Hwk2_bs2996

Bangda Sun

September 27, 2016

Part 1: Loading and cleaning data in **R**

i. Load the data into a dataframe called **housing**

```
housing <- read.csv("C:\\Users\\Bangda\\Desktop\\GR5206 Materials\\NYChousing.csv", header = T)
```

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

```
rows <- dim(housing)[1]; rows
```

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

```
columns <- dim(housing)[2]; columns
```

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

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                               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
```

It counts the number of `NA` in each column. Say there are 15 `NA` in variable **Lon** and **Lat**.

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

```
housing2 <- na.omit(housing)
```

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

```
na.rows <- dim(housing)[1] - dim(housing2)[1]; na.rows # number of rows being removed
```

```
## [1] 1876
```

```
na.count <- apply(housing, 1, function(x) length(which(!is.na(x)))) # cont the na in each row
na.rows == length(which(na.count < 22)) # check if it is same with (iv)
```

```
## [1] TRUE
```

In (iii), we count the number of `NA` in each column, here we remove the rows that contains `NA` in it, it is possible that there are some `NA` in one column and other `NA` in another column that two columns of `NA` with no overlap part, therefore we cannot find the quantity relationship between the number of `NA` in rows and columns.

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**?

```
housing2$logValue <- log(housing2$Value)
summary.logValue <- c(0,0,0,0)
names(summary.logValue) <- c("Min", "Median", "Mean", "Max")
summary.logValue[1] <- min(housing2$logValue) # min
summary.logValue[2] <- median(housing2$logValue) # median
summary.logValue[3] <- mean(housing2$logValue) # mean
summary.logValue[4] <- max(housing2$logValue) # max
summary.logValue
```

```
##      Min   Median     Mean      Max
## 10.06002 14.64957 14.64720 20.21847
```

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**.

```
housing2$logUnits <- log(housing2$UnitCount)
```

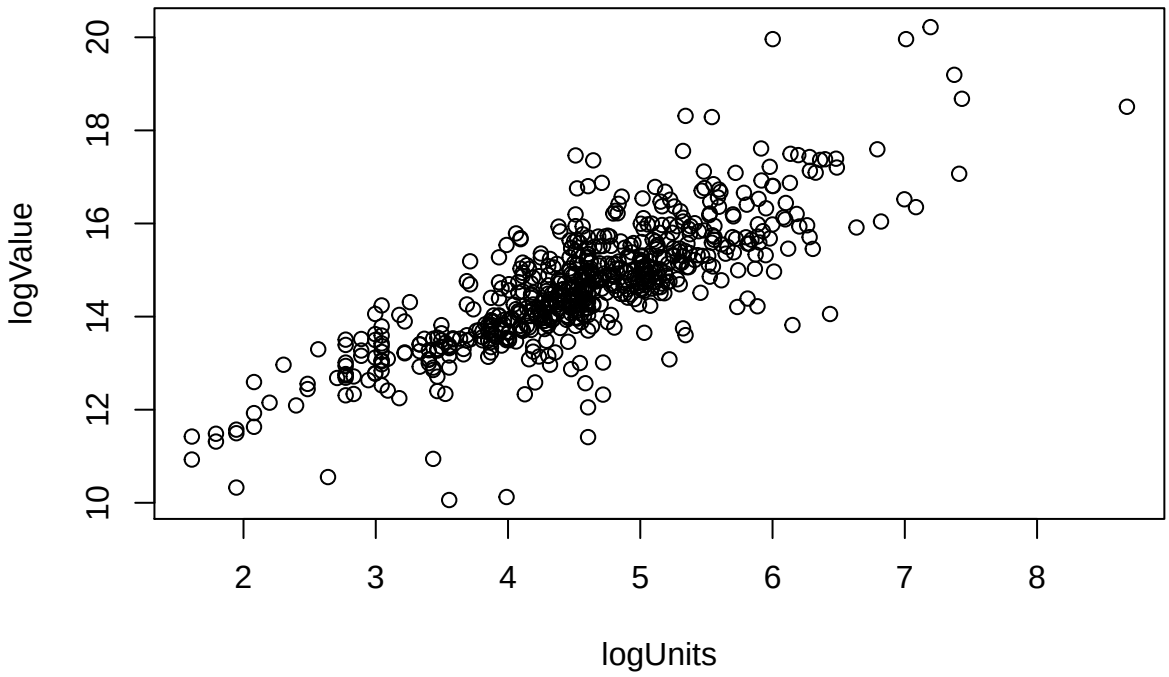
viii. Finally create a new variable in the dataset called **after1950** which equals `TRUE` if the property was built in or after 1950 and `FALSE` otherwise. You'll want to use the **YearBuilt** variable here. This can be done in a single line of code.

```
housing2$after1950 <- (housing2$YearBuilt >= 1950)
```

Part 2: EDA

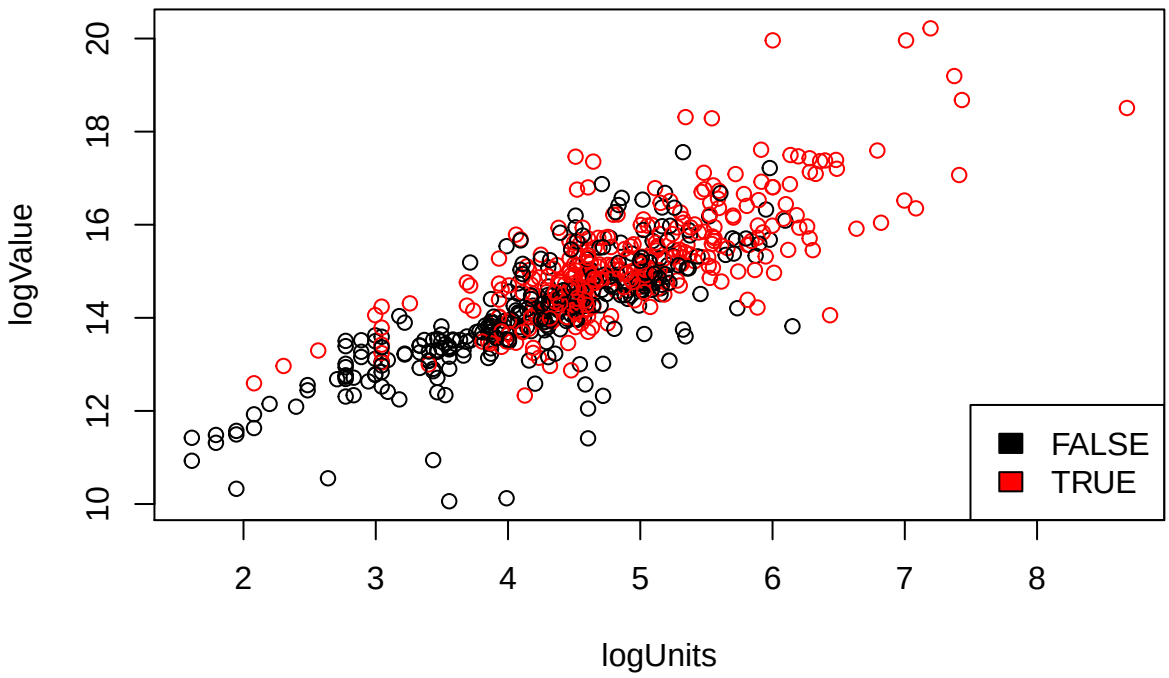
i. Plot property logValue against property **logUnits**. Name the x and y labels of the plot appropriately. **logValue** should be on the y-axis.

```
plot(housing2$logUnits, housing2$logValue,'p', xlab = 'logUnits', ylab = 'logValue')
```



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?

```
housing2$after1950 <- factor(housing2$after1950)
plot(housing2$logUnits, housing2$logValue, xlab = 'logUnits', ylab = 'logValue', col = housing2$after195
legend("bottomright", legend = levels(housing2$after1950), fill
= unique(housing2$after1950))
```



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(housing2$logValue, housing2$logUnits) # whole data
```

```
## [1] 0.7988655
```

```
cor(housing2$logValue[which(housing2$Borough == 'Manhattan')], housing2$logUnits[which(housing2$Borough
```

```
## [1] 0.8710823
```

```
cor(housing2$logValue[which(housing2$Borough == 'Brooklyn')], housing2$logUnits[which(housing2$Borough =
```

```
## [1] 0.8053241
```

```
cor(housing2$logValue[which(housing2$after1950 == T)], housing2$logUnits[which(housing2$after1950 == T)]
```

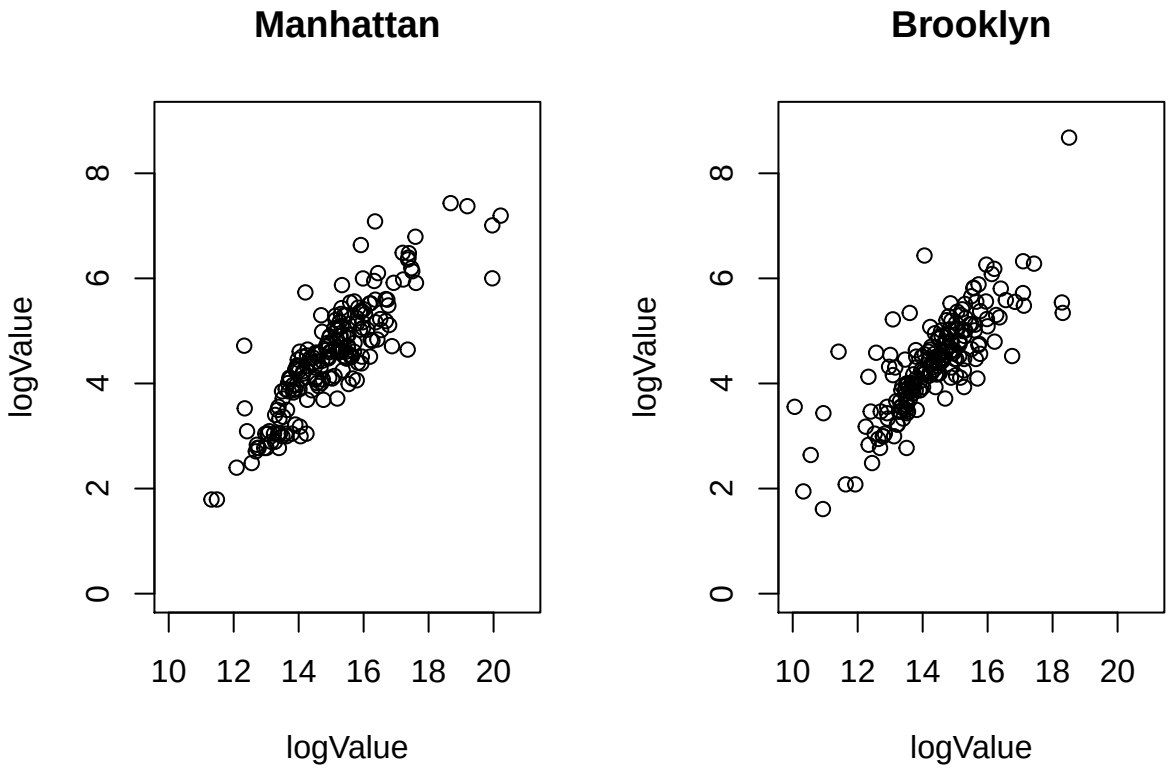
```
## [1] 0.746731
```

```r
cor(housing2$logValue[which(housing2$after1950 == F)], housing2$logUnits[which(housing2$after1950 == F)]
```

```
## [1] 0.7720285
```

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.)

```r
par(mfrow = c(1,2))
plot(housing2$logValue[which(housing2$Borough == 'Manhattan')], housing2$logUnits[which(housing2$Boroug
plot(housing2$logValue[which(housing2$Borough == 'Brooklyn')], housing2$logUnits[which(housing2$Borough
```



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.

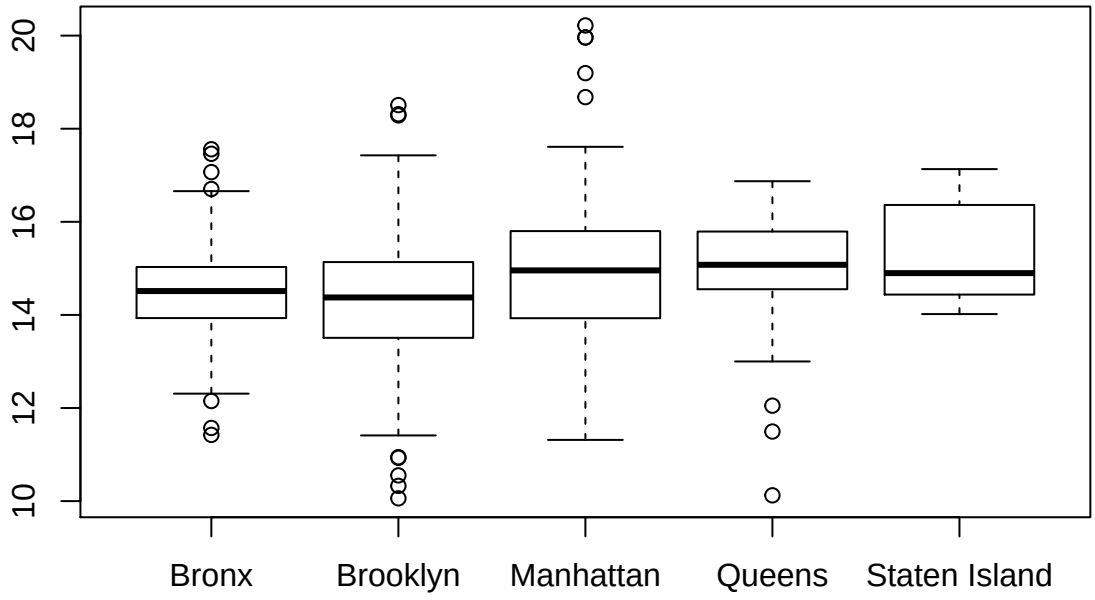
```r
manhat.props <- c()
for (props in 1:nrow(housing2)) {
  if (housing2$Borough[props] == "Manhattan") {
      manhat.props <- c(manhat.props, props)
  }
}
med.value <- c()
for (props in manhat.props) {
  med.value <- c(med.value, housing2$Value[props])
}
med.value <- median(med.value, na.rm = TRUE)
```

The goal of the codes above is to find the median value of house in Manhattan,

```
med.value <- median(housing2$Value[which(housing2$Borough == 'Manhattan')])
```

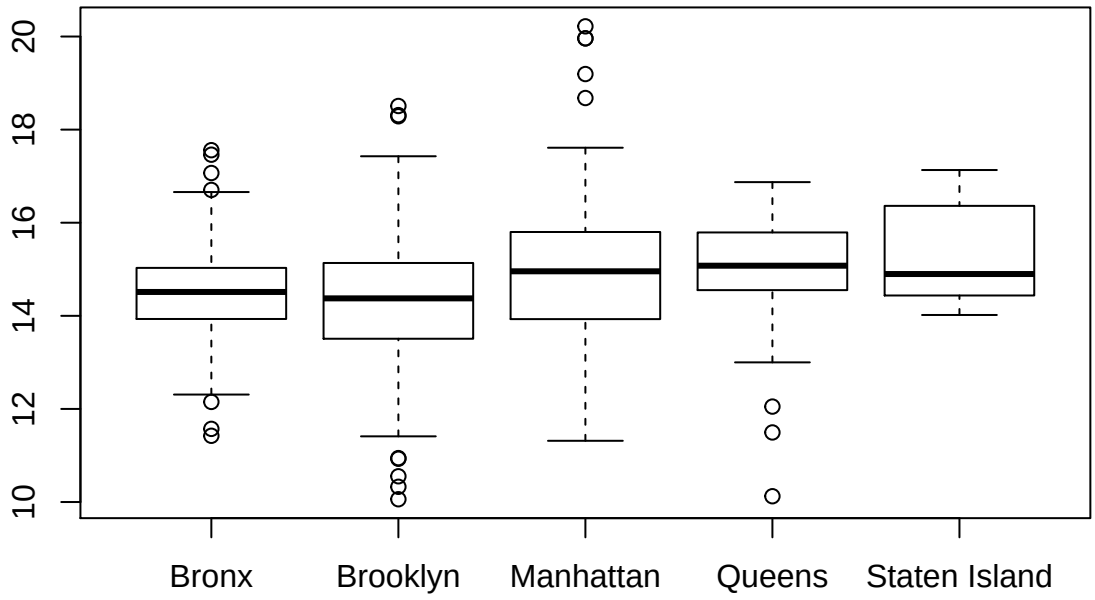
vi. Make side-by-side box plots comparing property **logValue** across the five boroughs.

```
boxplot(housing2$logValue ~ housing2$Borough)
```



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

```
median <- boxplot(housing2$logValue ~ housing2$Borough)$stat[3,]
```




```
names(median) <- c("Bronx", "Brooklyn", "Manhattan", "Queens", "Staten Island")
median <- exp(median); median
```

```
##         Bronx      Brooklyn     Manhattan        Queens Staten Island
##       2008260       1749461       3129300       3529800       2952900
```