Group 10 TIME SERIES FORECASTS FOR TOURISM FROM UK to THAILAND

2022-12-08

```
library(forecast)
```

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

```
df <-read.csv("Thaitourism.csv", header= TRUE)
head(df)
```

```
##   region nationality year month tourists
## 1 Africa   AfrOthers 2010     1     6553
## 2 Africa   AfrOthers 2010     2     5618
## 3 Africa   AfrOthers 2010     3     6689
## 4 Africa   AfrOthers 2010     4     5210
## 5 Africa   AfrOthers 2010     5     4537
## 6 Africa   AfrOthers 2010     6     4683
```

```
summary(df)
```

```
##     region          nationality             year          month
##  Length:4452        Length:4452        Min.   :2010   Min.   : 1.00
##  Class :character   Class :character   1st Qu.:2011   1st Qu.: 3.75
##  Mode  :character   Mode  :character   Median :2013   Median : 6.50
##                                        Mean   :2013   Mean   : 6.50
##                                        3rd Qu.:2015   3rd Qu.: 9.25
##                                        Max.   :2016   Max.   :12.00
##     tourists
##  Min.   :   104
##  1st Qu.:  5500
##  Median : 14216
##  Mean   : 38545
##  3rd Qu.: 49871
##  Max.   :958204
```

```
df1<-subset(df,nationality=="UnitedKingdom")
df1
```

```
##      region   nationality year month tourists
## 3109 Europe UnitedKingdom 2010     1    80652
## 3110 Europe UnitedKingdom 2010     2    82493
## 3111 Europe UnitedKingdom 2010     3    84157
## 3112 Europe UnitedKingdom 2010     4    70817
## 3113 Europe UnitedKingdom 2010     5    43303
## 3114 Europe UnitedKingdom 2010     6    52456
## 3115 Europe UnitedKingdom 2010     7    62368
## 3116 Europe UnitedKingdom 2010     8    63774
## 3117 Europe UnitedKingdom 2010     9    51644
## 3118 Europe UnitedKingdom 2010    10    63471
## 3119 Europe UnitedKingdom 2010    11    70259
## 3120 Europe UnitedKingdom 2010    12    85333
## 3121 Europe UnitedKingdom 2011     1    75932
## 3122 Europe UnitedKingdom 2011     2    73223
## 3123 Europe UnitedKingdom 2011     3    78390
## 3124 Europe UnitedKingdom 2011     4    87716
## 3125 Europe UnitedKingdom 2011     5    53726
## 3126 Europe UnitedKingdom 2011     6    58510
## 3127 Europe UnitedKingdom 2011     7    69171
## 3128 Europe UnitedKingdom 2011     8    73031
## 3129 Europe UnitedKingdom 2011     9    55816
## 3130 Europe UnitedKingdom 2011    10    62691
## 3131 Europe UnitedKingdom 2011    11    66270
## 3132 Europe UnitedKingdom 2011    12    90496
## 3133 Europe UnitedKingdom 2012     1    78220
## 3134 Europe UnitedKingdom 2012     2    78265
## 3135 Europe UnitedKingdom 2012     3    83652
## 3136 Europe UnitedKingdom 2012     4    77039
## 3137 Europe UnitedKingdom 2012     5    55810
## 3138 Europe UnitedKingdom 2012     6    61690
## 3139 Europe UnitedKingdom 2012     7    67456
## 3140 Europe UnitedKingdom 2012     8    68004
## 3141 Europe UnitedKingdom 2012     9    57129
## 3142 Europe UnitedKingdom 2012    10    68775
## 3143 Europe UnitedKingdom 2012    11    79707
## 3144 Europe UnitedKingdom 2012    12    97306
## 3145 Europe UnitedKingdom 2013     1    79664
## 3146 Europe UnitedKingdom 2013     2    80177
## 3147 Europe UnitedKingdom 2013     3    91649
## 3148 Europe UnitedKingdom 2013     4    71406
## 3149 Europe UnitedKingdom 2013     5    58069
## 3150 Europe UnitedKingdom 2013     6    62823
## 3151 Europe UnitedKingdom 2013     7    75482
## 3152 Europe UnitedKingdom 2013     8    71713
## 3153 Europe UnitedKingdom 2013     9    56834
## 3154 Europe UnitedKingdom 2013    10    71852
## 3155 Europe UnitedKingdom 2013    11    84826
## 3156 Europe UnitedKingdom 2013    12   100529
## 3157 Europe UnitedKingdom 2014     1    85771
## 3158 Europe UnitedKingdom 2014     2    77526
## 3159 Europe UnitedKingdom 2014     3    78029
```

```
## 3160 Europe UnitedKingdom 2014     4    89495
## 3161 Europe UnitedKingdom 2014     5    59930
## 3162 Europe UnitedKingdom 2014     6    59569
## 3163 Europe UnitedKingdom 2014     7    74764
## 3164 Europe UnitedKingdom 2014     8    70530
## 3165 Europe UnitedKingdom 2014     9    55882
## 3166 Europe UnitedKingdom 2014    10    71137
## 3167 Europe UnitedKingdom 2014    11    80446
## 3168 Europe UnitedKingdom 2014    12   104798
## 3169 Europe UnitedKingdom 2015     1    85023
## 3170 Europe UnitedKingdom 2015     2    81016
## 3171 Europe UnitedKingdom 2015     3    86755
## 3172 Europe UnitedKingdom 2015     4    82892
## 3173 Europe UnitedKingdom 2015     5    62141
## 3174 Europe UnitedKingdom 2015     6    64806
## 3175 Europe UnitedKingdom 2015     7    80449
## 3176 Europe UnitedKingdom 2015     8    74094
## 3177 Europe UnitedKingdom 2015     9    73362
## 3178 Europe UnitedKingdom 2015    10    73362
## 3179 Europe UnitedKingdom 2015    11    86563
## 3180 Europe UnitedKingdom 2015    12   111689
## 3181 Europe UnitedKingdom 2016     1    94085
## 3182 Europe UnitedKingdom 2016     2    91788
## 3183 Europe UnitedKingdom 2016     3   102343
## 3184 Europe UnitedKingdom 2016     4    81455
## 3185 Europe UnitedKingdom 2016     5    64760
## 3186 Europe UnitedKingdom 2016     6    67701
## 3187 Europe UnitedKingdom 2016     7    86028
## 3188 Europe UnitedKingdom 2016     8    79525
## 3189 Europe UnitedKingdom 2016     9    62276
## 3190 Europe UnitedKingdom 2016    10    73696
## 3191 Europe UnitedKingdom 2016    11    86747
## 3192 Europe UnitedKingdom 2016    12   112982
```

```
df1_train<-subset(df1,year %in% c(2010,2011,2012,2013,2014,2015))
df1_train
```

```
##      region   nationality year month tourists
## 3109 Europe UnitedKingdom 2010     1    80652
## 3110 Europe UnitedKingdom 2010     2    82493
## 3111 Europe UnitedKingdom 2010     3    84157
## 3112 Europe UnitedKingdom 2010     4    70817
## 3113 Europe UnitedKingdom 2010     5    43303
## 3114 Europe UnitedKingdom 2010     6    52456
## 3115 Europe UnitedKingdom 2010     7    62368
## 3116 Europe UnitedKingdom 2010     8    63774
## 3117 Europe UnitedKingdom 2010     9    51644
## 3118 Europe UnitedKingdom 2010    10    63471
## 3119 Europe UnitedKingdom 2010    11    70259
## 3120 Europe UnitedKingdom 2010    12    85333
## 3121 Europe UnitedKingdom 2011     1    75932
## 3122 Europe UnitedKingdom 2011     2    73223
## 3123 Europe UnitedKingdom 2011     3    78390
## 3124 Europe UnitedKingdom 2011     4    87716
## 3125 Europe UnitedKingdom 2011     5    53726
## 3126 Europe UnitedKingdom 2011     6    58510
## 3127 Europe UnitedKingdom 2011     7    69171
## 3128 Europe UnitedKingdom 2011     8    73031
## 3129 Europe UnitedKingdom 2011     9    55816
## 3130 Europe UnitedKingdom 2011    10    62691
## 3131 Europe UnitedKingdom 2011    11    66270
## 3132 Europe UnitedKingdom 2011    12    90496
## 3133 Europe UnitedKingdom 2012     1    78220
## 3134 Europe UnitedKingdom 2012     2    78265
## 3135 Europe UnitedKingdom 2012     3    83652
## 3136 Europe UnitedKingdom 2012     4    77039
## 3137 Europe UnitedKingdom 2012     5    55810
## 3138 Europe UnitedKingdom 2012     6    61690
## 3139 Europe UnitedKingdom 2012     7    67456
## 3140 Europe UnitedKingdom 2012     8    68004
## 3141 Europe UnitedKingdom 2012     9    57129
## 3142 Europe UnitedKingdom 2012    10    68775
## 3143 Europe UnitedKingdom 2012    11    79707
## 3144 Europe UnitedKingdom 2012    12    97306
## 3145 Europe UnitedKingdom 2013     1    79664
## 3146 Europe UnitedKingdom 2013     2    80177
## 3147 Europe UnitedKingdom 2013     3    91649
## 3148 Europe UnitedKingdom 2013     4    71406
## 3149 Europe UnitedKingdom 2013     5    58069
## 3150 Europe UnitedKingdom 2013     6    62823
## 3151 Europe UnitedKingdom 2013     7    75482
## 3152 Europe UnitedKingdom 2013     8    71713
## 3153 Europe UnitedKingdom 2013     9    56834
## 3154 Europe UnitedKingdom 2013    10    71852
## 3155 Europe UnitedKingdom 2013    11    84826
## 3156 Europe UnitedKingdom 2013    12   100529
## 3157 Europe UnitedKingdom 2014     1    85771
## 3158 Europe UnitedKingdom 2014     2    77526
## 3159 Europe UnitedKingdom 2014     3    78029
```

```
## 3160 Europe UnitedKingdom 2014     4    89495
## 3161 Europe UnitedKingdom 2014     5    59930
## 3162 Europe UnitedKingdom 2014     6    59569
## 3163 Europe UnitedKingdom 2014     7    74764
## 3164 Europe UnitedKingdom 2014     8    70530
## 3165 Europe UnitedKingdom 2014     9    55882
## 3166 Europe UnitedKingdom 2014    10    71137
## 3167 Europe UnitedKingdom 2014    11    80446
## 3168 Europe UnitedKingdom 2014    12   104798
## 3169 Europe UnitedKingdom 2015     1    85023
## 3170 Europe UnitedKingdom 2015     2    81016
## 3171 Europe UnitedKingdom 2015     3    86755
## 3172 Europe UnitedKingdom 2015     4    82892
## 3173 Europe UnitedKingdom 2015     5    62141
## 3174 Europe UnitedKingdom 2015     6    64806
## 3175 Europe UnitedKingdom 2015     7    80449
## 3176 Europe UnitedKingdom 2015     8    74094
## 3177 Europe UnitedKingdom 2015     9    73362
## 3178 Europe UnitedKingdom 2015    10    73362
## 3179 Europe UnitedKingdom 2015    11    86563
## 3180 Europe UnitedKingdom 2015    12   111689
```

```
df1_test<-subset(df1,year == 2016)
df1_test
```

```
##      region   nationality year month tourists
## 3181 Europe UnitedKingdom 2016     1    94085
## 3182 Europe UnitedKingdom 2016     2    91788
## 3183 Europe UnitedKingdom 2016     3   102343
## 3184 Europe UnitedKingdom 2016     4    81455
## 3185 Europe UnitedKingdom 2016     5    64760
## 3186 Europe UnitedKingdom 2016     6    67701
## 3187 Europe UnitedKingdom 2016     7    86028
## 3188 Europe UnitedKingdom 2016     8    79525
## 3189 Europe UnitedKingdom 2016     9    62276
## 3190 Europe UnitedKingdom 2016    10    73696
## 3191 Europe UnitedKingdom 2016    11    86747
## 3192 Europe UnitedKingdom 2016    12   112982
```

```
df1_train.ts <-ts(df1_train[,3:5], frequency = 12, start = c(2010,1), end = c(2015,12))
head(df1_train.ts)
```

```
##          year month tourists
## Jan 2010 2010     1    80652
## Feb 2010 2010     2    82493
## Mar 2010 2010     3    84157
## Apr 2010 2010     4    70817
## May 2010 2010     5    43303
## Jun 2010 2010     6    52456
```

```
tail(df1_train.ts)
```

```
##          year month tourists
## Jul 2015 2015     7    80449
## Aug 2015 2015     8    74094
## Sep 2015 2015     9    73362
## Oct 2015 2015    10    73362
## Nov 2015 2015    11    86563
## Dec 2015 2015    12   111689
```

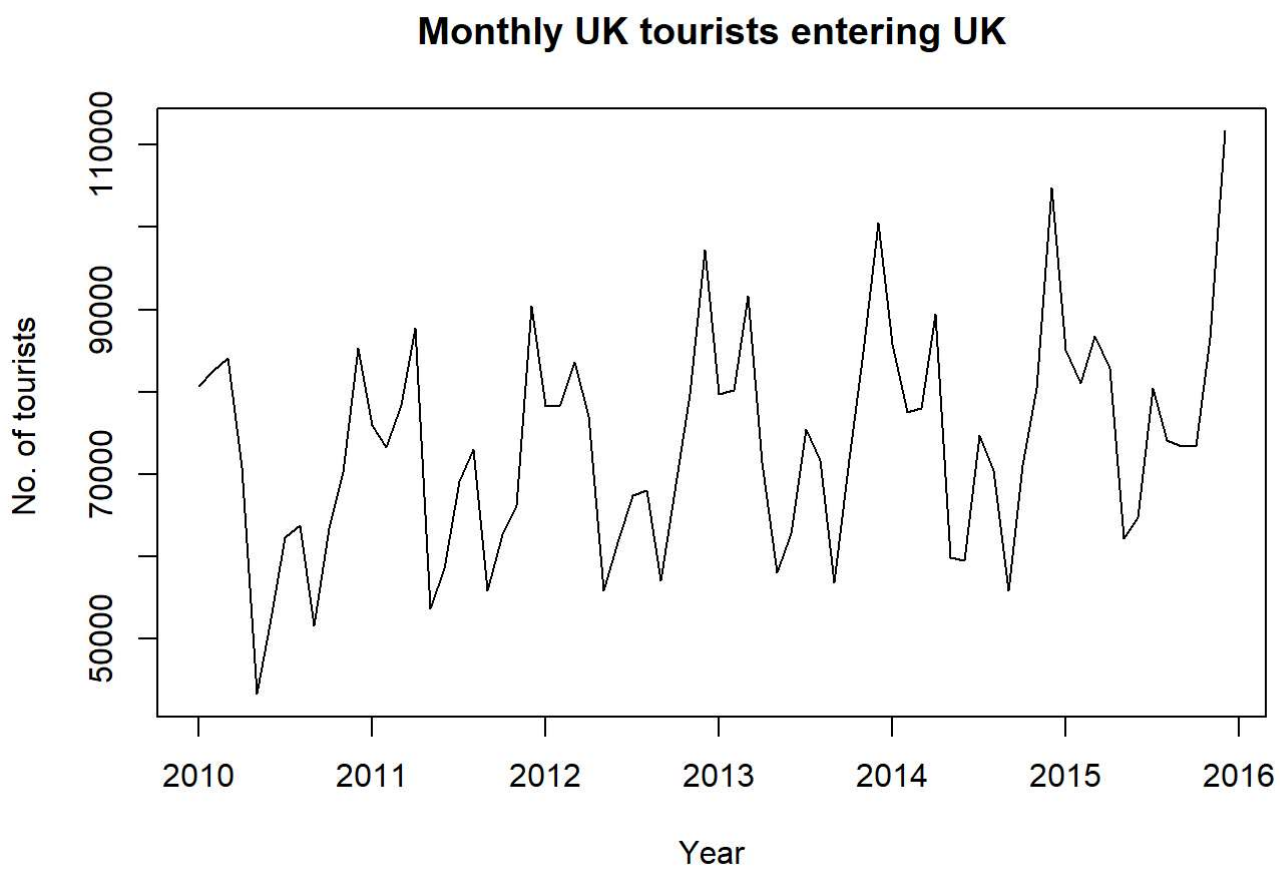
```
df1_test.ts <-ts(df1_test[,3:5], frequency = 12, start = c(2016,1), end = c(2016,12))
head(df1_test.ts)
```

```
##          year month tourists
## Jan 2016 2016     1    94085
## Feb 2016 2016     2    91788
## Mar 2016 2016     3   102343
## Apr 2016 2016     4    81455
## May 2016 2016     5    64760
## Jun 2016 2016     6    67701
```

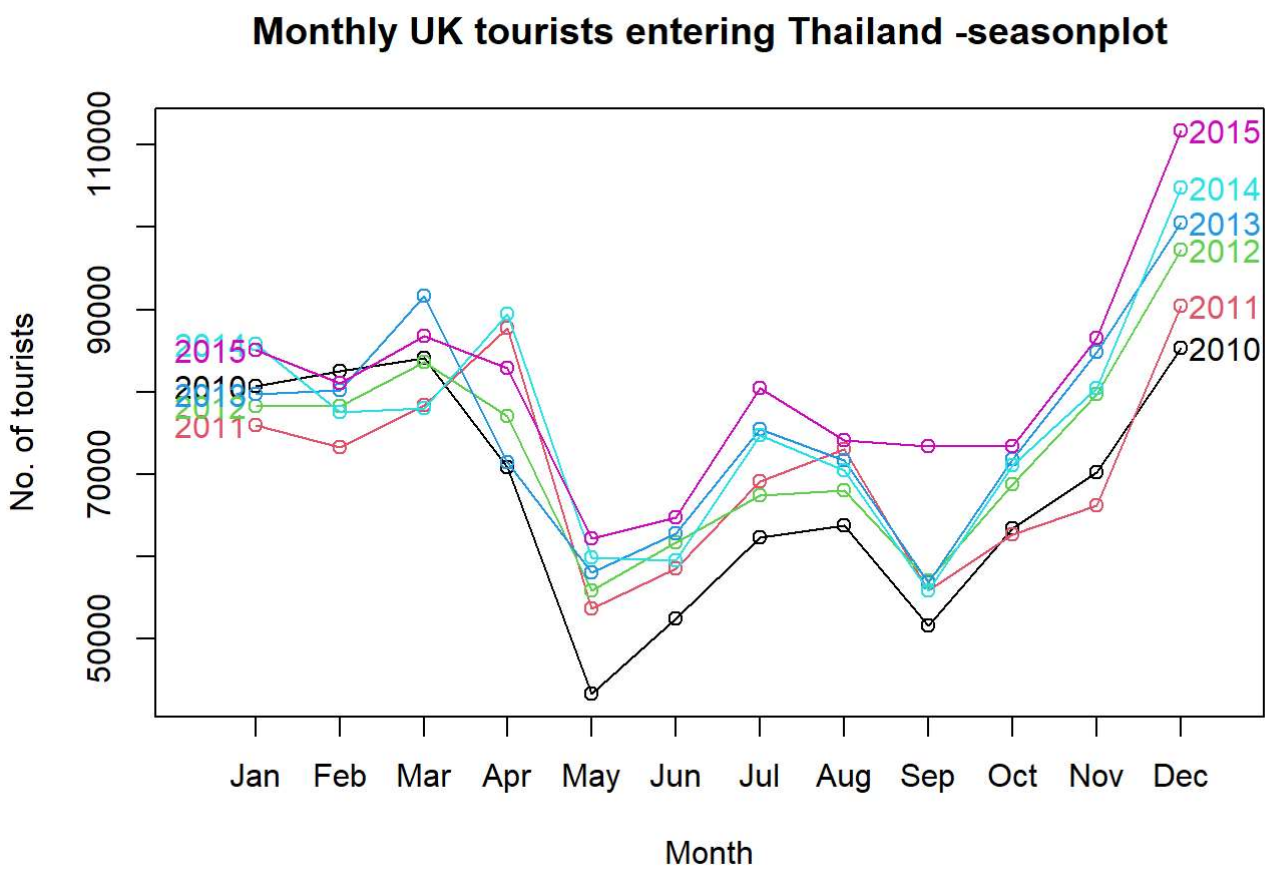
```
tail(df1_test.ts)
```

```
##          year month tourists
## Jul 2016 2016     7    86028
## Aug 2016 2016     8    79525
## Sep 2016 2016     9    62276
## Oct 2016 2016    10    73696
## Nov 2016 2016    11    86747
## Dec 2016 2016    12   112982
```

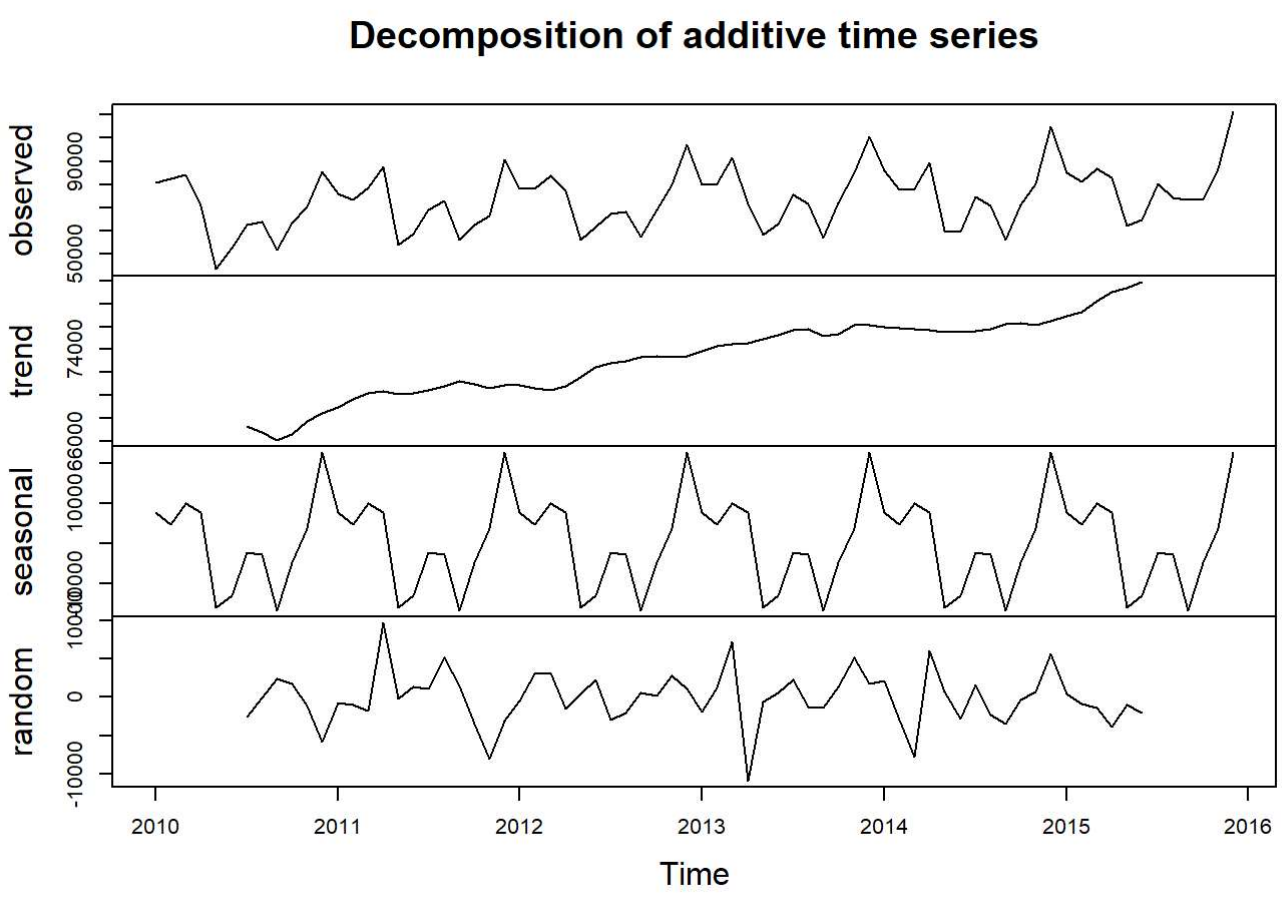
```
plot.ts(df1_train.ts[,3], main = "Monthly UK tourists entering UK", xlab = "Year", ylab = "No. o
f tourists ")
```




```
seasonplot(df1_train.ts[,3], year.labels = TRUE, year.labels.left=TRUE, col=1:10,main = "Monthly
UK tourists entering Thailand -seasonplot",xlab = "Month", ylab = "No. of tourists")
```




```
plot(decompose(df1_train.ts[,3]))
```



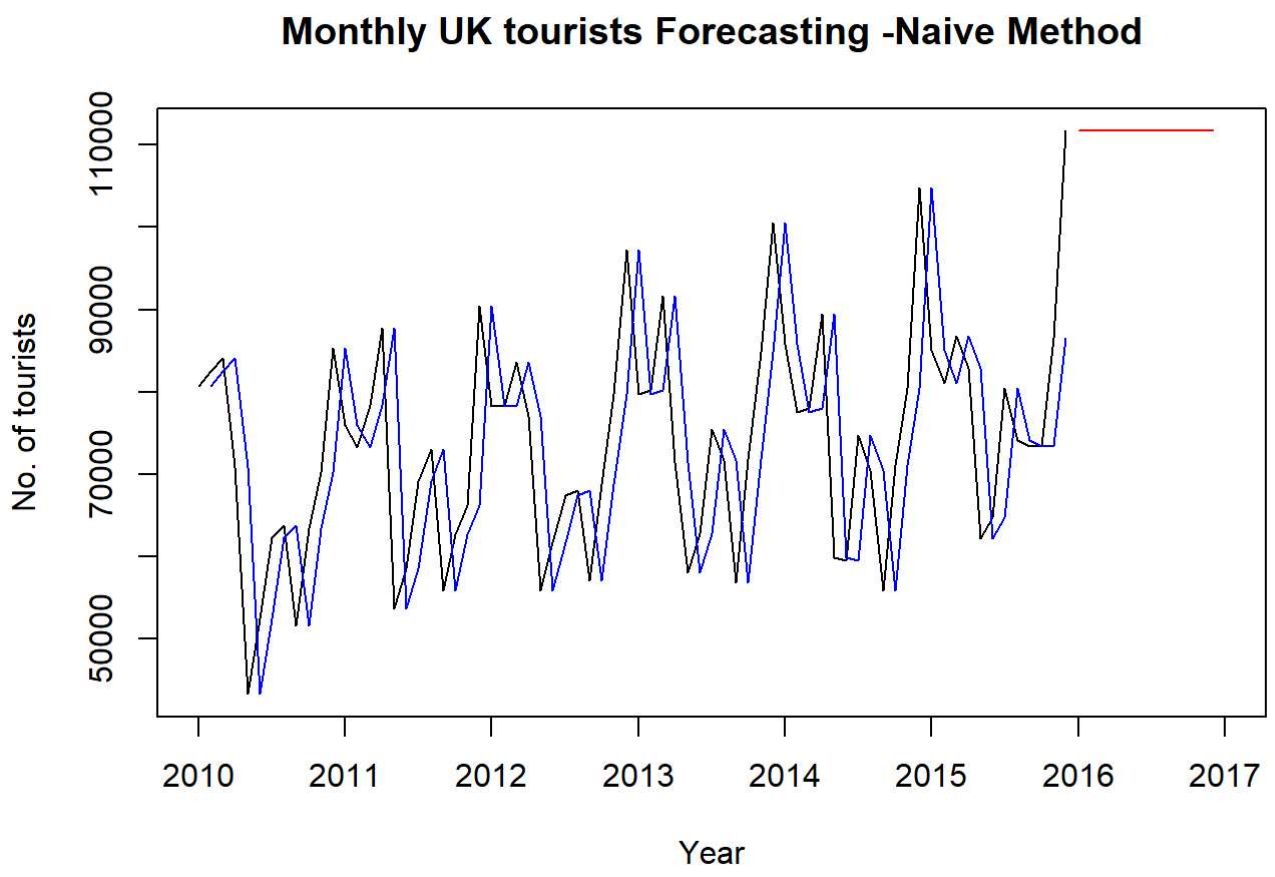

```
naive_mod <- naive(df1_train.ts[,3], h = 12)
summary(naive_mod)
```

```
##
## Forecast method: Naive method
##
## Model Information:
## Call: naive(y = df1_train.ts[, 3], h = 12)
##
## Residual sd: 13168.109
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE         ACF1
## Training set 437.1408 13168.11 10703.51 -1.292921 15.27411 2.027058 -0.07763328
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2016         111689 94813.39 128564.6 85879.98 137498.0
## Feb 2016         111689 87823.28 135554.7 75189.53 148188.5
## Mar 2016         111689 82459.58 140918.4 66986.47 156391.5
## Apr 2016         111689 77937.78 145440.2 60070.96 163307.0
## May 2016         111689 73953.99 149424.0 53978.28 169399.7
## Jun 2016         111689 70352.36 153025.6 48470.07 174907.9
## Jul 2016         111689 67040.33 156337.7 43404.75 179973.2
## Aug 2016         111689 63957.56 159420.4 38690.07 184687.9
## Sep 2016         111689 61062.17 162315.8 34261.94 189116.1
## Oct 2016         111689 58323.63 165054.4 30073.71 193304.3
## Nov 2016         111689 55718.93 167659.1 26090.17 197287.8
## Dec 2016         111689 53230.17 170147.8 22283.93 201094.1
```

```
naive_forecast <-forecast(naive_mod, h = 12, level = c(80,95))
naive_forecast
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 2016         111689 94813.39 128564.6 85879.98 137498.0
## Feb 2016         111689 87823.28 135554.7 75189.53 148188.5
## Mar 2016         111689 82459.58 140918.4 66986.47 156391.5
## Apr 2016         111689 77937.78 145440.2 60070.96 163307.0
## May 2016         111689 73953.99 149424.0 53978.28 169399.7
## Jun 2016         111689 70352.36 153025.6 48470.07 174907.9
## Jul 2016         111689 67040.33 156337.7 43404.75 179973.2
## Aug 2016         111689 63957.56 159420.4 38690.07 184687.9
## Sep 2016         111689 61062.17 162315.8 34261.94 189116.1
## Oct 2016         111689 58323.63 165054.4 30073.71 193304.3
## Nov 2016         111689 55718.93 167659.1 26090.17 197287.8
## Dec 2016         111689 53230.17 170147.8 22283.93 201094.1
```

```
plot.ts(df1_train.ts[,3], main = "Monthly UK tourists Forecasting -Naive Method", xlab = "Year",
ylab = "No. of tourists",  xlim = c(2010, 2017))
lines(naive_forecast$fitted, col ="blue")
lines(naive_forecast$mean, col = "red")
```



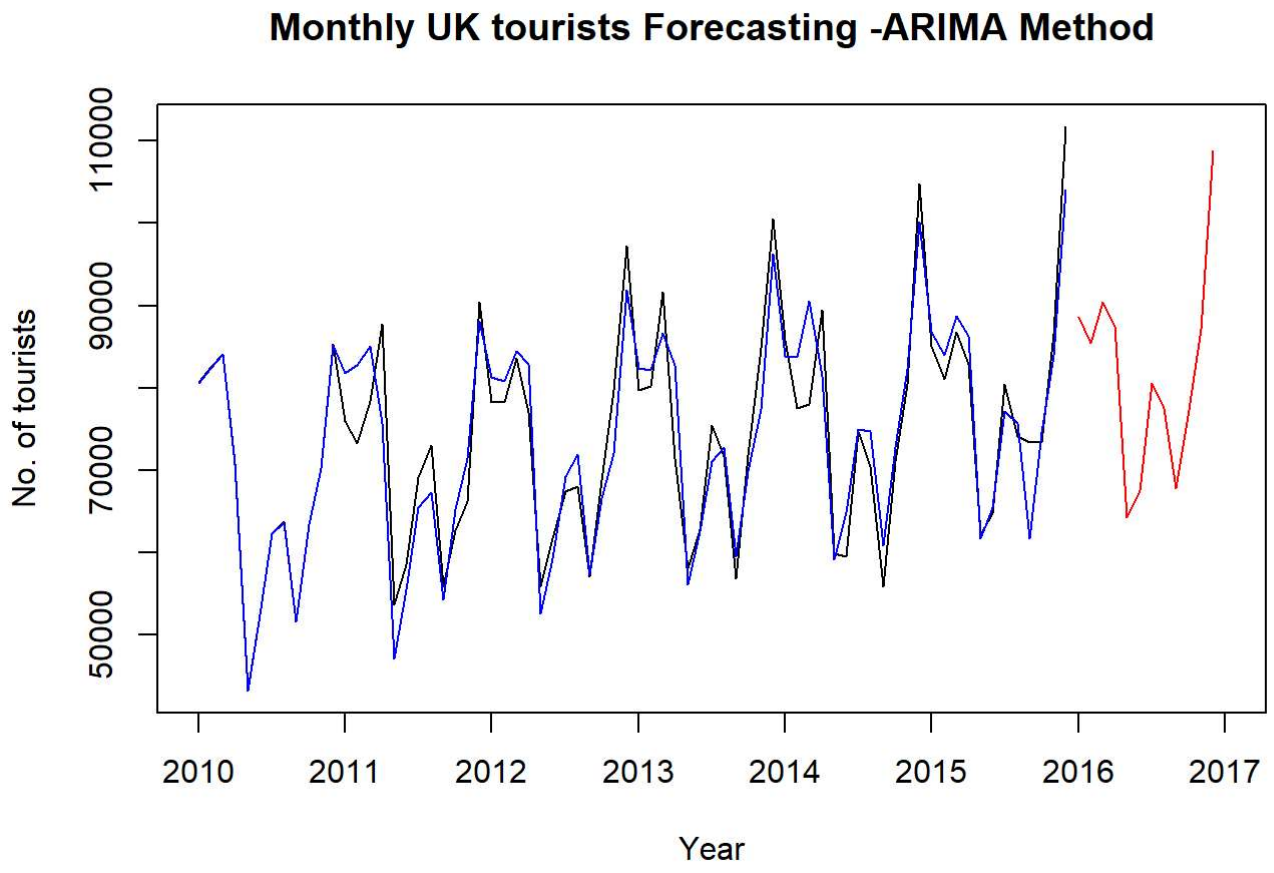

```
arima_model <- auto.arima(df1_train.ts[,3])
summary(arima_model)
```

```
## Series: df1_train.ts[, 3]
## ARIMA(0,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          sma1     drift
##       -0.6821  195.0464
## s.e.   0.1769   29.7180
##
## sigma^2 = 26144730:  log likelihood = -600.19
## AIC=1206.37   AICc=1206.8   BIC=1212.66
##
## Training set error measures:
##                    ME     RMSE      MAE         MPE    MAPE      MASE      ACF1
## Training set 60.48797 4589.23 3352.966 -0.08848885 4.45633 0.6349934 0.106831
```

```
arima_forecast <-forecast(arima_model, h = 12, level = c(80,95))
arima_forecast
```

```
##          Point Forecast     Lo 80     Hi 80    Lo 95     Hi 95
## Jan 2016       88588.66  82017.92  95159.40 78539.59  98637.74
## Feb 2016       85404.48  78833.74  91975.21 75355.40  95453.55
## Mar 2016       90456.70  83885.96  97027.43 80407.62 100505.77
## Apr 2016       87573.84  81003.10  94144.58 77524.76  97622.91
## May 2016       64206.60  57635.86  70777.33 54157.52  74255.67
## Jun 2016       67608.55  61037.81  74179.28 57559.47  77657.62
## Jul 2016       80605.33  74034.60  87176.07 70556.26  90654.41
## Aug 2016       77504.45  70933.71  84075.18 67455.37  87553.52
## Sep 2016       67762.77  61192.03  74333.50 57713.69  77811.84
## Oct 2016       76544.45  69973.72  83115.19 66495.38  86593.53
## Nov 2016       87171.45  80600.71  93742.18 77122.37  97220.52
## Dec 2016      108853.37 102282.64 115424.11 98804.30 118902.45
```

```
plot.ts(df1_train.ts[,3], main = "Monthly UK tourists Forecasting -ARIMA Method", xlab = "Year",
ylab = "No. of tourists",  xlim = c(2010, 2017))
lines(arima_forecast$fitted, col ="blue")
lines(arima_forecast$mean, col = "red")
```



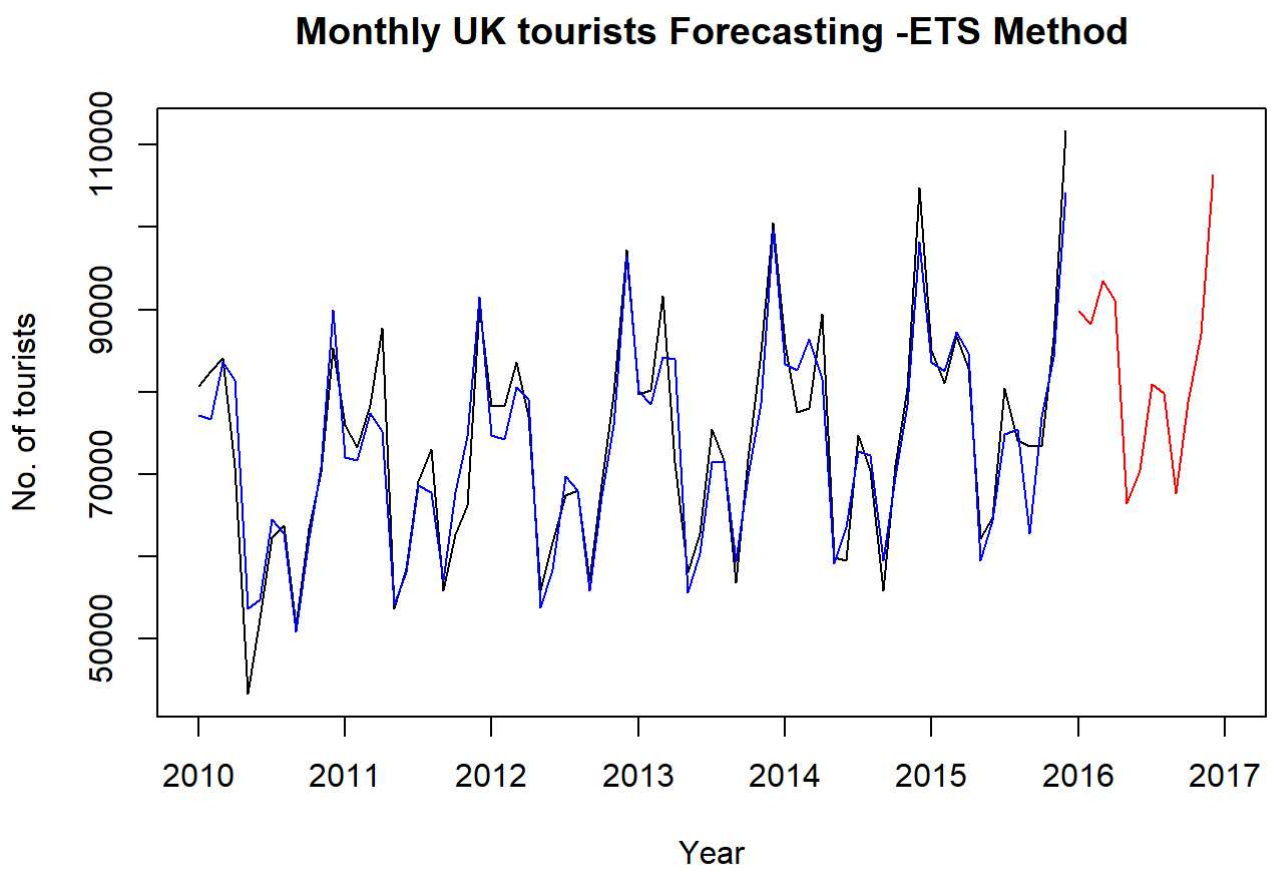

```
ets_model <- ets(df1_train.ts[,3])
summary(ets_model)
```

```
## ETS(M,N,A)
##
## Call:
##  ets(y = df1_train.ts[, 3])
##
##   Smoothing parameters:
##     alpha = 0.2863
##     gamma = 0.0034
##
##   Initial states:
##     l = 70805.0335
##     s = 23002.32 3693.41 -4638.799 -15690.56 -3489.53 -2444.516
##            -12833.89 -16915.88 7816.383 10159.28 4939.476 6402.298
##
##   sigma:  0.0683
##
##      AIC     AICc      BIC
## 1546.598 1555.169 1580.748
##
## Training set error measures:
##                     ME     RMSE     MAE       MPE     MAPE      MASE       ACF1
## Training set 608.0165 4501.834 3341.63 0.3939268 4.589232 0.6328464 0.04143736
```

```
ets_forecast <-forecast(ets_model, h = 12, level = c(80,95))
ets_forecast
```

```
##          Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
## Jan 2016       89788.64 81929.64  97647.64 77769.33 101807.94
## Feb 2016       88299.24 80248.28  96350.20 75986.36 100612.12
## Mar 2016       93508.46 84733.82 102283.10 80088.80 106928.12
## Apr 2016       91131.84 82235.07 100028.62 77525.40 104738.29
## May 2016       66413.08 59026.39  73799.76 55116.11  77710.04
## Jun 2016       70505.31 62653.80  78356.82 58497.46  82513.16
## Jul 2016       80919.86 72152.13  89687.59 67510.77  94328.95
## Aug 2016       79859.34 70931.41  88787.28 66205.25  93513.44
## Sep 2016       67664.85 59300.58  76029.12 54872.80  80456.90
## Oct 2016       78692.42 69460.54  87924.30 64573.48  92811.36
## Nov 2016       87049.71 77061.69  97037.73 71774.35 102325.07
## Dec 2016      106374.58 94828.40 117920.77 88716.21 124032.96
```

```
plot.ts(df1_train.ts[,3], main = "Monthly UK tourists Forecasting -ETS Method", xlab = "Year", y
lab = "No. of tourists",  xlim = c(2010, 2017))
lines(ets_forecast$fitted, col ="blue")
lines(ets_forecast$mean, col = "red")
```




```
h_modelw <-hw(df1_train.ts[,3], h = 12, seasonal = "additive")
summary(h_modelw)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = df1_train.ts[, 3], h = 12, seasonal = "additive")
##
##   Smoothing parameters:
##     alpha = 0.0171
##     beta  = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 68339.6547
##     b = 173.2573
##     s = 22950.5 3606.006 -4852.79 -16427.6 -2895.968 -2434.066
##            -13070.34 -16259.23 7733.423 9982.258 4687.743 6980.063
##
##   sigma:  4895.599
##
##      AIC     AICc      BIC
## 1547.263 1558.596 1585.966
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -417.5801 4317.512 3233.545 -1.068358 4.434415 0.6123771 0.1670509
##
## Forecasts:
##          Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
## Jan 2016       87238.26 80964.30  93512.22 77643.06  96833.46
## Feb 2016       85116.20 78841.31  91391.09 75519.58  94712.82
## Mar 2016       90580.67 84304.84  96856.50 80982.62 100178.73
## Apr 2016       88501.01 82224.23  94777.79 78901.50  98100.52
## May 2016       64678.27 58400.53  70956.02 55077.29  74279.25
## Jun 2016       68038.11 61759.39  74316.82 58435.64  77640.57
## Jul 2016       78845.14 72565.44  85124.84 69241.17  88449.12
## Aug 2016       78552.81 72272.12  84833.51 68947.32  88158.31
## Sep 2016       65192.42 58910.72  71474.12 55585.39  74799.45
## Oct 2016       76936.49 70653.77  83219.20 67327.90  86545.07
## Nov 2016       85566.05 79282.30  91849.79 75955.89  95176.20
## Dec 2016      105081.30 98796.52 111366.08 95469.56 114693.04
```

```
UK.f3 <-forecast(h_modelw , h = 12, level = c(80,95))
UK.f3
```

```
##          Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
## Jan 2016       87238.26 80964.30  93512.22 77643.06  96833.46
## Feb 2016       85116.20 78841.31  91391.09 75519.58  94712.82
## Mar 2016       90580.67 84304.84  96856.50 80982.62 100178.73
## Apr 2016       88501.01 82224.23  94777.79 78901.50  98100.52
## May 2016       64678.27 58400.53  70956.02 55077.29  74279.25
## Jun 2016       68038.11 61759.39  74316.82 58435.64  77640.57
## Jul 2016       78845.14 72565.44  85124.84 69241.17  88449.12
## Aug 2016       78552.81 72272.12  84833.51 68947.32  88158.31
## Sep 2016       65192.42 58910.72  71474.12 55585.39  74799.45
## Oct 2016       76936.49 70653.77  83219.20 67327.90  86545.07
## Nov 2016       85566.05 79282.30  91849.79 75955.89  95176.20
## Dec 2016      105081.30 98796.52 111366.08 95469.56 114693.04
```

```
plot.ts(df1_train.ts[,3], main = "Monthly UK tourists Forecasting -Holt-Winters", xlab = "Year",
ylab = "No. of tourists", xlim = c(2010, 2017))
lines(UK.f3$fitted, col ="blue")
lines(UK.f3$mean, col = "red")
```