PRACTICAL 5: Practical of Time-Series Forecasting

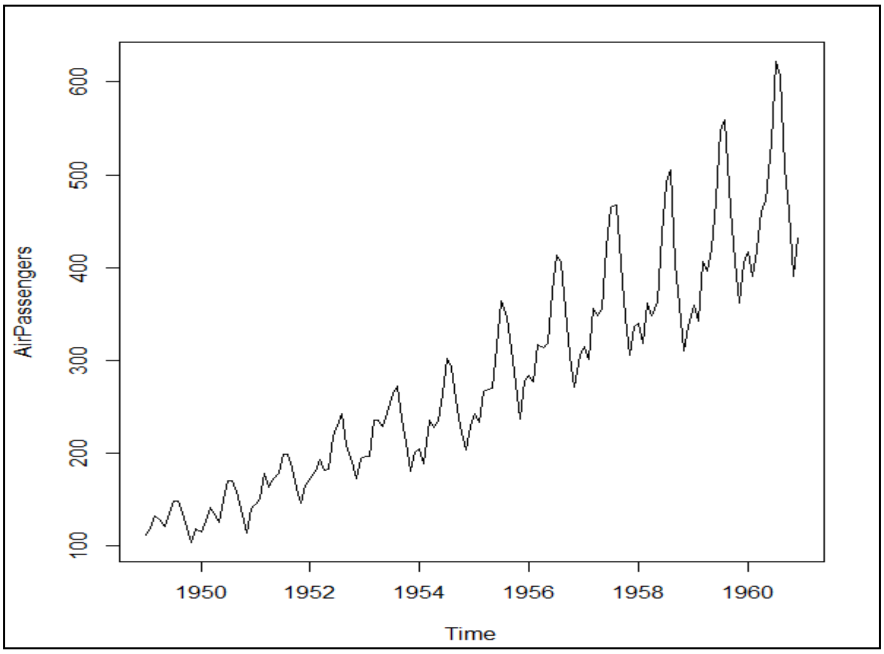
Aim: Practical of time series

Step 1: Adding Data Air Passenger and information about data

```
#consider inbuilt data set AirPassengers
data("AirPassengers")
#to know the format of data set here
class(AirPassengers)
#to know start of time series
start(AirPassengers)
#to know the end of time series
end(AirPassengers)
#to know frequency of time series, 12 means its on a monthly basis
frequency(AirPassengers)
#to know summary of the dataset i.e. mean, median, etc.
summary(AirPassengers)
```

```
> data(AirPassengers)
> class(AirPassengers)
[1] "ts"
> start(AirPassengers)
[1] 1949    1
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  104.0   180.0   265.5   280.3   360.5   622.0 
```

```
plot(AirPassengers)
```



Analysis: After plotting the time series model we can see that number of passengers increases every year. And there is higher variance as the year increases.

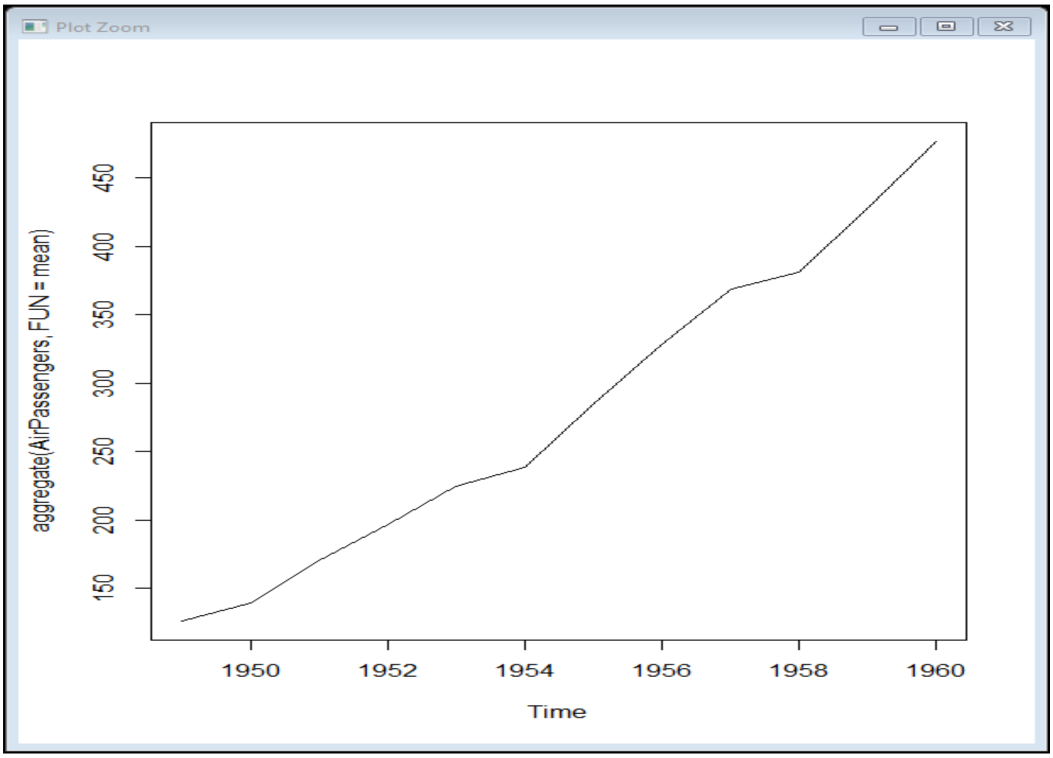
Step 2:

```
#To plot a best fit line which can be used for regression
abline(reg=lm(AirPassengers~time(AirPassengers)))
#To print the cycle across years
cycle(AirPassengers)
```

```
> abline(reg=lm(AirPassengers~time(AirPassengers)))
> cycle(AirPassengers)
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949   1   2   3   4   5   6   7   8   9  10  11  12
1950   1   2   3   4   5   6   7   8   9  10  11  12
1951   1   2   3   4   5   6   7   8   9  10  11  12
1952   1   2   3   4   5   6   7   8   9  10  11  12
1953   1   2   3   4   5   6   7   8   9  10  11  12
1954   1   2   3   4   5   6   7   8   9  10  11  12
1955   1   2   3   4   5   6   7   8   9  10  11  12
1956   1   2   3   4   5   6   7   8   9  10  11  12
1957   1   2   3   4   5   6   7   8   9  10  11  12
1958   1   2   3   4   5   6   7   8   9  10  11  12
1959   1   2   3   4   5   6   7   8   9  10  11  12
1960   1   2   3   4   5   6   7   8   9  10  11  12
```

Step 3:

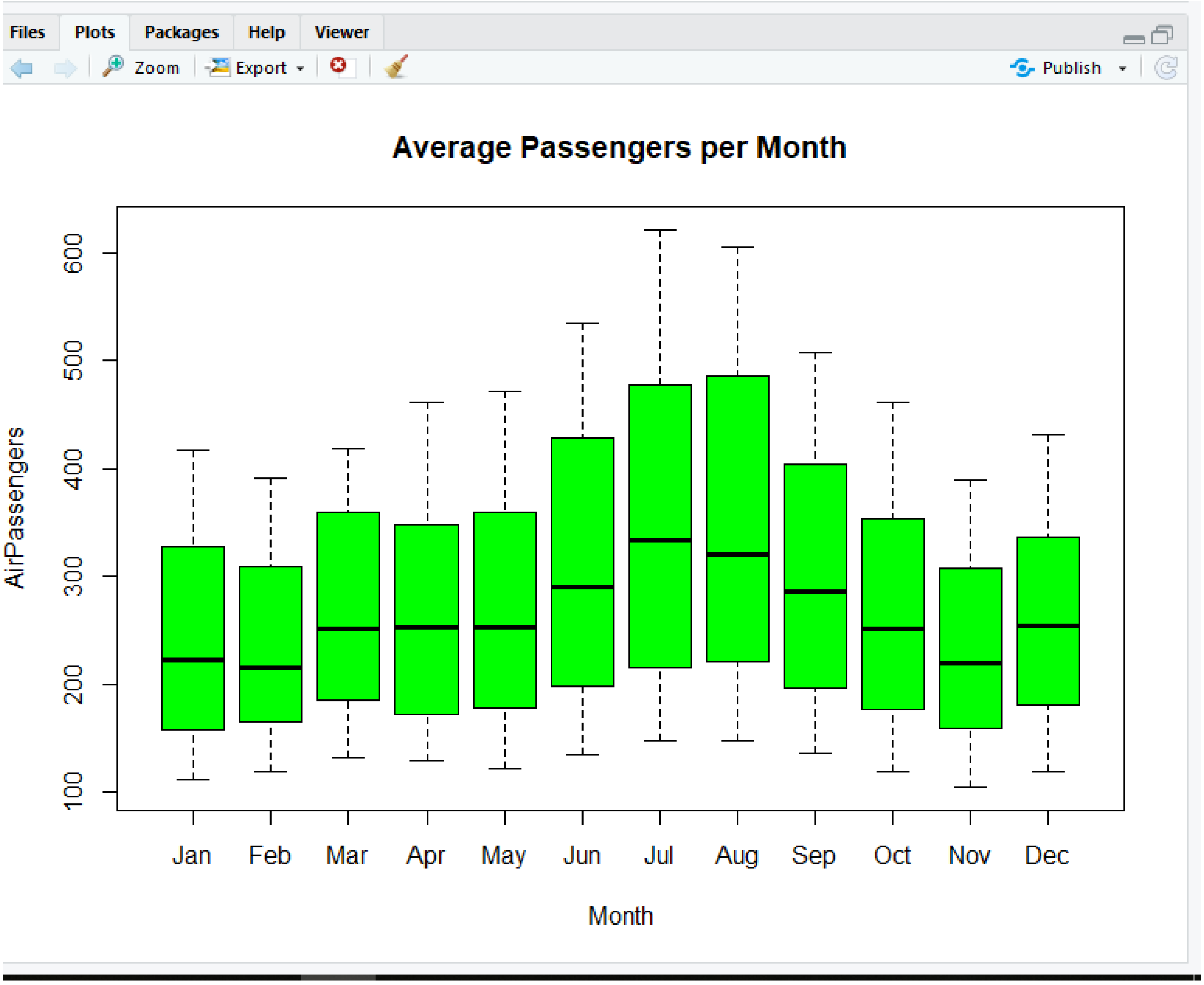
```
#Aggregate cycles and show a year to year trend
plot(aggregate(AirPassengers,FUN=mean))
```



Analysis: we see that passengers tend to increase year by year

Step 4:

```
#Using a box plot we will try to get a sense for a possible seasonal effect
boxplot(AirPassengers~cycle(AirPassengers), xlab = "Month", ylab = "AirPassengers", main = "Average Passengers per Month", names = month.abb, col = "green")
```

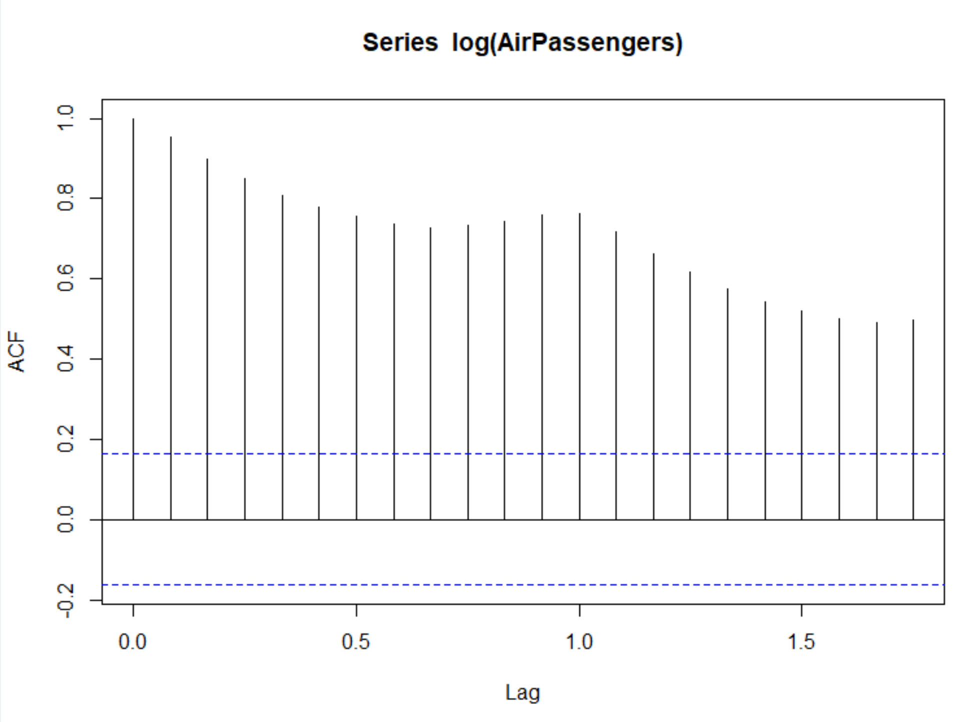


Analysis:

1. The number of passengers appears to increase each year, with variance getting higher as the number of passengers increase
2. The year to year trend also shows that passengers tend to increase with time
3. The variance and mean values in July and August are much higher compared to the rest of the months
4. Even though the mean value for each month is different, the variance is small. Therefore, we have strong seasonal effect with a cycle of 12 months or less

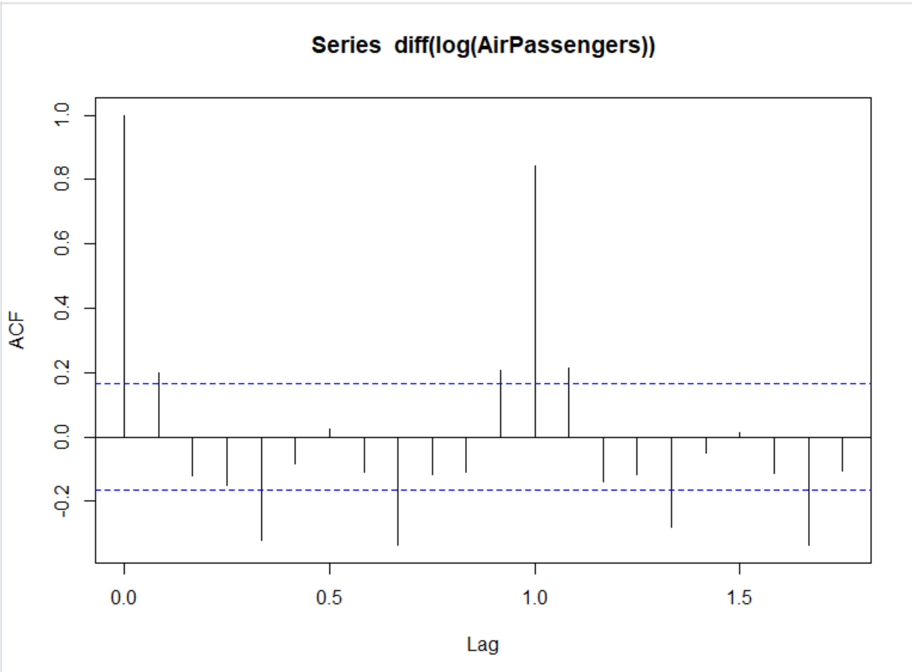
Step 5:

```
#We now need to check whether population is stationary or non-stationary. For that we use acf(autocorrelation) function
acf(log(AirPassengers))
```



Analysis: Clearly, the decay of ACF chart is very slow, which means that the population is not stationary.

```
#We now regress on the difference of logs rather than log directly.
acf(diff(log(AirPassengers)))
```



Analysis: Clearly, ACF plot cuts off after the first lag.

Step 6:

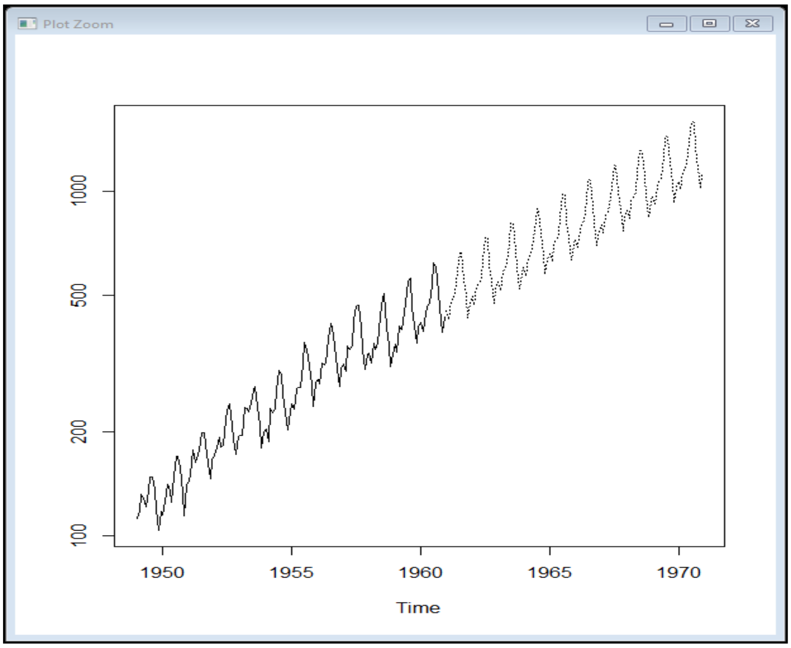
```
#Finally, we fit an ARIMA model to our time series and predict the
future 10 years. In addition, we will try to fit a seasonal
component in the ARIMA formulation.
```

```
(fit <- arima(log(AirPassengers), c(0, 1, 1),seasonal = list(order = c(0,
1, 1), period = 12)))
```

```
> (fit <- arima(log(AirPassengers), c(0,1,1), seasonal = list(order = c(0,1,1),
period=12)))

Call:
arima(x = log(AirPassengers), order = c(0, 1, 1), seasonal = list(order = c(0,
    1, 1), period = 12))

Coefficients:
          ma1     sma1
      -0.4018  -0.5569
s.e.   0.0896   0.0731

sigma^2 estimated as 0.001348:  log likelihood = 244.7,  aic = -483.4
```

Step 7 :

```
#We now visualize the prediction along with the training data.
pred <- predict(fit, n.ahead = 10*12)
ts.plot(AirPassengers,2.718^pred$pred, log = "y", lty = c(1,3))
```



explanations for the ts.plot arguments provided:

1) 2.718^APforecast$pred: we are undoing the log from the values.In order to do that, we need to find the log inverse of what we have got.
   i.e. log(forecast) = pred$pred
   hence, forecast = e ^ pred$pred
   e= 2.718
2) log = "y' is to plot on a logarithmic scale
3) lty = c(1,3) will set the LineType to 1 (for solid) for the original time series and 3 (for dotted) for the predicted time series.

Analysis:

From the predicted data we can see that even for the next 10 years same trend will be seen i.e.:
passengers tend to increase year by year and there is higher variance as the year increases.