Visulization using GGplot

”Kumar Rahul

kumarrah@iimb.ac.in; rahul235@gmail.com

Contents

1 Data Visualization in R

To demonstrate usage of `ggplot`, we will use IPL dataset (described in the next section) to load data into data.frame and perform descriptive analytics. We will use `data.table` and `ggplot2` to perform data wrangling.

We will prepare plots such as bar plot, histogram, distribution plot, box plot, scatter plot, tren plot and heat maps.

1.1 IPL Dataset Description

The list features are:

Data Code	Description
AGE	Age of the player at the time of auction classified into 3 categories. Category 1 (L25) means the player is less than 25 years old, 2 means that the age is between 25 and 35 years (B25- 35)and category 3 means that the age is more than 35 (A35).
RUNS S	Number of runs scored by a player
RUNS C	Number of runs conceded by a player
HS	Highest score by a batsman in IPL
AVE B	Average runs scored by a batsman in IPL
AVE BL	Bowling average (Number of runs conceded / number of wickets taken) in IPL.
SR B	Batting strike rate (ratio of the number of runs scored to the number of balls faced) in IPL
SR BL	Bowling strike rate (ratio of the number of balls bowled to the number of wickets taken) in IPL
SIXERS	Number of six runs scored by a player in IPL
WKTS	Number of wickets taken by a player in IPL
ECON	Economy rate of a bowler (number of runs conceded by the bowler per over) in IPL

Data Code	Description
CAPTAINCY EXP	Captained either an T20 team or a national team
ODI SR B	Batting strike rate in One Day Internationals
ODI SR BL	Bowling strike rate in One Day Internationals
ODI RUNS S	Runs scored in One Day Internationals
ODI WKTS	Wickets taken in One Day Internationals
T RUNS S	Runs scored in Test matches
T WKTS	Wickets taken in Test matches
PLAYER SKILL	Player's primary skill (batsman, bowler, or all-rounder)
COUNTRY	Country of origin of the player (AUS: Australia; IND: India; PAK: Pakistan; SA: South Africa; SL: Sri Lanka; NZ: New Zealand; WI: West Indies; OTH: Other countries)
YEAR A	Year of Auction in IPL
IPL TEAM	Team(s) for which the player had played in the IPL (CSK: Chennai Super Kings; DC: Deccan Chargers; DD: Delhi Daredevils; KXI: Kings XI Punjab; KKR: Kolkata Knight Riders; MI: Mumbai Indians; PWI: Pune Warriors India; RR: Rajasthan Royals; RCB: Royal Challengers Bangalore). A + sign was used to indicate that the player had played for more than one team. For example, CSK+ would mean that the player had played for CSK as well as for one or more other teams.

1.2 Invoke packages

Load packages

```r
library(dplyr)
library(DT)
library(data.table)
library(ggplot2)
```

Read the data

```r
ipl_dt = fread(file='./files/IPL.csv',stringsAsFactors=TRUE)
colnames(ipl_dt) =  gsub(" ", "_", colnames(ipl_dt))
knitr::kable(head(ipl_dt[,1:10]), caption = "IPL Data")
```

1.3 Exploration of Data using Visualization

We will be discussing various plots that we can draw using R by using `ggplot2` package. `ggplot` implements the grammer of graphics. More about the `grammer of graphics` at http://vita.had.co.nz/papers/layered-

Table 2: IPL Data

Sl_NO	PLAYER_NAME	AGE	COUNTRY	TEAM	PLAYING_ROLE	T_RUNS	T_WKTS	ODI_RUNS_
1	Abdulla, YA	2	SA	KXIP	Allrounder	0	0	
2	Abdur Razzak	2	BAN	RCB	Bowler	214	18	6
3	Agarkar, AB	2	IND	KKR	Bowler	571	58	12
4	Ashwin, R	1	IND	CSK	Bowler	284	31	2
5	Badrinath, S	2	IND	CSK	Batsman	63	0	
6	Bailey, GJ	2	AUS	CSK	Batsman	0	0	1

Table 3: Summary

AVE_BL	SR_BL	TEAM
0.00	0.00	RCB
36.52	24.90	KKR
22.96	22.14	CSK
25.81	19.40	CSK+
18.73	15.57	CSK
25.72	21.19	KKR+

grammar.pdf

To understand the basics of `grammer of graphics`, let's take a simple example of three variables. Bowling Average (`AVE_BL`) in past IPL matches, Bowling strike rate (`SR_BL`) in past IPL matches and team these players belong too `TEAM` for all the bowlers from `ipl_dt`. From the definition sake,

- AVE_BL is `number of runs conceded / number of wickets taken`. The lower the bowling average, the better the performance.
- SR_BL is `number of balls bowled / number of wickets taken`. The lower the bowling strike rate, the better the performance.

```
knitr::kable(head(ipl_dt[PLAYING_ROLE=="Bowler",.(AVE_BL,SR_BL,TEAM)]), caption = "Summary")
```

- ggplot(): creates a coordinate system (empty graph) to which different layers are added.
- geom_objects: A geom is the geometrical object that a plot uses to represent data. geom_point() adds a layer of points which creates a scatter plot.

ggplot2 comes with many geom functions that each add a different type of layer to a plot.

1.3.1 Scatter Plot

Scatter plot assists in understanding if there is any relationship between two variables. The relationship could be linear or non-linear.

Let's us plot `AVE_BL` vs. `BL_SR` using the `ggplot` package

```
scatter_plot = ggplot(ipl_dt) +
geom_point(mapping=aes(x = AVE_BL, y = SR_BL, color = TEAM)) +
xlab(" Bowling Average") +
ylab("Strike Rate") +
ggtitle("Plot Bowling average \n and strike rate")
```

```
scatter_plot
```

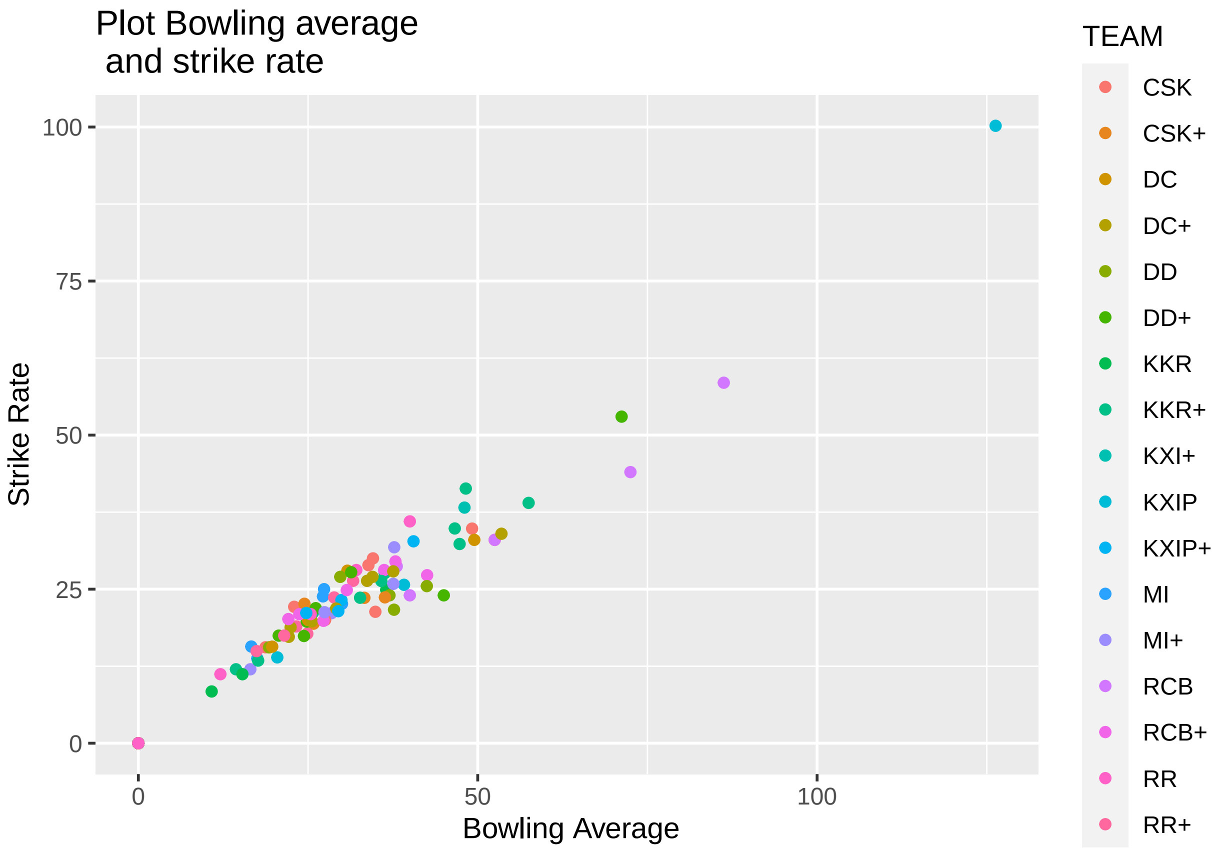

Figure 1: Scatter Plot 1

Notice that, we have used `plot annotations` to name the x-axis, y-axis, and title (by default is left alinged). `theme` gives more control on the `plot annotations`. Get the help on `?ggplot2::theme` and `?ggplot2::element_text` which works conjunction with `theme`

```
scatter_plot + theme(plot.title = element_text( face = 'bold.italic', colour = "blue", size = 12,
                                               hjust= 0.5),
                     axis.title.x = element_text(face = 'bold', colour = "red", size = 12),
                     axis.title.y = element_text(face = 'bold', colour = "red", size = 12))
```

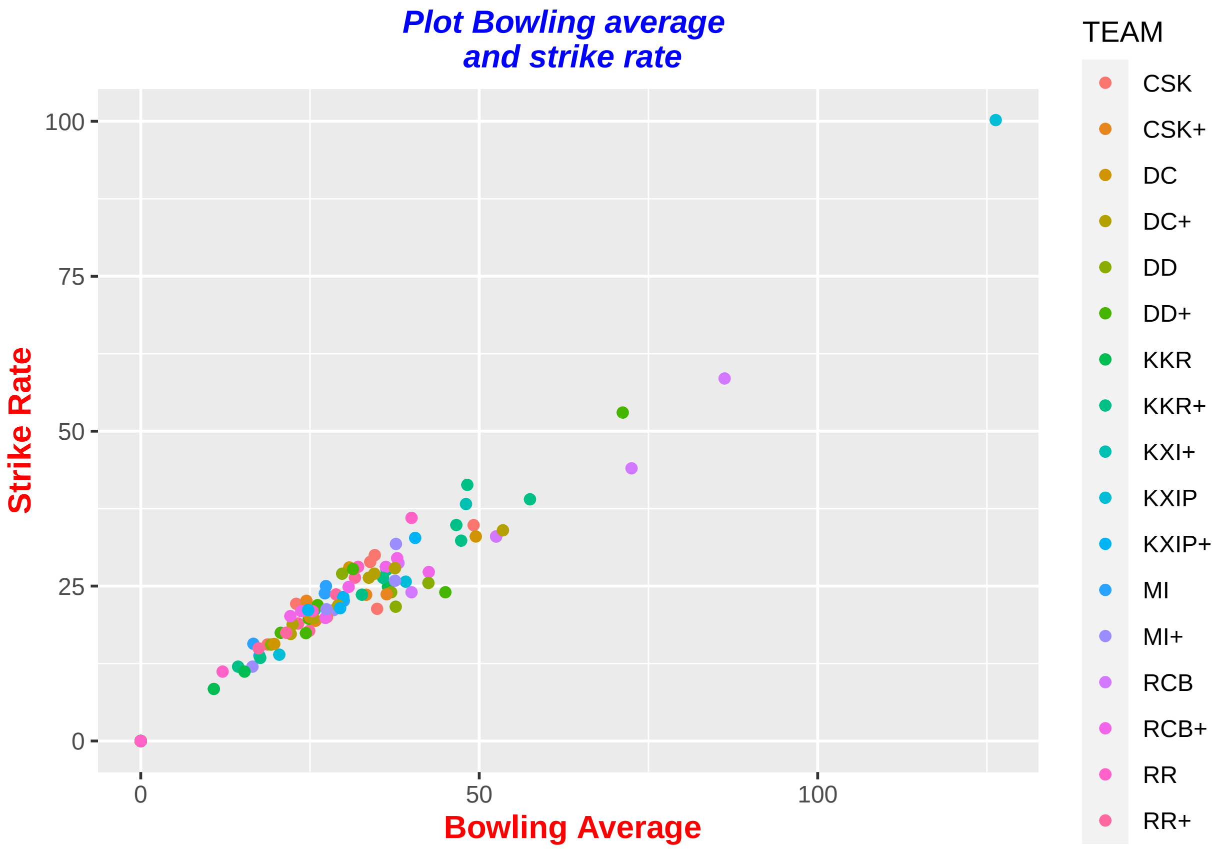

Figure 2: Scatter Plot 2

```
scatter_plot + theme(plot.title = element_text( face = 'bold.italic', colour = "blue", size = 12,
                                               hjust= 0.5),
                     axis.title.x = element_text(face = 'bold', colour = "red", size = 12),
                     axis.title.y = element_text(face = 'bold', colour = "red", size = 12),
                     # remove panel border
                     panel.border = element_blank(),
                     # remove grid lines
                     panel.grid.major = element_blank(),
                     panel.grid.minor = element_blank(),
                     # Remove panel background
                     panel.background = element_blank(),
                     # Change axis line
                     axis.line = element_line(colour = "black"))
```

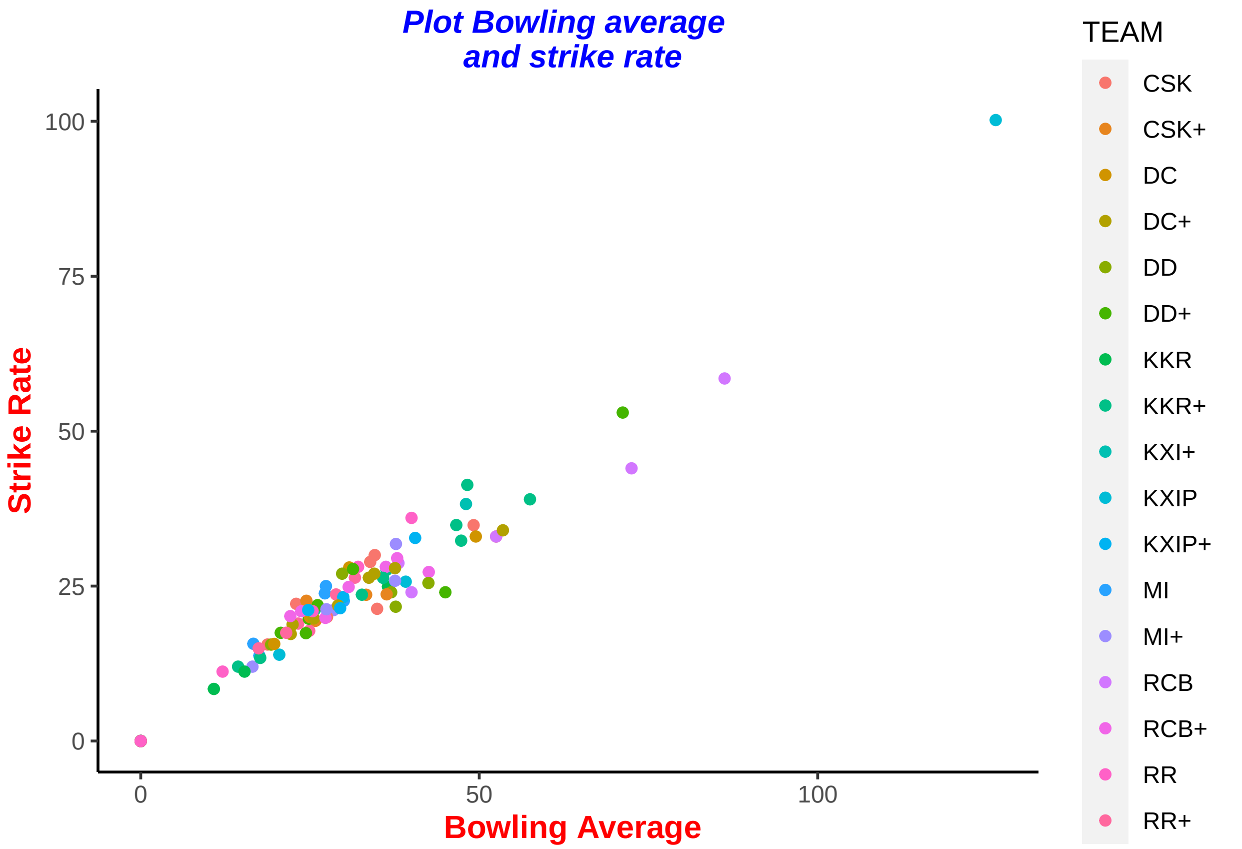

Figure 3: Scatter Plot 3

We can also remove the title and axis labels by using `element_blank()` within the `theme()`.

1.3.1.1 Exercise

Create a custom theme and use across all the plots.

1.3.2 Bar Chart

Bar chart is a frequency chart for qualitative variable (or categorical variable). Bar chart can be used to assess the most-occurring and least-occurring categories within a dataset.

To draw a bar chart, we can:

- use geom_bar() which makes the height of the bar proportional to the number of cases in each group (or if the weight aesthetic is supplied, it sums up the weights).
- use `geom_col`, in case we want height of the bar to represent actual values. The dataframe should be passed in the parameter data.

`geom_bar()` uses `stat_count()` by default and thus counts the number of cases at each x position. `geom_col()` uses `stat_identity()` which leaves the data as is.

```
ipl_dt$AGE = as.factor(ipl_dt$AGE)
```

To get a count of players by `PLAYING_ROLE`

```
ggplot(ipl_dt, mapping = aes(x =AGE))+
geom_bar(mapping = aes( fill = AGE))+
geom_text(stat='count', mapping = aes(label=..count..), vjust=1) +
ylab("Count of Players") +
ggtitle("Player Count") +
theme_bw()
```

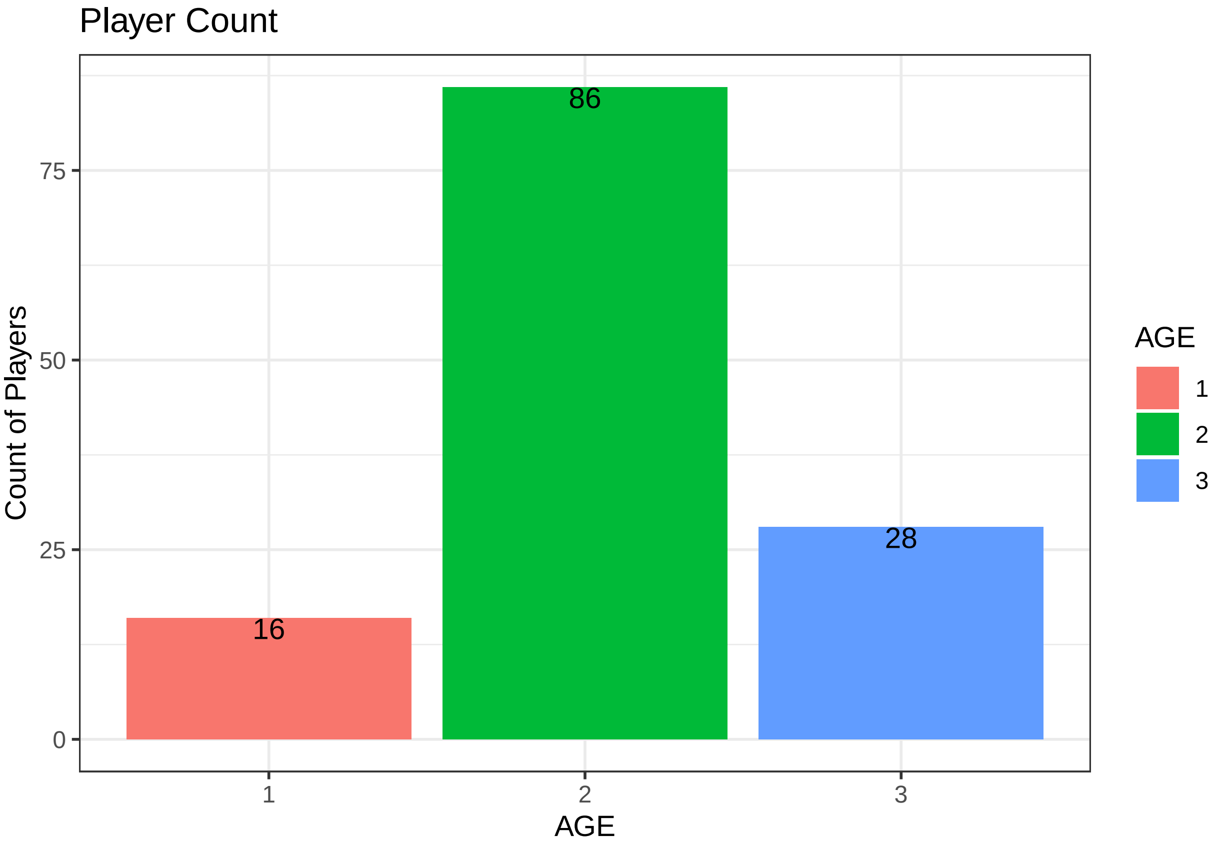

Figure 4: Bar Plot 1

Majority of the players are in Age group of 25-35 (AGE 2). To get the sum of `SOLD_PRICE` for different `AGE`.

1.3.2.1 Exercise - Bar plot 1

Change the above plot to center the title.

1.3.3 Bar plot cont...

Note that to show the y axis values in `dollars` we have passed labels argument in `scale_y_continuous` function. It can take three values i.e. `scales::percent`, `scales::dollar` and `scales::comma`.

```
ggplot(ipl_dt) +
geom_bar(aes(x = AGE, weight = SOLD_PRICE, fill = AGE)) +
ylab("Sold Price") +
ggtitle("Sum of Sold Price") +
theme_bw() +
scale_y_continuous(labels=scales::dollar)
```

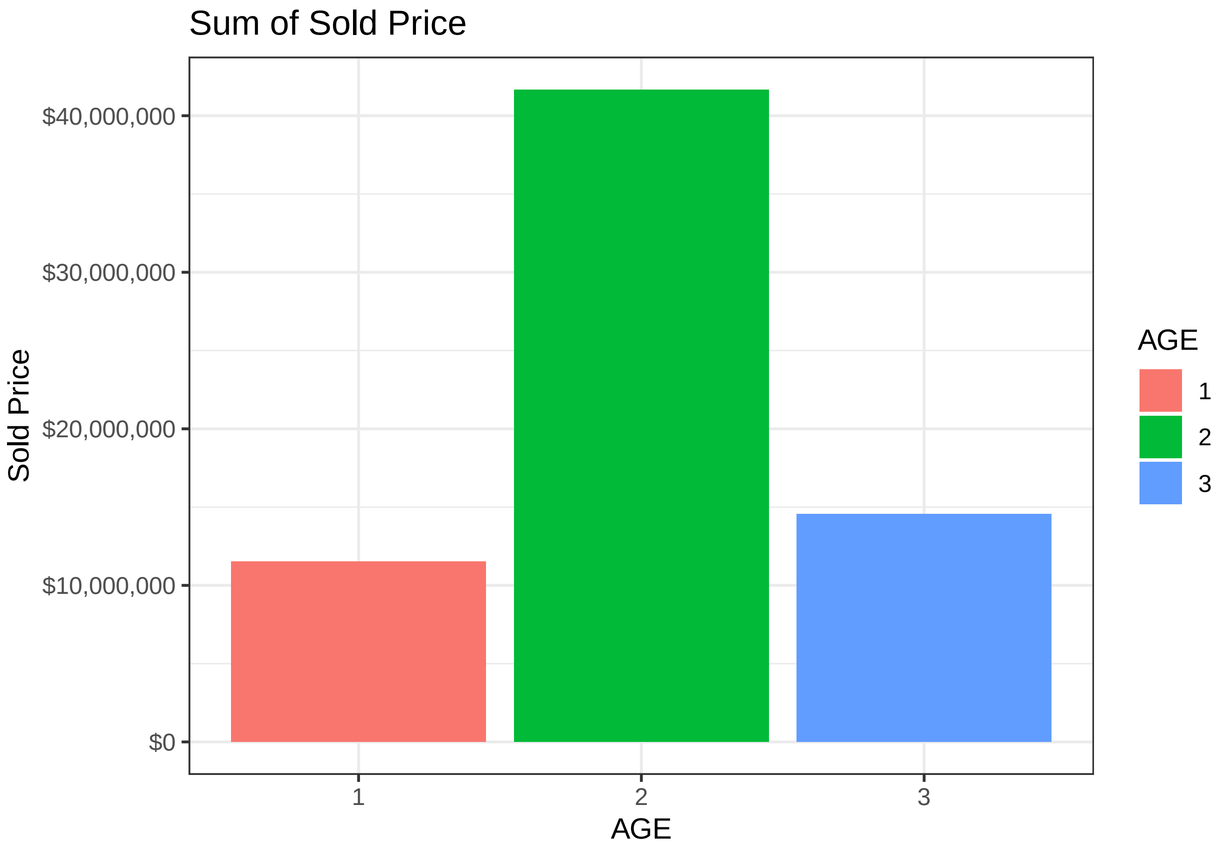

Figure 5: Bar Plot 2

To display the total sold price by each age category, we can use `geom_col` as shown below.

```
ggplot(ipl_dt) +
geom_col(aes(x = AGE, y = SOLD_PRICE, fill = AGE)) +
ylab("Sold Price") +
ggtitle("Sum of Sold Price") +
theme_bw() +
scale_y_continuous(labels=scales::dollar)
```

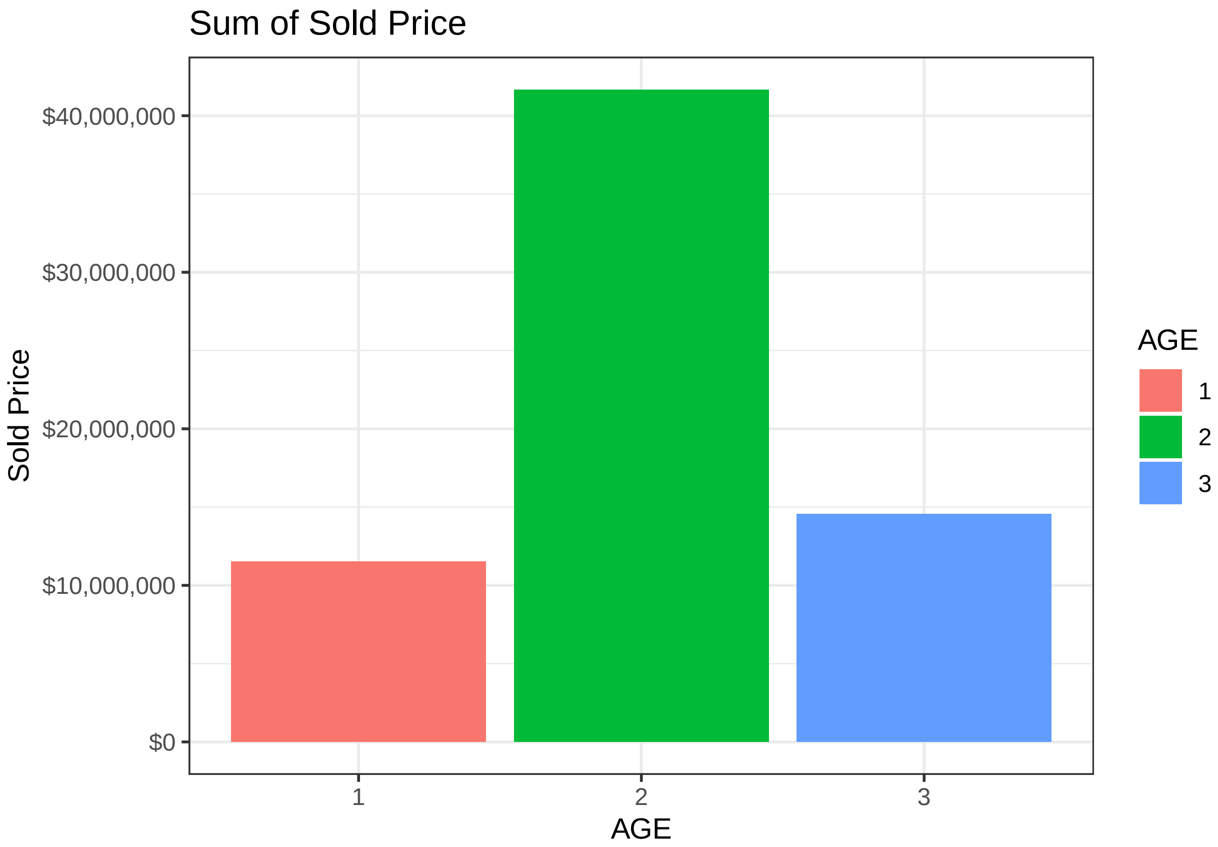

Figure 6: Bar Plot 3

We can also create bar charts which can be grouped by a third variable.

The parameter `fill` takes the third variable as a groupby parameter. In this case, we pass PLAYING ROLE as fill parameter.

```
ggplot(ipl_dt) +
geom_col(aes(x = AGE, y = SOLD_PRICE, fill = PLAYING_ROLE)) +
ylab("Sold Price") +
ggtitle("Sum of Sold Price") +
theme_bw() +
scale_y_continuous(labels=scales::dollar)
```

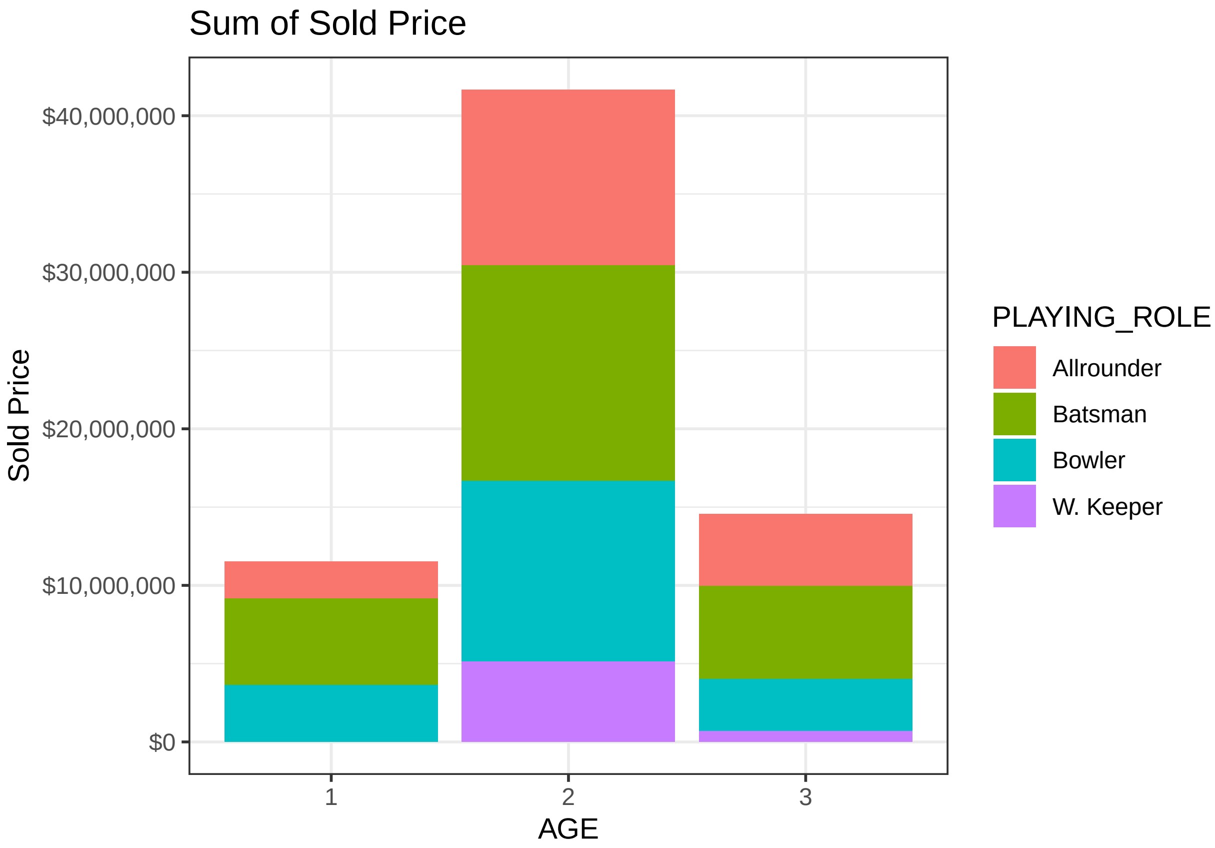

Figure 7: Bar Plot 4

In case, we want to show the relative proporations at each AGE group, we can use `position=position_fill()` which stacks the bars and then standardizes each bar to have the same height.

```
ggplot(ipl_dt) +
geom_col(aes(x = AGE, y = SOLD_PRICE, fill = PLAYING_ROLE), position = position_fill()) +
ylab("Sold Price") +
ggtitle("Sum of Sold Price") +
theme_bw()
```

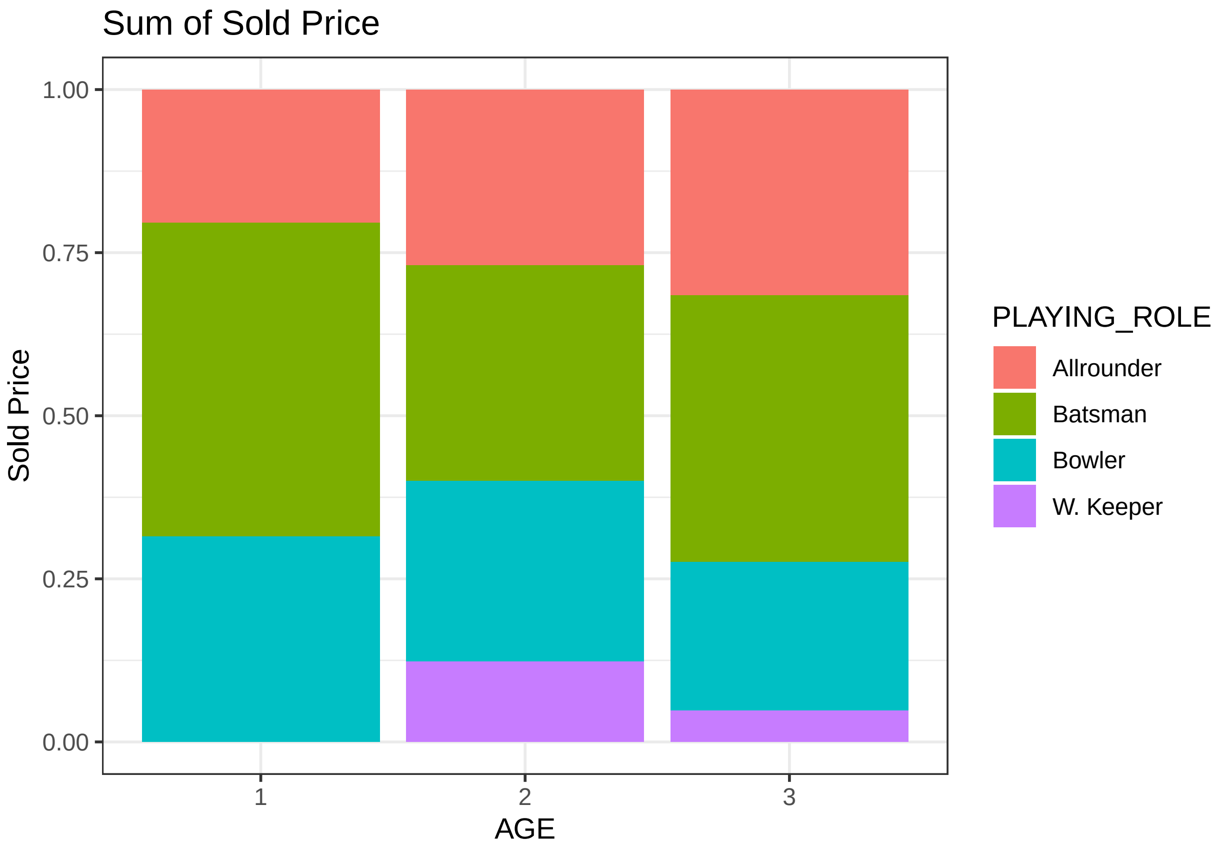

Figure 8: Bar Plot 5

We can compare the sold price for batsman in three different age group and come to conclusion that the batsman were sold for maximum value in AGE group 1, then in AGE group 3 and finally in Age group 2.

By default, the `geom_bar` or `geom_col` will give a stacked bar chart. To show the bars in dodged position, we set the position agrument as `position = position_dodge()`. By default `position = stack`.

```
ggplot(ipl_dt) +
geom_col(aes(x = AGE, y = SOLD_PRICE, fill = PLAYING_ROLE), position = position_dodge()) +
ylab("Sold Price") +
ggtitle("Sum of Sold Price") +
theme_bw() +
scale_y_continuous(labels =scales::dollar )
```

Table 4: Mean Sold Price by Age

AGE	AVG_SOLD_PRICE
1	720250.0
2	484534.9
3	520178.6

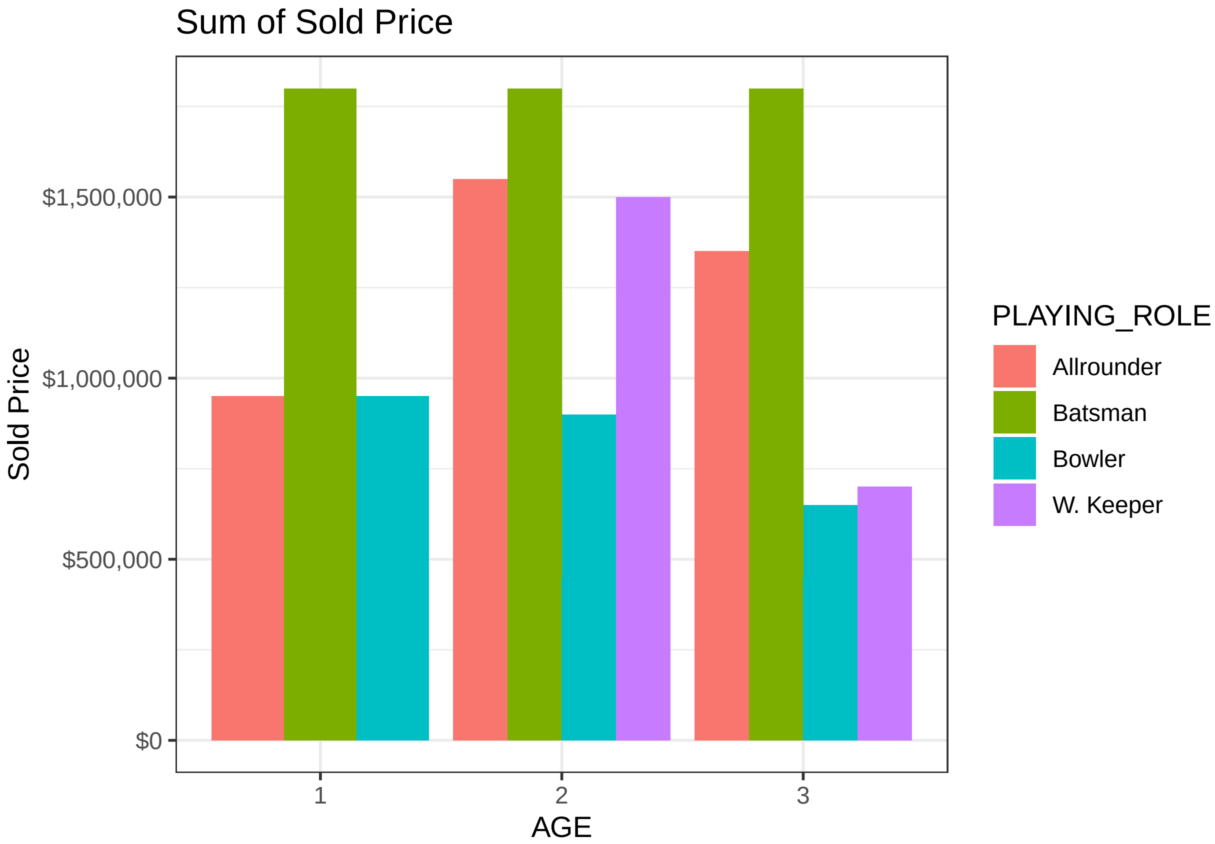

Figure 9: Bar Plot 6

If we want to show the average sold price for each AGE group, we will first create a data.table with this statistics

```
mean_sold_price_dt = ipl_dt[, .(AVG_SOLD_PRICE = mean(SOLD_PRICE)), keyby = .(AGE)]
knitr::kable(mean_sold_price_dt,caption = "Mean Sold Price by Age")
```

```
ggplot(mean_sold_price_dt) +
geom_col(mapping=aes(x = AGE, y = AVG_SOLD_PRICE), fill='blue') +
ylab("Avg Sold Price") +
ggtitle("Avg Sold Price with AGE") +
theme_bw() +
scale_y_continuous(labels =scales::dollar )
```

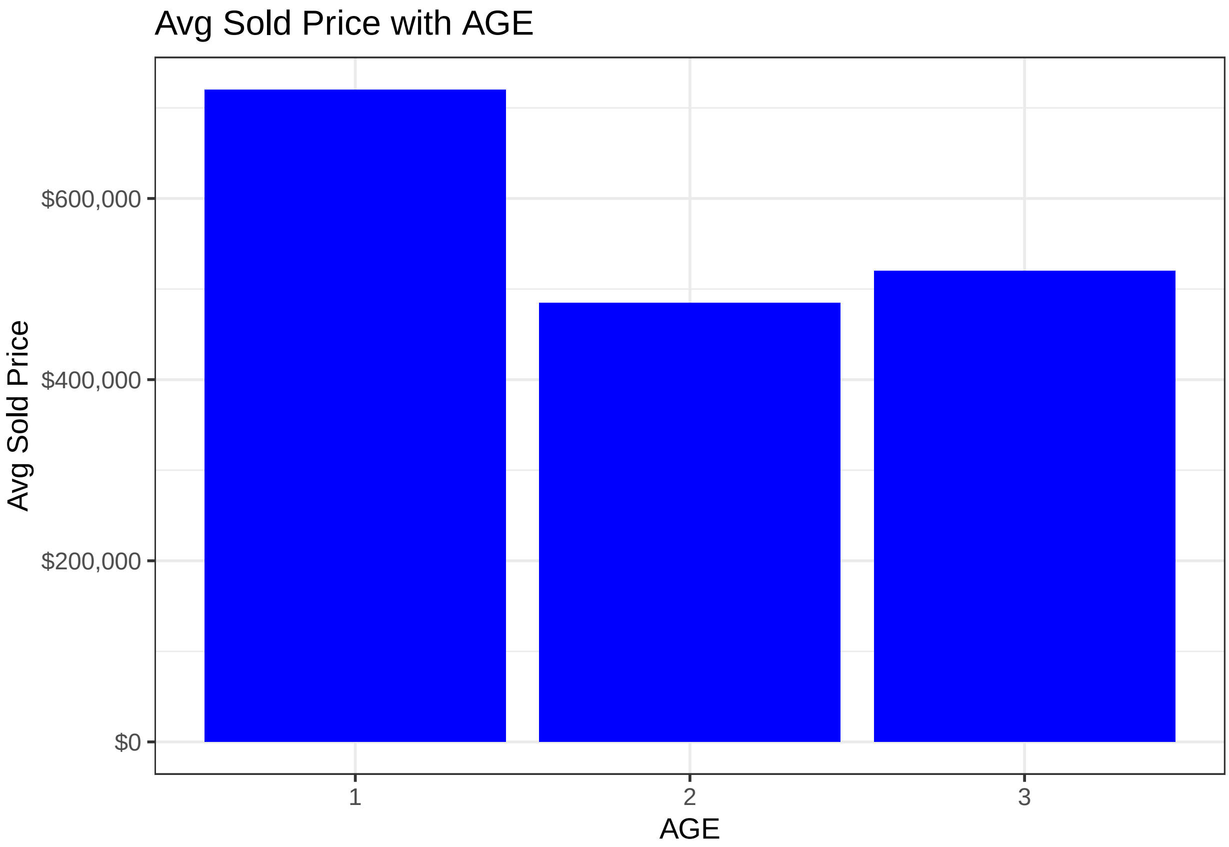

Figure 10: Bar Plot 7

Note that the `fill` is passed as an aes outside the mapping argument.

1.3.3.1 Exercise - Bar plot 2

Create bar charts to show average sold price by each age category but grouped by playing roles.

1.3.4 Histogram

A histogram is a plot that shows the frequency distribution of a set of continuous data. It may also be used to understand the skweness in the data

A histogram plot is skewed if one of its tails is longer than the other. The first distribution shown has a positive skew. This means that it has a long tail in the positive direction.

The distribution below has a negative skew since it has a long tail in the negative direction.

```
> If the distribution has a negative skew, the median is larger than the mean.
> If the distribution has a positive skew, the mean is larger than the median.
```

To draw a histogram use geom_histogram() function of ggplot2.

```
ggplot(ipl_dt) +
geom_histogram(aes(x = SOLD_PRICE), bins=10, fill = 'blue') +
ylab("Count") +
ggtitle("Histogram Plot of Sold Price") +
theme_bw() +
scale_x_continuous(labels =scales::dollar )
```

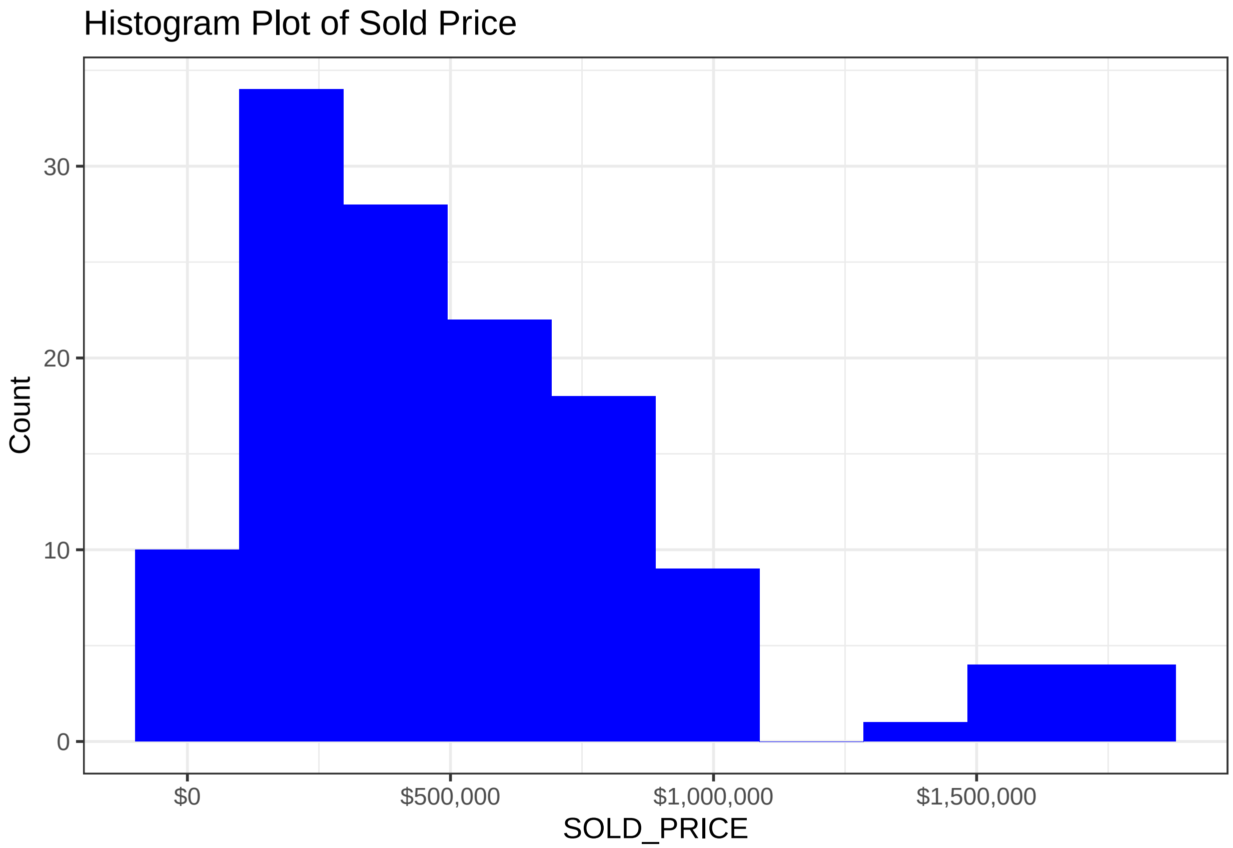

Figure 11: Histogram Plot 1

The histogram shows SOLD PRICE is right skewed. Most players are auctioned at low price range of 250000 and 500000, whereas there are few players who are paid very highly like more than 1 million dollars.

We can use the `fill` aesthetic to show a histogram plot of sold price of players with different playing roles.

```
ggplot(ipl_dt) +
geom_histogram(aes(x = SOLD_PRICE, fill = PLAYING_ROLE), bins=50) +
ylab("Count") +
ggtitle("Histogram Plot of Sold Price") +
theme_bw() +
scale_x_continuous(labels =scales::dollar )
```

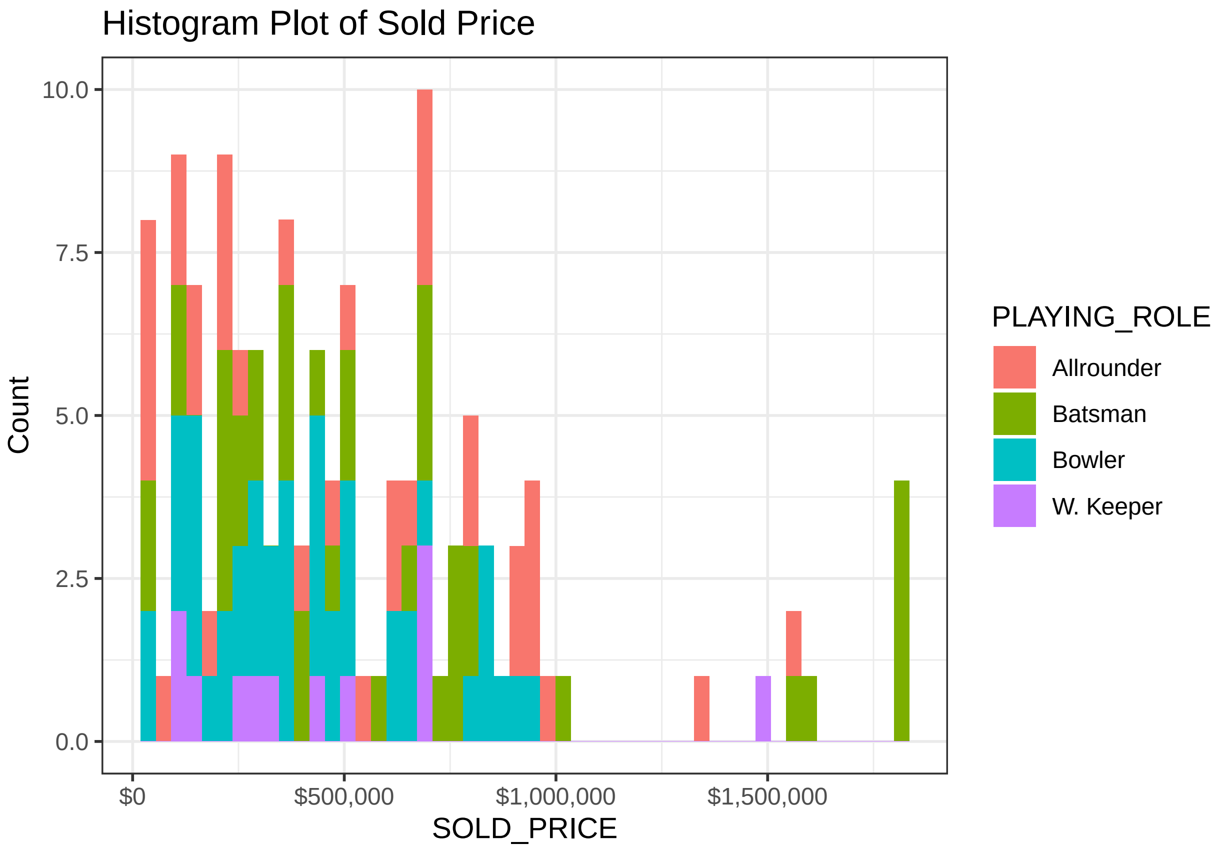

Figure 12: Histogram Plot 3

A variation of histogram `2d histogram` implemented via `geom_bin2d` or through `geom_hex` can be useful to show the relationship between two numeric features when the datasize is huge.

```
ggplot(ipl_dt) +
geom_hex(aes(x = BASE_PRICE, y = SOLD_PRICE), bins = 10) +
xlab("Base Price") +
ylab("Sold Price") +
theme_bw() +
scale_x_continuous(labels =scales::dollar ) +
scale_y_continuous(labels =scales::dollar ) +
scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0)
```

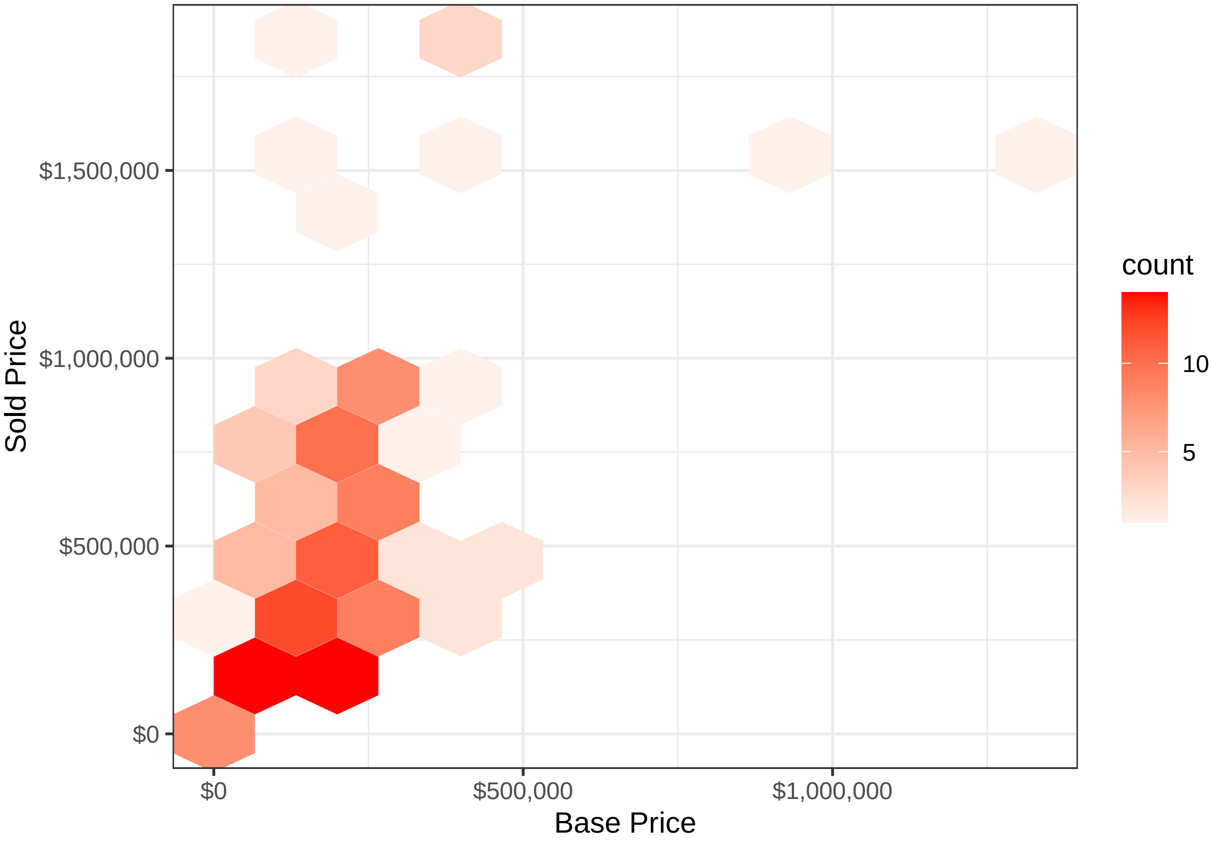

Figure 13: 2D Histogram Plot 2

Darker shade reflects a count of 10 or more players in the hexagonal bin and these players were sold on the base price. The lighter shade reflects a count of less than 5 players in the bin and some of these players were sold above $150,000 though there base price was approximately `$50,000` to `$100,000`.

1.3.4.1 Exercise - Histogram

Hexagon plot by filling the hexagons based on playing role.

1.3.5 Distribution or Density Plot

Before, understanding the density plot, let us understand some basic terminology which will be used to undestand how density plot is constructed:

- Probability density function (PDF): A function which outputs a specific value, $f(x)$ for a given observation o in a Variable V. The Variable V can have a set of values/observations O and can be called a sample space.

The interpretation of $f(x)$, is as follows: "*of all the values/observations O of the variable V, $f(x)$ is the relative likelihood (probability) of observing the observation (o)*".

The PDF satisfies the following properties:

- function is non-negative
- function is real-valued
- functions definite integral over the support dataset equals to 1

Some commonly known PDFs are uniform distribution, normal distributions, exponential distribution etc.

- Kernel: A kernel is a special type of probability density function (PDF) with the added property that it must be even. Thus, a kernel is a function with the following properties

Some common PDFs which satisfies the properties of kernel functions, as well, are uniform distribution, standard normal distributions.

Density plot, also know as **kernal density plot**, plots a smooth curve depicting the probability density function (PDF) or distribution of data. The peak of the curve helps in visulaizing where the values are concentrated in any given data.

To draw the distribution plot, we can use `geom_density` which computes and draws kernel density estimate.

Density plot for the outcome variable "SOLD PRICE".

```
ggplot(ipl_dt) +
geom_density(aes(x = SOLD_PRICE), fill = 'blue') +
ylab("Density") +
ggtitle("Density Plot of Sold Price") +
theme_bw() +
scale_x_continuous(labels =scales::dollar )
```

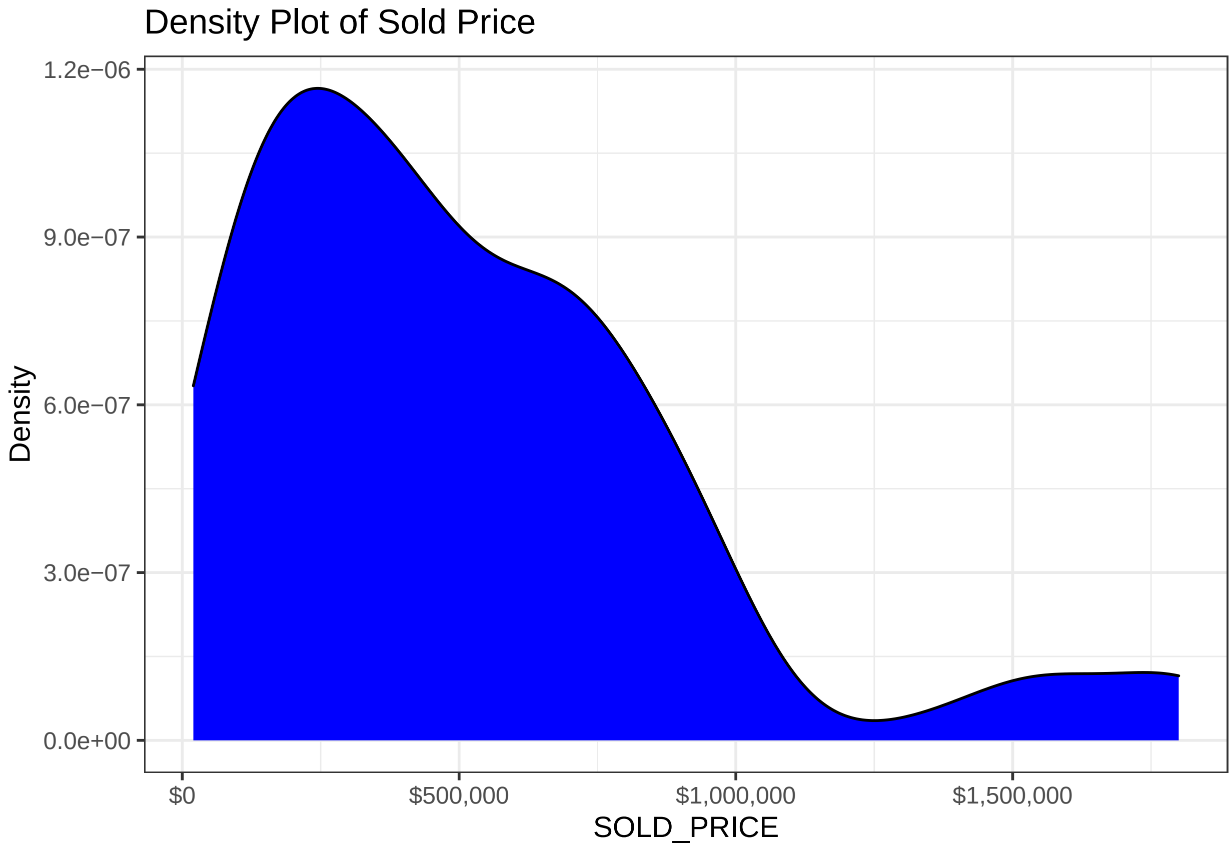

Figure 14: Density Plot 1

Figure <>, shows the density plot filled by `PLAYING_ROLE`

```
ggplot(ipl_dt) +
geom_density(aes(x = SOLD_PRICE, group=PLAYING_ROLE,fill=PLAYING_ROLE), position="stack") +
ylab("Density") +
ggtitle("Density Plot of Sold Price") +
theme_bw() +
scale_x_continuous(labels =scales::dollar )
```

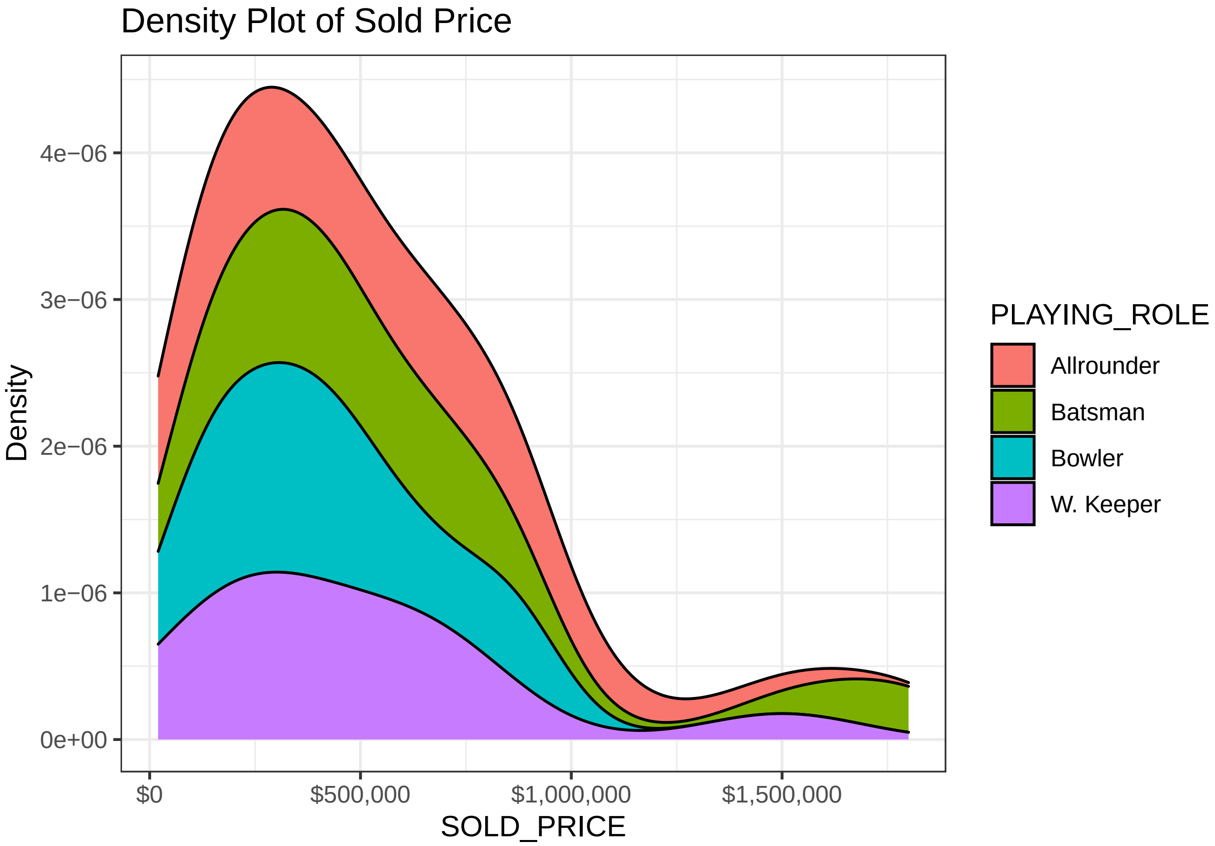

Figure 15: Density Plot 2

1.3.5.1 Exercise - Density Plot

Plot `density plot` for density plot for `SOLD_PRICE` and `BASE_PRICE` in the same graph. Explore the use of alpha as a aesthetic in `geom_density`.

1.3.6 Box Plot

Box plot is a graphical representation of numerical data that can be used to understand the variability of the data and the existence of outliers.

A boxplot is a graph that gives you a good indication of how the values in the data are spread out.

To generate a box plot: Assume data as : 98, 77, 85, 88, 82, 83, 87, 67, 100, 63, 105

- Arrange data in ascending order: 63, 67, 77, 82, 83, 85, 87, 88, 98, 100, 105
- Calculate the median (middle value of the data, 85). This is Q2
- Calculate the median of the first half of the data, 77). This is Q1.
- Calculate the median of the second half of the data, 98). This is Q3.
- The box joins Q1 to Q3 (contains middle 50% of data).
- IQR = Q3 - Q1 = 11
- LIF = Q1 - 1.5*IQR = 60.5 ; UIF = Q3 + 1.5 IQR = 114.5
- The point adjancent to LIF is 67 and point adjancent to UIF is 105.
- The smallest observation greater than or equal to `LIF` builds lower whisker.

- The largest observation less than or equal to UIF builds upper whisker.

  Point outside the fences are outliers.

Intrepret boxplot:

- If wide box and long whiskers, then maybe the data doesn't cluster.
- If box is small and the whiskers are short, then probably your data does indeed cluster
- If box is small and the whiskers are long, then maybe the data clusters, but have some "outliers"

The boxplot on ipl_dt can be created by using geom_boxplot.The outliers above the upper whisker is marked with black points.

```
ggplot(ipl_dt) +
geom_boxplot(aes(y = SOLD_PRICE), fill = 'blue') +
ylab("Sold Price") +
ggtitle("Box Plot for Sold Price") +
theme_bw() +
scale_y_continuous(labels =scales::dollar )
```

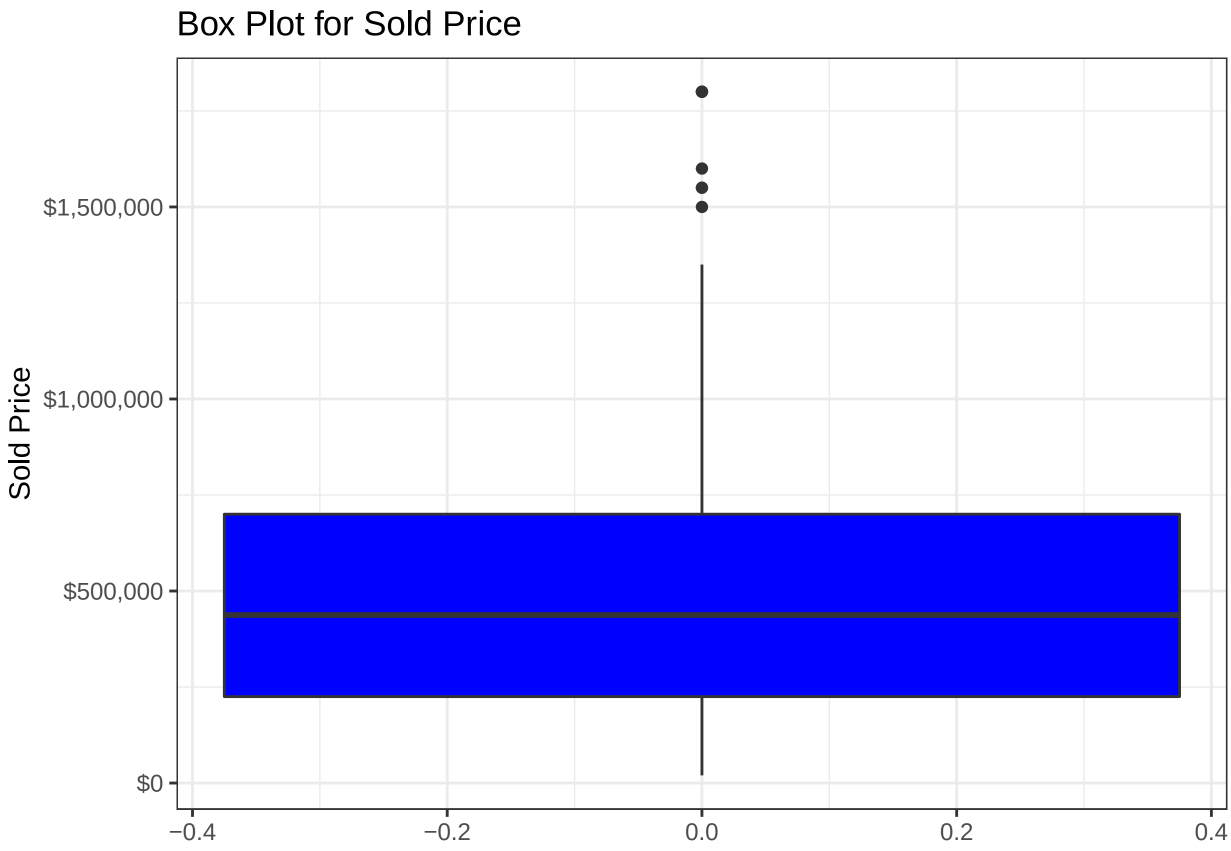

Figure 16: Box Plot 1

1.3.6.1 Exercise- Box Plot

Build a boxplot of SOLD_PRICE by PLAYING_ROLE. Is there any outlier?

1.3.7 Correlation and Heatmap

List of all numeric features in the ipl_dt

Table 5: Correlation Matrix

	T_RUNS	T_WKTS	ODI_RUNS_S	ODI_SR_B	ODI_WKTS
T_RUNS	1.00	0.03	0.89	0.23	0.05
T_WKTS	0.03	1.00	-0.09	0.01	0.82
ODI_RUNS_S	0.89	-0.09	1.00	0.32	0.06
ODI_SR_B	0.23	0.01	0.32	1.00	0.16
ODI_WKTS	0.05	0.82	0.06	0.16	1.00

```r
feature_name = colnames(ipl_dt)[!colnames(ipl_dt) %in% c('Sl_NO','AUCTION_YEAR')]
numeric_feature = feature_name[(sapply(ipl_dt[, ..feature_name], is.numeric))]
numeric_feature
```

```
##  [1] "T_RUNS"        "T_WKTS"        "ODI_RUNS_S"    "ODI_SR_B"
##  [5] "ODI_WKTS"      "ODI_SR_BL"     "CAPTAINCY_EXP" "RUNS_S"
##  [9] "HS"            "AVE_B"         "SR_B"          "SIXERS"
## [13] "RUNS_C"        "WKTS"          "AVE_BL"        "ECON"
## [17] "SR_BL"         "BASE_PRICE"    "SOLD_PRICE"    "Ln(soldp)"
```

Find correlation amongst numeric featrues using the `cor()` function.

```r
correlation_matrix = round(cor(ipl_dt[, ..numeric_feature]),2)
knitr::kable(correlation_matrix[1:5,1:5], caption = "Correlation Matrix")
```

`geom_tile()` can be used to generate a heatmap.

```r
ggplot(data = reshape2::melt(correlation_matrix, na.rm=TRUE)) +
geom_tile(mapping = aes(x=Var1, y=Var2, fill=value)) +
ggtitle("Correlation Matrix - Heat Map") +
ylab("") +
xlab("") +
theme_bw() +
theme(
    # x axis rotate
    axis.text.x=element_text(size=6, angle=90,hjust=0.9,vjust=0.1),
    axis.text.y=element_text(size=6,hjust=0.9,vjust=0.1)) +
scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1),
   name="Pearson Correlation") +
coord_fixed()
```

Table 7: Average Sold Price

AUCTION_YEAR	AVG_SOLD_PRICE
2009	458400
2008	492067
2011	604762
2010	290000

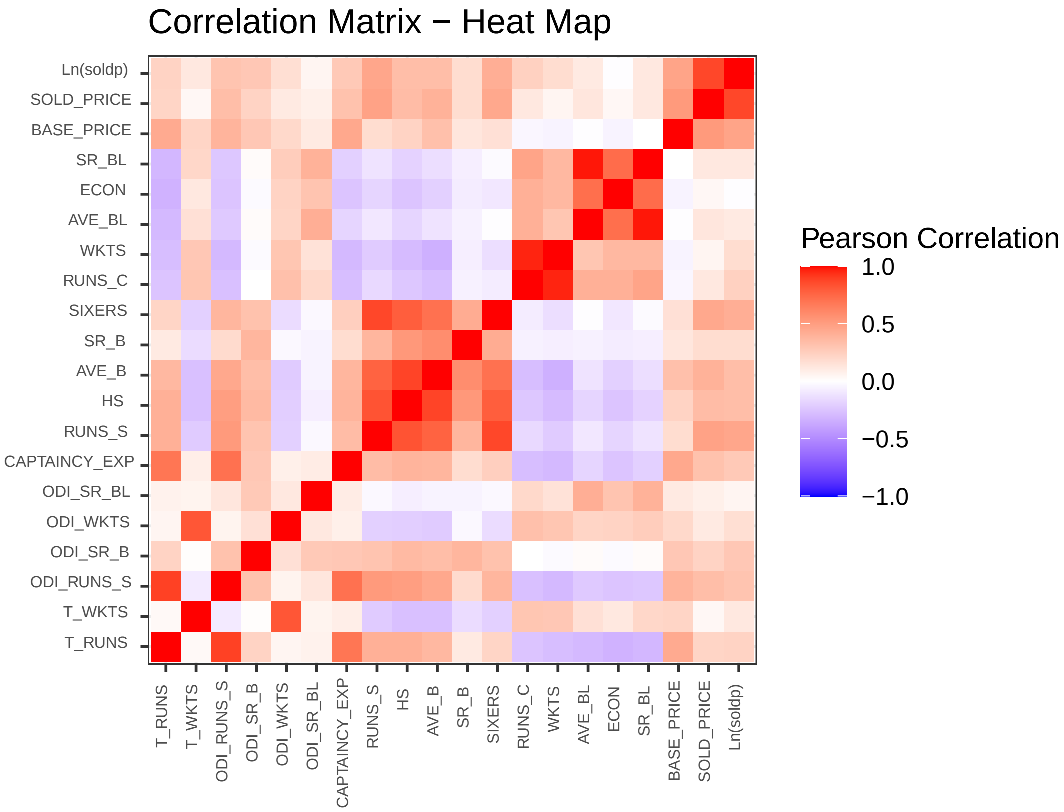

Figure 17: Heatmap 1

As the correlation coefficient ranges from -1 to 1, the argument `limit` of function scale_fill_gradient2 is set as `limit = c(-1,1)`. `coord_fixed()` ensures that one unit on the x-axis is the same length as one unit on the y-axis.

1.3.8 Trend plot

Let's plot the average `SOLD_PRICE` of players in different years in which auction was held. The number of players, in the `ipl_dt`, who were auctioned in different years:

2008	2009	2010	2011
75	10	3	42

```
ipl_sold_price = ipl_dt[, .(AVG_SOLD_PRICE = round(mean(SOLD_PRICE),0)), by=AUCTION_YEAR]
knitr::kable(ipl_sold_price, caption = "Average Sold Price")
```

We will use `geom_line()` which connects the observations in order of the variable on the x axis.

```
ggplot(data = ipl_sold_price) +
geom_line(mapping = aes(x=AUCTION_YEAR, y=AVG_SOLD_PRICE), colour = 'blue') +
geom_text(mapping = aes(x=AUCTION_YEAR, y=AVG_SOLD_PRICE, label=AVG_SOLD_PRICE) ,
          size = 2.5, hjust = 0.2, vjust=0.2) +
ylab("Sold Price") +
xlab("Year of Auction") +
ggtitle("Average Sold Price") +
theme_bw() +
scale_y_continuous(labels =scales::dollar )
```

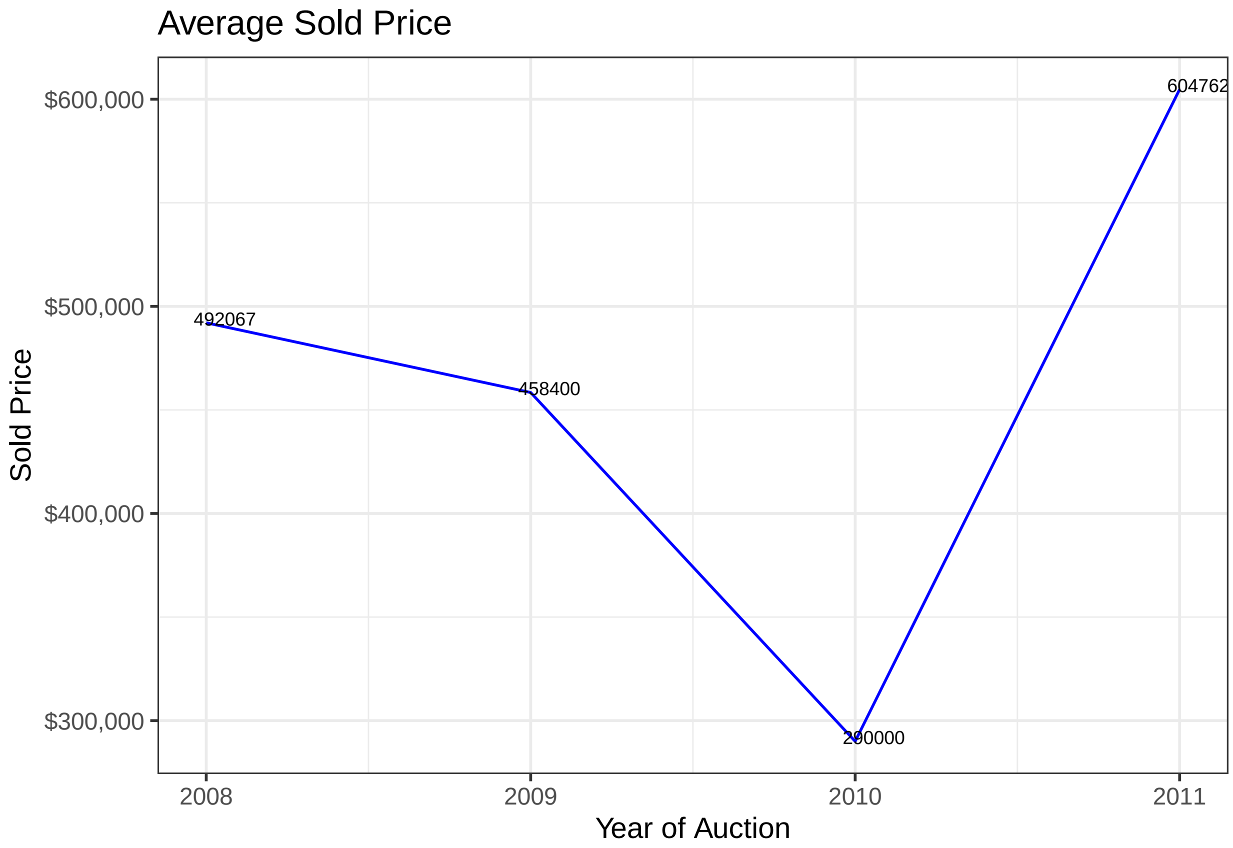

Figure 18: Trend Plot 1

1.3.8.1 Exercise - Trend plot

Represents the average sold price using geom_step. What difference do you see compared to the previous plot?

1.3.9 Lay out panels in a grid

`facet_grid()` is a useful annotation from `ggplot`.

Creating a plot with faceting variable as `AGE` and `PLAYING_ROLE` , means that we want to visulaize the plot with `AGE` and `PLAYING_ROLE` as two more dimensions of the data.

`facet_grid()`, forms a matrix of panels defined by row and column faceting variables. It is most useful when there are two discrete variables in the data and all combinations of the variables exist in the data.

A box plot create a facet grid with AGE and PLAYING_ROLE as faceting variable.

```
ggplot(ipl_dt) +
geom_boxplot(aes(y = SOLD_PRICE)) +
ylab("Sold Price") +
ggtitle("Box Plot for Sold Price") +
theme_bw() +
theme(plot.title = element_text(hjust= 0.5)) +
scale_y_continuous(labels =scales::dollar ) +
facet_grid(rows = vars(AGE),  cols = vars(PLAYING_ROLE))
```

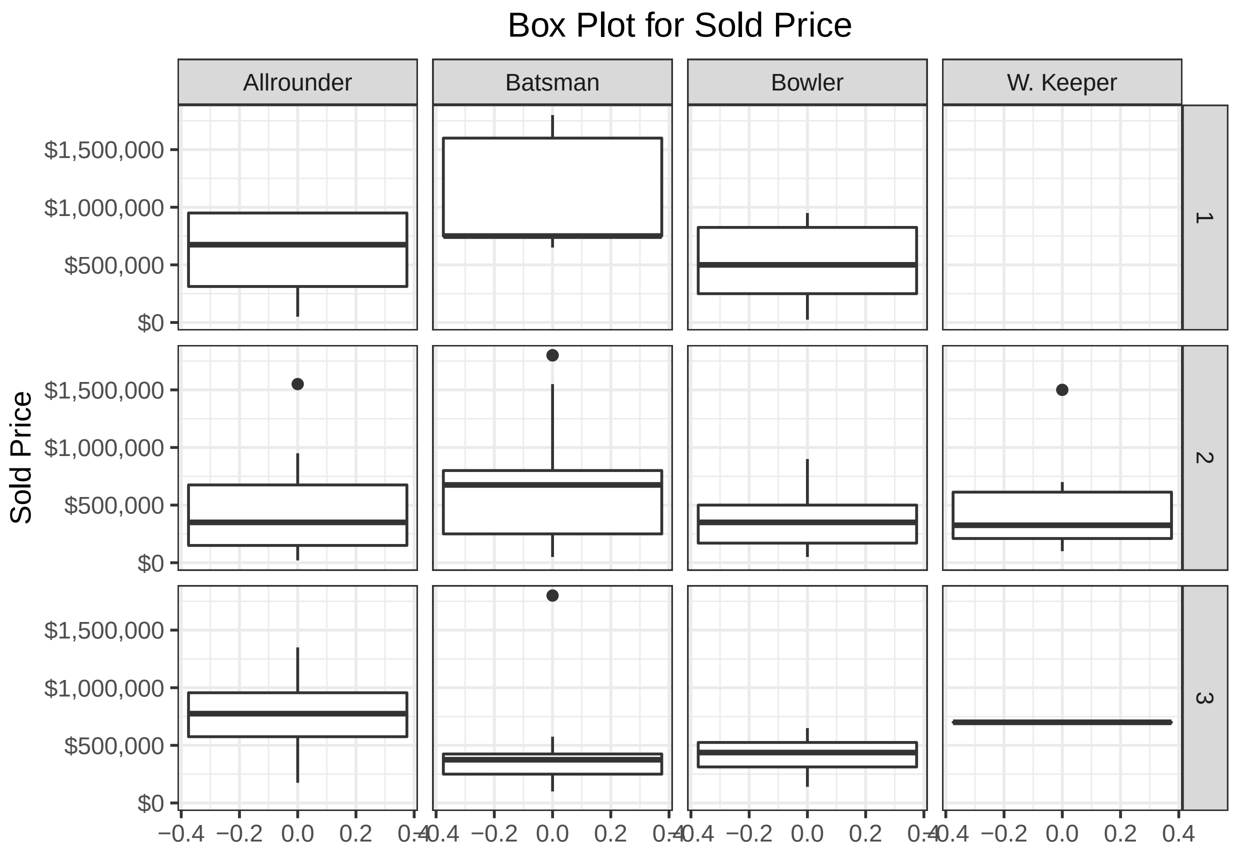

Figure 19: Boxplot with Facet grid

1.3.9.1 Exercise - Facet grid

Scatter plot and facet it with AGE and PLAYING_ROLE.

1.4 Thank you

Practice on the exercises given and redo all the example codes given here using R.