Statistics 5014: Homework 2

Due Monday September 11, 10 am

Stephen Park

2018-09-02

Problem 4: How would version control help in the classroom?

When writing long script files with complex codes, e.g., LaTeX code lines that need to be knitted to see their output, version control would make it efficient to backtrack to any minor errors that may have been made and correct them. The presence of a version control system would make myself more bold in attempting new coding techniques and syntaxes as well. It will also be more easy to collaborate on projects with fellow classmates without having to be physically in the same work area.

Problem 5: Create tidy datasets from Wu and Hamada (2009)

a. Sensory data from five operators

There are a total of 150 values that each correspond to an Item (1:10) and Operator (1:5). The first issue is that the variables are stored in both rows (Item) and columns (Operator). The second issue is that the column headers are set as the various Operator treatments, not the variable names (i.e., Operator, Item, and Value). In the tidy dataset, each observation would include an Item and Operator, and a value, giving it a dimension of $(150, 3)$. A third issue is that for i $\in \{1, 2, \ldots, 9, 10\}$, every $(3i - 2)$th row has a 6th integer entry on the leftmost side - which should correspond to the Item variable, while the $(3i - 1)$th and $3i$th rows only have 5 data entries each. Therefore, the data is manipulated as follows to obtain the targeted dataset with 150 observations of 3 variables.

```r
urla <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
# Store URL for sensory data
data_mess <- read.table(urla, skip = 1, header = T, fill = T, sep = " ")
# Import data; use 2nd row as header
data_tidy <- data_mess
# Preserve messy data and manipulate using a new variable
for (i in 1:10) {
    data_tidy[3 * i - 1, 2:6] <- data_mess[3 * i - 1, 1:5]
    # For each (3*i-1)th row, 'push' data to the right to align with rest of the data
    data_tidy[3 * i, 2:6] <- data_mess[3 * i, 1:5]  # Do the same for each (3*i)th row
}
colnames(data_tidy) <- c("Item", "1", "2", "3", "4", "5")  # Assign Operator numbers
data_tidy <- data_tidy %>% gather(key = Item, value = Value)  # Gather and drop columns
data_tidy <- cbind(data_tidy[, 1], as.data.frame(rep(1:10, each = 3)), data_tidy[,
    2])  # Insert Item values into the center column
colnames(data_tidy) <- c("Operator", "Item", "Value")  # Rename variable names
data_tidy$Item <- as.factor(data_tidy$Item)  # Change Item variable from numeric to factor
```

```r
head(data_tidy, 5)      # Preview sensory data from five operators
```

```
##   Operator Item Value
## 1        1    1   4.3
## 2        1    1   4.3
## 3        1    1   4.1
## 4        1    2   6.0
## 5        1    2   4.9
```

```r
summary(data_tidy)   # Summarize sensory data
```

```
##  Operator      Item        Value      
##  1:30     1      :15   Min.   :0.700  
##  2:30     2      :15   1st Qu.:3.025  
##  3:30     3      :15   Median :4.700  
##  4:30     4      :15   Mean   :4.657  
##  5:30     5      :15   3rd Qu.:6.000  
##           6      :15   Max.   :9.400  
##           (Other):60
```

```r
str(data_tidy)       # Display sensory data structure
```

```
## 'data.frame':    150 obs. of  3 variables:
##  $ Operator: Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Item    : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ Value   : num  4.3 4.3 4.1 6 4.9 6 2.4 3.9 1.9 7.4 ...
```

b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0.

There are a total of 44 values - 22 Year values (-4:92) and 22 long jump records. The first issue is that both the Year and Record variables are listed in multiple columns. The second issue is that the space (" “) separator creates 12 column names for 8 columns of data, although it appears that the column names were intended to be a repeat of”Year" and “Long Jump.” Therefore, the data is manipulated as follows to obtain the targeted dataset with 22 observations of 2 variables.

```r
urlb <- 'https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat'
     # Store URL for long jump data
datb_mess <- read.table(urlb, fill = T, header = F, sep = " ", skip = 1)
     # Import data; don't use header because space separator is used
datb_tidy <- datb_mess
     # Preserve messy data and manipulate using a new variable
names(datb_tidy) <- rep(c("Year", "Record"), 4)     # Rename columns
datb_tidy <- as.data.frame(lapply(split(as.list(datb_tidy), names(datb_tidy)), unlist))
     # Split according to names, rejoin into a list of 2, then re-combine into a dataset
datb_tidy <- datb_tidy[ 1:22,2:1]     # Rearrange columns and remove NA observations
datb_tidy$Year <- as.ordered(datb_tidy$Year)
     # Change Year variable from numeric to ordered factor
head(datb_tidy, 4)      # Preview long jump data
```

```
##   Year Record
## 1   -4 249.75
## 2    0 282.88
## 3    4 289.00
## 4    8 294.50
```

```
summary(datb_tidy)   # Summarize long jump data
```

```
##       Year        Record     
##  -4     : 1   Min.   :249.8  
##  0      : 1   1st Qu.:295.4  
##  4      : 1   Median :308.1  
##  8      : 1   Mean   :310.3  
##  12     : 1   3rd Qu.:327.5  
##  20     : 1   Max.   :350.5  
##  (Other):16
```

```
str(datb_tidy)       # Display long jump data structure
```

```
## 'data.frame':    22 obs. of  2 variables:
##  $ Year  : Ord.factor w/ 22 levels "-4"<"0"<"4"<"8"<..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Record: num  250 283 289 294 299 ...
```

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

There are a total of 124 values - 62 pairs of brain weights body weights, each corresponding to a mammalian species. The first issue is that both the Brain_g and Body_kg variables are listed in multiple columns. The second issue is that the space (" “) separator creates 12 column names for 6 columns of data, although it appears that the column names were intended to be a repeat of”Brain Wt" and “Body Wt.” Therefore, the data is manipulated as follows to obtain the targeted dataset with 62 observations of 2 variables.

```
urlc <- 'http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat'
     # Store URL for brain and body weight data
datc_mess <- read.table(urlc, fill = T, header = F, sep = " ", skip = 1)
     # Import data; don't use header because space separator is used
datc_tidy <- datc_mess
     # Preserve messy data and manipulate using a new variable
names(datc_tidy) <- rep(c("Brain_g", "Body_kg"), 3)     # Rename columns

datc_tidy <- as.data.frame(lapply(split(as.list(datc_tidy), names(datc_tidy)), unlist))
     # Split according to names, rejoin into a list of 2, then re-combine into a dataset
datc_tidy <- datc_tidy[ 1:62,2:1]     # Rearrange columns and remove NA observations
head(datc_tidy, 5)       # Preview brain weight data
```

```
##   Brain_g Body_kg
## 1   3.385    44.5
## 2   0.480    15.5
## 3   1.350     8.1
## 4 465.000   423.0
## 5  36.330   119.5
```

```
summary(datc_tidy)    # Summarize brain weight data
```

```
##     Brain_g            Body_kg       
##  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  
```

```r
str(datc_tidy)        # Display brain weight data structure
```

```
## 'data.frame':    62 obs. of  2 variables:
##  $ Brain_g: num  3.38 0.48 1.35 465 36.33 ...
##  $ Body_kg: num  44.5 15.5 8.1 423 119.5 ...
```

d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities.

There are a total of 18 tomato yield values, each corresponding to a tomato variety (Ife #1 and Pusa Early Dwarf), and planting density (10,000, 20,000, and 30,000 plants/ha).

```r
urld <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
# Store URL for brain and body weight data
datd_mess <- fread(urld, header = F, sep = " ")
# Import data; don't use header
datd_tidy <- datd_mess
# Preserve messy data and manipulate using a new variable
datd_tidy <- datd_tidy %>% separate(V2, c(paste("10000", 1:3, sep = "_")), sep = ",",
    remove = T, extra = "drop") %>% separate(V3, c(paste("20000", 1:3, sep = "_")),
    sep = ",", remove = T, extra = "drop") %>% separate(V4, c(paste("30000", 1:3,
    sep = "_")), sep = ",", remove = T, extra = "drop")
# Separate data in each column at commas
datd_tidy <- datd_tidy %>% select(-V1)
# Drop tomato names
datd_tidy <- datd_tidy %>% gather(key = Density, value = Yield)  # Gather and drop columns
datd_tidy <- as.data.frame(cbind(as.character(c(datd_mess[1, 1], datd_mess[2, 1])),
    as.numeric(rep(c(10000, 20000, 30000), 6)), as.numeric(datd_tidy[, 2])))
# Re-add Variety and Density variable columns
colnames(datd_tidy) <- c("Variety", "Density", "Yield")  # Rename columns
datd_tidy$Yield <- as.numeric(datd_tidy$Yield)
# Change Yield variable from ordered to numeric factor
head(datd_tidy, 9)  # Preview tomato data
```

```
##          Variety Density Yield
## 1         Ife\\#1   10000     8
## 2 PusaEarlyDwarf   20000    16
## 3         Ife\\#1   30000     6
## 4 PusaEarlyDwarf   10000    17
## 5         Ife\\#1   20000    10
## 6 PusaEarlyDwarf   30000     1
## 7         Ife\\#1   10000     9
## 8 PusaEarlyDwarf   20000     3
## 9         Ife\\#1   30000    13
```

```r
summary(datd_tidy)  # Summarize tomato data
```

```
##            Variety    Density      Yield
##  Ife\\#1       :9   10000:6   Min.   : 1.000
##  PusaEarlyDwarf:9   20000:6   1st Qu.: 4.250
##                     30000:6   Median : 8.500
##                               Mean   : 8.722
##                               3rd Qu.:12.750
##                               Max.   :17.000
```

```
str(datd_tidy)  # Display tomato data structure
```

```
## 'data.frame':    18 obs. of  3 variables:
##  $ Variety: Factor w/ 2 levels "Ife\\#1","PusaEarlyDwarf": 1 2 1 2 1 2 1 2 1 2 ...
##  $ Density: Factor w/ 3 levels "10000","20000",..: 1 2 3 1 2 3 1 2 3 1 ...
##  $ Yield  : num  8 16 6 17 10 1 9 3 13 4 ...
```

Problem 6

```
Foliage_Color_ <- plants$Foliage_Color[!is.na(plants$Foliage_Color) &
      !is.na(plants$pH_Min) & !is.na(plants$pH_Max)]
phmin <- plants$pH_Min[!is.na(plants$Foliage_Color) & !is.na(plants$pH_Min) &
      !is.na(plants$pH_Max)]
phmax <- plants$pH_Max[!is.na(plants$Foliage_Color) & !is.na(plants$pH_Min) &
      !is.na(plants$pH_Max)]     # Exclude rows with missing fcolr, phmin, or phmax values
plants_tidy <- cbind.data.frame(phmin, phmax, Foliage_Color_)     # Combine 3 columns
str(plants_tidy)       # Display tidy plant data structure
```

```
## 'data.frame':    832 obs. of  3 variables:
##  $ phmin         : num  4 7 5.9 5 4.5 4.4 4.8 5.8 4.7 4 ...
##  $ phmax         : num  6 8.5 7 7.8 7.3 6.5 7.2 7 7.3 7.3 ...
##  $ Foliage_Color_: Factor w/ 6 levels "Dark Green","Gray-Green",..: 3 3 3 3 3 3 3 6 3 3 ...
```

```
model <- lm(formula = phmin + phmax ~ Foliage_Color_, data = plants_tidy)     # Linear model
kable(summary(model)$coef, digits = 3, format = "pandoc",
      caption = "Linear Model Coefficients for Min pH + Max pH ~ Foliage Color")
```

Table 1: Linear Model Coefficients for Min pH + Max pH ~ Foliage Color

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11.999	0.119	100.810	0.000
Foliage_Color_Gray-Green	0.825	0.246	3.351	0.001
Foliage_Color_Green	0.369	0.126	2.935	0.003
Foliage_Color_Red	0.326	0.552	0.591	0.555
Foliage_Color_White-Gray	0.890	0.378	2.352	0.019
Foliage_Color_Yellow-Green	-0.124	0.269	-0.461	0.645

```
kable(anova(model), digits = 3, format = "pandoc",
      caption = "ANOVA table for Min pH + Max pH ~ Foliage Color")
```

Table 2: ANOVA table for Min pH + Max pH ~ Foliage Color

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Foliage_Color_	5	22.991	4.598	3.958	0.001
Residuals	826	959.530	1.162		

Virginia Tech Honor Pledge:
"I have neither given nor received unauthorized assistance on this assignment."

Problem 8

Finish this homework by pushing your changes to your repo and submitting me a pull request. In general, your workflow for this should be:

1. In R: pull (Git tab, down arrow) – to make sure you have the most recent repo

2. In R: do some work

3. In R: check files you want to commit

4. In R: commit, make message INFORMATIVE and USEFUL

5. In R: push – this pushes your local changes to the repo

6. In Github: submit a pull request – this tells me you are wanting me to pull in your changes to my master repo

If you have difficulty with steps 1-5, git is not correctly or completely setup. The above will pull from your repo, so does not include anything to get from MY repo, ie nothing new will show up.

To get stuff from my repo which you then can push to your repo, modify the above to be:

1. In R: do some work

2. At command prompt: git pull upstream master – to make sure you have the most recent repo
3. In R: check files you want to commit (this MAY include files I added/changed)

4. In R: commit, make message INFORMATIVE and USEFUL

5. In R: push – this pushes OUR local changes to YOUR repo

6. In Github: submit a pull request – this tells me you are wanting me to pull in your changes to my master repo

Only submit the .Rmd and .pdf solution files. Names should be formatted HW2_lastname_firstname.Rmd and HW2_lastname_firstname.pdf

Optional preperation for next class:

Next week we will talk about R logic and good programming practices. If you have time and are interested, please read:

Google's R Style Guide: https://google.github.io/styleguide/Rguide.xml
Hadley Wickam's R Style Guide: http://r-pkgs.had.co.nz/style.html