Stats 405 Final Project

6/14/2017

Introduction

My objective is to create a MySQL database with data pulled from www.insideairbnb.com. I will populate the database with the listings, neighbourhoods, and reviews datasets for the cities on the US West Coast. This database will be on my local system, but with the same methodology, one could create this database on the cloud.

After uploading the files to my database, I will clean the tables to facilitate joins. I will perform several joins and confirm that the generated dataset is of correct dimension.

I am interested in investigating the rental listings of the last two neighbourhoods I have lived in: The Mission District in San Francsico and Culver City in Los Angeles. Specifically, I am interested in the geographic distribution of the prices of the rentals. I will query the database and generate visualizations that shed some light on this.

Connecting to MySQL

I already have MySQL downloaded on my system, so I won't include how to do that. It is also very easy to spin up an instance on AWS with MySQL preloaded on the virtual machine image. Accordingly, if you start an appropriate server, you can follow these instructions to create this database on the cloud.

First, I go to my System Preferences and make sure that MySQL server is running on my system. After confirming this, I open the terminal and we use the following to open the MySQL shell:

/usr/local/mysql/bin/mysql -u root -p

This prompts me for the temporary password that was given to me in the MySQL download flow. After typing the password, I am in the MySQL shell. Now I reset the password and create the blank database from this shell.

create database stat_405_final;

Now that this database is created, I want to create a connection object in R through the RMySQL package to this database.

```
mydb <- dbConnect(MySQL(), user='root', password='garbanz0', dbname='stats_405_final', host='localhost
```

Creating the database

Now, through this connection object, I can write my data to the database. I pulled the neighbourhood, review and listing files in csv format from www.insideairbnb.com. For efficiency, I read the data files with fread(), convert them to data.table objects, and write them to the databases through the connection. I chose the cities on the US West Coast for this database, so I have data from San Deigo (SD), Los Angeles (LA), Santa Cruz (SC), San Francisco (SF), Oakland / East Bay (EB), Portland (PL) and Seattle (ST). Note: I have commented out the write commands in this .Rmd file becuase these commands do not need to be exected again.

```
# Los Angeles
listings_LA <- as.data.table(fread("~/Desktop/airbnb/listings_LA.csv"))
#dbWriteTable(mydb, value = listings_LA, name = "listings_LA", append = TRUE )

reviews_LA <- as.data.table(fread("~/Desktop/airbnb/reviews_LA.csv"))
#dbWriteTable(mydb, value = reviews_LA, name = "reviews_LA", append = TRUE )

neighbourhoods_LA <- as.data.table(fread("~/Desktop/airbnb/neighbourhoods_LA.csv"))
#dbWriteTable(mydb, value = neighbourhoods_LA, name = "neighbourhoods_LA", append = TRUE )

# San Diego
listings_SD <- as.data.table(fread("~/Desktop/airbnb/listings_SD.csv"))
#dbWriteTable(mydb, value = listings_SD, name = "listings_SD", append = TRUE )

reviews_SD <- as.data.table(fread("~/Desktop/airbnb/reviews_SD.csv"))
#dbWriteTable(mydb, value = reviews_SD, name = "reviews_SD", append = TRUE )

neighbourhoods_SD <- as.data.table(fread("~/Desktop/airbnb/neighbourhoods_SD.csv"))
#dbWriteTable(mydb, value = neighbourhoods_SD, name = "neighbourhoods_SD", append = TRUE )

# San Francisco
listings_SF <- as.data.table(fread("~/Desktop/airbnb/listings_SF.csv"))
#dbWriteTable(mydb, value = listings_SF, name = "listings_SF", append = TRUE )

reviews_SF <- as.data.table(fread("~/Desktop/airbnb/reviews_SF.csv"))
#dbWriteTable(mydb, value = reviews_SF, name = "reviews_SF", append = TRUE )

neighbourhoods_SF <- as.data.table(fread("~/Desktop/airbnb/neighbourhoods_SF.csv"))
#dbWriteTable(mydb, value = neighbourhoods_SF, name = "neighbourhoods_SF", append = TRUE )

# Santa Cruz
listings_SC <- as.data.table(fread("~/Desktop/airbnb/listings_SC.csv"))
#dbWriteTable(mydb, value = listings_SC, name = "listings_SC", append = TRUE )

reviews_SC <- as.data.table(fread("~/Desktop/airbnb/reviews_SC.csv"))
#dbWriteTable(mydb, value = reviews_SC, name = "reviews_SC", append = TRUE )

neighbourhoods_SC <- as.data.table(fread("~/Desktop/airbnb/neighbourhoods_SC.csv"))
#dbWriteTable(mydb, value = neighbourhoods_SC, name = "neighbourhoods_SC", append = TRUE )

# Oakland
listings_EB <- as.data.table(fread("~/Desktop/airbnb/listings_EB.csv"))
#dbWriteTable(mydb, value = listings_EB, name = "listings_EB", append = TRUE )

reviews_EB <- as.data.table(fread("~/Desktop/airbnb/reviews_EB.csv"))
#dbWriteTable(mydb, value = reviews_EB, name = "reviews_EB", append = TRUE )

neighbourhoods_EB <- as.data.table(fread("~/Desktop/airbnb/neighbourhoods_EB.csv"))
#dbWriteTable(mydb, value = neighbourhoods_EB, name = "neighbourhoods_EB", append = TRUE )

# Portland
listings_PL <- as.data.table(fread("~/Desktop/airbnb/listings_PL.csv"))
#dbWriteTable(mydb, value = listings_PL, name = "listings_PL", append = TRUE )
```

```
reviews_PL <- as.data.table(fread("~/Desktop/airbnb/reviews_PL.csv"))
#dbWriteTable(mydb, value = reviews_PL, name = "reviews_PL", append = TRUE )

neighbourhoods_PL <- as.data.table(fread("~/Desktop/airbnb/neighbourhoods_PL.csv"))
#dbWriteTable(mydb, value = neighbourhoods_PL, name = "neighbourhoods_PL", append = TRUE )

# Seattle
listings_ST <- as.data.table(fread("~/Desktop/airbnb/listings_ST.csv"))
#dbWriteTable(mydb, value = listings_ST, name = "listings_ST", append = TRUE )

reviews_ST <- as.data.table(fread("~/Desktop/airbnb/reviews_ST.csv"))
#dbWriteTable(mydb, value = reviews_ST, name = "reviews_ST", append = TRUE )

neighbourhoods_ST <- as.data.table(fread("~/Desktop/airbnb/neighbourhoods_ST.csv"))
#dbWriteTable(mydb, value = neighbourhoods_ST, name = "neighbourhoods_ST", append = TRUE )
```

Now, I run some queries in MySQL to make sure that the datasets were uploaded properly. I like to check the columns for each table using the following query:

SHOW COLUMNS FROM listings_SC;

When I initially uploaded the data, these queries showed me that there was no shared column between these two tables. After examining the data, I relaized that the column "id" in listings, is intended to match "listing_id" from the review table. This problem existed across all 7 cities I chose. In the MySQL shell, I changed the column name so that I could join the listing and review table on the shared variable "listing_id".

ALTER TABLE listings_SC CHANGE id listing_id bigint(20);

After I do this for each listing tbale, the database is in a friendly format, ready for joins and queries.

Joining datasets

Below, we will perform a few joins on datasets through a common variable. It is worth noting that the datasets were deleted from my local system and they exist solely in the SQL database. These joins and queries are flowing through the RMySQL connection object.

```
inner_SC <- dbGetQuery(mydb, "SELECT * FROM listings_SC INNER JOIN reviews_SC ON listings_SC.listing_i

left_SC <- dbGetQuery(mydb, "SELECT * FROM listings_SC LEFT JOIN reviews_SC ON listings_SC.listing_id

right_SC <- dbGetQuery(mydb, "SELECT * FROM listings_SC RIGHT JOIN reviews_SC ON listings_SC.listing_i

dim(inner_SC)
```

```
## [1] 22121    19
```

```
dim(left_SC)
```

```
## [1] 22233    19
```

```
dim(right_SC)
```

```
## [1] 22121    19
```

Looking at the dimensions, we see that the right-joined and the left-joined datsets have equal dimension, which is what we expect.

Pulling data

Now we wan to pull some interesting data so we can analyze it in RStudio. I am interested in comparing the last 2 neighbourhoods I have lived in: San Francisco's Mission District and Culver City in Los Angeles. My experience indicates that The Mission is very expensive, and one of the most desirable places to stay while visiting San Francisco. Culver City is less expensive than The Mission, and I suspect that there is less demand for AirBnb rentals. I want to shed some light on these impressions, so I will pull the data from both neighbourhoods and will pull data from the database, run some data summaries, and generate some visulaizations.

First, I will just pull the data from my SQL database and import it into R as data.frames.

```
SF_all <- dbGetQuery(mydb, "SELECT * FROM listings_SF")
LA_all <- dbGetQuery(mydb, "SELECT * FROM listings_LA")
SF_100 <- dbGetQuery(mydb, "SELECT * FROM listings_SF WHERE price > 100")
LA_100 <- dbGetQuery(mydb, "SELECT * FROM listings_LA WHERE price > 100")
# these work
SF_mission <- dbGetQuery(mydb, "SELECT latitude, longitude, price FROM listings_SF WHERE neighbourhooc
LA_culver <- dbGetQuery(mydb, "SELECT * FROM listings_LA WHERE neighbourhood = 'Culver City'")
```

Now, I will run some simple queries to get familiar with my data:

```
dim(SF_mission)
```

```
## [1] 1036    3
```

```
dim(LA_culver)
```

```
## [1] 252  17
```

```
summary(SF_mission$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      20      99     150     212     240    4000
```

```
summary(LA_culver$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    27.0    70.0   100.0   151.3   151.0  5000.0
```

This tells me that there are four times as many as rentals listed in the Mission than in Culver City. This confirms my initial suspicions that the Mission has a higher demand for listings. There is also some interesting information in the price summaries. The Mission's distribution is slightly more expensive, with the exception of the most expensive listing. There is a $5000 / night unit in Culver City, which I did not imagine. Now I am curious where exactly the super expensive units in Culver City are.

Generating Visualizations

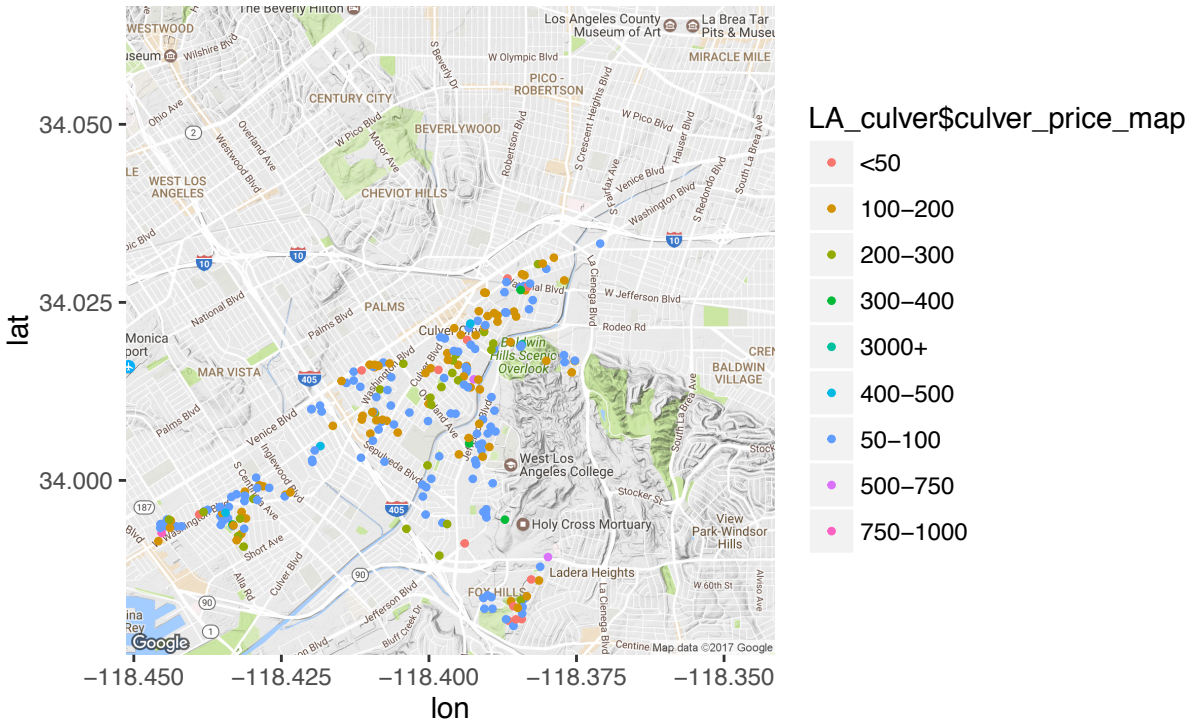
Now, I want to visualize this data on a map. I will use the ggmap and ggplot2 libraries to do this. In these visualizations, I am interested in seeing the geographic distribution of the prices of the rentals. I know both these neighborhoods intimately, so I want to see if there are discernible patterns here. Since the price distribution is heavily skewed for both these nieghborhoods, plotting price directly as a color aesthetic didn't have enough variation. So I created a categorical variable of price ranges.

```
# make sure that ggmap and ggplot2 are enabled on the session


culver_price_map <- ifelse(LA_culver$price < 50, "<50",
       ifelse((LA_culver$price >= 50) & (LA_culver$price <= 100), "50-100",
       ifelse((LA_culver$price >= 100) & (LA_culver$price <= 200), "100-200",
       ifelse((LA_culver$price > 200) & (LA_culver$price <= 300), "200-300",
       ifelse((LA_culver$price > 300) & (LA_culver$price <= 400), "300-400",
       ifelse((LA_culver$price > 400) & (LA_culver$price <= 500), "400-500",
       ifelse((LA_culver$price > 500) & (LA_culver$price <= 750), "500-750",
       ifelse((LA_culver$price > 750) & (LA_culver$price <= 1000), "750-1000",
       ifelse((LA_culver$price > 1000) & (LA_culver$price <= 3000), "1000-3000",
              "3000+"))))))))))

LA_culver <- mutate(LA_culver, culver_price_map)

mission_price_map <- ifelse(SF_mission$price < 50, "<50",
                     ifelse((SF_mission$price >= 50) & (SF_mission$price <= 100), "50-100",
                     ifelse((SF_mission$price >= 100) & (SF_mission$price <= 200), "100-200",
                     ifelse((SF_mission$price > 200) & (SF_mission$price <= 300), "200-300",
                     ifelse((SF_mission$price > 300) & (SF_mission$price <= 400), "300-400",
                     ifelse((SF_mission$price > 400) & (SF_mission$price <= 500), "400-500",
                     ifelse((SF_mission$price > 500) & (SF_mission$price <= 750), "500-750",
                     ifelse((SF_mission$price > 750) & (SF_mission$price <= 1000), "750-1000",
                     ifelse((SF_mission$price > 1000) & (SF_mission$price <= 3000), "1000-3000",
                     "3000+"))))))))))

SF_mission <- mutate(SF_mission, mission_price_map)

myMap <- get_map("culver city", 13)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=culver+city&zoom=13&size=640x64
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=culver%20city&sensc
```

```
ggmap(myMap)+
  geom_point(aes(x =LA_culver$longitude, y = LA_culver$latitude,     color=LA_culver$culver_price_map)
             data = LA_culver, size =.85)
```



The colors might not come out well in this document, but in RStudio, I am able to zoom in on the ggmap and inspect a little closer. I notice several things in this plot. First, there are 3 places that are anomously expensive: 1 unit on the border of Ladera Heights overlooking Holy Cross Mortuary, 1 unti overlooking Baldwin Hills, and 1 unit on W. Washington essentially in the Marina. I know all these areas to have great views and expensive homes. I also notice that Culver City Center does not have the most expensive units, but there aren't any cheap ones there either. This plot also shows that Culver City is bordered by Venice Blvd., so this limits the patterns we can see.

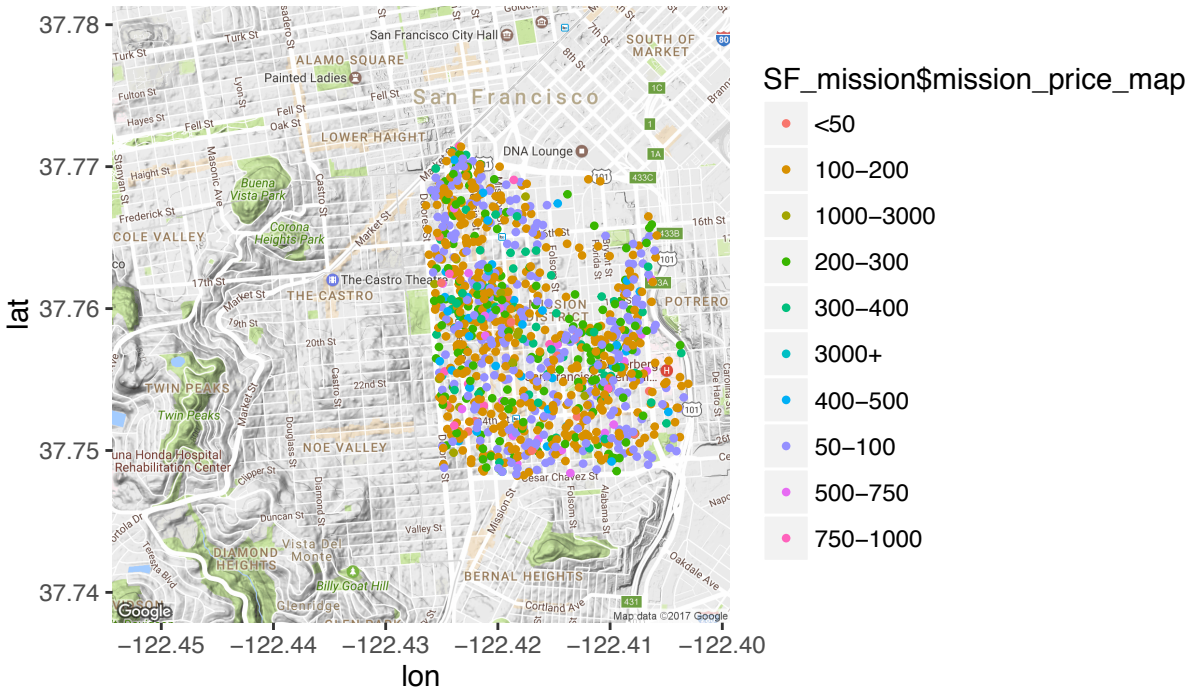
Let's look at the Mission.

```
myMap <- get_map("dolores park", 14)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=dolores+park&zoom=14&size=640x6
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=dolores%20park&sens
```

```
ggmap(myMap)+
  geom_point(aes(x =SF_mission$longitude, y = SF_mission$latitude, color=SF_mission$mission_price_map)
             data = SF_mission, size =.85)
```



First things that jumps out is the density of rental units here. If you don't reduce the size parameter of the geom_point object nothings is discernible. This jives with the narrative that as the Mission gentrifies, family homes are converted to for-profit rentals. Nearly all the units in this neighborhood are expensive by my budjet, but Guerreo St, south of 24th St and near the hospital seems to have the most expensive units. These are more residential areas. I also notice a few pockets relatively devoid of rental units. Specifically, a stretch around Folsom and the intersection of 16th and Mission. Both these make perfect sense. 16th and Mission has a BART station that is very popular with drug dealers and that section of Folsom street has dozens of homeless living in tents on the sidewalks.

Below, we have a map of reported crimes (Arson, Assault, Drugs / Alcohol Violations, Homicide, Motor Vehicle Theft, Robbery, Sex Crimes, Theft / Larceny, Vandalism, Vehicle Break-In / Theft, Weapons Violations) in the Mission Distric from the week of June 9th - June 16th, 2017. The screenshot below is taken from https://www.crimemapping.com/map/ca/sanfrancisco. This neighborhood epitomizes gentrification, masses of tech workers and homeless.

In this image, we can see that Mission Street (vertical, centered) has a string of incidents. I can confirm that this is standard in the Mission. Notice that the gunshot in the top center of this image is almost centered on the pocket of limited rentals in the airbnb data. This is the aforementioned 16th and Mission BART stop. We can also see that Guerrero St has expensive rental units and almost no violent crime (at least in the past week).

Conclusions

Ok, so what is the point of pulling and mapping AirBnb data, given that they have a nice interactive map on their website? Well, remember that we are looking at historic data that is no longer on their website. So there is some novelty in analyzing it. Imagine integrating crime data with the AirBnb listings. It seems pretty feasible to write an algorithm that reports expensive listings in close proximity to violent crime. This could be a flag to identify renters who are questionably ethical.

Just looking at the historical mapping of AirBnb rental prices also sheds some light on the neighborhood. For example, San Francsico is famous for intricately painted Victorian houses. If you think that the most

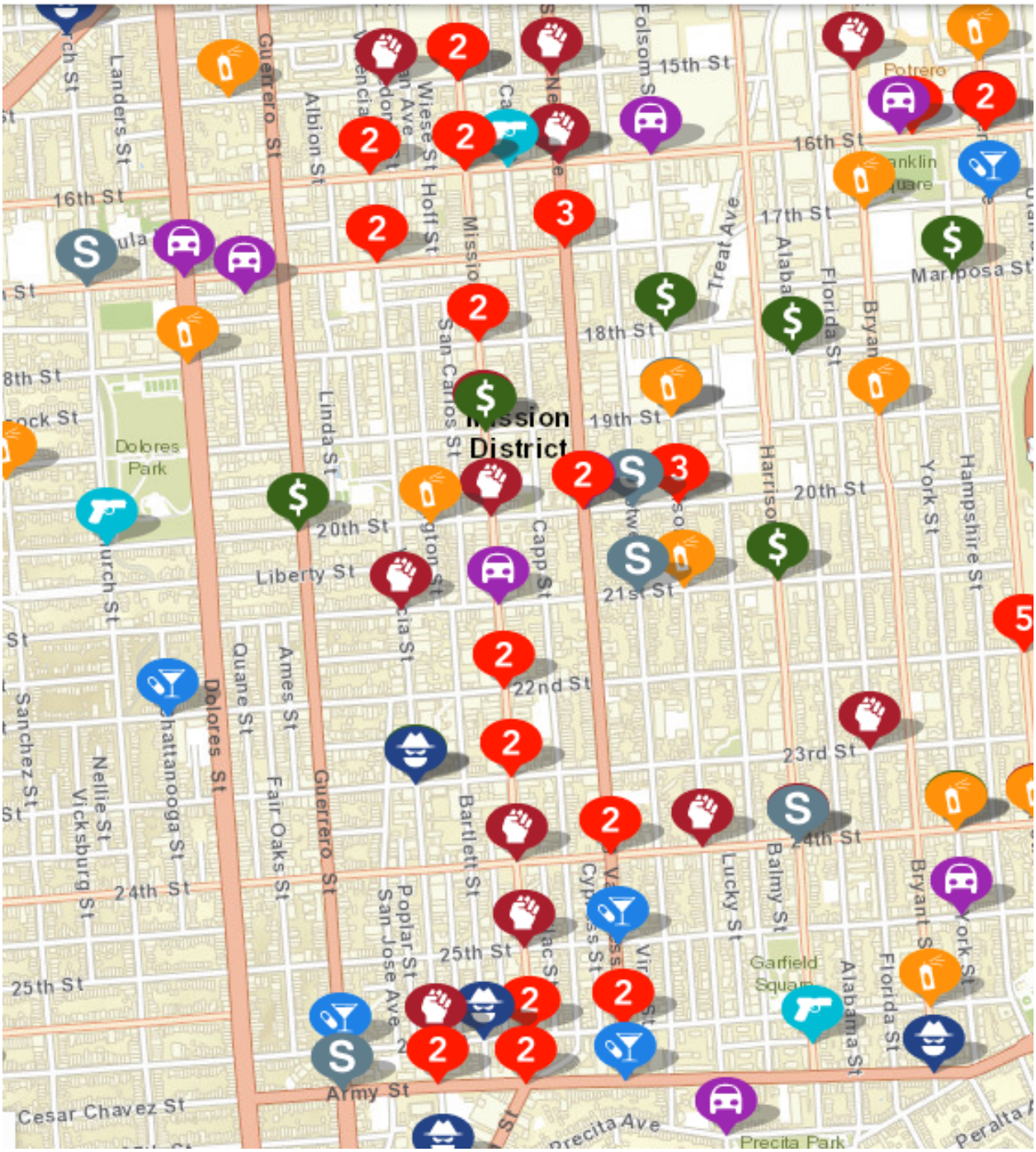


Figure 1: Mission District Crime Map

charming neighborhoods with the best viewws will have the highest rental prices, then this can be used to map an architectural walking tour.

My objective was to create a database, upload data, perform some joins, query data, and play around with some visualizations. I think I accomplished this and it was informative, practial experience working with SQL and geo-spatial data. My conclusions from the visualizations aren't particularily edifying, but this dataset is a nice sandbox and I will keep playing with it. Any ideas for further exploration are appreciated!