DATA ANALYTICS HANDS ON SESSION

UE20CS312

WEEK 1

GAURAV DNYANESH MAHAJAN PES1UG20CS150 Section C

Basics of R

R COMMAND LINE

```
> 22+32
[1] 54
> 2**3
[1] 8
> 2*3
[1] 6
> 2/3
[1] 0.6666667
> 7%%4
[1] 3
> 2^3
[1] 8
> 
```

```
> a<-15
> print(a)
[1] 15
> print(class(a))
[1] "numeric"
> b<-"HELLO,WELCOME TO DA CLASS"
> cat("b's value: ",b,"b's class : ",class(b))
b's value:  HELLO,WELCOME TO DA CLASS b's class :  character
> c<-FALSE
> cat("c's value : ",c,"c's class : ",class(c))
c's value :  FALSE c's class :  logical
```

Vectors and Sequences

```
> vector_a<-c(10,20,30,40) #numeric vector
> cat("vector_a : ",vector_a," vector_a's class : ",class(vector_a),
+     " length of vector_a : ",length(vector_a),"\n")
vector_a :  10 20 30 40  vector_a's class :  numeric  length of vector_a :  4
> sequence_a <- seq(4,15)
> print(sequence_a)
 [1]  4  5  6  7  8  9 10 11 12 13 14 15
```

Loops and Conditional Statements

```
R 4.2.0 · ~/
> a<-seq(1,10)
> for(digit in a){print(digit)}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
>
```

```r
a<-seq(11,20)
i<-1
while(i<=length(a)){
  print(a[i])
  i<-i+1
  }
```

```
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
[1] 20
```

```
> a<-37
> if(a%%2){
+     print("Number is odd")
+ }else{
+     print("Number is even")
+ }
[1] "Number is odd"
> a<-10
> ifelse(a%%2,"Number is odd","Number is even")
[1] "Number is even"
>
```

Functions in R

```
> isEven <-function(a){
+     if(a%%2){
+         print("Number is odd")
+     }else{
+         print("Number is even")
+     }
+ }
> isEven(7)
[1] "Number is odd"
> isEven(24)
[1] "Number is even"
>
```

Installing and loading a package

```
> library(ggplot2)
Use suppressPackageStartupMessages() to eliminate package startup
messages
> search()
 [1] ".GlobalEnv"        "package:ggplot2"   "tools:rstudio"
 [4] "package:stats"     "package:graphics"  "package:grDevices"
 [7] "package:utils"     "package:datasets"  "package:methods"
[10] "Autoloads"         "org:r-lib"         "package:base"
> |
```

Dataframes and Visualization

```
> df <- txhousing
> head(df)
# A tibble: 6 x 9
  city     year month sales   volume median listings inventory  date
  <chr>   <int> <int> <dbl>    <dbl>  <dbl>    <dbl>     <dbl> <dbl>
1 Abilene  2000     1    72  5380000  71400      701       6.3 2000
2 Abilene  2000     2    98  6505000  58700      746       6.6 2000.
3 Abilene  2000     3   130  9285000  58100      784       6.8 2000.
4 Abilene  2000     4    98  9730000  68600      785       6.9 2000.
5 Abilene  2000     5   141 10590000  67300      794       6.8 2000.
6 Abilene  2000     6   156 13910000  66900      780       6.6 2000.
> tail(df)
# A tibble: 6 x 9
  city          year month sales   volume median listings inventory  date
  <chr>        <int> <int> <dbl>    <dbl>  <dbl>    <dbl>     <dbl> <dbl>
1 Wichita Falls  2015     2   100 11646765  94000      795       6.8 2015.
2 Wichita Falls  2015     3   152 16716584  89200      818       6.8 2015.
3 Wichita Falls  2015     4   129 15482194 105300      760       6.4 2015.
4 Wichita Falls  2015     5   174 19188181 100000      776       6.4 2015.
5 Wichita Falls  2015     6   143 18820752 118800      770       6.2 2015.
6 Wichita Falls  2015     7   172 23850905 116700      811       6.5 2016.
> |
```

Basic Operations

```
> colnames(df)
[1] "city"      "year"      "month"     "sales"     "volume"    "median"
[7] "listings"  "inventory" "date"
> dim(df)
[1] 8602    9
> top5 <- df[1:5,]
> top5
# A tibble: 5 x 9
  city     year month sales   volume median listings inventory  date
  <chr>   <int> <int> <dbl>    <dbl>  <dbl>    <dbl>     <dbl> <dbl>
1 Abilene  2000     1    72  5380000  71400      701       6.3 2000
2 Abilene  2000     2    98  6505000  58700      746       6.6 2000.
3 Abilene  2000     3   130  9285000  58100      784       6.8 2000.
4 Abilene  2000     4    98  9730000  68600      785       6.9 2000.
5 Abilene  2000     5   141 10590000  67300      794       6.8 2000.
> cities <- df$city
> cities2 <- df["city"]
> cities[1:10]
 [1] "Abilene" "Abilene" "Abilene" "Abilene" "Abilene" "Abilene" "Abilene"
 [8] "Abilene" "Abilene" "Abilene"
> head(cities2)
# A tibble: 6 x 1
  city
  <chr>
1 Abilene
2 Abilene
3 Abilene
4 Abilene
5 Abilene
6 Abilene
>
```

Preliminary Analysis

```
> mean(df$sales,na.rm=TRUE)
[1] 549.5646
> median(df$sales,na.rm=TRUE)
[1] 169
> min(df$sales,na.rm=TRUE)
[1] 6
> max(df$sales,na.rm=TRUE)
[1] 8945
>
```

Calculating Summary

```
> summary(df)
     city                year          month            sales
 Length:8602        Min.   :2000   Min.   : 1.000   Min.   :   6.0
 Class :character   1st Qu.:2003   1st Qu.: 3.000   1st Qu.:  86.0
 Mode  :character   Median :2007   Median : 6.000   Median : 169.0
                    Mean   :2007   Mean   : 6.406   Mean   : 549.6
                    3rd Qu.:2011   3rd Qu.: 9.000   3rd Qu.: 467.0
                    Max.   :2015   Max.   :12.000   Max.   :8945.0
                                                    NA's   :568
     volume              median          listings       inventory
 Min.   :8.350e+05   Min.   : 50000   Min.   :    0   Min.   : 0.000
 1st Qu.:1.084e+07   1st Qu.:100000   1st Qu.:  682   1st Qu.: 4.900
 Median :2.299e+07   Median :123800   Median : 1283   Median : 6.200
 Mean   :1.069e+08   Mean   :128131   Mean   : 3217   Mean   : 7.175
 3rd Qu.:7.512e+07   3rd Qu.:150000   3rd Qu.: 2954   3rd Qu.: 8.150
 Max.   :2.568e+09   Max.   :304200   Max.   :43107   Max.   :55.900
 NA's   :568         NA's   :616      NA's   :1424    NA's   :1467
      date
 Min.   :2000
 1st Qu.:2004
 Median :2008
 Mean   :2008
 3rd Qu.:2012
 Max.   :2016
```

Sorting a Dataframe

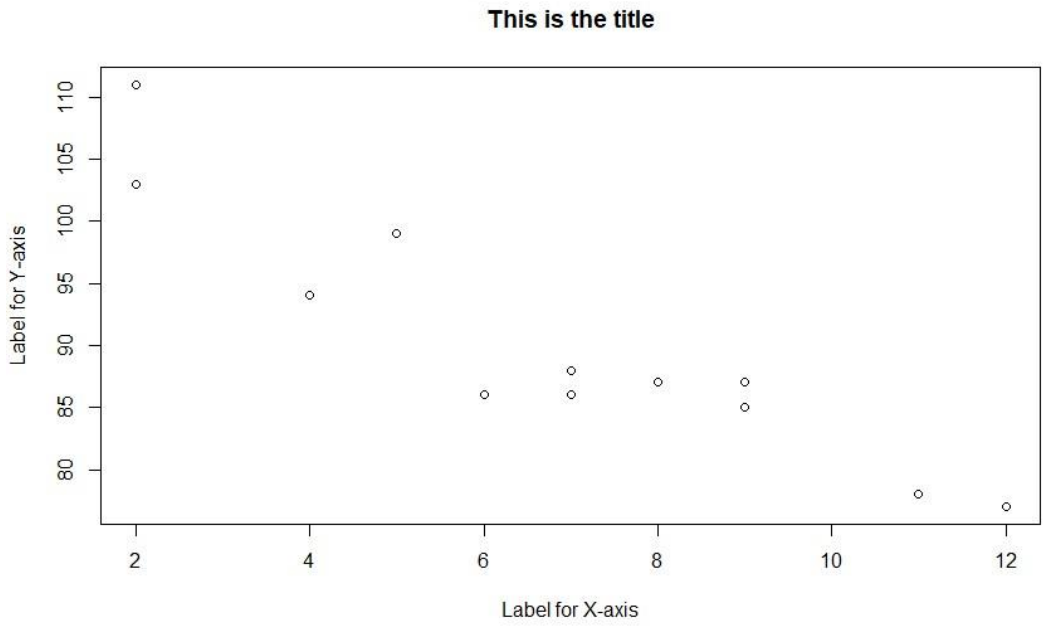
```
> sortdf <- df[order(df$sales, decreasing = TRUE),]
> head(sortdf)
# A tibble: 6 x 9
  city     year month sales     volume median listings inventory  date
  <chr>   <int> <int> <dbl>      <dbl>  <dbl>    <dbl>     <dbl> <dbl>
1 Houston  2015     7  8945 2568156780 217600    23875       3.4 2016.
2 Houston  2006     6  8628 1795898108 155200    36281       5.6 2006.
3 Houston  2013     7  8468 2168720825 187800    21497       3.3 2014.
4 Houston  2015     6  8449 2490238594 222400    22311       3.2 2015.
5 Houston  2013     5  8439 2121508529 186100    20526       3.3 2013.
6 Houston  2014     6  8391 2342443127 211200    19725       2.9 2014.
>
```

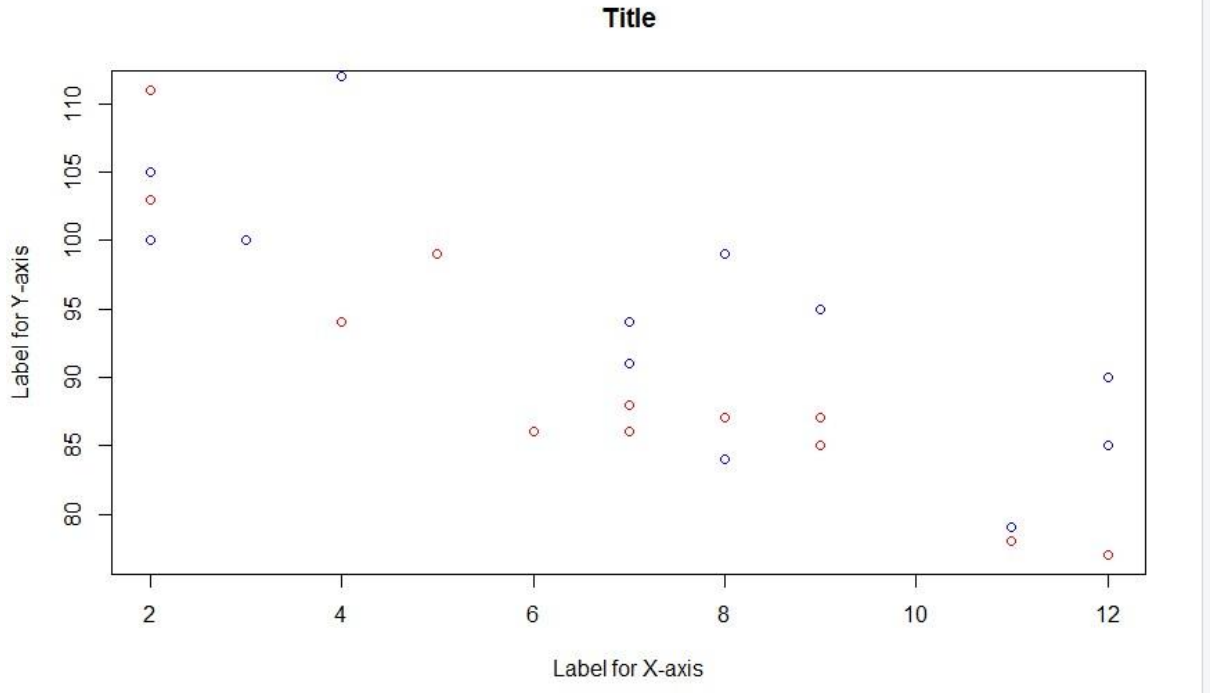
Filtering a Dataframe

```
> houston_data <- df[df$city=="Houston",]
> head(houston_data)
# A tibble: 6 x 9
  city     year month sales    volume median listings inventory  date
  <chr>   <int> <int> <dbl>     <dbl>  <dbl>    <dbl>     <dbl> <dbl>
1 Houston  2000     1  2653 381805283 102500    16768       3.9 2000
2 Houston  2000     2  3687 536456803 110300    16933       3.9 2000.
3 Houston  2000     3  4733 709112659 109500    17058       3.9 2000.
4 Houston  2000     4  4364 649712779 110800    17716       4.1 2000.
5 Houston  2000     5  5215 809459231 112700    18461       4.2 2000.
6 Houston  2000     6  5655 887396592 117900    18959       4.3 2000.
>
```

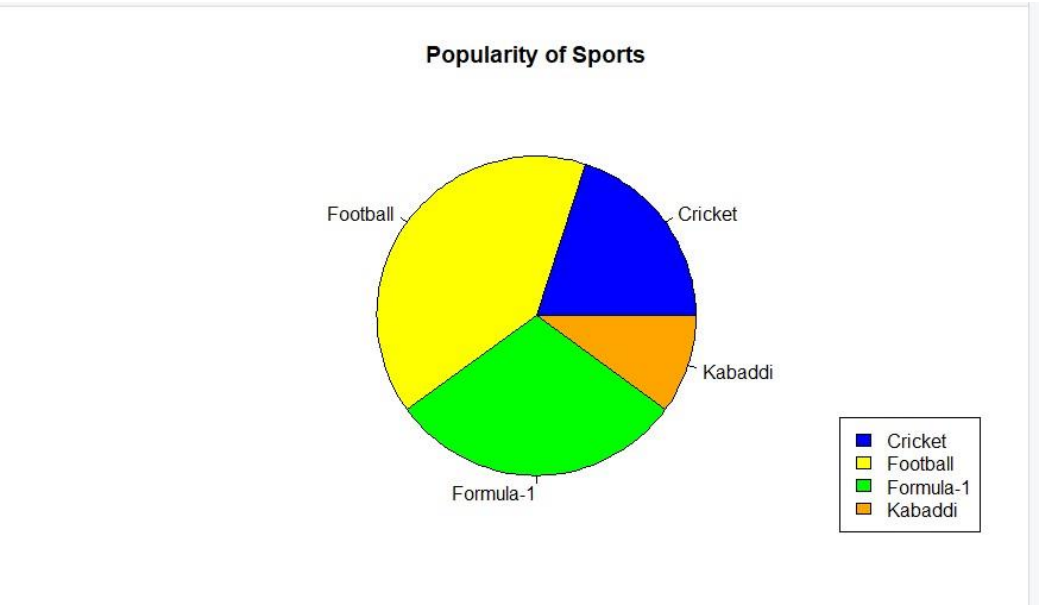
Visualization

Scatter Plot

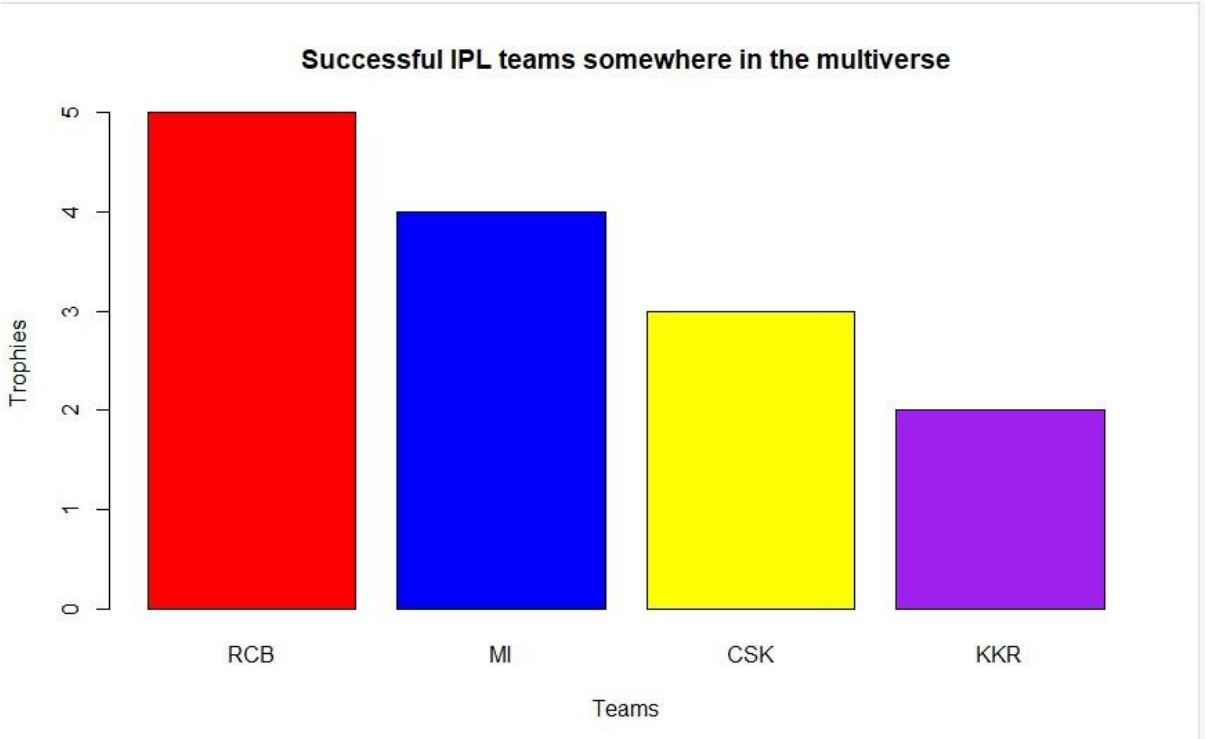




Pie Chart


Popularity of Sports
Football
Cricket
Kabaddi
Formula-1
Cricket
Football
Formula-1
Kabaddi

Bar Plot


Successful IPL teams somewhere in the multiverse
Trophies
0
1
2
3
4
5
RCB
MI
CSK
KKR
Teams