# DATA VISUALIZATION PRACTICAL REPORT 1-9

## Software's used:-

**R Studio**

**Mongo dB**

**Tableau**

**Name: Shreyas Patil**

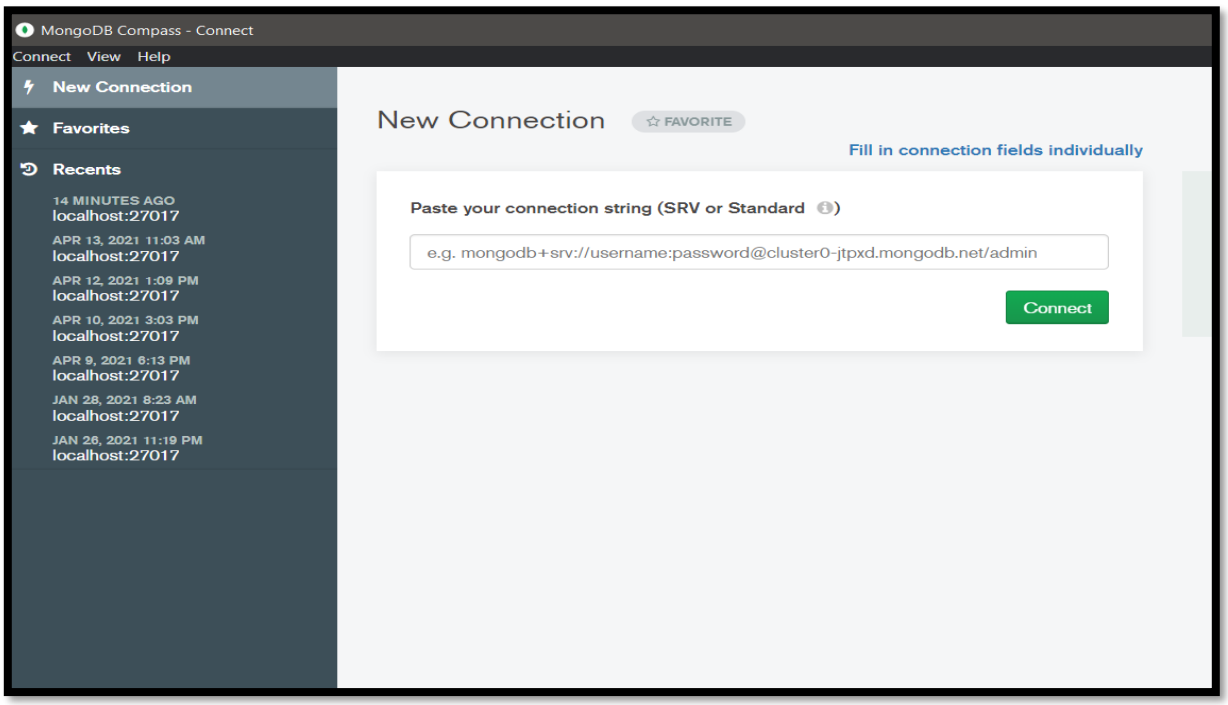
**UID: 185036**

**Roll No: 49**

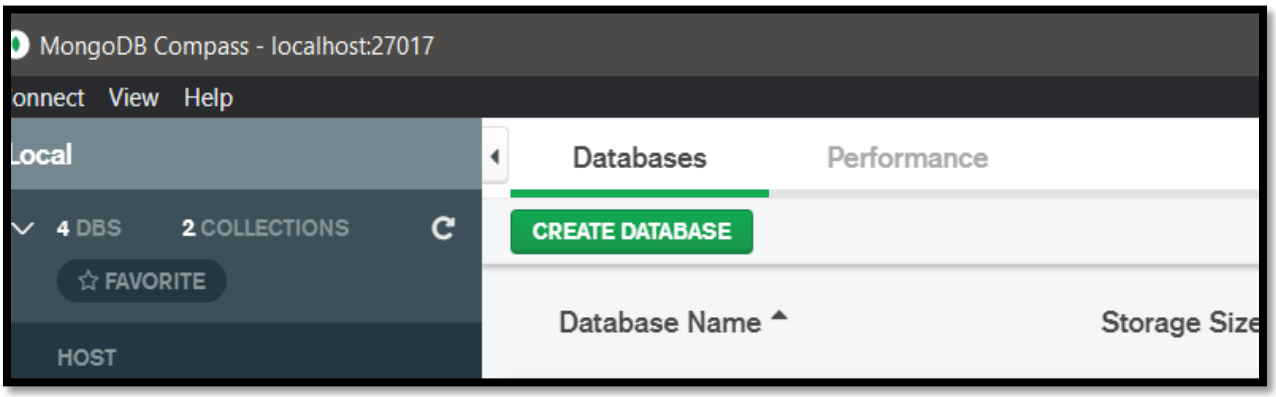
# PRACTICAL 1

**AIM:-** To load Chicago Crimes Dataset into MongoDB Compass and connect it with R Studio.

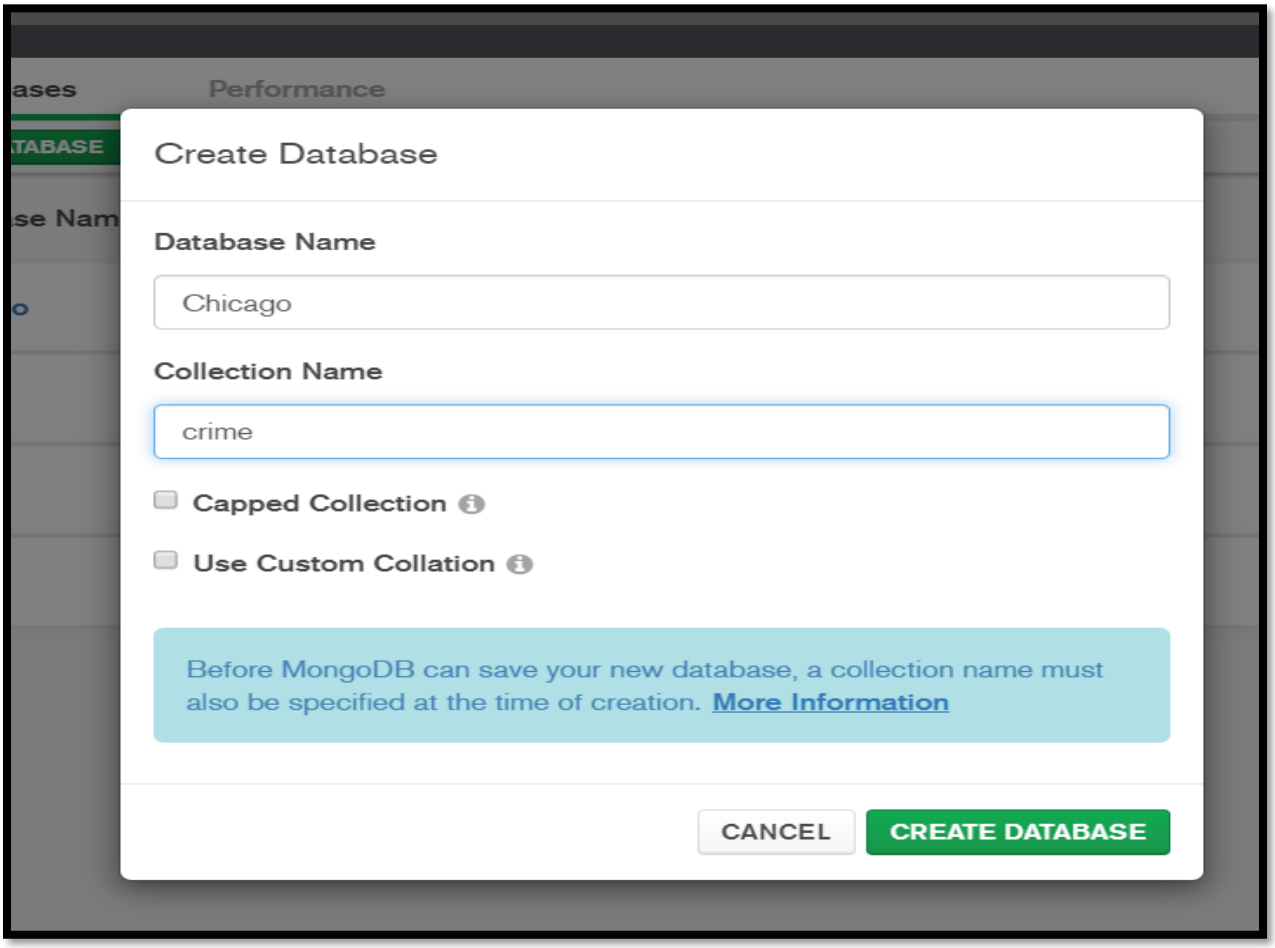
Step 1:



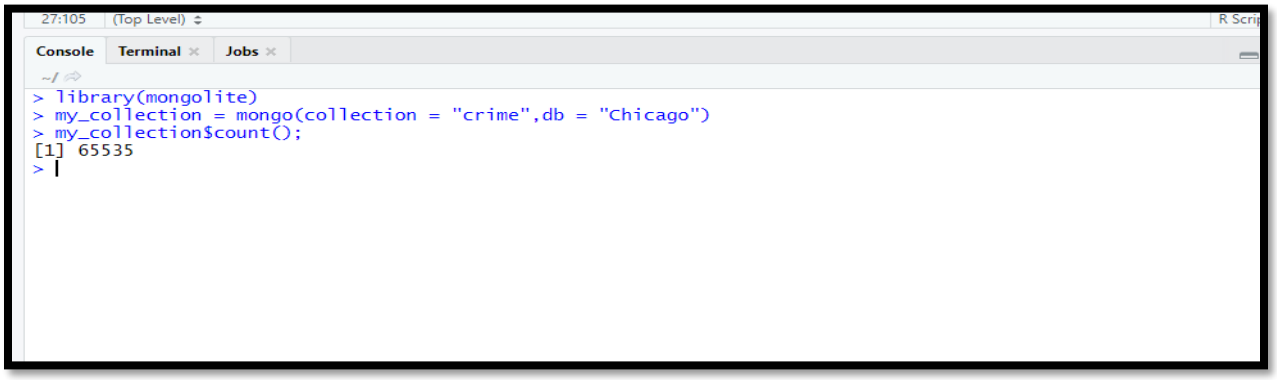

Step 2:



Step 3:


ases
Performance
TABASE
se Nam
Create Database
Database Name
Chicago
Collection Name
crime
Capped Collection
Use Custom Collation
Before MongoDB can save your new database, a collection name must also be specified at the time of creation. More Information
CANCEL
CREATE DATABASE

# Implementation on R/Connection in R



# PRACTICAL 2

## PACKAGES REQUIRED :

1. mongolite

2. dplyr

3 ggplot2

## ALGORITHM:

1. Group the data according to location

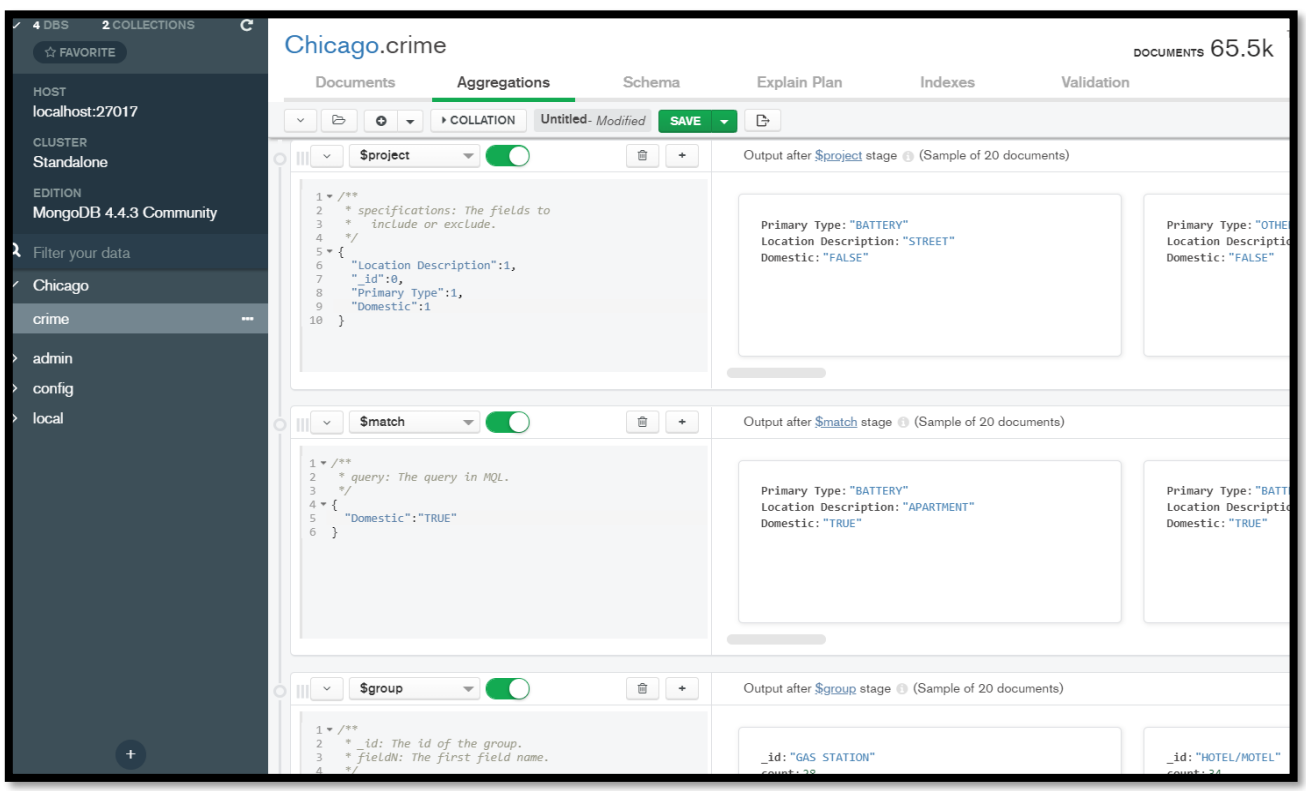
2. Remove null values if any

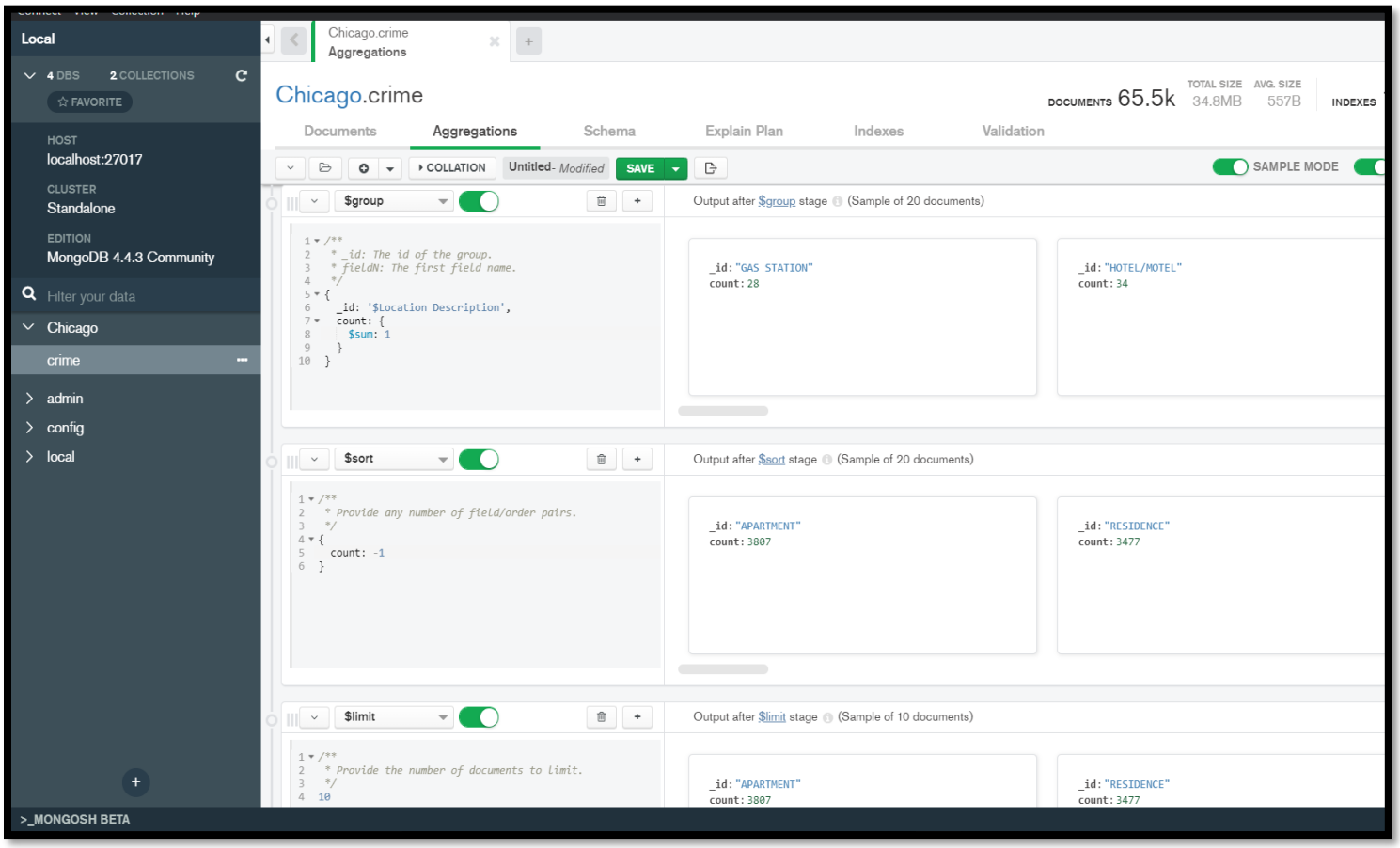
3. Arrange data

4. Sort top 10

5. Put in descending order

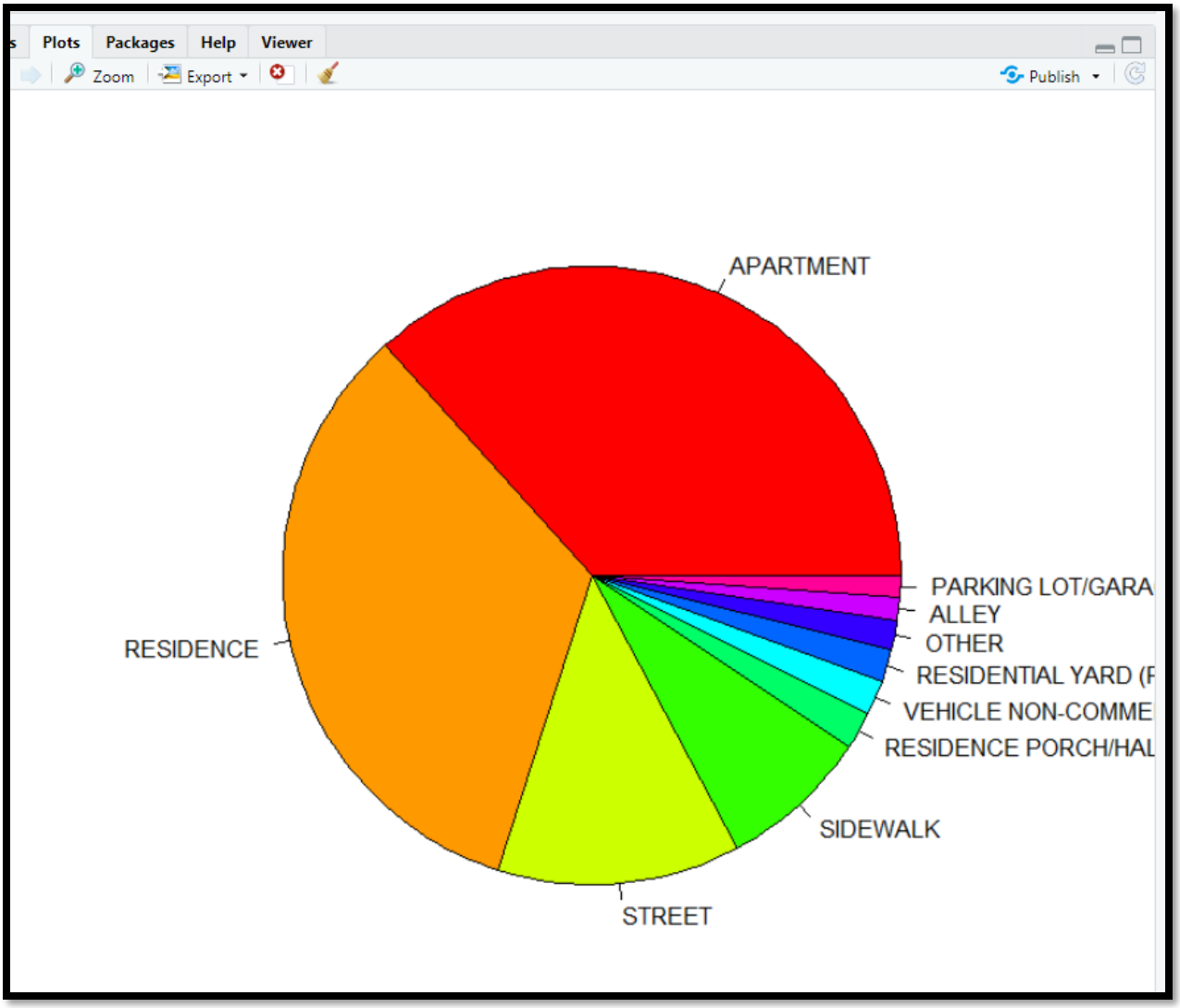
## Implementation on MongoDb






## Implementation on R studio

```
> library(mongolite)
> library(dplyr)
> my_collection = mongo(collection = "crime",db = "Chicago");
> data =  my_collection$aggregate('[{"$match":{"Domestic":"TRUE"}},{"$group":{"_id":"$Location Description", "c
ount": {"$sum":1}}}]')%>%na.omit()%>%arrange(desc(count))%>%head(10)
> data
                            _id count
1                     APARTMENT  3807
2                     RESIDENCE  3477
3                        STREET  1317
4                      SIDEWALK   813
5         RESIDENCE PORCH/HALLWAY   212
6          VEHICLE NON-COMMERCIAL   183
7   RESIDENTIAL YARD (FRONT/BACK)   177
8                         OTHER   161
9                         ALLEY   124
10 PARKING LOT/GARAGE(NON.RESID.)   119
> pie(data$count,data$'_id', col = rainbow(count(data)), radius = 0.9)
>
```


Plots
Packages
Help
Viewer
Zoom
Export
Publish
APARTMENT
PARKING LOT/GARA
ALLEY
OTHER
RESIDENTIAL YARD (F
VEHICLE NON-COMME
RESIDENCE PORCH/HAL
SIDEWALK
STREET
RESIDENCE

# PRACTICAL 3

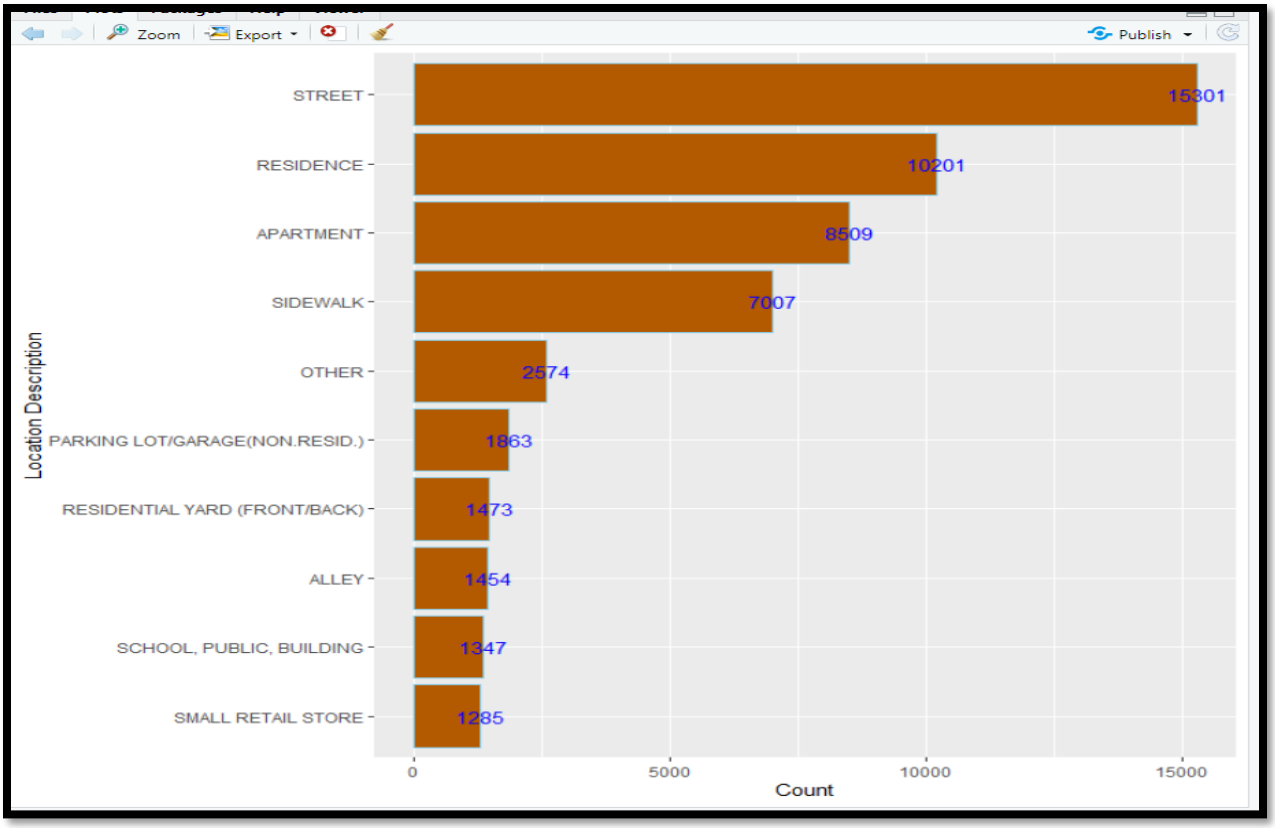
**PACKAGES REQUIRED** :

1. mongolite
2. dplyr
3. ggplot2

**PART 1:** Plot a sideways bar graph of top 10 Location Description where crimes took place based on each of their count using the Chicago Crime dataset.

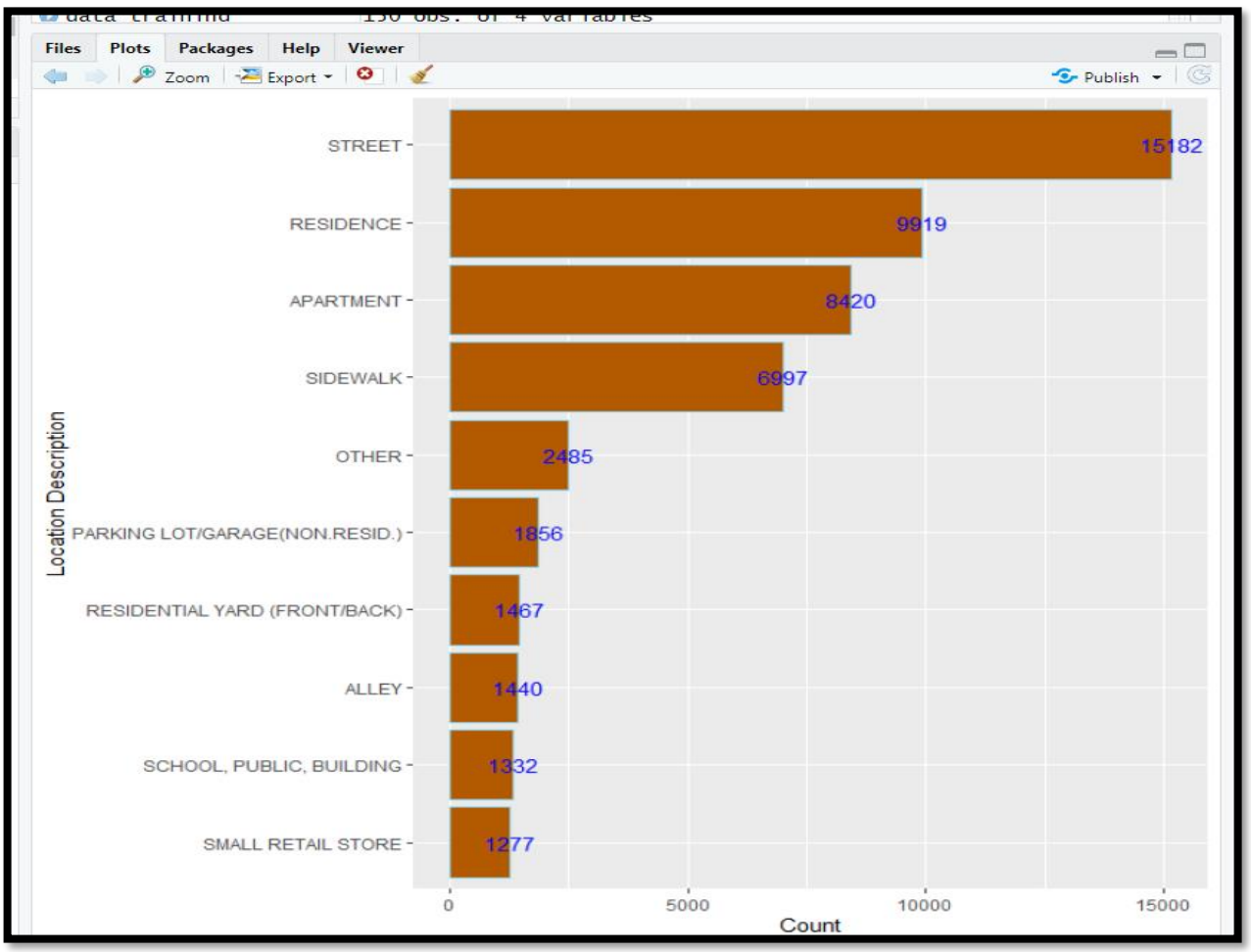
**Implementation on R Studio**

```
30:192  (Top Level) ÷                                                    R Script ÷
Console   Terminal ×   Jobs ×
~/
> library(mongolite)
> library(dplyr)
> library(ggplot2)
Warning message:
package 'ggplot2' was built under R version 3.6.3
> my_collection = mongo(collection = "crime",db = "Chicago");
> my_collection$aggregate('[{"$group":{"_id":"$Location Description","Count":{"$sum:1}}}]')%>%na.omit()arrange
(desc(Count))%>%head(10)
Error: unexpected symbol in "my_collection$aggregate('[{"$group":{"_id":"$Location Description","Count":{"$sum:
1}}}]')%>%na.omit()arrange"
> my_collection$aggregate('[{"$group":{"_id":"$Location Description","Count":{"$sum:1}}}]')%>%na.omit()%>%arran
ge(desc(Count))%>%head(10)
Error: Invalid JSON object: [{"$group":{"_id":"$Location Description","Count":{"$sum:1}}}]
> my_collection$aggregate('[{"$group":{"_id":"$Location Description","Count":{"$sum":1}}}]')%>%na.omit()%>%arra
nge(desc(Count))%>%head(10)
                            _id Count
1                        STREET 15301
2                     RESIDENCE 10201
3                     APARTMENT  8509
4                      SIDEWALK  7007
5                         OTHER  2574
6  PARKING LOT/GARAGE(NON.RESID.)  1863
7   RESIDENTIAL YARD (FRONT/BACK)  1473
8                         ALLEY  1454
9        SCHOOL, PUBLIC, BUILDING  1347
10             SMALL RETAIL STORE  1285
> my_collection$aggregate('[{"$group":{"_id":"$Location Description","Count":{"$sum":1}}}]')%>%na.omit()%>%arra
nge(desc(Count))%>%head(10)%>%ggplot(aes(x=reorder(`_id`,Count),y=Count))+geom_bar(stat="identity",color='skybl
ue',fill='#b35900')+geom_text(aes(label = Count), color = "blue") +coord_flip()+xlab("Location Description")
> |
```


Zoom
Export
Publish
STREET
RESIDENCE
APARTMENT
SIDEWALK
OTHER
PARKING LOT/GARAGE(NON.RESID.)
RESIDENTIAL YARD (FRONT/BACK)
ALLEY
SCHOOL, PUBLIC, BUILDING
SMALL RETAIL STORE
Location Description
15301
10201
8509
7007
2574
1863
1473
1454
1347
1285
0
5000
10000
15000
Count

**<u>PART 2:</u>** Plot a sideways bar graph of top 10 Location Description of **2015** where crimes took place based on each of their count using the Chicago Crime dataset.

```
> data=my_collection$aggregate('[{"$match":{"Year":"2015"}},{"$group":{"_id":"$Location Description","Count":{"$sum":1}}}]')%>%na.omit()%>%arrange(desc(Count))%>%he
ad(10)%>%ggplot(aes(x=reorder(`_id`,Count),y=Count))+geom_bar(stat="identity",color='skyblue',fill='#b35900')+geom_text(aes(label = Count), color = "blue") +coord_f
lip()+xlab("Location Description")
> data
>
```



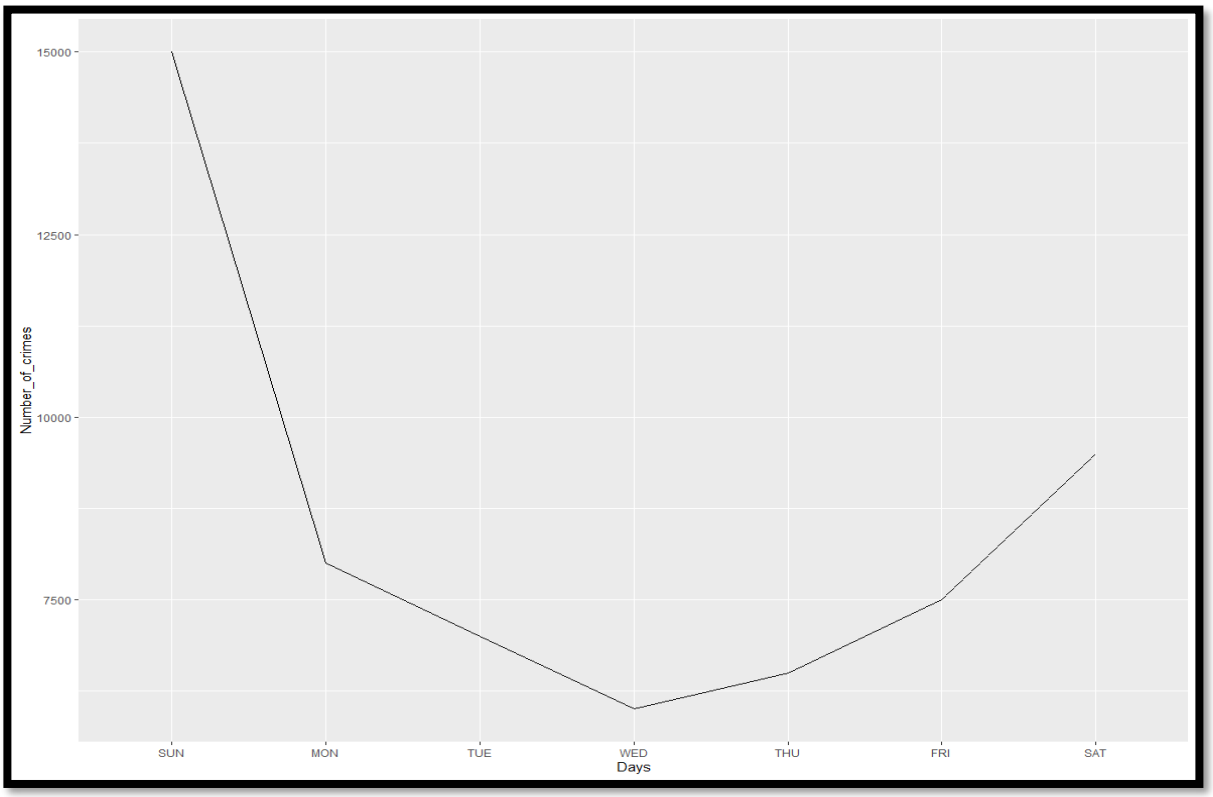
# PRACTICAL 4

## PACKAGES REQUIRED :

1. mongolite
2. ggplot2
3. lubridate

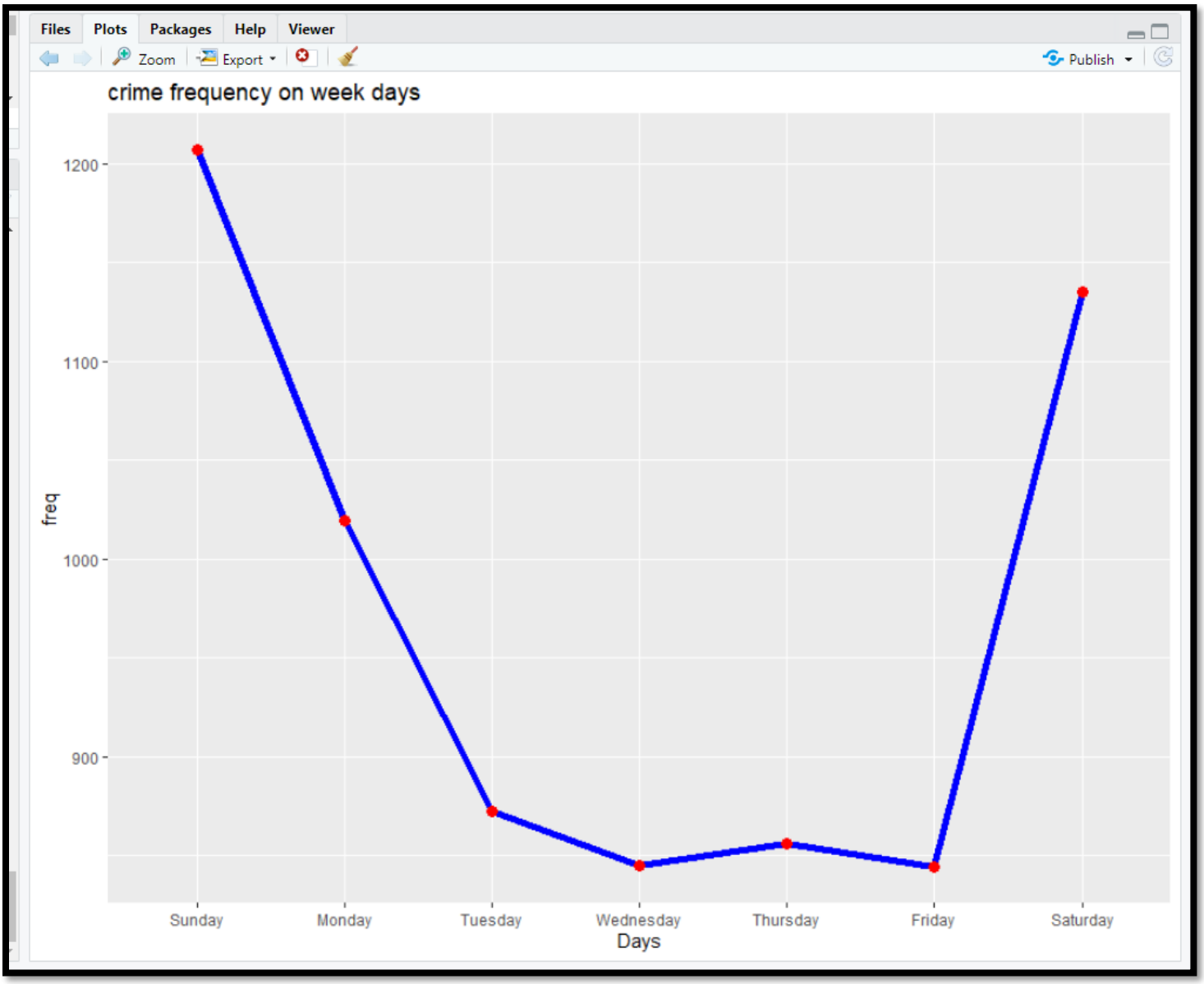
**AIM:** Plot a line graph of crimes that took place on each weekday for type domestic based on their count using the Chicago Crime dataset.

```
> library(lubridate)
> library(ggplot2)
> library(dplyr)
> library(mongolite)
> a=data.frame(Days = c("SUN","MON","TUE","WED","THU","FRI","SAT"),
+ Number_of_crimes = c(15000,8000,7000,6000,6500,7500,9500))
> a$Days<-factor(a$Days,levels=a$Days)
> ggplot(data=a,aes(x=Days,y=Number_of_crimes,group=1))+
+ geom_line()+
+ geom_point()
> ggplot(data=a,aes(x=Days,
+ y=Number_of_crimes,group=1))+geom_line()
>
```



```
54
44:5  (Top Level)
Console  Terminal  Jobs
~/
> library(mongolite)
> library(lubridate)
> library(ggplot2)
> my_collection = mongo(collection = "crime",db="Chicago")
> data = my_collection$find('{"Domestic":"TRUE"}',fields = '{"Domestic":1,"Date":1,"_id":0}')
> data$Date = mdy_hms(data$Date)
Warning message:
 4139 failed to parse.
> data$weekday = weekdays(data$Date)
> data
                    Date Domestic   weekday
1   2015-03-18 22:45:00     TRUE Wednesday
2   2015-03-18 23:00:00     TRUE Wednesday
3   2015-03-18 21:35:00     TRUE Wednesday
4   2015-03-15 16:10:00     TRUE    Sunday
5   2015-03-18 23:30:00     TRUE Wednesday
6   2015-03-18 22:45:00     TRUE Wednesday
7   2015-03-19 02:00:00     TRUE  Thursday
8   2015-03-19 01:00:00     TRUE  Thursday
9   2015-03-19 02:30:00     TRUE  Thursday
10  2015-03-19 03:00:00     TRUE  Thursday
11  2015-03-19 03:51:00     TRUE  Thursday
12  2015-03-19 00:01:00     TRUE  Thursday
13  2015-03-18 22:30:00     TRUE Wednesday
14  2015-03-19 04:38:00     TRUE  Thursday
15  2015-03-18 17:00:00     TRUE Wednesday
16  2015-03-18 12:20:00     TRUE Wednesday
17  2015-03-19 05:30:00     TRUE  Thursday
18  2015-03-18 12:50:00     TRUE Wednesday
19  2015-03-18 21:15:00     TRUE Wednesday
```

```
~/
321 2015-03-22 04:19:00     TRUE    Sunday
322 2015-03-22 04:45:00     TRUE    Sunday
323 2015-03-22 05:15:00     TRUE    Sunday
324 2015-03-19 23:30:00     TRUE  Thursday
325 2015-03-20 18:30:00     TRUE    Friday
326 2015-03-22 01:40:00     TRUE    Sunday
327 2015-03-22 05:50:00     TRUE    Sunday
328 2015-03-21 14:15:00     TRUE  Saturday
329 2015-03-22 08:20:00     TRUE    Sunday
330 2015-03-22 06:45:00     TRUE    Sunday
331 2015-03-22 09:00:00     TRUE    Sunday
332 2015-03-22 09:20:00     TRUE    Sunday
333 2015-03-22 07:50:00     TRUE    Sunday
 [ reached 'max' / getOption("max.print") -- omitted 10584 rows ]
> count = as.data.frame(table(data$weekday))
> count
       Var1 Freq
1    Friday  844
2    Monday 1019
3  Saturday 1135
4    Sunday 1207
5  Thursday  856
6   Tuesday  872
7 Wednesday  845
> test = data.frame(Days= factor(c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"),levels = c("Sunday","Monday","Tuesday","Wednesday","Thurs
day","Friday","Saturday")),freq = c(1207,1019,872,845,856,844,1135))
>  test
       Days freq
1    Sunday 1207
2    Monday 1019
3   Tuesday  872
4 Wednesday  845
5  Thursday  856
6    Friday  844
7  Saturday 1135
> ggplot(data = test,aes(x=Days,y=freq,group = 1))+geom_line(color="blue",size=2)+geom_point(color="red",size = 3)+ggtitle("crime frequency on week days")
>
```


Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
crime frequency on week days
1200
1100
1000
900
freq
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Days

# PRACTICAL 5

**PACKAGES REQUIRED** :

1. mongolite

2. ggplot2

3. lubridate

**AIM:**

1. Plot a line graph of crimes that took place in each month for type domestic = false and year 2015 based on their count using the Chicago Crime dataset.

2. Plot a line graph of crimes that took place at every hour of the day for type domestic = false and year 2015 based on their count using the Chicago Crime dataset.

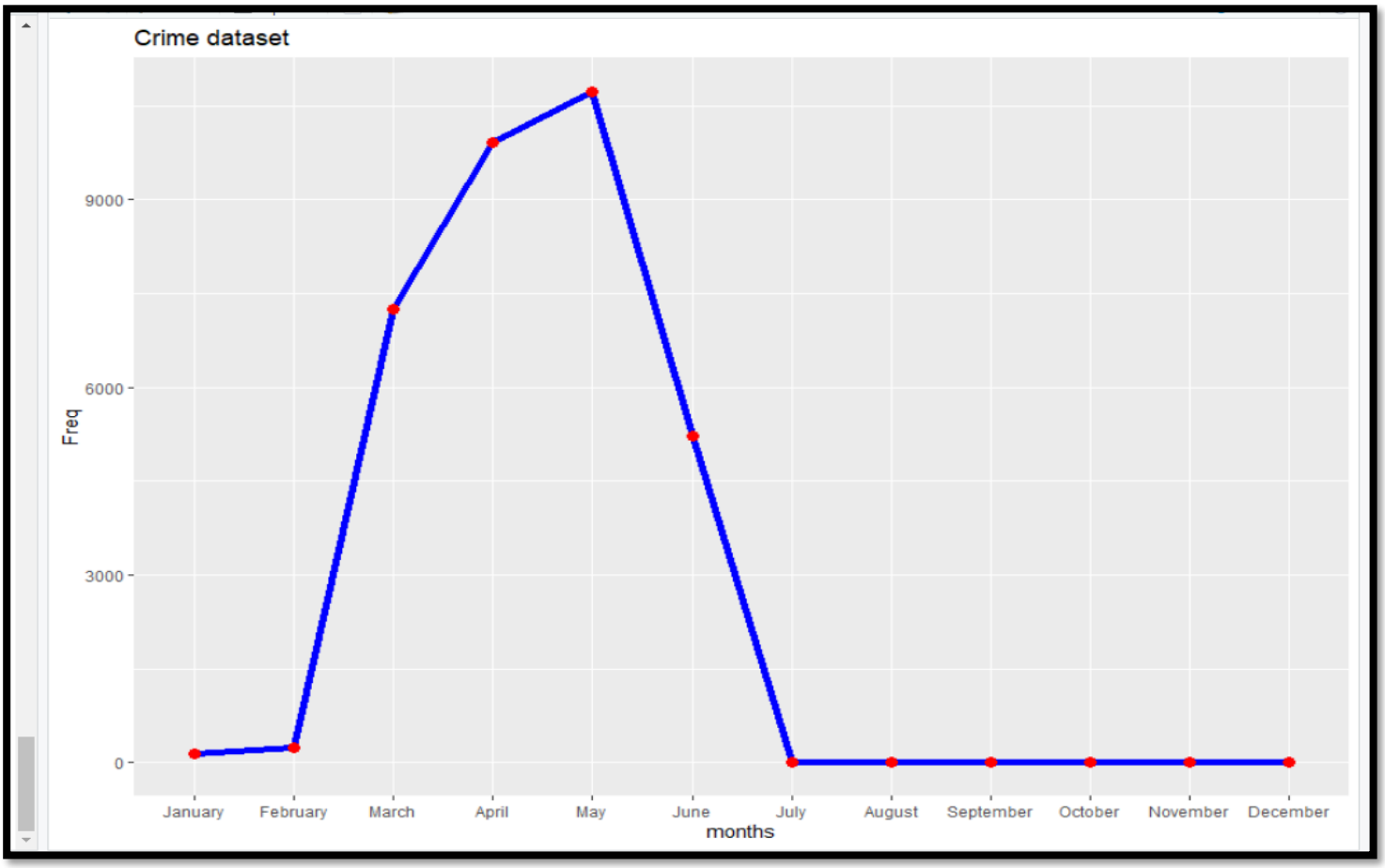
3. Plot a (multiple line) line graph grouped weekday-wise of crimes that took place at every hour of the day for type domestic = false and year 2015 based on their count using the Chicago Crime dataset. Each line in the plot represents each day of the week.

**Implementation on R Studio**

## Part 1.

```
> library(mongolite)
> library(lubridate)
> library(ggplot2)
> my_collection = mongo(collection = "crime", db = "Chicago")
> domestic=my_collection$find('{"Domestic":"FALSE","Year":"2015"}',fields='{"_id":0,"Domestic":1,"Da
te":1}')
> domestic$Date = mdy_hms(domestic$Date)
Warning message:
 20406 failed to parse.
> domestic$month = months(domestic$Date)
> domestic
                    Date Domestic    month
1   2015-03-18 19:44:00    FALSE    March
2   2015-03-18 23:00:00    FALSE    March
3   2015-03-18 22:30:00    FALSE    March
4   2015-03-18 21:00:00    FALSE    March
5   2015-03-18 22:00:00    FALSE    March
6   2015-03-18 22:09:00    FALSE    March
7   2015-03-18 21:25:00    FALSE    March
8   2015-03-18 21:30:00    FALSE    March
9   2015-03-18 21:14:00    FALSE    March
10  2015-03-18 22:50:00    FALSE    March
11  2015-03-18 22:31:00    FALSE    March
12  2015-03-18 12:55:00    FALSE    March
13  2015-03-18 20:00:00    FALSE    March
14  2015-03-18 21:00:00    FALSE    March
```

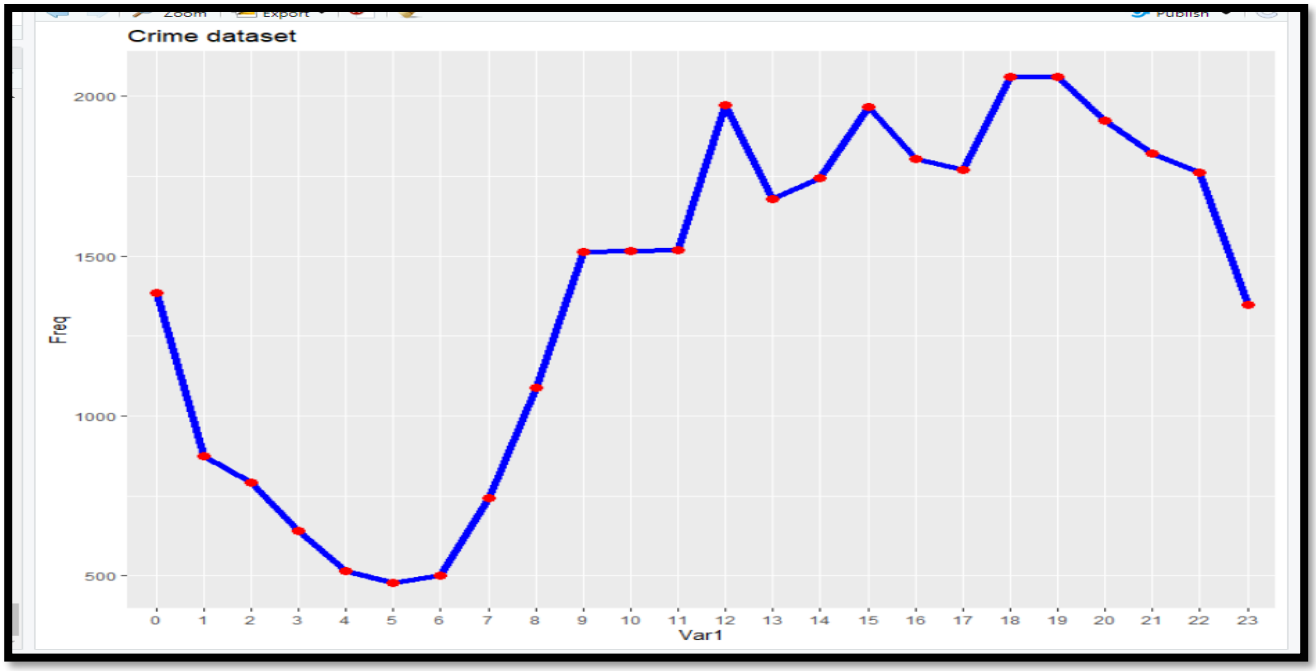
```
325 2015-03-19 14:00:00    FALSE    March
326 2015-03-18 21:00:00    FALSE    March
327 2015-03-19 11:30:00    FALSE    March
328                <NA>    FALSE     <NA>
329 2015-03-19 15:10:00    FALSE    March
330 2015-03-16 16:30:00    FALSE    March
331 2015-03-19 08:30:00    FALSE    March
332 2015-03-19 13:45:00    FALSE    March
333 2015-03-19 07:51:00    FALSE    March
 [ reached 'max' / getOption("max.print") -- omitted 53548 rows ]
> count=as.data.frame(table(domestic$month))
> count
      Var1  Freq
1    April  9916
2 February   228
3  January   130
4     July     4
5     June  5215
6    March  7250
7      May 10731
8 November     1
> test=data.frame(months=factor(c("January","February","March","April","May","June","July","August","September","October","November","Decem
r"),levels = c("January","February","March","April","May","June","July","August","September","October","November","December")), Freq=c(130,
250,9916,10731,5215,4,0,0,0,1,0))
> test
      months  Freq
1    January   130
2   February   228
3      March  7250
4      April  9916
5        May 10731
6       June  5215
7       July     4
8     August     0
9  September     0
10   October     0
11  November     1
12  December     0
> ggplot(data=test,aes(x=months,y=Freq,group=1))+geom_line(color="blue",size=2)+geom_point(color="red",size=3)+ggtitle("Crime dataset")
>
```



## Part 2.

```
> domestic$hour=hour(domestic$Date)
> domestic
                   Date Domestic    month hour
1   2015-03-18 19:44:00    FALSE    March   19
2   2015-03-18 23:00:00    FALSE    March   23
3   2015-03-18 22:30:00    FALSE    March   22
4   2015-03-18 21:00:00    FALSE    March   21
5   2015-03-18 22:00:00    FALSE    March   22
6   2015-03-18 22:09:00    FALSE    March   22
7   2015-03-18 21:25:00    FALSE    March   21
8   2015-03-18 21:30:00    FALSE    March   21
9   2015-03-18 21:14:00    FALSE    March   21
10  2015-03-18 22:50:00    FALSE    March   22
11  2015-03-18 22:31:00    FALSE    March   22
12  2015-03-18 12:55:00    FALSE    March   12
```

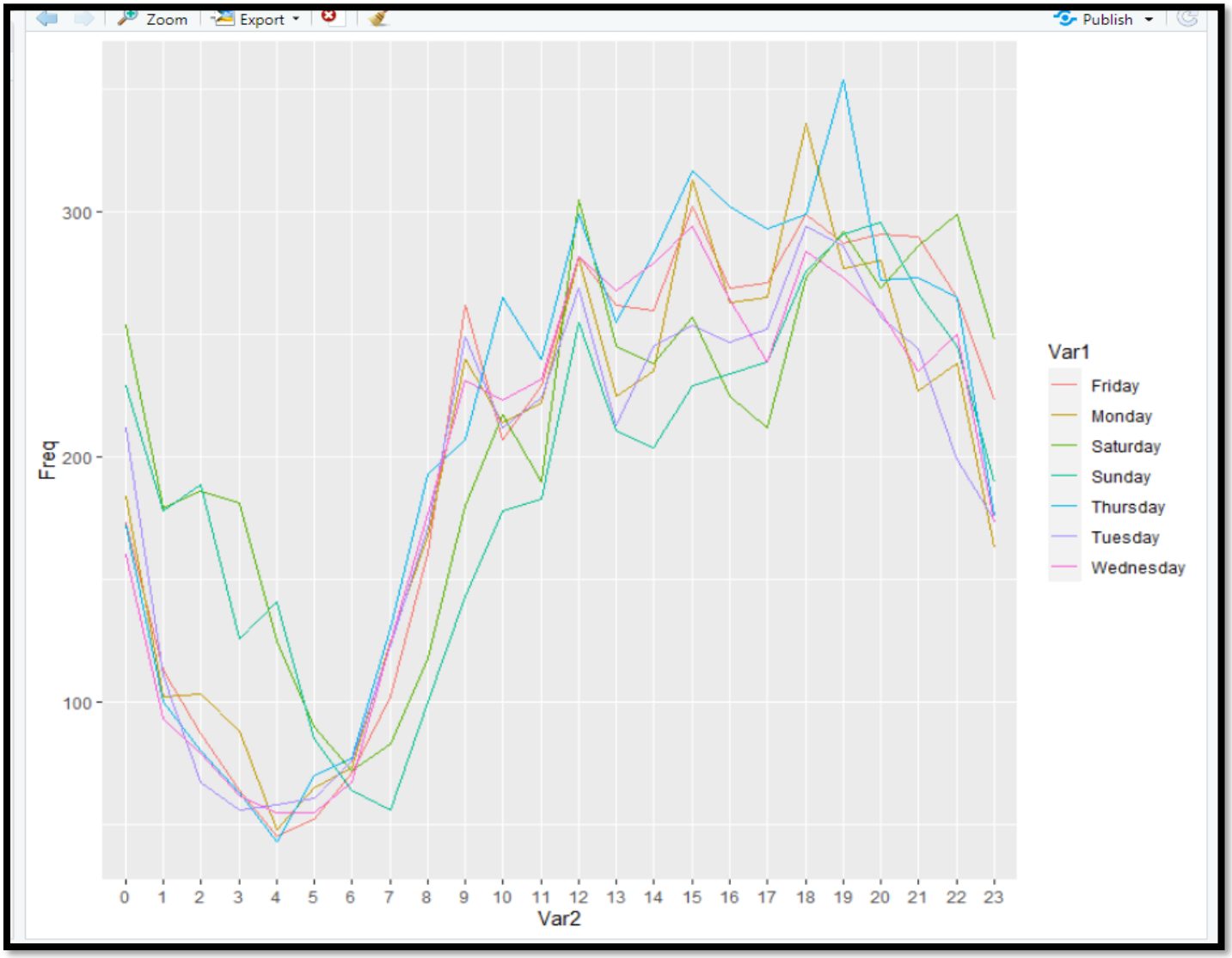
```
249                <NA>    FALSE     <NA>   NA
250                <NA>    FALSE     <NA>   NA
 [ reached 'max' / getOption("max.print") -- omitted 53631 rows ]
> hourCounts=as.data.frame(table(domestic$hour))
> hourCounts
   Var1 Freq
1     0 1384
2     1  876
3     2  791
4     3  640
5     4  515
6     5  478
7     6  500
8     7  743
9     8 1088
10    9 1512
11   10 1516
12   11 1520
13   12 1973
14   13 1679
15   14 1744
16   15 1966
17   16 1804
18   17 1771
19   18 2061
20   19 2060
21   20 1924
22   21 1822
23   22 1761
24   23 1347
> ggplot(data=hourCounts,aes(x=Var1,y=Freq,group=1))+geom_line(color="blue",size=2)+geom_point(color="red",size=3)+ggtitle("Crime dataset")
>
```




## Part 3.

```
> domestic$weekday=weekdays(domestic$Date)
> domestic$Hour=hour(domestic$Date)
> DayHourCounts=as.data.frame(table(domestic$weekday,domestic$Hour))
> DayHourCounts
         Var1 Var2 Freq
1      Friday    0  173
2      Monday    0  184
3    Saturday    0  254
4      Sunday    0  229
5    Thursday    0  172
6     Tuesday    0  212
7   Wednesday    0  160
8      Friday    1  113
9      Monday    1  102
10   Saturday    1  179
11     Sunday    1  178
12   Thursday    1  100
13    Tuesday    1  111
```

```
159  Thursday   22  265
160   Tuesday   22  199
161 Wednesday   22  250
162    Friday   23  223
163    Monday   23  163
164  Saturday   23  248
165    Sunday   23  190
166  Thursday   23  176
167   Tuesday   23  174
168 Wednesday   23  173
> ggplot(DayHourCounts,aes(x=Var2,y=Freq))+geom_line(aes(group=Var1,color=Var1))+ylab("Freq")
>
```


Zoom
Export
Publish
Freq
300
200
100
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Var2
Var1
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday

# PRACTICAL 6

**PACKAGES REQUIRED :**

1. mongolite
2. ggplot2
3. dplyr
4. gridExtra

**AIM:**

1. Plot separate line graphs of top 4 crimes (based on primary type) that took place on each day of the week based on their count using the Chicago Crime dataset.

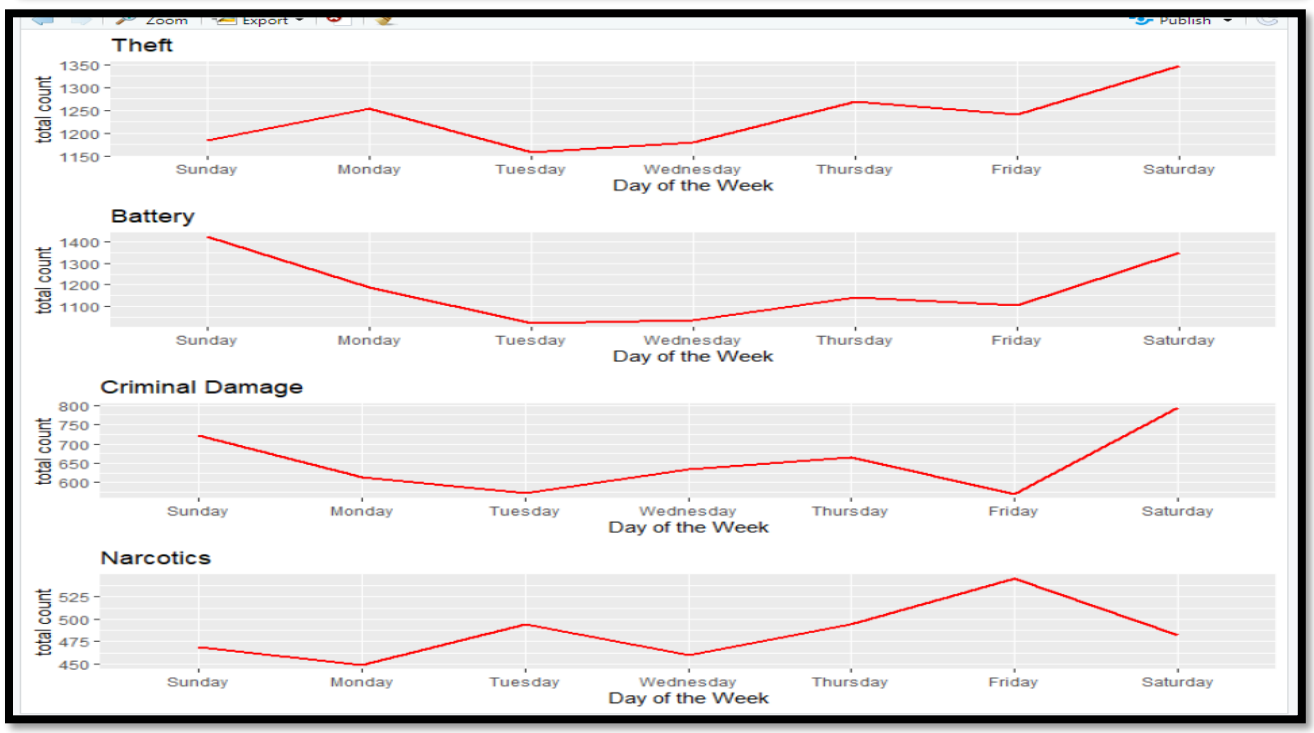
## Implementation on R studio

## Part 1.

```
> library(mongolite)
> library(lubridate)
> library(ggplot2)
> library(gridExtra)
> my_collection = mongo(collection = "crime", db="Chicago")
> data=my_collection$aggregate('[{"$group":{"_id":"$Primary Type", "total": {"$sum":1}}}]')%>% na.omit()%>%ar
range(desc(total))%>%head(4)
> data
              _id total
1           THEFT 13947
2         BATTERY 13083
3 CRIMINAL DAMAGE  7288
4       NARCOTICS  5563
> subData=my_collection$find('{}',fields = '{"_id":0,"Primary Type":1,"Date":1}')
> subData
                     Date                     Primary Type
1   03/18/2015 07:44:00 PM                          BATTERY
2   03/18/2015 11:00:00 PM                    OTHER OFFENSE
3   03/18/2015 10:45:00 PM                          BATTERY
4   03/18/2015 10:30:00 PM                          BATTERY
5   03/18/2015 09:00:00 PM                          ROBBERY
6   03/18/2015 10:00:00 PM                          BATTERY
7   03/18/2015 11:00:00 PM                          BATTERY
8   03/18/2015 09:35:00 PM                          BATTERY
9   03/18/2015 10:09:00 PM                        NARCOTICS
10  03/18/2015 09:25:00 PM                          BATTERY
11  03/18/2015 09:20:00 PM                   CRIMINAL DAMAGE
```

```
497 03/19/2015 04:30:00 PM                          BATTERY
498 03/19/2015 09:00:00 AM                         BURGLARY
499 03/19/2015 03:45:00 PM                          ROBBERY
500 03/19/2015 06:00:00 AM                            THEFT
 [ reached 'max' / getOption("max.print") -- omitted 65035 rows ]
> tf=data$"_id"
> data=filter(subData,subData$"Primary Type" %in% tf)
> count(data)
      n
1 39881
> data$Date=mdy_hms(data$Date)
Warning message:
 15022 failed to parse.
> data$weekday=weekdays(data$Date)
> data
                   Date    Primary Type   weekday
1   2015-03-18 19:44:00         BATTERY Wednesday
2   2015-03-18 22:45:00         BATTERY Wednesday
3   2015-03-18 22:30:00         BATTERY Wednesday
4   2015-03-18 22:00:00         BATTERY Wednesday
5   2015-03-18 23:00:00         BATTERY Wednesday
```

```
330 2015-03-19 19:25:00       NARCOTICS  Thursday
331 2015-03-19 18:00:00 CRIMINAL DAMAGE  Thursday
332 2015-03-19 12:30:00         BATTERY  Thursday
333 2015-03-19 15:30:00 CRIMINAL DAMAGE  Thursday
 [ reached 'max' / getOption("max.print") -- omitted 39548 rows ]
> g = function(data){WeekdayCounts = as.data.frame(table(data$weekday))
+     WeekdayCounts$Var1 = factor(WeekdayCounts$Var1, ordered=TRUE, levels=c("Sunday", "Monday", "Tuesday",
 "Wednesday", "Thursday", "Friday","Saturday"))
+     ggplot(WeekdayCounts, aes(x=Var1, y=Freq)) + geom_line(aes(group=1),size=1,color="red") + xlab("Day of
 the Week")}
> theft = filter(data,data$`Primary Type`=="THEFT")
> battery = filter(data,data$`Primary Type`=="BATTERY")
> cm = filter(data,data$`Primary Type`=="CRIMINAL DAMAGE")
> narc = filter(data,data$`Primary Type`=="NARCOTICS")
> g1 = g(theft)+ggtitle("Theft")+ylab("total count")
> g2= g(battery)+ggtitle("Battery")+ylab("total count")
> g3= g(cm)+ggtitle("Criminal Damage")+ylab("total count")
> g4= g(narc)+ggtitle("Narcotics")+ylab("total count")
> grid.arrange(g1,g2,g3,g4,ncol = 1)
>
```



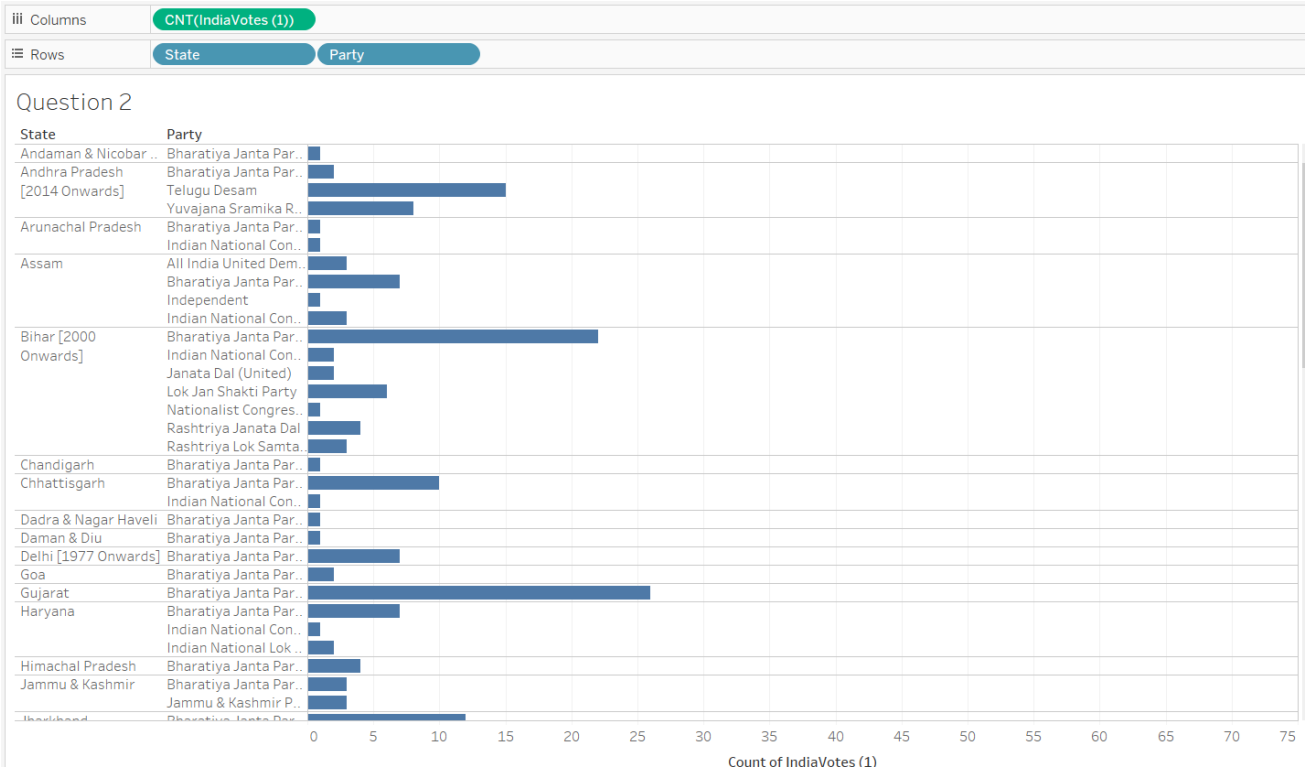
# PRACTICAL 7

**AIM:** .

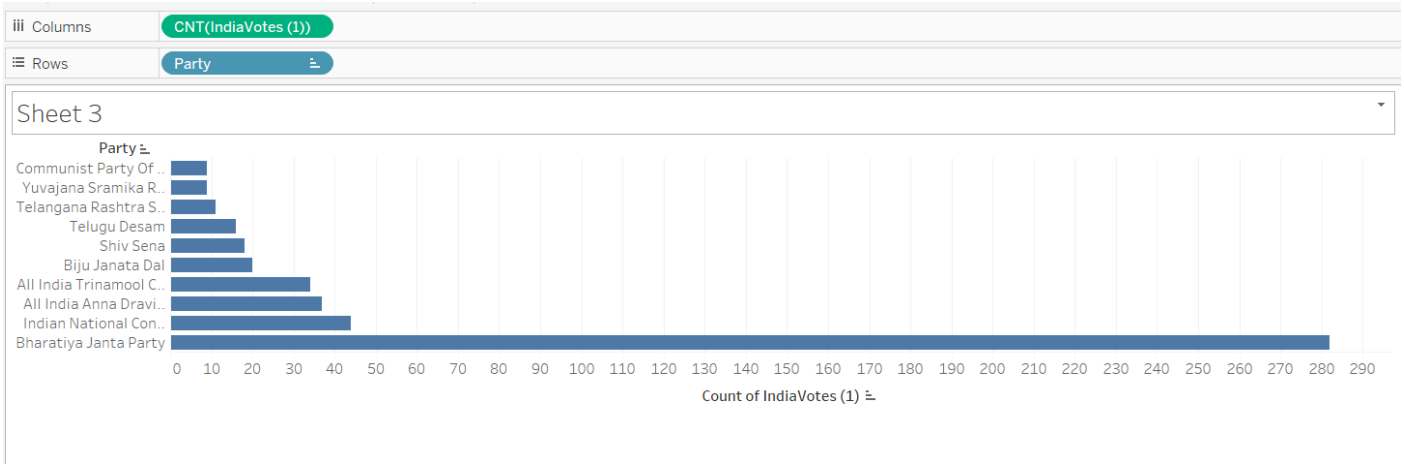
Solve the following questions on IndiaVotes dataset graphically using Tableau software.

Q1. Display party-wise number of seats.

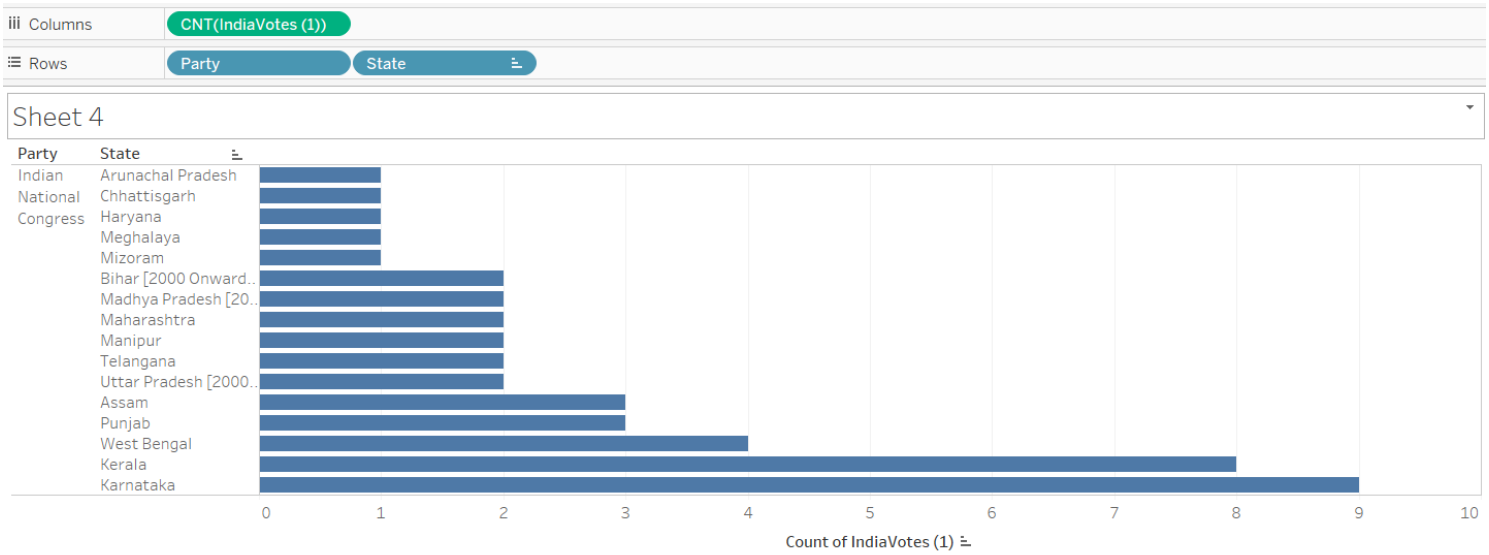
**Q2.** Display state-wise, party-wise number of seats



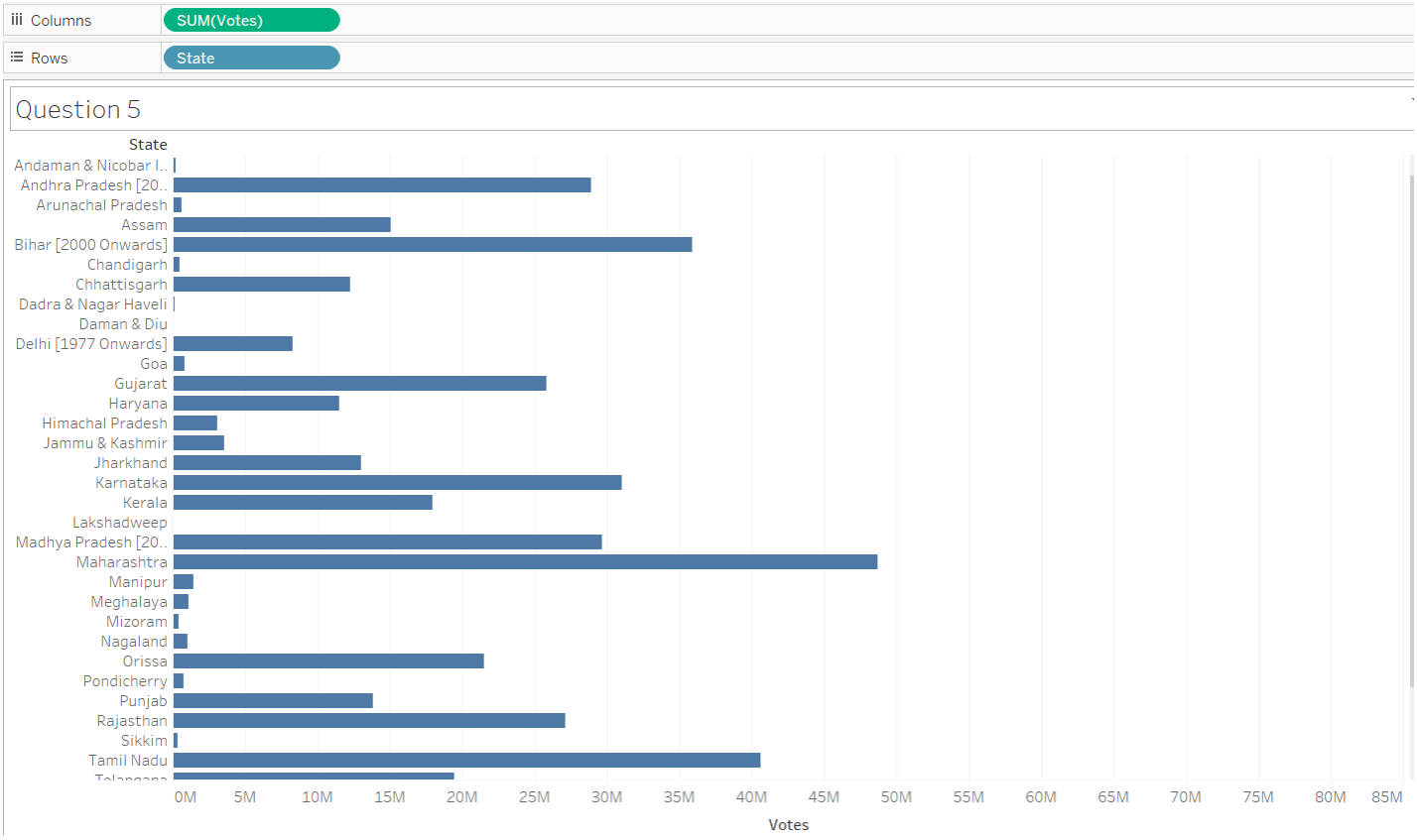

**Q3:** ) Display top 10 parties depending on the number of seats.



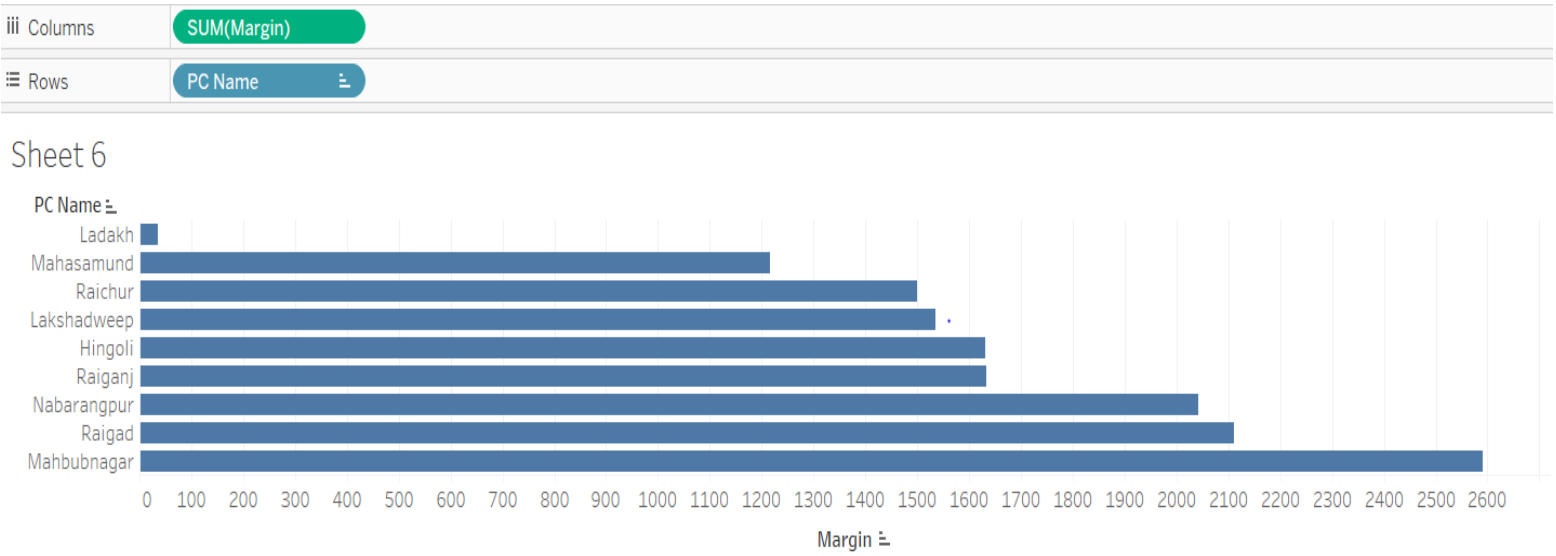
## **Q4:** Display performance of Indian National Congress in each state



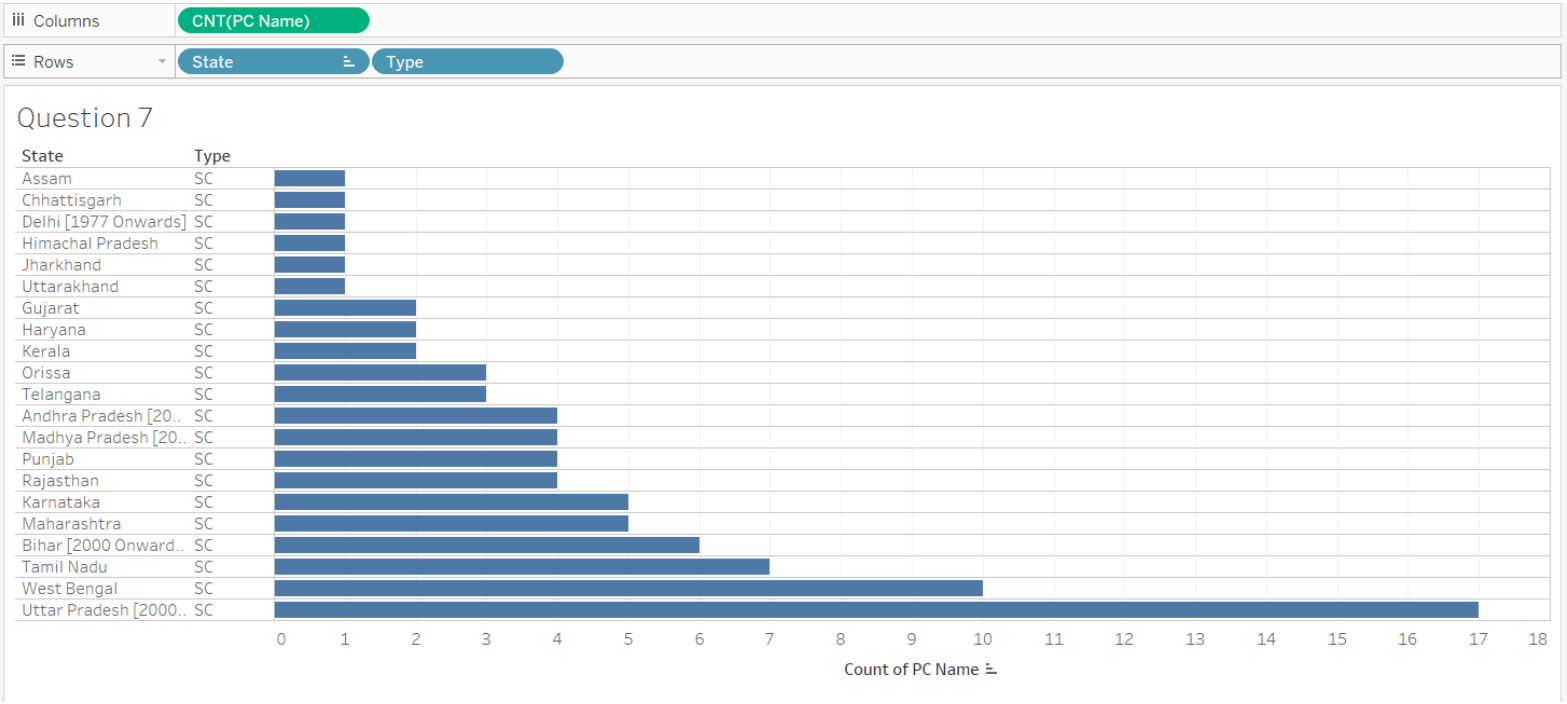

## **Q5:** Display state-wise number of votes casted.



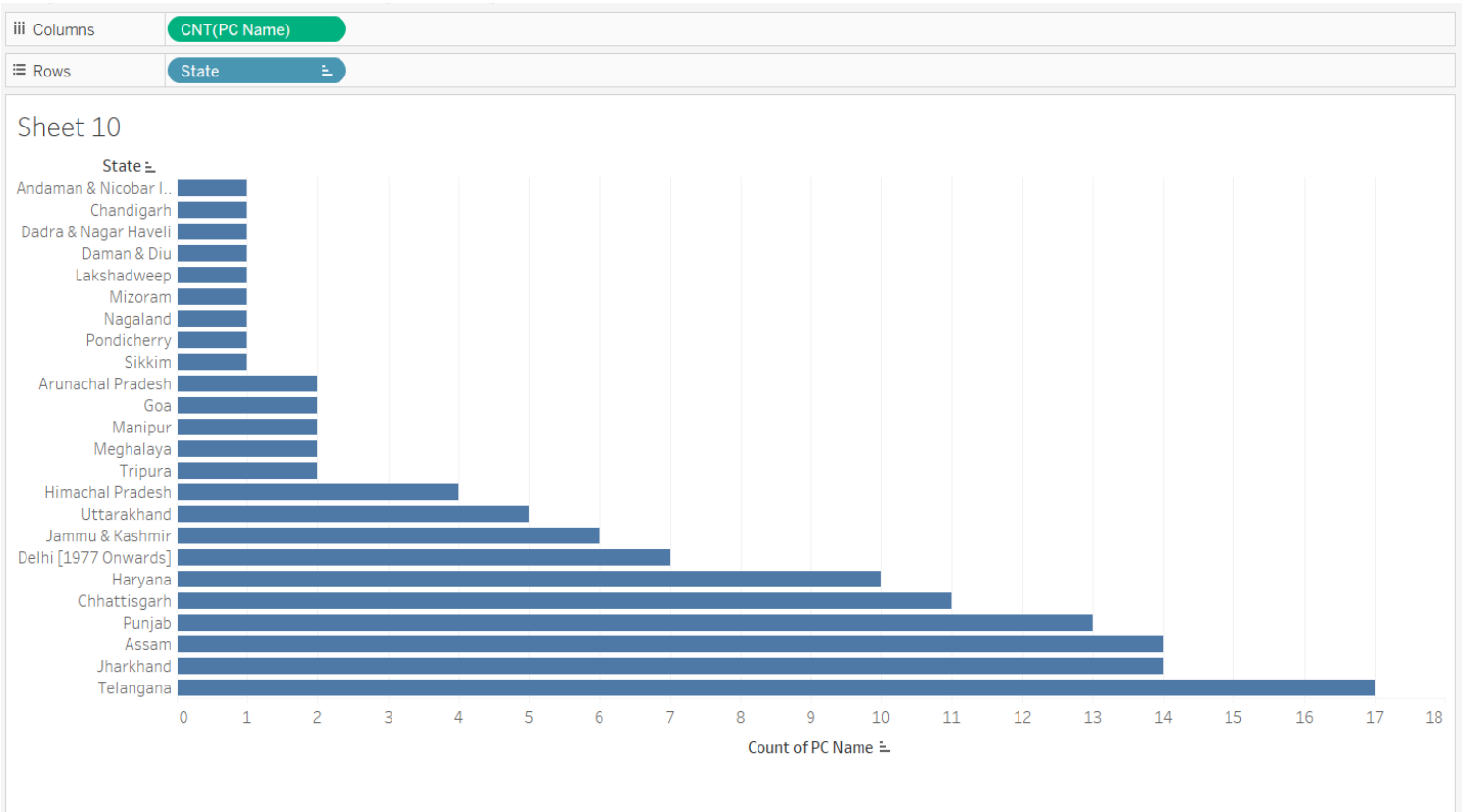
**Q6:-** Display PCs (Parliamentary Constituencies) where margin of votes is less than 3000.




Q7: Display the number of SC constituencies in each state



Q8: Display states with number of PCs less than 20

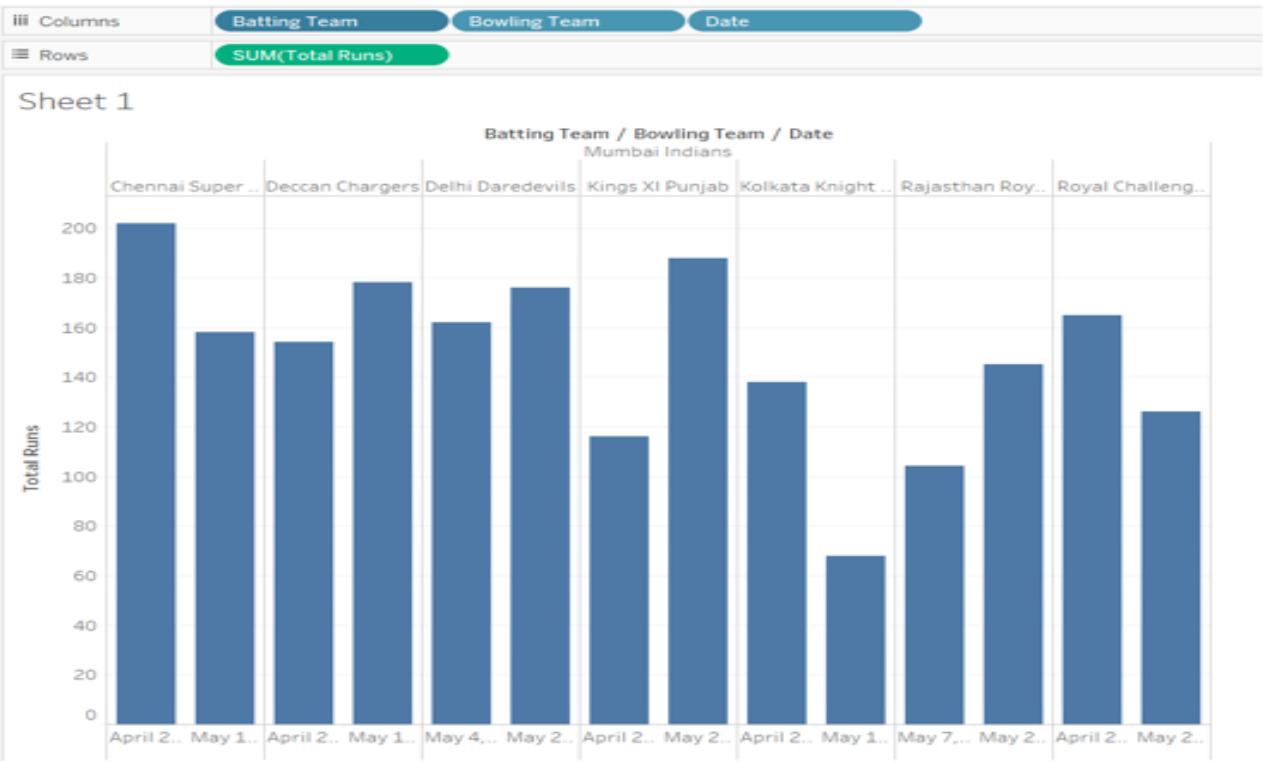


# PRACTICAL 8

## AIM:

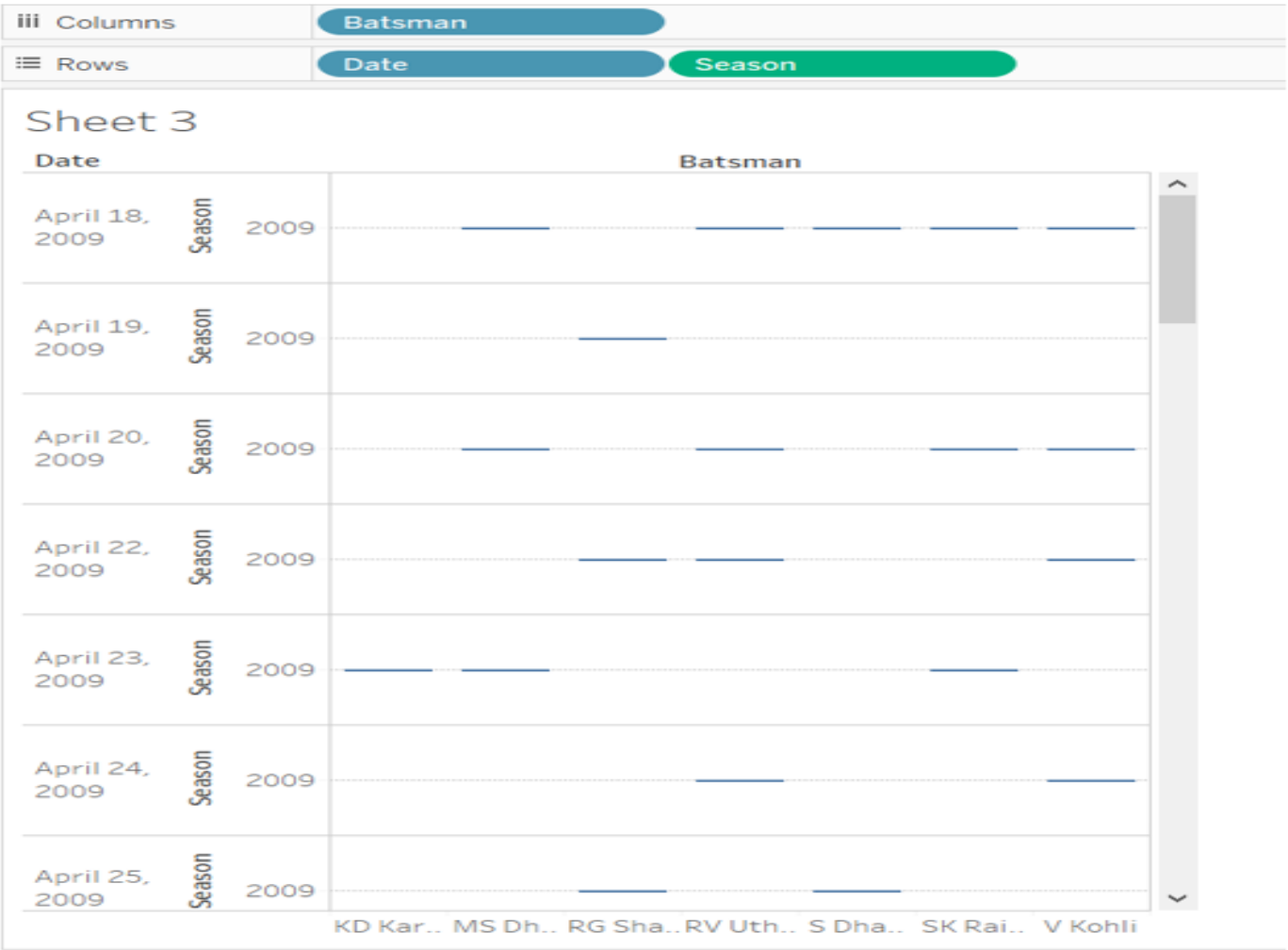
Solve the following questions on IPL dataset graphically using Tableau software.

Q1: Performance of Mumbai Indians vs other team in the year 2008(total runs)

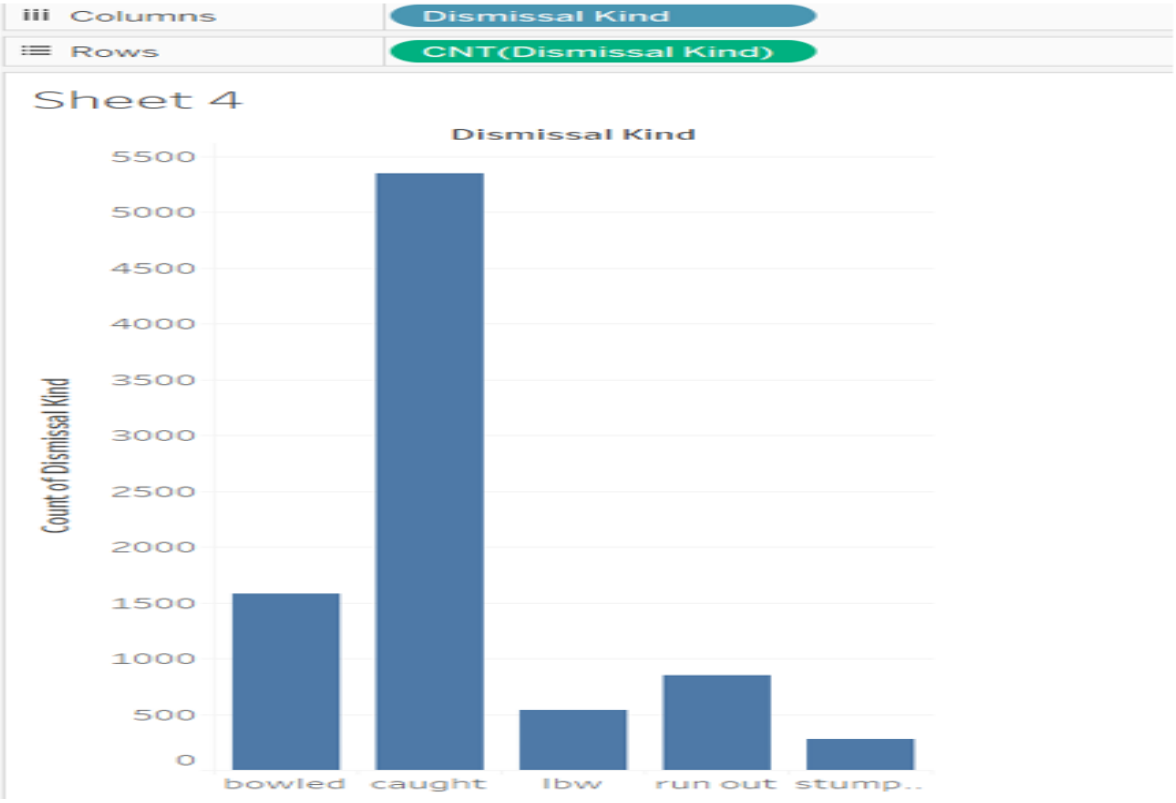


**Q2:** Top 5 wicket takers of Rajasthan royals

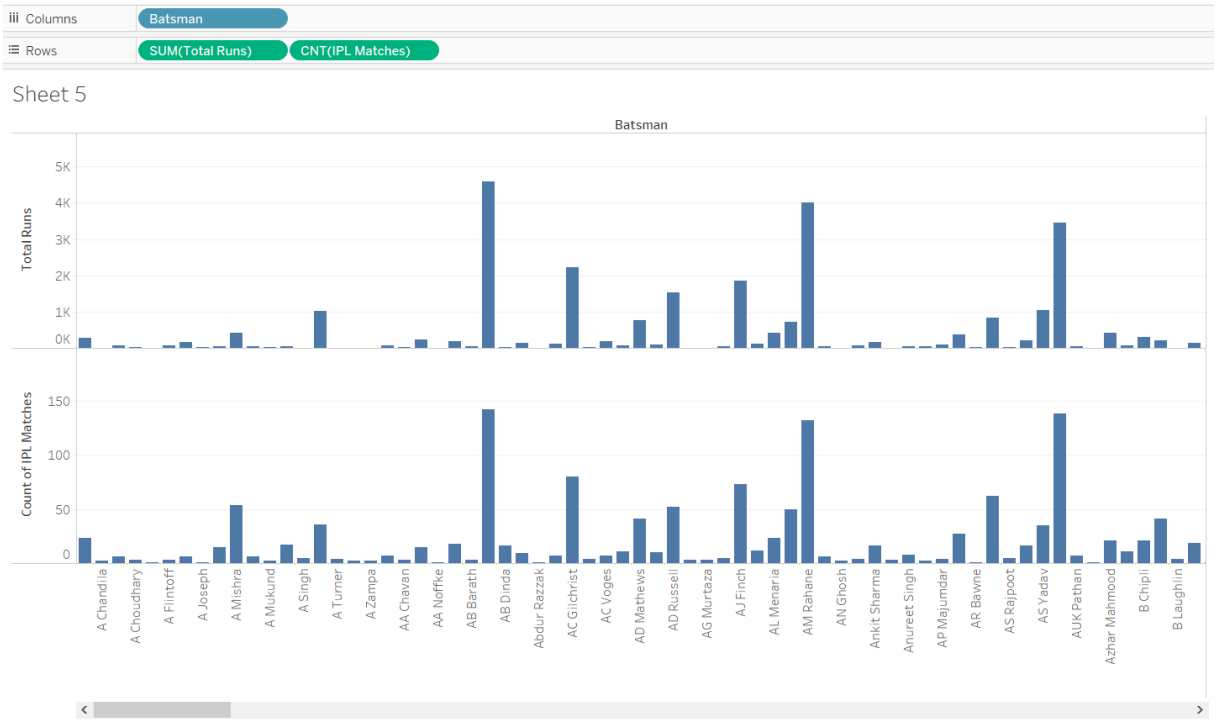
**Q3:** Top 7 batsman of the season of 2009



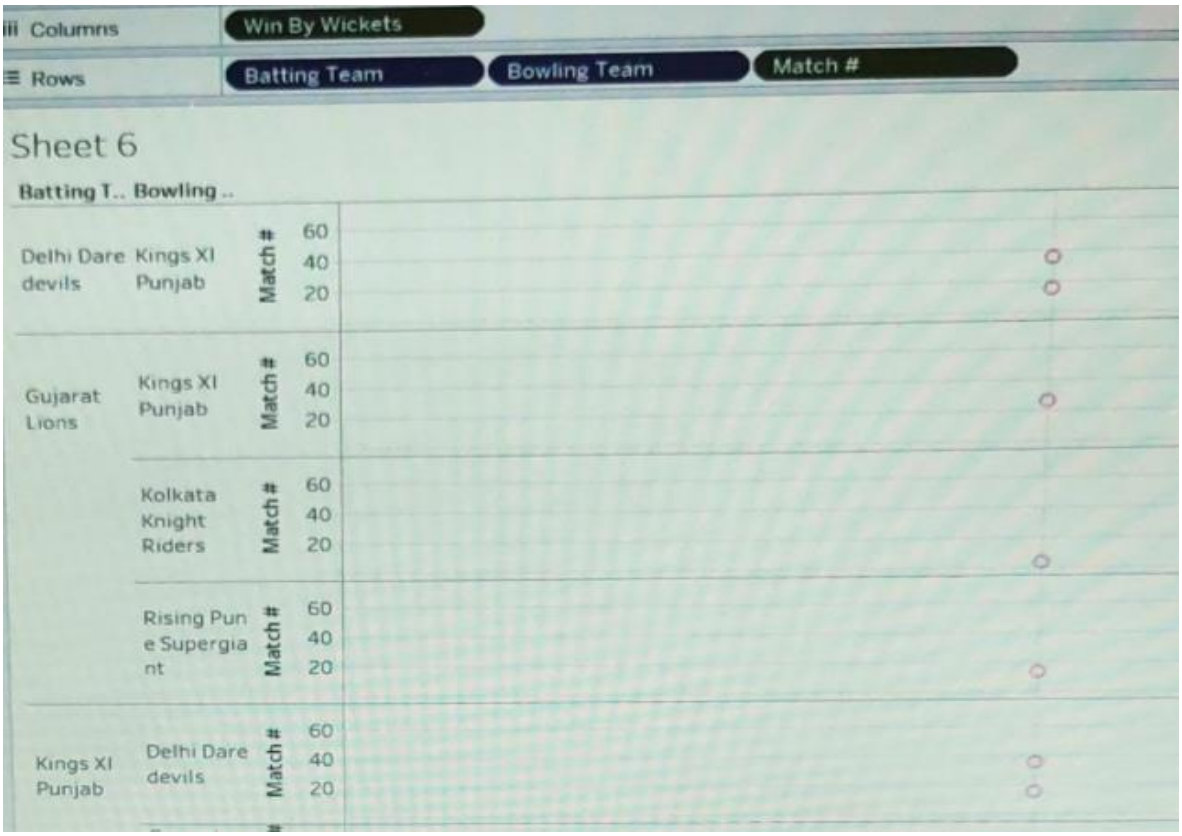
**Q4:** . Top 5 dissmissal kind



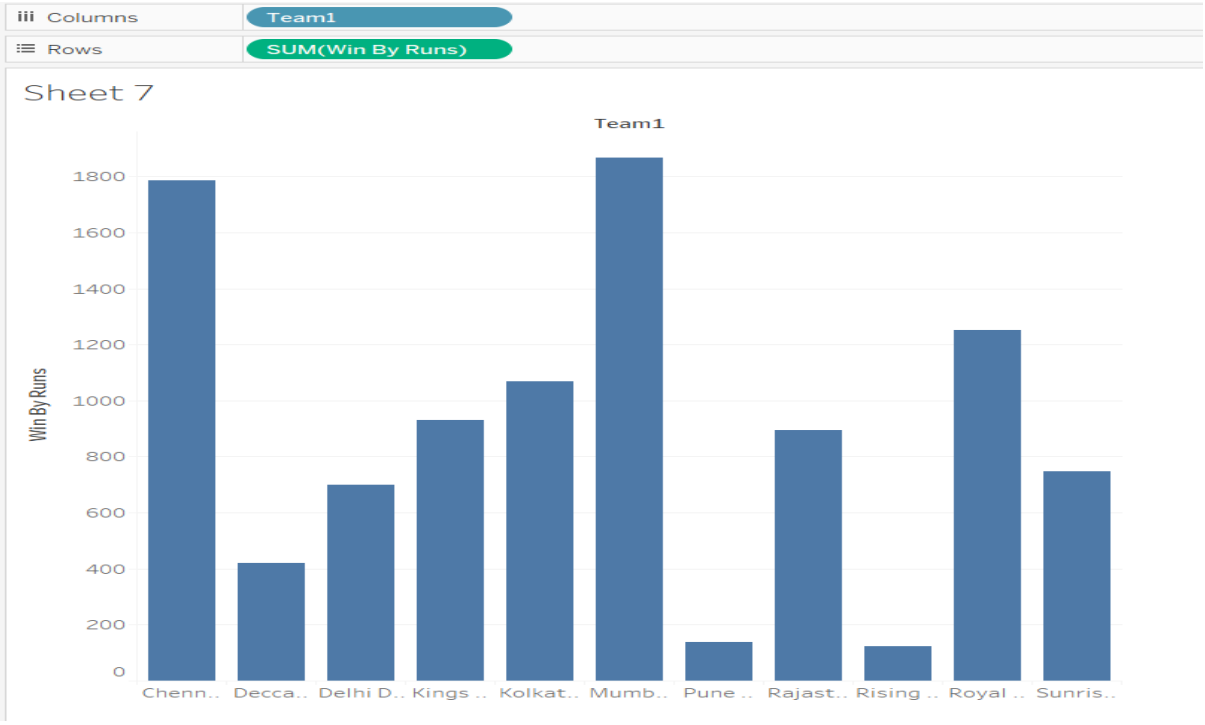

**Q5:** Playerwise total runs (history of IPL) entire data



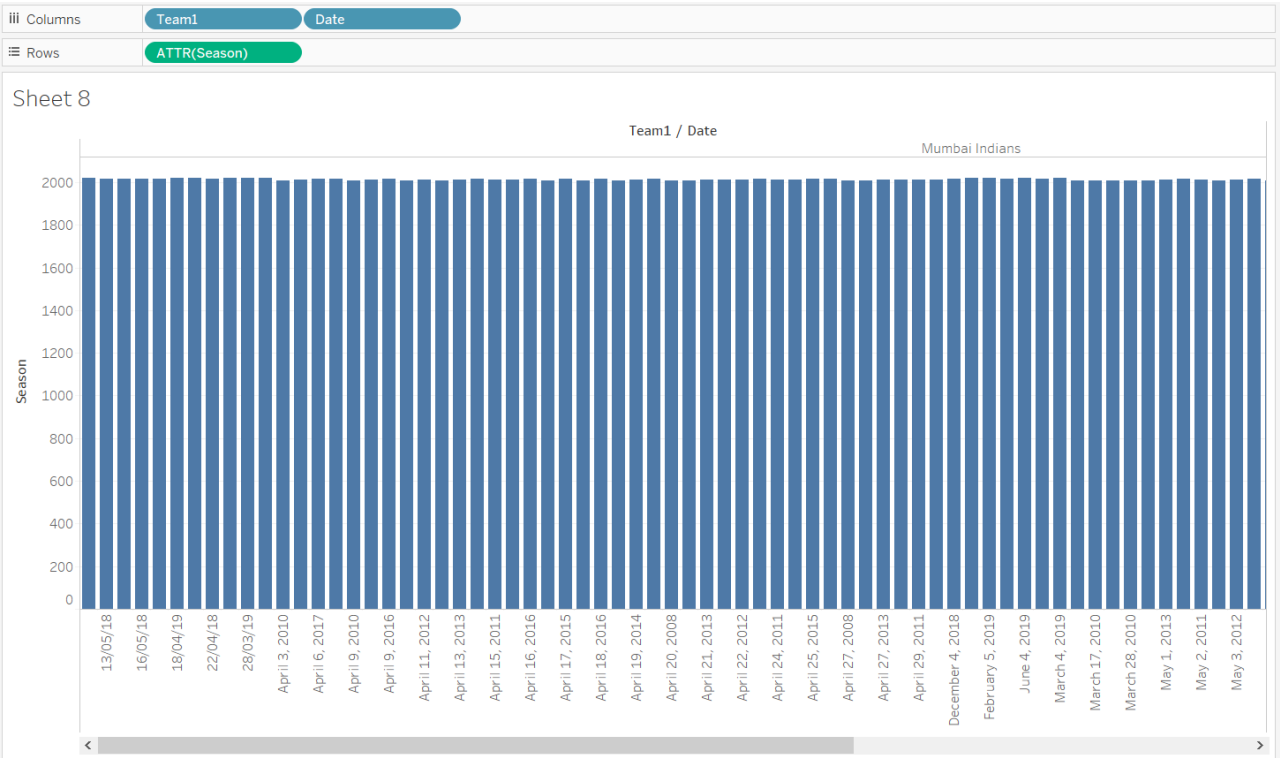
**Q6:** Details of the match won by 10 wickets



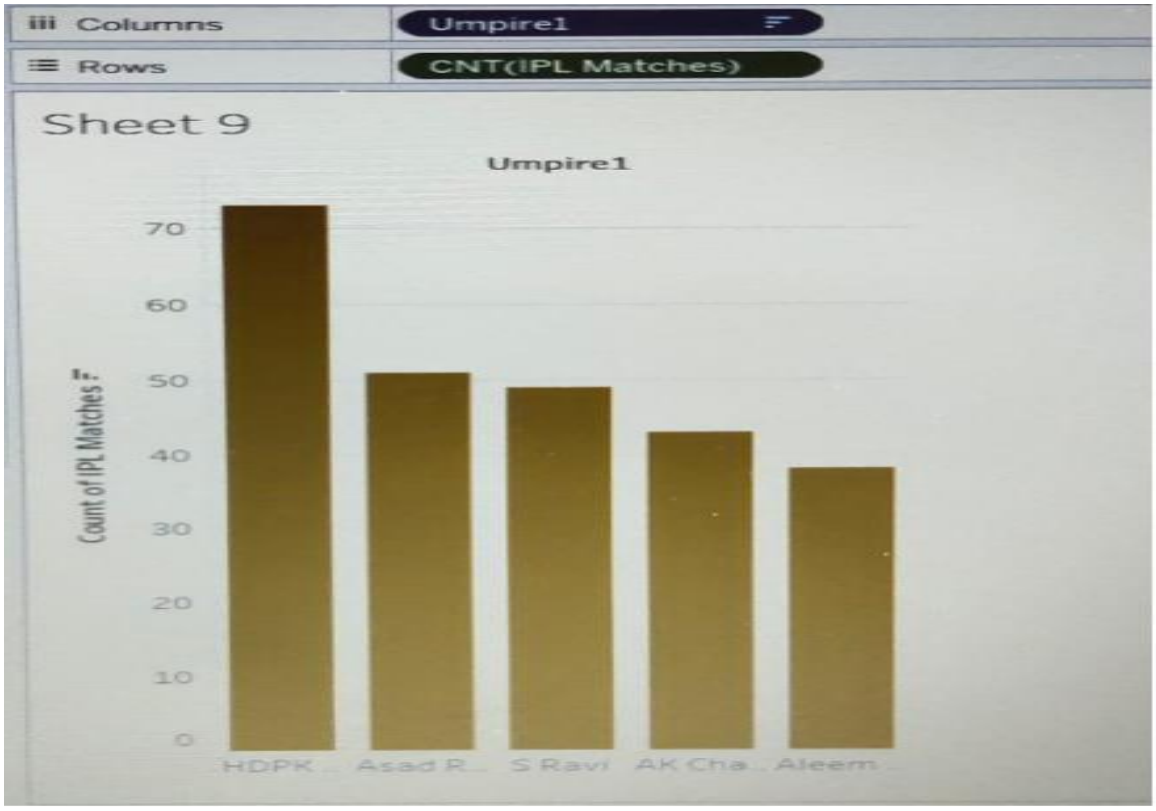

**Q7:** Won by more than 100 runs



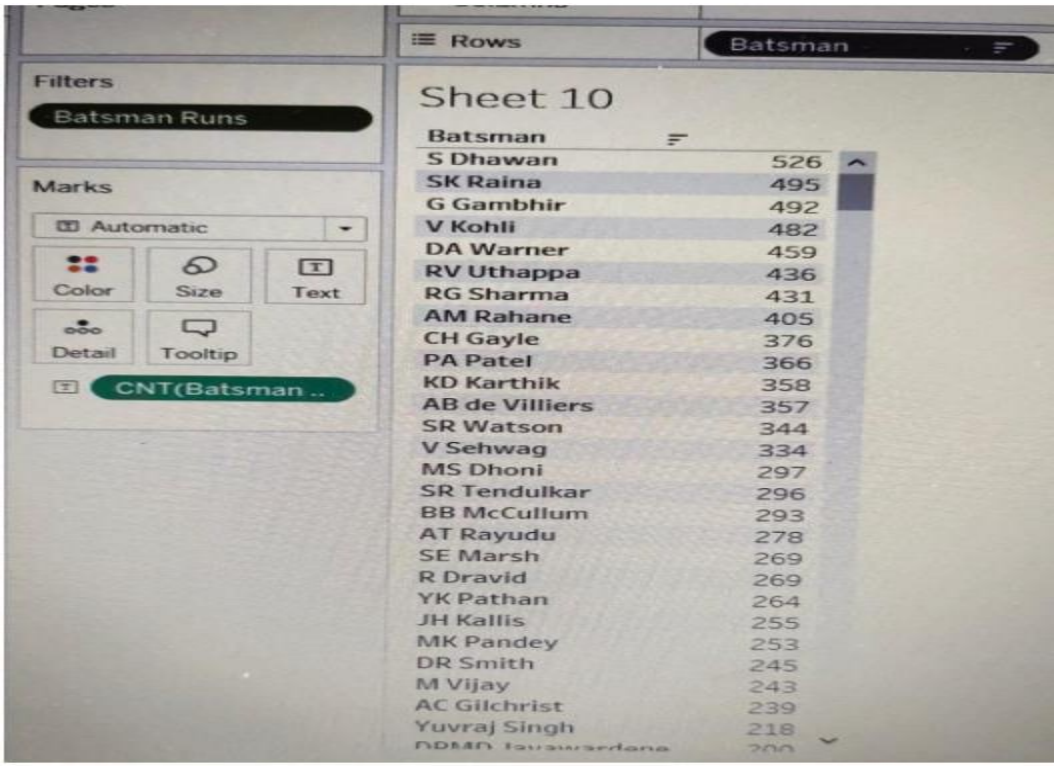
**Q8:** Performance of Mumbai Indians from 2008 to 2019




**Q9:** Umpire who supervised maximum number of match (top 5)

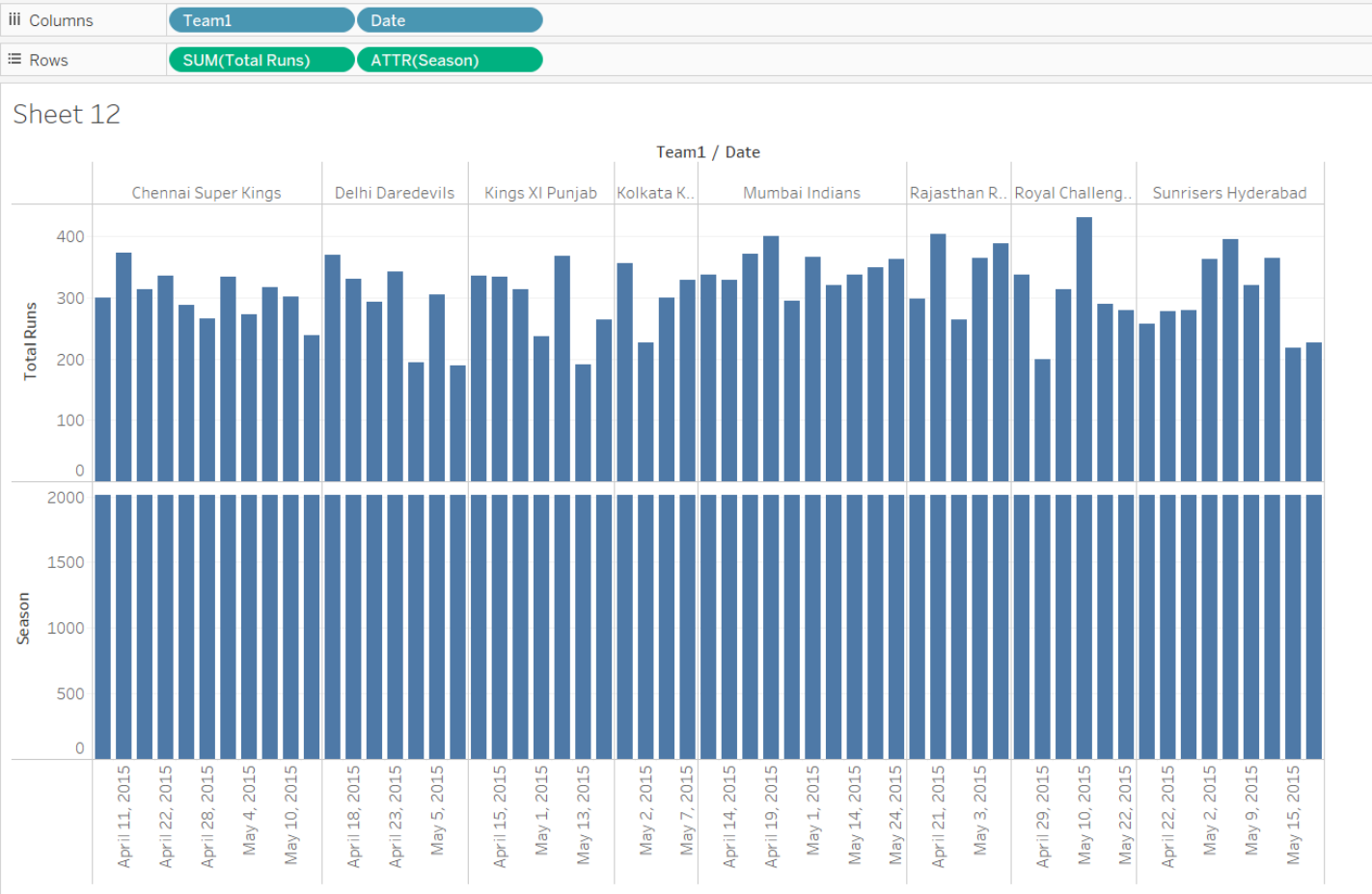


**Q 10:** Player wise no. of four(IPL history)




**Q11:** Player wise no. of six (IPL history)

**Q12:** Total runs scored by each team in the season 2015



# PRACTICAL 9

## AIM:

Make a summary dashboard of important data from IPL dataset graphically using Tableau software.