Live Session 6 Case Study Practice

Matthew Thacker and Damon Resnick

October 19, 2016

This is a live session assignment and practice for the Case Study Assignemnt I.

The goal is to learn more about reproducable research by getting and cleaning data, creating an RStudio project, and doing analysis with documentation that will help the researcher and other researchers in the future.

This project will include a README file, a data directory, an analysis directory, a paper directory, and a link to the an online repository in GitHub.

The data was taken from the Rolling Housing Sales for NYC:

<http://www1.nyc.gov/home/search/index.page?search-terms=Rolling+sales+update>

Our assignment was to look at the data for the borough of Manhattan.

The original code that this project is based off of was presented by the

#### Original Author: Benjamin Reddy

Taken from pages 49-50 of O'Neil and Schutt

require(gdata)

require(plyr) #Added by Monnie McGee

Install the gdata and plyr packages and set working directory in R.

library(plyr)  
library(gdata)  
setwd("C:/Users/hp/Desktop/SMU/Doing Data Science/Live Classes/Unit 7")

.

You need a perl interpreter to load an xls file in Windows. It's automatic in Mac mn <- read.xls("rollingsales\_brooklyn.xls",pattern="BOROUGH")

So we will save the file as a csv and use read.csv instead:

mn <- read.csv("./data/rollingsales\_manhattan.csv",skip=4,header=TRUE)

.

#### Check the data:

head(mn, 3)

## BOROUGH NEIGHBORHOOD  
## 1 1 ALPHABET CITY   
## 2 1 ALPHABET CITY   
## 3 1 ALPHABET CITY   
## BUILDING.CLASS.CATEGORY TAX.CLASS.AT.PRESENT BLOCK  
## 1 01 ONE FAMILY DWELLINGS 1 390  
## 2 03 THREE FAMILY DWELLINGS 1 376  
## 3 07 RENTALS - WALKUP APARTMENTS 2B 373  
## LOT EASE.MENT BUILDING.CLASS.AT.PRESENT  
## 1 61 NA A4  
## 2 24 NA C0  
## 3 16 NA C1  
## ADDRESS APARTMENT.NUMBER ZIP.CODE  
## 1 189 EAST 7TH STREET 10009  
## 2 264 EAST 7TH STREET 10009  
## 3 326 EAST 4TH STREET 10009  
## RESIDENTIAL.UNITS COMMERCIAL.UNITS TOTAL.UNITS LAND.SQUARE.FEET  
## 1 1 0 1 987   
## 2 3 0 3 2,059   
## 3 10 0 10 2,204   
## GROSS.SQUARE.FEET YEAR.BUILT TAX.CLASS.AT.TIME.OF.SALE  
## 1 2,183 1860 1  
## 2 3,696 1900 1  
## 3 8,625 1899 2  
## BUILDING.CLASS.AT.TIME.OF.SALE SALE.PRICE SALE.DATE  
## 1 A4 - 9/23/2015  
## 2 C0 3,775,000 10/22/2015  
## 3 C1 20,000,000 11/17/2015

tail(mn, 3)

## BOROUGH NEIGHBORHOOD  
## 22801 1 WASHINGTON HEIGHTS UPPER   
## 22802 1 WASHINGTON HEIGHTS UPPER   
## 22803 1 WASHINGTON HEIGHTS UPPER   
## BUILDING.CLASS.CATEGORY TAX.CLASS.AT.PRESENT  
## 22801 44 CONDO PARKING 4  
## 22802 44 CONDO PARKING 4  
## 22803 44 CONDO PARKING 4  
## BLOCK LOT EASE.MENT BUILDING.CLASS.AT.PRESENT  
## 22801 2180 1215 NA RG  
## 22802 2180 1216 NA RG  
## 22803 2180 1216 NA RG  
## ADDRESS APARTMENT.NUMBER ZIP.CODE  
## 22801 736 WEST 187 STREET PRK15 10033  
## 22802 736 WEST 187TH STREET PRK16 10033  
## 22803 736 WEST 187 STREET PRK16 10033  
## RESIDENTIAL.UNITS COMMERCIAL.UNITS TOTAL.UNITS LAND.SQUARE.FEET  
## 22801 0 0 1 -   
## 22802 0 0 1 -   
## 22803 0 0 1 -   
## GROSS.SQUARE.FEET YEAR.BUILT TAX.CLASS.AT.TIME.OF.SALE  
## 22801 - 2003 4  
## 22802 - 2003 4  
## 22803 - 2003 4  
## BUILDING.CLASS.AT.TIME.OF.SALE SALE.PRICE SALE.DATE  
## 22801 RG 42,000 10/7/2015  
## 22802 RG - 5/20/2016  
## 22803 RG - 1/29/2016

summary(mn)

## BOROUGH NEIGHBORHOOD   
## Min. :1 MIDTOWN WEST : 4718   
## 1st Qu.:1 UPPER EAST SIDE (59-79) : 2037   
## Median :1 UPPER EAST SIDE (79-96) : 1671   
## Mean :1 UPPER WEST SIDE (59-79) : 1575   
## 3rd Qu.:1 MIDTOWN EAST : 1399   
## Max. :1 UPPER WEST SIDE (79-96) : 1034   
## (Other) :10369   
## BUILDING.CLASS.CATEGORY  
## 13 CONDOS - ELEVATOR APARTMENTS :7807   
## 10 COOPS - ELEVATOR APARTMENTS :6144   
## 45 CONDO HOTELS :2504   
## 26 OTHER HOTELS :1457   
## 17 CONDO COOPS :1006   
## 07 RENTALS - WALKUP APARTMENTS : 783   
## (Other) :3102   
## TAX.CLASS.AT.PRESENT BLOCK LOT EASE.MENT   
## 2 :16117 Min. : 5 Min. : 1.0 Mode:logical   
## 4 : 5124 1st Qu.: 861 1st Qu.: 37.0 NA's:22803   
## 2C : 583 Median :1044 Median :1010.0   
## 1 : 339 Mean :1105 Mean : 787.1   
## 2B : 276 3rd Qu.:1410 3rd Qu.:1302.0   
## 2A : 186 Max. :2250 Max. :9139.0   
## (Other): 178   
## BUILDING.CLASS.AT.PRESENT  
## R4 :7697   
## D4 :5914   
## RH :2504   
## H3 :1440   
## R9 :1006   
## C6 : 762   
## (Other):3480   
## ADDRESS APARTMENT.NUMBER  
## 1335 AVENUE OF THE AMERIC : 1777 :11549   
## 102 WEST 57TH STREET : 1325 HU2 : 1236   
## 1335 AVENUE OF THE AMER : 443 TIMES : 987   
## 200 EAST 94TH STREET : 249 5A : 87   
## 50 RIVERSIDE BOULEVARD : 173 3A : 86   
## 301 WEST 53RD STREET : 145 4A : 78   
## (Other) :18691 (Other) : 8780   
## ZIP.CODE RESIDENTIAL.UNITS COMMERCIAL.UNITS TOTAL.UNITS   
## Min. : 0 Min. : 0.000 Min. : 0.0000 Min. : 0.000   
## 1st Qu.:10016 1st Qu.: 0.000 1st Qu.: 0.0000 1st Qu.: 0.000   
## Median :10019 Median : 0.000 Median : 0.0000 Median : 1.000   
## Mean : 9968 Mean : 2.417 Mean : 0.4015 Mean : 2.953   
## 3rd Qu.:10027 3rd Qu.: 1.000 3rd Qu.: 0.0000 3rd Qu.: 1.000   
## Max. :10463 Max. :8759.000 Max. :313.0000 Max. :8805.000   
##   
## LAND.SQUARE.FEET GROSS.SQUARE.FEET YEAR.BUILT   
## - :19335 - :19425 Min. : 0   
## 7,532 : 1424 112,850 : 1423 1st Qu.:1920   
## 2,500 : 34 3,600 : 13 Median :1960   
## 2,510 : 31 9,695 : 10 Mean :1770   
## 1,665 : 25 4,000 : 9 3rd Qu.:1985   
## 2,008 : 25 7,500 : 8 Max. :2016   
## (Other): 1929 (Other) : 1915   
## TAX.CLASS.AT.TIME.OF.SALE BUILDING.CLASS.AT.TIME.OF.SALE  
## Min. :1.000 R4 :7807   
## 1st Qu.:2.000 D4 :5914   
## Median :2.000 RH :2504   
## Mean :2.438 H3 :1441   
## 3rd Qu.:2.000 R9 :1006   
## Max. :4.000 C6 : 762   
## (Other):3369   
## SALE.PRICE SALE.DATE   
## - : 4519 10/28/2015: 354   
## 45,000 : 182 11/23/2015: 348   
## 10 : 95 12/21/2015: 319   
## 1,100,000 : 90 7/14/2016 : 286   
## 850,000 : 88 1/13/2016 : 267   
## 650,000 : 78 6/30/2016 : 255   
## (Other) :17751 (Other) :20974

str(mn) # This is a very handy function in R! Displays the structure of an object.

## 'data.frame': 22803 obs. of 21 variables:  
## $ BOROUGH : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ NEIGHBORHOOD : Factor w/ 39 levels "ALPHABET CITY ",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ BUILDING.CLASS.CATEGORY : Factor w/ 41 levels "01 ONE FAMILY DWELLINGS ",..: 1 3 5 5 5 5 5 5 5 5 ...  
## $ TAX.CLASS.AT.PRESENT : Factor w/ 9 levels " ","1","1A",..: 2 2 7 7 5 5 5 5 5 5 ...  
## $ BLOCK : int 390 376 373 373 377 377 385 387 387 387 ...  
## $ LOT : int 61 24 16 17 2 2 2 119 119 119 ...  
## $ EASE.MENT : logi NA NA NA NA NA NA ...  
## $ BUILDING.CLASS.AT.PRESENT : Factor w/ 122 levels " ","A1","A4",..: 3 11 12 12 18 18 18 18 18 18 ...  
## $ ADDRESS : Factor w/ 11614 levels "1-2 RIVER TERRACE ",..: 3134 5488 6904 6922 732 732 3894 10437 10437 10433 ...  
## $ APARTMENT.NUMBER : Factor w/ 2686 levels " ",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ ZIP.CODE : int 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 ...  
## $ RESIDENTIAL.UNITS : int 1 3 10 10 22 22 12 22 22 22 ...  
## $ COMMERCIAL.UNITS : int 0 0 0 0 3 3 4 2 2 2 ...  
## $ TOTAL.UNITS : int 1 3 10 10 25 25 16 24 24 24 ...  
## $ LAND.SQUARE.FEET : Factor w/ 1027 levels " - "," 1,003 ",..: 1026 402 437 437 783 783 751 669 669 669 ...  
## $ GROSS.SQUARE.FEET : Factor w/ 1457 levels " - "," 1,000 ",..: 357 674 1332 1332 343 343 65 179 179 179 ...  
## $ YEAR.BUILT : int 1860 1900 1899 1900 1900 1900 1900 1920 1920 1920 ...  
## $ TAX.CLASS.AT.TIME.OF.SALE : int 1 1 2 2 2 2 2 2 2 2 ...  
## $ BUILDING.CLASS.AT.TIME.OF.SALE: Factor w/ 121 levels "A1","A4","A5",..: 2 10 11 11 17 17 17 17 17 17 ...  
## $ SALE.PRICE : Factor w/ 6787 levels " - "," 1 ",..: 1 3095 2464 1 1665 1 1 1215 1 1 ...  
## $ SALE.DATE : Factor w/ 350 levels "1/1/2016","1/10/2016",..: 336 45 70 70 322 322 152 122 122 122 ...

.

Next up is to clean/format the data with regular expressions. More on these later. For now, know that the pattern "[^[:digit:]]" refers to members of the variable name that start with digits. We will use the gsub command to replace them with a blank space.

We then create a new variable that is a "clean" version of sale.price; we make sale.price.n numeric, not a factor.

#### This means that we replaced the -'s with NA and made a new column called SALE.PRICE.N so that R can deal with the data better.

The gsub() function is useful for finding and replacing characters. Counting the data use the count() function helps to tell how many numeric values and NAs there are in SALE.PRICE.N.

mn$SALE.PRICE.N <- as.numeric(gsub("[^[:digit:]]","", mn$SALE.PRICE))  
count(is.na(mn$SALE.PRICE.N))

## x freq  
## 1 FALSE 18284  
## 2 TRUE 4519

.

The names() function makes all variable names lower case.

names(mn) <- tolower(names(mn))

.

#### Get rid of leading digits

The frst 2 lines of the code below will again replace '-' with NA in R so that missing data will not throw off the analysis.

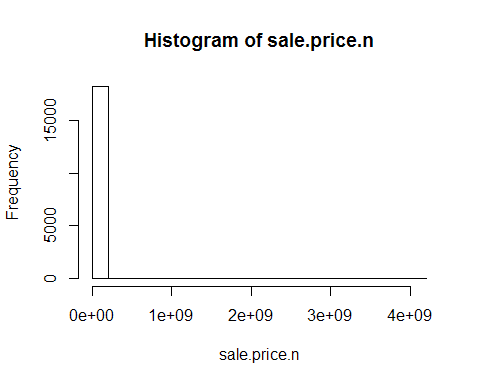
mn$gross.sqft <- as.numeric(gsub("[^[:digit:]]","", mn$gross.square.feet))  
mn$land.sqft <- as.numeric(gsub("[^[:digit:]]","", mn$land.square.feet))  
mn$year.built <- as.numeric(as.character(mn$year.built))

.

#### Here we do a bit of exploration to make sure there's not anything weird going on with sale prices

Attaching 'mn' means that the commands after will automatically reference this data without needing to specify 'mn' for each command.

attach(mn)  
hist(sale.price.n) #Something is wrong! The histogram shows that most sales seem to be close to $0.



detach(mn)

.

#### Keep only the actual sales

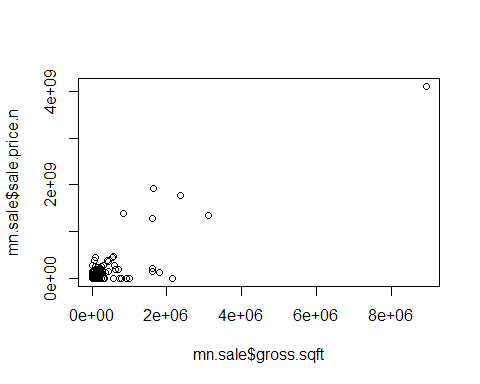
[columns,rows] This will create a data table called mn.sale from mn. This identifies 0's (no given price) within the sale.price column and converts the entire observation to NA. This allows R to ignore those observations when performing analysis on the sale data.

mn.sale <- mn[mn$sale.price.n!=0,]

.

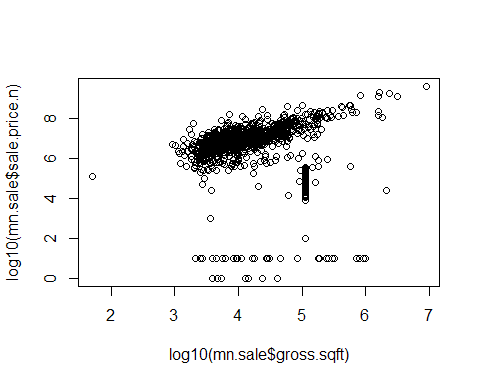
#### Create a scatterplot of the mn.sale data when compared to sq. footage.

plot(mn.sale$gross.sqft,mn.sale$sale.price.n)

 .

#### Perform a log base 10 transformation of the data and create a new scatterplot.

plot(log10(mn.sale$gross.sqft),log10(mn.sale$sale.price.n))

 .

#### Let's look at 1, 2, and 3-family homes

This creates a new data table called mn.homes, which provides a list of family homes. It accomplishes this by searching for the term 'family' within mn.sale building class and passes it to mn.homes.

mn.homes <- mn.sale[which(grepl("FAMILY",mn.sale$building.class.category)),]

.

Look at the dimensions for the mn.homes object (observations and variables)

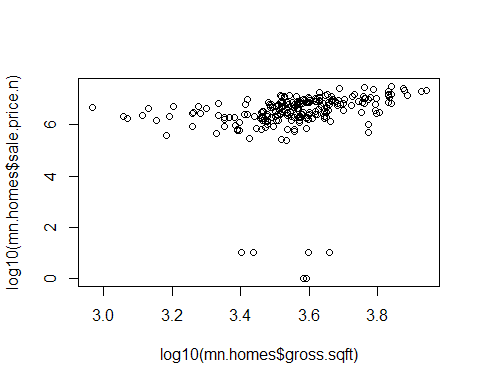
dim(mn.homes)

## [1] 225 24

.

#### Plot the log10 of sale.price vs. gross.sqft and look at the summary of sale.price

plot(log10(mn.homes$gross.sqft),log10(mn.homes$sale.price.n))



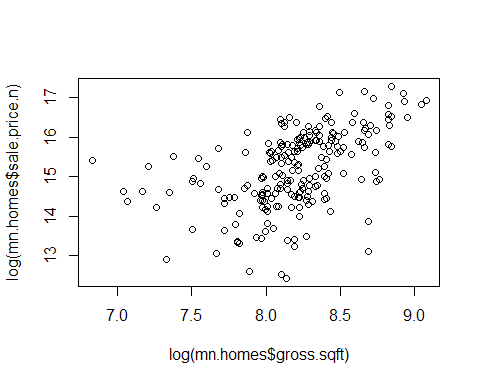
summary(mn.homes[which(mn.homes$sale.price.n<100000),])

## borough neighborhood  
## Min. :1 GREENWICH VILLAGE-WEST :2   
## 1st Qu.:1 GREENWICH VILLAGE-CENTRAL:1   
## Median :1 HARLEM-EAST :1   
## Mean :1 MIDTOWN EAST :1   
## 3rd Qu.:1 UPPER WEST SIDE (59-79) :1   
## Max. :1 ALPHABET CITY :0   
## (Other) :0   
## building.class.category  
## 01 ONE FAMILY DWELLINGS :2   
## 02 TWO FAMILY DWELLINGS :2   
## 03 THREE FAMILY DWELLINGS :2   
## 04 TAX CLASS 1 CONDOS :0   
## 07 RENTALS - WALKUP APARTMENTS :0   
## 08 RENTALS - ELEVATOR APARTMENTS :0   
## (Other) :0   
## tax.class.at.present block lot ease.ment   
## 1 :6 Min. : 542.0 Min. : 17.0 Mode:logical   
## :0 1st Qu.: 603.0 1st Qu.: 36.5 NA's:6   
## 1A :0 Median : 891.5 Median : 43.5   
## 1C :0 Mean : 988.0 Mean : 51.0   
## 2 :0 3rd Qu.:1279.0 3rd Qu.: 47.5   
## 2A :0 Max. :1688.0 Max. :119.0   
## (Other):0   
## building.class.at.present address   
## C0 :2 109 MAC DOUGAL STREET :1   
## A4 :1 129 WEST 78TH STREET :1   
## B3 :1 22 CHRISTOPHER STREET :1   
## S1 :1 234 EAST 49TH STREET :1   
## S2 :1 310 WEST 11 STREET :1   
## :0 339 EAST 116 STREET :1   
## (Other):0 (Other) :0   
## apartment.number zip.code residential.units commercial.units  
## :6 Min. :10012 Min. :1.00 Min. :0.0000   
## (-) :0 1st Qu.:10014 1st Qu.:1.25 1st Qu.:0.0000   
## \* :0 Median :10016 Median :2.00 Median :0.0000   
## 1 :0 Mean :10018 Mean :2.00 Mean :0.3333   
## 1-2W :0 3rd Qu.:10022 3rd Qu.:2.75 3rd Qu.:0.7500   
## 1-A :0 Max. :10029 Max. :3.00 Max. :1.0000   
## (Other) :0   
## total.units land.square.feet gross.square.feet year.built   
## Min. :1.000 1,235 :1 2,520 :1 Min. :1899   
## 1st Qu.:2.000 1,580 :1 2,739 :1 1st Qu.:1899   
## Median :2.500 1,635 :1 3,840 :1 Median :1900   
## Mean :2.333 1,817 :1 3,902 :1 Mean :1902   
## 3rd Qu.:3.000 1,908 :1 3,970 :1 3rd Qu.:1901   
## Max. :3.000 3,140 :1 4,560 :1 Max. :1910   
## (Other):0 (Other):0   
## tax.class.at.time.of.sale building.class.at.time.of.sale sale.price  
## Min. :1 C0 :2 10 :4   
## 1st Qu.:1 A4 :1 1 :2   
## Median :1 B3 :1 - :0   
## Mean :1 S1 :1 1,000 :0   
## 3rd Qu.:1 S2 :1 1,000,000 :0   
## Max. :1 A1 :0 1,000,391 :0   
## (Other):0 (Other) :0   
## sale.date sale.price.n gross.sqft land.sqft   
## 6/28/2016 :2 Min. : 1.00 Min. :2520 Min. :1235   
## 12/14/2015:1 1st Qu.: 3.25 1st Qu.:3014 1st Qu.:1594   
## 12/9/2015 :1 Median :10.00 Median :3871 Median :1726   
## 8/9/2016 :1 Mean : 7.00 Mean :3588 Mean :1886   
## 9/9/2015 :1 3rd Qu.:10.00 3rd Qu.:3953 3rd Qu.:1885   
## 1/1/2016 :0 Max. :10.00 Max. :4560 Max. :3140   
## (Other) :0

.

#### Remove outliers that seem like they weren't actual sales and show the plot

mn.homes$outliers <- (log10(mn.homes$sale.price.n) <=5) + 0  
mn.homes <- mn.homes[which(mn.homes$outliers==0),]  
plot(log(mn.homes$gross.sqft),log(mn.homes$sale.price.n))

 .

#### Create a .csv file called "cleaned\_rollingsales\_manhattan.csv" to save the cleaned data

write.csv(mn.homes, "cleaned\_rollingsales\_manhattan.csv", row.names = TRUE)