AMS597_HW4_Solution_Spring2024

Question 1(a)

```
dat <- read.delim('https://www.ams.sunysb.edu/~pfkuan/Teaching/AMS597/Data/leukemiaDataSet.txt',
                  header=T,sep='\t')
#str(dat) not running to save space
## 72 observations and 3572 variables; only group variable is chr, the rest are num
set.seed(123)
trainID <- sample(1:72,round(0.7*72))
trainData <- dat[trainID,]
testData <- dat[-trainID,]
```

Question 1(b)

This dataset has 72 observations and 3572 variables. The number of variables is greater than the number of observations, so we will use penalized regression (can use either LASSO or elastic net).

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
trainX <- model.matrix(Group ~., trainData)[, -1]
trainY <- as.factor(trainData$Group)

# LASSO
cv.lasso <- cv.glmnet(trainX, trainY, alpha=1, nfolds=5, family="binomial")
cv.lasso$lambda.min
```

```
## [1] 0.004251892
```

```
model.lasso <- glmnet(trainX, trainY, alpha=1, family="binomial", lambda=cv.lasso$lambda.min)

## Selected Genes
c <- coef(model.lasso, s='lambda.min')
inds <- which(coef(model.lasso) != 0)
variables <- row.names(c)[inds]
(variables<-variables[!(variables %in% '(Intercept)')])
```



```
##  [1] "Gene456"  "Gene657"  "Gene672"  "Gene888"  "Gene956"  "Gene979"
##  [7] "Gene1219" "Gene1865" "Gene1946" "Gene2141" "Gene2230" "Gene2268"
## [13] "Gene2387" "Gene2481" "Gene2727" "Gene2859" "Gene3038" "Gene3098"
## [19] "Gene3158" "Gene3181"
```

```
# Elastic Net
alph_lamb <- data.frame(matrix(ncol=3, nrow=0))
alphavec = seq(0,1,0.1)

for (i in 1:length(alphavec)){
  fit <- cv.glmnet(trainX, trainY, family="binomial", alpha=alphavec[i], nfolds = 10)
  min_mse <- min(fit$cvm)
  lam <- fit$lambda[which(fit$cvm == min_mse)]
  alph_lamb <- rbind(alph_lamb, c(alphavec[i], lam, min_mse))
}
colnames(alph_lamb) <- c("alpha", "lambda", "cvm")
alph_lamb
```

```
##    alpha      lambda       cvm
## 1    0.0 4.251891657 0.3702788
## 2    0.1 0.042518917 0.2211761
## 3    0.2 0.021259458 0.2141442
## 4    0.3 0.014172972 0.2147562
## 5    0.4 0.010629729 0.2562608
## 6    0.5 0.008503783 0.2368164
## 7    0.6 0.007086486 0.2437063
## 8    0.7 0.006074131 0.1893075
## 9    0.8 0.005314865 0.2174901
## 10   0.9 0.004724324 0.3252489
## 11   1.0 0.004251892 0.3430809
```

```
# Find the predictors with non-zero coefficients in the final model
min_cvm <- min(alph_lamb$cvm)
opt_alpha <- alph_lamb[which(alph_lamb$cvm == min_cvm),1]
opt_lambda <- alph_lamb[which(alph_lamb$cvm == min_cvm),2]

# Print results
cat("\nThe (alpha, lambda) pair that corresponds to minimum cross-validation error is: \n\n(",
opt_alpha, ",", opt_lambda, ")\n\n")
```

```
##
## The (alpha, lambda) pair that corresponds to minimum cross-validation error is:
##
## ( 0.7 , 0.006074131 )
```

```
model.elastic <- glmnet(trainX, trainY, family="binomial", alpha=opt_alpha, lambda=opt_lambda)

fitCoefs <- coef(model.elastic)

terms <- which(fitCoefs != 0)
cat("\nThe predictors that are retained in the final model are: \n", rownames(fitCoefs)[terms], "\n", s
```



```
##
## The predictors that are retained in the final model are:
##
## (Intercept)
## Gene178
## Gene219
## Gene278
## Gene376
## Gene456
## Gene626
## Gene657
## Gene672
## Gene874
## Gene888
## Gene918
## Gene956
## Gene979
## Gene1007
## Gene1027
## Gene1099
## Gene1182
## Gene1187
## Gene1219
## Gene1569
## Gene1620
## Gene1652
## Gene1749
## Gene1780
## Gene1796
## Gene1835
## Gene1865
## Gene1946
## Gene2064
## Gene2141
## Gene2198
## Gene2230
## Gene2239
## Gene2268
## Gene2387
## Gene2481
## Gene2537
## Gene2546
## Gene2571
## Gene2727
## Gene2859
## Gene2888
## Gene3038
## Gene3098
## Gene3158
## Gene3201
## Gene3216
## Gene3218
## Gene3292
## Gene3441
```


Question 1(c)

```
# LASSO
testX <- model.matrix(Group ~., testData)[,-1]
pred_test_lasso <- predict(model.lasso, newx = testX, s='lambda.min', type = "class")
# AML percentage
(sum(pred_test_lasso == "AML" & testData$Group == "AML"))/sum(testData$Group == "AML")
```

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

```
# ALL percentage
(sum(pred_test_lasso == "ALL" & testData$Group == "ALL"))/sum(testData$Group == "ALL")
```

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

```
# overall
sum((pred_test_lasso==testData$Group))/nrow(testX)
```

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

```
# Elastic Net
testX <- model.matrix(Group ~., testData)[,-1]
pred_test_elastic <- predict(model.elastic, newx = testX, s='lambda.min', type = "class")
# AML percentage
(sum(pred_test_elastic == "AML" & testData$Group == "AML"))/sum(testData$Group == "AML")
```

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

```
# ALL percentage
(sum(pred_test_elastic == "ALL" & testData$Group == "ALL"))/sum(testData$Group == "ALL")
```

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

```
# overall
sum((pred_test_elastic==testData$Group))/nrow(testX)
```

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

Question 2

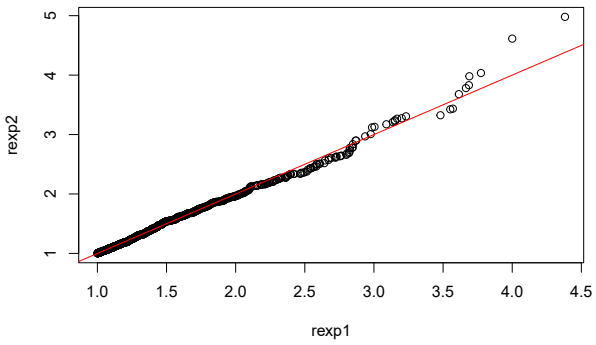
$$u = F(x) = \int_v^x \lambda e^{-\lambda(t-v)} dt = [-e^{-\lambda(t-v)}]_v^x = 1 - e^{-\lambda(x-v)}$$

$$x = v - \frac{log(1-u)}{\lambda}$$

Note that if u is a standard uniform random variable then $1-u$ is also a standard uniform random variable.

```r
my.rexp <- function(n, lambda, v){
  u <- runif(n)
  x <- v - log(u)/(lambda) # can also use v - log(1-u)/(lambda)
  return(x)
}

rexp1 <- my.rexp(1000, 2, 1)
library(tolerance)
rexp2 <- r2exp(1000, rate = .5, shift = 1)
{qqplot(rexp1, rexp2)
abline(0,1,col='red')}
```



Question 3

$$u = \frac{1}{\pi}\int_{-\infty}^{x}\frac{1}{1+t^2}dt = \frac{1}{\pi}[\tan^{-1}(t)]_{-\infty}^{x} = \frac{1}{\pi}\tan^{-1}(x) + \frac{1}{2}$$

$$x = \tan(\pi(u - \frac{1}{2}))$$

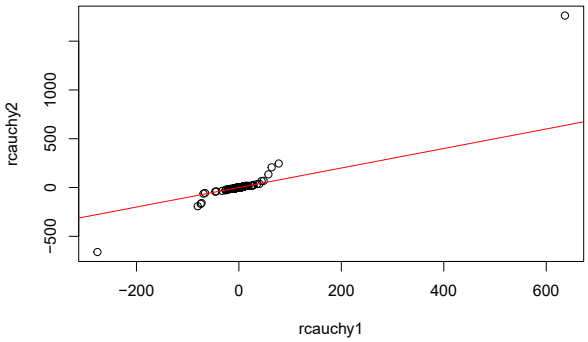
```r
my.rcauchy <- function(n){
  u <- runif(n)
  x <- tan(pi*(u-0.5))
  return(x)
```



```r
}

rcauchy1 <- my.rcauchy(1000)
rcauchy2 <- rcauchy(1000)

qqplot(rcauchy1, rcauchy2)
abline(0,1,col='red')
```



Question 4

If $Y \sim \text{Beta}(a, b)$ for $a, b > 1$, the pdf is

$$f(y) = \frac{y^{a-1}(1-y)^{b-1}}{B(a,b)}, y \in [0,1]$$

Find when $f(y)$ is maximized by taking the derivative and setting it equal to 0.

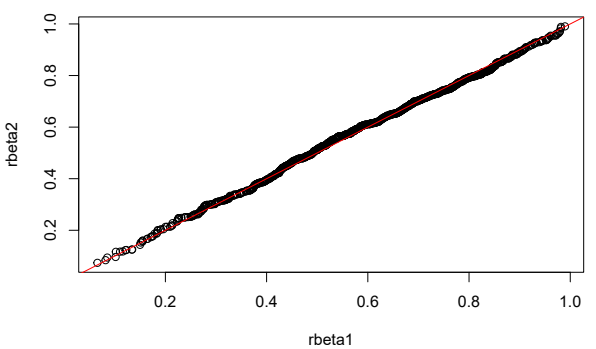
$$y_{\max} = \frac{a-1}{a+b-2}$$

We choose $c = f(y_{\max}) = f\left(\frac{a-1}{a+b-2}\right)$ and $g(y) = 1$.

$$U \leq \frac{f(y)}{cg(y)} = \frac{y^{a-1}(1-y)^{b-1}}{cB(a,b)} =$$


```r
my.rbeta = function(n,a,b){
  k <- 0 ## counting number of acceptance
  x <- rep(NA,n)
  x.max = (a-1) / (a+b-2)
  while(k<n){
      y <- runif(1) ## step1
      u <- runif(1) ## step2
      if(u <= y^(a-1)*(1-y)^(b-1)/(x.max^(a-1)*(1-x.max)^(b-1))){ ##step3
          k <- k+1
          x[k] <- y
      }
  }
return(x)
}

rbeta1 <-my.rbeta(1000,3,2)
rbeta2 <- rbeta(1000,3,2)
{qqplot(rbeta1, rbeta2)
abline(0,1,col='red')}
```



Question 5

We want to generate a random sample from $f(y) \sim \text{Gamma}(\alpha, 1)$.

$$f(y) = \frac{y^{\alpha-1}e^{-y}}{\Gamma(\alpha)}$$

We want to use $g(y) \sim \exp(\frac{1}{\alpha})$ as our proposal distribution.

$$g(y) = \frac{1}{\alpha}e^{-\frac{1}{\alpha}y}$$

Choose y such that $\frac{f(y)}{g(y)}$ is maximized (take the derivative and set it equal to 0, check if it is a maximum using second derivative test).

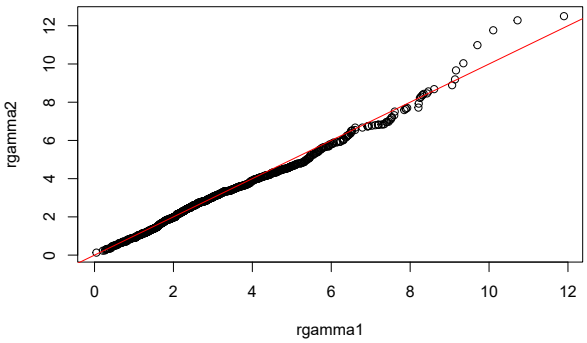
$$\frac{f(y)}{g(y)} = \frac{\alpha y^{\alpha-1}e^{y(\frac{1}{\alpha}-1)}}{\Gamma(\alpha)}$$

This is maximized when $y = \alpha$. Hence, $c = \frac{\alpha^{\alpha}e^{1-\alpha}}{\Gamma(\alpha)}$.

$$U \leq \frac{f(y)}{cg(y)} = \frac{\alpha y^{\alpha-1}e^{y(\frac{1}{\alpha}-1)}}{\frac{\alpha^{\alpha}e^{1-\alpha}}{\Gamma(\alpha)}\Gamma(\alpha)} = \alpha^{1-\alpha}y^{\alpha-1}e^{\alpha-1+\frac{1}{\alpha}y-y}$$

```r
my.rgamma = function(n,a){
  k <- 0 ## counting number of acceptance
  x <- rep(NA,n)
  while(k<n){
    y <- rexp(1,1/a) ## step1
    u <- runif(1) ## step2
    if(u <= exp(a-1-y+(y/a))*(y/a)^(a-1)){ ## step3
      k <- k+1
      x[k] <- y
    }
  }
  return(x)
}

rgamma1<- my.rgamma(1000,3)
rgamma2 <- rgamma(1000,3)
{qqplot(rgamma1, rgamma2)
abline(0,1,col='red')}
```


rgamma2
rgamma1
0
2
4
6
8
10
12