PES UNIVERSITY EC Campus
Electronic City, Hosur Road,
Bangalore-100

A Project Report on

Indian Premier League Match Analysis

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

For the Academic year 2020

by

Abhishek V	**PES2UG19CS012**
Anmol	**PES2UG19EC017**
Balendra D P	**PES2UG19CS100**
Vineeth Kumar	**PES2UG19ME056**

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

Features of R

As stated earlier, R is a programming language and software environment for statistical analysis, graphics representation and reporting. The following are the important features of R –

- R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- R has an effective data handling and storage facility,
- R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
- R provides a large, coherent and integrated collection of tools for data analysis.
- R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.

Installation and getting started with R

Windows Installation

The Windows installer version of R can be downloaded from R-4.0.3 for Windows (32/64 bit) and save it in a local directory.

As it is a Windows installer (.exe) with a name "R-version-win.exe", Double click and run the installer accepting the default settings

After installation the icon to run the Program is located in a directory structure "R\R4.0.3\bin\i386\Rgui.exe" under the Windows Program Files. Clicking this icon brings up the R-GUI which is the R console to execute R programs.

Linux Installation

R is available as a binary for many versions of Linux at the location R Binaries.

The instruction to install Linux varies from flavour to flavour. These steps are mentioned under each type of Linux version in the mentioned link.

R Integrated Development Environment

Rstudio

Rstudio is the premier integrated development environment for R. It is available in open source and commercial editions on the desktop (Windows, Mac, and Linux) and from a web browser to a Linux server running Rstudio Server.

Google Colab

Colab is an interactive notebook provided by Google for writing and running Python and R through a browser. We can perform data analysis, create models, evaluate these models in Colab. The processing is done on Google-owned servers in the cloud. We only need a browser and a fairly stable internet connection.

System Configurations

The Minimum system requirements are:

1. 4 GB RAM
2. 2 Core processor
3. 50 GB free Disk space
4. Stable internet connection (online notebook)

Libraries Used

The primary uses of R is and will always be, statistic, visualization, and machine learning. About 12,000 libraries of R are stored in CRAN. CRAN is a free and open source. We can download and use the numerous libraries to perform various operations.

The libraries used in this Project are:

1. ggplot2
2. corrplot
3. plotrix

The Data Set

The data set contains details of the matches in Indian Premier League (IPL) played in the years 2008 to 2019.
The data set contains 756 records and 18 variables.

Description of Data Set Variables:

1. Id: Unique ID given to every match played
2. Season: There are 11 seasons
3. Date: Date of match
4. City: Name of the city where the match was played
5. Team 1: Name of the team
6. Team 2: Name of the team
7. Toss Winner: Team that won the toss
8. Toss Decision: Decision bat or field made by the toss winner
9. Result: Tells whether the match was normal or a tie
10. Winner: The winning team
11. DL Applied: States whether Duckworth Lewis Stern method was applied
12. Win by Runs: No. of runs the match was won by
13. Win by wickets: No. of wickets the match was won by
14. Player of the Match: Player who received the player of the match award
15. Venue: The stadium where the match was played
16. Umpire 1: Umpire name
17. Umpire 2: Umpire name
18. Umpire 3: TV umpire name

Scope of Data Analysis

Draw analysis, team performance, apply and learn statistical methods on real data.

CHAPTER-2

Data Structures

A data structure is a particular way of organizing data in a computer so that it can be used effectively.
R has the following basic data structures:

1.Vectors
Vectors are single-dimensional, homogeneous data structures. To create a vector, use the c( ) function.
Storing win by runs in a Vector: `runs = c(df$win_by_runs)`

2. List
Lists are heterogeneous data structures. They are very similar to vectors except they can store data of different types. To create a list, we use the list() function.
Storing team names in a List: `list(unique(df$team1))`

3. Matrix
Matrices are two-dimensional, homogeneous data structures. This means that all values in a matrix have to be of the same type. Coercion takes place if there is more than one data type. They have rows and columns.

4. Data Frame
Data frames are two-dimensional, heterogeneous data structures. They are lists of vectors of equal lengths.
Importing the dataset

```
urlfile<-'https://raw.githubusercontent.com/Abhishek4848/IPL-Match-Analysis/main/matches.csv'
df<-read.csv(url(urlfile))
```

Displaying the Top 5 and the bottom 5 records of the data frame

```
head(df,5)
tail(df,5)
```

5. Arrays
Arrays are three dimensional, homogeneous data structures. They are collections of matrices stacked one on top of the other in layers.
Storing win by wickets in a Array: `wick = array(df$win_by_wickets)`

CHAPTER-3

String and Date Operations

Operations on Date

R provides several options for dealing with date and date/time data and formatting it.. The built in as.Date function handles dates.

Finding the No. days where 2 matches were played

```
dates = data.frame(sort(table(df$date),decreasing = T))
colnames(dates) = c("Date","No. of matches")
dates$Date = format(as.Date(dates$Date),format = "%d/%m/%y")
#converting character string to date object and formatting it
n = 0
for (x in dates$`No. of matches`)
{
  if (x == 2)
  {
    n=n+1
  }
}
paste("No. of matches held on weekends: ",n)
```

Output: 'No. of matches held on weekends: 210'

Generally, 2 matches are held on the same day if it's a weekend.

String Operations

Comparing strings and fixing inconsistencies in data

```
df$winner[df$winner=="Rising Pune Supergiant"] = "Rising Pune Supergian
ts"
df$winner[df$toss_winner=="Rising Pune Supergiant"] = "Rising Pune Supe
rgiants"
df$city[df$city=="Bengaluru"] = "Bangalore"
```

CHAPTER-4

Looping and Conditional Statements on Data Structures

Looping and conditional statement on **Vector** to find Biggest Margin of Victory by Batting team.

```
Max = df$win_by_runs[1]
runs = c(df$win_by_runs)
for(runs in runs)
{
  if(runs > max)
  {
    max = runs
  }
}
Max
```

Output: 146

```
df$winner[df$win_by_runs == max]
```

Output: 'Mumbai Indians'

Mumbai Indians defeated Delhi Daredevils by 146 runs which is the highest win margin in any IPL match.

```
paste(df$team1[df$win_by_runs == max], "vs", df$team2[df$win_by_runs ==
 max])
```

Output: 'Mumbai Indians vs Delhi Daredevils'

Looping and conditional statement on **Data Frame** to find the win percentage based on toss decision.

```
First_field_win = 0
for(x in seq(1,length(df$id)))
{
  if(df$toss_decision[x] == "field" & df$toss_winner[x] == df$winner[x]
)
  {
    first_field_win = first_field_win + 1
  }
}
first_field_win
```

Output: 259

```
first_bat_win = 0
for(x in seq(1,length(df$id)))
{if(df$toss_decision[x] == "bat" & df$toss_winner[x] == df$winner[x])
  {
    first_bat_win = first_bat_win + 1
  }
}
first_bat_win
```

Output: 134

```
paste("Win % when the toss winner chose to FIELD first: ",round((first_
field_win/length(df$id))*100,digits = 2),"%")
paste("Win % when the toss winner chose to BAT first: ",round((first_ba
t_win/length(df$id))*100,2),"%")
```

Output:

'Win % when the toss winner chose to FIELD first: 34.26 %'
'Win % when the toss winner chose to BAT first: 17.72 %'

Higher no. of matches are won when the toss winner opts to field first.

Looping and conditional statement on **Array** to find Biggest Margin of Victory by Bowling team.

```
Max = df$win_by_wickets[1]
wick = array(df$win_by_wickets)
for(wickets in wick)
{
  if(wickets > max)
  {
    max = wickets
  }
}
max
```

Output: 10

When chasing a target, the biggest victory was by 10 wickets.

Finding the mean Win by Runs using **Recursion**

```
findMean = function(A,len)
{
  if (len == 1)
    return (A[len])
  else
    return((findMean(A,len-1)*(len-1)+A[len-1])/len)
}
findMean(df$win_by_runs,length(df$win_by_runs))
```

Output: 13.328

The mean win by runs is 13

CHAPTER-5

Built-in R Functions

R provides a wide range of functions that can be used for data manipulations and data analysis.

Using **rank** function to rank the teams based on the no. of victories

```
rank(table(df$winner))
```

Output:

```
Chennai Super Kings1 Deccan Chargers7 Delhi Capitals3 Delhi Daredevils9
Gujarat Lions5 Kings XI Punjab11 Kochi Tuskers Kerala2 Kolkata Knight
Riders13 Mumbai Indians15 Pune Warriors4 Rajasthan Royals10 Rising Pune
Supergiants6 Royal Challengers Bangalore12 Sunrisers Hyderabad8
```

Using **all** function to Check whether every team which played has won at least one match

```
all_teams = unique(c(df$team1,df$team2))
all(all_teams %in% df$winner)
```

Output: `TRUE`

Using **which** function to find the teams that have won a match by more than 50 runs

```
t = c()
teams50 = which(df$win_by_runs > 50)
for(y in teams50)
{
  if(y == df$id[y])
  {
    t = append(t,df$winner[y])
  }
}
unique(t)
```

Output:

1. 'Delhi Daredevils'
2. 'Kolkata Knight Riders'
3. 'Rising Pune Supergiants'
4. 'Mumbai Indians'
5. 'Kings XI Punjab'
6. 'Rajasthan Royals'
7. 'Royal Challengers Bangalore'
8. 'Chennai Super Kings'
9. 'Deccan Chargers'
10. 'Sunrisers Hyderabad'

Using **setdiff** function to find the teams that have NOT won a match by more than 50 runs

```
setdiff(all_teams,unique(t))
```

Output:

1. 'Gujarat Lions'
2. 'Kochi Tuskers Kerala'
3. 'Pune Warriors'
4. 'Delhi Capitals'

Using **match** function to find the best team for Winning by Runs

```
mode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
mode(df$venue[df$win_by_runs])
```

Output: 'Wankhede Stadium'

Using **any** function to check if any match has been won by more than 150 runs

```
any(which(df$win_by_runs >150))
```

Output: FALSE

Finding the top 10 players who have the maximum no. of player of the match title using **sort** function

```
player_match = data.frame(sort(table(df$player_of_match),decreasing = T
))
colnames(player_match) = c("Player Name","No. Player of the Match award
s won")
head(player_match,10)
```

Output:

A data.frame: 10 × 2

	Player Name	No. Player of the Match awards won
	<fct>	<int>
1	CH Gayle	21
2	AB de Villiers	20
3	DA Warner	17
4	MS Dhoni	17
5	RG Sharma	17
6	YK Pathan	16
7	SR Watson	15
8	SK Raina	14
9	G Gambhir	13
10	AM Rahane	12

Five Indian players have figured in the top 10 IPL players list

CHAPTER-6

Exploratory Data Analysis

Exploratory data analysis is the process of performing investigations on data so as to discover patterns, spot anomalies and to test for hypothesis with the help of summary statistics and graphical visualizations.

The Correlation Matrix

Correlation matrix is used to get insights about the relationships between the variables of the dataset

```
x = data.frame(df$season,df$win_by_runs,df$win_by_wickets,
df$dl_applied)
corrplot(cor(x),method="pie")
```

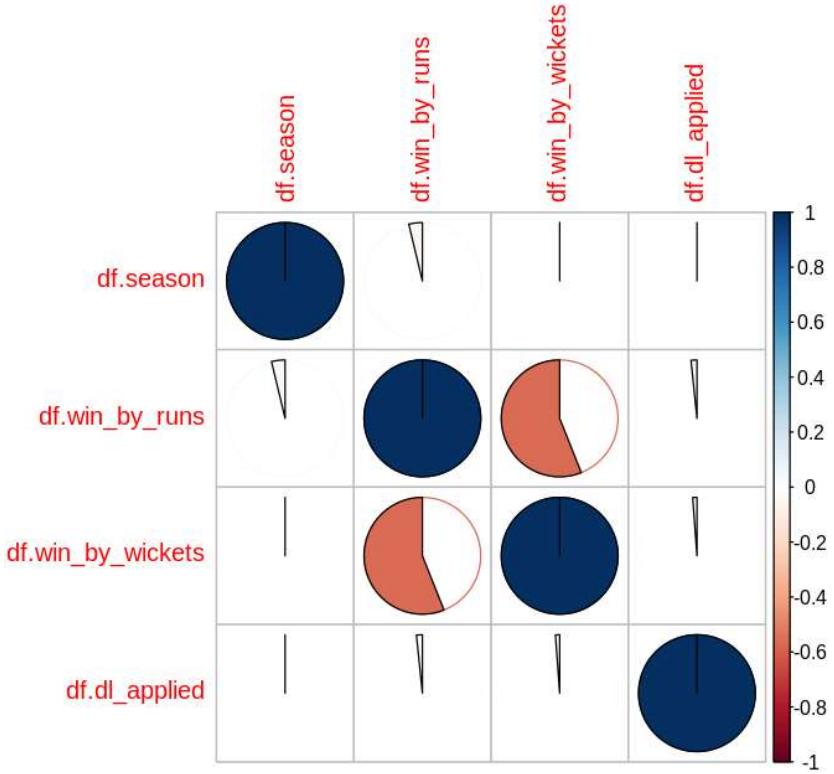


Figure 1

From the above correlation matrix, we can infer that there's no strong relationships between the different numeric variables of the data frame.

Pie Chart

A pie chart is a circular chart that shows how data sets relate to one another. The arclength of each section is proportional to the quantity it represents, usually resulting in a shape similar to a slice of pie.

Pie chart representing the most successful teams in the history of IPL

```
pie(table(df$winner),main="Win % Of Teams")
```

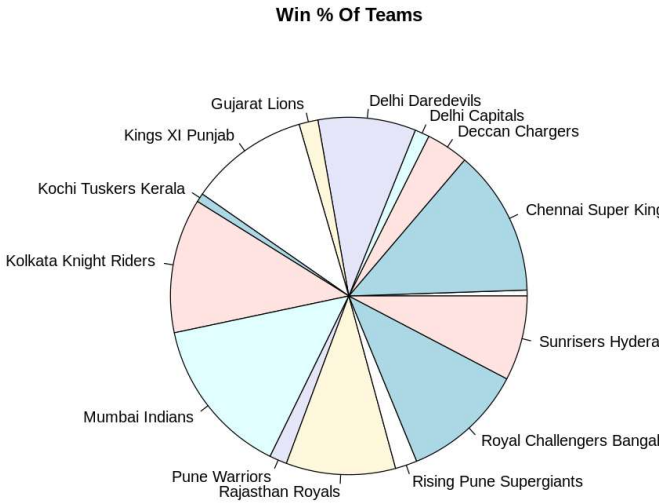


Figure 2

From this visualization it's evident that MI and CSK are the most successful teams.

Pie chart representing the Toss Decision

```
toss = data.frame(table(df$toss_decision))
colnames(toss) = c("Decision","Count")
pieper = round(100*toss$Count/sum(toss$Count), 1)
toss
pie3D(table(df$toss_decision),labels=pieper,main="Toss Decision",col=c(
"brown","green"))
legend("topright",c("Bat","Field"),fill=c("brown","green"))
```

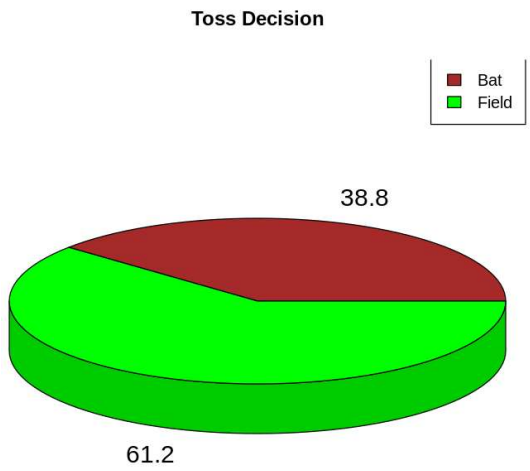


Figure 3

61.2% of the toss winners chose to field first

Pie chart representing the DLS Status

```
dls = data.frame(table(df$dl_applied))
colnames(dls) = c("Status","Count")
dls$Status = as.character(dls$Status)
dls$Status[dls$Status == "0"] = "Not Applied"
dls$Status[dls$Status == "1"] = "Applied"
piepercent = round(100*dls$Count/sum(dls$Count), 1)
pie3D(dls$Count, labels=piepercent,col=c("yellow","blue"),main="DLS")
legend("topright",c(dls$Status),fill=c("yellow","blue"))
dls
```

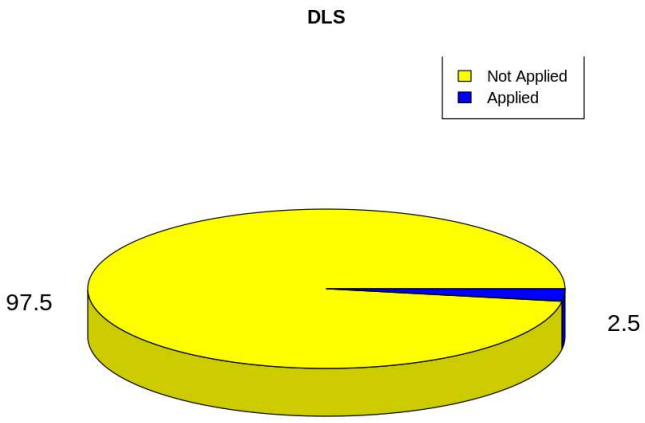


Figure 4

A very few matches (2.5 %) was affected due to bad weather conditions. Hence DLS method was used.

Bar graphs

The bar plot helps plot data in rectangular bins that represent the total amount of observations in the data for that category.

Bar graph representing the No. of matches played in a season

```
b = barplot(table(df$season), xlab="Season",ylab="No. of matches palyed
",main="No. of matches played in each season",col="cyan")
text(x=b, y= table(df$season)+1, labels=as.character(table(df$season)))
```

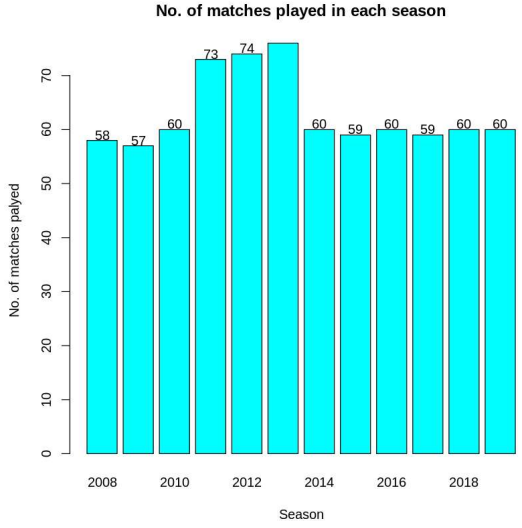
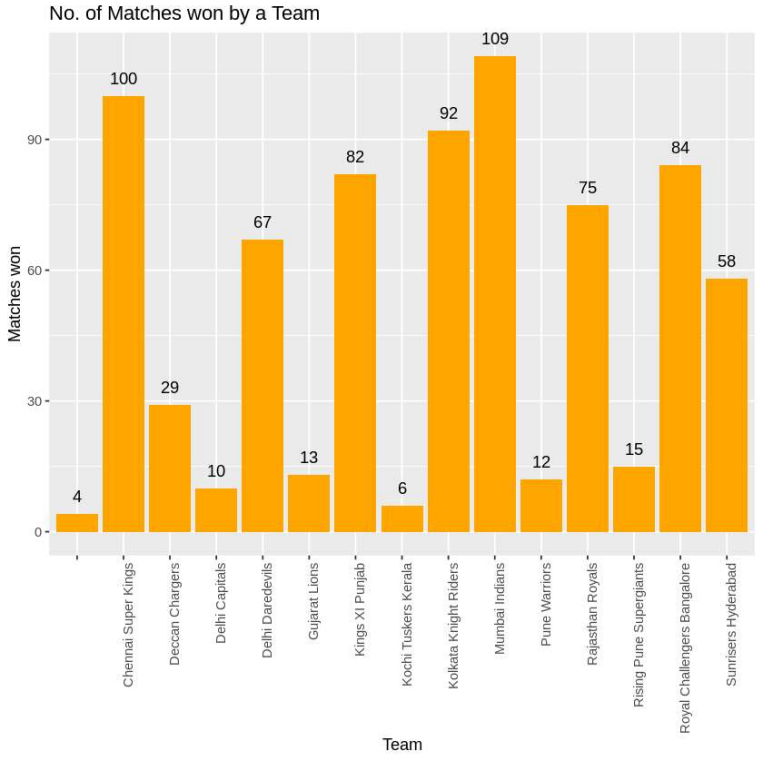


Figure 5

Higher no. of matches was played in the seasons 2011-2013 due to more no. of playing teams

Bar graph representing the No. of matches won by a team

```
ggplot(df,aes(winner)) +geom_bar(fill="orange") +theme(axis.text.x = el
ement_text(angle = 90, hjust = 1))+ xlab("Team")+ylab("Matches won")+
stat_count(aes(y=..count..,label=..count..),geom="text",vjust=-
1)+ggtitle("No. of Matches won by a Team")
```



From this visualization it's evident that MI and CSK are the most successful teams.

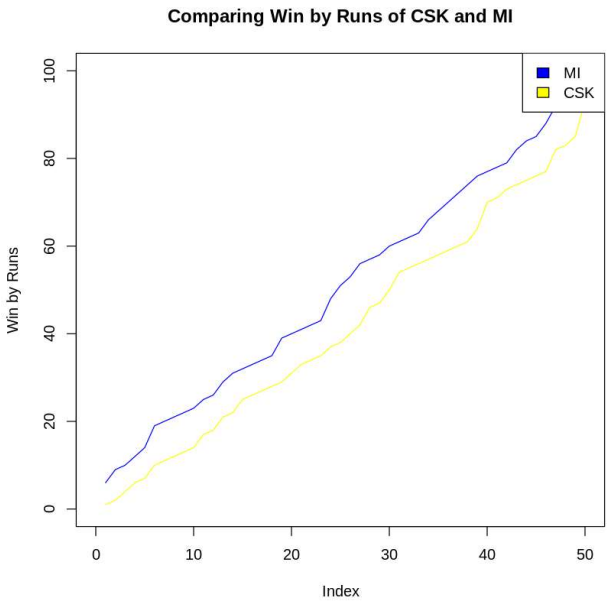
Figure 6

Line Graph

The line graph measures change over time by plotting individual data points connected by straight lines.

Line graph comparing Win by Runs of Mi and CSK

```
c = which(df$win_by_runs[df$winner=="Chennai Super Kings"]>0)
m = which(df$win_by_runs[df$winner=="Mumbai Indians"]>0)
plot(c,type="l",col="yellow",main="Comparing Win by Runs of CSK and MI"
,ylab="Win by Runs",ylim = c(0,100),xlim=c(0,50))
lines(m,type="l",col="blue")
legend("topright",c("MI","CSK"),fill=c("blue","yellow"))
```



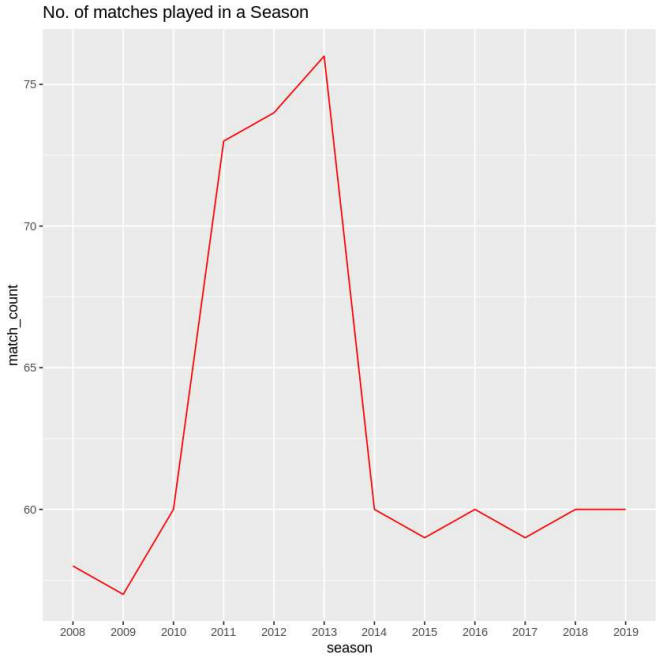
Mumbai Indians has won the matches by a higher margin of runs than Chennai Super Kings.

Figure 7

Mumbai Indians have won more matches than Chennai Super Kings by a higher margin of runs

Line Graph representing No. of matches played in a Season

```
season = data.frame(table(df$season))
colnames(season) = c("season","match_count")
ggplot(data=season, aes(season,match_count,group=1))+geom_line(color="red")+ggtitle("No. of matches played in a Season")
```



Higher no. of matches was played in the seasons 2011-2013 due to more no. of playing teams

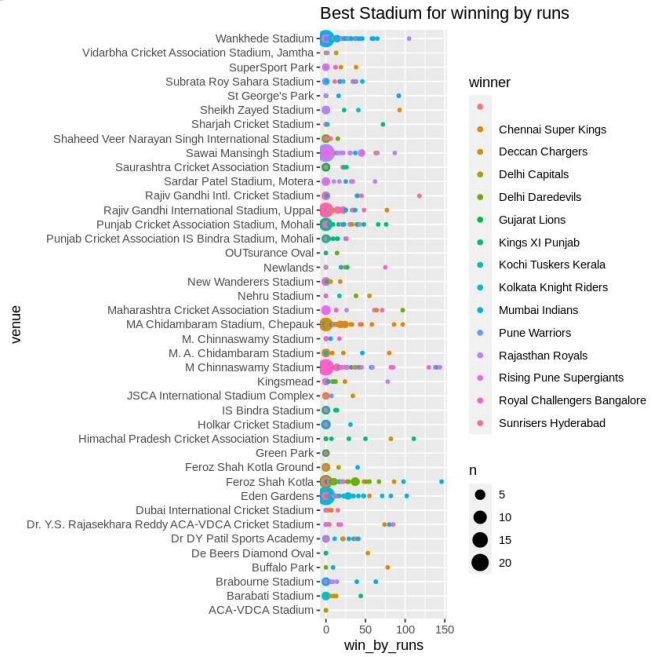
Figure 8

Scatter Plot

A scatter plot uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

Scatter Plot representing the Best stadiums to win by a large margin of runs

```
ggplot(df, aes(win_by_runs, venue, colour = winner)) + geom_count() + g
gtitle("Best Stadium for winning by runs")
```

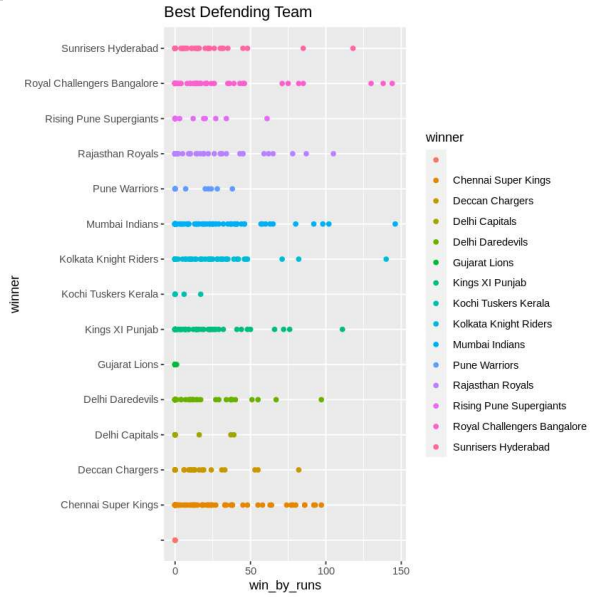


Mumbai Wankhede Stadium is the best stadium for winning by runs

Figure 9

Scatter plot representing the Best Defending Teams

```
ggplot(df, aes(win_by_runs, winner, colour = winner)) + geom_point() +
ggtitle("Best Defending Team")
```



Mumbai Indians is the best defending team

Figure 10

CHAPTER-7

Descriptive Statistics

Descriptive statistics are brief descriptive coefficients that summarize a given data set, which can be either a representation of the entire or a sample of a population.

Measures of Central Tendency

Boxplot representing the Win by Wickets

```
boxplot(df$win_by_wickets, horizontal=T, varwidth=TRUE,outline=TRUE, bo
xwex=0.2, border=c("blue"),xlab = "No. of Wickets",main="Boxplot of Win
 by Wickets")
paste("Mean: ",round(mean(df$win_by_wickets)))
paste("Variance: ",round(var(df$win_by_wickets)))
paste("Standard deviation: ",round(sd(df$win_by_wickets)))
paste("1st Quartile: ",quantile(df$win_by_wickets,prob=c(0.25)))
paste("Median: ",quantile(df$win_by_wickets,prob=c(0.50)))
paste("3rd Quartile: ",quantile(df$win_by_wickets,prob=c(0.75)))
```

Output:

'Mean: 3'
'Variance: 11'
'Standard deviation: 3'
'1st Quartile: 0'
'Median: 4'
'3rd Quartile: 6'

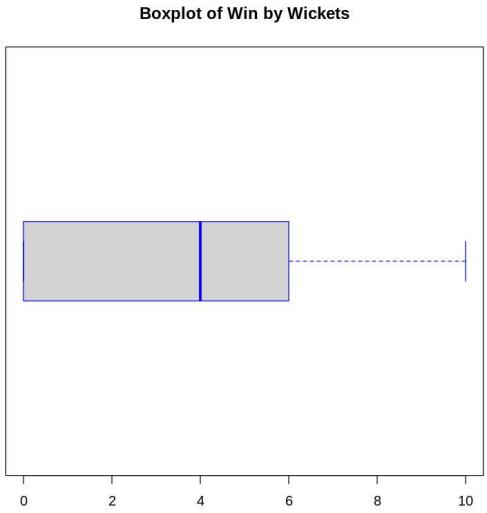


Figure 11

Summary of Win by Runs

```
summary(df$win_by_runs)
```

Output:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.00   13.28   19.00  146.00
```

Boxplot comparing Win by Runs of different teams

```
ggplot(df, aes(winner,win_by_runs,color=winner)) + geom_boxplot() + theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

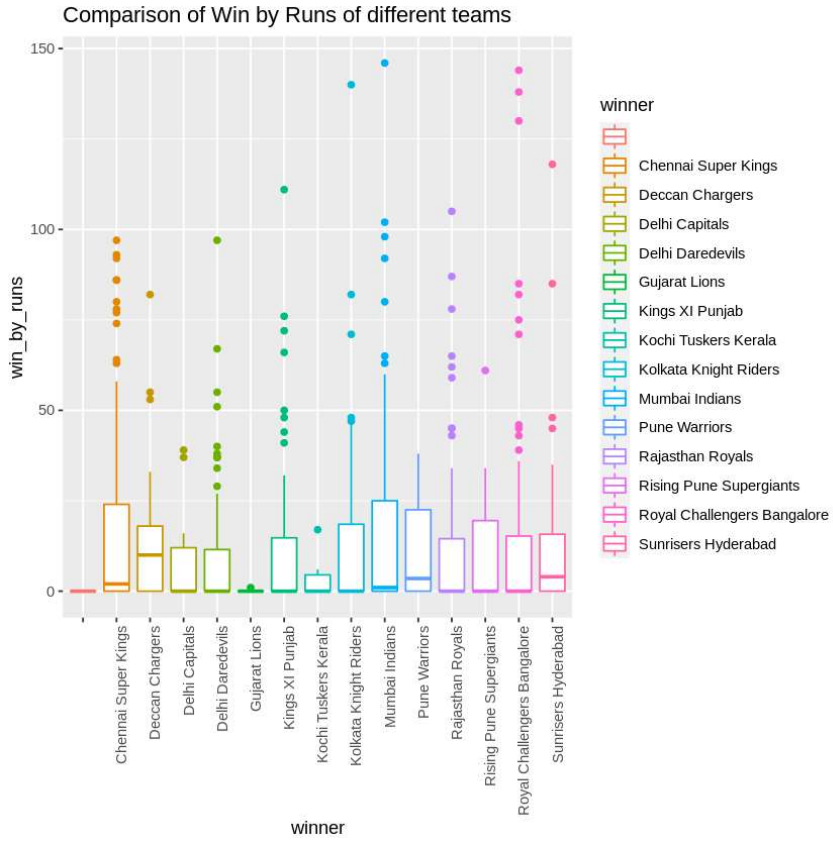


Figure 12

Finding the Best chasing team in IPL using **mode** function

```
mode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
mode(df$winner[df$win_by_wickets])
```

Output: 'Kings XI Punjab'

Kings XI Punjab is the best chasing team in the history of IPL

Histogram

Histograms provide a visual interpretation of numerical data by indicating the number of data points that lie within a range of values.

Histogram representing No. of matches played in a venue

```
library("ggplot2")
ggplot(data = df,aes(venue)) +geom_histogram(fill="violet",stat="count"
) +ylab("No. of Matches Played")+theme(axis.text.x = element_text(angle
 = 90, hjust = 1))+
stat_count(aes(y=..count..,label=..count..),geom="text",vjust=-
1)+ggtitle("No. of matches played in a Venue")
```

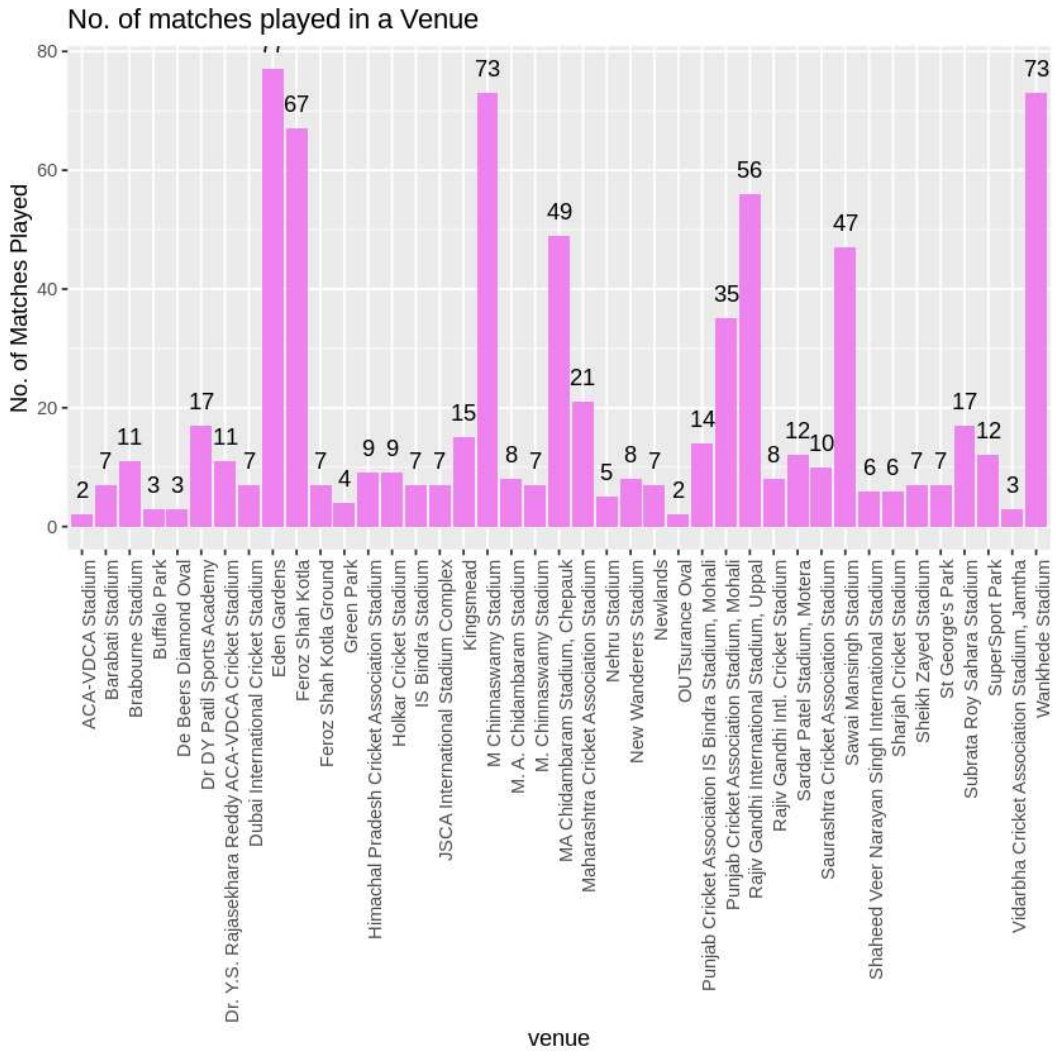


Figure 13

Eden Gardens and Wankhede Stadium have hosted the maximum no. if IPL matches.

CHAPTER-8

Probability Distributions

A probability distribution is a mathematical description of the probabilities of events, subsets of the sample space.

Poisson Distribution

A Poisson distribution is a probability distribution that results from the Poisson experiment. A Poisson experiment is a statistical experiment that classifies the experiment into two categories, such as success or failure.

Probability that a match is won by 10 or less than 10 runs

```
mean_win_runs = mean(df$win_by_runs)
p = ppois(10,mean_win_runs)
p
```

Output: 0.2284

There's a 22% probability that a match is won by 10 or less runs.

Probability that a match is won exactly by 5 runs

```
dpois(5,mean_win_runs)
```

Output: 0.0058

Probability distribution of win by runs

```
bar = dpois(c(0:50),mean_win_runs)
barplot(bar,c(0:50),names.arg=c(0:50),main="Probability distribution of
 win by runs",col=c("red","green","blue"),xlab = "Win by Runs",ylab = "
Probability")
```

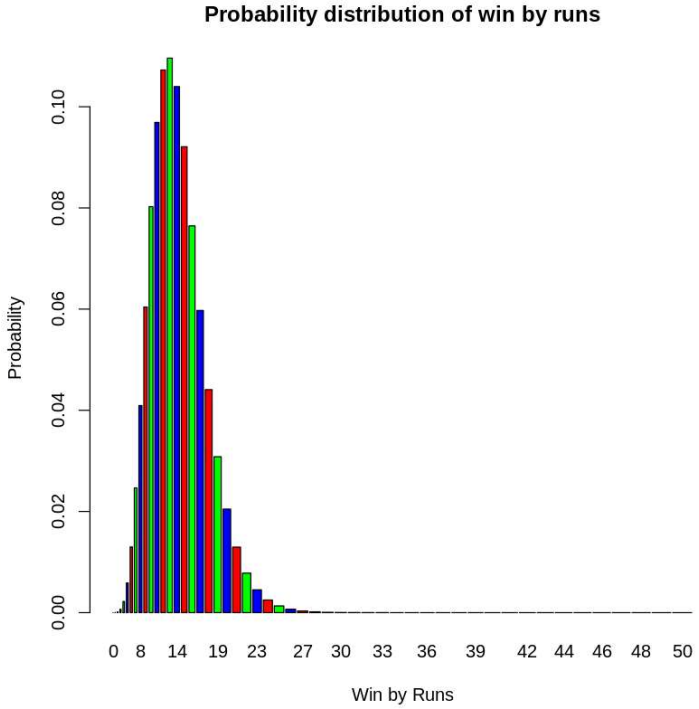


Figure 14

It is less probable that a match is won by a higher margin of runs over 50.

Probability that a match is won by more than 6 wickets

```
mean_win_wickets = mean(df$win_by_wickets)
p1 = 1-ppois(6,mean_win_wickets)
p1
```

Output: 0.0543

There's a 5% probability that a match is won by more than 6 wickets.

Probability distribution of win by Wickets

```
plt = dpois(c(0:10),mean_win_wickets)
barplot(plt,c(0:10),names.arg = c(0:10),xlab="Win by Wickets",ylab="Probability",main="Probability Distribution of Win by Wickets",col=c("black","red"))
```

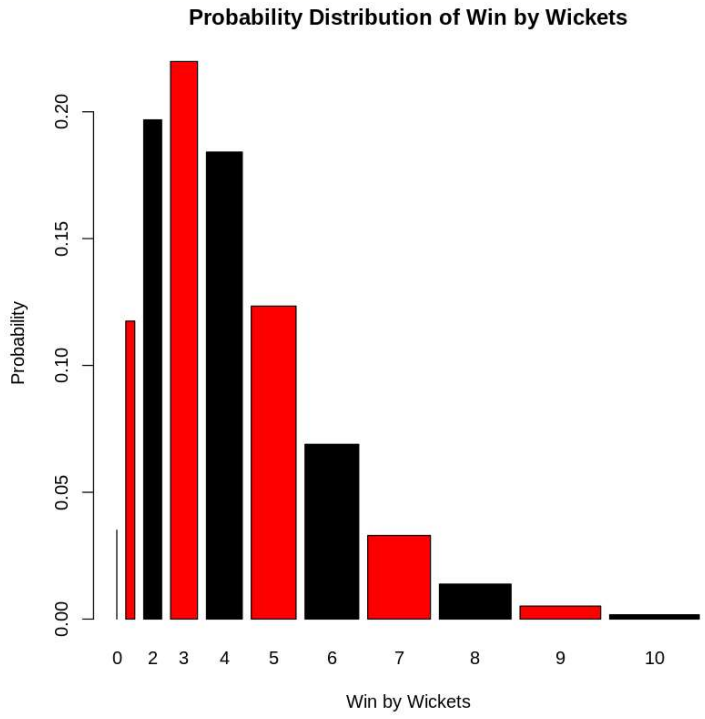


Figure 15

There's high probability that matches are won by a low no. of wickets.

CHAPTER-9

Hypothesis Testing

The general idea of hypothesis testing involves: Making an initial assumption. Collecting evidence (data). Based on the available evidence (data), deciding whether to reject or not reject the initial assumption.

Null Hypothesis H0: Mean win by runs is more than 10 at a confidence level of 90%

Alternate hypothesis H1: Mean win by runs is NOT more than 10

```
sample = sample(df$win_by_runs,75)
sample_mean = mean(sample)
mew = 10
population_sd = sd(df$win_by_runs)
n = length(sample)
z = (sample_mean - mew)/(population_sd/sqrt(n))
pval = pnorm(z, lower.tail=FALSE)
confidence = 0.90
alpha = 1-confidence
if(pval < alpha){
  print("Rejected")
} else {
  print("Accepted")
}
```

Output: `"Accepted"`

Hence, we fail to reject the NULL hypothesis.

It is evident that a match is won by a mean of more than 10 runs with a confidence level of 90%

Q-Q Normal Plot

The Q-Q Normal Plot of A is then the Q-Q plot of A against the standard normal distribution. This will be a straight line if the distribution of A is normal of any mean and standard deviation.

```
qqnorm(df$win_by_runs)
```

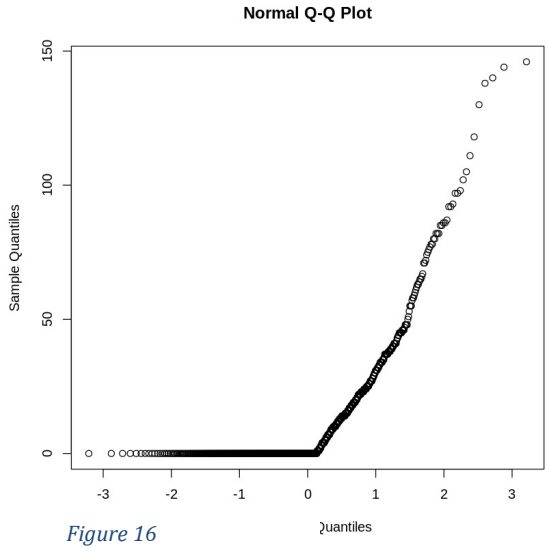


Figure 16

CONCLUSIONS

Different factors effects on the team winning as we seen in the above analysis and above visualized data was accurate to the real world IPL statistics.

Inferences drawn from the IPL Data Analysis:

- Mumbai Indians is the most successful team in IPL.
- Mumbai Indians is the best defending team.
- Till 2019, 40 venues have hosted 756 IPL matches.
- More than 70 matches have been played in the seasons 2011-2013 due to more number of playing teams.
- A few matches was affected due to bad weather(2.5%) hence, Duckworth Lewis Stern method was applied.
- When defending a total, the biggest victory was by 146 runs (Mumbai Indians defeated Delhi Daredevils by 146 runs).
- Mumbai Indians have won more matches than Chennai Super Kings by a higher margin of runs.
- When chasing a target, the biggest victory was by 10 wickets.
- Mumbai has hosted the highest number of IPL matches.
- Mumbai Wankhede Stadium is the best stadium for winning by runs.
- Chris Gayle has won the maximum number of player of the match title.
- Kings XI Punjab is the best chasing team in the IPL.
- 61.2% of the toss winning teams chose to field first.
- 34.26% of the matches are won when the toss winner opts to field first.
- 17.72% of the matches are won when the toss winner opts to bat first.
- Five Indian players have figured in the top ten IPL players list.
- Eden Gardens has hosted the maximum number of IPL matches.
- There's a 5% probability that a match is won by more than 6 wickets.
- It is less probable that a match is won by a higher margin of runs over 50.
- There's high probability that matches are won by 3 wickets.
- The mean win by runs is above 10 for 90% of the matches.

BIBILIOGRAPHY

1. **https://towardsdatascience.com/how-to-use-r-in-google-colab-b6e02d736497**
2. **https://www.reddit.com/r/programming/**
3. **http://manuals.bioinformatics.ucr.edu/home/programming-in-r**
4. **https://www.tutorialspoint.com/r/index.htm**

Links

1. Dataset Used: https://www.kaggle.com/nowke9/ipldata
2. Google Collaboratory: https://colab.research.google.com/drive/10n_r7peQzhCRvb5PSjCIW7fLAd7pySXe?usp=sharing
3. GitHub repository: https://github.com/Abhishek4848/IPL-Match-Analysis