Analyse_3: Graph analysis.

Georgie Knight

16 September, 2016

```
library("dplyr")
library("tidyr")
library("lubridate")
library("readr")
library("ggplot2")
library("ggmap")
library("igraph")
library("popgraph")
library("zipcode")
trip_read    <- read_csv("trip_full_updated2.csv")
status_read  <- read_csv("status_full_updated.csv")
trip         <- dplyr::tbl_df(trip_read)
status       <- dplyr::tbl_df(status_read)
```

Introduction

We're going to do use some graph theory to analyse the BABS bike network. This will help us figure out which stations are the most integral to the network.

Create the network

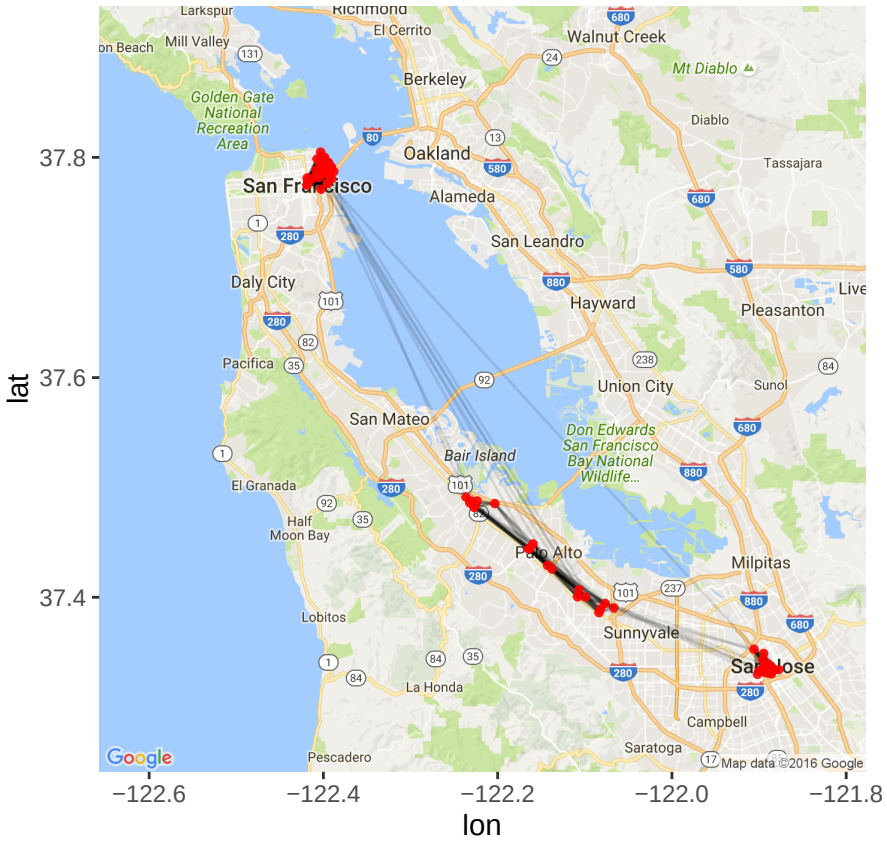
We'll first create a network out of our bike network with stations as the nodes and number of trips between stations as the weights on the edges. We'll normalise these weights.

```
station <-read_csv("201508_station_data.csv")
station <- dplyr::tbl_df(station)
station$name[36] = "Washington at Kearny" #correct the misspellings!
station$name[37] = "Post at Kearny"

tripNumbers <- trip %>%
  group_by(Start.Station,End.Station) %>%
  summarise(weight = n())  %>%
  ungroup() %>%
  mutate(weight = (weight - min(weight))/(max(weight)-min(weight)))

bikeGraph <- graph_from_data_frame(tripNumbers, directed=TRUE) %>%
set_vertex_attr("latitude", value = station$lat) %>%
set_vertex_attr("longitude", value = station$long) %>%
set_vertex_attr("city", value = station$landmark)
```

Visualise the network

We can firstly visualise our network:

```
location <- c( mean(V(bikeGraph)$longitude), mean(V(bikeGraph)$latitude))
BABSmap <- get_map(location,maptype="roadmap", zoom=10)

bayTrips <-trip %>%
  select(Start.Station, End.Station, start_lat, start_long,end_lat, end_long) %>%
  count(Start.Station, End.Station,start_lat, start_long, end_lat, end_long)

ggmap(BABSmap)+
  geom_segment(data = bayTrips,
               aes(y= start_lat,
                   x = start_long,
                   yend =  end_lat,
                   xend = end_long), alpha = 0.1)+
  theme(legend.position="none")+
  geom_nodeset( aes(x=longitude, y=latitude), bikeGraph, size=1, alpha =0.9, col="red")
```



That's a pretty long trip from San Jose to San Francisco!

```
longTrip <- trip %>%
  filter(startLandmark =="San Jose", endLandmark =="San Francisco") %>%
  select(Trip.ID,
         Duration,
         Start_trip,
         Start.Station,
         End_trip,
         End.Station,
```

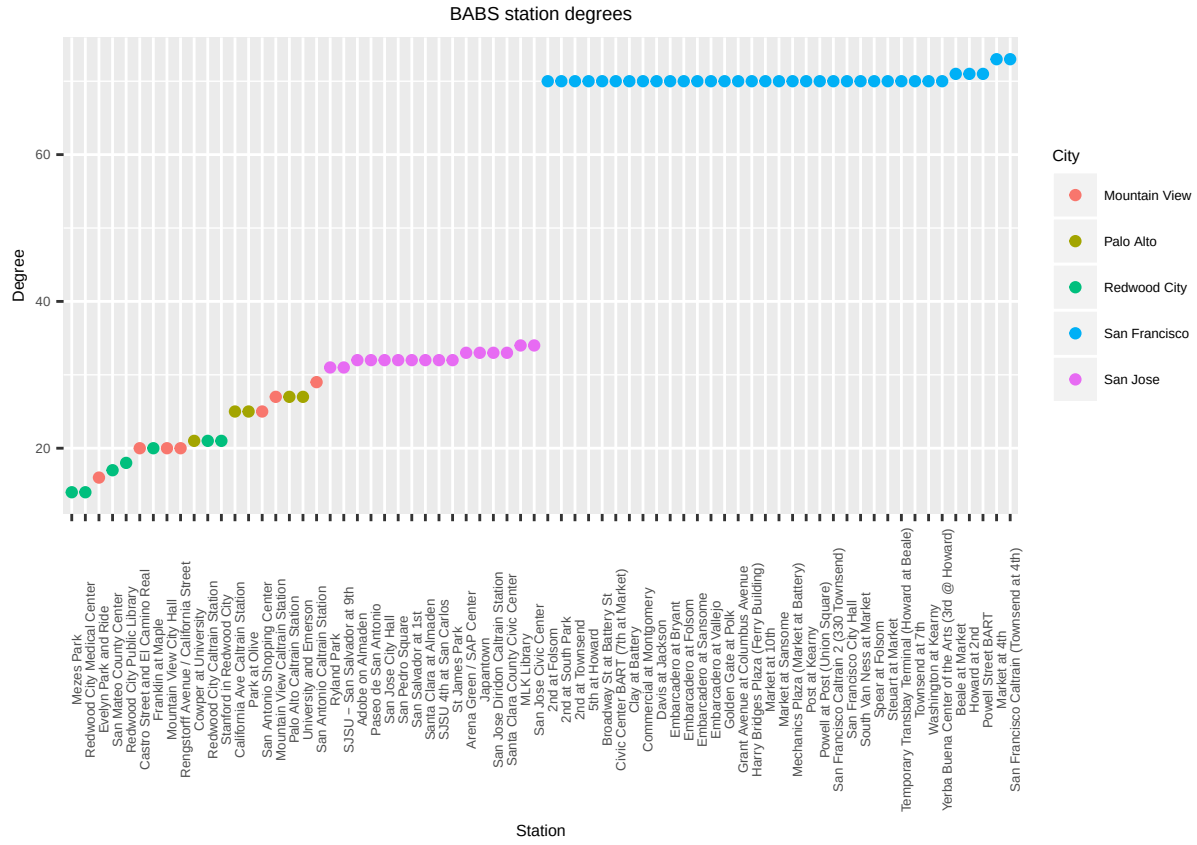
```
        Subscriber.Type)
longTrip
```

```
## # A tibble: 1 x 7
##   Trip.ID Duration          Start_trip Start.Station            End_trip
##     <int>    <int>              <time>         <chr>              <time>
## 1  695326    29942 2015-03-24 13:04:00   MLK Library 2015-03-24 21:23:00
## # ... with 2 more variables: End.Station <chr>, Subscriber.Type <chr>
```

Degree

The degree of a node tells us how many stations it connects to. We'll calculate the sum of the in and out degrees for each station

```
deg <- degree(bikeGraph, mode= "total")
deg <- sort(deg)
degDF <- data.frame(station = names(deg), degree = deg)
degDF <- dplyr::tbl_df(degDF)
degreeDF <- left_join(degDF, station, by = c("station" = "name"))

ggplot(degreeDF, aes(x=station, y = degree, col = landmark)) +
  geom_point() +
  theme(text = element_text(size=6), axis.text.x = element_text(angle=90, vjust=1))+
  labs(x="Station", y= "Degree", col = "City", title = "BABS station degrees") +
  scale_x_discrete(limits = degreeDF$station)
```

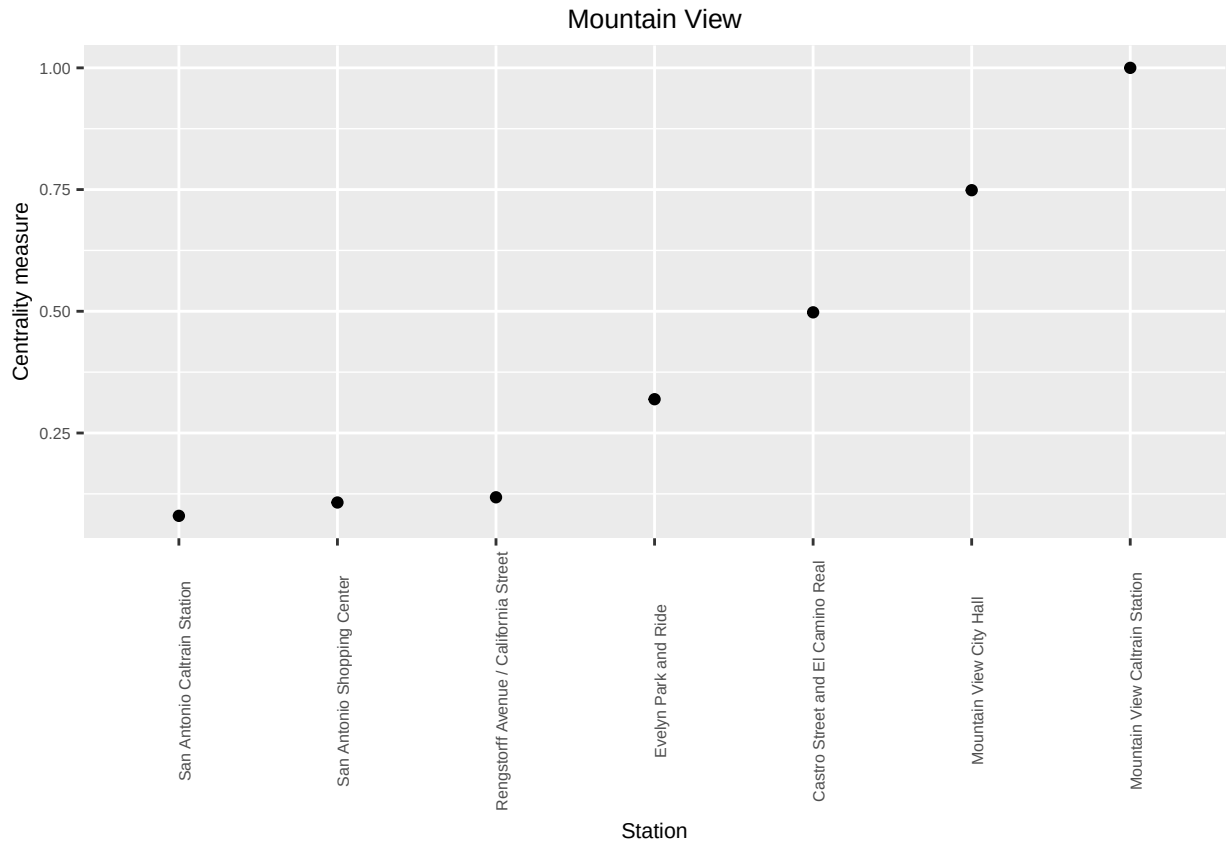


This is interesting but we can use a more discriminatory property.

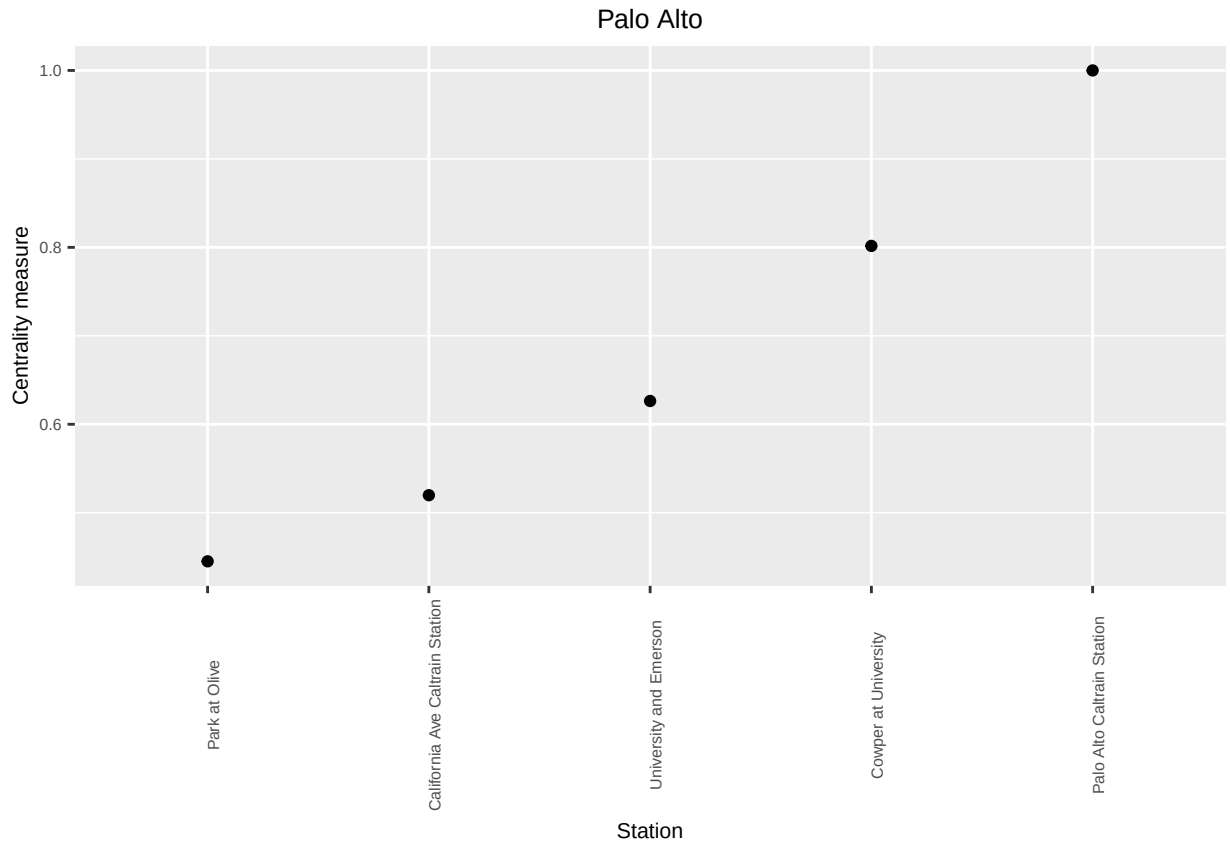
Eigenvalue centrality

We'll now split the network up and analyse the individual cities by calculating the eigenvalue centrality.

```
tripNumbers <- trip %>%
  filter(startLandmark == "Mountain View", endLandmark == "Mountain View") %>%
  group_by(Start.Station,End.Station) %>%
  summarise(weight = n())  %>%
  ungroup() %>%
  mutate(weight = (weight - min(weight))/(max(weight)-min(weight)))

bikeGraph <- graph_from_data_frame(tripNumbers, directed=TRUE)
bikeE <- eigen_centrality(bikeGraph, directed = TRUE)
bikeEigen <- data.frame(station = names(bikeE$vector), eig = bikeE$vector)
bikeEigen <- dplyr::tbl_df(bikeEigen)
bikeEigen <- mutate(bikeEigen, station = as.character(station))
bikeEigen <- arrange(bikeEigen, eig)
ggplot(bikeEigen, aes(x=station, y = eig))+
  geom_point()+
  theme(text = element_text(size=8), axis.text.x = element_text(angle=90, vjust=1))+
  labs(x="Station", y= "Centrality measure", title = "Mountain View")+
  scale_x_discrete(limits = bikeEigen$station)
```

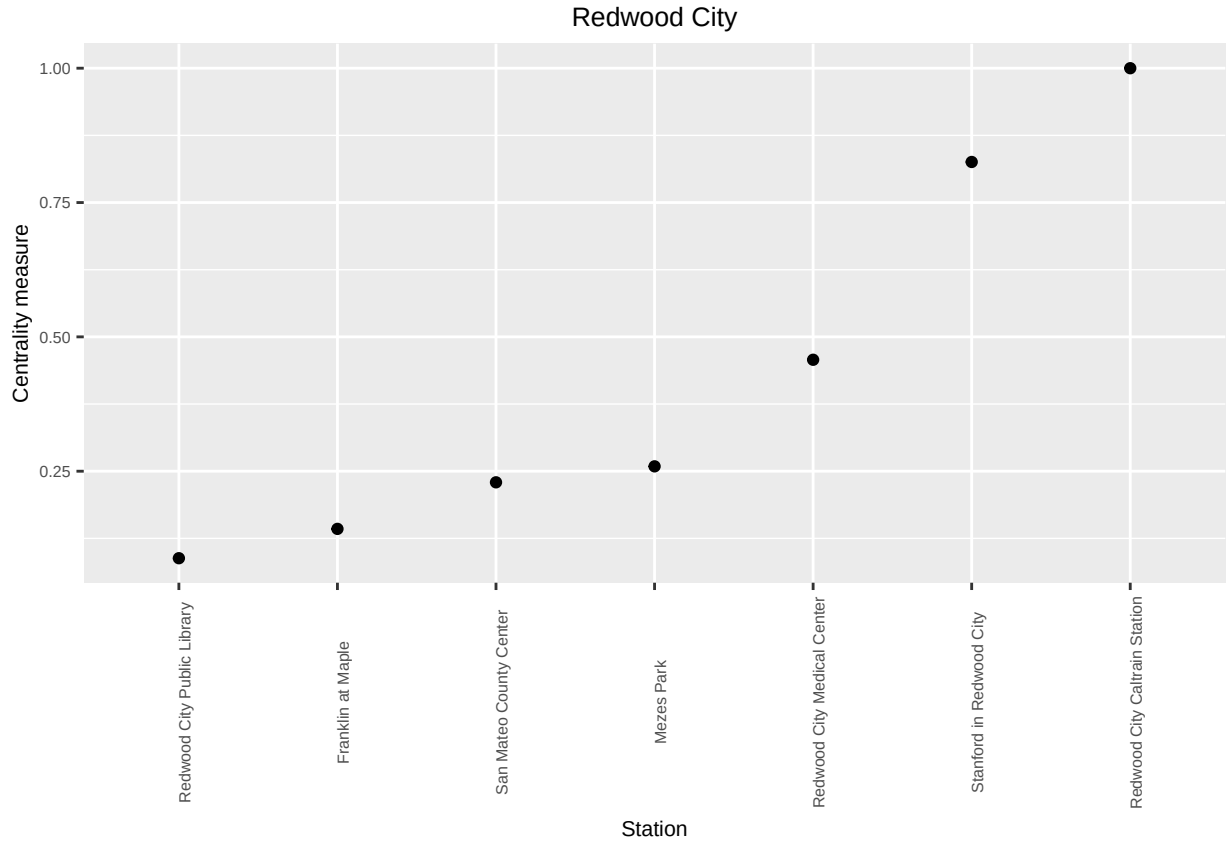


```
tripNumbers <- trip %>%
  filter(startLandmark == "Palo Alto", endLandmark == "Palo Alto") %>%
  group_by(Start.Station,End.Station) %>%
  summarise(weight = n())  %>%
  ungroup() %>%
  mutate(weight = (weight - min(weight))/(max(weight)-min(weight)))

bikeGraph <- graph_from_data_frame(tripNumbers, directed=TRUE)
bikeE <- eigen_centrality(bikeGraph, directed = TRUE)
bikeEigen <- data.frame(station = names(bikeE$vector), eig = bikeE$vector)
bikeEigen <- dplyr::tbl_df(bikeEigen)
bikeEigen <- mutate(bikeEigen, station = as.character(station))
bikeEigen <- arrange(bikeEigen, eig)
ggplot(bikeEigen, aes(x=station, y = eig))+
  geom_point()+
  theme(text = element_text(size=8), axis.text.x = element_text(angle=90, vjust=1))+
  labs(x="Station", y= "Centrality measure", title = "Palo Alto")+
  scale_x_discrete(limits = bikeEigen$station)
```



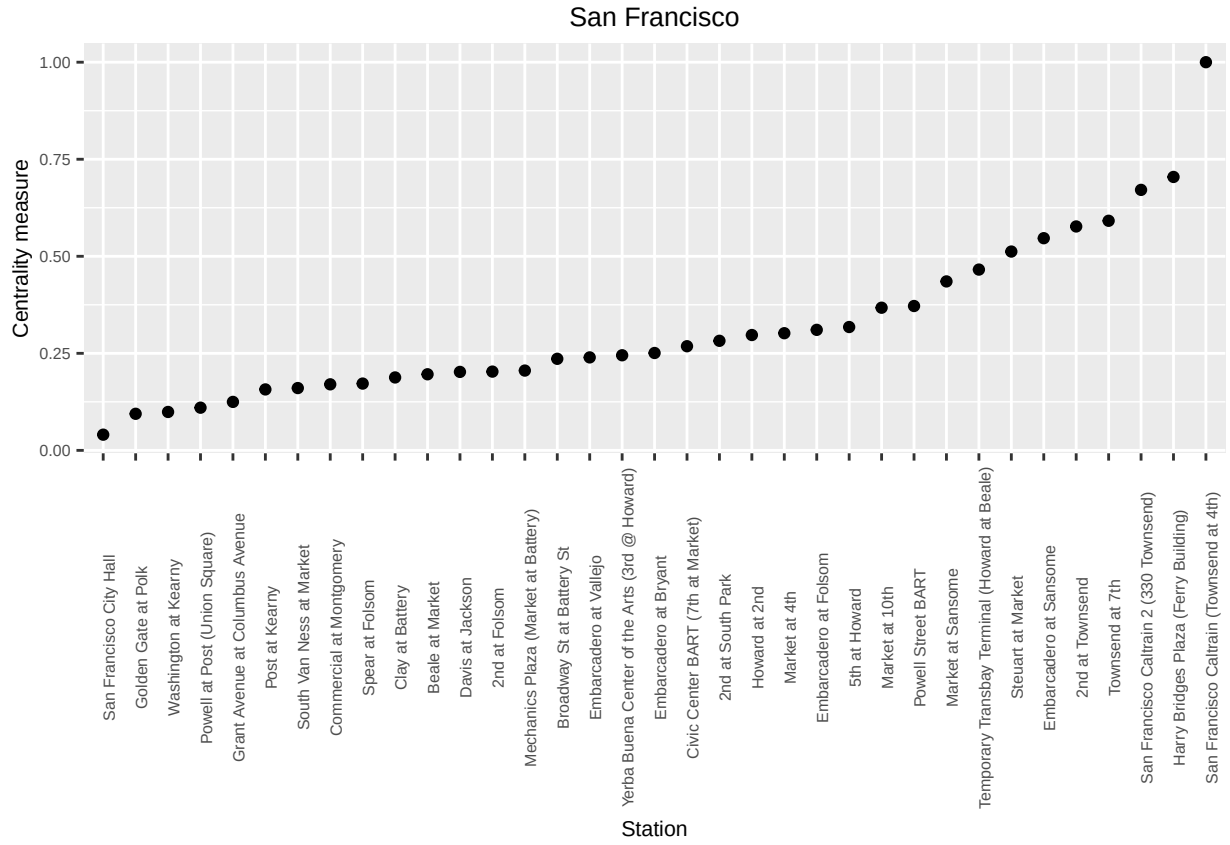

```
tripNumbers <- trip %>%
  filter(startLandmark == "Redwood City", endLandmark == "Redwood City") %>%
  group_by(Start.Station,End.Station) %>%
  summarise(weight = n())  %>%
  ungroup() %>%
  mutate(weight = (weight - min(weight))/(max(weight)-min(weight)))
```

```
bikeGraph <- graph_from_data_frame(tripNumbers, directed=TRUE)
bikeE <- eigen_centrality(bikeGraph, directed = TRUE)
bikeEigen <- data.frame(station = names(bikeE$vector), eig = bikeE$vector)
bikeEigen <- dplyr::tbl_df(bikeEigen)
bikeEigen <- mutate(bikeEigen, station = as.character(station))
bikeEigen <- arrange(bikeEigen, eig)
ggplot(bikeEigen, aes(x=station, y = eig))+
  geom_point()+
  theme(text = element_text(size=8), axis.text.x = element_text(angle=90, vjust=1))+
  labs(x="Station", y= "Centrality measure", title = "Redwood City")+
  scale_x_discrete(limits = bikeEigen$station)
```

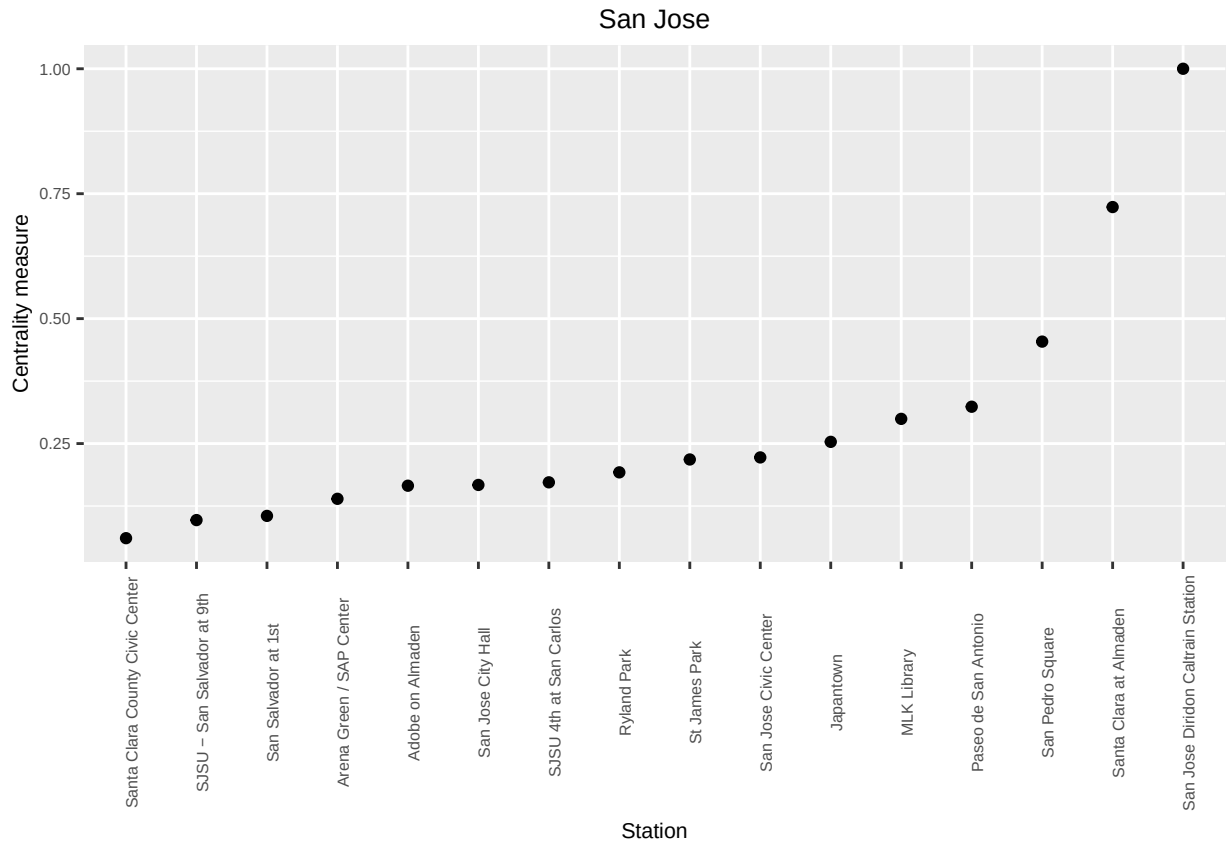



```
tripNumbers <- trip %>%
  filter(startLandmark == "San Francisco", endLandmark == "San Francisco") %>%
  group_by(Start.Station,End.Station) %>%
  summarise(weight = n())  %>%
  ungroup() %>%
  mutate(weight = (weight - min(weight))/(max(weight)-min(weight)))

bikeGraph <- graph_from_data_frame(tripNumbers, directed=TRUE)
bikeE <- eigen_centrality(bikeGraph, directed = TRUE)
bikeEigen <- data.frame(station = names(bikeE$vector), eig = bikeE$vector)
bikeEigen <- dplyr::tbl_df(bikeEigen)
bikeEigen <- mutate(bikeEigen, station = as.character(station))
bikeEigen <- arrange(bikeEigen, eig)
```

```
ggplot(bikeEigen, aes(x=station, y = eig))+
  geom_point()+
  theme(text = element_text(size=8), axis.text.x = element_text(angle=90, vjust=1))+
  labs(x="Station", y= "Centrality measure", title = "San Francisco")+
  scale_x_discrete(limits = bikeEigen$station)
```




```
tripNumbers <- trip %>%
  filter(startLandmark == "San Jose", endLandmark == "San Jose") %>%
  group_by(Start.Station,End.Station) %>%
  summarise(weight = n())  %>%
  ungroup() %>%
  mutate(weight = (weight - min(weight))/(max(weight)-min(weight)))

bikeGraph <- graph_from_data_frame(tripNumbers, directed=TRUE)
bikeE <- eigen_centrality(bikeGraph, directed = TRUE)
bikeEigen <- data.frame(station = names(bikeE$vector), eig = bikeE$vector)
bikeEigen <- dplyr::tbl_df(bikeEigen)
bikeEigen <- mutate(bikeEigen, station = as.character(station))
bikeEigen <- arrange(bikeEigen, eig)
ggplot(bikeEigen, aes(x=station, y = eig))+
  geom_point()+
  theme(text = element_text(size=8), axis.text.x = element_text(angle=90, vjust=1))+
  labs(x="Station", y= "Centrality measure", title = "San Jose")+
  scale_x_discrete(limits = bikeEigen$station)
```



Interestingly we see that the busiest station in each city is located by a train station. This is important if we want to expand into new cities as people seem to be using them as part of their commute. We can investigate this further by analysing the types of trips being taken.

```
tripTypes <- trip  %>%
 group_by(Start.Station, End.Station) %>%
 summarise(count =n()) %>%
 ungroup() %>%
 arrange(desc(count)) %>%
 left_join(station, by = c("Start.Station" = "name"))
```

We can look at the most common trips in each city

```
tripTypes  %>%
  filter(landmark == "Mountain View")  %>%
  select(Start.Station, End.Station, count)  %>%
  mutate(Start.Station = abbreviate(Start.Station, 30),
         End.Station = abbreviate(End.Station, 30))%>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##                    Start.Station                   End.Station count
##                            <chr>                         <chr> <int>
## 1  Mountain View Caltrain Station       Mountain View City Hall  1419
## 2         Mountain View City Hall Mountain View Caltrain Station  1308
```

```
## 3     CastroStreetandElCaminoReal Mountain View Caltrain Station  1041
## 4  Mountain View Caltrain Station    CastroStreetandElCaminoReal   940
## 5            Evelyn Park and Ride Mountain View Caltrain Station   808
## 6    San Antonio Caltrain Station    San Antonio Shopping Center   768
## 7     San Antonio Shopping Center   San Antonio Caltrain Station   740
## 8  Mountain View Caltrain Station           Evelyn Park and Ride   608
## 9  Mountain View Caltrain Station Mountain View Caltrain Station   254
## 10 RengstorffAvenu/CalifornStreet Mountain View Caltrain Station   218
```

```
tripTypes  %>%
  filter(landmark == "Palo Alto")  %>%
  select(Start.Station, End.Station, count)  %>%
  mutate(Start.Station = abbreviate(Start.Station, 35),
         End.Station = abbreviate(End.Station, 35))%>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##                      Start.Station                     End.Station count
##                              <chr>                           <chr> <int>
## 1       Palo Alto Caltrain Station            Cowper at University   387
## 2             Cowper at University      Palo Alto Caltrain Station   377
## 3           University and Emerson          University and Emerson   348
## 4       Palo Alto Caltrain Station                    Park at Olive   241
## 5       Palo Alto Caltrain Station California Ave Caltrain Station   215
## 6                    Park at Olive      Palo Alto Caltrain Station   174
## 7       Palo Alto Caltrain Station      Palo Alto Caltrain Station   162
## 8  California Ave Caltrain Station          University and Emerson   101
## 9  California Ave Caltrain Station      Palo Alto Caltrain Station    95
## 10 California Ave Caltrain Station California Ave Caltrain Station    85
```

```
tripTypes  %>%
  filter(landmark == "Redwood City")  %>%
  select(Start.Station, End.Station, count)  %>%
  mutate(Start.Station = abbreviate(Start.Station, 35),
         End.Station = abbreviate(End.Station, 35)) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##                     Start.Station                   End.Station count
##                             <chr>                         <chr> <int>
## 1       Stanford in Redwood City Redwood City Caltrain Station   337
## 2  Redwood City Caltrain Station      Stanford in Redwood City   332
## 3  Redwood City Caltrain Station   Redwood City Medical Center   202
## 4                    Mezes Park Redwood City Caltrain Station   175
## 5  Redwood City Caltrain Station                    Mezes Park   112
## 6    Redwood City Medical Center Redwood City Caltrain Station   109
## 7  Redwood City Caltrain Station       San Mateo County Center    74
## 8  Redwood City Caltrain Station Redwood City Caltrain Station    70
## 9  Redwood City Caltrain Station             Franklin at Maple    61
## 10      Stanford in Redwood City      Stanford in Redwood City    49
```

```
tripTypes  %>%
  filter(landmark == "San Francisco")  %>%
  select(Start.Station, End.Station, count)  %>%
  mutate(Start.Station = abbreviate(Start.Station, 30),
         End.Station = abbreviate(End.Station, 30)) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##                      Start.Station                     End.Station count
##                              <chr>                           <chr> <int>
## 1  SanFranciscCaltran2(330Twnsnd)                Townsend at 7th  3748
## 2  HarryBridgesPlaz(FerryBuildng)          Embarcadero at Sansome  3145
## 3                 2nd at Townsend HarryBridgesPlaz(FerryBuildng)  2973
## 4                 Townsend at 7th SanFranciscCaltran2(330Twnsnd)  2734
## 5  HarryBridgesPlaz(FerryBuildng)                2nd at Townsend  2640
## 6           Embarcadero at Folsom SanFranciscCaltrn(Twnsndat4th)  2439
## 7               Steuart at Market                2nd at Townsend  2356
## 8          Embarcadero at Sansome              Steuart at Market  2330
## 9                 Townsend at 7th SanFranciscCaltrn(Twnsndat4th)  2192
## 10 TemporaryTrnsbyTrmnl(HwrdatBl) SanFranciscCaltrn(Twnsndat4th)  2184
```

```
tripTypes  %>%
  filter(landmark == "San Jose")  %>%
  select(Start.Station, End.Station, count)  %>%
  mutate(Start.Station = abbreviate(Start.Station, 30),
         End.Station = abbreviate(End.Station, 30)) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##                     Start.Station                   End.Station count
##                             <chr>                         <chr> <int>
## 1  SanJoseDiridonCaltrainStation        Santa Clara at Almaden  1232
## 2         Santa Clara at Almaden SanJoseDiridonCaltrainStation  1214
## 3  SanJoseDiridonCaltrainStation              San Pedro Square   687
## 4                   MLK Library SanJoseDiridonCaltrainStation   668
## 5               San Pedro Square SanJoseDiridonCaltrainStation   614
## 6  SanJoseDiridonCaltrainStation          Paseo de San Antonio   497
## 7  SanJoseDiridonCaltrainStation                   MLK Library   461
## 8                   Ryland Park SanJoseDiridonCaltrainStation   395
## 9                 St James Park SanJoseDiridonCaltrainStation   335
## 10 SanJoseDiridonCaltrainStation                St James Park   318
```

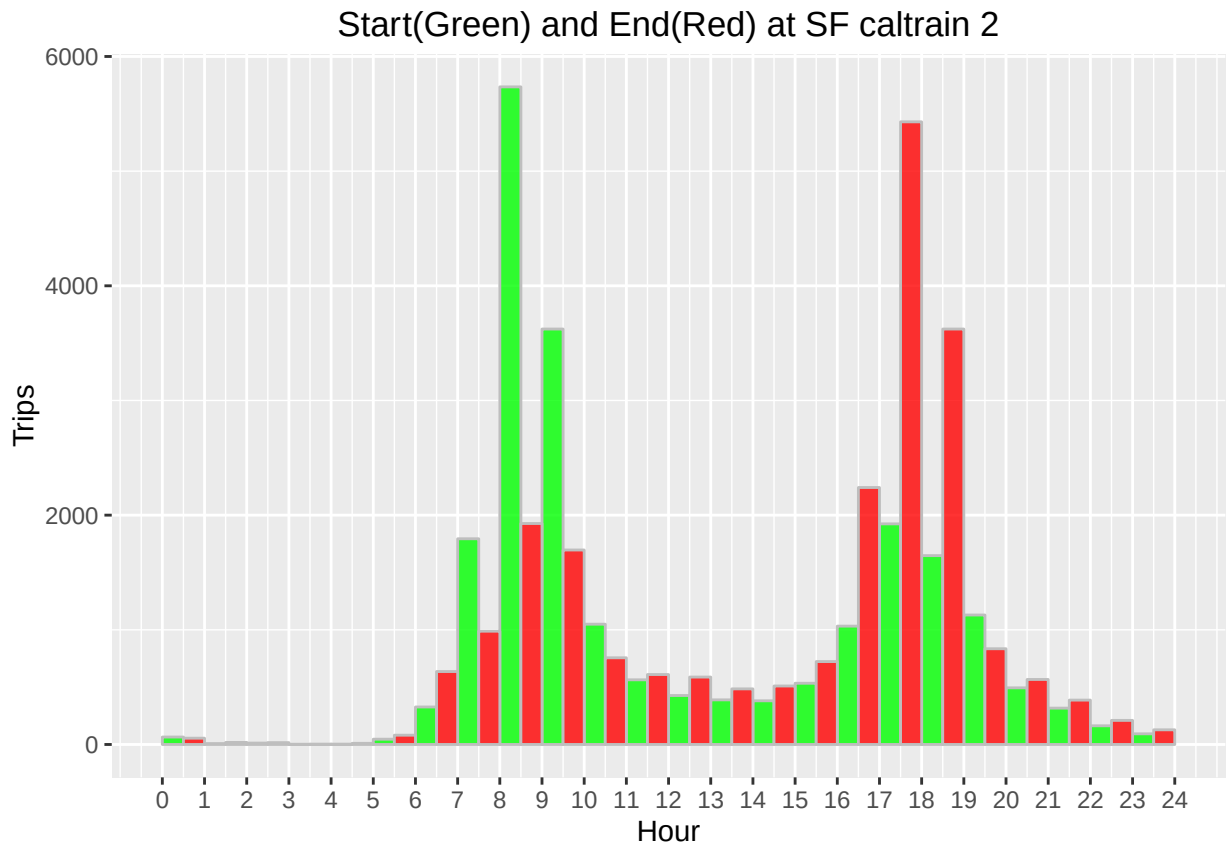
From which we see how integral the train stations are to this bike network. Who is taking these trips? Is it subscribers?

```
trip %>% filter(Start.Station == "San Francisco Caltrain 2 (330 Townsend)") %>%
  group_by(Subscriber.Type) %>% summarise(count = n())
```

```
## # A tibble: 2 x 2
##   Subscriber.Type count
##             <chr> <int>
## 1        Customer   835
## 2      Subscriber 20923
```

We see it is mainly subscribers using this station. What time of day is it being used?

```
trip_per_hour_start <- trip %>%
  filter(Start.Station == "San Francisco Caltrain 2 (330 Townsend)") %>%
  mutate(sHour = hour(Start_trip))

trip_per_hour_end <- trip %>%
  filter(End.Station == "San Francisco Caltrain 2 (330 Townsend)") %>%
  mutate(eHour = hour(End_trip))

ggplot(trip_per_hour_start, aes(x = sHour))+
geom_bar(col = 'grey', fill = "green", alpha = 0.8,
         width = 0.5, position = position_nudge(x = 0.25))+
geom_bar(data = trip_per_hour_end, aes(x=eHour),
         col = 'grey', fill = "red", alpha = 0.8, width = 0.5,
         position = position_nudge(x = 0.75))+
labs(x= "Hour", y="Trips", title = "Start(Green) and End(Red) at SF caltrain 2")+
  scale_x_continuous(breaks =seq(0,24,1))
```



From which we see it it mainly at the hours of people arriving for work and leaving from work.

Where do they live? we can use the zipcode information to answer this combined with the zipcode data and library package.

```
data("zipcode")
zips <- trip %>%
```

```
  filter(Start.Station == "San Francisco Caltrain 2 (330 Townsend)") %>%
  group_by(Zip.Code) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  mutate(zip = clean.zipcodes(Zip.Code)) %>%
  merge(zipcode, by.x='zip', by.y='zip') %>%
  arrange(desc(count))

head(zips)
```

```
##     zip Zip.Code count          city state latitude longitude
## 1 94107    94107  5192 San Francisco    CA 37.76653 -122.3958
## 2 94103    94103   896 San Francisco    CA 37.77233 -122.4109
## 3 94403    94403   663     San Mateo    CA 37.53969 -122.3027
## 4 94102    94102   617 San Francisco    CA 37.77933 -122.4192
## 5 94402    94402   490     San Mateo    CA 37.55159 -122.3277
## 6 94010    94010   487    Burlingame    CA 37.57494 -122.3633
```

But note there is one subscriber from as far as Hawaii!

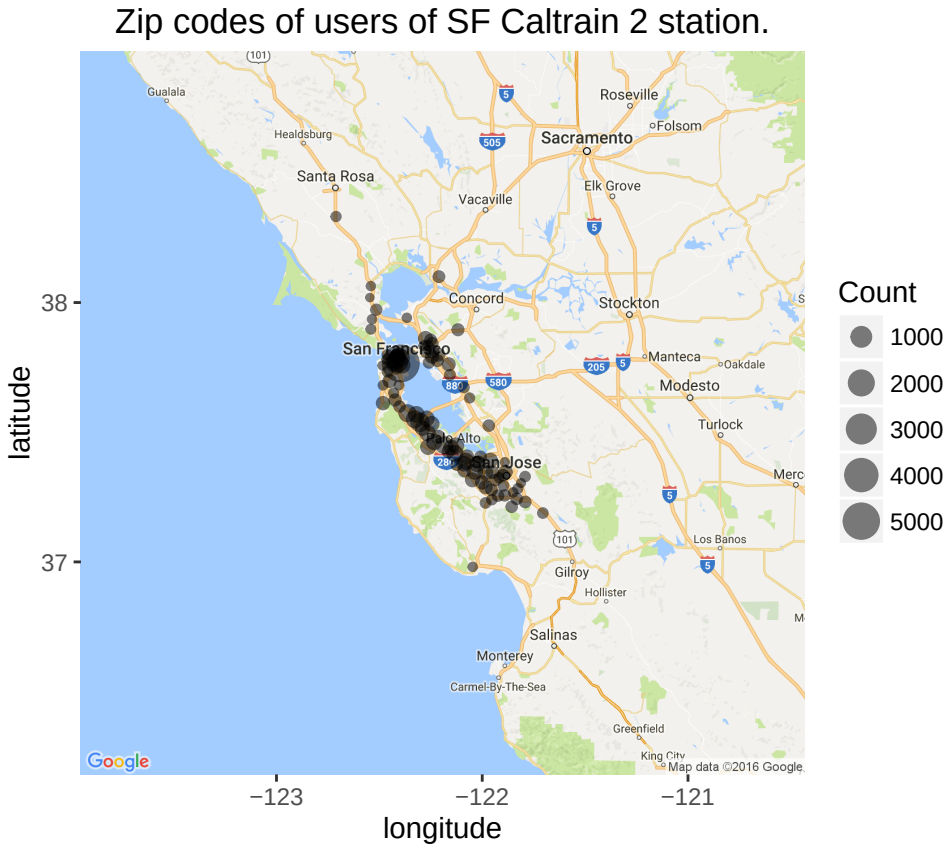
```
filter(zips, state == "HI")
```

```
##     zip Zip.Code count     city state latitude longitude
## 1 96818    96818     1 Honolulu    HI 21.36425 -157.9632
```

(I doubt they are commuting though)

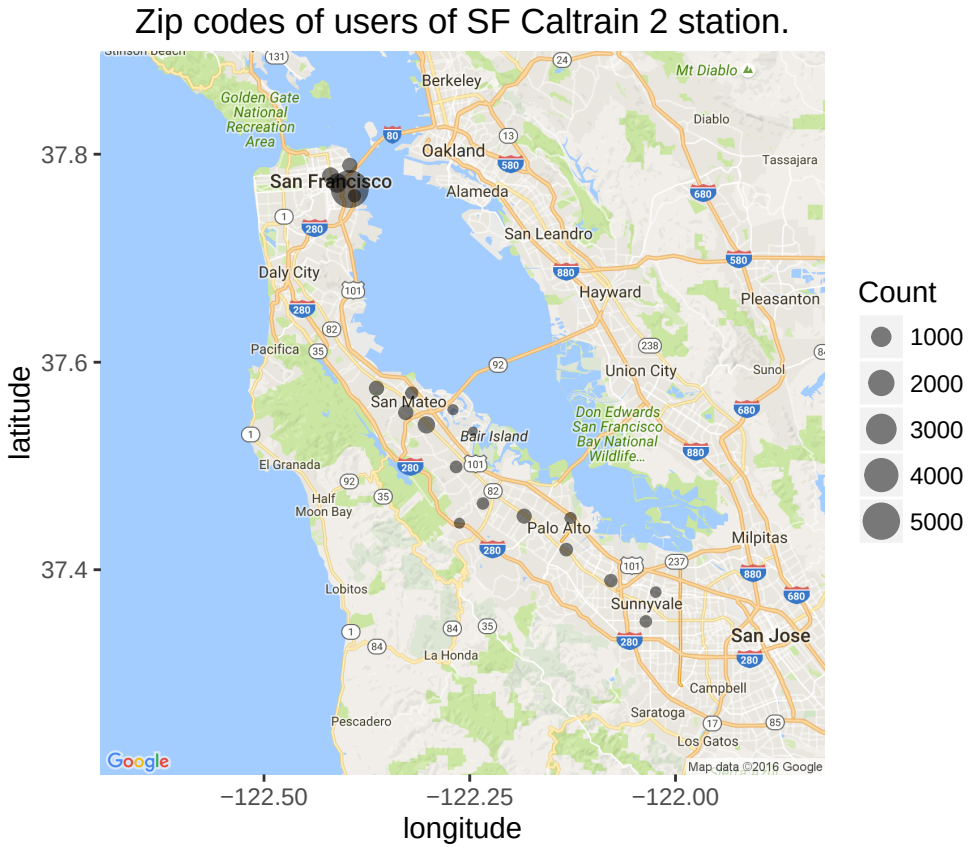
Let's look at the top 100 zipcodes which are all in California.

```
zipsShort <- slice(zips, 1:100)
location <- c( mean(zipsShort$longitude), mean(zipsShort$latitude))
BABSmap <- get_map(location,maptype="roadmap", zoom=8)
ggmap(BABSmap)+
  geom_point(data = zipsShort, aes(x=longitude, y=latitude, size = count), alpha = 0.5)+
  labs(x = "longitude", y = "latitude", size = "Count",
       title = "Zip codes of users of SF Caltrain 2 station.")
```



People travel from afar to use this station. Let's look more specifically by choosing the top 20 zipcodes.

```
zipsShorter <- slice(zips, 1:20)
location <- c( mean(zipsShorter$longitude), mean(zipsShorter$latitude))
BABSmap <- get_map(location,maptype="roadmap", zoom=10)
ggmap(BABSmap)+
  geom_point(data = zipsShorter, aes(x=longitude, y=latitude, size = count), alpha = 0.5)+
  labs(x = "longitude", y = "latitude", size = "Count",
       title = "Zip codes of users of SF Caltrain 2 station.")
```



Clusters

Let's try to find communities within the city of San Francisco using the infomap algorithm

Here is our network:

```
tripNumbers <- trip %>%
  filter(startLandmark == "San Francisco", endLandmark == "San Francisco") %>%
  group_by(Start.Station,End.Station) %>%
  summarise(weight = n())  %>%
  ungroup() %>%
  mutate(weight = (weight - min(weight)+0.0000001)/(max(weight)-min(weight)))

stationSF <- station  %>% filter(landmark == "San Francisco") %>% arrange(name)

bikeGraph <- graph_from_data_frame(tripNumbers, directed=TRUE)
location <- c( mean(stationSF$long), mean(stationSF$lat))
BABSmap <- get_map(location,maptype="roadmap", zoom=14)
```

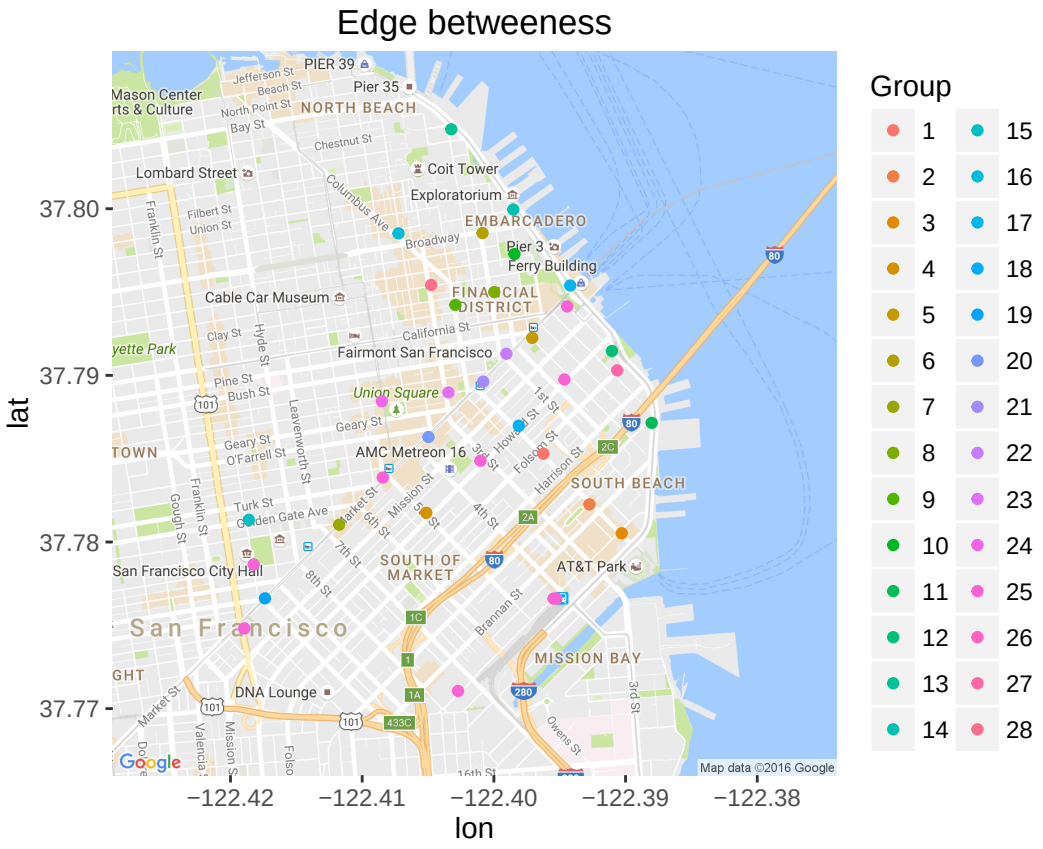
Let's look at some clustering algorithms

```
clus_eb <- cluster_edge_betweenness(bikeGraph)
clus_op <- cluster_optimal(bikeGraph)
clus_sg <- cluster_spinglass(bikeGraph)
clus_wt <- cluster_walktrap(bikeGraph)
```

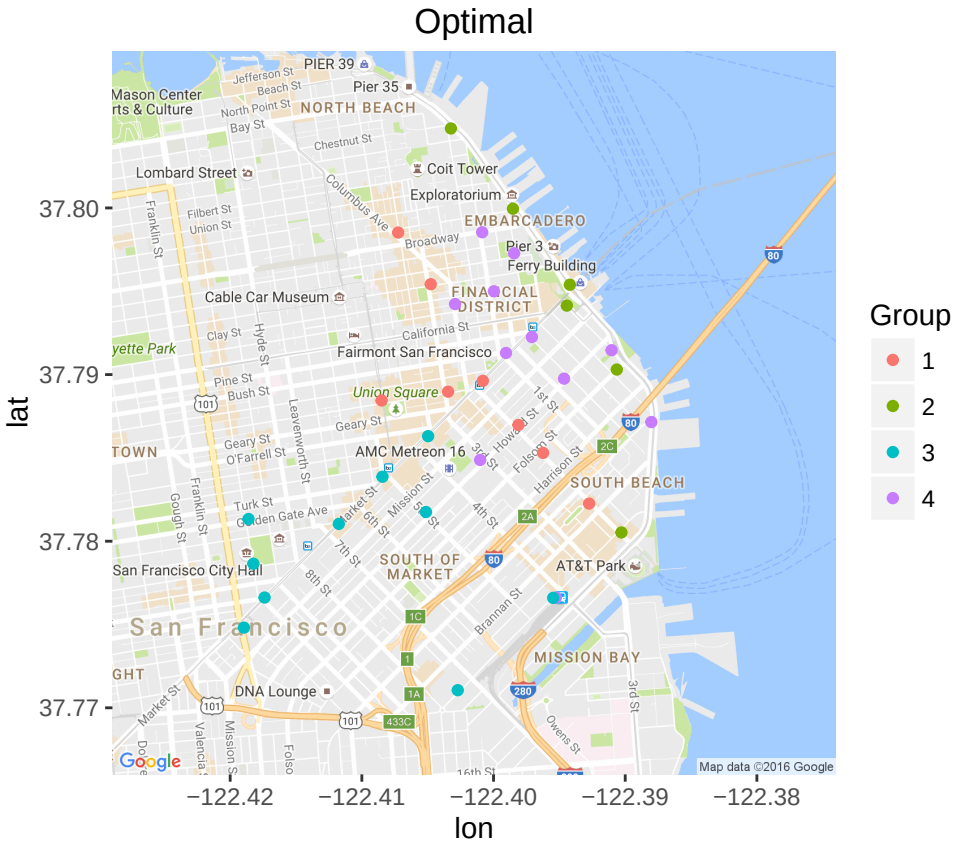
```
stationSF$eb <-clus_eb$membership
stationSF$op <-clus_op$membership
stationSF$sg <-clus_sg$membership
stationSF$wt <-clus_wt$membership
```

and see what results they give us

```
ggmap(BABSmap)+
  geom_point(data = stationSF,
             aes( y= lat,
                  x =long,
                  col = as.factor(eb)))+
  labs(title = "Edge betweeness", col ="Group")
```

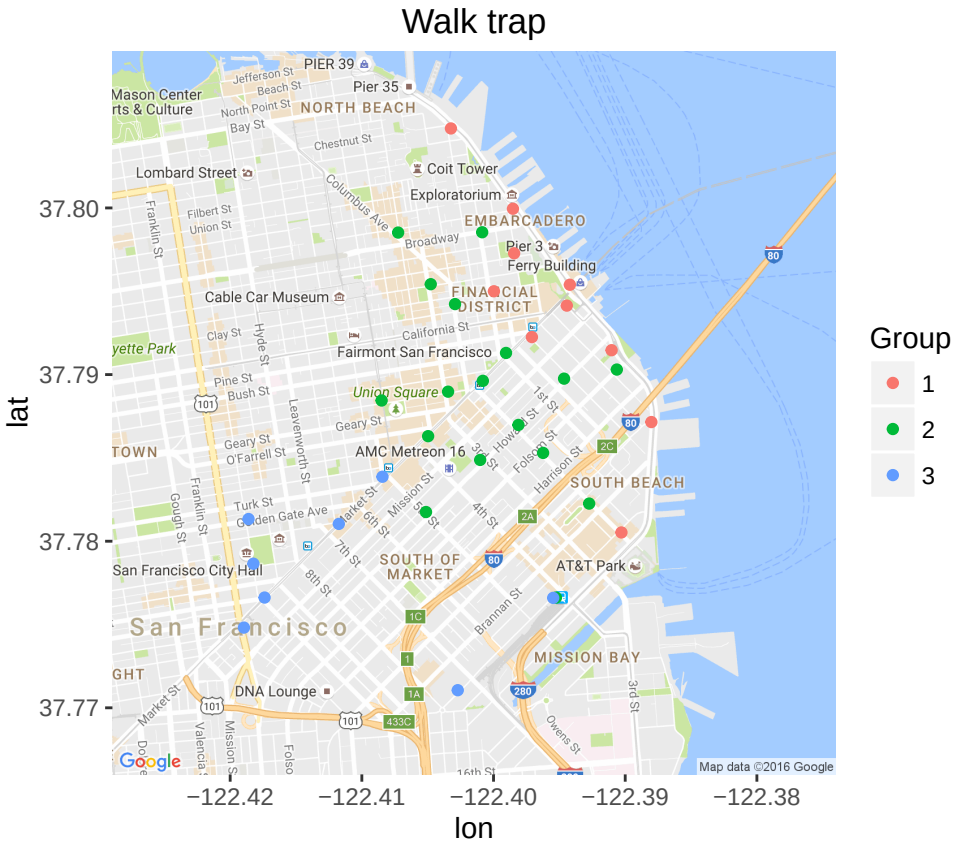



```
ggmap(BABSmap)+
  geom_point(data = stationSF,
             aes( y= lat,
                  x =long,
                  col = as.factor(op)))+
  labs(title = "Optimal", col ="Group")
```




```
ggmap(BABSmap)+
  geom_point(data = stationSF,
             aes( y= lat,
                  x =long,
                  col = as.factor(sg)))+
  labs(title = "Spin glass", col ="Group")
```


```
ggmap(BABSmap)+
  geom_point(data = stationSF,
             aes( y= lat,
                  x =long,
                  col = as.factor(wt)))+
  labs(title = "Walk trap", col ="Group")
```



The useful clusterings seem to be mainly correlated with location, indicating lots of short trips within these clusters.