Predict_413_Sec55_Homework_2

Singh, Gurjeet

February 18, 2018

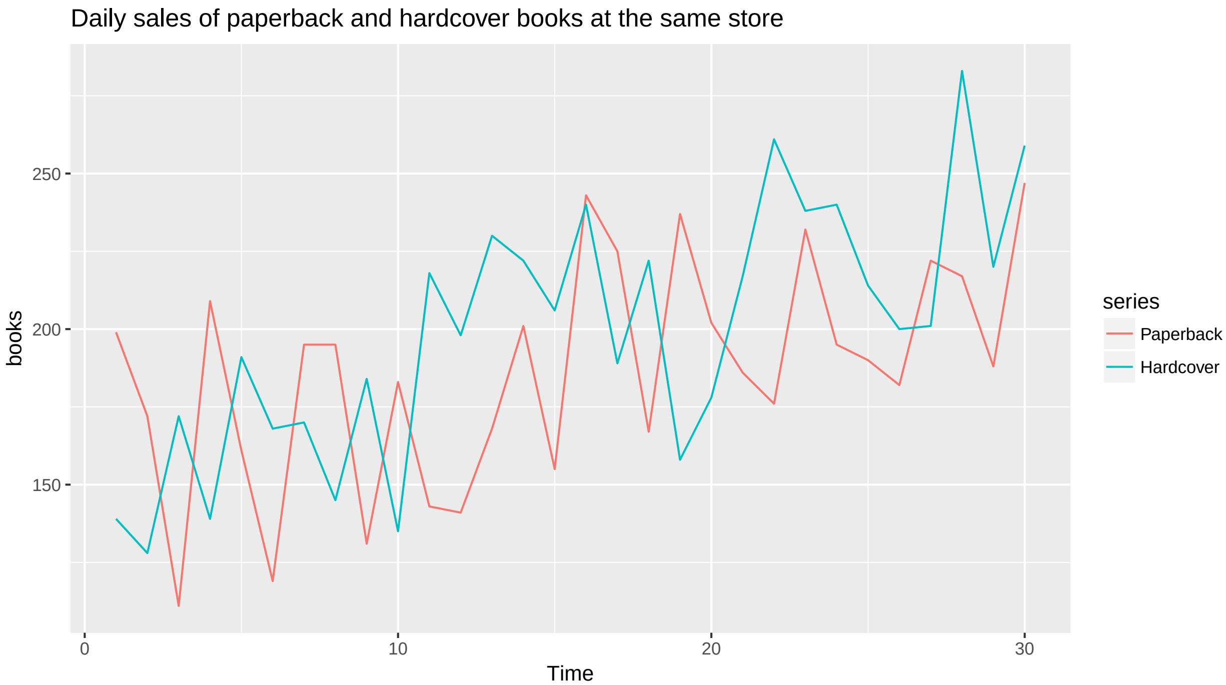
Contents

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.

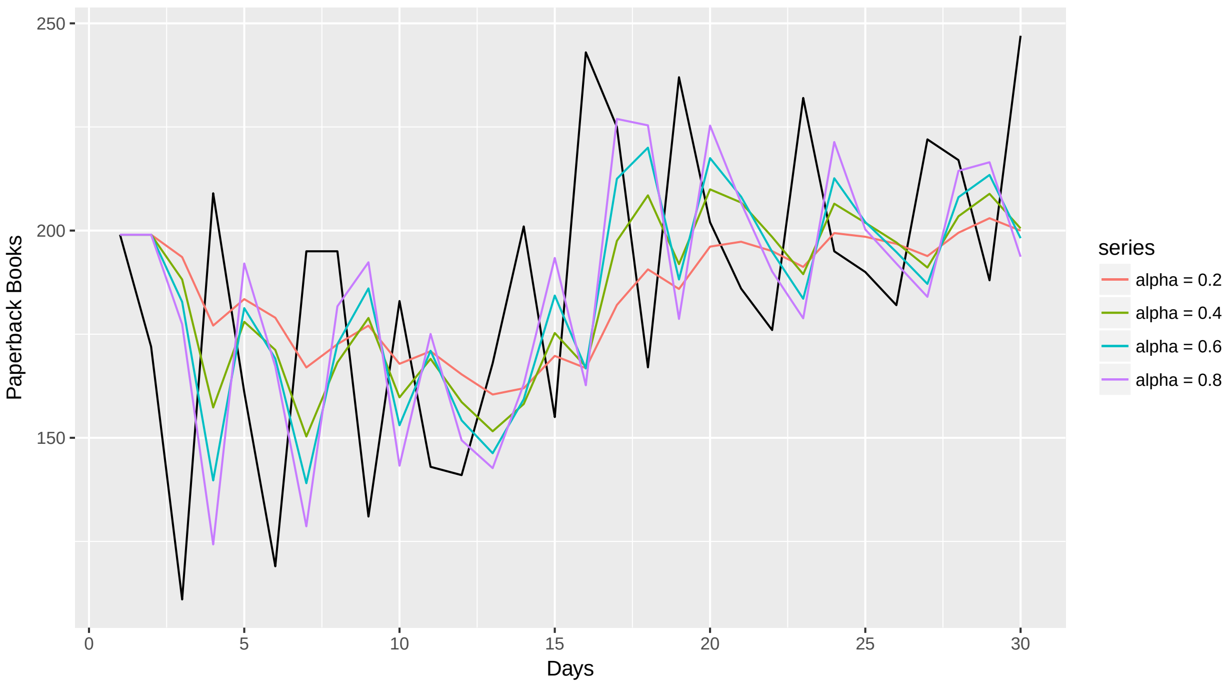
The figure below shows the four different sets of forecasts.

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
```

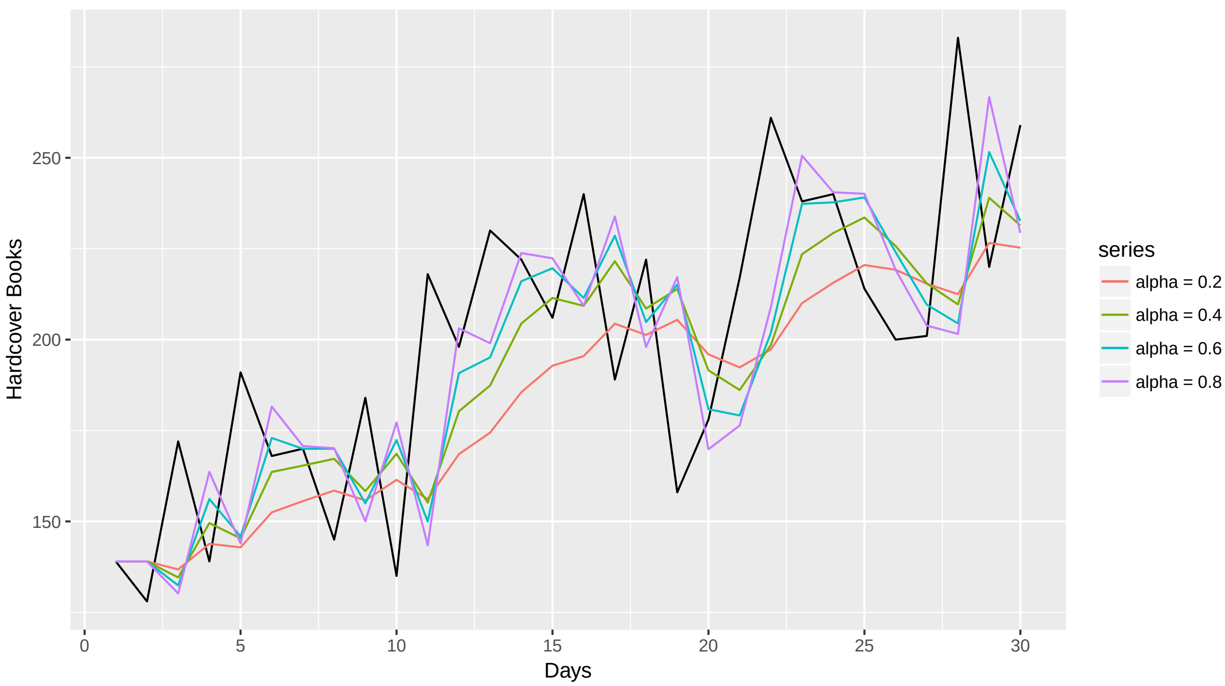
Table 5: Metrics from each model

	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)

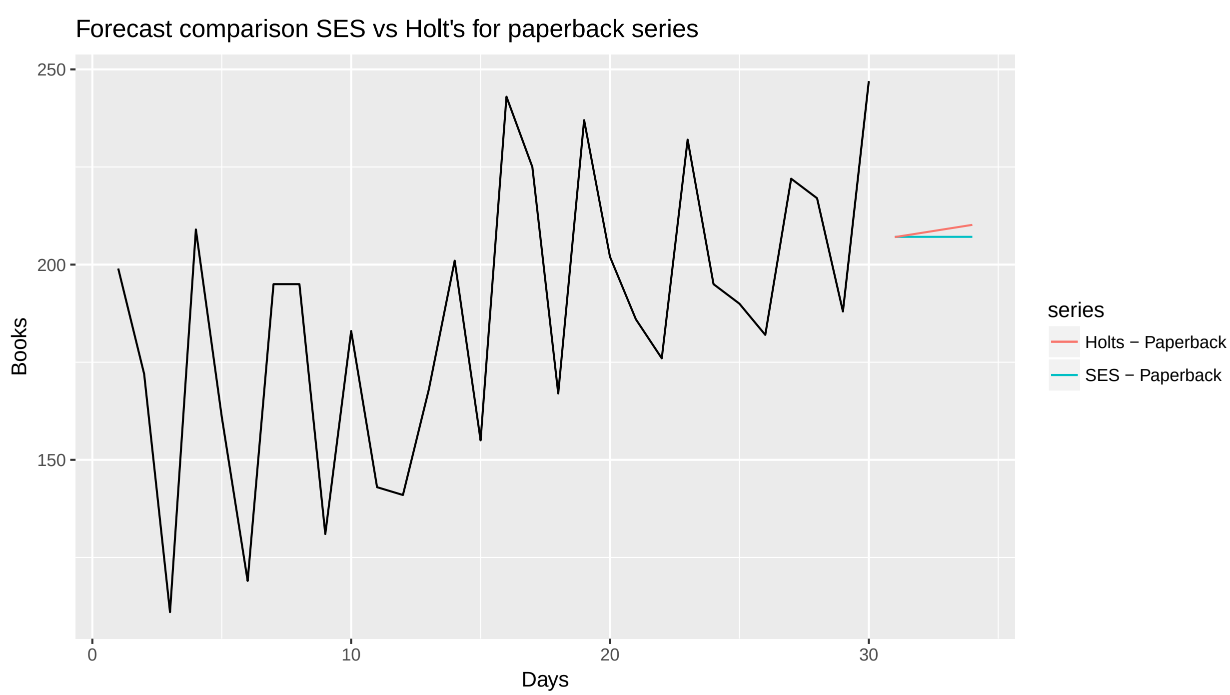
Table 9 shows the SSE value of paperback and hardback series obtained using Holt's linear method. These measures are much better than any of the SSE values obtained using SES. Earlier, we had doubts that data may have some trend behaviour. The result confirms that there's definitely trend behaviour present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

Table 9: SSE Values

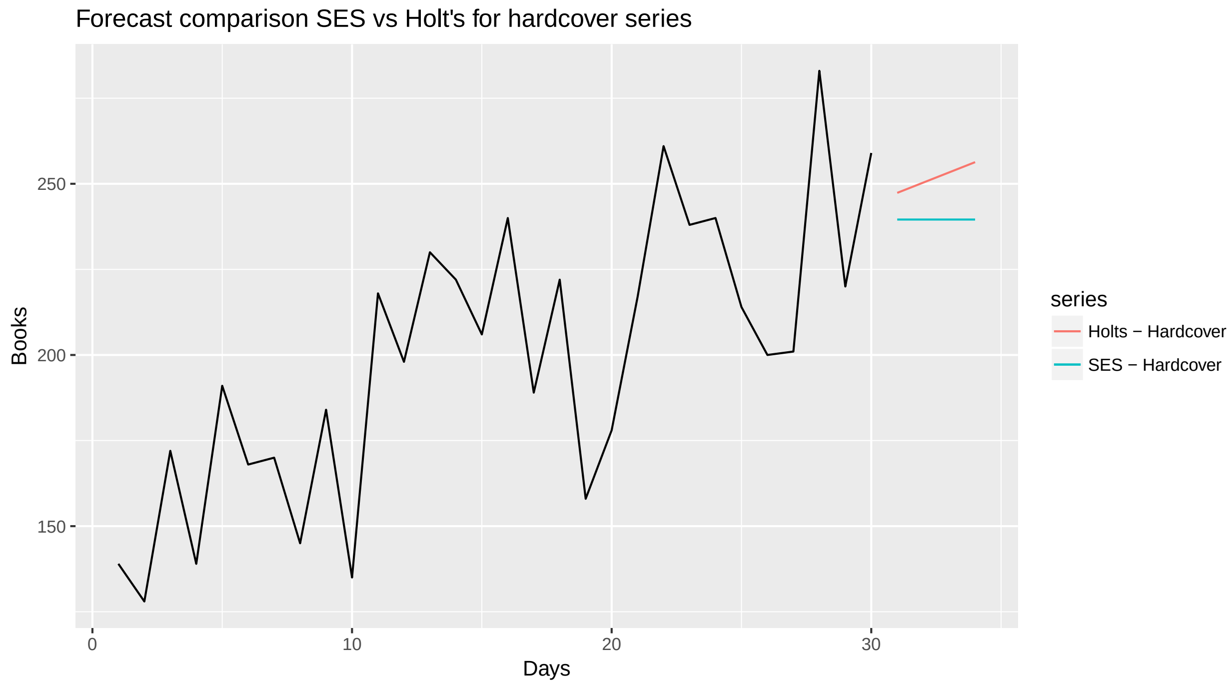
	Holt's Linear - Paperback	Holt's Linear - Hardcover
SSE Value	30074.17	22581.83

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.

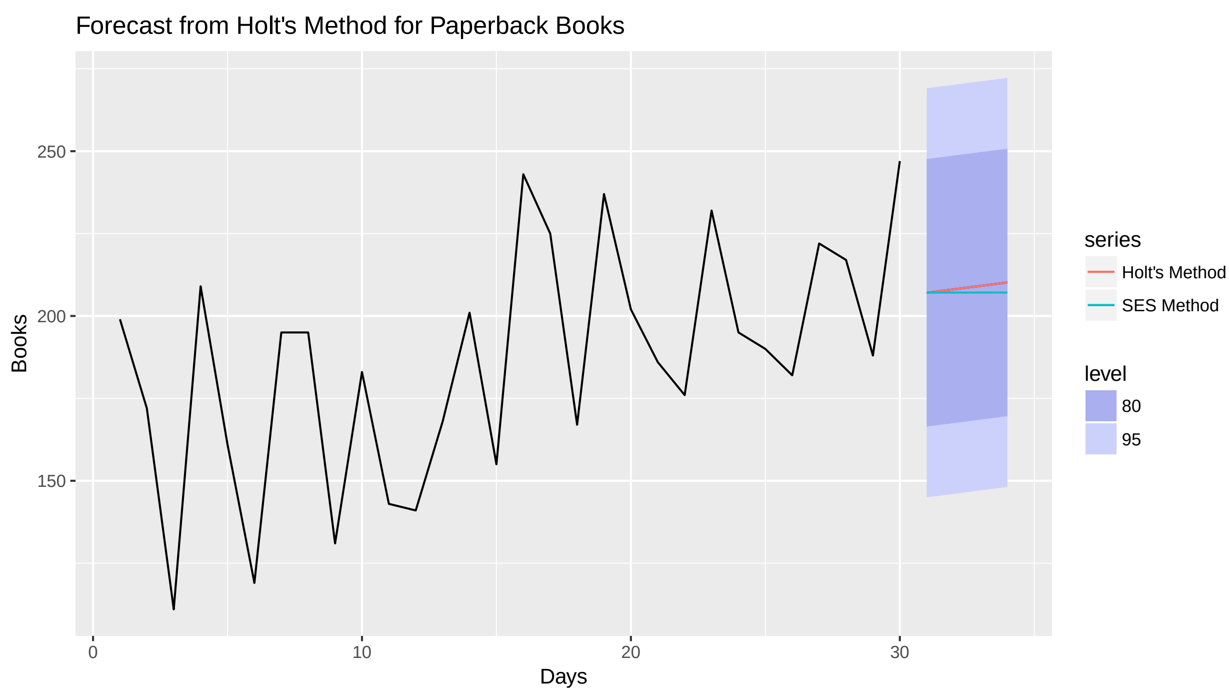


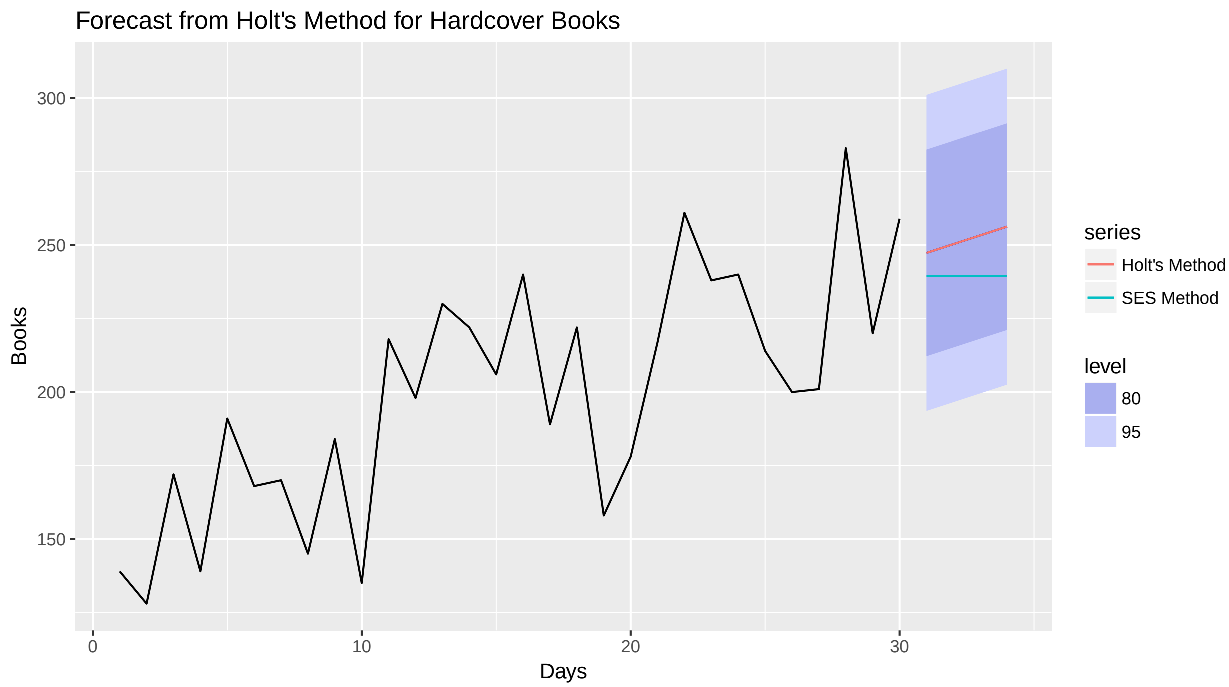
The figure below shows the forecast of Simple exponential smoothing and Holt's linear methods for hardcover books. Again, it appears that Holt's linear method performed better.



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.





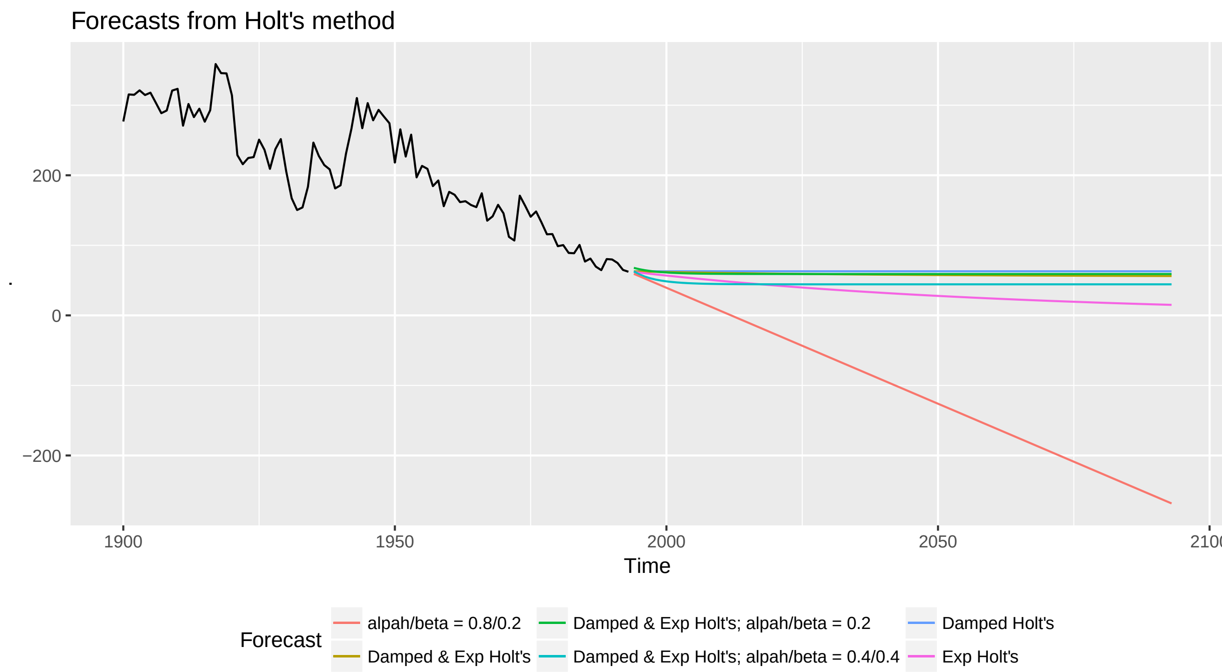
Question 3

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

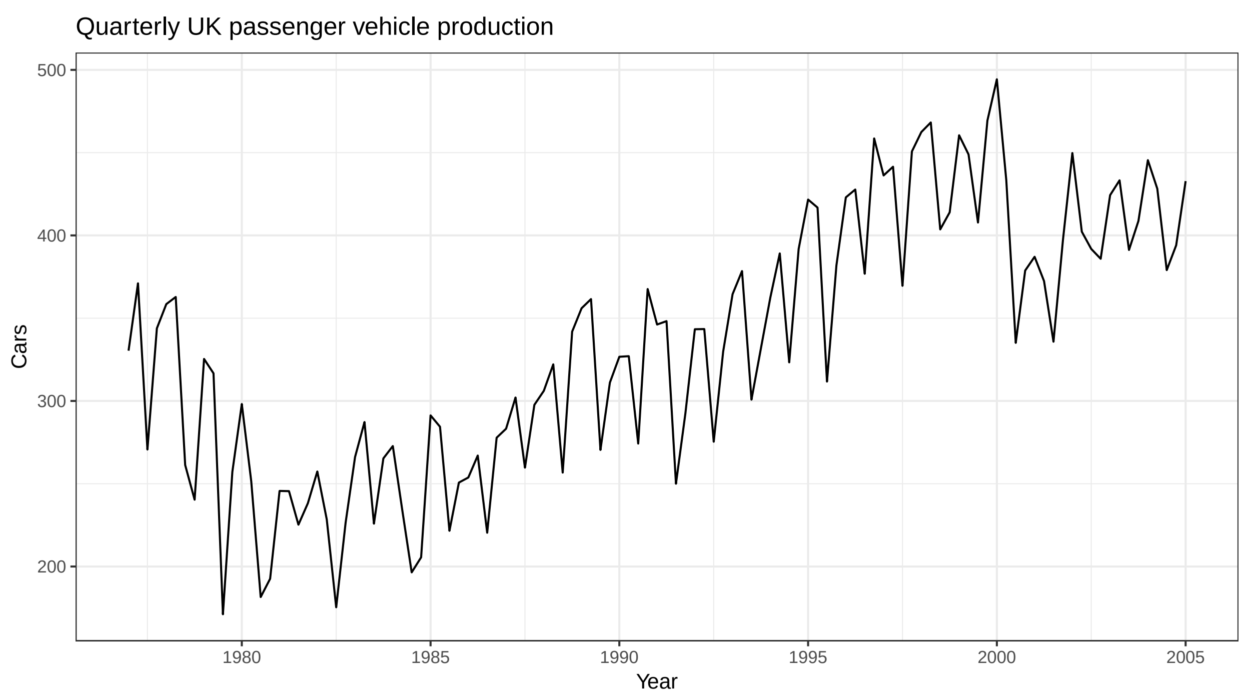
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Damped Holt's	-3.092	26.662	19.512	-3.023	10.110	0.962	-0.006
Damped & Exp Holt's	-0.882	26.526	19.514	-2.101	10.015	0.963	0.005
Exp Holt's	0.476	26.386	19.222	-1.280	9.754	0.948	0.007
Damped & Exp Holt's; alpah/beta = 0.2	-4.943	32.458	23.215	-4.007	12.019	1.145	0.477
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



Question 4

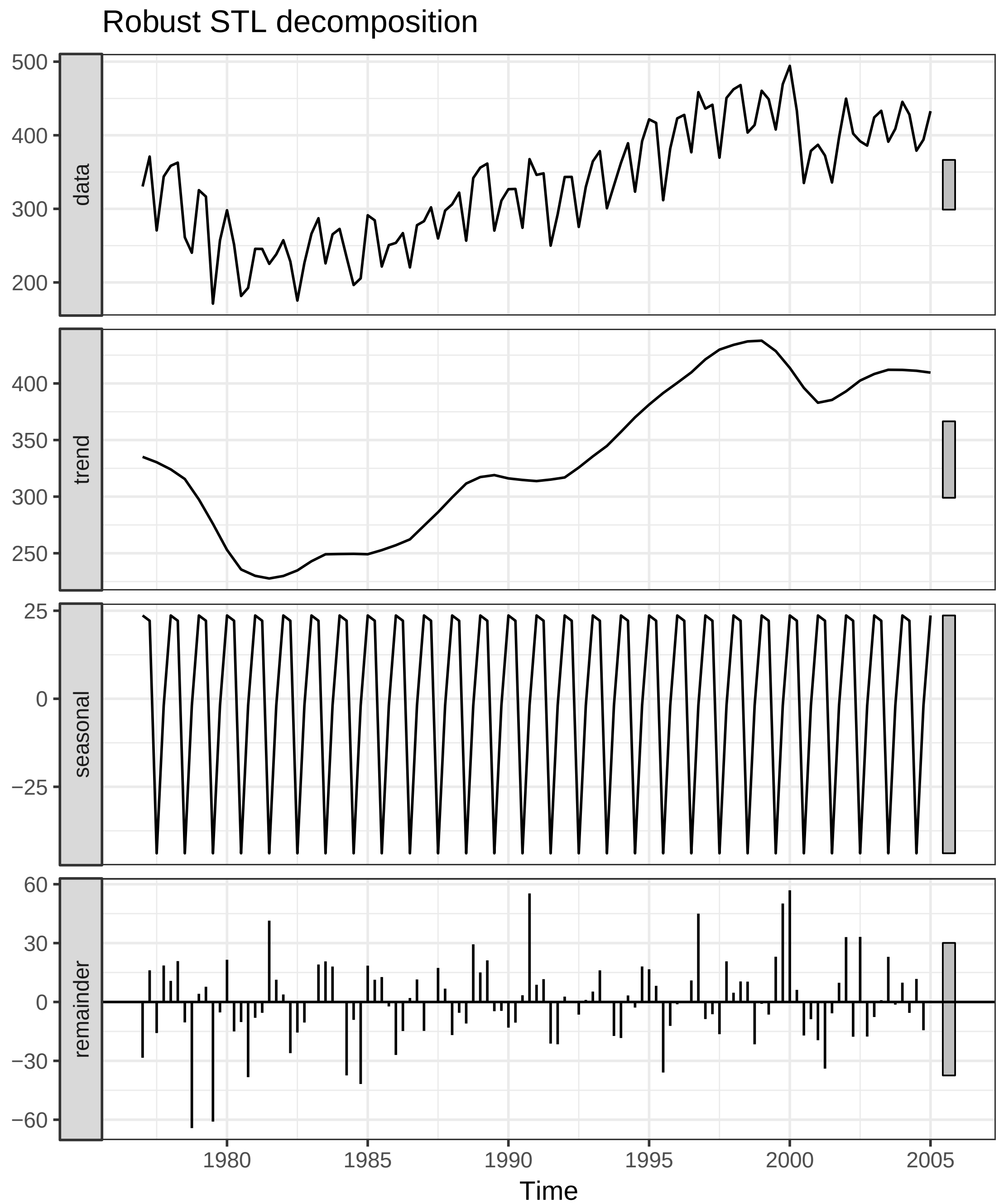
Ch7.Q4.a)

The figure below shows the plot of the quarterly UK passenger vehicle production from January 1977 to January 2005.The data in the figure below display seasonality as well as trend behaviour.

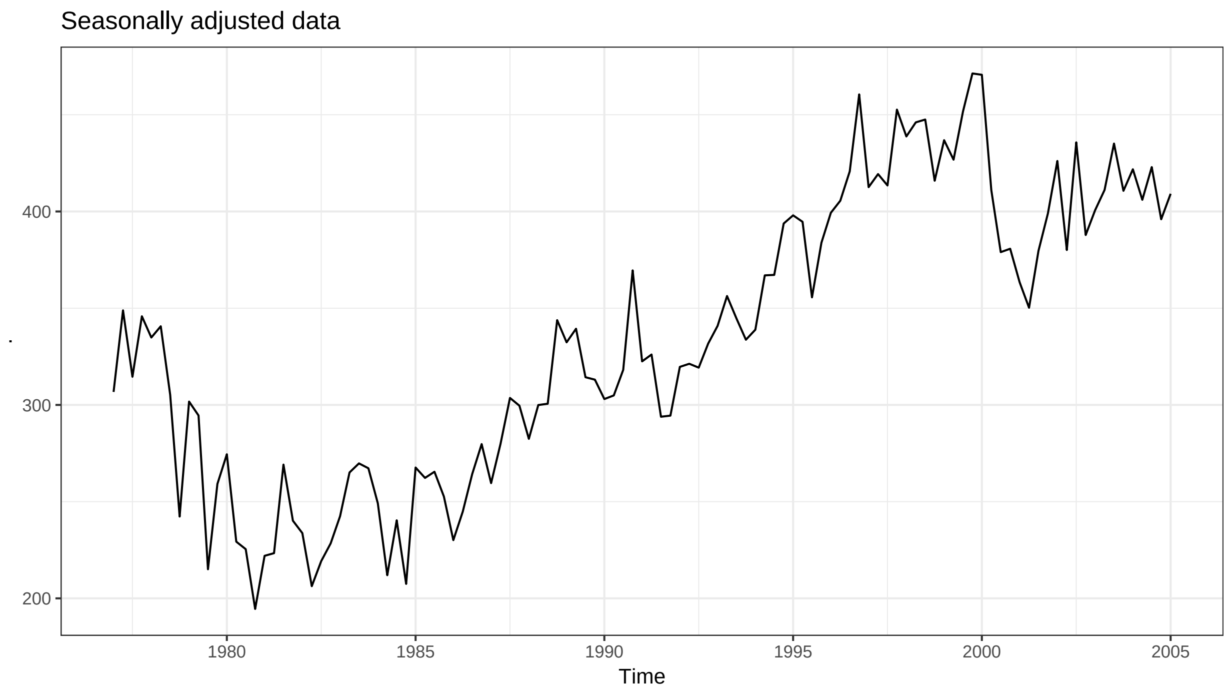


Ch7.Q4.b)

The figure below shows trend-cycle and seasonal indices of the STL decomposition. We can see the increasing trend from the beginning of 1980 to 1998. Earlier we mentioned that there are some seasonality in the data. Here, we can clearly see the seasonal effects in the third panel. The large gray bar in the third panel shows that variation in the seasonal component is small as compared to the variation in the data.



The figure below shows the plot of the seasonally adjusted data obtained using the robust STL decomposition.



Ch7.Q4.c)

The table below shows the metrics for an additive damped trend method applied to the seasonally adjusted data and reseasonalized forecast.

Table 11: Metrics

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Additive Damped	2.564590	25.18007	20.42583	0.3230029	6.530163	0.6656680	0.0364072
Reseasonalize Forecast	1.269787	25.27916	20.09403	-0.1870221	6.591482	0.6548548	0.0280683

Ch7.Q4.d)

The table below shows the metrics from the Holt's linear method applied to the seasonally adjusted data and reseasonalized forecast.

Table 12: Metrics

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Holt's linear	-0.2015904	25.36133	20.04100	-0.6076226	6.461718	0.6531263	0.0327424
Reseasonalize Forecast	1.2697874	25.27916	20.09403	-0.1870221	6.591482	0.6548548	0.0280683

Ch7.Q4.e)

The summary below shows the ETS(A,A,A) seasonal model with additive errors.

The table below shows the metrics for an additive damped trend method applied to the seasonally adjusted data and reseasonalized forecast.

```
## ETS(A,Ad,A)
##
## Call:
##  ets(y = ukcars, model = "AAA")
##
##   Smoothing parameters:
##     alpha = 0.5656
##     beta  = 1e-04
##     gamma = 1e-04
##     phi   = 0.9221
##
##   Initial states:
##     l = 344.5017
##     b = -7.4634
##     s=-0.64 -45.5093 20.8332 25.3161
##
##   sigma:  25.1687
##
##      AIC     AICc      BIC
## 1283.181 1285.338 1310.455
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE    MAPE      MASE
## Training set 2.347886 25.16871 20.52751 0.2258293 6.69942 0.6689815
##                    ACF1
## Training set 0.04225926
```

Ch7.Q4.f)

Based on the table below, it seems like ETS model had a better in-sample fit.

Table 13: Metrics

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Additive Damped	2.5645901	25.18007	20.42583	0.3230029	6.530163	0.6656680	0.0364072
Holt's linear	-0.2015904	25.36133	20.04100	-0.6076226	6.461718	0.6531263	0.0327424
ETS	2.3478864	25.16871	20.52751	0.2258293	6.699420	0.6689815	0.0422593

Ch7.Q4.e)

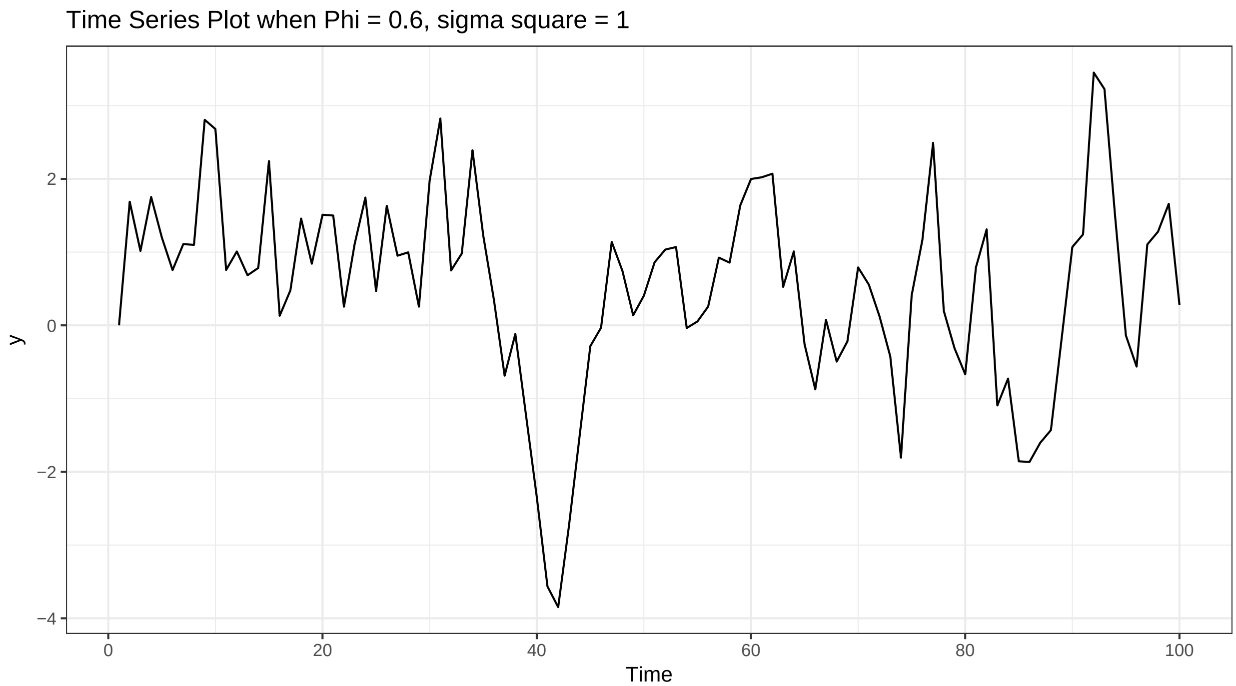
The forecasts generated Holt's linear method approach display a constant trend (increasing or decreasing) indefinitely into the future. Generally, this method over-forecast for longer forecast horizons. However, in our case, we are only forecasting for 4 additional days. Therefore, our forecast doesn't seem to over-forecast and is pretty close to the forecast of other methods. The forecast generated by additive damped trend seems to perform better over Holt's linear method. We got a better RMSE. However, in this cast, ETS method with additive seasonal component and additive error seems to perform the best.

Chapter 8

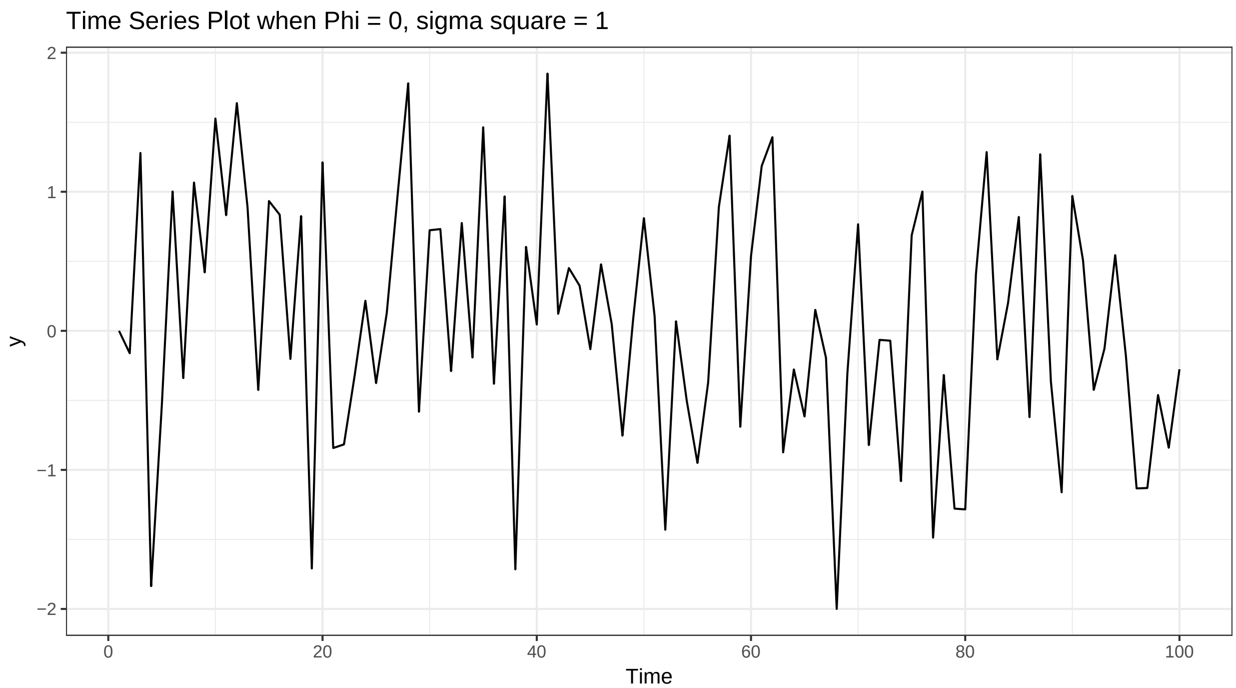
Question 5

Ch8.Q5.a&b)

THe plot below show the time series plot of the data generated from an AR(1) model with Phi = 0.6 and sigma squared = 1.



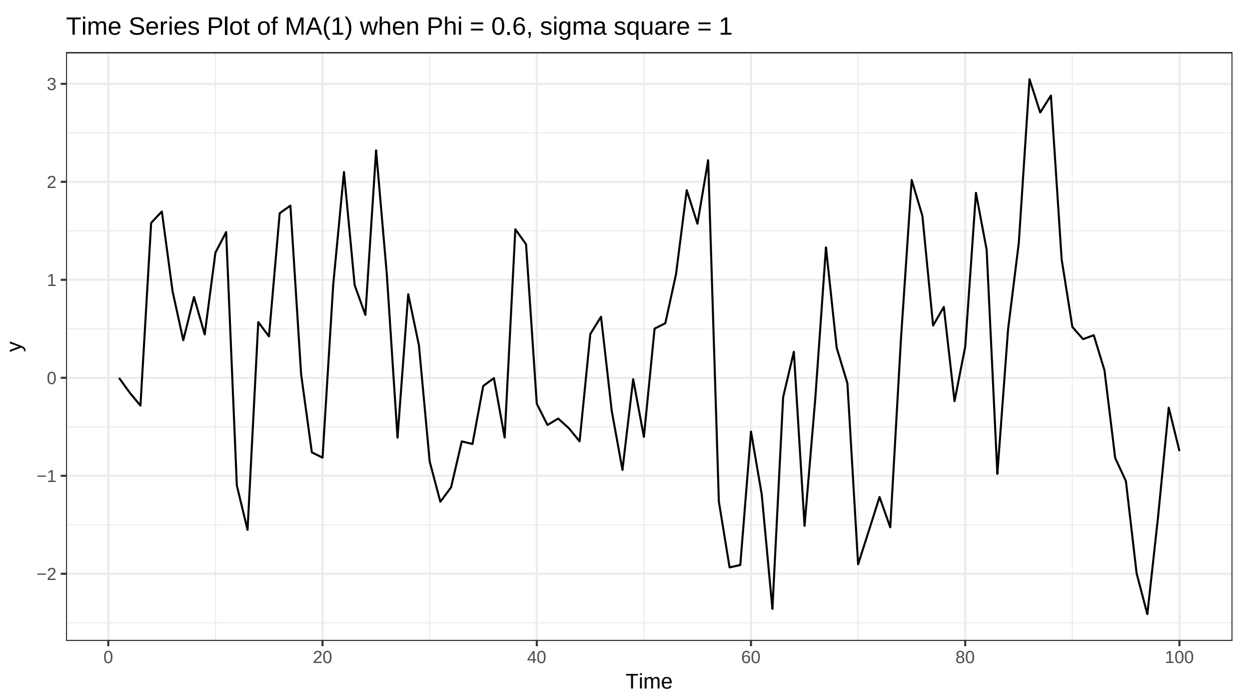
The plot below show the time series plot of the data generated from an AR(1) model with Phi = 0 and sigma squared = 1. By changing the Phi, it appears that the seasonal effect got more closer.



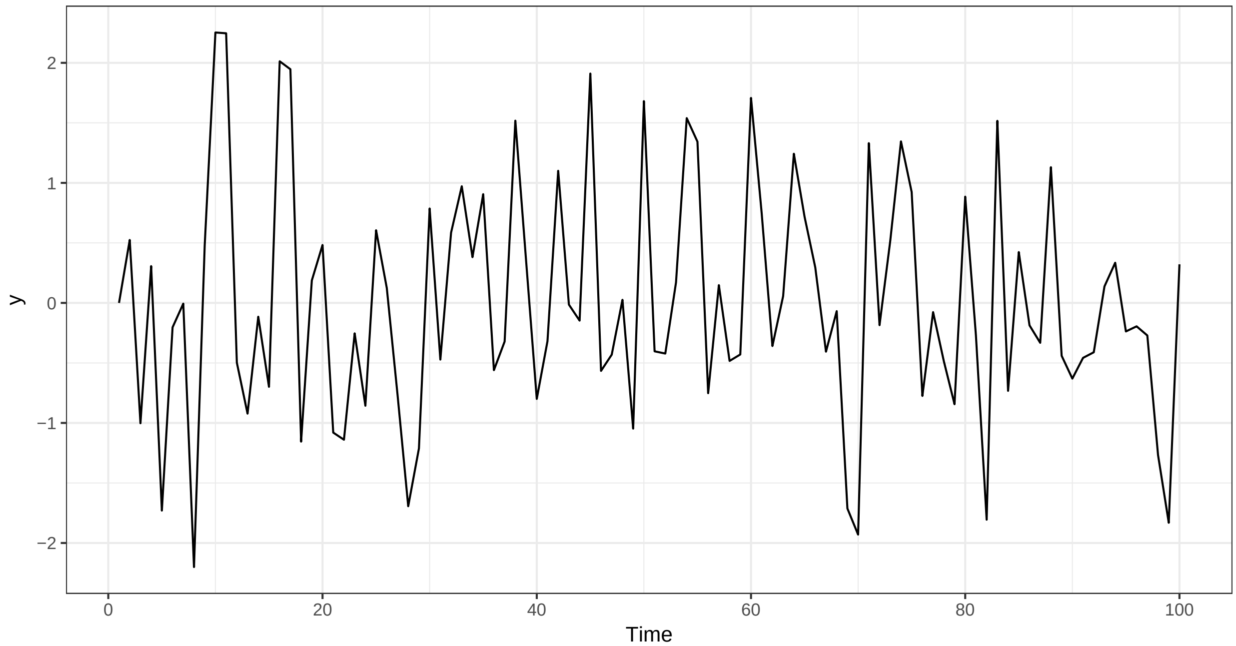
Ch8.Q5.c&d)

THe plot below show the time series plot of the data generated from an MA(1) model with Phi = 0.6 and sigma squared = 1.

The second plot below show the time series plot of the data generated from an MA(1) model with Phi = 0 and sigma squared = 1. By changing the Phi, it appears that the seasonal effect got more closer.



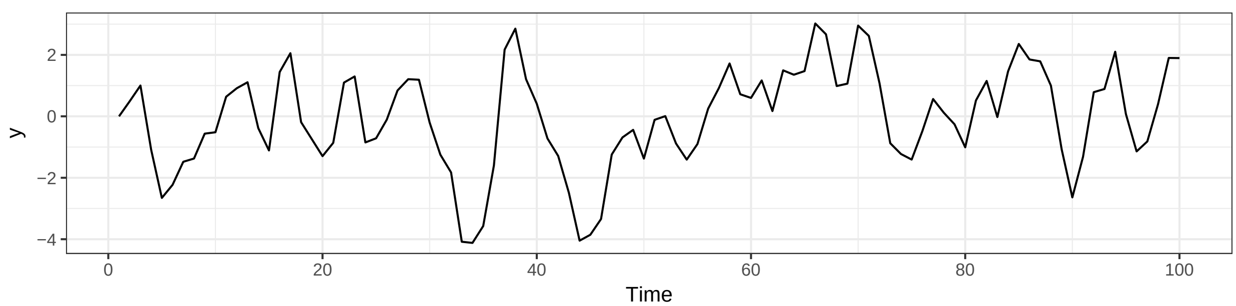
Time Series Plot of MA(1) when Phi = 0, sigma square = 1



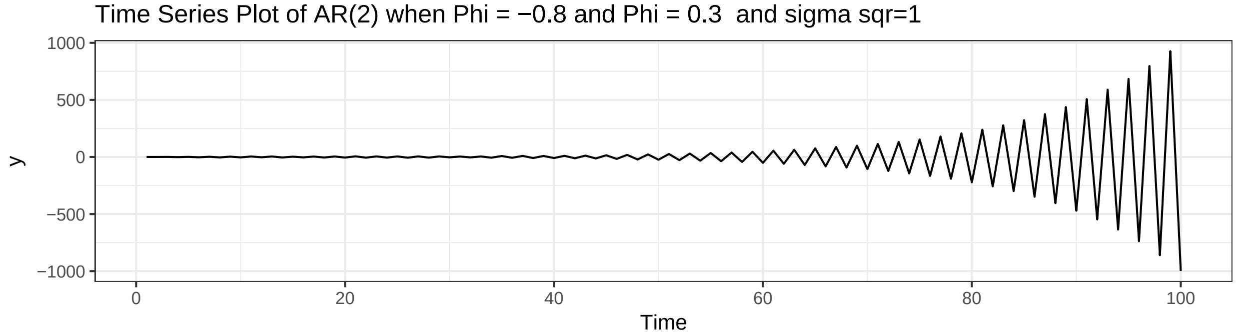
Ch8.Q5.e,f,&g)

The top plot below show the time series plot of the data generated from an ARMA(1,1) model with Phi = 0.6 and Theta = 0.6 and sigma sqr = 1. The second plot below show the time series plot of the data generated from an AR(2) model with Phi = -0.8 and Phi = 0.3 and sigma sqr = 1. It appears that the second plot is a non-stationary series while the first one seem to be a stationary series.

Time Series Plot of ARMA(1,1) when Phi = 0.6 and Theta=0.6 and sigma sqr=1



Time Series Plot of AR(2) when Phi = −0.8 and Phi = 0.3 and sigma sqr=1

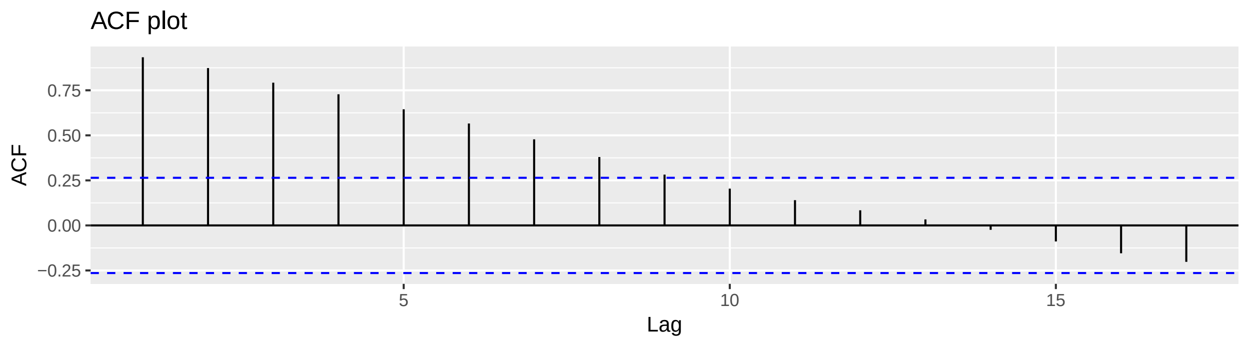


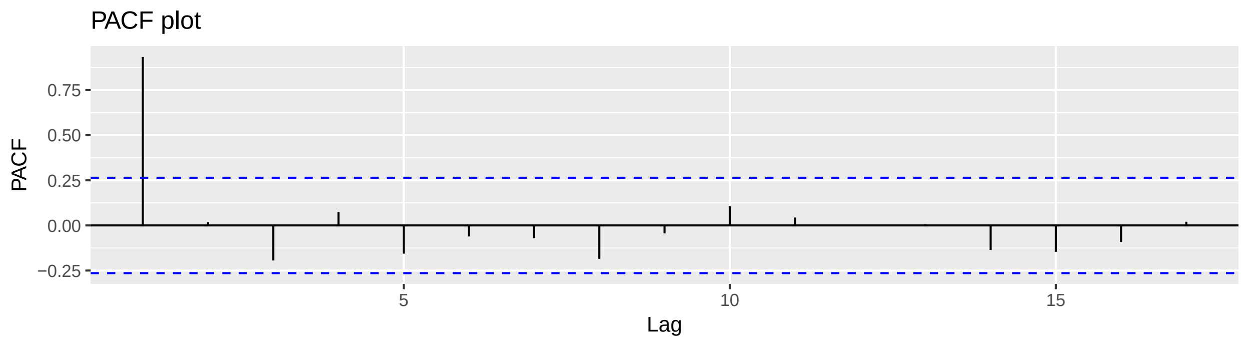
Question 6

Ch8.Q6.a)

The figure below shows the time series plot of the number of wonmen murdered each year in the US.

The figure below shows the ACF and PACF plots for the number of women murdered each year in the US. In the figure below, it appears that the data follow an ARIMA(p,d,0) model because the plot of the differenced data show the ACF is exponentially decaying and there is a significant spike at lag p in PACF, but none beyond lag p. Therefore, in ACF plot, there are nine (9) spikes decreasing with the lag and then no significant spikes thereafter. Hence, the pattern in the first nine spikes is what we would expect from an ARIMA(9,0,0) as the ACF tends to decay exponentially.





Ch8.Q6.b)

Based on the results from ACF and PACF plots, it doesn't seem to be a need for constant in the model because the long term forecasts will not go to the mean of the data, follow a straight line, or follow a quadratic trend.

Ch8.Q6.c)

ARIMA(p,d,q) model is given by:

((1 - B)^d)*Yt = mu + (theta(B)/phi(B))at

Where,
t - indexes time
mu - is the mean term
B - is the backshift operator; that is, BXt=Xt-1
phi(B) - is the autoregressive operator
theta(B) - is the moving-average operator
at - is the independent disturbance, also called the random error

The ARIMA(9,0,0) model in terms of the backshift operator is shown below:

((1 - B)^d)*Yt = mu + ((1 - theta0(B))/(1 - phi1(B) - … - phi9(B)^9))at

Ch8.Q6.d)

We create three models, auto.arima, ARIMA(9,0,0) - our model, and ARIMA(9,1,0). Two of the models created is to compare with our model. Below is the summary from each model. When comparing the RMSE for each model, it seems like our ARIMA(9,0,0) model performed the best.

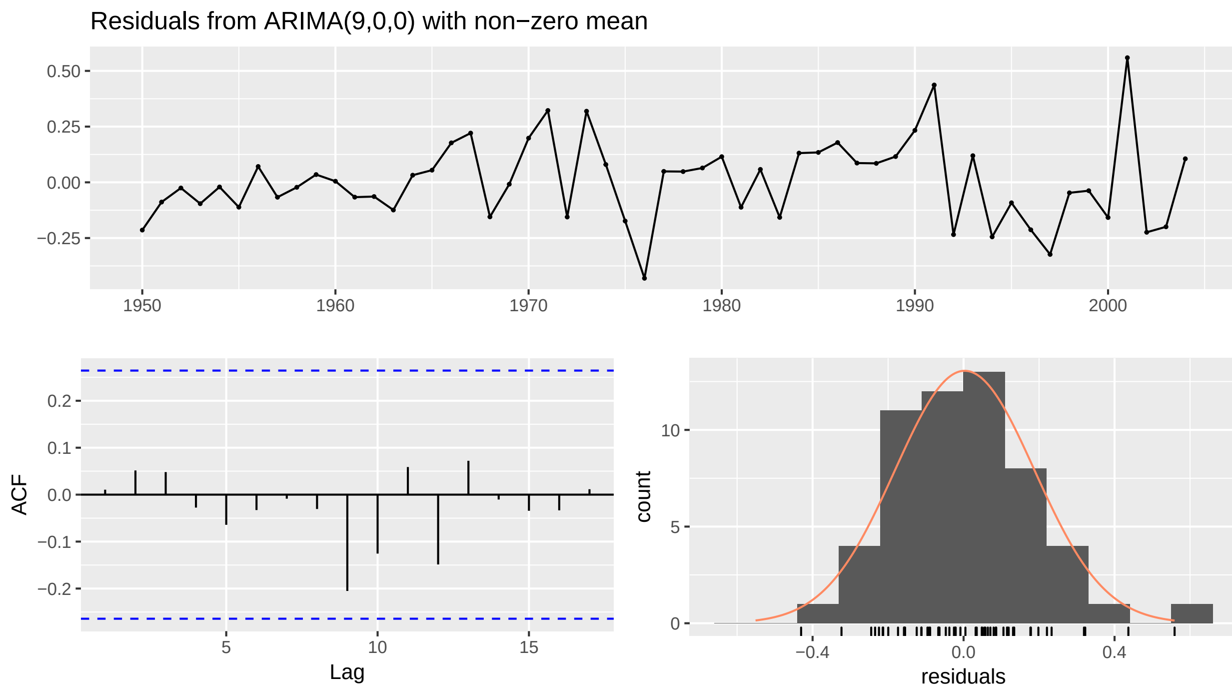
```
## Series: wmurders
## ARIMA(1,2,1)
##
## Coefficients:
##          ar1      ma1
##      -0.2434  -0.8261
## s.e.  0.1553   0.1143
##
## sigma^2 estimated as 0.04632:  log likelihood=6.44
## AIC=-6.88   AICc=-6.39   BIC=-0.97
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set -0.01065956 0.2072523 0.1528734 -0.2149476 4.335214 0.9400996
##                    ACF1
## Training set 0.02176343
```

```
## Series: wmurders
## ARIMA(9,0,0) with non-zero mean
##
## Coefficients:
##         ar1     ar2      ar3      ar4     ar5     ar6     ar7      ar8
##      0.7990  0.4250  -0.2329  -0.1273  0.0136  0.2903  0.0263  -0.1817
## s.e.  0.1332  0.1736   0.1803   0.2075  0.2128  0.2130  0.2087   0.1982
##          ar9    mean
##      -0.1073  3.2967
## s.e.  0.1498  0.2839
##
## sigma^2 estimated as 0.04096:  log likelihood=13.3
## AIC=-4.6   AICc=1.54   BIC=17.48
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.003053233 0.1830756 0.1436313 -0.2545438 4.174703 0.8832652
##                    ACF1
## Training set 0.01058207
```

```
## Series: wmurders
## ARIMA(9,1,0)
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ar5     ar6     ar7     ar8
##      -0.1013  0.3494  0.0645  -0.0978  -0.0905  0.2117  0.2903  0.0013
## s.e.   0.1310  0.1334  0.1401   0.1497   0.1482  0.1510  0.1506  0.1455
##          ar9
##      -0.2284
## s.e.   0.1463
##
## sigma^2 estimated as 0.04227:  log likelihood=13.08
## AIC=-6.17   AICc=-1.05   BIC=13.72
##
## Training set error measures:
##                         ME      RMSE       MAE         MPE     MAPE
```

```
## Training set 0.0006014757 0.1859611 0.1378977 -0.01577436 3.991845
##                     MASE        ACF1
## Training set 0.8480061 -0.04622825
```

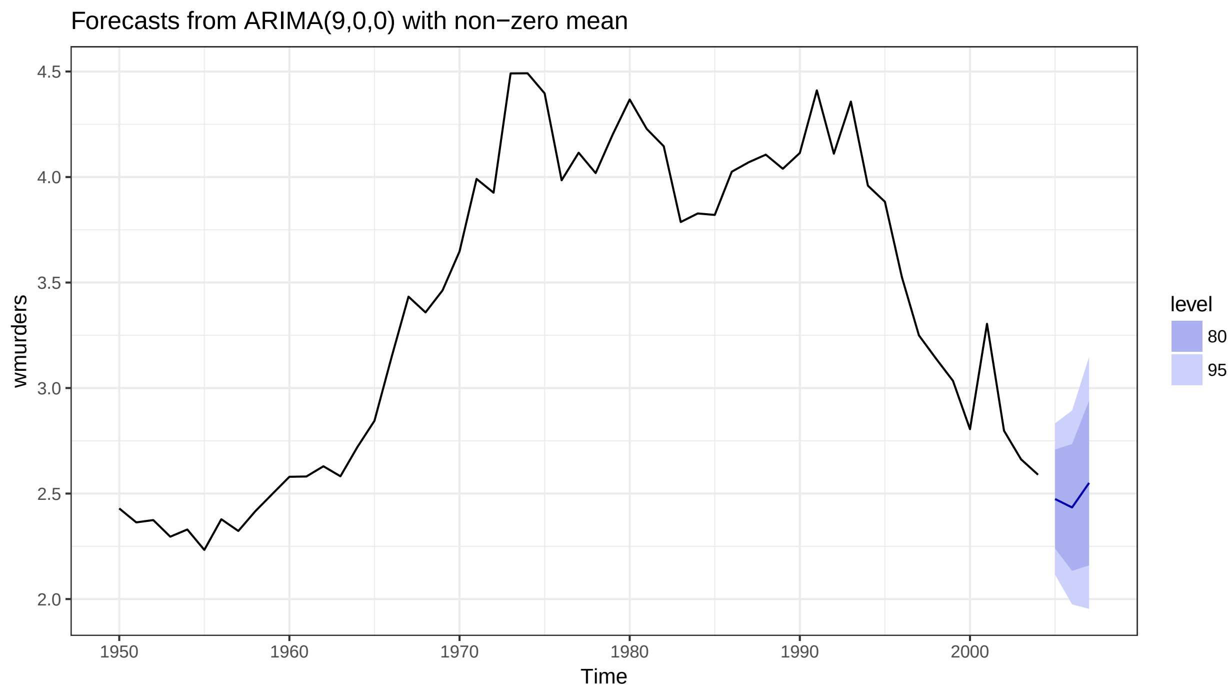
The figure below shows the residuals from ARIMA(9,0,0) model with non-zero mean. In the ACF plot, all the spikes are now within the significance limits, and so the residual apprea to be white noise. Therefore, the model is satisfactory.




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(9,0,0) with non-zero mean
## Q* = 6.9555, df = 3, p-value = 0.07333
##
## Model df: 10.   Total lags used: 13
```

Ch8.Q6.e&f)

The figure below shows the forecast for the next three years with the 80% and 95% prediction interval.



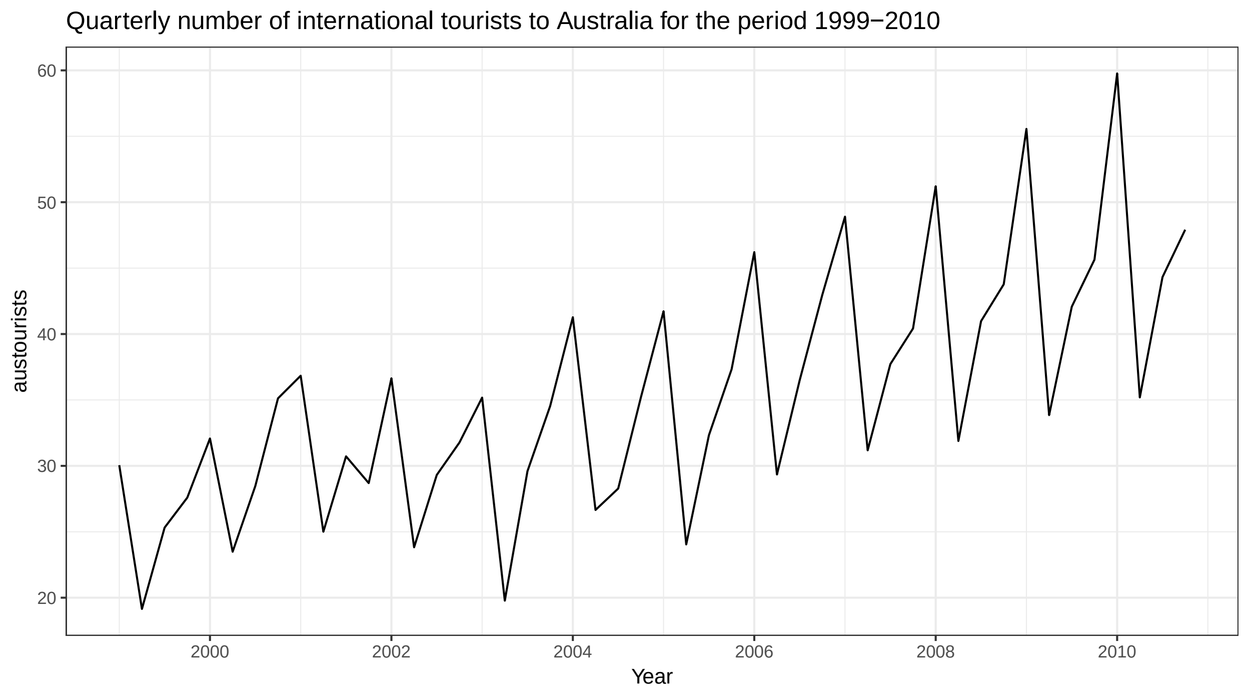
Ch8.Q6.g)

In section Ch8.Q6.d), I have compared auto.arima with our model. The auto.arima do not give us the same model. The model produced by auto.arima is ARIMA(1,2,1) which did not perform better than our model. The RMSE is little higher than our model. Therefore, ARIMA(9,0,0) model is the better model.

Question 7

Ch8.Q7.a)

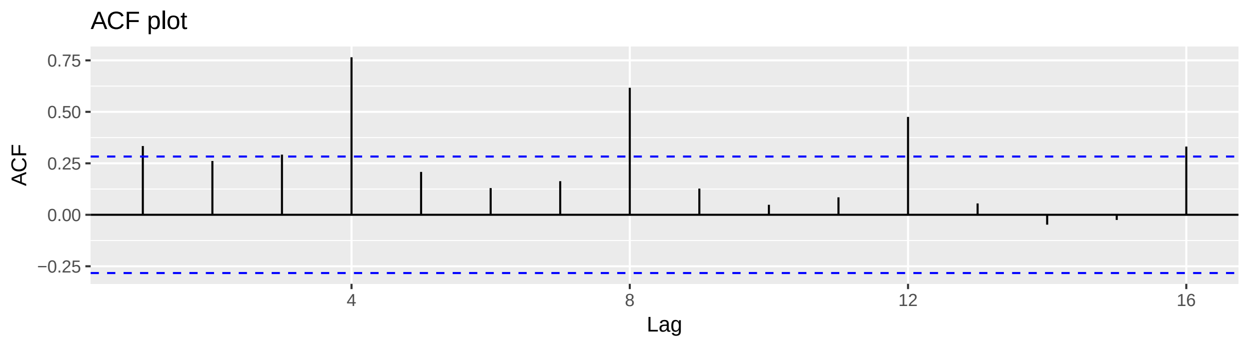
The figure below shows the quarterly number of international tourists to Australia for the period 1999-2010. The data follows the seasonal patterns and have the increasing trend.

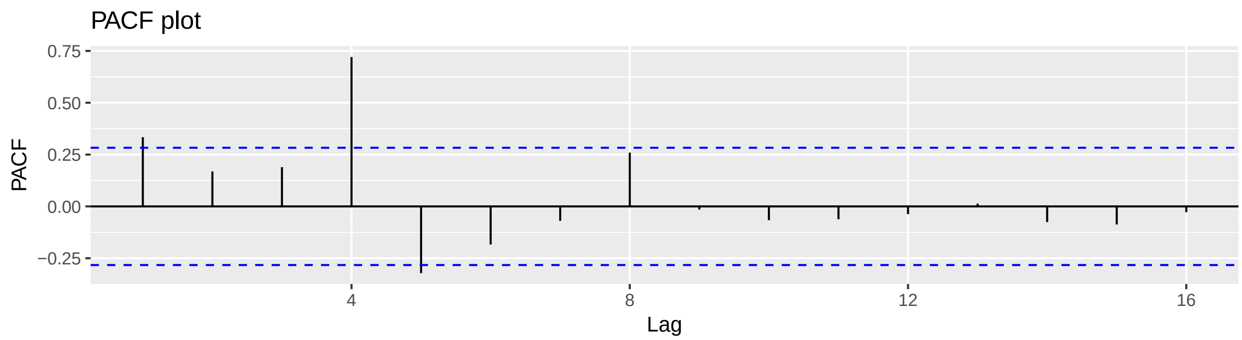


Ch8.Q7.b&c)

The figure below shows the ACF and PACF plots for the quarterly number of international tourists to Australia for the period 1999-2010. In the figure below, ACF plot shows that the lag is every fourth term because there are spikes at lag 4, lag 8, lag 12, and lag 16 and is exponentially decaying in the deasonal lags of the ACF. Hence, the data will follow an ARIMA(0,0,0)(1,0,0)4 model.

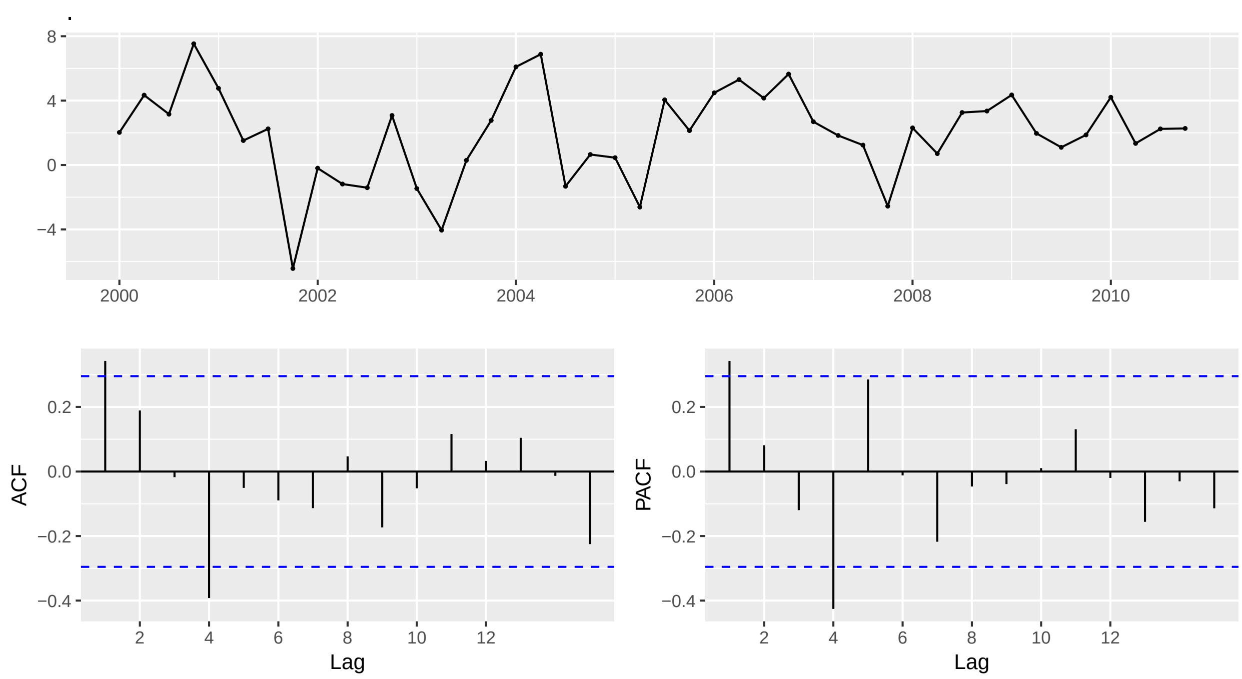
data follow an ARIMA(p,d,0) model because the plot of the differenced data show the ACF is exponentially decaying and there is a significant spike at lag p in PACF, but none beyond lag p. Therefore, in ACF plot, there are nine (9) spikes decreasing with the lag and then no significant spikes thereafter. Hence, the pattern in the first nine spikes is what we would expect from an ARIMA(9,0,0) as the ACF tends to decay exponentially.





Ch8.Q7.d)

The figure below shows the seasonally differenced data along with ACF and PACF plots. There is a significant spike at lag 1 in the ACF suggests a non-seasonal MA(1) component, and the significant spike at lag 4 in the ACF suggests a seasonal MA(1) component. Therefore, we begin with an ARIMA(0,1,1)(0,1,1)[4] model.



Ch8.Q7.e)

The summary below shows the summaries of auto.arima and ARIMA(0,1,1)(0,1,1)[4]. The auto.arima gave us the model ARIMA(1,0,0)(1,1,0)[4] with drift. The auto.arima model is the better model for this data because AIC and RMSE are lower than the ARIMA(0,1,1)(0,1,1)[4] model.

```
## Series: austourists
## ARIMA(1,0,0)(1,1,0)[4] with drift
##
## Coefficients:
##          ar1     sar1   drift
##       0.4493  -0.5012  0.4665
## s.e.  0.1368   0.1293  0.1055
##
## sigma^2 estimated as 5.606:  log likelihood=-99.47
## AIC=206.95   AICc=207.97   BIC=214.09
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.03377709 2.188233 1.632832 -0.6731192 5.000182 0.5633341
##                     ACF1
## Training set -0.0525015
```

```
## Series: austourists
## ARIMA(0,1,1)(0,1,1)[4]
##
## Coefficients:
##           ma1     sma1
##       -0.5746  -0.4990
## s.e.   0.1726   0.1164
##
## sigma^2 estimated as 6.617:  log likelihood=-101.45
## AIC=208.89   AICc=209.51   BIC=214.17
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.0535148 2.377357 1.776751 -0.9062089 5.410523 0.6129867
##                     ACF1
## Training set 0.03165183
```

Ch8.Q7.f)

Since we are using auto.arima as our final model, we will write the model in terms of the backshift operator as follows:

Seasonal ARIMA models are expressed in factored form by the notation ARIMA(p,d,q)(P,D,Q)s, where

P - is the order of the seasonal autoregressive part
D - is the order of the seasonal differencing (rarely should D > 1 be needed)
Q - is the order of the seasonal moving-average process
s - is the length of the seasonal cycle

Given a dependent time series [Yt : 1 <= t <= n] ,mathematically the ARIMA seasonal model is written as (1-B)$^{d(1\text{-}B}$s)^D Yt = mu + ((theta(B) thetas(B^s))/(phi(B) phis(B^s)))at

where,
phis(B^s) - is the seasonal autoregressive operator
thetas(B^s) - is the seasonal moving-average operator

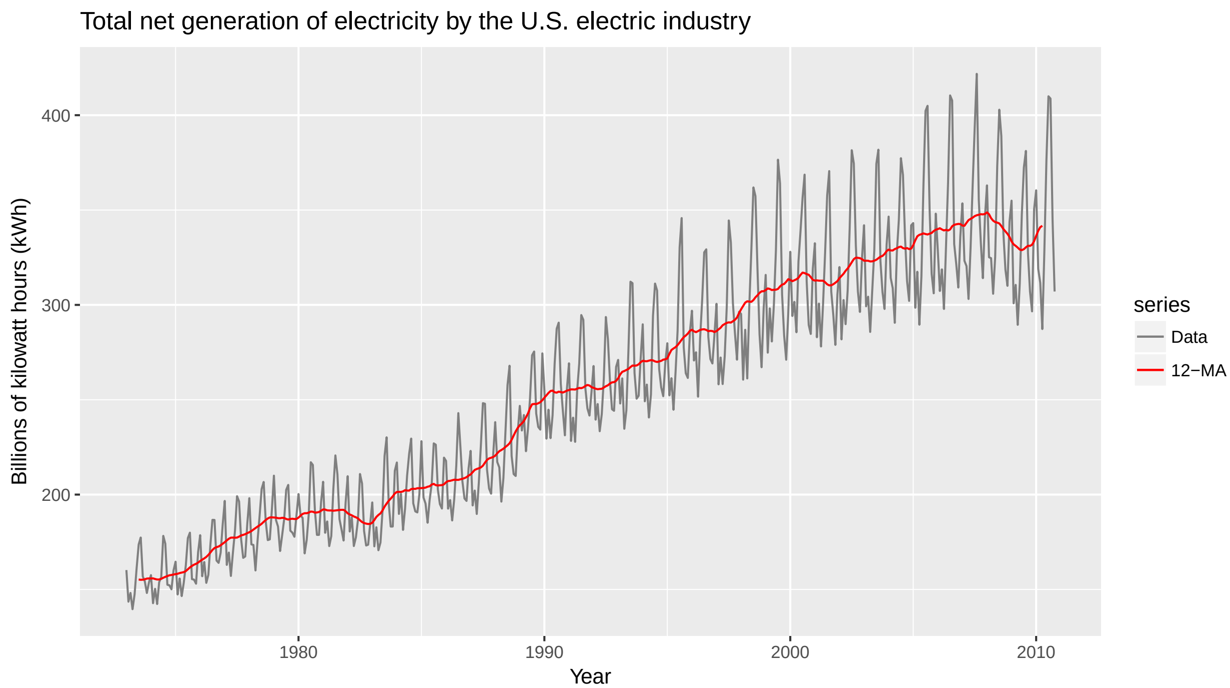
Therefore, our model ARIMA(1,0,0)(1,1,0)[4] can we written as:

(1-B^4)Yt = mu + ((1 - theta1(B))(1-thetas1(B^4) - thetas2 0^{B}8))/(1 - phi(B) (1 - phis $1^{(B}$4)))at

Question 8

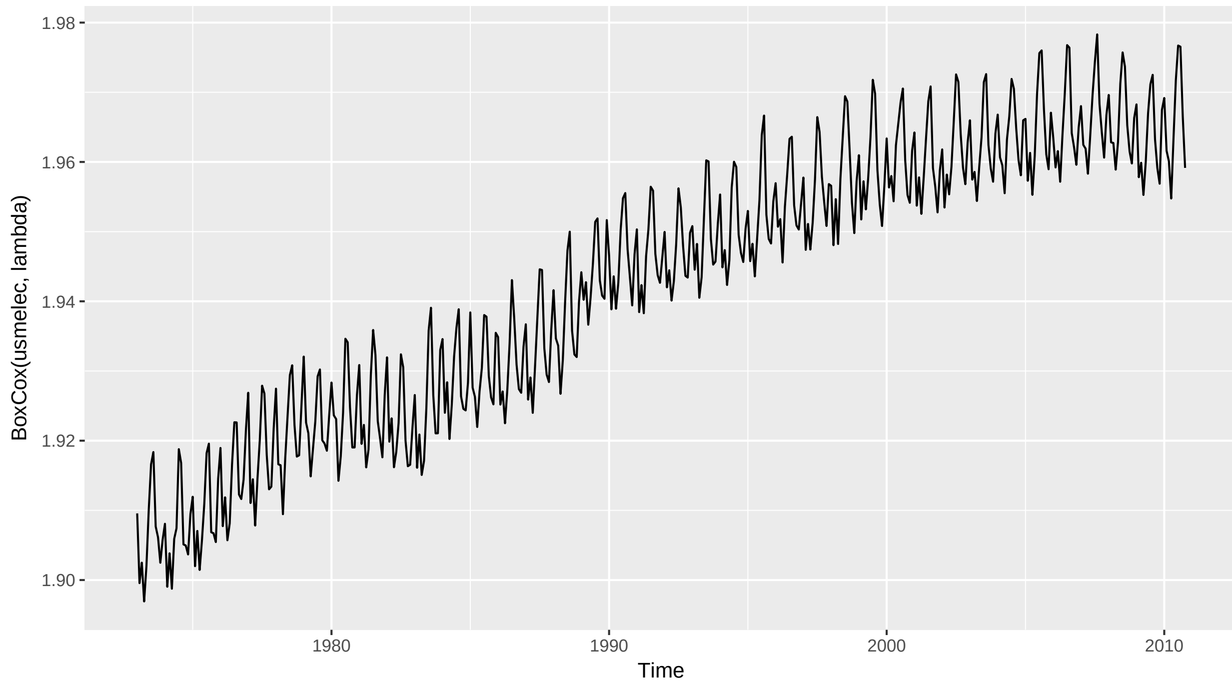
Ch8.Q8.a)

In the figure below, we see the increasing trend which means there is an increase in the electricity generation over time. The original data also show the strong seasonal component.



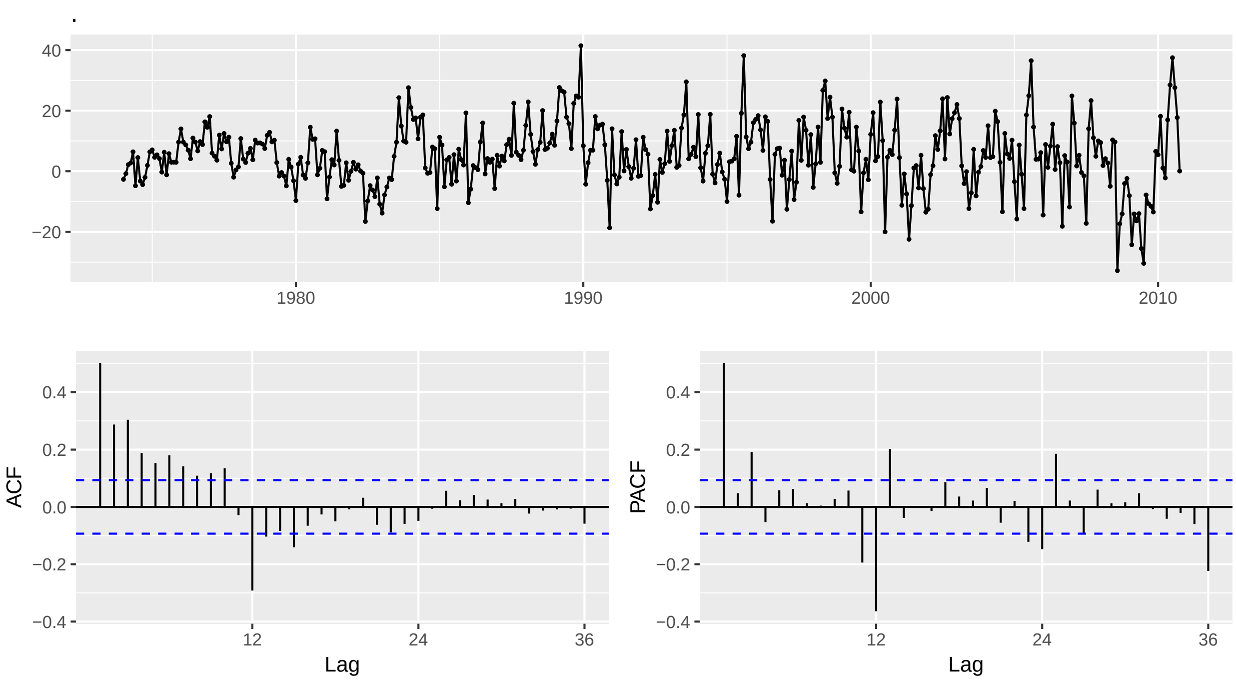
Ch8.Q8.b)

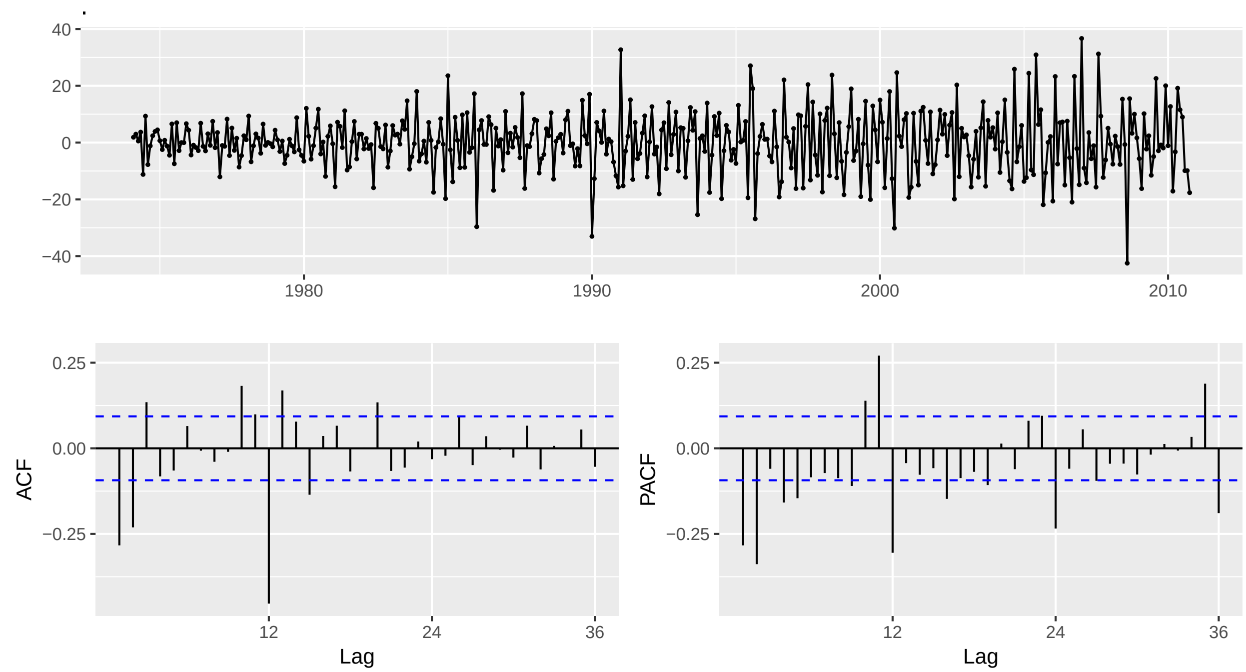
In the previous section, we noticed variation in the data. Therefore, transformation will be useful. We will use the Box-Cox transformation.



Ch8.Q8.c)

The data does not appear to be stationary. The first figure below shows the seasonal difference. This figure shows that the data still appear to be non-stationary. Therefore, we take an additional first difference, shown in the next figure. It seems to me that the data now appears to be stationary. However, there are significant spikes at various lags.





Ch8.Q8.d)

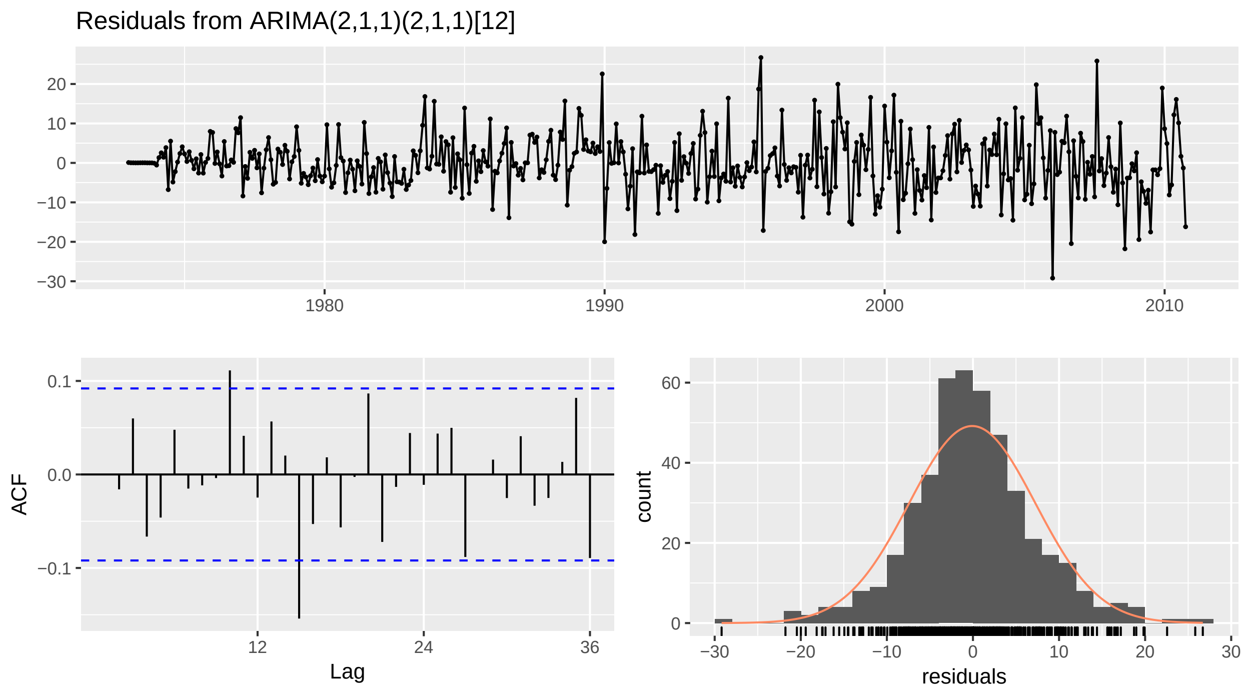
We created four ARIMA models (including one auto.arima model). According to the AIC values, our best model is ARIMA(2,1,1)(2,1,1).

Table 14: AIC Values

	Arima(1,0,1)(1,1,1)	Arima(2,1,1)(2,1,1)	Arima(2,2,2)(2,2,1)	Arima(1,0,2)(0,1,1) with drift
AIC	3087.396	3047.929	3050.492	3050.492

Ch8.Q8.e)

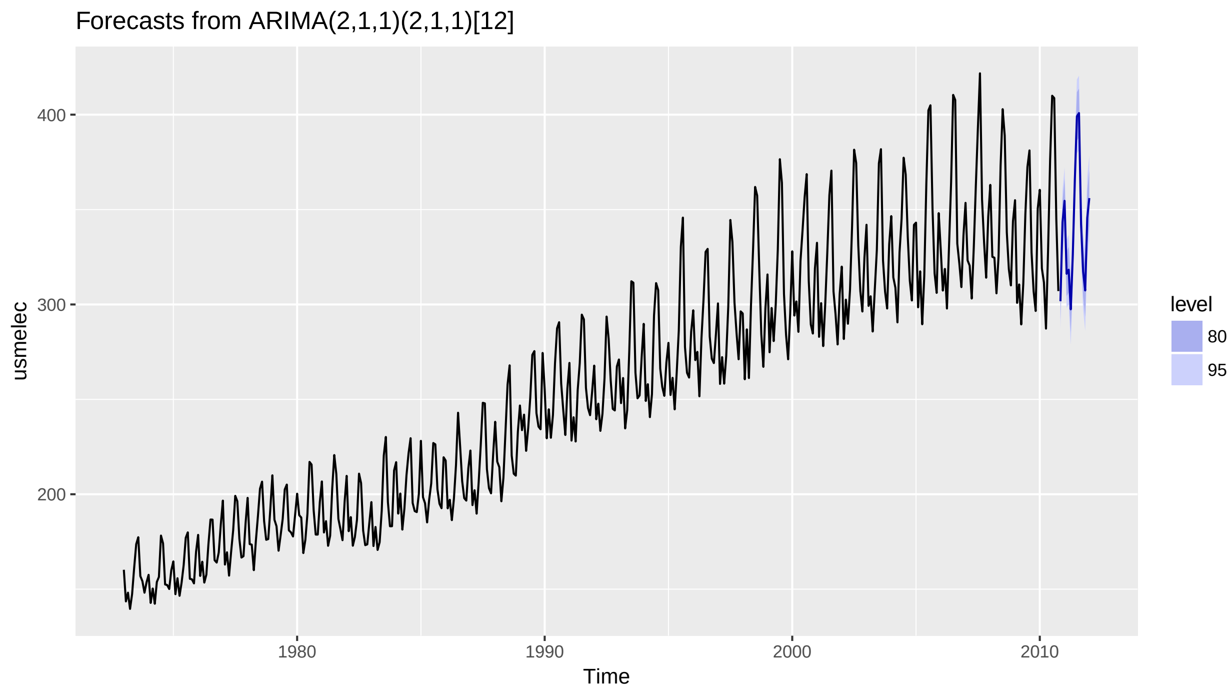
The figure below shows the ARIMA model selected and estimated. The ARIMA model captures all the dynamics in the data as the residual seems to be white noise. The residuals are also distributed normally.




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)(2,1,1)[12]
## Q* = 35.918, df = 18, p-value = 0.007229
##
## Model df: 6.   Total lags used: 24
```

Ch8.Q8.f)

```
##                       ME     RMSE      MAE         MPE     MAPE      MASE
## Training set -0.0812833 7.345784 5.434283 -0.08745037 2.076303 0.5968880
## Test set      1.1477713 9.425368 7.359269  0.19939132 2.094873 0.8083236
##                       ACF1 Theil's U
## Training set -0.0003960831        NA
## Test set      0.4050887506 0.2830902
```



Appendix I: R Code

```r
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)
```

```
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") +
```

```r
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") +
```

```r
  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")

### Ch7.Q4.a)
data("ukcars")
autoplot(ukcars) +
  ggtitle("Quarterly UK passenger vehicle production") +
```

```
  xlab("Year") +
  ylab("Cars") +
  theme_bw()

### Ch7.Q4.b)

fit_stl <- stl(ukcars, t.window=13, s.window="periodic", robust=TRUE)

fit_stl %>%
  autoplot() +
    ggtitle("Robust STL decomposition") +
    theme_bw()

### Ch7.Q4.b)
fit_stl %>%
  seasadj() %>%
  autoplot() +
  ggtitle("Seasonally adjusted data") +
  theme_bw()

### Ch7.Q4.c)
seasAdj_ts <- seasadj(fit_stl)

fit_damp <- holt(seasAdj_ts, damped = T, h=8)
acc_damp <- accuracy(fit_damp)
rownames(acc_damp) <- "Additive Damped"

acc_reseason <- accuracy(forecast(fit_stl))
rownames(acc_reseason) <- "Reseasonalize Forecast"

knitr::kable(rbind(acc_damp, acc_reseason), caption = "Metrics")

### Ch7.Q4.d)
seasAdj_ts <- seasadj(fit_stl)

fit_holt <- holt(seasAdj_ts, h=8)
acc_holt <- accuracy(fit_holt)
rownames(acc_holt) <- "Holt's linear"

acc_reseason_ln <- accuracy(forecast(fit_stl))
rownames(acc_reseason_ln) <- "Reseasonalize Forecast"

knitr::kable(rbind(acc_holt, acc_reseason_ln), caption = "Metrics")

### Ch7.Q4.e)
fit_ets_A <- ets(ukcars, model = "AAA")
summary(fit_ets_A)
acc_ets <- accuracy(fit_ets_A)
rownames(acc_ets) <- "ETS"
#knitr::kable(acc_ets, caption = "Metrics")

### Ch7.Q4.f)
knitr::kable(rbind(acc_damp,acc_holt, acc_ets), caption = "Metrics")
```

```r
y <- ts(numeric(100))
e <- rnorm(100)
for(i in 2:100){
  y[i] <- 0.6*y[i-1] + e[i]
}
autoplot(y) +
  ggtitle("Time Series Plot when Phi = 0.6, sigma square = 1") +
    theme_bw()
y <- ts(numeric(100))
e <- rnorm(100)
for(i in 2:100){
  y[i] <- 0*y[i-1] + e[i]
}
autoplot(y) +
  ggtitle("Time Series Plot when Phi = 0, sigma square = 1") +
  theme_bw()
y <- ts(numeric(100))
e <- rnorm(100)
for(i in 2:100){
  y[i] <- e[i] + 0.6*e[i-1]
}
autoplot(y) +
  ggtitle("Time Series Plot of MA(1) when Phi = 0.6, sigma square = 1") +
  theme_bw()

y <- ts(numeric(100))
e <- rnorm(100)
for(i in 2:100){
  y[i] <- e[i] + 0*e[i-1]
}
autoplot(y) +
  ggtitle("Time Series Plot of MA(1) when Phi = 0, sigma square = 1") +
  theme_bw()
y <- ts(numeric(100))
e <- rnorm(100)
for(i in 2:100){
  y[i] <-  0.6*y[i-1] + 0.6*e[i-1] + e[i]
}
P1 <- autoplot(y) +
  ggtitle("Time Series Plot of ARMA(1,1) when Phi = 0.6 and Theta=0.6 and sigma sqr=1") +
  theme_bw()

y <- ts(numeric(100))
e <- rnorm(100)
for(i in 3:100){
  y[i] <- y[i-1]*(-0.8) + 0.3*y[i-2] + e[i]
}
P2 <- autoplot(y) +
  ggtitle("Time Series Plot of AR(2) when Phi = -0.8 and Phi = 0.3  and sigma sqr=1") +
  theme_bw()

gridExtra::grid.arrange(P1, P2, nrow=2)
data("wmurders")
```

```r
#View(wmurders)

autoplot(wmurders) +
  ggtitle("Number of women murdered each year in the US") +
  theme_bw() +
  xlab("Year")

gg1 <- ggAcf(wmurders, main="ACF plot")
gg2 <- ggPacf(wmurders, main="PACF plot")

gridExtra::grid.arrange(gg1,gg2,nrow = 2)
### Ch8.Q6.d)
summary(auto.arima(wmurders))
summary(Arima(wmurders, order = c(9,0,0)))
summary(Arima(wmurders, order = c(9,1,0)))

checkresiduals(arima(wmurders, order = c(9,0,0)))

fcast <- forecast(arima(wmurders, order = c(9,0,0)), h=3)

autoplot(fcast) +
  theme_bw()
data("austourists")
autoplot(austourists) +
  ggtitle("Quarterly number of international tourists to Australia for the period 1999-2010") +
  theme_bw() +
  xlab("Year")
gg1 <- ggAcf(austourists, main="ACF plot")
gg2 <- ggPacf(austourists, main="PACF plot")

gridExtra::grid.arrange(gg1,gg2,nrow = 2)

austourists %>%
    diff(lag=4) %>%
    ggtsdisplay()
atArm <- auto.arima(austourists)
fitarm <- Arima(austourists, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 4))
# fitarm <- Arima(austourists, order = c(0,1,1), seasonal = list(order = c(0,1,1)))
summary(atArm)
summary(fitarm)
data("usmelec")

  autoplot(usmelec, series="Data") +
  autolayer(ma(usmelec,12), series="12-MA") +
  xlab("Year") + ylab("Billions of kilowatt hours (kWh)") +
  ggtitle("Total net generation of electricity by the U.S. electric industry") +
  scale_colour_manual(values=c("Data"="grey50","12-MA"="red"),
                      breaks=c("Data","12-MA"))


lambda <- BoxCox.lambda(usmelec)
autoplot(BoxCox(usmelec,lambda))
```

```
usmelec %>%
    diff(lag=12) %>%
    ggtsdisplay()

usmelec %>%
    diff(lag=12) %>%
    diff() %>%
    ggtsdisplay()
fit1 <- Arima(usmelec, order = c(1,0,1), seasonal = c(1,1,1))
aic1 <- fit1$aic
fit2 <- Arima(usmelec, order = c(2,1,1), seasonal = c(2,1,1))
aic2 <- fit2$aic
fit3 <- Arima(usmelec, order = c(2,2,2), seasonal = c(2,2,1))
aic3 <- fit3$aic
fit4 <- auto.arima(usmelec)
aic4 <- fit4$aic

aic <- cbind(aic1,aic2,aic4,aic4)
rownames(aic) <- "AIC"
colnames(aic) <- c("Arima(1,0,1)(1,1,1)","Arima(2,1,1)(2,1,1)","Arima(2,2,2)(2,2,1)", "Arima(1,0,2)(0,1
knitr::kable(aic, caption = "AIC Values")


checkresiduals(fit2)
elect <- read_csv("electricity-overview.csv")
colnames(elect) <- c("Month", "Electricity")
elect_ts <- ts(elect$Electricity, start = c(1973,1), frequency = 12)
fcast <- forecast(fit2, h=15)
accuracy(fcast, elect_ts)

autoplot(fcast)
```