Final_project_MA799

Code

Nam Pham, Linh Truong

December 18, 2016

1 Introduction

Hubway is a bike-sharing system, based in the Greater Boston Metropolitian areas. Launching in 2011, Hubway has now owned over 1,600 bikes at 160+ stations across Boston, Brookline, Cambridge and Somerville. As fast as Hubway bike network has been growing, its users also showed different types of behaviors. The goal of this project is to investigate into further insights about Hubway users and their corresponding behaviors, based on the Hubway data recorded about trips taken from July 2011 to November 2013. We aim to develop a good understanding of behaviors of different demographics of Hubway users.

As such, the ultimate goal of the project is to investigate Hubway stations and trips. We will analyze the geographical locations of all Hubway stations, as well as the traffic in-between these stations. We aim to develop a good understanding of how people use this service and recommend any solution to improve the service and quality of the overall system.

2 Data Description

The dataset used in this report is obtained from Hubway Data Visualization Challenge. We obtained two CSV files. The first file `df.stations` lists the details regarding 142 Hubway bike stations, which include properties such as id (integer), terminal code (character), station name (character), municipal (character), latitude (num), longitude (num), status (existing/removed-character).

The second file `df.trips` describes the trips from July 2011 to December 2013. Details about these trips include:

- seq_id: unique record ID (integer)
- hubway_id: trip ID (integer)

- status: Closed- meaning a trip is completed (character)
- duration: length of trip in seconds (integer)
- start_date: start date of trip with date and time (string)
- str_station: start station (integer)
- end_date: start date of trip with date and time (string)
- end_station: end station (integer).
- bike_nr: bike number (character)
- subsc_type: subscription type (Casual/Registered)(character)
- zip_code: zip code of users (character)
- birth_date: birth year of users (integer)
- gender: male/female (character)

3 Data Preparation

We use the following libraries in our project:

Hide

```
library(igraph)
library(dplyr)
library(readr)
library(ggplot2)
library(maptools)
library(RNeo4j)
library(stringr)
library(ggmap)
```

We import the dataset from two CSV files located locally:

Hide

```
csv.folder = "~/Documents/R projects"
file.stations = paste(csv.folder, "hubway_stations.csv", sep="/")
file.trips    = paste(csv.folder, "hubway_trips.csv"   , sep="/")
df.stations = read_csv(file=file.stations)
df.trips    = read_csv(file=file.trips)
```

The original `df.trips` has 1,579,025 observations. As we analyze the trips dataset, we have the following consideration:

- All of the trips have status “Closed”, meaning all trips are completed. Thus this variable is not selected
- Remove the NA for start station and end station. These missing values do not give any insights about the trips, thus we only select the trips with complete start and end station
- We remove all the trips with duration less than a minute. We believe that these trips occur because renters decided to check out and changed their minds or something happen to their bikes, thus they return immediately. We only focus on trips longer than a minute (or 60 seconds)
- Zip code contains some apostrophe in the beginning, so we removed this. Some of the zipcodes has less than 5 characters because of the missing zero in the beginning, so we enforce 5 character for all observations and add a leading 0.
- We also add new columns. Since we are interested in the timeframe of the trip, we convert the

start_date, which is a character string, into a date variable and subsequently extract month, day of the week. We are also interested in the age of the renter, so we subtract their birth year from 2013 to get their age at the time of this dataset.

Hide

```
df.trips %>%
  filter(!is.na(strt_statn),
         !is.na( end_statn),
         duration > 60) %>%           #include trip > 1 minute
  mutate(age=2013-birth_date) %>%     #add age variable at the time of dataset
  mutate(raw.date=as.Date(gsub( " .*$", "", start_date), "%m/%d/%Y")) %>%
  #extract the date before the white space and convert to Date type
  mutate(duration=duration/60)%>% #convert duration to minutes
  mutate(zip_code=str_pad((gsub("'", '',zip_code)), 5, pad="0")) %>%
  #remove ' in the beginning and add a leading 0 if zipcode has less than 4 letter
  select(seq_id, hubway_id, strt_statn, end_statn,
         start_date, end_date, subsc_type,
         duration, zip_code, birth_date, gender, age, raw.date, bike_nr)%>%
         {.}->df.trips
```

The resulting `df.trips` dataset now has 1,566,507 observations; we have removed 12,518 observations. The `df.stations` has 142 observations with 6 columns. There are 12 removed stations and 130 existing. However, the removed stations are moved only across the street and the new stations are given the same terminal name, so we keep all the variables and observations for this dataset.

4 Variables Summary

Based on the Hubway data from 2011- 2013, we would want to first look at the demographic of Hubway users then examine the characteristics of normal Hubway bike trips and duration.

4.1 Customer Profiles

- In this section, we are interested in the demographic and characteristics of Hubway users. We will look at gender and age distribution of registered Hubway users from 2011 to 2013, then take into account casual users in trip duration.

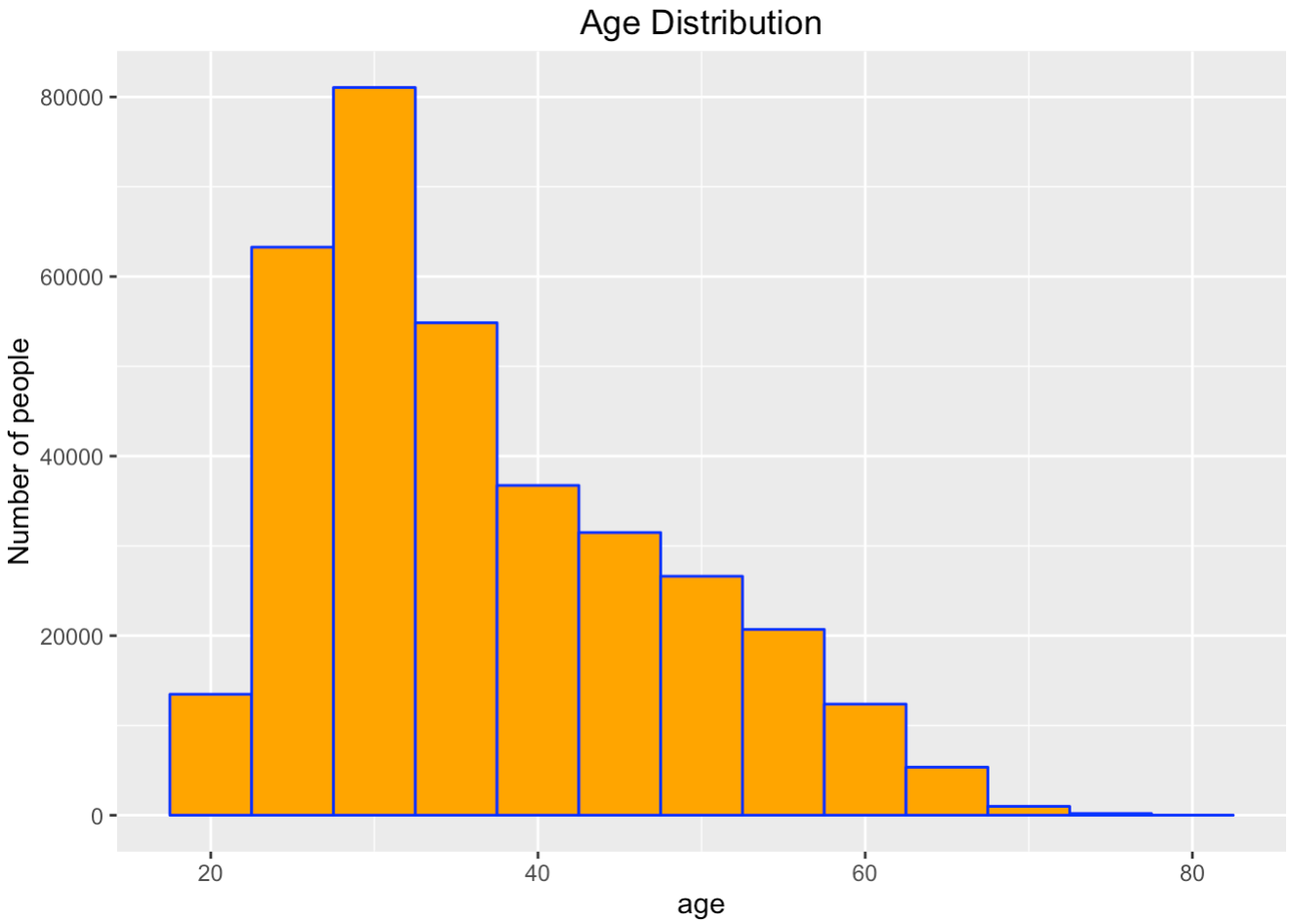
4.1.1 Age distribution

- This following graph looks at the distribution of Hubway users regarding age. This graph only considers registered users, not the casual users. Each column represents 5 age difference (20-25 years old, 25-30 years old etc).

Hide

```
df.trips %>%
  ggplot(aes(x=age)) +
  geom_histogram(binwidth=5,color="blue", fill="orange")+
  ylab("Number of people")+
  ggtitle("Age Distribution")
```

```
## Warning: Removed 1219379 rows containing non-finite values (stat_bin).
```



- From the graph, we can see that age of Hubway registered users vary from 20 years old to 70 years old.
- However, it is also noticable that majority our Hubway registered users is in their 30s. Number of Hubway users at 25-30 and 30-35 years old (around 62,000 and 81,000 respectively) is at the highest, followed by number of users at 35-40 years old (55,000).
- There is very small number of users bellow 25 years old or above 70 years old. The number of users decreases for every age group after 30.

4.1.2 Gender distribution

- This following table will examine the ditribution of registered Hubway users by gender. It also only considers registered, not casual Hubway users.

Hide

```
df.trips %>%
  group_by(gender)%>%
  summarize(mean_duration=mean(duration,na.rm=TRUE),
            count_trips=length(unique(seq_id))) %>%
  arrange(desc(mean_duration))
```

```
## # A tibble: 3 × 3
##   gender mean_duration count_trips
##    <chr>         <dbl>       <int>
## 1   <NA>      35.58769      472471
## 2 Female      13.69908      268568
## 3   Male      11.82712      825468
```

- Based on the users who indicated their sex, male Hubway users use Hubway bikes significantly more than female users. The number of trips by male users (825,468 trips) are roughly three times as much as the number of trips by female users (268,568 trips).
- Male users, however, tend to have slightly shorter trips than female users, with mean duration for male and female being 11.82 minutes and 13.11 minutes, respectively). This data could be interpreted that females bike at a slower speed than their male counterparts, hence, they take longer time on their trips than males.
- We should also note that this data only consider registered users who reported their gender.

4.1.3 Taking casual user into account

- In the above sections, we only consider registered users, but disregard casual users. As a result, in this section, we want to look at the effect of casual users in the duration of trips.
- Taking the subscription type (registered vs. casual) as a dummy variable, the following linear regression examines how trip duration (dependent variable) depends on subscription type (independent variable).

Hide

```
summary(lm(duration~subsc_type,df.trips))
```

```
##
## Call:
## lm(formula = duration ~ subsc_type, data = df.trips)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##    -35     -9     -5      1 199872
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           35.5877     0.4238   83.98   <2e-16 ***
## subsc_typeRegistered -23.3010     0.5071  -45.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 291.3 on 1566505 degrees of freedom
## Multiple R-squared:  0.001346,   Adjusted R-squared:  0.001345
## F-statistic:  2112 on 1 and 1566505 DF,  p-value: < 2.2e-16
```

- This linear regression result tells us that: keeping all other factors constant, a trip will be 23 minute shorter if the person is subscribed.
- We can interpret this result that a person who is registered is likely to use more commute reason and probably use more and shorter trips.
- However, the R-square value is relatively low, so it only explains 0.13% of the variation; so this is only used as a general guideline.

4.2 Trips and Duration

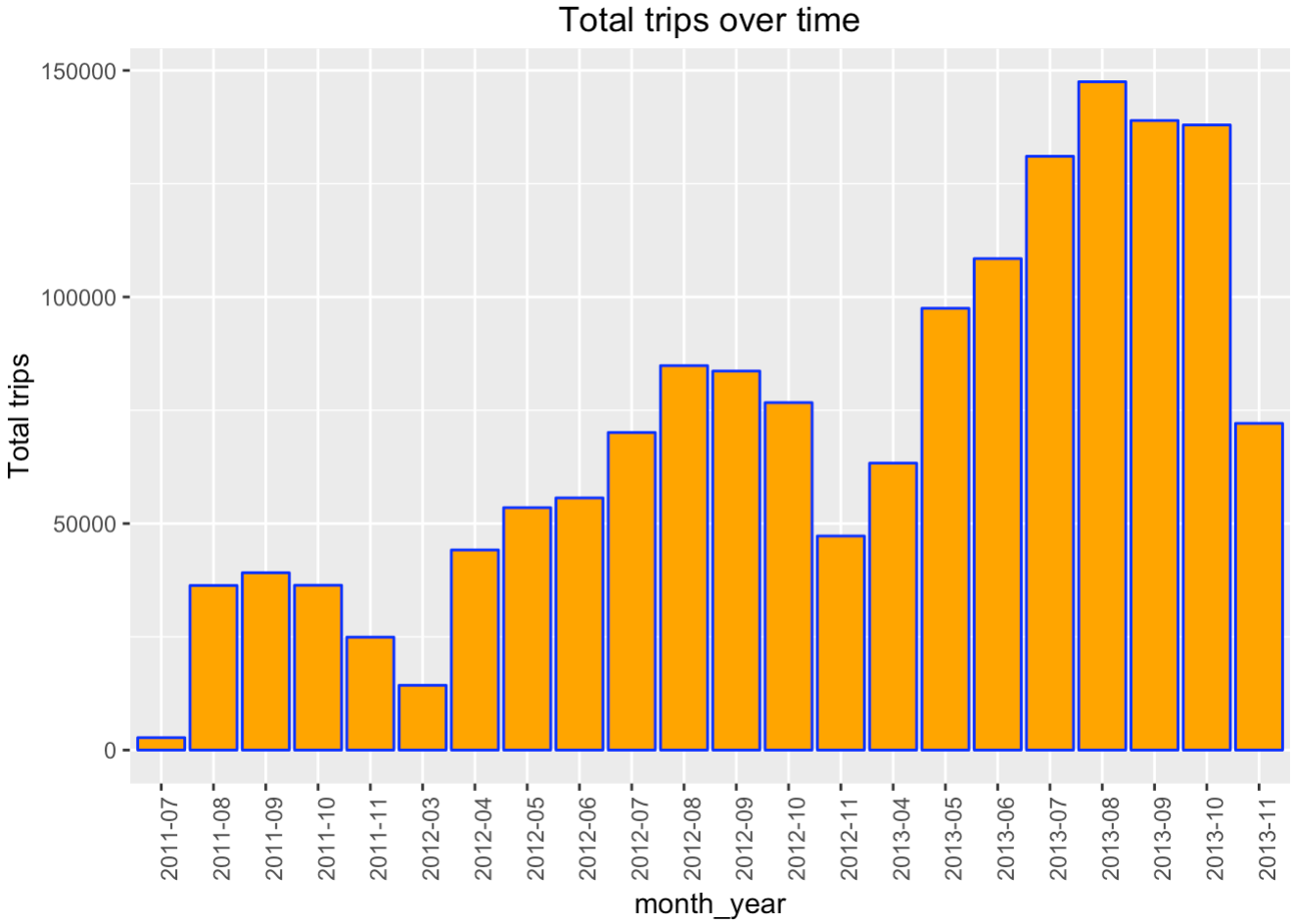
After studying the demographic and characteristics of Hubway users in the previous section, we now want to further examine the characteristics of the trips by those Hubway users. First, we will look at the number of Hubway trips per month to identify the trend, across 3 years and the pattern within a year. We will continue by looking at number of trips and trip duration regarding on each day of the week to see the differences in number of trips/ duration between weekdays and weekend. Lastly, we will examine the trip distribution by duration to see if users usually use Hubway bikes for long trips or short trips.

4.2.1 Number of trips per month

- First, we would want to look at Hubway number trips per month, based on from July 2011 to November 2013.
- It should be noted that Hubway stations closed from December to February of 2011 and 2012, and from December to March of 2013.

Hide

```
df.trips %>%
  mutate(month_year=format(raw.date,"%Y-%m")) %>%
  ggplot(aes(x=month_year,y=(unique(seq_id))))+
  stat_summary(fun.y=length,geom="bar",fill="orange",color="blue")  +
  #count the length of unique trip id, which is total trips
  theme(axis.text.x=element_text(angle=90,hjust=1))+
  #rotate the x-axis by 90 degree for easier viewing
  ggtitle("Total trips over time") +
  ylab("Total trips")
```



- From the histogram, we can see that the number of trips has grown significantly from 2011 (peaked at around 40,000 trips in September 2011) to 2013 (peaked at nearly 150,000 trips in August 2013)
- There is also some seasonality in total of trips per month within the year. Number of trips starts low in early of the year (March, April), then grows gradually until it peaks in August or September, before it stumbles again in November. This trend makes sense because those low-traffic months are usually winter time and commuting by bikes would be difficult in inclement weather.

4.2.2 Number of trips and duration sorting each day of the week

- Next, we would want to investigate the pattern of number of trips and duration in different days of the week. The following table summarize number of trips and average durationg regarding each day.

Hide

```
df.trips %>%
  mutate(weekday=weekdays(raw.date)) %>% #extract weekday from raw date
  group_by(weekday)%>%
  summarize(count_trips=length(unique(seq_id)),
            mean_duration=mean(duration,na.rm=TRUE)) %>%
  arrange(desc(count_trips))
```

```
## # A tibble: 7 × 3
##     weekday count_trips mean_duration
##       <chr>       <int>         <dbl>
## 1 Wednesday      236947      15.82577
## 2  Thursday      232949      15.99679
## 3   Tuesday      230074      15.60927
## 4    Friday      229488      18.90661
## 5    Monday      228645      18.57396
## 6  Saturday      214067      26.11510
## 7    Sunday      194337      25.79301
```

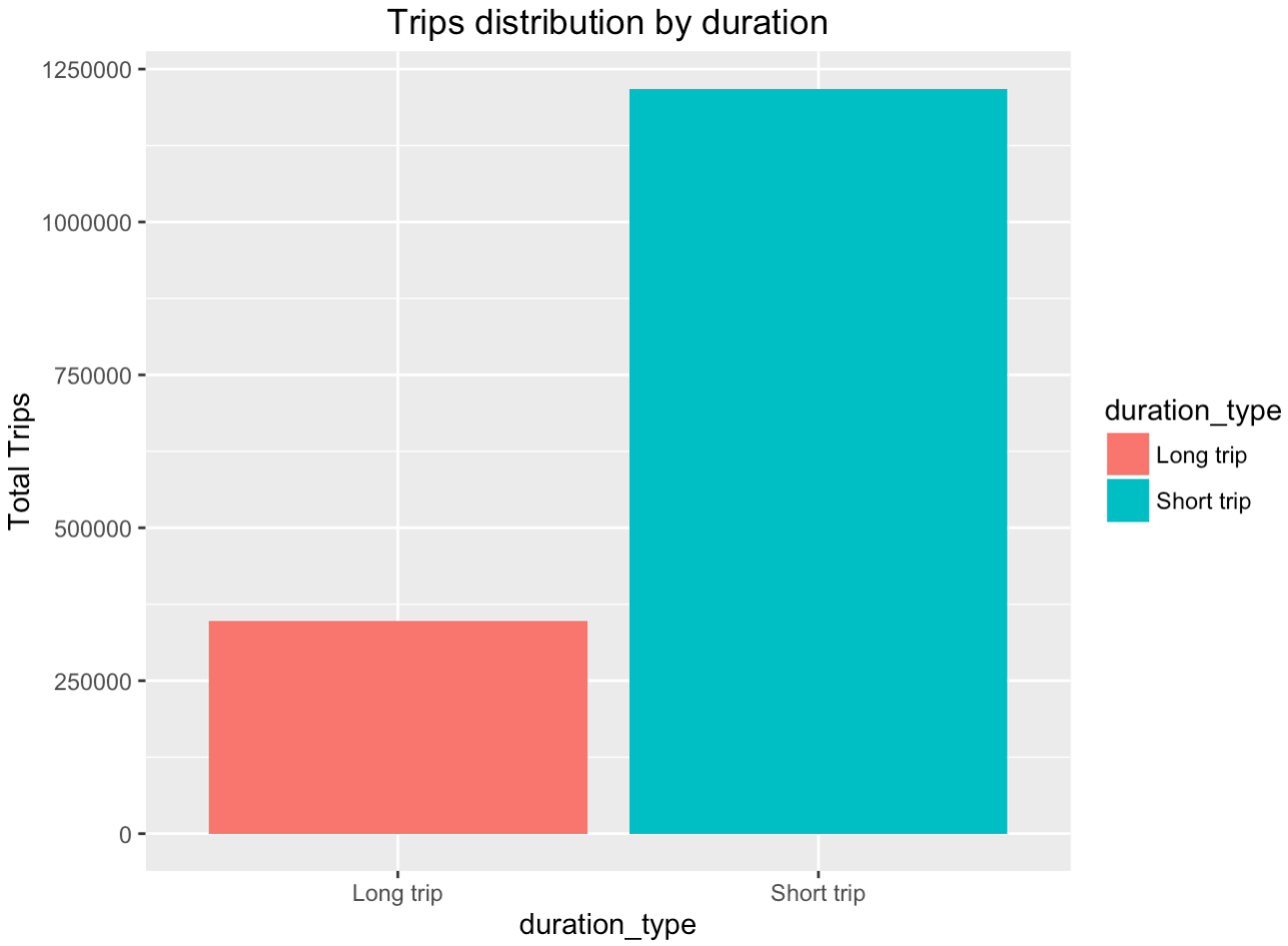
- From the above table, we can see that Tuesday, Wednesday, and Thursday are the most trafficked days of the week (with over 230,000 trips), yet these days have the shortest average trip duration (only 15 minutes per trip).
- Traffic on weekends, (Staturday and Sunday), is at the lowest (around 200,000 trips), yet has the longest trip duration (at 26 minutes).
- The pattern of number of trips per day increases gradually from Monday (228,645 trips) to Tuesday (230,047 trips), then peaked on Wednesday (236,947 trips) and Thursday (232,949 trips), then gradually drops on Friday to 228,645 trips and on Saturday to 214,067 trip. Number of trips per day at its lowest on Sunday, with 194,337 trips. It might be possible that people tend to be busier in the beginning of the week and may not use Hubway as much.
- On the contrary, the trip duration stumbles from 18.57 minutes/trips on Monday to 15.60 minutes/trips on Tuesday, and stays the same on Wednesday, Thursday, with 15.82 and 15.99 minutes/trips respectively. The trip duration gradually increases again on Friday, at 18.90 minutes/trip and peaked at 26.11 minutes/trip on Saturday and 25.79 minutes/trip on Sunday. This indicates that there's a large number of people who use on the weekend for leisure purposes.

4.2.3 Trip distribution by duration

- From the previous section, we know that Hubway users have long bike trips on weekends and short bike trips on weekdays. To continue, we want to compare the total number of long trip to total the number of long trips across 3 years
- We categorize long trip as the ones more than 20 minutes, and short trips as the ones under 20 minutes

Hide

```
df.trips %>%
  mutate(duration_type=ifelse(df.trips$duration >=20, "Long trip", "Short trip"))%>%
  #if duration is longer than 20 minutes, then "Long trip", otherwise it is short
  ggplot  (aes(x   =duration_type)) +
  geom_bar(aes(fill=duration_type)) +
  ylab("Total Trips")+
  ggtitle("Trips distribution by duration")
```



- From this graph, it is noticable that there are significantly more short trips compare to long trips. This means that Hubway users usually use Hubway bikes for short trips rather than long trips. This is consistent with our findings that Hubway core customers are young professionals who bike to work.

5 Network Analysis

Our network analysis aims to understand the network of Hubway through the stations and trips. First, we want to visualize where the stations are located. We will plot the choropleth map on all the station on the Boston map. We also want to include the size of each points to indicate the traffic in each location. In order to do that, we add new column to the `df.stations` by calculating the total incoming trips from the `df.trips` dataset.

Hide

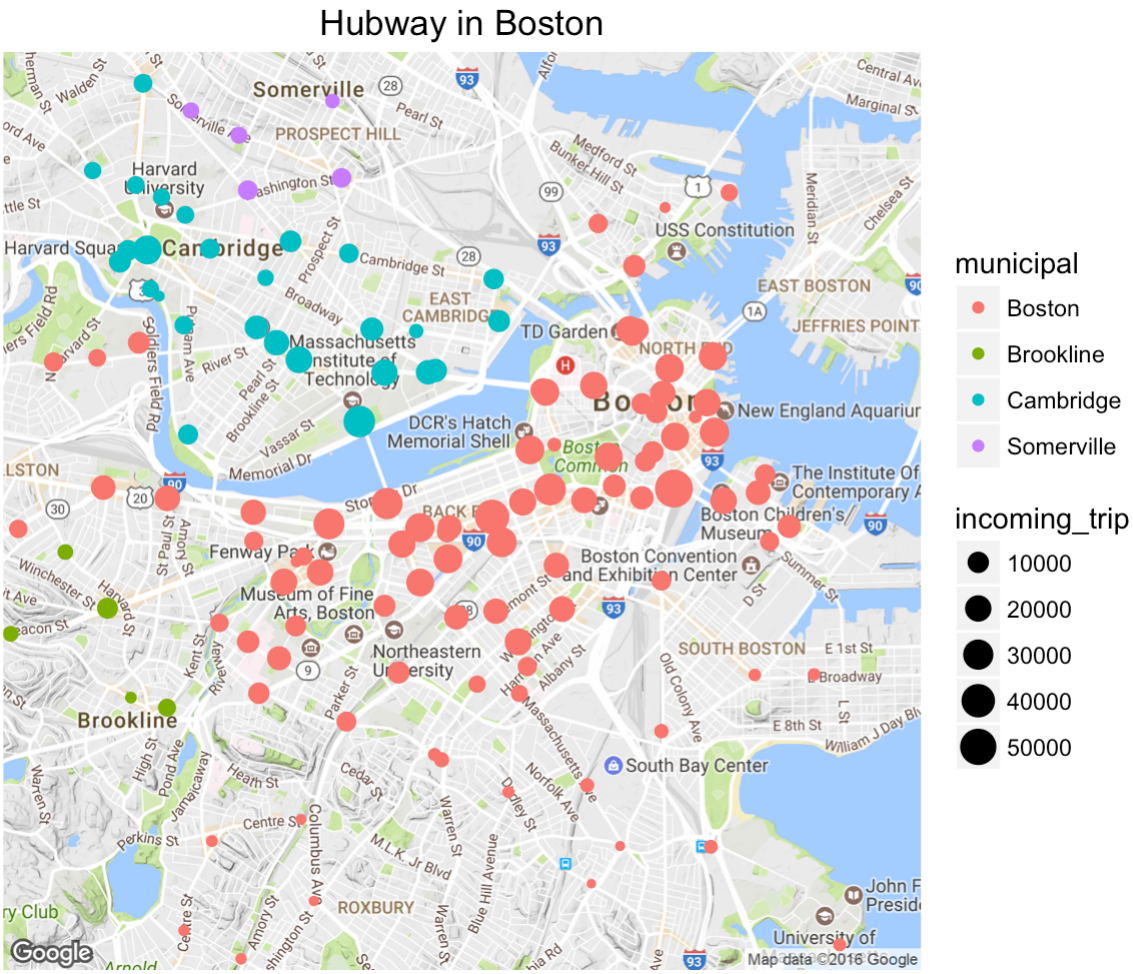
```
df.trips%>%
  group_by(end_statn)%>%
  summarize(incoming_trip=length(end_statn))-> count_trips
  #count number of times that end_station appears, grouped by itself

df.stations%>%
  mutate(incoming_trip=count_trips$incoming_trip)%>%
  #add new column of incoming trips to existing station, because station_id in both t
ables are arranged from 3->145, making it possible to merge
  {.}->df.stations.graph
```

We then plot the map using the `ggmap` and `ggplot2` libraries

Hide

```
qmap(location = "Back Bay, MA", zoom=13, maptype= "terrain" ) +
  geom_point(data=df.stations.graph,aes(x=lng,y=lat,col=municipal, size=incoming_trip
))+
  ggtitle("Hubway in Boston")
```



As we can see from the map, Boston areas are the busiest locations for Hubway. The average incoming trips to Boston stations are much higher, and the stations are located densely.

We then fetch the data into Neo4j for graph analysis. We are interested in the network for registered user from 1/1/2013 to 12/31/2013

Hide

```
setwd("~/Documents/Neo4j/default.graphdb/import")
#set current directory to Neo4j
df.trips%>%
  filter(raw.date>"2012-12-31")%>%
  filter(subsc_type=="Registered")%>%
  select(duration, strt_statn,end_statn,duration,bike_nr,zip_code,age)%>%
  {.}->df.trips.neo
#filter trips dataset to make it smaller and easier to load in Neo4j
write_csv(df.stations,"hubway_stations.csv")
write_csv(df.trips.neo,"hubway_trips.csv")
#write into new CSV files
```

We then add the following code in Neo4j console to create the stations and relationships among them. We also added the station status, municipal properties for stations, and duration, zipcode, age for trips.

```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:/hubway_stations.csv" as row
CREATE (:STATION {station_id:row.id, station_name:row.station,
station_status:row.status, station_municipal:row.municipal})

USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:/hubway_trips.csv" as row
MATCH (from_station {station_id:row.strt_statn})
MATCH ( to_station {station_id:row.end_statn})
CREATE (from_station)-[:TRIP {duration:row.duration, zip_code:row.zip_code,
age:row.age}]->(to_station)
```

We get the graph from Neo4j:

Hide

```
graph = startGraph("http://localhost:7474/db/data/",
                   username = "neo4j",
                   password = "neo4j12345")
```

Our analysis plan for Hubway network is as follow:

- First, we want to find out the “top” stations in the network. There are different criteria for finding this station: we can look at the total traffic that coming in/out of the stations, the total stations that it serves to, and the duration that people use the service at this station.
- Next, we look at the municipal with the lowest traffic. We want to explore the current stations in this area, and compare the top and bottom station to find out if there is any specific patterns from the result. From that, we can recommend any solution to improve the least-performing station.

First, we look at the top stations by the number of trips (both to and from). We take the total of trips in each station and divided by the total to get the total percentage of traffic that it is responsible for.

Hide

```
cypher(graph,"match (n)-[t:TRIP]->(m)
with count(distinct t) as Total_Traffic
       match (n)-[t:TRIP]-(m)
       return n.station_name as Station,count(distinct t) as Total_Trips, round(count
(distinct t)*100)/Total_Traffic + '%' as Frequency
       order by Total_Trips desc limit 5")
```

```
##                                  Station Total_Trips           Frequency
## 1       South Station - 700 Atlantic Ave.       42438  6.500162358548944%
## 2            MIT at Mass Ave / Amherst St       38576 5.9086258340021685%
## 3  TD Garden - Causeway at Portal Park #1       29849  4.571924837181946%
## 4                    Beacon St / Mass Ave       26685  4.087299885430005%
## 5 MIT Stata Center at Vassar St / Main St       25403 3.8909379422738772%
```

In the first part of the query, we get the Total_Traffic variable from a directed match to get the total number of distinct trips in 2013. In the second part, we use a non-directed match, because we want to get all the trips from and to that station. The total will get us the number of trips that each station is responsible for; after that, we divided by the total trip to get the frequency of total traffic that particular station appears in. From the result, South Station has the highest number of total trips in 2013 with 6.5% frequency in toal traffic.

Next, we look at the top station by the number of stations that people travel to from that station.

Hide

```
cypher(graph,"match (n)-[t:TRIP]->(m)
return n.station_name as Station_name, n.station_municipal as Municipal , count(disti
nct m) as Total_stations
order by Total_stations desc limit 5")
```

```
##                                  Station_name Municipal Total_stations
## 1                        Boylston at Fairfield    Boston            127
## 2      Boston Public Library - 700 Boylston St.    Boston            126
## 3                           Boylston / Mass Ave    Boston            125
## 4 Charles Circle - Charles St. at Cambridge St.    Boston            124
## 5                  MIT at Mass Ave / Amherst St Cambridge            124
```

We found out that from Boylston st. station, renters traveled to a total of 127 stations out of 142 stations. This suggests that although the traffic at Boylston is not as high as South Station, people like to travel to a wider variety of different stations, probably for leisure purposes.

Lastly, we rank station by how long users rent their bikes at that station:

Hide

```
cypher(graph,"match (n)-[t:TRIP]->(m)
        return n.station_name as Station_name, n.station_municipal as Municipal ,avg(
toInt(t.duration)) as Avg_Duration, count(t) as Total_Trip
       order by Avg_Duration desc limit 5")
```

```
##                                Station_name Municipal Avg_Duration
## 1         University of Massachusetts Boston    Boston     26.22487
## 2              Jackson Square T at Centre St    Boston     22.95726
## 3                        JFK / UMASS Station    Boston     22.37458
## 4        New Balance - Guest St. at Life St.    Boston     20.82008
## 5 JP Centre - Centre Street at Myrtle Street    Boston     18.52018
##   Total_Trip
## 1        378
## 2        117
## 3        598
## 4        717
## 5        223
```

UMass Boston station ranks 1st; on average, renters use their bikes for 26 minutes, which is a very long amount of time. The total trips from this station is much less than those from South Station or MIT. This suggests that this station is not located in a busy area, and people travel for very long distance. This could also mean that the stations around UMass are located further away from one another.

After analyzing the overall picture, we want to focus on the municipal that uses Hubway the least and analyze the stations in that area. First we want to find out which area uses the least by calculating the total number of trips departing from that area.

Hide

```
cypher(graph,"match(n)-[t:TRIP]->(m)
       with count(t) as total_trips
       match(n)-[t:TRIP]->(m)
       return m.station_municipal as Municipal,count(t) as Total,round(100* count(t))
/total_trips+'%' as Percentage
       order by Total desc")
```

```
##    Municipal  Total          Percentage
## 1     Boston 431834  66.14334115513512%
## 2  Cambridge 182548 27.960592823139464%
## 3 Somerville  26152  4.005661105631084%
## 4  Brookline  12342 1.8904049160943273%
```

Consistent with the first result from previous section, the query returns Boston and Cambridge as the ones whose renters travel most frequently. Combined together, these two areas account for 94.1% of total trips in 2013. This could be explained because these two areas are the hubs for workplaces and universities, and probably they have higher total population and population density. On the other hand, Brookline renters only accounts for 1.8% of total traffic.

We are interested in from where people are biking to Brookline:

Hide

```
cypher(graph, "match (n)-[t:TRIP]->(m)
where m.station_municipal='Brookline'
with count(t) as Total_Trips_to_Brookline
match (n)-[t:TRIP]->(m)
where m.station_municipal='Brookline'
return n.station_municipal as Municipal_From, count(t) as Total_Trips, round(count(t)
*100)/Total_Trips_to_Brookline + '%'  as Percentage
order by Total_Trips desc")
```

```
##   Municipal_From Total_Trips          Percentage
## 1         Boston        8811   71.3903743315508%
## 2      Brookline        1789  14.49521957543348%
## 3      Cambridge        1656 13.417598444336413%
## 4     Somerville          86 0.6968076486793064%
```

And where Brookline renters are going to:

Hide

```
cypher(graph, "match (n)-[t:TRIP]->(m)
where n.station_municipal='Brookline'
with count(t) as Total_Trips_from_Brookline
match (n)-[t:TRIP]->(m)
where n.station_municipal='Brookline'
return m.station_municipal as Municipal_To, count(t) as Total_Trips, round(count(t)*1
00)/Total_Trips_from_Brookline + '%'  as Percentage
order by Total_Trips desc")
```

```
##   Municipal_To Total_Trips          Percentage
## 1       Boston        9040  71.40600315955766%
## 2    Brookline        1789 14.131121642969985%
## 3    Cambridge        1777 14.036334913112164%
## 4   Somerville          54 0.4265402843601896%
```

From the two tables, we can quickly realize that people in Brookline travels to Boston more than to Brookline itself. It suggests that people are using the bike to commute to work in Boston and return home afterwards. Hubway is served mainly for commuting reasons at these locations.

Next, we will look at all the stations in Brookline:

Hide

```
cypher(graph,"match (n)-[t:TRIP]->(m)
where n.station_municipal='Brookline'
       with count(t) as Total_Traffic
       match (n)-[t:TRIP]->(m)
       where n.station_municipal='Brookline'
       return n.station_id as Id,n.station_name as Station_name, n.station_municipal
as Municipal,
       count(t) as Total_Trips,round(100.0*count(t))/Total_Traffic + '%' as percent
       order by Total_Trips desc")
```

```
##    Id                                    Station_name Municipal
## 1  69          Coolidge Corner - Beacon St @ Centre St Brookline
## 2  86        Brookline Village - Station Street @ MBTA Brookline
## 3 127      JFK Crossing at Harvard St. / Thorndike St. Brookline
## 4 126 Washington Square at Washington St. / Beacon St. Brookline
##   Total_Trips             percent
## 1        4794 37.867298578199055%
## 2        2939  23.21484992101106%
## 3        2522 19.921011058451818%
## 4        2405 18.996840442338073%
```

There are only four stations in total, in which Coolidge Corner is used most frequently. The utilization of other three stations are relatively comparable. Lastly, we will compare Coolidge Corner to Washington Square to see the difference in behavior of users.

Hide

```
cypher(graph,"match (n)-[t:TRIP]->(m)
where n.station_id in ['69','126']
       return n.station_name as Station_name,count(distinct m.station_id) as Total_De
stinations, avg(toInt(t.duration)) as Avg_Duration, count(distinct t.zip_code) as Dis
tinct_zip
       order by Total_Destinations desc")
```

```
##                                    Station_name Total_Destinations
## 1          Coolidge Corner - Beacon St @ Centre St                110
## 2 Washington Square at Washington St. / Beacon St.                100
##   Avg_Duration Distinct_zip
## 1     14.31060           96
## 2     15.43576           65
```

From the result, we can see that people from Washington travel to less destinations and spends more time on their bikes. This suggests that this location is placed further than Coolidge Corner on the map. The number of distinct zip code shows the number of people from different areas who signed up for Hubway. The result suggests that demographic in Washington Square is less diverse or its population is much smaller than Coolidge Corner's.

6 Conclusion

Based on the analysis of the Hubway system, we conclude the following characteristics of trips and customers of Hubway:

- Hubway is utilized mostly by young professionals during the week to commute to their work
- There is a smaller population group who used them casually, and tend to use them longer

Based on the network analysis of registered users in 2013, we have the following take-aways:

- Boston and Cambridge, which are the hubs for workplaces and universities, are the areas that utilized Hubway the most
- Brookline accounts for the least amount of traffic of Hubway. Brookline only has four stations so far, but it has been accepted by working professionals who made round-trips to Boston for work. Among those four station, Washington square has the lowest utilization, mostly because of their further distance to Boston.
- To improve the Hubway utilization rate in Brookline, we suggest that Hubway should put more stations near Boston and Cambridge. These stations will likely attract the target audiences of Hubway, who would use their bikes to commute to work everyday.