Applied Data Science with R Capstone project

Felipe de Oliveira

13/07/2024

Outline

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

Executive Summary

- Data contextualization and analysis goal
- Methodology description
  - Data gathering
  - Data analysis
  - Data visualizations
- Results presentation supported with graphs and trends
- Discussion of overall findings and implications regarding the results previously exposed
- Final conclusions of the carried out research

Introduction

- Analyzing how weather affects demand for bike sharing in urban areas
- Collecting and processing data related to weather and bike share demand from multiple sources
- Rental bikes are available in many cities around the world. It is important that each of these cities provides a reliable supply of rental bicycles to optimize availability and accessibility to the public at all times.
- Understanding the influence of weather on demand for bicycles rentals

Methodology

Data collection

- Web scraping was used to extract the table from Wiki Global Bike-Sharing Systems and related techniques to store the table in csv format. The second set of data was selected through the OpenWeather API, creating a call and converting the API data into a csv table

Web scraping → Extract the HTML table → Convert from HTML to Data frame → Exporting to CSV

Getting current weather data → Creating dataframe → Getting 5-day weather forecasts → Exporting to CSV

Data wrangling

- First with the help of regex, column names were standardized for all collected data sets. We then remove the unwanted reference links from the scraped bike-sharing systems dataset, and finish by extracting only the numerical value of unwanted text annotations. With the dplyr package, we detect and manipulate missing values, create indicator (dummy) variables for categorical variables, and normalize the data.

Standardize column names for all collected datasets → Remove undesired reference links → Extract only the numeric value from undesired text annotations

Detect and handle missing values → **Create indicator (dummy) variables** → **Normalize data**

EDA with SQL

- Using SQL queries with the RSQLite R package
- Record Count
- Operational Hours
- Weather Outlook
- Seasons
- Total Bike Count and City Info for Seoul
- Hourly popularity and temperature by season
- Rental Seasonality

EDA with data visualization

- Recast the date column as a date
- Cast hours as a categorical variable
- Dataset Summary
- Calculating how many holidays there are
- Calculating the percentage of records that fall on a holiday
- Given the observations for the 'FUNCTIONING_DAY' how many records must there be?

Predictive analysis

- I started by splitting data into training and testing datasets, then built a linear regression model using only the weather variables. With this, I created a linear regression model using climate and date variables, and evaluated the models for the important variables. I decided to refine the model, adding polynomial terms, interaction terms, regularization terms and experimented to look for improved models.

Split data into training and testing datasets → **Build a linear regression model using only the weather variables** → **Build a linear regression model**

Add polynomial terms → **Add interactions terms** → **Add regularizations terms**

Build a R Shiny dashboard

- I added a base map of cycling maximum forecast overview
- I added a selection entry (dropdown) to select a specific city
- I added a static temperature trendline
- Added an interactive bike share demand forecast trendline
- I added a static humidity demand forecast correlation chart and bike sharing

Results

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

EDA with SQL

Busiest bike rental times

A data.frame: 1 × 3

DATE	HOUR	MAXIMUM_COUNT
<chr>	<int>	<int>
19/06/2018	18	3556

- By creating a subquery, we were able to determine that on 06/19/2018, for 18 hours there were 3556 bicycle rentals.

Hourly popularity and temperature by seasons

A data.frame: 10 × 4

SEASONS	HOUR	AVG(RENTED_BIKE_COUNT)	AVG(TEMPERATURE)
<chr>	<int>	<dbl>	<dbl>
Summer	18	2135.141	29.38791
Autumn	18	1983.333	16.03185
Summer	19	1889.250	28.27378

- With the analysis, we noticed a preference for summer, with very high numbers. Autumn is not far behind in second place with numbers very close

Rental Seasonality

A data.frame: 4 × 5

SEASONS	AVG_S_COUNT	MIN_S_COUNT	MAX_S_COUNT	DETOUR_S_COUNT
<chr>	<dbl>	<int>	<int>	<dbl>
Summer	1034.0734	9	3556	690.0884
Autumn	924.1105	2	3298	617.3885
Spring	746.2542	2	3251	618.5247
Winter	225.5412	3	937	150.3374

- As mentioned previously, the numbers in summer are very high compared to other seasons. Indicating a greater willingness to rent bicycles in high temperatures.

Weather Seasonality

SEASONS	AVG_S_COUNT	AVG_S_TEMP	AVG_S_HUMIDITY	AVG_WIND_SPEED	AVG_VISIBILITY	AVG_DEW_POINT	AVG_SOLAR_RADIATION	AVG_RAINFALL	AVG_SNOWFALL
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Summer	1034.0734	26.587711	64.98143	1.609420	1501.745	18.750136	0.7612545	0.25348732	0.00000000
Autumn	924.1105	13.821580	59.04491	1.492101	1558.174	5.150594	0.5227827	0.11765617	0.06350026
Spring	746.2542	13.021685	58.75833	1.857778	1240.912	4.091389	0.6803009	0.18694444	0.00000000
Winter	225.5412	-2.540463	49.74491	1.922685	1445.987	-12.416667	0.2981806	0.03282407	0.24750000

- With the help of the image, we can observe in more detail how the weather affects people's willingness to rent bicycles. With an average temperature of 26 in summer and relatively high humidity

Bike-sharing info in Seoul

A data.frame: 1 × 6

BICYCLES	CITY	COUNTRY	LAT	LNG	POPULATION
<int>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
20000	Seoul	South Korea	37.5833	127	21794000

- We can see that 20k bicycles available to more than 20 million people in Seoul is a very low number. We can assume here that demand is not high, perhaps the rainy weather could be one of the factors behind the low demand for bicycle rentals

Cities similar to Seoul

A data.frame: 9 × 6

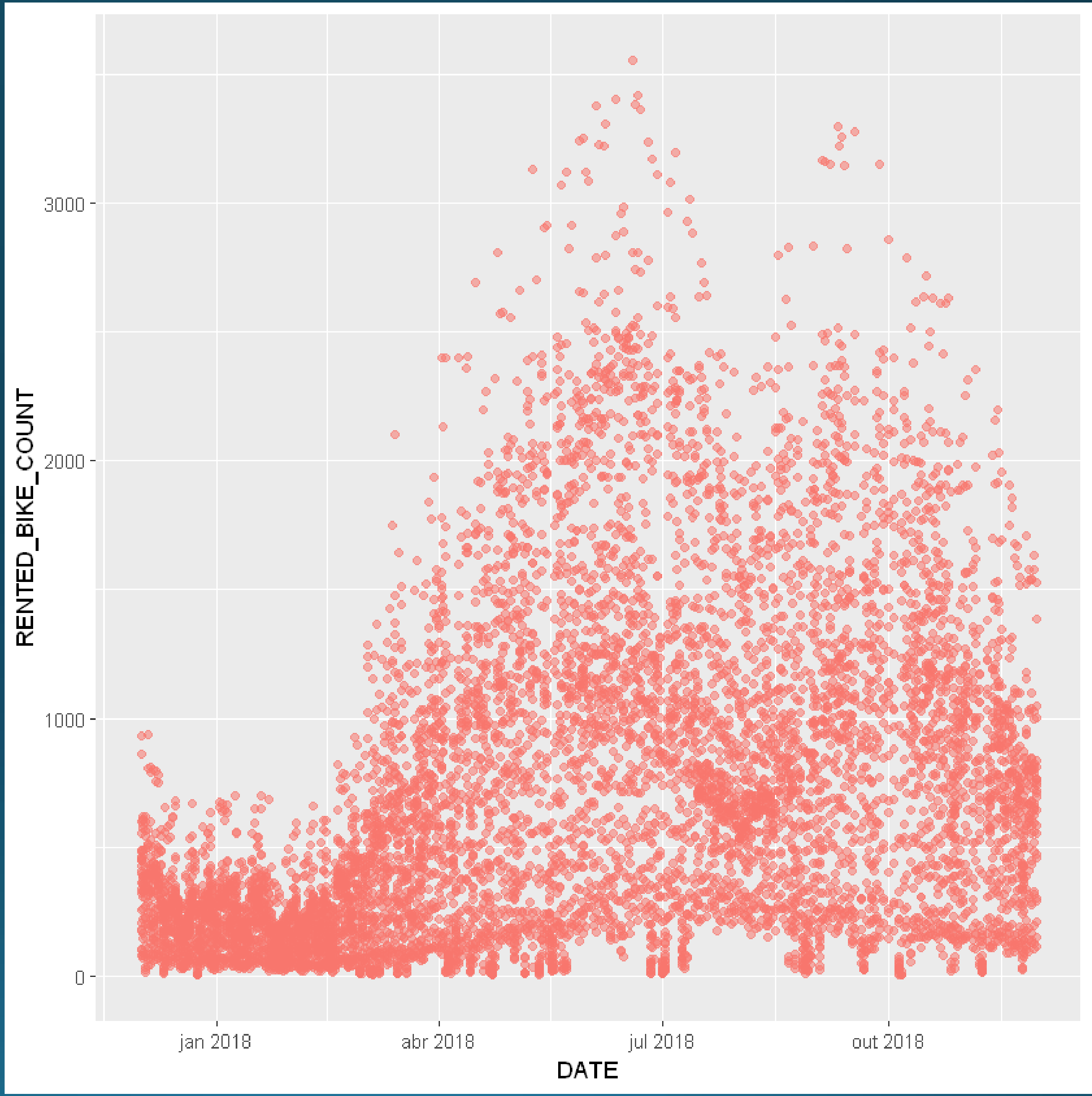
BICYCLES	CITY	COUNTRY	LAT	LNG	POPULATION
<int>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
20000	Kunshan	China	NA	NA	NA
20000	Weifang	China	36.7167	119.1000	9373000
20000	Xi'an	China	34.2667	108.9000	7135000
20000	Zhuzhou	China	27.8407	113.1469	3855609
20000	Seoul	South Korea	37.5833	127.0000	21794000
19165	Shanghai	China	31.1667	121.4667	22120000

- We can observe that the number of bicycles available is a preference adopted by East Asia.

EDA with Visualization

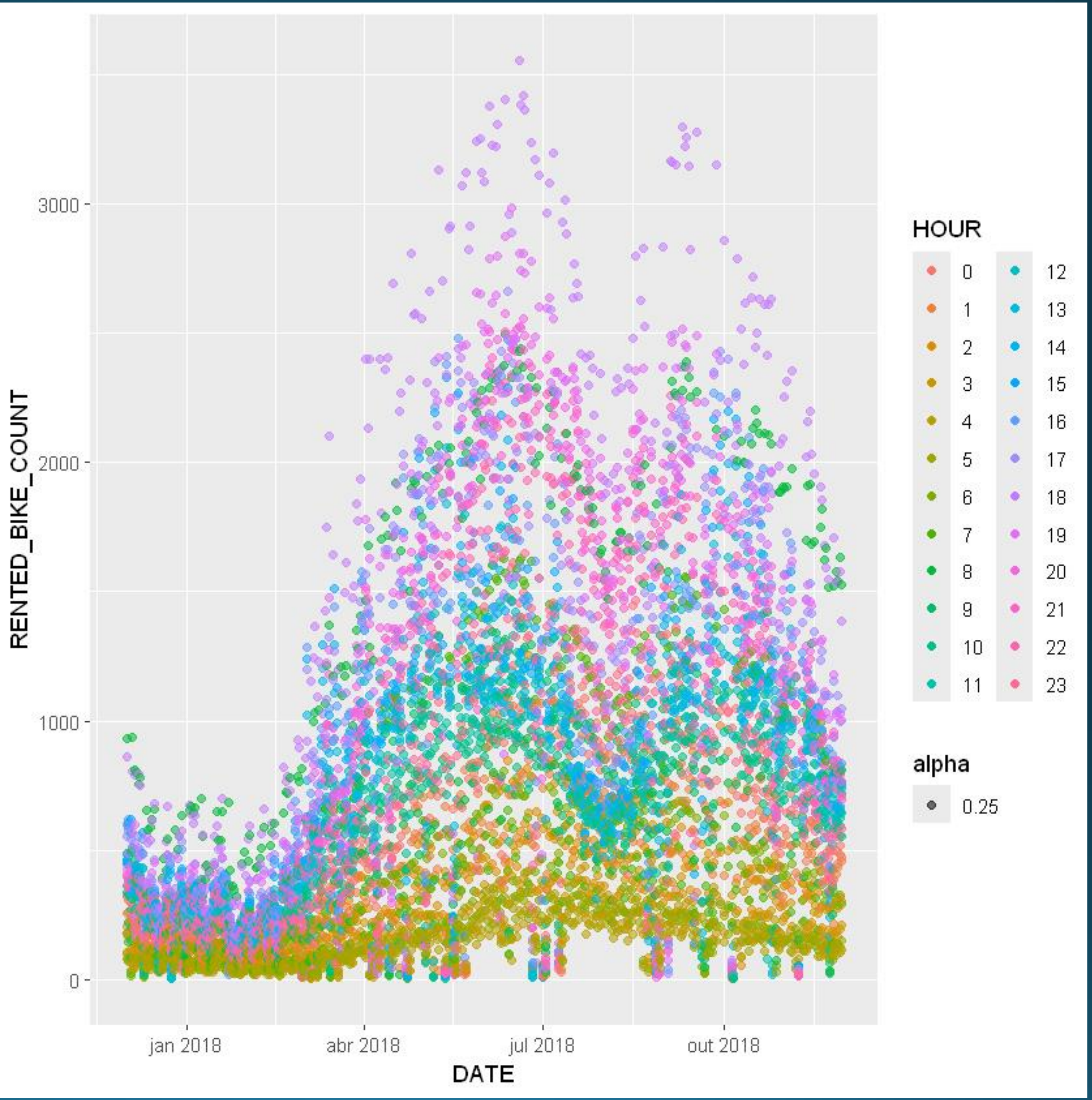
Bike rental vs. Date

We can observe an increase that started around February and March, with a peak between May and July. A considerable drop can be observed starting in August, increasing again in September and decreasing at the end of the year.



Bike rental vs. Datetime

We can observe a very low preference for renting bikes in the early hours of the morning. Now, a considerable increase begins to emerge throughout the day, with its maximum at dusk between 6 and 7.



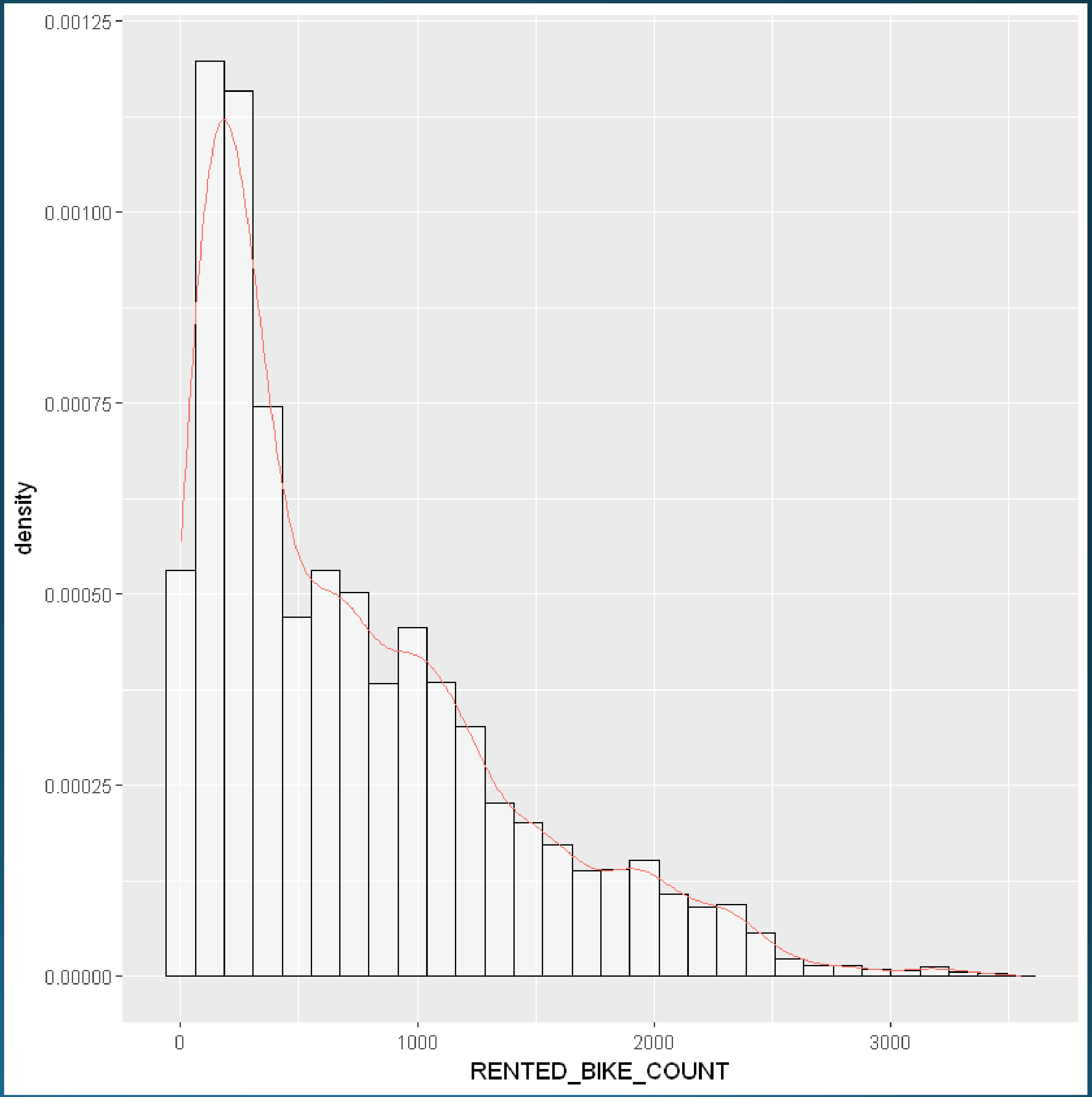
Bike rental histogram

Consider what it's shape tells you, and keep your findings for your presentation in the final project.

We can see from the histogram that there are relatively few rented bikes. The highest frequency of rented bicycles is about 250.

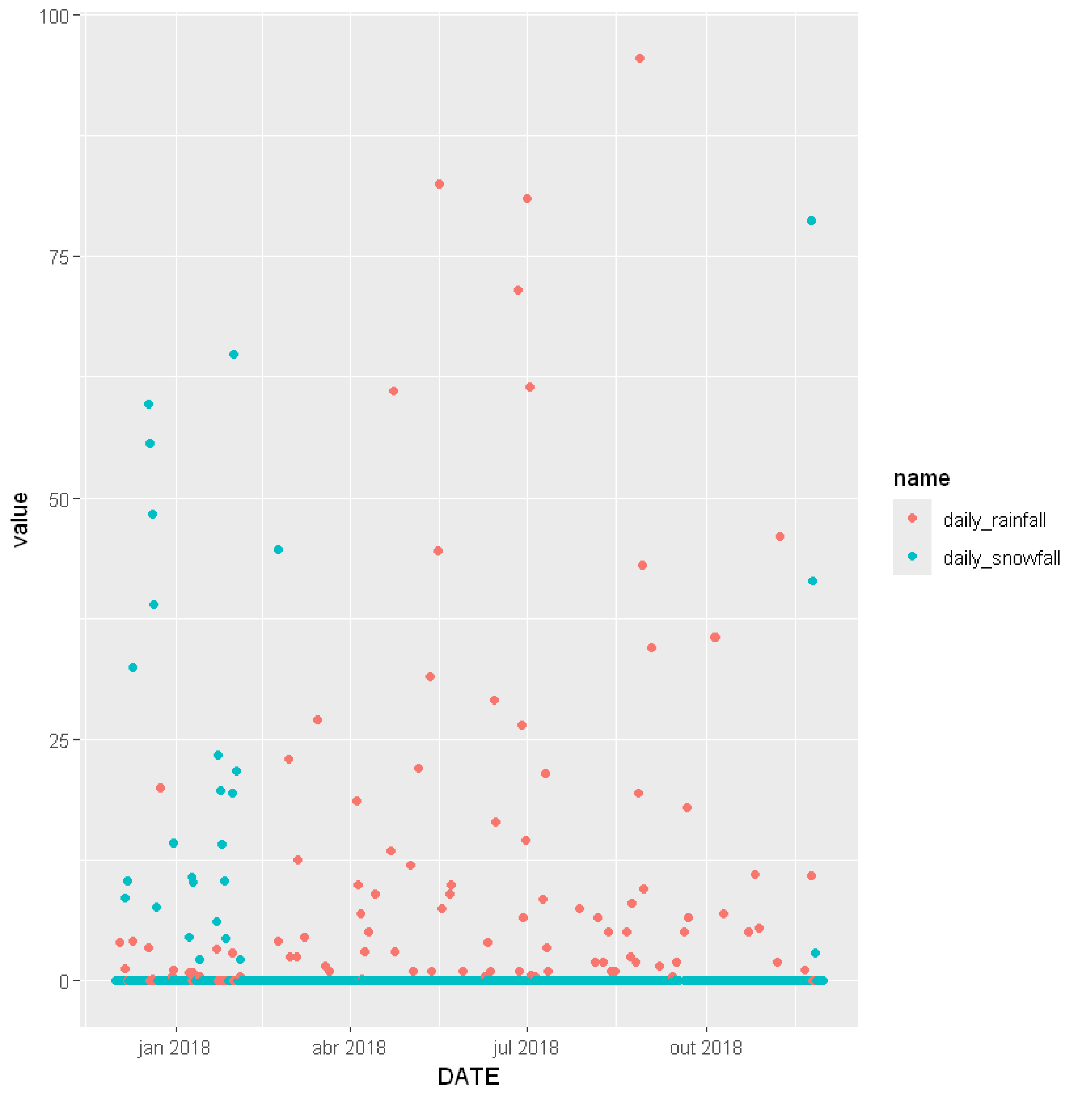
Judging by the "bumps" in about 700 to 3200 bikes, there may be hidden factors that are influencing the data.

It is interesting to analyze that judging by the distribution tail, on rare occasions there are many more rented bicycles than usual.



Daily total rainfall and snowfall

We can observe that between January and February there was a concentration of rain and snow. We can use it as a correlation parameter with the numbers of rented bicycles.



Predictive analysis

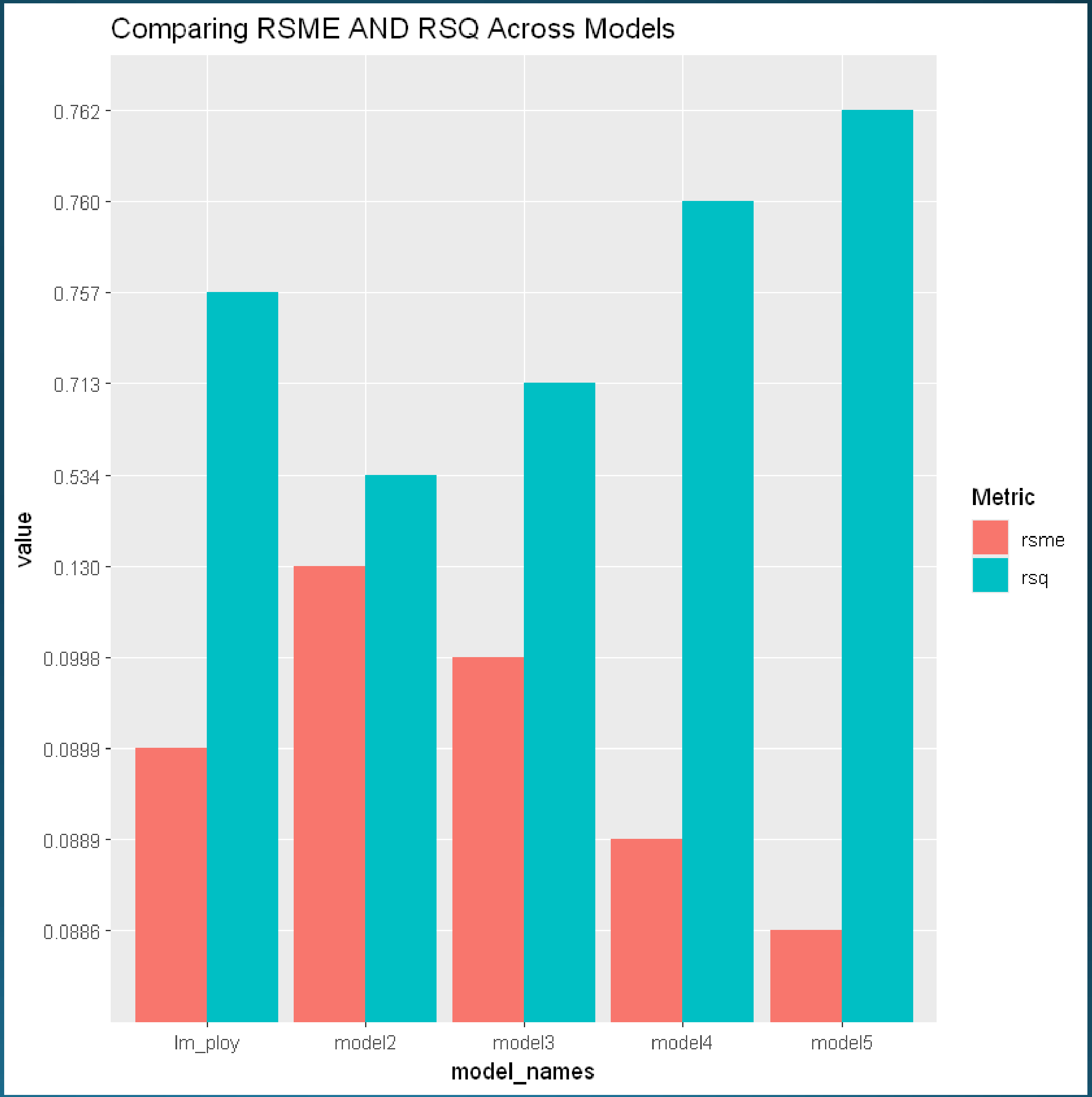
Ranked coefficients

As you can imagine, weather conditions can affect people's bike rental decisions. For example, on a cold and rainy day, you can choose alternative transportation, such as a bus or taxi. While on a nice bright day, you may want to rent a bike for a short distance trip. So, can we predict a city's bike share demand based on its local weather information? We tried to build a regression model to do this.

```
Coefficients:
        (Intercept)         TEMPERATURE            HUMIDITY
           0.059144            0.220219           -0.249502
         WIND_SPEED          VISIBILITY DEW_POINT_TEMPERATURE
           0.008979            0.006154            0.168370
    SOLAR_RADIATION            RAINFALL            SNOWFALL
           0.077907           -0.580933            0.073431
             AUTUMN              SPRING              SUMMER
           0.101013            0.053845            0.055752
             WINTER             HOLIDAY          NO_HOLIDAY
                 NA           -0.035009                  NA
                `0`                 `1`                `10`
          -0.008244           -0.032878           -0.066831
               `11`                `12`                `13`
          -0.069607           -0.058622           -0.053842
               `14`                `15`                `16`
          -0.054148           -0.030876            0.006508
               `17`                `18`                `19`
           0.085973            0.223636            0.147155
                `2`                `20`                `21`
          -0.066745            0.121552            0.125656
               `22`                `23`                 `3`
           0.096410            0.029209           -0.090003
                `4`                 `5`                 `6`
          -0.108692           -0.102060           -0.057434
                `7`                 `8`                 `9`
           0.030039            0.126893                  NA
```

Model evaluation

- Built at least 5 different models using polynomial terms, interaction terms, and regularizations
- Visualize the refined models' RMSE and R-squared using grouped bar chart



Find the best performing model

- Select the best performing model with:
  - RMSE must be less than 330
  - R-squared must be larger than 0.72

```
model5 <- linear_reg(penalty = 0.0015, mixture = 0.2) %>%
  set_engine("glmnet")
model5_fit <- model5 %>%
  fit(RENTED_BIKE_COUNT ~ . + poly(TEMPERATURE, 6) + WINTER * `18` +
      poly(DEW_POINT_TEMPERATURE, 6) + poly(SOLAR_RADIATION, 6) +
      poly(VISIBILITY, 6) + SUMMER * `18` + TEMPERATURE * HUMIDITY +
      poly(HUMIDITY, 6) + RAINFALL * TEMPERATURE + SNOWFALL * TEMPERATURE +
      RAINFALL * HUMIDITY + SNOWFALL * HUMIDITY, data = bike_sharing_training)

model5_train_results <- model5_fit %>%
  predict(new_data = bike_sharing_training) %>%
  mutate(truth = bike_sharing_training$RENTED_BIKE_COUNT)

model5_train_results$.pred <- replace(model5_train_results$.pred, model5_train_results$.pred < 0, 0)

# Save their rmse and rsq values
rsq_model5 <- rsq(model5_train_results,
  truth = truth,
  estimate = .pred
)

rmse_model5 <- rmse(model5_train_results,
  truth = truth,
  estimate = .pred
)
```

A tibble: 1 × 3

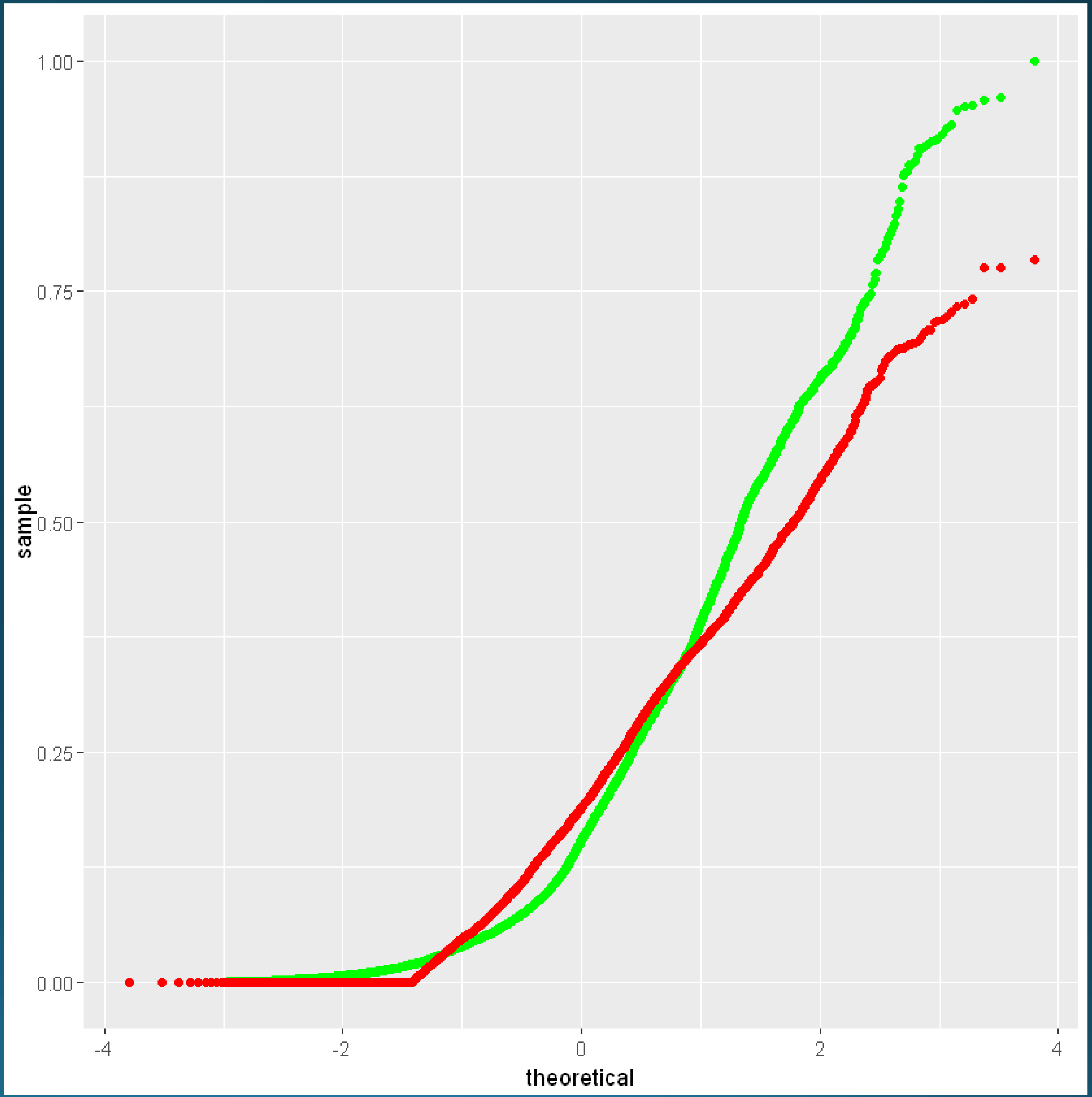
.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
rsq	standard	0.7753122

A tibble: 1 × 3

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
RMSE	standard	0.08643008

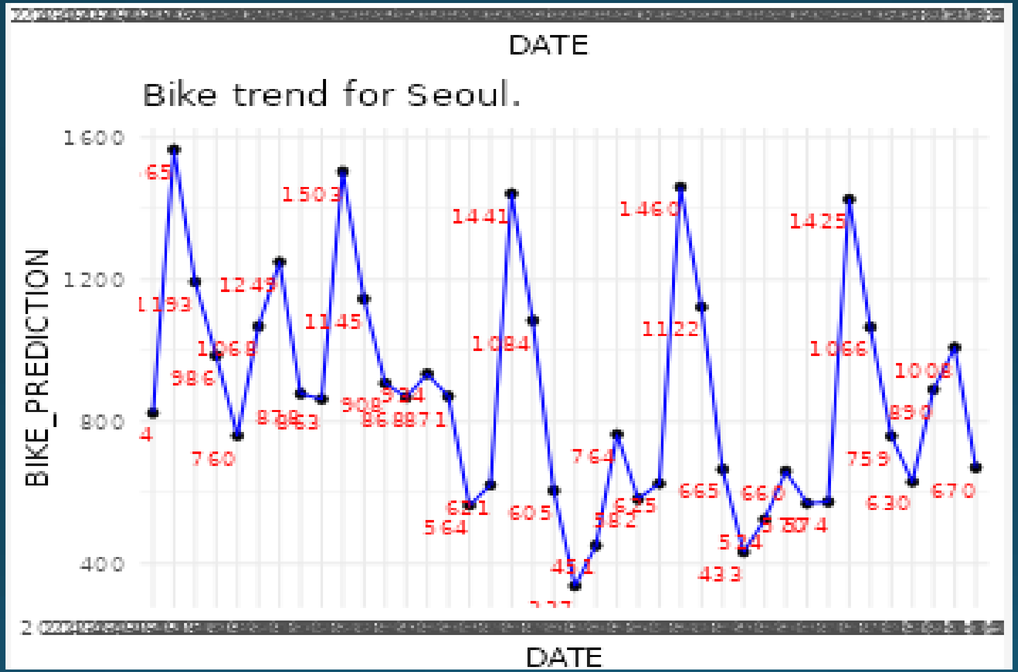
Q-Q plot of the best model

Plot the Q-Q plot of the best model's test results vs the truths



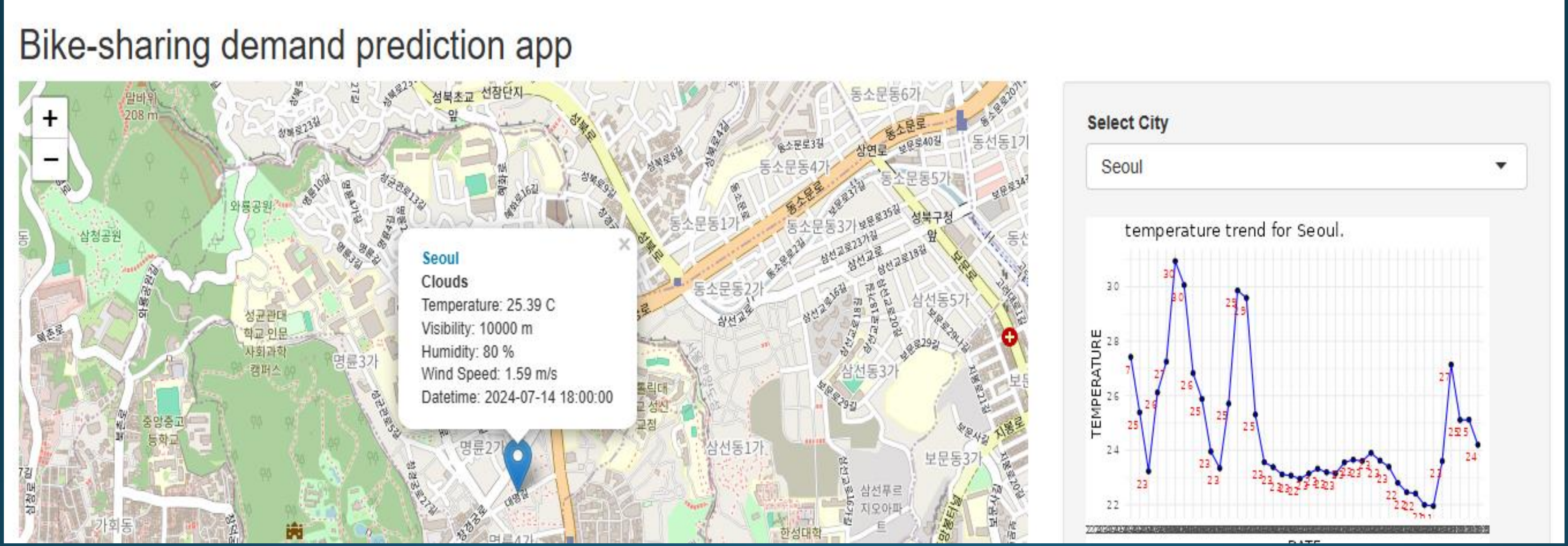
Dashboard

Dashboard: Max bike-sharing prediction



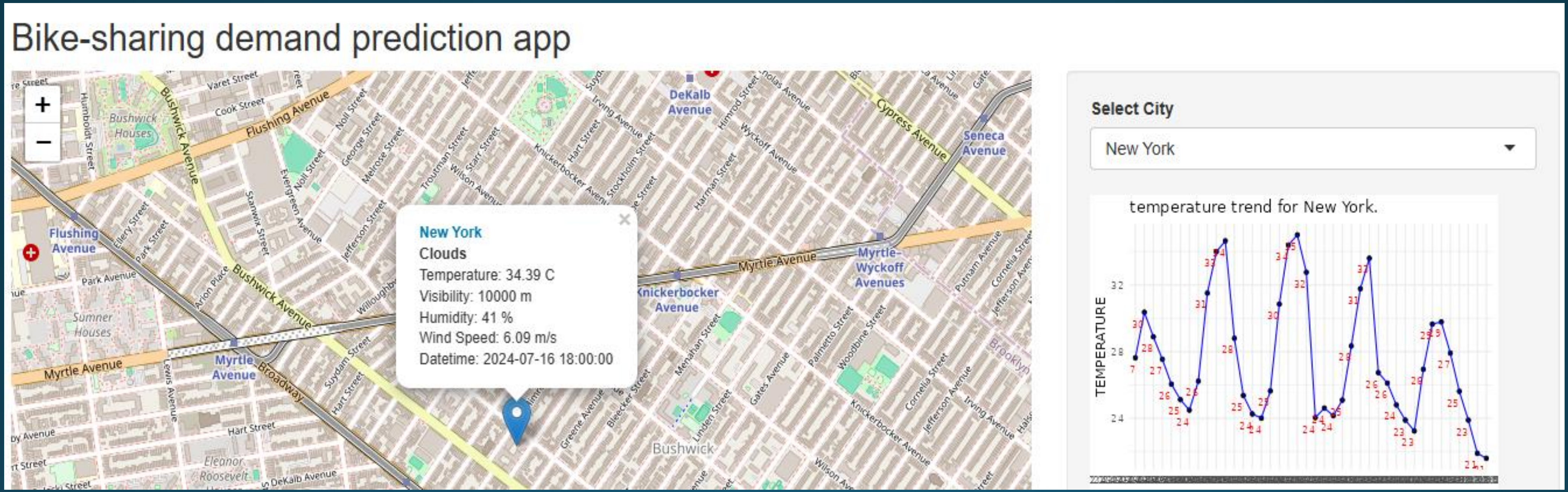
- We can see from the daily 'ripples' that people are probably alternating between using cars and bicycles for transportation

Dashboard: City is selected



- With the city selected we can have a summary of statistics related to it.

Dashboard screenshot 3

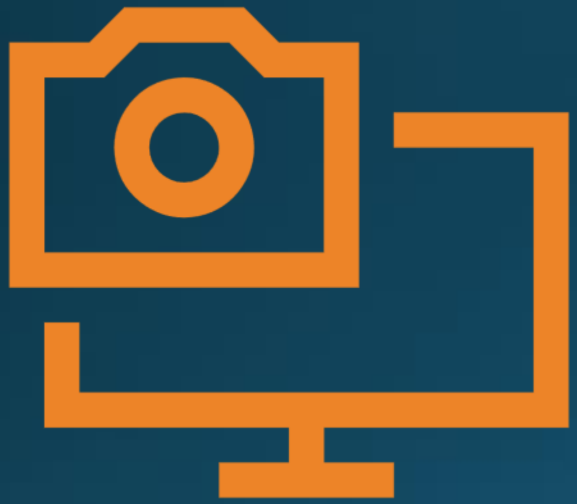


- Selecting New York and getting some statistics

CONCLUSION

- Climatic factors affect people's willingness to use bicycles.
- There is a low number of bicycles available compared to the population. Perhaps a health or environmental awareness campaign is necessary.
- The seasons have a certain preference in people's search for bicycles. We can observe a large concentration in the summer and autumn periods.
- We can see that people are probably alternating between using automobiles and bicycles for transportation. Perhaps a greater concentration of bicycles in urban areas all the way to commercial areas would be a good move.

APPENDIX



- Include any relevant assets like R code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

OpenWeatherAPI and Webscrping

[2]:
```r
url <- "https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems"
# Get the root HTML node by calling the `read_html()` method with URL
# Obtenha o nó HTML raiz chamando o método `read_html()` com URL
root_node <- read_html(url)
root_node
```

```
{html_document}
<html>
[1] <body><p>"COUNTRY","City","Name","SYSTEM","OPERATOR","LAUNCHED","DISCONTI ...
```

[4]:
```r
table_nodes <- html_nodes(root_node, "table")

for(table in table_nodes) {
    print(table)
}
```

[1]:
```r
# URL for Current Weather API
current_weather_url <- 'https://api.openweathermap.org/data/2.5/weather'
```

Em seguida, vamos criar uma lista para armazenar parâmetros de URL para a API de clima atual

[3]:
```r
# need to be replaced by your real API key
your_api_key <- "d07f56210c6fbabf3719ea01e5712c2f"
# Input `q` is the city name
# Input `appid` is your API KEY,
# Input `units` are preferred units such as Metric or Imperial
current_query <- list(q = "Seoul", appid = your_api_key, units="metric")
```

Regular expressions, missing values handling and generating indicator columns

TODO: Write a loop to iterate over the above datasets and convert their column names `for`

```
for (dataset_name in dataset_list){
    # Ler conjunto de dados
    dataset <- read_csv(dataset_name)
    # Padronizou suas colunas:

    # Converte todos os nomes de colunas para letras maiúsculas
    names(dataset) <- toupper(names(dataset))
    # Substitua quaisquer separadores de espaço em branco por sublinhados, usando a função str_replace_all
    names(dataset) <- str_replace_all(names(dataset), " ", "_")
    # Salve o conjunto de dados
    write.csv(dataset, dataset_name, row.names=FALSE)
}
```

```
16]: # Convert SEASONS, HOLIDAY, FUNCTIONING_DAY, and HOUR co
col <- c("SEASONS", "HOLIDAY", "HOUR")

for (column in col) {
    bike_sharing_df <- bike_sharing_df %>%
        mutate(dummy = 1) %>%
        spread(key = column, value = dummy, fill = 0)
}
```

```
17]: # Print the dataset summary again to make sure the indic
summary(bike_sharing_df)
```

WHOLE: Drop rows with missing values in the column `RENTED_BIKE_COUNT`

```
[7]: # Drop rows with `RENTED_BIKE_COUNT` column == NA
bike_sharing_df <- drop_na(bike_sharing_df, RENTED_BIKE_COUNT)
```

```
[8]: # Print the dataset dimension again after those rows are dropped
dim(bike_sharing_df)
```

8465 · 14

Screenshots of all required SQL queries

Total Bike Count and City Info for Seoul

```
# provide your solution here
dbGetQuery(conn, "SELECT B.BICYCLES, B.CITY, B.COUNTRY,
          W.LAT, W.LNG, W.POPULATION
          FROM BIKE_SHARING_SYSTEMS AS B
          LEFT JOIN WORLD_CITIES AS W ON B.CITY = W.CITY_ASCII
          WHERE B.CITY = 'Seoul'")
```

A data.frame: 1 × 6

BICYCLES	CITY	COUNTRY	LAT	LNG	POPULATION
<int>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
20000	Seoul	South Korea	37.5833	127	21794000

Hourly popularity and temperature by season

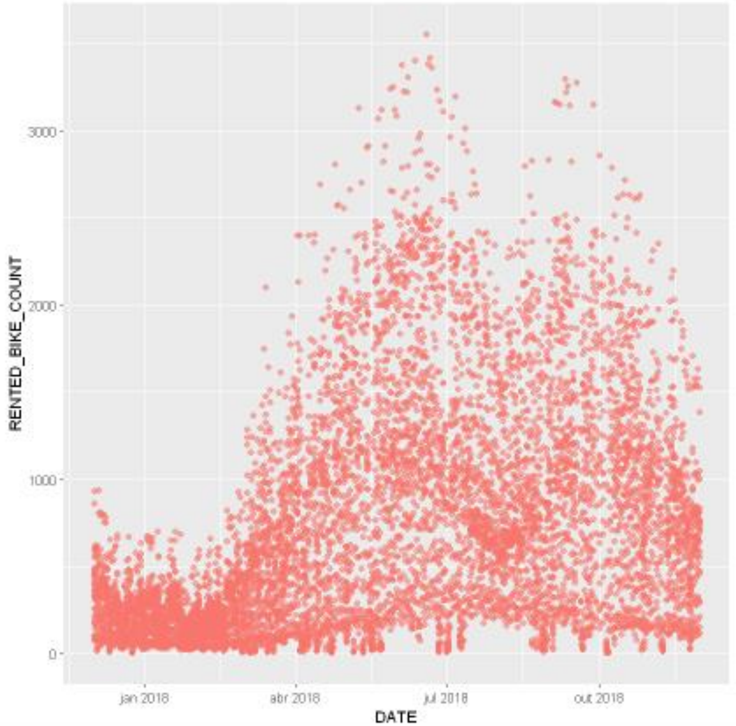
```
# provide your solution here
dbGetQuery(conn, "SELECT SEASONS, HOUR, AVG(RENTED_BIKE_COUNT), AVG(TEMPERATURE)
          FROM SEOUL_BIKE_SHARING
          GROUP BY SEASONS, HOUR
          ORDER BY AVG(RENTED_BIKE_COUNT) DESC
          LIMIT 10")
```

A data.frame: 10 × 4

SEASONS	HOUR	AVG(RENTED_BIKE_COUNT)	AVG(TEMPERATURE)
<chr>	<int>	<dbl>	<dbl>
Summer	18	2135.141	29.38791
Autumn	18	1983.333	16.03185
Summer	19	1889.250	28.27378
Summer	20	1801.924	27.06630
Summer	21	1754.065	26.27826
Spring	18	1689.311	15.97222
Summer	22	1567.870	25.69891
Autumn	17	1562.877	17.27778
Summer	17	1526.293	30.07691
Autumn	19	1515.568	15.06346

Adding screenshots of your ggplot code snippets

```
[40]: # provide your solution here
      ggplot(seoul_bike_sharing, aes(x = DATE, y = RENTED_BIKE_COUNT,
                                     color = "blue", alpha = 0.25)) +
      geom_point() +
      theme(legend.position = "none")
```




```
[43]: # provide your solution here
      ggplot(seoul_bike_sharing, aes(x = DATE, y = RENTED_BIKE_COUNT,
                                     color = HOUR, alpha = 0.25)) +
      geom_point()
```