Applied Data Science with R Capstone Project: Seoul Bike Sharing Demand Prediction

Rajdeep Sandhu

15 September 2023

Outline

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

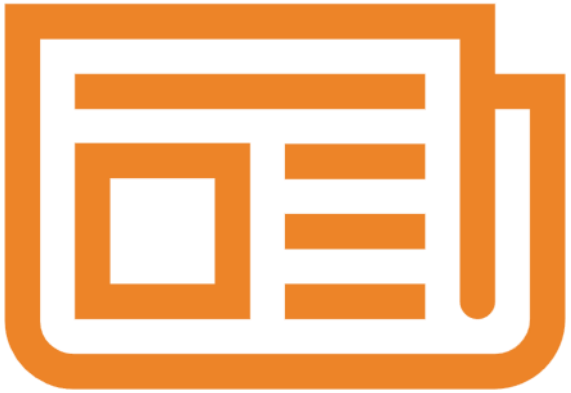
Executive Summary

- In Seoul, a total of 20,000 bikes serve a population of almost 21.8 million.
- Bike sharing demand
  - Highest in the evenings, followed by mornings, possibly due to commuting
  - Highest in summer.
  - Spikes in June and September, possibly due to vacations and tourism
  - Low with high humidity.
- A small proportion of bikes is in use with occasional spikes.
- Important predictors
  - Rainfall, humidity and temperature.
  - Season and hour.
- Higher order polynomial regression performs the best for demand prediction.

Introduction

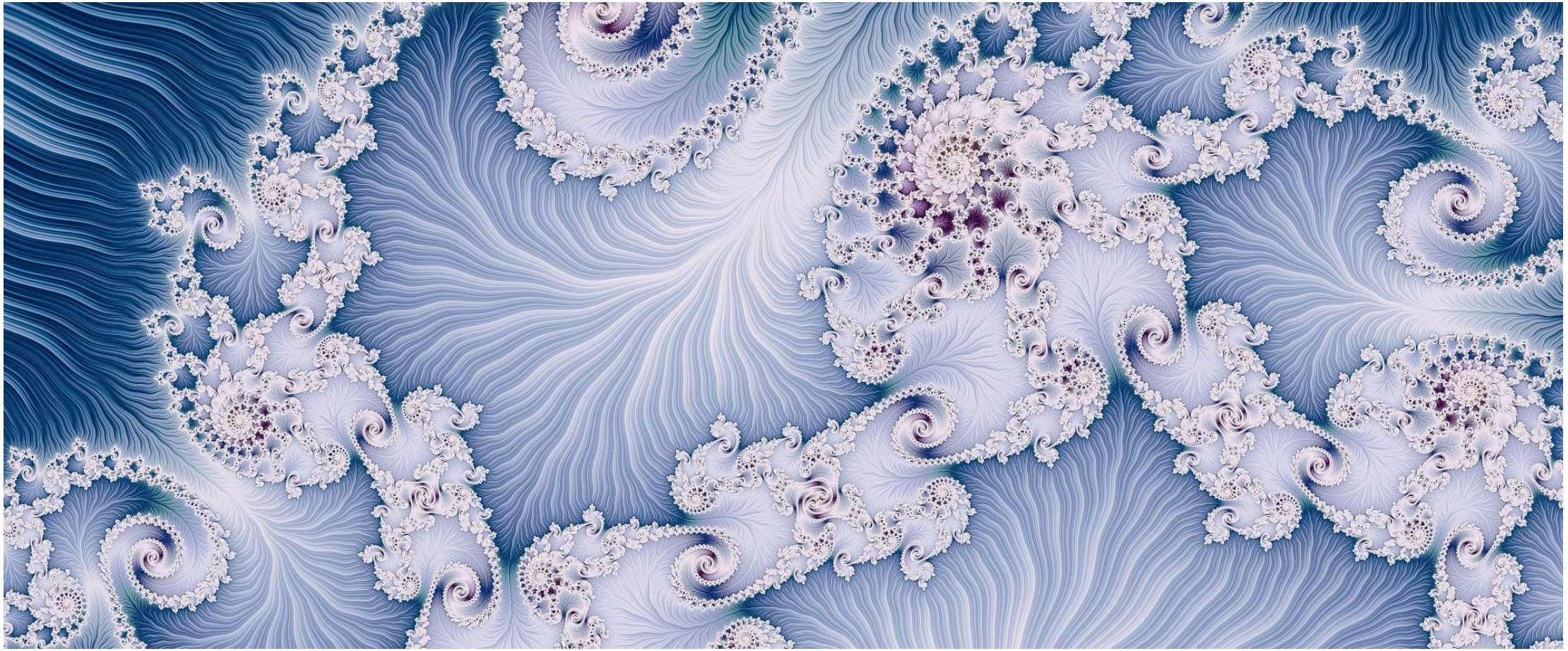
- Background
  - Several bike sharing schemes worldwide
  - Optimal, accessible and reliable supply needed at all times.
  - Cost minimization by minimizing supply to meet demand
- Aim: Weather data analysis to predict Bike Sharing demand
  - To predict the number of bikes required each hour of the day
  - Based on current conditions such as the weather

Methodology



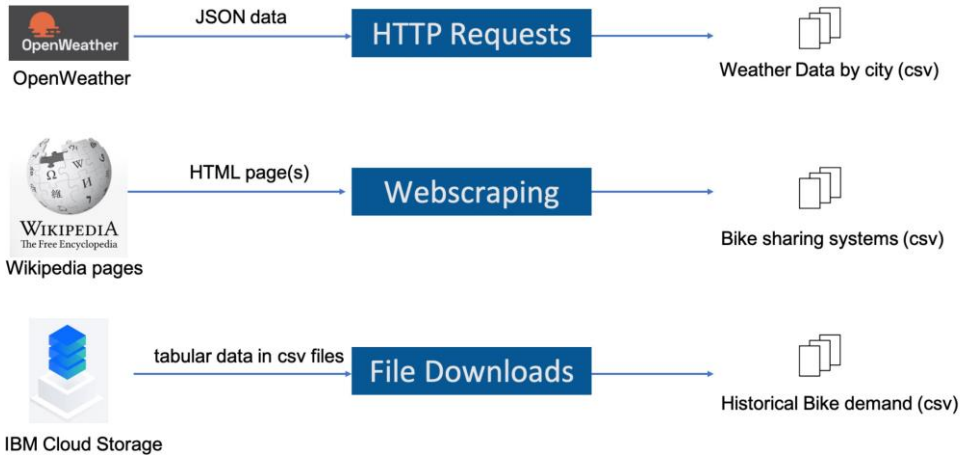
- Perform data collection
- Perform data wrangling
- Perform exploratory data analysis (EDA) using SQL and visualization
- Perform predictive analysis using regression models
  - How to build the baseline model
  - How to improve the baseline model
- Build a R Shiny dashboard app

Methodology



Data collection

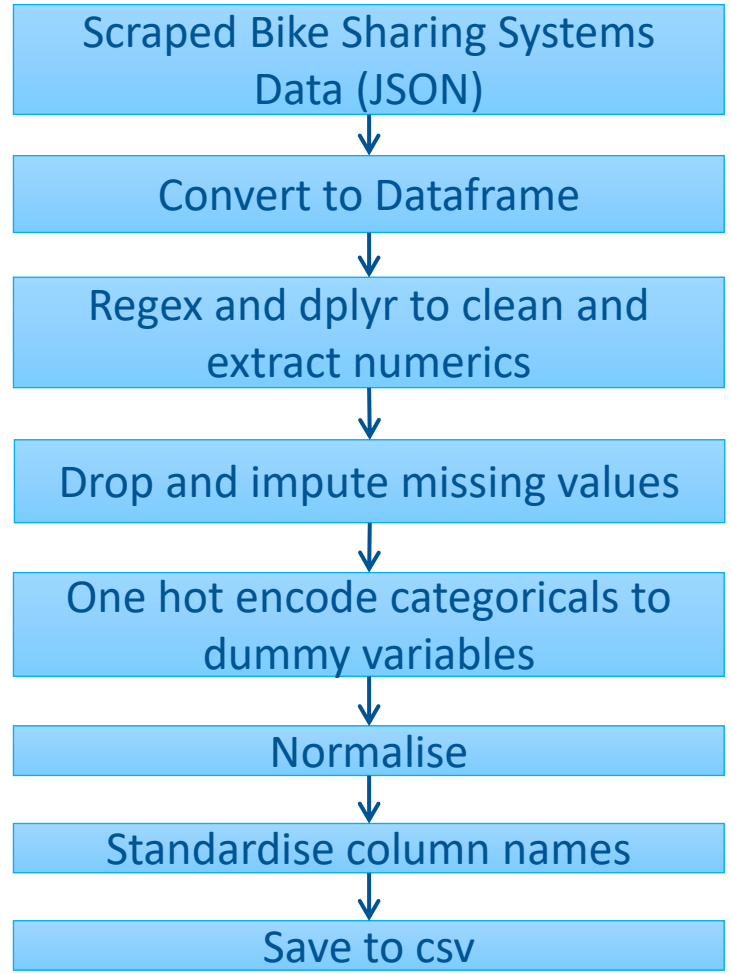
- Current and forecasted OpenWeather Data using REST API to csv.
- Wikipedia Bike Sharing Systems Page webscraped for worldwide bike sharing systems data to csv
  - https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems
- Aggregated data (.csv) downloaded from IBM Cloud Storage
  - Seoul Bike Sharing Demand Data Set
    - Weather and bikes rentals per hour and date
  - The Global Bike Sharing Cities Dataset
  - World Cities Data
    - Geolocation data for major cities worldwide



Credit: IBM Data Analysis for R Course

Data Preprocessing

- Web-scraped Wikipedia Bike Sharing Systems data
  - Dataframe from JSON
  - Regex and dplyr
    - To remove extraneous text, characters, links, inconsistent formatting
    - Extract numerics
  - Missing values handled
  - Categorical variables converted into indicator variables
  - Normalisation applied



Data Preprocessing

- Seoul bike sharing historical demand dataset
  - Using Tidyverse
    - Missing values dropped from RENTED_BIKE_COUNT
      - As these are a small proportion of dataset
      - This cannot have missing values as a response variable
    - Average imputed for summer TEMPERATURE
      - Important predictor variable.
    - SEASONS, HOLIDAY, and HOUR converted into dummy (indicator) variables (one hot encoding)
    - FUNCTIONING_DAY has not converted (as no categorical data)
    - min-max normalization applied to numerical columns
- Column names standardised for all datasets
- All data written to CSV

EDA with SQL

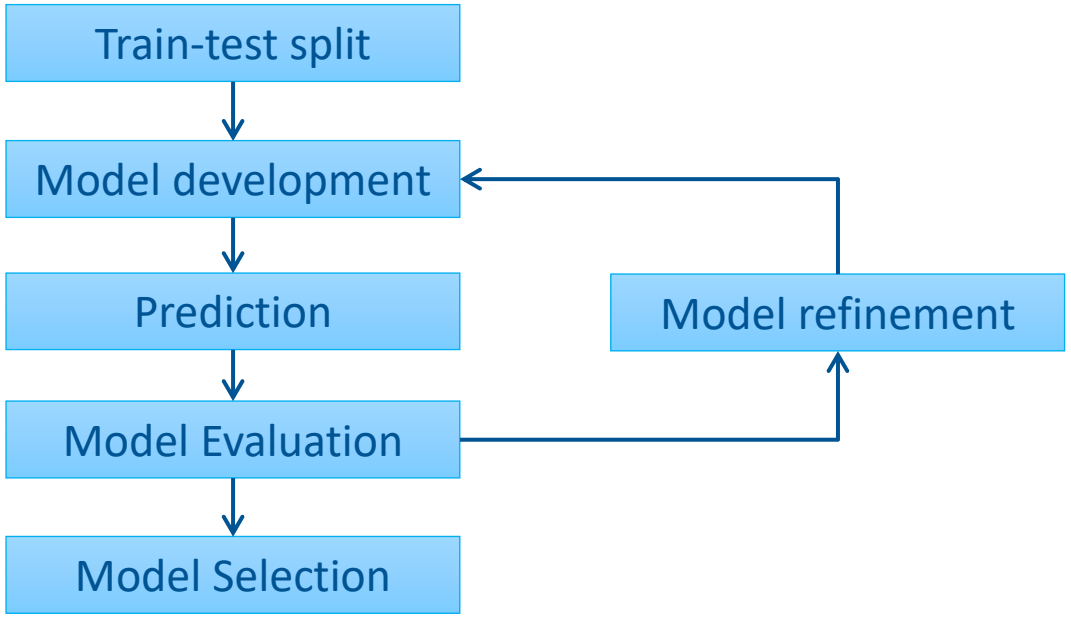
- The following SQL queries were performed
  - Number of records in the seoul_bike_sharing dataset
  - Number of hours with non-zero rented bike count.
  - Weather forecast for Seoul over the next 3 hours.
  - Seasons included in the seoul_bike_sharing dataset.
  - First and last dates in the seoul_bike_sharing dataset.
  - Busiest bike rental times
  - Hourly popularity and temperature by seasons
  - Rental Seasonality
  - Weather Seasonality
  - Bike-sharing (Total bike count and city) info for Seoul
  - Cities comparable to Seoul’s bike sharing system

EDA with data visualization

- The following charts were plotted
  - Scatterplot of Bike rental vs. Date
  - Scatterplot of Bike rental vs. Datetime
  - Bike rental histogram and kernel density
  - Scatterplot of Bike rental vs. Temperature faceted by Season
  - Boxplot of Bike Rental vs Hour faceted by Season
  - Daily total rainfall and snowfall (bar chart)
  - Bike rental per hour faceted by Season (boxplot)

Predictive analysis (Model Development)

- RENTED_BIKE_COUNT is the response (target) variable
- Predictors include
  - Weather features (temperature, wind speed, rainfall/snowfall)
  - Date/time features (hour, day of week, month, vacation)
- The dataset was split into testing and training sets
- A linear regression model was built using only weather features
- A linear regression model was built using all features
  - Features with large coefficients were identified.
- Predictions were made using each model.
- The models were evaluated using R-squared and RMSE.



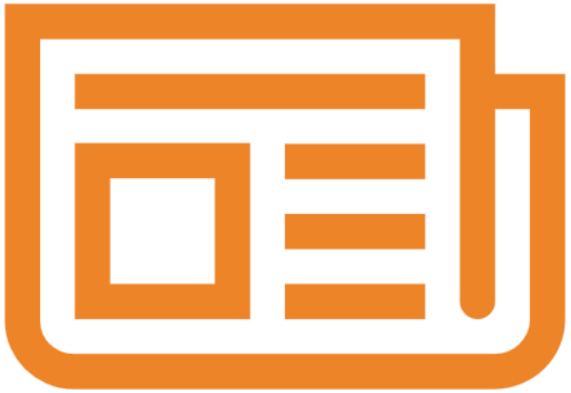
Predictive analysis (Model Refinement)

- Further models were built
  - Addition of higher order polynomial terms to the linear regression model
  - Interaction terms were added to the polynomial regression model
  - Regularisation terms were added using a glmnet linear regression model
  - 5 models were built in total
- Predictions were made with each model.
- The models were evaluated using R-squared and RMSE.

Build a R Shiny dashboard

- Leaflet-based interactive map
  - Shows the maximum predicted bike-sharing demand in the next 5 days based on a regression model using OpenWeather forecast data
  - An overview map displays basic bike prediction
  - A dropdown allows selection of a specific city (New York, USA, Paris, France, Suzhou, London)
- Detailed bike-sharing demand plots for a user selected city
  - ggplot renders on drilling down to a city
    - Interactive bike-sharing prediction trend
    - Static temperature trend line for the next 5 days
    - Humidity
    - Static humidity and bike-sharing demand prediction correlation plot

Results



- Exploratory data analysis results
  - See individual slides
- Predictive analysis results
  - The Polynomial Regression model with added terms was found to be the best performing.
- See dashboard slides for screenshots

EDA with SQL

Busiest bike rental times

- The maximum rental count and the associated date and hour indicate that the busiest period was 1800h on 19th June 2018.

```
# provide your solution here
query <- 'select date, hour, rented_bike_count from seoul_bike_sharing
    where rented_bike_count = (select max(rented_bike_count) from seoul_bike_sharing);'

dbGetQuery(conn, query)
```

A data.frame: 1 × 3

DATE	HOUR	RENTED_BIKE_COUNT
<chr>	<dbl>	<dbl>
19/06/2018	18	3556

Hourly popularity and temperature by seasons

```
# provide your solution here
#query <- 'select avg(temperature), avg(rented_bike_count) from seoul_bike_sharing
#   group by seasons;'

query <- 'select seasons, hour, avg(temperature) as avg_temperature, avg(rented_bike_count) as avg_rented_count from seoul_bike_sharing
    group by seasons, hour
    order by avg_rented_count desc
    limit 10;'

dbGetQuery(conn, query)
```

A data.frame: 10 × 4

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

The top 10 average rental counts alongwith the associated average temperature, grouped by season and hour indicate that evenings in summer and autumn are most popular.

Rental Seasonality

```
# provide your solution here
# SQLite has no standard deviation function

query <- 'select seasons,
    avg(rented_bike_count) as avg_hourly_count,
    min(rented_bike_count) as min_hourly_count,
    max(rented_bike_count) as max_hourly_count,
    sqrt(avg(rented_bike_count * rented_bike_count) - avg(rented_bike_count) * avg(rented_bike_count)) as std_deviation
    from seoul_bike_sharing
    group by seasons'

dbGetQuery(conn, query)
```

A data.frame: 4 × 5

SEASONS	avg_hourly_count	min_hourly_count	max_hourly_count	std_deviation
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
Autumn	924.1105	2	3298	617.3885
Spring	746.2542	2	3251	618.5247
Summer	1034.0734	9	3556	690.0884
Winter	225.5412	3	937	150.3374

- Summer is the most popular, followed closely by autumn and spring.
- Rentals drop sharply in winter, with a much smaller standard deviation.

Weather Seasonality

- A similar pattern is evident of summer being the most popular, with rentals reducing during autumn and spring and falling sharply during winter.
- There seems to be a correlation with temperature, visibility, sunlight and snowfall, which needs further exploration.

```
# provide your solution here
query <- 'select seasons,
            avg(rented_bike_count) as AVG_RENTED_COUNT,
            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_pt_temp,
            avg(solar_radiation) as avg_solar_radiation,
            avg(rainfall) as avg_rainfall,
            avg(snowfall) as avg_snowfall
            from seoul_bike_sharing
        group by seasons
        order by avg_rented_count desc'

dbGetQuery(conn, query)
```

A data.frame: 4 × 10

SEASONS	AVG_RENTED_COUNT	avg_temperature	avg_humidity	avg_wind_speed	avg_visibility	avg_dew_pt_temp	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

Bike-sharing info in Seoul

```
# provide your solution here
query <- 'select w.city, w.country, w.lat, w.lng, w.population, b.bicycles from world_cities w, bike_sharing_systems b
            where lower(w.city) = lower(b.city) and lower(w.city = "Seoul")'

dbGetQuery(conn, query)
```

A data.frame: 1 × 6

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

A total of 20,000 bikes serve a population of almost 21.8 million.

Cities similar to Seoul

```
# provide your solution here
query <- 'select w.city, w.country, w.lat, w.lng, w.population, b.bicycles from world_cities w, bike_sharing_systems b
            where lower(w.city_ascii) = lower(b.city) and (b.bicycles between 15000 and 20000)'

dbGetQuery(conn, query)

# Later, try to figure out how to return all cities with comparable systems, regardless of whether coordinate data exists
```

A data.frame: 7 × 6

CITY	COUNTRY	LAT	LNG	POPULATION	BICYCLES
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
Beijing	China	39.9050	116.3914	19433000	16000
Ningbo	China	29.8750	121.5492	7639000	15000
Shanghai	China	31.1667	121.4667	22120000	19165
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	Korea, South	37.5833	127.0000	21794000	20000

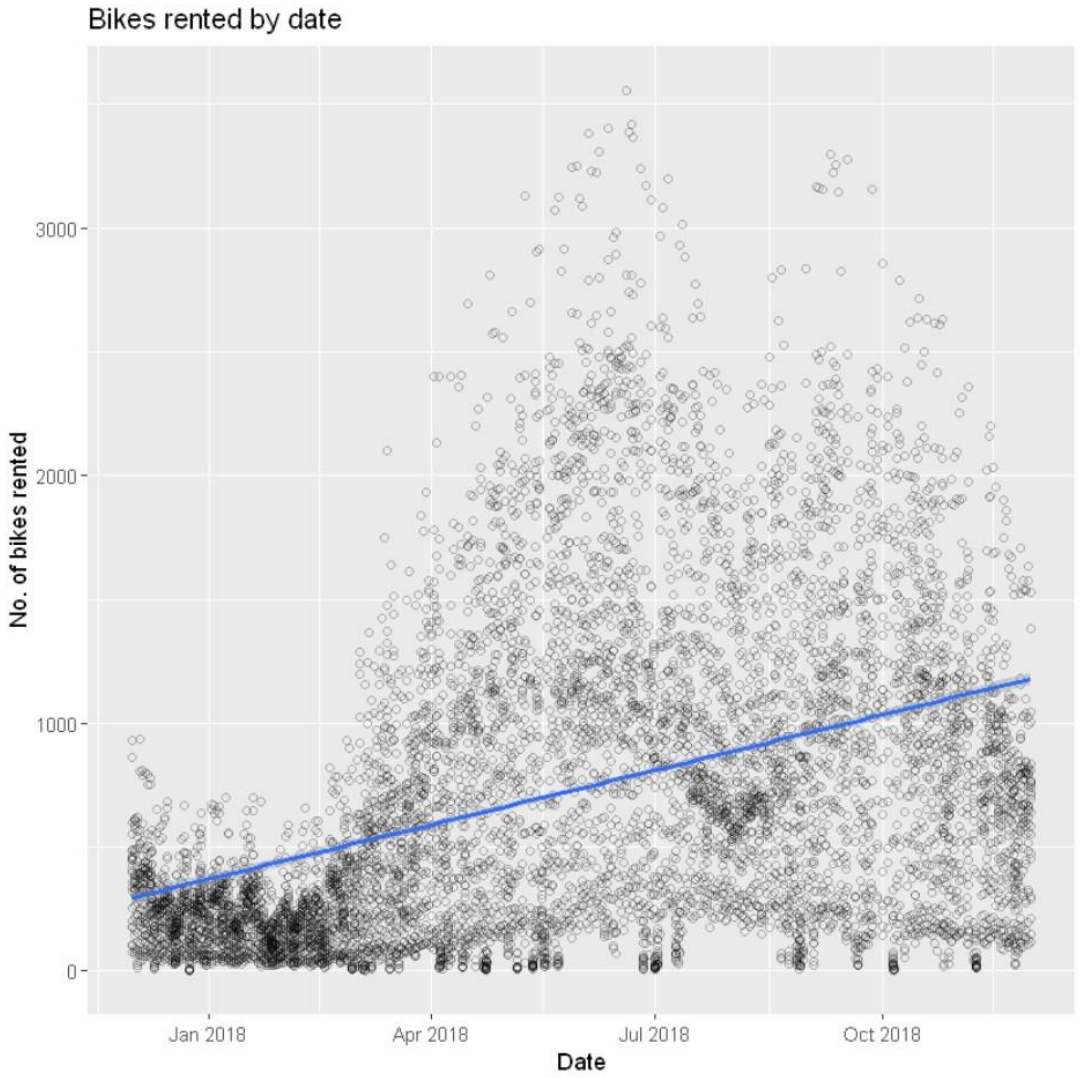
Interestingly, all other cities with comparable bike share systems are based in China.

EDA with Visualization

Bike rental vs. Date

Without the benefit of prior or subsequent data, a scatterplot of bikes rented by date reveals:

1. Demand is lowest earlier in the year and increases gradually, as illustrated by a linear regression line.
2. There are two spikes, one around June and another around September, which could be accounted for by holidays and tourism.
3. Demand reduces closer to December and seems to be minimum during January and February, possibly due to cold weather and a preference for other modes of transport.



Bike rental vs. Datetime

The same scatterplot, colour coded by hour.

The increase in demand during the spikes seems to have a big contribution from the evenings, especially around 1800h to 1900h, with another slightly smaller contribution around 0600h to 0800h.

These might be due to commuting hours.

Bike rental histogram

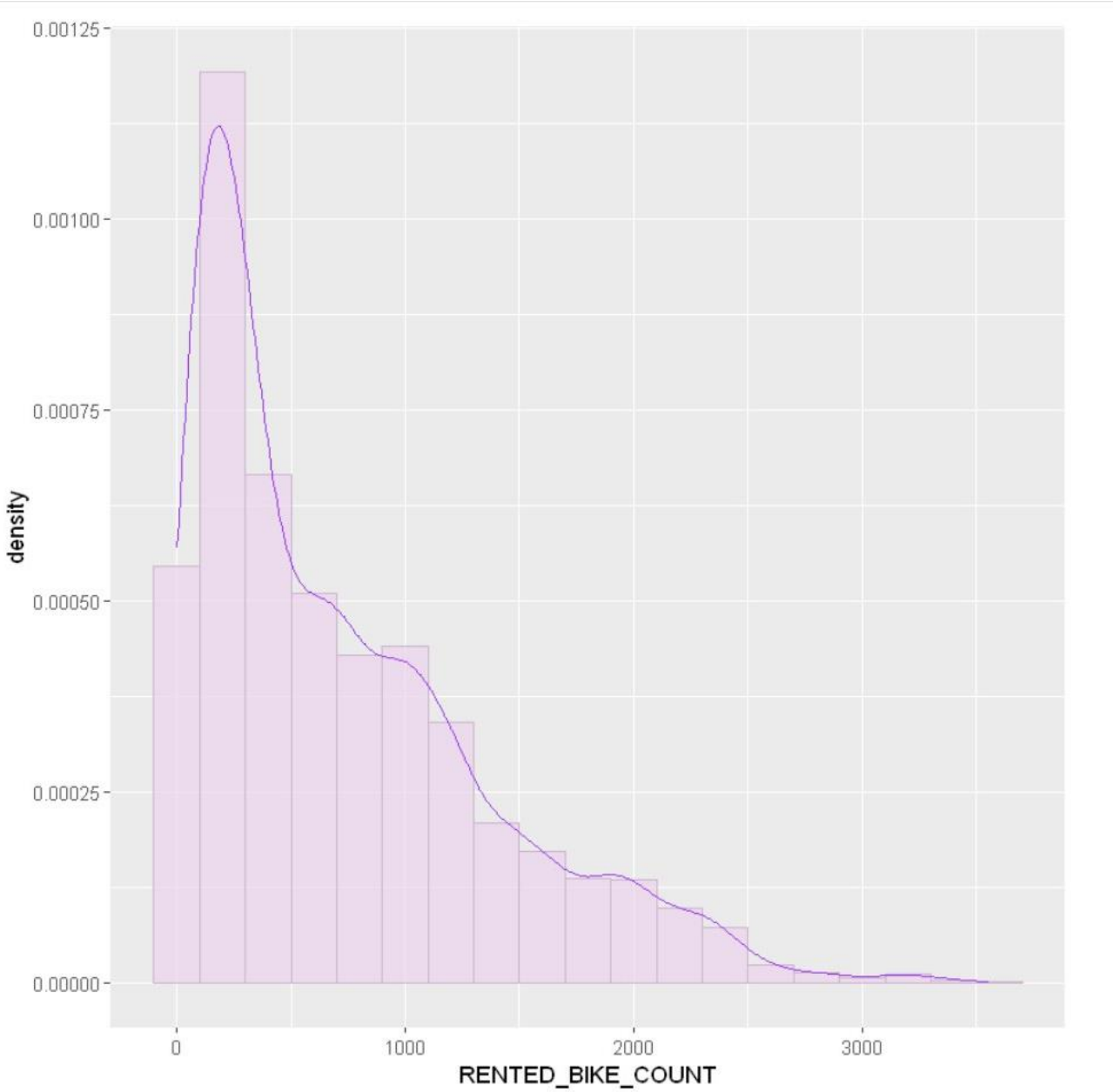
Histogram of bike rentals with a kernel density curve

Only a few bikes are rented most of the time.

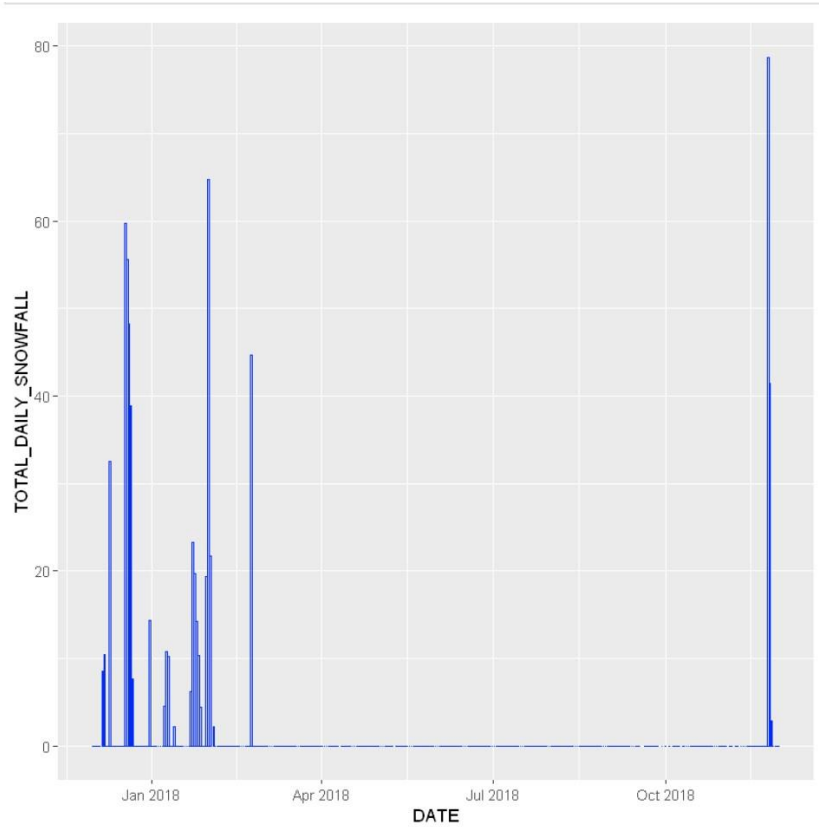
The mode is between 200 and 300.

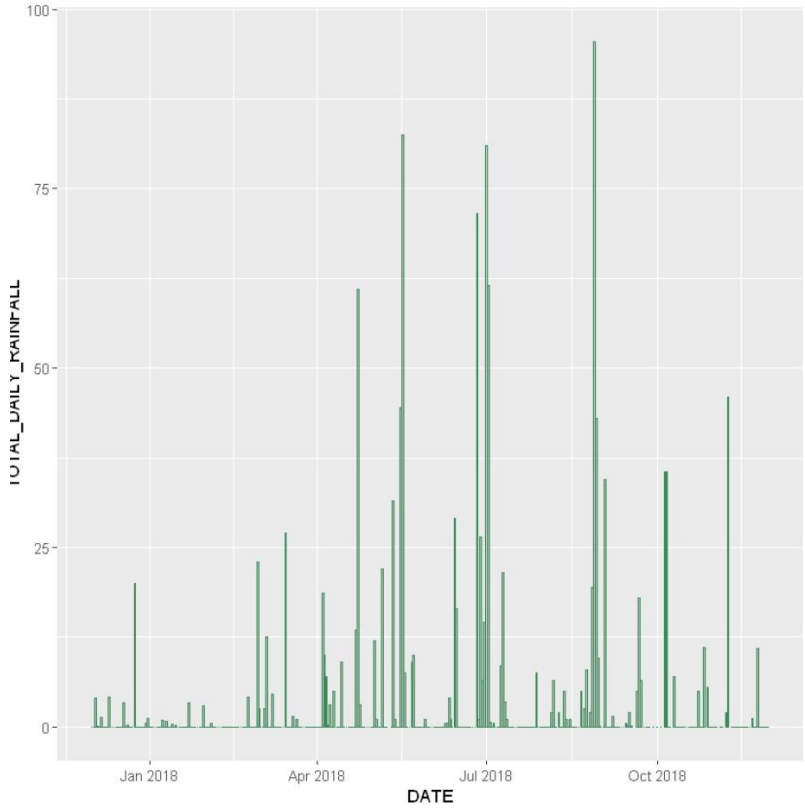
On occasion, a high number of bikes gets rented.

This might have implications on supply meeting demand.



Daily total rainfall and snowfall





- A total daily snowfall bar chart reveals concentration in the winter months.
- A similar chart for rainfall reveals spikes around May, July and September.

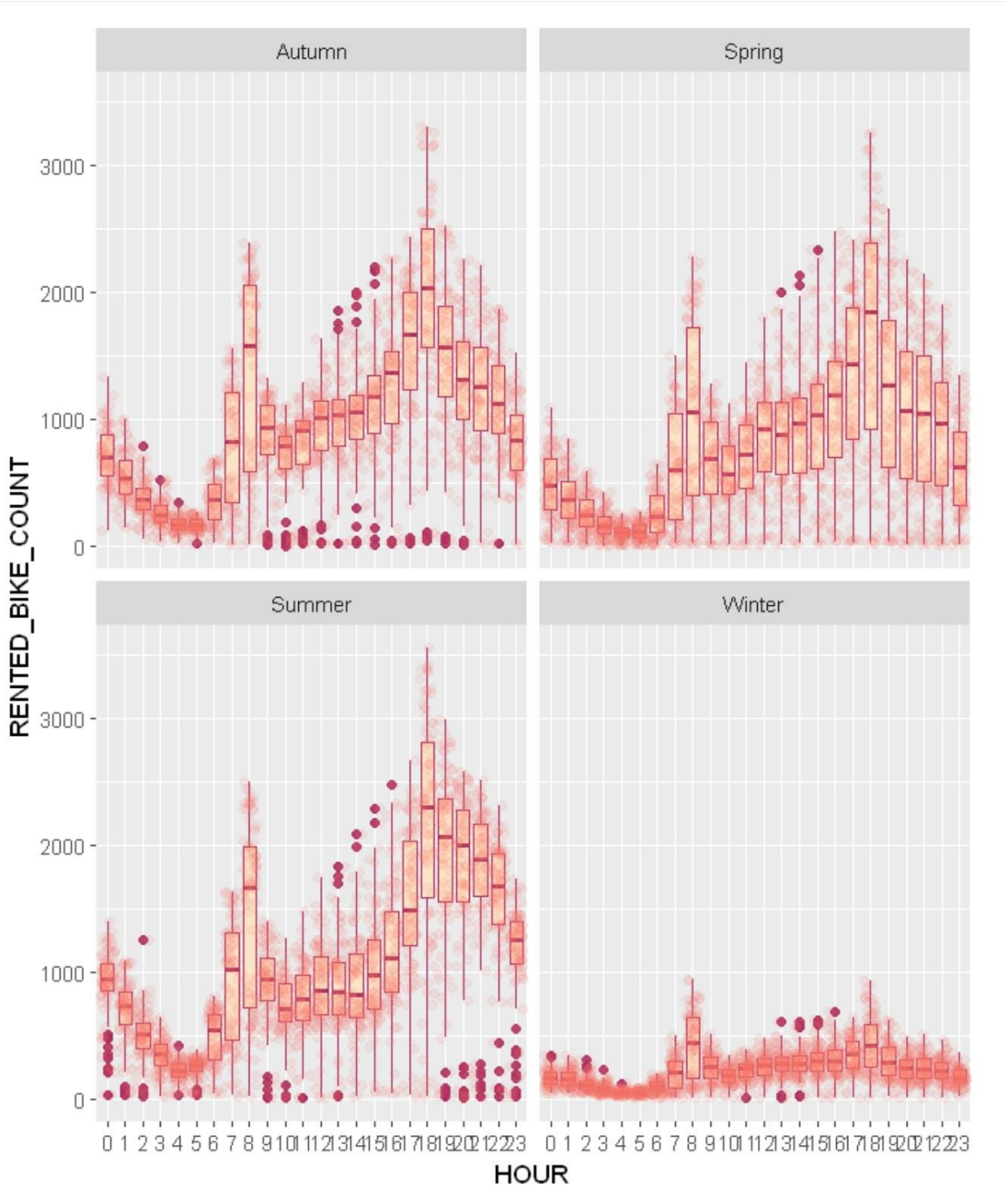
Bike Rental per hour by Season

Boxplot of rentals per hour by season

1.Usage peaks at 0800h and 1800h, possibly associated with commuting to and from work.

2.Usage reduces significantly in winter.

3.Peak hours are the same across seasons.



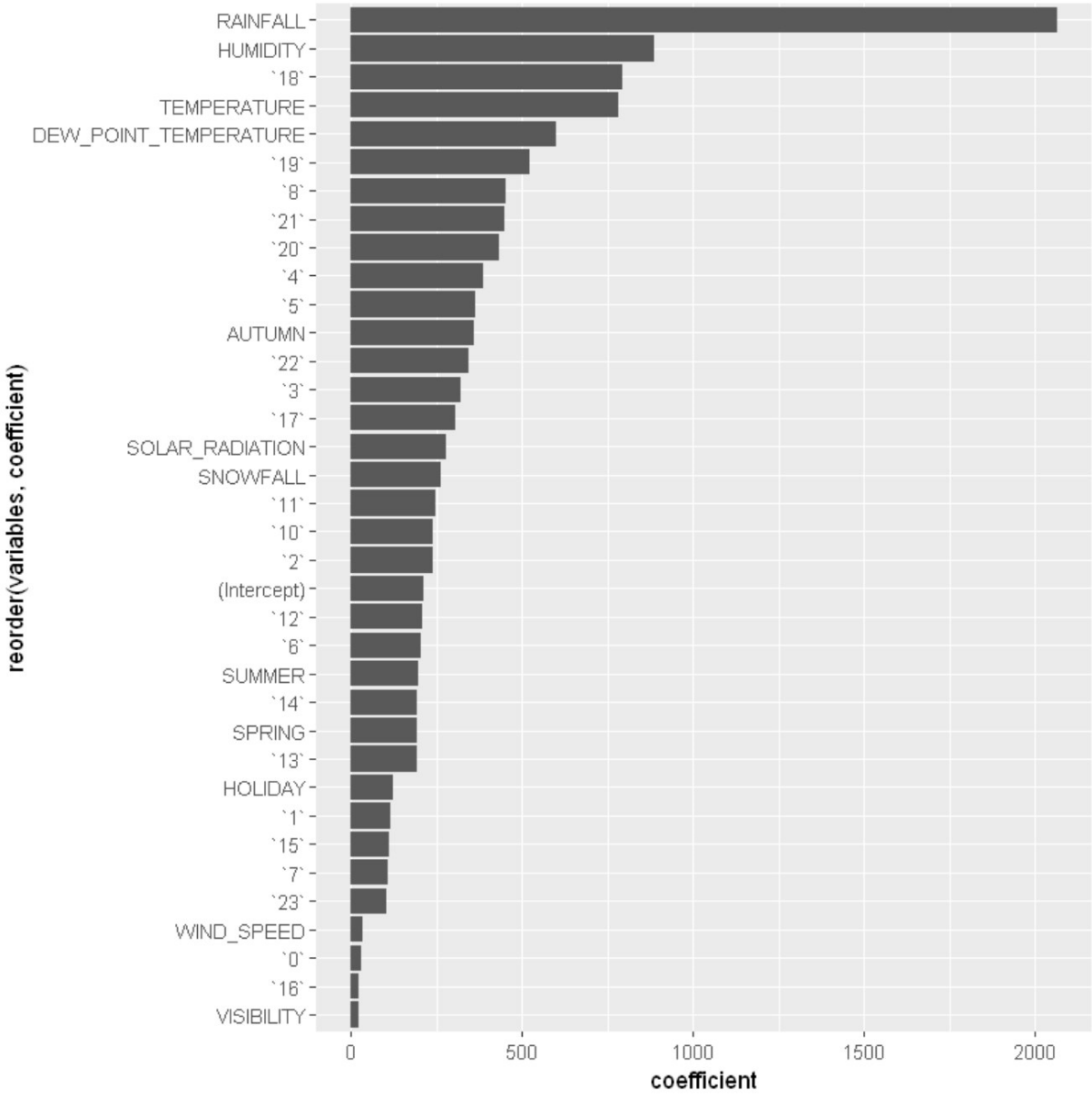
Predictive analysis

Ranked coefficients

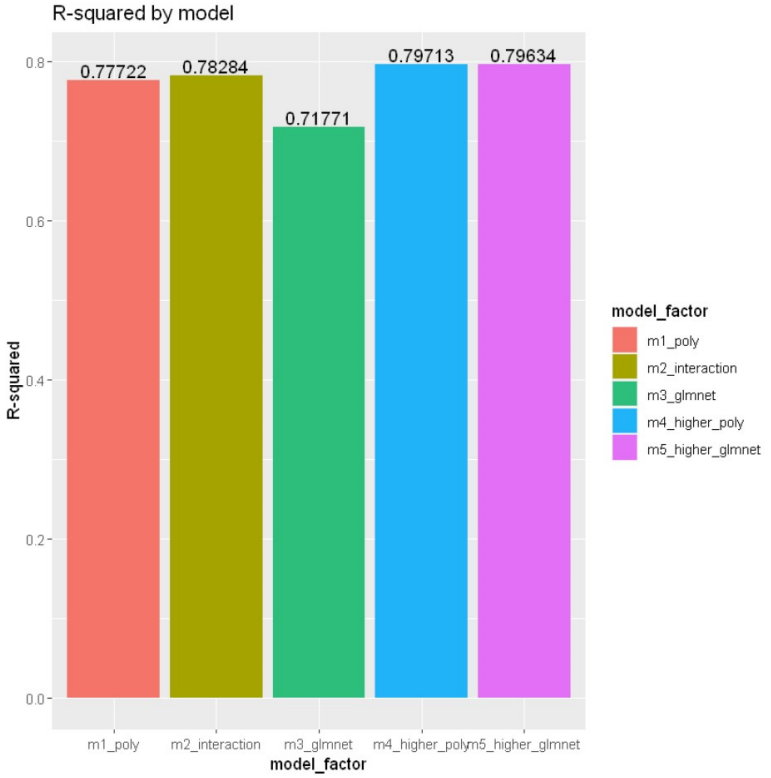
The predictors with the most effect on bike rental demand appear to be rainfall, humidity and temperature.

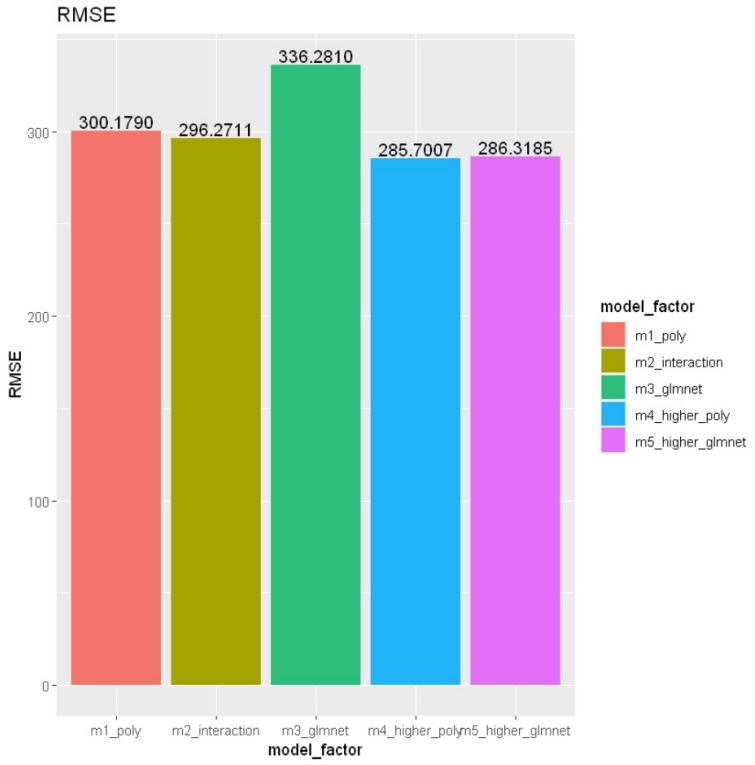
Visibility and wind speed seem to have the least effect.

The hour of the day seems linked, but should be considered as a grouped variable instead of individually.



Model evaluation





Grouped bar charts of: polynomial regression, polynomial with interaction terms, glmnet regression, polynomial with added terms, glmnet with added terms.

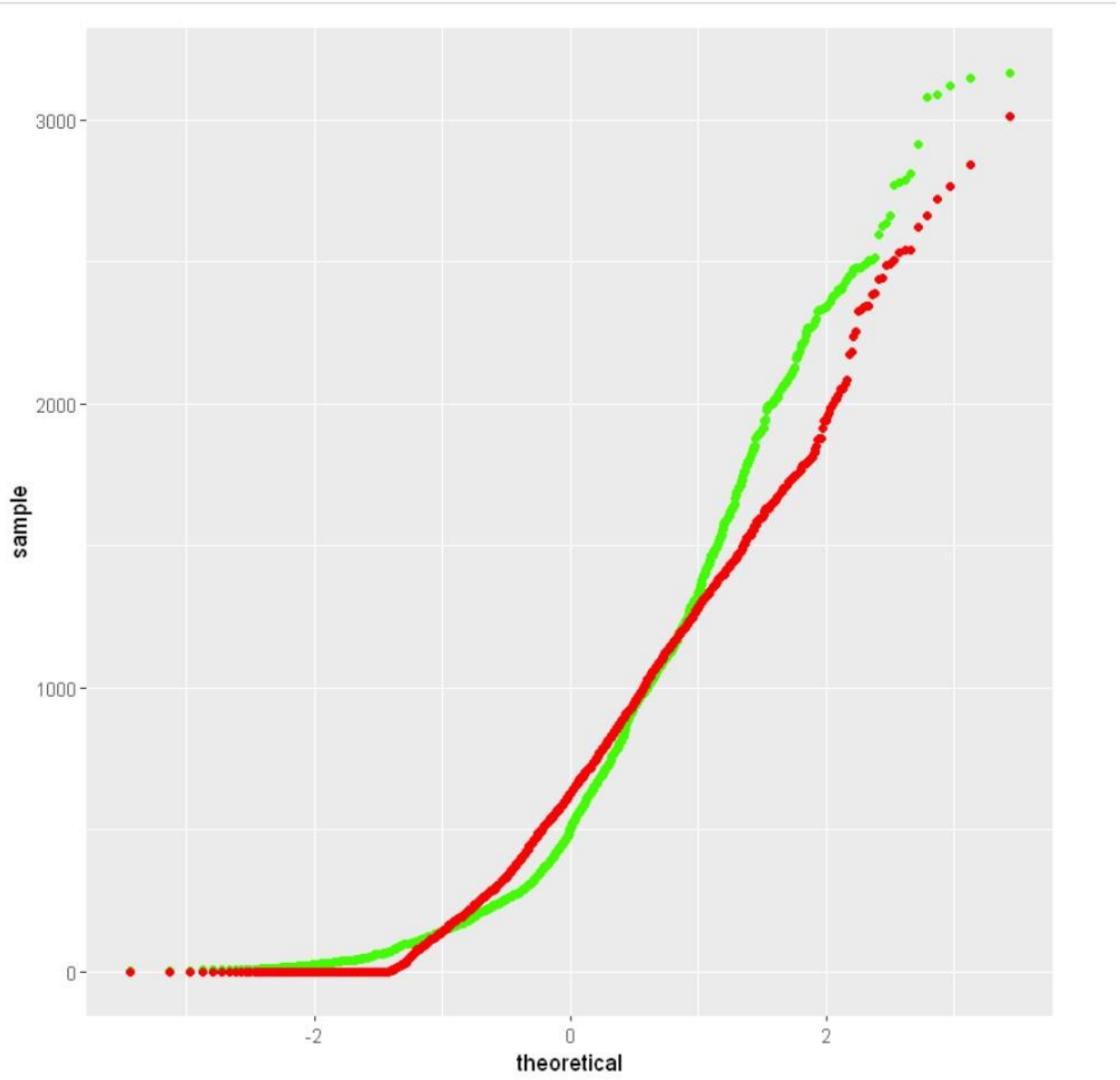
The Polynomial Regression model with added terms has the highest R-squared (0.79713) and the lowest RMSE (285.7007)

Find the best performing model

Model Formula

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

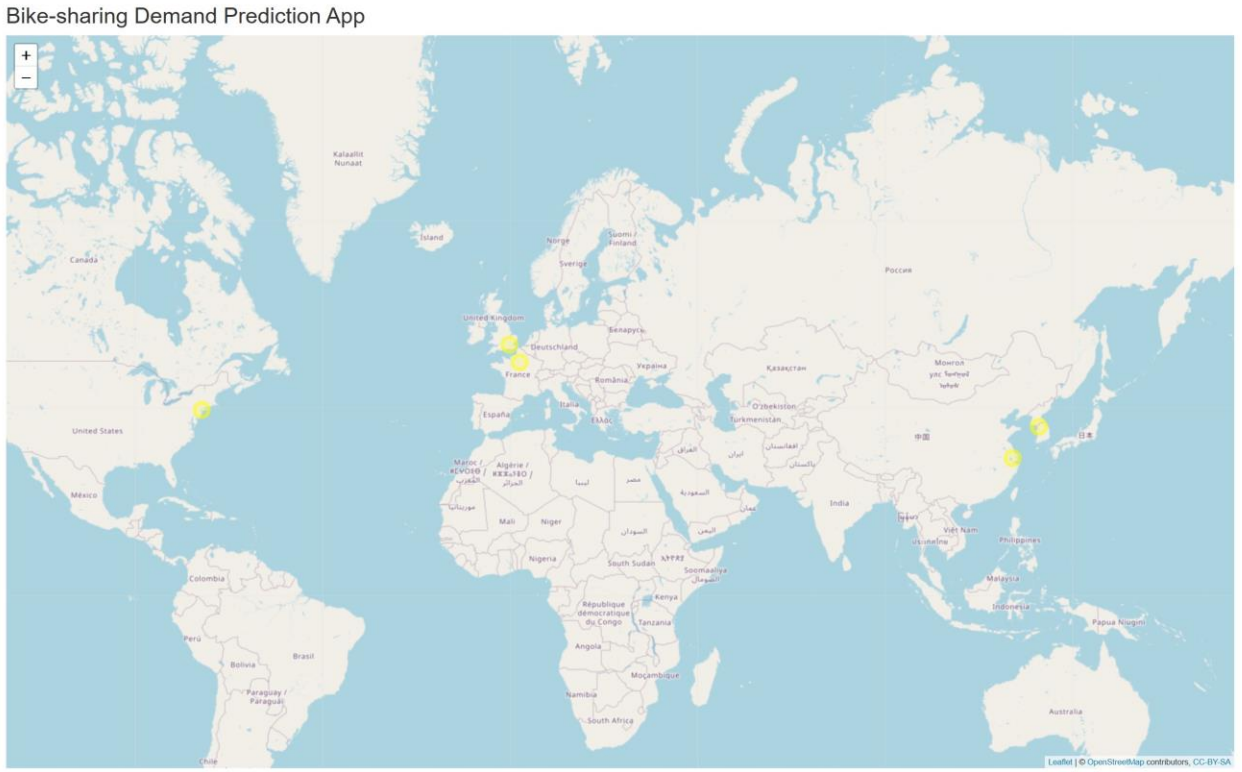
Q-Q plot of the best model

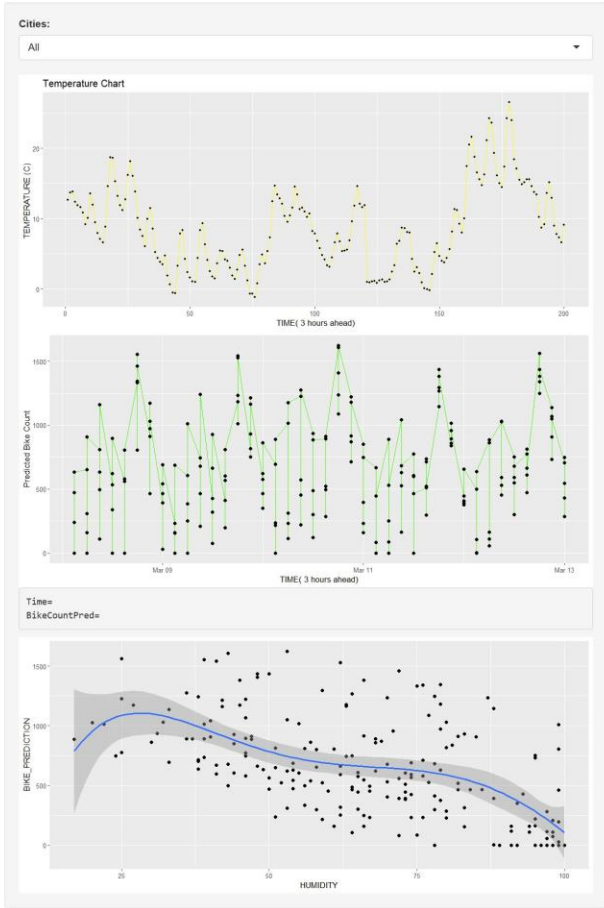


Dashboard

Worldwide Demand Prediction

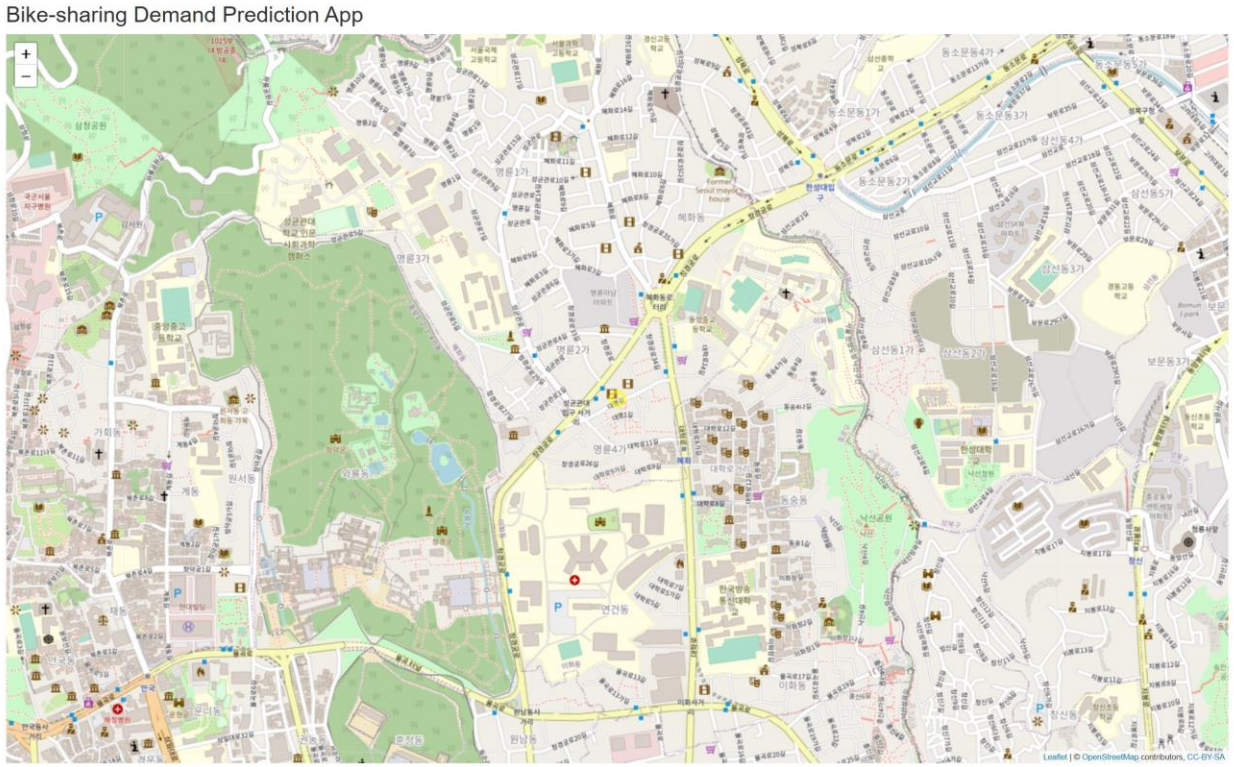
- Leaflet-based interactive worldwide demand prediction map
- Dropdown to select a specific city (New York, USA, Paris, France, Suzhou, London)
- Temperature chart
- Predicted bike demand chart
- Correlation between bike demand and humidity

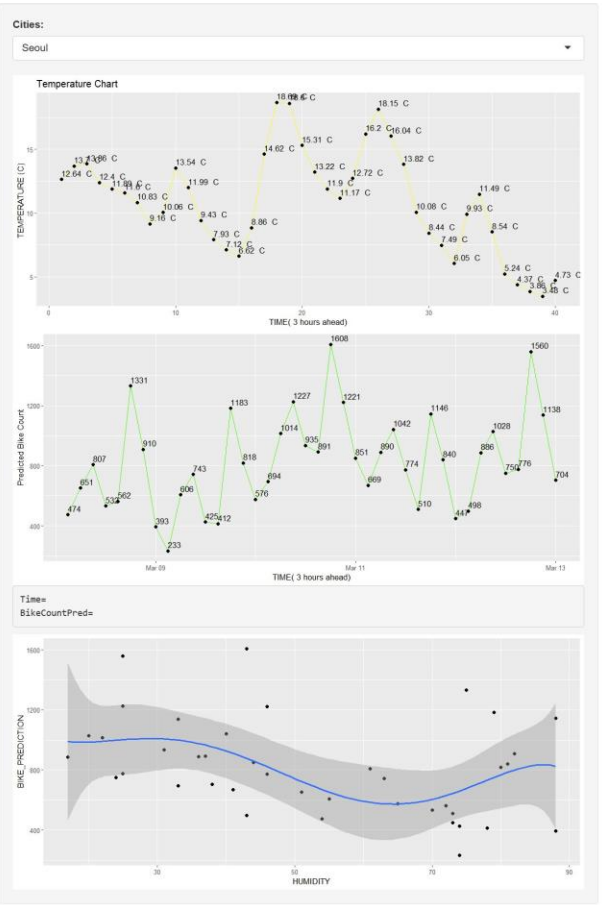




Bike Demand Map (Seoul)

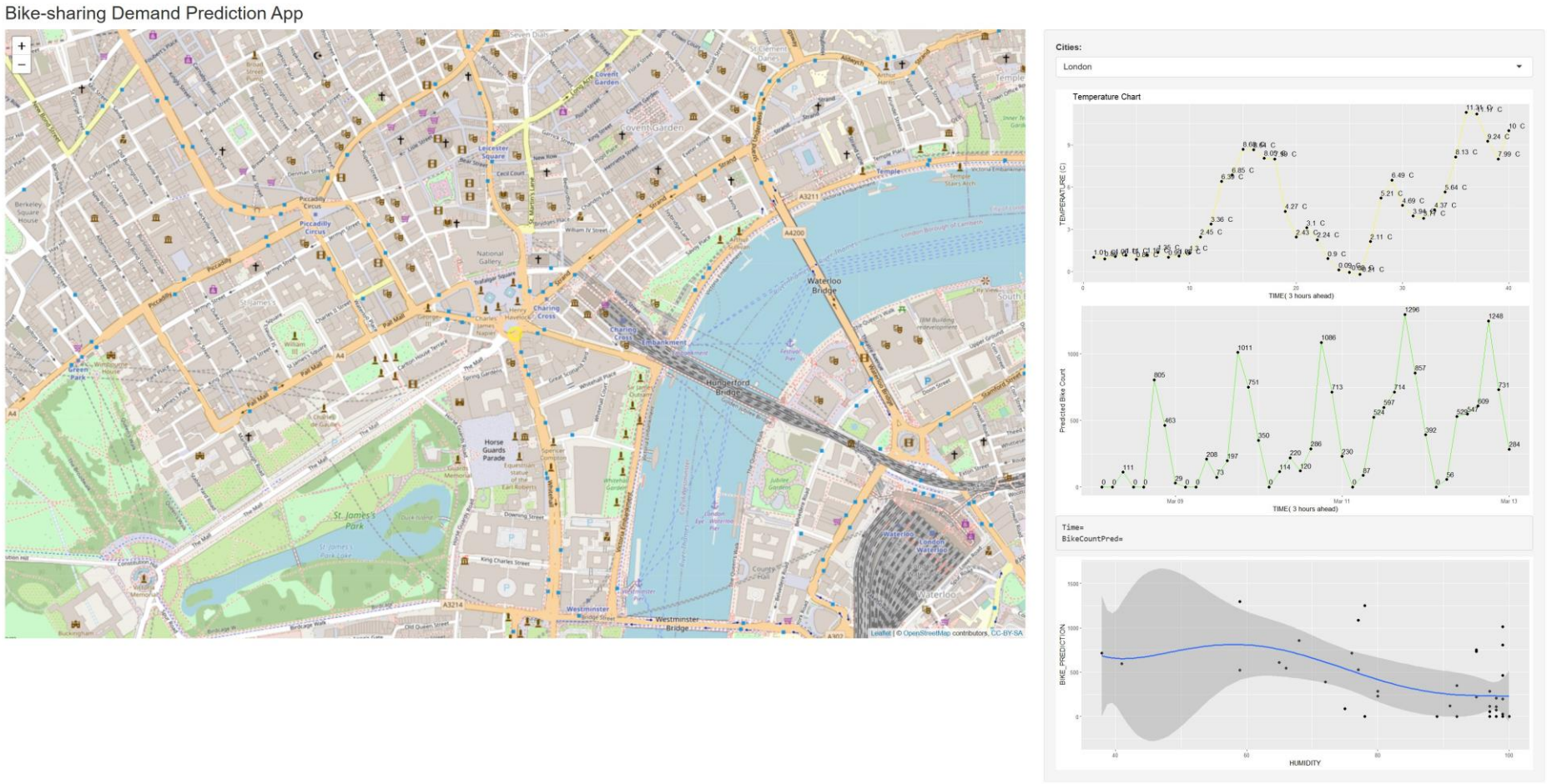
- Leaflet-based interactive city demand prediction map
- Dropdown with Seoul selected
- Temperature chart
- Predicted bike demand chart
- Correlation between bike demand and humidity





Bike Demand Map (London)

- Leaflet-based interactive city demand prediction map
- Dropdown with London selected
- Temperature chart
- Predicted bike demand chart
- Correlation between bike demand and humidity



CONCLUSION

- Bike sharing demand is highest in the summer.
- Bimodal diurnal peak.
- Affected by temperature, humidity, precipitation.
- Regression models have predictive value for demand and logistics.

APPENDIX (Additional SQL Queries)

Task 1 - Record Count

Determine how many records are in the seoul_bike_sharing dataset.

Solution 1

```
# provide your solution here
query <- 'select count(*) from seoul_bike_sharing'
dbGetQuery(conn, query)
```

A data.frame: 1 × 1

count(*)
<int>
8465

Task 2 - Operational Hours

Determine how many hours had non-zero rented bike count.

Solution 2

```
# provide your solution here
query <- 'select count(*) from seoul_bike_sharing
    where rented_bike_count <> 0;'
dbGetQuery(conn, query)
```

A data.frame: 1 × 1

count(*)
<int>
8465

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.

Solution 3

```
# provide your solution here
query <- 'select * from cities_weather_forecast
    where lower(city) = "seoul"
    limit 1;'
dbGetQuery(conn, query)
```

A data.frame: 1 × 12

CITY	WEATHER	VISIBILITY	TEMP	TEMP_MIN	TEMP_MAX	PRESSURE	HUMIDITY	WIND_SPEED	WIND_DEG	SEASON	FORECAST_DATETIME
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>
Seoul	Clear	10000	12.32	10.91	12.32	1015	50	2.18	248	Spring	1618574400

Task 4 - Seasons

Find which seasons are included in the seoul bike sharing dataset.

Solution 4

```
# provide your solution here
query <- 'select distinct seasons from seoul_bike_sharing;'
dbGetQuery(conn, query)
```

A data.frame: 4 × 1

SEASONS
<chr>
Winter
Spring
Summer
Autumn

Task 5 - Date Range

Find the first and last dates in the Seoul Bike Sharing dataset.

Solution 5

```
# provide your solution here

# Reverse and format date string by extracting substrings and rearranging them
#query <- 'select date(substr(date,7)||"-"||substr(date,4,2)||"-"||substr(date,1,2)) as date from seoul_bike_sharing limit 5;'

# Use reversed date string to select first and last dates
query <- 'select min(date(substr(date,7)||"-"||substr(date,4,2)||"-"||substr(date,1,2))) as first_date,
    max(date(substr(date,7)||"-"||substr(date,4,2)||"-"||substr(date,1,2))) as last_date from seoul_bike_sharing;'

dbGetQuery(conn, query)
```

A data.frame: 1 × 2

first_date	last_date
<chr>	<chr>
2017-12-01	2018-11-30

APPENDIX (VISUALISATION CODE)

Task 10 - Create a scatter plot of `RENTED_BIKE_COUNT` vs `DATE`. ¶

Tune the opacity using the `alpha` parameter such that the points don't obscure each other too much.

Solution 10

```
# provide your solution here
ggplot(data = seoul_bike_sharing, mapping = aes(x = DATE, y = RENTED_BIKE_COUNT)) +
    geom_point(shape = 1, alpha = 0.3) +
    geom_smooth(method = "lm", na.rm = TRUE) +
    labs(title = "Bikes rented by date",
        x = "Date", y = "No. of bikes rented")
```

Task 11 - Create the same plot of the `RENTED_BIKE_COUNT` time series, but now add `HOURS` as the colour.

Solution 11

```
# provide your solution here
# provide your solution here
ggplot(data = seoul_bike_sharing, aes(x = DATE, y = RENTED_BIKE_COUNT, color = HOUR), alpha = 0.3) +
    geom_point() +
    labs(title = "Bikes rented by date",
        x = "Date", y = "No. of bikes rented")
```

Task 12 - Create a histogram overlaid with a kernel density curve

Normalize the histogram so the y axis represents 'density'. This can be done by setting `y=..density..` in the aesthetics of the histogram.

▶ Click here for a hint

▶ Click here for another hint

Solution 12

```
# provide your solution here
ggplot(seoul_bike_sharing, mapping = aes(x = RENTED_BIKE_COUNT, y = after_stat(density))) + # y = ..density.. has been deprecated
    geom_histogram(binwidth = 200, fill = "thistle2", color = "thistle3", alpha = 0.7) +
    geom_density(color = "purple")
```

Outliers (boxplot) ¶

Task 14 - Create a display of four boxplots of `RENTED_BIKE_COUNT` vs. `HOUR` grouped by `SEASONS`.

Use `facet_wrap` to generate four plots corresponding to the seasons.

Solution 14

```
# provide your solution here
ggplot(seoul_bike_sharing, mapping = aes(x = HOUR, y= RENTED_BIKE_COUNT)) +
    geom_boxplot(fill = "bisque", color = "maroon", alpha = 0.9) +
    geom_jitter(aes(color = "slategray"), alpha=0.1) +
    labs(colour = "") +
    facet_wrap(.~SEASONS)
```

Solution 15

```
# provide your solution here
seoul_daily_precipitation <- seoul_bike_sharing %>%
    group_by(DATE) %>%
    summarize(TOTAL_DAILY_RAINFALL = sum(RAINFALL), TOTAL_DAILY_SNOWFALL = sum(SNOWFALL))

head(seoul_daily_precipitation)
```

A tibble: 6 × 3

DATE	TOTAL_DAILY_RAINFALL	TOTAL_DAILY_SNOWFALL
<date>	<dbl>	<dbl>
2017-12-01	0.0	0.0
2017-12-02	0.0	0.0
2017-12-03	4.0	0.0
2017-12-04	0.1	0.0
2017-12-05	0.0	0.0
2017-12-06	1.3	8.6

```
ggplot(seoul_daily_precipitation, aes(x = DATE, y = TOTAL_DAILY_RAINFALL)) +
    geom_bar(stat = "identity", color = "seagreen4", alpha = 0.3)
```

APPENDIX (VISUALISATION CODE)

Task 13 - Use a scatter plot to visualize the correlation between `RENTED_BIKE_COUNT` and `TEMPERATURE` by `SEASONS`.

Start with `RENTED_BIKE_COUNT` vs. `TEMPERATURE`, then generate four plots corresponding to the `SEASONS` by adding a `facet_wrap()` layer. Also, make use of colour and opacity to emphasize any patterns that emerge.
Use `HOUR` as the color.

Solution 13

```
# provide your solution here
ggplot(seoul_bike_sharing, aes(x=TEMPERATURE, y=RENTED_BIKE_COUNT, colour=HOUR)) +
    geom_point(alpha = 0.4) +
    facet_wrap(.~SEASONS)
```