Cricket_classification

Biswas

7/18/2021

Uploading File (ball- by-ball dataset)

```
raw_data<-read.csv("IPL Ball-by-Ball 2008-2020.csv",header = TRUE)
dim(raw_data)
```

```
## [1] 193468     18
```

```
#converting na values to "0"
raw_data[is.na(raw_data)]<-0

# Dislaying first 3 rows of dataset
head(raw_data,3)
```

```
##       id inning over ball     batsman non_striker    bowler batsman_runs
## 1 335982      1    6    5  RT Ponting BB McCullum AA Noffke            1
## 2 335982      1    6    6 BB McCullum  RT Ponting AA Noffke            1
## 3 335982      1    7    1 BB McCullum  RT Ponting    Z Khan            0
##   extra_runs total_runs non_boundary is_wicket dismissal_kind player_dismissed
## 1          0          1            0         0              0                0
## 2          0          1            0         0              0                0
## 3          0          0            0         0              0                0
##   fielder extras_type          batting_team                bowling_team
## 1       0           0 Kolkata Knight Riders Royal Challengers Bangalore
## 2       0           0 Kolkata Knight Riders Royal Challengers Bangalore
## 3       0           0 Kolkata Knight Riders Royal Challengers Bangalore
```

```
#dimension of datast
cat("The dimesion of raw data-set :\n Row:",dim(raw_data)[1],"\n Column:",dim(raw_data)[2])
```

```
## The dimesion of raw data-set :
##  Row: 193468
##  Column: 18
```

```
# creating group with respect to  batsman
grouped_batsman<-raw_data %>% arrange(batsman)
head(grouped_batsman)
```

```
##       id inning over ball        batsman non_striker       bowler batsman_runs
## 1 548346      1   13    1 A Ashish Reddy   JP Duminy  RJ Peterson            0
## 2 548346      1   13    2 A Ashish Reddy   JP Duminy  RJ Peterson            0
## 3 548346      1   13    3 A Ashish Reddy   JP Duminy  RJ Peterson            0
## 4 548346      1   13    4 A Ashish Reddy   JP Duminy  RJ Peterson            1
## 5 548346      1   13    6 A Ashish Reddy   JP Duminy  RJ Peterson            6
## 6 548346      1   14    4 A Ashish Reddy   JP Duminy JEC Franklin            2
##   extra_runs total_runs non_boundary is_wicket dismissal_kind player_dismissed
## 1          0          0            0         0              0                0
## 2          0          0            0         0              0                0
## 3          0          0            0         0              0                0
## 4          0          1            0         0              0                0
## 5          0          6            0         0              0                0
## 6          0          2            0         0              0                0
##   fielder extras_type    batting_team   bowling_team
## 1       0           0 Deccan Chargers Mumbai Indians
## 2       0           0 Deccan Chargers Mumbai Indians
## 3       0           0 Deccan Chargers Mumbai Indians
## 4       0           0 Deccan Chargers Mumbai Indians
## 5       0           0 Deccan Chargers Mumbai Indians
## 6       0           0 Deccan Chargers Mumbai Indians
```

```r
# Creating batting dataframe with all the batting attributes (summary)
#Batsman and balls_faced attribute
bat_tab<-count(grouped_batsman, 'batsman')
names(bat_tab)[1]<-"Batsman"
names(bat_tab)[2]<-"Balls_faced"

# adding Innings to the dataframe
id<- aggregate(id ~ batsman, grouped_batsman, function(x) length(unique(x)))[2]
bat_tab["Innings_played"]<-id

#adding Runs_scored to the dataframe
Runs_scored<- aggregate(batsman_runs ~ batsman, grouped_batsman, function(x) sum(x))[2]
bat_tab["Runs_scored"]<- Runs_scored

#adding No_4s and No_6s to the dataframe

No_4s<- aggregate(batsman_runs==4 ~ batsman, grouped_batsman, function(x) sum(x))[2]
bat_tab["No_4s"]<- No_4s

No_6s<- aggregate(batsman_runs==6 ~ batsman, grouped_batsman, function(x) sum(x))[2]
bat_tab["No_6s"]<- No_6s

#adding Batting_avg to the dataframe i.e runs/innings
Batting_avg<-round(bat_tab['Runs_scored']/bat_tab['Innings_played'],1)
bat_tab["Batting_avg"]<-Batting_avg

#adding Strike_rate to the dataframe i.e (Runs_scored/Balls_faced) X 100
Strike_rate<-round((bat_tab['Runs_scored']/bat_tab['Balls_faced'])*100,1)
bat_tab["Strike_rate"]<-Strike_rate
```

```r
# creating group with respect to  bowler
grouped_bowler<-raw_data %>% arrange(bowler)
head(grouped_bowler)
```

```
##       id inning over ball   batsman non_striker         bowler batsman_runs
## 1 548329      2    5    1 DA Warner     NV Ojha A Ashish Reddy            0
## 2 548329      2    5    2 DA Warner     NV Ojha A Ashish Reddy            6
## 3 548329      2    5    3 DA Warner     NV Ojha A Ashish Reddy            6
## 4 548329      2    5    4 DA Warner     NV Ojha A Ashish Reddy            4
## 5 548329      2    5    5 DA Warner     NV Ojha A Ashish Reddy            0
## 6 548329      2   12    5   NV Ojha   DA Warner A Ashish Reddy            2
##   extra_runs total_runs non_boundary is_wicket dismissal_kind player_dismissed
## 1          0          0            0         0              0                0
## 2          0          6            0         0              0                0
## 3          0          6            0         0              0                0
## 4          0          4            0         0              0                0
## 5          0          0            0         0              0                0
## 6          0          2            0         0              0                0
##   fielder extras_type     batting_team    bowling_team
## 1       0           0 Delhi Daredevils Deccan Chargers
## 2       0           0 Delhi Daredevils Deccan Chargers
## 3       0           0 Delhi Daredevils Deccan Chargers
## 4       0           0 Delhi Daredevils Deccan Chargers
## 5       0           0 Delhi Daredevils Deccan Chargers
## 6       0           0 Delhi Daredevils Deccan Chargers
```

```r
# Exploring dismissal kind data
dismissal_list<-unique(grouped_bowler['dismissal_kind'])
print(dismissal_list)
```

```
##              dismissal_kind
## 1                         0
## 34                  bowled
## 36                  caught
## 94                     lbw
## 187       caught and bowled
## 239               run out
## 300               stumped
## 14575          hit wicket
## 27935         retired hurt
## 83013 obstructing the field
```

```r
# dismissal kind that will be accounted for wicket for bowlers are :
print("dismissal kind that will be accounted for wicket for bowlers are :
      caught,bowled,lbw,stumped,caught and bowled, hit wicket . The irrelevant
      for retired hurt, run out and obstructing the field shall be removed")
```

```
## [1] "dismissal kind that will be accounted for wicket for bowlers are : \n      caught,bowled,lbw,stu
```

```r
# Creating bowling dataframe with all the bowling attributes (summary)
#Bowler and Balls_bowled attribute
```

```r
bowl_tab<-count(grouped_bowler, 'bowler')
names(bowl_tab)[1]<-"Bowler"
names(bowl_tab)[2]<-"Balls_bowled"

# adding Wickets_taken to the dataframe
# removing irrelevant rows related to wicket count for bowlers

wkts_to_bowler_df<-grouped_bowler[!(grouped_bowler$dismissal_kind=="retired hurt" |
                                      grouped_bowler$dismissal_kind=="run out") |
                                    grouped_bowler$dismissal_kind=="obstructing the field" ,]


Wickets_taken<- aggregate(is_wicket ~ bowler, wkts_to_bowler_df, function(x) sum(x))[2]
bowl_tab["Wickets_taken"]<-Wickets_taken

# adding Overs_bowled to the dataframe
Overs_bowled<- aggregate(c(over + id) ~ bowler, grouped_bowler, function(x) length(unique(x)))[2]
bowl_tab["Overs_bowled"]<-Overs_bowled


# adding Runs_conceded to the dataframe

Runs_conceded<- aggregate(batsman_runs ~ bowler, grouped_bowler, function(x) sum(x))[2]
bowl_tab["Runs_conceded"]<-Runs_conceded
#adding runs because of wide and no_balls


# Filtering extras with respect to wides and no-balls
Extras<-filter(grouped_bowler,grouped_bowler$extras_type == "wides" |
                 grouped_bowler$extras_type =="noballs")[,c("bowler","extra_runs")]


df1<- aggregate(extra_runs ~ bowler, data=Extras, FUN=sum)
names(df1)[names(df1) == "bowler"] <- "Bowler"

bowl_tab<-merge(bowl_tab, df1, by = "Bowler",all=TRUE)
bowl_tab[is.na(bowl_tab)]<-0


# adding Total_runs_conceded = Runs_conceded + Extras to the dataframe
bowl_tab$Total_runs_conceded<- bowl_tab$Runs_conceded +bowl_tab$extra_runs

# Dropping Runs_conceded and extra_runs from dataframe asthese are  unnecessary columns, taken care by

bowl_tab = subset(bowl_tab, select = -c(Runs_conceded,extra_runs))

# adding Bowling_economy to the dataframe
bowl_tab["Bowling_economy"] <- round(bowl_tab$Total_runs_conceded/
                                       bowl_tab$Overs_bowled,1)
```

```r
# Preparing comprehensive list to include all players from the raw database
# including batsman and bowler list along with General attributes
```

```
raw_match_data<-read.csv("IPL Matches 2008-2020.csv",header = TRUE)
dim(raw_match_data)
```

```
## [1] 816  17
```

```
fielders<-table(unlist(strsplit(raw_data$fielder, ',')))

fielders_list<- as.data.frame(fielders)[1]


substitute_fielder <- fielders_list[str_detect(fielders_list$Var1, "(sub)"), ]

#removing substitute fielders from fielders list
filtered_fielders<-fielders_list[!(fielders_list$Var1 %in% substitute_fielder),]
filtered_fielders<-as.data.frame(filtered_fielders)[-1,]
filtered_fielders<-as.data.frame(filtered_fielders)

Players_list<-unique(c(raw_data$batsman,raw_data$bowler,raw_data$non_striker,
                       filtered_fielders$filtered_fielders))
Players_list<-as.data.frame(Players_list)


Players<- Players_list[!grepl("^\\d+$", Players_list$Players_list), ]

# List of complete players for the final list
Players<-as.data.frame(Players)
Players<-Players %>% filter(is.na(as.numeric(Players)))
```

```
## Warning in mask$eval_all_filter(dots, env_filter): NAs introduced by coercion
```

```
Bowler_id<-grouped_bowler%>% distinct(id, bowler, .keep_all = FALSE)

Batsman_id<-grouped_batsman %>% distinct(id, batsman, .keep_all = FALSE)

Non_striker_id<-raw_data %>% distinct(id, non_striker, .keep_all = FALSE)

Fielder_id<-raw_data %>% distinct(id, fielder, .keep_all = FALSE)

#Removing zero values from dataframe  of fielder
Fielder_id<-Fielder_id[Fielder_id$fielder != 0, ]


# Creating new row for multiple fielders clubed together
Fielder_id_sep<-Fielder_id %>%
    mutate( fielder = strsplit(as.character(fielder), ",")) %>%
    unnest(fielder)

# Removing all substitute players from the dataframe (sub)
Fielder_id_final<-Fielder_id_sep[!grepl('(sub)',Fielder_id_sep$fielder),]

# Merging dataset by all ids together
merge_players<-merge(Bowler_id, Batsman_id, by = "id",all=TRUE)
```

```r
nonstriker_fielder<-merge(Non_striker_id,Fielder_id_final, by = "id",all=TRUE)

merged_list<-merge(nonstriker_fielder,merge_players, by = "id",all=TRUE)

df1 <- merged_list %>%
  group_by(id) %>%
  summarise(value = list(unique(c(non_striker,bowler,fielder,batsman)))) %>%
  unnest(value)



setDT(merged_list)
df2 <- melt(merged_list, id.vars = 'id')[, .(value = list(unique(value))), id]

#unique list of players for unique ids
players_list_per_match<- df2$value

players_df_per_match <- data.frame(matrix(unlist(players_list_per_match),
                                          nrow=length(players_list_per_match), byrow=TRUE))
```

```
## Warning in matrix(unlist(players_list_per_match), nrow =
## length(players_list_per_match), : data length [17495] is not a sub-multiple or
## multiple of the number of rows [816]
```

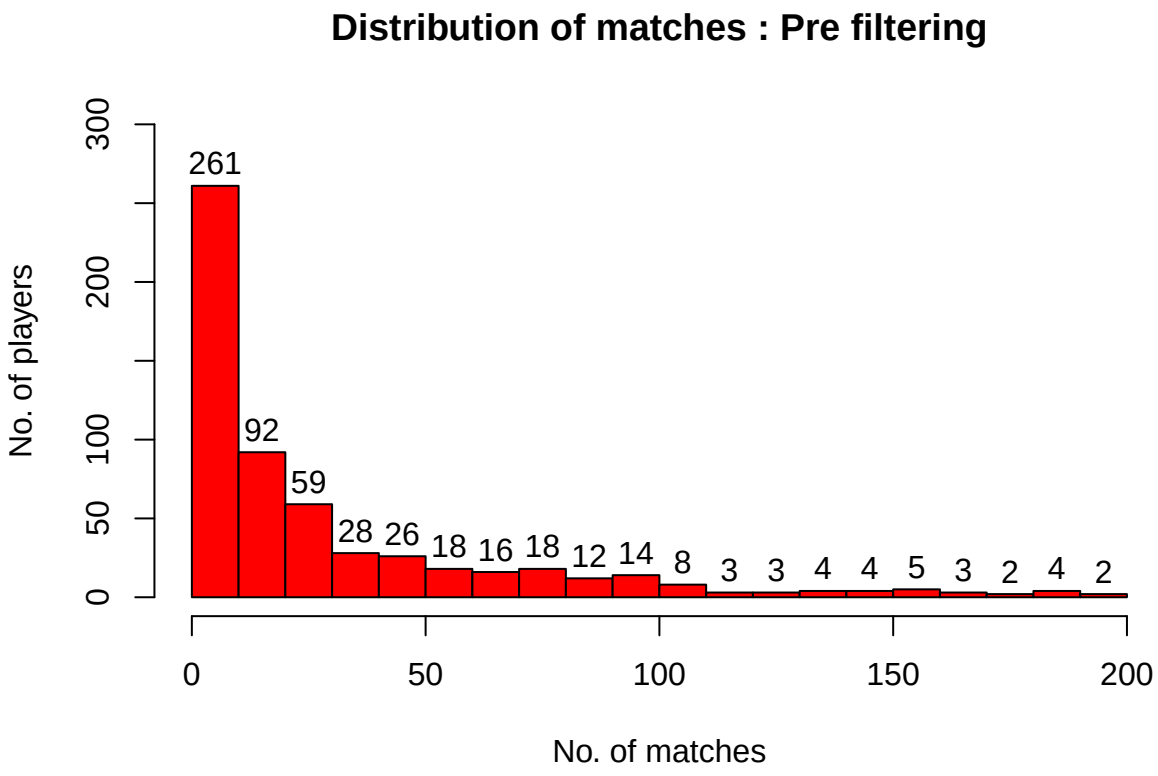
```r
players_match<-table(unlist(lapply(df2$value, unique)))
players_match_df<-as.data.frame(players_match)
colnames(players_match_df)<- c("Player","No_matches")
```

```r
#combining overall players statistics
bat_tab["Player"]<-bat_tab$Batsman
bowl_tab["Player"]<-bowl_tab$Bowler


#combining batsman and bowlers attributes


bat_bowl<-merge(bat_tab,bowl_tab, by = "Player",all=TRUE )
bat_bowl[is.na(bat_bowl)]<-0

bat_bowl_matches<-merge(bat_bowl,players_match_df, by = "Player",all=TRUE)
#Removing Basman and Bowler columns from data frame and keeping onl player
Players_stats<-bat_bowl_matches[,-c(2,10)]


#combining Catches_taken attribute in Caught and Bowled and catches
# i. Caught and bowed as bowler

C_B<-raw_data[(raw_data$dismissal_kind %in% 'caught and bowled'),]

df_catches_as_bowlers<-aggregate(dismissal_kind ~ bowler, C_B,
                                 function(x) sum(length(x)))
colnames(df_catches_as_bowlers)<- c("Player","Catches")
```

```r
# i. Caught as fielder
C<-raw_data[(raw_data$dismissal_kind %in% 'caught'),]

df_catches_as_fielder<-aggregate(dismissal_kind ~ fielder, C,
                                 function(x) sum(length(x)))

colnames(df_catches_as_fielder)<- c("Player","Catches")
#combining both catches by player name
Catches_df<-merge(df_catches_as_bowlers,df_catches_as_fielder,
                  by="Player",all=TRUE)
Catches_df[is.na(Catches_df)]<-0
Catches_df["Catches"]<-Catches_df$Catches.x + Catches_df$Catches.y

Catches_taken = subset(Catches_df, select = -c(Catches.x,Catches.y))
Players_statistics<-merge(Players_stats,Catches_taken,by="Player",all=TRUE)
Players_statistics[is.na(Players_statistics)]<-0

dim(Players_statistics)
```

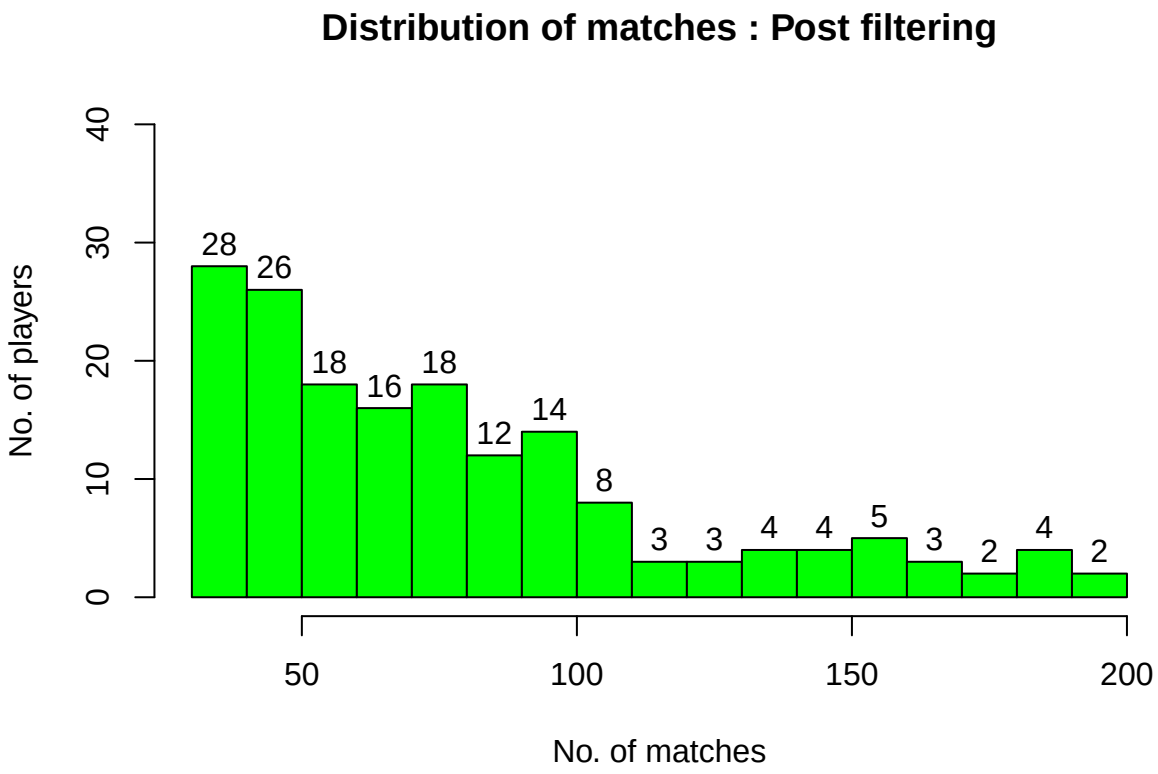
```
## [1] 649  15
```

```r
# Removing all substitute players from the dataframe (sub)
# Stats for final list of 582 players
Players_statistics_final<-Players_statistics[!grepl("(sub)",
                                                    Players_statistics$Player),]

# check players vs matches frequency
hist_matches<-hist(Players_statistics_final$No_matches,
                   main = "Distribution of matches : Pre filtering",
                   xlab = "No. of matches", ylab = "No. of players",
                   col = "red",ylim=c(0,300),breaks=20)
text(hist_matches$mids,hist_matches$counts,labels=hist_matches$counts,
     adj=c(0.5, -0.5))
```




```
# My criteria : Filtering out players who has played less than 30 matches to avoid bias result
## Normally a team has to play minimimum 14 matches in a season, minimum 2 season considering
```

```
# selecting players who has played more than 30 matches for reliable comparision with Gayle stats

Players_statistics_final <- subset(Players_statistics_final,
                                   No_matches>30)
#Data Scaling (numeric values)
# updating players name as column
rownames(Players_statistics_final)<-Players_statistics_final$Player
# No of players played more than 90 matches

hist_matches<-hist(Players_statistics_final$No_matches,
                   main = "Distribution of matches : Post filtering",
                   xlab = "No. of matches", ylab = "No. of players",
                   col = "green",ylim=c(0,40),breaks=20)
text(hist_matches$mids,hist_matches$counts,labels=hist_matches$counts,
     adj=c(0.5, -0.5))
```

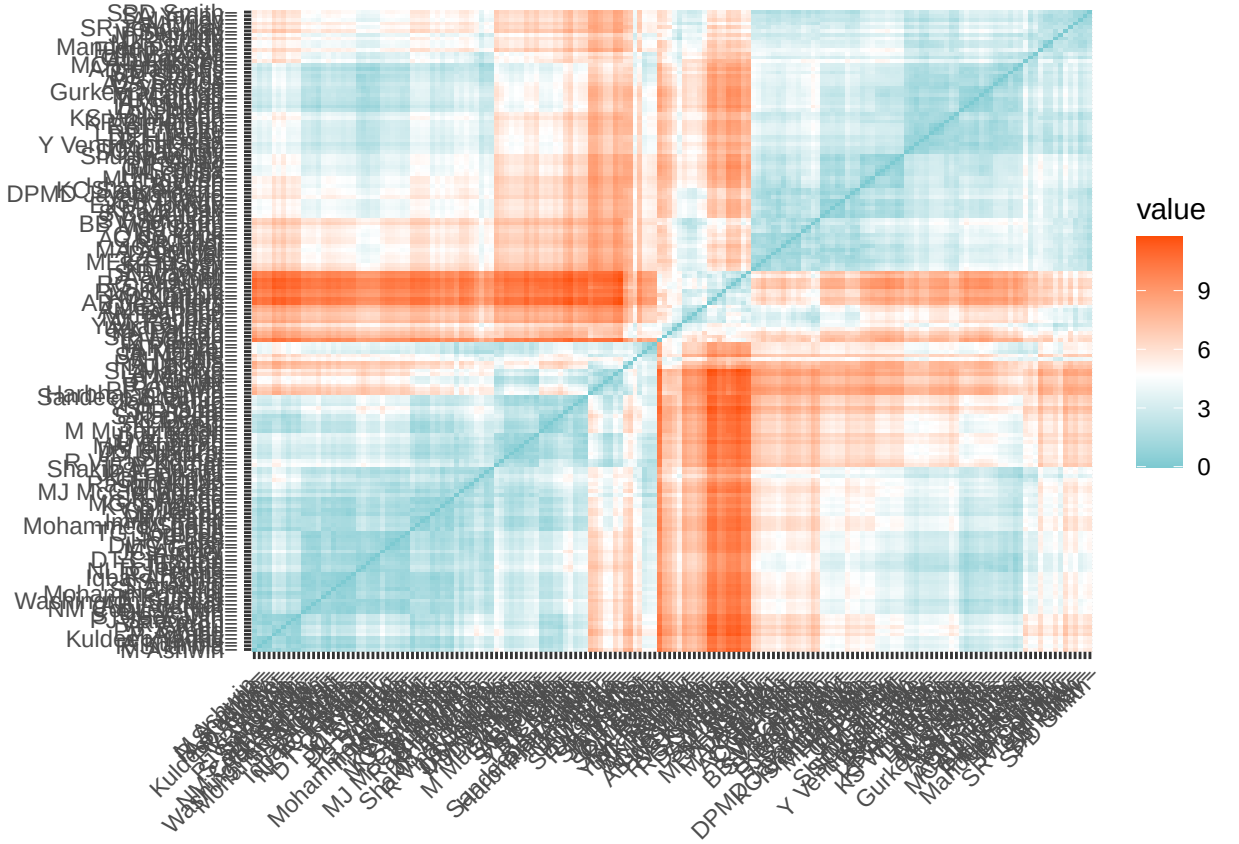



```
scaled_stats<-scale(Players_statistics_final[,2:15])
summary(scaled_stats) # checking mean=0 etc for proper scaling
```

```
##   Balls_faced      Innings_played     Runs_scored          No_4s
##  Min.   :-0.9103   Min.   :-1.1442   Min.   :-0.8902   Min.   :-0.8500
##  1st Qu.:-0.8164   1st Qu.:-0.8128   1st Qu.:-0.8041   1st Qu.:-0.7864
##  Median :-0.3187   Median :-0.2604   Median :-0.2810   Median :-0.3412
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.4450   3rd Qu.: 0.4631   3rd Qu.: 0.4244   3rd Qu.: 0.4257
##  Max.   : 3.5084   Max.   : 3.0314   Max.   : 3.4594   Max.   : 3.7054
##      No_6s          Batting_avg        Strike_rate       Balls_bowled
##  Min.   :-0.7975   Min.   :-1.50220   Min.   :-3.0695   Min.   :-0.9646
##  1st Qu.:-0.7270   1st Qu.:-0.98288   1st Qu.:-0.4294   1st Qu.:-0.9577
##  Median :-0.3123   Median : 0.01081   Median : 0.2727   Median :-0.1386
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3053   3rd Qu.: 0.86968   3rd Qu.: 0.6712   3rd Qu.: 0.5620
##  Max.   : 5.3605   Max.   : 2.11304   Max.   : 2.3255   Max.   : 2.9728
##  Wickets_taken      Overs_bowled     Total_runs_conceded Bowling_economy
##  Min.   :-0.9169   Min.   :-0.9829   Min.   :-0.9870     Min.   :-1.7366
##  1st Qu.:-0.9169   1st Qu.:-0.9744   1st Qu.:-0.9786     1st Qu.:-0.1905
##  Median :-0.2377   Median :-0.1046   Median :-0.1235     Median : 0.4097
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000     Mean   : 0.0000
##  3rd Qu.: 0.5627   3rd Qu.: 0.5766   3rd Qu.: 0.5857     3rd Qu.: 0.6098
##  Max.   : 3.2066   Max.   : 2.9672   Max.   : 2.9627     Max.   : 2.1437
##    No_matches         Catches
```

```
##  Min.   :-1.1582   Min.   :-1.2025
##  1st Qu.:-0.7971   1st Qu.:-0.6501
##  Median :-0.2193   Median :-0.3457
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.4065   3rd Qu.: 0.2857
##  Max.   : 2.8379   Max.   : 4.0739
```

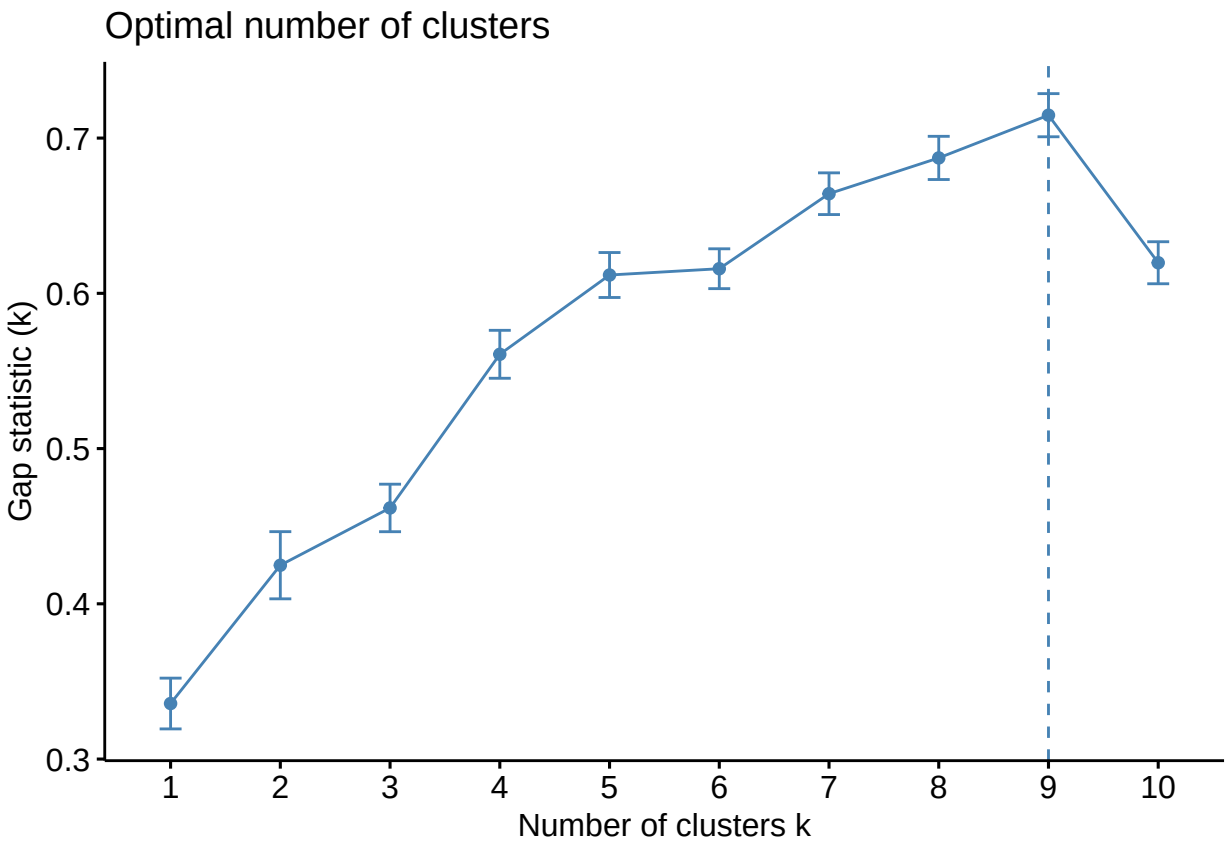
```
# clustering

## finding optimum nmbers of clusters using "factoextra" library and packages



dist_object<-get_dist(scaled_stats)

# Visualizing distance matrix
viz_dist<-fviz_dist(dist_object, gradient = list(low = "#00AFBB",
                                                 mid = "white", high = "#FC4E07"))

viz_dist
```



Cluster using Gap_stats

```
# Determining optimum number of cluster , method = "gap_stat"
optimal_cluster_gap_stat<-fviz_nbclust(scaled_stats, kmeans,
```

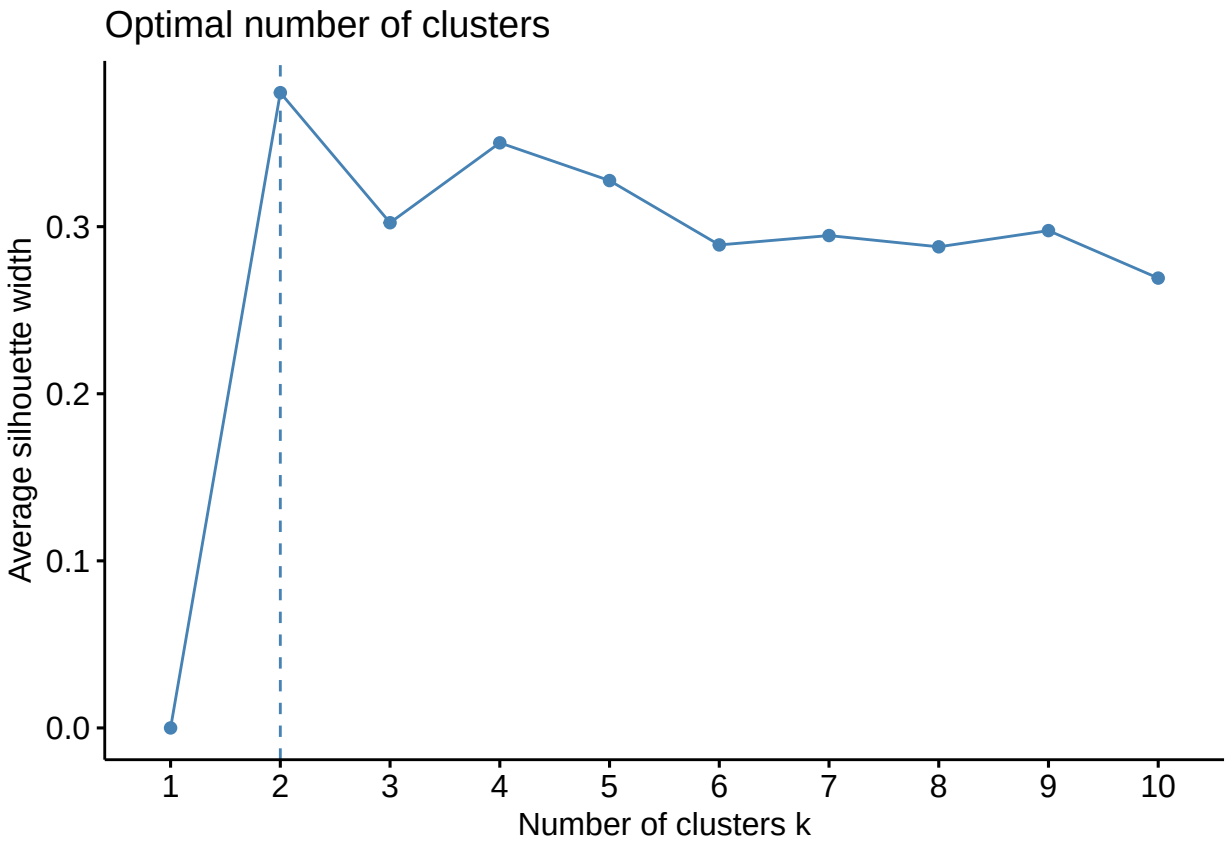
```
                                        method = "gap_stat")
optimal_cluster_gap_stat
```




```
cat("The Optimal number of cluser  with gap_stats method : 9 \n")
```

```
## The Optimal number of cluser  with gap_stats method : 9
```

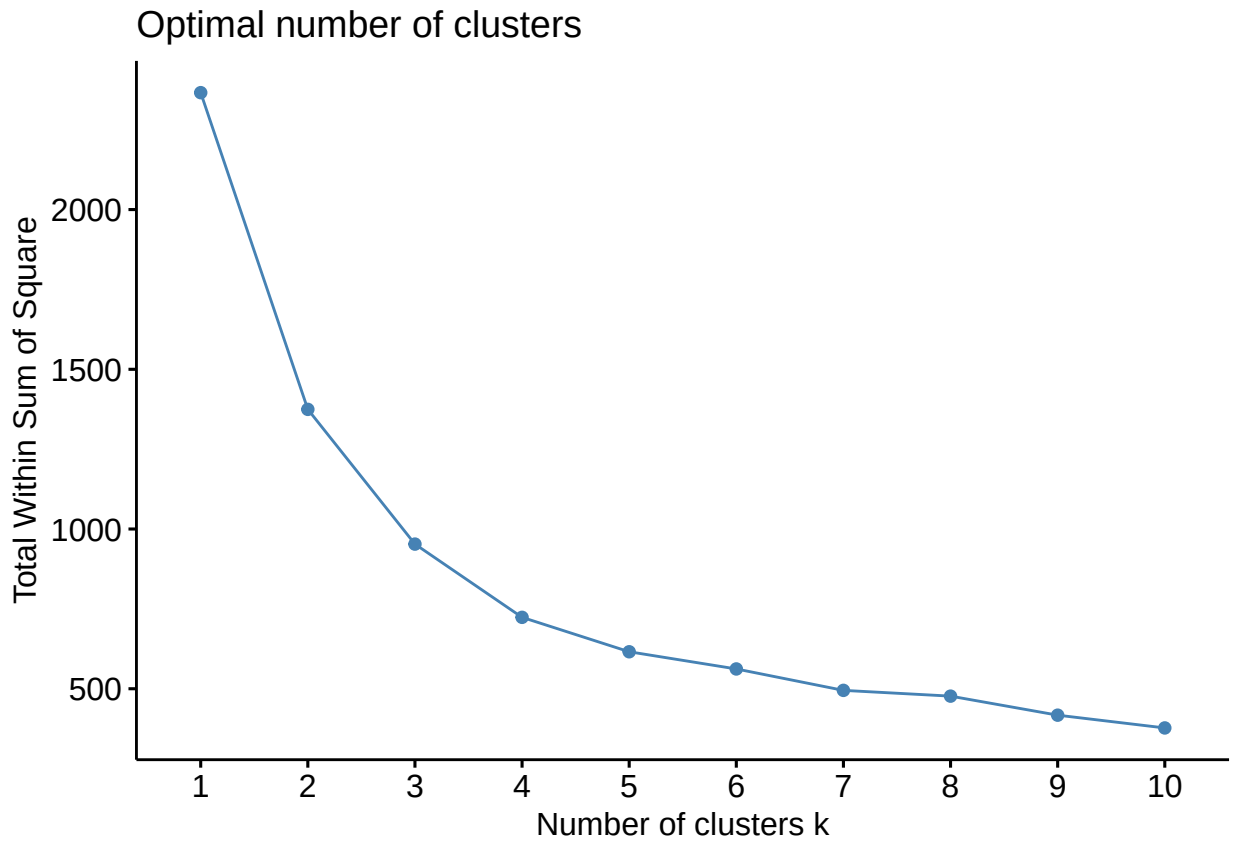
```
# Determining optimum number of cluster , method = "silhouette" (for average silhouette width)
optimal_cluster_silhouette<-fviz_nbclust(scaled_stats, kmeans,
                                         method = "silhouette")
optimal_cluster_silhouette
```




```
cat("The Optimal number of cluser  with silhouette method : 2 \n")
```

```
## The Optimal number of cluser  with silhouette method : 2
```

```
# Determining optimum number of cluster (elbow method), method = "wss" (for total within sum of square)

optimal_cluster_wss<-fviz_nbclust(scaled_stats, kmeans, method = "wss")
optimal_cluster_wss
```




```
cat("The Optimal number of cluser  with wss method : 4 \n")
```

```
## The Optimal number of cluser  with wss method : 4
```

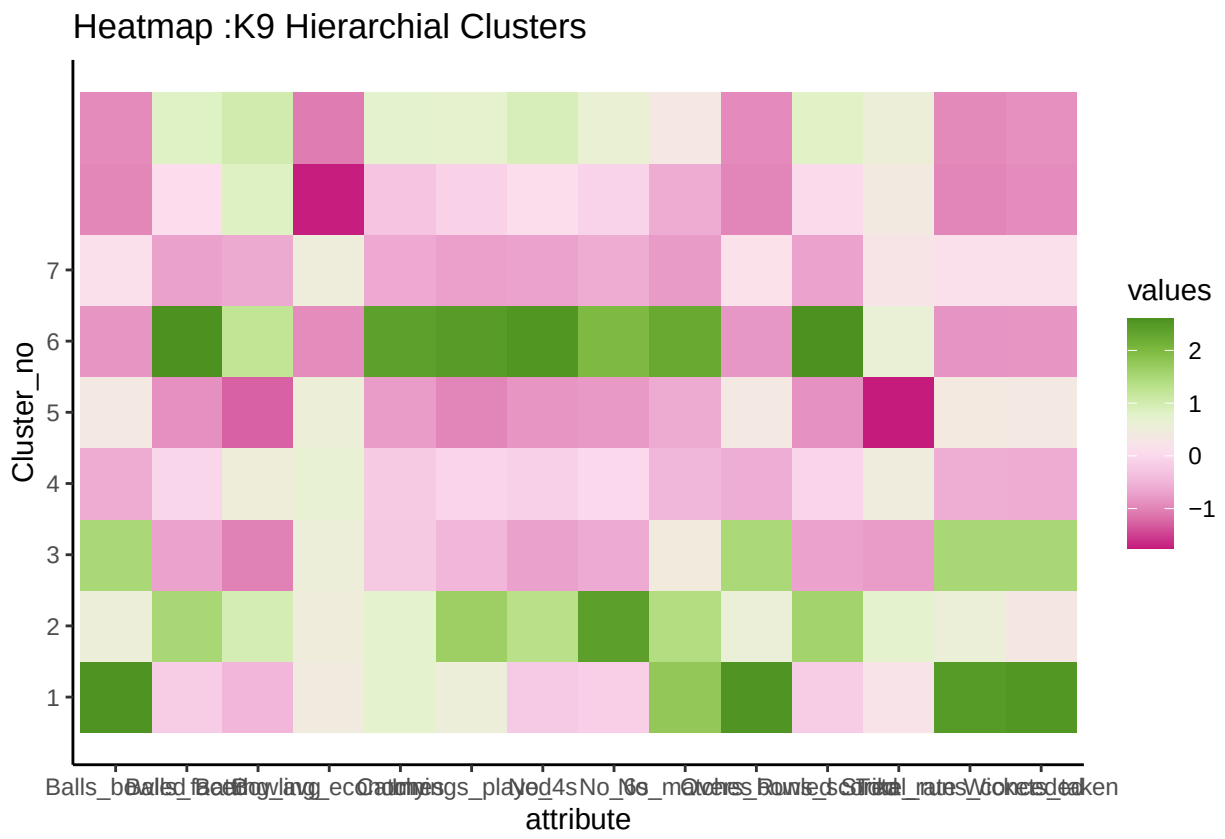
```
# K means for 9 clusters
set.seed(123)
k_9 <- kmeans(scaled_stats, 9, nstart = 25)

cluster_players_no<-table(k_9$cluster)
# visualizing kmeans clusters
Players_clusters<-fviz_cluster(k_9, data = scaled_stats,
             ggtheme = theme_minimal(),
             main = "Kmeans Clustering of Players"
             )



scale_value <- 1
ggsave(eval(Players_clusters), width = 20 * scale_value,
       height = 20 * scale_value, file = "Players_clusters.pdf")

tiff("Kmeans_Clusters.tiff", units="in", width=5, height=5, res=300)
eval(Players_clusters)
dev.off()
```

```
## pdf
##   2
```

```
#heatmap
k_9_centers<-k_9$centers

write.csv(k_9_centers, file ='k_9_centers.csv')


Cluster_no <- c(1: 9)
#building dataframe with centers and clusters
k_9_center_df <- data.frame(Cluster_no, k_9_centers)

library(comprehenr)
Players<- to_vec(for(i in 1:9) table(k_9$cluster)[[i]])

K_9_cluster_df <- data.frame(Cluster_no, Players)

#frequency of players in each K-9 cluster
K_9_cluster_df
```

```
##   Cluster_no Players
## 1          1       7
## 2          2       6
## 3          3      20
## 4          4      29
## 5          5      23
## 6          6      12
## 7          7      33
## 8          8      18
## 9          9      22
```

```
#reshape
center_reshape <- gather(k_9_center_df, attribute, values, Balls_faced: Catches)

library(RColorBrewer)
# Create the palette
my_palette <-colorRampPalette(brewer.pal(8, "PiYG"))(25)

# Plot the heat map
ggplot(data = center_reshape, aes(x = attribute, y = Cluster_no, fill = values)) +
    ggtitle("Heatmap :K9 Hierarchial Clusters")+
    scale_y_continuous(breaks = seq(1, 7, by = 1)) +
    geom_tile() +
    coord_equal() +
    scale_fill_gradientn(colours = my_palette) +
    theme_classic()
```




```
#defining function
Players_clust<-function(data_input){

df_cluster<-as.data.frame(count(data_input))
colnames(df_cluster)<-c("Cluster","No. of Players")

names(data_input)

N<-length(data_input)

df_k9<-NULL
for (i in 1:N){
df_k9= rbind(df_k9,data.frame(Players= data_input[i],
                              Cluster=data_input[[i]]))
}

df_k9$Player<-rownames(df_k9)
Players_K9<-df_k9 %>% arrange(Cluster)


rownames(Players_K9)<-NULL
K9_Players<-Players_K9[,2:3]

C1<-Players_K9 %>% filter(Cluster=="1")
C_1<-C1[,3]
```

```
C2<-Players_K9 %>% filter(Cluster=="2")
C_2<-C2[,3]

C3<-Players_K9 %>% filter(Cluster=="3")
C_3<-C3[,3]

C4<-Players_K9 %>% filter(Cluster=="4")
C_4<-C4[,3]

C5<-Players_K9 %>% filter(Cluster=="5")
C_5<-C5[,3]

C6<-Players_K9 %>% filter(Cluster=="6")
C_6<-C6[,3]

C7<-Players_K9 %>% filter(Cluster=="7")
C_7<-C7[,3]

C8<-Players_K9 %>% filter(Cluster=="8")
C_8<-C8[,3]

C9<-Players_K9 %>% filter(Cluster=="9")
C_9<-C9[,3]


max_length <- max(c(length(C_1), length(C_2), length(C_3),
                    length(C_4), length(C_5), length(C_6), length(C_7),
                    length(C_8), length(C_9)))
DT_cluster = data.table( "Cluster 1" =c(C_1,  rep(" ", max_length - length(C_1))),
                 "Cluster 2" = c(C_2,  rep(" ", max_length - length(C_2))),
                 "Cluster 3" = c(C_3,  rep(" ", max_length - length(C_3))),
                 "Cluster 4" = c(C_4,  rep(" ", max_length - length(C_4))),
                 "Cluster 5" = c(C_5,  rep(" ", max_length - length(C_5))),
                 "Cluster 6" = c(C_6,  rep(" ", max_length - length(C_6))),
                 "Cluster 7" = c(C_7,  rep(" ", max_length - length(C_7))),
                 "Cluster 8" = c(C_8,  rep(" ", max_length - length(C_8))),
                 "Cluster 9" = c(C_9,  rep(" ", max_length - length(C_9))))

cat("Players in each cluster : \n" )
return(DT_cluster) }

# Cluster for K_9
K9_Cluster<-Players_clust(k_9$cluster)
```

```
## Players in each cluster :
```

```
write.csv(K9_Cluster, file ='K_means_cluster.csv')

C<-K9_Cluster %>% filter_all(any_vars(. %in% "CH Gayle"))
C # cluster 2
```

```
##    Cluster 1 Cluster 2 Cluster 3 Cluster 4 Cluster 5      Cluster 6  Cluster 7
## 1:  A Mishra  CH Gayle   A Nehra A Symonds  A Kumble AB de Villiers AB Agarkar
```

```
##    Cluster 8    Cluster 9
## 1:   CA Lynn AC Gilchrist
```

```
Gayle_cluster<-K9_Cluster$`Cluster 2`
# Players categorized alongwith Gayle
as.data.frame(Gayle_cluster)
```

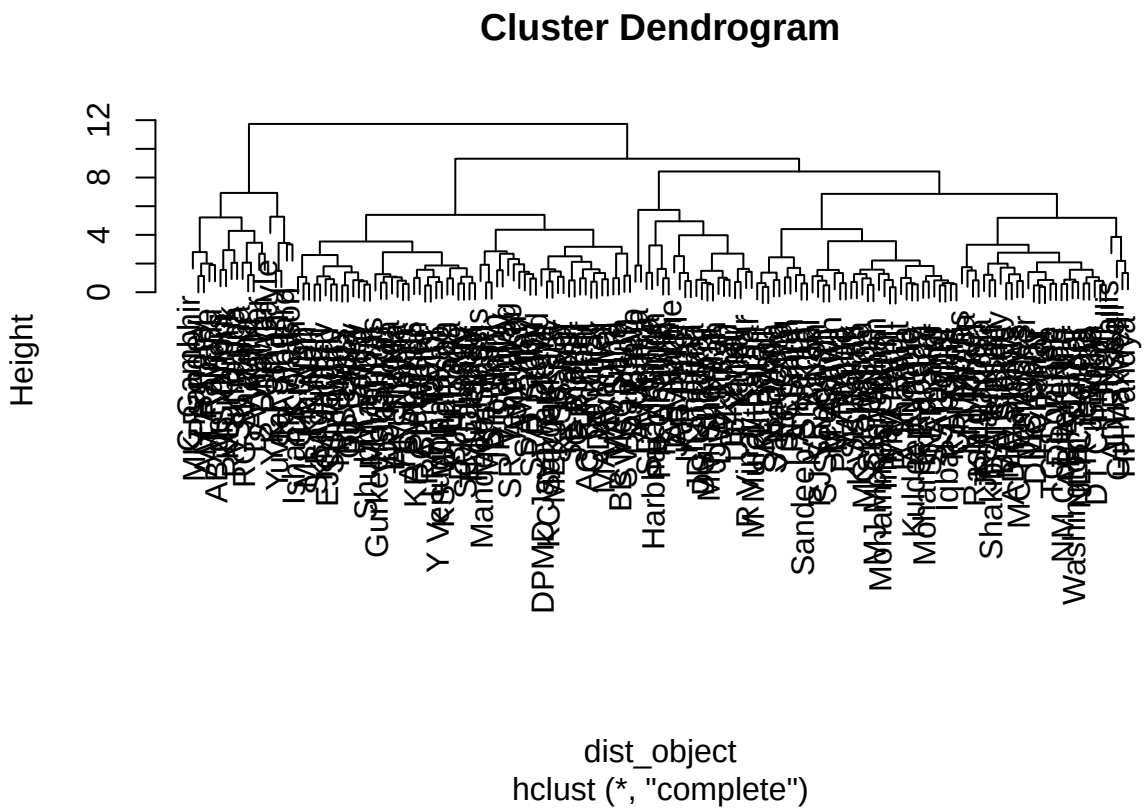
```
##    Gayle_cluster
## 1       CH Gayle
## 2      JH Kallis
## 3     KA Pollard
## 4      SR Watson
## 5      YK Pathan
## 6   Yuvraj Singh
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33
```

```
names_list<-Gayle_cluster[1:6]

cat("\nPlayers categorized alongwith CH Gayle are :\n",names_list,sep = "\n")
```

```
##
## Players categorized alongwith CH Gayle are :
##
## CH Gayle
## JH Kallis
## KA Pollard
## SR Watson
```

```
## YK Pathan
## Yuvraj Singh
```

```
## HIERARCHIAL CLUSTERING (agglomerative) , Method 1
dist_object<-dist(scaled_stats)
clusters <- hclust(dist_object)
plot(clusters)
```



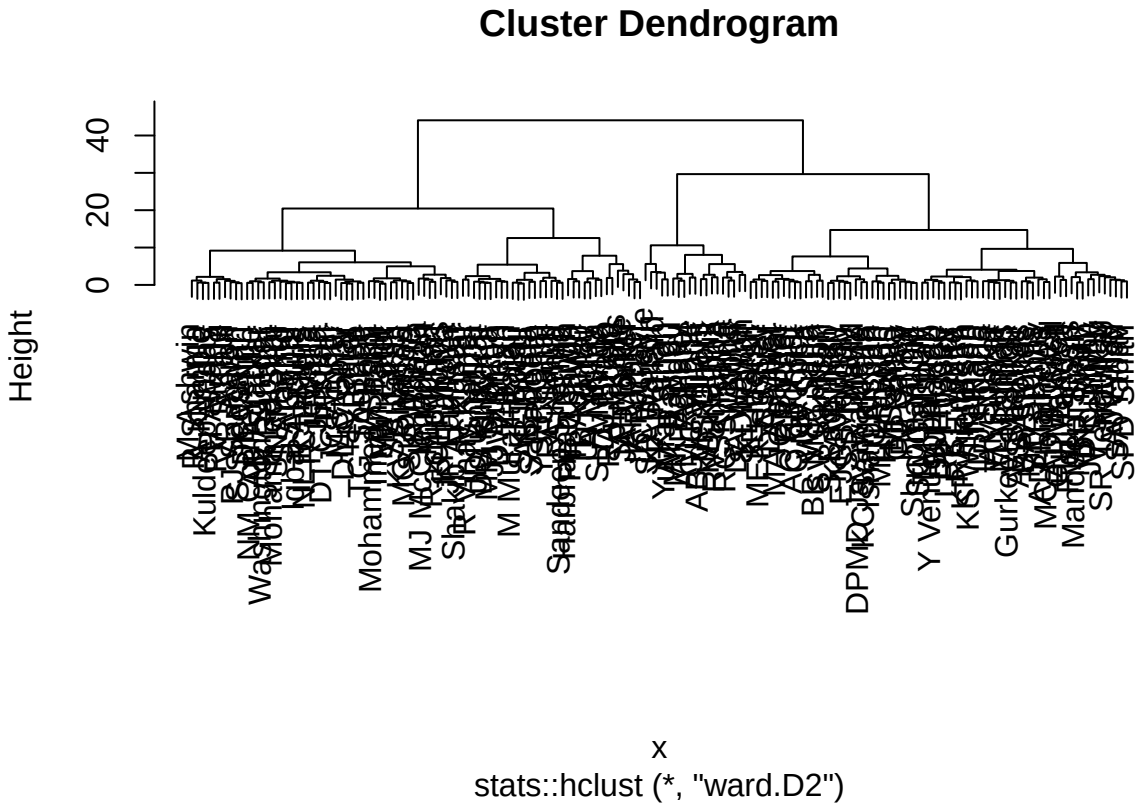

```
#visualuize using Dendogram
Players_cluster<-fviz_dend(clusters , rect = TRUE, cex = 0.5,show_labels = F,)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
# looking at the tree , the ideal clusters look like 9
clusterCut <- cutree(clusters, 9)

table(clusterCut)
```

```
## clusterCut
##  1  2  3  4  5  6  7  8  9
## 37 20 33 31 14 28  1  2  4
```

```
## HIERARCHIAL CLUSTERING (agglomerative)
# Hierarchial Clustering (9 clusters) ,method = hc_method
h_cluster <- hcut(scaled_stats, k = 9, stand = TRUE)
plot(h_cluster)
```




```
#visualuize using Dendogram

Players_h_cluster<-fviz_dend(h_cluster , rect = TRUE,
                             cex = 0.5,show_labels = F,)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
scale_value <- 1
ggsave(eval(Players_h_cluster), width = 20 * scale_value,
       height = 20 * scale_value, file = "Players_H_clusters.pdf")

tiff("Players_h_cluster.tiff", units="in", width=5, height=5, res=300)
eval(Players_h_cluster)
dev.off()
```

```
## pdf
##   2
```

```
Players_h_cluster_1<-fviz_dend(h_cluster , rect = TRUE,
                               cex = 1,show_labels = T,
    type = "phylogenic",)

scale_value <- 1
ggsave(eval(Players_h_cluster_1), width = 20 * scale_value,
       height = 20 * scale_value, file = "Players_H_clusters_1.pdf")

cluster_Cut <- cutree(h_cluster, 9)



#Calling the function for cluster_Cut

Hierarchial_Clust<-Players_clust(cluster_Cut )
```

```
## Players in each cluster :
```

```
write.csv(Hierarchial_Clust, file ='Hierarchial_Clust.csv')


HC<-Hierarchial_Clust %>% filter_all(any_vars(. %in% "CH Gayle"))

# visualizing Hierarchial clusters
Players_clusters_h<-fviz_cluster(h_cluster, data = scaled_stats,
             ggtheme = theme_minimal(),
             main = "Hierarchical Clustering of Players"
             )
scale_value <- 1
ggsave(eval(Players_clusters_h), width = 20 * scale_value,
       height = 20 * scale_value, file = "H_Players_clusters.pdf")

tiff("H_Clusters.tiff", units="in", width=5, height=5, res=300)
eval(Players_clusters_h)
dev.off()
```

```
## pdf
##   2
```

```
Gayle_cluster_HC<-Hierarchial_Clust$`Cluster 9`
# Players categorized alongwith Gayle
as.data.frame(Gayle_cluster_HC)
```

```
##    Gayle_cluster_HC
## 1          CH Gayle
## 2        KA Pollard
## 3         SR Watson
## 4         YK Pathan
## 5      Yuvraj Singh
## 6
## 7
## 8
```

```
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
## 37
## 38
## 39
```

```
names_list_HC<-Gayle_cluster_HC[1:5]

cat("\nPlayers categorized alongwith CH Gayle are :\n",names_list_HC,sep = "\n")
```

```
##
## Players categorized alongwith CH Gayle are :
##
## CH Gayle
## KA Pollard
## SR Watson
## YK Pathan
## Yuvraj Singh
```

Predicting performance of shortlisted players (ML models)

```
# the shortlisted players are :

shortlisted_players<-c("CH Gayle","KA Pollard","SR Watson","Yuvraj Singh","YK Pathan")


Player_stats_model<- function(player){
```

```
Player_stats_batsman<-grouped_batsman %>% filter(batsman==player)


Player_runs_per_innings<- aggregate(batsman_runs ~ id,
                                    data=Player_stats_batsman, FUN=sum)
colnames(Player_runs_per_innings)<-c("id","Runs_scored")
Player_runs_per_innings["batting_team"]<- aggregate(batting_team~ id,
                                                    Player_stats_batsman, function(x) unique(x))[2]

Player_runs_per_innings["bowling_team"]<- aggregate(bowling_team~ id,
                                                    Player_stats_batsman, function(x) unique(x))[2]

Player_runs_per_innings["No_4s"]<- aggregate(batsman_runs==4 ~ id,
                                             Player_stats_batsman, function(x) sum(x))[2]

Player_runs_per_innings["No_6s"]<- aggregate(batsman_runs==6 ~ id,
                                             Player_stats_batsman, function(x) sum(x))[2]

Player_runs_per_innings["Balls_faced"]<-count(Player_stats_batsman, 'id')[2]


#adding Strike_rate to the dataframe i.e (Runs_scored/Balls_faced) X 100
Player_runs_per_innings["Strike_rate"]<-round((Player_runs_per_innings['Runs_scored']/
                                                 Player_runs_per_innings['Balls_faced'])*100,1)



id_ciy<-raw_match_data[,1:2]
Player_runs_per_innings<-merge(Player_runs_per_innings,id_ciy,by="id")

df_raw<-raw_match_data %>% filter(player_of_match==player)
df_raw<-df_raw[,c(1,4)]
df_raw["Man_of_match"]<-rep(1,length(df_raw$player_of_match))
df_raw<-df_raw[,c(1,3)]

Player_runs_per_innings<-merge(Player_runs_per_innings,df_raw,by="id",all.x=TRUE)
Player_runs_per_innings[is.na(Player_runs_per_innings)]<-0



# Adding Nos of 30 scores ( a milestone in T20 cricket)

Score_30<-Player_runs_per_innings %>% filter(Runs_scored >=30 & Runs_scored <=49)

Score_30<-Score_30[,1:2]
Score_30["No_30"]<-rep(1,length(Score_30$Runs_scored))
Score_30<-Score_30[,c(1,3)]
Player_runs_per_innings<-merge(Player_runs_per_innings,Score_30,by="id",all.x=TRUE)
Player_runs_per_innings[is.na(Player_runs_per_innings)]<-0

# Adding Nos of 50 scores ( a milestone in T20 cricket)
```

```
Score_50<-Player_runs_per_innings %>% filter(Runs_scored >=50 & Runs_scored <=99)

Score_50<-Score_50[,1:2]
Score_50["No_50"]<-rep(1,length(Score_50$Runs_scored))
Score_50<-Score_50[,c(1,3)]
Player_runs_per_innings<-merge(Player_runs_per_innings,Score_50,by="id",all.x=TRUE)
Player_runs_per_innings[is.na(Player_runs_per_innings)]<-0

# Adding Nos of 100 scores ( a big milestone in T20 cricket)

Score_100<-Player_runs_per_innings %>% filter(Runs_scored >=100)

Score_100<-Score_100[,1:2]
Score_100["No_100"]<-rep(1,length(Score_100$Runs_scored))
Score_100<-Score_100[,c(1,3)]
Player_runs_per_innings<-merge(Player_runs_per_innings,Score_100,by="id",all.x=TRUE)
Player_runs_per_innings[is.na(Player_runs_per_innings)]<-0




Player_runs_per_innings<- Player_runs_per_innings%>%
  mutate(Runs_category= case_when(Runs_scored >= 80 ~ '80_Plus',Runs_scored >= 50 ~ '50_79',
                                  Runs_scored >= 30 ~ '30_49', Runs_scored >= 10 ~ '10_29',
                                  Runs_scored >= 0 ~ '0_9',TRUE ~ 'Low'))

return(Player_runs_per_innings)}

#year in dataframe
date_id<-raw_match_data[,c(1,3)]



Gayle<-shortlisted_players[1]
Gayle_Stats<-Player_stats_model(Gayle)
Gayle_Stats<-merge(Gayle_Stats,date_id,by="id",all.x=TRUE)

write.csv(Gayle_Stats, file ='Gayle_bat.csv')

Gayle_train_test<-Gayle_Stats%>% filter(date <= 2017)
Gayle_pred_data<-Gayle_Stats%>% filter(date > 2017 & date <=2020)

Pollard<-shortlisted_players[2]
Pollard_Stats<-Player_stats_model(Pollard)
Pollard_Stats<-merge(Pollard_Stats,date_id,by="id",all.x=TRUE)
write.csv(Pollard_Stats, file ='Pollard_bat.csv')

Pollard_train_test<-Pollard_Stats%>% filter(date <= 2017)
Pollard_pred_data<-Pollard_Stats%>% filter(date > 2017 & date <=2020)
# 2 years dataset for prediction
write.csv(Pollard_pred_data, file ='Pollard_pred_data.csv')

Watson<-shortlisted_players[3]
```

```
Watson_Stats<-Player_stats_model(Watson)
Watson_Stats<-merge(Watson_Stats,date_id,by="id",all.x=TRUE)
write.csv(Watson_Stats, file ='Watson_bat.csv')

Watson_train_test<-Watson_Stats%>% filter(date <= 2017)
Watson_pred_data<-Watson_Stats%>% filter(date > 2017 & date <=2020)
# 2 years dataset for prediction
write.csv(Watson_pred_data, file ='Watson_pred_data.csv')

Yuvraj<-shortlisted_players[4]
Yuvraj_Stats<-Player_stats_model(Yuvraj)
Yuvraj_Stats<-merge(Yuvraj_Stats,date_id,by="id",all.x=TRUE)
write.csv(Yuvraj_Stats, file ='Yuvraj_bat.csv')

Yuvraj_train_test<-Yuvraj_Stats%>% filter(date <= 2017)
Yuvraj_pred_data<- Yuvraj_Stats%>% filter(date > 2017 & date <=2020)
# 2 years dataset for prediction
write.csv(Yuvraj_pred_data, file ='Yuvraj_pred_data.csv')

Yusuf<-shortlisted_players[5]
Yusuf_Stats<-Player_stats_model(Yusuf)
Yusuf_Stats<-merge(Yusuf_Stats,date_id,by="id",all.x=TRUE)
write.csv(Yusuf_Stats, file ='Yusuf_bat.csv')


Yusuf_train_test<-Yusuf_Stats%>% filter(date <= 2017)
Yusuf_pred_data<- Yusuf_Stats%>% filter(date > 2017 & date <=2020)
# 2 years dataset for prediction
write.csv(Yusuf_pred_data, file ='Yusuf_pred_data.csv')

#Overall dataset for training and testion till the year 2017
bat_train_test_dataset<-rbind(Gayle_train_test,Watson_train_test,
                              Pollard_train_test,Yuvraj_train_test,Yusuf_train_test)
write.csv(bat_train_test_dataset, file ='bat_train_test_dataset.csv')
```

Creating Inning-wise bowling dataframe

```
Player_stats__bowl_model<- function(player){


Player_stats_bowler<-grouped_bowler %>% filter(bowler==player)

wkts_to_bowler_df<-Player_stats_bowler[!(Player_stats_bowler$dismissal_kind=="retired hurt" |
                                           Player_stats_bowler$dismissal_kind=="run out") |
                                         Player_stats_bowler$dismissal_kind=="obstructing the field" ,]

Player_wicket_per_innings<- aggregate(is_wicket ~ id, data=wkts_to_bowler_df, FUN=sum)
colnames(Player_wicket_per_innings)<-c("id","Wickets_taken")


# adding Overs_bowled to the dataframe
```

```r
Overs_bowled<- aggregate(over  ~ id, Player_stats_bowler, function(x) length(unique(x)))[2]
Player_wicket_per_innings["Overs_bowled"]<-Overs_bowled
Player_wicket_per_innings
```

```r
Player_wicket_per_innings["batting_team"]<- aggregate(batting_team~ id,
                                                      wkts_to_bowler_df, function(x) unique(x))[2]

Player_wicket_per_innings["bowling_team"]<- aggregate(bowling_team~ id,
                                                      wkts_to_bowler_df, function(x) unique(x))[2]

Player_wicket_per_innings["Total_runs_conceded"]<- aggregate(total_runs ~ id,
                                                             wkts_to_bowler_df, function(x) sum(x))[2]

#adding bowling economy  to the dataframe i.e (Runs_conceded/overs bowled)
Player_wicket_per_innings["Bowling_economy"]<-round((Player_wicket_per_innings['Total_runs_conceded']/
                                                      Player_wicket_per_innings['Overs_bowled']),1)
```

```r
id_ciy<-raw_match_data[,1:2]
Player_wicket_per_innings<-merge(Player_wicket_per_innings,id_ciy,by="id")
```

```r
# Adding Nos of 3 Wickets haul ( a milestone in T20 cricket)

Wickets_3<-Player_wicket_per_innings %>% filter(Wickets_taken >2 & Wickets_taken <=4)

Wickets_3<-Wickets_3[,1:2]
Wickets_3["3 Wickets"]<-rep(1,length(Wickets_3$Wickets_taken))
Wickets_3<-Wickets_3[,c(1,3)]
Player_wicket_per_innings<-merge(Player_wicket_per_innings,Wickets_3,by="id",all.x=TRUE)
Player_wicket_per_innings[is.na(Player_wicket_per_innings)]<-0

# Adding Nos of 5 Wickets haul ( a milestone in T20 cricket)

Wickets_5<-Player_wicket_per_innings %>% filter(Wickets_taken >=5 )

Wickets_5<-Wickets_5[,1:2]
Wickets_5["5 Wickets"]<-rep(1,length(Wickets_5$Wickets_taken))
Wickets_5<-Wickets_5[,c(1,3)]
Player_wicket_per_innings<-merge(Player_wicket_per_innings,Wickets_5,by="id",all.x=TRUE)
Player_wicket_per_innings[is.na(Player_wicket_per_innings)]<-0


Player_wicket_per_innings<- Player_wicket_per_innings %>%
  mutate(Wickets_Category = case_when(Wickets_taken > 3 ~ '4',
                                      Wickets_taken > 2 ~ '3', Wickets_taken >=  1 ~ '2'
                                      , Wickets_taken == 0 ~ '1',TRUE ~ 'Low'))
```

```r
return(Player_wicket_per_innings)}

#year in dataframe
date_id<-raw_match_data[,c(1,3)]



Gayle<-shortlisted_players[1]
Gayle_Stats_bowl<-Player_stats__bowl_model(Gayle)
Gayle_Stats_bowl<-merge(Gayle_Stats_bowl,date_id,by="id",all.x=TRUE)

write.csv(Gayle_Stats_bowl, file ='Gayle_bowl.csv')

Gayle_train_test_bowl<-Gayle_Stats_bowl%>% filter(date <= 2014)
Gayle_pred_data_bowl<-Gayle_Stats_bowl%>% filter(date > 2014 & date <=2020)

Pollard<-shortlisted_players[2]
Pollard_Stats_bowl<-Player_stats__bowl_model(Pollard)
Pollard_Stats_bowl<-merge(Pollard_Stats_bowl,date_id,by="id",all.x=TRUE)
write.csv(Pollard_Stats_bowl, file ='Pollard_bowl.csv')

Pollard_train_test_bowl<-Pollard_Stats_bowl%>% filter(date <= 2014)
Pollard_pred_data_bowl<-Pollard_Stats_bowl%>% filter(date > 2014 & date <=2020)
write.csv(Pollard_pred_data_bowl, file ='Pollard_pred_data_bowl.csv')


Watson<-shortlisted_players[3]
Watson_Stats_bowl<-Player_stats__bowl_model(Watson)
Watson_Stats_bowl<-merge(Watson_Stats_bowl,date_id,by="id",all.x=TRUE)
write.csv(Watson_Stats_bowl, file ='Watson_bowl.csv')

Watson_train_test_bowl<-Watson_Stats_bowl%>% filter(date <= 2014)
Watson_pred_data_bowl<-Watson_Stats_bowl%>% filter(date > 2014 & date <=2020)
write.csv(Watson_pred_data_bowl, file ='Watson_pred_data_bowl.csv')

Yuvraj<-shortlisted_players[4]
Yuvraj_Stats_bowl<-Player_stats__bowl_model(Yuvraj)
Yuvraj_Stats_bowl<-merge(Yuvraj_Stats_bowl,date_id,by="id",all.x=TRUE)
write.csv(Yuvraj_Stats_bowl, file ='Yuvraj_bowl.csv')

Yuvraj_train_test_bowl<-Yuvraj_Stats_bowl%>% filter(date <= 2014)
Yuvraj_pred_data_bowl<-Yuvraj_Stats_bowl%>% filter(date > 2014 & date <=2020)

write.csv(Yuvraj_pred_data_bowl, file ='Yuvraj_pred_data_bowl.csv')

Yusuf<-shortlisted_players[5]
Yusuf_Stats_bowl<-Player_stats__bowl_model(Yusuf)
Yusuf_Stats_bowl<-merge(Yusuf_Stats_bowl,date_id,by="id",all.x=TRUE)
write.csv(Yusuf_Stats_bowl, file ='Yusuf_bowl.csv')

Yusuf_train_test_bowl<-Yusuf_Stats_bowl%>% filter(date <= 2014)
```

```
Yusuf_pred_data_bowl<-Yusuf_Stats_bowl%>% filter(date > 2014 & date <=2020)
write.csv(Yusuf_pred_data_bowl, file ='Yusuf_pred_data_bowl.csv')


#Overall dataset for training and testion till the year 2017
train_test_dataset_bowl<-rbind(Gayle_train_test_bowl,Watson_train_test_bowl,
                               Pollard_train_test_bowl,Yuvraj_train_test_bowl,
                               Yusuf_train_test_bowl)
write.csv(train_test_dataset_bowl, file ='train_test_dataset_bowl.csv')


write.csv(Players_statistics_final, file ='Players_statistics_final.csv')

write.csv(Players_statistics_final, file ='Players_statistics_final.csv')

Players_statistics_final_t <- as.data.frame(t(Players_statistics_final))
write.csv(Players_statistics_final_t, file ='Players_statistics_final_t.csv')
```

Code Appendix