Stat 5014 HW3

Bob Settlage

2019-09-25

Problem 1

Swirl – done.

Problem 2

This document.

Problem 3

Here we will read in, clean and filter datasets with the final goal of creating tidy datasets. I am going to create a table as proof of *tidy* data.

Part A: Sensory data

The first data set is the sensory data set. The main problems are the header doesn't import with the correct number of fields and the array is ragged. I am going to use the skip and fill settings to deal with this on import and then manually set the column names. Although, I could read in the column names using readLines using a different sep setting.

```
############################ Problem5_Sensory_analysis get data
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
Sensory_raw <- read.table(url, header = F, skip = 1, fill = T,
    stringsAsFactors = F)
# I recommend downloading and saving, you then work from
# the downloaded version so this knits well from the
# web, I am not saveRDS(Sensory_raw, 'Sensory_raw.RDS')
# Sensory_raw <- readRDS('Sensory_raw.RDS')
Sensory_tidy <- Sensory_raw[-1, ]
Sensory_tidy_a <- filter(.data = Sensory_tidy, V1 %in% 1:10) %>%
    rename(Item = V1, V1 = V2, V2 = V3, V3 = V4, V4 = V5,
        V5 = V6)
Sensory_tidy_b <- filter(.data = Sensory_tidy, !(V1 %in%
    1:10)) %>% mutate(Item = rep(as.character(1:10), each = 2)) %>%
    mutate(V1 = as.numeric(V1)) %>% select(c(Item, V1:V5))
Sensory_tidy <- bind_rows(Sensory_tidy_a, Sensory_tidy_b)
colnames(Sensory_tidy) <- c("Item", paste("Person", 1:5,
    sep = "_"))
Sensory_tidy <- Sensory_tidy %>% gather(Person, value, Person_1:Person_5) %>%
    mutate(Person = gsub("Person_", "", Person)) %>% arrange(Item)

############################
```

That wasn't too bad. The summary of the dataset is included in Table 1 below. Item and Person are characters and value is a number as expected. A plot would be good, but since we aren't going to analyze it, eh.

Table 1: Sensory data summary

Item	Person	value
Length:150	Length:150	Min. :0.70
Class :character	Class :character	1st Qu.:3.02
Mode :character	Mode :character	Median :4.70
NA	NA	Mean :4.66
NA	NA	3rd Qu.:6.00
NA	NA	Max. :9.40

Part B: Long Jump data

As the code is the problem, I include it here. Many times, the code is important to keep and show, but not important to the discussion, so it would get pushed to the Appendix. See Appendix. Again, skip and fill are useful. The main problem here is the table is essentially wrapped and we need to stack it. For good measure, I am adding converting the year code to actual year.

```
############################ Problem5_LongJump_analysis get data
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
LongJump_raw <- read.table(url, header = F, skip = 1, fill = T,
    stringsAsFactors = F)
# saveRDS(LongJump_raw,'LongJump_raw.RDS') LongJump_raw
# <- readRDS('LongJump_raw.RDS')
colnames(LongJump_raw) <- rep(c("V1", "V2"), 4)
LongJump_tidy <- rbind(LongJump_raw[, 1:2], LongJump_raw[,
    3:4], LongJump_raw[, 5:6], LongJump_raw[, 7:8])
LongJump_tidy <- LongJump_tidy %>% filter(!(is.na(V1))) %>%
    mutate(YearCode = V1, Year = V1 + 1900, dist = V2) %>%
    select(-V1, -V2)

############################
```

Again, not terrible. See Table 2. Field values are numbers and there appears to be no missing values.

Table 2: Long Jump data summary

YearCode	Year	dist
Min. :-4.0	Min. :1896	Min. :250
1st Qu.:21.0	1st Qu.:1921	1st Qu.:295
Median :50.0	Median :1950	Median :308
Mean :45.5	Mean :1945	Mean :310
3rd Qu.:71.0	3rd Qu.:1971	3rd Qu.:328
Max. :92.0	Max. :1992	Max. :350

Part C: Brain vs Body data

More munging. Code in Appendix to show how to do multiple code chunks. This one is similar to the last, it is wrapped and we need to stack it.

Ditto to the previous. See table 3.

Table 3: Brain/Body weight data summary

Brain	Body
Min. : 0.00	Min. : 0.10
1st Qu.: 0.60	1st Qu.: 4.25
Median : 3.34	Median : 17.25
Mean : 198.79	Mean : 283.13
3rd Qu.: 48.20	3rd Qu.: 166.00
Max. :6654.00	Max. :5712.00

Part C: Tomato data

This time, the problem is the names are causing troubles. Still not too bad using some regex.

Table 4 gives output from tidying of the tomato data.

Table 4: Tomato data summary

Clone	Replicate	value	Variety
Length:18	Length:18	Length:18	Length:18
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

Appendix 1: R code

```
############################ Problem5_LongJump_analysis get data
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
LongJump_raw <- read.table(url, header = F, skip = 1, fill = T,
    stringsAsFactors = F)
# saveRDS(LongJump_raw,'LongJump_raw.RDS') LongJump_raw
# <- readRDS('LongJump_raw.RDS')
colnames(LongJump_raw) <- rep(c("V1", "V2"), 4)
LongJump_tidy <- rbind(LongJump_raw[, 1:2], LongJump_raw[,
    3:4], LongJump_raw[, 5:6], LongJump_raw[, 7:8])
LongJump_tidy <- LongJump_tidy %>% filter(!(is.na(V1))) %>%
    mutate(YearCode = V1, Year = V1 + 1900, dist = V2) %>%
    select(-V1, -V2)

############################ Problem5_BrainBody_analysis get data
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
BrainBody_raw <- read.table(url, header = F, skip = 1, fill = T,
    stringsAsFactors = F)
# saveRDS(BrainBody_raw,'BrainBody_raw.RDS')
# BrainBody_raw<-readRDS('BrainBody_raw.RDS')
colnames(BrainBody_raw) <- rep(c("Brain", "Body"), 3)
BrainBody_tidy <- rbind(BrainBody_raw[, 1:2], BrainBody_raw[,
    3:4], BrainBody_raw[, 5:6])
BrainBody_tidy <- BrainBody_tidy %>% filter(!(is.na(Brain)))

############################
```