Analysis of House Rentals Posts from Craigslist

Wenjia Wang

Statistics Department

University of California, Davis

1 Introduction

Craigslist is a website that allows people to post classified advertisements for free. These posts span a variety of subjects, including job postings, housing rentals, and item sales. Craigslist started in the San Francisco Bay Area and now serves 570 cities worldwide. Posts are grouped by city to make relevant ads easy to find.

To analyze the recent Craigslist posts for apartment rentals in California, I first scraped relevant posts from Craigslist. However, since Craigslist has few rules about how posts should be formatted, this data set is messy. So, this report mainly contains 3 sections: (1) extract features from the original post texts by using regular expression and explore the multicollinearity among these features; (2) preliminarily analyze the factors of rental prices; (3) Explore the relationship of these house features to geographical characteristics of rental markets and make further inference by considering demographic map.

2 Preprocessing Data

The “messy” folder contains Craigslist posts in Los Angeles, Sacramento, San Diego and San Francisco Bay Area, and each post is a separate text file. Before importing all files into R, I checked the format of some files and found that in each file, the first line is the title, followed by “QR Code Link to This Post”, and then the description in the middle part is of great diversity. However, the last 7 lines in each file record the date posted, the rental price, latitude, longitude, the number of bedrooms, the number of bathrooms, and the area of the apartment.

2.1 Read in Post Files and Remove Duplications

Initially, I tried to separate each line as a single element, but the middle part of each file are different from each other. So if I applied the function to different post files, the return vectors would have different lengths. Consider the above format in each file again, I designed a function named “read_posts” to import one file into R given the direction of the file. In order to make my future analysis a little bit easier, this function will not only read in a post file but also make a preliminary separation. It combines lines in the middle part of each file as one element, but take the title and the last 7 lines as 8 individual elements. Finally, this function will return a vector containing 9 elements: "Title", "Date posted", "Price", "Latitude", "Longitude", "Bedrooms", "Bathrooms", "Sqft".

However, the “read_posts” function can only read in one file each time. To import data efficiently, I will design a function to read in a batch of files one time. The read_all_posts function applies read_posts function to each file under a specific direction, and retruns a data frame. The rows and columns of the data frame are similar to the "cl_apartments" file in assignment 3 and 4. Each row in the data frame represents a post, and in order to mark the region of each post, I add a new column named "region". Therefore, this function finally will return a data frame with 10 columns representing the 10 features of each post file.

Next, I will apply the “read_all_posts” to read in all post files under “messy” folder. Since there are 9 regions, to make is more efficient, I used “lapply” to apply “read_all_posts” function to each folder under “messy”, and obtained a list containing 9 data frames. Then, I bound the 9 data frames together.

The last step before moving on to the following questions is to remove the duplications in this data. It makes sense that posts whose titles are the same tend to be duplicated. It turns out that San Francisco Bay Area have the most duplications. Then, I removed those 8227 rows with the same titles. After that, we got a data set with 37618 observations, which is the final data set I used to explore further.

2.2 Rental Price Extracted from Title

After reading some of the posts, I find that the price in the title starts with "\$", followed by several numbers. So I use pattern like"$?[0-9,]+" to obtain the value of price. The price extracted from title have 37585 non-missing values as well as 33 missing values. Then I looked closely at those titles excluding rental price, and find that some of them indeed didn’t mention rental price, while many are actually not about apartment rental, but advertisements, such as Pest Control Services.

As far as I am concerned, the user-specified prices refer to the 6th last line in each original post file or in other words, the “Price” attribute in my data set. I reprocessed the “Price” attribute here to only keep the number of price and then compared with the price extracted from the title. It turned out that except 15 files, the price extracted from title is the same as the user-specified prices in all other files. Actually, the values in price attribute of these 15 exceptions are 0. I think this is because that most of them are advertisements, such as door repairing service, or services to painting. However, only one observation among them is about apartment rental and have an exact price $3200. I will replace the original price value in the data set with 3200 and get rid of the remaining meaningless observations (also including those rows missing price value). Now, the data set contains 37474 rows, and I will use it to do further analysis.

2.3 Rental Deposit

Having read some description carefully, I found that the amount of deposit has similar pattern, such as "security deposit of $4,100", "Deposit: $700",etc. So I used a pattern basely like "Deposit: $" (you can refer to the appendix to get the specific pattern) as an anchor to extract the number of deposit from the description. Even though I know this pattern also appropriate for pet deposit, in most cases it actually can only get the apartment deposit because “str_match” function only outputs the first match and the apartment deposit always is mentioned before pet deposit in most posts. When I firstly tried this pattern, I only got 5965 deposits, which is quite small based on the overall 37618 observations. So I checked whether

deposit missing values happened only when the corresponding description didn't mention apartment deposit. Unfortunately, there are some descriptions actually contain the rental deposit like "$800 deposit". That means that the pattern I used before is not adequate. So I used the new pattern to extract rental deposit again, and chose the larger value of my first and second results under the assumption that rental deposit will be no less than pet deposits. I made up my first pattern and finally obtained 8475 apartment deposit values.

Figure 1 below compares the security deposit with the price of each department and the points on the red straight line represent the apartment with the same deposits and rental price. We can see that there are main two patterns in the plot: (a) The deposit amount is very closed to the price for some departments; (b) For others, no matter how much the rental price is, the security deposit is fixed. I think this makes sense, since the security deposit is equal to the first month rental price on many occasions, but sometimes, the deposit is just a fixed number for security. Besides, I guess that different regions may have different fixed security deposits.

Figure 1

2.4 Pet Policy

In this part, I will add a categorical variable representing the pet policy of each apartment based on its description. In order to focus on the sentences mentioning pet policy, I will first filter the posts which mention "pet", "cat", "dog" or other possible pet, then only extract the sentences including "pet", "cat", "dog" or other possible pet among these posts, and it turned out that 14721 posts including this kind of sentences. In this way, it is much easier to figure out the pattern in each post refered to its pet policy.

I found many sentences contain "pet-friendly" or "pets generally accepted", or "pets deposit", but this doesn't mean that both dogs and cats are allowed. Actually, under this condition, if one sentences also contain "cats allowed(okay, ect.)", but not "small dogs", this post is more

likely to allow cats but not dogs, and vice versa. And if the description contains neither "cats allowed(okay, ect.)" nor "small dogs", but only "pet-friendly", it is regarded as allowing both.

On the other hand, for those posts excluding "pet-friendly" or "pets generally accepted", if it contains both "cats allowed(okay, ect.)" and "small dogs", this apartment may allow both, while if it just contains one of the pattern, the apartment may only allow one type of pets. However, except all the above situations, if the description contains "no pets", it tends to forbid any pet.

Actually, the above analysis are preliminary. In terms of the results, the strategy worked since most of the 14721 observations were classified correctly, but it is not enough to classify all observations. Next, I will use supplementary pattern to make it up. (For the details of the supplementary pattern, please refer to the appendix). Finally, it turns out that 11628 rental posts allow both cats and dogs, 925 only allow cats, 796 only allow dogs and 1372 don't allow any pet. I also added the categorical feature to my data set, and use the updated data set to analyze further.

To extract pet deposit, I applied a similar strategy. Firstly, I filter the posts which allow pets, and cut the sentences mentioning pets in those posts. This preprocess can not only simplify my following analysis, but also avoid to extract rental deposit incorrectly. Then I read these sentences to summarize several patterns. Each time I tried one pattern, and check whether there are sentences including pet deposit but left out, then read these left-out sentences again and tried another pattern to make it up, so on so forth. Finally, I obtained 7881 pet deposit values, which is ok. (The specific steps are in appendix)

As the Figure 2 shows, almost all deposits are less than 1000. Among those, three most common pets deposits are about $50, $300, and $500.

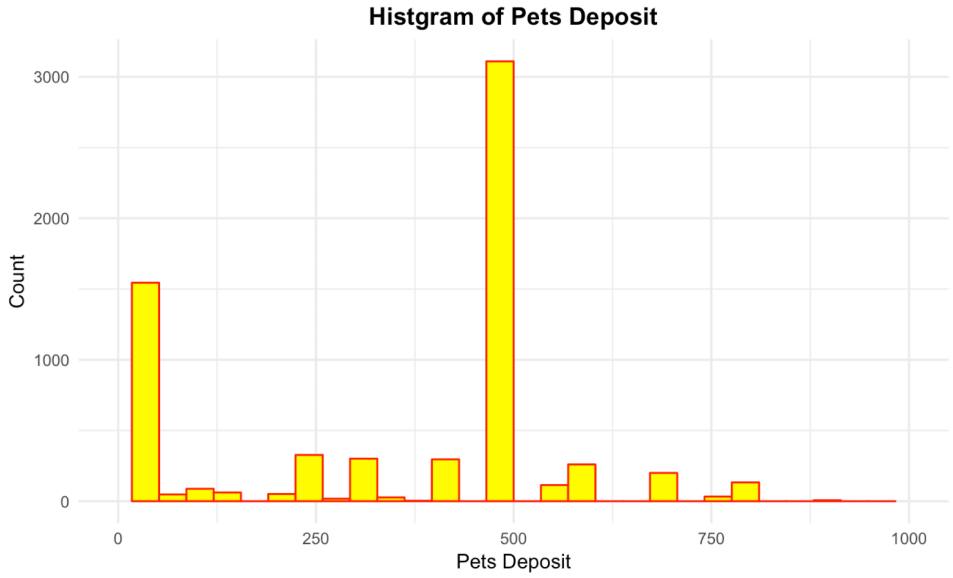

Figure 2

Before detecting whether there are any other allowed pets besides cats and dogs, I firstly read some files to find some other pets mentioned. Then I decided a strategy applied to search other pets. Initially, I just detected "birds", "fishes" and other animal names from the full descriptions, but it does not work since many avenues or places named by animal names. So,

I took the similar strategy. First step is to extract the sentences related to pets description, and then detect animal names in these sentences. As a result, several apartments also allow birds and fish, but birds should be kept in a cage, in addition, a couple of posts allow chicken, lizard and horse as well.

2.5 Categorical Feature of Heating Methods

In this part, I created a categorical variable representing the heating type of apartment. I classified both "central heating" and "heater" as heater, "fireplace" as "fireplace", in this way, apartments can be classified into three classes: with heater(also including central heating), with fireplace, or with both heater and fireplace. It turned out that 5233 posts have fireplace, 3387 have heater and 913 have both heater and fireplace. Actually, many observations neither mention fireplace nor heater, but I don't think it definitely means no heater or fireplace, so I just regarded them as missing values. Then, add this column to my data set.

For the air conditioning, there are 8239 posts have air conditioning. In summary, comparing air conditioning feature variable and heating feature variable, I got 3282 posts with both heating (refering to all heating types here) and air conditioning, while 4957 and 6251 only have air conditioning and heating respectively.

To make it more clearly, I compared the heating and air conditioning among nine regions in Figure 3 below. It shows that in most regions of San Francisco Bay Area and Sacramento, heating is much more popular than air conditioning, except sfbay_sby where there are more apartments with air conditioning than with heating. In South California, even though the number of posts with heating is slightly larger than posts with air conditioning, the difference is much smaller than that of other regions. I think this makes sense, because the temperature in Los Angeles and San Diego is higher than that in San Francisco Bay Area on average, so air conditioning may be more popular.

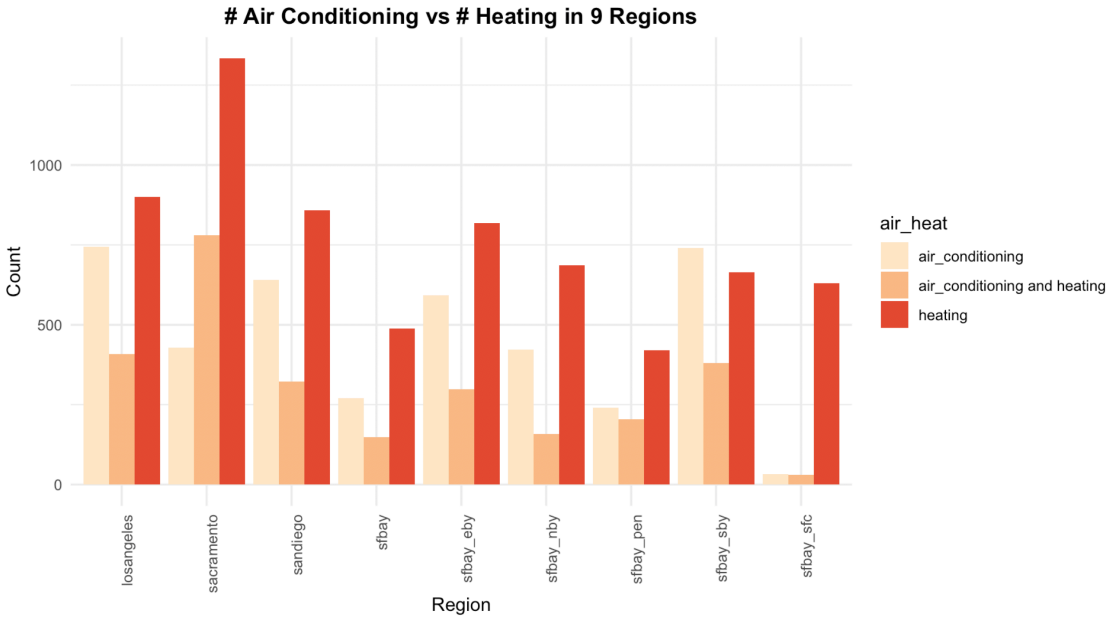

Figure 3

2.6 Posts Hiding Email Address and Phone Number

In order to filter the posts which have hidden the Email Address and Phone Number of the poster, I searched craigslist website: https://sfbay.craigslist.org/d/apts-housing-for-rent/search/sby/apa and found that the pattern “show contact info” will be included in the posts if the poster want to hide email addresses and phone numbers from web scrapers. So, I detected each posts description and got 28554 observations containing “show contact info”, that means that the majority of poster had chosen to hide their contact information.

Figure 4 compares the ratio of posts hiding contact information in each region. From this figure, we can conclude that most poster prefer to hide their contact information, except “sfbay_sfc” where almost half poster would like to show their contact information directly.

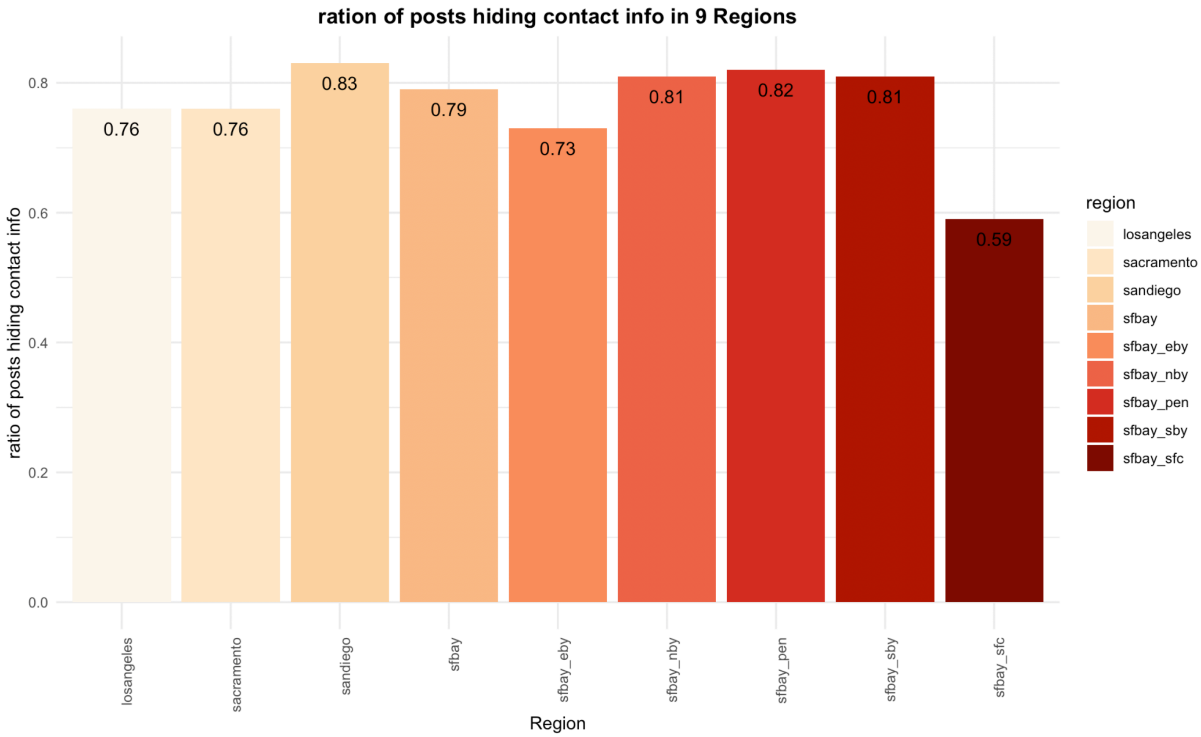

Figure 4

3 Analysis of Rental Prices

When I examined the price range, I found prices that ranged from $0 USD to $34,083,742. When these individual posts were closely examined, it appeared that the small handful of high values were listings to purchase houses rather than rent and that many of the items listed below about $500 were either daily or weekly prices or item sales that had been erroneously posted in the monthly apartment rental section of the website. Therefore, the data was subsetted for prices between $500 and $20,000 per month, which are both plausible, based on region and number of bedrooms. The higher range tended to be 4 or 5 bedrooms in cities and the lower range tended to occur in areas that weren’t prominent cities. The new subsetted data had 6934 observations, and this is the dataset that was then used for the analysis of data in the rest of the report. The price range for this is shown in the histogram Figure 5 and goes from $504 to $17,700, with a mean of $2517 with standard deviation $1238 and a median on

$2288. Therefore, a monthly rental unit in California seems to be on average about $2517 but there is quite a range.

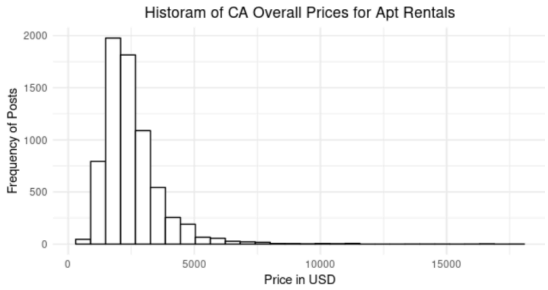

Figure 5

3.1 Pricing Based on Bedrooms and Bathrooms

Next, let us explore whether prices increased more quickly due to bathrooms or bedrooms. To do so, we fit a linear model through the data. The correlation coefficients for both were low (41.0% for bathrooms and 44.3% for bedrooms), suggesting that linear fit may not be the best regression. However, we decided to use this metric anyway. Our linear model was Y=807.3X1 + 507.7X2. Bedrooms had a higher slope (807.3 vs 507.7) and slope corresponds to how quickly the prices increases with each additional bedroom or bathroom, respectively. Therefore, bedrooms add more to price than bathrooms. Additionally, we present plots to illustrate this on the following page.

As shown in Figure 6, prices for 1,2,and 3 bedroom apartments increase with ad- ditional bathrooms. Results for 4,5, and 6 bedrooms were not included because there were very few data points for those. In the plots, the data varies slightly by number of bedrooms and you can see that the biggest distinction seems to be in 1 bedrooms between 1 bathroom versus 1.5 or 2 bathrooms. For 2 bedrooms, there are slight increases in price based on 1 through 2 bedrooms and then higher price increases once you get to 2.5 or 3 bathrooms. For 3 bedrooms, the data is more variable and 1 and 1.5 bathrooms gives comparable prices, while 2 appears lower and 2.5 to 3.5 bathrooms give higher prices.

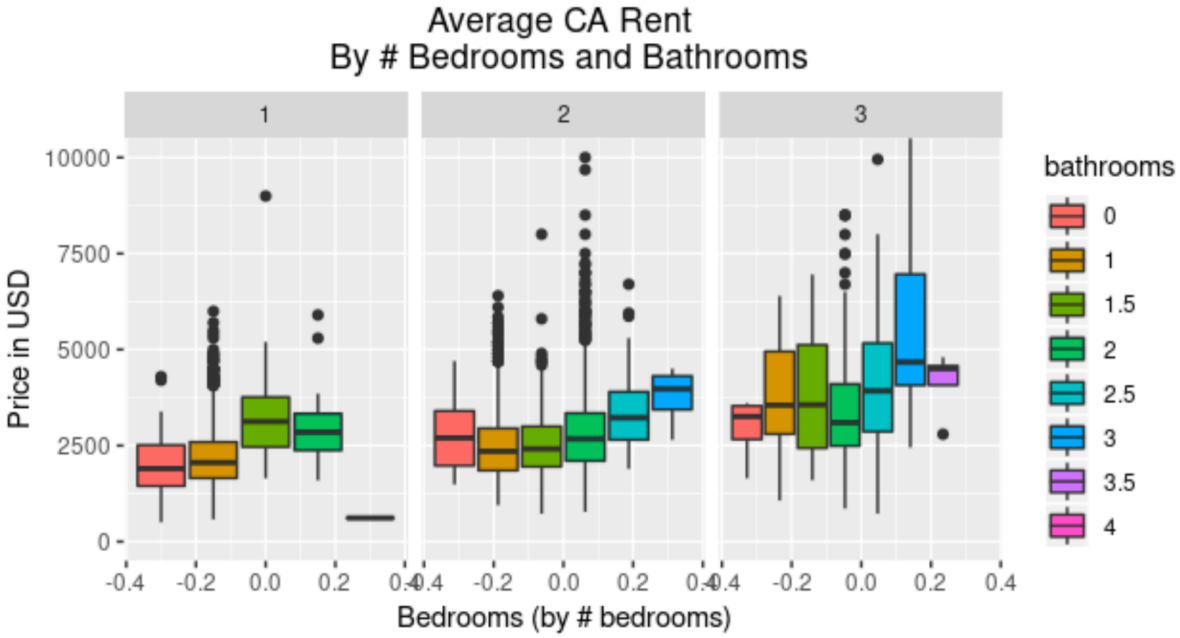

Figure 6

3.2 Pricing Based on Pet Policy

In this part, we will explore whether the price of apartments in San Francisco is affected by dog policy. Intuitively, a slight extra charge may be made in case the pet causes damage. Having done the analysis, it turns out that the average price in SF with a dog is $2360 and without one is $2200. Figure 7 compares the price of apartments that allow dogs with those do not allow dogs over different numbers of bedrooms. However, this conclusion is still suspicious, because the reason that places that allow dogs are more expensive may be just that these places tend to be bigger and/or have backyards. To clarify the suspicion, I instead compared the price per sqft of apartments that allow dogs with those do not allow dogs, as Figure 8 shows. The price difference becomes less significant, which imply that size is a confounding factor.

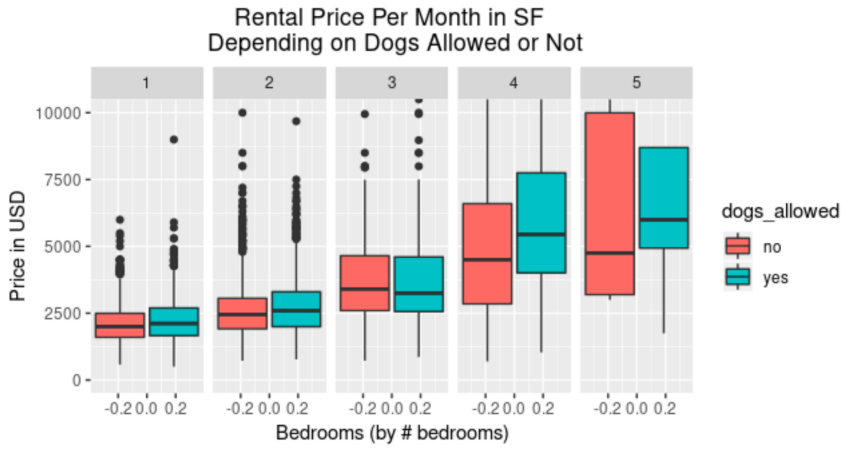

Figure 7

Figure 8

4 Further Inference

In this section, I will explore the relationship between the house features and the geographical characteristics, and make further inference by considering demo demographic map based on the online data sets provided by US Census Bureau.

4.1 Explore Rental Market at Davis

As the map below shows that the apartments distributed very sparsely. Most apartments zonally distributed in north of UC Davis. And a few apartments cluster round Davis Senior High School. There are also several apartments located in south of Downtown along highway.

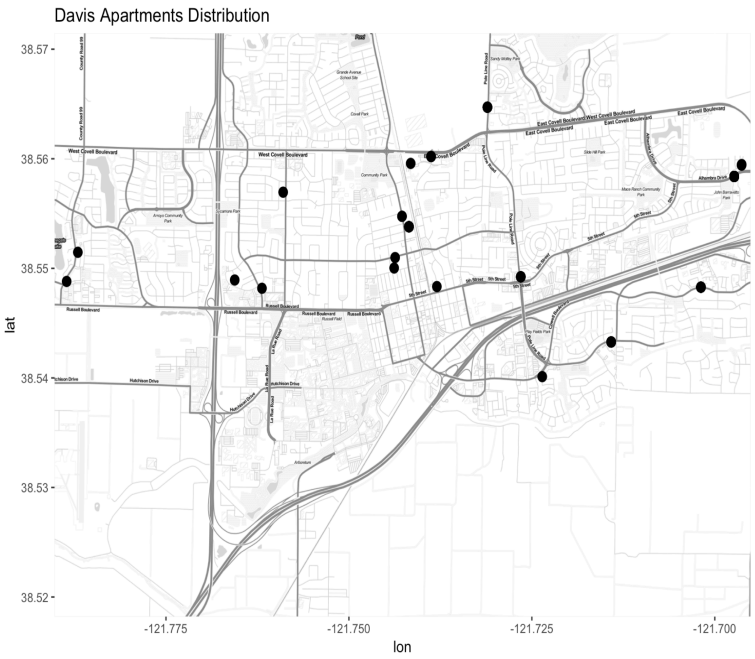

Figure 9

Next, let us explore whether the expensive apartments tend to cluster round some area, and what features of these expensive apartments tend to have. Base on the overall price distribution of apartments in Davis, I take the upper quantile 1839 dollars as the cutoff of expensive apartments. The map below shows that expensive apartments distribute along Pole Line Road and closed to the center. However, I guess the price is also influenced by the space(the number of bedrooms) of the apartments. Generally, the more bedrooms an apartment has, the more expensive the apartment is. So, I will compare Figure 10 with Figure 11.

Figure 10

In Figure 3, It seems that the expensive apartments are 2-bedroom, while some 3 or 4 bedroom apartments are not expensive. So, I guess that the expensive apartments on Figure 2 may be due to their location.

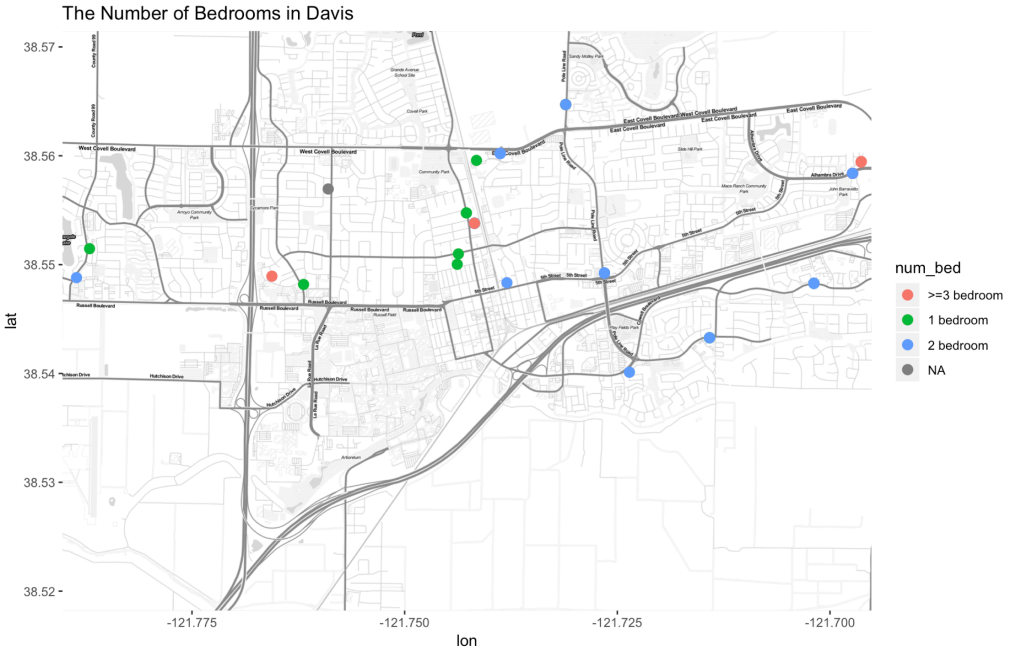

Figure 11

Figure 12 shows the distribution of apartments in Davis with various parking choices. Generally speaking, most apartments can only park off-street. However, there is an apartment near north of Pole Line Road, which is expensive according to Figure 10, having no parking place.

Actually, all of findings I mentioned above are not quite reliable, because the sample are so small that it is hard to draw convincing conclusions.

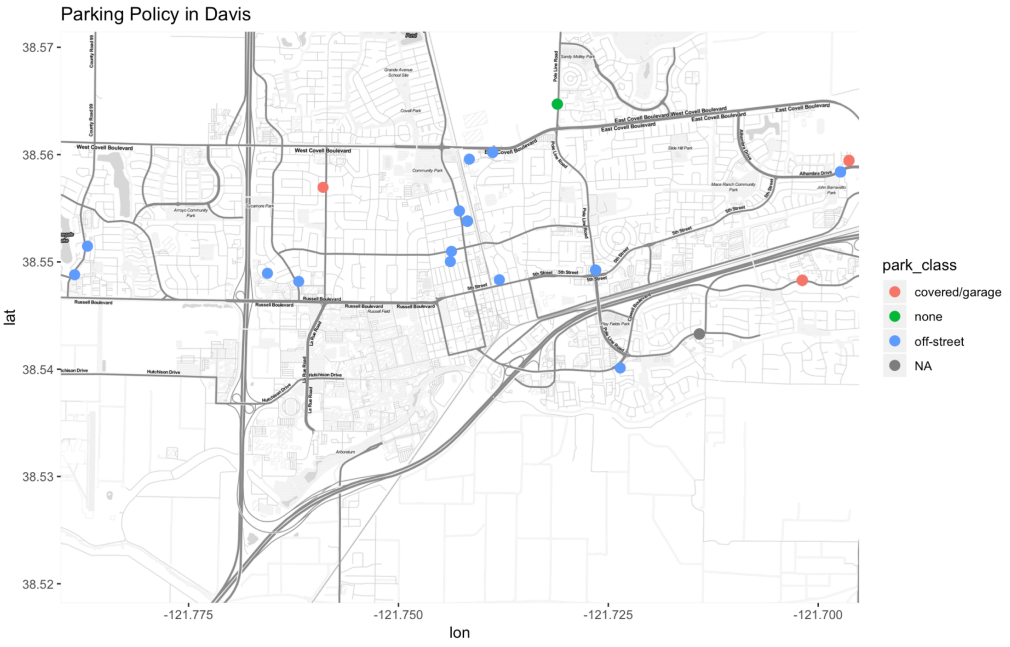

Figure 12

4.2 Explore Rental Market in San Francisco Bay Area

Figure 13 displays that the apartments rental markets around San Francisco Bay are quite active. However, the area which is far from bay has less apartments such as Contra Costa county. In particularly, San Francisco, Berkeley, Oakland, Silicon Valley and San Jose have much more apartments available than other places. I guess this may be because the larger populations in these cities. And I will confirm this point later.

Next, let us explore whether the expensive apartments tend to cluster round some area, and what features of these expensive apartments tend to have. I divided the price into 4 categories based on quantile: over 5000 dollars is extremely expensive; 2975-5000 dollars is expensive; 2275-2975 dollars is cheap; less than 2275 dollars is quite cheap. In Figure 14, the green and blue point represents expensive apartment, while the purple and red point represents cheap apartment. We can discover that the apartments along the west bank of bay tend to be more expensive than the apartments along the east ban of bay, especially, the apartments in San Francisco, Palo Alto and Silicon Valley are more likely to be expensive. I think this is mainly because that there are many high-technique companies in San Francisco and some apartments here are rent for business not for living in. Besides, since the Palo Alto is very closed to Stanford University, it is sensible that the apartments there are expensive. On the other hand, although the north of bay area is more likely to have cheap apartments, Berkeley and Oakland have more expensive apartments than its neighborhood. I think their short distance to UC Berkeley is also an important reason.

As what I discovered in Figure 14, expensive apartments mainly cluster in San Francisco, Palo Alto, Silicon Valley as well as Berkeley. However, Figure 15 below shows that there are much more blue points in San Francisco, which means that there are lots of apartments without parking place, even though these apartments are relatively more expensive. As far as I am concerned that the limited parking places in San Francisco, Berkeley and Oakland is caused by relatively small areas compared to their enormous populations. To be more specific, San Francisco has second largest population among bay area, but its area is only approximately a quarter of the area of San Jose. Similarly, Berkeley ranks 11 in terms of population, while its area is only 10.47 square mile. Therefore, due to the limited per capita living area, apartments are less likely to have spare space for parking.

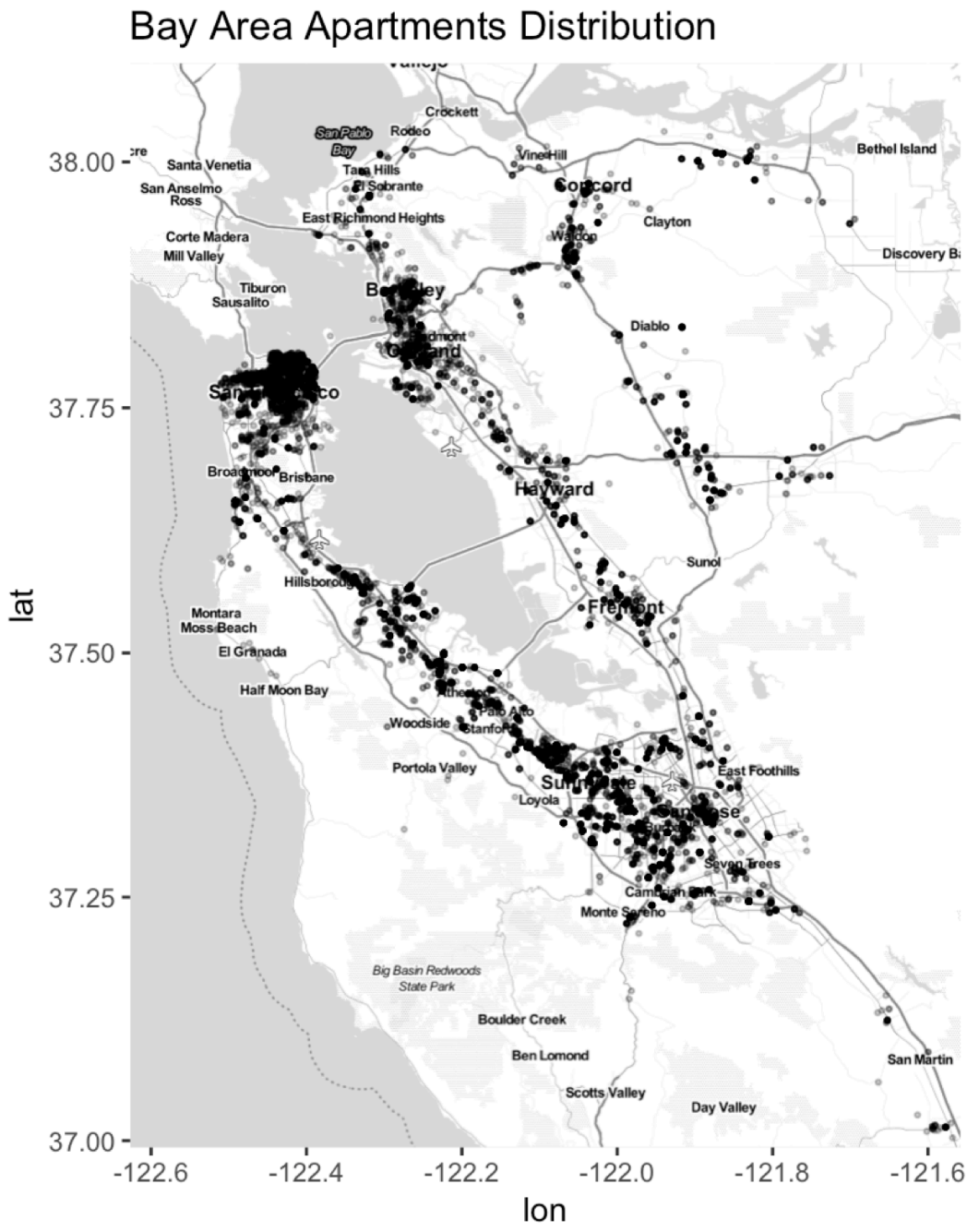

Figure 13

, San

Figure 16 illustrates the distribution apartments with various pets policies in bay area. Taking Figure 14 and Figure 15 into consideration, we can find that no parking place, no pets and high price tend to co-occur in San Francisco, Palo Alto, Berkeley and Oakland. I infer that one of the reasons is the limited space but large population in these cities. Since the large population means more demands of living apartments, the price will go up. And it may be hard to spare extra room from the small per capita living area to keep pets or parking.

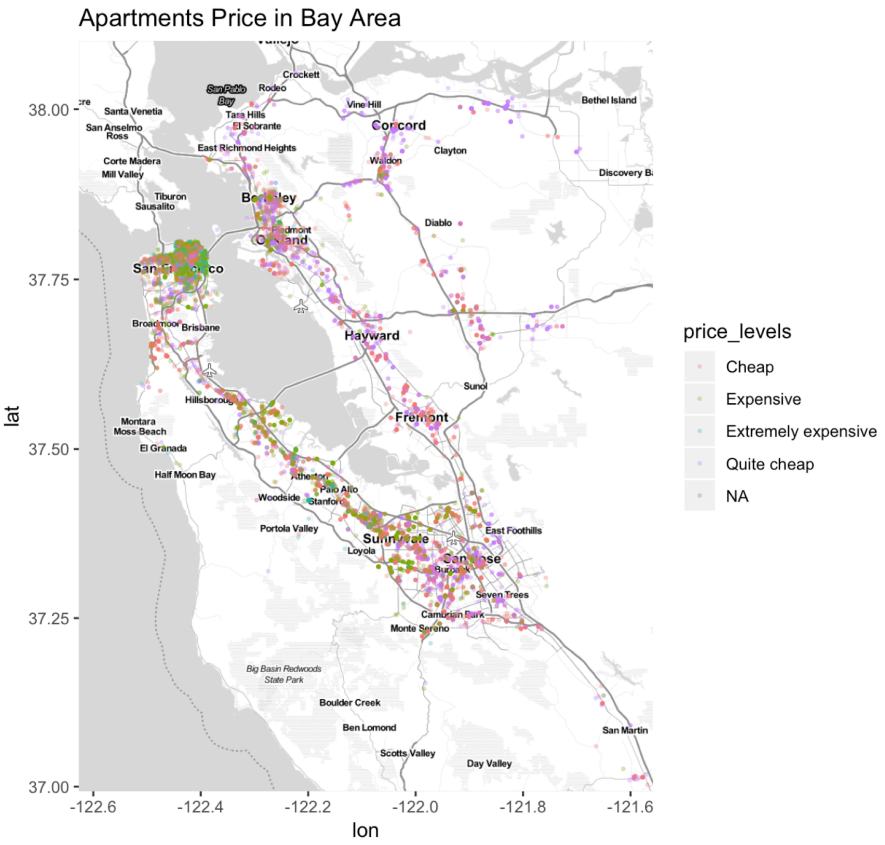

Figure 14

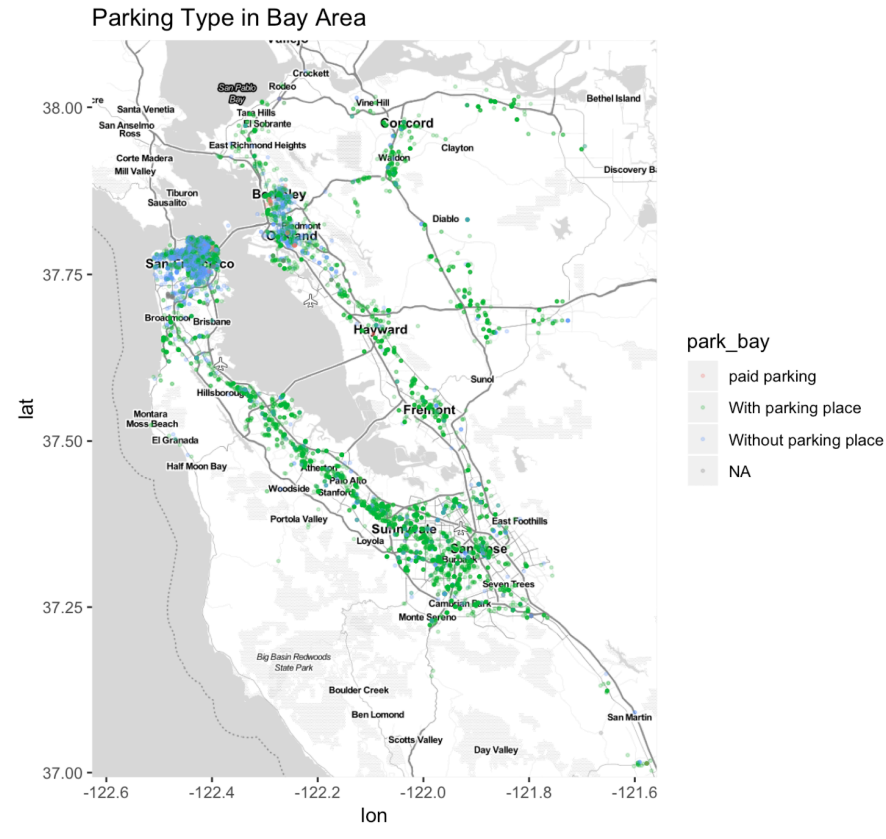

Figure 15

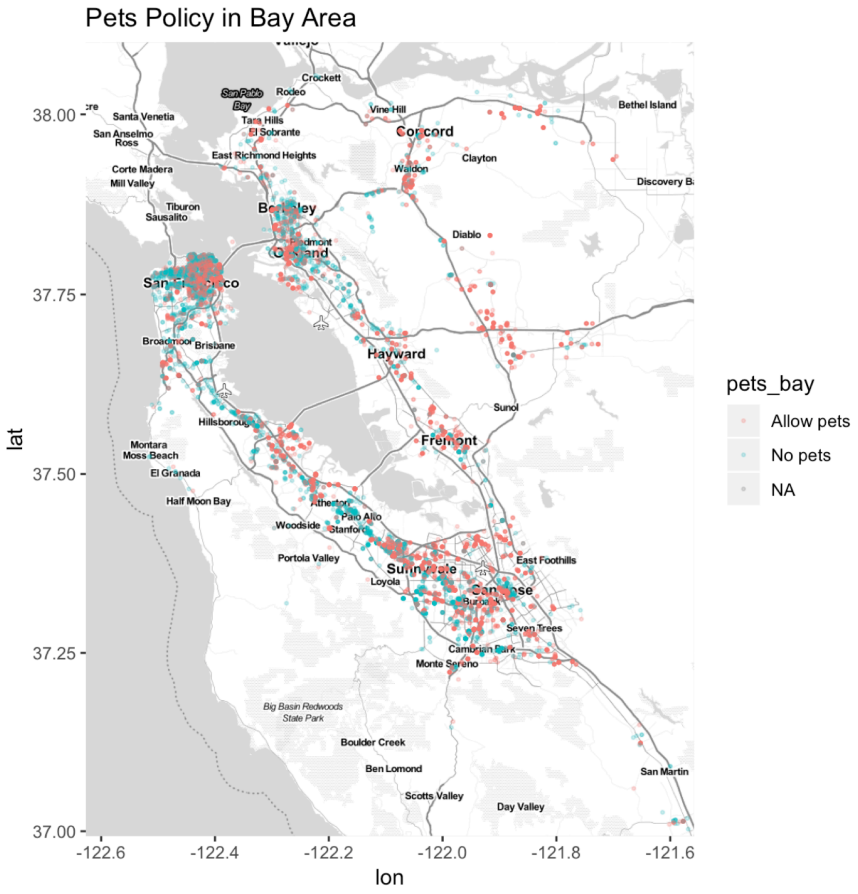

Figure 16

4.3 Further Inference by Considering Demographic Map

Since, I guess that the pattern of parking choices, pets policy as well as rental price in a city are related to the area per person. Next I will figure out whether the higher price, the less parking space, and intolerant pets policy are more likely to happen in crowded cities.

Firstly I download data set about the area of all cities in California state from American FactFinder1, then I divide the land area of each city by its total population to get the area per person. After matching these values to apartment data set, I classify area per person into 3 categories based on quantile. The map Figure 17 shows the levels of area per person of each city in bay area. The smaller the area each person has, the more crowded the city is. The area with much more blue points means there are many crowded cities, so the land will be very limited there. The most crowded area in the both bank of the bay, mainly from San Francisco to San Mateo, and from Berkeley to Hayward. Consider this with Figure 16 and Figure 15, we can find that the apartments in crowded area such as Berkeley and San Francisco tend to have no parking place, and in Figure 16, lots of apartments in crowded area also do not allow to keep pets. So, I am more confident to say that the limited parking places and intolerant pets polities may be due to the limited space. However, even though the area around Palo Alto are not crowded, most apartments don't allow tenants to keep pets. I suppose there are other

[1] https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?src=CF

factors causing this pattern, for instance, people living here are aged so they cannot spare energy to look after pets.

Compared to Davis, the southern half of the San Francisco Bay Area have more apartments. In this way, the pattern are more clear and it is more possible to figure out some potential tendency.

Finally, I will explore how the oldest populations in the southern San Francisco relate to the rental market. In my opinion, the old population means the population of over 65 years old. So I compare the percentage of over 65 population among the cities in the southern San Francisco Bay Area, and find that Walnut Creek has the oldest population and 26.6% of its population are over 65 years old. The maps below shows the distribution of old cities. Since the average percentage of aging population in US is 15.63% 2, I set it as cutoff of very old city and take those cities whose percentages of aging population are below the median value 11.20% as young city. On the map, the area with lots of red points or green points are old and very old respectively. We can say that the area on south west bank of bay from San Francisco to Atherton has a large number of old cities, and the area around Sunnyvale is also old. In particular, Stanford and Palo Alto are very old. On the east bank of bay, Berkeley has larger old population than Oakland and further east, area around Waldon is very old.

Comparing Figure 18 with Figure 14, we can find that the old or very old cities tend to have more expensive apartments such as San Francisco, Palo Alto and Berkeley. And it seems that these area are less tolerant about pets. I guess that only the elder people who are successful and have already made awesome achievements can afford the expensive apartments in these cities. And most of the elder people cannot spare energy to look after pets or they're just unwilling to be disturbed by pets.

[2] According to Wiki https://en.wikipedia.org/wiki/Demography_of_the_United_States#Ages

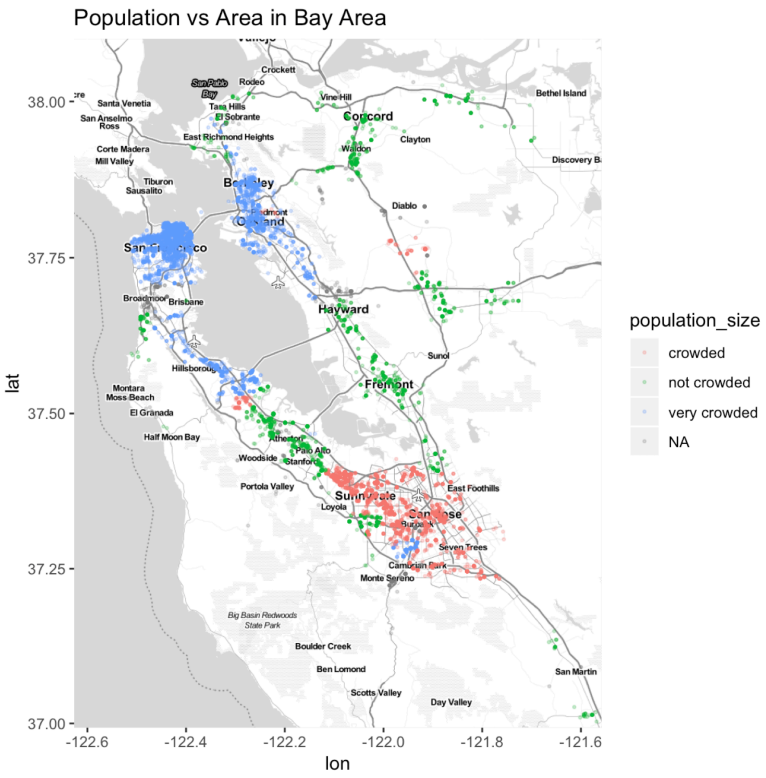

Figure 17

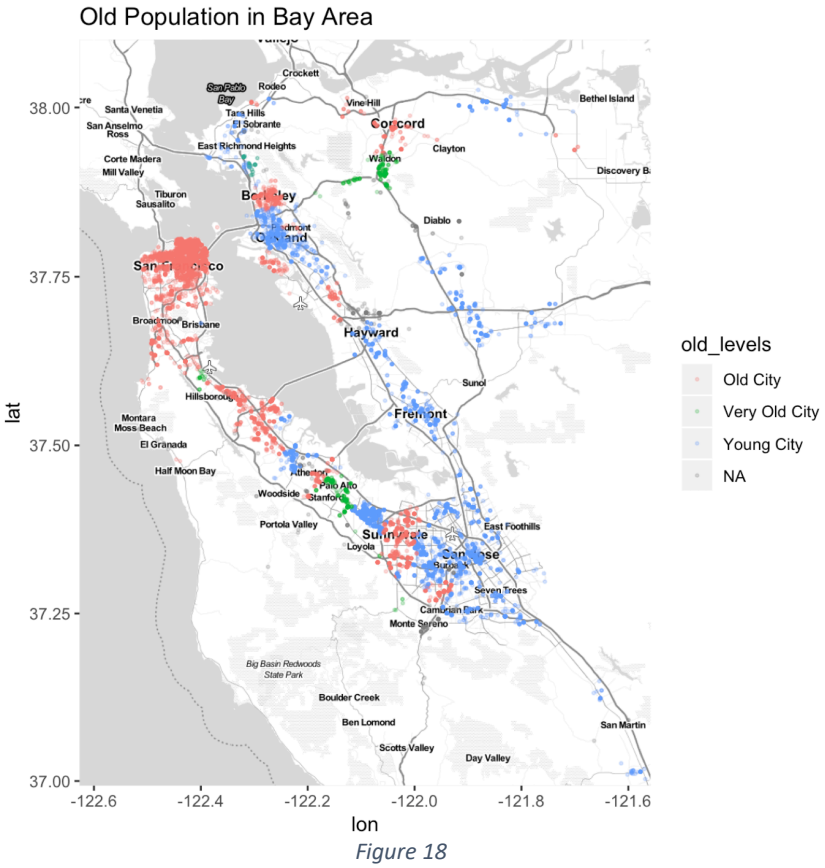

Figure 18

Appendix

```
library(stringr)
library(dplyr)
library(xtable)
library(ggrepel)

#1. write the function read_post to read a single post from a text file
read_post=function(filedr){
  post=readLines(filedr)
  n=length(post)
  # the vector include "title","date
posted","price","Latitude","Longitude","Bedrooms","Bathrooms","sqft"
  # combine the lines before Date Posted and after title as the
"description"
  features=character(9)
  features[1]=post[1]
  features[2]=paste(post[c(-1,seq(-n+6,-n,by=-1))],collapse = ";")
  features[3]=post[n-6]
  features[4]=post[n-5]
  features[5]=post[n-4]
  features[6]=post[n-3]
  features[7]=post[n-2]
  features[8]=post[n-1]
  features[9]=post[n]
  return(features)
}
read_post("messy/sfbay_eby/_eby_apa_6729057879.txt")

# test:
testnames=list.files("messy/sfbay_eby",full.names = T)
textposts=sapply(filenames, read_post)
class(textposts)
testposts=t(all_posts)
dim(testposts)
row.names(testposts)=1:5869
row.names(testposts)
colnames(testposts)=c("Title","Description",
                      "Date
posted","Price","Latitude","Longitude","Bedrooms","Bathrooms","Sqft")
testposts=as.data.frame(testposts)
head(testposts)

# 2. write a function read_all_posts to read all posts
read_all_posts=function(direction){
  # get the direction of each file
  filenames=list.files(direction,full.names = T)
  # apply read_post function to each file
  all_posts=sapply(filenames, read_post)

  # transform the list into a data frame
  all_posts=t(all_posts)
  row.names(all_posts)=1:nrow(all_posts)
  colnames(all_posts)=c("Title","Description",
                        "Date
posted","Price","Latitude","Longitude","Bedrooms","Bathrooms","Sqft")
  all_posts=as.data.frame(all_posts)
```

```
  all_posts$region=basename(direction)
  return(all_posts)
}

all_regions=list.files("messy",full.names = TRUE)
posts=lapply(all_regions,read_all_posts)
head(posts[[1]])
posts_df = do.call(rbind, posts)

# Next, I will split the attributes and discard the character parts
remove_chr=function(attri) {
  attri=as.character(attri)
  attri=str_remove(attri,"^[A-z ]+: ?" )
}
posts_df[,3:9]=as.data.frame(sapply(posts_df[,3:9],remove_chr))
head(posts_df)
levels(posts_df$Bedrooms)
levels(posts_df$Bathrooms)
posts_df[posts_df$Bedrooms=="123",]

# Save the data frame
saveRDS(posts_df,"allposts_df.rds")


## Before moving on, I want to remove the duplication firstly.

# according to the 4-8 questions, we need:
# ---rental price(in title and user-speci ed prices)
# ---deposit amount (Security Deposit)
# ---pets policy
# ---pets deposits
# ---heating
# ---air conditioning
# ---whether the position where there should have been email or phone
number is empty
# I think if it feasible i also should keep the apartment location

#6.Extract a categorical feature from each Craigslist post (any part)
that measures whether the apartment allows pets: cats, dogs, both, or
none.
posts_uni_price$Description[1:40]
# ;*Pet-Friendly;* Pet-Friendly;Pet-friendly;Pet
Friendly;Cats;dog;Type: Cats and Dogs; Max Number of Pets: 2;Pet
deposit fee max: $500
#Pets Policy: No Pets Allowed;Pet Policy;  Cats not allowed; Dogs not
allowed; Pets - Max 2 allowed, Max weight 25 lb each, Deposit
$500.00Comments:
#Pets generally accepted. The lower tenant has a small dog.No pets;Cat
Okay. Dog Negotiable
#Pet Friendly with additional deposit of $300.00 per pet (maximum 2)
and $30.00 monthly pet rent (per pet).
#   Breed restrictions; small dog breeds only and no pets over 20
pounds
#Cats okay! Sorry, no pets other than cats ($200 pet deposit required)

# maybe we can use a loop to test each pattern and then set a special
combination:
```

```
#  if the description contains cats allowed but without dogs, or maybe
also contian pet-friendly
#a description firstly passes "cats allowed", if not test "dogs", if
yes test" no cats allowed"
pets_policy=function(vec) {
  #input: description column
  #output: pets policy
  #first step is to extract sentences where pets are mentioned
  pets_reg="(pet|dog|cat)s?"
  is_pets=str_detect(vec,regex(pets_reg,TRUE))
  pet_desc=rep(NA,nrow(posts_uni_price))
  pet_desc[is_pets]=vec[is_pets]
  pet_string=str_extract(pet_desc,regex("[\\.-;].{0,100}
(pet|dog|cat)s? .{0,100}[\\.-;]",TRUE))
  #There are 14721 non-missing values, I think this is ok
  #Next, read the sentences mentioning pets and find the pattern
reference pet policy
  #-Dogs and Cats Welcome|-Pet Friendly|Onsite Dog Park|pet-
friendly|Pet Friendly|Pet Policy:Pets Allowed: Small Dogs and Cats
  #Dogs and Cats Allowed|No Pets Allowed|Pets - Max 2 allowed, Max
weight 25 lb each, Rent $25.00, Deposit $300.00Comments:"

  is_cats=str_detect(pet_string,regex("cats?
(allowed|okay|deposit)",TRUE))
  is_dogs=str_detect(pet_string,regex("small dogs?
(allowed|okay|deposit)",TRUE))
  is_pet=str_detect(pet_string,regex("pets?(-friendly| friendly| [A-Za-
z]* accepted| deposit| allowed)",
                                     ignore_case = TRUE))
  no_pet=str_detect(pet_string,regex("no pets?|pets? not allowed|pets?
not permitted",TRUE))
  pet_string[is_pet & is_dogs & !is_cats]="dogs"
  pet_string[is_pet & is_cats & !is_dogs]="cats"
  pet_string[is_pet & !is_cats & !is_dogs]="both"
  pet_string[!is_pet & is_cats & is_dogs]="both"
  pet_string[!is_pet & !is_cats & is_dogs]="dogs"
  pet_string[!is_pet & is_cats & !is_dogs]="cats"
  pet_string[no_pet & !is_cats & !is_dogs & !is_pet]="none"
  # return(pet_string) here and found that there are still some strings
  #summarize the pattern among the remaining string again
  #Pet Spa & Dog Run
  dogs=str_detect(pet_string,regex("dogs?",TRUE))
  cats=str_detect(pet_string,regex("cats?",TRUE))
  pet=str_detect(pet_string,regex("[^no] ?pets?",TRUE))
  pet_string[dogs & cats]="both"
  pet_string[dogs & !cats]="dogs"
  pet_string[!dogs & cats]="cats"
  pet_string[pet]="both"
  return(pet_string)
}
pet_policy=pets_policy(as.character(posts_uni_price$Description))
posts_uni_price$pets=factor(pet_policy)
saveRDS(posts_uni_price,"allposts_pets.rds")
table(factor(pet_policy))

# pet deposit: try to use "(?<=pets?)(.*?)(?=\$)"
#Pattern Examples:
#  $300.00 Pet Security Deposit
```

```
#  Pet friendly: $300 deposit for first pet, $200 for each additional
pet
#  Pet friendly- $500 additional deposit
#  Pet friendly $300.00 pet deposit with $35 pet fee
#  A CAT IS WELCOME WITH A $500 PET DEPOSIT
#  Pet deposit fee max: $500
#  Pet Friendly with additional deposit of $300.00 per pet
#  Dogs: $500 Deposit, $25/monthCats: $500 Deposit
#  $500 pet deposit
#  Pets - Max 2 allowed, Max weight 25 lb each, Rent $25.00, Deposit
$300.00
pets_deposit=function(vec) {
  #input: description column
  #output: pets policy
  #first step is to extract sentences where pets are mentioned
  pets_reg="(pet|dog|cat)s?"
  is_pets=str_detect(vec,regex(pets_reg,TRUE))
  pet_desc=rep(NA,nrow(posts_uni_price))
  pet_desc[is_pets]=vec[is_pets]
  pet_string=str_extract(pet_desc,regex("[\\.-;].{0,100}
(pet|dog|cat)s? .{0,100}[\\.-;]",TRUE))
  #There are 14721 non-missing values, I think this is ok
  #Next, read the sentences mentioning pets and find the pattern
reference pet policy
  #-Dogs and Cats Welcome|-Pet Friendly|Onsite Dog Park|pet-
friendly|Pet Friendly|Pet Policy:Pets Allowed: Small Dogs and Cats
  #Dogs and Cats Allowed|No Pets Allowed|Pets - Max 2 allowed, Max
weight 25 lb each, Rent $25.00, Deposit $300.00Comments:"
  pet_string[pet_policy=="none"]=NA
  pet_deposit1=str_match(pet_string,regex("\\$([0-9[,.]]+) (per )?pet
(security )?deposit",TRUE))[,2]
  pet_string[!is.na(pet_deposit1)]=pet_deposit1[!is.na(pet_deposit1)]
  #Dogs: $500 Deposit, $25| Pet Fees; Deposit: $500 (up to 2
pets) ;|Will consider pet with deposit. $2600 per month
  #$500 deposit for Dog;$300 deposit for Cat|Small pets welcomed with
additional $250 deposit!
  #Owner Will Consider Pet With Additional Deposit of $300/Cat $500/
  #We welcome cats only on approval with an additional deposit of $350
per cat and $40 pet rent per cat.
  #Dogs- $500 deposit and monthly pet fee of $25;Cats-$300 deposit
  dog_deposit=str_match(pet_string,regex("dogs?[-:] \\$([0-9[,.]]+)
deposit",TRUE))[,2]
  pet_string[!is.na(dog_deposit)]=dog_deposit[!is.na(dog_deposit)]
  cat_deposit=str_match(pet_string,regex("[cats?|pets?] .{0,30} \\$([0-
9[,.]]+)",TRUE))[,2]
  pet_string[!is.na(cat_deposit)]=cat_deposit[!is.na(cat_deposit)]
  pet_deposit2=str_match(pet_string,regex("\\$?([0-9[,.]]+) pets?
deposit",TRUE))[,2]
  pet_string[!is.na(pet_deposit2)]=pet_deposit2[!is.na(pet_deposit2)]
  pet_string[nchar(pet_string)>=10]=NA
  return(pet_string)
}
pet_deposit=pets_deposit(as.character(posts_uni_pets$Description))
pet_deposit=str_remove(pet_deposit,",")
pet_deposit=str_remove(pet_deposit,"\\.[0-9]+")
posts_uni_price$pet_deposit=pet_deposit
posts_uni_price$pet_deposit=as.numeric(pet_deposit)
saveRDS(posts_uni_price,"allposts_pets_deposit.rds")
```

```
ggplot(posts_uni_price[!is.na(pet_deposit),])+
  geom_histogram(aes(pet_deposit),fill="yellow",col="red")+
  theme_minimal()+
  xlim(0,1000)+
  labs(title="Histgram of Pets Deposit",x="Pets Deposit",y="Count")+
  theme(plot.title = element_text(hjust = 0.5,face = "bold"))

##other pets
other_pets=function(vec) {
  #input: description column
  #output: pets policy
  #first step is to extract sentences where pets are mentioned
  pets_reg="(pet|dog|cat)s?"
  is_pets=str_detect(vec,regex(pets_reg,TRUE))
  pet_desc=rep(NA,nrow(posts_uni_price))
  pet_desc[is_pets]=vec[is_pets]

pet_string=str_extract(pet_desc,regex("[\\.].{0,100}(pet|dog|cat)s? .{0
,150}[\\.]",TRUE))
  other_pet=str_detect(pet_string,
                       regex("(bird|horse|iguana|chicken)[es]?",TRUE))
  return(pet_string[other_pet])
  #There are 14721 non-missing values, I think this is ok
}
pets_other=other_pets(as.character(posts_uni_price$Description))
pets_other[!is.na(pets_other)]
# Birds must be kept in a cage;chickens;Birds and Fish Allowed;BIRDS
AND LIZARD ARE OK.
# Horses welcome

#7. heating
posts_uni_price$Description=as.character(posts_uni_price$Description)
posts_uni_price$Description[21:40]
head(posts_uni_price$Description[str_detect(posts_uni_price$Description
,regex("heater",TRUE))])
# Air Conditioning Ceiling Fan(s)
#along with central heating and air inside
#central heating/air conditioning
#Air Conditioning
#Apartment Features:;-Air Conditioning
#Central Air Conditioner
#Central Air or AC Wall Unit
description=as.character(posts_uni_price$Description)
heating=rep(NA,nrow(posts_uni_price))
is_heater=str_detect(description,regex("heater|central heating",TRUE))
is_fireplace=str_detect(description,regex("fireplace",TRUE))
heating[is_heater & is_fireplace]="heater and fireplace"
heating[is_heater & !is_fireplace]="heater"
heating[!is_heater & is_fireplace]="fireplace"
table(heating)
posts_uni_price$heating=heating

aircondition=str_detect(posts_uni_price$Description,regex("air
(conditioning|conditioner|inside)",TRUE))
centralair=str_detect(posts_uni_price$Description,regex("central air|
ac ",TRUE))
air=aircondition|centralair
```

```
allheating=!is.na(heating)
table(air,allheating)
sum(air)
sum(allheating)
air_heat=rep(NA, length(allheating))
air_heat[allheating==TRUE & aircondition==TRUE]="air_conditioning and
heating"
air_heat[allheating==TRUE & aircondition==FALSE]="heating"
air_heat[allheating==FALSE & aircondition==TRUE]="air_conditioning"
posts_uni_price$air_heat=air_heat
posts_uni_heating=posts_uni_price[!is.na(air_heat),]
ggplot(posts_uni_heating)+
  geom_bar(aes(region,fill=air_heat),position = "dodge")+
  theme_minimal()+
  labs(title="# Air Conditioning vs # Heating in 9
Regions",x="Region",y="Count")+
  theme(plot.title = element_text(hjust = 0.5,face = "bold"))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  scale_fill_brewer(palette = "OrRd")

#8.
hide=str_detect(posts_uni_price$Description,regex("show contact
info",TRUE))
table(hide)

posts_uni_price$hide=hide

hide_posts=posts_uni_price %>%
  group_by(region) %>%
  summarize(hide_ratio=round(sum(hide)/length(hide),2))
ggplot(hide_posts,aes(x=region, y=hide_ratio,fill=region))+
  geom_bar(stat = "identity")+
  theme_minimal()+
  labs(title="ration of posts hiding contact info in 9
Regions",x="Region",y="ratio of posts hiding contact info")+
  theme(plot.title = element_text(hjust = 0.5,face = "bold"))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  geom_text(aes(label=hide_ratio),vjust=2,color="black",size=4)+
  scale_fill_brewer(palette = "OrRd")



cl <- readRDS("cl_apartments.rds")

# 1.(1) the area of UC Davis is (-121.746582,38.542454,-
121.745613,38.546288)
# according a cluster of apartments are near ucdavis
davis = c(
  -121.790225, 38.518210, # bottom left
  -121.695120, 38.571458 # top right
)
sum(cl$city=="Davis",na.rm = TRUE)
cl_davis=cl[cl$city=="Davis",]
library(ggmap)
citation("ggmap")
davismap=get_stamenmap(davis,zoom = 15,maptype = "toner-lite") #maptype
= "toner-lite"
# apart apartments near UCD from those away from UCD
```

```
ggmap(davismap)+geom_point(aes(longitude,latitude),size=3,cl_davis)+lab
s(title = "Davis Apartments Distribution")

# compare the price of apartments in Davis
summary(cl_davis$price,na.rm=TRUE)
price_davis=cl_davis$price
price_davis[price_davis<1839]=1
price_davis[price_davis>=1839]=2
price_davis[price_davis==2]="Expensive"
price_davis[price_davis==1]="Not expensive"
cl_davis$price_davis=factor(price_davis)
ggmap(davismap)+geom_point(aes(longitude,latitude,color=price_davis),si
ze=3,cl_davis)+labs(title = "Apartments Price in Davis")

# compare the types of apartments in UC Davis
table(cl_davis$bedrooms,cl_davis$bathrooms)
num_bed=cl_davis$bedrooms
num_bed[num_bed==1]="1 bedroom"
num_bed[num_bed==2]="2 bedroom"
num_bed[num_bed==3|num_bed==4]=">=3 bedroom"
cl_davis$num_bed=factor(num_bed)
ggmap(davismap)+geom_point(aes(longitude,latitude,color=num_bed),size=3
,cl_davis)+labs(title="The Number of Bedrooms in Davis")

# compare the parking choice in ucdavis
table(cl_davis$parking)
park_class=as.character(cl_davis$parking)
park_class[park_class=="covered"|park_class=="garage"]="covered/garage"
park_class[park_class=="none"]="none"
park_class[park_class=="off-street"]="off-street"
cl_davis$park_class=factor(park_class)
ggmap(davismap)+geom_point(aes(longitude,latitude,color=park_class),siz
e=3,cl_davis)+labs(title = "Parking Policy in Davis")



# finding and limitations is that the data size is too small to be
convincing.
# I want to border the uc davis on ggmap, how can i do that??

#2.
cl_bay=cl[cl$county=="San Francisco"|cl$county=="San
Mateo"|cl$county=="Santa Clara"|
            cl$county=="Alameda"|cl$county=="Contra Costa",]
c_bay=c(-122.626913,36.9931,-121.5583,38.1)
m_bay=get_stamenmap(c_bay,maptype = "toner-lite")
# The latitude and longitude of the San Francisco Bay Area:
# san francisco:-122.7661,37.6403,-122.2818,37.9298
# San Mateo:-122.5882,37.0479,-122.0817,37.7084
# Santa Clara: -122.2027,36.8931,-121.2083,37.4846
# Alameda:-122.373749,37.453949,-121.469093,37.906689
# Contra Costa:-122.441505,37.718479,-121.534271,38.104511
ggmap(m_bay)+geom_point(aes(longitude,latitude),cl_bay,size=0.5,alpha=0
.25)+labs(title = "Bay Area Apartments Distribution")
# compare the price in bay area
range(cl_bay$price,na.rm = TRUE)
cl_bay=cl_bay[cl_bay$price<10000 & cl_bay$price>500,]
table(cl_bay$pets)
```

```
table(cl_bay$parking)
table(cl_bay$bedrooms)
summary(cl_bay$price,na.rm = TRUE)
price_levels=cl_bay$price
price_levels[price_levels<=2275]=1
price_levels[price_levels<=2975 & price_levels>2275]=2
price_levels[price_levels<5000 & price_levels>2975]=3
price_levels[price_levels>=5000 & is.na(price_levels)==FALSE]=4
price_levels[price_levels==1]="Quite cheap(<2275)"
price_levels[price_levels==2]="Cheap([2275,2975])"
price_levels[price_levels==3]="Expensive([2975,5000])"
price_levels[price_levels==4]="Extremely expensive(>5000)"
cl_bay$price_levels=factor(price_levels)
ggmap(m_bay)+geom_point(aes(longitude,latitude,color=price_levels),cl_b
ay,size=0.5,alpha=0.25)+labs(title = "Apartments Price in Bay Area")

# compare the number of bedrooms in bay area
table(cl_bay$bedrooms)
bed_bay=cl_bay$bedrooms
bed_bay[bed_bay==0|bed_bay==1|bed_bay==2]="Small(<=2 bedrooms)"
bed_bay[bed_bay==3|bed_bay==4|bed_bay==5|bed_bay==6]="Big(>=3
bedrooms)"
cl_bay$bed_bay=factor(bed_bay)
ggmap(m_bay)+geom_point(aes(longitude,latitude,color=bed_bay),cl_bay,si
ze=0.5,alpha=0.25)+labs(title="The Number of Bedrooms in Bay Area")

# compare the parking choice in bay area
table(cl_bay$parking)
park_bay=as.character(cl_bay$parking)
park_bay[park_bay=="covered"|park_bay=="garage"|park_bay=="off-
street"]="With parking place"
park_bay[park_bay=="street"|park_bay=="none"]="Without parking place"
park_bay[park_bay=="paid"|park_bay=="valet"]="paid parking"
cl_bay$park_bay=factor(park_bay)
ggmap(m_bay)+geom_point(aes(longitude,latitude,color=park_bay),cl_bay,s
ize=0.5,alpha=0.25)+labs(title = "Parking Type in Bay Area")

# compare the pets policy in bay area
table(cl_bay$pets)
pets_bay=as.character(cl_bay$pets)
pets_bay[pets_bay=="both"|pets_bay=="cats"|pets_bay=="dogs"]="Allow
pets"
pets_bay[pets_bay=="none"|pets_bay=="negotiable"]="No pets"
cl_bay$pets_bay=factor(pets_bay)
ggmap(m_bay)+geom_point(aes(longitude,latitude,color=pets_bay),cl_bay,s
ize=0.5,alpha=0.25)+labs(title="Pets Policy in Bay Area")

#(1) which area is more likely to have expensive house?
#(2) which area is more likely to have big house?


#3.
sfann<-
read.csv("2010_census_data/DEC_10_SF1_SF1DP1_with_ann.csv",header =
TRUE)
sfann=sfann[,names(sfann)!="HD02_S020"]

# The "HD02_S020" variable is repeated!! we can delete it!
```

```
# what's the meaning of Race alone or in combination with one or more
other races: [4] - White
# this census contains:
# (1) the size and percentage of population in different age range
(every 4 years is a period)
# (2) the size and percentage of male population in different age range
# (3) the size and percentage of female.......
# (4) the size and percentage of one race......(white,black,asian,etc.)
# (5).........................two races.......
# (6).........................race alone or in combination with one
or more other races
# (7)HISPANIC OR LATINO AND RACE
# (8)RELATIONSHIP:householder

sfann[1,]
#HD02_S025
#GEO.display.label

## I find that the census data is like a misture of two data sets, one
is based on cities while the other is based on "CDP"
## So, in case of duplication, I only choose the census data of cities
and match them with apartment data set according to cities' names.
library(stringr)
sfann$GEO.display.label=str_remove_all(sfann$GEO.display.label,",
California")
cities=sfann[endsWith(sfann$GEO.display.label,"city"),]
cities$GEO.display.label=str_remove_all(cities$GEO.display.label,"
city")
cities$GEO.display.label=str_remove_all(cities$GEO.display.label,"
city")
cities$GEO.display.label

bay_area=cl[cl$county=="San Francisco"|cl$county=="San
Mateo"|cl$county=="Santa Clara"|
              cl$county=="Alameda"|cl$county=="Contra Costa",]
census_cities=merge(bay_area,cities[,c("GEO.display.label","HD02_S025",
"HD01_S001")],by.x = "city",by.y = "GEO.display.label",all.x = TRUE)
table(is.na(census_cities$HD02_S025),is.na(census_cities$city))
sort(table(census_cities$city[is.na(census_cities$city)==FALSE
&is.na(census_cities$HD02_S025)==TRUE]),decreasing = TRUE)
# After checking the results of merging process, I found that some
cities in apartment data set actually correspond to town in census data
set.
# In case of potential repeated values, i get rid of these values.

census_cities$HD02_S025=as.numeric(as.character(census_cities$HD02_S025
))
census_cities$HD01_S001=as.numeric(as.character(census_cities$HD01_S001
))
head(census_cities$HD02_S025)
head(census_cities$HD01_S001)
summary(census_cities$HD02_S025)
census_cities[which.max(census_cities$HD02_S025),]
#Walnut Creek 26.6

old_levels=census_cities$HD02_S025
old_levels[old_levels<11.70]=0
old_levels[old_levels>=11.70 & old_levels<15.63]=1
```

```
old_levels[old_levels>=15.63]=2
old_levels[old_levels==0]="Young City"
old_levels[old_levels==1]="Old City"
old_levels[old_levels==2]="Very Old City"
table(old_levels)
census_cities$old_levels=factor(old_levels)
head(census_cities)
ggmap(m_bay)+geom_point(aes(longitude,latitude,color=old_levels),census
_cities,size=0.5,alpha=0.25)+labs(title="Old Population in Bay Area")
sum(table(bay_area$city)!=0)

# Import the data about the land area of each city in California
# Calculate the land area per person in each city
# Classfy cities into 3 categoties based on their average land area and
merge this variable to apartments data set.
area=read.csv("DEC_00_SF1_GCTPH1.ST10/DEC_00_SF1_GCTPH1.ST10_with_ann.c
sv",header = TRUE)
head(area)
area$GCT_STUB.display.label.1=str_remove_all(area$GCT_STUB.display.labe
l.1,", California")
area$GCT_STUB.display.label.1
area_cities=area[endsWith(area$GCT_STUB.display.label.1,"city"),]
area_cities$GCT_STUB.display.label.1=str_remove_all(area_cities$GCT_STU
B.display.label.1," city")
area_cities$GCT_STUB.display.label.1
census_area=merge(census_cities,area_cities[,c("GCT_STUB.display.label.
1","HC06")],by.x = "city",by.y ="GCT_STUB.display.label.1",all.x =
TRUE )
census_area$HC06=as.numeric(as.character(census_area$HC06))
census_area$per_area=census_area$HC06/census_area$HD01_S001*10000
summary(census_area$per_area)
population_size=census_area$per_area
population_size[population_size<1.5662]=0
population_size[population_size>=1.5662 & population_size<1.8764]=0.1
population_size[population_size>=1.8764]=0.2
population_size[population_size==0]="very crowded"
population_size[population_size==0.1]="crowded"
population_size[population_size==0.2]="not crowded"
census_area$population_size=factor(population_size)
ggmap(m_bay)+geom_point(aes(longitude,latitude,color=population_size),c
ensus_area,size=0.5,alpha=0.25)+labs(title="Population vs Area in Bay
Area")
```