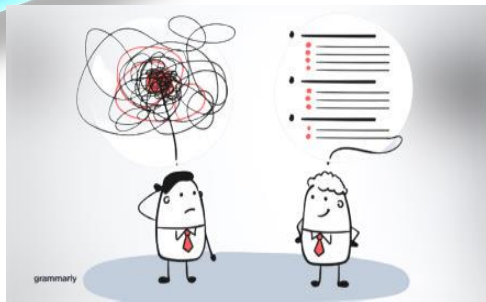
Introduction to R Programming

By-

Aarti Sharma
(MAT/19/28)
Jyoti Bahukhandi
(MAT/19/8)

Outline….

*History & Introduction
*Pros & Cons of R
*Comparison with other languages
*Why learn R?
*What is RStudio?
*R-IDE
*Data Structures
*Data visualization in R
*Boxplot



HISTORY

- R is a programming language , it was an implementation over S language. R was first designed by Ross Ihaka and Robert Gentlemen at the University of Auckland in 1993.

INTRODUCTION

R is a-

- A Programming Language
- A Statistical Package
- An Interpreter
- Open source
- Object Oriented Language

Pros and Cons of R

Benefits

1. Powerful Graphics
2. ML Operations
3. Array of Packages
4. Cross Compatibility

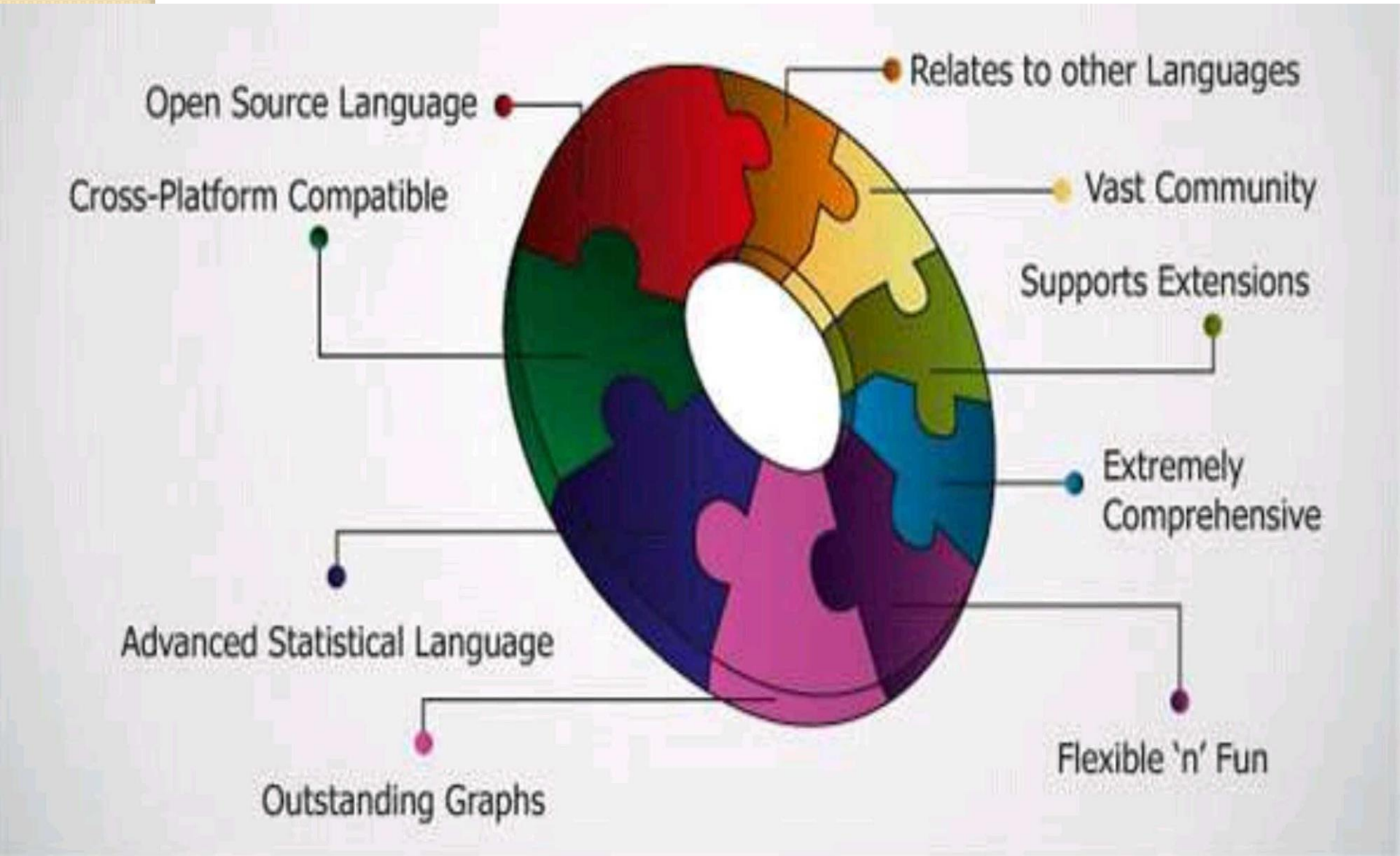
Limitations

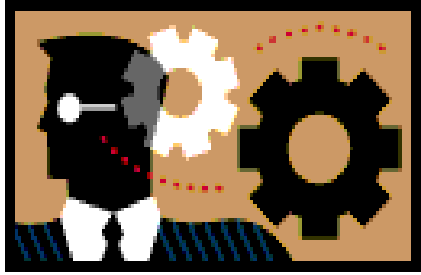
1. Slow
2. Poor Security
3. Low Support

Comparison with other languages

R Programming	Python	Java
•It was stably released in 2014.	•It was stably released in 1996.	•It was stably released in 1995.
•It has more functions and packages.	•It has less functions and packages.	•It has large number of inbuilt functions and packages.
•It is an interpreter base language	•It is an interpreter base language	•It is interpreter and compiled based language.
•It is statistical design and graphics programming language.	•It is general purpose language.	•It is general purpose programming language designed for web applications .
•It is difficult to learn and understand.	•It is easy to understand.	•It is easy to learn and understand.

Why Learn R?


Open Source Language
Relates to other Languages
Cross-Platform Compatible
Vast Community
Supports Extensions
Extremely Comprehensive
Advanced Statistical Language
Flexible 'n' Fun
Outstanding Graphs

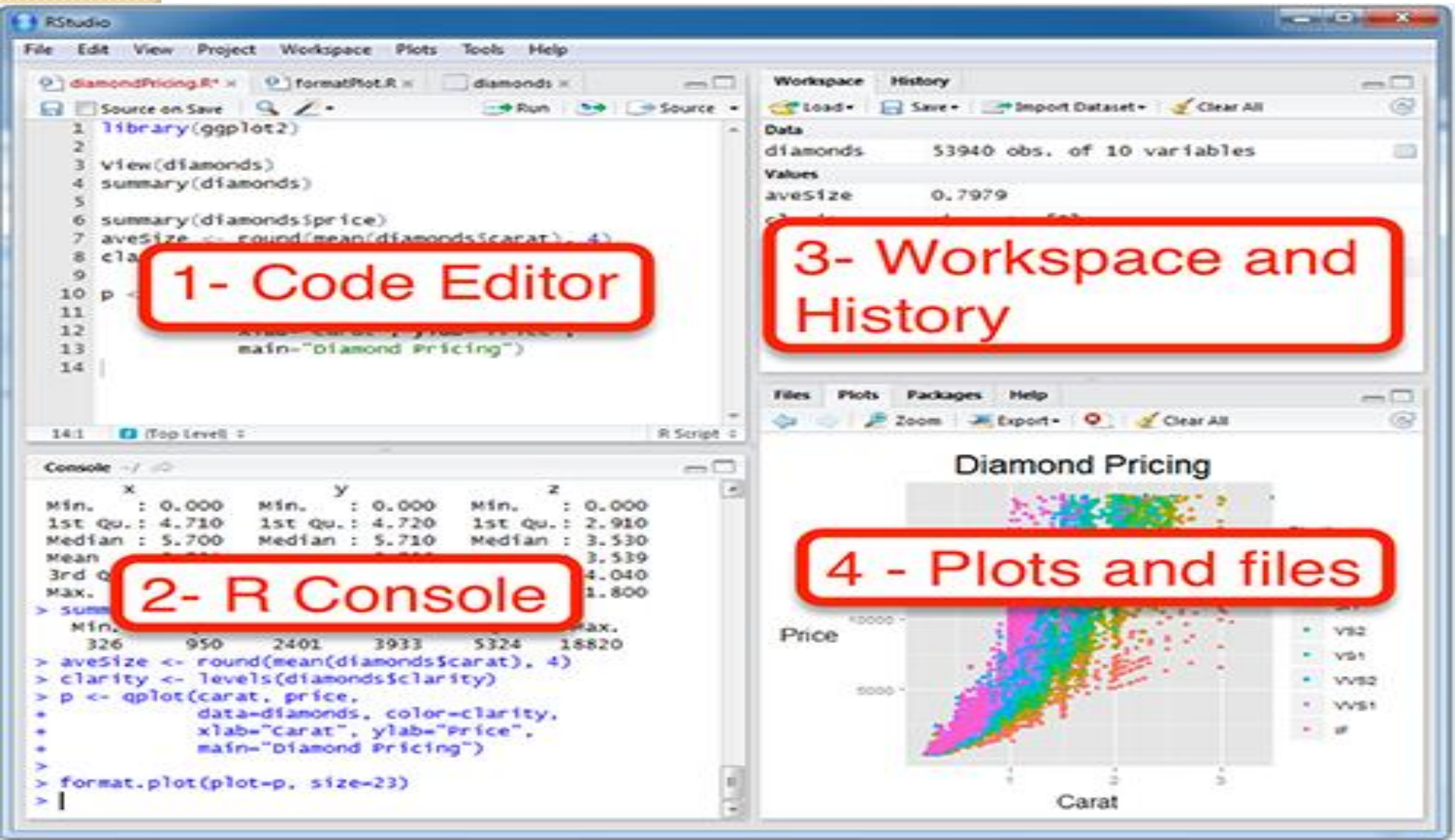


What is RStudio?

RStudio is an integrated development environment (IDE) for R. It includes a *console, syntax-highlighting editor* that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

RStudio is available in **open source and commercial** editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, Red Hat/CentOS, and SUSE Linux).

R-IDE

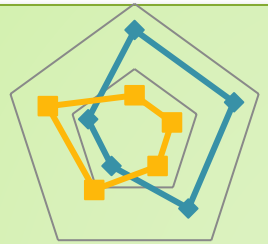

RStudio
File Edit View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save Run Source
1 library(ggplot2)
2
3 view(diamonds)
4 summary(diamonds)
5
6 summary(diamonds$price)
7 aveSize <- round(mean(diamonds$carat), 4)
8 cla
9
10 p <
11
12
13 main="Diamond Pricing")
14
14:1 (Top Level) R Script
1- Code Editor
Workspace History
Load Save Import Dataset Clear All
Data
diamonds 53940 obs. of 10 variables
Values
aveSize 0.7979
3- Workspace and History
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
Carat
4 - Plots and files
Console
x y z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean 3.539
3rd Q 4.040
Max. 1.800
> summ
Min. Max.
326 950 2401 3933 5324 18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+ data=diamonds, color=clarity,
+ xlab="Carat", ylab="Price",
+ main="Diamond Pricing")
>
> format.plot(plot=p, size=23)
>
2- R Console

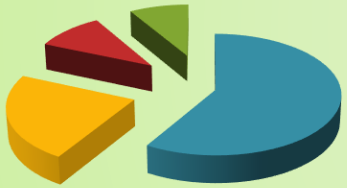
Data Structures	Definations	Examples
1. Vectors	It is 1-D & consists of same type of objects. 5 classes of objects:- *Characters *Logical *Numeric *Integers *Complex Nos.	* y<-c("ram","sita") *class(y)-Characters* [1] ram sita * w<-c(TRUE, FALSE) *class(w)-Logical* [1] TRUE FALSE * x<-c(1,3,7,2) *class(x)-Numeric* [1] 1 3 7 2 * z<-(7L,4L) *class(z)-Integers* [1] 7 4 * v<-c(3+3i,3+9i) *class(v)-complex* [1] 3+3i 3+9i
2. Matrices	It is 2-D with same type of objects.	* X<- matrix(c(1,2,3,4,5,6),nrow=2,ncol=3); print(X) Output:- [,1] [,2] [,3] [1,] 1 3 5 [2,] 2 4 6 * matrix(1,nrow=50,ncol=50) Output:- creates a 50×50 matrix, every entry of which is 1

3. List	It contains a heterogeneous selection of objects.	l<-list(x=c(2,1,3),y="ram", z =TRUE) l Output:- $x [1] 2 1 3 $y [1] "ram" $z [1] TRUE
4. Data Frame	Useful way to represent tabular data. Essentially a matrix with named columns may also include non-numerical variables.	df=data.frame(a=c(1,2,3,4,5),b=c(2 ,3,4,5,6)) Print(df) Output:- a b 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6

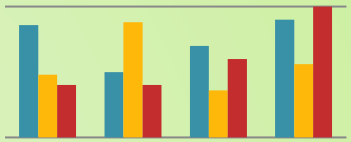
Data Visualization in R

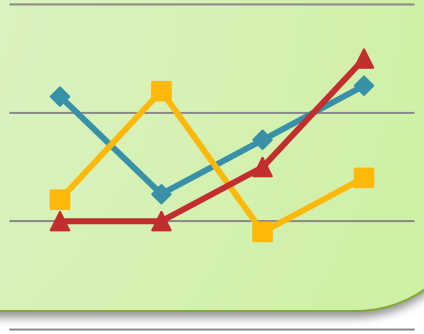
R supports various types of graphics including:-





- Histogram
- Bar Chart
- Pie Chart
- Dot Chart
- Line Chart
- Box Plot
- Scatter Plot

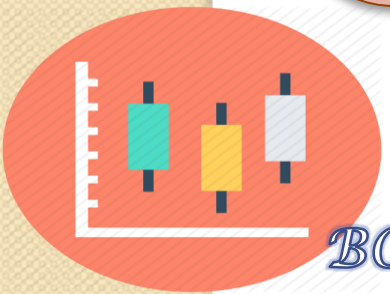




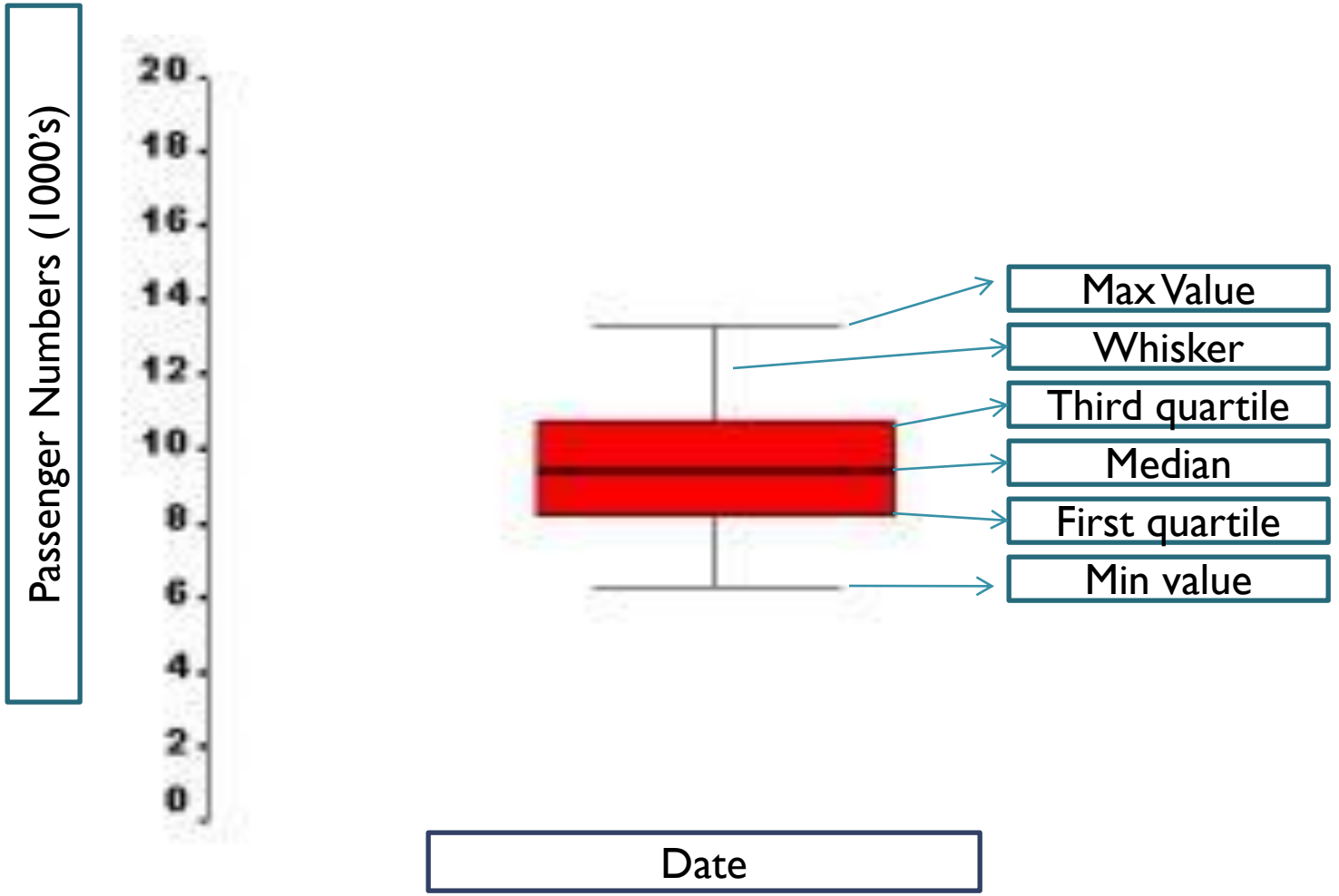
Let's look at box plots:-

Also known as whisker diagrams, boxplots display the distribution of data based on:-

Minimum
First Quartile
Median
Third Quartile
Maximum



To create a boxplot, we simply give: Boxplot(data)




Thank you!