DA Lab 2

Analysis of IPL-2019 Dataset

Name: Dheeraj Chaudhary

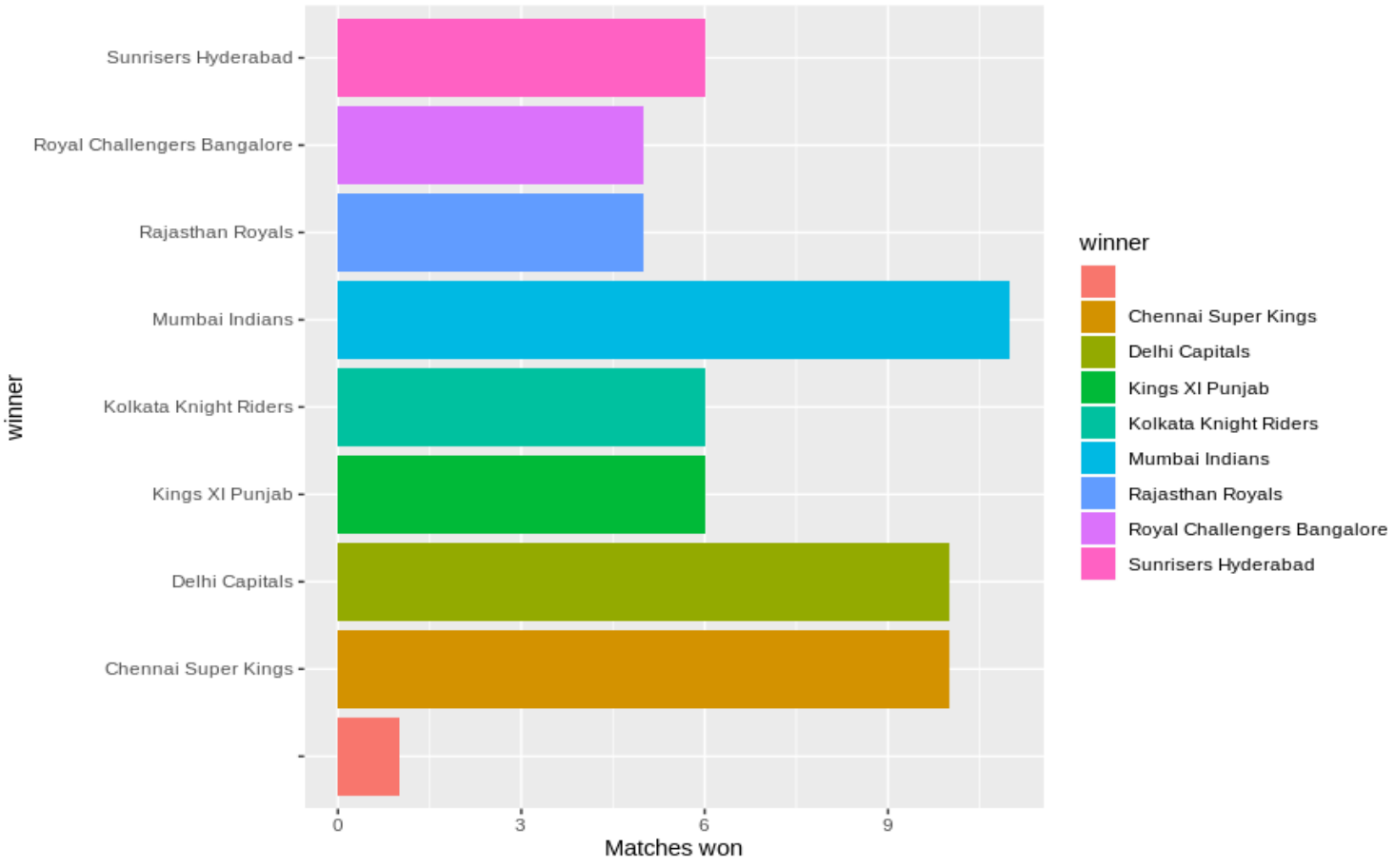
Roll: 17BCS009

```
library(tabulizer)
library(dplyr)
library(ggplot2)
library(reshape2)
library(magrittr)
library(tidyr)
```

```
#########################   READING MATCHES CSV FILE ######################
matches <- read.csv("/home/dheeraj/Desktop/Lecture/6th_sem_Academics/DataScience/Lab2/
matches.csv", stringsAsFactors = FALSE)
data <- read.csv("/home/dheeraj/Desktop/Lecture/6th_sem_Academics/DataScience/Lab2/
deliveries.csv", stringsAsFactors = FALSE)
matches <- matches[,-18]
data$wickets <- as.numeric(ifelse(data$player_dismissed =="" ,"",1))
```

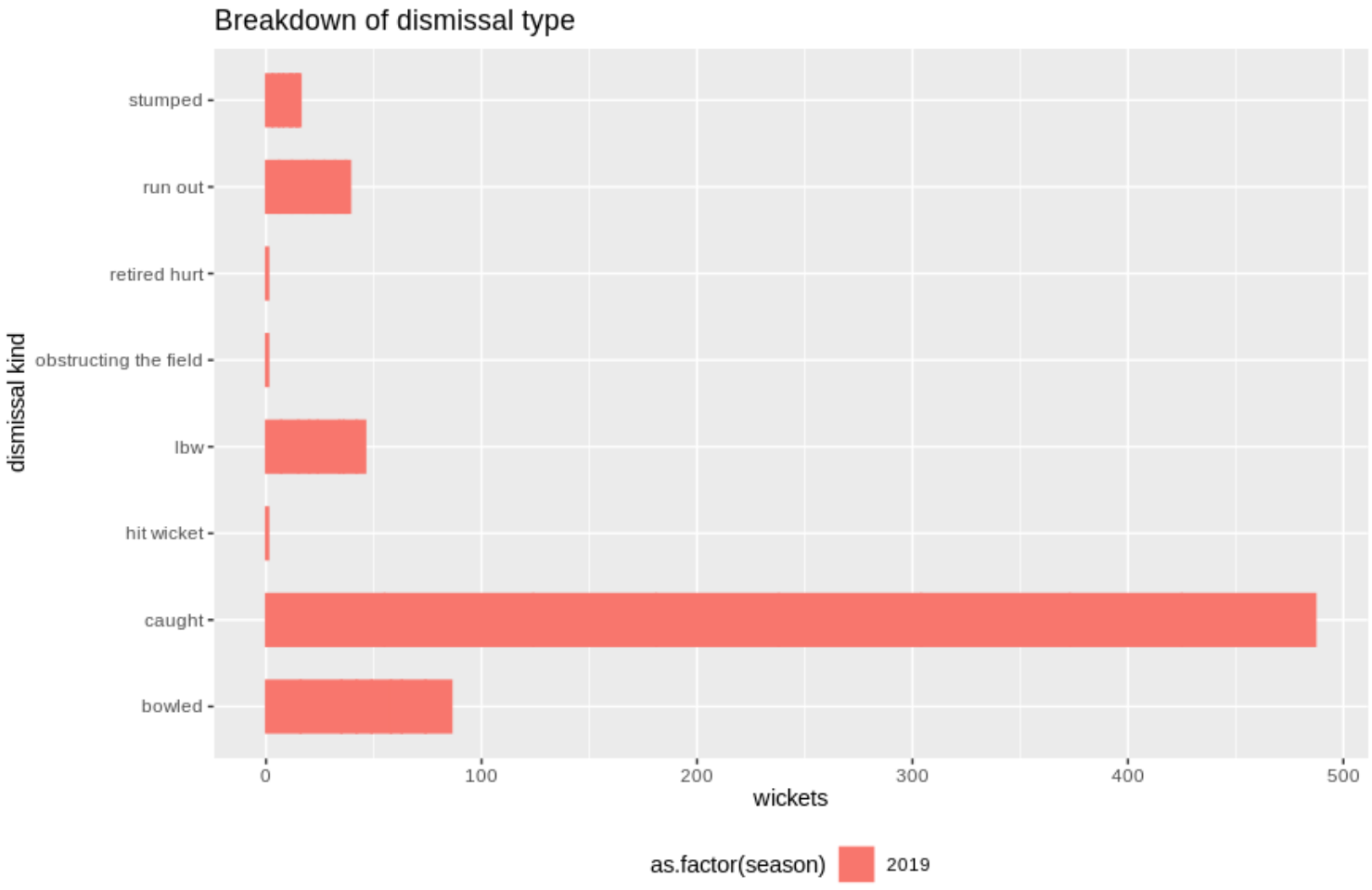
```
######## Number of matches in the dataset   (We can see 60 matches were played in IPL'2019)
summarize(matches,no_of_matches = n())
##### OUTPUT >   no_of_matches  60
```

```
######### Which Team won by maximum runs?    (We can see SRH won y 118 runs)
max_run <- matches[which.max(matches$win_by_runs),]
select(max_run, winner, win_by_runs)
######  Output >            winner win_by_runs
                     11 Sunrisers Hyderabad         118

######### Which Team won by maximum wickets?   (We ca see SRH won by 9 wickets)
max_run <- matches[which.max(matches$win_by_wickets),]
select(max_run, winner, win_by_wickets)
###### Output > winner win_by_wickets
                     38 Sunrisers Hyderabad              9

############################ Teams and matches won  (We can see MI wo maximum matches)
matches%>%
  group_by(winner)%>%
  summarize(most_win = n())%>%
  ggplot(aes(x = winner,y = most_win,fill = winner))+
  geom_bar(stat = "identity")+
  coord_flip()+
  scale_y_continuous("Matches won")
```



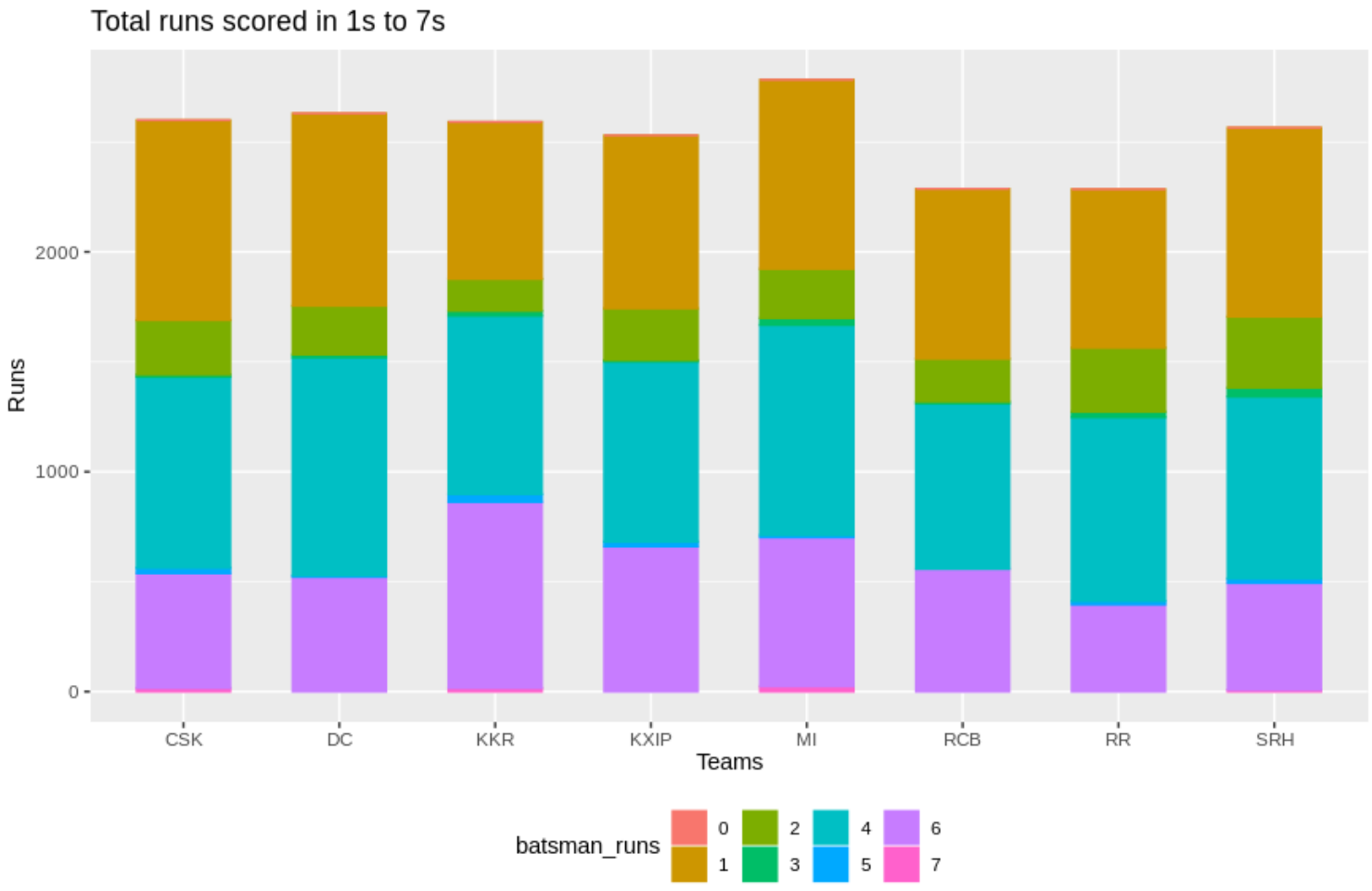
```r
teams <- data %>% select(batting_team)%>%
  distinct()
teams <- rename(teams, team = batting_team)
teams
###### Output >              team           (following teams played in IPL 2019)
          1 Royal Challengers Bangalore
          2         Chennai Super Kings
          3         Sunrisers Hyderabad
          4       Kolkata Knight Riders
          5              Delhi Capitals
          6              Mumbai Indians
          7             Kings XI Punjab
          8            Rajasthan Royals




s_team <- c("RCB","CSK","SRH","KKR","DC","MI","KXIP","RR")
s_team
#####  OUTPUT > [1] "RCB"  "CSK"  "SRH"  "KKR"  "DC"   "MI"   "KXIP" "RR"



teams <- cbind(teams, s_team)
player_of_match <- matches%>% select(id,player_of_match,season) %>%
  distinct()
player_of_match <- rename(player_of_match, player=player_of_match)
matches$city <- as.character(matches$city)
matches$city[matches$city==""] <- "Dubai"
venue_city <- matches %>%
  select(city)%>%
  distinct()
```

```
############### Dissmissal type and number of dismissal#############################
dismissal <- data%>%
  left_join(matches, by=c("match_id"="id"))%>%
  left_join(teams,by=c("batting_team"="team"))%>%
  filter(dismissal_kind!="")%>%
  group_by(season,dismissal_kind,s_team)%>%
  summarize(wickets =n())
ggplot(dismissal,aes(x=dismissal_kind,y=wickets,colour=as.factor(season),
fill=as.factor(season)))+
  geom_bar(position = "stack", show.legend = TRUE, width =.6,stat="identity")+
  theme(legend.position="bottom")+
  coord_flip()+
  theme(legend.direction = "horizontal") +
  scale_y_continuous(name="wickets")+
  scale_x_discrete(name="dismissal kind")+
  ggtitle("Breakdown of dismissal type ")
```



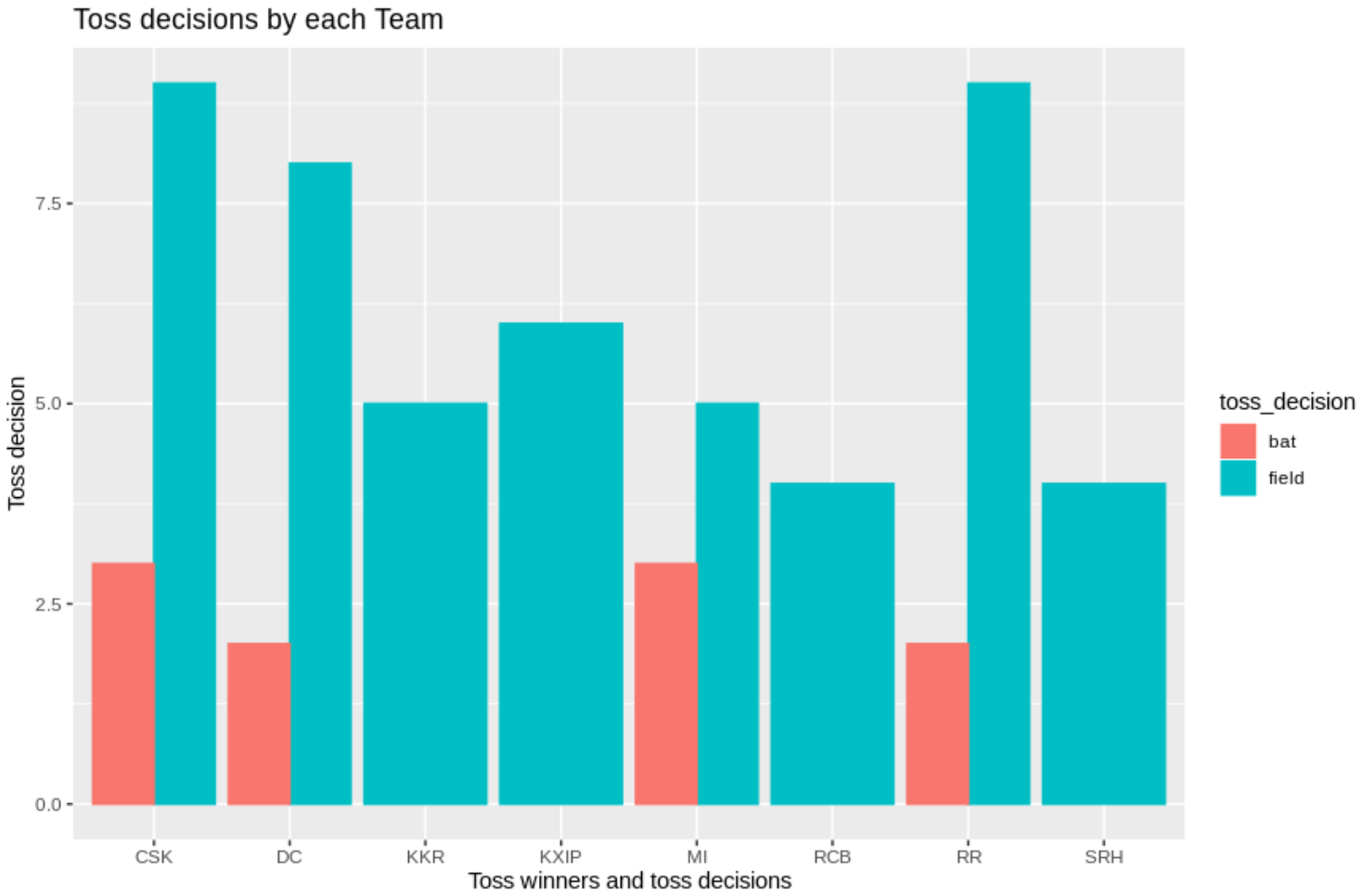
(We ca see in above plot that maximum dismissal was happened due to caught)

```
######################## Run scored in 1s to 7s  ####################
runs_cat <- data %>%
  left_join(matches,by=c("match_id"="id"))%>%
  left_join(teams,by=c("batting_team"="team"))%>%
  group_by(s_team,batsman_runs)%>%
  summarize(no=n(),runs=sum(total_runs))

runs_cat$batsman_runs <- as.factor(runs_cat$batsman_runs)

ggplot(runs_cat,aes(x=s_team,y=runs,colour=batsman_runs,fill=batsman_runs))+
  geom_bar(position = "stack", show.legend = TRUE, width =.6,stat="identity")+
  theme(legend.position="bottom")+
  theme(legend.direction = "horizontal") +
  scale_y_continuous(name="Runs")+
  scale_x_discrete(name="Teams")+
  ggtitle("Total runs scored in 1s to 7s")
```



(We ca see in above plot that most of the runs were scored in 1st, 3rd and 6th ball)

```
##################### toss decision of toss winner ######################
wins_1 <- matches%>%
  left_join(teams,by=c("toss_winner"="team") )%>%
  select(s_team,toss_winner,toss_decision)%>%
  group_by(s_team,toss_decision)%>%
  summarize(wins=n())



ggplot(wins_1,aes(x=s_team,y=wins,colour=toss_decision,fill=toss_decision))+
  geom_bar(position = "dodge",stat = "identity")+
  theme(legend.position="right")+
  scale_y_continuous(name="Toss decision")+
  scale_x_discrete(name="Toss winners and toss decisions")+
  ggtitle("Toss decisions by each Team")
```

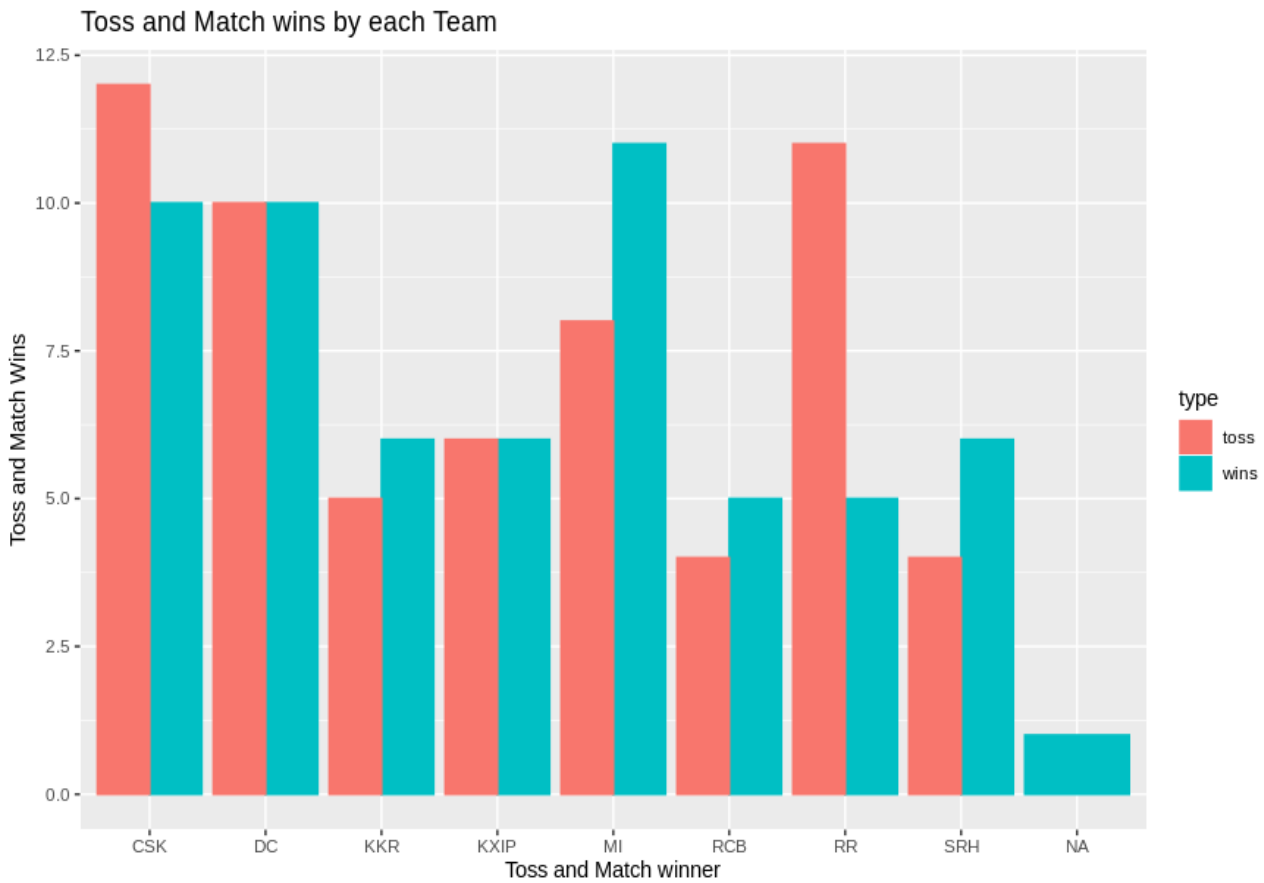


(We can see that CSK and RR choosen fielding after winning the toss and KKR, KXIP, SRH never batted first after winning the toss)

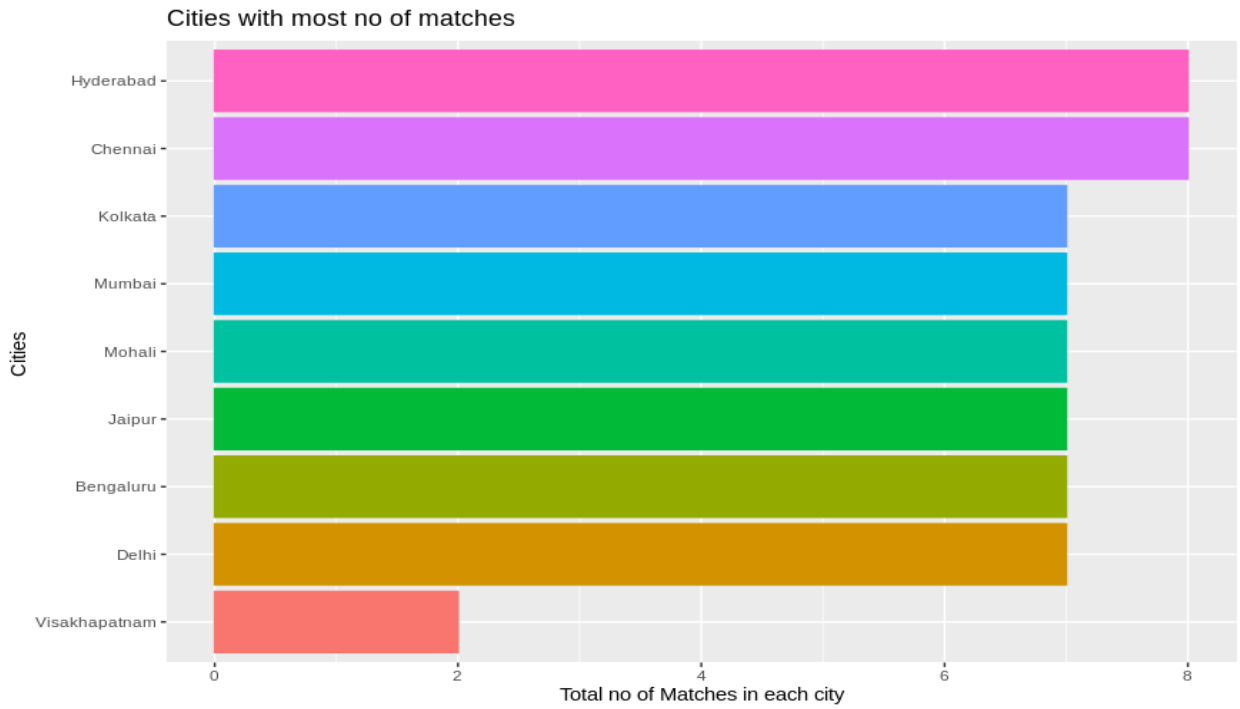
```
############################## Toss and match win  ###################################
toss <- matches%>%
  left_join(teams,by=c("toss_winner"="team") )%>%
```

```
  select(s_team,toss_winner)%>%
  group_by(s_team)%>%
  summarize(wins=n())
toss$type <- "toss"
wins <-matches%>%
  left_join(teams,by=c("winner"="team") )%>%
  select(s_team,winner)%>%
  group_by(s_team)%>%
  summarize(wins=n())
wins$type <- "wins"
toss_w <- rbind(toss,wins)
toss_w <- toss_w %>%
  group_by(s_team, type)%>%
  summarize(wins=sum(wins))
ggplot(toss_w,aes(x=s_team,y=wins,colour=type,fill=type))+
  geom_bar(position = "dodge",stat = "identity")+
  theme(legend.position="right")+
  scale_y_continuous(name="Toss and Match Wins")+
  scale_x_discrete(name="Toss and Match winner")+
  ggtitle("Toss and Match wins by each Team")
```

(We can see in the below plot that DC and XXIP won every match when they won the toss)



```
######################## city with most number of match  ###################
venue_c <- data%>%
  left_join(matches,by=c("match_id"="id"))%>%
  select(match_id,city,total_runs,wickets)%>%
  group_by(city)%>%
  summarize(runs=sum(total_runs),wickets=sum(wickets,na.rm=TRUE))
city_mat <- matches %>%
  group_by(city)%>%
  summarize(matches=n())
venue_c <- venue_c %>%
  left_join(city_mat, by=c("city"="city"))%>%
  mutate(Avg_runs=runs/matches)%>%
  mutate(Avg_wkt =wickets/matches)%>%
  arrange(city)
venue_all <- venue_c%>%
  left_join(venue_city, by=c("city"="city"))%>%
  arrange(Avg_runs)
venue_all$city <- factor(venue_all$city, levels = venue_all$city[order(venue_all$matches)])
ggplot(venue_all,aes(x=city,y=matches,colour=city,fill=city))+
  geom_bar(position = "dodge",stat = "identity")+
  theme(legend.position="none")+ coord_flip()+
  scale_y_continuous(name="Total no of Matches in each city")+
  scale_x_discrete(name="Cities ")+
 ggtitle("Cities with most no of matches")
```



(We can see in the above plot that most of the maximum of 9 matches were played in Chennain and Hyderabad)

Analysis on Batsman of IPL 2019 by giving priorities to their performance measures

```
############### READIG FILE ###################################


most_runs <- read.csv("/home/dheeraj/Desktop/Lecture/6th_sem_Academics/DataScience/Lab2/
batting_stats.csv")
head(most_runs)


############## Ordering According to Priority ################
a <- most_runs[order(-most_runs$RUNS),]
b <- a[order(-most_runs$AVG),]
c <- b[order(most_runs$INN),]
d <- c[order(-most_runs$SR),]
e <- d[order(-most_runs$X4S),]
```

(I gave priority in the following order by highest runs scored, maximum avg of the player, minimum innings played, highest Strike rate, maximum number of fours)

```
############# Player who topped the list ###################
select(head(e, n=1), PLAYER, RUNS)
####### Output >     PLAYER
               David Warner

########### Player who scored maximum runs #################
max_run <- e[which.max(e$RUNS),]
select(max_run, PLAYER)
####### Output >      PLAYER
               David Warner


########### Player who've highest Strike Rate ##############
max_sr <- e[which.max(e$SR),]
select(max_sr, PLAYER)
####### Output >     PLAYER
               Andre Russell
```

```
############## Player who hit highest 4rs ###################
max_fours <- e[which.max(e$X4S),]
select(max_fours, PLAYER)
####### Output >       PLAYER
               Shikhar Dhawan


############## Player who've highest Average ###################
max_avg <- e[which.max(e$AVG),]
select(max_avg, PLAYER)
####### Output >     PLAYER
                  MS Dhoni


############## Player who hit highest Sixes ###################
max_sixes <- e[which.max(e$X6S),]
select(max_sixes, PLAYER)
####### Output >       PLAYER
                   Andre Russell

############## Player who played minimum match ###############
min_match <- e[which.min(e$MATCHES),]
select(min_match, PLAYER)
####### Output >       PLAYER
               K Khaleel Ahmed


############## Top ten player's name in my list ###################
select(head(e, n=10), PLAYER)
####### Output >         PLAYER
              1     David Warner
              2     Lokesh Rahul
              3   Shikhar Dhawan
              4  Jonny Bairstow
              5     Shreyas Iyer
              6  Ajinkya Rahane
              7  Quinton de Kock
              8   Hardik Pandya
              9         MS Dhoni
              10    Shane Watson
```

```
########## Top ten player's with their data in my list ###########

select(head(e, n=10), PLAYER,INN, RUNS,   AVG,     SR ,X4S, X6S )

####### Output >       PLAYER INN RUNS   AVG     SR X4S X6S

          1     David Warner  12  692 69.20 143.87  57  21
          2     Lokesh Rahul  14  593 53.91 135.39  49  25
          3   Shikhar Dhawan  16  521 34.73 135.68  64  11
          4  Jonny Bairstow  10  445 55.62 157.24  48  18
          5     Shreyas Iyer  16  463 30.87 119.95  41  14
          6  Ajinkya Rahane  13  393 32.75 137.89  45   9
          7  Quinton de Kock  16  529 35.27 132.91  45  25
          8   Hardik Pandya  15  402 44.67 191.43  28  29
          9        MS Dhoni  12  416 83.20 134.63  22  23
          10    Shane Watson  17  398 23.41 127.56  42  20
```

Analysis on Bowlers of IPL 2019 by giving priorities to their performance measures

```
############### READIG FILE ###################################


bowling_stats <- read.csv("/home/dheeraj/Desktop/Lecture/6th_sem_Academics/DataScience/Lab2/
bowling_stats.csv")

head(bowling_stats)


############## Ordering According to Priority ################

a <- bowling_stats[order(-bowling_stats$WKTS),]

b  <- a[order(bowling_stats$BALLS),]

c <- b[order(bowling_stats$MATCHES),]

e <- c[order(-bowling_stats$RUNS),]
```

(Priority given in the following order by maximum wickets taken, minimum balls throw by him , minimum match played, and runs given by him)

```
############# Player who topped the list ##################

select(head(e, n=1), PLAYER)

####### Output >       PLAYER

                Imran Tahir

########### Player who taken maximum wicket ###############

max_wkt <- e[which.max(e$WKTS),]

select(max_wkt, PLAYER)

####### Output >     PLAYER

                 Imran Tahir
```

```
########### Player who've thrown maximum balls ##############
max_run <- e[which.max(e$BALLS),]
select(max_run, PLAYER)
####### Output >        PLAYER
                 Deepak Chahar

############# Player who gave minimum runs #################
max_run <- e[which.min(e$RUNS),]
select(max_run, PLAYER)
####### Output >     PLAYER
                Amit Mishra


############# Player who played minimum match ###############
max_run <- e[which.min(e$MATCHES),]
select(max_run, PLAYER)
####### Output >      PLAYER
               K Khaleel Ahmed

############# Top ten player's name in my list ###################
select(head(e, n=10), PLAYER)
####### Output >       PLAYER
              1      Imran Tahir
              2      Axar Patel
              3  K Khaleel Ahmed
              4   Mohammed Shami
              5     Amit Mishra
              6   Navdeep Saini
              7   Ishant Sharma
              8    Shreyas Gopal
              9      Sam Curran
              10    Kagiso Rabada

########## Top ten player's with their data in my list ###########
select(head(e, n=10), PLAYER, MATCHES, BALLS, RUNS, WKTS )
####### Output >       PLAYER MATCHES BALLS RUNS WKTS
          1      Imran Tahir      17   386  431   26
          2      Axar Patel       14   306  364   10
          3  K Khaleel Ahmed       9   209  287   19
          4   Mohammed Shami      14   324  469   19
          5     Amit Mishra       11   240  270   11
          6   Navdeep Saini       13   288  397   11
          7   Ishant Sharma       13   276  349   13
          8    Shreyas Gopal      14   288  347   20
          9      Sam Curran        9   198  323   10
          10    Kagiso Rabada     12   282  368   25
```

Analysis of 10 Players and their 10 match individual scores finding median, consistency, corelation between few pairs of Players

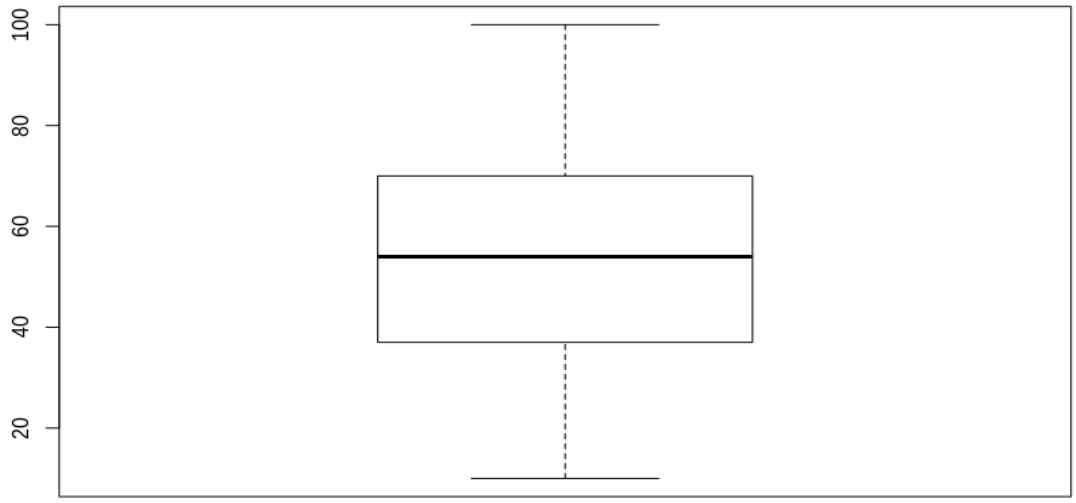
```
########  loading data of top 10 players and their individual 10 match score ######

ind_ply <- read.csv("/home/dheeraj/Desktop/Lecture/6th_sem_Academics/DataScience/Lab2/
player_eachmatch.csv")

View(ind_ply)

median(ind_ply$DavidWarner)

####### Output >  [1] 54


median(ind_ply$LokeshRahul)

####### Output > [1] 33.5


median(ind_ply$ShikharDhawan)

####### Output > [1] 32.5


median(ind_ply$JonnyBairstow)

####### Output > [1] 43

median(ind_ply$ShreyasIyer)

####### Output > [1] 35.5


median(ind_ply$AjinkyaRahane)

####### Output > [1] 20


median(ind_ply$QuintondeKock )

####### Output > [1] 31


median(ind_ply$HardikPandya)

####### Output > [1] 26.5


median(ind_ply$MSDhoni)

####### Output > [1] 37


median(ind_ply$ShaneWatson )

####### Output > [1] 9.5
```

Below is consitency of each players (player which have values near to 0 are more consistent and which’re near to 1 are inconsistent)

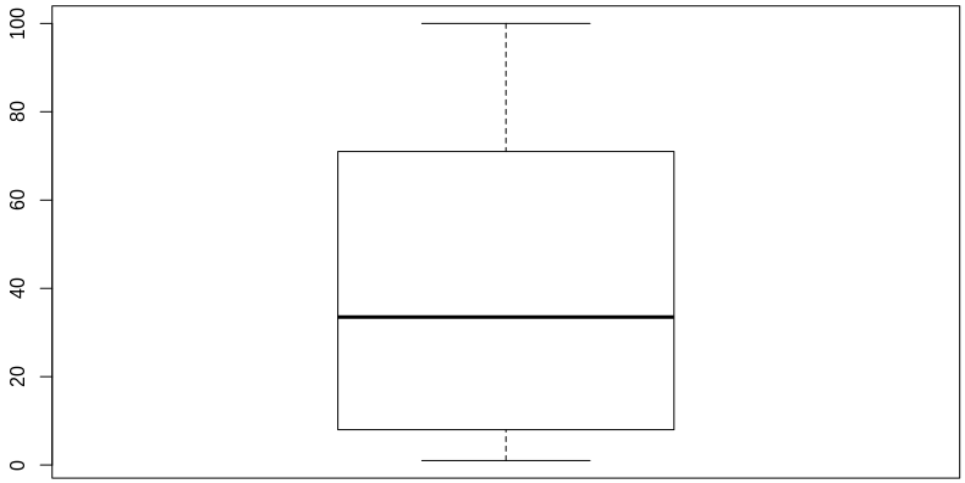
```
######### Finding Coefficient of variance ###########
sd(ind_ply$LokeshRahul)/mean(ind_ply$DavidWarner)
####### Output > [1] 0.6536159

sd(ind_ply$LokeshRahul)/mean(ind_ply$LokeshRahul)
####### Output > [1] 0.9039726

sd(ind_ply$ShikharDhawan)/mean(ind_ply$ShikharDhawan)
####### Output > [1] 0.8334738

sd(ind_ply$JonnyBairstow)/mean(ind_ply$JonnyBairstow)
####### Output > [1] 0.7892764

sd(ind_ply$ShreyasIyer)/mean(ind_ply$ShreyasIyer)
####### Output > [1] 0.66783

sd(ind_ply$AjinkyaRahane)/mean(ind_ply$AjinkyaRahane)
####### Output > [1] 1.052911

sd(ind_ply$QuintondeKock )/mean(ind_ply$QuintondeKock )
####### Output > [1] 0.6318267

sd(ind_ply$HardikPandya)/mean(ind_ply$HardikPandya)
####### Output > [1] 0.4513528

sd(ind_ply$MSDhoni)/mean(ind_ply$MSDhoni)
####### Output > [1] 0.6289972

sd(ind_ply$ShaneWatson )/mean(ind_ply$ShaneWatson )
####### Output > [1] 1.002898
```

Box plots for each 10 players We can see median from the plot at mid line is median

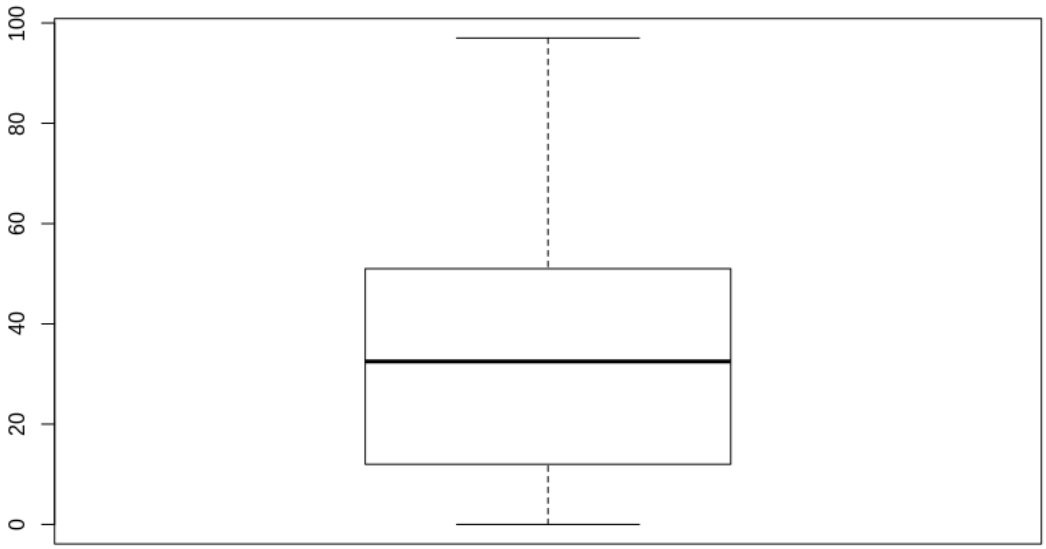
```
################ PLOT 1: Box plot for David Warner ###############
boxplot(ind_ply$DavidWarner)
num = as.numeric(ind_ply$DavidWarner)
outvalues = boxplot(num)$out
which(ind_ply$DavidWarner %in% outvalues)
```



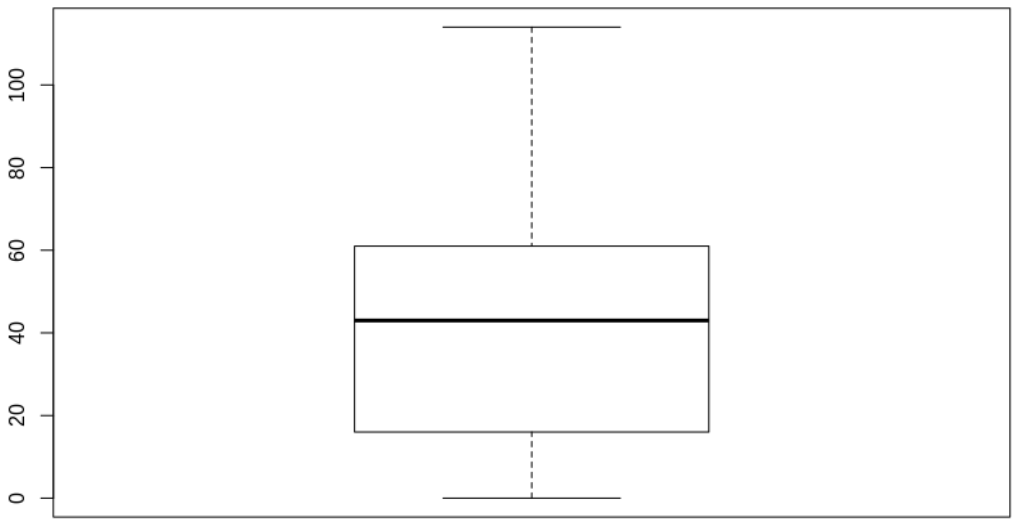

```
################ PLOT 2: Box plot for LokeshRahul ###############
boxplot(ind_ply$LokeshRahul)
num = as.numeric(ind_ply$LokeshRahul)
outvalues = boxplot(num)$out
which(ind_ply$LokeshRahul %in% outvalues)
```



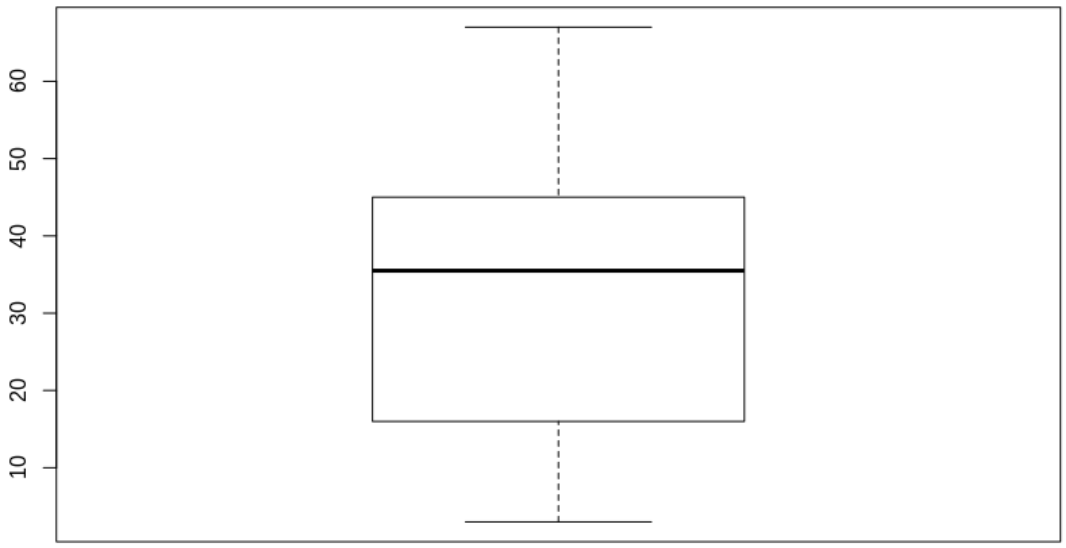
```
################ PLOT 3: Box plot for ShikharDhawan ###############
boxplot(ind_ply$ShikharDhawan)
num = as.numeric(ind_ply$ShikharDhawan)
outvalues = boxplot(num)$out
which(ind_ply$ShikharDhawan %in% outvalues)
```



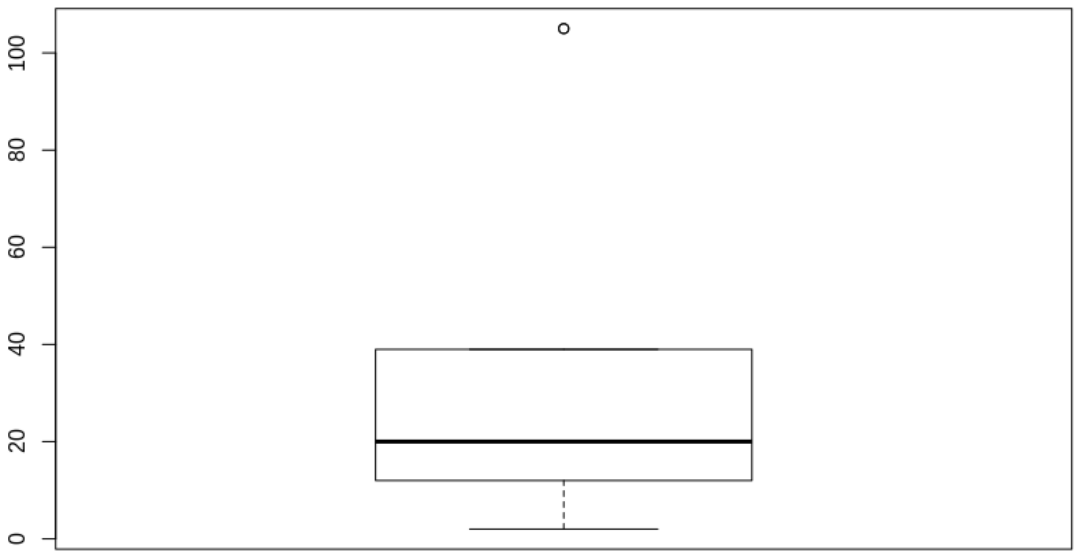

```
################ PLOT 4: Box plot for JonnyBairstow ###############
boxplot(ind_ply$JonnyBairstow)
num = as.numeric(ind_ply$JonnyBairstow)
outvalues = boxplot(num)$out
which(ind_ply$JonnyBairstow %in% outvalues)
```



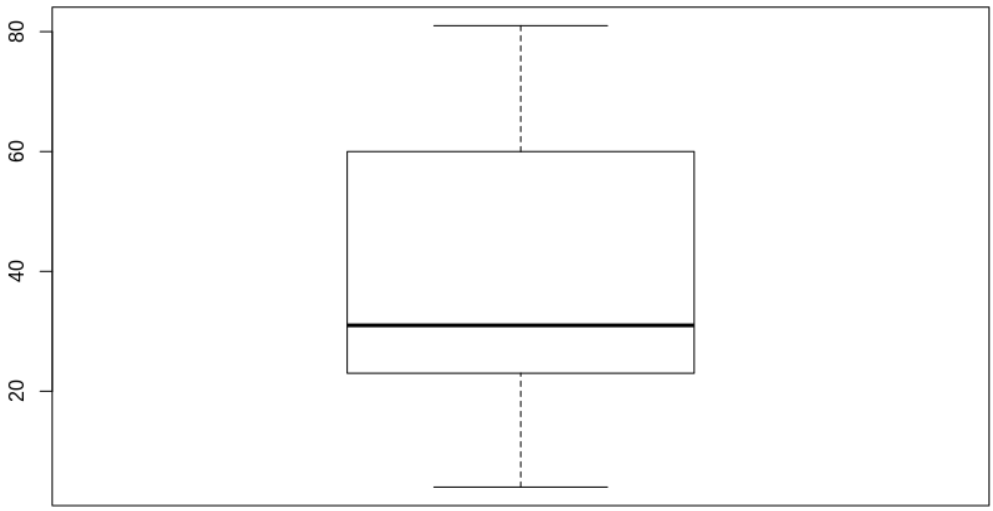
```
############### PLOT 5: Box plot for ShreyasIyer ###############
boxplot(ind_ply$ShreyasIyer)
num = as.numeric(ind_ply$ShreyasIyer)
outvalues = boxplot(num)$out
which(ind_ply$ShreyasIyer %in% outvalues)
```



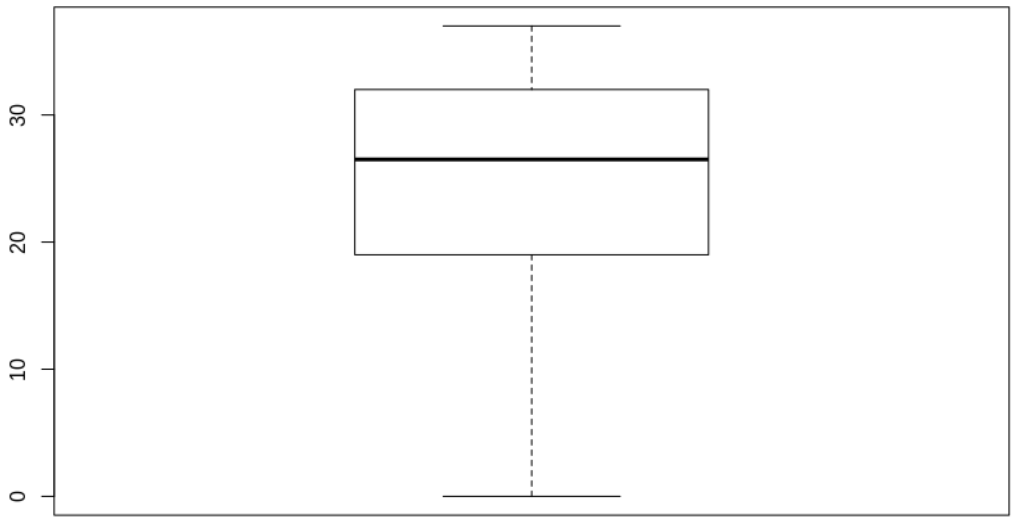

```
############### PLOT 6: Box plot for AjinkyaRahane ###############
boxplot(ind_ply$AjinkyaRahane)
num = as.numeric(ind_ply$AjinkyaRahane)
outvalues = boxplot(num)$out
which(ind_ply$AjinkyaRahane %in% outvalues)   (We can see an Outliar here i.e, at score 2)
```



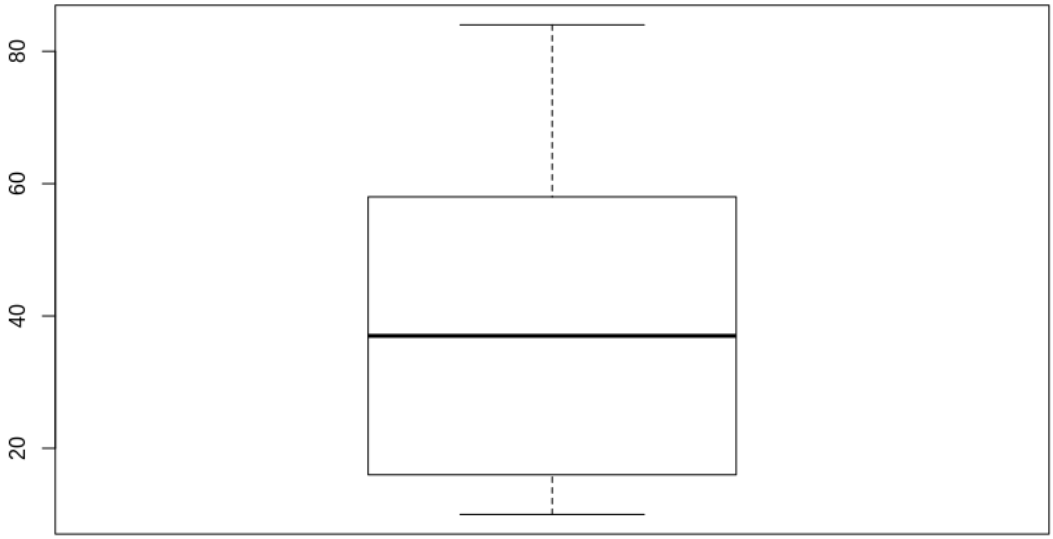
```
################ PLOT 7: Box plot for QuintondeKock ###############
boxplot(ind_ply$QuintondeKock)
num = as.numeric(ind_ply$QuintondeKock)
outvalues = boxplot(num)$out
which(ind_ply$QuintondeKock %in% outvalues)
```



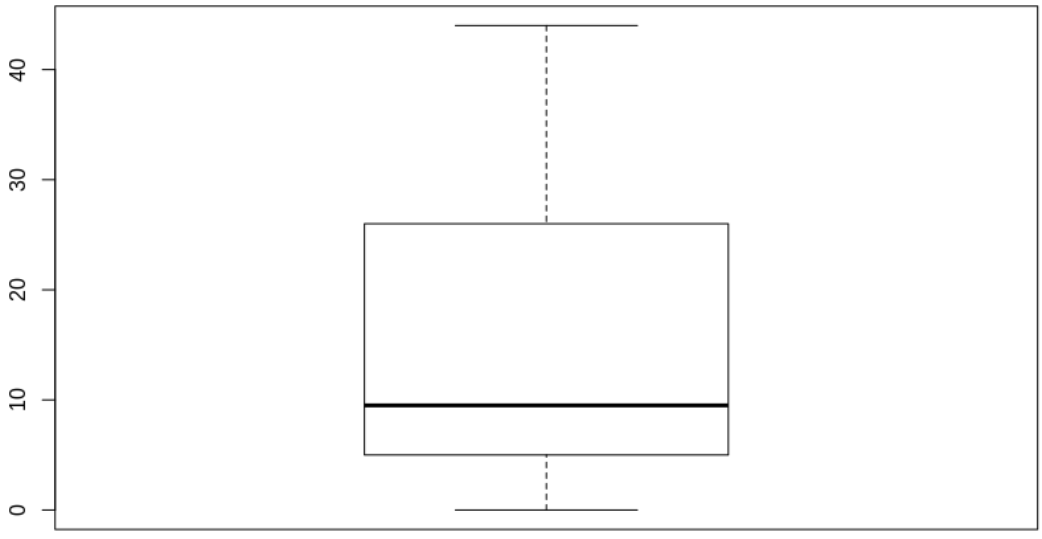

```
################ PLOT 8: Box plot for HardikPandya ###############
boxplot(ind_ply$HardikPandya)
num = as.numeric(ind_ply$HardikPandya)
outvalues = boxplot(num)$out
which(ind_ply$HardikPandya %in% outvalues)
```



```
################ PLOT 9: Box plot for MSDhoni ################
boxplot(ind_ply$MSDhoni)
num = as.numeric(ind_ply$MSDhoni)
outvalues = boxplot(num)$out
which(ind_ply$MSDhoni %in% outvalues)
```




```
################ PLOT 10: Box plot for ShaneWatson ################
boxplot(ind_ply$ShaneWatson)
num = as.numeric(ind_ply$ShaneWatson)
outvalues = boxplot(num)$out
which(ind_ply$ShaneWatson %in% outvalues)
```



Correlation between two few pairs of players

```
######## corelation between Davidwarner and Lokesh rahul ############

cor.test(ind_ply$DavidWarner, ind_ply$LokeshRahul,  method = "spearman")

####### Output >          Spearman's rank correlation rho

data:  ind_ply$DavidWarner and ind_ply$LokeshRahul
S = 150.96, p-value = 0.8152
alternative hypothesis: true rho is not equal to 0
sample estimates:
        rho
0.08510678


######## corelation between ShikharDhawan and JonnyBairstow ############

cor.test(ind_ply$ShikharDhawan, ind_ply$JonnyBairstow,  method = "spearman")

####### Output >      Spearman's rank correlation rho


data:  ind_ply$ShikharDhawan and ind_ply$JonnyBairstow
S = 188, p-value = 0.7072
alternative hypothesis: true rho is not equal to 0
sample estimates:
        rho
-0.1393939


######## corelation between ShreyasIyer and AjinkyaRahane ############

cor.test(ind_ply$ShreyasIyer, ind_ply$AjinkyaRahane,  method = "spearman")

####### Output >      Spearman's rank correlation rho


data:  ind_ply$ShreyasIyer and ind_ply$AjinkyaRahane
S = 248.52, p-value = 0.1355
alternative hypothesis: true rho is not equal to 0
sample estimates:
        rho
-0.5062114


######## corelation between QuintondeKock and HardikPandya ############

cor.test(ind_ply$QuintondeKock, ind_ply$HardikPandya,  method = "spearman")

####### Output >          Spearman's rank correlation rho

data:  ind_ply$QuintondeKock and ind_ply$HardikPandya
S = 120.87, p-value = 0.455
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
0.2674784

######## corelation between MSDhoni warner and ShaneWatson ############

cor.test(ind_ply$MSDhoni, ind_ply$ShaneWatson,  method = "spearman")

####### Output >         Spearman's rank correlation rho
data:  ind_ply$MSDhoni and ind_ply$ShaneWatson
S = 189.22, p-value = 0.6857
alternative hypothesis: true rho is not equal to 0
sample estimates:
        rho
-0.1467897
```