Final code

Neeraj Gupta

2/18/2021

Data Processing Code

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

#install.packages("rlang")  
#install.packages( "ggplot2",repos = c("http://rstudio.org/\_packages", "http://cran.rstudio.com"))  
  
library(ggplot2)  
library(ggmap)

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.

library(ggrepel)  
library(plyr)  
library(sqldf)

## Loading required package: gsubfn

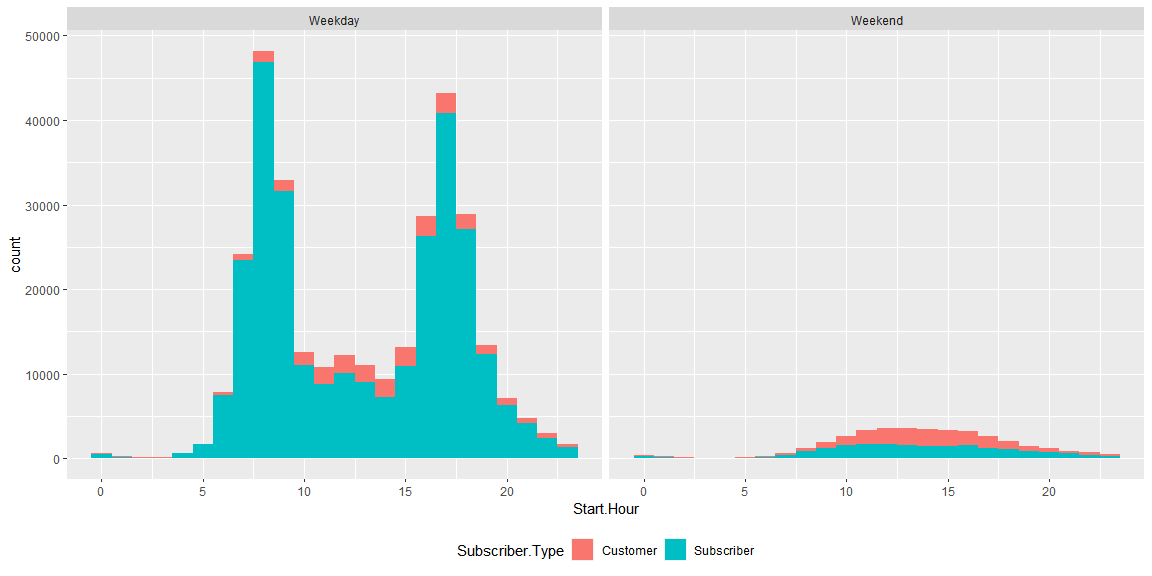
## Loading required package: proto

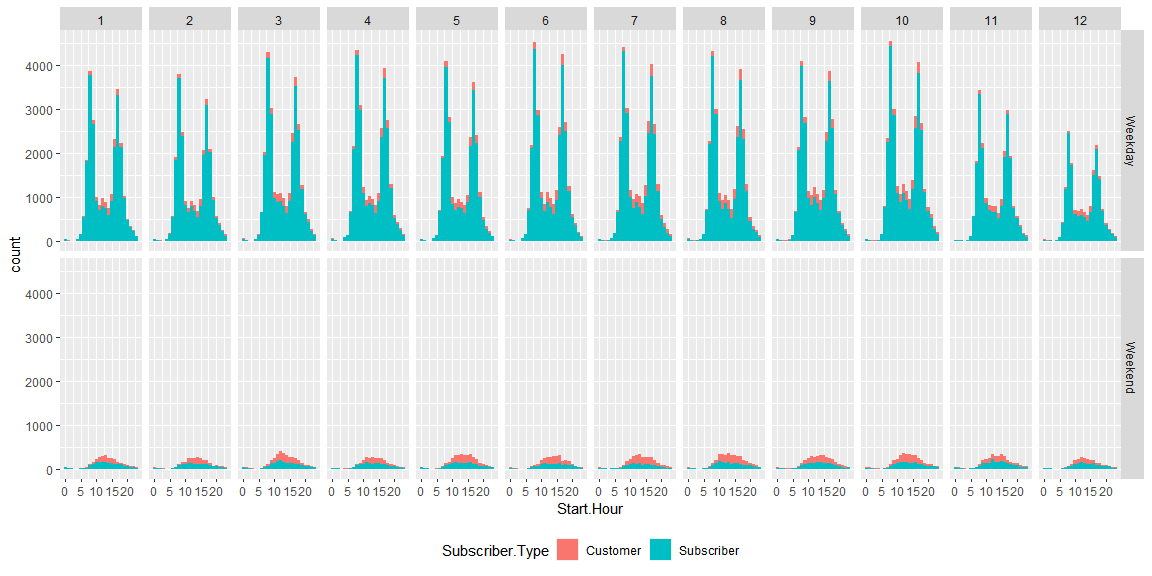
## Loading required package: RSQLite

library(knitr)

getTrips <- function(source\_file){  
 trips = read.csv(source\_file)  
  
 trips$Trip.ID <- as.factor(trips$Trip.ID)  
 trips$Start.Date <- mdy\_hm(as.character(trips$Start.Date))  
 trips$Start.Day <- as.Date(trips$Start.Date)  
  
 trips$Start.Hour <- hour(trips$Start.Date)  
 trips$DayPart <- "1. Morning"  
 trips[trips$Start.Hour > 11,]$DayPart <- "2. Afternoon"  
 trips[trips$Start.Hour > 17,]$DayPart <- "3. Evening"  
 trips$DayPart <- as.factor(trips$DayPart)  
  
 trips$Start.Minute <- minute(trips$Start.Date)  
 trips$Start.Month.Number <- as.numeric(month(trips$Start.Date))  
 trips$Start.Month <- paste("(",trips$Start.Month.Number,")"," ", months(trips$Start.Date), sep="")  
 trips$End.Date <- mdy\_hm(as.character(trips$End.Date))  
 trips$Start.Terminal <- as.factor(trips$Start.Terminal)  
 trips$End.Terminal <- as.factor(trips$End.Terminal)  
 trips$Bike <- as.factor(trips[,"Bike.."])  
 trips <- trips[ , !(names(trips) %in% c("Bike.."))]  
 trips$Weekday <- as.factor(wday(trips$Start.Date))  
 trips$DayType <- "Weekday"  
 trips[trips$Weekday %in% c(1,7),]$DayType <- "Weekend"  
 trips$DayType <- as.factor(trips$DayType)  
 trips$row\_weight = 1  
  
 trips  
}  
  
getStations <- function(source\_file){  
 stations = read.csv(source\_file)  
  
 stations$id <- as.factor(stations$station\_id)  
 stations$amount\_out <- apply(stations, 1, function(station) nrow(trips[trips$Start.Terminal == station['id'],]))  
 stations$amount\_in <- apply(stations, 1, function(station) nrow(trips[trips$End.Terminal == station['id'],]))  
 stations$traffic <- stations$amount\_in + stations$amount\_out  
 stations$amount\_net <- stations$amount\_in - stations$amount\_out  
 stations$amount\_net\_percentage <- 100\*stations$amount\_net/(stations$amount\_in + stations$amount\_out)  
 stations <- stations[order(-stations$traffic),]  
  
 stations  
}  
  
# this function was taken from   
# http://stackoverflow.com/questions/27418461/calculate-the-modes-in-a-multimodal-distribution-in-r  
find\_modes <- function(x) {  
 modes <- NULL  
 for ( i in 2:(length(x)-1) ){  
 if ( (x[i] > x[i-1]) & (x[i] > x[i+1]) ) {  
 modes <- c(modes,i)  
 }  
 }  
 if ( length(modes) == 0 ) {  
 modes = 'This is a monotonic distribution'  
 }  
 return(modes)  
}  
plotBikeUsageHistogram <- function(bikes){  
 md <- find\_modes(density(bikes$freq)$y)  
  
 ggp <- ggplot(bikes, aes(freq)) + geom\_histogram(binwidth = 10)   
 ggp <- ggp + geom\_vline(xintercept = density(bikes$freq)$x[md][1], color = "red", linetype="dashed")  
 ggp <- ggp + geom\_vline(xintercept = density(bikes$freq)$x[md][2], color = "red", linetype="dashed")  
   
 ggp  
}  
  
getBikesPosition <- function(trips){  
 bikes\_usage <- sqldf("select Bike, max(`End.Date`) as dt, count(\*) as used from trips group by Bike")  
 bikes <- sqldf("select t.\*, bu.used from bikes\_usage bu inner join trips t on t.`End.Date` = bu.dt and t.Bike = bu.Bike order by bu.used desc")  
  
 bikes  
}  
getTripsNet <- function(trips){  
 trips\_last\_day <- trips[trips$Start.Day == max(trips$Start.Day),]  
 trips\_aggregated <- sqldf("select `Start.Terminal`,`End.Terminal`, count(\*) as freq from trips\_last\_day where `Start.Terminal`!= `End.Terminal` group by `Start.Terminal`, `End.Terminal` ")  
 trips\_net <- sqldf("  
 select Terminal, sum(amount) as amount from (  
 select `End.Terminal` as Terminal, sum(freq) as amount from trips\_aggregated group by `End.Terminal`  
 union all   
 select `Start.Terminal` as Terminal, -1\*sum(freq) as amount from trips\_aggregated group by `Start.Terminal`  
 )   
 group by Terminal  
 having sum(amount)!=0  
 ")  
 trips\_net  
}  
getTransferBikesRecommendations <- function(trips\_net, bikes, stations){  
 transfer\_bikes <- sqldf("select s.id as Terminal, s.name as Station , s.traffic, tn.amount from stations s left join trips\_net tn on s.id = tn.Terminal where tn.amount !=0 ")  
 transfer\_bikes$amount <- as.numeric(transfer\_bikes$amount)  
   
 transfer\_bikes$Recommendation <- apply(transfer\_bikes, 1, function(station){  
 station['traffic']<- as.numeric(station['traffic'])  
 station\_bikes <- bikes[bikes$End.Terminal == station['Terminal'],]  
 recommendation <- ""  
 abs\_amount <- abs(as.numeric(station['amount']))  
 if (as.numeric(station['traffic']) > as.numeric(median(transfer\_bikes$traffic))){  
 if (as.numeric(station['amount'])>0){  
 recommendation <- paste("Take off: ",paste(head(station\_bikes,abs\_amount)$Bike, collapse=", "),sep="")  
 }else{  
 recommendation <- paste("Bring ",abs(as.numeric(station['amount']))," bikes used few times",sep="")  
 }  
 }else{  
 if (as.numeric(station['amount'])>0){  
 recommendation <- paste("Take off: ",paste(tail(station\_bikes,abs\_amount)$Bike, collapse=", "),sep="")  
 }else{  
 recommendation <- paste("Bring ",abs(as.numeric(station['amount']))," heavily used bikes",sep="")  
 }  
 }  
 recommendation  
 })  
 transfer\_bikes$Recommendation <- as.factor(transfer\_bikes$Recommendation)  
   
 transfer\_bikes  
}  
  
getAggregatedTrips <- function(trips){  
 # Aggregate trips by Terminals, Subscriber and Daytime  
 trips\_daily = count(trips[trips$End.Terminal != trips$Start.Terminal,], .(Start.Terminal, End.Terminal, Subscriber.Type, DayType, DayPart))  
  
 # Augment the aggregated trips with geo positions of stations  
 trips\_daily$Start.Station.long <- apply(trips\_daily, 1, function(trip) stations[stations$id == trip['Start.Terminal'],'long'])  
 trips\_daily$Start.Station.lat <- apply(trips\_daily, 1, function(trip) stations[stations$id == trip['Start.Terminal'],'lat'])  
 trips\_daily$End.Station.long <- apply(trips\_daily, 1, function(trip) stations[stations$id == trip['End.Terminal'],'long'])  
 trips\_daily$End.Station.lat <- apply(trips\_daily, 1, function(trip) stations[stations$id == trip['End.Terminal'],'lat'])  
 trips\_daily$Start.City <- apply(trips\_daily, 1, function(trip) stations[stations$id == trip['Start.Terminal'],'landmark'])  
 trips\_daily$End.City <- apply(trips\_daily, 1, function(trip) stations[stations$id == trip['End.Terminal'],'landmark'])  
  
 # Augment the aggregated trips with the direction of trips  
 trips\_daily$Direction <- "North"  
 trips\_daily[trips\_daily$End.Station.lat < trips\_daily$Start.Station.lat,]$Direction <- "South"  
 trips\_daily$Direction <- as.factor(trips\_daily$Direction)  
  
 trips\_daily  
}   
getFlowPlot <- function(area\_map, trips\_daily, stations, show\_labels = TRUE){  
 FlowPlot <- ggmap(area\_map, extent = "device", ylab = "Latitude", xlab = "Longitude",darken = 0.75) +   
 geom\_curve(data = trips\_daily, aes(x=Start.Station.long, xend=End.Station.long, y=Start.Station.lat, yend=End.Station.lat, color= Direction, alpha = freq, size = freq), curvature = 0.05, inherit.aes = TRUE) +  
 facet\_grid(DayPart ~ Subscriber.Type) +  
 scale\_alpha(limits=c(0, max(trips\_daily$freq)), guide=FALSE) +  
 scale\_size(limits=c(0, max(trips\_daily$freq)), guide=FALSE) + theme(legend.position="bottom")+coord\_cartesian()  
   
 if (show\_labels == TRUE){  
 FlowPlot <- FlowPlot + geom\_text\_repel(data=stations, aes(long, lat, label = name, alpha = traffic), color = 'white', size =2, segment.color = 'white')  
 }  
  
 FlowPlot  
}  
getStationsBalancePlot <- function(area\_map, stations){  
 ggmap(area\_map, extent = "device", ylab = "Latitude", xlab = "Longitude",darken = 0.75) +   
 geom\_point(data = stations, aes(x=long, y=lat, size = traffic), shape = 1, color = "white", )+  
 scale\_fill\_gradientn(colors=c("#f9ed32", "#ee2a7b","#002aff"),limits=c(-35, 35), name="Traffic in over out, %")+  
 scale\_shape\_discrete(solid=F) +  
 geom\_label\_repel(  
 data=stations,  
 aes(long, lat, fill = stations$amount\_net\_percentage, label = paste(name, round(stations$amount\_net\_percentage,0),"%", sep=" ")),  
 fontface = 'bold', color = 'white',  
 box.padding = unit(0.25, "lines"),  
 point.padding = unit(0.5, "lines")  
 ) + theme(legend.position="bottom")  
}

trips <- getTrips("C:/Users/Pravin/Documents/MS in Analytics/ANALY 500 - Principles & Applications/Project/201508\_trip\_data.csv")





stations <- getStations("C:/Users/Pravin/Documents/MS in Analytics/ANALY 500 - Principles & Applications/Project/201508\_station\_data.csv")

# Get maps of areas  
library(ggmap)  
library(ggplot2)  
#if(!requireNamespace("devtools")) install.packages("devtools")  
#devtools::install\_github("dkahle/ggmap")   
register\_google(key = "XYZ")  
  
MapBayArea <- get\_map(location = c(mean(stations$long),mean(stations$lat)), zoom = 10, color = "bw")

## Source : https://maps.googleapis.com/maps/api/staticmap?center=37.590243,-122.218416&zoom=10&size=640x640&scale=2&maptype=terrain&language=en-EN&key=xxx

MapSanFrancisco <- get\_map("Union Square, San Francisco", zoom = 14, color = "bw")

## Source : https://maps.googleapis.com/maps/api/staticmap?center=Union%20Square,%20San%20Francisco&zoom=14&size=640x640&scale=2&maptype=terrain&language=en-EN&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Union+Square,+San+Francisco&key=xxx

MapPaloAlto <- get\_map("Palo Alto", zoom = 12, color = "bw")

## Source : https://maps.googleapis.com/maps/api/staticmap?center=Palo%20Alto&zoom=12&size=640x640&scale=2&maptype=terrain&language=en-EN&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Palo+Alto&key=xxx

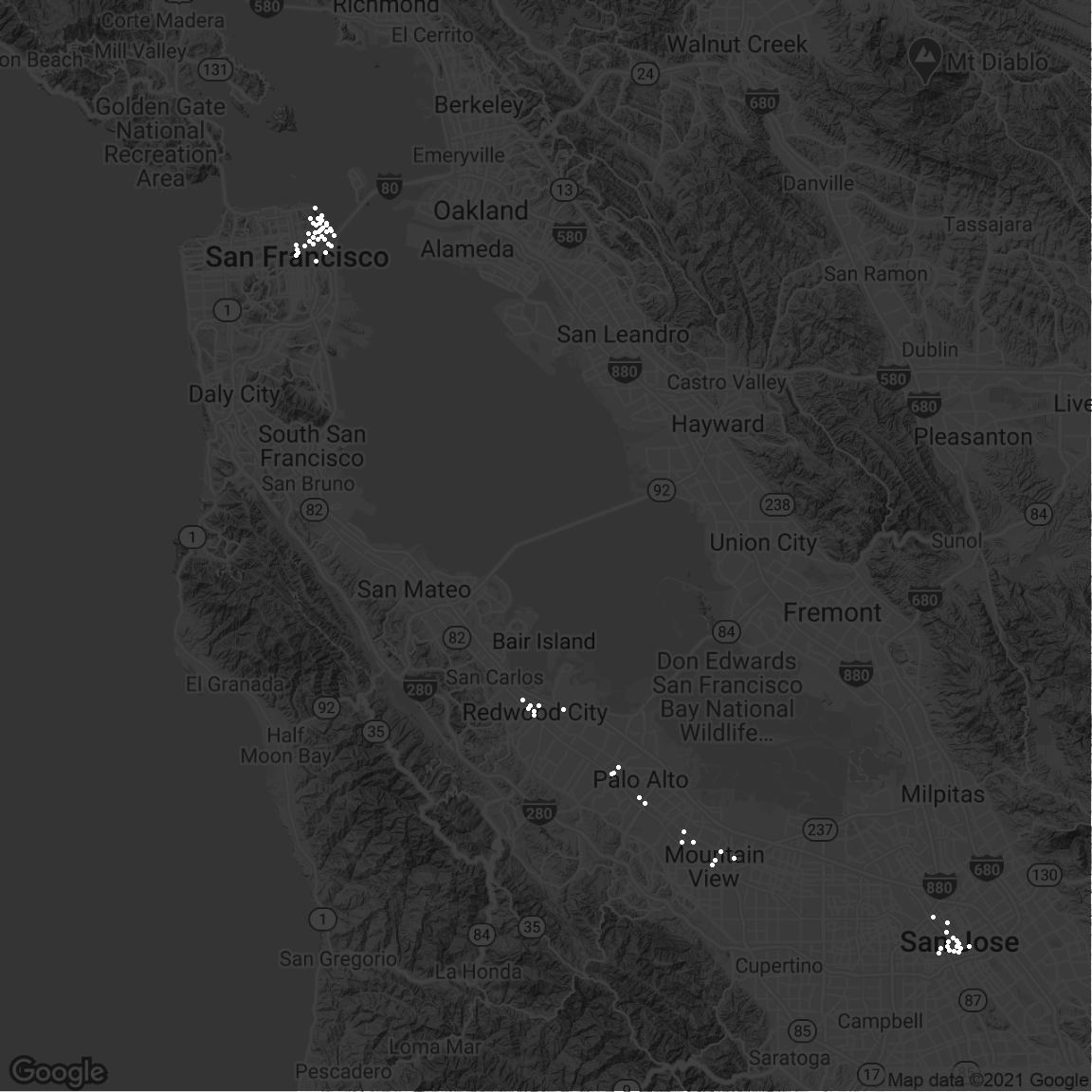
MapSanJose <- get\_map("San Jose", zoom = 14, color = "bw")

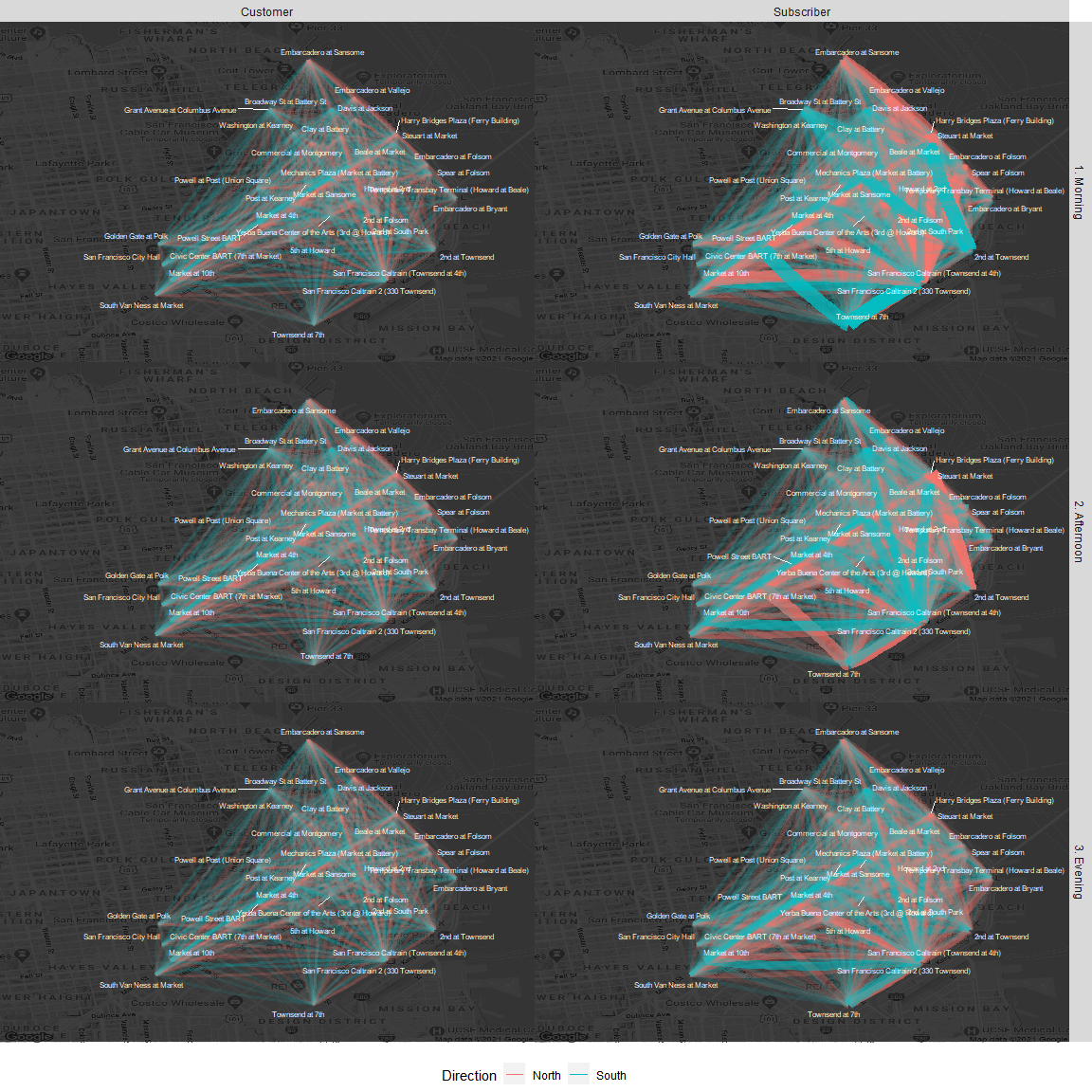
## Source : https://maps.googleapis.com/maps/api/staticmap?center=San%20Jose&zoom=14&size=640x640&scale=2&maptype=terrain&language=en-EN&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=San+Jose&key=xxx

Bike Share program words not only in San Francisco, but also in . Here is how 70 stations are spread in Bay Area (each white dot is an individual station):

ggmap(MapBayArea, extent = "device", ylab = "Latitude", xlab = "Longitude",darken = 0.75) + geom\_point(data = stations, aes(x=long, y=lat), color = "white")





#### San Francisco - Trips During Weekends

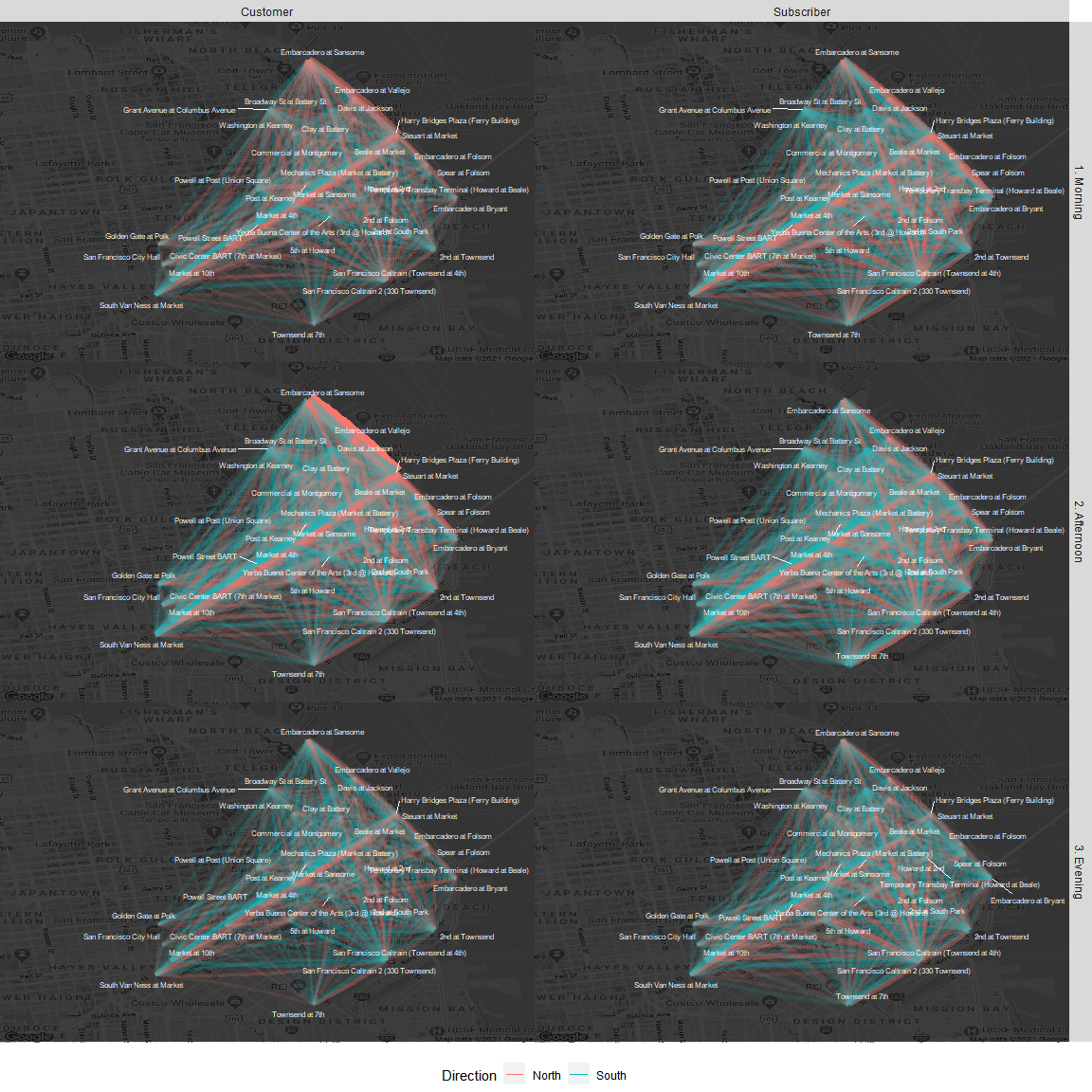
SanFranciscoFlowPlot <- getFlowPlot(MapSanFrancisco, trips\_daily[trips\_daily$DayType == "Weekend",], stations)

## Coordinate system already present. Adding new coordinate system, which will replace the existing one.

SanFranciscoFlowPlot

## Warning: Removed 977 rows containing missing values (geom\_curve).

## Warning: Removed 210 rows containing missing values (geom\_text\_repel).



#### Palo Alto - Trips During Weekdays

PaloAltoFlowPlot <- getFlowPlot(MapPaloAlto, trips\_daily[trips\_daily$DayType == "Weekday",], stations)

## Coordinate system already present. Adding new coordinate system, which will replace the existing one.

PaloAltoFlowPlot

## Warning: Removed 7436 rows containing missing values (geom\_curve).

## Warning: Removed 306 rows containing missing values (geom\_text\_repel).

## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps  
  
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps  
  
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps  
  
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps  
  
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps  
  
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



#### San Jose - Trips During Weekdays

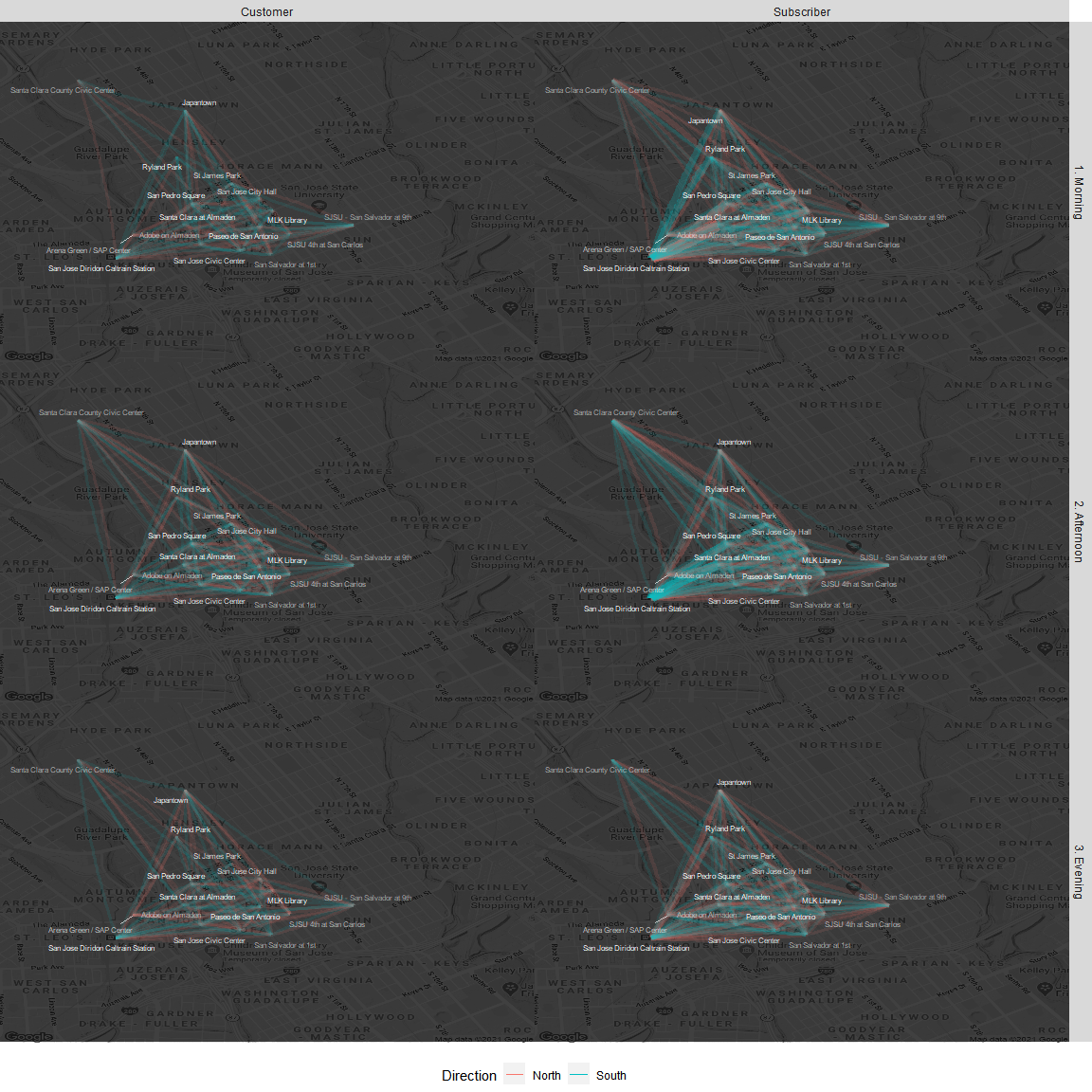
SanJoseFlowPlot <- getFlowPlot(MapSanJose, trips\_daily[trips\_daily$DayType == "Weekday",], stations)

## Coordinate system already present. Adding new coordinate system, which will replace the existing one.

SanJoseFlowPlot

## Warning: Removed 7011 rows containing missing values (geom\_curve).

## Warning: Removed 324 rows containing missing values (geom\_text\_repel).

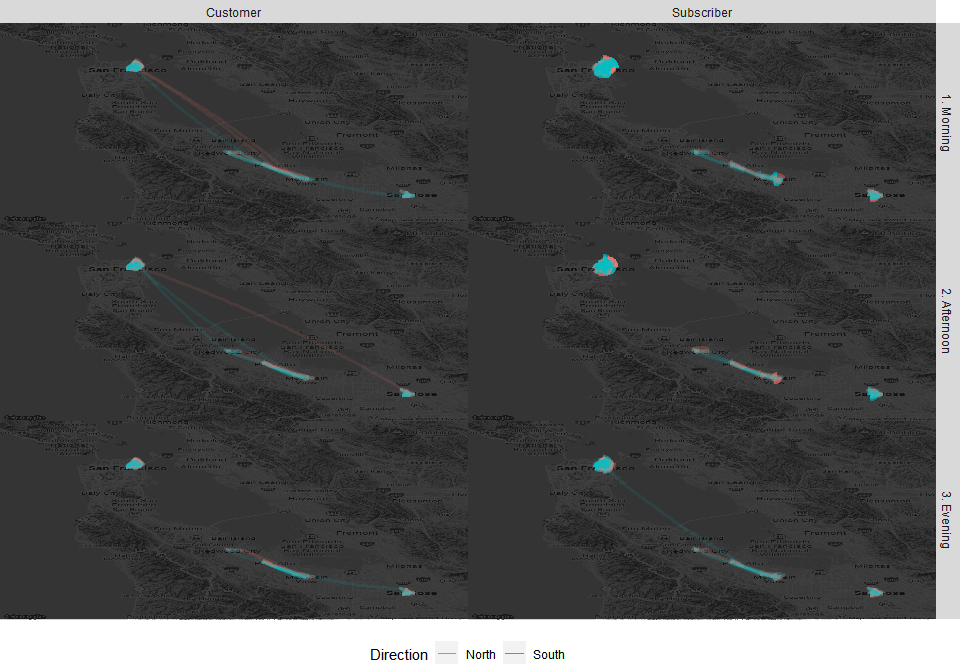
 Many

### 2.4 Intercity trips in Bay Area

InterCityTripsPlot <- getFlowPlot(MapBayArea, trips\_daily[trips\_daily$DayType == "Weekday",], stations, F)

## Coordinate system already present. Adding new coordinate system, which will replace the existing one.

InterCityTripsPlot



trips\_intercity <- trips\_daily[trips\_daily$Start.City != trips\_daily$End.City,]

kable(friends\_trip[,c('Trip.ID','Start.Date','End.Date','Start.Station','End.Station','DayType')])

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Trip.ID | Start.Date | End.Date | Start.Station | End.Station | DayType |
| 228371 | 608728 | 2015-01-18 10:28:00 | 2015-01-18 15:36:00 | University and Emerson | San Francisco Caltrain (Townsend at 4th) | Weekend |
| 228382 | 608715 | 2015-01-18 10:07:00 | 2015-01-18 15:36:00 | University and Emerson | San Francisco Caltrain (Townsend at 4th) | Weekend |
| 228383 | 608714 | 2015-01-18 10:07:00 | 2015-01-18 15:37:00 | University and Emerson | San Francisco Caltrain (Townsend at 4th) | Weekend |

## 3. Potential Issue Analysis

### 3.1 Unbalanced Stations

SanFranciscoStationsPlot <- getStationsBalancePlot(MapSanFrancisco,stations)

Blue/Purple are the stations which tend to have more bikes arriving than departing (up to 21%). Yellow are those stations that tend to have more bikes departing than arriving (up to 32%).

#### Stations in San Francisco

SanFranciscoStationsPlot

## Warning: Use of `stations$amount\_net\_percentage` is discouraged. Use  
## `amount\_net\_percentage` instead.  
  
## Warning: Use of `stations$amount\_net\_percentage` is discouraged. Use  
## `amount\_net\_percentage` instead.

## Warning: Removed 35 rows containing missing values (geom\_point).

## Warning: Removed 35 rows containing missing values (geom\_label\_repel).

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf



PaloAltoStationsPlot <- getStationsBalancePlot(MapPaloAlto,stations)

#### Stations in Palo Alto, Redwood City, Mountain View

PaloAltoStationsPlot

## Warning: Use of `stations$amount\_net\_percentage` is discouraged. Use  
## `amount\_net\_percentage` instead.  
  
## Warning: Use of `stations$amount\_net\_percentage` is discouraged. Use  
## `amount\_net\_percentage` instead.

## Warning: Removed 51 rows containing missing values (geom\_point).

## Warning: Removed 51 rows containing missing values (geom\_label\_repel).

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning: ggrepel: 5 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



SanJoseStationsPlot <- getStationsBalancePlot(MapSanJose,stations)

#### Stations in San Jose

SanJoseStationsPlot

## Warning: Use of `stations$amount\_net\_percentage` is discouraged. Use  
## `amount\_net\_percentage` instead.  
  
## Warning: Use of `stations$amount\_net\_percentage` is discouraged. Use  
## `amount\_net\_percentage` instead.

## Warning: Removed 54 rows containing missing values (geom\_point).

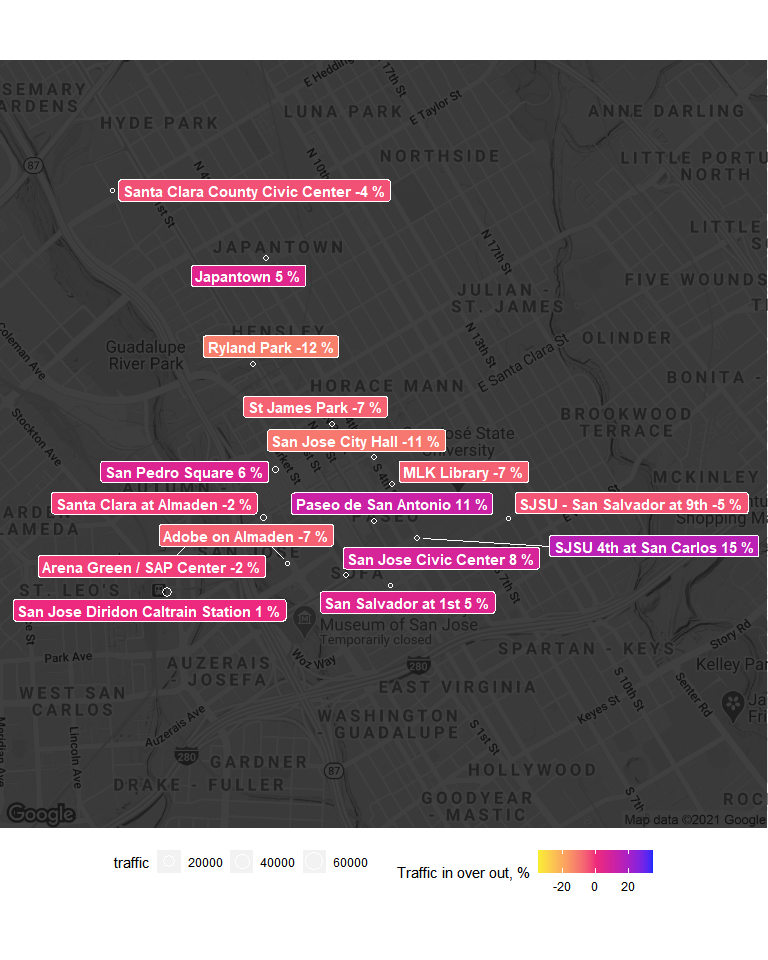
## Warning: Removed 54 rows containing missing values (geom\_label\_repel).

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

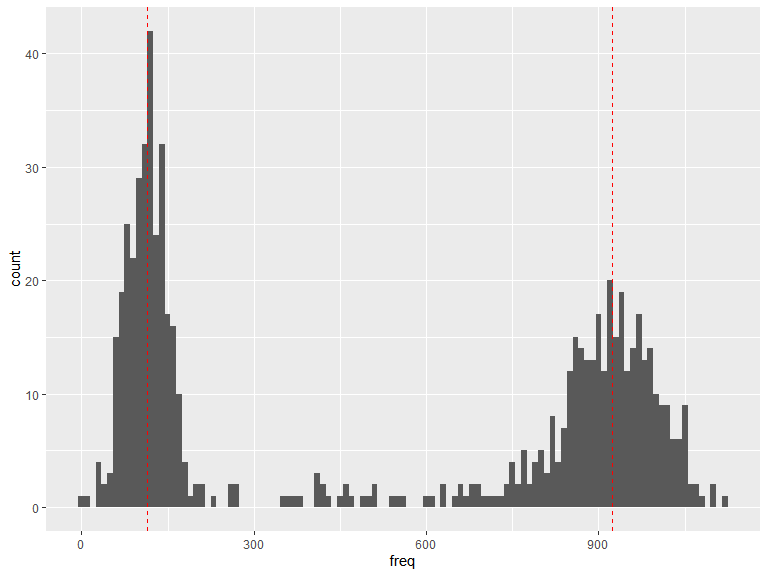
## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf



### 3.2 Bicycle Usage

bikes = count(trips, .(Bike))  
plotBikeUsageHistogram(bikes)



usageModes <- find\_modes(density(bikes$freq)$y)

## Recommendations

Here below we provide recommendations how to make the distribution of bicycle usage a more uniform or normal rather than bimodal. To do it we believe that bicycles which were extensively used in areas with high traffic should be moved to stations with low traffic, while bicycles which are almost new should be moved from stations with low traffic to stations with high traffic. Moving bicycles is also a cost so we believe that the right way to do this transfer is to do it along the regular bicycle transfer caused by inbalanced stations usage.

trips\_net <- getTripsNet(trips)

## Warning: namespace 'cachem' is not available and has been replaced  
## by .GlobalEnv when processing object '<unknown>'

bikes\_positions <- getBikesPosition(trips)   
recommendations <- getTransferBikesRecommendations(trips\_net, bikes\_positions, stations)

**Based on the trips users did in the last day (based on the current dataset) we suggest to transfer bicycles based on the following recommendations.** These recommendations are balanced (the total number of bicycles to take off is equal to the total number to bring). The number of heavily- and used few times might be not balanced, but they are more priorities than an action order.

kable(recommendations[,c('Terminal','Station','Recommendation')])

|  |  |  |
| --- | --- | --- |
| Terminal | Station | Recommendation |
| 10 | San Jose City Hall | Bring 1 heavily used bikes |
| 11 | MLK Library | Bring 2 heavily used bikes |
| 16 | SJSU - San Salvador at 9th | Take off: 181 |
| 2 | San Jose Diridon Caltrain Station | Take off: 213, 165, 663 |
| 22 | Redwood City Caltrain Station | Bring 1 heavily used bikes |
| 25 | Stanford in Redwood City | Take off: 126, 196 |
| 26 | Redwood City Medical Center | Bring 1 heavily used bikes |
| 27 | Mountain View City Hall | Take off: 35, 139 |
| 29 | San Antonio Caltrain Station | Take off: 24 |
| 30 | Evelyn Park and Ride | Bring 1 heavily used bikes |
| 31 | San Antonio Shopping Center | Bring 1 heavily used bikes |
| 34 | Palo Alto Caltrain Station | Bring 3 heavily used bikes |
| 37 | Cowper at University | Take off: 140, 230 |
| 39 | Powell Street BART | Take off: 464, 423 |
| 4 | Santa Clara at Almaden | Bring 2 heavily used bikes |
| 41 | Clay at Battery | Take off: 445 |
| 42 | Davis at Jackson | Take off: 569, 86 |
| 45 | Commercial at Montgomery | Bring 4 bikes used few times |
| 46 | Washington at Kearney | Take off: 395, 451, 290, 547 |
| 47 | Post at Kearney | Take off: 523 |
| 48 | Embarcadero at Vallejo | Take off: 491, 504 |
| 49 | Spear at Folsom | Bring 8 bikes used few times |
| 5 | Adobe on Almaden | Take off: 714 |
| 50 | Harry Bridges Plaza (Ferry Building) | Take off: 366, 609, 292, 419, 404, 583, 620 |
| 51 | Embarcadero at Folsom | Bring 7 bikes used few times |
| 55 | Temporary Transbay Terminal (Howard at Beale) | Bring 2 bikes used few times |
| 56 | Beale at Market | Bring 13 bikes used few times |
| 57 | 5th at Howard | Bring 11 bikes used few times |
| 58 | San Francisco City Hall | Bring 4 heavily used bikes |
| 59 | Golden Gate at Polk | Bring 3 heavily used bikes |
| 6 | San Pedro Square | Take off: 125, 163 |
| 60 | Embarcadero at Sansome | Take off: 409 |
| 61 | 2nd at Townsend | Take off: 463, 66 |
| 62 | 2nd at Folsom | Bring 16 bikes used few times |
| 63 | Howard at 2nd | Bring 1 bikes used few times |
| 64 | 2nd at South Park | Bring 4 bikes used few times |
| 66 | South Van Ness at Market | Bring 5 bikes used few times |
| 67 | Market at 10th | Bring 1 bikes used few times |
| 68 | Yerba Buena Center of the Arts (3rd @ Howard) | Bring 2 bikes used few times |
| 69 | San Francisco Caltrain 2 (330 Townsend) | Take off: 878, 334, 507, 16, 137, 278, 465, 602, 268, 516, 549, 214, 508, 526, 390, 222, 525, 614, 403, 594, 611, 353, 517 |
| 70 | San Francisco Caltrain (Townsend at 4th) | Take off: 532, 29, 540, 310, 328, 327, 416, 67, 342, 556, 459, 500, 484, 158, 548, 372, 575, 597, 360, 422, 579, 531, 371, 432, 413, 709, 441, 427, 109, 274, 288, 538, 336, 619, 559, 495, 629, 635, 387, 535, 187, 637 |
| 71 | Powell at Post (Union Square) | Bring 7 heavily used bikes |
| 72 | Civic Center BART (7th at Market) | Take off: 326, 370 |
| 73 | Grant Avenue at Columbus Avenue | Bring 10 bikes used few times |
| 74 | Steuart at Market | Take off: 418 |
| 75 | Mechanics Plaza (Market at Battery) | Take off: 512, 581, 462, 325 |
| 76 | Market at 4th | Bring 8 bikes used few times |
| 77 | Market at Sansome | Take off: 322, 592, 361, 622, 189, 375, 434, 458, 563, 567, 510 |
| 8 | San Salvador at 1st | Take off: 130 |
| 82 | Broadway St at Battery St | Bring 3 bikes used few times |
| 84 | Ryland Park | Bring 2 heavily used bikes |
| 9 | Japantown | Bring 1 heavily used bikes |