Assignment 2

Giang Vu

2/9/2021

Assignment 2

Part 1

i. Data was imported.

```
#set working dir.
setwd("/Users/giangvu/Desktop/STAT 2301 - Statistical Computing and Data Science/HW/HW2")
#import dataset
housing <- read.csv("NYChousing.csv",header = T,as.is = T,sep = ",")
```

ii. There are 2506 rows and 22 columns in the 'housing' dataframe.

```
nrow(housing)
```

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

```
ncol(housing)
```

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

iii. This command below applies the sum function to the function to check which elements are missing in all columns of the dataframe. In other words, it tells us how many missing entries there are in each column.

```
#check how many NAs there are in each column
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
```

iv. and v. After removing the rows that have NA in variable Value, the dataframe now only has 2454 rows. Compared to the original 2506 rows, 52 rows were omitted, which agrees with the result from part iii.

```
#remove rows with NA in Value
housing <- housing[!is.na(housing$Value),]
#check number of rows
nrow(housing)
```

```
## [1] 2454
```

vi. A new variable logValue is created and its summary statistics are obtained below.

```
#create new variable logValue
housing$logValue <- log(housing$Value)
#get summary stats for new variable
summary(housing$logValue)
```

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

vii. A new variable logUnits is created as follows.

```
#create new variable logUnits
housing$logUnits <- log(housing$UnitCount)
```

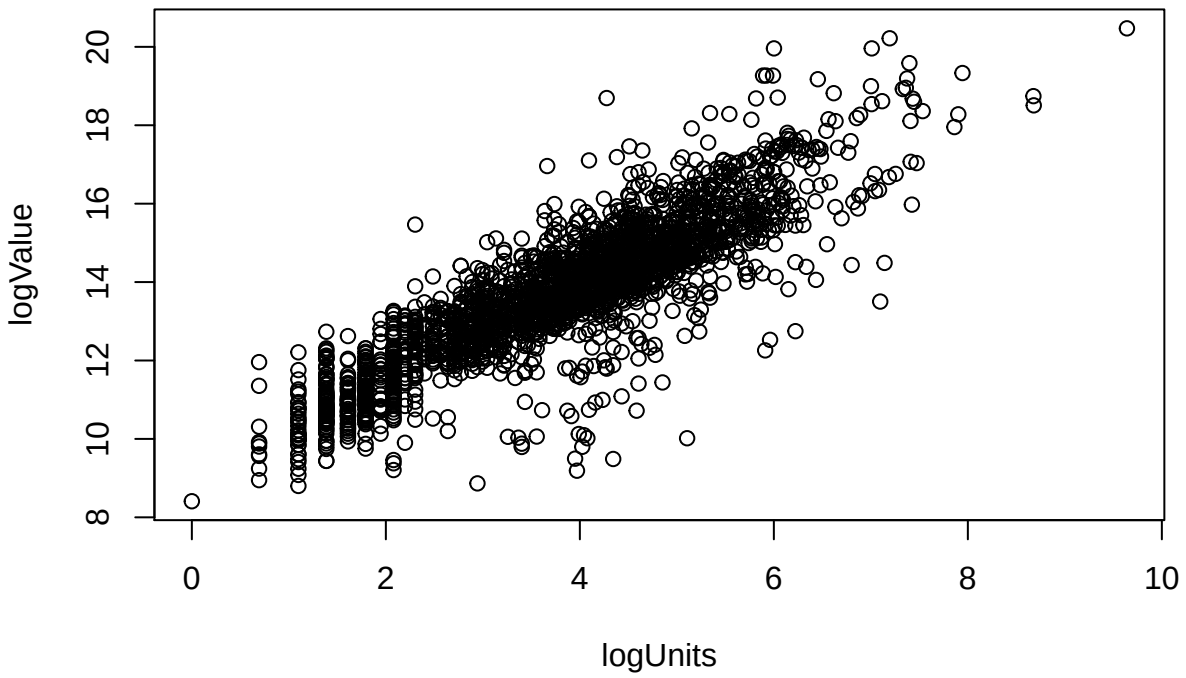
viii. A new variable after1950 is created as follows.

```
#create new variable after1950
housing$after1950 <- ifelse(housing$YearBuilt>=1950,TRUE,FALSE)
```

Part 2

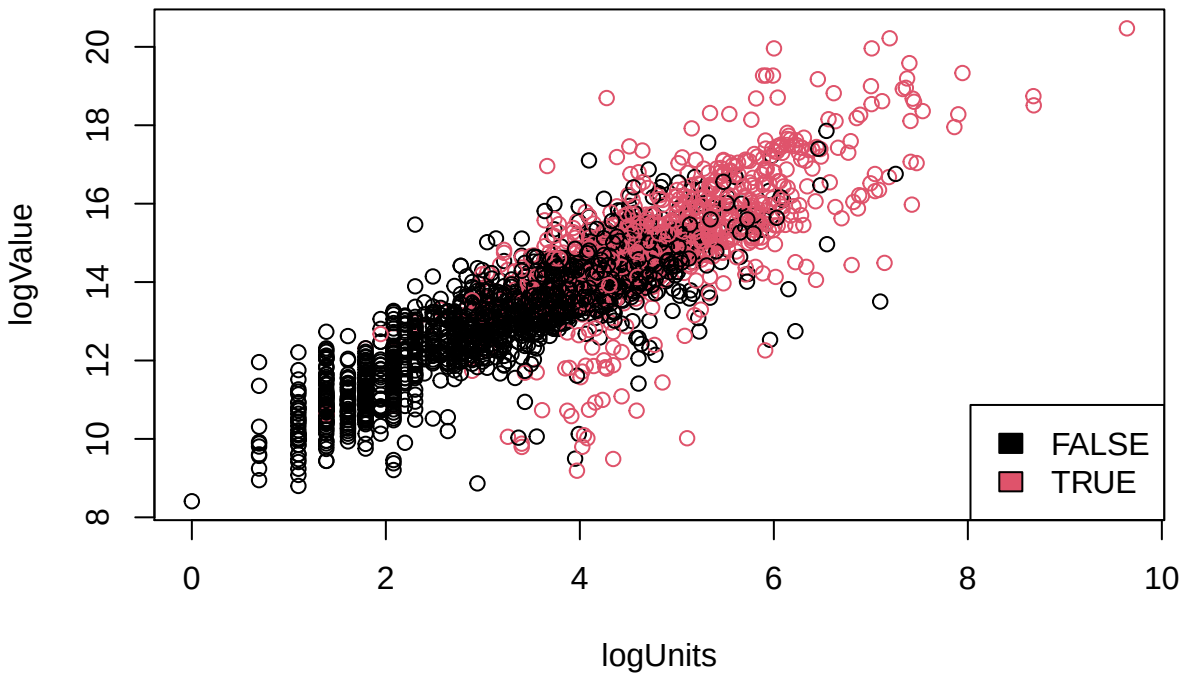
i. Plot of logValue against logUnits.

```
#plot logvalue against logunits
plot(housing$logUnits,housing$logValue, xlab = 'logUnits',ylab = 'logValue')
```



ii. Below is the plot of logValue against logUnits with colors by whether the property was built after 1950 or not. We could see that there is an upward trend in the plot, meaning that the more units a property has, the higher its value is. The coloring tells us which property (data point) is built before 1950 (black) or after 1950 (pink). We could see that generally newer property has higher value and also more units than older property.

```
#plot logvalue against logunits
plot(housing$logUnits,housing$logValue,
     xlab = 'logUnits',ylab = 'logValue',
     col=factor(housing$after1950))
#add legends
legend("bottomright", legend = levels(factor(housing$after1950)),
       fill = unique(factor(housing$after1950)))
```



iii.

(i) Correlation between logValue and logUnits in the whole data is 0.873

```
#corr for whole data
cor(housing$logValue,housing$logUnits)
```

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

(ii) Correlation between logValue and logUnits in just Manhattan is 0.883

```
#corr for whole data
cor(housing[housing$Borough=="Manhattan",]$logValue,
    housing[housing$Borough=="Manhattan",]$logUnits)
```

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

(iii) Correlation between logValue and logUnits in just Brooklyn is 0.910

```
#corr for whole data
cor(housing[housing$Borough=="Brooklyn",]$logValue,
    housing[housing$Borough=="Brooklyn",]$logUnits)
```

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

(iv) Correlation between logValue and logUnits for properties built after 1950 is 0.722

```
#corr for whole data
cor(housing[housing$after1950==T,]$logValue,
    housing[housing$after1950==T,]$logUnits)
```

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

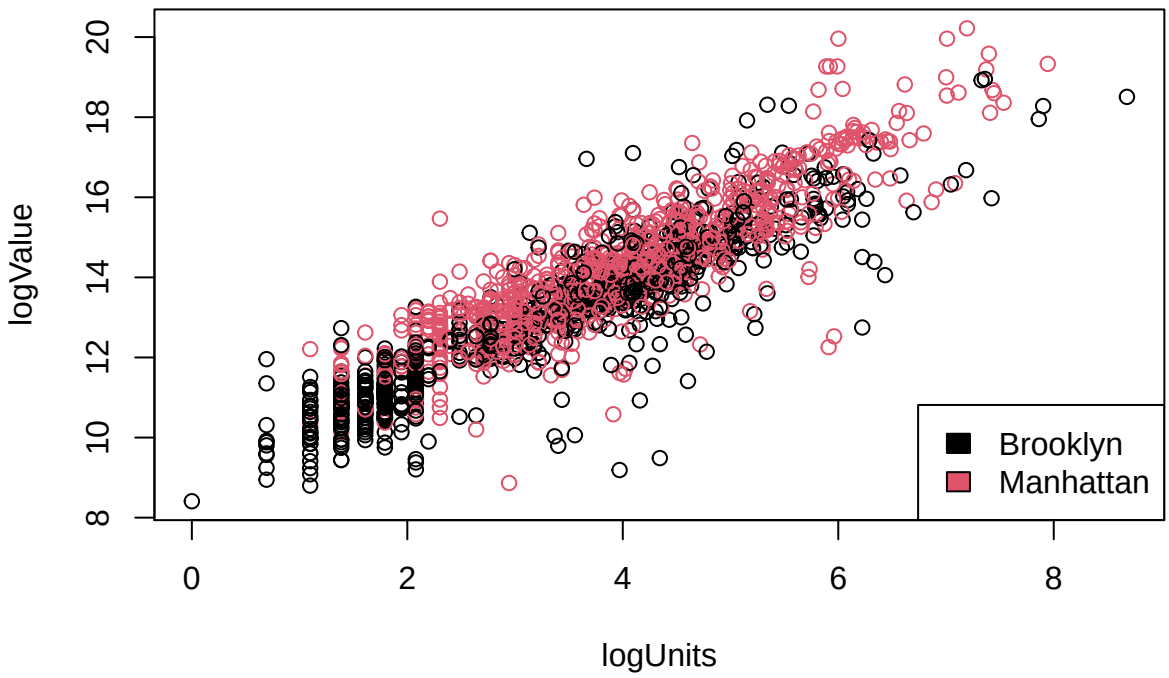
(v) Correlation between logValue and logUnits for properties built before 1950 is 0.864

```
#corr for whole data
cor(housing[housing$after1950==F,]$logValue,
    housing[housing$after1950==F,]$logUnits)
```

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

iv. Plot logValue against logUnits for Manhattan (pink) and Brooklyn (black).

```
#plot logvalue against logunits for Manhattan and Brooklyn
plot(x = housing[housing$Borough%in%c("Brooklyn","Manhattan"),]$logUnits,
     y = housing[housing$Borough%in%c("Brooklyn","Manhattan"),]$logValue,
     xlab = 'logUnits',ylab = 'logValue',
     col=factor(housing[housing$Borough%in%c("Brooklyn","Manhattan"),]$Borough))
#add legends
legend("bottomright",
       legend = levels(factor(housing[housing$Borough%in%c("Brooklyn","Manhattan"),]$Borough)),
       fill = unique(factor(housing[housing$Borough%in%c("Brooklyn","Manhattan"),]$Borough)))
```



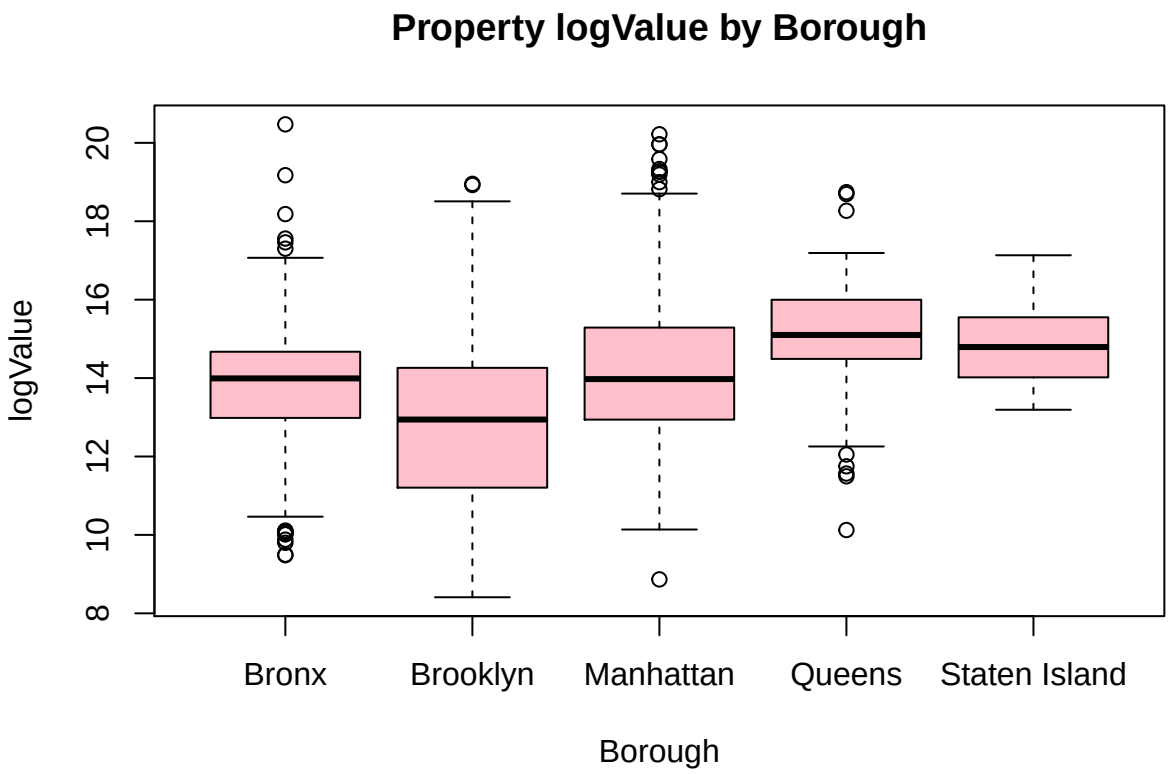
v. The block of code given is essentially for calculating the median property value for properties in Manhattan. We can achieve the same result, which is 1172362, with one single line of code as follows.

```
#median value for manhattan
median(housing[housing$Borough=="Manhattan",]$Value)
```

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

vi. A side-by-side boxplot of logValue across five boroughs is generated as follows.

```
#side by side box plot of logValue by borough
boxplot(housing$logValue ~ housing$Borough, col="pink",
        main="Property logValue by Borough",
        ylab="logValue", xlab="Borough")
```



vii. The median property values for each borough are calculated below using tapply.

```
#side by side box plot of logValue by borough
tapply(housing$Value,housing$Borough,median)
```

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