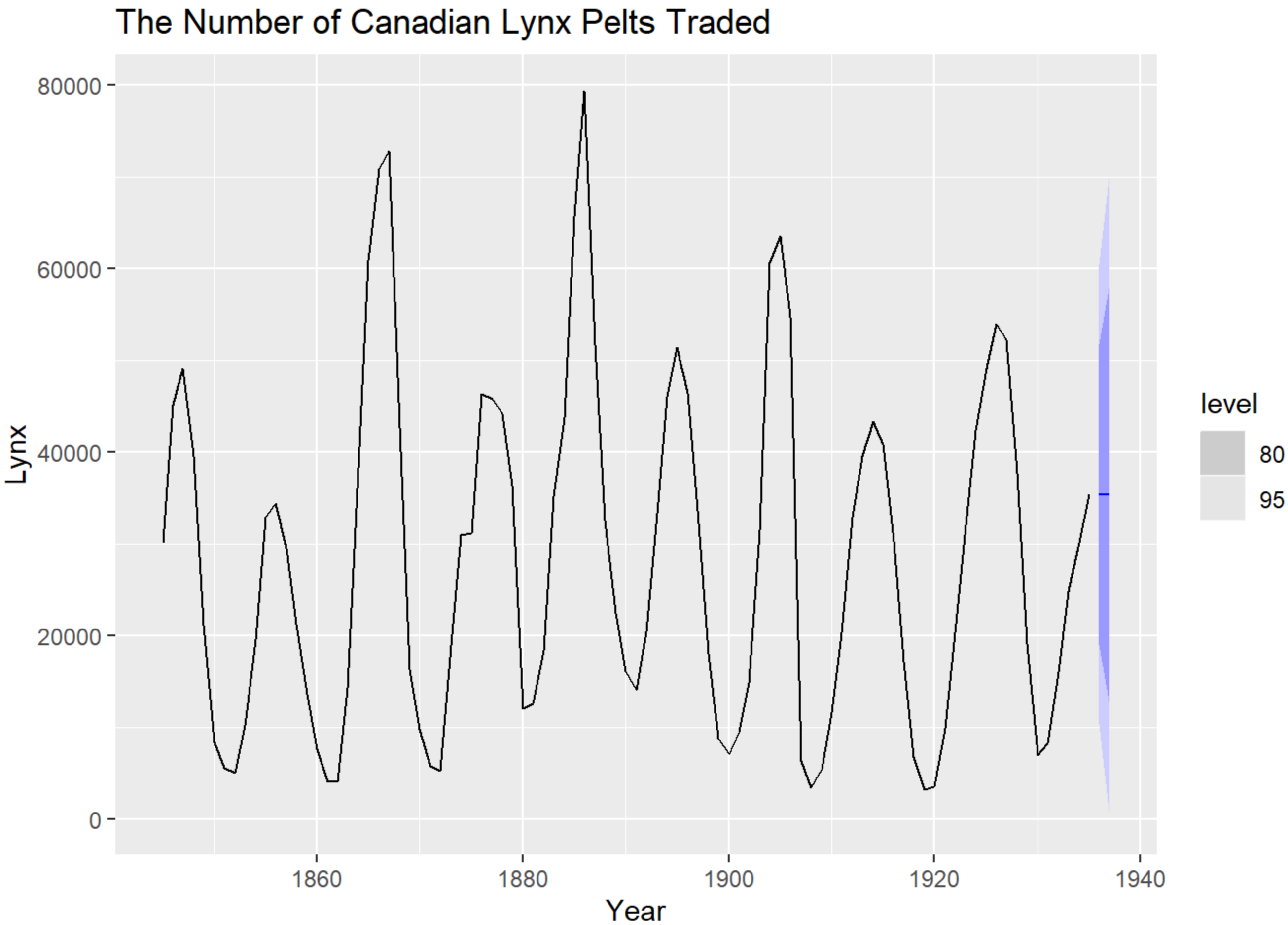
The best forecast for closing stock prices according to ETS seems to be the naive method similar to what we talked about in class.

```
help(pelt)
head(pelt)
```

```
# A tsibble: 6 x 3 [1Y]
   Year  Hare  Lynx
  <dbl> <dbl> <dbl>
1  1845 19580 30090
2  1846 19600 45150
3  1847 19610 49150
4  1848 11990 39520
5  1849 28040 21230
6  1850 58000  8420
```

```
# Forecast lynx pelts traded
pelt %>%
model(ETS(Lynx)) %>%
report() %>%
forecast() %>%
autoplot(pelt) + labs(title = "The Number of Canadian Lynx Pelts Traded")
```

```
Series: Lynx
Model: ETS(A,N,N)
  Smoothing parameters:
    alpha = 0.9998998

  Initial states:
     l[0]
 25737.12

  sigma^2:  162584145

     AIC     AICc      BIC
2134.976 2135.252 2142.509
```



Alpha rounds to 1, so this forecast is practically identical to a naive forecast. There is no trend or seasonal component being captured by the ETS model in this case.

ETS does not always give good forecasts.

b. Find an example where it does not work well. Can you figure out why?

ETS forecasts by combining a level, trend, and seasonal components to describe the time series. In this case, I am estimating the smoothing parameters by maximizing the “likelihood.” ETS does not work as well when it cannot capture the smoothing parameters alpha, beta, and gamma from the data. The ETS model works best when there is a clear trend and clear seasonality in the data. Another limitation for ETS is that ETS does not capture cyclicity in the data. This is shown by the Pelt forecast plot above where the best forecast is equal to the naive model because the model does not account for cyclicity.