

# HW3\_GUO

*Qing Guo*

9/12/2019

## Problem 3

a. Sensory data from five operators

```
#install.packages("tibble")
library(tibble)
#install.packages("dplyr")
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

url<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
a<-read.table(url,header=T,sep=" ",skip=1,fill=T)
a1<-filter(a,!is.na(X5))
a2<-filter(a,is.na(X5))
Item<-c()
for(i in 1:10){Item<-c(Item,rep(i,2))}
a22<-a2%>%select(Item:X4)%>%colnames<-`(c("X1", "X2", "X3", 'X4', 'X5'))
a22<-cbind(Item,a22)
a<-rbind(a22,a1)
a%>%arrange(Item)

##   Item  X1  X2  X3  X4  X5
## 1     1  4.3  4.5  4.0  5.5  3.3
## 2     1  4.1  5.3  3.4  5.7  4.7
## 3     1  4.3  4.9  3.3  5.3  4.4
## 4     2  4.9  6.3  4.2  5.5  4.9
## 5     2  6.0  5.9  4.7  6.3  4.6
## 6     2  6.0  5.3  4.5  5.9  4.7
## 7     3  3.9  3.0  2.8  2.7  1.3
## 8     3  1.9  3.9  2.6  4.6  2.2
## 9     3  2.4  2.5  2.3  3.1  2.4
## 10    4  7.1  7.9  5.9  7.3  6.1
## 11    4  6.4  7.1  6.9  7.0  6.7
## 12    4  7.4  8.2  6.4  6.8  6.0
## 13    5  5.8  5.7  5.4  6.2  6.5
## 14    5  5.8  6.0  6.1  7.0  4.9
## 15    5  5.7  6.3  5.4  6.1  5.9
## 16    6  3.0  1.8  2.1  4.0  1.7
## 17    6  2.1  3.3  1.1  3.3  2.1
## 18    6  2.2  2.4  1.7  3.4  1.7
```

```
## 19    7 1.3 2.4 0.8 1.2 1.3
## 20    7 0.9 3.1 1.1 1.9 1.6
## 21    7 1.2 1.5 1.2 0.9 0.7
## 22    8 3.0 4.5 4.7 4.9 4.6
## 23    8 4.8 4.8 4.7 4.8 4.3
## 24    8 4.2 4.8 4.5 4.6 3.2
## 25    9 9.0 7.7 6.7 9.0 7.9
## 26    9 8.9 9.2 8.1 9.1 7.6
## 27    9 8.0 8.6 9.0 9.4 8.8
## 28   10 5.4 5.0 3.4 4.9 4.6
## 29   10 2.8 5.2 4.1 3.9 5.5
## 30   10 5.0 4.8 3.9 5.5 3.8
```

b. Gold Medal performance

```
url<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
a<-read.table(url,sep=" ",skip=1,fill=T)
a=as.data.frame(a)
a<-rbind(as.matrix(a[,1:2]),as.matrix(a[,3:4]),as.matrix(a[,5:6]),as.matrix(a[,7:8]))
a_new<-a%>%as.data.frame()%>%filter(!is.na(V1&V2))%>%colnames<-c("year","long_jump")
summarise_at(a_new, vars(long_jump),list(~n(), ~mean(., na.rm = TRUE),~median(.,na.rm = TRUE)))
```

```
##      n      mean  median
## 1 22 310.2873 308.125
```

```
summary(lm(long_jump~year,a_new))
```

```
##
## Call:
## lm(formula = long_jump ~ year, data = a_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.4665  -4.4821  -0.8236   6.3372  24.2246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  278.05315     4.25339   65.372 < 2e-16 ***
## year          0.70915     0.07801    9.091 1.53e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.02 on 20 degrees of freedom
## Multiple R-squared:  0.8052, Adjusted R-squared:  0.7954
## F-statistic: 82.64 on 1 and 20 DF,  p-value: 1.532e-08
```

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

```
url<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
a<-read.table(url,sep=" ",skip=1,fill=T)
a<-rbind(as.matrix(a[,1:2]),as.matrix(a[,3:4]),as.matrix(a[,5:6]))
a<-a[1:62,]
a<-as.data.frame(a)
colnames(a)<-c("Body WT","Brain WT")
a
```

```
##      Body WT Brain WT
```

## 1	3.385	44.50
## 2	0.480	15.50
## 3	1.350	8.10
## 4	465.000	423.00
## 5	36.330	119.50
## 6	27.660	115.00
## 7	14.830	98.20
## 8	1.040	5.50
## 9	4.190	58.00
## 10	0.425	6.40
## 11	0.101	4.00
## 12	0.920	5.70
## 13	1.000	6.60
## 14	0.005	0.10
## 15	0.060	1.00
## 16	3.500	10.80
## 17	2.000	12.30
## 18	1.700	6.30
## 19	2547.000	4603.00
## 20	0.023	0.30
## 21	187.100	419.00
## 22	521.000	655.00
## 23	0.785	3.50
## 24	10.000	115.00
## 25	3.300	25.60
## 26	0.200	5.00
## 27	1.410	17.50
## 28	529.000	680.00
## 29	207.000	406.00
## 30	85.000	325.00
## 31	0.750	12.30
## 32	62.000	1320.00
## 33	6654.000	5712.00
## 34	3.500	3.90
## 35	6.800	179.00
## 36	35.000	56.00
## 37	4.050	17.00
## 38	0.120	1.00
## 39	0.023	0.40
## 40	0.010	0.30
## 41	1.400	12.50
## 42	250.000	490.00
## 43	2.500	12.10
## 44	55.500	175.00
## 45	100.000	157.00
## 46	52.160	440.00
## 47	10.550	179.50
## 48	0.550	2.40
## 49	60.000	81.00
## 50	3.600	21.00
## 51	4.288	39.20
## 52	0.280	1.90
## 53	0.075	1.20
## 54	0.122	3.00

```
## 55    0.048    0.33
## 56  192.000   180.00
## 57    3.000   25.00
## 58  160.000  169.00
## 59    0.900    2.60
## 60    1.620   11.40
## 61    0.104    2.50
## 62    4.235   50.40
```

```
summarise_at(a, vars(c("Body WT", "Brain WT")), list(~n(), ~mean(., na.rm = TRUE), ~median(., na.rm = TRUE))
```

```
##   Body WT_n Brain WT_n Body WT_mean Brain WT_mean Body WT_median
## 1         62         62      198.79      283.1344         3.3425
##   Brain WT_median
## 1              17.25
```

d. Triplicate measurements of tomato yield

```
library(stringr)
url<-"https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
a<-read.table(url, sep=" ", skip=1, fill=T, comment.char="")
a<-as.data.frame(a)
d<-c()
for (i in 1:2){
  for (j in 1:3){
    bi_j<-a[i,j]
    bi_j<-unlist(str_split(bi_j, " "))
    bi_j<-as.numeric(bi_j[1:3])
    d<-c(d, bi_j)
  }
}
b1<-d[1:9]
b2<-d[10:18]
b3<-c(rep(10000, 3), rep(20000, 3), rep(30000, 3))
dataframe<-tibble(b3, b1, b2)
colnames(dataframe)<-c("Number", "Ife\\#1", "PusaEarlyDwarf")
dataframe<-as.data.frame(dataframe)
summarise_at(dataframe, vars(c("Ife\\#1", "PusaEarlyDwarf")), list(~n(), ~mean(., na.rm = TRUE), ~median(
```

```
##   Ife\\#1_n PusaEarlyDwarf_n Ife\\#1_mean PusaEarlyDwarf_mean
## 1         9              9    -18.11111    -12.02222
##   Ife\\#1_median PusaEarlyDwarf_median
## 1             -18             -12.7
```

```
t = dataframe %>% group_by(Number) %>%
  summarise_at(vars("Ife\\#1": "PusaEarlyDwarf"), funs(n(), mean(., na.rm = TRUE)))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
```

```
## Please use a list of either functions or lambdas:
```

```
##
```

```
##   # Simple named list:
```

```
##   list(mean = mean, median = median)
```

```
##
```

```
##   # Auto named with `tibble::lst()`:
```

```
##   tibble::lst(mean, median)
```

```
##
```

```
##   # Using lambdas
```

```
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```
t
```

```
## # A tibble: 3 x 5
##   Number `lfe`\\#1_n` PusaEarlyDwarf_n `lfe`\\#1_mean` PusaEarlyDwarf_mean
##   <dbl>      <int>      <int>      <dbl>      <dbl>
## 1 -30000         3         3      -19.9      -14.5
## 2 -20000         3         3      -18.1      -12.6
## 3 -10000         3         3      -16.3      -8.93
```

## Praoblem 5

```
#install.packages("swirl")
library(swirl)
```

```
##
## | Hi! I see that you have some variables saved in your workspace. To keep
## | things running smoothly, I recommend you clean up before starting swirl.
##
## | Type ls() to see a list of the variables in your workspace. Then, type
## | rm(list=ls()) to clear your workspace.
##
## | Type swirl() when you are ready to begin.
```

```
install_course("R_Programming_E")
```

```
##
|
| 0%
|
|== 3%
|
|=== 4%
|
|===== 8%
|
|===== 10%
|
|===== 12%
|
|===== 15%
|
|===== 19%
|
|===== 23%
|
|===== 24%
|
|===== 26%
|
|===== 27%
|
|===== 28%
|
```

=====	32%
=====	36%
=====	40%
=====	42%
=====	44%
=====	48%
=====	51%
=====	52%
=====	53%
=====	53%
=====	54%
=====	55%
=====	57%
=====	58%
=====	60%
=====	61%
=====	65%
=====	66%
=====	67%
=====	71%
=====	72%
=====	74%
=====	77%
=====	79%
=====	83%
=====	85%
=====	87%

```

|=====| 89%
|=====| 91%
|=====| 95%
|=====| 100%

##
## | Course installed successfully!

swirl()
bye()

##
## | Leaving swirl now. Type swirl() to resume.

# Path to data
.datapath <- file.path(path.package('swirl'), 'Courses',
                        'R_Programming_E', 'Looking_at_Data',
                        'plant-data.txt')

# Read in data
plants <- read.csv(.datapath, strip.white=TRUE, na.strings="")
.cols2rm <- c('Accepted.Symbol', 'Synonym.Symbol')
plants <- plants[, !(names(plants) %in% .cols2rm)]
# Make names pretty
names(plants) <- c('Scientific_Name', 'Duration', 'Active_Growth_Period',
                  'Foliage_Color', 'pH_Min', 'pH_Max',
                  'Precip_Min', 'Precip_Max',
                  'Shade_Tolerance', 'Temp_Min_F')
na_plants<-apply(is.na(plants),1,sum)
plants<-plants[-which((na_plants)>0),]
plants_1<-plants%>%select(c('Scientific_Name','Foliage_Color', 'pH_Min', 'pH_Max'))
plants<-plants_1
plants$PH_Mean<-(plants$pH_Min+plants$pH_Max)/2
plants$Foliage_Color<-factor(plants$Foliage_Color)
head(plants)

##      Scientific_Name Foliage_Color pH_Min pH_Max PH_Mean
## 4      Abies balsamea      Green    4.0    6.0    5.00
## 9      Acacia constricta      Green    7.0    8.5    7.75
## 14     Acalypha virginica      Green    5.9    7.0    6.45
## 17      Acer negundo      Green    5.0    7.8    6.40
## 19      Acer nigrum      Green    4.5    7.3    5.90
## 20      Acer pensylvanicum      Green    4.4    6.5    5.45

aggregate(plants$PH_Mean,by=list(plants$Foliage_Color),FUN=mean)

##      Group.1      x
## 1  Dark Green 5.999390
## 2  Gray-Green 6.370833
## 3      Green 6.175111
## 4      Red 6.400000
## 5  White-Gray 6.444444
## 6 Yellow-Green 5.937500

```

```
aggregate(plants$PH_Mean,by=list(plants$Foliage_Color),FUN=sd)
```

```
##          Group.1          x
## 1   Dark Green 0.5560272
## 2   Gray-Green 0.6387686
## 3      Green 0.5250715
## 4      Red 0.9836158
## 5  White-Gray 0.7380059
## 6 Yellow-Green 0.6043428
```

```
plants.anova<-aov(plants$PH_Mean~plants$Foliage_Color,data=plants)
F1<-summary(plants.anova)
F1
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## plants$Foliage_Color    5    5.23   1.0452    3.613 0.00308 **
## Residuals              807 233.48   0.2893
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plants$Foliage_Color<-plants$Foliage_Color%>%as.factor()%>%as.numeric()
F2<-lm(plants$PH_Mean~plants$Foliage_Color,data=plants)
F2<-summary(F2)
F2
```

```
##
## Call:
## lm(formula = plants$PH_Mean ~ plants$Foliage_Color, data = plants)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.93924 -0.36441 -0.01441  0.33559  2.03559
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.08959    0.06858  88.792  <2e-16 ***
## plants$Foliage_Color 0.02494    0.02297   1.086   0.278
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5421 on 811 degrees of freedom
## Multiple R-squared:  0.001451,    Adjusted R-squared:  0.0002202
## F-statistic: 1.179 on 1 and 811 DF,  p-value: 0.2779
```