

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of the UGC Act, 1956

MINOR-PROJECT SYNOPSIS

IPL PLAYER ANALYSIS

Submitted By

| | | |
|----------|----------------------|-----------------|
| 1 | Aniket Kumar | 2005990 |
| 2 | Utkarsh Singh | 2005627 |
| 3 | Dhruti Pal | 2005511 |
| 4 | Ananya Anand | 20051705 |

CONTENTS :-

- **Introduction**
- **Motivation**
- **Libraries and database used**
- **Working**
- **Constraints and Limitations**
- **Future Scope**
- **References**

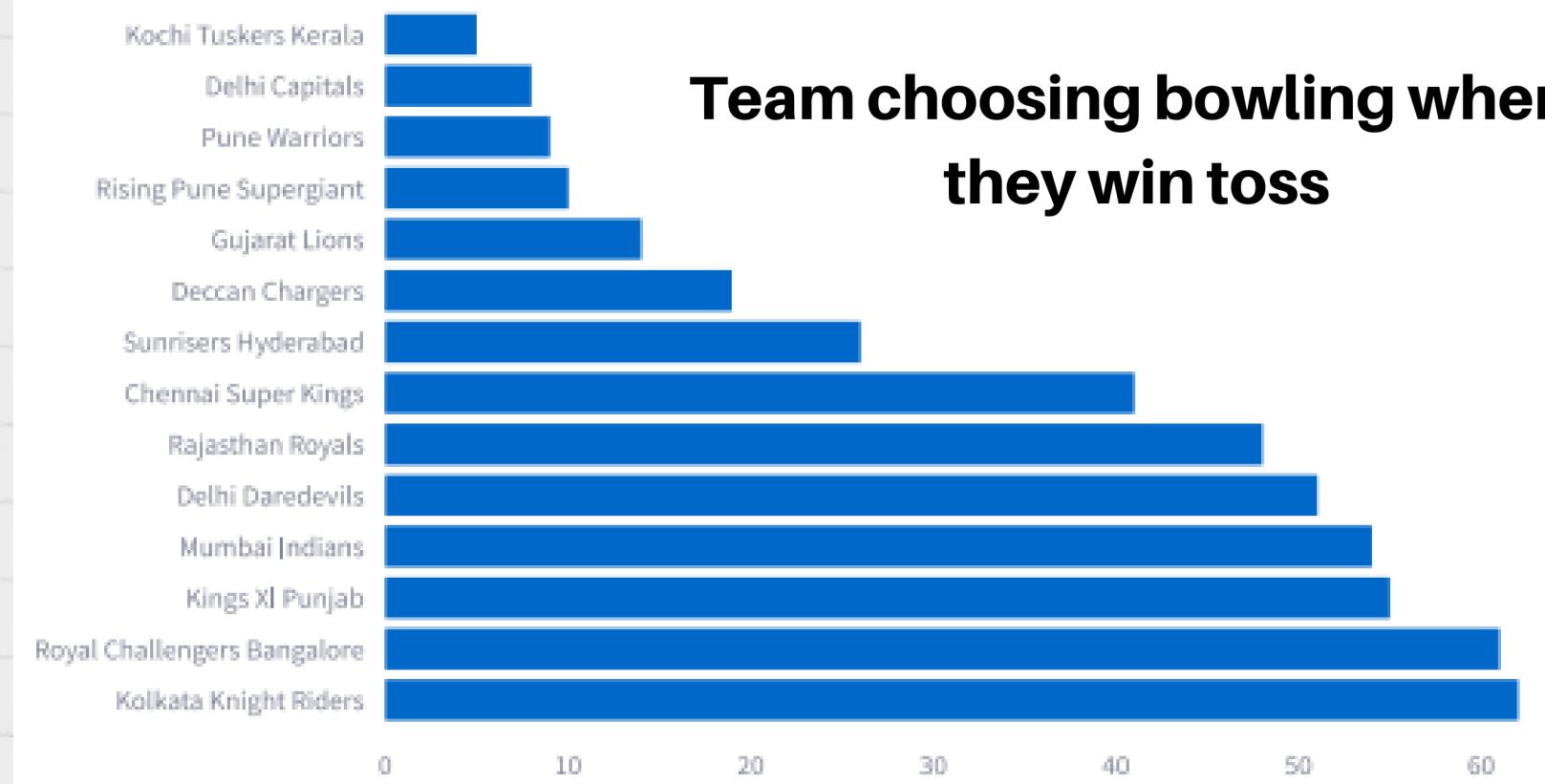
INTRODUCTION

The code is a Python program that analyze and visualize data related to the Indian Premier League (IPL) from 2008 to 2019. The code retrieves data from CSV files hosted on GitHub.

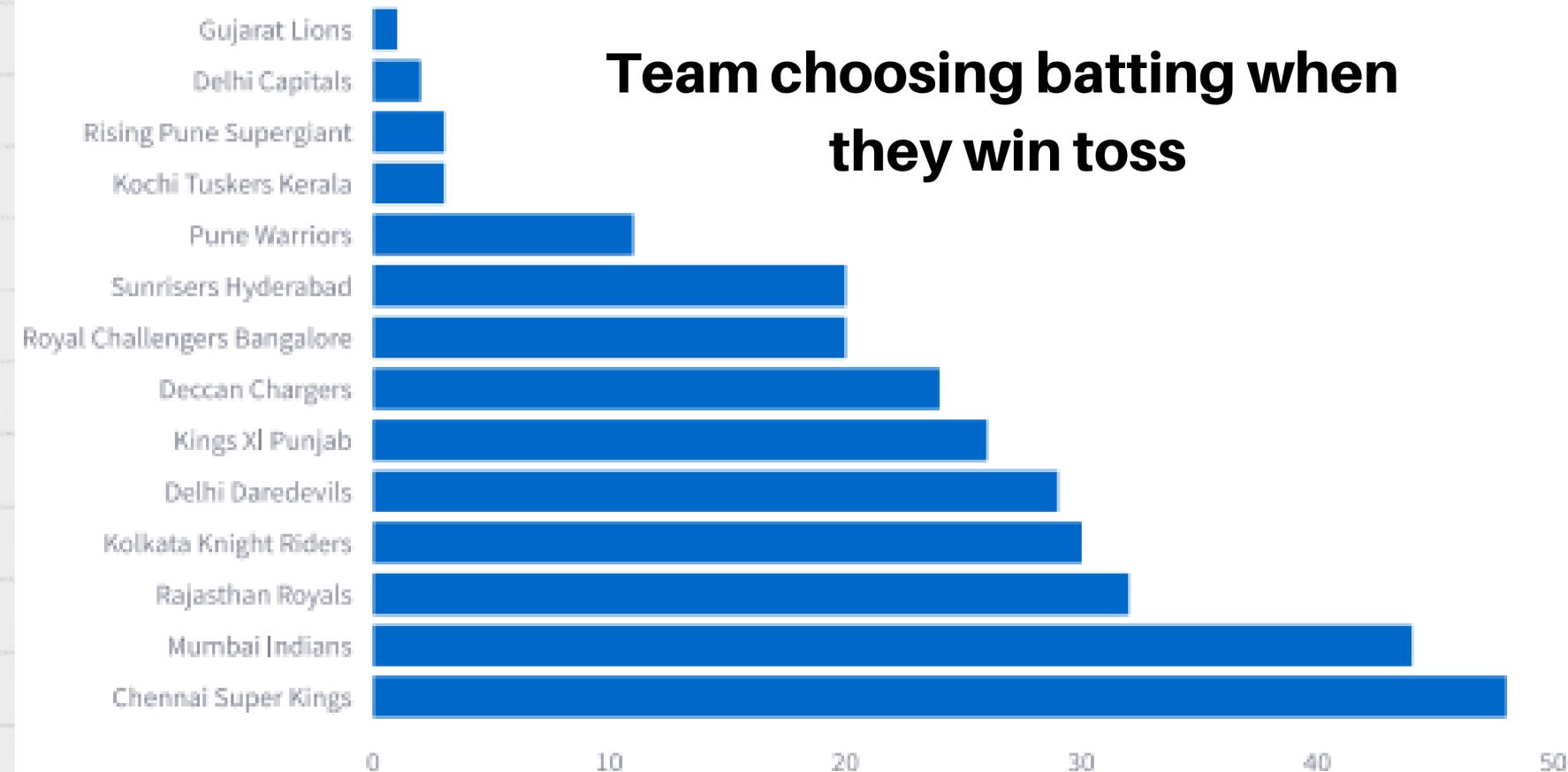
The program generates various charts like bar charts, pie charts, and scatter plots to analyze different aspects of IPL like the performance of top batsmen, teams' preference for batting or bowling after winning the toss, the most successful teams in terms of win count, top bowlers based on wickets, and more. The program also generates a player of the match award chart and the bowlers who are leaking more runs based on their economy rate.

This project can be used as a starting point for anyone who wants to analyze IPL data or similar sports data using Python. The data used in this project is publicly available and can be updated as new data becomes available.

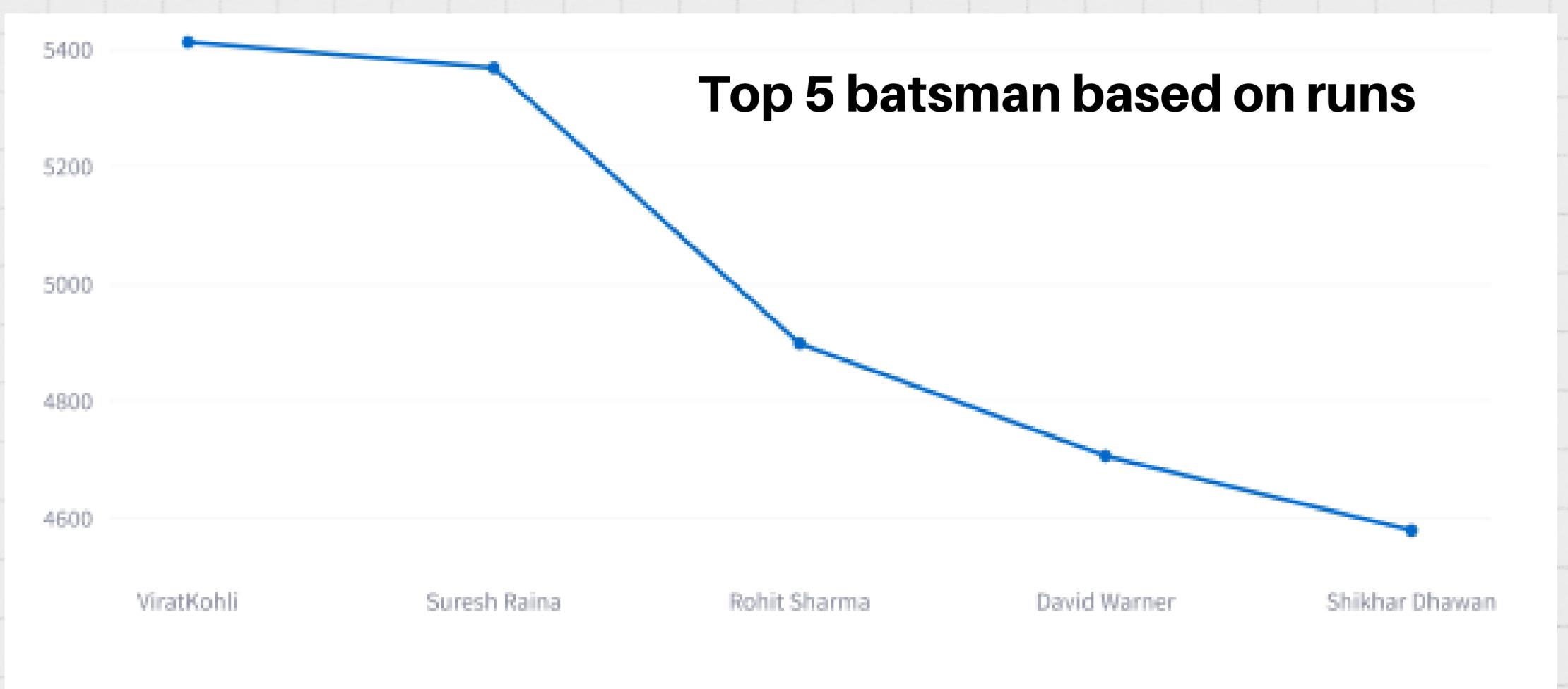
Team choosing bowling when they win toss



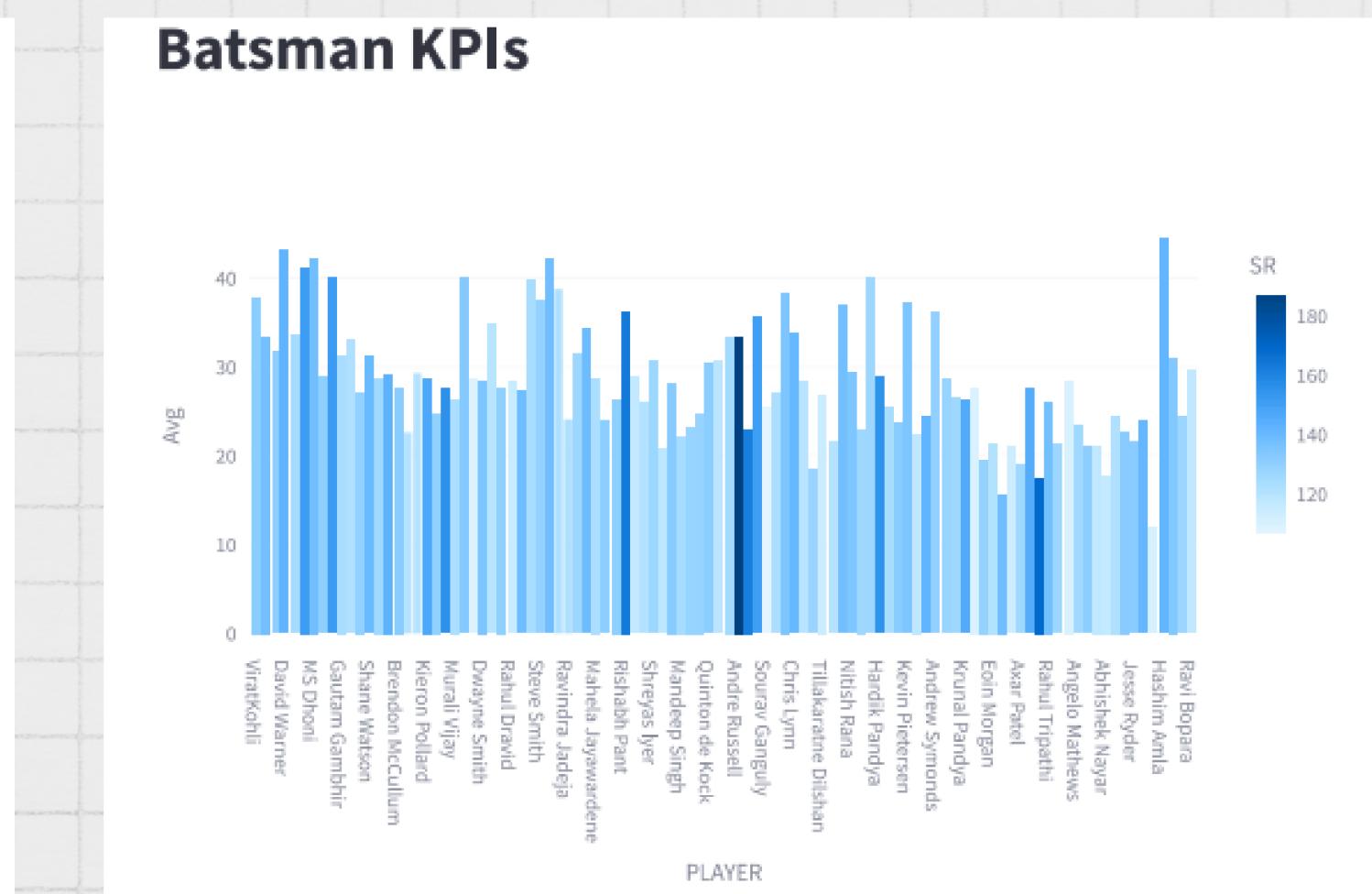
Team choosing batting when they win toss



Top 5 batsman based on runs



Batsman KPIs



21

22

20

19

18

17

16

15

CH Gayle

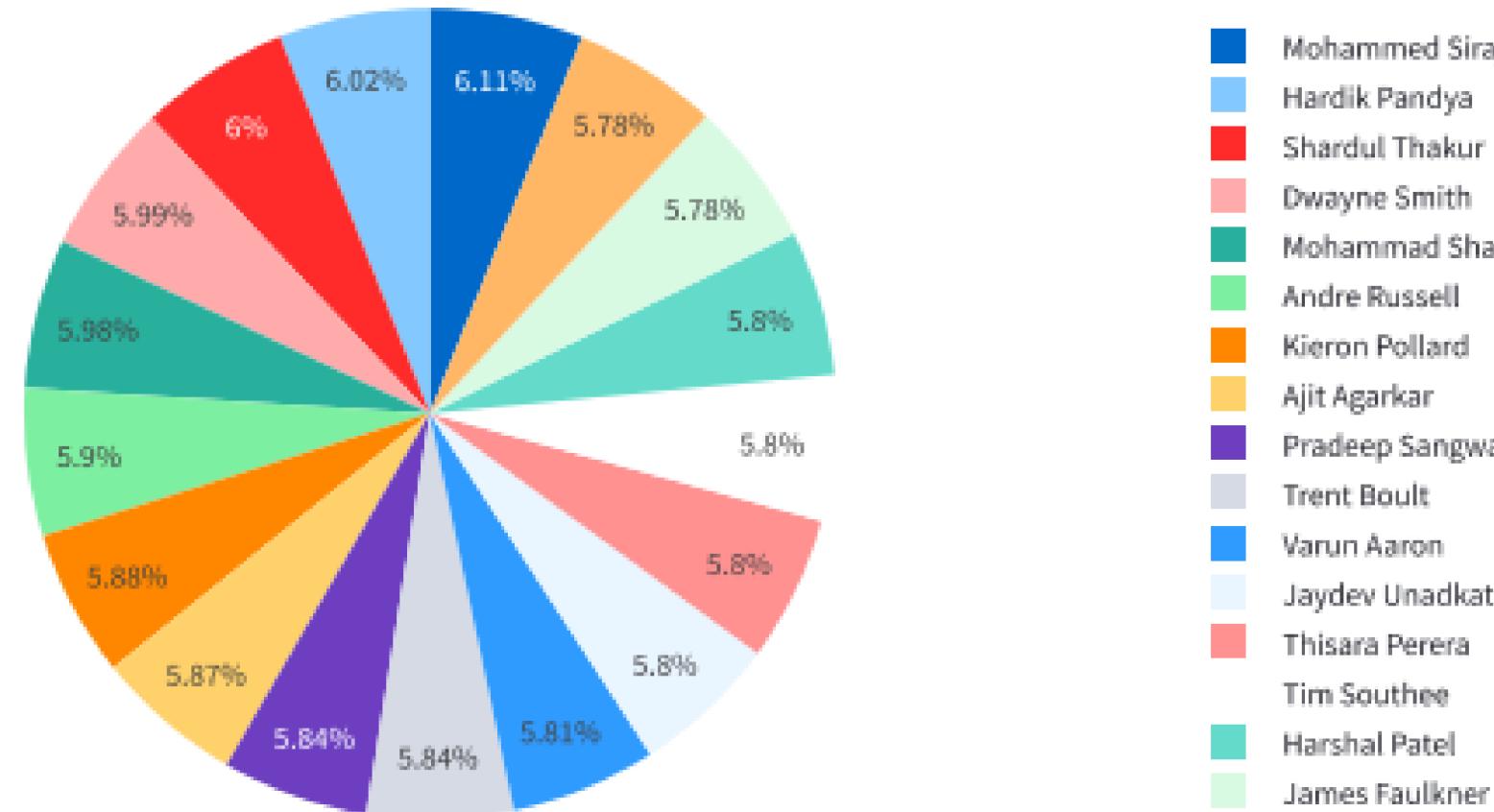
AB de Villiers

RG Sharma

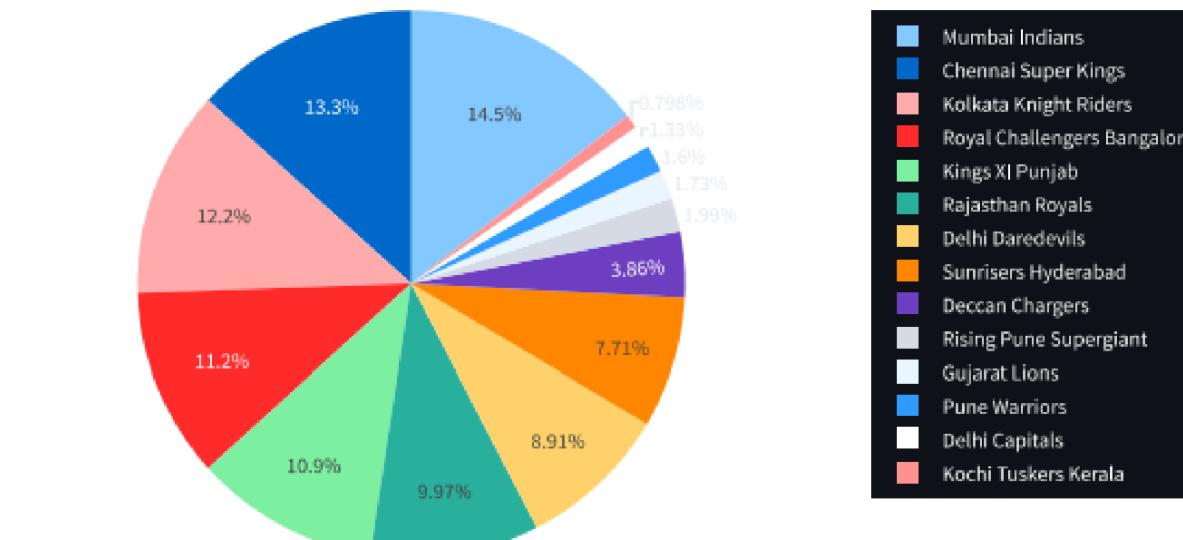
MS Dhoni

DA Warner

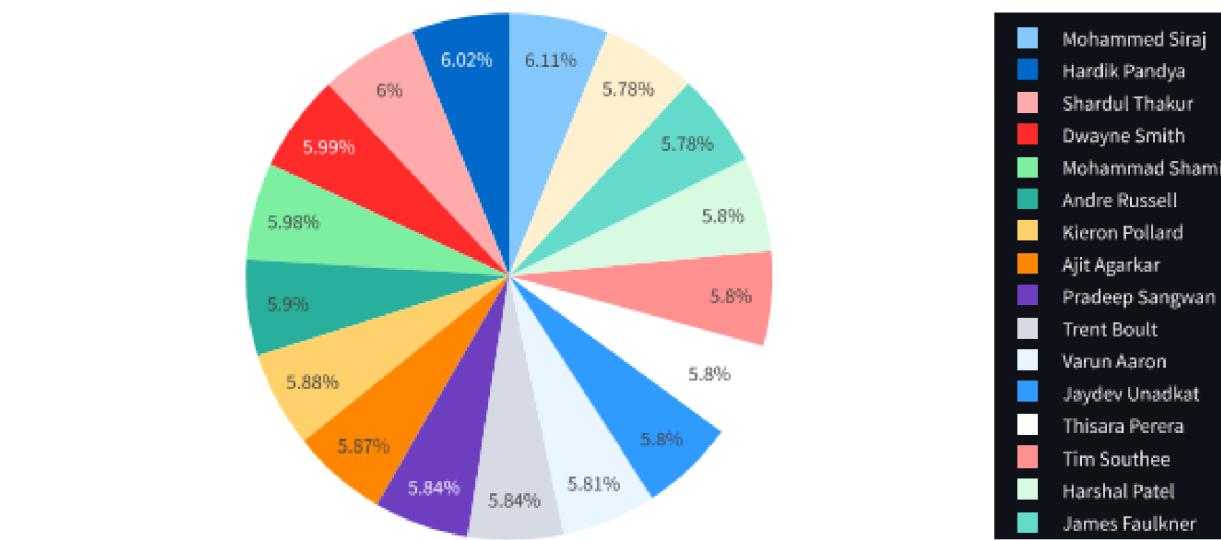
Bowlers leaking more runs



Most successful teams based on win count



Bowers who leaking more runs



MOTIVATION

The motivation behind this project could be to analyze the data related to the Indian Premier League (IPL) and extract insights from it. IPL is one of the most popular and widely watched T20 cricket leagues in the world. Therefore, analyzing data related to it could provide valuable insights into the performance of teams and players, which can be used to make informed decisions in the future. Additionally, the project could be a good opportunity for learning and practicing data analysis and visualization techniques.

LIBRARIES AND DATABASES

- 1. Streamlit:** Streamlit is a Python library used for building interactive web applications. It allows data scientists and developers to create applications quickly and easily using only Python scripts.
- 2. Numpy:** Numpy is a Python library used for scientific computing. It provides support for large, multi-dimensional arrays and matrices, along with a large collection of mathematical functions to operate on these arrays.
- 3. Pandas:** Pandas is a Python library used for data manipulation and analysis. It provides data structures for efficiently storing and manipulating large datasets, as well as a range of data analysis tools.
- 4. Plotly:** Plotly is a Python library used for creating interactive data visualizations. It provides a range of graphing tools and options, and can be used to create a variety of visualizations including line charts, scatter plots, bar charts, and more.
- 5. Github:** Github is a web-based platform used for version control and collaboration. In the code, Github has been used to host the datasets used in the analysis.

WORKING OF THE PROGRAM

The app is divided into several sections, each of which visualizes different aspects of IPL cricket data.

- The first section displays a bar chart that shows the average score (Avg) and the strike rate (SR) of the top 100 batsmen. The chart is interactive and can be filtered by selecting a particular player's name.**
- The second section displays a line chart that shows the runs scored by the top 5 batsmen. The chart is interactive and can be filtered by selecting a particular player's name.**
- The third section displays a bar chart that shows the number of times each team chose to bat first after winning the toss.**
- The fourth section displays a bar chart that shows the number of times each team chose to bowl first after winning the toss.**

- The fifth section displays a pie chart that shows the overall toss mapping, which is the combination of the number of times each team chose to bat or bowl first after winning the toss.
- The sixth section displays a pie chart that shows the number of times each team has won the IPL trophy.
- The seventh section displays a scatter plot that shows the top 5 players who have won the Player of the Match award the most number of times.
- The eighth section displays a line chart that shows the number of wickets taken by the top 5 bowlers.
- The ninth section displays a bar chart that shows the economy rate (Econ) and the number of wickets taken by the top 100 bowlers. The chart is interactive and can be filtered by selecting a particular player's name.
- The tenth section displays a pie chart that shows the bowlers who have an economy rate of 8.50 or higher.

CONSTRAINTS AND LIMITATIONS

- **Data source:** The code relies on a specific set of data sources for analysis. While these sources may be sufficient for the given analysis, they may not be comprehensive or up-to-date.
- **Visualizations:** The code primarily relies on bar charts, pie charts, and line charts for visualization. While these charts are effective for the given analysis, other visualization techniques may be more appropriate for different types of data or analysis.
- **Model limitations:** The code does not use any machine learning models or advanced statistical techniques. While this may be appropriate for the given analysis, more complex models may be necessary for more sophisticated analyses.
- **Interactive functionality:** The code primarily uses Streamlit for interactive functionality. While Streamlit is a powerful tool for creating interactive web applications, it may have limitations in terms of customization or compatibility with different devices or browsers.

FUTURE SCOPE

- **Data update and preprocessing:** The current code is based on data up to 2019. As new IPL seasons take place, the code will need to be updated to include the latest data. Additionally, the current code does not handle missing data or outliers in the dataset. Further preprocessing steps could be added to improve data quality.
- **Advanced statistical analysis:** The current code provides basic KPIs for batsmen and bowlers, but more advanced statistical analysis could be performed to gain deeper insights into player performance. For example, machine learning algorithms could be used to predict player performance or identify key factors that contribute to team success.
- **Social media integration:** The IPL is a highly popular event, and social media platforms like Twitter and Facebook are widely used to discuss and share updates about the tournament. The code could be extended to include social media data analysis, such as sentiment analysis of tweets related to the IPL or analysis of social media trends related to specific players or teams.

- **Will help in auction process:** Knowing which players are the top performers in the league based on runs, wickets, or player of the match awards can help teams identify potential targets for acquisition. Similarly, understanding the tendencies of teams to choose batting or bowling after winning the toss can provide valuable information to formulate strategies for the auction.

REFERENCE

- Streamlit documentation: <https://docs.streamlit.io/en/stable/>
- Exploratory Data Analysis on IPL: <https://www.kaggle.com/nitinkhandagale/eda-on-ipl-data-2008-2019>
- IPL Data Analysis using Python: <https://www.analyticsvidhya.com/blog/2020/09/ipl-data-analysis-2008-2019/>
- Building Data Apps with Streamlit - Tutorial: <https://www.youtube.com/watch?v=ZZ4B0QUHuNc>