Data 624_Exercise 8.8_HW5

Jiaxin Zheng

2025-03-31

8.8 Exercises:

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.4.2

## Registered S3 method overwritten by 'tsibble':
##   method               from
##   as_tibble.grouped_df dplyr

## -- Attaching packages -------------------------------------------- fpp3 1.0.1 --

## v tibble      3.2.1     v tsibble     1.1.6
## v dplyr       1.1.4     v tsibbledata 0.4.1
## v tidyr       1.3.1     v feasts      0.4.1
## v lubridate   1.9.4     v fable       0.4.1
## v ggplot2     3.5.1

## Warning: package 'dplyr' was built under R version 4.4.3

## Warning: package 'ggplot2' was built under R version 4.4.2

## Warning: package 'tsibbledata' was built under R version 4.4.2

## Warning: package 'feasts' was built under R version 4.4.2

## Warning: package 'fabletools' was built under R version 4.4.2

## Warning: package 'fable' was built under R version 4.4.2

## -- Conflicts ---------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(dplyr)
```

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

- a. Use the ETS() function to estimate the equivalent model for simple exponential smoothing. Find the optimal values of
 and 0, and generate forecasts for the next four months.
- alpha = 0.3221247; initial states = 100646.6

```
?aus_livestock
```

```
## starting httpd help server ... done
```

```
head(aus_livestock)
```

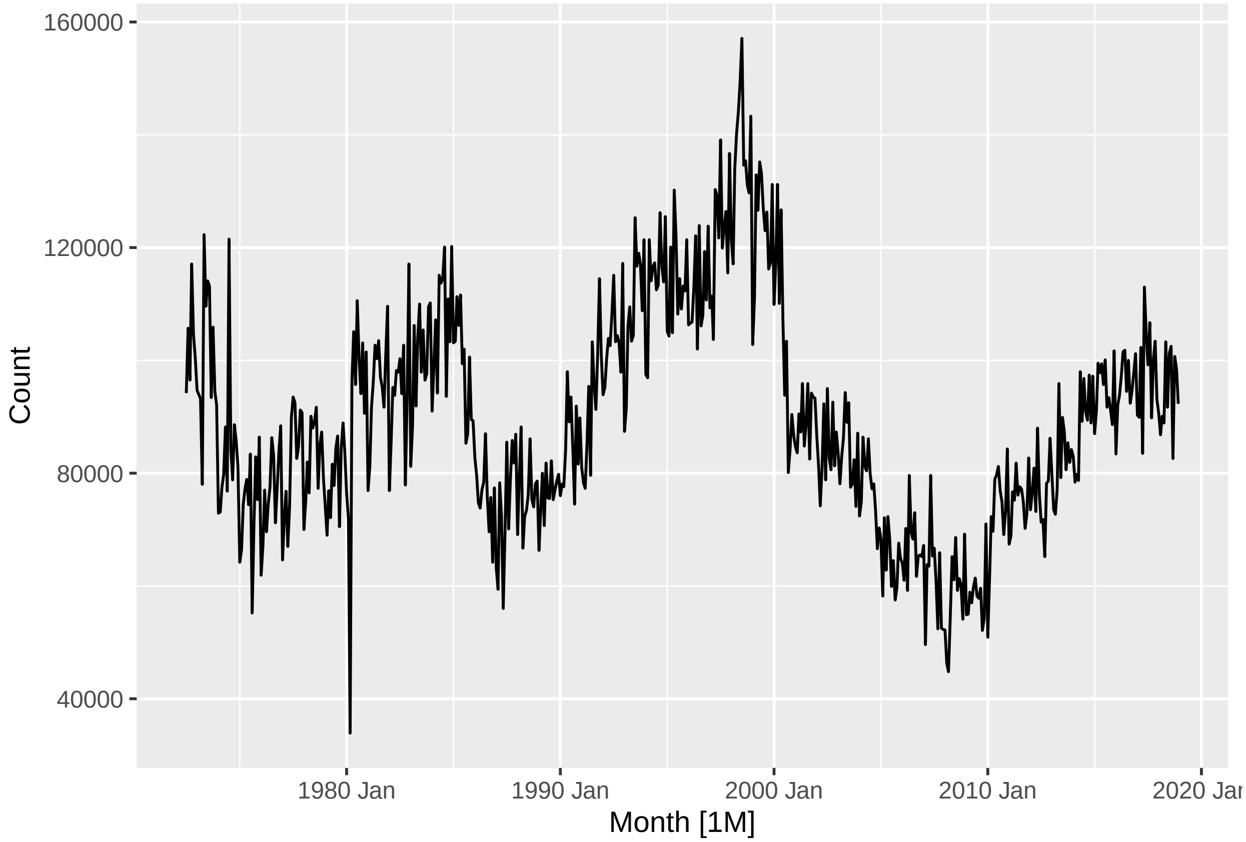
```
## # A tsibble: 6 x 4 [1M]
## # Key:       Animal, State [1]
##      Month Animal                     State                        Count
##      <mth> <fct>                      <fct>                        <dbl>
## 1 1976 Jul Bulls, bullocks and steers Australian Capital Territory  2300
## 2 1976 Aug Bulls, bullocks and steers Australian Capital Territory  2100
## 3 1976 Sep Bulls, bullocks and steers Australian Capital Territory  2100
## 4 1976 Oct Bulls, bullocks and steers Australian Capital Territory  1900
## 5 1976 Nov Bulls, bullocks and steers Australian Capital Territory  2100
## 6 1976 Dec Bulls, bullocks and steers Australian Capital Territory  1800
```

```
vic_pigs <- aus_livestock %>%
  filter(State == "Victoria", Animal == "Pigs")
vic_pigs
```

```
## # A tsibble: 558 x 4 [1M]
## # Key:       Animal, State [1]
##       Month Animal State     Count
##       <mth> <fct>  <fct>     <dbl>
##  1 1972 Jul Pigs   Victoria  94200
##  2 1972 Aug Pigs   Victoria 105700
##  3 1972 Sep Pigs   Victoria  96500
##  4 1972 Oct Pigs   Victoria 117100
##  5 1972 Nov Pigs   Victoria 104600
##  6 1972 Dec Pigs   Victoria 100500
##  7 1973 Jan Pigs   Victoria  94700
##  8 1973 Feb Pigs   Victoria  93900
##  9 1973 Mar Pigs   Victoria  93200
## 10 1973 Apr Pigs   Victoria  78000
## # i 548 more rows
```

```
autoplot(vic_pigs)
```

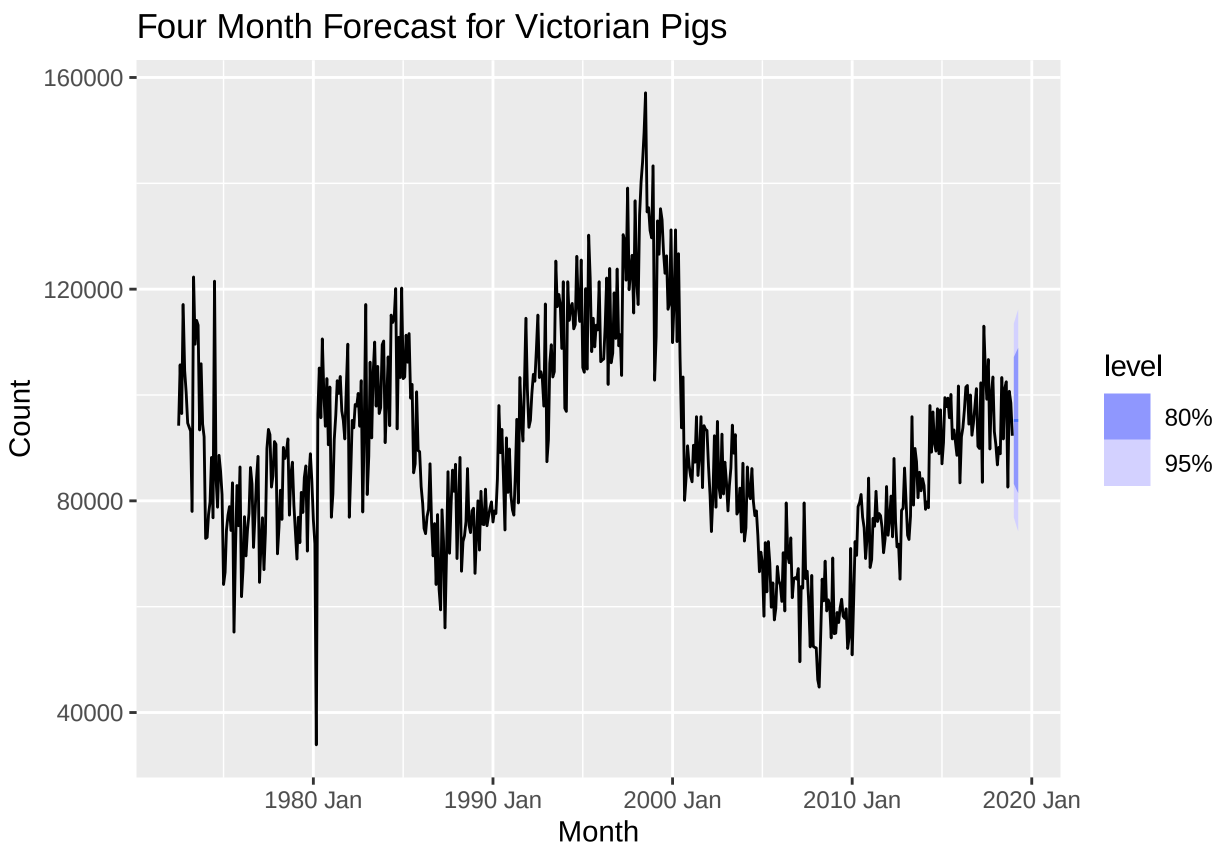
```
## Plot variable not specified, automatically selected `.vars = Count`
```



```r
# fit model
fit <- vic_pigs %>%
  model(ETS(Count ~ error("A") + trend("N") + season("N")))

report(fit)
```

```
## Series: Count
## Model: ETS(A,N,N)
##   Smoothing parameters:
##     alpha = 0.3221247
##
##   Initial states:
##       l[0]
##   100646.6
##
##   sigma^2:  87480760
##
##       AIC     AICc      BIC
## 13737.10 13737.14 13750.07
```

```r
fc <- fit %>% forecast(h = "4 months")
fc %>%
  autoplot(vic_pigs)+
  labs(title="Four Month Forecast for Victorian Pigs")
```



- b. Compute a 95% prediction interval for the first forecast using ^y±1.96s where s is the standard deviation of the residuals. Compare your interval with the interval produced by R.

```
# Get residuals standard deviation
s <- sd(residuals(fit)$.resid)

# Get first forecast value
first_forecast <- fc$.mean[1]

# Calculate manual prediction interval
lower <- first_forecast - 1.96 * s
upper <- first_forecast + 1.96 * s

paste0("lower:", lower)
```

```
## [1] "lower:76871.0124775157"
```

```
paste0("upper:", upper)
```

```
## [1] "upper:113502.102384467"
```

```
# R's interval
fc |>
    head(1) |>
    hilo() |>
    pull(`95%`)
```

```
## <hilo[1]>
## [1] [76854.79, 113518.3]95
```

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

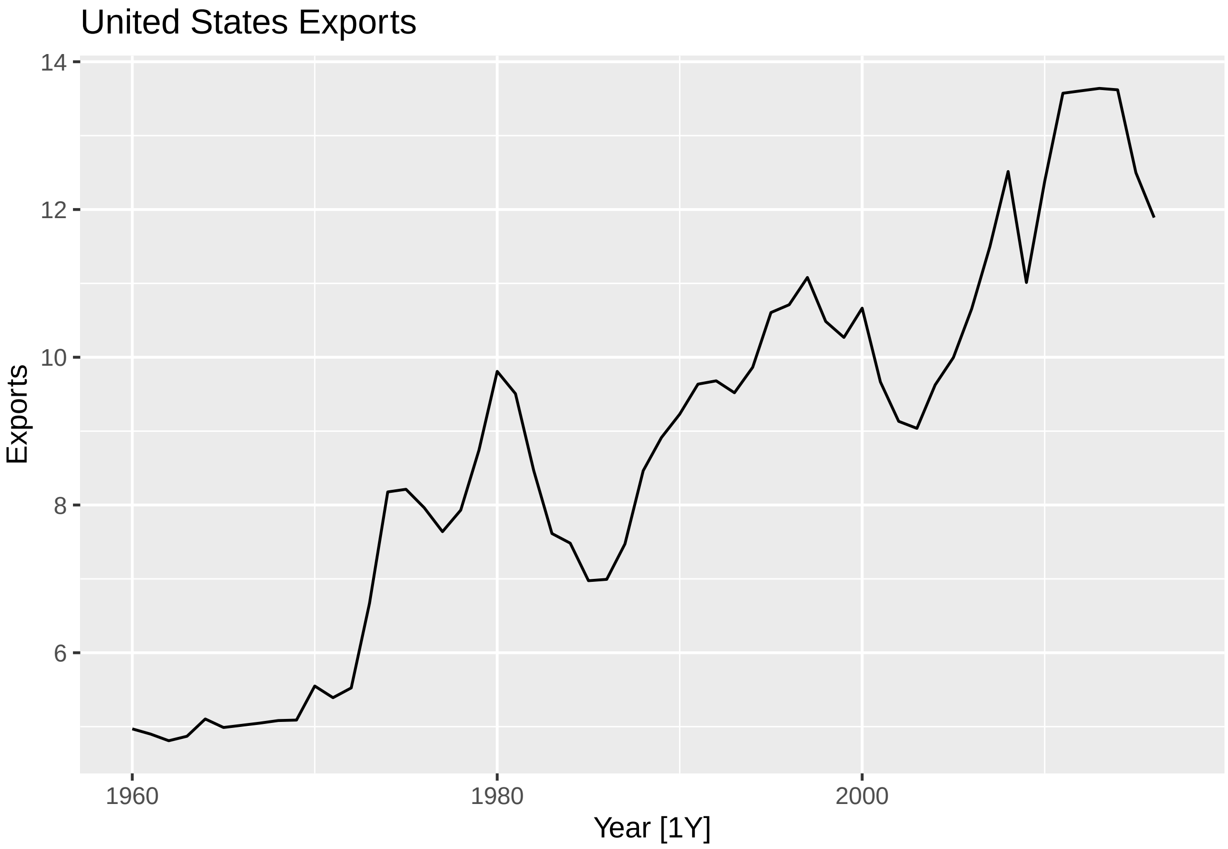
- a.Plot the Exports series and discuss the main features of the data.

```
head(global_economy)
```

```
## # 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
```

```
US <- global_economy %>%
  filter(Country == "United States")

US %>%
  autoplot(Exports) +
  labs(title = "United States Exports")
```

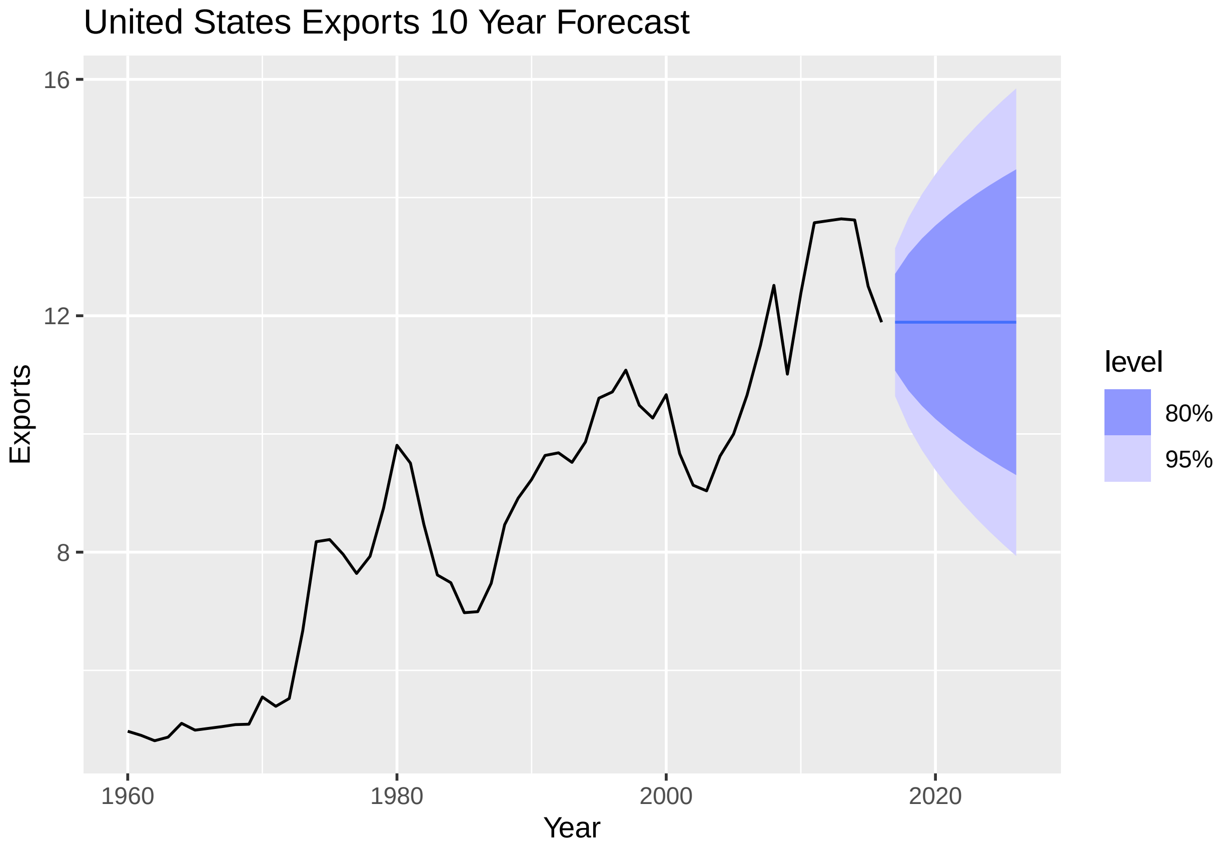
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



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

```
fit <- US %>%
  filter(!is.na(Exports)) %>%
  model(ETS(Exports ~ error("A") + trend("N")+ season("N")))

fc <- fit %>% forecast(h = 10)

autoplot(fc, US) +
  labs(title = "United States Exports 10 Year Forecast")
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



- c.Compute the RMSE values for the training data.
- RMSE number is 0.6270672.

```
accuracy(fit)
```

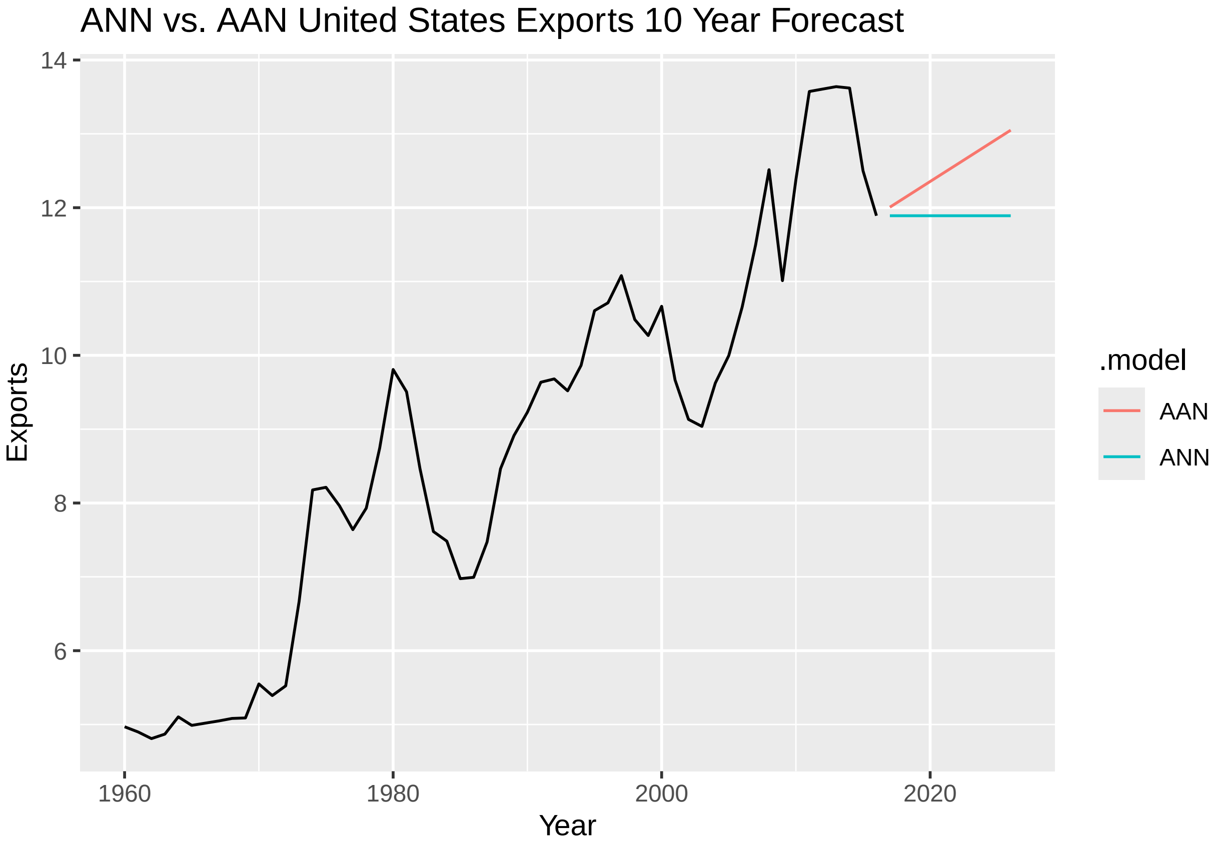
```
## # A tibble: 1 x 11
##   Country       .model      .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <fct>         <chr>       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 United States "ETS(Expo~ Trai~ 0.125 0.627 0.465  1.37  5.10 0.990 0.992 0.239
```

- 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.

- The AAN have RMSE number 0.6270672.

```
fit_2 <- US %>%
  filter(!is.na(Exports)) %>%
  model(ANN= ETS(Exports ~ error("A") + trend("N")+ season("N")),
        AAN = ETS(Exports ~ error("A") + trend("A")+ season("N")))

fc_2 <- fit_2 %>% forecast(h = 10)

autoplot(fc_2, US, level = NULL) +
  labs(title = "ANN vs. AAN United States Exports 10 Year Forecast")
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



```
accuracy(fit_2)
```

```
## # A tibble: 2 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 United States ANN    Train~ 0.125   0.627 0.465  1.37     5.10 0.990 0.992 0.239
## 2 United States AAN    Train~ 0.0108  0.615 0.466 -0.0570   5.19 0.992 0.973 0.238
```

- e.Compare the forecasts from both methods. Which do you think is best?
- The AAN method seems better, the RMSE number is lower than ANN.
- 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.

```
# Get residuals standard deviation
s_2 <- sd(residuals(fit_2)$.resid)

# Get first forecast value
first_forecast_2 <- fc_2$.mean[1]

# Calculate manual prediction interval
```

```
lower_2 <- first_forecast_2 - 1.96 * s
upper_2 <- first_forecast_2 + 1.96 * s

paste0("lower:", lower)
```

```
## [1] "lower:76871.0124775157"
```

```
paste0("upper:", upper)
```

```
## [1] "upper:113502.102384467"
```

```
# R's interval
fc_2 |>
    head(1) |>
    hilo() |>
    pull(`95%`)
```

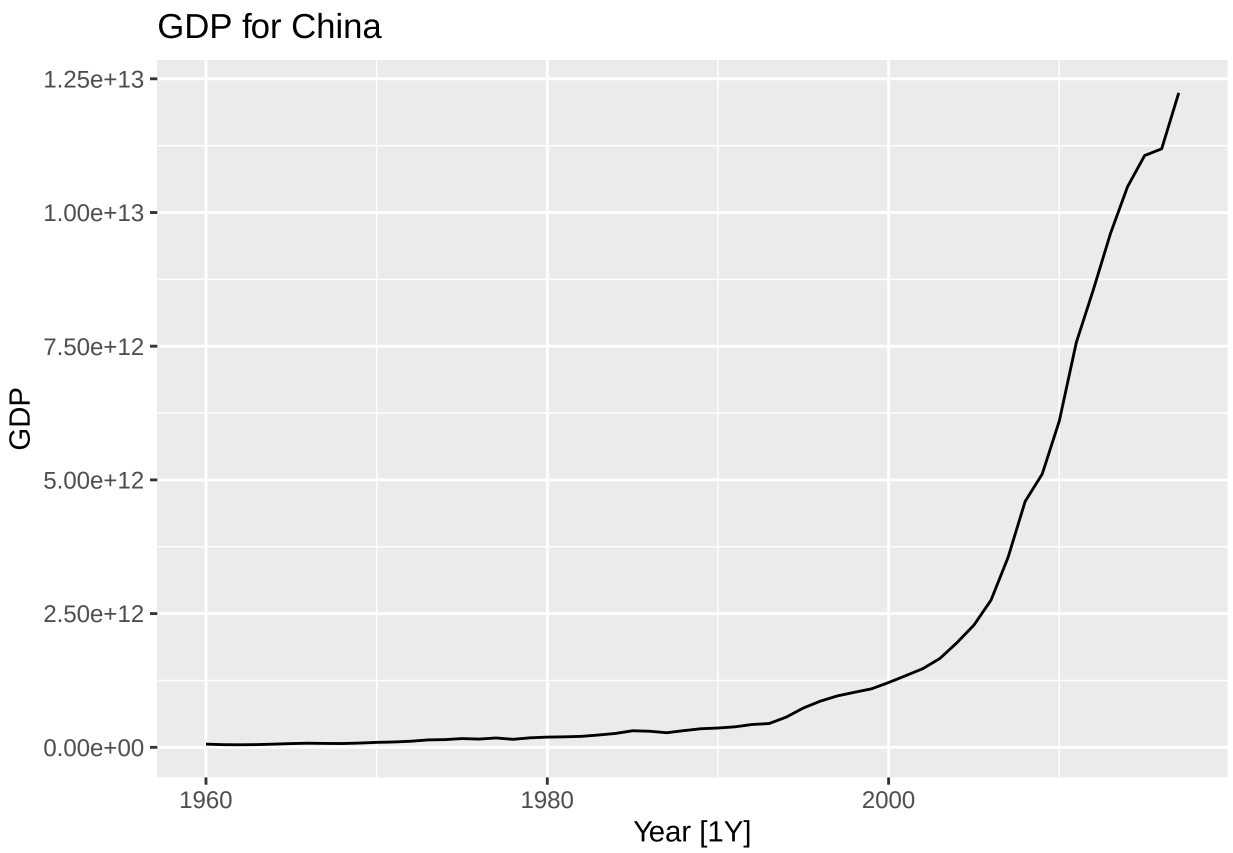
```
## <hilo[1]>
## [1] [10.63951, 13.14186]95
```

‘

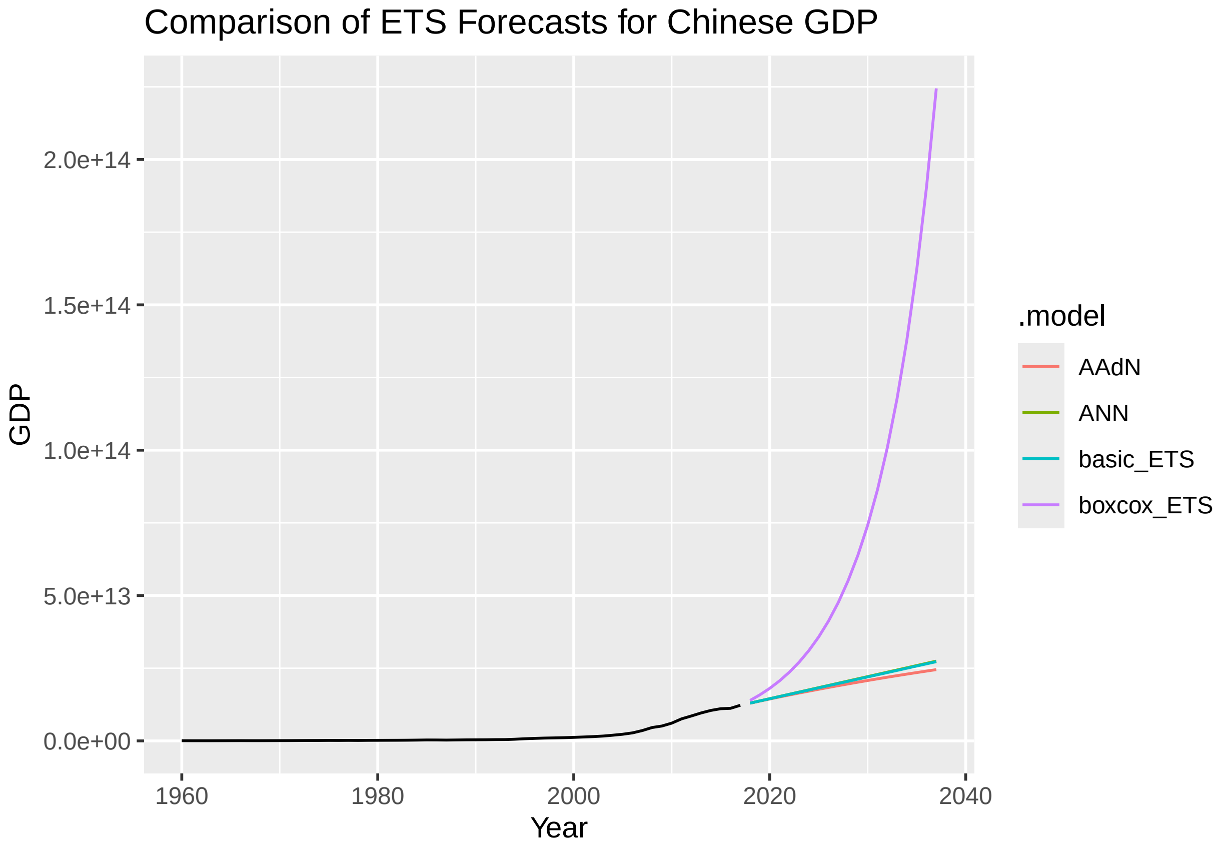
6. Forecast the Chinese GDP from the global_economy data set using an ETS model. Experiment with the various options in the ETS() function to see how much the forecasts change with damped trend, or with a Box-Cox transformation. Try to develop an intuition of what each is doing to the forecasts.

- 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.

```
china_gdp <- global_economy %>%
  filter(Country == "China")

china_gdp %>%
  autoplot(GDP) +
  labs(title = "GDP for China")
```



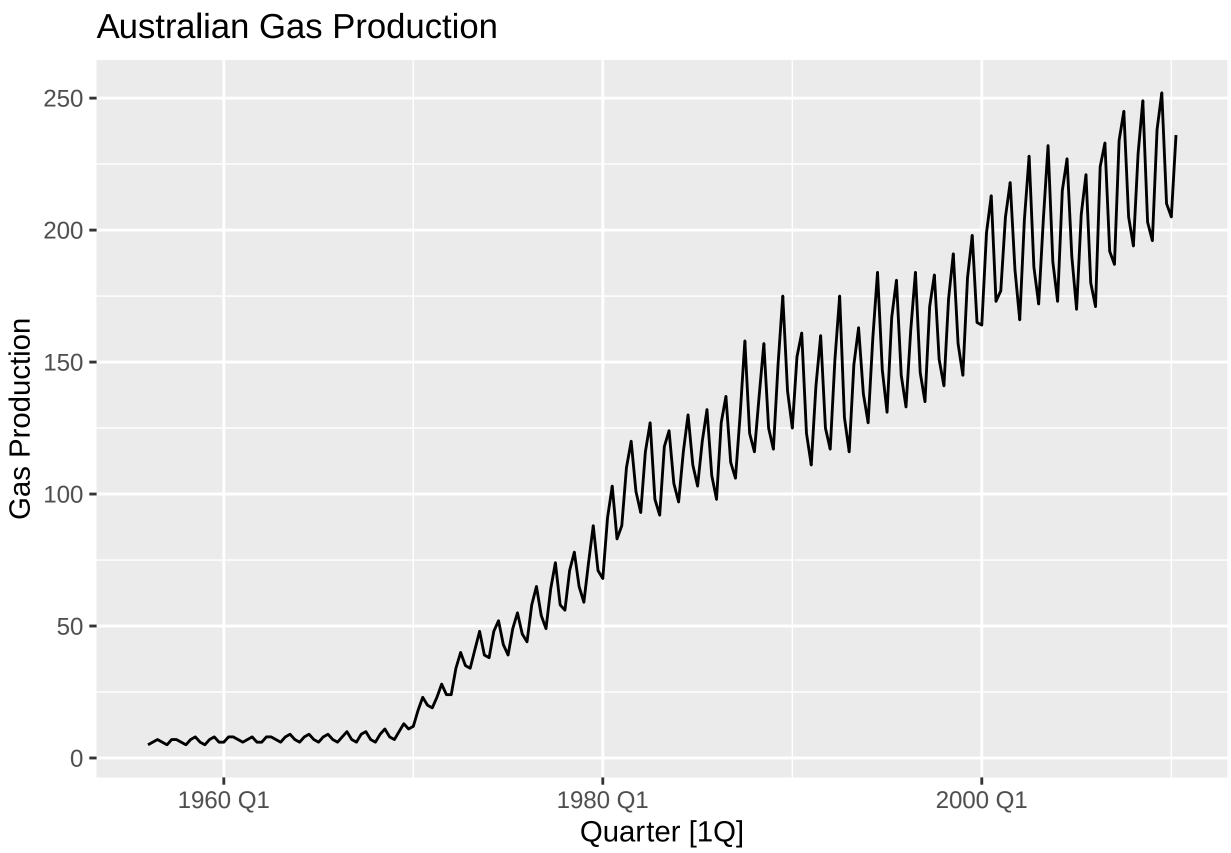
```
# Compute optimal Box-Cox lambda
lambda <- china_gdp %>%
  features(GDP, features = guerrero) %>%
  pull(lambda_guerrero)

# Fit three models
models <- china_gdp %>%
  model(
    basic_ETS = ETS(GDP),
    AAdN = ETS(GDP ~ error("A") + trend("Ad") + season("N")),
    ANN = ETS(GDP ~ error("A") + trend("A") + season("N")),
    boxcox_ETS = ETS(box_cox(GDP, lambda) ~ error("A") + trend("A") + season("N"))
  )

# Forecast 20 years
forecasts <- models %>%
  forecast(h = 20)

forecasts %>%
  autoplot(china_gdp, level = NULL) +
  labs(title = "Comparison of ETS Forecasts for Chinese GDP",
       y = "GDP",
       x = "Year")
```



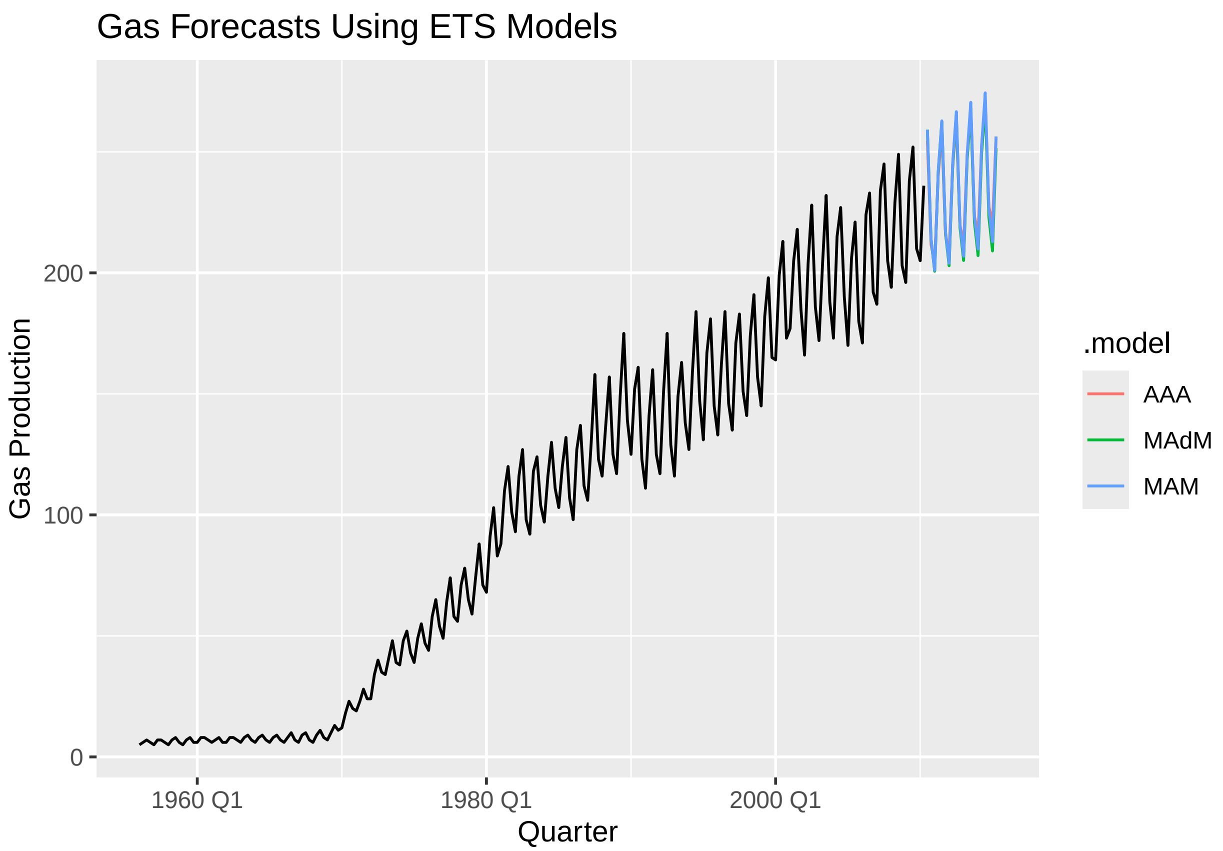
7.Find an ETS model for the Gas data from aus_production and forecast the next few years. Why is multiplicative seasonality necessary here? Experiment with making the trend damped. Does it improve the forecasts?

- Looking at the model fits, the Multiplicative seasonality is a bit better than others because the RMSE is low.

```
gas <- aus_production %>%
  select(Quarter, Gas) %>%
  filter(!is.na(Gas))

gas %>%
  autoplot(Gas) +
  labs(title = "Australian Gas Production", y = "Gas Production")
```



```r
# Fit ETS models
gas_models <- gas %>%
  model(
    AAA = ETS(Gas ~ error("A") + trend("A") + season("A")),
    MAM = ETS(Gas ~ error("M") + trend("A") + season("M")),
    MAdM = ETS(Gas ~ error("M") + trend("Ad") + season("M"))
  )

# Forecast 5 years
gas_forecasts <- gas_models %>%
  forecast(h = "5 years")

gas_forecasts %>%
  autoplot(gas, level = NULL) +
  labs(title = "Gas Forecasts Using ETS Models",
       y = "Gas Production")
```



```
accuracy(gas_models)
```

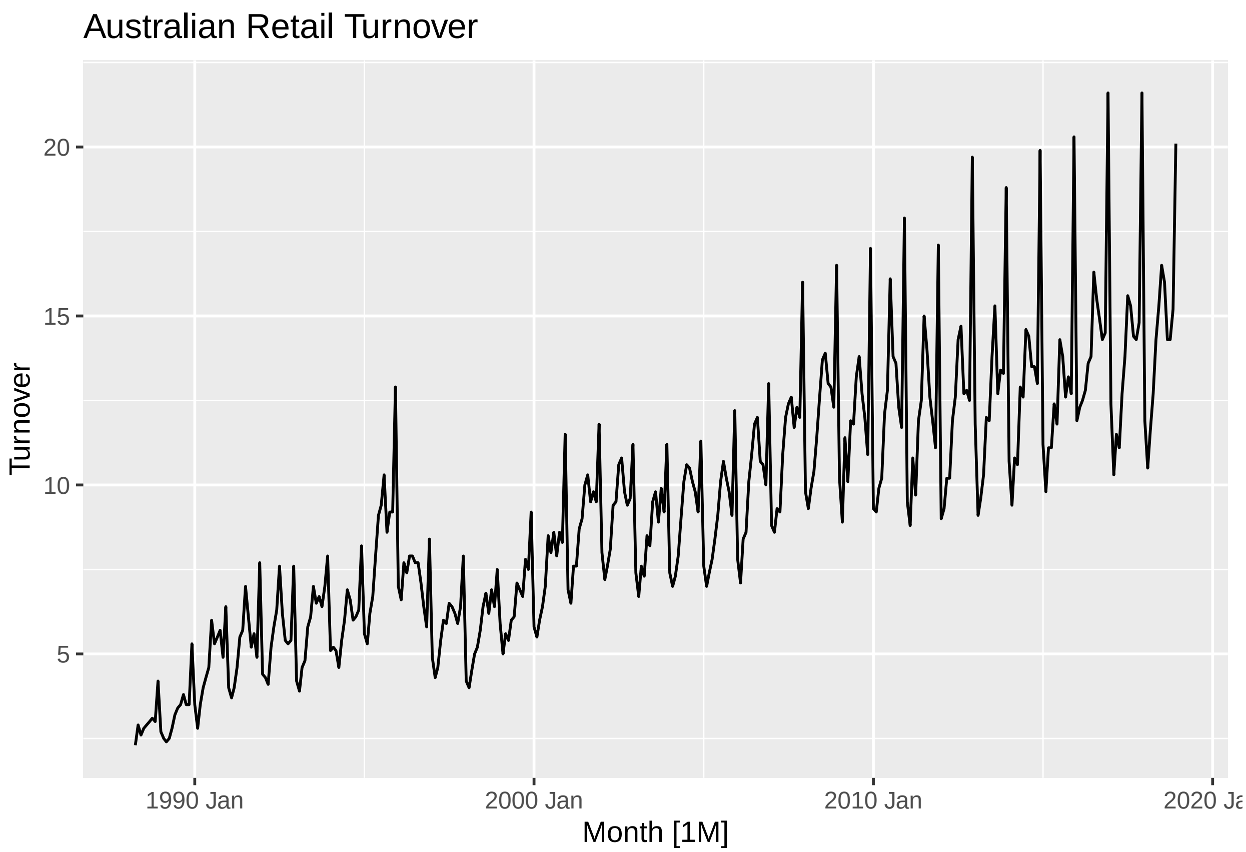
```
## # A tibble: 3 x 10
##   .model .type           ME  RMSE   MAE    MPE  MAPE  MASE RMSSE    ACF1
##   <chr>  <chr>        <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 AAA    Training  0.00525   4.76  3.35 -4.69  10.9  0.600 0.628  0.0772
## 2 MAM    Training -0.115     4.60  3.02  0.199  4.08 0.542 0.606 -0.0131
## 3 MAdM   Training -0.00439   4.59  3.03  0.326  4.10 0.544 0.606 -0.0217
```

8.Recall your retail time series data (from Exercise 7 in Section 2.10).

- a.Why is multiplicative seasonality necessary for this series?
- There is an exhibited season pattern and multiplicative seasonality is needed due to the increase in the variance of the seasonality overtime.

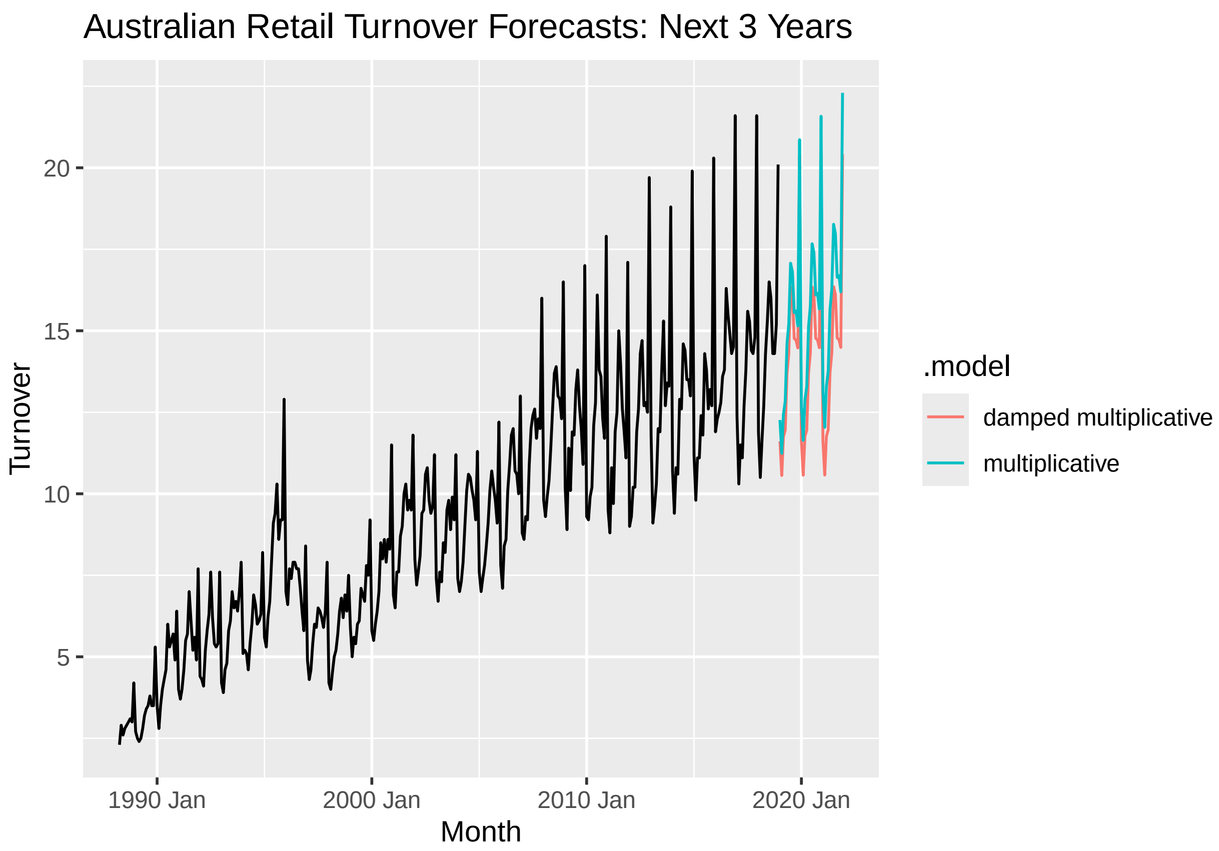
```
set.seed(12345678)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

```
myseries %>%
  autoplot(Turnover) +
  labs(title = 'Australian Retail Turnover')
```



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

```
fit_myseries <- myseries|>
  model("multiplicative" = ETS(Turnover ~ error("M") + trend("A") + season("M")),
        "damped multiplicative" = ETS(Turnover ~ error("M") + trend("Ad") + season("M")))

fc_myseries <- fit_myseries |>
  forecast(h = "3 years")

fc_myseries |>
  autoplot(myseries, level=NULL) +
  labs(title = "Australian Retail Turnover Forecasts: Next 3 Years")
```



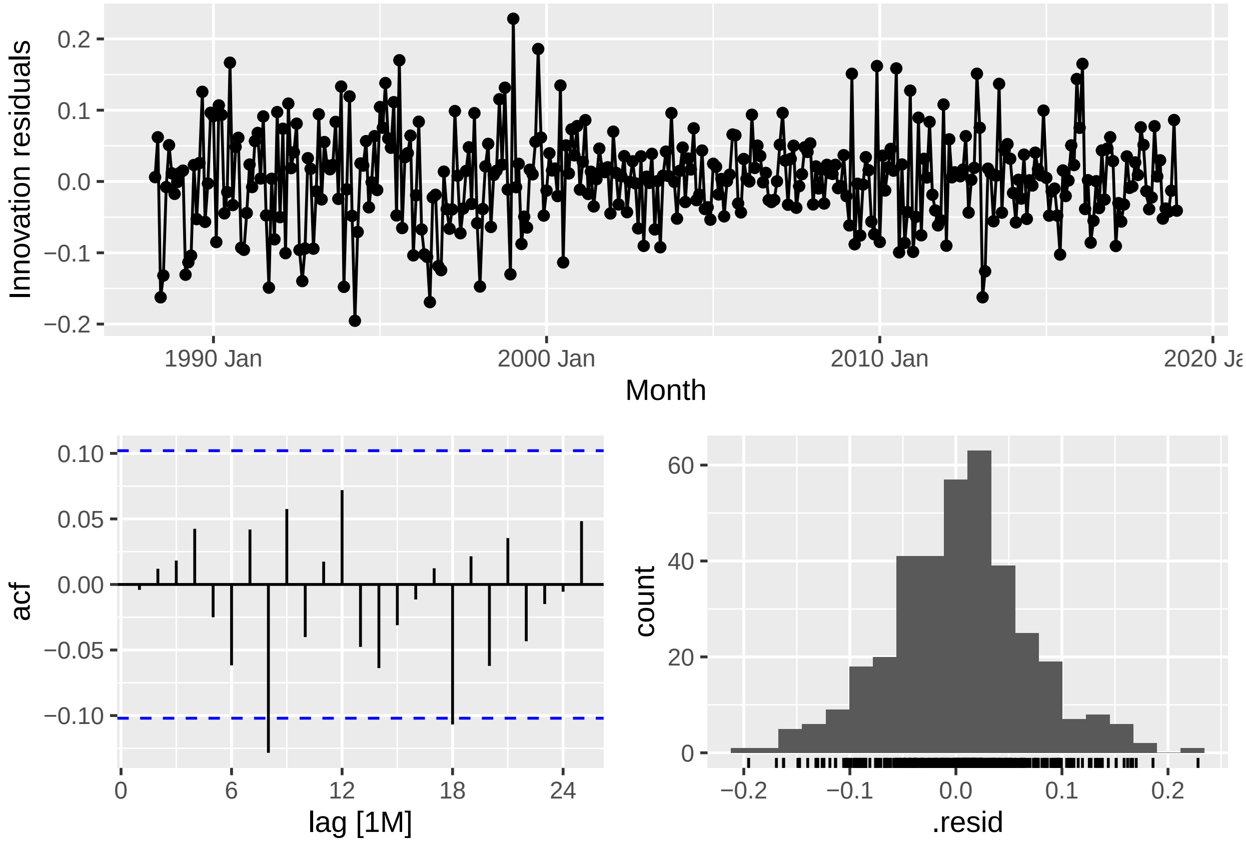
- c.Compare the RMSE of the one-step forecasts from the two methods. Which do you prefer?
- Compare the RMSE here is no much different. But the model damped multiplicative is slightly better then multiplicative.

```
accuracy(fit_myseries)
```

```
## # A tibble: 2 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~ multi~ Trai~ -0.0128 0.613 0.450 -0.469   5.15 0.513 0.529
## 2 Northern ~ Clothin~ dampe~ Trai~  0.0398 0.616 0.444 -0.0723  5.06 0.507 0.531
## # i 1 more variable: ACF1 <dbl>
```

- d.Check that the residuals from the best method look like white noise.
- It looks like white noise.
- In residuals over time plot, there is no strong patterns, no clear tends.
- In ACF plot: Most bars are within the blue dash lines, only two minor lags.
- In histogram plot: Looks like normally distributed, even the plot slight skew, but still good.

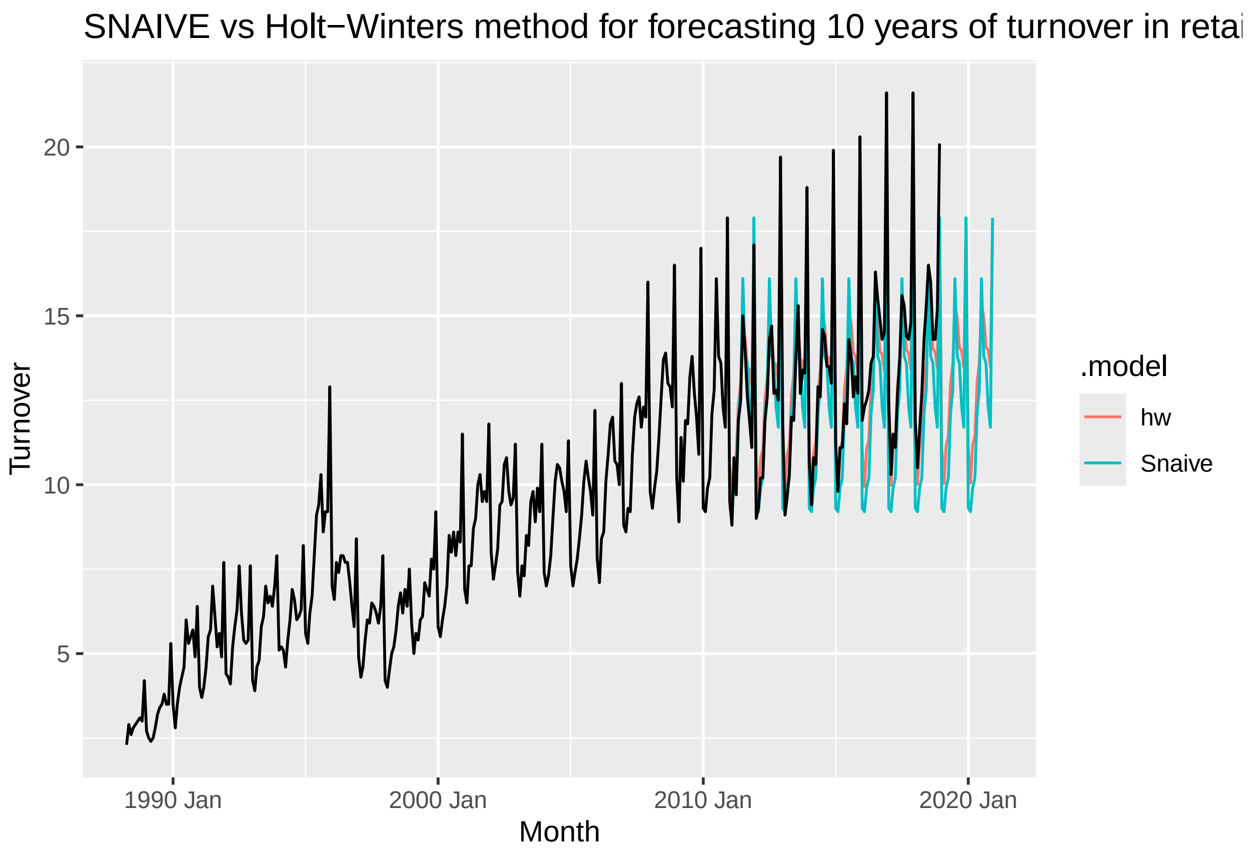
```
best_mod <- myseries %>%
  model(ETS(Turnover ~ error("M") + trend("Ad") + season("M")))

best_mod %>%
  gg_tsresiduals()
```



- 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?
- Holt-Winters method is better. Holt-Winters method get RMSE 0.5185084, and Snaive get RMSE 1.2137305.

```
myseries_train <- myseries %>%
  filter(year(Month) < 2011)
```

```
my_fit_3 <- myseries_train %>%
  model(
    hw = ETS(Turnover ~ error('M') + trend('Ad') + season('M')),
    Snaive = SNAIVE(Turnover)
  )

my_fc_3 <- my_fit_3 %>%
  forecast(h = '10 year')

my_fc_3 %>%
  autoplot(myseries, level = NULL) +
  ggtitle('SNAIVE vs Holt-Winters method for forecasting 10 years of turnover in retail')
```



```
accuracy(my_fit_3)
```

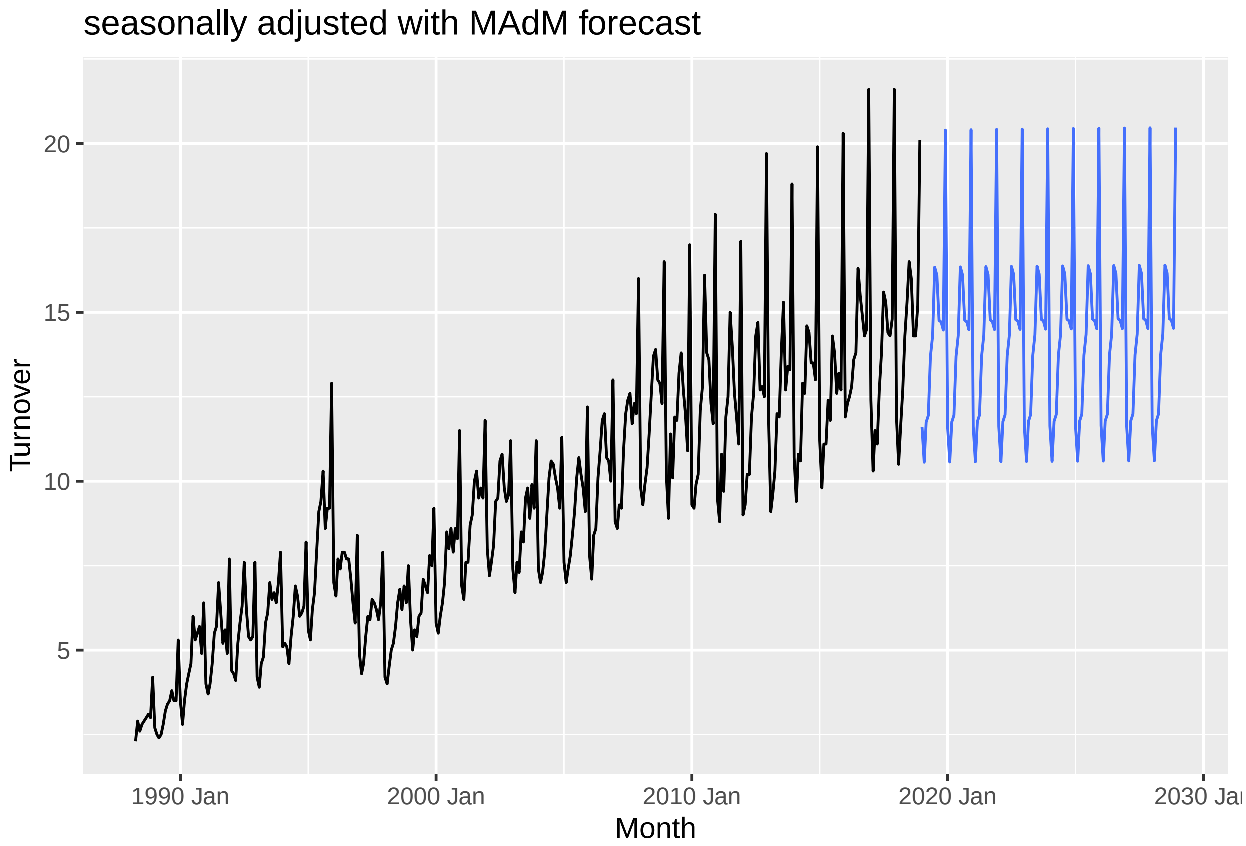
```
## # A tibble: 2 x 12
##   State  Industry .model .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>  <chr>    <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 North~ Clothin~ hw     Trai~ 0.0357 0.519 0.392 0.158  5.34 0.428 0.427 0.0233
## 2 North~ Clothin~ Snaive 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?

- After STL decomposition applied to the Box-Cox transformed series, the RMSE number is 0.6155067. It is larger than my best previous forecasts, it means my best previous forecasts is better.

```
#find the best lambda:
lambda <- myseries %>%
  features(Turnover, features = guerrero) %>%
  pull(lambda_guerrero)
#box cox to STL decomposition (chapter 3.6)
stl_model <- myseries %>%
  model(
    STL = STL(box_cox(Turnover, lambda) ~ trend() + season(window = "periodic"))
  ) %>%
  components()
```

```
#Apply ETS methods to the transformed data
ets_4<- myseries %>%
  model(
    MAdA = ETS(Turnover ~ error('M') + trend('Ad')+season('M'))
  )

#forecast for 10 years
fc_4<- ets_4 %>%
  forecast(h = '10 year')

fc_4 %>%
  autoplot(myseries, level = NULL)+
  labs(title='seasonally adjusted with MAdM forecast')
```



```
accuracy(ets_4)
```

```
## # 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 T~ Clothin~ MAdA   Trai~ 0.0398 0.616 0.444 -0.0723  5.06 0.507 0.531
## # i 1 more variable: ACF1 <dbl>
```