HW2_Brown

Nick Brown

9/10/2019

R Markdown

Problem 3

Version control is necessary to save all documents in a centralized repository. For example, I switched computers today, but started HW2 on a different computer. My old computer was reformated, but I forgot to save this homework into a cloud-based repository. Now, I must re configure my git enviroment and redo this homework assignment. This is great for reinforcement knowledge, but had I pushed my working document onto github, I could have pulled it then continued working.

I will continue to use version control for the rest of my PhD tenure as version control is a vital skill set to have as a future professor and/or data scientist.

Problem 4

a. Sensory data from five operators:

Process:The process includes creating a url for the sensory data, then creating a table of the sensory data using the read.table function. At first, the function could not read the table since the fields were of unequal sizes. By using the fill = TRUE statement, then I was able to read the table. Afterward, I used the View function to view the contents of the function in the data frame.

Observations and Issues with the data: These data are not in the correct fields which is why the error occurred when trying to read in the data. It appears that Operators 1 through 5 should be headers for each column, but the header appears to be added in with the data. Additionally, the Items rows appear to be comingled with the data, as these should simply be the headers for each row.

Furthermore, there appears to be 10 items with a total of three repeated measures. To clean these data up, there should be a column for the five operators and their respective senses and another column with the item numbers to note the repeated measures occurrences.

```
#sensory_url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
sensory_table <- read.table("https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat", strin
View(sensory_table)
```

b. Gold Medal performance for Olympic Men's Long Jump:

Process:The process includes creating a url for the gold medal data, then creating a table of the gold medal data using the read.table function. At first, the function could not read the table since the fields were of unequal sizes. By using the fill = TRUE statement, then I was able to read the table. Afterward, I used the View function to view the contents of the function in the data frame.

Observations and Issues with the data: The gold medals data did not read in correctly as there are uneven fields. The issue appears to be that the title of the columns were parsed and hence too many variables were created. As a result, the data were placed into successive fields.

Solution–There should only be two fields: "Year" and "Long Jump Distance". The year should begin with 1896 then conclude at 1992 using 4 year intervals in between. The corresponding gold medal long jump distance should be provided adjacent to each year.

```r
gold_url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
gold_table <- read.table(gold_url,stringsAsFactors = FALSE, fill = TRUE)
View(gold_table)
```

c. Brain weight and body weight:

Process:The process includes creating a url for the brain and body weight data, then creating a table of the brain and body weight data using the read.table function. At first, the function could not read the table since the fields were of unequal sizes. By using the fill = TRUE statement, then I was able to read the table. Afterward, I used the View function to view the contents of the function in the data frame.

Observations and Issues with the data: Again, the read.table function did not want to read in this data due to uneven fields. The fill = TRUE statement was required to allow for the data to be imported into a table. Similar to the issue above, the header titles were parsed based on spaces, so extra variables were created and data were inserted into these unnecessary variables.

Solution–There should be only two columns: "Brain weight" and "Body weight". The larger values should be placed into the body weight variable and the lesser weights should be placed into the brain weight variable.

```r
brain_url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
brain_table <- read.table(brain_url,stringsAsFactors = FALSE, fill = TRUE)
View(brain_table)
```

d. Triplicate measurements of tomato yield for two varieties of tomatoes:

Process:The process includes creating a url for the tomato data, then creating a table of the tomato data using the read.table function. At first, the function could not read the table since the fields were of unequal sizes. By using the fill = TRUE statement, then I was able to read the table. Afterward, I used the View function to view the contents of the function in the data frame.

Observations and Issues with the data: These data were not read into the tables correctly due to the contents within each variable. The contents within each variable included multiple values that were separated by commas. As a result, these data show single values within the variable with three comma separated values, representing the triplicate measurements.

Solution–There should be a column/variable required for each of the three measurements. There should be a fourth column to designate tomato variety.

```r
tomato_url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
tomato_table <- read.table(tomato_url,fill = TRUE)
View(tomato_table)
```

Problem 5

Code to import and clean the dataset for Plants

```r
# Install the swirl package
options(repos = c(CRAN="http://cran.us.r-project.org"))
install.packages("swirl")
```

```
## Installing package into 'C:/Users/Nick/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
## package 'swirl' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Nick\AppData\Local\Temp\RtmpYvjkVK\downloaded_packages
```

```r
install.packages("psych")
```

```
## Installing package into 'C:/Users/Nick/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
## package 'psych' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Nick\AppData\Local\Temp\RtmpYvjkVK\downloaded_packages
```

```r
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
library(psych)
install_course("R_Programming_E")
```

```
##
  |
  |                                                                 |   0%
  |
  |=                                                                |   1%
  |
  |==                                                               |   3%
  |
  |===                                                              |   4%
  |
  |=====                                                            |   8%
  |
  |=======                                                          |  10%
  |
  |==========                                                       |  15%
  |
  |=============                                                    |  20%
  |
  |==============                                                   |  22%
  |
  |===============                                                  |  23%
  |
  |================                                                 |  25%
  |
  |==================                                               |  28%
  |
  |===================                                              |  29%
  |
  |=====================                                            |  33%
  |
  |=======================                                          |  36%
  |
  |=========================                                        |  38%
  |
  |===========================                                      |  41%
  |
  |=============================                                    |  45%
  |
  |================================                                 |  49%
  |
  |================================                                 |  50%
  |
  |=================================                                |  51%
  |
  |====================================                             |  55%
  |
  |=====================================                            |  57%
  |
  |========================================                         |  61%
  |
  |========================================                         |  62%
  |
  |===========================================                      |  66%
  |
  |==============================================                   |  70%
  |
  |================================================                 |  75%
  |
  |===================================================              |  79%
  |
  |======================================================           |  83%
  |
  |========================================================         |  86%
  |
  |=========================================================        |  87%
  |
  |===========================================================      |  91%
  |
  |==============================================================   |  95%
  |
  |=================================================================| 100%
```

```
##
## | Course installed successfully!
```

```r
# Create a path to the plants data
.datapath <- file.path(path.package("swirl"), 'Courses', 'R_Programming_E', 'Looking_at_Data', 'plant-da

# Read in the plants data
plants <- read.csv(.datapath, strip.white = TRUE, na.strings = "")

# Remove excessive columns in plants data
.cols2rm <- c('Accepted.Symbol', 'Synonym.Symbol')
plants <- plants[, !(names(plants) %in% .cols2rm)]

# Relabel columns in plants data
names(plants) <- c('Scientific_Name', 'Duration', 'Active_Growth_Period', 'Foliage_Color', 'pH_Min', 'pH

# Remove rows and columns with NA values in plants data

complete_plants <- na.omit(plants)

# Summary of plants
summary(complete_plants)
```

```
##            Scientific_Name                        Duration
##  Abies balsamea     :  1    Perennial                 :692
##  Acacia constricta  :  1    Annual                    : 64
##  Acalypha virginica:  1    Annual, Perennial         : 33
##  Acer negundo       :  1    Annual, Biennial          :  8
##  Acer nigrum        :  1    Annual, Biennial, Perennial:  6
##  Acer pensylvanicum:  1    Biennial, Perennial       :  6
##  (Other)            :807    (Other)                   :  4
##             Active_Growth_Period      Foliage_Color     pH_Min
##  Spring and Summer      :443       Dark Green  : 82   Min.   :3.000
##  Spring                 :143       Gray-Green  : 24   1st Qu.:4.500
##  Spring, Summer, Fall   : 90       Green       :675   Median :5.000
##  Summer                 : 87       Red         :  3   Mean   :4.988
##  Summer and Fall        : 20       White-Gray  :  9   3rd Qu.:5.500
##  Fall, Winter and Spring: 15       Yellow-Green: 20   Max.   :7.000
##  (Other)                : 15
##      pH_Max          Precip_Min      Precip_Max        Shade_Tolerance
##  Min.   : 5.100   Min.   : 4.00   Min.   : 16.00   Intermediate:239
##  1st Qu.: 7.000   1st Qu.:17.00   1st Qu.: 55.00   Intolerant  :332
##  Median : 7.300   Median :29.00   Median : 60.00   Tolerant    :242
##  Mean   : 7.335   Mean   :25.66   Mean   : 58.64
##  3rd Qu.: 7.700   3rd Qu.:32.00   3rd Qu.: 60.00
##  Max.   :10.000   Max.   :60.00   Max.   :200.00
##
##    Temp_Min_F
##  Min.   :-79.00
##  1st Qu.:-38.00
##  Median :-33.00
##  Mean   :-22.57
##  3rd Qu.:-18.00
##  Max.   : 52.00
##
```

```
# Determine the unique foliage colors in the dataset
unique(complete_plants[,4], incomparables = FALSE)
```

```
## [1] Green        Yellow-Green Dark Green   White-Gray   Gray-Green  
## [6] Red         
## Levels: Dark Green Gray-Green Green Red White-Gray Yellow-Green
```

```
# Dummy code the names of colors into independent variables with 0 and 1 for each of the six different
colors <- dummy.code(complete_plants$Foliage_Color)

# Add new color variables to the data.frame
new_complete_plants <- data.frame(colors, complete_plants)

# Reorder the variables to include only the necessary variables: name, color, and pH values
reordered_plants <- new_complete_plants[c(7,11,12,10,1,2,3,4,5,6)]

# Find midpoint between pH_Min and pH_max and store in ph_combined variable
ph_combined <- (reordered_plants$pH_Min + reordered_plants$pH_Max)/2

# Create final dataset with ph_combined as the dependent variable
final_plants <- data.frame(ph_combined, reordered_plants)
View(final_plants)

# Multiple linear regression model with pH_level as the dependent variable and color as the predictors
pH_level_model <- lm(ph_combined ~ White.Gray + Yellow.Green + Dark.Green + Gray.Green + Green + Red, da
summary(pH_level_model)
```

```
## 
## Call:
## lm(formula = ph_combined ~ White.Gray + Yellow.Green + Dark.Green + 
##     Gray.Green + Green + Red, data = final_plants)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.63750 -0.37083 -0.02511  0.32489  2.02489 
## 
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   6.40000    0.31055  20.609   <2e-16 ***
## White.Gray    0.04444    0.35859   0.124    0.901    
## Yellow.Green -0.46250    0.33303  -1.389    0.165    
## Dark.Green   -0.40061    0.31618  -1.267    0.206    
## Gray.Green   -0.02917    0.32939  -0.089    0.929    
## Green        -0.22489    0.31124  -0.723    0.470    
## Red                NA         NA      NA       NA    
## ---
## 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
```

Based on the regression model, the R-Squared value is 0.02. The is interpreted as 2% of the variation in pH levels is caused by the colors of the foliage.