HW2_wgeither

Warren Geither

9/7/2020

Problem 3

In terms of team collaboration, version control seems like by far the best way to go about a project way more control and systematic then going back and forth through emails. I've already started using github to back up work for other classes, so i think it is a great tool.

Problem 4

a. The data is of sensory data from different operators. It currently has 2 columns and 31 rows. As we tidy it will end with 3 columns and 150 rows. Item column, value, and operator. It has data on 10 different items with 5 different operators.

```
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"

#sensory_data <- fread(url, fill=TRUE, header=TRUE)
#saveRDS(sensory_data, "sensory_data_raw.RDS")
sensory_df <- readRDS("sensory_data_raw.RDS")
```

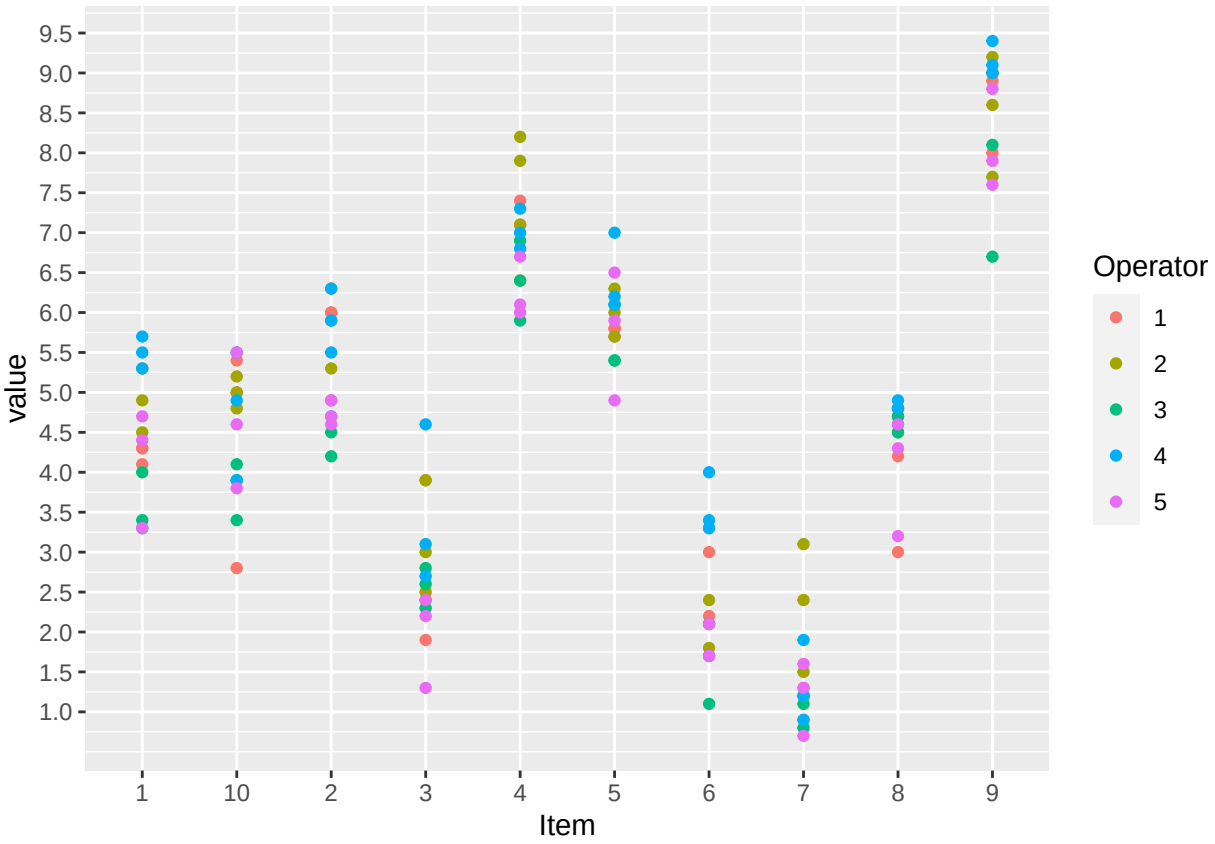
First we'll clean it using base R.

```
# delete first row
base_r_sensory_df <- sensory_df[2:nrow(sensory_df)]

# delete na column
base_r_sensory_df <- base_r_sensory_df[, 1]

# split up values into multiple columns

# create a function so we can just take the right values
right = function(x,n){
  substring(x,nchar(x)-n+1)
}

string_list <- strsplit(right(as.character(base_r_sensory_df$V1),19),' ')

base_r_sensory_df <- data.frame(do.call(rbind, string_list))

# rename columns
colnames(base_r_sensory_df) <- c("Op_1", "Op_2", "Op_3", "Op_4", "Op_5")

# create dataframe of item numbers
item_df <- data.frame("Item" = c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6,7,7,7,8,8,8,9,9,9,10,10,10))
```

```r
# bind item numbers on our other dataframe
base_r_sensory_df <- cbind(base_r_sensory_df,item_df)

# re-order columns
base_r_sensory_df <- base_r_sensory_df[,c(6,1,2,3,4,5)]

#partition new columns so we can stack them, and create an operator column

partition_df_1 <- as.data.frame(cbind(Item=base_r_sensory_df[,1],
                                      value=base_r_sensory_df[,2],
                                      Operator = c(rep(1,30))))
partition_df_2 <- as.data.frame(cbind(Item=base_r_sensory_df[,1],
                                      value=base_r_sensory_df[,3],
                                      Operator = c(rep(2,30))))
partition_df_3 <- as.data.frame(cbind(Item=base_r_sensory_df[,1],
                                      value=base_r_sensory_df[,4],
                                      Operator = c(rep(3,30))))
partition_df_4 <- as.data.frame(cbind(Item=base_r_sensory_df[,1],
                                      value=base_r_sensory_df[,5],
                                      Operator = c(rep(4,30))))
partition_df_5 <- as.data.frame(cbind(Item=base_r_sensory_df[,1],
                                      value=base_r_sensory_df[,6],
                                      Operator = c(rep(5,30))))

tidy_base_r_sensory_df <- rbind(partition_df_1,partition_df_2,partition_df_3,partition_df_4,partition_d

#  Show summary of tidy data table
knitr::kable(tidy_base_r_sensory_df[1:5,])
```

Item	value	Operator
1	4.3	1
1	4.3	1
1	4.1	1
2	6.0	1
2	4.9	1

```r
#informative plot

# change data column class to numeric so we can plot
tidy_base_r_sensory_df[,2] <- as.numeric(tidy_base_r_sensory_df[,2])

# plot data
ggplot(tidy_base_r_sensory_df, aes(x=Item, y=value, color=Operator)) +
  scale_y_continuous(breaks = seq(1, 10, by = .5)) +
  geom_point()
```



Now we'll do the same operations using tidyverse functions

```
# remove first row
tidyverse_sensory_df <- slice(sensory_df[2:31])

# Remove column full of na's
tidyverse_sensory_df <- select(tidyverse_sensory_df, -Operator)

# seperate values
tidyverse_sensory_df <- tidyverse_sensory_df %>%
                          separate(col=V1,into=c("Item", "1", "2", "3", "4", "5"), sep=" ", fill="left"

# remove first column since its missing values
tidyverse_sensory_df <- select(tidyverse_sensory_df, -Item)

# add item column
tidyverse_sensory_df <- tidyverse_sensory_df %>%
                                add_column("Item" = c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6,7,7,7,8,8,8,9

# gather all the year values into one column
tidy_tidyverse_sensory_df <- tidyverse_sensory_df %>% gather(key="Operator", value="Value", "1","2", "3

# show tidy data
knitr::kable(tidy_tidyverse_sensory_df[1:5,])
```

Item	Operator	Value
1	1	4.3

Item	Operator	Value
1	1	4.3
1	1	4.1
2	1	6.0
2	1	4.9

b. The data is year and Long jump performance of gold medalists. Looks like the year and longjump columns are broken up into multiple columns. We need to combine all data together into 2 columns "Year" and "Jump" with 60 rows

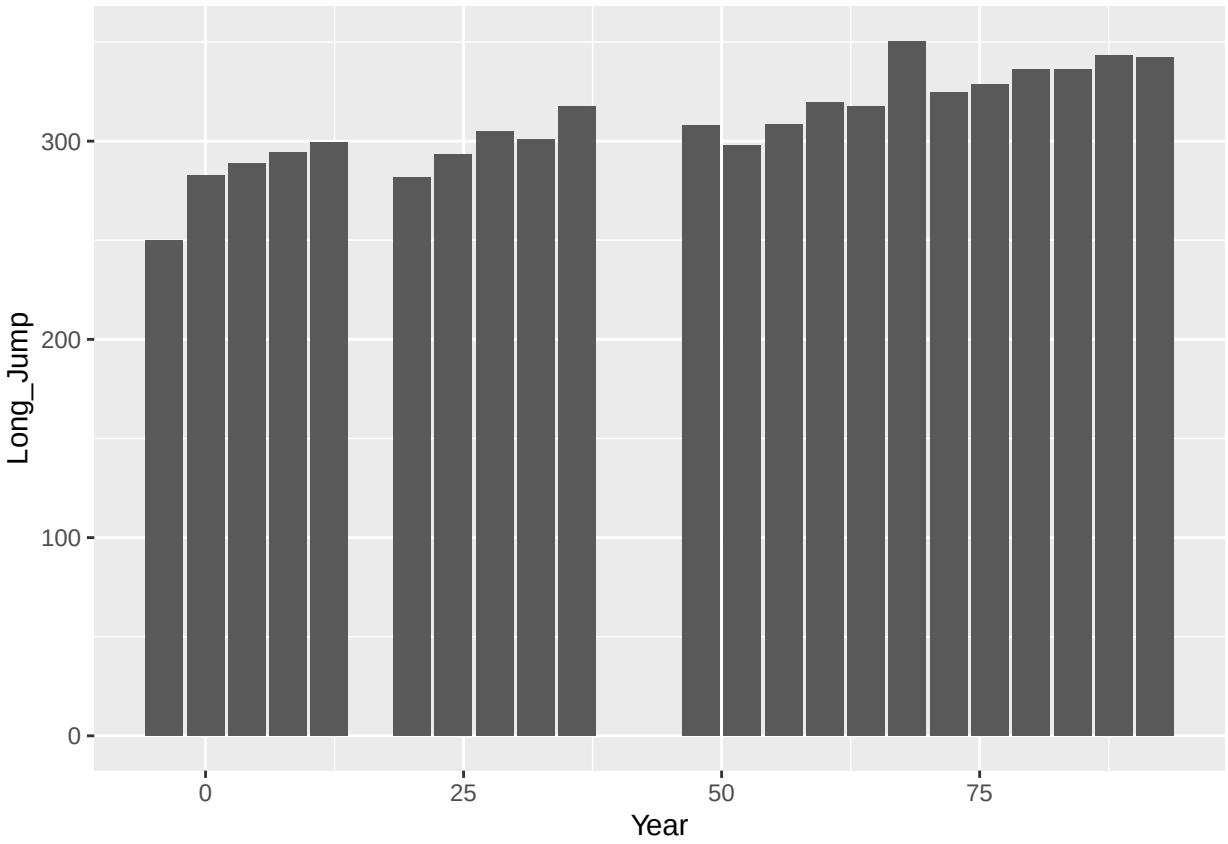
```
# read in url
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"

# save to rds so we are reslient against changes on the internet
#olympic_data <- fread(url, fill=TRUE, header=TRUE)
#saveRDS(olympic_data, "olympic_data_raw.RDS")
olympic_df <- readRDS("olympic_data_raw.RDS")
```

First we'll tidy the data using base r

```
# partition data into separate 2 column data frames
partition1_df = as.data.frame(olympic_df[,  1:2])
partition2_df = as.data.frame(olympic_df[,  3:4])
partition3_df = as.data.frame(olympic_df[,  5:6])
partition4_df = as.data.frame(olympic_df[,  7:8])

# rename columns
colnames(partition1_df) <- c("Year","Long_Jump")
colnames(partition2_df) <- c("Year","Long_Jump")
colnames(partition3_df) <- c("Year","Long_Jump")
colnames(partition4_df) <- c("Year","Long_Jump")

# remove NA values
partition4_df = partition4_df[complete.cases(partition4_df), ]

# concate all dataframes together
tidy_olympic_df = rbind(partition1_df,partition2_df,partition3_df,partition4_df)

# show tidy data
knitr::kable(tidy_olympic_df[1:5,])
```

Year	Long_Jump
-4	249.75
0	282.88
4	289.00
8	294.50
12	299.25

```
# informative plot
ggplot(data=tidy_olympic_df, aes(x=Year, y=Long_Jump)) +
  geom_bar(stat="identity")
```



Again, using tidyverse.

```
# so we make sure we are just working in this cell
tidverse_olympic_df <- olympic_df

# I was getting a names must be unique error, so I need to rename the columns first
colnames(tidverse_olympic_df) <- c("Year_1", "Jump_1", "Year_2", "Jump_2", "Year_3", "Jump_3", "Year_4"

# select just the populated columns
tidverse_olympic_df <- select(tidverse_olympic_df, 1:8)

# gather all the year values into one column
olympic_year_df <- tidverse_olympic_df %>% gather(key="na", value="Year", "Year_1","Year_2", "Year_3",

# gather all the jump values into one column
olympic_jump_df <- tidverse_olympic_df %>% gather(key="na", value="Jump", "Jump_1","Jump_2", "Jump_3",

# bind the columns together in a new dataframe
tidy_tidyverse_olympic_df <- bind_cols(olympic_year_df, olympic_jump_df)

# Remove NA rows
tidy_tidyverse_olympic_df <- slice(tidy_tidyverse_olympic_df, 1:22)

# show tidy data
knitr::kable(tidy_tidyverse_olympic_df[1:5,])
```

Year	Jump

Year	Jump
-4	249.75
0	282.88
4	289.00
8	294.50
12	299.25

c. This data has Brain weight (g) and body weight (kg) for 62 species. It seems to have a similar problem as the previous datasets with alternating columns needing concated together. Also has 6 Na columns at the end. Final data will have 2 columns Brain and Body weight with 62 rows.

```
# read in url
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"

# save to rds so we are reslient against changes on the internet
#brain_data <- fread(url, fill=TRUE, header=TRUE)
#saveRDS(brain_data, "brain_data_raw.RDS")
brain_df <- readRDS("brain_data_raw.RDS")
```

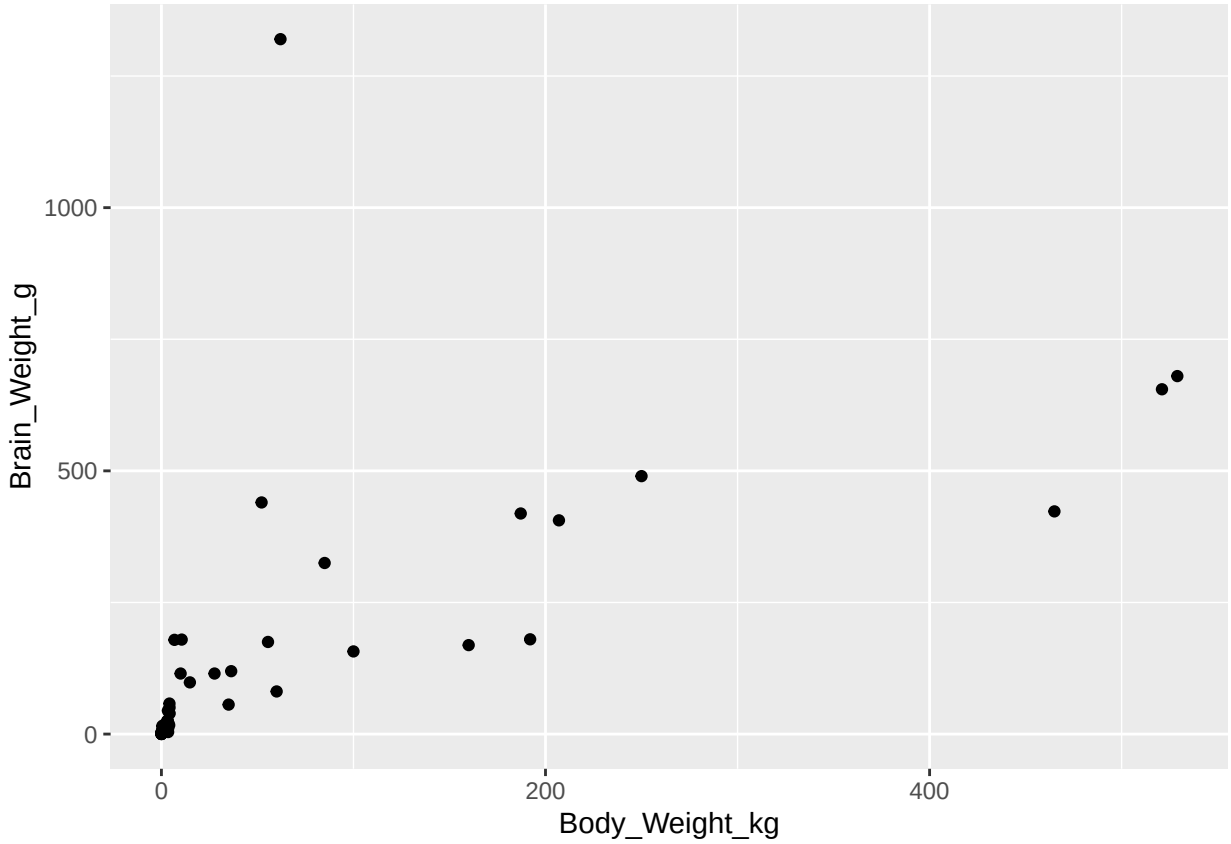
Cleaning using Base R

```
# partition data into separate 2 column data frames
partition1_df = as.data.frame(brain_df[,  1:2])
partition2_df = as.data.frame(brain_df[,  3:4])
partition3_df = as.data.frame(brain_df[,  5:6])

# rename columns
colnames(partition1_df) <- c("Body_Weight_kg","Brain_Weight_g")
colnames(partition2_df) <- c("Body_Weight_kg","Brain_Weight_g")
colnames(partition3_df) <- c("Body_Weight_kg","Brain_Weight_g")

# remove NA values
partition3_df = partition3_df[complete.cases(partition3_df), ]

# concate all dataframes together
tidy_brain_df = rbind(partition1_df,partition2_df,partition3_df)

# show tidy data
knitr::kable(tidy_brain_df[1:5,])
```

Body_Weight_kg	Brain_Weight_g
3.385	44.5
0.480	15.5
1.350	8.1
465.000	423.0
36.330	119.5

```
# informative plot

# take some of the noise out to get a cleaner plot
```

```
clean_brain_df <- filter(tidy_brain_df, Body_Weight_kg<2000)

# create a scatterplot
ggplot(clean_brain_df, aes(x=Body_Weight_kg, y=Brain_Weight_g)) + geom_point()
```



Cleaning using Tidy_verse

```
# so we make sure we are just working in this cell
tidverse_brain_df <- brain_df

# I was getting a names must be unique error, so I need to rename the columns first
colnames(tidverse_brain_df) <- c("Body_1", "Brain_1", "Body_2", "Brain_2", "Body_3", "Brain_3", "nacol"

# select just the populated columns
tidverse_brain_df <- select(tidverse_brain_df, 1:6)

# gather all the year values into one column
body_wt_df <- tidverse_brain_df %>% gather(key="na", value="Body Weight(kg)", "Body_1","Body_2", "Body_

# gather all the jump values into one column
brain_wt_df <- tidverse_brain_df %>% gather(key="na", value="Brain Weight(g)", "Brain_1","Brain_2", "Bra

# bind the columns together in a new dataframe
tidy_tidyverse_brain_df <- bind_cols(body_wt_df, brain_wt_df)

# show tidy data
knitr::kable(tidy_tidyverse_brain_df[1:5,])
```

Body Weight(kg)	Brain Weight(g)
3.385	44.5
0.480	15.5
1.350	8.1
465.000	423.0
36.330	119.5

d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities. The column title are in the first row, there are 2 columns of NA values. and cells have 3 values each inside of them. We need to transform it so that there are 3 columns, tomato variety, plant density and trip measurement, where each trip measurement has its own row, 18 rows total

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

# save to rds so we are reslient against changes on the internet
#tomato_data <- fread(url, fill=TRUE, header=TRUE)
#saveRDS(tomato_data, "tomato_data_raw.RDS")
tomato_df <- readRDS("tomato_data_raw.RDS")
```

Cleaning using Base R.

```
# remove NA columns
base_r_tomato_df <- as.data.frame(tomato_df[,1:4])

# remove first row, since its actually the column names
base_r_tomato_df <- as.data.frame(base_r_tomato_df[2:3,])

# rename columns
colnames(base_r_tomato_df) <- c("Tomato_Variety","pd_1000","pd_2000","pd_3000")

# split up Triplicate measurements in the cell into 3 separate rows

string_list_1 <- strsplit(base_r_tomato_df$pd_1000, split = ",")
pd_1000_df <- data.frame(Tomato_Variety = rep(base_r_tomato_df$Tomato_Variety, sapply(string_list_1, le

string_list_2 <- strsplit(base_r_tomato_df$pd_2000, split = ",")
pd_2000_df <- data.frame(pd_2000 = unlist(string_list_2))

string_list_3 <- strsplit(base_r_tomato_df$pd_3000, split = ",")
pd_3000_df <- data.frame(pd_3000 = unlist(string_list_3))

# bind columns together
base_r_tomato_df <- cbind(pd_1000_df, pd_2000_df, pd_3000_df)

partition_df_1 <- as.data.frame(cbind(Tomato_Variety=base_r_tomato_df[,1],
                                      Trip_Measurement=base_r_tomato_df[,2],
                                      Plant_Density = c(rep(1000,6))))
partition_df_2 <- as.data.frame(cbind(Tomato_Variety=base_r_tomato_df[,1],
                                      Trip_Measurement=base_r_tomato_df[,3],
                                      Plant_Density = c(rep(2000,6))))
partition_df_3 <- as.data.frame(cbind(Tomato_Variety=base_r_tomato_df[,1],
                                      Trip_Measurement=base_r_tomato_df[,4],
                                      Plant_Density = c(rep(3000,6))))
```

```
tidy_base_r_tomato_df <- rbind(partition_df_1,partition_df_2,partition_df_3)

# show tidy data
knitr::kable(tidy_base_r_tomato_df[1:5,])
```

Tomato_Variety	Trip_Measurement	Plant_Density
Ife#1	16.1	1000
Ife#1	15.3	1000
Ife#1	17.5	1000
PusaEarlyDwarf	8.1	1000
PusaEarlyDwarf	8.6	1000

Now with Tidyverse functions

```
# remove NA columns
tidyverse_tomato_df <- select(tomato_df, 1:4)

# remove first row, since its actually the column names
tidyverse_tomato_df <- slice(tidyverse_tomato_df[2:3])

# rename columns
colnames(tidyverse_tomato_df) <- c("Tomato_Variety","1000","2000","3000")

# separate values into rows, I was having trouble with this function, so need to use it 3 times
partition_1_df <- tidyverse_tomato_df %>%
                              separate_rows("1000")  %>%
                              slice(1:6) %>%
                              select(1:2)

partition_2_df <- tidyverse_tomato_df %>%
                              separate_rows("2000")  %>%
                              select(3)

partition_3_df <- tidyverse_tomato_df %>%
                              separate_rows("3000")  %>%
                              select(4)

# bind the columns, tidyverse style
tidyverse_tomato_df <- bind_cols(partition_1_df, partition_2_df, partition_3_df)

# gather all the year values into one column
tidy_tidyverse_tomato_df <- tidyverse_tomato_df %>% gather(key="Plant_Density", value="Trip_Measurement

# show tidy data
knitr::kable(tidy_tidyverse_tomato_df[1:5,])
```

Tomato_Variety	Plant_Density	Trip_Measurement
Ife#1	1000	16.1
Ife#1	1000	15.3
Ife#1	1000	17.5
PusaEarlyDwarf	1000	8.1
PusaEarlyDwarf	1000	8.6

Tomato_Variety	Plant_Density	Trip_Measurement

```
#informative plot
ggplot(tidy_tidyverse_tomato_df, aes(x=Plant_Density, y=Trip_Measurement, shape=Tomato_Variety, color=T
  geom_point()
```