Homework 1

Shreya Rao sr3843

5/5/2021

Part 1: Loading, Cleaning the Exploring Data in R

i. Load the data into a dataframe called housing.

```
housing <- read.csv("NYChousing.csv")
```

ii. How many rows and columns does the dataframe have?

```
ncol(housing)
```

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

```
old_rows <- nrow(housing)
old_rows
```

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

2506 rows and 22 columns

iii. Run the appropriate function to display the variable names of the dataframe.

```
colnames(housing)
```

```
##  [1] "UID"                          "PropertyName"
##  [3] "Lon"                          "Lat"
##  [5] "AgencyID"                     "Name"
##  [7] "Value"                        "Address"
##  [9] "Violations2010"               "REACNumber"
## [11] "Borough"                      "CD"
## [13] "CityCouncilDistrict"          "CensusTract"
## [15] "BuildingCount"                "UnitCount"
## [17] "YearBuilt"                    "Owner"
## [19] "Rental.Coop"                  "OwnerProfitStatus"
## [21] "AffordabilityRestrictions"    "StartAffordabilityRestrictions"
```

iv. Run this command, and explain, in words, what this does:

```
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
```

This is counting the number of NA values in each column of the housing dataset. is.na(housing) generates a TRUE/FALSE vector that indicates what values are NA. 2 indicates that the function must be performed on the columns. sum is the function that is being performed on the columns.

v. Remove the rows of the dataset for which the variable Value is NA

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

vi. How many rows did you remove with the previous call? Does this agree with your result from (iv)?

```
new_rows <- nrow(housing)
old_rows - new_rows
```

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

52 rows were removed. This agrees with the result from (iv) because the number of NA values in the Value column is 52.

vii. Calculate the third quartile of the property values, i.e., the third quartile Q3 is the 75th percentile. Use the quantile() function to complete this task.

```
Q3 <- quantile(housing$Value, 0.75)
Q3
```

```
##     75%
## 2684851
```

viii. Create a new variable in the dataset called HighValue that is equal to "High" if the property's value is greater than Q3 and is equal to "NotHigh" if the property's value is less than or equal to Q3.

```
housing["HighValue"] = ifelse(housing$Value > Q3, "High", "NotHigh")
```

ix. Display a contingency table that shows the proprtions of HighValue split by Borough. Note that the table() function is the easiest way to tackle this problem but the table() function gives raw counts.

```
cont_table <- prop.table(table(housing$HighValue, housing$Borough))

# round(table(housing$HighValue,
# housing$Borough)/length(housing$HighValue), 7)
cont_table
```

```
##
##                 Bronx    Brooklyn   Manhattan      Queens Staten Island
##   High    0.055827221 0.053789731 0.114506927 0.019559902   0.006519967
##   NotHigh 0.214751426 0.284026080 0.232273839 0.011817441   0.006927465
```

x. What is the proportion of properties whose values are in the upper quartile and are located in The Bronx?

```
mean(housing$HighValue == "High" & housing$Borough == "Bronx")
```

```
## [1] 0.05582722
```

xi. Given a randomly selected property is in The Bronx, what is the probability that its value is in the upper quartile? Solve this question in two ways: (1) by using the table from (ix), and (2) by using logical/relational/filtering commands and using the function mean().

```
mean(housing$HighValue == "High" & housing$Borough == "Bronx")/mean(housing$Borough ==
    "Bronx")
```

```
## [1] 0.2063253
```

xii. Create a new variable in the dataset called logValue that is equal to the logarithm of the property's Value. What are the minimum, median, mean, and maximum values of logValue?

```
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
```

Min: 8.41 Median: 13.75 Mean: 13.68 Max: 20.47

xiii. Create a new variable in the dataset called logUnits that is equal to the logarithm of the number of units in the property. The number of units in each piece of property is stored in the variable UnitCount.

```
housing["logUnits"] <- log(housing$UnitCount)
```

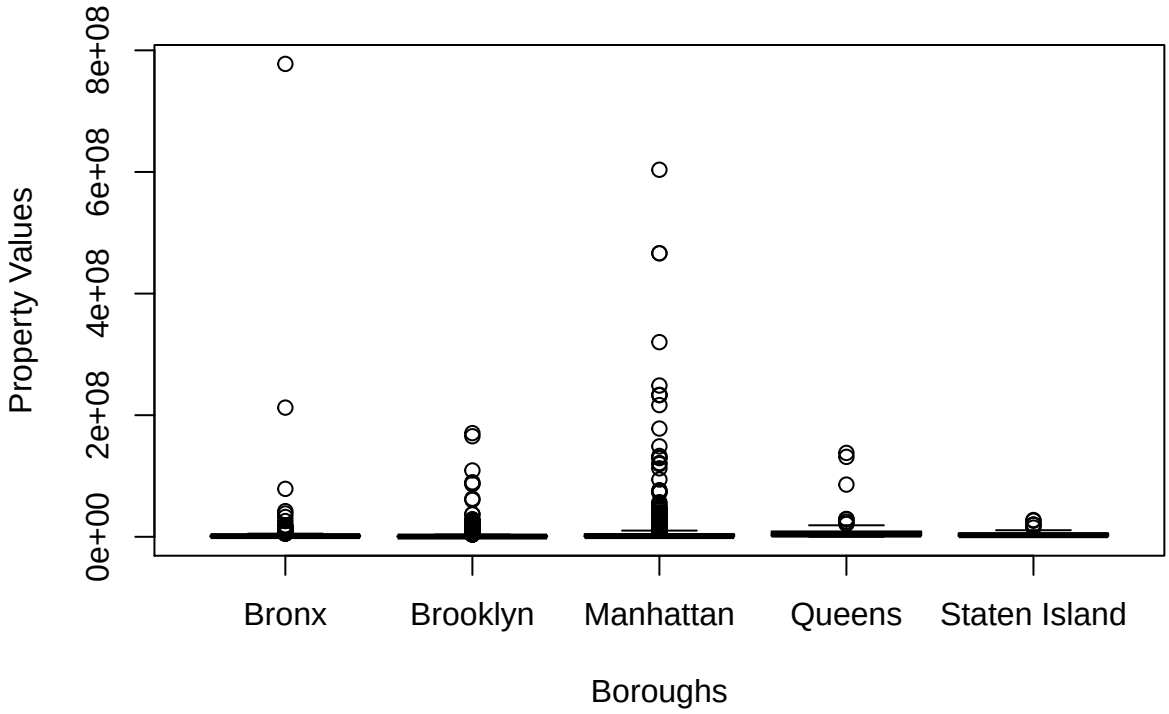
xiv. Finally create a new variable in the dataset called after1950 which equals TRUE if the property was built in or after 1950 and FALSE otherwise. You'll want to use the YearBuilt variable here. This can be done in a single line of code.

```
housing["after1950"] <- ifelse(housing["YearBuilt"] >= 1950,
    TRUE, FALSE)
```

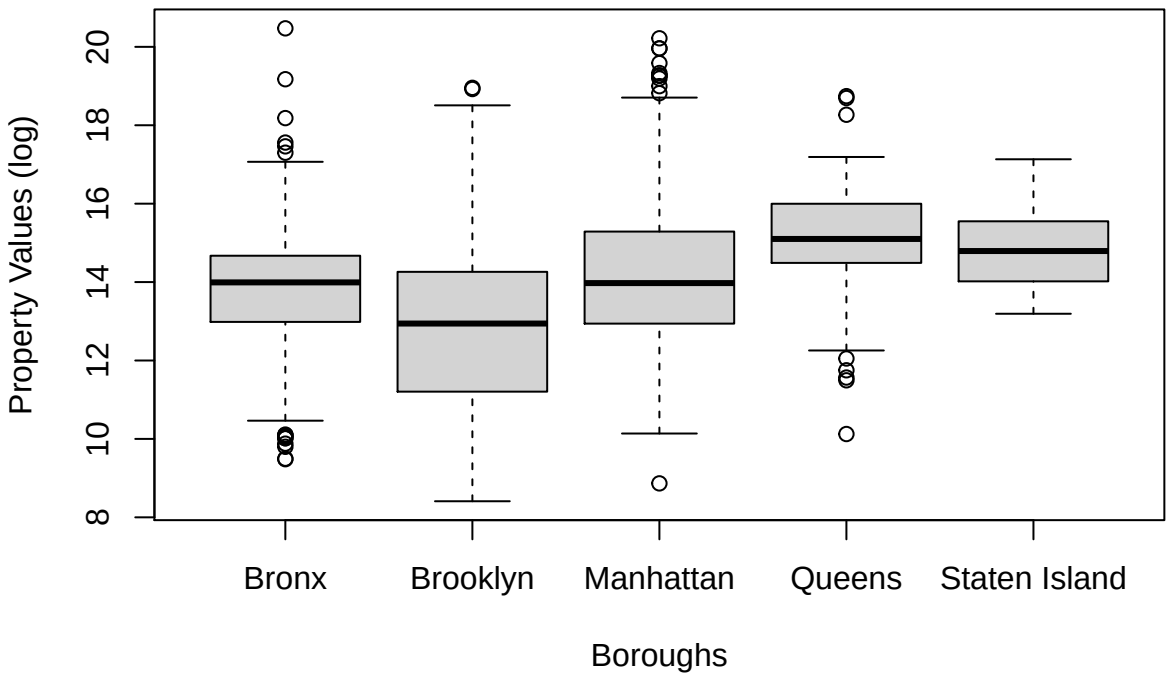
Part 2: EDA

2i. Create a multiple boxplot (side-by-side boxplots) comparing property value across the five boroughs. Create a multiple boxplot (side-by-side boxplots) comparing property logValue across the five boroughs. Make sure to label the plots appropriately.

```
boxplot(Value ~ Borough, data = housing, xlab = "Boroughs", ylab = "Property Values")
```

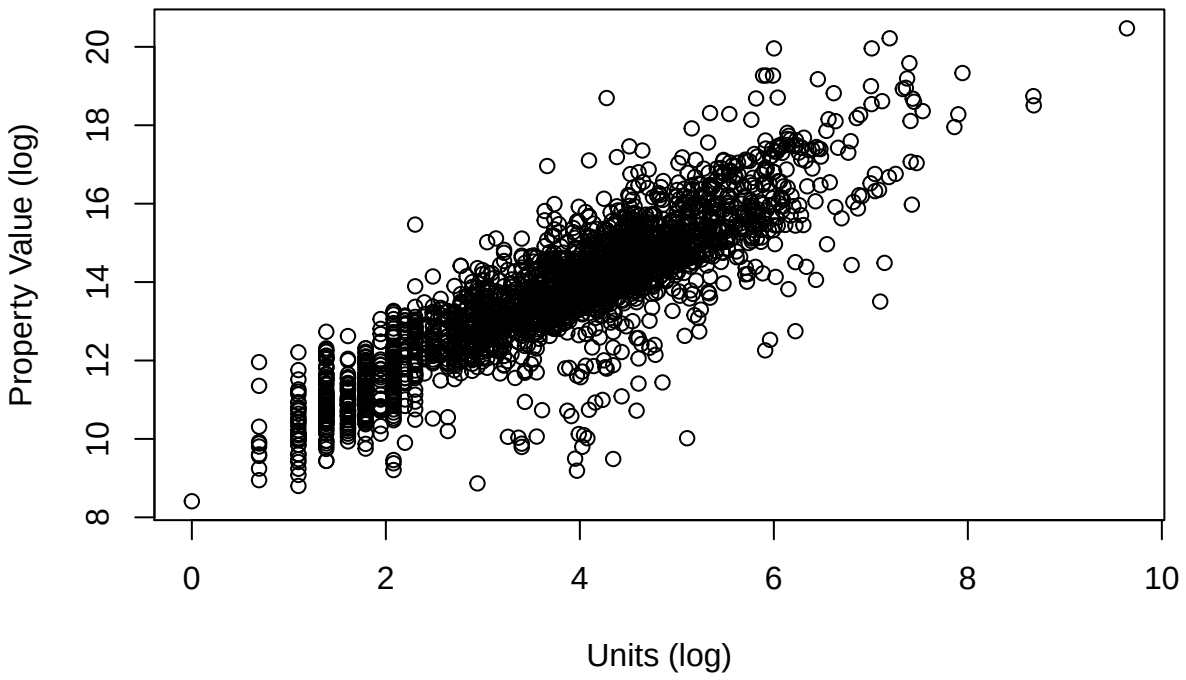



```
boxplot(logValue ~ Borough, data = housing, xlab = "Boroughs",
    ylab = "Property Values (log)")
```



2ii. Plot property logValue against property logUnits. Name the x and y labels of the plot appropriately. logValue should be on the y-axis.

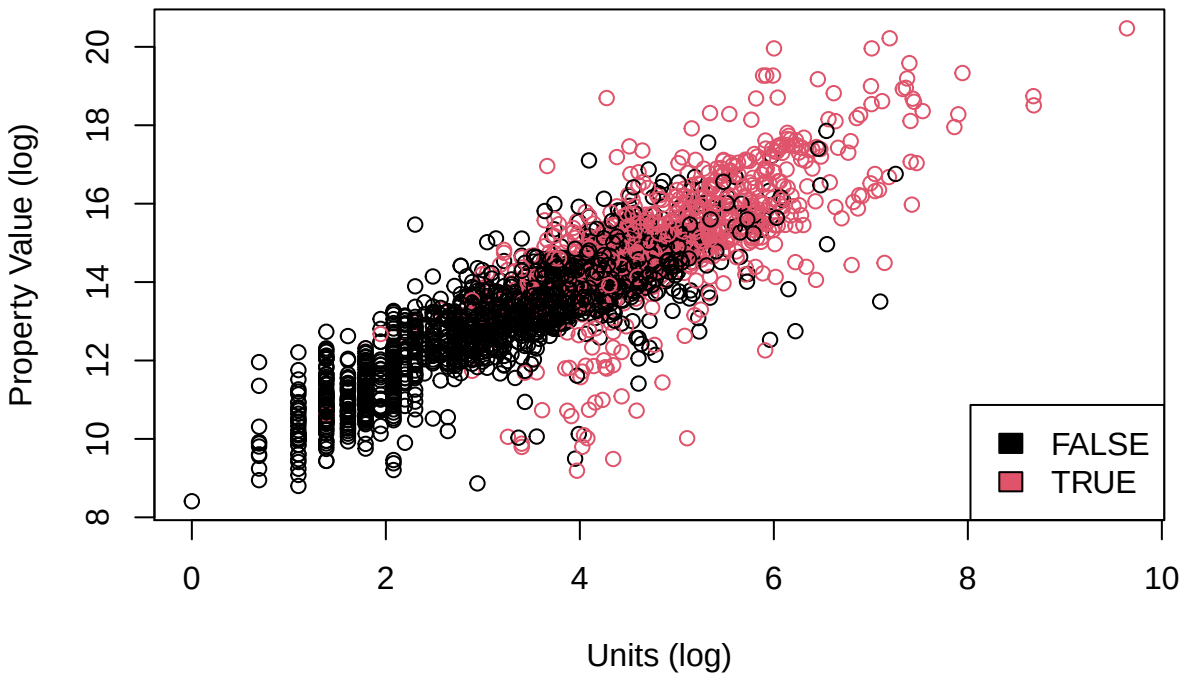
```
plot(housing$logUnits, housing$logValue, xlab = "Units (log)",
    ylab = "Property Value (log)")
```



2iii. Make the same plot as above, but now include the argument col = factor(housing$after1950). Describe this plot and the covariation between the two variables. What does the coloring in the plot tell us?

```
plot(housing$logUnits, housing$logValue, xlab = "Units (log)",
    ylab = "Property Value (log)", col = factor(housing$after1950))

legend("bottomright", legend = levels(factor(housing$after1950)),
    fill = unique(factor(housing$after1950)))
```



This plot shows an approximately linear relationship between Property Value and Number of Units. The coloring indicates that properties built after 1950 tend to have more units and are more expensive that the ones build before 1950.

2iv. The cor() function calculates the correlation coefficient between two variables. What is the correlation between property logValue and property logUnits in (i) the whole data, (ii) just Manhattan (iii) just Brooklyn (iv) for properties built after 1950 (v) for properties built before 1950?

```
# (i) correlation between property logValue and property
# logUnits
cor(housing$logValue, housing$logUnits)
```

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

```
# (ii) correlation between property logValue and property
# logUnits just Manhattan
cor(housing$logValue[housing$Borough == "Manhattan"], housing$logUnits[housing$Borough ==
    "Manhattan"])
```

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

```
# (iii) correlation between property logValue and property
# logUnits just Brooklyn
cor(housing$logValue[housing$Borough == "Brooklyn"], housing$logUnits[housing$Borough ==
    "Brooklyn"])
```

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

```
# (iv) correlation between property logValue and property
# logUnits for properties built after 1950
cor(housing$logValue[housing$after1950 == TRUE], housing$logUnits[housing$after1950 ==
    TRUE])
```

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

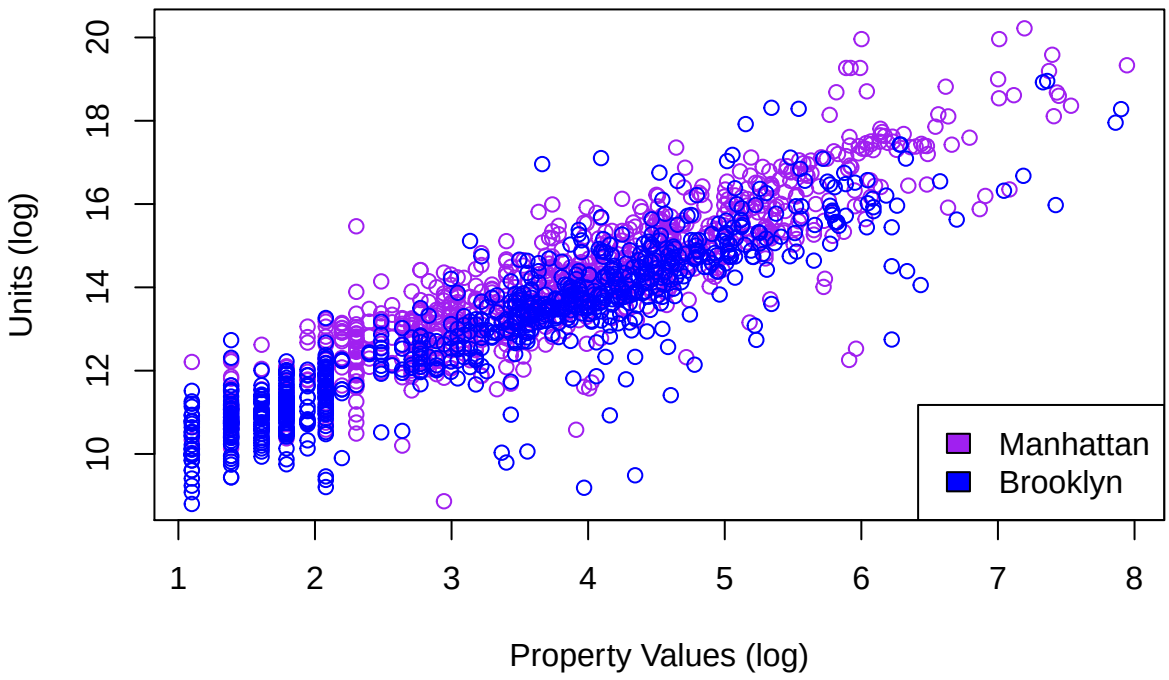
```
# (v) correlation between property logValue and property
# logUnits for properties built before 1950
cor(housing$logValue[housing$after1950 == FALSE], housing$logUnits[housing$after1950 ==
    FALSE])
```

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

2v. Make a single plot showing property logValue against property logUnits for Manhattan and Brooklyn. When creating this plot, clearly distinguish the two boroughs.

```
plot(x = housing$logUnits[housing$Borough == "Manhattan"], y = housing$logValue[housing$Borough ==
    "Manhattan"], col = "Purple", xlab = "Property Values (log)",
    ylab = "Units (log)")
points(x = housing$logUnits[housing$Borough == "Brooklyn"], y = housing$logValue[housing$Borough ==
    "Brooklyn"], col = "Blue")

legend("bottomright", legend = c("Manhattan", "Brooklyn"), fill = c("Purple",
    "Blue"))
```



2vi. Consider the following block of code. Give a single line of R code which gives the same final answer as the block of code.

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

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

2vii. For five boroughs, what are the median property values? (Use Value here, not logValue.)

```
levels(as.factor(housing$Borough))
```

```
## [1] "Bronx"         "Brooklyn"      "Manhattan"     "Queens"
## [5] "Staten Island"
```

```
# Bronx
median(housing$Value[housing$Borough == "Bronx"])
```

```
## [1] 1192950
```

```
# Brooklyn
median(housing$Value[housing$Borough == "Brooklyn"])
```

```
## [1] 417610
```

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

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

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

```
## [1] 3611700
```

```r
# Staten Island
median(housing$Value[housing$Borough == "Staten Island"])
```

```
## [1] 2654100
```