HW3_dsteberg1vt

Dylan Steberg

2023-10-25

Firstly I loaded the tidyverse package which installs all the libraries needed for this data cleaning assignment. I also loaded the knitr and ggplot2 packages which were used to make tables and graphs respectively.

Part A)

First I loaded in the data. This data includes measurements of ten parts taken by three different operators of the measurement apparatus. The three operators measured each part twice, thus there are sixty total measurements. As the data is read in from a table, the first two rows contained names that were pretty much worthless. I skipped these rows (by including `skip = 2` in our `read.table()` function) and addressed the column names during the data cleaning.

```
url_a = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/ThicknessGauge.dat"
data_a_raw = read.table(file = url_a, fill = TRUE, skip = 2)
```

The first step is to address the column names which is done using the function `rename()`. Aside from the first column which is the part, the columns were renamed using the outline "operator.measure". For example, the column containing the first measure by the first operator was named "1.1". All the measurements were then stacked in one column and the operator.measure, which were the column headers, were stacked in another column. This column was then separated into two columns, one for the operator, and one for the measure. The final columns include "Part" which is the part number, "Operator" which is the operator that took the measurement, "Measure", which denotes if it is the first or second measurement by the operator, and "Measurement" which is the value of measurement after subtracting off a nominal dimension of 685mm.

```
data_a_clean = (data_a_raw
  %>% rename("Part" = V1,
             "1.1" = V2,
             "1.2" = V3,
             "2.1" = V4,
             "2.2" = V5,
             "3.1" = V6,
             "3.2" = V7
            )
  %>% pivot_longer("1.1":"3.2", names_to = "Operator.Measure", values_to = "Measurement")
  %>% separate_wider_delim(Operator.Measure, delim = ".", names = c("Operator", "Measure"))
)
```

A look at the first few rows of the clean data set.

```
(kable(head(data_a_clean), caption = "Thinkness gauge data"))
```

Table 1: Thinkness gauge data

Part	Operator	Measure	Measurement
1	1	1	0.953
1	1	2	0.952
1	2	1	0.954
1	2	2	0.954
1	3	1	0.954
1	3	2	0.956

Finally, I created a summary of the data (only a head is shown) and created a plot. This summary looks at the average measurement for each part by the three operators. Graphing this shows that operator 1 tended to have a lower average measurement while operator three tended to have a higher average measurement.

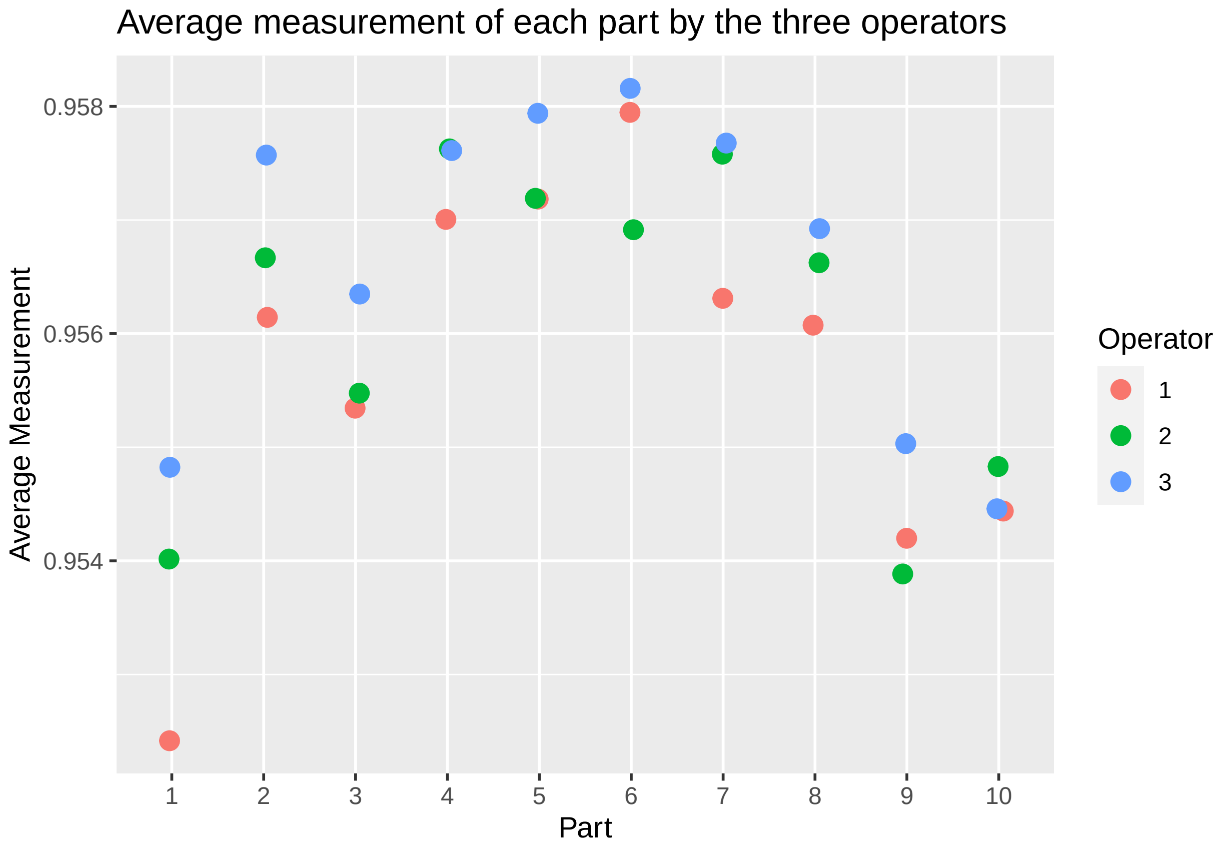
```
summary_a = (data_a_clean
  %>% mutate(Part = as.factor(Part))
  %>% group_by(Part, Operator)
  %>% summarise("Average Measurement" = mean(Measurement), .groups = "rowwise")
)

(kable(head(summary_a), caption = "Average measurement of each part by the three operators"))
```

Table 2: Average measurement of each part by the three operators

Part	Operator	Average Measurement
1	1	0.9525
1	2	0.9540
1	3	0.9550
2	1	0.9560
2	2	0.9565
2	3	0.9575

```
ggplot(data = summary_a) + geom_jitter(aes(x = Part, y = `Average Measurement`, col = Operator), width =
  labs(title = "Average measurement of each part by the three operators")
```



Part B)

First I loaded in the data. This data includes brain weight (in grams) and body weight (in kilograms) of sixty-two unidentified species of animals resulting in 124 total measurements As the data is read in from a table, the first row contained column names that were formatted weird. I instead skipped this row (by including `skip = 1` in our `read.table()` function) and addressed the column names during the data cleaning.

```
url_b = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
data_b_raw = read.table(file = url_b, fill = TRUE, skip = 1)
```

The first thing I did was stack the data, it was originally in six columns with default names. I addressed the names using a mutate after the stack. Columns one, three, and five contain body weight measurements and columns two, four, and six contain brain weight measurements. The measurements were stacked into one column and the column names were stacked into another. Then using a mutate I changed the default column names to be "Body Wt" and "Brain Wt". I then added an "Animal column to make sure we can pair brain weight and body weight of each species of animals together. Finally, since the last two columns of the original data table only had twenty measurements (compared to twenty-one in the other columns) I removed the entries which had NA values that resulted from stacking the stacking. The final columns include"Animal" which has the values one through sixty-two each referring to one of the unidentified species of animals, "Type of Measurement" which tells if each measurement is either a brain weight or a body weight, and "Measurement" which is the value of the measurement.

```
data_b_clean = (data_b_raw
  %>% pivot_longer(1:6, names_to = "Type_of_Measurement", values_to = "Measurement")
  %>% mutate("Type of Measurement" = ifelse(Type_of_Measurement %in% c("V1", "V3", "V5"), "Body Wt", "Br
  %>% mutate(Animal = rep(1:63, rep_len(2, 63)))
  %>% drop_na(Measurement)
  %>% select(Animal, "Type of Measurement", Measurement)
)
```

A look at the first few rows of the clean data set.

```
(kable(head(data_b_clean), caption = "Brain and body weight data"))
```

Table 3: Brain and body weight data

Animal	Type of Measurement	Measurement
1	Body Wt	3.385
1	Brain Wt	44.500
2	Body Wt	521.000
2	Brain Wt	655.000
3	Body Wt	2.500
3	Brain Wt	12.100

Finally, I created a summary of the data (only a head is shown) and created a plot. This summary looks at the ratio of each animal's brain weight to their body weight. Looking at a histogram of this shows that most animals tend to have a small brain weight to body weight ratio. We do see that a few animals have a larger ratio. I wonder which animal has the largest ratio as it is all alone above the other animals.

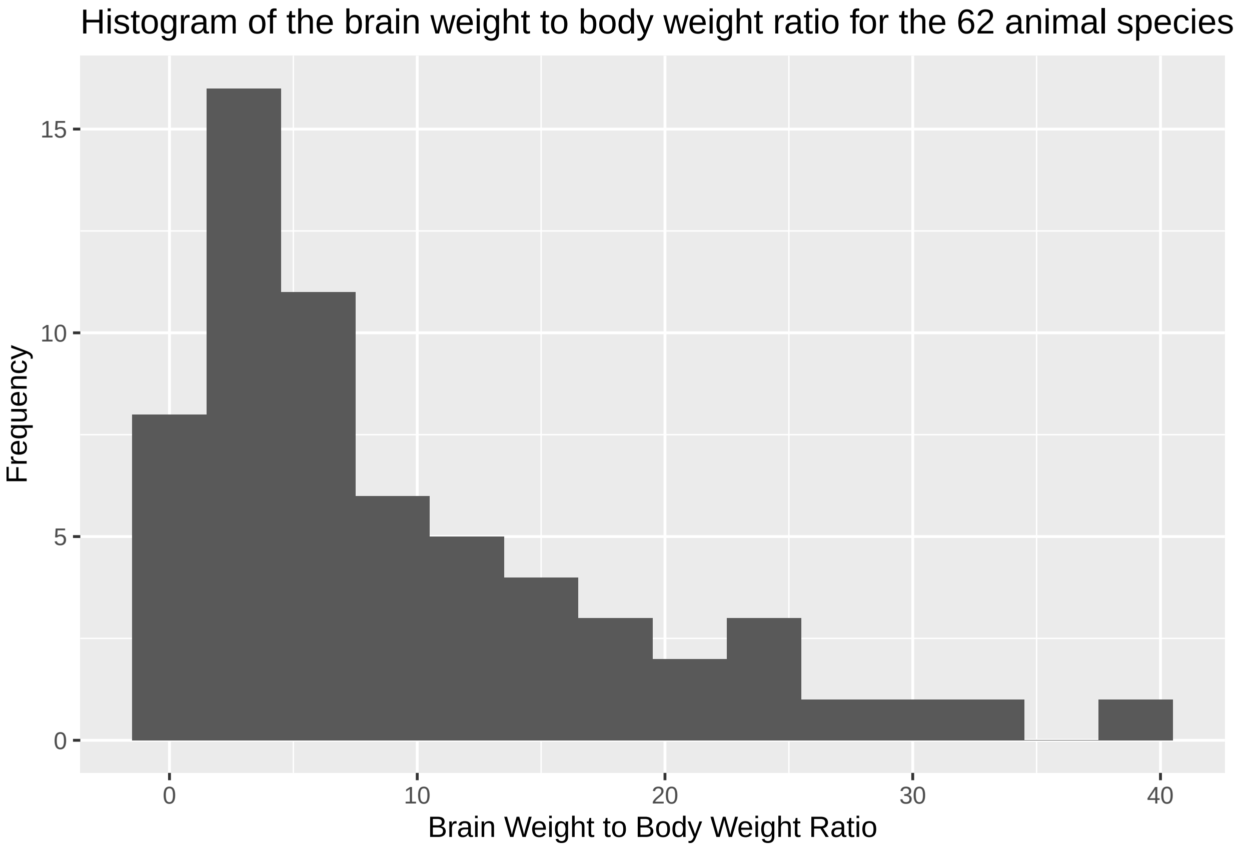
```
summary_b = (data_b_clean
  %>% pivot_wider(names_from = "Type of Measurement", values_from = "Measurement")
  %>% mutate(Brain_to_Body_Ratio = `Brain Wt`/`Body Wt`)
  %>% select(Animal, Brain_to_Body_Ratio)
)

(kable(head(summary_b), caption = "Brain weight to body weight ratio of 62 animal species"))
```

Table 4: Brain weight to body weight ratio of 62 animal species

Animal	Brain_to_Body_Ratio
1	13.146233
2	1.257198
3	4.840000
4	32.291667
5	4.458599
6	3.153153

```
ggplot(data = summary_b) + geom_histogram(aes(x = Brain_to_Body_Ratio), binwidth = 3) +
  labs(title = "Histogram of the brain weight to body weight ratio for the 62 animal species") +
       xlab("Brain Weight to Body Weight Ratio") + ylab("Frequency")
```



Part C)

First I loaded in the data. This data includes data on the men's gold medal performance in the long jump (in inches) at the Olympics which are held every four years (except the years where there were no Olympics due to war) from 1896 to 1992. The year 1900 was encoded as 0 in the data set. Again, I skipped the first row of weirdly formatted names and addressed the issue during the data cleaning.

```
url_c = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
data_c_raw = read.table(file = url_c, fill = TRUE, skip = 1)
```

The first thing I did was stack the years together and then stacked long jump distances together. After each stack I created an id column which was used to join the stacked year and stacked long jump distance data together in the next step. Once I joined them together I removed the columns that were not needed, un-coded the year values, and arranged the table in chronological order. Finally, I removed NA values that were a result of the stacking. The final columns include "Year" which is the year of that specific Olympic games and "Long Jump" which is the measurement in inches of the gold medal performance in the men's long jump.

```
data_c_clean = (data_c_raw
  %>% pivot_longer(c(V1, V3, V5, V7), names_to = "names", values_to = "Year")
  %>% mutate(id = 1:n())
  %>% inner_join(y = (data_c_raw
                      %>% pivot_longer(c(V2, V4, V6, V8), names_to = "names", values_to = "Long Jump"
                      %>% mutate(id = 1:n())
                     ),
                 by = "id",
                 keep = FALSE
                )
  %>% select(Year, "Long Jump")
  %>% mutate(Year = Year + 1900)
  %>% arrange(Year)
  %>% drop_na(Year)
)
```

A look at the first few rows of the clean data set.

```
(kable(head(data_c_clean), caption = "Long jump data"))
```

Table 5: Long jump data

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

Finally, I created a summary of the data and created a plot. This summary looks at the maximum, median, minimum, and mean long jump performance from this data. The graph plots a smoother through the data. We can see from the smoother that the long jump performance has increased since the first Olympics but there was a period where the increase was less. That is right around WWII when the Olympics were not held so that may have some effect as well.

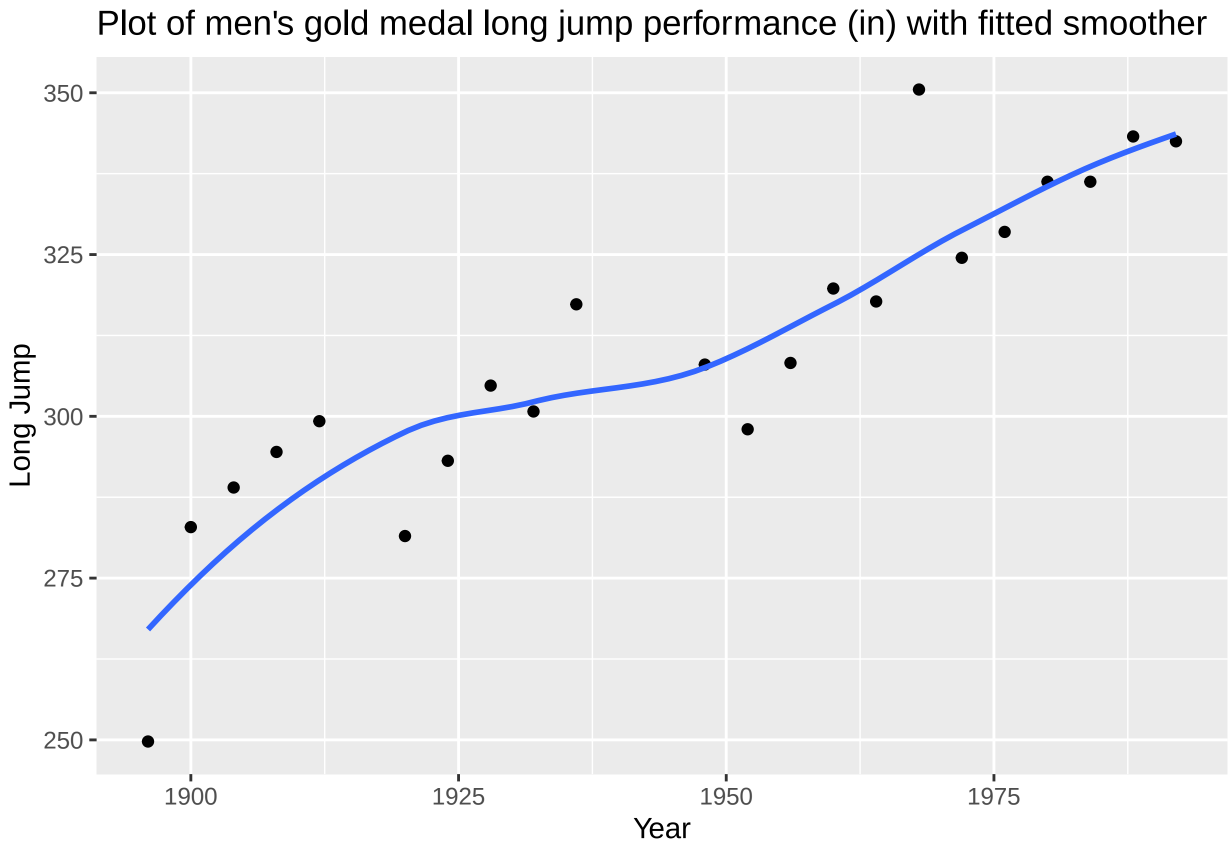
```
summary_c = (data_c_clean
  %>% summarize(Min = min(`Long Jump`),
                Median = median(`Long Jump`),
                Max = max(`Long Jump`),
                Mean = mean(`Long Jump`),
                .groups = "rowwise")
)

(kable(summary_c, caption = "Min, median, max, and mean gold medal men's long jump performance"))
```

Table 6: Min, median, max, and mean gold medal men's long jump performance

Min	Median	Max	Mean
249.75	308.125	350.5	310.2873

```r
ggplot(data = data_c_clean) + geom_point(aes(x = Year, y = `Long Jump`)) +
  geom_smooth(aes(x = Year, y = `Long Jump`), method = "loess", formula = y ~ x, se = FALSE) +
  labs(title = "Plot of men's gold medal long jump performance (in) with fitted smoother")
```



Part D)

First I loaded in the data. This data comes from an experiment to study the effect of variety and plant density on tomato yield. The data includes to varieties, ife #1 and Pusa Early Dwarf, and three plant densities, 10,000, 20,000, and 30,000. The study used three replicates for a total of eighteen measurements. Again, I skipped the first two rows of weirdly formatted names and addressed the issue during the data cleaning.

```r
url_d = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
data_d_raw = read.table(file = url_d, comment.char = "", fill = TRUE, skip = 2)
```

The first thing I did was separate the values all three replicates were squished into one cell. The names were changed to reflect the density and replicate the yield was from similar to how I did it in part A. For example, "10.1" corresponds to a density of 10,000 and the first replicate. After separating those values I stacked them together to have a data frame of the variety, yield and density/replicate. I then split the density.replicate into two columns for density and replicate, changed a column name, and multiplied the plant density by 1,000 to get the correct value. The final columns include, "Variety" which is the variety of tomato plant, "Density" which is the plant density, "Replicate" which denotes the replicate, and Yield which represents the tomato yield in tons per hectare.

```
data_d_clean = (data_d_raw
  %>% separate_wider_delim(V2, delim = ",", too_many = "drop", names = c("10.1", "10.2", "10.3"))
  %>% separate_wider_delim(V3, delim = ",", names = c("20.1", "20.2", "20.3"))
  %>% separate_wider_delim(V4, delim = ",", names = c("30.1", "30.2", "30.3"))
  %>% pivot_longer("10.1":"30.3", names_to = "Density.Replicate", values_to = "Yield")
  %>% separate_wider_delim(Density.Replicate, delim = ".", names = c("Density", "Replicate"))
  %>% mutate(Variety = ifelse(V1 == "PusaEarlyDwarf", "Pusa Early Dwarf", "Ife #1"))
  %>% mutate("Plant Density" = as.numeric(Density)*1000)
  %>% select(Variety, "Plant Density", Replicate, Yield)
)
```

A look at the first few rows of the clean data set.

```
(kable(head(data_d_clean), caption = "Yield data for the tomato experiment"))
```

Table 7: Yield data for the tomato experiment

Variety	Plant Density	Replicate	Yield
Ife #1	10000	1	16.1
Ife #1	10000	2	15.3
Ife #1	10000	3	17.5
Ife #1	20000	1	16.6
Ife #1	20000	2	19.2
Ife #1	20000	3	18.5

Finally, I created a summary of the data and created a plot. This summary looks at the average tomato yield by variety and plant density. Graphing this shows that on average a higher density gives a higher yield for each variety and on average the ife #1 variety provides a higher yield than the Pusa Early Dwarf variety.

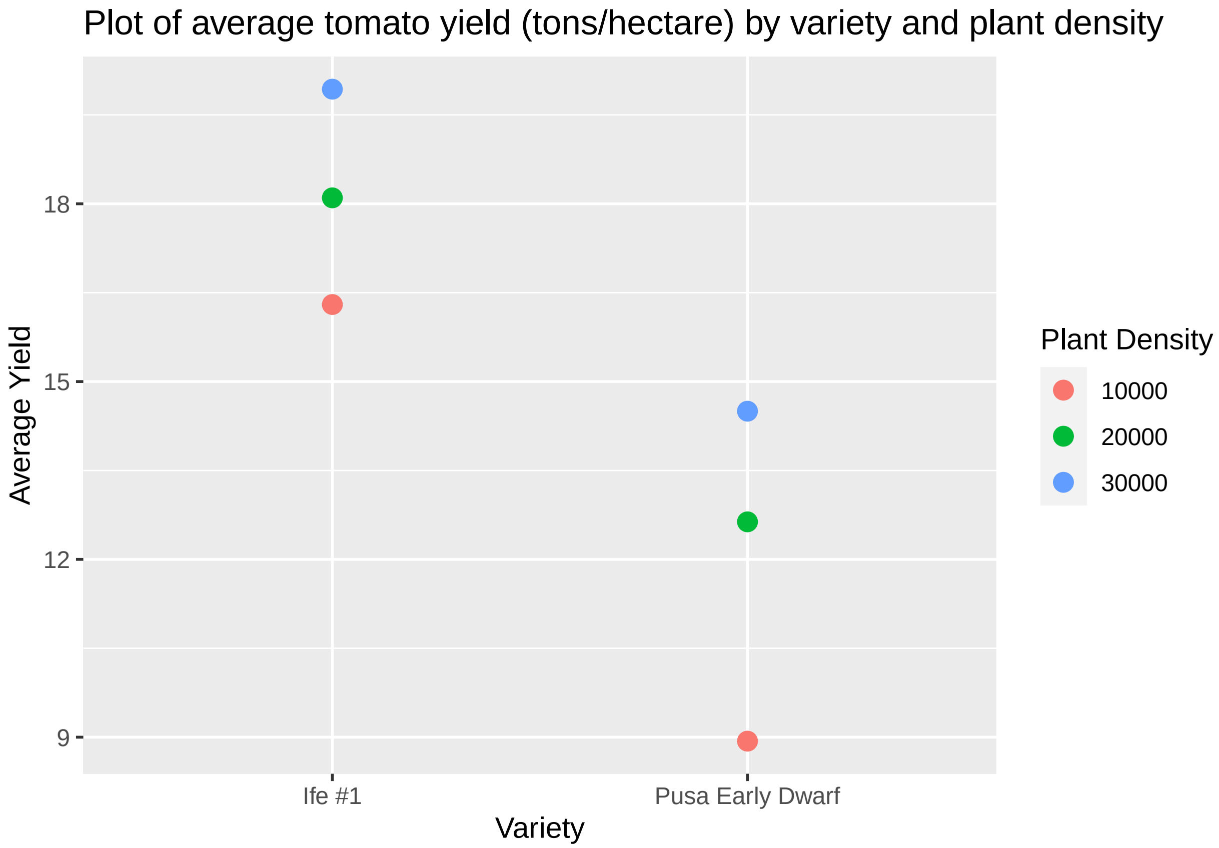
```
summary_d = (data_d_clean
             %>% mutate(Yield = as.numeric(Yield))
             %>% mutate(`Plant Density`= as.factor(`Plant Density`))
             %>% group_by(Variety, `Plant Density`)
             %>% summarise("Average Yield" = mean(Yield), .groups = "rowwise")
)

(kable(summary_d, caption = "Average tomato yield (tons/hectare) by variety and plant density"))
```

Table 8: Average tomato yield (tons/hectare) by variety and plant density

Variety	Plant Density	Average Yield
Ife #1	10000	16.300000
Ife #1	20000	18.100000
Ife #1	30000	19.933333
Pusa Early Dwarf	10000	8.933333
Pusa Early Dwarf	20000	12.633333
Pusa Early Dwarf	30000	14.500000

```
ggplot(data = summary_d) + geom_point(aes(x = Variety, y = `Average Yield`, col = `Plant Density`), size
  labs(title = "Plot of average tomato yield (tons/hectare) by variety and plant density")
```



Part E)

First I loaded in the data. This data comes from an experiment where five treatments for cockchafer larvae were studied using a randomized block design with eight blocks. The response is the number of larvae classified into two age classes for the five treatments. Again, I skipped the first three rows of weirdly formatted names and addressed the issue during the data cleaning.

```
url_e = "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LarvaeControl.dat"
data_e_raw = read.table(file = url_e, fill = TRUE, skip = 3)
```

This data was a little more complicated to clean as much of the information was contained in the headers. I essentially broke this down into two problems and used a join. The first problem took the columns that corresponded to treatments one through five for the Age1 response and stacked them together. Then I did the same thing with the Age2 response. Before I joined the data I changed the names of the treatments to be correct. Then when I had the joined data I stacked it again to get all the information out of the column headers. The final columns include, "Block" which represents the block, "Treatment" which represents the treatment, "Response Group" which indicates if the response was for Age1 or Age2, and the Response which is the number of larvae counted in each plot.

```
data_e_clean = (data_e_raw
  %>% select(V1:V6)
  %>% pivot_longer(V2:V6, names_to = "Treatment", values_to = "Age 1")
  %>% mutate(Treatment = case_when(Treatment == "V2" ~ "1",
                                   Treatment == "V3" ~ "2",
                                   Treatment == "V4" ~ "3",
                                   Treatment == "V5" ~ "4",
                                   Treatment == "V6" ~ "5"))
  %>% inner_join(y = (data_e_raw
                      %>% select(V1, V7:V11)
                      %>% pivot_longer(V7:V11, names_to = "Treatment", values_to = "Age 2")
                      %>% mutate(Treatment = case_when(Treatment == "V7" ~ "1",
                                                       Treatment == "V8" ~ "2",
                                                       Treatment == "V9" ~ "3",
                                                       Treatment == "V10" ~ "4",
                                                       Treatment == "V11" ~ "5"))
                    ),
                 by = join_by(V1, Treatment == Treatment),
                 keep = FALSE
                )
  %>% pivot_longer(c("Age 1", "Age 2"), names_to = "Response Group", values_to = "Response")
  %>% rename("Block" = V1)
)
```

A look at the first few rows of the clean data set.

```
(kable(head(data_e_clean), caption = "Larvae control experiment data"))
```

Table 9: Larvae control experiment data

Block	Treatment	Response Group	Response
1	1	Age 1	13
1	1	Age 2	28
1	2	Age 1	16
1	2	Age 2	12
1	3	Age 1	13
1	3	Age 2	40

Finally, I created a summary of the data and created a plot. This summary looks at the average response by response group and treatment. Graphing this shows that the average response was higher for the Age2 group. It seems like within each response group that the average response was lower for treatment five and higher for treatment one.

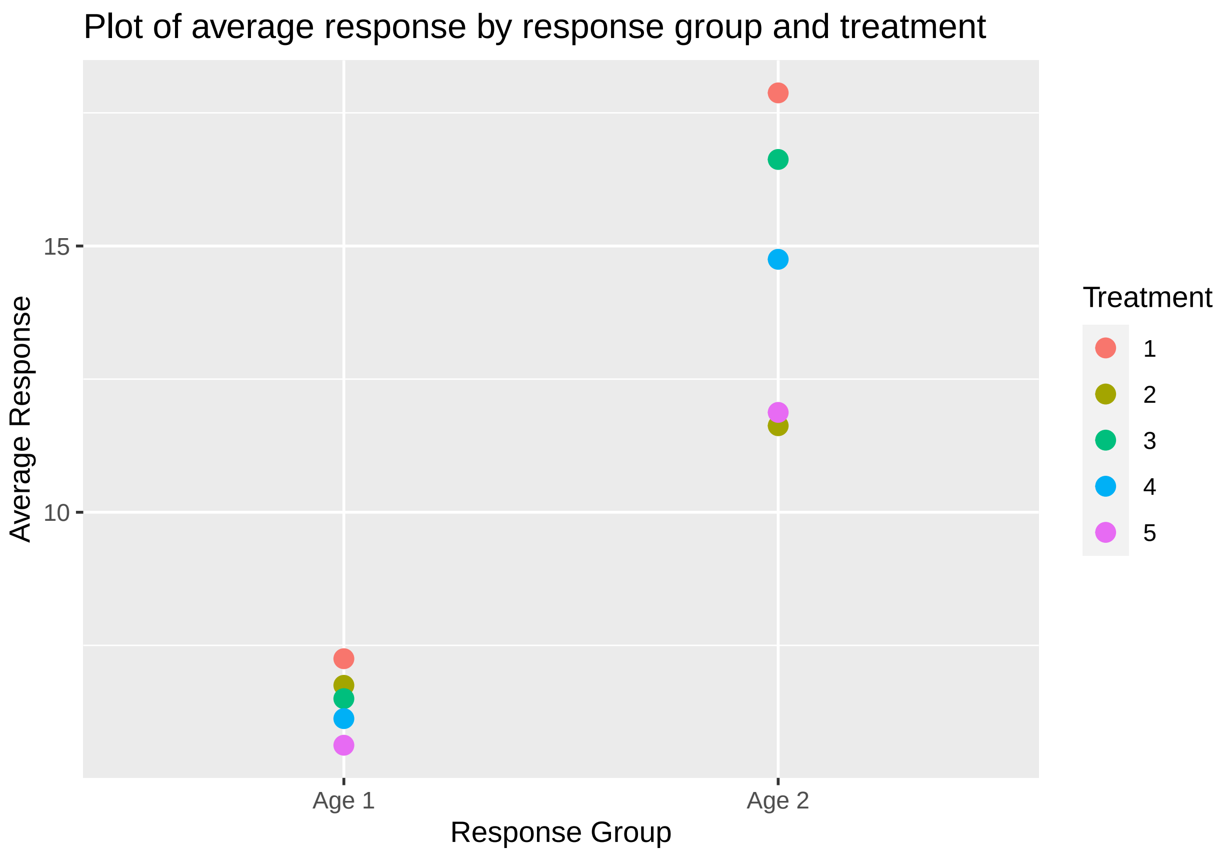
```
summary_e = (data_e_clean
             %>% mutate(Block = as.factor(Block))
             %>% group_by(`Treatment`, `Response Group`)
             %>% summarise("Average Response" = mean(Response), .groups = "rowwise")
)

(kable(summary_e, caption = "Average response by response group and treatment"))
```

Table 10: Average response by response group and treatment

Treatment	Response Group	Average Response
1	Age 1	7.250
1	Age 2	17.875
2	Age 1	6.750
2	Age 2	11.625
3	Age 1	6.500
3	Age 2	16.625
4	Age 1	6.125
4	Age 2	14.750
5	Age 1	5.625
5	Age 2	11.875

```
ggplot(data = summary_e) + geom_point(aes(x = `Response Group`, y = `Average Response`, col = `Treatmen
  labs(title = "Plot of average response by response group and treatment")
```


Plot of average response by response group and treatment
Average Response
15
10
Age 1
Age 2
Response Group
Treatment
1
2
3
4
5