regression_matrix

Raw data

Divvy provides two types of raw data – station-level and trip-level. The folder titled "Sample raw data" folder collects two csv files exemplifying station-level and trip-level data. All raw data is collected in the folder "Raw data (2013-2018)".

- Divvy_Stations_2015.csv - station-level information in year of 2015
  - station id
  - station names
  - latitude/longitude of the station
  - dpcapacity: # of total docks at each station
  - online_data: date the station went live in the system
- Divvy_Trips_2015_09.csv - all trip-level info
  - trip info: trip id, start and stop time (day and time trip started, in CST), duration (in seconds), origin station id, destination id, and their names
  - bike id: ID attached to each bike
  - user type ("Customer" is a rider who purchased a 24-Hour Pass; "Subscriber" is a rider who purchased an Annual Membership), gender and birth year if subscriber

Note: in Jan. 24th 2020, Divvy reorganized its public data and remove all station-level information after 2014. (https://divvy-tripdata.s3.amazonaws.com/index.html).

Preparing Zipcode-monthly Level Datasets

The first objective is to conduct analysis at the zipcode-monthly level. To achieve this objective, we have done the following data preparation tasks:

- I. Based on "Divvy_Stations_2015.csv", we prepared zipcode-level station information (e.g., dpcapacity, expansion events, new stations added, etc.). The final outputs are "zipcode_dpcapacity" and "zipcode_stations". In addition, csv files are generated alongside.
- II. Collecting trip and demographic information at zipcode level and the final output file is "Zipcode_level.all_V3.csv"
  - The data cleaning procedure is documented in a separate file.
- III. Merging the above two data sources and preparing the final data for regressions.

I. Zipcode-level station information

Add zipcodes to "Divvy_Stations_2015.csv"

Based on the location information (i.e.,latitude and longitude), Googlemap Api is used to obtain the zip code for each station and stored in the csv file titled "station_zipcode_v1.csv". In this file, we have 581 stations in total. Note Google API is not 100 percent correct. At our best, we cross-validate and manually check the accuracy. We store the corrected version in "station_zipcode_v2.csv".

The code is as follows:

```
divvy_data <- as.data.frame(read_csv("Divvy_Stations_2015.csv"))

# use ggmap to obtian the addresses from the google maps API
coordinates <- cbind(divvy_data$longitude, divvy_data$latitude)
divvy_file <- "DivvyAddresses.csv"

# To perform the code, the register key should be applied:
register_google(key =  "AIzaSyD6l6lidbeKJZsct6zLGy8RzTJperDA3Hc", write = TRUE)

if(!divvy_file %in% list.files(getwd())){
  address<- do.call(rbind, lapply(1:nrow(coordinates),
                                  function(i) revgeocode(coordinates[i,])))
}
divvy <- cbind(divvy_data, address)

divvy_df <- divvy %>%
  mutate(zip_code = str_trim(str_extract(divvy$address, "[\\d]{5}")))%>%
  select(id, name, dpcapacity, online_date,zip_code)
write.csv(divvy_df, file = "station_zipcode_v1.csv")
#In this file, we have all the station with their corresponding zip codes.
#The resulting file has nrow=581
```

Since we focus on 2015 expansion at the moment, we only keep stations with online datas before 2015-12-31 and generate a new CSV file called "station_zipcode_2015_final.csv". In total, we have 475 stations. The data frame is named "df" for further cleaning.

```
divvy_df <-  as.data.frame(read_csv("station_zipcode_v2.csv"))
#changed the online_date into date format
divvy_df$online_date<-as.Date(divvy_df$online_date, "%m/%d/%Y")
#We selected the stations whose online date is in 2015.
#The resulting data frame "df" has nrow=475.
df<-filter(divvy_df,online_date<"2016-01-01")
write.csv(df,"station_zipcode_2015_final.csv")
```

Here are the histograms and summary stats of stations by zip code (min,max, medium, avg.).

```
#This line count the number of station for each zip code
df %>% group_by(zip_code) %>% summarise(count_station=n())
```

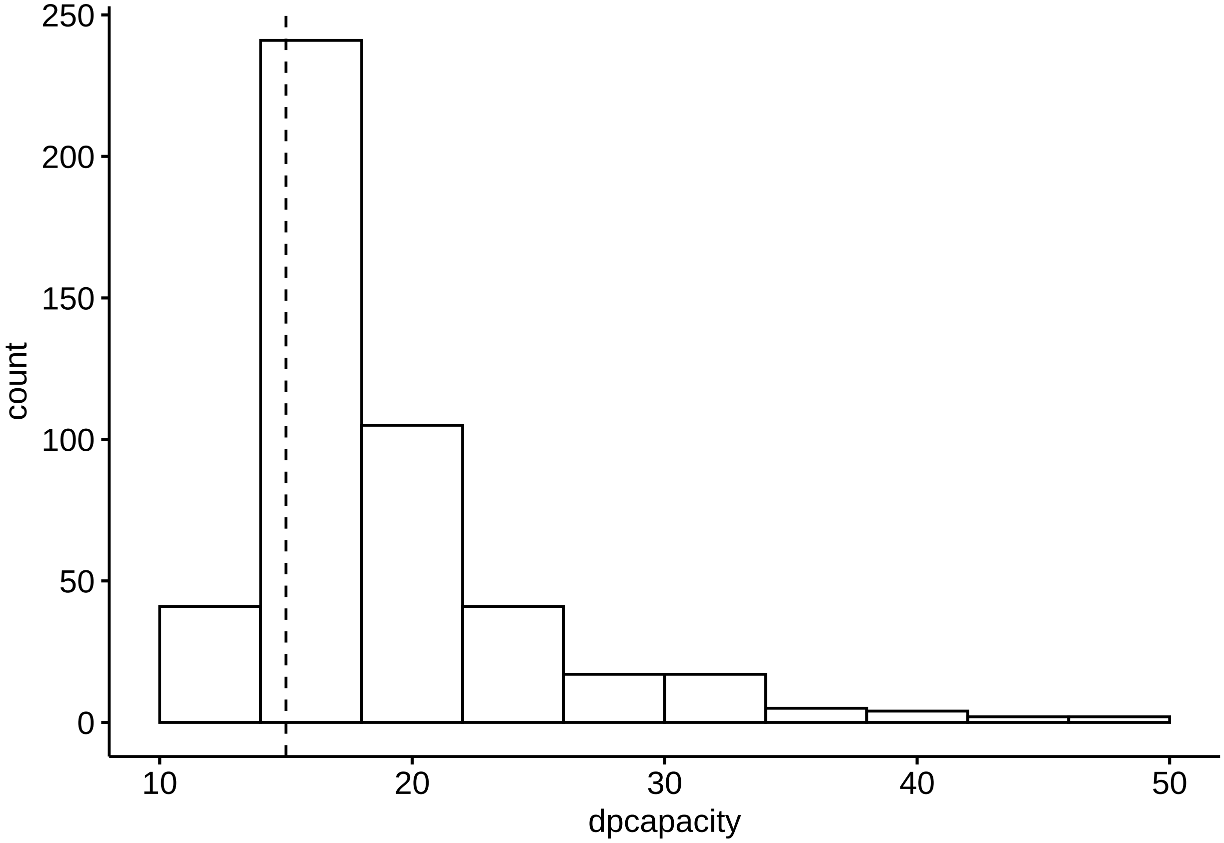
```
## # A tibble: 40 x 2
##    zip_code count_station
##       <dbl>         <int>
##  1    60601             8
##  2    60602             6
##  3    60603             3
##  4    60604             4
##  5    60605            13
##  6    60606             5
##  7    60607            22
##  8    60608            26
##  9    60609            11
## 10    60610            15
## # ... with 30 more rows
```

```
summary(df[,"dpcapacity"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   15.00   15.00   17.92   19.00   47.00
```

```
gghistogram(df, x = "dpcapacity", bins = 10,add = "median")
```

```
## Warning in (function (mapping = NULL, data = NULL, ..., xintercept, na.rm = FALSE, : Using both `xin
```

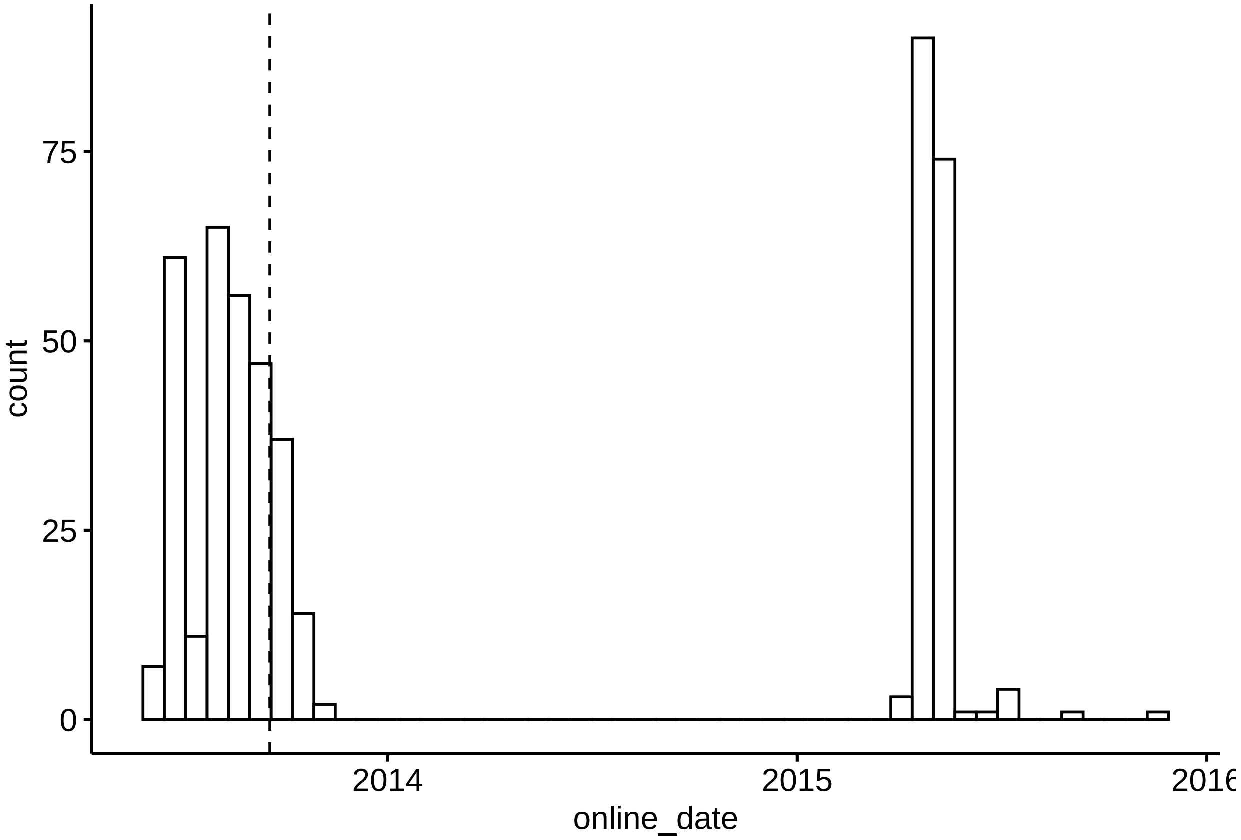


```
summary(df[,"online_date"])
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2013-06-13" "2013-08-02" "2013-09-18" "2014-04-01" "2015-04-24" "2015-11-25"
```

```
gghistogram(df, x = "online_date", bins = 48,add = "median")
```

```
## Warning in (function (mapping = NULL, data = NULL, ..., xintercept, na.rm = FALSE, : Using both `xin
```



```
# from this graph, we will see the online date on a time line
```

Code dummies to denote expansions in 2015

Event backgrount: Divvy underwent two expansions in 2015 and 2016. According to Divvy's public statement, the first expansion started from April 2015. 93 stations added to existing communities whereas 82 stations added to 16 new communities including lower-income communities of Englewood (4), Humboldt Park (4) and North Lawndale (4).

According to the variable "online_data", Divvy's 2015 expansion concentrated in April and May with few exceptions (first online date in the system was July or Septemnber 2015). Hence, we code four dummies for each station:

- new_April: whether the station was online between 2015-04-01 and 2015-04-30
- new_May: whether the station was online between 2015-05-01 and 2015-05-31
- before_expansions: whether the station was online before 2015-04-01
- after_expansionsnone: whether the station was online after 2015-05-31

The code is shown here:

```
# Add 4 expansion dummies:
expansion_dummy <- data.frame( new_April = df$online_date>="2015-04-01"&
                                 df$online_date <="2015-04-30",
                    new_May =df$online_date>="2015-05-01" & df$online_date<="2015-05-31",
                    before_expansions = df$online_date<"2015-04-01",
                    after_expansions = df$online_date>"2015-05-31"
                   )
```

```
df <-cbind(expansion_dummy,df)
```

Divvy's second expansion (started from June 2016) further targeted lower-income communities: Austin (14), West Englewood (6), West Garfield Park (5). But at this moment, we only focus on the 2015 expansion.

Aggregate dpcapacity before and after April/May expansions

In "df", 300 out of 475 stations exist before 2015 expansion. These 300 stations belong to 28 zip codes. For these 300 stations, we sum up station-level dpcapacity within one zip code (out of 28 zip codes). Then we reach zipcode-level dpcapacity before 2015 expansion, which is labeled as "d1".

```
#sum up the dpcapacity for each time period
#section1: capcacity before any expansions
section1 <- df[ which(df$before_expansions == 'TRUE'),]
dim(section1)
```

```
## [1] 300   9
```

```
d1<- section1 %>%
  group_by(zip_code) %>%
  summarise(dpcapacity1=sum(dpcapacity))
dim(d1)
```

```
## [1] 28  2
```

```
head(d1)
```

```
## # A tibble: 6 x 2
##   zip_code dpcapacity1
##      <dbl>       <dbl>
## 1    60601         139
## 2    60602         174
## 3    60603          81
## 4    60604          81
## 5    60605         235
## 6    60606         151
```

April 2015 expansion added 86 new stations. Six new zip codes have stations through April expansion. These six new zip codes all belong to low-income zip codes (sorted by poverty level - percentage population below poverty level in the U.S.): 60621(60.68%), 60624(54.54%), 60623(45.39%), 60651 (38.82%), 60649 (34.35%), 60619(22.64%). Source: http://zipatlas.com/us/il/chicago/zip-code-comparison/population-below-poverty-level.htm

We replicate the previous procedure to get zipcode-level dpcapacity by April 2015 expansion for 34 zip codes. Note if a zip code had no new stations by April, its dpcapacity remains unchanged.

```
#section2: capacity with newly added stations in April expansion
section2 <- df[ which(df$before_expansions == 'TRUE'| df$new_April=='TRUE'),]
dim(section2)
```

```
## [1] 386   9
```

```
d2<- section2 %>%
  group_by(zip_code) %>%
  summarise(dpcapacity2=sum(dpcapacity))
dim(d2)
```

```
## [1] 34  2
```

```r
head(d2)
```

```
## # A tibble: 6 x 2
##   zip_code dpcapacity2
##      <dbl>       <dbl>
## 1    60601         216
## 2    60602         174
## 3    60603          81
## 4    60604          96
## 5    60605         285
## 6    60606         151
```

May expansion added 82 new stations. 6 new zip codes had stations in May. The six new zip codes are 60626 (26.12%), 60660 (19.56%), 60659 (17.71%), 60645 (16.78%), 60641 (12.43%), 60630 (7.31%). The percentage in the bracket presents the poverty level.

We replicate the previous procedure to get zipcode-level dpcapacity by May 2015 expansion for 40 zip codes.

```r
#section3: capacity with newly-added stations in the May expansion
section3 <- df[ which(df$before_expansions == 'TRUE'| df$new_April=='TRUE'|
                        df$new_May == 'TRUE'),]
dim(section3)
```

```
## [1] 468   9
```

```r
d3<- section3 %>%
  group_by(zip_code) %>%
  summarise(dpcapacity3=sum(dpcapacity))
dim(d3)
```

```
## [1] 40  2
```

7 stations added to existing zip codes.

```r
#section4: new stations whose online date is after the May expansion
section4 <- df[ which(df$before_expansions == 'TRUE'| df$new_April=='TRUE'|
                        df$new_May == 'TRUE' | df$after_expansions == 'TRUE'),]
dim(section4)
```

```
## [1] 475   9
```

```r
d4<- section4 %>%
  group_by(zip_code) %>%
  summarise(dpcapacity4=sum(dpcapacity))
dim(d4)
```

```
## [1] 40  2
```

Lastly, we put dpcapacity1~4 in one table and replace NA with 0 stations/capacity. "zipcode_dpcapacity.csv" is generated alongside.

```r
t1<-merge(d1,d2,by="zip_code",all=TRUE)
t2<-merge(t1,d3,by="zip_code",all=TRUE)
t3<-merge(t2,d4,by="zip_code",all=TRUE)
t3[is.na(t3)] <- 0
zipcode_dpcapacity <- t3
dim(zipcode_dpcapacity)
```

```
## [1] 40  5
```

```r
head(zipcode_dpcapacity,2)
```

```
##   zip_code dpcapacity1 dpcapacity2 dpcapacity3 dpcapacity4
## 1    60601         139         216         216         216
## 2    60602         174         174         174         174
```

```r
#write the output into file  "zipcode_dpcapacity.csv"
write.csv(zipcode_dpcapacity, "zipcode_dpcapacity.csv")
```

Aggregate the number of added/total stations before and after April/May expansions and generate zipcode-level expansion dummies

```r
# new and total stations before/after April/May expansions
zipcode_stations <- df%>%group_by(zip_code)%>%
  summarise(existing_before_expansions_sumStation1=  sum(before_expansions),
            new_station_April=  sum(new_April),
            new_station_May=  sum(new_May),
            new_station_after_expansions=  sum(after_expansions),
            total_station_by_April = sum(before_expansions)+sum(new_April),
            total_station_by_May=sum(before_expansions)+ sum(new_April)+ sum(new_May),
            total_station_after_expansions_sumStation2=sum(before_expansions)+
              sum(new_April)+sum(new_May)+sum(after_expansions))
head(zipcode_stations,2)
```

```
## # A tibble: 2 x 8
##   zip_code existing_before~ new_station_Apr~ new_station_May new_station_aft~
##      <dbl>            <int>            <int>           <int>            <int>
## 1    60601                5                3               0                0
## 2    60602                6                0               0                0
## # ... with 3 more variables: total_station_by_April <int>,
## #   total_station_by_May <int>,
## #   total_station_after_expansions_sumStation2 <int>
```

We code three dummies for each zip code:

- expApr: whether the zip code has new stations online between 2015-04-01 and 2015-04-30
- expMay: whether the zip code has new stations online between 2015-05-01 and 2015-05-31
- expboth: whether the zip code has new stations online during both expansions

The code is shown here:

```r
# Add additional three zipcode-level expansion dummies
zipcode_stations$expApr <- as.numeric(zipcode_stations$new_station_April>0)
zipcode_stations$expMay <- as.numeric(zipcode_stations$new_station_May>0)
zipcode_stations$expboth <- as.numeric(zipcode_stations$new_station_April>0 &
                                               zipcode_stations$new_station_May>0)
dim(zipcode_stations)
```

```
## [1] 40 11
```

```r
head(zipcode_stations,2)
```

```
## # A tibble: 2 x 11
##   zip_code existing_before~ new_station_Apr~ new_station_May new_station_aft~
##      <dbl>            <int>            <int>           <int>            <int>
## 1    60601                5                3               0                0
## 2    60602                6                0               0                0
## # ... with 6 more variables: total_station_by_April <int>,
## #   total_station_by_May <int>,
## #   total_station_after_expansions_sumStation2 <int>, expApr <dbl>,
## #   expMay <dbl>, expboth <dbl>
```

```
# create "zipcode_stations.csv".
write.csv(zipcode_stations,"zipcode_stations.csv")
```

In the code below, we list the zip codes for each expansion.

```
#The following code gets the zip code which has a expansion in April
# 60601 60604 60605 60608 60609 60612 60615 60616 60619 60621
# 60622 60623 60624 60637 60647 60649 60651 60653 60654 60661

zipcode_stations[which(zipcode_stations$expApr == 1),] %>% select("zip_code")
```

```
## # A tibble: 20 x 1
##    zip_code
##       <dbl>
##  1    60601
##  2    60604
##  3    60605
##  4    60608
##  5    60609
##  6    60612
##  7    60615
##  8    60616
##  9    60619
## 10    60621
## 11    60622
## 12    60623
## 13    60624
## 14    60637
## 15    60647
## 16    60649
## 17    60651
## 18    60653
## 19    60654
## 20    60661
```

```
#The following code gets the zip code which has a expansion in May
zipcode_stations[which(zipcode_stations$expMay == 1),] %>% select("zip_code")
```

```
## # A tibble: 22 x 1
##    zip_code
##       <dbl>
##  1    60605
##  2    60607
##  3    60608
##  4    60610
##  5    60611
##  6    60612
##  7    60614
##  8    60616
##  9    60618
## 10    60621
## # ... with 12 more rows
```

```
# 60605 60608 60610 60611 60612 60614 60616 60618 60621 60625
# 60626 60630   60637   60640   60641   60642   60645   60647   60649   60659
# 60660

#The following code gets the zip code which has both expansion in April and May
# 60605 60608   60612   60616   60621   60637   60647 60649
zipcode_stations[which(zipcode_stations$expboth == 1),] %>% select("zip_code")
```

```
## # A tibble: 8 x 1
##   zip_code
##      <dbl>
## 1    60605
## 2    60608
## 3    60612
## 4    60616
## 5    60621
## 6    60637
## 7    60647
## 8    60649
```

```
#The following code gets the zipcode which does not have any expansion in April or May
# 60602 60603 60606 60607 60613 60657 60699
zipcode_stations[which(zipcode_stations$expApr == 0 &
                         zipcode_stations$expMay == 0),] %>% select("zip_code")
```

```
## # A tibble: 6 x 1
##   zip_code
##      <dbl>
## 1    60602
## 2    60603
## 3    60606
## 4    60613
## 5    60657
## 6    60699
```

From the above analysis, we know that 20 zip codes have new stations in April, 21 zip codes having new stations in May, 8 of them having new stations in both April and May, and 7 zip codes not having new stations during 2015 expansions.

II. Zipcode-level trip and demographics information

"Zipcode_level_all_V3.csv" contains the following categories of information at the zipcode level:

- weekly, monthly and quarterly trip volumes (e.g., incoming, outgoing, usage by members, usage by females, etc.)
- Bike score, walk score and transit score. These scores were queried by an API script (also in the folder) at a third-party website to measure the goodness of local infrastructure for biking, walking and transit.
- Demographics, including the distributions of race, gender, population, income, education, marriage status, surveyed counts of transportation modes by age group.

The folder "info about Zipcode_level_all_V3" provides a list of attributes labels ("attributes_description.xlsx") and data sources.

III. Merging the above two sources with additional information and creating the final zipcode-monthly data for regression

Interim step: create an auxiliary csv table "information.xlsx"

The goal of this file is to gather variables from different files on which our regression matrix based.

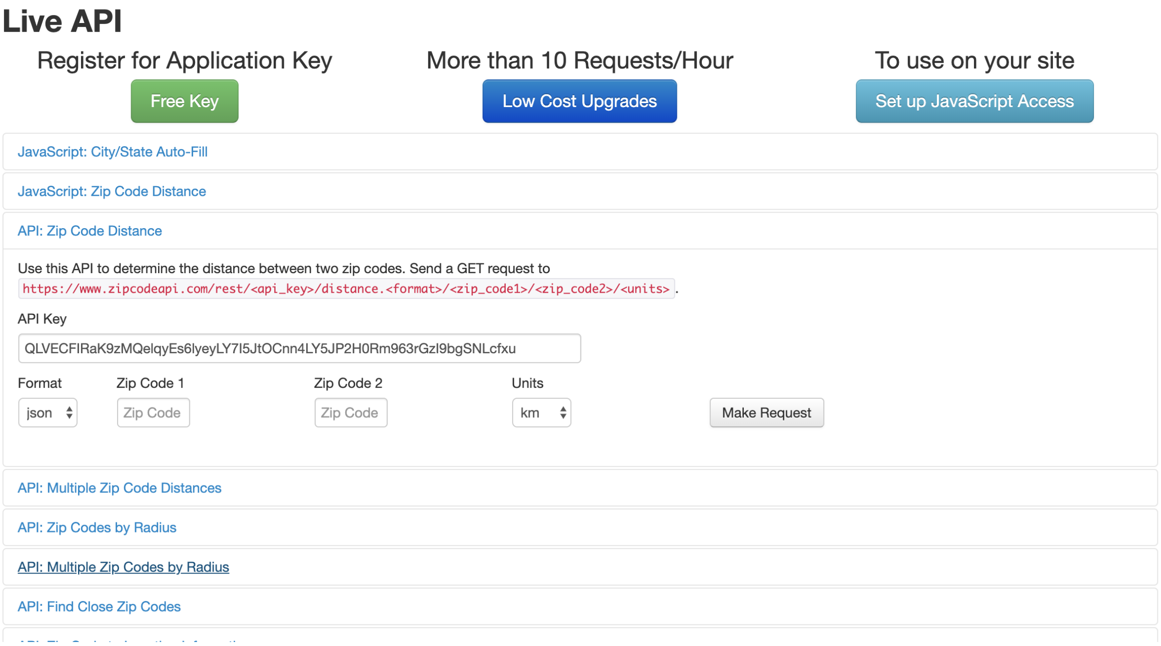
In the "information.xlsx", we include the following three categories of variables:

1. Merge zipcode_stations and zipcode_dpcapacity

Selecting existing_before_expansions, total_station_by_April, total_station_after_expansions, expApr, expMay from zipcode_stations and store them as sum_Station_1, total_station_by_April, sum_Station_2, expApr and expMay respectively in information.xlsx. Selecting dpcapacity1, dpcapacity2, dpcapacity3, dpcapacity4 from zipcode_dpcapacity and store them as their original names in information.xlsx.

2. Add a variable "distance2DT"

The distance of zip code to downtown(Cathedral district 60611). We use a zip code Api website (source: https://www.zipcodeapi.com/) to acquire the distance(unit: km) between zip code and the centre zip code of the city. We regard zip code 60611 as the centre.


Live API
Register for Application Key
Free Key
More than 10 Requests/Hour
Low Cost Upgrades
To use on your site
Set up JavaScript Access
JavaScript: City/State Auto-Fill
JavaScript: Zip Code Distance
API: Zip Code Distance

Use this API to determine the distance between two zip codes. Send a GET request to `https://www.zipcodeapi.com/rest/<api_key>/distance.<format>/<zip_code1>/<zip_code2>/<units>`.

API Key

QLVECFIRaK9zMQelqyEs6lyeyLY7I5JtOCnn4LY5JP2H0Rm963rGzI9bgSNLcfxu

Format: json

Zip Code 1: Zip Code

Zip Code 2: Zip Code

Units: km

Make Request
API: Multiple Zip Code Distances
API: Zip Codes by Radius
API: Multiple Zip Codes by Radius
API: Find Close Zip Codes

Figure 1: A caption

3. From "Zipcode_level_all_V3.csv", select 12 monthly trips (from stations) (SumTripsmonth#De, # represents the number of months),53 weekly trips(from stations)(ST#De, # represents the number of weeks), "income", "walk score", "bike score","TotalPopulation", "Whitealone", "Population65yearsandover".

The information.xlsx contains the variables we want to include in our regression model. However, the row of this file is on zip code level(total 40 rows). To prepare for a zip code - month level (total 40 x 12 rows),data, we use the following Matlab code "create_matrix_month.m". Meanwhile, we generate several dummy variables

Matlab code

In our Matlab code, we turn rows from zipcode level to zipcode_month level. We then transform the monthly_trip_from_stations variables(SumTripsmonth3De,40x12 matrix, # represents the number of months) into a 480x1 column. Based on the row information, the Matlab file "create_matix_month.m" creates the zip code dummies(sub_60601_dummy,sub_60602_dummy,. . . ,sub_60699_dummy), month dummies (Jan_dummy,Feb_dummy,. . . ,Dec_dummy).

We compute the white_ratio and old_ratio based on "TotalPopulation", "Whitealone",and "Population65yearsandover".

We also code a dummy variable D4E_zipcode to reflect the effect of D4E program. According to the policy of Divvy company, households with income less than 35000 will enjoy a reduced fee of 5 dollars a year (detailed information is provided in a sepearate document regarding the event background).

Ideally, we should observe whether a user is "treated" with D4E or not. However, we have no access to the company's private data. We adopt an alternative approach: we code a dummy variable D4E_zipcode such that zipcodes with median income lower than 35000 are denoted as 1 (assumption: if there are more qualified low-income households residing in a zipcode, we consider that zipcode to have a larger likelihood of getting D4E treatment compared with other zipcodes .)

The output file is called "regression_matrix_month.csv". The data has 480 rows (40 zip codes and 12 months) and 72 columns in total.

The final dataset "regression_matrix_v2.csv"

We return to R to do further data manipulation.

```
#load the data created by matlab
regression_matrix <- read_csv("regression_matrix_month.csv")
dim(regression_matrix)
```

```
## [1] 480  72
```

Var: "D4E_zipcode" & "D4E_treat"

D4E_zipcode is defined according to zipcode median income level. As mentioned earlier, for the zipcodes whose median income <35000 (income level required by D4E program), D4E_zipcode = 1; otherwise, D4E_zipcode = 0.

We code a new dummy variable called D4E_treat based on D4E_zipcode and month_dummy variables to capture before and after effect. The launch date of D4E is July 7th. We set D4E_treat = 1 after July 7th in the implementation area.

The code is as below:

```
regression_matrix$D4E_treat <- regression_matrix$D4E_zipcode *(regression_matrix$Jul_dummy+
                                                              regression_matrix$Aug_dummy+
                                                              regression_matrix$Sep_dummy+
                                                              regression_matrix$Oct_dummy+
                                                              regression_matrix$Nov_dummy+
                                                              regression_matrix$Dec_dummy)
```

With D4E_treat, we could achieve the following effects:

- Zipcodes where D4E is implemented: After launch date: D4E_treat = 1. Before launch date: D4E_treat = 0.
- zipcodes where D4E is not implemented: After launch_date: D4E_treat = 0. Before launch_date: D4E_date = 0.

Var: “expApr, expMay”

Expansions in April and May are one-time event. We modify 2 dummy variables expApr and expMay in the following way:

- expApr = 1 for all months in and after the April expansion is implemented in a zipcode.
- expMay = 1 for all months in and after the May expansion is implemented in a zipcode.

```
regression_matrix$expApr <- regression_matrix$expApr*(regression_matrix$Apr_dummy+
                                                         regression_matrix$May_dummy+
                                                         regression_matrix$Jun_dummy+
                                                         regression_matrix$Jul_dummy+
                                                         regression_matrix$Aug_dummy+
                                                         regression_matrix$Sep_dummy+
                                                         regression_matrix$Oct_dummy+
                                                         regression_matrix$Nov_dummy+
                                                         regression_matrix$Dec_dummy)
regression_matrix$expMay <-  regression_matrix$expMay*(regression_matrix$May_dummy+
                                                         regression_matrix$Jun_dummy+
                                                         regression_matrix$Jul_dummy+
                                                         regression_matrix$Aug_dummy+
                                                         regression_matrix$Sep_dummy+
                                                         regression_matrix$Oct_dummy+
                                                         regression_matrix$Nov_dummy+
                                                         regression_matrix$Dec_dummy)
```

Var: “sumStation”

In the regression model, we include the information about the number of stations in each zip code by a variable named sumStation. This variable will have different values before and after expansions (Find more information in zipcode_stations.csv). To ensure that we use the correct corresponding quantity for each row, we consider values from four phases.

- sum_station1 is the total number of stations in each zip code before expansions in 2015.
- total_station_by_April is the total number of stations in each zipcode after the April expansion and before the May expansion in 2015.
- total_station_by_May is the total number of stations in each zipcode after the May expansion and before the other expansions of the station in the remaining year of 2015.
- sum_station2 is the total number of stations after April, May expansions in 2015. It includes the number of stations whose online date is after May.

For each row, we only need one of these four values to represent the number of stations at the corresponding time. We make the selection based on month dummies. By doing so, we achieve the following goals:

- Before April, we use sum_station1 as sumStation.
- In April, we use total_station_by_April as sumStation.
- In May, we use total_station_by_May as sumStation.
- After May, we use sum_station2 as sumStation.

```
regression_matrix$sum_station1 <- regression_matrix$sum_station_1*
  (regression_matrix$Jan_dummy+
     regression_matrix$Feb_dummy+
     regression_matrix$Mar_dummy)
```

```
regression_matrix$total_station_by_April<-regression_matrix$total_station_by_April*
  regression_matrix$Apr_dummy

regression_matrix$total_station_by_May<-regression_matrix$total_station_by_May*
  regression_matrix$May_dummy


regression_matrix$sum_station2 <- regression_matrix$sum_station_2*
  (regression_matrix$Jun_dummy+
     regression_matrix$Jul_dummy+
     regression_matrix$Aug_dummy+
     regression_matrix$Sep_dummy+
     regression_matrix$Oct_dummy+
     regression_matrix$Nov_dummy+
     regression_matrix$Dec_dummy)

regression_matrix$sumStation <- regression_matrix$sum_station1+
  regression_matrix$total_station_by_April+
  regression_matrix$total_station_by_May+
  regression_matrix$sum_station2
```

Var: “dpcapacity”

We include the total number of docks in each zipcode in the regression model as dpcapcity. Since there are two expansions during 2015, we have different dpcapcity for different periods. In our previous code, we compute the dpcapacity for each periods and store the values in file “zipcode_dpcapacity.csv” as dpcapacity1 (the dock capacity of each zip code before 2015 expansions), dpcapacity2 (the dock capacity of each zipcode after 2015 April expansion), dpcapacity3 (the dock capacity of each zipcode after 2015 May expansion), dpcapacity4 (the dock dpcapacity of each zip code after 2015 expansions). To avoid redundancy, we code a variable “dpcapacity” based on the four variables. For different periods, the variable dpcapacity is equal to the dpcapacity of that period.

```
regression_matrix$dpcapacity <- regression_matrix$dpcapacity1*
  (regression_matrix$Jan_dummy+
  regression_matrix$Feb_dummy+
  regression_matrix$Mar_dummy)+
  regression_matrix$dpcapacity2*
  (regression_matrix$Apr_dummy)+
  regression_matrix$dpcapacity3*
  (regression_matrix$May_dummy)+
  regression_matrix$dpcapacity4*
     (regression_matrix$Jun_dummy+
     regression_matrix$Jul_dummy+
     regression_matrix$Aug_dummy+
     regression_matrix$Sep_dummy+
     regression_matrix$Oct_dummy+
     regression_matrix$Nov_dummy+
     regression_matrix$Dec_dummy)
```

```
#Write the output file
write.csv(regression_matrix,"regression_matrix_month_v2.csv")
```

We save the file as regression_matrix_month_v2.csv. This file is used in the regression model.

```r
model1 <- lm(monthly_trip_from_station ~ distance2DT+expApr+expMay+income
            +walk_score+transit_score+bike_score+D4E_treat
            +sumStation+dpcapacity, data=regression_matrix)
summary(model1)
```

```
##
## Call:
## lm(formula = monthly_trip_from_station ~ distance2DT + expApr +
##     expMay + income + walk_score + transit_score + bike_score +
##     D4E_treat + sumStation + dpcapacity, data = regression_matrix)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14526  -3042   -463   1422  49708
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.888e+03  5.626e+03   1.580  0.11481
## distance2DT   -2.188e+02  1.415e+02  -1.546  0.12281
## expApr         1.790e+02  7.134e+02   0.251  0.80195
## expMay         2.060e+03  7.093e+02   2.904  0.00385 **
## income         4.285e-02  1.671e-02   2.565  0.01064 *
## walk_score     9.031e+01  3.988e+01   2.265  0.02399 *
## transit_score -1.577e+02  5.639e+01  -2.797  0.00537 **
## bike_score    -7.587e+01  6.023e+01  -1.260  0.20839
## D4E_treat     -1.254e+02  1.155e+03  -0.109  0.91360
## sumStation    -1.252e+03  2.300e+02  -5.445 8.39e-08 ***
## dpcapacity     9.315e+01  1.437e+01   6.482 2.30e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6581 on 469 degrees of freedom
## Multiple R-squared:  0.3612, Adjusted R-squared:  0.3476
## F-statistic: 26.52 on 10 and 469 DF,  p-value: < 2.2e-16
```

```r
cor(regression_matrix$sumStation,regression_matrix$dpcapacity)
```

```
## [1] 0.9746438
```

```r
# use dpcapacity
model2 <- lm(monthly_trip_from_station ~ distance2DT+income
            +walk_score+transit_score+bike_score+D4E_treat
            +dpcapacity+Jan_dummy+Feb_dummy+Mar_dummy+Apr_dummy
            +May_dummy+Jun_dummy+Jul_dummy+Aug_dummy+Sep_dummy
            +Oct_dummy+Nov_dummy, data=regression_matrix)
summary(model2)
```

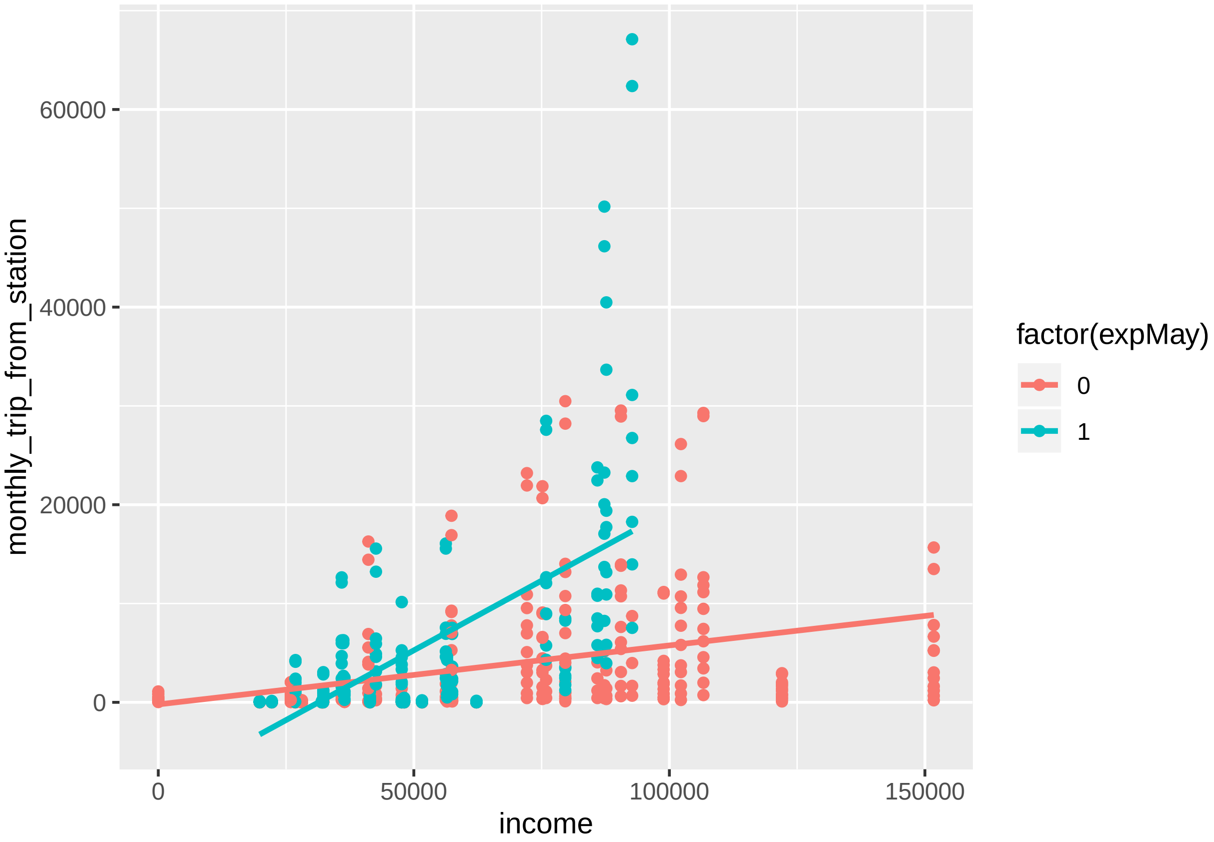
```
##
## Call:
## lm(formula = monthly_trip_from_station ~ distance2DT + income +
##     walk_score + transit_score + bike_score + D4E_treat + dpcapacity +
##     Jan_dummy + Feb_dummy + Mar_dummy + Apr_dummy + May_dummy +
##     Jun_dummy + Jul_dummy + Aug_dummy + Sep_dummy + Oct_dummy +
##     Nov_dummy, data = regression_matrix)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max 
## -13474  -3283     37   2242  44399 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    1.355e+04  4.922e+03   2.752 0.006151 ** 
## distance2DT   -4.311e+02  1.123e+02  -3.838 0.000141 ***
## income         2.802e-02  1.420e-02   1.973 0.049118 *  
## walk_score     5.112e+01  3.313e+01   1.543 0.123479    
## transit_score -4.336e+01  4.316e+01  -1.005 0.315657    
## bike_score    -1.555e+02  5.063e+01  -3.072 0.002256 ** 
## D4E_treat     -4.631e+03  9.956e+02  -4.651 4.32e-06 ***
## dpcapacity     1.421e+01  2.005e+00   7.085 5.23e-12 ***
## Jan_dummy     -1.816e+03  1.283e+03  -1.416 0.157571    
## Feb_dummy     -1.115e+03  1.283e+03  -0.869 0.385092    
## Mar_dummy     -1.559e+03  1.283e+03  -1.215 0.224920    
## Apr_dummy     -8.377e+02  1.274e+03  -0.658 0.511154    
## May_dummy     -3.639e+02  1.270e+03  -0.287 0.774526    
## Jun_dummy      1.362e+03  1.269e+03   1.073 0.283890    
## Jul_dummy      4.169e+03  1.245e+03   3.349 0.000877 ***
## Aug_dummy      1.144e+04  1.245e+03   9.194  < 2e-16 ***
## Sep_dummy      1.048e+04  1.245e+03   8.417 4.93e-16 ***
## Oct_dummy      3.799e+03  1.245e+03   3.052 0.002402 ** 
## Nov_dummy      2.285e+03  1.245e+03   1.836 0.067025 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5566 on 461 degrees of freedom
## Multiple R-squared:  0.5509, Adjusted R-squared:  0.5333 
## F-statistic: 31.41 on 18 and 461 DF,  p-value: < 2.2e-16
```

```
model3 <- lm(monthly_trip_from_station ~ income + distance2DT
             +walk_score+transit_score+bike_score+D4E_treat
             +dpcapacity+expMay+expApr, data=regression_matrix)
summary(model3)
```

```
## 
## Call:
## lm(formula = monthly_trip_from_station ~ income + distance2DT + 
##     walk_score + transit_score + bike_score + D4E_treat + dpcapacity + 
##     expMay + expApr, data = regression_matrix)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -12771  -3723   -647   1731  51003 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   8583.1445  5794.2174   1.481 0.139189    
## income           0.0456     0.0172   2.651 0.008300 ** 
## distance2DT   -416.2023   140.9075  -2.954 0.003297 ** 
## walk_score      55.3683    40.5413   1.366 0.172679    
## transit_score  -31.3776    52.9403  -0.593 0.553669    
## bike_score    -119.9034    61.4746  -1.950 0.051716 .  
## D4E_treat      -460.0831  1188.3457  -0.387 0.698811
## dpcapacity       16.1642     2.6426   6.117 2.01e-09 ***
## expMay         2422.4528   727.3658   3.330 0.000935 ***
## expApr          -47.3814   733.5764  -0.065 0.948528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6779 on 470 degrees of freedom
## Multiple R-squared:  0.3209, Adjusted R-squared:  0.3078
## F-statistic: 24.67 on 9 and 470 DF,  p-value: < 2.2e-16
```

```
ggplot(model1,aes(y=monthly_trip_from_station,x=income,
                 color=factor(expMay)))+geom_point()+stat_smooth(method="lm",se=FALSE)
```


monthly_trip_from_station
income
factor(expMay)