Project 1: Capital Bikeshare Analysis

Soren Dunn, Tytus Wilam, Aim Wonghirundacha, and Simone Zhang

2023-02-09

Background and Introduction

In this project, we deal with the public data set from Washington D.C.'s bikeshare program, which records every individual ride taken. The analysis on the patterns of change of riding time of routes and identification on routes that have significant change is meaningful. For example, the government get to learn whether the infrastructure such as bridge or road built over time actually benefit people as providing convenience for commuting. Moreover, other topics such as cost and benefit analysis can also be discussed further with the combination of extra data sets.

We aim to find whether the duration time of each route changes over time controlling for possible confounders such as rush hour, membership, season and weekdays. To achieve this goal, we've constructed task specific permutation tests to check for the correlation between a route's duration and time. We've built linear regression models to test on the significance of time. At the same time, addressing the multiple testing issue by applying modified BH.

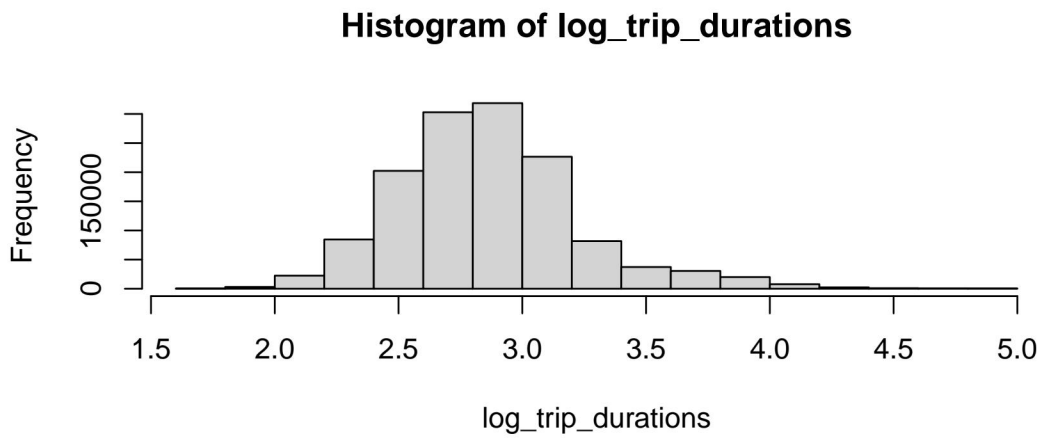
Basic Data Exploration

```
# Create dataframe of all the data
bike_df = as.data.frame(starttime)
bike_df$duration = duration
bike_df$station_start = station_start
bike_df$station_end = station_end
bike_df$bike_num = bikenum
bike_df$member = member
bike_df$days_since_Jan1_2010 = days_since_Jan1_2010
bike_df$day_of_week = day_of_week

# Define a function to help convert the start and end points to a properly formatted
# route character string
convert_to_route = function(x) {
  paste("r",as.character(x),sep="")
}

# Create a unique variable which is different for each route (each combination of start and end points)
bike_df$route = paste(lapply(station_end,convert_to_route),lapply(station_start,as.character), sep="_")
bike_df$is_weekday = (bike_df$day_of_week != "Saturday") & (bike_df$day_of_week != "Sunday")

# Rename the original variable names to more descriptive ones
names(bike_df)[names(bike_df) == 'V1'] <- 'Year'
names(bike_df)[names(bike_df) == 'V2'] <- 'Month'
names(bike_df)[names(bike_df) == 'V3'] <- 'Day'
names(bike_df)[names(bike_df) == 'V4'] <- 'Hour'
```

```r
names(bike_df)[names(bike_df) == 'V5'] <- 'Minute'
names(bike_df)[names(bike_df) == 'V6'] <- 'Second'

# Identify the trips that occurred during rush hour
# Let's define rush hour as 7-9 am and 5-7 pm
# This corresponds to hours 7-9 and 17-19
bike_df$rush_hour = bike_df$Hour %in% c(7,8,9,17,18,19)

# Define new factor levels for each season.
bike_df$Month = as.factor(bike_df$Month)
bike_df$season = revalue(bike_df$Month, c("1"="Winter", "2"="Winter","3"="Spring","4"="Spring","5"="Spr

head(bike_df)
```

```
##   Year Month Day Hour Minute Second duration station_start station_end bike_num
## 1 2010     9  20   11     27      4     1012         31208       31108      742
## 2 2010     9  20   11     41     22       61         31209       31209       32
## 3 2010     9  20   12      5     37     2690         31600       31100      993
## 4 2010     9  20   12      6      5     1406         31600       31602      344
## 5 2010     9  20   12     10     43     1413         31100       31201      883
## 6 2010     9  20   12     14     27      982         31109       31200      850
##   member days_since_Jan1_2010 day_of_week        route is_weekday rush_hour
## 1   TRUE                  262      Monday r31108_31208       TRUE     FALSE
## 2   TRUE                  262      Monday r31209_31209       TRUE     FALSE
## 3   TRUE                  262      Monday r31100_31600       TRUE     FALSE
## 4   TRUE                  262      Monday r31602_31600       TRUE     FALSE
## 5   TRUE                  262      Monday r31201_31100       TRUE     FALSE
## 6   TRUE                  262      Monday r31200_31109       TRUE     FALSE
##   season
## 1   Fall
## 2   Fall
## 3   Fall
## 4   Fall
## 5   Fall
## 6   Fall
```

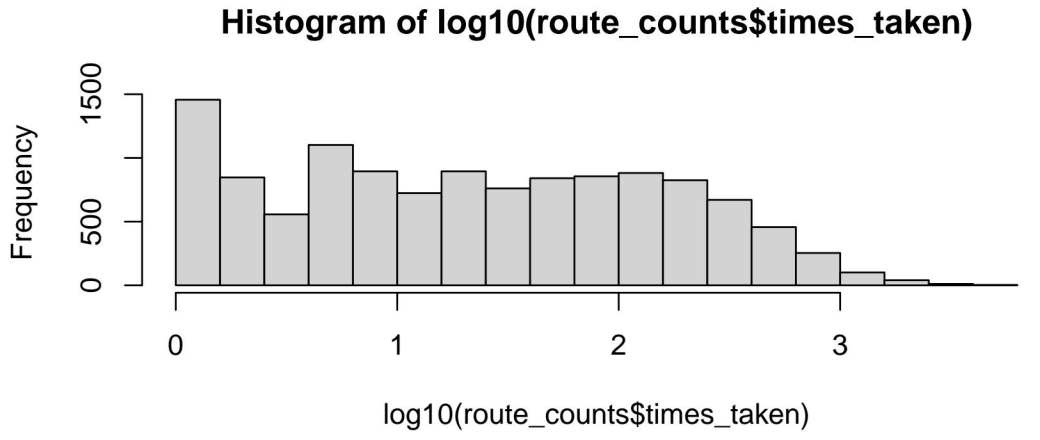
```r
# Check the basic route distribution
log_trip_durations = log10(bike_df$duration)
hist(log_trip_durations)
```




```
sum = summary(bike_df$duration)
total = sum(bike_df$duration > 0)
over_4 = sum(bike_df$duration > 60 * 60 * 4)
over_12 = sum(bike_df$duration > 60 * 60 * 12)
over_24 = sum(bike_df$duration > 60 * 60 * 24)
```

To better understand the Capital Bikeshare dataset we first investigated some basic properties of the dataset. It has data from 1,342,364 trips out of which 5,251 lasted more than 4 hours, 951 more than 12 hours, and none lasted more than 24 hours. As seen above, this still results in a distribution of ride times that spans almost 3 orders of magnitude. Due to the large span of the duration times, we chose to use log(duration) over pure duration for many of our regressions and visualizations to reduce the skew of the data (which makes visualizations more clear).

```
# Count the number of times each unique route was taken
times_taken = t(rbind(table(bike_df$route)))
route_counts = data.frame(times_taken)
route_counts$route <- rownames(route_counts)

hist(log10(route_counts$times_taken))
```

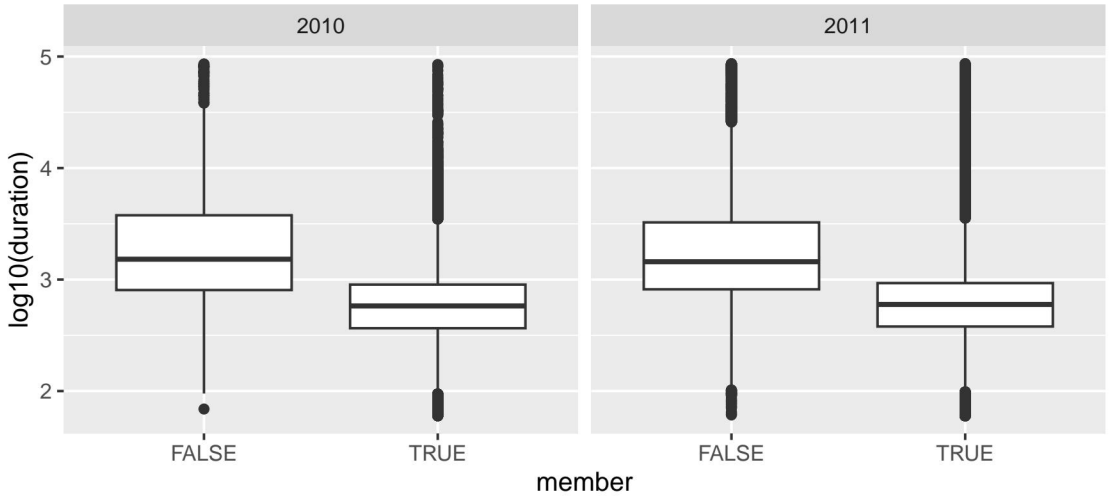


```
routes_taken_20 = sum(route_counts$times_taken < 20)
routes_taken_100 = sum(route_counts$times_taken > 100)
routes_taken_1000 = sum(route_counts$times_taken > 1000)
sum = summary(route_counts$times_taken)
```

The number of times each route was taken is also a very skewed distributions with around 6017 having been taken fewer than 20 times, around 3244 having been taken over a hundred times, and 155 of the routes having been taken over a thousand times. This skew can also be seen in the histogram of the number of times different routes were taken with a log transform again being necessary to properly visualize the data. As we will discuss later on, we will select routes with more than 100 rides taken for both the realistic implication and permutation test purposes.

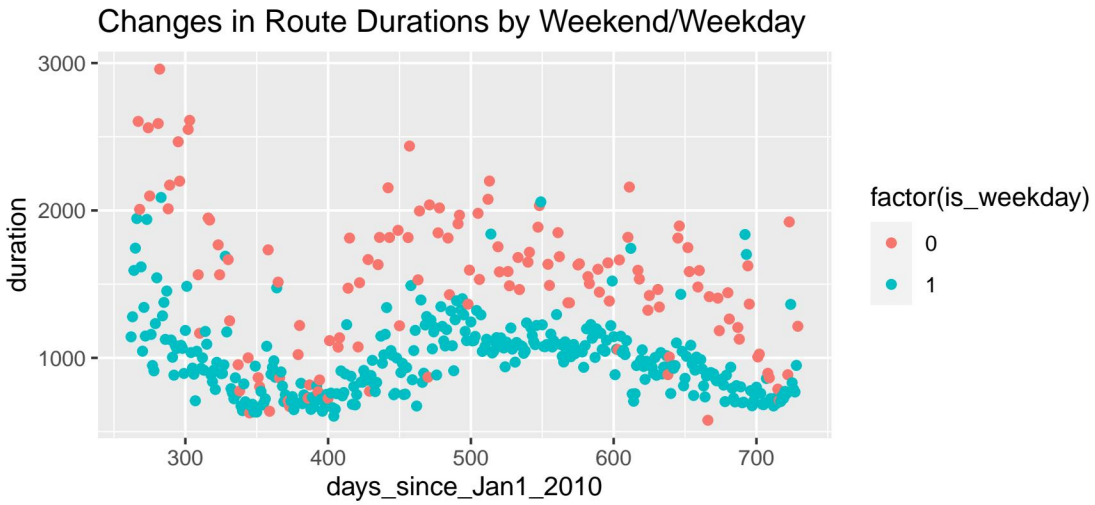
Checking for useful confounders

```
ggplot(data=bike_df, aes(x=member,y=log10(duration))) +
  geom_boxplot() +
  facet_wrap(~Year)
```



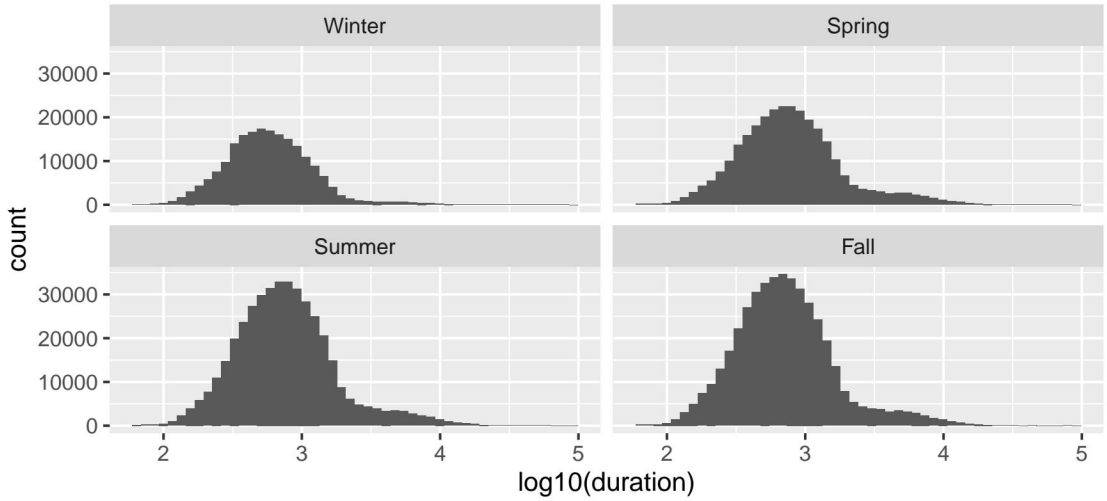
After these basic checks, we visualized the likely confounders for the route duration to select confounders to use for later permutation tests. We had to select from among the plausible confounders since sub-setting the data into too many groups for the permutation tests would yield too little data for the majority of the routes. First we simultaneously plotted boxplots by year and whether the bike rider was a member. Boxplots were chosen for this visualization since they nicely showed not just the median between each subset but also gave a clear visual way to distinguish between each subset's inter-quartile range. Based on this plot, whether the rider was a member seems like a more important confounder than the year.

```
grouped_df = aggregate(cbind(duration,is_weekday) ~ days_since_Jan1_2010, data=bike_df, mean)
ggplot(data=grouped_df, aes(x=days_since_Jan1_2010,y=duration,color=factor(is_weekday))) +
  geom_point() +
  ggtitle("Changes in Route Durations by Weekend/Weekday")
```



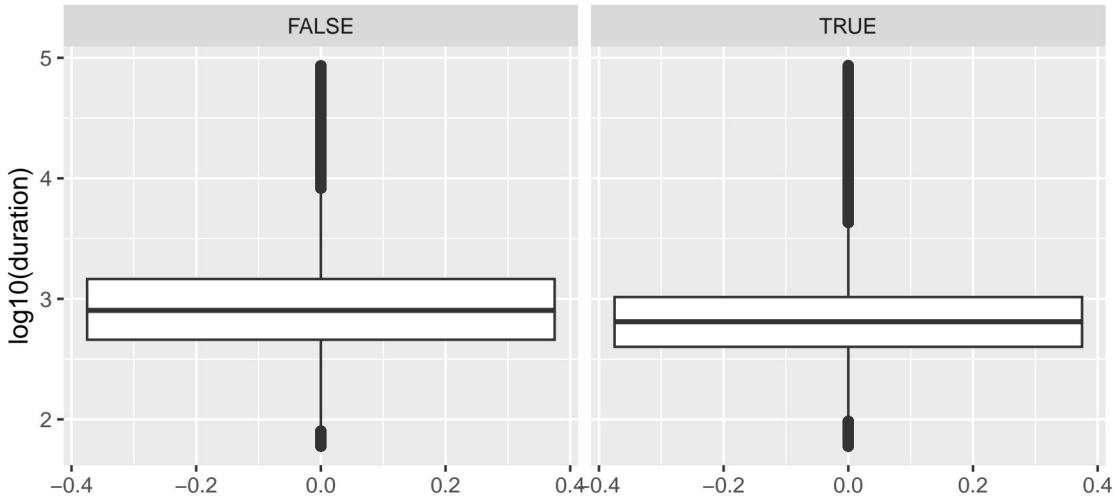
The changes in route durations over time were also visualized by whether the trip was taken on a weekday. The routes that were taken on weekends have persistently larger duration than weekdays. However the routes' duration do not seem to have a linear relationship with days since Janurary 2010. We see that routes' duration dip during winter months are shorter than summer months.

```
ggplot(data=bike_df, aes(x=log10(duration))) +
  geom_histogram(bins = 50) +
  facet_wrap(~season)
```



These seasonal differences can be seen more clearly in the histograms above. Using season as a confounder results in some differences between the groups, but not as many as we would initially expect from the previous trends in route duration over time. Generally the distribution of route durations is shifted to the left during Winter and Fall and to the right during Spring and Summer. This makes sense since individuals would be much more likely to go on long bike-rides during warmer months than colder ones.

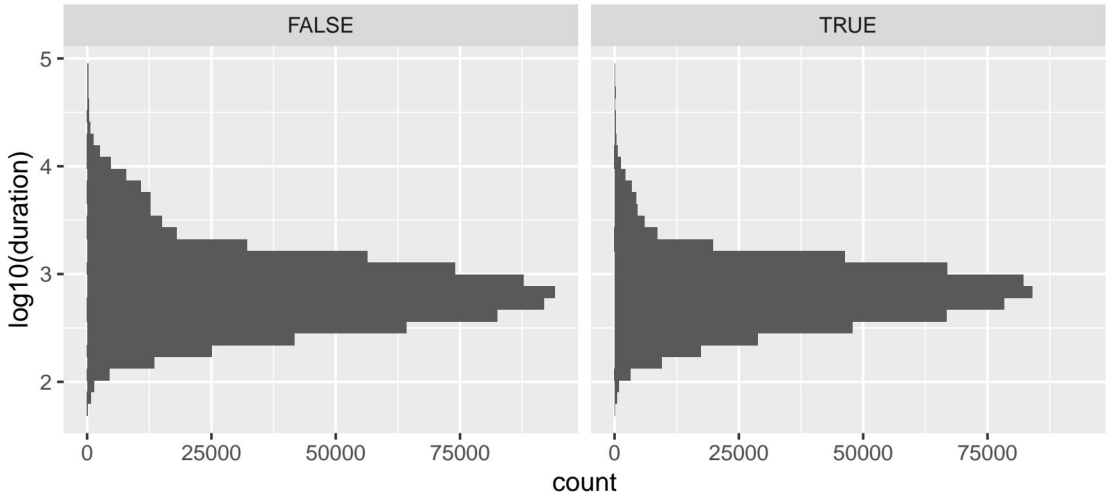
```
ggplot(data=bike_df, aes(y=log10(duration))) +
  geom_boxplot() +
  facet_wrap(~is_weekday)
```



As shown in the boxplot above, whether the trip happened on a weekday or weekend has a stark difference in the route duration and thus is another highly important confounder to test.

A possible explanation for this is that people who takes this route during weekday is for commuting purposes so that their duration would be shorter than those who use the route in weekends for entertaining purpose for example.

```
ggplot(data=bike_df, aes(y=log10(duration))) +
  geom_histogram() +
  facet_wrap(~rush_hour)
```



Based on this plot, rush hour actually does not appear to be a significant confounder. For this reason and technical issues of dividing the subgroup, we do not account for whether the trip happened during rush hour in most of the permutation test analysis. However we did include it in the linear regression analysis.

One concern we have with selecting confounders to use in the later permutation tests and regression based on visualizations is that it is in some ways using the same data twice: once to select which confounders to use and another time to run the main analysis. Ideally we would hope to expand this work by (having selected confounders to control for using the 2010-2011 data) running the same analysis on future years' routes to get results unbaised by using the data to both select and analyze the confounders.

Permutation Tests

```
# Multiple Testing Functions

# Regular BH procedure
regular_BH = function(p, alpha) {
  n <- length(p)
  k <- 1:n
  idx = order(p)
  reject <- ifelse(p[idx] <= alpha * (k / n), TRUE, FALSE)
  k <- max(which(reject == TRUE))
  if (k == -Inf) {
    return("No signals")
  } else {
    reject[1:k] <- TRUE
    reject = reject[order(p)]
    return(reject)
  }
}

# Storey BH procedure
storey_BH = function(P, alpha, gamma){
    n = length(P)
    order_index = order(P, decreasing=FALSE)
    pi_hat = sum(P>gamma)/n/(1-gamma)
    P[P>gamma] = Inf    # never reject any p > gamma
    ks = which(P[order_index] <= (1:n)*alpha/n/pi_hat)
    if(length(ks)==0){
        return(NULL)
    }else{
        k = max(ks)
        rej_index = order_index[1:k]
        return(rej_index)
    }
}

# Group adaptive BH with groups being the starting station.
group_adaptive_BH_startingstation <- function(P,groups,alpha,gamma) {
  n = length(P)
  pval = split(p_values, groups)
  pi = 1:length(groups)
  for (i in 1:length(pval)) {
    pi[groups == names(pval)[i]] = length(pval[[i]][pval[[i]] > gamma]) / (length(pval[[i]])*(1-gamma))
  }
  p_tilda <- pi * P
  reject = storey_BH(p_tilda, alpha, gamma)
  return(reject)
}

# Takes in a vector of p-values, alpha, and group size (number of p-values for each route). Runs BH. If
confounder_adjustment = function(P, alpha, confounder_group_size) {
  P[is.na(P)] = 1
  l = length(P) / confounder_group_size
  total_rejections = (p.adjust(p_values,method="BH") < 0.05)
```

```
  split_rejections = split(total_rejections, rep(1:l, each=confounder_group_size))
  route_rejections = rep(FALSE, l)
  for (i in 1:l) {
    if (TRUE %in% split_rejections[[i]]) {
        route_rejections[i] = TRUE
    }
    else {
        route_rejections[i] = FALSE
    }
  }
  return(route_rejections)
}
```

To determine changes in each route, we started by conducting a permutation test for each route. First, we selected only routes with over 100 data points. We do not think we would get any reliable changes in the routes with too few data points. We think it is also more computationally feasible to run. If we only choose routes that have more than 100 rides, it will cover 86.7% of the original data points.

The test statistic we used is the correlation between the days since Jan 2010 and the duration of the bike ride. We first try this without any confounders and only look at the relationship between the days and the duration. We run 10000 permutations and calculate a p-value for each route. Since we are comparing the p-values for multiple routes, we chose to use the Benjamini-Hochberg procedure to determine the routes which may have a significant change over the time period. We chose Benjamini-Hochberg over the Holm-Bonferroni procedure because with over 6000 routes, we expect Holm-Bonferroni would be too conservative and it would be difficult to reject any p-values since we also only run 10000 permutations.

We also tried accounting for confounders that had appeared significant in the initial data analysis. We did not simply permute the days and run multiple linear regressions for each route with the permuted day values because this sort of permutation test does not actually test the correct null. It tests the null that the independent variable has no association to the dependent variable or any of the confounders. This is a very unrealistic assumption since confounders are often associated with the independent variable. For this reason we tried many different strategies for accounting for confounders within each of the routes to get around this issue. The confounders we included included whether the individual taking the trip was a member, whether the trip was taken on a weekday or weekend, and what season the trip was taken in (Winter, Spring, Summer, or Fall). The season condition was included as a granular test for the weather that the route was taken in; quite intuitively the initial visualizations showed that trips took longer in Winter (likely due to snow and cold slowing bikers down). This resulted in a total of 16 sets of distinct combinations of confounders for 16 sets of confounder groups per route. We then permuted within each of these sets of confounder groups within each route to see if any of the confounder groups had a greater correlation between days since January 2010 and route duration than would be suggested by the null. This corresponded to the null that there were no changes in route durations for any of the confounder groups for any of the routes. We then counted a route as displaying a significant change in route duration over time if any of the subpopulations within the route displayed a significant p-value after conducting a BH adjustment on the entire set of p-values. One issue with this approach is that it does not control the primary FDR we aim to control (the FDR in terms of routes). One approach that could be applied in further work to solve this issue would be to permute within each of the groups but then only calculate one p-value for each route based on the permuted days (that were only permuted within each of the confounding groups). We did this solely when controlling for rush hour but ideally we would extend that analysis to other confounders.

```
# Get p-values for each route without confounders and run
permutation_test = function(route_subset, tfunc) {
  b_data = tfunc(route_subset$duration,route_subset$days_since_Jan1_2010)
  unshuffled_routes = matrix(rep(route_subset$days_since_Jan1_2010,10000), ncol=10000)
  shuffled_routes = colShuffle(unshuffled_routes)
  statistics = tfunc(route_subset$duration, shuffled_routes)
```

```
  return((1+sum(abs(statistics)>=abs(b_data)))/(1+length(statistics)))
}
```

```
route_count = as.data.frame(table(bike_df$route))
usable = unique(as.character(route_count$Var1[route_count$Freq >= 100]))
p_values = rep(0,length(usable))

for (route in usable) {
  route_df = bike_df[bike_df$route == route,]
  b_data = cor(route_df$duration,route_df$days_since_Jan1_2010)
  unshuffled_routes = matrix(rep(route_df$days_since_Jan1_2010,10000), ncol=10000)
  shuffled_routes = colShuffle(unshuffled_routes)
  correlations = cor(route_df$duration, shuffled_routes)
  #p_values = c(p_values, (1+sum(abs(correlations)>=abs(b_data)))/(1+length(correlations)))
  p_values[usable == route] = (1+sum(abs(correlations)>=abs(b_data)))/(1+length(correlations))
}

save(p_values, file = "no_confounder_p_values_100.rda")

sum(p.adjust(p_values,method="holm") < 0.05)
sum(p.adjust(p_values,method="BH") < 0.05)
usable = unique(as.character(route_count$Var1[route_count$Freq > 100]))
p_values_100 = c()
for (route in usable) {
  route_df = bike_df[bike_df$route == route,]
  p_values_100 = c(p_values_100, permutation_test(route_subset = route_df, cor))
}
save(p_values_100, file = "no_confounder_p_values_100.rda")
```

After getting the p-values, we try different multiple testing procedures. If we were doing a real analysis to report significant results, we would select a procedure beforehand, here we try different procedures to see whether they will reject a different number of routes. For Bonferroni and Holm-Bonferroni, we control the FWER at 0.05 and for the BH procedures, we are controlling FDR at 0.05. We try a group adaptive BH procedure which uses the starting station as the groups because it is possible that if there is a change in duration of a route due to construction near a particular starting station, we would see routes with the same start station be affected by the construction. Therefore, we use the group adapative BH method to give more weight to p-values from certain starting stations.

With the Holm Bonferroni or Bonferroni adjustment we do not observe any routes which reached significance threshold but find 366 and 425 routes (respectively) using the false discovery rate control procedures BH and Storey BH. Using group adaptive BH we find 721 routes which we think have significant changes over time. We observe that methods for controlling FWER do not have any rejections. This may be because the methods are too conservative and we are limited by the number of simulations we are able to run. On the other hand, we do get rejections when using BH. Storey-BH and Group Adaptive BH improve on this even further.

```
# Now we plot the routes which the group adaptive BH procedure rejects
route_rejected = usable[group_adaptive_BH_startingstation(p_values, groups, 0.05, 0.5)]
for (route in route_rejected) {
    route_df = bike_df[bike_df$route == route,]
    plot(route_df$days_since_Jan1_2010, route_df$duration,
         xlab="days since Jan1 2020", ylab="duration")
}
```

```
# With Membership Status, Weekday/Weekend , Season as confounders
route_count = as.data.frame(table(bike_df$route))
usable = unique(as.character(route_count$Var1[route_count$Freq > 1000]))
p_values_100_confound = c()
for (route in usable) {
  for (membership_status in c(TRUE,FALSE)) {
    for (weekday in c(TRUE,FALSE)) {
      for (in_season in c("Winter","Spring","Summer","Fall")) {
          route_df = bike_df[bike_df$route == route,]
          p_values_1000_confound = c(p_values_100_confound, permutation_test(route_subset = route_df, ti
      }
    }
  }
}
print(p_values_1000_confound[1:100])
save(p_values_1000_confound, file = "real_confounder_p_values_1000.rda")
confounder_adjustment(p_values, 0.05, 8)
```

```
# With rush hour instead of season as a confounder
p_values = c()

route_count = as.data.frame(table(bike_df$route))
usable = unique(as.character(route_count$Var1[route_count$Freq > 1000]))

for (route in usable) {
  for (membership_status in c(TRUE,FALSE)) {
    for (weekday in c(TRUE,FALSE)) {
      for (rush_hour in c(TRUE,FALSE)) {
        route_df = bike_df[(bike_df$route == route) &
                           (bike_df$member == membership_status) &
                           (bike_df$is_weekday == weekday) &
                           (bike_df$rush_hour == rush_hour),]
        b_data = cor(route_df$duration,route_df$days_since_Jan1_2010)
        unshuffled_routes = matrix(rep(route_df$days_since_Jan1_2010,10000), ncol=10000)
        shuffled_routes = colShuffle(unshuffled_routes)
        correlations = cor(route_df$duration, shuffled_routes)
        p_values = c(p_values, (1+sum(abs(correlations)>=abs(b_data)))/(1+length(correlations)))
      }
    }
  }
}
sum(confounder_adjustment(p_values, 0.05, 8))
save(p_values, file = "rush_hour_real_confounder_p_values_1000.rda")
```

```
# Calculating 1 p-value by permuting within groups but calculating only 1 correlation.
# With rush hour instead of season as a confounder
p_values = c()

route_count = as.data.frame(table(bike_df$route))
usable = unique(as.character(route_count$Var1[route_count$Freq > 1000]))
tester = c("r31007_31009", "r31007_31011")
for (route in usable) {
  # get correlation for route
  route_df = bike_df[bike_df$route == route,]
```

```
  b_data = cor(route_df$duration,route_df$days_since_Jan1_2010)
  shuffled_routes = c()

  for (membership_status in c(TRUE,FALSE)) {
    for (weekday in c(TRUE,FALSE)) {
      for (rush_hour in c(TRUE,FALSE)) {
        route_conf_df = bike_df[(bike_df$route == route) &
                          (bike_df$member == membership_status) &
                          (bike_df$is_weekday == weekday) &
                          (bike_df$rush_hour == rush_hour),]
        unshuffled_routes = matrix(rep(route_conf_df$days_since_Jan1_2010,10000), ncol=10000)
        shuffled_routes = rbind(shuffled_routes, colShuffle(unshuffled_routes))
      }
    }
  }

  correlations = cor(route_df$duration, shuffled_routes)
  p_values = c(p_values, (1+sum(abs(correlations)>=abs(b_data)))/(1+length(correlations)))
}

save(p_values, file = "rush_hour_real_confounder_p_values_1000.rda")


# BH
sum(p.adjust(p_values,method="BH") < 0.05)
```

We got three results from the tests controlling for confounders. First we use permute within each unique combination of membership status, weekday, and season (since those confounders appeared most significant in the initial visualizations). Next we permute within the same groups except substituting season for rush hour (since season was the most computationally expensive to control for since it had the most levels). Since the above procedure didn't really control for FDR we also did a version where we only calculated one correlation per route while controlling for rush hour. Each of these procedures rejected 39, 31, and 35 tests routes as having significant changes over the time period, respectively. For all of these procedures we accounted for multiple comparisons with BH with a threshold of 0.05.

Alternative Permutation Tests

Outside of the methods attempted here there is a whole literature on different methods for controlling for confounders with permutation tests in regressions that we did not have time to implement. One of these is the conditional randomization test proposed by Candes et al (1). If the conditional distribution of the independent variable given the confounders is known (which is often the case with large amounts of observational data) their procedure suggests sampling a new copy of the dataset X values from that distribution. This distribution should be equal to the original distribution under the null. Berrett et al (2) present a further variation on this approach where the permuted independent variables are computed as above but add the constraint that each of the generated vectors of independent variables must be a permutation of the original vector of independent variables.

Frossard et al. (3) also implemented a variety of permutation strategies for linear models in an r package. Their default method involves first fitting a linear regression with only the confounding variables, permuting that models residuals, and adding those residuals to the fitted y values. This procedure is intended to approximate permuting the y values. We would have experimented with applying these methods as well given additional time. One drawback of these linear methods however is that they heavily rely on linear modeling assumptions and are as a rule approximate and with various other drawbacks specific to each particular method

```
times_taken = t(rbind(table(bike_df$route)))
route_counts = data.frame(times_taken)
route_counts$route <- rownames(route_counts)
route_more_than_100 <- route_counts %>% filter(times_taken >= 100)
```

If we choose routes that have more than 100 rides, it will cover 86.7% of the original datapoints and most of the data points get to be keeped with this cutoff.

```
data_by_route = bike_df[bike_df$route %in% route_more_than_100$route, ]
nrow(data_by_route) / nrow(bike_df)
```

```
## [1] 0.867372
```

Linear Regression

Besides permutation test to see whether the correlation signal is significant after permutation, another way to examine the duration over time is through linear regression method. By applying linear regression on each route, we can examine whether the $\hat{\beta}'s$ on days_since_Jan1_2020 are significant or not. If one route has increasing / decreasing trend on its duration over time, the coefficient on days would be significant.

Linear model assumption check

Since we want to use linear regression, we should first check if the linear assumption hold, for our inference on the $\hat{\beta}$ to be valid.

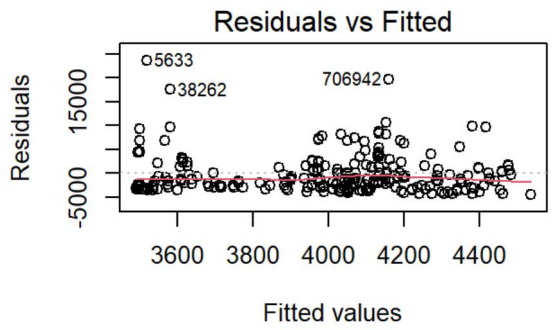
The best way to check linear assumptions is to look at the residual plot and Q-Q plot. However, since there are too many routes and thus too many linear regression models to be checked, let's just look at the graphs of one representative route that has enough rides. We will then use the nvcTest() command to check for non-constant variance and use modified BH method to correct for the multiple testing issue.

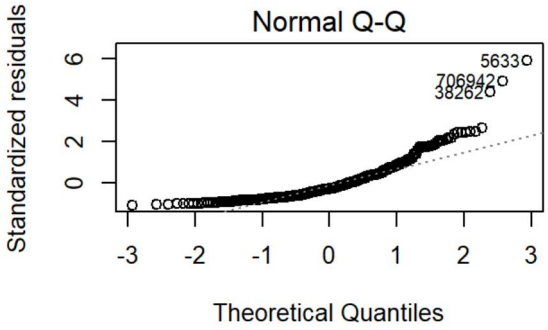
Checking linear assumptions for a single route.

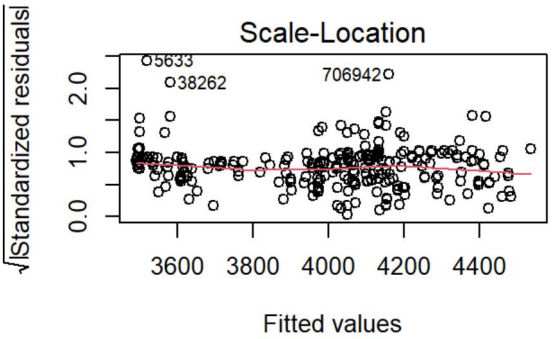
```
data_single_route <- bike_df[bike_df$route == "r31011_31011",]
```

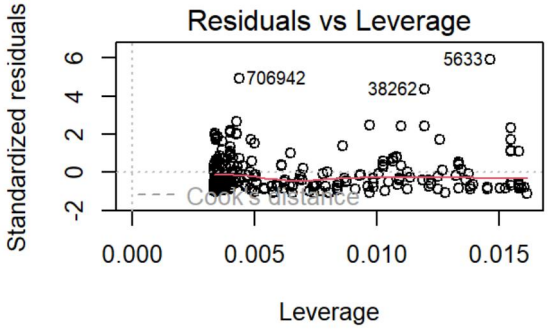
Here we choose the route 31011->31011 which has `nrow(data_single_route)` rides which we considered enough data points for us to check the linear assumptions.

```
fit1 <- lm(duration ~ days_since_Jan1_2010, data_single_route)
par(mfrow=c(2,2))
plot(fit1)
```

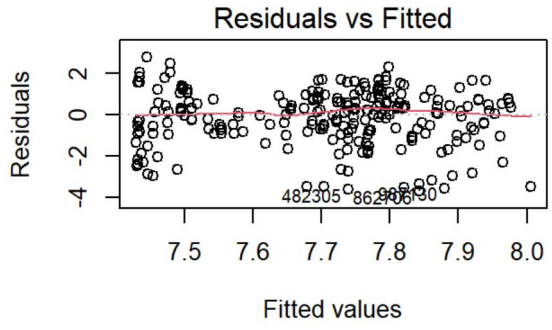


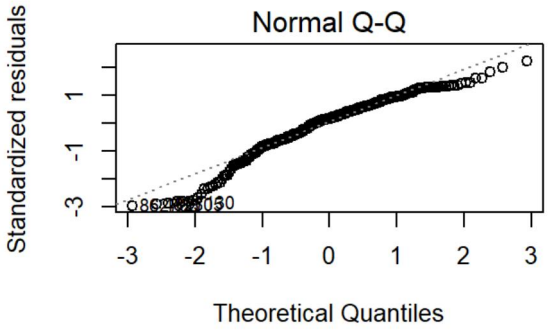


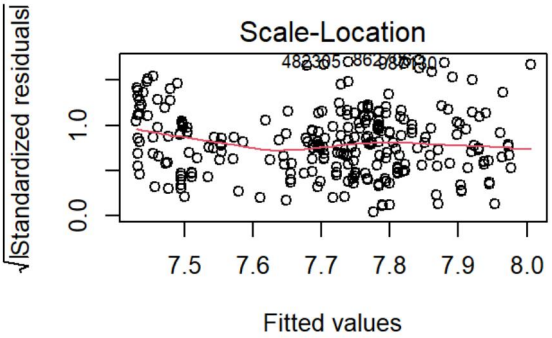


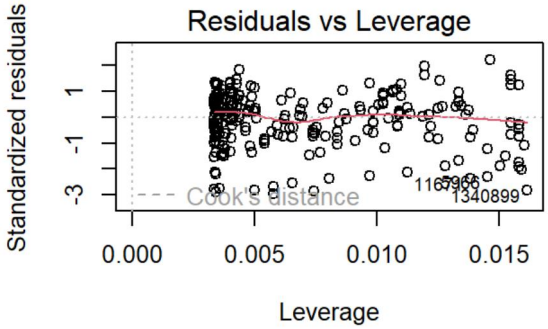



```
fit2 <- lm(log(duration) ~ days_since_Jan1_2010, data_single_route)
par(mfrow=c(2,2))
plot(fit2)
```









From the plot we can see that after log transformation on the response variable, the residual plot looks much nicer and the influence of large leverage points gets undermined as few points stick out on the graph.

Checking linear assumptions for all routes.

Let's next examine all the routes by checking the statistic ncvTest p-value. A significant p value generated by this test means that we should reject the null (the non-constant variance assumption hold).

```
route_list <- route_more_than_100$route
ncv_score <- c()
for (route in route_list){
  data_group <- data_by_route[data_by_route$route == route,]
  fit <- lm(duration ~ days_since_Jan1_2010, data=data_group)
  ncv_score <- append(ncv_score, ncvTest(fit)$p)
}
```

```
length(modified_BH(ncv_score, 0.05, 0.5)) / length(route_list)
```

```
## [1] 0.817402
```

After BH correction, there is still a large proportion significant p values, meaning that there is evidence showing that the non-constant assumption is violated, so we need to make transformations if we want to apply linear regression. Let's try log transformation on the response variable duration.

```
ncv_score1 <- c()
for (route in route_list){
  data_group <- data_by_route[data_by_route$route == route,]
  fit <- lm(log(duration) ~ days_since_Jan1_2010, data=data_group)
  ncv_score1 <- append(ncv_score, ncvTest(fit)$p)
}
```

```
length(modified_BH(ncv_score1, 0.05, 0.5)) / length(route_list)
```

```
## [1] 0.8180147
```

We see that the log transformation slightly improves the the situation but there are still a certain amount of models (55.3%) that do not satisfy the assumption. This means that we might not be able build confidence intervals or other statistical inference on the $\hat{\beta}$. However, since we believe we've controlled for the multi-collinearity inside the model, the significance on the $\hat{\beta}$ shouldn't be too much influenced.

Linear regression not controlling any confounders

```
df_beta <- data.frame(route=rep(NA, length(route_list)),
                      estimate=rep(NA, length(route_list)),
                      p_value=rep(NA, length(route_list)))
i = 1
for (route in route_list){
  data_group <- data_by_route[data_by_route$route == route,]
  fit <- lm(log(duration) ~ days_since_Jan1_2010, data=data_group)
  df_beta[i, 1] <- route
  df_beta[i, c("estimate", "p_value")] <- c(summary(fit)$coefficients[2,1], summary(fit)$coefficients[2,4])
  i = i+1
}
```

```
reject_list <- modified_BH(df_beta$p_value, 0.05, 0.5)
length(reject_list)
```

```
## [1] 1014
```

Since this is also a multiple testing issue as we're trying to look at how many p values are significant among a large amount of test results, we also need to run modified BH to correct for it. For the regression without any confounders, 1015 results got rejected meaning that after multiple testing correction, 1015 routes shows an increase or decrease in duration over time. This is a large amount since we didn't control for other factors that could've affected the duration.

Linear regression controlling for all confounders (weekday, rush hour, season, membership)

```
route_list <- route_more_than_100$route
df_beta_conf <- data.frame(route=rep(NA, length(route_list)),
                      estimate=rep(NA, length(route_list)),
                      p_value=rep(NA, length(route_list)))
i = 1
for (route in route_list){
  data_group <- data_by_route[data_by_route$route == route,]
  fit <- lm(log(duration) ~ days_since_Jan1_2010 +
              is_weekday + rush_hour + season + member, data=data_group)
  df_beta_conf[i, 1] <- route
  df_beta_conf[i, c("estimate", "p_value")] <-
    c(summary(fit)$coefficients[2,1], summary(fit)$coefficients[2,4])
  i = i+1
}
```

Run modified BH for the regression including all the confounders at the same time and 472 got rejected. This means that after multiple testing correction, controlling for other factors that might influence the duration of that route, 472 appeared to show an increasing / decreasing trend in duration over time. This is a lot less than the previous regression without controlling for any confounders.

```
reject_list_conf <- modified_BH(df_beta_conf$p_value, 0.05, 0.5)
length(reject_list_conf)
```

```
## [1] 472
```

There are 389 routes shows significant results taking the intersection of the two.

```
length(intersect(reject_list, reject_list_conf))
```

```
## [1] 389
```

Comparing this number of routes that shows an significant change on duration over time with the result we get from permutation tests, we get to see that 366 routes out of the routes with more than 100 trips. This is reassuring since both types of tests resulted in similar numbers of rejected routes. The linear method resulted in greater numbers of rejected routes likely due to the linear assumptions.

Permutation II: comparing difference in mean monthly duration controlling for month and membership

As advised in office hours, we also conducted the permutation test using a different test statistic. In the analysis below we examine the change in mean duration for each route controlling for confounding effects of seasonality and membership. We control for season by comparing the difference in duration only between the same month each year, i.e., November 2010 to November 2011. We control for confounding by membership by running separate permutations for trips taken by members and nonmembers. Additionally we validate our findings by performing data splitting: we independently perform the analysis first for November 2010 and 2011 and then for Oct 2010 and 2011 (i.e., on new data), seeking routes that show a statistically significant change in duration in both pairwise comparisons.

Permutation on November data controlling for membership

We screen the routes that have not been taken enough times. We want to throw away the data points from which it is not possible to make any discoveries and by doing so reduce the extent of our multiple testing adjustments later in the analysis. A rough calculation tells us that the number of rides for the permutation test to have a minimum power is 10-20, and so we do not include any routes that have been taken a fewer number of times in our analysis. To achieve minimum acceptable power, we keep only the routes that have at least 10 rides in each comparison group. E.g., for permutation on rides taken by members, each route selected needed to have been taken at least ten times by members in November 2010 and 10 times in November 2011 for a minimum of 20 trips per route per condition. This is not a data dependent choice, as we base our choice based on the number of observations and not on the observed values of the variables. This procedure yielded 1053 eligible routes taken by members and only 100 routes taken by nonmembers. We perform the analysis on routes taken in November and October, as those are the months with the highest number of trips taken.

We then conduct permutation tests with 500 permutations. For both member trips and nonmember trips we have two groups of observations and we want to test whether the mean of one group is greater than the other, we can compute the difference in means between the two groups, and then randomly shuffle the observations between the two groups many times. Each time we shuffle the observations, we compute the difference in means. The distribution of these differences in means, under the assumption that the two groups are the same, is called the permutation distribution. We then compare the original difference in means, computed from the original data, to the permutation distribution. For trips where the original difference in means is more extreme than what we would expect to see if the two groups were the same, we reject the null hypothesis that the two groups are the same, and conclude that there is a significant difference between the two groups.

Because we run multiple permutation tests, we run into a multiple testing problem, which we deal with using the modified Benjamini-Hochberg (BH) procedure. The basic idea of the BH procedure is to adjust the p-value threshold used for declaring significance based on the number of tests being performed and the desired FDR level. The FDR is the expected proportion of false positive results among all positive results. We conducted BH at the FDR level of 0.1.

This procedure yields two lists of p-values, one for trips taken by members and one by nonmembers. Because we are looking for routes which have changed, ideally we would expect the travel duration to change for both members and nonmembers, and so we would report only the routes that show statistically significant change for both members and nonmembers. In our case, however, there are few routes where enough rides where taken by nonmembers and so in most cases we didn't have enough data points to check a route for both members and nonmembers, so we report all three results.

Among the routes analyzed with data from members, there are 99 routes where duration changed between Nov 2010 and Nov 2011. For the nonmembers, there are 11 such routes. There is one route that show change for bot members and nonmembers.

```r
permutation.test <- function(treatment, outcome, n){
  distribution=c()
  result=0
  for(i in 1:n){
    distribution[i]=diff(by(outcome, sample(treatment, length(treatment), FALSE), mean))
  }
  result=sum(abs(distribution) >= abs(original))/(n)
  return(list(result, distribution))
}
```

```r
route_pvals_500_members <- tibble(route = character(), p_val = numeric())

for (route_no in unique(bike_tibble_members$route)){

  test_point <- bike_tibble %>% filter(route==route_no)
  treatment <- test_point$Year
  outcome <- test_point$duration
  original <- diff(tapply(outcome, treatment, mean))
  test1 <- permutation.test(treatment, outcome, 500)
  route_pvals_500_members <- route_pvals_500_members %>% add_row(route = route_no, p_val = test1[[1]])
}

# Apply BH correction using p.adjust
p_adjusted_members <- p.adjust(route_pvals_500_members$p_val, method = "BH")

# Find significant results based on adjusted p-values
significant_results_members <- which(p_adjusted_members < 0.1)

route_pvals_500_nonmembers <- tibble(route = character(), p_val = numeric())
i= 0

for (route_no in unique(bike_tibble_nonmembers$route)){

  test_point <- bike_tibble %>% filter(route==route_no)
  treatment <- test_point$Year
  outcome <- test_point$duration
  original <- diff(tapply(outcome, treatment, mean))
  test1 <- permutation.test(treatment, outcome, 500)
  route_pvals_500_nonmembers <- route_pvals_500_nonmembers %>% add_row(route = route_no, p_val = test1[
}

# Apply BH correction using p.adjust
p_adjusted_nonmembers <- p.adjust(route_pvals_500_nonmembers$p_val, method = "BH")

# Find significant results based on adjusted p-values
significant_results_nonmembers <- which(p_adjusted_nonmembers < 0.1)
```

```r
changed_routes_members_Nov <- unique(bike_tibble_members$route)[significant_results_members]

changed_routes_nonmembers_Nov <- unique(bike_tibble_nonmembers$route)[significant_results_nonmembers]

changed_routes_both_Nov <- unique(bike_tibble_nonmembers$route)[significant_results_nonmembers][unique(
```

Validation with previously unused, novel data for October 2010 and 2011

We then perform below the same analysis for a different month, October. These are novel, independent data that were not used before and so I can uses it to check if I am overfitting. I take advantage of it to check if the same routes will turn out to have a significant change of duration between years (a change in route should affect the trip duration independent of the season). Indeed out of the 99 routes significant for members when comparing between Nov 2010 and 2011, 19 are also significant for October. For the 11 routes significant on Nov data for nonmembers, 3 are also significant for the November data. However the single route significant for the Nov data for both members and nonmembers is not significant for October.

```
route_pvals_500_members <- tibble(route = character(), p_val = numeric())

for (route_no in unique(bike_tibble_members$route)){

  test_point <- bike_tibble %>% filter(route==route_no)
  treatment <- test_point$Year
  outcome <- test_point$duration
  original <- diff(tapply(outcome, treatment, mean))
  test1 <- permutation.test(treatment, outcome, 500)
  route_pvals_500_members <- route_pvals_500_members %>% add_row(route = route_no, p_val = test1[[1]])
}


# Apply BH correction using p.adjust
p_adjusted_members <- p.adjust(route_pvals_500_members$p_val, method = "BH")

# Find significant results based on adjusted p-values
significant_results_members <- which(p_adjusted_members < 0.1)

route_pvals_500_nonmembers <- tibble(route = character(), p_val = numeric())

for (route_no in unique(bike_tibble_nonmembers$route)){

  test_point <- bike_tibble %>% filter(route==route_no)
  treatment <- test_point$Year
  outcome <- test_point$duration
  original <- diff(tapply(outcome, treatment, mean))
  test1 <- permutation.test(treatment, outcome, 500)
  route_pvals_500_nonmembers <- route_pvals_500_nonmembers %>% add_row(route = route_no, p_val = test1[
}

# Apply BH correction using p.adjust
p_adjusted_nonmembers <- p.adjust(route_pvals_500_nonmembers$p_val, method = "BH")

# Find significant results based on adjusted p-values
significant_results_nonmembers <- which(p_adjusted_nonmembers < 0.1)

changed_routes_members_Oct <- unique(bike_tibble_members$route)[significant_results_members]

changed_routes_nonmembers_Oct <- unique(bike_tibble_nonmembers$route)[significant_results_nonmembers]

changd_routes_both_Oct <- unique(bike_tibble_nonmembers$route)[significant_results_nonmembers][unique(bi
```

```r
print("Count of intersection of significant routes for member data controlling for month:")
```

```
## [1] "Count of intersection of significant routes for member data controlling for month:"
```

```r
sum(changed_routes_members_Nov %in% changed_routes_members_Oct) # intersection of significant routes fo
```

```
## [1] 19
```

```r
print("Count of intersection of significant routes for nonmember data controlling for month:")
```

```
## [1] "Count of intersection of significant routes for nonmember data controlling for month:"
```

```r
sum(changed_routes_nonmembers_Nov %in% changed_routes_nonmembers_Oct) # intersection of significant rou
```

```
## [1] 3
```

```r
print("Count of intersection of route that were significant for both members and nonmembers for both Nov
```

```
## [1] "Count of intersection of route that were significant for both members and nonmembers for both No
```

```r
sum(changed_routes_both_Nov %in% changd_routes_both_Oct)
```

```
## [1] 0
```

Conclusion

Overall, we first visualized the data to both gain a deeper understanding of the distribution of routes and trips and to test which confounders might be valuable to control for. We then conducted permutation tests with correlation within each route both with and without confounders. For routes with over 100 trips (without adjusting for confounders) we tried several different multiple comparison adjustment techniques for the p-values. Using the Holm Bonferroni or Bonferroni adjustment we did not observe any routes which reached significance threshold but found 366, 425, and 721 routes (respectively) using the false discovery rate control procedures BH and Storey BH and using group adaptive BH. For routes with over 1000 trips we saw 39, 31, and 35 routes with changes in duration from 2010-2011 using correlations and different sets of confounders. We tried running linear regressions within each route not accounting for confounders and identified 1015 routes with significant changes over time after modified BH adjustment. We also tried running linear regressions for each route controlling for weekday, rush hours, season, and whether the biker was a member. Using this procedure resulted in 472 routes identified as having significant changes over the time period. Lastly we compared difference in mean monthly duration controlling for month and membership which resulted in a total of 109 routes with changes over the time period. Overall the number as well as the specific routes which are considered to have changes in trip duration over the time period seem to vary significantly depending on the specific method used for testing and the specific group of confounders chosen to control for.

References

1. Rina Foygel Barber and Emmanuel Cand‘es. On the construction of knockoffs in case–control studies. Stat, 8(1):e225, 2019.
2. Berrett, T.B., Wang, Y., Barber, R.F., & Samworth, R.J. (2018). The conditional permutation test for independence while controlling for confounders. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 82.
3. Frossard, J., & Renaud, O. (2021). Permutation Tests for Regression, ANOVA, and Comparison of Signals: The permuco Package. Journal of Statistical Software, 99(15), 1–32. https://doi.org/10.18637/jss.v099.i15

1 Project 1 80 / 100

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

Grading criteria:

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

* The real world meaning of the data is 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, are addressed thoughtfully

* The analysis shows 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

+ 40 pts Click here to replace this description.

+ 39 pts Click here to replace this description.

+ 38 pts Click here to replace this description.

+ 37 pts Click here to replace this description.

+ 36 pts Click here to replace this description.

+ 35 pts Click here to replace this description.

+ 34 pts Click here to replace this description.

✓ **+ 33 pts** *Click here to replace this description.*

+ 32 pts Click here to replace this description.

+ 31 pts Click here to replace this description.

+ 30 pts Click here to replace this description.

+ 29 pts Click here to replace this description.

+ 28 pts Click here to replace this description.

+ 27 pts Click here to replace this description.

+ 26 pts Click here to replace this description.

+ 25 pts Click here to replace this description.

+ 24 pts Click here to replace this description.

+ 23 pts Click here to replace this description.

+ 22 pts Click here to replace this description.

+ 21 pts Click here to replace this description.

+ 20 pts Click here to replace this description.

Statistical methodology (30pts)

Grading criteria:

* The analysis identifies all the major relevant challenges, such as issues of multiple testing, non-independence, confounding, exploratory data analysis, etc
* These challenges are handled appropriately in the analysis, applying existing tools or developing new techniques
* Assumptions are tested and examined using, e.g., using diagnostic plots
* Any remaining issues that cannot be addressed, are discussed

+ 30 pts Click here to replace this description.

+ 29 pts Click here to replace this description.

+ 28 pts Click here to replace this description.

+ 27 pts Click here to replace this description.

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

+ 25 pts Click here to replace this description.

+ 24 pts Click here to replace this description.

+ 23 pts Click here to replace this description.

+ 22 pts Click here to replace this description.

+ 21 pts Click here to replace this description.

+ 20 pts Click here to replace this description.

+ 19 pts Click here to replace this description.

+ 18 pts Click here to replace this description.

+ 17 pts Click here to replace this description.

+ 16 pts Click here to replace this description.

+ 15 pts Click here to replace this description.

Report & code (30pts)

Grading criteria:

* The report is 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, are addressed as needed
* The code is clear, well organized, and appears readable and reproducible

* Sufficient details are given to understand the specifics of the analysis being run and the choices made along the way

+ 30 pts Click here to replace this description.

+ 29 pts Click here to replace this description.

+ 28 pts Click here to replace this description.

+ 27 pts Click here to replace this description.

+ 26 pts Click here to replace this description.

✓ **+ 25 pts** *Click here to replace this description.*

+ 24 pts Click here to replace this description.

+ 23 pts Click here to replace this description.

+ 22 pts Click here to replace this description.

+ 21 pts Click here to replace this description.

+ 20 pts Click here to replace this description.

+ 19 pts Click here to replace this description.

+ 18 pts Click here to replace this description.

+ 17 pts Click here to replace this description.

+ 16 pts Click here to replace this description.

+ 15 pts Click here to replace this description.

- 4 *Point adjustment*

Nice work on your project! Below are comments for each of the 3 sections of the rubric. (The points adjustment is due to the late submission.)

1. Your project proposes a task-specific permutation test procedure which controls confounders, and then applies group adaptive/modified BH to figure out the routes which duration changes over time. You also consider a linear regression approach, and a different permutation approach using monthly data.

Using group adaptive BH to group by starting station is a very nice idea -- it definitely makes sense that factors like construction etc may be grouped together. (To extend this we might consider geographical proximity more generally, or shared roads along the route, rather than shared starting station.)

The different approaches taken in the project are all interesting but are not sufficiently tied together -- how do they compare to each other / build on each other / make same or different assumptions / etc?

A few additional things:
* (Page 4) We should not consider Year as a confounder because this is exactly the question we want to test.
* Since we know that there will likely be issues of confounding, it may not be useful to run models/methods with no confounders included before adding them in -- or if we do, we can state more clearly that this is just a baseline and we don't expect it to be valid.

2. Overall the project shows good statistical methodology with attention to issues of multiple testing, etc. A few things that could be improved:
* While it's reasonable to avoid including too many confounders, looking for a marginal association between a potential confounder X_j and the response Y doesn't definitively answer the question of whether X_j plays a role in the larger model
* It's great to look at diagnostic plots for linear regression but it would be better to at least look at several routes since any single route may not be representative of the others.
* The diagnostic plots are shown for the regression of the response onto only the covariate of interest, but this is not the relevant model -- the relevant model is the one that also includes confounders and it would be important to consider diagnostics here.
* There is some inconsistency across the different parts of the report in terms of which confounders are included or not.

3. Strengths of the report:
* Nice descriptions and visualizations of the data during the exploration at the beginning of the report
* Good justification of many of the decisions made throughout the permutation analysis.

In some places, the report was a bit hard to follow and some more organization, and perhaps more precise notation, would be helpful. For example the long paragraph on page 8 (which is labeled as page 2) is hard to follow. It looks like multiple reports were merged -- the formats are not very consistent, and more organization/narrative across these different pieces to tie them together would improve the report.