HW5_Xu_Peng

Peng Xu

2017/10/1

Problem 3

A good figure should demonstrate its contents clearly and neatly. It contains all the required elements, such as legend and caption. It is easy for viewers to obtain the overview.

It also contains important factors and their relationships. It can recover some important details and helps viewers to consider from different aspects.

It would be excellent if the figure exceeds viewers' expectation. Some valuable and surprising results could be revelt from the elements of the figure.

Problem 4

Part a

```
CountSuccess <- function(vec){
    count <- sum(vec)
    return(count)
}
```

Part b

```
set.seed(12345)
P4b_data <- matrix(rbinom(10, 1, prob = (30:40)/100), nrow = 10,
ncol = 10)
```

Part c

```
apply(P4b_data,MARGIN = 1, CountSuccess)
```

```
##  [1] 10 10 10 10  0  0  0  0 10 10
```

```
apply(P4b_data,MARGIN = 2, CountSuccess)
```

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

It is observed that the columns of matrix is simply ten replicates of one experiment set. The list of different probabilities's results are not generated.

Part d

```r
GenerateRandom <- function(p){
    return(rbinom(10, 1, prob = p))
}

ProbList <- c(30:40)/100
ProbList <- matrix(ProbList, nrow = 1)

ResultMatrix <- apply(ProbList, MARGIN = 2 ,GenerateRandom)
row.names(ResultMatrix)<-c('E1','E2','E3','E4','E5','E6','E7','E8','E9','E10')
ResultMatrix
```

```
##     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## E1     0    0    1    1    1    1    1    1    1     0     1
## E2     0    0    0    0    1    0    0    0    1     1     0
## E3     1    1    0    1    0    1    0    0    0     1     1
## E4     0    1    1    1    0    1    0    0    0     1     0
## E5     0    0    0    0    1    1    0    0    1     0     1
## E6     0    0    0    0    0    0    0    0    0     1     1
## E7     0    1    1    0    1    1    1    1    1     1     1
## E8     0    0    1    0    0    0    1    0    1     1     0
## E9     0    0    0    0    0    0    0    1    0     0     0
## E10    1    0    0    0    0    1    0    0    0     0     0
```

Then the vectors of different probabilites are shown above.

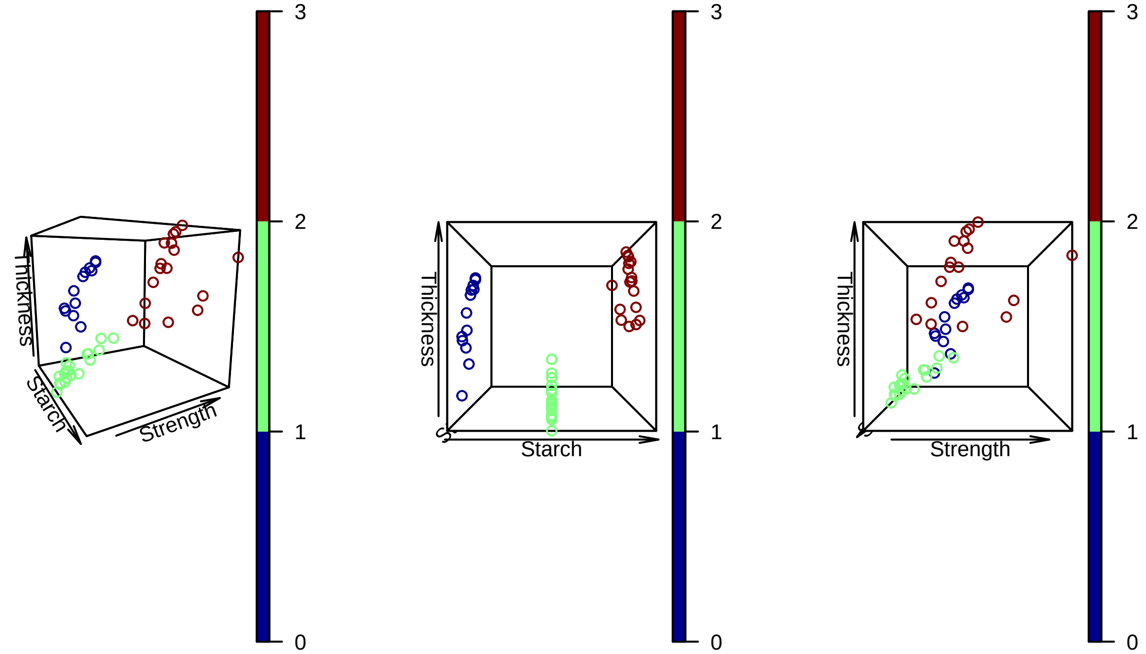
Problem 5

```r
library("plot3D")
```

```
## Warning: package 'plot3D' was built under R version 3.3.3
```

```r
URL <- "http://www2.isye.gatech.edu/~jeffwu/book/data/starch.dat"
Starch_raw<-read.table(URL, header=T, skip=0, fill=T, stringsAsFactors = F)

#Starch_raw$starch <- as.factor(Starch_raw$starch)

G<-matrix(c('CA',1,'CO',2,'PO',3),ncol=2,byrow=T)
for (i in 1:length(Starch_raw$starch)) {
  if (Starch_raw$starch[i]%in%G[,1]) {
  Starch_raw$starch[i]=G[which(G[,1]==Starch_raw$starch[i]),2]
  }
}

x <- as.numeric(Starch_raw$starch)
y <- Starch_raw$strength
z <- Starch_raw$thickness

par(mfrow=c(1,3))
scatter3D(x, y, z, colvar = x, phi = 10, theta = 60, col = NULL,
          breaks = c(0,1,2,3), xlab = 'Starch', ylab = 'Strength', zlab = 'Thickness')
scatter3D(x, y, z, colvar = x, phi = 0, theta = 0, col = NULL,
          breaks = c(0,1,2,3), xlab = 'Starch', ylab = 'Strength', zlab = 'Thickness')
```

```
scatter3D(x, y, z, colvar = x, phi = 0, theta = 90, col = NULL,
          breaks = c(0,1,2,3), xlab = 'Starch', ylab = 'Strength', zlab = 'Thickness')
```



Problem 6

Part a

```
states <- fread(input = "./us_cities_and_states/states.sql",
                skip = 19, sep = "'", sep2 = ",", header = F, select = c(2,4))
colnames(states) <- c('State','abbr')

cities1 <- fread(input = "./us_cities_and_states/cities.sql",
                skip = 19, sep = "'", sep2 = ",", header = F, select = c(2,4))
cities2 <- fread(input = "./us_cities_and_states/cities_extended.sql",
                skip = 19, sep = "'", sep2 = ",", header = F, select = c(2,4))

cities <- rbind(cities1,cities2)
colnames(cities) <- c('City','State')

CleanCities <- cities %>%
    group_by(City, State) %>%
    filter(row_number() == 1) %>%
    ungroup()
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

Part b

The number of cities of all states are shown below.

```
CityNum <- table(CleanCities$State)
CityNum <- CityNum[-40]
CityNum <- CityNum[-8]
CityNum
```

```
##
##   AK   AL   AR   AZ   CA   CO   CT   DE   FL   GA   HI   IA   ID   IL   IN
##  229  579  605  264 1239  400  269   57  524  629   92  937  266 1287  738
##   KS   KY   LA   MA   MD   ME   MI   MN   MO   MS   MT   NC   ND   NE   NH
##  634  803  479  511  430  461  885  810  942  440  360  762  373  528  255
##   NJ   NM   NV   NY   OH   OK   OR   PA   RI   SC   SD   TN   TX   UT   VA
##  579  346   99 1612 1069  585  379 1802   70  377  364  548 1466  250  839
##   VT   WA   WI   WV   WY
##  288  493  753  753  176
```

Part c

```
LetterCount <- function(string){
    LetterList <- c('a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u'
    Result <- as.table(rep(0,26))
    row.names(Result) <- LetterList

    sp <- strsplit(tolower(as.character(string)), split="")[[1]]
    for(i in 1:26){
        Result[i] <- length(which(sp == LetterList[i]))
    }
    return(Result)
}

#LetterCount('adsfasdfaa')
```

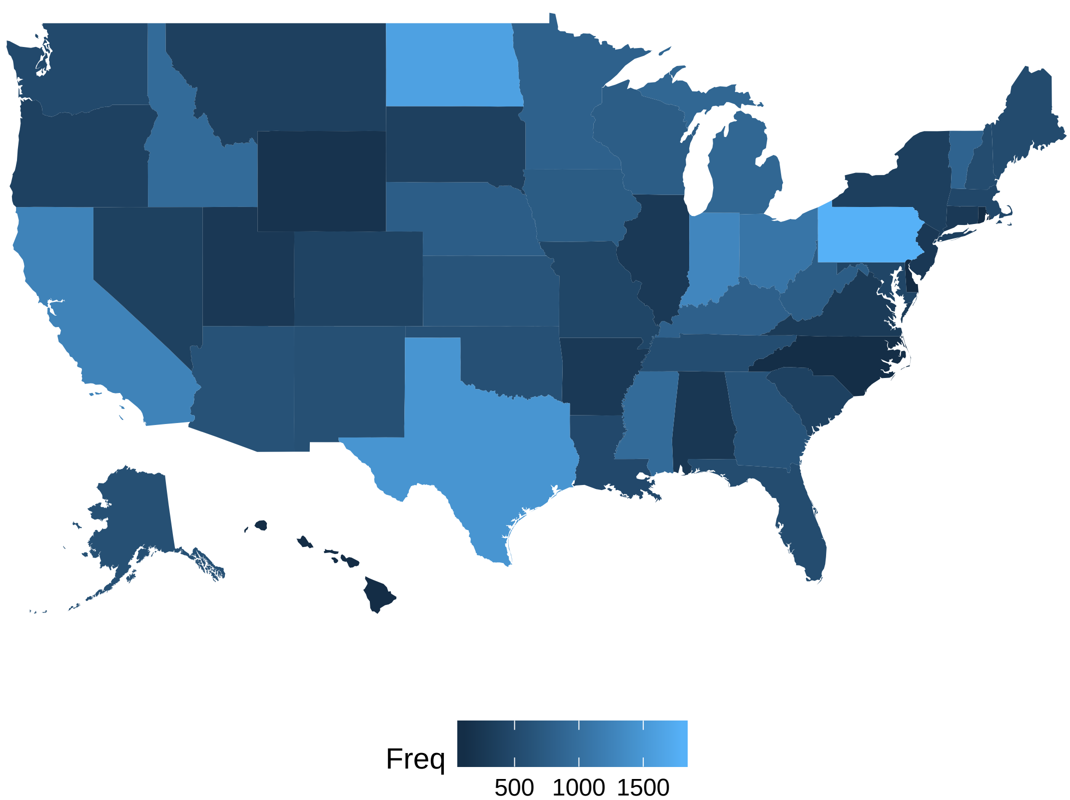
```
Results <- t(apply(as.matrix(states$State),MARGIN = 1,LetterCount))
Results <- Results[-8,]    # remove DC
head(Results)
```

```
##      a b c d e f g h i j k l m n o p q r s t u v w x y z
## [1,] 3 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## [2,] 4 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 3 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 2 0 0 0 0 0 0 0
## [4,] 2 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1
## [5,] 2 0 1 0 0 1 0 0 2 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0
## [6,] 1 0 1 1 0 0 0 0 0 0 0 1 0 0 3 0 0 1 0 0 0 0 0 0 0 0
```

```
temp <- apply(Results,MARGIN = 1, max)
ThreeLetterList <- floor(temp/3)
```

Part d

```r
data("fifty_states") # this line is optional due to lazy data loading
crimes <- data.frame(state = tolower(rownames(USArrests)),
USArrests)

StateData <- cbind(crimes,CityNum)
StateData <- cbind(StateData,ThreeLetterList)


# map_id creates the aesthetic mapping to the state name
# column in your data
p <- ggplot(StateData, aes(map_id = state)) + # map points to the fifty_states shape data
geom_map(aes(fill = Freq), map = fifty_states) + expand_limits(x = fifty_states$long,
y = fifty_states$lat) + coord_map() + scale_x_continuous(breaks = NULL) +
scale_y_continuous(breaks = NULL) + labs(x = "", y = "") +
theme(legend.position = "bottom", panel.background = element_blank())
p
```




```r
# ggsave(plot = p, file =
# 'HW5_Problem6_Plot_Settlage.pdf')
```

```r
p <- ggplot(StateData, aes(map_id = state)) + # map points to the fifty_states shape data
geom_map(aes(fill = ThreeLetterList), map = fifty_states) + expand_limits(x = fifty_states$long,
y = fifty_states$lat) + coord_map() + scale_x_continuous(breaks = NULL) +
scale_y_continuous(breaks = NULL) + labs(x = "", y = "") +
```

```r
theme(legend.position = "bottom", panel.background = element_blank())
p
```