STATS419 Survey of Multivariate Analysis

Week 03 Assignment

Minju Lee
(minju.lee@wsu.edu)
[11638071]

Instructor: Monte J. Shaffer

21 September 2020

0.1 1. Create the "rotate matrix" functions as described in lectures. Apply to the example "myMatrix".

```
myMatrix = matrix (c (
                            1,0,2,
                            0,3,0,
                            4,0,5
                            ), nrow=3,
byrow = T);


transposeMatrix = function(mat)
  {
  t(mat);
  }

transposeMatrix(myMatrix)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    4
## [2,]    0    3    0
## [3,]    2    0    5
```

```
#I need to revere the rows befor applying transpose

rotateMatrix90 = function(mat)
  {
  t(mat[nrow(myMatrix):1,])
  }

rotateMatrix90(myMatrix)
```

```
##      [,1] [,2] [,3]
## [1,]    4    0    1
## [2,]    0    3    0
## [3,]    5    0    2
```

```
rotateMatrix180 = function(mat)
  {
  t(rotateMatrix90(mat)[nrow(rotateMatrix90(mat)):1,])
  }

rotateMatrix180(myMatrix)
```

```
##      [,1] [,2] [,3]
## [1,]    5    0    4
## [2,]    0    3    0
## [3,]    2    0    1
```

```
rotateMatrix270 = function(mat)
  {
  t(rotateMatrix180(mat)[nrow(rotateMatrix180(mat)):1,])
  }

rotateMatrix270(myMatrix)
```

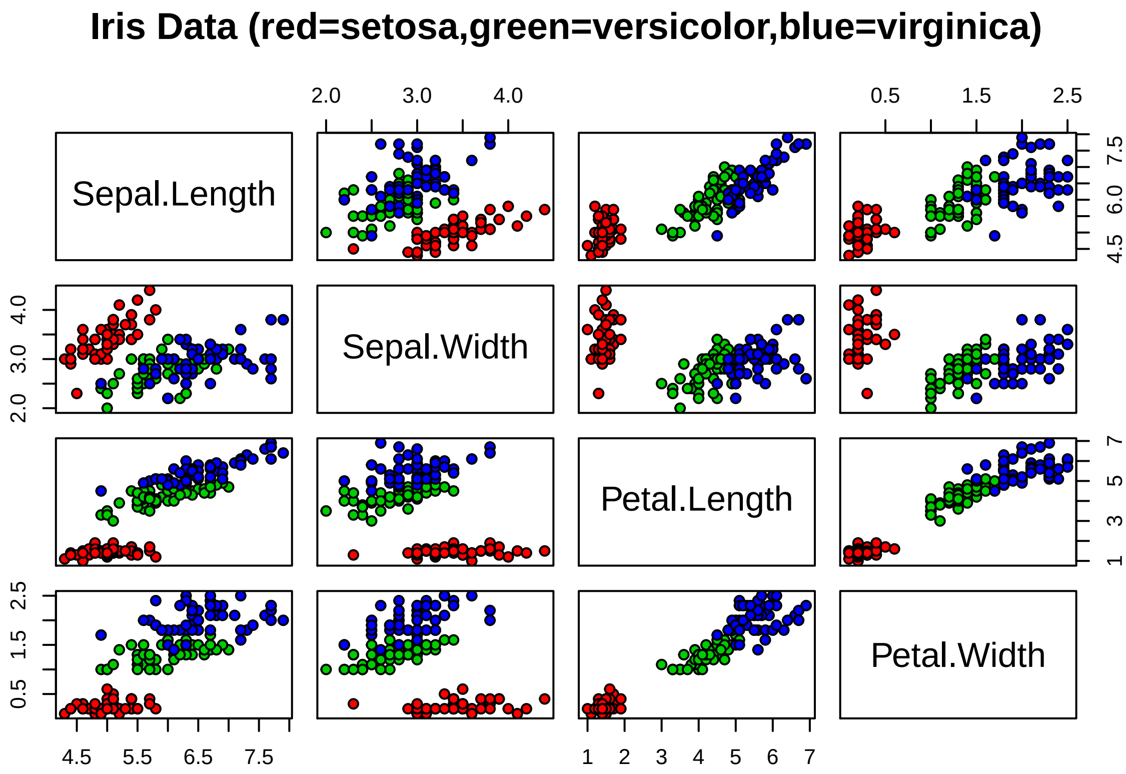
```
##      [,1] [,2] [,3]
## [1,]    2    0    5
## [2,]    0    3    0
## [3,]    1    0    4
```

0.2 2. Recreate the graphic for the IRIS Data Set using R. Same titles, same scales, same colors. See: https://en.wikipedia.org/wiki/Iris_flower_data_set#/media/File:Iris_dataset_scatterplot.svg

```
#getting the data and investigate the data
data(iris)
#dim(iris)
#attributes(iris)
#summary(iris)

#create the graph
# This code can be found at https://commons.wikimedia.org/wiki/File:Iris_dataset_scatterplot.svg

pairs(iris[1:4],main="Iris Data (red=setosa,green=versicolor,blue=virginica)", pch=21,
bg=c("red","green3","blue")[unclass(iris$Species)])
```



0.3 **3. Right 2-3 sentences concisely defining the IRIS Data Set. Maybe search KAGGLE for a nice template. Be certain the final writeup are your own sentences (make certain you modify what you find, make it your own, but also cite where you got your ideas from). NOTE: Watch the video, Figure 8 has a +5 EASTER EGG.**

0.4 **4. Import "personality-raw.txt" into R. Remove the V00 column. Create two new columns from the current column "date_test": year and week. Stack Overflow may help: https://stackoverflow.com/questions/22439540/how-to-get-week-numbers-from-dates . . . Sort the new data frame by YEAR, WEEK so the newest tests are first . . . The newest tests (e.g., 2020 or 2019) are at the top of the data frame. Then remove duplicates using the unique function based on the column "md5_email". Save the data frame in the same "pipe-delimited format" (| is a pipe) with the headers. You will keep the new data frame as "personality-clean.txt" for future work (you will not upload it at this time). In the homework, for this tasks, report how many records your raw dataset had and how many records your clean dataset has.**

The raw dataset had 838 rows. The clean dataset has 678 rows.

```
# read in the raw data
pers_raw <- read.table("personality-raw.txt", sep="|", header=TRUE)
#pers_raw
```

```
# remove V00 column
pers_new <- subset(pers_raw, select = -V00)
#pers_new

# create two new columns for year and week
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(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
pers_new$date_test <- mdy_hm(pers_new$date_test)
pers_new$year <- year(pers_new$date_test)
pers_new$week <- format(pers_new$date_test, "%m/%d")

# sort the data
pers_sort <- arrange(pers_new, desc(year), desc(week))

# remove duplicates
length(unique(pers_sort$md5_email))
```

```
## [1] 678
```

```
pers_clean <- pers_sort %>% distinct(pers_sort$md5_email, .keep_all=TRUE)
#pers_clean

ms <- filter(pers_clean, md5_email =='b62c73cdaf59e0a13de495b84030734e')

ms <- as.matrix(ms)
ms
```

```
##      md5_email                          date_test             V01   V02   V03
## [1,] "b62c73cdaf59e0a13de495b84030734e" "2020-04-06 12:57:00" "3.4" "4.2" "2.6"
##      V04   V05   V06   V07   V08   V09   V10   V11   V12   V13   V14   V15
## [1,] "4.2" "2.6" "2.6" "4.2" "2.6" "3.4" "4.2" "4.2" "3.4" "3.4" "4.2" "5"
##      V16   V17 V18   V19   V20   V21   V22   V23   V24   V25 V26   V27   V28
## [1,] "3.4" "5" "3.4" "1.8" "2.6" "2.6" "2.6" "4.2" "3.4" "5" "2.6" "4.2" "3.4"
##      V29   V30   V31   V32   V33   V34   V35   V36   V37   V38   V39   V40  
## [1,] "2.6" "2.6" "4.2" "1.8" "3.4" "4.2" "4.2" "4.2" "2.6" "4.2" "2.6" "4.2"
##      V41   V42   V43   V44   V45   V46   V47   V48   V49   V50   V51   V52  
## [1,] "4.2" "4.2" "4.2" "2.6" "4.2" "4.2" "2.6" "3.4" "2.6" "4.2" "1.8" "4.2"
##      V53   V54   V55   V56   V57   V58   V59   V60   year   week   
## [1,] "2.6" "3.4" "4.2" "4.2" "1.8" "4.2" "2.6" "4.2" "2020" "04/06"
##      pers_sort$md5_email               
## [1,] "b62c73cdaf59e0a13de495b84030734e"
```

```
# save cleaned data

# write.table(pers_clean, "personality-clean.txt", sep="|",row.names = FALSE)
```

0.5 5. Write functions for doSummary and sampleVariance and doMode ... test these functions in your homework on the "monte.shaffer@gmail.com" record from the clean dataset. Report your findings. For this "monte.shaffer@gmail.com" record, also create z-scores. Plot(x,y) where x is the raw scores for "monte.shaffer@gmail.com" and y is the z-scores from those raw scores. Include the plot in your assignment, and write 2 sentences describing what pattern you are seeing and why this pattern is present.

```
# Test on 'b62c73cdaf59e0a13de495b84030734e'

doSummary = function(x)
    {
    len <- length(x)
    mean <- mean(x)
    sd <-sd(x)
  #mean <- sum(x)/len
    #med <- median(as.numeric(x[,c(3:62)]))
  l <- c("Length: ",len)
  m <- c("Mean: ", mean)
  #me <- c("Median: ", med)
  s <- c("Standard Deviation: ", sd)
  print(l, quote=FALSE)
  print(m, quote=FALSE)
  #print(me, quote=FALSE)
  prtin(s, quote=FALSE)
  }

doSampleVariance = function(x, method)
    {
    if(method=="naive")
        {


        }
        else
            {
            # two-pass algorithm

            }
    }

doMode = function(x)
    {
    result = c();
        # freq ... # high freqencies
        # ties ... store all of the ties
        # which.min() or which.max()

    result;
    }
```

0.6 6. Compare Will Smith and Denzel Washington. [See 03_n greater 1-v2.txt for the necessary functions and will-vs-denzel.txt for some sample code and in DROPBOX: //student_access//unit_01_exploratory_data_analysis//week_02//imdb-example] You will have to create a new variable $millions.2000 that converts each movie's $millions based on the $year of the movie, so all dollars are in the same time frame. You will need inflation data from about 1980-2020 to make this work.

0.7 7. Build side-by-side box plots on several of the variables (including #6) to compare the two movie stars. After each box plot, write 2+ sentence describing what you are seeing, and what conclusions you can logically make. You will need to review what the box plot is showing with the box portion, the divider in the box, and the whiskers.