

Name: **Varun Sudhir**

Reg No: **21BDSoo40**

Exploratory Data Analysis Assignment

First,we will import the dataset and display it

```
dataset = read.csv("C:/Users/Varun/Downloads/DS3_Match.csv")  
View(dataset)
```

	Match_Id	Match_Date	Team_Name_Id	Opponent_Team_Id	Season_Id	Venue_Name	Toss_Winner_Id	Toss_Decision	IS_Superover	IS_Result	Is_DuckWorthLewis	Win_Type	Won_By
1	335987	18-Apr-08		2	1	M Chinnaswamy Stadium		2 field	0	1		0 by runs	140
2	335988	19-Apr-08		4	3	Punjab Cricket Association Stadium, Mohali		3 bat	0	1		0 by runs	33
3	335989	19-Apr-08		6	5	Feroz Shah Kotla		5 bat	0	1		0 by wickets	9
4	335990	20-Apr-08		7	2	Wankhede Stadium		7 bat	0	1		0 by wickets	5
5	335991	20-Apr-08		1	8	Eden Gardens		8 bat	0	1		0 by wickets	5
6	335992	21-Apr-08		5	4	Sawai Mansingh Stadium		4 bat	0	1		0 by wickets	6
7	335993	22-Apr-08		8	6	Rajiv Gandhi International Stadium, Uppal		8 bat	0	1		0 by wickets	9
8	335994	23-Apr-08		3	7	MA Chidambaram Stadium, Chepauk		7 field	0	1		0 by runs	6
9	335995	24-Apr-08		8	5	Rajiv Gandhi International Stadium, Uppal		5 field	0	1		0 by wickets	3
10	335996	25-Apr-08		4	7	Punjab Cricket Association Stadium, Mohali		7 field	0	1		0 by runs	66
11	335997	26-Apr-08		2	5	M Chinnaswamy Stadium		5 field	0	1		0 by wickets	7
12	335998	26-Apr-08		3	1	MA Chidambaram Stadium, Chepauk		1 bat	0	1		0 by wickets	9
13	335999	27-Apr-08		7	8	Dr DY Patil Sports Academy		8 field	0	1		0 by wickets	10
14	336000	27-Apr-08		4	6	Punjab Cricket Association Stadium, Mohali		6 bat	0	1		0 by wickets	4
15	336001	28-Apr-08		2	3	M Chinnaswamy Stadium		3 bat	0	1		0 by runs	13
16	336002	29-Apr-08		1	7	Eden Gardens		1 bat	0	1		0 by wickets	7
17	336003	30-Apr-08		6	2	Feroz Shah Kotla		2 field	0	1		0 by runs	10
18	336004	01-May-08		8	4	Rajiv Gandhi International Stadium, Uppal		4 field	0	1		0 by wickets	7
19	336005	01-May-08		5	1	Sawai Mansingh Stadium		5 bat	0	1		0 by runs	45
20	336006	02-May-08		3	6	MA Chidambaram Stadium, Chepauk		3 bat	0	1		0 by wickets	8
21	336007	25-May-08		8	2	Rajiv Gandhi International Stadium, Uppal		8 bat	0	1		0 by wickets	5
22	336008	03-May-08		4	1	Punjab Cricket Association Stadium, Mohali		4 bat	0	1		0 by runs	9
23	336009	04-May-08		7	6	Dr DY Patil Sports Academy		6 field	0	1		0 by runs	29
24	336010	04-May-08		5	3	Sawai Mansingh Stadium		3 bat	0	1		0 by wickets	8
25	336011	05-May-08		2	4	M Chinnaswamy Stadium		4 field	0	1		0 by wickets	6

Won_By	Match_Winner_Id	Man_Of_The_Match_Id	First_Umpire_Id	Second_Umpire_Id	City_Name	Host_Country
140	1	2	470	477	Bangalore	India
33	3	19	471	487	Chandigarh	India
9	6	90	472	512	Delhi	India
5	2	11	473	476	Mumbai	India
5	1	4	474	486	Kolkata	India
6	5	32	472	513	Jaipur	India
9	6	41	475	492	Hyderabad	India
6	3	18	476	512	Chennai	India
3	5	31	470	471	Hyderabad	India
66	4	26	472	492	Chandigarh	India
7	5	32	471	475	Bangalore	India
9	3	22	474	479	Chennai	India
10	8	53	470	487	Mumbai	India
4	4	28	477	514	Chandigarh	India
13	3	20	478	513	Bangalore	India
7	7	44	474	479	Kolkata	India
10	6	118	472	514	Delhi	India
7	4	100	478	513	Hyderabad	India
45	5	101	477	512	Jaipur	India
8	6	41	474	486	Chennai	India
5	2	81	470	477	Hyderabad	India
9	4	29	476	514	Chandigarh	India
29	7	49	475	477	Mumbai	India
8	5	102	470	479	Jaipur	India
6	4	77	473	478	Bangalore	India

Let's perform some basic operations to know more about the dataset and its features

```
# Descriptive operations to know more about the dataset
```

```
# Varun Sudhir 21BDS0040
```

```
# 1) Summary statistics for the dataset
```

```
summary(dataset)
```

```
print("Varun Sudhir 21BDS0040")
```

Output:

```
> summary(dataset)
  Match_Id      Match_Date      Team_Name_Id      Opponent_Team_Id      Season_Id      Venue_Name      Toss_Winner_Id      Toss_Decision      IS_Superover      IS_Result
Min.   :335987  Length:577  Min.    : 1.000  Min.    : 1.000  Min.   :1.000  Length:577  Min.    : 1.000  Length:577  Min.   :0.0000  Min.   :0.0000
1st Qu.:419140  Class :character 1st Qu.: 3.000 1st Qu.: 3.000 1st Qu.:3.000  Class :character 1st Qu.: 3.000  Class :character 1st Qu.:0.0000 1st Qu.:1.0000
Median :548353  Mode  :character Median : 5.000 Median : 5.000 Median :5.000  Mode  :character Median : 5.000  Mode  :character Median :0.0000 Median :1.0000
Mean   :591636      Mean  : 5.102      Mean  : 5.211      Mean  :5.029      Mean  : 5.192      Mean  :0.0104      Mean  :0.9948
3rd Qu.:734004      3rd Qu.: 7.000      3rd Qu.: 7.000      3rd Qu.:7.000      3rd Qu.: 7.000      3rd Qu.:0.0000      3rd Qu.:1.0000
Max.   :981024      Max.   :13.000      Max.   :13.000      Max.   :9.000      Max.   :13.000      Max.   :1.0000      Max.   :1.0000

Is_DuckworthLewis  Win_Type      Won_By      Match_Winner_Id  Man_Of_The_Match_Id  First_Umpire_Id  Second_Umpire_Id  City_Name      Host_Country
Min.   :0.000      Length:577  Length:577  Min.   : 1.000  Min.   : 1.0  Min.   :470.0  Min.   :471.0  Length:577  Length:577
1st Qu.:0.000      Class :character  Class :character 1st Qu.: 3.000 1st Qu.: 40.0 1st Qu.:475.0 1st Qu.:488.0  Class :character  Class :character
Median :0.000      Mode  :character  Mode  :character Median : 5.000 Median :105.5 Median :482.0 Median :493.0  Mode  :character  Mode  :character
Mean   :0.026      Mean  : 4.991      Mean  :139.8      Mean  :484.1      Mean  :495.2
3rd Qu.:0.000      3rd Qu.: 7.000      3rd Qu.:209.5      3rd Qu.:493.0      3rd Qu.:500.0
Max.   :1.000      Max.   :13.000      Max.   :460.0      Max.   :511.0      Max.   :521.0
NA's   :3          NA's   :3

> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```

```
# 2) Structure of the dataset
```

```
str(dataset)
```

```
print("Varun Sudhir 21BDS0040")
```

Output:

```
> # 2) Structure of the dataset
> str(dataset)
'data.frame': 577 obs. of 19 variables:
 $ Match_Id      : int  335987 335988 335989 335990 335991 335992 335993 335994 335995 335996 ...
 $ Match_Date    : chr   "18-Apr-08" "19-Apr-08" "19-Apr-08" "20-Apr-08" ...
 $ Team_Name_Id  : int   2 4 6 7 1 5 8 3 8 4 ...
 $ Opponent_Team_Id : int   1 3 5 2 8 4 6 7 5 7 ...
 $ Season_Id     : int   1 1 1 1 1 1 1 1 1 1 ...
 $ Venue_Name    : chr    "M Chinnaswamy Stadium" "Punjab Cricket Association Stadium, Mohali" "Feroz Shah Kotla" "Wankhede Stadium" ...
 $ Toss_Winner_Id : int   2 3 5 7 8 4 8 7 5 7 ...
 $ Toss_Decision  : chr    "field" "bat" "bat" "bat" ...
 $ IS_Superover   : int    0 0 0 0 0 0 0 0 0 ...
 $ IS_Result      : int    1 1 1 1 1 1 1 1 1 ...
 $ Is_DuckworthLewis : int    0 0 0 0 0 0 0 0 0 ...
 $ Win_Type       : chr    "by runs" "by runs" "by wickets" "by wickets" ...
 $ Won_By         : chr    "140" "33" "9" "5" ...
 $ Match_Winner_Id : int    1 3 6 2 1 5 6 3 5 4 ...
 $ Man_Of_The_Match_Id : int   2 19 90 11 4 32 41 18 31 26 ...
 $ First_Umpire_Id : int   470 471 472 473 474 472 475 476 470 472 ...
 $ Second_Umpire_Id : int   477 487 512 476 486 513 492 512 471 492 ...
 $ City_Name      : chr    "Bangalore" "Chandigarh" "Delhi" "Mumbai" ...
 $ Host_Country   : chr    "India" "India" "India" "India" ...

> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```

```
# 3) Number of unique values in each column
sapply(dataset, function(x) length(unique(x)))
print("Varun Sudhir 21BDS0040")
```

```
> # 3) Number of unique values in each column
> sapply(dataset, function(x) length(unique(x)))
      Match_Id      Match_Date      Team_Name_Id      Opponent_Team_Id      Season_Id      Venue_Name      Toss_Winner_Id      Toss_Decision      IS_Superover
      577          407          13          13          9          35          13          2          2
      IS_Result      Is_DuckworthLewis      Win_Type      Won_By      Match_Winner_Id      Man_Of_The_Match_Id      First_Umpire_Id      Second_Umpire_Id      City_Name
      2          2          4          83          14          188          42          44          29
      Host_Country
      3
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
>
```

Analysing the dataset to find redundant and missing values

```
# Varun Sudhir 21BDS0040
# Finding redundant values, missing values, outliers etc
```

```
# Check for duplicate rows
duplicate_rows <- dataset %>%
  filter(duplicated(.))
print("Duplicate rows in the dataset:")
print(duplicate_rows)
print("Varun Sudhir 21BDS0040")
```

Output:

```
> # Check for duplicate rows
> duplicate_rows <- dataset %>%
+   filter(duplicated(.))
> print("Duplicate rows in the dataset:")
[1] "Duplicate rows in the dataset:"
> print(duplicate_rows)
[1] Match_Id      Match_Date      Team_Name_Id      Opponent_Team_Id      Season_Id      Venue_Name      Toss_Winner_Id      Toss_Decision
[9] IS_Superover      IS_Result      Is_DuckworthLewis      Win_Type      Won_By      Match_Winner_Id      Man_Of_The_Match_Id      First_Umpire_Id
[17] Second_Umpire_Id      City_Name      Host_Country
<0 rows> (or 0-length row.names)
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
>
```

As we can see, there are no duplicate records in our dataset

```
# Check for missing values
missing_values <- sapply(dataset, function(x) sum(is.na(x)))
print("Missing values in each column:")
print(missing_values)
print("Varun Sudhir 21BDS0040")
```

Output:

```
> # Check for missing values
> missing_values <- sapply(dataset, function(x) sum(is.na(x)))
> print("Missing values in each column:")
[1] "Missing values in each column:"
> print(missing_values)
      Match_Id      Match_Date      Team_Name_Id      Opponent_Team_Id      Season_Id      Venue_Name      Toss_Winner_Id      Toss_Decision      IS_Superover
      0          0          0          0          0          0          0          0          0
IS_Result      Is_DuckworthLewis      Win_Type      Won_By      Match_Winner_Id      Man_Of_The_Match_Id      First_Umpire_Id      Second_Umpire_Id      City_Name
      0          0          0          0          3          3          0          0          0
Host_Country
      0
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```

We can see that few of the columns of the dataset have na values

Removing the missing values of the dataset

Function to impute mode for character columns

```
impute_mode <- function(x) {
  uniqv <- unique(x)
  uniqv[which.max(tabulate(match(x, uniqv)))]
}
```

Impute missing values for numeric columns

```
dataset <- dataset %>%
  mutate_if(is.numeric, ~ ifelse(is.na(.), mean(., na.rm = TRUE), .))
```

Impute missing values for character columns

```
dataset <- dataset %>%
  mutate_if(is.character, ~ ifelse(is.na(.), impute_mode(.), .))
```

Check for any remaining missing values

```
missing_values <- sapply(dataset, function(x) sum(is.na(x)))
print("Number of missing values in each column after imputation:")
print(missing_values)
print("Varun Sudhir 21BDS0040")
```

Output:

```
[1] "Number of missing values in each column after imputation:"
> print(missing_values)
      Match_Id      Match_Date      Team_Name_Id      Opponent_Team_Id      Season_Id      Venue_Name      Toss_Winner_Id      Toss_Decision      IS_Superover
      0          0          0          0          0          0          0          0          0
IS_Result      Is_DuckworthLewis      Win_Type      Won_By      Match_Winner_Id      Man_Of_The_Match_Id      First_Umpire_Id      Second_Umpire_Id      City_Name
      0          0          0          0          0          0          0          0          0
Host_Country
      0
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
> |
```

We can see there are no longer any missing values in the dataset

Now, let's perform some operations on the dataset

```
# Operations on the dataset
# Varun Sudhir 21BDS0040

# Frequency distribution of 'Match_Winner_Id'
match_winner_freq <- dataset %>%
  count(Match_Winner_Id, sort = TRUE)
print(match_winner_freq)
print("Varun Sudhir 21BDS0040")
```

Output:

```
> print(match_winner_freq)
  Match_Winner_Id  n
1                7 80
2                3 79
3                2 70
4                1 68
5                4 63
6                5 63
7                6 56
8               11 34
9                8 29
10               10 12
11               13  9
12                9  6
13               12  5
14               NA  3
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```

```
# Average and median of 'Team_name_Id' column
mean_won_by <- mean(dataset$Team_Name_Id, na.rm = TRUE)
median_won_by <- median(dataset$Team_Name_Id, na.rm = TRUE)
print(paste("Mean of Team_name_Id: ", mean_won_by))
print(paste("Median of Team_name_Id: ", median_won_by))
print("Varun Sudhir 21BDS0040")
```

Output:

```
> print(paste("Mean of Team_name_Id: ", mean_won_by))
[1] "Mean of Team_name_Id:  5.10225303292894"
> print(paste("Median of Team_name_Id: ", median_won_by))
[1] "Median of Team_name_Id:  5"
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```

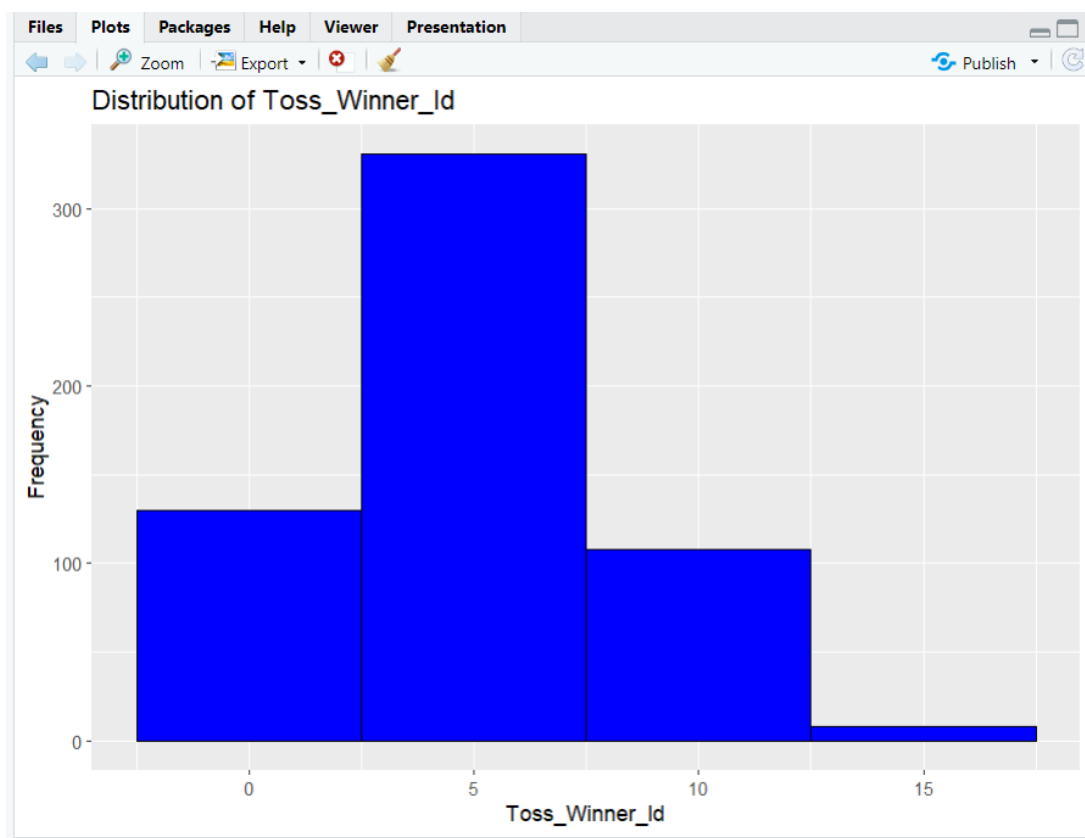
```
# Standard deviation and variance of 'Man_Of_The_Match_Id' column
sd_won_by <- sd(dataset$Man_Of_The_Match_Id, na.rm = TRUE)
var_won_by <- var(dataset$Man_Of_The_Match_Id, na.rm = TRUE)
print(paste("Standard Deviation of Man_Of_The_Match_Id: ", sd_won_by))
print(paste("Variance of Man_Of_The_Match_Id: ", var_won_by))
print("Varun Sudhir 21BDS0040")
```

Output:

```
> print(paste("Standard Deviation of Man_Of_The_Match_Id: ", sd_won_by))
[1] "Standard Deviation of Man_Of_The_Match_Id: 117.304943952277"
> print(paste("Variance of Man_Of_The_Match_Id: ", var_won_by))
[1] "Variance of Man_Of_The_Match_Id: 13760.4498756468"
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```

```
# Distribution of 'Toss_Winner_Id' column
ggplot(dataset, aes(x = Toss_Winner_Id)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") +
  labs(title = "Distribution of Toss_Winner_Id", x = "Won_By", y =
"Frequency")
```

Output:



```

# Convert 'Won_By' to integer (assuming it contains numeric values in
character format)
dataset$Won_By <- as.integer(dataset$Won_By)

# Group by 'Match_Winner' and calculate the average 'Won_By'
average_won_by <- dataset %>%
  group_by(Match_Winner) %>%
  summarize(avg_won_by = mean(Won_By, na.rm = TRUE))
print(average_won_by)
print("Varun Sudhir 21BDS0040")

```

Output:

```

> print(average_won_by)
# A tibble: 14 × 2
  Match_Winner_Id avg_won_by
      <int>         <dbl>
1           1         15.3
2           2         20.7
3           3         22.7
4           4         16.4
5           5         17.3
6           6         11.9
7           7         19.9
8           8          17
9           9          8.83
10          10         14.6
11          11         13.3
12          12         14.6
13          13          4.67
14          NA          NA
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"

```

```

# Arrange the dataset by Match_Number
arranged_dataset <- dataset %>%
  arrange(Match_Id)

# Display the first few rows of the arranged dataset
print(head(arranged_dataset))
print("Varun Sudhir 21BDS0040")

```

Output:

```
> # Display the first few rows of the arranged dataset
> print(head(arranged_dataset))
  Match_Id Match_Date Team_Name_Id Opponent_Team_Id Season_Id Venue_Name
1 335987 18-Apr-08      2          1          1 M Chinnaswamy Stadium
2 335988 19-Apr-08      4          3          1 Punjab Cricket Association Stadium, Mohali
3 335989 19-Apr-08      6          5          1 Feroz Shah Kotla
4 335990 20-Apr-08      7          2          1 Wankhede Stadium
5 335991 20-Apr-08      1          8          1 Eden Gardens
6 335992 21-Apr-08      5          4          1 Sawai Mansingh Stadium
  Toss_Winner_Id Toss_Decision IS_Superover IS_Result Is_DuckworthLewis Win_Type Won_By Match_Winner_Id
1              2          field            0          1              0 by runs    140          1
2              3            bat            0          1              0 by runs     33          3
3              5            bat            0          1              0 by wickets    9          6
4              7            bat            0          1              0 by wickets    5          2
5              8            bat            0          1              0 by wickets    5          1
6              4            bat            0          1              0 by wickets    6          5
  Man_Of_The_Match_Id First_Umpire_Id Second_Umpire_Id City_Name Host_Country
1                  2              470              477 Bangalore      India
2                  19              471              487 Chandigarh    India
3                  90              472              512 Delhi          India
4                  11              473              476 Mumbai          India
5                   4              474              486 Kolkata          India
6                  32              472              513 Jaipur          India
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```

Finding observations where the match is hosted in India

```
filtered_dataset <- dataset %>%
  filter(Host_Country == "India")
```

```
# Display the filtered dataset
print(head(filtered_dataset))
print("Varun Sudhir 21BDS0040")
```

Output:

```
> print(head(filtered_dataset))
  Match_Id Match_Date Team_Name_Id Opponent_Team_Id Season_Id Venue_Name
1 335987 18-Apr-08      2          1          1 M Chinnaswamy Stadium
2 335988 19-Apr-08      4          3          1 Punjab Cricket Association Stadium, Mohali
3 335989 19-Apr-08      6          5          1 Feroz Shah Kotla
4 335990 20-Apr-08      7          2          1 Wankhede Stadium
5 335991 20-Apr-08      1          8          1 Eden Gardens
6 335992 21-Apr-08      5          4          1 Sawai Mansingh Stadium
  Toss_Winner_Id Toss_Decision IS_Superover IS_Result Is_DuckworthLewis Win_Type Won_By Match_Winner_Id
1              2          field            0          1              0 by runs    140          1
2              3            bat            0          1              0 by runs     33          3
3              5            bat            0          1              0 by wickets    9          6
4              7            bat            0          1              0 by wickets    5          2
5              8            bat            0          1              0 by wickets    5          1
6              4            bat            0          1              0 by wickets    6          5
  Man_Of_The_Match_Id First_Umpire_Id Second_Umpire_Id City_Name Host_Country
1                  2              470              477 Bangalore      India
2                  19              471              487 Chandigarh    India
3                  90              472              512 Delhi          India
4                  11              473              476 Mumbai          India
5                   4              474              486 Kolkata          India
6                  32              472              513 Jaipur          India
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
```


Normalization of the columns

```
# Normalizing the columns
# Varun Sudhir 21BDS0040

numeric_column <- "Won_By"

# Function to normalize a column [ 0, 1 ] range
normalize <- function(x) {
  return((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm =
TRUE)))
}

# Normalize the chosen column and add it as a new column
dataset[[paste0(numeric_column, "_Normalized")]] <-
normalize(dataset[[numeric_column]])
print(dataset$Won_By_Normalized)
print("Varun Sudhir 21BDS0040")
```

Output:

```
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
> print(dataset$Won_By_Normalized)
[1] 0.972027972 0.223776224 0.055944056 0.027972028 0.027972028 0.034965035 0.055944056 0.034965035 0.013986014
[10] 0.454545455 0.041958042 0.055944056 0.062937063 0.020979021 0.083916084 0.041958042 0.062937063 0.041958042
[19] 0.307692308 0.048951049 0.027972028 0.055944056 0.195804196 0.048951049 0.034965035 0.041958042 0.041958042
[28] 0.020979021 0.027972028 0.048951049 0.055944056 0.118881119 0.153846154 0.013986014 0.055944056 0.153846154
[37] 0.055944056 0.279720280 0.076923077 0.048951049 0.034965035 0.447552448 0.167832168 0.013986014 0.027972028
[46] 0.034965035 0.000000000 0.090909091 0.034965035 0.027972028 0.062937063 0.013986014 0.013986014 0.027972028
[55] 0.041958042 0.727272727 0.055944056 0.013986014 0.125874126 0.517482517 0.062937063 0.048951049 0.636363636
[64] 0.069930070 0.160839161 0.055944056 NA 0.041958042 0.076923077 0.034965035 0.181818182 0.034965035
[73] 0.636363636 0.027972028 0.027972028 0.013986014 0.034965035 0.258741259 0.055944056 0.048951049 0.013986014
[82] 0.118881119 0.034965035 0.055944056 0.538461538 0.538461538 0.055944056 0.125874126 0.041958042 0.076923077
[91] 0.041958042 0.013986014 0.041958042 0.104895105 0.041958042 0.363636364 0.034965035 0.048951049 0.076923077
[100] 0.006993007 0.006993007 0.034965035 0.041958042 0.034965035 0.000000000 0.090909091 0.041958042 0.041958042
[109] 0.020979021 0.160839161 0.020979021 0.076923077 0.034965035 0.034965035 0.034965035 0.069930070 0.020979021
[118] 0.027972028 0.041958042 0.209790210 0.034965035 0.048951049 0.377622378 0.678321678 0.062937063 0.027972028
[127] 0.034965035 0.230769231 0.041958042 0.062937063 NA 0.041958042 0.244755245 0.209790210 0.027972028
[136] 0.048951049 0.265734266 0.111888112 0.111888112 0.279720280 0.272727273 0.020979021 0.027972028 0.461538462
[145] 0.160839161 0.034965035 0.153846154 0.433566434 0.048951049 0.251748252 0.006993007 0.160839161 0.055944056
[154] 0.090909091 0.041958042 0.034965035 0.034965035 0.041958042 0.041958042 0.251748252 0.083916084 0.265734266
[163] 0.055944056 0.027972028 0.034965035 0.027972028 0.391608392 0.048951049 0.034965035 0.069930070 0.055944056
[172] 0.237762238 0.258741259 0.055944056 0.146853147 0.006993007 0.048951049 0.034965035 0.048951049 0.041958042
[181] 0.055944056 0.034965035 0.055944056 0.034965035 0.020979021 0.223776224 0.055944056 0.048951049 0.139860140
[190] 0.048951049 0.013986014 0.048951049 0.041958042 0.104895105 0.041958042 0.034965035 0.328671329 0.048951049
[199] 0.055944056 0.195804196 0.251748252 0.048951049 0.167832168 0.013986014 0.048951049 0.377622378 0.111888112
[208] 0.041958042 0.174825175 0.258741259 0.048951049 0.034965035 0.125874126 0.153846154 0.041958042 0.132867133
[217] 0.048951049 0.139860140 0.111888112 0.020979021 0.587412587 0.062937063 0.216783217 0.055944056 0.027972028
[226] 0.433566434 0.034965035 0.524475524 0.055944056 0.118881119 0.034965035 0.020979021 0.062937063 0.195804196
[235] 0.048951049 0.034965035 0.769230769 0.069930070 0.041958042 0.062937063 0.566433566 NA 0.048951049
[244] 0.027972028 0.034965035 0.020979021 0.293706294 0.398601399 0.048951049 0.048951049 0.188811189 0.209790210
```

```
# Normalize data between -1 and 1
# 21BDS0040 Varun Sudhir
```

```
normalize_range <- function(x) {
```

```

2 * ((x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm =
TRUE))) - 1
}

# Apply normalization to the 'Team_Name_Id' column
dataset <- dataset %>%
  mutate(Team_Number_normalized = normalize_range(Team_Name_Id))

print(dataset$Team_Number_normalized)
print("Varun Sudhir 21BDS0040")

```

Output:

```

[291] -0.8333333  0.5000000 -0.6666667 -1.0000000 -0.5000000  0.0000000 -0.8333333 -0.1666667  0.5000000  0.1666667
[301]  0.0000000 -0.3333333  0.5000000 -1.0000000 -0.6666667 -0.3333333 -0.5000000 -0.8333333 -1.0000000 -0.1666667
[311]  0.0000000 -0.5000000 -0.1666667  0.1666667 -0.5000000  0.5000000  0.1666667 -0.3333333 -0.1666667 -0.6666667
[321] -0.1666667 -1.0000000 -1.0000000 -0.8333333  0.6666667 -0.1666667 -0.6666667  0.5000000  0.6666667 -0.3333333
[331]  0.0000000 -0.5000000 -0.8333333  0.5000000 -0.1666667  0.0000000 -0.6666667 -1.0000000 -0.3333333 -0.6666667
[341] -0.5000000 -0.8333333  0.5000000 -0.3333333 -0.1666667  0.6666667 -1.0000000 -0.8333333 -0.1666667 -0.5000000
[351] -0.6666667 -0.8333333 -0.5000000 -1.0000000 -0.6666667 -1.0000000 -0.3333333  0.0000000 -0.6666667 -0.1666667
[361] -0.3333333  0.0000000  0.5000000  0.6666667 -0.1666667 -0.6666667  0.5000000 -1.0000000  0.6666667 -0.8333333
[371]  0.0000000 -0.3333333 -0.8333333 -0.3333333  0.0000000  0.6666667 -0.5000000  0.5000000 -0.1666667  0.5000000
[381] -0.5000000 -1.0000000 -0.3333333 -0.1666667  0.0000000 -1.0000000 -0.6666667  0.0000000 -0.5000000  0.6666667
[391] -0.5000000  0.5000000 -0.8333333  0.6666667 -0.6666667 -0.3333333  0.0000000 -0.6666667  0.0000000 -0.1666667
[401] -0.6666667  0.6666667 -0.8333333 -1.0000000 -0.3333333 -0.6666667 -0.5000000 -0.3333333 -0.8333333  0.6666667
[411] -0.6666667 -0.3333333 -1.0000000 -0.1666667  0.6666667 -0.5000000 -1.0000000  0.0000000 -0.6666667  0.0000000
[421] -0.1666667 -0.8333333 -0.3333333 -0.1666667  0.0000000 -0.1666667 -0.5000000 -0.3333333 -0.8333333 -0.1666667
[431]  0.0000000 -0.5000000 -0.8333333  0.6666667 -0.6666667 -0.8333333  0.6666667 -1.0000000 -0.3333333 -0.6666667
[441]  0.6666667 -0.3333333 -0.1666667  0.6666667 -1.0000000 -0.5000000 -1.0000000 -0.6666667  0.0000000 -0.5000000
[451] -0.8333333 -1.0000000 -0.5000000  0.0000000 -0.5000000 -0.6666667 -0.6666667 -1.0000000 -1.0000000 -0.6666667
[461] -0.5000000 -0.6666667 -1.0000000 -0.1666667  0.0000000 -0.8333333 -0.3333333 -1.0000000 -0.5000000  0.6666667
[471]  0.0000000  0.6666667 -0.5000000 -0.3333333 -0.8333333 -0.1666667 -0.3333333  0.6666667 -0.8333333 -0.1666667
[481] -0.3333333  0.0000000 -0.6666667 -0.1666667 -0.5000000 -1.0000000 -0.8333333 -0.6666667 -0.1666667  0.0000000
[491] -0.8333333  0.6666667 -0.5000000 -0.3333333 -0.6666667 -1.0000000  0.0000000 -0.8333333 -0.3333333 -0.6666667
[501] -1.0000000 -0.1666667  0.0000000 -0.6666667  0.6666667 -0.1666667 -0.5000000  0.0000000  0.6666667 -0.5000000
[511] -0.3333333 -0.8333333  0.6666667 -0.6666667 -0.8333333 -0.6666667  0.0000000  0.0000000 -1.0000000 -0.5000000
[521] -0.8333333 -1.0000000  1.0000000 -0.1666667  0.6666667  0.0000000 -0.5000000 -0.8333333  0.6666667 -0.5000000
[531]  0.0000000  1.0000000  0.8333333 -0.1666667  0.6666667  1.0000000  0.8333333 -0.5000000  0.6666667 -0.1666667
[541]  0.0000000  0.8333333 -0.1666667  0.6666667  1.0000000  0.8333333 -0.8333333  1.0000000 -1.0000000 -0.1666667
[551]  0.6666667 -0.8333333 -0.5000000  0.0000000 -1.0000000 -0.5000000  0.8333333 -0.8333333  0.6666667  0.0000000
[561] -0.8333333 -1.0000000 -0.5000000  0.0000000 -1.0000000  0.8333333 -0.8333333  1.0000000 -0.1666667  0.8333333
[571]  1.0000000 -1.0000000 -0.1666667  1.0000000  0.6666667  1.0000000 -0.8333333
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"

```

Finding the range of columns

```

# Varun Sudhir 21BDS0040
# Calculating the range

# Finding the range of Won-By runs
print(max(dataset$Won_By, na.rm=TRUE) - min(dataset$Won_By, na.rm=TRUE))
print("Varun Sudhir 21BDS0040")

```

Output:

```
> # Finding the range of Won-By runs
> print(max(dataset$Won_By,na.rm=TRUE)-min(dataset$Won_By, na.rm=TRUE))
[1] 143
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
> |
```

```
# Finding the range of Match_IDs
print(max(dataset$Match_Id,na.rm=TRUE)-min(dataset$Match_Id, na.rm=TRUE))
print("Varun Sudhir 21BDS0040")
```

Output:

```
> # Finding the range of Match_IDs
> print(max(dataset$Match_Id,na.rm=TRUE)-min(dataset$Match_Id, na.rm=TRUE))
[1] 645037
> print("Varun Sudhir 21BDS0040")
[1] "Varun Sudhir 21BDS0040"
> |
```