HW2_tianjiang

Tian Jiang

9/11/2020

Problem 3

- Making my life easier in the long run.
- Making others appreciate my work.

Problem 4

Note. There are warnings of "NAs generated" in the cleaning process. But I removed the NAs in the following process so the results were not influenced. Thanks!

```
install.packages("dplyr")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
install.packages("tidyr")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(stringr)
library(ggplot2)
```

a. A Sensory data from five operators

Cleaning 2X

After this step, no outliers were found.

```
sensory <- read.delim("http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat")

#Base
sensory1 <- data.frame(lapply(sensory, as.character), stringsAsFactors=FALSE)
sensorylist <- strsplit(sensory1$X," ")
sensorylist2<-{}
for (i in 1:length(sensorylist))
{sensorylist1<-unlist(sensorylist[i])
 if (length(sensorylist1)>5)
    {sensorylist1<-sensorylist1[-1]}
     sensorylist2<-rbind(sensorylist2,sensorylist1)
}
sensory7<-as.data.frame(sensorylist2)
my_vector <- c(1)
for (i in 1:10)
{my_vector1 <- rep(i,3)
 my_vector<-c(my_vector,my_vector1)}
sensory7<-cbind(my_vector,sensory7)
sensory7 <- sensory7[-c(1), ]
colnames(sensory7) <- c("item","Operator1", "Operator2","Operator3","Operator4","Operator5")
sensory7<-as.data.frame(sensory7)
row.names(sensory7)<-c(1:30)
#sensory7

#Tidyverse
sensory<-sensory %>% separate(col = X, into = c("1","2","3","4","5","6"), sep = " ") %>% select(-Operat
```

```
## Warning: Expected 6 pieces. Missing pieces filled with `NA` in 20 rows [3, 4, 6,
## 7, 9, 10, 12, 13, 15, 16, 18, 19, 21, 22, 24, 25, 27, 28, 30, 31].
```

```
sensory2<-sensory[-1,]
for (i in 1:30)
{if ((i %% 3)!=1)
    {for (j in 2:6)
         sensory2[i,8-j]<- sensory2[i,8-j-1]
         sensory2[i,1]<-sensory2[i-1,1]
     }
 }
colnames(sensory2) <- c("item","Operator1", "Operator2","Operator3","Operator4","Operator5")
row.names(sensory2)<-c(1:30)
sensory8<-as.data.frame(sensory2)
t<-(sensory8)
sensory8=matrix(unlist(t(t)), byrow=T,30,6)
colnames(sensory8) <- c("item","Operator1", "Operator2","Operator3","Operator4","Operator5")
class(sensory8) <- "numeric"
df <- as_tibble(sensory8)
#df
```

Summary Table

Because no obvious outliers in the initial dataset were found, a table summarsing mean values of every item by each operator was built. Medians or ranges were not summarised because the numerical difference is not large in this dataset. But they would be calculated in the next question.

```
table<-df %>% group_by(item) %>% summarize(MeanOfOperator1 = mean(Operator1),MeanOfOperator2 = mean(Ope
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
table
```

```
## # A tibble: 10 x 6
##     item MeanOfOperator1 MeanOfOperator2 MeanOfOperator3 MeanOfOperator4
##    <dbl>           <dbl>           <dbl>           <dbl>           <dbl>
##  1     1            4.23            4.9             3.57            5.5
##  2     2            5.63            5.83            4.47            5.9
##  3     3            2.73            3.13            2.57            3.47
##  4     4            6.97            7.73            6.4             7.03
##  5     5            5.77            6               5.63            6.43
##  6     6            2.43            2.5             1.63            3.57
##  7     7            1.13            2.33            1.03            1.33
##  8     8            4               4.7             4.63            4.77
##  9     9            8.63            8.5             7.93            9.17
## 10    10            4.4             5               3.8             4.77
## # ... with 1 more variable: MeanOfOperator5 <dbl>
```

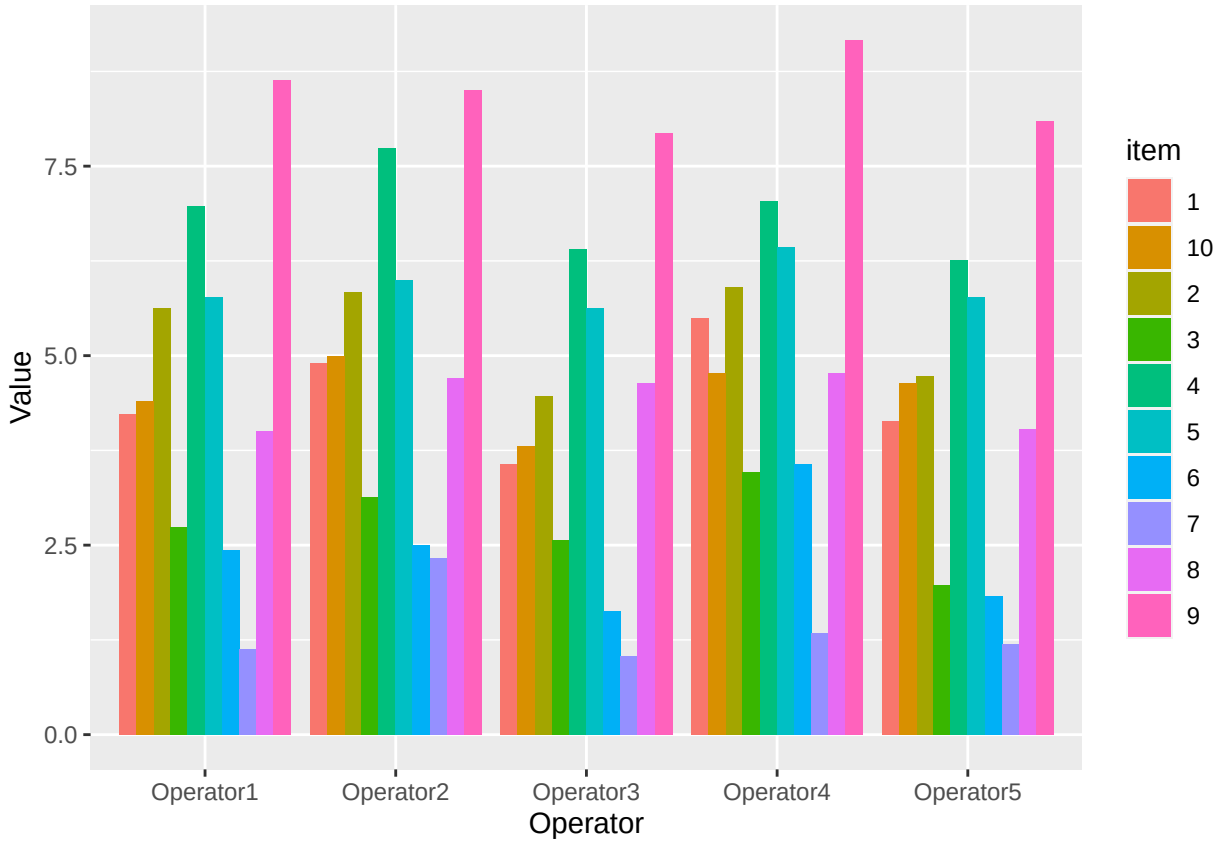
```
table1<-(cbind(table,rowMeans(table)))
table2<-cbind(table1[,1],table1[,7])
colnames(table2)<-c("item","Meanof5Operators")
table2 <- as_tibble(table2)
class(table2$item)<-"character"
table2
```

```
## # A tibble: 10 x 2
##    item  Meanof5Operators
##    <chr>            <dbl>
##  1 1                 3.89
##  2 2                 4.76
##  3 3                 2.81
##  4 4                 6.4
##  5 5                 5.77
##  6 6                 2.99
##  7 7                 2.34
##  8 8                 5.02
##  9 9                 8.56
## 10 10                5.43
```

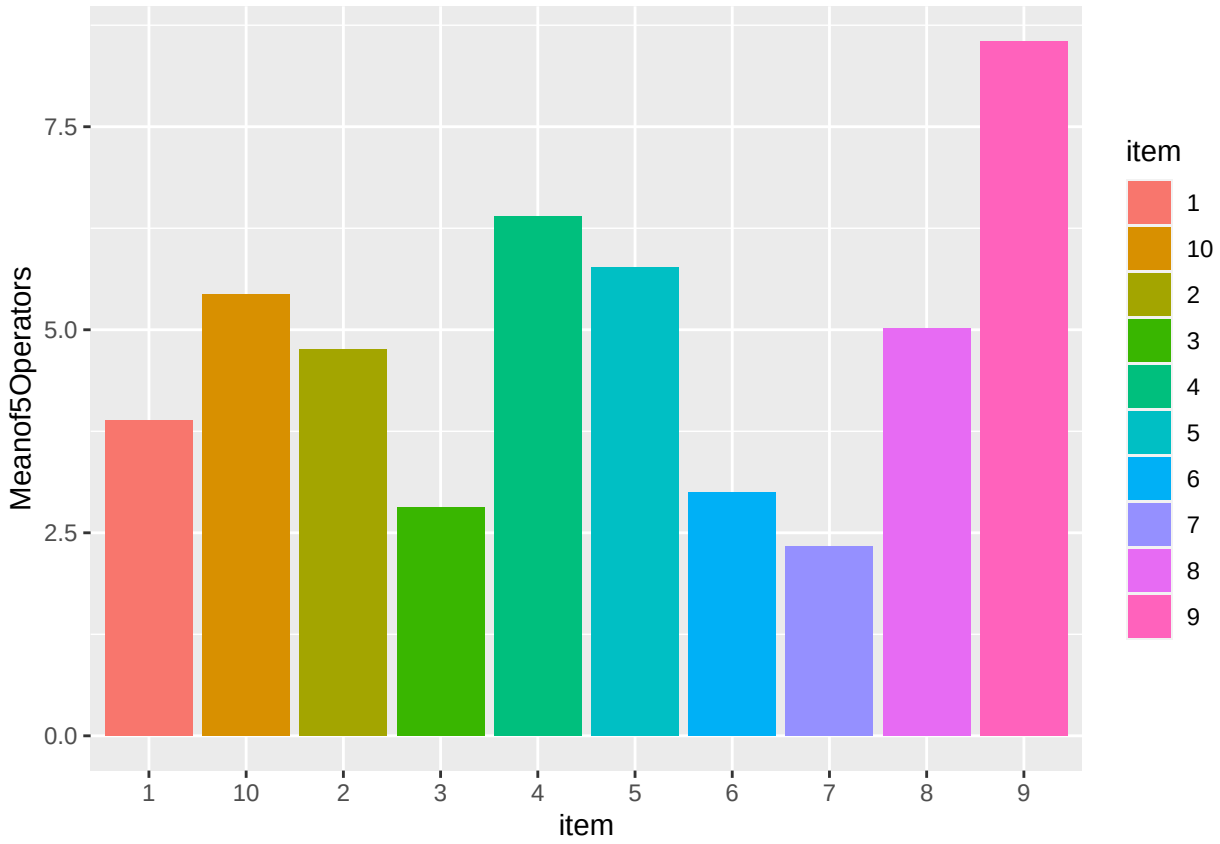
Plot

From the plot we see Item 9 got highest values with all of the 5 operators. And the values for a certain item among 5 operators don't vary much. For future endeavors, I'll also compare behaviors of data derived each time because there 3 sets of values for each item.

```
DF <- data.frame(item = c("1", "2", "3","4","5","6","7","8","9","10"),
                 Operator1 = c(table$MeanOfOperator1),
                 Operator2 = c(table$MeanOfOperator2),
                 Operator3 = c(table$MeanOfOperator3),
                 Operator4 = c(table$MeanOfOperator4),
                 Operator5 = c(table$MeanOfOperator5))
DFtall <- DF %>% gather(key = Operator, value = Value, Operator1:Operator5)
plot<-ggplot(DFtall, aes(Operator, Value, fill =item)) + geom_col(position = "dodge")
plot
```




```
plot1<-ggplot(table2, aes(item, Meanof5Operators, fill =item)) + geom_col(position = "dodge")
plot1
```



b. Gold Medal performance for Olympic Men's Long Jump

Cleaning 2X No outlier detected

```
LongJumpData <- read.delim("http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat")

#Base
LongJumpData1 <- data.frame(lapply(LongJumpData, as.character), stringsAsFactors=FALSE)
LongJumpDatalist <- strsplit(LongJumpData1$Year.Long.Jump.Year.Long.Jump.Year.Long.Jump.Year.Long.Jump,
LongJumpDataframe<-as.data.frame(LongJumpDatalist[1:4])
t<-t(LongJumpDataframe)
LongJumpDatamatrix1=matrix(unlist(t(t)), byrow=T, 16, 2)
LongJumpDataframe<-as.data.frame(LongJumpDatalist[5:6])
t<-t(LongJumpDataframe)
LongJumpDatamatrix2=matrix(unlist(t(t)), byrow=T, 6, 2)
df6<-rbind(LongJumpDatamatrix1,LongJumpDatamatrix2)
colnames(df6) <- c("Year","Score")
class(df6) <- "numeric"

#Tidyverse
colnames(LongJumpData) <- c("X")
LongJumpData1<-LongJumpData %>% separate(col=X,into=c("1","2","3","4","5","6","7","8"),sep = " ")
```

```
## Warning: Expected 8 pieces. Missing pieces filled with `NA` in 2 rows [5, 6].
```

```
df1 <-LongJumpData1 %>% select(1, 2)
df2 <-LongJumpData1 %>% select(3, 4)
colnames(df2) <- c("1","2")
df3 <-LongJumpData1 %>% select(5, 6)
colnames(df3) <- c("1","2")
df4 <-LongJumpData1 %>% select(7, 8)
colnames(df4) <- c("1","2")
df5<-rbind(df1,df2,df3,df4)
colnames(df5) <- c("Year","Score")
df5<-drop_na(df5)
sensory8<-as.data.frame(df5)
t<-(sensory8)
sensory8=matrix(unlist(t(t)), byrow=T,22,2)
colnames(sensory8) <- c("Year","Score")
class(sensory8) <- "numeric"
df <- as_tibble(sensory8)
df[,1]<-df[,1]+1900
```

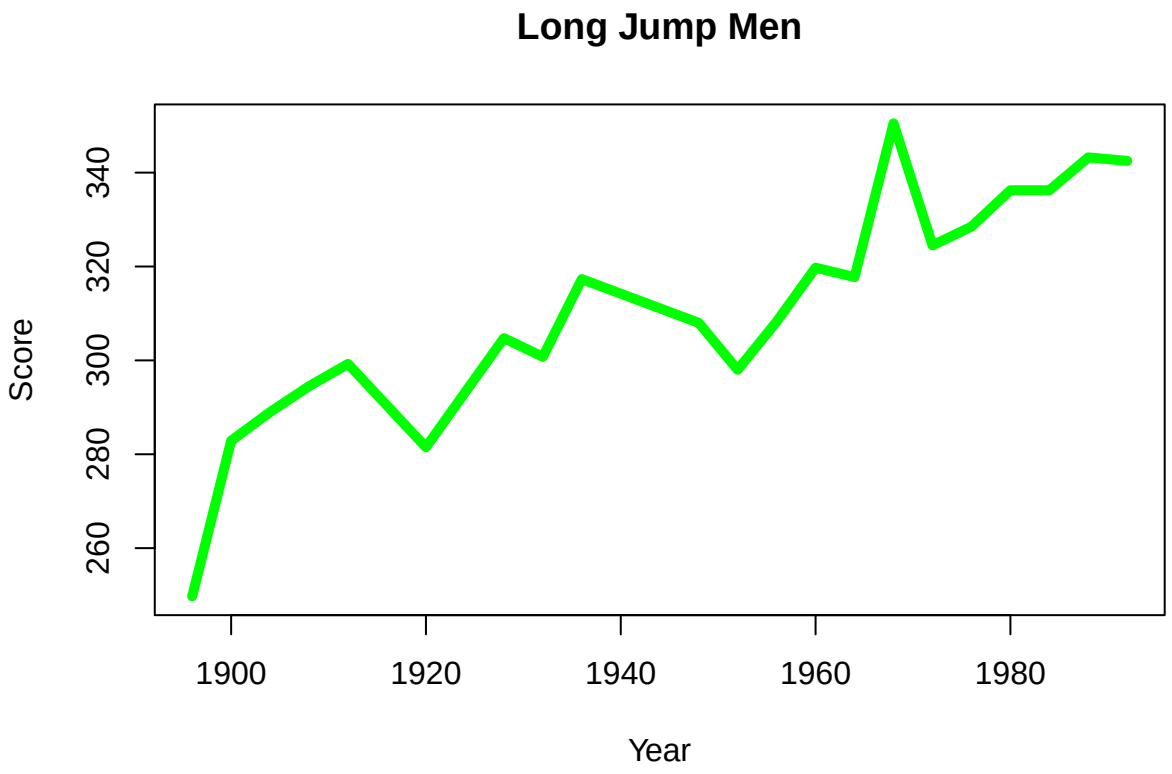
Summary Table

```
table<-df %>%  summarize(MaxScore = max(Score),MinScore = min(Score),MeanScore = mean(Score),MedianScor
table
```

```
## # A tibble: 1 x 4
##   MaxScore MinScore MeanScore MedianScore
##      <dbl>    <dbl>     <dbl>       <dbl>
## 1     350.     250.      310.        308.
```

Plot

```
plot(df$Year,df$Score, type="l", col="green", lwd=5, xlab="Year", ylab="Score", main="Long Jump Men")
```



c. Brain weight (g) and body weight (kg) for 62 species

Issues

Different species can have very different scales of weights. But the meaningful thing to do is to look at the Brain-Body weight ratio of every species. ### Cleaning 2X

```
weightData <- read.delim("http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat")

#Base
weightData1 <- data.frame(lapply(weightData, as.character), stringsAsFactors=FALSE)
weightDatalist <- strsplit(weightData1$Body.Wt.Brain.Wt.Body.Wt.Brain.Wt.Body.Wt.Brain.Wt," ")
weightDataframe<-as.data.frame(weightDatalist[1:20])
t<-t(weightDataframe)
weightDatamatrix1=matrix(unlist(t(t)), byrow=T, 60, 2)
weightDataframe<-as.data.frame(weightDatalist[21])
t<-t(weightDataframe)
weightDatamatrix2=matrix(unlist(t(t)), byrow=T, 2, 2)
weight<-rbind(weightDatamatrix1,weightDatamatrix2)
colnames(weight) <- c("Body.Wt","Brain.Wt")
class(weight) <- "numeric"

#Tidyverse
colnames(weightData) <- c("X")
weightData1<-weightData %>% separate(col=X,into=c("1","2","3","4","5","6"),sep = " ")
```

```
## Warning: Expected 6 pieces. Missing pieces filled with ‘NA‘ in 1 rows [21].
```

```
df1 <-weightData1 %>% select(1, 2)
df2 <-weightData1 %>% select(3, 4)
colnames(df2) <- c("1","2")
df3 <-weightData1 %>% select(5, 6)
colnames(df3) <- c("1","2")
df5<-rbind(df1,df2,df3)
colnames(df5) <- c("Body.Wt","Brain.Wt")
df5<-drop_na(df5)
sensory8<-as.data.frame(df5)
t<-(sensory8)
sensory8=matrix(unlist(t(t)), byrow=T,62,2)
colnames(sensory8) <- c("Body.Wt","Brain.Wt")
class(sensory8) <- "numeric"
df <- as_tibble(sensory8)
df$Ratio<-(df$Brain.Wt)*0.001/(df$Body.Wt)
```

Summary Table

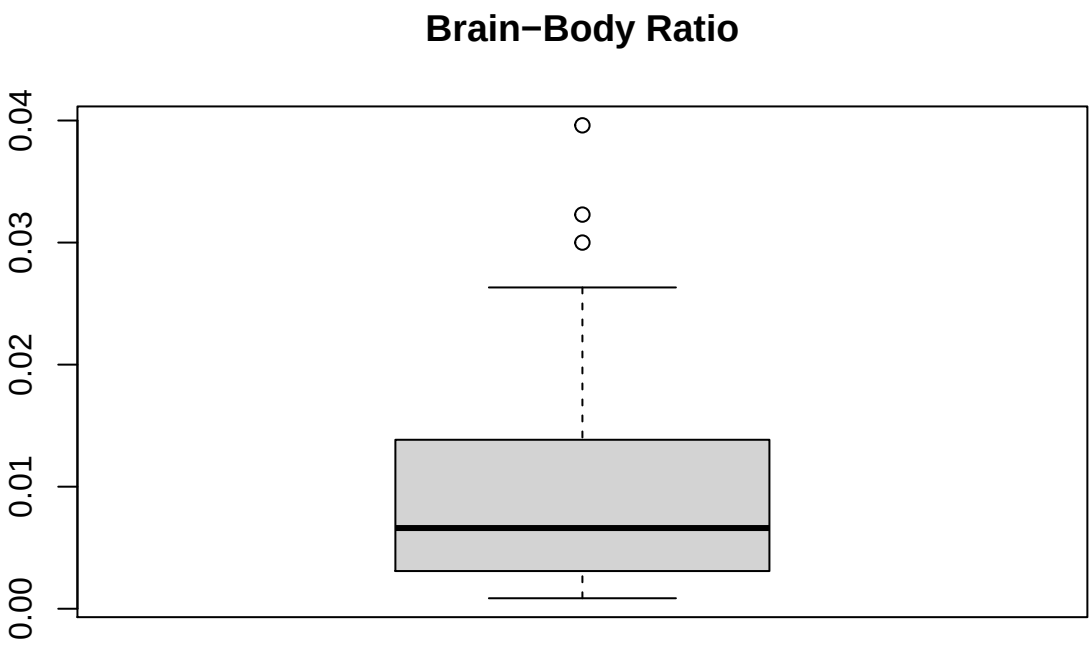
```
table<-df %>%  summarize(Max = max(Body.Wt),Min = min(Body.Wt),Mean = mean(Body.Wt),Median = median(Bod
table1<-df %>%  summarize(Max = max(Brain.Wt),Min = min(Brain.Wt),Mean = mean(Brain.Wt),Median = median
table2<-df %>%  summarize(Max = max(Ratio),Min = min(Ratio),Mean = mean(Ratio),Median = median(Ratio))
Variable<-c("Body.Wt/kg","Brian.Wt/g","Ratio")
a<-rbind(table,table1,table2)
FinalTable<-cbind(Variable,a)
FinalTable
```

```
##     Variable          Max         Min         Mean       Median
## 1 Body.Wt/kg 6.654000e+03 0.005000000 1.987900e+02  3.342500000
## 2 Brian.Wt/g 5.712000e+03 0.100000000 2.831344e+02 17.250000000
## 3      Ratio 3.960396e-02 0.000858431 9.575827e-03  0.006610856
```

A valuable plot

The ratios for 50% of the species are between about 0.005 and 0.015. Only 2 of them are larger than 0.03. We could further explore if the 2 species have higher IQ.

```
boxplot(df$Ratio,main = "Brain-Body Ratio")
```



d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities.

Cleaning 2X No outlier detected.

```
tomato <- read.delim("http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat")

#Base
tomato1 <- data.frame(lapply(tomato, as.character), stringsAsFactors=FALSE)
tomatolist <- strsplit(tomato1$X.this.needs.reformatting.to.read.into.Splus,"\\,\\s|\\,|\\s")
I<-cbind(c("tomato"))
R <- cbind(c(10000))[rep(1,3), ]
X <- cbind(c(20000))[rep(1,3), ]
Z <- cbind(c(30000))[rep(1,3), ]
matrix<-cbind(I,t(R),t(X),t(Z))
for (i in 2:3)
{tomatoDataframe<-as.data.frame(tomatolist[i])
t<-t(tomatoDataframe)
tomatoDatamatrix1=matrix(unlist(t(t)))
tomatoDatamatrix1<-tomatoDatamatrix1[tomatoDatamatrix1 != ""]
matrix=rbind(matrix,tomatoDatamatrix1)}
colnames(matrix) <- matrix[1,]
matrix<-matrix[2:3,]
row.names(matrix)<-c(1,2)

#Tidyverse
df <- data.frame(matrix(ncol = 4, nrow = 2))
Dataframe<-{}
for (i in 1:2)
{x<- trimws(tomato[i+1,])
 y<- gsub("\\s+"," ", x)
```

```
 z<-strsplit(y, " ")
 X_Dataframe = as.data.frame(t(unlist(z)))
 Dataframe<-rbind(Dataframe,X_Dataframe)
}
Dataframe1<-Dataframe %>% separate(col=V2,into=c("1","2","3"),sep = ",")
```

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

```
Dataframe2<-Dataframe1 %>% separate(col=V3,into=c("4","5","6"),sep = ",")
Dataframe3<-Dataframe2 %>% separate(col=V4,into=c("7","8","9"),sep = ",")
colnames(Dataframe3) <- cbind(I,t(R),t(X),t(Z))
```

Summary Table

```
sensory8<-as.data.frame(Dataframe3)
data<-cbind(colnames(sensory8),t(sensory8))
class(data) <- "numeric"
```

```
## Warning in class(data) <- "numeric": NAs introduced by coercion
```

```
g<-as_tibble(data)
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if `.name_repair` is
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
g<-g[2:10,]
colnames(g)<-c("Density","Ife\\#1","PusaEarlyDwarf")
table<-g %>% group_by(Density) %>% summarize(IfeMeanYield = mean(`Ife\\#1`),PusaMeanYield = mean(PusaEa
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
table
```

```
## # A tibble: 3 x 3
##   Density IfeMeanYield PusaMeanYield
##     <dbl>        <dbl>         <dbl>
## 1   10000         16.3          8.93
## 2   20000         18.1         12.6
## 3   30000         19.9         14.5
```

Plot

Ife always performs better than Pusa under the 3 densities. And the larger the density is, the greater yield we get. But the gap between 20000 and 10000 is larger than that between 30000 and 20000. We guess increasing the density plays a less important role in improving the yield as the density becomes larger.

```
DF <- data.frame(Density = c("10000", "20000","30000"),
                 IfeTomato = c(table$IfeMeanYield),
                 PusaTomato = c(table$PusaMeanYield))
DFtall <- DF %>% gather(key = Tomato, value = Yield, IfeTomato:PusaTomato)
DFtall
```

```
##   Density     Tomato     Yield
## 1   10000  IfeTomato 16.300000
## 2   20000  IfeTomato 18.100000
## 3   30000  IfeTomato 19.933333
## 4   10000 PusaTomato  8.933333
## 5   20000 PusaTomato 12.633333
## 6   30000 PusaTomato 14.500000
```

```
plot<-ggplot(DFtall, aes(Tomato, Yield, fill =Density)) + geom_col(position = "dodge")
plot
```