Powerplays and Predictions: Unlocking IPL Intelligence

Submission 01: Data Visualization using Pandas dataframe

Ashwin Shastry Paturi, Akarsh Lakshmana, Divakar Reddy Ravi, Sara Verkiyani

Team-09, CS506\_02\_IN: Programming for Computing, Spring 2025, MS-Computer Science, City University of Seattle

[paturiashwinshastry@cityuniversity.edu](mailto:paturiashwinshastry@cityuniversity.edu), [lakshmanaakarsh@cityuniversity.edu](mailto:lakshmanaakarsh@cityuniversity.edu), [ravidivakarreddy@cityuniversity.edu](mailto:ravidivakarreddy@cityuniversity.edu), [verkiyanisara@cityuniversity.edu](mailto:verkiyanisara@cityuniversity.edu)

**Abstract**

In this module, we perform an exhaustive exploratory data analysis (EDA) of Indian Premier League (IPL) match data using Python's Pandas library. From tidy match-level data for several seasons, we look at some key patterns of match results, toss, team performance, and margin of victory. Actionable match trends and team strategy insights are derived with concise DataFrame operations - filtering, grouping, and aggregation. The analysis applies Seaborn and Matplotlib plots side-by-side to capture significant findings expressed clearly. Pandas' key discussions, such as apply(), groupby(), and date manipulation, are all shown within example applications. Submission lays the basis for machine learning and deep learning analyses to be carried out in subsequent modules by providing a solid foundation on the significance of understanding data in sports analysis.

**Keywords:** Pandas, IPL, Exploratory Data Analysis, DataFrame, Python, Sports Analytics, Toss Impact, Win Margins, Data Visualization, T20 Cricket

**1. Introduction**

In this module, we perform an extensive exploratory data analysis (EDA) of Indian Premier League (IPL) match data using the Pandas library. IPL, a significant T20 cricket competition, provides a rich dataset with match results, player performances, team strategies, and logistics. This analysis aims to demonstrate key Pandas DataFrame operations by applying them using actual structured sports data.



This work aims to identify patterns in match results, team victories, margins of victory, length of the match, and player influence through core Pandas functions. The module also offers elegant and interpretable data visualization through Seaborn and Matplotlib to support our results.

**2. Overview of the Dataset:** The dataset contains match-level data for some IPL seasons. It has over 70 columns and includes information on:

* Team-level data: home team, away team, scores, captains, playing XIs
* Match result information: winner, toss decision, result type, player of the match
* Time and place: start/end dates, venue name, umpires
* Scoring statistics: runs, wickets, boundaries
* Metadata: result descriptions, match days, super over information

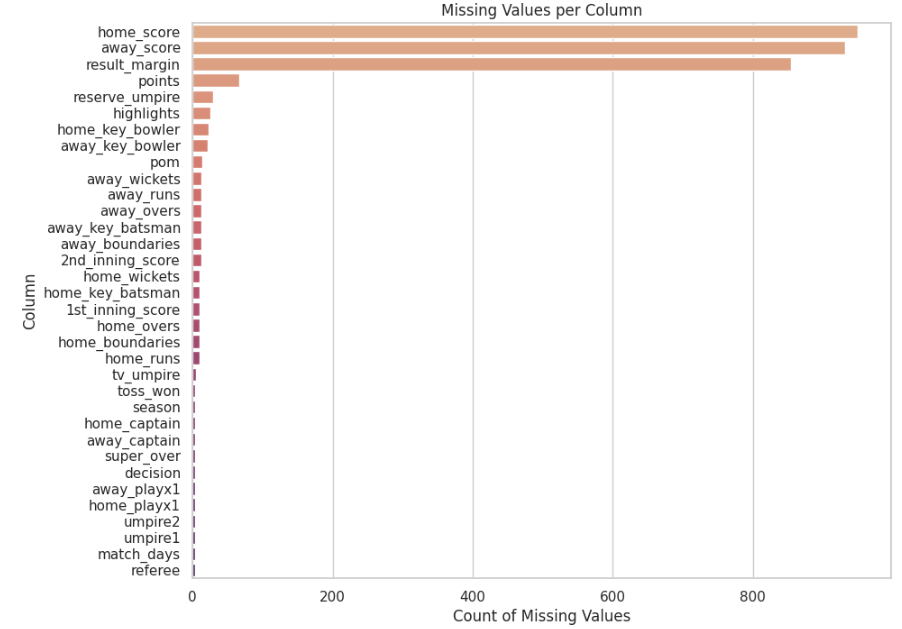
This dense dataset provides an ideal environment to work with and demonstrate Pandas' functionality, ranging from basic I/O to more involved group-by operations and aggregations.

**3. Pandas Concepts and Their Applications**

**3.1 Data Preprocessing and Cleanup**

We began by preparing and cleaning the data with:

* pd.to\_datetime() to convert start\_date and end\_date columns into datetime objects.
* dropna() to handle missing match winners (no result).
* apply() together with lambda functions to extract result types from descriptive results like "Team A won by five wickets" to standardized categories: "By Wickets", "By Runs", "Tie", "No result".

****

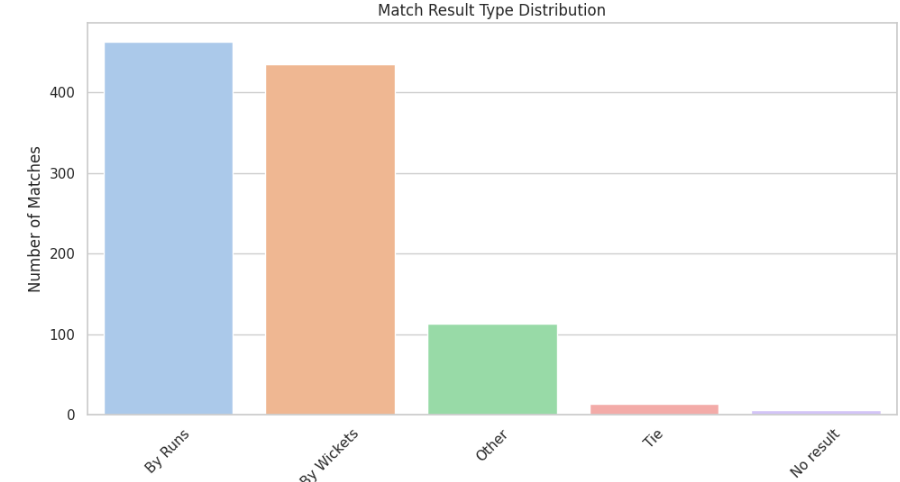
**The above figure summarizes the missing values presented within the selected dataset.**

This action helped us clean the data and work with a more refined dataset. All the missing values were replaced with null.

The above preprocessing cleaned out any dirt that could have affected the following analysis.

**3.2 Match Result Analysis**

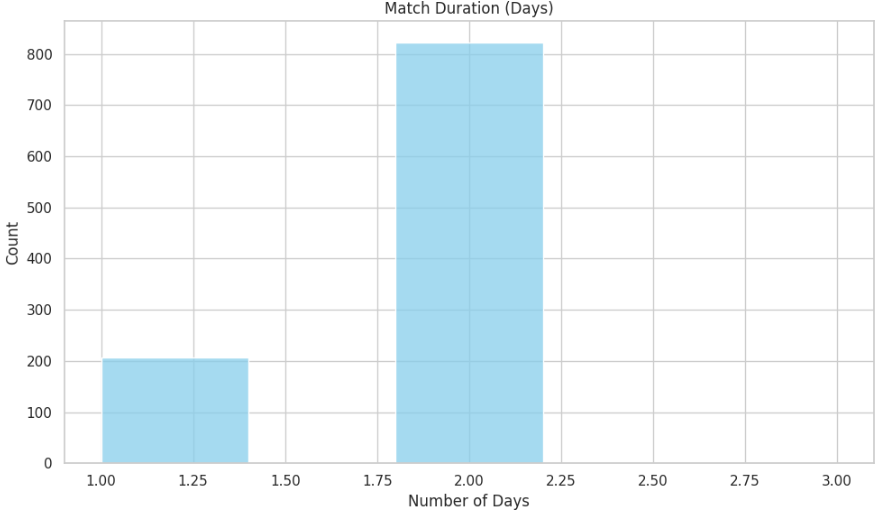
By utilizing the value\_counts() and groupby() functions, we counted and grouped the number of runs, wickets, or ties won in matches. This helped us prepare a summary distribution of match results using Seaborn bar plots.



This section shows how value frequency and categorical aggregation are done in Pandas.

**3.3 Match Duration Analysis**

We calculated match duration in days by subtracting the end\_date and start\_date in the computation to determine delayed or rain-affected games. This was achieved through datetime subtraction and the .dt.days accessor.



A histogram showed that most matches end on the same day, but some go beyond due to external factors, as mentioned in the above figure.

**3.4 Victory Margin Analysis**

**3.4.1 Extracting Win Margins**

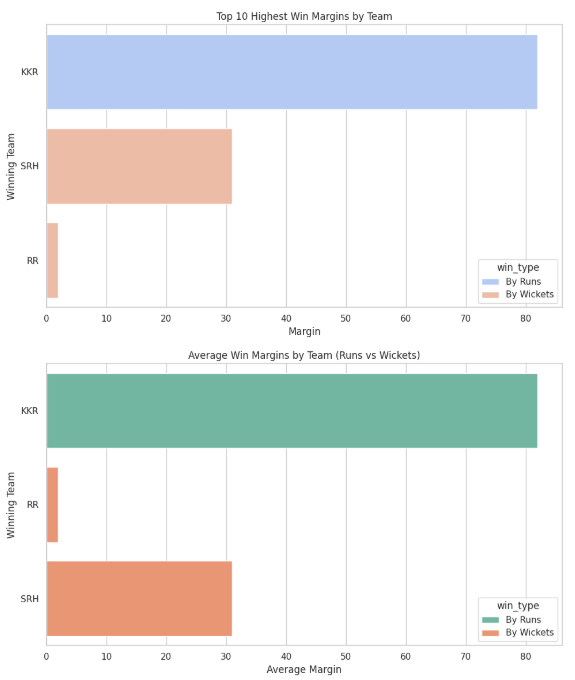
We categorized match outcomes into two groups:

**Runs-based wins (defending team wins)**

Wickets-based victories (chasing team victory)

Using boolean filters and assign(), we created new columns like win\_margin, winner\_team, and win\_type.

The image below shows the histogram representation for three teams with extremely varied win-loss margin differences. This was represented and evaluated using the functions groupby() and agg() to give the average scores.

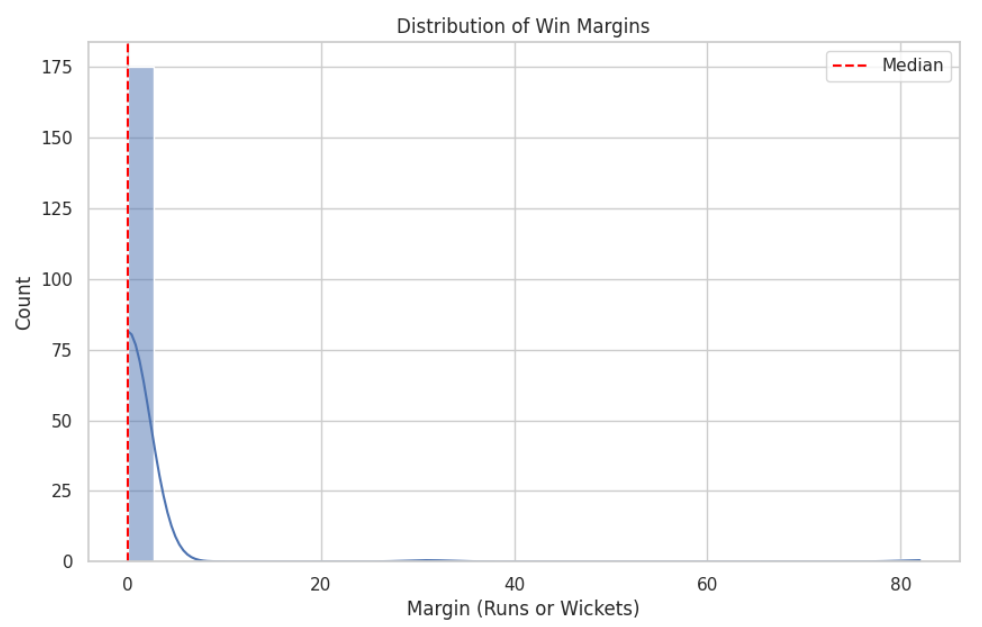


This assisted us in exploring victory margins — highest and average — for each team by format.

**3.4.2 Top and Average Margins**

We used:

* groupby() to group by team and win type.
* max() and.mean() to fetch top and average win margins, respectively.



The plot results indicated that teams like the Mumbai Indians and the Chennai Super Kings often had the biggest winning margins.

Some teams preferred to win time and again by wickets, an expression of their chasing abilities.

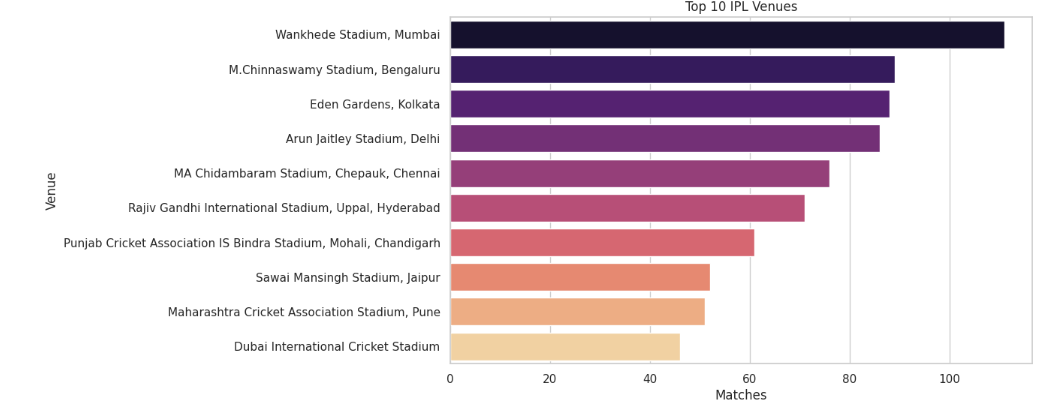
**3.5 Visualization Tools**

We used during the module:

* Matplotlib: for barebones plotting control
* Seaborn: for pretty and to-the-point statistical plots

Plot types used were:

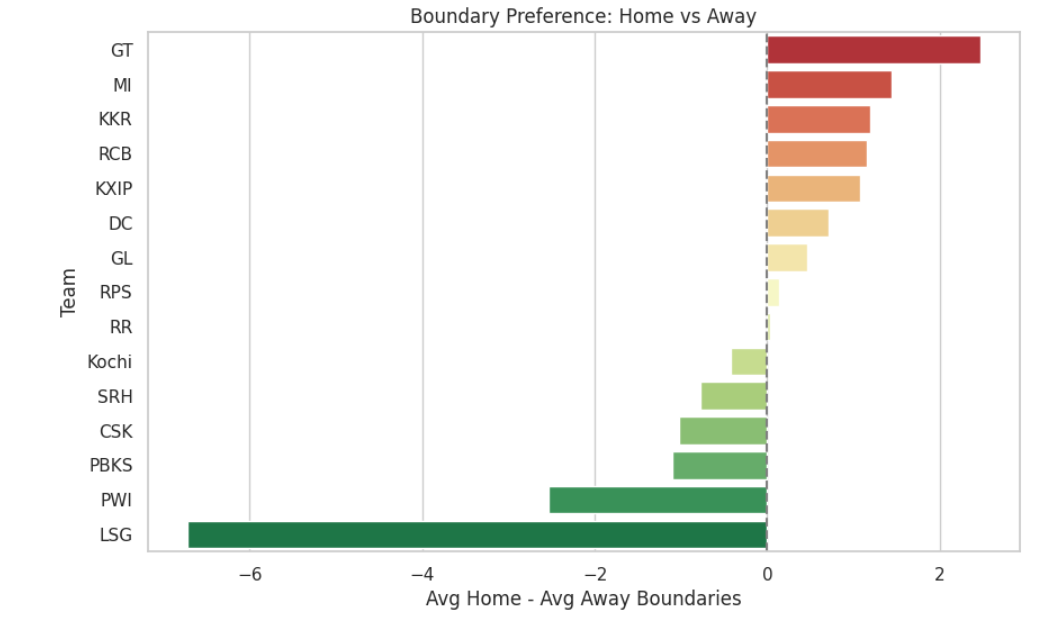
* Barplots for categorical summaries
* Histograms for distributions (e.g., match duration)
* Scatter/horizontal bar plots for numerical comparisons



These visualizations augment inferences based on Pandas operations and make the results more palatable.

**4. Summary of Key Insights**

* Runs vs Wickets: Most IPL matches are won by wickets, emphasizing the role of chasing in T20 cricket.
* Dominant Teams: Strong teams like MI and CSK always have bigger win margins.
* Toss Analysis: The teams that win the toss and decide to field win the game; there is scope for ML analysis in future modules.
* Match Duration: Most matches are one-day affairs, but a few take several days, indicating delays or complexities in scheduling.



**5. Learning Outcomes**

At the end of this module, we practiced:

* End-to-end Pandas workflows with I/O, preprocessing, transformation, aggregation, and joining.
* Real-world interpretation of structured sports data.
* Use of visualization libraries to narrate data-driven stories.
* Key skills to set us up to model with Scikit-learn and PyTorch in future modules.

**REFERENCES**

McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.

IPL T20 | Indian Premier League Official Website. (n.d.). <https://www.iplt20.com/>

Indian Premier League (IPL) - All seasons. (2024, May 13). Kaggle. <https://www.kaggle.com/datasets/rajsengo/indian-premier-league-ipl-all-seasons>

Using Matplotlib — Matplotlib 3.10.1 documentation. (n.d.). <https://matplotlib.org/stable/users/index.html>

pandas documentation — pandas 2.2.3 documentation. (n.d.). <https://pandas.pydata.org/docs/>

Seaborn: statistical data visualization — seaborn 0.13.2 documentation. (n.d.). <https://seaborn.pydata.org/>