STAT 579 Homework 4

Yifan Zhu

October 8, 2016

Problem 1

(a) read the data

```
# read the data and take apostroph as character
senate109 <- read.table(file = "http://maitra.public.iastate.edu/stat579/datasets/senate-109.txt",
    sep = "\t", quote = "\"", header = T)
```

(b) Bill type

i.

```
# remove the characters after "_"
billtype <- sub(pattern = "_.*", replacement = "", x = senate109$bill_type_bill_name_bill_ID)
```

ii. tabulate each type of bill

```
table(billtype)
```

```
## billtype
##                    Abortion Issues                  Agriculture Issues
##                                 19                                   7
##                     Appropriations                Arts and Humanities
##                                111                                   1
##           Budget, Spending and Taxes              Business and Consumers
##                                 75                                  11
## Campaign Finance and Election Issues                    Civil Liberties
##                                 13                                   7
##               Congressional Affairs                       Crime Issues
##                                 10                                   3
##                            Defense                        Drug Issues
##                                 29                                   2
##                          Education    Employment and Affirmative Action
##                                 16                                   2
##                      Energy Issues              Environmental Issues
##                                 18                                   8
##                   Executive Branch          Family and Children Issues
##                                 44                                   5
##        Foreign Aid and Policy Issues                  Government Reform
##                                 19                                   1
##                         Gun Issues                      Health Issues
##                                  6                                  22
##                        Immigration                              Labor
##                                 22                                   8
##                       Legal Issues                    Military Issues
##                                  9                                   6
```

```
##            National Security Issues                       Regulatory Issues
##                                  14                                       3
##         Science and Medical Research    Senior and Social Security Issues
##                                   1                                       7
##                       Social Issues          Technology and Communication
##                                   3                                       7
##                        Trade Issues                Transportation Issues
##                                  12                                      13
##                     Veterans Issues                   Welfare and Poverty
##                                   3                                       5
```

(c) Data quality

```r
# the matrix
X <- as.matrix(senate109[3: ncol(senate109)])

# extract the diagnal elements of XX'
XX.diag <- diag(X %*% t(X))

# add the missing votes

number.senators <- XX.diag + senate109$missing_votes

#check if there is any discrepancy

prod(number.senators == ncol(senate109) - 2)
```

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

From the result above we can see the returned vector should be all 1. Thus there is no discrepancy.

(d)

```r
leader <- senate109$William.H...Bill..Frist..TN.
# create the matrix using the leader's opnion
leader.matrix <- matrix(rep(x = leader, times = ncol(X)), ncol = ncol(X))
# determine whether other senator's for/against/indifferent with the
# leader's choice
trend <- X * leader.matrix
# remove the rows where the leader did not record a vote
trend <- trend[leader != 0, ]
trend <- trend[, -ncol(trend)]
```

(e)

```r
# count the for number in each column
fornum <- apply(X = trend == 1, MARGIN = 1, FUN = sum)
# count the against number in each column
againstnum <- apply(X = trend == -1, MARGIN = 1, FUN = sum)
# count the indifferent number in each column
indifferentnum <- apply(X = trend == 0, MARGIN = 1, FUN = sum)
```

```r
trendcount <- data.frame(bill_type = billtype[leader != 0], for_number = fornum,
    against_number = againstnum, indifferent_number = indifferentnum)


# using aggregate calculating the mean for each bill type
treandcount.mean <- aggregate(trendcount, by = list(trendcount$bill_type), FUN = mean)[-2]
treandcount.mean
```

```
##                                Group.1 for_number against_number
## 1                      Abortion Issues   35.78947       30.68421
## 2                   Agriculture Issues   36.71429       25.42857
## 3                       Appropriations   62.36364       14.05455
## 4                  Arts and Humanities   58.00000        7.00000
## 5            Budget, Spending and Taxes   43.98667       32.50667
## 6               Business and Consumers   56.63636       32.72727
## 7  Campaign Finance and Election Issues   41.46154       29.30769
## 8                      Civil Liberties   41.00000       23.57143
## 9                Congressional Affairs   51.60000       24.50000
## 10                       Crime Issues   36.33333       30.33333
## 11                            Defense   52.06897       18.89655
## 12                        Drug Issues   45.50000       16.00000
## 13                          Education   54.62500       19.25000
## 14   Employment and Affirmative Action   28.50000       46.00000
## 15                      Energy Issues   54.94444       33.27778
## 16               Environmental Issues   45.50000       30.62500
## 17                   Executive Branch   62.76744       14.09302
## 18           Family and Children Issues   64.00000       25.20000
## 19         Foreign Aid and Policy Issues   43.52632       23.36842
## 20                  Government Reform   70.00000        8.00000
## 21                         Gun Issues   61.00000       24.16667
## 22                      Health Issues   39.77273       33.36364
## 23                        Immigration   46.18182       29.77273
## 24                              Labor   34.00000       35.75000
## 25                       Legal Issues   46.77778       17.66667
## 26                    Military Issues   66.16667       25.83333
## 27            National Security Issues   62.14286       21.21429
## 28                  Regulatory Issues   36.33333       16.33333
## 29          Science and Medical Research   62.00000       37.00000
## 30   Senior and Social Security Issues   51.57143       35.00000
## 31                      Social Issues   48.66667       31.00000
## 32          Technology and Communication   58.57143       10.28571
## 33                       Trade Issues   53.00000       27.16667
## 34             Transportation Issues   55.07692       11.53846
## 35                    Veterans Issues   52.00000       45.00000
## 36                 Welfare and Poverty   37.50000       12.75000
##    indifferent_number
## 1           32.526316
## 2           36.857143
## 3           22.581818
## 4           34.000000
## 5           22.506667
## 6            9.636364
## 7           28.230769
```

```
## 8             34.428571
## 9             22.900000
## 10            32.333333
## 11            28.034483
## 12            37.500000
## 13            25.125000
## 14            24.500000
## 15            10.777778
## 16            22.875000
## 17            22.139535
## 18             9.800000
## 19            32.105263
## 20            21.000000
## 21            13.833333
## 22            25.863636
## 23            23.045455
## 24            29.250000
## 25            34.555556
## 26             7.000000
## 27            15.642857
## 28            46.333333
## 29             0.000000
## 30            12.428571
## 31            19.333333
## 32            30.142857
## 33            18.833333
## 34            32.384615
## 35             2.000000
## 36            48.750000
```

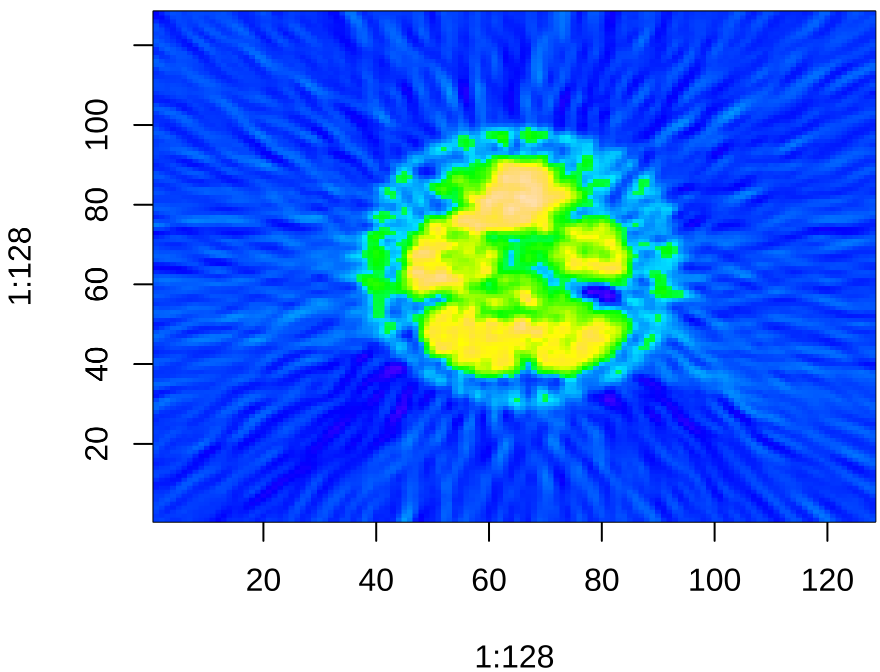
Problem 2

(a) Read the data as matrix

```
PET <- matrix(scan(file = "http://maitra.public.iastate.edu/stat579/datasets/fbp-img.dat"),
    ncol = 128, nrow = 128, byrow = T)
```

(b) image the plot

```
image(1:128, 1:128, PET[, 128:1], col = topo.colors(128^2))
```



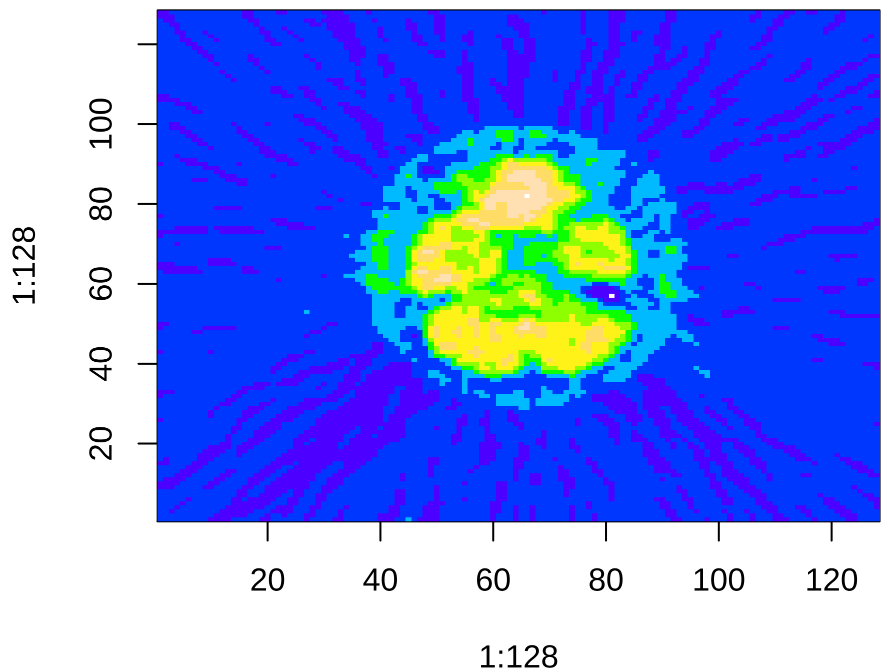
(c) i.

```
range <- max(PET) - min(PET)
bin <- seq(min(PET), max(PET), length.out = 9)
# vector of the mid points
bin.mid <- (bin[-1] + bin[- length(bin)]) / 2

# function to determine which bin is the value x in and return the mid point
deterbin <- function(x){
  y <- bin.mid[(bin[-1] > x)&((bin[-length(bin)] < x))]
  return(y)
}
PET_new1 <- apply(PET, MARGIN = c(1,2), FUN = deterbin)
```

```
mode(PET_new1) <- 'numeric'

image(1:128, 1:128, PET_new1[, 128:1], col = topo.colors(128^2))
```

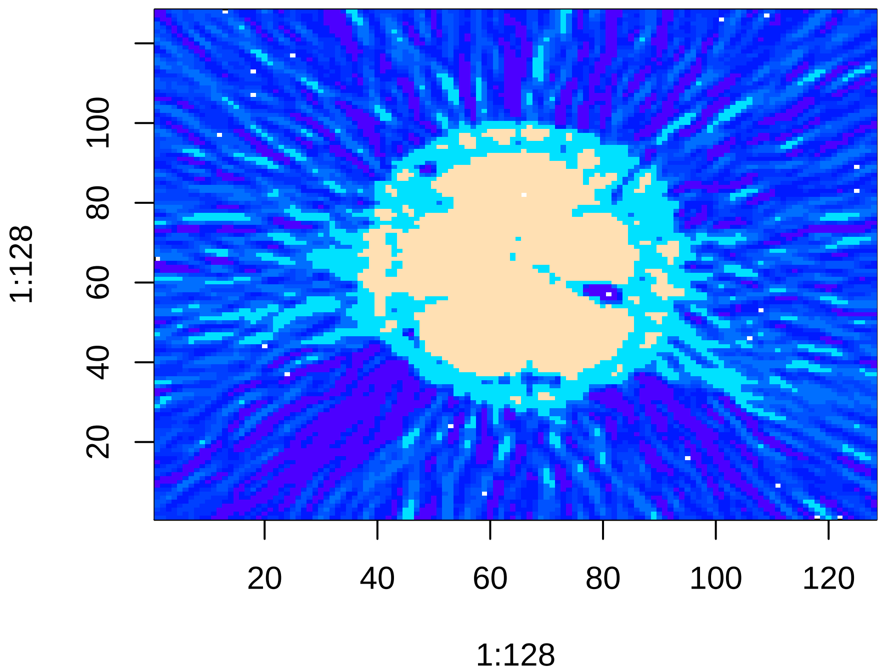


ii.

```
qtbin <- quantile(PET, probs = seq(0, 1, length = 9))
# vector of the mid points
qtbin.mid <- (qtbin[-1] + qtbin[- length(qtbin)]) / 2

# function to determine which quantile bin is the value x in and return the mid point
deterqtbin <- function(x){
  y <- qtbin.mid[(qtbin[-1] > x)&((qtbin[-length(qtbin)] < x))]
  return(y)
}
PET_new2 <- apply(PET, MARGIN = c(1,2), FUN = deterqtbin)
mode(PET_new2) <- 'numeric'

image(1:128, 1:128, PET_new2[, 128:1], col = topo.colors(128^2))
```



iii. Comment: These two methods can both discribe the image, but in the detail of the background, the method of quantile bin is better. In my opinion, using quantile bin is better. This method reflects the concentration of data and can discribe the image better.

Problem 3

(a) read the data

```
auto <- read.table(file = "C:/Users/fanne/Desktop/STAT579/Auto.txt", header = T)
```

(b)

```
X <- matrix(c(auto$horsepower, rep(1, length(auto$horsepower))), ncol = 2)
Y <- auto$mpg

beta_hat <- solve(t(X) %*% X) %*% t(X) %*% Y
beta_hat
```

```
##             [,1]
## [1,] -0.1578447
## [2,] 39.9358610
```

(c)

```
lmbeta <- lm(formula = mpg ~ horsepower, data = auto)
summary(lmbeta)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

The result of these two methods are similar.