Homework 2

Adeline Guthrie

September 11, 2019

Problem 3

Version control is helpful in the classroom for collaborative coding projects, allowing students to edit code that belongs to a group, still preserving other versions of the code so changes may be checked by other members of the group before finalized. It also allows the instructor to keep better track of who in each group has been working on the code, possibly to determine if each group member has done their share.

Problem 4

Part a

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

#Read in Data, skip first line with just "Operator", treat the second row as the header,
#and fill empty spaces with "NA"
sensory <-read.table(url, header = T, sep = " ", skip = 1, fill = T)

#Shift data values to the right in the rows missing the item number and add the item number
c <- 0
for( i in 1:30){
  if(i%%3 != 1){
    sensory[i,2:6] <- sensory[i, 1:5]
    sensory[i,1] <- c
  }else{
    c <- c + 1
  }
}

#Give each observation its own row
sensory <- gather(sensory, X1, response, -Item)

#Name columns
colnames(sensory) <- c("Item", "Operator", "Response")

#Remove X's infront of operator numbers
sensory$Operator <- parse_number(sensory$Operator)

#Final Dataset:
sensory
```

```
##     Item Operator Response
## 1      1        1      4.3
## 2      1        1      4.3
```

```
## 3      1        1      4.1
## 4      2        1      6.0
## 5      2        1      4.9
## 6      2        1      6.0
## 7      3        1      2.4
## 8      3        1      3.9
## 9      3        1      1.9
## 10     4        1      7.4
## 11     4        1      7.1
## 12     4        1      6.4
## 13     5        1      5.7
## 14     5        1      5.8
## 15     5        1      5.8
## 16     6        1      2.2
## 17     6        1      3.0
## 18     6        1      2.1
## 19     7        1      1.2
## 20     7        1      1.3
## 21     7        1      0.9
## 22     8        1      4.2
## 23     8        1      3.0
## 24     8        1      4.8
## 25     9        1      8.0
## 26     9        1      9.0
## 27     9        1      8.9
## 28    10        1      5.0
## 29    10        1      5.4
## 30    10        1      2.8
## 31     1        2      4.9
## 32     1        2      4.5
## 33     1        2      5.3
## 34     2        2      5.3
## 35     2        2      6.3
## 36     2        2      5.9
## 37     3        2      2.5
## 38     3        2      3.0
## 39     3        2      3.9
## 40     4        2      8.2
## 41     4        2      7.9
## 42     4        2      7.1
## 43     5        2      6.3
## 44     5        2      5.7
## 45     5        2      6.0
## 46     6        2      2.4
## 47     6        2      1.8
## 48     6        2      3.3
## 49     7        2      1.5
## 50     7        2      2.4
## 51     7        2      3.1
## 52     8        2      4.8
## 53     8        2      4.5
## 54     8        2      4.8
## 55     9        2      8.6
## 56     9        2      7.7
```

```
## 57     9        2      9.2
## 58    10        2      4.8
## 59    10        2      5.0
## 60    10        2      5.2
## 61     1        3      3.3
## 62     1        3      4.0
## 63     1        3      3.4
## 64     2        3      4.5
## 65     2        3      4.2
## 66     2        3      4.7
## 67     3        3      2.3
## 68     3        3      2.8
## 69     3        3      2.6
## 70     4        3      6.4
## 71     4        3      5.9
## 72     4        3      6.9
## 73     5        3      5.4
## 74     5        3      5.4
## 75     5        3      6.1
## 76     6        3      1.7
## 77     6        3      2.1
## 78     6        3      1.1
## 79     7        3      1.2
## 80     7        3      0.8
## 81     7        3      1.1
## 82     8        3      4.5
## 83     8        3      4.7
## 84     8        3      4.7
## 85     9        3      9.0
## 86     9        3      6.7
## 87     9        3      8.1
## 88    10        3      3.9
## 89    10        3      3.4
## 90    10        3      4.1
## 91     1        4      5.3
## 92     1        4      5.5
## 93     1        4      5.7
## 94     2        4      5.9
## 95     2        4      5.5
## 96     2        4      6.3
## 97     3        4      3.1
## 98     3        4      2.7
## 99     3        4      4.6
## 100    4        4      6.8
## 101    4        4      7.3
## 102    4        4      7.0
## 103    5        4      6.1
## 104    5        4      6.2
## 105    5        4      7.0
## 106    6        4      3.4
## 107    6        4      4.0
## 108    6        4      3.3
## 109    7        4      0.9
## 110    7        4      1.2
```

```
## 111    7        4      1.9
## 112    8        4      4.6
## 113    8        4      4.9
## 114    8        4      4.8
## 115    9        4      9.4
## 116    9        4      9.0
## 117    9        4      9.1
## 118   10        4      5.5
## 119   10        4      4.9
## 120   10        4      3.9
## 121    1        5      4.4
## 122    1        5      3.3
## 123    1        5      4.7
## 124    2        5      4.7
## 125    2        5      4.9
## 126    2        5      4.6
## 127    3        5      2.4
## 128    3        5      1.3
## 129    3        5      2.2
## 130    4        5      6.0
## 131    4        5      6.1
## 132    4        5      6.7
## 133    5        5      5.9
## 134    5        5      6.5
## 135    5        5      4.9
## 136    6        5      1.7
## 137    6        5      1.7
## 138    6        5      2.1
## 139    7        5      0.7
## 140    7        5      1.3
## 141    7        5      1.6
## 142    8        5      3.2
## 143    8        5      4.6
## 144    8        5      4.3
## 145    9        5      8.8
## 146    9        5      7.9
## 147    9        5      7.6
## 148   10        5      3.8
## 149   10        5      4.6
## 150   10        5      5.5
```

```
#Summarize data
sensory %>%
  group_by(Operator) %>%
  summarise(Mean = mean(Response), Std_dev = sd(Response), Min = min(Response),
            Median = median(Response), Max = max(Response))
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
## # A tibble: 5 x 6
##   Operator  Mean Std_dev   Min Median   Max
##      <dbl> <dbl>   <dbl> <dbl>  <dbl> <dbl>
## 1        1  4.59    2.24   0.9   4.55   9
```

```
## 2        2  5.06    2.05   1.5   4.95   9.2
## 3        3  4.17    2.10   0.8   4.15   9
## 4        4  5.19    2.13   0.9   5.4    9.4
## 5        5  4.27    2.14   0.7   4.6    8.8
```

Part b

```
#Store URL
url2 <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"

#Read in data, skip the first row since the labels are messy, fill empty spaces with "NA"
jumps <- read.table(url2, header = F, sep = " ", skip = 1, fill = T)

#Resize the original matrix to give each observation its own row
jumps <- resize(jumps, nrow = 24, ncol =2, byrow = T)

#Remove rows of "NA"
jumps <-na.omit(jumps)

#Name columns
colnames (jumps) <- c("Year", "Long_Jump")

#Convert to tbl_class to use dplyr functions
jumps <- tbl_df(jumps)

#Sort by Year
jumps <- arrange(jumps, Year)

#Final Dataset:
jumps
```

```
## # A tibble: 22 x 2
##     Year Long_Jump
##    <dbl>     <dbl>
##  1    -4      250.
##  2     0      283.
##  3     4      289
##  4     8      294.
##  5    12      299.
##  6    20      282.
##  7    24      293.
##  8    28      305.
##  9    32      301.
## 10    36      317.
## # ... with 12 more rows
```

```
#Summarize data
summary(jumps$Long_Jump)
```

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

Part c

```
#Store URL
url3 <-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"

#Read in data, skip the first row since the labels are messy, fill empty spaces with "NA"
weights <-read.table(url3, header = F, sep = " ", skip = 1, fill = T)

#Resize the original matrix to give each observation its own row
weights <- resize(weights, nrow = 63, ncol =2, byrow = T)

#Name columns
colnames(weights) <- c("Body_Wt", "Brain_Wt")

#Final Dataset:
weights
```

```
##        Body_Wt Brain_Wt
##  [1,]    3.385    44.50
##  [2,]  521.000   655.00
##  [3,]    2.500    12.10
##  [4,]    0.480    15.50
##  [5,]    0.785     3.50
##  [6,]   55.500   175.00
##  [7,]    1.350     8.10
##  [8,]   10.000   115.00
##  [9,]  100.000   157.00
## [10,]  465.000   423.00
## [11,]    3.300    25.60
## [12,]   52.160   440.00
## [13,]   36.330   119.50
## [14,]    0.200     5.00
## [15,]   10.550   179.50
## [16,]   27.660   115.00
## [17,]    1.410    17.50
## [18,]    0.550     2.40
## [19,]   14.830    98.20
## [20,]  529.000   680.00
## [21,]   60.000    81.00
## [22,]    1.040     5.50
## [23,]  207.000   406.00
## [24,]    3.600    21.00
## [25,]    4.190    58.00
## [26,]   85.000   325.00
## [27,]    4.288    39.20
## [28,]    0.425     6.40
## [29,]    0.750    12.30
## [30,]    0.280     1.90
## [31,]    0.101     4.00
## [32,]   62.000  1320.00
## [33,]    0.075     1.20
## [34,]    0.920     5.70
## [35,] 6654.000  5712.00
```

```
## [36,]    0.122     3.00
## [37,]    1.000     6.60
## [38,]    3.500     3.90
## [39,]    0.048     0.33
## [40,]    0.005     0.10
## [41,]    6.800   179.00
## [42,]  192.000   180.00
## [43,]    0.060     1.00
## [44,]   35.000    56.00
## [45,]    3.000    25.00
## [46,]    3.500    10.80
## [47,]    4.050    17.00
## [48,]  160.000   169.00
## [49,]    2.000    12.30
## [50,]    0.120     1.00
## [51,]    0.900     2.60
## [52,]    1.700     6.30
## [53,]    0.023     0.40
## [54,]    1.620    11.40
## [55,] 2547.000  4603.00
## [56,]    0.010     0.30
## [57,]    0.104     2.50
## [58,]    0.023     0.30
## [59,]    1.400    12.50
## [60,]    4.235    50.40
## [61,]  187.100   419.00
## [62,]  250.000   490.00
## [63,]       NA       NA
```

```
#Summarize data
summary(weights)
```

```
##     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
##  NA's   :1          NA's   :1
```

Part d

```
#Store URL
url4 <-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"

#Read in data, skip first row and use the second row as header, remove the
#comment character = "#"
tomatoes <- read.table(url4, header = T, sep = "", skip = 1, comment.char = "")

#Split the data in each of the six cells by ","
I_X10000 <- do.call("rbind", strsplit(toString(tomatoes[1,1]), ","))
```

```
I_X20000 <- do.call("rbind", strsplit(toString(tomatoes[1,2]), ","))
I_X30000 <- do.call("rbind", strsplit(toString(tomatoes[1,3]), ","))
P_X10000 <- do.call("rbind", strsplit(toString(tomatoes[2,1]), ","))
P_X20000 <- do.call("rbind", strsplit(toString(tomatoes[2,2]), ","))
P_X30000 <- do.call("rbind", strsplit(toString(tomatoes[2,3]), ","))

#Combine the vectors of split data
I <- cbind(I_X10000, I_X20000, I_X30000)
P <- cbind(P_X10000, P_X20000, P_X30000)

tomatoes <- rbind(I,P)

#Name columns
colnames(tomatoes) <- c("10000_1", "10000_2", "10000_3","20000_1", "20000_2",
                        "20000_3", "30000_1", "30000_2", "30000_3")

#Convert to tbl_class to use dplyr and tidyr functions
tomatoes <- tbl_df(tomatoes)

#Create column with variety names (I = Ife\#1, P = PusaEarlyDwarf)
tomatoes <- mutate(tomatoes, "Variety" = c("I", "P"))

#Reorder columns
tomatoes <- select(tomatoes, "Variety", "10000_1", "10000_2", "10000_3","20000_1",
                   "20000_2", "20000_3", "30000_1", "30000_2", "30000_3" )

#Give each observation its own row
tomatoes <- gather(tomatoes, X1, Response, -Variety)

#Fix density labels and remove unnecessary column
tomatoes <- separate(tomatoes, X1, c("Density", "temp"))
tomatoes <- select(tomatoes, Variety, Density, Response)

#Sort by variety and then density
tomatoes <- arrange(tomatoes, Variety, Density)

#Convert Response type to Numeric
tomatoes <- transform(tomatoes, Response = as.numeric(Response))

#Final Dataset:
tomatoes
```

```
##    Variety Density Response
## 1        I   10000     16.1
## 2        I   10000     15.3
## 3        I   10000     17.5
## 4        I   20000     16.6
## 5        I   20000     19.2
## 6        I   20000     18.5
## 7        I   30000     20.8
## 8        I   30000     18.0
## 9        I   30000     21.0
## 10       P   10000      8.1
```

```
## 11       P   10000      8.6
## 12       P   10000     10.1
## 13       P   20000     12.7
## 14       P   20000     13.7
## 15       P   20000     11.5
## 16       P   30000     14.4
## 17       P   30000     15.4
## 18       P   30000     13.7
```

```
#Summarize data by variety
tomatoes %>%
  group_by(Variety) %>%
  summarise(Mean = mean(Response), Std_dev = sd(Response), Min = min(Response),
            Median = median(Response), Max = max(Response))
```

```
## # A tibble: 2 x 6
##   Variety  Mean Std_dev   Min Median   Max
##   <chr>   <dbl>   <dbl> <dbl>  <dbl> <dbl>
## 1 I        18.1    1.99  15.3   18    21
## 2 P        12.0    2.60   8.1   12.7  15.4
```

```
#Summarize data by density
tomatoes %>%
  group_by(Density) %>%
  summarise(Mean = mean(Response), Std_dev = sd(Response), Min = min(Response),
            Median = median(Response), Max = max(Response))
```

```
## # A tibble: 3 x 6
##   Density  Mean Std_dev   Min Median   Max
##   <chr>   <dbl>   <dbl> <dbl>  <dbl> <dbl>
## 1 10000    12.6    4.15   8.1   12.7  17.5
## 2 20000    15.4    3.19  11.5   15.2  19.2
## 3 30000    17.2    3.21  13.7   16.7  21
```

Problem 5

```
# 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')

#Convert to tbl_class to use dplyr functions
plants <- tbl_df(plants)

#Remove rows with NA values
plants <- na.omit(plants)

#Add column "pH_Mid" considering the midpoint of ph_Min and pH_Max
plants <- mutate(plants, 'pH_Mid' = ((pH_Min + pH_Max) / 2))

#Final dataset
plants
```

```
## # A tibble: 813 x 11
##    Scientific_Name Duration Active_Growth_P~ Foliage_Color pH_Min pH_Max
##    <fct>           <fct>    <fct>            <fct>          <dbl>  <dbl>
##  1 Abies balsamea  Perenni~ Spring and Summ~ Green            4      6
##  2 Acacia constri~ Perenni~ Spring and Summ~ Green            7      8.5
##  3 Acalypha virgi~ Annual   Spring, Summer,~ Green            5.9    7
##  4 Acer negundo    Perenni~ Spring and Summ~ Green            5      7.8
##  5 Acer nigrum     Perenni~ Spring and Summ~ Green            4.5    7.3
##  6 Acer pensylvan~ Perenni~ Spring and Summ~ Green            4.4    6.5
##  7 Acer platanoid~ Perenni~ Spring and Summ~ Green            4.8    7.2
##  8 Acer pseudopla~ Perenni~ Spring and Summ~ Yellow-Green     5.8    7
##  9 Acer rubrum     Perenni~ Spring and Summ~ Green            4.7    7.3
## 10 Acer saccharin~ Perenni~ Spring and Summ~ Green            4      7.3
## # ... with 803 more rows, and 5 more variables: Precip_Min <int>,
## #   Precip_Max <int>, Shade_Tolerance <fct>, Temp_Min_F <int>,
## #   pH_Mid <dbl>
```

```
#Summarize data by foliage color
plants %>%
  group_by(Foliage_Color) %>%
  summarise(Mean = mean(pH_Mid), Std_dev = sd(pH_Mid), Min = min(pH_Mid),
            Median = median(pH_Mid), Max = max(pH_Mid))
```

```
## # A tibble: 6 x 6
##   Foliage_Color  Mean Std_dev   Min Median   Max
##   <fct>         <dbl>   <dbl> <dbl>  <dbl> <dbl>
## 1 Dark Green     6.00   0.556  4.75   6     7.15
## 2 Gray-Green     6.37   0.639  5.25   6.28  7.5
## 3 Green          6.18   0.525  4.65   6.15  8.2
## 4 Red            6.4    0.984  5.5    6.25  7.45
## 5 White-Gray     6.44   0.738  5.5    6.25  7.75
## 6 Yellow-Green   5.94   0.604  4.3    6     7.2
```

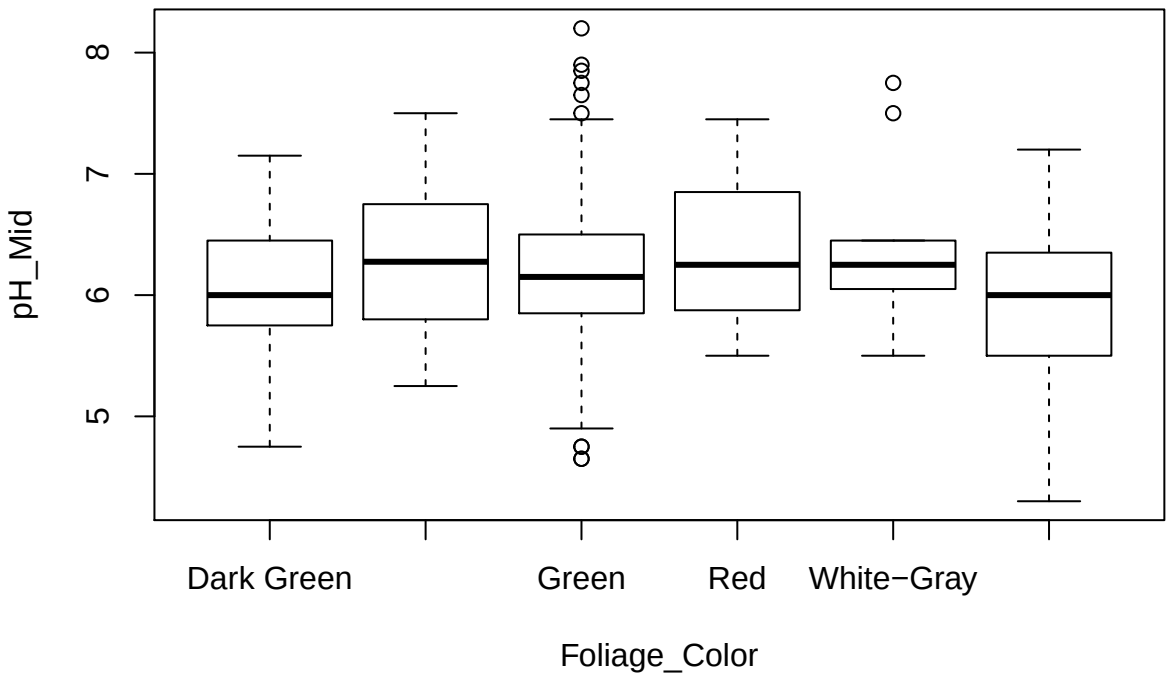
```
#Fit linear model
plants.lm <- lm(pH_Mid ~ Foliage_Color, plants)

#Table of coefficients
summary(plants.lm)
```

```
##
## Call:
## lm(formula = pH_Mid ~ Foliage_Color, data = plants)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63750 -0.37083 -0.02511  0.32489  2.02489
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.99939    0.05940 101.001  < 2e-16 ***
## Foliage_ColorGray-Green     0.37144    0.12483   2.976  0.00301 **
## Foliage_ColorGreen          0.17572    0.06290   2.793  0.00534 **
## Foliage_ColorRed            0.40061    0.31618   1.267  0.20551
## Foliage_ColorWhite-Gray     0.44505    0.18888   2.356  0.01870 *
## Foliage_ColorYellow-Green -0.06189    0.13414  -0.461  0.64465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5379 on 807 degrees of freedom
## Multiple R-squared:  0.02189,    Adjusted R-squared:  0.01583
## F-statistic: 3.613 on 5 and 807 DF,  p-value: 0.003077
```

```
#ANOVA results
plants.anova <- aov(pH_Mid ~ Foliage_Color, plants)
summary(plants.anova)
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## Foliage_Color   5   5.23  1.0452   3.613 0.00308 **
## Residuals     807 233.48  0.2893
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Boxplot
plot(pH_Mid ~ Foliage_Color, plants)
```


pH_Mid
5
6
7
8
Dark Green
Green
Red
White−Gray
Foliage_Color