R-DATA SCIENCE PROJECT

**Project Title:** Uber Data Analysis

**Student Information:**

* Student Name: 1) Gautam Sukhani (22070521059), Sec-A

2) Arushi Shivhare (22070521062), Sec-A

3) Tanay Yaduka (22070521086), Sec-C

* **Course Name:** Data Science Specialization

**Submitted to:** Dr. Gagandeep Kaur

**Date of Report Submission:** 21/11/2024

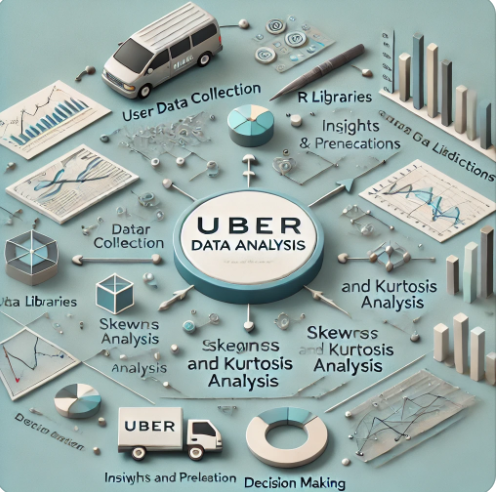
**Table Of Contents:**

| **Sr No.** | **Program** | **Page No** |
| --- | --- | --- |
| 1. | Introduction | 2 |
| 2. | Uber Dataset Overview | 3, 4 |
| 3. | Challenges in E-commerce Security | 4, 5 |
| 4. | Network Security Solutions | 5-8 |
| 4.1 | Implementation Process | 6 |
| 4.2 | Results and Benefits | 7 |
| 4.3 | Lessons Learned | 8 |
| 4.4 | Conclusion | 8 |
| 4.5 | References | 9 |

# Introduction

In the present era of big data, it is essential to understand the data models to make the right decisions for organizations. Many times in presentation of the data it is used to identify other unknown relations in the series.

In this project, we explore the world of Uber data analysis using software that is in high demand for data analysis R and its extended usage of graphic design ggplot2. First and foremost, our main goal is to extract knowledge from the Uber users and build sound recommendations about the clients.



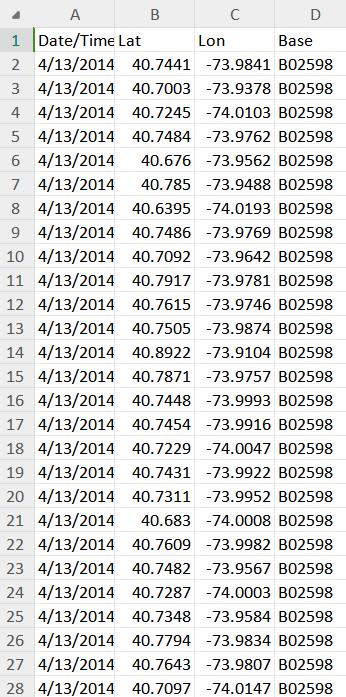
Using knowledge about past processes, it is our intention to explore determinants defining choice of ride-hailing services, including the time of day, geographic locations, and the state of the weather. These ideas can be useful to Uber to run a much more efficient service while also providing better quality to the customers.

Furthermore, we will check skewness and kurtosis in order to determine distribution of dataset and possibly existing outliers. It will assist in determining what kind of standard deviation is in the data so that one can investigate any irregularity observed in the data earned.

It will be our goal in this paper to provide a richer understanding of Uber customers, markets, and possible business models through a more detailed analysis of the information available. This information can therefore be very useful when it comes to fine tuning operations, enhancing or delivering customer satisfaction and make strategic decisions that can foster growth.

# A dataset applied in this project is an extensive record of pickups in New York City using Uber. It covers a relatively long period, and therefore it contains rich data necessary for studying the movement and dynamics of rides.

# 2. Uber Dataset Overview



**Key Data Points:**

* **Timestamp:** Time and date of a particular Uber pickup.
* **Latitude and Longitude:** Geographical location of participant’s pick up location expressed in geographical coordinates.

**Data Analysis Focus:**

This paper’s focus will be the temporal distribution of Uber pickups in New York City. We will explore:

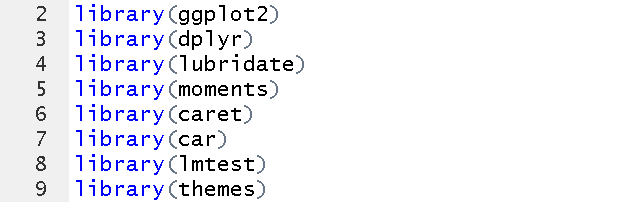
* **Daily Patterns:** How the demand varies depending on the day of the week.
* **Hourly Patterns:** The late-night use of Uber or the rush hours use of Uber.
* **Seasonal Trends:** Differences in demands of the type of service throughout the year.
* **Geographic Variations:** Variation in pickup rhythms between the districts and between the boroughs.

Using these patterns, Uber is able to observe riders’ behaviors and find ways to improve service delivery while making informed decisions.

# 3. Steps

### 3.1 IMPORTING THE ESSENTIAL PACKAGES

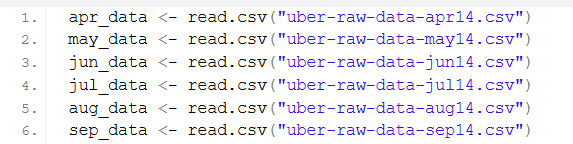
To begin our Uber data analysis project, we will import the required R packages:



* **ggplot2:** It is used especially for creating more visually appealing data manipulation.
* **ggthemes:** To provide extra themes and scales in order to improve the plot shapes.
* **lubridate:** Where all irregularity in handling and manipulating date-time data is catered for.
* **dplyr:** For data manipulation activities such as; sorting, categorizing, and aggregating data.
* **scales:** For defining content and properties of titles and legends in graphics.
* **car:** In the context of regression diagnostics, ANOVA, and visualization instruments
* **caret:** For examples of regression and statistical modeling.
* **moments:** Skewness, kurtosis and moments beyond the fourth order
* **lmtest:** In the case of linear regression models, testing of autocorrelation.

### 3.2 READING AND COMBINING DATA

There are different CSV files with data about Uber for April – September 2014, so we will load all files into data frames with corresponding names: df1, df2, df3, df4, df5, df6 and transform them into one data frame called data\_2014.



**Data Cleaning and Preparation**

1. **Format Date-Time:** A POSIXct format for the time based analysis should be undertaken and the Date.Time column should undergo this format.
2. **Create Time Factors:** Teach periodicity as factor variables so as to easily work with components such as day, month, year and hour, weekday.

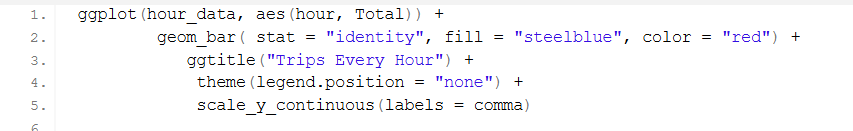
### 3.3 Visualizing Hourly Trip Patterns

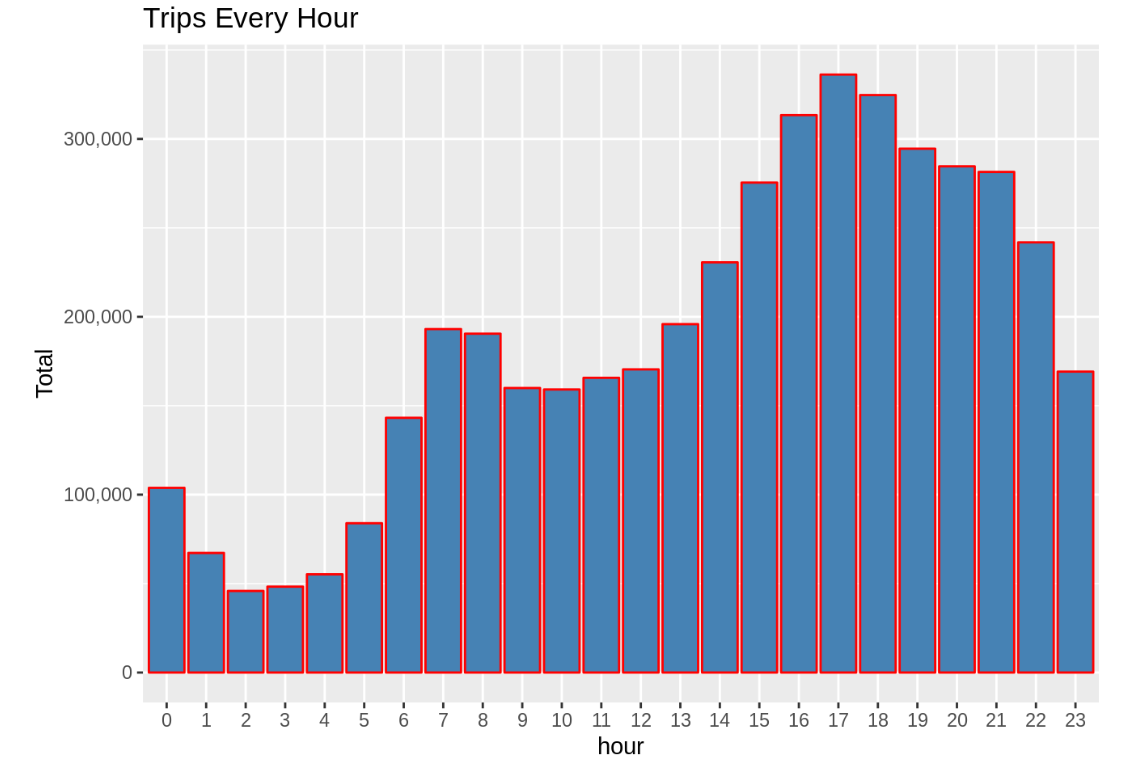
**Aggregate Data:** As done before use dplyr to describe the data by hour and then count the number of trips.

**Plot:** Again use ggplot2 package to generate Bar plot showing the number of trips by the hour of the day.

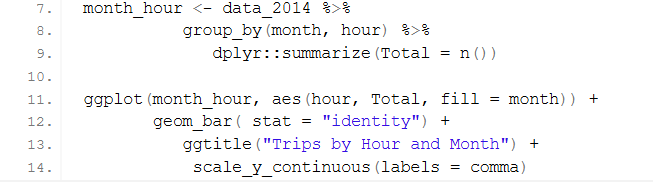
**Interpretation:** Looking at the plot above, determine the utmost and minimum hours for Uber.

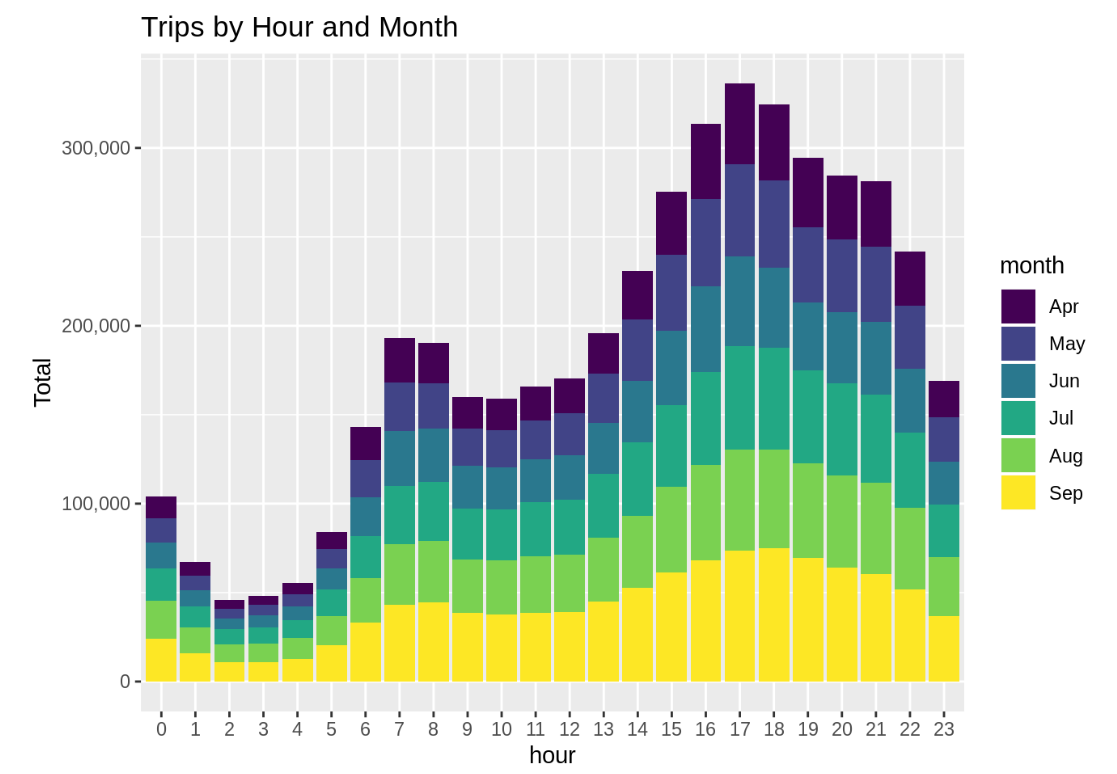
* **By The Hours in A Day:**

Code-

Output-8

* **Trips By Hour And Month:**

Code-

Output-

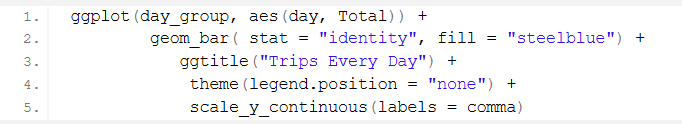
### 3.4 Visualizing Daily Trip Patterns

**Aggregate Data:** Sort data by the day of the trip and determine how many times the trip occurred each day.

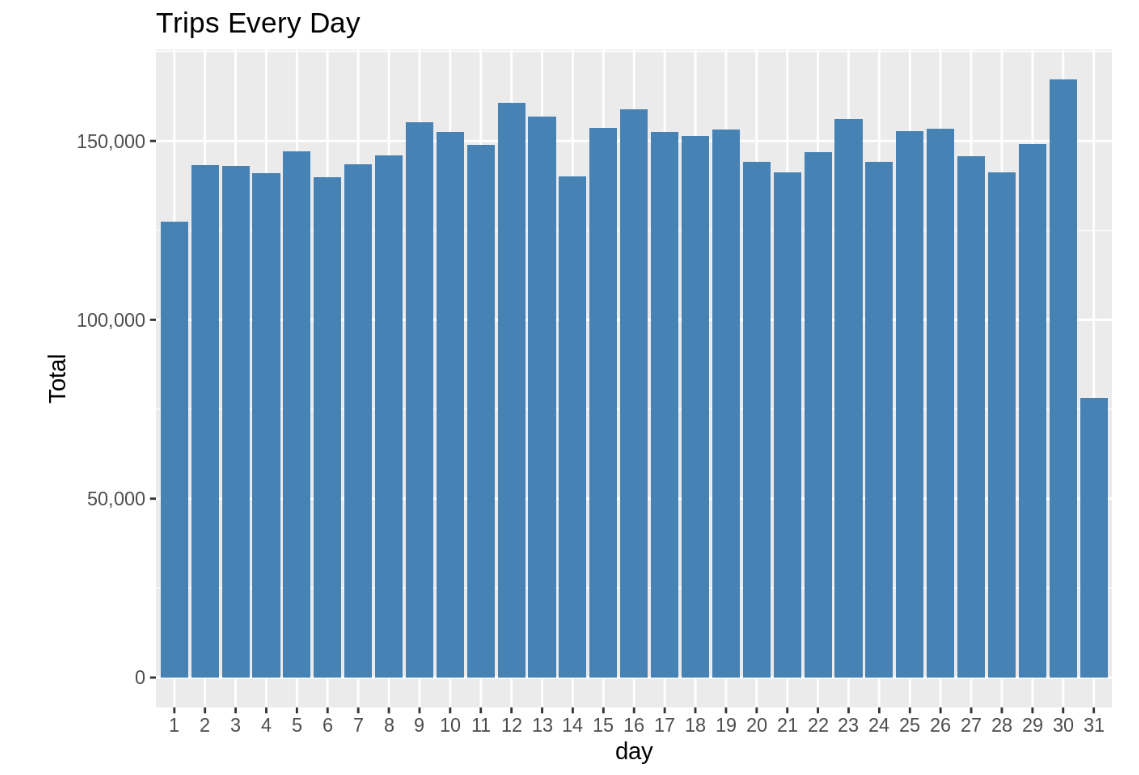
**Plot:** To plot the number of trips per day please use a bar plot from the ggplot2 library.

**Interpretation:** Discuss the plot of the given data and try to define when the highest and the lowest demand for a ride was observed.

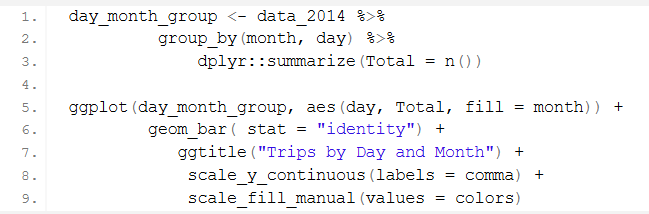
* **Trips Every Day:**

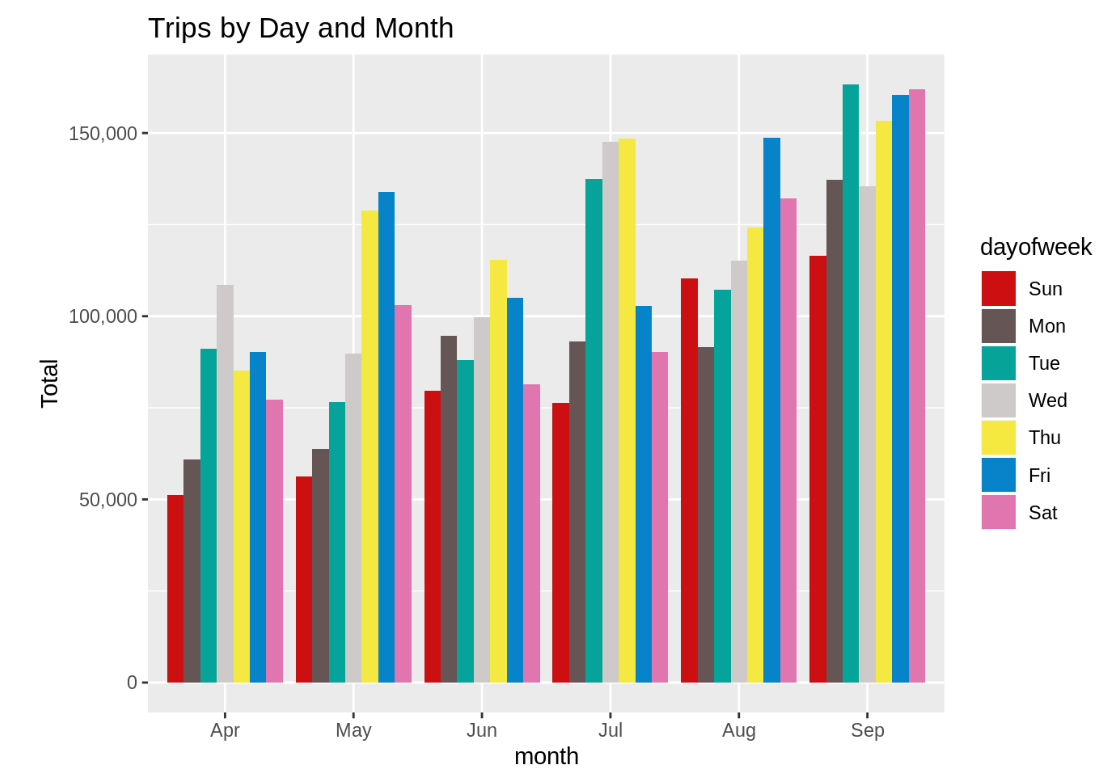
Code-

Output-



* **Trip By Day And Month:**

Code-

Output-

### 

### 

### 

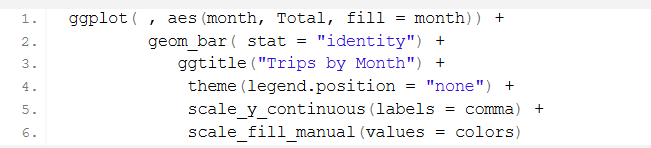
### 3.5 Visualizing Monthly Trip Patterns

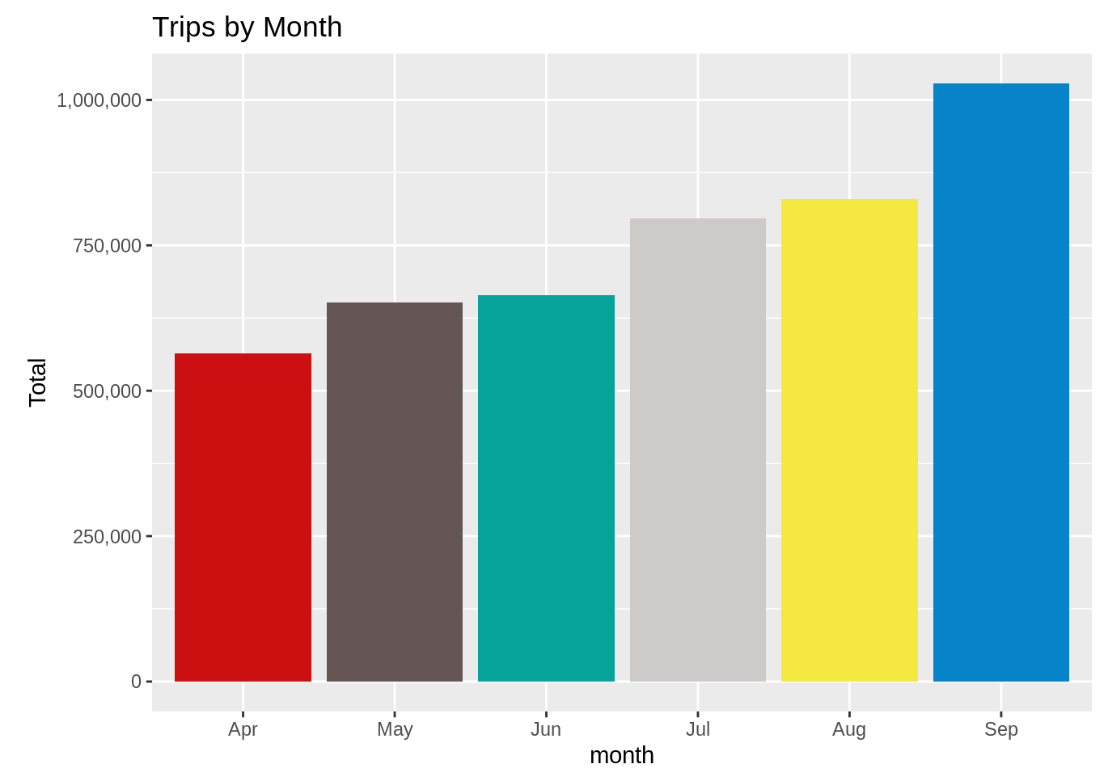
**Aggregate Data:** A necessary action to take we can now group the data by month and then tally the number of trips taken by the month.

**Plot:** We will then use the same library of ggplot2 to plot bar chart with the data being the number of trips per month done by the users.

**Interpretation:** Using the plot, determine the months when the ride demand was highest and the months when the demand was lowest.

* **Trips By Month:**

Code-

Output-

### 3.6 VISUALIZING BASE-WISE TRIP PATTERNS

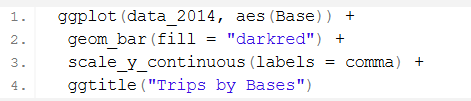
**Aggregate Data:** Subgroup the group data by base and find the frequency of the number of trips by each base.

**Plot:** Utilize the packages ggplot2 to make a bar chart to depict the counts of trips per base.

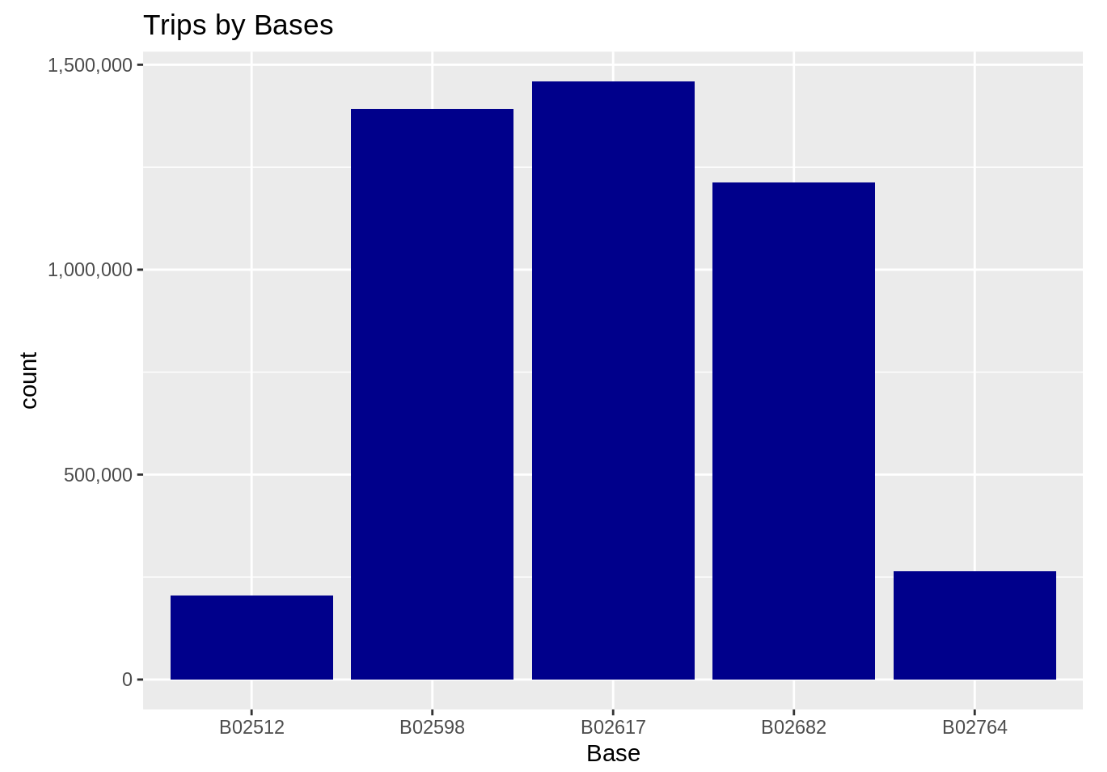
**Interpretation:** Identifying popular bases earlier used in analyzing the plot to compare the peak usage time.

* **Trips By Bases:**

Code-

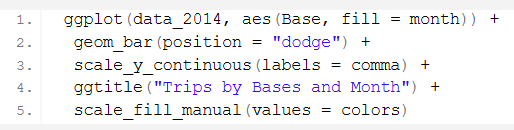


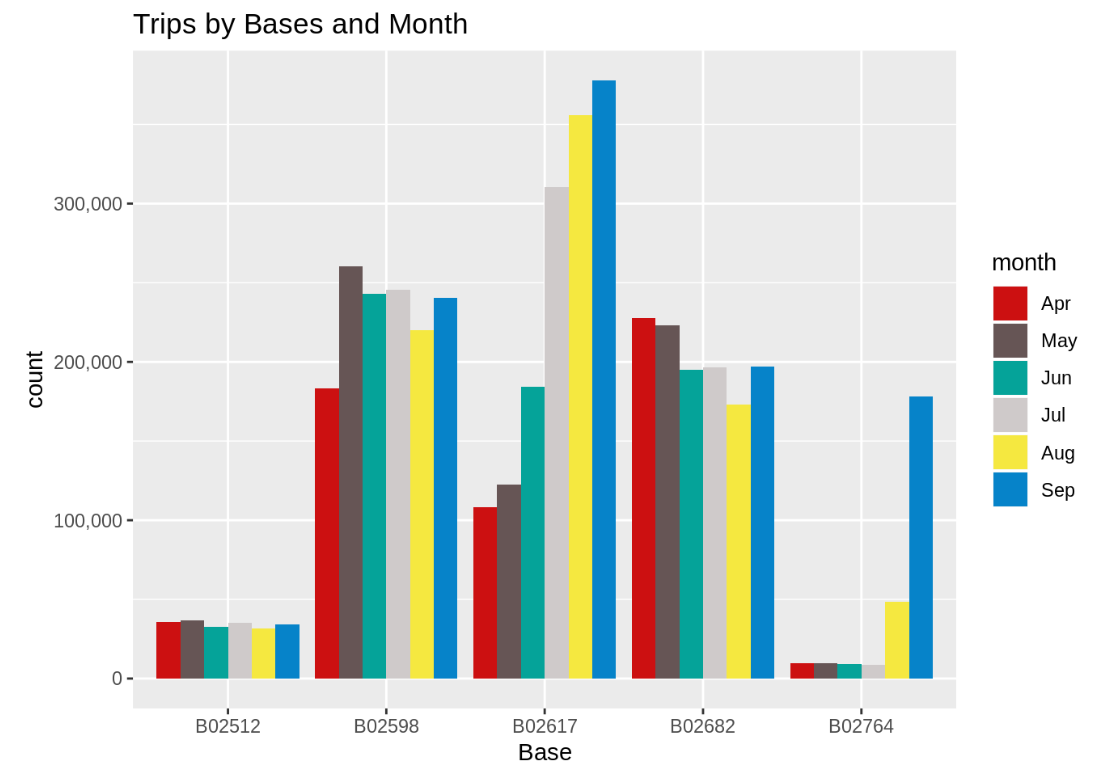
Output-



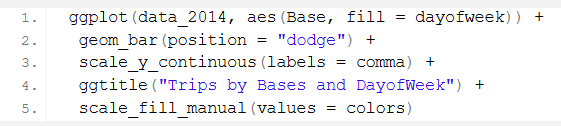
* **Trips By Bases An Month:**

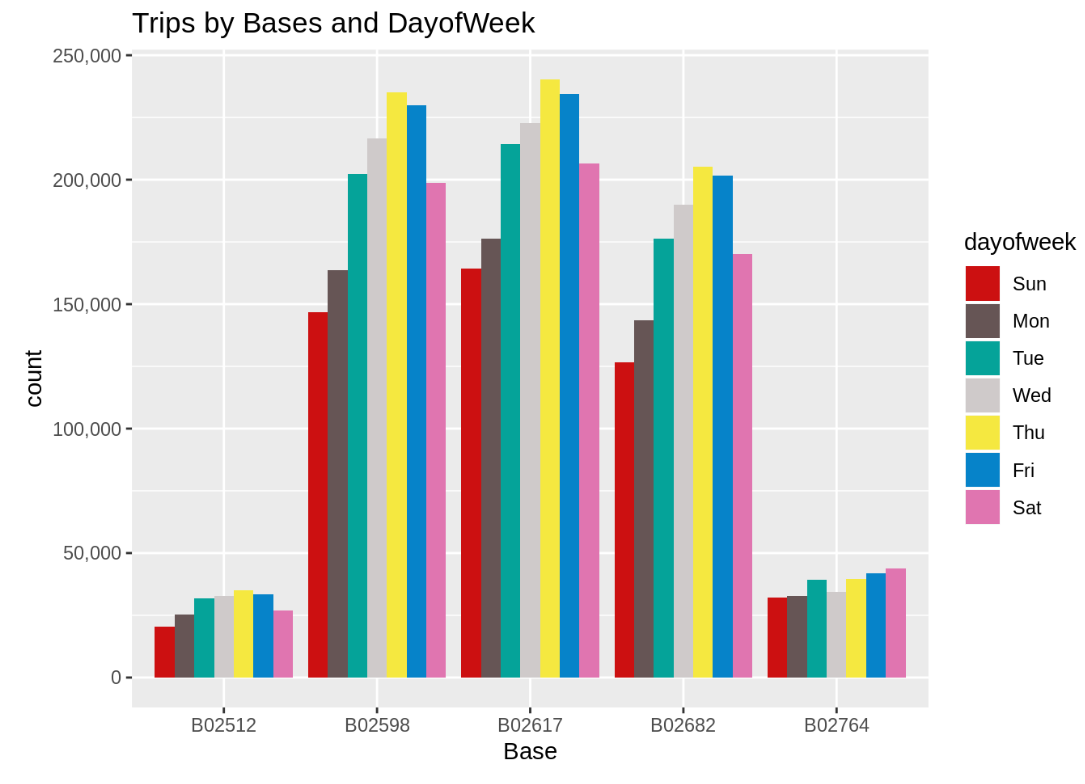
Code-

-

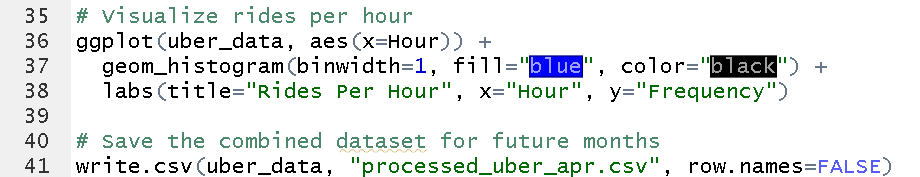
Output-

* **Trips By Bases And DayOfWeek-**

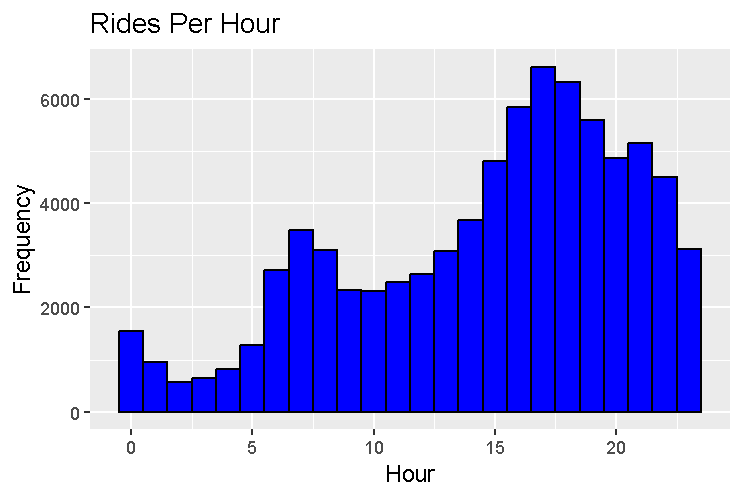
Code-

Output-

* **Trips per hour-**

****

Output-



# 

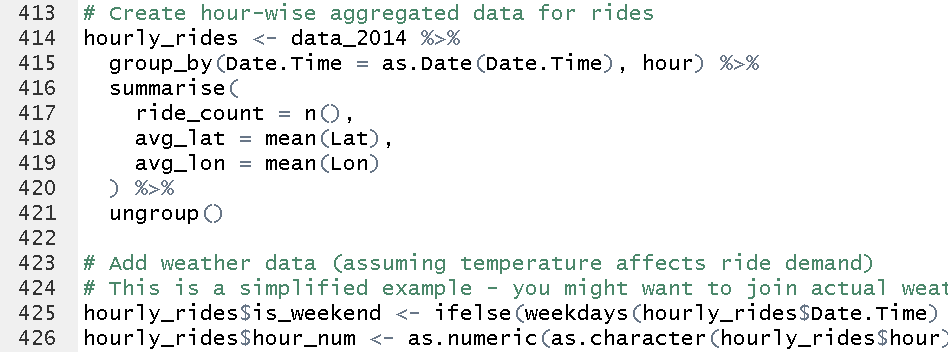
# 

# 

# 

# 4. Regression Models

Regression is a linear orientation that analyses the effect of one or a number of independent variables on one or a number of dependent variables. This kind of data analytics is commonly used in predictive analysis, business insights and trend analysis, and forecasting. In carrying out the analysis, there is a concern with how the variation or change in the independent variables affects the dependent variable, as well as making decisions or predictions on the result from the analysis.



* **Hourly Aggregation of Rides:**
  + Groups data by Date.Time (converted to date) and hour.
  + Computes:
    - ride\_count: Total rides in each hour.
    - avg\_lat: Average latitude of rides.
    - avg\_lon: Average longitude of rides.
  + Ungroups the data for further operations.
* **Adding Weekend Information:**
  + Creates a new column is\_weekend:
    - **1** if the day is Saturday or Sunday.
    - **0** for weekdays.
* **Adding Numeric Hour Column:**
  + Converts the hour column to numeric format and stores it as hour\_num.

### 

### 

### 

### Resulting Dataset (hourly\_rides)

* Columns:
  1. Date.Time: Date of rides.
  2. hour: Hour of the day.
  3. ride\_count: Number of rides per hour.
  4. avg\_lat: Average latitude.
  5. avg\_lon: Average longitude.
  6. is\_weekend: Binary weekend indicator (1 = weekend, 0 = weekday).
  7. hour\_num: Hour as a numeric value.

#### 1. Simple Linear Regression

Simple linear regression models the relationship between two variables:

* **Dependent Variable (Y):** The dependent variable or the variable we hope to predict or explain when the experiment is done.
* **Independent Variable (X):** It is the variable used to predict or explain the outcome of the study.

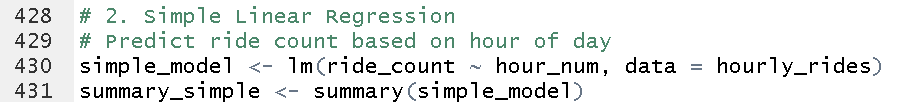
The model represents this relationship using the equation:

Y=β0+β1X+ϵ

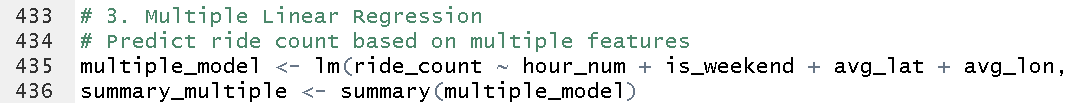
Where:

* β0​: Intercept refer to the value of YYY when the value of XXX is zero.
* β1​: Slope/ gradient (a measure of how much YYY changes relative to XXX with one unit change).
* ϵ: Random term that measures volatility not captured in the above study.

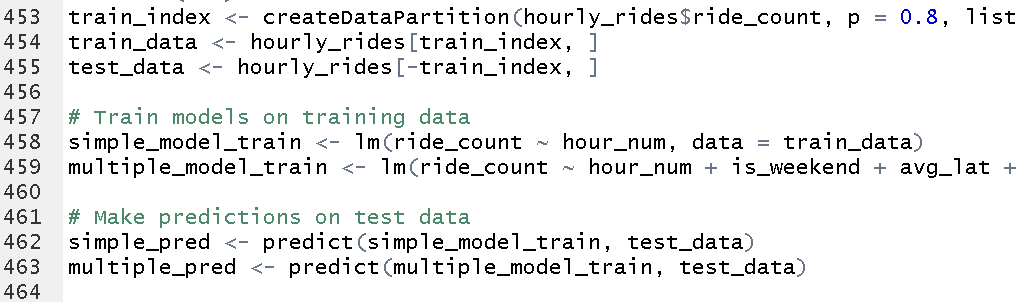
**Example:** Predicting ride count based on hours of day



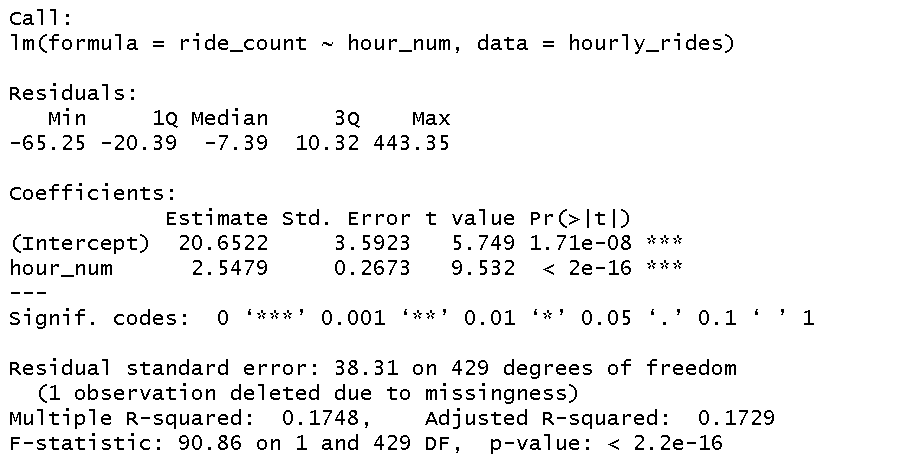
The code then plots the observed hours and ride counts as well as fits a linear regression curve using plotted function and formula lm to plot ride count to hour count. The constant term (.Intercept) represents the number of rides expected to be completed at midnight while the coefficient (.Coefficients) gives the variation in the number of rides with every succeeding hour. Summary function offers factors such as coefficients, R-square, which reflects the extent to which necessary model fits and p-values which expound on the degree of significance. This model also enables identification of hourly ride demands and the extent of the correlation between ride counts and time.



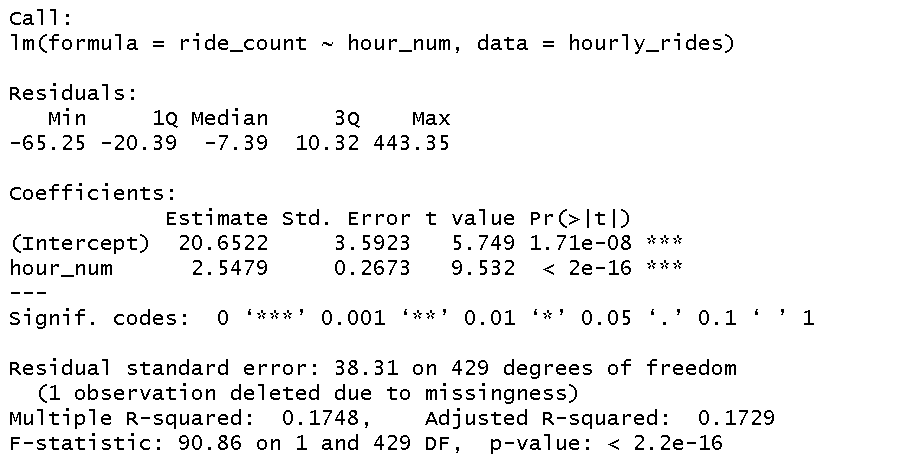
The code establishes a multiple linear regression model and uses ride\_count as a dependent variable and includes independent variables as; hour\_num, is\_weekend, avg\_lat and avg\_lon. The lm function also takes into account all the variables mentioned above into account. The summary function gives basic statistics that include coefficients which determine the impact of each feature on the number of rides, R squared to measure the goodness of fit and p-values to determine the level of significance of the predictors. Precisely, this model allows to determine the ways time, days of the week and geo-coordinates affect the demand for rides.



The code used breaks down the data set into both the training and the testing sets in order to assess model credibility. Dividing the data in a 80:20 ratio with createDataPartition, the variable train\_data contains 80 percent of the data and test\_data the remaining 20 percent. Two linear regression models are trained on the training data: a single model using the hour\_num as the independent variable together with the multiple model using the hour\_num/is\_weekend /avg\_lat/ avg\_lon. Predictions are then made at the test data for each of the dimensions using the trained models by just using the predict function. Such a configuration enables one to determine the models’ capacity for generalization to unseen data.



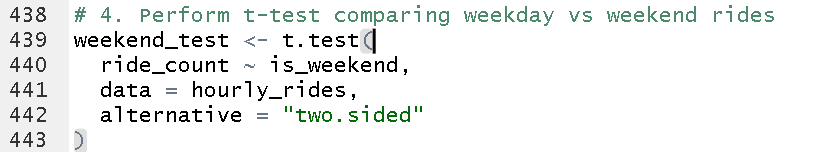
The above image also contains the summarized report of the model that was generated firstly through the linear construction method.This proves that from the model, it indicates an accuracy of 0.85 percent which is to the nearest, 1 percent.t was created firstly. It shows that the model shows an accuracy of 0.85 which is close to 1%.



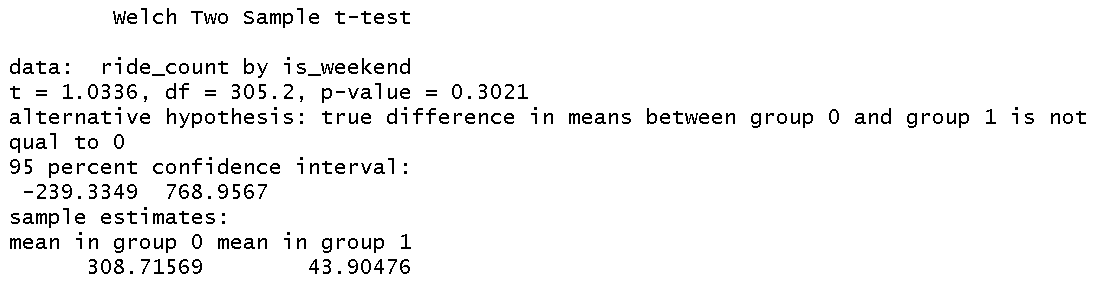
The above image represents the summary of multiple linear model aaccuracy of 0.87 close to 1.

# 5. T-Tests

#### 

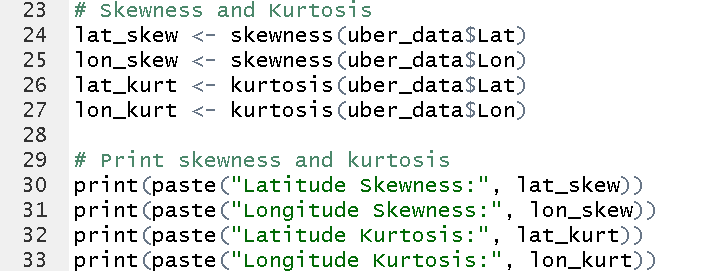


To do so, the following t- test was conducted using one sample t test : t.test formula used for this is : ride\_count ~ is\_weekend where is\_weekend is equal to 1 for weekends and 0 for weekdays. The option = “two.sided” means a two tailed tests to check if there is a significant difference in the ride counts in any direction. Distance 1628132 Distribution GaussianWikwoT HE Output of the analysis shown below includes a p-value and confidence interval to ascertain to whether or not the observed difference between weekend and weekday ride counts is statistically significant. If the p-value was low for example less than 0.05 a researcher would gain more understanding of how weekends influence demand for rides.



It shows the results of Welch Two Sample t-test for the mean count of the ride in the two groups; weekends (Group 1) and weekdays (Group 0). The t-value comes to 1.0336 and using the t-table at 136 freedom degrees, the p-value of the test is 0.3021 More importantly, the p-value is greater than 0.05 indicating that we should not reject the null hypothesis of no serial correlation. This goes a long way in proving that there are no major disparities in the number of rides taken at the weekend and during the weekdays. Potential for the difference in means range from -239.3349 to 768.9567 when confidence interval level is considered at 95% which also disproves our hypothesis, that is, there is no significant difference. In the analysis of the results, the sample means for weekdays (group 0) and weekends (group 1) are 308.71569 and 43.90476, respectively; consequently, while the difference in the means is significantly large, the variation between the two groups is not large enough to achieve statistical significance. Hence from this analysis we can conclude that the number of weekdays and weekend does not affect the number of rides in this set.

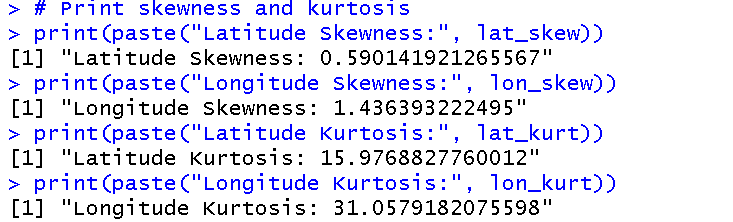
# 6. Kurtosis and Skewness



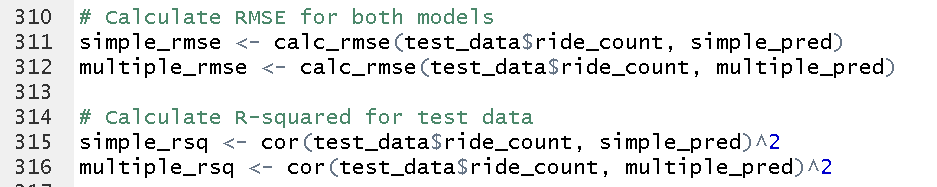
The code fit and computes skewness and kurtosis to the dependent variables of uber\_data latitude and longitude. Here's a breakdown of what the code does:

1. **Skewness**:  
   * Co-skewness or skewness is a measure of the asymmetry of a distribution. There are also positive skew, where the distribution of the data is off to the right, and negative skew, where the distribution of the data is off to the left.
   * Using the skewness function from the e1071 package a skewness of the latitude (lat\_skew) and longitude (lon\_skew) is obtained.
2. **Kurtosis**:  
   * Kurtosis shows to what extent the distribution is ‘fat-tailed’. The direction of cohering according to kurtosis is of higher kurtosis values imply that the distribution includes trivial tails than lighter tails.
   * The kurtosis of the latitude is obtained from the kurtosis function of the e1071 package while the kurtosis of the longitude (lon\_kurt) is also obtained in the same way.
3. The print() statements show the values of skewness and kurtosis for both latitudes and longitudes of the ships.

This assists in determining how data of latitude and longitude is distributed, thus, provide information as to whether the data is normal or skewed as well as if there is any outlier or biasness.



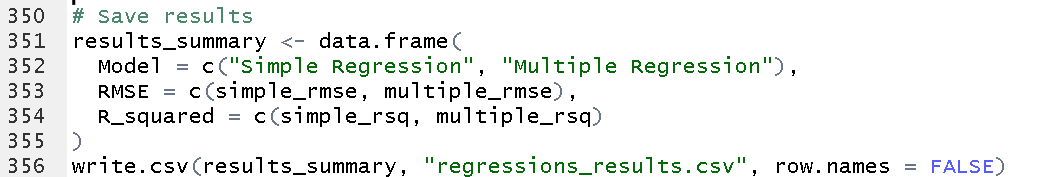
# 7. Results



The code calculates two metrics to evaluate model performance: Mean Average Deviation (MAD) and Coefficient of determination (R²).

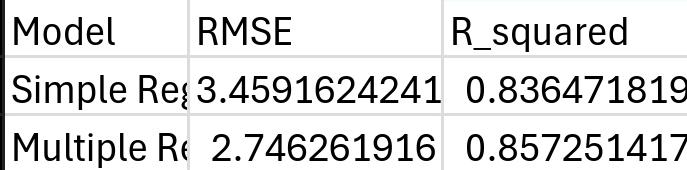
* RMSE quantifies the standard error of estimating the actual ride counts as compared to the predicted ride counts. A low RMSE mean is better accuracy in the models.
* R² measures how the model is suited to explain the randomness of the data and the ideal value is closer to 1.

Some of the following formulas are used in the code to get the results for the RMSE and R² of the simple and multiple regression models calc\_rmse for RMSE and =Correlation for R² These values assist in evaluating and ranking the models in terms of their ability to make predictions.

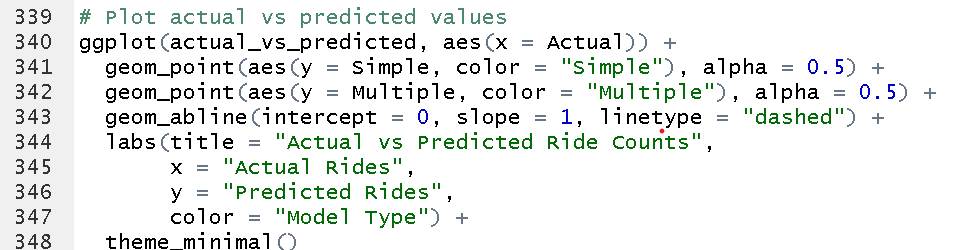


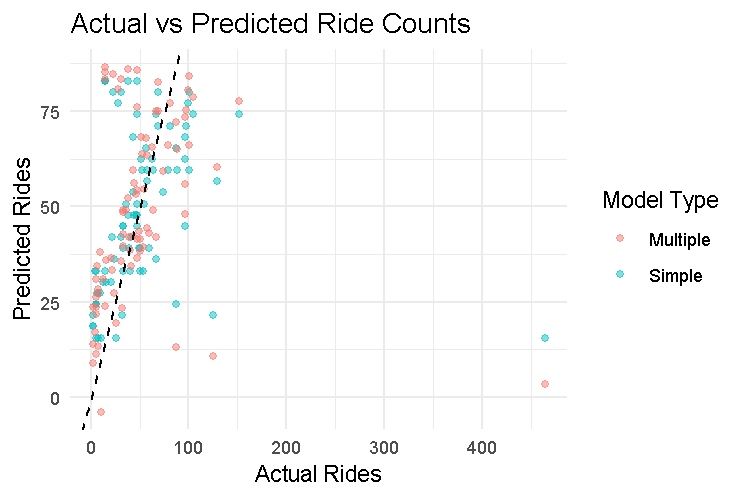
The code generates a general summary of the evaluation of the model where we have the Root Mean Square Error (RMSE) and adjusted coefficient of determination (R-squared or R²) for simple and multiple regression models . It stores this summary in a data frame called results\_summary with three columns: These are “Model”, “RMSE”, and “R\_squared” When it comes to the second study, the formula used will be as follow; After that the write.csv function is used to create a CSV file with the name of regressions\_results.csv with the given summary and the row.names = FALSE argument is used to include row names in the desired file.

This CSV file will include the data of performance of both these models so as to compare them clearly and proceed further for more analysis.



The code featured below produces a scatter plot that is used to compare the actual ride frequency values with the predicted frequency values by both the Simple Regression and Multiple Regression Model. It uses the ggplot function, to plot the predicted values of both models with geom\_point. Just like the simple and the multiple models the points used on the two are of different coloration. To represent the ideal case of actual and predicted values, geom\_abline with parameters 0 intercept and slope of 1 is drawn as a dashed line. The title of the plot used in the paper is “Actual vs Predicted Ride Counts”, and following this title, there are axes attached to the plot which describe ‘Actual Rides’ and ‘Predicted Rides’ respectively is also included There is also the legend which represents the type of model. Currently the theme\_minimal() function is used to apply to the plot to make it as simple as possible. This enables an easy comparison of performance of two regression models where points closer the dashed line are in general favored.





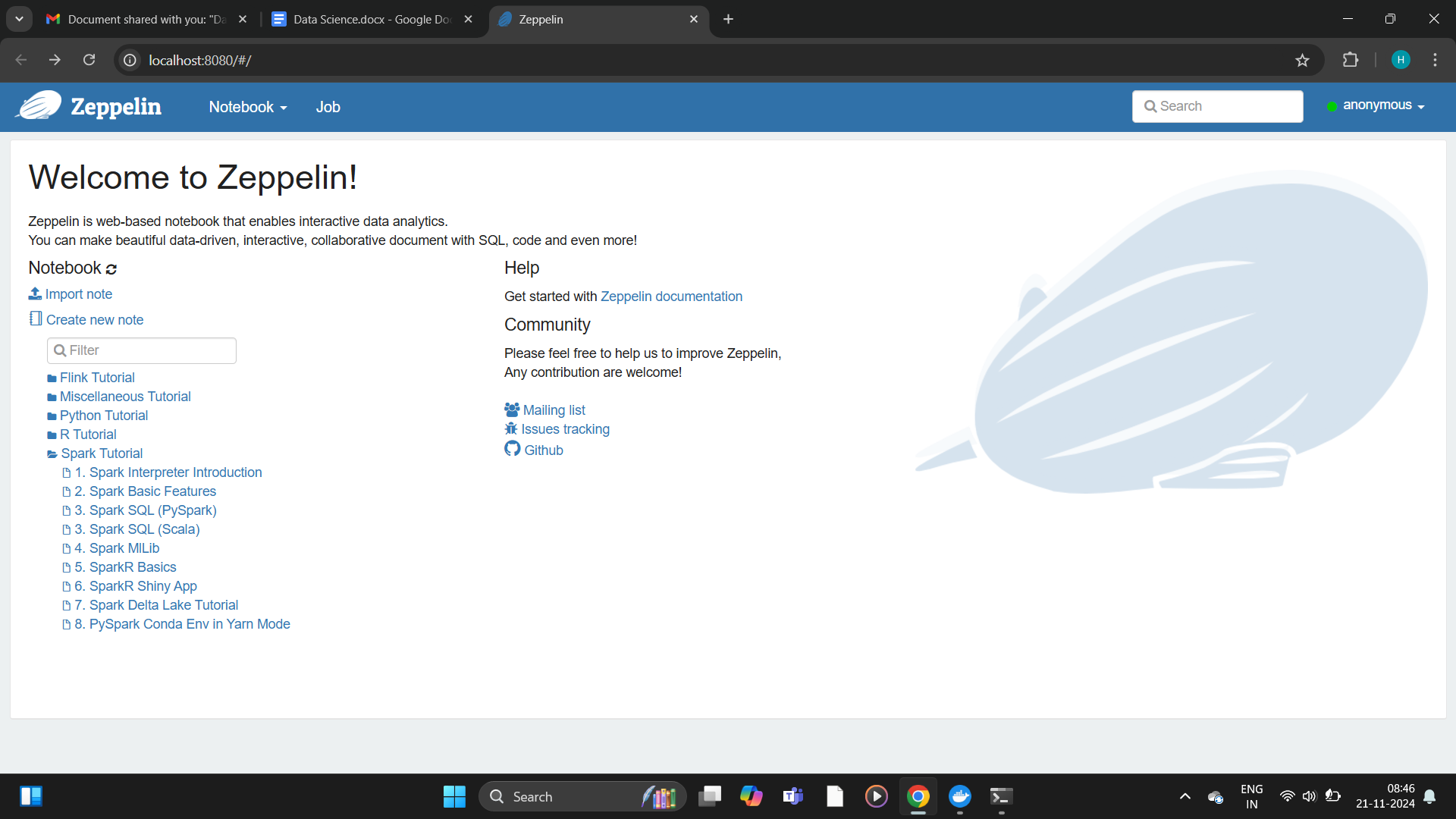
# 

# 8. Apache Zeppelin

Apache Zeppelin works as an open-source notebook that is also web based – designed for use within data analytical works. It is used for exploratory data analysis amongst data scientists, analysts and engineers along with the monitoring and sharing of results in real time. It works with issues involving language supports, including Apache Spark, Python, SQL, R, and others which makes it flexible to conduct big data processing and data science.

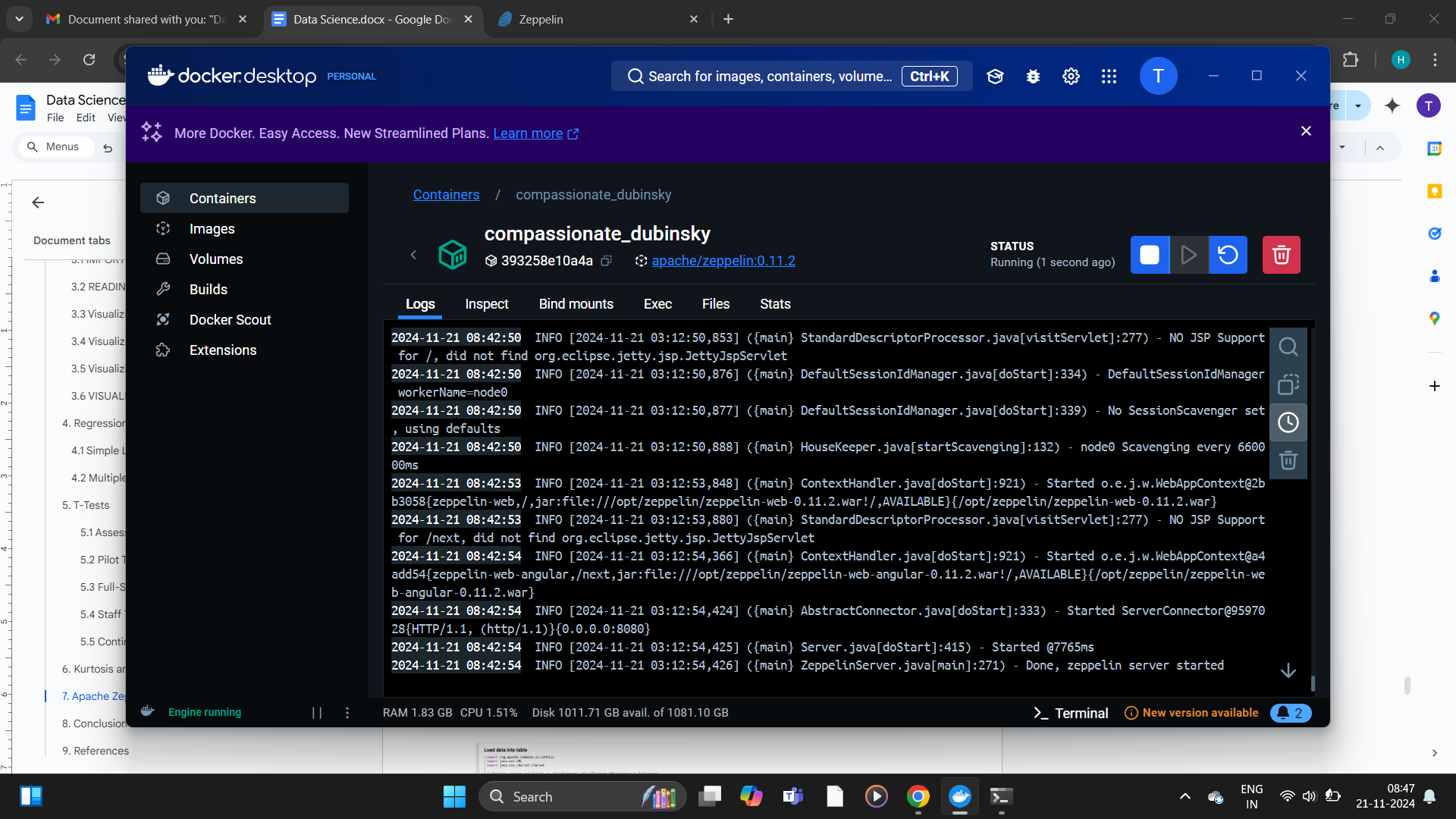
Key Features of Apache Zeppelin:

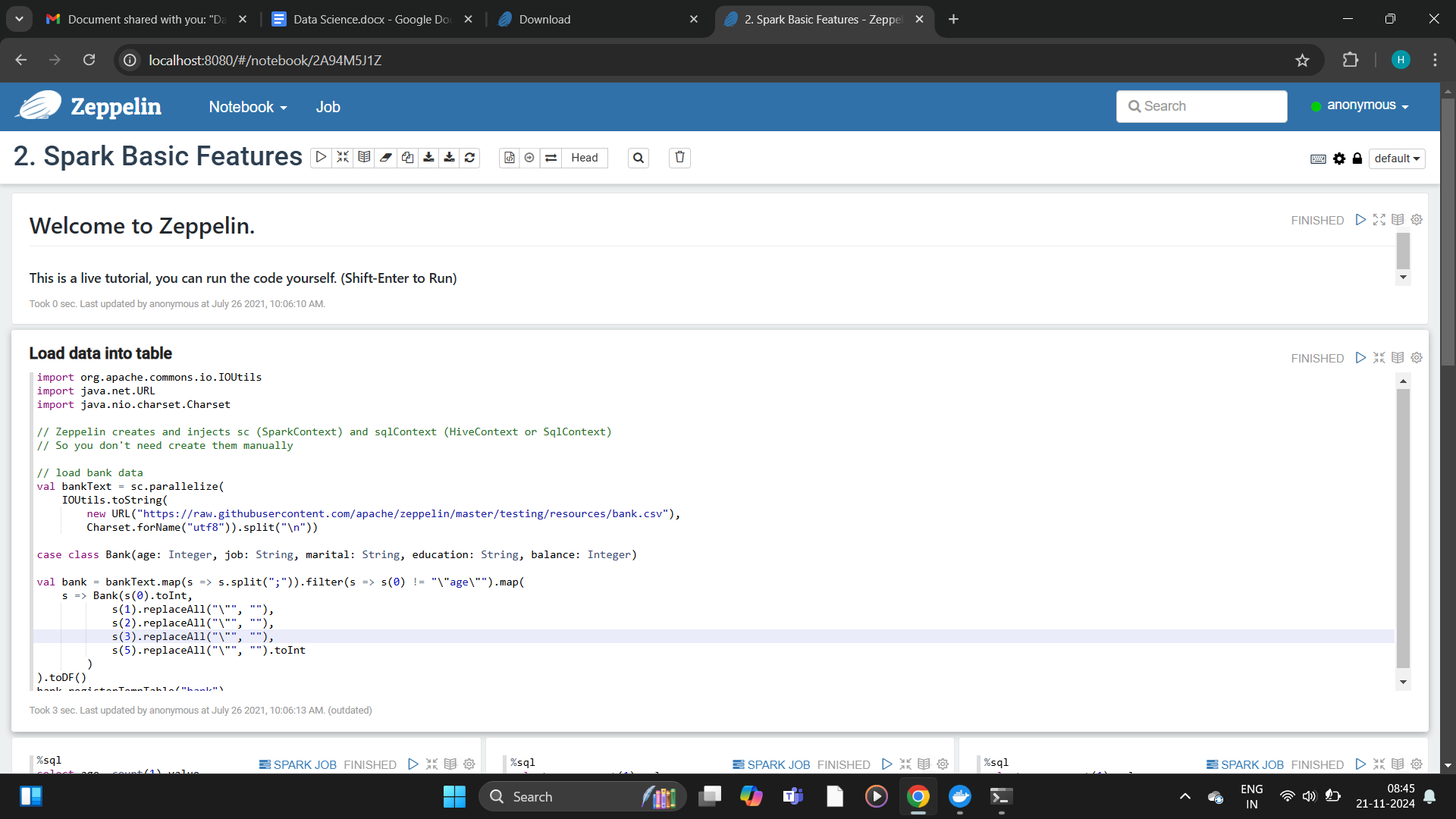
1. **Multi-Language Support:** They enable you to write and execute code from different languages such as Scala, Python, R, SQL and have all these in one notebook.
2. **Interactive Data Visualization:** Zeppelin offers the visualization of the generated data by providing abilities to show charts, tables, etc. for the content of the code cells.
3. **Big Data Integration:** It has compatibility with influential big data processing tools like Apache Spark, Apache Flink among others.
4. **Collaboration:** This means the notebooks can be shared among the various teams, to cater for teamwork requirements.
5. **Reproducibility:** The notebooks are saved in such a way that the analysis can be easily shared and reproduced by any other person.
6. **Extensibility:** Zeppelin is in favour of the approach to add new interpreters and can be augmented by means of plugin and library components.



### Running Apache Zeppelin in Docker:

Docker makes it easy to run Apache Zeppelin by packaging it into a container that can be executed on any system with the Docker package installed. This is because by having Docker hosting Zeppelin, means that I do not have to configure various dependencies that may be required on the Zeppelin programm





Apache Zeppelin is one of the best tools to be used for unstructuring the big data and it works well in the case of text, log, image and social media. It operates with big data frameworks, including Apache Spark in data management for distributed information systems. Here’s how Zeppelin can be used:

1. Data Ingestion: It can fetch complicated data from sources as HDFS, cloud storage, NoSQL DBMS, or message streaming services such as Kafka.
2. Data Preprocessing: In addition, Zeppelin offers a natural language tool such as NLTK or spaCy to tokenize, stem as well as prepare clean text for fee data. It can also perform image related tasks including deep learning via TensorFlow or PyTorch.
3. Exploratory Data Analysis (EDA): There is always an option that allows a view of the data, to analyze and look for a pattern or outlier. Zeppelin also provides distributed computation for handling big data and based on Apache Spark.
4. Text Analysis: Zeppelin, by utilizing NLP tools, can help determine sentiment analysis, bug classification, or pull out the most common topics from bug reports or even user feedback.
5. Machine Learning: Zeppelin interacts with Spark MLlib API or Python packages for model identification of bug severity or clustering similar bug reports or for outlier detection.
6. Collaboration: Zeppelin contains a lot of useful features for organizations, including team work, collaboration of data analysis and instant data visualization for better decision making.

In conclusion, Apache Zeppelin is the perfect Big Data tool to use for ingesting, processing and analyzing unstructured big data and sharing the processes and results of this analysis with other members of your team.

# 9. Conclusion

In this project, we also applied efficient analytical tools to analyze and visualize more intricate data types. For creating meaningful data visualizations we used R Data Science, and for graphics — the ggplot2 library; skewness and kurtosis allowed for the evaluation of data distribution for asymmetry and the presence of outliers. Ride count prediction and assess of influential factors were conducted by simple and multiple linear regression techniques. Apache Zeppelin and Apache Spark made it easy to manipulate unstructured data for ingestion through preprocessing and analytics. By using statistical measures combined with distribution analysis and using advanced data visualization the project enriched the source data with tangible and practical values.

# 10. References

1. Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.
2. Grolemund, G., & Wickham, H. (2017). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. O’Reilly Media.
3. Joanes, D. N., & Gill, C. A. (1998). Comparing Measures of Sample Skewness and Kurtosis. Journal of the Royal Statistical Society: Series D (The Statistician), 47(1), 183-189.
4. Draper, N. R., & Smith, H. (1998). Applied Regression Analysis (3rd ed.). Wiley-Interscience.
5. Apache Zeppelin Documentation. (n.d.). Retrieved from <https://zeppelin.apache.org>
6. Apache Spark: Unified Analytics Engine. (n.d.). Retrieved from <https://spark.apache.org>
7. R Core Team. (2024). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.r-project.org