Assignment 4

gzahn

December 30, 2018

Assignment 4

In this assignment, you will use R (within R-Studio) to:

- Use several methods to load external data files into R
- Explore parameters available for customizing the way data is read from a file
- Use base R tools to view attributes of data sets
- Perform simple summary statistics
- Begine to do basic data set manipulations such as transposing and subsetting
- Use the built-in visualization methods to quickly look at relationships in your data
- Demonstrate one method for saving visualizations to an image file

All file paths should be relative, starting from the Assignment_3 directory!! (where you found this file)

This means that you need to create a new R-Project named "Assignment_3.Rproj" in your Assignment_3 directory, and work from scripts within that.

For credit...

1. Push a completed version of your Rproj and R-script (details at end of this assignment) to GitHub
2. Your score will also depend on whether any files generated in this workflow are found in your repository
3. Upload a copy of a plaintext file with numbered answers to the **bolded** assignment questions to Canvas. This shows that you worked through the assignment and lets me know to pull a fresh copy of your GitHub repo to grade.

It would be terribly inconvenient if R made us manually enter our data. Thankfully, there are dozens (hundreds?) of ways that we can read external data into R for analysis. Most of the time the data we want to analyze comes in the form of an Excel spreadsheet. There are special ways to import Excel spreadsheets directly, but typically we don't want to store our data as .xlsx because it's a large bloated file format. People who work with data in "rectangular" format (like a spreadsheet) often use a form called "comma-separated-values," or .csv

We will use the built-in function read.table() to load some data.

```r
?read.table() #This brings up the help file
df = read.csv("../../Data/landdata-states.csv") # why did I change to read.csv ???
class(df) # what type of object is df?
```

```
## [1] "data.frame"
```

```r
head(df) # shows the first 6 elements of an object (first 6 rows if you give it a data frame)
```

```
##   State region    Date Home.Value Structure.Cost Land.Value
## 1    AK   West 2010.25     224952         160599      64352
## 2    AK   West 2010.50     225511         160252      65259
## 3    AK   West 2009.75     225820         163791      62029
## 4    AK   West 2010.00     224994         161787      63207
## 5    AK   West 2008.00     234590         155400      79190
## 6    AK   West 2008.25     233714         157458      76256
##   Land.Share..Pct. Home.Price.Index Land.Price.Index Year Qrtr
## 1             28.6            1.481            1.552 2010    1
## 2             28.9            1.484            1.576 2010    2
## 3             27.5            1.486            1.494 2009    3
## 4             28.1            1.481            1.524 2009    4
## 5             33.8            1.544            1.885 2007    4
## 6             32.6            1.538            1.817 2008    1
```

Now, we have a data frame loaded into R as an object called "df." If you open that same file with a plain text editor you'd see a bunch of values separated by commas. The read.csv() function is a convenient way to tell R that those commas represent different values and each "\n" (newline) character means a new row. It automatically treats the first row as column headers.

Questions:

- **1. What other stuff does read.csv() do automatically?**
- **2. How is it different from read.csv2()?**
- **3. Why does read.csv2() even exist?**

Now, I notice that each column in this data frame has its own class. Let's look at a couple

```r
class(df$State)
```

```
## [1] "factor"
```

```r
class(df$Date)
```

```
## [1] "numeric"
```

Questions:

- **4. How could I change the parameters of read.csv() to make it so the class of the "State" column is "character" instead of "factor?"**

Now, let's explore this data set a bit with basic descriptive stats...

```r
dim(df) # dimensions of the data frame (rows, columns)
```

```
## [1] 7803   11
```

```r
str(df) # another nice way to glimpse a data frame
```

```
## 'data.frame':    7803 obs. of  11 variables:
##  $ State           : Factor w/ 51 levels "AK","AL","AR",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ region          : Factor w/ 4 levels "Midwest","N. East",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Date            : num  2010 2010 2010 2010 2008 ...
##  $ Home.Value      : int  224952 225511 225820 224994 234590 233714 232999 232164 231039 229395 ...
##  $ Structure.Cost  : int  160599 160252 163791 161787 155400 157458 160092 162704 164739 165424 ...
##  $ Land.Value      : int  64352 65259 62029 63207 79190 76256 72906 69460 66299 63971 ...
##  $ Land.Share..Pct.: num  28.6 28.9 27.5 28.1 33.8 32.6 31.3 29.9 28.7 27.9 ...
##  $ Home.Price.Index: num  1.48 1.48 1.49 1.48 1.54 ...
##  $ Land.Price.Index: num  1.55 1.58 1.49 1.52 1.89 ...
##  $ Year            : int  2010 2010 2009 2009 2007 2008 2008 2008 2008 2009 ...
##  $ Qrtr            : int  1 2 3 4 4 1 2 3 4 1 ...
```

```r
summary(df) # summary() works differently for different data classes. Note how it summarizes factors us
```

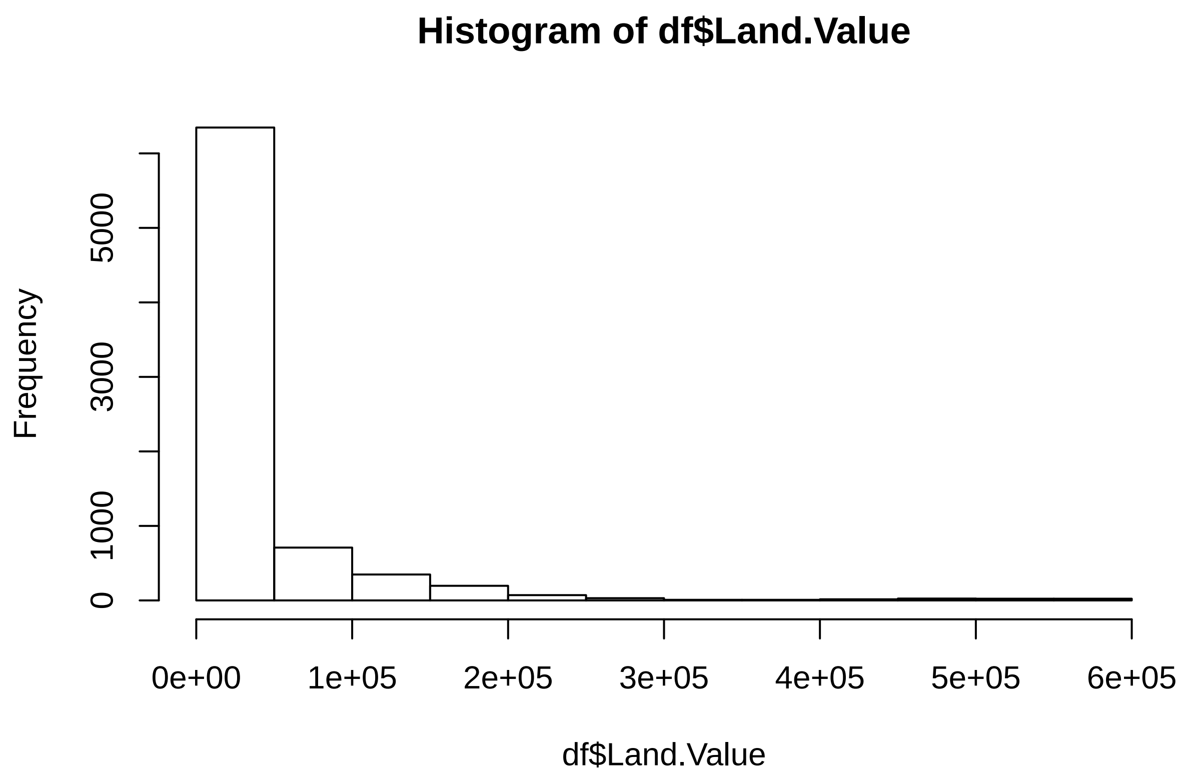
```
##      State           region          Date         Home.Value
##  AK      : 153   Midwest:1836   Min.   :1975   Min.   : 18763  
##  AL      : 153   N. East:1377   1st Qu.:1985   1st Qu.: 62235  
##  AR      : 153   South  :2448   Median :1994   Median :108724  
##  AZ      : 153   West   :1989   Mean   :1994   Mean   :135313  
##  CA      : 153   NA's   : 153   3rd Qu.:2004   3rd Qu.:172030  
##  CO      : 153                  Max.   :2013   Max.   :862885  
##  (Other):6885                                                  
##  Structure.Cost     Land.Value     Land.Share..Pct. Home.Price.Index
##  Min.   : 17825   Min.   :   938   Min.   : 5.00    Min.   :0.1350  
##  1st Qu.: 53776   1st Qu.:  4178   1st Qu.: 5.00    1st Qu.:0.4550  
##  Median : 88352   Median :  9478   Median :10.40    Median :0.7830  
##  Mean   : 99534   Mean   : 35779   Mean   :18.17    Mean   :0.8695  
##  3rd Qu.:134871   3rd Qu.: 38631   3rd Qu.:26.30    3rd Qu.:1.2075  
##  Max.   :325595   Max.   :594417   Max.   :81.70    Max.   :2.8930  
##                                                                     
##  Land.Price.Index       Year           Qrtr     
##  Min.   : 0.0000   Min.   :1975   Min.   :1.00  
##  1st Qu.: 0.0020   1st Qu.:1984   1st Qu.:1.00  
##  Median : 0.2520   Median :1994   Median :2.00  
##  Mean   : 0.9912   Mean   :1994   Mean   :2.49  
##  3rd Qu.: 1.1510   3rd Qu.:2003   3rd Qu.:3.00  
##  Max.   :15.4340   Max.   :2013   Max.   :4.00  
## 
```

Questions:

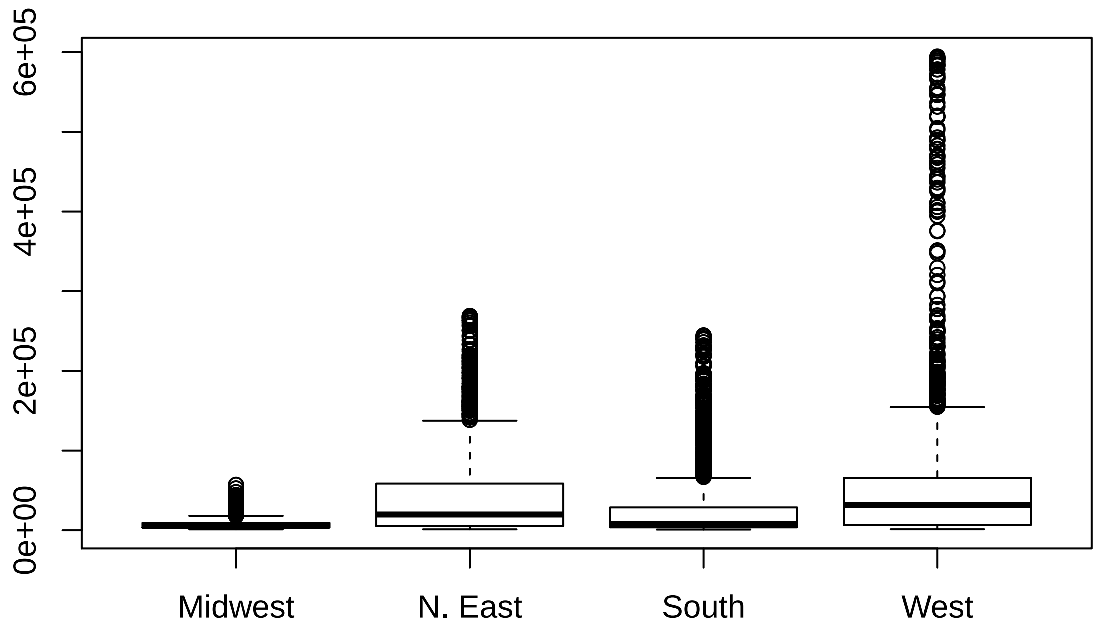
- **5. What command would give the summary stats for ONLY the Home.Value column?**
- **6. What value is returned by the command: names(df)[4] ?**

We can do some very basic visualizations of our data as well. In many cases, a good image is much more descriptive than a boring table of summary statistics…

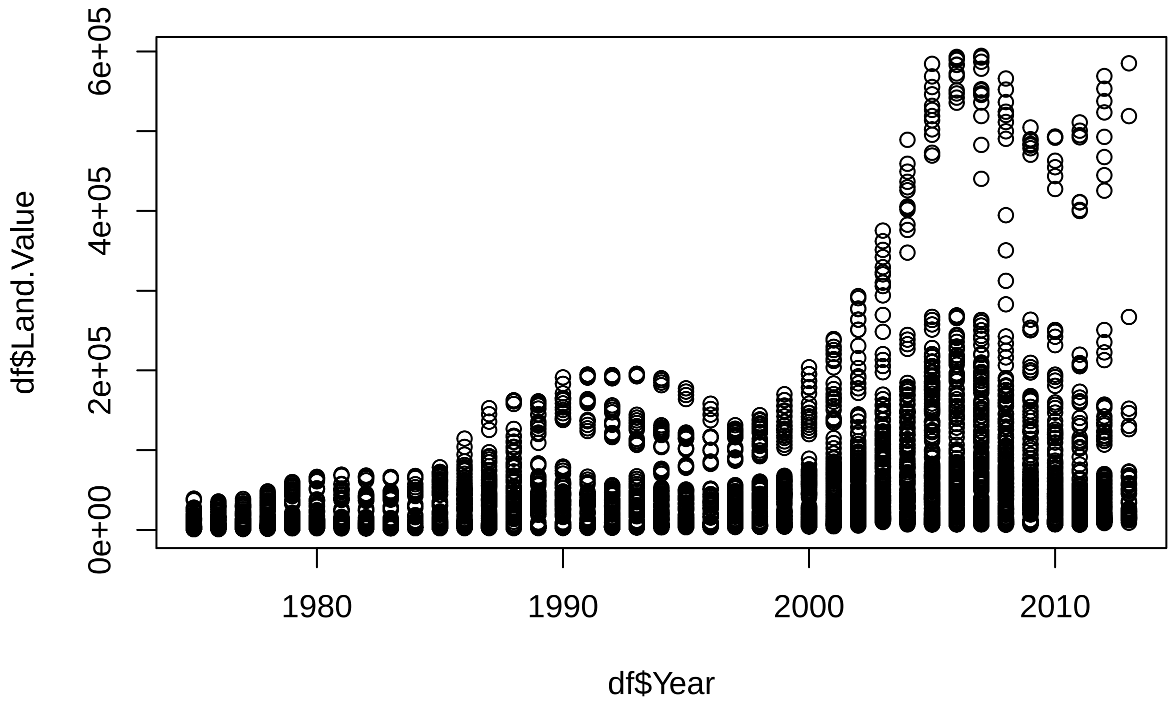
```
hist(df$Land.Value) # histogram showing number of times each numeric value was seen in the vector "Land
```



```
# If you want to look at land value by region, you could do this:
plot(x=df$region,y=df$Land.Value)
```



```
# Land value by year
plot(x=df$Year,y=df$Land.Value)
```



Note that the plot() function automatically tries to determine the best type of plot for your data based on the classes of vectors that you give it.

Questions:

- **7. What is happening when you add (. . . col=df$region) to the above plotting code? In other words, what happens when you run: plot(x=df$Year,y=df$Land.Value,col=df$region)**

Now, for the rest of the assignment. . .

1. Create a new R script as part of your Assignment 4 R-project. Name it "Assignment_4.R"
2. That script should do the following:
    - Read in the file: "/Data/ITS_mapping.csv" . . . good luck with that, hahaha!
    - Somehow summarize all of the columns and do a bit of additional exploration (play with some functions)
    - Make a boxplot where "Ecosystem"" is on the x-axis and "Lat" is on the y-axis
    - Write code to export this boxplot to a new file in your Assignment_4 directory called "silly_boxplot.png" Hints on below . . .
3. Make sure to save your completed script and Rproject and make sure your png file is saved correctly
4. Push all these saved changes and new files onto your GitHub repository so I can grade them
5. Don't forget the plaintext file with answers to bolded questions needs to go to Canvas as well!

To use the base R method to save a plot to an image file, you just wrap your code for the image between two commands.

```r
png(filename = "./silly_boxplot.png")
#whatevercodeyoucameupwithforyourplot
dev.off()
```

The png() function has lots of options you can tweak, but it opens a “graphics device” that starts collecting any output from R into a .png image file.

The dev.off() function just closes the graphics device and writes the previous input to the file you specified.