Bike Maintenance Modeling

Group 4: Hritik Arasu, Audrey Cho, Mingcheng Du, Kurt Ji

Client: Darren Glosemeyer

3 May 2023

1. Introduction
2. Overview

Our client, for this consulting project, is a major bike rental company located in Seoul, South Korea. We were given data taken from the autumn months of 2018. These months were September, October, and November. The data obtained initially included 2184 observations with 14 different variables giving the hourly data of bike usage for each day of the month given. Once we took a look at the data and the tasks, our group did decide to change some of the variables which we will get into during the data portion of this report.

1. Tasks

Our goal for this project is to create the best maintenance schedule for a biking company that will follow bike usage. In order to do so, we want to see when bikes are most used or not when looking at the different variables such as temperature, visibility, and many more. With the end result, we would like to propose a model that could be followed to write a fall/autumn maintenance schedule for these bikes. We recognize that it is important to look at what sort of days result in the least amount of bike usage so that we can schedule the bike maintenance most efficiently for the company.

1. Data

To begin looking at the data, we initially changed the format of how the dates were written. Initially they were written as day/month/year which we set to month/day/year. Then, we only kept the observations with a functioning day of yes and then removed that variable because we only wanted to look at the data for when the bikes were in use. We also thought to remove the season since we were only looking at months within Autumn. Finally, we added what day of the week each day fell on as well as wrote out which month each observation was taking place. At this point, we began to move into exploratory data analysis and figuring out some descriptive statistics.

1. Graphs and Charts

**Figure 1**

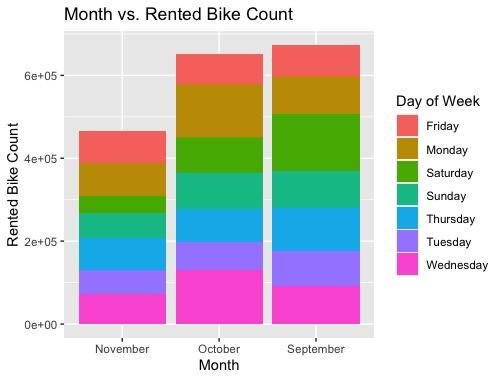
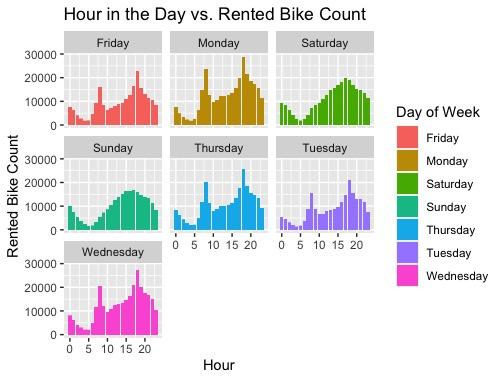
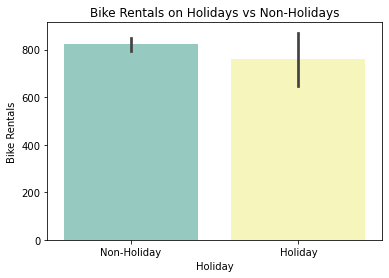


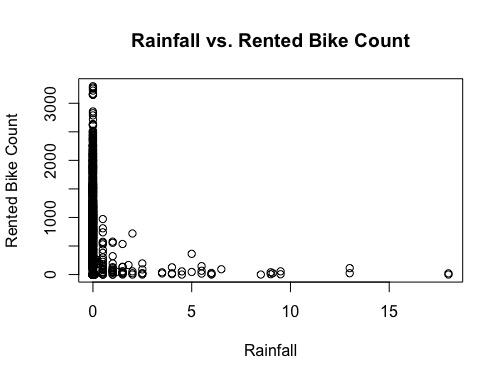
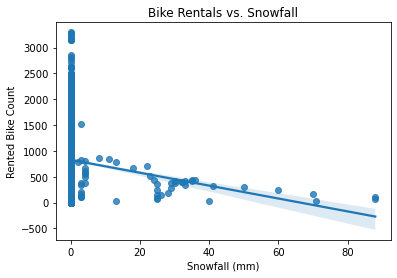
Figure 1 is a bar graph that shows the amount of bikes rented per month. It also color codes the amount of bikes used per day of the week within the months. Some important things we thought to note here was that November had the least amount of bikes being used. We hypothesized that this could be due to the fact that November is a colder month compared to the others. In October, Wednesday and Monday had the most use of bikes while in September, we see that Saturday had the most bike usage.

**Figure 2**

In figure 2 we graphed the hour of the day against the amount of rented bikes. We also divided it by the day of the week which is color coated for easier viewing purposes. As you can see, the weekdays follow roughly the same trends with peaks at 8am and then 6pm. We believed that this was due to the fact that these were the times people would be commuting to and from work.

**Figure 3**

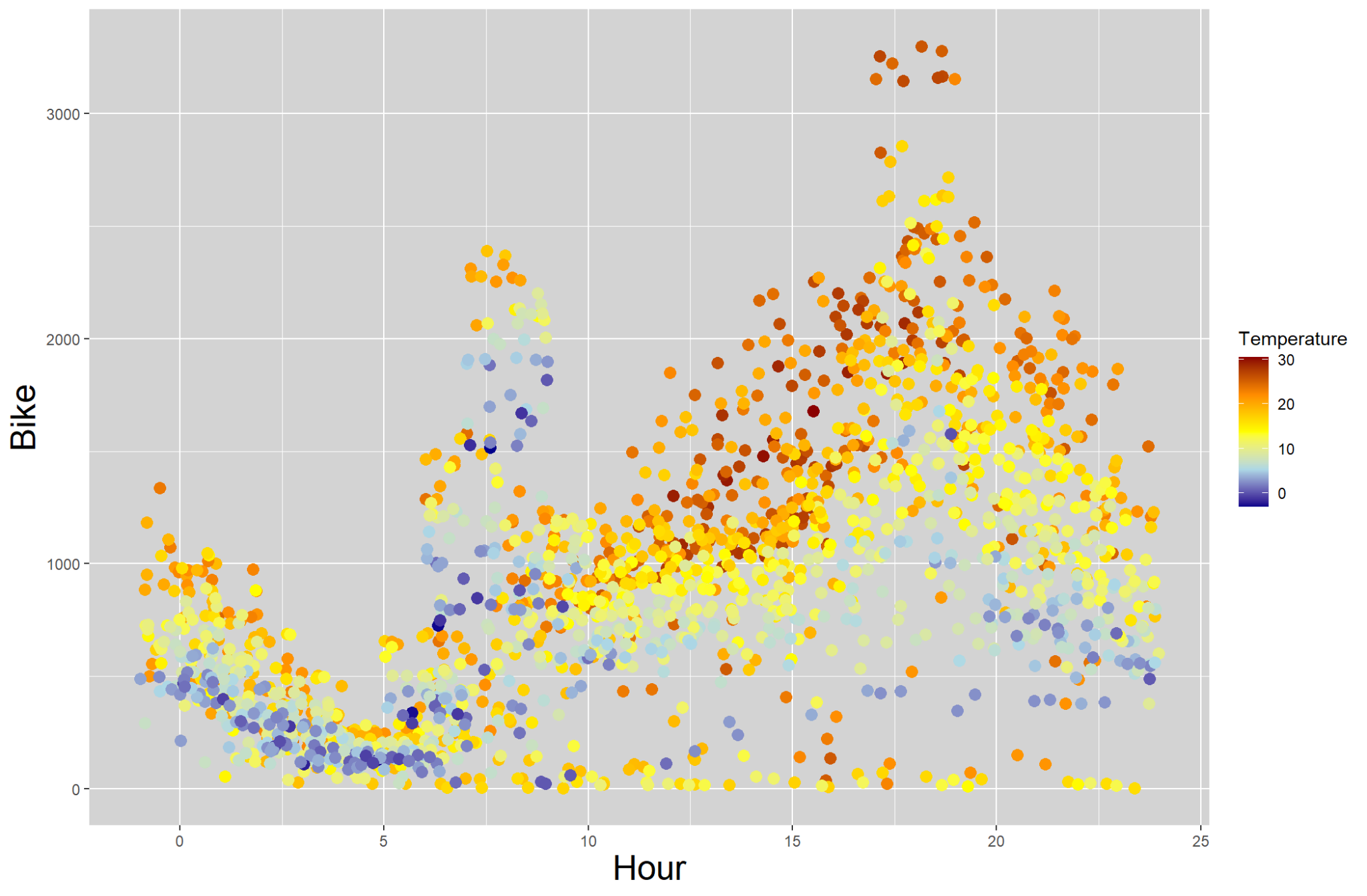
For figure three, we are looking at holidays and non holidays versus the rented bike count. There were significantly less bikes rented during the holidays however it is important to note that there were only 2 holidays given in our data. Because of this, we would think that there being a holiday is overall not too significant.

**Figure 4 & 5**

In figures 4 & 5 we graphed our weather variables against the rented bike count. One thing we noted is that there weren’t many observations that contained rain or snow. This is obviously due to the fact that neither of these really occur during the autumn seasons. However, with the few observations we did get with them, we can see a slightly negative correlation between the two. This means that whenever it rained or snowed, there would be less bike usage.

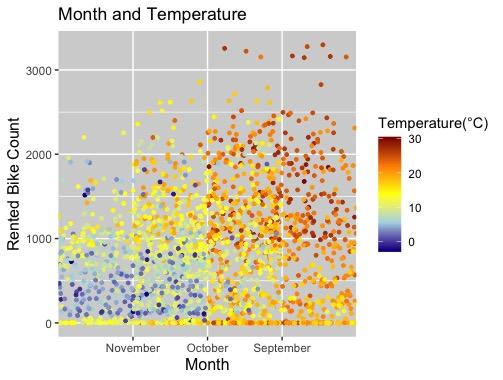
**Figure 6**

**Hour of Day vs. Rented Bike Count with Temperature**



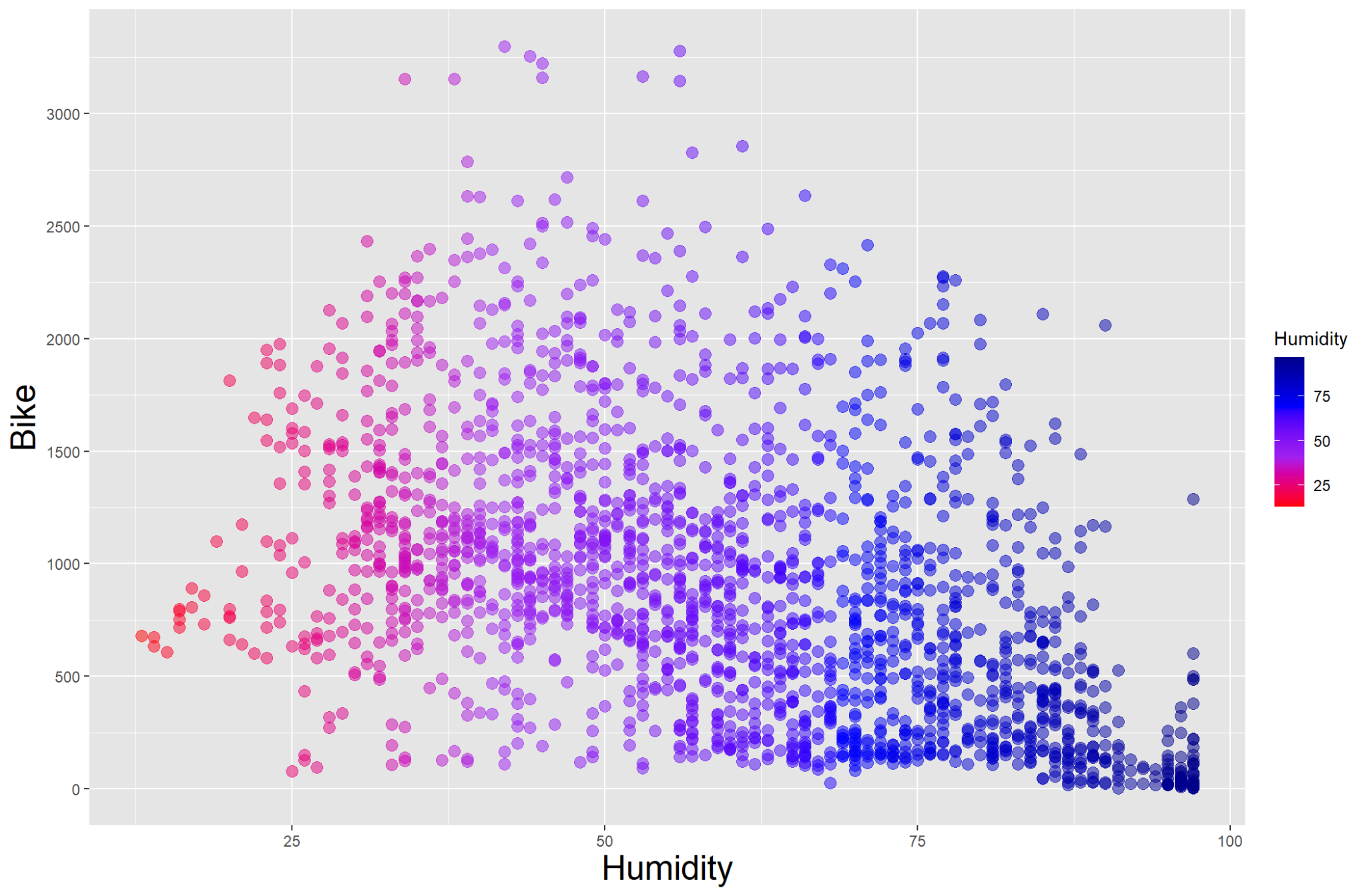
In figure 6, we plotted the hour of the day vs. the rented bike count with temperature as well. This way we were able to see that as the day gets warmer, the more bikes get rented. We see the rise and fall starting from hour 10 until hour 24. Even though 8am is usually a bit colder, we still see a spike there since that is a usual commute time.

**Figure 7**



In figure 7 we are looking at the month vs the rented bike count. We added the coloring of the temperature so that we can also add that into the comparison. As you can see more clearly here, November had the least amount of bike usage with also the coldest temperatures. All the while, September has the most usage as well as the highest temperatures.

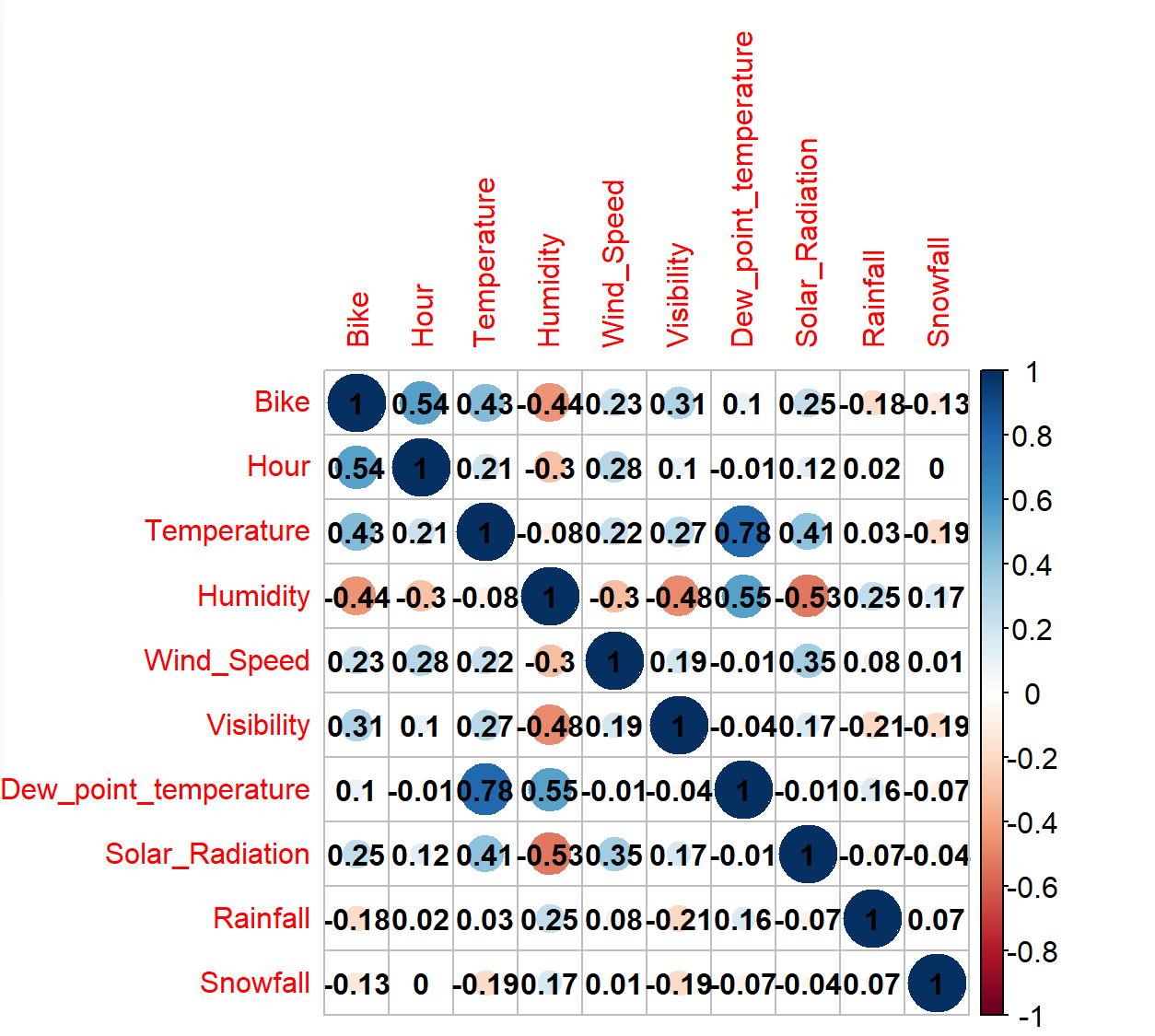
**Figure 8**

Humidity vs. Rented Bike Count

In figure 8, we see a bit of a negative correlation curve when there is humidity. This

means that when it was more humid, there were less bikes rented. This would make sense

because in general, most people would not want to bike when it’s humid outside.

**Figure 9**

Finally, a variable correlation graph. From this, we manually removed irrelevant variables

like dew-point temperature and from this we also used specific metrics to remove others

that we thought would change the data.

1. **Methods**
2. **Linear Regression**

To predict bike demand at different levels of predictor variables, we started by trying a regression model approach. In other words, we wanted to understand the relationship between bike demand and the rest of the predictor variables by finding a line that can best fit the data. The simplest model to start with is the linear regression model, which means fitting a straight line by assuming linear relationships between variables.

* **Model Performance**

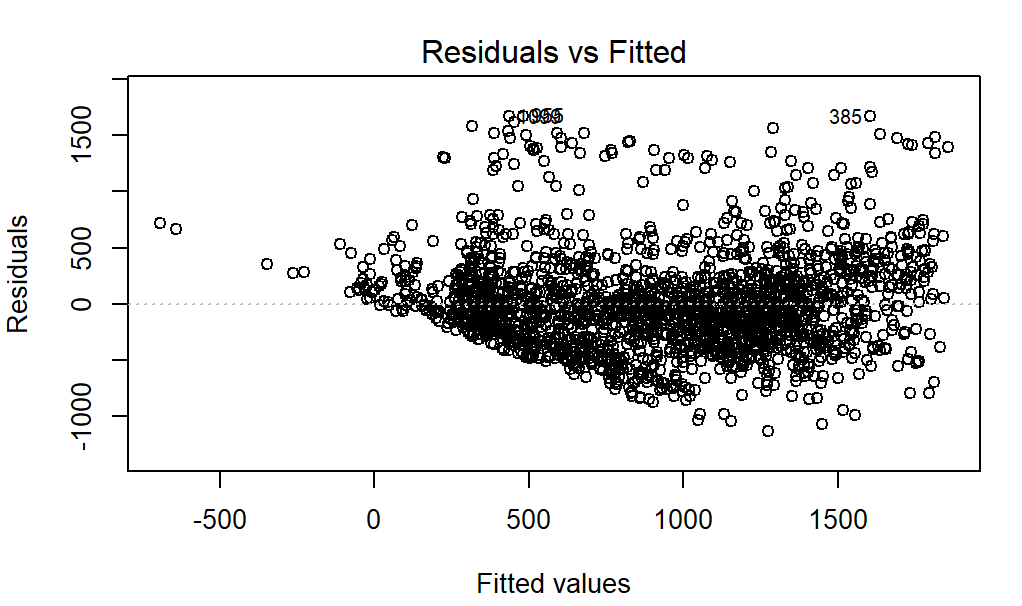
After removing irrelevant variables such as ‘Dates’, ‘Functioning Days’, and ‘Seasons’, we fit a linear regression model with the rest of the variables. The results showed that hour, humidity, dew point temperature, solar radiation, rainfall, holiday, and weekdays were significant. By significant, we mean that these variables have some significant impact on how many bikes were rented. However, the model has a relatively low R-squared value at 0.52, indicating that our model might not be good enough to explain the variation of bike usage by the predictor variables.

* **Multicollinearity**

In addition, we also observed that there exists a collinearity issue in the data, which means some redundant variables behave similarly and they may tell the same information. This issue can be crucial for linear regression as it is a violation of the underlying assumption. A good indicator to assess collinearity is VIF, which stands for the Variance Inflation Factor. It is a measure of how much the variance of a regression coefficient is inflated due to multicollinearity in the data. Multicollinearity occurs when two or more independent variables in a regression model are highly correlated with each other, making it difficult to estimate the effect of each variable on the dependent variable separately. A VIF greater than 10 usually indicates that the variable may be affected by multicollinearity, so our target is to make sure the VIFs for all of the predictor variables in our model are below this threshold by removing the undesired ones. After running a test, we found that both ‘Temperature’, ‘Humidity’, and ‘Dew\_point\_temperature’ have VIFs that are higher than 10. Fortunately, once we excluded ‘Dew Point Temperature’ from our model, all VIFs are below 10.

* **Non-linearity and Heteroscedasticity**

Another problem we had when conducting a linear regression analysis is that bike demand could be related to other variables in more complex ways. By simply assuming a linear relationship, we might have heteroskedasticity presented in our model. When we use our model to predict the values, the difference between the predicted values and the actual values is called residual. Heteroskedasticity refers to a violation of the assumption that the variance of residuals is constant across different levels of predictor values. In other words, the accuracy of our linear regression model is not consistent when using different predictor values. This can be confirmed by the residual plot of our model (Figure 10).

**Figure 10**

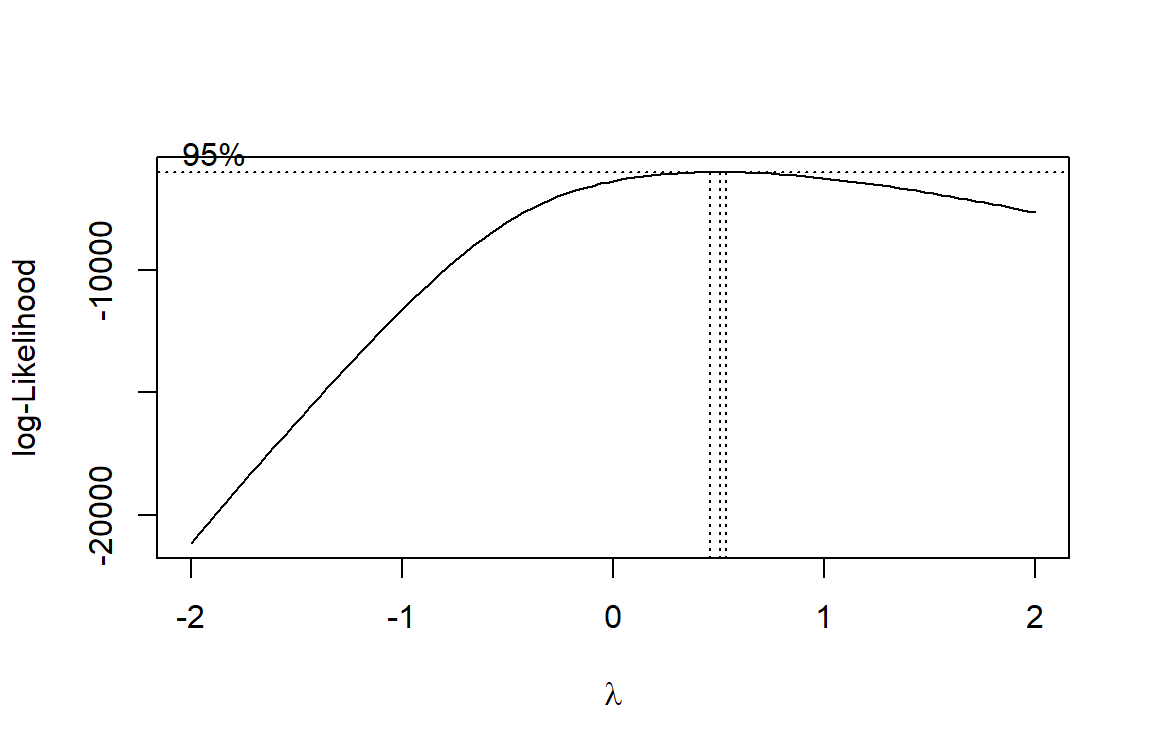
The horizontal axis shows the fitted values of bike usage, and the vertical axis shows the residuals. This plot can help us to understand if the relationship between bike usage and other predictor variables is linear or not. If the residuals are randomly scattered around the horizontal line at zero, it suggests that the model's assumptions are being met, and there is no heteroscedasticity or nonlinearity in the relationship between the dependent and independent variables. However, as our residual plot shows, there is a decreasing trend line at the left, which means the residuals are clearly not randomly scattered around the horizontal line.

1. **Transformed Regression Model**

To address the problem of heteroscedasticity and nonlinearity we met in the linear regression model, we can make some transformations on our variables.

* **Transformation of the response variable**

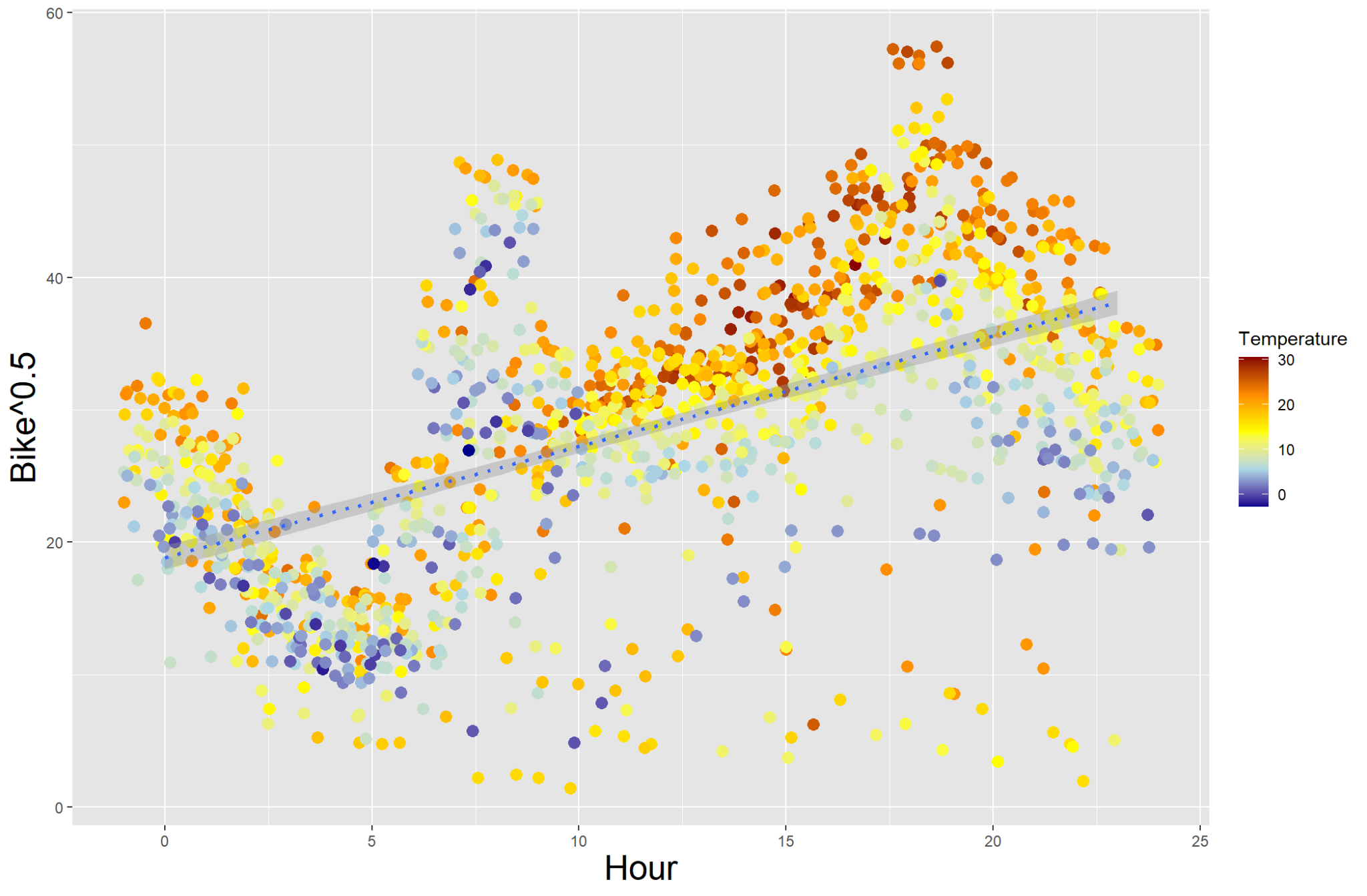
The first approach we tried was to transform our response variable, which is bike usage, by using the Box-Cox function. This function can help us to find the best power transformation of bike usage to reduce both non-linearity and heteroscedasticity. The transformed data can be used to build more accurate models and to make better predictions.

**Figure 11**

Our result shows that 0.5 is the optimal power transformation, which means that we will take the square root of the number of bike usage and treat it as our new response variable.

* **Transformation of the predictor variables**

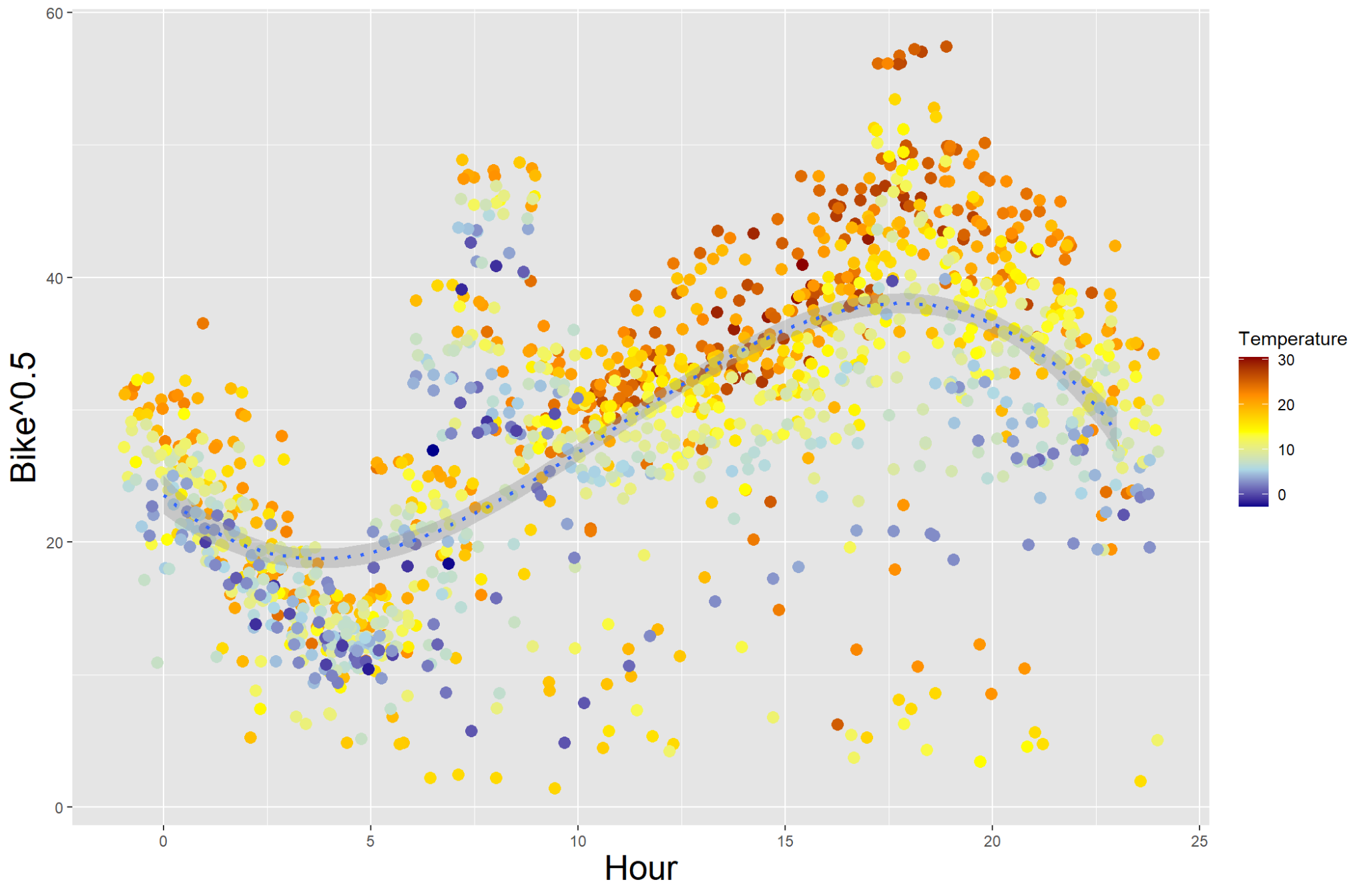
Next, we can do some transformations on our predictor variables. Taking the Hour variable as an example, from Figure 12, we can see that the relationship between the square root of bike usage and hour is clearly not linear, the demand reaches a spike at 8 am and then decreases, then increases again to another spike at around 6 pm. If we fit a straight line, it can not capture this trend very well. So it might be good if we add some polynomial terms of Hour into our model.

**Figure 12**

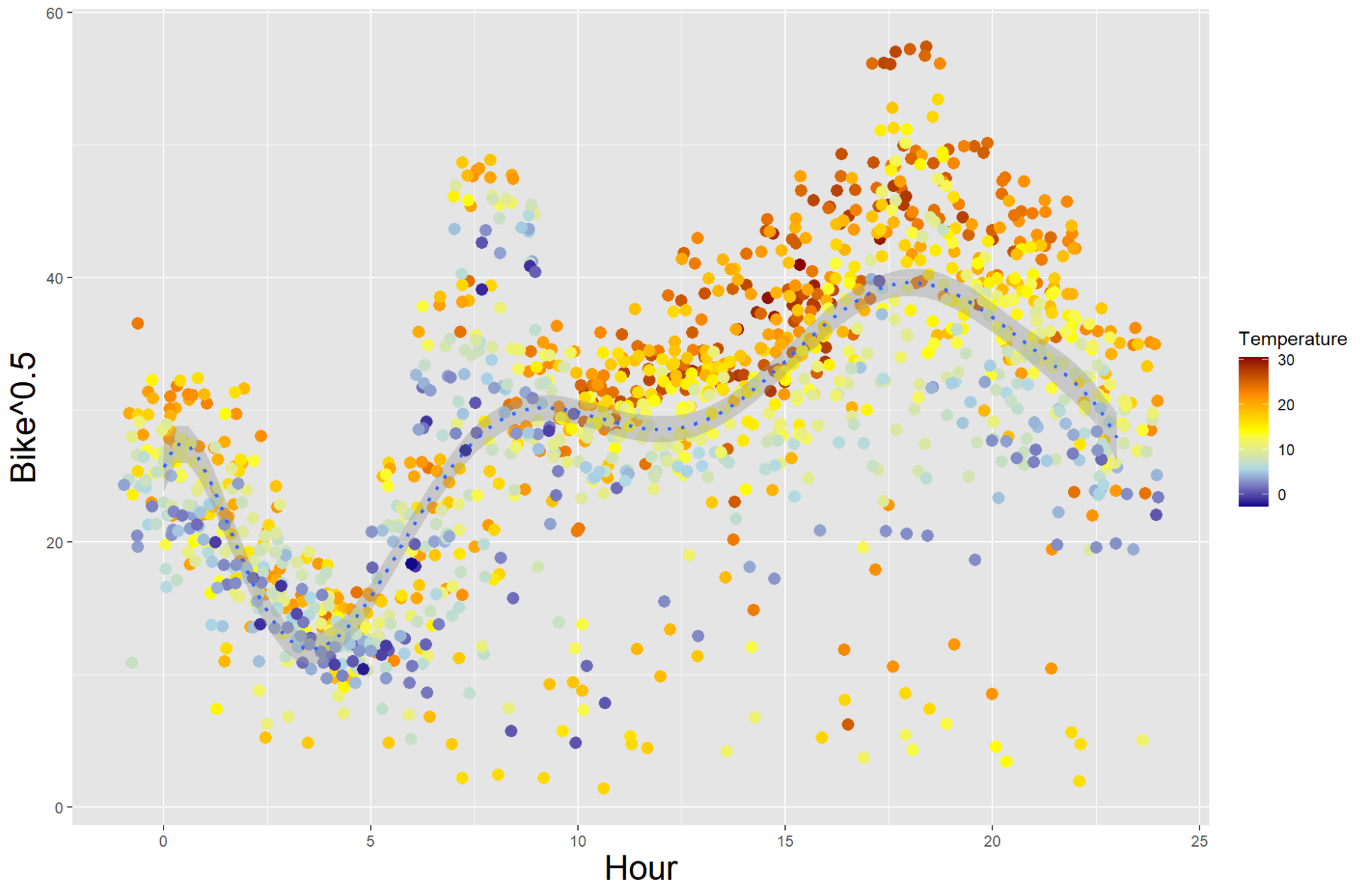
The Square Root of Bike Usage vs. Hour

To illustrate the result of adding polynomial terms of Hour in the model, we fit different lines on the same data to see which line can best capture the trend.

If we add polynomial terms of Hour up to three, we will see a nice curve showing on the graph (Figure 13), which can better capture the demand trend than a straight line. But it still has room to improve.

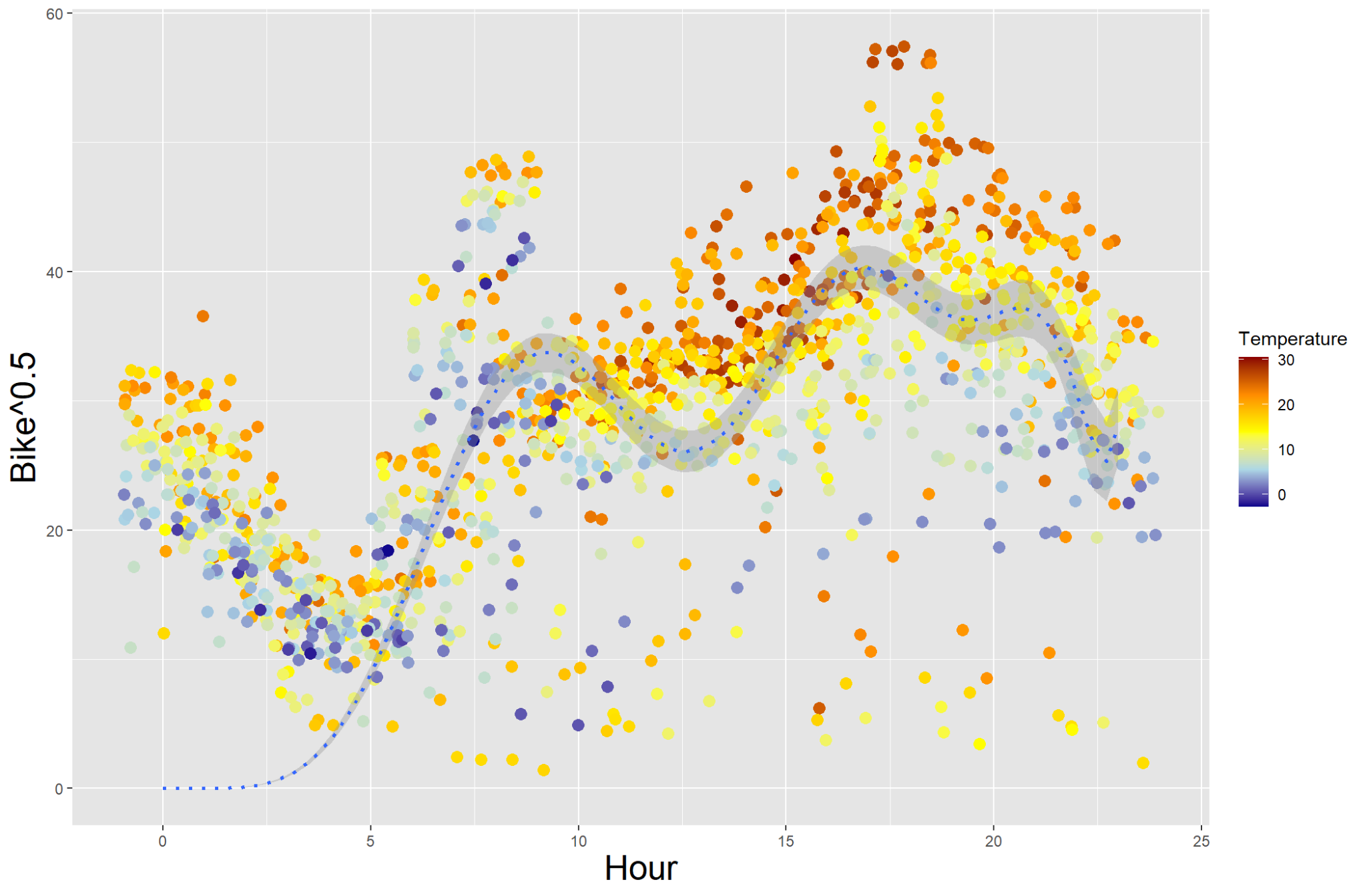
**Figure 13**

If we raise the power of the Hour variable to 8, the two spikes of demand are represented (as shown in Figure 14), which is even better than the curve from before. It seems that adding higher polynomial terms will improve the model fit. However, we need to be aware of overfitting when doing so. This happens when the model is trained too closely on the training data and cannot generalize well to new, unseen data.

**Figure 14**

If we raise the power to 12, the spikes become sharper, which is good. However, we notice that the model can not predict very well when the hour is between 0 to 4, as shown at the bottom left of the graph (Figure 15).

**Figure 15**



Therefore, we finally chose 8 as our final polynomial transformation for ‘Hour’. Also, we chose 4 as the highest order polynomial transformation for ‘Humidity’.

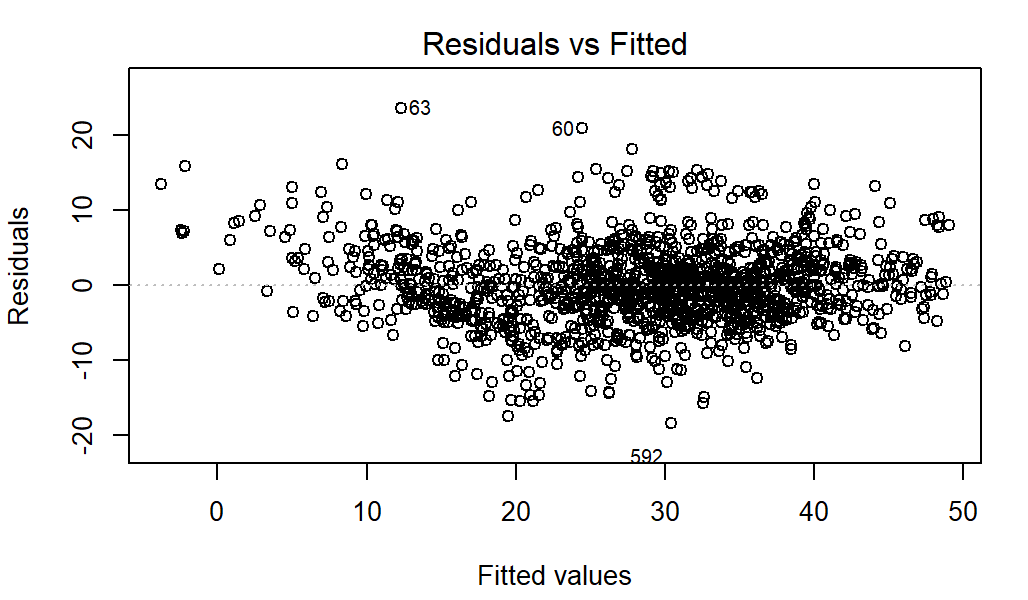
* **Adding interaction term**

We can also include some interaction terms in our model to see if the effect of one predictor variable is constant across different levels of other variables. We considered all possible interactions between predictor variables and then remove those that are not statistically significant.

* **Transformation Results**

After transformation, we get a higher R-squared from 0.52 to 0.76, indicating a better fit. Also, we have a lower MAE from 311 to 203, which suggests the transformed model has a higher accuracy when predicting bike demand. Specifically, it means that we will predict 203 bikes off the true values on average when using this transformed model. Thus, it is still not ideal in terms of accuracy, we might need to try other models like Random Forest to see if it can be improved.

Also, by looking at the residual plot of the transformed model, we can see that heteroscedasticity is reduced significantly. The overall pattern of residuals is flatter. In other words, the residuals scattered around the horizontal line at 0 more randomly, indicating a better linear relationship of the model.

**Figure 16**

* **Interpretation of the model**

From our transformed regression model, we can get some key insights by looking at how the predictor variables affect bike demand. The response variable of the transformed model is the square root of bike usage. The predictor variables that are statistically significant are ‘Hour’, ‘Temperature’, ‘Humidity’, ‘Solar Radiation’, ‘Rainfall’, ‘Snowfall’, ‘Holiday’, ‘Weekday’, ‘Temperature & Humidity’, and ‘Solar Radiation and Temperature’.

Among them, both ‘Hour’ and ‘Humidity’ have polynomial terms, which means it is difficult to interpret how they affect bike demand. Both ‘Solar Radiation’ and ‘Temperature’ have positive coefficients, which means that people tend to ride bikes more often when the weather is warmer and sunnier. Both ‘Rainfall’, ‘Snowfall’, and ‘Holiday’ have negative coefficients, which means that people tend to ride bikes less often when it is rainy, snowy, or during the holiday. Furthermore, take the interaction term ‘Solar Radiation & Temperature’ as an example, it has a negative coefficient, which suggests that the positive effect of solar radiation on bike demand becomes weaker or even negative when the temperature is high. This could be due to factors such as discomfort caused by excessive heat that discourage people from riding bikes.

1. **Random Forest Regression**

* **Model**

Random forest is a popular ensemble learning method used in machine learning for classification, regression, and other tasks. It builds multiple decision trees using a technique called "bagging" or bootstrap aggregating, where each tree is trained on a randomly selected subset of the training data with replacement. The random forest algorithm combines the predictions of these individual trees to generate a final prediction, which is typically more accurate and robust than any single decision tree. Additionally, the random forest model is less prone to overfitting and can handle a large number of input variables with different scales and types of data. Random forest has been widely applied in various domains, including finance, healthcare, and ecology, and it has shown impressive performance in both predictive accuracy and computational efficiency.

* **Training**

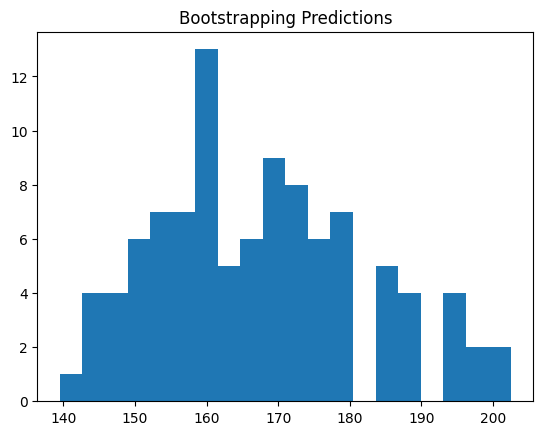
The model is trained and tested on a 75/25 train/test split. The data is inspected for integrity, during which no outliers or missing data is found. However, as in previous models, the variable “dew\_point\_temperature” is removed due to collinearity.

The parameter for this model is the number of decision trees used. Each decision tree in the forest is constructed independently using a different “bootstrap” sample of the training data, and the final predictions are based on the average (for regression) or majority vote (for classification) of the predictions from all trees in the forest. Increasing the number of trees generally improves the performance but more trees also require more computational resources and time. The model is tuned using a grid-search method, a value of 600 decision trees is selected.

* **Confidence Interval**

Bootstrapping is used to construct a prediction interval for the random forest model. This involves randomly sampling the original dataset with replacement to create 100 bootstrapped datasets. Each of these datasets is then used to train a separate random forest model, and these models are used to predict the target variable for a new test observation. The distribution of the predicted values across all of the bootstrapped models can then be used to calculate a prediction interval. Specifically, the lower and upper bounds of the prediction interval can be obtained by calculating the 2.5 and 97.5 of the predicted values, respectively, to create a 95% confidence interval. The interval suggests how much the prediction varies for a particular test observation.

**Figure 17**



Above is an example of a group of 100 predictions for a single testing data point. The bootstrapping prediction interval for random forest does not rely on any specific assumptions regarding the data distribution. However, it is based on the assumption that the training data is a representative sample of the population, and that the random forest model is well-calibrated and accurately captures the relationship between the predictor variables and the response variable. Additionally, the construction of the prediction interval assumes that the errors in the model are homoscedastic (i.e., the variance of the errors is constant across the range of predictor values) and normally distributed. While the assumption of normality is not strictly required for the construction of the interval, it can be used to estimate the standard error of the predictions and to calculate the confidence bounds.

1. **Classification Model**

* **Background**

Upon request, we developed models to predict high demand hours and low demand hours. The high demand threshold is specified at 1100 bikes, while the low demand is 200 bikes. We developed a model for each task. Therefore, the models are binary classification models. The models are evaluated based on accuracy and business effect (ability to predict true positives).

* **Model**

For classification tasks, we used the random forest and the XGBoost model. Random forest classifier differs from the regressor in that during prediction, each tree in the forest produces a class prediction, and the final class prediction is determined by a majority vote among the trees. XGBoost (Extreme Gradient Boosting) is an ensemble learning method that combines multiple weak predictive models to form a strong model. XGBoost uses a combination of decision trees and gradient boosting to improve accuracy, speed, and scalability. It is similar to gradient boosting, but uses a more regularized approach to prevent overfitting and improve performance.

* **Probability**

We attached predicted probability for each prediction, in order to inform the confidence for them. In a random forest classifier, the probabilities for predictions are calculated by aggregating the predicted probabilities from each individual decision tree in the forest. Each tree in the forest assigns a probability to each class, based on the majority class of the training samples in the corresponding terminal node. The predicted probability for a given class is then calculated as the proportion of trees in the forest that predict that class. For example, if a random forest classifier consists of 100 decision trees, and 70 trees predict class A and 30 trees predict class B for a particular input, then the predicted probability for class A would be 0.7 and the predicted probability for class B would be 0.3. Similarly, in an XGBoost classifier, the probabilities for predictions are calculated by averaging the predicted probabilities from all the individual decision trees in the ensemble.

1. **Model Comparison**
2. **Accuracy Comparison**

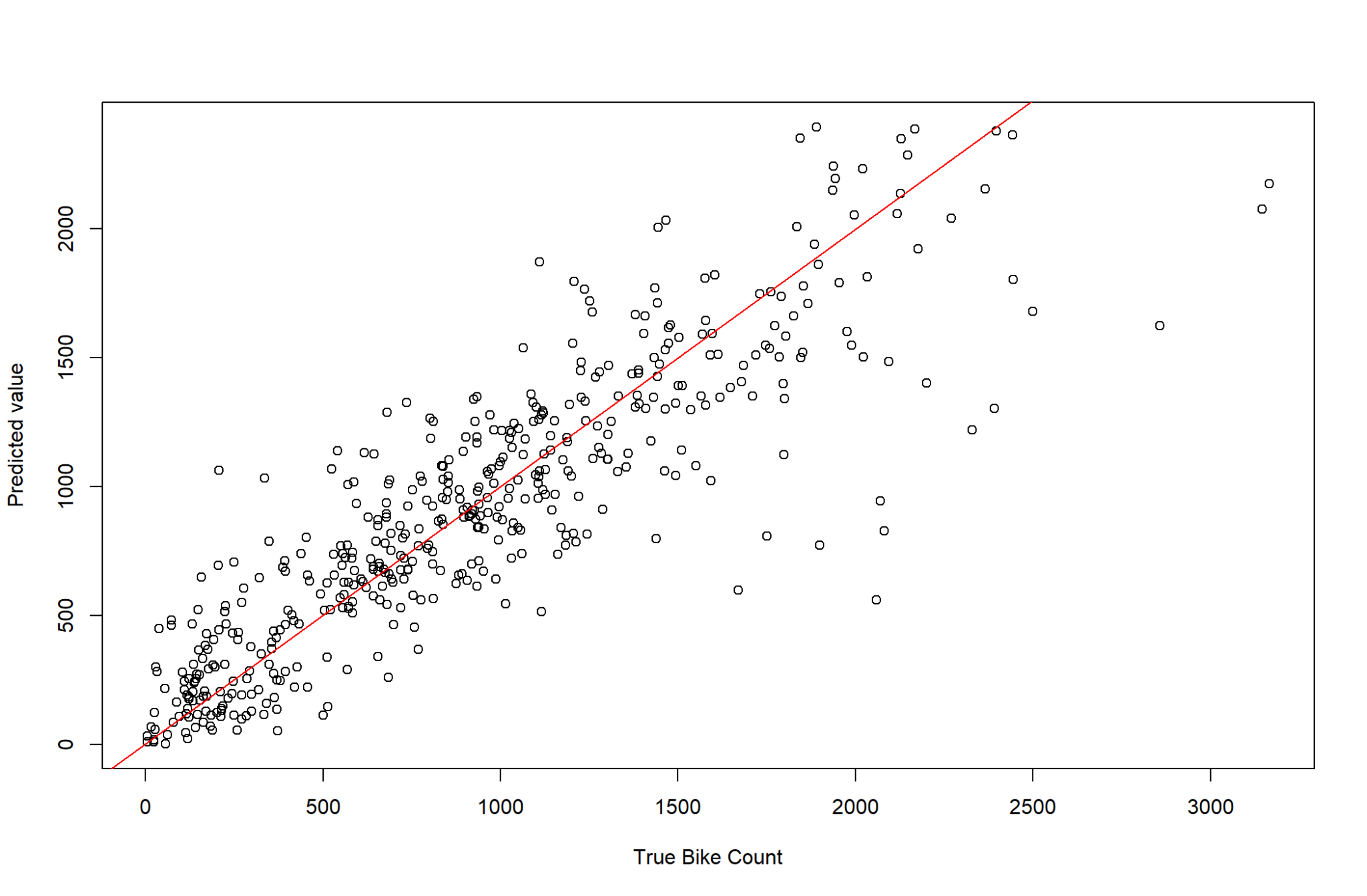
* **Performance**

The performance of our regression models are measured on three metrics: the Pseudo R^2 measures how well the model explains the variation in the training data. The MAE and RMSE both measure the accuracy on the testing data. Mean Absolute Error (MAE) takes the average of all absolute errors, while Root Mean Square Error squares the error, adds them and takes the square root. RMSE penalizes large errors more than the MAE, but MAE also tells on average how the predictions are off.

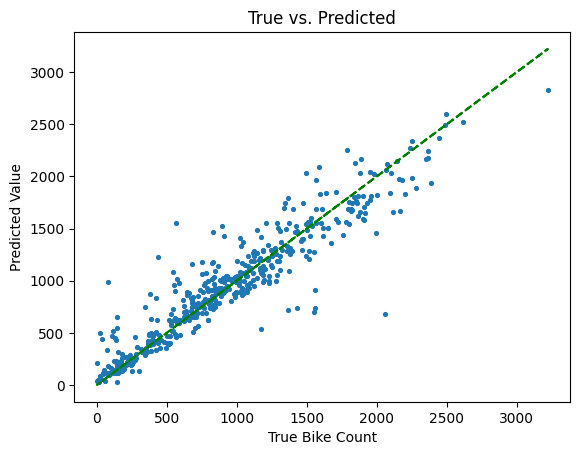
|  |  |  |  |
| --- | --- | --- | --- |
| Model/Metric | Pseudo R^2 (Training) | MAE(Testing) | RMSE (Testing) |
| Linear Regression (Transformed) | 0.76 | 203 | 293 |
| ElasticNet Penalized Regression | 0.53 | NA | 411 |
| Negative Binomial Regression | 0.37 | 220 | 312 |
| Random Forest Regression | 0.98 | 132 | 205 |

**Figure, Linear regression vs. Random forest**

**Figure 18. Predicted Values vs. True Values (Transformed Linear Regression)**



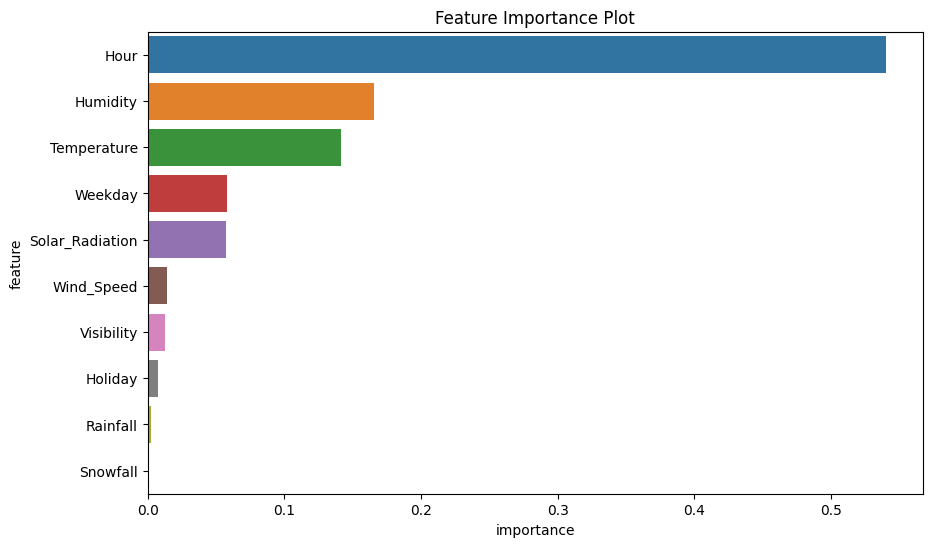
**Figure 19. Predicted Values vs. True Values (Random Forest)**



The random forest model is our best model in terms of accuracy. It reaches an RMSE of around 200 on testing data. It provides a sound prediction across all data ranges. It is significantly better at predicting smaller demand, for example, below 500, and large values over 1500.

* **Interpretation**

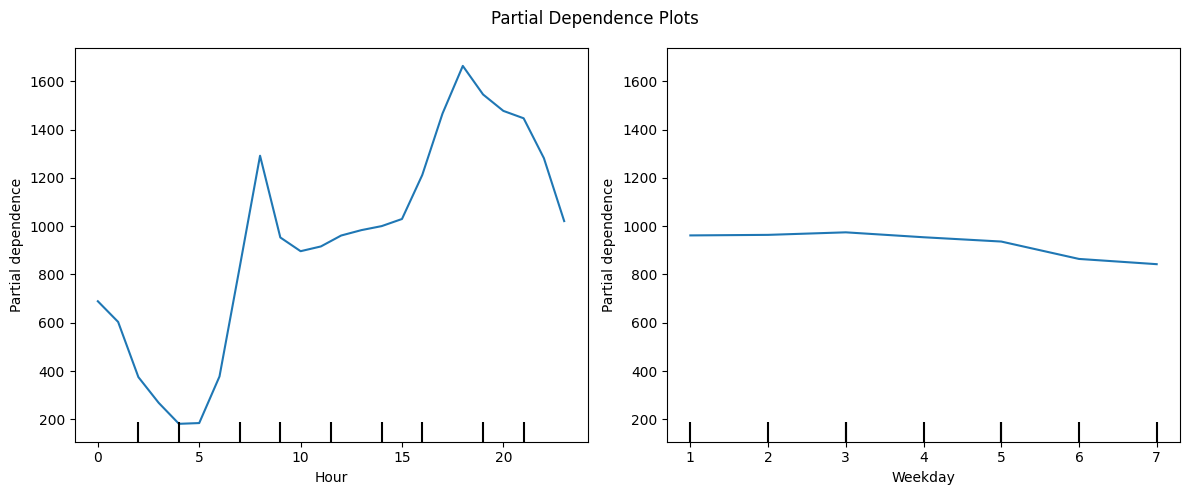
**Figure 20**

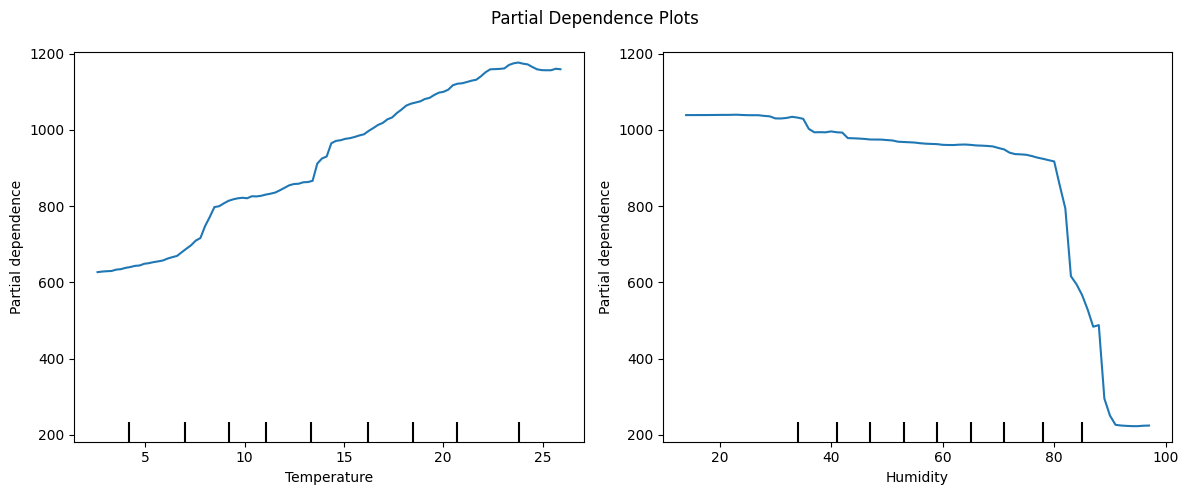


The feature importance plot is used in machine learning to help understand the relative importance of the input predictors. In a random forest model, feature importance is often calculated based on how much the mean decrease impurity (MDI) of the decision trees in the forest is reduced by splitting on a particular feature. The MDI is the sum of the impurity decrease over all nodes in a decision tree when a split is made. The impurity decrease is calculated using Gini impurity. The feature importance score is then normalized to have a sum of 1, with higher values indicating greater importance. Normally, feature importance is not the same as causality but is useful in understanding the mechanics.

The feature importance plot shows that Hour has the biggest contribution of all variables. Temperature and humidity are most important among weather predictors. Weekday (Day of week) as another time factor also has a large impact. We can also see that Rainfall and Snowfall do not have a significant impact.

**Figure 21 & 22**



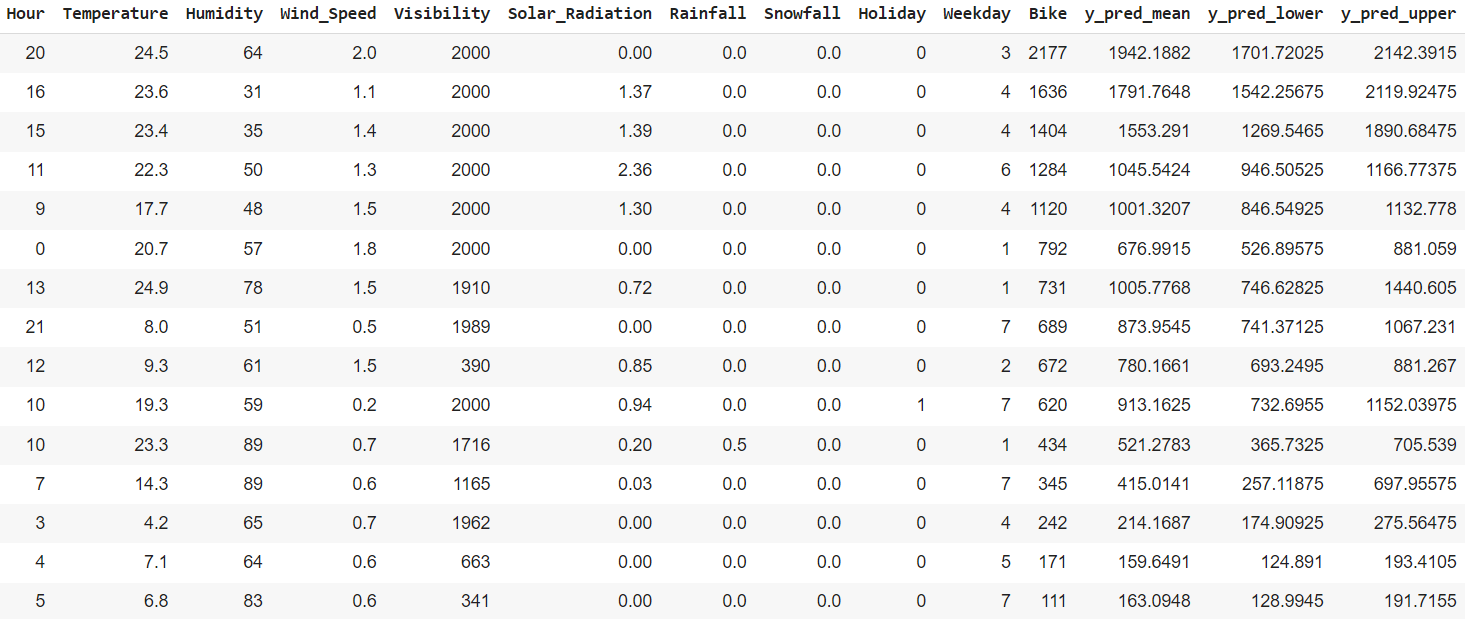


Partial dependence plot (PDP) is a graphical representation of the relationship between a target variable and a set of input features, while controlling for the influence of other features in a machine learning model. It helps to visualize how changes in the values of a specific feature affect the model's prediction, while holding all other features constant at their mean.

In the PDPs, we observe the Hour variable shows a similar association as in the exploratory analysis in Figure 2, where the valley occurs before dawn, and peaks occur during the rush hours. Day of week also matters, with the weekends having a lower demand than weekdays. We also see the temperature has a positive linear association, which is seen in the linear regression model, and humidity shows a negative impact, where the demand drops significantly after humidity hits 80.

* **Conclusion & Demo**

Random Forest regression model outperforms other regression models and is our final choice for bike counts predictions. The model produces a “mean” predicted bike counts, which represents its final decision for the prediction. We also recommend using the model with a prediction interval, which is explained in the “Methods” section. The prediction intervals show the degree of uncertainty of the predictions. In the demo below, the lower bound and upper bound of the prediction interval are located at the last 2 columns of the demo, and the mean prediction is the 3rd column from right. When a large confidence interval occurs (7th observation in the demo), we recommend the client seek further consultation for that particular prediction, such as gathering more information or more data points. When the prediction is small, the prediction usually has high confidence, but does not directly lead to being correct. It generally means that the model explains most patterns happening for that data points, making it confident about the prediction. However, there may be rare cases of highly confident wrong predictions. If such a case occurs it should be recorded carefully.



1. **Results for High/Low Demand Models**

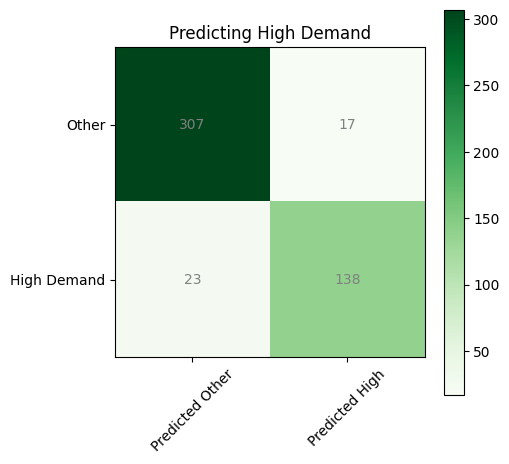
* **Performance**

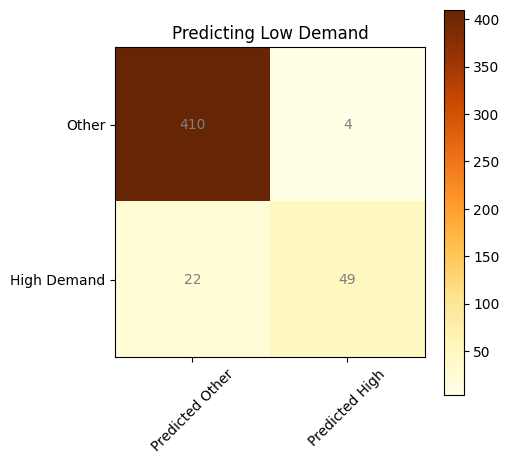
Our high/low demand models are both evaluated through classification metrics. We are particularly interested in these several metrics:

* Overall accuracy: Measures how many of the predictions are true.
* Sensitivity and Specificity: Sensitivity measures how many of the true positive values (i.e. true high demand samples and true low demand samples) are captured by the model. Specificity measures true negative values.
* F1-Score: measures the overall accuracy of a binary classification model (harmonic mean of sensitivity and precision)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model/Metric | Accuracy | Sensitivity | Specificity | **F1-Score(unweighted)** |
| High Demand | 91.7% | 85.7% | 94.7% | 0.873 |
| Low Demand | 93.2% | 69% | 99% | 0.718 |

Accuracy is a more biased measure in that it does not take into account the imbalancement of the data, while sensitivity and specificity only measure positive or negative samples. F1 score measures the precision of predictions, and also considers the ability to capture all samples. In our case, both models have high accuracy, meaning we can trust their predictions with confidence. However, the high demand model has a lower sensitivity of 85.7%, and the low demand model is even worse. This means that although they are both accurate, they are too “conservative” so they miss a portion of positive samples. This is explained by the high specificity for both, where they can capture almost all of the negative samples, introducing some bias to the accuracy. Therefore we prefer F1 score for our model, which measures both the accuracy and counters the “bias” from low sensitivity. Given 0.87 is a high F1 score, the high demand model is a rather reliable model. But the F1 score of 0.718 and a specificity of 69% is not recommended for a classification model, so we only recommend using the high demand model.

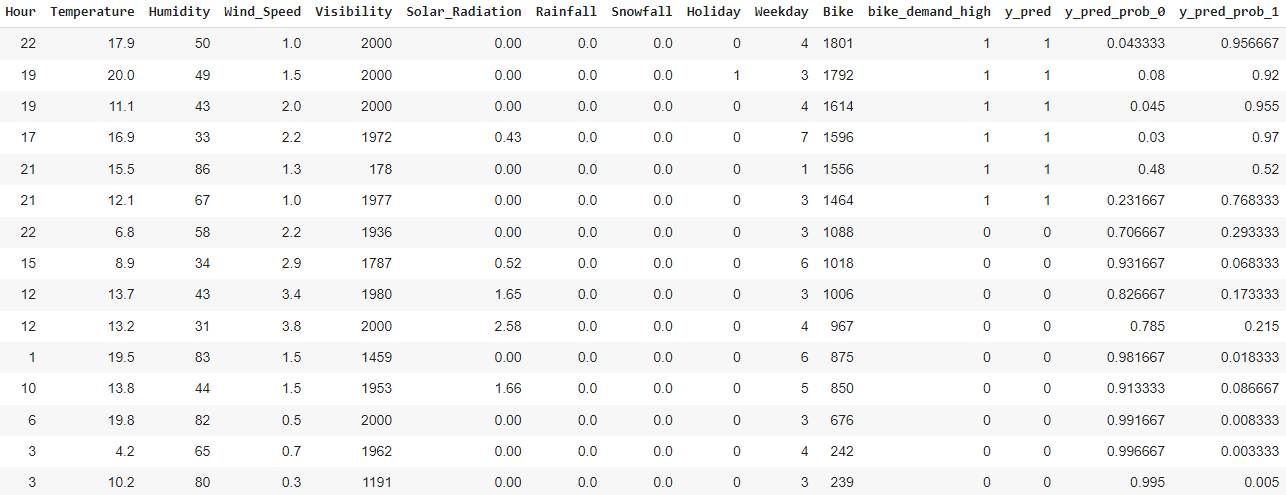
* **Interpretation**

For a small testing sample for the high demand model, we can see that the bottom right corner is the true predictions for high demand samples. We want this area to be as high as possible compared to the 2 shallow areas, which would mean we not only captured most high demand samples, we are also accurate with the predictions. The high demand model has good performance on both aspects.

On the other hand, we can see that the low demand model missed 22 true low demand samples and captured 49. While the predictions are accurate (4 wrong vs. 49 right), 22 missed samples would give a low sensitivity and result in a poor performance.

* **Conclusion & Demo**

Based on our performance inspection of the models, we recommend using the high demand model for accuracy and avoiding the low demand model for most cases since it is too conservative and misses its targets. We suspect this is because the low demand samples only account for about 15% of the total data, with a high degree of randomness, making it difficult to identify patterns. However, the client can still rely on this model due to its precision, where a prediction of low demand hour is almost guaranteed to occur, although at the cost of potentially missing some low demand hours. Furthermore, the client can seek further consultation for low demand predictions, such as adjusting the threshold of the model to make the model more aggressive at capturing low demand samples, at a cost of lower precision.



To illustrate the usage of the high-demand model, a small sample of 15 testing data points is predicted using the model. We can see that the model does its job perfectly. (the third column from the right is the predictions) The last columns are the probabilities of negative predictions and positive predictions. Positive probabilities show the model’s confidence with its predictions. The higher the probability, the more confident the model is about the observation.

1. **Conclusion**
2. **Notable patterns and variables**

Some notable patterns we found at the end were that there was very low bike usage from

4am-6am. We also saw that November had the lowest bike usage. We again believe this was due to the fact that it also had the lowest temperature. Less people will tend to use a bike less in the cold. This also is the same for both rainfall and snowfall. When there’s more rain or snow, people will use the bikes less. However it is important to note that there were not many rainfall or snowfall days in our data. For holidays, we also thought it important to note that although we can see a significant difference in the usage, there were only 2 holidays throughout our data. This means that overall, holidays don’t have a significant impact on our data overall. We thought it also important to note that weekdays follow the same trajectory with peaks at 8 am and 6 pm whereas the weekends follow their own same trajectory.

1. **Model Recommendations**

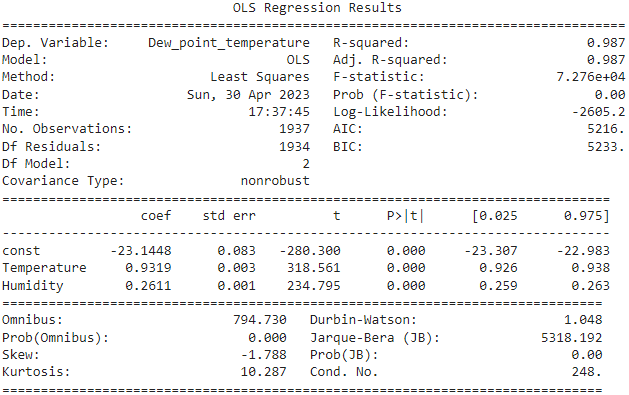
* Among all models, the random forest regressor is the most accurate one so our final recommendation for predicting bike counts is the random forest regressor. It comes with a prediction interval so the client knows if the prediction is reliable. If the client sees a large prediction interval, the true value might not be accurately predicted and we recommend further consultation.
* Although the transformed regression model is not accurate as the random forest model, it can provide some key insight that can help clients understand the relationship between bike demand and predictor variables.
* We recommend the high-demand model to predict high-demand hours. It is most reliable and accurate. We do not recommend the low-demand model as it is too conservative. But the low demand model can still be used as reference as its precision of prediction is very reliable. When the model shows low probability, we recommend more consultation.

1. **Appendices**
2. Dropping “Dew\_Point\_Temperature”

The variable “Dew\_Point\_Temperature” is evaluated based on its theoretical linear relationship between “Temperature” and “Humidity”. According to online references, the dew point temperature is calculated directly from temperature and humidity and can be written as a linear combination of the two. We regress it with respect to the other two variables and had a model with an R^2 of 0.99. It means there exists almost perfect collinearity in the data, which is one of the reasons we eventually abandoned it in our modeling process. The coefficients is very close to the equation we found online (Dew = -23.14 + 0.93\*Temp + 0.26\*Humidity)

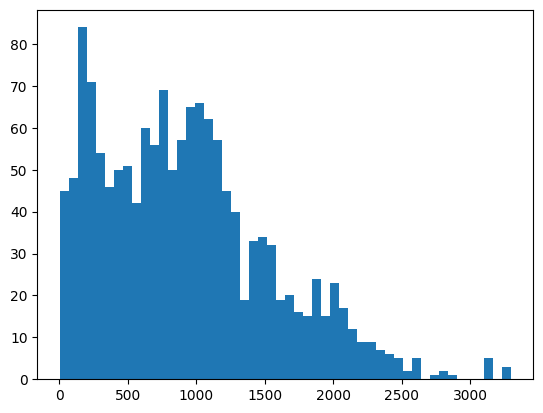
1. Random Forest Regressor Tuning

We used 10-fold cross-validation in a grid search to find the best hyperparameter for the random forest regressor. The hyperparameter is the number of decision trees in our random forest. Part of the grid search results are as follows:

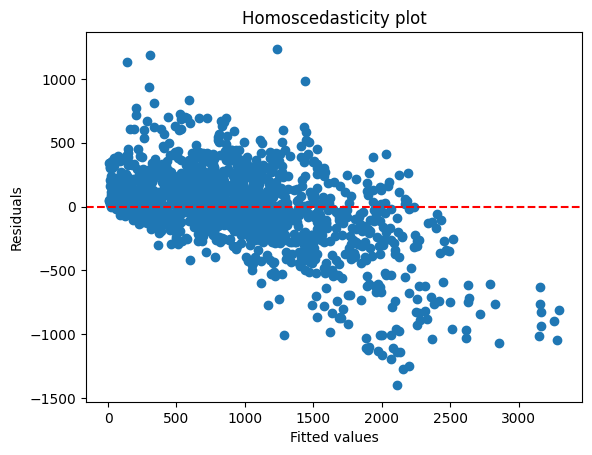
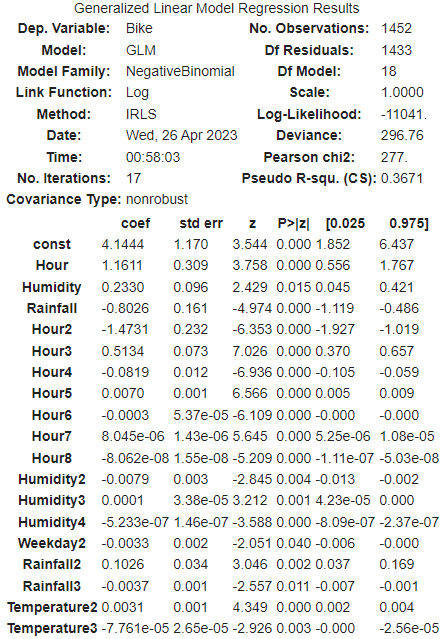


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. Trees | 150 | 250 | 400 | 500 | 600 | 700 |
| CV MSE | 42101 | 41451 | 41697 | 41438 | 41309 | 41421 |

1. Distribution of response and choice of regression



The distribution of bike counts is strictly non-negative and has 2 peaks. It is not a common distribution so we tried several GLM distributions to fit the data. From poisson, negative binomial, and log-normal (log transformation of linear regression), we decided to implement log-normal regression as our final model for interpretation. However, the result for the negative-binomial model (with polynomial predictors) has a healthy dispersion parameter and has similar performance as the log-normal model, as well as a healthy residual plot. The regression summary is as below; the RMSE mentioned before is similar to log-transformed linear regression:



The final GLM model also has a high-degree term for Hour and weekday. However, it recognizes temperature as a non-linear relationship. We also do not observe the y = -x line for the residual plot.

VI. Group Contribution

1. Hritik
   1. Create figures 3 and 5
   2. Helped in writing analysis for figures 3, 4, 5, 6, 8, and 9
   3. Was in charge of talking about specific data visuals within our presentation
2. Audrey
   1. Wrote the overview, tasks, parts of the data portion, and the notable patterns and variable conclusion.
   2. Created figures 1, 2, 4, & 7 along with the analysis of those
   3. Was in charge of giving the agenda, introducing the project, talking about the data, and talking about specific data visuals for the presentation
3. Mincheng
   1. Created data visualizations including Figures 6, 8, 9, 10, 11, 12, 13, 14, 15, 16.
   2. Conducted a linear regression analysis to predict the outcome variable and addressed issues of multicollinearity, heteroskedasticity, and non-linearity
   3. Applied Box-Cox transformation and polynomial transformation to deal with non-normality and non-linearity in the data, respectively.
   4. Selected appropriate interaction terms to incorporate complex relationships between the predictors and the outcome variable.
   5. Wrote analysis of both linear regression and transformed regression model and took charge of the introduction of the regression approach during presentation
4. Kurt
5. Coded GLM, penalized regression, Random Forest, and other models for regression.
6. Coded classification models for high/low demand.
7. Wrote the report and analysis for the above models.
8. Created the other figures.