DATA624-HW5-ExpoSmoothing

FPP-Hyndman exercises 7.1, 7.5, 7.6, 7.7, 7.8, 7.9

Michael Y.

3/08/2020

Contents

```r
library(fable)
```

```
## Loading required package: fabletools
```

```r
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##
## Attaching package: 'forecast'
```

```
## The following objects are masked from 'package:fabletools':
##
##     GeomForecast, StatForecast
```

```r
library(fpp2)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

```r
library(kableExtra)
```

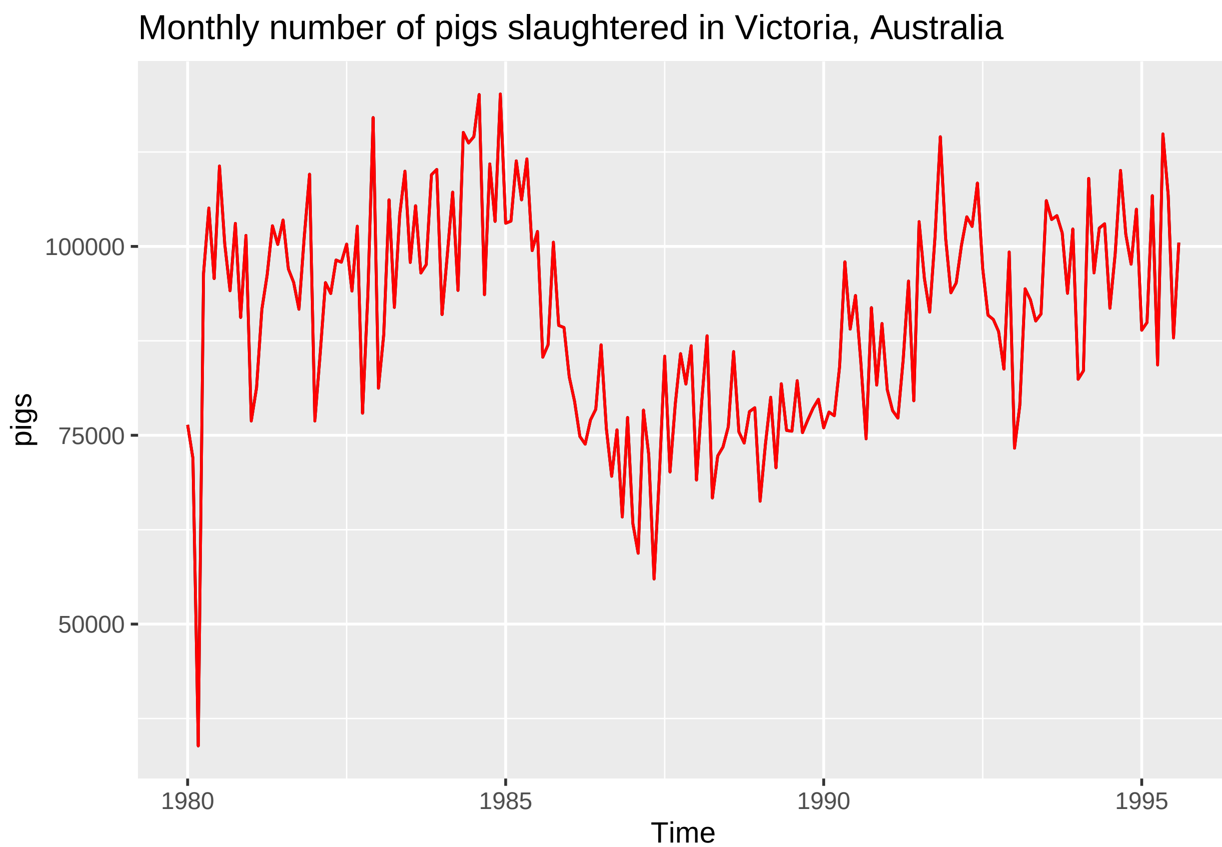
Homework 5 - Exponential Smoothing

Do exercises 7.1, 7.5, 7.6, 7.7, 7.8 and 7.9 in Hyndman. Please submit both your Rpubs link as well as attach the .rmd file with your code.

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

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.

```
# Monthly total number of pigs slaughtered in Victoria, Australia (Jan 1980 - Aug 1995)
pigs.title <- "Monthly number of pigs slaughtered in Victoria, Australia"
autoplot(pigs) + ggtitle(pigs.title) + geom_line(color="red")
```




```
pigs.ses_forecast <- ses(pigs, h=4)
summary(pigs.ses_forecast)
```

```
##
## 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
```

```
pigs.params <- pigs.ses_forecast$model$fit$par
pigs.alpha <- pigs.params[1]
pigs.l_0 <-pigs.params[2]
```

The optimal value of α is 0.2971488 and the optimal value of ℓ_0 is 77260.0561459 .

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.

```
# Compute the first forecast, and the standard deviation
pigs.ses_stdev <- sd(pigs.ses_forecast$residuals)
pigs_ses_forecast_1 <- pigs.ses_forecast$mean[1]

# Compute the prediction interval
pigs.my_pred95 <- c(
  my.Lower.95 = pigs_ses_forecast_1 - 1.96 * pigs.ses_stdev,
  my.Upper.95 = pigs_ses_forecast_1 + 1.96 * pigs.ses_stdev
)
# 95% prediction interval for the first forecast - calculated
pigs.my_pred95
```

```
## my.Lower.95 my.Upper.95
##    78679.97   118952.84
```

```


pigs.R_pred95 <- c(
  R.Lower = pigs.ses_forecast$lower[1,"95%"],
  R.Upper = pigs.ses_forecast$upper[1,"95%"])
# 95% prediction interval for the first forecast - as produced by R
pigs.R_pred95
```

Compare your interval with the interval produced by R.

```
## R.Lower.95% R.Upper.95%
##    78611.97   119020.84
```

The interval computed by R is slightly wider than the interval computed manually:

```
pigs.compare95 = rbind(pigs.my_pred95,
                       pigs.R_pred95)
colnames(pigs.compare95) <- c("Lower.95",
                              "Upper.95")
pigs.compare95
```

```
##                Lower.95 Upper.95
## pigs.my_pred95 78679.97 118952.8
## pigs.R_pred95  78611.97 119020.8
```

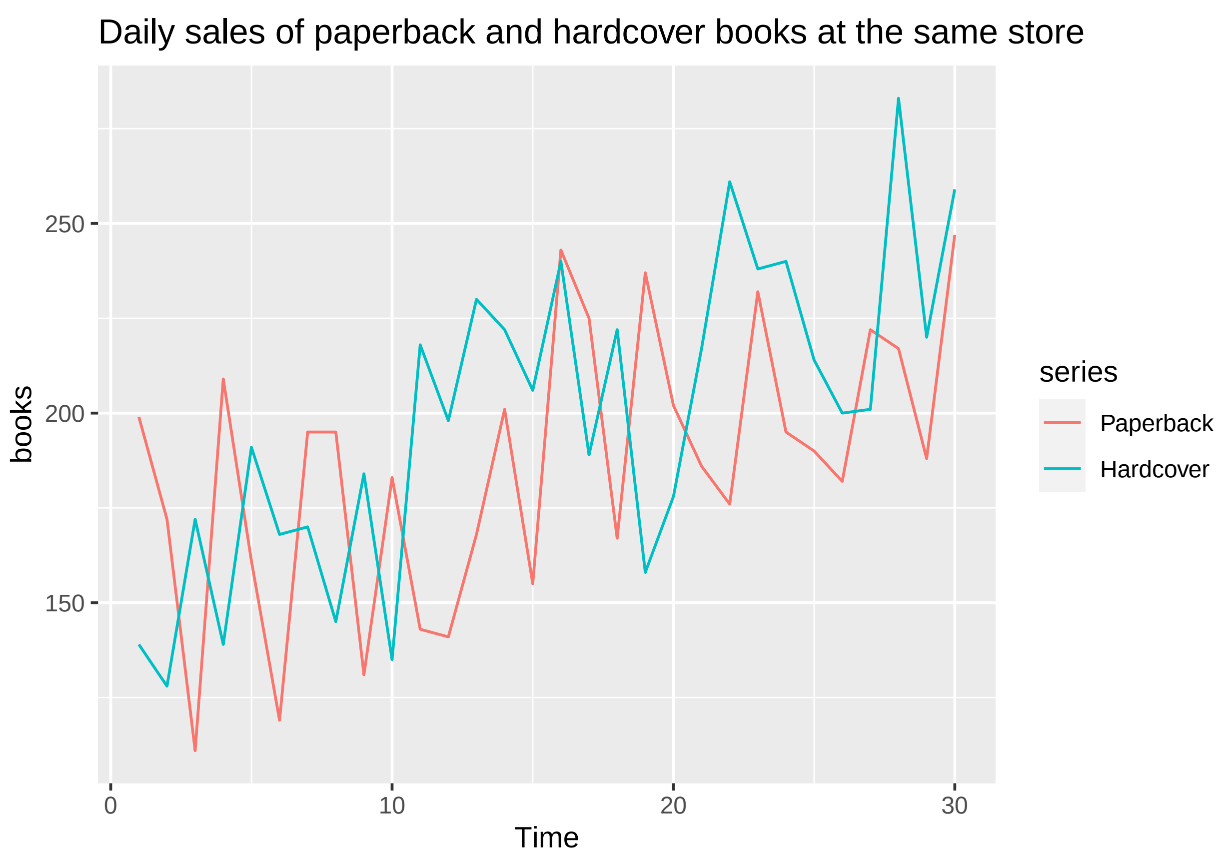
7.5 Data set `books` contains the daily sales of paperback and hardcover books at the same store. The task is to forecast the next four days' sales for paperback and hardcover books.

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

```r
# Daily sales of paperback and hardcover books at the same store.
summary(books)
```

```
##    Paperback       Hardcover
##  Min.   :111.0   Min.   :128.0
##  1st Qu.:167.2   1st Qu.:170.5
##  Median :189.0   Median :200.5
##  Mean   :186.4   Mean   :198.8
##  3rd Qu.:207.2   3rd Qu.:222.0
##  Max.   :247.0   Max.   :283.0
```

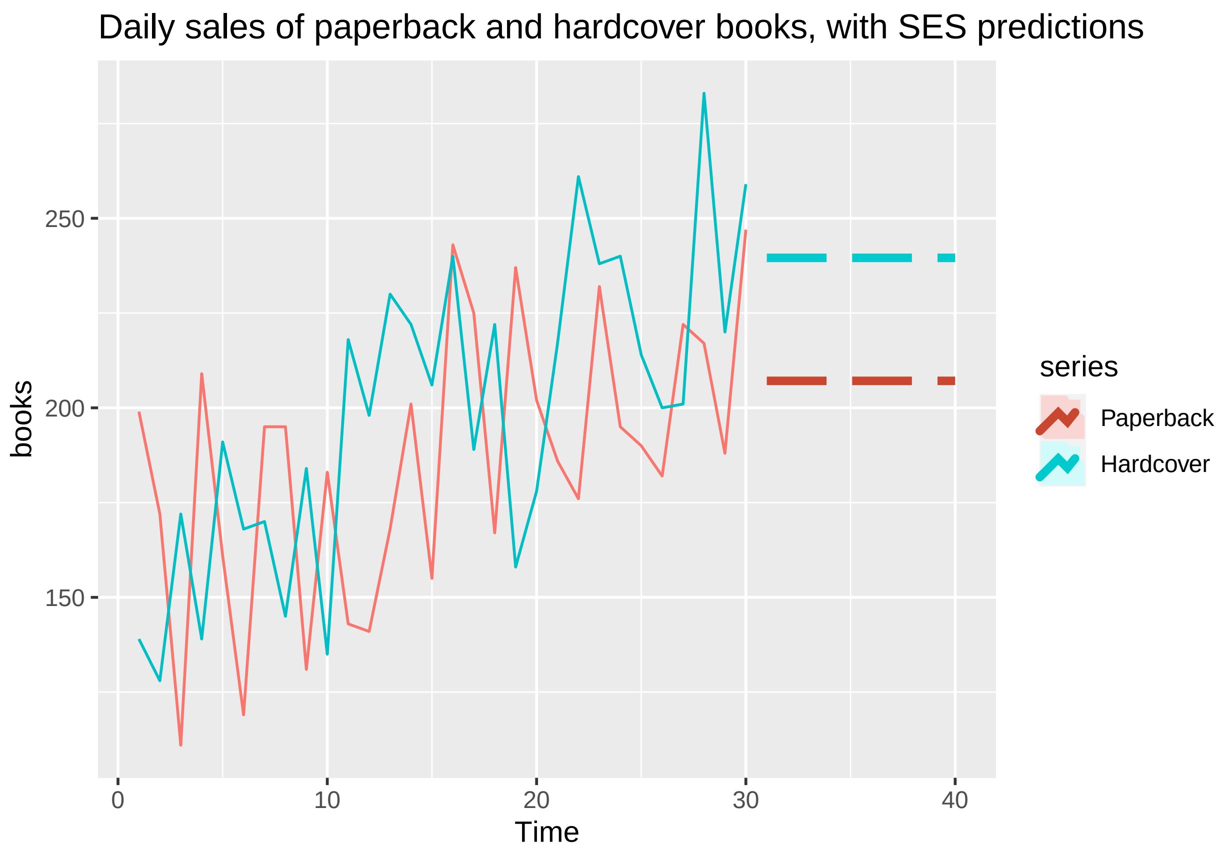
```r
autoplot(books)+ggtitle("Daily sales of paperback and hardcover books at the same store")
```



The dataset contains the daily sales of paperback and hardcover books over a period of 30 days.
Both series exhibit similarly upward trends over the month, but no seasonality (e.g., during the course of each "week" is evident.
The dataset is not labeled in a way which could enable identification of days of the week (e.g., weekdays vs. weekends.))

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

```
# SES forecasts of books data
hardcover_forecast_ses <- ses(books[,"Hardcover"])
paperback_forecast_ses <- ses(books[,"Paperback"])
autoplot(books) + ggtitle("Daily sales of paperback and hardcover books, with SES predictions") +
  autolayer(hardcover_forecast_ses, series="Hardcover", PI=FALSE,size=1.5,linetype=5) +
  autolayer(paperback_forecast_ses, series="Paperback", PI=FALSE,size=1.5,linetype=5)
```



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

```
# Hardcover RMSE
hardcover_RMSE_ses <- sqrt(hardcover_forecast_ses$model$mse)
# Paperback RMSE
paperback_RMSE_ses <- sqrt(paperback_forecast_ses$model$mse)
books_RMSE_ses <- c(Hardcover=hardcover_RMSE_ses,
                PaperBack=paperback_RMSE_ses)
# RMSE ses
books_RMSE_ses
```

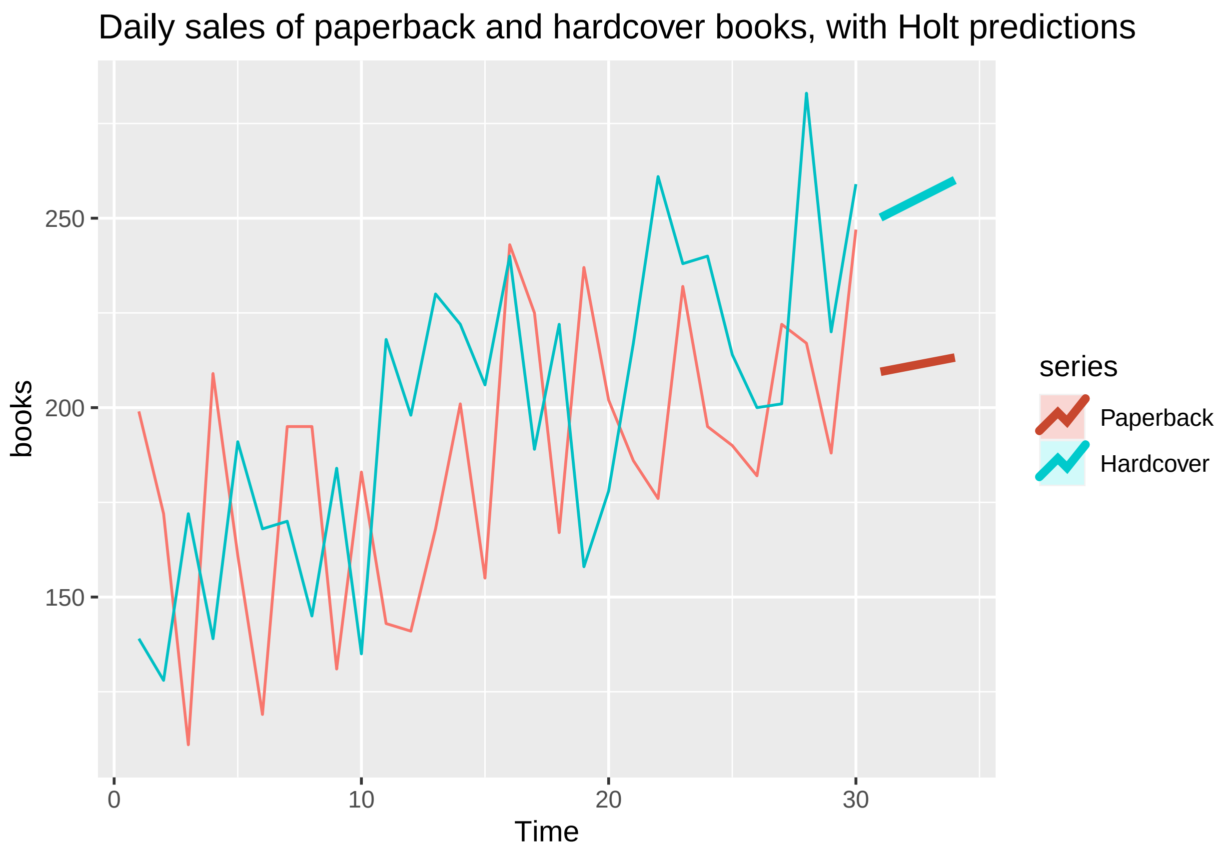
```
## Hardcover PaperBack
##  31.93101  33.63769
```

Under the SES model, the hardcover RMSE is 31.931015 and the paperback RMSE is 33.6376868.

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.

```
# Holt predictions for books
hardcover_forecast_holt <- holt(books[,"Hardcover"], h=4)
paperback_forecast_holt <- holt(books[,"Paperback"], h=4)
autoplot(books) + ggtitle("Daily sales of paperback and hardcover books, with Holt predictions") +
  autolayer(hardcover_forecast_holt, series="Hardcover", PI=FALSE,size=1.5,linetype=1) +
  autolayer(paperback_forecast_holt, series="Paperback", PI=FALSE,size=1.5,linetype=1)
```



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.)

```
# Hardcover
hardcover_RMSE_holt <- sqrt(hardcover_forecast_holt$model$mse)
# Paperback
paperback_RMSE_holt <- sqrt(paperback_forecast_holt$model$mse)
# Holt RMSE
```

```
books_RMSE_holt <- c(Hardcover=hardcover_RMSE_holt,
                     PaperBack=paperback_RMSE_holt)
books_RMSE_holt
```

```
## Hardcover PaperBack
##  27.19358  31.13692
```

Under the HOLT model, the hardcover RMSE is 27.193578 and the paperback RMSE is 31.136923.

```
books_RMSE <- rbind(books_RMSE_ses,
                    books_RMSE_holt)
books_RMSE
```

```
##                 Hardcover PaperBack
## books_RMSE_ses   31.93101  33.63769
## books_RMSE_holt  27.19358  31.13692
```

The RMSE for Holt is lower than that for SES.

Discuss the merits of the two forecasting methods for these data sets. The SES method provides a simple, straightline forecast, while the Holt forecasting method incorporates the trend, which is increasing.

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

```
# Forecasts
rbind(SES=hardcover_forecast_ses$mean[1:4],
      Holt=hardcover_forecast_holt$mean[1:4]) %>% t
```

```
##           SES     Holt
## [1,] 239.5601 250.1739
## [2,] 239.5601 253.4765
## [3,] 239.5601 256.7791
## [4,] 239.5601 260.0817
```

The Holt forecast appears better, as it incorporates the increasing trend, and is thus always greater than the SES forecast.

d) Calculate a 95% prediction interval for the first forecast for each series, using the RMSE values and assuming normal errors.

Compute hardcover Holt prediction intervals

```
# Hardcover forecast under HOLT
hardcover_Upper95_holt <- hardcover_forecast_holt$mean[1] + 1.96*hardcover_RMSE_holt
hardcover_Lower95_holt <- hardcover_forecast_holt$mean[1] - 1.96*hardcover_RMSE_holt
# Hardcover HOLT 95% PI
my_hardcover_holt_95 <- c(`Lower.95%` = hardcover_Lower95_holt,
                          `Upper.95%` = hardcover_Upper95_holt)
```

Table 1: Hardcover Prediction Intervals (Holt

	Lower.95%	Upper.95%
my_hardcover_holt_95	196.8745	303.4733
R_hardcover_holt_95	192.9222	307.4256

```
#my_hardcover_holt_95

# Holt 95% PI returned by R
R_hardcover_holt_95 <- c(Lower=hardcover_forecast_holt$lower[1,"95%"],
                         Upper=hardcover_forecast_holt$upper[1,"95%"])
#R_hardcover_holt_95

rbind(my_hardcover_holt_95,R_hardcover_holt_95)%>% kable() %>% kable_styling(c("bordered","striped"),ful
```

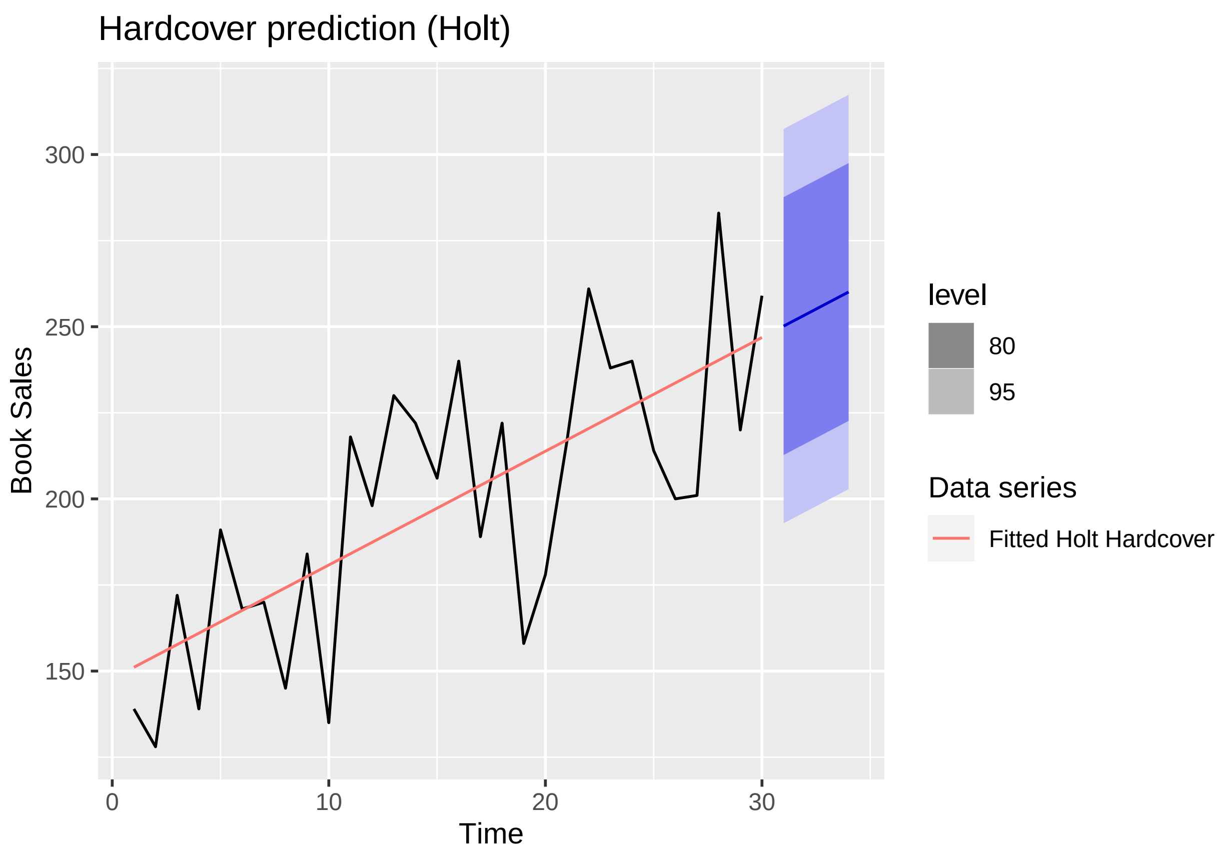
	Lower.95%	Upper.95%
my_hardcover_holt_95	196.8745	303.4733
R_hardcover_holt_95	192.9222	307.4256

Compare hardcover Holt prediction intervals

```
hardcover_holt_compare95 = rbind(my_hardcover_holt_95,
                                  R_hardcover_holt_95)
hardcover_holt_compare95 %>%
  kable(caption = "Hardcover Prediction Intervals (Holt") %>%
  kable_styling(c("bordered","striped"),full_width = F)
```

The interval calculated by R is wider.

```
autoplot(hardcover_forecast_holt) +
autolayer(fitted(hardcover_forecast_holt), series = "Fitted Holt Hardcover") +
ggtitle("Hardcover prediction (Holt)") +
  xlab("Time") +
  ylab("Book Sales") +
  guides(colour=guide_legend(title="Data series"),
         fill=guide_legend(title="Prediction interval"))
```



Compute paperback Holt prediction intervals

```
# paperback forecast under HOLT
paperback_Upper95Holt <- paperback_forecast_holt$mean[1] + 1.96*paperback_RMSE_holt
paperback_Lower95Holt <- paperback_forecast_holt$mean[1] - 1.96*paperback_RMSE_holt
# paperback HOLT 95% PI
my_paperback_holt_95 <- c(`Lower.95%` = paperback_Lower95Holt, `Upper.95%` = paperback_Upper95Holt)
#my_paperback_holt_95

# paperback HOLT computed by R
R_paperback_holt_95 <- c(Lower=paperback_forecast_holt$lower[1,"95%"],
                         Upper=paperback_forecast_holt$upper[1,"95%"])
#R_paperback_holt_95
```

Compare paperback Holt prediction intervals

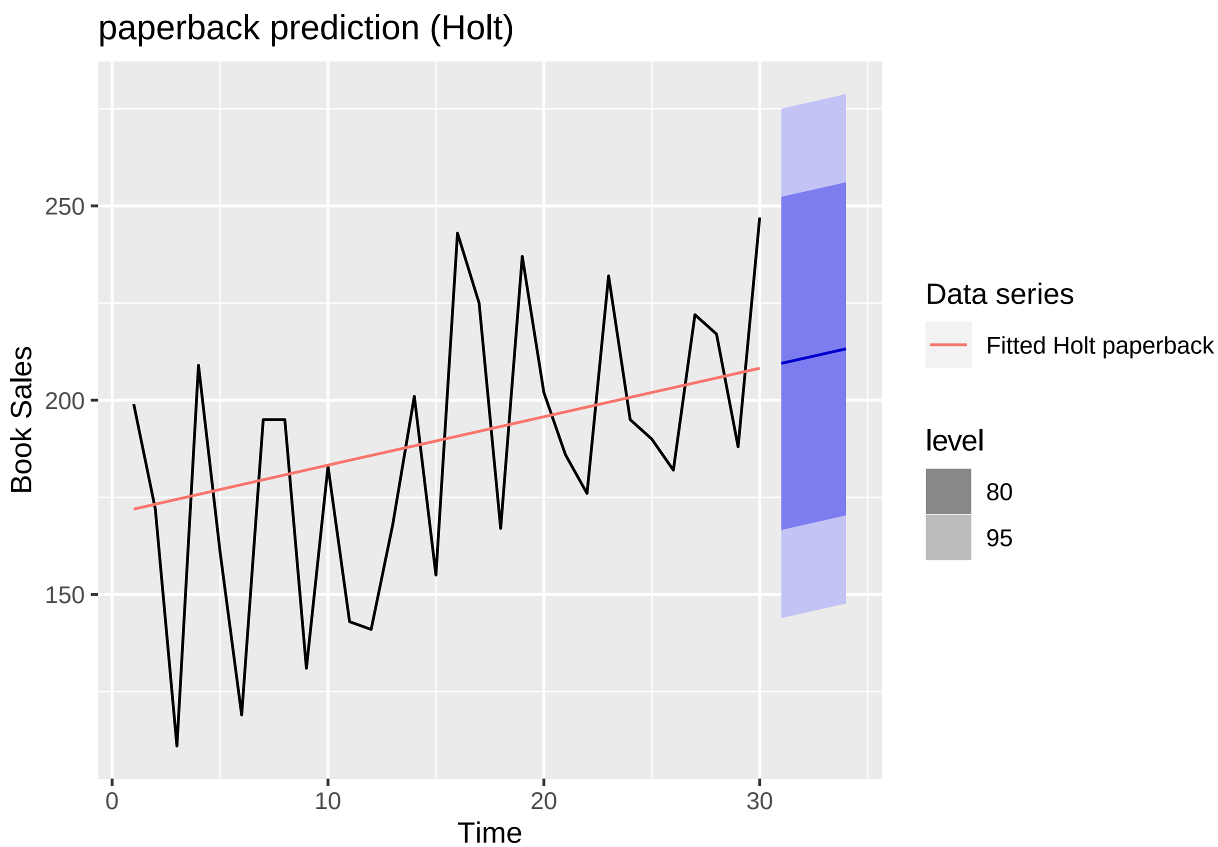
```
paperback.compare95 = rbind(my_paperback_holt_95,
                            R_paperback_holt_95)
paperback.compare95 %>%
  kable(caption = "Paperback Prediction Intervals (Holt)") %>%
  kable_styling(c("bordered","striped"),full_width = F)
```

The interval calculated by R is wider.

Table 2: Paperback Prediction Intervals (Holt)

	Lower.95%	Upper.95%
my_paperback_holt_95	148.4384	270.4951
R_paperback_holt_95	143.9130	275.0205

```
autoplot(paperback_forecast_holt) +
autolayer(fitted(paperback_forecast_holt), series = "Fitted Holt paperback") +
ggtitle("paperback prediction (Holt)") +
  xlab("Time") +
  ylab("Book Sales") +
  guides(colour=guide_legend(title="Data series"),
         fill=guide_legend(title="Prediction interval"))
```



Compare your intervals with those produced using `ses` and `holt`. Compute hardcover SES prediction intervals

```
# Hardcover forecast under ses
hardcover_Upper95_ses <- hardcover_forecast_ses$mean[1] + 1.96*hardcover_RMSE_ses
hardcover_Lower95_ses <- hardcover_forecast_ses$mean[1] - 1.96*hardcover_RMSE_ses
# Hardcover ses 95% PI
my_hardcover_ses_95 <- c(`Lower.95%` = hardcover_Lower95_ses,
                          `Upper.95%` = hardcover_Upper95_ses)
```

Table 3: Hardcover Prediction Intervals (SES)

	Lower.95%	Upper.95%
my_hardcover_ses_95	176.9753	302.1449
R_hardcover_ses_95	174.7799	304.3403

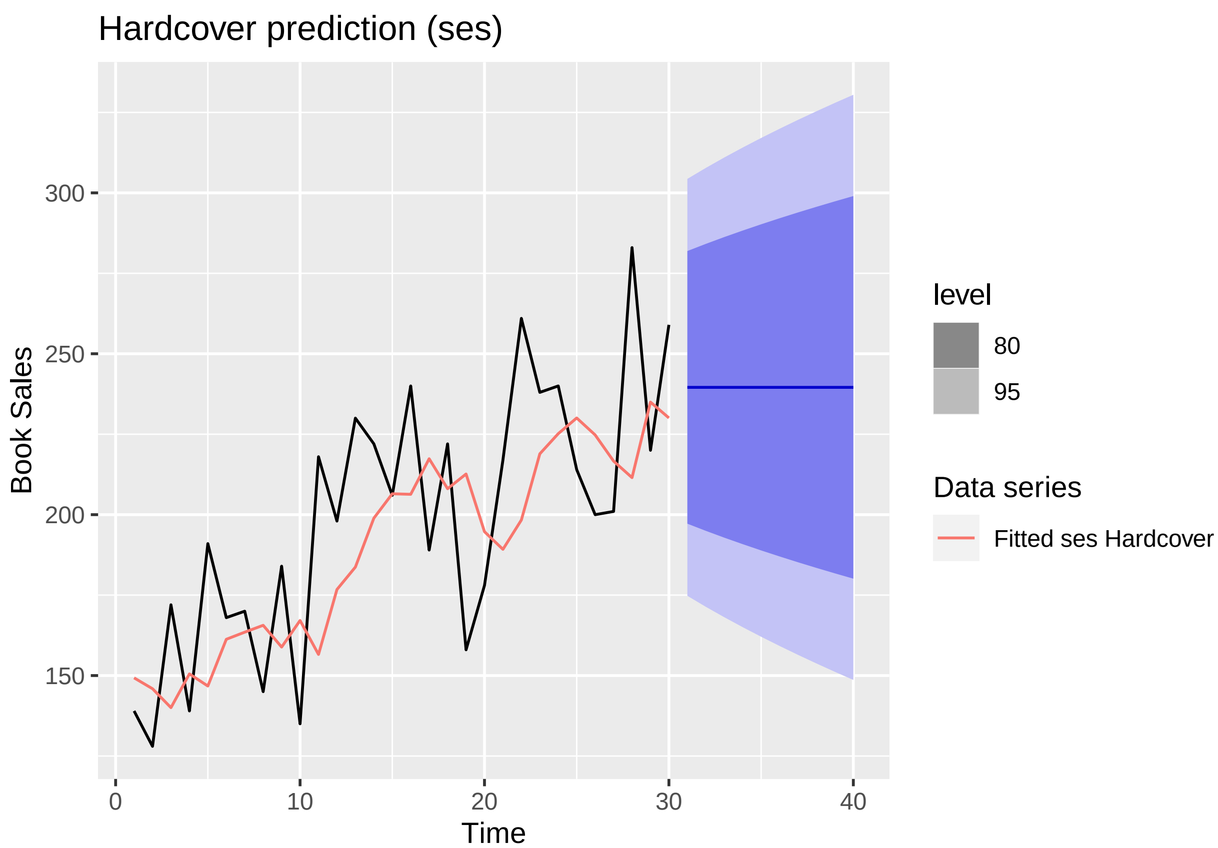
```
#my_hardcover_ses_95

# ses 95% PI returned by R
R_hardcover_ses_95 <- c(Lower=hardcover_forecast_ses$lower[1,"95%"],
                         Upper=hardcover_forecast_ses$upper[1,"95%"])
#R_hardcover_ses_95
```

Compare hardcover SES prediction intervals

```
hardcover_ses_compare95 = rbind(my_hardcover_ses_95,
                            R_hardcover_ses_95)
hardcover_ses_compare95  %>%
  kable(caption = "Hardcover Prediction Intervals (SES)") %>%
  kable_styling(c("bordered","striped"),full_width = F)
```

```
autoplot(hardcover_forecast_ses) +
autolayer(fitted(hardcover_forecast_ses), series = "Fitted ses Hardcover") +
ggtitle("Hardcover prediction (ses)") +
  xlab("Time") +
  ylab("Book Sales") +
  guides(colour=guide_legend(title="Data series"),
         fill=guide_legend(title="Prediction interval"))
```



Compute paperback SES prediction intervals

```r
# paperback forecast under ses
paperback_Upper95_ses <- paperback_forecast_ses$mean[1] + 1.96*paperback_RMSE_ses
paperback_Lower95_ses <- paperback_forecast_ses$mean[1] - 1.96*paperback_RMSE_ses
# paperback ses 95% PI
my_paperback_ses_95 <- c(`Lower.95%` = paperback_Lower95_ses,
                         `Upper.95%` = paperback_Upper95_ses)
#my_paperback_ses_95

# ses 95% PI returned by R
R_paperback_ses_95 <- c(Lower=paperback_forecast_ses$lower[1,"95%"],
                        Upper=paperback_forecast_ses$upper[1,"95%"])
#R_paperback_ses_95
```

Compare paperback SES prediction intervals

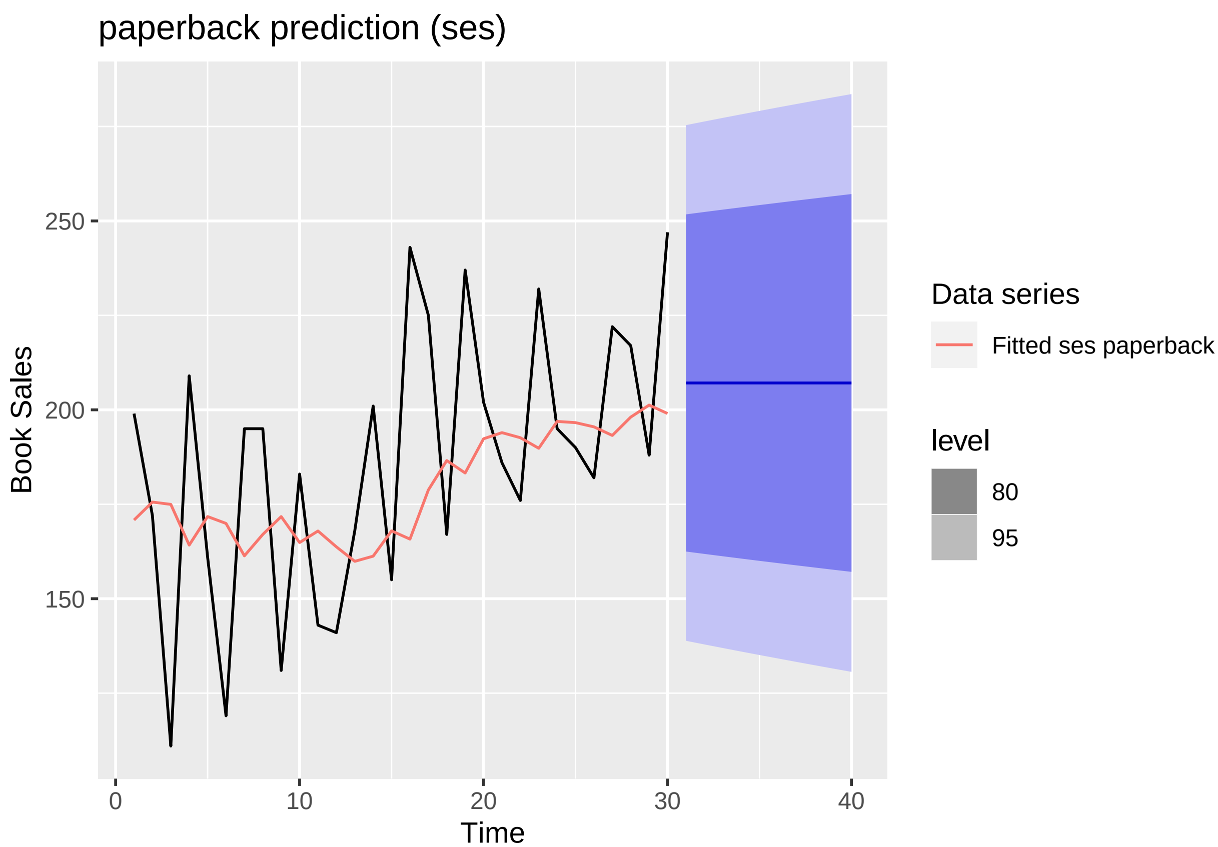
```r
paperback_ses_compare95 = rbind(my_paperback_ses_95,
                                R_paperback_ses_95)
paperback_ses_compare95  %>%
  kable(caption = "Paperback Prediction Intervals (SES)") %>%
  kable_styling(c("bordered","striped"),full_width = F)
```

The intervals computed by R are wider.

Table 4: Paperback Prediction Intervals (SES)

	Lower.95%	Upper.95%
my_paperback_ses_95	141.1798	273.0395
R_paperback_ses_95	138.8670	275.3523

```
autoplot(paperback_forecast_ses) +
autolayer(fitted(paperback_forecast_ses), series = "Fitted ses paperback") +
ggtitle("paperback prediction (ses)") +
  xlab("Time") +
  ylab("Book Sales") +
  guides(colour=guide_legend(title="Data series"),
         fill=guide_legend(title="Prediction interval"))
```



The intervals computed by R are wider than those computed using the RMSE.

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 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.]

```r
# Best model:
fc <- holt(eggs, h=100, lambda=0.04, damped=FALSE)
accuracy(fc)
```

```
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.7425725 26.35977 19.05123 -1.171722 9.687307 0.9397685
##                    ACF1
## Training set 0.02850751
```

```r
indexes=1:1000
n=length(indexes)
lambda_grid = rep(0,n)
ME_grid = rep(0,n)
RMSE_grid = rep(0,n)
MAE_grid = rep(0,n)
MPE_grid = rep(0,n)
MAPE_grid = rep(0,n)
MASE_grid = rep(0,n)
ACF1_grid = rep(0,n)
mse_grid = rep(0,n)
amse_grid = rep(0,n)
meanresid2 = rep(0,n)
sqrtmeanresid2 = rep(0,n)
for (i in indexes) {
  lambda_grid[i] = i/1000
  result = holt(eggs, h=100, lambda=lambda_grid[i], damped=FALSE)
  result.acc = accuracy(result)
  #print(c(lambda_grid[i],result.acc))
  ME_grid[i] = result.acc[1,"ME"]
  RMSE_grid[i] = result.acc[1,"RMSE"]
  MAE_grid[i] = result.acc[1,"MAE"]
  MPE_grid[i] = result.acc[1,"MPE"]
  MAPE_grid[i] = result.acc[1,"MAPE"]
  MASE_grid[i] = result.acc[1,"MASE"]
  ACF1_grid[i] = result.acc[1,"ACF1"]
  mse_grid[i] = result$model$mse
  amse_grid[i] = result$model$amse
  meanresid2[i] = mean(result$residuals^2)
  sqrtmeanresid2[i] = sqrt(mean(result$residuals^2))
```

```
}

biggrid <- cbind(lambda=lambda_grid,
      ME=ME_grid,
      RMSE=RMSE_grid,
      MAE=MAE_grid,
      MPE=MPE_grid,
      MAPE=MAPE_grid,
      MASE=MASE_grid,
      ACF1=ACF1_grid,
      mse=mse_grid,
      amse=amse_grid,
      meanresid2=meanresid2,
      sqrtmeanresid2=sqrtmeanresid2)

minRMSE = min(RMSE_grid)
whichlambda = which(RMSE_grid==minRMSE)
print(paste("The minimum RMSE = ", minRMSE," occurs when lambda = ", lambda_grid[whichlambda]))
```

Which model gives the best RMSE?

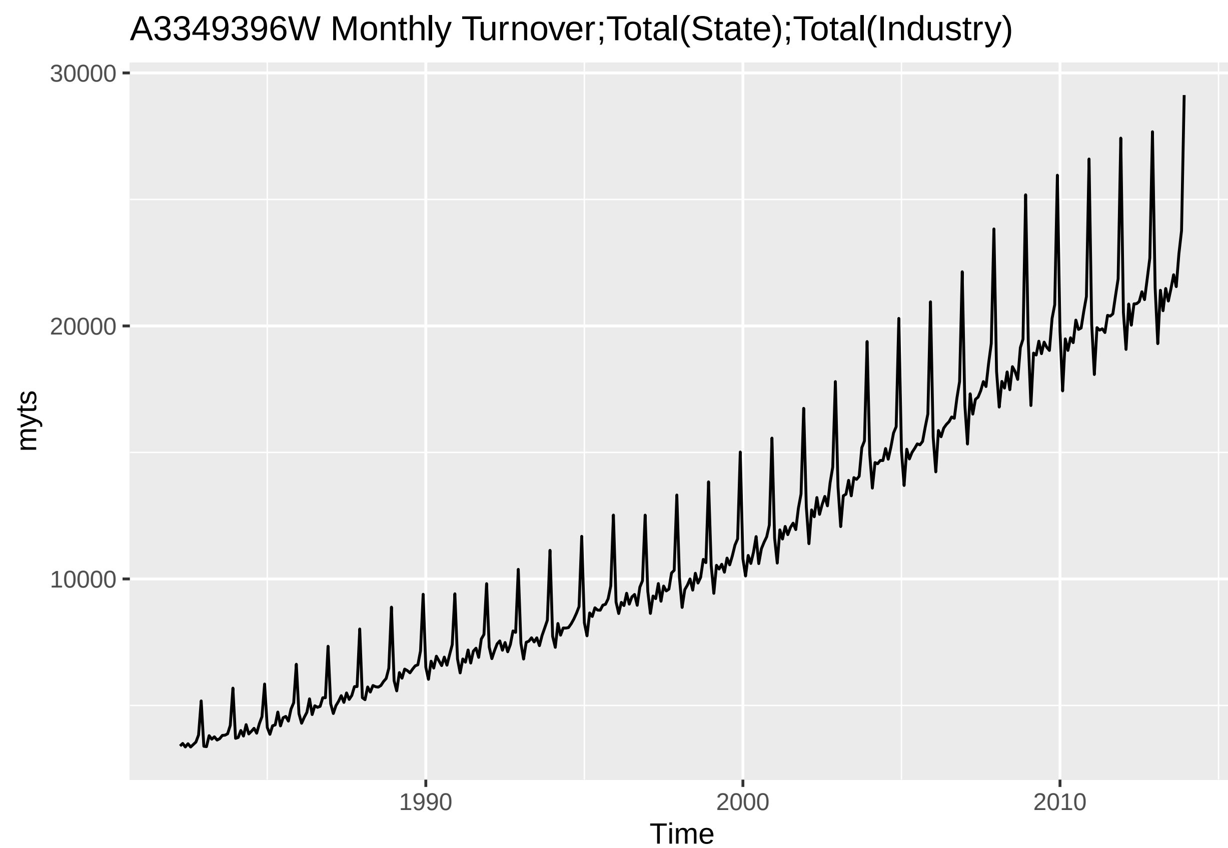
```
## [1] "The minimum RMSE =  26.3597707115837  occurs when lambda =  0.04"
```

The minimum RMSE = 26.3597707 occurs when lambda = 0.04

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

a) Why is multiplicative seasonality necessary for this series?

```
mycode <- "A3349396W"
mytitle <-  "Monthly Turnover;Total(State);Total(Industry)"
mymain <- paste(mycode,mytitle)
myts <- readxl::read_excel("retail.xlsx", skip=1)[,mycode] %>%
  ts(frequency=12, start=c(1982,4))
autoplot(myts,main=mymain)
```



The seasonal variation increases with the level of the series. Therefore, we need to use multiplicative seasonality.

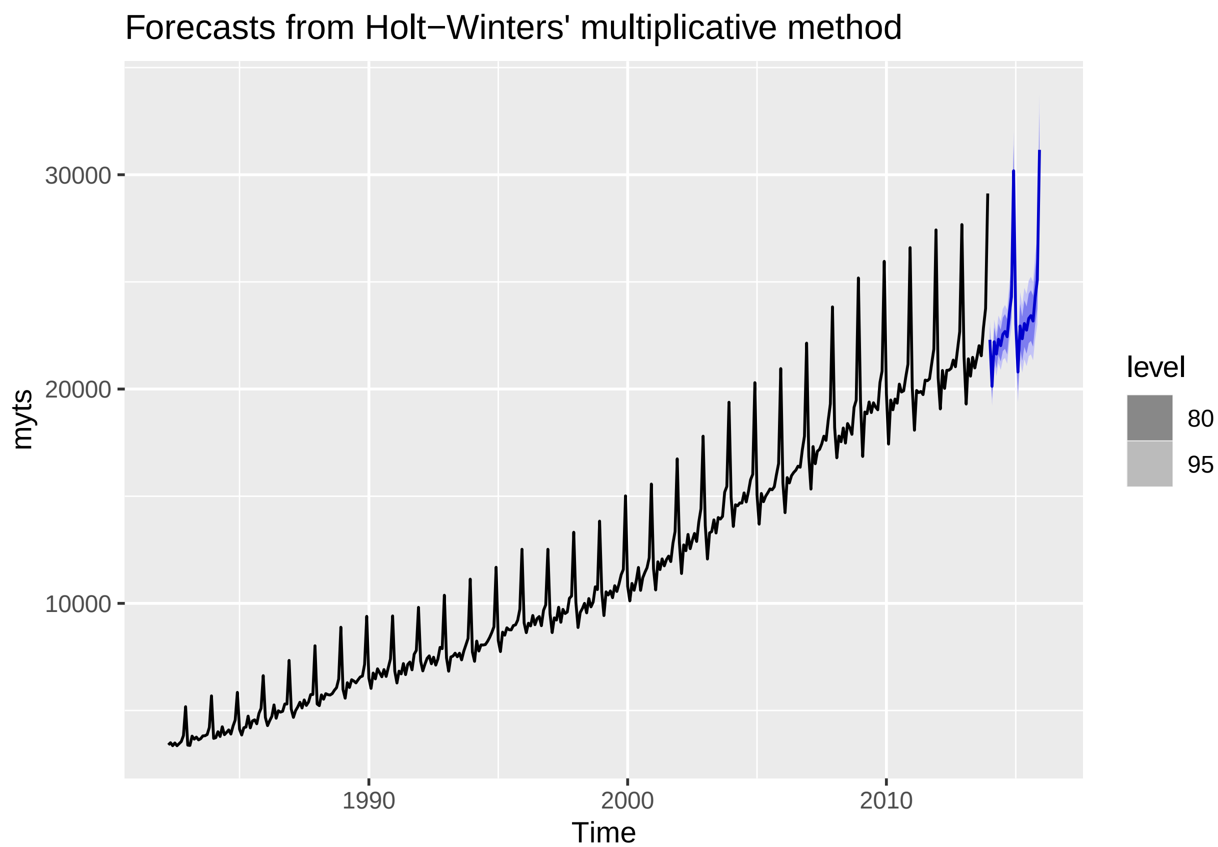
b) Apply Holt-Winters' multiplicative method to the data.

```
HoltWintersMult1 <- hw(myts, seasonal='multiplicative', damped=FALSE)
HoltWintersMult1$mean
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2014 22297.15 20124.85 22208.45 21632.11 22323.80 22023.66 22553.45 22687.24
## 2015 23039.88 20793.37 22944.16 22346.76 23059.27 22747.26 23292.44 23428.59
##           Sep      Oct      Nov      Dec
```

```
## 2014 22440.53 23527.65 24298.79 30185.18
## 2015 23171.84 24292.31 25086.38 31160.95
```

```
autoplot(HoltWintersMult1)
```

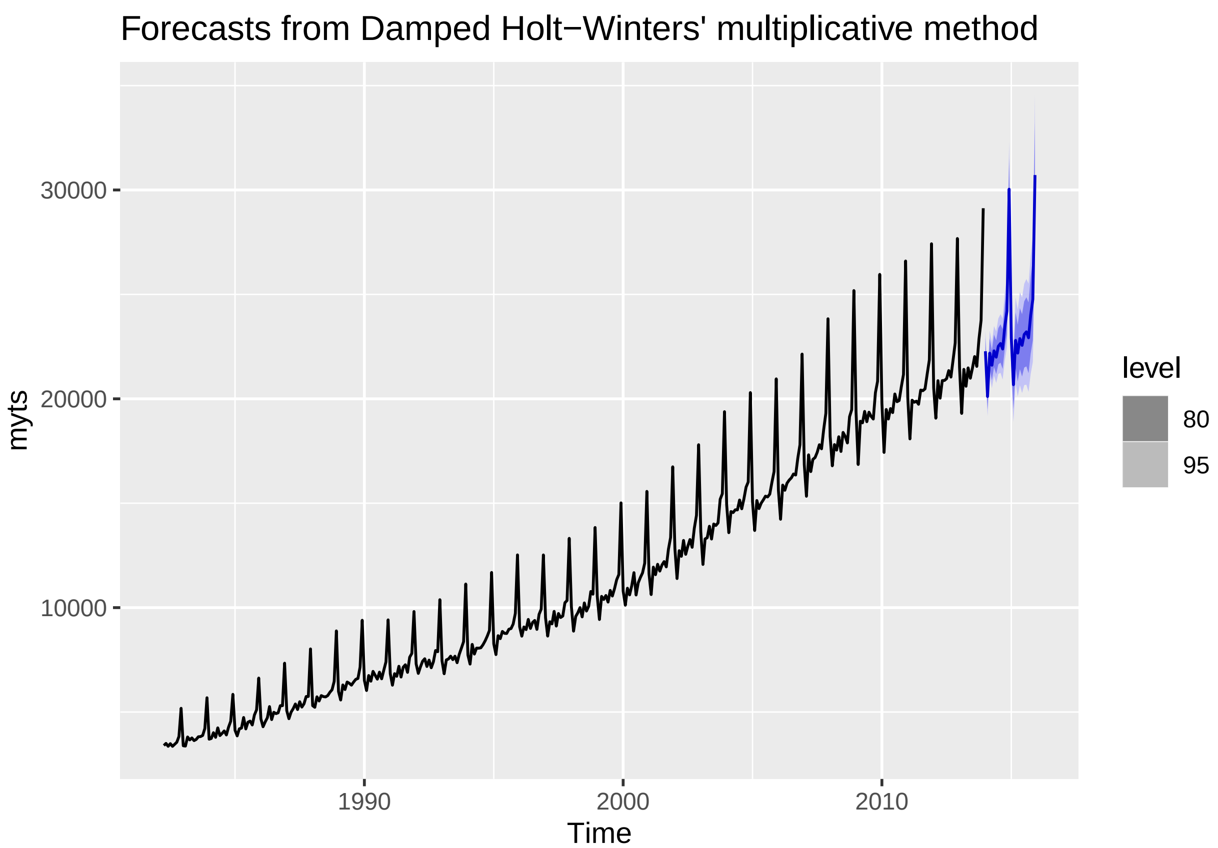


Experiment with making the trend damped.

```
HoltWintersMultDamped <- hw(myts, seasonal='multiplicative', damped=T)
HoltWintersMultDamped$mean
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2014 22276.11 20108.13 22190.12 21606.22 22296.49 21992.63 22515.73 22649.45
## 2015 22918.44 20674.87 22801.46 22188.11 22883.52 22558.73 23082.38 23206.80
##           Sep      Oct      Nov      Dec
## 2014 22383.95 23452.83 24209.06 30039.58
## 2015 22922.55 24004.66 24766.10 30715.56
```

```
autoplot(HoltWintersMultDamped)
```



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

```
print("Holt-Winters Multiplicative (not damped):")
```

```
## [1] "Holt-Winters Multiplicative (not damped):"
```

```
accuracy(HoltWintersMult1)
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 8.069135 221.5921 173.1308 0.05953008 1.726817 0.2859523
##                    ACF1
## Training set -0.1631927
```

```
print("Holt-Winters Multiplicative Damped:")
```

```
## [1] "Holt-Winters Multiplicative Damped:"
```

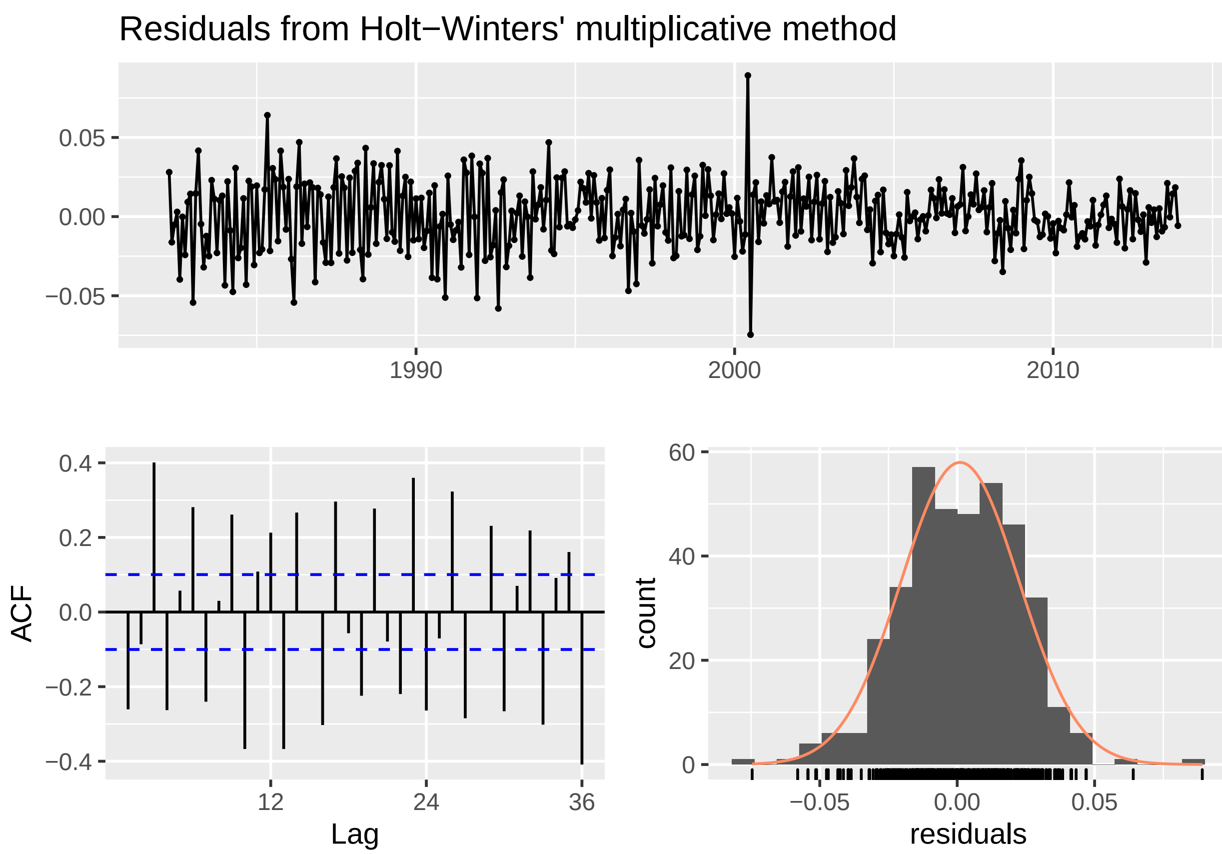
```
accuracy(HoltWintersMultDamped)
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
## Training set 30.12733 224.6438 175.4635 0.2717607 1.762675 0.2898051 -0.1748363
```

The RMSE is slightly lower on the non-damped model, which I would prefer as the trend is upward.

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

```r
checkresiduals(HoltWintersMult1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Holt-Winters' multiplicative method
## Q* = 584.69, df = 8, p-value < 0.00000000000000022
##
## Model df: 16.   Total lags used: 24
```

There are significant correlations in the residuals.

There appears to be a quarterly pattern, as sales may be affected by whether one is in the beginning or the end of each quarter, with substantial reversion.

Therefore, these residuals do not look like white noise.

e) Now find the test set RMSE, while training the model to the end of 2010.

```r
myts %>%
  window(end=c(2010,12)) %>%
```

```
  hw(seasonal='multiplicative', damped=FALSE) -> myresults

myresults
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2011       20349.84 19754.77 20944.92 19439.75 21259.94
## Feb 2011       18202.63 17648.89 18756.37 17355.76 19049.50
## Mar 2011       20151.23 19514.46 20787.99 19177.38 21125.07
## Apr 2011       19756.25 19108.67 20403.82 18765.87 20746.62
## May 2011       20332.60 19642.14 21023.05 19276.64 21388.56
## Jun 2011       20020.07 19316.57 20723.56 18944.17 21095.97
## Jul 2011       20609.79 19861.19 21358.39 19464.90 21754.68
## Aug 2011       20447.17 19680.24 21214.10 19274.25 21620.09
## Sep 2011       20381.22 19592.56 21169.88 19175.07 21587.37
## Oct 2011       21440.46 20585.29 22295.62 20132.60 22748.32
## Nov 2011       22050.76 21144.95 22956.57 20665.45 23436.08
## Dec 2011       27761.87 26588.26 28935.47 25966.99 29556.74
## Jan 2012       21127.55 20167.35 22087.76 19659.05 22596.06
## Feb 2012       18896.07 18015.53 19776.61 17549.41 20242.74
## Mar 2012       20916.48 19917.60 21915.36 19388.83 22444.14
## Apr 2012       20504.14 19501.12 21507.16 18970.16 22038.12
## May 2012       21099.90 20043.11 22156.69 19483.67 22716.12
## Jun 2012       20773.21 19708.42 21838.00 19144.75 22401.66
## Jul 2012       21382.70 20261.48 22503.91 19667.94 23097.45
## Aug 2012       21211.60 20074.26 22348.93 19472.19 22951.00
## Sep 2012       21140.82 19982.16 22299.48 19368.81 22912.84
## Oct 2012       22237.06 20991.80 23482.33 20332.59 24141.53
## Nov 2012       22867.52 21559.56 24175.48 20867.17 24867.88
## Dec 2012       28787.01 27105.85 30468.17 26215.90 31358.12
```

```
accuracy(myresults,x=myts)
```

```
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set  10.07947 215.6308 168.7072  0.06360609 1.805918 0.2804746
## Test set    -222.58896 353.0499 271.9102 -1.00551589 1.249486 0.4520490
##                   ACF1 Theil's U
## Training set -0.1743335        NA
## Test set     -0.1375517 0.1647206
```

Can you beat the seasonal naïve approach from Exercise 8 in Section 3.7? The test set RMSE is 353.0499 compared to 1389.337 for the seasonal naive method (Homework 2, final problem.) So, on this dataset, the Holt-Winters method is much better that the seasonal naive method.

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.

```
myts %>%
  window(end=c(2010,12)) %>%
  stlf(lambda=0) -> mySTLdecomposition
mySTLdecomposition
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2011       20313.41 19798.92 20841.27 19531.86 21126.24
## Feb 2011       18256.07 17778.98 18745.96 17531.49 19010.59
## Mar 2011       20155.53 19611.66 20714.49 19329.72 21016.63
## Apr 2011       19762.16 19211.22 20328.89 18925.82 20635.45
## May 2011       20324.80 19739.14 20927.83 19435.98 21254.26
## Jun 2011       20045.40 19448.20 20660.93 19139.30 20994.39
## Jul 2011       20590.38 19956.01 21244.91 19628.15 21599.78
## Aug 2011       20523.25 19869.30 21198.71 19531.60 21565.23
## Sep 2011       20469.28 19794.72 21166.82 19446.68 21545.65
## Oct 2011       21547.18 20812.84 22307.43 20434.29 22720.68
## Nov 2011       22162.94 21381.91 22972.49 20979.66 23412.95
## Dec 2011       27832.05 26818.06 28884.38 26296.33 29457.46
## Jan 2012       21099.11 20304.61 21924.70 19896.21 22374.74
## Feb 2012       18962.19 18224.39 19729.86 17845.51 20148.74
## Mar 2012       20935.12 20093.71 21811.77 19662.07 22290.60
## Apr 2012       20526.53 19674.65 21415.29 19238.10 21901.25
## May 2012       21110.93 20206.57 22055.76 19743.63 22572.93
## Jun 2012       20820.72 19900.42 21783.59 19429.83 22311.19
## Jul 2012       21386.79 20411.79 22408.37 19913.78 22968.76
## Aug 2012       21317.06 20315.13 22368.39 19803.95 22945.77
## Sep 2012       21261.00 20231.20 22343.23 19706.40 22938.25
## Oct 2012       22380.60 21263.93 23555.91 20695.54 24202.86
## Nov 2012       23020.17 21837.53 24266.86 21236.27 24953.92
## Dec 2012       28908.56 27380.01 30522.44 26603.84 31412.94
```

```
accuracy(mySTLdecomposition,x=myts)
```

```
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set  -11.38664 192.8008 150.0699 -0.113479 1.602350 0.2494903
## Test set     -275.22587 390.4325 298.4588 -1.246303 1.362937 0.4961859
##                     ACF1 Theil's U
## Training set -0.07696934        NA
## Test set     -0.02333538 0.1839252
```

How does that compare with your best previous forecasts on the test set?

Here the RMSE is 390.4325, which is worse than the 353.0499 obtained from Holt-Winters.
