HW2_hwenjun

Wenjun

2020-09-13

Problem 3

In the lecture, there were two links to StackOverflow questions on why one should use version control. In your own words, summarize your thoughts (2-3 sentences) on version control in your future work. No penalties here if you say, useless!

Summary:

It is good to have a version control system to help back up and store my files. It allows me to easily transfer my work from machine to machine and decrease the complexity of managing files. In future, I can keep selective files that generate data or reprots under version constrol system.

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

a. Sensory data from five operators

we are looking at the sensory experiment data obtained by 5 different operators from Wu and Hamada's book: http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat

First, we will get the data from the link above:

```r
## http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
url1 <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
sensory_data_raw <- fread(url1, fill=TRUE, data.table = FALSE)
```

```
saveRDS(sensory_data_raw, "sensory_data_raw.RDS")
sensory_data_raw <- readRDS("sensory_data_raw.RDS")
```

Need to tidy the data, data rows are not aligned, the way display data is not convenient, need to push them into a column.

```
## stack data and fix columns
# Since the data rows are not aligned, we manually delete some rows
sensory_data_raw_dl <- sensory_data_raw[-1,-2]

# Then we convert the value from string to numeric
library(stringr)
sensory_data_raw_dl_nu <- as.numeric(unlist(str_extract_all
                                            (sensory_data_raw_dl, "[.-9]+")))

# We know that 1-10 are number of item, not true value in our table, so we delete them
sensory_data_raw_value <- sensory_data_raw_dl_nu[-c(1,17,33,49,65,81,97,113,129,145)]

## Then we reconstruct data

# Now we change the data frame, so that each row data will have
# item(human subject),operator number(1:5), data value. We can see that each
# experiment repeated 3 times for each operator.

sensory_data_tidy_br <- data.frame(item=sort(rep(1:10,15)),
                                   operator=rep(c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5),10),
                               values=sensory_data_raw_value)
```

Now we have the tidy version of sensory data in baseR.

Now we use tidyverse library to clean our datasets.

```
# We also can use skip to make another raw data frame
sensory_data_raw_new <- fread(url1, fill=TRUE, skip = 1,data.table = FALSE)
saveRDS(sensory_data_raw_new, "sensory_data_raw_new.RDS")
sensory_data_raw_new <- readRDS("sensory_data_raw_new.RDS")

# We can see some of the data locates on wrong columns.
# We need to move some of the data to different columns.
num <- 1:30
seq <- c(1,4,7,10,13,16,19,22,25,28)

for (i in num[-seq]){
  sensory_data_raw_new[i,6] <- sensory_data_raw_new[i,5]
  sensory_data_raw_new[i,5] <- sensory_data_raw_new[i,4]
  sensory_data_raw_new[i,4] <- sensory_data_raw_new[i,3]
  sensory_data_raw_new[i,3] <- sensory_data_raw_new[i,2]
  sensory_data_raw_new[i,2] <- sensory_data_raw_new[i,1]
}

# Let first column Item has number 1:5
sensory_data_raw_new[,1] <- sort(rep(c(1:10),3))

# Now we have data in a much better format, rename columns
colnames(sensory_data_raw_new) <-c("Item","1","2","3","4","5")
```

```
## stack and fix column names using tidyverse

sensory_data_tidy_tv <-sensory_data_raw_new %>%
                      gather(key = "operator",value = "value", "1":"5")

head(sensory_data_tidy_tv)
```

```
##   Item operator value
## 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
```

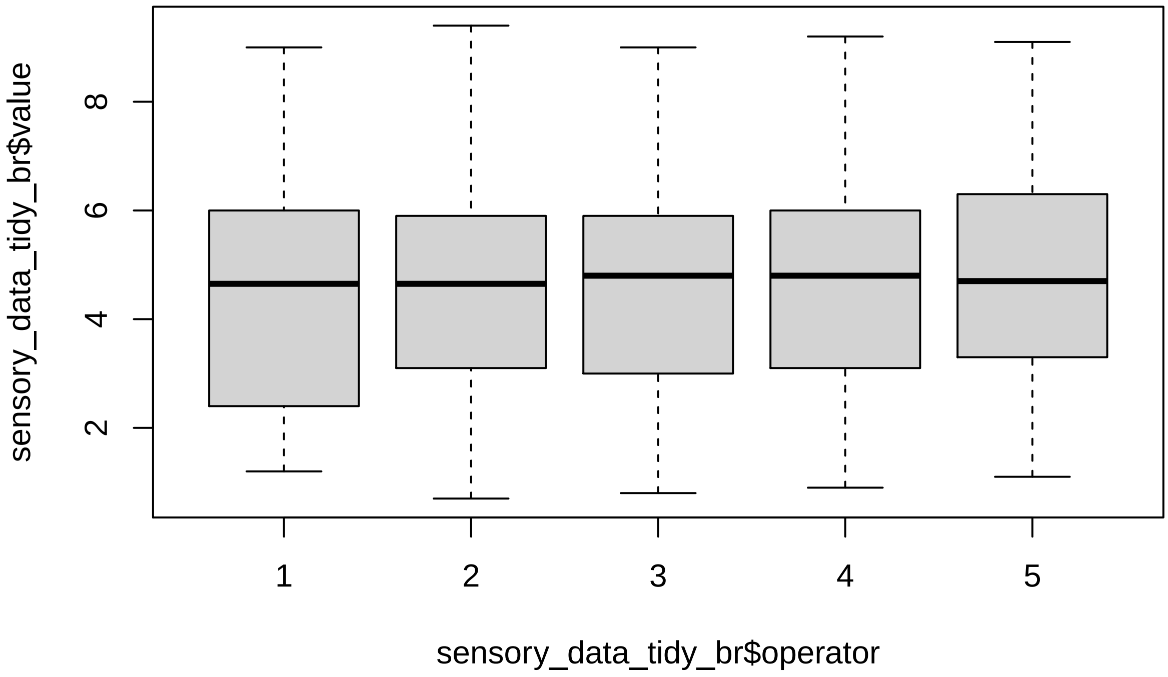
Now we have the tidy version of sensory data in tidyverse.

We have converted the data frames to tidy data frames using the base functions. Here is a summary of the data:

```
knitr::kable(summary(sensory_data_tidy_br))
```

item	operator	values
Min. : 1.0	Min. :1	Min. :0.700
1st Qu.: 3.0	1st Qu.:2	1st Qu.:3.025
Median : 5.5	Median :3	Median :4.700
Mean : 5.5	Mean :3	Mean :4.657
3rd Qu.: 8.0	3rd Qu.:4	3rd Qu.:6.000
Max. :10.0	Max. :5	Max. :9.400

```
boxplot(sensory_data_tidy_br$value~sensory_data_tidy_br$operator)
```



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

we are looking at the Olympic Men's Long Jump data obtained from 6 years from Wu and Hamada's book: http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat

First, we will get the data from the link above:

Need to tidy the data and reshape the data, there are two invalid NA data sets exist.

```
## Reshape data and fix columns
jump_data_tidy_br <- data.frame(Year = as.vector(
  cbind(jump_data_raw[,1],jump_data_raw[,3],jump_data_raw[,5],jump_data_raw[,7])),
  Long_Jump = as.vector(cbind(jump_data_raw[,2],jump_data_raw[,4],
                              jump_data_raw[,6],jump_data_raw[,8])))
# Delete last two rows which has NA
jump_data_tidy_br <- jump_data_tidy_br[-(23:24),]
```

Now we have the tidy version of long jump data in baseR.

Now we use tidyverse library to clean our datasets.

```
## stack and fix column names using tidyverse, get rid of NA values

jump_data_tidy_tv <- bind_rows(jump_data_raw[,1:2], jump_data_raw[,3:4],
                               jump_data_raw[,5:6], jump_data_raw[1:4,7:8])

head(jump_data_tidy_tv)
```

```
##   Year Long_Jump
## 1   -4    249.75
## 2    0    282.88
## 3    4    289.00
## 4    8    294.50
## 5   12    299.25
## 6   20    281.50
```

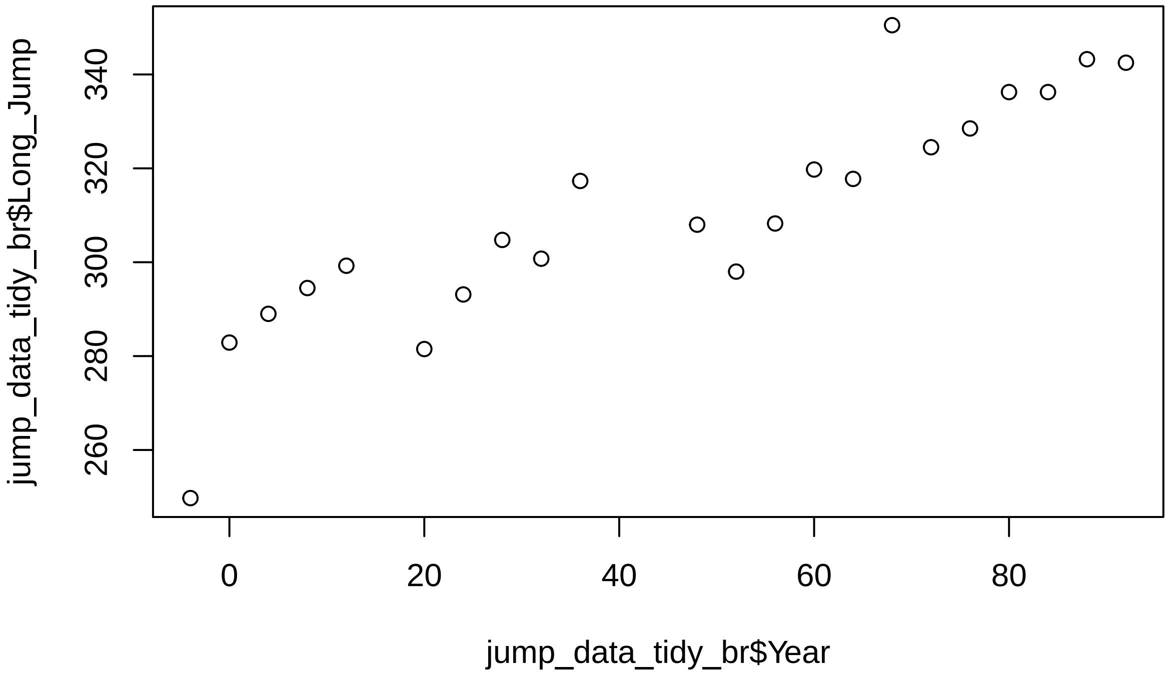
Now we have the tidy version of long jump data in tidyverse library.

We have converted the data frames to tidy data frames using the base functions. Here is a summary of the data:

```
knitr::kable(summary(jump_data_tidy_br))
```

Year	Long_Jump
Min. :-4.00	Min. :249.8
1st Qu.:21.00	1st Qu.:295.4
Median :50.00	Median :308.1
Mean :45.45	Mean :310.3
3rd Qu.:71.00	3rd Qu.:327.5
Max. :92.00	Max. :350.5

```
plot(jump_data_tidy_br$Long_Jump~jump_data_tidy_br$Year)
```



c. Brain weight (g) and body weight (kg)

First, we will get the data from the link above:

```
## http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
url3 <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
brain_data_raw <- fread(url3, fill=TRUE, skip =1,data.table = FALSE)
colnames(brain_data_raw) <-c("Body_Wt","Brain_Wt","Body_Wt",
                             "Brain_Wt","Body_Wt","Brain_Wt")
saveRDS(brain_data_raw, "brain_data_raw.RDS")
brain_data_raw <- readRDS("brain_data_raw.RDS")
```

Need to tidy the data and reshape the data, there is one invalid NA data set exist.

```
## Reshape data and fix columns
brain_data_tidy_br <- data.frame(Body_Wt = as.vector(
  cbind(brain_data_raw[,1],brain_data_raw[,3],brain_data_raw[,5])),
  Brain_Wt = as.vector(cbind(brain_data_raw[,2],brain_data_raw[,4],
                              brain_data_raw[,6])))
# Delete last row which is NA
brain_data_tidy_br <- brain_data_tidy_br[-63,]
```

Now we have the tidy version of brain and body weight data in baseR.

Now we use tidyverse library to clean our datasets.

```
## stack and fix column names using tidyverse, get rid of NA values

brain_data_tidy_tv <- bind_rows(brain_data_raw[,1:2], brain_data_raw[,3:4],
                               brain_data_raw[1:20,5:6])

head(brain_data_tidy_tv)
```

```
##   Body_Wt Brain_Wt
## 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
## 6  27.660    115.0
```

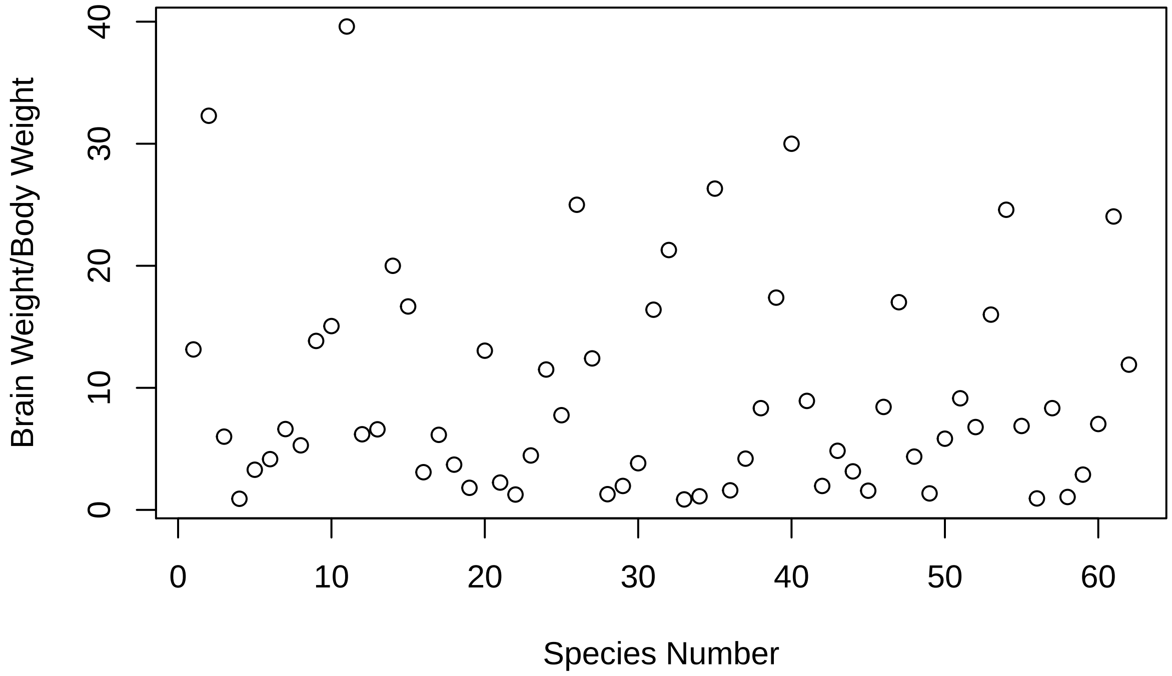
Now we have the tidy version of long jump data in tidyverse library.

We have converted the data frames to tidy data frames using the base functions. Here is a summary of the data:

```
knitr::kable(summary(brain_data_tidy_br))
```

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.: 48.202	3rd Qu.: 166.00
Max. :6654.000	Max. :5712.00

```
plot(brain_data_tidy_br$Brain_Wt/brain_data_tidy_br$Body_Wt~c(1:62),
     xlab = "Species Number",
     ylab = "Brain Weight/Body Weight")
```



d. Triplicate measurements of tomato yield for two varieties of tomatos

First, we will get the data from the link above:

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

```r
tomato_data_raw <- fread(url4, fill=TRUE, skip =1, data.table = FALSE)
saveRDS(tomato_data_raw, "tomato_data_raw.RDS")
tomato_data_raw <- readRDS("tomato_data_raw.RDS")

# We cam see the data is very messy
# First, we construct a data frame that is at least readable

# We only take values from table and convert string to numeric
tomato_data_raw_va <- tomato_data_raw[2:3,2:4]
tomato11 <- as.numeric(unlist(str_extract_all(tomato_data_raw_va[1,1], "[.-9]+")))
tomato21 <-as.numeric(unlist(str_extract_all(tomato_data_raw_va[2,1], "[.-9]+")))
tomato12 <-as.numeric(unlist(str_extract_all(tomato_data_raw_va[1,2], "[.-9]+")))
tomato22 <-as.numeric(unlist(str_extract_all(tomato_data_raw_va[2,2], "[.-9]+")))
tomato13 <-as.numeric(unlist(str_extract_all(tomato_data_raw_va[1,3], "[.-9]+")))
tomato23 <-as.numeric(unlist(str_extract_all(tomato_data_raw_va[2,3], "[.-9]+")))

# We reconstruct data frame
tomato_data_raw_nu <- data.frame(Ife=c(tomato11, tomato12, tomato13),
                                   PusaEarlyDwarf=c(tomato21, tomato22, tomato23),
                               Density=sort(rep(c(10000,20000,30000),3)))
```

For now, we have a readable data frame, but it still need to be further pre-processed.

```r
## Stack and fix columns using baseR

# Reconstruct data frame to a tidy format
tomato_data_tidy_br <- data.frame(stack(tomato_data_raw_nu[,-3]),
                                  Density=rep(tomato_data_raw_nu$Density,2))

colnames(tomato_data_tidy_br) <-c("Yield","Type","Density")
```

Now we have the tidy version of tomato data in baseR.

Now we use tidyverse library to clean our datasets.

```r
## stack and fix column names using tidyverse

tomato_data_tidy_tv <- tomato_data_raw_nu %>%
                      gather(key = "Type", value = "value", Ife:PusaEarlyDwarf)
colnames(tomato_data_tidy_tv) <-c("Density","Type","Yield")

head(tomato_data_tidy_tv)
```

```
##   Density Type Yield
## 1   10000  Ife  16.1
## 2   10000  Ife  15.3
## 3   10000  Ife  17.5
## 4   20000  Ife  16.6
## 5   20000  Ife  19.2
## 6   20000  Ife  18.5
```

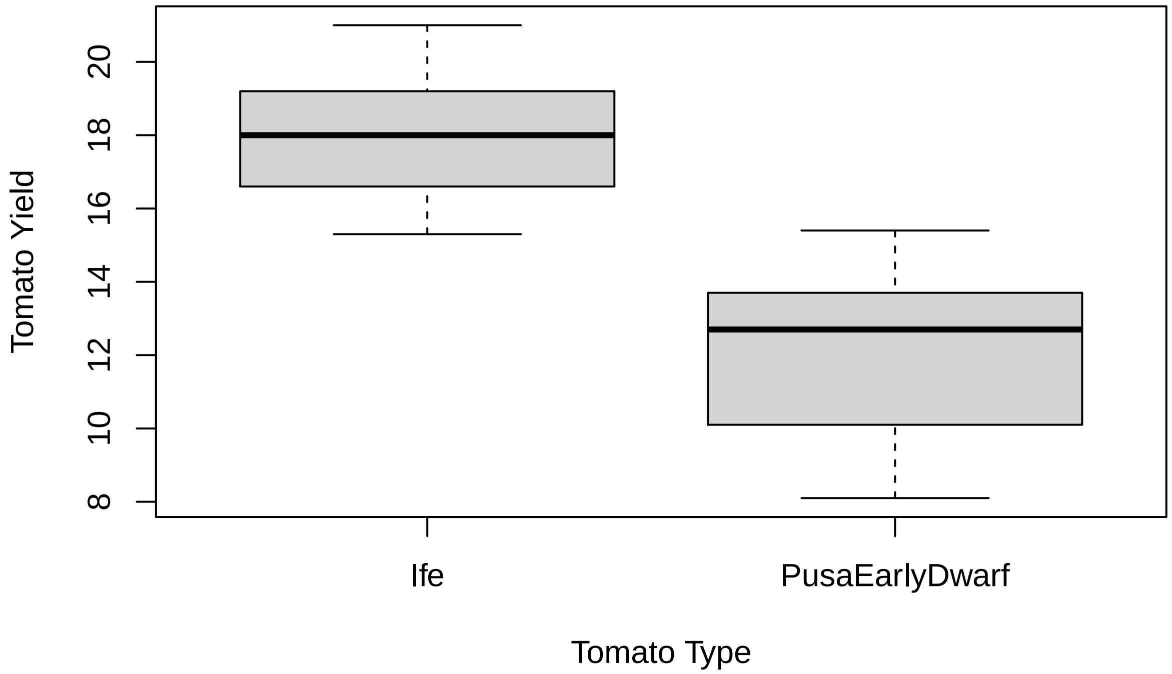
Now we have the tidy version of tomato data in tidyverse library.

We have converted the data frames to tidy data frames using the base functions. Here is a summary of the data:

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

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

```r
boxplot(tomato_data_tidy_br$Yield~tomato_data_tidy_br$Type,
        xlab = "Tomato Type",
        ylab = "Tomato Yield")
```



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

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