Data 607 Final Project

Nfn Tenzin Dakar

2024-12-11

Introduction

For this final project, I will be working alone. I am going to analyze a dataset that holds data for the Real-Estate industry as reported by Zestimate. Zestimate was created to give consumers as much information as possible about homes and the housing market, marking the first time consumers had access to this type of home value information at no cost

Data Source

Datasets are available at: https://www.kaggle.com/c/zillow-prize-1/data (https://www.kaggle.com/c/zillow-prize-1/data) File descriptions

properties_2016.csv - all the properties_2017 with their home features for 2016. properties_2017.csv - all the properties_2017 with their home features for 2017 train_2016.csv - the training set with transactions from 1/1/2016 to 12/31/2016 train_2017.csv - the training set with transactions from 1/1/2017 to 9/15/2017 sample_submission.csv - a sample submission file in the correct format zillow_data_dictionary.xlsx - definition of data fields

Project Goal

I seek to establish an understanding of the relationship between home prices and characteristcs of housing as described in Zestimate's datasets. I will draw a relation to factors such as economic forces, home features, and geographical inequality; and their impact on prices of homes.

I will exhaustively apply the various concepts that I have learnt this far, to come up with a good analysis report of this dataset.

Loading Libraries

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(dplyr)
library(shiny)
library(leaflet)
library(ggmap)
```

```
## ℹ Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>
## ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following objects are masked from 'package:psych':
##
##     alpha, rescale
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:ggmap':
##
##     inset
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(bit64)
```

```
## Loading required package: bit
##
## Attaching package: 'bit'
##
## The following object is masked from 'package:data.table':
##
##     setattr
##
## The following object is masked from 'package:psych':
##
##     keysort
##
## The following object is masked from 'package:dplyr':
##
##     symdiff
##
## The following object is masked from 'package:base':
##
##     xor
##
## Attaching package bit64
## package:bit64 (c) 2011-2017 Jens Oehlschlaegel
## creators: integer64 runif64 seq :
## coercion: as.integer64 as.vector as.logical as.integer as.double as.character as.bitstring
## logical operator: ! & | xor != == < <= >= >
## arithmetic operator: + - * / %/% %% ^
## math: sign abs sqrt log log2 log10
## math: floor ceiling trunc round
## querying: is.integer64 is.vector [is.atomic} [length] format print str
## values: is.na is.nan is.finite is.infinite
## aggregation: any all min max range sum prod
## cumulation: diff cummin cummax cumsum cumprod
## access: length<- [ [<- [[ [[<-
## combine: c rep cbind rbind as.data.frame
## WARNING don't use as subscripts
## WARNING semantics differ from integer
## for more help type ?bit64
##
## Attaching package: 'bit64'
##
## The following object is masked from 'package:utils':
##
##     hashtab
##
## The following objects are masked from 'package:base':
##
##     %in%, :, is.double, match, order, rank
```

```
library(lubridate)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(h2o)
```

```
##
## ----------------------------------------------------------------------
##
## Your next step is to start H2O:
##     > h2o.init()
##
## For H2O package documentation, ask for help:
##     > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit https://docs.h2o.ai
##
## ----------------------------------------------------------------------
##
##
## Attaching package: 'h2o'
##
## The following object is masked from 'package:bit64':
##
##     %in%
##
## The following objects are masked from 'package:data.table':
##
##     hour, month, week, year
##
## The following objects are masked from 'package:lubridate':
##
##     day, hour, month, week, year
##
## The following objects are masked from 'package:stats':
##
##     cor, sd, var
##
## The following objects are masked from 'package:base':
##
##     %*%, %in%, &&, ||, apply, as.factor, as.numeric, colnames,
##     colnames<-, ifelse, is.character, is.factor, is.numeric, log,
##     log10, log1p, log2, round, signif, trunc
```

```
library(lime)
```

```
##
## Attaching package: 'lime'
##
## The following object is masked from 'package:dplyr':
##
##     explain
```

```
library(lubridate)
library(magrittr)
library(data.table)
library(bit64)
library(tidyverse)
library(lubridate)
library(mice)
```

```
##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(corrplot)
```

Reading the csv file

```
properties_2017 <-  read.csv(file="C:\\Users\\tenzi\\OneDrive\\Documents\\Data607\\Final Project
\\properties_2017.csv", row.names = NULL)
str(properties_2017)
```

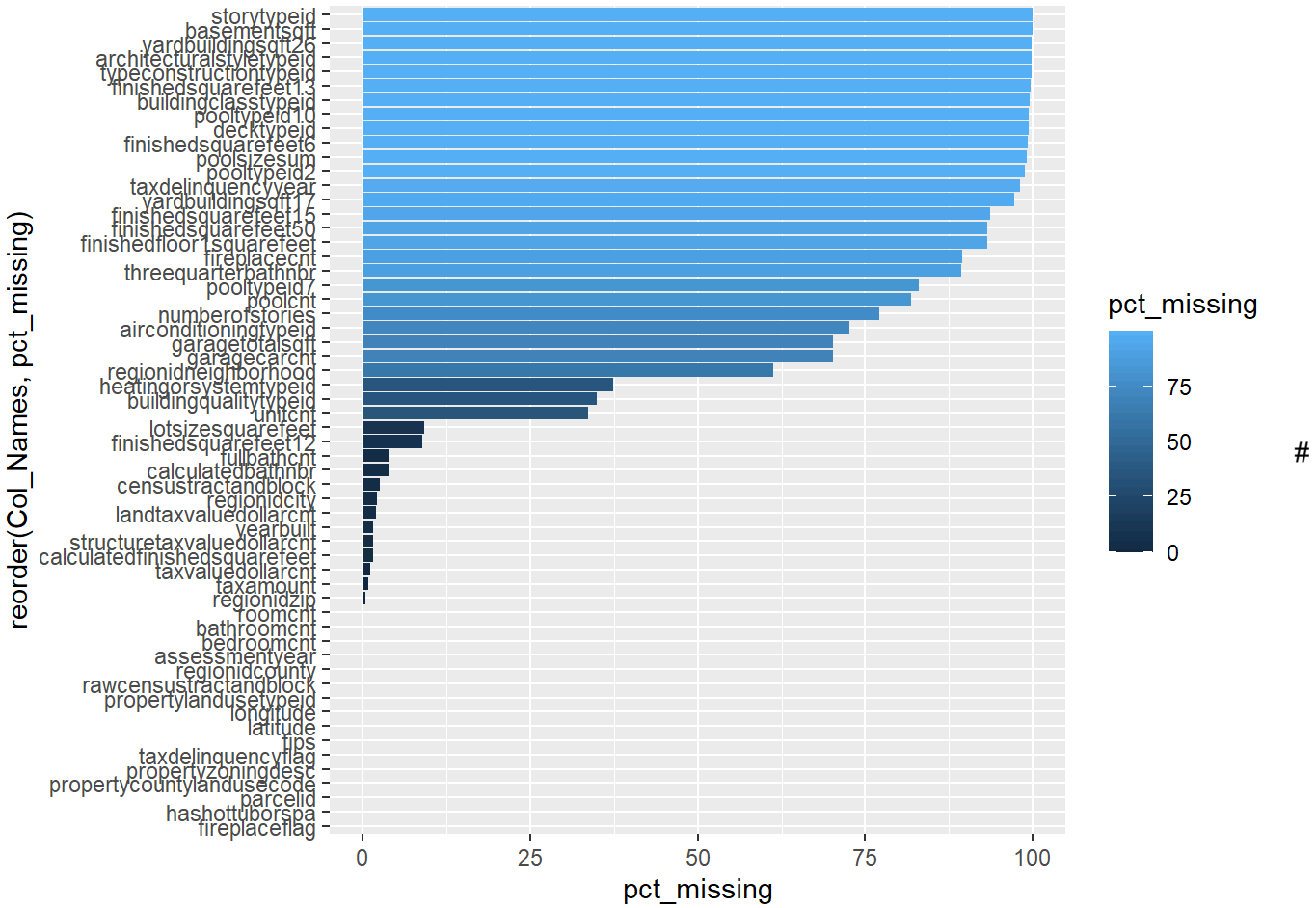
```
## 'data.frame':    2985217 obs. of  58 variables:
##  $ parcelid                    : int  10754147 10759547 10843547 10859147 10879947 10898347 1
0933547 10940747 10954547 10976347 ...
##  $ airconditioningtypeid       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ architecturalstyletypeid    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ basementsqft                : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ bathroomcnt                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedroomcnt                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ buildingclasstypeid         : int  NA NA 5 3 4 4 NA NA NA 3 ...
##  $ buildingqualitytypeid       : int  NA NA NA 6 NA 4 NA NA NA 4 ...
##  $ calculatedbathnbr           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ decktypeid                  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ finishedfloor1squarefeet    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ calculatedfinishedsquarefeet: num  NA NA 73026 5068 1776 ...
##  $ finishedsquarefeet12        : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ finishedsquarefeet13        : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ finishedsquarefeet15        : int  NA NA 73026 5068 1776 2400 NA 3611 NA 3754 ...
##  $ finishedsquarefeet50        : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ finishedsquarefeet6         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ fips                        : int  6037 6037 6037 6037 6037 6037 6037 6037 6037 6037 ...
##  $ fireplacecnt                : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ fullbathcnt                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ garagecarcnt                : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ garagetotalsqft             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ hashottuborspa              : chr  "" "" "" "" ...
##  $ heatingorsystemtypeid       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ latitude                    : int  34144442 34140430 33989359 34148863 34194168 34171873 3
4131929 34171345 34218210 34289776 ...
##  $ longitude                   : int  -118654084 -118625364 -118394633 -118437206 -118385816
-118380906 -118351474 -118314900 -118331311 -118432085 ...
##  $ lotsizesquarefeet           : num  85768 4083 63085 7521 8512 ...
##  $ poolcnt                     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ poolsizesum                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ pooltypeid10                : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ pooltypeid2                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ pooltypeid7                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ propertycountylandusecode   : chr  "010D" "0109" "1200" "1200" ...
##  $ propertylandusetypeid       : int  269 261 47 47 31 31 260 31 269 31 ...
##  $ propertyzoningdesc          : chr  "" "LCA11*" "LAC2" "LAC2" ...
##  $ rawcensustractandblock      : num  60378002 60378001 60377030 60371412 60371232 ...
##  $ regionidcity                : int  37688 37688 51617 12447 12447 12447 12447 396054 396054
47547 ...
##  $ regionidcounty              : int  3101 3101 3101 3101 3101 3101 3101 3101 3101 3101 ...
##  $ regionidneighborhood        : int  NA NA NA 27080 46795 46795 274049 NA NA NA ...
##  $ regionidzip                 : int  96337 96337 96095 96424 96450 96446 96049 96434 96436 9
6366 ...
##  $ roomcnt                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ storytypeid                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ threequarterbathnbr         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ typeconstructiontypeid      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ unitcnt                     : int  NA NA 2 NA 1 NA NA NA NA NA ...
##  $ yardbuildingsqft17          : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ yardbuildingsqft26          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ yearbuilt                   : num  NA NA 1959 1948 1947 ...
##  $ numberofstories             : int  NA NA 1 1 1 1 NA 1 NA 1 ...
##  $ fireplaceflag               : chr  "" "" "" "" ...
##  $ structuretaxvaluedollarcnt  : num  NA NA 660680 580059 196751 ...
##  $ taxvaluedollarcnt           : num  9 27516 1434941 1174475 440101 ...
##  $ assessmentyear              : int  2016 2015 2016 2016 2016 2016 2016 2016 2016 2016 ...
##  $ landtaxvaluedollarcnt       : num  9 27516 774261 594416 243350 ...
##  $ taxamount                   : num  NA NA 20800 14558 5725 ...
##  $ taxdelinquencyflag          : chr  "" "" "" "" ...
##  $ taxdelinquencyyear          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ censustractandblock         : num  NA NA NA NA NA NA NA NA NA NA ...
```

DATA TRANSFORMATION

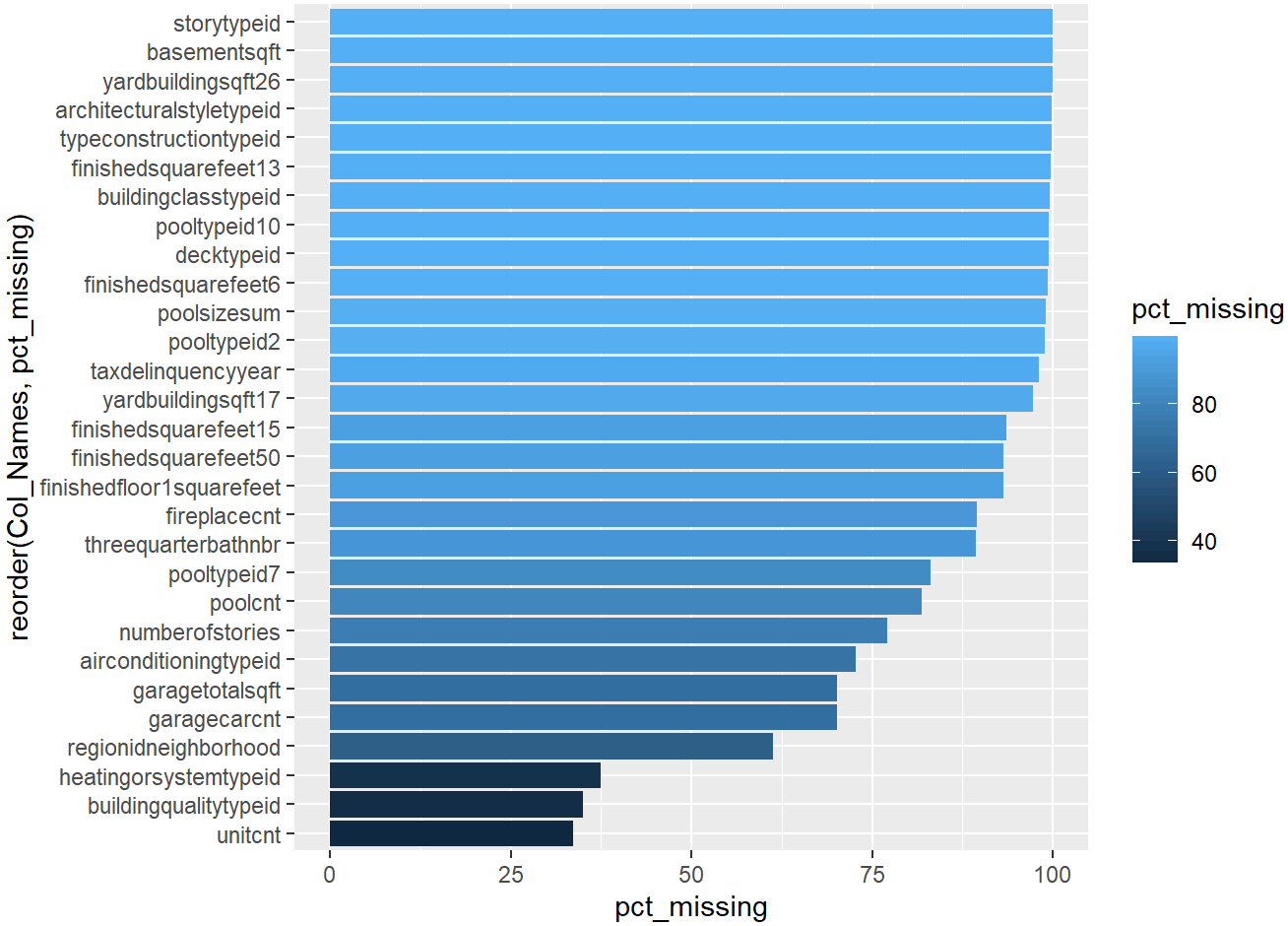
Missing Data

```
prop_miss <- function(x){
  sum(is.na(x)) / length(x)*100
}

missing.bycol <- apply(properties_2017,2,prop_miss)
missing.byrow <-  apply(properties_2017,1,prop_miss)

missdata.df <- as.data.frame(missing.bycol)
setDT(missdata.df, keep.rownames = TRUE)
names(missdata.df) <- c('Col_Names', 'pct_missing')

plt1<-ggplot(data = missdata.df , aes(x= reorder(Col_Names, pct_missing), y=pct_missing)) + geom
_bar(stat = "identity",aes(fill = pct_missing), position = position_stack(reverse= TRUE)) + coor
d_flip()
plt1
```



Columns with more than 20% missing value

```
missdata.df20 <- missdata.df %>% filter (pct_missing>=20)
plt2<-ggplot(data = missdata.df20  , aes(x= reorder(Col_Names, pct_missing), y=pct_missing)) + g
eom_bar(stat = "identity",aes(fill = pct_missing), position = position_stack(reverse= TRUE)) + c
oord_flip()
plt2
```



Using the 80% rule, missing values in more than 20% of samples may be removed from the dataset.

```
missing_prop <- sapply(properties_2017, function(x) sum(is.na(x))/length(x))
variables_to_remove <- names(missing_prop)[missing_prop > 1 - 0.8]
variables_to_remove
```

```
##  [1] "airconditioningtypeid"    "architecturalstyletypeid"
##  [3] "basementsqft"             "buildingclasstypeid"
##  [5] "buildingqualitytypeid"    "decktypeid"
##  [7] "finishedfloor1squarefeet" "finishedsquarefeet13"
##  [9] "finishedsquarefeet15"     "finishedsquarefeet50"
## [11] "finishedsquarefeet6"      "fireplacecnt"
## [13] "garagecarcnt"             "garagetotalsqft"
## [15] "heatingorsystemtypeid"    "poolcnt"
## [17] "poolsizesum"              "pooltypeid10"
## [19] "pooltypeid2"              "pooltypeid7"
## [21] "regionidneighborhood"     "storytypeid"
## [23] "threequarterbathnbr"      "typeconstructiontypeid"
## [25] "unitcnt"                  "yardbuildingsqft17"
## [27] "yardbuildingsqft26"       "numberofstories"
## [29] "taxdelinquencyyear"
```

```
# The dataset afterwards
df_transformed <- properties_2017[, !colnames(properties_2017) %in% variables_to_remove]
dim(df_transformed)
```

```
## [1] 2985217      29
```

```
str(df_transformed)
```

```
## 'data.frame':    2985217 obs. of  29 variables:
##  $ parcelid                    : int  10754147 10759547 10843547 10859147 10879947 10898347 1
0933547 10940747 10954547 10976347 ...
##  $ bathroomcnt                 : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ bedroomcnt                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ calculatedbathnbr           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ calculatedfinishedsquarefeet: num  NA NA 73026 5068 1776 ...
##  $ finishedsquarefeet12        : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ fips                        : int  6037 6037 6037 6037 6037 6037 6037 6037 6037 6037 ...
##  $ fullbathcnt                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ hashottuborspa              : chr  "" "" "" "" ...
##  $ latitude                    : int  34144442 34140430 33989359 34148863 34194168 34171873 3
4131929 34171345 34218210 34289776 ...
##  $ longitude                   : int  -118654084 -118625364 -118394633 -118437206 -118385816
-118380906 -118351474 -118314900 -118331311 -118432085 ...
##  $ lotsizesquarefeet           : num  85768 4083 63085 7521 8512 ...
##  $ propertycountylandusecode   : chr  "010D" "0109" "1200" "1200" ...
##  $ propertylandusetypeid       : int  269 261 47 47 31 31 260 31 269 31 ...
##  $ propertyzoningdesc          : chr  "" "LCA11*" "LAC2" "LAC2" ...
##  $ rawcensustractandblock      : num  60378002 60378001 60377030 60371412 60371232 ...
##  $ regionidcity                : int  37688 37688 51617 12447 12447 12447 12447 396054 396054
47547 ...
##  $ regionidcounty              : int  3101 3101 3101 3101 3101 3101 3101 3101 3101 3101 ...
##  $ regionidzip                 : int  96337 96337 96095 96424 96450 96446 96049 96434 96436 9
6366 ...
##  $ roomcnt                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ yearbuilt                   : num  NA NA 1959 1948 1947 ...
##  $ fireplaceflag               : chr  "" "" "" "" ...
##  $ structuretaxvaluedollarcnt  : num  NA NA 660680 580059 196751 ...
##  $ taxvaluedollarcnt           : num  9 27516 1434941 1174475 440101 ...
##  $ assessmentyear              : int  2016 2015 2016 2016 2016 2016 2016 2016 2016 2016 ...
##  $ landtaxvaluedollarcnt       : num  9 27516 774261 594416 243350 ...
##  $ taxamount                   : num  NA NA 20800 14558 5725 ...
##  $ taxdelinquencyflag          : chr  "" "" "" "" ...
##  $ censustractandblock         : num  NA NA NA NA NA NA NA NA NA NA ...
```

The dataset now has 29 variables, which I will use for the rest of my analysis.

DATA ANALYSIS

Variables

```
#Subset of columns from original data set
col_index<-c(2:9)
working_set<-df_transformed[,col_index]
head(working_set)
```

	bathroomcnt <dbl>	bedroom… <dbl>	calculatedbathnbr <dbl>	calculatedfinishedsquarefeet <dbl>	finisheds
1	0	0	*NA*	*NA*	
2	0	0	*NA*	*NA*	
3	0	0	*NA*	73026	
4	0	0	*NA*	5068	
5	0	0	*NA*	1776	
6	0	0	*NA*	2400	

6 rows | 1-7 of 9 columns

```
# proportion of each level under each variable using the table and summary functions

# Bedroom Count: nummber of bedrooms in a home
table(working_set$bedroomcnt)
```

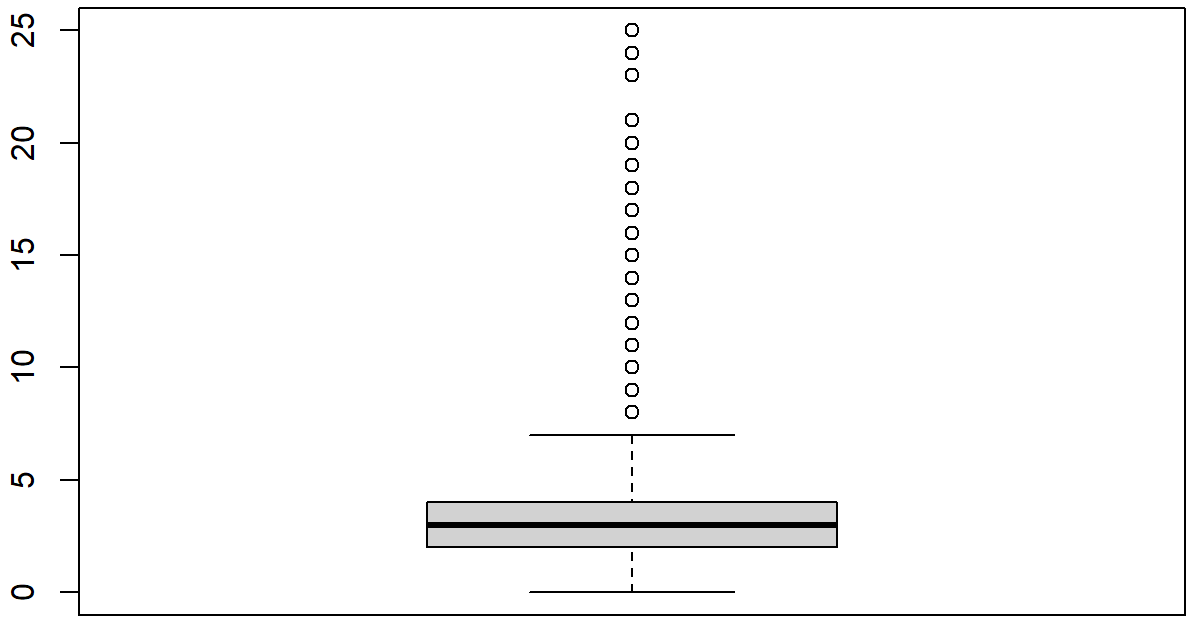
```
##
##      0      1      2      3      4      5      6      7      8      9
## 118705  86941 606782 1172757 731475 182765  48915  12763  13542   4279
##     10     11     12     13     14     15     16     17     18     19
##   1702    425    959     86     69     24     50     11      9      1
##     20     21     23     24     25
##      8      1      1      1      1
```

```
##
```

```
summary(working_set$bedroomcnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   2.000   3.000   3.093   4.000  25.000    2945
```

```
boxplot(working_set$bedroomcnt)
```




```
# Bathroom count: Number of bathrooms in home
bathroomcnt<-table(working_set$bathroomcnt)
bathroomcnt
```

```
##
##       0     0.5       1     1.5    1.75       2     2.5       3     3.5       4
##  113470      16  499332   45735       4 1219811  208809  633089   31835  133922
##     4.5       5     5.5       6     6.5       7     7.5       8     8.5       9
##   19864   38514    6275   16416    1352    6221     385    4548     113    1341
##     9.5      10    10.5      11    11.5      12    12.5      13      14    14.5
##      50     496      14     200       3     269       3      53      39       1
##      15      16      17      18      19    19.5      20      31      32
##      21      25       8      12       3       1       8       1       1
```

```
summary(working_set$bathroomcnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   2.000   2.000   2.216   3.000  32.000    2957
```

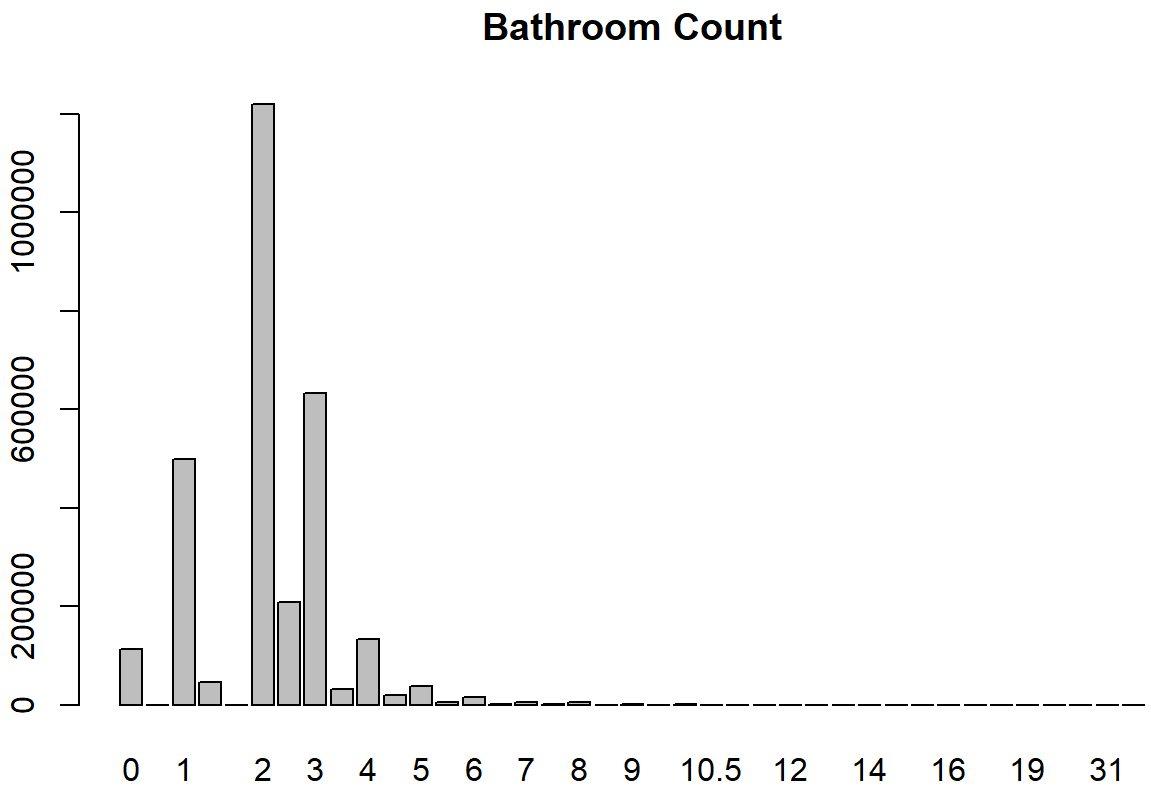
```
# Calculated Bathroom number: Number of bathrooms in home including fractional bathroom
table(working_set$calculatedbathnbr)
```

```
##
##       1     1.5       2     2.5       3     3.5       4     4.5       5     5.5
##  499324   45427 1219799  208578  633088   31773  133922   19811   38514    6259
##       6     6.5       7     7.5       8     8.5       9     9.5      10    10.5
##   16416    1340    6221     382    4548     110    1341      50     496      14
##      11    11.5      12    12.5      13      14    14.5      15      16      17
##     200       3     269       3      53      39       1      21      25       8
##      18      19    19.5      20      31      32
##      12       3       1       8       1       1
```

```
summary(working_set$calculatedbathnbr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     1.0     2.0     2.0     2.3     3.0    32.0  117156
```

```
barplot(bathroomcnt, main = "Bathroom Count")
```



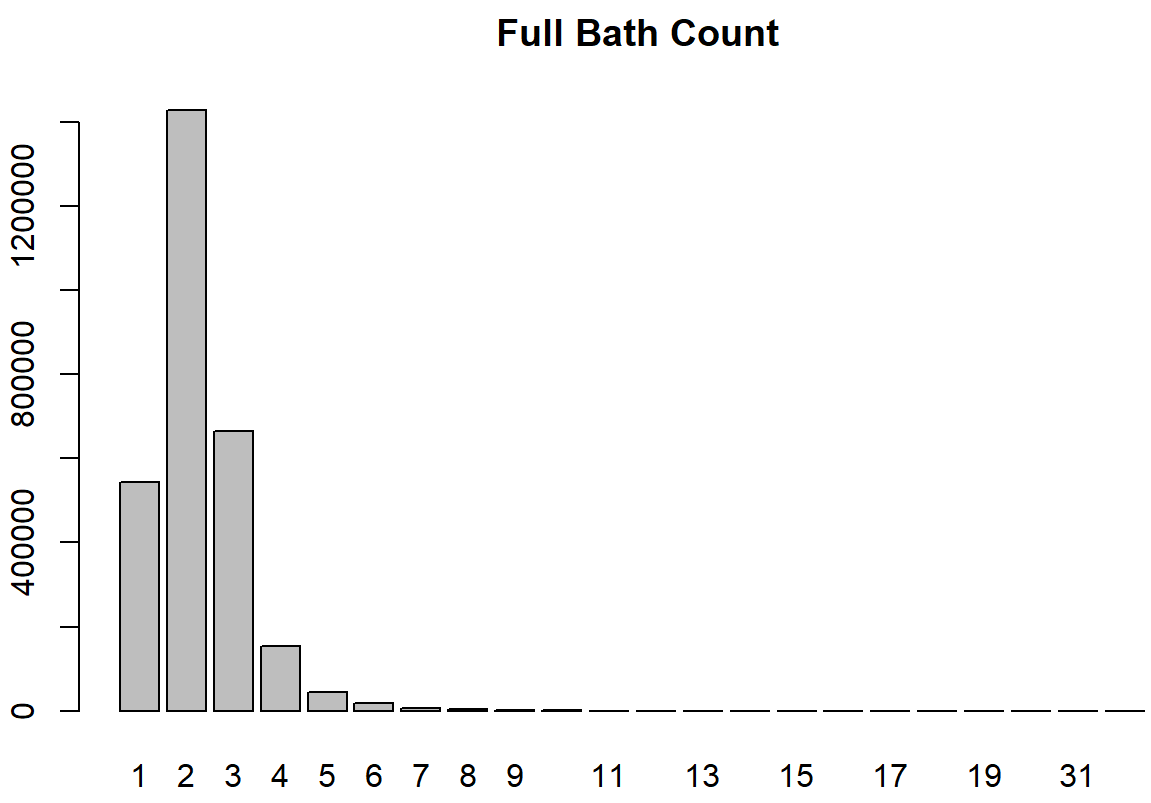

```
# Full bath count: Number of full bathrooms (sink, shower + bathtub, and toilet) present in home

fullbath<-table(working_set$fullbathcnt)
fullbath
```

```
##
##      1      2      3      4      5      6      7      8      9     10
## 544794 1428927 664914 153684  44721  17499   6468   4575   1347   495
##     11     12     13     14     15     16     17     18     19     20
##    197    268     54     39     20     25      8     12      4      8
##     31     32
##      1      1
```

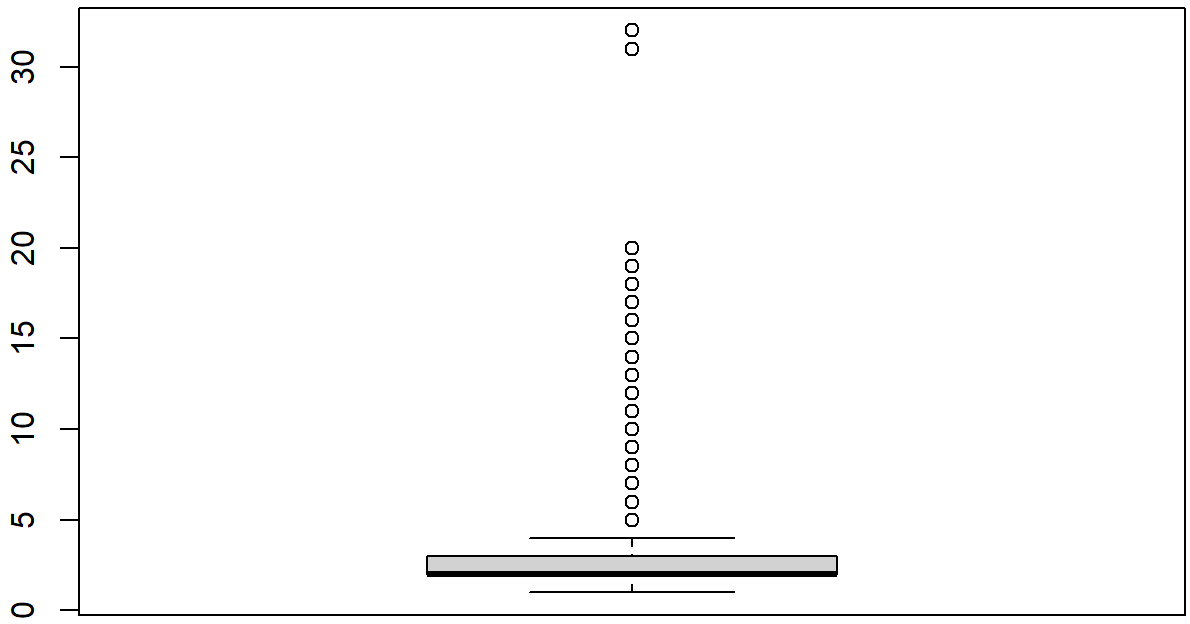
```
summary(working_set$fullbathcnt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00    2.00    2.00    2.25    3.00   32.00  117156
```

```
barplot(fullbath, main = " Full Bath Count")
```




```
boxplot(working_set$fullbathcnt)
```




```
# Hot Tub or Spa: whether or not a home has a hot tub or spa
summary(working_set$hashottuborspa)
```

```
##    Length     Class      Mode
##   2985217 character character
```

```
#Finished Square Feet: Finished total living room area of home
summary(working_set$finishedsquarefeet12)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       1    1198    1542    1764    2075  427079  264431
```

Land properties_2017

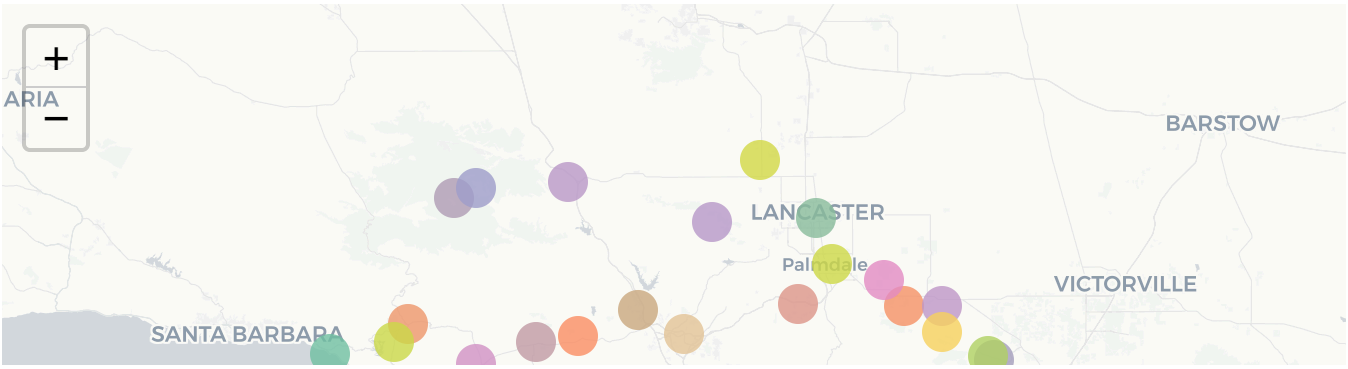
```
# Regionidcity
properties_2017 %>%
  count(regionidcity)
```

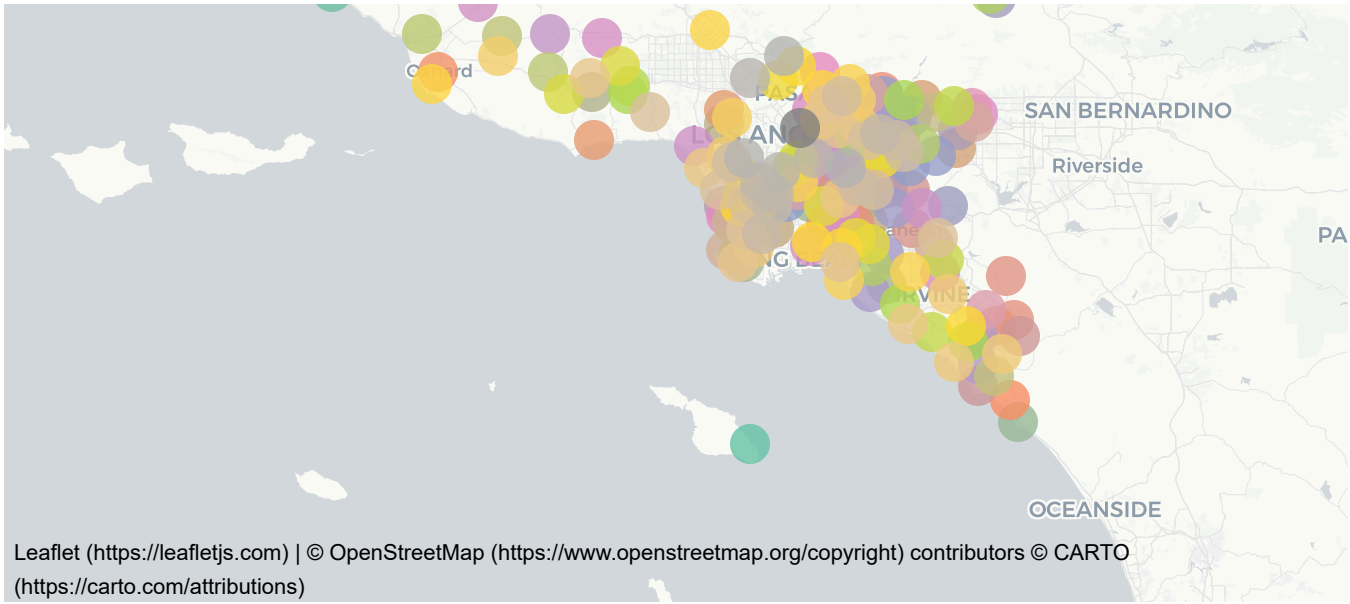
regionidcity	n
<int>	<int>
3491	1076
3980	6
4406	21331
5465	13499
5534	49123
6021	14769
6285	19
6395	7772
6822	781
8384	16361

1-10 of 187 rows Previous **1** 2 3 4 5 6 ... 19 Next

```
# geographic interactive widget to display where regionid is in relation to a regional map

city_info <- colorFactor("Set2", properties_2017$regionidcity)

df_transformed %>%
  group_by(regionidcity = as.factor(regionidcity)) %>%
  summarise(avg_lng = mean(longitude/1e6, na.rm = T),
            avg_lat = mean(latitude/1e6, na.rm =T)) %>%
  leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(lng = ~avg_lng, lat = ~avg_lat, color = ~city_info(regionidcity),
                   stroke = FALSE, fillOpacity = 0.8, popup = ~regionidcity)
```

```
## Warning in RColorBrewer::brewer.pal(max(3, n), palette): n too large, allowed maximum for pal
ette Set2 is 8
## Returning the palette you asked for with that many colors
## Warning in RColorBrewer::brewer.pal(max(3, n), palette): n too large, allowed maximum for pal
ette Set2 is 8
## Returning the palette you asked for with that many colors
```






```
col_index<-c(10,11,16,23:29)
working_dataset2<-df_transformed[,col_index]


# Landtaxvaluedollarcnt: The assessed value of the land area of the parcel
summary(working_dataset2$landtaxvaluedollarcnt)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##        1    79700   176619   268456   326100 94011079    59926
```

```
# Structuretaxvaluedollarcnt: The assessed value of the built structure on the parcel
summary(working_dataset2$structuretaxvaluedollarcnt)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##         1     77666    127066    178143    204000 255321161     46464
```

```
#Taxvaluedollarcnt: The total tax assessed value of the parcel
summary(working_dataset2$taxvaluedollarcnt)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##         1    188220    321161    443528    514072 319622473     34266
```

```
# Taxamount: The total property tax assessed for that assessment year
plt <- qplot(working_dataset2$taxamount, geom="histogram", binwidth = 1000, main = "Distribution
of Tax Amount", ylab =    "Frequency", xlab       = "Tax Amount", fill=I("blue"), col=I("red"),
alpha=I(.2), xlim=c(5000,30000)) +
      theme(axis.text.y = element_text(angle=45))
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
suppressWarnings(print(plt))
```