Assignment 2: IPL Data Analytics

By Riya Gupta (17BCS026)

Batting Data

Importing libraries

```
library(corrplot)
library(Hmisc)
library(tidyr)
library(stringr)
library(dplyr)
library(ggplot2)
library(plotly)
```

Set working directory

```
setwd("C:/Users/Riya/Desktop/DA-lab2/espenIPL")
```

Importing and cleaning data

```
# importing and cleaning data
player <- read.csv(file = 'batting.csv')[c(FALSE,TRUE),]
plyteam <- read.csv(file = 'batting.csv')[c(TRUE,FALSE),]
plyteam <- plyteam[-1,]
pteam <- plyteam$i..Batting.averages
player$team <- pteam
pdata <- as.data.frame(player)
colnames(pdata) <- c("player","matches","inns","no","runs","hs","avg","bf","sr","100s","50s","0","4s","6s","team")
pdata[pdata=="-"] <- NA
pdata$hs_clean <- stringr::str_replace(pdata$hs,'\\*','')
```

The original data – batting.csv looked as the following:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Batting averages													
2	Player	Mat	Inns	NO	Runs	HS	Ave	BF	SR	100	50	0	4s	6s
3	VR Aaron	5	1	1	3	3*	-	7	42.85	0	0	0	0	0
4	(Rajasthan Royals)													
5	Abhishek S	3	3	1	9	5*	4.5	9	100	0	0	0	1	0
6	(Sunrisers Hyderabad)													
7	AD Nath	8	5	0	61	24	12.2	57	107.01	0	0	0	5	2
8	(Royal Challengers Bangalore)													
9	MA Agarw	13	13	0	332	58	25.53	234	141.88	0	2	1	26	14
10	(Kings XI Punjab)													
11	KK Ahmed	9	1	0	0	0	0	1	0	0	0	1	0	0
12	(Sunrisers Hyderabad)													
13	MM Ali	11	10	2	220	66	27.5	133	165.41	0	2	0	16	17
14	(Royal Challengers Bangalore)													
15	JC Archer	11	5	3	67	27*	33.5	40	167.5	0	0	0	4	4
16	(Rajasthan Royals)													
17	Arshdeep S	3	-	-	-	-	-	-	-	-	-	-	-	-
18	(Kings XI Punjab)													
19	M Ashwin	10	3	2	3	1*	3	5	60	0	0	0	0	0
20	(Kings XI Punjab)													
21	R Ashwin	14	6	1	42	17*	8.4	28	150	0	0	2	3	3
22	(Kings XI Punjab)													
23	Avesh Kha	1	1	1	4	4*	-	3	133.33	0	0	0	1	0
24	(Delhi Capitals)													
25	JM Bairsto	10	10	2	445	114	55.62	283	157.24	1	2	1	48	18
26	(Sunrisers Hyderabad)													
27	Basil Tham	3	1	1	1	1*	-	1	100	0	0	0	0	0
28	(Sunrisers Hyderabad)													
29	JP Behrenc	5	-	-	-	-	-	-	-	-	-	-	-	-

Importing as following in player and plyteam objects in R and then combining the two as pdata, dataframe

	player	matches	inns	no	runs	hs	avg	bf	sr	100s	50s	0	4s	6s	team	hs_clean
2	VR Aaron	5	1	1	3	3*	NA	7	42.85	0	0	0	0	0	(Rajasthan Royals)	3
4	Abhishek Sharma	3	3	1	9	5*	4.5	9	100	0	0	0	1	0	(Sunrisers Hyderabad)	5
6	AD Nath	8	5	0	61	24	12.2	57	107.01	0	0	0	5	2	(Royal Challengers Bangalore)	24
8	MA Agarwal	13	13	0	332	58	25.53	234	141.88	0	2	1	26	14	(Kings XI Punjab)	58
10	KK Ahmed	9	1	0	0	0	0	1	0	0	0	1	0	0	(Sunrisers Hyderabad)	0
12	MM Ali	11	10	2	220	66	27.5	133	165.41	0	2	0	16	17	(Royal Challengers Bangalore)	66
14	JC Archer	11	5	3	67	27*	33.5	40	167.5	0	0	0	4	4	(Rajasthan Royals)	27
16	Arshdeep Singh	3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	(Kings XI Punjab)	NA
18	M Ashwin	10	3	2	3	1*	3	5	60	0	0	0	0	0	(Kings XI Punjab)	1
20	R Ashwin	14	6	1	42	17*	8.4	28	150	0	0	2	3	3	(Kings XI Punjab)	17
22	Avesh Khan	1	1	1	4	4*	NA	3	133.33	0	0	0	1	0	(Delhi Capitals)	4
24	JM Bairstow	10	10	2	445	114	55.62	283	157.24	1	2	1	48	18	(Sunrisers Hyderabad)	114
26	Basil Thampi	3	1	1	1	1*	NA	1	100	0	0	0	0	0	(Sunrisers Hyderabad)	1
28	JP Behrendorff	5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	(Mumbai Indians)	NA
30	RK Bhui	1	1	0	7	7	7	12	58.33	0	0	0	0	0	(Sunrisers Hyderabad)	7
32	SW Billings	1	1	0	0	0	0	4	0	0	0	1	0	0	(Chennai Super Kings)	0

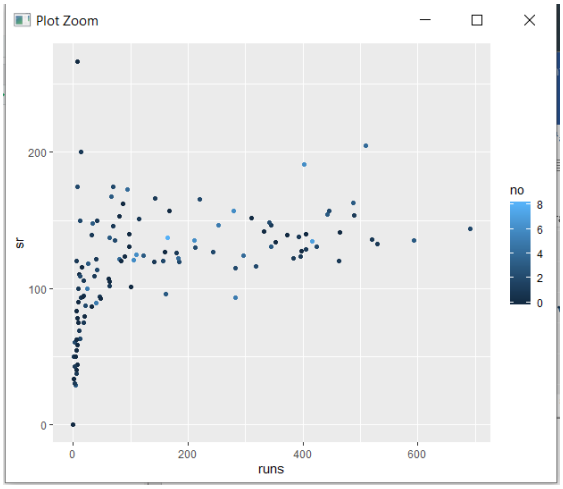
Checking the structure of pdata, by *str(pdata)* we get,

```
> str(pdata)
'data.frame':   162 obs. of  16 variables:
 $ player  : Factor w/ 171 levels "(Chennai Super Kings)",..: 165 11 12 90 76 97 62 18 86 119
 ...
 $ matches : Factor w/ 19 levels "","1","10","11",..: 14 12 17 6 18 4 4 12 3 7 ...
 $ inns    : Factor w/ 20 levels "","-","1","10",..: 3 13 15 7 3 4 15 NA 13 16 ...
 $ no      : Factor w/ 12 levels "","-","0","1",..: 4 4 3 3 3 5 6 NA 5 4 ...
 $ runs    : Factor w/ 94 levels "","-","0","1",..: 41 90 77 45 3 33 80 NA 41 61 ...
 $ hs      : Factor w/ 96 levels "","-","0","0*",..: 38 58 32 63 3 68 35 NA 6 24 ...
 $ avg     : Factor w/ 98 levels "","-","0","0.5",..: NA 75 9 41 3 46 63 NA 49 95 ...
 $ bf      : Factor w/ 97 levels "","-","0","1",..: 87 95 82 36 4 15 70 NA 77 47 ...
 $ sr      : Factor w/ 116 levels "","-","0","100",..: 92 4 9 56 3 74 77 NA 97 63 ...
 $ 100s    : Factor w/ 5 levels "","-","0","1",..: 3 3 3 3 3 3 3 NA 3 3 ...
 $ 50s     : Factor w/ 11 levels "","-","0","1",..: 3 3 3 5 3 5 3 NA 3 3 ...
 $ 0       : Factor w/ 6 levels "","-","0","1",..: 3 3 3 4 4 3 3 NA 3 5 ...
 $ 4s      : Factor w/ 43 levels "","-","0","1",..: 3 4 36 18 3 10 28 NA 3 21 ...
 $ 6s      : Factor w/ 30 levels "","-","0","1",..: 3 3 13 9 3 11 24 NA 3 22 ...
 $ team    : Factor w/ 171 levels "(Chennai Super Kings)",..: 6 8 7 3 8 7 6 3 3 3 ...
 $ hs_clean: chr  "3" "5" "24" "58" ...
```

Since the columns as in Factors any numeric manipulation is difficult on the data, we therefore convert these columns to numeric,

```
r2 <- as.data.frame(pdata)

r2$matches <- stringr::str_replace(r2$matches,'\\ ','')
r2$matches <- as.numeric(r2$matches)
r2$inns <- stringr::str_replace(r2$inns,'\\ ','')
r2$inns <- as.numeric(r2$inns)
r2$no <- stringr::str_replace(r2$no,'\\ ','')
r2$no <- as.numeric(r2$no)
r2$runs <- stringr::str_replace(r2$runs,'\\ ','')
r2$runs <- as.numeric(r2$runs)
r2$hs_clean <- stringr::str_replace(r2$hs_clean,'\\ ','')
r2$hs_clean <- as.numeric(r2$hs_clean)
r2$avg <- stringr::str_replace(r2$avg,'\\ ','')
```

On plotting runs vs strike rate graph for all the players, we get a graph that is not very intuitive since the list of players had data of bowlers also who have much scores in batting fields and the data, we have is not good for analyzing batting data.
Following is runs vs strike rate graph for all the players, color = not out.



Descriptive Analysis

Number of players

```
> # Number of players
> pdata %>% summarise(pdata_count = n())
  pdata_count
1         162
```

Number of teams

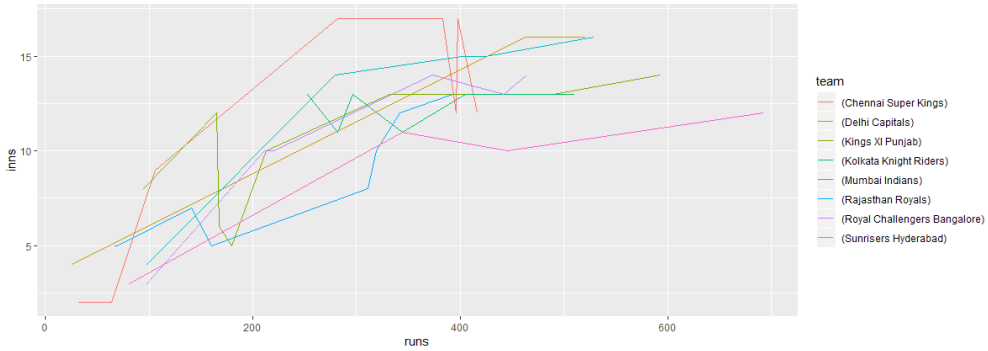
```
> # Number of teams
> team_count = length(unique(pdata$team))
> team_count
[1] 8
```

Which player wins with maximum runs?

```
> # Which player wins with maximum runs
> max_runs = r2[which.max(r2$runs),]
> max_runs %>% select('player','runs')
        player runs
310 DA Warner  692
```

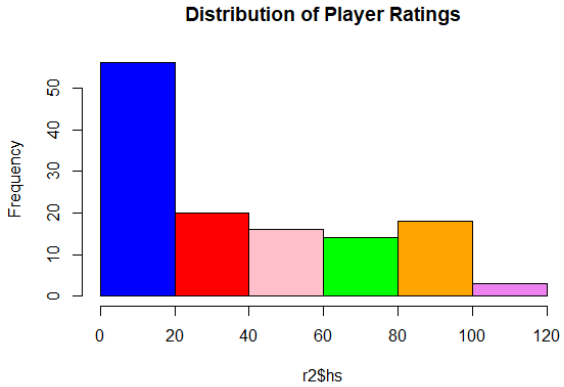
Inns vs Runs line graph of top 50 players according to teams

```
maxavg <- head(r2[
  order( r2[,7], r2[,2] , decreasing = TRUE),
  ],n=50)
ggplot(maxavg, aes(runs,inns), y=player) +
  geom_line(aes(color=team, group=team))
```



We don't see much trend except that Chennai Super Kings players are outperforming others.

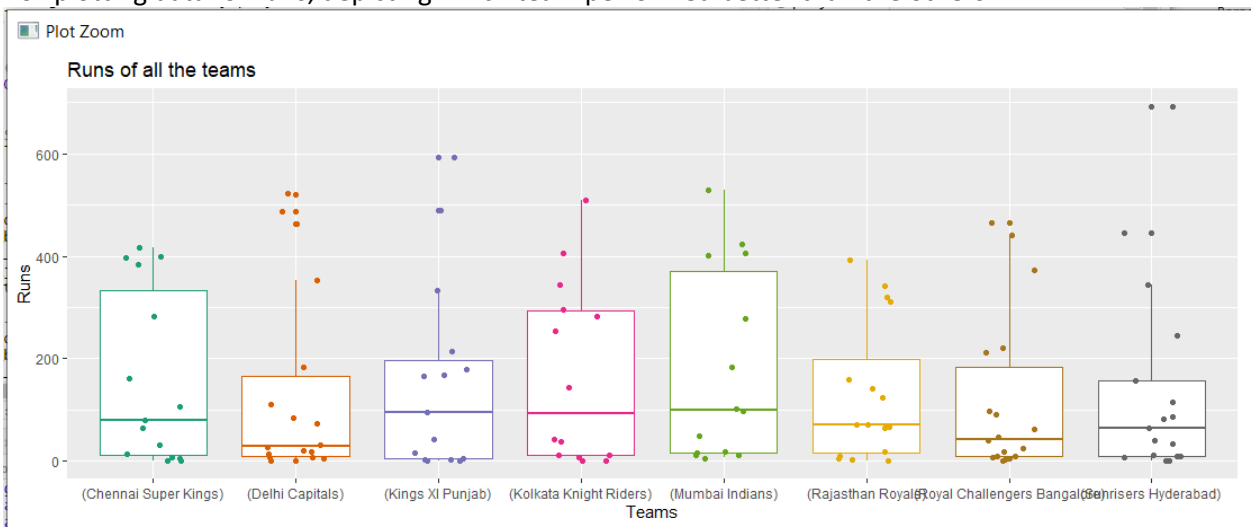
Plotting frequency vs high score graph



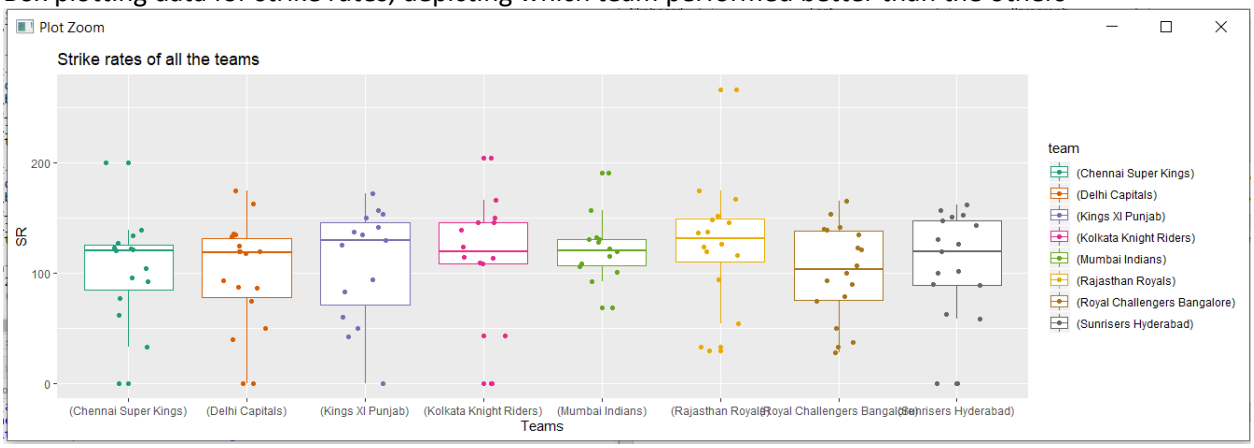
From the graph we get to know that maximum players have their high scores between 0 to 20, there is an equal number of players in 20-40, 40-60, 60-80 and 80-100 class intervals whereas there are only few players who have high scores above 100.

Box plots

Box plotting data for runs, depicting which team performed better than the others



Box plotting data for strike rates, depicting which team performed better than the others



Kings XI Punjab and Rajasthan Royals have better strike rates than the others. Moreover, the players of Rajasthan Royals perform better than Kings XI Punjab in general.

Fetching Top 15 Players

```
# Giving weights to the attributes
w1<-8.5
w2<-5
w3<-2
w4<-7.5
w5<-5
w6<-7
w7<-6
w8<-4
w9<-12.5

# Calulating scores
score <- as.numeric(pdata$hs_clean) * w1 + as.numeric(pdata$runs)/as.numeric(pdata$bf) * w2 +
  as.numeric(pdata$sr) * w7 + as.numeric(pdata$`100s`) * w3 + as.numeric(pdata$'50s') * w4 +
  as.numeric(pdata$`4s`) * w5 + as.numeric(pdata$`6s`) * w6 + as.numeric(pdata$no) * w8 +
  as.numeric(pdata$avg) * w9

#batScore = batStrikeRate * batAverage
bs <- as.numeric(pdata$runs) * as.numeric(pdata$sr) * 0.01

pdata$batscore <- as.numeric(bs) + as.numeric(score)
```

bowling.R × code.R* × r2 ×

Filter

	player	matches	inns	no	runs	hs	avg	bf	sr	100s	50s	0	4s	6s	team	batscore
24	JM Bairstow	10	10	2	445	114	55.62	283	157.24	1	2	1	48	18	(Sunrisers Hyderabad)	2861.607
310	DA Warner	12	12	2	692	100	69.20	481	143.86	1	8	0	57	21	(Sunrisers Hyderabad)	2799.070
232	AD Russell	14	13	4	510	80	56.66	249	204.81	0	4	1	31	52	(Kolkata Knight Riders)	2742.540
214	KL Rahul	14	14	3	593	100	53.90	438	135.38	1	6	0	49	25	(Kings XI Punjab)	2656.282
80	CH Gayle	13	13	1	490	99	40.83	319	153.60	0	4	0	45	34	(Kings XI Punjab)	2649.332
184	HH Pandya	16	15	6	402	91	44.66	210	191.42	0	1	1	28	29	(Mumbai Indians)	2640.463
72	MS Dhoni	15	12	7	416	84	83.20	309	134.62	0	3	0	22	23	(Chennai Super Kings)	2498.255
68	AB de Villiers	13	13	3	442	82	44.20	287	154.00	0	5	0	31	26	(Royal Challengers Bangalore)	2492.140
212	AM Rahane	14	13	1	393	105	32.75	285	137.89	1	1	1	45	9	(Rajasthan Royals)	2393.408
188	RR Pant	16	16	3	488	78	37.53	300	162.66	0	3	0	37	27	(Delhi Capitals)	2390.379
44	JC Buttler	8	8	0	311	89	38.87	205	151.70	0	3	0	38	14	(Rajasthan Royals)	2341.574
182	MK Pandey	12	11	3	344	83	43.00	263	130.79	0	3	0	34	6	(Sunrisers Hyderabad)	2328.821
70	S Dhawan	16	16	1	521	97	34.73	384	135.67	0	5	1	64	11	(Delhi Capitals)	2295.340
240	SV Samson	12	12	2	342	102	34.20	230	148.69	1	0	1	28	13	(Rajasthan Royals)	2281.354
124	V Kohli	14	14	0	464	100	33.14	328	141.46	1	2	0	46	13	(Royal Challengers Bangalore)	2270.753

```
# Fetching the top 15 players
r2 <- r2[
  order( r2[,16], r2[,2] , decreasing = TRUE),
  ]
r2 <- head(r2,n=15)
str(r2)
rm(pdata)
pdata <- r2
```

Visualizing data

```
# Getting descriptive stats
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

descriptive <- function(coln){
  cat("Mean: ",mean(coln,na.rm=TRUE),"\n")
  cat("Median: ",median(coln,na.rm=TRUE),"\n")
  #cat("Mode: ",getmode(coln),"\n")
  cat("MAX: ",max(coln,na.rm=TRUE),"\n")
  cat("MIN: ",min(coln,na.rm=TRUE),"\n")
  cat("Mean: ",mean(coln,na.rm=TRUE),"\n")
  cat("Range: ",range(coln,na.rm=TRUE),"\n")
  cat("Variance: ",var(coln,na.rm=TRUE),"\n")
  cat("Standard Deviation: ",sd(coln,na.rm=TRUE),"\n")
  #cat("Scale: ",scale(coln),"\n")
  summary(coln)
}
```

```
> descriptive(pdata$runs)
Mean:  456.8667
Median:  445
MAX:  692
MIN:  311
Mean:  456.8667
Range:  311 692
Variance:  9943.267
Standard Deviation:  99.71593
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  311.0   397.5   445.0   456.9   500.0   692.0
```

```
> descriptive(pdata$hs)
Mean:  93.6
Median:  97
MAX:  114
MIN:  78
Mean:  93.6
Range:  78 114
Variance:  112.5429
Standard Deviation:  10.60862
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   78.0    83.5    97.0    93.6   100.0   114.0
```

```
> descriptive(pdata$avg)
Mean:  46.83267
Median:  43
MAX:  83.2
MIN:  32.75
Mean:  46.83267
Range:  32.75 83.2
Variance:  210.7719
Standard Deviation:  14.51799
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  32.75   36.13   43.00   46.83   54.76   83.20
```

```
> descriptive(pdata$bf)
Mean:  304.7333
Median:  287
MAX:  481
MIN:  205
Mean:  304.7333
Range:  205 481
Variance:  6173.495
Standard Deviation:  78.57159
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  205.0   256.0   287.0   304.7   323.5   481.0
```

```
> descriptive(pdata$sr)
Mean:  152.2527
Median:  148.69
MAX:  204.81
MIN:  130.79
Mean:  152.2527
Range:  130.79 204.81
Variance:  440.8457
Standard Deviation:  20.99633
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  130.8   136.8   148.7   152.3   155.6   204.8
```

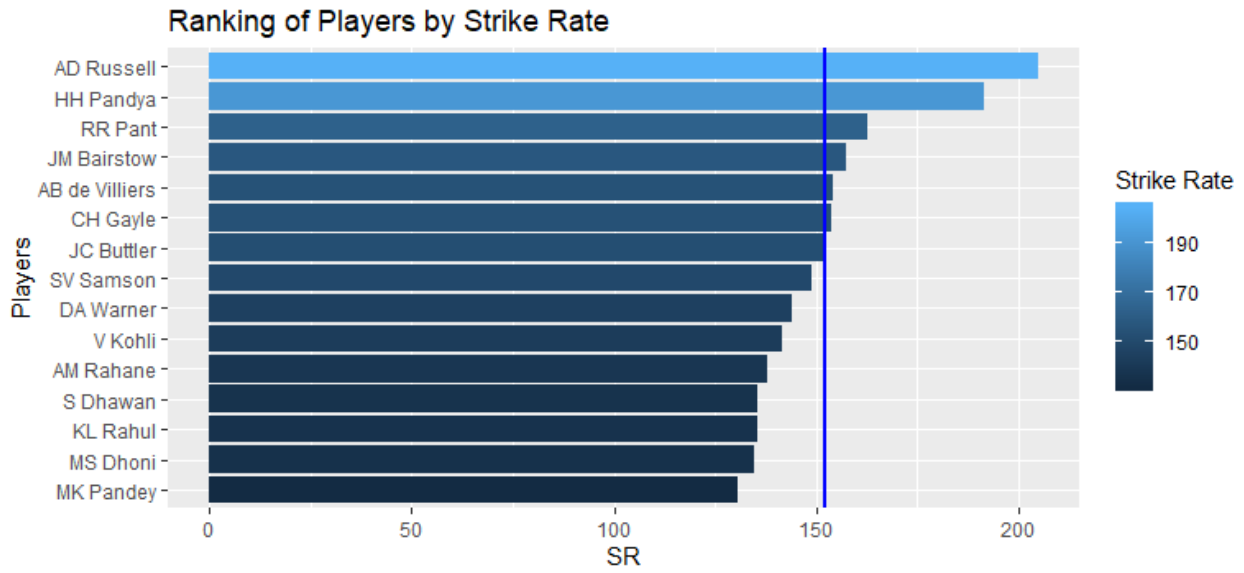
The above statistics described the measures of central tendency, i.e., mean, median and measures of dispersion, i.e., range and standard deviations of runs, high scores, average, balls faced and strike rates of the 15 players.

Ranking players by SR

```
# Ranking of Players by sr
ggplot(data=stat1,aes(x=reorder(player,sr),y=sr))+
  geom_bar(stat='identity',aes(fill=sr))+
  coord_flip() +
  theme_grey() +
  scale_fill_gradient(name="Strike Rate")+
  labs(title = 'Ranking of Players by Strike Rate',
       y='SR',x='Players')+
  geom_hline(yintercept = mean(stat1$sr),size = 1, color = 'blue')
```

The top 15 players are plotted with their strike rates.
A line intercepts the graph, this line is the mean of strike rates of these players.

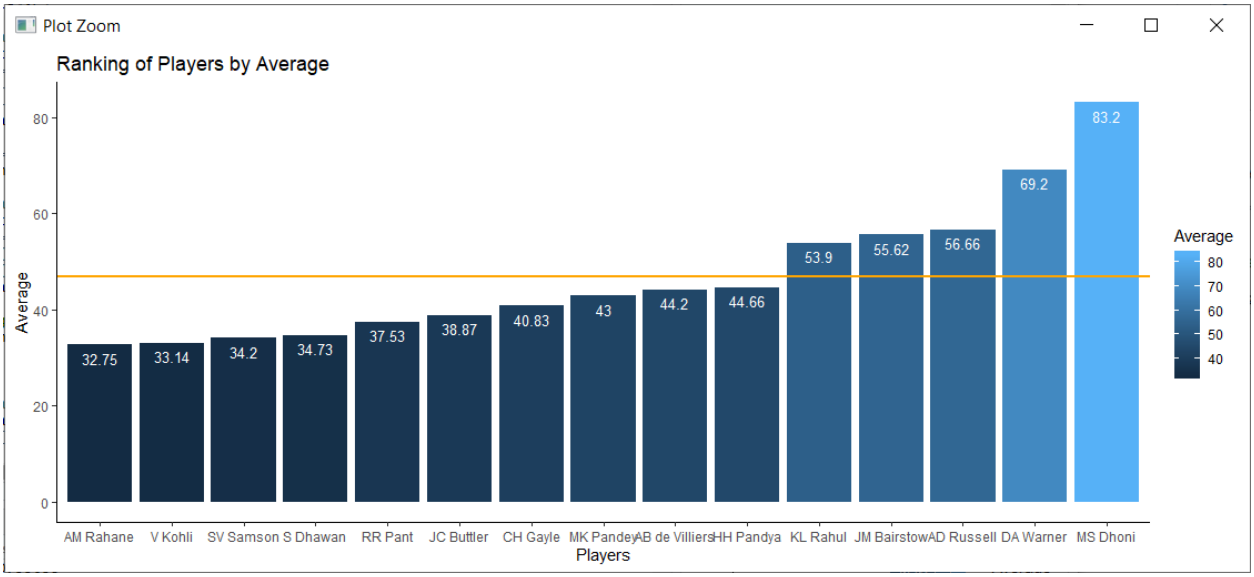
The graph as shown below.



AD Russell has the maximum Strike Rate

Ranking Players by their Averages

```
# Ranking of Players by avg
ggplot(data=stat1,aes(x=reorder(player,avg),y=avg))+
  geom_bar(stat='identity',aes(fill=avg))+
  geom_text(aes(label=avg), vjust=1.6, color="white", size=3.5)+
  theme_classic() +
  scale_fill_gradient(name="Average")+
  labs(title = 'Ranking of Players by Average',
       y='Average',x='Players')+
  geom_hline(yintercept = mean(stat1$avg),size = 1,color='orange')
```



MS Dhoni has the best average.

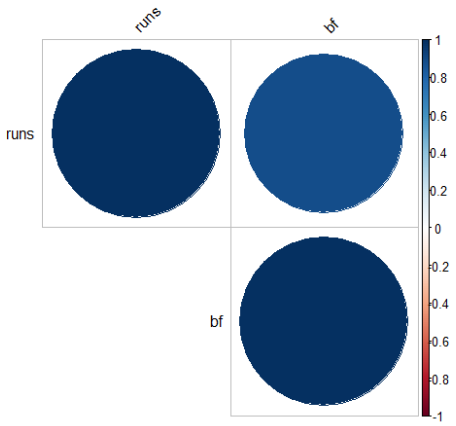
A line, mean (of the 15 players' avg) average passes through the graph.

Correlation Plots

```
# Between Runs and Balls Faced
stat2 <- as.data.frame(pdata[,c(1,5,8)])
res = cor(stat2[,-1])
res
corrplot(res, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
# Calucating p-value to see whether the correlation is significant
# the smaller the p-value, the more significant the correlation
res2 <- rcorr(as.matrix(stat2[,-1]))
res2
```

res output:

```
           runs        bf
runs 1.0000000 0.8829561
bf   0.8829561 1.0000000
```



Correlation plot between runs and balls. A high shade of blue depicts a high value of correlation.

```
> res2
     runs   bf
runs 1.00 0.88
bf   0.88 1.00

n= 15


P
     runs bf
runs       0
bf    0
```

Since the p-values are small (here, 0) we can understand that the correlation is significant.

```
# Runs and Average
X1 <- pdata$runs
Y1 <- pdata$avg
cov(X1,Y1)
cor(X1,Y1)
```

Covariance = 563.829 and Correlation = 0.3894722, i.e., positive and less than 0.5.

```
# Runs and Balls faced
X2 <- pdata$runs
Y2 <- pdata$bf
cov(X2,Y2)
cor(X2,Y2)
```

Covariance = 6917.819 and Correlation = 0.8829561, i.e., positive and greater than 0.5.

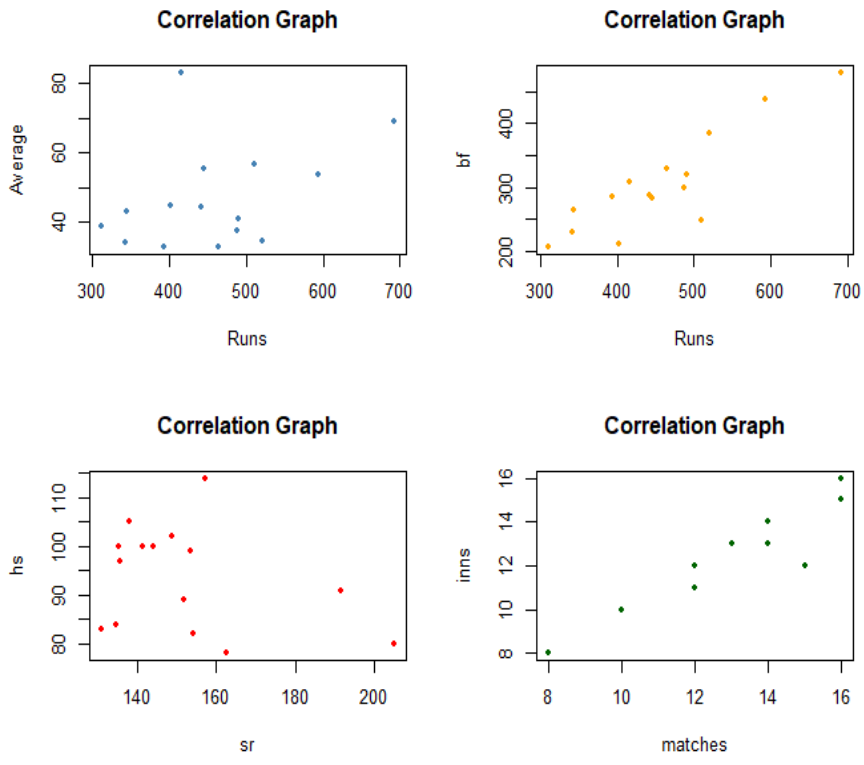
```
# SR and HS
X3 <- pdata$sr
Y3 <- pdata$hs
cov(X3,Y3)
cor(X3,Y3)
```

Covariance = -69.40814 and Correlation = -0.3116077, i.e., negative and greater than -0.5.

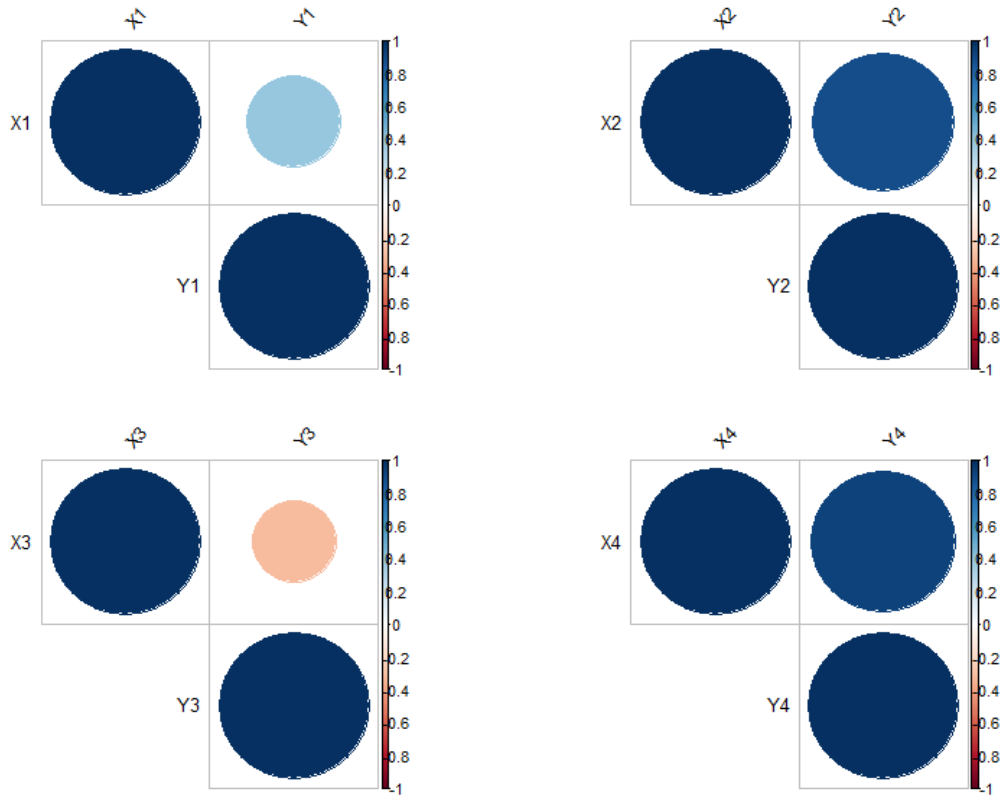
```
# SR and Average
X4 <- pdata$'matches'
Y4 <- pdata$'inns'
cov(X4,Y4)
cor(X4,Y4)
```

Covariance = 4.485714 and Correlation = 0.9291622, i.e., highly positive.

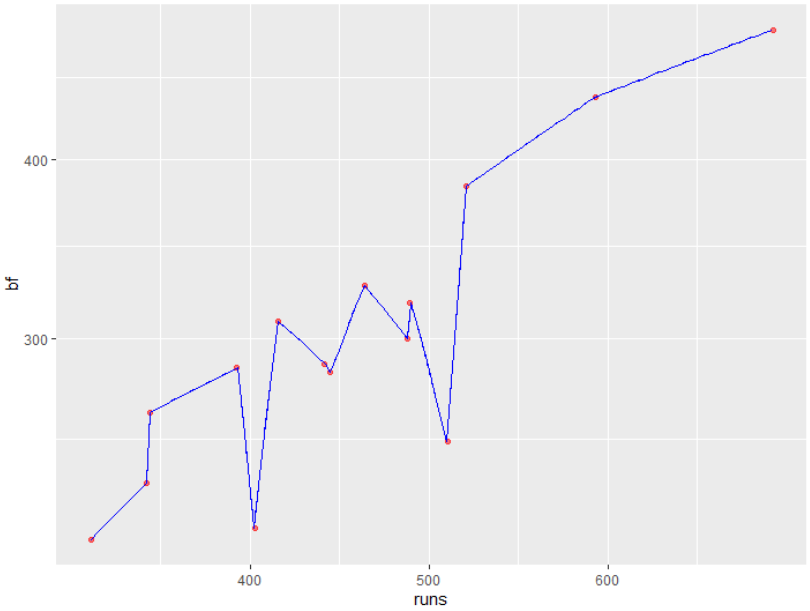
```
library(MASS)
plotstat <- cbind(X1,Y1)
# divide plot area as 2-by-2 array
par(mfrow = c(2, 2))
plot(plotstat, col = 'steelblue', pch = 20, xlab = 'Runs', ylab = 'Average',
     main = "Correlation Graph")

plotstat2 <- cbind(X2,Y2)
plot(plotstat2, col = 'orange', pch = 20, xlab = 'Runs', ylab = 'bf',
     main = "Correlation Graph")

plotstat3 <- cbind(X3,Y3)
plot(plotstat3, col = 'red', pch = 20, xlab = 'sr', ylab = 'hs',
     main = "Correlation Graph")

plotstat4 <- cbind(X4,Y4)
plot(plotstat4, col = 'green', pch = 20, xlab = 'matches', ylab = 'inns',
     main = "Correlation Graph")
```



The following is the correlation value plots for each correlation graphs, respectively (as given in the figure above)



Other graphs



Bowling Data

Data: bowling.csv

Bowling averages														
Player	Mat	Inns	Overs	Mdns	Runs	Wkts	BBI	Ave	Econ	SR	4	5	Ct	St
VR Aaron	5	5	12	1	116	4	20-Feb	29	9.66	18	0	0	1	0
(Rajasthan Royals)														
Abhishek S	3	2	2	0	21	1	10-Jan	21	10.5	12	0	0	1	0
(Sunrisers Hyderabad)														
AD Nath	8	-	-	-	-	-	-	-	-	-	-	-	1	0
(Royal Challengers Bangalore)														
MA Agarw	13	-	-	-	-	-	-	-	-	-	-	-	7	0
(Kings XI Punjab)														
KK Ahmed	9	9	34.5	0	287	19	30-Mar	15.1	8.23	11	0	0	0	0
(Sunrisers Hyderabad)														
MM Ali	11	9	25	0	169	6	18-Feb	28.16	6.76	25	0	0	1	0
(Royal Challengers Bangalore)														
JC Archer	11	11	43	2	291	11	15-Mar	26.45	6.76	23.4	0	0	3	0
(Rajasthan Royals)														
Arshdeep S	3	3	10	0	109	3	Feb-43	36.33	10.9	20	0	0	0	0
(Kings XI Punjab)														
M Ashwin	10	10	34	0	255	5	25-Feb	51	7.5	40.8	0	0	3	0
(Kings XI Punjab)														
R Ashwin	14	14	55	0	400	15	23-Mar	26.66	7.27	22	0	0	4	0
(Kings XI Punjab)														
Avesh Kha	1	1	3	0	30	0	-	-	10	-	0	0	1	0
(Delhi Capitals)														
JM Bairsto	10	-	-	-	-	-	-	-	-	-	-	-	9	2
(Sunrisers Hyderabad)														

Importing and cleaning data

```
setwd("C:/Users/Riya/Desktop/DA-lab2/espenIPL")

player <- read.csv(file = 'bowling.csv')[c(FALSE,TRUE),]
plyteam <- read.csv(file = 'bowling.csv')[c(TRUE,FALSE),]
plyteam <- plyteam[-1,]
pteam <- plyteam$i..Bowling.averages
player$team <- pteam
pdata <- as.data.frame(player)

rm(player,plyteam)

pdata[pdata=="-"] <- NA
pdata <- na.omit(pdata, cols = c(3))

library(ggplot2)
library(dplyr)

colnames(pdata) <- c("player","matches","inns","overs","mdns","runs","wkts","bbi","avg","econ","sr","4","5","ct","st","team")
dim(pdata)
```

Since the columns as in factor datatype they have to converted to numeric for manipulations. The function for cleaning and converting columns to numeric.

```
tonum <- function(coln){
  temp <- stringr::str_replace(coln,'\\ ','')
  temp <- as.numeric(temp)
  return(temp)
}
```

Function for descriptive statistics

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

descriptive <- function(coln){
  cat("Mean: ",mean(coln,na.rm=TRUE),"\n")
  cat("Median: ",median(coln,na.rm=TRUE),"\n")
  #cat("Mode: ",getmode(coln),"\n")
  cat("MAX: ",max(coln,na.rm=TRUE),"\n")
  cat("MIN: ",min(coln,na.rm=TRUE),"\n")
  cat("Mean: ",mean(coln,na.rm=TRUE),"\n")
  cat("Range: ",range(coln,na.rm=TRUE),"\n")
  cat("Variance: ",var(coln,na.rm=TRUE),"\n")
  cat("Standard Deviation: ",sd(coln,na.rm=TRUE),"\n")
  #cat("Scale: ",scale(coln),"\n")
  summary(coln)
}
```

```
pdata$matches <- tonum(pdata$matches)
descriptive(pdata$matches)
pdata$inns <- tonum(pdata$inns)
descriptive(pdata$inns)
pdata$overs <- tonum(pdata$overs)
descriptive(pdata$overs)
pdata$mdns <- tonum(pdata$mdns)
descriptive(pdata$mdns)
pdata$runs <- tonum(pdata$runs)
descriptive(pdata$runs)
pdata$wkts <- tonum(pdata$wkts)
descriptive(pdata$wkts)
pdata$avg <- tonum(pdata$avg)
descriptive(pdata$avg)
pdata$econ <- tonum(pdata$econ)
pdata$sr <- tonum(pdata$sr)
pdata$ct <- tonum(pdata$ct)
```

```
> descriptive(pdata$matches)
Mean:  8.218391
Median:  8
MAX:  17
MIN:  1
Mean:  8.218391
Range:  1 17
Variance:  22.73082
Standard Deviation:  4.767685
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.500   8.000   8.218  12.000  17.000
```

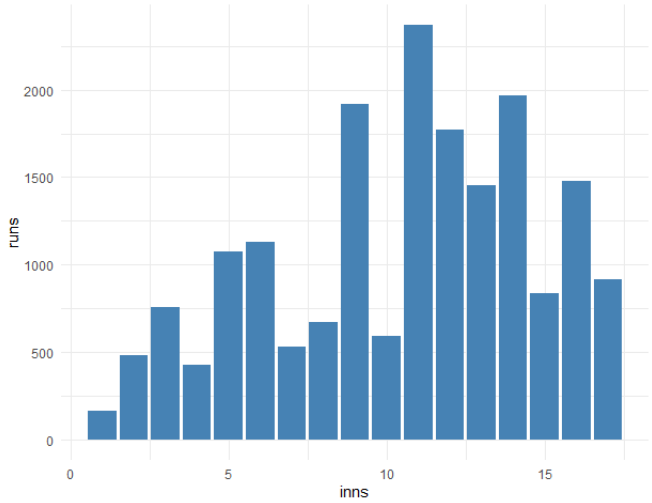
```
> descriptive(pdata$inns)
Mean:  7.632184
Median:  7
MAX:  17
MIN:  1
Mean:  7.632184
Range:  1 17
Variance:  22.70035
Standard Deviation:  4.764488
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   7.000   7.632  11.500  17.000
```

```
> descriptive(pdata$overs)
Mean:  25.84943
Median:  23
MAX:  64.3
MIN:  2
Mean:  25.84943
Range:  2 64.3
Variance:  334.0279
Standard Deviation:  18.27643
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    8.70   23.00   25.85   42.35   64.30
```

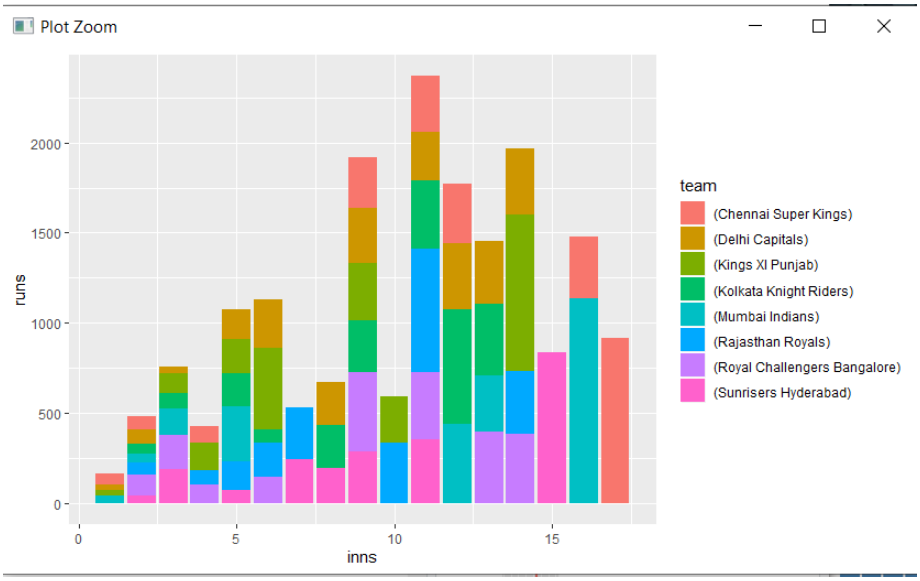
```
> descriptive(pdata$mdns)
Mean:  0.2298851
Median:  0
MAX:  2
MIN:  0
Mean:  0.2298851
Range:  0 2
Variance:  0.2256081
Standard Deviation:  0.4749822
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.2299  0.0000  2.0000
```

Similarly, for other columns these data were outputted.

Bar plot inns vs runs for all players had no meaning, just the addition of runs of the players who played same number of inns



Bar plot inns vs runs is more descriptive when teams were indicated



Now fetching the top 15 players

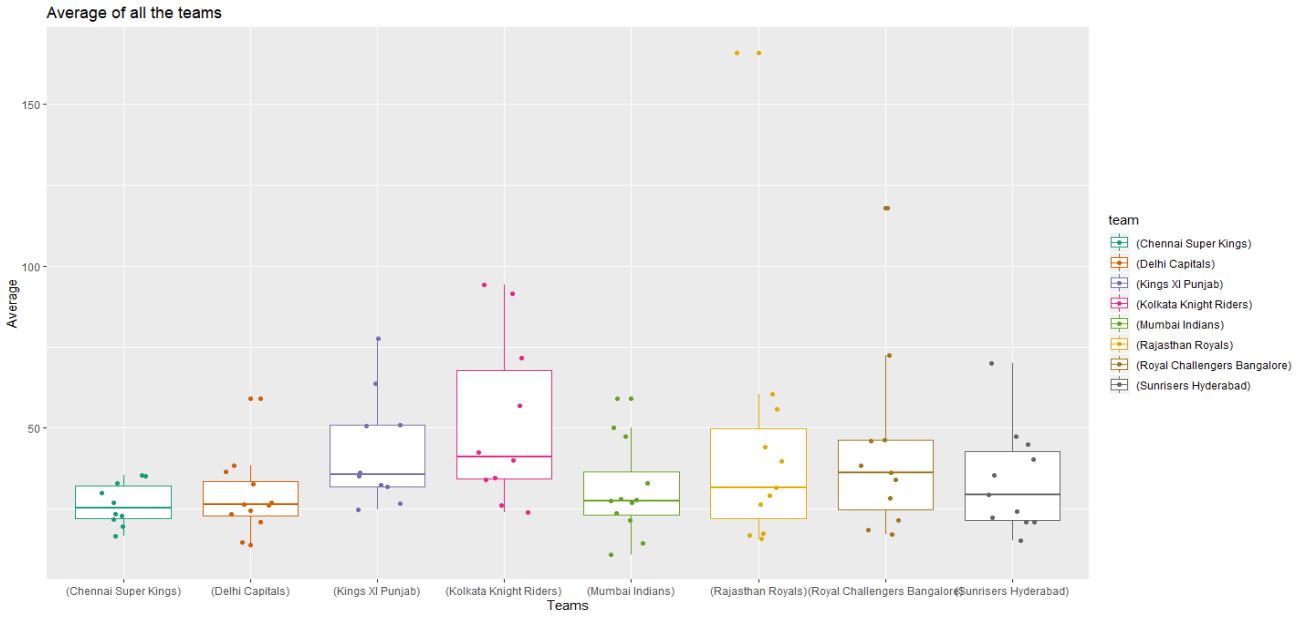
```
# Giving weights to the attributes
w1 <- 5 #econ
w2 <- 4 #sr
w3 <- 6 #wkts
w4 <- 7 #mdns
w5 <- 2 #overs
w6 <- 2 #ct

# Calulating scores
pdata$score <- as.numeric(pdata$econ)/w1 + as.numeric(pdata$sr)*w2 +
  as.numeric(pdata$wkts)*w3 + as.numeric(pdata$mdns)*w4 +
  as.numeric(pdata$overs)*w5 + as.numeric(pdata$ct)*w6

# Fetching the top 15 players
pdata <- pdata[
  order( pdata[,17], pdata[,2] , decreasing = TRUE),
  ]
pdata <- head(pdata,n=15)
```

Descriptive analysis

```
# Number of players
pdata %>% summarise(pdata_count = n())
pdata_count
         87

# Number of teams
team_count = length(unique(pdata$team))
team_count
.] 8
```

```
> # which player wins with lowest economy
> max_runs = pdata[which.min(pdata$econ),]
> max_runs %>% select('player','econ')
    player econ
230 AS Roy  5.5
```

Box plotting bowler's data average with respect to teams

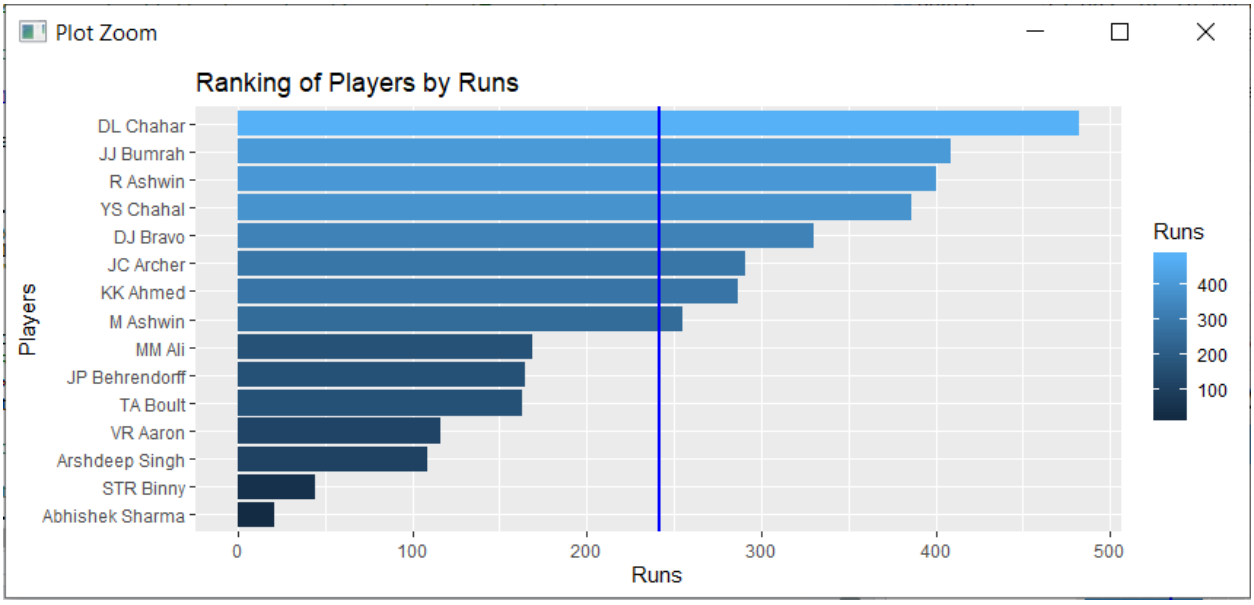


The performance of Kolkata Knight Riders is better among the others.

Ranking of top 15 players by runs

```
stat1 <- as.data.frame(pdata[,c(1,6)])
# Ranking of Players by runs
ggplot(data=stat1,aes(x=reorder(player,runs),y=runs))+
  geom_bar(stat='identity',aes(fill=runs))+
  coord_flip() +
  theme_grey() +
  scale_fill_gradient(name="Runs")+
  labs(title = 'Ranking of Players by Runs',
       y='Runs',x='Players')+
  geom_hline(yintercept = mean(stat1$runs),size = 1, color = 'blue')
```

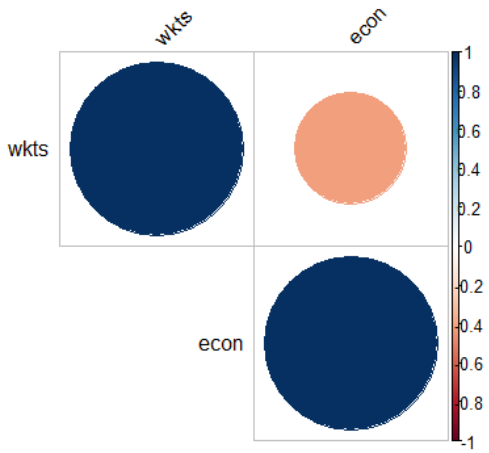
Mean of runs is cutting the graph.



Correlation plots

Between wickets and economy

```
> # Between wickects and Ecomony
> stat2 <- as.data.frame(pdata[,c(1,7,10)])
> res = cor(stat2[,-1])
> res
            wkts       econ
wkts  1.0000000 -0.4154392
econ -0.4154392  1.0000000
```



Given is the correlation plot between wickets and economy. The red color indicates a negative correlation, which is a correct indication since the greater number of wickets taken by a bowler in the least number of over, i.e. a lower economy of the bowler makes him a better bowler. A negative value shows that if the economy of player increases it means his wickets taken decreased.

Calculating p-value to see whether the correlation is significant the smaller the p-value, the more significant the correlation.

```
> res2 <- rcorr(as.matrix(stat2[,-1]))
> res2
      wkts  econ
wkts  1.00 -0.42
econ -0.42  1.00

n= 15


P
     wkts   econ
wkts        0.1236
econ 0.1236
```

Low p-values indicates the correlation is significant.

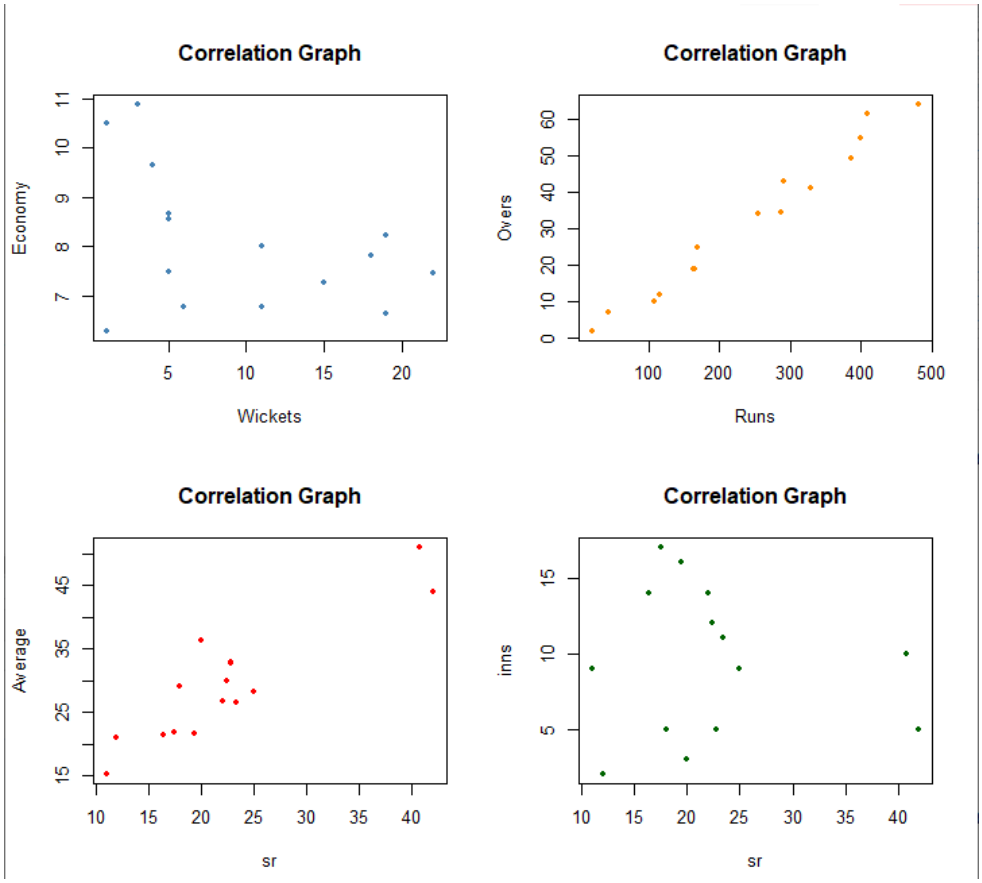
```
> # Wickects and Economy
> X1 <- pdata$wkts
> Y1 <- pdata$econ
> cov(X1,Y1)
[1] -4.190238
> cor(X1,Y1)
[1] -0.4154392
```

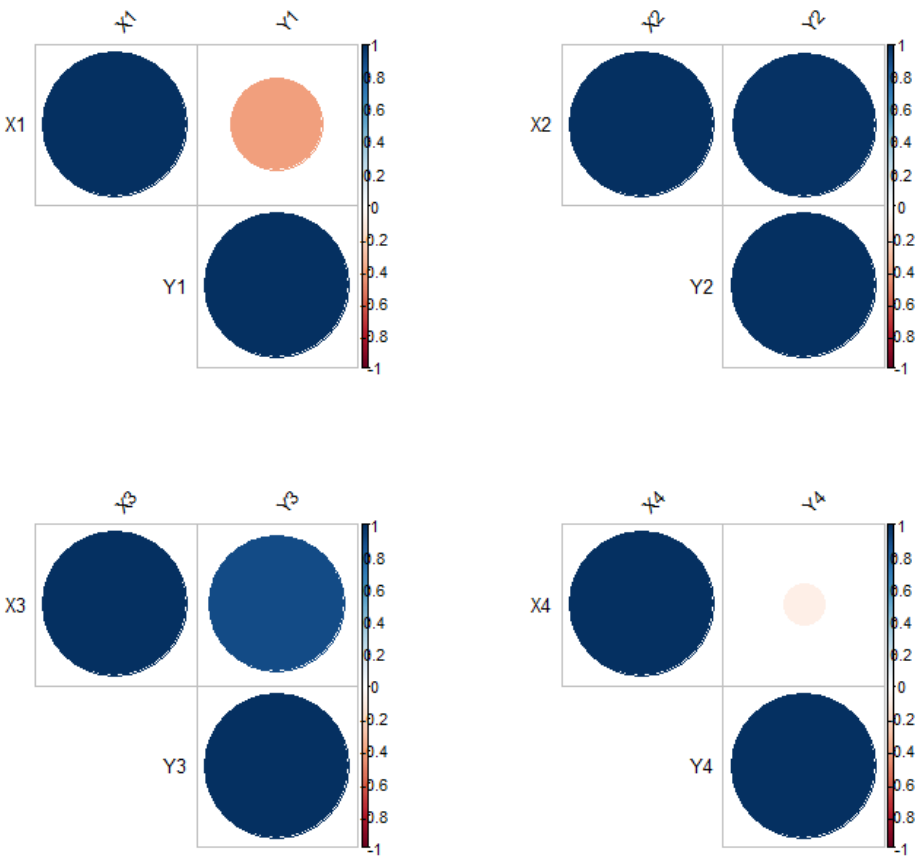
```
> # Runs and Overs
> X2 <- pdata$runs
> Y2 <- pdata$overs
> cov(X2,Y2)
[1] 2843.943
> cor(X2,Y2)
[1] 0.9883559
```

```
> # SR and Average
> X3 <- pdata$sr
> Y3 <- pdata$avg
> cov(X3,Y3)
[1] 73.20038
> cor(X3,Y3)
[1] 0.8978232
```

```
> # SR and Inns
> X4 <- pdata$sr
> Y4 <- pdata$inns
> cov(X4,Y4)
[1] -3.466667
> cor(X4,Y4)
[1] -0.08247131
```

Correlation between wickets and economy is negative and value greater than -0.5.
Correlation between runs and overs is highly positive.
Correlation between sr and average is positive and value greater than 0.5.
Correlation between sr and inns is negative and value less than -0.5.





The above two figures were correlation and their corresponding correlation value graphs for the four pairs considered previously.