Trial Task for IPO Project

ZhengTan

1/26/2020

Task 1

(1) Load Map

```
# Read in the map information
us <- readOGR(".", "gz_2010_us_040_00_500k")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/zhengtan/Desktop/Documents/Studying/GraduateStudy/2ndSem/IPOProject/IPOProject/Code"
## with 52 features
## It has 5 fields
```

```
# Drop Alaska and Hawaii
us.drop <- subset(us, NAME!='Alaska'&NAME!='Hawaii')
# Fortify to data frame
us.map <- fortify(us.drop)
```

```
## Regions defined for each Polygons
```

(2) Load IPO Dataset

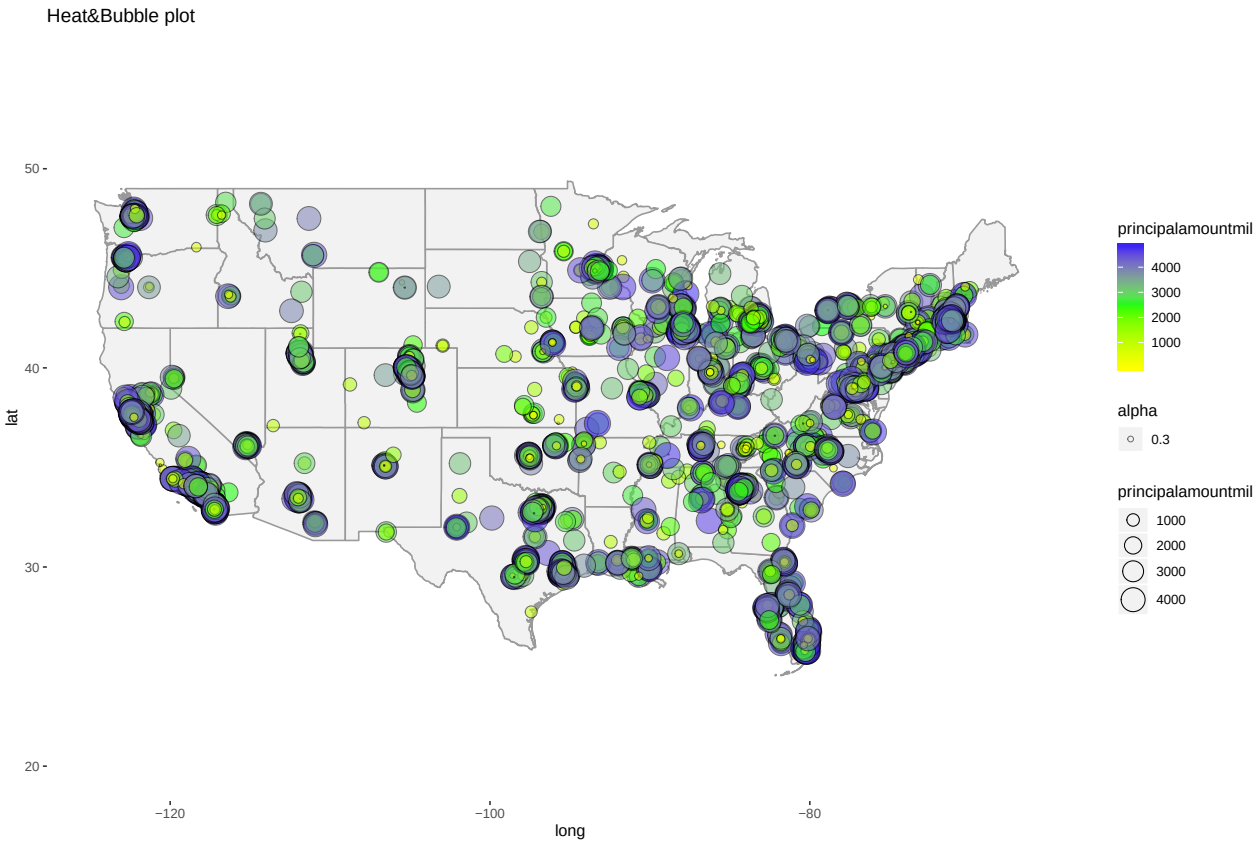
```
# Read in the csv
IPO <- read.csv("IPOcases_mergepriceindex200110.csv")
# Subset
ipodata0 <- cbind(IPO$DealNumber, IPO$long, IPO$lat, IPO$GEOID10, IPO$numberofemployees,
            IPO$sharesfiledinthismkt, IPO$principalamountmil, IPO$withdrawn_dummy)
ipodata0 <- data.frame(ipodata0)
# Column names
names(ipodata0) <- c("DealNumber", "long", "lat", "GEOID10", "numberofemployees",
                     "sharesfiledinthismkt", "principalamountmil", "withdrawn_dummy")
# Drop rows out of US mainland
ipodata <- ipodata0[-which(ipodata0$lat>49|ipodata0$lat<25|ipodata0$long<(-130)|ipodata0$long>-70),]
```

(3) Bubble Plot

```
ggplot()+
     geom_polygon(data=us.map, aes(x=long, y=lat, group=group), fill="grey95", colour="grey60")+
     geom_point(data=ipodata, aes(x = long,y = lat, size=principalamountmil, fill=principalamountmil,
```

```
                                  alpha=0.3), shape=21, colour="black")+
    ylim(20,55)+
    scale_size_area(max_size=8)+
    scale_fill_gradient2(low = "yellow", mid = "green",
                         high = "blue", midpoint=median(na.omit(ipodata$principalamountmil)))+
    ggtitle("Heat&Bubble plot")+
    theme(
         panel.grid = element_blank(),
         panel.background = element_blank(),
         )
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```



(4) Comment a. I chose the variable "principalamountmil" because firstly, I think it is a good measurement for company scale. For "numberofemployees", it is indeed a good measure, but in this dataset, there are too many missing values. And for "sharefiledinthismkt", I do not think it as a solid measurement, as companies may file shares in other markets, which can also contribute to the how large a company is. b. As can be seen from the bubble plot, most of the public companies are concentrated around East Coast and West Coast, and a lot of the rest are concentrated around the South East corner, in other words, Florida. For the remaining companies, the density on the middle east side is much higher than that on the middle west side. b. It is interesting that the distributions of companies of different sizes are even. From the plot, there is hardly any concentration of huge companies, nor gathering of small companies.

Task 2

(1) Time engineering

```
# Read in the csv
IPO <- read.csv("IPOcases_mergepriceindex200110.csv")

# Look for missing values
x <- as.numeric(IPO$filingdate)
index <- (x!=1)

# Drop missing values
IPO <- IPO[index,]

IPO$chardate <- as.character(IPO$filingdate)
IPO$chardatesplit <- strsplit(IPO$chardate, split = "/")

a <- IPO$chardatesplit

b <- unlist(a)

c <- matrix(b,ncol=3,byrow=T)

IPO$month <- as.numeric(c[ ,1])
IPO$day <- as.numeric(c[ ,2])
IPO$year <- as.numeric(c[ ,3])


for(i in 1:length(IPO$year)){
        if (IPO$year[i] < 50){
                IPO$year[i] <- IPO$year[i] + 2000
        }
        else{
                IPO$year[i] <- IPO$year[i] + 1900
        }
}
```

(2) Geospatial engineering

County indexing

```
IPO$chargeo <- as.character(IPO$GEOID10)
IPO$GEOID11 <- sprintf("%0*s", 11, IPO$chargeo)
IPO$county <- substr(IPO$GEOID11, 1, 5)

IPO$countynum <- as.numeric(IPO$county)
# Sort the dataset rows according to countynumbers
IPO.sort <- IPO[order(IPO$countynum,decreasing = F),]

# Mark the companies in the same county
```

```
g <- 0
for (i in 2:10319){
        if (IPO.sort$countynum[i]==IPO.sort$countynum[i-1]){
                IPO.sort$group[i] <- g
        }
        else{
                IPO.sort$group[i] <- g + 1
                g <- g + 1
        }
}
length(table(IPO.sort$group))
```

```
## [1] 541
```

```
#IPO.sort <- na.omit(IPO.sort[,IPO.sort$group])
```

(3) Calculating least time between IPOs within the same county

Least time between companies in the same county

```
# For companies in the same county, calculate IPO time interval
minyearinterval <- rep(0, length(table(IPO.sort$group)))

for (h in 1:(length(table(IPO.sort$group-1)))){
        group <- IPO.sort[IPO.sort$group==h,]
        group <- group[-1, ]
        l <- nrow(group)
                #sum(na.omit(IPO.sort$group==h))
        if (l==1|l==0){
                minyearinterval[h] <- NA
        }
        else{
                dif <- rep(0, (l)*(l-1)/2)
                k <- 1
                for (i in 1:(l-1)) {
                        for (j in (i+1):l){
                                dif[k] <- group$year[i] - group$year[j]
                                k <- k + 1
                        }
                }
                minyearinterval[h] <- min(abs(dif))
        }
}

total <- minyearinterval
```

Least time between companies (not withdrawn) in the same county

```
# For companies in the same county, calculate IPO time interval
minyearinterval <- rep(0, length(table(IPO.sort$group)))
```

```
for (h in 1:(length(table(IPO.sort$group-1)))){
        group <- IPO.sort[IPO.sort$group==h&IPO.sort$withdrawn_dummy==0,]
        group <- group[-1, ]
        l <- nrow(group)
                #sum(na.omit(IPO.sort$group==h))
        if (l==1|l==0){
                minyearinterval[h] <- NA
        }
        else{
                dif <- rep(0, (l)*(l-1)/2)
                k <- 1
                for (i in 1:(l-1)) {
                        for (j in (i+1):l){
                                dif[k] <- group$year[i] - group$year[j]
                                k <- k + 1
                        }
                }
                minyearinterval[h] <- min(abs(dif))
        }
}

withdrawn_0 <- minyearinterval
```

Least time between companies (withdrawn) in the same county

```
# For companies in the same county, calculate IPO time interval
minyearinterval <- rep(0, length(table(IPO.sort$group)))

for (h in 1:(length(table(IPO.sort$group-1)))){
        group <- IPO.sort[IPO.sort$group==h&IPO.sort$withdrawn_dummy==1,]
        group <- group[-1, ]
        l <- nrow(group)
                #sum(na.omit(IPO.sort$group==h))
        if (l==1|l==0){
                minyearinterval[h] <- NA
        }
        else{
                dif <- rep(0, (l)*(l-1)/2)
                k <- 1
                for (i in 1:(l-1)) {
                        for (j in (i+1):l){
                                dif[k] <- group$year[i] - group$year[j]
                                k <- k + 1
                        }
                }
                minyearinterval[h] <- min(abs(dif))
        }
}

withdrawn_1 <- minyearinterval
```

Extract the geoaptial information for each group

```
# For companies in the same county, calculate IPO time interval
county <- rep(' ', length(table(IPO.sort$group)))
long <- rep(0, length(table(IPO.sort$group)))
lat <- rep(0, length(table(IPO.sort$group)))

for (h in 1:(length(table(IPO.sort$group-1)))){
        county[h] <- IPO.sort[IPO.sort$group==h,]$county[2]
        long[h] <- IPO.sort[IPO.sort$group==h,]$long[2]
        lat[h] <- IPO.sort[IPO.sort$group==h,]$lat[2]
}
```

(4) Reform a new dataset

```
group_id <- c(0:540)
same <- data.frame(cbind(group_id, total, withdrawn_0, withdrawn_1, county, long, lat))
```

The new dataset contains informations of time intervals (years) between companies IPO within each county.

a. Variable 'total' contains the minimum time intervals between all IPO companies within each county.
b. Variable 'withdrawn_0' contains the minimum time intervals between not withdrawn IPO companies within each county.
c. Variable 'withdrawn_1' contains the minimum time intervals between withdrawn IPO companies within each county.
d. For all the NA values, it means that this county only has one IPO company. So we do not have to concern about these areas.

(5) Comment and Further Illustration:

For this question, I do not think dealing with county is the best solution. Counties are small and this data manipulation takes counties as discrete, isolated areas.

There might be near IPOs within the same area, though they are not in the same county, or there could be IPO in one county but affecting the real-estate prices in its around conuties.

So, instead, maybe it is better to define "spatial near" as a continuous term, for example, Euclidean Distance with longitude and latitude information.