HW4_arflowers

Anna Flowers

10/12/2021

Homework 4

Problem 3

Part A

For this data set, the main issue is that each operator's observations are taking up two columns, when we would prefer a single column for each operator's observations. I am going to assume that the observations are given with each operator's two observations next to each other, but it is not entirely clear if that is how the data is stored, or if it goes through each operator's first observation and then each operator's second observation. My goal is to build a table with 4 column's for the part, operator 1's observations, operator 2's observations, and operator 3's observations.

```
setwd("~/Desktop/VT/StatProgPackages")
ThicknessGauge <- read.csv('ThicknessGauge.dat',
                           sep = " ",
                           header = FALSE)
```

I will start by removing the first two rows (since they should be column descriptors) and renaming the columns to represent which operator took that observation. Then I am able to treat the two observations from each operator as a separate data set and combine them to make just four columns instead of the original seven.

```
tidy_thickness <- slice(ThicknessGauge, -c(1,2))
colnames(tidy_thickness) <- c("Part",
                              "Operator1", "Operator1" ,
                              "Operator2", "Operator2",
                              "Operator3", "Operator3")
tidy_thickness <- bind_rows(tidy_thickness[,c(1,2,4,6)],
                            tidy_thickness[,c(1,3,5,7)])
summary(tidy_thickness)
```

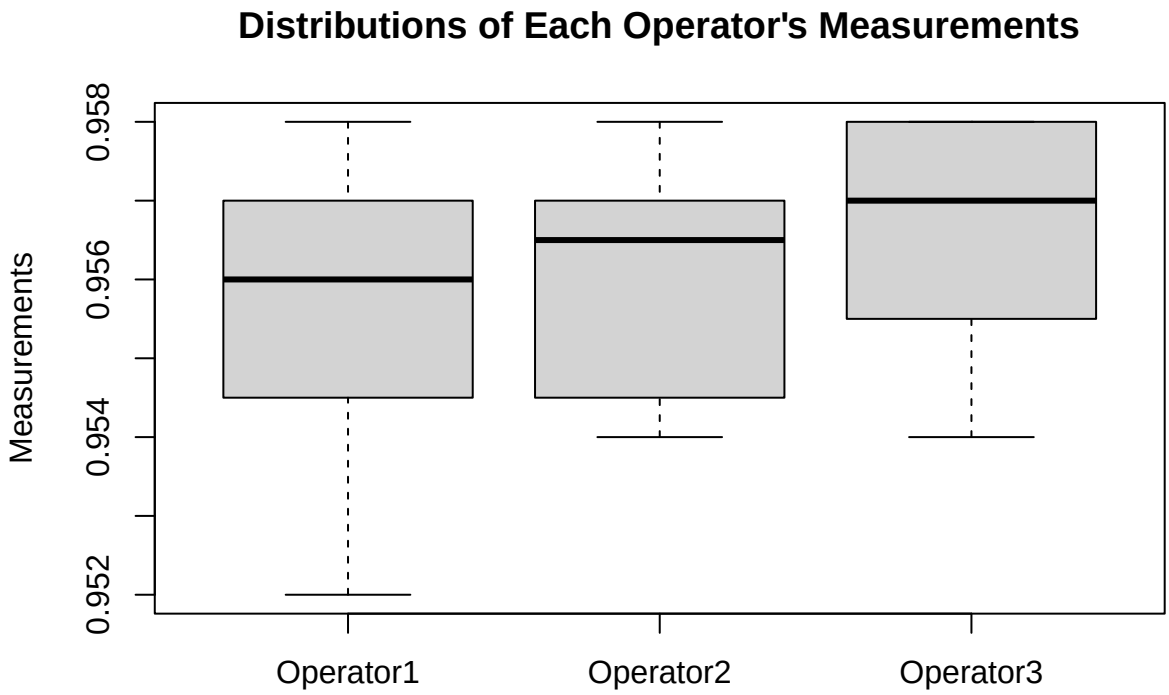
```
##      Part             Operator1        Operator2        Operator3
##  Length:20          Min.   :0.9520   Min.   :0.9540   Min.   :0.9540
##  Class :character   1st Qu.:0.9547   1st Qu.:0.9547   1st Qu.:0.9557
##  Mode  :character   Median :0.9560   Median :0.9565   Median :0.9570
##                     Mean   :0.9557   Mean   :0.9560   Mean   :0.9567
##                     3rd Qu.:0.9570   3rd Qu.:0.9570   3rd Qu.:0.9580
##                     Max.   :0.9580   Max.   :0.9580   Max.   :0.9580
```

```
kable(head(tidy_thickness),
      caption = "Thickness Gauge Data")
```

Table 1: Thickness Gauge Data

Part	Operator1	Operator2	Operator3
1	0.953	0.954	0.954
2	0.956	0.956	0.958
3	0.956	0.956	0.957
4	0.957	0.958	0.957
5	0.957	0.957	0.958
6	0.958	0.957	0.958

```
boxplot(tidy_thickness[,2:4],
        ylab = "Measurements",
        main = "Distributions of Each Operator's Measurements")
```



Part B

There are a few issues I notice with this data set. First, the column names have spaces in them, meaning that their names do not match up correctly with their observations. Additionally, the data is stored across six columns when we are only observing two different variables, so it should just be across two columns. The final issue is that the last two columns are missing an observation in the last row. My goal for this data set is a table with two columns, one for the brain weight and one for the body weight.

```
BrainAndBodyWeight <- read.csv('BrainandBodyWeight.dat',
                               sep = " ")
```

To begin, I subset the table so that we only have the six columns that actually contain the data. I then rename the columns to represent what is being observed, the brain weight or the body weight. That way I can treat the columns as three separate data sets and combine them to make one long data set with two columns. Then, I remove the last row (the missing observation) so that the table is only filled with the actual data.

```
tidy_weight <- BrainAndBodyWeight[,1:6]
colnames(tidy_weight) <- c("BodyWeight", "BrainWeight",
```

```
                           "BodyWeight", "BrainWeight",
                           "BodyWeight", "BrainWeight")
tidy_weight <- bind_rows(tidy_weight[,1:2],
                         tidy_weight[,3:4],
                         tidy_weight[,5:6])
tidy_weight <- slice(tidy_weight, -63)
summary(tidy_weight)
```

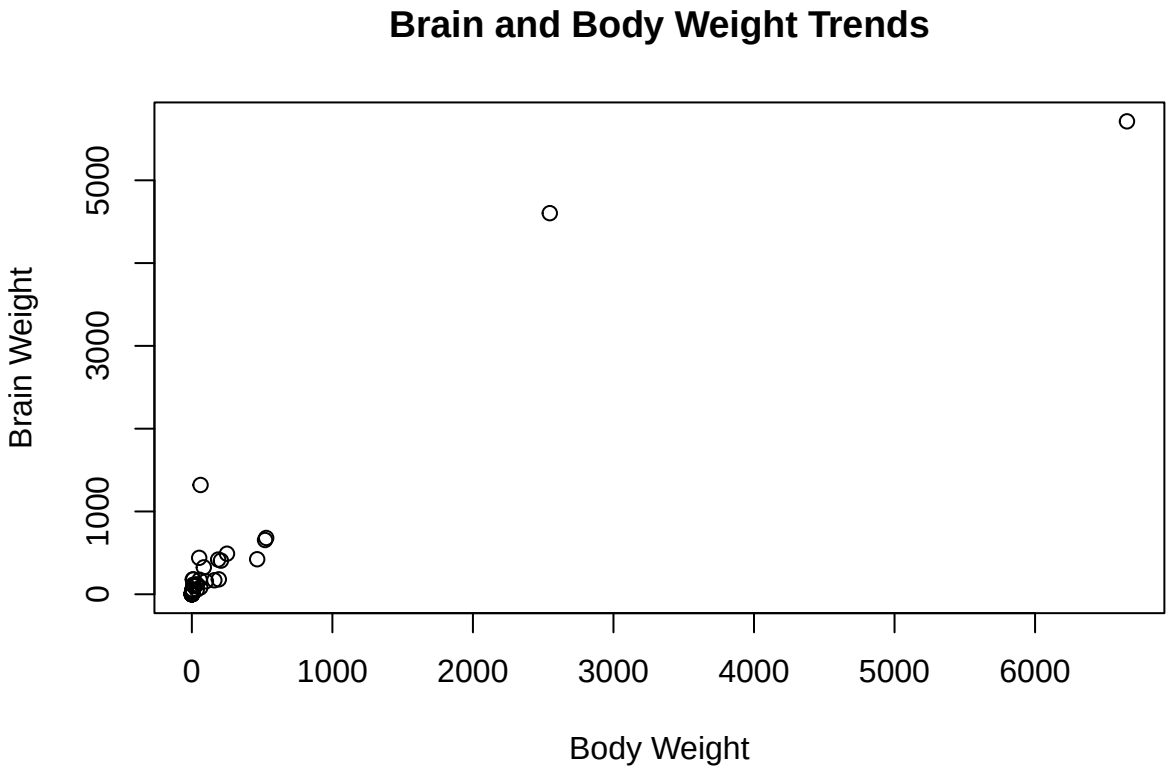
```
##    BodyWeight        BrainWeight
##  Min.   :   0.005   Min.   :   0.10
##  1st Qu.:   0.600   1st Qu.:   4.25
##  Median :   3.342   Median :  17.25
##  Mean   : 198.790   Mean   : 283.13
##  3rd Qu.:  48.202   3rd Qu.: 166.00
##  Max.   :6654.000   Max.   :5712.00
```

```
kable(head(tidy_weight),
      caption = "Brain and Body Weight Data")
```

Table 2: Brain and Body Weight Data

BodyWeight	BrainWeight
3.385	44.5
0.480	15.5
1.350	8.1
465.000	423.0
36.330	119.5
27.660	115.0

```
plot(tidy_weight$BodyWeight, tidy_weight$BrainWeight,
     xlab = "Body Weight",
     ylab = "Brain Weight",
     main = "Brain and Body Weight Trends")
```



Part C

The issues in this data set are very similar to the ones present in the brain and body weight data set from part b. The only extra issue in this one is the way that the year variable is stored: we want it to show the actual year instead of the code.

```
LongJump <- fread('LongJumpData.dat',
                  header = FALSE,
                  fill = TRUE)
```

Much of the cleaning I have done here is similar to previous parts. I remove the first row (since that would have been the column names) and then subset to only include columns that actually contain data. I rename the columns and then combine the four "smaller data sets" to make two columns for the year and the long jump record. I then remove the last two rows since there is no data.

```
tidy_jump <- slice(LongJump, -1)
tidy_jump <- tidy_jump[,1:8]
colnames(tidy_jump) <- c("Year", "LongJump",
                         "Year", "LongJump",
                         "Year", "LongJump",
                         "Year", "LongJump")
tidy_jump <- bind_rows(tidy_jump[,1:2],
                       tidy_jump[,3:4],
                       tidy_jump[,5:6],
                       tidy_jump[,7:8])
tidy_jump <- slice(tidy_jump, -c(23,24))
```

For this part, I noticed that although we are seeing numbers, R is storing them as character vectors. To take summaries or compute mathematical operations, they must be stored as integers. So I changed all observations to a numerical format, and then added 1900 to the year variable to offset the coded years in the original data set.

```
tidy_jump <- as.data.frame(apply(tidy_jump, c(1,2), as.numeric))
tidy_jump$Year <- tidy_jump$Year + 1900
summary(tidy_jump)
```

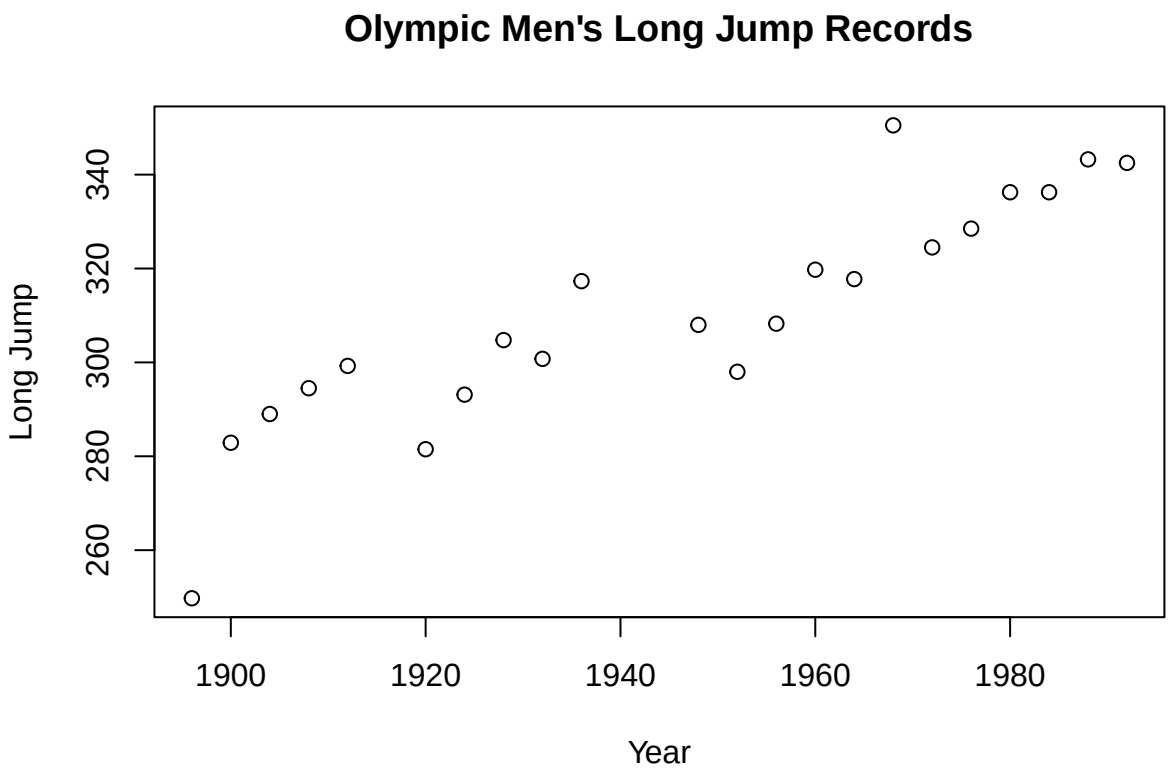
```
##       Year         LongJump
##  Min.   :1896   Min.   :249.8
##  1st Qu.:1921   1st Qu.:295.4
##  Median :1950   Median :308.1
##  Mean   :1945   Mean   :310.3
##  3rd Qu.:1971   3rd Qu.:327.5
##  Max.   :1992   Max.   :350.5
```

```
kable(head(tidy_jump),
      caption = "Olympic Long Jump Data")
```

Table 3: Olympic Long Jump Data

Year	LongJump
1896	249.75
1900	282.88
1904	289.00
1908	294.50
1912	299.25
1920	281.50

```
plot(tidy_jump$Year, tidy_jump$LongJump,
     xlab = "Year",
     ylab = "Long Jump",
     main = "Olympic Men's Long Jump Records")
```



Part D

This data set has by far the most issues. The data set has been separated by a combination of spaces and commas, making it difficult to import cleanly into R. Additionally, it is storing variables in a mixture between columns and rows. Ideally, we would have a data set with three columns: one with the tomato variety, one with the yield amount, and one with the density.

```
tomato <- fread('tomato.dat',
                header = FALSE)
```

Each density has been imported in as a separate variable, with each triplicate observation in the same block separated by commas. These observations must be separated into their own columns before we can continue cleaning the data.

```
tidy_tomato <- separate(tomato,
                        "V2",
                        into = c("x1", "x2", "x3"),
                        sep = ",")
```

```
## Warning: Expected 3 pieces. Additional pieces discarded in 1 rows [2].
```

```
tidy_tomato <- separate(tidy_tomato,
                        "V3",
                        into = c("y1", "y2", "y3"),
                        sep = ",")
tidy_tomato <- separate(tidy_tomato,
                        "V4",
                        into = c("z1", "z2", "z3"),
                        sep = ",")
```

Similar to previous data sets, I am renaming the columns based on some feature of the data. This time I am using the densities as the column names and then binding each "smaller data set" together.

```
colnames(tidy_tomato) <- c("Variety",
                           "1000", "1000", "1000",
                           "2000", "2000", "2000",
                           "3000", "3000", "3000")
tidy_tomato <- bind_rows(tidy_tomato[,c(1,2,5,8)],
                         tidy_tomato[,c(1,3,6,9)],
                         tidy_tomato[,c(1,4,7,10)])
```

The densities are still being stored in different columns, when we should have a single column with the density observed. I update the data set so that the density is stored correctly. Similar to earlier data sets, the yield observations are stored as characters and must be numeric, so I change them to a numeric format so that summaries can be taken.

```
tidy_tomato <- gather(tidy_tomato,
                      key = "Density",
                      value = "Yield",
                      `1000`, `2000`, `3000`)
tidy_tomato$Yield <- as.numeric(tidy_tomato$Yield)
summary(tidy_tomato)
```

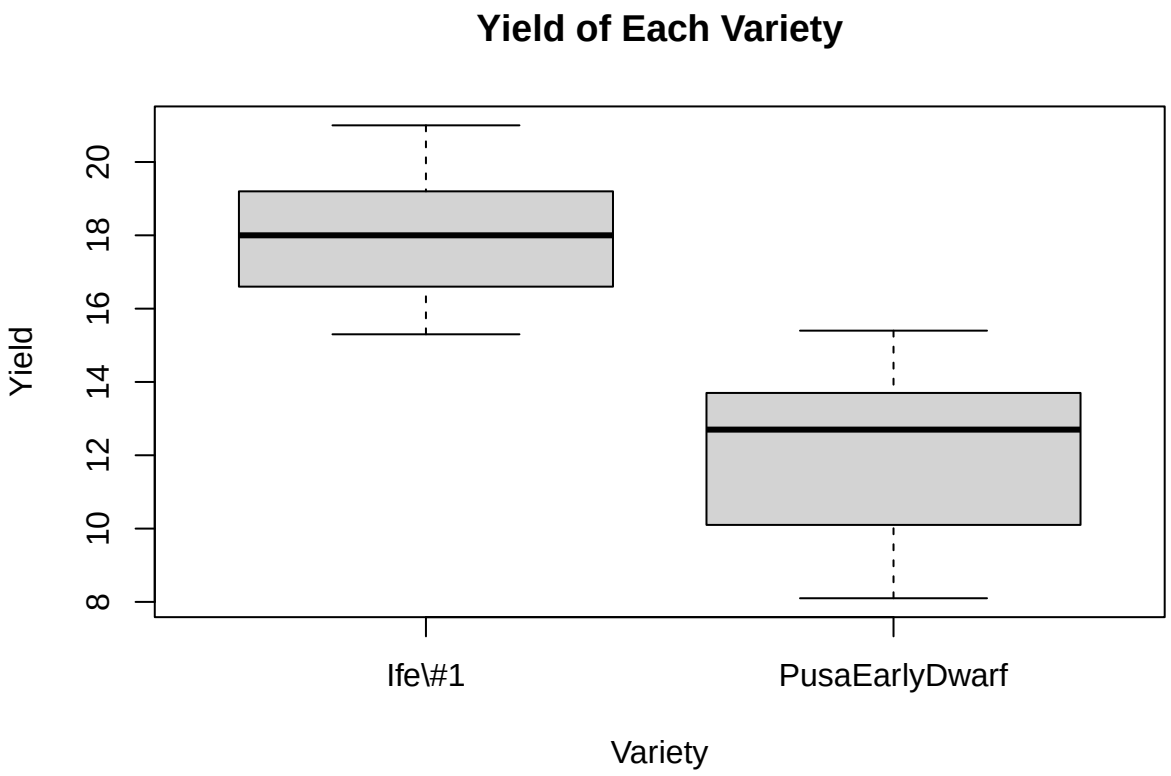
```
##    Variety            Density              Yield
##  Length:18          Length:18          Min.   : 8.10
##  Class :character   Class :character   1st Qu.:12.95
##  Mode  :character   Mode  :character   Median :15.35
##                                        Mean   :15.07
##                                        3rd Qu.:17.88
##                                        Max.   :21.00
```

```
kable(head(tidy_tomato), caption = "Tomato Yields")
```

Table 4: Tomato Yields

Variety	Density	Yield
Ife#1	1000	16.1
PusaEarlyDwarf	1000	8.1
Ife#1	1000	15.3
PusaEarlyDwarf	1000	8.6
Ife#1	1000	17.5
PusaEarlyDwarf	1000	10.1

```
boxplot(Yield ~ Variety,
        tidy_tomato,
        main = "Yield of Each Variety")
```



Part E

This data set again has issues with storing information in separate columns that should be in a single column with observations. Both the age and treatment should be variables with their observed values in that column, rather than being stored in several different columns. My goal for this data set is to have four columns for the block, age, treatment, and count.

```
LarvaeControl <- fread('LarvaeControl.dat')
```

I follow a similar process from many of the previous data sets with renaming the columns and then binding the "smaller data sets" together. But before I continue, I must make sure to account for the age variable, since it was not included when uploading the data. Since the data is ordered, this can be done by adding a vector of 1's and 2's to the data set. Then, I gather the separate treatment variables into one column, where the observation is the treatment given. This gives the final data set.

```
colnames(LarvaeControl) <- c("Block",
                             "1", "2", "3", "4", "5",
                             "1", "2", "3", "4", "5")
tidy_larvae <- bind_rows(LarvaeControl[,c(1,2:6)], LarvaeControl[,c(1,7:11)])
tidy_larvae <- mutate(tidy_larvae,
                      Age = c(rep(1,8), rep(2,8)))
tidy_larvae <- gather(tidy_larvae,
                      key = "Treatment",
                      value = "Counts",
                      `1`, `2`, `3`, `4`, `5`)
summary(tidy_larvae)
```

```
##      Block           Age       Treatment            Counts
##  Min.   :1.00   Min.   :1.0   Length:80          Min.   : 0.00
##  1st Qu.:2.75   1st Qu.:1.0   Class :character   1st Qu.: 2.75
##  Median :4.50   Median :1.5   Mode  :character   Median : 5.50
```

```
##  Mean   :4.50   Mean   :1.5                    Mean   :10.50
##  3rd Qu.:6.25   3rd Qu.:2.0                    3rd Qu.:13.00
##  Max.   :8.00   Max.   :2.0                    Max.   :61.00
```

```
kable(head(tidy_larvae), caption = "Larvae Counts")
```

Table 5: Larvae Counts

Block	Age	Treatment	Counts
1	1	1	13
2	1	1	29
3	1	1	5
4	1	1	5
5	1	1	0
6	1	1	1

```
boxplot(Counts ~ Age,
        tidy_larvae,
        main = "Counts of Larvae at Two Ages")
```