DATA 624: PREDICTIVE ANALYTICS HW5

Gabriel Campos

Last edited March 02, 2024

```
library(fpp3)
library(mlbench)
library(dplyr)
library(ggplot2)
library(tsibble)
library(tidyr)
library(corrplot)
library(cowplot)
library(psych)
library(MASS)
library(gridExtra)
library(tidyr)
library(stringr)
```

Introduction

Do exercises 8.1, 8.5, 8.6, 8.7, 8.8, 8.9 in Hyndman. Please submit both the link to your Rpubs and the .pdf file with your run code.

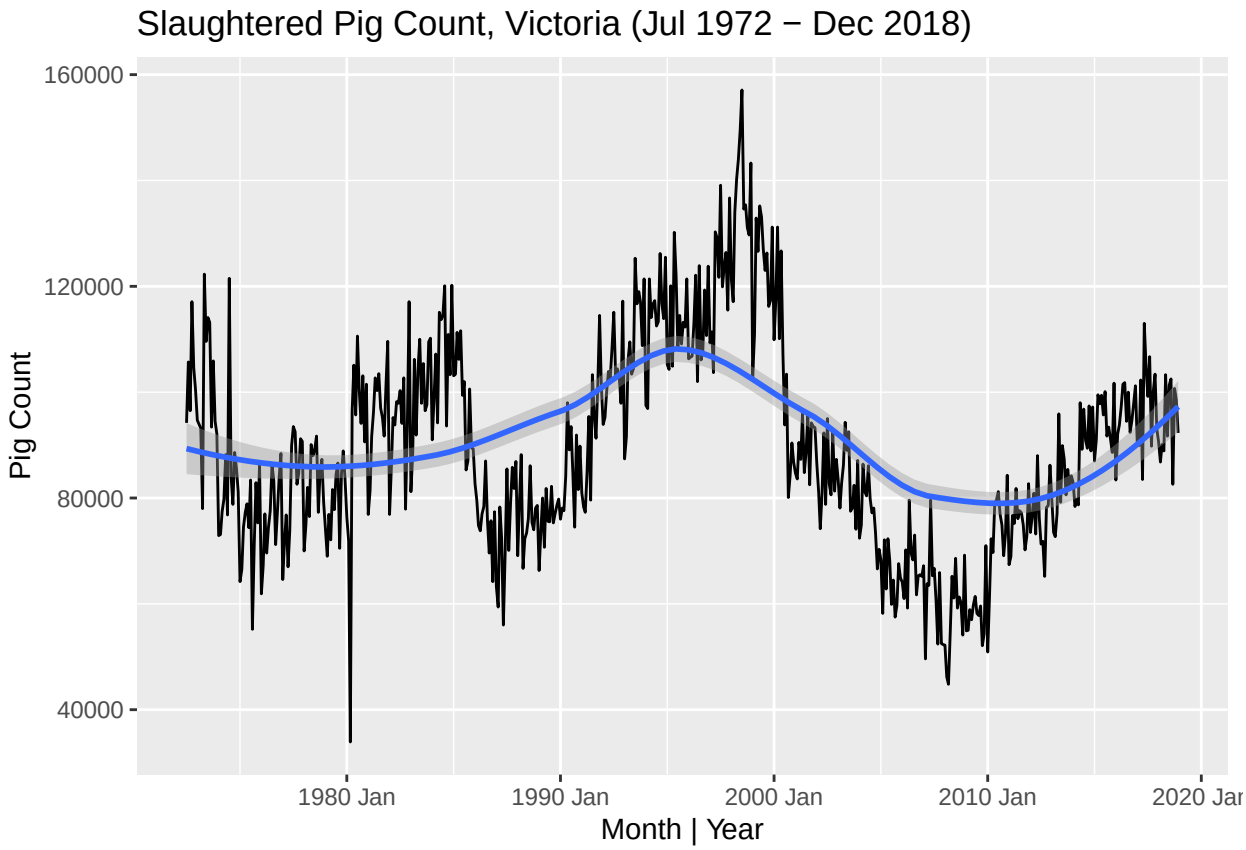
8.1

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

```
pigs_df<- aus_livestock%>%
            filter(str_detect(Animal,"Pigs"),str_detect(State,"Victoria"))
```

```
pigs_df%>%
  autoplot(Count)+
    labs(y="Pig Count",
         x="Month | Year",
         title = "Slaughtered Pig Count, Victoria (Jul 1972 - Dec 2018)")+
  geom_smooth(formula = y ~ x)
```

```
## `geom_smooth()` using method = 'loess'
```



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.

Error Trend and Seasonality or Exponential Smoothing (ETS) Model:

8.1 Example: Algerian export

```
      # Estimate parameters
      fit <- algeria_economy |>
        model(ETS(Exports ~ error("A") + trend("N") + season("N")))
      fc <- fit |>
        forecast(h = 5)
```

```
# Estimate parameters
pigs_fit <- pigs_df %>%
  model(ETS(Count ~ error("A") + trend("N") + season("N")))

report(pigs_fit)
```

```
## Series: Count
## Model: ETS(A,N,N)
##   Smoothing parameters:
##     alpha = 0.3221247
##
##   Initial states:
##      l[0]
##  100646.6
```

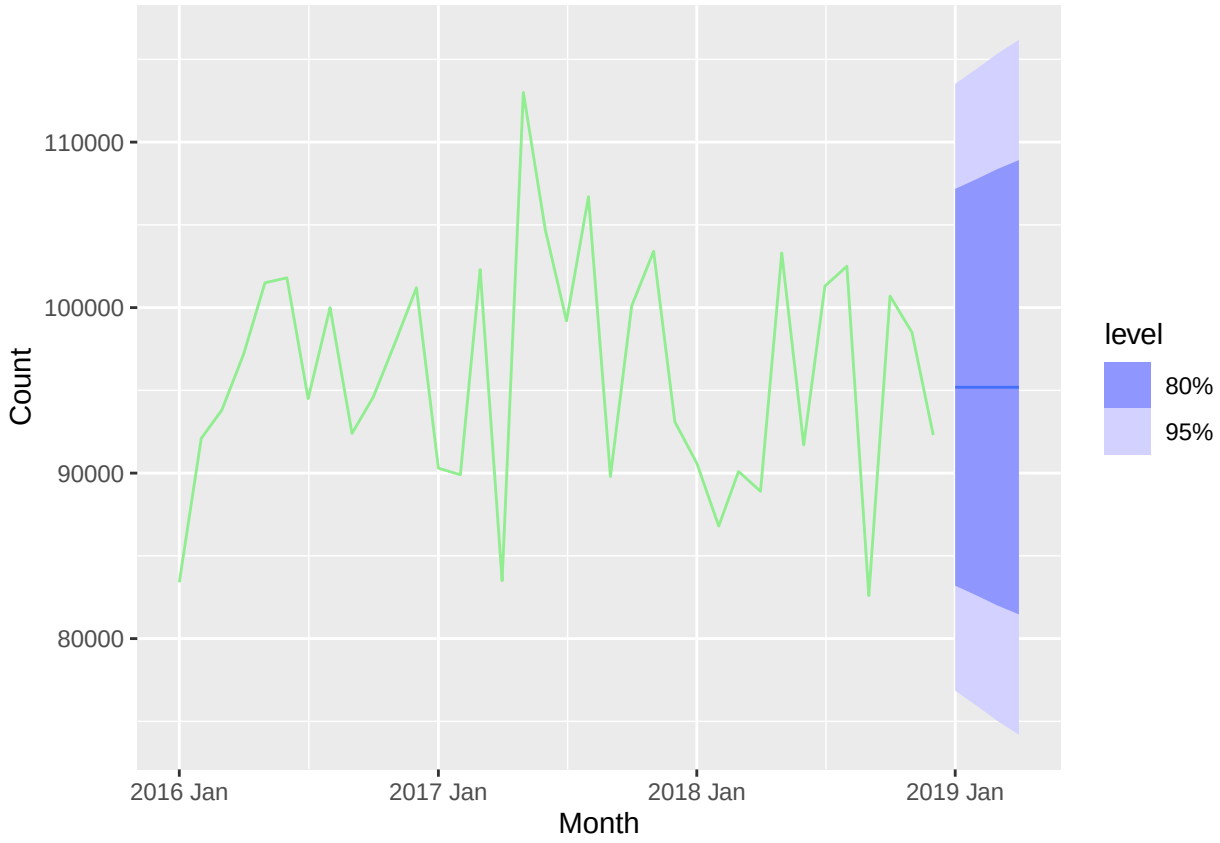
```
##
##   sigma^2:  87480760
##
##      AIC     AICc      BIC
## 13737.10 13737.14 13750.07
```

$OPTIMAL = \alpha = 0.3221247\ \ell = 100646.6$

```
(pigs_fc <- pigs_fit%>%
  forecast(h = 4))
```

```
## # A fable: 4 x 6 [1M]
## # Key:     Animal, State, .model [1]
##   Animal State    .model                              Month             Count  .mean
##   <fct>  <fct>    <chr>                               <mth>            <dist>  <dbl>
## 1 Pigs   Victoria "ETS(Count ~ error(\"A\") +~ 2019 Jan N(95187, 8.7e+07) 95187.
## 2 Pigs   Victoria "ETS(Count ~ error(\"A\") +~ 2019 Feb N(95187, 9.7e+07) 95187.
## 3 Pigs   Victoria "ETS(Count ~ error(\"A\") +~ 2019 Mar N(95187, 1.1e+08) 95187.
## 4 Pigs   Victoria "ETS(Count ~ error(\"A\") +~ 2019 Apr N(95187, 1.1e+08) 95187.
```

```
pigs_fc%>%
    autoplot()+
      geom_line(data=filter(pigs_df, Month >= yearmonth('2016 Jan')),
                aes(x=Month,y=Count),color="lightgreen")
```



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.

```
class(pigs_fc)
```

```
## [1] "fbl_ts"     "tbl_ts"     "tbl_df"     "tbl"        "data.frame"
```

```
pigs_yhat <- pigs_fc$.mean[1]
pigs_aug<- augment(pigs_fit)
pigs_sd<- sd(pigs_aug$.resid)

pigs_upper95<-pigs_yhat+(pigs_sd*1.96)
pigs_lower95<-pigs_yhat-(pigs_sd*1.96)

pigs_hilo<-pigs_fc%>%hilo()

paste0("Lower 95% ",pigs_lower95," Mean ",pigs_yhat, " Upper 95% ",pigs_upper95)
```

```
## [1] "Lower 95% 76871.0124775157 Mean 95186.5574309915 Upper 95% 113502.102384467"
```

```
paste0("While our forecast had the values", pigs_hilo$`95%`[1],
       " with a mean of ", pigs_hilo$.mean[1] )
```

```
## [1] "While our forecast had the values[76854.7888896402, 113518.325972343]95 with a mean of 95186.55
```

The values match if we were going off whole numbers, but using the functions in R provided a greater level of accuracy.

8.5

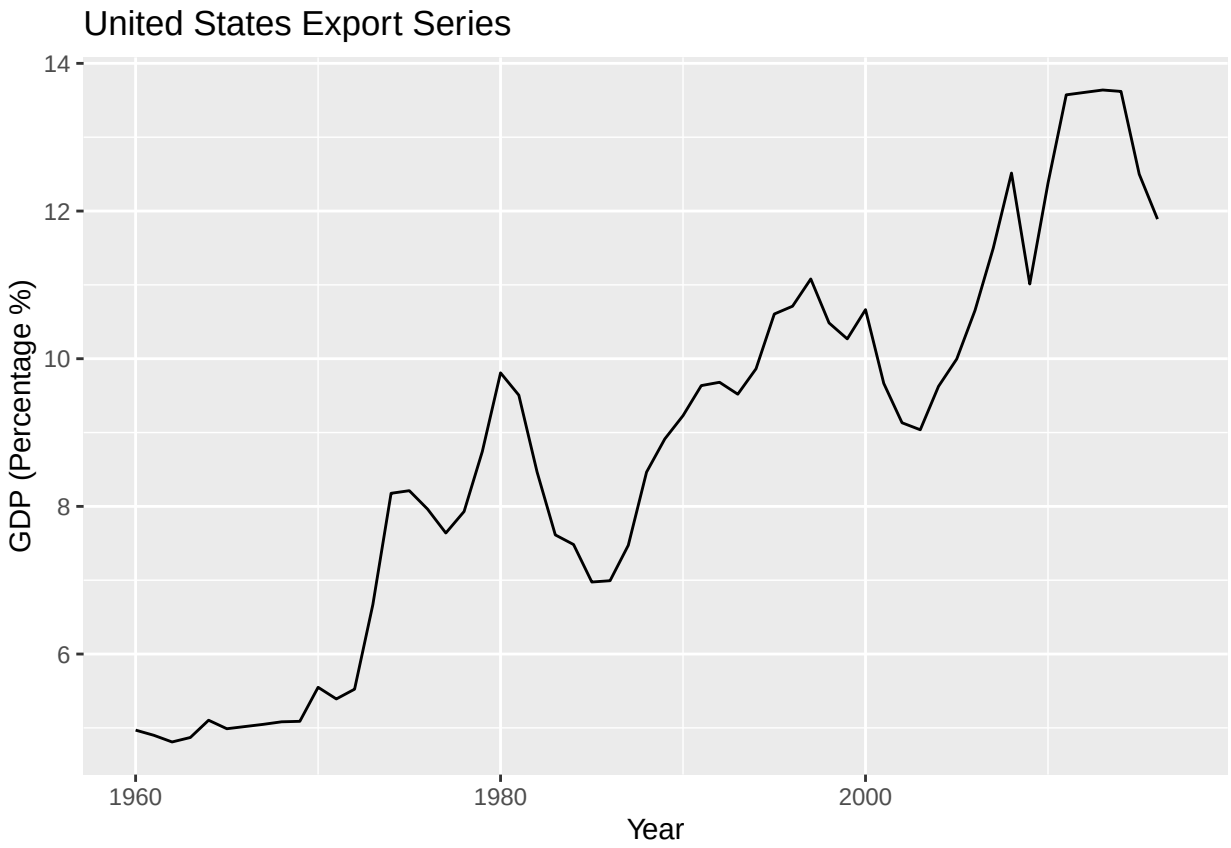
Data set `global_economy` contains the annual Exports from many countries. Select one country to analyse.

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

a.

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

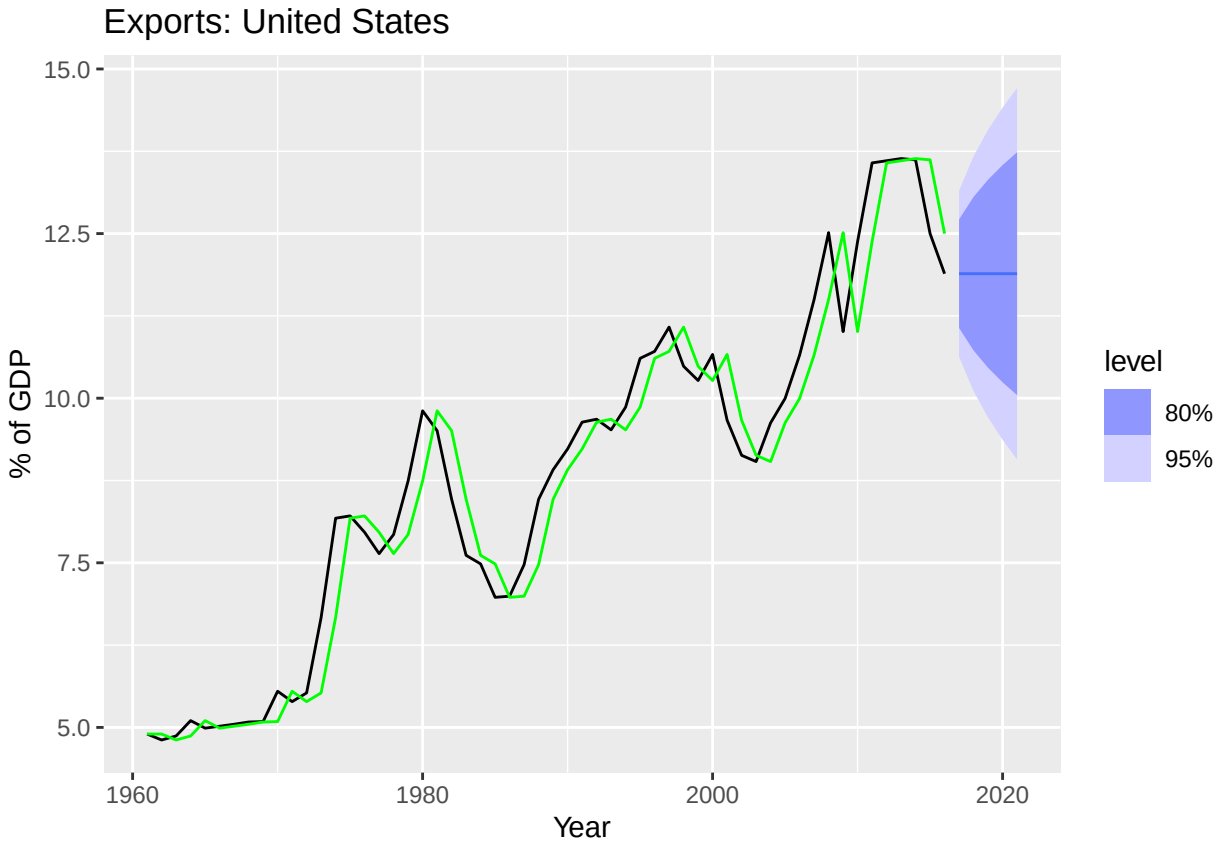
```
usa_df %>% autoplot(Exports)+
  labs(y = "GDP (Percentage %)",x = "Year", title = "United States Export Series")
```



b.

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

```
usa_df <- na.omit(usa_df)

usa_fit <- usa_df %>%
  model(ETS(Exports ~ error("A") + trend("N") + season("N")))


usa_fc <- usa_fit %>%
  forecast(h = 5)

usa_fc %>%
  autoplot(usa_df) +
  geom_line(aes(y = .fitted), col="green",
            data = augment(usa_fit)) +
  labs(y="% of GDP", title="Exports: United States") +
  guides(colour = "none")
```



c.

Compute the RMSE values for the training data.

```
accuracy(usa_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 0.632 0.468  1.35  5.09 0.982 0.991 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.

```
usa_RMSE_ETS<-accuracy(usa_df %>%
  model(
    ANN = ETS (Exports ~ error("A") + trend("N") + season("N")),
    AAN = ETS (Exports ~ error("A") + trend("A") + season("N"))
    ))[["RMSE"]]
usa_RMSE_train<-accuracy(usa_fit)[["RMSE"]]

paste0("Training Model RMSE is ",usa_RMSE_train)
```

```
## [1] "Training Model RMSE is 0.631987696877015"
```

```
paste0("Compared to ")
```

```
## [1] "Compared to "
```

```
paste0(usa_RMSE_ETS)
```

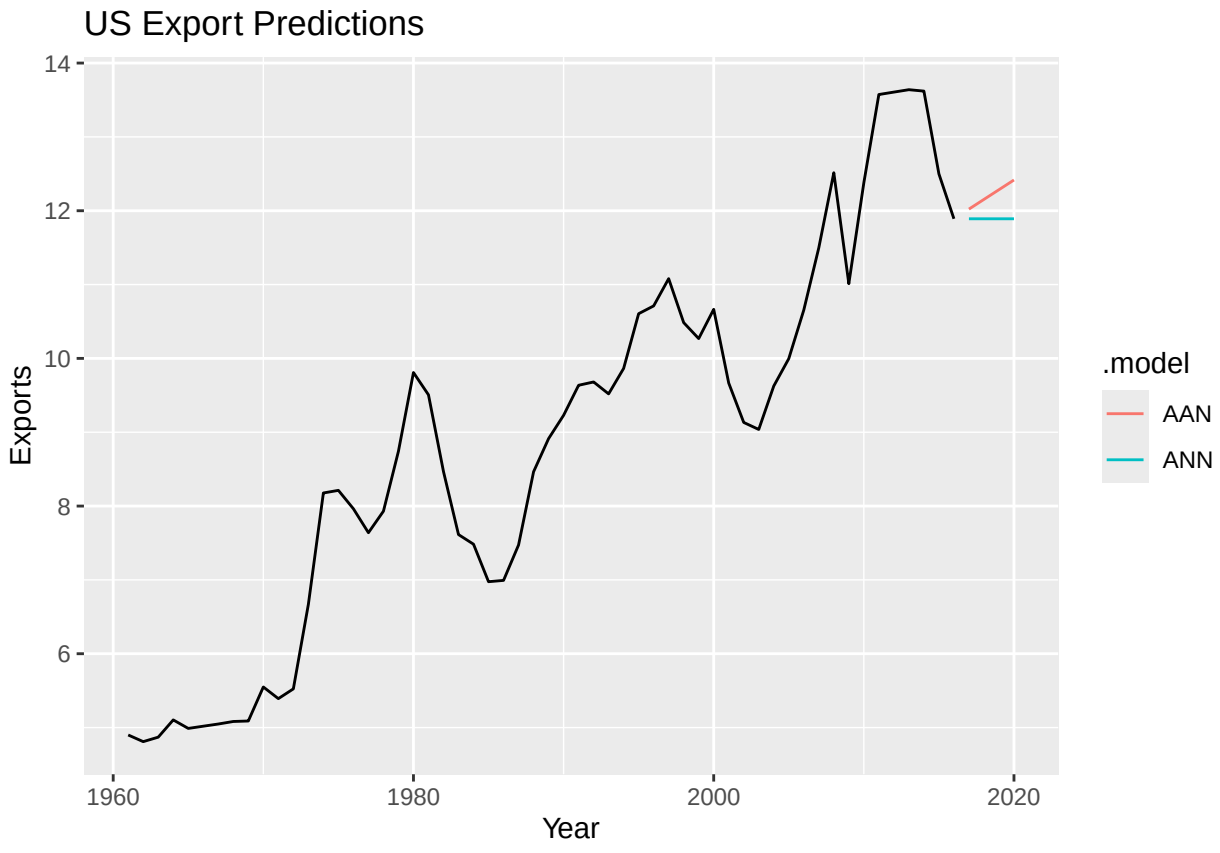
```
## [1] "0.631987696877015" "0.620934771848593"
```

With the AAN model providing a smaller RMSE by about 0.011 than ANN, it suggests a better model lies with AAN.

e.

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

```
usa_df %>%
  model(
    ANN = ETS (Exports ~ error("A") + trend("N") + season("N")),
    AAN = ETS (Exports ~ error("A") + trend("A") + season("N"))
    ) %>%
  forecast(h=4) %>%
  autoplot(usa_df, level=NULL) +
  labs(title="US Export Predictions")
```



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.

```
usa_yhat <- usa_fc$.mean[1]


usa_aug <- augment(usa_fit)


usa_sd <- sd(usa_aug$.resid)


usa_upper95 <- usa_yhat + (usa_sd * 1.96)
usa_lower95 <- usa_yhat - (usa_sd * 1.96)


usa_hilo <- usa_fc %>% hilo()

paste0("Lower 95% ",usa_lower95," Mean ",usa_yhat, " Upper 95% ",usa_upper95)
```

```
## [1] "Lower 95% 10.6654088104703 Mean 11.8906832823187 Upper 95% 13.115957754167"
```

```
paste0("While our forecast had the values", usa_hilo$`95%`[1],
       " with a mean of ", usa_hilo$.mean[1] )
```

```
## [1] "While our forecast had the values[10.6292803146377, 13.1520862499996]95 with a mean of 11.89068
```

Both are accurate up to the first decimal. The method using R vs the manual, accounts for degrees of freedom and has a more precise value for the critical values, but also does gave a greater range.

paste0("Lower 95%",pigs_lower95," Mean ",pigs_yhat," Upper 95% ",pigs_upper95) paste0("While our forecast had the values", pigs_hilo`95.mean[1])

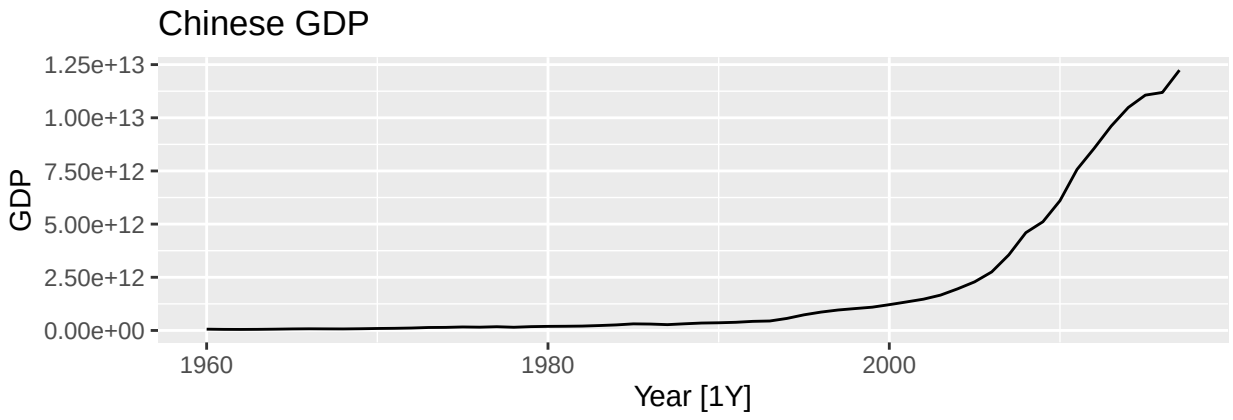
8.6

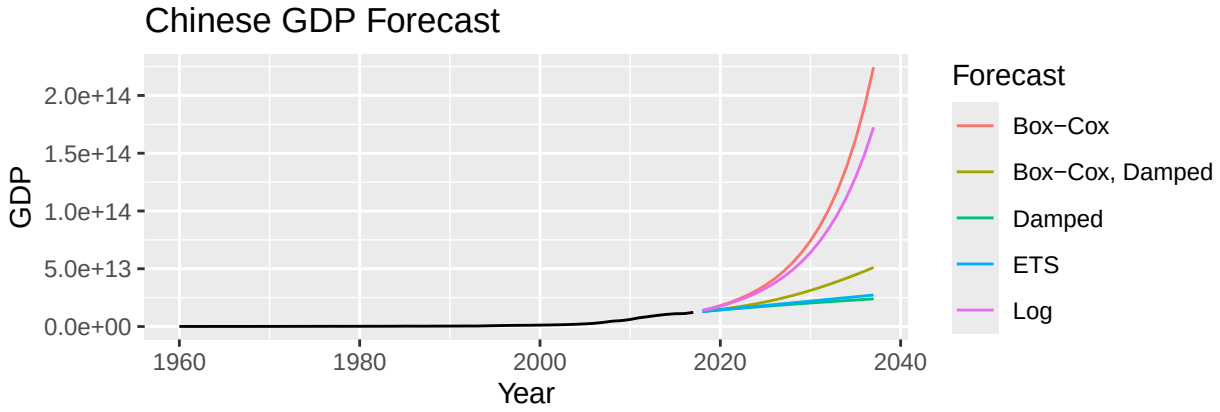
Forecast the Chinese GDP from the global_economy data set using an ETS model. Experiment with the various options in the ETS() function to see how much the forecasts change with damped trend, or with a Box-Cox transformation. Try to develop an intuition of what each is doing to the forecasts.

[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_df <- global_economy %>%
  filter(Country == "China")

china_plot1<-china_df %>% autoplot(GDP) +
  labs(title="Chinese GDP")

china_lambda <- china_df %>%
  features(GDP, features = guerrero) %>%
  pull(lambda_guerrero)

fit_china <- china_df %>%
  model(
    # ETS
    ETS = ETS(GDP),
    # Log Transformation
    `Log` = ETS(log(GDP)),
    # Damped Model
```

```
    `Damped` = ETS(GDP ~ trend("Ad")),
    # Box-Cox Transformation
    `Box-Cox` = ETS(box_cox(GDP, china_lambda)),
    # Damped Model w Box-Cox Transformation
    `Box-Cox, Damped` = ETS(box_cox(GDP, china_lambda) ~ trend("Ad"))
)

china_plot2<-fit_china %>%
  forecast(h="20 years") %>%
  autoplot(china_df, level = NULL)+
  labs(title="Chinese GDP Forecast") +
  guides(colour = guide_legend(title = "Forecast"))

plot_grid(china_plot1,
          china_plot2, nrow = 2)
```





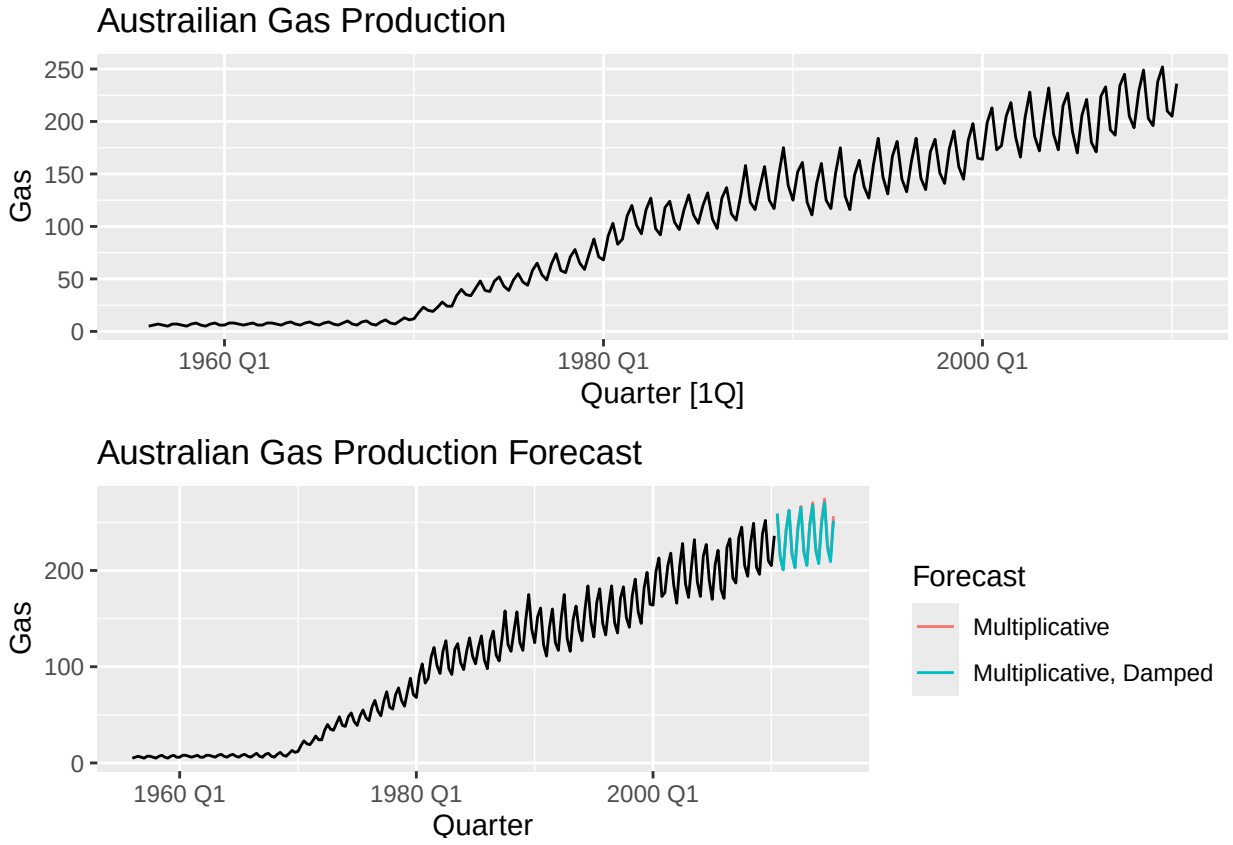
`Damped` and `ETS` show similar continued growth, while `Log` and `Box-Cox` seem to exaggerate its forecast. `Box-Cox`, `Damped` shows slightly more growth than `ETS` and `Damped`.

8.7

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?

*Seasonal variation makes multiplicative seasonality necessary.

```
gas_plot1<-aus_production %>% autoplot(Gas)+
  labs(title="Austrailian Gas Production")

gas_fit <- aus_production %>%
  model(
    # Multiplicative
    Multiplicative = ETS(Gas ~ error("M") + trend("A") + season("M")),
    # Damped multiplicative
    `Multiplicative, Damped` = ETS(Gas ~ error("M") + trend("Ad") + season("M"))
  )
gas_fc <- gas_fit %>% forecast(h = "5 years")

gas_plot2<-gas_fc %>%
  autoplot(aus_production, level = NULL) +
  labs(title="Australian Gas Production Forecast") +
  guides(colour = guide_legend(title = "Forecast"))

plot_grid(gas_plot1,
          gas_plot2, nrow = 2)
```



Very little difference between the two models. About 3.099 difference, meaning either would be accurate or optimal

```
rm(list = ls()[!grepl("^my", ls())])
```

8.8

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

a.

Why is multiplicative seasonality necessary for this series?

B/c there is clear seasonality and peaks on January

b.

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

```
set.seed(123)

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

myfit <- myseries %>%
  model(
    `Holt-Winters' Multiplicative` = ETS(Turnover ~ error("M") + trend("A") +
                                           season("M")),
    `Holt-Winters' Damped Multiplicative` = ETS(Turnover ~ error("M") + trend("Ad") +
                                           season("M"))
  )


myfc <- myfit %>% forecast(h = "5 years")
myfc %>%
  autoplot(myseries, level = NULL) +
  labs(title="Australian Department Stores",
       y="Turnover") +
  guides(colour = guide_legend(title = "Forecast"))
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <80>
```

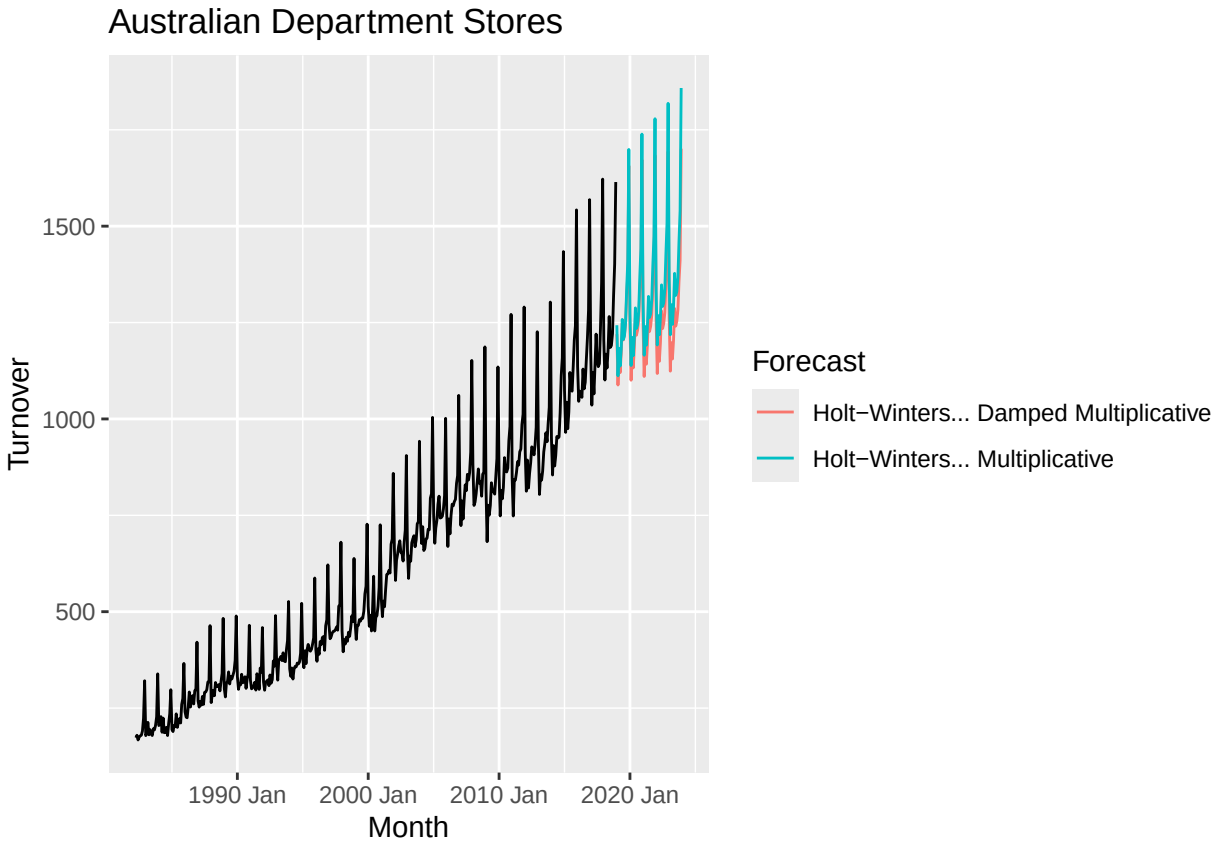
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <99>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <99>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <99>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Damped Multiplicative' in 'mbcsToSbcs':
## dot substituted for <99>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Holt-Winters' Multiplicative' in 'mbcsToSbcs': dot
## substituted for <99>
```



c.

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

```
accuracy(myfit)%>%
  dplyr::select(".model","RMSE")
```

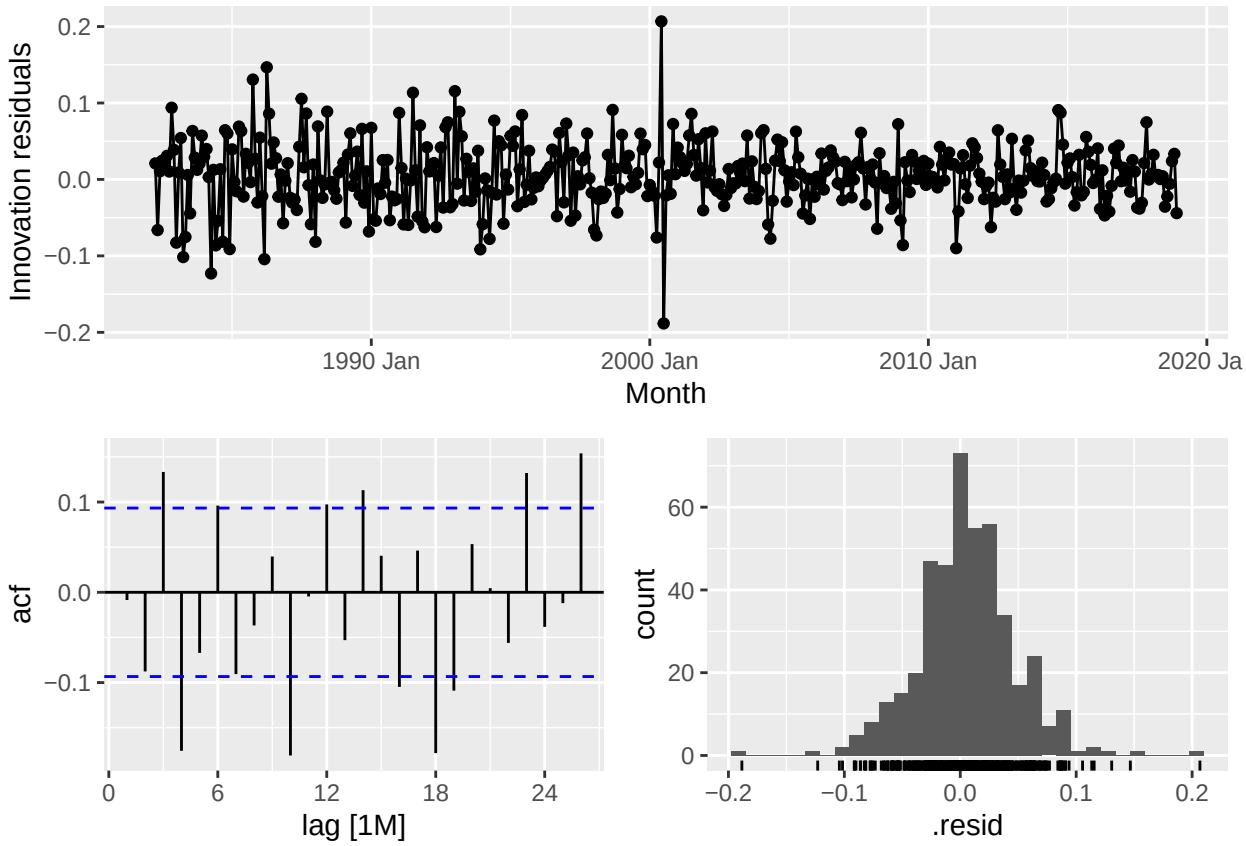
```
## # A tibble: 2 x 2
##   .model                                RMSE
##   <chr>                                <dbl>
## 1 Holt-Winters' Multiplicative          24.3
## 2 Holt-Winters' Damped Multiplicative   23.7
```

The values are pretty close, within 0.62109 of each other, but the `Dampled` has the lower of the two.

d.

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

```
myfit%>%
  dplyr::select("Holt-Winters' Damped Multiplicative")%>%gg_tsresiduals()
```



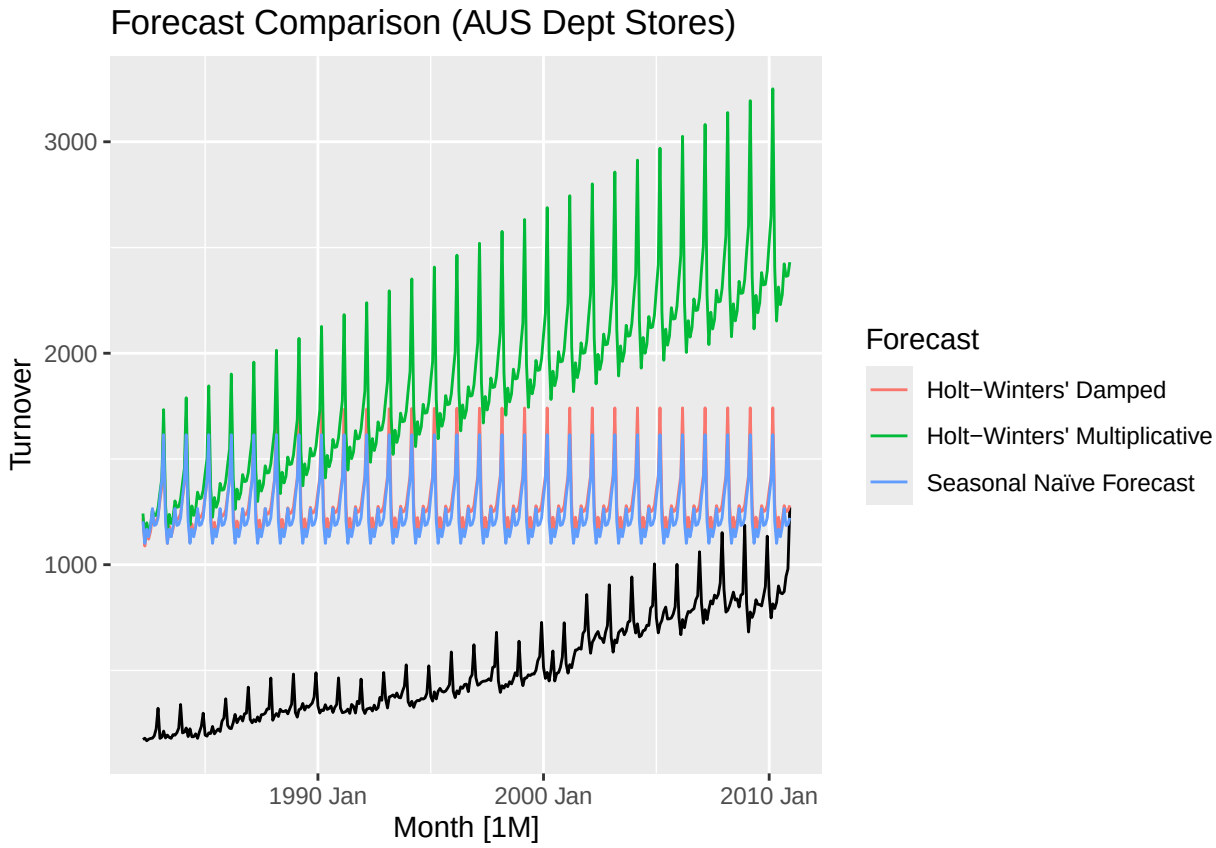
`Holt-Winters' Damped Multiplicative` is not white noise base on the `acf` plot, which has over 5% of the spikes out ousid of the bounds made by dashed lines.

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?

```
mytrain<-myseries%>%
  filter(Month >= yearmonth('2011 Jan'))

# seasonal naïve
myfit2 <- mytrain %>%
  model(
    "Holt-Winters' Damped" = ETS(Turnover ~ error("M") + trend("Ad") +
                                            season("M")),
    "Holt-Winters' Multiplicative" = ETS(Turnover ~ error("M") + trend("A") +
                                                    season("M")),
    "Seasonal Naïve Forecast" = SNAIVE(Turnover)
  )


comparison <- anti_join(myseries, mytrain,
                        by = c("State", "Industry", "Series ID", "Month", "Turnover"))

# Do the forecasting according to comparison data
myfc2 <- myfit2 %>%
```

```
      forecast(comparison)

# plot
autoplot(comparison, Turnover) +
  autolayer(myfc2, level = NULL) +
  guides(colour=guide_legend(title="Forecast")) +
  ggtitle('Forecast Comparison (AUS Dept Stores)')
```




```
accuracy(myfit2)
```

```
## # A tibble: 3 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 Victoria Household ~ Holt-~ Trai~  1.97   26.3  21.1  0.135   2.01 0.393 0.400
## 2 Victoria Household ~ Holt-~ Trai~  0.740  27.0  21.7 -0.0276  2.08 0.405 0.411
## 3 Victoria Household ~ Seaso~ Trai~ 47.0    65.7  53.6  4.23    4.90 1     1
## # i 1 more variable: ACF1 <dbl>
```

It appears the Damped model is the best performing, base on the RMSE values.

8.9

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?

```
#find optimal lambda
mylambda <- mytrain %>%
  features(Turnover, features = guerrero) %>%
  pull(lambda_guerrero)

#bc transformed data
ts_bc <- mytrain %>%
  mutate(
    bc_turnover = box_cox(Turnover, mylambda)
  )

# bc transformed model
fit <- ts_bc %>%
  model(
    'Box-Cox STL' = STL(bc_turnover ~ season(window = "periodic"),
              robust = T),
    'Box-Cox ETS' = ETS(bc_turnover)
  )

# best previous model
best_fit <-ts_bc %>%
  model(
    "Holt-Winters' Damped" = ETS(Turnover ~ error("M") + trend("Ad") +
                                                  season("M"))
  )

rbind(accuracy(fit),accuracy(best_fit))
```

```
## # A tibble: 3 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 Victor~ Househo~ Box-C~ Trai~  0.0482   0.417  0.285  0.126  0.702 0.310 0.370
## 2 Victor~ Househo~ Box-C~ Trai~ -0.00279  0.433  0.352 -0.0152 0.862 0.383 0.384
## 3 Victor~ Househo~ Holt-~ Trai~  1.97    26.3   21.1    0.135  2.01  0.393 0.400
## # i 1 more variable: ACF1 <dbl>
```

Based on the Values the Box-Cox ETS is the best performing of the three.