STAT 27850 1 Multiple Testing, Modern Inference, and Replicability Project 1

Gabriel Nicholson, Steven Shi

TOTAL POINTS

94 / 100

QUESTION 1

1 Project 94 / 100

1 Statistical methodology (30 pts)

The aims are:

* The analysis should identify all the major relevant challenges, such as issues of multiple testing, non-independence, confounding, exploratory data analysis, etc

* These challenges should be handled appropriately in the analysis, applying existing tools or developing new techniques

* Assumptions should be tested and examined using, e.g., using diagnostic plots

* Any remaining issues that cannot be addressed, should be discussed

+ **24 pts** Click here to replace this description.
+ **22 pts** Click here to replace this description.
+ **26 pts** Click here to replace this description.
+ **25 pts** Click here to replace this description.
+ **20 pts** Click here to replace this description.
+ **27 pts** Click here to replace this description.
+ **15 pts** Click here to replace this description.
+ **30 pts** Click here to replace this description.
+ **23 pts** Click here to replace this description.
+ **21 pts** Click here to replace this description.
+ **16 pts** Click here to replace this description.
+ **18 pts** Click here to replace this description.

✓ **+ 28 pts** **Click here to replace this description.**

2 Questions, ideas, & design of analysis (40 pts)

The aims are:

* The project should be designed in a creative and thoughtful way, to address interesting questions and challenges in the data

* The real world meaning of the data should be considered in an insightful way to guide the design of the analysis

* Choices made along the way, for example designing a test statistic or finding a way to measure or visualize results, should be addressed thoughtfully

* The analysis should show thorough understanding of any preexisting tools, code, packages, etc, that the group chose to use, and these choices are well suited to the problem at hand

+ **32 pts** Click here to replace this description.
+ **30 pts** Click here to replace this description.
+ **36 pts** Click here to replace this description.
+ **34 pts** Click here to replace this description.
+ **25 pts** Click here to replace this description.
+ **38 pts** Click here to replace this description.

✓ **+ 40 pts** **Click here to replace this description.**

+ **20 pts** Click here to replace this description.
+ **28 pts** Click here to replace this description.
+ **26 pts** Click here to replace this description.
+ **37 pts** Click here to replace this description.

3 Report & code (30 pts)

The aims are:

* The report should be clear and well-written, presenting a cohesive and well motivated explanation of the path followed in the analysis, and thoughtful and justified conclusions based on the findings

* Open questions, uncertainties due to

insufficient data, questions relating to untestable assumptions, etc, should be addressed as needed
* The code should be clear, well organized, and appears readable and reproducible
* Sufficient details should be given to understand the specifics of the analysis being run and the choices made along the way

+ **24 pts** Click here to replace this description.
+ **25 pts** Click here to replace this description.
+ **23 pts** Click here to replace this description.
+ **28 pts** Click here to replace this description.
+ **22 pts** Click here to replace this description.
+ **29 pts** Click here to replace this description.
+ **15 pts** Click here to replace this description.
+ **30 pts** Click here to replace this description.

✓ **+ 26 pts Click here to replace this description.**

+ **23 pts** Click here to replace this description.

Great work on your project! Below are comments for each of the 3 sections of the rubric:

1. Overall, your analysis approaches statistical issues very thoughtfully and does a great job of handling multiple testing, confounding, and other issues. Permuting residuals is a great way to avoid some of the issues that might arise. A few suggestions:
* We should be cautious of overinterpretation that might just arise from multiple testing - for example, your discussion of the meaning of coefficients on the lunch hour indicator is interesting and the trends are certainly worth exploring, but there is an implicit multiple testing issue here and it's not clear if there are any significant trends relating to this indicator that we should be interpreting.
* You mention that checking for nonconstant variance is mostly an issue for confidence intervals and should not affect the permutation test, but this is not necessarily the case. For example suppose that X_i ~ N(0, sigma^2_i) where sigma^2_i increases over time i=1,..,n. Then if we regress X_i onto time i, and run a permutation test to check for a change in the mean over time (i.e., the null is actually true), without permutation the variance is highest for large i and this might create some outliers at high leverage points (i.e., extreme X_i values at extreme i values) ; for the permuted data, on the other hand, the variances are randomly shuffled and this will not occur. This likely isn't a major issue but is worth considering.

2. You make many thoughtful choices in the analysis, such as considering interactions between time and day of week to account for rush hour. You find data to incorporate weather confounder, which is good. You compare the plots of residuals vs time for routes rejected in permutation test to the ones that are not rejected, which is great for a deeper understanding of the findings. Your choices are very well described and clearly considered, e.g. BH vs Storey, residuals vs grouped permutations, etc.

3. Your report is extremely detailed and well written, and includes many thoughtful discussions of different aspects of the analysis. You also discuss possible limitations and problems with assumptions along the way. It would be clearer with more organization, e.g., subsections to indicate the flow of the analysis. The report is also lacking in concrete details and models at times -- it would be better to use some equations etc throughout the report rather than describing procedures only in words, for greater clarity (for example, describing permuting residuals is done only in

words, and would be much clearer with a few equations.)

Gabriel Nicholson
Steven Shi
STAT 27850
2/12/2022

Group Project 1: Bike Routes in Washington

Question

Our question is as follows: are there any routes for which we can detect changes in the average time it took to travel the route because changes to the route made it easier or harder to travel, holding other time-varying conditions constant? For example, we are interested in changes that occur because bike lanes were added to roads, bridges connected areas near one station to another, or existing trails were improved. However, changes in average duration can occur for reasons we are not interested in. It is possible that in 2010, there were many rides on a given route taken in December (near the beginning of our data), when the weather is snowy, making traffic higher and rides slower. Suppose, however, that most of the rides on that route in 2011 occurred in the summer, when the weather does not cause as many disruptions. This would make it look like the average time to travel the route decreased over the period, even if the route did not change at all.

Data

We used data spanning from October 2010 to December 2011. We defined each route by identifying different combinations of stations where rides started and ended. First, we removed outliers from our data. There were many rides that lasted longer than 10 hours; some even lasted over 24 hours. However, other rides between the exact same stations lasted for less than an hour. We believe that these long rides were probably due to tourists or glitches in the system. For example, a tourist could use a bike to explore the city for the day and eventually return it to a different station rather than going from one station directly to another. If tourists are more common at the beginning of our sample than the end, it could appear that the average duration of the route decreased even if nothing about the route changed. However, we do not know exactly who is a tourist and who is not in our data, so we decided that the safest way to deal with outliers like this was to remove them from our data. To define outliers, we had to come up with a method that respected the fact that some routes generally take longer than others without removing outliers in a time-dependent manner. To do this, we computed the mean and standard deviation of the ride duration across the entire period separately for each route, then removed observations that had durations more than two standard deviations away from the mean.

After doing this, we merged weather data for Washington, D.C. for each day in our sample obtained from VisualCrossing weather API. This data included the average daily

temperature and conditions (clear, partially cloudy, rain, etc.). There were a total of 12,036 unique routes; we decided to keep the 500 most frequent routes in the data. These routes contained the most data with a minimum of 522 observations and an average of 939 observations. We made this decision for two main reasons. First, because our methodology uses a permutation test for each route, it is computationally expensive; running this procedure on all 12,036 routes would take several hours. Second, some of the routes have very few observations (1,444 of them had only 1 observation, and 7,606 routes had less than 50 observations). Restricting the data to the most common routes allows us to obtain precise estimates and reduces the influence of outliers and high leverage data points.

Methods

First, we identified several confounding variables that may distort the relationship between the ride duration and the passage of time, quantified by the days since January 1st, 2010. These included whether or not the rider was a member of the program, the day of the week, the time of day, and weather conditions. All of these variables may vary over time, and they have a strong potential to impact the duration of each ride. Members are likely to ride more frequently, and we would expect these people to be faster bikers along with using the bikes mostly for commuting, while non-members likely use the bikes for tourism and sightseeing. Furthermore, the day of the week can affect traffic patterns; for example, we would expect traffic to be heavier on weekdays compared to weekends because of commutes to work. The time of day has a similar effect—we would expect the traffic volume to be higher during morning and evening hours when workers are traveling to and from work; we also expect traffic to be heavier during lunch hours and lighter in the middle of the night. The weather is important because we would expect bikers to be more cautious when riding in rain or snow; this would cause them to slow down. These conditions could also make traffic heavier.

To account for the effects of these confounders, we fit an ordinary least squares regression of duration on these variables and obtained the residuals—one identical regression per route, which allows the effects of these variables to vary by route. This is important because some routes might be away from busy roads, meaning that riders would be less affected by traffic and hence the day of week or time of day. Other routes may be between residential areas and commercial centers, meaning that they would be very affected by traffic patterns and thus, the day of week or time of day. Under the null hypothesis that the duration it takes to travel a given route does not change over time given these confounding factors, these residuals should be independent and identically distributed relative to the number of days since January 1st, 2010. In other words, if there was no change to the route itself, the confounding factors should predict seasonal changes, etc. that could make it appear as if the time it takes to travel the route changed, and the number of days since January 1st, 2010 should not help predict the residuals. If the residuals become larger (smaller) as time passes, we are systematically underpredicting (overpredicting) the duration of the rides later in our data; this could be evidence that the time to

travel the route has increased (decreased) over time. The most intuitive way to test this is to compute the correlation between these residuals and the days since the start of our data. By permuting the residuals and computing the correlation between the permuted residuals and the days since the start of our data many times, we can obtain a null distribution for this correlation and use a two-sided test to obtain a p-value on the correlation we observed.

There are several benefits to this method over others. First, unlike a traditional permutation test, it allows us to account for confounding variables. If we wanted to account for the confounders with a permutation test, we would have to permute the durations across observations that have the same values of the confounding variables. With enough confounders, the number of observations that have the same value for each confounder becomes very small. In the extreme case where there is only one observation per combination of confounding values, there would be only one valid permutation. Our method allows us to permute regardless of the values of confounding variables because we already accounted for them in our linear regression. Furthermore, this method is more reliable than including the days since January 1st, 2010 in the regression and using traditional t-tests on the regression coefficient because it relies on fewer assumptions. The standard error and t-statistic obtained from a regression assume that the days since January 1st, 2010 variable enters the model linearly. For inference to be valid, the errors from the regression are also assumed to be homoscedastic and normally distributed. Here, we do not use either assumption to have valid inference.

However, there are still criteria that must be satisfied for our method to be valid. First, we must have the proper set of confounding factors; also, we must model the way the confounders affect the duration properly, or else the residuals are not entirely free of their effects. There are multiple concerns that guided our decisions. First, we specifically chose to use least squares regression rather than tools such as the lasso because all of our confounding variables have a theoretical basis for influencing the duration; thus, shrinkage and the complications of selective inference are unnecessary and undesirable. Since we are using ordinary least squares, we wanted to keep our regression as low-dimensional as possible to avoid overfitting and obtain accurate and precise estimates of the coefficients on each confounder, allowing us to obtain accurate fitted values, and thus, residuals. As a result, we want to ensure that the confounding factors model the ride duration correctly while avoiding unnecessarily high dimensionality.

For member status, we included this variable linearly with no interactions with the other variables. Since it is an indicator variable, higher-order terms are irrelevant. Furthermore, we do not expect members to be affected by the time of day, weather, or day of week differently than non-members, which is why there are no interaction terms. For the day of the week, we used a single indicator of whether the ride was on a weekend rather than several indicators for each day of the week. This reduces dimensionality in a reasonable way because we expect traffic patterns to be influenced mostly by the number of people commuting to work, which should be relatively constant across weekdays. On weekends, however, fewer people travel for work and more people travel for leisure, which is why traffic patterns should be different.

For the time of day, we identified five categories rather than using indicators for each different hour. The first category was between 6am and 9am, which marks the morning commute following the conventional 9am-5pm work schedule. The next category was between 11am and 2pm, since these are the hours when people may be commuting to and from restaurants for lunch. Since not everyone travels for lunch, this is a separate category from the morning commute. After this, we had a category for the evening commute from 5pm to 7pm; this is separate from the others because people may not leave work as hurriedly as they would get to work. Finally, we had a category for nighttime hours between 9pm and 6am, as we would not expect much traffic during these hours. All other hours fall into our baseline category where we would expect a "typical" level of traffic.

With the time of day, we expected there to be an interaction effect with the weekend indicator. This is because as we explained above, traffic may be heavier from 6am to 9am and 5pm to 7pm for work commutes. However, much fewer people work on the weekend. As a result, the effect of these time indicators should vary depending on whether the ride was on a weekend, which is why we included interactions between the morning and evening indicators and the weekend indicator. We also include an interaction between the lunch indicator and the weekend indicator because people are more likely to eat out and travel for lunch on weekends. However, we do not expect overnight traffic to vary depending on the day of the week, which is why we did not interact the night indicator with the weekend indicator.

For weather, we had data on the average daily temperature and conditions such as "partially cloudy," "overcast," "rain, partially cloudy", etc. To reduce the dimensionality here, we reduced the conditions into three categories: no precipitation, rain, and snow. We could argue that rain while it is partially cloudy may be less intense than rain when it is cloudy, but this is not guaranteed, and the slippery conditions induced by rain or snow of any intensity should be the most important factor slowing bikers down. Also, we elected not to include the temperature in our model. This is justified in our discussion.

After regressing the duration of each ride on these variables, we obtained the residuals and then computed the correlation between these residuals and the number of days since January 1st, 2010. Using a permutation test, we obtained a two sided p-value by computing the proportion of correlations between the permuted residuals and the number of days that had an absolute value larger than the absolute value of the unpermuted correlation, adding 1 to the numerator and denominator to ensure a super-uniform distribution. Theoretically, this results in a valid p-value; however, because we repeat this procedure once for each route, we end up with a multiple testing problem. To address this issue, we use the Benjamini-Hochberg procedure to control the false discovery rate at level $\alpha\frac{m_0}{m}$ where $\frac{m_0}{m}$ represents the proportion of true nulls. Although this is more conservative than the Storey (2002) modification where we estimate the proportion of true nulls, we chose it over the Storey method because we know that the Storey modification can cause FDR control to fail dramatically if there is strong positive dependence between the p-values. This is plausible because different routes may see changes in speed together. For example, if a bike lane is added to a popular road, it is possible that this lane can be

incorporated into many routes, reducing their duration. Phenomena like this would cause the p-values to be positively correlated. As a result, we avoid using the Storey modification, even if the original Benjamini-Hochberg method can be overconservative.

Checking Assumptions

Before discussing our results, we need to check the assumptions that are inherent when using a multiple linear regression method. Some of these assumptions must hold for our results to be valid, while others are not required. The first required assumption is that the variables we chose enter the model linearly. We already accounted for this in a way. For example, we knew that the time of day would not enter linearly, so we split it into various indicator variables. Since all of our confounders are indicators, we do not need to plot them against the duration to check linearity. One concern, however, is that there could be interaction effects between the variables that we did not account for. If rain impacts duration differently on weekends compared to weekdays, we would need to include this interaction. The typical way to check this is to include interaction terms in the regression and check if they are significant. However, since we controlled for so many indicators, we did not have time to check for interaction effects for every combination. Instead, we chose the ones that made the most sense from a theoretical perspective (weekend interacted with the morning, lunch, and evening indicators).

Next, we must check that the confounders are not highly correlated. If the confounders are highly correlated, our estimations and predictions would be imprecise. This would be concerning because bad predictions result in residuals that are inaccurate, which would make our correlation inaccurate. A permutation test would not fix this issue since it would simply permute residuals that were inaccurate to begin with. To check if this is the case, we computed the correlation matrix of the confounders, which is displayed in figure 1 (note that all figures and tables are at the end of our report so we could enlarge them for visibility). This tells us that in the entire sample, most of the confounders have close to zero correlation, and none of them are concerningly large. Of course, this does not guarantee that there will be no correlation among these variables for individual routes. However, we checked the correlation structure for some routes individually, including some routes that we found to have a significant change in duration over time and some routes that did not. In the interest of space, we did not include the correlation matrices for these routes.

Additionally, multiple regression assumes homoscedasticity of the errors. To test this, we plot residuals versus fitted values for regressions run on a random set of routes in figure 2. The three plots in the first row are for routes that experienced significant changes in duration, while the three plots on the second row are for routes that did not have significant changes in duration. First, notice that the fitted values are discrete, meaning that the residuals are clustered on vertical lines. This is because all of our predictors are categorical variables. In terms of homoscedasticity, the variance of the residuals seem to be constant across fitted values except in the third plot in the first row, which has the highest variance for intermediate fitted values. However, we are not

very concerned about heteroscedasticity here. The homoscedasticity assumption ensures that the confidence intervals for regression coefficients and fitted values are accurate. We do not make use of confidence intervals obtained from the regression in our analysis, so we are less concerned about this assumption than the others.

Another regression assumption that we will check for the sake of completeness but is not necessary for our results to be valid is the assumption that the errors are normally distributed. The typical way to check this is to use Q-Q plots of the standardized residuals against quantiles of the normal distribution. We did this for three significant and three insignificant routes in figure 3. As we can see, the residuals do not look normally distributed for any of the routes; instead, they have heavier tails. This makes inference on regression coefficients and fitted values problematic, just like heteroscedasticity. However, we do not need the residuals to be normally distributed to check if they are related to the number of days since January 1st, 2010. By computing correlations and then obtaining the null distribution of correlations using a permutation test, we do not make any assumptions about the distribution of the residuals.

Results

First, we examine the intermediate regression results of duration on the confounders. Tables 1, 2, and 3 display the regression results for the three routes (randomly selected earlier) that showed a significant change in duration over time, while tables 4, 5, and 6 display results for the routes that had no significant change over time. As we can see, the member indicator has a large, negative coefficient for every route, and it is significant at the $\alpha = 0.05$ level for five of the routes. Recall that significance using the t-test may not be valid here due to the violations of normality and homoscedasticity of the residuals we observed above. However, the overall trend fits our anticipation that members would ride faster than non-members. The rain and snow indicators have positive coefficients for some routes but negative coefficients for others. However, in the two cases when the coefficients are significant by the t-test, they are both positive. This trend fits our conjecture that rain and snow slow rides down, making the duration longer. Similarly, the weekend indicator is insignificant for five of the six routes; in the one route for which it is significant, it is negative. This makes sense, as we expect traffic to be lighter on weekends, allowing for faster rides.

For the time of day indicators and their interactions, we see some interesting results. By itself, the lunchtime indicator has a positive coefficient for some routes but a negative coefficient for others, and the coefficient is never significant. However, the coefficient on the interaction between the lunchtime indicator and the weekend indicator is positive for 5 of the routes and significantly positive for 3 of them. This suggests that rides are not necessarily longer during lunch time on weekdays but longer on weekends. There could be two reasons for this. First, people could pack lunches on weekdays but travel for lunch on the weekend, increasing traffic and making rides longer. Another explanation is that 11am to 2pm on weekends is a popular time for sightseeing and riding for leisure, and rides for leisure should be longer.

The coefficient on the morning indicator, however, is negative for 5 of the routes and significantly negative for 4 of them, while the coefficients on the interaction between morning and weekend are small and never significant. The negative sign for the morning indicator is opposite of what we expected, as mornings should have more traffic on weekdays, slowing rides down. However, the negative coefficient could be explained by the fact that riders may need to hurry to work during these hours, motivating them to bike faster. It is not surprising that the coefficient on the morning*weekend interaction term is insignificant, as both of the factors affecting ride duration during these hours described above are due to work, and much fewer people commute to work on weekends.

For the evening indicator, we have 4 negative coefficients and 2 positive coefficients. One of each sign is significant. The positive coefficients are what we expected, since traffic is heavier during these hours. However, the significant negative coefficient for one of the routes is surprising. As with the morning commute, perhaps riders are motivated to bike faster to arrive home on their evening commute on weekdays. As with the morning*weekend interaction, the coefficients on the evening*weekend interaction terms have various signs and are never significant. As before, the traffic at 5pm on the weekend should not be too different from 4pm on the weekend, as not many people need to leave work.

Finally, the coefficient on the night indicator is significantly negative for 2 routes and significantly positive for 1 route. Again, this could be due to differences in what the routes are usually used for. Routes along roads that are empty at night should take less time to travel, while routes along roads that are busier at night (perhaps for overnight deliveries, etc.) could take longer. The varying signs of the coefficients on the night indicator and the other variables mentioned above demonstrate the importance of running a separate regression for each route; each of the confounders may affect travel times on different routes differently.

Assuming that these regressions have properly accounted for the effects of time-varying confounders on the ride duration, the distribution of the residuals from the regressions should not depend on the number of days since January 1st, 2010. The histogram of correlations between the residuals and days since January 1st, 2010 across routes are displayed in figure 4. As we can see, they are centered slightly to the left of zero, indicating that the residuals decreased with time for most routes. However, this does not tell us if these correlations are significant. The histogram of p-values associated with these correlations obtained from the permutation tests are displayed in figure 5. Of these, 176 are below 0.05. After using the Benjamini-Hochberg procedure with $\alpha = 0.05$ to account for multiple testing, we rejected 110 null hypotheses that the correlation between the residuals for the route and the days since January 1st, 2010 were zero. Of these, 9 correlations were positive, meaning that the duration increased over time, while 101 were negative, meaning that the duration decreased over time. The fact that most significant changes were negative makes sense. After all, we would expect urban planners to improve routes over time.

Figure 6 provides several examples of the plots of residuals from the regression against the days since January 10th, 2010. To make the plot less messy, we plotted the average residual

for each day. The three plots in the first row are associated with the randomly chosen routes for which we rejected the null hypothesis, while the three plots in the second row come from the routes for which we could not reject the null hypothesis. We plotted the trend line (computed with all residuals rather than the average) for convenience; it is easy to see that the residuals tend to increase with time for the first plot and decrease with time for the second and third plots in the first row, while there is no discernable pattern in the second row of plots. In the first row, we see that the residuals increase (for the first plot) or decrease (for the second and third plot) slowly over time. As a result, this is probably because the route was worn out or repaved slowly over time. If a bridge had opened or there was some other sudden event, we would expect the residuals to suddenly jump up or down at a breakpoint rather than increasing/decreasing slowly like this.

Figure 7 visualizes how we obtained the unadjusted p-value for each of these routes (each of the six plots correspond to the same route as the plot in that position in figure 5). It shows the histogram of correlations obtained from permuting the data 25,000 times. The actual correlation is denoted by the vertical red line, and the p-value is the proportion of correlations falling to the left or right of the absolute value of this correlation, with 1 added in the numerator and denominator to ensure a super-uniform distribution.

Figure 8 is similar to figure 6 and features the exact same routes, but the duration of each ride is on the y-axis instead of the residuals. This is the relationship between the ride duration and days since January 1st, 2010 we would see without any of our analysis above. As we can see from the trend lines, the durations for the routes in the first row are already trending upward (for the first plot) and downward (for the second and third plot) over time. However, we cannot be sure that this is due to a true change to the route over time or some confounding factor such as the weather. After running our analysis above and finding that the patterns remain in the residuals, we can conclude that this pattern is due to something that is not accounted for by the confounders, such as a renovation to the route.

Discussion: Critiques and Limitations

Of course, there are always potential pitfalls in our analysis. Although we did our best to account for confounding factors, there are different ways to account for these variables in the regression, and there may still be variables that we failed to take into account. For example, we acknowledge that the categories for time of day we used are slightly arbitrary. We grouped 9pm-6am in one category, but it is possible that certain routes lie along roads with heavier traffic during certain parts of the night due to overnight deliveries, etc. This is hard to know without knowing the intricacies of traffic in the Washington area or looking at the data, so we grouped the hours according to our best judgment.

Regarding confounding factors, an important one we did not account for is the number of times a rider has traveled the route. As a certain rider travels a given route more often, they probably become faster over time as they become more accustomed to that route. This could

explain why we found a large negative coefficient on the member indicator for the vast majority of our regressions; members almost certainly ride more, making the membership worth it. However, the member indicator does not capture this phenomenon entirely, as even members could get faster as they accumulate more experience over time. This would make it appear as if the time it took to travel the route decreased even if the route did not experience any changes. However, since we do not have a rider ID variable of any kind, we are not able to reconstruct the number of rides a biker has taken before a given ride. As a result, this confounding factor still has the potential to skew our results and cause us to conclude that more routes experienced decreases in the average travel time than we should have.

Another confounding factor we did not have data for was whether a rider stopped along the route for reasons unrelated to traffic. Suppose, for example, that for some route there was a lemonade stand that was open at the beginning of our data, which closes after a while. If a decent number of riders stop to support the lemonade stand while it is open, this would increase their route duration even if nothing happened to the route. When the lemonade stand closes and riders no longer stop during the route, it may look like there was a decrease in the average duration of the route over time. This is just one example in which the chance that riders stop along the route may change over time, which is why this could be an important confounding factor.

Relatedly, we have already discussed how it would be helpful to know which of the riders are tourists. Even though we removed outliers from our data, some rides are still several times as long as others. Besides stopping along the route, some riders may deliberately take long paths from one station to another for sightseeing purposes. If there are more tourists at the beginning of the data than the end, this may make it appear as if the average duration has decreased. Unfortunately, there is currently no reliable way to identify tourists in our data. Tourists are presumably more likely to be non-members, but this does not capture everything.

Tourists could be more common during certain months than others, so one way to incorporate this into our model is to use indicators for the month of the ride. However, we excluded this from our model for two reasons. First, the most important factor impacting traffic patterns and ride durations that vary with the month should be the weather, as this affects all riders and not just a few. This would introduce collinearity into our regressors. Second, controlling for each month would require 11 indicators, quickly increasing the dimensionality of the model; however, reducing this to, for example, 3 indicators for seasons would fail to capture the complexity of the potential confounding effects while making them more collinear with the weather. Nonetheless, not controlling for months/seasons could introduce confounding effects besides tourist frequency. For example, there may be less traffic along bus routes in the summer months, December/January, or late March due to students being on summer, winter, or spring break, respectively, but there could be more traffic along other routes as students spend more time driving to restaurants and malls for social gatherings. This would slow rides down during these time periods.

Controlling for weather conditions is also why we chose not to include the temperature in our model, even though we had this data. This is because temperature is already highly

correlated with weather conditions, as it is colder when it rains or snows, and temperature itself may have little to do with the duration of bike rides besides this. A biker might ride slightly faster when it is warmer since their muscles may work better. However, this should not make much of a difference when rides are short, and any effect would be far outweighed by the effect of weather conditions on traffic patterns.

Another confounding variable we have data for is the bike numbers. It is possible that certain bikes are faster or slower than others, and they might be used more frequently over time if riders discover that this is the case. This would make it seem like the time it takes to travel routes has decreased even if nothing about the route changed. However, we decided not to include indicators for the bike numbers in our model. First of all, there are 1,313 different bikes in our data, so adding indicators for each one would increase the dimensionality of our model by a lot even if not every bike is used on every route. Furthermore, the vast majority of the bikes are probably so similar that there would be no gain or loss from using a given bike over the others. As a result, many of these indicators may not even improve our model. Additionally, 5,292 rides have missing bike numbers. It does not make sense to place all of these observations in the same category as they have the potential to represent many different bikes; this just introduces more unnecessary noise into the model.

Even after assuming that we managed to capture the relevant confounding effects with our regression model, the correlation between the residuals and the days since January 10th, 2010 is not a measure that can capture everything we are interested in. There are many ways the duration of a route can change over time, and a correlation does not distinguish between them. For example, if a shortcut is added to a route, we would expect to see a sudden decrease in the residuals plotted against the days since January 1st, 2010, which would occur when the shortcut is added. However, the duration of a route could also increase or decrease gradually over time. For example, perhaps a route does not become longer or shorter, but the bike lane becomes re-paved/improved one section at a time. In this case, the residuals would also decrease over time—but they would do so gradually. In both cases, the correlation we are examining would be negative; however, we cannot tell which of the cases we are in from the correlation without looking at the residuals.

Furthermore, it is possible for the correlation to miss temporary changes in the route duration. If a shortcut is added to a route and then removed, the residuals should decrease and then increase over time. If the timing of this occurs in the middle of our sample, we could obtain a correlation that is very close to zero. Permuting the residuals would not result in a distribution of correlations that is very spiked at zero, so we would obtain a large p-value from our permutation test. The worst case scenario in which we miss such an occurrence entirely would only occur if the temporary changes in durations were symmetric around the middle of the period for which we have data. However, the correlation still does not tell the entire story even if the change occurs away from the center of our data. For example, suppose a shortcut is added to the route in the latter half of our sample, but it is removed before the end of our sample. The residuals would be low during this period, and they would be low when the number of days since

January 1st, 2010 is large. This would lead to a negative correlation. However, since the route was removed before the end of the data, a person traveling the route after our data ends (assuming no further modifications to the route) should not expect to travel any faster than if they had traveled the route at the beginning of our sample. In other words, even if our methodology leads to a significant, negative correlation, this does not guarantee that the rides at the very end of our sample were shorter than the rides at the beginning of the sample. It cannot tell us whether the change in durations were permanent once made, or if they were due to something temporary.

Another critique of our method that does not involve the regression model is the way we corrected for multiple testing. We know that the Benjamini-Hochberg procedure has the potential to fail at bounding the FDR when the p-values are not independent, and we explained earlier that we do not expect the p-values to be independent. However, we believe that the Benjamini-Hochberg procedure should still perform reasonably well here. This is because it still bounds the FDR at level $\alpha\frac{m_0}{m}$ when the p-values' dependence is PRDS (positive regression dependency on a subset). We explained earlier how we expect the p-values to be positively correlated, which is similar to PRDS. The dependency structures under which Benjamini-Hochberg fails dramatically are rare, so it is unlikely (although still possible) that we have such a problem here. Since it is more powerful than the very conservative Bonferroni and Holm-Bonferroni procedures, we elected to use it despite this potential drawback.

Figures

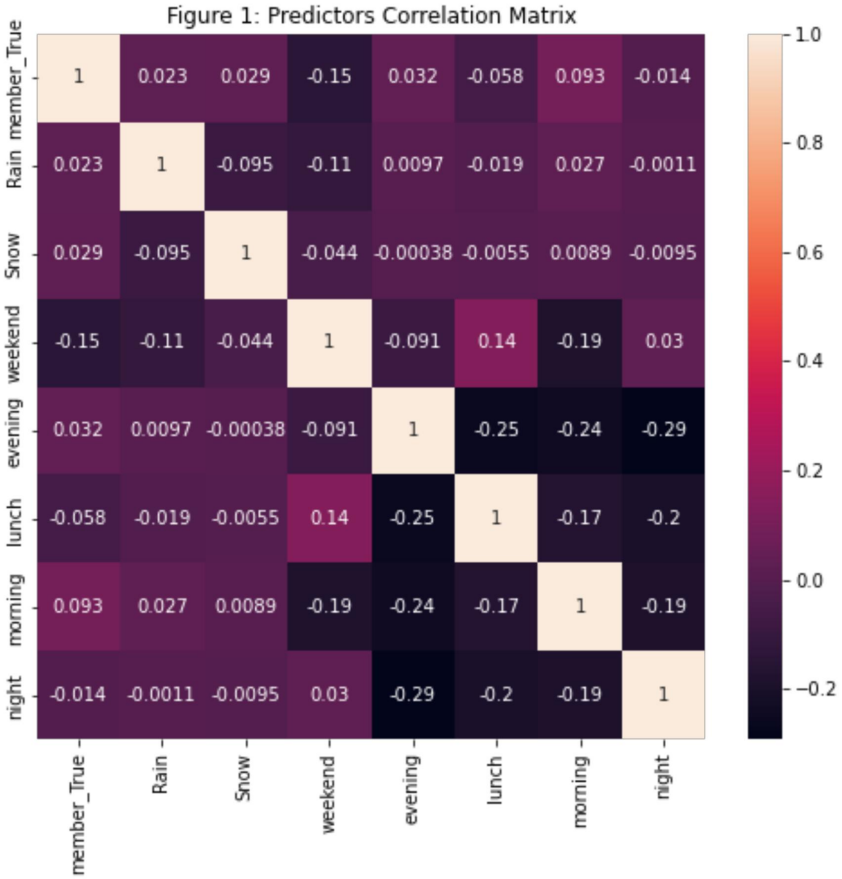


Figure 1: Predictors Correlation Matrix
member_True
Rain
Snow
weekend
evening
lunch
morning
night
1 0.023 0.029 -0.15 0.032 -0.058 0.093 -0.014
0.023 1 -0.095 -0.11 0.0097 -0.019 0.027 -0.0011
0.029 -0.095 1 -0.044 -0.00038 -0.0055 0.0089 -0.0095
-0.15 -0.11 -0.044 1 -0.091 0.14 -0.19 0.03
0.032 0.0097 -0.00038 -0.091 1 -0.25 -0.24 -0.29
-0.058 -0.019 -0.0055 0.14 -0.25 1 -0.17 -0.2
0.093 0.027 0.0089 -0.19 -0.24 -0.17 1 -0.19
-0.014 -0.0011 -0.0095 0.03 -0.29 -0.2 -0.19 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Figure 2: Fitted Values vs. Residuals for 3 Significant and 3 Null Routes

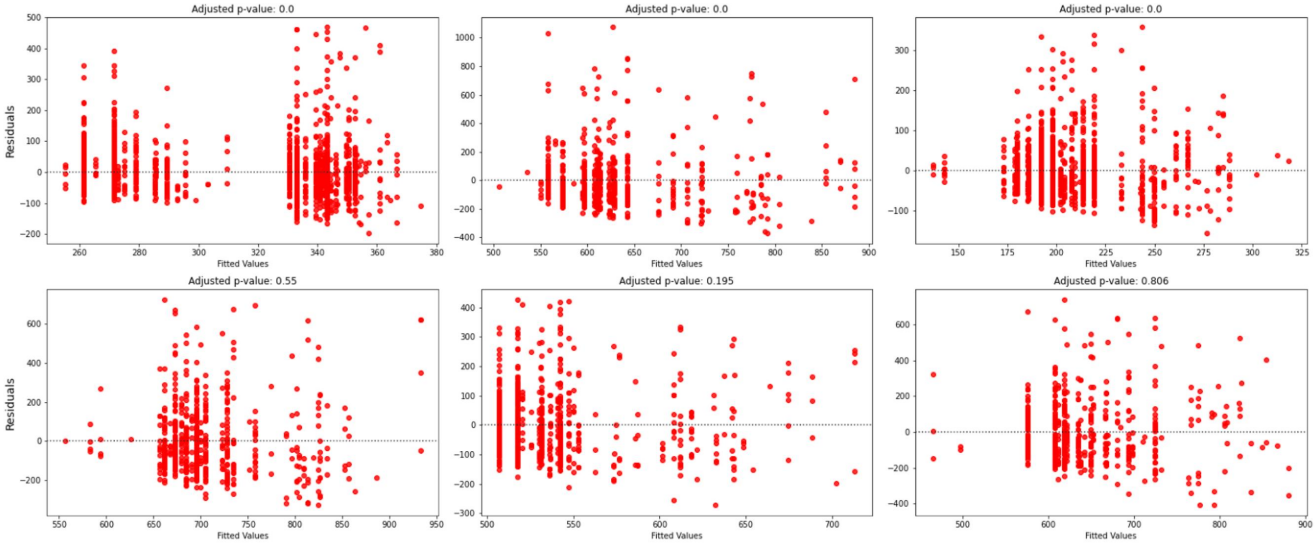


Figure 3: Quantile-Quantile Plot for 3 Significant and 3 Null Routes

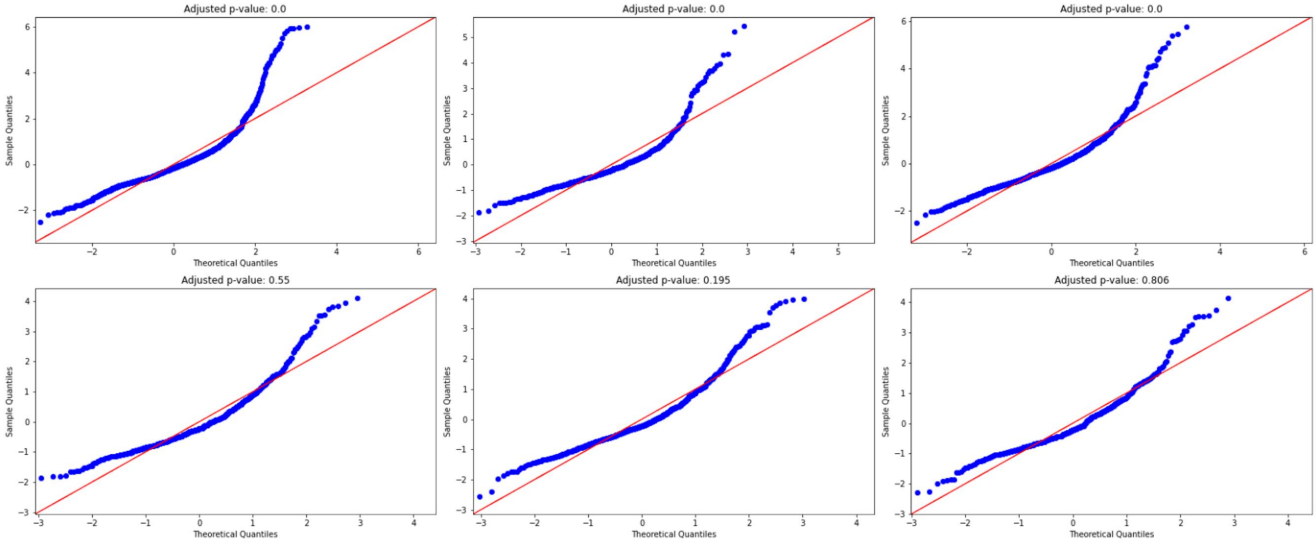
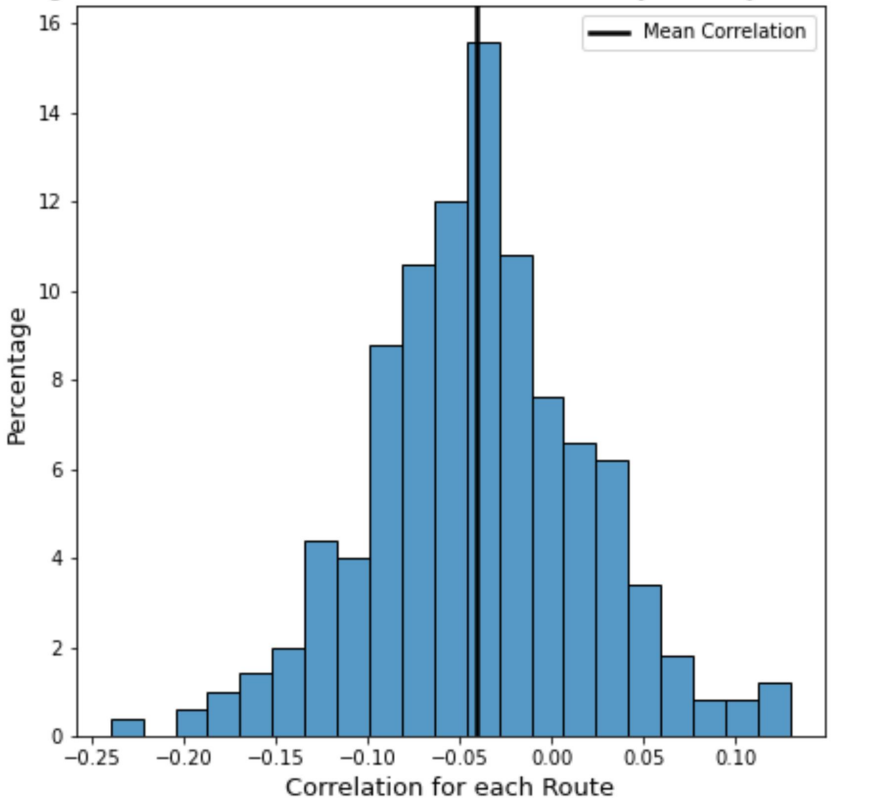


Figure 4: Correlations Between Residuals and Days Since Jan1 2010



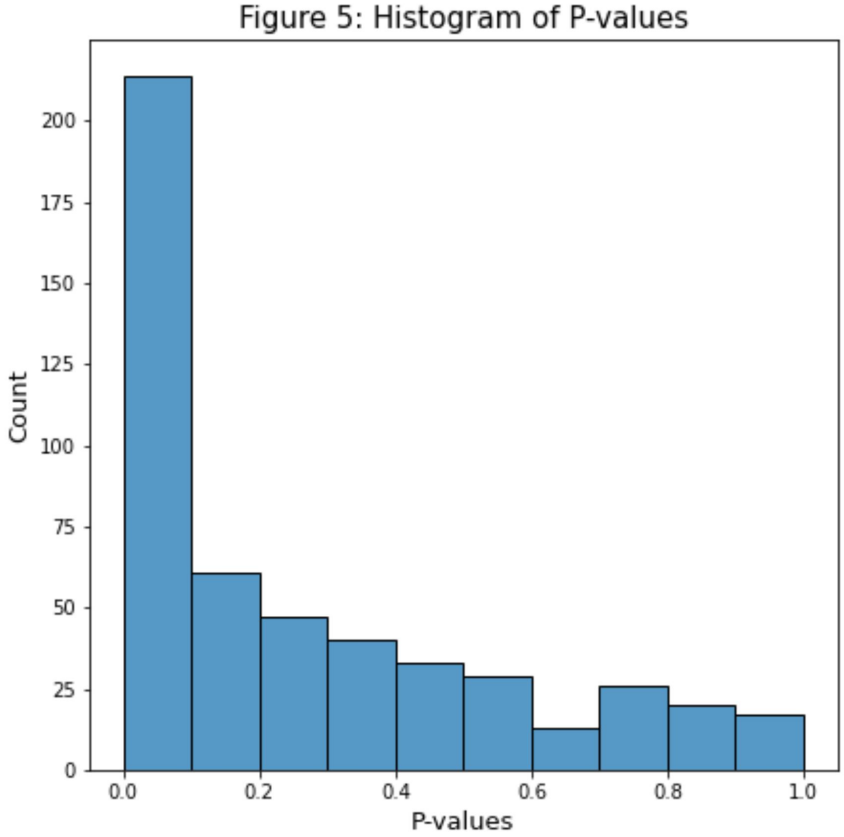


Figure 5: Histogram of P-values
Count
P-values
0
25
50
75
100
125
150
175
200
0.0
0.2
0.4
0.6
0.8
1.0

Figure 6: Residual Plots from 3 Significant and 3 Null Routes

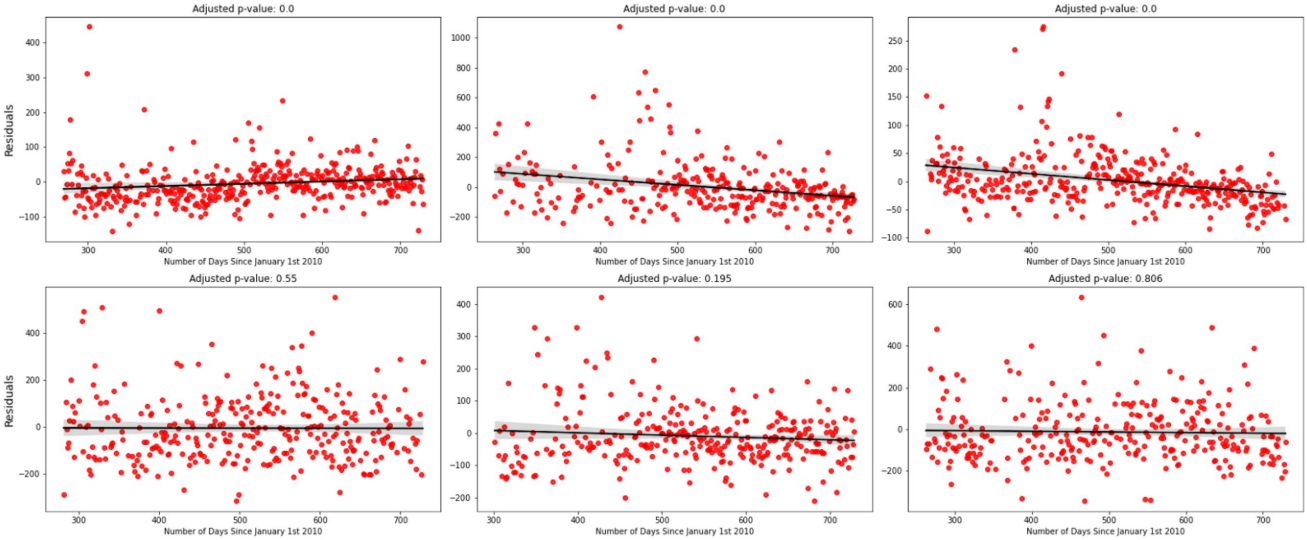


Figure 7: Null Distribution of Correlations

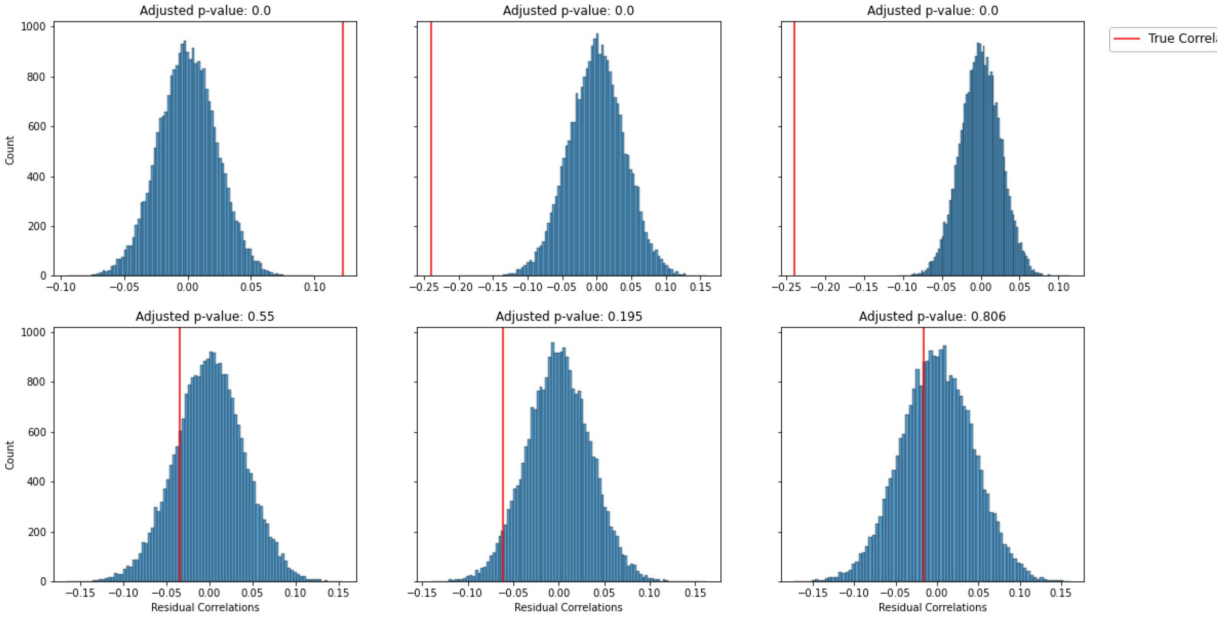
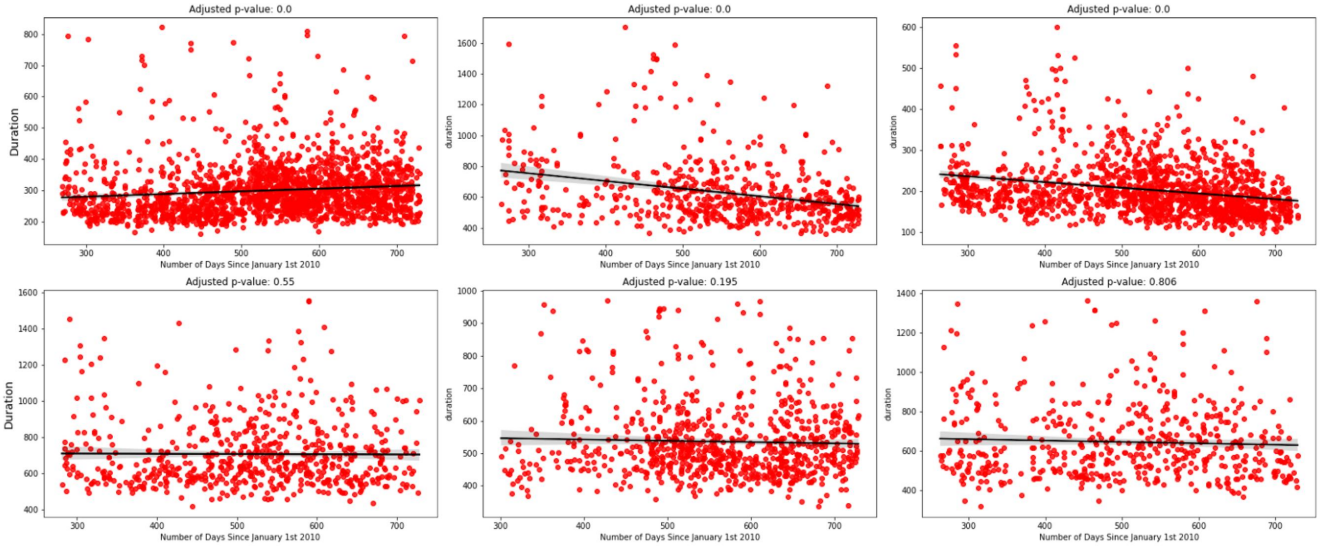


Figure 8: Number of Days Since January 1st Against Ride Duration for 3 Significant and 3 Null Routes



Tables

Table 1: OLS Regression Results for Significant Route

Dep. Variable:	duration	R-squared:	0.183
Model:	OLS	Adj. R-squared:	0.178
Method:	Least Squares	F-statistic:	37.73
Date:	Wed, 16 Feb 2022	Prob (F-statistic):	1.04e-73
Time:	17:33:58	Log-Likelihood:	-10743.
No. Observations:	1861	AIC:	2.151e+04
Df Residuals:	1849	BIC:	2.158e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	346.7605	9.822	35.303	0.000	327.496	366.025
member_True	-13.8121	8.841	-1.562	0.118	-31.152	3.527
Rain	10.3015	3.697	2.786	0.005	3.051	17.552
Snow	11.6232	14.275	0.814	0.416	-16.374	39.621
weekend	6.3960	8.034	0.796	0.426	-9.362	22.154
lunch	9.4628	8.019	1.180	0.238	-6.264	25.189
morning	-71.5992	5.141	-13.926	0.000	-81.683	-61.516
evening	-2.4287	7.895	-0.308	0.758	-17.914	13.056
night	-54.0233	8.226	-6.568	0.000	-70.156	-37.890
morning:weekend	-12.7768	23.242	-0.550	0.583	-58.361	32.807
lunch:weekend	1.7389	14.506	0.120	0.905	-26.710	30.188
evening:weekend	-0.9797	14.895	-0.066	0.948	-30.192	28.232

Omnibus:	873.325	Durbin-Watson:	1.731
Prob(Omnibus):	0.000	Jarque-Bera (JB):	6250.459
Skew:	2.079	Prob(JB):	0.00
Kurtosis:	10.957	Cond. No.	21.3

Table 2: OLS Regression Results for Significant Route

Dep. Variable:	duration	R-squared:	0.109
Model:	OLS	Adj. R-squared:	0.093
Method:	Least Squares	F-statistic:	6.501
Date:	Wed, 16 Feb 2022	Prob (F-statistic):	2.98e-10
Time:	17:34:02	Log-Likelihood:	-3982.6
No. Observations:	594	AIC:	7989.
Df Residuals:	582	BIC:	8042.
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	791.3505	33.283	23.776	0.000	725.980	856.721
member_True	-162.7571	27.634	-5.890	0.000	-217.031	-108.483
Rain	-15.3329	16.825	-0.911	0.363	-48.378	17.712
Snow	-105.9279	116.439	-0.910	0.363	-334.619	122.763
weekend	-18.6285	30.909	-0.603	0.547	-79.336	42.079
lunch	-6.0986	37.579	-0.162	0.871	-79.906	67.709
morning	-55.0908	28.789	-1.914	0.056	-111.634	1.452
evening	-16.8176	29.215	-0.576	0.565	-74.198	40.563
night	13.5019	28.068	0.481	0.631	-41.625	68.628
morning:weekend	11.1144	78.982	0.141	0.888	-144.010	166.239
lunch:weekend	118.2221	54.271	2.178	0.030	11.632	224.812
evening:weekend	97.7655	51.255	1.907	0.057	-2.902	198.433

Omnibus:	264.041	Durbin-Watson:	1.781
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1207.516
Skew:	2.011	Prob(JB):	6.18e-263
Kurtosis:	8.710	Cond. No.	21.6

Table 3: OLS Regression Results for Significant Route

Dep. Variable:	duration	R-squared:	0.124
Model:	OLS	Adj. R-squared:	0.117
Method:	Least Squares	F-statistic:	18.49
Date:	Wed, 16 Feb 2022	Prob (F-statistic):	5.98e-35
Time:	17:34:05	Log-Likelihood:	-8046.1
No. Observations:	1452	AIC:	1.612e+04
Df Residuals:	1440	BIC:	1.618e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	288.1200	8.060	35.745	0.000	272.309	303.931
member_True	-68.9418	7.411	-9.302	0.000	-83.480	-54.404
Rain	-5.8114	3.311	-1.755	0.079	-12.306	0.683
Snow	35.2571	13.181	2.675	0.008	9.401	61.113
weekend	-40.3892	8.738	-4.622	0.000	-57.529	-23.249
lunch	-11.5729	7.773	-1.489	0.137	-26.821	3.675
morning	-33.4805	5.699	-5.875	0.000	-44.660	-22.301
evening	-21.4088	4.425	-4.838	0.000	-30.090	-12.728
night	30.3719	7.271	4.177	0.000	16.109	44.635
morning:weekend	-2.4341	20.374	-0.119	0.905	-42.399	37.531
lunch:weekend	36.6181	16.623	2.203	0.028	4.010	69.226
evening:weekend	27.7326	14.213	1.951	0.051	-0.149	55.614

Omnibus:	517.808	Durbin-Watson:	1.669
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2207.673
Skew:	1.664	Prob(JB):	0.00
Kurtosis:	8.041	Cond. No.	22.1

Table 4: OLS Regression Results for Null Route

Dep. Variable:	duration	R-squared:	0.080
Model:	OLS	Adj. R-squared:	0.064
Method:	Least Squares	F-statistic:	4.856
Date:	Wed, 16 Feb 2022	Prob (F-statistic):	3.24e-07
Time:	17:34:08	Log-Likelihood:	-4124.3
No. Observations:	626	AIC:	8273.
Df Residuals:	614	BIC:	8326.
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	856.3858	29.112	29.417	0.000	799.216	913.556
member_True	-129.1158	25.126	-5.139	0.000	-178.458	-79.773
Rain	-22.7856	14.395	-1.583	0.114	-51.056	5.484
Snow	-101.6669	57.312	-1.774	0.077	-214.218	10.884
weekend	-4.5709	24.711	-0.185	0.853	-53.099	43.957
lunch	29.9062	33.215	0.900	0.368	-35.322	95.134
morning	46.6443	81.121	0.575	0.566	-112.665	205.953
evening	-32.0518	19.899	-1.611	0.108	-71.129	7.026
night	-43.2107	21.216	-2.037	0.042	-84.875	-1.546
morning:weekend	-342.4591	197.508	-1.734	0.083	-730.332	45.414
lunch:weekend	73.6833	54.405	1.354	0.176	-33.159	180.525
evening:weekend	-0.3008	47.336	-0.006	0.995	-93.260	92.659

Omnibus:	149.934	Durbin-Watson:	1.947
Prob(Omnibus):	0.000	Jarque-Bera (JB):	310.366
Skew:	1.316	Prob(JB):	4.03e-68
Kurtosis:	5.230	Cond. No.	43.9

Table 5: OLS Regression Results for Null Route

Dep. Variable:	duration	R-squared:	0.117
Model:	OLS	Adj. R-squared:	0.105
Method:	Least Squares	F-statistic:	9.632
Date:	Wed, 16 Feb 2022	Prob (F-statistic):	1.73e-16
Time:	17:34:11	Log-Likelihood:	-4954.5
No. Observations:	814	AIC:	9933.
Df Residuals:	802	BIC:	9989.
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	643.1525	15.870	40.526	0.000	612.001	674.304
member_True	-101.0206	14.143	-7.143	0.000	-128.782	-73.259
Rain	-10.5752	7.705	-1.373	0.170	-25.699	4.549
Snow	32.8453	29.374	1.118	0.264	-24.814	90.504
weekend	5.2203	17.186	0.304	0.761	-28.516	38.956
lunch	-1.3339	18.816	-0.071	0.944	-38.268	35.600
morning	-24.6165	10.206	-2.412	0.016	-44.650	-4.583
evening	10.8619	20.080	0.541	0.589	-28.554	50.277
night	-11.1135	18.241	-0.609	0.543	-46.919	24.692
morning:weekend	64.6655	35.873	1.803	0.072	-5.751	135.082
lunch:weekend	65.9956	32.066	2.058	0.040	3.052	128.939
evening:weekend	15.0630	37.674	0.400	0.689	-58.888	89.014

Omnibus:	172.576	Durbin-Watson:	1.787
Prob(Omnibus):	0.000	Jarque-Bera (JB):	337.699
Skew:	1.216	Prob(JB):	4.67e-74
Kurtosis:	5.011	Cond. No.	19.5

Table 6: OLS Regression Results for Null Route

Dep. Variable:	duration	R-squared:	0.116
Model:	OLS	Adj. R-squared:	0.097
Method:	Least Squares	F-statistic:	6.126
Date:	Wed, 16 Feb 2022	Prob (F-statistic):	1.76e-09
Time:	17:34:14	Log-Likelihood:	-3461.3
No. Observations:	524	AIC:	6947.
Df Residuals:	512	BIC:	6998.
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	774.2311	31.246	24.778	0.000	712.845	835.618
member_True	-155.7838	28.632	-5.441	0.000	-212.034	-99.533
Rain	31.4807	16.639	1.892	0.059	-1.209	64.170
Snow	16.7291	61.823	0.271	0.787	-104.728	138.187
weekend	16.2049	28.778	0.563	0.574	-40.333	72.742
lunch	48.4620	36.706	1.320	0.187	-23.650	120.574
morning	-42.4284	21.191	-2.002	0.046	-84.060	-0.796
evening	74.7571	28.563	2.617	0.009	18.642	130.872
night	2.7101	30.571	0.089	0.929	-57.351	62.771
morning:weekend	-126.4160	86.997	-1.453	0.147	-297.331	44.499
lunch:weekend	-72.2498	54.960	-1.315	0.189	-180.225	35.726
evening:weekend	-29.2206	57.886	-0.505	0.614	-142.944	84.502

Omnibus:	107.150	Durbin-Watson:	1.854
Prob(Omnibus):	0.000	Jarque-Bera (JB):	200.632
Skew:	1.161	Prob(JB):	2.71e-44
Kurtosis:	4.948	Cond. No.	17.3

Code Appendix

February 16, 2022

This code is nearly identical to the file provided on Canvas, but we include it here for completeness (everything needed to replicate our report is included)

[ ]:
```
import requests
from zipfile import ZipFile
import csv
import re
import numpy as np
import pickle
```

[ ]:
```
# citations:
# https://www.tutorialspoint.com/downloading-files-from-web-using-python
# https://www.analyticsvidhya.com/blog/2021/08/
 ↪python-tutorial-working-with-csv-file-for-data-science/

header = []
data = []
# We decided to stick with this time range for the sake of computation
for i in range(2010, 2012):
    url = "https://s3.amazonaws.com/capitalbikeshare-data/" + str(i) +␣
 ↪"-capitalbikeshare-tripdata.zip"
    r = requests.get(url, allow_redirects = True)
    zipfile_name = 'bikedata.zip'
    open(zipfile_name, 'wb').write(r.content)
    with ZipFile(zipfile_name, 'r') as zip:
        zip.extractall()
    csv_name = str(i) + "-capitalbikeshare-tripdata.csv"
    csvreader = csv.reader(open(csv_name))
    file = open(csv_name)
    csvreader = csv.reader(file)
    header = next(csvreader)
    for row in csvreader:
        data.append(row)
    file.close()
```

[ ]:
```
# the variables with names that end with "_tmp" will be further preprocessed
n = len(data)
duration = []
```

```
# duration of the ride in seconds
starttime_tmp = []
# start time of ride #i
station_start = []
# station ID where the bike was checked out
station_end = []
# station ID where the bike was returned
member_tmp = []
# member (1) or nonmember (0)
station_start_name = []
station_end_name = []
bikenum_tmp = []
for i in range(n):
    cur = data[i]
    duration.append(cur[0])
    starttime_tmp.append(cur[1])
    station_start.append(cur[3])
    station_end.append(cur[5])
    member_tmp.append(cur[8])
    station_start_name.append(cur[4])
    station_end_name.append(cur[6])
    bikenum_tmp.append(cur[7])
station_start = np.array(station_start).astype(np.intc)
station_end = np.array(station_end).astype(np.intc)
```

[ ]:
```
# preprocessing starttime_tmp
starttime = np.empty((n, 6))
# row i = year/month/date/hour/minute/second for ride #i
for i in range(n):
    starttime[i,] = np.array(re.split('-|:| ', starttime_tmp[i])).astype(np.
 ↪intc)
starttime = starttime.astype(np.intc)
```

[ ]:
```
# preprocessing member_tmp
member = np.array(member_tmp)
member = member == "Member"
# member (1) or nonmember (0)
```

[ ]:
```
# preprocessing bikenum_tmp
bikenum = []
for i in range(n):
    tmp = re.split('w|W|,| ', bikenum_tmp[i])
    bikenum.append([item for item in tmp if item != ''][0])
for i in range(n):
    cur = bikenum[i]
    if "?" in cur[0]:
        bikenum[i] = np.NAN
```

```
bikenum = np.genfromtxt(np.array(bikenum))
# some are NA, the data is messy for this one
```

[ ]:
```
# creating stations
# stations[i,0] = station ID for the i-th station,
# stations[i,1] = station location for the i-th station
all_stations = np.unique(np.concatenate((station_start, station_end)))
stations = []
for item in all_stations:
    ind = np.argwhere(station_start == item)
    if(len(ind) != 0):
        location = station_start_name[ind[0][0]]
    else:
        location = station_end_name[np.argwhere(station_end == item)[0][0]]
    stations.append([item, location])
stations = np.array(stations)
# note that stations get added to the program over time
```

[ ]:
```
# creating days_in_month
days_in_month = np.array([31,28,31,30,31,30,31,31,30,31,30,31]
                         + [31,28,31,30,31,30,31,31,30,31,30,31])
# Jan 2010, ..., Dec 2011
```

[ ]:
```
# creating days_since_Jan1_2010
term1 = (starttime[:,0] - 2010) * 365
term2 = np.cumsum(days_in_month)[(starttime[:,1] - 1)]
term3 = days_in_month[(starttime[:,1] - 1)]
term4 = (starttime[:,2] - 1)
days_since_Jan1_2010 = term1 + term2 - term3 + term4
```

[ ]:
```
# creating day_of_week
ind = np.mod((days_since_Jan1_2010 + 4), 7)
day_of_week = np.array(['Monday','Tuesday','Wednesday','Thursday',
               'Friday','Saturday','Sunday'])[ind]
```

[ ]:
```
# saves the data into a pickle file
filename = "bikedata.pickle"
with open(filename, 'wb') as f:
    pickle.dump([starttime, duration, bikenum, stations, station_start,
                 station_end, member, days_since_Jan1_2010, day_of_week], f)
```

Cleaning Code

February 16, 2022

[ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import tqdm
import pickle
plt.rcParams["figure.figsize"] = (7,7)
```

[ ]:
```
# Converting pickle file from the given code to a CSV
filename = "bikedata.pickle"
with open(filename, 'rb') as f:
    data = pickle.load(f)

df = pd.DataFrame(data)
df.to_csv('df.csv')
```

[ ]:
```
# Loading data for cleaning
df = pd.read_csv('df.csv')

# These fix formatting issues.
df.drop('Unnamed: 0', axis=1, inplace=True)
df['starttime'] = df['starttime'].apply(lambda x: x[1:-1].split(','))
df['month'] = df['starttime'].apply(lambda x: int(x[1]))
df['date'] = pd.to_datetime(df['starttime'].apply(lambda x: ' -'.join(x[:3])))
```

0.0.1 Data Cleaning

[ ]:
```
# Around 60k people rerack their bike at the same station. Doesn't help with␣
 ↪our analysis of routes.
df = df[df['station_start'] != df['station_end']]
#  Create a new column that can be used for convenience.
df['route'] = df['start_addy'] + ' ' + df['addy_end']
```

[ ]:
```
# Obtain number of routes and routes satisfying different conditions for the␣
 ↪report
df['route'].nunique()
np.count_nonzero(data_all['route'].value_counts() < 50)
```

```
np.count_nonzero(data_all['route'].value_counts() = 1)
```

[ ]:
```
# Isolate our analysis to the top 500 most popular routes to minimize noise and␣
↪maximize signal.
top_500_df = df[df['route'].isin(df['route'].value_counts()[:500].index)]
```

[ ]:
```
df['bikenum'].nunique()
sum(df['bikenum'].isna())
```

0.0.2 Weather data concatenation

[ ]:
```
# https://www.visualcrossing.com/weather/weather-data-services

weather = pd.read_csv('Washington 2010-09-20 to 2011-12-31.csv')
weather['datetime'] = pd.to_datetime(weather['datetime'])
weather = weather[['datetime','temp', 'conditions']]

temp = []
conditions = []
for i in range(len(weather)):
    num = sum(df['date'] == weather['datetime'][i])
    array = np.ones(num) * weather['temp'][i]
    cond = [weather['conditions'][i]] * num
    temp.append(array)
    conditions.append(cond)

temp_fin = []
for i in temp:
    temp_fin.extend(i)

temp_cond = []
for i in conditions:
    temp_cond.extend(i)

df['temperature'] = temp_fin
df['conditions'] = temp_cond
```

[ ]:
```
# This removes outliers above or below 2 standard deviations from the mean

length_before_outliers = len(df)
standard_dev = df.groupby('route')['duration'].std()
remove_index = []
outliers_above = dict(df.groupby('route')['duration'].mean() + 2 * standard_dev)

for k,v in tqdm(outliers_above.items()):
    route_df = df[df['route'] == k]
    lower = route_df['duration'].mean() - 2 * standard_dev[k]
```

```
    outlier_index = route_df[route_df['duration'].apply(lambda x: x >= v or x␣
↪<= lower)].index
    df.drop(outlier_index, inplace=True)

print('number of samples before outliers are removed: ', length_before_outliers)
print('number of samples after outliers are removed:  ', len(df))

# Keeps 500 most common routes for analysis
top_500_df = df[df['route'].isin(df['route'].value_counts()[:500].index)]
top_500_df.to_csv('final_df.csv')
```

Analysis

February 16, 2022

[ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.stats as stat
from tqdm import tqdm
import statsmodels.formula.api as smf
plt.rcParams["figure.figsize"] = (7, 7)
```

[ ]:
```
# Remember to add in cleaning code.
df = pd.read_csv('final_df.csv')

# Removing a weird CSV formatting issue.
df.drop('Unnamed: 0', axis=1, inplace=True)
```

[ ]:
```
# Formatting the "conditions" column to only include 3 categories.
df['conditions'].replace('Partially cloudy', 'Clear', inplace=True)
df['conditions'].replace('Overcast', 'Clear', inplace=True)
df['conditions'].replace('Rain, Partially cloudy', 'Rain', inplace=True)
df['conditions'].replace('Rain, Overcast', 'Rain', inplace=True)
df['conditions'].replace('Snow, Partially cloudy', 'Snow', inplace=True)
df['conditions'].replace('Snow, Overcast', 'Snow', inplace=True)
df['conditions'].unique()
```

[ ]:
```
# Creating a new "hour" column

df['starttime'] = df['starttime'].apply(lambda x: x[1:-1].split(','))
df['hour'] = df['starttime'].apply(
    lambda x: int(x[3].strip().replace('\'', '')))

def time_day(x):
    ''' Reformats the starttime column to a categorical variable.
    '''
    if x >= 6 and x < 9:
        return 'morning'
    elif x >= 11 and x < 14:
```

```
        return 'lunch'
    elif x >= 17 and x <= 19:
        return 'evening'
    elif x >= 20:
        return 'night'
    elif x < 6:
        return 'night'
    else:
        return 'everything_else'

df['hour'] = df['hour'].apply(time_day)
print(df['hour'].value_counts())
```

[ ]:
```
# Defines the weekend indicator
def day_week(x):
    ''' Reformats "day_of_week" column to a binary category.
    '''
    if x != 'Saturday' and x != 'Sunday':
        return 'weekday'
    else:
        return 'weekend'

df['day_of_week'] = df['day_of_week'].apply(day_week)
```

[ ]:
```
reg_df = df[['duration', 'member', 'days_since_Jan1_2010', 'day_of_week',␣
 ↪'temperature', 'conditions', 'route', 'hour']]

# Creating Dummy Variables:
member = pd.get_dummies(reg_df['member'], drop_first=True, prefix='member')
conditions = pd.get_dummies(reg_df['conditions'], drop_first=True)
day_of_week = pd.get_dummies(reg_df['day_of_week'], drop_first=True)
time_of_day = pd.get_dummies(reg_df['hour'], drop_first=False)

reg_df = pd.concat((reg_df, member, conditions, day_of_week,
                    time_of_day), axis=1)
reg_df.drop(['member', 'day_of_week', 'conditions',
             'temperature', 'hour', 'everything_else'], axis=1, inplace=True)
```

[ ]:
```
# Creating figure 1 in our report
plt.title('Figure 1: Predictors Correlation Matrix')
confounders = reg_df.drop(['days_since_Jan1_2010', 'route', 'duration'], axis=1)
sns.heatmap(confounders.corr(), annot=True)
plt.tight_layout()
plt.savefig('fig1.png')
```

[ ]:
```
# Regression formula
```

```
formula='duration ~ member_True + Rain + Snow + weekend + lunch + morning +␣
 ↪evening + night + morning * weekend + lunch * weekend + evening * weekend'
```

[ ]:
```
# Calculates the p-value by permuting the days since Jan 1st and calculating␣
 ↪the correlation of the days since
# Jan 1 with the OLS residuals. The null hypothesis is that there should be no␣
 ↪correlation when
# permuting the days since and calculating the correlation with the residuals.

def permut_p_value(resid, days_since, num_perm=25000):
    avg_correlation = []
    for i in range(num_perm):
        permutation = np.random.permutation(days_since)  # Permutes the column.
        avg_correlation.append(np.corrcoef(results.resid, permutation)[1, 0])
    avg_correlation = np.array(avg_correlation)
    # Checks if the real correlation is significantly different from the␣
 ↪permuted one
    p_val = (1 + sum(np.abs(avg_correlation) > np.abs(np.corrcoef(results.
 ↪resid, days_since)[1, 0])))/(1 + num_perm)
    return p_val
```

[ ]:
```
# Running regressions and permutation tests
p_vals = []
model_resuls = []
number_observations = []

for route in tqdm(reg_df.route.unique()):
    # Getting the pandas dataframe into the right form for least squares.
    X = reg_df[reg_df['route'] == route]
    days_since = X['days_since_Jan1_2010']
    y = X['duration']
    # X.drop(['route', 'days_since_Jan1_2010', 'duration'], axis=1,␣
 ↪inplace=True)
    number_observations.append(len(X))

    # Running OLS
    X = sm.add_constant(X)
    model = smf.ols(formula = formula, data = X)
    # model = sm.OLS(y, X)
    results = model.fit()

    # Collecting the results
    p_val = permut_p_value(results.resid, days_since)
    p_vals.append(p_val)
    model_resuls.append(results)
```

[ ]:
```
# Getting smallest and average number of observations for our report
print('Smallest number of observations: ', min(number_observations))
print('Average Number of Observations: ',  np.mean(number_observations))
```

[ ]:
```
# Creating histogram of p-values (Figure 5 in paper)
p = sns.histplot(p_vals)
p.set_xlabel('P-values', fontsize=13)
p.set_ylabel('Count', fontsize=13)
p.set_title('Figure 5: Histogram of P-values', fontsize=15)
print('Number of unadjusted p-values less than or equal to 0.05', sum(np.
 ↪array(p_vals) <= 0.05))
plt.savefig('fig5.png')
```

Going over Multiple Testing:

[ ]:
```
# Implementing Benjamini-Hochberg Procedure
reject_lst, corrected_pvals, z, k = stat.multitest.multipletests(
    p_vals, alpha=0.05, method='fdr_bh', returnsorted=False)
print('Number of rejections using BH', sum(reject_lst))
```

[ ]:
```
# This code block will track the real correlation values (not the permuted␣
 ↪ones) for plotting and analysis.
lst_real_corr = []
for route in tqdm(reg_df.route.unique()):
    # X = reg_df.drop(['duration'], axis=1)
    # X = X[X['route'] == route]

    X = reg_df[reg_df['route'] == route]

    y = reg_df[reg_df['route'] == route]['duration']
    days_since = X['days_since_Jan1_2010']

    # X.drop(['route', 'days_since_Jan1_2010'], axis=1, inplace=True)

    X = sm.add_constant(X)
    model = smf.ols(formula = formula, data = X)
    # model = sm.OLS(y, X)
    results = model.fit()

    real_corr = np.corrcoef(results.resid, days_since)[1, 0]
    lst_real_corr.append(real_corr)
```

[ ]:
```
# Creates figure 4 in our report (histogram of actual correlations)
p = sns.histplot(lst_real_corr, stat='percent')
p.set_xlabel('Correlation for each Route', fontsize=13)
p.set_ylabel('Percentage', fontsize=13)
p.set_title(
```

```
    'Figure 4: Correlations Between Residuals and Days Since Jan1 2010',␣
 ↪fontsize=13)

plt.axvline(np.mean(lst_real_corr), color='k', linestyle='-',
            linewidth=2.5, label='Mean Correlation')
plt.legend()
np.mean(lst_real_corr)
plt.savefig('fig4.png')
```

[ ]:
```
# Checks number of positive and negative correlations out of the ones rejected␣
 ↪by BH
positive = 0
neg = 0
signif_pvals = np.where(corrected_pvals <= 0.05)[0]
for i in signif_pvals:
    if lst_real_corr[i] >= 0:
        positive += 1
    else:
        neg += 1

print(f'significant positive correlation {positive}')
print(f'significant negative correlation {neg}')
```

0.0.1 This section focuses on the main 6 routes for plotting.

[ ]:
```
# Slightly Changing original function so it can plot residuals
def permut_p_value(resid, days_since, num_perm=25000):
    permutation_corr = []
    for i in range(num_perm):
        permutation = np.random.permutation(days_since)
        # The [1,0] indexing just grabs one of the correlation values in the␣
 ↪matrix.
        permutation_corr.append(np.corrcoef(results.resid, permutation)[1, 0])
    permutation_corr = np.array(permutation_corr)
    p_val = (1+sum(np.abs(permutation_corr) > np.abs(
        np.corrcoef(results.resid, days_since)[1, 0])))/(1+num_perm)
    return p_val, permutation_corr
```

[ ]:
```
# Indexes of 3 significant and 3 insiginficant routes chosen randomly
chosen_routes = [184, 114, 104, 255, 432, 72]
lst_real_corr = []
perm_corr = []
lst_residuals = []
lst_days_since = []
p_values_lst = []
model_results = []
```

```
# This loop is a very similar setup to the initial model creation.
for route_index in chosen_routes:
    # Creating individual dataframes for each Route.
    route = reg_df.route.unique()[route_index]
    X = reg_df[reg_df['route'] == route]
    y = reg_df[reg_df['route'] == route]['duration']
    days_since = X['days_since_Jan1_2010']

    # Running OLS.
    y = reg_df[reg_df['route'] == route]['duration']
    days_since = X['days_since_Jan1_2010']
    X = sm.add_constant(X)
    model = smf.ols(formula = formula, data = X)
    X = sm.add_constant(X)

    results = model.fit()
    print(results.summary())
    real_corr = np.corrcoef(results.resid, days_since)[1, 0]
    p_val, corr_avg = permut_p_value(results.resid, days_since)

    # Collecting Results.
    lst_residuals.append(results.resid)
    lst_days_since.append(days_since)
    lst_real_corr.append(real_corr)
    p_values_lst.append(p_val)
    perm_corr.append(corr_avg)
    model_results.append(results)
```

[ ]:
```
# Standardizing residuals for Q-Q plot
lst_std_resid = []
for result in model_results:
    lst_std_resid.append((result.resid - result.resid.mean()) / result.resid.
 ↪std())
```

[ ]:
```
# Creating Q-Q plots of standardized residuals (figure 3)

fig, axes = plt.subplots(2, 3, figsize=(23, 10))

p = sm.qqplot(data = lst_std_resid[0], ax = axes[0,0], line="45", marker='o')

axes[0, 0].set_title(f'Adjusted p-value: {np.
 ↪around(corrected_pvals[chosen_routes[0]],3)}')

p = sm.qqplot(lst_std_resid[1], ax = axes[0,1], line="45", marker='o')
axes[0, 1].set_title(f'Adjusted p-value: {np.
 ↪around(corrected_pvals[chosen_routes[1]],3)}')
```

```
p = sm.qqplot(lst_std_resid[2], ax = axes[0,2], line="45", marker='o')
axes[0, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[2]],3)}')

p = sm.qqplot(lst_std_resid[3], ax = axes[1,0], line="45", marker='o')
axes[1, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[3]],3)}')

p = sm.qqplot(lst_std_resid[4], ax = axes[1,1], line="45")
axes[1, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[4]],3)}')

p = sm.qqplot(lst_std_resid[5], ax = axes[1,2], line="45", marker='o')
axes[1, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[5]],3)}')

plt.suptitle('Figure 3: Quantile-Quantile Plot for 3 Significant and 3 Null␣
 ↪Routes', fontsize=20)
plt.tight_layout()
plt.savefig('fig3.png')
```

[ ]:
```
# Gathering residuals after grouping by "days since Jan 10th" for future␣
 ↪plotting.
plot_resid = []
plot_days_since = []
for i in range(len(lst_residuals)):
    k = pd.DataFrame({'lst_residuals': lst_residuals[i], 'lst_days_since':␣
 ↪lst_days_since[i]}).groupby(
        'lst_days_since')['lst_residuals'].mean()
    plot_days_since.append(k.index)
    plot_resid.append(k.values)
```

[ ]:
```
# Plotting null distribution of correlations from permutation tests (figure 7)
fig, axes = plt.subplots(2, 3, figsize=(18, 10), sharey=True)

sns.histplot(ax=axes[0, 0], data=perm_corr[0], )
axes[0, 0].axvline(lst_real_corr[0], color='r')
axes[0, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[0]],3)}')

sns.histplot(ax=axes[0, 1], data=perm_corr[1], )
axes[0, 1].axvline(lst_real_corr[1], color='r', label='True Correlation Value')
axes[0, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[1]],3)}')

sns.histplot(ax=axes[0, 2], data=perm_corr[2], )
```

```
axes[0, 2].axvline(lst_real_corr[2], color='r', label='True Correlation Value')
axes[0, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[2]],3)}')

sns.histplot(ax=axes[1, 0], data=perm_corr[3], )
axes[1, 0].axvline(lst_real_corr[3], color='r')
axes[1, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[3]],3)}')
axes[1, 0].set_xlabel('Residual Correlations')

sns.histplot(ax=axes[1, 1], data=perm_corr[4], )
axes[1, 1].axvline(lst_real_corr[4], color='r', label='True Correlation Value')
axes[1, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[4]],3)}')
axes[1, 1].set_xlabel('Residual Correlations')

sns.histplot(ax=axes[1, 2], data=perm_corr[5], )
axes[1, 2].axvline(lst_real_corr[5], color='r', label='True Correlation Value')
axes[1, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[5]],3)}')
axes[1, 2].set_xlabel('Residual Correlations')

plt.suptitle('Figure 7: Null Distribution of Correlations', fontsize=15)

axes[0, 2].legend(loc="upper right", bbox_to_anchor=(
    1.7, 1), prop={'size': 12})
plt.savefig('fig7.png')
```

[ ]:
```
# Plotting average residuals against days since Jan1 (figure 6)
fig, axes = plt.subplots(2, 3, figsize=(23, 10))
p = sns.regplot(ax=axes[0, 0], x=plot_days_since[0], y=plot_resid[0], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
p.set_ylabel('Residuals', fontsize=14)
axes[0, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[0]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')

p = sns.regplot(ax=axes[0, 1], x=plot_days_since[1], y=plot_resid[1], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[0, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[1]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')

p = sns.regplot(ax=axes[0, 2], x=plot_days_since[2], y=plot_resid[2], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[0, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[2]],3)}')
```

```
p.set_xlabel('Number of Days Since January 1st 2010')


p = sns.regplot(ax=axes[1, 0], x=plot_days_since[3], y=plot_resid[3], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[3]],3)}')
p.set_ylabel('Residuals', fontsize=14)
p.set_xlabel('Number of Days Since January 1st 2010')


p = sns.regplot(ax=axes[1, 1], x=plot_days_since[4], y=plot_resid[4], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[4]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')

p = sns.regplot(ax=axes[1, 2], x=plot_days_since[5], y=plot_resid[5], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[5]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')


plt.suptitle(
    'Figure 6: Residual Plots from 3 Significant and 3 Null Routes ',␣
 ↪fontsize=20)
plt.tight_layout()
plt.savefig('fig6.png')
```

[ ]:
```
# Plotting residuals against fitted values (figure 2)
fig, axes = plt.subplots(2, 3, figsize=(23, 10))
p = sns.residplot(ax=axes[0, 0], x=model_results[0].fittedvalues,␣
 ↪y=model_results[0].resid, line_kws={
                "color": "black"}, scatter_kws={"color": "red"})

p.set_ylabel('Residuals', fontsize=14)
axes[0, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[0]],3)}')

p.set_xlabel('Fitted Values')


p = sns.residplot(ax=axes[0, 1], x=model_results[1].fittedvalues,␣
 ↪y=model_results[1].resid, line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[0, 1].set_title(
```

```
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[1]],3)}')

p.set_xlabel('Fitted Values')

p = sns.residplot(ax=axes[0, 2], x=model_results[2].fittedvalues,␣
 ↪y=model_results[2].resid, line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[0, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[2]],3)}')

p.set_xlabel('Fitted Values')


p = sns.residplot(ax=axes[1, 0], x=model_results[3].fittedvalues,␣
 ↪y=model_results[3].resid, line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[3]],3)}')
p.set_ylabel('Residuals', fontsize=14)
p.set_xlabel('Fitted Values')


p = sns.residplot(ax=axes[1, 1], x=model_results[4].fittedvalues,␣
 ↪y=model_results[4].resid, line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[4]],3)}')
p.set_xlabel('Fitted Values')

p = sns.residplot(ax=axes[1, 2], x=model_results[5].fittedvalues,␣
 ↪y=model_results[5].resid, line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[5]],3)}')

p.set_xlabel('Fitted Values')


plt.suptitle('Figure 2: Fitted Values vs. Residuals for 3 Significant and 3␣
 ↪Null Routes', fontsize=20)
plt.tight_layout()

plt.savefig('fig2.png')
```

[ ]:
```
# Plotting ride durations against days since Jan1 (figure 8)
duration = []
days_since = []
```

```
chosen_routes = [184, 114, 104, 255, 432, 72]
for i in chosen_routes:
    route = reg_df.route.unique()[i]
    mask = reg_df[reg_df['route'] == route]
    days_since.append(mask['days_since_Jan1_2010'])
    duration.append(mask['duration'])

fig, axes = plt.subplots(2, 3, figsize=(23, 10))
p = sns.regplot(ax=axes[0, 0], x=days_since[0], y=duration[0], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
p.set_ylabel('Duration', fontsize=14)
axes[0, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[0]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')


p = sns.regplot(ax=axes[0, 1], x=days_since[1], y=duration[1], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[0, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[1]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')

p = sns.regplot(ax=axes[0, 2], x=days_since[2], y=duration[2], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[0, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[2]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')


p = sns.regplot(ax=axes[1, 0], x=days_since[3], y=duration[3], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 0].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[3]],3)}')
p.set_ylabel('Duration', fontsize=14)
p.set_xlabel('Number of Days Since January 1st 2010')


p = sns.regplot(ax=axes[1, 1], x=days_since[4], y=duration[4], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 1].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[4]],3)}')
p.set_xlabel('Number of Days Since January 1st 2010')

p = sns.regplot(ax=axes[1, 2], x=days_since[5], y=duration[5], line_kws={
                "color": "black"}, scatter_kws={"color": "red"})
axes[1, 2].set_title(
    f'Adjusted p-value: {np.around(corrected_pvals[chosen_routes[5]],3)}')
```

```
p.set_xlabel('Number of Days Since January 1st 2010')


plt.suptitle(
    'Figure 8: Number of Days Since January 1st Against Ride Duration for 3␣
 ↪Significant and 3 Null Routes', fontsize=20)
plt.tight_layout()
plt.savefig('fig8.png')
```

1 Project **94 / 100**

1 Statistical methodology (30 pts)

The aims are:

* The analysis should identify all the major relevant challenges, such as issues of multiple testing, non-independence, confounding, exploratory data analysis, etc

* These challenges should be handled appropriately in the analysis, applying existing tools or developing new techniques

* Assumptions should be tested and examined using, e.g., using diagnostic plots

* Any remaining issues that cannot be addressed, should be discussed

+ **24 pts** Click here to replace this description.
+ **22 pts** Click here to replace this description.
+ **26 pts** Click here to replace this description.
+ **25 pts** Click here to replace this description.
+ **20 pts** Click here to replace this description.
+ **27 pts** Click here to replace this description.
+ **15 pts** Click here to replace this description.
+ **30 pts** Click here to replace this description.
+ **23 pts** Click here to replace this description.
+ **21 pts** Click here to replace this description.
+ **16 pts** Click here to replace this description.
+ **18 pts** Click here to replace this description.

✓ **+ 28 pts Click here to replace this description.**

2 Questions, ideas, & design of analysis (40 pts)

The aims are:

* The project should be designed in a creative and thoughtful way, to address interesting questions and challenges in the data

* The real world meaning of the data should be considered in an insightful way to guide the design of the analysis

* Choices made along the way, for example designing a test statistic or finding a way to measure or visualize results, should be addressed thoughtfully

* The analysis should show thorough understanding of any preexisting tools, code, packages, etc, that the group chose to use, and these choices are well suited to the problem at hand

+ **32 pts** Click here to replace this description.
+ **30 pts** Click here to replace this description.
+ **36 pts** Click here to replace this description.
+ **34 pts** Click here to replace this description.
+ **25 pts** Click here to replace this description.
+ **38 pts** Click here to replace this description.

✓ **+ 40 pts Click here to replace this description.**

+ **20 pts** Click here to replace this description.

+ **28 pts** Click here to replace this description.
+ **26 pts** Click here to replace this description.
+ **37 pts** Click here to replace this description.

3 Report & code (30 pts)

The aims are:

* The report should be clear and well-written, presenting a cohesive and well motivated explanation of the path followed in the analysis, and thoughtful and justified conclusions based on the findings
* Open questions, uncertainties due to insufficient data, questions relating to untestable assumptions, etc, should be addressed as needed
* The code should be clear, well organized, and appears readable and reproducible
* Sufficient details should be given to understand the specifics of the analysis being run and the choices made along the way

+ **24 pts** Click here to replace this description.
+ **25 pts** Click here to replace this description.
+ **23 pts** Click here to replace this description.
+ **28 pts** Click here to replace this description.
+ **22 pts** Click here to replace this description.
+ **29 pts** Click here to replace this description.
+ **15 pts** Click here to replace this description.
+ **30 pts** Click here to replace this description.

✓ **+ 26 pts Click here to replace this description.**

+ **23 pts** Click here to replace this description.

Great work on your project! Below are comments for each of the 3 sections of the rubric:

1. Overall, your analysis approaches statistical issues very thoughtfully and does a great job of handling multiple testing, confounding, and other issues. Permuting residuals is a great way to avoid some of the issues that might arise. A few suggestions:

* We should be cautious of overinterpretation that might just arise from multiple testing - for example, your discussion of the meaning of coefficients on the lunch hour indicator is interesting and the trends are certainly worth exploring, but there is an implicit multiple testing issue here and it's not clear if there are any significant trends relating to this indicator that we should be interpreting.
* You mention that checking for nonconstant variance is mostly an issue for confidence intervals and should not affect the permutation test, but this is not necessarily the case. For example suppose that X_i ~ N(0, sigma^2_i) where sigma^2_i increases over time i=1,..,n. Then if we regress X_i onto time i, and run a permutation test to check for a change in the mean over time (i.e., the null is actually true), without permutation the variance is highest for large i and this might create some outliers at high leverage points (i.e., extreme X_i values at extreme i values) ; for the permuted data, on the other hand, the variances are randomly shuffled and this will not occur. This likely isn't a major issue but is worth considering.

2. You make many thoughtful choices in the analysis, such as considering interactions between time and day of week to account for rush hour. You find data to incorporate weather confounder, which is good. You compare the plots of residuals vs time for routes rejected in permutation test to the ones that are not rejected, which is great for a deeper understanding of the findings. Your choices are very well described and clearly considered, e.g. BH vs Storey, residuals vs grouped permutations, etc.

3. Your report is extremely detailed and well written, and includes many thoughtful discussions of different aspects of the analysis. You also discuss possible limitations and problems with assumptions along the way. It would be clearer with more organization, e.g., subsections to indicate the flow of the analysis. The report is also lacking in concrete details and models at times -- it would be better to use some equations etc throughout the report rather than describing procedures only in words, for greater clarity (for example, describing permuting residuals is done only in words, and would be much clearer with a few equations.)