Stats Computing Homework 2

Rocio Meza

February 19, 2017

Part 1.

i. Load the data into a dataframe called housing.

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.2.5
```

```
housing <- read_csv("~/Documents/Statistics/Housing.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   UID = col_integer(),
##   Lon = col_double(),
##   Lat = col_double(),
##   Value = col_integer(),
##   Violations2010 = col_integer(),
##   REACNumber = col_integer(),
##   CityCouncilDistrict = col_integer(),
##   BuildingCount = col_integer(),
##   UnitCount = col_integer(),
##   YearBuilt = col_integer(),
##   StartAffordabilityRestrictions = col_integer()
## )
```

```
## See spec(...) for full column specifications.
```

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

```
dim.data.frame(housing)
```

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

It has 2056 rows and 22 columns.

iii. 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                           11
##                        Value                      Address
##                           52                           45
##               Violations2010                   REACNumber
##                            0                         1873
##                      Borough                           CD
```

```
##                              0                              0
##            CityCouncilDistrict                    CensusTract
##                             10                             19
##                  BuildingCount                      UnitCount
##                              0                              0
##                      YearBuilt                          Owner
##                              0                             29
##                    Rental.Coop              OwnerProfitStatus
##                              0                           1164
##     AffordabilityRestrictions StartAffordabilityRestrictions
##                              0                              5
```

This command shows the number of values that are NA.

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

```
Housing1 <-housing[complete.cases(housing),]
```

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

```
dim.data.frame(Housing1)
```

```
## [1] 96 22
```

1960 rows were removed.

vi. 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?

```
Housing1$logValue <-log(Housing1$Value)
summary(Housing1$logValue)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.68   14.54   14.97   15.18   15.71   19.96
```

vii. 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.

```
Housing1$logUnits<-log(Housing1$UnitCount)
```

viii. 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.

```
Housing1$after1950<-ifelse(Housing1$YearBuilt>1950,"TRUE","FALSE")
```

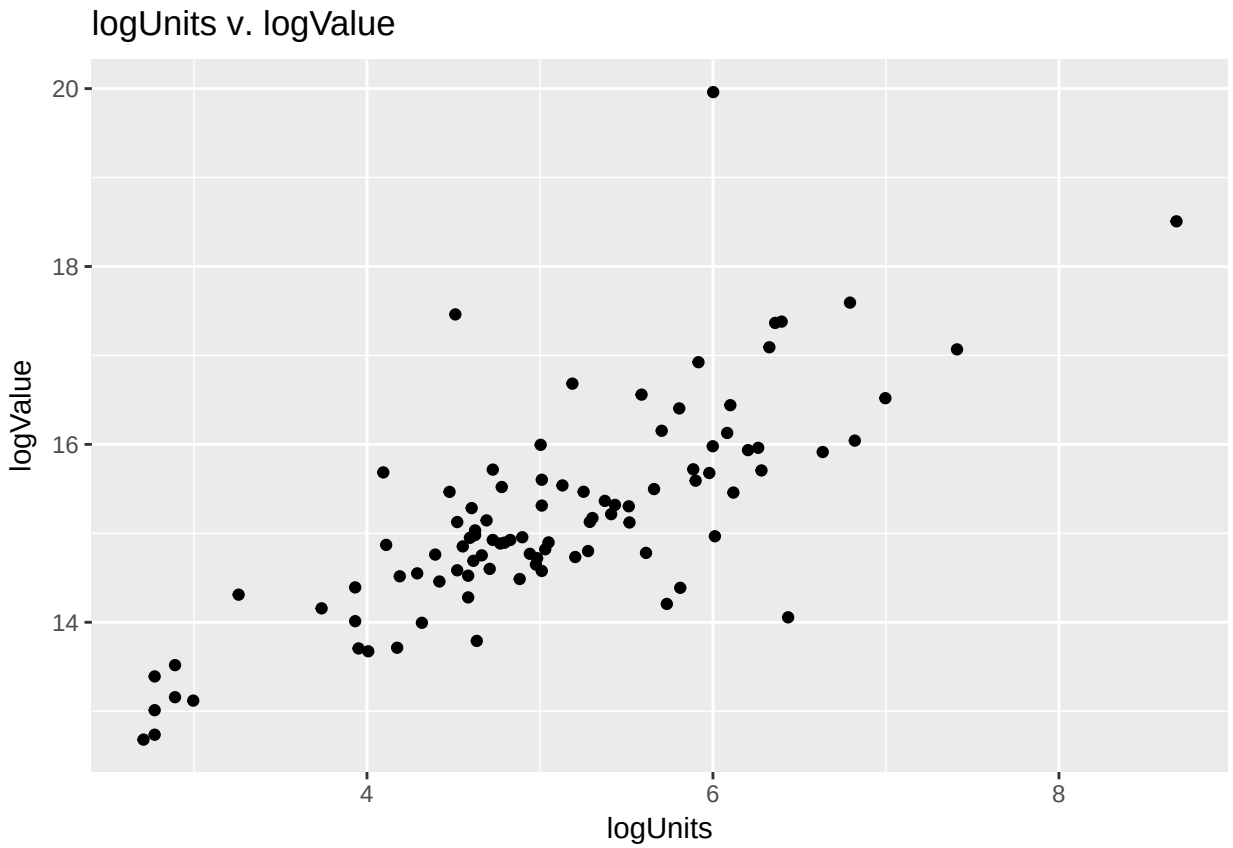
Part 2.

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

```
library(ggplot2)
```

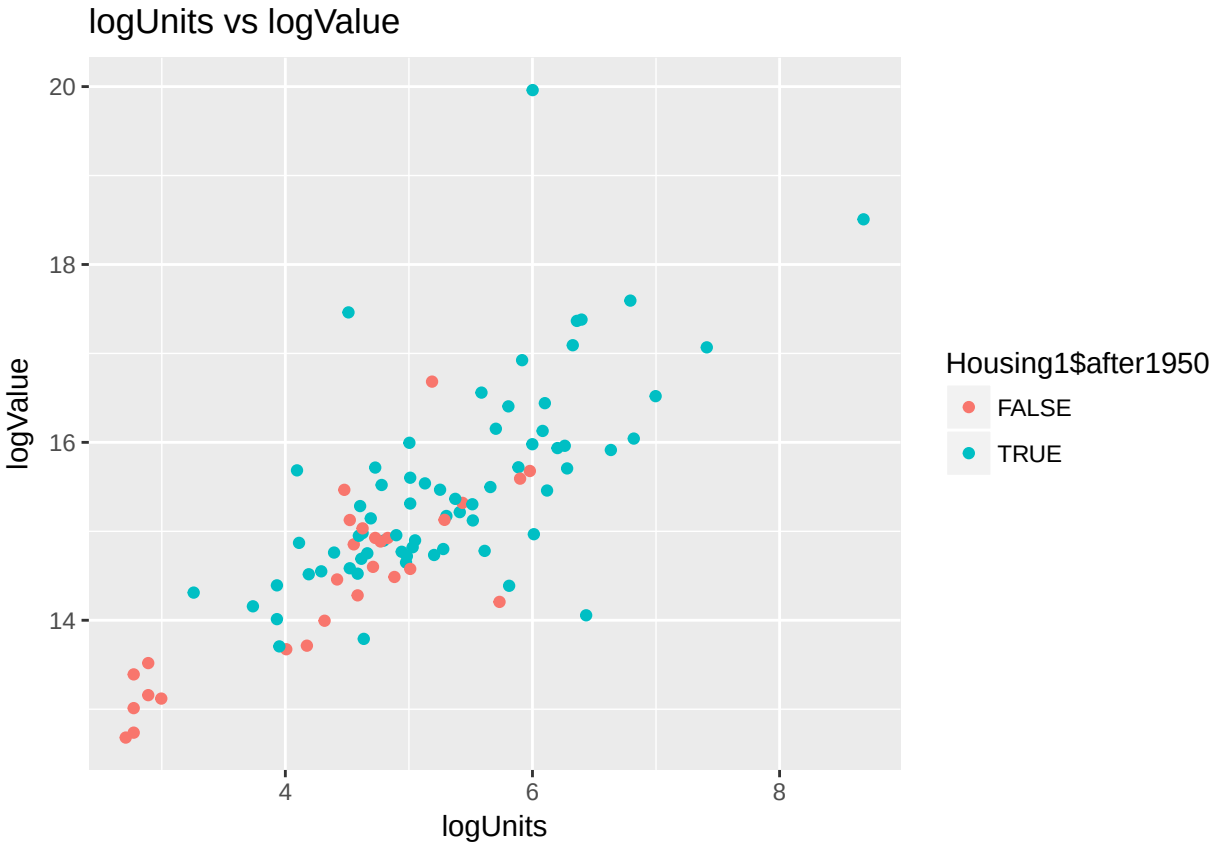
```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
qplot(Housing1$logUnits, Housing1$logValue, xlab="logUnits", ylab="logValue", main="logUnits v. logValu
```



ii. 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?

```
qplot(Housing1$logUnits, Housing1$logValue, col=Housing1$after1950, xlab="logUnits", ylab="logValue", ma
```



Properties before 1950 have less units and lower log-values than properties built after 1950.

iii. 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?

```
cor(Housing1$logValue, Housing1$logUnits)
```

```
## [1] 0.762187
```

```
just.mann <- Housing1[Housing1$Borough == "Manhattan",]
cor(just.mann$logValue,just.mann$logUnits)
```

```
## [1] 0.836214
```

```
just.brook <-Housing1[Housing1$Borough == "Brooklyn",]
cor(just.brook$logValue,just.brook$logUnits)
```

```
## [1] 0.7892677
```

```
just.after1950 <-Housing1[Housing1$after1950 == "TRUE",]
cor(just.after1950$logValue,just.after1950$logUnits)
```

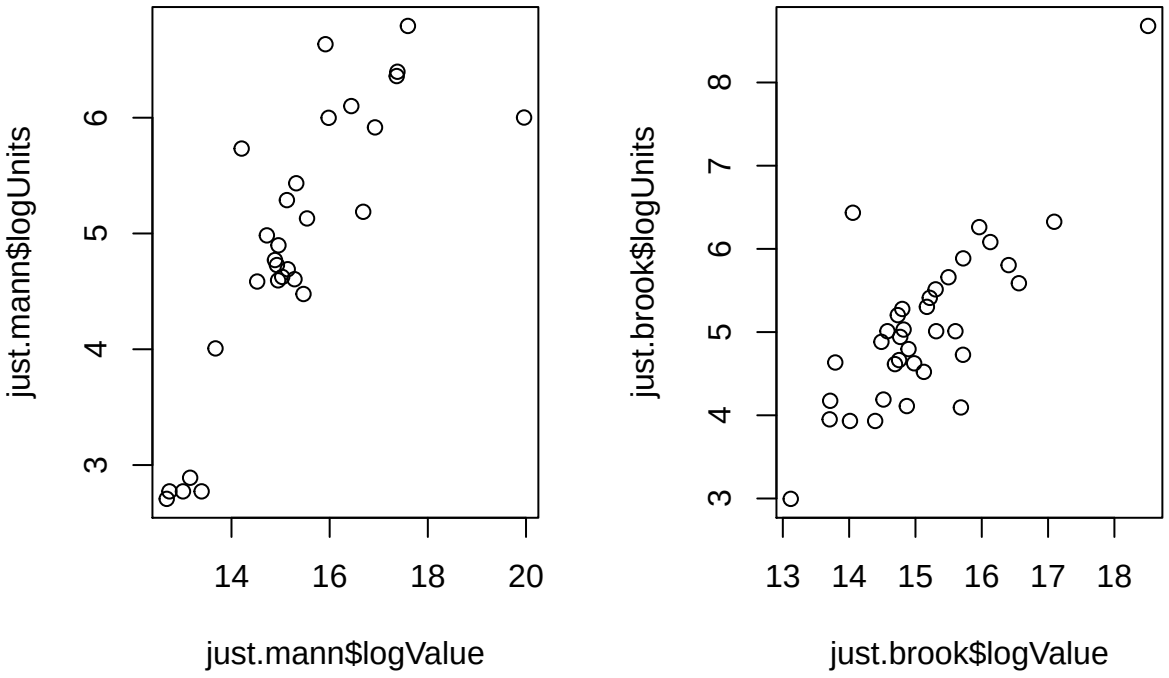
```
## [1] 0.6631507
```

```
just.before1950 <-Housing1[Housing1$after1950 == "FALSE",]
cor(just.before1950$logValue,just.before1950$logUnits)
```

```
## [1] 0.8494562
```

iv. Make two plots showing property logValue against property logUnits for Manhattan and Brooklyn. (If you can fit the information into one plot, clearly distinguishing the two boroughs, that's OK too.)

```
par(mfrow=c(1,2))
plot(just.mann$logValue, just.mann$logUnits)
plot(just.brook$logValue, just.brook$logUnits)
```



v. Consider the following block of code. Give a single line of R code which gives the same final answer as the block of code. There are a few ways to do this.

```
manhat.props <- c()
for (props in 1:nrow(housing)) {
if (housing$Borough[props] == "Manhattan") {
manhat.props <- c(manhat.props, props)
}
manhat.props

}
med.value <- c()
for (props in manhat.props) {
med.value <- c(med.value, housing$Value[props])
}
med.value <- median(med.value, na.rm = TRUE)
med.value
```

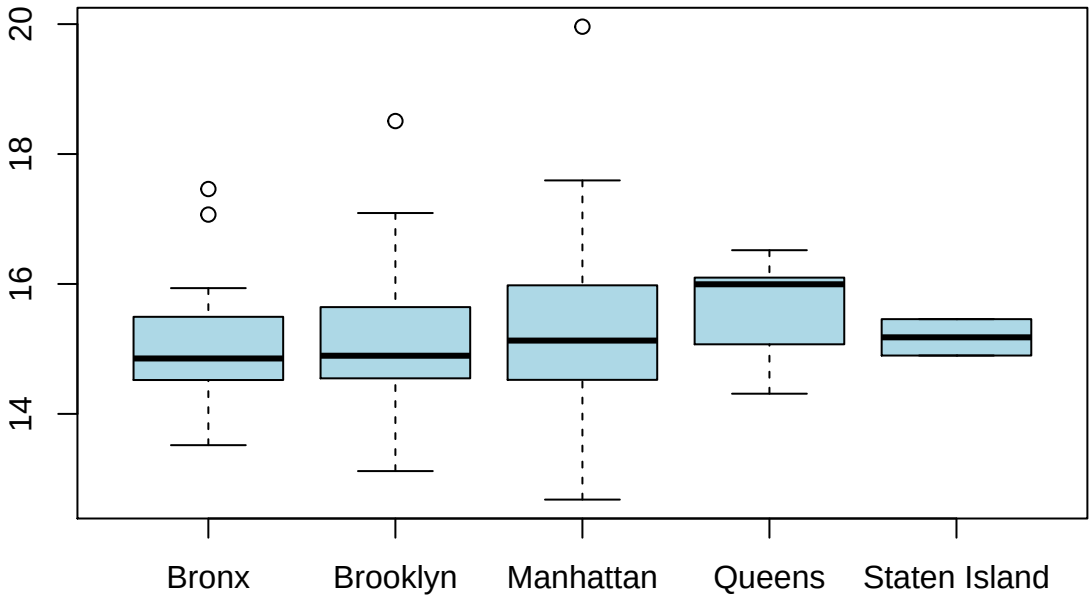
```
## [1] 1172362
```

```
apply(housing[housing$Borough=="Manhattan", "Value"], 2, median, na.rm = TRUE)
```

```
##   Value
## 1172362
```

vii. Make side-by-side box plots comparing property logValue across the fivve boroughs.

```
boxplot(logValue ~ Borough, data = Housing1, col = "lightblue")
```



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

```
apply(Housing1[Housing1$Borough=="Manhattan", "Value"], 2, median, na.rm = TRUE)
```

```
##   Value
## 3717720
```

```
apply(housing[housing$Borough=="Brooklyn", "Value"], 2, median, na.rm = TRUE)
```

```
##  Value
## 417610
```

```
apply(housing[housing$Borough=="Bronx", "Value"], 2, median, na.rm = TRUE)
```

```
##   Value
## 1192950
```

```
apply(housing[housing$Borough=="Queens", "Value"], 2, median, na.rm = TRUE)
```

```
##   Value
## 3611700
```

```
apply(housing[housing$Borough=="Staten Island", "Value"], 2, median, na.rm = TRUE)
```

```
##   Value
## 2654100
```