BIKE SHARING DEMAND IN URBAN AREAS

PRESENTED BY: SHISHIR BHATTARAI

Outline

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

Executive Summary

- Summary of methodologies:

✓ Data Collection through API

✓ Data Collection with Web Scraping

✓ Data Wrangling with DPLYR

✓ Exploratory Data Analysis with SQL

✓ Exploratory Data Analysis with Visualization (using ggplot2)

✓ Model Prediction

✓ Interactive Analysis with Shiny R

- Summary of results:

✓ Exploratory Data Analysis Results

✓ Interactive analytics result

✓ Predictive analytics result

Introduction

- **BIKE SHARING SYSTEM:**

Rental Bikes are available in many cities around the globe. So, it's very important for each of these cities to provide a reliable supply of rental bikes to optimize availability and accessibility to the public at all times by minimizing the cost of these programs. This system works on the basis of supply of bikes as per the demand. So, it's very necessary to predict the numbers of bikes required each hour of the day, based on the current conditions such as the weather.

- **PROBLEMS YOU WANT TO FIND ANSWERS**

✓ What factors determines the number of bikes rented each hour?

✓ Relation of various weather factors with the number of bikes rented

Methodology

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

Data collection

- ✓ Data was collected using OpenWeather API and web scraping was done from Wikipedia using httr.
- ✓ The response content were in a json format which were converted to a dataframe.
- ✓ Web Scraping was done from Wikipedia for Bicycle Sharing Systems using rvest.
- ✓ The main objective was to extract the table for the further analysis.

Data collection (Web Scraping)

- ✓ We applied web scraping to scrap Bicycle-Sharing Systems records from Wikipedia.

```r
require (rvest)
library (rvest)
#Extracting and exporting

#getting the root of html node
url <- "https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems"
#Reading the html node from the url
root_node <- read_html(url)
#Getting the table nodes
table_node <- html_nodes(root_node,"table")
table_data <- html_table(table_node,fill=TRUE)[[1]] #extracts the first table
#Converting the table into dataframe
raw_bike_sharing_systems <- as.data.frame(table_data)
#Summarizing the dataframe
summary(raw_bike_sharing_systems)

#Exporting the dataframe as csv file
write.csv(raw_bike_sharing_systems,"raw_bike_sharing_systems.csv")

```

Data collection (OpenWeather API)

- ✓ We used GET request to the OpenWeather API to collect data and saved it as "raw_cities_weather_forecast.csv" for further analysis.
- ✓ The data were collected for cities: Seoul, Washington, D.C., Paris and Suzhou
- ✓ The github link for both Web scraping and API's code is :

  Webscraping.R

Data wrangling

The following activities were performed in the data wrangling section:

- ✓ Datasets were read as prescribed using read_csv().
- ✓ All the column names were converted to Uppercase using toupper().
- ✓ Used grepl to search strings for non-digital characters.
- ✓ Checked and removed the reference pattern [A-z0-9) from COUNTRY, CITY and SYSTEM Column.
- ✓ Detected and handled missing values in RENTED_BIKE_COUNT column
- ✓ Applied min-max normalization on `RENTED_BIKE_COUNT`, `TEMPERATURE`, `HUMIDITY`, `WIND_SPEED`, `VISIBILITY`, `DEW_POINT_TEMPERATURE`, `SOLAR_RADIATION`, `RAINFALL`, `SNOWFALL`
- ✓ The github link for the code is given below:

dataWrangling.R

dataWranglingWithDPLYR.R

Data wrangling (Output)

```
>
>
>
> sub_bike_sharing_df %>%
+   select(CITY, SYSTEM, BICYCLES) %>%
+   filter(find_reference_pattern(CITY) | find_reference_pattern(SYSTEM) | find_reference_pattern(BICYCLES))
# A tibble: 228 × 3
   CITY                    SYSTEM            BICYCLES
   <chr>                   <chr>             <chr>
 1 Tirana[5]               NA                200
 2 Buenos Aires[6][7]      Serttel Brasil[8] 4000
 3 Mendoza[10]             NA                40
 4 Melbourne[12]           PBSC & 8D         676
 5 Melbourne[12]           4 Gen. oBike      1250
 6 Brisbane[14][15]        3 Gen. Cyclocity  2000
 7 Lower Austria[16]       3 Gen. nextbike   1300
 8 Different locations[19] Blue-bike         1790 (2019)[21]
 9 Brussels[24]            3 Gen. Cyclocity  4115[25]
10 Namur[26]               3 Gen. Cyclocity  200
# ℹ 218 more rows
# ℹ Use `print(n = ...)` to see more rows
>
> head(sub_bike_sharing_df)
# A tibble: 6 × 4
  COUNTRY   CITY                  SYSTEM            BICYCLES
  <chr>     <chr>                 <chr>             <chr>
1 Albania   Tirana[5]             NA                200
2 Argentina Buenos Aires[6][7]    Serttel Brasil[8] 4000
3 Argentina Mendoza[10]           NA                40
4 Argentina Rosario               NA                480
5 Argentina San Lorenzo, Santa Fe Biciudad          80
6 Australia Melbourne[12]         PBSC & 8D         676
>
> #Write a custom function using `stringr::str_extract` to extract the first digital substring match and convert it into nume
ric type For example, extract the value '32' from `32 (including 6 rollers) [162]`.
> # Extract the first number
> extract_num <- function(columns){
+   # Define a digital pattern
+   digitals_pattern <- "[^0-9]"
+   # Find the first match using str_extract
```

```
# A tibble: 6 × 4
  COUNTRY   CITY                  SYSTEM            BICYCLES
  <chr>     <chr>                 <chr>             <chr>
1 Albania   Tirana[5]             NA                200
2 Argentina Buenos Aires[6][7]    Serttel Brasil[8] 4000
3 Argentina Mendoza[10]           NA                40
4 Argentina Rosario               NA                480
5 Argentina San Lorenzo, Santa Fe Biciudad          80
6 Australia Melbourne[12]         PBSC & 8D         676
>
> #Write a custom function using `stringr::str_extract` to extract the first digital substring match and convert it into nume
ric type For example, extract the value '32' from `32 (including 6 rollers) [162]`.
> # Extract the first number
> extract_num <- function(columns){
+   # Define a digital pattern
+   digitals_pattern <- "[^0-9]"
+   # Find the first match using str_extract
+   str_extract(columns,digitals_pattern)
+   # Convert the result to numeric using the as.numeric() function
+   columns <- as.numeric(columns)
+ }
>
> # Use the mutate() function on the BICYCLES column
>
> sub_bike_sharing_df <- sub_bike_sharing_df %>%
+   mutate(BICYCLES = extract_num(BICYCLES))
Warning message:
There was 1 warning in `mutate()`.
ℹ In argument: `BICYCLES = extract_num(BICYCLES)`.
Caused by warning in `extract_num()`:
! NAs introduced by coercion
>
> #summary
> summary(sub_bike_sharing_df$BICYCLES)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
      4      70     300    1879    1000   78000     147
>
> #Write dataset to `bike_sharing_systems.csv`
> write.csv(sub_bike_sharing_df, "bike_sharing_systems.csv")
>
```

EDA with SQL

Following SQL queries were performed while doing Exploratory Data Analysis with SQL :

- ✓ CREATE : To create various tables.
- ✓ SELECT: To select desired or needed columns.
- ✓ GROUP BY: To produce the result with respect to the desired group.
- ✓ ORDER BY: To order the data in ascending or descending order.
- ✓ MIN/ MAX / AVG : To calculate minimum, maximum and average values respectively.
- ✓ RIGHT JOIN: To return all records from right table and the matching one from left table.
- ✓ INNER JOIN : To select both tables having matching values.

EDA with SQL (CODE SAMPLE)

```
#TASK 1 : FIND NO. OF RECORDS IN THE SEOUL_BIKE_SHARING_DATASET
dbGetQuery(con, "SELECT COUNT(*) AS NO_OF_RECORDS FROM SEOUL_BIKE_SHARING")

#TASK 2: Determine how many hours had non-zero rented bike count.
dbGetQuery(con,
           "SELECT COUNT(HOUR) FROM seoul_bike_sharing
           WHERE RENTED_BIKE_COUNT >0")

#TASK 3: Weather Outlook Query the the weather forecast for Seoul over the next 3 hours.
#Recall that the records in the CITIES_WEATHER_FORECAST dataset are 3 hours apart,
#so we just need the first record from the query.
dbGetQuery(con,
           "SELECT * FROM CITIES_WEATHER_FORECAST
           WHERE CITY='Seoul'
           LIMIT 1
            ")

#TASK 4: Find which seasons are included in the seoul bike sharing dataset
dbGetQuery(con,
           "SELECT DISTINCT(SEASONS) AS SEASONS_INCLUDED
           FROM SEOUL_BIKE_SHARING")

#TASK 5: Find the first and last dates in the Seoul Bike Sharing dataset.
dbGetQuery(con,
           "SELECT MIN(DATE) AS FIRST_DATE, MAX(DATE) AS LAST_DATE
           FROM SEOUL_BIKE_SHARING")

#TASK 6: determine which date and hour had the most bike rentals
dbGetQuery(con,
           "SELECT DATE,HOUR
           FROM SEOUL_BIKE_SHARING
           WHERE RENTED_BIKE_COUNT= (SELECT MAX(RENTED_BIKE_COUNT)
           FROM SEOUL_BIKE_SHARING)")
```

✓ **For rest of the code, kindly visit the Following github link:**

RSQLITE.R

EDA with data visualization

The following charts were used for visualization:

- ✓ Scatter Plot
- ✓ Histogram
- ✓ Outliers (Boxplot)

Predictive analysis

The following activities were done for the predictive analysis:

- ✓ The normalized dataset of seoul_bike_sharing was read.
- ✓ Initial_split() was performed with prob= ¾ and splitted into training and testing data.
- ✓ Linear regression model was developed using both specified variables and all variables.
- ✓ Performed higher order polynomial fits and visualized too.
- ✓ Made predictions using polynomial models.
- ✓ The github link for the code is as follows:

  linearRegressionModel.R

  improveLinearModel.R

Build a R Shiny dashboard

For the Shiny R dashboard, following tasks were performed:

- ✓ Used leaflet library for map .
- ✓ Used selectInput() for drop down list to select city.
- ✓ Plotted a temperature line of the selected city.
- ✓ Added a bike sharing demand prediction line.
- ✓ Also, built prediction chart to show correlation of humidity and bike sharing demand prediction.
- ✓ The github link for the code are as follows:

  ui.R

  server.R

Results

- Exploratory data analysis results
- Predictive analysis results
- A dashboard demo in screenshots

EDA with SQL

Busiest bike rental times

In date 19/06/2018 on 18th hour, the Seoul bike sharing has the most bike rentals. The no. of bike rentals was 3556. It was the season of summer. From this it can be analyzed that, the summer season with sunny weather is preferred and suitable for the bike rentals and can be made available based on these parameters.

	DATE	HOUR	RENTED_BIKE_COUNT	SEASONS
1	19/06/2018	18	3556	Summer

Hourly popularity and temperature by seasons

- ✓ The result shows that the summer has the highest average bike rentals with approx. 2135 bikes rented at 6pm (hour 18). This suggests that people prefer to rent bikes during the summer months, likely due to more favorable weather conditions.
- ✓ Similarly, evening hours (18-22) see higher rentals compared to other times of the day. This suggests that people tend to rent bikes for recreational purposes in the evenings.
- ✓ Also, the data shows positive correlation between temperature and average bike rentals. Temperature likely to influence the bike rentals but not directly.

	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

Rental Seasonality

- ✓ The data shows significant variation in average rental counts between seasons. Summer has the highest average rental count, followed by Autumn and Spring. Winter has the lowest average rental count, likely due to less favorable weather conditions for cycling.
- ✓ The standard deviation provides insights into the variability of rental counts with each seasons. Summer and Autumn have higher standard deviations compared to Spring and Winter. This suggests that the rental counts fluctuate more in Summer and Autumn compared to Spring and Winter.

	SEASONS	AVG(RENTED_BIKE_COUNT)	MIN(RENTED_BIKE_COUNT)	MAX(RENTED_BIKE_COUNT)	STANDARD_DEVIATION
1	Autumn	924.1105	2	3298	617.3885
2	Spring	746.2542	2	3251	618.5247
3	Summer	1034.0734	9	3556	690.0884
4	Winter	225.5412	3	937	150.3374

Weather Seasonality

- ✓The data shows a clear correlation between weather conditions and average bike rentals in Seoul across different seasons.
- ✓We see a greater alignment between favorable conditions and high bike rentals. Autumn has milder temperature and moderate humidity compared to Spring, and also has a higher average rental count. Similar case goes to Spring and Winter.

	SEASONS	AVG(TEMPERATURE)	AVG(HUMIDITY)	AVG(WIND_SPEED)	AVG(VISIBILITY)	AVG(DEW_POINT_TEMPERATURE)
1	Summer	26.587711	64.98143	1.609420	1501.745	18.750136
2	Autumn	13.821580	59.04491	1.492101	1558.174	5.150594
3	Spring	13.021685	58.75833	1.857778	1240.912	4.091389
4	Winter	-2.540463	49.74491	1.922685	1445.987	-12.416667

	AVG(SOLAR_RADIATION)	AVG(RAINFALL)	AVG(SNOWFALL)	AVG(RENTED_BIKE_COUNT)
1	0.7612545	0.25348732	0.00000000	1034.0734
2	0.5227827	0.11765617	0.06350026	924.1105
3	0.6803009	0.18694444	0.00000000	746.2542
4	0.2981806	0.03282407	0.24750000	225.5412

Bike-sharing info in Seoul

There were not much info about Seoul, except latitude, longitude and population. Probably, the dataset has some error. It shows the attached output.

```
> dbGetQuery(con,
+            "SELECT B.BICYCLES, B.CITY, B.COUNTRY, W.LAT, W.LNG, W.POPULATION
+             FROM BIKE_SHARING_SYSTEMS AS B
+             RIGHT JOIN WORLD_CITIES AS W ON B.CITY = W.CITY_ASCII
+             WHERE W.CITY_ASCII ='Seoul'"
+ )
  BICYCLES CITY COUNTRY     LAT LNG POPULATION
1       NA <NA>    <NA> 37.5833 127   21794000
>
```

Cities similar to Seoul

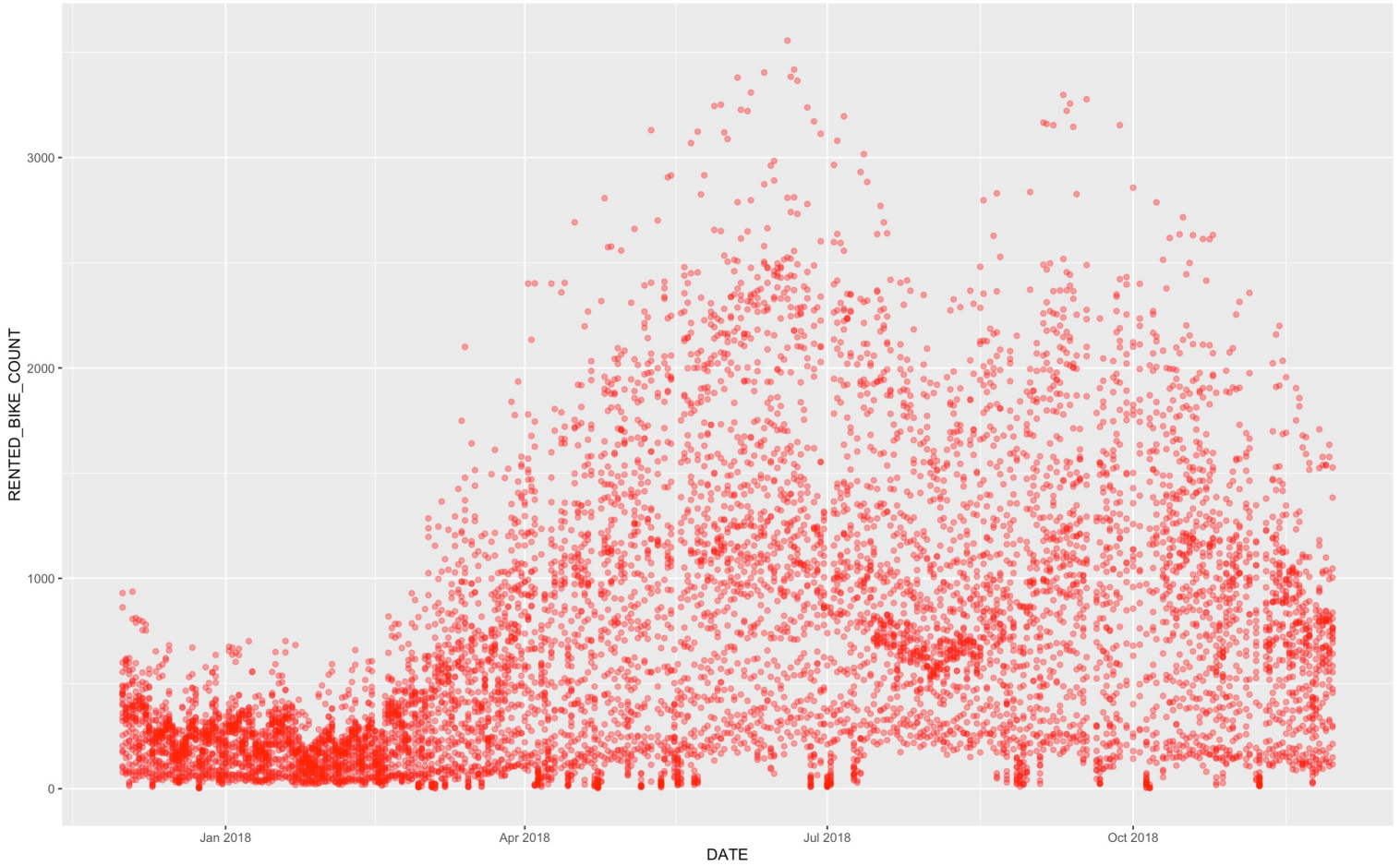
- ✓ All five cities listed (Beijing, Ningbo, Weifang, Xi'an, and Zhuzhou) have a moderate number of bicycles in their bike-sharing systems, ranging from 15,000 to 20,000.
- ✓ So, China has a strong presence in implementing moderate-sized bike-sharing systems in its cities as all the cities listed are in China.
- ✓ Population doesn't necessarily correspond to bicycle count. For instance, Beijing has the highest population but doesn't have the highest number of bicycles.

	CITY	COUNTRY	LAT	LNG	POPULATION	BICYCLES
1	Beijing	China	39.9050	116.3914	19433000	16000
2	Ningbo	China	29.8750	121.5492	7639000	15000
3	Weifang	China	36.7167	119.1000	9373000	20000
4	Xi'an	China	34.2667	108.9000	7135000	20000
5	Zhuzhou	China	27.8407	113.1469	3855609	20000

EDA with Visualization

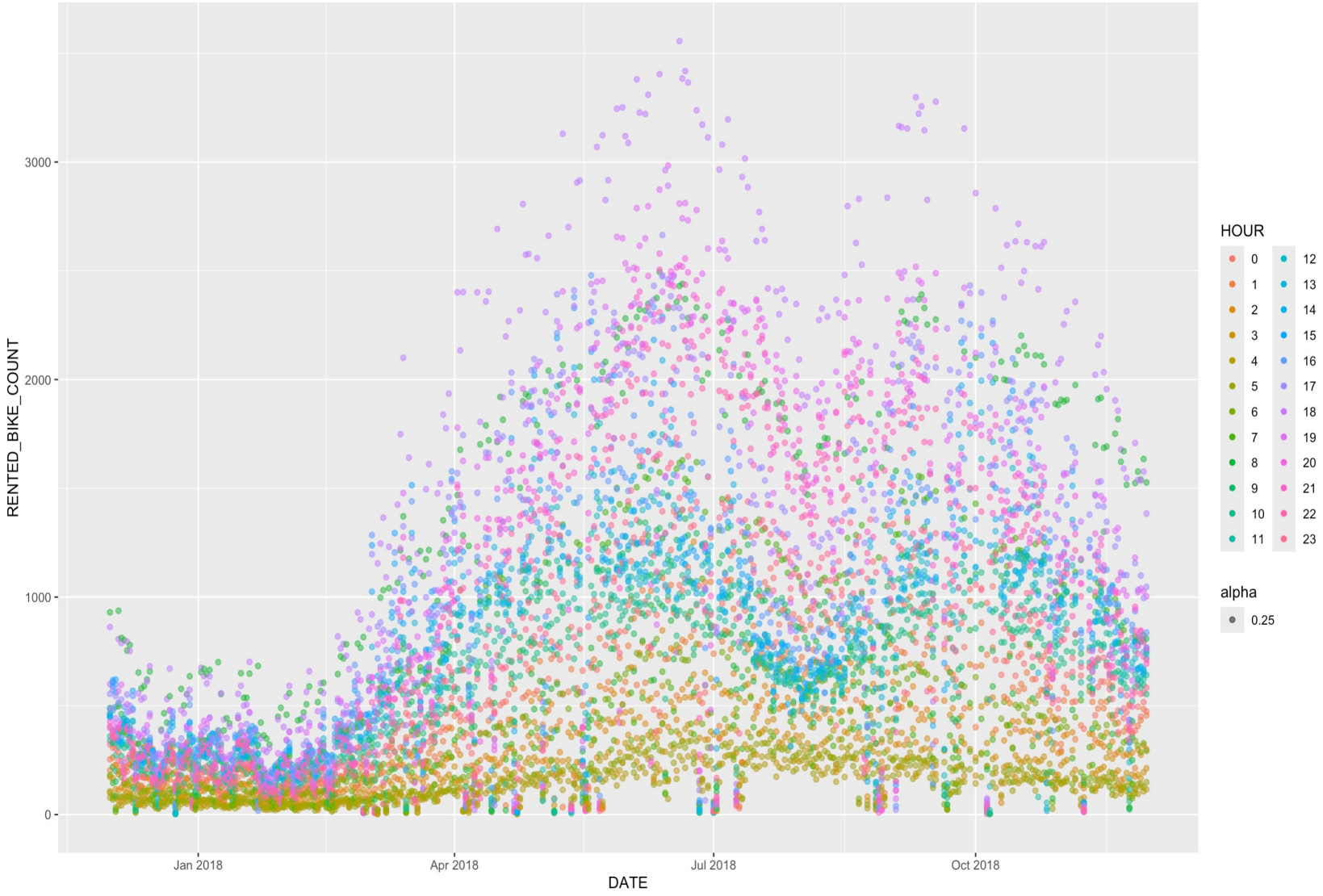
Bike rental vs. Date

- ✓ The scatter plot shows that the no. of bike rentals are increasing as the month Passes by.
- ✓ The month of April to July has the highest no. of rentals many times. It suggests that the summer has the highest number of bike rentals compared to others. And, the month of winter has significantly lesser no. of bike rentals.



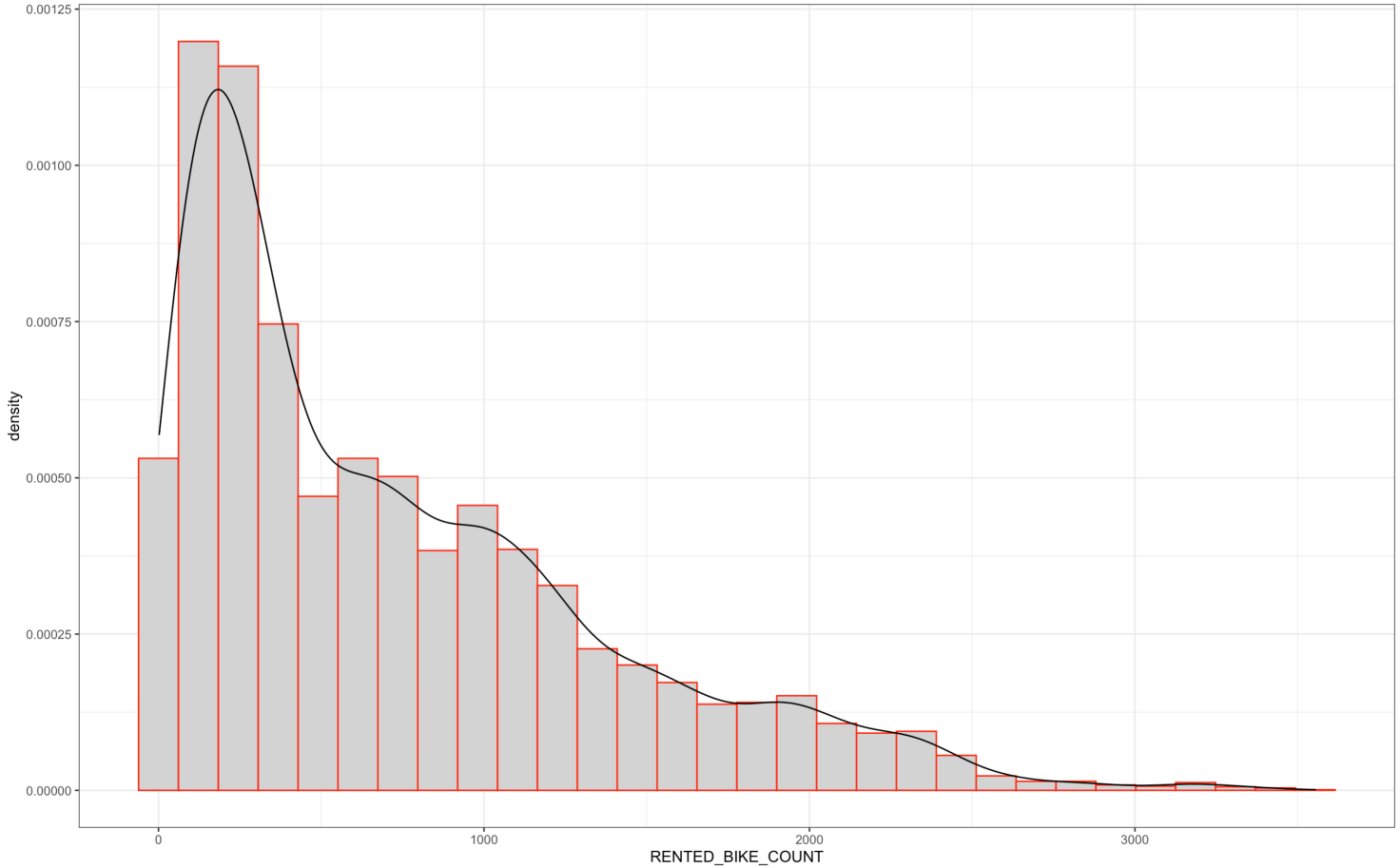
Bike rental vs. Datetime

- ✓ The scatter plot shows the number of bikes rented over a period of time where color represents the hours at which bikes are rented.
- ✓ The plot shows that the evening period has the highest number of bike rentals as compared to morning and afternoon.
- ✓ So, it suggests to have more bikes for the evening specially in the summer season.



Bike rental histogram

- ✓ The graph shows the most common number of bikes rented (peak of the curve). This tells you what to expect on an average day.
- ✓ The width of the curve reveals how much the daily rentals deviate from the typical amount. Wider curve means more variation, possibly due to weather or events.
- ✓ The curve's ends hint at days with significantly higher or lower rentals than usual.



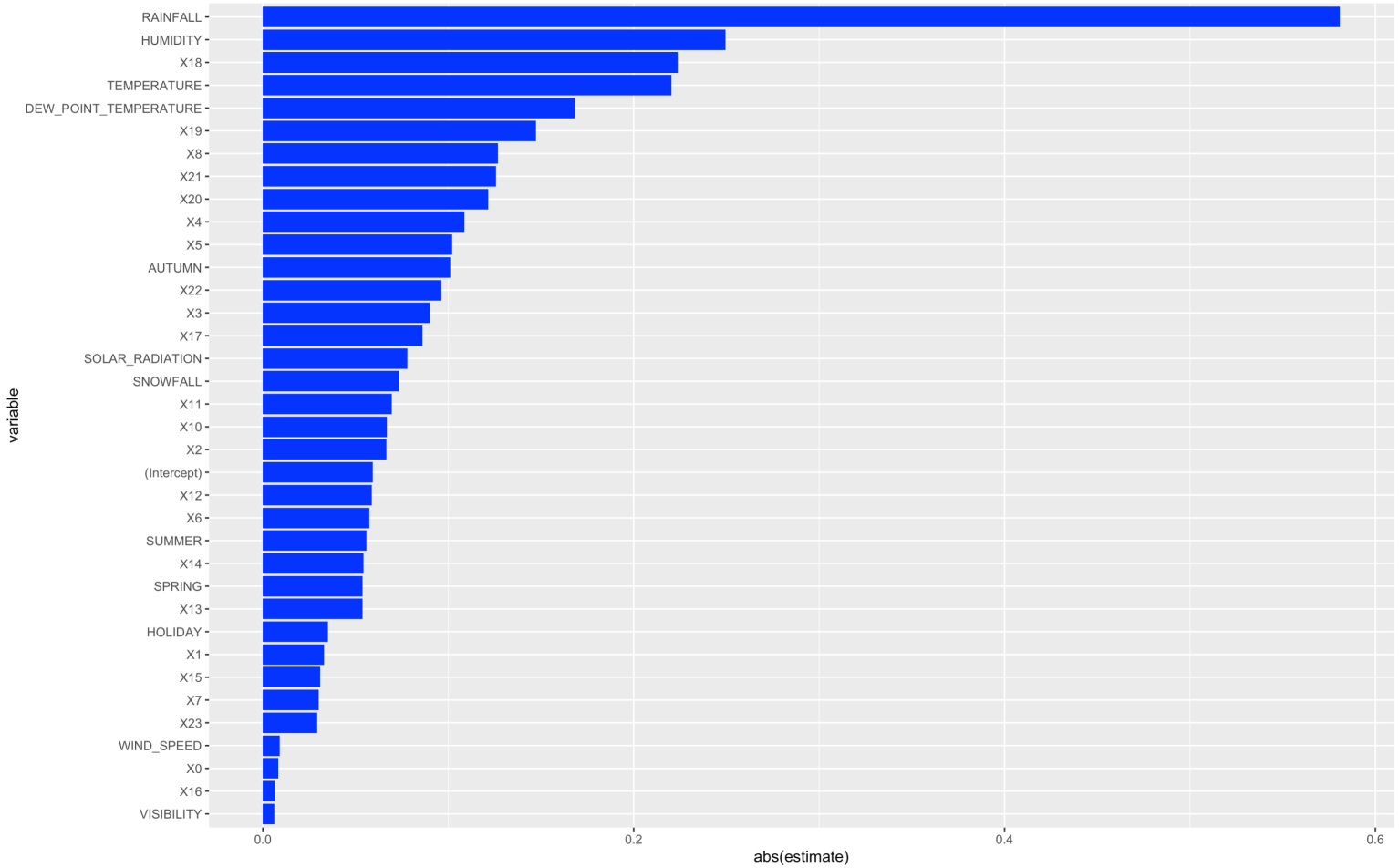
Daily total rainfall and snowfall

- ✓ The bar chart shows the increase in rainfall from April reaching its peak on around October.
- ✓ Also, the snowfall occurs heavily at the end of the year and moderate at the starting of the year.

Predictive analysis

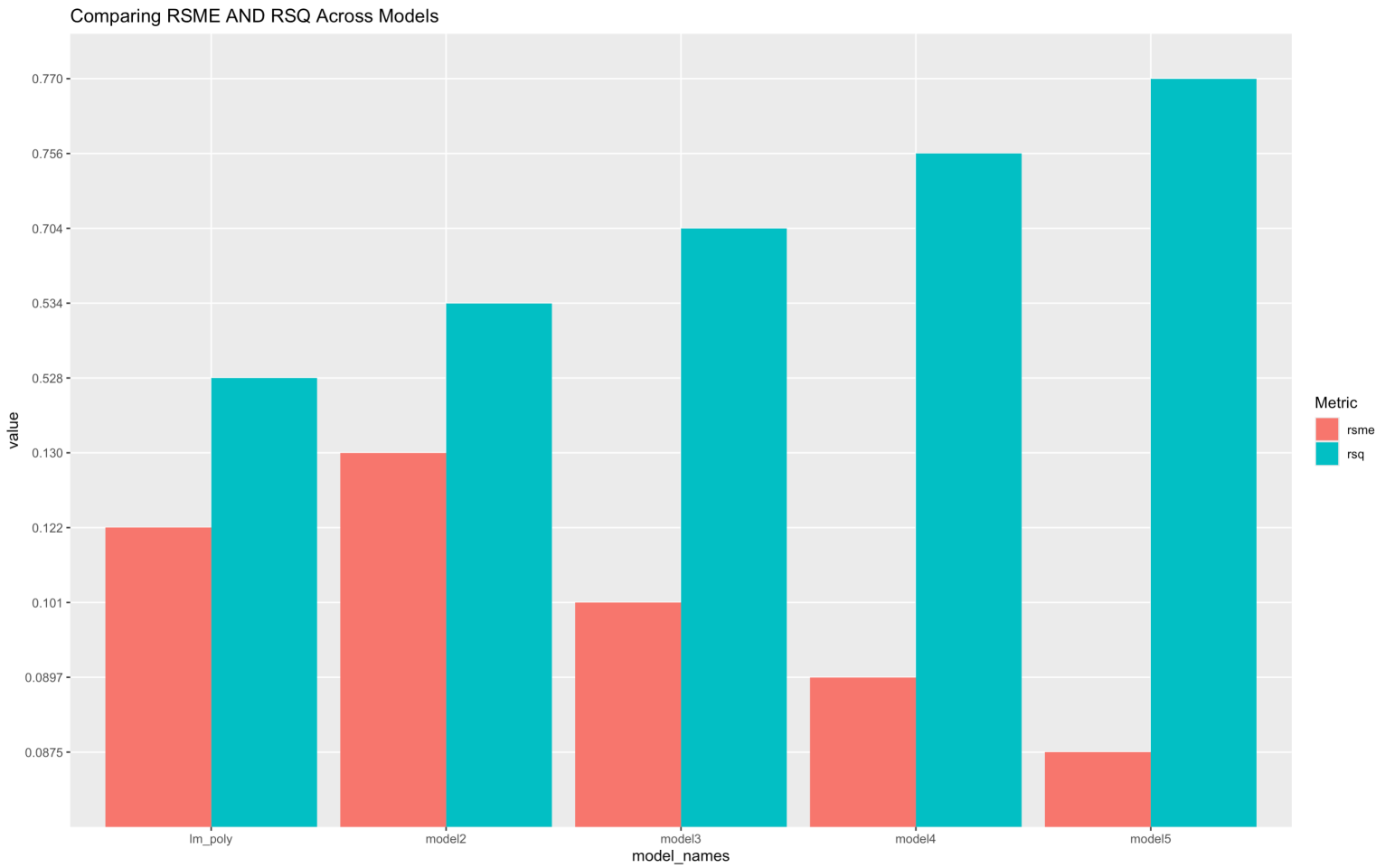
Ranked coefficients

- ✓ The chart simply shows the variables with the highest coefficient value to the lowest.
- ✓ Rainfall has the highest while visibility has the lowest .
- ✓ Some variables like WIND_SPEED, VISIBILITY has less influence over the bike rentals. It's possibly due to the very favorable situation over these entities in the particular city.



Model evaluation

- ✓ The graph shows the RSME and RSQ of the five models.
- ✓ Among the five models, model5 seems to be the best model as it has lesser RSME value comparatively and has more RSQ value as compared to other models.



The best performing model

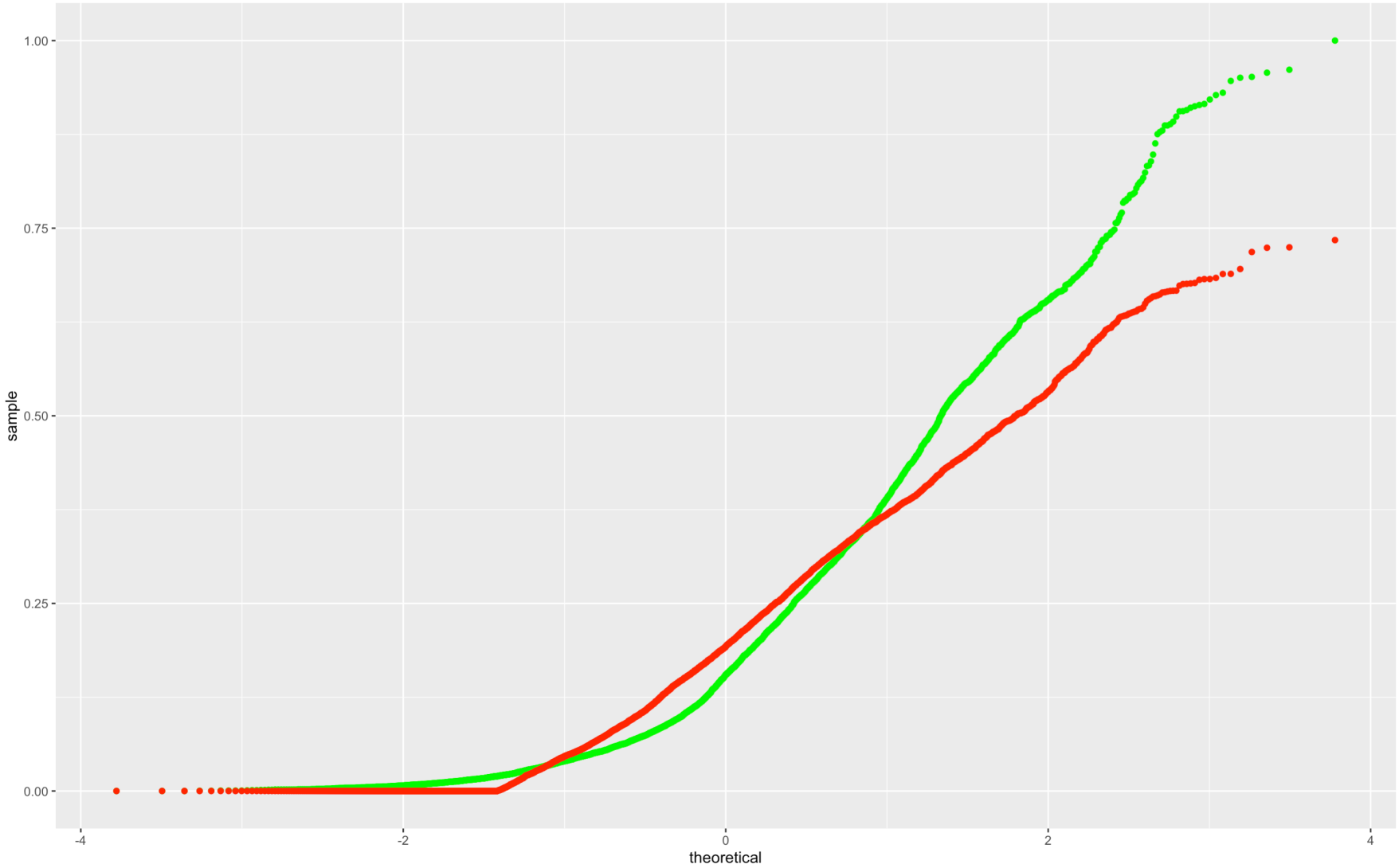
✓ model5 is the best performing model among others.

✓ The formula for the model is:

*RENTED_BIKE_COUNT ~ . + poly(TEMPERATURE, 6) + WINTER+ poly(DEW_POINT_TEMPERATURE, 6) + poly(SOLAR_RADIATION, 6) + poly(VISIBILITY, 6) + SUMMER+ TEMPERATURE * HUMIDITY + poly(HUMIDITY, 6) + RAINFALL * TEMPERATURE + SNOWFALL * TEMPERATURE + RAINFALL * HUMIDITY + SNOWFALL * HUMIDITY*

```
> rsq_model5
# A tibble: 1 × 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.770
> rmse_model5
# A tibble: 1 × 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard      0.0875
```

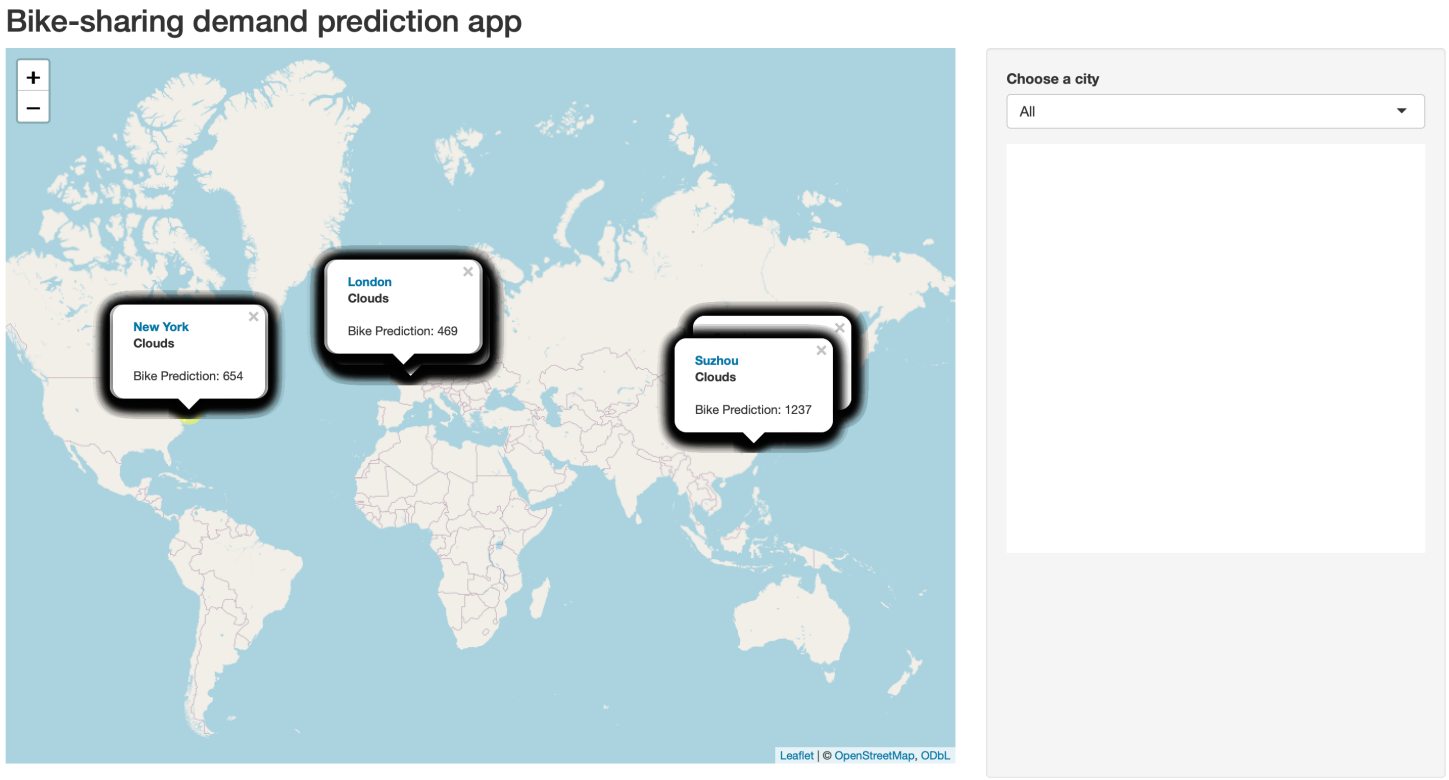
Q-Q plot of the best model



Dashboard

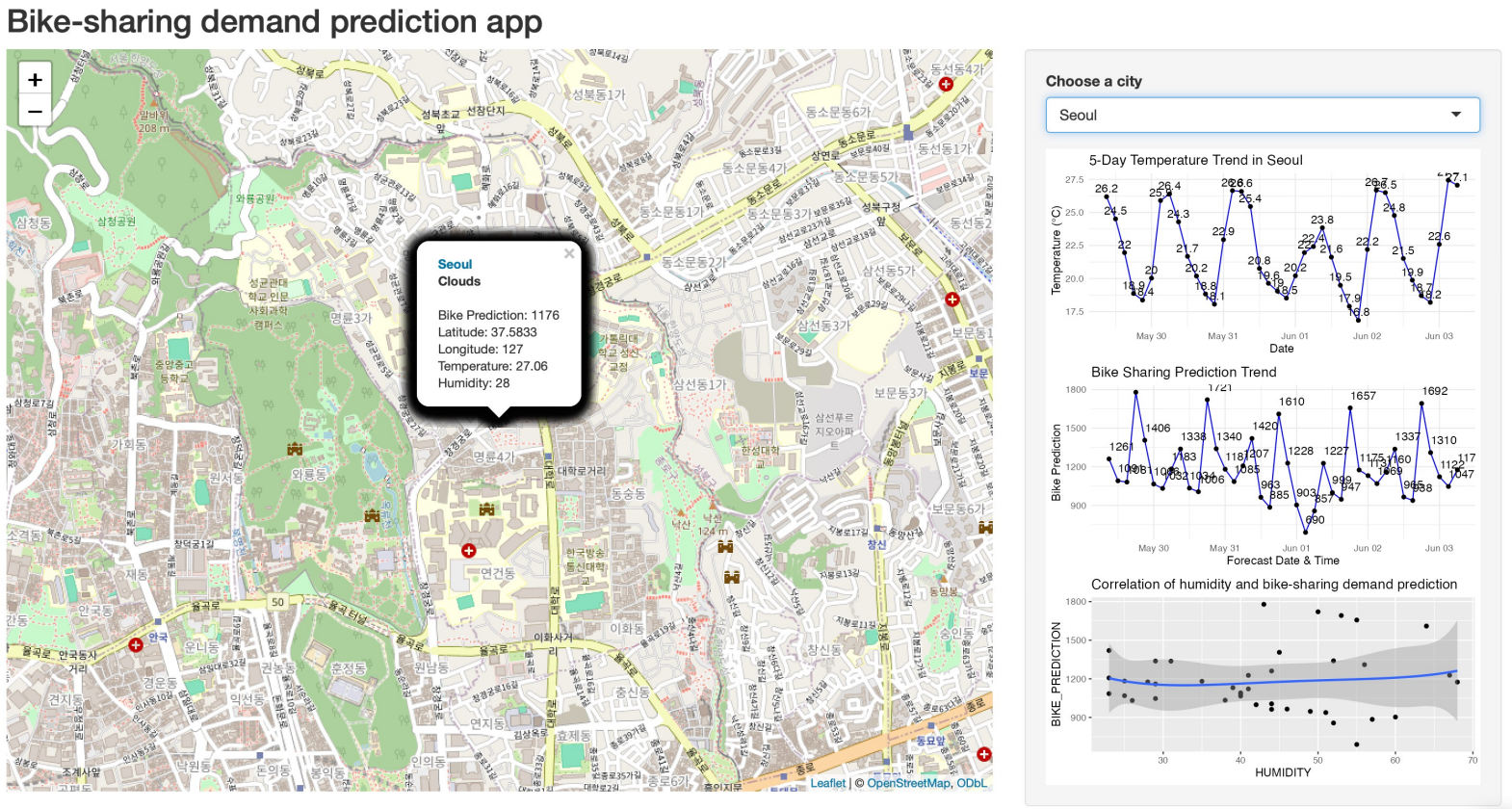
Dashboard (Bike-sharing predictions)

- ✓ The dashboard shows the bike prediction of various cities.
- ✓ When “All” is selected, the overall view of the map is only visible.
- ✓ Below the name of the city, the weather condition of the particular city is visible.



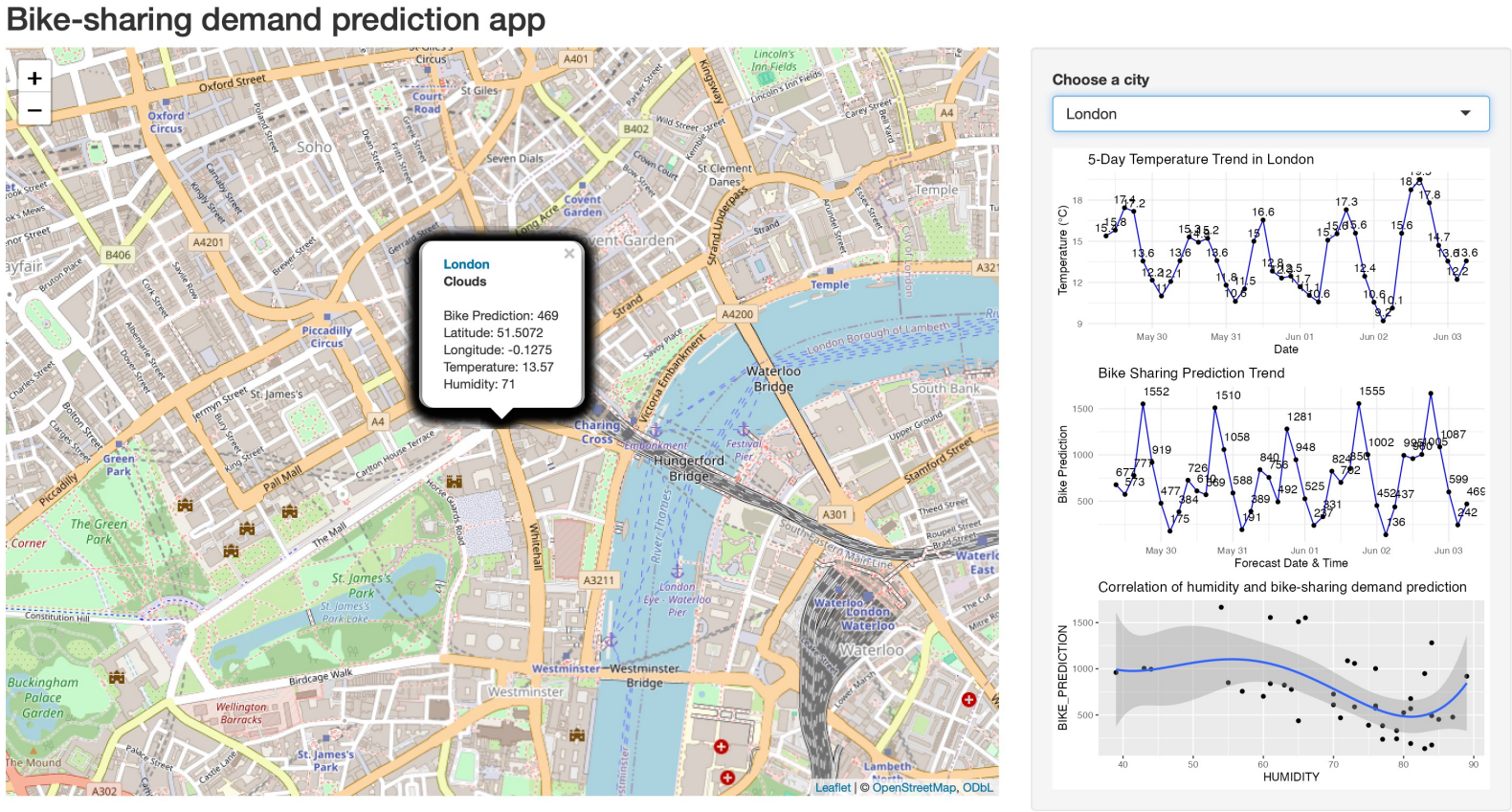
Dashboard (One City Selection)

- ✓ The dashboard shows the bike sharing demand for “Seoul” city.
- ✓ When “Seoul” is selected, the details about Seoul is shown in the map.
- ✓ Similarly, 5 day temperature trend in Seoul is also shown as line graph against Date.
- ✓ Also, Bike Sharing Prediction was shown against Forecast Date & Time.
- ✓ Correlation of humidity and bike-sharing demand prediction was also shown for the “Seoul” city.



Dashboard (Another City Selection)

- ✓ The dashboard shows the bike sharing demand for “London” city.
- ✓ When “London” is selected, the details about London is shown in the map.
- ✓ Similarly, 5 day temperature trend in London is also shown as line graph against Date.
- ✓ Also, Bike Sharing Prediction was shown against Forecast Date & Time.
- ✓ Correlation of humidity and bike-sharing demand prediction was also shown for the “London” city.



CONCLUSION

- ✓The analysis for the bike sharing demand is performed in this project.
- ✓Eventually, it can be concluded that the bike prediction is highly dependent on the various factors such as Rainfall, Humidity, Temperature, etc.
- ✓The summer season and the evening hours highly influenced the number of bike rentals. So, it is highly recommended to have the proper numbers of bikes arranged to meet the demand by the customers.

APPENDIX

For the code of the whole project, can visit the following github link:

IBM DATA SCIENCE CAPSTONE PROJECT WITH R