HW2_wgeither

Warren Geither

9/7/2020

Problem 3

In terms of team collaboration, version control seems like by far the best way to go about a project way more control and systematic then going back and forth through emails. I've already started using github to back up work for other classes, so i think it is a great tool.

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 dataset, you should perform the cleaning 2x: first with base R functions (ie no dplyr, piping, etc), second using tidyverse function. Make sure you weave your code and text into a complete description of the process and end by creating a tidy dataset describing the variables, create a summary table of the data (summary, NOT full listing), note issues with the data, and include an informative plot.

a. Sensory data from five operators. – see video, I am doing this one http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat

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

sensory_data <- fread(url, fill=TRUE, header=TRUE)
saveRDS(sensory_data, "sensory_data_raw.RDS")
sensory_df <- readRDS("sensory_data_raw.RDS")
```

Looks like data has 2 columns and 31 rows, it should have 6 columns.

```
# delete first row
sensory_df <- sensory_df[2:nrow(sensory_df)]

# delete na column
sensory_df <- sensory_df[, 1]

# split up values into multiple columns
right = function(x,n){
  substring(x,nchar(x)-n+1)
}

out <- strsplit(right(as.character(sensory_df$V1),19),' ')
test_df <- data.frame(do.call(rbind, out))

# rename columns
colnames(test_df) <- c("Op_1", "Op_2", "Op_3", "Op_4", "Op_5")
```

```
# create dataframe of item numbers
item_df <- data.frame("Item" = c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6,7,7,7,8,8,8,9,9,9,10,10,10))

# bind item numbers on our other dataframe
new_df <- cbind(test_df,item_df)

# re-order columns
tidy_sensory_df <- new_df[,c(6,1,3,2,4,5)]
```

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

Looks like this data has way too many columns. We need to combine all data together into 2 columns "Year" and "Long Jump"

```
# read in url
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"

# save to rds so we are reslient against changes on the internet
#olympic_data <- fread(url, fill=TRUE, header=TRUE)
#saveRDS(olympic_data, "olympic_data_raw.RDS")
olympic_df <- readRDS("olympic_data_raw.RDS")
```

```
# tried stack, not working so moving on with other method for now
#tidy_olympic_df <- data.frame(stack(olympic_df[c(1,3,5,7)]), stack(olympic_df[c(2,4,6,8)]))

# partition data into separate 2 column data frames
partition1_df = as.data.frame(olympic_df[,  1:2])
partition2_df = as.data.frame(olympic_df[,  3:4])
partition3_df = as.data.frame(olympic_df[,  5:6])
partition4_df = as.data.frame(olympic_df[,  7:8])

# rename columns
colnames(partition1_df) <- c("Year","Long Jump")
colnames(partition2_df) <- c("Year","Long Jump")
colnames(partition3_df) <- c("Year","Long Jump")
colnames(partition4_df) <- c("Year","Long Jump")

# remove NA values
partition4_df = partition4_df[complete.cases(partition4_df), ]

# concate all dataframes together
tidy_olympic_df = rbind(partition1_df,partition2_df,partition3_df,partition4_df)
```

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

```
# read in url
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"

# save to rds so we are reslient against changes on the internet
#brain_data <- fread(url, fill=TRUE, header=TRUE)
#saveRDS(brain_data, "brain_data_raw.RDS")
brain_df <- readRDS("brain_data_raw.RDS")
```

```r
# tried stack, not working so moving on with other method for now

# partition data into separate 2 column data frames
partition1_df = as.data.frame(brain_df[,  1:2])
partition2_df = as.data.frame(brain_df[,  3:4])
partition3_df = as.data.frame(brain_df[,  5:6])

# rename columns
colnames(partition1_df) <- c("Body Wt(kg)","Brain Weight(g)")
colnames(partition2_df) <- c("Body Wt(kg)","Brain Weight(g)")
colnames(partition3_df) <- c("Body Wt(kg)","Brain Weight(g)")

# remove NA values
partition3_df = partition3_df[complete.cases(partition3_df), ]

# concate all dataframes together
tidy_brain_df = rbind(partition1_df,partition2_df,partition3_df)
```

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
# read in url
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"

# save to rds so we are reslient against changes on the internet
tomato_data <- fread(url, fill=TRUE, header=TRUE)
saveRDS(tomato_data, "tomato_data_raw.RDS")
tomato_df <- readRDS("tomato_data_raw.RDS")
```

```r
# remove NA columns
tomato_df <- as.data.frame(tomato_df[,1:4])

# remove first row, since its actually the column names
tomato_df <- as.data.frame(tomato_df[2:3,])

# rename columns
colnames(tomato_df) <- c("Tomato_Variety","pd_1000","pd_2000","pd_3000")

# split up Triplicate measurements in the cell into 3 columns
out <- strsplit(as.character(tomato_df$pd_1000),',')
partition1_df <- data.frame(tomato_df$Tomato_Variety, do.call(rbind, out))

out <- strsplit(as.character(tomato_df$pd_2000),',')
partition2_df <- data.frame(tomato_df$Tomato_Variety, do.call(rbind, out))

out <- strsplit(as.character(tomato_df$pd_3000),',')
partition3_df <- data.frame(tomato_df$Tomato_Variety, do.call(rbind, out))

# rename columns
colnames(partition1_df) <- c("Tomato_Variety","pd_1000_m1","pd_1000_m2","pd_1000_m3")
colnames(partition2_df) <- c("Tomato_Variety","pd_2000_m1","pd_2000_m2","pd_2000_m3")
colnames(partition3_df) <- c("Tomato_Variety","pd_3000_m1","pd_3000_m2","pd_3000_m3")

# merge data frames
one_and_two_merge_df <- merge(partition1_df,partition2_df,by="Tomato_Variety")
```

```r
tidy_tomato_df <- merge(one_and_two_merge_df,partition3_df,by="Tomato_Variety")
```

Problem 5

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