Homework 2 Solutions

October 11, 2018

Part 1

i. Since *NYChousing* is a *.csv* file I use *read.csv()* to import the data into R.

```
setwd("~/Desktop/Data")
housing <- read.csv("NYChousing.csv", as.is = TRUE)
```

ii. The function *dim()* provides the dimension of its input object.

```
orig_dim <- dim(housing)
orig_dim
```

```
## [1] 2506   22
```

iii.

```
apply(is.na(housing), 2, sum)
```

```
##                         UID                    PropertyName
##                           0                               0
##                         Lon                             Lat
##                          15                              15
##                    AgencyID                            Name
##                           0                               0
##                       Value                         Address
##                          52                               0
##              Violations2010                      REACNumber
##                           0                            1873
##                     Borough                              CD
##                           0                               0
##         CityCouncilDistrict                     CensusTract
##                          10                               0
##               BuildingCount                       UnitCount
##                           0                               0
##                   YearBuilt                           Owner
##                           0                               0
##                 Rental.Coop               OwnerProfitStatus
##                           0                               0
##     AffordabilityRestrictions StartAffordabilityRestrictions
##                           0                               5
```

The command *is.na(housing)* creates a matrix of the same dimensions as *housing* with each element being TRUE or FALSE depending on whether or not the corresponding element in *housing* is an NA value. Then the full call *apply(is.na(housing), 2, sum)* counts the number of NA values each column of *housing*.

iv.

```
housing <- housing[!is.na(housing$Value), ]
```

The call **is.na(housing$Value)** returns a logical vector with TRUE where **housing$Value** is **NA**, therefore I filter using **!is.na(housing$Value)** to get only the rows where **Value** is not **NA**. I reassign my **housing** dataframe, to be the filtered dataframe.

v.

```
new_dim <- dim(housing)
orig_dim[1] - new_dim[1]
```

```
## [1] 52
```

I removed 52 rows of my dataframe which is what I expect, since my ouput in (iii) told me that I have 52 missing values in **Value**.

v.

```
housing$logValue <- log(housing$Value)
summary(housing$logValue)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.41   12.49   13.75   13.68   14.80   20.47
```

vi.

```
housing$logUnits <- log(housing$UnitCount)
```

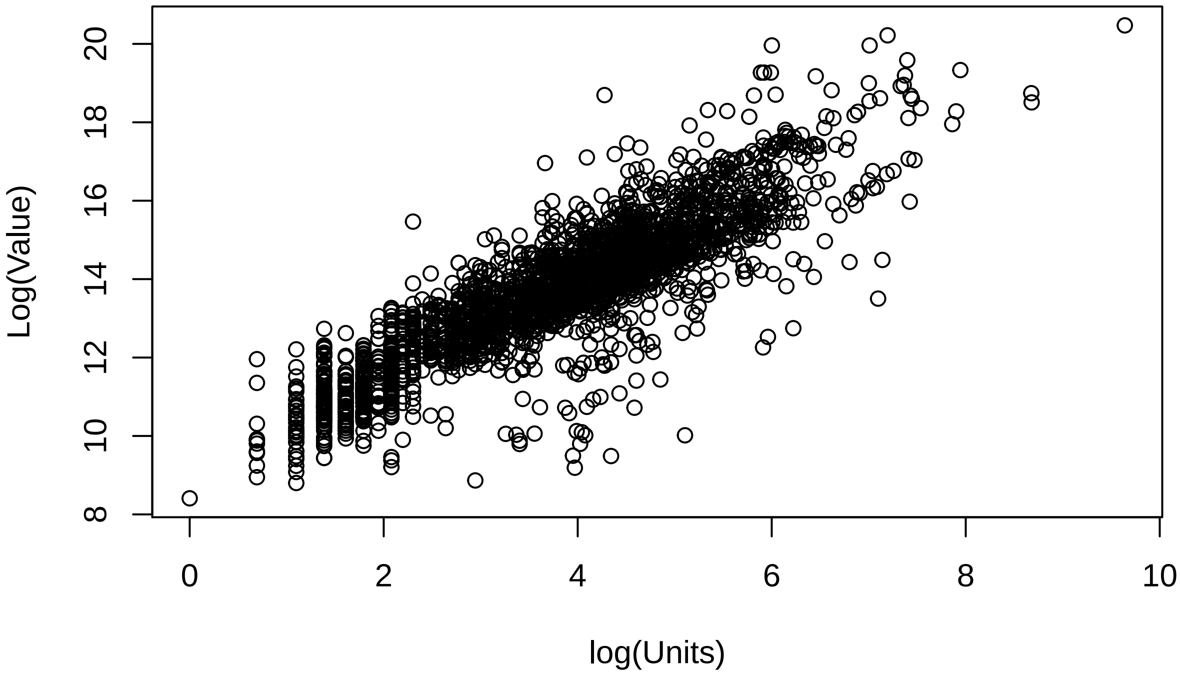
vii.

```
housing$after1950 <- housing$YearBuilt >= 1950
```

Part 2: EDA

i.

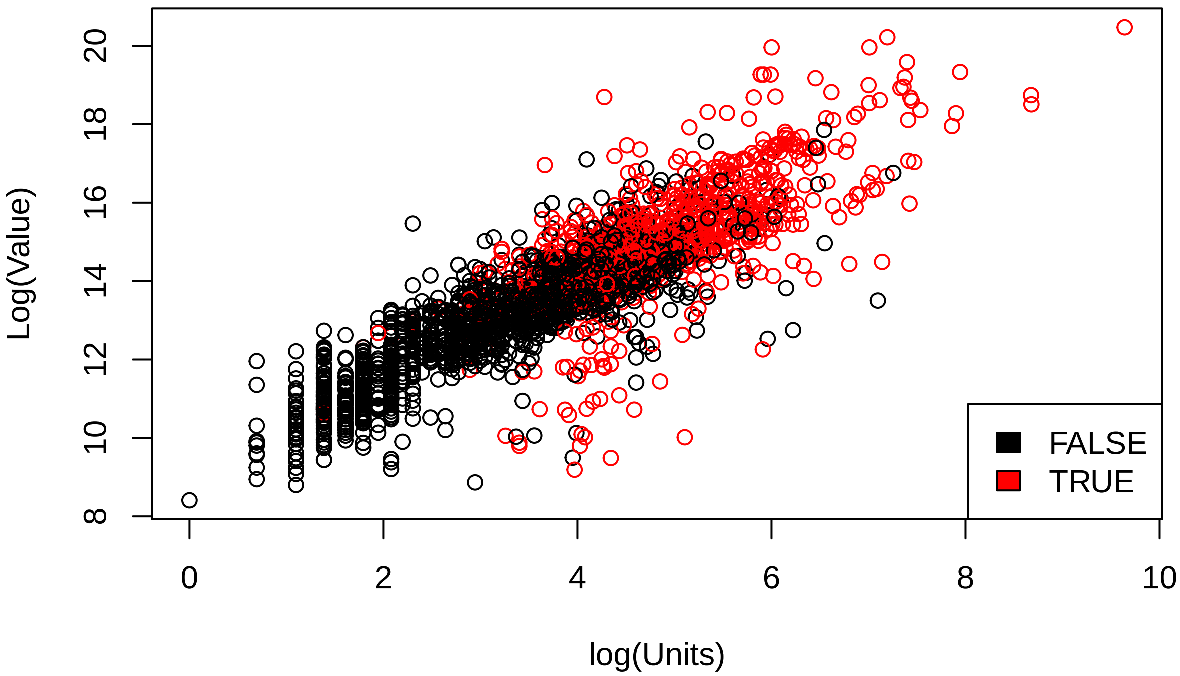
```
plot(housing$logUnits, housing$logValue, xlab = "log(Units)", ylab = "Log(Value)")
```



I plot a scatterplot with the **plot()** command and add argument **xlab =** and **ylab =** for the labels.

ii.

```r
plot(housing$logUnits, housing$logValue, col = factor(housing$after1950), xlab = "log(Units)", ylab = "L
legend("bottomright", legend = levels(factor(housing$after1950)), fill = unique(factor(housing$after1950
```



There appears to be a pretty strong linear reltionship between **logValue** and **logUnits**. When colored according to the **after1950** variable, it is clear that newer buildings (those built after 1950) tend to be more expensive and have more units than older buildings.

iii.

```r
cor(housing$logValue, housing$logUnits)
```

```
## [1] 0.8727348
```

```r
cor(housing$logValue[housing$Borough == "Manhattan"], housing$logUnits[housing$Borough == "Manhattan"])
```

```
## [1] 0.8830348
```

```r
cor(housing$logValue[housing$Borough == "Brooklyn"], housing$logUnits[housing$Borough == "Brooklyn"])
```

```
## [1] 0.9102601
```

```r
cor(housing$logValue[housing$after1950], housing$logUnits[housing$after1950])
```

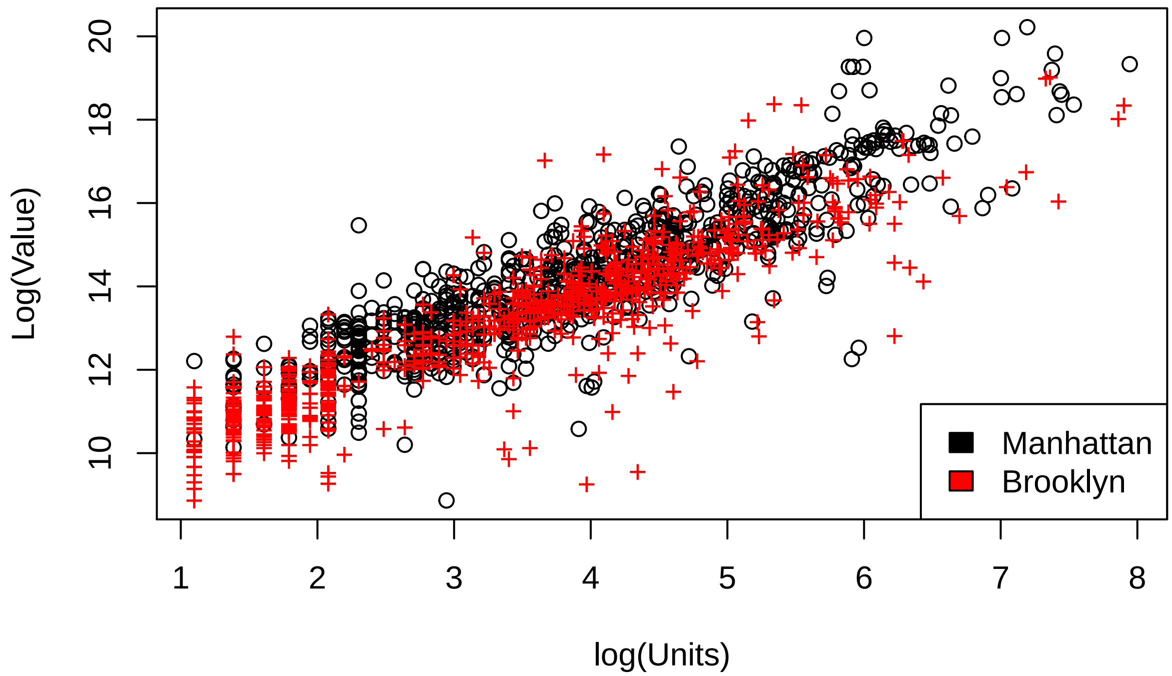
```
## [1] 0.721735
```

```r
cor(housing$logValue[!housing$after1950], housing$logUnits[!housing$after1950])
```

```
## [1] 0.8643297
```

iv.

```r
plot(housing$logUnits[housing$Borough == "Manhattan"], housing$logValue[housing$Borough == "Manhattan"]
points(housing$logUnits[housing$Borough == "Brooklyn"], housing$logValue[housing$Borough == "Brooklyn"]
legend("bottomright", legend = c("Manhattan", "Brooklyn"), fill = c("black", "red"))
```



v.

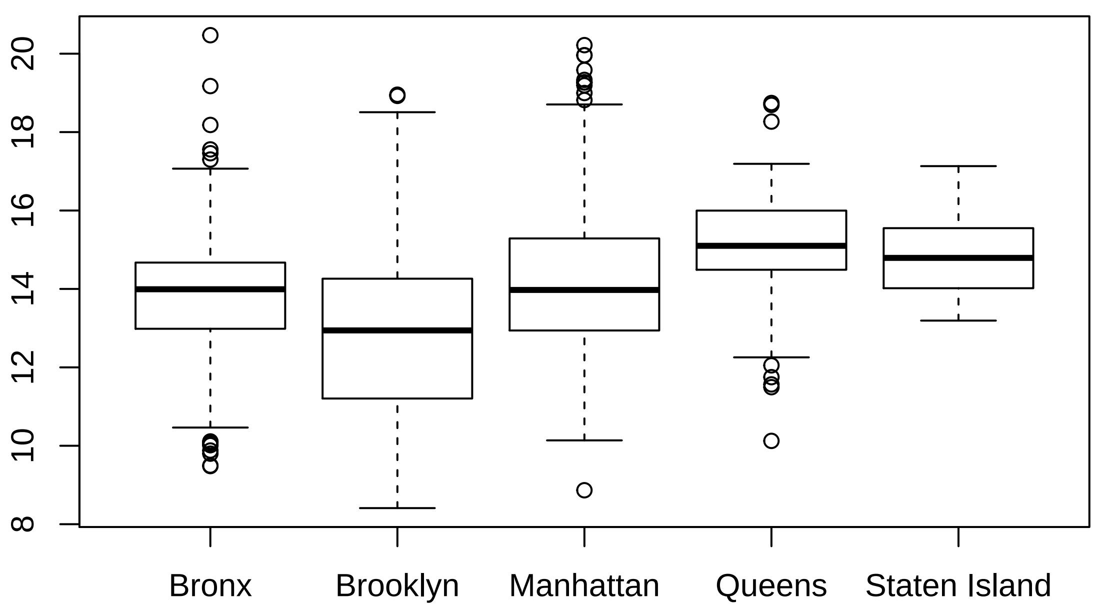
```r
median(housing$Value[housing$Borough == "Manhattan"])
```

```
## [1] 1172362
```

The code calculates the median property value for all properties in Manhattan.

vi.

```r
boxplot(housing$logValue ~ housing$Borough)
```



vii.

```r
tapply(housing$Value, housing$Borough, median)
```

```
##         Bronx      Brooklyn     Manhattan        Queens Staten Island
##       1192950        417610       1172362       3611700       2654100
```

We use **tapply()** which splits the property value into groups based on **Borough** and then calculated the median within each group.