Exploration of Time Series Data in R

Here we'll learn to handle time series data on R. Our scope will be restricted to data exploring in a time series type of data set and not go to building time series models.

I have used an inbuilt data set of R called AirPassengers. The dataset consists of monthly totals of international airline passengers, 1949 to 1960.

Loading the Data Set

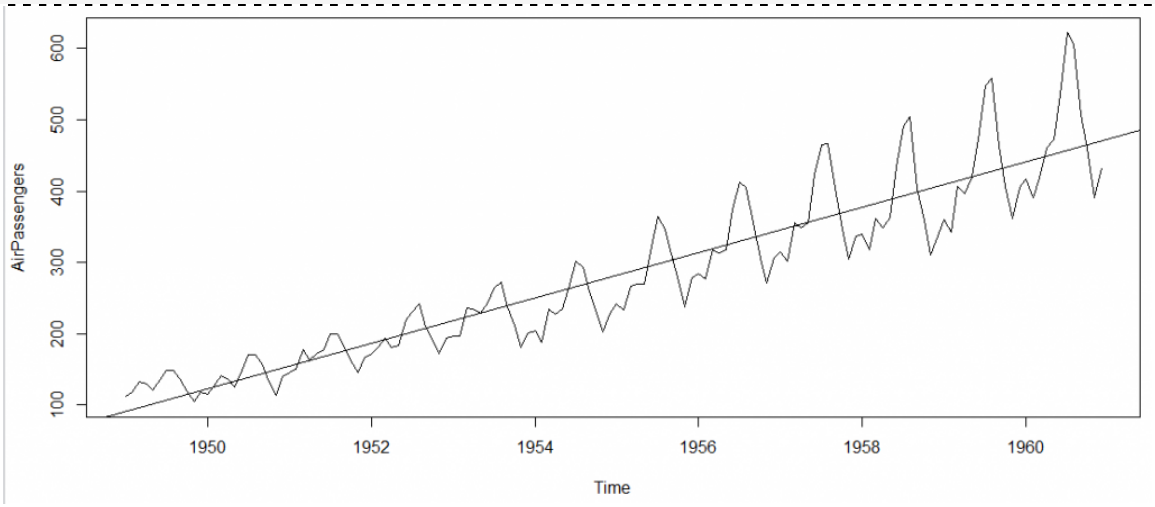
Following is the code which will help you load the data set and spill out a few top level metrics.

```
> data(AirPassengers)
 > class(AirPassengers)
 [1] "ts"
#This tells you that the data series is in a time series format
 > start(AirPassengers)
 [1] 1949 1
#This is the start of the time series
> end(AirPassengers)
 [1] 1960 12
#This is the end of the time series
> frequency(AirPassengers)
 [1] 12
#The cycle of this time series is 12months in a year
 > summary(AirPassengers)
 Min. 1st Qu. Median Mean 3rd Qu. Max.
 104.0 180.0 265.5 280.3 360.5 622.0
```

Detailed Metrics

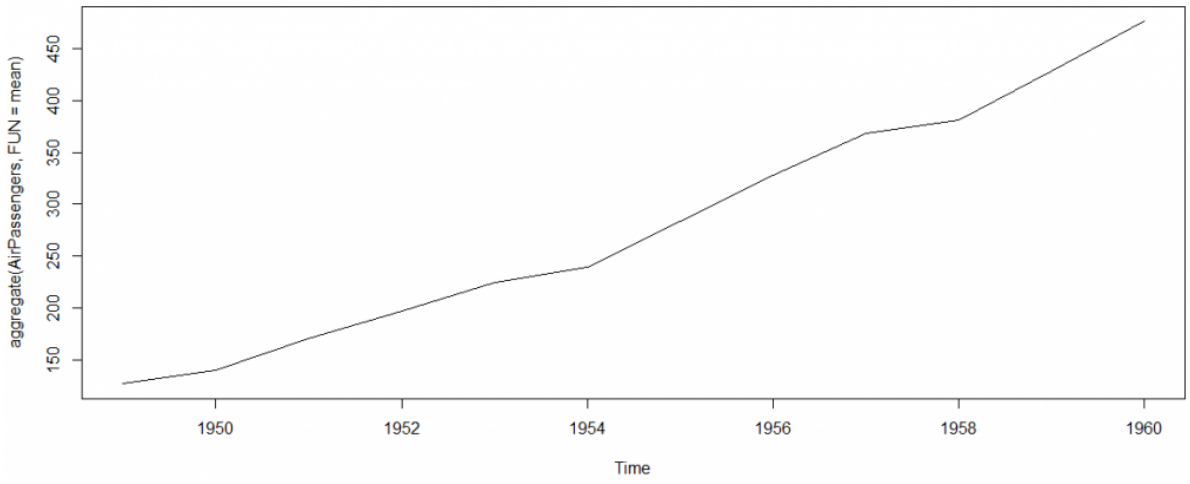
```
#The number of passengers are distributed across the spectrum
> plot(AirPassengers)
```

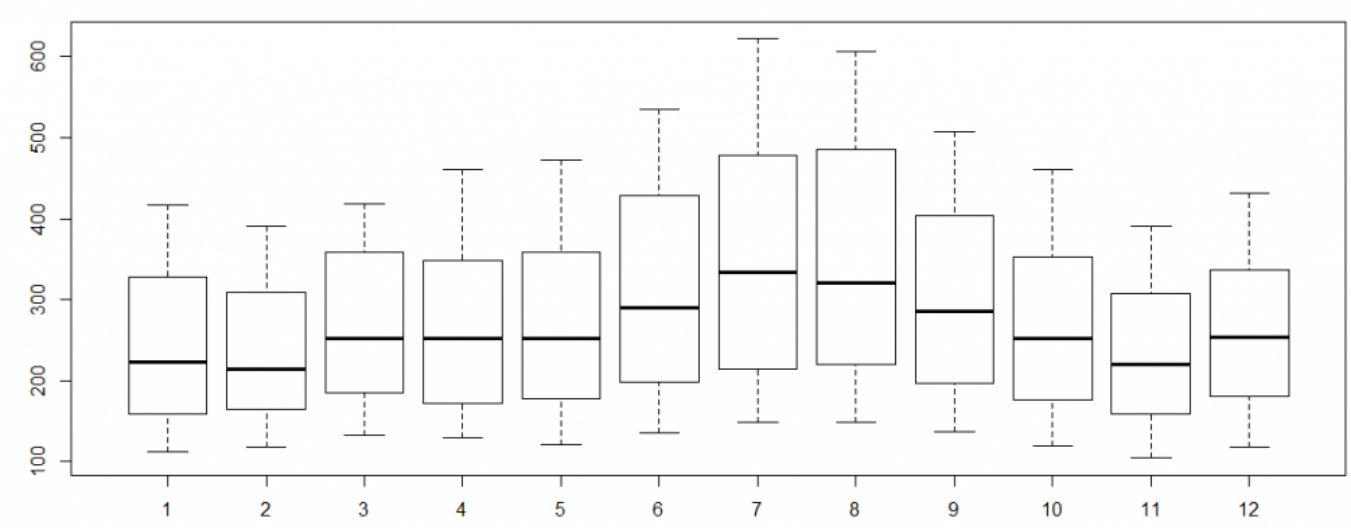
```
#This will plot the time series
>abline(reg=lm(AirPassengers~time(AirPassengers)))
# This will fit in a line
```



Here are a few more operations you can do:

```
> cycle(AirPassengers)
#This will print the cycle across years.
>plot(aggregate(AirPassengers,FUN=mean))
#This will aggregate the cycles and display a year on year trend
> boxplot(AirPassengers~cycle(AirPassengers))
#Box plot across months will give us a sense on seasonal effect
```





Important Inferences

1. The year on year trend clearly shows that the #passengers have been increasing without fail.
2. The variance and the mean value in July and August is much higher than rest of the months.
3. Even though the mean value of each month is quite different their variance is small. Hence, we have strong seasonal effect with a cycle of 12 months or less.

Decomposing Time Series

Decomposing a time series means separating it into its constituent components, which are usually a trend component and an irregular component, and if it is a seasonal time series, a seasonal component.

Decomposing Non-Seasonal Data

To estimate the trend component of a non-seasonal time series that can be described using an additive model, it is common to use a smoothing method, such as calculating the simple moving average of the time series.

The SMA() function in the "TTR" R package can be used to smooth time series data using a simple moving average. To use this function, we first need to install the "TTR" R package (for

instructions on how to install an R package, Once you have installed the “TTR” R package, you can load the “TTR” R package by typing:

```
install.packages(“TTR”)
library("TTR")
SMA3 <- SMA(AirPassengers,n=3)
plot.ts(SMA3)
SMA8 <- SMA(AirPassengers,n=8)
plot.ts(SMA8)
```

Decomposing Seasonal Data

A seasonal time series consists of a trend component, a seasonal component and an irregular component. Decomposing the time series means separating the time series into these three components: that is, estimating these three components.

To estimate the trend component and seasonal component of a seasonal time series that can be described using an additive model, we can use the “decompose()” function in R. This function estimates the trend, seasonal, and irregular components of a time series that can be described using an additive model.

The function “decompose()” returns a list object as its result, where the estimates of the seasonal component, trend component and irregular component are stored in named elements of that list objects, called “seasonal”, “trend”, and “random” respectively.

To estimate the trend, seasonal and irregular components of this time series, we type:

```
> AirPassengerscomp <- decompose(AirPassengers)
```

The estimated values of the seasonal, trend and irregular components are now stored in variables AirPassengerscomp $seasonal, AirPassengerscomp $trend and AirPassengerscomp $random. For example, we can print out the estimated values of the seasonal component by typing:

```
> AirPassengerscomp $seasonal
```

```
plot(AirPassengerscomp)
```