MACHINE LEARNING ALGORITUMS TO FORECAST

Final Exam

MASTER OF SCIENCE IN BUSINESS

By

Manpreet Singh

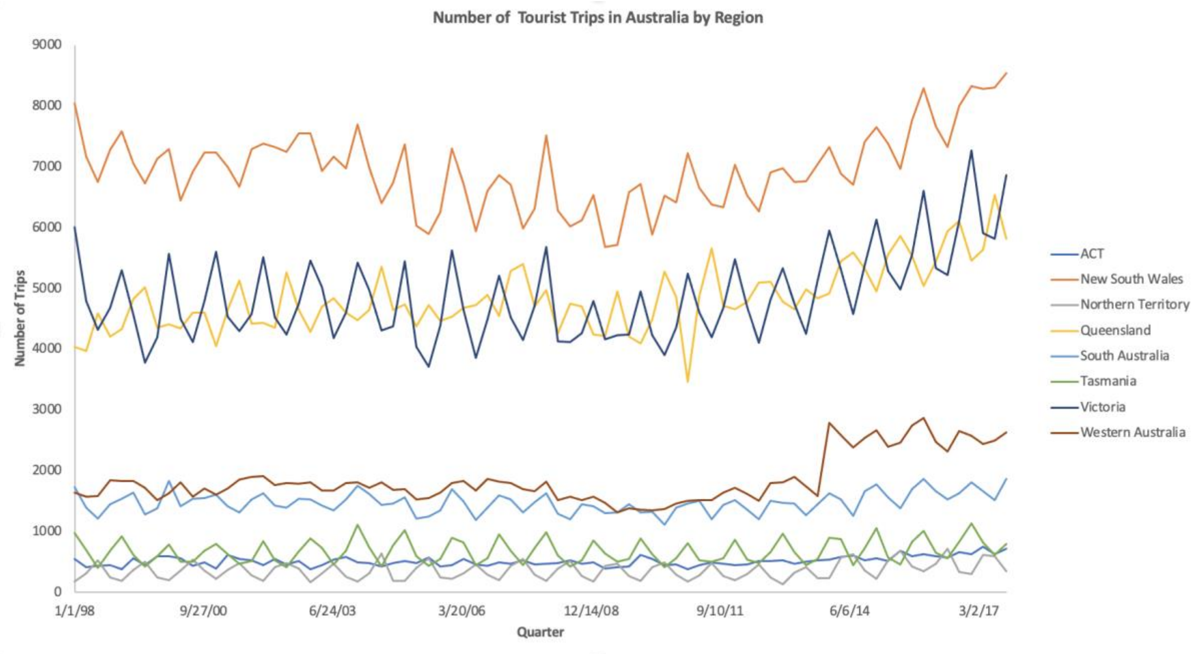
Register No: V00953330

Under the guidance of

Jason Merrick, Ph. D.

VCU School of Business

Australian Tourism



The figure above shows the number of trips to various regions in Australia from 1998 to 2017. You are asked to focus trips to the Victoria region, and produce as accurate as possible forecasts. Start by partitioning the data: use the period from 1998 to 2014 as the training set, keeping 2015, 2016, and 2017 as the validation set.

1. Fit a regression model to the training data with a linear trend and additive seasonality (10 points)
   - Create a plot to compare the fit to the training data. Do not show the validation data.
   - Create a plot to show the forecast and prediction interval.
2. Fit an ARIMA model (20 points)
   - Looking at the training data:
     - What level of differencing do you need?
     - Create an ACF and PACF plot on the differenced data.
     - What AR and MA terms do you need?
   - What ARIMA model would you recommend for this data? (If you are unsure then try several)
   - Fit your recommended ARIMA model.
   - Create a plot to compare the fit to the training data. Do not show the validation data.
   - Create a plot to show the forecast and prediction interval.
3. Use auto.arima() to fit an ARIMA (p, d, q) (P, D, Q) model to the training data (10 points)
   - Create a plot to compare the fit to the training data. Do not show the validation data.

    - Create a plot to show the forecast and prediction interval.
4. Fit an exponential smoothing model (20 points)
    - Looking at the training data:
        - Is there a trend? What form does it take?
        - Is there seasonality? What form does it take?
        - What ETS() model would you recommend for this data? (If you are unsure then try several)
    - Fit your recommended ets model.
    - Create a plot to compare the fit to the training data. Do not show the validation data.
    - Create a plot to show the forecast and prediction interval.
5. Fit an ETS model allowing the algorithm to choose the structure for error, trend and seasonality from the training data (10 points)
    - Create a plot to compare the fit to the training data. Do not show the validation data.
    - Create a plot to show the forecast and prediction interval.
6. Assess the predictive accuracy of your five models in cross-validation (10 points)
7. Which model would you recommend to the Australian tourism board for forecasting trips to the Victoria region (20 points)
    - Create a plot to compare the fit of your recommended model to the training and validation data.
    - Create a plot to show a 3-year forecast and prediction interval for your chosen model based on the full dataset.

R-markdown attached after report(Page-18 to 39)
Extra Exploratory Analysis trying different libraries (Page 39 -53)

Final Project

1.Fit a regression model to the training data with a linear trend and additive seasonality.

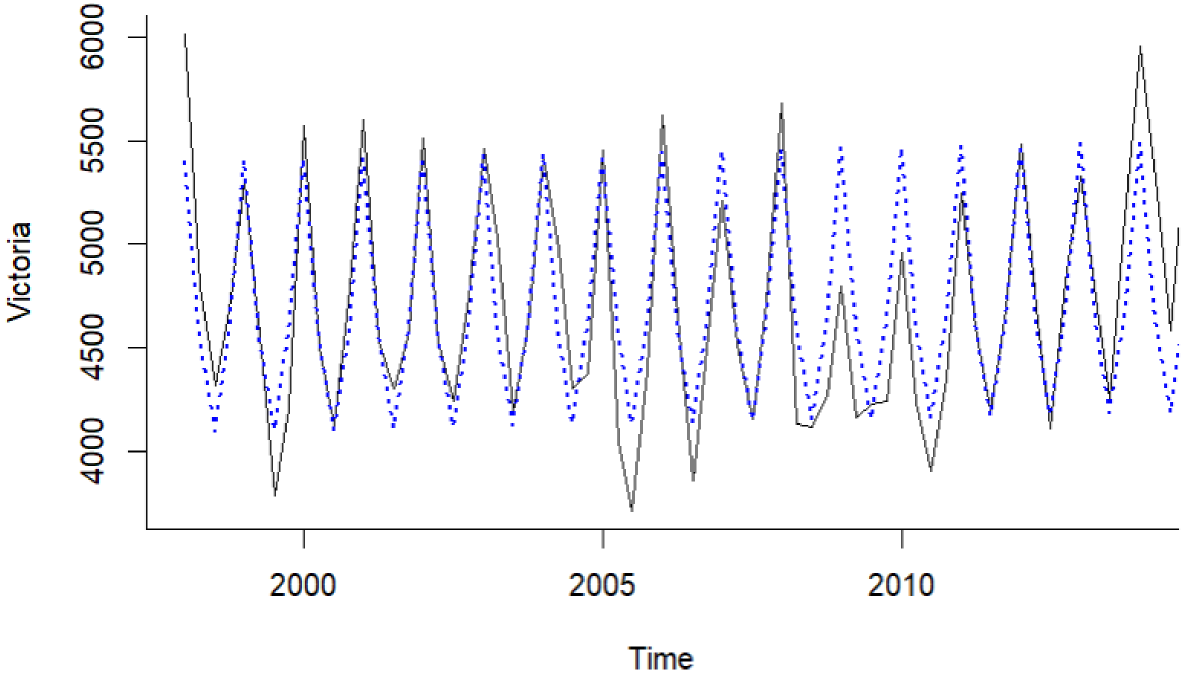
First of all, I read the data from csv file and filtered the data for Victoria only. Then I separated the data into train and test data. The data from 1998 to 2014 is used for training data and data from 2015 to 2017 is used as test data. Then I fitted the regression model with a linear trend and additive seasonality. The summary of linear model is shown below:

```
Call:
tslm(formula = train.ts ~ trend + season)

Residuals:
    Min      1Q  Median      3Q     Max 
-666.77 -220.61   11.57  158.84  733.09 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5399.199     94.335  57.234  < 2e-16 ***
trend           1.438      1.841   0.781    0.438    
season2      -860.197    102.068  -8.428 6.39e-12 ***
season3     -1310.335    102.118 -12.832  < 2e-16 ***
season4      -836.805    102.201  -8.188 1.68e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 297.5 on 63 degrees of freedom
Multiple R-squared:  0.7318,	Adjusted R-squared:  0.7148 
F-statistic: 42.97 on 4 and 63 DF,  p-value: < 2.2e-16
```

Above plot shows that the p-value of coefficient trend is greater than significance level 0.05, so trend don’t have significant relationship with dependent variable. The p-values for seasonal coefficients are less than 0.05 which means they have significant relation with dependent variable. The R-squared value of model is 0.7318 which mean 73.18% of variation in output variable is explained by this model. The coefficient value for Q2 is -860.92 which means average trips in Q2 are -860.92 less as compare to Q1. The coefficient value for Q3 is -1310.335 which means on average Q3 has -1310.335 trips less as compare to Q1. The Q4 coefficient value is -836.805 which means for Q4 the average trips are 836.805 less as compare to Q1.

Create a plot to compare the fit to the training data. Do not show the validation data.

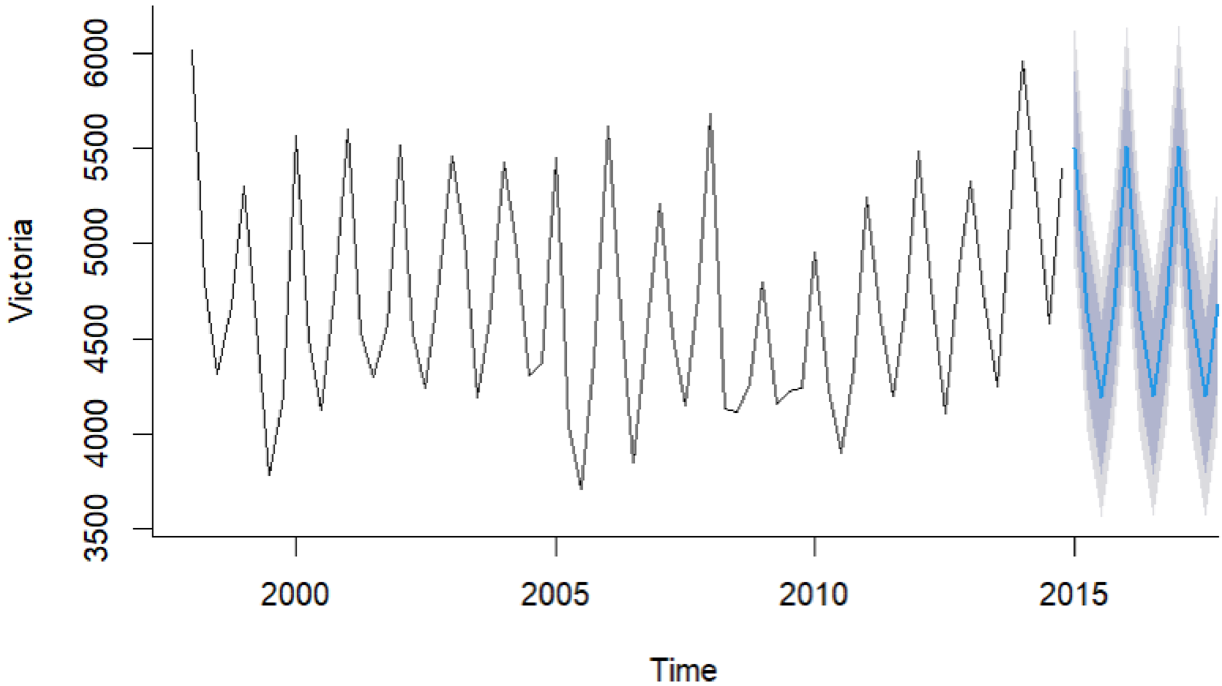
The plot for fit to the training data is shown below:



The plot shows that there is slight variation in the actual value and predicted values which is expected as no model can fits exactly the same actual values.

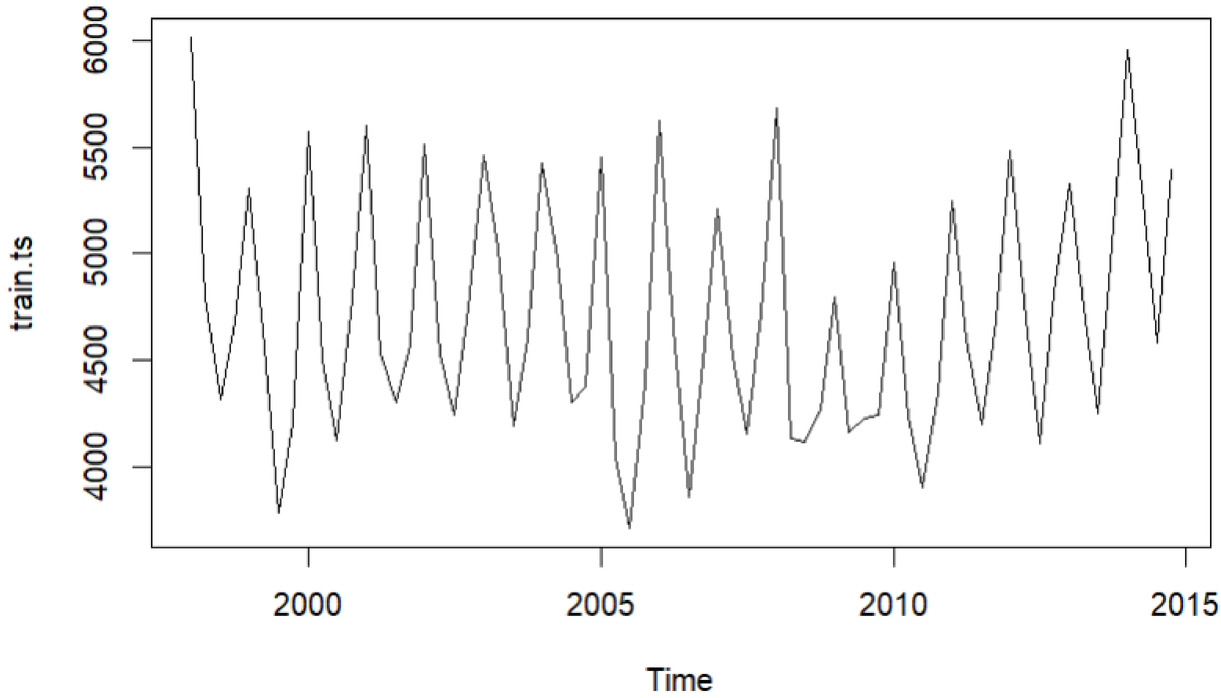
Create a plot to show the forecast and prediction interval.

The plot to shows the forecast and prediction interval is shown below:



In above plot the blue line shows the forecast and grey area shows the prediction interval. From above plot it can be seen that the forecast and prediction intervals aligned with the training data so the prediction from model seems reasonable.

2) Fit an ARIMA model
Looking at the training data:

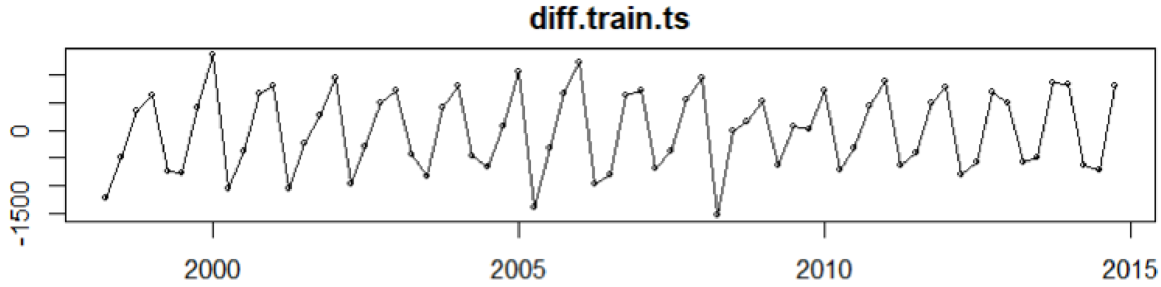


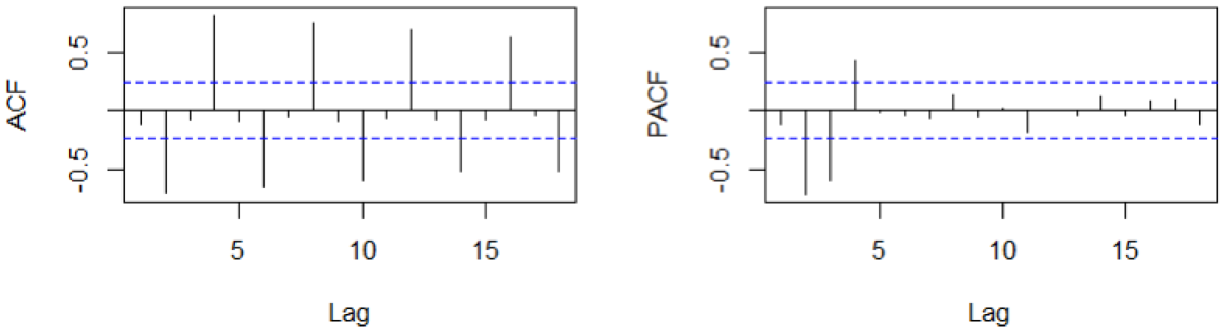
What level of differencing do you need?

From above plot it looks like the 1 or 2 level of differencing I needed. I used ndiffs() to check and it's output value is 0 and the nsdiffs() outputted the value 1 so I need the level 1 differencing for this.

Create an ACF and PACF plot on the differenced data.

The ACF an PACF plot on the differenced data is shown below:





What AR and MA terms do you need?

The AR and MA terms 1 will be good fit to this data, but I have to check by building modes with different terms to find out which one has lowest AIC value. I used models with different orders. The first model has (1,0,0) for order and (1,1,0) for seasonal. The second model has (0,0,1) for order and (0,1,1) for seasonal (0,1,1). The third model has (0,1,1) for order and (1,1,1). After fitting all these models, I concluded that third model that with order p = 0, d = 1, q = 1, P = 1, D = 1, Q = 1 has lowest root mean square error and mean percentage model, so this model is best fit among all three models.

Fit your recommended ARIMA model.

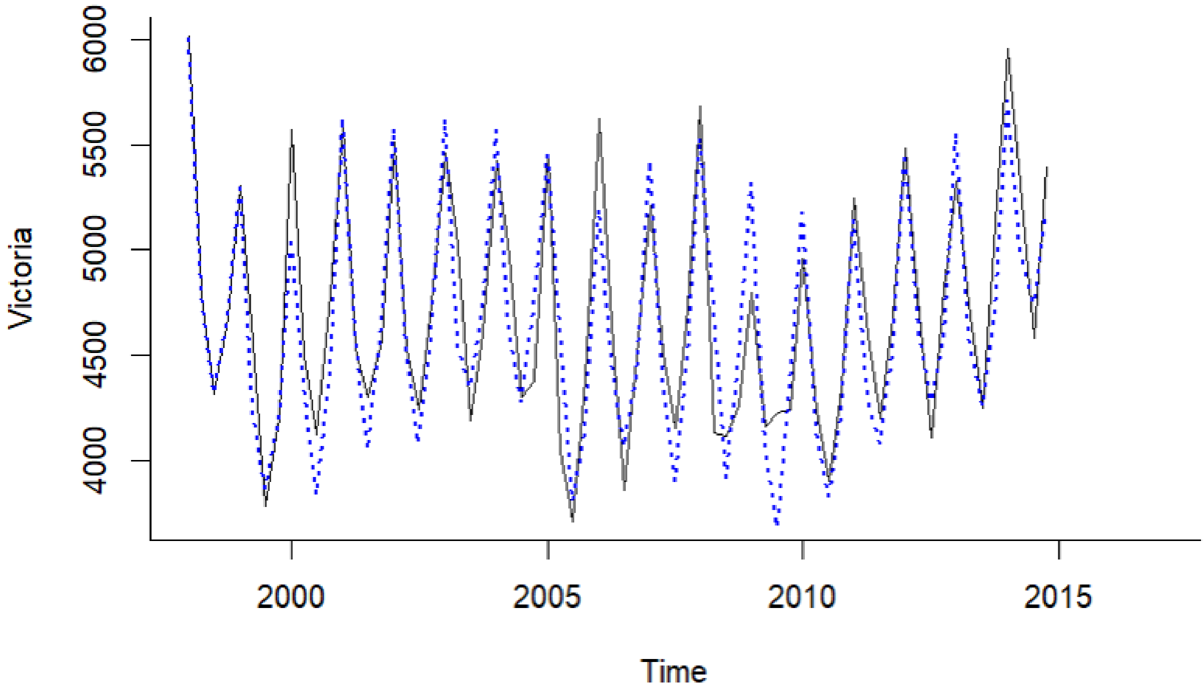
I fitted the recommended ARIMA model with p =0, d = 1, q = 1, P = 1, D = 1, Q = 1. The output of recommend arima model is shown below:

```
Series: train.ts
ARIMA(0,1,1)(1,1,1)[4]

Coefficients:
          ma1     sar1     sma1
      -0.5077   0.0244  -0.8676
s.e.   0.1287   0.1610   0.1399

sigma^2 = 66453:  log likelihood = -440.54
AIC=889.08   AICc=889.77   BIC=897.65

Training set error measures:
                   ME     RMSE      MAE      MPE     MAPE      MASE
Training set 54.64954 242.1466 189.2346 1.058719 4.080805 0.7523571
                    ACF1
Training set -0.02291503
```

The above output shows that AIC value of this model is 889.08, the root mean squared on training data is 241.1466 and mean absolute percentage error on training data is 4.090805.

Create a plot to compare the fit to the training data. Do not show the validation data.

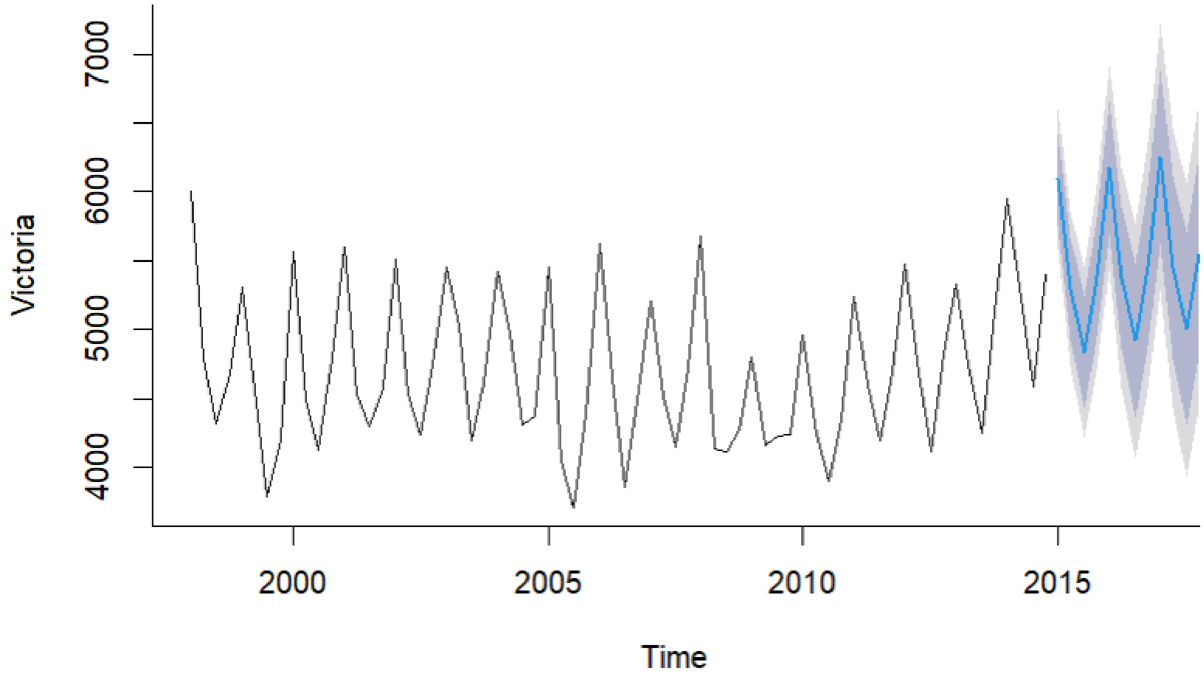
The plot is shown below:



The plot shows that the variation in actual and fitted values is not too high which means the model is a good fit.

Create a plot to show the forecast and prediction interval.

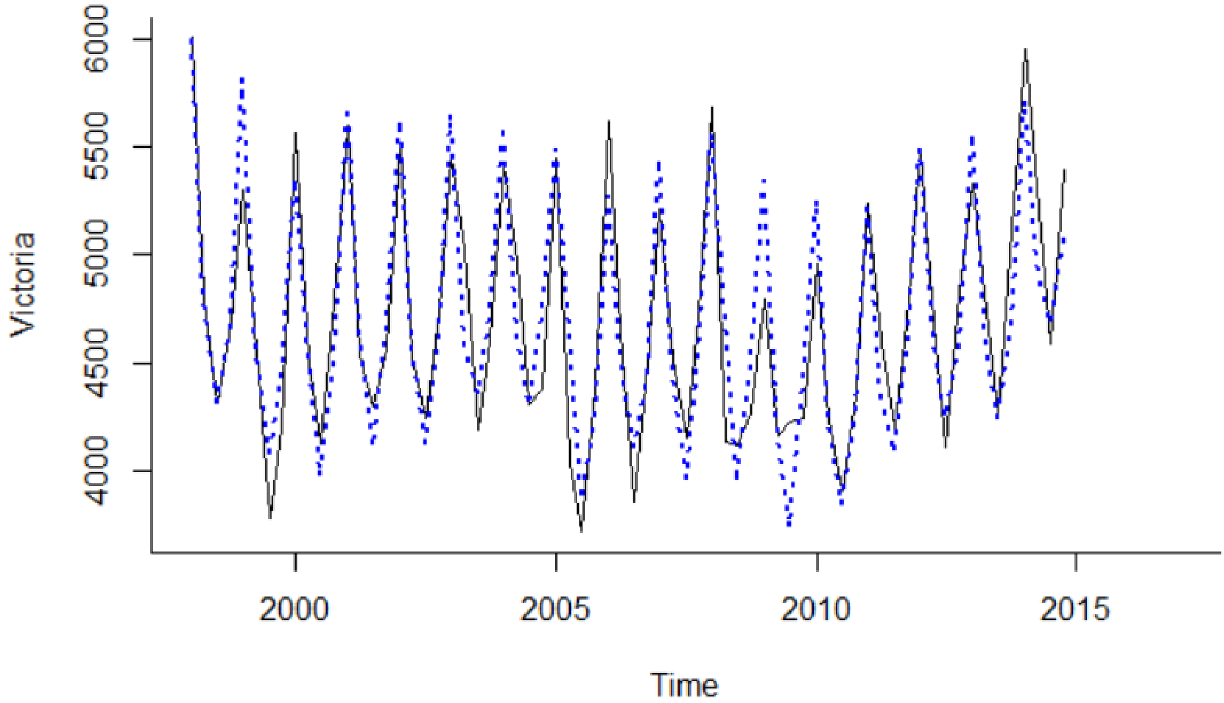
The plot for forecast and prediction interval is shown below:



3)Use auto.arima() to fit an ARIMA (p, d, q) (P, D, Q) model to the training data.
I used auto.arima function to fit the auto ARIMA model. The model has chosen order = (1,0,1) and seasonal = (0,1,1). The output of model is shown below:

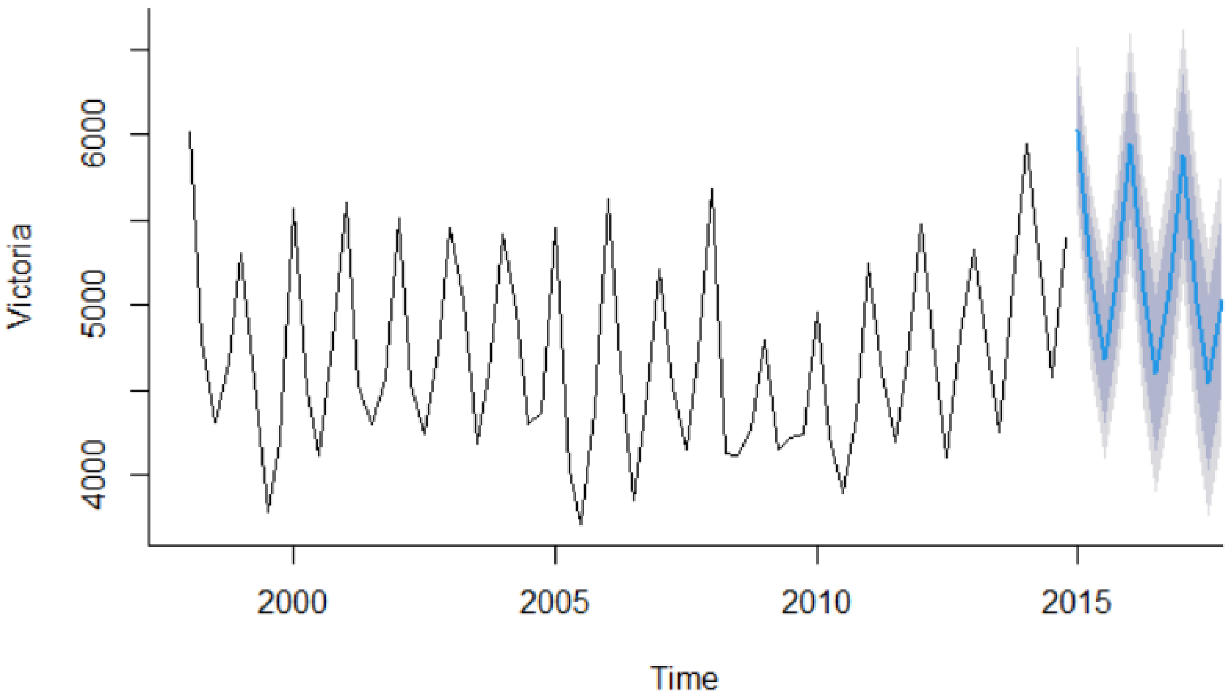
```
Series: train.ts
ARIMA(1,0,1)(0,1,1)[4]

Coefficients:
         ar1      ma1     sma1
      0.9330  -0.4702  -0.9188
s.e.  0.1165   0.1625   0.2401

sigma^2 = 62422:  log likelihood = -445.65
AIC=899.3   AICc=899.98   BIC=907.94

Training set error measures:
                   ME     RMSE      MAE         MPE     MAPE      MASE       ACF1
Training set 3.538936 236.6351 184.2156 -0.04522999 3.965776 0.7324024 -0.0183292
```

Create a plot to compare the fit to the training data. Do not show the validation data.
The plot to compare the fit to the training data is shown below:



From above plot, there is not much variation in the fitted values and training data which means the model may fits well to the data.

Create a plot to show the forecast and prediction interval.



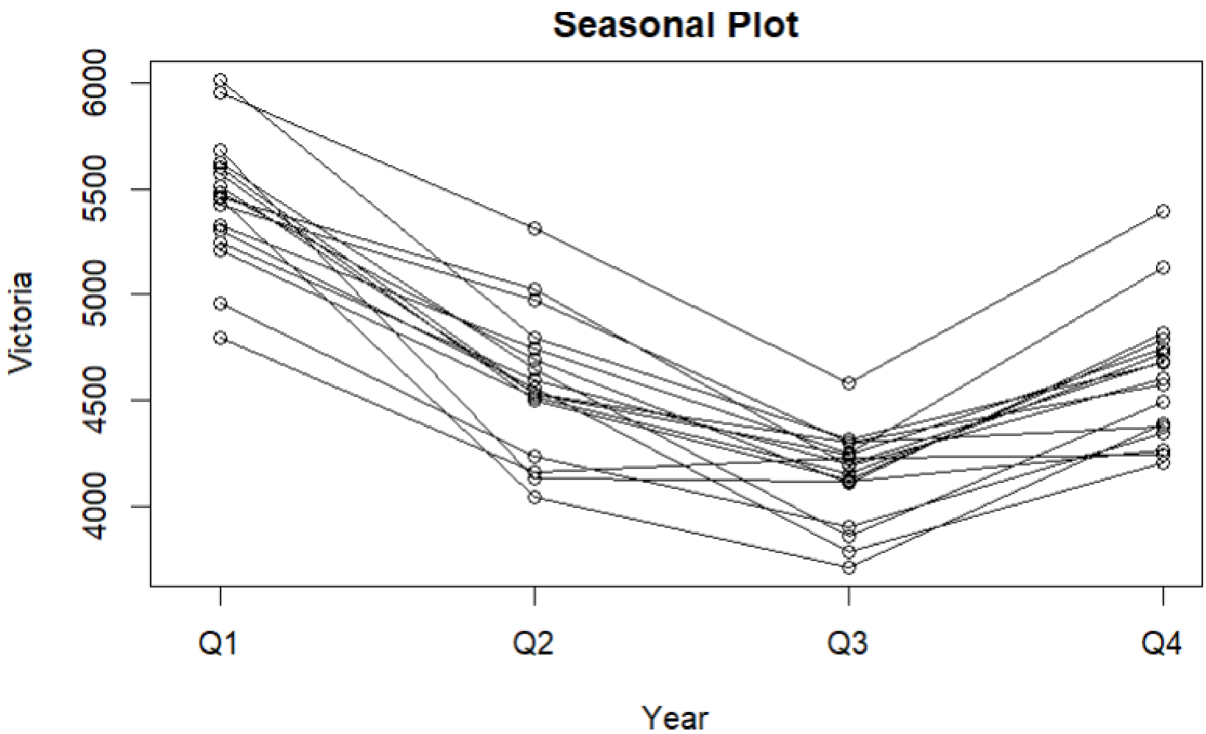
4)Fit an exponential smoothing model

Is there a trend? What form does it take?

The plot for training data shows that there is no trend in the training.

Is there seasonality? What form does it take?

The plot for seasonality is shown below:



The plot shows that there is a seasonality in the data and it occurred at each quarter in year.

What ETS () model would you recommend for this data? (If you are unsure then try several)

I used different ETS model to find out the best model among all of them. The different models, I tried additive error, multiplicative trend, additive season. The second model is additive error, multiplicative trend, no season. The third model is additive error, additive trend, additive season. The third model has additive error, additive trend and no season. The fifth model has multiplicative error, multiplicative trend, and additive season. The sixth model has multiplicative error, multiplicative trend and no season. The seventh model multiplicative error, additive trend, additive season. The seventh model has multiplicative error, additive trend, additive season. The eight model has multiplicative error, additive trend and no season.

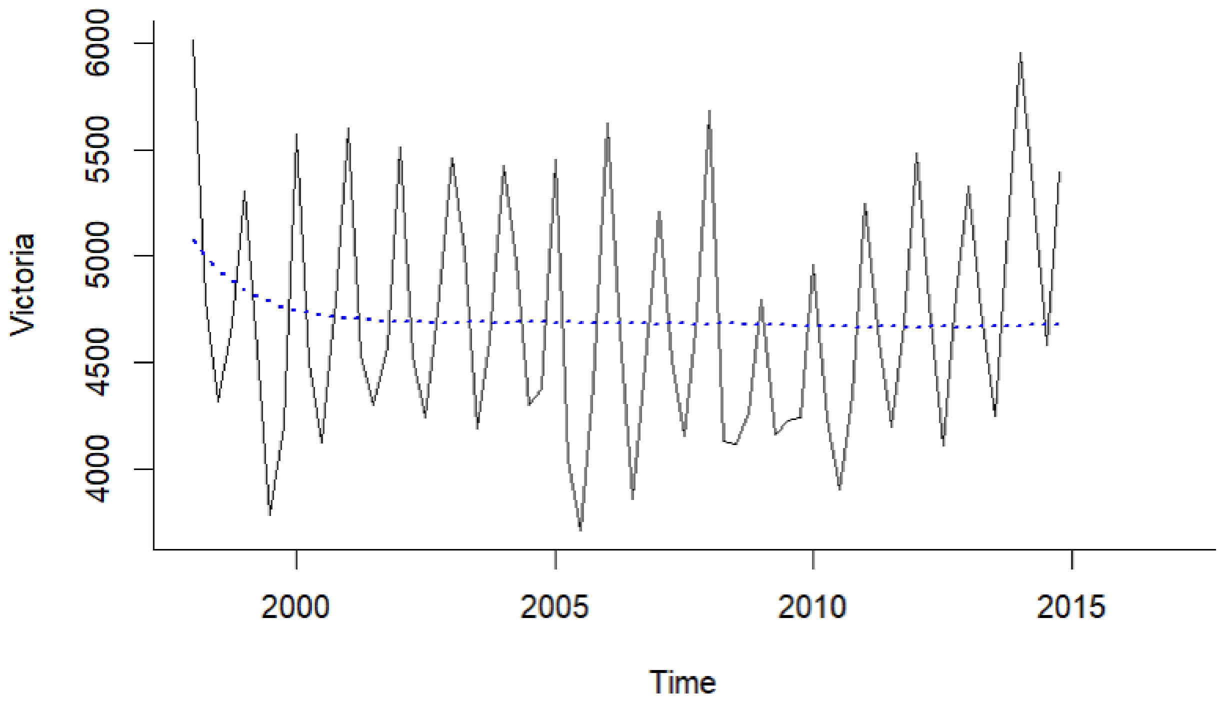
Fit your recommended ets model.

From above all models the model with additive trend, multiplicative trend, no season has lowest. I fitted the models. The output of fitted models is shown below:

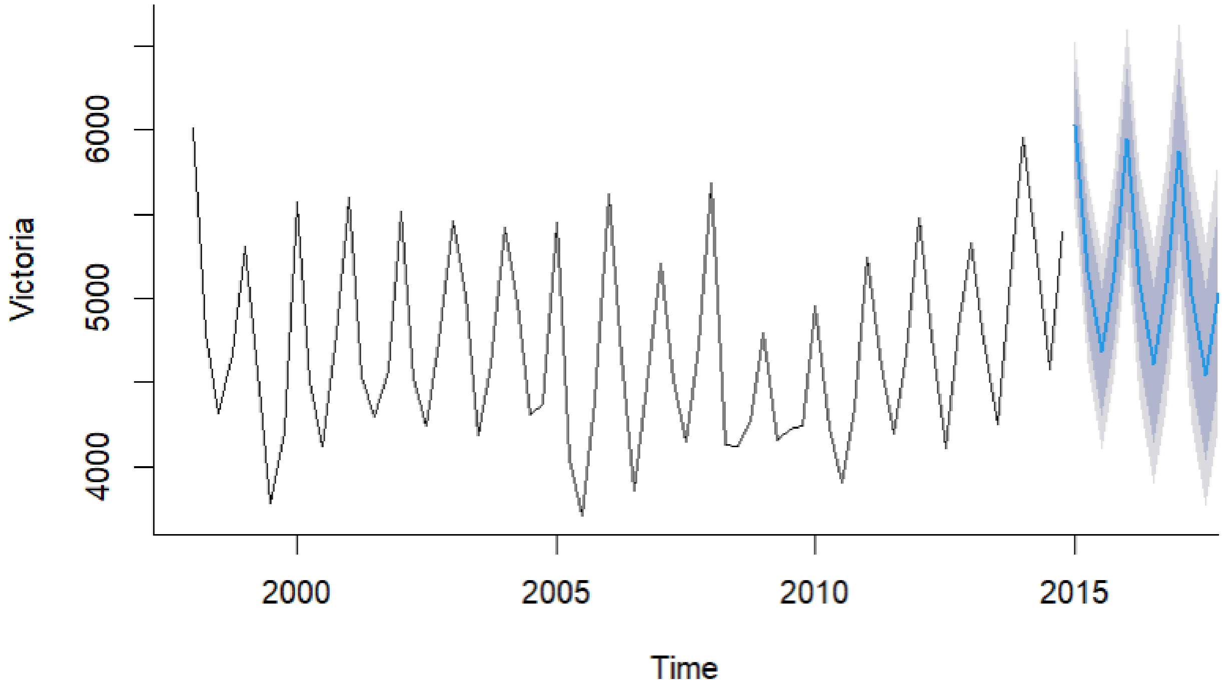
```
ETS(A,Md,N)

Call:
 ets(y = train.ts, model = "AMN", restrict = FALSE)

  Smoothing parameters:
    alpha = 0.0048
    beta  = 1e-04
    phi   = 0.8

  Initial states:
    l = 5180.5844
    b = 0.9753

  sigma:  568.9684

     AIC     AICc      BIC
1156.493 1157.870 1169.810
```

Create a plot to compare the fit to the training data. Do not show the validation data.

The plot to compare the fit to the training data is shown below:



Create a plot to show the forecast and prediction interval.

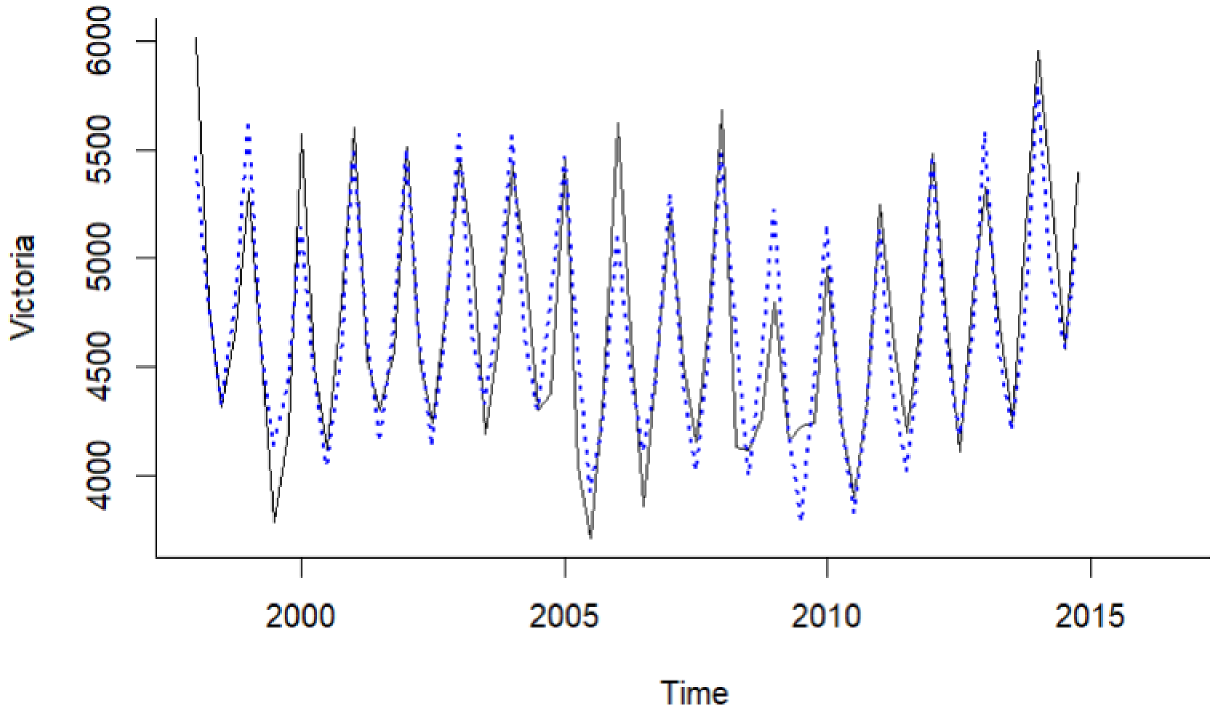


5)Fit an ETS model allowing the algorithm to choose the structure for error, trend and seasonality from the training data.

The output of ETS model allowing the algorithm to choose the structure for error, trend and seasonality is shown below:

```
ETS(M,N,M)

Call:
 ets(y = train.ts, restrict = FALSE)

  Smoothing parameters:
    alpha = 0.4366
    gamma = 1e-04

  Initial states:
    l = 4700.6455
    s = 0.9794 0.8806 0.9777 1.1624

  sigma:  0.0526

     AIC     AICc      BIC
1042.427 1044.293 1057.963
```

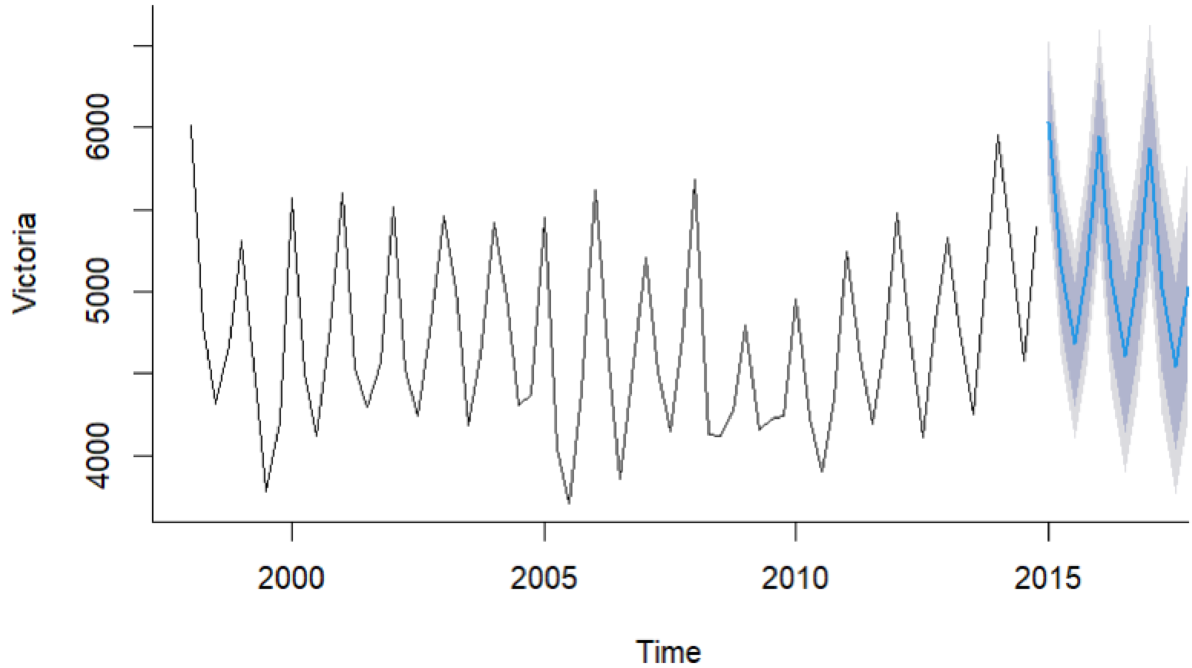
Create a plot to compare the fit to the training data. Do not show the validation data.



The plot shows that there is not much difference in the fitted values and actual values.

Create a plot to show the forecast and prediction interval.

The plot to show the forecast and prediction interval for ETS model is shown below:



6.Assess the predictive accuracy of your five models in cross-validation

The accuracies of different models are shown below:

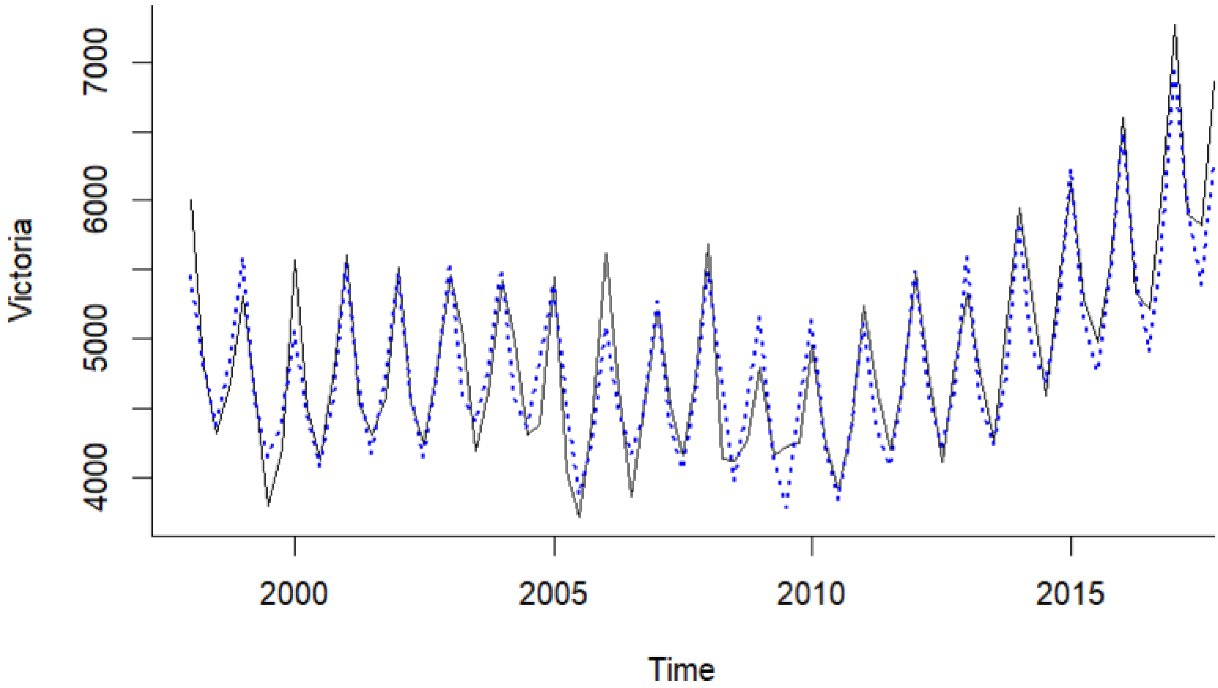
```
[1] "Accuracy for Linear Model"
                                   ME      RMSE       MAE        MPE
Training set   -0.0000000000002676944  286.3821  215.5395 -0.3678315
Test set     1167.7638634681366056611 3 1260.8883 1167.7639 19.3354736
                  MAPE      MASE      ACF1 Theil's U
Training set  4.618621 0.8569399 0.5134102        NA
Test set     19.335474 4.6427838 0.5065477  1.519638
[1] "Accuracy for main Arima Model"
                    ME     RMSE      MAE      MPE     MAPE      MASE
Training set  54.64954 242.1466 189.2346 1.058719 4.080805 0.7523571
Test set     440.42417 606.2667 449.3299 6.949725 7.117039 1.7864412
                    ACF1 Theil's U
Training set -0.02291503        NA
Test set      0.43484906 0.7322113
[1] "Accuracy for auto Arima Model"
                     ME     RMSE      MAE         MPE      MAPE      MASE
Training set   3.538936 236.6351 184.2156 -0.04522999  3.965776 0.7324024
Test set     739.862432 910.9337 739.8624 11.95974713 11.959747 2.9415376
                   ACF1 Theil's U
Training set -0.0183292        NA
Test set      0.5479558   1.09991
[1] "Accuracy fro main ETS model"
                     ME      RMSE       MAE       MPE      MAPE     MASE
Training set  -10.39124  547.6511  459.9462 -1.547655  9.782864 1.828650
Test set     1239.27091 1412.6480 1239.2709 19.922392 19.922392 4.927081
                   ACF1 Theil's U
Training set 0.08890038        NA
Test set     0.10493480  1.631959
[1] "Accuracy for auto ETS model"
                     ME     RMSE      MAE         MPE      MAPE      MASE
Training set   3.538936 236.6351 184.2156 -0.04522999  3.965776 0.7324024
Test set     739.862432 910.9337 739.8624 11.95974713 11.959747 2.9415376
                   ACF1 Theil's U
Training set -0.0183292        NA
Test set      0.5479558   1.09991
```

7.Which model would you recommend to the Australian tourism board for forecasting trips to the Victoria region.

The auto ETS model has lowest root mean square error for test data, so I would recommend aut ETS model to the Australian tourism board for forecasting trips to the Victoria region.

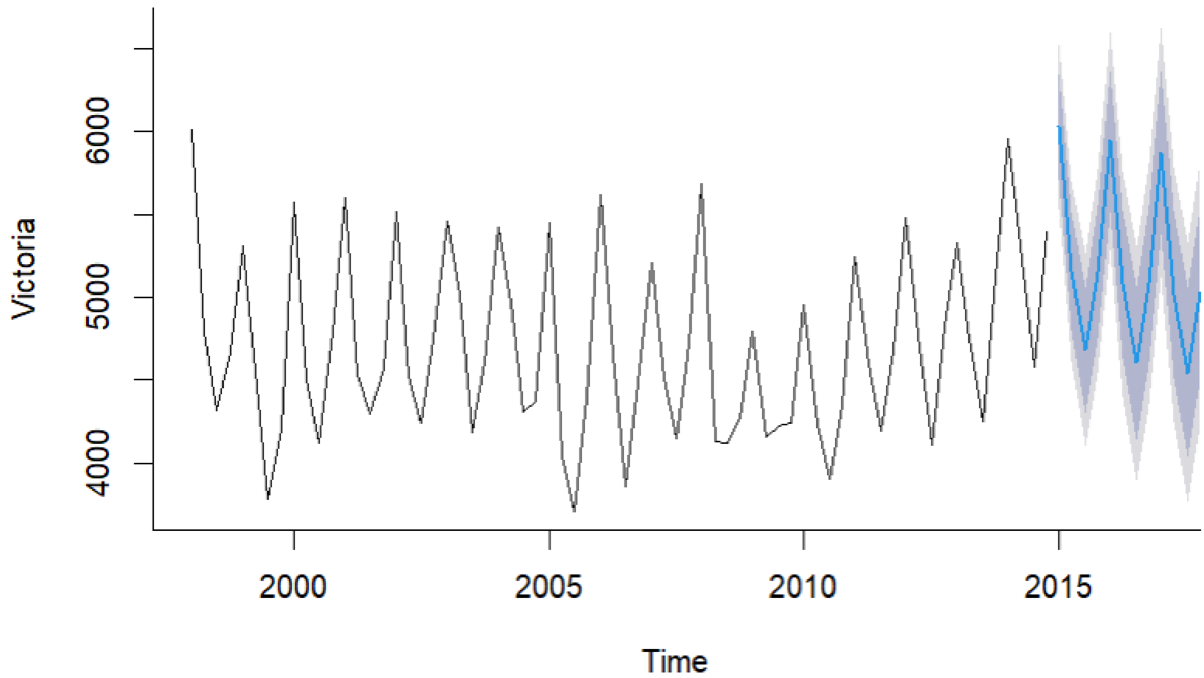
Create a plot to compare the fit of your recommended model to the training and validation data.

The plot is shown below:



Create a plot to show a 3-year forecast and prediction interval for your chosen model based on the full dataset.

The plot for prediction and confidence interval is also shown below:



Comparing all the models and making suggestion on which can be used in real life for reasonable predictions.

After comparing all the models and their performance using different metrics like root mean squared error and mean absolute percentage error, I concluded that the auto ETS model has best performance among all models and can be used in real life for reasonable applications.

Giving a real world application of the analysis you have done.

The above analysis has done using real world example of number of trips to Victoria regions in Australia from 1998 to 2017. There can be many real world applications for time series analysis. These analyses can be used for sales of a particular store over time, weather temperature of a particular region, for passengers in airline etc….

Conclusion

In this project, I have used data for number of trips to region Victoria in Australia. The data is from year 1998 to year 2017. I split the data into train data from 1998 to 2014 and test data from 2015 to 2017. Then I implemented different time series forecasting methods to find out the best one. The methods that I used are linear model with trend and seasonal, ARIMA models with different values for order and seasonality, auto ARIMA model, ETS model with different combinations for error, trend and seasonality, and then auto ETS model. After implementing all the models on training data, I compared their accuracy for test data and found out that auto ETS model has highest accuracy on test data and is the best model among all of them. So, I concluded that auto ETS model can be used to make forecast for number of trips to Victoria region in future.

R-markdown

R Notebook

Hide

```
#reading data from csv file
data <- read.csv("AustralianTourism.csv")
head(data)
```

	Quarter <chr>	ACT <dbl>	New.South.Wales <dbl>	Northern.Territory <dbl>	Queensland <dbl>	So
1	01/01/98	551.0019	8039.795	181.4488	4041.370	
2	04/01/98	416.0256	7166.014	313.9362	3967.905	
3	07/01/98	436.0290	6747.936	528.4369	4593.894	
4	10/01/98	449.7984	7282.082	247.7028	4202.829	
5	01/01/99	378.5728	7584.777	184.8896	4332.491	
6	04/01/99	558.1781	7054.039	366.0928	4824.480	

6 rows | 1-8 of 9 columns

Hide

```
#converting data into time series object
library(forecast)
```

```
Warning: package ‘forecast’ was built under R version 4.2.2Registered S3 me
thod overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

Hide

```
data.ts <- ts(data$Victoria, start = c(1998, 1), frequency = 4)
data.ts
```

```
         Qtr1     Qtr2     Qtr3     Qtr4
1998 6010.424 4795.247 4316.845 4674.829
1999 5304.334 4561.711 3783.601 4201.422
2000 5566.857 4501.904 4122.392 4787.177
2001 5600.465 4532.807 4299.892 4573.434
2002 5513.749 4527.764 4242.311 4743.849
2003 5457.857 5022.066 4188.183 4605.985
2004 5423.238 4971.627 4303.826 4377.440
```

```
2005 5449.373 4043.765 3713.005 4389.352
2006 5619.681 4650.075 3854.865 4489.165
2007 5210.300 4517.820 4151.475 4718.530
2008 5680.625 4132.828 4117.808 4266.836
2009 4797.162 4159.243 4225.957 4242.203
2010 4954.956 4235.305 3904.572 4348.585
2011 5243.208 4598.356 4195.711 4684.458
2012 5480.851 4690.424 4110.335 4819.917
2013 5328.953 4743.042 4252.258 5124.552
2014 5951.318 5310.471 4583.531 5393.294
2015 6126.936 5284.471 4981.169 5550.824
2016 6599.700 5335.230 5221.881 6113.016
2017 7269.527 5901.387 5817.972 6865.399
```

Hide

```
#splitting data into train and test data
nTest <- 12
nTrain <- length(data.ts) - nTest
train.ts <- window(data.ts, start = c(1998, 1), end = c(2014,4))
test.ts <- window(data.ts, start = c(2015, 1))
test.ts
```

```
         Qtr1     Qtr2     Qtr3     Qtr4
2015 6126.936 5284.471 4981.169 5550.824
2016 6599.700 5335.230 5221.881 6113.016
2017 7269.527 5901.387 5817.972 6865.399
```

Fit a regression model to the training data with a linear trend and additive seasonality

Hide

```
#fitting a regression model
train.lm <- tslm(train.ts ~ trend + season)
summary(train.lm)
```

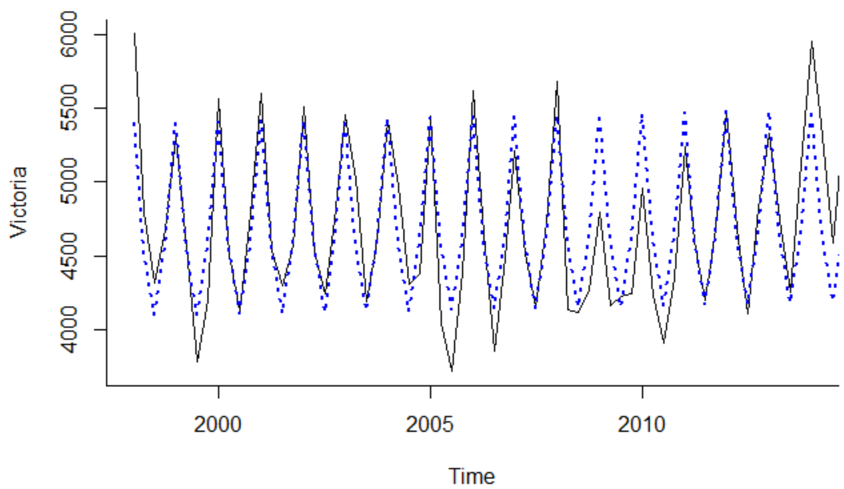
```

Call:
tslm(formula = train.ts ~ trend + season)


Residuals:
    Min      1Q  Median      3Q     Max
-666.77 -220.61   11.57  158.84  733.09
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5399.199     94.335  57.234  < 2e-16 ***
trend           1.438      1.841   0.781    0.438
season2      -860.197    102.068  -8.428 6.39e-12 ***
season3     -1310.335    102.118 -12.832  < 2e-16 ***
season4      -836.805    102.201  -8.188 1.68e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 297.5 on 63 degrees of freedom
Multiple R-squared:  0.7318,    Adjusted R-squared:  0.7148
F-statistic: 42.97 on 4 and 63 DF,  p-value: < 2.2e-16
```

o Create a plot to compare the fit to the training data. Do not show the validation data.

Hide

```
#plot to compare fit to the training data
plot(train.ts, ylab = "Victoria", xlab = "Time", bty = "l",
     xlim = c(1998,2014), main = "")
train.lm.pred <- forecast(train.lm, h = 12)
lines(train.lm.pred$fitted, lwd = 2, col = "blue", lty = 3)
```

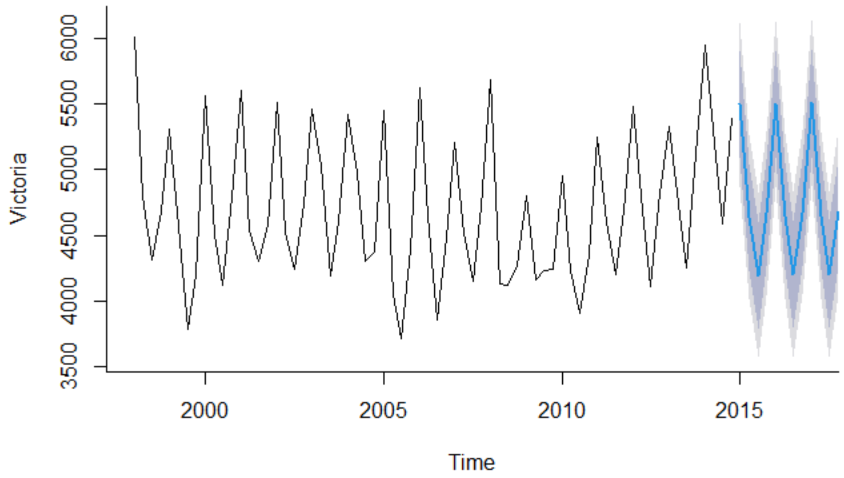


o Create a plot to show the forecast and prediction interval.

Hide

```
#showing forecast and prediction interval
```

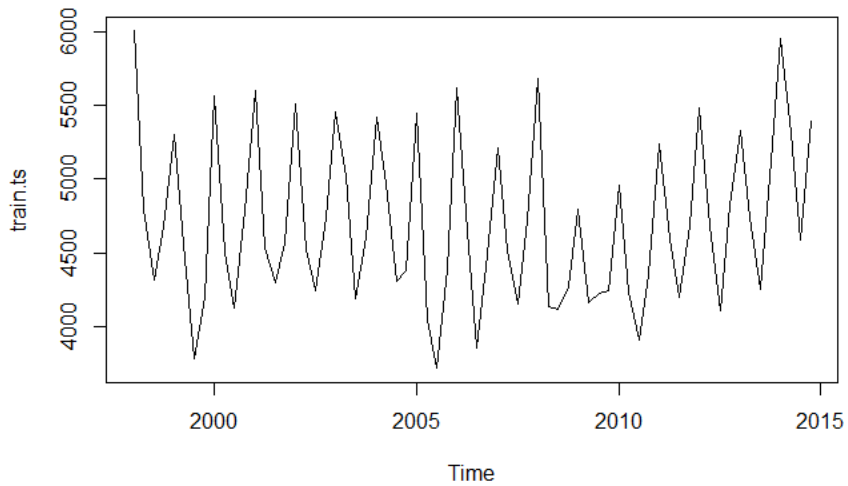
```
plot(train.lm.pred, ylab = "Victoria", xlab = "Time", bty = "l",
      xlim = c(1998,2017), main = "")
```



Fit an ARIMA model o Looking at the training data: ♣ What level of differencing do you need?

Hide

```
#plotting training data
plot(train.ts)
```



Hide

```
#checking number of differences needed
ndiffs(train.ts)
[1] 0
```

Hide

```
#so D = 0
```

Hide

```
nsdiffs(train.ts)
```

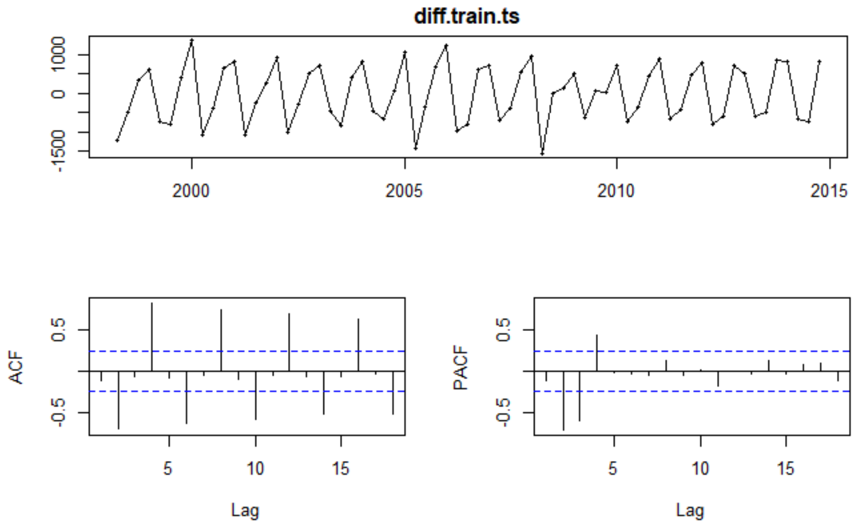
```
[1] 1
```

Hide

```
#so d = 1
```

Hide

```
#taking difference and plotting acf and pacf
diff.train.ts <- diff(train.ts, 1)
tsdisplay(diff.train.ts)
```



♣ Create an ACF and PACF plot on the differenced data. ♣ What AR and MA terms do you need? o What ARIMA model would you recommend for this data? (If you are unsure then try several)

Hide

```
#ARIMA model with order = c(1,0,0), seasonal = c(1,1,0)
ARIMA.fit1 <- Arima(train.ts, order = c(1,0,0), seasonal = c(1,1,0))
summary(ARIMA.fit1)
```

```
Series: train.ts
ARIMA(1,0,0)(1,1,0)[4]


Coefficients:
         ar1     sar1
      0.4513  -0.4025
s.e.  0.1166   0.1222


sigma^2 = 80264:  log likelihood = -451.62
```

```
AIC=909.25   AICc=909.65   BIC=915.73


Training set error measures:
                   ME     RMSE      MAE        MPE     MAPE      MASE
ACF1
Training set 13.6648 270.5215 210.7866 0.09317224 4.592481 0.8380432 -0.065
39552
```

Hide

```
#ARIMA model with order = c(0,0,1), seasonal = c(0,1,1)
ARIMA.fit2 <- Arima(train.ts, order = c(0,0,1), seasonal = c(0,1,1))
summary(ARIMA.fit2)
```

```
Series: train.ts
ARIMA(0,0,1)(0,1,1)[4]


Coefficients:
         ma1     sma1
      0.3977  -0.5513
s.e.  0.1060   0.1550


sigma^2 = 79147:  log likelihood = -451.55
AIC=909.1   AICc=909.5   BIC=915.58


Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE
ACF1
Training set 12.78142 268.6318 204.9116 0.04374984 4.421909 0.8146853 0.089
18046
```

Hide

```
#ARIMA model with order = c(0,1,1), seasonal = c(1,1,1)
ARIMA.fit3 <- Arima(train.ts, order = c(0,1,1), seasonal = c(1,1,1))
summary(ARIMA.fit3)
```

```
Series: train.ts
ARIMA(0,1,1)(1,1,1)[4]


Coefficients:
          ma1    sar1     sma1
      -0.5077  0.0244  -0.8676
```

```
s.e.    0.1287  0.1610   0.1399


sigma^2 = 66453:  log likelihood = -440.54
AIC=889.08   AICc=889.77   BIC=897.65


Training set error measures:
                   ME      RMSE       MAE       MPE      MAPE       MASE
ACF1
Training set 54.64954 242.1466 189.2346 1.058719 4.080805 0.7523571 -0.0229
1503
```

#p = 0, d = 1, q = 1, P = 1, D = 1, Q = 1

#o Fit your recommended ARIMA model.

Hide

```
#ARIMA model with order = c(0,1,1), seasonal = c(1,1,1)
main.ARIMA <- Arima(train.ts, order = c(0,1,1), seasonal = c(1,1,1))
summary(main.ARIMA)
```

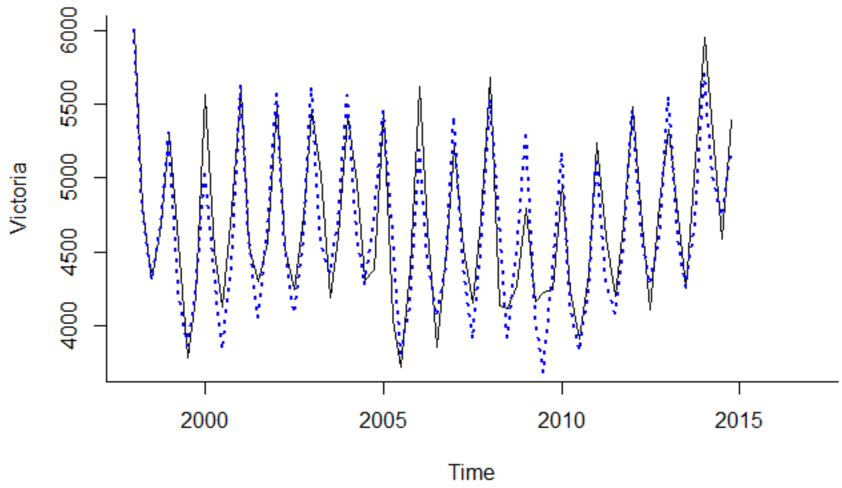

```
Series: train.ts
ARIMA(0,1,1)(1,1,1)[4]


Coefficients:
          ma1     sar1      sma1
      -0.5077   0.0244   -0.8676
s.e.    0.1287  0.1610   0.1399


sigma^2 = 66453:  log likelihood = -440.54
AIC=889.08   AICc=889.77   BIC=897.65


Training set error measures:
                   ME      RMSE       MAE       MPE      MAPE       MASE
ACF1
Training set 54.64954 242.1466 189.2346 1.058719 4.080805 0.7523571 -0.0229
1503
```


#o Create a plot to compare the fit to the training data. Do not show the validation data.

Hide

```
#comparing fit to the training data
main.ARIMA.pred <- forecast(main.ARIMA, h = 12)
plot(train.ts, ylab = "Victoria", xlab = "Time", bty = "l",
```

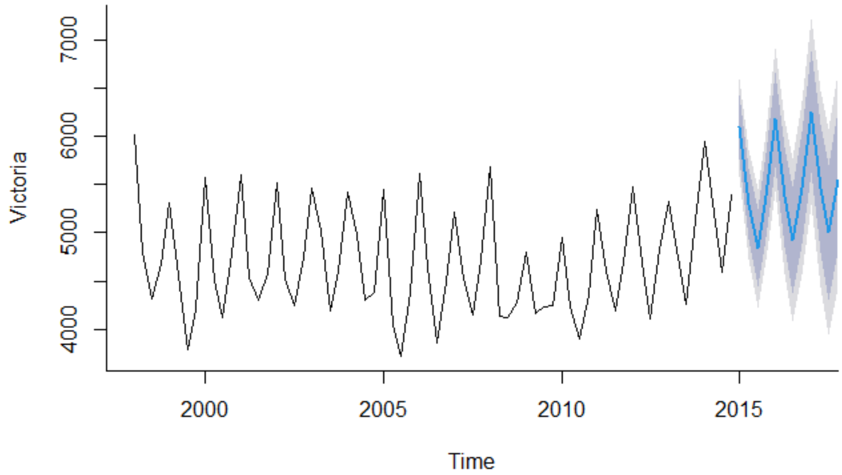
```
      xlim = c(1998, 2017), main = "")
lines(main.ARIMA.pred$fitted, lwd = 2, col = "blue", lty = 3)
```



o *Create a plot to show the forecast and prediction interval.*

Hide

```
#showing forecast and prediction interval
plot(main.ARIMA.pred, ylab = "Victoria", xlab = "Time", bty = "l", xlim = c
(1998, 2017), main = "")
```



Use auto.arima() to fit an ARIMA (p, d, q) (P, D, Q) model to the training data

Hide

```
#fitting auto ARIMA model
auto.ARIMA.fit <- auto.arima(train.ts)
summary(auto.ARIMA.fit)
```

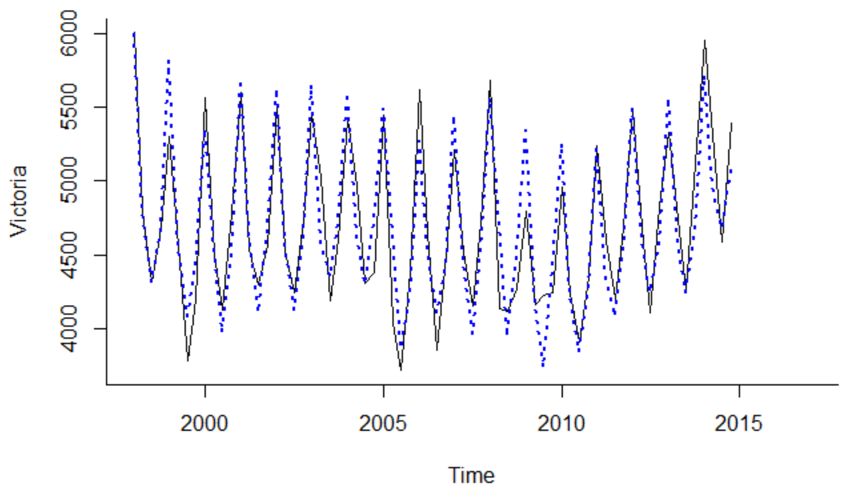
```
Series: train.ts
```

```
ARIMA(1,0,1)(0,1,1)[4]

Coefficients:
         ar1      ma1     sma1
      0.9330  -0.4702  -0.9188
s.e.  0.1165   0.1625   0.2401

sigma^2 = 62422:  log likelihood = -445.65
AIC=899.3   AICc=899.98   BIC=907.94

Training set error measures:
                   ME     RMSE      MAE         MPE     MAPE      MASE
ACF1
Training set 3.538936 236.6351 184.2156 -0.04522999 3.965776 0.7324024 -0.0
183292
```

o Create a plot to compare the fit to the training data. Do not show the validation data.

Hide

```
#comparing fit to the training data
plot(train.ts, ylab = "Victoria", xlab = "Time", bty = "l",
     xlim = c(1998, 2017), main = "")
lines(auto.ARIMA.fit$fitted, lwd = 2, col = "blue", lty = 3)
```

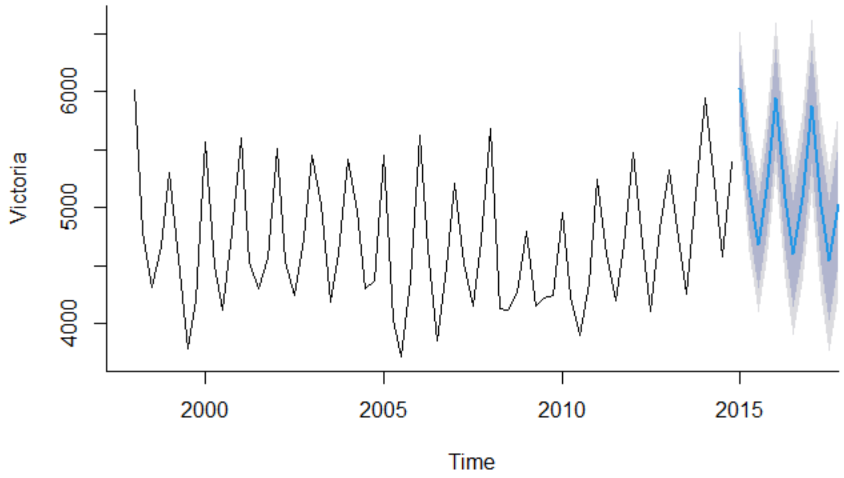


#o Create a plot to show the forecast and prediction interval.

Hide

```
#showing the forecast and prediction interval
auto.ARIMA.pred <- forecast(auto.ARIMA.fit, h = 12)
```

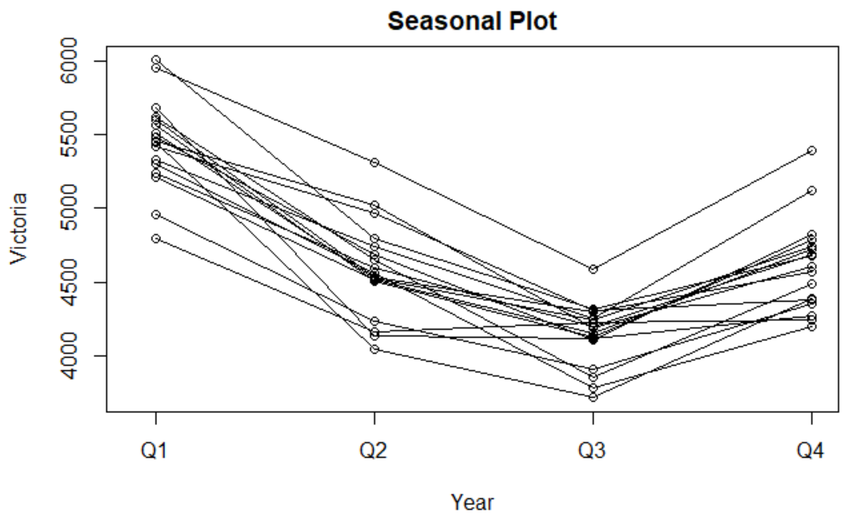
```
plot(auto.ARIMA.pred, ylab = "Victoria", xlab = "Time", bty = "l", xlim = c
(1998,2017), main = "")
```



Fit an exponential smoothing model o Looking at the training data:

Hide

```
#creating seasonal plot

seasonplot(train.ts, ylab = "Victoria", xlab = "Year", main = "Seasonal Plo
t")
```



♣ Is there a trend? What form does it take? #It seems that there is no trend in the model, if there is a trend then it will be additive or mulitiplicative. ♣ Is there seasonality? What form does it take? The season is weak but if there is a season it is additive, so additivie or no seaosn. ♣ What ETS() model would you recommend for this data? (If you are unsure then try several)

Hide

```
#ETS model with Additive error, multiplicative trend, additive season

ETS.model1 <- ets(train.ts, model = "AMA", restrict=FALSE)
```

```
ETS.model1
```

```
ETS(A,M,A) 

Call:
 ets(y = train.ts, model = "AMA", restrict = FALSE) 

  Smoothing parameters:
    alpha = 0.4936 
    beta  = 1e-04 
    gamma = 2e-04 

  Initial states:
    l = 5064.5235 
    b = 1.0007 
    s = -78.9853 -552.6534 -107.534 739.1727

  sigma:  247.8353

     AIC     AICc      BIC 
1046.151 1049.255 1066.127 
```

Hide

```
#ETS model with Additive error, multiplicative trend, no season
ETS.model2 <- ets(train.ts, model = "AMN", restrict=FALSE)
ETS.model2
```

```
ETS(A,Md,N) 

Call:
 ets(y = train.ts, model = "AMN", restrict = FALSE) 

  Smoothing parameters:
    alpha = 0.0048 
    beta  = 1e-04 
    phi   = 0.8 

  Initial states:
    l = 5180.5844 
```

```
    b = 0.9753

  sigma:  568.9684

     AIC     AICc      BIC
1156.493 1157.870 1169.810
```

Hide

```
#ETS model with Additive error, additive trend, additive season
ETS.model3 <- ets(train.ts, model = "AAA", restrict=FALSE)
ETS.model3
```

```
ETS(A,A,A)

Call:
 ets(y = train.ts, model = "AAA", restrict = FALSE)

  Smoothing parameters:
    alpha = 0.4841
    beta  = 0.0161
    gamma = 1e-04

  Initial states:
    l = 5058.5413
    b = -12.6284
    s = -96.3167 -546.6731 -100.1986 743.1884

  sigma:  249.4617

     AIC     AICc      BIC
1047.041 1050.144 1067.017
```

Hide

```
#ETS model with Additive error, additive trend, no season
ETS.model4 <- ets(train.ts, model = "AAN", restrict=FALSE)
ETS.model4
```

```
ETS(A,Ad,N)

Call:
```

```
 ets(y = train.ts, model = "AAN", restrict = FALSE)

  Smoothing parameters:
    alpha = 1e-04
    beta  = 1e-04
    phi   = 0.9307

  Initial states:
    l = 5176.8039
    b = -39.6601

  sigma:  580.1806

     AIC     AICc      BIC
1159.147 1160.524 1172.464
```

Hide

```
#ETS model with Multiplicative error, multiplicative trend, additive season
ETS.model5 <- ets(train.ts, model = "MMA", restrict=FALSE)
ETS.model5
```

```
ETS(M,Md,A)

Call:
 ets(y = train.ts, model = "MMA", restrict = FALSE)

  Smoothing parameters:
    alpha = 0.3916
    beta  = 1e-04
    gamma = 0.0065
    phi   = 0.8

  Initial states:
    l = 5067.8059
    b = 0.9655
    s = -68.9169 -509.3208 -96.4794 674.717

  sigma:  0.0538
```

```
     AIC     AICc      BIC 
1048.062 1051.922 1070.257 
```

Hide

```
#ETS model with Multiplicative error, multiplicative trend, no season
ETS.model6 <- ets(train.ts, model = "MMN", restrict=FALSE)
ETS.model6
```

```
ETS(M,Md,N) 

Call:
 ets(y = train.ts, model = "MMN", restrict = FALSE) 

  Smoothing parameters:
    alpha = 1e-04 
    beta  = 1e-04 
    phi   = 0.9254 

  Initial states:
    l = 5178.2731 
    b = 0.9913 

  sigma:  0.1218

     AIC     AICc      BIC 
1158.362 1159.739 1171.679 
```

Hide

```
#ETS model with Multiplicative error, additive trend, additive season
ETS.model7 <- ets(train.ts, model = "MAA", restrict=FALSE)
ETS.model7
```

```
ETS(M,A,A) 

Call:
 ets(y = train.ts, model = "MAA", restrict = FALSE) 

  Smoothing parameters:
    alpha = 0.4528 
```

```
    beta  = 1e-04
    gamma = 0.0071

  Initial states:
    l = 5030.3256
    b = 4.469
    s = -61.3191 -517.6836 -109.4666 688.4693

  sigma:  0.0541

     AIC     AICc      BIC
1048.992 1052.096 1068.968
```

Hide

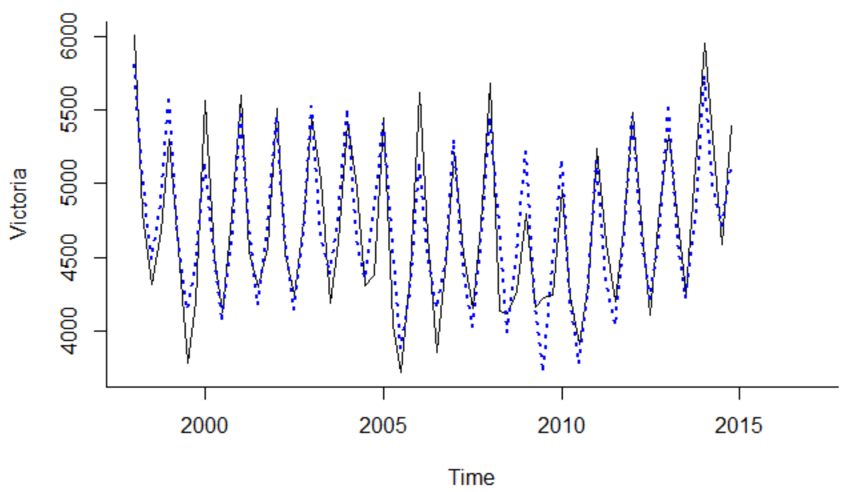
```
#ETS model with Multiplicative error, additive trend, no season
ETS.model8 <- ets(train.ts, model = "MAN", restrict=FALSE)
ETS.model8
```

```
ETS(M,Ad,N)

Call:
 ets(y = train.ts, model = "MAN", restrict = FALSE)

  Smoothing parameters:
    alpha = 0.0358
    beta  = 0.0358
    phi   = 0.8092

  Initial states:
    l = 5177.6354
    b = -18.6573

  sigma:  0.125

     AIC     AICc      BIC
1160.488 1161.865 1173.805
```

The AMA model has lowest value of AIC so that model is best one.

Hide

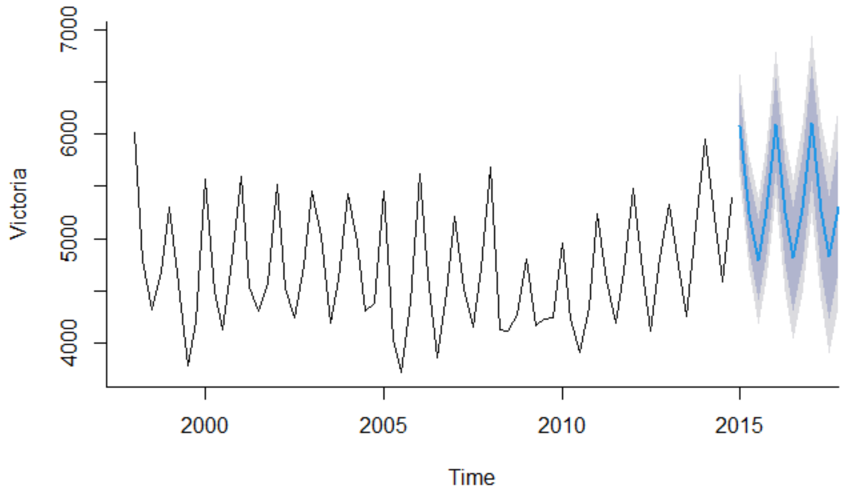
```
#comparing predition to the training data
plot(train.ts, ylab = "Victoria", xlab = "Time", bty = "l",
     xlim = c(1998, 2017), main = "")
lines(ETS.model1$fitted, lwd = 2, col = "blue", lty = 3)
```



#o Create a plot to show the forecast and prediction interval.

Hide

```
#showing forecast and prediction interval
ETS.model1.pred <- forecast(ETS.model1, h = 12)
plot(ETS.model1.pred, ylab = "Victoria", xlab = "Time", bty = "l", xlim = c
(1998,2017), main = "")
```



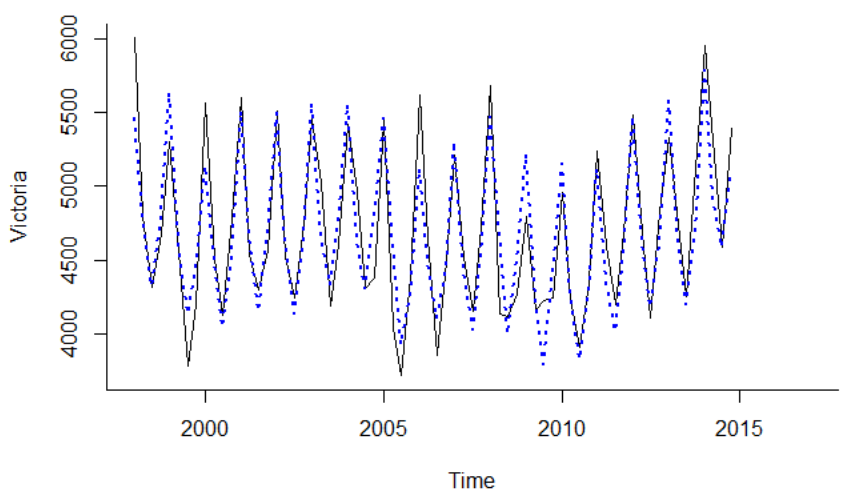
Hide

```
#auto ETS model
ETS.alg.model <- ets(train.ts, restrict = FALSE)
ETS.alg.model
ETS(M,N,M)


Call:
 ets(y = train.ts, restrict = FALSE)


  Smoothing parameters:
    alpha = 0.4366
    gamma = 1e-04


  Initial states:
    l = 4700.6455
    s = 0.9794 0.8806 0.9777 1.1624


  sigma:  0.0526


     AIC     AICc      BIC
1042.427 1044.293 1057.963
```

Hide

```
#creating plot to compare prediction to the training data
plot(train.ts, ylab = "Victoria", xlab = "Time", bty = "l",
     xlim = c(1998, 2017), main = "")
```

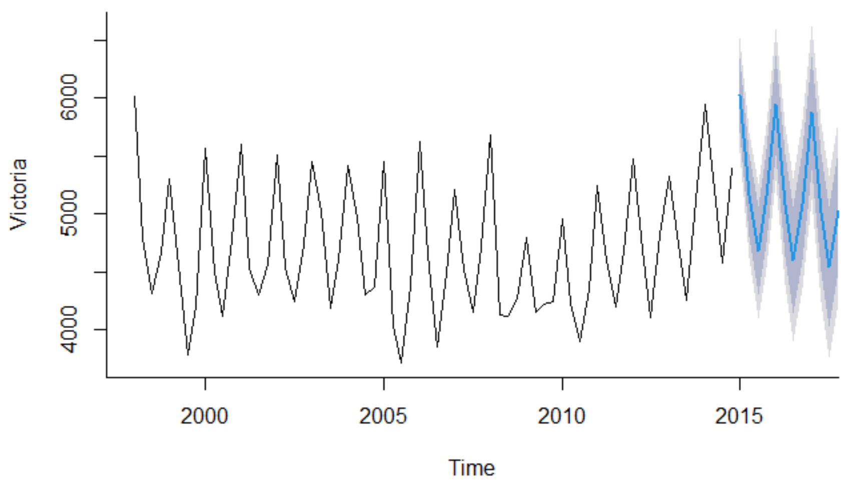
```
lines(ETS.alg.model$fitted, lwd = 2, col = "blue", lty = 3)
```



#o Create a plot to show the forecast and prediction interval.

Hide

```
#creating a plot to show the forecast and prediction interval
ETS.alg.mod.pred <- forecast(auto.ARIMA.fit, h = 12)
plot(ETS.alg.mod.pred, ylab = "Victoria", xlab = "Time", bty = "l", xlim =
c(1998,2017), main = "")
```



Hide

```
#checking accuracy of all models on both train and test data
options(scipen = 999)
print("Accuracy for Linear Model")
[1] "Accuracy for Linear Model"
```

Hide

```
accuracy(train.lm.pred, test.ts)
```

```
                                          ME       RMSE        MAE         MPE
MAPE      MASE      ACF1 Theil's U
Training set   -0.0000000000002676944  286.3821  215.5395 -0.3678315  4.61
8621 0.8569399 0.5134102        NA
Test set     1167.7638634681366056611 3 1260.8883 1167.7639 19.3354736 19.33
5474 4.6427838 0.5065477  1.519638
```

Hide

```
print("Accuracy for main Arima Model")
```

```
[1] "Accuracy for main Arima Model"
```

Hide

```
accuracy(main.ARIMA.pred, test.ts)
```

```
                    ME      RMSE        MAE       MPE      MAPE       MASE
ACF1 Theil's U
Training set  54.64954 242.1466 189.2346 1.058719 4.080805 0.7523571 -0.022
91503        NA
Test set     440.42417 606.2667 449.3299 6.949725 7.117039 1.7864412  0.434
84906 0.7322113
```

Hide

```
print("Accuracy for auto Arima Model")
```

```
[1] "Accuracy for auto Arima Model"
```

Hide

```
accuracy(auto.ARIMA.pred, test.ts)
```

```
                     ME      RMSE        MAE           MPE        MAPE       MASE
ACF1 Theil's U
Training set   3.538936 236.6351 184.2156 -0.04522999  3.965776 0.7324024 -
0.0183292        NA
Test set     739.862432 910.9337 739.8624 11.95974713 11.959747 2.9415376
0.5479558   1.09991
```

Hide

```
print("Accuracy fro main ETS model")
```

```
[1] "Accuracy fro main ETS model"
```

Hide

```
accuracy(ETS.model1.pred, test.ts)
```

```
                     ME      RMSE        MAE          MPE       MAPE       MASE
ACF1 Theil's U
Training set   1.916735 232.8007 187.2007 -0.1844728 4.036090 0.7442706 0.0
09791523        NA
```

```
Test set     561.554086 730.9644 561.5541  8.9685387 8.968539 2.2326211 0.4
96746408 0.8831839
```

Hide

```
print("Accuracy for auto ETS model")
```

```
[1] "Accuracy for auto ETS model"
```

Hide

```
accuracy(ETS.alg.mod.pred, test.ts)
```

```
                      ME      RMSE       MAE          MPE        MAPE        MASE
ACF1 Theil's U

Training set   3.538936 236.6351 184.2156 -0.04522999  3.965776 0.7324024 -
0.0183292        NA

Test set     739.862432 910.9337 739.8624 11.95974713 11.959747 2.9415376
0.5479558   1.09991
```

The main ARIMA model has lowest mean square error and mean absolute error for test data which means this model performs better as compare to other models.

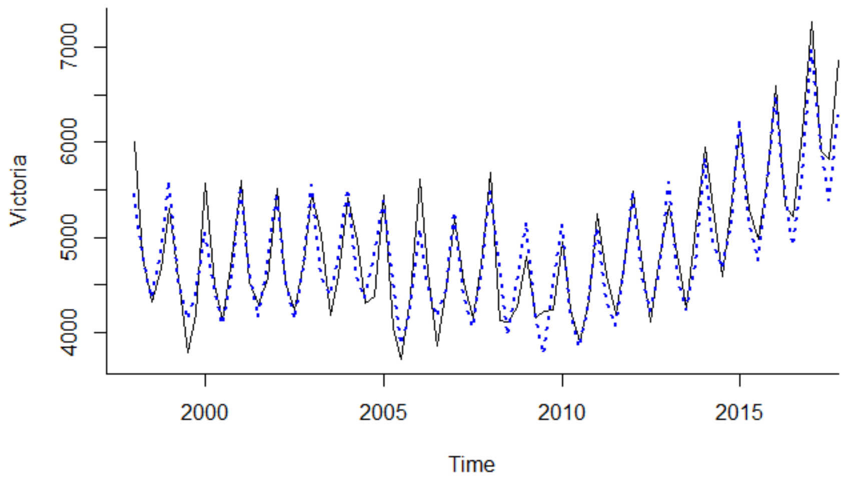
Hide

```
#fitting final model on whole data
ETS.alg.model <- ets(data.ts, model = "MNM")
ETS.alg.model
```

```
ETS(M,N,M)


Call:
 ets(y = data.ts, model = "MNM")


  Smoothing parameters:
    alpha = 0.5607
    gamma = 0.0001


  Initial states:
    l = 4699.2274
    s = 0.9847 0.8828 0.9713 1.1612


  sigma:  0.0529


     AIC     AICc      BIC
1244.010 1245.565 1260.684
```

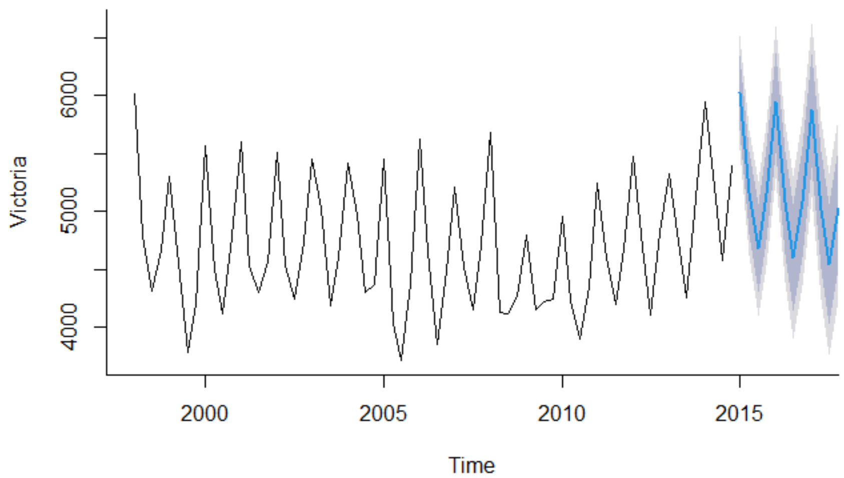
Hide

```
#plotting fit of recommended model to the training and validation data
plot(data.ts, ylab = "Victoria", xlab = "Time", bty = "l",
     xlim = c(1998, 2017), main = "")
lines(ETS.alg.model$fitted, lwd = 2, col = "blue", lty = 3)
```



Hide

```
#plotting prediction and confidence interval for final model
ETS.alg.mod.pred <- forecast(auto.ARIMA.fit, h = 12)
plot(ETS.alg.mod.pred, ylab = "Victoria", xlab = "Time", bty = "l", xlim =
c(1998,2017), main = "")
```



..............................End of Main Submission ....

Extra- Work

Exploratory Analysis
Trying different Libraries

Extra- Work

Exploratory Analysis
Trying different Libraries

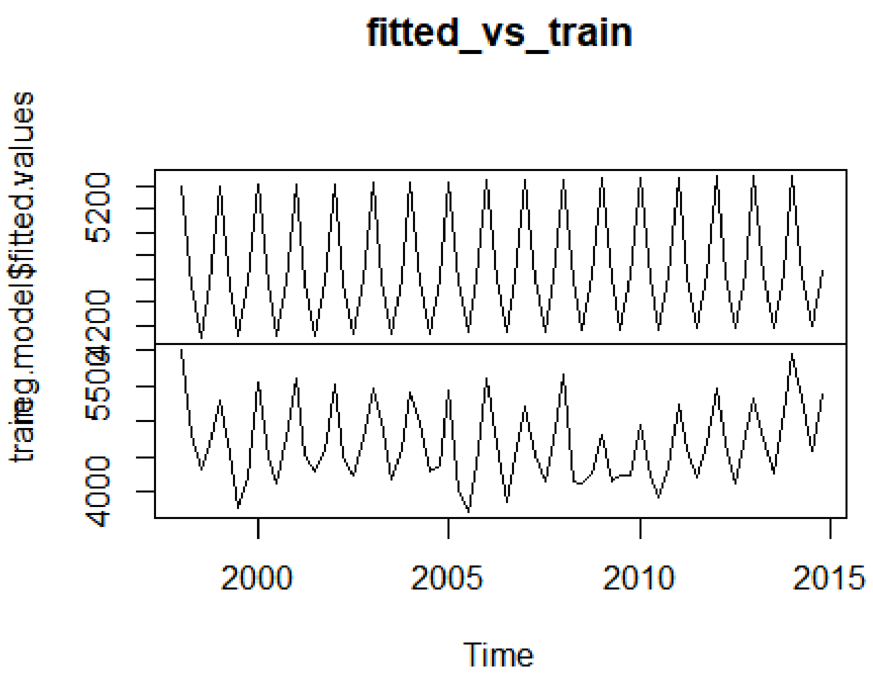
```
data<-data[,8]
numrows<-80
Date<-seq(as.Date("1998-01-01"), by="quarter", length.out = numrows)
ts<-as.xts(data,order.by = Date)
#Change to time series object
data.ts <- ts(data, start=c(1998, 1),frequency=4)
#Split into train and test sets
train<-window(data.ts,1998,c(2014,4))
test<-window(data.ts,2015,c(2017,4))
```

1

For the logistic regression model the training time series will be the dependent variable while its trend and seasonality are the predictor variables.

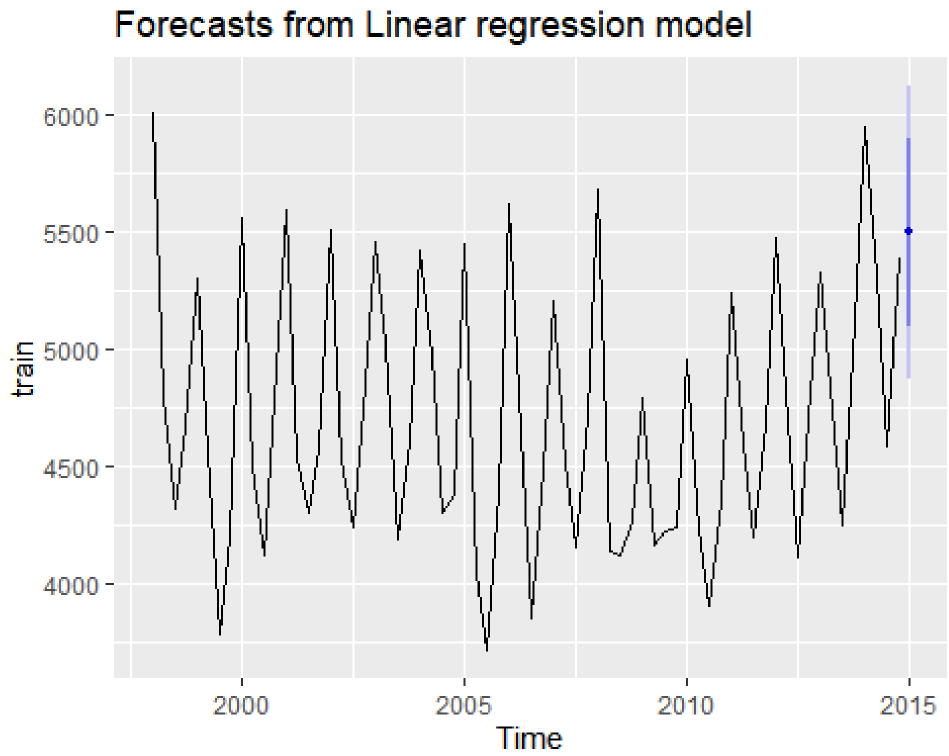
```
#Fit regression model
library(forecast)
reg.model<- tslm(train~trend+season)
```

```
fitted_vs_train<-cbind(reg.model$fitted.values,train)
plot(fitted_vs_train)
```



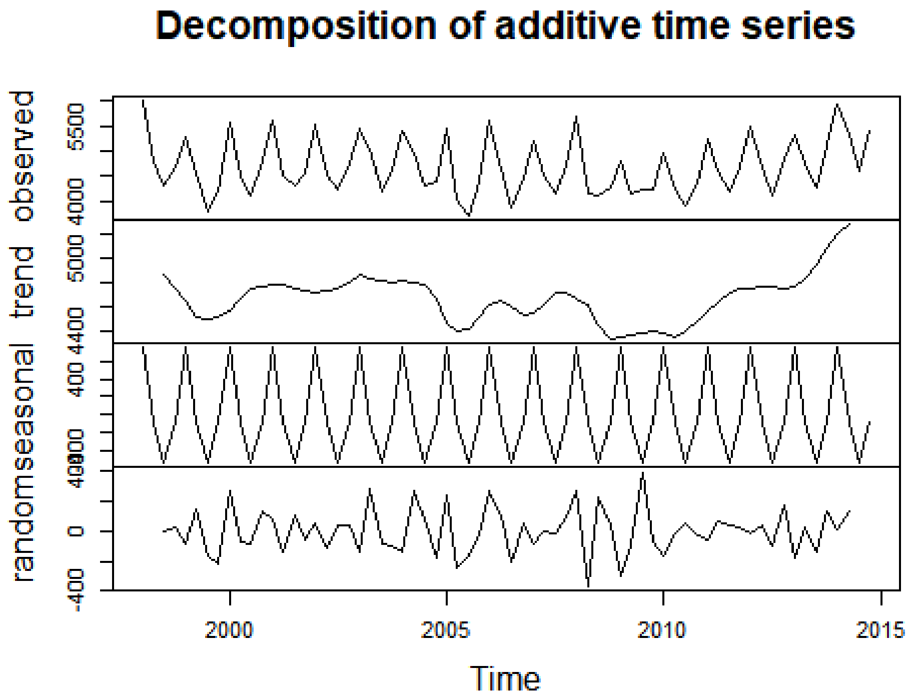
The fitted values have a more regular seasonal pattern when compared to the the actual values which have a irregular pattern

```
##Create a plot to show the forecast and prediction interval
pred.reg<-forecast::forecast(reg.model,36);pred.reg

##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2015 Q1       5498.451 5092.752 5904.151 4872.474 6124.429

autoplot(pred.reg)
```



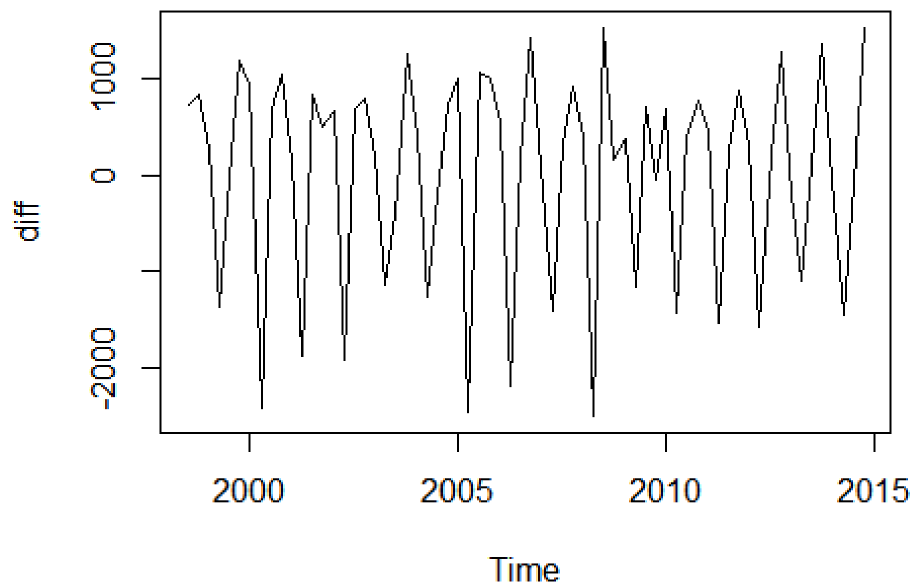
2

```
# decompose time series
decomp<-decompose(train)
plot(decomp)
```



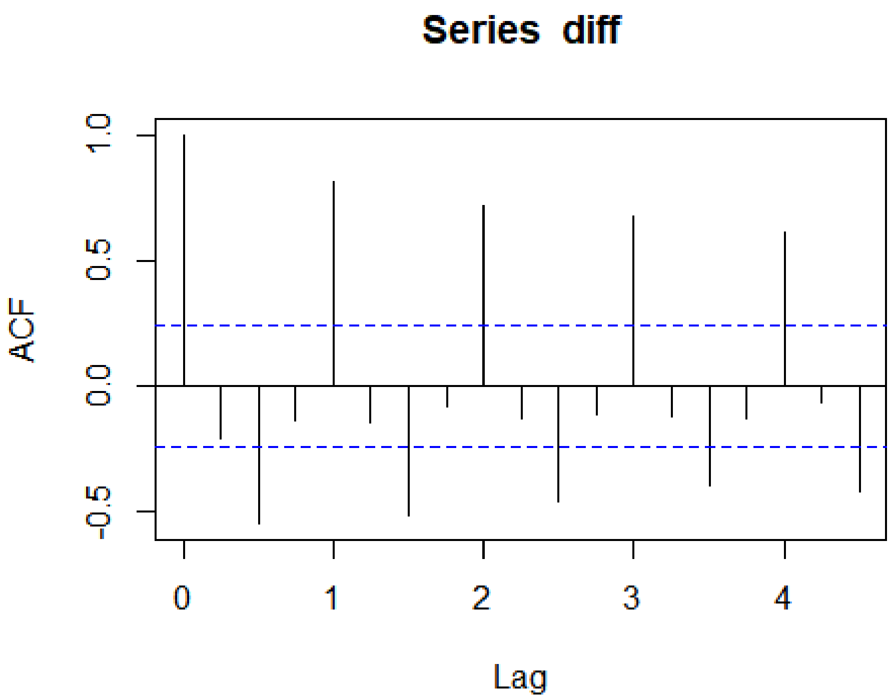
Based on the above plot 2nd order difference is required.Since it has a time varying trend, I need second order differencing.

```
diff<-diff(train,differences=2)
plot(diff)
```

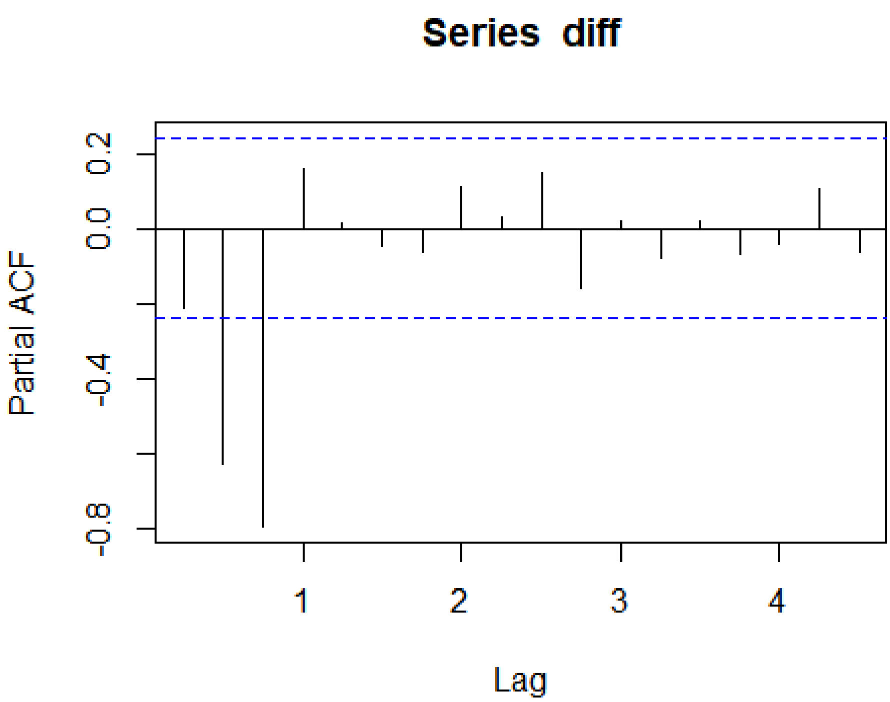


After differencing the time series has become more stationary. We can now proceeed to plot the acf and pacf.

```
## ACF and PACF plot
acf(diff,type="correlation")
```



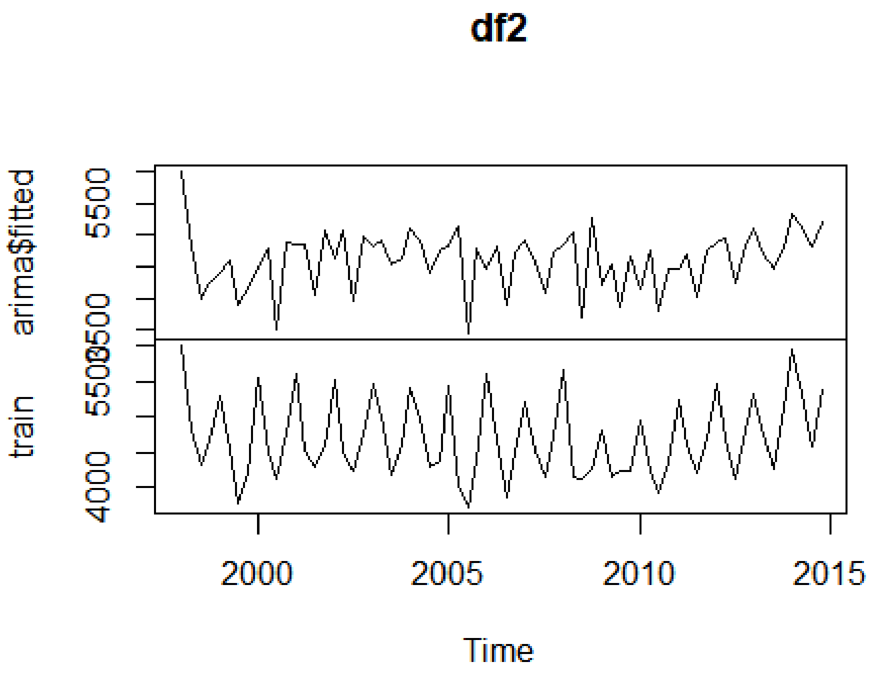

```
acf(diff,type = "partial")
```



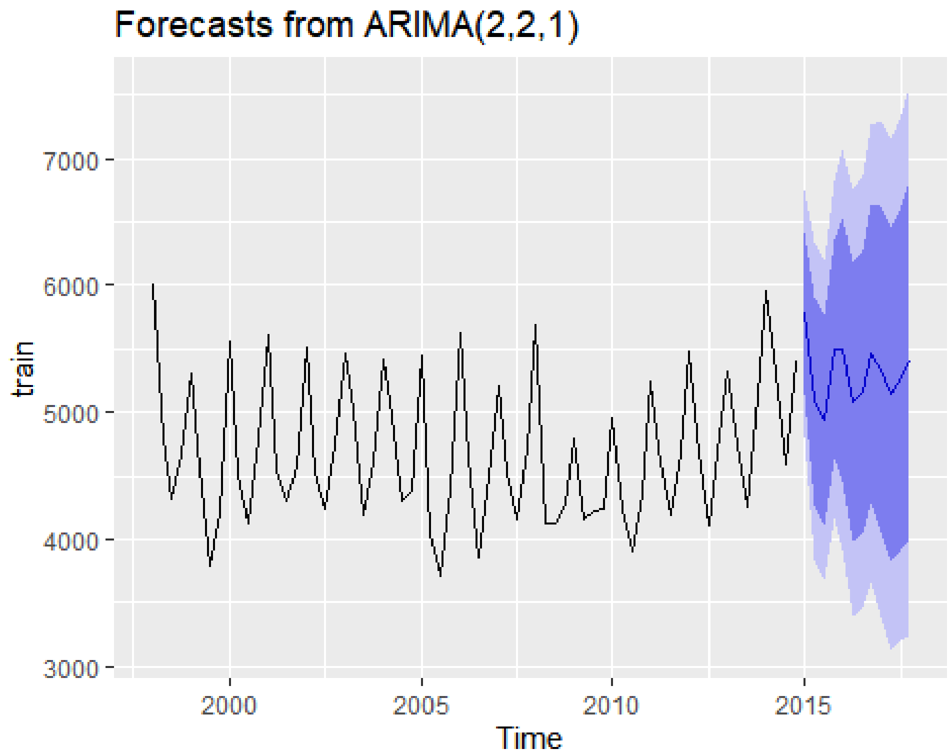
Note that the PACF plot has a significant spike only at lag 1, meaning that all the higher-order autocorrelations seen in the ACF plot are effectively explained by the lag-1 autocorrelation.

The PACFc sharply cuts off , hence the needed rwo or more AR terms are required. On the other hand, MA terms is pilled from the ACF plot

```
# Fit recommended ARIMA model
arima<-Arima(train,order =c(2,2,1),method ="ML" )
df2<-cbind(arima$fitted,train)
plot(df2)
```

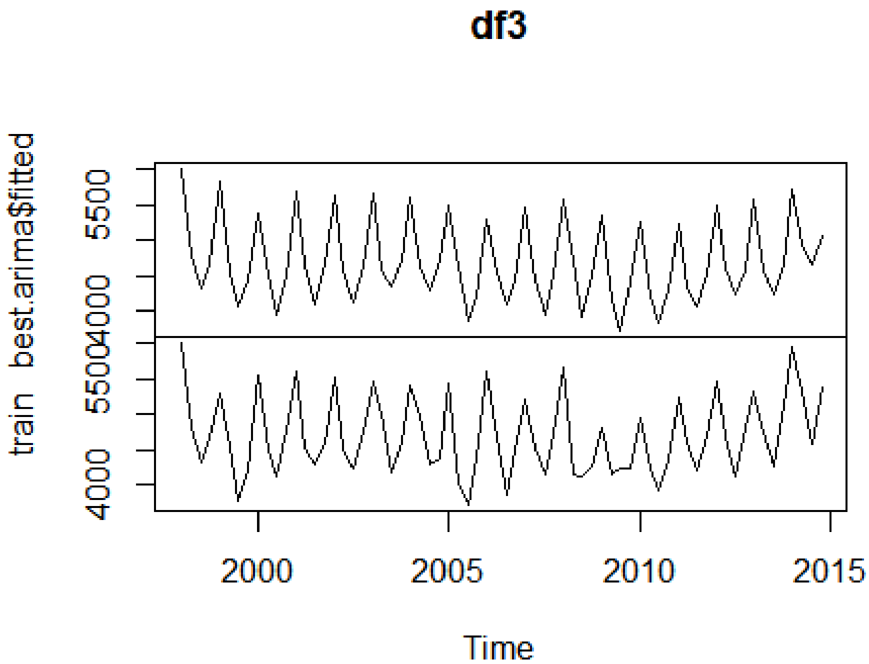



```
pred.arima<-forecast::forecast(arima,12)
autoplot(pred.arima)
```

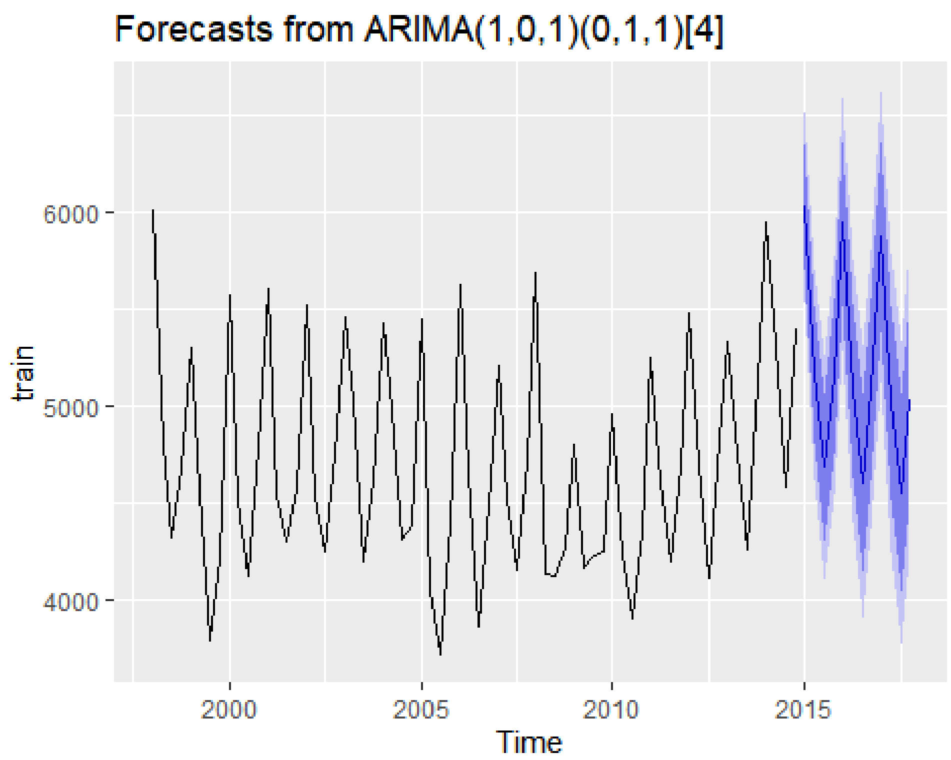


3

```
##Choose Best Arima model
best.arima<-auto.arima(train)
best.arima

## Series: train
## ARIMA(1,0,1)(0,1,1)[4]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.9330  -0.4702  -0.9188
## s.e.  0.1165   0.1625   0.2401
##
## sigma^2 = 62422:  log likelihood = -445.65
## AIC=899.3   AICc=899.98   BIC=907.94

#Compare training and fitted values
df3<-cbind(best.arima$fitted,train)
plot(df3)
```

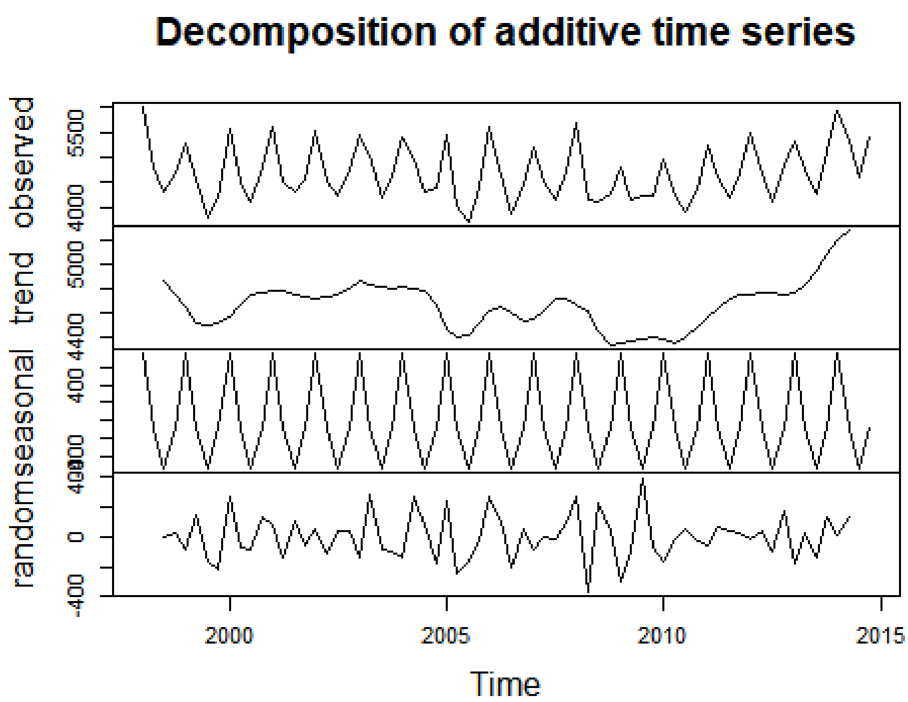



```r
## Create a plot to show the forecast and prediction interval.
pred.arima2<-forecast::forecast(best.arima,12);pred.arima2

##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2015 Q1       6029.453 5707.979 6350.926 5537.802 6521.104
## 2015 Q2       5162.238 4808.248 5516.229 4620.856 5703.620
## 2015 Q3       4679.701 4299.759 5059.643 4098.630 5260.772
## 2015 Q4       5151.339 4750.275 5552.402 4537.965 5764.712
## 2016 Q1       5941.296 5513.679 6368.914 5287.311 6595.281
## 2016 Q2       5079.988 4634.546 5525.429 4398.743 5761.232
## 2016 Q3       4602.961 4142.644 5063.277 3898.967 5306.954
## 2016 Q4       5079.739 4606.944 5552.535 4356.661 5802.818
## 2017 Q1       5874.494 5384.424 6364.563 5124.997 6623.990
## 2017 Q2       5017.660 4516.063 5519.258 4250.533 5784.788
## 2017 Q3       4544.809 4033.459 5056.159 3762.766 5326.852
## 2017 Q4       5025.484 4505.874 5545.094 4230.809 5820.159

autoplot(pred.arima2)
```



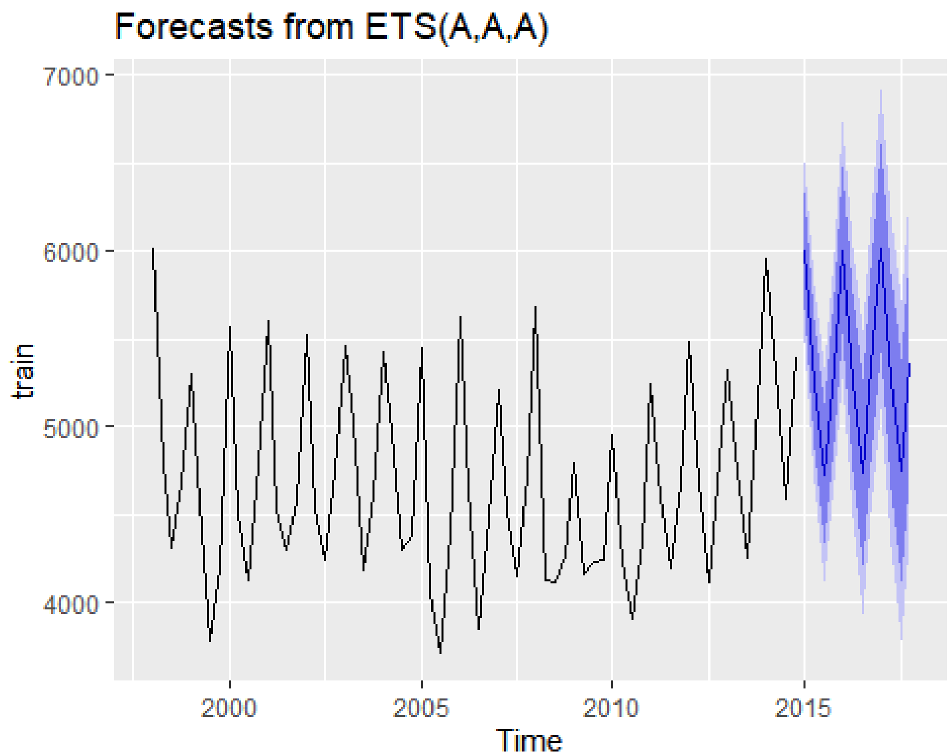
4

```
# decompose training data
plot(decompose(train))
```



First, there is an irregular trend over the given period. However, the seasonality has regular seasonality.

```r
exp.smooth<-ets(train,model="AAA",gamma=0.21)
pred.ets<-forecast::forecast(exp.smooth,12);autoplot(pred.ets)
```

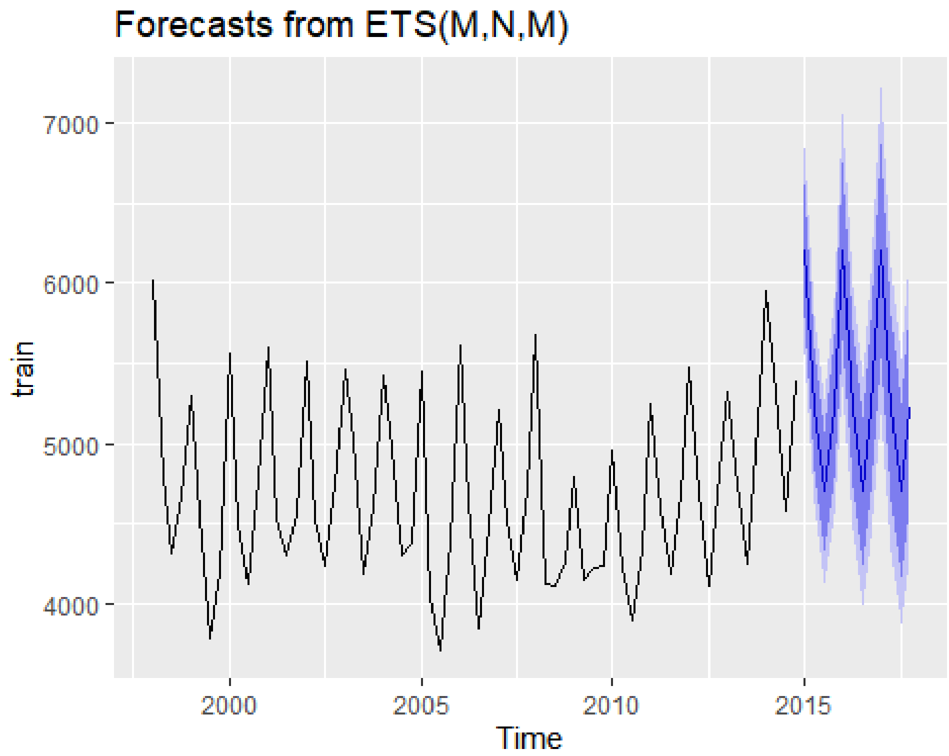


5

```r
best.ets<-ets(train);best.ets
```

```
## ETS(M,N,M)
##
## Call:
##  ets(y = train)
##
##   Smoothing parameters:
##     alpha = 0.4366
##     gamma = 1e-04
##
##   Initial states:
##     l = 4700.6455
##     s = 0.9794 0.8806 0.9777 1.1624
##
##   sigma:  0.0526
##
##      AIC     AICc      BIC
## 1042.427 1044.293 1057.963
```

```r
pred.ets2<-forecast::forecast(best.ets,12);
autoplot(pred.ets2)
```



6

```
library(forecast)
forecast::accuracy(pred.reg,test)# model 1

##                          ME     RMSE      MAE        MPE      MAPE
MASE
## Training set -2.676944e-14 286.3821 215.5395 -0.3678315  4.618621 0.856
9399
## Test set      6.284843e+02 628.4843 628.4843 10.2577260 10.257726 2.498
7215
##                   ACF1
## Training set 0.5134102
## Test set            NA

forecast::accuracy(pred.arima,test)#model 2

##                     ME     RMSE      MAE      MPE     MAPE     MASE
ACF1
## Training set  92.06684 474.8423 362.8071 1.336435 7.608786 1.442445 -0.
5383448
## Test set     616.99535 846.6619 616.9953 9.546519 9.546519 2.453044  0.
1437700
##              Theil's U
## Training set        NA
## Test set     0.9976302

forecast::accuracy(pred.arima2,test)#model 3
```

```
##                         ME      RMSE       MAE          MPE       MAPE       MA
SE
## Training set   3.538936 236.6351 184.2156 -0.04522999  3.965776 0.73240
24
## Test set     739.862432 910.9337 739.8624 11.95974713 11.959747 2.94153
76
##                     ACF1 Theil's U
## Training set -0.0183292        NA
## Test set      0.5479558   1.09991
```

```
forecast::accuracy(pred.ets,test)#model 4
```

```
##                        ME      RMSE       MAE          MPE      MAPE       MASE
## Training set   5.95897 244.2978 198.0688 -0.06026762 4.264937 0.7874801
## Test set     586.19277 748.0836 586.1928  9.38239891 9.382399 2.3305793
##                     ACF1 Theil's U
## Training set 0.01277295        NA
## Test set     0.46801866 0.9007058
```

```
forecast::accuracy(pred.ets2,test)#model 5
```

```
##                        ME      RMSE       MAE        MPE      MAPE       MASE
## Training set  21.70711 236.2571 185.1006 0.2320075 3.952035 0.7359211
## Test set     582.66778 759.6801 595.9466 9.4160559 9.632784 2.3693583
##                     ACF1 Theil's U
## Training set 0.05315406        NA
## Test set     0.53824889 0.9206154
```

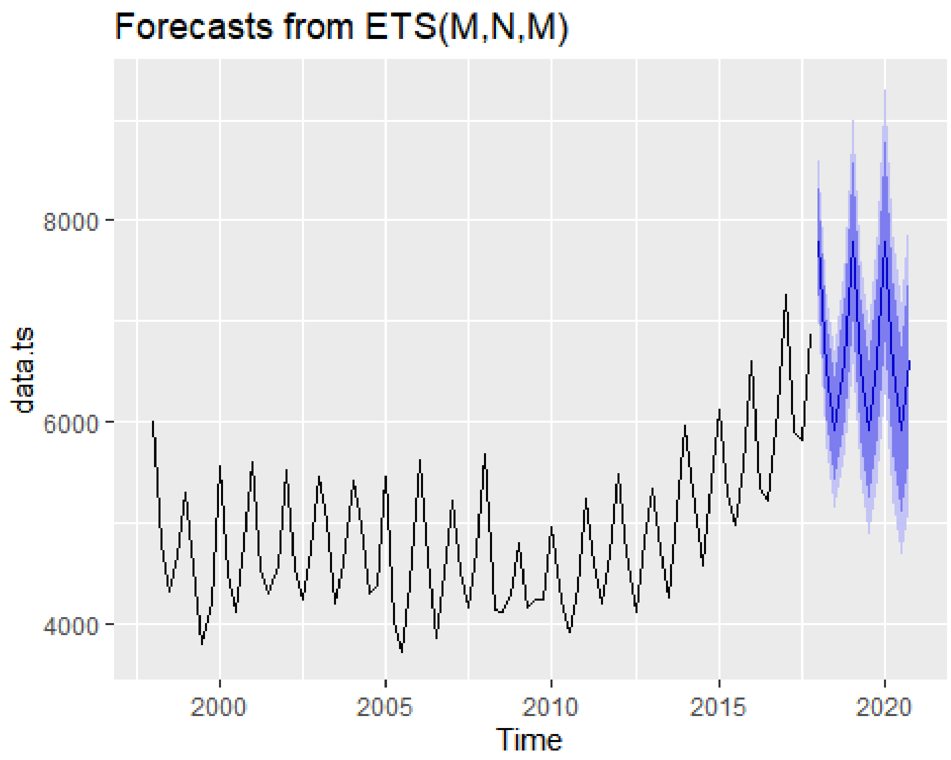
Model 5 has the highest prediction accuracy thus it is the best model to be adopted by the Australian Tourism Board to forecast trips

7

```
##3 year Prediction based on the full data set
best.model<-ets(data.ts);best.model
```

```
## ETS(M,N,M)
##
## Call:
##  ets(y = data.ts)
##
##   Smoothing parameters:
##     alpha = 0.5607
##     gamma = 1e-04
##
##   Initial states:
##     l = 4699.2274
##     s = 0.9847 0.8828 0.9713 1.1612
##
##   sigma:  0.0529
##
##      AIC     AICc      BIC
## 1244.010 1245.565 1260.684
```

```r
prediction<-forecast::forecast(best.model,12)
autoplot(prediction)
```



8

In my opinion, the exponential smoothing model is the superior method. It can be used to conduct real life analysis and forecasts. Unlike the ARIMA models it can fit non-stationary data, hence its flexibility gives it an edge over the others. Additionally, they have varying weight assigning methods and exponential smoothing has the superior weight assignment procedure Assuming that ARIMA models are more flexible than exponential smoothing is a widespread misconception. Non-linear exponential smoothing models are not special examples of ARIMA models like their linear counterparts are. Conversely, many ARIMA models do not have analogues in the form of exponential smoothing. To be more specific, some ARIMA models are stationary while all ETS models are non-stationary. There are two unit roots in the ETS models that incorporate seasonality, non-damped trend, or both. In other words, they need two levels of differencing to make them stationary. In contrast, there is only one unit root in all other ETS models they need one level of differencing. The results shown above compare the predictive abilities of the two models in question using the same test data. Comparing the RMSE, MAPE, and MASE on the test set, the ETS model appears to be the somewhat more accurate model.

Conclusion

In order to foretell future data points, it is necessary to examine past data and draw conclusions based on those conclusions. These projections can have a significant impact on a organizations' ability to plan both in the near and distant future. Time itself is the independent variable in our model of time series forecasting. These time intervals range from one hour to a whole year and will appear in a variety of sequences and repetitions. The most appropriate ETS model can generate demand forecast for business/products over the year, and also identify a repeatable trend, which further indicates forecasting at relatively high accuracy is possible.

End of Extra- Work
Exploratory Analysis
Trying different Libraries