HW2_wliu3

Wei Liu

9/13/2020

Problem 1

Finished in the RStudio Cloud

Problem 2

Finished in the RStudio Cloud

Problem 3

I think the version control is an very efficient tool to keep track all changes and scripts we made, so we can check back anytime and anywhere. This is a good way to get the reproducible results.

Problem 4

a. Sensory data from five operators

we are looking at the sensory data from five operators. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat

First, we will get the data from the link above:

Need to tidy the data, some rows are misplaced. There should be variables for each column: items, observation numbers, operators and data. However, five different operators are now in different columns. In addition, observation numbers are missing.

Method 1: base R

```
# check the column and row numbers of sensory_data_raw
dim(sensory_data_raw)
```

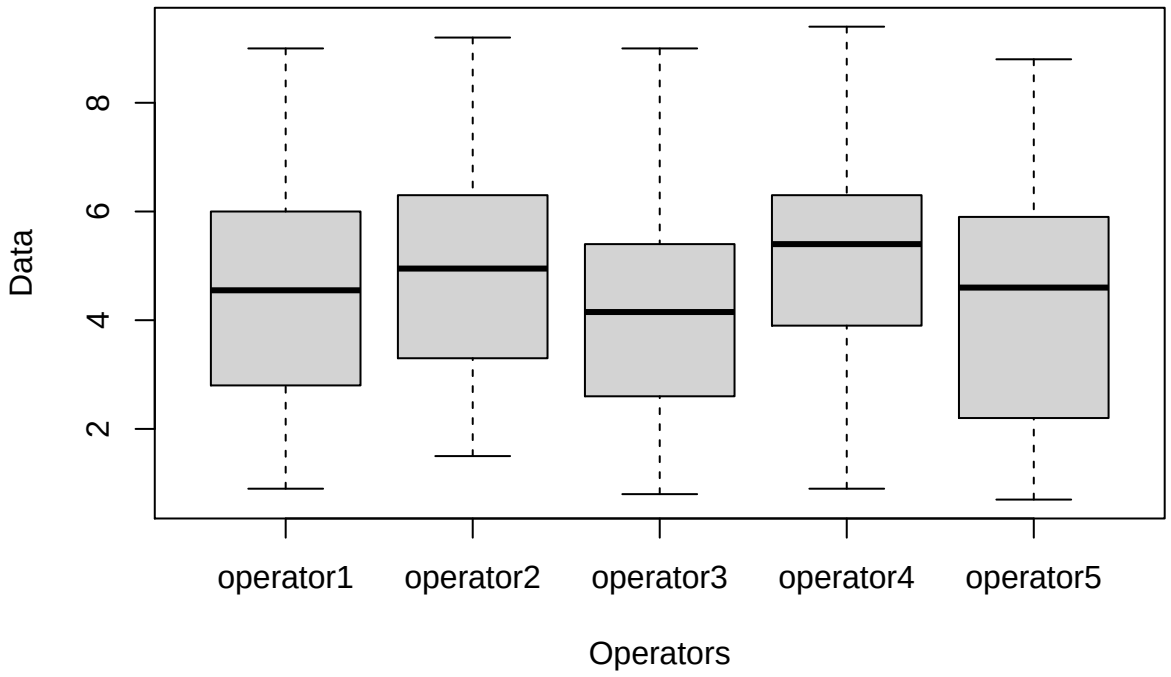
```
## [1] 30  6
```

```
# dim=(30,6)

# Use loop to move the NA in the column V6 to the first column V1
for (i in 1:30) {
```

```
  if(is.na(sensory_data_raw$V6[i])) {
  sensory_data_raw[i,-1] <- sensory_data_raw[i,1:5]
  sensory_data_raw[i,1] <- NA
  }
}

# There are 10 items in total, so need fix the items numbers in the first column.
sensory_data_raw$V1 <- rep(1:10, each = 3)

# create a new column V7 for observation numbers.
sensory_data_raw$V7 <- rep(c("ob.1","ob.2","ob.3"),10)
sensory_data_raw1<-sensory_data_raw

# set the column names
colnames(sensory_data_raw1)<-
  c("items", "operator1", "operator2", "operator3", "operator4", "operator5","observations")

# choose the columns of operator1,2,3,4,5 and then combine them into one column.
data<-c(sensory_data_raw1[,2],sensory_data_raw1[,3],sensory_data_raw1[,4],
        sensory_data_raw1[,5],sensory_data_raw1[,6])

# create the new columns for variables of items, observations, and operators.
items<-rep(rep(1:10, each = 3),5)
observations<-rep(rep(c("ob.1","ob.2","ob.3"),10),5)
operator<-rep(c("operator1","operator2","operator3","operator4","operator5"), each=30)

# bind the above columns together with the sensory data
sensory_data_tidy_br<-as.data.frame(cbind(items,operator,observations,data))

# convert the columns of items and data into numeric
sensory_data_tidy_br$items <- as.numeric(as.character(sensory_data_tidy_br$items))
sensory_data_tidy_br$data <- as.numeric(as.character(sensory_data_tidy_br$data))
```

We have converted the dataframes to tidy data frames using the base functions. Here is a summary of the data:

items	operator	observations	data
Min. : 1.0	Length:150	Length:150	Min. :0.700
1st Qu.: 3.0	Class :character	Class :character	1st Qu.:3.025
Median : 5.5	Mode :character	Mode :character	Median :4.700
Mean : 5.5	NA	NA	Mean :4.657
3rd Qu.: 8.0	NA	NA	3rd Qu.:6.000
Max. :10.0	NA	NA	Max. :9.400



Method 2: tidyverse R

```
#stack column using tidyverse
sensory_data_tidy_tv <- sensory_data_raw1 %>%
  gather(key="operator", value="data", operator1:operator5)

# the head of sensory_data_tidy_tv
knitr::kable(head(sensory_data_tidy_tv))
```

items	observations	operator	data
1	ob.1	operator1	4.3
1	ob.2	operator1	4.3
1	ob.3	operator1	4.1
2	ob.1	operator1	6.0
2	ob.2	operator1	4.9
2	ob.3	operator1	6.0

We have converted the dataframes to tidy data frames using the tidyverse functions. Here is a summary of the data:

items	observations	operator	data
Min. : 1.0	Length:150	Length:150	Min. :0.700
1st Qu.: 3.0	Class :character	Class :character	1st Qu.:3.025
Median : 5.5	Mode :character	Mode :character	Median :4.700
Mean : 5.5	NA	NA	Mean :4.657
3rd Qu.: 8.0	NA	NA	3rd Qu.:6.000
Max. :10.0	NA	NA	Max. :9.400

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

we are looking at Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat

First, we will get the data from the link above:

Need to tidy the data, the data for "year" and "longjump" are in several different columns and the "year" should be converted to normal values.

Method 1: base R

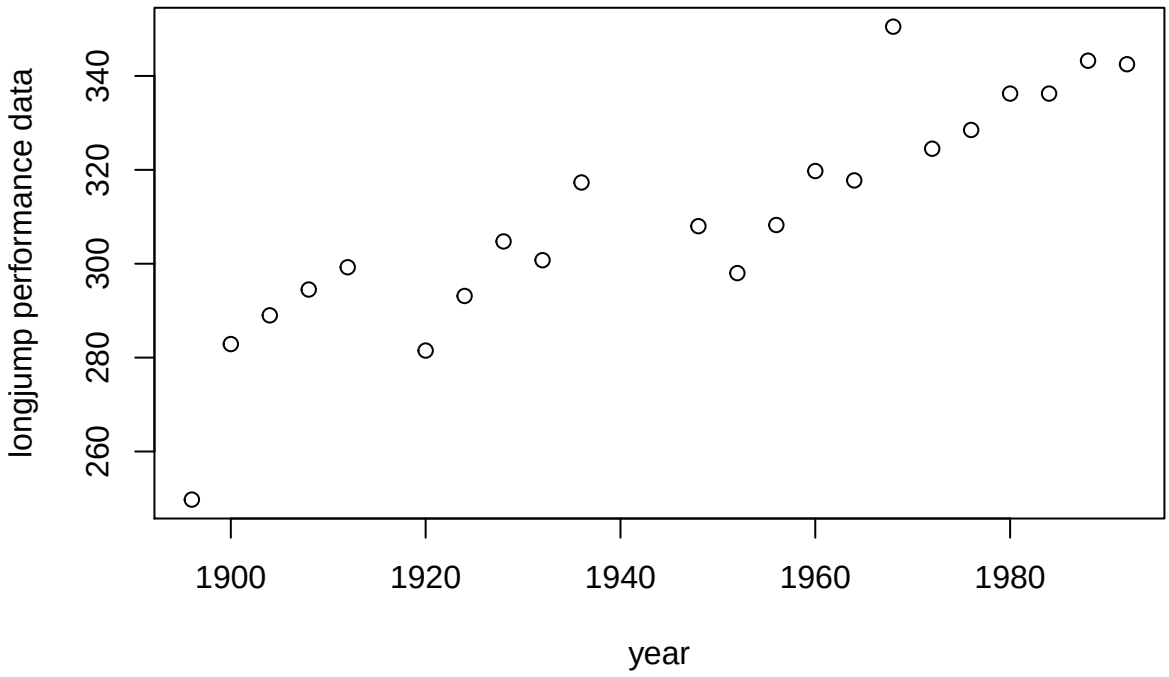
```
# check the column and row numbers of sensory_data_raw
dim(performance_import_raw)
```

```
## [1] 6 8
```

```
# dim=(6,8)

# select the columns for the "year"
year<-c(performance_import_raw[,1], performance_import_raw[,3], performance_import_raw[,5], performance

# select the columns for the "longjump"
longjump<-c(performance_import_raw[,2], performance_import_raw[,4], performance_import_raw[,6], performa

# bind the columns for the "year" and "longjump" together
performance_import_raw1<-cbind(year,longjump)

# creat a new column for converted year values
year1<-performance_import_raw1[,1]+1900
performance_data_tidy_br<-cbind(year1, performance_import_raw1)
performance_data_tidy_br1<-performance_data_tidy_br[,-2]

performance_data_tidy_br2<-na.omit(performance_data_tidy_br1)
```

We have converted the dataframes to tidy data frames using the base functions. Here is a summary of the data:

year1	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



Method 2: tidyverse R

```
#select the columns for the "year"
year_select<-performance_import_raw %>%
                    select(c(V1,V3,V5,V7))

#stack the columns for the "year"
year_output<-stack(year_select) %>% select(values) %>%
                    `colnames<-`("year")

#select the columns for the "longjump"
longjump_select<-performance_import_raw %>%
                    select(c(V2,V4,V6,V8))

#stack the columns for the "longjump"
longjump_output<-stack(longjump_select) %>%
            select(values) %>%
            `colnames<-`("longjump")

#bind the columns of "year" and "longjump" together
performance_data_tidy_tv<-cbind(year_output,longjump_output)%>%
                            mutate(year22=year+1900)%>%
                            select(year22,longjump)%>%
                            na.omit()

# the head of performance_data_tidy_tv
knitr::kable(head(performance_data_tidy_tv))
```

year22	longjump
1896	249.75

year22	longjump
1900	282.88
1904	289.00
1908	294.50
1912	299.25
1920	281.50

We have converted the dataframes to tidy data frames using the tidyverse functions. Here is a summary of the data:

year22	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

c. Brain weight (g) and body weight (kg) data

we are looking at Brain weight (g) and body weight (kg) for 62 species. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat

First, we will get the data from the link above:

Need to tidy the data, the data for "brain wt" and "body wt" are in several different columns.

Method 1: base R

```
# check the column and row numbers of weight_import_raw
dim(weight_import_raw)
```

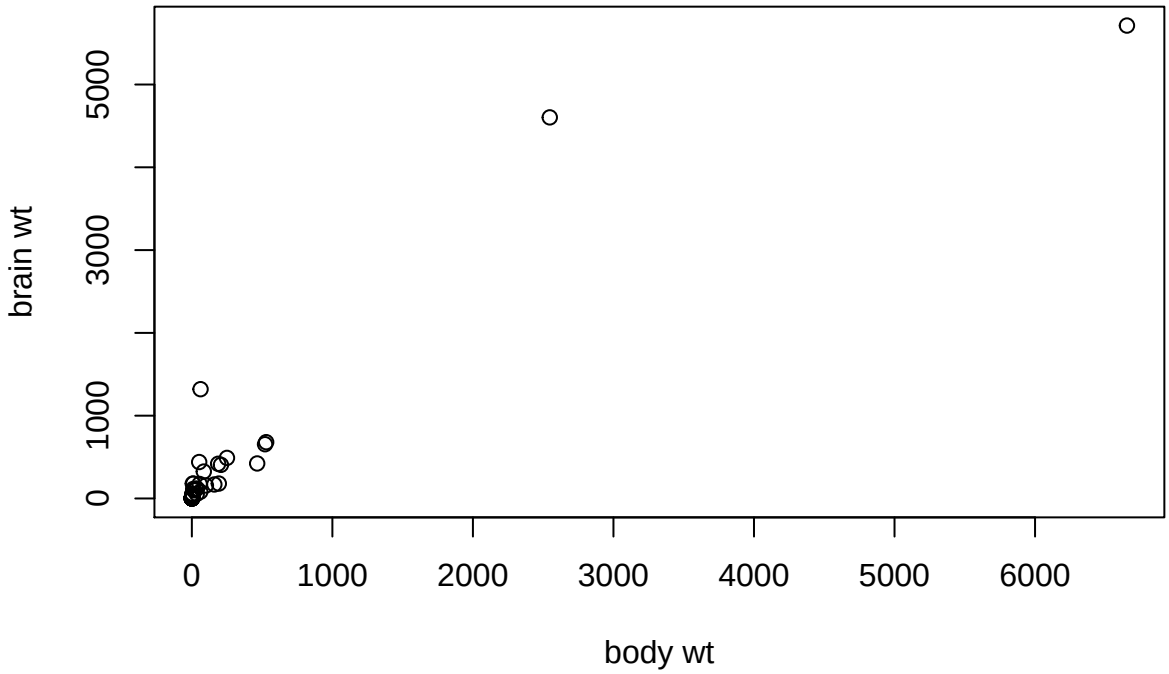
```
## [1] 21  6
```

```
# dim=(21,6)

# select the columns for the "body wt"
body_wt<-c(weight_import_raw[,1], weight_import_raw[,3], weight_import_raw[,5])

# select the columns for the "weight wt"
brain_wt<-c(weight_import_raw[,2],weight_import_raw[,4],weight_import_raw[,6])

# bind the columns for the "body wt" and "weight wt" together
weigiht_data_tidy_br<-cbind(body_wt,brain_wt)

weigiht_data_tidy_br<-na.omit(weigiht_data_tidy_br)
```

We have converted the dataframes to tidy data frames using the base functions. Here is a summary of the data:

body_wt	brain_wt

body_wt	brain_wt
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



Method 2: tidyverse R

```
#select the columns for the "body wt"
body_wt1<- weight_import_raw %>%
                    select(c(V1,V3,V5))

#stack the columns for the "body wt"
body_wt2<-stack(body_wt1) %>% select(values) %>%
                    `colnames<-`("body_wt")

#select the columns for the "brain wt"
brain_wt1<- weight_import_raw %>%
                    select(c(V2,V4,V6))

#stack the columns for the "brain wt"
brain_wt2<-stack(brain_wt1) %>%
            select(values) %>%
            `colnames<-`("brain_wt")
```

```r
weight_data_tidy_tv<-cbind(body_wt2,brain_wt2) %>% na.omit()

# the head of weight_data_tidy_tv
knitr::kable(head(weight_data_tidy_tv))
```

body_wt	brain_wt
3.385	44.5
0.480	15.5
1.350	8.1
465.000	423.0
36.330	119.5
27.660	115.0

We have converted the dataframes to tidy data frames using the tidyverse functions. Here is a summary of the data:

body_wt	brain_wt
Min. : 0.005	Min. : 0.005
1st Qu.: 0.550	1st Qu.: 1.375
Median : 3.300	Median : 6.800
Mean : 198.374	Mean : 240.889
3rd Qu.: 44.245	3rd Qu.: 91.600
Max. :6654.000	Max. :6654.000

d. Triplicate measurements of tomato yield

Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities. http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat

First, we will get the data from the link above:

Need to tidy the data, several different values for "tomato yield" are in one column.

Method 1: base R

For the base R method, I am still working on find a way without using "select" and "separate"

```r
# select the density columns for "10000","20000" and "30000"
tomato_import_raw1<-select(tomato_import_raw, "10000","20000","30000")

# separate the yield values for "10000","20000" and "30000"
tomato_import_raw2<-separate(tomato_import_raw1, "10000", into = c("10000m1", "10000m2","10000m3"),sep=
```

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

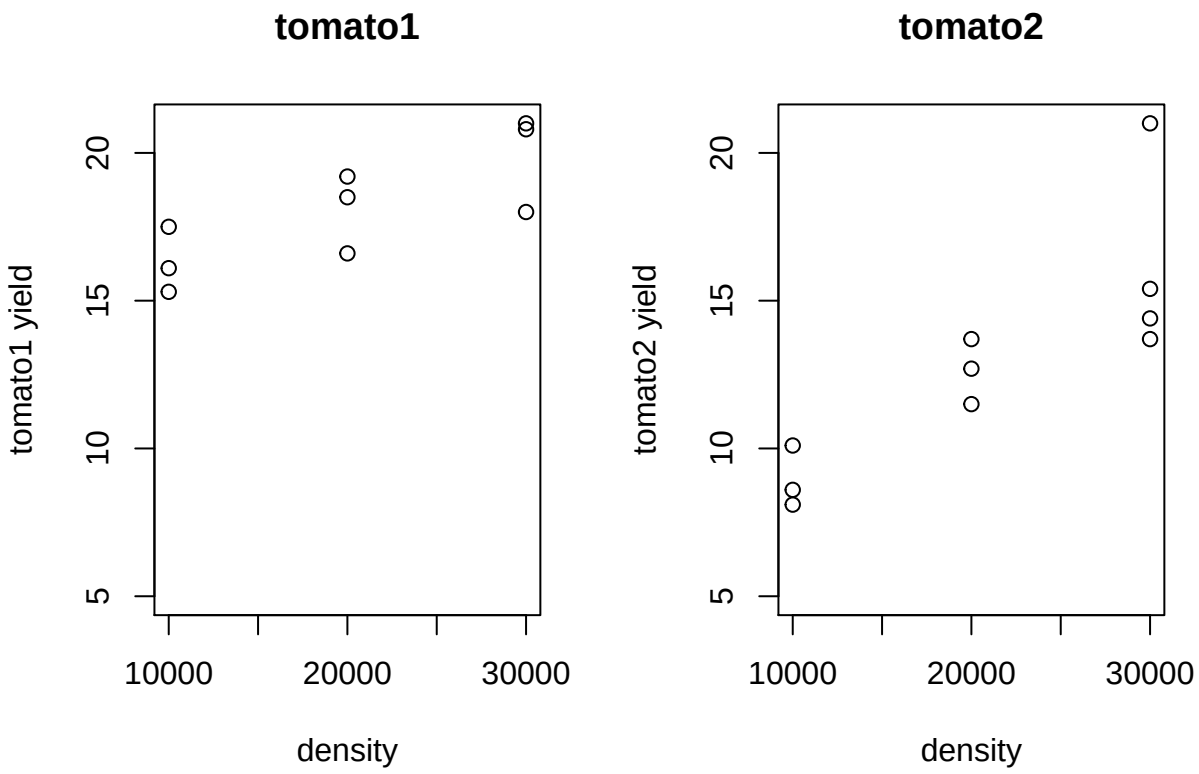
```r
tomato_import_raw3<-separate(tomato_import_raw2, "20000", into = c("20000m1", "20000m2","20000m3"),sep=
tomato_import_raw4<-separate(tomato_import_raw3, "30000", into = c("30000m1", "30000m2","30000m3"),sep=

# transpose the data and convert to numeric
tomato_import_raw4<-t(tomato_import_raw4)
tomato_import_raw5<-c(tomato_import_raw4[,1],tomato_import_raw4[,2])
```

```r
tomato_import_raw6<-as.numeric(tomato_import_raw5)

# create the new columns for tamato_type, density, and observations
tomato_type<-rep(c("tomato1","tamato2"),each=9)
density<-rep(rep(c(10000,20000,30000),each=3),2)
observations<-rep(rep(c("ob.1","ob.2","ob.3"),3),2)

# bind the above columns into one dataset
tomato_data_tidy_br<-as.data.frame(cbind(tomato_type,density,observations,tomato_import_raw6))
colnames(tomato_data_tidy_br)<-c("tomato_type","density","observations","yield")

tomato_data_tidy_br$yield<-as.numeric(tomato_data_tidy_br$yield)

#the head of tomato_data_tidy_br
knitr::kable(head(tomato_data_tidy_br))
```

tomato_type	density	observations	yield
tomato1	10000	ob.1	16.1
tomato1	10000	ob.2	15.3
tomato1	10000	ob.3	17.5
tomato1	20000	ob.1	16.6
tomato1	20000	ob.2	19.2
tomato1	20000	ob.3	18.5

We have converted the dataframes to tidy data frames using the base functions. Here is a summary of the data:

tomato_type	density	observations	yield
Length:18	Length:18	Length:18	Min. : 8.10
Class :character	Class :character	Class :character	1st Qu.:12.95
Mode :character	Mode :character	Mode :character	Median :15.35
NA	NA	NA	Mean :15.07
NA	NA	NA	3rd Qu.:17.88
NA	NA	NA	Max. :21.00



Method 2: tidyverse R

```
# select the density columns for "10000","20000" and "30000" and separate the values
tomato_import_raw7 <- tomato_import_raw %>%
    select("10000","20000","30000") %>%
    separate("10000", into = c("10000m1", "10000m2","10000m3"),sep=",")%>%
    separate("20000", into = c("20000m1", "20000m2","20000m3"),sep=",")%>%
    separate("30000", into = c("30000m1", "30000m2","30000m3"),sep=",")
```

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

```
# transpose the data and convert to numeric
tomato_import_raw7<-t(tomato_import_raw7)
tomato_import_raw8<-c(tomato_import_raw7[,1],tomato_import_raw7[,2])
tomato_import_raw9<-as.numeric(as.character(tomato_import_raw8))


# bind the above columns into one dataset
tomato_data_tidy_tv<-as.data.frame(cbind(tomato_type,density,observations,tomato_import_raw9))

colnames(tomato_data_tidy_tv)<-c("tomato_type","density","observations","yield")
tomato_data_tidy_tv$yield<-as.numeric(tomato_data_tidy_tv$yield)
```

We have converted the dataframes to tidy data frames using the tidyverse function. Here is a summary of the data:

tomato_type	density	observations	yield
Length:18	Length:18	Length:18	Min. : 8.10
Class :character	Class :character	Class :character	1st Qu.:12.95

tomato_type	density	observations	yield
Mode :character	Mode :character	Mode :character	Median :15.35
NA	NA	NA	Mean :15.07
NA	NA	NA	3rd Qu.:17.88
NA	NA	NA	Max. :21.00