

DATA624_Homework5

John Ferrara

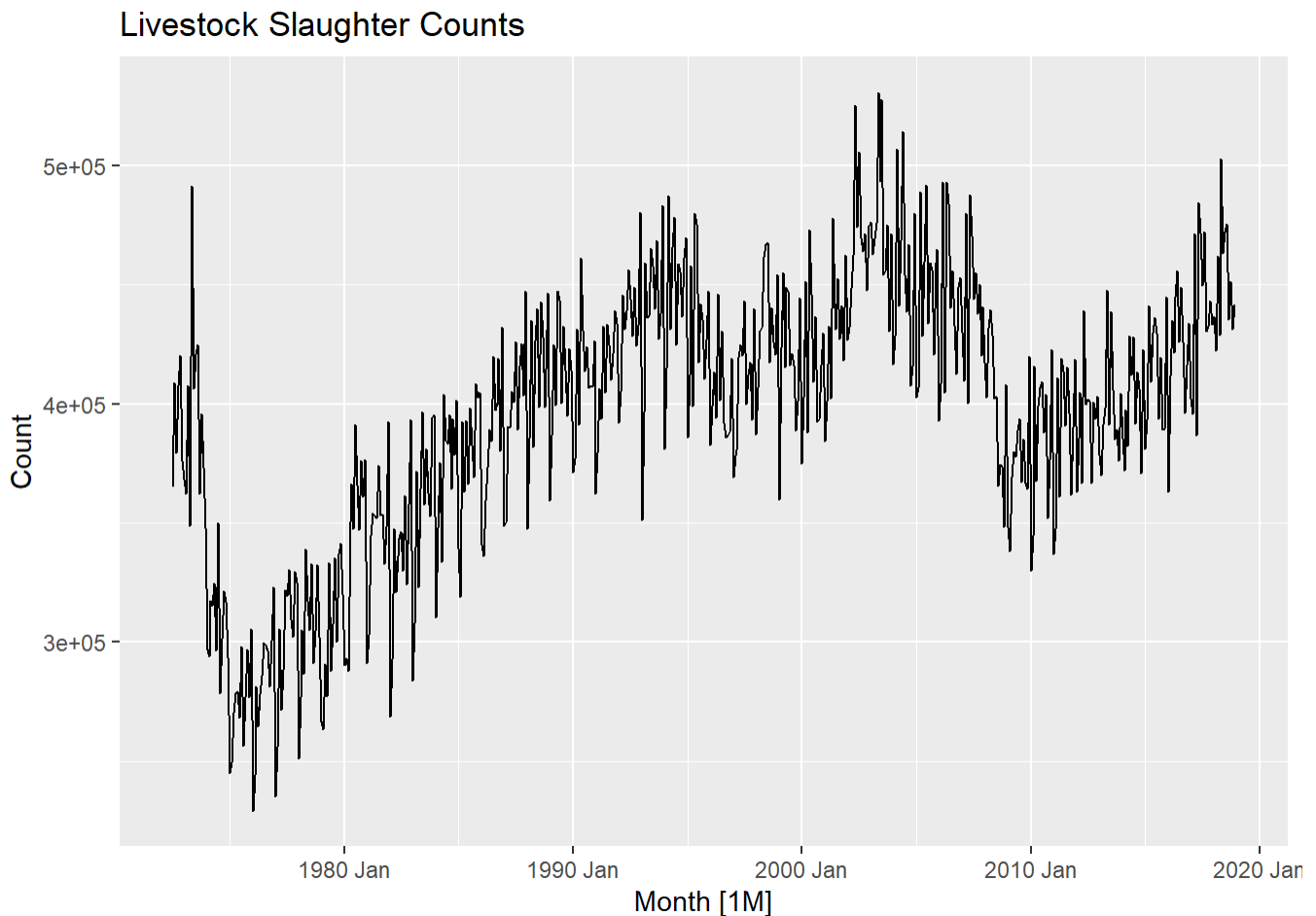
2025-03-07

1) Consider the the number of pigs slaughtered in Victoria, available in the `aus_livestock` dataset.

```
#colnames(aus_livestock)
#unique(aus_livestock$Animal)

pigs <- aus_livestock |> filter(Animal=="Pigs") |> summarise(Count = sum(Count))

## Initial Plotting
pigs |>
  autoplot(Count) +
  labs(y = "Count", title = "Livestock Slaughter Counts")
```



a) Use the `ETS()` function to estimate the equivalent model for simple exponential smoothing. Find the optimal values of α and ℓ_0 , and generate forecasts for the next four months. Using the ETS modeling method with Simple Exponential Smoothing function, the optimal values for α and ℓ_0 are 2.229786e-01 and

3.885214e+05, respectively.

```
fit <- pigs |> model(ETS(Count ~ error("A") + trend("N") + season("N")))
## Seeing optimal values
print(tidy(fit))
```

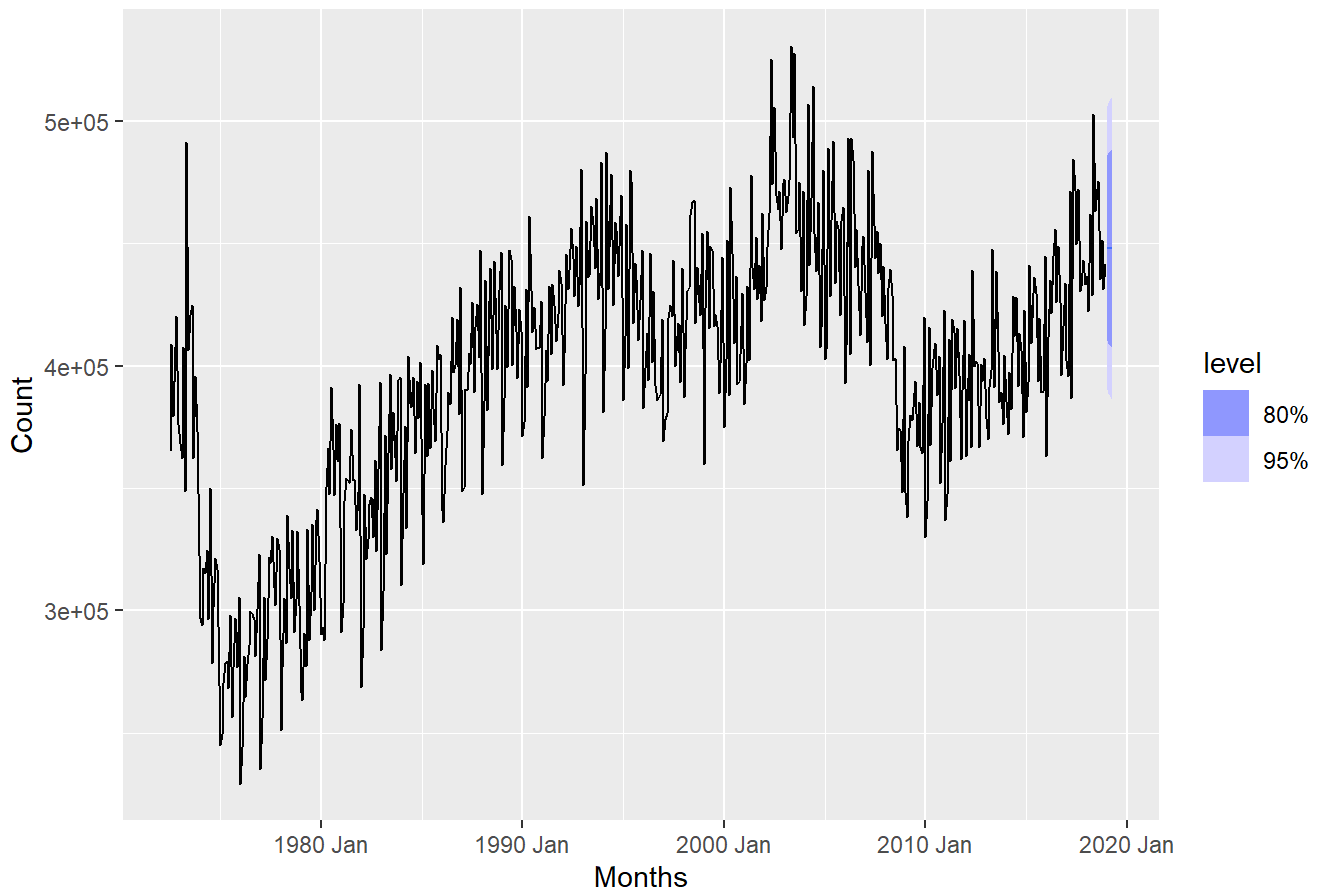
```
## # A tibble: 2 × 3
##   .model                                term      estimate
##   <chr>                                <chr>      <dbl>
## 1 "ETS(Count ~ error(\"A\") + trend(\"N\") + season(\"N\"))" alpha      0.223
## 2 "ETS(Count ~ error(\"A\") + trend(\"N\") + season(\"N\"))" l[0] 388521.
```

```
## Generating and Printing the forecast
fc <- fit |> forecast(h = 4) #4 months
print(fc)
```

```
## # A fable: 4 x 4 [1M]
## # Key:      .model [1]
##   .model                                Month
##   <chr>                                <mth>
## 1 "ETS(Count ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2019 Jan
## 2 "ETS(Count ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2019 Feb
## 3 "ETS(Count ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2019 Mar
## 4 "ETS(Count ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2019 Apr
## # i 2 more variables: Count <dist>, .mean <dbl>
```

```
#Plotting
fc |>
  autoplot(pigs) +
  labs(x="Months", y="Count",
       title = "Australia Livestock Slaughters - Pigs")
```

Australia Livestock Slaughters - Pigs



b) Compute a 95% prediction interval for the first forecast using $\hat{y} \pm 1.96s$ where s is the standard deviation of the residuals. Compare your interval with the interval produced by R. The range of the prediction interval with a 95% confidence using the first mean prediction value is 390,388.7 through 506,012.1.

```
## Taking the predicted values
first_predicted_mean_value <- fc$.mean[1] #(y^)
## Extracting Std. Dev from Counts
print(fc$Count[1])
```

```
## <distribution[1]>
## [1] N(448200, 8.7e+08)
```

```
#<distribution[1]>
#[1] N(448200, 8.7e+08)
# Manually getting s
s <- sqrt(8.7e+08)
print(s)
```

```
## [1] 29495.76
```

```
## Calculations with the formula
upper_lim <- first_predicted_mean_value + (1.96*s)
lower_lim <- first_predicted_mean_value - (1.96*s)
print(upper_lim)
```

```
## [1] 506012.1
```

```
print(lower_lim)
```

```
## [1] 390388.7
```

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

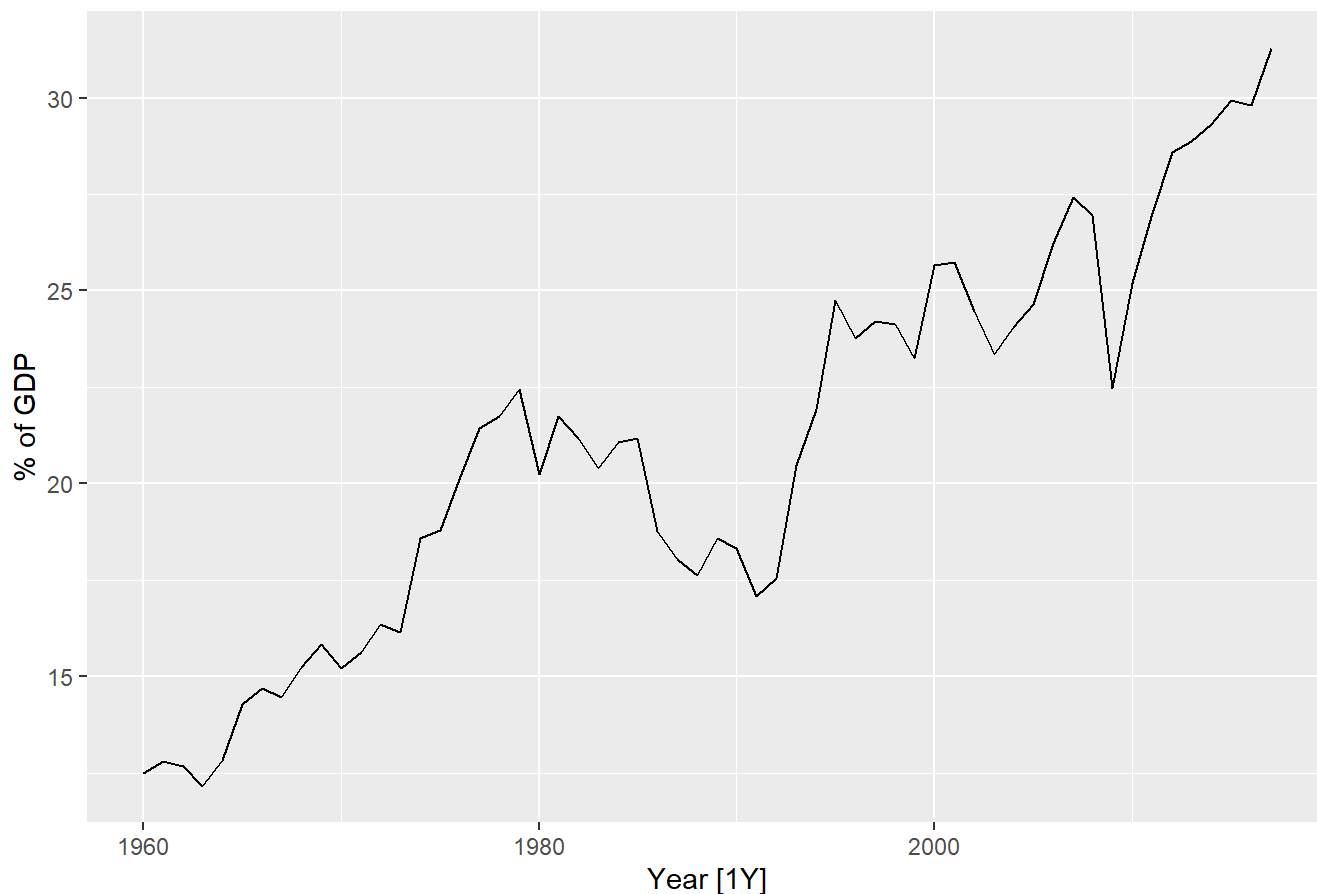
a) Plot the Exports series and discuss the main features of the data.

When looking at the exports for Italy, there is a larger trend of increasing exports over the time covered by the chart. There isn't much seasonality in the chart, however the trend is not constant, there are downturns in the data - specifically in the 1980s there was a larger decreasing trend, which was reversed in the 1990s. Similarly, in 2008 there was a large decrease in exports.

```
italy_economy <- global_economy |>
  filter(Country == "Italy")

italy_economy |>
  autoplot(Exports) +
  labs(y = "% of GDP", title = "Exports: Italy")
```

Exports: Italy



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

```
fit <- italy_economy |> model(ETS(Exports ~ error("A") + trend("N") + season("N")))
```

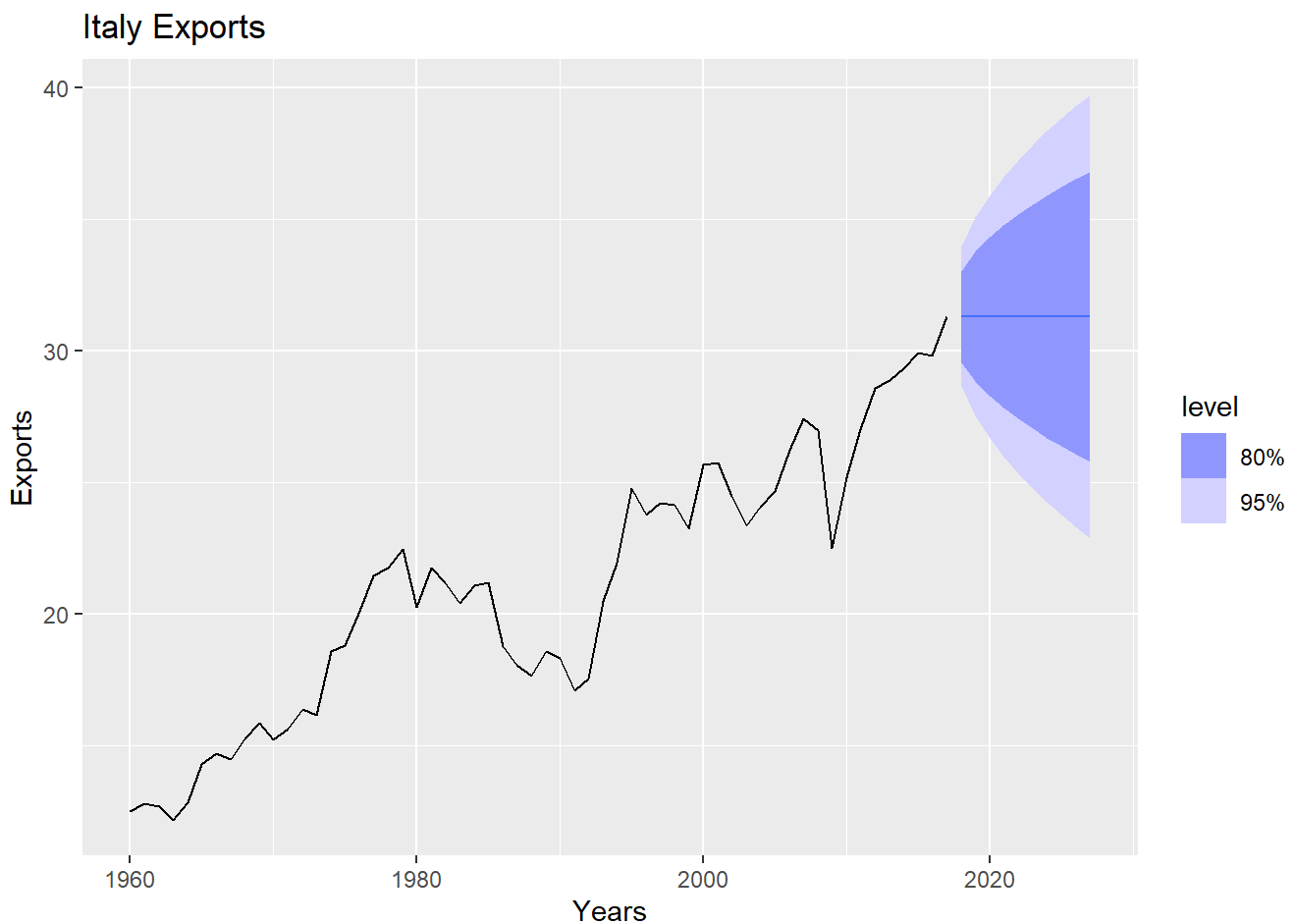
```
## Seeing optimal values
print(tidy(fit))
```

```
## # A tibble: 2 × 4
##   Country .model                term estimate
##   <fct>    <chr>                <chr>     <dbl>
## 1 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"... alpha      1.00
## 2 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"... l[0]      12.5
```

```
## Generating and Printing the forecast
fc <- fit |> forecast(h = 10) #10 years
print(fc)
```

```
## # A fable: 10 x 5 [1Y]
## # Key:   Country, .model [1]
##   Country .model                                     Year
##   <fct>   <chr>                                     <dbl>
## 1 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2018
## 2 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2019
## 3 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2020
## 4 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2021
## 5 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2022
## 6 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2023
## 7 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2024
## 8 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2025
## 9 Italy   "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2026
## 10 Italy  "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2027
## # i 2 more variables: Exports <dist>, .mean <dbl>
```

```
#Plotting
fc |>
  autoplot(italy_economy) +
    labs(x="Years", y="Exports",
         title = "Italy Exports")
```



c) Compute the RMSE values for the training data.

Using the accuracy function the RMSE for the model for the training data is ~1.335.

```
accuracy(fit)
```

```
## # A tibble: 1 × 11
##   Country .model      .type    ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <fct>   <chr>      <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Italy  "ETS(Exports... Trai... 0.324  1.34  1.00  1.39  4.75  0.983  0.991 -0.00701
```

```
# RMSE 1.335065
```

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.

When looking at the results of the second model (AAN) versus the first model (ANN), the second model's results are better. All of the Error measurements are closer to zero in the AAN model than the ANN model. This is because there is a trend in the data, and the additive shift in the second model to accomodate that reality allows it to better fit the data.

```
fit_2 <- italy_economy |> model(ETS(Exports ~ error("A") + trend("A") + season("N")))
print(tidy(fit_2))
```

```
## # A tibble: 4 × 4
##   Country .model      term estimate
##   <fct>   <chr>      <chr>      <dbl>
## 1 Italy  "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... alpha  9.48e-1
## 2 Italy  "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... beta   1.00e-4
## 3 Italy  "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... l[0]   1.22e+1
## 4 Italy  "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... b[0]   3.36e-1
```

```
print(accuracy(fit_2))
```

```
## # A tibble: 1 × 11
##   Country .model      .type    ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <fct>   <chr>      <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Italy  "ETS(Expor... Trai... -0.00841  1.30  0.934 -0.291  4.47  0.916  0.962  0.0375
```

```
# RMSE is 1.295316, along with all of the other Error measures are less or closer to zero than the ETA(ANN) model.
```

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

To elaborate on what was partially outlined in my answer to 5d, the second model fits the data better than the first. The shift away from not accommodating any longer term trend in the data model allows the second model, with an additive accomodation, to more accurately fit the initial data set. As mentioned before all of the Error measurements in the second model are closer to zero than those in the first, indicating a generally better

model.

f) Calculate a 95% prediction interval for the first forecast for each model, using the RMSE values and assuming normal errors. Compare your intervals with those produced using R.

The upper and lower bounds for both model 1 (ANN) and model 2 (AAN) when calculated with the RMSE are as follows: - MODEL 1 (ANN): 29.0317 - 33.56106 - MODEL 2 (AAN): 29.34364 - 33.80507

The same models with the versions generated with r: - MODEL 1 (ANN): 28.37038 - 33.62962 - MODEL 2 (AAN): 29.37038 - 34.62962

The ranges calculated manually with RMSE and those calculated by default with R are slightly different. Those calculated by default with r via forecast have a slightly wider range, while the RMSE ones are calculated to be more narrow. The RMSE is looking back to the errors for past data, while the other value is a forecast.

```
# ----- FIRST MODEL
# getting prediction number (First ANN Model)
print(tidy(fit))
```

```
## # A tibble: 2 × 4
##   Country .model                term estimate
##   <fct>   <chr>                <chr>    <dbl>
## 1 Italy  "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"... alpha      1.00
## 2 Italy  "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"... l[0]      12.5
```

```
fc <- fit |> forecast(h = 1)
print(fc)
```

```
## # A fable: 1 × 5 [1Y]
## # Key:      Country, .model [1]
##   Country .model                Year
##   <fct>   <chr>                <dbl>
## 1 Italy  "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2018
## # i 2 more variables: Exports <dist>, .mean <dbl>
```

```
first_predicted_mean_value <- fc$.mean[1] #(y^)
print(fc$Exports[1])
```

```
## <distribution[1]>
## [1] N(31, 1.8)
```

```
# Manually getting s for first model
print(accuracy(fit)) #1.335065
```



```
## # A tibble: 1 × 11
##   Country .model      .type    ME  RMSE   MAE   MPE  MAPE  MASE  RMSSE    ACF1
##   <fct>   <chr>      <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 Italy   "ETS(Exports... Trai... 0.324  1.34  1.00  1.39  4.75  0.983  0.991 -0.00701
```

```
mod_1_RMSE<- accuracy(fit)
s <- sqrt(mod_1_RMSE$RMSE)
print(s)
```

```
## [1] 1.15545
```

```
## Calculations with the formula
upper_lim <- first_predicted_mean_value + (1.96*s)
lower_lim <- first_predicted_mean_value - (1.96*s)
print(upper_lim)#33.56106
```

```
## [1] 33.56106
```

```
print(lower_lim)#29.0317
```

```
## [1] 29.0317
```

```
#----- SECOND MODEL#
# getting prediction number (Second AAN Model)
print(tidy(fit_2))
```

```
## # A tibble: 4 × 4
##   Country .model      term estimate
##   <fct>   <chr>      <chr>     <dbl>
## 1 Italy   "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... alpha  9.48e-1
## 2 Italy   "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... beta   1.00e-4
## 3 Italy   "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... l[0]   1.22e+1
## 4 Italy   "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"... b[0]   3.36e-1
```

```
fc_2 <- fit_2 |> forecast(h = 1)
print(fc_2)
```

```
## # A fable: 1 × 5 [1Y]
## # Key:      Country, .model [1]
##   Country .model      Year
##   <fct>   <chr>      <dbl>
## 1 Italy   "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"N\"))" 2018
## # i 2 more variables: Exports <dist>, .mean <dbl>
```

```
first_predicted_mean_value <- fc_2$.mean[1] #(y^)  
print(fc_2$Exports[1])
```

```
## <distribution[1]>  
## [1] N(32, 1.8)
```

```
# Manually getting s for first model  
print(accuracy(fit_2)) #1.295316
```

```
## # A tibble: 1 × 11  
##   Country .model      .type      ME  RMSE  MAE    MPE  MAPE  MASE  RMSSE  ACF1  
##   <fct>   <chr>      <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Italy  "ETS(Expor... Trai... -0.00841  1.30  0.934 -0.291  4.47  0.916  0.962  0.0375
```

```
mod_2_RMSE<- accuracy(fit_2)  
s <- sqrt(mod_2_RMSE$RMSE)  
print(s)
```

```
## [1] 1.13812
```

```
## Calculations with the formula  
upper_lim <- first_predicted_mean_value + (1.96*s)  
lower_lim <- first_predicted_mean_value - (1.96*s)  
print(upper_lim)#33.80507
```

```
## [1] 33.80507
```

```
print(lower_lim)#29.34364
```

```
## [1] 29.34364
```

```
## Looking at what r produced in the models:  
#Model 1  
print(fc)#N(31, 1.8)
```

```
## # A fable: 1 × 5 [1Y]  
## # Key:      Country, .model [1]  
##   Country .model      Year  
##   <fct>   <chr>      <dbl>  
## 1 Italy  "ETS(Exports ~ error(\"A\") + trend(\"N\") + season(\"N\"))" 2018  
## # i 2 more variables: Exports <dist>, .mean <dbl>
```

```
value <- 31
s <- sqrt(1.8)
lower_bound <- value - (1.96 * s)
print(lower_bound)#28.37038
```

```
## [1] 28.37038
```

```
upper_bound <- value + (1.96 * s)
print(upper_bound)#33.62962
```

```
## [1] 33.62962
```

```
# Model 2
print(fc_2)#N(32, 1.8)
```

```
## # A tibble: 1 x 5 [1Y]
## # Key:   Country, .model [1]
##   Country .model                                Year
##   <fct>   <chr>                                <dbl>
## 1 Italy   "ETS(Exports ~ error(\"A\") + trend(\"A\") + season(\"N\"))" 2018
## # i 2 more variables: Exports <dist>, .mean <dbl>
```

```
value <- 32
s <- sqrt(1.8)
lower_bound <- value - (1.96 * s)
print(lower_bound)#29.37038
```

```
## [1] 29.37038
```

```
upper_bound <- value + (1.96 * s)
print(upper_bound)#34.62962
```

```
## [1] 34.62962
```

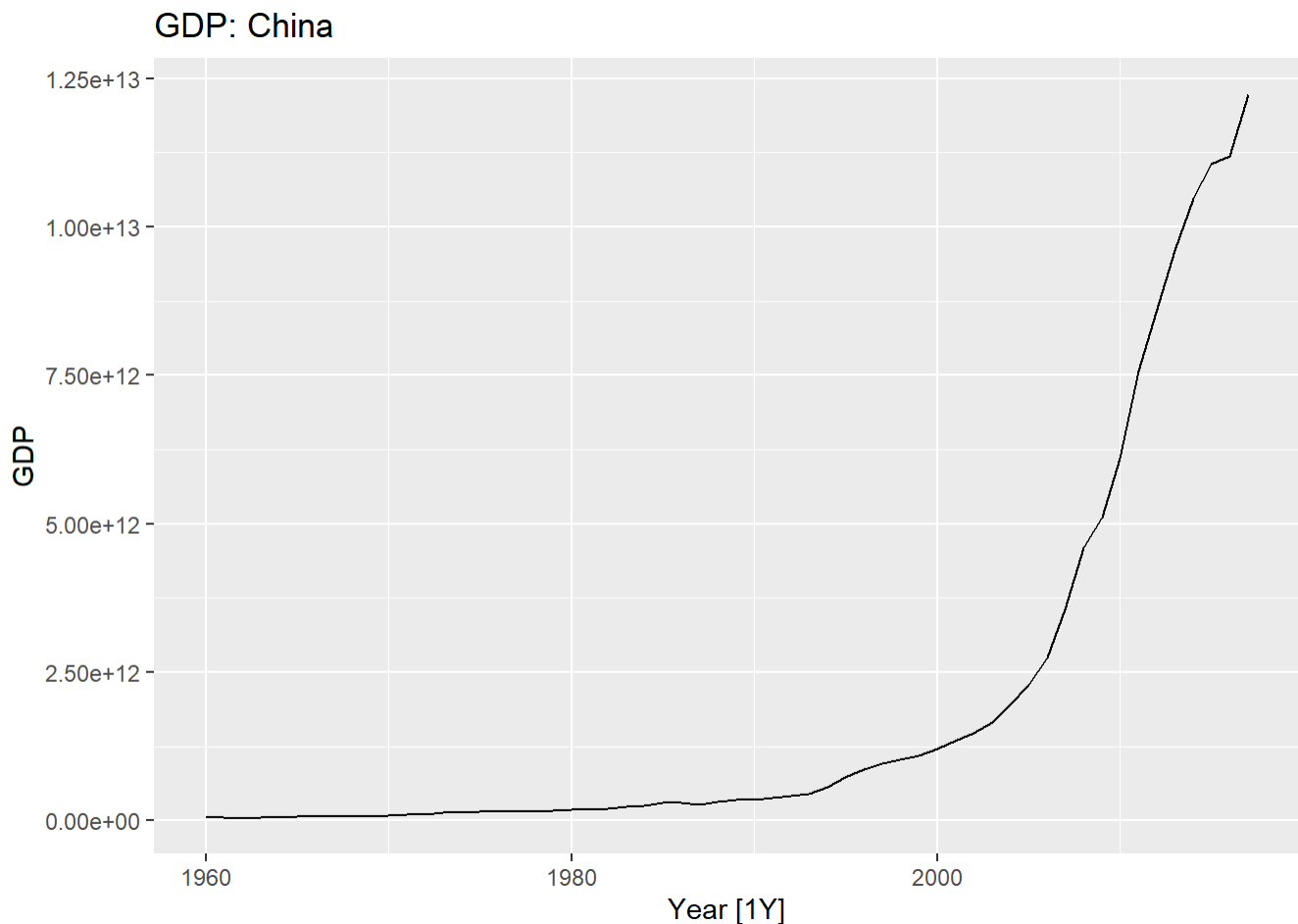
6) Forecast the Chinese GDP from the global_economy data set using an ETS model. Experiment with the various options in the ETS() function to see how much the forecasts change with damped trend, or with a Box-Cox transformation. Try to develop an intuition of what each is doing to

the forecasts.

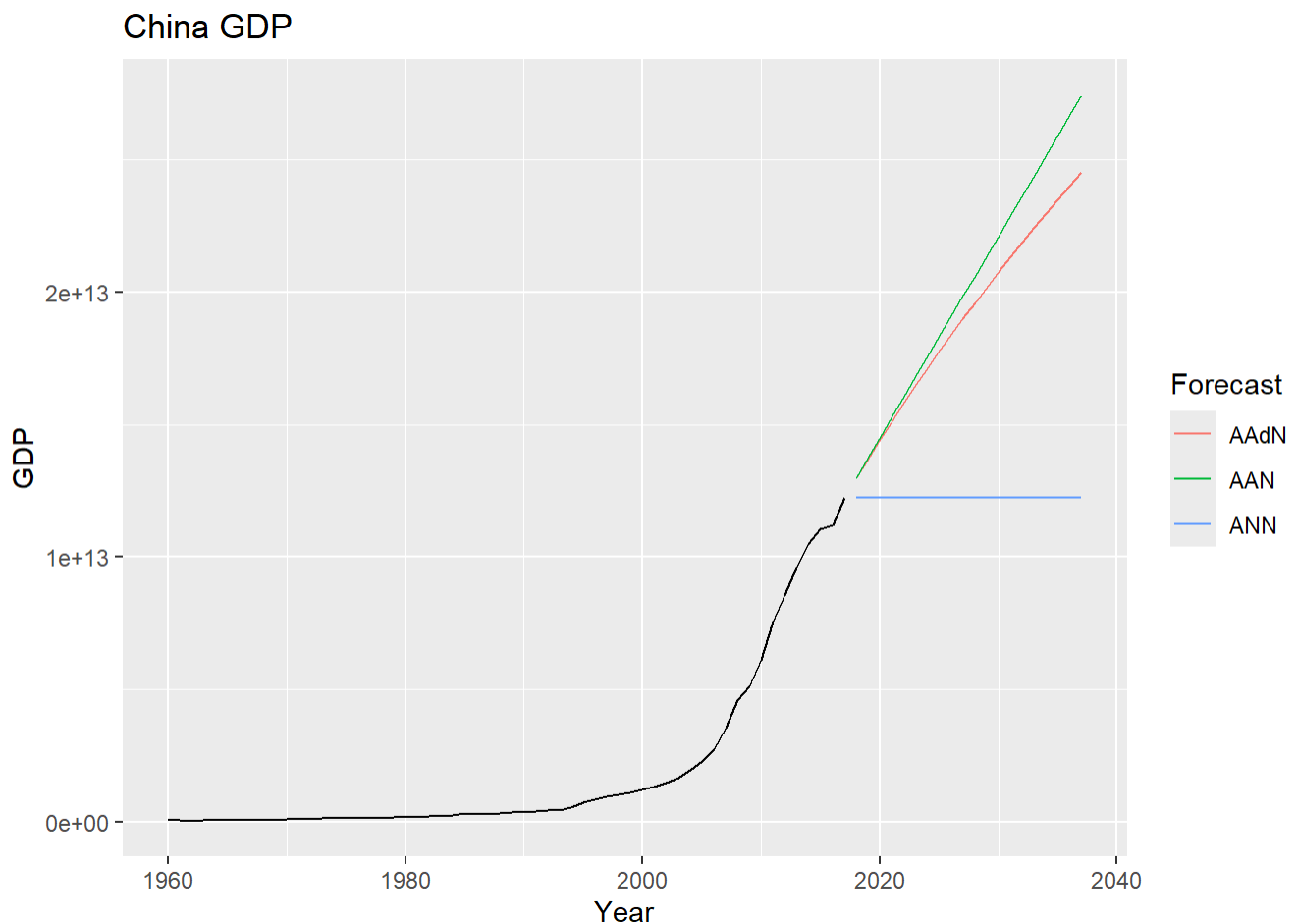
[Hint: use a relatively large value of h when forecasting, so you can clearly see the differences between the various options when plotting the forecasts.]

For this exercise, I generated three different models. One with No trend or Seasonal accommodations, so it was Simple Exponential Smoothing, the second with an additive trend accommodation (Holt's Linear Model), and then a dampened trend model. Of all of them I think the dampened trend model seems to look the most promising, as the Holt Linear model is, as the textbook states “display[s] a constant trend... indefinitely into the future”. Simple Exponential Smoothing does not take into account the trends in the data, and the dampened projection dampens newer data's consistent influence on the data so as to control for overestimating its influence.

```
china_economy <- global_economy |>
  filter(Country == "China")
# Initial Plot of the data
china_economy |>
  autoplot(GDP) +
  labs(y = "GDP", title = "GDP: China")
```



```
# Starting to Model
china_economy |>
  model(
    ## NO Trend Accommodation
    `ANN` = ETS(GDP ~ error("A") + trend("N") + season("N")),
    ## With Additive Trend Accommodation
    `AAN` = ETS(GDP ~ error("A") + trend("A") + season("N")),
    # Damped Additive Trend Accommodation
    `AAdN` = ETS(GDP ~ error("A") + trend("Ad") + season("N"))
  ) |>
  forecast(h = 20) |>
  autoplot(china_economy, level = NULL) +
  labs(title = "China GDP",
        y = "GDP") +
  guides(colour = guide_legend(title = "Forecast"))
```



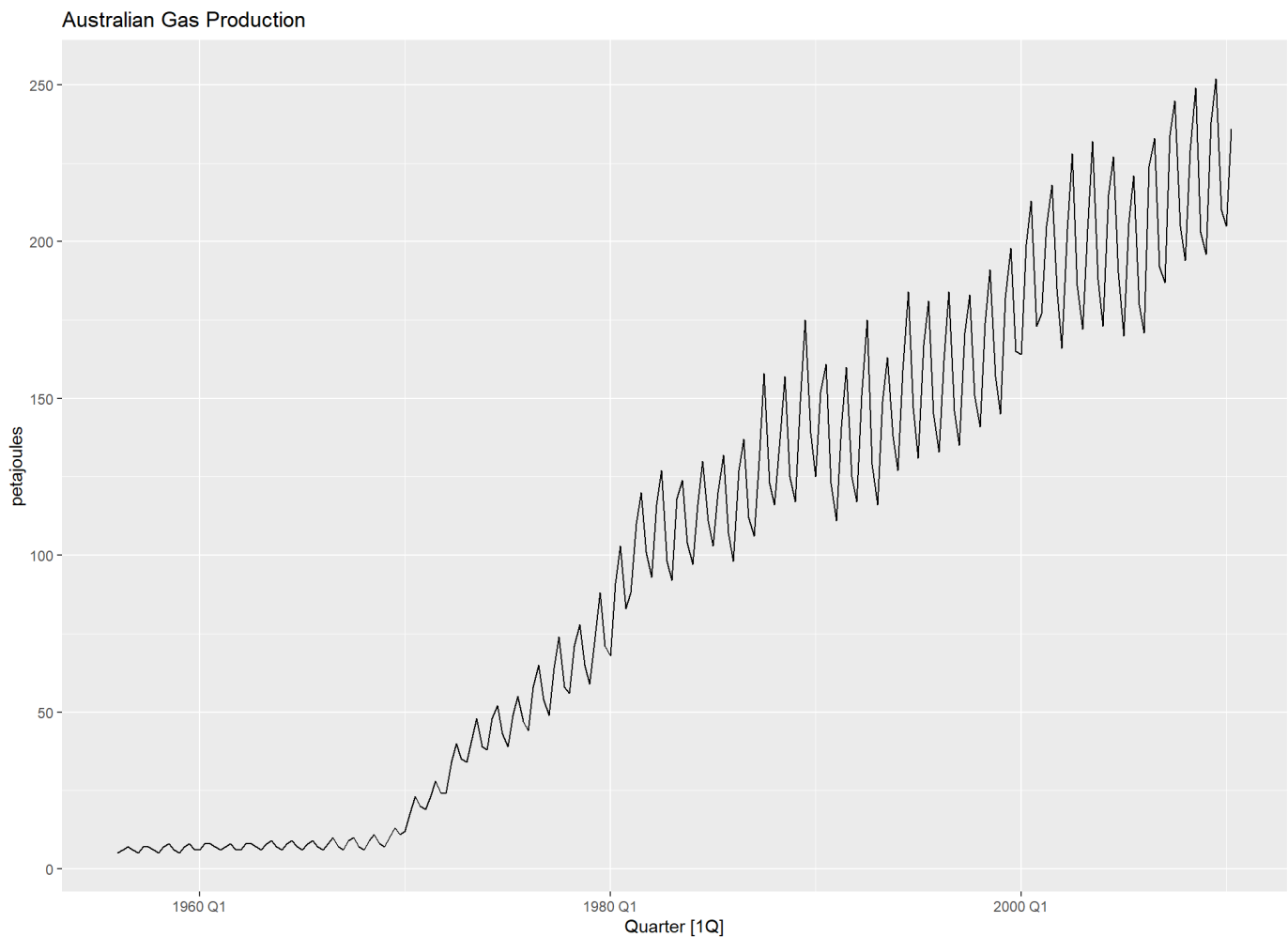
7) Find an ETS model for the Gas data from aus_production and forecast the next few years. Why is multiplicative seasonality necessary here? Experiment with making the trend damped. Does it

improve the forecasts?

Firstly, multiplicative seasonality is necessary here as there is a strong level of seasonal variation from quarter to quarter. This needs to be accommodated for. Additionally, there are changes from seasons to season so we want to avoid absolute values for this measure, so multiplicative seasonality is needed.

Using two different models here to compare, the first considered an additive non-dampened trend with multiplicative seasonality and the second the seasonality via multiplication. Of these models the damped trend model seems to fit better than the non trend damped model. Taking a look at the Error measures it shows that the non-damped model performed better.

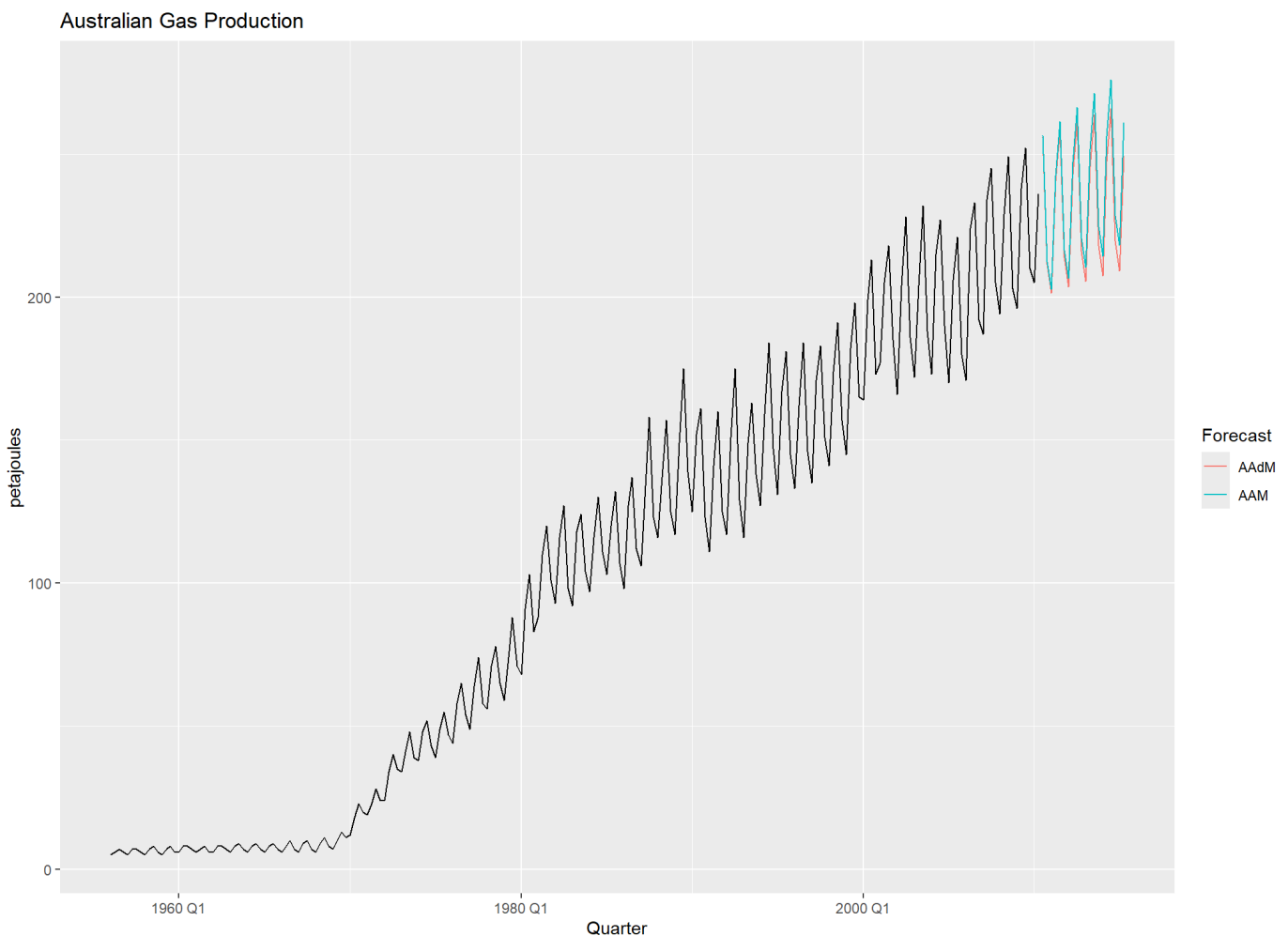
```
##?aus_production
aus_gas<- aus_production |> select(Quarter, Gas)
# Initial Plot of the data
aus_gas |>
  autoplot(Gas) +
  labs(y = "petajoules", title = "Australian Gas Production")
```



```
## Firstly there is a trend and there is seasonality here b/c of question we will do an AAN model, AAdM and an AAM
```

```
# Starting to Model
```

```
aus_gas |>  
  model(  
    ## With Additive Trend Accommodation  
    `AAM` = ETS(Gas ~ error("A") + trend("A") + season("M")),  
    # Damped Additive Trend Accommodation  
    `AAdM` = ETS(Gas ~ error("A") + trend("Ad") + season("M"))  
  ) |>  
  forecast(h = 20) |> # Four years  
  autoplot(aus_gas, level = NULL) +  
  labs(title = "Australian Gas Production",  
        y = "petajoules") +  
  guides(colour = guide_legend(title = "Forecast"))
```



```
aam <- aus_gas |> model(ETS(Gas ~ error("A") + trend("A") + season("M")))  
print(tidy(aam))
```

```
## # A tibble: 9 × 3
##   .model                                     term estimate
##   <chr>                                     <chr>     <dbl>
## 1 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" alpha  0.613
## 2 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" beta   0.00786
## 3 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" gamma  0.224
## 4 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" l[0]    3.62
## 5 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" b[0]    0.610
## 6 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" s[0]    0.980
## 7 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" s[-1]   1.15
## 8 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" s[-2]   1.08
## 9 "ETS(Gas ~ error(\"A\") + trend(\"A\") + season(\"M\"))" s[-3]   0.797
```

```
print(accuracy(aam))
```

```
## # A tibble: 1 × 10
##   .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
##   <chr>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 "ETS(Gas ~ error(\"A\"... Trai... 0.218 4.19 2.84 -0.920 5.03 0.510 0.553 0.0405
```

```
aadm <- aus_gas |> model(ETS(Gas ~ error("A") + trend("Ad") + season("M")))
print(tidy(aadm))
```

```
## # A tibble: 10 × 3
##   .model                                     term estimate
##   <chr>                                     <chr>     <dbl>
## 1 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" alpha  0.610
## 2 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" beta   0.0249
## 3 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" gamma  0.223
## 4 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" phi    0.980
## 5 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" l[0]    5.64
## 6 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" b[0]   -0.0618
## 7 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" s[0]    0.925
## 8 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" s[-1]   1.06
## 9 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" s[-2]   1.12
## 10 "ETS(Gas ~ error(\"A\") + trend(\"Ad\") + season(\"M\"))" s[-3]   0.890
```

```
print(accuracy(aadm))
```

```
## # A tibble: 1 × 10
##   .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
##   <chr>   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 "ETS(Gas ~ error(\"A\"... Trai... 0.548 4.22 2.81 1.32 4.11 0.505 0.556 0.0265
```

8) Recall your retail time series data (from Exercise

7 in Section 2.10).

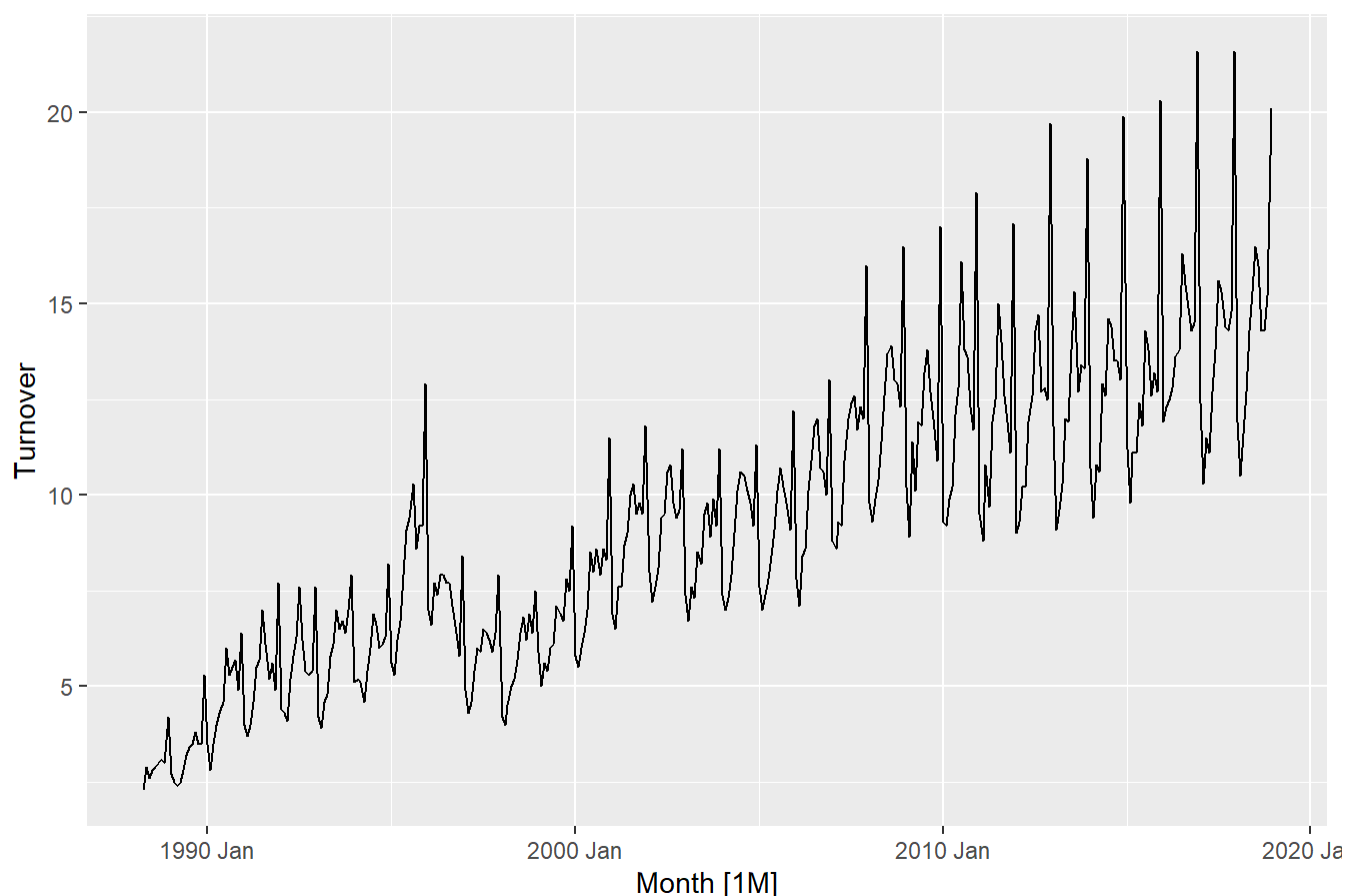
```
set.seed(12345678)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

a) Why is multiplicative seasonality necessary for this series?

Multiplicative seasonality is necessary for this series because the seasonal variations are changing over time, therefore we want a relative measurement for predicting values, not an absolute one stemming from addition.

```
## Plotting the time series
myseries |>
  autoplot(Turnover) +
  labs(y = "Turnover", title = "Australian Retail Turnover")
```

Australian Retail Turnover



b) Apply Holt-Winters' multiplicative method to the data. Experiment with making the trend damped.

```
aam <- myseries |> model(ETS(Turnover ~ error("A") + trend("A") + season("M")))
print(tidy(aam))
```

```
## # A tibble: 17 × 5
##   State      Industry      .model term estimate
##   <chr>      <chr>      <chr> <chr>    <dbl>
## 1 Northern Territory Clothing, footwear and personal acc... "ETS(... alpha 0.550
## 2 Northern Territory Clothing, footwear and personal acc... "ETS(... beta 0.000100
## 3 Northern Territory Clothing, footwear and personal acc... "ETS(... gamma 0.202
## 4 Northern Territory Clothing, footwear and personal acc... "ETS(... l[0] 2.35
## 5 Northern Territory Clothing, footwear and personal acc... "ETS(... b[0] 0.0334
## 6 Northern Territory Clothing, footwear and personal acc... "ETS(... s[0] 0.813
## 7 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1] 0.798
## 8 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-2] 0.775
## 9 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-3] 1.32
## 10 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-4] 0.992
## 11 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-5] 1.06
## 12 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-6] 1.00
## 13 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-7] 1.14
## 14 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-8] 1.21
## 15 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-9] 1.02
## 16 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1... 0.998
## 17 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1... 0.871
```

```
print(accuracy(aam))
```

```
## # A tibble: 1 × 12
##   State      Industry .model .type      ME RMSE MAE MPE MAPE MASE RMSSE
##   <chr>      <chr>      <chr> <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Northern ... Clothin... "ETS(... Trai... -0.00119 0.600 0.443 -0.265 5.21 0.506 0.517
## # i 1 more variable: ACF1 <dbl>
```

```
aadm <- myseries |> model(ETS(Turnover ~ error("A") + trend("Ad") + season("M")))
print(tidy(aadm))
```

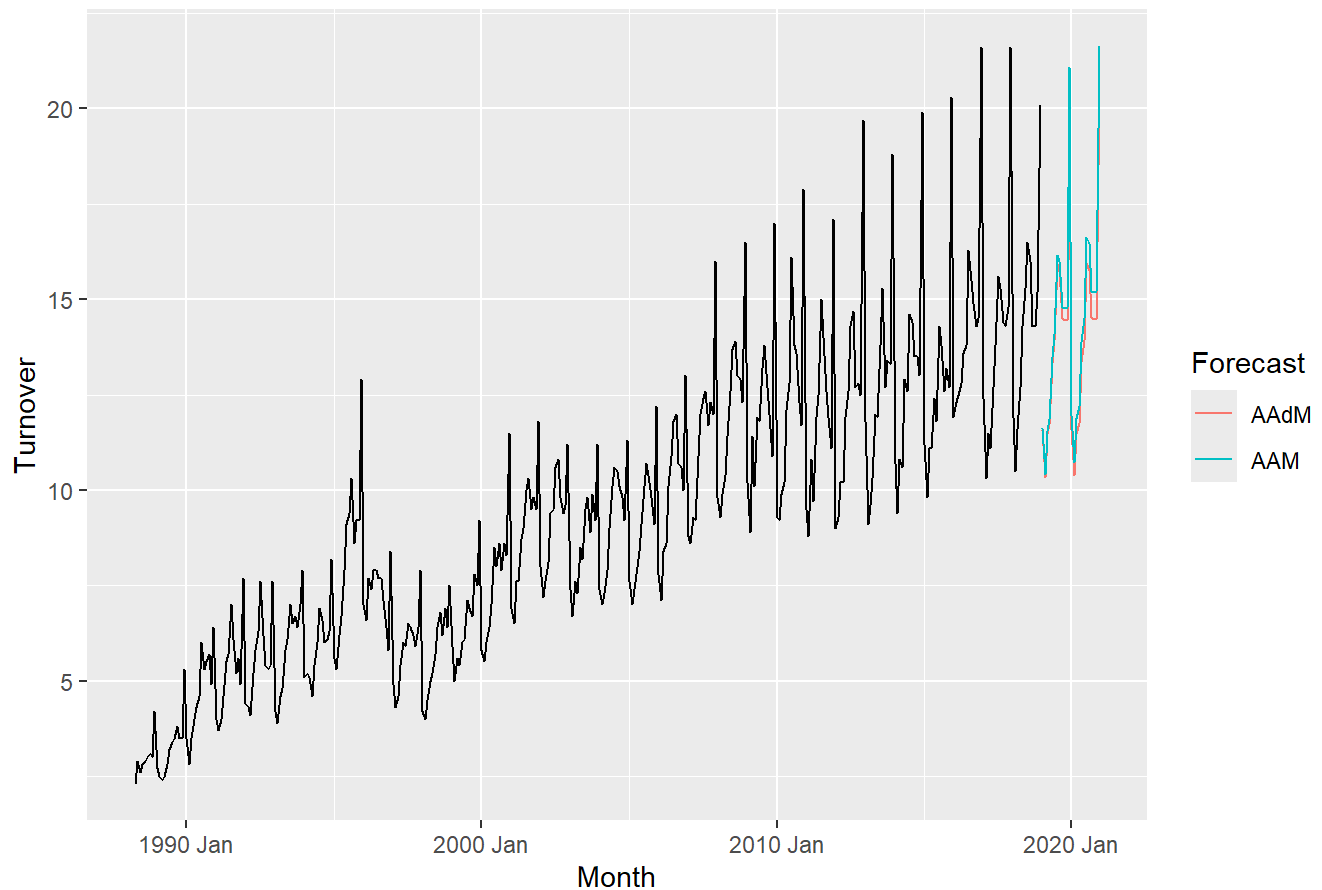
```
## # A tibble: 18 × 5
##   State      Industry      .model term estimate
##   <chr>      <chr>      <chr> <chr>      <dbl>
## 1 Northern Territory Clothing, footwear and personal acc... "ETS(... alpha 5.56e-1
## 2 Northern Territory Clothing, footwear and personal acc... "ETS(... beta 1.07e-4
## 3 Northern Territory Clothing, footwear and personal acc... "ETS(... gamma 2.07e-1
## 4 Northern Territory Clothing, footwear and personal acc... "ETS(... phi 8.44e-1
## 5 Northern Territory Clothing, footwear and personal acc... "ETS(... l[0] 2.64e+0
## 6 Northern Territory Clothing, footwear and personal acc... "ETS(... b[0] -5.16e-2
## 7 Northern Territory Clothing, footwear and personal acc... "ETS(... s[0] 8.03e-1
## 8 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1] 8.06e-1
## 9 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-2] 8.20e-1
## 10 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-3] 1.31e+0
## 11 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-4] 9.72e-1
## 12 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-5] 1.06e+0
## 13 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-6] 1.03e+0
## 14 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-7] 1.10e+0
## 15 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-8] 1.22e+0
## 16 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-9] 1.05e+0
## 17 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1... 9.99e-1
## 18 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1... 8.41e-1
```

```
print(accuracy(aadm))
```

```
## # A tibble: 1 × 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 Nort... Clothin... "ETS(... Trai... 0.0582 0.602 0.449 0.499 5.23 0.512 0.520 -0.0490
```

```
# Starting to Model
myseries |>
  model(
    'AAM' = ETS(Turnover ~ error("A") + trend("A") + season("M")),
    'AAdM' = ETS(Turnover ~ error("A") + trend("Ad") + season("M"))
  ) |>
  forecast(h = 24) |> # 2 years
  autoplot(myseries, level = NULL) +
  labs(title = "Australian Retail Turnover",
    y = "Turnover") +
  guides(colour = guide_legend(title = "Forecast"))
```

Australian Retail Turnover



c) Compare the RMSE of the one-step forecasts from the two methods. Which do you prefer?

While based on the plot and visualization of the data forecast i would say the damped trend looks to be more accurate, the RMSE for these models shows that the lower error value is yielded from the Additive non damped trend accommodation with a RMSE of ~0.599, while the damped model had a RMSE of 0.602. Both values are extremely close.

```
aam <- myseries |> model(ETS(Turnover ~ error("A") + trend("A") + season("M")))
print(tidy(aam))
```

```
## # A tibble: 17 x 5
##   State      Industry      .model term estimate
##   <chr>      <chr>      <chr> <chr>    <dbl>
## 1 Northern Territory Clothing, footwear and personal acc... "ETS(... alpha 0.550
## 2 Northern Territory Clothing, footwear and personal acc... "ETS(... beta 0.000100
## 3 Northern Territory Clothing, footwear and personal acc... "ETS(... gamma 0.202
## 4 Northern Territory Clothing, footwear and personal acc... "ETS(... l[0] 2.35
## 5 Northern Territory Clothing, footwear and personal acc... "ETS(... b[0] 0.0334
## 6 Northern Territory Clothing, footwear and personal acc... "ETS(... s[0] 0.813
## 7 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1] 0.798
## 8 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-2] 0.775
## 9 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-3] 1.32
## 10 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-4] 0.992
## 11 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-5] 1.06
## 12 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-6] 1.00
## 13 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-7] 1.14
## 14 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-8] 1.21
## 15 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-9] 1.02
## 16 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1... 0.998
## 17 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1... 0.871
```

```
aam_forecast <- aam |> forecast(h = 12)
print(aam_forecast)
```

```
## # A fable: 12 x 6 [1M]
## # Key:      State, Industry, .model [1]
##   State      Industry      .model   Month      Turnover .mean
##   <chr>      <chr>      <chr>    <mth>      <dist> <dbl>
## 1 Northern Territory Clothing, footwear and... "ETS(... 2019 Jan sample[5000] 11.6
## 2 Northern Territory Clothing, footwear and... "ETS(... 2019 Feb sample[5000] 10.4
## 3 Northern Territory Clothing, footwear and... "ETS(... 2019 Mar sample[5000] 11.6
## 4 Northern Territory Clothing, footwear and... "ETS(... 2019 Apr sample[5000] 11.9
## 5 Northern Territory Clothing, footwear and... "ETS(... 2019 May sample[5000] 13.5
## 6 Northern Territory Clothing, footwear and... "ETS(... 2019 Jun sample[5000] 14.1
## 7 Northern Territory Clothing, footwear and... "ETS(... 2019 Jul sample[5000] 16.2
## 8 Northern Territory Clothing, footwear and... "ETS(... 2019 Aug sample[5000] 16.0
## 9 Northern Territory Clothing, footwear and... "ETS(... 2019 Sep sample[5000] 14.8
## 10 Northern Territory Clothing, footwear and... "ETS(... 2019 Oct sample[5000] 14.8
## 11 Northern Territory Clothing, footwear and... "ETS(... 2019 Nov sample[5000] 14.8
## 12 Northern Territory Clothing, footwear and... "ETS(... 2019 Dec sample[5000] 21.1
```

```
print(accuracy(aam))#RMSE 0.5995194
```

```
## # A tibble: 1 x 12
##   State      Industry .model .type      ME  RMSE  MAE    MPE  MAPE  MASE  RMSSE
##   <chr>      <chr>    <chr> <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Northern ... Clothin... "ETS(... Trai... -0.00119 0.600 0.443 -0.265 5.21 0.506 0.517
## # i 1 more variable: ACF1 <dbl>
```

```
aadm <- myseries |> model(ETS(Turnover ~ error("A") + trend("Ad") + season("M")))
print(tidy(aadm))
```

```
## # A tibble: 18 × 5
##   State      Industry      .model term estimate
##   <chr>      <chr>      <chr> <chr>      <dbl>
## 1 Northern Territory Clothing, footwear and personal acc... "ETS(... alpha  5.56e-1
## 2 Northern Territory Clothing, footwear and personal acc... "ETS(... beta   1.07e-4
## 3 Northern Territory Clothing, footwear and personal acc... "ETS(... gamma  2.07e-1
## 4 Northern Territory Clothing, footwear and personal acc... "ETS(... phi    8.44e-1
## 5 Northern Territory Clothing, footwear and personal acc... "ETS(... l[0]    2.64e+0
## 6 Northern Territory Clothing, footwear and personal acc... "ETS(... b[0]   -5.16e-2
## 7 Northern Territory Clothing, footwear and personal acc... "ETS(... s[0]    8.03e-1
## 8 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1]   8.06e-1
## 9 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-2]   8.20e-1
## 10 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-3]   1.31e+0
## 11 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-4]   9.72e-1
## 12 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-5]   1.06e+0
## 13 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-6]   1.03e+0
## 14 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-7]   1.10e+0
## 15 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-8]   1.22e+0
## 16 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-9]   1.05e+0
## 17 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1...  9.99e-1
## 18 Northern Territory Clothing, footwear and personal acc... "ETS(... s[-1...  8.41e-1
```

```
aadm_forecast <-aadm |> forecast(h = 12)
print(aadm_forecast)
```

```
## # A fable: 12 × 6 [1M]
## # Key:   State, Industry, .model [1]
##   State      Industry      .model   Month   Turnover .mean
##   <chr>      <chr>      <chr>    <mth>    <dist> <dbl>
## 1 Northern Territory Clothing, footwear and... "ETS(... 2019 Jan sample[5000] 11.5
## 2 Northern Territory Clothing, footwear and... "ETS(... 2019 Feb sample[5000] 10.3
## 3 Northern Territory Clothing, footwear and... "ETS(... 2019 Mar sample[5000] 11.4
## 4 Northern Territory Clothing, footwear and... "ETS(... 2019 Apr sample[5000] 11.7
## 5 Northern Territory Clothing, footwear and... "ETS(... 2019 May sample[5000] 13.3
## 6 Northern Territory Clothing, footwear and... "ETS(... 2019 Jun sample[5000] 13.9
## 7 Northern Territory Clothing, footwear and... "ETS(... 2019 Jul sample[5000] 15.9
## 8 Northern Territory Clothing, footwear and... "ETS(... 2019 Aug sample[5000] 15.7
## 9 Northern Territory Clothing, footwear and... "ETS(... 2019 Sep sample[5000] 14.4
## 10 Northern Territory Clothing, footwear and... "ETS(... 2019 Oct sample[5000] 14.4
## 11 Northern Territory Clothing, footwear and... "ETS(... 2019 Nov sample[5000] 14.4
## 12 Northern Territory Clothing, footwear and... "ETS(... 2019 Dec sample[5000] 20.4
```

```
print(accuracy(aadm)) #RSME 0.6024724
```

```
## # A tibble: 1 × 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 Nort... Clothin... "ETS(... Trai... 0.0582 0.602 0.449 0.499  5.23 0.512 0.520 -0.0490
```

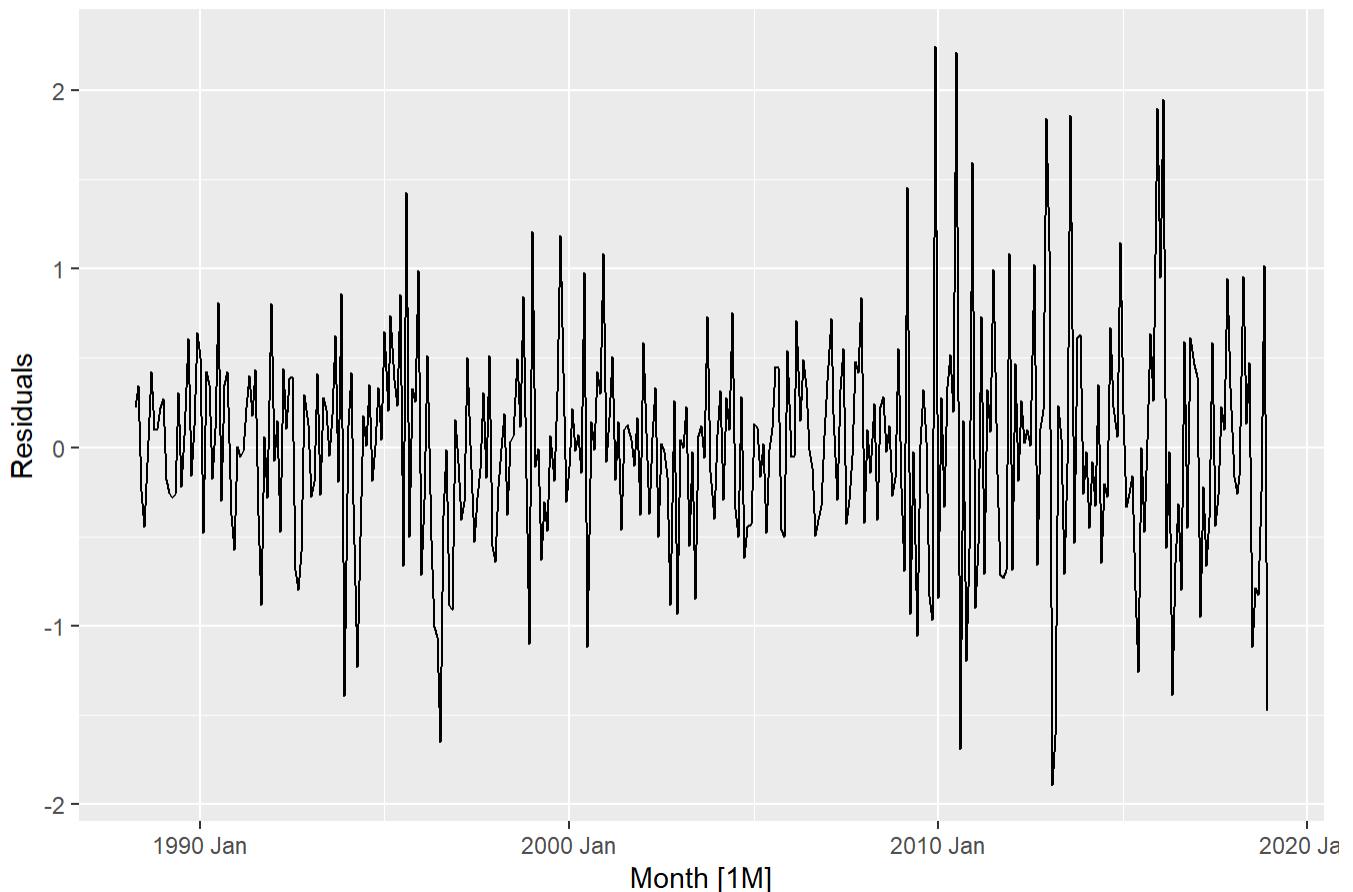
d) Check that the residuals from the best method look like white noise.

This plot of residuals looks like mostly white noise, with some variation exceptions towards the more recent / right portion of the chart. That being said, the variations don't have a pattern behind them, so can also be considered white noise.

```
##AAM (Best method based on slightly better RMSE value)
autoplot(residuals(aam))+
  labs(title = "Residuals Plot", y = "Residuals")
```

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

Residuals Plot



e) Now find the test set RMSE, while training the model to the end of 2010. Can you beat the seasonal naïve approach from Exercise 7 in Section 5.11?

Yes, by using an Additive trend and multiplicative seasonal accommodations the AAM type ETS model was able to obtain a smaller RMSE value than the SNAIVE model from Section 5.

```
lim_2010 <- myseries |> filter(year(Month)<=2010)
## Taking better performing model

model_aam <- lim_2010 |> model(ETS(Turnover ~ error("A") + trend("A") + season("M")))
aam_fc <- model_aam |> forecast(h = 12)
print(aam_fc)
```

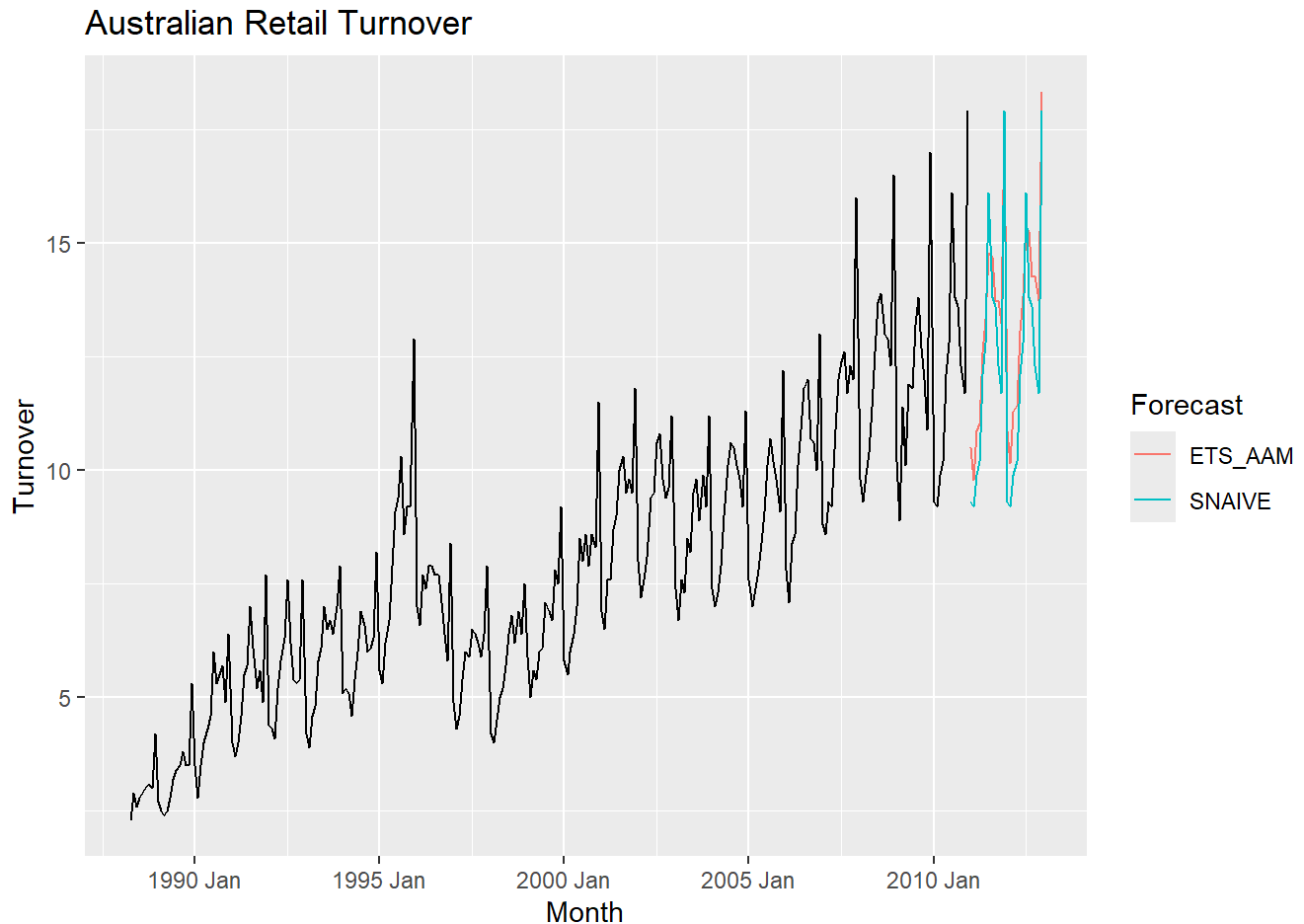
```
## # A fable: 12 x 6 [1M]
## # Key:      State, Industry, .model [1]
##   State      Industry      .model   Month      Turnover .mean
##   <chr>      <chr>      <chr>    <mth>      <dist> <dbl>
## 1 Northern Territory Clothing, footwear and... "ETS(... 2011 Jan sample[5000] 10.5
## 2 Northern Territory Clothing, footwear and... "ETS(... 2011 Feb sample[5000]  9.80
## 3 Northern Territory Clothing, footwear and... "ETS(... 2011 Mar sample[5000] 10.9
## 4 Northern Territory Clothing, footwear and... "ETS(... 2011 Apr sample[5000] 11.0
## 5 Northern Territory Clothing, footwear and... "ETS(... 2011 May sample[5000] 12.6
## 6 Northern Territory Clothing, footwear and... "ETS(... 2011 Jun sample[5000] 13.3
## 7 Northern Territory Clothing, footwear and... "ETS(... 2011 Jul sample[5000] 14.8
## 8 Northern Territory Clothing, footwear and... "ETS(... 2011 Aug sample[5000] 14.7
## 9 Northern Territory Clothing, footwear and... "ETS(... 2011 Sep sample[5000] 13.7
## 10 Northern Territory Clothing, footwear and... "ETS(... 2011 Oct sample[5000] 13.8
## 11 Northern Territory Clothing, footwear and... "ETS(... 2011 Nov sample[5000] 13.3
## 12 Northern Territory Clothing, footwear and... "ETS(... 2011 Dec sample[5000] 17.7
```

```
snaive_model <- lim_2010 |> model(SNAIVE(Turnover))
snaive_fc <- snaive_model |> forecast(h = 12)
print(snaive_fc)
```

```
## # A fable: 12 x 6 [1M]
## # Key:      State, Industry, .model [1]
##   State      Industry      .model   Month
##   <chr>      <chr>      <chr>    <mth>
## 1 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Jan
## 2 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Feb
## 3 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Mar
## 4 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Apr
## 5 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 May
## 6 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Jun
## 7 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Jul
## 8 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Aug
## 9 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Sep
## 10 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Oct
## 11 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Nov
## 12 Northern Territory Clothing, footwear and personal accessory... SNAIV... 2011 Dec
## # i 2 more variables: Turnover <dist>, .mean <dbl>
```



```
#Plotting
lim_2010 |>
  model('ETS_AAM' = ETS(Turnover ~ error("A") + trend("A") + season("M")),
        'SNAIVE' = SNAIVE(Turnover)
  ) |>
  forecast(h = 24) |> # 2 years
  autoplot(lim_2010, level = NULL) +
  labs(title = "Australian Retail Turnover",
        y = "Turnover") +
  guides(colour = guide_legend(title = "Forecast"))
```



```
## Looking at accuracy
print(accuracy(model_aam))#RMSE 0.5115981
```

```
## # A tibble: 1 × 12
##   State      Industry .model .type      ME  RMSE  MAE    MPE  MAPE  MASE  RMSSE
##   <chr>      <chr>   <chr> <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Northern T... Clothin... "ETS(... Trai... -0.0249 0.512 0.382 -0.661 5.28 0.418 0.422
## # i 1 more variable: ACF1 <dbl>
```

```
print(accuracy(snaive_model)) #1.213731 RMSE
```

```
## # A tibble: 1 × 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 Norther... Clothin... SNAIV... Trai... 0.439  1.21 0.915  5.23 12.4    1    1 0.768
```

9) For the same retail data, try an STL decomposition applied to the Box-Cox transformed series, followed by ETS on the seasonally adjusted data. How does that compare with your best previous forecasts on the test set?

The RMSE for the season adjusted data with the same model on the limited training data Also used before is lower, meaning the error is lower so this model is better fitted.

```
## BOX COX From Chapter 3
lambda <- lim_2010 |> features(Turnover, features = guerrero) |> pull(lambda_guerrero)
print(lambda)
```

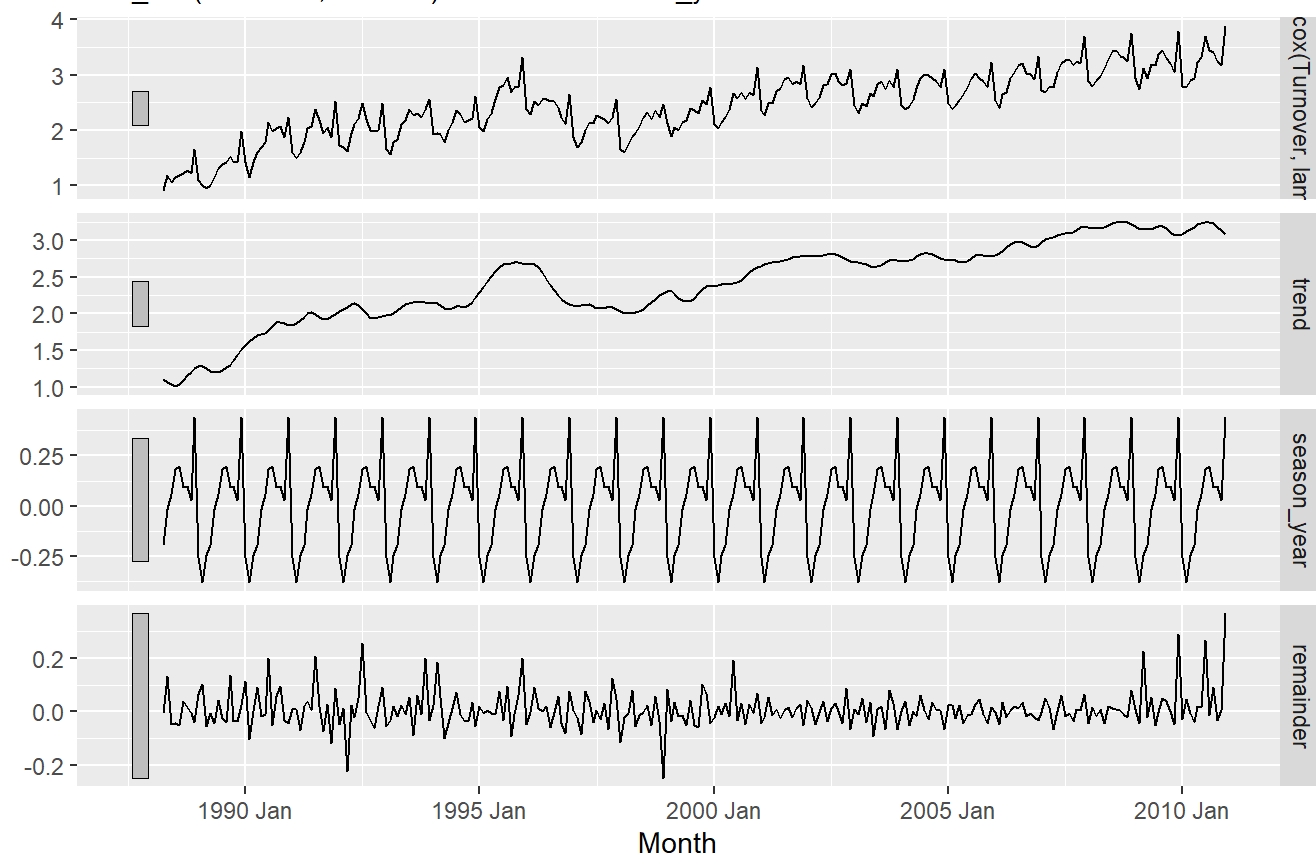
```
## [1] 0.1971431
```

```
stl_decomp <- lim_2010 |>
  model(
    STL(box_cox(Turnover, lambda) ~ trend(window = 7) +
      season(window = "periodic"),
    robust = TRUE)) |>
  components()

## Decomp Plot
stl_decomp|> autoplot()
```

STL decomposition

``box_cox(Turnover, lambda)` = trend + season_year + remainder`



```
#Takign the season adjusted data out of the decomp
seasonally_adjusted_data <- stl_decomp |> select(season_adjust)

#Modeling using AAM from before on adjusted data.
ets_decomp <-
  seasonally_adjusted_data |>
  model(ETS(season_adjust ~ error("A") + trend("A") + season("M")))

## Comparing to model before
accuracy(ets_decomp)#0.1036053 RMSE
```

```
## # A tibble: 1 × 10
##   .model      .type      ME  RMSE   MAE   MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>      <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 "ETS(season_adjus... Trai... -9.35e-4 0.104 0.0804 0.0103 3.70 0.425 0.413 0.0916
```

```
accuracy(model_aam)#0.5115981 RMSE
```

```
## # A tibble: 1 × 12
##   State      Industry .model .type      ME  RMSE  MAE    MPE  MAPE  MASE  RMSSE
##   <chr>      <chr>    <chr> <chr>    <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 Northern T... Clothin... "ETS(... Trai... -0.0249 0.512 0.382 -0.661  5.28 0.418 0.422
## # i 1 more variable: ACF1 <dbl>
```