Homework 2

Han Chen

2019-09-09

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:

```r
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.

Answer To keep track every steps and possible branches of my project, so I can review the chronicle of the project and make it a clear and reproduciable research work.

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

1. Overview the dataset from the url. The first 2 rows have different lengths with other rows.

```
u1 <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
readLines(u1, n =40)
```

2. Import dataset and seperate item number from sensory data. Combine the complete item number column with sensory data.

```
d1 <- scan(u1, skip = 2)
item.no <- seq(from = 1, to = 145, by = 16)
d1 <- d1[-item.no]
d1 <- matrix(d1, ncol = 5, byrow = T)
item.col <- as.factor(rep(1:10, each = 3))
d1 <- data.frame(item.col, d1)
```

3. Name the column (variables) and display. The table below gives 6 columns. The 1st column gives the item number which operators sense, and the last 5 columns give the sensory data on the item by each operator.

```
colnames(d1) <- c("item", paste(rep("operate", times = 5), 1:5, sep=" "))
knitr::kable(d1)
```

item	operate 1	operate 2	operate 3	operate 4	operate 5
1	4.3	4.9	3.3	5.3	4.4
1	4.3	4.5	4.0	5.5	3.3
1	4.1	5.3	3.4	5.7	4.7
2	6.0	5.3	4.5	5.9	4.7
2	4.9	6.3	4.2	5.5	4.9
2	6.0	5.9	4.7	6.3	4.6
3	2.4	2.5	2.3	3.1	2.4
3	3.9	3.0	2.8	2.7	1.3
3	1.9	3.9	2.6	4.6	2.2
4	7.4	8.2	6.4	6.8	6.0
4	7.1	7.9	5.9	7.3	6.1
4	6.4	7.1	6.9	7.0	6.7
5	5.7	6.3	5.4	6.1	5.9
5	5.8	5.7	5.4	6.2	6.5
5	5.8	6.0	6.1	7.0	4.9
6	2.2	2.4	1.7	3.4	1.7
6	3.0	1.8	2.1	4.0	1.7
6	2.1	3.3	1.1	3.3	2.1
7	1.2	1.5	1.2	0.9	0.7
7	1.3	2.4	0.8	1.2	1.3
7	0.9	3.1	1.1	1.9	1.6
8	4.2	4.8	4.5	4.6	3.2
8	3.0	4.5	4.7	4.9	4.6
8	4.8	4.8	4.7	4.8	4.3
9	8.0	8.6	9.0	9.4	8.8
9	9.0	7.7	6.7	9.0	7.9
9	8.9	9.2	8.1	9.1	7.6
10	5.0	4.8	3.9	5.5	3.8
10	5.4	5.0	3.4	4.9	4.6

item	operate 1	operate 2	operate 3	operate 4	operate 5
10	2.8	5.2	4.1	3.9	5.5

4. Summary statistics of each operator.

```
knitr::kable(summary(d1[, 2:6]))
```

operate 1	operate 2	operate 3	operate 4	operate 5
Min. :0.900	Min. :1.500	Min. :0.800	Min. :0.900	Min. :0.700
1st Qu.:2.850	1st Qu.:3.450	1st Qu.:2.650	1st Qu.:3.925	1st Qu.:2.250
Median :4.550	Median :4.950	Median :4.150	Median :5.400	Median :4.600
Mean :4.593	Mean :5.063	Mean :4.167	Mean :5.193	Mean :4.267
3rd Qu.:5.950	3rd Qu.:6.225	3rd Qu.:5.400	3rd Qu.:6.275	3rd Qu.:5.800
Max. :9.000	Max. :9.200	Max. :9.000	Max. :9.400	Max. :8.800

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

1. Overview the dataset. The 1st row which is the variable name has two space between two names and the last 2 rows have difference length than other rows. These are issues needed to be addressed.

```
u2 <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
readLines(u2)
```

2. Read variables names and data value seperately and then combine them. Sort the data by year.

```
d2.names <- readLines(u2, n=1)
d2.names <- strsplit(d2.names, split = " ")[[1]]
d2.names <- c(d2.names[1],paste(d2.names[2], d2.names[3]))

d2 <- scan(u2, skip = 1, what = "double")
d2 <- data.frame(matrix(as.numeric(d2), byrow = T, ncol = 2))
colnames(d2) <- d2.names
d2 <- d2[order(d2$Year),]
```

3. Display the data and summary statistics.

```
knitr::kable(d2)
```

	Year	Long Jump
1	-4	249.75
5	0	282.88
9	4	289.00
13	8	294.50
17	12	299.25
20	20	281.50
2	24	293.13
6	28	304.75
10	32	300.75
14	36	317.31
18	48	308.00
21	52	298.00
3	56	308.25

	Year	Long Jump
7	60	319.75
11	64	317.75
15	68	350.50
19	72	324.50
22	76	328.50
4	80	336.25
8	84	336.25
12	88	343.25
16	92	342.50

```
knitr::kable(summary(d2)[,2])
```

x
Min. :249.8
1st Qu.:295.4
Median :308.1
Mean :310.3
3rd Qu.:327.5
Max. :350.5

c. Brain weight (g) and body weight (kg) for 62 species.

http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat

1. Overview of the dataset.

```
u3 <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
readLines(u3, n =3)
```

2. Input dataset. Seperate names and value from dataset and set them into right types.

```
d3 <- readLines(u3)
d3.names <- d3[1]
d3 <- d3[2:22]

d3.names <- unique(strsplit(d3.names, split = " ")[[1]])
d3.names <- c(paste(d3.names[1], d3.names[2]), paste(d3.names[3], d3.names[2]))

d3 <- as.numeric(unlist(strsplit(d3, split = " ")))
d3 <- matrix(d3, byrow = T, ncol = 2)
colnames(d3) <- d3.names
```

3. Display the data and summry statistics

```
knitr::kable(d3)
```

Body Wt	Brain Wt
3.385	44.50
521.000	655.00
2.500	12.10
0.480	15.50
0.785	3.50

Body Wt	Brain Wt
55.500	175.00
1.350	8.10
10.000	115.00
100.000	157.00
465.000	423.00
3.300	25.60
52.160	440.00
36.330	119.50
0.200	5.00
10.550	179.50
27.660	115.00
1.410	17.50
0.550	2.40
14.830	98.20
529.000	680.00
60.000	81.00
1.040	5.50
207.000	406.00
3.600	21.00
4.190	58.00
85.000	325.00
4.288	39.20
0.425	6.40
0.750	12.30
0.280	1.90
0.101	4.00
62.000	1320.00
0.075	1.20
0.920	5.70
6654.000	5712.00
0.122	3.00
1.000	6.60
3.500	3.90
0.048	0.33
0.005	0.10
6.800	179.00
192.000	180.00
0.060	1.00
35.000	56.00
3.000	25.00
3.500	10.80
4.050	17.00
160.000	169.00
2.000	12.30
0.120	1.00
0.900	2.60
1.700	6.30
0.023	0.40
1.620	11.40
2547.000	4603.00
0.010	0.30
0.104	2.50

Body Wt	Brain Wt
0.023	0.30
1.400	12.50
4.235	50.40
187.100	419.00
250.000	490.00

```r
knitr::kable(summary(d3))
```

Body Wt	Brain Wt
Min. : 0.005	Min. : 0.10
1st Qu.: 0.600	1st Qu.: 4.25
Median : 3.342	Median : 17.25
Mean : 198.790	Mean : 283.13
3rd Qu.: 48.203	3rd Qu.: 166.00
Max. :6654.000	Max. :5712.00

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

1. Overview of the dataset.

```r
u4 <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
readLines(u4, n =5)
```

2. Clean the data

```r
d4 <- readLines(u4)

d4.density <- as.numeric(unlist(strsplit(d4[2],split = "\\s+")))[-1]
d4.density <- rep(d4.density, times = 2, each = 3)

d4.r3 <- unlist(strsplit(d4[3],split = "\\s+"))
d4.r4 <- unlist(strsplit(d4[4],split = "\\s+"))

d4.variety <- c(d4.r3[1], d4.r4[1])
d4.variety <- rep(d4.variety, each = 9)

d4.yield.r3 <- c()
d4.yield.r4 <- c()
for (i in 2:4){
  d4.yield.r3 <- c(d4.yield.r3, unlist(strsplit(d4.r3[i],split = ",")))
  d4.yield.r4 <- c(d4.yield.r4, unlist(strsplit(d4.r4[i],split = ",")))
}

d4.yield <- as.numeric(c(d4.yield.r3, d4.yield.r4))

d4 <- data.frame(d4.variety, d4.density, d4.yield)
colnames(d4) <- c("Variety", "Density", "Yield")
```

3. Display data and summary statistics

```
knitr::kable(d4)
```

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

```
knitr::kable(summary(d4))
```

Variety	Density	Yield
Ife#1 :9	Min. :10000	Min. : 8.10
PusaEarlyDwarf:9	1st Qu.:10000	1st Qu.:12.95
NA	Median :20000	Median :15.35
NA	Mean :20000	Mean :15.07
NA	3rd Qu.:30000	3rd Qu.:17.88
NA	Max. :30000	Max. :21.00

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.

```
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
# Path to data
.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')
```

1. Define a variable ph difference to combine the information in pH_Min and pH_Max. Remove NAs row in foliage color and pH difference from the dataset.

```r
pH_Dif <- plants$pH_Max - plants$pH_Min
plants <- data.frame(plants, pH_Dif)
plants <- plants[is.na(plants$Foliage_Color) == FALSE &
                   is.na(plants$pH_Dif) == FALSE,  ]
```

2. Summary statistics of pH difference of different foliage color groups

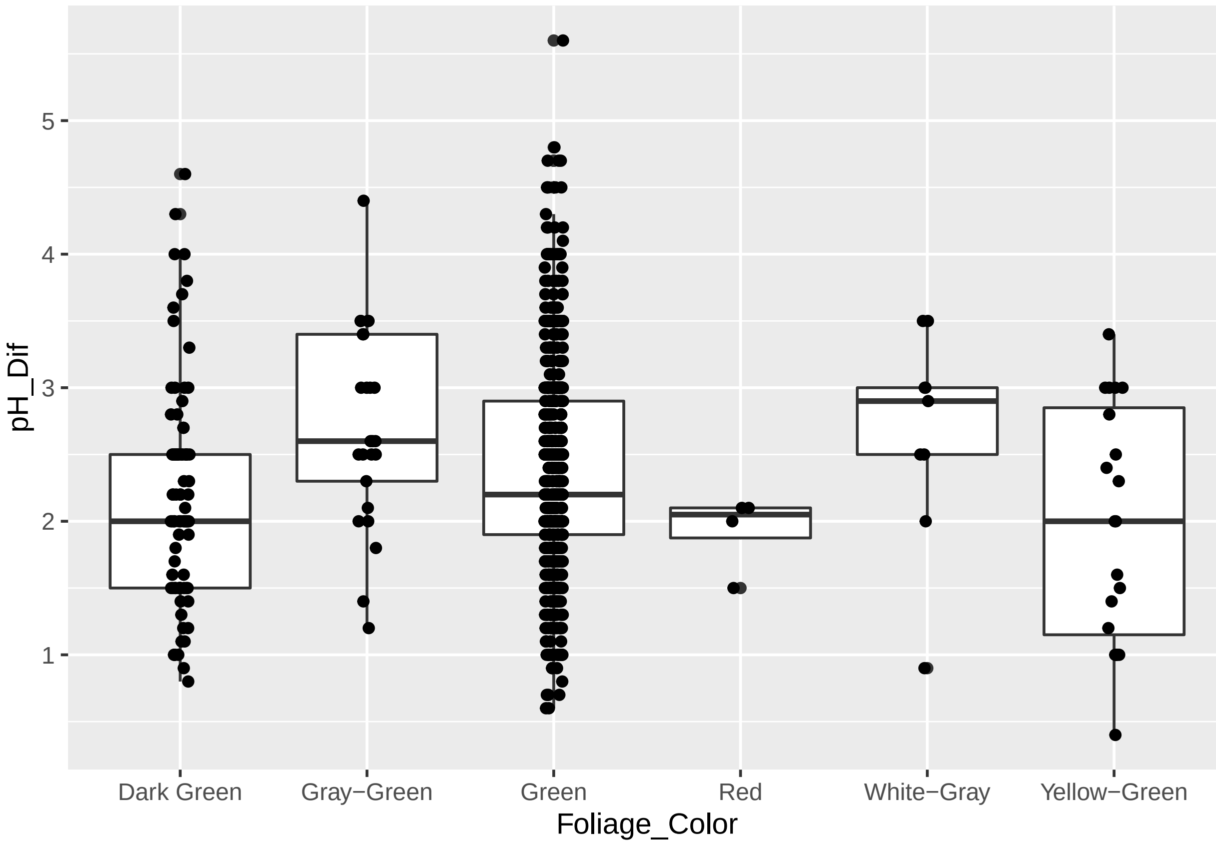
```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
group_by(plants, Foliage_Color) %>%
  summarise(
    count = n(),
    mean = mean(pH_Dif, na.rm = TRUE),
    sd = sd(pH_Dif, na.rm = TRUE)
  )
```

```
## # A tibble: 6 x 4
##   Foliage_Color count  mean    sd
##   <fct>         <int> <dbl> <dbl>
## 1 Dark Green       82  2.18 0.823
## 2 Gray-Green       25  2.71 0.742
## 3 Green           692  2.37 0.781
## 4 Red               4  1.92 0.287
## 5 White-Gray        9  2.64 0.813
## 6 Yellow-Green     20  1.98 0.890
```

```r
library(ggplot2)
plants %>% ggplot(aes(Foliage_Color, pH_Dif)) +
        geom_boxplot() +
        geom_jitter(width = 0.05)
```



3. Use linear model and ANOVA results to check the relationship between foliage color and ph Difference.

```r
lm(pH_Dif ~ Foliage_Color, plants)
```

```
##
## Call:
## lm(formula = pH_Dif ~ Foliage_Color, data = plants)
##
## Coefficients:
##               (Intercept)    Foliage_ColorGray-Green
##                    2.1768                     0.5352
##        Foliage_ColorGreen           Foliage_ColorRed
##                    0.1908                    -0.2518
##   Foliage_ColorWhite-Gray  Foliage_ColorYellow-Green
##                    0.4676                    -0.2018
```

```r
summary(aov(pH_Dif ~ Foliage_Color, plants))
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Foliage_Color   5   10.3   2.053   3.322 0.00561 **
## Residuals     826  510.5   0.618
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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