Predict_413_Sec55_Homework_2

Singh, Gurjeet

February 18, 2018

Contents

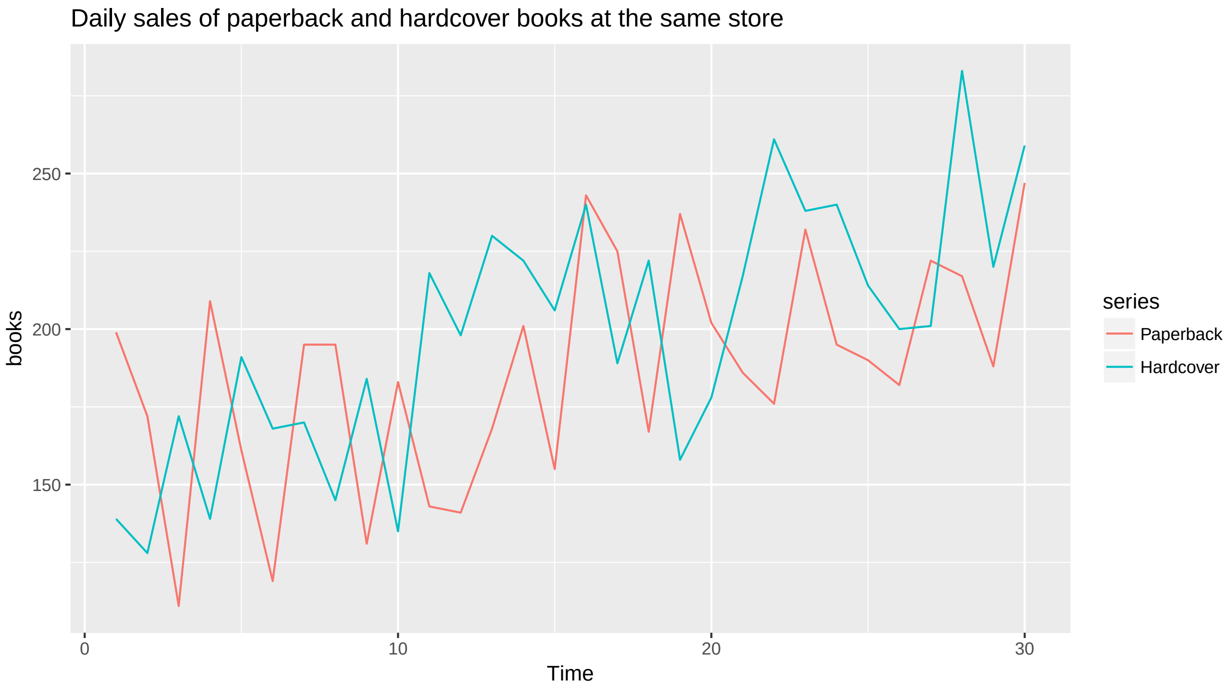
Chapter 7 **2**
Question 1 . . . 2
Ch7.Q1.a) . . . 2
Ch7.Q1.b) . . . 2
Ch7.Q1.c) . . . 3
Ch7.Q1.d) . . . 4
Ch7.Q1.e) . . . 5
Question 2 . . . 8
Ch7.Q2.a) . . . 8
Ch7.Q2.b) . . . 8
Ch7.Q2.c) . . . 9
Question 3 . . . 10
Ch7.Q3) . . . 10
Appendix I: R Code . . . 11

Chapter 7

Question 1

Ch7.Q1.a)

The figure below shows the plot of the daily sales of paperback and hardcover books at the same store. The data in the figure below do not display any seasonality for paperback and hardcover books. However, there could some trend here. We will know more once we plug this data into various models.



Ch7.Q1.b)

The Table 1 below shows the various matrics from each SES models with different alpha. It appears that as alpha increases, the error increases as well. Hence, alpha = 0.2 works best.

The Table 2 shows the SSE for each models using different alpha.

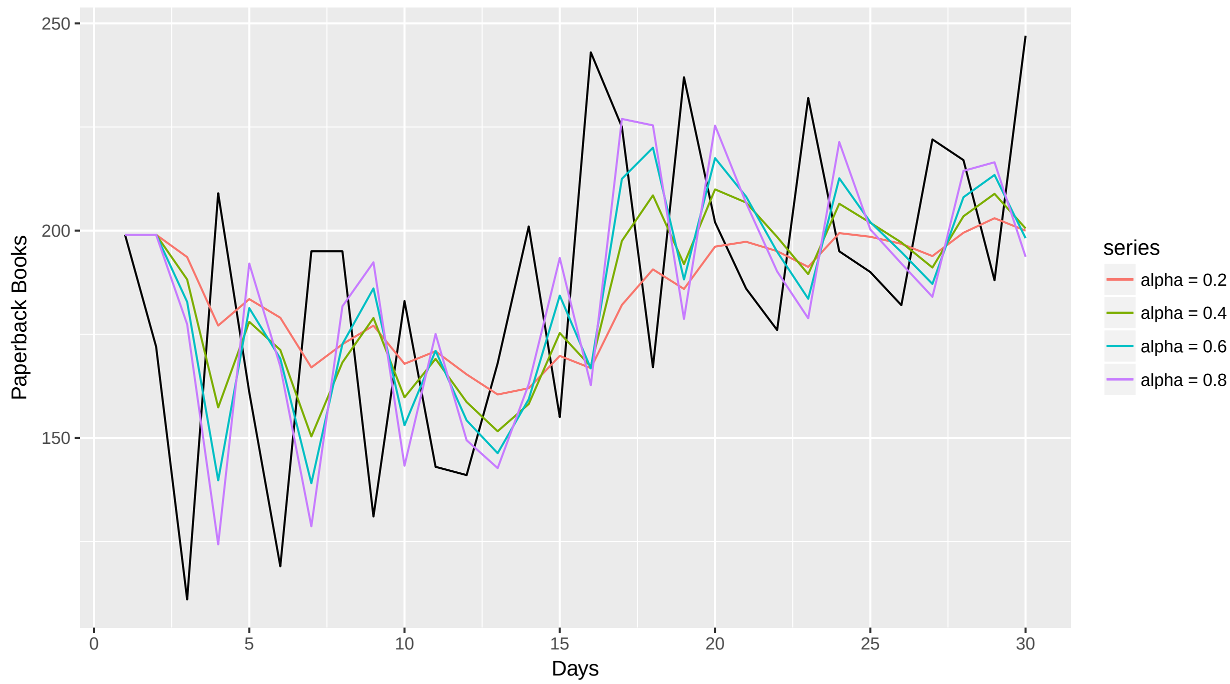
The figure below shows the four different sets of forecasts.

Table 1: Metrics from each model

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
alpha = 0.2	1.731375	34.79911	28.51298	-2.805469	16.51268	0.7190230	-0.1128428
alpha = 0.4	1.676073	35.93438	30.83034	-2.645121	17.64277	0.7774607	-0.2758328
alpha = 0.6	1.581515	38.61891	33.06412	-2.719647	18.81258	0.8337908	-0.3685653
alpha = 0.8	1.555801	42.21222	35.32225	-2.795040	19.96585	0.8907351	-0.4311305

Table 2: SSE values of each model

	alpha = 0.2	alpha = 0.4	alpha = 0.6	alpha = 0.8
SSE	36329.34	38738.4	44742.62	53456.14



Ch7.Q1.c)

The summary statistic below is from SES model selected the optimal value of alpha. As we mentioned earlier that lower value of alpha gives us the best model, it is quite evident with the result that alpha = 0.1685 gave us the lowest RMSE and SSE.

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = books[, 1], h = 4)
##
##   Smoothing parameters:
##     alpha = 0.1685
##
##   Initial states:
##     l = 170.8257
##
##   sigma:  33.6377
##
##      AIC     AICc      BIC
## 318.9747 319.8978 323.1783
```

```
##
## Error measures:
##                    ME     RMSE     MAE       MPE     MAPE      MASE
## Training set 7.176212 33.63769 27.8431 0.4737524 15.57782 0.7021303
##                    ACF1
## Training set -0.2117579
##
## Forecasts:
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       207.1098 164.0013 250.2182 141.1811 273.0384
## 32       207.1098 163.3934 250.8261 140.2513 273.9682
## 33       207.1098 162.7937 251.4258 139.3342 274.8853
## 34       207.1098 162.2021 252.0174 138.4294 275.7901
```

Table 3: SSE - SES Select

	alpha = 0.1685
SSE Value	33944.82

Ch7.Q1.d)

The summary statistic below shows that with the initial = "optimal" option, we get the same alpha and initial states. There's no difference from SES selecting an optimal value without the option and after setting the optimal option.

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = books[, 1], h = 4, initial = "optimal")
##
##   Smoothing parameters:
##     alpha = 0.1685
##
##   Initial states:
##     l = 170.8257
##
##   sigma:  33.6377
##
##      AIC     AICc      BIC
## 318.9747 319.8978 323.1783
##
## Error measures:
##                    ME     RMSE     MAE       MPE     MAPE      MASE
## Training set 7.176212 33.63769 27.8431 0.4737524 15.57782 0.7021303
##                    ACF1
## Training set -0.2117579
##
## Forecasts:
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
```

```
## 31       207.1098 164.0013 250.2182 141.1811 273.0384
## 32       207.1098 163.3934 250.8261 140.2513 273.9682
## 33       207.1098 162.7937 251.4258 139.3342 274.8853
## 34       207.1098 162.2021 252.0174 138.4294 275.7901
```

Table 4: SSE - Optimal Select

	alpha = 0.1685
SSE Value	33944.82

Ch7.Q1.e)

We run SES model for Hardcover books. The Table 5 below shows the various matrics from each SES models with different alpha. When alpha = 0.4, the RMSE is the lowest. Hence, forecast works best.

The Table 6 shows the SSE for each models using different alpha. It is clear that alpha = 0.4 gives us the best forecast.

The figure below shows the four different sets of forecasts.

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = books[, 2], h = 4, initial = "simple", alpha = 0.4)
##
##   Smoothing parameters:
##     alpha = 0.4
##
##   Initial states:
##     l = 139
##
##   sigma:  32.0912
## Error measures:
##                   ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 8.62003 32.09116 26.19605 2.540241 13.01568 0.7815695
##                    ACF1
## Training set -0.1952743
##
## Forecasts:
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       242.4404 201.3139 283.5668 179.5428 305.3379
## 32       242.4404 198.1458 286.7349 174.6977 310.1830
## 33       242.4404 195.1896 289.6911 170.1766 314.7041
## 34       242.4404 192.4078 292.4729 165.9222 318.9585
```

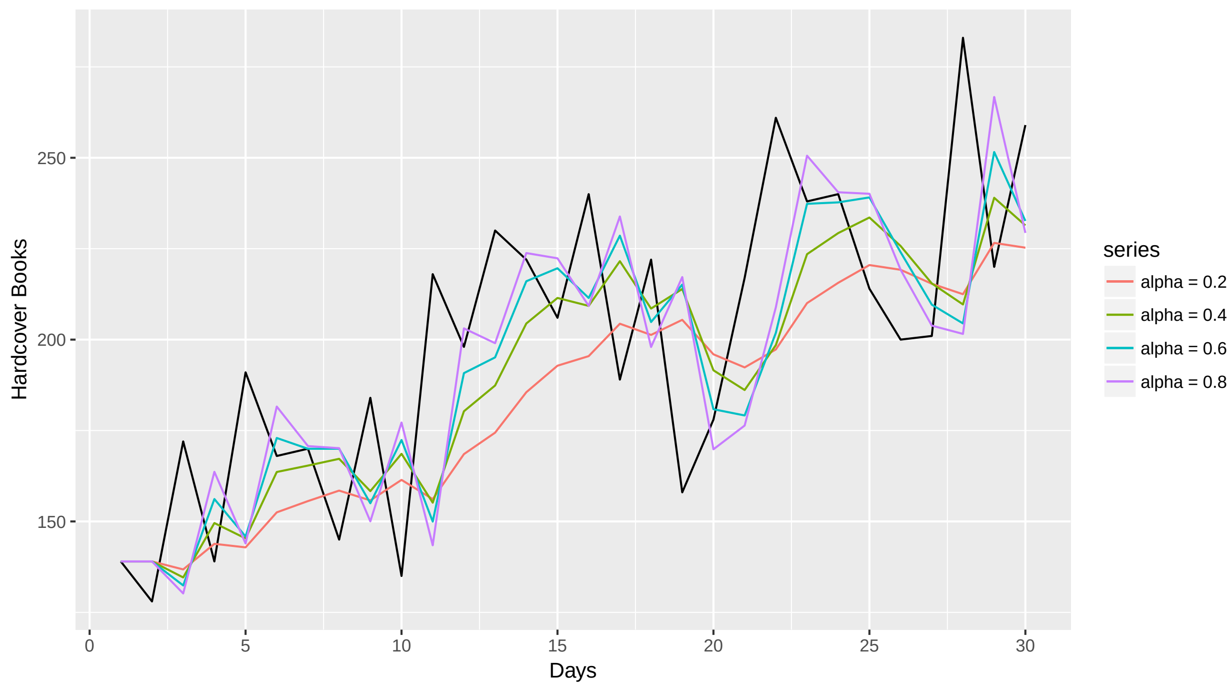
Table 5: Metrics from each model

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
alpha = 0.2	15.502918	33.24062	27.71224	6.0168595	13.48065	0.8268056	-0.0826814
alpha = 0.4	8.620030	32.09116	26.19605	2.5402410	13.01568	0.7815695	-0.1952743

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
alpha = 0.6	6.080697	33.19635	25.94767	1.2503379	13.01841	0.7741589	-0.3338129
alpha = 0.8	4.752856	35.42212	28.25713	0.5435556	14.27080	0.8430625	-0.4797522

Table 6: SSE values of each model

	alpha = 0.2	alpha = 0.4	alpha = 0.6	alpha = 0.8
SSE	33148.16	30895.27	33059.93	37641.79



The summary statistic below is from SES model selected the optimal value of alpha. The model selects the optimal value of alpha = 0.3283. Forecast values are lower than than the results in 2 for alpha = 0.4.

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = books[, 2], h = 4)
##
##   Smoothing parameters:
##     alpha = 0.3283
##
##   Initial states:
##     l = 149.2836
##
##   sigma:  31.931
##
```

```
##      AIC     AICc      BIC
## 315.8506 316.7737 320.0542
##
## Error measures:
##                    ME     RMSE     MAE      MPE     MAPE      MASE
## Training set 9.166918 31.93101 26.7731 2.636328 13.39479 0.7987858
##                    ACF1
## Training set -0.1417817
##
## Forecasts:
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       239.5602 198.6390 280.4815 176.9766 302.1439
## 32       239.5602 196.4905 282.6299 173.6908 305.4297
## 33       239.5602 194.4443 284.6762 170.5613 308.5591
## 34       239.5602 192.4869 286.6336 167.5677 311.5527
```

Table 7: SSE - SES Select

	alpha = 0.1685
SSE Value	30587.69

The summary statistic below shows that with the initial = "optimal" option, we get the same alpha and initial states. There's no difference from SES selecting an optimal value without the option and after setting the optimal option.

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = books[, 2], h = 4, initial = "optimal")
##
##   Smoothing parameters:
##     alpha = 0.3283
##
##   Initial states:
##     l = 149.2836
##
##   sigma:  31.931
##
##      AIC     AICc      BIC
## 315.8506 316.7737 320.0542
##
## Error measures:
##                    ME     RMSE     MAE      MPE     MAPE      MASE
## Training set 9.166918 31.93101 26.7731 2.636328 13.39479 0.7987858
##                    ACF1
## Training set -0.1417817
##
## Forecasts:
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 31       239.5602 198.6390 280.4815 176.9766 302.1439
```

```
## 32       239.5602 196.4905 282.6299 173.6908 305.4297
## 33       239.5602 194.4443 284.6762 170.5613 308.5591
## 34       239.5602 192.4869 286.6336 167.5677 311.5527
```

Table 8: SSE - Optimal Select

	alpha = 0.1685
SSE Value	30587.69

Question 2

Ch7.Q2.a)

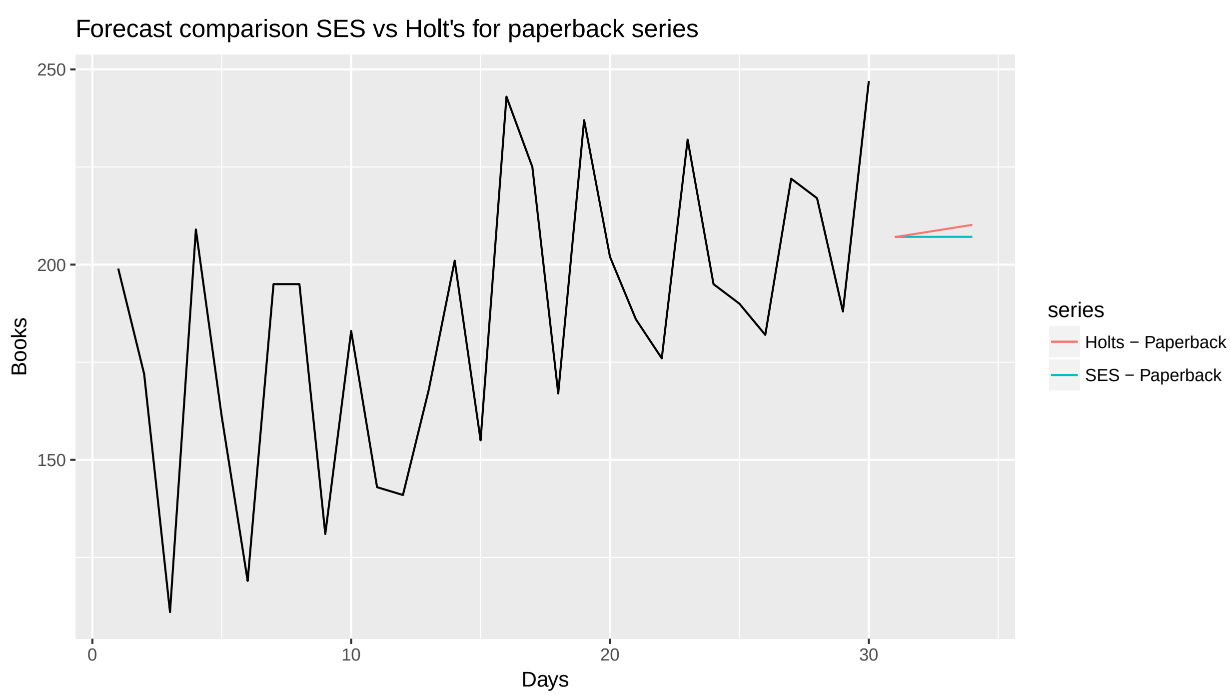
Table 9 shows the SSE value of paperback and hardback series obtained using Holt's linear method. These measures are much better than any of the SSE values obtained using SES. Earlier, we had doubts that data may have some trend behaviour. The result confirms that there's definitely trend behaviour present in the data. Hence, simple exponential smoothing may not be the best option for this kind of data.

Table 9: SSE Values

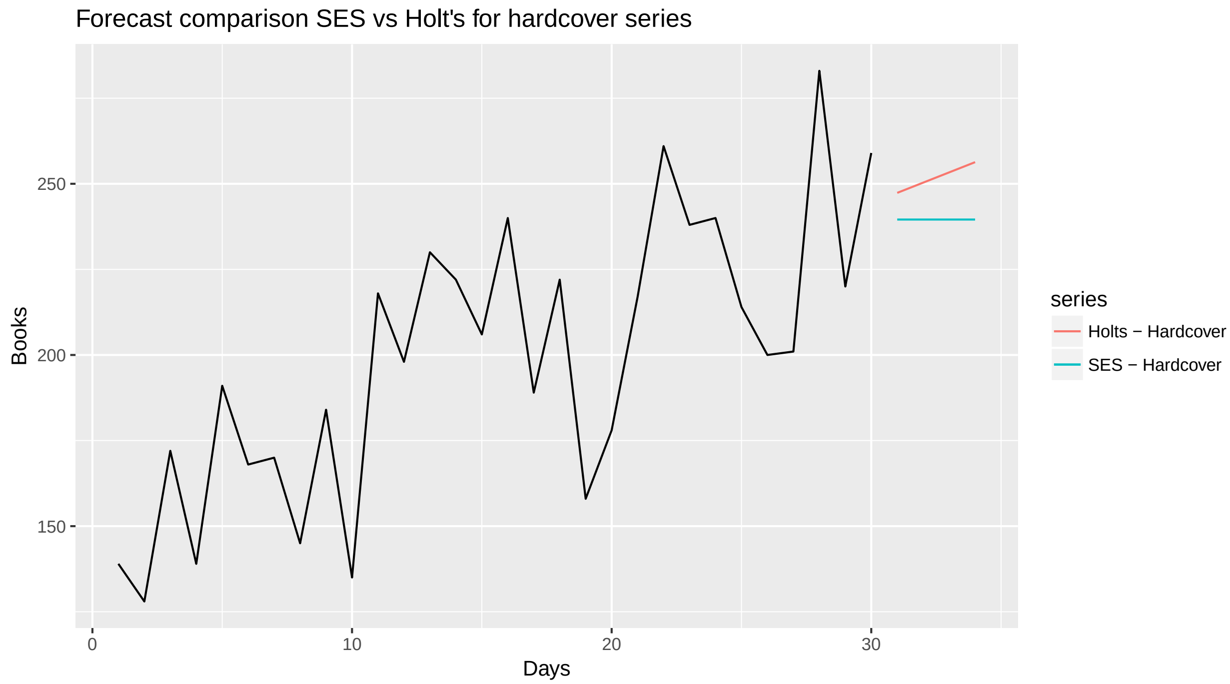
	Holt's Linear - Paperback	Holt's Linear - Hardcover
SSE Value	30074.17	22581.83

Ch7.Q2.b)

The figure below shows the forecast of Simple exponential smoothing and Holt's linear methods for paperback books. It appears that Holt's linear method performed better. Forecast from SES seems to be flat.

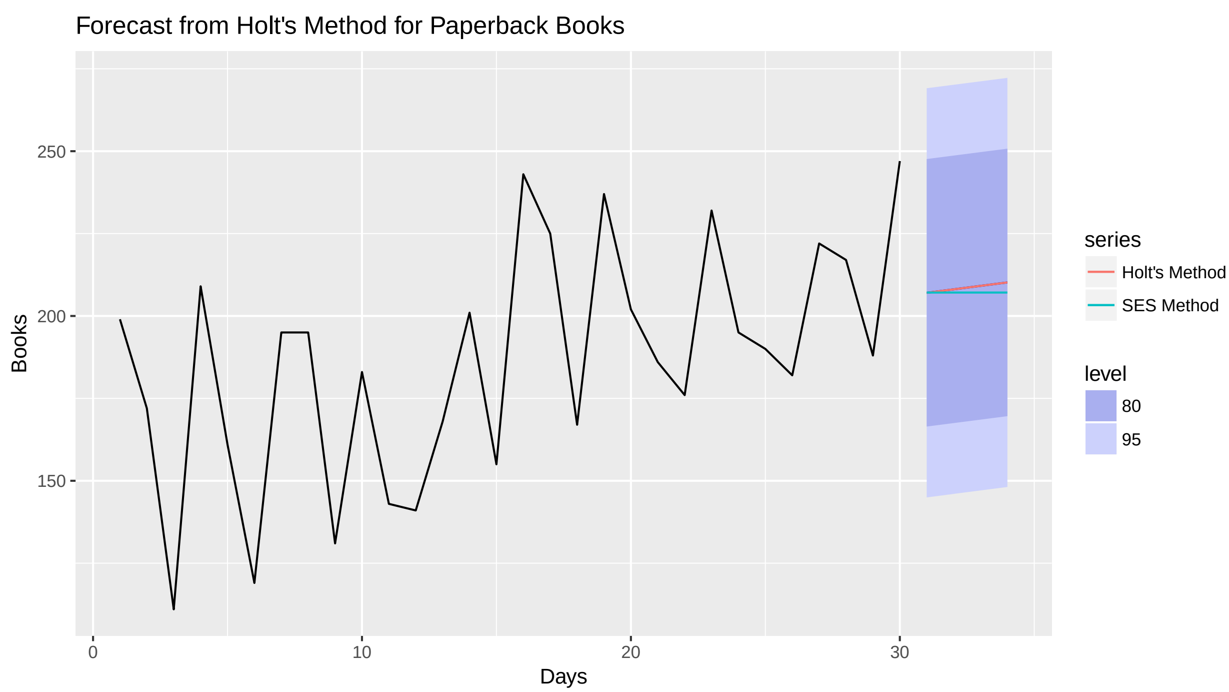


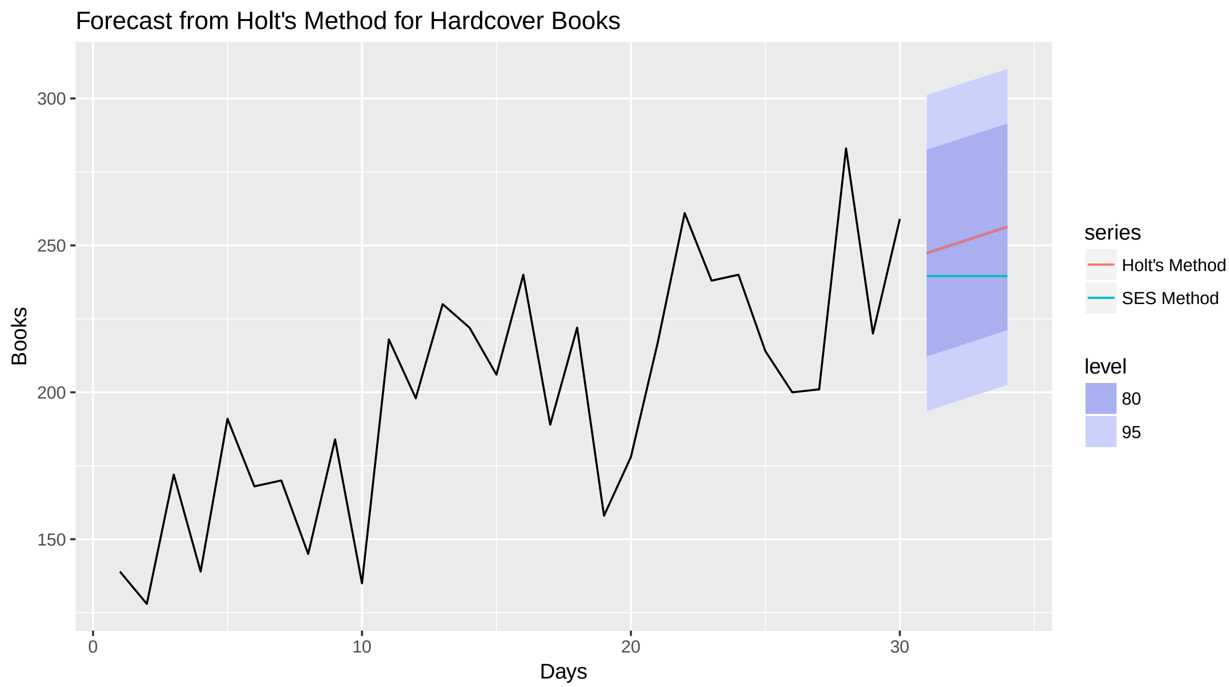
The figure below shows the forecast of Simple exponential smoothing and Holt's linear methods for hardcover books. Again, it appears that Holt's linear method performed better.



Ch7.Q2.c)

The figures below shows the 95% prediction interval for the forecase for each series using Holt's and Simple Exponential Smoothing methods. Both the methods are forecasting within the prediction interval.





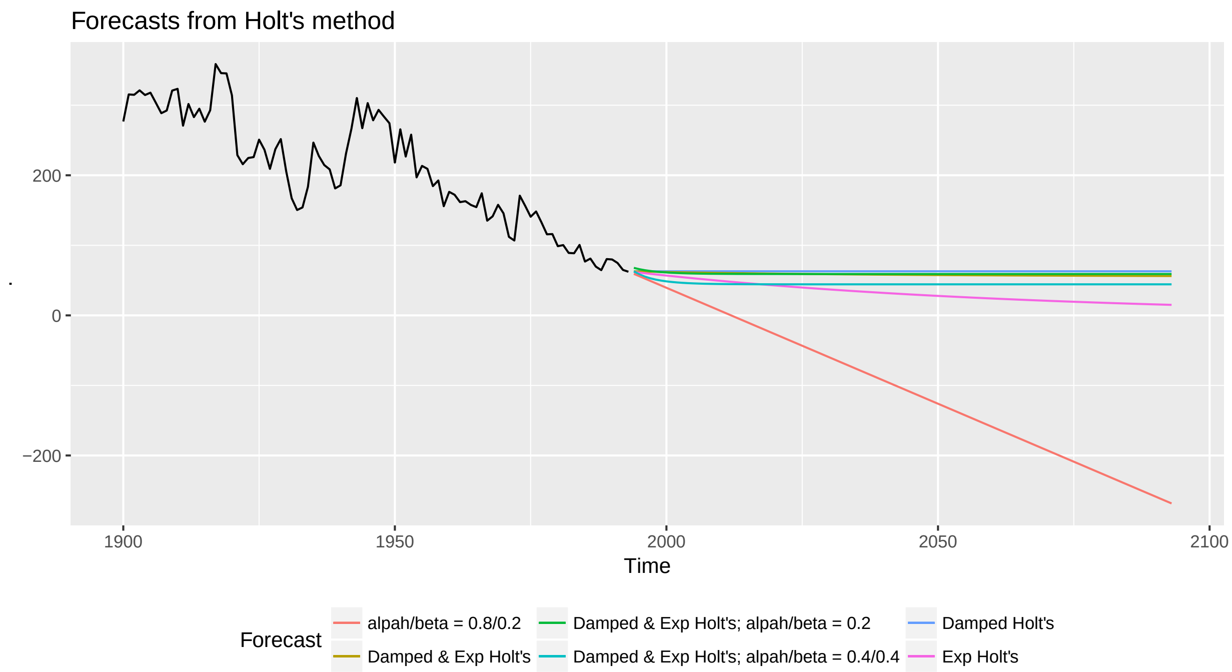
Question 3

Ch7.Q3)

The table below give the metrics from each model. It shows that model with exponential trend gave the best RSME of 26.386. The figure below shows the forecast of each model. Again, it is quite evident that exponential trend forecase seems to be a lot better as compared to others.

Table 10: Metrics for each model

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Damped Holt's	-3.092	26.662	19.512	-3.023	10.110	0.962	-0.006
Damped & Exp Holt's	-0.882	26.526	19.514	-2.101	10.015	0.963	0.005
Exp Holt's	0.476	26.386	19.222	-1.280	9.754	0.948	0.007
Damped & Exp Holt's; alpah/beta = 0.2	-4.943	32.458	23.215	-4.007	12.019	1.145	0.477
Damped & Exp Holt's; alpah/beta = 0.4/0.4	-3.814	32.123	23.592	-2.988	12.041	1.164	0.288
alpah/beta = 0.8/0.2	-0.545	28.825	21.759	-0.793	10.928	1.073	0.017



Appendix I: R Code

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE )
library(fpp)
library(tidyverse)

### Ch7.Q1.a)
data("books")
autoplot(books) +
  ggtitle("Daily sales of paperback and hardcover books at the same store")

### Ch7.Q1.b)
fit1_pb <- ses(books[,1], initial = "simple", alpha = 0.2, h=4)
SSE_fit1 <- (accuracy(fit1_pb)[2]^2)*30

fit2_pb <- ses(books[,1], initial = "simple", alpha = 0.4, h=4)
SSE_fit2 <- (accuracy(fit2_pb)[2]^2)*30

fit3_pb <- ses(books[,1], initial = "simple", alpha = 0.6, h=4)
SSE_fit3 <- (accuracy(fit3_pb)[2]^2)*30

fit4_pb <- ses(books[,1], initial = "simple", alpha = 0.8, h=4)
SSE_fit4 <- (accuracy(fit4_pb)[2]^2)*30

Metrics_Values <- rbind.data.frame(accuracy(fit1_pb), accuracy(fit2_pb), accuracy(fit3_pb),
                                   accuracy(fit4_pb))
rownames(Metrics_Values) <- c("alpha = 0.2", "alpha = 0.4", "alpha = 0.6", "alpha = 0.8")
knitr::kable(Metrics_Values, caption = "Metrics from each model")

SSE_Values <- cbind.data.frame(SSE_fit1, SSE_fit2, SSE_fit3, SSE_fit4)
```

```r
colnames(SSE_Values) <- c("alpha = 0.2", "alpha = 0.4", "alpha = 0.6", "alpha = 0.8")
rownames(SSE_Values) <- "SSE"
knitr::kable(SSE_Values, caption = "SSE values of each model")

autoplot(books[,1]) +
  autolayer(fitted(fit1_pb), series = "alpha = 0.2") +
autolayer(fitted(fit2_pb), series = "alpha = 0.4") +
autolayer(fitted(fit3_pb), series = "alpha = 0.6") +
  autolayer(fitted(fit4_pb), series = "alpha = 0.8") +
  xlab("Days") + ylab("Paperback Books")

### Ch7.Q1.c)
fit5_pb <- ses(books[,1], h=4)
summary(fit5_pb)
SSE_fit5 <- data.frame((accuracy(fit5_pb)[2]^2)*30)
colnames(SSE_fit5) <- c("alpha = 0.1685")
rownames(SSE_fit5) <- "SSE Value"
knitr::kable(SSE_fit5, caption = "SSE - SES Select")

### Ch7.Q1.d)
fit6_pb <- ses(books[,1], initial = "optimal" ,h=4)
summary(fit6_pb)
SSE_fit6 <- data.frame((accuracy(fit6_pb)[2]^2)*30)
colnames(SSE_fit6) <- c("alpha = 0.1685")
rownames(SSE_fit6) <- "SSE Value"
knitr::kable(SSE_fit6, caption = "SSE - Optimal Select")

### Ch7.Q1.e) - Part B
fit1_hc <- ses(books[,2], initial = "simple", alpha = 0.2, h=4)
#summary(fit1_pb)
SSE_fit1_hc <- (accuracy(fit1_hc)[2]^2)*30

fit2_hc <- ses(books[,2], initial = "simple", alpha = 0.4, h=4)
SSE_fit2_hc <- (accuracy(fit2_hc)[2]^2)*30
summary(fit2_hc)
fit3_hc <- ses(books[,2], initial = "simple", alpha = 0.6, h=4)
SSE_fit3_hc <- (accuracy(fit3_hc)[2]^2)*30

fit4_hc <- ses(books[,2], initial = "simple", alpha = 0.8, h=4)
SSE_fit4_hc <- (accuracy(fit4_hc)[2]^2)*30

Metrics_Values_hc <- rbind.data.frame(accuracy(fit1_hc), accuracy(fit2_hc), accuracy(fit3_hc),
                                      accuracy(fit4_hc))
rownames(Metrics_Values_hc) <- c("alpha = 0.2", "alpha = 0.4", "alpha = 0.6", "alpha = 0.8")
knitr::kable(Metrics_Values_hc, caption = "Metrics from each model")

SSE_Values_hc <- cbind.data.frame(SSE_fit1_hc, SSE_fit2_hc, SSE_fit3_hc, SSE_fit4_hc)
colnames(SSE_Values_hc) <- c("alpha = 0.2", "alpha = 0.4", "alpha = 0.6", "alpha = 0.8")
rownames(SSE_Values_hc) <- "SSE"
knitr::kable(SSE_Values_hc, caption = "SSE values of each model")

autoplot(books[,2]) +
  autolayer(fitted(fit1_hc), series = "alpha = 0.2") +
```

```
autolayer(fitted(fit2_hc), series = "alpha = 0.4") +
autolayer(fitted(fit3_hc), series = "alpha = 0.6") +
  autolayer(fitted(fit4_hc), series = "alpha = 0.8") +
  xlab("Days") + ylab("Hardcover Books")

### Ch7.Q1.e) - Part C
fit5_hc <- ses(books[,2], h=4)
summary(fit5_hc)
SSE_fit5_hc <- data.frame((accuracy(fit5_hc)[2]^2)*30)
colnames(SSE_fit5_hc) <- c("alpha = 0.1685")
rownames(SSE_fit5_hc) <- "SSE Value"
knitr::kable(SSE_fit5_hc, caption = "SSE - SES Select")

### Ch7.Q1.e) - Part D
fit6_hc <- ses(books[,2], initial = "optimal" ,h=4)
summary(fit6_hc)
SSE_fit6_hc <- data.frame((accuracy(fit6_hc)[2]^2)*30)
colnames(SSE_fit6_hc) <- c("alpha = 0.1685")
rownames(SSE_fit6_hc) <- "SSE Value"
knitr::kable(SSE_fit6_hc, caption = "SSE - Optimal Select")

### Ch7.Q2.a)
fit1_holts <- holt(books[,1], h = 4)
SSE_holts <- data.frame(sum(fit1_holts$residuals^2))
colnames(SSE_holts) <- c("Holt's Linear - Paperback")
rownames(SSE_holts) <- "SSE Value"

fit2_holts <- holt(books[,2], h = 4)
SSE_holts_2 <- data.frame(sum(fit2_holts$residuals^2))
colnames(SSE_holts_2) <- c("Holt's Linear - Hardcover")
rownames(SSE_holts_2) <- "SSE Value"

knitr::kable(cbind(SSE_holts,SSE_holts_2) , caption = "SSE Values")

### Ch7.Q2.b)
autoplot(books[,1]) +
  autolayer(fit5_pb$mean, series = "SES - Paperback") +
  autolayer(fit1_holts$mean, series = "Holts - Paperback") +
  ggtitle("Forecast comparison SES vs Holt's for paperback series") +
  xlab("Days") +
  ylab("Books")
### Ch7.Q2.b)
autoplot(books[,2]) +
  autolayer(fit5_hc$mean, series = "SES - Hardcover") +
  autolayer(fit2_holts$mean, series = "Holts - Hardcover") +
  ggtitle("Forecast comparison SES vs Holt's for hardcover series") +
  xlab("Days") +
  ylab("Books")

### Ch7.Q2.c)
autoplot(fit1_holts) +
    autolayer(fit1_holts$mean, series = "Holt's Method") +
    autolayer(fit5_pb$mean, series = "SES Method") +
```

```
  ggtitle("Forecast from Holt's Method for Paperback Books") +
  xlab("Days") +
  ylab("Books")

autoplot(fit2_holts) +
    autolayer(fit2_holts$mean, series = "Holt's Method") +
    autolayer(fit5_hc$mean, series = "SES Method") +
  ggtitle("Forecast from Holt's Method for Hardcover Books") +
  xlab("Days") +
  ylab("Books")
### Ch7.Q3)
data("eggs")

fit_eggs_1 <- holt(eggs, damped = T, h=100)
#summary(fit_eggs_1)
fit_eggs_2 <- holt(eggs, damped = T, exponential = T, h=100)
fit_eggs_3 <- holt(eggs, damped = F, exponential = T, h=100)
fit_eggs_4 <- holt(eggs, damped = T, exponential = T,
                   alpha = 0.2, beta = 0.2 ,h=100)
fit_eggs_5 <- holt(eggs, damped = T, exponential = T,
                   alpha = 0.4, beta = 0.4 ,h=100)
fit_eggs_6 <- holt(eggs, damped = F, exponential = F,
                   alpha = 0.8, beta = 0.2 ,h=100)


metrics <- rbind(accuracy(fit_eggs_1),accuracy(fit_eggs_2),
                 accuracy(fit_eggs_3),accuracy(fit_eggs_4),
                 accuracy(fit_eggs_5),accuracy(fit_eggs_6))

rownames(metrics) <- c("Damped Holt's","Damped & Exp Holt's", "Exp Holt's",
                       "Damped & Exp Holt's; alpah/beta = 0.2",
                       "Damped & Exp Holt's; alpah/beta = 0.4/0.4",
                       "alpah/beta = 0.8/0.2")
knitr::kable(round(metrics,3), caption = "Metrics for each model")

eggs %>%
    autoplot() +
    autolayer(fit_eggs_1$mean, series = "Damped Holt's") +
    autolayer(fit_eggs_2$mean, series = "Damped & Exp Holt's") +
      autolayer(fit_eggs_3$mean, series = "Exp Holt's") +
      autolayer(fit_eggs_4$mean,
                series = "Damped & Exp Holt's; alpah/beta = 0.2") +
      autolayer(fit_eggs_5$mean,
                series = "Damped & Exp Holt's; alpah/beta = 0.4/0.4") +
      autolayer(fit_eggs_6$mean, series = "alpah/beta = 0.8/0.2") +
    ggtitle("Forecasts from Holt's method") +
  guides(colour=guide_legend(title="Forecast")) +
  theme(legend.position="bottom")
```