Homework 2

Due September 12, 2019

Eric Bae

2019-09-11

For each assignment, turn in by the due date/time. Late assignments must be arranged prior to submission. In every case, assignments are to be typed neatly using proper English in Markdown.

This week, we spoke about Reproducible Research, R and version control, getting, cleaning and munging data and finally, summarizing data. Again, we are focusing on Reproducible Analysis which, for us, is accomplished by mixing code, figures and text into a cohesive document that fully describes both the process we took to go from data to results and the rational behind our data driven conclusions. This week we begin creating tidy data sets. While others have proposed standards for sharing data with statiticians, as practicing data scientists, we realize the often onerous task of getting, cleaning and formatting data is usually in our hands. From here on out, we will use GitHub to retrieve and turn in the homework assignments.

Problem 1

Work through the "R Programming E" lesson parts 4-7, 14 (optional 12 - only takes 5 min).

From the R command prompt:

```
install.packages("swirl")
library(swirl)
install_course("R_Programming_E")
swirl()
```

Problem 2

Create a new R Markdown file within your local GitHub repo folder (file–>new–>R Markdown–>save as).

The filename should be: HW2_lastname, i.e. for me it would be HW2_Settlage

You will use this new R Markdown file to solve problems 3-5.

Problem 3

In the lecture, there were two links to StackOverflow questions on why one should use version control. In your own words, summarize in 2-3 sentences how you think version control can help you in the classroom.

I think version control in this classroom is a good idea because I am prone to causing irreversible mistakes to my codes, and since we will continue to upload a large number of files to GitHub, which could lead to me losing track of some files. I have had situations where I accidentally overwrote a file against my intention. In fact, I already did with HW1, though that one was semi-intentional.

Problem 4

In this exercise, you will import, munge, clean and summarize datasets from Wu and Hamada's *Experiments: Planning, Design and Analysis* book you will use in the Spring. For each one, please weave your code and text to describe both your process and observations. Make sure you create a tidy dataset describing the variables, create a summary table of the data, note issues with the data.

a. Sensory data from five operators.
   http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat
b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0.
   http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat

c. Brain weight (g) and body weight (kg) for 62 species.
   http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat

d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities.
   http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat

```r
# Sensory Table
sensory.url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"

# Importing data as dataframe
sensory <- as.data.frame(fread(sensory.url, fill = TRUE, skip = 2))
N <- nrow(sensory) # Number of rows = 30
D <- ncol(sensory) - 1 # Number of objectives/variables = 5

# Rearranging data so that NA goes to the first row
for (i in 1:N) {
  if (is.na(sensory$V6[i])) {
    sensory[i,-1] <- sensory[i,1:D]
    sensory[i,1] <- NA
  }
}

# Total number of items = 10
I <- length(sensory$V1[which(is.na(sensory$V1) == FALSE)])

# Reorganizes item
sensory$V1 <- rep(1:I, each = 3)

# Tidying up, one column by column
Item <- sort(rep(sensory$V1, each = D))
Observation <- rep(rep(1:3, each = 5), I)
Operator <- rep(rep(1:D), 3*I)
Dat <- c(t(sensory[,-1]))

# Combining the columns
sensory <- as_tibble(cbind(Item, Observation, Operator, Dat))

# Summary statistics
summary(sensory)
```

```
##       Item       Observation    Operator      Dat
##  Min.   : 1.0   Min.   :1    Min.   :1   Min.   :0.700
##  1st Qu.: 3.0   1st Qu.:1    1st Qu.:2   1st Qu.:3.025
##  Median : 5.5   Median :2    Median :3   Median :4.700
##  Mean   : 5.5   Mean   :2    Mean   :3   Mean   :4.657
##  3rd Qu.: 8.0   3rd Qu.:3    3rd Qu.:4   3rd Qu.:6.000
##  Max.   :10.0   Max.   :3    Max.   :5   Max.   :9.400
```

```r
sensory %>%
  group_by(Operator) %>%
  summarize(Mean = mean(Dat), SD = sd(Dat), Min = min(Dat), Max = max(Dat))
```

```
## # A tibble: 5 x 5
##   Operator  Mean    SD   Min   Max
##      <dbl> <dbl> <dbl> <dbl> <dbl>
## 1       1.  4.59  2.24 0.900  9.00
## 2       2.  5.06  2.05 1.50   9.20
## 3       3.  4.17  2.10 0.800  9.00
## 4       4.  5.19  2.13 0.900  9.40
## 5       5.  4.27  2.14 0.700  8.80
```

```
sensory %>%
  group_by(Item) %>%
  summarize(Mean = mean(Dat), SD = sd(Dat), Min = min(Dat), Max = max(Dat))
```

```
## # A tibble: 10 x 5
##     Item  Mean    SD   Min   Max
##    <dbl> <dbl> <dbl> <dbl> <dbl>
##  1    1.  4.47 0.779 3.30   5.70
##  2    2.  5.31 0.713 4.20   6.30
##  3    3.  2.77 0.841 1.30   4.60
##  4    4.  6.88 0.665 5.90   8.20
##  5    5.  5.92 0.500 4.90   7.00
##  6    6.  2.39 0.818 1.10   4.00
##  7    7.  1.41 0.642 0.700  3.10
##  8    8.  4.43 0.574 3.00   4.90
##  9    9.  8.47 0.761 6.70   9.40
## 10   10.  4.52 0.825 2.80   5.50
```

```
#sensory %>%
#  group_by(Item, Operator) %>%
#  summarize(Mean = mean(Dat), SD = sd(Dat), Min = min(Dat), Max = max(Dat))
```

The summary statistics shows that the all-time minimum sensory measurement (no idea what the unit is) is 0.700, the maximum is 9.400, and the mean is 4.657.
Grouping by operator, operator 3 had the minimum average measurement with the value of 4.17, and operator 4 had the maximum average measurement with the value of 5.19.
Grouping by item, item 7 had the minimum mean measurement at 1.41 while item 9 had the highest at 8.47.

```
# Kable
kable(sensory, format = "markdown", caption = "Sensory data, reformated")
```

Item	Observation	Operator	Dat
1	1	1	4.3
1	1	2	4.9
1	1	3	3.3
1	1	4	5.3
1	1	5	4.4
1	2	1	4.3
1	2	2	4.5
1	2	3	4.0
1	2	4	5.5
1	2	5	3.3
1	3	1	4.1

Item	Observation	Operator	Dat
1	3	2	5.3
1	3	3	3.4
1	3	4	5.7
1	3	5	4.7
2	1	1	6.0
2	1	2	5.3
2	1	3	4.5
2	1	4	5.9
2	1	5	4.7
2	2	1	4.9
2	2	2	6.3
2	2	3	4.2
2	2	4	5.5
2	2	5	4.9
2	3	1	6.0
2	3	2	5.9
2	3	3	4.7
2	3	4	6.3
2	3	5	4.6
3	1	1	2.4
3	1	2	2.5
3	1	3	2.3
3	1	4	3.1
3	1	5	2.4
3	2	1	3.9
3	2	2	3.0
3	2	3	2.8
3	2	4	2.7
3	2	5	1.3
3	3	1	1.9
3	3	2	3.9
3	3	3	2.6
3	3	4	4.6
3	3	5	2.2
4	1	1	7.4
4	1	2	8.2
4	1	3	6.4
4	1	4	6.8
4	1	5	6.0
4	2	1	7.1
4	2	2	7.9
4	2	3	5.9
4	2	4	7.3
4	2	5	6.1
4	3	1	6.4
4	3	2	7.1
4	3	3	6.9
4	3	4	7.0
4	3	5	6.7
5	1	1	5.7
5	1	2	6.3
5	1	3	5.4

Item	Observation	Operator	Dat
5	1	4	6.1
5	1	5	5.9
5	2	1	5.8
5	2	2	5.7
5	2	3	5.4
5	2	4	6.2
5	2	5	6.5
5	3	1	5.8
5	3	2	6.0
5	3	3	6.1
5	3	4	7.0
5	3	5	4.9
6	1	1	2.2
6	1	2	2.4
6	1	3	1.7
6	1	4	3.4
6	1	5	1.7
6	2	1	3.0
6	2	2	1.8
6	2	3	2.1
6	2	4	4.0
6	2	5	1.7
6	3	1	2.1
6	3	2	3.3
6	3	3	1.1
6	3	4	3.3
6	3	5	2.1
7	1	1	1.2
7	1	2	1.5
7	1	3	1.2
7	1	4	0.9
7	1	5	0.7
7	2	1	1.3
7	2	2	2.4
7	2	3	0.8
7	2	4	1.2
7	2	5	1.3
7	3	1	0.9
7	3	2	3.1
7	3	3	1.1
7	3	4	1.9
7	3	5	1.6
8	1	1	4.2
8	1	2	4.8
8	1	3	4.5
8	1	4	4.6
8	1	5	3.2
8	2	1	3.0
8	2	2	4.5
8	2	3	4.7
8	2	4	4.9
8	2	5	4.6

Item	Observation	Operator	Dat
8	3	1	4.8
8	3	2	4.8
8	3	3	4.7
8	3	4	4.8
8	3	5	4.3
9	1	1	8.0
9	1	2	8.6
9	1	3	9.0
9	1	4	9.4
9	1	5	8.8
9	2	1	9.0
9	2	2	7.7
9	2	3	6.7
9	2	4	9.0
9	2	5	7.9
9	3	1	8.9
9	3	2	9.2
9	3	3	8.1
9	3	4	9.1
9	3	5	7.6
10	1	1	5.0
10	1	2	4.8
10	1	3	3.9
10	1	4	5.5
10	1	5	3.8
10	2	1	5.4
10	2	2	5.0
10	2	3	3.4
10	2	4	4.9
10	2	5	4.6
10	3	1	2.8
10	3	2	5.2
10	3	3	4.1
10	3	4	3.9
10	3	5	5.5

The above is the tidy version of the data set. There is obviously a lot of issues with this data set. Starting with the fact that it is extremely long, it is hard to navigate and confusing to understand.
I also had to use a lot of hard-coding, which is not ideal.

```
# Gold Medals Table
medals.url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
medals <- as.data.frame(fread(medals.url, fill = TRUE, skip = 1))

# Assign names
colnames(medals) <- rep(c("Year", "Long Jump"), 4)

# Tibble
medals <- as_tibble(rbind(medals[,1:2], medals[,3:4], medals[,5:6], medals[,7:8]))

# Summary statistics
```

```r
summary(medals)
```

```
##       Year          Long Jump    
##  Min.   :-4.00   Min.   :249.8  
##  1st Qu.:21.00   1st Qu.:295.4  
##  Median :50.00   Median :308.1  
##  Mean   :45.45   Mean   :310.3  
##  3rd Qu.:71.00   3rd Qu.:327.5  
##  Max.   :92.00   Max.   :350.5  
##  NA's   :2       NA's   :2      
```

```r
medals <- medals %>%
  mutate(Rank = round(rank(-`Long Jump`), 0)) # ADded rankings

# Kable
kable(medals, format = "markdown", caption = "Gold Medals data, reformatted")
```

Year	Long Jump	Rank
-4	249.75	22
0	282.88	20
4	289.00	19
8	294.50	17
12	299.25	15
20	281.50	21
24	293.13	18
28	304.75	13
32	300.75	14
36	317.31	10
48	308.00	12
52	298.00	16
56	308.25	11
60	319.75	8
64	317.75	9
68	350.50	1
72	324.50	7
76	328.50	6
80	336.25	4
84	336.25	4
88	343.25	2
92	342.50	3
NA	NA	23
NA	NA	24

The above table is the table of the record long jump of the male gold medalist by olympic year. The index year (0) is 1900. I also added the rankings as the third column.
The minimum record was 249.75, recorded at the 1896 Olympics and the maximum record was 350.50, recorded at the 1968 Olympics. The mean record across the years was 310.3.
This one went a little bit better than the Sensory data set, though some hard coding was used, especially for rbind().

```r
# Brain and Body Weight Table
brain.url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
```

```r
brain <- as.data.frame(fread(brain.url, fill = TRUE, skip = 1))

# Assign names
colnames(brain) <- rep(c("Body Wt", "Brain Wt"), 3)

# As tibble
brain <- as_tibble(rbind(brain[,1:2], brain[,3:4], brain[,5:6]))

# Add brain weight/body weight ratio
brain <- brain %>%
  arrange(`Body Wt`) %>%
  mutate(`Brain-to-Body Ratio` = `Brain Wt`/`Body Wt`)

# Summary statistics
summary(brain)
```

```
##     Body Wt            Brain Wt       Brain-to-Body Ratio
##  Min.   :   0.005   Min.   :   0.10   Min.   : 0.8584
##  1st Qu.:   0.600   1st Qu.:   4.25   1st Qu.: 3.1026
##  Median :   3.342   Median :  17.25   Median : 6.6109
##  Mean   : 198.790   Mean   : 283.13   Mean   : 9.5758
##  3rd Qu.:  48.203   3rd Qu.: 166.00   3rd Qu.:13.6684
##  Max.   :6654.000   Max.   :5712.00   Max.   :39.6040
##  NA's   :1          NA's   :1         NA's   :1
```

```r
# Kable
kable(brain, format = "markdown", caption = "Brain and Body Weight data, reformated")
```

Body Wt	Brain Wt	Brain-to-Body Ratio
0.005	0.10	20.0000000
0.010	0.30	30.0000000
0.023	0.30	13.0434783
0.023	0.40	17.3913043
0.048	0.33	6.8750000
0.060	1.00	16.6666667
0.075	1.20	16.0000000
0.101	4.00	39.6039604
0.104	2.50	24.0384615
0.120	1.00	8.3333333
0.122	3.00	24.5901639
0.200	5.00	25.0000000
0.280	1.90	6.7857143
0.425	6.40	15.0588235
0.480	15.50	32.2916667
0.550	2.40	4.3636364
0.750	12.30	16.4000000
0.785	3.50	4.4585987
0.900	2.60	2.8888889
0.920	5.70	6.1956522
1.000	6.60	6.6000000
1.040	5.50	5.2884615
1.350	8.10	6.0000000
1.400	12.50	8.9285714
1.410	17.50	12.4113475

Body Wt	Brain Wt	Brain-to-Body Ratio
1.620	11.40	7.0370370
1.700	6.30	3.7058824
2.000	12.30	6.1500000
2.500	12.10	4.8400000
3.000	25.00	8.3333333
3.300	25.60	7.7575758
3.385	44.50	13.1462334
3.500	10.80	3.0857143
3.500	3.90	1.1142857
3.600	21.00	5.8333333
4.050	17.00	4.1975309
4.190	58.00	13.8424821
4.235	50.40	11.9008264
4.288	39.20	9.1417910
6.800	179.00	26.3235294
10.000	115.00	11.5000000
10.550	179.50	17.0142180
14.830	98.20	6.6217127
27.660	115.00	4.1576283
35.000	56.00	1.6000000
36.330	119.50	3.2892926
52.160	440.00	8.4355828
55.500	175.00	3.1531532
60.000	81.00	1.3500000
62.000	1320.00	21.2903226
85.000	325.00	3.8235294
100.000	157.00	1.5700000
160.000	169.00	1.0562500
187.100	419.00	2.2394441
192.000	180.00	0.9375000
207.000	406.00	1.9613527
250.000	490.00	1.9600000
465.000	423.00	0.9096774
521.000	655.00	1.2571977
529.000	680.00	1.2854442
2547.000	4603.00	1.8072242
6654.000	5712.00	0.8584310
NA	NA	NA

The average weight of all bodies observed was 198.790 kg whereas the average weight of all brains was 283.13 g. However, the median weight of bodies and brains were 3.342 kg and 17.25 g, respectively, suggesting that both weight were skewed heavily to the right. The weights ranged from 0.005 kg to 6,654.000 kg for body and 0.10 g to 5,712.00 g for brain.

I added the brain-to-body weight ratio (in 1000s) as the third column just for reference. Though I did not perform statistical analysis, simple eyeballing it seems to indicate that the larger the body weight the lower the ratio.

The issue here was more similar to the issue I faced with the medals dataset in that there was some hard coding with binding the three pairs of columns into one.

```r
# Tomato Yield Table
tomato.url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
```

```r
tomato <- as.data.frame(fread(tomato.url, fill = TRUE, skip = 1))

# Assign names
colnames(tomato) <- as.character(tomato[1, c(4, 1:3)])
colnames(tomato)[1] <- "Tomato Variety"
tomato <- tomato[-1,]
tomato <- tomato[rep(1:2, each = 3),]

new.tomato <- tomato
for (i in seq(1, nrow(tomato), by = 3)) {
  for (j in 2:ncol(tomato)) {
    new.tomato[seq(i, i+2),j] <- strsplit(as.character(tomato[i,j]), ",")[[1]]
  }
}

# Tidying up, one column by column
Tomato <- c(t(rep(new.tomato$`Tomato Variety`, each = ncol(new.tomato) - 1)))
Density <- rep(colnames(new.tomato)[-1], length(unique(Tomato)))
Yield <- c(t(new.tomato[,-1]))
tomato <- as_tibble(cbind(Tomato, Density, Yield))

# Kable
kable(tomato, format = "markdown", caption = "Tomato yield by variety, reformatted")
```

Tomato	Density	Yield
Ife#1	10000	16.1
Ife#1	20000	16.6
Ife#1	30000	20.8
Ife#1	10000	15.3
Ife#1	20000	19.2
Ife#1	30000	18.0
Ife#1	10000	17.5
Ife#1	20000	18.5
Ife#1	30000	21.0
PusaEarlyDwarf	10000	8.1
PusaEarlyDwarf	20000	12.7
PusaEarlyDwarf	30000	14.4
PusaEarlyDwarf	10000	8.6
PusaEarlyDwarf	20000	13.7
PusaEarlyDwarf	30000	15.4
PusaEarlyDwarf	10000	10.1
PusaEarlyDwarf	20000	11.5
PusaEarlyDwarf	30000	13.7

The first column represents the tomato variety, the second the planting density, and the third the yield. The lowest yield was found in the tomato variety called "PusaEarlyDwarf" and planting density of 10,000, while the highest yield was found in "Ife#1" and planting density of 30,000.

I had a lot of trouble with this data set because the yield data was not numeric but actually some sort of characters, delineated by commas. I had to find a way to split them up by the commas, but unable to change the classes of the yields without causing the numbers to change. This is why summary table is not available here because I could not find a way to generate summary statistics for character outputs.

Problem 5

In the swirl lessons, you played with a dataset "plants". Our ultimate goal is to see if there is a relationship between pH and Foliage_Color. Consider a statistic that combines the information in pH_Min and pH_Max. Clean, summarize and transform the data as appropriate. Use function *lm* to test for a relationship. Report both the coefficients and ANOVA results in table form.

Note that if you didn't just do the swirl lesson, it is now not available. Add the following code to your project to retrieve it.

```r
# Path to data
library(swirl)
```

```
##
## | Hi! I see that you have some variables saved in your workspace. To keep
## | things running smoothly, I recommend you clean up before starting swirl.
##
## | Type ls() to see a list of the variables in your workspace. Then, type
## | rm(list=ls()) to clear your workspace.
##
## | Type swirl() when you are ready to begin.
```

```r
.datapath <- file.path(path.package('swirl'), 'Courses',
                       'R_Programming_E', 'Looking_at_Data',
                       'plant-data.txt')
# Read in data
plants <- read.csv(.datapath, strip.white=TRUE, na.strings="")
# Remove annoying columns
.cols2rm <- c('Accepted.Symbol', 'Synonym.Symbol')
plants <- plants[, !(names(plants) %in% .cols2rm)]
# Make names pretty
names(plants) <- c('Scientific_Name', 'Duration', 'Active_Growth_Period',
                   'Foliage_Color', 'pH_Min', 'pH_Max',
                   'Precip_Min', 'Precip_Max',
                   'Shade_Tolerance', 'Temp_Min_F')
```

```r
attach(plants)
plants.full <- plants[-which(is.na(pH_Min)|is.na(Foliage_Color)|is.na(pH_Max)),]
plants.full$pH_Avg <- rowMeans(cbind(plants.full$pH_Max, plants.full$pH_Min))
summary(plants.full$Foliage_Color)
```

```
##   Dark Green   Gray-Green        Green          Red   White-Gray
##           82           25          692            4            9
## Yellow-Green
##           20
```

```r
summary(plants.full$pH_Avg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.30    5.80    6.15    6.17    6.50    8.20
```

```r
summary(lm(pH_Avg ~ Foliage_Color, data = plants.full))
```

```
##
## Call:
## lm(formula = pH_Avg ~ Foliage_Color, data = plants.full)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.63750 -0.33410  0.00061  0.31590  2.01590
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.99939    0.05951 100.810  < 2e-16 ***
## Foliage_ColorGray-Green     0.41261    0.12312   3.351 0.000841 ***
## Foliage_ColorGreen          0.18471    0.06294   2.935 0.003430 **
## Foliage_ColorRed            0.16311    0.27594   0.591 0.554617
## Foliage_ColorWhite-Gray     0.44505    0.18924   2.352 0.018914 *
## Foliage_ColorYellow-Green -0.06189    0.13440  -0.461 0.645275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5389 on 826 degrees of freedom
## Multiple R-squared:  0.0234, Adjusted R-squared:  0.01749
## F-statistic: 3.958 on 5 and 826 DF,  p-value: 0.00149
```

```
summary(aov(pH_Avg ~ Foliage_Color, data = plants.full))
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Foliage_Color   5   5.75  1.1495   3.958 0.00149 **
## Residuals     826 239.88  0.2904
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The linear model above compares the foliage color of plants in question to the average of the maximum and minimum pH. This is obviously problematic because it does not take into consideration species type and any other environmental factors. However, for the purpose of this exercise, I decided to do so anyway because why not. Also, all observations with missing values in any of the three variables used were eliminated.
The foliage color is a nominal, factored variable whereas the pH is a numeric, continuous variable. The foliage color "dark green" was the index in our linear regression, and there were five other colors - Gray-green, green, red, white-gray, and yellow-green. Based on the summary, gray-green, green, and white-gray gave us with $p - value < 0.05$, which I will determine to be our alpha. All of them had positive slope coefficients. This means that the three colors - gray-green, green, and white-gray were associated with higher average pH of soil than the index - dark green - by approximately 0.4126, 0.1847, and 0.4451, respectively.
Based on the ANOVA, the p-value was 0.00149, which was, again, lower than 0.05. This indicates that there is a significant relationship between foliage color and the average pH.

Problem 6

Finish this homework by pushing your changes to your repo. In general, your workflow for this should be:

1. git pull – to make sure you have the most recent repo

2. In R: do some work

3. git add – this tells git to track new files

4. git commit – make message INFORMATIVE and USEFUL

5. git push – this pushes your local changes to the repo

If you have difficulty with steps 1-5, git is not correctly or completely setup. See me for help.

Only submit the .Rmd and .pdf solution files. Names should be formatted HW2_lastname.Rmd and HW2_lastname.pdf

Optional preperation for next class:

TBD