IPL MEN’s MATCH DATA ANALYSIS

NAME : N. Jaishree

DESIGNATION: IT ENGINEER

## INDEX

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENTS** | **PAGE NO** |
|  | **ABSTRACT** |  |
| **1** | **INTRODCTION** | 1 |
| **2** | **SYSTEM REQUIRMENTS**   * 1. HARDWRAE SPECIFICATION   2. SOFTWARE SPECIFICATION   3. ABOUT THE SOFTWARE | 3 |
| **3** | **SYSTEM ANALYSIS**   * 1. MODULE DESCRIPTION | 15 |
| **4** | **SYSTEM TESTING**   * 1. UNIT TESTING   2. INTEGRATION TESTING   3. VALIDATION TESTING | 21 |
| **5** | **CONCLISION** | 23 |
| **6** | **LIMITATIONS** | 24 |
| **7** | **BIBLOOGRAPHY** | 27 |
| **8** | **APPENDIX**  SCREENSHOTS | 36 |

ABSTRACT

## ABSTRACT

This project analyzed the Indian Premier League (IPL) men’s match data to understand team performance across seasons. Using pandas for data manipulation and matplotlib and seaborn for visualization, the analysis focused on deriving key insights from match results.

**Data Collection and Preprocessing**: The dataset encompassed various IPL matches, detailing team performances and match outcomes. Data cleaning involved addressing missing values and ensuring accurate data formatting.

**Analysis**: The analysis highlighted the top-performing teams each season by calculating the number of wins. Season-wise performance was assessed to identify dominant teams and track performance trends over time.

**Visualizations**:

* **Matplotlib and Seaborn**: Bar plots depicted wins by top teams for each season, while line plots showed performance trends. Heatmaps illustrated win distributions across teams and seasons.
* **Tableau and Power BI**: Interactive dashboards were created for deeper data exploration, enhancing the ability to spot trends and outliers.

**Key Findings**:

* **Dominant Teams**: Identified the most successful teams across different seasons, showcasing their consistency.
* **Performance Trends**: Revealed trends and fluctuations in team performance over the years.

INTRODUCTION

## INTRODUCTION

The Indian Premier League (IPL) is one of the most popular and lucrative professional Twenty20 cricket leagues globally. Founded in 2008, the IPL has grown into a major sporting event with significant fan engagement and media coverage. The league features franchises representing various cities and regions in India, competing annually in a highly competitive format.

The IPL men's match dataset provides comprehensive information about matches played in the league, including details such as match dates, teams, venues, and outcomes. This dataset is crucial for analyzing team performances, player statistics, and match trends. It typically includes various attributes such as:

* **Match ID**: A unique identifier for each match.
* **Season**: The year or edition of the IPL in which the match took place.
* **Batting and Bowling Teams**: The teams involved in each match, distinguishing between the team batting first and the team bowling.
* **Match Outcome**: Information on the winner, match result, and any relevant scores.
* **Date and Venue**: Details about when and where the match was played.

Analyzing this dataset helps uncover patterns in team performances over time, identify trends and anomalies, and provide insights into the competitive dynamics of the league. With rich data on match results and team statistics, stakeholders—including teams, analysts, and fans—can gain a deeper understanding of the league's historical and current performance dynamics. This analysis is not only valuable for strategic decision-making but also enhances the viewing experience by providing detailed performance metrics and historical context.

**SYSTEM REQUIREMENT**

## SYSTEM REQUIRMENTS

#### hardware specification

* RAM :4GB
* PROCESSOR: Intel cories
* KEYBOARD: standard Ps2

#### Software Specification

* OPERATING SYSTEM : Microsoft Windows 10
* FRONT END : Tableau and Powerbi
* BACK END : Pandas, Matplotlib and Seaborn

#### About the software

#### FRONT END:

#### TABLEAU

**Tableau** is a leading data visualization software used for business intelligence and data analytics. Renowned for its user-friendly interface, Tableau enables users to create interactive and visually compelling dashboards without needing to write code. It supports a wide variety of data sources, including spreadsheets, databases, cloud services, and big data platforms, allowing seamless connectivity and integration.

Tableau's key features include powerful visualization capabilities, where users can create charts, maps, and complex graphics that highlight trends, patterns, and insights within the data. It also supports real-time data analysis, enabling up-to-date reporting, which is crucial for timely decision-making.

With Tableau's advanced analytics, users can perform calculations, forecast trends, and conduct statistical analysis directly within the platform. The software also allows for easy sharing and collaboration, with options to publish dashboards on Tableau Server, Tableau Online, or Tableau Public, facilitating secure access and teamwork.

Tableau Prep, an additional tool, helps in data cleaning and preparation, making it easier to shape data before analysis. The Tableau ecosystem is supported by a strong user community and extensive resources, making it accessible to both beginners and experts.

Overall, Tableau is an essential tool for organizations seeking to leverage their data for strategic insights, making it indispensable for fields ranging from business intelligence to healthcare and finance. Its versatility and ease of use have made it a go-to solution for data professionals worldwide.

* + 1. **POWER BI**

**Power BI** is a business analytics tool developed by Microsoft that enables users to visualize and share insights from their data. It is widely used for creating interactive reports and dashboards that help organizations make informed decisions. Power BI connects seamlessly with a broad range of data sources, including Excel, SQL Server, cloud services, and online databases, allowing users to integrate data from multiple platforms.

One of Power BI's standout features is its ability to handle real-time data, making it ideal for monitoring key metrics and performance indicators. Users can create visually appealing and dynamic reports using a variety of chart types, maps, and other visualization tools. The drag-and-drop interface makes it accessible even to those without advanced technical skills.

Power BI's collaboration features allow users to share reports and dashboards within an organization securely. Through Power BI Service, reports can be published to the web or shared across mobile devices, ensuring stakeholders have access to the latest data insights anytime, anywhere. Additionally, Power BI integrates with other Microsoft tools, such as Excel and Azure, enhancing its functionality and ease of use within the Microsoft ecosystem.

Overall, Power BI is a versatile and powerful tool for data analysis, offering organizations the ability to harness their data for strategic decision-making, from small businesses to large enterprises. Its combination of ease of use, flexibility, and robust features makes it a leading choice in the business intelligence space.

#### 2.3.1 PANDAS

## Pandas is a powerful Python library used for data manipulation and analysis. It provides data structures and functions needed to work with structured data seamlessly. Its primary data structures are:

## DataFrame: A 2-dimensional, size-mutable, and potentially heterogeneous tabular data structure with labeled axes (rows and columns). It's similar to a table in a database or an Excel spreadsheet.

## Series: A 1-dimensional labeled array capable of holding any data type.

## Key Features:

## Data Cleaning: Functions for handling missing data, removing duplicates, and filtering data.

## Data Transformation: Methods for aggregating, merging, reshaping, and pivoting data.

## Time Series Analysis: Tools for handling time series data, such as date range generation and frequency conversion.

## Efficient Operations: Vectorized operations that are optimized for performance.

## 2.3.2 MATPLOTLIB

**Matplotlib** is a comprehensive Python library for creating static, animated, and interactive visualizations. It is widely used for plotting and visualizing data in a variety of formats.

**Key Features**:

* **Flexibility**: Allows for detailed customization of plots, including color, size, and style.
* **Variety**: Supports a wide range of plot types, including line plots, bar charts, scatter plots, histograms, and more.
* **Integration**: Works well with other libraries such as pandas and numpy.
  + 1. **SEABORN**

**Seaborn** is a Python visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics. It integrates well with pandas DataFrames and is designed for data visualization with built-in themes and color palettes.

**Key Features**:

* **Statistical Plots**: Built-in functions for common statistical plots like heatmaps, violin plots, and pair plots.
* **Aesthetic Styles**: Offers improved visual aesthetics and easier customization compared to Matplotlib.
* **DataFrames Integration**: Directly works with pandas DataFrames, making it easy to plot data without extensive data transformation.

# SYSTEM ANALYSIS

## SYSTEM ANALYSIS

* 1. **MODULE DESCRIPTION**

A module is a collection of source files and build settings that let you divide your project into discrete units of functionality. Your project can have one or many modules, and one module can use another module as a dependency. You can independently build, test and debug each module.

* Data Cleaning
* Data Preprocessing
* Data Visualization
* Tableau Dashboard
* Powerbi Dashboard

### Data Cleaning

### Data cleaning is the process of identifying and correcting errors, inconsistencies, and inaccuracies in a dataset. This involves handling missing values, removing duplicates, correcting data entry errors, and standardizing formats. It may also include outlier detection, ensuring data types are correct, and addressing inconsistencies across records. Data cleaning improves the quality of the data, making it more reliable and accurate for analysis. The process is essential for ensuring that the insights drawn from the data are valid and trustworthy, ultimately leading to better decision-making and analysis.

### Data Preprocessing

**Data preprocessing** involves transforming raw data into a clean and usable format for analysis. Key methods include data cleaning (handling missing values, removing duplicates), data transformation (normalization, scaling), and data reduction (dimensionality reduction, feature selection). It also involves encoding categorical variables, splitting datasets into training and testing sets, and handling outliers. These steps ensure that the data is consistent, accurate, and in a suitable format for machine learning models or statistical analysis, improving the quality and efficiency of the analysis process. Proper data preprocessing is crucial for accurate and reliable results.

### Data Visualization

### Matplotlib and Seaborn are essential Python libraries for data visualization. Matplotlib offers a flexible framework for creating a wide variety of plots, including line charts, bar charts, and scatter plots, with detailed customization options. Seaborn builds on Matplotlib, providing a simpler interface for creating visually appealing statistical plots like heatmaps, violin plots, and pair plots with less code. It integrates well with Pandas and offers beautiful default styles and color palettes. Together, they enable the creation of informative and attractive visualizations, making them indispensable for data analysis and exploration.

### Tableau

This Tableau story provides an in-depth analysis of IPL men’s matches, offering insights into team and player performances across various seasons. The analysis begins with identifying the **top 5 teams with the highest runs**, showcasing their dominance over time. Conversely, it highlights the **last 5 teams with the lowest runs**, giving a clear contrast in performance.

Focusing on individual achievements, the story delves into the **top 10 batsmen with the most runs**, emphasizing the key players who have consistently contributed to their teams. This is further refined by analyzing **top batsmen by team**, providing a detailed view of standout performers within each squad.

The story also explores the **number of extras separated by type**, offering insights into how teams manage additional runs. The **venue with the maximum matches hosted** is identified, revealing the most popular grounds. A thorough examination of the **distribution of wicket types** gives an understanding of how wickets fall in different scenarios.

Finally, all these elements culminate in a **comprehensive IPL men’s match analysis dashboard**, bringing together these insights to provide a holistic view of IPL performance trends.

**POWER BI**

In Power BI, you've developed a dashboard similar to your Tableau version, but with added interactivity through slicers. The dashboard starts with **Basic Details**, displaying key statistics like the count of seasons, total runs scored off the bat, total wickets taken, and total innings played. A calculated field is used to determine the total number of unique teams, providing a foundational overview.

You’ve highlighted the **Top 5 Teams** based on performance metrics like wins or total runs, as well as the **Last 5 Teams** for a comparative analysis. The **Top Matches by Venue** section focuses on venue-specific performance, showcasing significant matches.

A detailed analysis of **Different Wicket Types** is included, showing how wickets were taken across various matches. To enhance user interaction, you've added slicers for **Wicket Type** and **Venue**, allowing users to filter the data and visualizations based on these categories.

Lastly, bar charts provide insights into **Wides, Innings, Balls, and No-balls**, offering a comprehensive view of gameplay aspects. The added slicers make the dashboard more interactive, enabling users to explore the data in greater depth, tailored to specific wickets or venues. This Power BI dashboard combines detailed analysis with user-friendly features, making it a powerful tool for IPL match data exploration.

# SYSTEM TESING

## SYSTEM TESTING

#### Unit Testing

Unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which is a unit method, which may belong to base/super class, abstract class or derived/child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/fake objects are used to assist in unit testing. Testing comprises the set of tests performed by an individual programmer prior to integration of the unit into a larger system. There are four categories of test that can be performed on a program unit.

#### Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing tasks as its input modules that have been unit tested, groups them in larger aggregate, applies tests defined in an integration test plan those aggregates, delivers as its output and integrated system ready for system testing.

#### Validation Testing

Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e, the software meets the user requirements, not as specifications artifacts or as needs of those who will operate the software only; but, as the needs of all the stakeholders (such as users, operators, administrators, managers, investors, etc.). There are two ways to perform software validation: internal and external. During internal software validation it is assumed that the goals of the stakeholders were correctly understood and that they were expressed in the requirements artifacts precise and comprehensively. If the software meets the requirements specification, it has been internally validated. External validation happens when it is performed by asking the stakeholders if the software meets their needs external validation can be discrete or a continuous flow.

# CONCLUSION

## CONCLUSION

In Conclusion, the IPL Men’s Match dataset was thoroughly analyzed using a combination of data cleaning, exploratory analysis, and visualization techniques. Pandas was employed to clean the data and remove null values, ensuring accuracy. Matplotlib provided initial visual insights into key trends, laying the groundwork for deeper analysis. Tableau was then used to create an interactive dashboard, showcasing team performance, match outcomes, and wicket types. Finally, Power BI enhanced the analysis with added slicers for wicket types and venues, allowing for tailored data exploration. This approach effectively transformed raw data into actionable insights.

# LIMITATIONS

## LIMITATIONS

Each and every project got certain limitation as well, it is very difficult to study every aspect is change of every in it.be comp

Some of the limitation of project one

* **Loss of Data**: Removing null values might discard potentially useful information.
* **Error Propagation**: Initial data cleaning errors can affect subsequent analysis.
* **Manual Intervention**: Data cleaning and transformation can be time-consuming and prone to human error.
* **Tool Limitations**: Excel has limited capabilities for handling large datasets and complex analytics.
* **Data Integrity**: Converting datasets to Excel can lead to formatting issues and data loss.
* **Scalability**: Excel may struggle with large-scale data, leading to performance issues.
* **Lack of Automation**: Excel's limited automation features can result in repetitive manual tasks.

# BIBLIOGRAPHY

## 7 BIBLIOGRAPHY

**LINKS**

* [**https://kaagle.com**](https://kaagle.com)
* [**https://pandas.pydata.org/docs**](https://pandas.pydata.org/docs)
* [**https://matplotlib.org/stable/contents.html**](https://matplotlib.org/stable/contents.html)
* [**https://help.tableau.com/current/pro/desktop/en-us/tableau\_overview.htm**](https://help.tableau.com/current/pro/desktop/en-us/tableau_overview.htm)
* [**https://docs.microsoft.com/en-us/power-bi/**](https://docs.microsoft.com/en-us/power-bi/)
* [**https://www.python.org/**](https://www.python.org/)
* [**https://support.microsoft.com/en-us/excel**](https://support.microsoft.com/en-us/excel)

### Book Reference

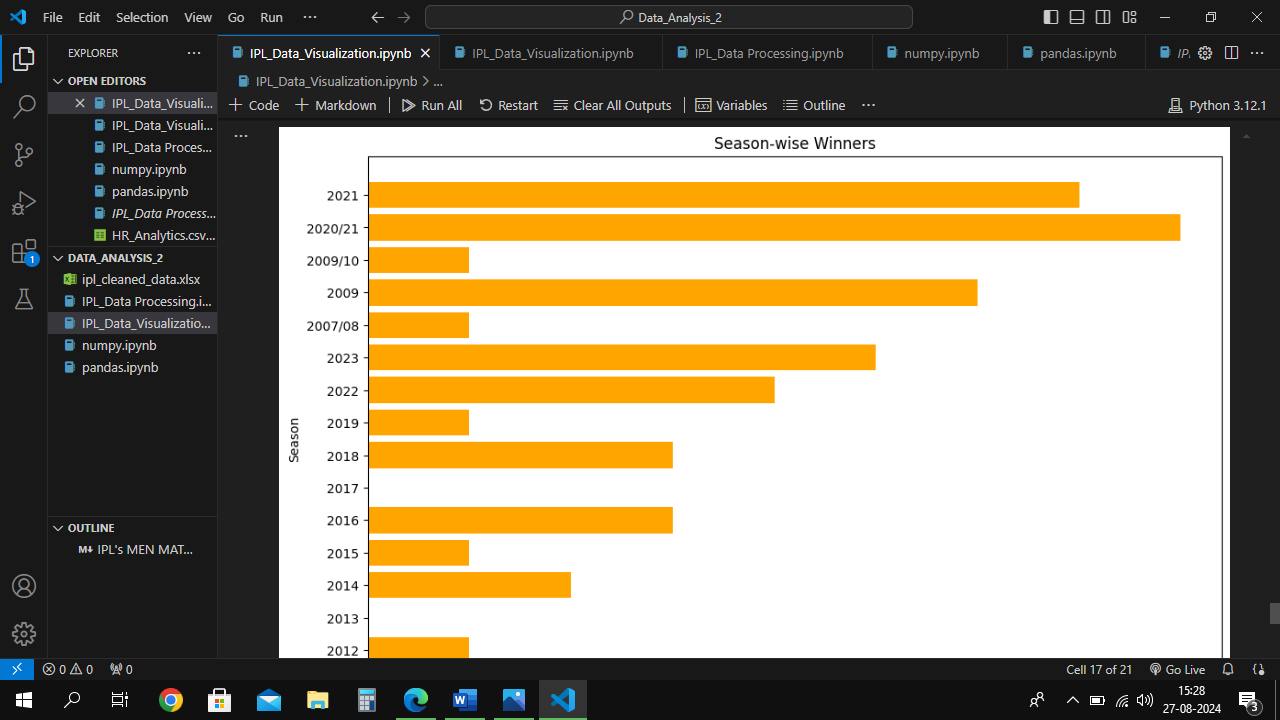
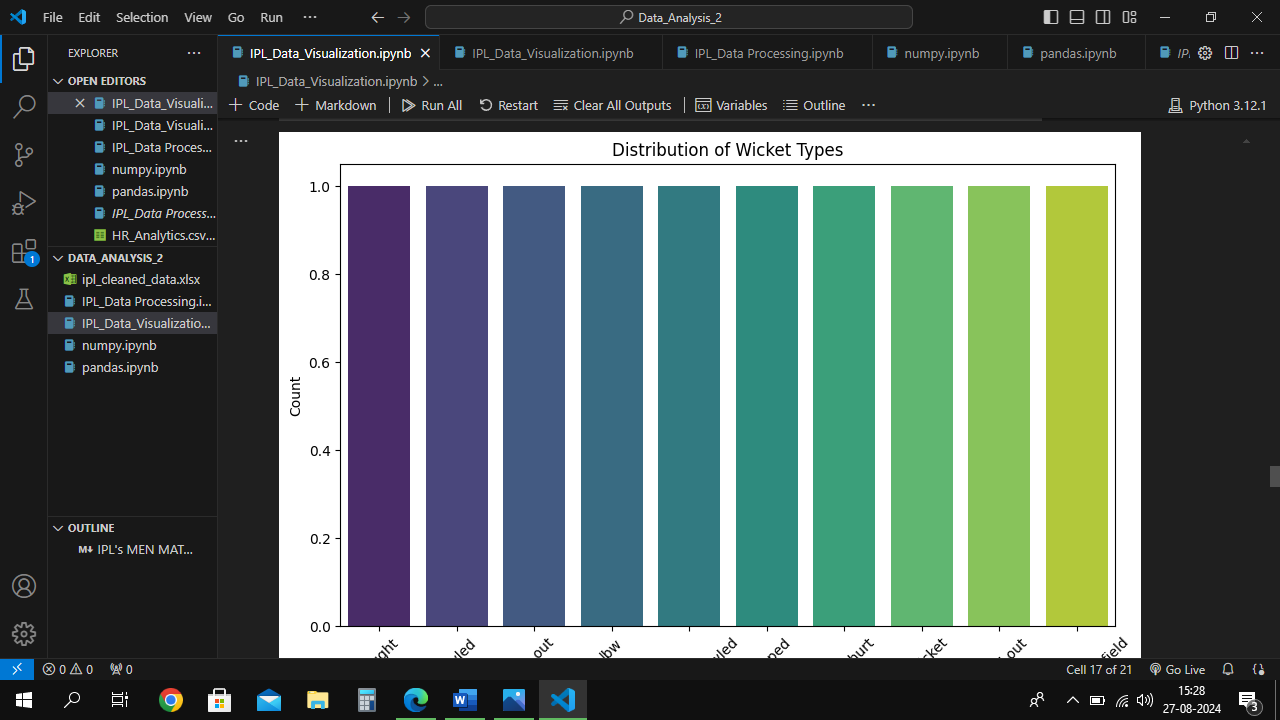
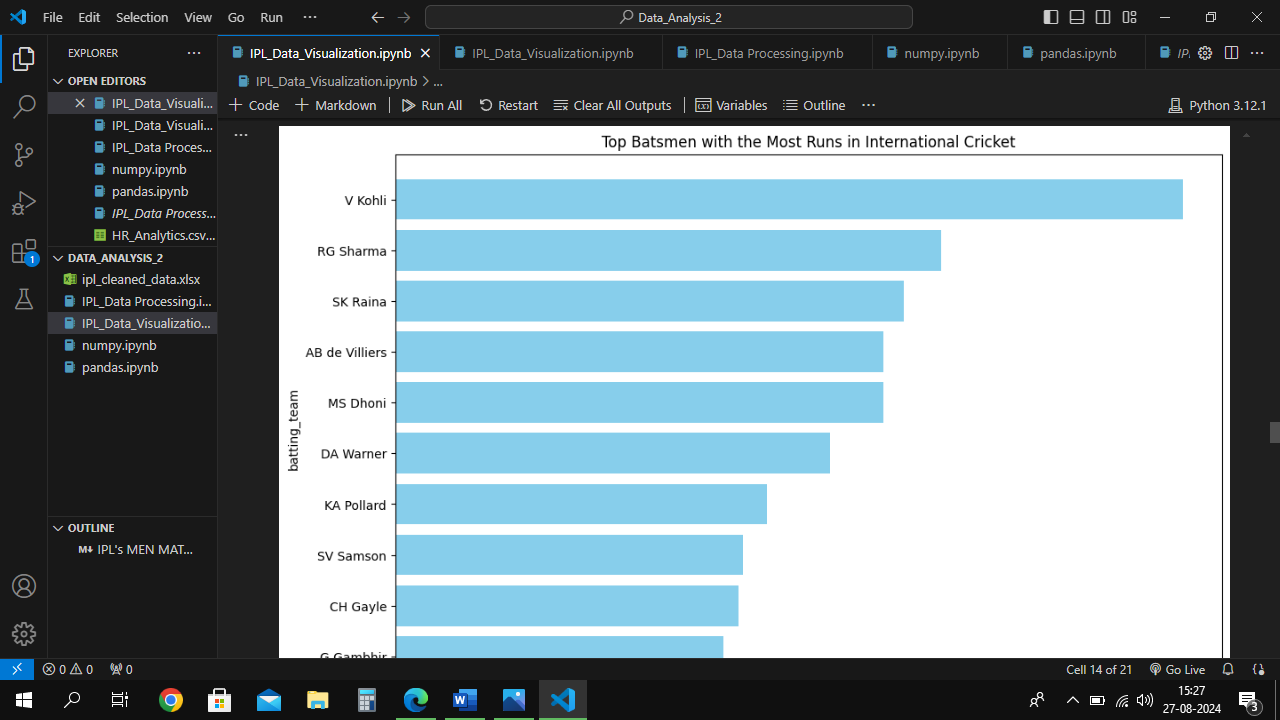
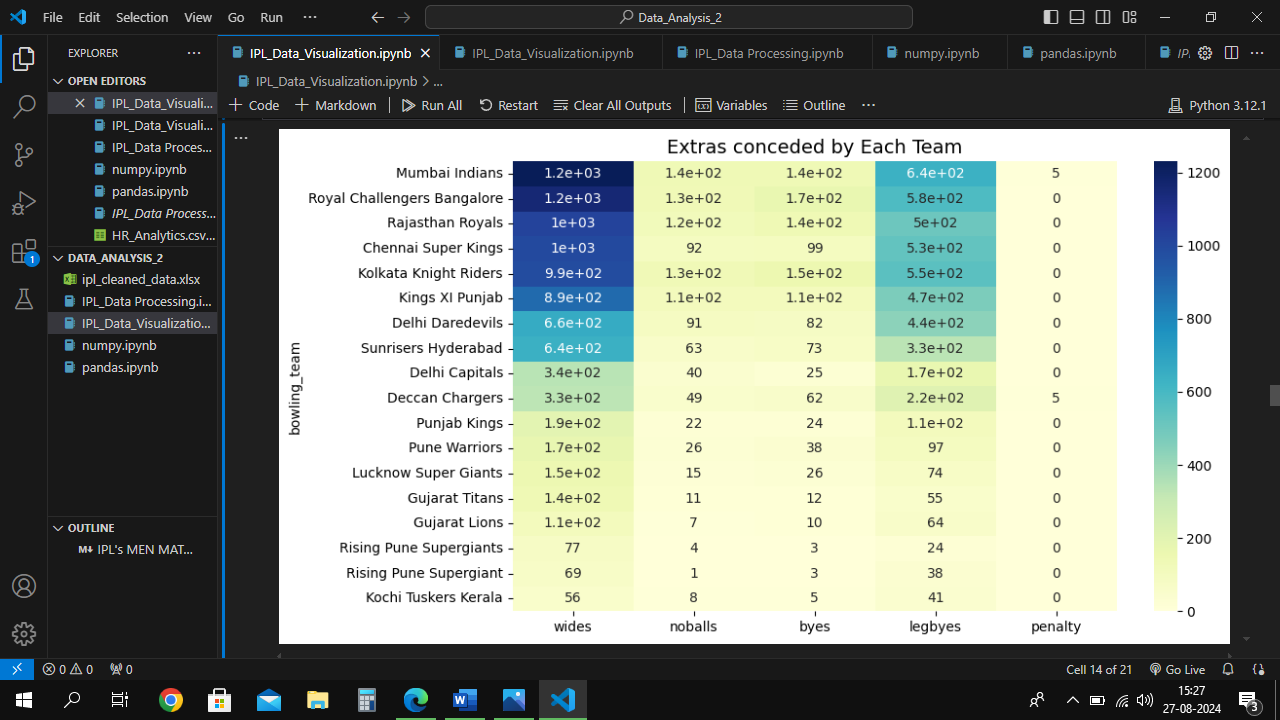
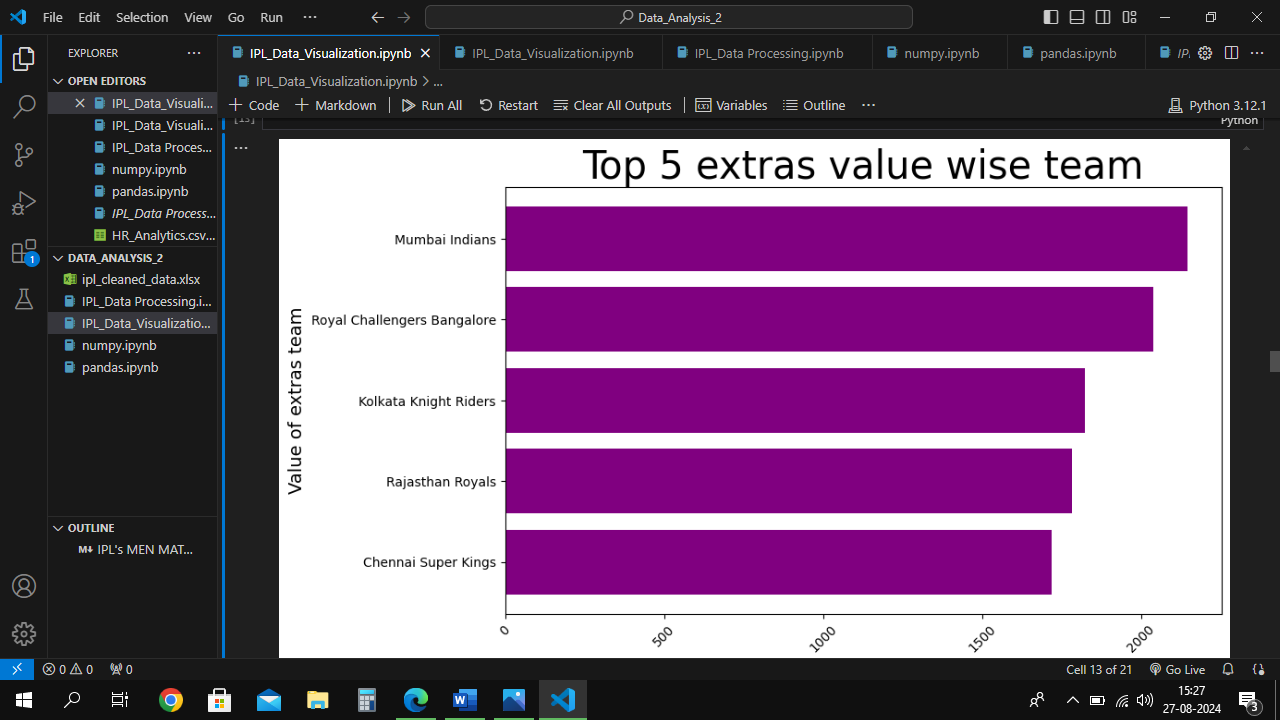
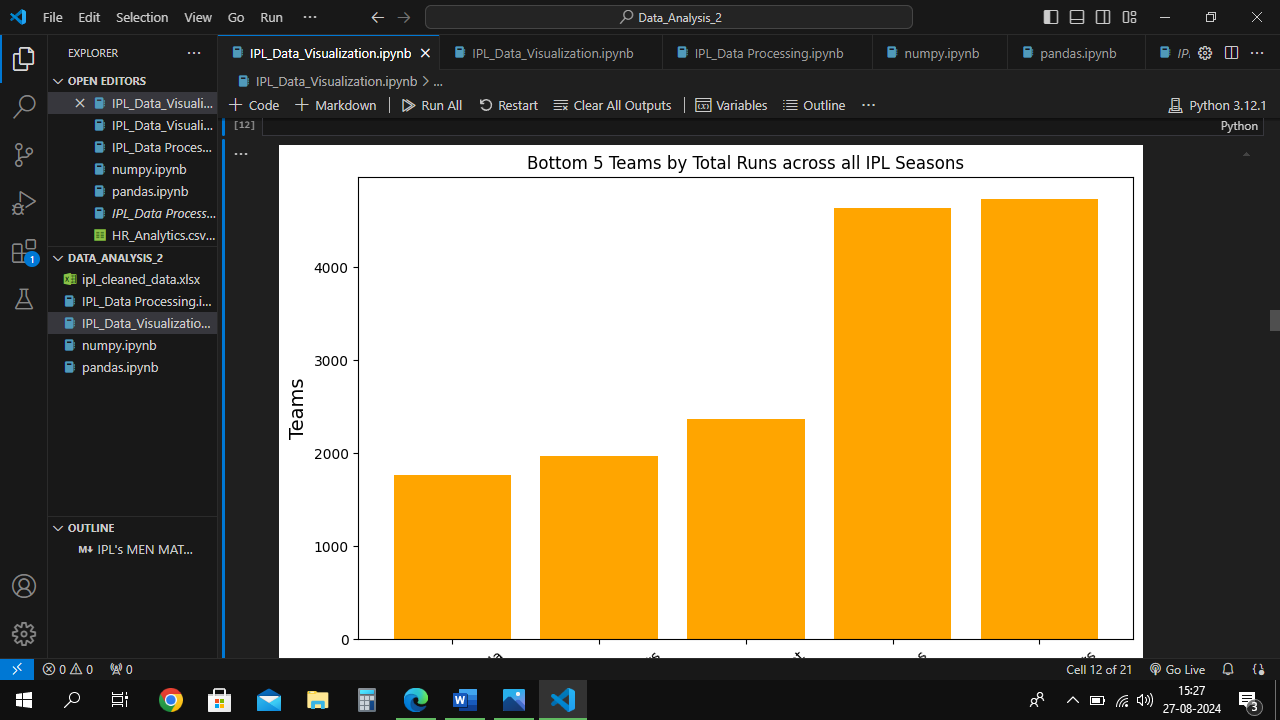
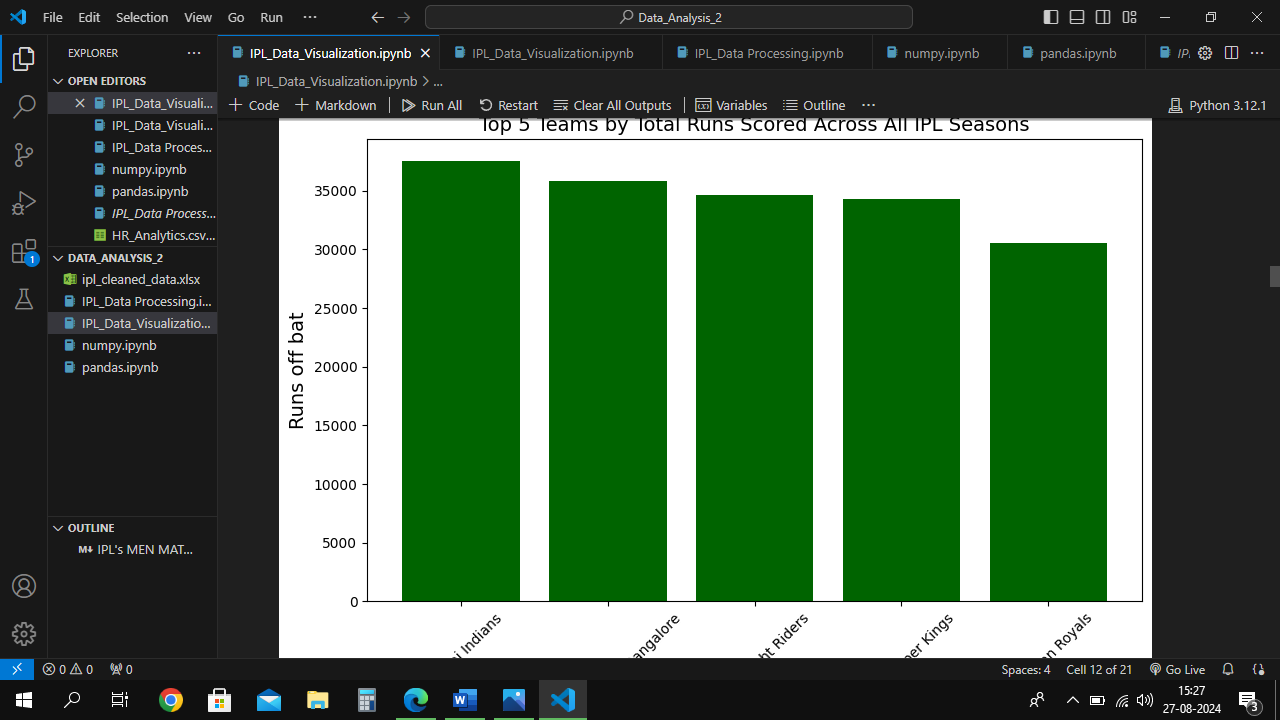
* **Python for Data Analysis** by Wes McKinney
* **Practical Statistics for Data Scientists** by Peter Bruce and Andrew Bruce
* **Storytelling with Data: A Data Visualization Guide for Business Professionals** by Cole Nussbaumer Knaflic
* **Tableau 10 for Data Science** by Kirill Eremenko and SuperDataScience Team.
* **Power BI Data Analytics and Visualization for Beginners** by Steve Williams

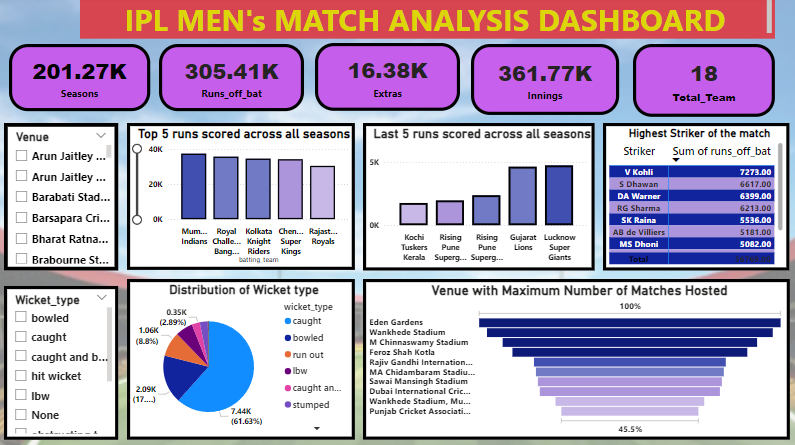
**APPENDIX**

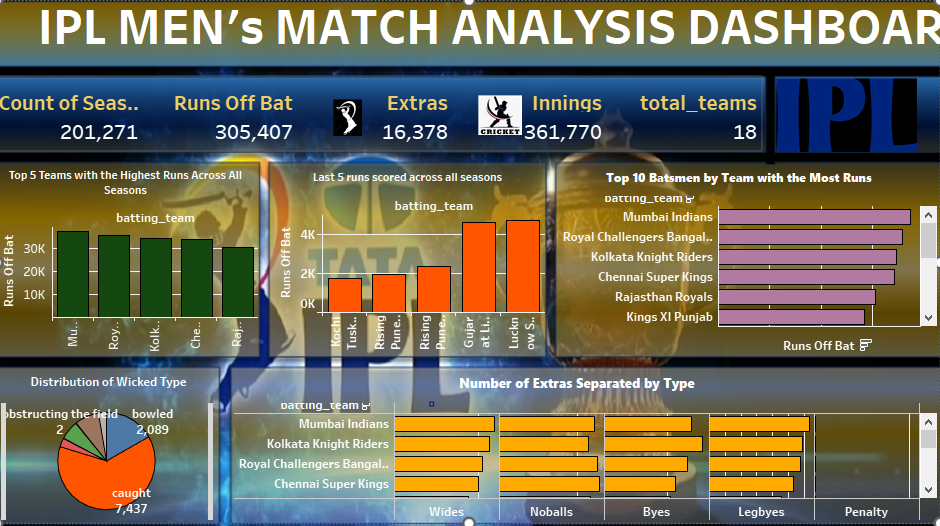
## 8 APPENDIX

**8.1 SCREENSHOTS**

**DATA VISUALIZATION**



**POWERBI DASHBOARD**

**TABLEAU DASHBOARD**T

**TABLEAU STORY**

