Homework 2

Steven Barnett

9/15/2020

Problem 3

Version control is essential to collaboration in any team environment, enabling engineers and analysts to work on the same code base simultaneously. Version control is also extremely helpful in understanding the history of a code base and allows all changes are to be easily reviewed and tracked. Version control is simple for the first time user but has enormous capacity for the those who invest time in understanding its full offering.

Problem 4

a) Sensory Data

First, I am going to pull down the dataset and save it to local storage.

```
## sensory_data_url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
## sensory_data <- fread(sensory_data_url, skip = 1, fill = TRUE, data.table = FALSE)
## saveRDS(sensory_data, "dwnldd_data/sensory_data_raw.RDS")
sensory_data <- readRDS("dwnldd_data/sensory_data_raw.RDS")
```

The dataset has some incorrect row lengths due to indices being included in the data. We will fix these using Base R functions.

```
sensory_data_base_r <- sensory_data
# Loop through each row that is too long and remove the first entry
for (i in seq(from = 1, to = 30, by = 3)) {
  sensory_data_base_r[i,] = sensory_data_base_r[i,][2:6]
}
# Remove the sixth column that is full of NA
sensory_data_base_r <- subset(sensory_data_base_r, select = -c(6))
names(sensory_data_base_r) <- c("op1", "op2", "op3", "op4", "op5")
```

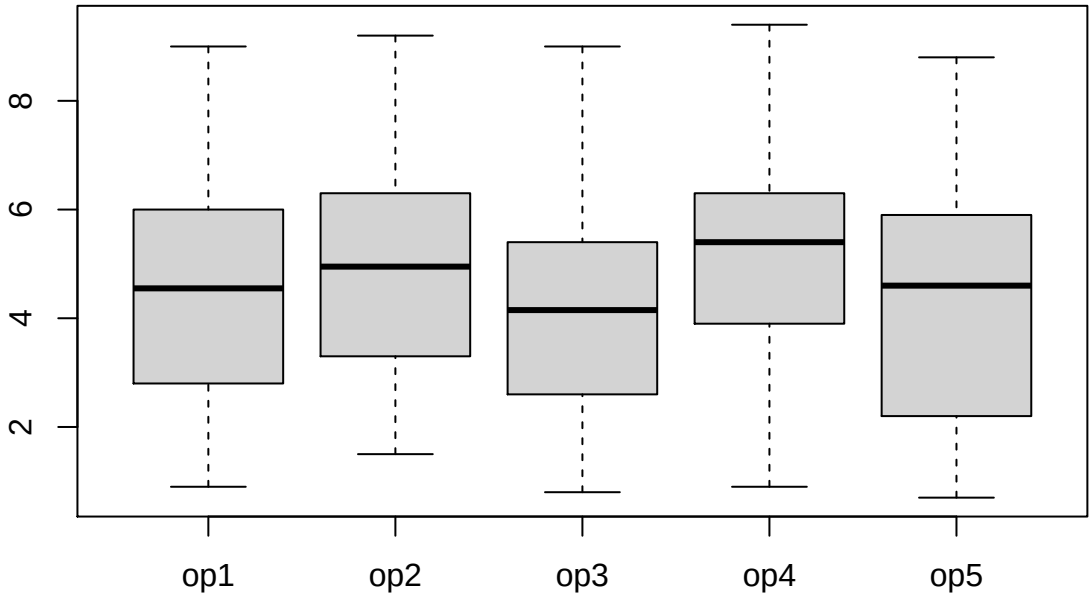
Now that we have cleaned the sensory data set with Base R functions, we will attempt to do a similar cleaning with tidyverse functions.

```
sensory_data_tidyverse <- sensory_data
# Create data set of rows with extra index
jagged_data <- select(filter(sensory_data_tidyverse, !is.na(`5`)), !Item)
```

```r
# Create data set of rows without extra index
unjagged_data <- select(filter(sensory_data_tidyverse, is.na(`5`)), !`5`)
colnames(unjagged_data) <- colnames(sensory_data_tidyverse)[2:6]
sensory_data_tidyverse <- union(jagged_data, unjagged_data)
colnames(sensory_data_tidyverse) <- c("op1", "op2", "op3", "op4", "op5")
```

After cleaning the dataset using Base R and tidyverse functions, I am going to display the cleaned data.

op1	op2	op3	op4	op5
Min. :0.900	Min. :1.500	Min. :0.800	Min. :0.900	Min. :0.700
1st Qu.:2.850	1st Qu.:3.450	1st Qu.:2.650	1st Qu.:3.925	1st Qu.:2.250
Median :4.550	Median :4.950	Median :4.150	Median :5.400	Median :4.600
Mean :4.593	Mean :5.063	Mean :4.167	Mean :5.193	Mean :4.267
3rd Qu.:5.950	3rd Qu.:6.225	3rd Qu.:5.400	3rd Qu.:6.275	3rd Qu.:5.800
Max. :9.000	Max. :9.200	Max. :9.000	Max. :9.400	Max. :8.800



b) Gold Medal Data

First, I will pull the data down and store it locally.

```r
## gold_medal_data_url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
## gold_medal_data <- fread(gold_medal_data_url, fill=TRUE)
## saveRDS(gold_medal_data, "dwnldd_data/gold_medal_data_raw.RDS")
gold_medal_data <- readRDS("dwnldd_data/gold_medal_data_raw.RDS")
```

The data set contains values for two columns, but the values are spread across 8 columns, along with some empty values. We will fix these using Base R functions.

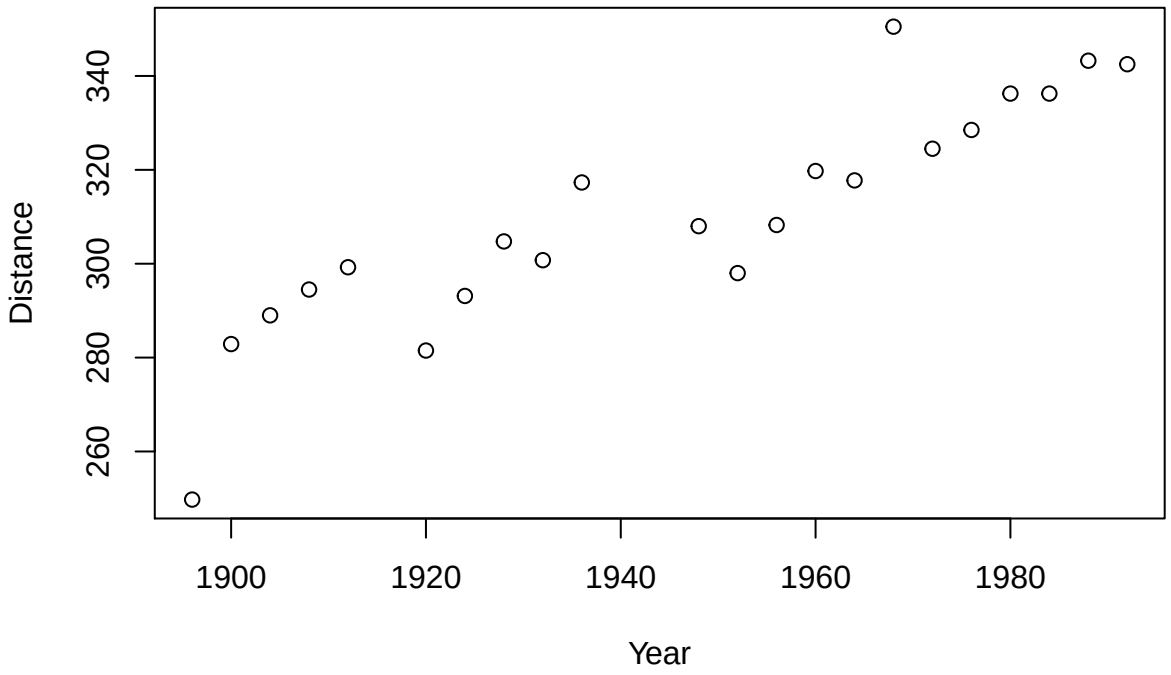
```
gold_medal_data_base_r <- gold_medal_data
## Create empty data frame to fill
gold_medal_data_frame <- data.frame(Year=integer(), Distance=numeric())
## Loop through data set by every other column and fill data frame with Year and Distance values
for (i in seq(from = 1, to = 11, by = 2)) {
  for (j in seq(from = 1, to = 6, by = 1)) {
    data_row <- gold_medal_data_base_r[j,]
    year <- data_row[[i]]
    distance <- data_row[[i+1]]
    gold_medal_data_frame <- rbind(gold_medal_data_frame, c(year, distance))
  }
}
names(gold_medal_data_frame) <- c("Year", "Distance")
gold_medal_data_frame <- na.omit(gold_medal_data_frame)
## Adjust year values to reflect actual year and not difference from 1900
gold_medal_data_frame$Year <- gold_medal_data_frame$Year + 1900
```

We will now attempt to clean the data set in a similar fashion using tidyverse functions.

```
gold_medal_data_tidyverse <- gold_medal_data_base_r
colnames(gold_medal_data_tidyverse) <- c(rep(c("Year_diff", "Distance"), 6))
gold_medal_data_tidyverse_joined <- union(select(gold_medal_data_tidyverse, 1:2), select(gold_medal_data
gold_medal_data_tidyverse_joined <- union(gold_medal_data_tidyverse_joined, select(gold_medal_data_tidy
gold_medal_data_tidyverse_joined <- union(gold_medal_data_tidyverse_joined, select(gold_medal_data_tidy
gold_medal_data_tidyverse_joined <- mutate(gold_medal_data_tidyverse_joined, Year = Year_diff + 1900)
gold_medal_data_tidyverse_joined <- select(gold_medal_data_tidyverse_joined, !Year_diff)
```

After cleaning the Gold Medal dataset using both Base R and tidyverse functions, I am going to display the cleaned data.

Year	Distance
Min. :1896	Min. :249.8
1st Qu.:1921	1st Qu.:295.4
Median :1950	Median :308.1
Mean :1945	Mean :310.3
3rd Qu.:1971	3rd Qu.:327.5
Max. :1992	Max. :350.5



c) Brain weight and body weight data

First, I am going to pull down the dataset and save it to local storage.

```
## brain_body_weight_url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.
## brain_body_weight_data <- fread(brain_body_weight_url, skip = 1, fill = TRUE, data.table = FALSE)
## saveRDS(brain_body_weight_data, "dwnldd_data/brain_body_weight_data_raw.RDS")
brain_body_weight_data <- readRDS("dwnldd_data/brain_body_weight_data_raw.RDS")
```

Similar to the Gold Medal data set, the data is spread across multiple columns, but should be reduced to just two columns. We will initially fix these using Base R functions.

```
brain_body_weight_data_base_r <- brain_body_weight_data
## Create empty data frame to fill with data from untidy data
brain_body_weight_data_frame <- data.frame(BrainWeight=numeric(), BodyWeight=numeric())
## Loop through every other column and iterate through the rows, filling the empty data frame
for (i in seq(from = 1, to = 5, by = 2)) {
  for (j in seq(from = 1, to = 21, by = 1)) {
    data_row <- brain_body_weight_data_base_r[j,]
    body_weight <- data_row[[i]]
    brain_weight <- data_row[[i+1]]
    brain_body_weight_data_frame <- rbind(brain_body_weight_data_frame, c(body_weight, brain_weight))
  }
}
brain_body_weight_data_frame <- na.omit(brain_body_weight_data_frame)
names(brain_body_weight_data_frame) <- c("BodyWeight_kg", "BrainWeight_g")
```

Now we will attempt to clean the data set using tidyverse functions.

```
brain_body_weight_data_tidyverse <- brain_body_weight_data
colnames(brain_body_weight_data_tidyverse) <- c(rep(c("BodyWeight_kg", "BrainWeight_g"), 3))
## Split the data into column pairs and then rejoin as a data frame with two columns
brain_body_weight_data_tidyverse <- union(union(select(brain_body_weight_data_tidyverse, 1:2), select(br
brain_body_weight_data_tidyverse <- brain_body_weight_data_tidyverse %>% filter(!is.na(BodyWeight_kg))
```

After cleaning the dataset using Base R and tidyverse functions, I am going to display the cleaned data.

BodyWeight_kg	BrainWeight_g
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

d) Tomato Yield data

First, I am going to pull down the dataset and save it to local storage.

```
##tomato_yield_url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
##tomato_yield_data <- fread(tomato_yield_url, skip = 1, fill = TRUE, data.table = FALSE)
##saveRDS(tomato_yield_data, "dwnldd_data/tomato_yield_data_raw.RDS")
tomato_yield_data <- readRDS("dwnldd_data/tomato_yield_data_raw.RDS")
```

This data set has data in the first row as column names and has multiple values within one entry. We will fix this untidy data using Base R functions.

```
tomato_yield_data_base_r <- tomato_yield_data
## Create empty data frame to fill with tidy data
tomato_yield_data_frame <- data.frame(TomatoVariety=character(), PlantDensity=character(), Yield=charac
## Loop through the second two colums that contain yield measurements
for (i in seq(from = 2, to = 3)) {
  for (j in seq(from = 2, to = 4)) {
    data_row <- tomato_yield_data_base_r[i,]
    tomato_yields <- data_row[[j]]
    ## Split the yield values by comma
    split_yields <- str_split(tomato_yields, ',')[[1]]
    ## Store plant variety taht should be part of every observation
    tomato_variety <- data_row[[1]]
    ## Store plant density that will be used as a column value instead of a "column name"
    plant_density <- tomato_yield_data_base_r[1,][[j - 1]]
    for (yield in split_yields) {
      ## Account for empty yields that result from a trailing comma in the data
      if (yield != "") {
        tomato_yield_data_frame <- rbind(tomato_yield_data_frame, c(tomato_variety, plant_density, yiel
      }
    }
  }
}
names(tomato_yield_data_frame) <- c("TomatoVariety", "PlantDensity", "Yield")
tomato_yield_data_frame$PlantDensity <- as.integer(tomato_yield_data_frame$PlantDensity)
tomato_yield_data_frame$Yield <- as.numeric(tomato_yield_data_frame$Yield)
```

Now we will clean this dataset using tidyverse functions

```
tomato_yield_data_tidyverse <- tomato_yield_data
colnames(tomato_yield_data_tidyverse) <- c("TomatoVariety", tomato_yield_data_tidyverse[1,1:3])
tomato_yield_data_tidyverse <- tomato_yield_data_tidyverse[-1,]
tomato_yield_data_tidyverse <- tomato_yield_data_tidyverse %>%
  gather(key = "PlantDensity", value = "Yield", 2, 3, 4) %>%
  mutate(Yield_clean = gsub(",*$", "", Yield)) %>%
  select(!Yield) %>%
  separate(Yield_clean, into = c("Yield1", "Yield2", "Yield3"), sep = ",") %>%
  gather(key = "to_remove", value = "Yield", 3, 4, 5) %>%
  select(!to_remove) %>%
  arrange(TomatoVariety, PlantDensity)
```

After cleaning the dataset using Base R and tidyverse functions, I am going to display the cleaned data

TomatoVariety	PlantDensity	Yield
Length:18	Min. :10000	Min. : 8.10
Class :character	1st Qu.:10000	1st Qu.:12.95
Mode :character	Median :20000	Median :15.35

TomatoVariety	PlantDensity	Yield
NA	Mean :20000	Mean :15.07
NA	3rd Qu.:30000	3rd Qu.:17.88
NA	Max. :30000	Max. :21.00