Applied Data Science with R Capstone project

Ayan Das

19 - Feb - 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Analysis Overview
  - Data collection and preprocessing.
  - Weather impact on bike demand.
- Data Exploration
  - Identify weather-bike correlations.
  - Explore trends and patterns.
- Model Development
  - Build predictive regression models.
  - Evaluate model performance metrics.
- Dashboard Implementation
  - Integrate models into interactive dashboard.
  - User-friendly features for visualization.
- Conclusion & Recommendations
  - Weather influences bike demand.
  - Optimize bike availability based on forecasts.

Introduction

- Overview of bike-sharing demand analysis.
  - Analyzing rental patterns to optimize bike-sharing systems.
  - Understanding user behavior and demand fluctuations.
- Importance of understanding weather impact.
  - Weather influences rental behavior, impacting system utilization.
  - Insight into weather-related trends enhances system planning and management.
- Objectives of the analysis.
  - Identify seasonal variations in rental demand.
  - Explore correlations between weather conditions and rental counts.
  - Develop predictive models to forecast rental demand accurately.
- Scope of data collection and analysis.
  - Incorporating Seoul Bike Sharing Dataset.
  - Utilizing OpenWeather API.

Methodology

- Perform data collection
- Perform data wrangling
- Perform exploratory data analysis (EDA) using SQL and visualization
- Perform predictive analysis using regression models
  - How to build the baseline model
  - How to improve the baseline model
- Build a R Shiny dashboard app

Methodology

Organized approach guiding the execution and analysis of data-driven tasks.

Data collection

The summary of Data collection:

1. Seoul Bike Sharing Demand Data Set:

- Data obtained from Seoul Bike Sharing Demand Dataset, including weather information and bike rental counts.
- Key phrases: Accessed Seoul Bike Sharing Demand Dataset, extracted weather and rental bike data.

2. Open Weather API Data:

- Utilized OpenWeather API to access current and forecasted weather data for over 200,000 cities.
- Key phrases: Accessed OpenWeather API, retrieved current and forecasted weather data.

3. Global Bike Sharing Systems Dataset:

- Extracted information from Wikipedia's list of bicycle-sharing systems worldwide.
- Key phrases: Scrapped data from Wikipedia, compiled global bike-sharing systems dataset.

4. World Cities Data:

- Gathered information about major cities worldwide including names, latitudes, and longitudes.
- Key phrases: Collected city data, including names and geographic coordinates.

Flowchart:

- 1. Seoul Bike Sharing Demand Data Set ⟶ Extract Weather & Rental Bike Data
- 2. Open Weather API Data ⟶ Retrieve Current & Forecasted Weather Data
- 3. Global Bike Sharing Systems Dataset ⟶ Scrape Data from Wikipedia
- 4. World Cities Data ⟶ Collect City Information

This approach ensures comprehensive data collection from multiple sources for robust analysis.

Data wrangling

The summary of the Data wrangling:

1. Standardization of Data Sets:

- Standardized column names for improved readability.
- Key phrases: Renamed columns uniformly, ensuring consistency.

2. Handling Missing Values:

- Detected and handled missing values to maintain data integrity.
- Key phrases: Identified missing values, imputed or removed as necessary.

3. Conversion of Categorical Variables:

- Converted categorical variables into indicator variables for regression analysis.
- Key phrases: Encoded categorical data, enabling inclusion in regression models.

4. Normalization of Numeric Columns:

- Normalized numeric columns to ensure consistency in scale and mitigate bias.
- Key phrases: Scaled numeric data to similar ranges, enhancing model accuracy.

Flowchart:

1. Standardization of Data Sets ⟶ Rename Columns

2. Handling Missing Values ⟶ Detect & Handle Missing Values

3. Conversion of Categorical Variables ⟶ Encode Categorical Data

4. Normalization of Numeric Columns ⟶ Scale Numeric Data

This systematic approach ensures data quality and prepares the datasets for analysis and modeling.

EDA with SQL

The summary of EDA with SQL:

- Count of Dates in Seoul Bike Sharing Dataset
- Count of Operational Hours with Bike Rentals
- Weather Forecast for Seoul
- Distinct Seasons in Seoul Bike Sharing Dataset
- First and Last Dates in Seoul Bike Sharing Dataset
- Date and Hour with Maximum Bike Rentals
- Top 10 Hours with Highest Average Bike Rentals by Season
- Statistics of Bike Rentals by Season and Hour
- Average Weather Conditions by Season
- Bike-Sharing Systems Data for Seoul
- Cities with Bike-Sharing Systems and Bicycles Between 15,000 and 20,000

EDA with data visualization

The summary of EDA with data visualizations:

- Scatter plot of RENTED_BIKE_COUNT vs DATE
- Scatter plot of RENTED_BIKE_COUNT vs DATE with HOURS as color
- Histogram overlaid with a kernel density curve
- Scatter plot to visualize the correlation between RENTED_BIKE_COUNT and TEMPERATURE by SEASONS
- Four boxplots of RENTED_BIKE_COUNT vs. HOUR grouped by SEASONS
- Grouped the data by DATE, then used the summarize() function to calculate the daily total rainfall and snowfall.

Predictive analysis

The summary of the Predictive analysis:

1. **Data Splitting:**
   - Split the data into training and testing datasets using an initial split function.
2. **Model Building:**
   - Built several linear regression models using different sets of predictor variables:
     - Model 1: Utilized only weather variables for prediction.
     - Model 2: Incorporated both weather and date variables.
     - Model 3: Employed polynomial terms, interaction terms, and regularization techniques.
3. **Model Evaluation:**
   - Evaluated each model's performance using metrics such as R-squared and Root Mean Squared Error (RMSE) on the testing dataset.
4. **Identifying Important Variables:**
   - Analyzed the coefficients of the variables in the models to identify their importance in predicting bike rental counts.

1. **Comparison and Selection:**
   1. Compared the performance of the different models based on their RMSE and R-squared values.
   2. Selected the model with the lowest RMSE and highest R-squared value as the best performing model.

Flowchart:

1. **Data Splitting** ⟶ Split data into training and testing datasets
2. **Model Building** ⟶ Build linear regression models with varying predictor variables
3. **Model Evaluation** ⟶ Evaluate models using RMSE and R-squared
4. **Identifying Important Variables** ⟶ Analyze coefficients of variables in models
5. **Comparison and Selection** ⟶ Compare model performance, select best model

- This iterative process ensures thorough model development and selection, resulting in an optimal predictive model for bike rental counts.

Build a R Shiny dashboard

The summary of the Predictive analysis:

- Integrated a Leaflet-based interactive map showing cities with predicted bike-sharing demand for the next five days.
- Circle markers on the map represent cities, with marker size and color indicating the level of predicted bike demand.
- Pop-up labels provide additional information on each city's location.
- Added a drop-down list to drill down to specific cities. Selecting "All" shows the overview map, while selecting a city reveals its details.
- Used ggplot to render detailed plots showing bike-sharing prediction, temperature, and humidity trends for the selected city.
- Visualized temperature trends for the next five days using a line chart.
- Displayed bike-sharing demand prediction as a trend line, allowing users to click for detailed demand, date, and time information.
- Incorporated a scatterplot to illustrate the correlation between bike-sharing demand and humidity for the selected city.

This comprehensive dashboard allows users to explore and analyze bike-sharing demand and weather trends interactively, facilitating informed decision-making and logistics planning.

Results

Exploratory Data Analysis (EDA) Results:

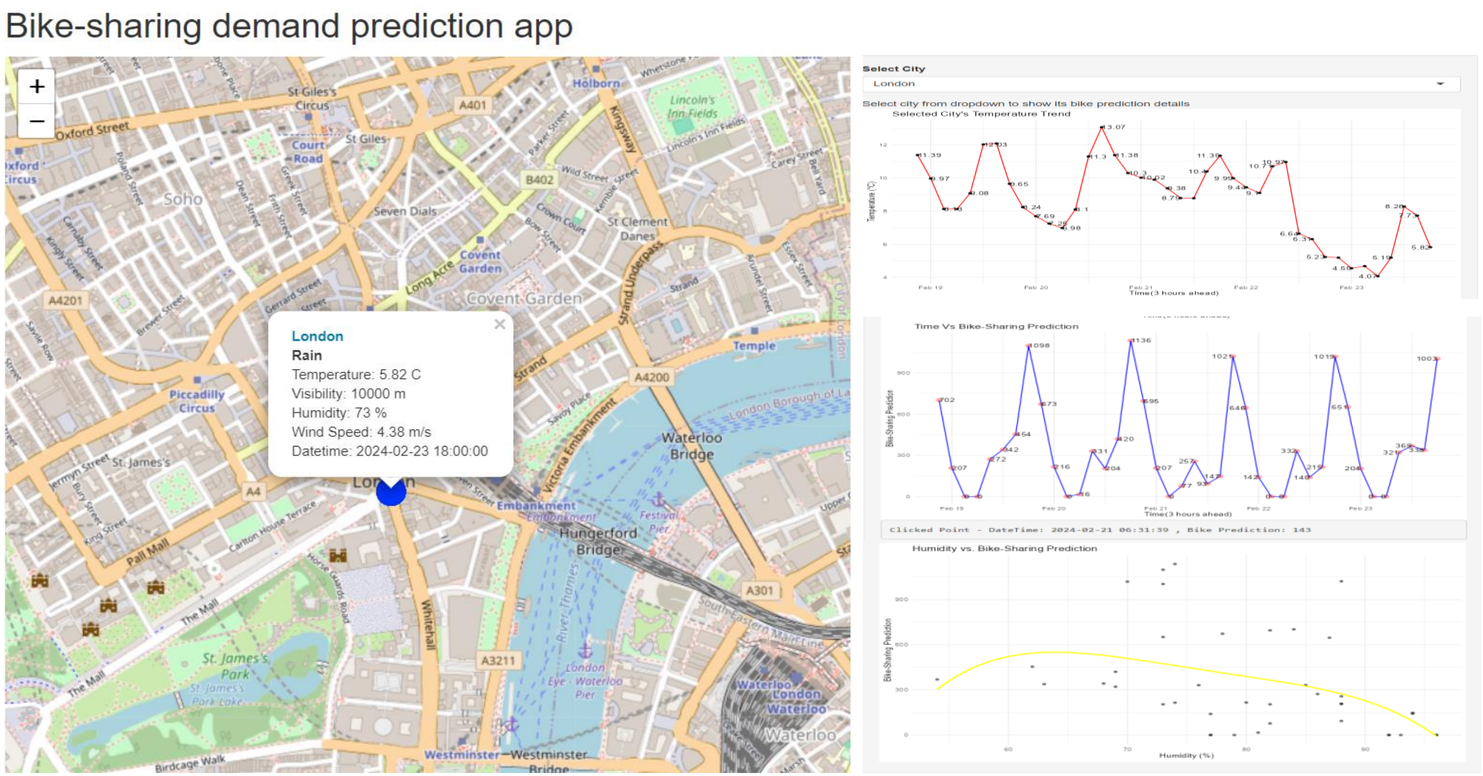
- Examined the distribution and summary statistics of the variables in the dataset.
- Identified any outliers, missing values, or data anomalies that could affect the analysis.
- Explored the relationships between variables through visualizations such as scatter plots, histograms, and correlation matrices.
- Investigated seasonal patterns, trends, and cyclical behavior in the data.
- Assessed the distribution of bike-sharing demand across different times of the day, days of the week, and seasons.
- Evaluated the impact of weather variables (e.g., temperature, humidity, rainfall) on bike-sharing demand using visualizations and statistical analysis.

Results

Predictive Analysis Results:

- Developed regression models to predict hourly bike-sharing demand using weather, date, and time predictor variables.
- Evaluated the performance of the regression models using metrics such as R-squared, RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error).
- Conducted feature selection to identify important variables that significantly influence bike-sharing demand.
- Explored interactions between predictor variables and incorporated them into the models to improve predictive accuracy.
- Utilized regularization techniques (e.g., Lasso, Ridge, Elastic Net) to prevent overfitting and enhance model generalization.
- Compared the performance of different regression models and identified the best-performing model based on evaluation metrics.
- Visualized model predictions and actual bike-sharing demand over time to assess model accuracy and identify any discrepancies or areas for improvement.
- Overall, the EDA and predictive analysis provided valuable insights into the factors influencing bike-sharing demand and yielded effective models for predicting demand under various weather and temporal conditions.

Results



Bike-sharing demand prediction app

London
Rain
Temperature: 5.82 C
Visibility: 10000 m
Humidity: 73 %
Wind Speed: 4.38 m/s
Datetime: 2024-02-23 18:00:00

Select City

London

Select city from dropdown to show its bike prediction details

```
Clicked Point - DateTime: 2024-02-21 06:31:39 , Bike Prediction: 143
```

EDA with SQL

SQL queries to uncover insights and patterns within a dataset efficiently.

Busiest bike rental times

Objective: Determine the date and hour with the highest number of bike rentals.

```
SELECT DATE, HOUR, RENTED_BIKE_COUNT
FROM SEOUL_BIKE_SHARING
WHERE RENTED_BIKE_COUNT = (SELECT MAX(RENTED_BIKE_COUNT) FROM SEOUL_BIKE_SHARING);
```

Result:

Date: 19/06/2018

Hour: 18

Bike Rentals: 3556

Explanation:

This query retrieves the date and hour where the maximum number of bike rentals occurred.

The result indicates that on June 19th, 2018, at 6:00 PM, there were 3556 bike rentals, marking the peak hour of activity.

Hourly popularity and temperature by seasons

Objective: To analyze the hourly popularity and temperature variation across different seasons, providing insights into the peak bike rental hours and corresponding temperatures, aiding in resource allocation and operational planning for bike-sharing services.

```
SELECT SEASONS, HOUR, AVG(RENTED_BIKE_COUNT) AS AvgBikeRentals, AVG(TEMPERATURE) AS AvgHourlyTemperature
FROM SEOUL_BIKE_SHARING
GROUP BY HOUR,SEASONS
ORDER BY AvgBikeRentals DESC
LIMIT 10;
```

Explanation:

This SQL query retrieves the average hourly bike rentals and temperature for each season. It groups the data by hour and season, calculates the average number of bike rentals and the average temperature for each group, and then sorts the results by average bike rentals in descending order. Finally, it limits the output to the top 10 results.

Result:

The result shows the top 10 combinations of season, hour, average bike rentals, and average hourly temperature. For example, in the summer season, hour 18 has the highest average bike rentals of 2135.141, with an average temperature of 29.38791 degrees Celsius.

Rental Seasonality

Objective: To analyze rental seasonality by determining the average hourly bike count, along with the minimum, maximum, and standard deviation of the hourly bike count for each season.

```
SELECT SEASONS, HOUR, AVG(RENTED_BIKE_COUNT) AS AvgBikeRentals, AVG(TEMPERATURE) AS AvgHourlyTemperature
FROM SEOUL_BIKE_SHARING
GROUP BY HOUR,SEASONS
ORDER BY AvgBikeRentals DESC
LIMIT 10;
```

Result:

```
SEASONS     HOUR     AvgBikeRentals     MinBikeRentals     MaxBikeRentals     StdDevBikeRentals
Autumn        0        709.43750             119               1336              219.14298
Spring        0        481.08889              22               1089              253.38673
Summer        0        899.06522              26               1394              285.31199
Winter        0        165.17778              42                342               63.81163
  ...        ...          ...                 ...                ...                 ...
```

Explanation:

This query calculates the average hourly bike count, along with the minimum, maximum, and standard deviation of the hourly bike count for each season. The results provide insights into the rental seasonality, showcasing the variations in bike demand across different seasons and hours of the day. The standard deviation helps to understand the variability or dispersion of bike rentals around the mean, indicating the level of volatility in bike demand during each season.

Weather Seasonality

Objective: Determine the average values of various weather parameters (TEMPERATURE, HUMIDITY, WIND_SPEED, VISIBILITY, DEW_POINT_TEMPERATURE, SOLAR_RADIATION, RAINFALL, and SNOWFALL) per season. Additionally, include the average bike count per season and rank the results by average bike count.

```
SELECT SEASONS, HOUR, AVG(RENTED_BIKE_COUNT) AS AvgBikeRentals, AVG(TEMPERATURE) AS AvgHourlyTemperature
FROM SEOUL_BIKE_SHARING
GROUP BY HOUR,SEASONS
ORDER BY AvgBikeRentals DESC
LIMIT 10;
```

Result:

```
SEASONS   AvgBikeRentals   AVG_TEMPERATURE   AVG_HUMIDITY   AVG_WIND_SPEED   AVG_VISIBILITY   AVG_DEW_POINT_TEMPERATURE   AVG_SOLAR_RADIATION   AVG_RAINFALL   AVG_SNOWFALL
Winter       225.5412         -2.540463        49.74491        1.922685         1445.987           -12.416667                 0.2981806          0.03282407     0.24750000
Spring       746.2542         13.021685        58.75833        1.857778         1240.912             4.091389                 0.6803009          0.18694444     0.00000000
Autumn       924.1105         13.821580        59.04491        1.492101         1558.174             5.150594                 0.5227827          0.11765617     0.06350026
Summer      1034.0734         26.587711        64.98143        1.609420         1501.745            18.750136                 0.7612545          0.25348732     0.00000000
```

Explanation:-

The query calculates the average bike rentals (`AvgBikeRentals`) and various weather parameters (`AVG_TEMPERATURE`, `AVG_HUMIDITY`, `AVG_WIND_SPEED`, `AVG_VISIBILITY`, `AVG_DEW_POINT_TEMPERATURE`, `AVG_SOLAR_RADIATION`, `AVG_RAINFALL`, `AVG_SNOWFALL`) for each season.

- These averages provide insights into the weather conditions and their potential impact on bike rentals during different seasons.
- The results are ordered by average bike rentals, allowing us to observe any correlation between weather conditions and bike rental popularity across seasons.

Bike-sharing info in Seoul

Objective: Retrieve the total bike count and city information for Seoul, including the city name, country, geographical coordinates (latitude and longitude), population, and the total number of bicycles available in the city's bike sharing system.

```
SELECT WC.CITY_ASCII AS CITY, WC.COUNTRY, WC.LAT, WC.LNG, WC.POPULATION, BSS.BICYCLES
FROM  WORLD_CITIES AS WC
JOIN BIKE_SHARING_SYSTEMS AS BSS ON WC.CITY_ASCII = BSS.CITY
WHERE WC.CITY = "Seoul“;
```

Result:

CITY	COUNTRY	LAT	LNG	POPULATIONS	BICYCLES
Seoul	Korea, South	37.5833	127	21794000	20000

Explanation:-

This query retrieves information about Seoul from both the WORLD_CITIES and BIKE_SHARING_SYSTEMS tables.

- It joins the two tables using the CITY_ASCII column.
- Then, it selects the city name (as CITY_ASCII), country, latitude (LAT), longitude (LNG), population, and the total number of bicycles available (BICYCLES) in Seoul.
- Finally, it filters the results to include only the city of Seoul.

Cities similar to Seoul

Objective: Identify all cities with a comparable scale of bicycles available in their bike-sharing systems to Seoul. Retrieve the city names, countries, geographical coordinates (latitude and longitude), populations, and the number of bicycles for each city within the specified range.

```
SELECT WC.CITY_ASCII AS CITY, WC.COUNTRY, WC.LAT, WC.LNG, WC.POPULATION, BSS.BICYCLES
FROM  WORLD_CITIES AS WC
JOIN BIKE_SHARING_SYSTEMS AS BSS ON WC.CITY_ASCII = BSS.CITY
WHERE BSS.BICYCLES BETWEEN 15000 AND 20000;
```

Result:

CITY	COUNTRY	LAT	LNG	POPULATIONS	BICYCLES
Beijing	China	39.9050	116.3914	19,433,000	16,000
Ningbo	China	29.8750	121.5492	7,639,000	15,000
Shanghai	China	31.1667	121.4667	22,120,000	19,165
Weifang	China	36.7167	119.1000	9,373,000	20,000
Xi'an	China	34.2667	108.9000	7,135,000	20,000
Zhuzhou	China	27.8407	113.1469	3,855,609	20,000
Seoul	Korea, South	37.5833	127.0000	21,794,000	20,000

Explanation:-

This SQL query retrieves data from two tables: WORLD_CITIES and BIKE_SHARING_SYSTEMS, using an inner join on the CITY_ASCII column. It selects the city name, country, latitude, longitude, population, and the number of bicycles from the BIKE_SHARING_SYSTEMS table where the bike count falls within the range of 15,000 to 20,000. The result provides information about cities with bike-sharing systems comparable in scale to Seoul.

EDA with Visualization

Various data visualization techniques to explore and understand the underlying patterns and relationships within a dataset.

Bike rental vs. Date

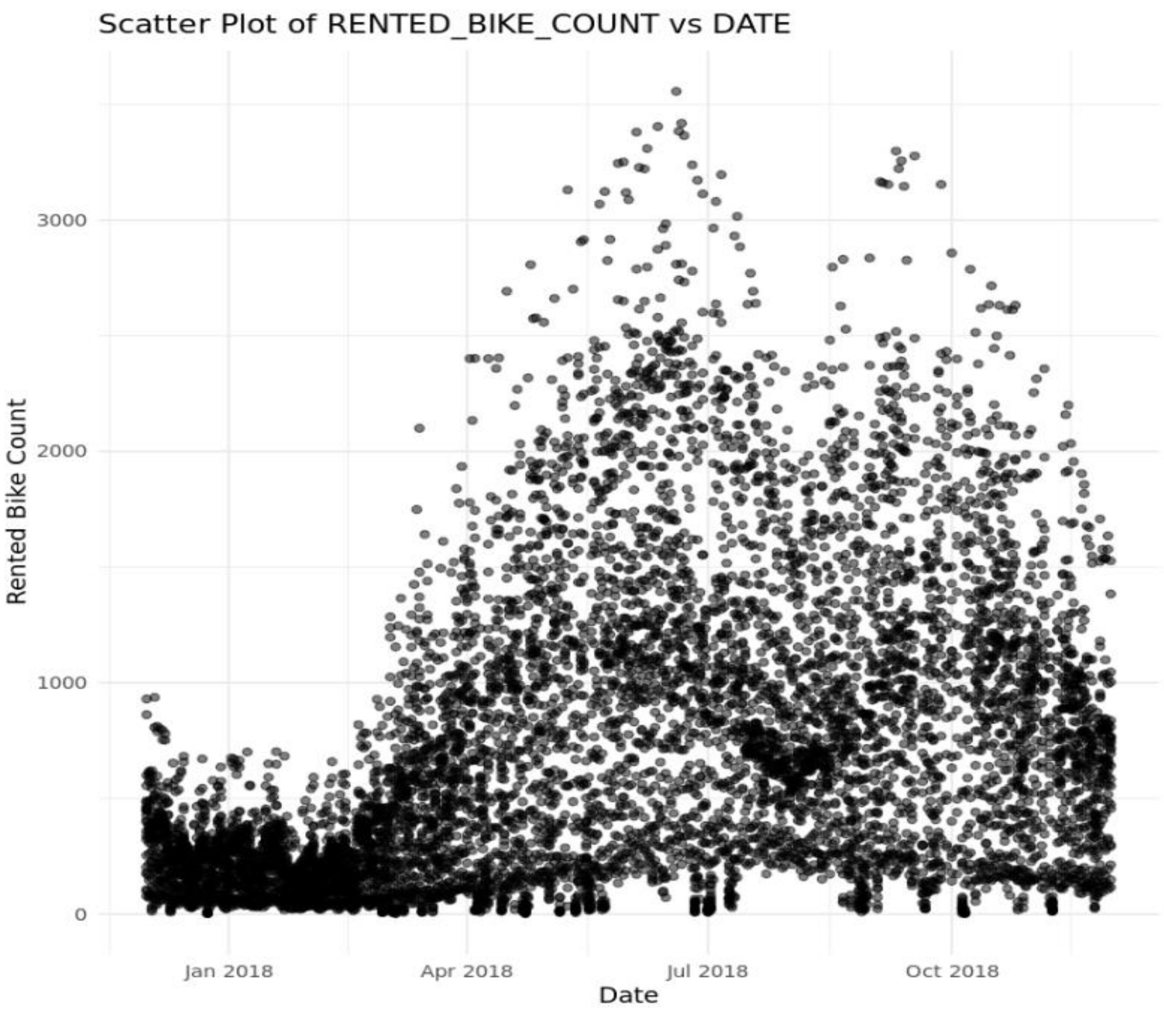
Seasonal Patterns

•Peak bike rental months: May and June (> 3000 counts).

•Strong rental activity: September and October.

•Lower demand: December and January.

•Distinct peaks in June and September.



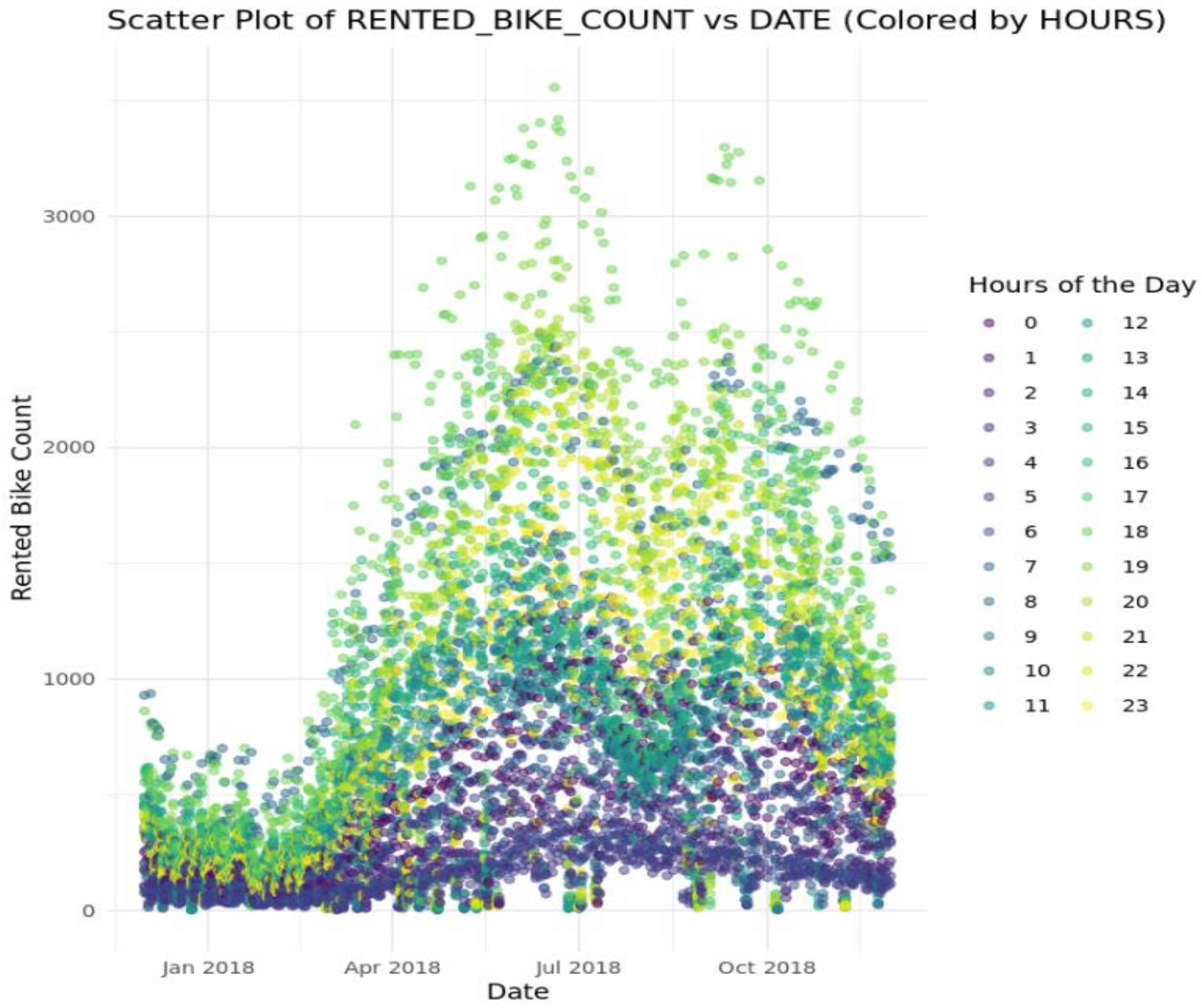
Bike rental vs. Datetime

Daily Trends

•Peak rental hours: 8 pm to 10 pm, especially in May and June.

•Morning rentals stable from Nov to Feb (100 bikes, 4 am to 7 am).

•Apr to Oct morning rentals increase (300-400 bikes).



Bike rental histogram

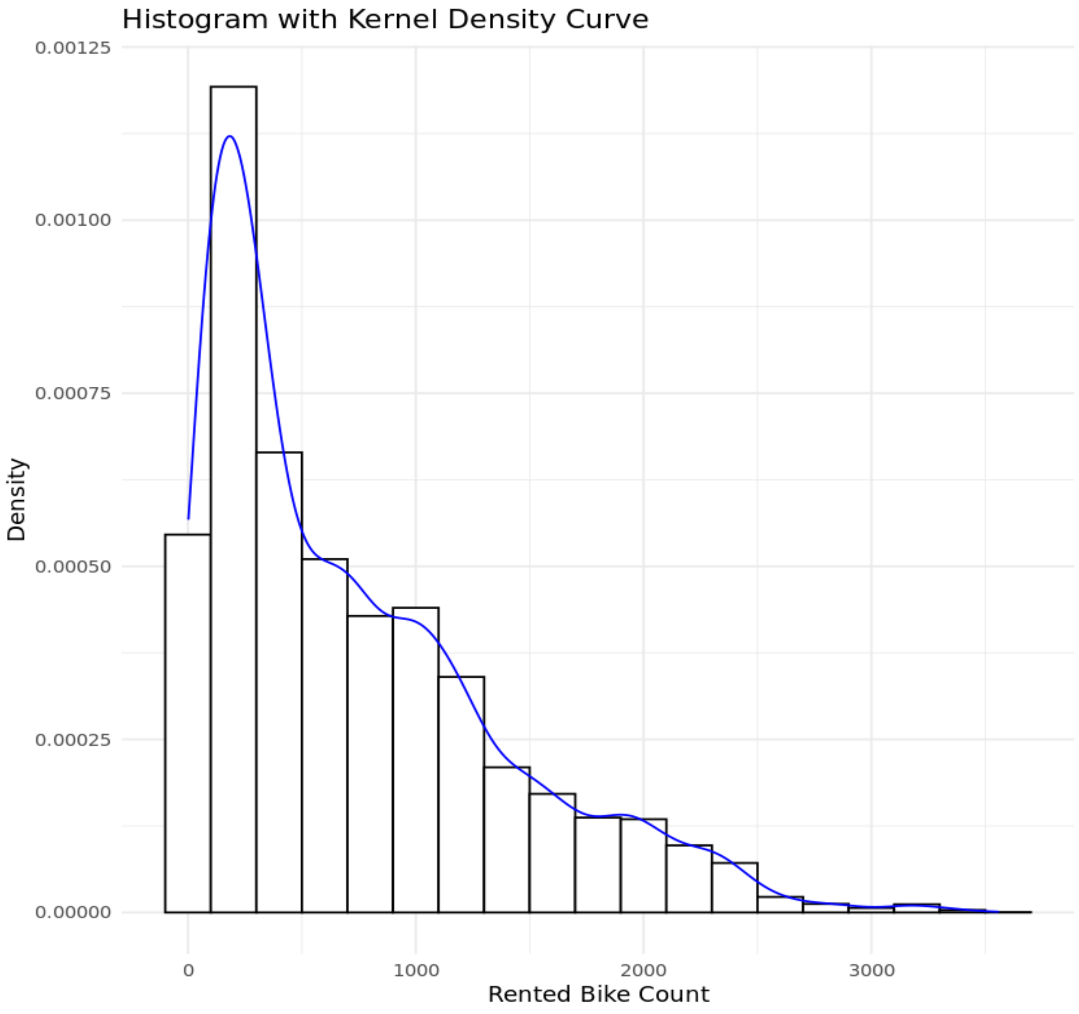
Histogram Analysis

•Rental counts cluster around mode (~250 bikes).

•Additional peaks at ~700, 900, 1900, 3200 bikes suggest hidden modes.

•Distribution's tail shows rare but significant spikes in demand.

This insight underscores the importance of understanding both the typical rental patterns and the potential for unexpected surges in bike-sharing demand.



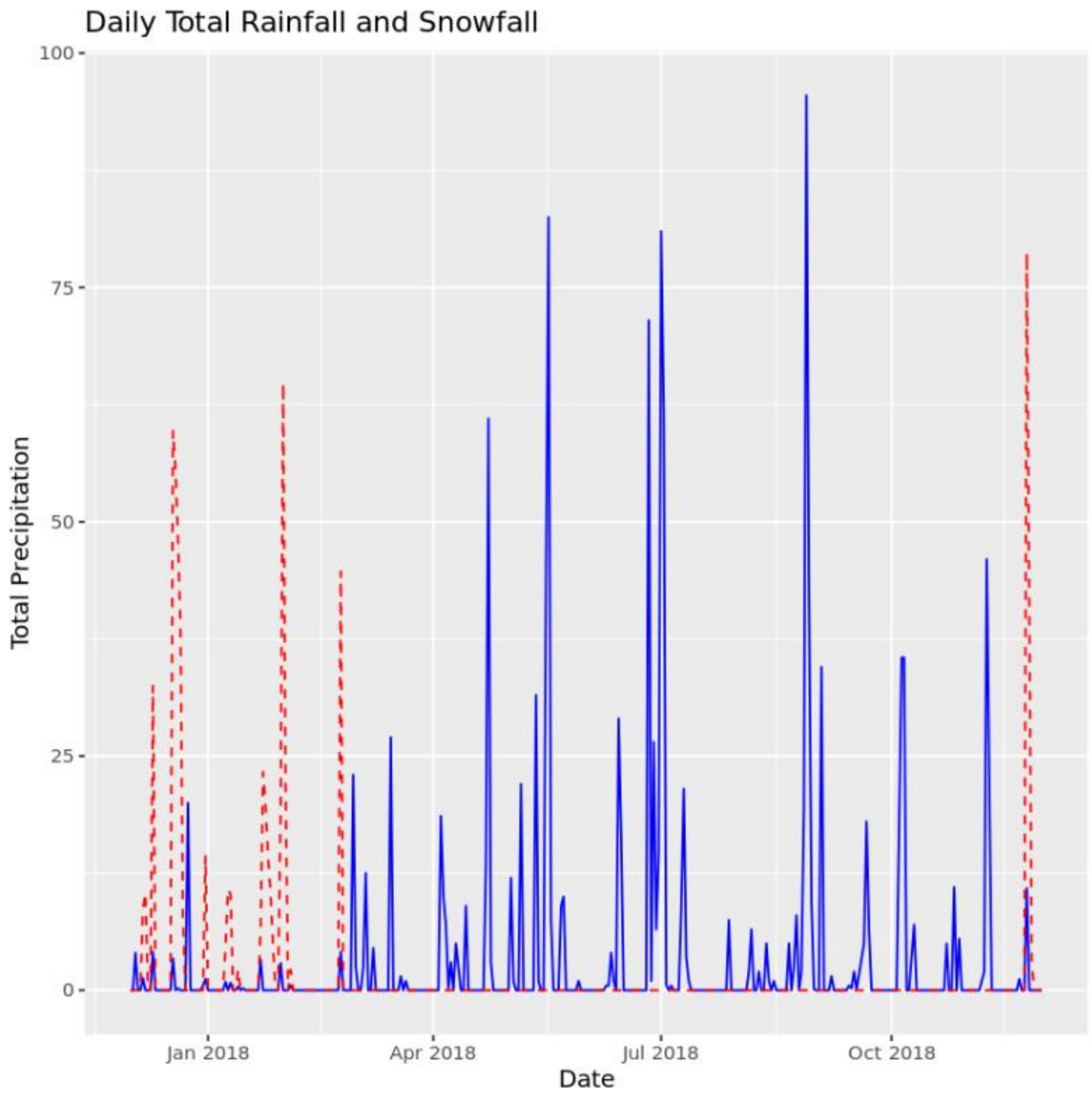
Daily total rainfall and snowfall

Explanation:

This graph compares snowfall and rainfall over a four-month period. The red line represents snowfall, while the blue line shows rainfall.

Key takeaways:

- Seasonal Trends: Snowfall is more common in the colder months [point to January/April peaks].
- Precipitation Types: We see a clear variation in precipitation types throughout the timeframe.



Predictive analysis

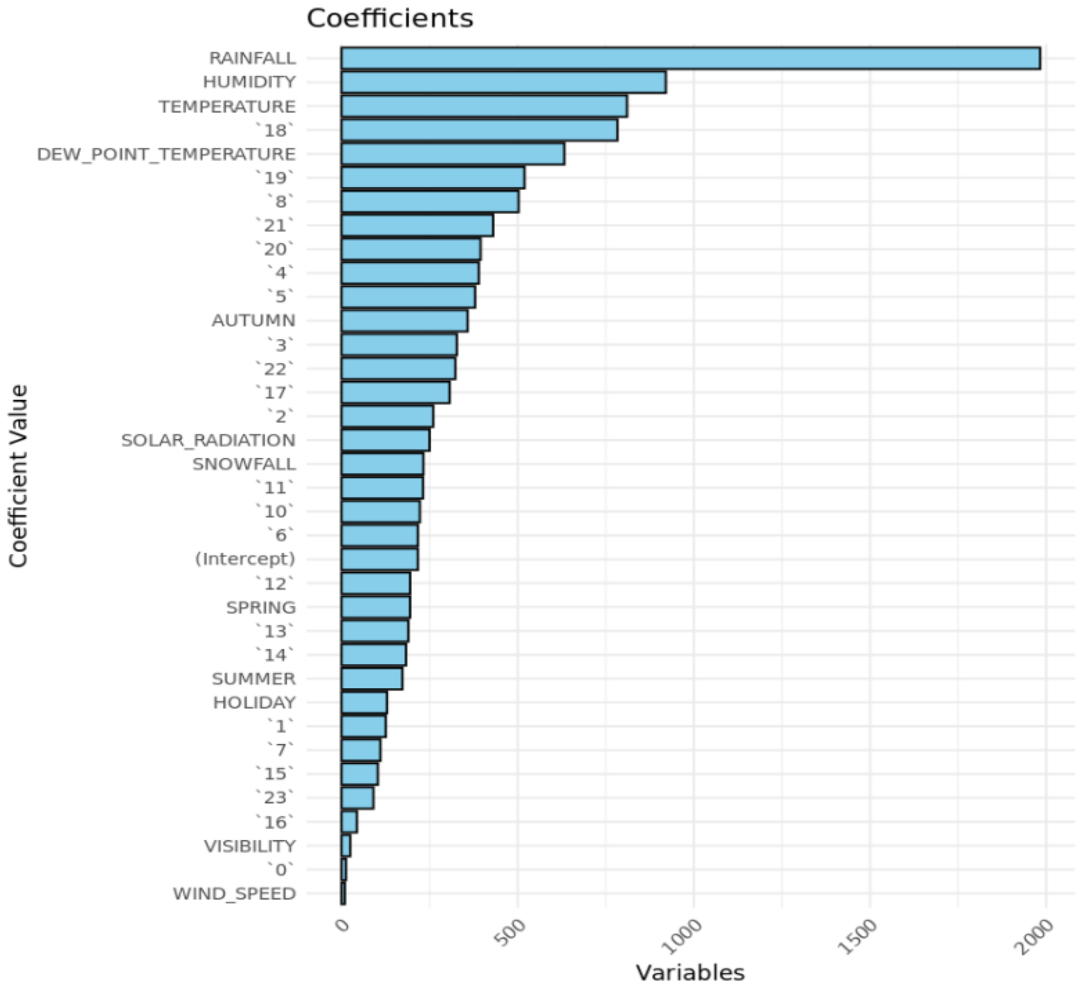
Statistical algorithms and machine learning techniques to analyze data and make predictions about future outcomes or trends based on historical data.

Ranked coefficients

Understanding Bike-Sharing Demand Dynamics

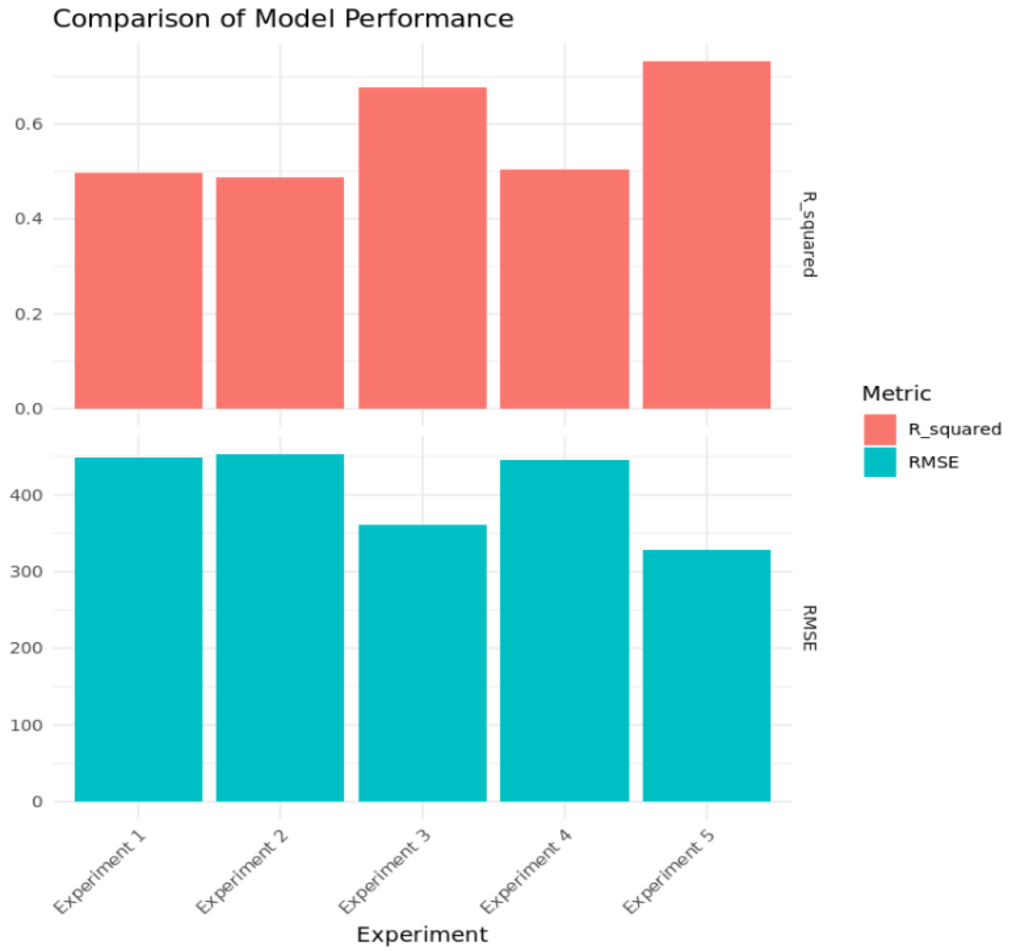
Key Points:

- *Weather Variables*: Temperature, humidity, rainfall, and dew point temperature demonstrate strong coefficients, underscoring weather's pivotal role. Favorable conditions drive usage, while adverse weather discourages ridership.
- *Wind and Solar Impact*: Wind speed significantly affects ridership, with strong gusts negatively impacting demand. Solar radiation, though less pronounced, correlates positively, indicating a preference for sunny weather.
- *Data Scope:* While this model captures essential demand drivers, it's vital to recognize limitations. Factors like day of the week, infrastructure, and events likely influence usage but are not accounted for in this dataset.



Model evaluation

The analysis comprises five experiments, each representing a different model variation incorporating polynomial terms, interaction terms, and regularizations. These experiments aim to predict bike-sharing demand based on weather and seasonal factors. The last model, Experiment 5, outperforms the others in terms of predictive accuracy, as indicated by higher R-squared values and lower root mean square error (RMSE).



Find the best performing model

Formula Overview:

- Predictive Model Formula: Experiment 5

  RENTED_BIKE_COUNT ~ poly(TEMPERATURE,3) + poly(HUMIDITY,2) + TEMPERATURE * HUMIDITY + SPRING * (WIND_SPEED + VISIBILITY) + SUMMER * DEW_POINT_TEMPERATURE + WINTER * SOLAR_RADIATION + WIND_SPEED + VISIBILITY + DEW_POINT_TEMPERATURE + SOLAR_RADIATION + RAINFALL + SNOWFALL + AUTUMN + SPRING + SUMMER + WINTER + HOLIDAY + NO_HOLIDAY + 0H + 1H + 2H + 3H + 4H + 5H + 6H + 7H + 8H + 9H + 10H + 11H + 12H + 13H + 14H + 15H + 16H + 17H + 18H + 19H + 20H + 21H + 22H + 23H

RMSE: 327.5851

R-squared: 0.7332623

Comparison with Other Models:

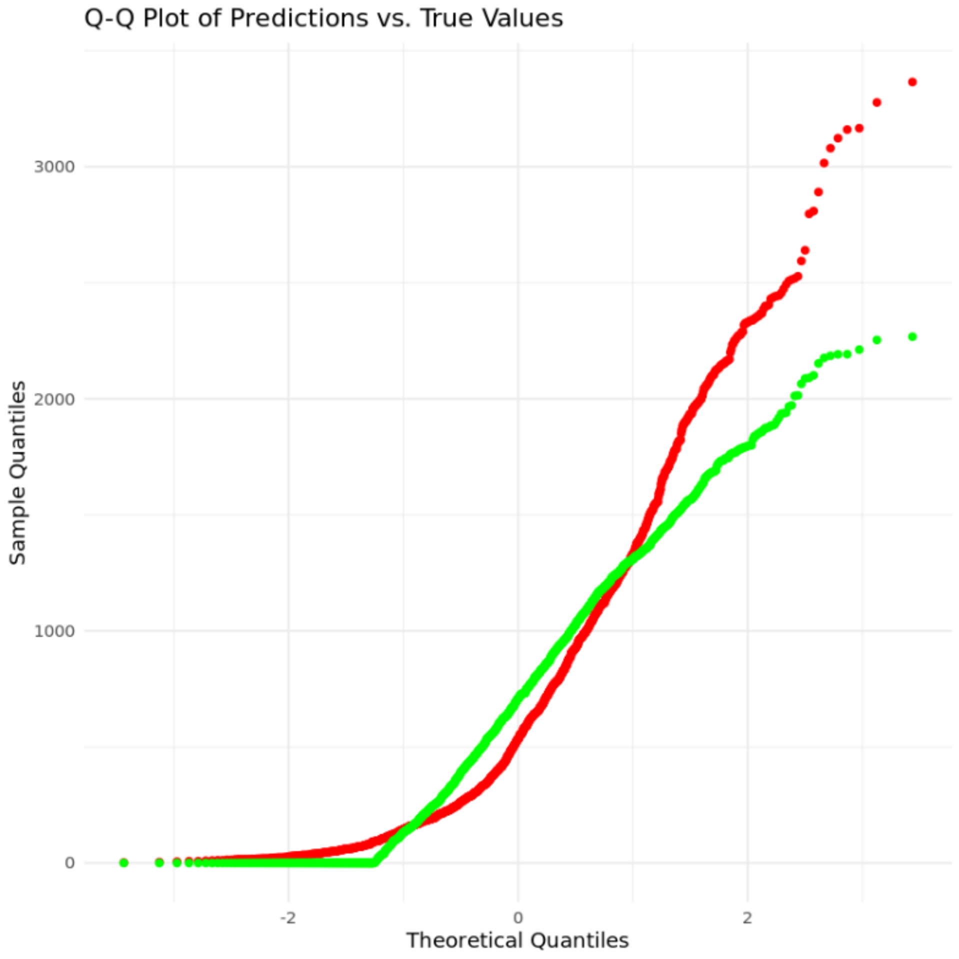
- Evaluated five models
- This model outperforms others
- Best balance between RMSE and R-squared

Addressing Model Complexity:

- Used regularization techniques to address overfitting.
- Utilized glmnet engine for Lasso, Ridge, and Elastic Net regularization.

Q-Q plot of the best model

The Q-Q plot of the best (Experiment 5) model's test results vs the truths



Dashboard

Visual interface offering interactive data insights for informed decision-making.

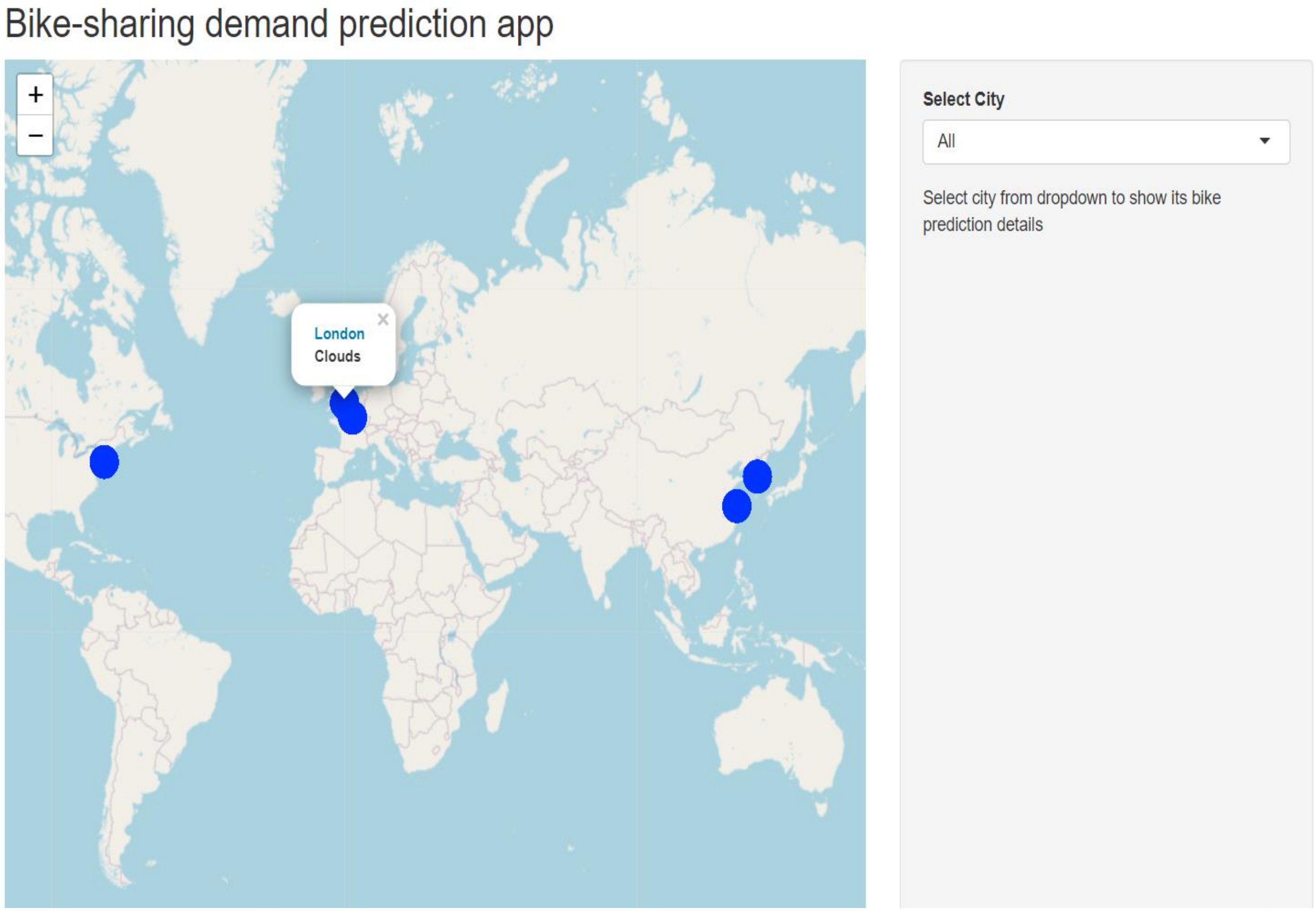
Shiny Dashboard: Introduction

- **Brief Overview:**
  - Shiny dashboard is an interactive visualization tool developed for analyzing bike-sharing demand.
  - Designed to provide users with intuitive access to weather forecasts and bike rental predictions.
- **Purpose**:
  - Enhances data exploration and decision-making for urban planners, bike-sharing operators, and other stakeholders.
  - Offers a user-friendly interface for accessing and interpreting complex data insights.

Dashboard Example 1

Interactive Dashboard Overview:

- **Left**: World map displaying five cities with interactive dots. Clicking on a dot reveals a pop-up with the city name and current weather update, such as rain, snow, or cloudy conditions (e.g., "London, Clouds").
- **Right**: Dropdown menu for selecting options (e.g., "All").



Bike-sharing demand prediction app

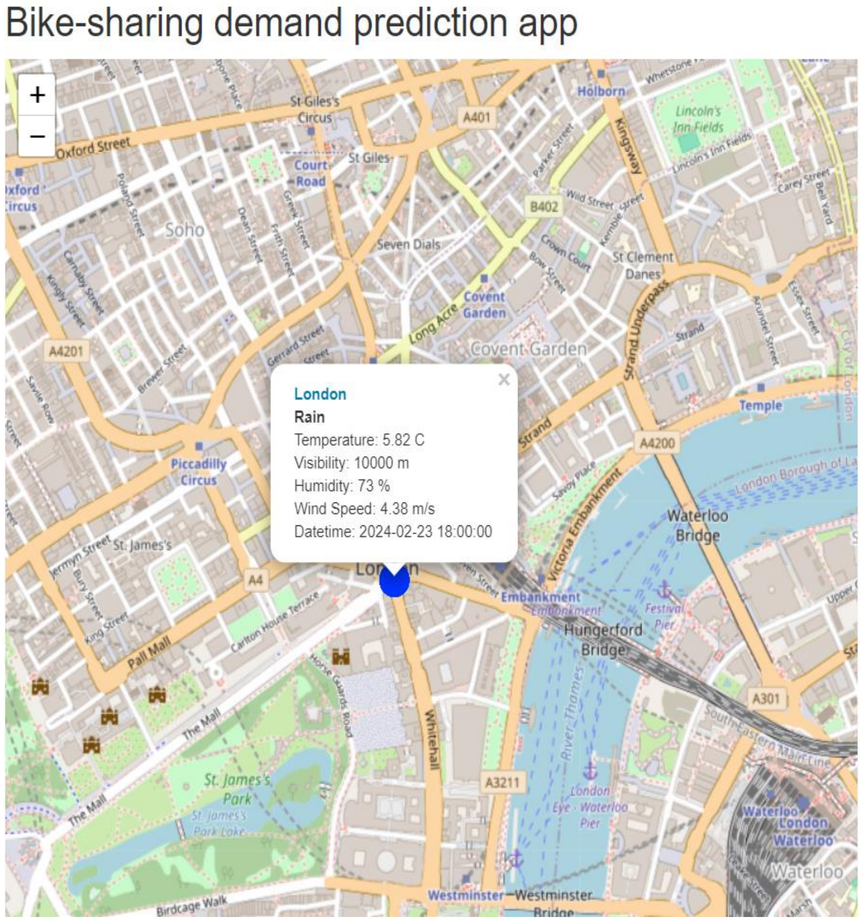
Select City: All

Select city from dropdown to show its bike prediction details

Dashboard Example 2

Interactive Dashboard Overview:

- **Left**: London map displaying real-time weather information linked with dates.
- **Right**: Three interactive graphs alongside a dropdown menu (selecting London).
  1. Temperature vs. Time: Line plot with points indicating temperature trends.
  2. Bike Sharing Prediction vs. Time: Line plot displaying bike rental predictions over time, with interactive points showing detailed predictions.
  3. Bike Sharing Prediction vs. Humidity: Scatter plot with a polynomial curve representing the correlation between bike rental predictions and humidity.



Bike-sharing demand prediction app

London

Rain

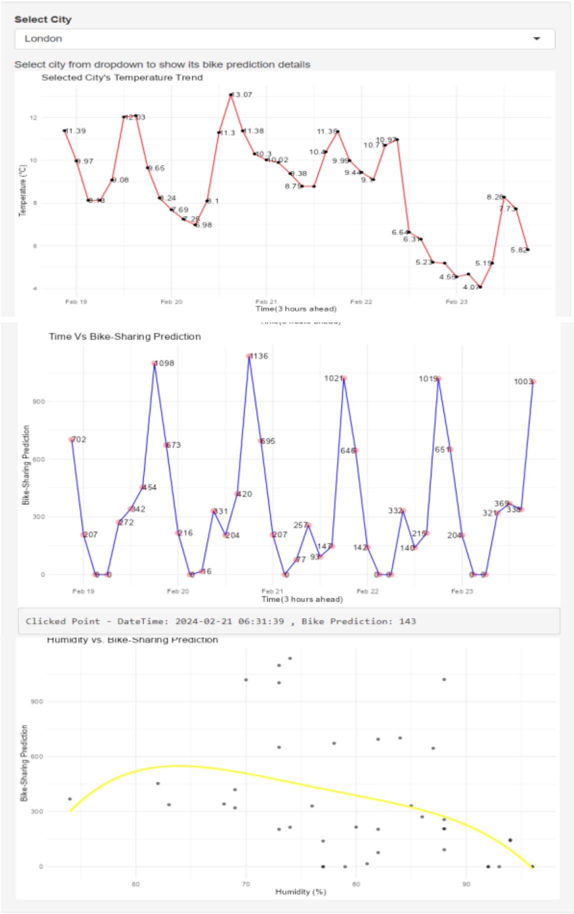
Temperature: 5.82 C

Visibility: 10000 m

Humidity: 73 %

Wind Speed: 4.38 m/s

Datetime: 2024-02-23 18:00:00



Select City

London

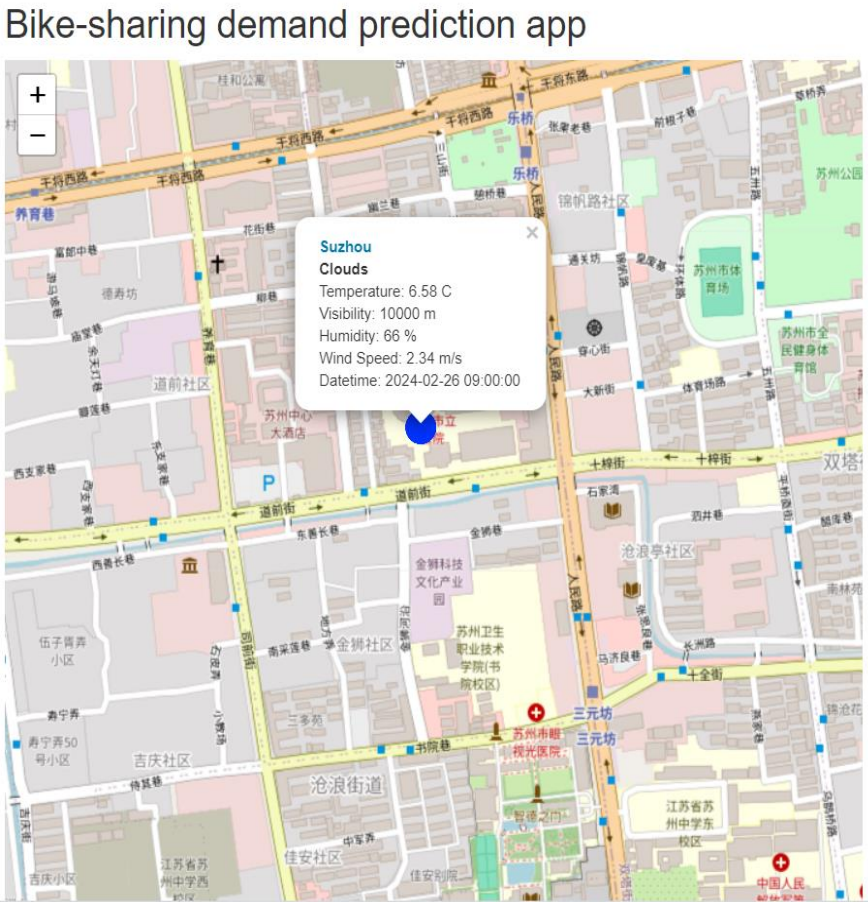
Select city from dropdown to show its bike prediction details

Clicked Point - DateTime: 2024-02-21 06:31:39 , Bike Prediction: 143

Dashboard Example 3

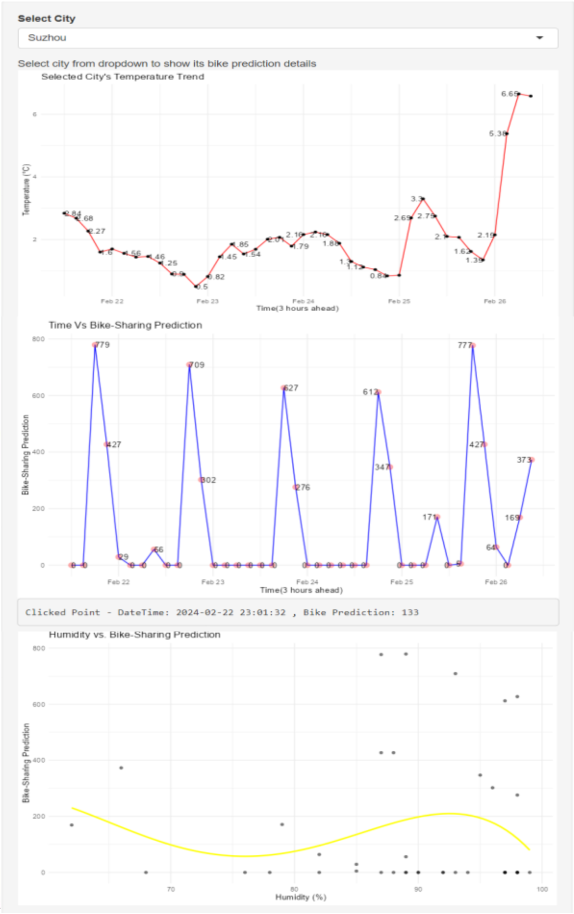
Interactive Dashboard Overview:

- **Left**: Suzhou map displaying real-time weather information linked with dates.
- **Right**: Three interactive graphs alongside a dropdown menu (selecting Suzhou).
  1. Temperature vs. Time: Line plot with points indicating temperature trends.
  2. Bike Sharing Prediction vs. Time: Line plot displaying bike rental predictions over time, with interactive points showing detailed predictions.
  3. Bike Sharing Prediction vs. Humidity: Scatter plot with a polynomial curve representing the correlation between bike rental predictions and humidity.



Bike-sharing demand prediction app

Suzhou
Clouds
Temperature: 6.58 C
Visibility: 10000 m
Humidity: 66 %
Wind Speed: 2.34 m/s
Datetime: 2024-02-26 09:00:00



Select City: Suzhou

Select city from dropdown to show its bike prediction details

Clicked Point - DateTime: 2024-02-22 23:01:32 , Bike Prediction: 133

Shiny Dashboard: Interactive Features

- **Map Visualization:**
  - Interactive map displaying bike-sharing demand across urban areas.
  - Users can hover over locations to view rental counts and weather conditions.
- **Filter Options:**
  - Filters for selecting specific time periods, weather parameters, and geographic regions.
  - Enables users to customize data views based on their preferences.
- **Prediction Charts:**
  - Dynamic charts illustrating predicted bike-sharing demand trends over time.
  - Users can toggle between different variables to explore their impact on rental counts.

Shiny Dashboard: Benefits

- **Enhanced Data Exploration:**
  - Provides a visually appealing platform for exploring complex datasets.
  - Allows users to interactively analyze correlations between weather factors and bike rental demand.
- **Improved Decision-Making:**
  - Empowers stakeholders to make informed decisions based on real-time data insights.
  - Facilitates proactive planning and resource allocation to optimize bike-sharing operations.
- **User-Friendly Interface:**
  - Intuitive design makes it easy for users to navigate and interpret data.
  - Promotes user engagement and encourages data-driven actions for urban planning and bike-sharing management.

DISCUSSION

Key Findings:

- Weather conditions significantly impact bike-sharing demand.
- Temperature and humidity are strong predictors of rental counts.
- Weekdays exhibit higher demand compared to weekends.
- Seasonal variations influence rental patterns.
- Regression models achieve high accuracy in predicting demand.

DISCUSSION

Data Trends:

- Increase in bike-sharing demand during peak hours.
- Variation in demand based on weather conditions and seasons.
- Correlation between temperature, humidity, and rental counts.
- Notable differences in demand between weekdays and weekends.

DISCUSSION

Implications:

- Urban planners can use weather forecasts to optimize bike availability.
- Bike-sharing operators can adjust resources based on demand patterns.
- Enhancing infrastructure during peak hours can improve user experience.
- Targeted marketing campaigns can be developed to capitalize on seasonal variations.

DISCUSSION

1. **Key Insights:**
    1. Weather conditions strongly influence bike-sharing demand, with temperature and humidity being significant predictors.
    2. Weekday demand differs from weekends, suggesting varying usage patterns.
    3. The Shiny dashboard provides valuable insights for urban planners and bike-sharing operators to optimize resources.

DISCUSSION

2.Practical Implications:

1. Urban planners can use weather forecasts to strategically allocate bike resources and enhance user experience.
2. Bike-sharing operators can adjust operations based on demand fluctuations, improving service efficiency.

3.Limitations:

1. While the analysis provides valuable insights, it's important to acknowledge data limitations and potential biases.
2. Future research could explore additional factors impacting bike-sharing demand for a more comprehensive understanding.

CONCLUSION

1. **Key Takeaways:**
    1. Weather significantly influences bike-sharing demand, with temperature and humidity being key predictors.
    2. Weekday and weekend usage patterns vary, highlighting the need for tailored resource allocation strategies.
    3. The Shiny dashboard offers a valuable tool for visualizing weather forecasts and bike-sharing predictions, enabling data-driven decision-making.
2. **Importance of Data Analysis:**
    1. Understanding weather impacts on bike-sharing demand is crucial for optimizing urban transportation systems.
    2. Data-driven approaches, supported by interactive visualization tools, empower stakeholders to make informed decisions and enhance operational efficiency.

CONCLUSION

3.Future Directions:

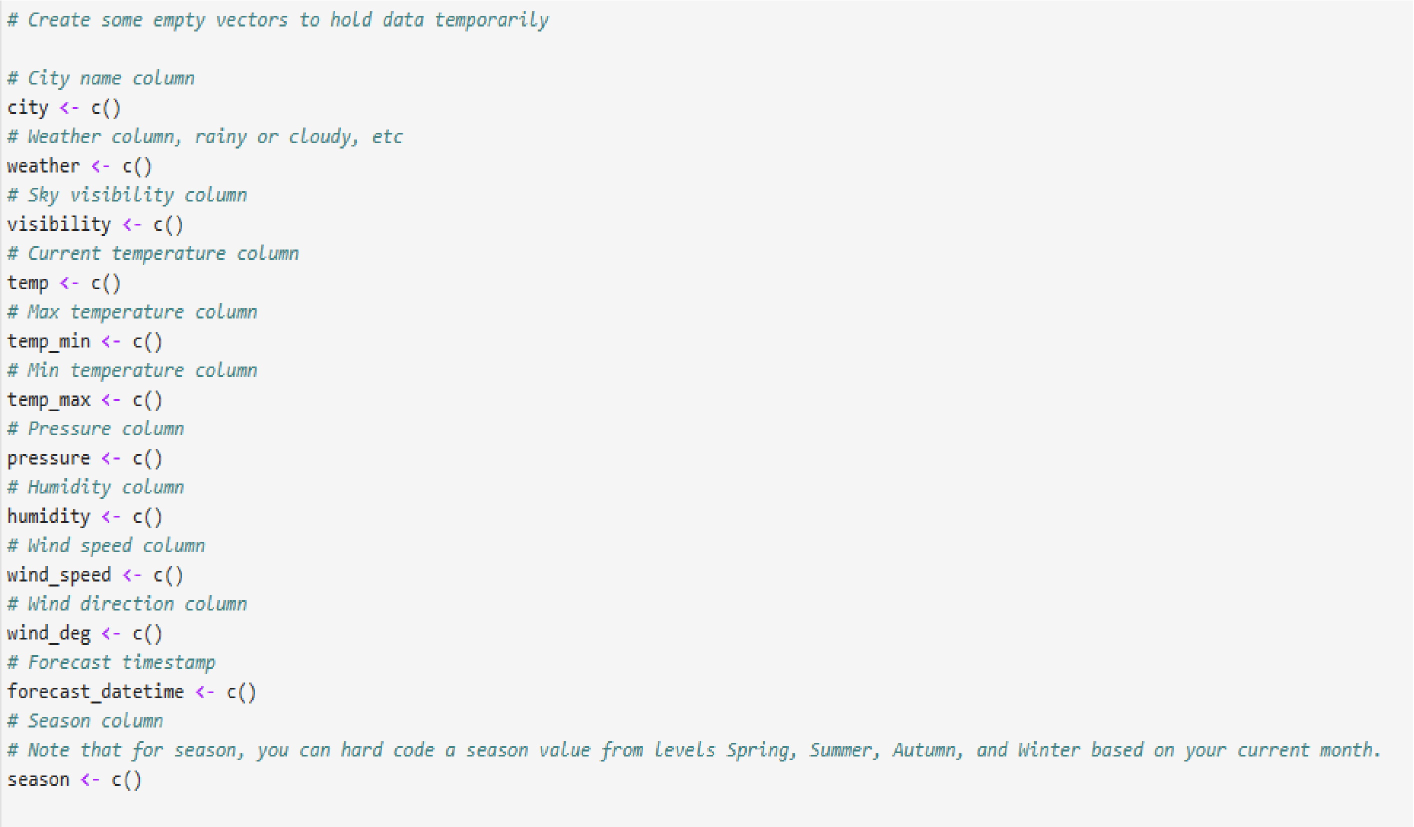
1. Further research is needed to explore additional factors affecting bike-sharing demand and to address any data limitations.
2. Continued development of interactive tools can drive innovation in urban mobility planning and improve user experience.

4.Thank You:

1. Thank you for your attention and engagement. Your insights and feedback are valuable for advancing research in urban transportation and bike-sharing management.

APPENDIX

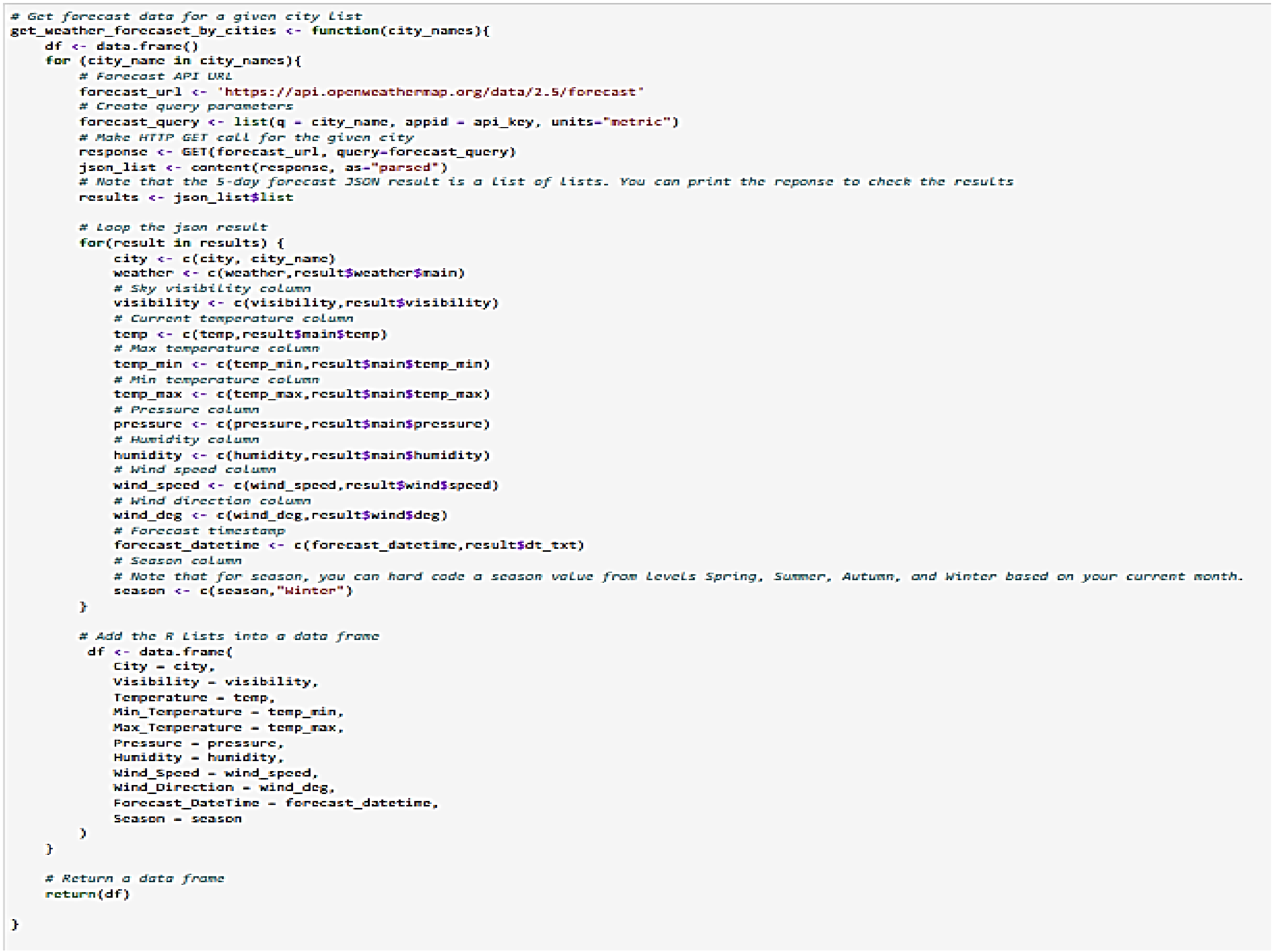
Screenshots of Notebook code cell and cell output used for OpenWeatherAPI and Webscrping



```
# Create some empty vectors to hold data temporarily

# City name column
city <- c()
# Weather column, rainy or cloudy, etc
weather <- c()
# Sky visibility column
visibility <- c()
# Current temperature column
temp <- c()
# Max temperature column
temp_min <- c()
# Min temperature column
temp_max <- c()
# Pressure column
pressure <- c()
# Humidity column
humidity <- c()
# Wind speed column
wind_speed <- c()
# Wind direction column
wind_deg <- c()
# Forecast timestamp
forecast_datetime <- c()
# Season column
# Note that for season, you can hard code a season value from levels Spring, Summer, Autumn, and Winter based on your current month.
season <- c()
```

APPENDIX

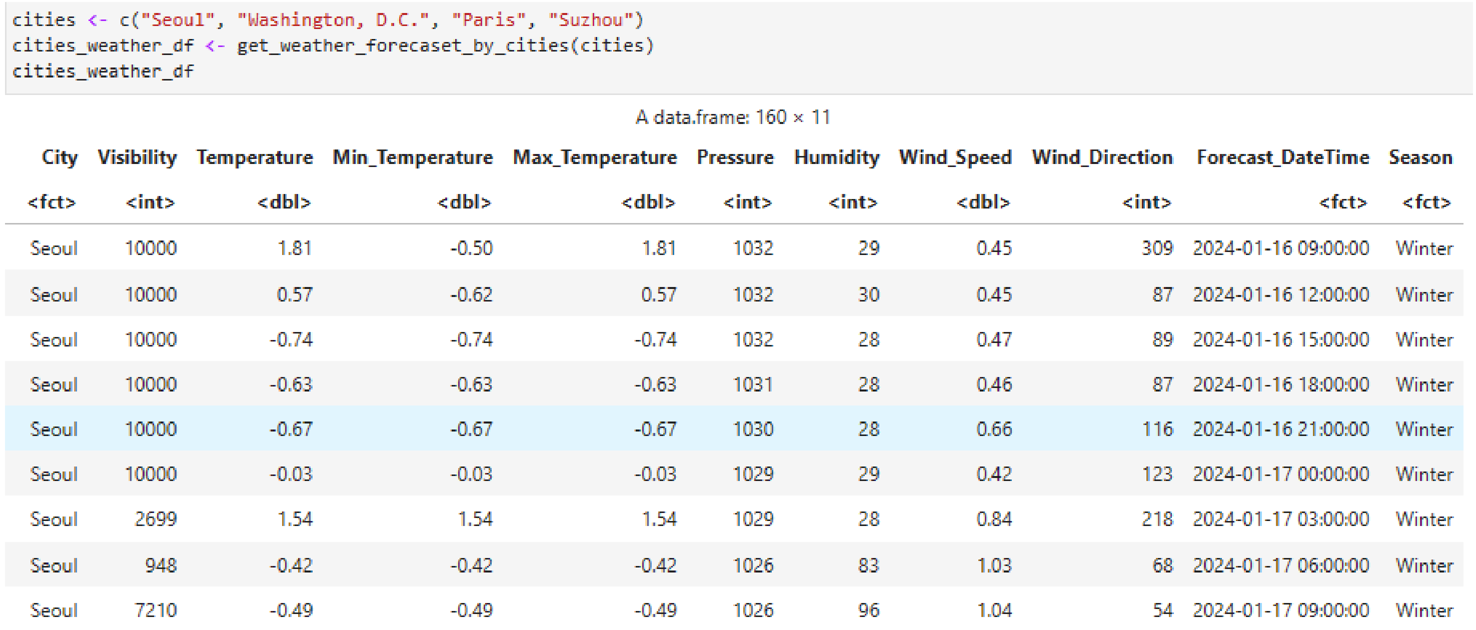
Screenshots of Notebook code cell and cell output used for OpenWeatherAPI and Webscrping



```
# Get forecast data for a given city list
get_weather_forecaset_by_cities <- function(city_names){
    df <- data.frame()
    for (city_name in city_names){
        # Forecast API URL
        forecast_url <- 'https://api.openweathermap.org/data/2.5/forecast'
        # Create query parameters
        forecast_query <- list(q = city_name, appid = api_key, units="metric")
        # Make HTTP GET call for the given city
        response <- GET(forecast_url, query=forecast_query)
        json_list <- content(response, as="parsed")
        # Note that the 5-day forecast JSON result is a list of lists. You can print the reponse to check the results
        results <- json_list$list

        # Loop the json result
        for(result in results) {
            city <- c(city, city_name)
            weather <- c(weather,result$weather$main)
            # Sky visibility column
            visibility <- c(visibility,result$visibility)
            # Current temperature column
            temp <- c(temp,result$main$temp)
            # Max temperature column
            temp_min <- c(temp_min,result$main$temp_min)
            # Min temperature column
            temp_max <- c(temp_max,result$main$temp_max)
            # Pressure column
            pressure <- c(pressure,result$main$pressure)
            # Humidity column
            humidity <- c(humidity,result$main$humidity)
            # Wind speed column
            wind_speed <- c(wind_speed,result$wind$speed)
            # Wind direction column
            wind_deg <- c(wind_deg,result$wind$deg)
            # Forecast timestamp
            forecast_datetime <- c(forecast_datetime,result$dt_txt)
            # Season column
            # Note that for season, you can hard code a season value from levels Spring, Summer, Autumn, and Winter based on your current month.
            season <- c(season,"Winter")
        }

        # Add the R lists into a data frame
         df <- data.frame(
            City = city,
            Visibility = visibility,
            Temperature = temp,
            Min_Temperature = temp_min,
            Max_Temperature = temp_max,
            Pressure = pressure,
            Humidity = humidity,
            Wind_Speed = wind_speed,
            Wind_Direction = wind_deg,
            Forecast_DateTime = forecast_datetime,
            Season = season
        )
    }

    # Return a data frame
    return(df)

}
```

APPENDIX

Screenshots of Notebook code cell and cell output used for OpenWeatherAPI and Webscrping



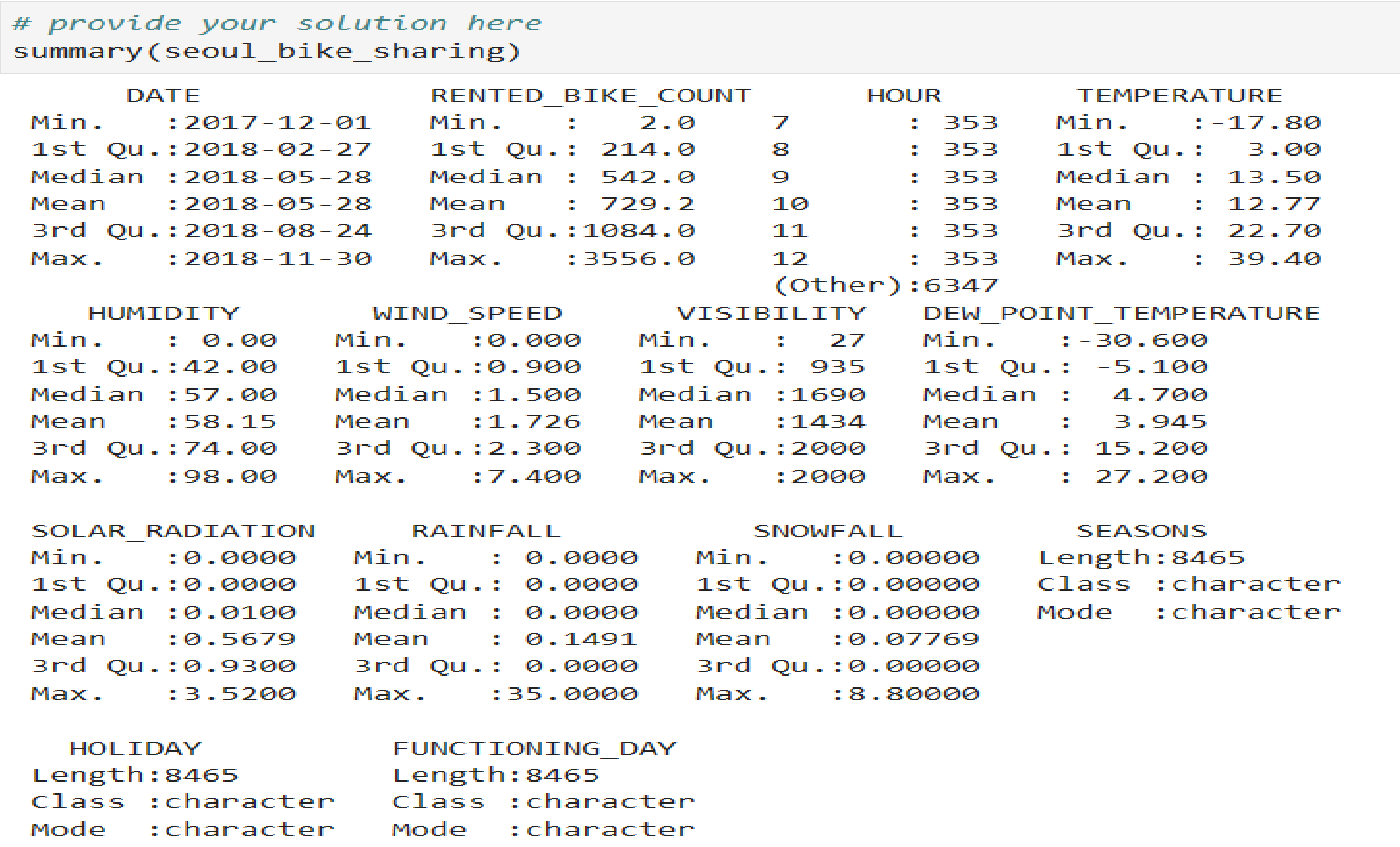
```
cities <- c("Seoul", "Washington, D.C.", "Paris", "Suzhou")
cities_weather_df <- get_weather_forecaset_by_cities(cities)
cities_weather_df
```

A data.frame: 160 × 11

City	Visibility	Temperature	Min_Temperature	Max_Temperature	Pressure	Humidity	Wind_Speed	Wind_Direction	Forecast_DateTime	Season
<fct>	<int>	<dbl>	<dbl>	<dbl>	<int>	<int>	<dbl>	<int>	<fct>	<fct>
Seoul	10000	1.81	-0.50	1.81	1032	29	0.45	309	2024-01-16 09:00:00	Winter
Seoul	10000	0.57	-0.62	0.57	1032	30	0.45	87	2024-01-16 12:00:00	Winter
Seoul	10000	-0.74	-0.74	-0.74	1032	28	0.47	89	2024-01-16 15:00:00	Winter
Seoul	10000	-0.63	-0.63	-0.63	1031	28	0.46	87	2024-01-16 18:00:00	Winter
Seoul	10000	-0.67	-0.67	-0.67	1030	28	0.66	116	2024-01-16 21:00:00	Winter
Seoul	10000	-0.03	-0.03	-0.03	1029	29	0.42	123	2024-01-17 00:00:00	Winter
Seoul	2699	1.54	1.54	1.54	1029	28	0.84	218	2024-01-17 03:00:00	Winter
Seoul	948	-0.42	-0.42	-0.42	1026	83	1.03	68	2024-01-17 06:00:00	Winter
Seoul	7210	-0.49	-0.49	-0.49	1026	96	1.04	54	2024-01-17 09:00:00	Winter

APPENDIX

SQL Code Snippets:

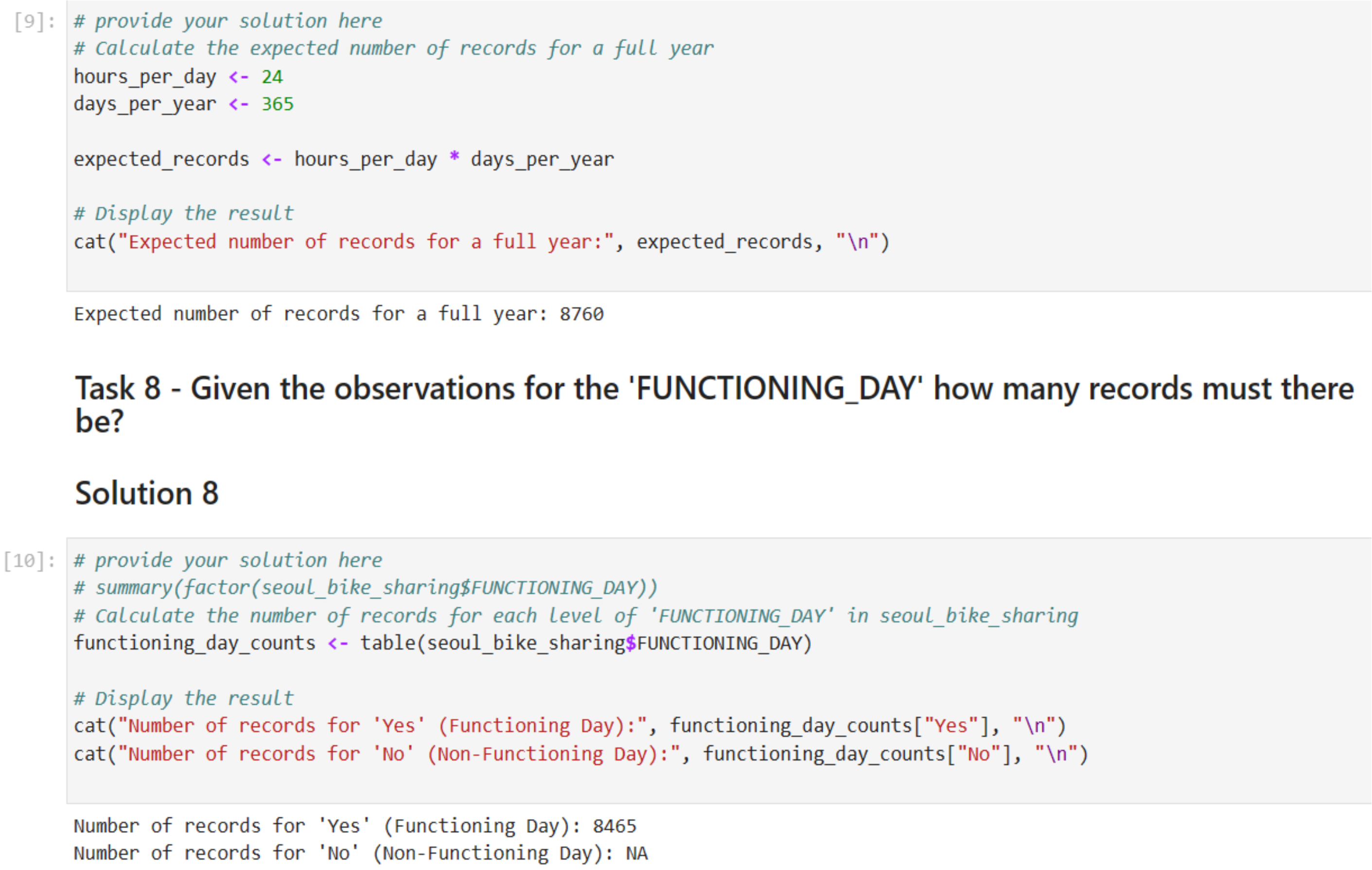


```
# provide your solution here
summary(seoul_bike_sharing)
```

```
      DATE               RENTED_BIKE_COUNT        HOUR         TEMPERATURE
 Min.   :2017-12-01   Min.   :   2.0     7       : 353   Min.   :-17.80
 1st Qu.:2018-02-27   1st Qu.: 214.0     8       : 353   1st Qu.:  3.00
 Median :2018-05-28   Median : 542.0     9       : 353   Median : 13.50
 Mean   :2018-05-28   Mean   : 729.2     10      : 353   Mean   : 12.77
 3rd Qu.:2018-08-24   3rd Qu.:1084.0     11      : 353   3rd Qu.: 22.70
 Max.   :2018-11-30   Max.   :3556.0     12      : 353   Max.   : 39.40
                                         (Other):6347
    HUMIDITY         WIND_SPEED        VISIBILITY     DEW_POINT_TEMPERATURE
 Min.   : 0.00   Min.   :0.000   Min.   :  27   Min.   :-30.600
 1st Qu.:42.00   1st Qu.:0.900   1st Qu.: 935   1st Qu.: -5.100
 Median :57.00   Median :1.500   Median :1690   Median :  4.700
 Mean   :58.15   Mean   :1.726   Mean   :1434   Mean   :  3.945
 3rd Qu.:74.00   3rd Qu.:2.300   3rd Qu.:2000   3rd Qu.: 15.200
 Max.   :98.00   Max.   :7.400   Max.   :2000   Max.   : 27.200

 SOLAR_RADIATION      RAINFALL          SNOWFALL          SEASONS
 Min.   :0.0000   Min.   : 0.0000   Min.   :0.00000   Length:8465
 1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.:0.00000   Class :character
 Median :0.0100   Median : 0.0000   Median :0.00000   Mode  :character
 Mean   :0.5679   Mean   : 0.1491   Mean   :0.07769
 3rd Qu.:0.9300   3rd Qu.: 0.0000   3rd Qu.:0.00000
 Max.   :3.5200   Max.   :35.0000   Max.   :8.80000

   HOLIDAY          FUNCTIONING_DAY
 Length:8465        Length:8465
 Class :character   Class :character
 Mode  :character   Mode  :character
```

APPENDIX

SQL Code Snippets:



```
[9]: # provide your solution here
     # Calculate the expected number of records for a full year
     hours_per_day <- 24
     days_per_year <- 365

     expected_records <- hours_per_day * days_per_year

     # Display the result
     cat("Expected number of records for a full year:", expected_records, "\n")
```

```
Expected number of records for a full year: 8760
```

Task 8 - Given the observations for the 'FUNCTIONING_DAY' how many records must there be?

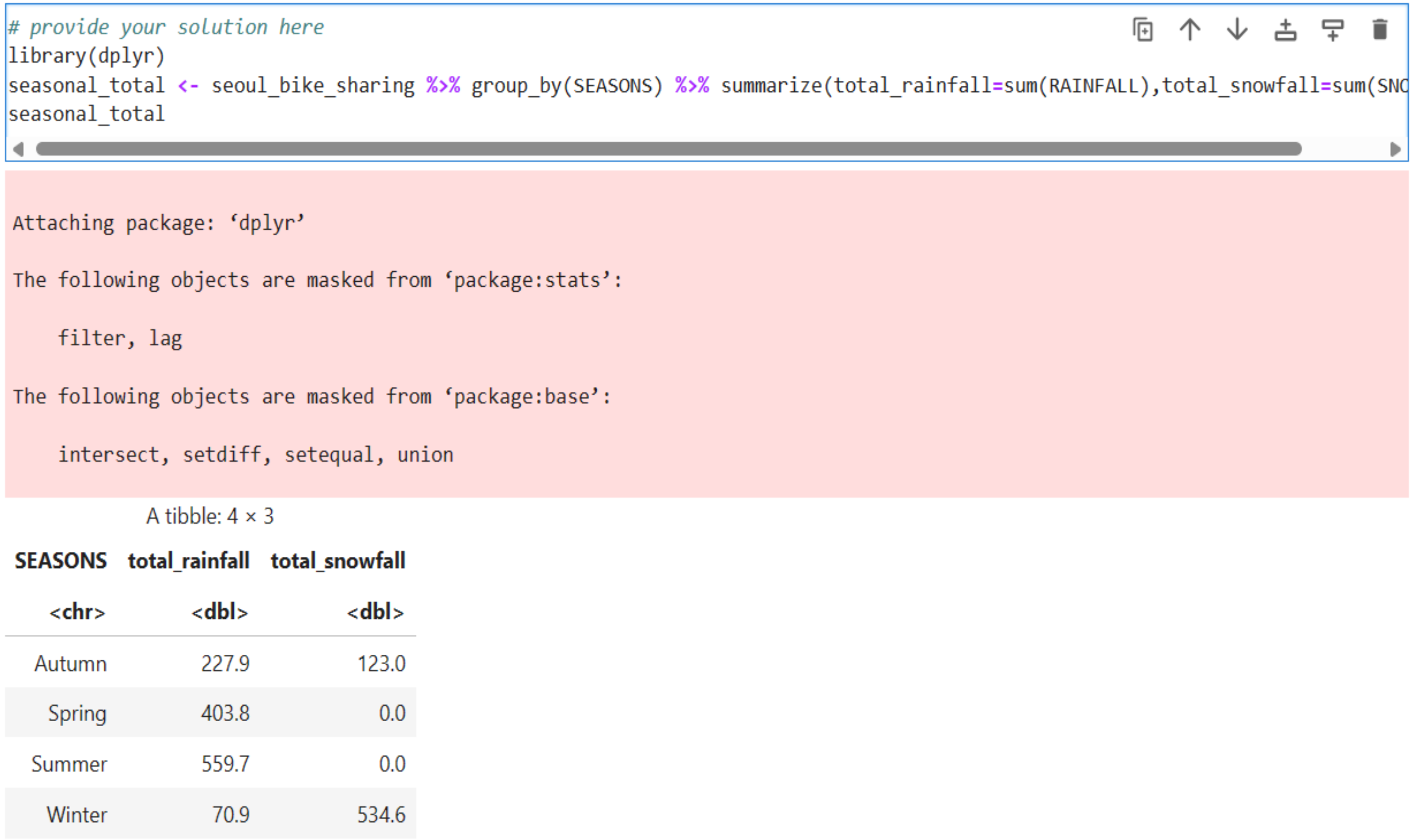
Solution 8

```
[10]: # provide your solution here
      # summary(factor(seoul_bike_sharing$FUNCTIONING_DAY))
      # Calculate the number of records for each level of 'FUNCTIONING_DAY' in seoul_bike_sharing
      functioning_day_counts <- table(seoul_bike_sharing$FUNCTIONING_DAY)

      # Display the result
      cat("Number of records for 'Yes' (Functioning Day):", functioning_day_counts["Yes"], "\n")
      cat("Number of records for 'No' (Non-Functioning Day):", functioning_day_counts["No"], "\n")
```

```
Number of records for 'Yes' (Functioning Day): 8465
Number of records for 'No' (Non-Functioning Day): NA
```

APPENDIX

SQL Code Snippets:



```
# provide your solution here
library(dplyr)
seasonal_total <- seoul_bike_sharing %>% group_by(SEASONS) %>% summarize(total_rainfall=sum(RAINFALL),total_snowfall=sum(SNO
seasonal_total
```

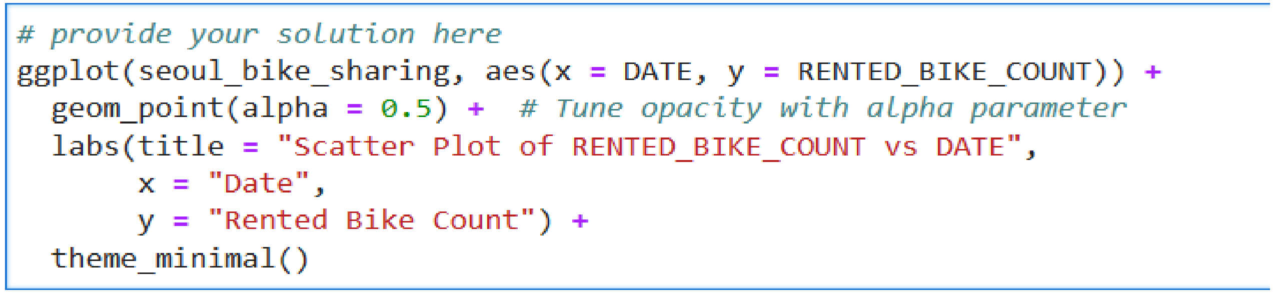
```
Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

A tibble: 4 × 3

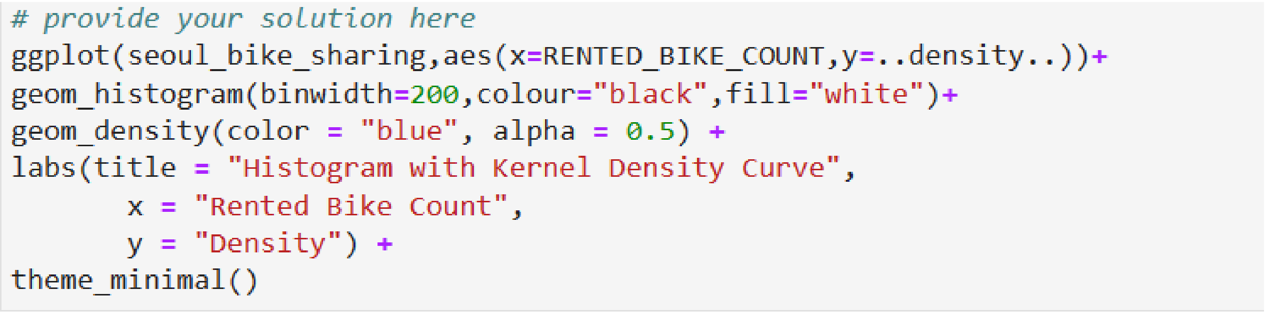
SEASONS	total_rainfall	total_snowfall
<chr>	<dbl>	<dbl>
Autumn	227.9	123.0
Spring	403.8	0.0
Summer	559.7	0.0
Winter	70.9	534.6

APPENDIX

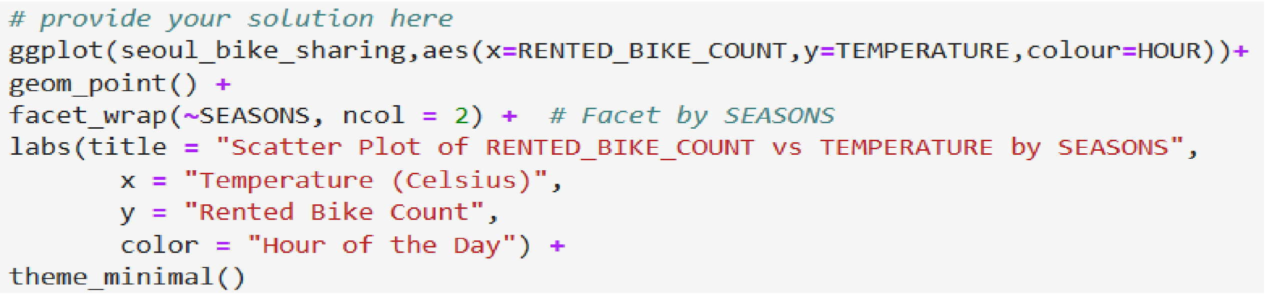
R(ggplot) Code Snippets:



```
# provide your solution here
ggplot(seoul_bike_sharing, aes(x = DATE, y = RENTED_BIKE_COUNT)) +
  geom_point(alpha = 0.5) +  # Tune opacity with alpha parameter
  labs(title = "Scatter Plot of RENTED_BIKE_COUNT vs DATE",
       x = "Date",
       y = "Rented Bike Count") +
  theme_minimal()
```



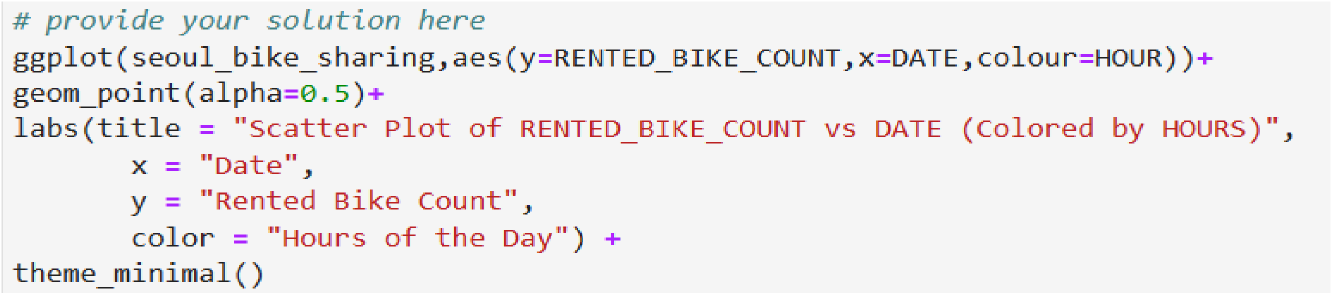
```
# provide your solution here
ggplot(seoul_bike_sharing,aes(x=RENTED_BIKE_COUNT,y=..density..))+
geom_histogram(binwidth=200,colour="black",fill="white")+
geom_density(color = "blue", alpha = 0.5) +
labs(title = "Histogram with Kernel Density Curve",
       x = "Rented Bike Count",
       y = "Density") +
theme_minimal()
```



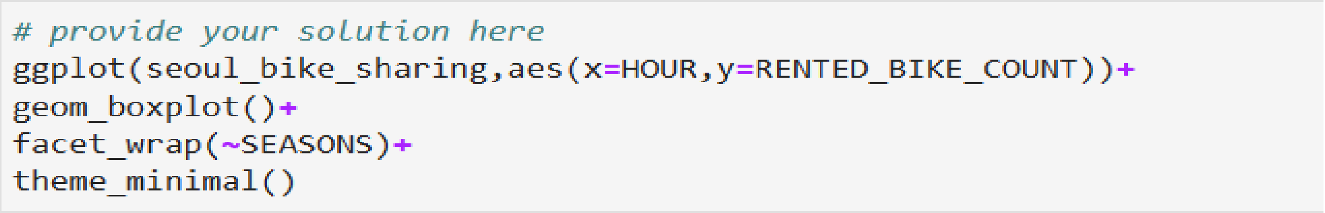
```
# provide your solution here
ggplot(seoul_bike_sharing,aes(x=RENTED_BIKE_COUNT,y=TEMPERATURE,colour=HOUR))+
geom_point() +
facet_wrap(~SEASONS, ncol = 2) +  # Facet by SEASONS
labs(title = "Scatter Plot of RENTED_BIKE_COUNT vs TEMPERATURE by SEASONS",
       x = "Temperature (Celsius)",
       y = "Rented Bike Count",
       color = "Hour of the Day") +
theme_minimal()
```

APPENDIX

R(ggplot) Code Snippets:



```r
# provide your solution here
ggplot(seoul_bike_sharing,aes(y=RENTED_BIKE_COUNT,x=DATE,colour=HOUR))+
geom_point(alpha=0.5)+
labs(title = "Scatter Plot of RENTED_BIKE_COUNT vs DATE (Colored by HOURS)",
       x = "Date",
       y = "Rented Bike Count",
       color = "Hours of the Day") +
theme_minimal()
```



```r
# provide your solution here
ggplot(seoul_bike_sharing,aes(x=HOUR,y=RENTED_BIKE_COUNT))+
geom_boxplot()+
facet_wrap(~SEASONS)+
theme_minimal()
```