



Predicting Bike Rental Demand in Washington, D.C.



Group 4-
Sally Gao (sg2zv)
Jo Pan (hp4zw)
Pragati Shah (pvs3vf)
Abhimanyu Roy (ar3dd)

Executive Summary

Project Goals

Our project predicts the demand for bicycle rentals in Washington, DC. We used a dataset containing a 3-month usage log of Capital Bikeshare rentals, in combination with a dataset of weather data over the same period. The aim of our analysis was to answer the following questions:

1. Can data related to weather conditions and time of the hour/week/year be used to predict the demand in the number of bicycles?
2. What is the relationship between the number of bike users in a given day and weather conditions such as temperature, wind speed and humidity?
3. How does the number of bike users at a given time vary by hour?
4. Given the above variables, can we accurately build a linear model predict demand in the number of bicycles?

Analysis Results

Using linear regression, we built a model that predicts the total daily bike count over the 90-day period covered by our dataset. The model uses the following predictors:

<ul style="list-style-type: none">• Average temperature• Average humidity• Amount of rain in inches• Day of the week• Whether or not the day is a public holiday	<ul style="list-style-type: none">• Whether it snowed on that day• Whether it rained on that day• Average windspeed• Month• Day of the month
--	--

Our model is statistically significant according to its F-statistic, indicating that collectively, we are confident that these predictors explain at least some of the variation in the daily total bike count. The adjusted R^2 value of our model is 0.831, indicating that our model explains over 80% of the variability in our data.

We also built an hourly model that predicted the residual bike count from our first model. Here, we achieved an R^2 value of 0.637, explaining a further 63% of the variation in bike count within each day.

Recommendations and Further Areas of Study:

1. We recommend that Capital Bikeshare use our model to predict demand, which can be used as a base reference for resource allocation;
2. We suggest that our predictions can be further refined by incorporating hourly weather data into the model;
3. In order to improve the usefulness of our model, a potential extension is to predict station-by-station bike counts so resources can be allocated on a per-station basis.

Data Description

Our dataset pertains to Capital Bikeshare usage logs in the first quarter of 2017, spanning January, February and March. The raw data has more than 646k entries and the following attributes:

1. Duration – Duration of each individual trip
2. Start Date and Time of the trip – The start date and time of the trip
3. End Date and Time of the trip – The end date and time of the trip
4. Start Station – Starting station address and number
5. End Station – Ending station address and number
6. Bike Number – ID number of the bike used for the trip
7. Member Type - Indicates whether user was a "registered" member (Annual Member, 30-Day Member or Day Key Member) or a "casual" rider (Single Trip, 24-Hour Pass, 3-Day Pass or 5-Day Pass)

Additionally, weather data for the first quarter of 2017 was scraped from Weather Underground to determine the impact of weather conditions on the volume of bicycles rented. The weather data has the following attributes:

1. Date
2. Year
3. Month
4. Day
5. Average, high and low temperatures
6. Average, high and low dew point
7. Average, high and low humidity
8. Average, high and low pressure
9. Visibility
10. Wind
11. Rain
12. Weather Conditions (Normal, Rain, Thunderstorm)

Data Pre-Processing

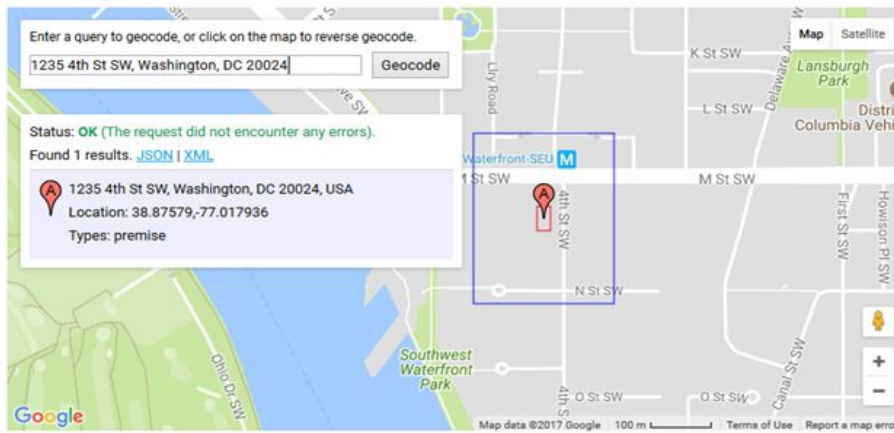
Creating a "Distance" Variable

Our dataset does not contain information on the distance covered during each trip. In an urban setting such as Washington D.C., time may not always be an indicator of distance – for short trips that traverse roads with high density traffic, it may take longer for the rider to reach their destination than for trips with longer distances but with open roads. We used the addresses of the start and end stations provided in the dataset to approximate the distance covered.

Geocoding

The first step in this process was geocoding — converting physical addresses to geographical coordinates. We geocoded our data by taking the following steps:

1. Built the geocode conversion function using RCurl and RJSONIO packages in R to request latitude and longitude data from the Google Maps API. Since there is a request limit, we added in sleep timer to prevent from being blocked by the API.
2. Cleaned up the addresses by removing building names, retaining only the street address. To reduce the amount of requests made to the API, we kept unique addresses only.
3. Converted all the addresses using the geocode conversion function. Then, we manually converted the non-convertible addresses (total of 6).
4. Lastly, we combined the geocoded information with the original dataset.



Haversine formula

After extracting latitude and longitude of the start and end stations, approximated the distance between them using the Haversine formula. The Haversine formula determines two points on a sphere given their latitude and longitude. The distance between two points is given by the formula

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

where:

- d is the distance between the two points on a sphere
- r is the radius of the sphere,
- φ_1, φ_2 : latitude of point 1 and latitude of point 2, in radians
- λ_1, λ_2 : longitude of point 1 and longitude of point 2, in radians

Other Pre-Processing Tasks

- The timestamps were processed to retrieve a separate date and hour column.
- The day of the week was extracted from the date column.

- We also used the date column to determine the prevailing weather conditions and whether the day in question was a weekday or a weekend or a holiday (New Year's Day, Inauguration Day, Washington's Birthday, Martin Luther King Jr. Day).
- The weather conditions were converted from text to dummy variables: Rain, Fog, Snow, and Thunder.
- The Rain_Inches variable that showed the amount of precipitation on the day had trace amounts imputed as 0.
- Only the average wind speed was retained from all the wind-related variables, because we observed there were inconsistencies in the Wind_High and Wind_Low variables — namely, Wind_High was not always the highest observation, and Wind_Low was not always the lowest observation across all the wind-related variables.
- The total bike counts were aggregated by day.

Analysis Results

Model 1: Predicting Daily Bike Count

Model Building Procedure

1. Create a full model with all predictor variables.
2. Investigate multicollinearity between variables, and transform or delete variables as appropriate.
3. Create models using forward selection, stepwise selection and backward elimination approaches.
4. Analyze residuals and influential points.
5. Cross-validation of the final model.

Multicollinearity

Our first model, which was created using the full and untransformed set of variables, was significant according to the F-statistic and achieved an adjusted R^2 value of 0.8092. However, there were several multicollinearity issues that affected the beta estimates of this model:

- An investigation of a correlation matrix of all of the numeric variables revealed that there were a number severe pairwise near-linear dependencies (defined by a correlation coefficient of greater than 0.9 or less than -0.9) between the following variables:
 - Temp_High and Temp_Avg
 - Temp_Low and Temp_Avg
 - Dew_High and Dew_Avg
 - Dew_Low and Dew_Avg
 - Hum_High and Hum_Avg
 - Pres_High and Pres_Avg
 - Pres_Low and Pres_Avg

- The correlation matrix only reveals *pairwise* multicollinearity, so we also looked at the Variance Inflation Factor (VIF) scores of each variable in our model. All of the variables above had severely inflated VIF scores, indicating that the standard errors of their beta coefficients have been inflated due to multicollinearity. In addition, Hum_Low, Vis_Avg, Vis_Low and Weekday all had VIF scores of greater than 5.

	GVIF
Month	3.559411
Day	1.656365
Temp_High	695.228068
Temp_Avg	2052.724330
Temp_Low	426.783671
Dew_High	44.518719
Dew_Avg	98.285957
Dew_Low	41.051390
Hum_High	31.355194
Hum_Avg	84.774390
Hum_Low	19.923179
Pres_High	63.282545
Pres_Avg	219.937390
Pres_Low	75.356917
Vis_Avg	13.778563
Vis_Low	8.826324
Wind_Avg	1.753527
Rain_Inches	3.077308
Rain	3.288142
Fog	2.912139
Snow	2.681124
Thunderstorm	2.504619
Weekday	8.362161
Is.Holiday	1.626052

To deal with multicollinearity issues between variables, we took the following steps:

1. Collapsed “High” and “Low” weather variables into “Range” variables. For instance, we created a Temp_Range variable by subtracting Temp_Low from Temp_High, and created similar variables for dew, humidity, pressure and visibility. In doing this, we hoped to avoid multicollinearity with the “Avg” weather variables, while preserving the information contained in the High and Low variables.
2. After dropping the “High” and “Low” variables for the “Range” variables, an examination of the new correlation matrix showed that there were no more pairwise correlations greater than 0.9.
3. A second model built with the altered predictor set resulted in an adjusted R^2 value of 0.8192. An examination of the VIF scores for the new model showed that while the new VIF scores shown on the right were much lower than the old ones, Temp_Avg, Dew_Avg, Hum_Avg, Vis_Avg all had VIF scores of > 10 and Vis_Range had a VIF score of > 5 .
4. According to the correlation matrix, Hum_Avg and Temp_Avg are both correlated with Dew_Avg by 0.85317738 and 0.80249561 respectively, so we dropped Dew_Avg. We also dropped Vis_Range because it was correlated with Vis_Avg. We built a full model for a third time and inspected the VIF scores. This time, almost all VIF scores were below 5.

	GVIF
Month	3.095478
Day	1.565564
Temp_Avg	3.755938
Hum_Avg	5.706210
Pres_Avg	2.279594
Vis_Avg	6.020056
Wind_Avg	1.718878
Rain_Inches	2.526757
Rain	2.616741
Fog	2.566960
Snow	1.984125
Thunderstorm	1.905676
Weekday	4.619713
Is.Holiday	1.524687
Temp_Range	2.447595
Dew_Range	2.721013
Hum_Range	2.392311
Pres_Range	2.460558

Keeping in mind that Vis_Avg and Hum_Av had VIF scores of 6.02 and 5.71, we felt that we could move onto building a model with a narrowed set of predictors.

Semi-Automated Model Building

We used three semi-automated model building approaches: forward selection, backward elimination and stepwise selection.

All three methods resulted in the same model. The coefficients of the selected variables and statistics related to the model are shown to the right.

This model has an adjusted R^2 value of 0.831 and a multiple R^2 value of 0.862.

Coefficients:











	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	742.37	889.59	0.835	0.406715
Temp_Avg	199.84	17.22	11.607	< 2e-16 ***
Hum_Avg	-20.72	12.87	-1.610	0.111725
Rain_Inches	-4095.34	847.51	-4.832	7.26e-06 ***
WeekdayMonday	193.90	462.12	0.420	0.676021
WeekdaySaturday	-387.06	463.46	-0.835	0.406360
WeekdaySunday	-1709.30	456.99	-3.740	0.000363 ***
WeekdayThursday	-86.17	458.35	-0.188	0.851398
WeekdayTuesday	-40.64	468.28	-0.087	0.931076
WeekdayWednesday	-198.34	475.14	-0.417	0.677591
Is.Holiday	-1733.18	683.12	-2.537	0.013312 *
Wind_Avg	-93.84	32.83	-2.858	0.005545 **
MonthJan	-813.52	345.39	-2.355	0.021194 *
MonthMarch	321.06	316.05	1.016	0.313068
Day	31.76	15.05	2.111	0.038214 *
Snow1	679.38	409.87	1.658	0.101696
Rain1	-570.10	354.92	-1.606	0.112529

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1134 on 73 degrees of freedom
Multiple R-squared: 0.8617, Adjusted R-squared: 0.8314
F-statistic: 28.43 on 16 and 73 DF, p-value: < 2.2e-16

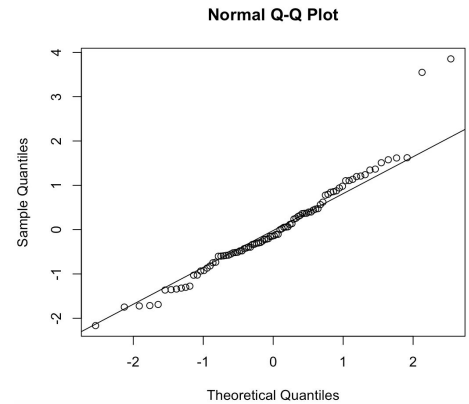
Interpretation of the Model

We interpret the model above as:

Interpretation	Description
 1 F + 200 	For every unit (degree F) increase in temperature, we note that the bike count increases by about 200.
Every inch of  - 4095 	For every unit (inch) increase in rain, we observe that the count of bikes rented reduces by 4095. This is highly indicative of bicycle rider behavior in the rain, and the model reaffirms our intuition. Note: The range of this variable in our data is (0, 0.96).
1 mph  - 93 	For every unit (mph) increase in average wind speed we note that count of bicycles rented reduces by about 94.
  - 1709  - 1733 	Weekends and Holidays have a negative coefficient which indicates that demand for bicycle rentals is higher on weekdays, and more specifically working days.

Residual Analysis and Influential Points

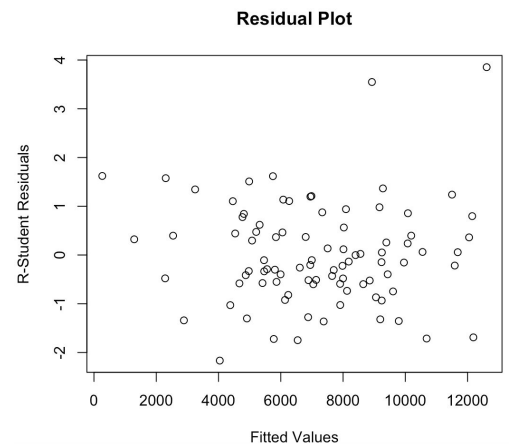
The normal probability plot of our model residuals is shown to the right. It shows that the residuals are more or less normally distributed (if slightly heavily-tailed), meeting the normality assumption of the linear model. There are two anomalous points in the top right corner, suggesting that there are two observations for which the actual bike count is much higher than what we would expect.



A plot of the R-student residuals against the fitted values of the models is shown below. The variance of the residuals seems more or less constant for the vast majority of data points. Again, there are two rather large points near the top right hand corner.

Given the two anomalous-looking observations, we decided to use measures of influence to investigate potential influential points. The following points were identified as influential:

- Obs. 84 has a COVRATIO of 0.07, a DFFIT of 1.94 and a DFBETA of 1.03 for Temp_Avg. This suggests that this observation has unusually high influence over the beta estimate of Temp_Avg, and that the fitted values are being affected by the presence of this data point.
- Obs. 50 has a DFFIT of 1.56 and a COVRATIO of 0.1, suggesting that this point also exerts an unusual amount of influence on the fitted values.



These two observations corresponded to Feb 19, 2017 and Mar 25, 2017. There was nothing out of the ordinary about the predictor values in these two rows. However, March 25, 2017 had the highest bike count of the entire dataset, with a total bike count of 16,191. Feb 19, 2017 also had a relatively high bike count: 12,350, which places it in the topmost quartile.

We tested whether removing these two data points would change the model's beta coefficients. We found a small increase in adjusted R^2 and discovered that a few of the beta estimates were greatly affected.

Coefficients with observations 50 and 84 removed:

#	(Intercept)	Temp_Avg	Hum_Avg	Rain_Inches	WeekdayMonday	WeekdaySaturday	WeekdaySunday
# 1540.024182	174.092476	-13.834248	-4194.415747	146.885718	-801.287525	-2075.045727	
#	WeekdayThursday	WeekdayTuesday	WeekdayWednesday	Is.Holiday	Wind_Avg	MonthJan	MonthMarch
# 12.829004	-7.907761	-92.695427	-1789.780420	-92.307566	-850.845000	324.486204	
#	Day	Snow1	Rain1				
# 30.187008	491.145747	-547.949283					

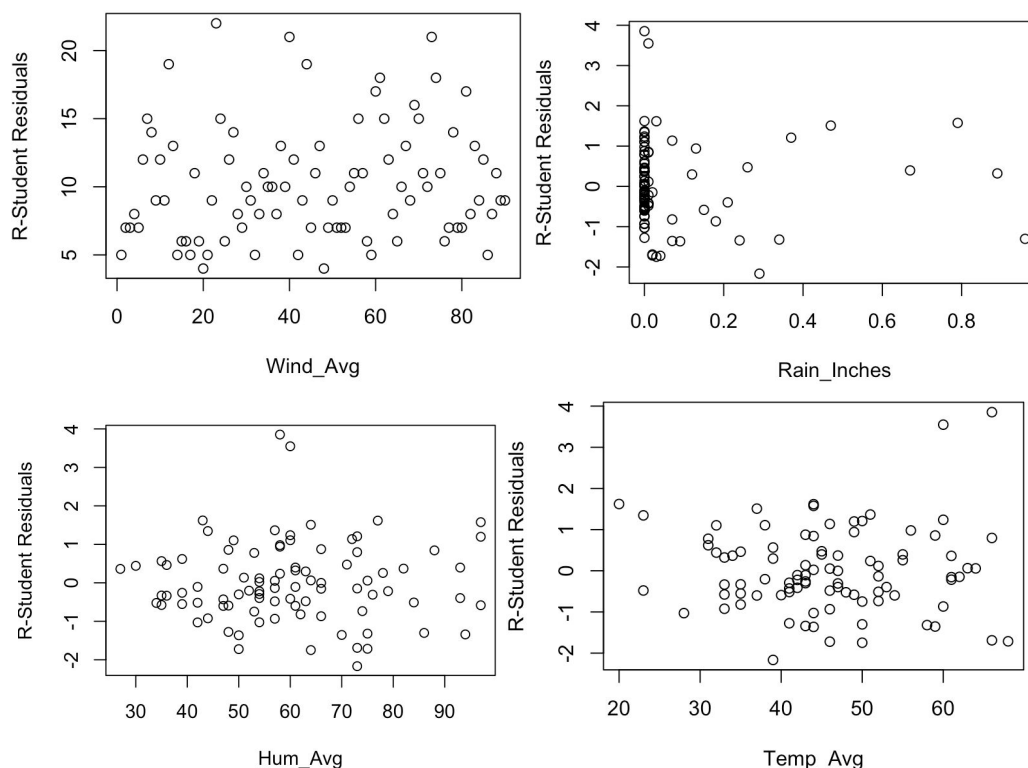
Coefficients with all observations:

#	(Intercept)	Temp_Avg	Hum_Avg	Rain_Inches	WeekdayMonday	WeekdaySaturday	WeekdaySunday
# 742.37329	199.83904	-20.72061	-4095.33739	193.89938	-387.05692	-1709.29852	
#	WeekdayThursday	WeekdayTuesday	WeekdayWednesday	Is.Holiday	Wind_Avg	MonthJan	MonthMarch
# -86.16931	-40.64184	-198.33614	-1733.17718	-93.83794	-813.51719	321.05561	
#	Day	Snow1	Rain1				
# 31.76259	679.38475	-570.10086					

We considered removing the two influential observations from our dataset. However, after extensive research, we found no evidence of any special events taking place in Washington, D.C. on the dates corresponding to the two data points in question.

We cannot justify removing these points in our model as we have no evidence that these two observations are bad data points. It's possible that they *seem* influential in this model because we are working with a small sample ($n = 90$). Therefore, we decided to build the model leaving the two influential observations in.

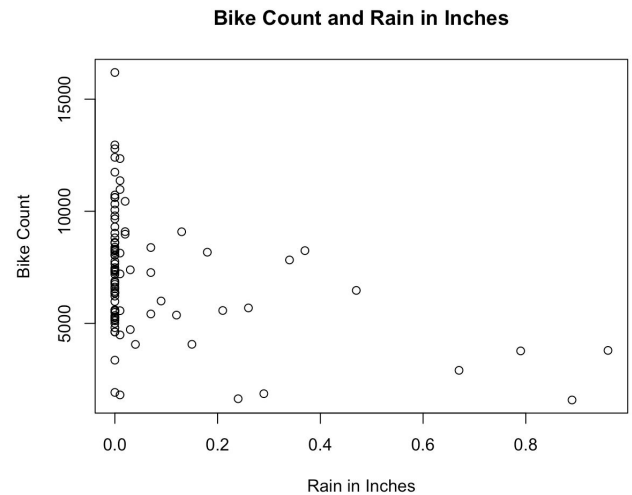
The plots of the R-student residuals against the four continuous variables in the model are shown below:



The residual plots for Hum_Avg, Temp_Avg and Wind_Avg are acceptable. However, the residuals are extremely irregular when plotted against Rain_Inches. Most of the points are compressed against the far left side of the plot, indicating severe nonconstant variance.

Further inspection of the Rain_Inches variable revealed that over half of the observations in that column had a value of 0, indicating either no rain or trace amounts of rain. The relationship between bike count and Rain_Inches is shown in the right scatterplot:

We explored transformations of the Rain_Inches variable, as well as interactions with other predictors. However, we could not find a satisfactory transformation or interaction, as all of them resulted in a loss in our model's predictive power. We decided to keep this variable in the model as it is.

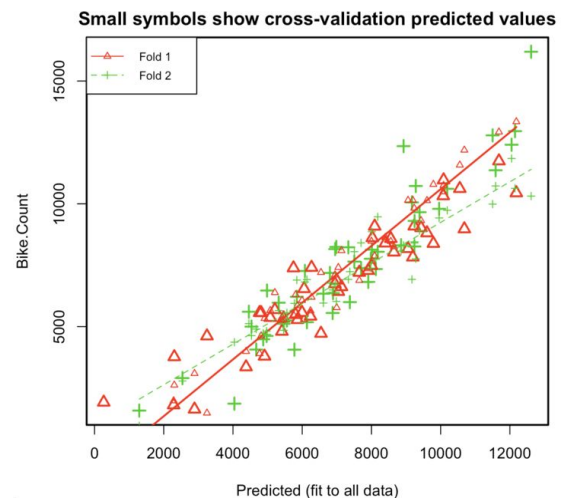


Cross-Validation

We used 2-fold cross validation to test our model performance. To the right is a plot of predicted versus actual bike counts, outputted by the cross-validation function.

The average of the mean squared error (MSE) across the two folds is 2324055. By taking the square root of the MSE, we find that on average, our predictions are off by 1,534 bikes. As a point of comparison, the average daily bike count is 7,183.

Because we had a sample size of $n = 90$, two-fold cross validation reduces our sample size to $n = 45$. With such small sample sizes, we can expect our models to be highly variable using this cross-validation method.

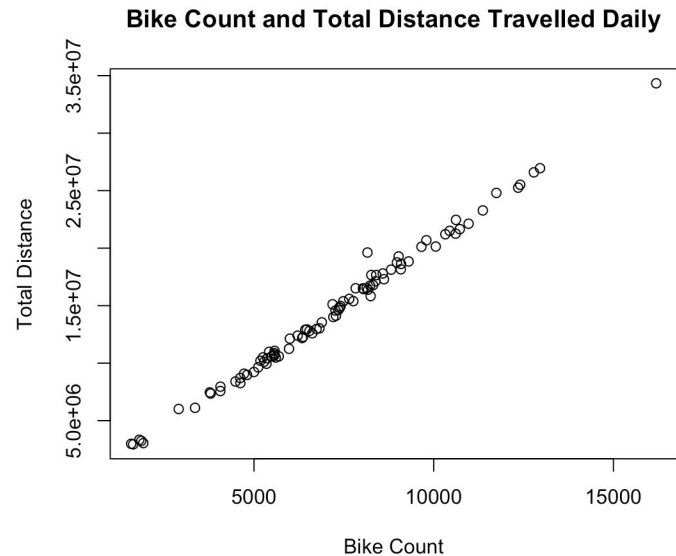


We also looked at the PRESS statistic, another way to evaluate how our model will perform on held-out data. The PRESS statistic is equivalent to the sum of squared residuals (SSRes) from leave-one-out cross-validation. By dividing the PRESS statistic by the degrees of freedom associated with the residuals, we obtained an alternate estimate of the test MSE: 1,977,129. By this estimate, our predictions are off by 1,406 bikes on average.

Model 2: Predicting Total Distance

In our proposal, we wanted to build a model that predicted the total distance travelled by all Capital bikeshare users on a given day. On average, we discovered that users travelled 2000 meters per trip. We discovered that our total distance and bike count variables are highly correlated (0.9963577), as shown in the scatterplot to the right.

Given the high correlation between model predicting Total.Dist would use the exact same predictors as the model above. Using stepwise selection, we found that this was indeed the case.

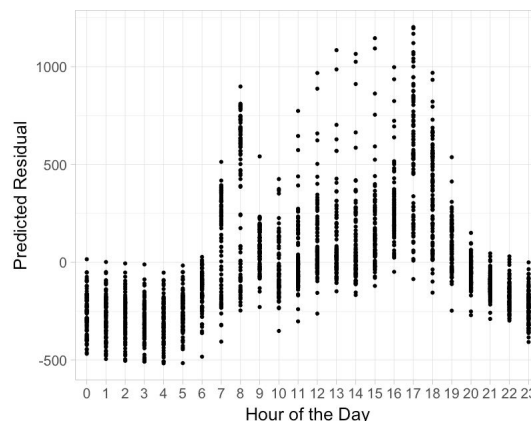


This shows that predicting total distance is essentially the same modeling problem as predicting bike count. As a result, we decided not to create a separate model predicting total distance.

Model 3: Predicting Hourly Bike Count

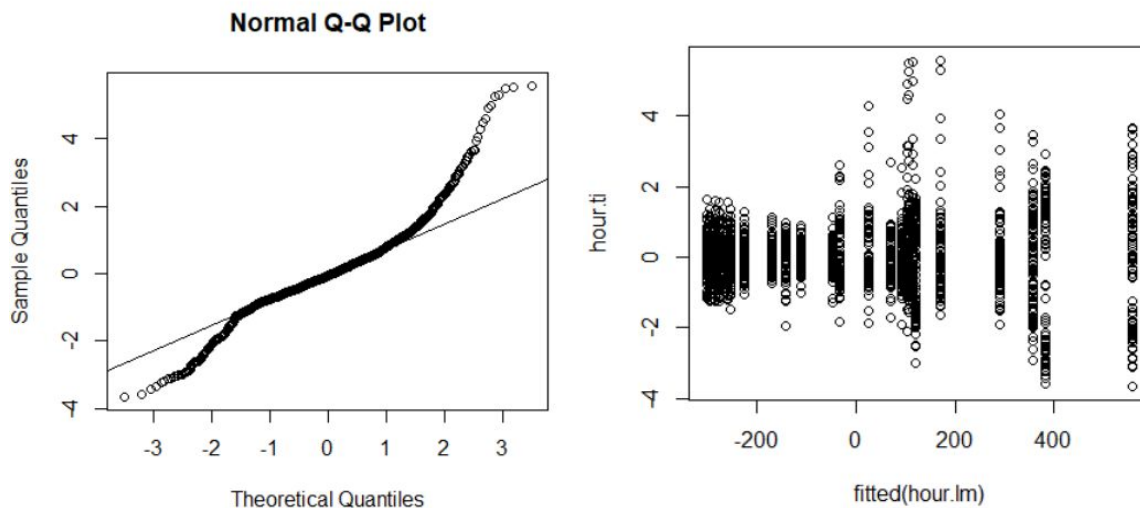
Having created a model that predicts the daily bike count, we wanted to predict the hourly bike count in order to explain the variation in bike demand over the course of a day. To do this, we created a categorical variable, Start Hour, from the starting timestamp of each observation in our unaggregated dataset.

The response variable for this model is the residual of our predicted bike counts divided by 24 — in other words, the daily bike count divided by the number of hours in the day, then subtracted from the actual bike count on that day. Our predictor is the categorical variable, Start Hour.



The summary of the model obtained by following the steps above yields a statistically significant model with an adjusted R-squared value of 0.637. This indicates that 63.7% of our residuals can be explained by taking into account the start hour.

The QQ plot of this model exhibits extremely large and heavy tails. The plot of residuals versus fitted values is also highly abnormal. This is partially due to the fact that our only predictor is a categorical variable.



The plots above show that, despite explaining over 60% of the variance in our residuals, our hourly data does not fully meet the assumptions of a linear model. It is possible that other models are more appropriate in this context. However, our model provides a good starting point for anyone wishing to make a comparison with an alternate model.

Conclusions and Recommendations

Mainly serving customers in the area of Washington DC, Capital Bikeshare is the second largest bicycle sharing system in the United States. Like with most businesses, Capital Bikeshare wants to expand its operations while at the same time keeping its bicycle inventory low. Our analysis can be used by Capital Bikeshare in the following ways to enable it to make data-driven decisions regarding bicycle inventory:

1. In combination with weather forecast information, Capital Bikeshare can use our model to predict daily demand as a reference resource allocation.
2. In order to improve the usefulness of our model, a potential extension of our project would be to predict station-by-station bike counts so resources can be allocated on a per-station basis.
3. Our predictions can potentially be further refined by incorporating hourly weather data into the model.
4. Washington D.C. is a popular travel destination. An exploratory analysis of our data reveals that significant amount of bikeshare users are casual users rather than registered members. Thus, we think that incorporating traveler prediction for D.C. from travel agencies like Airbnb and Tripadvisor could greatly improve our linear model. Additionally, as the membership price for travelers using short term memberships are higher, this would also help bring in more revenue and increase profitability for Capital Bikeshare.
5. By looking at the system map on Capital Bikeshare's website, we discovered that there are empty stations and some stations with almost full rack of bikes. According to our analysis, bike demand is highest during the middle of the day day. Thus, we recommend that any resource reallocation changes should take place at night.