Data624 - Homework5

Amit Kapoor

3/7/2021

Contents

Exercise 7.1

Consider the `pigs` series — the number of pigs slaughtered in Victoria each month.

```
str(pigs)
```

```
##  Time-Series [1:188] from 1980 to 1996: 76378 71947 33873 96428 105084 ...
```

a)

Use the `ses()` function in R to find the optimal values of α and ℓ_0, and generate forecasts for the next four months.

```
# Using ses for pigs
pigs_ses <- ses(pigs, h=4)
```

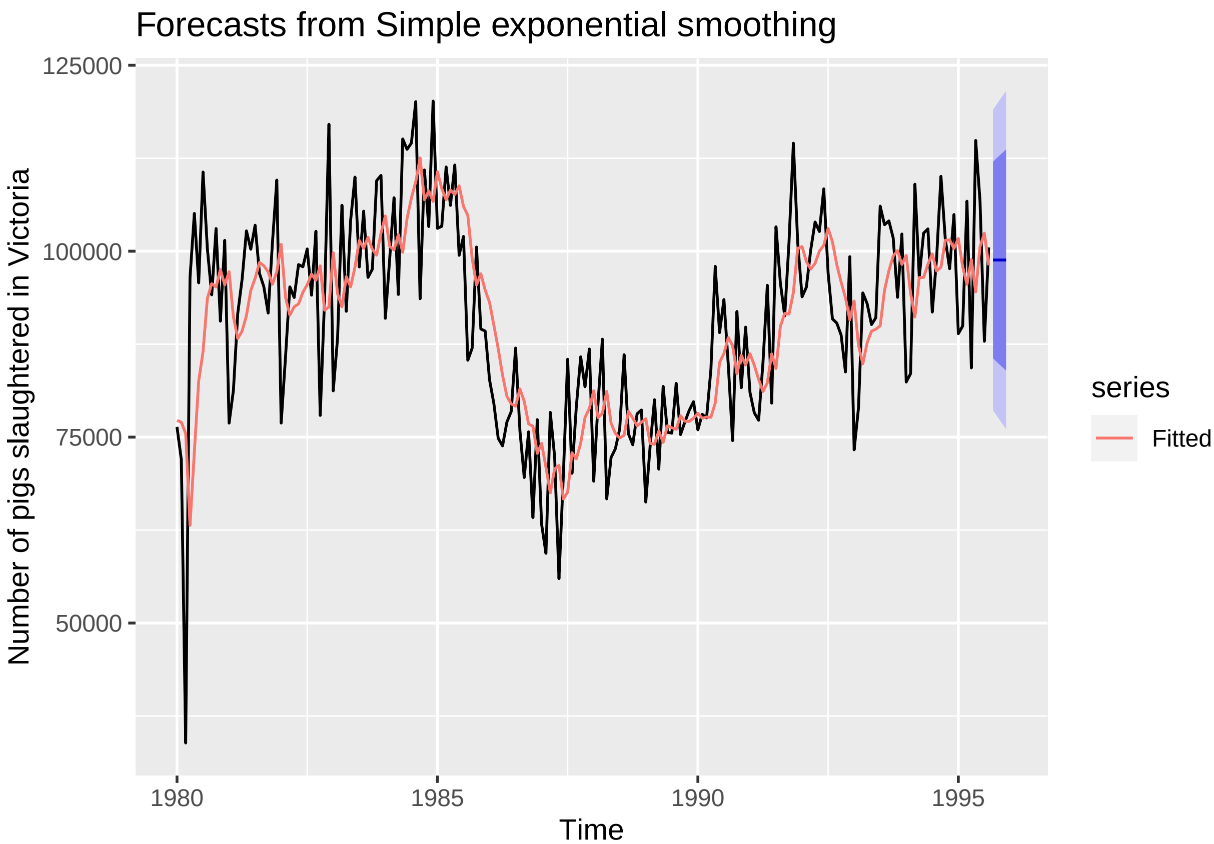
```r
#summary
summary(pigs_ses)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = pigs, h = 4)
##
##   Smoothing parameters:
##     alpha = 0.2971
##
##   Initial states:
##     l = 77260.0561
##
##   sigma:  10308.58
##
##      AIC     AICc      BIC
## 4462.955 4463.086 4472.665
##
## Error measures:
##                    ME    RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 385.8721 10253.6 7961.383 -0.922652 9.274016 0.7966249 0.01282239
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Sep 1995       98816.41 85605.43 112027.4 78611.97 119020.8
## Oct 1995       98816.41 85034.52 112598.3 77738.83 119894.0
## Nov 1995       98816.41 84486.34 113146.5 76900.46 120732.4
## Dec 1995       98816.41 83958.37 113674.4 76092.99 121539.8
```

Above summary shows the the optimal values of α and ℓ_0 are 0.2971 and 77260.0561 respectively. Using these values forecast is generated for next 4 months.

Next plot shows the forecast from simple exponential smoothing. Also one-step-ahead fitted values are plotted with the data over the period.

```r
autoplot(pigs_ses) +
  autolayer(fitted(pigs_ses), series="Fitted") +
  ylab("Number of pigs slaughtered in Victoria")
```



b)

Compute a 95% prediction interval for the first forecast using $\hat{y} \pm 1.96\sigma$ where σ is the standard deviation of the residuals. Compare your interval with the interval produced by R.

```
# 95% prediction interval for the first forecas
sd <- sd(residuals(pigs_ses))
ci95 <- c(lower = pigs_ses$mean[1] - 1.96*sd, upper = pigs_ses$mean[1] + 1.96*sd)
ci95
```

```
##     lower     upper
##  78679.97 118952.84
```

```
# By R
ci95_R <- c(pigs_ses$lower[1, "95%"], pigs_ses$upper[1, "95%"])
names(ci95_R) <- c("lower", "upper")
ci95_R
```

```
##     lower     upper
##  78611.97 119020.84
```

It appears the 95% prediction interval calculated by R is a little wider than the one given by the formula.

Exercise 7.5

Data set `books` contains the daily sales of paperback and hardcover books at the same store. The task is to forecast the next four days' sales for paperback and hardcover books.

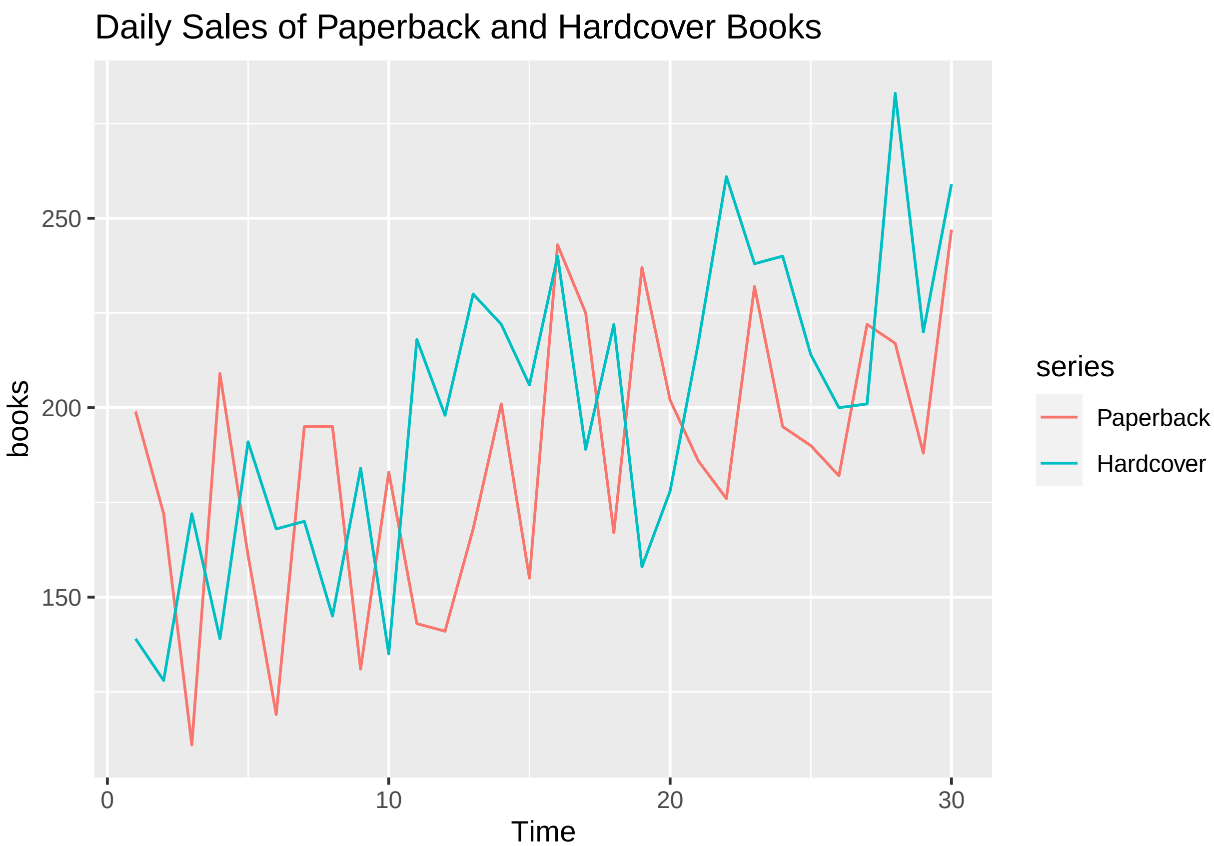
```r
head(books)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
##   Paperback Hardcover
## 1       199       139
## 2       172       128
## 3       111       172
## 4       209       139
## 5       161       191
## 6       119       168
```

a)

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

```r
# plot series
autoplot(books) +
  labs(title = "Daily Sales of Paperback and Hardcover Books")
```

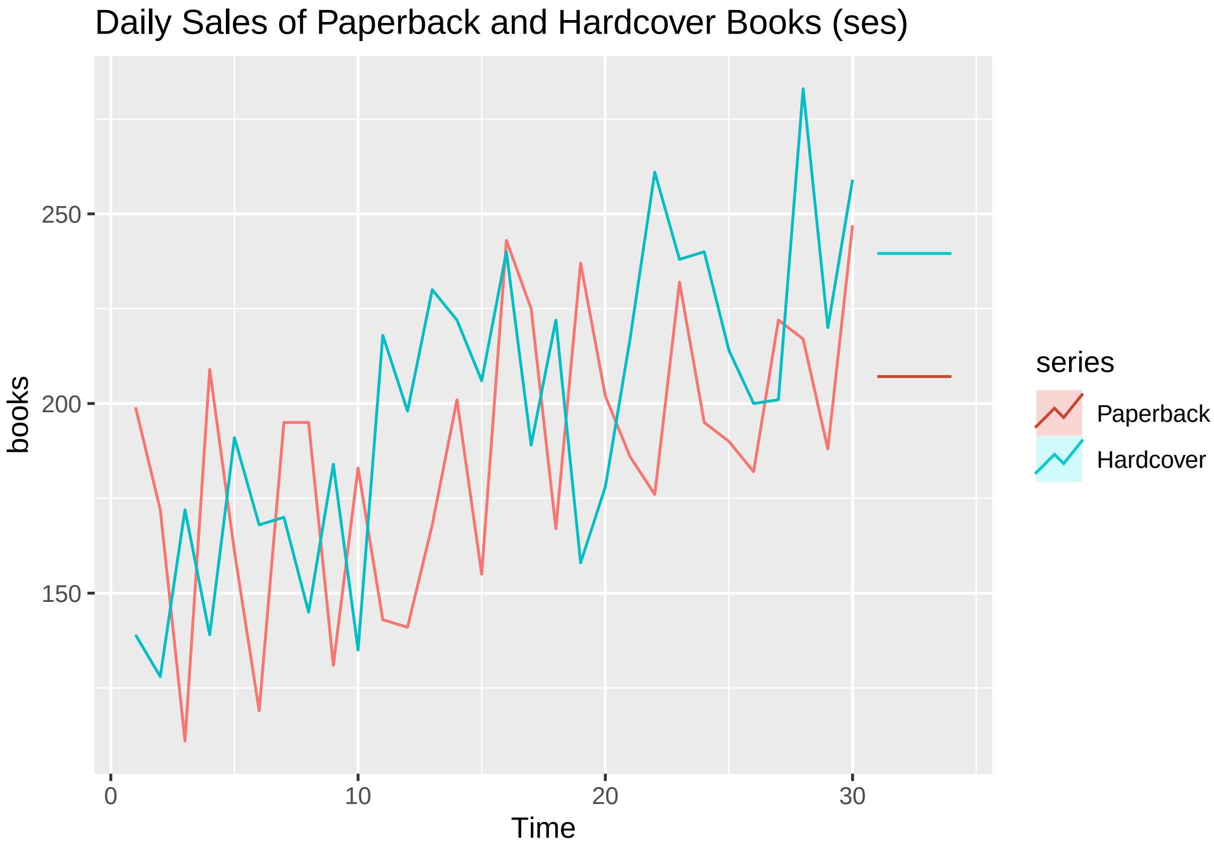


The series has an upward trend but don't see any seasonality or cyclicity in the plot. Also its only a 30 days of data so difficult to speak about seasonality. Another observation is hardcover sales in better than paperback.

b)

Use the `ses()` function to forecast each series, and plot the forecasts.

```r
# Using ses for books
pb_ses <- ses(books[, 'Paperback'], h=4)
hc_ses <- ses(books[, 'Hardcover'], h=4)

autoplot(books) +
  autolayer(pb_ses, series="Paperback", PI=FALSE) +
  autolayer(hc_ses, series="Hardcover", PI=FALSE) +
  labs(title = "Daily Sales of Paperback and Hardcover Books (ses)")
```



The simple exponential smoothing plot above shows flat forecast and doesnt appear to capture upward trend.

c)

Compute the RMSE values for the training data in each case.

```r
# RMSE for paperback
round(accuracy(pb_ses)[2], 2)
```

```
## [1] 33.64
```

```r
# RMSE for hardcover
round(accuracy(hc_ses)[2], 2)
```

```
## [1] 31.93
```

RMSE of hardcover for the training data is slightly better than of paperback.

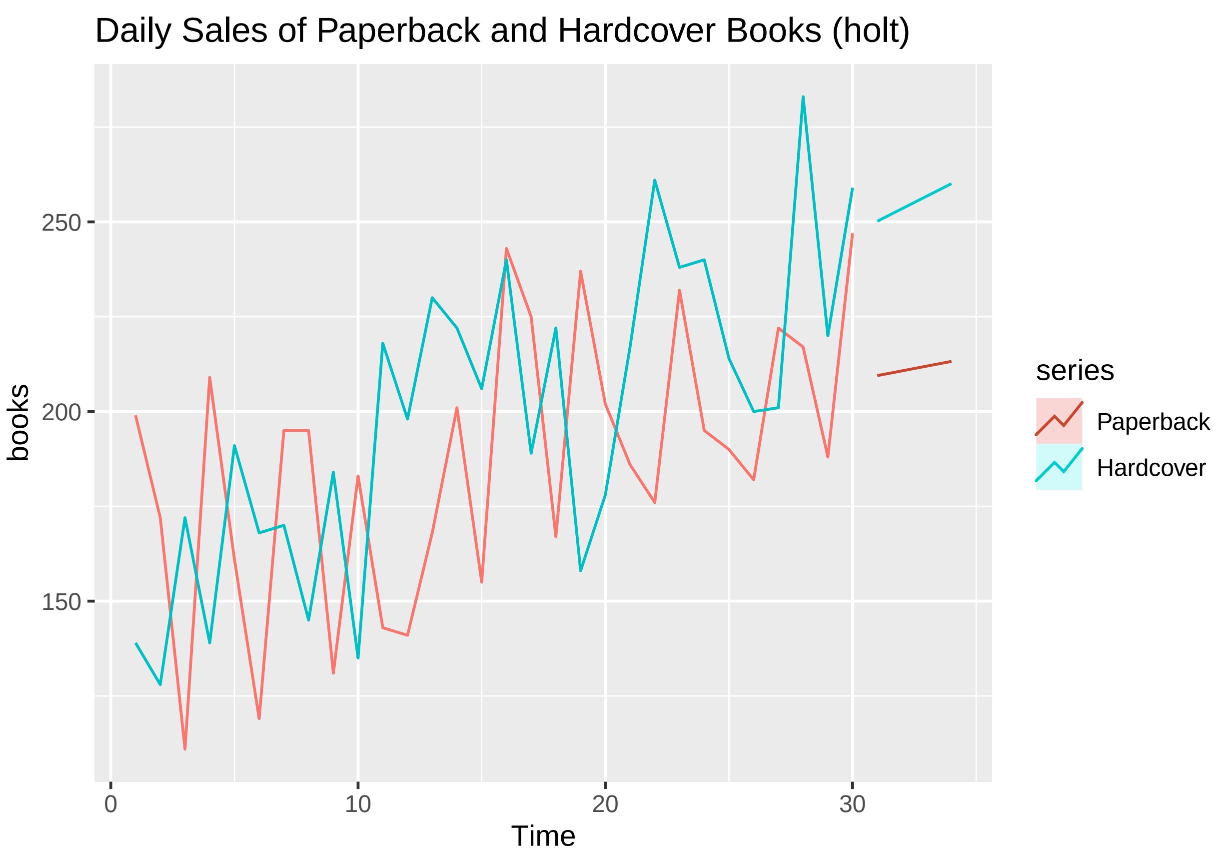
Exercise 7.6

We will continue with the daily sales of paperback and hardcover books in data set books.

a)

Apply Holt's linear method to the `paperback` and `hardback` series and compute four-day forecasts in each case.

```
pb_holt <- holt(books[, 'Paperback'], h=4)
hc_holt <- holt(books[, 'Hardcover'], h=4)

autoplot(books) +
  autolayer(pb_holt, series="Paperback", PI=FALSE) +
  autolayer(hc_holt, series="Hardcover", PI=FALSE) +
  labs(title = "Daily Sales of Paperback and Hardcover Books (holt)")
```



Holt's linear forecast seems better as it is able to capture the upward trend of time series.

b)

Compare the RMSE measures of Holt's method for the two series to those of simple exponential smoothing in the previous question. (Remember that Holt's method is using one more parameter than SES.) Discuss the merits of the two forecasting methods for these data sets.

```
# RMSE for paperback - holt
round(accuracy(pb_holt)[2], 2)
```

```
## [1] 31.14
```

```r
# RMSE for hardcover - holt
round(accuracy(hc_holt)[2], 2)
```

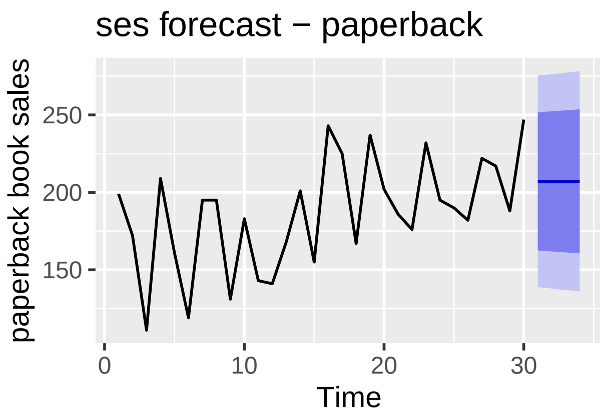
```
## [1] 27.19
```

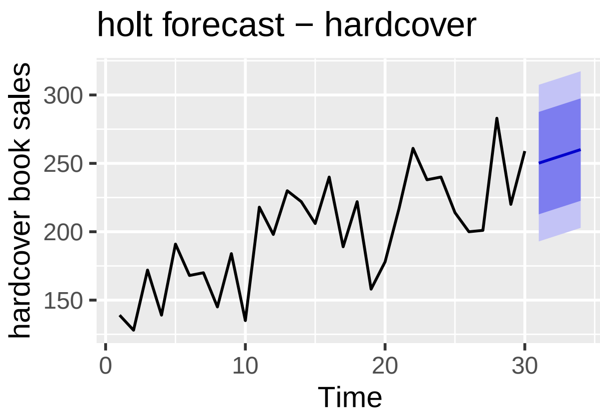
The RMSEs for paperback and hardcover books sale are improved using holt's linear method as compared from simple exponential smoothing method. It happens since holt extended simple exponential smoothing to allow the forecasting of data with a trend and we see in above plot of holts, capturing upward trend.

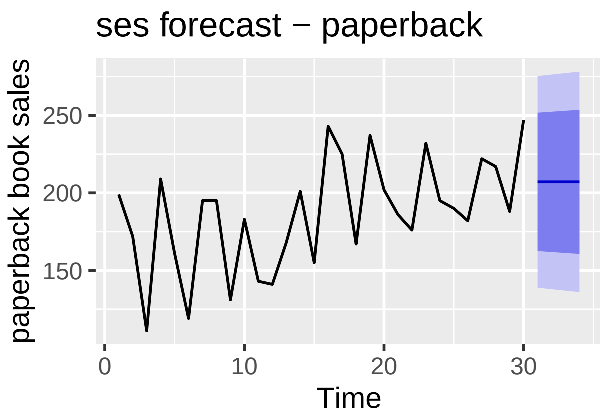
c)

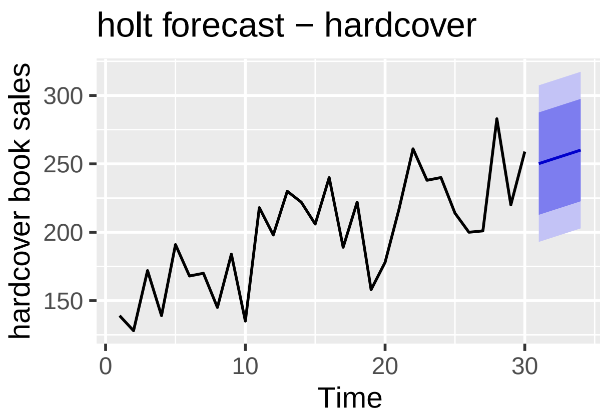
Compare the forecasts for the two series using both methods. Which do you think is best?

```r
# ses and holt comparison for paperback
s1 <- autoplot(pb_ses) +
  ylab("paperback book sales") +
  labs(title = "ses forecast - paperback")
h1 <- autoplot(pb_holt) +
  ylab("paperback book sales") +
  labs(title = "holt forecast - paperback")

# ses and holt comparison for hardcover
s2 <- autoplot(hc_ses) +
  ylab("hardcover book sales") +
  labs(title = "ses forecast - hardcover")
h2 <- autoplot(hc_holt) +
  ylab("hardcover book sales") +
  labs(title = "holt forecast - hardcover")

gridExtra::grid.arrange(s1, h2, s1,h2, nrow=2, ncol=2)
```









Compare the forecasts for the two series using the two, Holt's method is better than simple exponential smoothing method sith holt extended simple exponential smoothing to allow the forecasting of data with a trend. The RMSE is smaller for holt method.

d)

Calculate a 95% prediction interval for the first forecast for each series, using the RMSE values and assuming normal errors. Compare your intervals with those produced using `ses` and `holt`.

```
# 95% prediction interval for the first forecast
df <- data.frame(
  Pred_Int = c("Paperback-SES", "Paperback-Holt","Hardcover-SES", "Hardcover-Holt"),
  lower = c(pb_ses$mean[1] - 1.96*accuracy(pb_ses)[2],
            pb_holt$mean[1] - 1.96*accuracy(pb_holt)[2],
            hc_ses$mean[1] - 1.96*accuracy(hc_ses)[2],
            hc_holt$mean[1] - 1.96*accuracy(hc_holt)[2]),
  upper = c(pb_ses$mean[1] + 1.96*accuracy(pb_ses)[2],
            pb_holt$mean[1] + 1.96*accuracy(pb_holt)[2],
            hc_ses$mean[1] + 1.96*accuracy(hc_ses)[2],
            hc_holt$mean[1] + 1.96*accuracy(hc_holt)[2])
)

df
```

```
##         Pred_Int    lower    upper
## 1  Paperback-SES 141.1798 273.0395
## 2 Paperback-Holt 148.4384 270.4951
## 3  Hardcover-SES 176.9753 302.1449
```

```
## 4 Hardcover-Holt 196.8745 303.4733
```

```
df2 <- data.frame(
  Pred_Int = c("Paperback-SES", "Paperback-Holt","Hardcover-SES", "Hardcover-Holt"),
  lower = c(pb_ses$lower[1, "95%"],
            pb_holt$lower[1, "95%"],
            hc_ses$lower[1, "95%"],
            hc_holt$lower[1, "95%"]),
  upper = c(pb_ses$upper[1, "95%"],
            pb_holt$upper[1, "95%"],
            hc_ses$upper[1, "95%"],
            hc_holt$upper[1, "95%"])
)

df2
```

```
##         Pred_Int    lower    upper
## 1  Paperback-SES 138.8670 275.3523
## 2 Paperback-Holt 143.9130 275.0205
## 3  Hardcover-SES 174.7799 304.3403
## 4 Hardcover-Holt 192.9222 307.4256
```

From the interval range above, it is apparent that the interval calculated using RMSE is slightly narrower than from R using holt ans ses methods.

Exercise 7.7

For this exercise use data set `eggs`, the price of a dozen eggs in the United States from 1900–1993. Experiment with the various options in the `holt()` function to see how much the forecasts change with the damped trend, or with a Box-Cox transformation. Try to develop an intuition of what each argument is doing to the forecasts.

[Hint: use `h = 100` when calling `holt()` so you can clearly see the differences between the various options when plotting the forecasts.]

Which model gives the best RMSE?

```
head(eggs)
```

```
## Time Series:
## Start = 1900
## End = 1905
## Frequency = 1
## [1] 276.79 315.42 314.87 321.25 314.54 317.92
```

```
autoplot(eggs) +
  labs(title = "Price of a dozen eggs in the United States from 1900-1993")
```

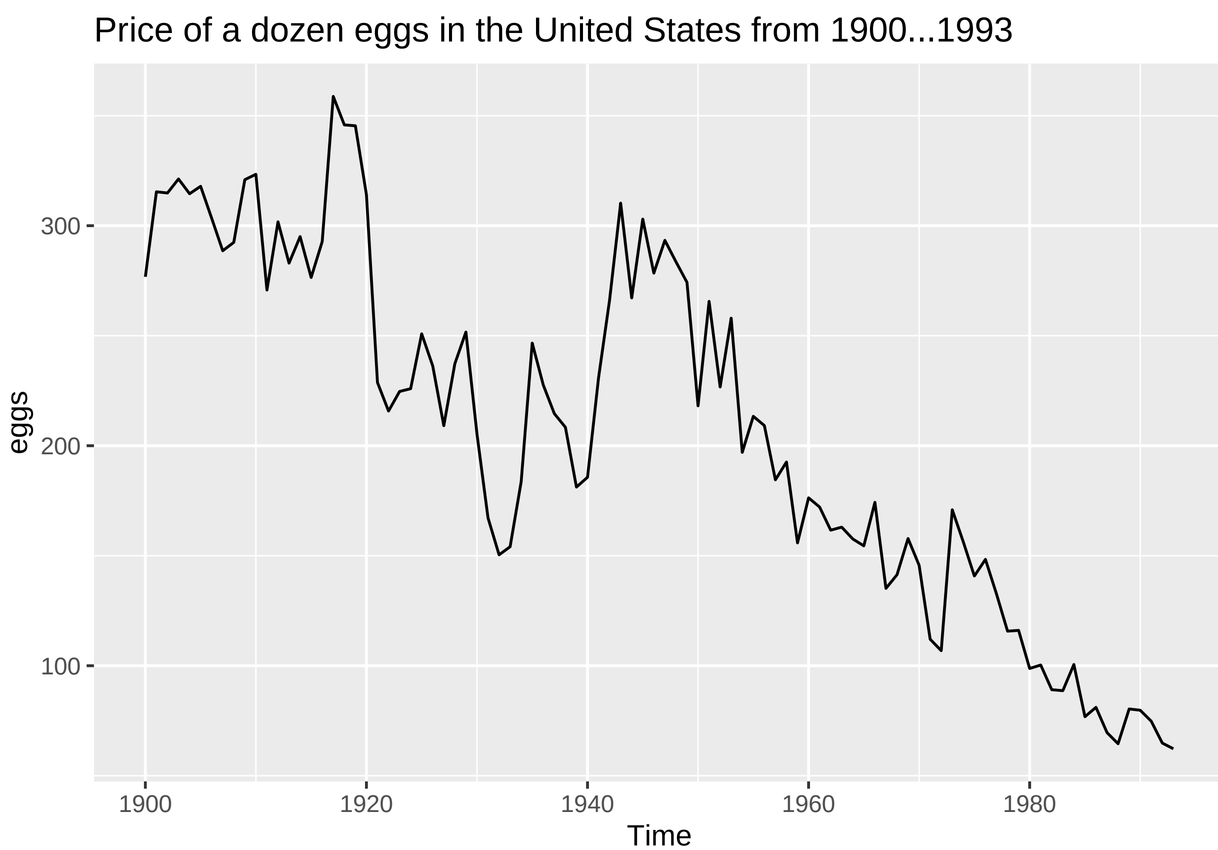
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
```

```
## 1993' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <93>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Price of a dozen eggs in the United States from 1900-
## 1993' in 'mbcsToSbcs': dot substituted for <93>
```



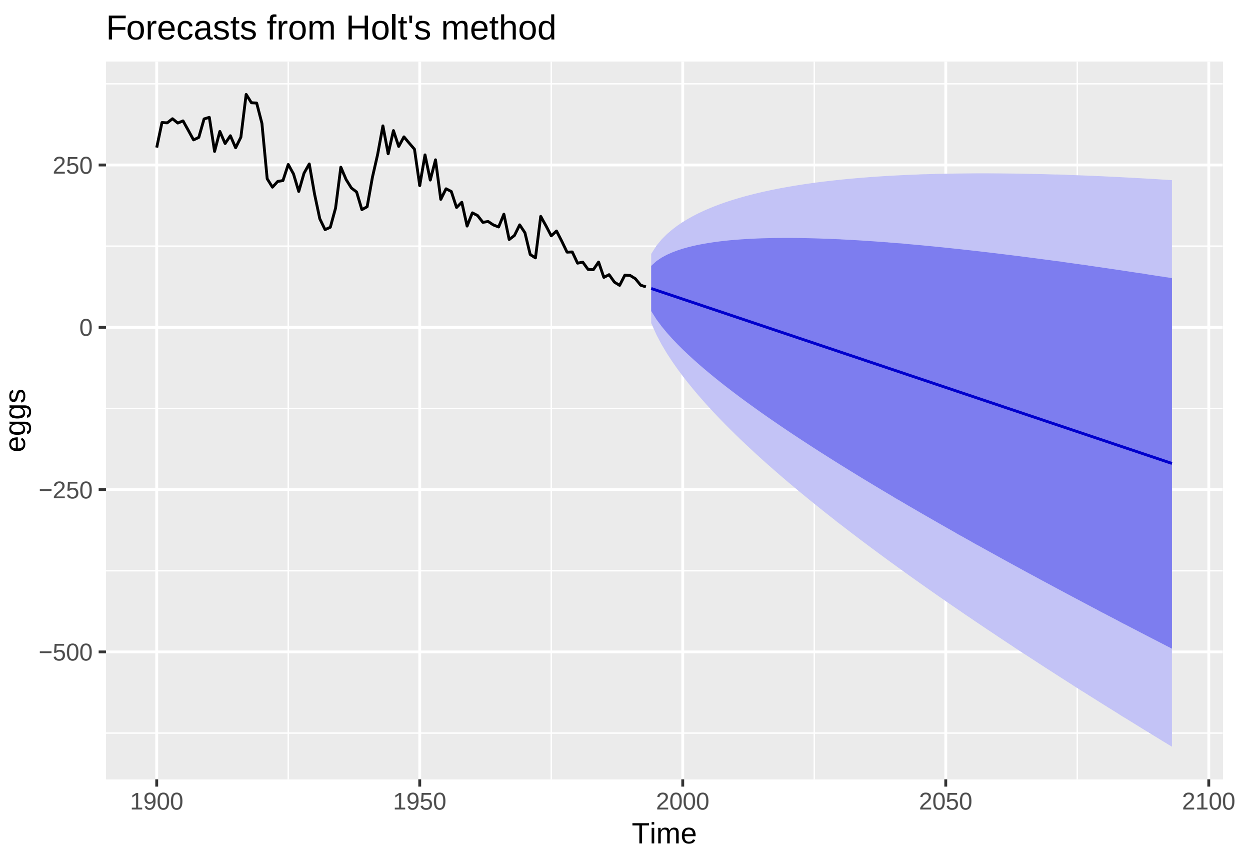
The time series shows the downward trend. It has the frequency as 1 that shows yearly record. We will perform forecast using below methods:

- Holt's
- Holts with damped trend
- Box-Cox transformation
- Box-Cox with damped trend

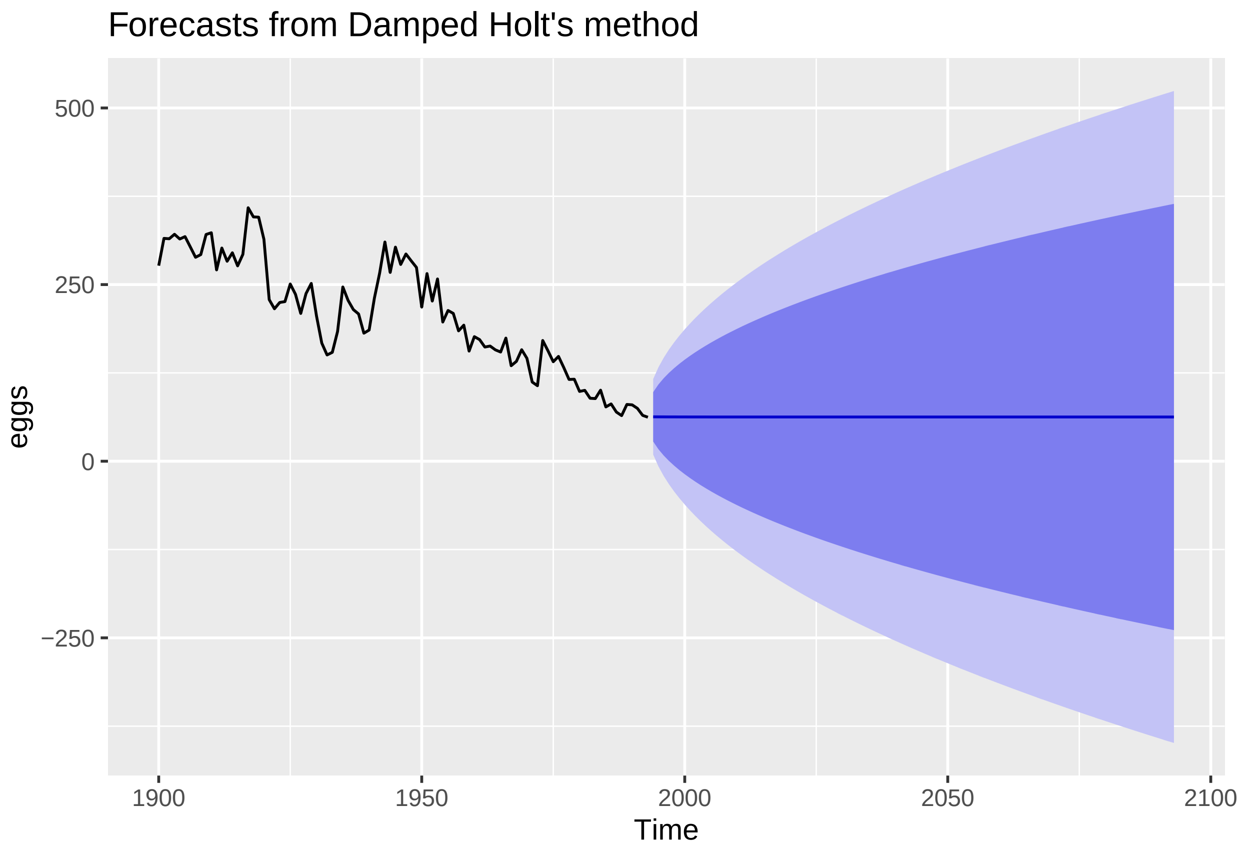
- Exponential
- Exponential with damped trend

```
h <- 100

# holts
eggs_holt <- holt(eggs, h=h)
# holts with damped trend
eggs_damped <- holt(eggs, h=h, damped = T)
# Box-Cox transformation
eggs_boxcox <- holt(eggs, h=h, lambda = "auto")
# Box-Cox with damped trend
eggs_boxcox_d <- holt(eggs, h=h, lambda = "auto", damped = T)
# exponential
eggs_exp <- holt(eggs, h=h, exponential = T)
# exponential with damped trend
eggs_exp_d <- holt(eggs, h=h, exponential = T, damped = T)
```

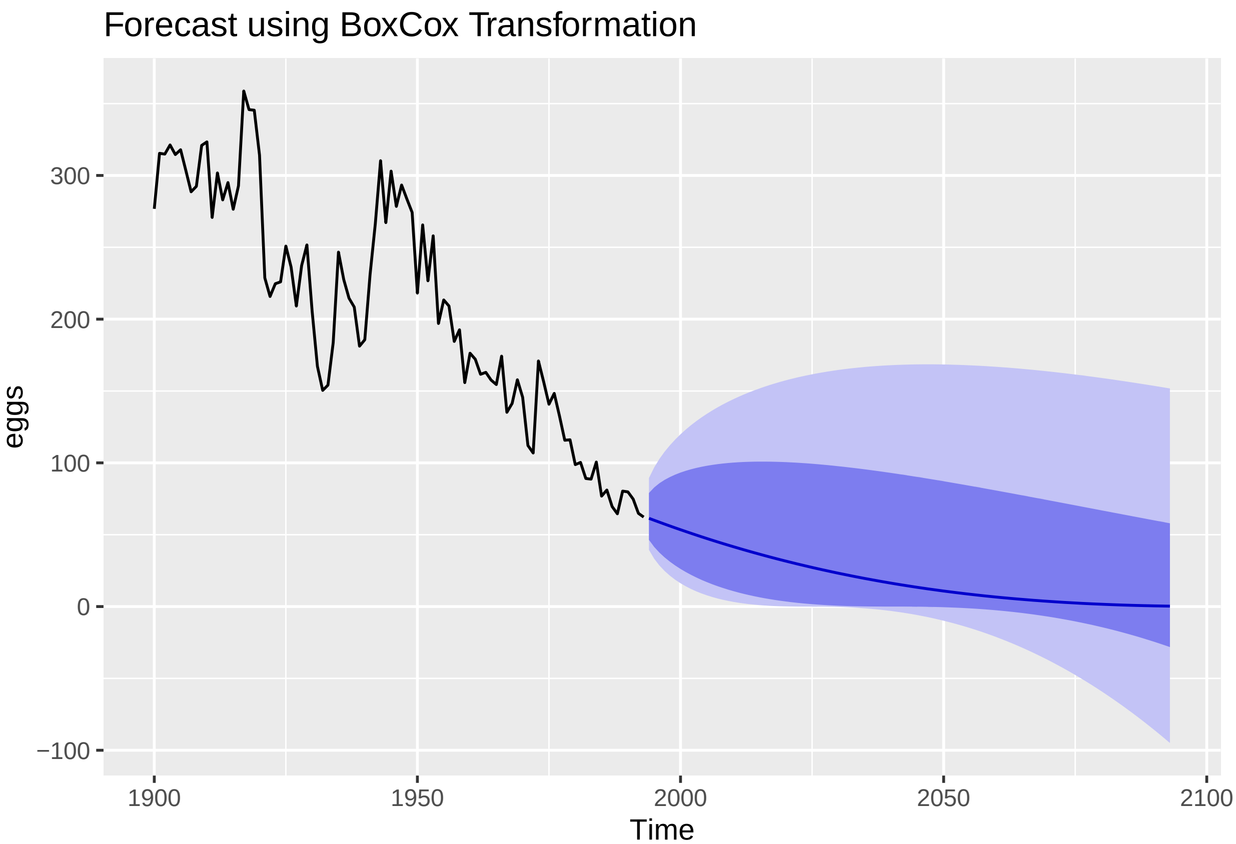
```
# Forcast from holt's
autoplot(eggs_holt)
```



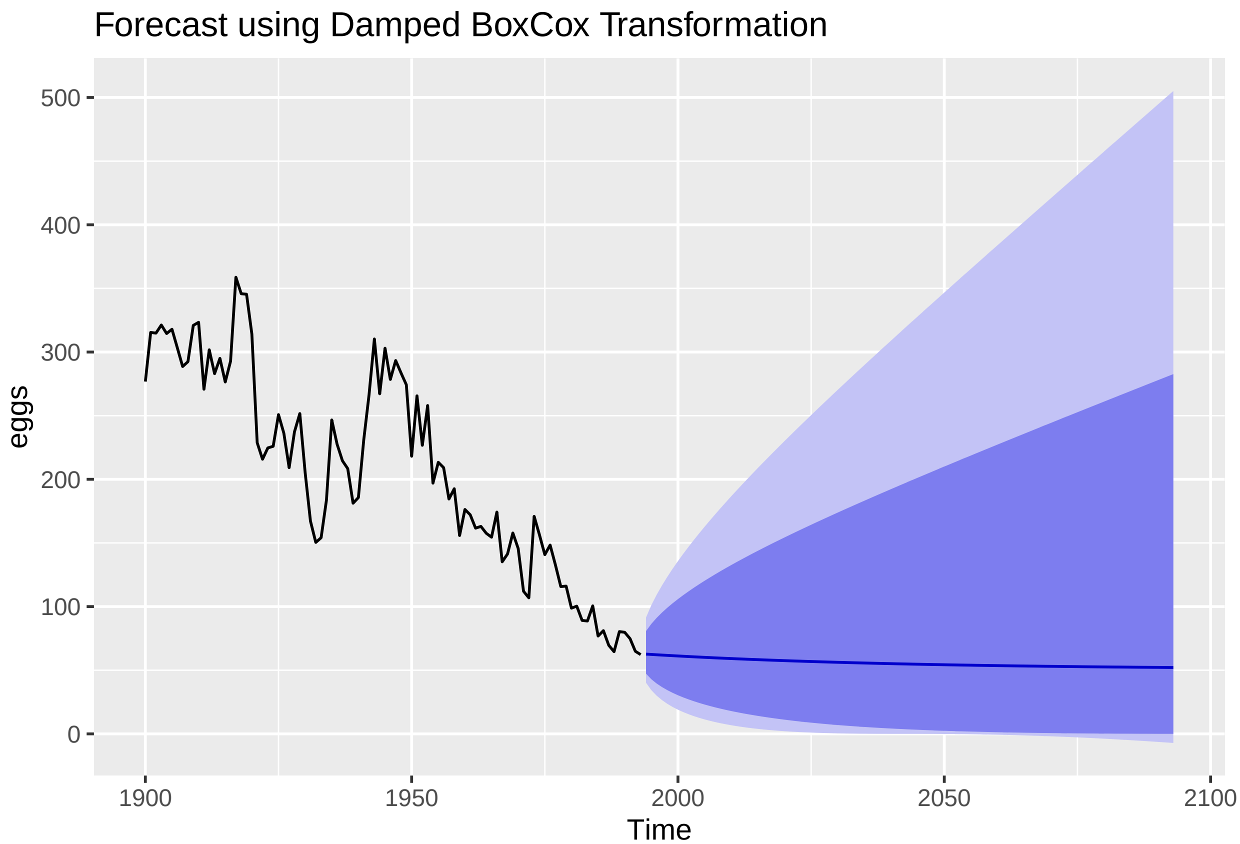

```
# Forecast from damped holt's
autoplot(eggs_damped)
```



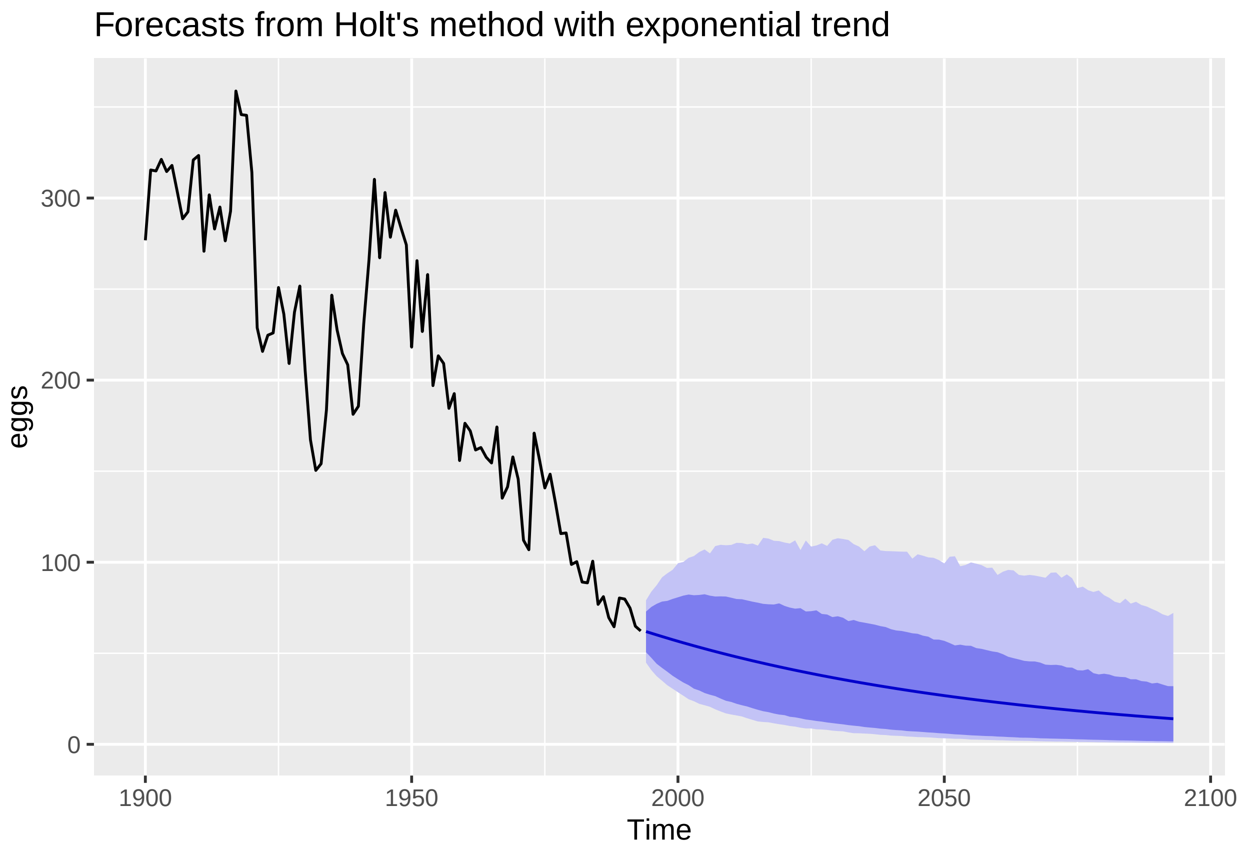

```r
# forecast from boxcox transformation
autoplot(eggs_boxcox) +
  labs(title = "Forecast using BoxCox Transformation")
```



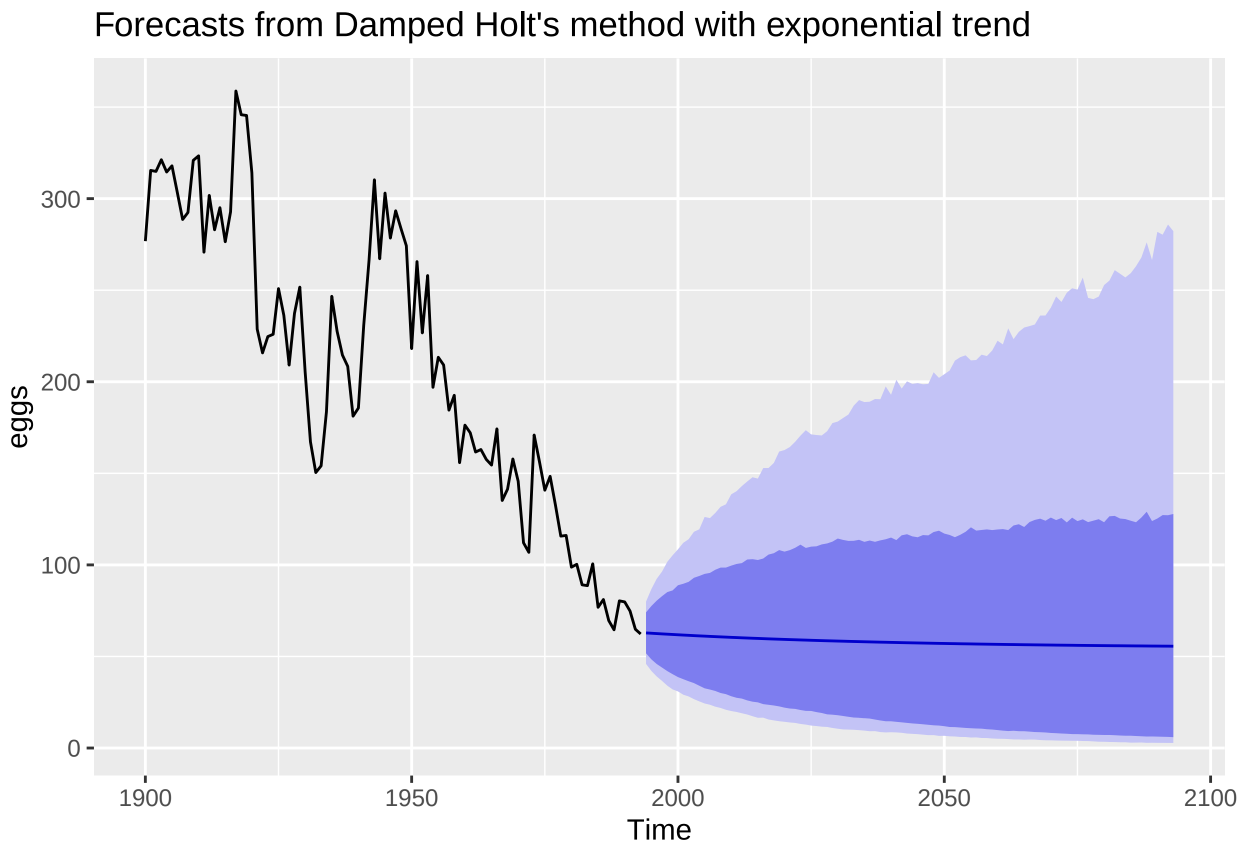

```
# forecast from damped boxcox transformation
autoplot(eggs_boxcox_d) +
  labs(title = "Forecast using Damped BoxCox Transformation")
```




```
# Forecast using exponential trend
autoplot(eggs_exp)
```




```
# Forecast using exponential trend and damped
autoplot(eggs_exp_d)
```




```
rmse_eggdf <- data.frame(
  Method = c("Holt's", "Damped Holt's","BoxCox", "Damped BoxCox", "Exponential", "Damped Exponential"),
  RMSE = c(accuracy(eggs_holt)[2],
           accuracy(eggs_damped)[2],
           accuracy(eggs_boxcox)[2],
           accuracy(eggs_boxcox_d)[2],
           accuracy(eggs_exp)[2],
           accuracy(eggs_exp_d)[2])
)

rmse_eggdf
```

```
##                Method     RMSE
## 1              Holt's 26.58219
## 2       Damped Holt's 26.54019
## 3              BoxCox 26.39376
## 4       Damped BoxCox 26.53321
## 5         Exponential 26.49795
## 6 Damped Exponential 26.59113
```

Analyzing all the RMSEs above for all the methods, it appears BoxCox transformation is the lowest (=26.39376). From all the 6 graphs above it is clear too that BoxCox transformation forecast captures the decline trend and good enough among all.

Exercise 7.8

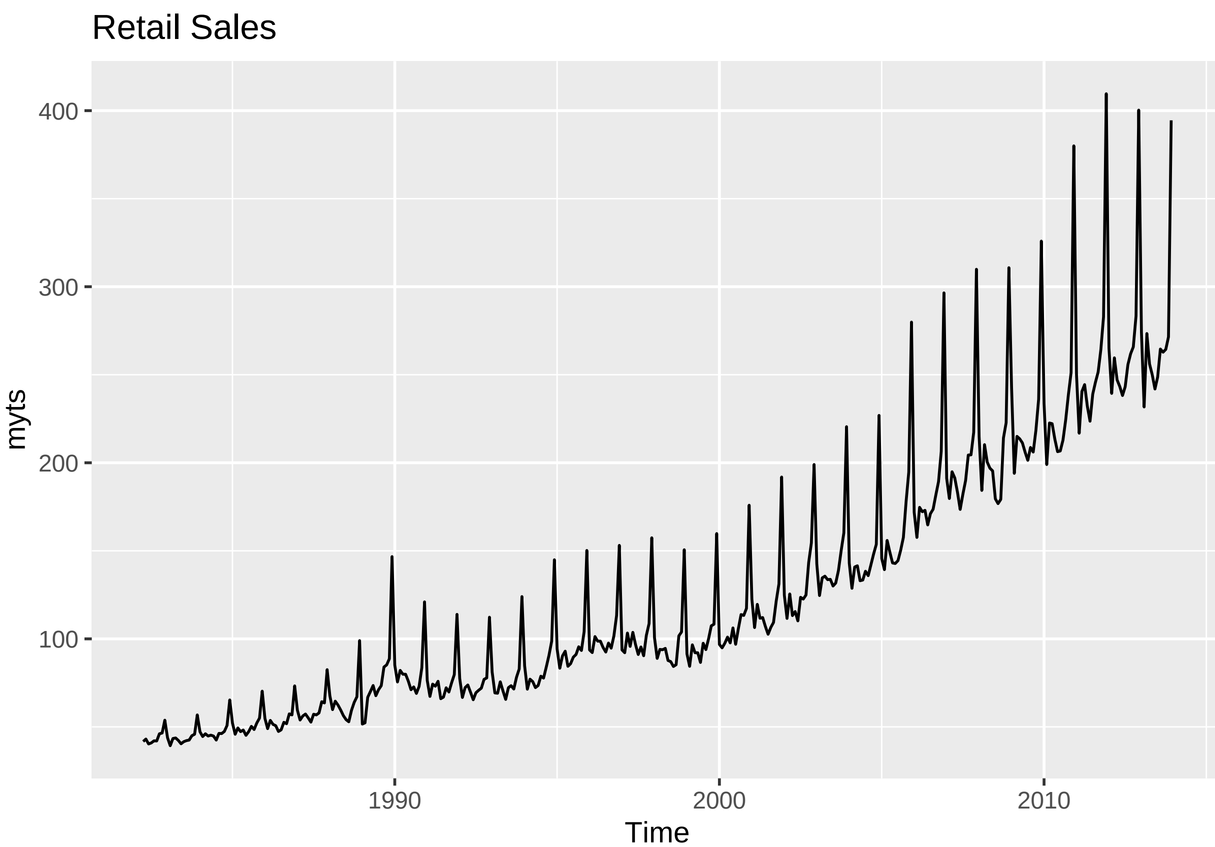
Recall your retail time series data (from Exercise 3 in Section 2.10).

```r
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
myts <- ts(retaildata[,"A3349627V"], frequency=12, start=c(1982,4))
```

a)

Why is multiplicative seasonality necessary for this series?

```r
autoplot(myts) +
  labs(title="Retail Sales")
```



The multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series. It appears in above graph that the variability in the series is increasing over years therefore multiplicative seasonality is necessary for the series.

b)

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

```r
# multiplicative
myts_hw <- hw(myts, seasonal = "multiplicative")
summary(myts_hw)
```

```
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
```

```
## Call:
##  hw(y = myts, seasonal = "multiplicative")
##
##   Smoothing parameters:
##     alpha = 0.5253
##     beta  = 0.0038
##     gamma = 0.0743
##
##   Initial states:
##     l = 44.3692
##     b = 0.1934
##     s = 0.9684 0.8895 0.9594 1.5454 1.1037 1.0445
##            0.9269 0.923 0.9101 0.8944 0.9305 0.9043
##
##   sigma:  0.0542
##
##      AIC     AICc      BIC
## 3620.609 3622.295 3687.637
##
## Error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.3076124 5.916807 4.201477 0.07243871 3.779909 0.4485563
##                     ACF1
## Training set -0.03122619
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2014       264.8837 246.4755 283.2918 236.7308 293.0365
## Feb 2014       233.7293 215.3569 252.1016 205.6312 261.8274
## Mar 2014       259.0735 236.5752 281.5718 224.6653 293.4817
## Apr 2014       251.3886 227.6492 275.1279 215.0823 287.6948
## May 2014       245.7384 220.7899 270.6869 207.5830 283.8939
## Jun 2014       238.1008 212.3348 263.8669 198.6950 277.5067
## Jul 2014       240.9197 213.3164 268.5229 198.7042 283.1352
## Aug 2014       246.5606 216.8117 276.3095 201.0636 292.0576
## Sep 2014       251.6342 219.8029 283.4655 202.9524 300.3160
## Oct 2014       268.4657 232.9920 303.9393 214.2134 322.7179
## Nov 2014       286.0274 246.6729 325.3820 225.8398 346.2151
## Dec 2014       409.4041 350.9061 467.9020 319.9392 498.8689
## Jan 2015       273.2441 232.4002 314.0881 210.7787 335.7096
## Feb 2015       241.0872 203.8497 278.3247 184.1374 298.0371
## Mar 2015       267.2081 224.6357 309.7805 202.0992 332.3170
## Apr 2015       259.2615 216.7199 301.8030 194.1997 324.3232
## May 2015       253.4145 210.6486 296.1804 188.0097 318.8193
## Jun 2015       245.5192 202.9598 288.0785 180.4303 310.6081
## Jul 2015       248.4065 204.2272 292.5859 180.8401 315.9730
## Aug 2015       254.2031 207.8659 300.5404 183.3364 325.0699
## Sep 2015       259.4141 210.9944 307.8338 185.3626 333.4656
## Oct 2015       276.7448 223.8995 329.5900 195.9250 357.5646
## Nov 2015       294.8257 237.2766 352.3748 206.8119 382.8394
## Dec 2015       421.9655 337.8314 506.0995 293.2935 550.6374
```

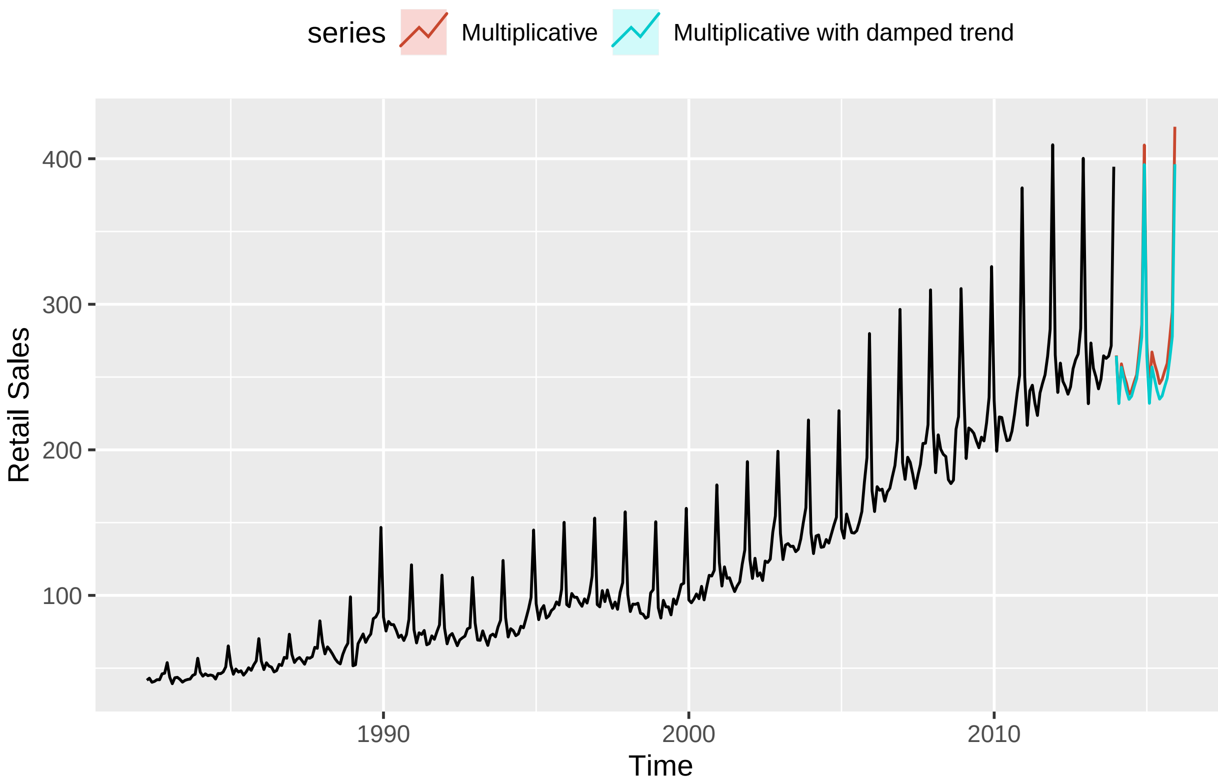
```r
# multiplicative damped
myts_hwd <- hw(myts, seasonal = "multiplicative", damped = T)
```

```
summary(myts_hwd)
```

```
##
## Forecast method: Damped Holt-Winters' multiplicative method
##
## Model Information:
## Damped Holt-Winters' multiplicative method
##
## Call:
##  hw(y = myts, seasonal = "multiplicative", damped = T)
##
##   Smoothing parameters:
##     alpha = 0.5562
##     beta  = 0.0185
##     gamma = 0.0079
##     phi   = 0.98
##
##   Initial states:
##     l = 43.9263
##     b = 0.0809
##     s = 0.9818 0.8807 1.0092 1.5184 1.061 0.9992
##            0.9525 0.9305 0.9055 0.8993 0.9149 0.947
##
##   sigma:  0.0526
##
##      AIC     AICc      BIC
## 3596.970 3598.860 3667.941
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.3511596 5.628031 4.018041 0.1776603 3.685373 0.4289724
##                     ACF1
## Training set -0.05517684
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2014       264.5113 246.6953 282.3274 237.2640 291.7586
## Feb 2014       231.7667 213.7561 249.7774 204.2218 259.3116
## Mar 2014       256.6838 234.2502 279.1173 222.3746 290.9929
## Apr 2014       248.5371 224.5225 272.5517 211.8100 285.2642
## May 2014       240.6016 215.2147 265.9886 201.7756 279.4276
## Jun 2014       234.6576 207.8710 261.4442 193.6910 275.6242
## Jul 2014       236.8817 207.8416 265.9217 192.4687 281.2946
## Aug 2014       243.1918 211.3633 275.0203 194.5143 291.8693
## Sep 2014       248.9366 214.3251 283.5481 196.0029 301.8703
## Oct 2014       262.3951 223.7986 300.9916 203.3669 321.4233
## Nov 2014       277.6314 234.5812 320.6817 211.7918 343.4711
## Dec 2014       395.9630 331.4370 460.4890 297.2790 494.6469
## Jan 2015       264.7371 219.4790 309.9952 195.5208 333.9535
## Feb 2015       231.9607 190.5023 273.4191 168.5556 295.3658
## Mar 2015       256.8944 208.9939 304.7948 183.6370 330.1518
## Apr 2015       248.7370 200.4449 297.0290 174.8807 322.5933
## May 2015       240.7913 192.1990 289.3836 166.4758 315.1068
## Jun 2015       234.8390 185.6580 284.0199 159.6232 310.0548
```

```
## Jul 2015       237.0611 185.6149 288.5074 158.3808 315.7415
## Aug 2015       243.3724 188.7151 298.0298 159.7812 326.9637
## Sep 2015       249.1179 191.2914 306.9444 160.6798 337.5559
## Oct 2015       262.5825 199.6560 325.5089 166.3448 358.8201
## Nov 2015       277.8258 209.1632 346.4884 172.8154 382.8361
## Dec 2015       396.2347 295.3451 497.1243 241.9374 550.5321
```

```
autoplot(myts) +
  autolayer(myts_hw, PI=F, series='Multiplicative') +
  autolayer(myts_hwd, PI=F, series='Multiplicative with damped trend') +
  theme(legend.position = "top") +
  ylab("Retail Sales")
```



Seeing the forecast, it appears multiplicative damped forecast the trend increases slowly as compared to only multiplicative one.

c)

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

```
rmse_retdf <- data.frame(
  Method = c("Mulitplicative", "Damped Mulitplicative"),
  RMSE = c(accuracy(myts_hw)[2],
           accuracy(myts_hwd)[2])
)

rmse_retdf
```

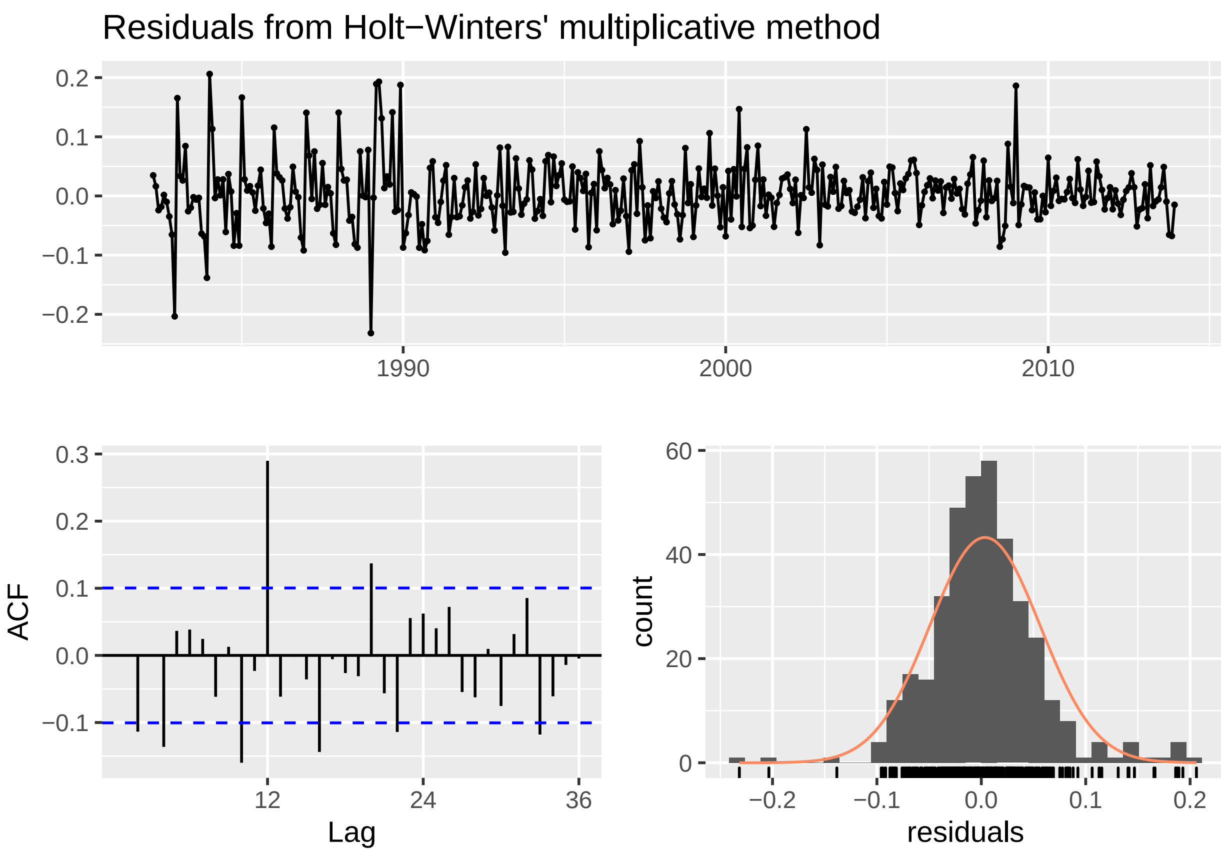
```
##                  Method     RMSE
## 1        Mulitplicative 5.916807
## 2 Damped Mulitplicative 5.628031
```

Comparing RMSEs for both these methods, it is apparent that multiplicative with damped method is better than multiplicative only.

d)

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

```
checkresiduals(myts_hw)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method
## Q* = 86.468, df = 8, p-value = 2.442e-15
##
## Model df: 16.   Total lags used: 24
```

For white noise series, we expect each autocorrelation to be close to zero. If one or more large spikes are outside these bounds, or if substantially more than 5% of spikes are outside these bounds, then the series is probably not white noise. Ljung-Box test result and ACF plot show that the residuals aren't white noise.

e)

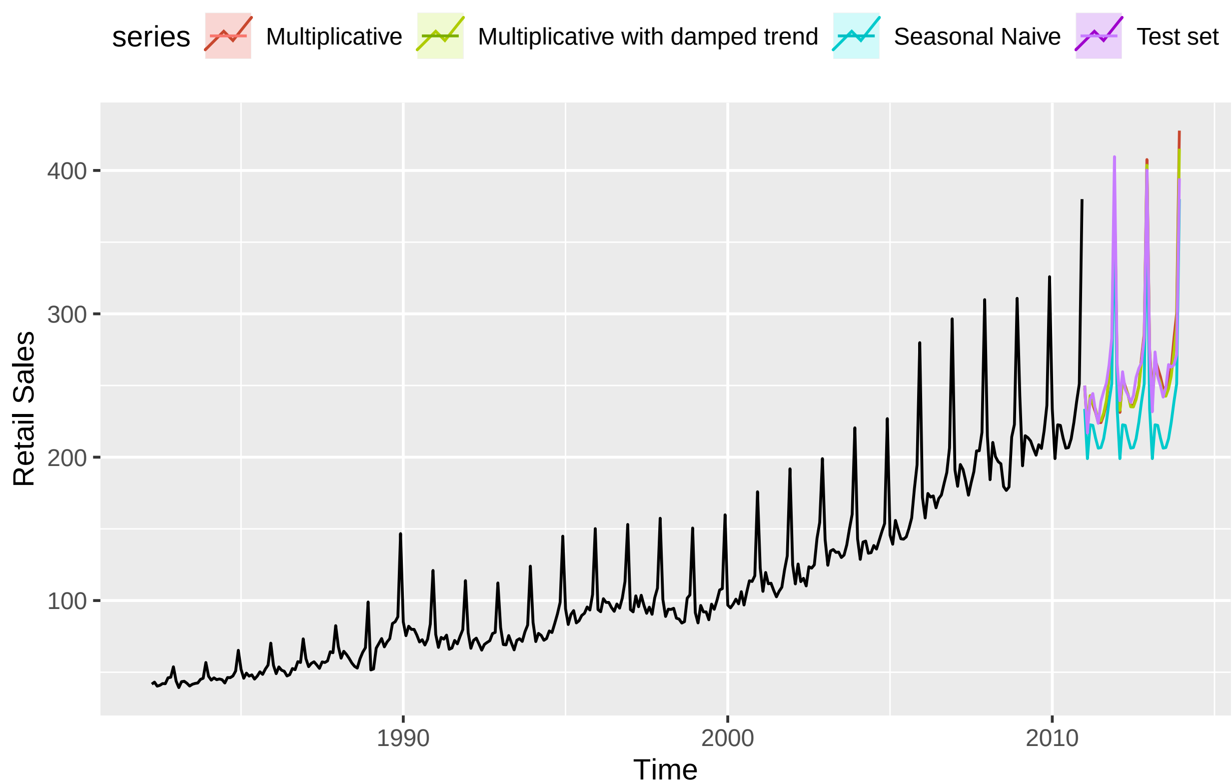
Now find the test set RMSE while training the model to the end of 2010. Can you beat the seasonal naive approach from Exercise 8 in Section 3.7?

```
myts_train <- window(myts, end=c(2010, 12))
myts_test <- window(myts, start = 2011)
```

```
myts_train_hw <- hw(myts_train, h=36, seasonal = "multiplicative")
myts_train_hwd <- hw(myts_train, h=36, seasonal = "multiplicative", damped = T)
myts_train_sn <- snaive(myts_train, h=36)

ap <- autoplot(myts_train) +
  autolayer(myts_train_hw, PI=F, series='Multiplicative') +
  autolayer(myts_train_hwd, PI=F, series='Multiplicative with damped trend') +
  autolayer(myts_train_sn, PI=F, series='Seasonal Naive') +
  autolayer(myts_test, PI=F, series='Test set') +
  theme(legend.position = "top") +
  ylab("Retail Sales")
```

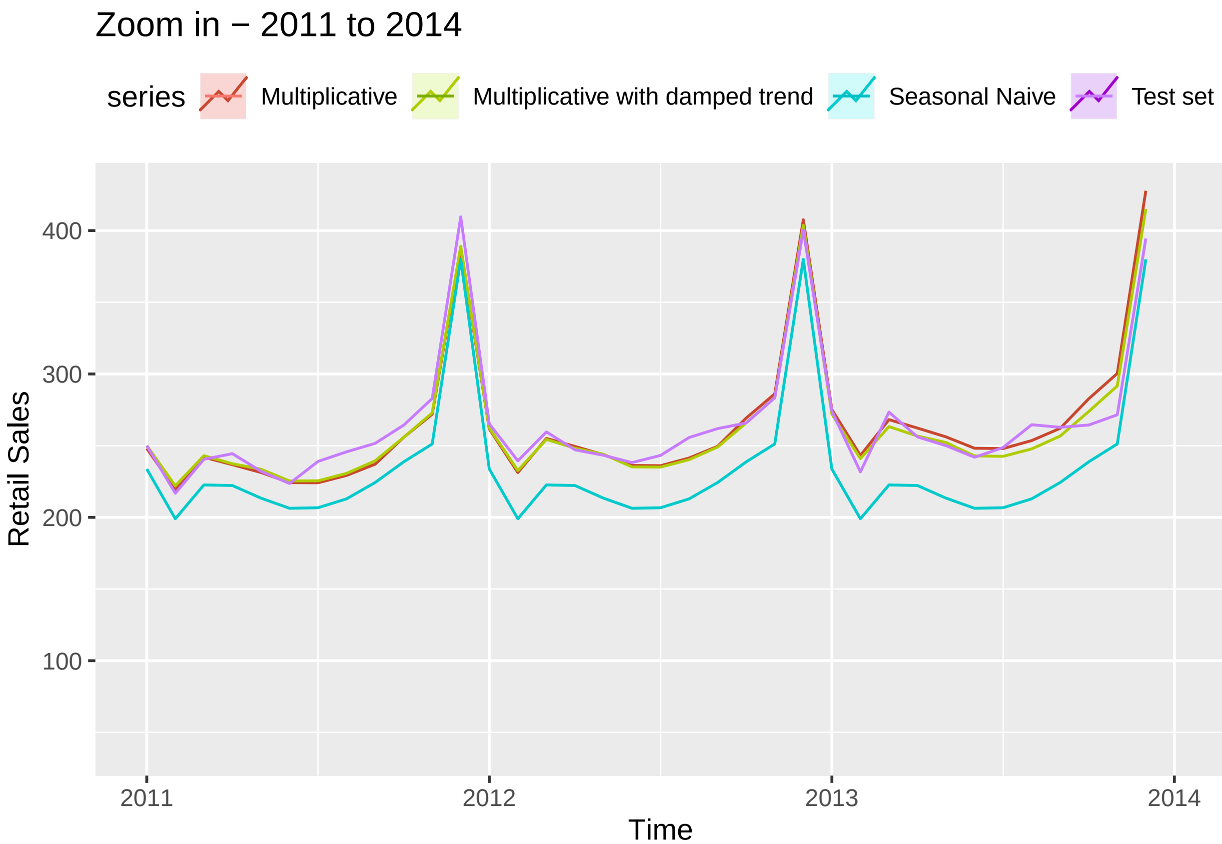
```
## Warning: Ignoring unknown parameters: PI
```

```
ap
```




```
ap + labs(title = "Zoom in - 2011 to 2014") + xlim(c(2011,2014))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```




```r
# RMSE- Holt-Winters' multiplicative method
accuracy(myts_train_hw, myts_test)[,2]
```

```
## Training set     Test set
##      5.36735     11.45707
```

```r
# RMSE- Holt-Winters' multiplicative method with damped trend
accuracy(myts_train_hwd, myts_test)[,2]
```

```
## Training set     Test set
##     5.390419     9.566560
```

```r
# RMSE- Seasonal Naive
accuracy(myts_train_sn, myts_test)[,2]
```

```
## Training set     Test set
##     12.27525     32.04685
```

Seeing the RMSEs for training and test sets, it seems Holt-Winters' multiplicative method with damping does fit the timeseries best among all. Therfore Holt-Winters' seems far more better than Seasonal Naive.

Exercise 7.9

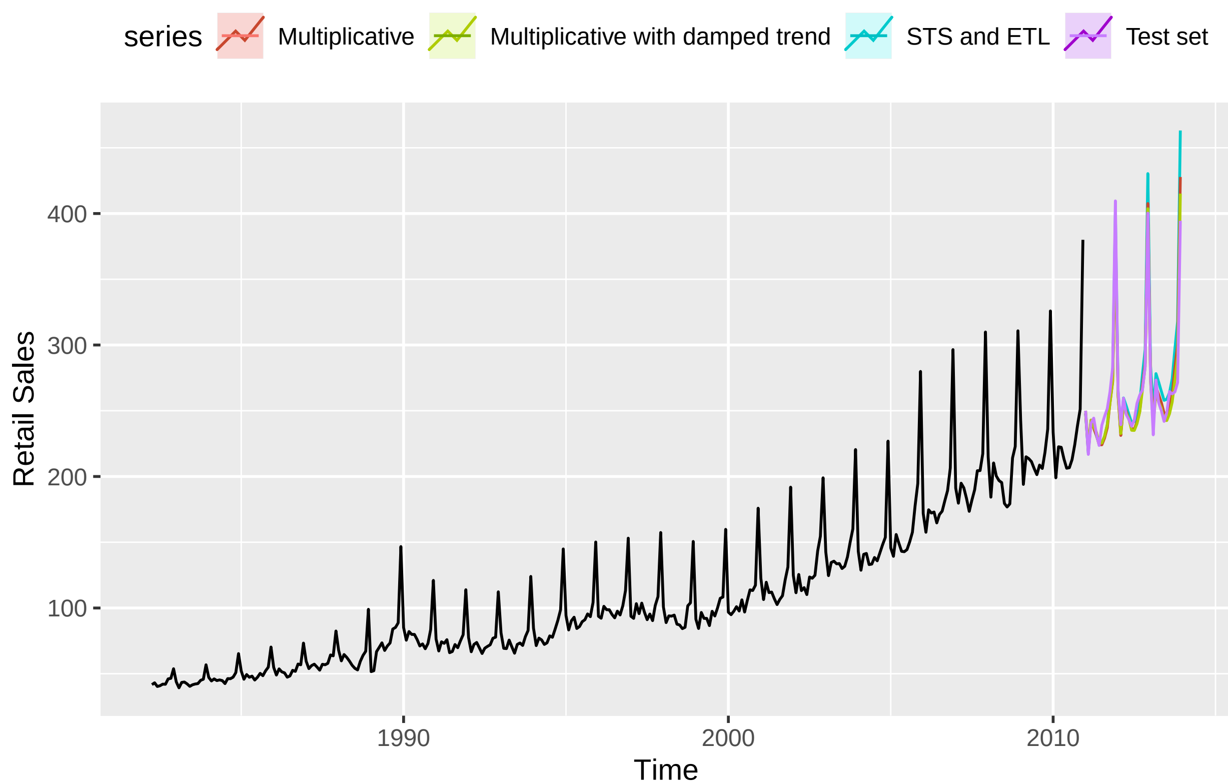
For the same retail data, try an STL decomposition applied to the Box-Cox transformed series, followed by ETS on the seasonally adjusted data. How does that compare with your best previous forecasts on the test set?

stlf() function is used to build the model that accepts time series, apply box cox transformation with ETS model.

```r
# STL and ETS
# ets to use for forecasting the seasonally adjusted series
# ZZN - N=none, A=additive, M=multiplicative and Z=automatically
# lambda=auto - Box-Cox transformation parameter
myts_t_stlets <- stlf(myts_train,
                      h=36,
                      method="ets",
                      etsmodel="ZZN",
                      lambda = "auto",
                      allow.multiplicative.trend = TRUE)

ap_stl <- autoplot(myts_train) +
  autolayer(myts_t_stlets, PI=F, series='STS and ETL') +
  autolayer(myts_train_hw, PI=F, series='Multiplicative') +
  autolayer(myts_train_hwd, PI=F, series='Multiplicative with damped trend') +
  autolayer(myts_test, PI=F, series='Test set') +
  theme(legend.position = "top") +
  ylab("Retail Sales")
```

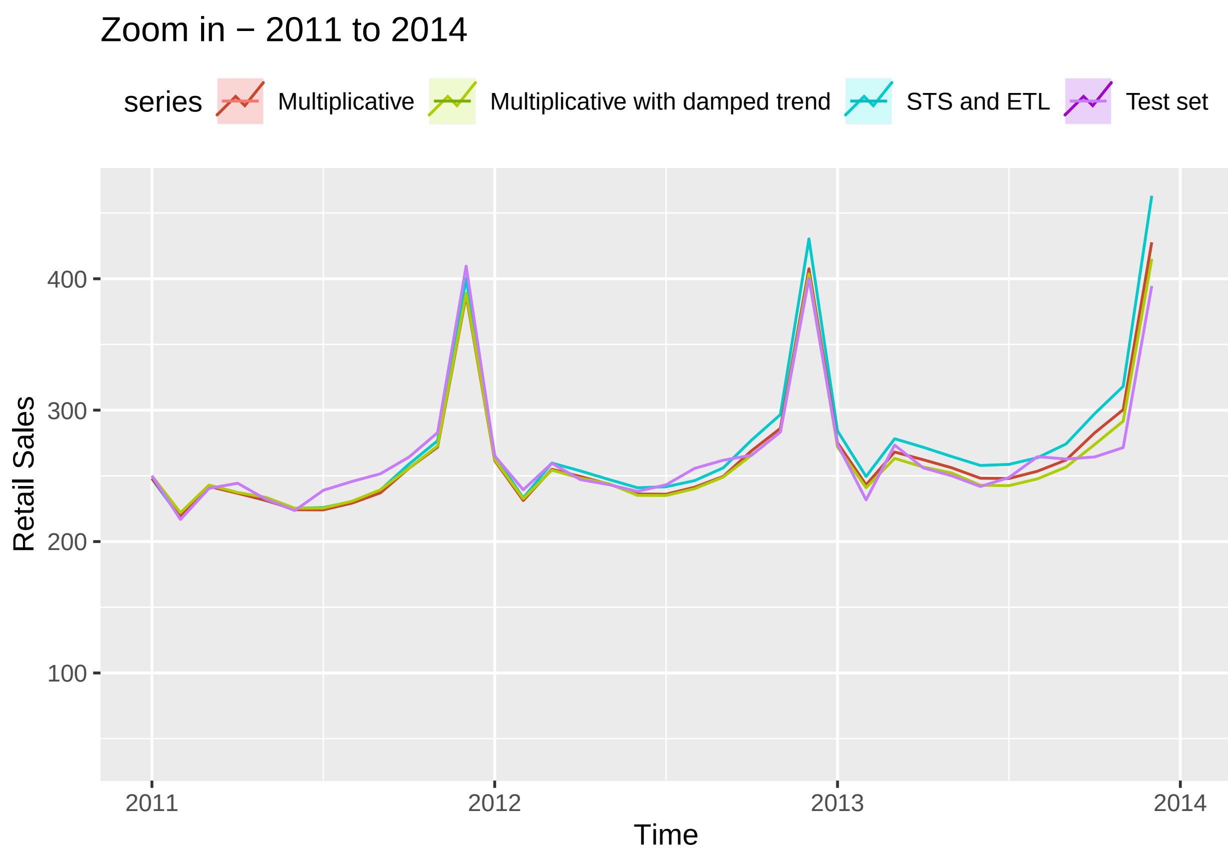
```
## Warning: Ignoring unknown parameters: PI
```

```r
ap_stl
```




```r
# zoom in
ap_stl + labs(title = "Zoom in - 2011 to 2014") + xlim(c(2011,2014))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```




```r
# RMSE- stl ets
accuracy(myts_t_stlets, myts_test)[,2]
```

```
## Training set     Test set
##     4.955934    17.936879
```

```r
# RMSE- Holt-Winters' multiplicative method with damped trend
accuracy(myts_train_hwd, myts_test)[,2]
```

```
## Training set     Test set
##     5.390419     9.566560
```

```r
# RMSE- Holt-Winters' multiplicative method
accuracy(myts_train_hw, myts_test)[,2]
```

```
## Training set     Test set
##      5.36735     11.45707
```

From the results, it is clear it seems Holt-Winters' multiplicative method with damping still fits the timeseries better than STL decomposition applied to the Box-Cox transformed series, followed by ETS on the seasonally adjusted data. RMSE difference shows the same results.