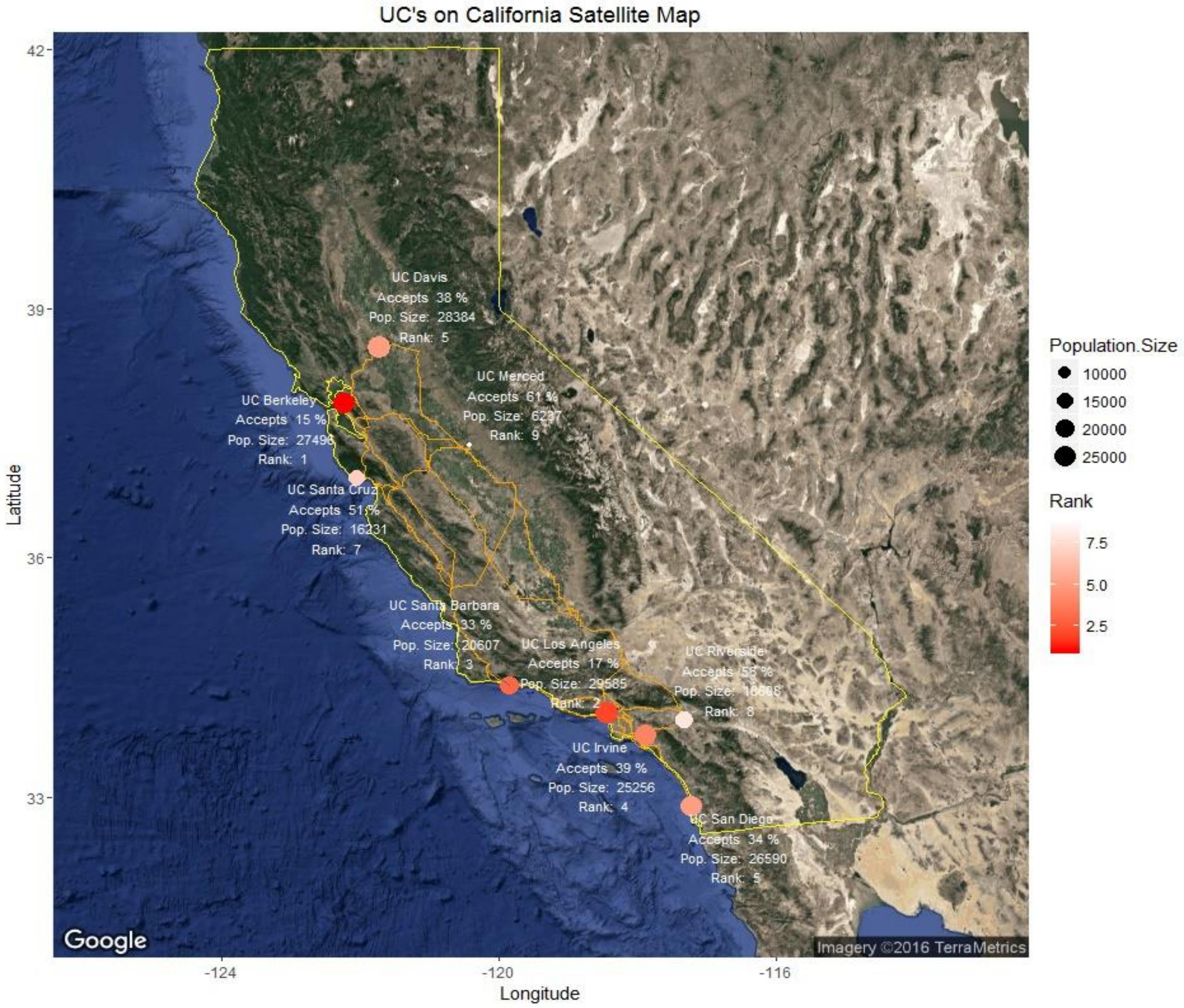
KATHLEEN ZHEN

STA 141A HOMEWORK 3

999210972

The codes and results derived by using these codes constitute my own work. I have consulted the following resources regarding this assignment:
http://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html

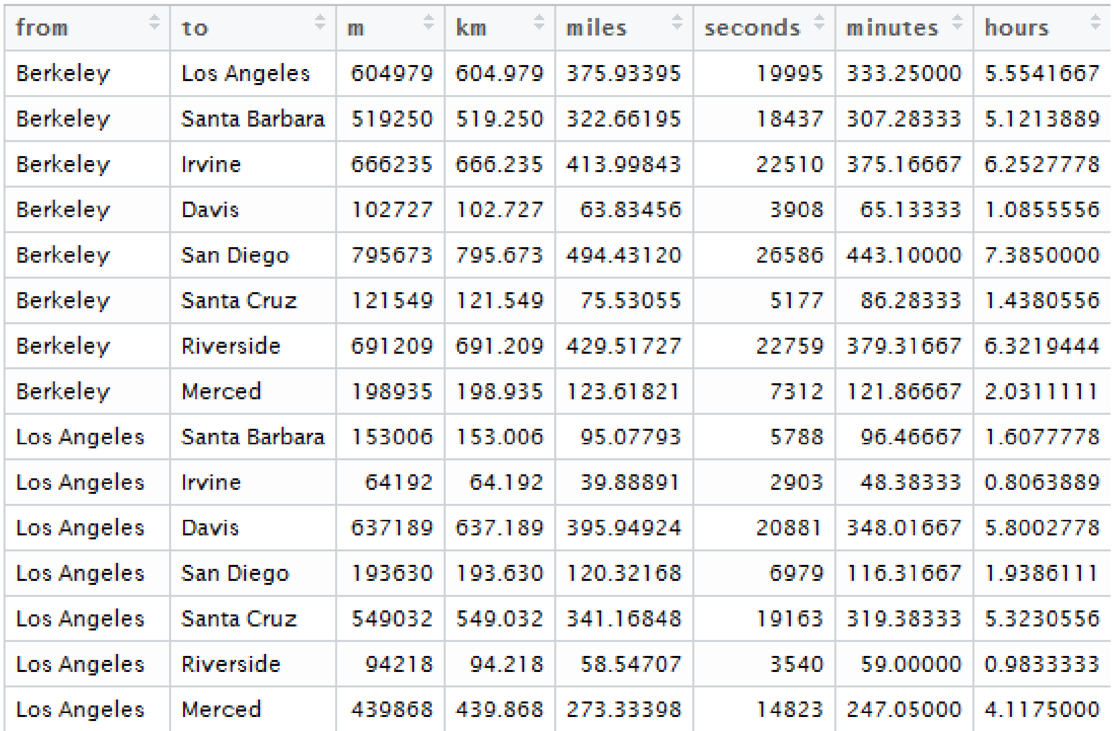
NUMBER 1



In the map depicted above, it is a satellite map of California outlined in yellow. The nine UC campuses are indicted with circle dots from colors white to red. The lighter the color (rank) is lower ranked the school is relative to each other. A darker color means the school is highly ranked relative to the others. For example, UC Berkeley is rank 1, represented with a dark red circle. The size of the circles represents the student population enrollment at the respective school. The bigger the circle, the bigger the enrollment is. UC Los Angeles has the biggest enrollment; therefore, it has the largest circle. The orange lines represent the bike routes connecting every pair of the campuses. Also on the map are the data corresponding to the school. There is the name, acceptance rate, population size, and the rank.

In order to produce the above map, I first got the satellite map in R using ggmap(). I got California's data from extracting it from the database of all 50 states in map_data() to draw the boundary with geom_polygon(). I used geocode() to get the longitude and latitudes of all the UC campuses. In order to calculate the bike routes between the campuses, I used combn() to get all the pairs of the schools. I then called a function called bike_route() that gets the bike routes between each pair, concatenated with an identifier. The identifier separates all the routes of the pairs so R knows where to do the routes. I used mapply to get the data between all pairs and then converted it to a data frame and then plotted it with geom_path(). I used geom_point() to plot the locations of the campuses with different aesthetics to show their features. I used scale_colour_gradient() to indicate the different color hues in the points. Lastly, I used geom_text_repel() to label the data points. The repel part doesn't allow the text to overlap with each other.

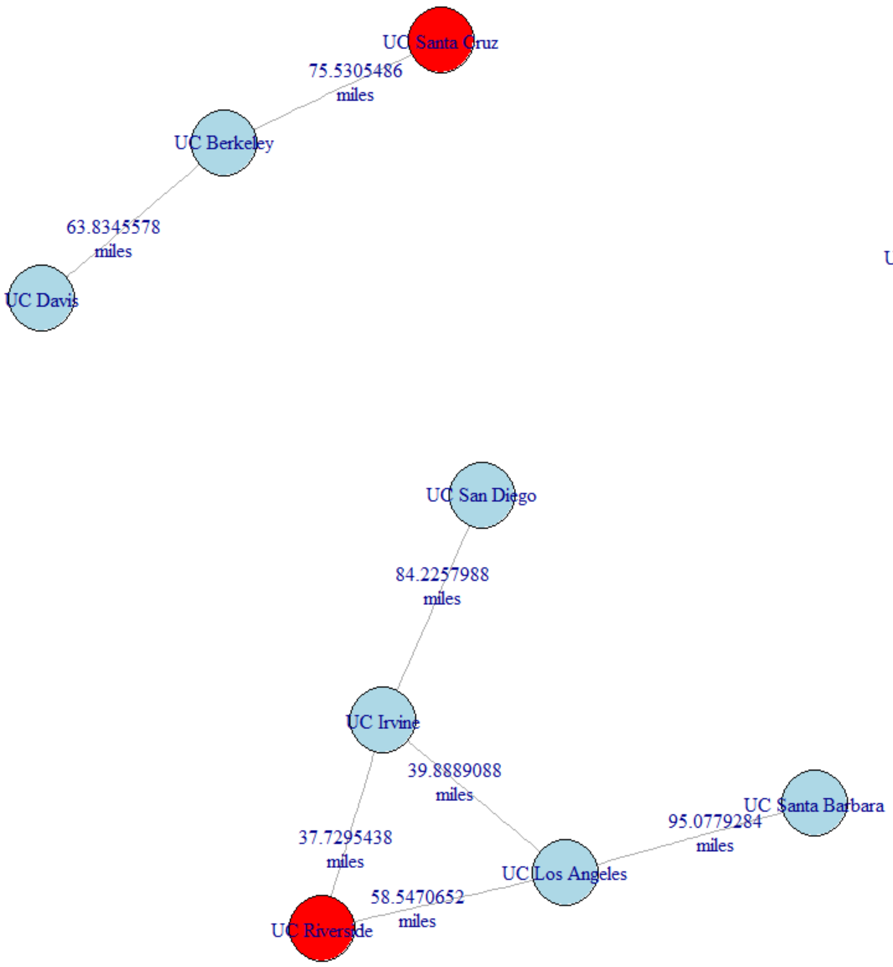
NUMBER 2



from	to	m	km	miles	seconds	minutes	hours
Berkeley	Los Angeles	604979	604.979	375.93395	19995	333.25000	5.5541667
Berkeley	Santa Barbara	519250	519.250	322.66195	18437	307.28333	5.1213889
Berkeley	Irvine	666235	666.235	413.99843	22510	375.16667	6.2527778
Berkeley	Davis	102727	102.727	63.83456	3908	65.13333	1.0855556
Berkeley	San Diego	795673	795.673	494.43120	26586	443.10000	7.3850000
Berkeley	Santa Cruz	121549	121.549	75.53055	5177	86.28333	1.4380556
Berkeley	Riverside	691209	691.209	429.51727	22759	379.31667	6.3219444
Berkeley	Merced	198935	198.935	123.61821	7312	121.86667	2.0311111
Los Angeles	Santa Barbara	153006	153.006	95.07793	5788	96.46667	1.6077778
Los Angeles	Irvine	64192	64.192	39.88891	2903	48.38333	0.8063889
Los Angeles	Davis	637189	637.189	395.94924	20881	348.01667	5.8002778
Los Angeles	San Diego	193630	193.630	120.32168	6979	116.31667	1.9386111
Los Angeles	Santa Cruz	549032	549.032	341.16848	19163	319.38333	5.3230556
Los Angeles	Riverside	94218	94.218	58.54707	3540	59.00000	0.9833333
Los Angeles	Merced	439868	439.868	273.33398	14823	247.05000	4.1175000

Above is a snippet of the distances between the campuses indicted in the column "miles". To see the full list of distances, it is located in the variable "miless".

Similar to number 1, I made a function called drive_dist() to calculate the driving distances between the pairs of the campuses. I called the function through mapply() to get a matrix of the miles between the campuses. I then transformed the matrix into a data frame. I also found the pairs of campuses that were less than 100 apart and schools with an enrollment greater than 20,000 by using subset(). With the schools that had an enrollment greater than 20,000, I used ifelse() to create a vector to label the schools apart from schools with an enrollment less than 20,000. To make the graph, I used graph.data.frame() with edges between schools less than 100 miles apart. I edited the colors with ifelse() and finally, I used plot() to generate the graph shown below.



Graph of Distances between UC Campuses

In the graph above, there is an edge connecting all UC Campuses that are at most 100 miles driving distance away from each other. UC Merced does not connect with anyone since their distances are greater than 100 miles. Each edge is labelled with the miles between each campus. The blue vertices are the campuses with a student enrollment greater than 20,000. The red vertices are the campuses with a student enrollment less than 20,000.

APPENDIX

```
ranking <- read.csv("~/R/STA 141A/HW3/ranking.txt", header=FALSE) #Read in data

#install.packages("maps")
#install.packages("ggmap")
#install.packages("ggrepel")
library("maps")
library("ggmap")
library("ggrepel")
library(RgoogleMaps)

colnames(ranking) = c('name','rank','Population.Size','percent_acceptance') #rename columns of data frame

cali = get_map("california", zoom = 6, maptype = "satellite") #Satellite map data with California in it
states = map_data("state") #getting all boundary information for 50 states

#add new column with the cities of the universities
ranking$location_name = c("Berkeley", "Los Angeles", "Santa Barbara", "Irvine", "Davis", "San Diego", "Santa Cruz", "Riverside", "Merced")

#Adding "California, USA" to each university to make it more specific
ranking$location = mapply(paste, ranking$name, "California, USA", sep = ", ")
location_lon_lat = geocode(ranking$location) #Lon/Lat of the schools
ranking$lon = location_lon_lat$lon
ranking$lat = location_lon_lat$lat
ranking$lat[5] = geocode("University of California, Davis")$lat
ranking$lon[5] = geocode("Univeristy of California, Davis")$lon
ranking$Rank= c(1:5,5,7:9) #Relative Rank
comb = combn(ranking$location, 2) #Obtaining pairs of each school

from_s = comb[1,] #from schools
to_s   = comb[2,] #to schools

#returns one pair's route with identifier
bike_route = function(from, to){
  route_df = route(from, to, mode = "bicycling", structure = "route")
  route_df$identifier = sprintf('%s to %s',from, to)
  return(route_df)
}

#calls function and get all the bike routes of the pairs
temp = mapply(bike_route, from_s, to_s, SIMPLIFY=FALSE )
temp_data_frame = do.call(rbind, temp) #convert to a data frame

#plot california
ggmap(cali) + ggtitle("UC's on California Satellite Map") + labs(x = "Longitude", y = "Latitude") +
  #draws california boundary
  geom_polygon(data = subset(states, region %in% c("california")), aes ( x = long, y = lat, group = group), color = "yellow", fill = NA) +
  #plots bike routes orange
  geom_path(data = temp_data_frame, color = "orange", aes(group = identifier)) +
  #plots universitys with aes
  geom_point(data = ranking, mapping = aes(x=lon, y=lat, size = Population.Size, colour = Rank)) +
  #colors aes differently
  scale_colour_gradient(limits=range(ranking$Rank), low = "red", high = "white") +
  #adds text
  geom_text_repel(data = ranking, aes(label = paste(as.character(name), "\n", "Accepts ", as.character(percent_acceptance), "%", "\n", "Pop. Size: ", as.character(Population.Size), "\n", "Rank: ", as.character(Rank))), size = 3, color = "white")

##PART 2

install.packages("igraph")
install.packages("sand")
install.packages("qgraph")
library("sand")
library("igraph")
library('qgraph')

temp = NULL
#function to calculate distance between pairs
drive_dist = function(from, to){
  dist_df = mapdist(from, to, mode = "driving")
  return(dist_df)
}
comb = combn(ranking$location_name, 2) #Obtaining pairs of each school

from_s = comb[1,] #from schools
to_s   = comb[2,] #to schools
#gets all the distances between pairs
temp = mapply(drive_dist, from_s, to_s, SIMPLIFY=FALSE )
miless = do.call(rbind, temp) #convert to data frame
newlocation = ranking$location_name
newmiles = subset(miless, miless$miles <= 100) #schools less than 100 miles apart
school20000 = subset(ranking, Population.Size > 20000) #schools with more than 20k Population.Size
school20000 = ifelse(ranking$Population.Size > 20000, ranking$location_name, "0")

t = graph.data.frame(newmiles, directed = FALSE, vertices = newlocation) #graph with edges for <100 miles school pairs
V(t)$color = ifelse(ranking$location_name[V(t)] == school20000, "lightblue", "red") #vertices diff colors for Population.Size >20k
V(t)$name = paste("UC", V(t)$name)
E(t)$label = paste(newmiles$miles, "miles", sep = "\n") #edges labels with miles
plot(t, vertex.color=V(t)$color, main = "Graph of Distances between UC Campuses")
```