IPL FIRST INNING SCORE PREDICTION MODEL

VAC INTERNSHIP PROJECT REPORT

Submitted by

BHARATHWAJ M – RA2011026020065

AAKASH S – RA2011026020069

ABIJITH P – RA2011026020070

MB HARIHARAN – RA2011026020096

Under the guidance of

Mr. BEZALEEL S, Robotics Engineer

Prag Robotics Private Limited, Chennai - 600096

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING WITH SPECIALISATION IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM CAMPUS, CHENNAI-600089

ABSTRACT

Cricket, being a sport with a rich and complex set of variables, has always posed a challenge to predict scores with high accuracy. However, with the advent of advanced machine learning algorithms, it has become possible to analyze large amounts of historical data and derive meaningful insights that can be leveraged to generate accurate predictions.

The proposed IPL first innings score predictor model aims to utilize these cutting-edge techniques to predict the scores of teams during the first innings of a match. The model will incorporate a variety of factors, including the pitch conditions, weather, team composition, past performance, and other relevant statistics. This will be accomplished through a series of algorithms that will take into account the historical performance of the teams in the league, the location of the match, the pitch conditions, and other relevant parameters.

To develop this model, we will be collecting and analyzing vast amounts of data, which will be processed using sophisticated machine learning techniques. The data will include various metrics such as team statistics, player performance, and past match records. The model will be trained on this data using advanced algorithms to ensure that it provides accurate and reliable predictions.

Once the model is developed, it will be tested using real-time IPL match data to validate its accuracy and effectiveness. The model will be refined further based on the feedback received from these tests, and the final version will be made available to the public for use.

The proposed IPL first innings score predictor model aims to provide an accurate and reliable tool for predicting the scores of teams during the first innings of IPL matches. The model will be developed using advanced machine learning techniques and will take into account various factors to ensure that it provides the most accurate predictions possible.

TABLE OF CONTENTS

INTRODUCTION

The game of T20 cricket is a complex sport with many factors that can influence the outcome of a match. The objective of our project is to create an IPL first innings score predictor that can accurately predict the outcome of T20 cricket matches. Our model focuses on the first innings score based on scores in the powerplay and stadium history, but we also aim to consider other factors such as total runs scored, winning the coin toss, home-ground advantage, and the number of balls with no runs scored. By including such aspects, we can increase the strength of our analysis and provide more accurate predictions. Our ultimate goal is to build a model that can predict the winner of a match. We believe that machine learning can play a significant role in predicting the outcome of cricket matches. Our project aims to design a result prediction system for a T20 cricket match, particularly for an IPL match while the match is in progress. We hope that our project will contribute to the development of more accurate and reliable prediction models for T20 cricket matches.

1.1 Project Domain:-

The project domain for predicting first innings IPL score using machine learning algorithms, mainly utilizing random forest regressor, falls under the field of sports analytics and data science. Specifically, this project belongs to the domain of cricket analytics, where data from past matches is analyzed to gain insights and make predictions about future matches. The project domain involves the collection, preprocessing, and analysis of a large amount of data related to IPL matches, including player performance, team statistics, pitch conditions, venue and other relevant factors. The machine learning algorithms used in this project, including the random forest regressor, belong to the domain of supervised learning, where the model is trained on labeled data to predict future outcomes accurately. Overall, the project domain involves the integration of machine learning algorithms with domain expertise in cricket to develop a robust model that can accurately predict the first innings scores of IPL matches.

1.2 Scope of the Project:-

The scope of the project for predicting first innings IPL score using machine learning algorithms, mainly utilizing random forest regressor, includes the following:

Data Collection: The project will involve the collection of a large amount of data related to IPL matches, including player performance, team statistics, pitch conditions, weather, and other relevant factors.

Data Preprocessing: The collected data will need to be preprocessed and cleaned to remove any inconsistencies or errors, and to convert it into a suitable format for analysis.

Data Analysis: Advanced machine learning algorithms, mainly the random forest regressor, will be used to analyze the data and generate accurate predictions for first innings IPL scores. Various factors such as pitch conditions, weather, team composition, past performance, and other relevant statistics will be incorporated into the model to ensure the most accurate predictions possible.

Model Validation: The developed model will be validated using real-time IPL match data to ensure that it provides accurate and reliable predictions.

Model Deployment: Once the model is developed and validated, it will be deployed for public use, allowing cricket enthusiasts and statisticians to gain valuable insights into the game and make more informed decisions based on the predictions generated by the model.

Overall, the scope of the project involves the integration of machine learning algorithms with domain expertise in cricket to develop a robust model that can accurately predict the first innings scores of IPL matches. The project will require extensive data collection, preprocessing, and analysis, as well as rigorous testing and validation to ensure that the model provides accurate and reliable predictions. The end result will be a valuable tool that decisions based on the predictions generated by the model.

1.3 Methodology:-

1.RANDOM FOREST

It is an essential learning approach for classification and regression to create a large number of decision trees. Preliminaries of decision trees are common approaches for a variety of machine learning problems. Tree learning is required for serving n off the self-produce for data mining since it is invariant despite scaling and several other changes. The trees are grown very deep in order to learn a high regular pattern. Random forest is a method of averaging several deep decision trees trained on various portions of the same training set. This comes at the price of a slight increase in bias and some interoperability.

2.Data Collection

The statistics were gathered from Bangalore home prices. The information includes many variables such as area type, availability, location, BHK, society, total square feet, bathrooms, and balconies.

3.Linear Regression

Linear regression is a supervised learning technique. It is responsible for predicting the value of a dependent variable (Y) based on a given independent variable (X). It is the connection between the input (X) and the output (Y). It is one of the most well-known and well-understood machine learning algorithms. Simple linear regression, ordinary least squares, Gradient Descent, and Regularization are the linear regression models.

4.Decision Tree Regression

It is an object that trains a tree-structured model to predict data in the future in order to provide meaningful continuous output. The core principles of decision trees, Maximizing Information Gain, Classification trees, and Regression trees are the processes involved in decision tree regression. The essential notion of decision trees is that they are built via recursive partitioning. Each node can be divided into child nodes, beginning with the root node, which is known as the parent node. These nodes have the potential to become the parent nodes of their resulting offspring nodes. The nodes at the informative features are specified as the maximizing information gain, to establish an objective function that is to optimize the tree learning method.

5.Classification Trees

Classification trees are used to forecast the object into classes of a categorical dependent variable based on one or more predictor variables.

6.Regression Trees

It supports both continuous and categorical input variables. Regression trees are regarded as research with various machine algorithms for the regression issue, with the Decision Tree approach providing the lowest loss. The R-Squared value for the Decision Tree is 0.998, indicating that it is an excellent model. The Decision Tree was used to complete the web development.

7.Support Vector Regression

Supervised learning is linked with learning algorithms that examine data for classification and regression analysis.

PROJECT DESCRIPTION

2.1 EXISTING SYSTEM:-

There are several existing systems for predicting the first innings score of an IPL match using machine learning algorithms. One such system is the IPL First Innings Score Prediction project, which uses the Linear Regression algorithm to predict the first innings score.

The existing system uses deep learning and machine learning algorithms to predict the first innings score. The paper uses Linear Regression and Ridge Regression algorithms to predict the first innings score. The system is trained on historical IPL data and tested on new data to evaluate its accuracy.

2.2 PROPOSED SYSTEM :-

The proposed system can use any of these four models - Random Forest Regressor, Linear Regression, Decision Tree Regressor, and ADA Boost Regressor which is a simple yet effective algorithm for predicting numerical values, but our model is mainly trained by Random Forest Regressor. The system will be trained on historical IPL data, including factors such as the venue, team composition, and past performance.

The data will be preprocessed to remove any outliers and normalize the features. The model will then be trained on the preprocessed data to predict the first innings score. Random forest is more accurate compared to other models.

2.3 FEASIBILITY STUDY:-

A Feasibility study is carried out to check the viability of the project and to analyze the strengths and weaknesses of the proposed system. The application of usage of mask in crowd areas must be evaluated. A feasibility study is a detailed analysis that considers all the critical aspects of a proposed project to determine the likelihood of it succeeding.

Although feasibility studies can help project managers determine the risk and return of pursuing a plan of action, several steps should be considered before moving forward. Detailed investigation has really helped in knowing the feasibility of various Software and hardware components as well as the overall project model in terms of technical, operational, and economical respects. Thus, this project is feasible in all respect.

The feasibility study is carried out in three forms:

• Economic Feasibility

The proposed system does not require any high-cost equipment. This project can be developed within the available software. This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

• Technical Feasibility

The proposed system is completely a Deep learning model. The main tools used in this project are Anaconda prompt, Visual studio, Jupyter Notebook, and the language used to execute the process in Python. This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team can convert the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

• Social Feasibility

Social feasibility is a determination of whether project will be acceptable or not. our project is Eco-friendly for society and there are no social issues. Our project must not be threatened by the system instead must accept it as a necessity. This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let’s say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization’s ideal location isn’t zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

2.4 HARDWARE SPECIFICATIONS

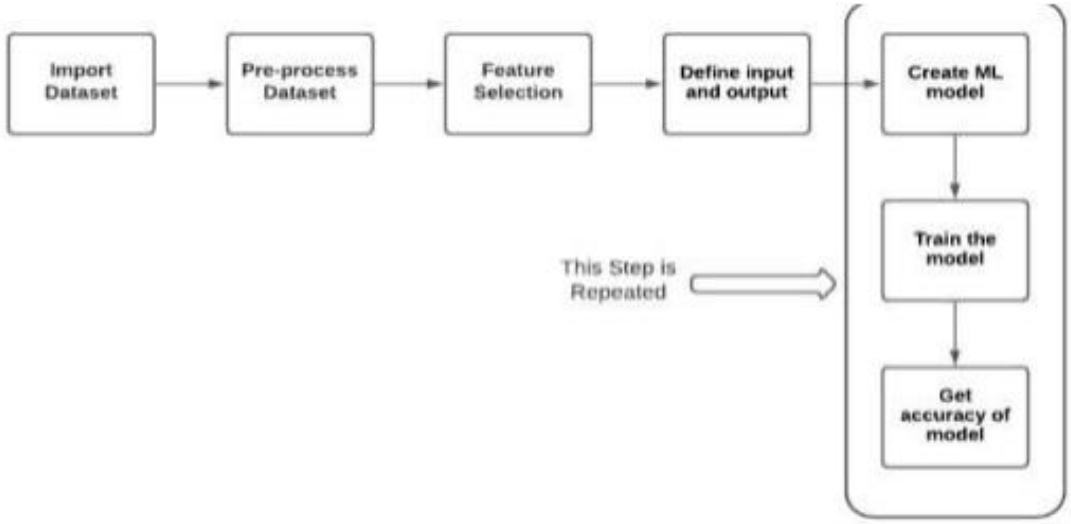
- PROCESSOR - INTEL I5-8250 CPU @1.60GHZ 1.80GHZ
- 512 GB SSD
- NVIDIA GEFORCE RTX
- CPU QUAD CORES

SOFTWARE USED

- R
- JUPYTER NOTEBOOK
- PYTHON
- FLASK

SOURCE CODE IMPLEMENTATION AND POSTER PRESENTATION

3.1 ARCHITECTURE



The architecture diagram of an IPL first innings score predictor using machine learning involves several steps. The first step is data collection, where data is collected from various sources such as the IPL website, cricket databases, and other sources. The next step is data pre-processing, where the collected data is cleaned, transformed, and prepared for analysis. The third step is feature selection, where the most relevant features are selected for analysis. The fourth step is model training, where the selected features are used to train the machine learning model. The fifth step is model evaluation, where the performance of the model is evaluated using various metrics such as accuracy, precision, recall, and F1 score. The final step is model deployment, where the trained model is deployed in a production environment for real-time prediction.

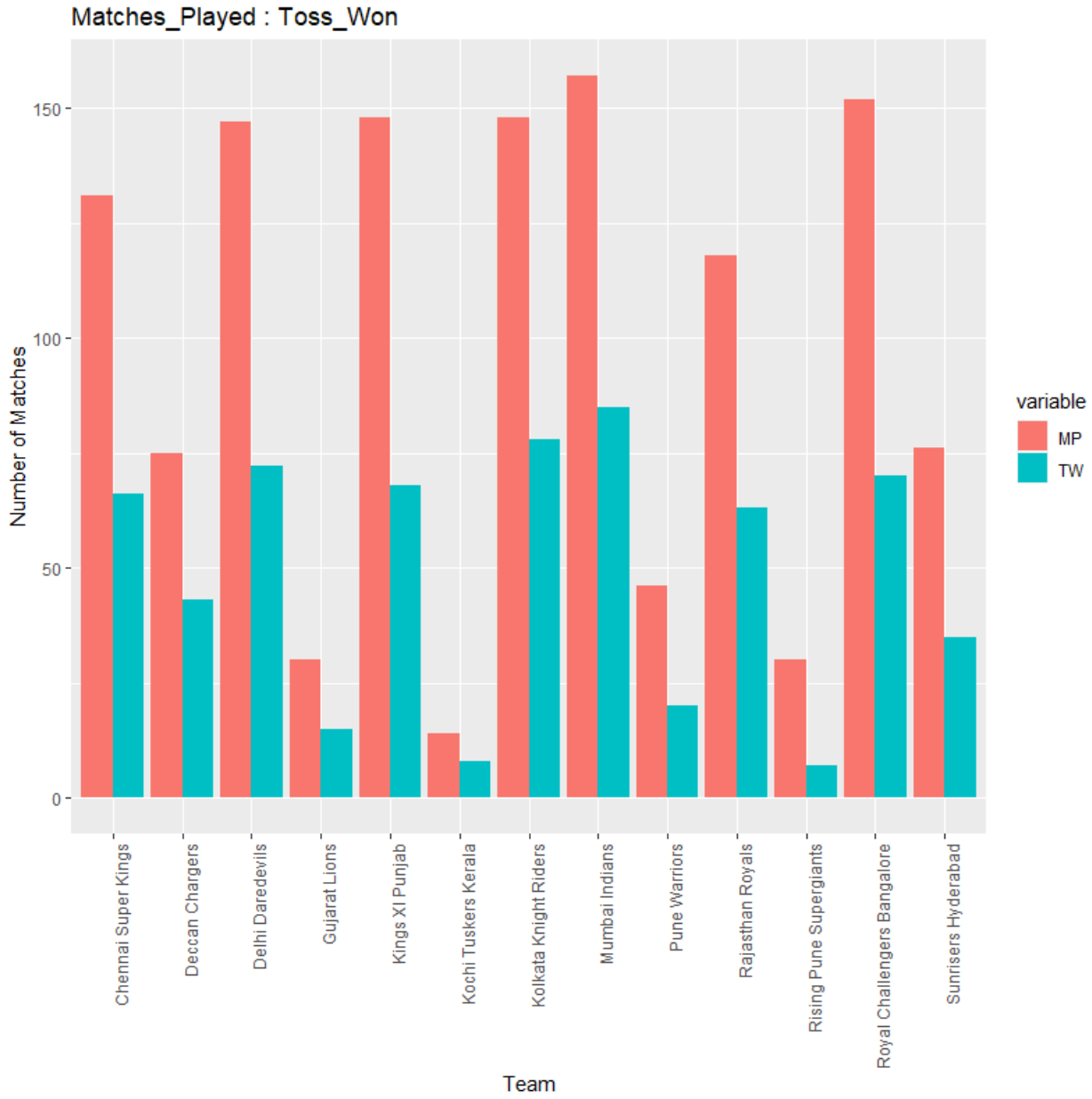
The model works by using machine learning algorithms such as random forests, linear regression, ridge regression, decision trees, and support vector machines to predict the first innings score of the IPL matches. The model focuses on the first innings score based on scores in the powerplay and stadium history. The model uses past datasets available from the IPL to predict the score of each match and the winning team. The model uses deep learning and machine learning algorithms for finding the first innings predicted score. The model attempts to predict the innings score considering the past data of match by match and ball by ball. The ultimate objective is to build a model that can predict and provide accurate predictions for the outcome of T20 cricket matches.

3.2 EXPLORATORY DATA ANALYSIS

Code:

```
melt(Team_BAT_Analysis) %>%
  filter(variable %in% c("MP","TW")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Team" , y = "Number of Matches",
       title = "Matches_Played : Toss_Won")
```

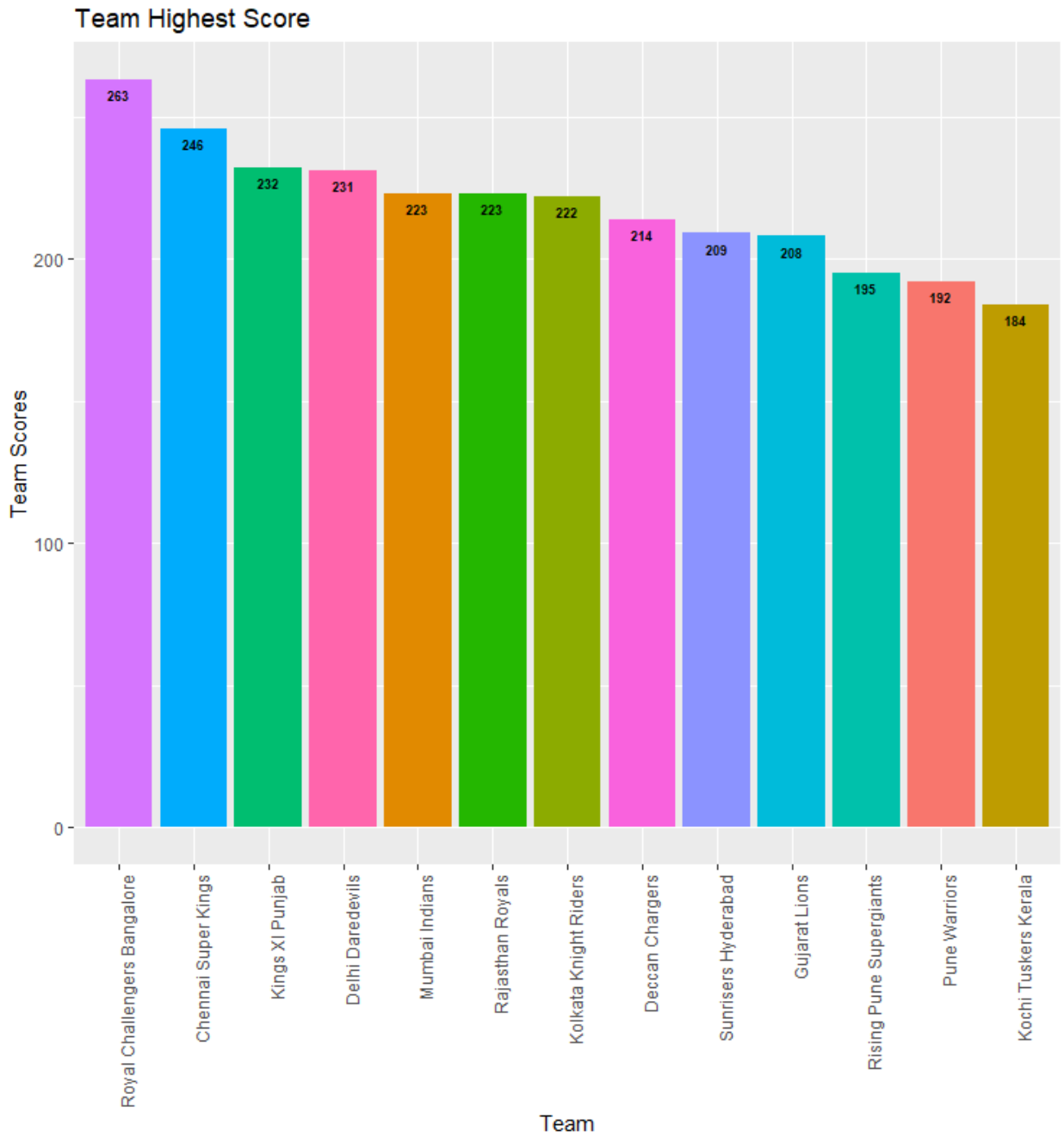
Output:



Code:

```
ggplot(data = Team_Analysis , aes(reorder(Team,-THS) , y = THS ,fill = rainbow(13)))+
  geom_bar(stat ="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Team" , y = "Team Scores", title = "Team Highest Score") +
  geom_text(aes(label = THS),color="black",fontface = "bold",size =2.5,vjust = 1.8)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  guides(fill=F)
```

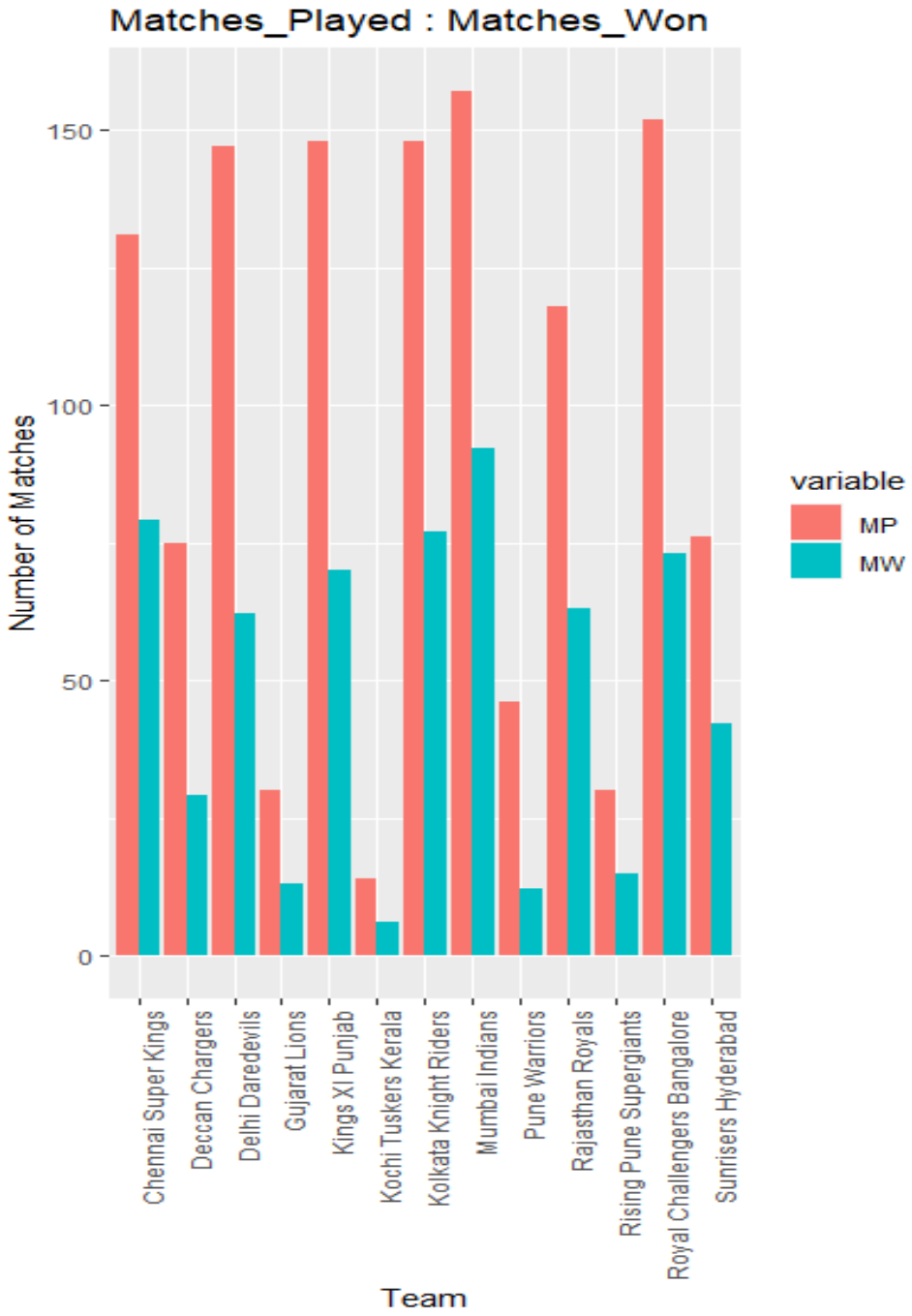
Output:



Code:

```
melt(Team_BAT_Analysis) %>%
  filter(variable %in% c("MP","MW")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Team" , y = "Number of Matches",
       title = "Matches_Played : Matches_Won")
```

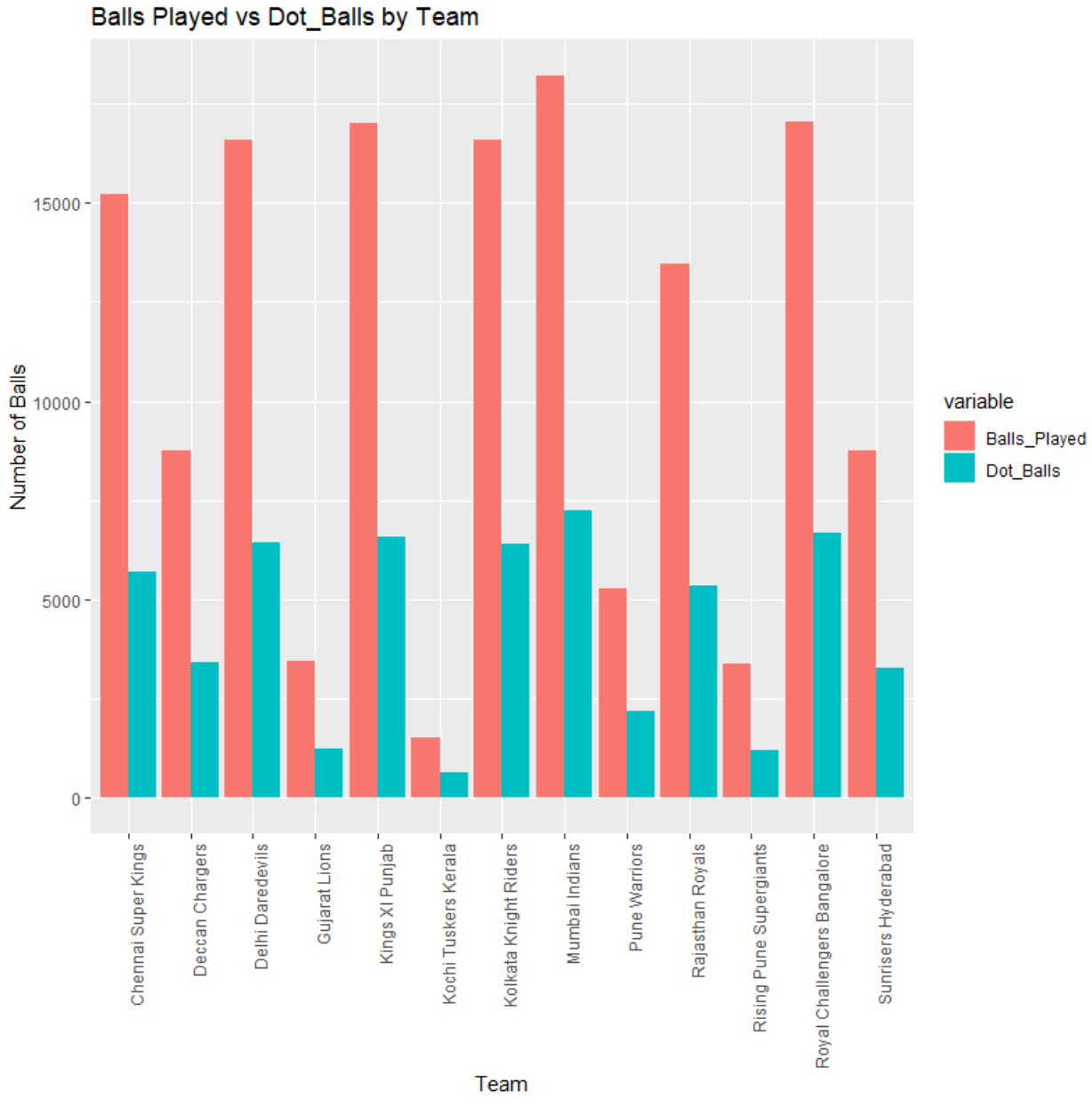
Output:



Code:

```
melt(Team_BAT_Analysis) %>%
  filter(variable %in% c("Balls_Played", "Dot_Balls")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Team" , y = "Number of Balls",
       title = "Balls Played vs Dot_Balls by Team")
```

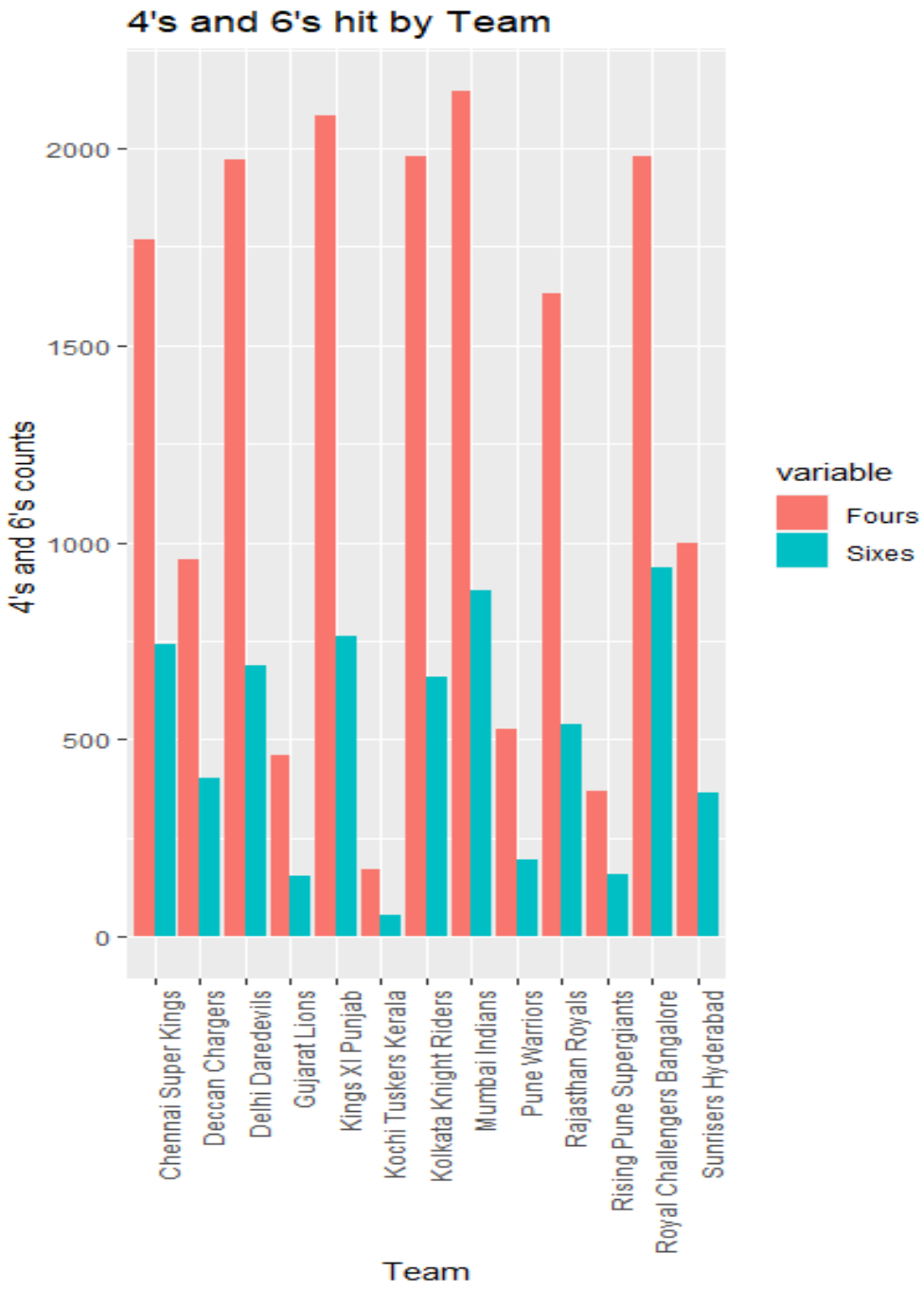
Output:



Code:

```
melt(Team_BAT_Analysis) %>%
  filter(variable %in% c("Fours", "Sixes")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Team" , y = "4's and 6's counts ",
       title = "4's and 6's hit by Team")
```

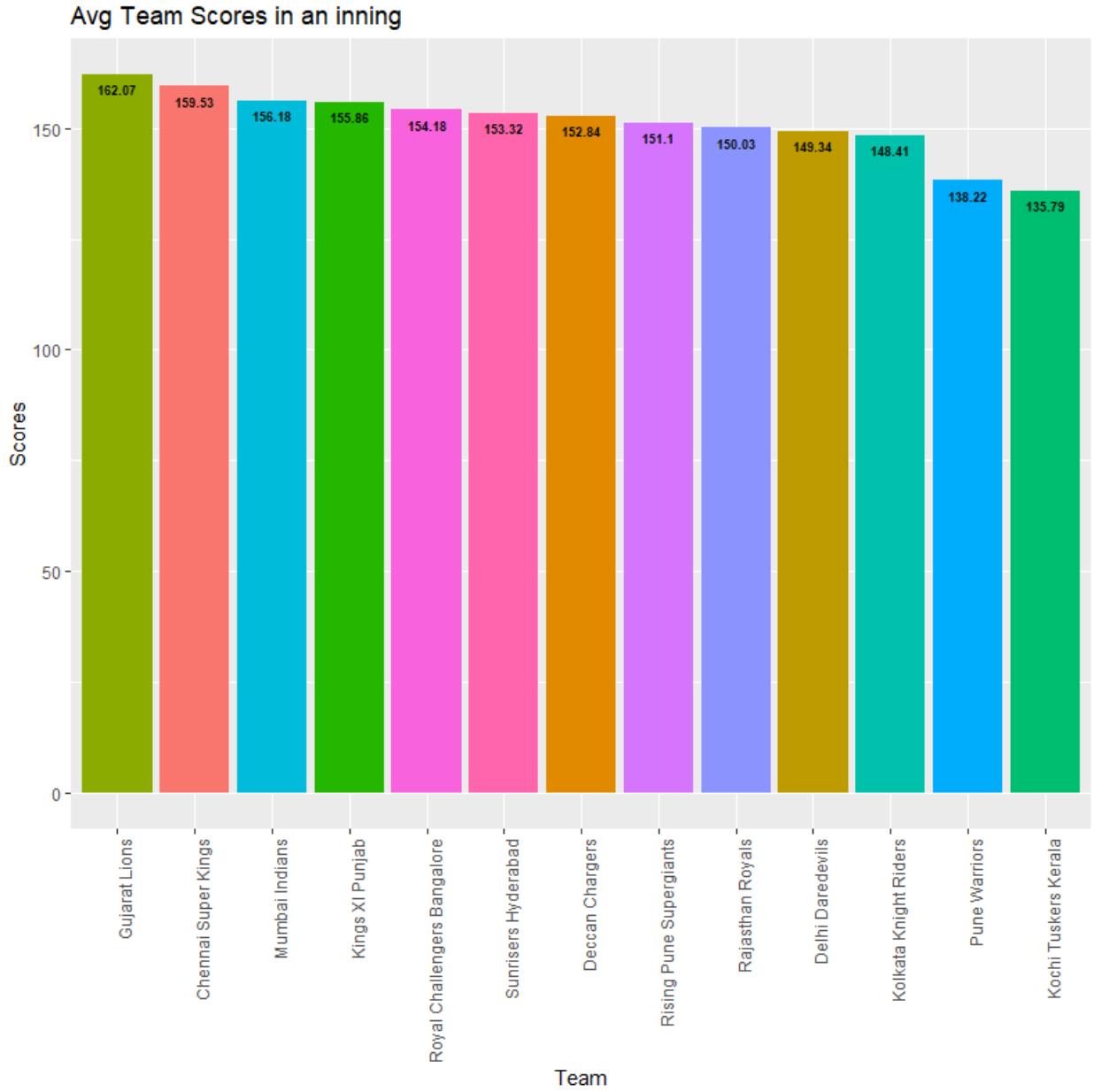
Output:



Code:

```
ggplot(data = table , aes(reorder(Team,-Avg_Score) , y = Avg_Score ,fill = Team))+
  geom_bar(stat ="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Team" , y = "Scores", title = "Avg Team Scores in an inning") +
  geom_text(aes(label = Avg_Score),color="black",fontface = "bold",size =2.5,vjust = 1.8)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  guides(fill=F)
```

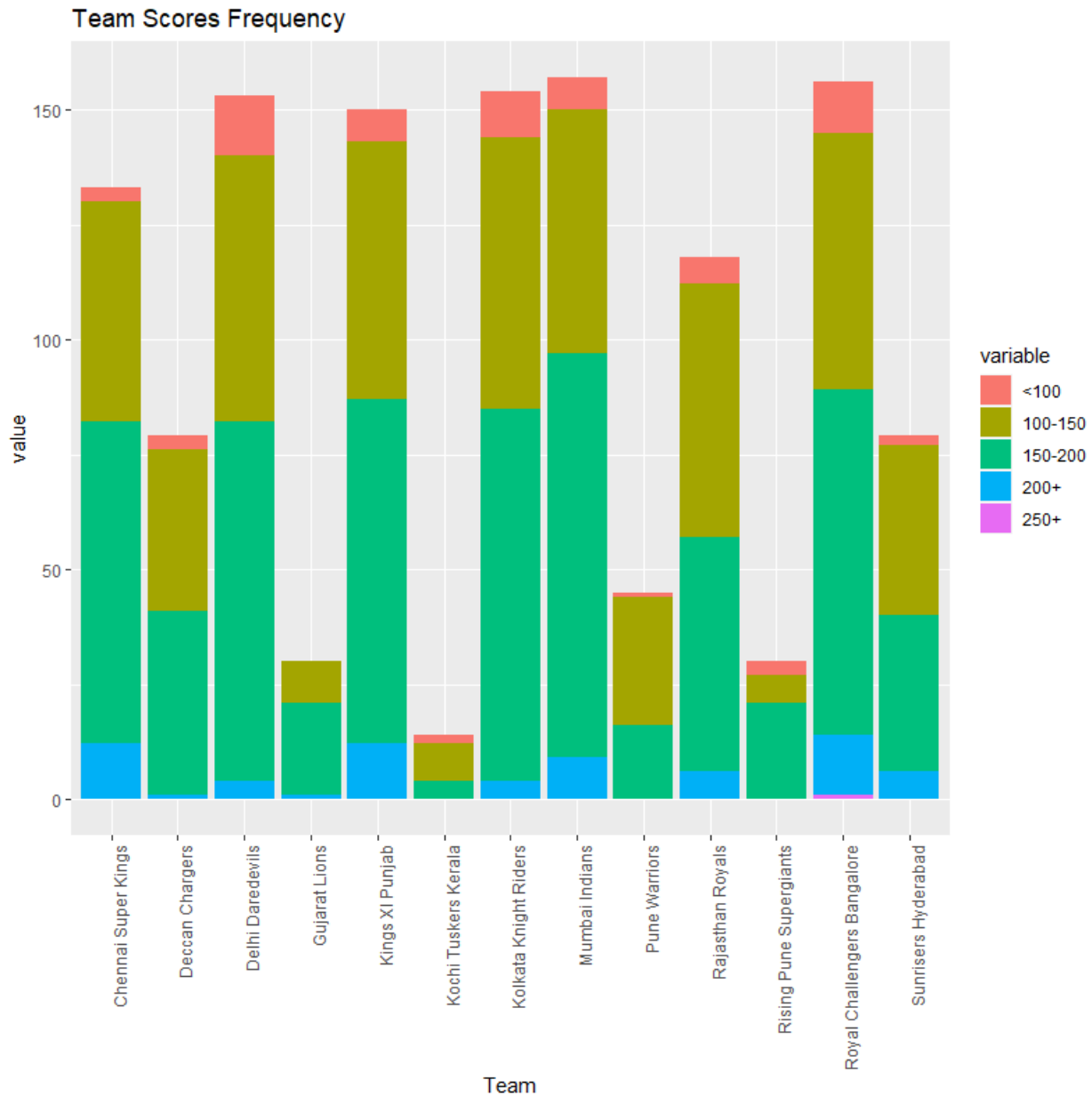
Output:



Code:

```
melt(Team_BAT_Analysis) %>%
  filter(variable %in% c("<100","100-150","150-200","200+","250+"))%>%
  ggplot(aes(Team,value,fill = variable))+
  geom_bar(stat ="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  ggtitle("Team Scores Frequency")
```

Output:

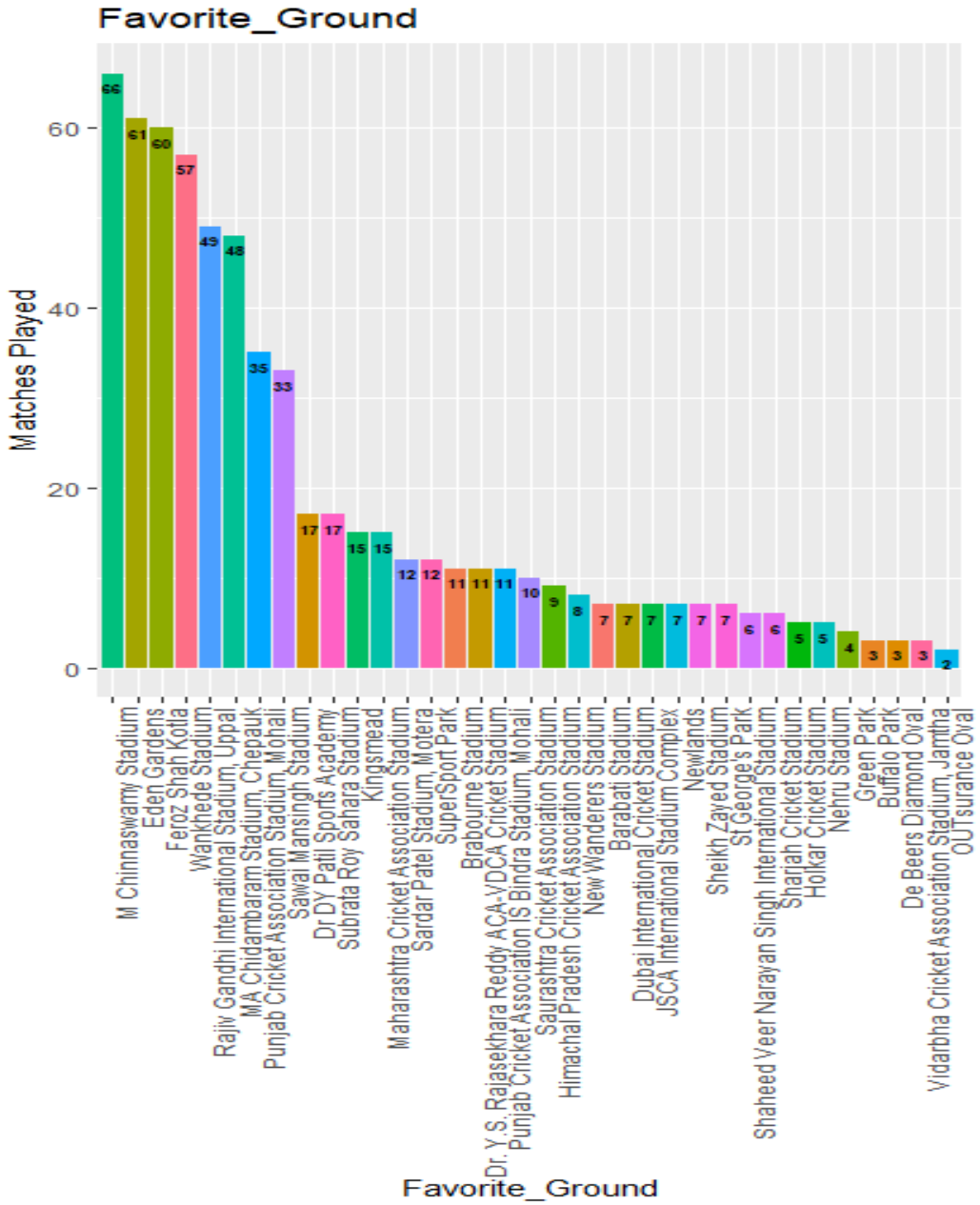


Code:

```
Favorite_Ground <- sqldf("select venue Favorite_Ground , count(venue) Matches from m
                 group by venue order by Matches desc")

ggplot(data = Favorite_Ground , aes(reorder(Favorite_Ground,-Matches) , y = Matches ,fill = Favorite_Ground))+
  geom_bar(stat ="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Favorite_Ground" , y = "Matches Played", title = "Favorite_Ground") +
  geom_text(aes(label = Matches),color="black",fontface = "bold",size =2,vjust = 1.8)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  guides(fill=F)
```

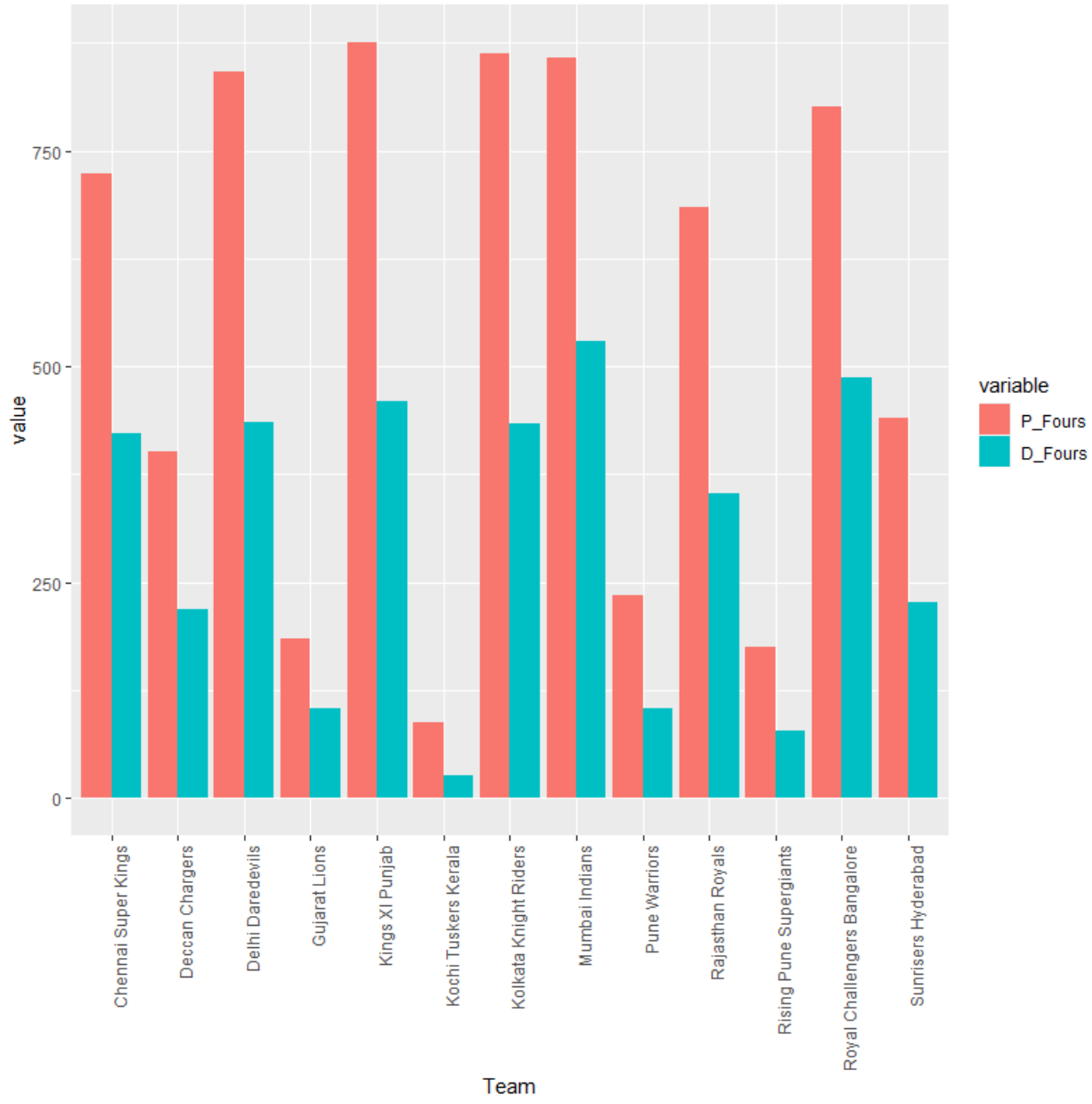
Output:



Code:

```
#Boundaries and Sixes hit by team in (Power Play  Vs Death Overs)
melt(PD_BAT_Team) %>%
  filter(variable %in% c("P_Fours","D_Fours")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

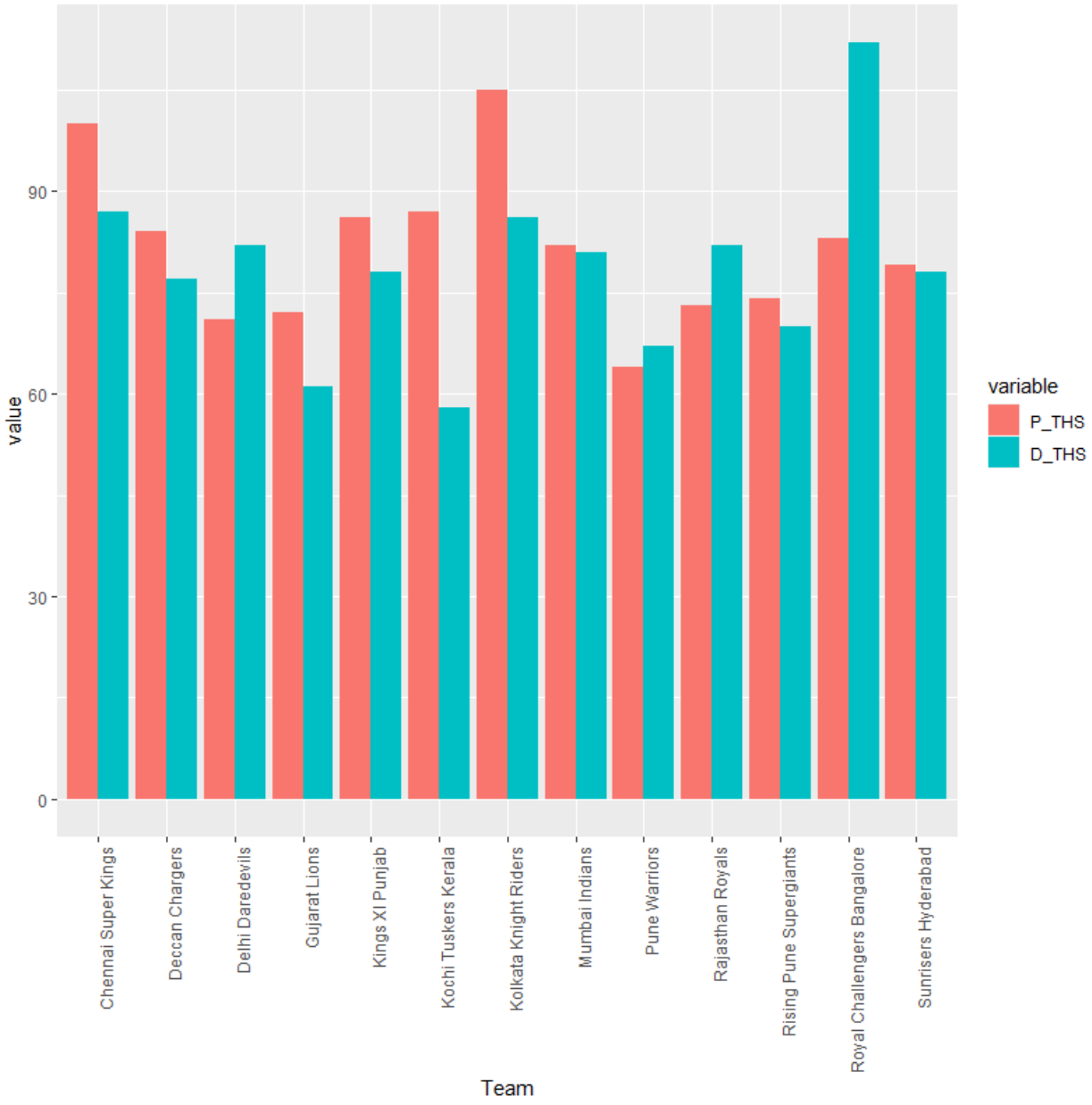
Output:



Code:

```
#Highest Score by team in (Power Play  Vs Death Overs)
melt(PD_BAT_Team) %>%
  filter(variable %in% c("P_THS","D_THS")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

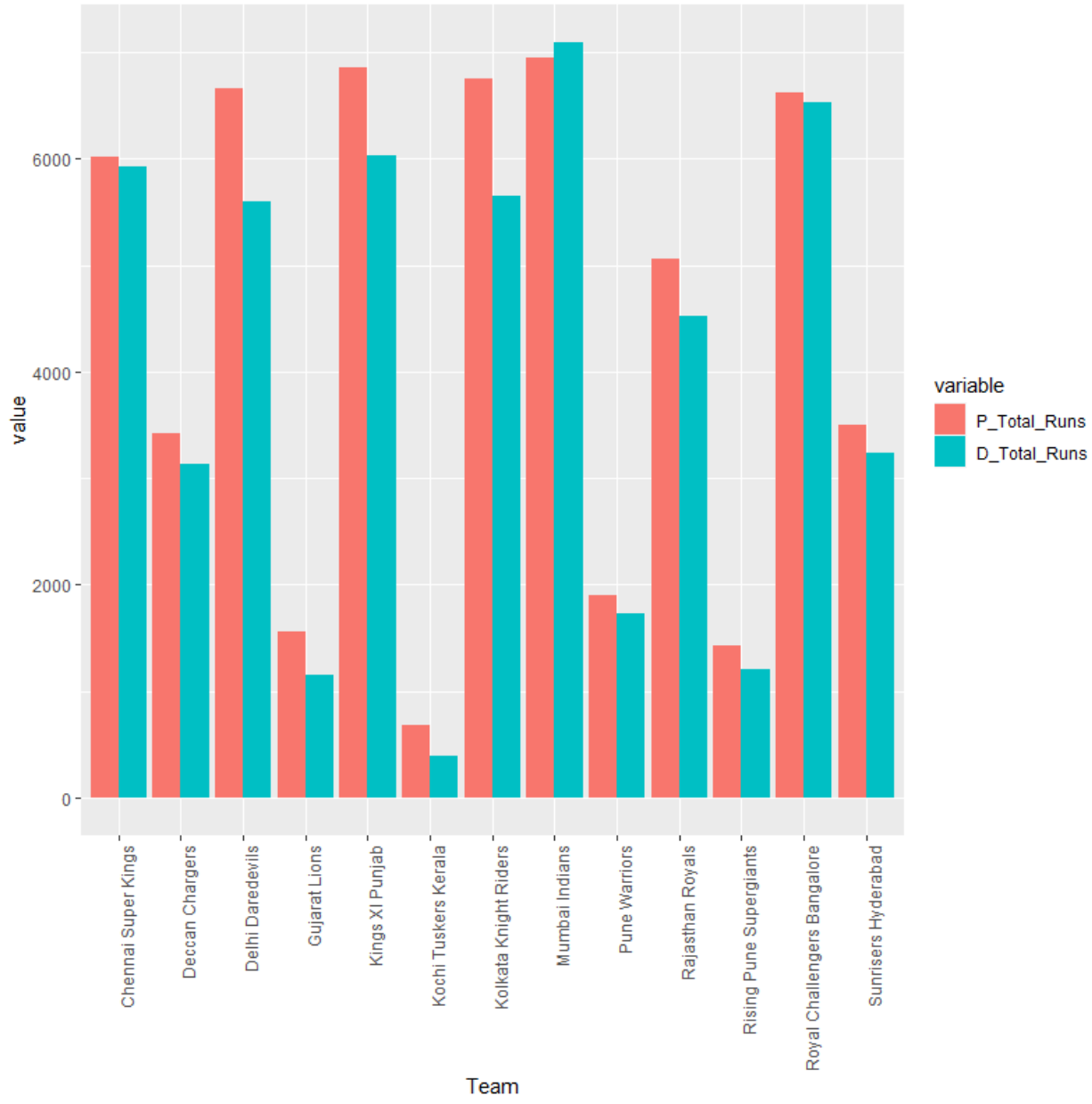
Output:



Code:

```
#Total_Runs (Power Play  Vs Death Overs)
melt(PD_BAT_Team) %>%
  filter(variable %in% c("P_Total_Runs","D_Total_Runs")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

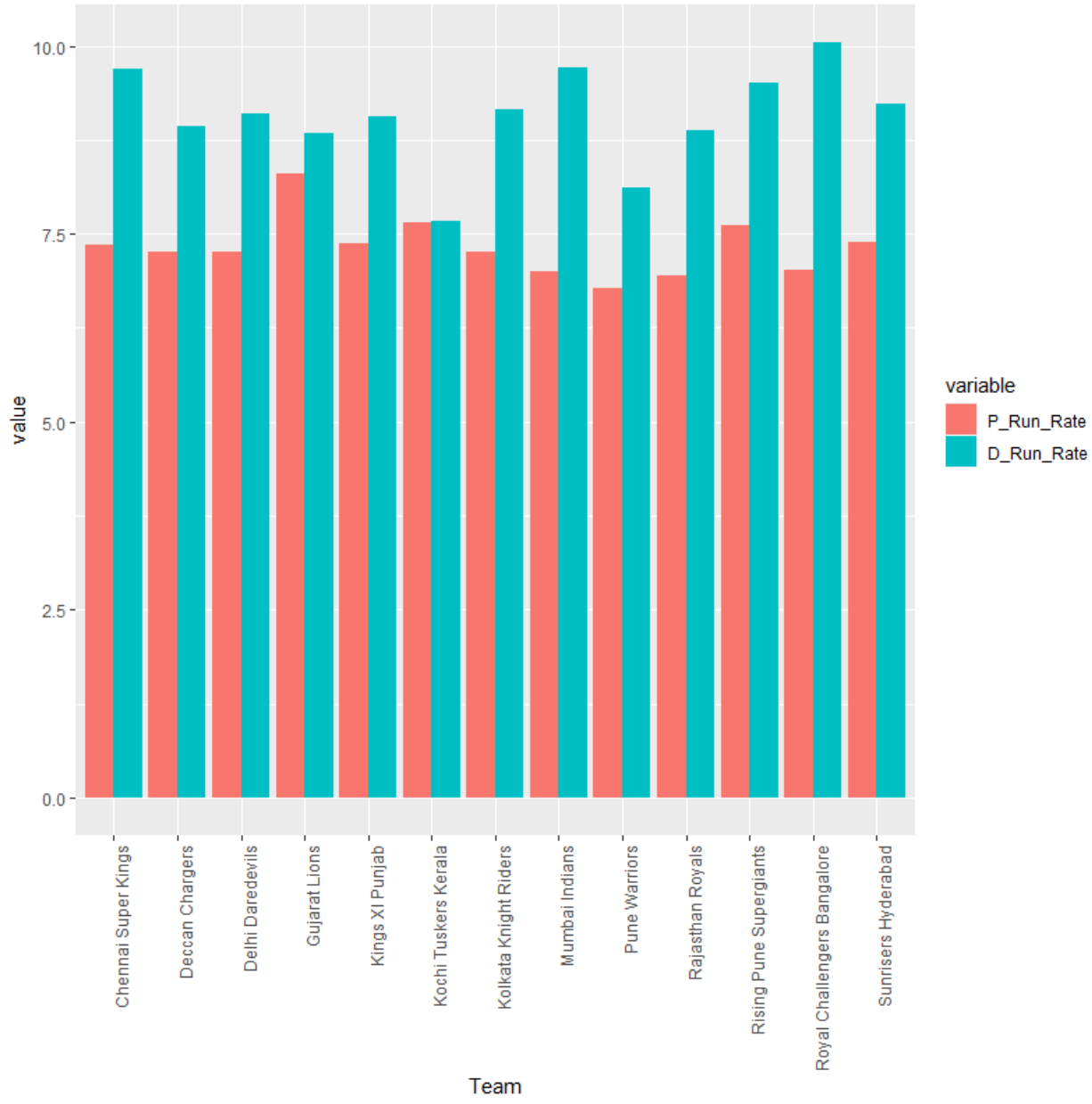
Output:



Code:

```
#Run_Rate (Power Play  Vs Death Overs)
melt(PD_BAT_Team) %>%
  filter(variable %in% c("P_Run_Rate","D_Run_Rate")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

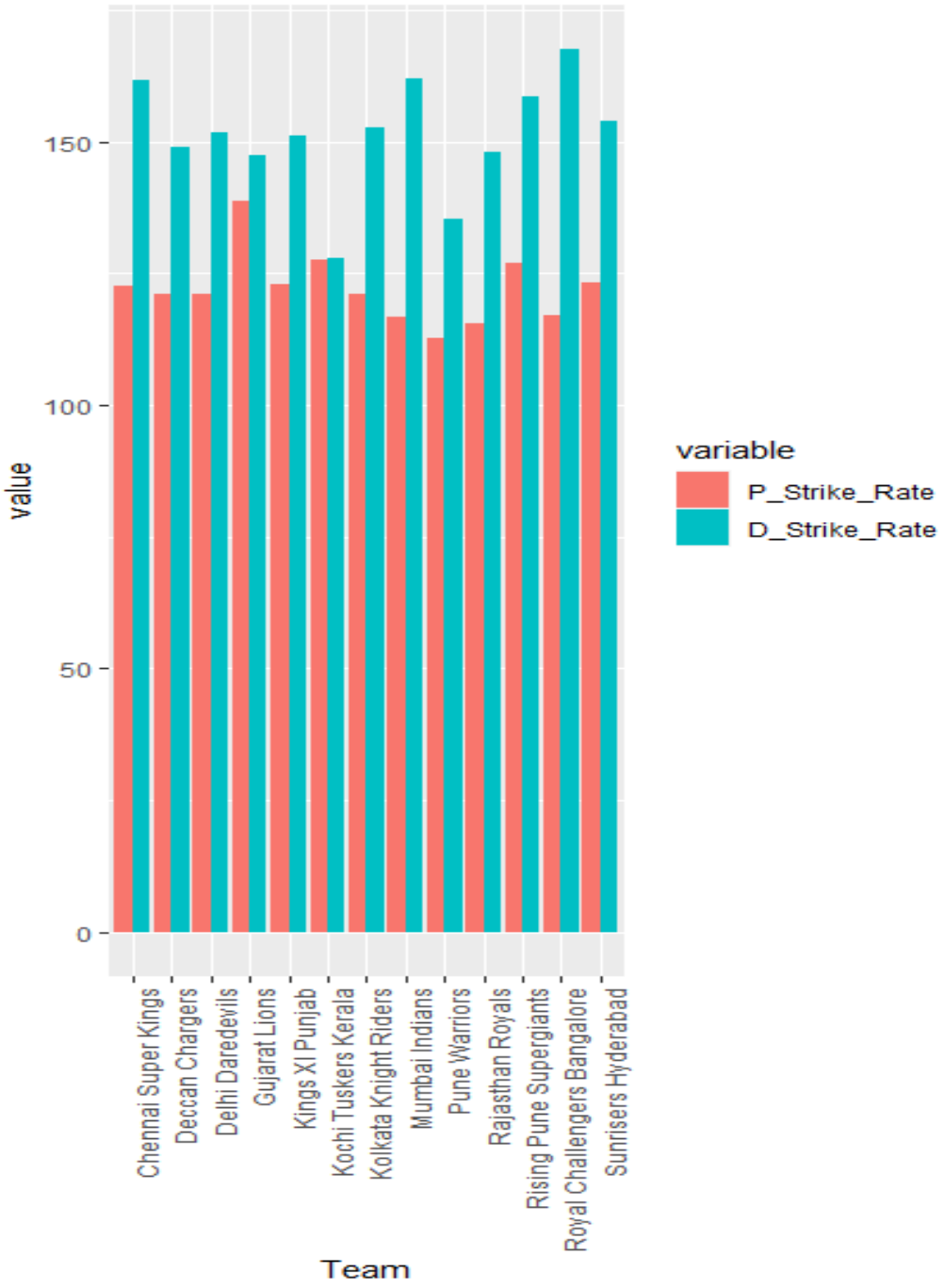
Output:



Code:

```
#Strike_Rate (Power Play  Vs Death Overs)
melt(PD_BAT_Team) %>%
  filter(variable %in% c("P_Strike_Rate","D_Strike_Rate")) %>%
  ggplot(aes(x =  Team , y = value,fill = variable))+
  geom_bar(stat ="identity",position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

Output:



3.3 Source Code

```
Team_Analysis <- sqldf("select tos.Team,
      tos.Ones,
      tt1.Twos,
      ttt.Threes,
      tf.fours,
      ts.Sixes,
      tp.Thirty_plus_count,
      fp.Fifty_plus_count,
      hp.Hundred_plus,
      ofp.One_Fifty_plus_count,
      ths.Team_Highest_Score,
      bpbt.Balls_Played,
      dbt.Dot_Balls
      from tos
      inner join tt1
      on tos.Team = tt1.Team
      inner join ttt
      on tos.Team = ttt.Team
      inner join tf
      on tos.Team = tf.Team
      inner join ts
      on tos.Team = ts.Team
      inner join tp
      on tos.Team = tp.Team
      inner join fp
      on tos.Team = fp.Team
      left join hp
      on tos.Team = hp.Team
      left join ofp
      on tos.Team = ofp.Team
      inner join ths
      on tos.Team = ths.Team
      inner join bpbt
      on tos.Team = bpbt.Team
      inner join dbt
      on tos.Team = dbt.Team
      ")

Team_Analysis

Team_Analysis[is.na(Team_Analysis)] <- 0  #Replace NA values with Zero

Team_Analysis <- rename(Team_Analysis ,"30+" = "Thirty_plus_count" )
Team_Analysis <- rename(Team_Analysis ,"50+" = "Fifty_plus_count" )
Team_Analysis <- rename(Team_Analysis ,"100+" = "Hundred_plus" )
Team_Analysis <- rename(Team_Analysis ,"150+" = "One_Fifty_plus_count" ) #Renaming Columns

Team_Analysis <- merge(x = Team_Analysis , y = Team_Bat , By = "Team")

Team_Analysis <- rename(Team_Analysis ,"THS" = "Team_Highest_Score")

Team_Analysis
```

```r
# Number of matches played by each team

mp <- as.data.frame(table(m$team1) + table(m$team2))

mp <- rename(mp , Team = Var1 , matches_played = Freq)

#Number of Matches won by each teams
mw <- as.data.frame(table(m$winner)) %>% filter(Freq != 3)

mw <- rename(mw , Team = Var1 , matches_won = Freq)

mw %>% group_by(Team) %>% top_n(14)

#join both data_frame mp and mw

table <- merge(x = mp , y = mw , by = "Team") #inner join

table$win_per <- round((table$matches_won / table$matches_played),4) * 100

print(table)

Toss <- sqldf("select toss_winner as Team, count(*) as Toss_Won from m group by toss_winner")

sqldf("select * from Toss order by Toss_Won desc")

table <- merge( x = table , y = Toss , by = "Team")

table

table$Toss_Win_Per = round((table$Toss_Won / table$matches_played ) * 100 , 2) #calculating Toss Win percentage

Total_Runs <- sqldf("select batting_team as Team , sum(total_runs) as Total_Runs from d group by
                    batting_team")


table <- merge( x = table , y = Total_Runs , by = "Team")
table$Avg_Score <- round((table$Total_Runs  / table$matches_played),2) #Team Average Score

table

Team_BAT_Analysis <- merge(x = Team_Analysis , y = table ,By = "Team")

Team_BAT_Analysis

setnames(Team_BAT_Analysis , c("matches_played","matches_won","win_per","Toss_Won","Toss_Win_Per","Total_Runs",
                               "Avg_Score"),c("MP","MW","MWP","TW","TWP","TRs","AS"))

Team_BAT_Analysis #Final Table
```

3.4 Implementation and testing

In [43]:
```
algo3 = {
    "Random Forest": {
        "model": RandomForestRegressor(),
        "params": {
            "n_estimators": [100, 200, 300],
            "criterion": ["mse", "friedman_mse"],
            "max_depth": [1, 3, 5, 7, 9, 10, 11, 12, 14, 15, 18, 20, 25, 28, 30, 33, 38, 40],
            "min_samples_split": [2, 4, 6, 8, 10, 15, 20],
            "min_samples_leaf": [i for i in range(1, 11)],
            "max_leaf_nodes": [None] + [i for i in range(10, 91, 10)],
            "max_features": ["auto", "log2", "sqrt", None]
        }
    }
}
```

Model Building

In [37]:
```
# here let's creat feature matrix X and target vector y

X = df2.drop(columns="final_total_runs") # 38 features
y = df2.final_total_runs
```

In [38]:
```
# let's split the data into train and test part

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

(X_train.shape, y_train.shape), (X_test.shape, y_test.shape)
```

Out[38]: (((88046, 36), (88046,)), ((22012, 36), (22012,)))

In [39]:
```
#Scaling the data is important
scaler = MinMaxScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [40]:
```
X_train[1].shape
```

Out[40]: (36,)

In [46]:
```
pd.set_option('display.max_colwidth', None)
pd.DataFrame(best_model_details)
```

Out[46]:

	Model Name	Best Score	Best Parameters
0	Random Forest	0.53319	{'n_estimators': 200, 'min_samples_split': 20, 'min_samples_leaf': 8, 'max_leaf_nodes': 80, 'max_features': 'auto', 'max_depth': 15, 'criterion': 'friedman_mse'}

In [47]:
```
test_model = []

for model_name, model in best_model.items():
    test_model.append({"Model Name": model_name, "Test Score": model.score(X_test, y_test)})

pd.DataFrame(test_model)
```

Out[47]:

	Model Name	Test Score
0	Random Forest	0.531382

In [49]:
```
print("-------- Training Data Error --------")
train_model_error
```

```
-------- Training Data Error --------
```

Out[49]:

	Model Name	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
0	Random Forest	14.860212	394.580062	19.864039

In [50]:
```
print("-------- Test Data Error --------")
test_model_error
```

```
-------- Test Data Error --------
```

Out[50]:

	Model Name	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
0	Random Forest	15.004829	405.308346	20.132271

First Innings Score Predictor for Indian Premier League (IPL)

A Machine Learning Web App, Built with Flask, Deployed using Heroku

--- Select a Batting team ---

--- Select a Bowling team ---

--- Select a Venue ---

Overs (>= 5.0) eg. 7.2

Runs eg. 64

Wickets eg. 4

Runs scored in previous 5 Overs eg. 42

Wickets taken in previous 5 Overs eg. 3

Predict Score

First Innings Score Predictor for Indian Premier League (IPL)

A Machine Learning Web App, Built with Flask, Deployed using Heroku

Mumbai Indians

Chennai Super Kings

Wankhede Stadium

6

50

2

40

1

Predict Score

First Innings Score Predictor for *Indian Premier League (IPL)*

A Machine Learning Web App, Built with Flask, Deployed using Heroku

The final predicted score (range): 144 to 159

Predicted Output: Run Range is between 144 – 159.

Actual Output: Mumbai Indians scored 157 runs.

PROJECT LINK:

https://github.com/BharathwajManoharan/IPL_First_Inning_Score_Prediction_2021

References:

1] T. Singh, V. Singla and P. Bhatia, "Score and winning prediction in cricket through data mining," 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI), Faridabad, India, 2015, pp. 60-66, doi: 10.1109/ICSCTI.2015.7489605.

[2] J. Kumar, R. Kumar and P. Kumar, "Outcome Prediction of ODI Cricket Matches using Decision Trees and MLP Networks," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 343-347, doi: 10.1109/ICSCCC.2018.8703301.

[3] A. Kaluarachchi and S. V. Aparna, "CricAI: A classification-based tool to predict the outcome in ODI cricket," 2010 Fifth International Conference on Information and Automation for Sustainability, Colombo, Sri Lanka, 2010, pp. 250-255, doi: 10.1109/ICIAFS.2010.5715668.

[4] A. I. Anik, S. Yeaser, A. G. M. I. Hossain and A. Chakrabarty, "Player's Performance Prediction in ODI Cricket Using Machine Learning Algorithms," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 2018, pp. 500-505, doi: 10.1109/CEEICT.2018.8628118. [5] N. Rodrigues, N. Sequeira, S. Rodrigues and V. Shrivastava, "Cricket Squad Analysis Using Multiple Random Forest Regression," 2019 1st International Conference on Advances in Information Technology (ICAIT), Chikmagalur, India, 2019, pp. 104-108, doi: 10.1109/ICAIT47043.2019.8987367.

[6] M. Jhawar and V. Pudi, "Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach", European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Riva del Garda, 2016

[7] A. I. Anik, S. Yeaser, A. G. M. I. Hossain and A. Chakrabarty, "Player's Performance Prediction in ODI Cricket Using Machine Learning Algorithms," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 2018, pp. 500-505, doi: 10.1109/CEEICT.2018.8628118. [8] Rameshwari Lokhande, P. M. Chawan "Live Cricket Score and Winning Prediction" Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, Volume-5 | Issue1 , February 2018, URL: http://www.ijtrd.com/papers/IJTRD12180.pdf

[9] H. Barot, A. Kothari, P. Bide, B. Ahir and R. Kankaria, "Analysis and Prediction for the Indian Premier League," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-7, doi: 10.1109/INCET49848.2020.9153972.