IPL analysis

Introduction

IPL(Indian Premier League) is a 20 over cricket format originated in india. The IPL is being conducted annually since the year 2008. All the data of the league is publicly available on its official website. For this project we will consider two data files, namely.
1)Deliveries- Contains ball by ball data for all the matches from the year 2008-2017
2)Matches- Contains miscellaneous information of match which is not mentioned in the deliveries file like "winner of the match","stadium name" etc.
(Refer to the ReadMe file for more information on the data file)

Analysing and finding insights in any sport can be helpful to both the team owners and the players. Owners can purchase low cost players with higher output, the players themselves can see their own statistics and improve their gameplay.Hence, we will analyse the above datasets by asking a set of questions and potentially finding some valuable insights.
This file will broadly be divided into 5 parts

1)Analysing the scores of the data
2)Phenomena of Sophmore Slump
3)Predicting chance of events
4)Miscellaneous
5)Conclusion/Key insights

PART-1 : Analysing the scores of the data

1 Comparing scores

1.1 Comparing scores of teams First let us compute total runs scored by the team for each and every match.

```
inning_runs=data.frame(expdat%>%group_by(id,inning)%>%summarise(Team=unique(batting_team)
,Runs=sum(total_runs)))
```

Here are first few observations.

```
head(inning_runs)
```

id	inning	Team	Runs
1	1	SRH	207
1	2	RCB	172
2	1	MI	184
2	2	RPS	187
3	1	GL	183
3	2	KKR	184

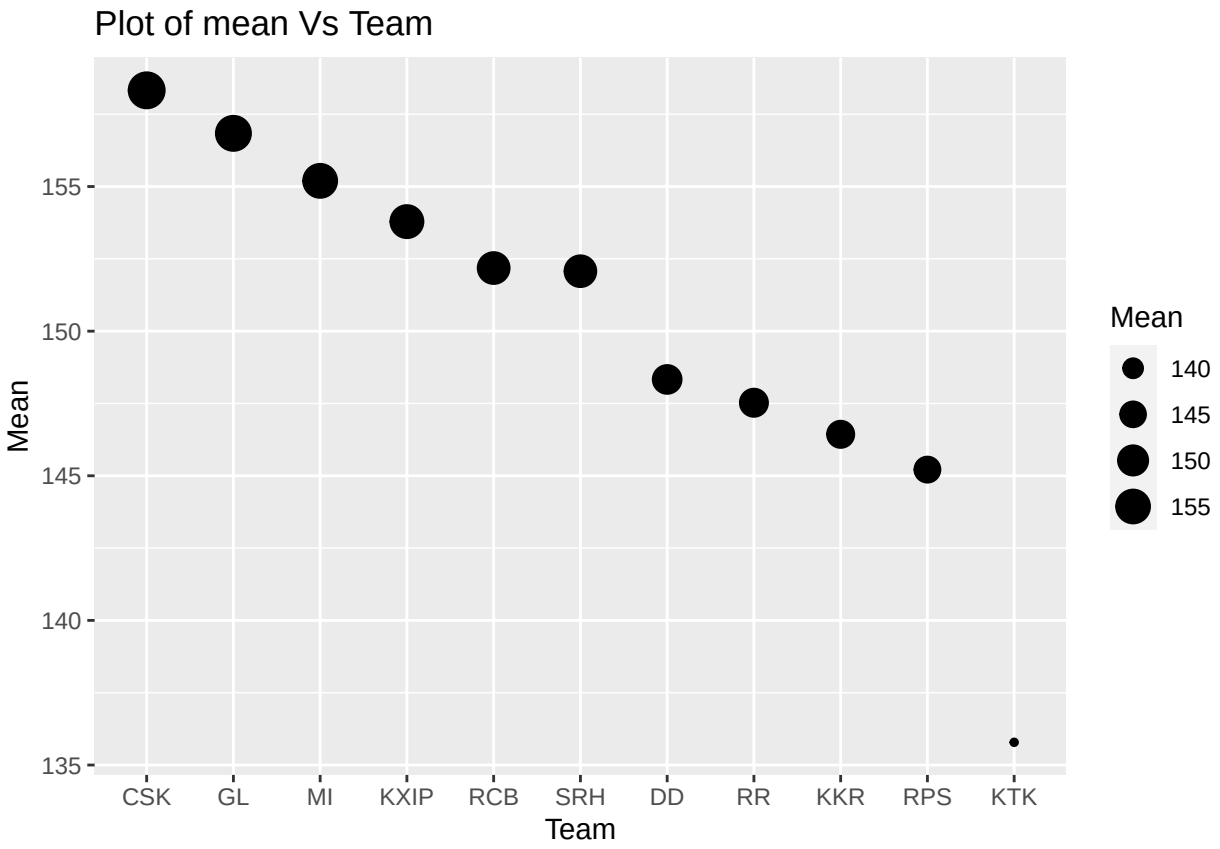
Now for comparision, let us calculate the mean and median scores along with the number of matches played for all the teams throughout the league(i.e from yr.2008 to yr.2017)

```
a1=data.frame(inning_runs%>%group_by(Team)%>%arrange(Runs)%>%summarise(Mean=mean(Runs)
,Median=median(Runs),Matches_played=n()))%>%arrange(desc(Mean))
a1
```

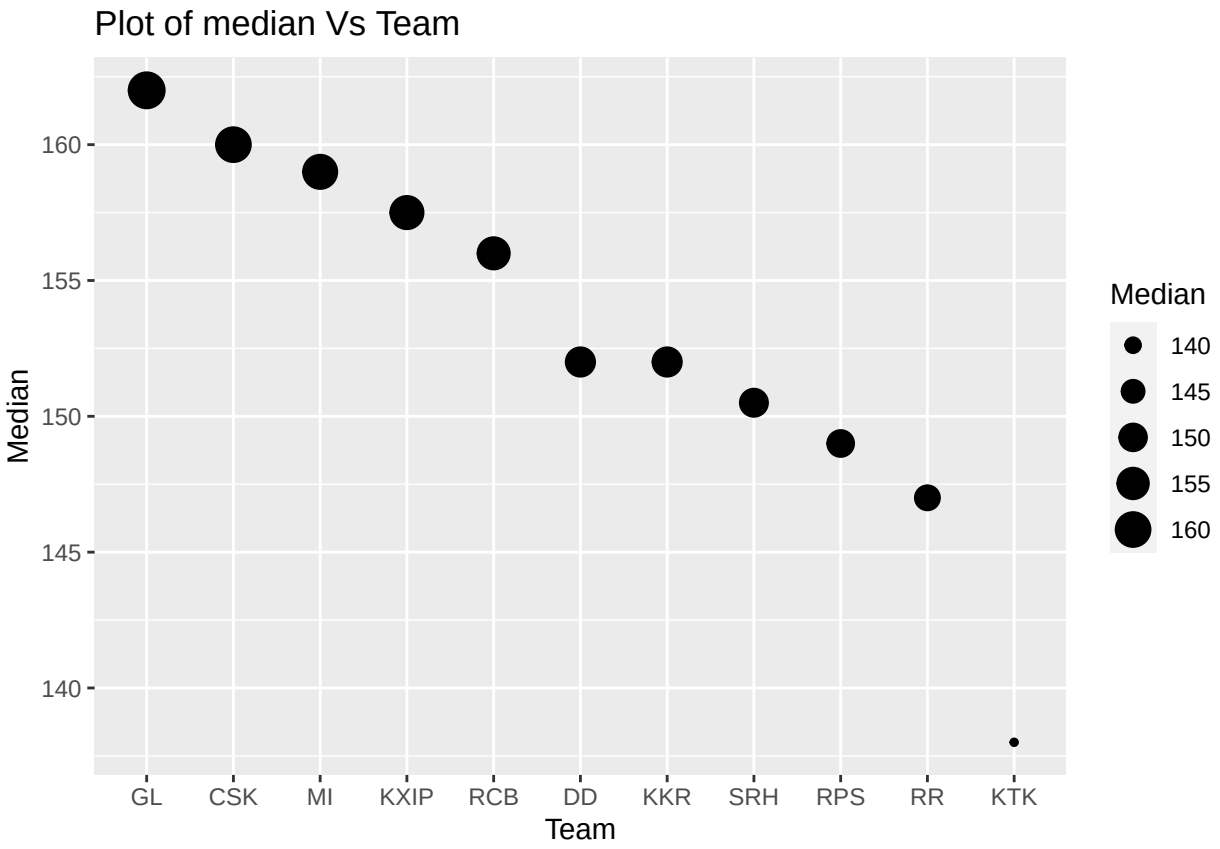
Team	Mean	Median	Matches_played
CSK	158	160	132
GL	157	162	31
MI	155	159	158
KXIP	154	158	150
RCB	152	156	154
SRH	152	150	152
DD	148	152	148
RR	148	147	120
KKR	146	152	150
RPS	145	149	75
KTK	136	138	14

Let us now plot the mean and median scores for each team

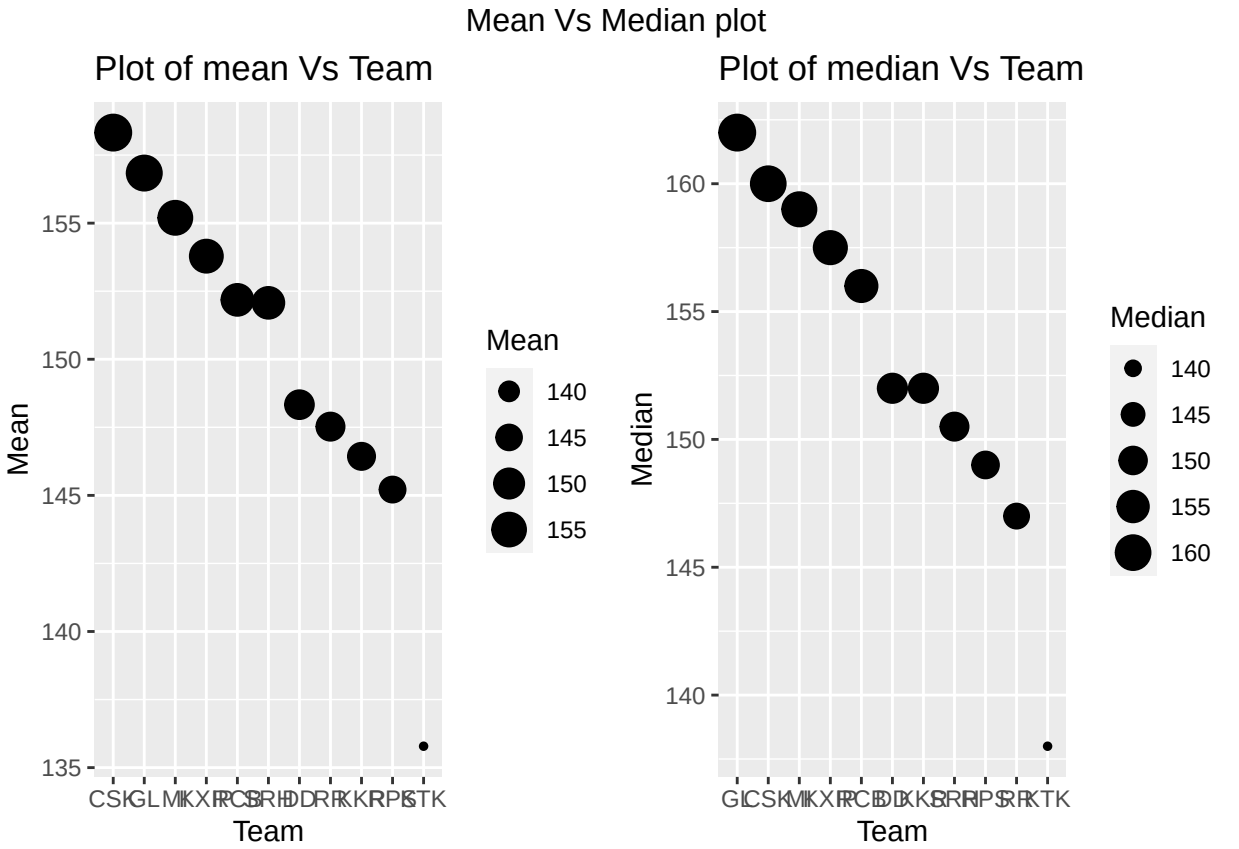
```
meanplot=a1%>%ggplot(aes(x=reorder(Team,-Mean),y=Mean,size=Mean))+geom_point()+labs(title="Plot of mean
meanplot
```




```
medianplot=a1%>%ggplot(aes(x=reorder(Team,-Median),y=Median,size=Median))+geom_point()+labs(title="Plot
medianplot
```




```
par(mfrow=c(1,2))
mvm=ggarrange(meanplot,medianplot)
annotate_figure(mvm,top="Mean Vs Median plot")
```



By seeing the above plots we see something interesting. If we observe closely, we see that the teams "RCB" and "SRH" have very similar(if not same) means but they have a very different and far apart medians.

```
a1[a1$Team=="RCB"|a1$Team=="SRH",]
```

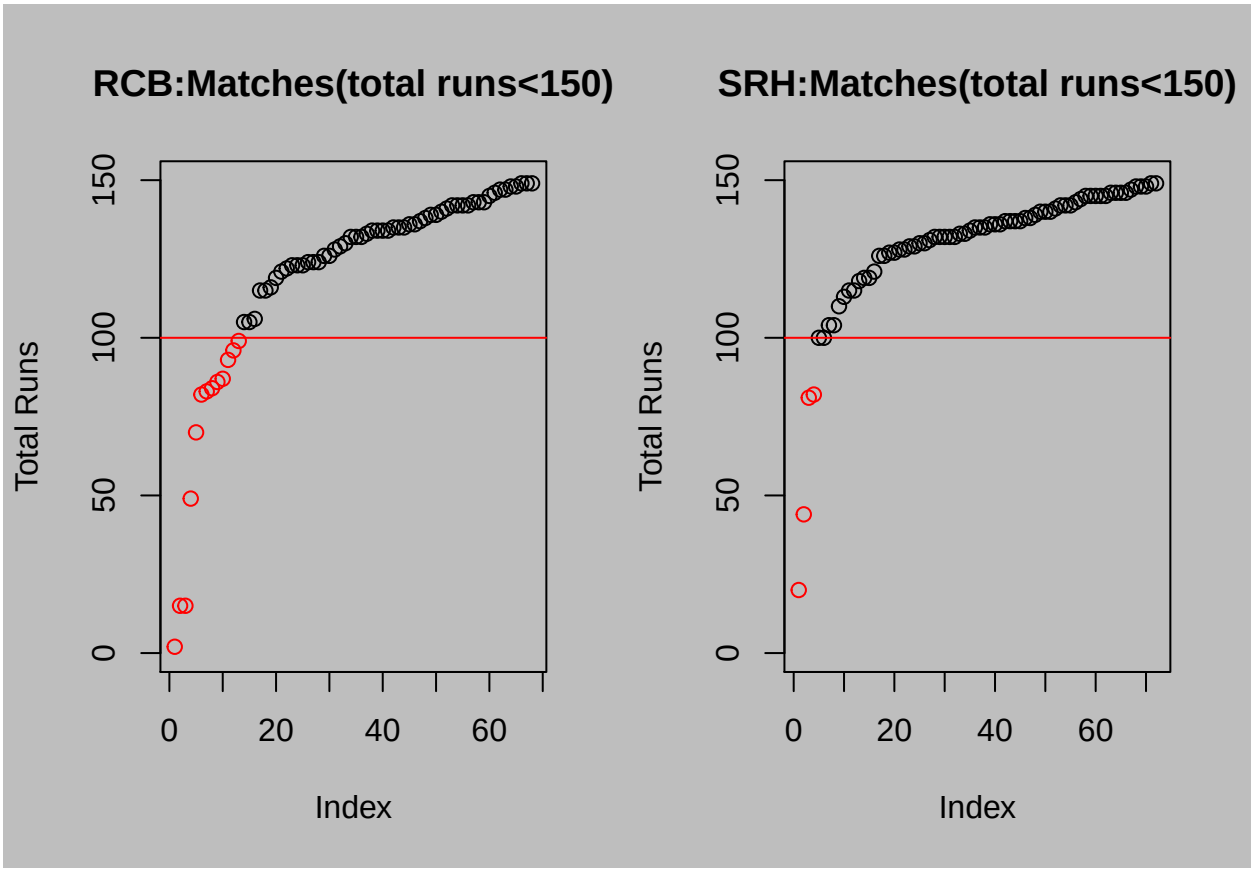
Team	Mean	Median	Matches_played
RCB	152	156	154
SRH	152	150	152

Why is that? We see that the median score for "RCB" is greater than that of "SRH" but the mean is almost the same. One possible explanation can be that RCB has scored very low scores in many matches in the league which pulled its mean to a lower value and on the other hand SRH is more consistent and maintained decent scores in all the matches

To check our explanation, let us now plot all the matches in which "RCB" and "SRH" have scored a total score below 150. By keeping a threshold of 100 runs,we will then compare how many matches were there such that the team had a total score below the threshold.

```
rcbplot=inning_runs%>%filter(Team=="RCB",Runs<150)%>%arrange(Runs)
srhplot=inning_runs%>%filter(Team=="SRH",Runs<150)%>%arrange(Runs)
```

```
par(mfrow=c(1,2),bg="grey")
plot(rcbplot$Runs,ylim =c(0,150),col=ifelse(rcbplot$Runs<100,"red","black")
,ylab="Total Runs",main="RCB:Matches(total runs<150)")
abline(h=100,col="red")
plot(srhplot$Runs,ylim =c(0,150),col=ifelse(srhplot$Runs<100,"red","black")
,ylab="Total Runs",main="SRH:Matches(total runs<150)")
abline(h=100,col="red")
```



We indeed see that our assumption was correct. We see in the above plots that the total number of matches where RCB has scored <100 is significantly more than that of SRH. Hence, the outliers in the lower end of the spectrum were more for RCB which reduced the mean score of the team inspite of having a larger median.
On the other hand we see that the median of the data is not affected by any outliers in the data.

Let's take another example.

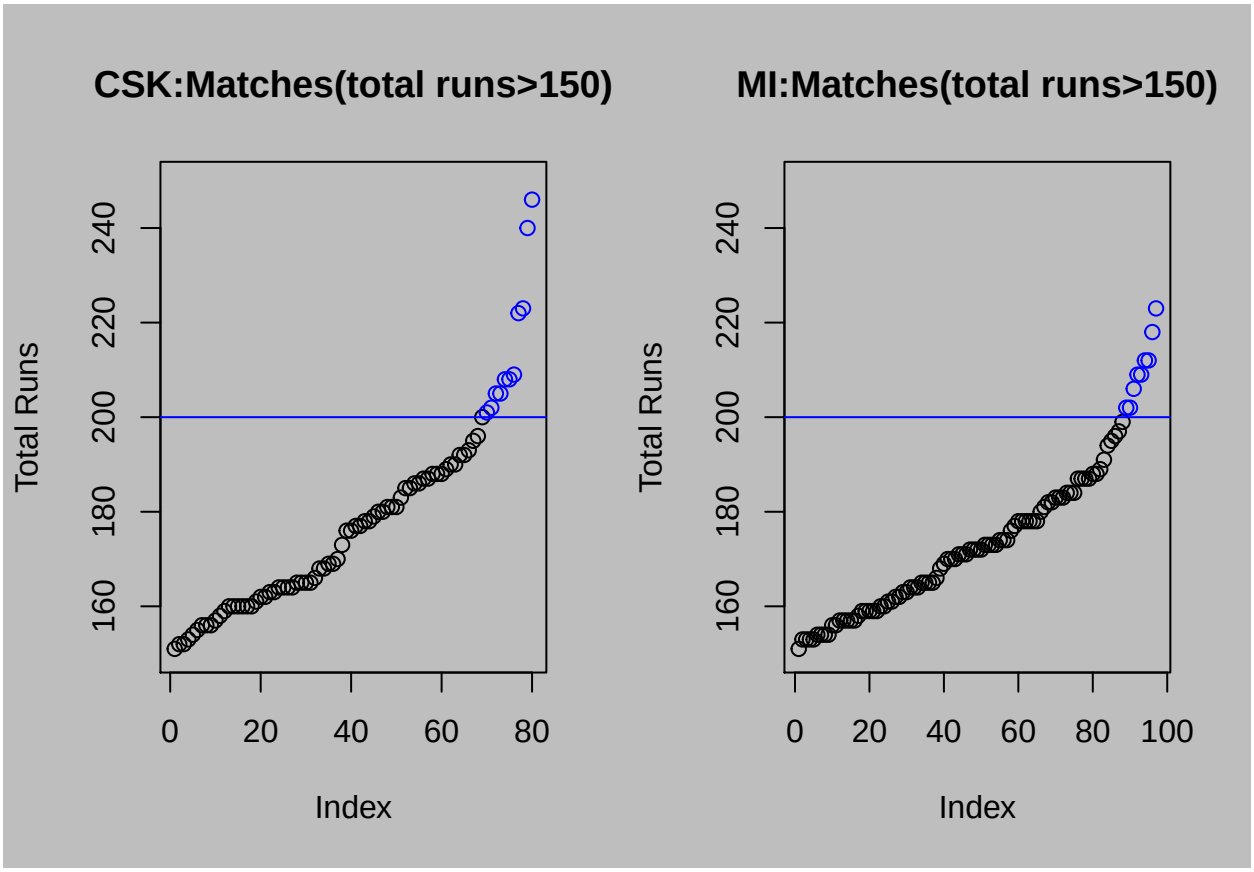
```
a1[a1$Team=="CSK"|a1$Team=="MI",]
```

Team	Mean	Median	Matches_played
CSK	158	160	132
MI	155	159	158

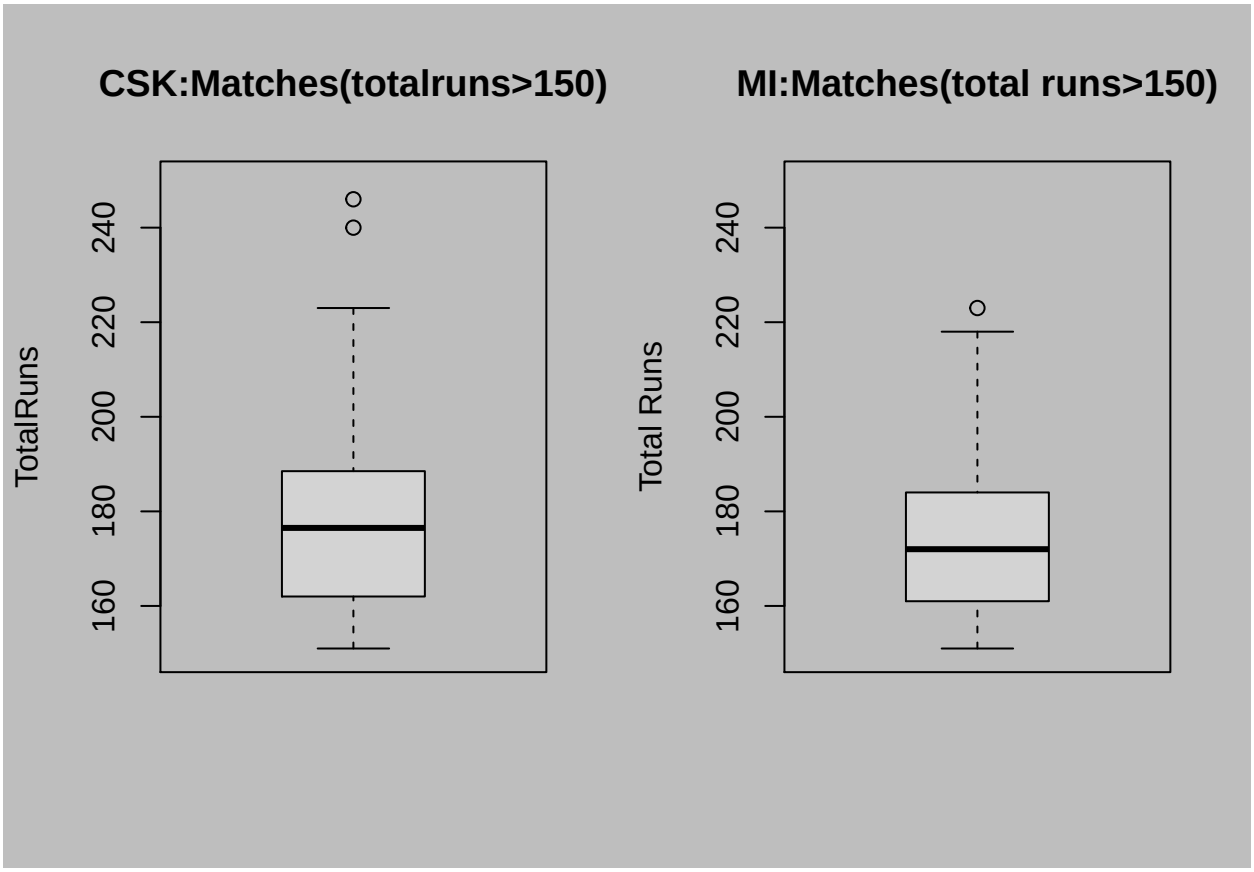
From the above table we see that "MI" and "CSK" have close median scores but a different mean score. This

example will be conclusive enough to show how outliers can greatly affect the mean To explain the above example we will plot the matchers where the total runs were on the higher side, lets say > 150. We will then compare the number of matches for each team where the total scores were above 200 runs.

```
cskplot=inning_runs%>%filter(Team=="CSK",Runs>150)%>%arrange(Runs)
miplot=inning_runs%>%filter(Team=="MI",Runs>150)%>%arrange(Runs)

par(mfrow=c(1,2),bg="grey")
plot(cskplot$Runs,ylim=c(150,250),col=ifelse(cskplot$Runs>200,"blue","black")
,ylab="Total Runs",main="CSK:Matches(total runs>150)")
abline(h=200,col="blue")
plot(miplot$Runs,ylim =c(150,250),col=ifelse(miplot$Runs>200,"blue","black")
,ylab="Total Runs",main="MI:Matches(total runs>150)")
abline(h=200,col="blue")
```




```
par(mfrow=c(1,2))
boxplot(cskplot$Runs,ylim=c(150,250),ylab="TotalRuns"
,main="CSK:Matches(totalruns>150)")
boxplot(miplot$Runs,ylim =c(150,250),ylab="Total Runs"
,main="MI:Matches(total runs>150)")
```



From the above plots we see that CSK has a couple of matches where it has scored extremely high(>230) while MI doesn't. These extremely large scores might be the reason for a greater mean score for the team "CSK".
Once again we saw that outliers in the data affected mean score but median remained unaffected.

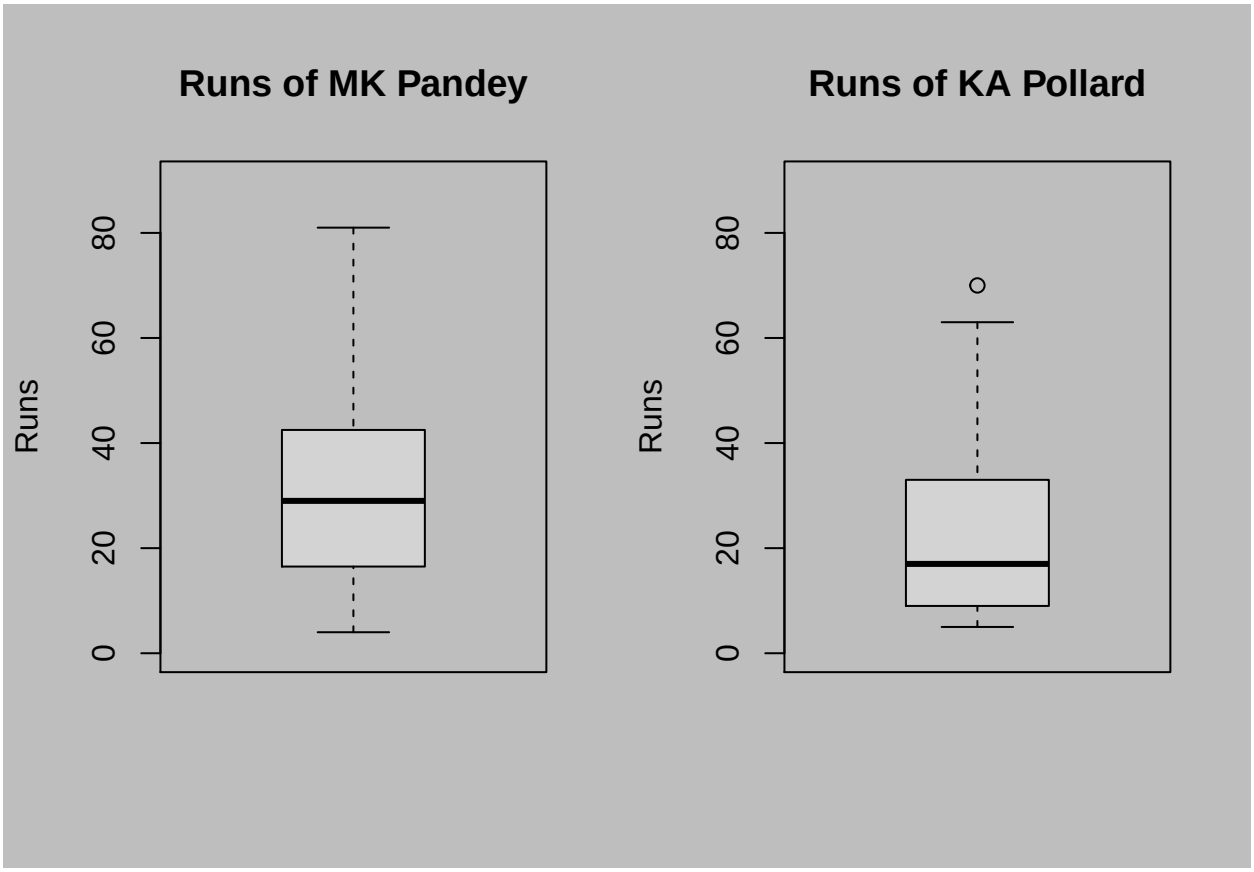
1.2 Comparing scores of players Now let us compare players based on their mean and median scores. We will be considering only the 2017 season for this.

```
playerruns=data.frame(expdat%>%filter(season==2017)%>%group_by(id,batsman)%>%
summarise(s=sum(batsman_runs)))%>%arrange(desc(s))
b1=data.frame(playerruns%>%group_by(batsman)%>%summarise(Total=sum(s),Mean=mean(s)
,Median=median(s)))%>%arrange(desc(Total))
b1[1:10,]
```

batsman	Total	Mean	Median
DA Warner	641	45.8	41.5
G Gambhir	498	31.1	20
S Dhawan	479	34.2	28.5
SPD Smith	472	31.5	27
SK Raina	442	31.6	30
HM Amla	420	42	26.5
MK Pandey	396	33	29
KA Pollard	395	24.7	17
PA Patel	395	24.7	24
RA Tripathi	391	27.9	29.5

We see from the above table that "MK Pandey" and "KA Pollard" have the sam total score but clearly the stats of "MK Pandey" is a lot higher. Let's see what might be the reason for this. Let us plot a boxplot of runs scored for each match for the players and see the differences.

```
pandey=playerruns%>%filter(batsman=="MK Pandey")%>%arrange(s)
pollard=playerruns%>%filter(batsman=="KA Pollard")%>%arrange(s)

par(mfrow=c(1,2),bg="grey")
boxplot(pandey$s,ylim=c(0,90),ylab="Runs",main="Runs of MK Pandey")
boxplot(pollard$s,ylim=c(0,90),ylab="Runs",main="Runs of KA Pollard")
```



We see that the runs distribution of MK Pandey is a lot wider than that of Pollard which explains why the median runs of Pandey is a lot higher. Pandey has also acheived many high scoring runs which explains why the mean is higher than that of Pollard.

CONCLUSION: After looking at all the above examples we can confidently conclude that extremely large or extremely small scores for a team/batsman can greatly affect the mean of the distribution. i.e mean is highly affected by outliers and thus can be misleading at times. On the other hand median is not affected by the outliers in the data and thus can be considered as a more reliable statistic.

2

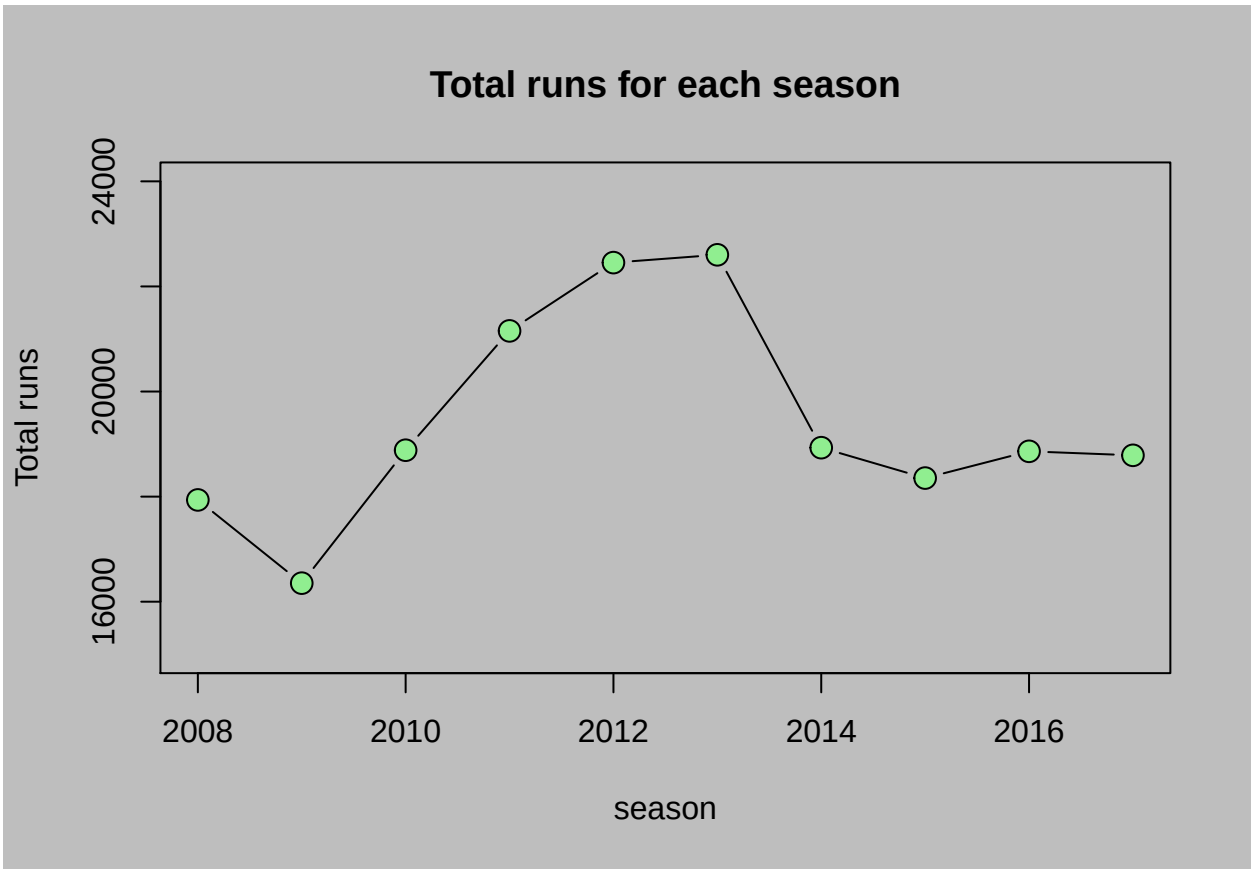
Let us now ask few interesting set of questions

2.1 Are total runs for the whole tournament increasing every year?
We can compute that with this fairly simple code.

```
total_runs=data.frame(expdat%>%group_by(season)%>%summarise(tot=sum(total_runs)))
total_runs
```

season	tot
2008	17937
2009	16353
2010	18883
2011	21154
2012	22453
2013	22602
2014	18931
2015	18353
2016	18862
2017	18786

```
par(mfrow=c(1,1),bg="grey")
plot(total_runs,col="black",type="b",ylab="Total runs"
,main="Total runs for each season",pch=21,bg="lightgreen",cex=1.5
,ylim=c(15000,24000))
```



We see that that there is a sharp increase in the total scores from 2011 to 2013. It is very unlikely are all players suddenly start scoring a lot more runs for few seasons and again fall back to normal.
To answer this question, let us see all the teams that participated in the league for each year.

```
data.frame(expdat%>%group_by(season)%>%summarise(No.of.Teams=length(unique(batting_team))))
```

season	No.of.Teams
2008	8
2009	8
2010	8
2011	10
2012	9
2013	9
2014	8
2015	8
2016	8
2017	8

We see that the number of teams that participated in the league increased in the years 2011,2012,2013. More teams directly means that more matches will be played which directly corelates to an increase in the total runs scored in the league. The list of teams can be seen for each season can be seen below.

```
t1=expdat%>%filter(season==2010)%>%select(y2010=batting_team)%>%distinct
t2=expdat%>%filter(season==2011)%>%select(y2011=batting_team)%>%distinct
t3=expdat%>%filter(season==2012)%>%select(y2012=batting_team)%>%distinct
t4=expdat%>%filter(season==2013)%>%select(y2013=batting_team)%>%distinct
t5=expdat%>%filter(season==2014)%>%select(y2014=batting_team)%>%distinct

df=merge(merge(merge(merge(t1,t2,by=0,all.y = T),t3,by="row.names",all=T),t4,by="row.names",all=T),t5,by
```

```
## Warning in merge.data.frame(merge(t1, t2, by = 0, all.y = T), t3, by =
## "row.names", : column name 'Row.names' is duplicated in the result
```

```
## Warning in merge.data.frame(merge(merge(t1, t2, by = 0, all.y = T), t3, : column
## names 'Row.names', 'Row.names' are duplicated in the result
```

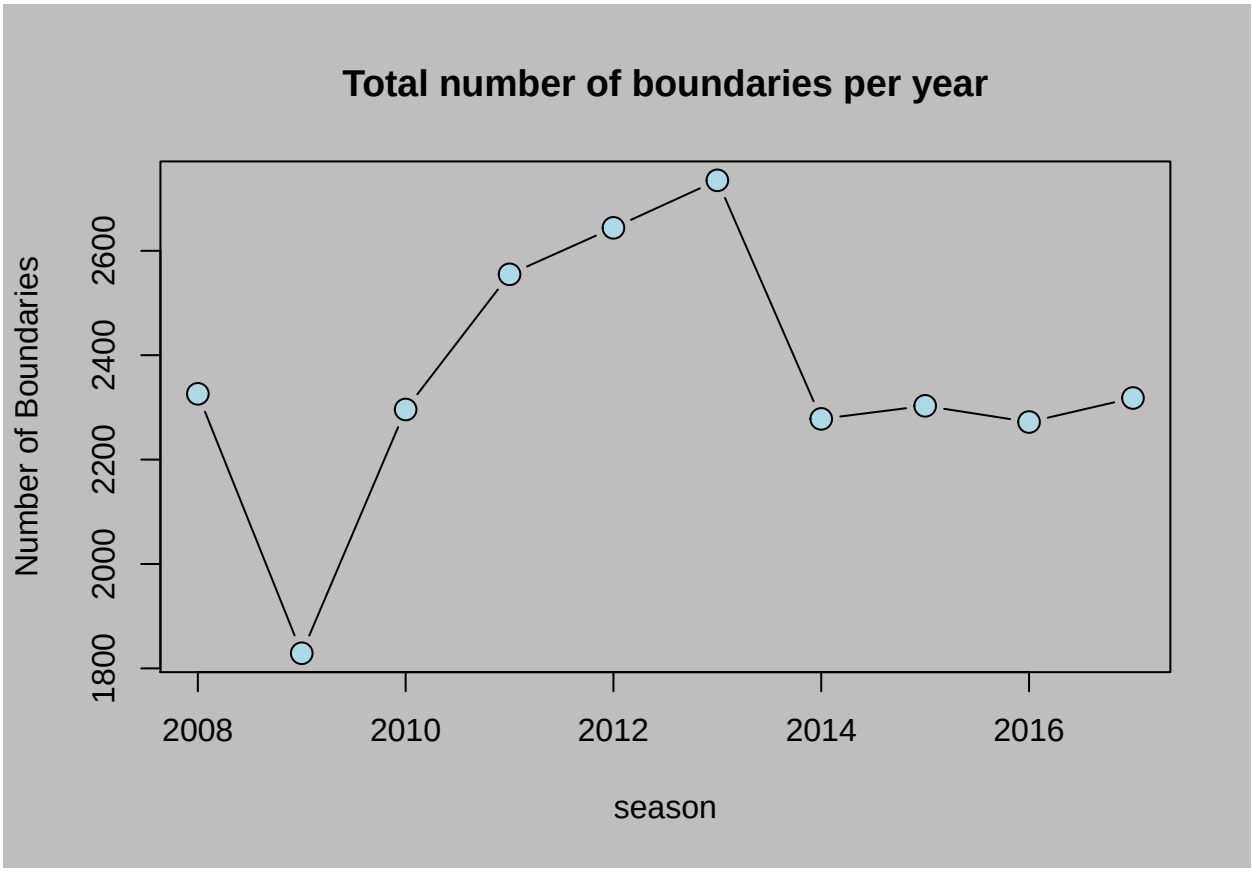
```
## Warning in merge.data.frame(merge(merge(merge(t1, t2, by = 0, all.y = T), :
## column names 'Row.names', 'Row.names', 'Row.names' are duplicated in the result
```

```
df=subset(df,select=-c(1,2,3,4))
df
```

y2010	y2011	y2012	y2013	y2014
KKR	CSK	CSK	DD	KKR
RCB	DD	RCB	KXIP	
CSK	MI	SRH		MI
	KXIP		KKR	DD
	RPS	MI	RCB	RCB
SRH	KKR	KKR	MI	CSK
MI	SRH	DD	SRH	KXIP
RR	RR	RPS	RPS	SRH
KXIP	KTK	RR	RR	RR
DD	RCB	KXIP	CSK	

2.2 Are total number boundaries increasing for every year?

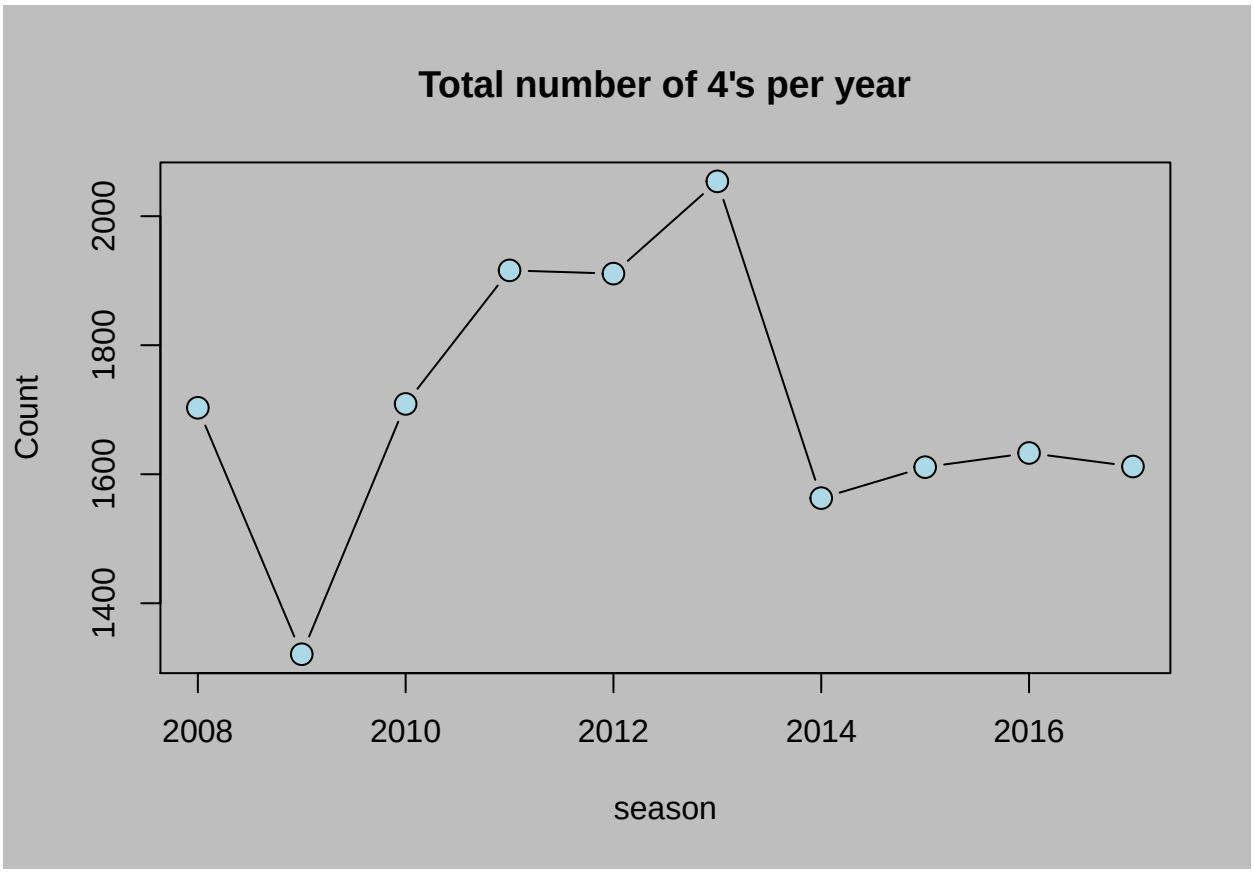
```
par(mfrow=c(1,1),bg="grey")
plot(expdat%>%group_by(season)%>%filter(batsman_runs==4|batsman_runs==6)%>%
summarise(bou=n()),ylab="Number of Boundaries"
,main="Total number of boundaries per year",type="b",pch=21,cex=1.5,bg="lightblue")
```



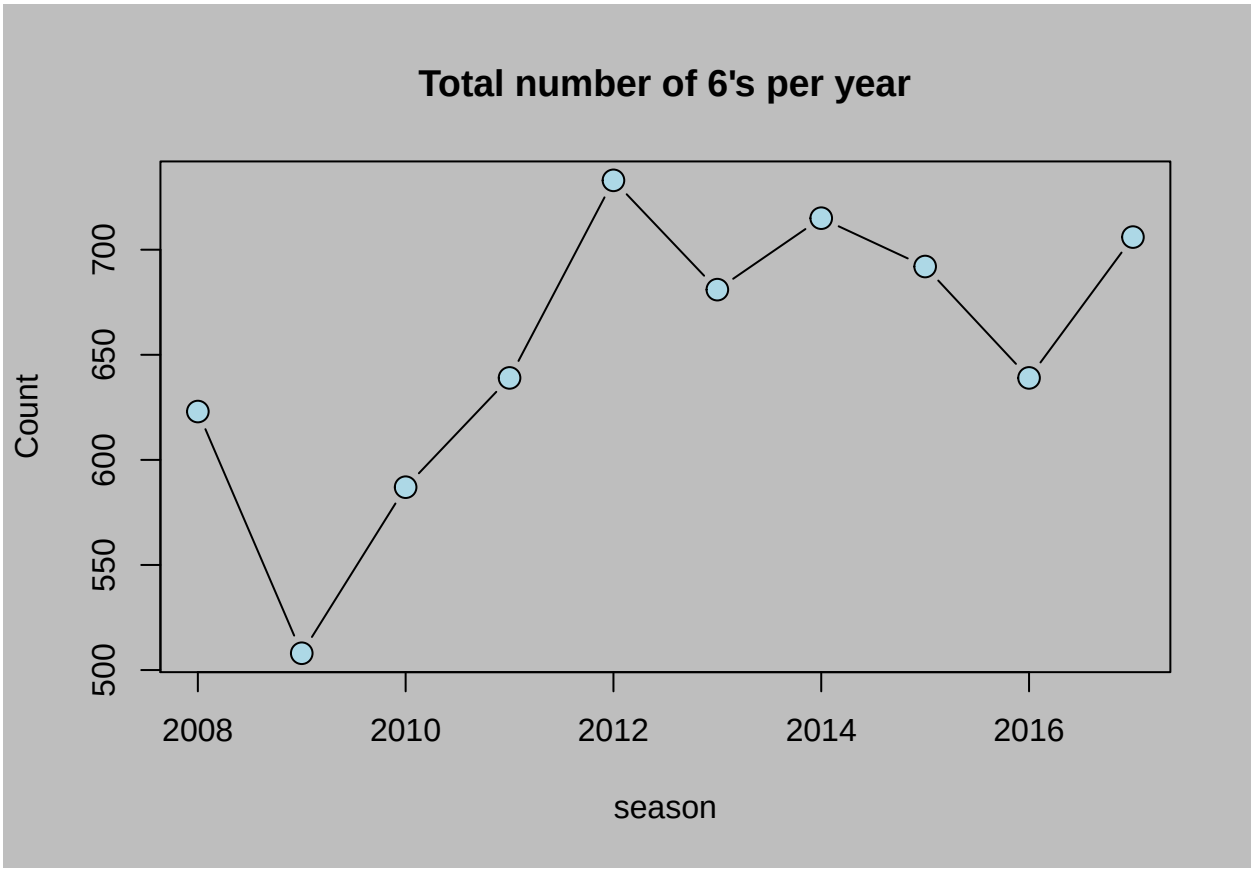
The number of boundaries gradually increased from 2010 to 2013 and then reduced in 2014. The reason for this is same as our previous question. The number of teams that participated were a lot more in the years 2011,2012,2013. This means more matches in the tourament and thus more boundaries.

Now let us compare the number of 4's and 6's individually

```
par(mfrow=c(1,1),bg="grey")
plot(expdat%>%group_by(season)%>%filter(batsman_runs==4)%>%summarise(bou=n())
,ylab="Count", main="Total number of 4's per year",type="b",pch=21,cex=1.5,bg="lightblue")
```




```
plot(expdat%>%group_by(season)%>%filter(batsman_runs==6)%>%summarise(bou=n())
,ylab="Count", main="Total number of 6's per year",type="b",pch=21,cex=1.5,bg="lightblue")
```



The trend of number of 4's per year is something we had aldredy expected. On the other hand we see something interesting going on with the number of sixes over the years. Initially due to more participation the number of sixes gradually increased but even after 2013 we see that it dosent change. The total number is still very high compared to the years before 2011.

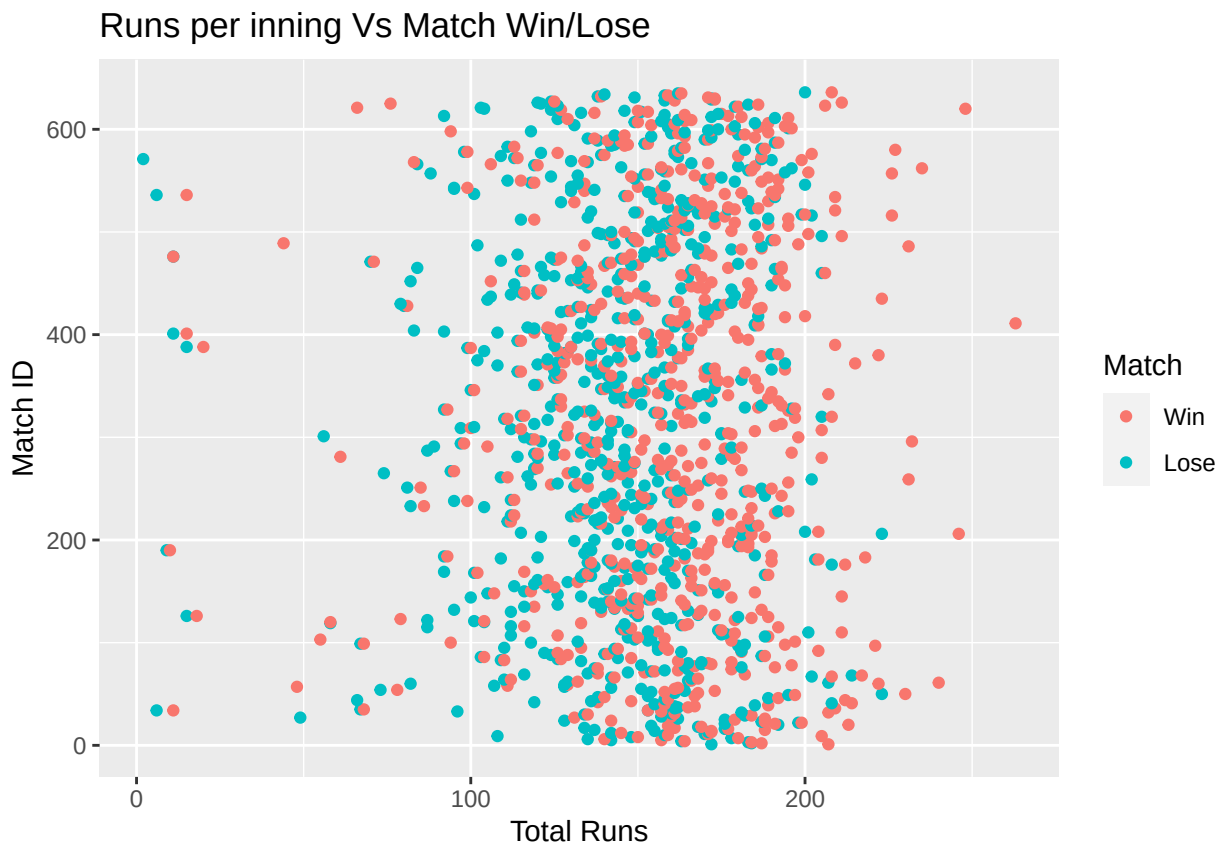
CONCLUSION: We can confidently conclude that the number of sixes over time have definetly increased and players are scoring more sixes than ever before

3 Corelation

In the next couple questions we will try to find out which variables have a direct affect on a team's victory or loss

3.1 The first thing that we can ask is.. does scoring more runs mean a higher chance of win?

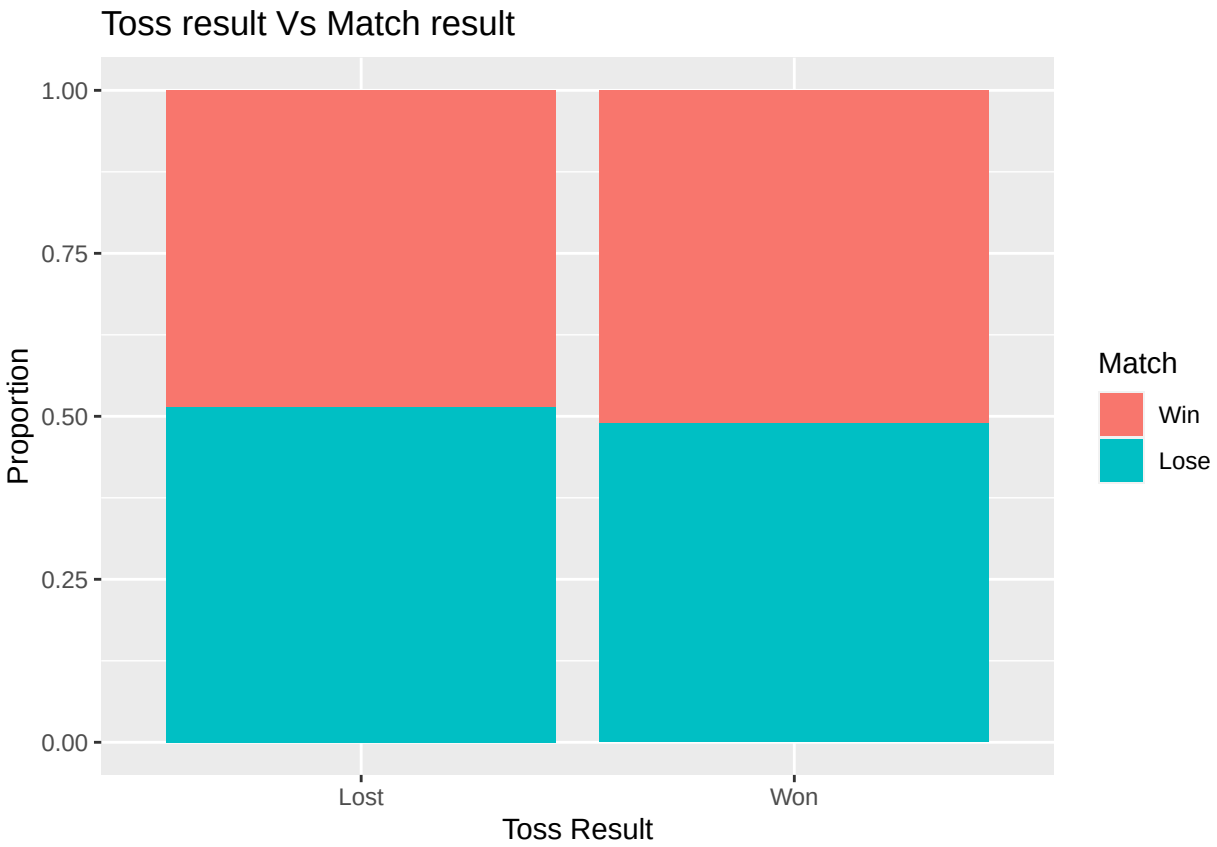
```
cor1=data.frame(expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team)
,s=sum(total_runs),Match=factor(ifelse(team==unique(winner),"Win","Lose"))
,xtr=sum(extra_runs)))

cor1%>%ggplot(aes(s,id,col=Match))+geom_point()+labs(x="Total Runs",y="Match ID"
,title="Runs per inning Vs Match Win/Lose")
```



The above graph is a plot of match id Vs the runs scored in the match. It also specifies the winning/losing score of the match
We see that there are more blue dots(corrosponding to lose) when the total score was on a lower side i.e around <150. But when the total score is >150 it is the total opposite. We see more red dots(corrosponding to win).
Hence, we can somewhat conclude that more score can increase a teams chances of win. The median score of all the observation turns out to be 154. So, a score above the median can be advantageous.

3.2 Another thing we can ask is.. how does a win/lose in toss affect a teams chances to win or lose the match.

```
toss=data.frame(expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team)
,s=sum(total_runs),tossdec=unique(toss_decision)
,Match=factor(ifelse(team==unique(winner),"Win","Lose"))
,tosswin=ifelse(unique(toss_winner)==unique(batting_team),"Won","Lost")))

toss%>%ggplot()+geom_bar(aes(factor(tosswin),fill=Match),position="fill")+
labs(x="Toss Result",y="Proportion",title="Toss result Vs Match result")
```



As we see there isnt much difference in the result of the match based on the result of the toss. Both toss win and toss lost have a very similar match result. Hence we can conclude that toss result does not play a crucial role in the result of the match

PART-2 : Sophmore Slump phenomena

According to wikipedia,"A sophomore slump or sophomore jinx or sophomore jitters refers to an instance in which a second, or sophomore, effort fails to live up to the relatively high standards of the first effort".
In short the phenomena says that, if the performance of a player in their sophmore or rookie year is very good, it is very unlikely that it will remain the same in the following year.
Firstly, the result this phenomena is claiming is very bold. There definetly has been atleast a single instance where we might have felt that a student/ player is performing extremely well for many years. But this claim contradicts our assumption.Hence, Let's us take a deeper dive into this
We will examine this phenomena by looking at the batting averages of the batsmen over a couple years in a row.

The rookie of the year is named as "emerging player of year" in the IPL. Since that information is not given in our dataset a quick google search will give us information about it. For this case, wetake the 2012 season. The rookie of the year for 2012 was "Mandeep Singh". We will look at his batting averages in 2012 and in 2013.

```
s=data.frame(expdat%>%filter(season==2012)%>%group_by(id,batsman)%>%
summarise(s=sum(batsman_runs),se=unique(season))%>%group_by(batsman)%>%
summarise(TotalRuns=sum(s),Mean=mean(s),Median=median(s),Season=unique(se))%>%
arrange(desc(TotalRuns)))
```

```
s1=data.frame(expdat%>%filter(season==2013)%>%group_by(id,batsman)%>%
summarise(s=sum(batsman_runs),se=unique(season))%>%group_by(batsman)%>%
summarise(TotalRuns=sum(s),Mean=mean(s),Median=median(s),Season=unique(se))%>%
arrange(desc(TotalRuns)))
rbind(s[s$batsman=="Mandeep Singh",],s1[s1$batsman=="Mandeep Singh",])
```

batsman	TotalRuns	Mean	Median	Season
Mandeep Singh	432	27	24	2012
Mandeep Singh	260	18.6	12.5	2013

We see something interesting here. Mandeep Singh was the "emerging player of year" for 2012 and we see good numbers above supporting that.But, in 2013 Mandeep's numbers decreased a lot. This result is in the favour of the phenomena we described!

Did we just get lucky? or is the phenomena really accurate.
We will now see all the players and not just the rookie of the year players for a broader picture. As usual we will see the batting averages of players. Lets see this for the year 2016 and 2017

```
allplayers=data.frame(expdat%>%filter(season==2016)%>%group_by(id,batsman)%>%
summarise(s=sum(batsman_runs),se=unique(season))%>%group_by(batsman)%>%
summarise(TotalRuns_2016=sum(s),Mean_2016=mean(s),Median_2016=median(s))%>%
arrange(desc(TotalRuns_2016)))

allplayers1=data.frame(expdat%>%filter(season==2017)%>%group_by(id,batsman)%>%
summarise(s=sum(batsman_runs),se=unique(season))%>%group_by(batsman)%>%
summarise(TotalRuns_2017=sum(s),Mean_2017=mean(s),Median_2017=median(s))%>%
arrange(desc(TotalRuns_2017)))
```

Lets see how the top 5 players of 2016 performed in 2017

```
top5=head(allplayers,5)
top2017=allplayers1[allplayers1$batsman%in%top5$batsman,]
merge(top5,top2017,by="batsman")
```

batsman	TotalRuns_2016	Mean_2016	Median_2016	TotalRuns_2017	Mean_2017	Median_2017
AB de Villiers	687	42.9	38	216	24	10
DA Warner	848	49.9	52	641	45.8	41.5
G Gambhir	501	33.4	34	498	31.1	20
S Dhawan	501	29.5	28	479	34.2	28.5
V Kohli	973	60.8	64.5	308	30.8	24

We see something interesting. Most players indeed underperform the following year of their success year. The sophmore slump phenomena is hence valid most of the times
Let us now look at something very different.. Let us see players who are underperforming in the year 2016 and compare it to 2017. Will their performance decrease even more? let's find out.

```
bottom_players=data.frame(expdat%>%filter(season==2016)%>%group_by(id,batsman)%>%
summarise(s=sum(batsman_runs),se=unique(season))%>%group_by(batsman)%>%
summarise(TotalRuns_2016=sum(s),Mean_2016=mean(s),Median_2016=median(s))%>%
arrange(desc(TotalRuns_2016)))

bottom_players1=data.frame(expdat%>%filter(season==2017)%>%group_by(id,batsman)%>%
summarise(s=sum(batsman_runs),se=unique(season))%>%group_by(batsman)%>%
summarise(TotalRuns_2017=sum(s),Mean_2017=mean(s),Median_2017=median(s))%>%
arrange(desc(TotalRuns_2017)))
```

The bottom most players will likely be the one who have played very few matches so it will not be useful to compare them, hence lets take players in the lower end of the spectrum, lets say from 50-55 rank

```
bot5=bottom_players[40:45,]
bot2017=bottom_players1[bottom_players1$batsman%in%bot5$batsman,]
merge(bot5,bot2017,by="batsman")
```

batsman	TotalRuns_2016	Mean_2016	Median_2016	TotalRuns_2017	Mean_2017	Median_2017
DA Miller	161	12.4	9	83	20.8	26
HM Amla	157	26.2	15	420	42	26.5
M Vohra	161	23	25	229	25.4	25
PA Patel	177	17.7	7.5	395	24.7	24
SE Marsh	159	26.5	26.5	264	33	25.5
SS Tiwary	170	24.3	17	52	52	52

We see something very different than what we had previously assumed. The players who are not performing very well seem to improve in the following year. So is this the reverse sophmore slump effect???

Turns out NO!. This kind of effect is fairly common in the real world. Turns out these values are just regressing to the mean or this can be called as the "regression to the mean" effect. These values are just approaching the mean of population and thus things that perform extremely good in one attempt tend to perform a bit poorly in the next attempt and vice versa.

PART-3 : Predicting the chance of an event

In the next set of questions we will take up an event and try to find out how likely does that particular event occur.

1

What is the chance that a team will will all the matches in a season?
We will predict this value for the 2017 season by using the data of all the previous seasons.
Now, something like this can be useful for the teams. Every team strives to win as many matches as possible and potentially win the tournament. By knowing these win-probabilities one can vaguely assume who can be a possible contender for the title.

We can compute these values with the help of a binomial distribution. The binomial formulae is " $P(X = x) = n_{C_x}.p^x.(1-p)^{n-x}$ ". Here, n is the maximum number of matches that can be played by a team, p is the probability of win for a team, x is the number of success.

```
binom1dat=expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team),s=sum(total_runs),w=ifelse
```

By a quick look at the official website, we see that the maximum matches per team is 14. Hence our $n = 14$ Since, we want to find the chance of winning all matches,our $x = 14$.Now, We just need to calculate p(probability of win)
The p value and total wins for each team is given below.

```
bi1=data.frame(expdat%>%group_by(id,inning)%>%summarise(Team=unique(batting_team)
,w=ifelse(Team==unique(winner),1,0))%>%select(Team,w))%>%select(Team,w)
```

```
## Adding missing grouping variables: `id`
```

```
bi2=data.frame(bi1%>%group_by(Team)%>%summarise(Total_wins=sum(w)
,Win_Probability_p=mean(w)))%>%arrange(desc(Total_wins))
bi2
```

Team	Total_wins	Win_Probability_p
MI	93	0.589
CSK	79	0.598
KKR	77	0.513
RCB	74	0.481
KXIP	72	0.48
SRH	72	0.474
RR	65	0.542
DD	62	0.419
RPS	27	0.36
GL	13	0.419
KTK	6	0.429

We now have the values of p for all teams. p value is nothing but the mean of wins for the team till the year 2016.
We have all the values so let's plug in the values and find the probabilities." $P(X = 14) = 14_{C14}.p^{14}.(1-p)^{14-14}$ " Let us call the required probability as "P" The probablity of winning all the matches for each team is given

```
bi2$P=dbinom(14,size=14,prob=bi2$Win_Probability_p)
bi2=bi2%>%arrange(desc(P))
colnames(bi2)=c("Team","Total_wins","p",' $\\mathrm{P}(X = 14)$ ')
bi2
```

Team	Total_wins	p	$\mathrm{P}(X = 14)$
CSK	79	0.598	0.000756
MI	93	0.589	0.000599
RR	65	0.542	0.000187
KKR	77	0.513	8.82e-05
RCB	74	0.481	3.5e-05
KXIP	72	0.48	3.45e-05
SRH	72	0.474	2.86e-05
KTK	6	0.429	7.05e-06
GL	13	0.419	5.2e-06
DD	62	0.419	5.13e-06
RPS	27	0.36	6.14e-07

As seen above, the chance that a team will win all matches is extremely low. But we see compared to others CSK and MI have a higher chance to win all matches. A person who follows cricket will know that CSK and MI had won the most titles. Maybe this explains why they were so successful.

2

Computing the chance of winning all the matches was maybe a bit impractical. Lets ask a question which is more realistic.
What is the probability that a team will quailfy for the qualifiers.?
Qualifiers are nothing but the top 4 teams which are selected after the end of all regular matches which continue to play the final set of matches called the "plauoffs". Since our dataset does'nt mention which matches were regular ones and which were the playoffs matches, a quick look into the website for the "minimum number of wins required per season in order to qualify for the qualifiers" will reveal that the median number of wins is around 7.
Like the previous question this also can be answered with the help pf binomial distribution. The difference is just that our x changes to $x > 7$.

Since we aldredy know all the other values,our equation is now " $P(X >= 7) = P(X = 7)+P(X = 8)+P(X = 9) + P(X = 10) + P(X = 11) + P(X = 12) + P(X = 13) + P(X = 14)$ " (This probability is being computed for the year 2017)

```
bi3=data.frame(bi2%>%mutate(P=dbinom(7,size=14,prob=p)+dbinom(8,size=14,prob=p)+
dbinom(9,size=14,prob=p)+dbinom(10,size=14,prob=p)+dbinom(11,size=14,prob=p)+
dbinom(12,size=14,prob=p)+dbinom(13,size=14,prob=p)+dbinom(14,size=14,prob=p))%>%select(Team,Total_wins
colnames(bi3)=c("Team","Total_wins","p","$\\mathrm{P}(X > 7)$")
bi3
```

Team	Total_wins	p	$\mathrm{P}(X > 7)$
CSK	79	0.598	0.847
MI	93	0.589	0.828
RR	65	0.542	0.72
KKR	77	0.513	0.643
RCB	74	0.481	0.547
KXIP	72	0.48	0.545
SRH	72	0.474	0.526
KTK	6	0.429	0.389
GL	13	0.419	0.362
DD	62	0.419	0.361
RPS	27	0.36	0.206

We again see that “CSK” and “MI” have a greater chance of qualifying and getting into playoffs.

3

Our question now is to predict what is the probability that a team will win given that the opponent has a total score greater than the median score.
Our first step is to find out the median total score of our data

```
data.frame(expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team)
,s=sum(total_runs)))%>%summarise(median=median(s))
```

median
154

The median score is 154.Now how do we solve this question?
We can use to the baye’s theorem to solve this. The baye’s formulae is as follows..$P(A \mid B) = \frac{P(B|A).P(A)}{P(B)}$
For this question we will compute the porbability for just a single team.Lets say its MI.

Our question now is $P(A \mid B)$
Here our event $A = Winning\ a\ match$
event $B = Opponent\ has\ a\ total\ score > 154$

$P(B \mid A) = Prob.\ of\ opponentscoring > 154\ given\ that\ our\ teamhad\ won\ the\ match$
This is computed below

```
test1=data.frame(expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team)
,w=ifelse(team==unique(winner),1,0),t1=unique(team1),t2=unique(team2)
,win=unique(winner),s=sum(total_runs)))%>%group_by(id)%>%
filter(t1=="MI"||t2=="MI",win=="MI")%>%select(id,team,inning,w,s)
ab154gw=data.frame(test1%>%filter(team!="MI"))%>%summarise(w=mean(s>154))
```

$P(A) = Prob.\ of\ our\ team\ winning$
This is computed below

```
wp=data.frame(expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team)
,w=ifelse(team==unique(winner),1,0),s=sum(total_runs)))%>%filter(team=="MI")%>%
summarise(pow=mean(w==1))
```

$P(B) = Prob.\ of\ opponent\ scoring > 154$
This is computed below

```
test=data.frame(expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team)
,w=ifelse(team==unique(winner),1,0),t1=unique(team1),t2=unique(team2)
,win=unique(winner),s=sum(total_runs)))%>%group_by(id)%>%filter(t1=="MI"||t2=="MI")%>%
select(id,team,inning,w,s)
ab154=data.frame(test%>%filter(team!="MI"))%>%summarise(pr=mean(s>154))
```

We need to calculate $P(A \mid B)$
Our required answer $P(A \mid B)$ is

```
answer=(ab154gw*wp)/ab154
answer
```

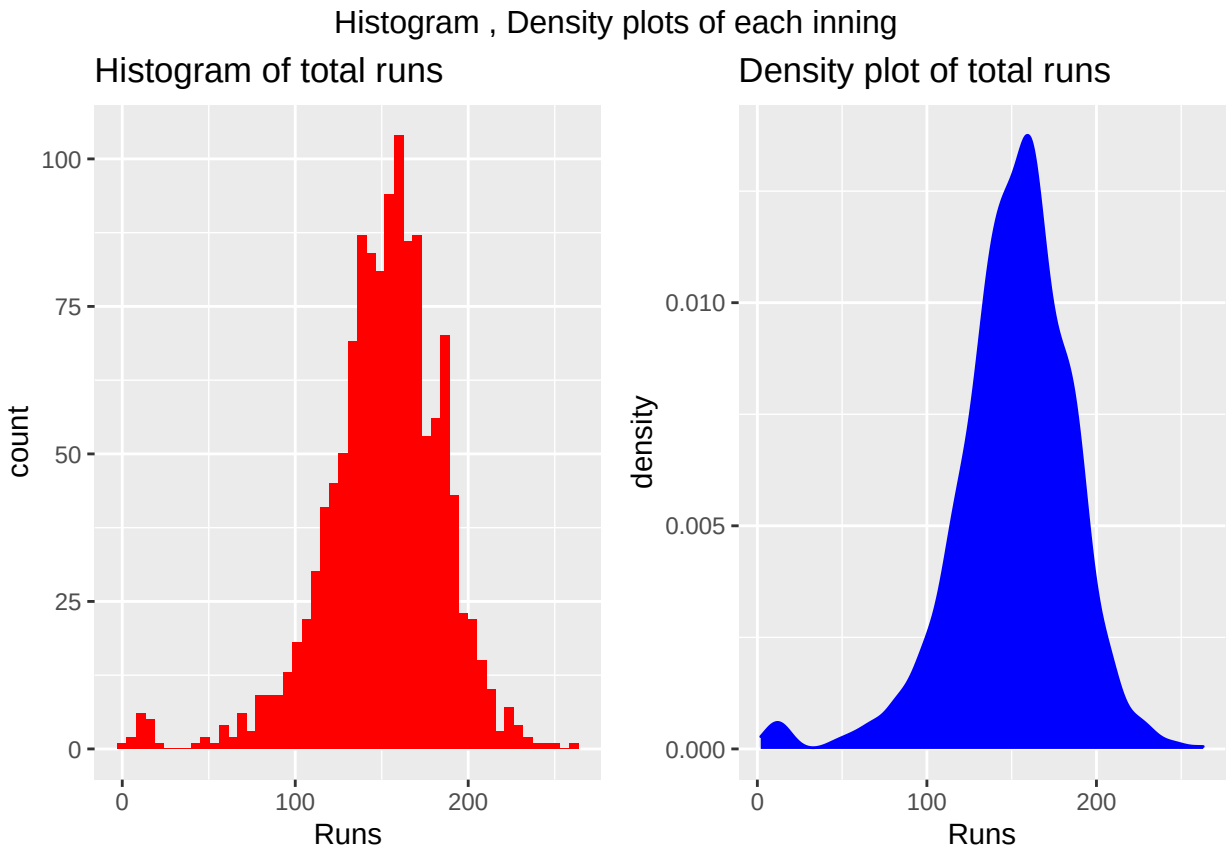
w
0.506

We see that MI wins more than 50% of times given that the opponent has scored more than 154 runs which is the median score.

4

In this question we will take a very different approach in solving the question. Our main question would be.. "What is the probability of a team scoring above a certain score"
We could just solve this by directly seeing all the total scores and computing it. But we will take a different approach
Below graph is a histogram and density plot of total scores of all innings

```
par(mfrow=c(1,1))
in1=inning_runs%>%ggplot(aes(Runs))+geom_histogram(fill="red",bins=50)+
labs(title="Histogram of total runs")
in2=inning_runs%>%ggplot(aes(Runs))+geom_density(color="blue",fill="blue")+
labs(title="Density plot of total runs")
fi=ggarrange(in1,in2)
annotate_figure(fi,top="Histogram , Density plots of each inning")
```



We see that these are very close to bell curves. These are not perfectly normal but let us see how close is it to a normal distribution

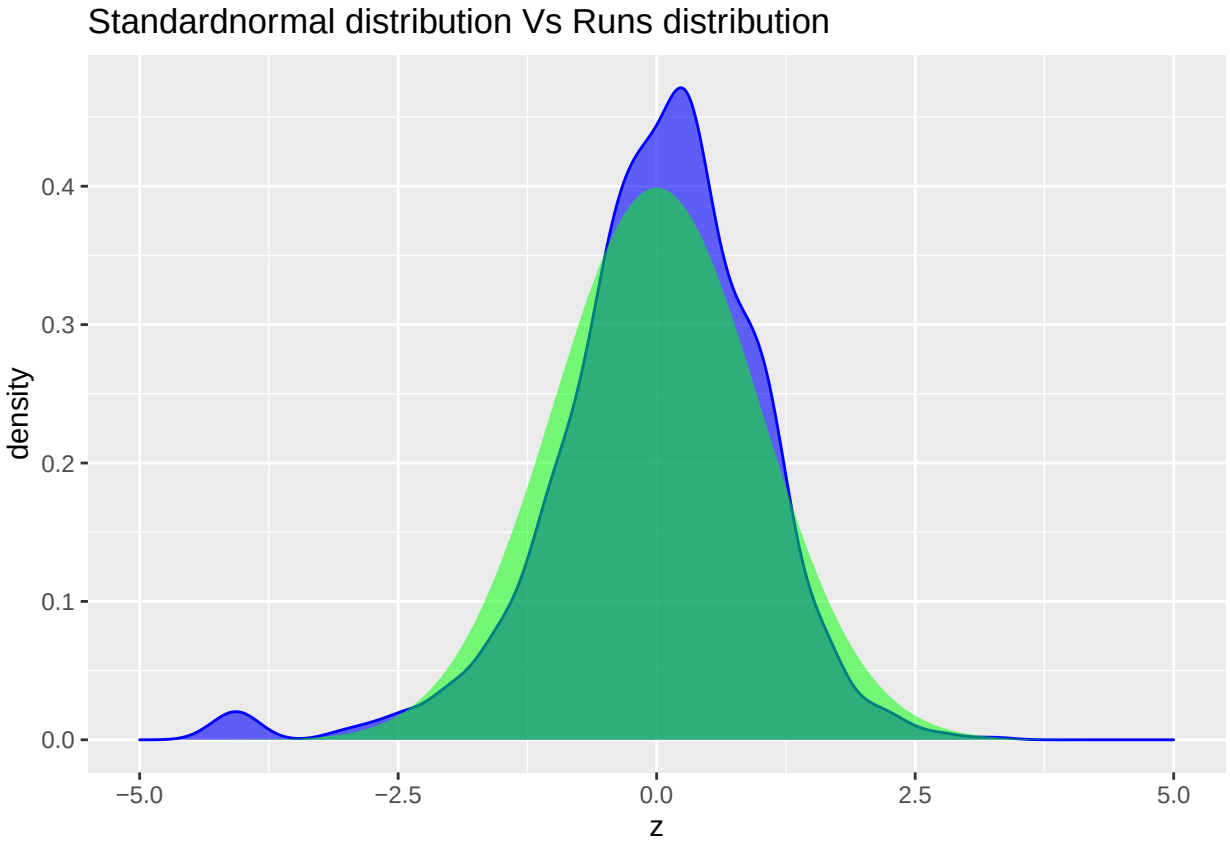
First let us convert our data into a z variable so that we can compare easily with a standard normal distribution. z is defined as $z = \frac{X_i - mean(X)}{sd(X)}$ The mean and standard deviation of our distribution are

```
inning_runs%>%summarise(Mean=mean(Runs),Sd=sd(Runs))
```

Mean	Sd
151	34.3

```
z_runs=inning_runs%>%mutate(z=(Runs-151.3349)/34.26773)%>%select(z)
range=seq(-5,5,0.01)
std_nrml=data.frame(z=dnorm(range,0,1))
zplot=ggplot(data=z_runs,aes(z))+geom_density(fill="blue",color="blue",alpha=0.6)+
```

```
geom_area(data=std_nrml,aes(range,z),fill="green",alpha=0.5)+
labs(title="Standardnormal distribution Vs Runs distribution")
zplot
```



The green distribution is the standard normal whereas the blue is the distribution of runs from our data. They are not same but they are very similar.
We will now compute our required problem with the stadard normal dist. and see how accurate is the standard normal dist. to our original runs distribution

Chance of scoring above (we will convert the runs into its respective z score and refer a z table)

1)the median score i.e 154 is

```
data.frame(calculated=1-pnorm((154-151.3349)/34.26773,mean=0,sd=1)
,actual=mean(inning_runs$Runs>154))
```

calculated	actual
0.469	0.494

2) 180 runs is

```
data.frame(calculated=1-pnorm((180-151.3349)/34.26773,mean=0,sd=1)
,actual=mean(inning_runs$Runs>180))
```

calculated	actual
0.201	0.185

3) 100 runs is

```
data.frame(calculated=1-pnorm((100-151.3349)/34.26773,mean=0,sd=1)
,actual=mean(inning_runs$Runs>100))
```

calculated	actual
0.933	0.936

In all the above cases,the calculated value is very close to the actual answer.

Conclusion: The distribution of total runs of each inning is very close to a normal distribution. A normal distribution with appropriate mean and varience gives us a close approximation when areas under the curve are calculated.

5

In part 1 we saw that the total score is related to win/lose for the match. The higher the total score, the higher the chances of winning a match.
Let us consider the median of the total scores i.e 154 as a threshold and calculate the probability of win if a team scores above 154.

```
clx=data.frame(expdat%>%group_by(id,inning)%>%summarise(team=unique(batting_team)
,s=sum(total_runs),Match=factor(ifelse(team==unique(winner),1,0))
,xtr=sum(extra_runs)))
mean(clx[clx$s>154,]$Match==1)
```

```
## [1] 0.6056782
```

The above states that, if a team scores a total score >154 it has a 60% chance of winning.
Considering that a 60% win probability is a good number, let us now assume that a total score >154 or runs per over >7.7 is considered as "good". Our question now is, For an individual team, if 6 balls(1 over) are chosen at random, what is the probability that the sum of runs of 6 balls is >7.7 runs.

We answer this question by first seeing the distribution of sum of runs of 6 balls chosen at random. We can plot this distribution with the help of Central Limit Theorem. We will see two examples here i.e plot the distribution for two individual teams.(CSK and DD)

CSK plot

```
clx1=data.frame(expdat%>%filter(batting_team=="CSK"))
```

Mean and standard deviation of runs(per ball) for CSK

```
clx1%>%summarise(Mean=mean(total_runs),Sd=sd(total_runs))
```

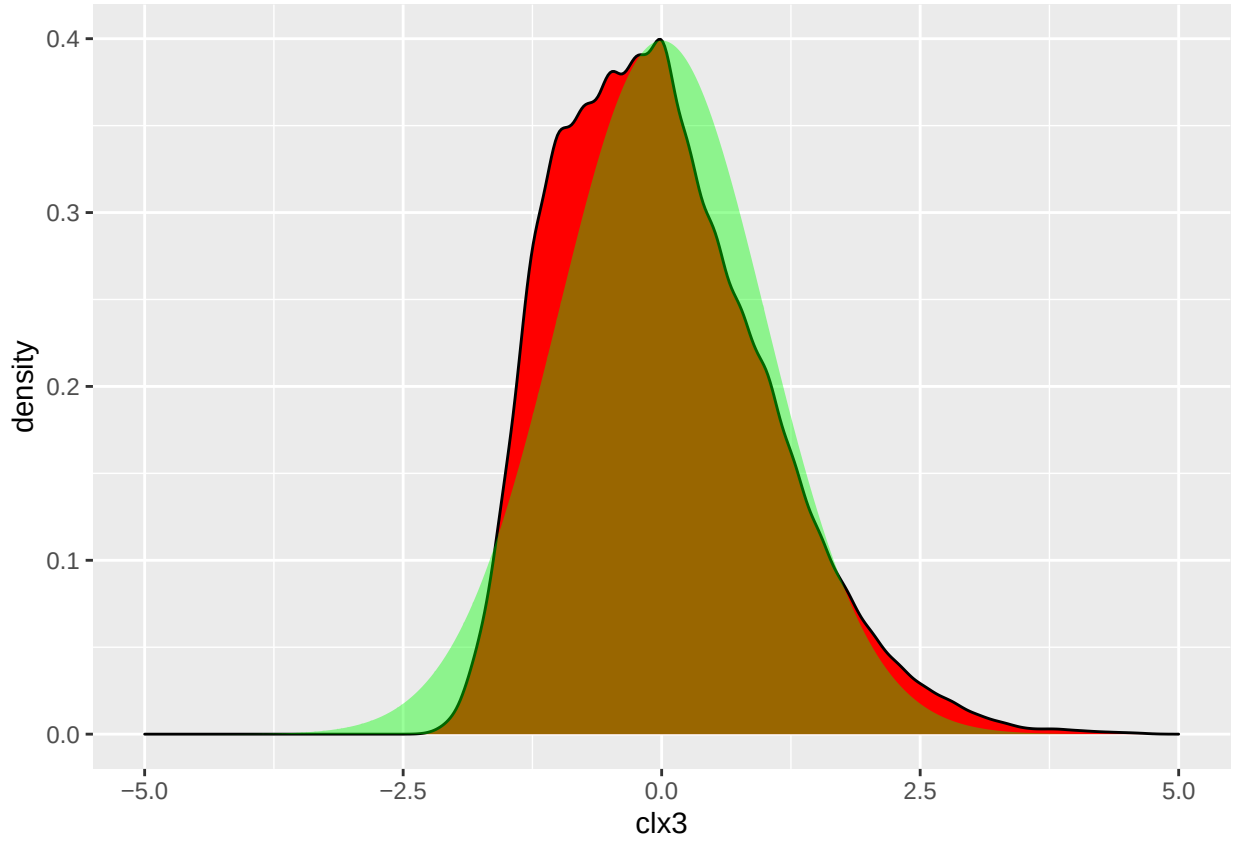
Mean	Sd
1.33	1.6

We will first take a sample of 6 balls at random and calculate the sum of it. We will then perform the same process a large number of times with the help of a Monte Carlo simulation and plot it.

```
clx2=replicate(10000,{
  po=sample(clx1$total_runs,6)
  sum(po)
})

clx3=data.frame(clx3=(clx2-mean(clx2))/sd(clx2))
```

The below is a plot of our desired distribution(distribution of sum of runs of 6balls)(converted into z scores) overlapped with a standard normal dist.

```
range1=seq(-5,5,0.01)
std_nrml1=data.frame(clx3=dnorm(range,0,1))
clx3%>%ggplot(aes(clx3))+geom_density(fill="red",alpha=1)+
geom_area(data=std_nrml1,aes(range1,clx3),fill="green",alpha=0.4)
```



As the CLT states, our distribution is very similar to a standard normal distruibution. We will now calculate the probability of scoring a total run greater than 7.7 in both the ways namely, the approximation of a standard normal and the original distribution.(The z score of 7.7 is given below)
The z score of 7.7 is

```
z7=(7.7-mean(clx2))/sd(clx2)
z7
```

```
## [1] -0.06860118
```

```
data.frame(z_prob=1-pnorm(z7,0,1),original_prob=mean(clx3$clx3>z7))
```

z_prob	original_prob
0.527	0.506

The standard normal dist. is close to the value of the original dist.
More importantly, there is a 50% chance that the sum of runs is greater than 7.7. Which can be considered a good number.

DD plot

```
cldd1=data.frame(expdat%>%filter(batting_team=="DD"))
```

Mean and standard deviation of runs(per ball) for DD

```
cldd1%>%summarise(Mean=mean(total_runs),Sd=sd(total_runs))
```

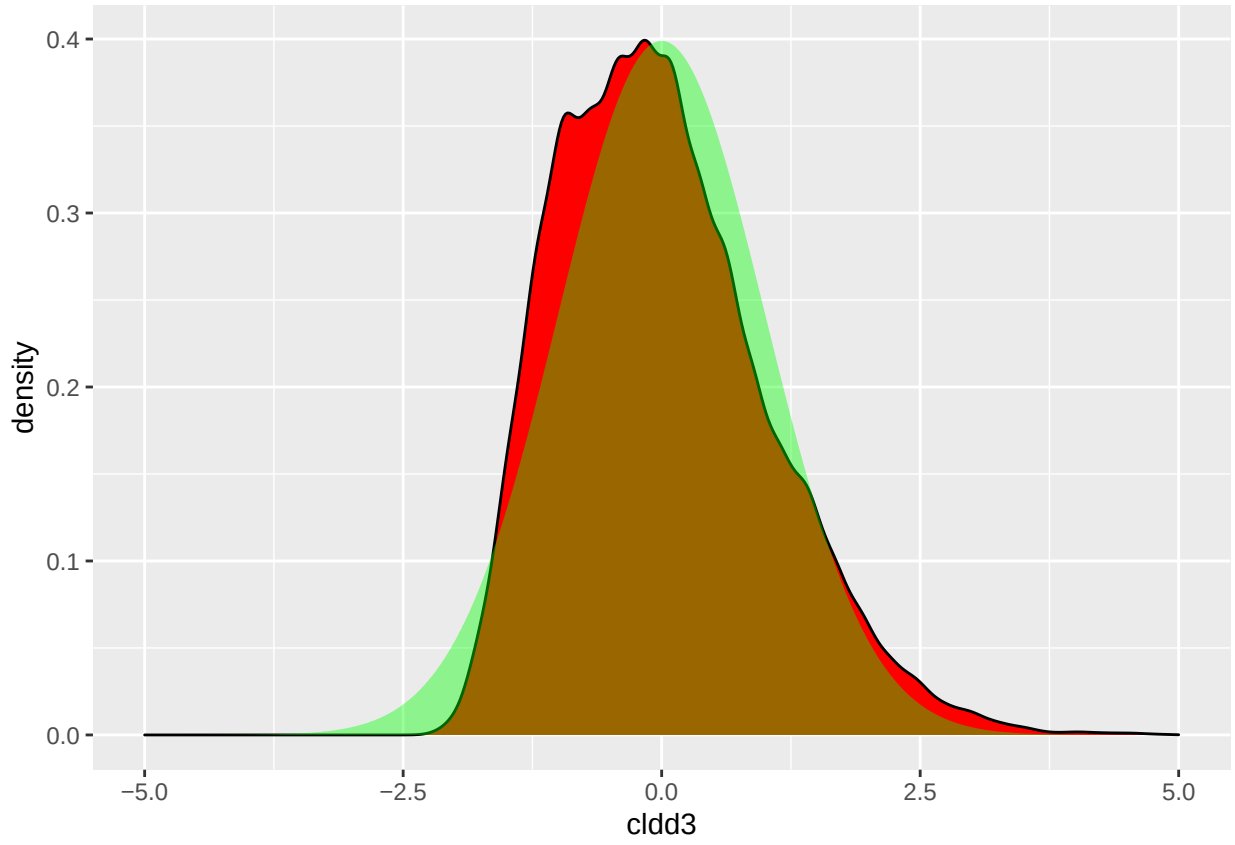
Mean	Sd
1.28	1.56

We will first take a sample of 6 balls at random and calculate the sum of it. We will then perform the same process a large number of times with the help of a Monte Carlo simulation and plot it.

```
cldd2=replicate(10000,{
  po=sample(cldd1$total_runs,6)
  sum(po)
})

cldd3=data.frame(cldd3=(cldd2-mean(cldd2))/sd(cldd2))
```

The below is a plot of our desired distribution(converted into z scores) overlapped with a standard normal dist.

```
range2=seq(-5,5,0.01)
std_nrml2=data.frame(cldd3=dnorm(range,0,1))
cldd3%>%ggplot(aes(cldd3))+geom_density(fill="red",alpha=1)+
geom_area(data=std_nrml2,aes(range2,cldd3),fill="green",alpha=0.4)
```



As the CLT states, our distribution is very similar to a standard normal distruibution. We will now calculate the probability of scoring a total run greater than 7.7 in both the ways namely, the approximation of a standard normal and the original distribution.(The z score of 7.7 is given below)
The z score of 7.7 is

```
zdd7=(7.7-mean(cldd2))/sd(cldd2)
zdd7
```

```
## [1] 0.01771344
```

```
data.frame(z_prob=1-pnorm(zdd7,0,1),original_prob=mean(cldd3$cldd3>zdd7))
```

z_prob	original_prob
0.493	0.471

The standard normal dist. is close to the value of the original dist. The probability is slightly lesser in this case which is close to 47%.

Lets compare both.(We will consider the z approximation value)

```
data.frame(CSK_prob=1-pnorm(z7,0,1),DD_prob=1-pnorm(zdd7,0,1))
```

CSK_prob	DD_prob
0.527	0.493

CONCLUSION: We had taken CSK and DD as examples because, if we see the overall statistics, CSK is towards the top of the list while DD is on a relatively lower side. And with the above example we can see why. The probability that CSK scores more runs per over is more than that of DD. This directly implies that CSK has scored more runs and thus won more matches.

6

In short, in this question, we will try to find out the interval in which population mean(μ)(of the total runs scored in a match) lies with the help of a given set sample observations. We will consider the matches played in 2017 season as our sample. And for individual team, we will estimate its population mean (total score in a match). We will calculate this in two ways

1)Known population varience(uses z dist.)
2)Unknown population varience(uses student's t dist.)

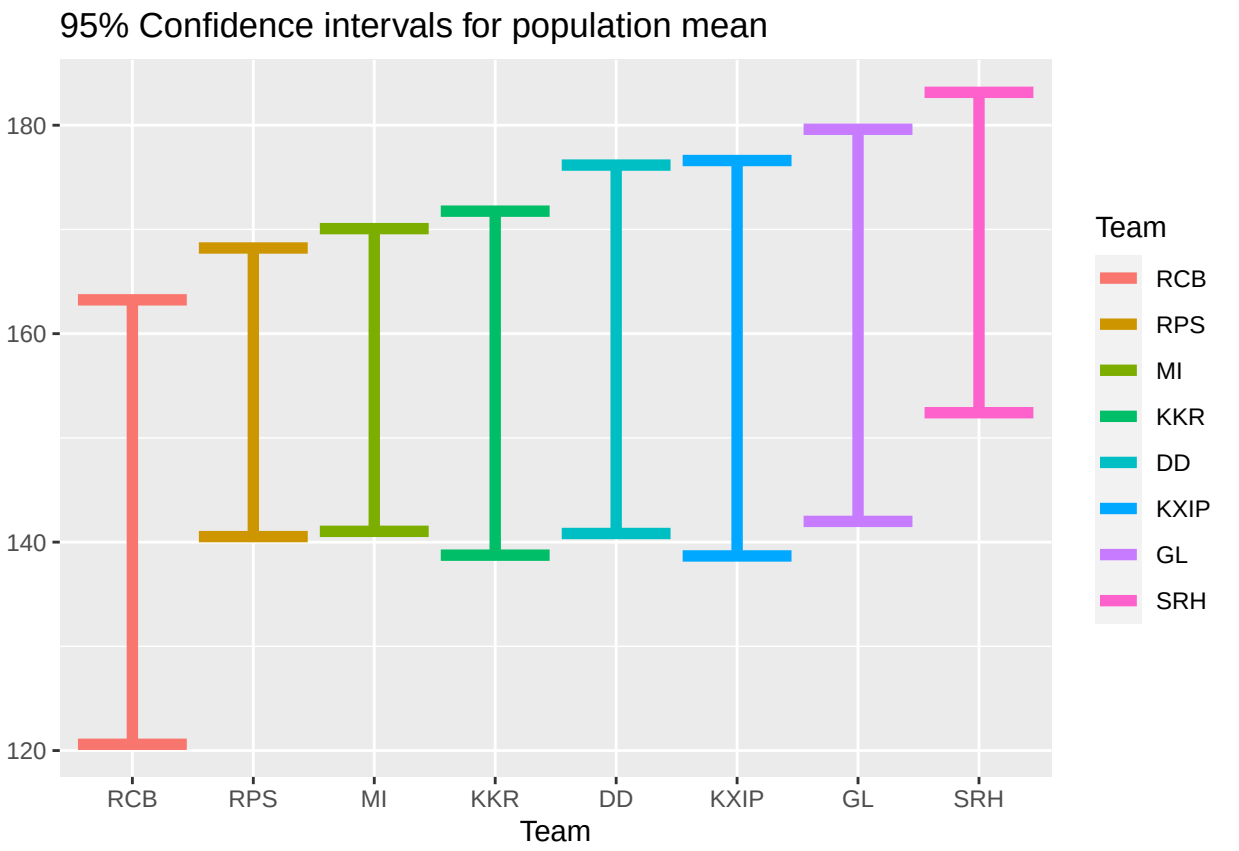
1. Known population varience (with a 95% confidence interval i.e $\alpha = 0.5$).

When the population varience σ^2 is known our formuale for the population mean(μ) interval estimator is $P(\overline{X} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} < \mu < \overline{X} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}})$ where $z_{\alpha/2} = \frac{\alpha}{2}$(calculated from a z table).

In this case $z_{\alpha/2}$ turns out to be close to 1.96. We can calculate the mean from our sample and we can find n by checking the total matches played by each team.

The confidence intervals for each team are given below.

```
ie=data.frame(expdat%>%filter(season==2017))
ie1=data.frame(ie%>%group_by(id,inning)%>%summarise(Team=unique(batting_team),Runs=sum(total_runs)))

tea=unique(ie$batting_team)
upperbou=NULL
lowerbou=NULL
te=NULL
for(x in tea){


  ie2=ie1%>%filter(Team==x)

  m=mean(ie2$Runs)
  univ=inning_runs%>%filter(Team==x)

  dev=(1.96*sd(univ$Runs))/sqrt(length(ie2$Runs))
  te=append(te,x)
  upperbou=append(upperbou,mean(ie2$Runs)+dev)
  lowerbou=append(lowerbou,mean(ie2$Runs)-dev)
}


ies=data.frame(u=upperbou,l=lowerbou,Team=te)
ies$Team=factor(ies$Team,levels=ies$Team[order(ies$u)])
ies%>%ggplot(aes(x=Team,col=Team))+geom_errorbar(aes(ymax=u,ymin=l),size=2)+
labs(title="95% Confidence intervals for population mean")
```

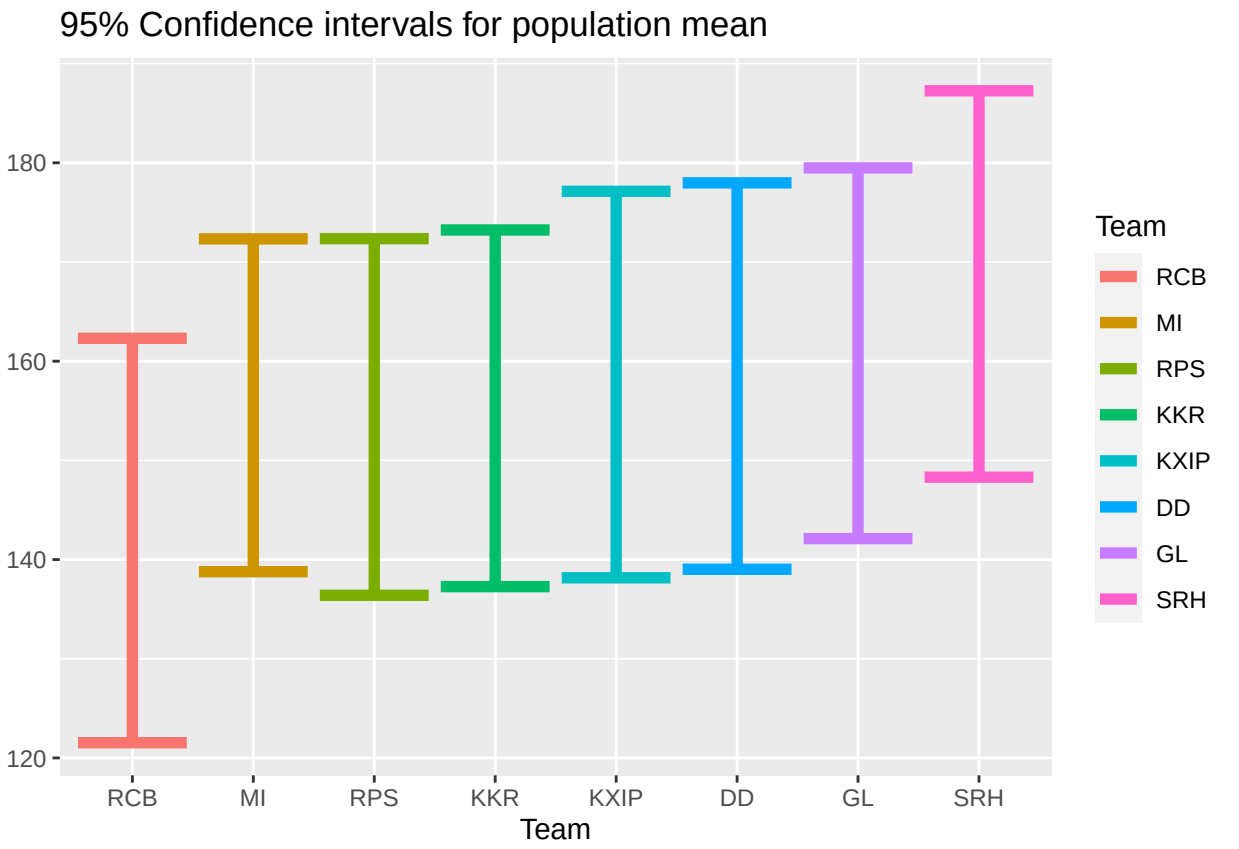


2. Unknown population varience (a 95% confidence interval i.e $\alpha = 0.5$)
   When the population varience is not known, our formulae is, $P(\overline{X} - t_{n-1,\alpha/2}\frac{S}{\sqrt{n}} < \mu < \overline{X} + t_{n-1,\alpha/2}\frac{S}{\sqrt{n}})$ where $t_{n-1,\alpha/2} = \frac{\alpha}{2}$ (calculated from a t dist. table).

S is the sample standard deviation and n is the total number of matches played by each team.
The confidence intervals for each team are given below.

```
tea=unique(ie$batting_team)
upperbou=NULL
lowerbou=NULL
te=NULL
for(x in tea){


  ie2=ie1%>%filter(Team==x)

  m=mean(ie2$Runs)

  dev=(qt(0.975,df=length(ie2$Runs)-1)*sd(univ$Runs))/sqrt(length(ie2$Runs))
  te=append(te,x)
  upperbou=append(upperbou,mean(ie2$Runs)+dev)
  lowerbou=append(lowerbou,mean(ie2$Runs)-dev)
}
```

```
ies=data.frame(u=upperbou,l=lowerbou,Team=te)
ies$Team=factor(ies$Team,levels=ies$Team[order(ies$u)])
ies%>%ggplot(aes(x=Team,col=Team))+geom_errorbar(aes(ymax=u,ymin=l),size=2)+
labs(title="95% Confidence intervals for population mean")
```



Conclusion: Both the above plots are very similar, but in the first case where we used a z distribution we have intervals with less width than in the second case where we used the t distribution. SRH has a relatively higher mean compared to other teams. As we saw in part 1, a higher score means a higher chance of win. Based on the above plots alone, we can conclude that SRH has a higher chance of winning than other teams.

7

In this question
We will try to predict if a team will win or lose a match given the total runs the team scored in that match. We will use a simple linear model to do this. Matches in 2017 season will be considered as test set while the rest will be considered as a training set.

```
ct=data.frame(expdat[expdat$season<2017,]%>%group_by(id,inning)%>%
summarise(team=unique(batting_team),s=sum(total_runs)
,w=ifelse(team==unique(winner),1,0)))%>%select(w,inning,s)

ct1=data.frame(expdat[expdat$season==2017,]%>%group_by(id,inning)%>%
summarise(team=unique(batting_team),s=sum(total_runs)
```

```
,w=ifelse(team==unique(winner),1,0)))%>%select(w,inning,s)
ct1$w=factor(ct1$w)


fit=lm(w~s,data=ct)
```

Slope and intercept of the regression line with best fit are given below

```
fit$coefficients
```

```
## (Intercept)           s
##  0.02565159  0.00313421
```

```
phat=predict(fit,newdata=ct1)
yhat=factor(ifelse(phat>0.5,1,0))
```

Below is a dataframe of predicted values based on our model and original values.(0-LOSS,1-WIN)

```
reg=data.frame(predicted=yhat,original=factor(ct1$w))
print(reg)
```

```
##     predicted original
## 1           1        1
## 2           1        0
## 3           1        0
## 4           1        1
## 5           1        0
## 6           1        1
## 7           1        0
## 8           1        1
## 9           1        1
## 10          0        0
## 11          0        0
## 12          0        1
## 13          1        0
## 14          1        1
## 15          0        0
## 16          0        1
## 17          1        1
## 18          0        0
## 19          1        0
## 20          1        1
## 21          1        0
## 22          1        1
## 23          0        0
## 24          0        1
## 25          1        0
## 26          1        1
## 27          1        1
## 28          1        0
## 29          1        1
```

```
## 30          0        0
## 31          1        0
## 32          1        1
## 33          1        1
## 34          0        0
## 35          1        0
## 36          1        1
## 37          1        1
## 38          1        0
## 39          1        1
## 40          1        0
## 41          1        1
## 42          1        0
## 43          1        0
## 44          1        1
## 45          1        0
## 46          1        1
## 47          0        1
## 48          0        0
## 49          1        0
## 50          1        1
## 51          1        1
## 52          1        0
## 53          0        1
## 54          0        0
## 55          1        1
## 56          1        0
## 57          1        0
## 58          1        1
## 59          0        0
## 60          0        1
## 61          1        0
## 62          1        1
## 63          1        1
## 64          1        0
## 65          1        1
## 66          0        0
## 67          1        0
## 68          1        1
## 69          0        1
## 70          0        0
## 71          0        0
## 72          0        1
## 73          1        1
## 74          1        0
## 75          1        0
## 76          1        1
## 77          1        0
## 78          1        1
## 79          1        0
## 80          1        1
## 81          1        0
## 82          1        1
## 83          1        0
```

```
## 84          1        1
## 85          0        1
## 86          0        0
## 87          0        1
## 88          0        0
## 89          1        1
## 90          0        0
## 91          1        0
## 92          1        1
## 93          1        0
## 94          1        1
## 95          0        0
## 96          0        1
## 97          1        1
## 98          1        0
## 99          1        0
## 100         1        1
## 101         1        1
## 102         1        0
## 103         1        1
## 104         1        0
## 105         1        0
## 106         1        1
## 107         1        1
## 108         1        0
## 109         0        0
## 110         0        1
## 111         1        1
## 112         0        0
## 113         1        1
## 114         0        0
## 115         0        0
## 116         0        1
## 117         0        0
## 118         0        1
## 119         0        1
## 120         0        0
```

The accuracy of our model is given below

```
confusionMatrix(yhat,factor(ct1$w))[[3]][1]
```

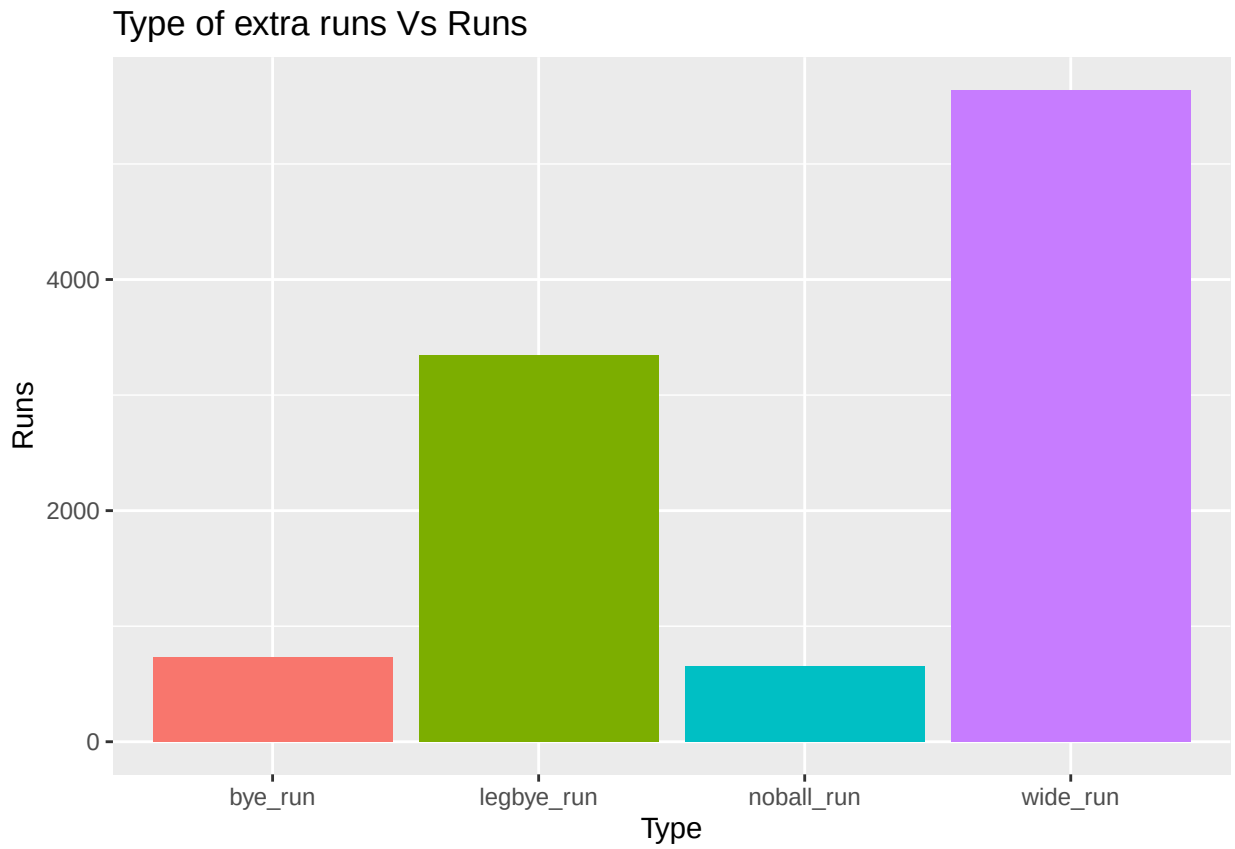
```
##  Accuracy
## 0.5666667
```

CONCLUSION: The accuracy acheived by our model is 56% which is very low and thus our model is not a good one.

PART-4 : Miscellaneous

1 Which type of extras resulted in giving more runs

```
rux=expdat%>%summarise(wide_run=sum(wide_runs),bye_run=sum(bye_runs),
legbye_run=sum(legbye_runs),noball_run=sum(noball_runs))
rux1=data.frame(dt=colnames(rux),v=as.matrix(rux)[1:4])
rux1%>%ggplot(aes(dt,v,fill=dt))+geom_bar(stat = "identity")+
labs(x="Type",y="Runs",title = "Type of extra runs Vs Runs")+
theme(legend.position = "none")
```



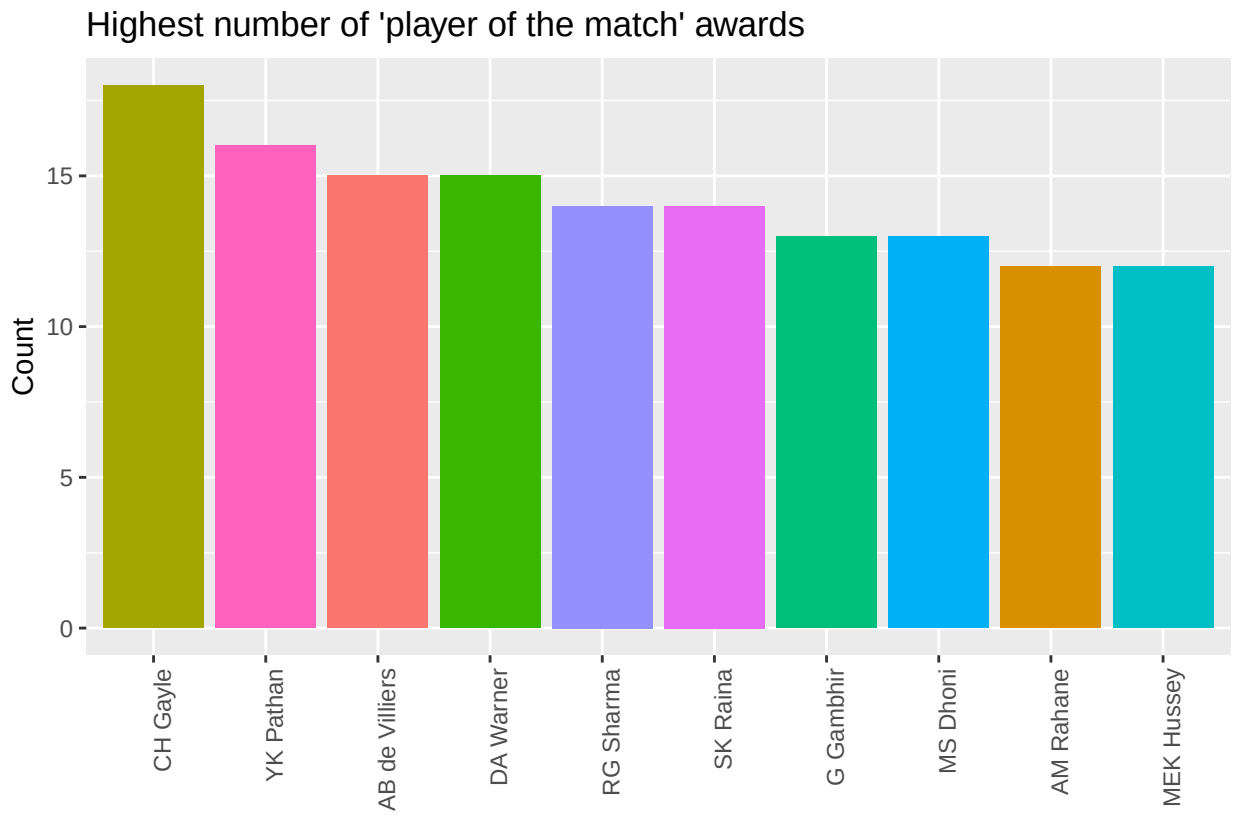
Conclusion: As seen above, wide runs contribute much more than any other type of extra. If we personally see few matches, we can clearly observe that wide runs are more prominent than others.

2 Which players received the most awards

After every match is completed, a "Player of the match" award is given to the best performing player of the match. Let's see which player received the most of them.

```
dyna=expdat%>%group_by(id)%>%summarise(pom=unique(player_of_match))
dyna1=data.frame(pom=dyna$pom,v=1)
dyna2=dyna1%>%group_by(pom)%>%summarise(n=sum(v))%>%arrange(desc(n))
```

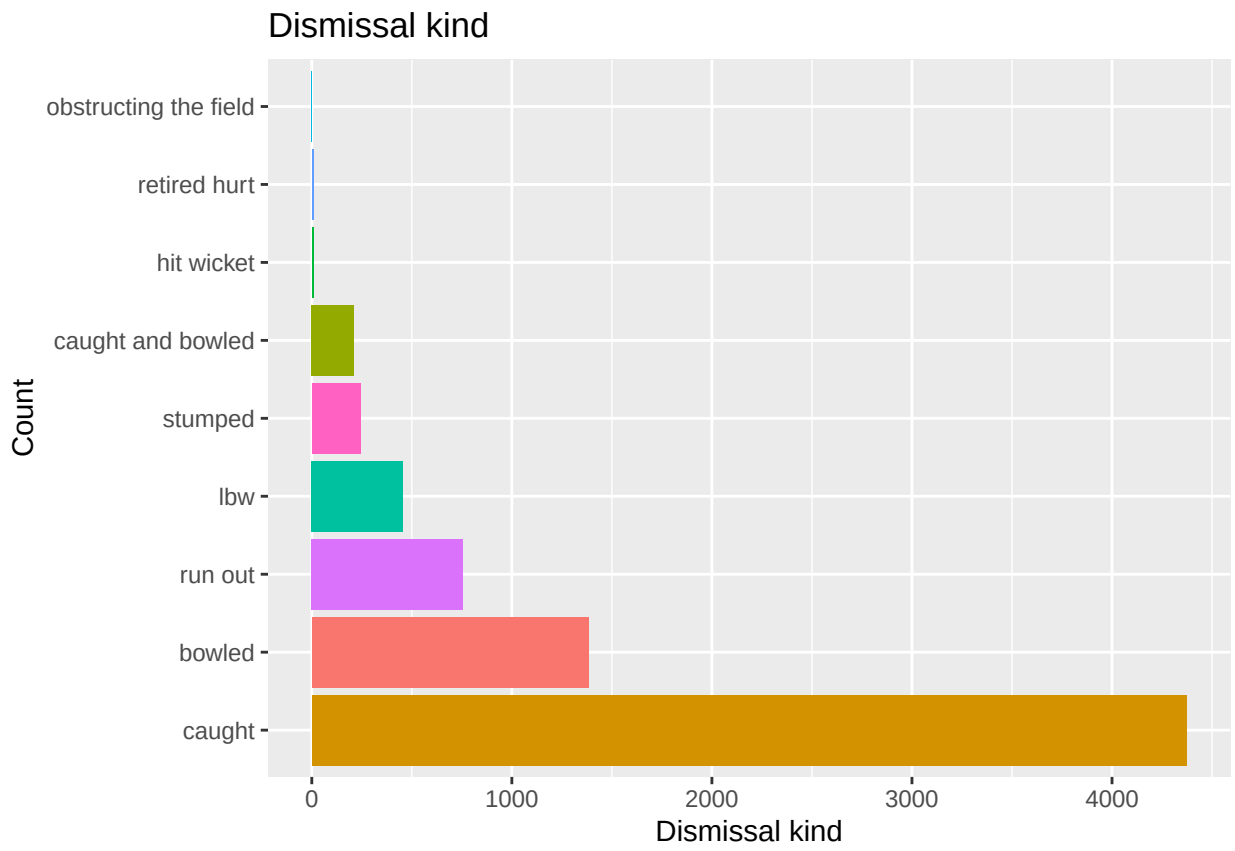
```
dyna2[1:10,]%>%ggplot(aes(reorder(pom,-n),n,fill=pom))+geom_bar(stat="identity")+
theme(axis.text.x = element_text(angle = 90, hjust = 1))+ theme(legend.position = "none")+
labs(x="",y="Count",title = "Highest number of 'player of the match' awards")
```



Conclusion: Chris Gyle tops the list with more than 16 awards followed by YK Pathan and AB de Villers.

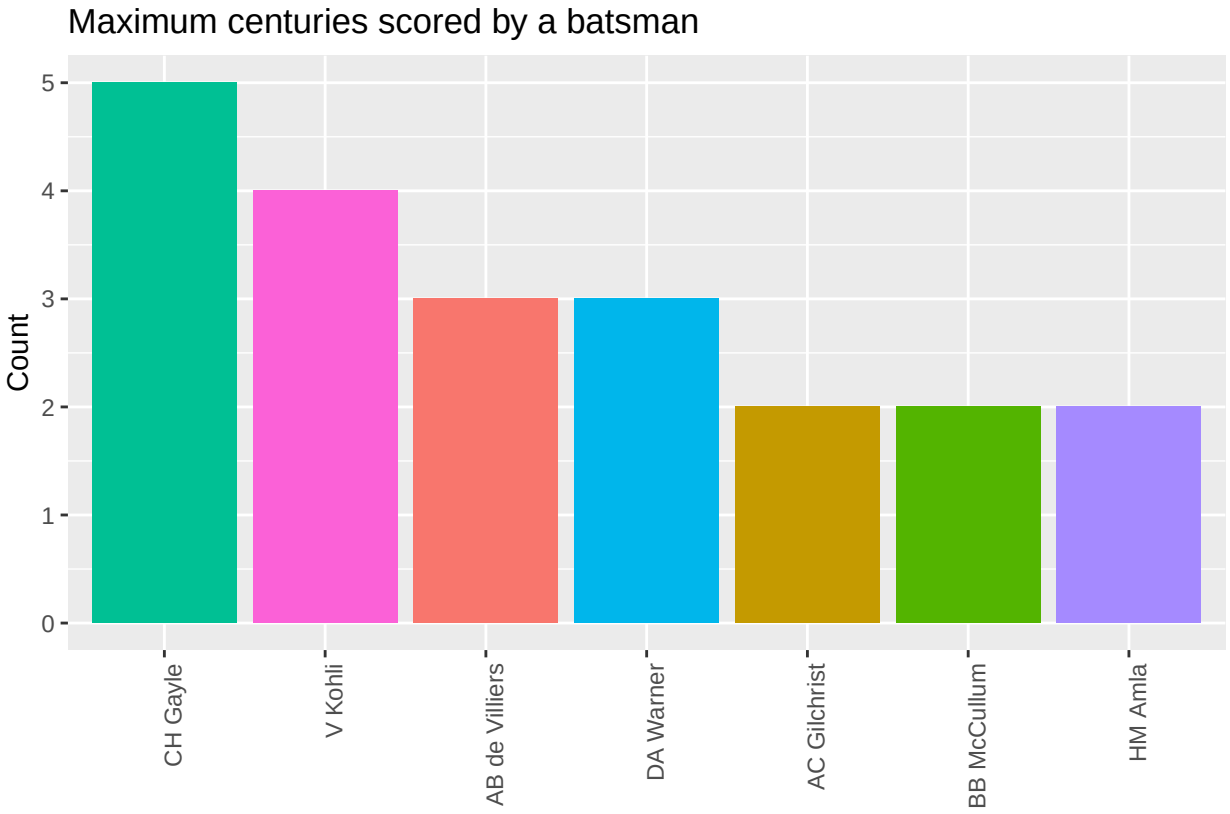
3 Which kind of dismmisal was more common?

```
bwluv=data.frame(dk=expdat$dismissal_kind,v=1)
bwluv1=bwluv%>%group_by(dk)%>%summarise(n=sum(v))
bwluv1=data.frame(bwluv1[2:10,]%>%arrange(desc(n)))
bwluv1%>%ggplot(aes(n,reorder(dk,-n),fill=dk))+geom_bar(stat = "identity")+
theme(legend.position = "none")+labs(x="Dismissal kind",y="Count",title="Dismissal kind")
```



4 Maximum centuries scored by a player

```
idol=expdat%>%group_by(id,batsman)%>%summarise(r=sum(batsman_runs),v=ifelse(r>99,1,0))
idol1=idol%>%group_by(batsman)%>%summarise(n=sum(v))%>%arrange(desc(n))

idol1[1:7,]%>%ggplot(aes(reorder(batsman,-n),n,fill=batsman))+geom_bar(stat = "identity")+
theme(axis.text.x = element_text(angle = 90, hjust = 1))+ theme(legend.position = "none")+
labs(x="",y="Count",title="Maximum centuries scored by a batsman")
```

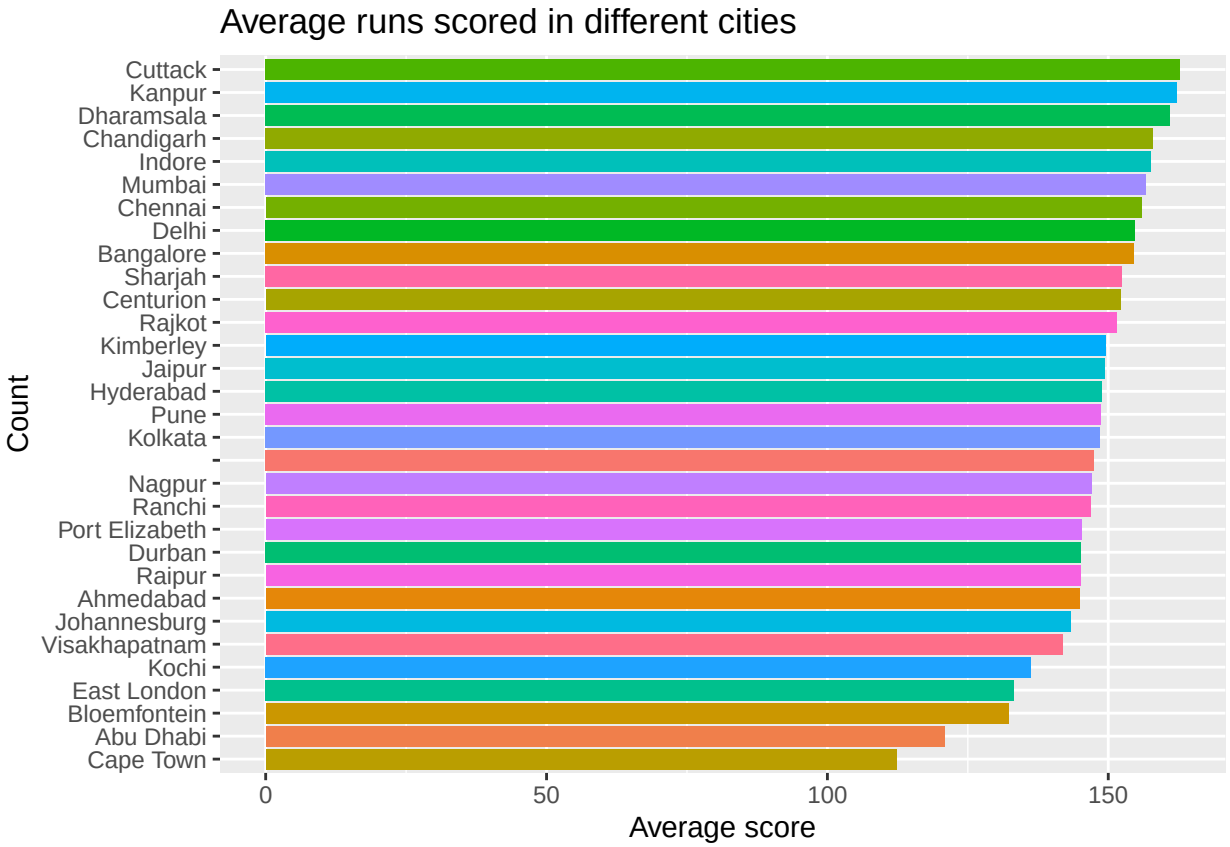


Conclusion: Chris Gyle again tops the list with 5 centuries followed by Virat Kohli with 4.

5 Average score per match in each city

Lots of conditions can affect a match which include the stadium they are playing in. Every stadium is not the same and can have a lot of variation. Few grounds may be short and easy to score runs, few might be very big and relatively difficult to score runs.So, let us see how the average scores for each inning changes for every stadium.

```
lou=expdat%>%group_by(id,inning)%>%summarise(t=sum(total_runs),c=unique(city))
lou1=lou%>%group_by(c)%>%summarise(ms=mean(t))%>%arrange(desc(ms))

lou1%>%ggplot(aes(ms,reorder(c,ms),fill=c))+geom_bar(stat = "identity")+
theme(legend.position = "none")+labs(x="Average score",y="Count",title="Average runs scored in differen
```



PART-5 : Key take-aways/ Conclusion

The Key take-aways from all the above analysis are:

1:

The "Mean" of data is highly affected by outliers in the data, on the contrary "Median" is relatively unaffected by the outliers

2:

The players are hitting more sixes than ever before. The number of sixes scored is gradually increasing over the years

3:

Scoring a higher total in a match leads to a greater chance in winning the match

4:

The result of the toss does not greatly effect the result of the match

5:

The Sophmore Slump phenomena is definetly true. Moreover, the "regression to the mean" effect is more prominent

6:

Teams "CSK" and "MI" have a very high chance of qualifying and also winning all the matches compared to their peer teams

7:

A simple linear regression model only acheives a 56% accuracy in predicting the result of a match which is not a huge number.

8:

The distribution of the runs scored is close to a bell curve. A normal distribution with appropriate mean and standard deviation is a close approximation to the runs distribution

9:

Wide runs were the largest contributor of extra runs compared to others

10:

Dismissal by a "catch" was more prominent than other dismissal types in the league

11:

Chris Gayle has the title of "Highest number of centuries scored" and also has the title of winning the most "Player of the match" awards.