HW2_Lian

Jiayi Lian

September 4, 2019

Homework2

P3

It can help me know how far have you gone through in your project.I can also see what changes have been done after each version created. There is more convenience. If I am not satisfied with my final version, I am able to find proper immediate version to start. It would save my time to go back step by step.

p4

Before munge, I took a glimpse of dats. I saw that colnames, rownames, obs are mixed together. I have to break the data into pieces then aggregate in a tidy format. Sometimes, obs are record in a string. So I should also split the string.

```
# Store url
# In the order of "sensory data, gold medal performance, brain weight vs body weight,
# triplicate measurements of tomato."
url_sensory<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
url_gold<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
url_brain<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
url_tomato<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
#download into local memory
Sensory<-read.csv(url_sensory,header =  F,skip = 2,sep = "")
#manipulate data into a tidy format
#transfer data into a vector
dim_ori<-dim(Sensory)
Sen_2=c()
for (i in 1:dim_ori[1]) {
  if (sum(i == seq(1,dim_ori[1],3)))
  {Sen_2=c(Sen_2,as.matrix(Sensory[i,2:dim_ori[2]]))}
  else
  {Sen_2=c(Sen_2,as.matrix(Sensory[i,1:(dim_ori[2]-1)]))}
}
#transfer the vector into a data.frame
Sen_2<-data.frame(matrix(Sen_2,nrow = dim_ori[1],ncol = dim_ori[2]-1))
#change the name of data
colnames(Sen_2)<-paste('Operator',as.character(1:5))

text_paste<-c()
for (i in 1:10) {
  text_paste<-c(text_paste,seq(i+0.1,i+0.3,0.1))
}
rownames(Sen_2)<-paste('Item',as.character(text_paste))
#Show table of Sensory data
Sen_2
```

```
##           Operator 1 Operator 2 Operator 3 Operator 4 Operator 5
## Item 1.1         4.3        2.4        5.7        1.2        8.0
```

```
## Item 1.2        4.9        2.5        6.3        1.5        8.6
## Item 1.3        3.3        2.3        5.4        1.2        9.0
## Item 2.1        5.3        3.1        6.1        0.9        9.4
## Item 2.2        4.4        2.4        5.9        0.7        8.8
## Item 2.3        4.3        3.9        5.8        1.3        9.0
## Item 3.1        4.5        3.0        5.7        2.4        7.7
## Item 3.2        4.0        2.8        5.4        0.8        6.7
## Item 3.3        5.5        2.7        6.2        1.2        9.0
## Item 4.1        3.3        1.3        6.5        1.3        7.9
## Item 4.2        4.1        1.9        5.8        0.9        8.9
## Item 4.3        5.3        3.9        6.0        3.1        9.2
## Item 5.1        3.4        2.6        6.1        1.1        8.1
## Item 5.2        5.7        4.6        7.0        1.9        9.1
## Item 5.3        4.7        2.2        4.9        1.6        7.6
## Item 6.1        6.0        7.4        2.2        4.2        5.0
## Item 6.2        5.3        8.2        2.4        4.8        4.8
## Item 6.3        4.5        6.4        1.7        4.5        3.9
## Item 7.1        5.9        6.8        3.4        4.6        5.5
## Item 7.2        4.7        6.0        1.7        3.2        3.8
## Item 7.3        4.9        7.1        3.0        3.0        5.4
## Item 8.1        6.3        7.9        1.8        4.5        5.0
## Item 8.2        4.2        5.9        2.1        4.7        3.4
## Item 8.3        5.5        7.3        4.0        4.9        4.9
## Item 9.1        4.9        6.1        1.7        4.6        4.6
## Item 9.2        6.0        6.4        2.1        4.8        2.8
## Item 9.3        5.9        7.1        3.3        4.8        5.2
## Item 10.1       4.7        6.9        1.1        4.7        4.1
## Item 10.2       6.3        7.0        3.3        4.8        3.9
## Item 10.3       4.6        6.7        2.1        4.3        5.5
```

```
str(Sen_2)
```

```
## 'data.frame':    30 obs. of  5 variables:
##  $ Operator 1: num  4.3 4.9 3.3 5.3 4.4 4.3 4.5 4 5.5 3.3 ...
##  $ Operator 2: num  2.4 2.5 2.3 3.1 2.4 3.9 3 2.8 2.7 1.3 ...
##  $ Operator 3: num  5.7 6.3 5.4 6.1 5.9 5.8 5.7 5.4 6.2 6.5 ...
##  $ Operator 4: num  1.2 1.5 1.2 0.9 0.7 1.3 2.4 0.8 1.2 1.3 ...
##  $ Operator 5: num  8 8.6 9 9.4 8.8 9 7.7 6.7 9 7.9 ...
```

```
summary(Sen_2)
```

```
##    Operator 1      Operator 2      Operator 3      Operator 4
##  Min.   :3.300   Min.   :1.300   Min.   :1.100   Min.   :0.700
##  1st Qu.:4.325   1st Qu.:2.625   1st Qu.:2.125   1st Qu.:1.225
##  Median :4.800   Median :5.250   Median :4.450   Median :3.050
##  Mean   :4.890   Mean   :4.827   Mean   :4.157   Mean   :2.917
##  3rd Qu.:5.500   3rd Qu.:6.875   3rd Qu.:5.875   3rd Qu.:4.600
##  Max.   :6.300   Max.   :8.200   Max.   :7.000   Max.   :4.900
##    Operator 5
##  Min.   :2.800
##  1st Qu.:4.825
##  Median :6.100
##  Mean   :6.493
##  3rd Qu.:8.750
##  Max.   :9.400
```

```r
#load gold medal performance dat
gold<-read.csv(url_gold,header =  F,skip = 1,sep = "")
#mung
dim_gold<-dim(gold)
gold_2=c()
for (i in 1:dim_gold[1]) {
  if (sum(i == c(5,6)))
  {gold_2=c(gold_2,as.matrix(gold[i,1:(dim_gold[2]-2)]))}
  else
  {gold_2=c(gold_2,as.matrix(gold[i,1:dim_gold[2]]))}
}
gold_2<-data.frame(matrix(gold_2,ncol = 2,nrow = 22,byrow = T))
colnames(gold_2)<-c('Year','Long Jump')
rownames(gold_2)<-1:dim(gold_2)[1]
gold_2[1]<-gold_2[1]+1900
#show
gold_2
```

```
##    Year Long Jump
## 1  1896    249.75
## 2  1924    293.13
## 3  1956    308.25
## 4  1980    336.25
## 5  1900    282.88
## 6  1928    304.75
## 7  1960    319.75
## 8  1984    336.25
## 9  1904    289.00
## 10 1932    300.75
## 11 1964    317.75
## 12 1988    343.25
## 13 1908    294.50
## 14 1936    317.31
## 15 1968    350.50
## 16 1992    342.50
## 17 1912    299.25
## 18 1948    308.00
## 19 1972    324.50
## 20 1920    281.50
## 21 1952    298.00
## 22 1976    328.50
```

```r
str(gold_2)
```

```
## 'data.frame':    22 obs. of  2 variables:
##  $ Year     : num  1896 1924 1956 1980 1900 ...
##  $ Long Jump: num  250 293 308 336 283 ...
```

```r
summary(gold_2)
```

```
##       Year        Long Jump
##  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
```

```
#load brain vs body weight dat
brain<-read.csv(url_brain,header = F,sep='',skip = 1)
#munge
dim_bra<-dim(brain)
braW_boW<-c()
for (i in 1:dim_bra[1]) {
  braW_boW<-c(braW_boW,as.matrix(brain[i,]))
}
not_na<-ifelse(is.na(braW_boW),F,T)
braW_boW<-braW_boW[not_na]
braW_boW<-data.frame(matrix(braW_boW,ncol=2,nrow = 62))
colnames(braW_boW)<-c('Body Weight','Brain Weight')
rownames(braW_boW)<-1:dim(braW_boW)[1]
#show
print(braW_boW)
```

```
##    Body Weight Brain Weight
## 1        3.385       62.000
## 2       44.500     1320.000
## 3      521.000        0.075
## 4      655.000        1.200
## 5        2.500        0.920
## 6       12.100        5.700
## 7        0.480     6654.000
## 8       15.500     5712.000
## 9        0.785        0.122
## 10       3.500        3.000
## 11      55.500        1.000
## 12     175.000        6.600
## 13       1.350        3.500
## 14       8.100        3.900
## 15      10.000        0.048
## 16     115.000        0.330
## 17     100.000        0.005
## 18     157.000        0.100
## 19     465.000        6.800
## 20     423.000      179.000
## 21       3.300      192.000
## 22      25.600      180.000
## 23      52.160        0.060
## 24     440.000        1.000
## 25      36.330       35.000
## 26     119.500       56.000
## 27       0.200        3.000
## 28       5.000       25.000
## 29      10.550        3.500
## 30     179.500       10.800
## 31      27.660        4.050
## 32     115.000       17.000
## 33       1.410      160.000
## 34      17.500      169.000
## 35       0.550        2.000
```

```
## 36       2.400       12.300
## 37      14.830        0.120
## 38      98.200        1.000
## 39     529.000        0.900
## 40     680.000        2.600
## 41      60.000        1.700
## 42      81.000        6.300
## 43       1.040        0.023
## 44       5.500        0.400
## 45     207.000        1.620
## 46     406.000       11.400
## 47       3.600     2547.000
## 48      21.000     4603.000
## 49       4.190        0.010
## 50      58.000        0.300
## 51      85.000        0.104
## 52     325.000        2.500
## 53       4.288        0.023
## 54      39.200        0.300
## 55       0.425        1.400
## 56       6.400       12.500
## 57       0.750        4.235
## 58      12.300       50.400
## 59       0.280      187.100
## 60       1.900      419.000
## 61       0.101      250.000
## 62       4.000      490.000
```

```
str(braW_boW)
```

```
## 'data.frame':    62 obs. of  2 variables:
##  $ Body Weight : num  3.38 44.5 521 655 2.5 ...
##  $ Brain Weight: num  62 1320 0.075 1.2 0.92 ...
```

```
summary(braW_boW)
```

```
##   Body Weight       Brain Weight
##  Min.   :  0.101   Min.   :   0.005
##  1st Qu.:  3.414   1st Qu.:   0.905
##  Median : 16.500   Median :   3.700
##  Mean   :104.103   Mean   : 377.822
##  3rd Qu.:111.250   3rd Qu.:  54.600
##  Max.   :680.000   Max.   :6654.000
```

```
#download tomato dat
tomato<-read.csv(url_tomato ,sep='',skip = 1,header = F)
#munge
t_colname<-as.matrix(paste('S_',as.character(seq(10000,30000,10000)),sep=""))
t_rowname<-as.matrix(tomato$V1[2:3])
t_meas<-c(as.matrix(tomato$V2[2]),as.matrix(tomato$V2[3]),as.matrix(tomato$V3[2]),as.matrix(tomato$V3[3]
t_meas<-data.frame(matrix(t_meas,ncol = 3,nrow = 2))
colnames(t_meas)<-t_colname

t_meas<-tbl_df(t_meas)
t_meas<-t_meas %>% mutate(S_10000=as.character(S_10000),S_20000=as.character(S_20000),S_30000=as.charac
t_meas<-t_meas %>% separate(S_10000, into = c('first_10000','Second_10000',"Thrid_10000"), sep=",", extr
```

```
t_meas<-data.frame(t_meas)
rownames(t_meas)<-t_rowname
#show and summary
t_meas
```

```
##                first_10000 Second_10000 Thrid_10000 first_20000
## Ife\\#1               16.1         15.3        17.5        16.6
## PusaEarlyDwarf         8.1          8.6        10.1        12.7
##                Second_20000 Thrid_20000 first_30000 Second_30000
## Ife\\#1                19.2        18.5        20.8         18.0
## PusaEarlyDwarf         13.7        11.5        14.4         15.4
##                Thrid_30000
## Ife\\#1               21.0
## PusaEarlyDwarf        13.7
```

```
str(t_meas)
```

```
## 'data.frame':    2 obs. of  9 variables:
##  $ first_10000 : chr  "16.1" "8.1"
##  $ Second_10000: chr  "15.3" "8.6"
##  $ Thrid_10000 : chr  "17.5" "10.1"
##  $ first_20000 : chr  "16.6" "12.7"
##  $ Second_20000: chr  "19.2" "13.7"
##  $ Thrid_20000 : chr  "18.5" "11.5"
##  $ first_30000 : chr  "20.8" "14.4"
##  $ Second_30000: chr  "18.0" "15.4"
##  $ Thrid_30000 : chr  "21.0" "13.7"
```

```
summary(t_meas)
```

```
##  first_10000        Second_10000       Thrid_10000
##  Length:2           Length:2           Length:2
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##  first_20000        Second_20000       Thrid_20000
##  Length:2           Length:2           Length:2
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##  first_30000        Second_30000       Thrid_30000
##  Length:2           Length:2           Length:2
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

P5

```
#Relationship between PH and Foliage_Color
#munge
#since there are a huge amount of missing values in pH_Min, pH_Max, and Foliage_Color and fill these NA
#After testing, Foliage_Color has 4334 NAs, pH_Max and pH_Max each has 4327 NAs, and these NAs almost a
plants_adj<-data.frame(Foliage_Color=plants$Foliage_Color[!is.na(plants$Foliage_Color)],pH_Min=plants$pH

#Creat new Ph var by average of PH-min and PH-max
plants_adj<-tbl_df(plants_adj)
plants_adj<-plants_adj %>% mutate(Ave_ph=(pH_Min+pH_Max)/2) %>% mutate(norm_ave=Ave_ph-mean(Ave_ph))
```

```r
# Set number for each color in an order
plants_adj<-plants_adj %>%mutate(Foliage_Color=ifelse(Foliage_Color=='Dark Green',1,Foliage_Color),
Foliage_Color=ifelse(Foliage_Color=='Gray-Green',2,Foliage_Color),
Foliage_Color=ifelse(Foliage_Color=='Green',3,Foliage_Color),
Foliage_Color=ifelse(Foliage_Color=='Red',4,Foliage_Color),
Foliage_Color=ifelse(Foliage_Color=='White-Gray',5,Foliage_Color),
Foliage_Color=ifelse(Foliage_Color=='Yellow-Green',6,Foliage_Color))

#Build Reg model
reg_Plant<-lm(Foliage_Color~norm_ave,plants_adj)

#Coefficients
reg_Plant
```

```
##
## Call:
## lm(formula = Foliage_Color ~ norm_ave, data = plants_adj)
##
## Coefficients:
## (Intercept)     norm_ave
##     2.87139      0.05227
```

```r
#Anova report
anova(reg_Plant)
```

```
## Analysis of Variance Table
##
## Response: Foliage_Color
##            Df Sum Sq Mean Sq F value Pr(>F)
## norm_ave    1   0.67 0.67118  0.9973 0.3182
## Residuals 830 558.57 0.67297
```