



DATA SCIENCE WITH R - CAPSTONE PROJECT

Joyce Mwangi

22/06/2024

OUTLINE



- Executive Summary
- Introduction
- Methodology
- Results
 - Visualization – Charts and models
 - Dashboard - Shiny
- Discussion
 - Findings
 - Conclusion
 - Recommendations
- Overall Conclusion
- Appendix

EXECUTIVE SUMMARY



- Analysis Overview
 - Data collection
 - Data preprocessing
- Data Exploration
 - Explore patterns and trends
 - Visualize distributions
 - Find weather-bike correlations
- Model Development
 - Build a predictive regression model (multiple, polynomial, and polynomial with interactions)
 - Regularization with Lasso and Ridge
- Dashboarding with Shiny
 - Create a shiny app that predicts weather-bike correlations
- Conclusions and Recommendations
 - Explore how weather influences bike demand
- Appendix

INTRODUCTION



- Bike-sharing demand data analysis explores factors (variables) that impact the bike demand and usage of bike-sharing services in different locations and time.
- Therefore, the purpose of this analysis was to understand the historical data by studying its trends and patterns to make predictions about future demands.
- This analysis used the statistical analysis methods like data cleaning and wrangling, explanatory data analysis with SQL and R, and visualization and dashboarding with Cognos Analytics, R, and R Shiny.
- The study also used various models, such as multiple linear regression, polynomial, and regularization techniques to enhance model tuning and predictive capabilities.
- It closes with a short discussion, conclusions, and recommendations.

OBJECTIVES OF THE STUDY

- Identify and analyze the key weather and date factors that influence bike rental demand.
- Create simple visualizations of this analysis to enhance insight uptake
- Use historical data and the predictor variables to create predictive models that can forecast bike rental demands using regression analysis (multiple and polynomial)
- Give conclusions and recommendations based on the shiny app predictions

PROBLEM OF THE STUDY, PURPOSE, AND RESEARCH QUESTIONS

- **Problem statement:** Predict the number of bikes that will be rented from a bike-sharing system at a given time based on weather and date (like time of day, day of the week, holidays, and seasons).
- **Purpose:** To build a bike-renting predictive model that forecast bike rental demand to enhance a system's overall bike-sharing efficiency.

Research Questions

- How does weather (e.g., wind, temperature, precipitation) affect bike rental demand?
- How does date affect bike rental demand
- Which is the strongest predictor of bike rental demand
- Can the model predict for more than one city?
- Which model predicts best?

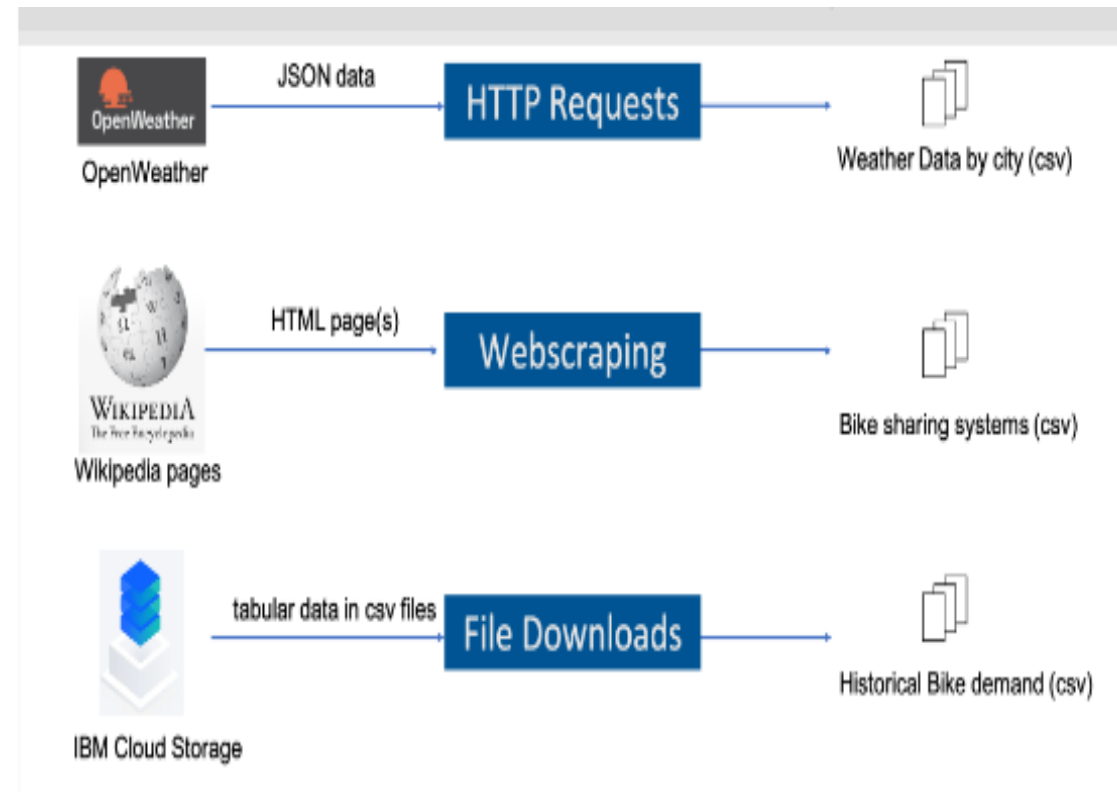
METHODOLOGY



- Perform data collection
- Perform data wrangling
- Perform exploratory data analysis (EDA) using SQL and visualization
- Perform predictive analysis using R regression models
- Build a shiny dashboard app to predict bike demand based on weather
- Discussion
- Conclusions and recommendations

DATA COLLECTION

- Extract bike sharing system data from a Wiki page and convert the data to a data frame
- OpenWeather APIs Calls
 - Get the current weather data for a city
 - Get 5-day weather forecast for a list of cities
- Download datasets in csv files from cloud storage



DATA WRANGLING

- Standardized column names
- Removed undesired reference links
- Extracted numeric values
- Detected and handled missing values
- Created dummy variables for categorical variables
- Normalized data

```
# Print its head  
head(bike_sharing_df)
```

A tibble: 6 × 10

COUNTRY	CITY	NAME	SYSTEM	OPERATOR	LAUNCHED	DISCONTINUED	STATIONS	BICYCLES	DAILY_RIDERSHIP
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
Albania	Tirana	Ecovolis	NA	NA	March 2011	NA	8	200	NA
Argentina	Mendoza	Metrobici	NA	NA	2014	NA	2	40	NA
Argentina	San Lorenzo, Santa Fe	Biciudad	Biciudad	NA	27 November 2016	NA	8	80	NA

RESULTS:

SEOUL BIKE SHARING DATASET ANALYSIS WITH SQL

- Total bike count and city info for Seoul
- Seoul hourly popularity and temperature by season

```
: dbGetQuery(con, "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

```
# provide your solution here  
dbGetQuery(con, "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

SEOUL BIKE SHARING DATASET ANALYSIS WITH SQL

- Seoul weather seasonality vs average bike

```
[: dbGetQuery(con, " SELECT SEASONS,
AVG(RENTED_BIKE_COUNT) as AVG_BIKE_COUNT, AVG(TEMPERATURE) as AVG_TEMP,
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,
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_BIKE_COUNT desc")
```

A data.frame: 4 × 10

SEASONS	AVG_BIKE_COUNT	AVG_TEMP	AVG_HUMIDITY	AVG_WIND_SPEED	AVG_VISIBILITY	AVG_DEW_POINT
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Summer	1034.0734	26.587711	64.98143	1.609420	1501.745	
Autumn	924.1105	13.821580	59.04491	1.492101	1558.174	
Spring	746.2542	13.021685	58.75833	1.857778	1240.912	
Winter	225.5412	-2.540463	49.74491	1.922685	1445.987	

- Cities with similar bike counts to Seoul
(15 000 – 20 000)

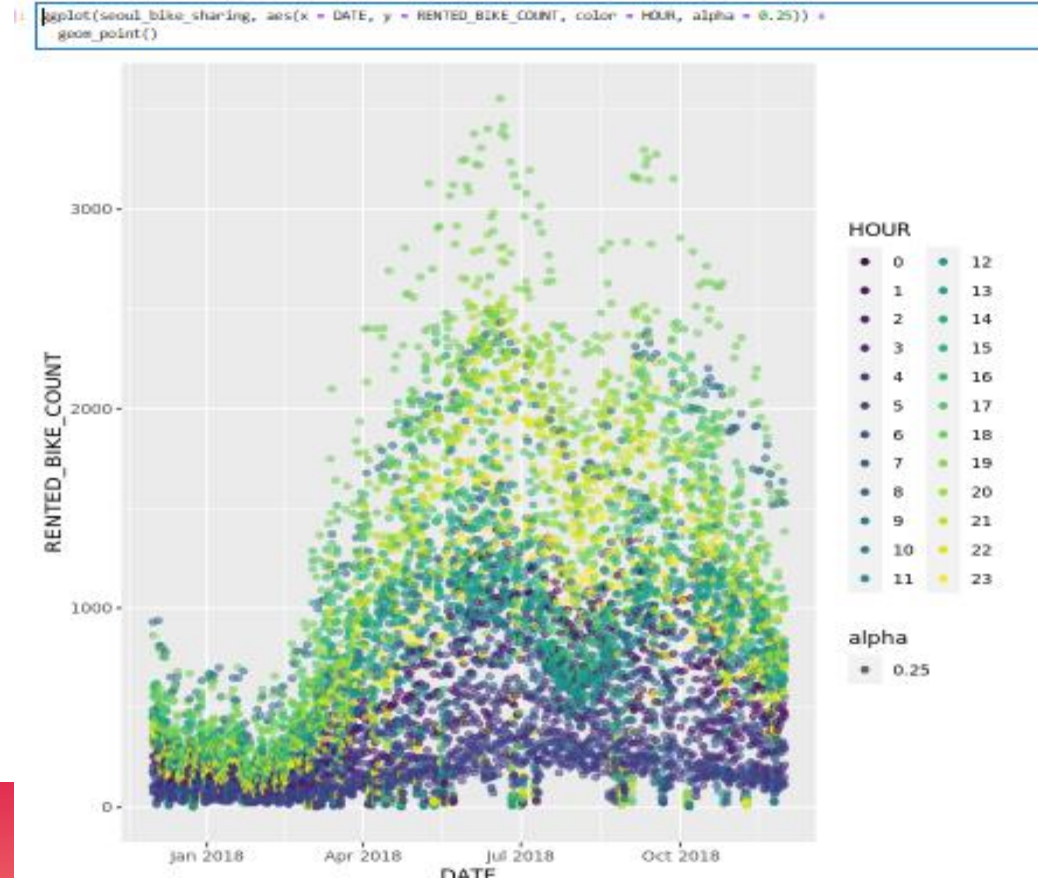
```
dbGetQuery(con, "SELECT B.BICYCLES, B.CITY, B.COUNTRY, W.LAT, W.LNG, W.POPULATION
FROM BIKE_SHARING_SYSTEMS AS B
LEFT JOIN WORLD_CITIES AS W ON B.CITY = W.CITY_ASCII
WHERE B.CITY = 'Seoul' OR B.BICYCLES BETWEEN 15000 AND 20000
order by B.BICYCLES desc")
```

A data.frame: 9 × 6

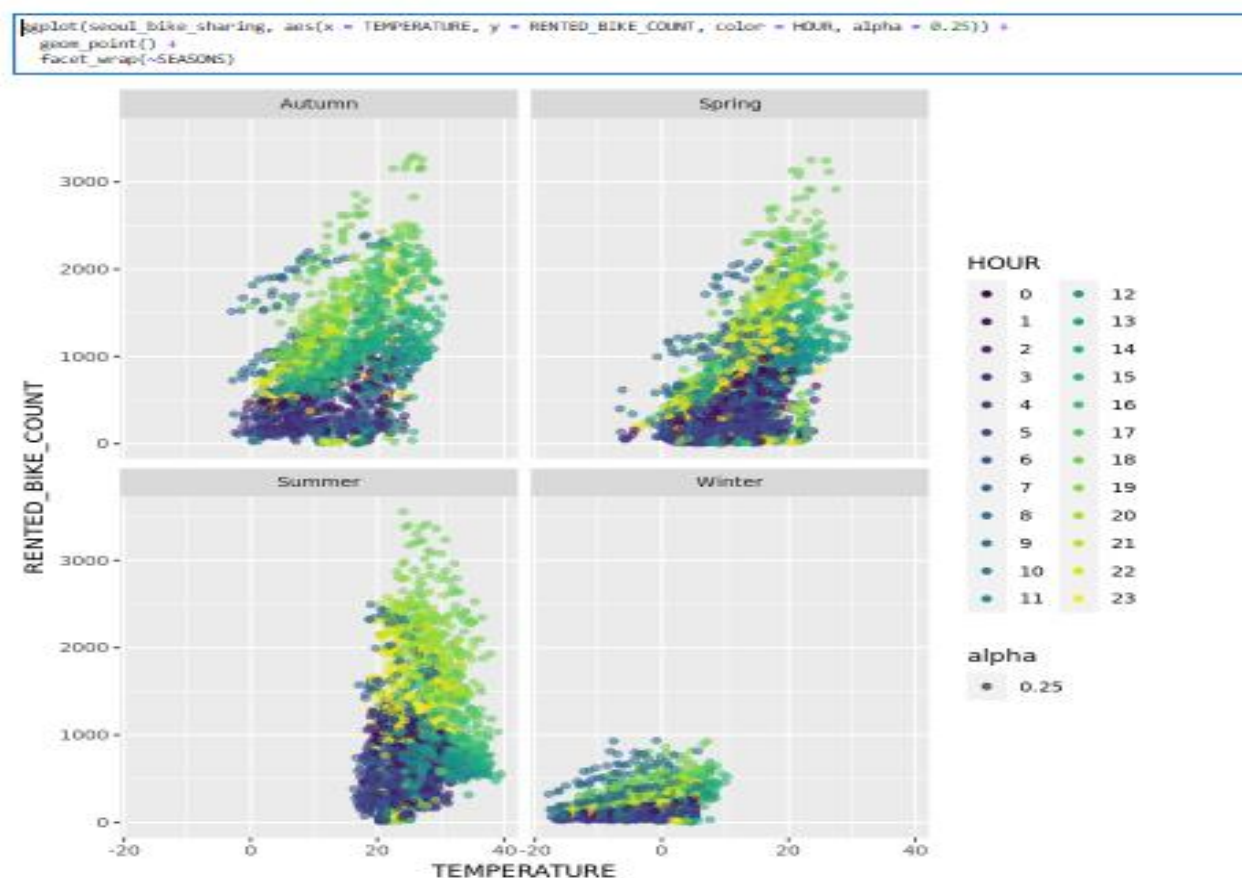
BICYCLES	CITY	COUNTRY	LAT	LNG	POPULATION
<int>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
20000	Kunshan	China	NA	NA	NA
20000	Weifang	China	36.7167	119.1000	9373000
20000	Xi'an	China	34.2667	108.9000	7135000
20000	Zhuzhou	China	27.8407	113.1469	3855609
20000	Seoul	South Korea	37.5833	127.0000	21794000

VISUALIZING SEOUL BIKE SHARING DATASET

- Rented bike count vs date vs hour

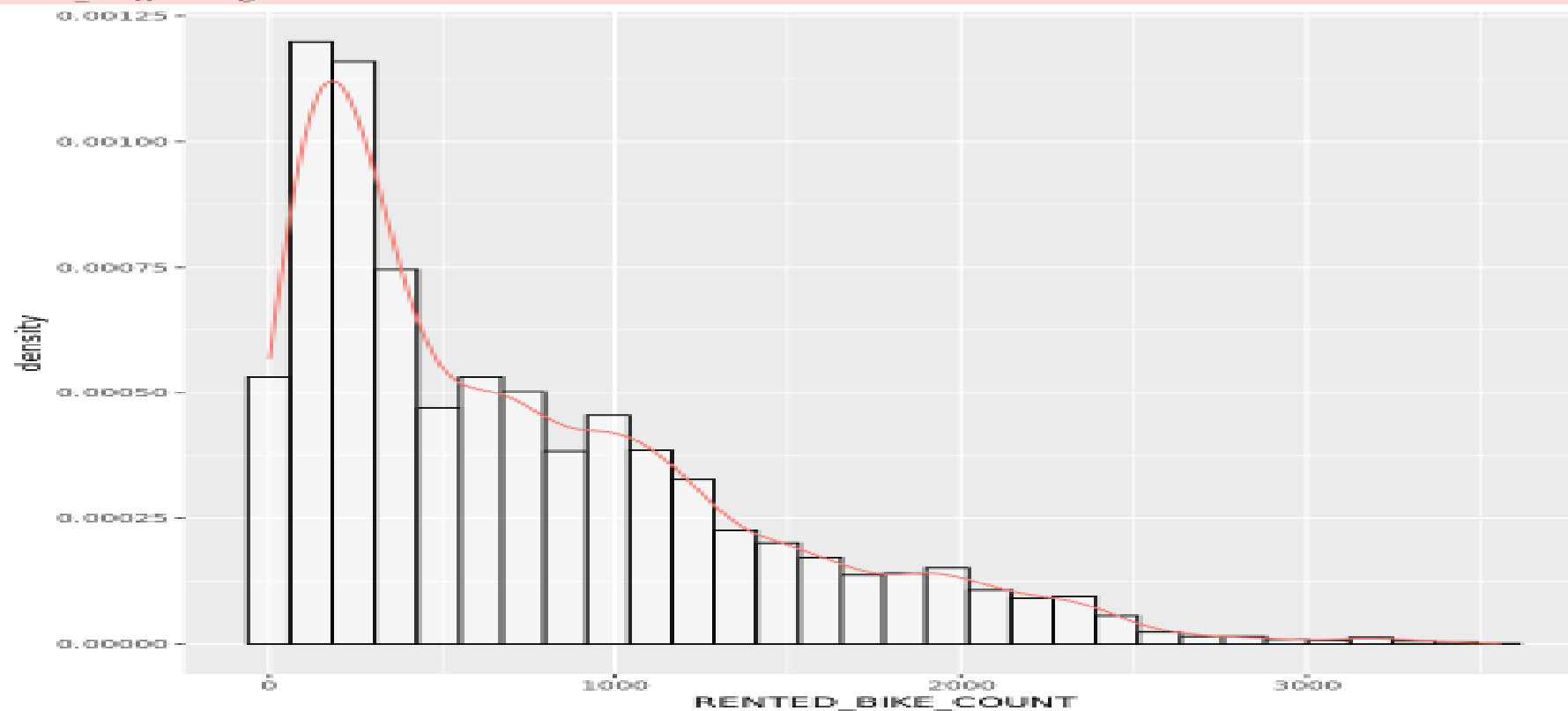


- Correlation between rented bike count vs temperature by seasons



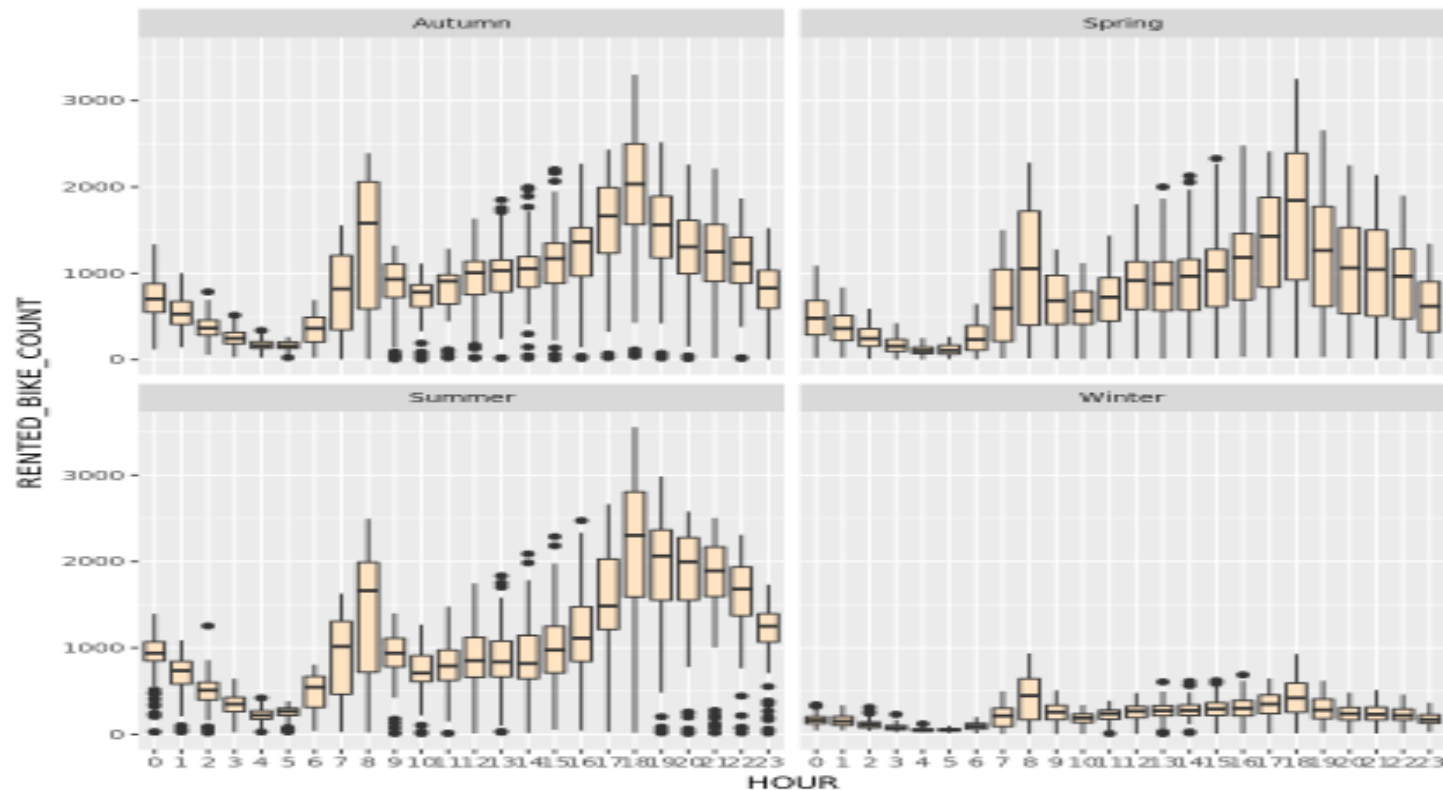
DATA DISTRIBUTION

```
ggplot(seoul_bike_sharing, aes(x = RENTED_BIKE_COUNT)) +  
  geom_histogram(aes(y = ..density..),  
    colour = "black", fill = "white", alpha = 0.5  
  ) +  
  geom_density(aes(color = "blue")) +  
  theme(legend.position = "none")  
  
# stat_bin() using 'bins = 30'. Pick better value with 'binwidth'.
```



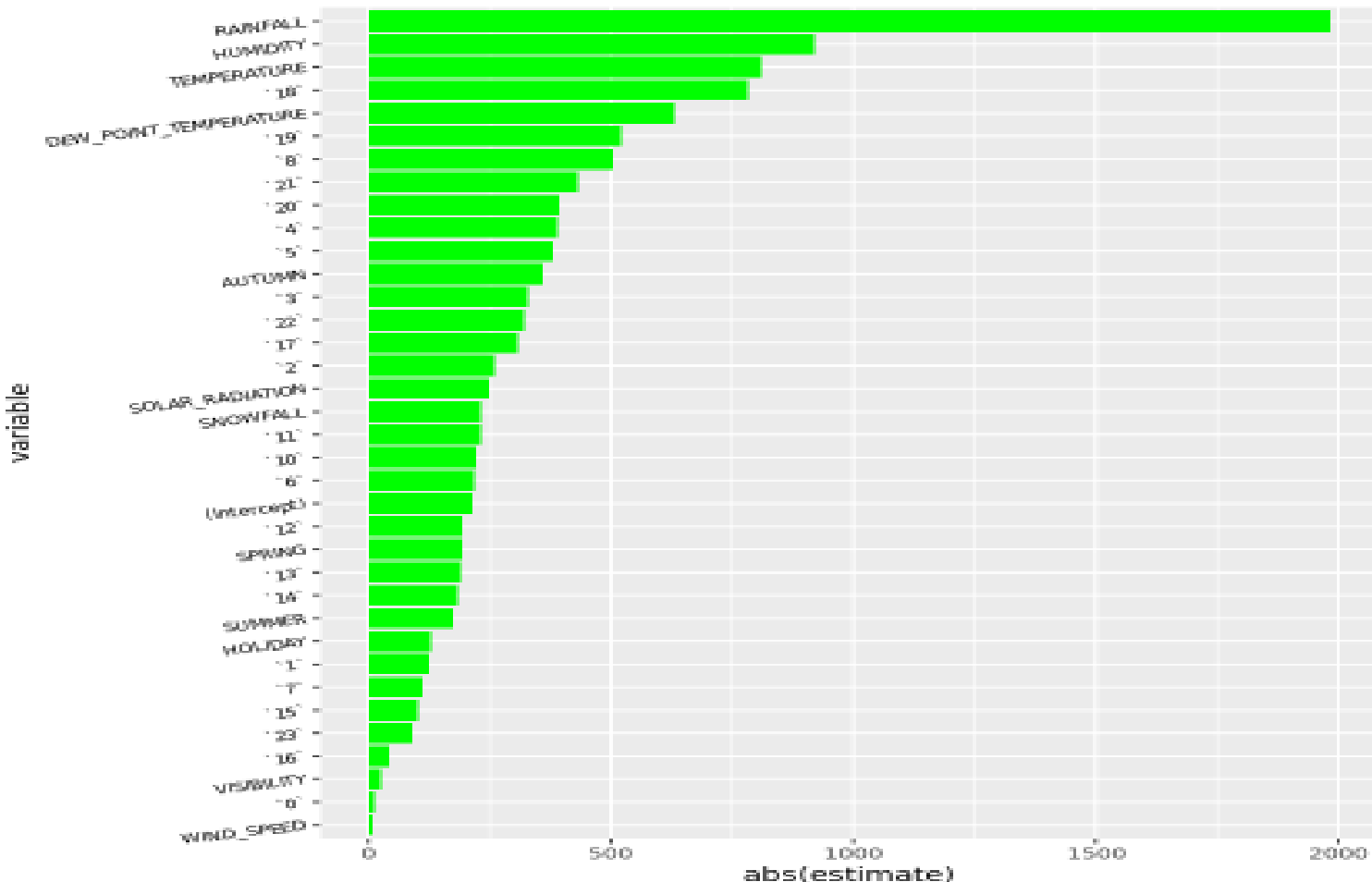
VISUALIZING OUTLIERS

```
ggplot(seoul_bike_sharing, aes(x = HOUR, y = RENTED_BIKE_COUNT)) +  
  geom_boxplot(fill = "bisque") +  
  facet_wrap(~SEASONS)
```



MODEL DEVELOPMENT: LINEAR REGRESSION MODEL WITH R

```
# Visualize the list using ggplot and geom_bar
all_fit %>%
  tidy() %>%
  filter(!is.na(estimate)) %>%
  ggplot(aes(x = fct_reorder(term, abs(estimate)), y = abs(estimate))) +
  geom_bar(stat = "identity", fill = "green") +
  coord_flip() +
  theme(axis.text.y = element_text(angle = 18, colour = "black", size = 7)) +
  xlab("variable")
```



- Weather predictors:

Temp, humidity, wind speed, visibility, dew-point, solar radiation, rainfall, and snowfall

- Date predictors:

Date (year, month, day), hour of day (0, 1, 2,...), holiday/no holiday, and seasons (winter, spring, summer, autumn)

REGRESSION MODEL PREDICTORS

- Weather predictors:

- Intercept: 147.647
 - Temperature: 2452.112
 - Humidity: -895.830
 - Wind Speed: 402.183
 - Visibility: 5.356
 - Dew Point Temperature: -368.982
 - Solar Radiation: -435.703
 - Rainfall: -1771.467
 - Snowfall: 354.761
- **Intercept (147.647)**: If all weather factors are at their minimum values, we start with 147 bike rentals.
 - **Temperature (2452.112)**: For every 1-degree increase in temperature, the number of bike rentals increases by 2452
 - **Humidity (-895.830)**: For every 1% increase in humidity, the number of bike rentals decreases by 896.

REGRESSION MODEL PREDICTORS

All predictor model: weather

- - Intercept: 216.584
- - Temperature: 810.604
- - Humidity: -920.587
- - Wind Speed: -9.313
- - Visibility: 24.368
- - Dew Point Temperature: 632.384
- - Solar Radiation: 249.752
- - Rainfall: -1982.940
- - Snowfall: 232.451

All predictor model: date

- Autumn: 357.978
- Spring: 194.421
- Summer: 172.901
- Winter: (NA, not included)
- Holiday: -129.167
- No Holiday: (NA, not included)
- #On average, being in the autumn season increases bike rentals by 358 compared to other seasons

TESTING THE LINEAR REGRESSION MODEL

Weather Related Model sample results

Predicted Actual

- 274.35 173
- 378.18 78
- 312.99 181
- 336.78 490
- For instance, model predicted 274.35 bikes based on weather, actual bike rented was 173

All predictor model sample results

Predicted Actual

- -78.25 173
- -191.16 78
- -25.99 181
- 251.97 490

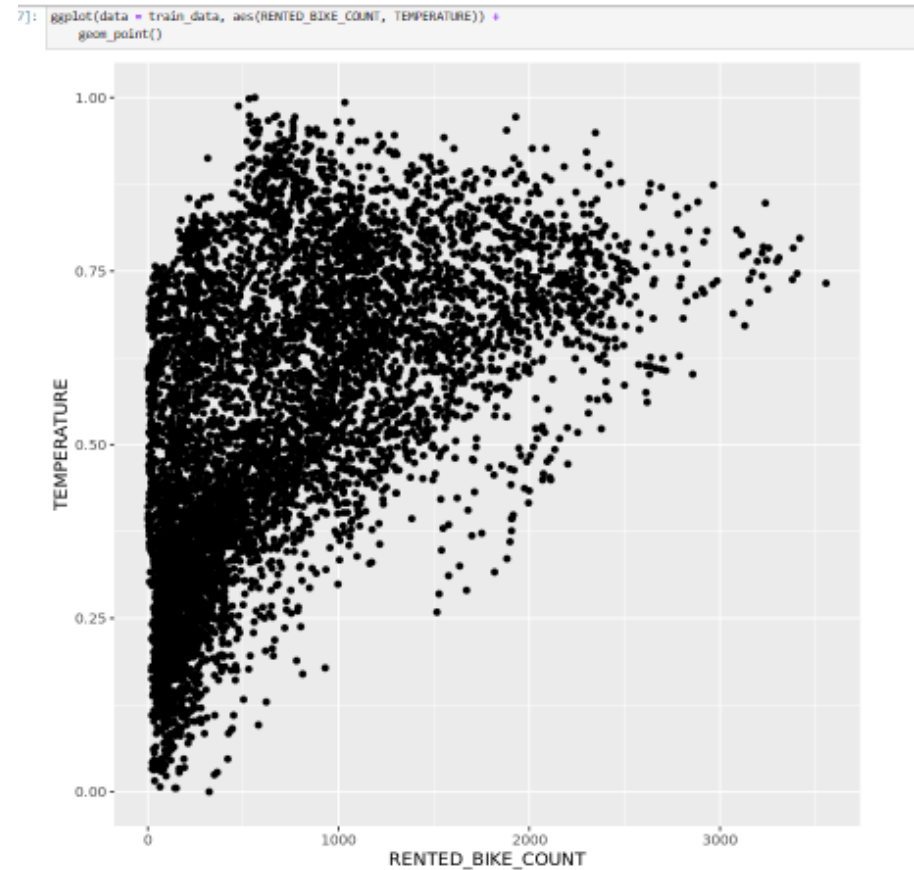
MULTIPLE LINEAR REGRESSION MODEL PERFORMANCE METRICS

- **R-squared** measures how well the model's predictions match the actual data. It ranges from 0 to 1, with higher numbers indicating better performance.
 - **Weather Model R^2** : 0.446 (45% of the variation in bike rentals can be explained by the weather variables).
 - **All Variables Model R^2** : 0.659 (66% of the variation in bike rentals can be explained by all the variables).
- **RMSE (Root Mean Squared Error)** measures how much the model's predictions differ from the actual data, with lower numbers indicating better performance.
 - **Weather Model RMSE**: 470.44 (on average, the predictions are off by about 470 bike rentals).
 - **All Variables Model RMSE**: 369.17 (on average, the predictions are off by about 369 bike rentals).

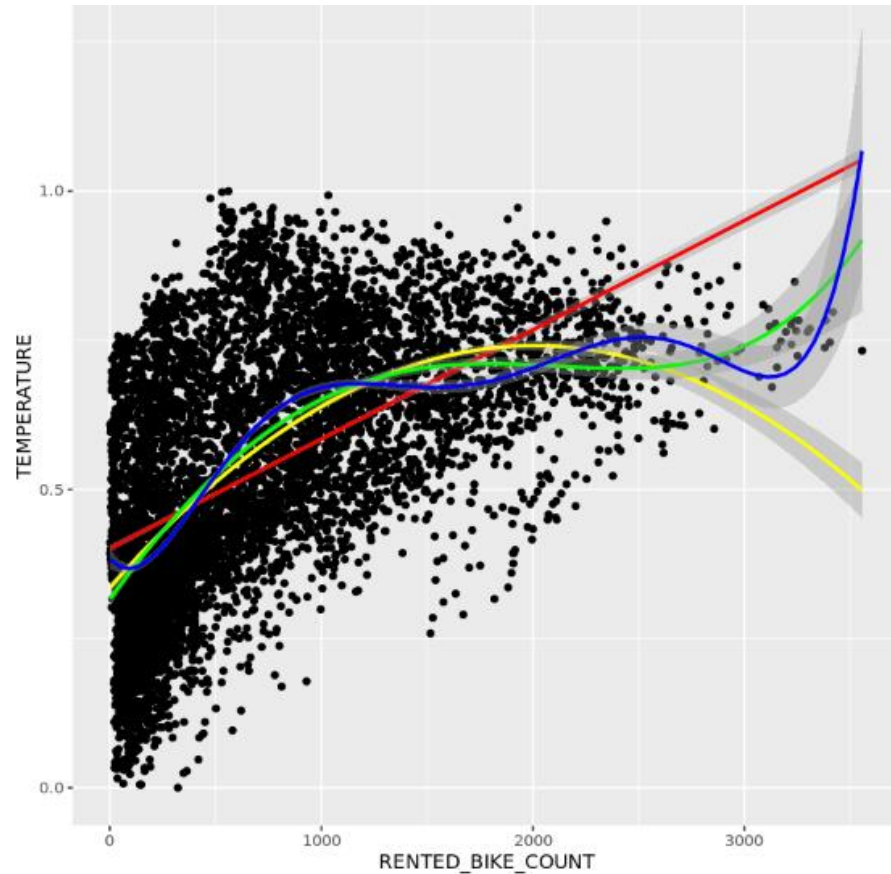
REFINE THE BASELINE REGRESSION MODELS

- A non-linear relationship between bikes rented and temperature.
- Therefore, polynomial orders are needed to improve the model.

```
ggplot(data = train_data, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +  
  geom_point()
```



ADDED POLYNOMIAL ORDER



```
# Plot the higher order polynomial fits
ggplot(data=train_data, aes(RENTED_BIKE_COUNT, TEMPERATURE)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, color="red") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), color="yellow") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 4), color="green") +
  geom_smooth(method = "lm", formula = y ~ poly(x, 6), color="blue")
```

POLYNOMIAL ORDER R-SQUARED AND RMSE

- R-squared
 - 0.735
 - Improved compared to basic linear regression model (had 0.446 for weather-related model and 0.659 for all variables-related model)
- RMSE
 - 326
 - Better than basic linear regression model (470 for weather-related variables and 369 for all variables-related model)

```
# Calculate R-squared and RMSE from the test results
#rsq
rsq_poly <- rsq(lm_ploy_test_results, truth = truth, estimate = .pred)
#rmse
rmse_poly <- rmse(lm_ploy_test_results, truth = truth, estimate = .pred)

print(rsq_poly)
print(rmse_poly)

# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>       <dbl>
1 rsq     standard      0.735
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>       <dbl>
1 rmse    standard     326.
```

ADDING INTERACTIONS TO POLY MODEL

- Interactions between variables can also affect bike renting
- The predictors added and their orders (order degrees obtained by trial and error)
- `formula <- RENTED_BIKE_COUNT ~ . +`

`poly(TEMPERATURE, 6) +`

`poly(HUMIDITY, 6) +`

`poly(DEW_POINT_TEMPERATURE, 4) +`

`RAINFALL * HUMIDITY +`

`HUMIDITY * TEMPERATURE +`

`AUTUMN * HOLIDAY`

PERFORMANCE METRICS FOR THE POLY + INTERACTIONS MODEL

```
# Calculate R-squared and RMSE for the new model to see if performance has improved
# There are negative results in the prediction column of both train and test data.
# Remove these values by replacing with zero to normalize the data

# Convert negative prediction results to zero
lm_poly_test_results <- lm_poly_test_results %>%
  mutate(.pred = ifelse(.pred < 0, 0, .pred))

# Calculate R-squared and RMSE
# train data
# Calculate R-squared and RMSE for training data
rsq_poly_train <- rsq(lm_poly_train_results, truth = RENTED_BIKE_COUNT, estimate = .pred)
rmse_poly_train <- rmse(lm_poly_train_results, truth = RENTED_BIKE_COUNT, estimate = .pred)

# test data
rsq_poly <- rsq(lm_poly_test_results, truth = RENTED_BIKE_COUNT, estimate = .pred)
rmse_poly <- rmse(lm_poly_test_results, truth = RENTED_BIKE_COUNT, estimate = .pred)
```

- **Training Data:** Used to create the model with interaction effects
 - **R-squared:** 0.759 (76% of the variation in bike rentals is explained by our model).
 - **RMSE:** 318 (predictions are off by about 318 rentals).
- **Test Data:** Used to test the predicting ability of the model
 - **R-squared:** 0.740 (74% of the variation in bike rentals is explained by our model on new data).
 - **RMSE:** 323 (our predictions are off by about 323 rentals).
- **Summary explanation:** R-squared fell, RMSE increased. Still both metrics have improved compared to poly model without interaction effects (R-squared = 0.735 and RMSE = 326)

MODEL REGULARIZATION WITH LASSO AND RIDGE

- Notice higher r-squared and lower rmse in lasso compared to ridge
- Lasso tunes the model better than ridge

```
> print(rsq_lasso)
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rsq      standard      0.756
> print(rmse_lasso)
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rmse     standard     314.
> print(rsq_ridge)
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rsq      standard      0.753
> print(rmse_ridge)
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>    <chr>         <dbl>
1 rmse     standard     317.
```

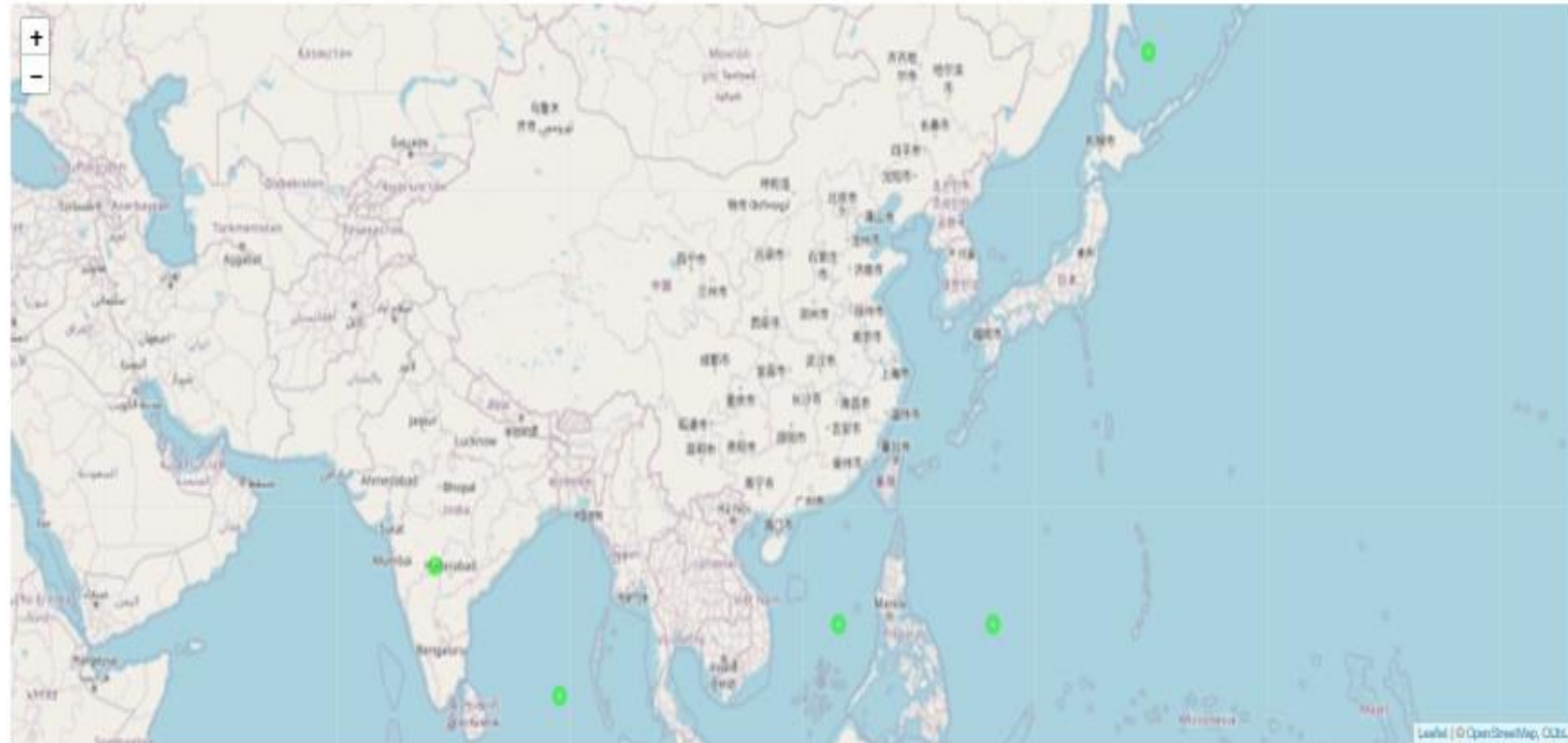
DASHBOARD: SHINY APP VIEW

RStudio Desktop - P...

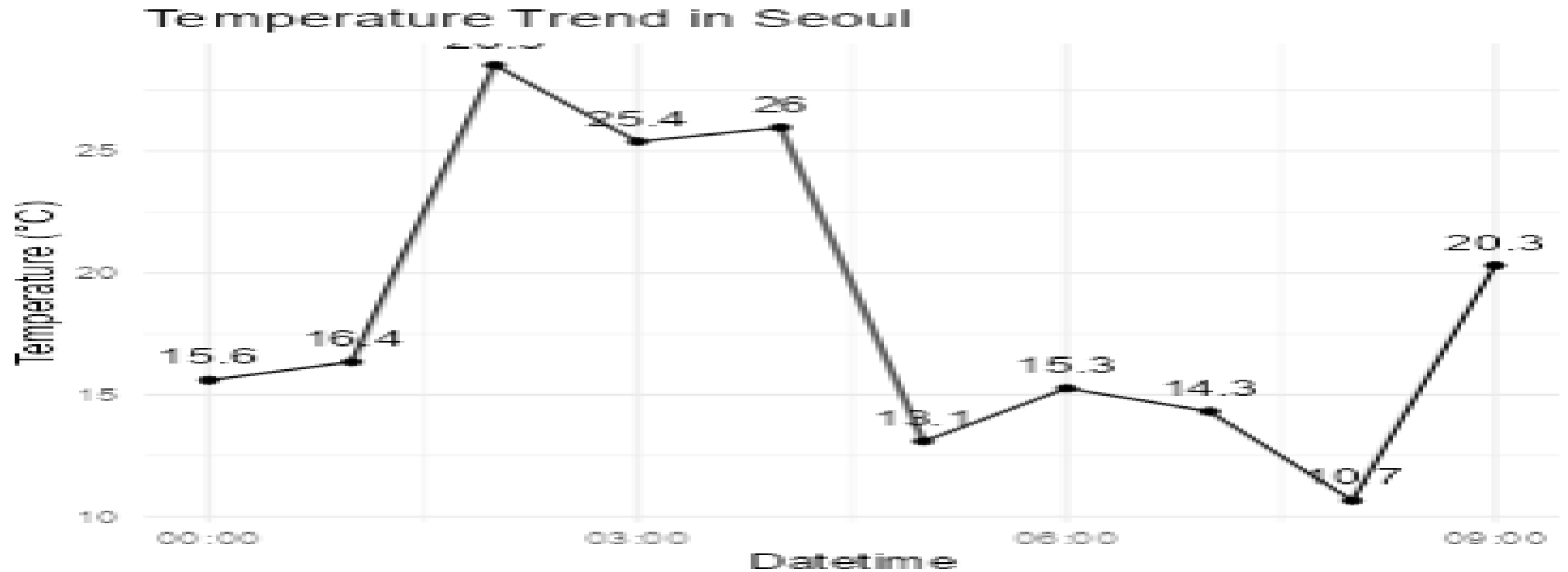
Bike Sharing Demand Prediction

Select City

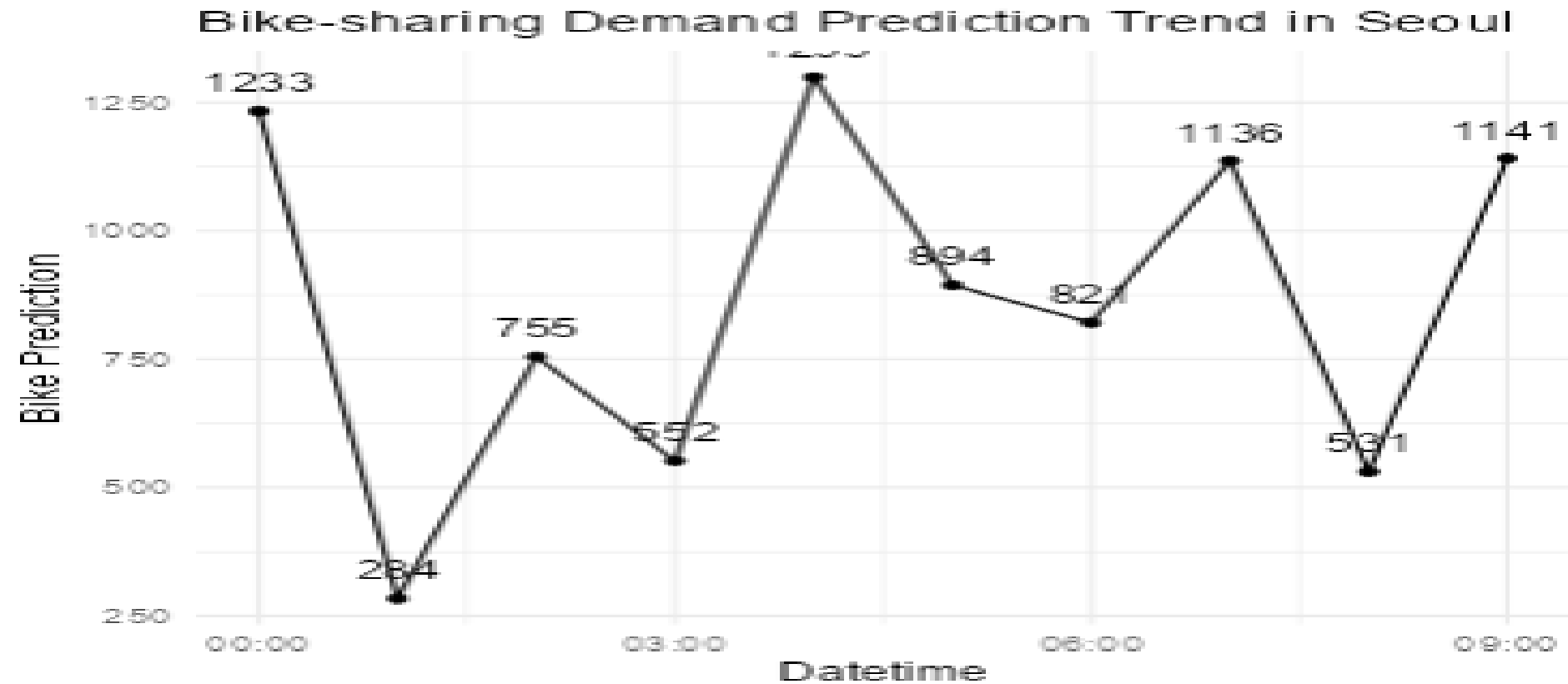
All



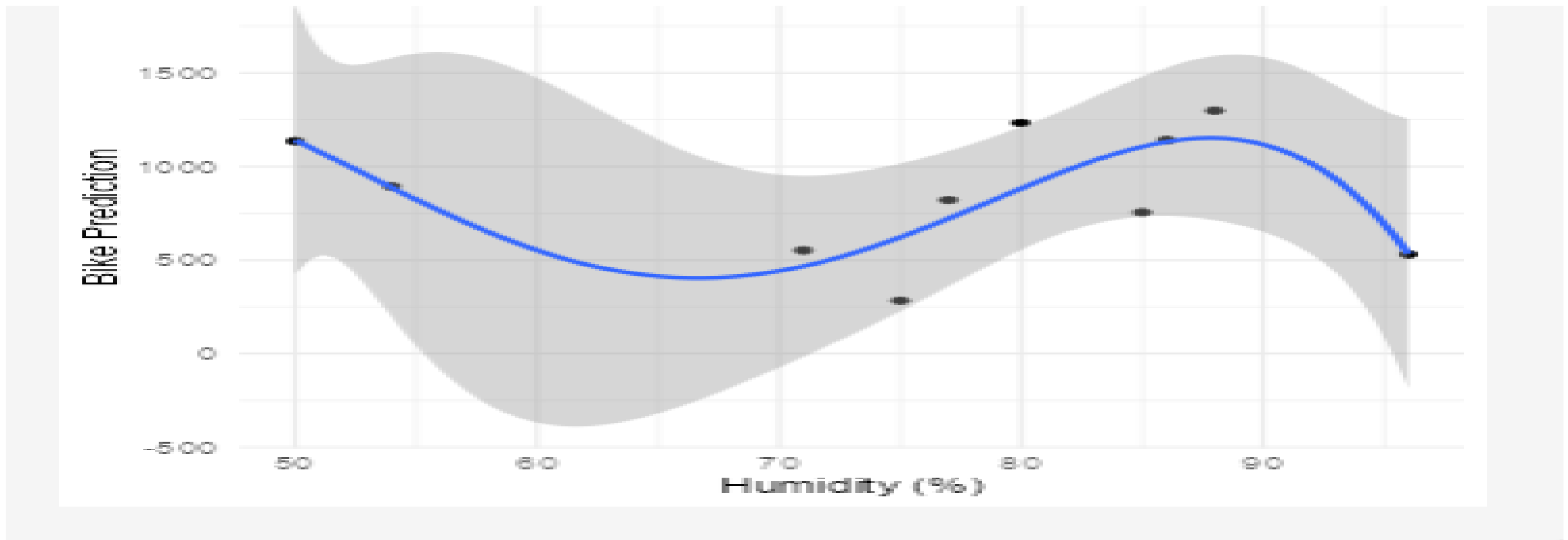
SEOUL BIKE SHARING: TEMPERATURE



SEOUL BIKE SHARING: DATE



SEOUL BIKE SHARING: HUMIDITY



DISCUSSION

Weather predictors

Temperature and humidity are the strongest indicators.

For every 1-degree increase in temperature, the number of bike rentals increases by 2452

- **Humidity (-895.830):** For every 1% increase in humidity, the number of bike rentals decreases by 896.

Date predictors

- Analysis shows Autumn: 357.978, Spring: 194.42, and Summer: 172.901 as the strongest date predictors in the all-variable model

DISCUSSION

- Results indicate all variable related model performs better than weather-based model.
- The R-squared and RMSE metrics indicate how well a model predicts based on historical data.
- The larger the R-squared and smaller the RMSE, the better the model.
- All weather-based model has a higher R-squared and RMSE compared to all-variables-model.
- Non-linear relationships also exist.
- Polynomial regression handle non-linear relationships.
- Considering the leading predictor variable, temperature, the analysis shows higher polynomial orders like 6 perform better compared to smaller degrees like 4 and 2.
- The R-squared of the polynomial order model is 0.735.
- This is an improvement compared to basic linear regression model (which had 0.446 for weather-related model and 0.659 for all variables-related model)

INTERACTIONS + POLYNOMIAL DEGREES

- Other than non-linearity, variables can interact to affect the demand for bike sharing across locations and time.
- For example, temperature can interact with humidity to facilitate bike sharing or interact with time of day or season.
- **R-squared:** 0.759 (76% of the variation in bike rentals is explained by our model).
 - **RMSE:** 318 (predictions are off by about 318 rentals).
- **Test Data:** Data used to test the predicting ability of the model
 - **R-squared:** 0.740 (74% of the variation in bike rentals is explained by our model on new data).
 - **RMSE:** 323 (our predictions are off by about 323 rentals).
- R-squared fell, RMSE increased. Still both metrics have improved compared to poly model without interaction effects (R-squared = 0.735 and RMSE = 326)

FINE-TUNING MODELS WITH LASSO

- The analysis indicates the lasso technique tunes the model better than ridge regularization.
- Lasso has r-squared 0.756 and RMSE 314 compared to ridge's 0.753 and 317.
- Lasso tunes better because of feature selection, sparsity, and regularization strength.
- Feature selection: Lasso shrinks some coefficients exactly to zero, which improves interpretability and performance. Ridge retains all coefficients when reducing them towards zero
- Sparsity: By setting some coefficients to zero, lasso leads to models that are simpler and less prone to overfitting
- **Regularization strength:** The L1 (lasso) penalty tends to have a stronger regularizing effect compared to L2 (ridge), which is less aggressive when shrinking coefficients

DASHBOARDING WITH SHINY

- **Benefits of using shiny app for visualization:**

Interactivity:

- **Engagement:** Interactive elements such as sliders, filters, and clickable charts allow users explore the data themselves and understand the findings better.
- **Customization:** Stakeholders can tailor the visualizations to their specific interests or questions, providing a more personalized and relevant experience.
- **Clarity and Simplicity:**
- **Intuitive Design:** Clear, intuitive design helps stakeholders grasp the insights quickly without needing to wade through complex data.
- **Focus on Key Metrics:** Highlighting the most important metrics and insights ensures that the main messages are not lost in a sea of data.

- **Real-Time Data:**

- **Up-to-Date Information:** Presenting real-time or near-real-time data can be crucial for decision-making processes that rely on the latest information.
- **Responsive Analysis:** Stakeholders can see the immediate impact of changing variables or parameters, aiding in scenario analysis and strategic planning.
- **Ease of Access:**
- **Web-Based Accessibility:** Shiny apps are web-based, allowing stakeholders to access the visualizations from any device with a web browser without needing special software or technical setup.
- **Shareability:** The ability to easily share a link to the Shiny app means that stakeholders can access the findings at their convenience and share them with others.

EXPLAINING SHINY APP

- The Shiny app visualizes bike-sharing demand prediction for three cities: Seoul, Suzhou, London, New York, and Paris. The app contains three types of plots for each city:
- **Temperature Trend:** Shows the variation in temperature over time.
- **Bike-sharing Demand Prediction Trend:** Displays the predicted demand for bike-sharing over the same time period.
- **Humidity vs. Bike-sharing Demand Prediction:** Depicts the relationship between humidity and bike-sharing demand with a regression line to indicate trends.

SEOUL: SHINY APP DISCUSSION AND FINDINGS



Seoul



Temperature Trend: The temperature fluctuates significantly, peaking at around 29.6°C and dropping to about 10.9°C.



Bike-sharing Demand Prediction Trend: The bike-sharing demand shows an upward trend initially, peaking at 114.6, then a decline, and another peak towards the end at 144.6.



Humidity vs. Bike-sharing Demand Prediction: There's a negative correlation between humidity and bike-sharing demand; as humidity increases, bike-sharing demand tends to decrease.

- **Conclusion:** Bike-sharing demand is negatively influenced by humidity and positively correlated with moderate temperatures.
- **Recommendation:** To increase bike-sharing usage, strategies could include providing more covered bike stations to protect riders from humidity and promoting bike-sharing during moderate temperature days.

SUZHOU: SHINY APP DISCUSSION AND FINDINGS

- **Temperature Trend:** Suzhou's temperature also varies widely, with the highest at 26.9°C and the lowest at 21.1°C.
- **Bike-sharing Demand Prediction Trend:** The demand fluctuates, with two notable peaks at 104.7 and 87.3, and a sharp drop before another peak at 160.4.
- **Humidity vs. Bike-sharing Demand Prediction:** There's a positive correlation between humidity and bike-sharing demand; as humidity increases, the demand also increases, though with significant variance.

Conclusion: The demand for bike-sharing increases with both temperature and humidity, although the temperature range is relatively narrower.

Recommendation: Enhance bike-sharing services during warmer and more humid days, possibly by increasing the number of available bikes and ensuring good maintenance to accommodate higher demand.

LONDON: SHINY APP DISCUSSION AND FINDINGS



London



Temperature Trend: London's temperature changes markedly, reaching a high of 24.6°C and a low of 10.9°C.



Bike-sharing Demand Prediction Trend: The bike-sharing demand peaks at 590 before declining and rising again to 496.



Humidity vs. Bike-sharing Demand Prediction: There's a positive correlation between humidity and bike-sharing demand, similar to Suzhou, indicating higher demand at higher humidity levels.


- **Conclusion:** Bike-sharing demand is higher during moderate temperatures and increases with humidity.
- **Recommendation:** Encourage bike-sharing usage during humid days, possibly by integrating humidity-tolerant bike designs or features that improve comfort in humid conditions. Additionally, increasing promotional activities during these periods can boost usage.

CONCLUSION

Weather-Adaptive Strategies: Implement weather-adaptive strategies such as dynamic pricing based on weather conditions to encourage bike-sharing.

A light purple arrow pointing downwards from the first box to the second box.

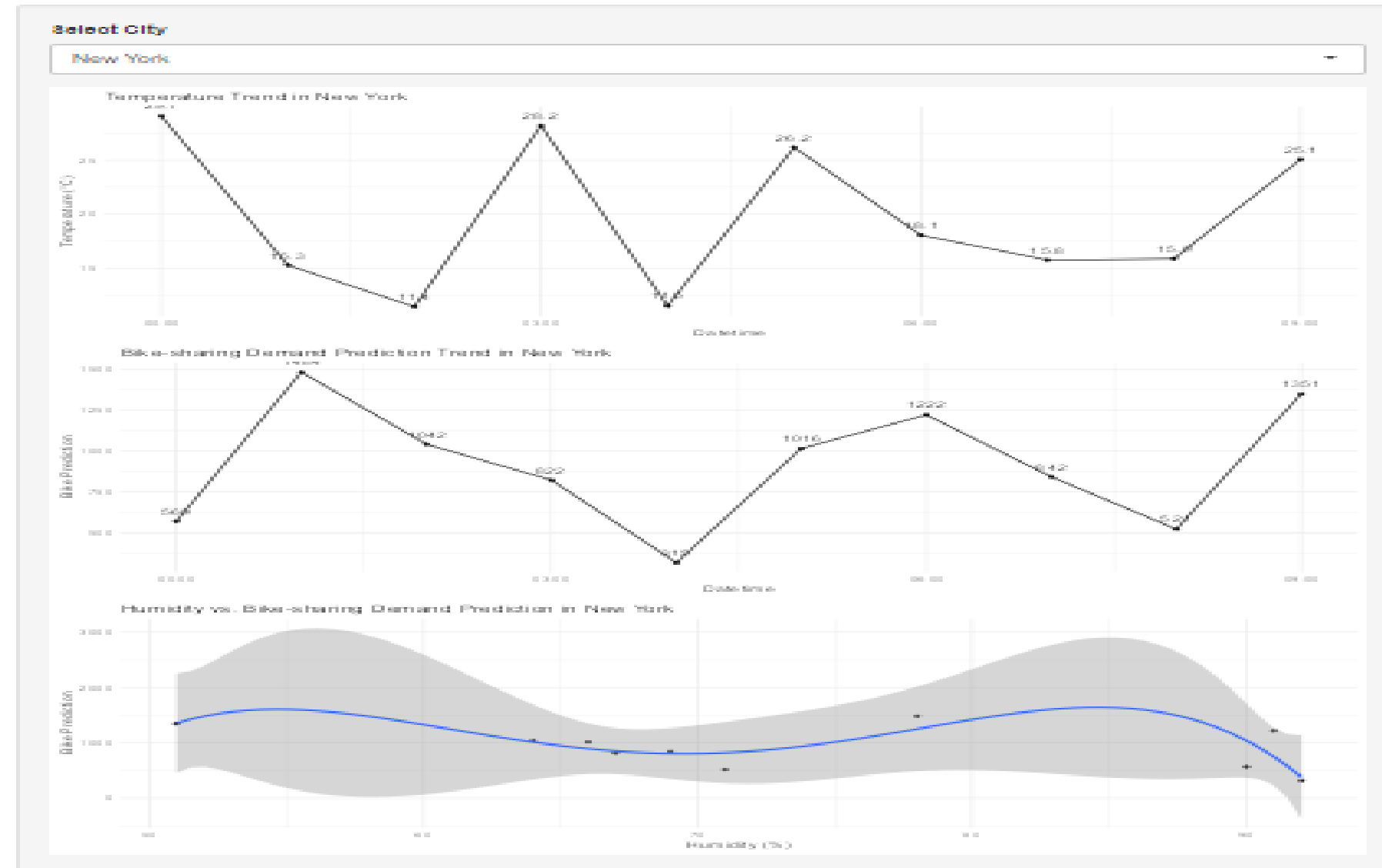
Infrastructure Improvements: Improve infrastructure to make bike-sharing more appealing during less favorable weather conditions, such as covered bike paths and better lighting.

A light purple arrow pointing downwards from the second box to the third box.

Promotional Campaigns: Conduct targeted promotional campaigns based on weather forecasts to maximize bike-sharing usage.

PREDICTION: TEMPERATURE, DATE, AND HUMIDITY

Bike Sharing Demand Prediction

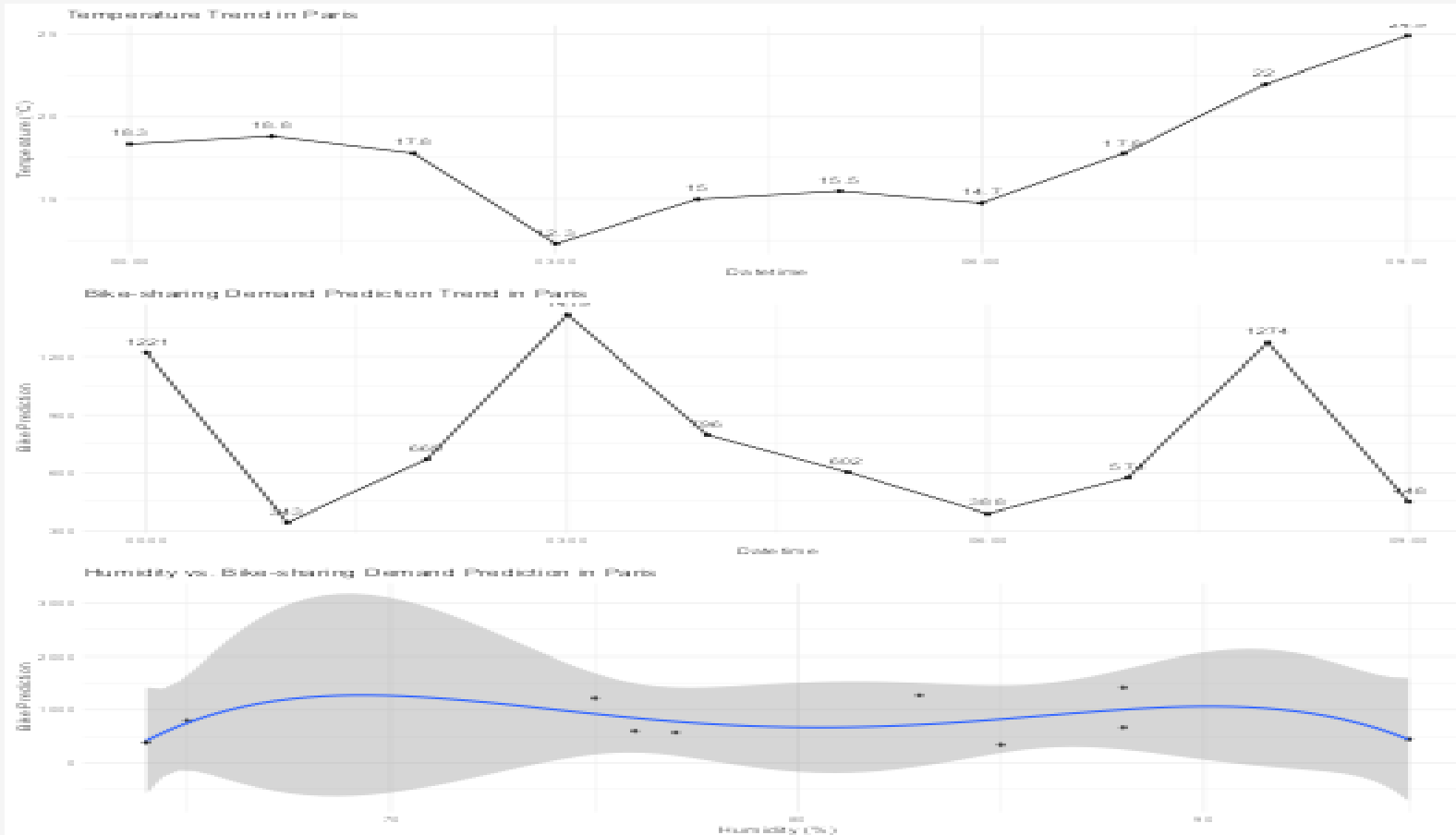


APPENDIX 2: PARIS BIKE PREDICTION: TEMPERATURE, DATE, AND HUMIDITY

Bike Sharing Demand Prediction

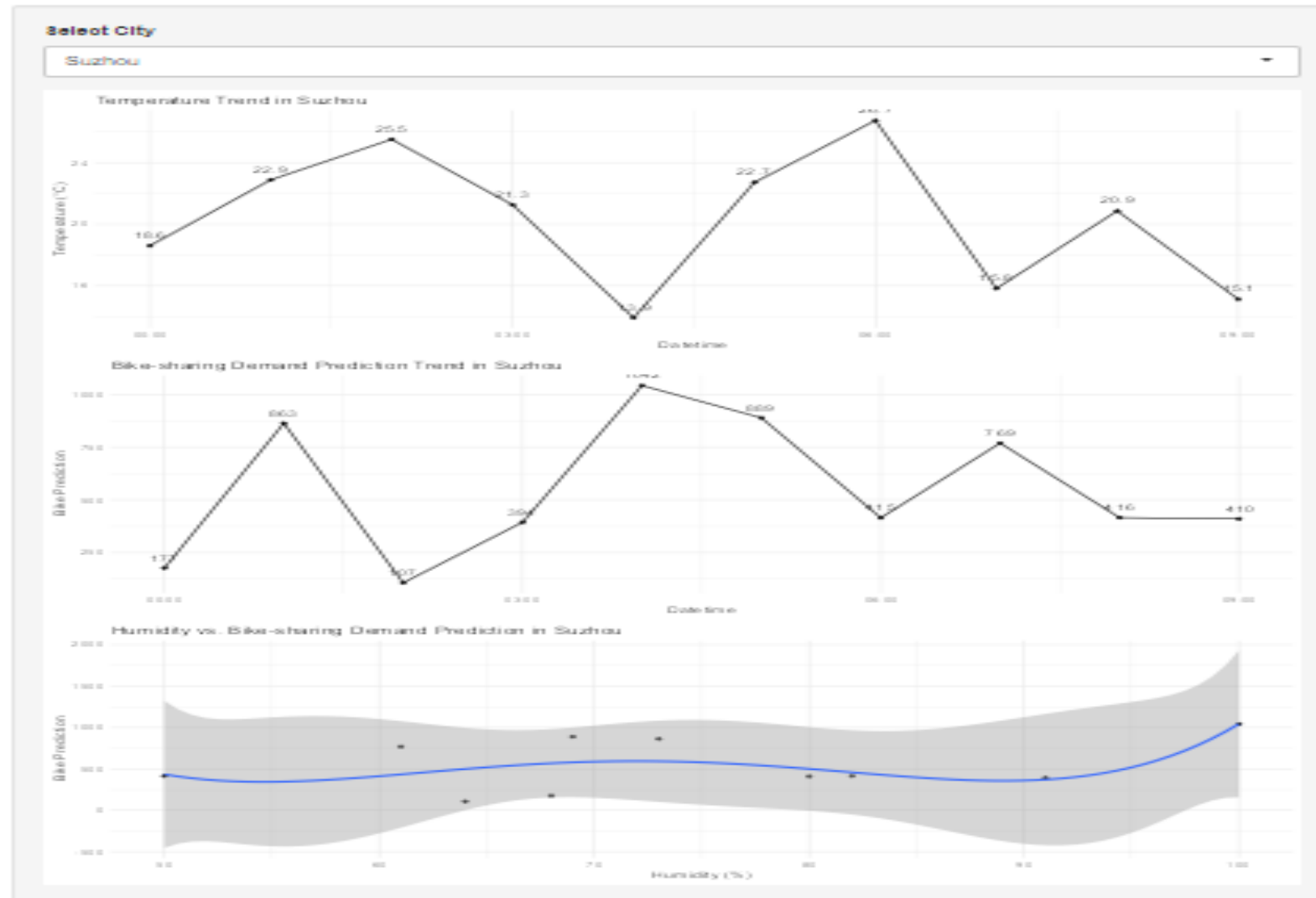
Select City

Paris

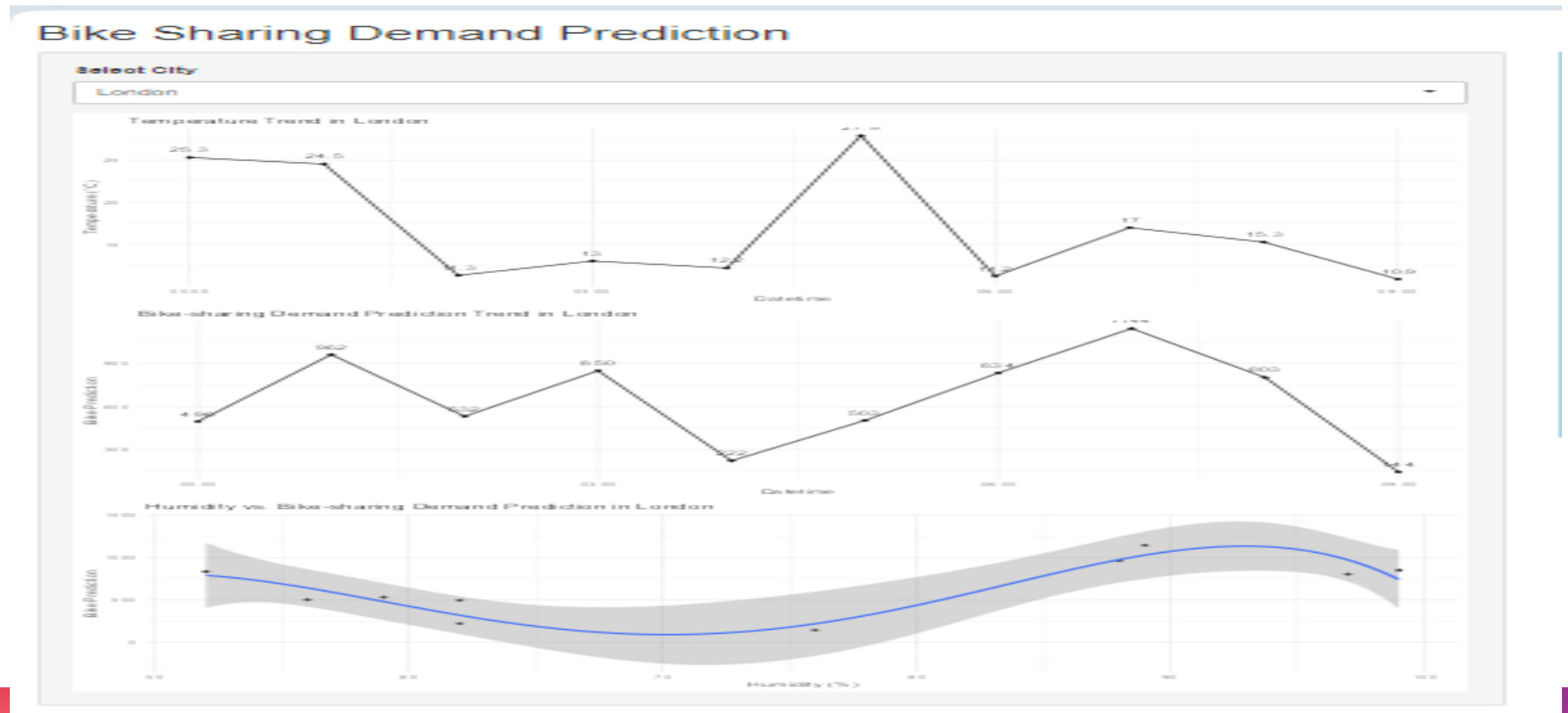


APPENDIX 3: SUZHOU BIKE PREDICTION: TEMPERATURE, DATE, AND HUMIDITY

Bike Sharing Demand Prediction



APPENDIX 4: LONDON BIKE PREDICTION: TEMPERATURE, DATE, AND HUMIDITY



CODE SNIPPETS: DATA COLLECTION

```
url <- "https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems"
# Get the root HTML node by calling the `read_html()` method with URL
root_node <- read_html(url)
root_node
table_nodes <- html_nodes(root_node, "table")
table_nodes
```

```
# Convert the bike-sharing system table into a dataframe
bike_df <- html_table(table_nodes, fill = TRUE)[[1]]
head(bike_df, n=2)
```

A dataframe: 2 × 10

	Country	City	Name	System	Operator	Launched	Discontinued	Stations	Bicycles	Daily ridership
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	Albania	Tirana[5]	Ecovolis			March 2011		8	200	
2	Argentina	Buenos Aires[6][7]	Ecobici	Serttel Brasil[8]	Bike In Baires Consortium[9]	2010		400	4000	21917

```
# Export the dataframe into a csv file
write.csv(bike_df, file = "bike_data.csv", row.names = FALSE)
```

LASSO VS RIDGE CODE

```
# Split the data
set.seed(1234)
data_split <- initial_split(bike_sharing_df, prop = 4/5)
train_data <- training(data_split)
test_data <- testing(data_split)

# Lasso Regression
lasso_spec <- linear_reg(penalty = 0.1, mixture = 1) %>%
  set_engine("glmnet") %>%
  set_mode("regression")

# Ridge Regression
ridge_spec <- linear_reg(penalty = 0.1, mixture = 0) %>%
  set_engine("glmnet") %>%
  set_mode("regression")

# Define formula with polynomial and interaction terms
formula <- RENTED_BIKE_COUNT ~ . +
  poly(TEMPERATURE, 6) +
  poly(HUMIDITY, 6) +
  poly(DEW_POINT_TEMPERATURE, 4) +
  RAINFALL * HUMIDITY +
  HUMIDITY * TEMPERATURE +
  AUTUMN * HOLIDAY

# Fit Lasso model
lasso_fit <- lasso_spec %>%
  fit(formula, data = train_data)

# Fit Ridge model
ridge_fit <- ridge_spec %>%
  fit(formula, data = train_data)

# Make predictions on test dataset
lasso_preds <- predict(lasso_fit, new_data = test_data) %>%
  mutate(truth = test_data$RENTED_BIKE_COUNT)

ridge_preds <- predict(ridge_fit, new_data = test_data) %>%
  mutate(truth = test_data$RENTED_BIKE_COUNT)

-- . . . .
```