IBM Applied Data Science with R Capstone Project

Predicting demand of rental-bikes based on weather data.

Arnav Bhatia

15th March 2024

OUTLINE

- Introduction
- Methodology
- Results
  - Visualizations (Charts)
  - Dashboard
- Findings
- Conclusion
- Appendix

INTRODUCTION

In response to the pressing need for sustainable transportation options amidst the challenge of global warming, bike-sharing systems have emerged as a vital solution. Cities worldwide have embraced rental bikes to enhance transportation accessibility, but maintaining an optimal bike supply remains a challenge due to highly variable demand based on environmental and seasonal factors. This project aims to develop a predictive model forecasting bike rental demand per hour based on current weather conditions. By leveraging environmental variables and factors like weekday and season, the model will facilitate supply optimization, ensuring a seamless bike-sharing experience for both managers and users.

METHODOLOGY

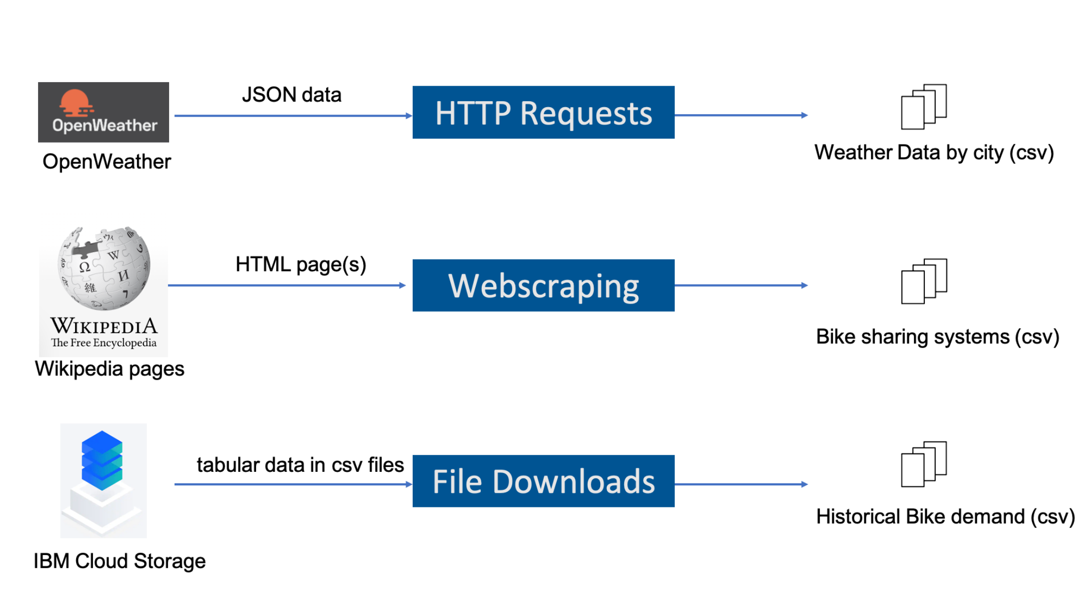
- ❖ Data Collection using:
  - APIs
  - Web Scraping
- ❖ Data Wrangling
  - Dealing with Duplicates and Missing Values
  - Normalizing Data
- ❖ Exploratory Analysis
  - SQL
  - Data Visualization
- ❖ Predictive Analysis using Regression Models
  - Building baseline model
  - Improving baseline model
- ❖ Building a R Shiny dashboard app

METHODOLOGY

Data Collection

Datasets used for the project:

- 5-day weather forecast obtained from OpenWeather API
- Global Bike Sharing Systems Dataset obtained from web scraping a Wikipedia page
- World Cities data, a csv file provided by IBM Cloud Storage
- Seoul Bike Sharing Demand Data Set, provided by IBM Cloud Storage



Data Wrangling

This stage was aimed at cleaning the data by checking for missing values, mis-formatted values and/or unexpected noises.

Initially, the process utilized the 'stringr' R package alongside regular expressions to streamline column names, eliminate extraneous reference links within tables, and extract numerical values from rows.

Subsequently, leveraging the 'dplyr' package, the focus shifted towards identifying and managing missing values within the dataset. Moreover, categorical variables underwent transformation into indicator (dummy) variables, while the data itself underwent normalization via min-max normalization techniques.

Exploratory Data Analysis with SQL

Exploratory Data Analysis was performed on the datasets using Structured Query Language (SQL). The questions were:

- Determine how many records are in the seoul_bike_sharing dataset.
- Determine how many hours had non-zero rented bike count.
- Query the weather forecast for Seoul over the next 3 hours.
- Find which seasons are included in the Seoul bike sharing dataset.
- Find the first and last dates in the Seoul Bike Sharing dataset.
- Determine which date and hour had the most bike rentals.
- Determine the average hourly temperature and the average number of bike rentals per hour over each season. List the top ten results by average bike count.
- Find the average hourly bike count during each season.
- Determine the average TEMPERATURE, HUMIDITY, WIND_SPEED, VISIBILITY, DEW_POINT_TEMPERATURE, SOLAR_RADIATION, RAINFALL, and SNOWFALL per season
- Determine the Total Bike Count and City Info for Seoul
- Find all city names and coordinates with comparable bike scale to Seoul's bike sharing system

EDA with Data Visualization

Exploratory Data Analysis was also performed to visually inspect the datasets using the ggplot2 library.

The ggplot2 library was used to;

- Create a scatter plot of `RENTED_BIKE_COUNT` vs `DATE`.
- Create the same plot of the `RENTED_BIKE_COUNT` time series, but now add `HOURS` as the colour.
- Create a histogram overlaid with a kernel density curve.
- Create a scatter plot to visualize the correlation between `RENTED_BIKE_COUNT` and `TEMPERATURE` by `SEASONS`.
- Create a display of four boxplots of `RENTED_BIKE_COUNT` vs. `HOUR` grouped by `SEASONS`.
- Create a line plot after grouping the data by `DATE`, and using the summarize() function to visualize the daily total rainfall and snowfall.

Predictive Analysis

At this stage, emphasis was placed on constructing and fine-tuning a regression model aimed at forecasting the hourly bike rental count, incorporating both weather and non-weather conditions.

The prepared and refined dataset was split into two distinct sets: the Train set, utilized for model development and refinement employing techniques such as polynomial regression, interaction, and regularization; and the Test set, employed for model evaluation. Evaluation metrics including RMSE and RSQ were employed, with the selection of the regression algorithm being guided by the attainment of the lowest RMSE and highest RSQ values observed within the Test data.

Building a dashboard with R Shiny

The results of the predictive linear regression model were combined with an interactive dashboard created using the shiny package in R. This dashboard contained:

- A basic max bike prediction overview map.
- A static temperature trend line.
- An interactive bike-sharing demand prediction trend line.
- A static humidity and bike-sharing demand prediction correlation plot

RESULTS

EDA with SQL - RESULTS

The forthcoming section will feature the RSQLite Query, its output, and the specific observations I made during the exploratory data analysis (EDA) process.

Task 1 – Busiest Bike Rental Times

```r
# Task 6 - Subquery - 'all-time high'
all_time_high <- dbGetQuery(conn, "SELECT DATE, HOUR, MAX(RENTED_BIKE_COUNT) AS max_rented_bike_count
                                   FROM SEOUL_BIKE_SHARING
                                   GROUP BY DATE, HOUR
                                   ORDER BY max_rented_bike_count DESC
                                   LIMIT 1")
```

RSQLite.R* | all_time_high | hourly_popularity_te

Filter

	DATE	HOUR	max_rented_bike_count
1	19/06/2018	18	3556

Observation: At 6pm on 19/06/2018, a total of 3556 bikes were rented

Task 2 – Hourly Popularity and Temperature by Seasons

```
# Task 7 - Hourly popularity and temperature by season
hourly_popularity_temp_season <- dbGetQuery(conn, "SELECT SEASONS, HOUR, AVG(RENTED_BIKE_COUNT) AS avg_rented_bike_count, AVG(TEMPERATURE) AS avg_temperature
                                            FROM SEOUL_BIKE_SHARING
                                            GROUP BY SEASONS, HOUR
                                            ORDER BY avg_rented_bike_count DESC, avg_temperature DESC
                                            LIMIT 10")
```

RSQLite.R* | all_time_high | hourly_popularity_temp_season

Filter

	SEASONS	HOUR	avg_rented_bike_count	avg_temperature
1	Summer	18	2135.141	29.38791
2	Autumn	18	1983.333	16.03185
3	Summer	19	1889.250	28.27378
4	Summer	20	1801.924	27.06630
5	Summer	21	1754.065	26.27826
6	Spring	18	1689.311	15.97222
7	Summer	22	1567.870	25.69891
8	Autumn	17	1562.877	17.27778
9	Summer	17	1526.293	30.07691
10	Autumn	19	1515.568	15.06346

Observation: The most common time period to rent bikes appears to be in the Summer during evening hours. (6pm onwards)

Task 3 – Rental Seasonality

```
# Task 8 - Rental Seasonality
rental_seasonality <- dbGetQuery(conn, "SELECT SEASONS, HOUR,
                                        AVG(RENTED_BIKE_COUNT) AS avg_rented_bike_count,
                                        MIN(RENTED_BIKE_COUNT) AS min_rented_bike_count,
                                        MAX(RENTED_BIKE_COUNT) AS max_rented_bike_count,
                                        SQRT(AVG(RENTED_BIKE_COUNT*RENTED_BIKE_COUNT) - AVG(RENTED_BIKE_COUNT)*AVG(RENTED_BIKE_COUNT)) AS std_dev_rented_bike_count
                                        FROM SEOUL_BIKE_SHARING
                                        GROUP BY SEASONS, HOUR")
```

RSQLite.R* | all_time_high | hourly_popularity_temp_season | rental_seasonality | weather_seasonality | total_bike_co

Filter

	SEASONS	HOUR	avg_rented_bike_count	min_rented_bike_count	max_rented_bike_count	std_dev_rented_bike_count
1	Autumn	0	709.43750	119	1336	219.14298
2	Autumn	1	552.50000	144	1001	191.54216
3	Autumn	2	377.47500	55	785	144.90134
4	Autumn	3	256.55000	28	514	102.53108
5	Autumn	4	169.02500	24	338	58.63957
6	Autumn	5	163.41250	24	264	53.88174
7	Autumn	6	359.48750	23	691	180.27049
8	Autumn	7	788.87654	5	1556	457.96861
9	Autumn	8	1345.03704	6	2391	758.35956
10	Autumn	9	848.43210	5	1322	334.52653
11	Autumn	10	715.27160	2	1113	252.56839
12	Autumn	11	802.95062	20	1284	302.57238
13	Autumn	12	934.64198	17	1634	347.69066
14	Autumn	13	1002.66667	18	1849	369.30997
15	Autumn	14	1058.82716	17	1995	403.85277
16	Autumn	15	1156.70370	8	2200	455.21810
17	Autumn	16	1293.20988	14	2270	498.10638
18	Autumn	17	1562.87654	23	2432	554.31649
19	Autumn	18	1983.33333	40	3298	778.44135
20	Autumn	19	1515.56790	19	2518	571.14972
21	Autumn	20	1321.70370	12	2259	508.99136
22	Autumn	21	1253.81481	21	2212	490.92858
23	Autumn	22	1130.20988	14	1868	411.96934
24	Autumn	23	828.72840	4	1523	292.97832

Observation: After studying the entire table, it was found that people on average rent more bikes during the summer. Renting bikes is avoided during winter

Task 4 – Weather Seasonality

```
# Task 9 - Weather Seasonality
weather_seasonality <- dbGetQuery(conn, "SELECT SEASONS,
                                         AVG(TEMPERATURE) AS avg_temperature,
                                         AVG(HUMIDITY) AS avg_humidity,
                                         AVG(WIND_SPEED) AS avg_wind_speed,
                                         AVG(VISIBILITY) AS avg_visibility,
                                         AVG(DEW_POINT_TEMPERATURE) AS avg_dew_point_temperature,
                                         AVG(SOLAR_RADIATION) AS avg_solar_radiation,
                                         AVG(RAINFALL) AS avg_rainfall,
                                         AVG(SNOWFALL) AS avg_snowfall,
                                         AVG(RENTED_BIKE_COUNT) AS avg_rented_bike_count
                                         FROM SEOUL_BIKE_SHARING
                                         GROUP BY SEASONS")
```

Observations: There were no unexpected phenomena noted, and the recorded temperature, rainfall and snowfall seemed appropriate for the season.

RSQLite.R* | all_time_high | hourly_popularity_temp_season | rental_seasonality | weather_seasonality | total_bike_count_seoul | similar_cities

Filter

	SEASONS	avg_temperature	avg_humidity	avg_wind_speed	avg_visibility	avg_dew_point_temperature	avg_solar_radiation	avg_rainfall	avg_snowfall	avg_rented_bike_count
1	Autumn	13.821580	59.04491	1.492101	1558.174	5.150594	0.5227827	0.11765617	0.06350026	924.1105
2	Spring	13.021685	58.75833	1.857778	1240.912	4.091389	0.6803009	0.18694444	0.00000000	746.2542
3	Summer	26.587711	64.98143	1.609420	1501.745	18.750136	0.7612545	0.25348732	0.00000000	1034.0734
4	Winter	-2.540463	49.74491	1.922685	1445.987	-12.416667	0.2981806	0.03282407	0.24750000	225.5412

Task 5 – Total Bike Count

```
# Task 10 - Total Bike Count and City Info for Seoul
total_bike_count_seoul <- dbGetQuery(conn, "SELECT SUM(BIKE_SHARING_SYSTEMS.BICYCLES) AS total_bike_count,
                                            WORLD_CITIES.CITY,
                                            WORLD_CITIES.COUNTRY,
                                            WORLD_CITIES.LAT,
                                            WORLD_CITIES.LNG,
                                            WORLD_CITIES.POPULATION
                                            FROM WORLD_CITIES
                                            INNER JOIN BIKE_SHARING_SYSTEMS
                                            ON WORLD_CITIES.CITY = BIKE_SHARING_SYSTEMS.CITY
                                            WHERE WORLD_CITIES.CITY = 'Seoul'")
```

RSQLite.R | total_bike_count_seoul | similar_cities | all_time_high

Filter

	total_bike_count	CITY	COUNTRY	LAT	LNG	POPULATION
1	20000	Seoul	Korea, South	37.5833	127	21794000

Observation: There are 20,000 bikes available for rent in Seoul.

Task 6 – Cities Similar to Seoul

```
# Task 11 - Find all city names and coordinates with comparable bike scale to Seoul's bike sharing system
similar_cities <- dbGetQuery(conn, "SELECT WORLD_CITIES.CITY, WORLD_CITIES.COUNTRY, WORLD_CITIES.LAT, WORLD_CITIES.LNG, WORLD_CITIES.POPULATION,
                                    BIKE_SHARING_SYSTEMS.BICYCLES AS total_bike_count
                                    FROM WORLD_CITIES
                                    INNER JOIN BIKE_SHARING_SYSTEMS
                                    ON WORLD_CITIES.CITY = BIKE_SHARING_SYSTEMS.CITY
                                    WHERE BIKE_SHARING_SYSTEMS.BICYCLES BETWEEN 15000 AND 20000")
dbDisconnect(conn)
```

RSQLite.R | total_bike_count_seoul | similar_cities | all_time_high | ho

Filter

	CITY	COUNTRY	LAT	LNG	POPULATION	total_bike_count
1	Beijing	China	39.9050	116.3914	19433000	16000
2	Ningbo	China	29.8750	121.5492	7639000	15000
3	Shanghai	China	31.1667	121.4667	22120000	19165
4	Weifang	China	36.7167	119.1000	9373000	20000
5	Zhuzhou	China	27.8407	113.1469	3855609	20000
6	Seoul	Korea, South	37.5833	127.0000	21794000	20000

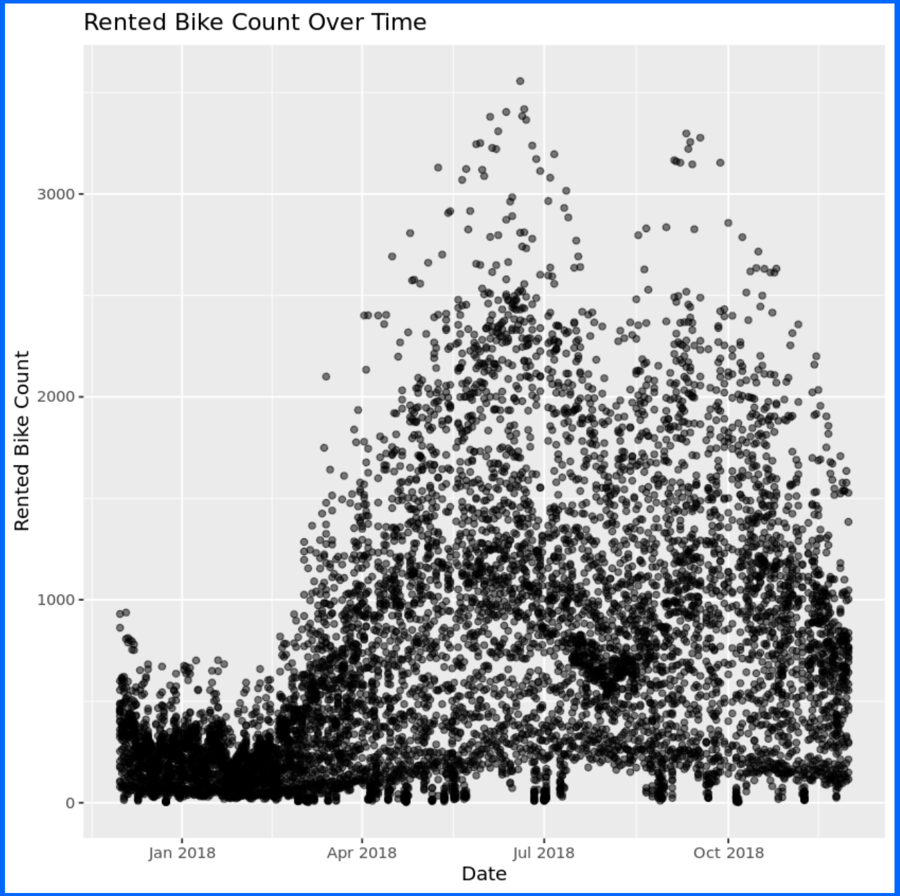
Observation: The five cities similar to Seoul are:

- Beijing
- Ningbo
- Shanghai
- Weifang
- Zhuzhou

EDA with Data Visualization - RESULTS

The forthcoming section will feature the Python code, the visualization, and the specific observations I made during the exploratory data analysis (EDA) process.

Task 1 – Rented Bike Count Over Time

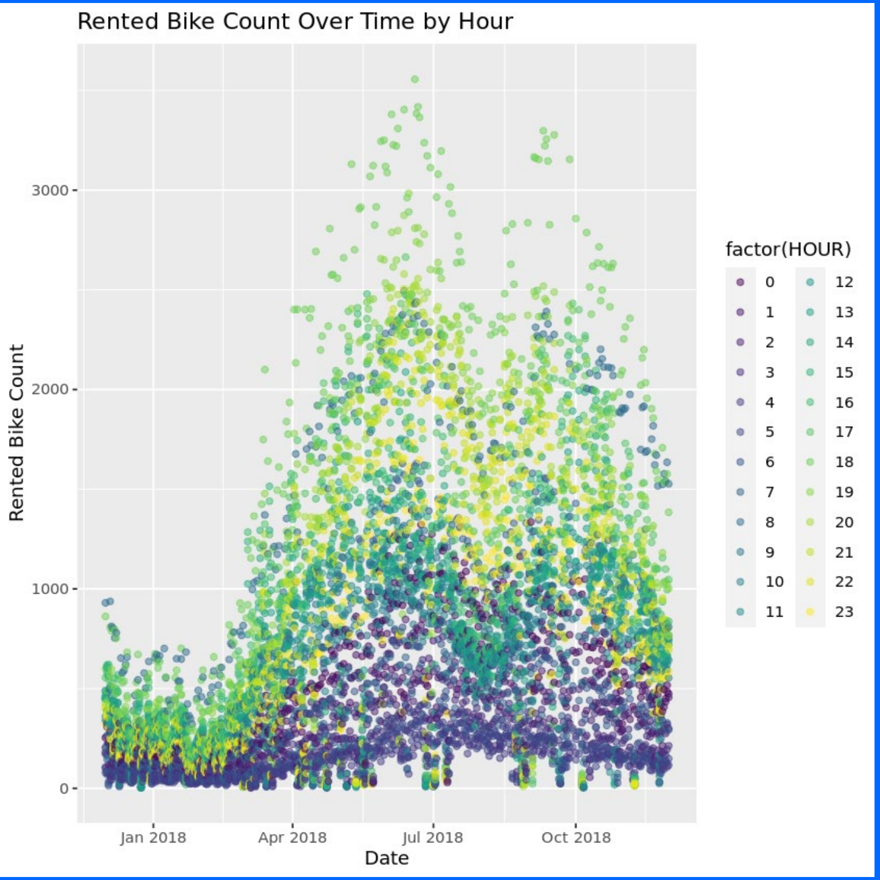


Solution 11

```
ggplot(seoul_bike_sharing, aes(x = DATE, y = RENTED_BIKE_COUNT, color = factor(HOUR))) +
  geom_point(alpha = 0.5) +
  labs(x = "Date", y = "Rented Bike Count", title = "Rented Bike Count Over Time by Hour")
[16]
```

Observations: We can observe the low renting of bikes in the winter months and a subsequent rise in renting bikes as summer and spring appear.

Task 2 – Rented Bike Count by Hour



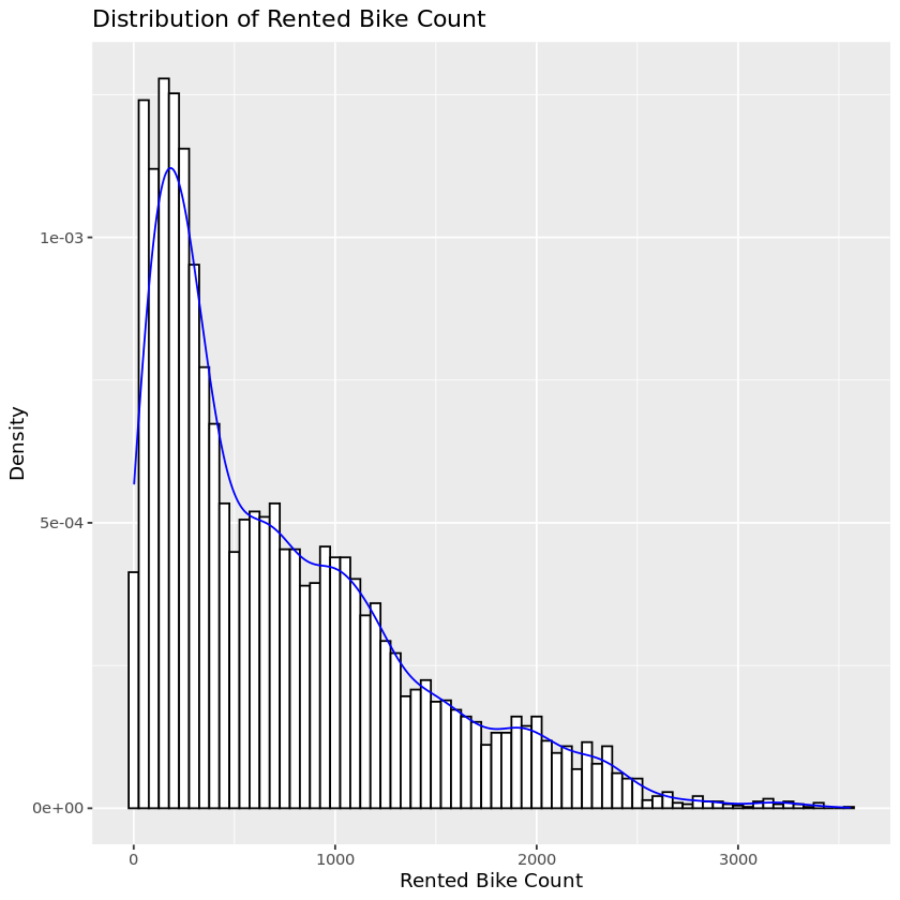
Solution 12

```
ggplot(seoul_bike_sharing, aes(x = RENTED_BIKE_COUNT, y = ..density..)) +
  geom_histogram(binwidth = 50, color = "black", fill = "white") +
  geom_density(color = "blue", alpha = 0.5) +
  labs(x = "Rented Bike Count", y = "Density", title = "Distribution of Rented Bike Count")
```

[17]

Observations: Majority of the bikes appear to be rented in the late afternoon and evening.

Task 3 – Distribution of Rented Bike Count



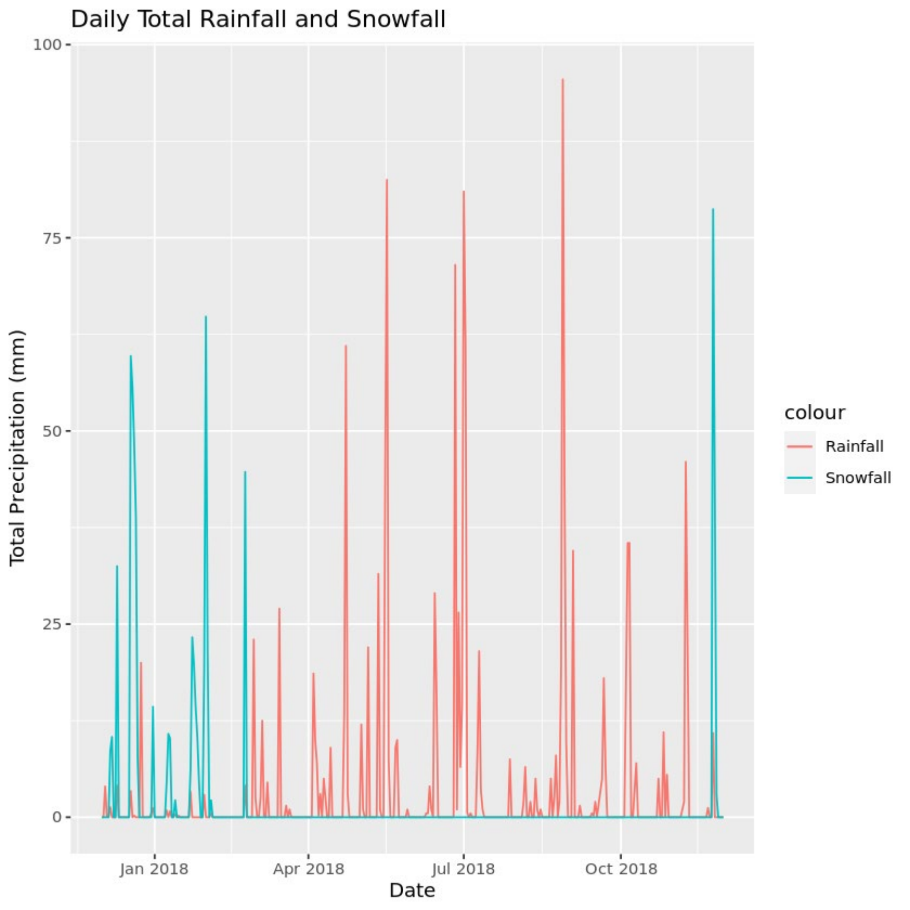
Solution 12

```
ggplot(seoul_bike_sharing, aes(x = RENTED_BIKE_COUNT, y = ..density..)) +
  geom_histogram(binwidth = 50, color = "black", fill = "white") +
  geom_density(color = "blue", alpha = 0.5) +
  labs(x = "Rented Bike Count", y = "Density", title = "Distribution of Rented Bike Count")
```

[17]

Observations: The number of bikes rented usually seems to be low, around 50 to 300 rented bikes at a time.

Task 4 – Daily Total Rainfall and Snowfall



Solution 15

```r
daily_weather_summary <- seoul_bike_sharing %>%
  group_by(DATE) %>%
  summarize(total_rainfall = sum(RAINFALL, na.rm = TRUE),
            total_snowfall = sum(SNOWFALL, na.rm = TRUE))

ggplot(daily_weather_summary, aes(x = DATE)) +
  geom_line(aes(y = total_rainfall, color = "Rainfall")) +
  geom_line(aes(y = total_snowfall, color = "Snowfall")) +
  labs(x = "Date", y = "Total Precipitation (mm)", title = "Daily Total Rainfall and Snowfall")
```

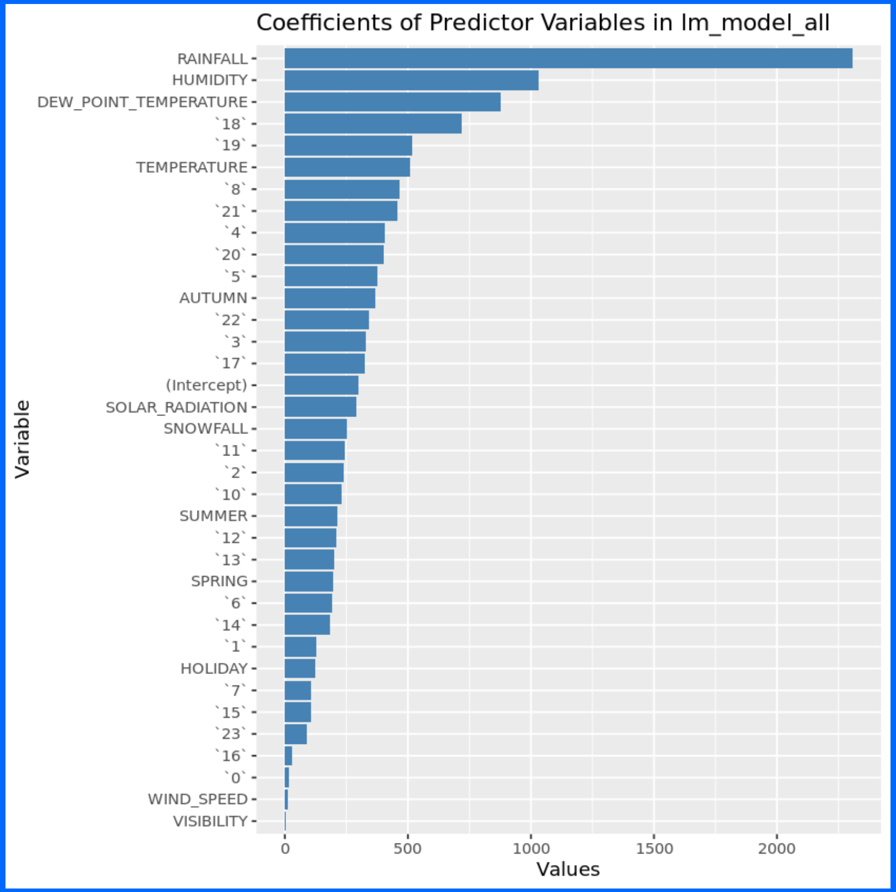
[23]

Observations: There were no unexpected phenomena noted, and the recorded rainfall and snowfall seemed appropriate for the season.

Predictive Analysis - RESULTS

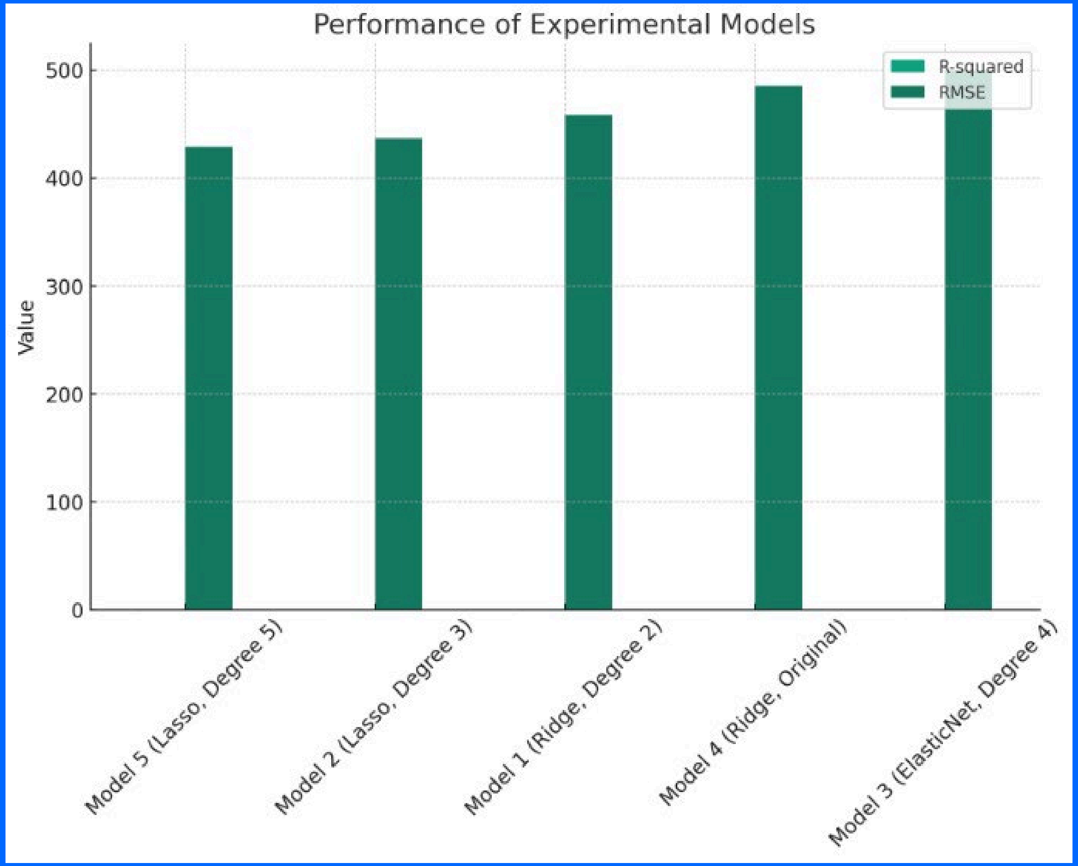
The forthcoming section will feature the visualizations representing the performance of the various regression models I created.

Task 1



Based on the bar chart displayed on the left, the weather-related variables with the highest coefficients include rainfall, humidity, and temperature. Another notable variable is the hour of 6 p.m., which also exhibits a considerable magnitude. However, it's important to note that several weather-related factors, such as rainfall and temperature, are correlated, making not all weather variables suitable for predicting bike rentals. Moreover, since rainfall and snowfall are subject to seasonal variations, they may not be suitable for inclusion in our model.

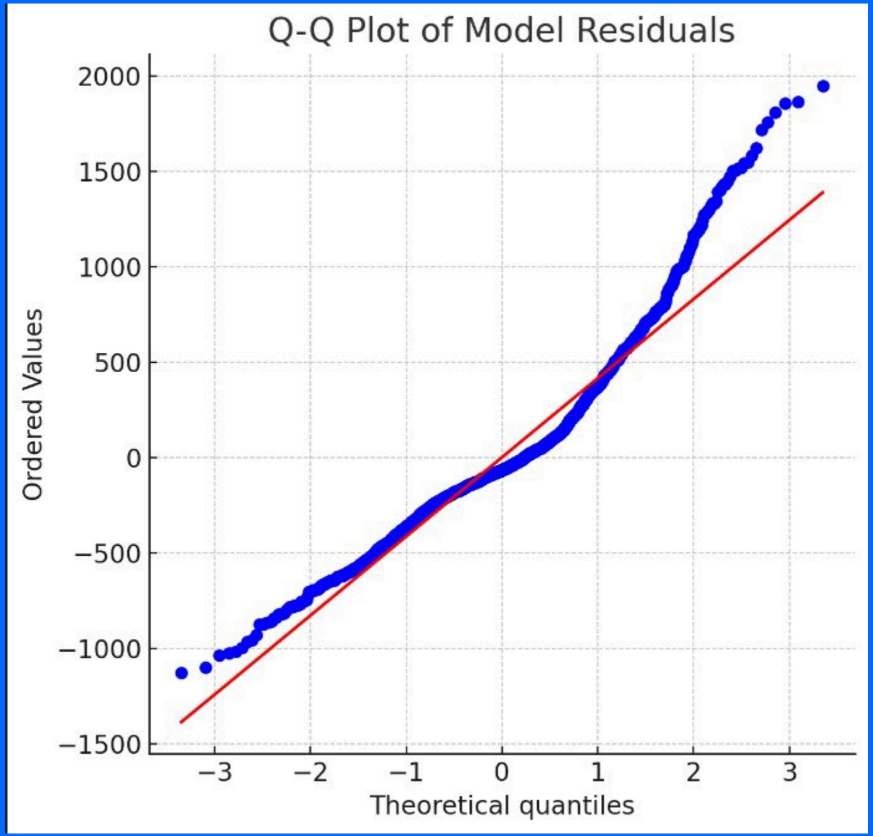
Task 2



Metric	Value
R-squared (R^2)	0.550
Root Mean Squared Error (RMSE)	428.96

My best performing model and how it compared to four other models I created

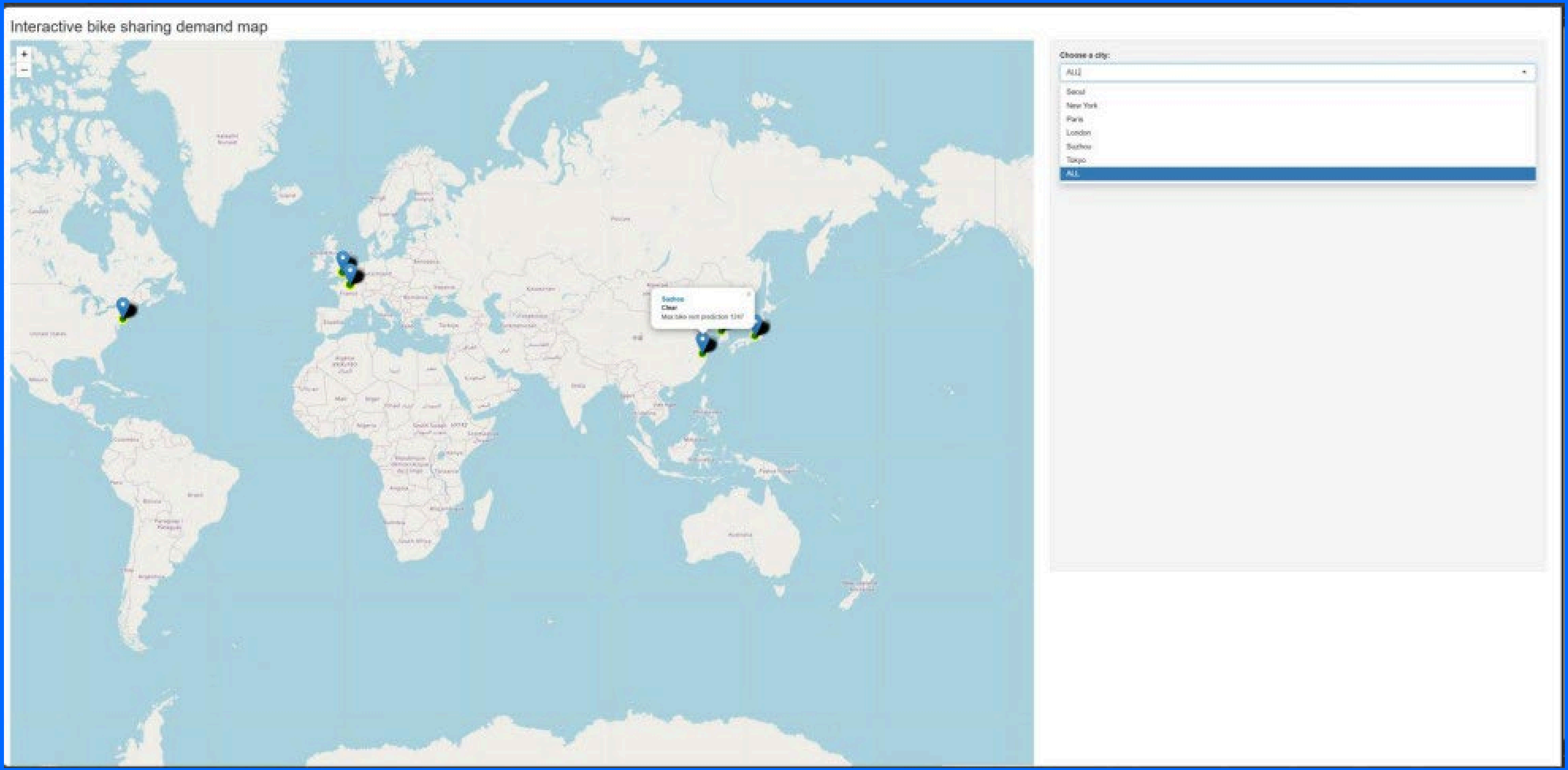
Task 3



QQ plot of the best model’s residuals

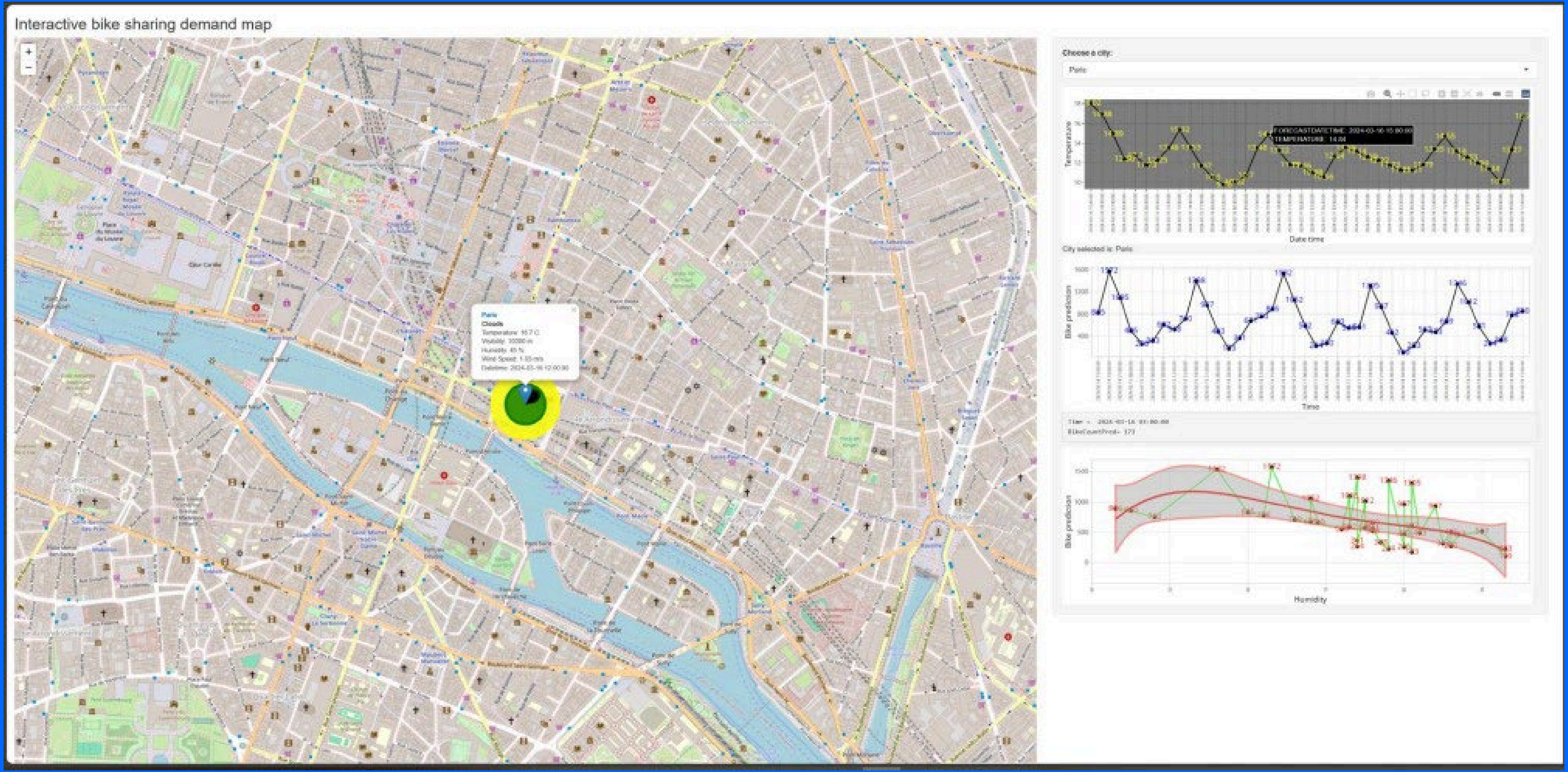
DASHBOARD

DASHBOARD TAB 1



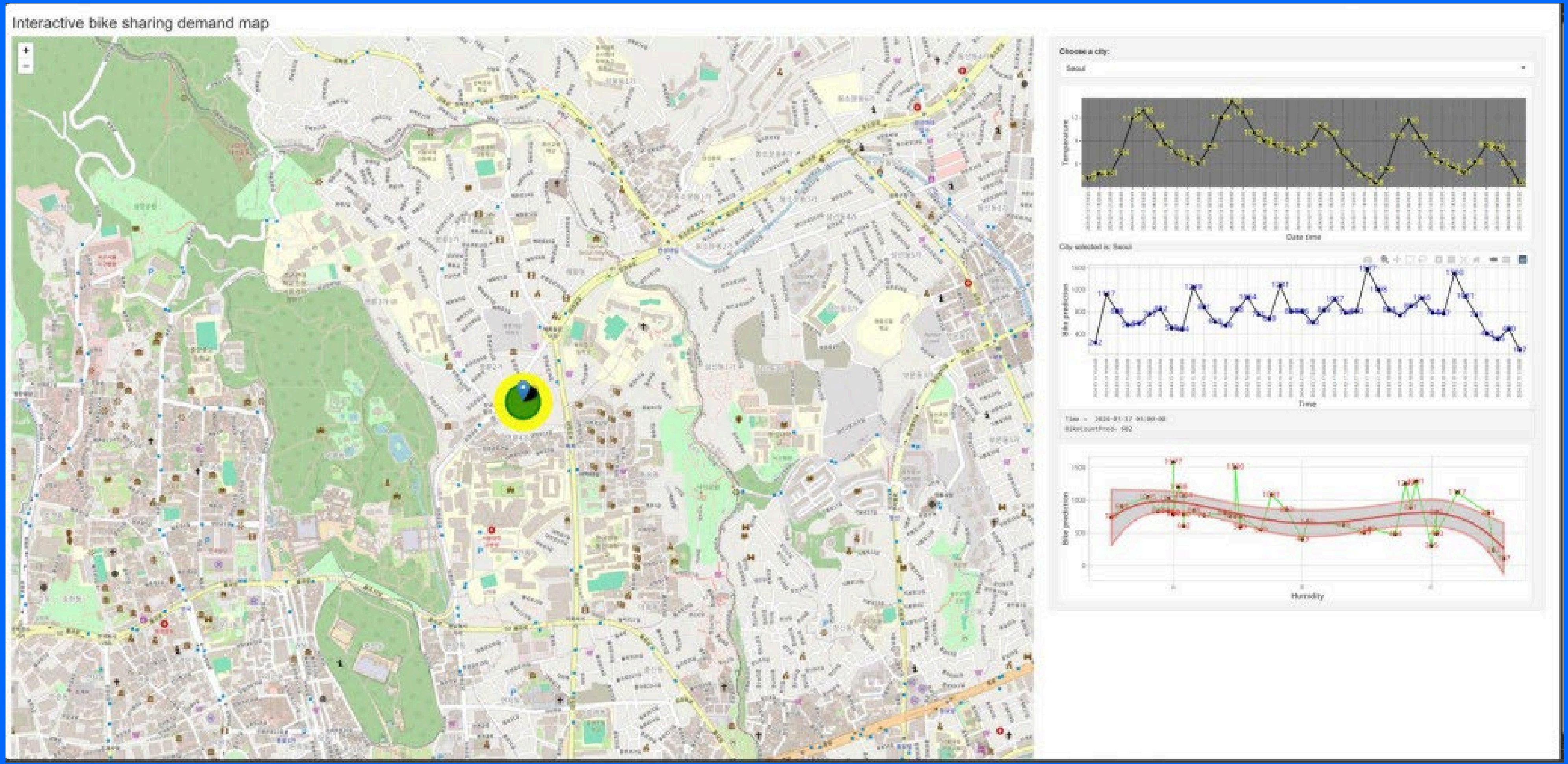
Overview of the cities which can be selected in the drop down menu on the side

DASHBOARD TAB 2



The Dashboard with Paris selected. The temperature trend lines, rental-bikes demand prediction line and the humidity and rental-bike demand correlation line are visible.

DASHBOARD TAB 3



The Dashboard with Seoul selected. The temperature trend lines, rental-bikes demand prediction line and the humidity and rental-bike demand correlation line are visible.

CONCLUSION

Summer reigns supreme: Bike sharing enjoys its peak ridership during the summer months, particularly June and July, when pleasant weather entices people outdoors.

Rush hour on two wheels: Weekday evenings, specifically between 6 and 7 pm, witness the highest demand for rentals as people commute home or embark on leisure rides.

Winter slumber: Cold weather significantly discourages cycling, leading to a substantial drop in bike sharing ridership during winter months.

Weather matters: Temperature and humidity play a key role in influencing ridership throughout the day. Comfortable temperatures and low humidity tend to encourage people to choose bicycles.

APPENDIX

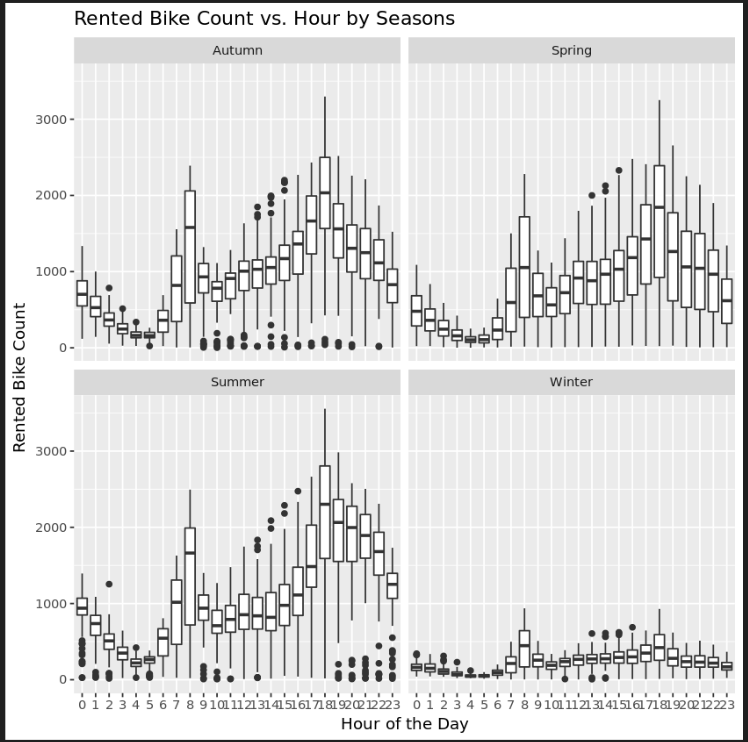
Appendix 1 -
The remaining RSQLite queries done during the EDA Process.

```r
# Task 1 - Record Count
record_count <- dbGetQuery(conn, "SELECT COUNT(*) FROM SEOUL_BIKE_SHARING")

# Task 2 - Operational Hours
operational_hours <- dbGetQuery(conn, "SELECT COUNT(*) FROM SEOUL_BIKE_SHARING WHERE RENTED_BIKE_COUNT > 0")

# Task 3 - Weather Outlook
weather_outlook <- dbGetQuery(conn, "SELECT * FROM CITIES_WEATHER_FORECAST WHERE CITY = 'Seoul' LIMIT 1")

# Task 4 - Seasons
seasons <- dbGetQuery(conn, "SELECT DISTINCT SEASONS FROM SEOUL_BIKE_SHARING")

# Task 5 - Date Range
date_range <- dbGetQuery(conn, "SELECT MIN(DATE) AS first_date, MAX(DATE) AS last_date FROM SEOUL_BIKE_SHARING")
```

```
> record_count
  COUNT(*)
1     8465
> operational_hours
  COUNT(*)
1     8465
> weather_outlook
   CITY WEATHER VISIBILITY  TEMP TEMP_MIN TEMP_MAX PRESSURE HUMIDITY WIND_SPEED WIND_DEG SEASON    FORECAST_DATETIME
1 Seoul   Clear      10000 12.32    10.91    12.32     1015       50       2.18      248 Spring 2021-04-16 12:00:00
> seasons
  SEASONS
1  Winter
2  Spring
3  Summer
4  Autumn
> date_range
  first_date  last_date
1 01/01/2018 31/12/2017
```

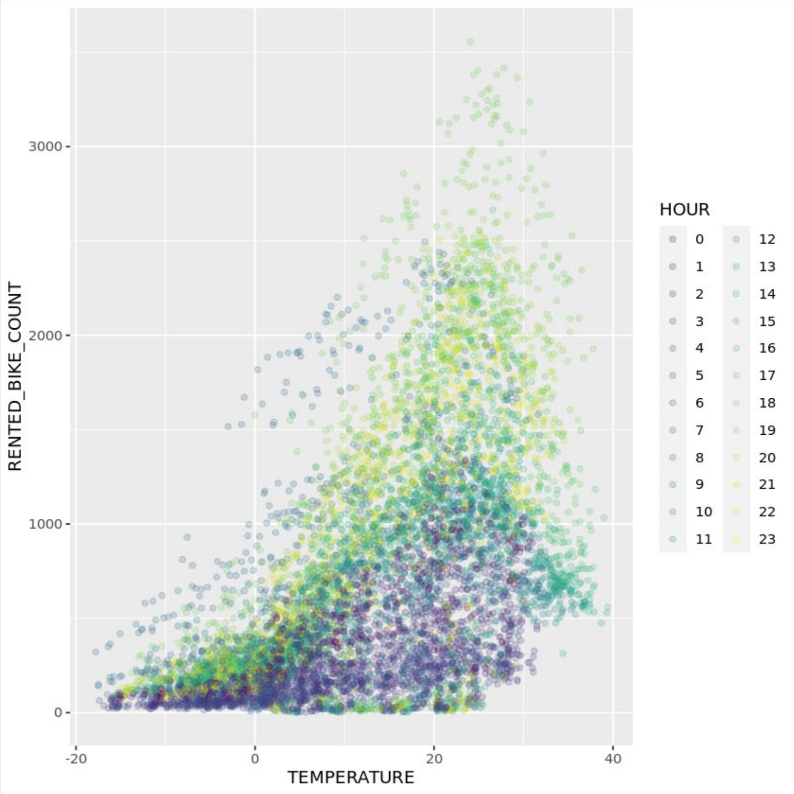
Appendix 2 -
Rented Bike Count vs Hour by Seasons

```r
# provide your solution here
ggplot(seoul_bike_sharing, aes(x = factor(HOUR), y = RENTED_BIKE_COUNT)) +
  geom_boxplot() +
  facet_wrap(~SEASONS) +
  labs(x = "Hour of the Day", y = "Rented Bike Count", title = "Rented Bike Count vs. Hour by Seasons")
```



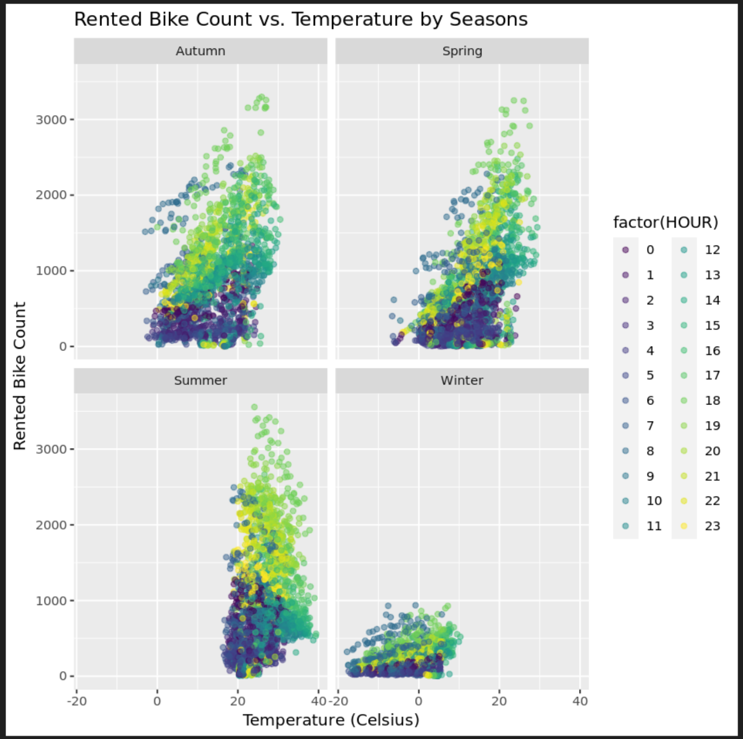
Appendix 3 -
Rented Bike Count by Temperature

```
ggplot(seoul_bike_sharing) +
  geom_point(aes(x=TEMPERATURE,y=RENTED_BIKE_COUNT,colour=HOUR),alpha=1/5)
```



Appendix 4 -
Rented Bike Count vs Temperature by Seasons

```
# provide your solution here
ggplot(seoul_bike_sharing, aes(x = TEMPERATURE, y = RENTED_BIKE_COUNT, color = factor(HOUR))) +
  geom_point(alpha = 0.5) +
  facet_wrap(~SEASONS) +
  labs(x = "Temperature (Celsius)", y = "Rented Bike Count", title = "Rented Bike Count vs. Temperature by Seasons")
```



Thank you

Arnav Bhatia