HW2_Do_Quyen

Quyen Do

September 4, 2018

Problem 4

Version control helps me manage and control my coding assignments. If I treat each assignment as a project. Version control helps me back up and save the progress of the assignment as I'm completing it. It is a way for me to keep the current codes while enabling backtracking to older versions if my new codes break. In a long run, version control helps me retain my codes, i.e. my work throught the course, so that I can always look back at what I did and got reminded of what I have learned.

Problem 5

a. Sensory data from five operators.

This data is messy because the columns of "Operator 1", "Operator 2", etc. are also values of an "Operator" variable. The value of variable "item" are stored in multiple rows.

```
#First 10 rows of the data
kable(head(sensory_cleaned,10),caption="First 10 rows of cleaned Sensory data")
```

Table 1: First 10 rows of cleaned Sensory data

Item	Operator	Measurement
1	1	4.3
1	1	4.3
1	1	4.1
2	1	6.0
2	1	4.9
2	1	6.0
3	1	2.4
3	1	3.9
3	1	1.9
4	1	7.4

```
#Create summary table
kable(summary(sensory_cleaned),caption="Sensory Data Summary")
```

Table 2: Sensory Data Summary

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

b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0.

This data is messy because variables "Long Jump" and "Year" are stored in multiple columns.

```
#First 10 rows of the cleaned data
kable(head(medal_cleaned,10),caption="First 10 rows of cleaned Long Jump data")
```

Table 3: First 10 rows of cleaned Long Jump data

Year	Long Jump
-4	249.75
0	282.88
4	289.00
8	294.50
12	299.25
20	281.50
24	293.13
28	304.75
32	300.75
36	317.31

```
#Create a summary table
kable(summary(medal_cleaned),caption="Long Jump Gold Medal Data Summary")
```

Table 4: Long Jump Gold Medal Data Summary

Year	Long Jump
Min. :-4.00	Min. :249.8
1st Qu.:21.00	1st Qu.:295.4
Median :50.00	Median :308.1
Mean :45.45	Mean :310.3
3rd Qu.:71.00	3rd Qu.:327.5
Max. :92.00	Max. :350.5

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

The data is messy because the variables "Body Weight" and "Brain Weight" are stored in multiple columns.

```
#Show the first 10 rows of the cleaned data
kable(head(bodyBrain_cleaned,10),caption="First 10 rows of cleaned Brain Wt and Body Wt Data")
```

Table 5: First 10 rows of cleaned Brain Wt and Body Wt Data

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
14.830	98.2
1.040	5.5
4.190	58.0

Body Wt	Brain Wt
0.425	6.4

```
#Create summary table
kable(summary(bodyBrain_cleaned),caption="Body and Brain Weight Data Summary")
```

Table 6: Body and Brain Weight Data Summary

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.: 48.203	3rd Qu.: 166.00
Max. :6654.000	Max. :5712.00

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

The data is messy because the columns of 10000, 20000, and 30000 are actually values of variable "density". The cell of the table holds three different values of the yield, whereas each cell should only hold one value of certain variable.

```
#Show the first 10 rows of the cleaned data
kable(head(tomato_cleaned,10),caption="First 10 rows of cleaned tomato data")
```

Table 7: First 10 rows of cleaned tomato data

Variety	Density	Yield
Ife#1	10000	16.1
Ife#1	10000	15.3
Ife#1	10000	17.5
Ife#1	20000	16.6
Ife#1	20000	19.2
Ife#1	20000	18.5
Ife#1	30000	20.8
Ife#1	30000	18.0
Ife#1	30000	21.0
PusaEarlyDwarf	10000	8.1

```
#Create a summary table
kable(summary(tomato_cleaned),caption="Tomato Data Summary")
```

Table 8: Tomato Data Summary

Variety	Density	Yield
Length:18	Min. :10000	Min. : 8.10
Class :character	1st Qu.:10000	1st Qu.:12.95
Mode :character	Median :20000	Median :15.35
NA	Mean :20000	Mean :15.07
NA	3rd Qu.:30000	3rd Qu.:17.88
NA	Max. :30000	Max. :21.00

Problem 6

Appendix

```
#Import raw data from url
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
sensory_raw <- read_csv(url,skip=1)

#Set up partial_cleaned table (as recorded by the researcher)
columns_names <- c("Item","Operator 1","Operator 2", "Operator 3", "Operator 4","Operator 5")
sensory_partial_cleaned <- data.frame(matrix(nrow=nrow(sensory_raw),ncol=length(columns_names)))
colnames(sensory_partial_cleaned) <- columns_names

##To keep track of the current item the row is referred to
current_item <- 0

for (i in 1:nrow(sensory_raw))
{
  row <- sensory_raw[i,][[1]]
  #Split the row by space
  row_data <- strsplit(row,split=" ")[[1]]
  row_data <- as.numeric(row_data)

  # If row_data contain 6 figures, the first figure must be the item number of that row and the next fe
  if (length(row_data) == 6){
    current_item <- row_data[1]

    #Get rid of item number from the row_data
    row_data <- row_data[-1]
  }
  sensory_partial_cleaned[i,] <- c(current_item,row_data)
}

#Turn sensory_partial_cleaned to a cleaned table
sensory_cleaned <- gather(sensory_partial_cleaned,key="Operator",value="Measurement",
                          "Operator 1","Operator 2","Operator 3","Operator 4","Operator 5")
sensory_cleaned$Operator <- strsplit(sensory_cleaned$Operator,split=" ")[[1]][2]

#First 10 rows of the data
kable(head(sensory_cleaned,10),caption="First 10 rows of cleaned Sensory data")

#Create summary table
kable(summary(sensory_cleaned),caption="Sensory Data Summary")

url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
medal_raw <- read.csv(url,header=FALSE, sep=" ",skip=1)
names(medal_raw) <- c("Year 1","LongJump 1", "Year 2","LongJump 2","Year 3", "LongJump 3","Year 4","Long

#medal_raw table is messy for the fact there are 8 columns depicting 2 variables. The 4 Year columns ne

#Create a clean dataframe with 2 variables
medal_cleaned <- data.frame(matrix(nrow=0,ncol=2))
```

```r
#Subsequently extract Year and corresponding LongJump columns from raw table and bind them onto the cle
medal_cleaned <- rbind(medal_cleaned,medal_raw[,c("Year 1","LongJump 1")])

names(medal_cleaned) <- c("Year 2","LongJump 2")
medal_cleaned <- rbind(medal_cleaned,medal_raw[,c("Year 2","LongJump 2")])

names(medal_cleaned) <- c("Year 3","LongJump 3")
medal_cleaned <- rbind(medal_cleaned,medal_raw[,c("Year 3","LongJump 3")])

names(medal_cleaned) <- c("Year 4","LongJump 4")
medal_cleaned <- rbind(medal_cleaned,medal_raw[,c("Year 4","LongJump 4")])


#Format clean table
names(medal_cleaned) <- c("Year","Long Jump")

#Remove NA row
medal_cleaned<- medal_cleaned[-c(23,24),]
#First 10 rows of the cleaned data
kable(head(medal_cleaned,10),caption="First 10 rows of cleaned Long Jump data")
#Create a summary table
kable(summary(medal_cleaned),caption="Long Jump Gold Medal Data Summary")

#Import data into a raw table
url <- "http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
bodyBrain_raw <- read.csv(url,sep=" ",header=FALSE,skip=1)
names(bodyBrain_raw) <- c("Body Wt 1","Brain Wt 1","Body Wt 2","Brain Wt 2","Body Wt 3","Brain Wt 3")

#bodyBrain_raw table is messy for the fact there are 6 columns depicting 2 variables. The 3 Body Wt col

#Create a clean dataframe with 2 variables
bodyBrain_cleaned <- data.frame(matrix(nrow=0,ncol=2))

#Subsequently extract Brain Wt and corresponding Body Wt columns from raw table and bind them onto the
bodyBrain_cleaned <- rbind(bodyBrain_cleaned,bodyBrain_raw[,c("Body Wt 1","Brain Wt 1")])

names(bodyBrain_cleaned) <- c("Body Wt 2","Brain Wt 2")
bodyBrain_cleaned <- rbind(bodyBrain_cleaned,bodyBrain_raw[,c("Body Wt 2","Brain Wt 2")])

names(bodyBrain_cleaned) <- c("Body Wt 3","Brain Wt 3")
bodyBrain_cleaned <- rbind(bodyBrain_cleaned,bodyBrain_raw[,c("Body Wt 3","Brain Wt 3")])

#Format clean table
names(bodyBrain_cleaned) <- c("Body Wt","Brain Wt")

#Remove NA row
bodyBrain_cleaned<- bodyBrain_cleaned[-63,]
#Show the first 10 rows of the cleaned data
kable(head(bodyBrain_cleaned,10),caption="First 10 rows of cleaned Brain Wt and Body Wt Data")
#Create summary table
kable(summary(bodyBrain_cleaned),caption="Body and Brain Weight Data Summary")
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
```

```r
tomato_raw <- read_lines(url)[-c(1,2)]

#Set up cleaned table
columns_names <- c("Variety","Density","Yield")
tomato_cleaned <- data.frame(matrix(nrow=0,ncol=length(columns_names)))
added_row <- data.frame(matrix(nrow=1,ncol=length(columns_names)))
colnames(added_row) <- columns_names
colnames(tomato_cleaned) <- columns_names

#Run through each line from the data to extract information
for (line in tomato_raw)
{
  #split each line by white space character
  line <- strsplit(line,split=" ")[[1]]

  #get rid of white space elements
  line <- line[line!=""]

  #extract the information
  variety <- line[1]
  density_10000<- as.numeric(strsplit(line[2],split=",")[[1]])
  density_20000 <- as.numeric(strsplit(line[3],split=",")[[1]])
  density_30000 <- as.numeric(strsplit(line[4],split=",")[[1]])

  for (i in density_10000)
  {
    added_row$Variety <- variety
    added_row$Density <- 10000
    added_row$Yield <- i
    tomato_cleaned <- rbind(tomato_cleaned,added_row)
  }

  for (i in density_20000)
  {
    added_row$Variety <- variety
    added_row$Density <- 20000
    added_row$Yield <- i
    tomato_cleaned <- rbind(tomato_cleaned,added_row)
  }

  for (i in density_30000)
  {
    added_row$Variety <- variety
    added_row$Density <- 30000
    added_row$Yield <- i
    tomato_cleaned <- rbind(tomato_cleaned,added_row)
  }
}
#Show the first 10 rows of the cleaned data
kable(head(tomato_cleaned,10),caption="First 10 rows of cleaned tomato data")
#Create a summary table
kable(summary(tomato_cleaned),caption="Tomato Data Summary")
```

```r
# Path to data
library(swirl)
.datapath <- file.path(path.package('swirl'), 'Courses',
                       'R_Programming', 'Looking_at_Data',
                       'plant-data.txt')
# Read in data
plants <- read.csv(.datapath, strip.white=TRUE, na.strings="")

# Remove annoying columns
.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')

# Consider the range of pH
plants$pH_range <- plants$pH_Max-plants$pH_Min
plants_working <- plants[,c('Foliage_Color','pH_range')]
plants_working_noNA <- na.omit(plants_working)

#Build regression model of pH_range using Foliage_Color as independent variable
plants_working_noNA$Foliage_Color <- as.factor(plants_working_noNA$Foliage_Color)
plants_model <- lm(pH_range~Foliage_Color,data=plants_working_noNA)
#Produce coefficients in table form
kable(summary(plants_model)$coefficients,caption="Coefficients Table of Plants Model")

#Produce ANOVA in table form
kable(anova(plants_model),caption = "ANOVA Table of Plants Model")
```