R Project

Junhao Ou & 21200066

2022-11-16

Part 1: Analysis

In this part of the project, I analyzed residential real estate data of USA to explore the historical house prices by state, city, residential property area and some other attributions. The data was collected on 27/Jun/2021 by ZenRows and the original data set is available at: https://www.zenrows.com/datasets/us-real-estate.

1. Loading data

```
#Load dataset
df_raw = read.csv("us-cities-real-estate-sample-zenrows.csv")
#Check the structure of the data set.
str(df_raw)
```

```
## 'data.frame':    10000 obs. of  47 variables:
##  $ zpid                     : int  115423997 2100098805 14543206 72977167 17599151 107827825 206997
##  $ id                       : int  115423997 2100098805 14543206 72977167 17599151 107827825 206997
##  $ providerListingId        : chr  "NULL" "NULL" "NULL" "NULL" ...
##  $ imgSrc                   : chr  "https://photos.zillowstatic.com/fp/688dbfb9d9af6a37bb906165abc5
##  $ hasImage                 : chr  "TRUE" "TRUE" "TRUE" "TRUE" ...
##  $ detailUrl                : chr  "https://www.zillow.com/homedetails/1053-Lutheran-Church-Rd-Bard
##  $ statusType               : chr  "FOR_SALE" "FOR_SALE" "FOR_SALE" "FOR_SALE" ...
##  $ statusText               : chr  "House for sale" "Active" "Townhouse for sale" "House for sale"
##  $ countryCurrency          : chr  "$" "$" "$" "$" ...
##  $ price                    : chr  "$330,000" "$99,900" "$390,000" "$254,900" ...
##  $ unformattedPrice         : chr  "330000" "99900" "390000" "254900" ...
##  $ address                  : chr  "1053 Lutheran Church Rd, Bardstown, KY 40004" "0 Old Swanzey Rd
##  $ addressStreet            : chr  "1053 Lutheran Church Rd" "0 Old Swanzey Rd" "1718 Woodcliff Ct
##  $ addressCity              : chr  "Bardstown" "Chesterfield" "Atlanta" "Wilmington" ...
##  $ addressState             : chr  "KY" "NH" "GA" "DE" ...
##  $ addressZipcode           : int  40004 3443 30329 19809 91730 24431 39531 73090 89701 89048 ...
##  $ isUndisclosedAddress     : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ beds                     : chr  "3" "NULL" "3" "3" ...
##  $ baths                    : chr  "3" "NULL" "3" "1" ...
##  $ area                     : chr  "2054" "NULL" "2154" "1025" ...
##  $ latitude                 : chr  "37.855387" "42.88115" "33.832699" "39.754839" ...
##  $ longitude                : chr  "-85.531778" "-72.399085" "-84.327653" "-75.50329" ...
##  $ isZillowOwned            : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ variableDataType         : chr  "DAYS_ON" "DAYS_ON" "PRICE_REDUCTION" "PRICE_REDUCTION" ...
##  $ variableDataText         : chr  "1 day on Zillow" "36 days on Zillow" "$5,000 (Jun 17)" "$10,000
```

```
##  $ variableDataIsFresh       : chr  "NULL" "NULL" "NULL" "NULL" ...
##  $ badgeInfo                 : chr  "ForSale" "ForSale" "NULL" "ForSale" ...
##  $ pgapt                     : chr  "For Sale (Broker)" "For Sale (Broker)" "ForSale" "For Sale (Brol
##  $ sgapt                     : chr  "291700" "NULL" "For Sale (Broker)" "255100" ...
##  $ zestimate                 : chr  "FALSE" "FALSE" "397900" "FALSE" ...
##  $ shouldShowZestimateAsPrice: chr  "FALSE" "FALSE" "FALSE" "FALSE" ...
##  $ has3DModel                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ hasVideo                  : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ isHomeRec                 : chr  "NULL" "Listing provided by NEREN" "FALSE" "NULL" ...
##  $ info2String               : chr  "https://photos.zillowstatic.com/fp/44097c1919ccfd9c21615a22d298
##  $ info3String               : chr  "Demaree & Hubbard" "NULL" "https://photos.zillowstatic.com/fp/d
##  $ brokerName                : chr  "TRUE" "FALSE" "Opendoor Brokerage, LLC" "TRUE" ...
##  $ hasAdditionalAttributions : chr  "FALSE" "FALSE" "TRUE" "FALSE" ...
##  $ isFeaturedListing         : logi  TRUE TRUE FALSE TRUE FALSE FALSE ...
##  $ list                      : logi  FALSE FALSE TRUE FALSE TRUE TRUE ...
##  $ relaxed                   : chr  "NULL" "NULL" "FALSE" "NULL" ...
##  $ hasOpenHouse              : chr  "NULL" "NULL" "NULL" "NULL" ...
##  $ openHouseStartDate        : chr  "NULL" "NULL" "NULL" "NULL" ...
##  $ openHouseEndDate          : chr  "NULL" "NULL" "NULL" "NULL" ...
##  $ openHouseDescription      : chr  "" "" "NULL" "" ...
##  $ info6String               : chr  "" "" "NULL" "" ...
##  $ X                         : chr  "" "" "" "" ...
```

The dataset contains a wide range of features. In the following step, I will select the features of interest in this project for the analysis.

2. Feature selection

```
#Select the features potentially helpful for analysis.
#Here I used `select` function from package `dplyr` to select the columns I wanted.
citation("dplyr")
```

```
##
## To cite package 'dplyr' in publications use:
##
##   Wickham H, François R, Henry L, Müller K (2022). _dplyr: A Grammar of
##   Data Manipulation_. R package version 1.0.10,
##   <https://CRAN.R-project.org/package=dplyr>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {dplyr: A Grammar of Data Manipulation},
##     author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},
##     year = {2022},
##     note = {R package version 1.0.10},
##     url = {https://CRAN.R-project.org/package=dplyr},
##   }
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
df_extracted = select(df_raw, c('id', 'statusText', 'addressStreet', 'addressCity', 'addressState', 'ar
head(df_extracted,20)
```

```
##            id                  statusText           addressStreet  addressCity
## 1   115423997             House for sale 1053 Lutheran Church Rd    Bardstown
## 2  2100098805                      Active        0 Old Swanzey Rd Chesterfield
## 3    14543206         Townhouse for sale    1718 Woodcliff Ct NE      Atlanta
## 4    72977167             House for sale        5 W Salisbury Dr   Wilmington
## 5    17599151             House for sale     7471 Matterhorn Ave    Cucamonga
## 6   107827825             House for sale       153 Black Bear Ln      Crimora
## 7  2069972083 Multi-family home for sale  354 Jim Money Rd LOT 1       Biloxi
## 8    52531107             House for sale           317 N Elm Ave   Union City
## 9     6870557                        New          1792 Walker Dr  Carson City
## 10  220809988        Lot / Land for sale        4041 S Cliff Ave      Pahrump
## 11 2077210521        Lot / Land for sale       LOT 14 N Shore Dr    La Pointe
## 12    7097538         Townhouse for sale            2546 Swan Ln    Las Vegas
## 13   88043528             House for sale         542 Hamilton St     Columbus
## 14  157957709         Townhouse for sale     43571 Helmsdale Ter    Chantilly
## 15   67984107             House for sale             923 12th St       Monroe
## 16   42782595             House for sale         1905 Laurel Ave  Panama City
## 17  110335022             House for sale       11 E 2nd North St  Green River
## 18  145250778 Multi-family home for sale           6600 Pryer Ln      Norfolk
## 19 2105450764        Lot / Land for sale            0 Windham Rd     Brooklyn
## 20 2074024447        Lot / Land for sale          50 Mill Run Rd   Fort Ashby
##    addressState area hasAdditionalAttributions isFeaturedListing  latitude
## 1            KY 2054                     FALSE              TRUE 37.855387
## 2            NH NULL                     FALSE              TRUE  42.88115
## 3            GA 2154                      TRUE             FALSE 33.832699
## 4            DE 1025                     FALSE              TRUE 39.754839
## 5            CA 1322                     FALSE             FALSE 34.118249
## 6            VA 2026                      TRUE             FALSE 38.148505
## 7            MS 3600                      TRUE             FALSE 30.406417
## 8            OK  925                     FALSE              TRUE 35.394454
## 9            NV 1828                     FALSE              TRUE   39.1779
## 10           NV NULL                      TRUE             FALSE 36.160083
## 11           WI NULL                     FALSE             FALSE 46.842508
## 12           NV 1414                     FALSE              TRUE 36.111801
## 13           WI 1808                     FALSE             FALSE 43.334351
## 14           VA 1537                      TRUE             FALSE  38.91821
## 15           WI 1430                     FALSE             FALSE 42.600452
## 16           FL 1387                     FALSE              TRUE 30.183208
## 17           WY 1791                      TRUE             FALSE 41.530144
```

```
## 18           VA 7200                    FALSE              TRUE  36.85113
## 19           CT NULL                    FALSE              TRUE 41.767635
## 20           WV NULL                     TRUE             FALSE 39.476717
##      longitude unformattedPrice
## 1   -85.531778           330000
## 2   -72.399085            99900
## 3   -84.327653           390000
## 4    -75.50329           254900
## 5  -117.583907           648700
## 6   -78.811744           339900
## 7    -88.96077           149000
## 8   -97.940477            75000
## 9  -119.721725           495000
## 10 -115.910671            10000
## 11  -90.657247            24900
## 12 -115.115736           264900
## 13  -89.027477           239900
## 14  -77.499949           409999
## 15  -89.647354           139900
## 16  -85.714081           129000
## 17 -109.466244           135000
## 18  -76.181971          4500000
## 19  -71.986799            74900
## 20  -78.790253            79900
```

From the 'df_raw', I selected the following features: 'id', 'statusText', 'addressStreet', 'addressCity', 'addressState', 'area', 'hasAdditionalAttributions', 'isFeaturedListing','latitude','longitude', 'unformattedPrice' as the features for the future analysis.

3. Data cleaning

```
# Remove duplicate rows
df_extracted = df_extracted[!duplicated(df_extracted), ]
```

```
# Remove rows having missing values and rows having string 'NULL' as value.
df_extracted = subset(df_extracted,rowSums(is.na(df_extracted)) <= 0)

# identify which rows in the dataset contain 'NULL'.
rows_to_remove = which(df_extracted[,-1] == 'NULL', arr.ind=T)[,1]
# subset these rows
df_extracted = df_extracted[-rows_to_remove,]
#Check the structure.
str(df_extracted)
```

```
## 'data.frame':    6840 obs. of  11 variables:
##  $ id                        : int  115423997 14543206 72977167 17599151 107827825 2069972083 5253110
##  $ statusText                : chr  "House for sale" "Townhouse for sale" "House for sale" "House for
##  $ addressStreet             : chr  "1053 Lutheran Church Rd" "1718 Woodcliff Ct NE" "5 W Salisbury D
##  $ addressCity               : chr  "Bardstown" "Atlanta" "Wilmington" "Cucamonga" ...
##  $ addressState              : chr  "KY" "GA" "DE" "CA" ...
##  $ area                      : chr  "2054" "2154" "1025" "1322" ...
```

```
##  $ hasAdditionalAttributions: chr  "FALSE" "TRUE" "FALSE" "FALSE" ...
##  $ isFeaturedListing        : logi  TRUE FALSE TRUE FALSE FALSE FALSE ...
##  $ latitude                 : chr  "37.855387" "33.832699" "39.754839" "34.118249" ...
##  $ longitude                : chr  "-85.531778" "-84.327653" "-75.50329" "-117.583907" ...
##  $ unformattedPrice         : chr  "330000" "390000" "254900" "648700" ...
```

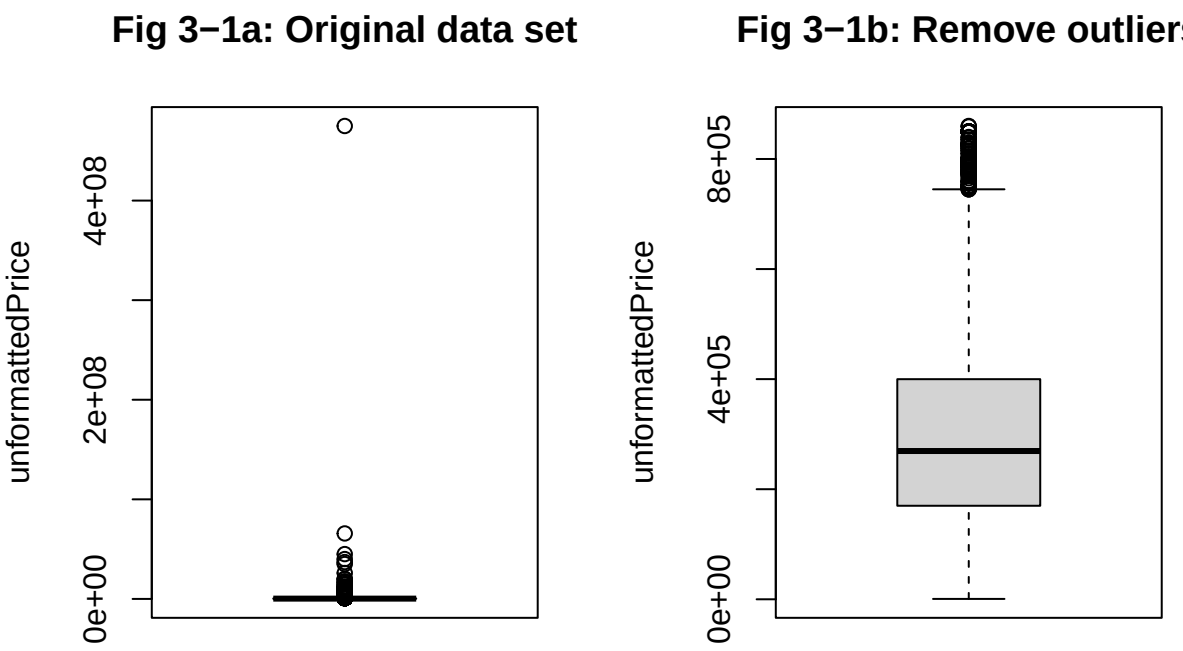
After removing any possible duplicated instances and instances having 'NULL' values, we have 6840 objects left in the dataset 'df_extracted'.

In the following step, I will convert the data into proper type for the analysis.

```
#Convert data into proper type
df_extracted$unformattedPrice <- as.numeric(df_extracted$unformattedPrice)
df_extracted$area <- as.numeric(df_extracted$area)
df_extracted$latitude <- as.numeric(df_extracted$latitude)
df_extracted$longitude <- as.numeric(df_extracted$longitude)
df_extracted$hasAdditionalAttributions <- as.logical(df_extracted$hasAdditionalAttributions)
```

The 'unformattedPrice' in the house data set contains lots of outliers (Fig 3-1a). I removed the outliers and only included the prices that are within 95 % of median price value. The box plot after removing outliers is shown in the Fig 3-1b.

```
#Remove the outliers
quartiles <- quantile(df_extracted$unformattedPrice, probs=c(.25, .75), na.rm = FALSE)
IQR <- IQR(df_extracted$unformattedPrice)

Lower <- quartiles[1] - 1.5*IQR
Upper <- quartiles[2] + 1.5*IQR

df_cleaned <- subset(df_extracted, df_extracted$unformattedPrice > Lower & df_extracted$unformattedPrice

par(mfrow=c(1,2))
boxplot(df_extracted$unformattedPrice,ylab = "unformattedPrice", main = 'Fig 3-1a: Original data set')
boxplot(df_cleaned$unformattedPrice,ylab = "unformattedPrice" ,main = 'Fig 3-1b: Remove outliers')
```



```
#Check the structure of df_cleaned
str(df_cleaned)
```

```
## 'data.frame':    6400 obs. of  11 variables:
##  $ id                         : int  115423997 14543206 72977167 17599151 107827825 2069972083 5253110
##  $ statusText                 : chr  "House for sale" "Townhouse for sale" "House for sale" "House for
##  $ addressStreet              : chr  "1053 Lutheran Church Rd" "1718 Woodcliff Ct NE" "5 W Salisbury D
##  $ addressCity                : chr  "Bardstown" "Atlanta" "Wilmington" "Cucamonga" ...
##  $ addressState               : chr  "KY" "GA" "DE" "CA" ...
##  $ area                       : num  2054 2154 1025 1322 2026 ...
##  $ hasAdditionalAttributions: logi  FALSE TRUE FALSE FALSE TRUE TRUE ...
##  $ isFeaturedListing          : logi  TRUE FALSE TRUE FALSE FALSE FALSE ...
##  $ latitude                   : num  37.9 33.8 39.8 34.1 38.1 ...
##  $ longitude                  : num  -85.5 -84.3 -75.5 -117.6 -78.8 ...
##  $ unformattedPrice           : num  330000 390000 254900 648700 339900 ...
```

After removing outliers, we have 6400 objects in the data set 'df_cleaned'.

4. Analysis

First of all, we will check the statistic summary of continuous features.

```
# load the library
library(psych)
describe(df_cleaned[ , c('area','unformattedPrice')])
```

```
##                   vars    n      mean       sd median   trimmed       mad min
## area                 1 6400   1957.27   7977.9   1568   1644.91    655.31   1
## unformattedPrice     2 6400 302501.78 177403.7 269375 285266.49 163641.97 700
##                      max  range  skew kurtosis      se
## area              435600 435599 49.96  2663.23   99.72
## unformattedPrice 859900 859200  0.83     0.21 2217.55
```

From the summary table above, we can see that the average area of real estate of USA is 1957.27, and the average transaction price is 302501.78.

In the following cells, we will try to count the number of real estate transactions in the USA and visualise the results based on different categorcial features.

```
library(ggplot2)
```

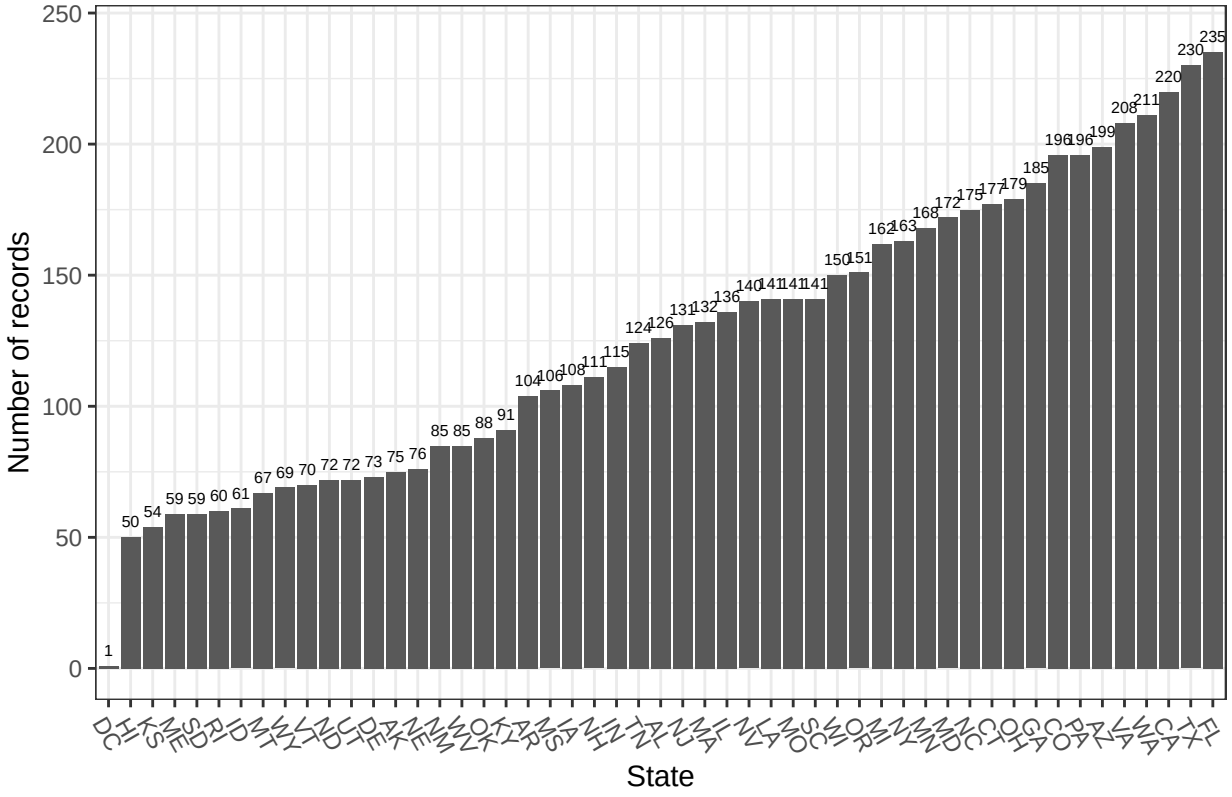
Number of records vs State

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```r
#Count number of records of each addressState and put the data in dataframe.
value_counts <- rle(sort(df_cleaned$addressState))
df_value_counts <- data.frame(index=value_counts$values, number=value_counts$lengths)

#Plot the number of records of real estate transaction for each state.
ggplot(df_value_counts, aes(x = reorder(index, +number), y= number))+geom_bar(stat="identity", position
```

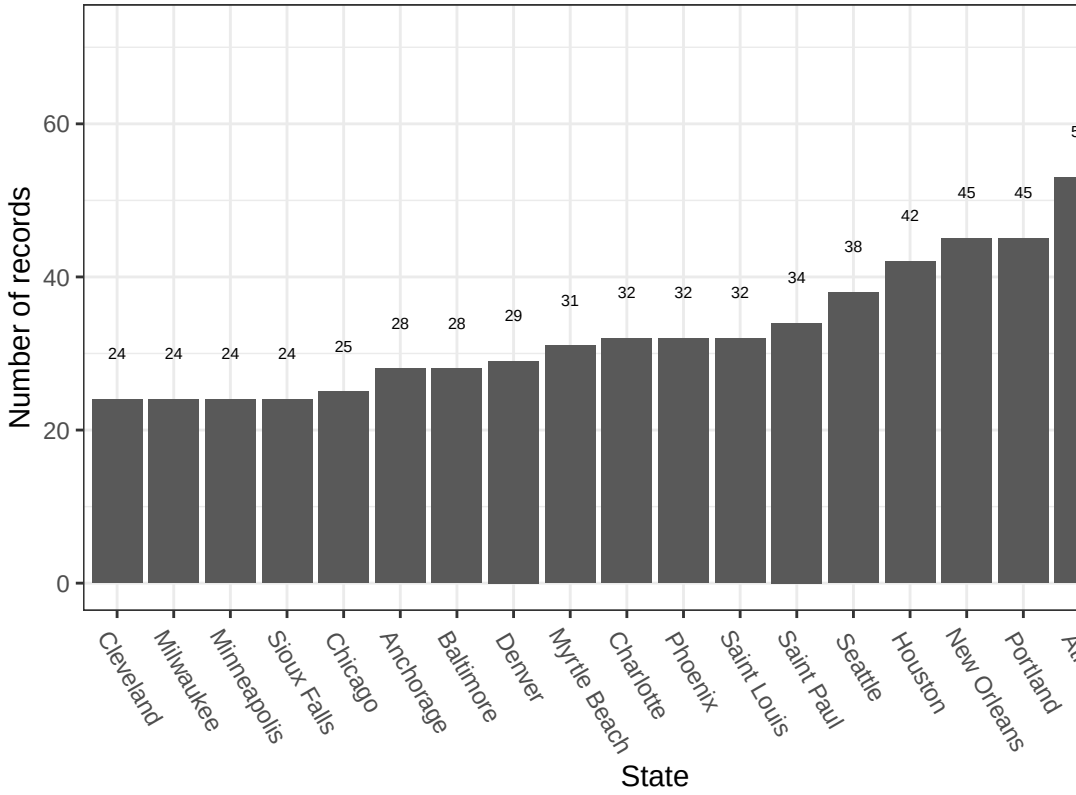
Fig 4−1: Number of records of different states of USA



From Fig 4-1, we can see that the top 5 states with the most real estate transaction records are Florida, Texas, California, Washington and Virginia.

```r
#Count number of records of each addressCity and put the data in dataframe.
value_counts <- rle(sort(df_cleaned$addressCity))
df_city_value_counts <- data.frame(index=value_counts$values, number=value_counts$lengths)
df_city_value_counts <- df_city_value_counts[with(df_city_value_counts,order(-number)),]
# Since there are too many different cities in the data set, we will only choose the top 20 highest val
df_city_value_counts_top20 <- df_city_value_counts[1:20,]

ggplot(df_city_value_counts_top20, aes(x = reorder(index, +number), y= number))+geom_bar(stat="identity
```

Fig 4−2: Top 20 number of records of different cities of USA

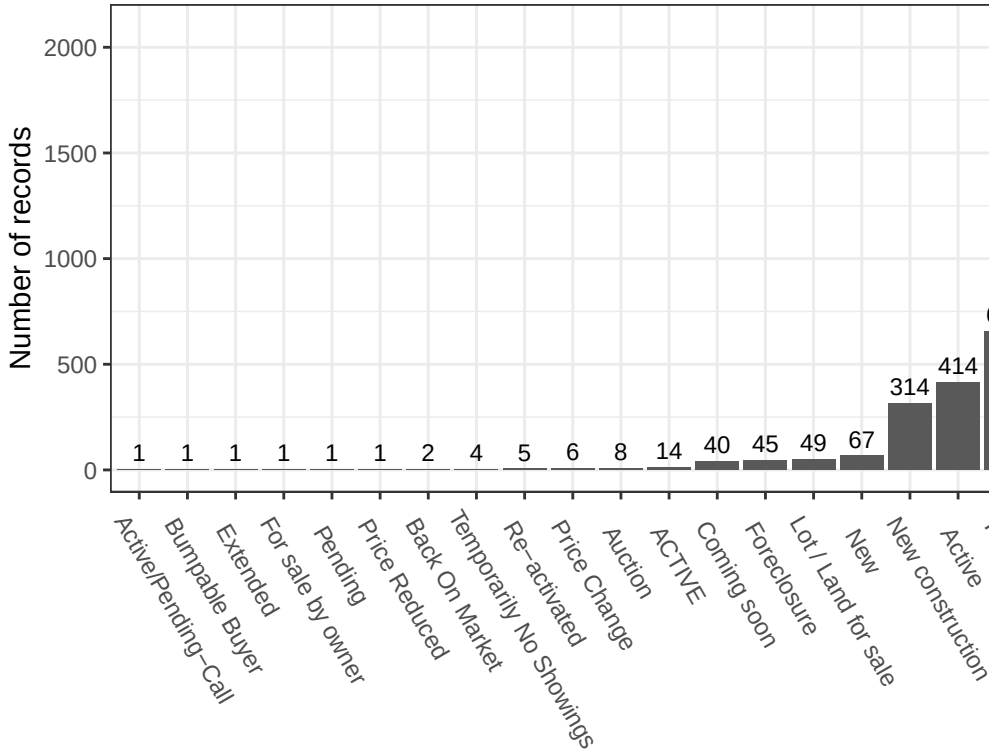


Number of records vs City

From Fig 4-2, we can see that the top 5 cities with the most real estate transaction records are Philadephia, Las Vegas, Atlanta, Seattle and Portland.

```
#Count number of records of each statusText and put the data in dataframe.
value_counts <- rle(sort(df_cleaned$statusText))
df_value_counts <- data.frame(index=value_counts$values, number=value_counts$lengths)

#Plot the number of records of real estate transaction for each statusText
ggplot(df_value_counts, aes(x = reorder(index, +number), y= number))+geom_bar(stat="identity", position
```

Fig 4−3: Number of records based on status text

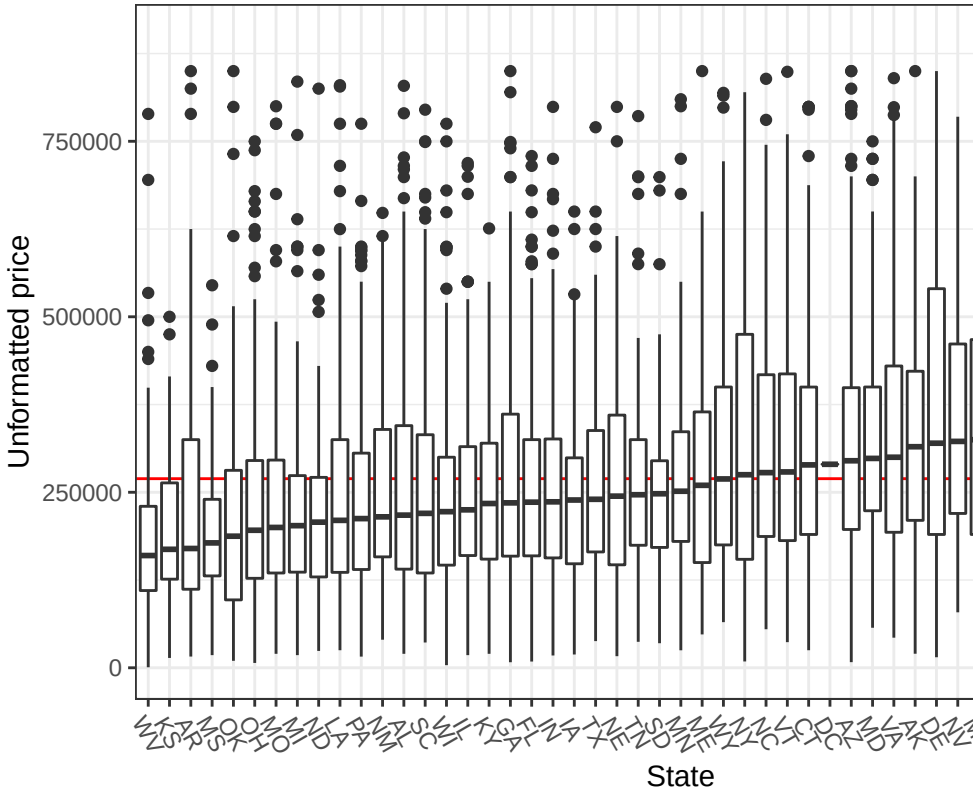


Number of records vs Status text

As we can see above, more than half of the records we have in the data set are classified as either 'House for sale' or 'Condo for sale'. We also have another approximately 1400 of records classified as 'Townhouse for sale', 'Multi-family home for sale' and 'Home for sale' respectively.

```
#Plot the boxplot of addressState vs unformattedPrice
ggplot(df_cleaned, aes(reorder(addressState,unformattedPrice,median), unformattedPrice)) + geom_hline(y
```

Fig 4−4: Boxplot of unformatted price based on state



addressState vs unformattedPrice

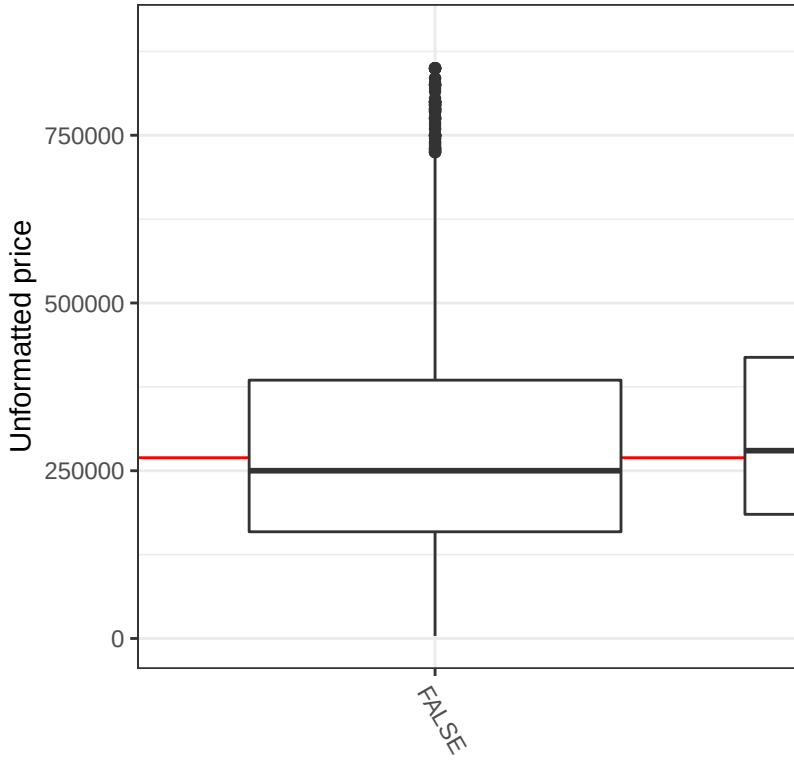
The median sold price in Idaho(ID) is the highest in USA, which is about a factor of 1.5 higher than the national median value of around 260000 (shown in red line). The median sold price in Colorado(CO) is the second highest in the country, which is followed by Massachusetts(MA) ranked 3rd.

There are several states exhibit a very wide distribution among other states, including Hawaii(HI), California(CA), Delaware(DE) and NY(New York). This probably indicates that the historical real estate sold prices fluctuate more in the above-stated states.

Interestingly, the house price in the New York city is thought to be very expensive, but the median sold price of the entire NY state is just slightly above the national median price. And for the other states with relatively high GDP such as Texas, Florida and Pennsylvania, the median sold prices of the real estate are all below the national median price.

```
ggplot(df_cleaned, aes(hasAdditionalAttributions, unformattedPrice)) + geom_hline(yintercept = median(d
```

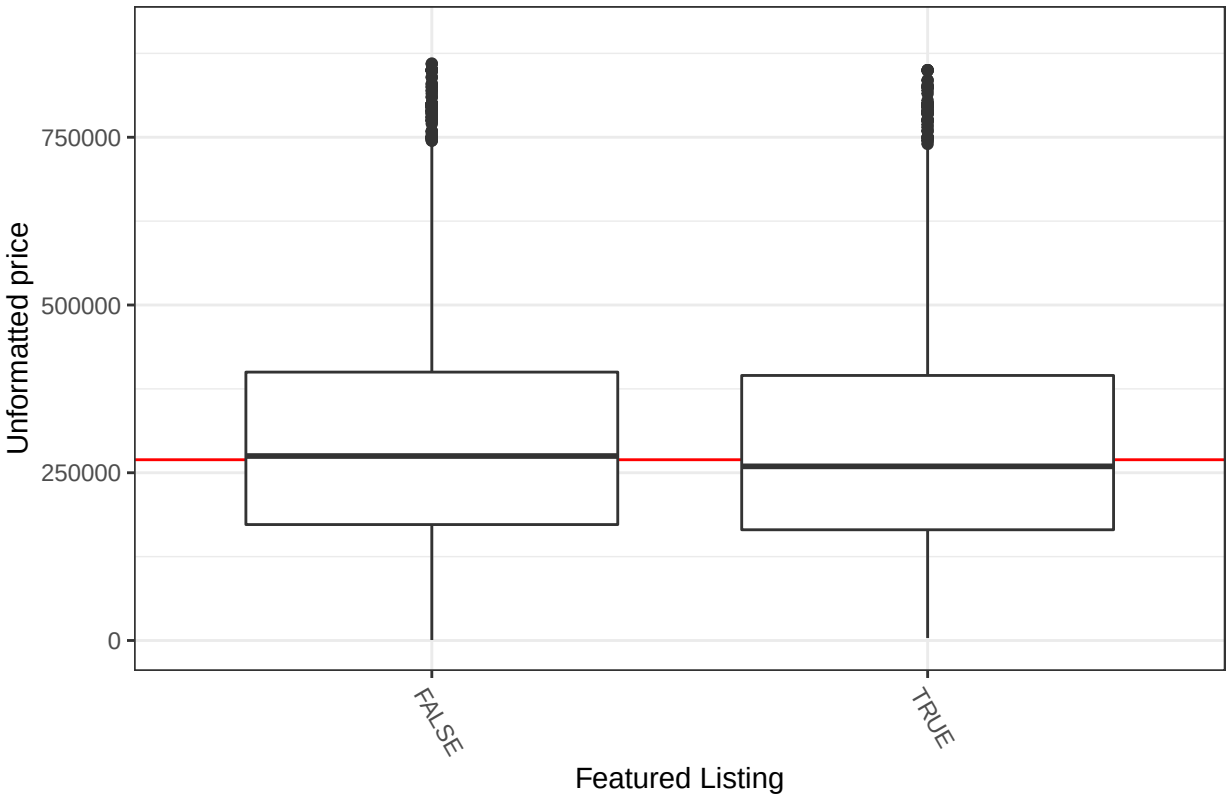
Fig 4−5: Boxplot of unformatted price vs Ad



hasAdditionalAttributions vs unformattedPrice

```
ggplot(df_cleaned, aes(isFeaturedListing, unformattedPrice)) + geom_hline(yintercept = median(df_cleaned
```

Fig 4−5: Boxplot of unformatted price vs Featured Listing

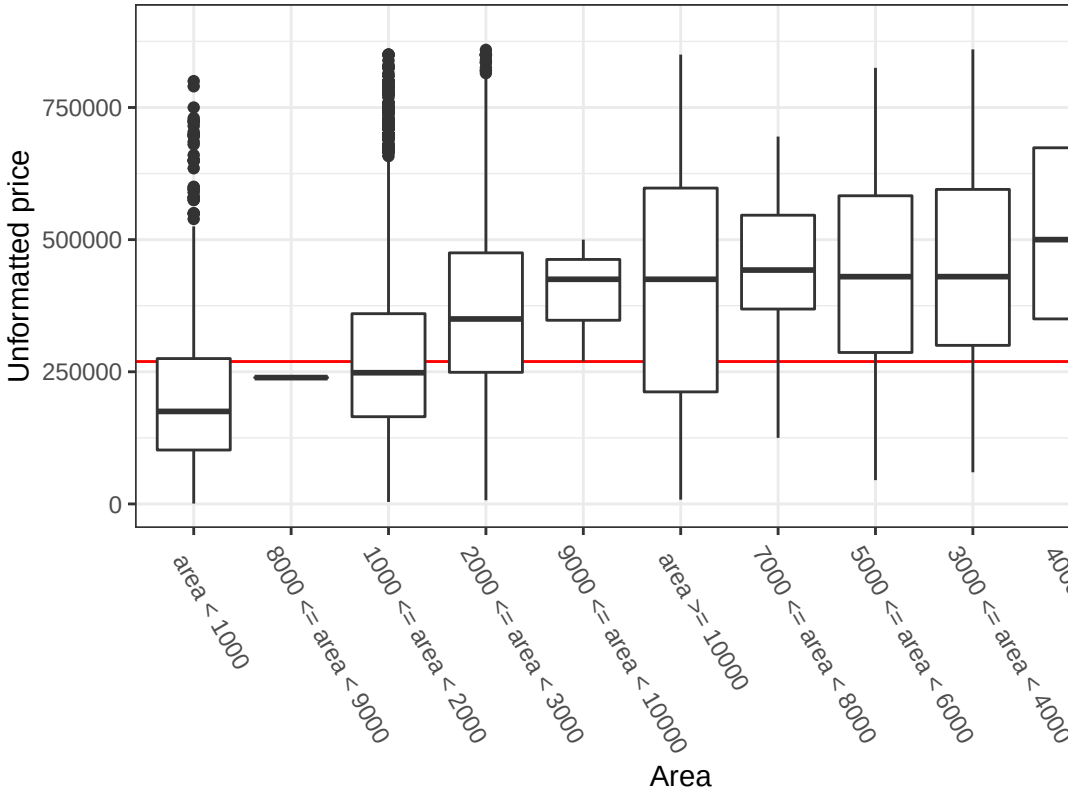


From Fig 4-4 and Fig 4-5, we can conclude that: Residential property with additional attributions has slightly higher median sold price than the property without additional attributions; Residential property that is not featured listing has slightly higher median sold price than the property that is featured listing.

```
#Before we plot the relationship between area and unformatted price, we need to implement the discretiz
df_cleaned <- within(df_cleaned, {
    area.cat <- NA # need to initialize variable
    area.cat[area < 1000] <- "area < 1000"
    area.cat[area >= 1000 & area < 2000] <- "1000 <= area < 2000"
    area.cat[area >= 2000 & area < 3000] <- "2000 <= area < 3000"
    area.cat[area >= 3000 & area < 4000] <- "3000 <= area < 4000"
    area.cat[area >= 4000 & area < 5000] <- "4000 <= area < 5000"
    area.cat[area >= 5000 & area < 6000] <- "5000 <= area < 6000"
    area.cat[area >= 6000 & area < 7000] <- "6000 <= area < 7000"
    area.cat[area >= 7000 & area < 8000] <- "7000 <= area < 8000"
    area.cat[area >= 8000 & area < 9000] <- "8000 <= area < 9000"
    area.cat[area >= 9000 & area < 10000] <- "9000 <= area < 10000"
    area.cat[area >= 10000] <- "area >= 10000"
} )
```

```
ggplot(df_cleaned, aes(reorder(area.cat,unformattedPrice), unformattedPrice)) + geom_hline(yintercept =
```

Fig 4−6: Boxplot of unformatted price vs Area



Area vs Unformatted price

The median sold price of the residential properties with the area between 6000 to 7000 is the highest in USA, which is about a factor of 2.5 higher than the national median price. And the properties with the area less than 1000 has the lowest median sold price.

It is interesting to see that as long as the area of the residential property goes beyond 2000, the median sold price of discrete bins are always greater than the national median price.

Conclusion

Based on the analysis above, we can generally conclude that the location (state/city) have a strong relation to the price of sold residential properties. Additionally, some other features such as the additional attributions, featured listing and the area of properties could also impact the price.

Part 2: R Package

1. Package Introduction

The package I chose to demonstrate in this part is ggmap. The package ggmap is bascially an extension of ggplot2 and allows us to download open sourced map objects, e.g., Google Maps or Open Street Maps. The basic idea driving ggmap is to take a downloaded map image, plot it as a contextual layer using ggplot2, and then plot additional content layers of data, statistics, or models on top of the map.

In ggmap, the process can be divided into two steps: 1. Download the maps and formatting them for plotting. We can use the function `get_map()` 2. Make the plot with `ggmap`, and any additional attributes.

2. ggplot in practice

In the following cells, I will demonstrate the usage of ggmap by plotting the sold price we have in the data set 'df_cleaned' in part 1.

```
citation('ggmap')
```

```
##
## To cite ggmap in publications, please use:
##
##   D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2.
##   The R Journal, 5(1), 144-161. URL
##   http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     author = {David Kahle and Hadley Wickham},
##     title = {ggmap: Spatial Visualization with ggplot2},
##     journal = {The R Journal},
##     year = {2013},
##     volume = {5},
##     number = {1},
##     pages = {144--161},
##     url = {https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf},
##   }
```

```
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.
```

```
#Enable google map service.
#Google Map API key is exclusively for this project, please keep the confidentiality.
#For any Google's Terms of Service: https://cloud.google.com/maps-platform/terms/
register_google(key = "AIzaSyCm-5ioMkkdryWNUolXpUOhyGp8bfWyAsU")
```

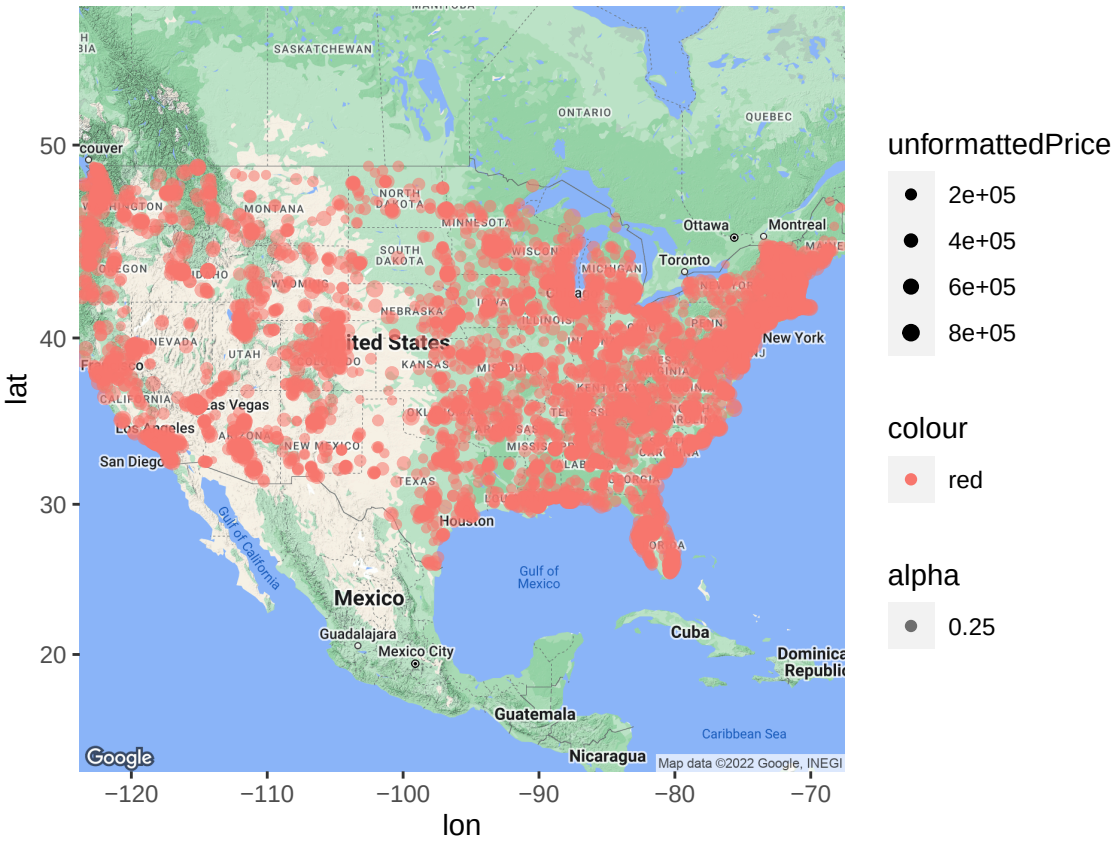
The function `register_google` enables us to access the goole map service through the google map API key. `get_map` enables us to obtain the USA map from google map API service. `geom_point` function enables us to plot the geographical data on the map. We set the paramater `size = unformattedPrice` so that the size of the dot will be customized based on the price. `scale_size` enables us to control the scale of the size of points on the map.

```
#Create a map and plot the data of real estate prices on the map.
map <- get_map(location = "usa", zoom = 4)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=usa&zoom=4&size=640x640&scale=2&mapty
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=usa&key=xxx-5ioMkkdryWNUolXpUOhyG
```

```
ggmap(map) + geom_point(data = df_cleaned, aes(x = longitude, y = latitude, size = unformattedPrice, al
```



From the map above, we can see that the east coast, west coast and the middle east of the USA have relatively more frequent real estate transactions. Meanwhile, sold prices on the East and west coasts are generally higher than those in the central region.

Interaction of ggmap with other packages.

In order to further explore the usage of ggmap, I tried to collaborate it with the function `filter` in package `dplyr` to demonstrate more integrated information.

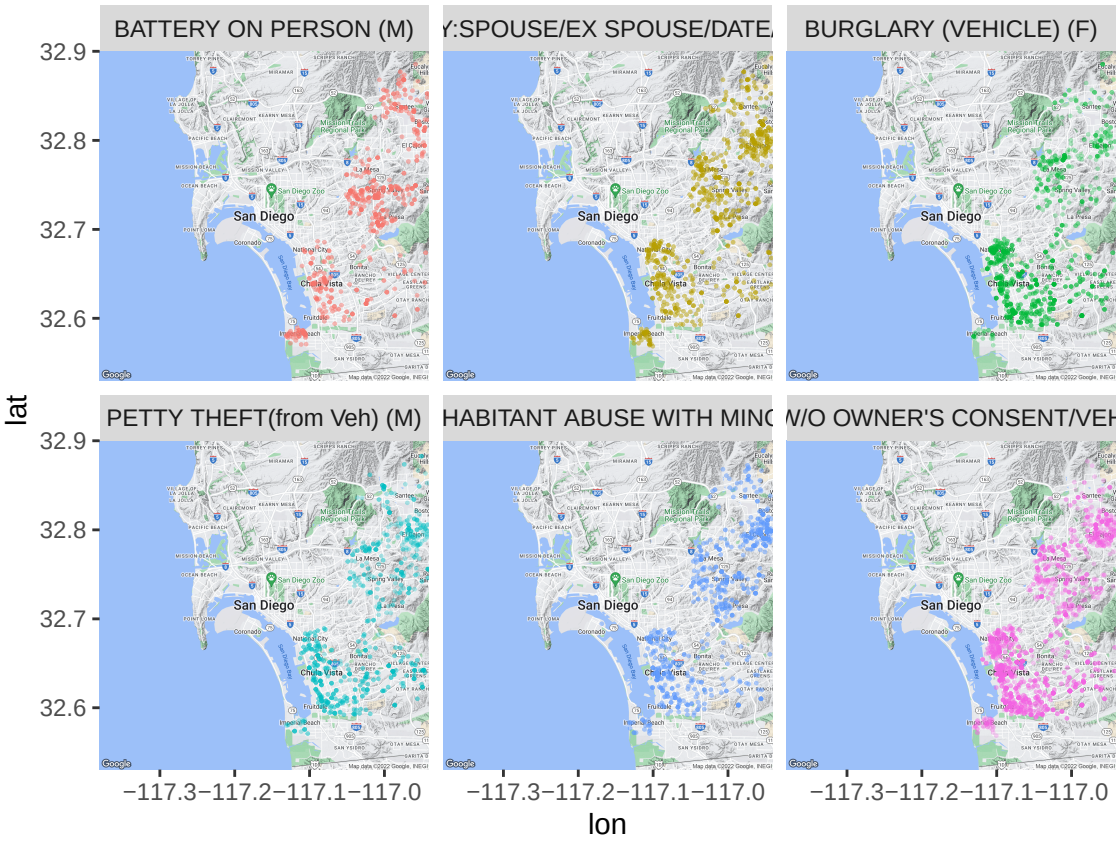
In this part of the project, I introduced another data set "sdcrime_20.csv" which contains the information on all reported crimes in San Diego in 2020. The original data set is availabe at: https://data.sandiegodata.org/dataset/sandiegodata-org-crime-victims/. Please note that the data set "sdcrime_20.csv" has been pre-cleaned by author before loading here.

```
#Load the data set.
sdcrime_20_df = read.csv("sdcrime_20.csv")
#Convert the data into proper data type.
sdcrime_20_df$intptlat <- as.numeric(sdcrime_20_df$intptlat)
sdcrime_20_df$intptlon <- as.numeric(sdcrime_20_df$intptlon)

#Load the map
san.diego.map <- get_map(location = "San Diego", zoom = 11)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=San%20Diego&zoom=11&size=640x640&scal
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=San+Diego&key=xxx-5ioMkkdryWNUolXp
```

```
library(dplyr)
#Filter the data set based on different type of crimes. Here I have filtered out all the type of crimes
sdcrime_20_df_sub <- sdcrime_20_df %>%
  filter(chargedescription %in% c("TAKE VEHICLE W/O OWNER'S CONSENT/VEHICLE THEFT (F)", 'PETTY THEFT(fr

#Plot the data on the map based on the different type of crimes
#Code resourced from: http://lab.rady.ucsd.edu/sawtooth/RAnalytics/maps.html
ggmap(san.diego.map) +
  geom_point(data=sdcrime_20_df_sub,aes(x=intptlon,y=intptlat,color=chargedescription),size=.15,alpha=.
  facet_wrap(~chargedescription) +
  theme(legend.position="none")
```



Here we can see that the data for each type of crime have been shown in the maps. Most of the crimes were recorded happening in the north east and south west of San Diego. Meanwhile, the crime type 'TAKE VEHICLE W/O OWNER'S CONSENT/VEHICLE THEFT' was reported to be the most happened in 2020 in San Diego.

Conclusion

In this part of the report, the author managed to elaborate the usage of ggmap and how can it collaborate with other packages. It is exciting to see the possibility of using ggmap for data analytics. For example, more investigation can be done to analyse the impact of crime on local housing prices in San Diego by plotting crime data and housing price data on the map if we can obtain more historicial housing transaction prices in San Diego.

Part 3: Functions/Programming

In this section of the project, I created functions providing statistic information of the state that user input. There are three functions including `print`, `summary` and `plot`, where `print` gives us the top 10 residential property transactions based on the sold price, `summary` gives us the statistic description of the house price and house area of selected state and `plot` shows us the historic sold prices on the map.

```
#Create the function to subset the data set based on the input from user.
new_state_info = function(stateName){
  df_subset = subset(df_cleaned, df_cleaned$addressState == stateName, class="state_info")
  return(df_subset)
}

#Turn a variable into a class s_data
s_data = new_state_info

#Providing an appropriate printing function
print.state_info = function(s_data){
  df_show = s_data[order(s_data$unformattedPrice, decreasing = TRUE), ]
  df_show = df_show[1:10,]
  return(df_show)
  #return(describe(s_data[ , c('area','unformattedPrice')]))
}

#Input the state and pass the s_data through our printing function
print.state_info(s_data("TX"))
```

```
##             id                  statusText            addressStreet addressCity
## 3952   27770165 Multi-family home for sale          606 Avondale St     Houston
## 7686   26690964 Multi-family home for sale        5109 Vickery Blvd      Dallas
## 5332  250827542                      Active        1108 Highknoll Ln  Georgetown
## 1995   64873459                      Active        4103 E 12th St #1      Austin
## 3842   60136967                      Active    311 W 5th St UNIT 708      Austin
## 5381   97610120           New construction           5013 Rapido Rd     Houston
## 8864   82732721         Townhouse for sale         22 Sweetwater Ct  Sugar Land
## 1400   84020365 Multi-family home for sale   6606 Lockwood Dr APT 8     Houston
## 6616   27502700 Multi-family home for sale       10907 Bob Stone Dr     El Paso
## 1182 2069986676                      Active 5924 S Congress Ave #31S      Austin
##      addressState area hasAdditionalAttributions isFeaturedListing latitude
## 3952           TX 3162                      TRUE             FALSE 29.74513
## 7686           TX 2477                     FALSE             FALSE 32.82280
## 5332           TX 3023                      TRUE             FALSE 30.65087
## 1995           TX 1550                      TRUE             FALSE 30.27857
## 3842           TX 1243                      TRUE             FALSE 30.26783
## 5381           TX 3400                      TRUE             FALSE 29.68137
## 8864           TX 3417                      TRUE             FALSE 29.57933
## 1400           TX 5425                      TRUE             FALSE 29.82012
## 6616           TX 4420                      TRUE             FALSE 31.77344
## 1182           TX 1760                      TRUE             FALSE 30.20012
##       longitude unformattedPrice           area.cat
## 3952  -95.38821           770000 3000 <= area < 4000
## 7686  -96.78428           649900 2000 <= area < 3000
## 5332  -97.68618           625000 3000 <= area < 4000
## 1995  -97.68953           600000 1000 <= area < 2000
```

```
## 3842  -97.74565           559900 1000 <= area < 2000
## 5381  -95.34470           549995 3000 <= area < 4000
## 8864  -95.63169           549000 3000 <= area < 4000
## 1400  -95.31788           540000 5000 <= area < 6000
## 6616 -106.31977           539500 4000 <= area < 5000
## 1182  -97.77832           535000 1000 <= area < 2000
```

The print function with 'TX' as the parameter show us the top 10 highest residential property transaction records in Texas.

```
#Providing an appropriate summary function
summary.state_info = function(s_data){
  return(describe(s_data[ , c('area','unformattedPrice')]))
}

#Input the state and pass the s_data through our summary function
summary.state_info(s_data("TX"))
```

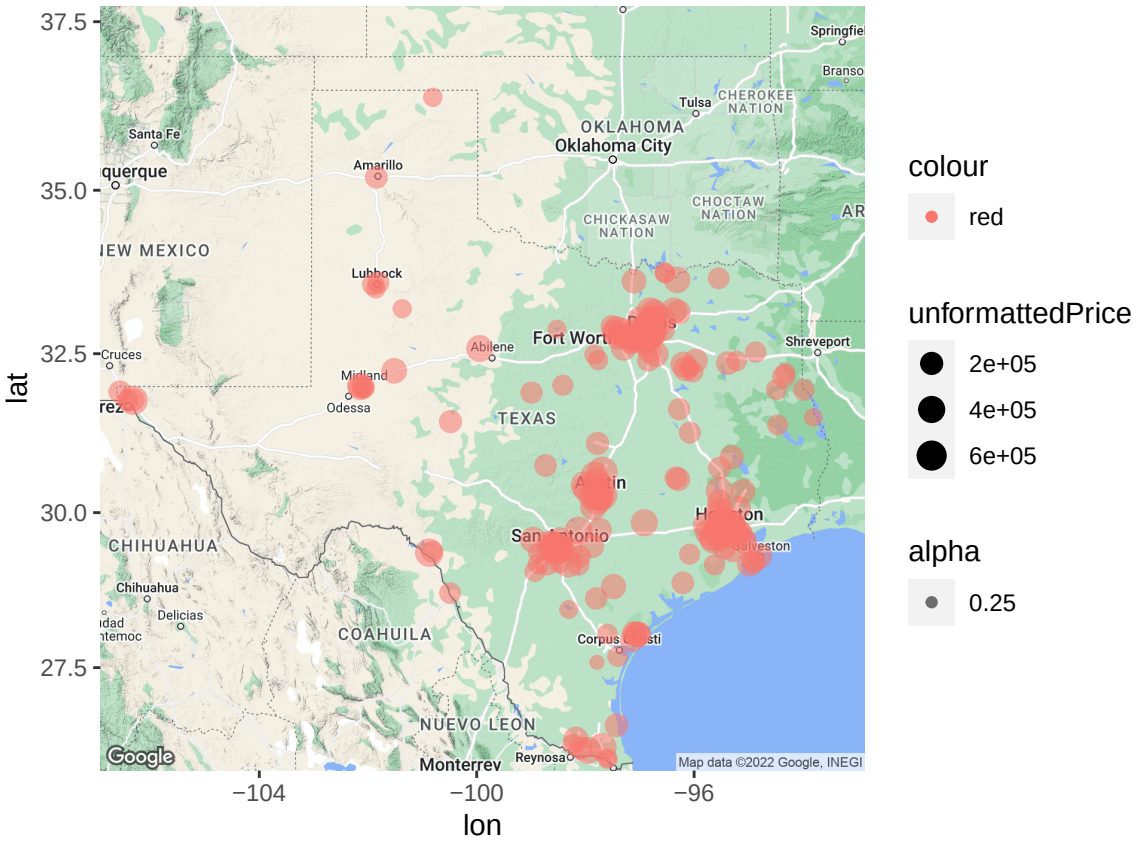
```
##                   vars   n      mean        sd   median   trimmed       mad
## area                 1 230   1889.81   1379.57   1625.5   1675.64    607.12
## unformattedPrice     2 230 262744.17 133197.69 240000.0 251656.39 123352.32
##                     min    max  range skew kurtosis      se
## area                522  13887  13365 5.28    37.33   90.97
## unformattedPrice 38000 770000 732000 0.79     0.43 8782.80
```

The summary function with 'TX' as the parameter give us the statistic description summary of the area and price data of the residential properties sold in Texas. The average sold area of properties is 1889.81 and the average sold price is 262744.17.

```
#Providing an appropriate plot function
plot.state_info = function(stateName, s_data){

  map <- get_map(location = stateName, zoom = 6)

  ggmap(map) + geom_point(data = s_data, aes(x = longitude, y = latitude, size = unformattedPrice, alpha
}

#Input the state and pass the s_data through our plot function
plot.state_info("TX",s_data("TX"))
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=TX&zoom=6&size=640x640&scale=2&maptyp
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=TX&key=xxx-5ioMkkdryWNUolXpUOhyGp
```



From the plot function above, we can see that the transaction records mostly cluster at the central state and coast areas of Texas.

Summary

This is a meaningful project for learning to harness R as a programming language. During the project, I was able to put the theoretic knowledge I learned from R lectures into practice and extend my abilities on R programming as well as on data analysis.

Struggles

Finding a proper data set for this project is not a easy job actually. In order to deliever a quality of work where I can not only demonstrate my understanding of different packages and functions in R, but also keep all the parts of project connected to each other to finally provide a project with strong consistancy, I chose the US historical real estate as my main research data set.

Further Exploration

In part 2, I have plotted the sold prices of residential properties on the US map. It would be interesting to explore some other features that could impact the housing prices and plot them on the map at the same time. For example, the GDP of each state/city, the crime rate or location of transportation hubs, etc.