

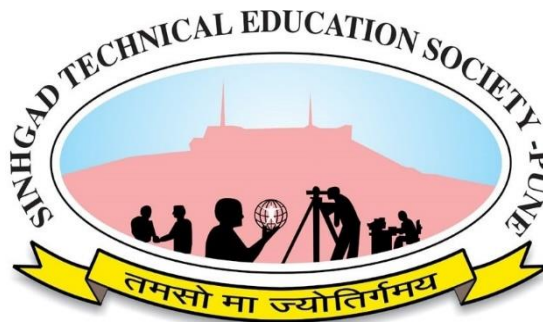
**A**  
**PROJECT REPORT**  
**ON**  
**“IDAS – THE IPL DATA ANALYSIS SYSTEM”**

**SUBMITTED BY**

**RITIK ANIL AGRAWAL.**

**PARTH NITIN SHARMA**

**SHREYAS SAMEER JOSHI**



**Sinhgad Institutes**

**GUIDED BY**

**PROF. L J. SANKPAL**

**DEPARTMENT OF COMPUTER ENGINEERING**

**STES'S**  
**SINHGAD ACADEMY OF ENGINEERING KONDHWA (BK), PUNE**  
**2020-2021**

# CERTIFICATE

This is to certify that the project report entitled

**“IDAS – The IPL Data Analysis System”**

**Submitted By**

**Ritik Anil Agrawal.**

**Parth Nitin Sharma.**

**Shreyas Sameer Joshi.**

In the fulfillment of

**B.E in Computer Engineering**

has been satisfactorily completed under

my guidance as per the requirement of

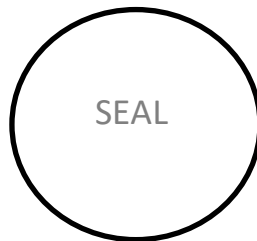
**Savitribai Phule Pune University,**

**Pune.**

**Prof. L J. Sankpal**

**(Project Guide)**

**External Examiner**



**Prof. B B. Gite**

**(H.O.D)**

**Prof. K P. Patil**

**(Principal)**

**Department of Computer Engineering**

**Sinhgad Academy of Engineering Kondhwa (Bk), Pune**

**2020 – 2021**

**Sinhgad Technical Education Society**  
**Sinhgad Academy of Engineering Kondhwa (Bk), Pune**  
**Department of Computer Engineering**

**B.E in Computer Engineering**

**Certificate of Originality**

This is to certify that I am responsible for the work submitted in this project report, the original work is my own except as specified in the reference & acknowledgment and the original work contained herein has not been undertaken or done by unspecified sources or persons.

**Ritik A. Agrawal**

**Parth N. Sharma**

**Shreyas S. Joshi**

## ACKNOWLEDGEMENT

---

It is difficult task to acknowledge all those who have been of tremendous help in this academic project work. Nevertheless, we have made an effort through this report to express our deepest gratitude to all those who have contributed their part for this project directly or indirectly.

We would like to express our sincere thanks to our principal **Prof. K P. Patil Sir** for forwarding us to do our project and offering adequate duration in completing our project.

We take upon the opportunity to express our deepest gratitude & heartily thanks to **Prof. B B. Gite Sir**, Head of Computer Engineering Department for their constructive suggestions & encouragement during our project.

With deep sense of gratitude, we extend our earnest & sincere thanks to our project guide **Prof. L J. Sankpal Ma'am**, Department of Computer Engineering for his valuable support & guidance during the development of this project & encouraging us.

I am in debt of **rest of the staff of Computer Department**, for constant inspiration in the pedagogical world of Computer Engineering Field.

**Ritik A. Agrawal**

**Parth N. Sharma**

**Shreyas S. Joshi**

## ABSTRACT

---

**“IDAS – IPL Data Analysis System”**, the design of different modules and the logic framework are aptly described in this report. In the backend we have used the SQLite and in the frontend we have used Python along with its libraries to analyse the data and plot the graphs. Also a dataset is analysed on which outputs are generated.

Our vision behind making this project is the most important. As we all know we are living in the 21<sup>st</sup> Century and the technology and the innovations going now a day are growing as fast as the yeast spreads in the breads. The Data is going to be the most important part of each and everyone's life in coming years. Data will get such a importance that it will be more precious than a human. So analysing, managing, securing the data is the most important task in coming days.

Analysing of data and representing it in a attractive manner is very interesting task. Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

Cricket is the very popular sport among many people and most popular sport in India. India's Cricket Board named BCCI hosts a tournament every year since 2008 named as IPL, which is very popular across globe.

Since 2008 IPL has a ton of data or cricketing information which people or team management have a look onto it, must be analysed and should be presented in a proper way. Hence, our project deals with the same analysis of the IPL data and generates various reports on some of the interesting IPL topics.

In this project we have taken a dataset of IPL containing data since 2008 till 2020, this dataset is in CSV format. We got this dataset from the Kaggle.com website which has thousands of dataset onto it and it is opensource hence it is very popular among beginners. After providing the dataset it is analysed using the python code and then drafted in a attractive format using the matplotlib. Some of the reports are also stored in the database. We have used SQLite database on which the reports outputs are being stored and can be easily accessed. This automation on manipulating the data has helped a lot in managing, accessing and viewing the data.

## INDEX

CHAPTER NO.	CHAPTER NAME	PAGE NO.
1.	INTRODUCTION	1
1.1	DATA ANALYSIS	1
1.2	INDIAN PREMIER LEAGUE	5
2.	SYSTEM ANALYSIS	11
2.1	PYTHON	11
2.2	R (PROGRAMMING LANGUAGE)	16
2.3	JUPYTER NOTEBOOK	19
2.4	SQLite	21
3.	MODULES	23
3.1	IMPLEMENTED MODULES	23
4.	OUTPUT	26
4.1	REPORT OF THE MODULES	26
5.	MERITS & DE-MERITS	31
5.1	ADVANTAGES	31
5.2	DIS-ADVANTAGES	31
5.3	APPLICATIONS	31
6.	CONCLUSION	32
7.	BIBLIOGRAPHY	33

## **INTRODUCTION**

---

### **1.1 Data Analysis**

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

Data mining is a particular data analysis technique that focuses on statistical modelling and knowledge discovery for predictive rather than purely descriptive purposes, while business intelligence covers data analysis that relies heavily on aggregation, focusing mainly on business information. In statistical applications, data analysis can be divided into descriptive statistics, exploratory data analysis (EDA), and confirmatory data analysis (CDA).

EDA focuses on discovering new features in the data while CDA focuses on confirming or falsifying existing hypotheses. Predictive analytics focuses on application of statistical models for predictive forecasting or classification, while text analytics applies statistical, linguistic, and structural techniques to extract and classify information from textual sources, a species of unstructured data. All of the above are varieties of data analysis.

Data integration is a precursor to data analysis, and data analysis is closely linked to data visualization and data dissemination.



### **1.1.1 The process of data analysis**

Data science process flowchart from Doing Data Science, by Schutt & O'Neil (2013) Analysis, refers to dividing a whole into its separate components for individual examination. Data analysis, is a process for obtaining raw data, and subsequently converting it into information useful for decision-making by users. Data, is collected and analysed to answer questions, test hypotheses, or disprove theories.

Statistician John Tukey, defined data analysis in 1961, as:

"Procedures for analysing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analysing data." The phases are iterative, in that feedback from later phases may result in additional work in earlier phases. The CRISP framework, used in data mining, has similar steps.

### **1.1.2 Data Requirements**

The data are necessary as inputs to the analysis, which is specified based upon the requirements of those directing the analysis or customers (who will use the finished product of the analysis). The general type of entity upon which the data will be collected is referred to as an experimental unit (e.g., a person or population of people). Specific variables regarding a population (e.g., age and income) may be specified and obtained. Data may be numerical or categorical.

### **1.1.3 Data Collection**

Data are collected from a variety of sources. The requirements may be communicated by analysts to custodians of the data; such as, Information Technology personnel within an organization. The data may also be collected from sensors in the environment, including traffic cameras, satellites, recording devices, etc. It may also be obtained through interviews, downloads from online sources, or reading documentation.

### **1.1.4 Data Processing**

The phases of the intelligence cycle used to convert raw information into actionable intelligence or knowledge are conceptually similar to the phases in data analysis.

Data, when initially obtained, must be processed or organized for analysis. For instance, these may involve placing data into rows and columns in a table format (known as structured data) for further analysis, often through the use of spreadsheet or statistical software.

### **1.1.5 Data Cleaning**

Once processed and organized, the data may be incomplete, contain duplicates, or contain errors. The need for data cleaning, will arise from problems in the way that the datum is entered and stored. Data cleaning is the process of preventing and correcting these errors. Common tasks include record matching, identifying inaccuracy of data, overall quality of existing data, deduplication, and column segmentation.

Such data problems can also be identified through a variety of analytical techniques. For example, with financial information, the totals for particular variables may be compared against separately published numbers, that are believed to be reliable.

Unusual amounts, above or below predetermined thresholds, may also be reviewed. There are several types of data cleaning, that are dependent upon the type of data in the set; this could be phone numbers, email addresses, employers, or other values. Quantitative data methods for outlier detection, can be used to get rid of data that appears to have a higher likelihood of being input incorrectly.

Textual data spell checkers, can be used to lessen the amount of mis-typed words, however, it is harder to tell if the words themselves are correct.

### 1.1.6 Exploratory Data Analysis

Once the datasets are cleaned, it can then be analysed. Analysts may apply a variety of techniques, referred to as exploratory data analysis, to begin understanding the messages contained within the obtained data.

The process of data exploration may result in additional data cleaning or additional requests for data; thus, the initialization of the iterative phases mentioned in the lead paragraph of this section. Descriptive statistics, such as, the average or median, can be generated to aid in understanding the data. Data visualization is also a technique used, in which the analyst is able to examine the data in a graphical format in order to obtain additional insights, regarding the messages within the data.

### 1.1.7 Modelling and Algorithms

Mathematical formulas or models (known as algorithms), may be applied to the data in order to identify relationships among the variables; for example, using correlation or causation. In general terms, models may be developed to evaluate a specific variable based on other variable(s) contained within the dataset, with some residual error depending on the implemented model's accuracy (e.g.,  $\text{Data} = \text{Model} + \text{Error}$ ).

Inferential statistics, includes utilizing techniques that measure the relationships between particular variables. For example, regression analysis may be used to model whether a change in advertising (independent variable X), provides an explanation for the variation in sales (dependent variable Y). In mathematical terms, Y (sales) is a function of X (advertising). It may be described as  $(Y = ax + b + \text{error})$ , where the model is designed such that (a) and minimize the error when the model predict(s) Y for a given range of value for (f).X. Analysts may also attempt to build models that are descriptive of the data, in an aim to simplify analysis and communicate results.

### **1.1.8 Data product**

A data product, is a computer application that takes data inputs and generates outputs, feeding them back into the environment. It may be based on a model or algorithm. For instance, an application that analyses data about customer purchase history, and uses the results to recommend other purchases the customer might enjoy.

### **1.1.9 Communication**

Data visualization to understand the results of a data analysis. Once the data are analysed, it may be reported in many formats to the users of the analysis to support their requirements. The users may have feedback, which results in additional analysis. As such, much of the analytical cycle is iterative.

When determining how to communicate the results, the analyst may consider implementing a variety of data visualization techniques, to help clearly and efficiently communicate the message to the audience. Data visualization uses information displays (graphics such as, tables and charts) to help communicate key messages contained in the data. Tables are a valuable tool by enabling the ability of a user to query and focus on specific numbers; while charts (e.g., bar charts or line charts), may help explain the quantitative messages contained in the data.

## **1.2 Indian Premier League**

Over the years the designers at Arduino.cc have developed a number of board designs. The first widely distributed Arduino board, the Diecimila, was released in 2007, and since its initial release the Arduino family has evolved to take advantage of the various types of Atmel AVR MCU devices. The Due, released in 2012, is the first Arduino to utilize a 32-bit ARM Cortex-M3 processor, and it breaks from the rest of the family in terms of both processing power and board pinout configuration. Other boards, like the Lily Pad and the Nano, also do not have the same pinout as the other members of the family, and are intended for a different range of

The Indian Premier League (IPL) is a professional Twenty20 cricket league in India usually contested between March and May of every year by eight teams representing eight different cities or states in India. The league was founded by the Board of Control for Cricket in India (BCCI) in 2007. The IPL has an exclusive window in ICC Future Tours Programme.

The IPL is the most-attended cricket league in the world and in 2014 ranked sixth by average attendance among all sports leagues. In 2010, the IPL became the first sporting event in the world to be broadcast live on YouTube. The brand value of the IPL in 2019 was ₹475 billion (US\$6.7 billion), according to Duff & Phelps. According to BCCI, the 2015 IPL season contributed ₹11.5 billion (US\$160 million) to the GDP of the Indian economy.

There have been thirteen seasons of the IPL tournament. The current IPL title holders are the Mumbai Indians, who won the 2020 season. The venue for the 2020 season was moved due to the COVID-19 pandemic and games were played in the United Arab Emirates.

### **1.2.1 Background**

The Indian Cricket League (ICL) was founded in 2007, with funding provided by Zee Entertainment Enterprises. The ICL was not recognised by the Board of Control for Cricket in India (BCCI) or the International Cricket Council (ICC) and the BCCI were not pleased with its committee members joining the ICL executive board. To prevent players from joining the ICL, the BCCI increased the prize money in their own domestic tournaments and also imposed lifetime bans on players joining the ICL, which was considered a rebel league by the board.

### **1.2.2 Foundation**

On 13 September 2007, the BCCI announced the launch of a franchise-based Twenty20 cricket competition called Indian Premier League whose first season was slated to start in April 2008, in a "high-profile ceremony" in New Delhi. BCCI vice-president Lalit Modi, said to be the mastermind behind the idea of the IPL,

spelled out the details of the tournament including its format, the prize money, franchise revenue system and squad composition rules. It was also revealed that the IPL would be run by a seven-man governing council composed of former India players and BCCI officials and that the top two teams of the IPL would qualify for that year's Champions League Twenty20. Modi also clarified that they had been working on the idea for two years and that the IPL was not started as a "knee-jerk reaction" to the ICL. The league's format was similar to that of the Premier League of England and the NBA in the United States.

In order to decide the owners for the new league, an auction was held on 24 January 2008 with the total base prices of the franchises costing around \$400 million.

At the end of the auction, the winning bidders were announced, as well as the cities the teams would be based in: Bangalore, Chennai, Delhi, Hyderabad, Jaipur, Kolkata, Mohali, and Mumbai. In the end, the franchises were all sold for a total of \$723.59 million. The Indian Cricket League soon folded in 2008.

### **1.2.3 Expansions and Terminations**

On 21 March 2010, it was announced that two new franchises – Pune Warriors India and Kochi Tuskers Kerala – would join the league before the fourth season in 2011. Sahara Adventure Sports Group bought the Pune franchise for \$370 million while Rendezvous Sports World bought the Kochi franchise for \$333.3 million. However, one year later, on 11 November 2011, it was announced that the Kochi Tuskers Kerala side would be terminated following the side breaching the BCCI's terms of conditions.

Then, on 14 September 2012, following the team not being able to find new owners, the BCCI announced that the 2009 champions, the Deccan Chargers, would be terminated. The next month, on 25 October, an auction was held to see who would be the owner of the replacement franchise, with Sun TV Network winning the bid for the Hyderabad franchise. The team would be named Sunrisers Hyderabad.

Pune Warriors India withdrew from the IPL on 21 May 2013 over financial differences with the BCCI. The franchise was officially terminated by the BCCI, on 26 October 2013, on account of the franchise failing to provide the necessary bank guarantee.

On 14 June 2015, it was announced that two-time champions, Chennai Super Kings, and the inaugural season champions, Rajasthan Royals, would be suspended for two seasons following their role in a match-fixing and betting scandal. Then, on 8 December 2015, following an auction, it was revealed that Pune and Rajkot would replace Chennai and Rajasthan for two seasons. The two teams were the Rising Pune Supergiant and the Gujarat Lions.

#### **1.2.4 Tournament Format**

Currently, with eight teams, each team plays each other twice in a home-and-away round-robin format in the league phase. At the conclusion of the league stage, the top four teams will qualify for the playoffs. The top two teams from the league phase will play against each other in the first Qualifying match, with the winner going straight to the IPL final and the loser getting another chance to qualify for the IPL final by playing the second Qualifying match. Meanwhile, the third and fourth place teams from league phase play against each other in an eliminator match and the winner from that match will play the loser from the first Qualifying match. The winner of the second Qualifying match will move onto the final to play the winner of the first Qualifying match in the IPL Final match, where the winner will be crowned the Indian Premier League champions.

#### **1.2.5 Player Acquisition, Squad Composition and Salaries**

A team can acquire players through any of the three ways: the annual player auction, trading players with other teams during the trading windows, and signing replacements for unavailable players. Players sign up for the auction and also set their base price, and are bought by the franchise that bids the highest for them. Unsold players at the auction are eligible to be signed up as replacement signings. In the trading windows, a player can only be traded with his consent, with the franchise

paying the difference if any between the old and new contracts. If the new contract is worth more than the older one, the difference is shared between the player and the franchise selling the player. There are generally three trading windows—two before the auction and one after the auction but before the start of the tournament. Players cannot be traded outside the trading windows or during the tournament, whereas replacements can be signed before or during the tournament.

Some of the team composition rules (as of 2020 season) are as follows:

The squad strength must be between 18 and 25 players, with a maximum of 8 overseas players.

Salary cap of the entire squad must not exceed ₹850 million (US\$12 million).

Under-19 players cannot be picked unless they have previously played first-class or List A cricket.

A team can play a maximum of 4 overseas players in their playing eleven.

The term of a player contract is one year, with the franchise having the option to extend the contract by one or two years. Since the 2014 season, the player contracts are denominated in the Indian rupee, before which the contracts were in U.S. dollars. Overseas players can be remunerated in the currency of the player's choice at the exchange rate on either the contract due date or the actual date of payment. Prior to the 2014 season, Indian domestic players were not included in the player auction pool and could be signed up by the franchises at a discrete amount while a fixed sum of ₹1 million (US\$14,000) to ₹3 million (US\$42,000) would get deducted per signing from the franchise's salary purse. This received significant opposition from franchise owners who complained that richer franchises were "luring players with under-the-table deals" following which the IPL decided to include domestic players in the player auction.

According to a 2015 survey by Sporting Intelligence and ESPN The Magazine, the average IPL salary when pro-rated is US\$4.33 million per year, the second highest



among all sports leagues in the world. Since the players in the IPL are only contracted for the duration of the tournament (less than two months), the weekly IPL salaries are extrapolated pro rata to obtain an average annual salary, unlike other sports leagues in which players are contracted by a single team for the entire year.

### **1.2.6 Match Rules**

IPL games utilise television timeouts and hence there is no time limit in which teams must complete their innings. However, a penalty may be imposed if the umpires find teams misusing this privilege. Each team is given a two-and-a-half-minute "strategic timeout" during each innings; one must be taken by the bowling team between the ends of the 6th and 9th overs, and one by the batting team between the ends of the 13th and 16th overs.

Since the 2018 season, the Umpire Decision Review System is being used in all IPL matches, allowing each team one chance to review an on-field umpire's decision per innings.

### **1.2.7 Prize Money**

The 2019 season of the IPL offered a total prize money of ₹500 million (US\$7.0 million), with the winning team netting ₹200 million (US\$2.8 million). The first and second runners up received ₹125 million (US\$1.8 million) and ₹87.5 million (US\$1.2 million), respectively, with the fourth placed team also winning ₹87.5 million (US\$1.2 million). The other teams are not awarded any prize money. The IPL rules mandate that half of the prize money must be distributed among the players.

**SYSTEM ANALYSIS**

---

**2.1. PYTHON**

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was created in the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python 2.0, released in 2000, introduced new features, such as list comprehensions, and a garbage collection system with reference counting, and was discontinued with version 2.7 in 2020. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. With Python 2's end-of-life, only Python 3.6.x and later are supported, with older versions still supporting e.g. Windows 7 (and old installers not restricted to 64-bit Windows).

Python interpreters are supported for mainstream operating systems and available for a few more (and in the past supported many more). A global community of programmers develops and maintains Python, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development. It currently ties with Java as the second most popular programming language in the world.

### **2.1.1. History**

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator for Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project. Guido van Rossum has since then withdrawn his nomination for the 2020 Steering council.

Python 2.0 was released on 16 October 2000 with many major new features, including a cycle-detecting garbage collector and support for Unicode. Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely backward-compatible. Many of its major features were backported to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

### **2.1.2. Design philosophy and features**

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including

by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution. Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions; list comprehensions, dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Readability counts.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and

preferably only one—obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

A common neologism in the Python community is *pythonic*, which can have a wide range of meanings related to program style. To say that code is *pythonic* is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called *unpythonic*.

### **2.1.3. Uses**

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of February 2020, it is the third most popular language (behind Java, and C). It was selected Programming Language of the Year in 2007, 2010, and 2018. An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving

string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".

Large organizations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify and some smaller entities like ILM and ITA. The social news networking site Reddit is written entirely in Python.

Python can serve as a scripting language for web applications, e.g., via `mod_wsgi` for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used as a data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.

Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a mathematical software with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus. OpenCV has python bindings with a rich set of features for computer vision and image processing.

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.

Python is commonly used in artificial intelligence projects and machine learning projects with the help of libraries like TensorFlow, Keras, Pytorch and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

## **2.2. R**

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, data mining surveys, and studies of scholarly literature databases show substantial increases in popularity; as of September 2020, R ranks 9th in the TIOBE index, a measure of popularity of programming languages.

A GNU package, the official R software environment is written primarily in C, Fortran, and R itself (thus, it is partially self-hosting) and is freely available under the GNU General Public License. Pre-compiled executables are provided for various operating systems. Although R has a command line interface, there are several third-party graphical user interfaces, such as RStudio, an integrated development environment, and Jupyter, a notebook interface.

### **2.2.1. History**

R is an implementation of the S programming language combined with lexical scoping semantics, inspired by Scheme. S was created by John Chambers in 1976 while at Bell Labs. A commercial version of S was offered as S-PLUS starting in 1988.

Much of the code written for S-PLUS runs unaltered in R.

In 1991 Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, began an alternative implementation of the basic S language, completely

independent of S-PLUS. They publicized this project starting in 1993. In 1995 Martin Maechler convinced Ihaka and Gentleman to make R free and open-source software under the GNU General Public License. The R Development Core Team was created to manage the further development of R. John Chambers became a member at least as of August 2018. R is named partly after the first names of the first two R authors and partly as a play on the name of S.

The first official release came in 1995. The Comprehensive R Archive Network (CRAN) was officially announced 23 April 1997 with 3 mirrors and 12 contributed packages. The first official "stable beta" version (v1.0) was released 29 February 2000.

### **2.2.2. Statistical features**

R and its libraries implement a wide variety of statistical and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and others. R is easily extensible through functions and extensions, and the R community is noted for its active contributions in terms of packages. Many of R's standard functions are written in R itself,[citation needed] which makes it easy for users to follow the algorithmic choices made. For computationally intensive tasks, C, C++, and Fortran code can be linked and called at run time. Advanced users can write C, C++, Java, .NET or Python code to manipulate R objects directly. R is highly extensible through the use of user-submitted packages for specific functions or specific areas of study. Due to its S heritage, R has stronger object-oriented programming facilities than most statistical computing languages.[citation needed] Extending R is also eased by its lexical scoping rules.

Another strength of R is static graphics, which can produce publication-quality graphs, including mathematical symbols. Dynamic and interactive graphics are available through additional packages.

R has Rd, its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both online in a number of formats and in hard copy.



### 2.2.3. Programming features

R is an interpreted language; users typically access it through a command-line interpreter. If a user types `2+2` at the R command prompt and presses enter, the computer replies with 4, as shown below:

```
> 2 + 2
```

```
[1] 4
```

This calculation is interpreted as the sum of two single-element vectors, resulting in a single-element vector. The prefix indicates that the list of elements following it on the same line starts with the first element of the vector (a feature that is useful when the output extends over multiple lines).

Like other similar languages such as APL and MATLAB, R supports matrix arithmetic. R's data structures include vectors, matrices, arrays, data frames (similar to tables in a relational database) and lists. Arrays are stored in column-major order. R's extensible object system includes objects for (among others): regression models, time-series and geo-spatial coordinates. The scalar data type was never a data structure of R. Instead, a scalar is represented as a vector with length one.

Many features of R derive from Scheme. R uses S-expressions to represent both data and code.[citation needed] Functions are first-class and can be manipulated in the same way as data objects, facilitating meta-programming, and allow multiple dispatch. Variables in R are lexically scoped and dynamically typed. Function arguments are passed by value, and are lazy—that is to say, they are only evaluated when they are used, not when the function is called.

R supports procedural programming with functions and, for some functions, object-oriented programming with generic functions. A generic function acts differently depending on the classes of arguments passed to it. In other words, the generic function dispatches the function (method) specific to that class of object. For example, R has a generic print function that can print almost every class of object in R with a simple `print(object name)` syntax.

Although used mainly by statisticians and other practitioners requiring an environment for statistical computation and software development, R can also operate as a general matrix calculation toolbox – with performance benchmarks comparable to GNU Octave or MATLAB.

### 2.3. JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension. A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, Restructured-Text, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell. To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

Jupyter Notebook provides a browser-based REPL built upon a number of popular open-source libraries:

IPython

ØMQ

Tornado (web server)

jQuery

Bootstrap (front-end framework)

## MathJax

languages. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 release (October 2014), there are currently 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell.

The Notebook interface was added to IPython in the 0.12 release (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 – Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. According to The Atlantic, Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

### **2.3.1. Jupyter kernels**

A Jupyter kernel is a program responsible for handling various types of requests (code execution, code completions, inspection), and providing a reply. Kernels talk to the other components of Jupyter using ZeroMQ, and thus can be on the same or remote machines. Unlike many other Notebook-like interfaces, in Jupyter, kernels are not aware that they are attached to a specific document, and can be connected to many clients at once. Usually kernels allow execution of only a single language, but there are a couple of exceptions.

### **2.3.2. JupyterHub**

JupyterHub is a multi-user server for Jupyter Notebooks. It is designed to support many users by spawning, managing, and proxying many singular Jupyter Notebook servers. [citation needed] While JupyterHub requires managing servers, third-party services like JupyterLab provide an alternative to JupyterHub by hosting and managing multi-user Jupyter notebooks in the cloud.

### **2.3.3. JupyterLab**

JupyterLab is a newer user interface for Project Jupyter. It offers the building blocks of the classic Jupyter Notebook (notebook, terminal, text editor, file browser, rich outputs, etc.) in a flexible user interface. The first stable release was announced on February 20, 2018.

### **2.3.4. Jupyter {Book}**

Jupyter Book is an open source project for building books and documents from computational material. It allows the user to construct the content in a mixture of Markdown, an extended version of Markdown called MyST, Maths & Equations using MathJax, Jupyter Notebooks, reStructuredText, the output of running Jupyter Notebooks at build time. Multiple output formats can be produced (currently single files, multipage HTML web pages and PDF files).

### **2.3.5. Nbgrader**

nbgrader is a tool for creating and grading (marking) assignments in Jupyter notebooks. It allows the instructor to create assignments that include coding exercises in python or any other supported kernel and text responses. The submitted assignments can be automatically marked, manually scored or a mixture of both.

## **2.4. SQLite**

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, generally following PostgreSQL syntax. However, SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity. This means that one can, for example, insert a string into a column defined as an integer. SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions and will store the data as-is if such a conversion is not possible.

SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

### 2.4.1 Features

SQLite implements most of the SQL-92 standard for SQL, but lacks some features. For example, it only partially provides triggers and cannot write to views (however, it provides INSTEAD OF triggers that provide this functionality). While it provides complex queries, it still has limited ALTER TABLE function, as it cannot modify or delete columns. SQLite uses an unusual type system for a SQL-compatible DBMS: instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is dynamically typed. Moreover, it is weakly typed in some of the same ways that Perl is: one can insert a string into an integer column (although SQLite will try to convert the string to an integer first, if the column's preferred type is integer). This adds flexibility to columns, especially when bound to a dynamically typed scripting language. However, the technique is not portable to other SQL products. A common criticism is that SQLite's type system lacks the data integrity mechanism provided by statically typed columns in other products. The SQLite web site describes a "strict affinity" mode, but this feature has not yet been added. However, it can be implemented with constraints like CHECK (typeof(x)='integer').

Tables normally include a hidden rowid index column, which gives faster access. If a database includes an Integer Primary Key column, SQLite will typically optimize it by treating it as an alias for rowid, causing the contents to be stored as a strictly typed 64-bit signed integer and changing its behaviour to be somewhat like an auto-incrementing column. Future[when?] versions of SQLite may include a command to introspect whether a column has behaviour like that of rowid to differentiate these columns from weakly typed, non-autoincrementing Integer Primary Keys. [failed verification] SQLite with full Unicode function is optional. Several computer processes or threads may access the same database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with an error code (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when dealing with temporary tables. This restriction is relaxed in version 3.7 when write-ahead logging (WAL) is turned on, enabling concurrent reads and writes.

### **3.1 IMPLEMENTED MODULES**

1. Most used Venue:

In this report we are printing which stadiums have been used the most for playing the IPL matches. By this report people can understand which venue is most preferred by management and also that particular venue which is the hometown of the particular team has hosted the greatest number of matches.

2. Win percentage of the team which is batting first or second on a particular venue:

In this report we are printing that which team has won the most matches while batting first or by bowling first. This report can be useful for the team management to decide whether to bowl first or bat first on that particular venue.

3. Team to win the greatest number of Matches:

In this report we are showing which IPL team has won the greatest number of matches throughout IPL seasons. This record can be used to analyse the win percentage of the teams.

4. Batsman performance over years:

In this report the performance of the particular player throughout the IPL seasons is being printed. This is very useful report for all to keep records of the player and to draft a particular player's career.

5. Batsman performance in a particular match:

In this report the performance of the particular player in a particular match has been printed. This report is helpful to track the player performance in a particular match, to keep record of that batsman's performance and easy analysis for the team management.

6. Player to win the greatest number of Man of the Match Award:

In this report which player has bagged the greatest number of the Man of the Match Award has been printed. This is very exciting for the cricket fans to know that whom among their favourite players have bagged the greatest number of Man of the Match award.

7. Favourite Umpire:

In this report the most favourite umpire among all has been printed. The umpire who is most famous among all the supporters and the management on the basis of their popularity and their whole career are drafted.

8. Batsman's overall Performance:

In this report the particular batsman's statistics of all IPL seasons have been printed. This is very useful report for all to keep records of the player and to draft a particular player's career. Also get his strike rate and other technical details.

9. Bowler's Overall Performance:

In this report the particular bowler's statistics of all IPL seasons have been printed. This is very useful report for all to keep records of the player and to draft a particular player's career. Also get his strike rate and other technical details.

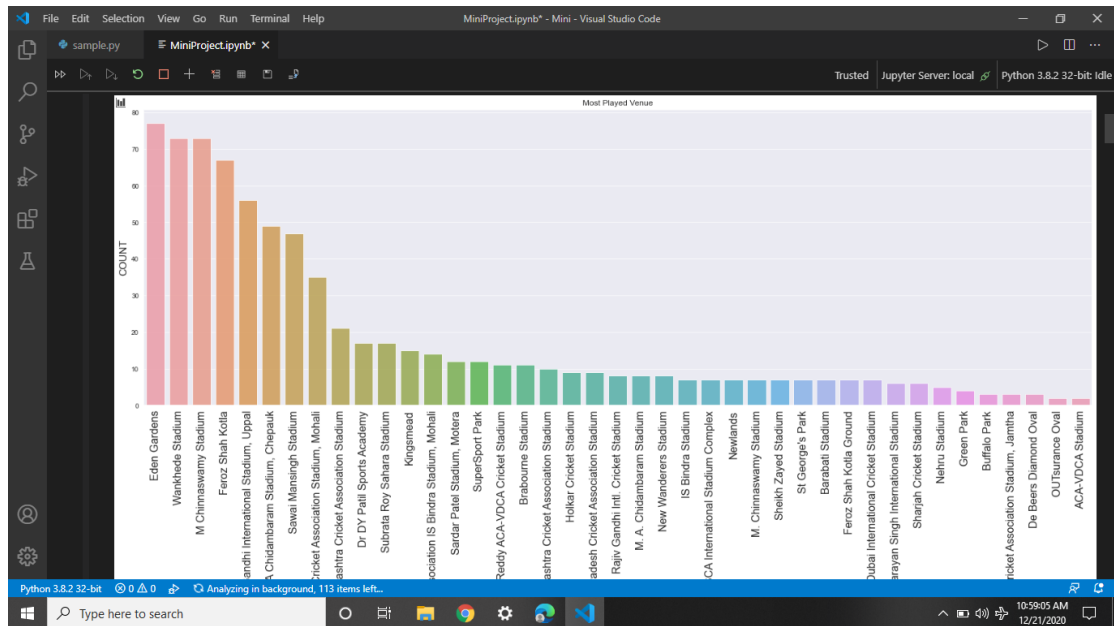
#### 10. Bowler vs Batsman Statistic:

In this report the stats of the particular bowler and batsman have been printed. This shows the rivalry between those two players, also it makes the game more interesting to watch and ofcourse to watch those two players when they are facing each other.

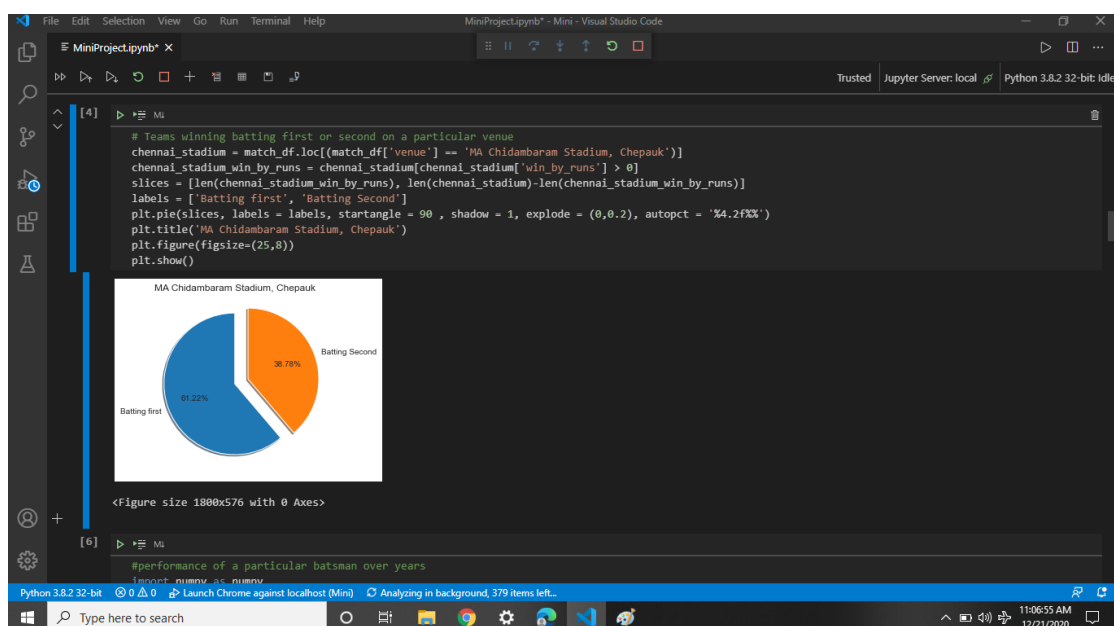


## 4.1 REPORT OF THE MODULES

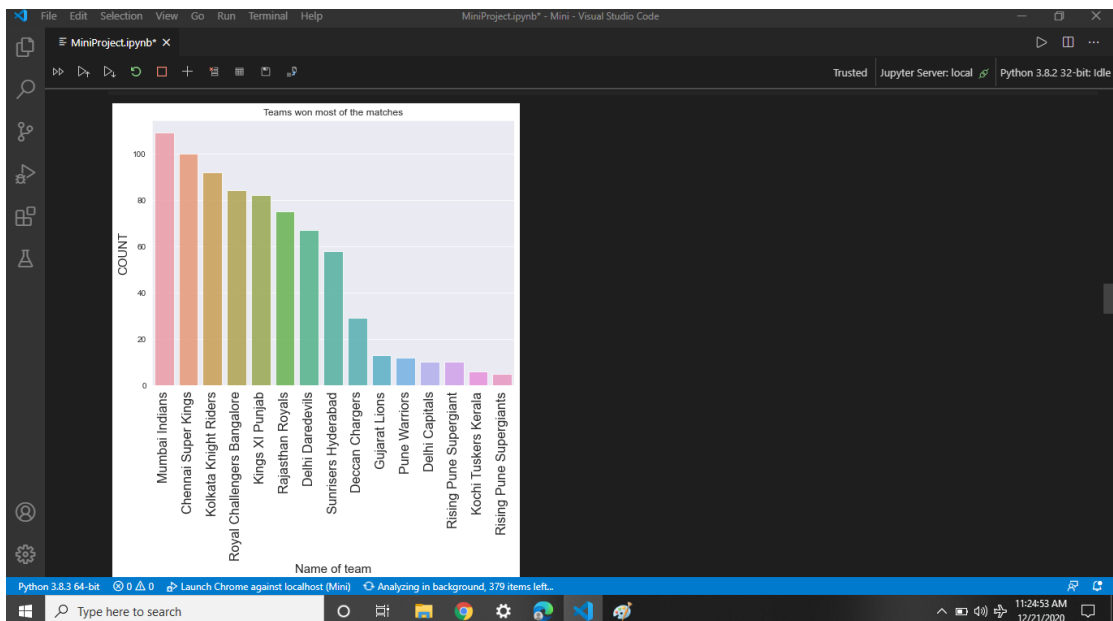
### 1. Most used Venue:



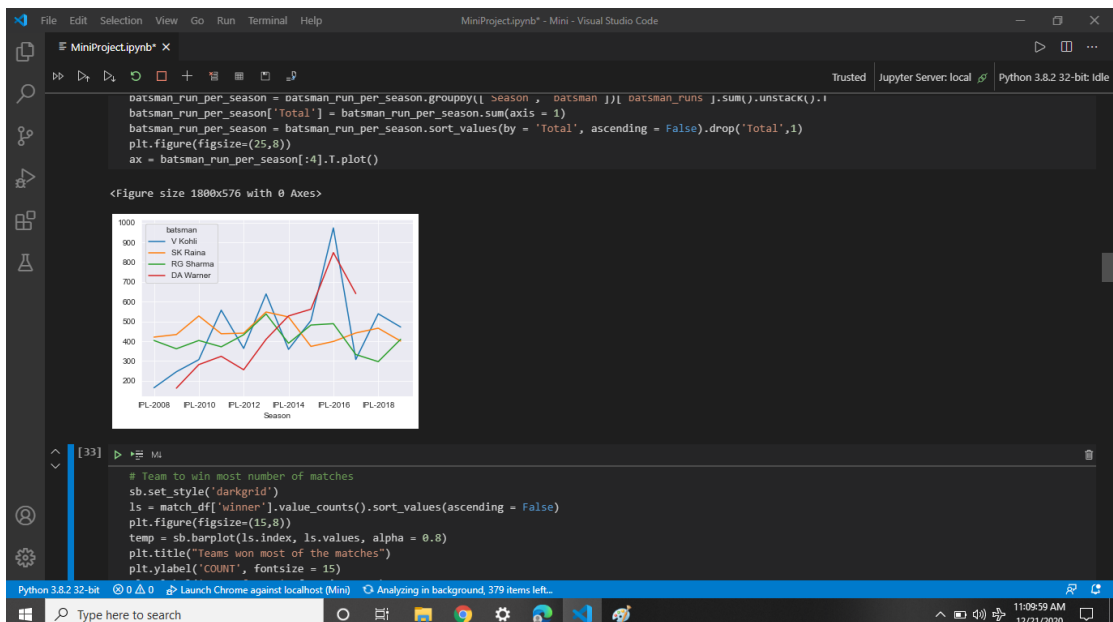
### 2. Win percentage of the team which is batting first or second on a particular venue:



## 3. Team to win the greatest number of Matches:



## 4. Batsman performance over years:



## 5. Batsman performance in a particular match:

```

ones = x['batsman_runs'] == 1
print("Number of singles taken:",ones['batsman_runs'].count())
one = int(ones['batsman_runs'].count())

totalruns = int(total_runs[1])
# creating an connection
conn = sqlite3.connect("ipl.db") # db - database

# Cursor object
cursor = conn.cursor()

# code to create a database table
# executing the above SQL code
#cursor.execute(create_table_sql)

# inserting data into the students table
insert_student_one_sql = """INSERT into batsman (id,name,runs,six,four,three,two,one) values (?,?,,?,,?,,?)""";
cursor.execute(insert_student_one_sql , (1,name,totalruns,six,four,three,two,one))

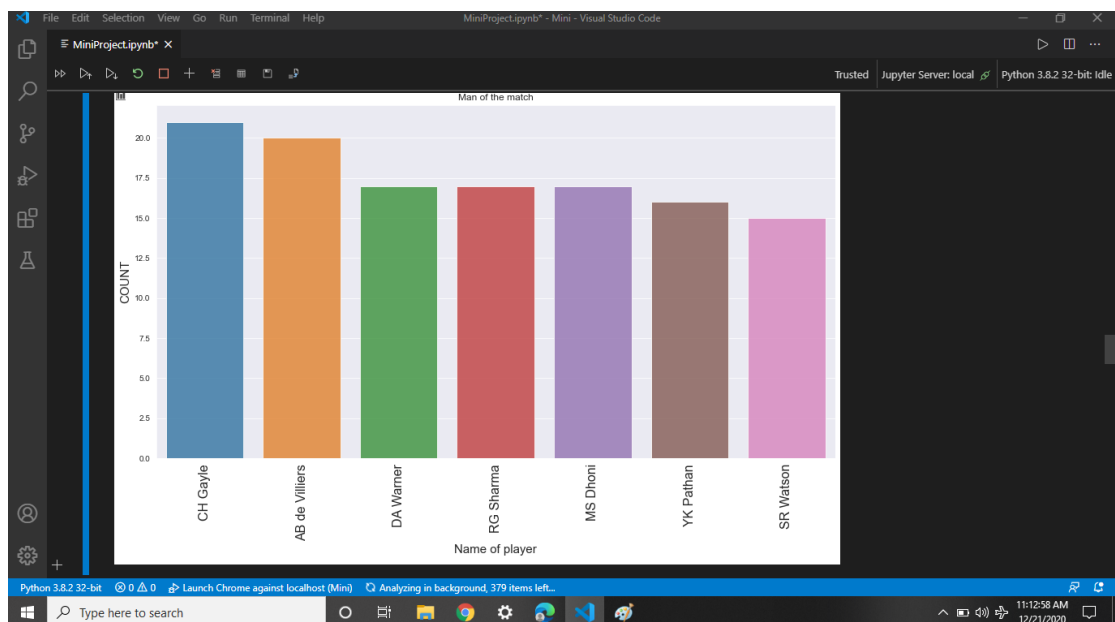
# saving the changes using commit method of connection
conn.commit()

# closing the connection
conn.close()

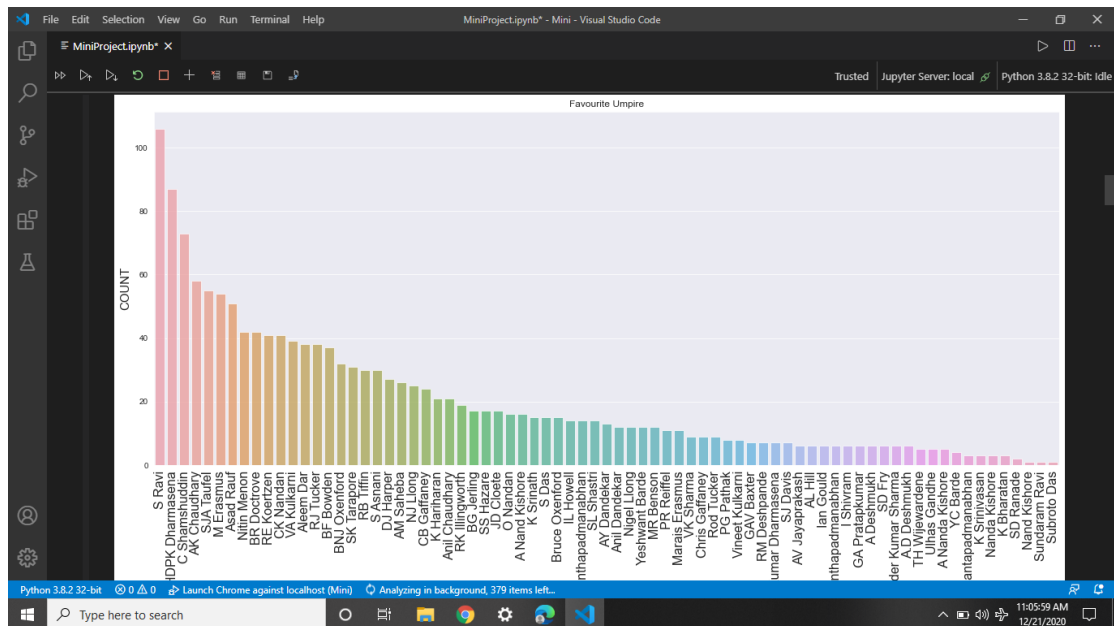
Total runs scored: 62 by 'Yuvraj Singh'
Number of six hit: 3
Number of fours hit: 7
Number of three's taken: 0
Number of doubles taken: 3
Number of singles taken: 10
    
```

Python 3.8.2 32-bit | Launch Chrome against localhost (Mini) | Analyzing in background, 379 items left...

## 6. Player to win the greatest number of Man of the Match Award:



## 7. Favourite Umpire:



## 8. Batsman's overall Performance:

```
# creating an connection
conn = sqlite3.connect("ipl.db") # db - database
# cursor object
cursor = conn.cursor()
#cursor.execute(create_table_sql)
#create_table_sql = "CREATE TABLE Overall_batsman (name VARCHAR(30),runs INTEGER,six INTEGER,four INTEGER,three INTEGER,two INTEGER,one INTEGER);
#cursor.execute(create_table_sql)

# inserting data into the students table

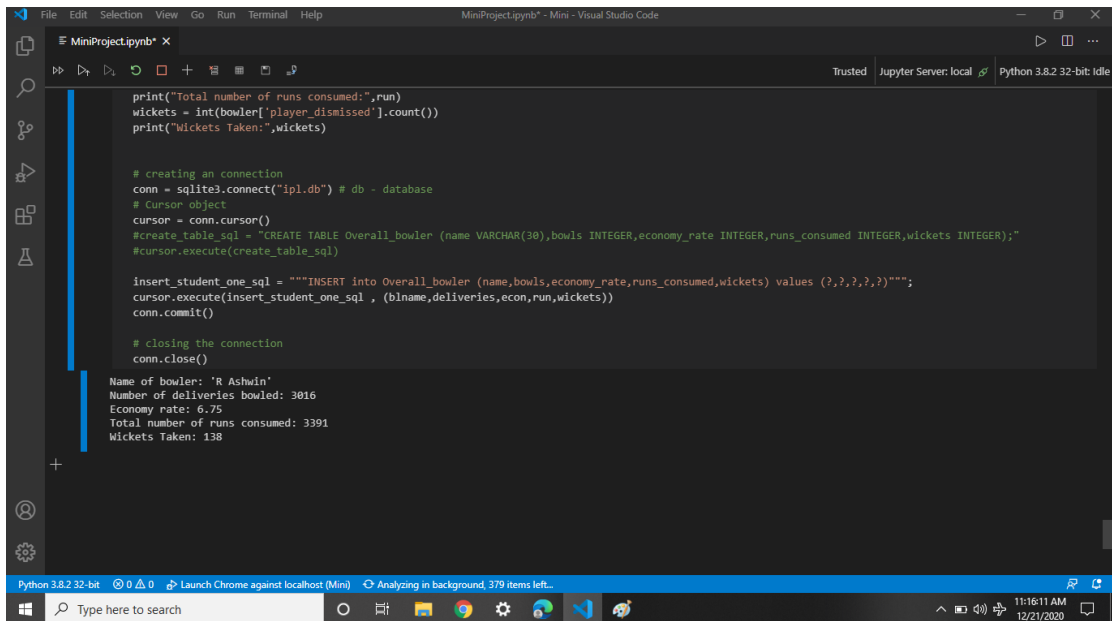
insert_student_one_sql = """INSERT into Overall_batsman (name,runs,six,four,three,two,one) values (?, ?, ?, ?, ?, ?, ?)""";
cursor.execute(insert_student_one_sql , (name,totalruns,six,four,three,two,one))

# saving the changes using commit method of connection
conn.commit()

# closing the connection
conn.close()

Total runs scored by 'RG Sharma' : 4914
Number of balls faced: 3816
Strike rate:128.77
Number of six hit: 194
Number of fours hit: 431
Number of three's taken: 5
Number of doubles taken: 205
Number of singles taken: 1589
```

### 9. Bowler's Overall Performance:



```
print("Total number of runs consumed:",run)
wickets = int(bowler['player_dismissed'].count())
print("Wickets Taken:",wickets)

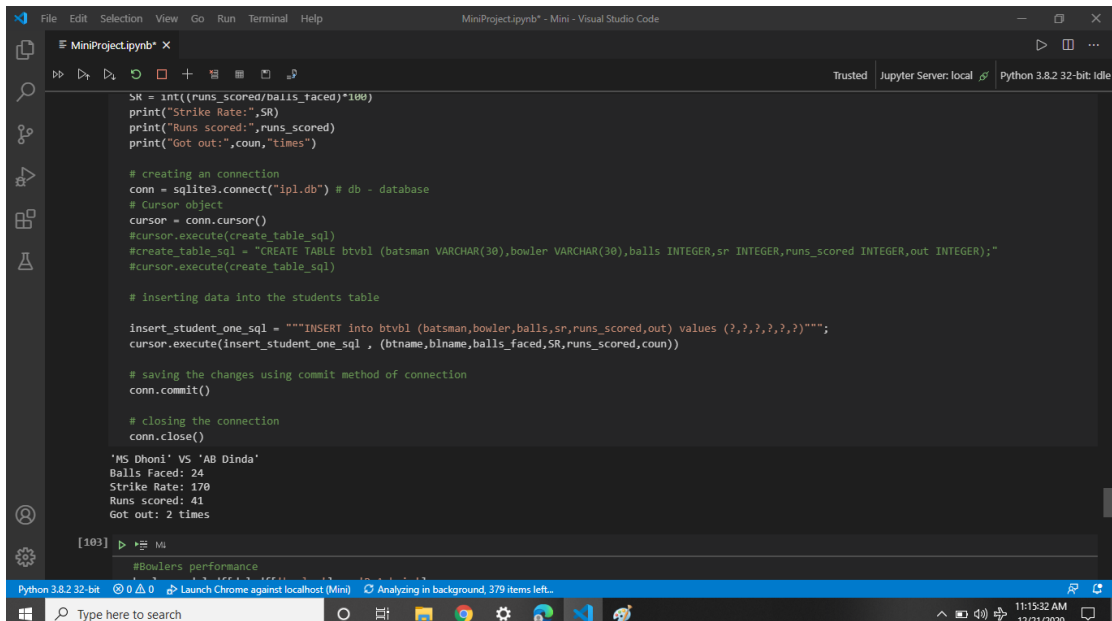
# creating an connection
conn = sqlite3.connect("ipl.db") # db - database
# Cursor object
cursor = conn.cursor()
#create_table_sql = "CREATE TABLE Overall_bowler (name VARCHAR(30),bowls INTEGER,economy_rate INTEGER,runs_consumed INTEGER,wickets INTEGER);"
#cursor.execute(create_table_sql)

Insert_student_one_sql = """INSERT into Overall_bowler (name,bowls,economy_rate,runs_consumed,wickets) values (?,?,,?)""";
cursor.execute(insert_student_one_sql , (bname,delliveries,econ,run,wickets))
conn.commit()

# closing the connection
conn.close()

Name of bowler: 'R Ashwin'
Number of deliveries bowled: 3016
Economy rate: 6.75
Total number of runs consumed: 3391
Wickets Taken: 138
```

### 10. Bowler vs Batsman Statistic:



```
SR = int((runs_scored/balls_faced)*100)
print("Strike Rate:",SR)
print("Runs scored:",runs_scored)
print("Got out:",count,"times")

# creating an connection
conn = sqlite3.connect("ipl.db") # db - database
# Cursor object
cursor = conn.cursor()
#cursor.execute(create_table_sql)
#create_table_sql = "CREATE TABLE btvbl (batsman VARCHAR(30),bowler VARCHAR(30),balls INTEGER,sr INTEGER,runs_scored INTEGER,out INTEGER);"
#cursor.execute(create_table_sql)

# inserting data into the students table

Insert_student_one_sql = """INSERT into btvbl (batsman,bowler,balls,sr,runs_scored,out) values (?,?,,?,?)""";
cursor.execute(insert_student_one_sql , (btname,bname,balls_faced,SR,runs_scored,count))

# saving the changes using commit method of connection
conn.commit()

# closing the connection
conn.close()

'MS Dhoni' VS 'AB Dinda'
Balls Faced: 24
Strike Rate: 170
Runs scored: 41
Got out: 2 times
```

## **MERITS & DE-MERITS**

---

### **7.1. Advantages**

1. Gives graphical representation of thousands of records in a sorted manner.
2. Easy to maintain records and access them.
3. Managing and working on huge data became easy.
4. Visualization in cricket data makes it more interesting to watch.
5. Speedy access to the records, made the team management and captain to make on field decision easily.

### **7.2. Dis-advantages**

1. Everytime you have to write the code for the reports.

### **7.3. Applications**

1. Can be used for Fantasy Cricket Games.
2. Can be used for Auction analysis.
3. Can be used to study the statistics before the match.
4. Can be used to show stats immediately in live matches.

## **Chapter: - 06**

### **CONCLUSION**

---

As the concluding chapter has reached, its time to conclude all things we learn, we performed and the main our vision is achieved. The aim behind making the project has been achieved.

During this whole journey we learn many new things as this whole journey has been very inspiring. We have been through the experience of handling and manipulating the data and many more things.

By this project we have understood and gone through many of the operations on the data like data analysis, data visualization, data cleaning.

The IDAS has been a successful project. Hence, we conclude as thanking each and every person who has been involved in this journey without their contribution this success was not possible.

## **BIBLIOGRAPHY**

---

During development of IDAS, we referred following websites and books, which give us some important guidelines for designing and documenting a project.

- 1) [www.kaggle.com](http://www.kaggle.com)
- 2) [www.wikipedia.com](http://www.wikipedia.com)



1. INTRODUCTION -----	1
1.1 Data Analysis-----	1
1.2 Indian Premier League -----	5
2. SYSTEM ANALYSIS-----	11
2.1 Python-----	11
2.2 R (Programming Language) -----	16
2.3 Jupyter Notebook-----	19
2.4 SQLite -----	21
3. MODULES-----	23
3.1 Implemented Modules -----	23
4. OUTPUT -----	26
4.1 Report of The Modules -----	26
5. MERITS & DE-MERITS -----	31
7.1 Advantages-----	31
7.2 Dis-Advantages -----	31
7.3 Applications -----	31
6. CONCLUSION-----	32
7. BIBLIOGRAPHY-----	33