Data Analytics Lab ASSIGNMENT-3

Name: Kapil Jhade
Roll no: 17BCS013

****************Predictive Modeling 2: Time Series Analysis**********************

➢ **Apply Time Series methods to analyze the data in any one time series data set**

- I have chosen a quarterly visitors dataset for four different countries from 1998 to 2012.

```
library(igraph)
library(ggplot2)
library(tseries)
library("TTR")
library("forecast")
library(forecast)
visitors=read.table("/home/kapil/desktop/study material notes/6TH SEM/Data Analysis/DA LAB/lab3/visitors.csv", sep = ",", header = TRUE)
visitors
```

Output:

```
> visitors
     Date Australia China Japan United.Kingdom
1  1998Q4     20288  1089  5938          13831
2  1999Q1     22047  1492  6925          23271
3  1999Q2     14362  1450  4353           9756
4  1999Q3     15775  1551  6855           7899
5  1999Q4     21209  2020  6216          15778
6  2000Q1     25261  2364  7061          25362
7  2000Q2     15891  2541  4417          11618
8  2000Q3     17117  2729  7505           8553
9  2000Q4     22761  3292  6778          17512
10 2001Q1     27539  3771  8169          29409
11 2001Q2     17867  3529  5290          12008
12 2001Q3     19460  4542  7710           9611
13 2001Q4     23603  5674  6213          18316
14 2002Q1     28197  6910  7952          34025
15 2002Q2     17807  7734  5476          14763
16 2002Q3     19420  8716  7990          11494
17 2002Q4     24955  8928  7747          21094
18 2003Q1     30426  9432  8908          36605
19 2003Q2     19857  7314  5369          17073
20 2003Q3     20960  6354  7794          13392
21 2003Q4     28140  5880  7543          23566
22 2004Q1     35468  5528  8533          40104
23 2004Q2     23361  4519  5564          18008
```

```
24 2004Q3     24367  4601  6997          13533
25 2004Q4     29689  4766  6678          23879
26 2005Q1     37330  4884  7730          41352
27 2005Q2     22458  4117  4790          18906
28 2005Q3     23878  4239  6027          15460
29 2005Q4     29919  4626  5877          23830
30 2006Q1     37291  5071  7413          42281
31 2006Q2     24032  4422  4459          18743
32 2006Q3     24942  4360  5966          13084
33 2006Q4     32850  4796  5305          24059
34 2007Q1     37850  5331  6238          43611
35 2007Q2     23846  4473  3689          17097
36 2007Q3     26384  4677  5146          11683
37 2007Q4     33016  5072  4799          22492
38 2008Q1     41378  5536  6044          43604
39 2008Q2     24400  4321  3695          17534
40 2008Q3     26825  4389  4795          11827
41 2008Q4     33855  5003  4147          21274
42 2009Q1     39344  6129  5254          37951
43 2009Q2     25402  4742  2679          15028
44 2009Q3     29355  4740  3428          10144
45 2009Q4     36848  6140  3635          21083
46 2010Q1     43797  7271  5391          37139
47 2010Q2     26320  5597  3142          13047
48 2010Q3     30642  5811  3986           8844
49 2010Q4     37501  7266  3807          18189
50 2011Q1     43260  8120  4732          32693
51 2011Q2     25213  5705  2513          11568
52 2011Q3     29521  6414  3834           9795
53 2011Q4     37552  7767  3133          19498
54 2012Q1     41987  9782  4405          29694
```

- **Building a Time Series object with the data.**

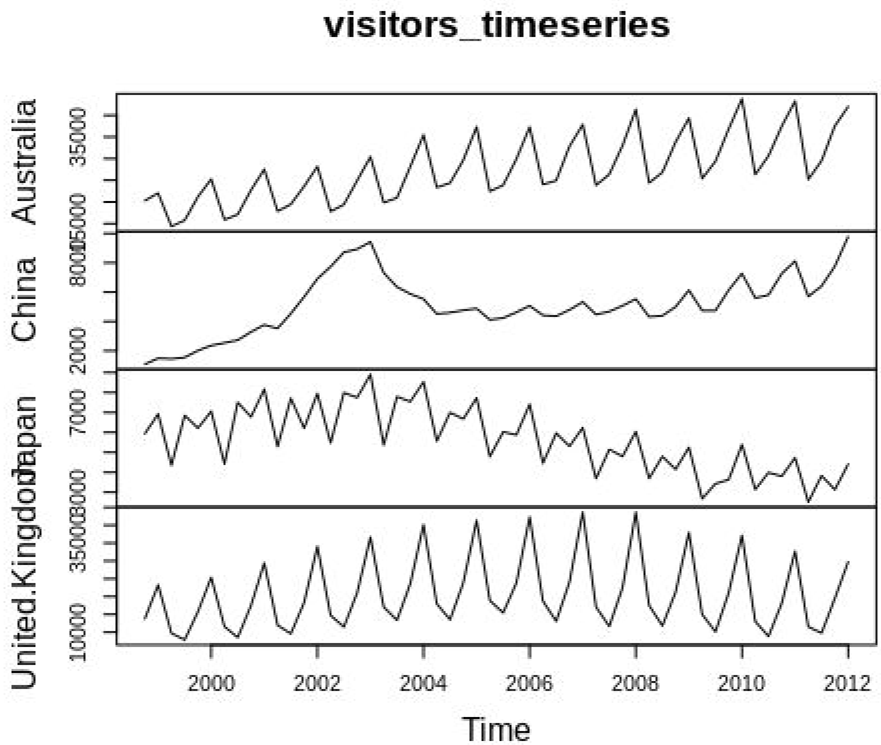
df <- data.frame(visitors)
new_ts <- within(df, rm(Date))
visitors_timeseries <- ts(new_ts, frequency=4, start=c(1998,4)) #this creates time series object
visitors_timeseries

class(visitors_timeseries) #this shows class of time series object
output:
```
> class(visitors_timeseries)
[1] "mts"    "ts"     "matrix"
```

- **Plotting using Time series object**

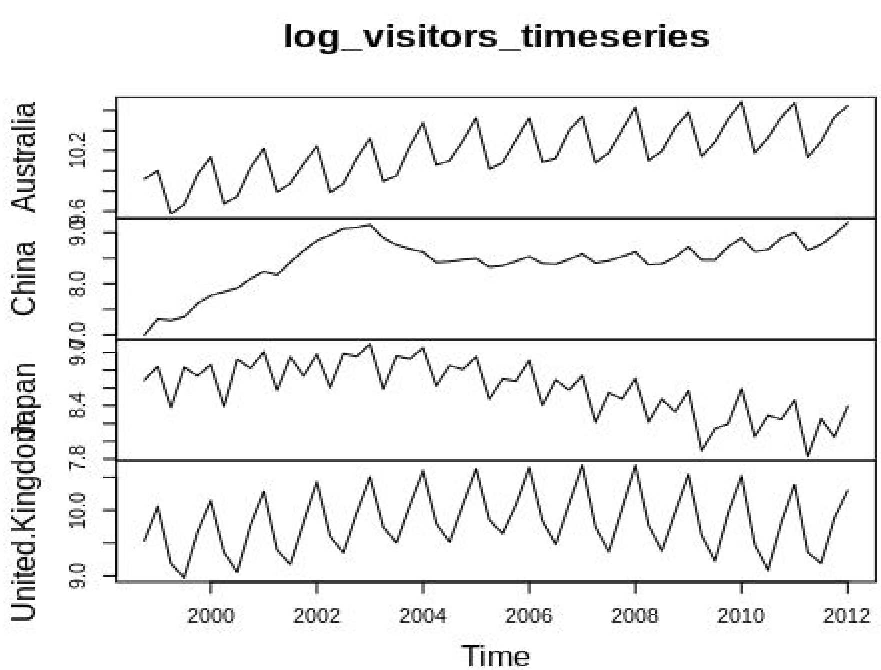
plot.ts(visitors_timeseries)

Output:



- For smoothing purpose plotting taking log

  log_visitors_timeseries <- log(visitors_timeseries)
  plot.ts(log_visitors_timeseries)   #plot of log of ts object

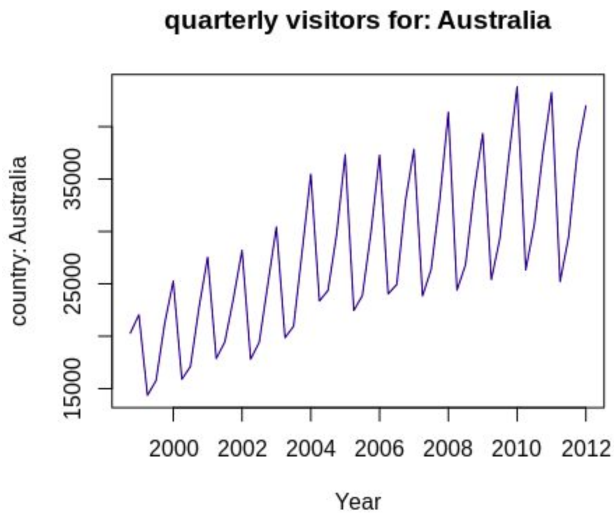


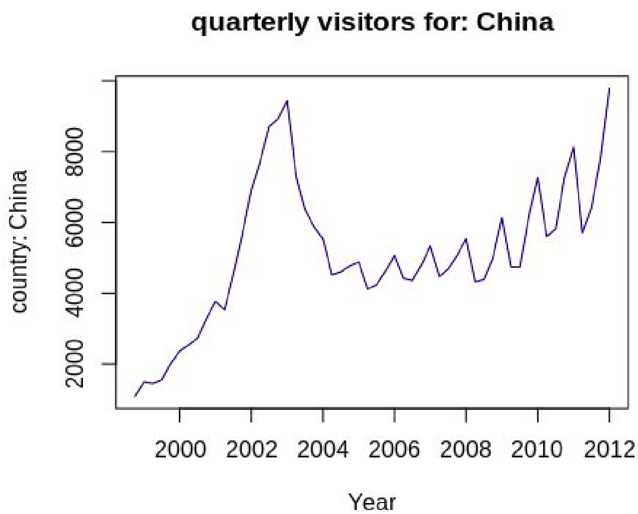
➢ ***Quarterly mean plot for number of visitors for different countries***

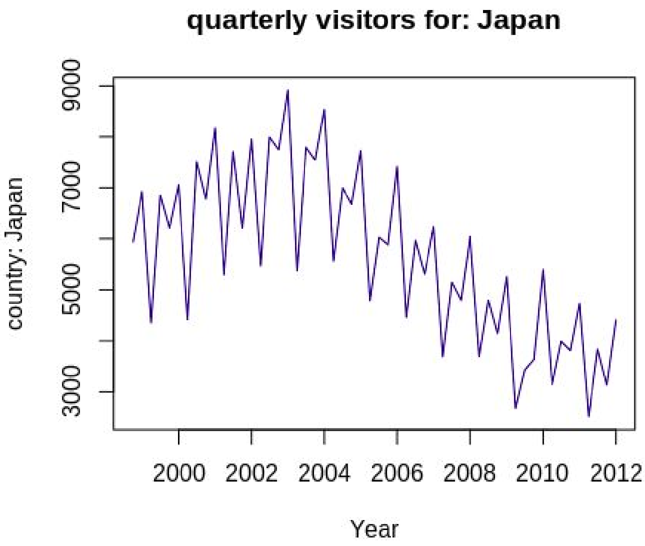
(Below is the mean plot for no of visitors per year)

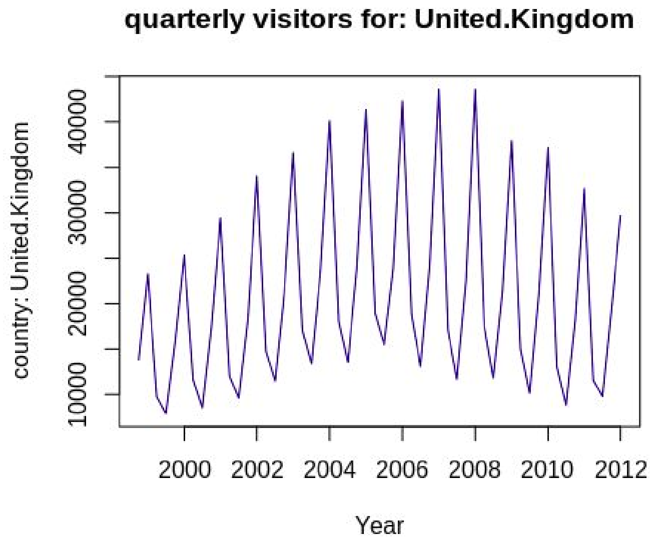
```
numrecords <- nrow(visitors)
numcountries <- ncol(visitors) - 1

#make all the country columns as numeric
for(i in 2:numcountries + 1) {
  visitors[ , i] <- as.numeric(visitors[ , i])
}

#Time series plot for all the 4 countries
for(symbol in 1:numcountries + 1) {

  # The ts function of R helps us to
  # construct a time series
  plot(ts(visitors[ , symbol],
        start=c(1998, 4), end=c(2012, 1),
        frequency=4),
      main=paste("quarterly visitors for:",
              colnames(visitors)[symbol]),
      xlab="Year", ylab=paste("country:",
                      colnames(visitors)[symbol]),
      col="navy")

}
```

Output:







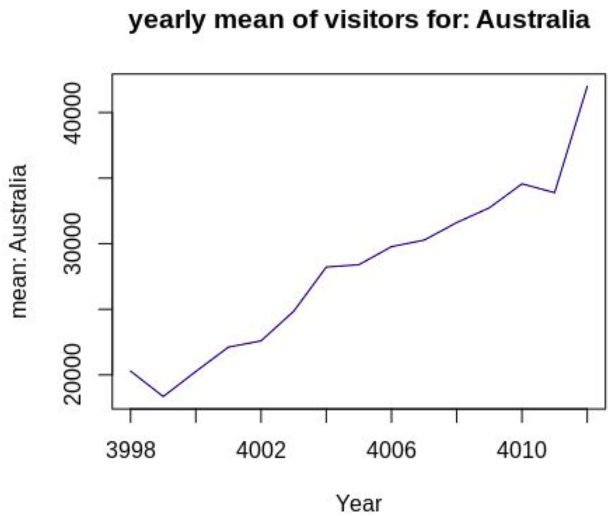


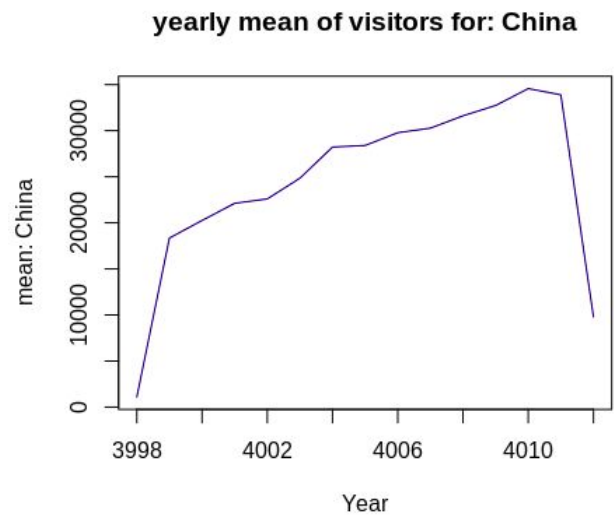
- **<u>Yearly boxplots and TS plot for mean number of visitors different countries</u>**

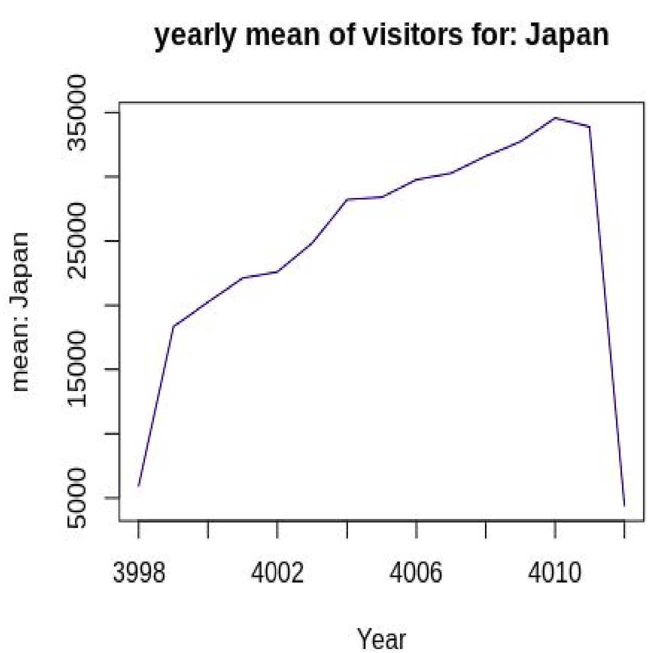
```
#Yearly mean plot of time series and box plot
for(symbol in 1:numcountries + 1) {
  x <- c()
  column <- colnames(visitors)[symbol]
  p <- head(visitors[[column]],1)
  x <- c(x,p)
  for(i in seq(from=2, to=53, by=4)){
   j <- i+3
   yearly_visitors <- visitors[i:j,]
   #print (yearly_visitors)
   m=mean(yearly_visitors$Australia)
   #print (m)
   x <- c(x,m)
  }
  column <- colnames(visitors)[symbol]
  q <- tail(visitors[[column]],1)
  x <- c(x,q)
  print (x)

  mean_list <- c()
  for(k in 1998:2012+1){
   mean_list <- c(mean_list,k)
  }
  print (mean_list)
  ml <- mean_list

  #TS plot for yearly mean of visitors
  plot(ts(x,ml),
      main=paste("yearly mean of visitors for:",
           colnames(visitors)[symbol]),
      xlab="Year", ylab=paste("mean:",
                colnames(visitors)[symbol]),
```

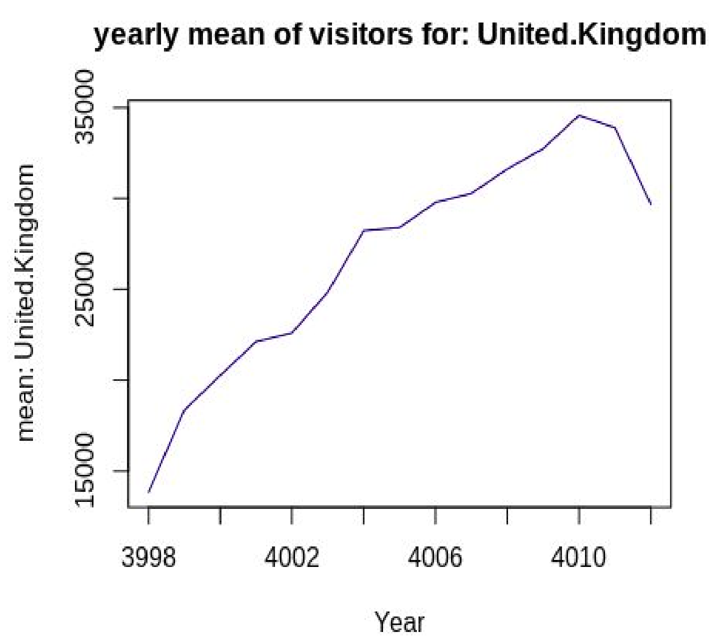
```
    col="navy")

  #Box plot for yearly mean of visitors
  ggplot() + geom_boxplot(mapping = aes(x = ml ,y = x)) + labs(x = "Year", y =  "Mean")

}
```

Output:



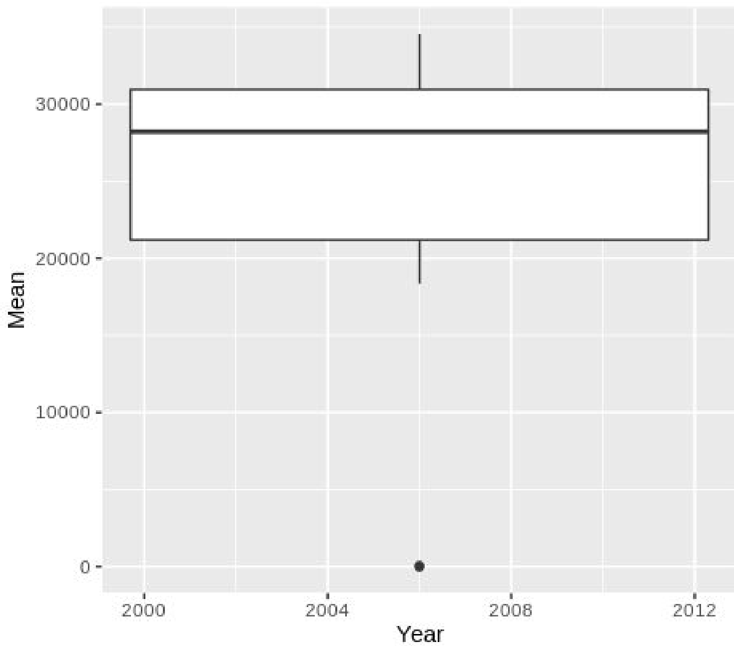




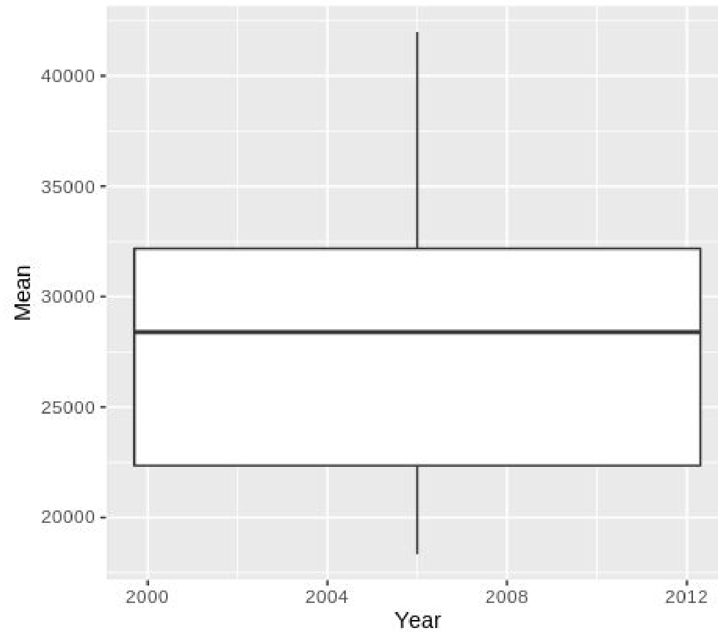


- Box plot for mean of all the visitor of different countries

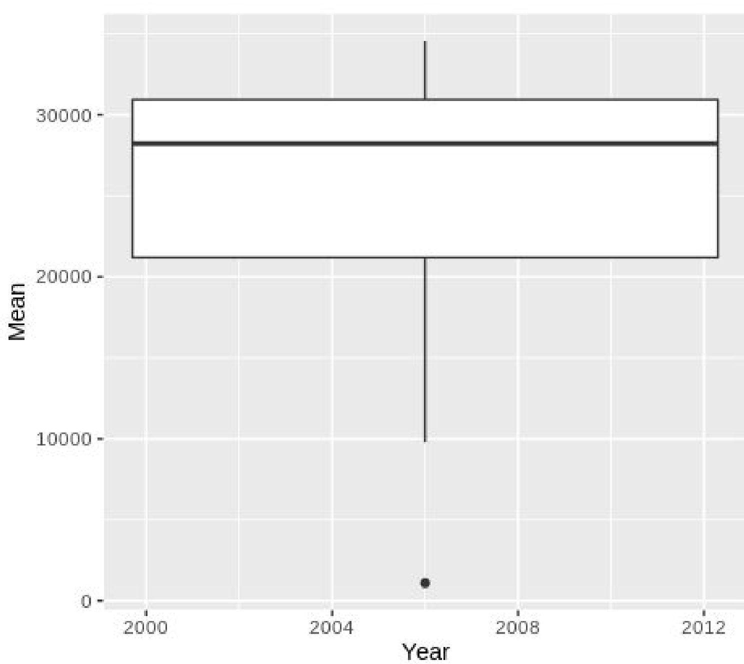
Australia



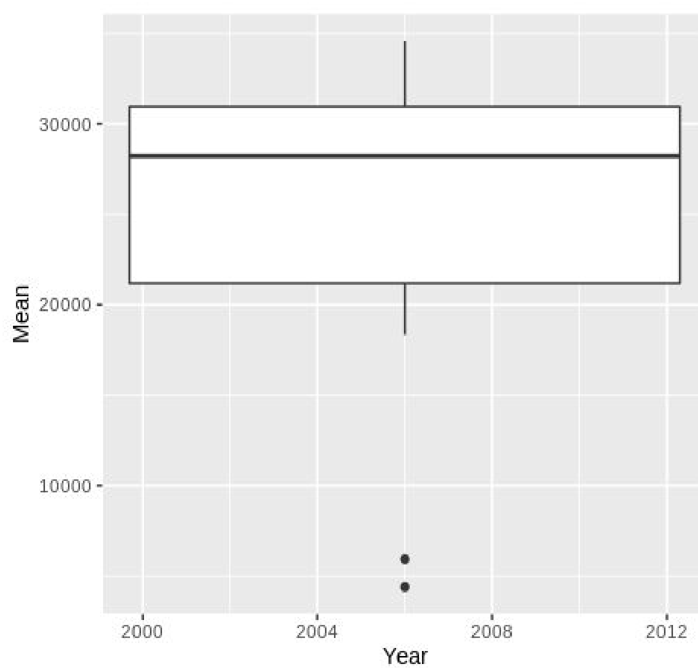
China



Japan



United Kingdom



Now for further Analysis we need single variant Time Series data, hence i have only taken the quarterly visitors of Australia from 1998 to 2012

Australia <- within(df, rm(Date,China,Japan,United.Kingdom))
Australia_ts <- ts(Australia, frequency=4, start=c(1998,4))
Australia_ts

Output:

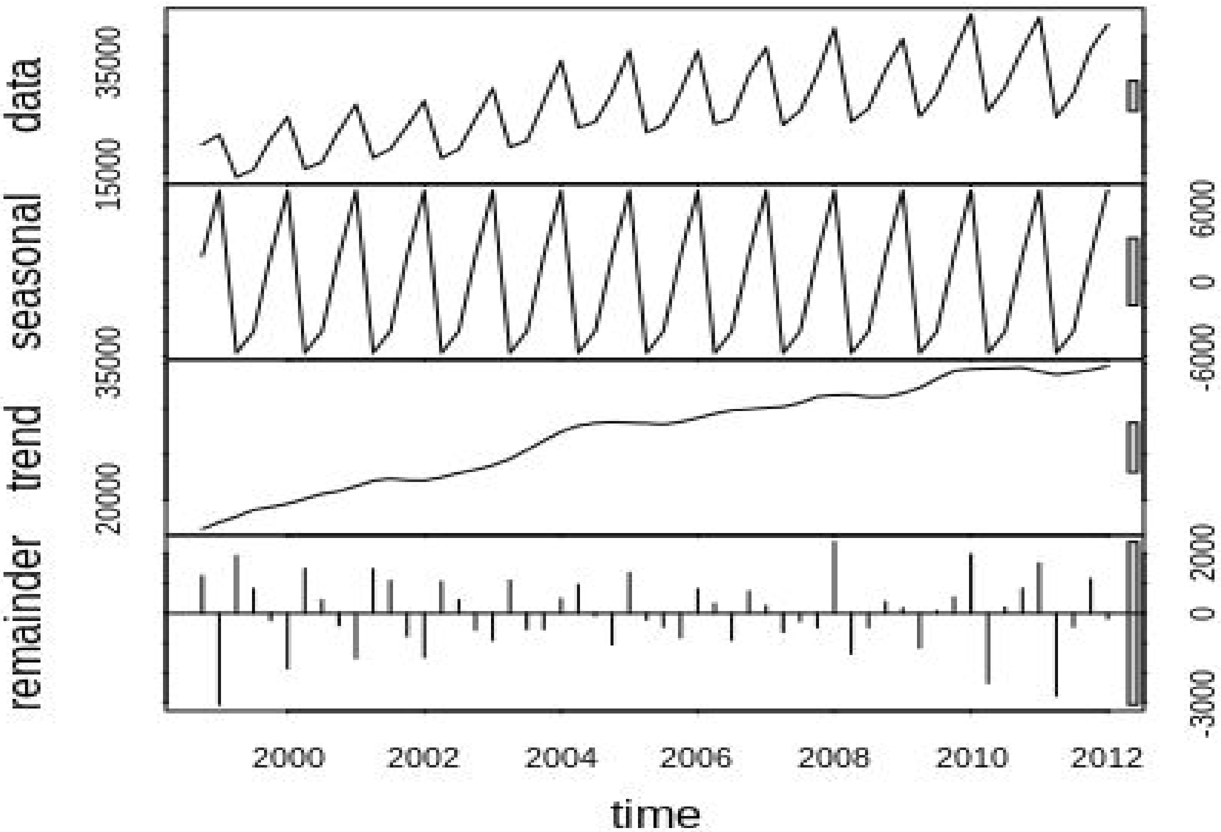
```
> Australia_ts
      Qtr1  Qtr2  Qtr3  Qtr4
1998                   20288
1999 22047 14362 15775 21209
2000 25261 15891 17117 22761
2001 27539 17867 19460 23603
2002 28197 17807 19420 24955
2003 30426 19857 20960 28140
2004 35468 23361 24367 29689
2005 37330 22458 23878 29919
2006 37291 24032 24942 32850
2007 37850 23846 26384 33016
2008 41378 24400 26825 33855
2009 39344 25402 29355 36848
2010 43797 26320 30642 37501
2011 43260 25213 29521 37552
2012 41987
```

- **Decomposing the time series using the stl function, trend, seasonality, and residue.**

(Below mentioned piece of code decompose the seasonal time series data into different components)

Australia_ts.stl = stl(Australia_ts[,1], s.window="periodic")
plot(Australia_ts.stl)

Plot below represents the type of trend, seasonality and the remainder left after removing the seasonality and the trend.



- The above model is a multiplicative model it suggests that the components are multiplied together as follows:
  y(t) = Level * Trend * Seasonality * Noise

- A multiplicative model is nonlinear, such as quadratic or exponential. Changes increase or decrease over time.
- **Trend here is non-linear.**
- **Seasonality here is also non-linear and has an increasing or decreasing frequency and/or amplitude over time.**
- **Components of the time series is Systematic that are consistent or recurring and can be described and modeled.**
- **Residue shows the Time series after removing trend and seasonality.**

➢ **<u>Building a model of the data using the HoltWinters method for the period upto about 75% of the data</u>**

#'alpha' is the exponential in the moving average model, 'beta' controls how trend is up and 'gamma' controls how is the updation of the seasonal value.

Aust_ts <- ts(Australia_ts, frequency=4, start=c(1998,4),end=c(2008,3))

Australia_mean <- HoltWinters(Aust_ts, gamma = FALSE)

#n.ahead below represent the 25% of my data as i have 54 rows

Australia.pred <- predict(Australia_mean,n.ahead=14,prediction.interval = TRUE)

#plotting prediction for rest 25% of data

plot.ts(Australia_ts, xlim = c(1998,2020), ylim = c(0,60000))

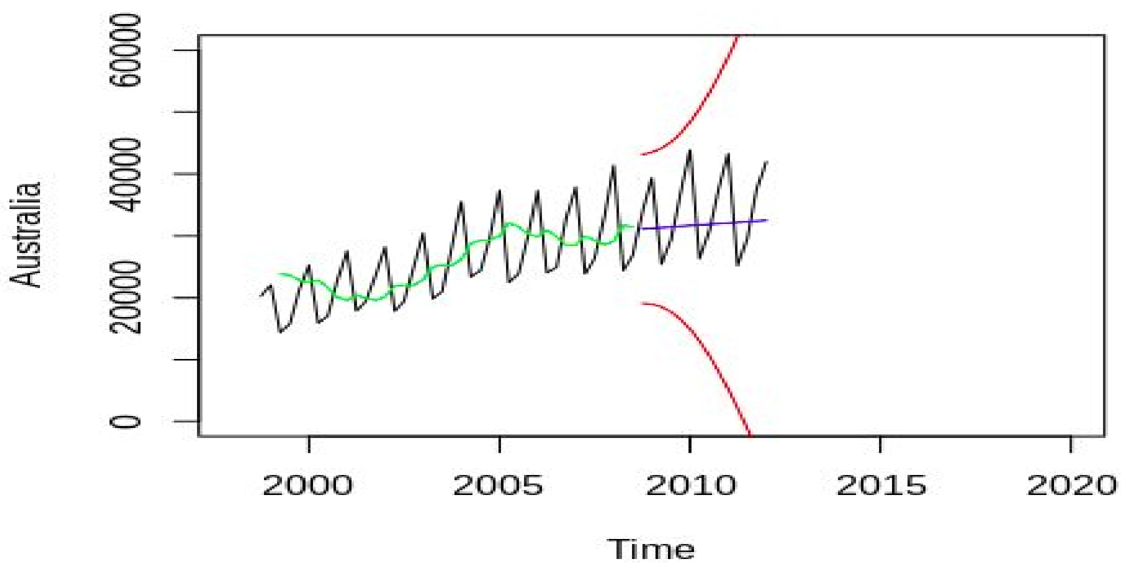
lines(Australia_mean$fitted[,1],col="green")

lines(Australia.pred[,1], col="blue")

lines(Australia.pred[,2], col="red")

lines(Australia.pred[,3], col="red")

Output:



- **Parameter values at gamma= FALSE**

Australia_mean <- HoltWinters(Aust_ts, gamma = FALSE)

Australia_mean

OUTPUT:

> Australia_mean

Holt-Winters exponential smoothing with trend and without seasonal component.

Call:

HoltWinters(x = Aust_ts, gamma = FALSE)

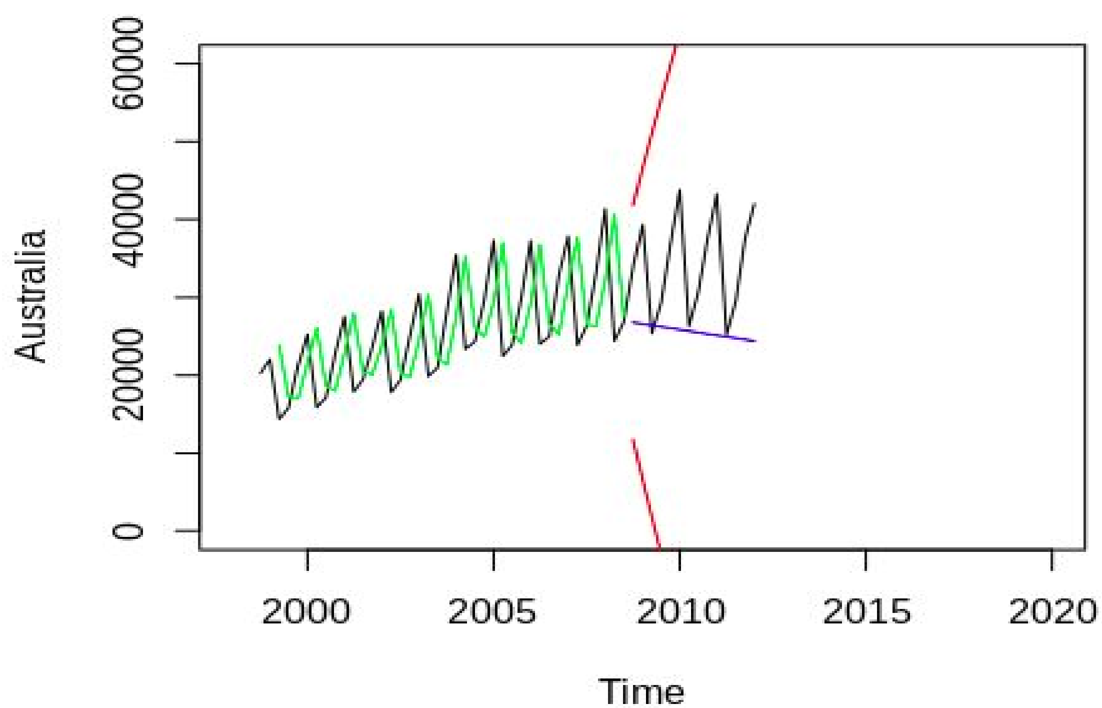
```
Smoothing parameters:
 alpha: 0.1016998
 beta : 1
 gamma: FALSE
Coefficients:
        [,1]
a 31015.0454
b   104.6343
```

- **Fine tuning the model by changing alpha, beta and gamma.**

  Australia_mean <- HoltWinters(Aust_ts, alpha = 0.8, beta = 0.1,gamma = FALSE)

  #prediction plot at above parameters

  Output:



(red lines represents the upper and the lower bound of the prediction and the blue line represents the prediction fit for the rest 25% of the data compared with the actual data)

- **Computing the rms error between the predicted and actual values.**

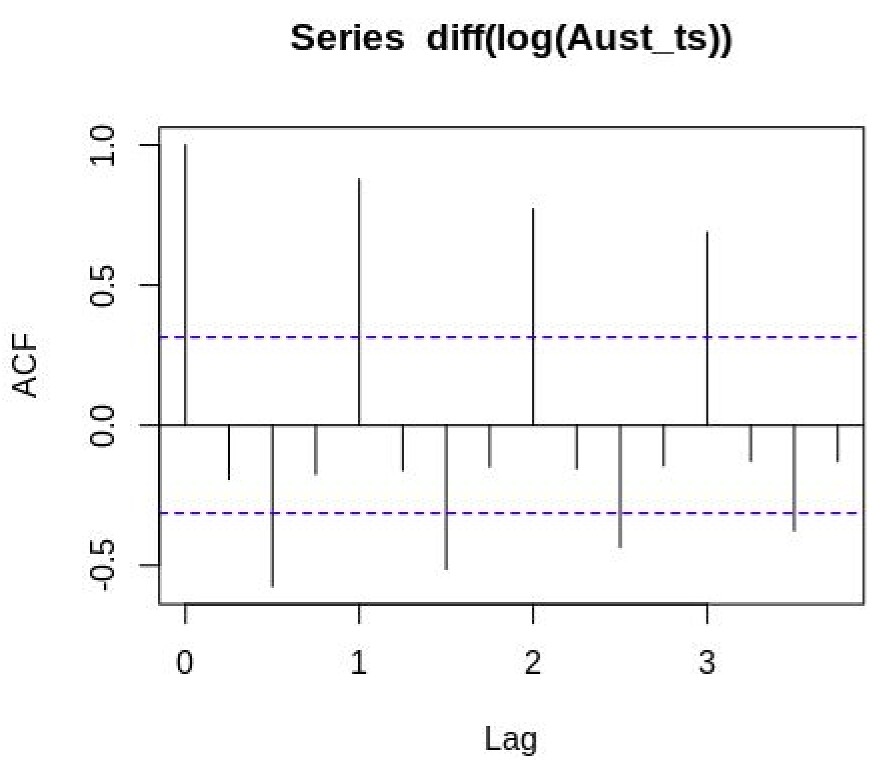
  sqrt(Australia_mean2$SSE)

  Output:

  ```
  > sqrt(Australia_mean$SSE)
  [1] 46827.84
  ```

➢ **Building an ARIMA model for the period up to about 75% of the data.**

- **Calculating autocorrelation function for 75% of data i.e. upto 2008,3**
  Aust_ts <- ts(Australia_ts, frequency=4, start=c(1998,4),end=c(2008,3))
  acf(diff(log(Aust_ts))) #q=0
  Output:

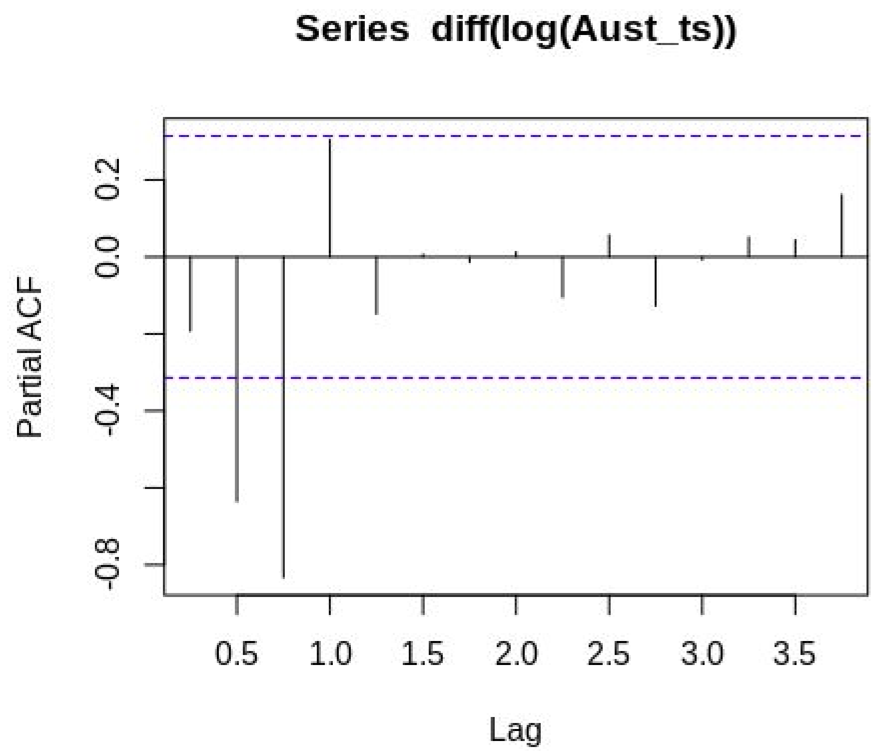


(This gives q=0 as second line is accepted)

- **Calculating partial autocorrelation function for 75% of data i.e. upto 2008,3**

  ```
  pacf(diff(log(Aust_ts))) #p=-1
  ```

  Output:

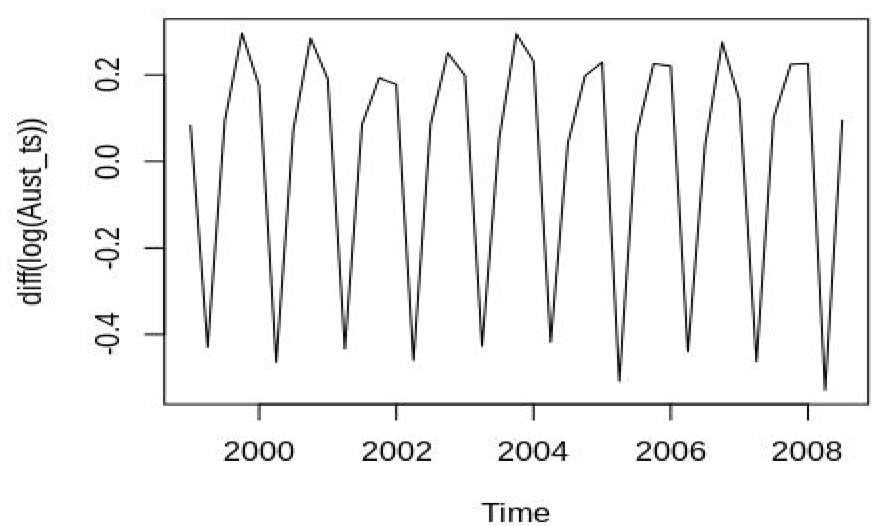


(here p=-1 as first line is accepted)

- **Plotting log differential of the 75% of the time series data**

  plot(diff(log(Aust_ts)))

  output:



- **Prediction for next 15 months**

  #c(p,d,q) where p is the number of autoregressive terms, d is the number of nonseasonal differences needed for stationarity, and q is the number of lagged forecast errors in the prediction equation.

  fit=arima(log(Aust_ts),c(-1,1,0),seasonal = list(order=c(-1,1,0),period=4))

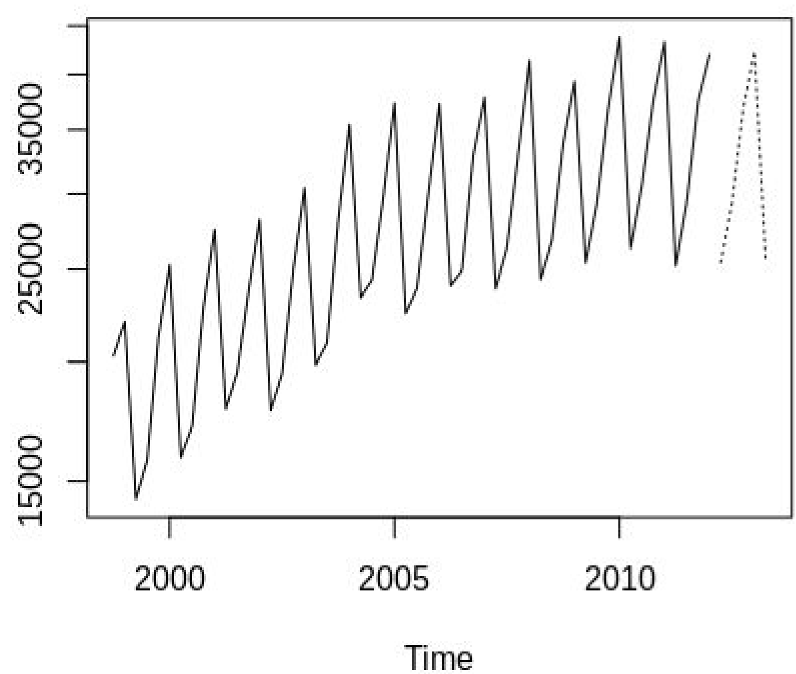
  fit

  *pred = predict(fit, n.ahead = 1.3*4)* ***#predicting for 15 months(1.3yrs) and 4(quarterly data)***

  pred1 = round(2.718^pred$pred,0) ***#converting prediction to decimal value***

  ts.plot(Australia_ts, pred1, log= "y", lty = c(1,3)) ***#plotting actual data and prediction for next 15 months***

  output:



(In above plot dotted line predicts for the next 15 months of the time series data)

- **Computing the rms error between the predicted and actual values.**

  sqrt(fit$SSE)

  Output:

```
> sqrt(fit$SSE)
[1] 37416.07
```

- Here the plot is finely tuned using the p and q values obtained from the acf and pacf plots. And we observe that the rms value of predicted and the original data decreases on fine tuning.

★ Based on my experiments, in forecasting seasonal data I would say the **Arima model is better** as it takes more parameters and provides better prediction than HoltWinters model which does not model the variables per se, they give you a procedure to forecast a given variable and is based on smoothing factors.

★ In order to implement these methods **I had to convert my dataset from multivariate to univariant**, as many time series functions are not applicable on multivariate data.

****