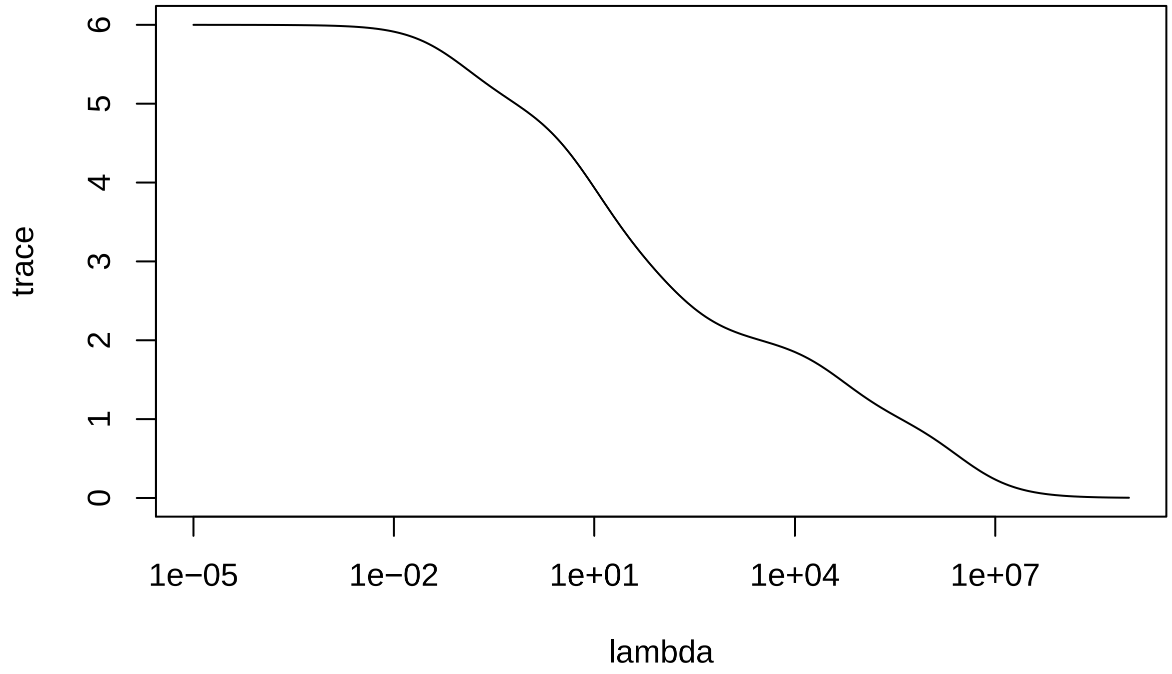
Coding and data analysis exercises

1.

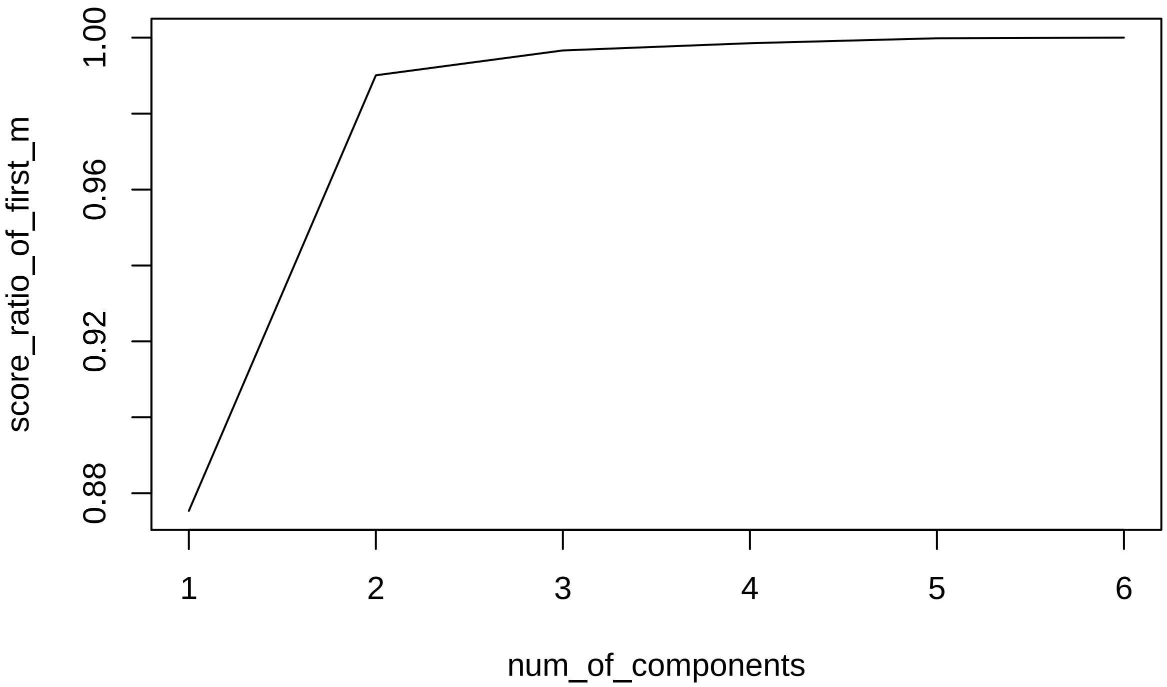
```r
require(stats)

myridge <- function(X, y, lambda){

    Design = cbind(1, X)
    s = svd(Design)
    D = s$d
    U = s$u
    V = s$v

    trace = c()
    beta = c()
    for(l in lambda){
     trace = c(trace, sum(D^2 / (D^2 + l)))
     beta = cbind(beta, V %*% diag((D^2 + l)^-1) %*% diag(D) %*% t(U) %*% y)
    }

    plot(lambda, trace, type = "l", log = "x")

    return(beta)
}

mtcars_selected = as.matrix(mtcars[c("mpg","cyl","disp","hp","drat","wt")])
X = mtcars_selected[,2:6]
y = mtcars_selected[,1]
lambda = 10 ^ seq(-5,9,length.out = 1000)
dull = myridge(X, y, lambda)
```



2.

This function `mypcr` use principle components regression and return with two lists of parameter `gamma` and fitted values `fiited_values` of every choice of largest several components. Along with a plot.

```
require(stats)

mypcr <- function(X, y){

    Design = cbind(1, X)
    s = svd(Design)
    D = s$d
    U = s$u
    V = s$v
    Z = U %*% diag(D)

    scores = c()
    score_ratio_of_first_m = c()
    gamma = list()
    fitted_values = list()
    num_of_components = 1:length(D)
    for(d in D){
     scores = c(scores, d)
     m = length(scores)
     score_ratio_of_first_m = c(score_ratio_of_first_m, sum(scores)/sum(D))
     if(m == 1){
         g = matrix(1/D[1:m]) %*% t(U[,1:m]) %*% y
         gamma[[m]] = g
     }
     else{
         g = diag(1/D[1:m]) %*% t(U[,1:m]) %*% y
         gamma[[m]] = g
     }
     f = U[,1:m] %*% t(U[,1:m]) %*% y
     fitted_values[[m]] = f
    }

    plot(num_of_components, score_ratio_of_first_m, type = "l")

    return(list("components" = Z,
                "gamma" = gamma,
                "fitted_values" = fitted_values,
                "ratios" = score_ratio_of_first_m))
}

mtcars_selected = as.matrix(mtcars[c("mpg","cyl","disp","hp","drat","wt")])
X = mtcars_selected[,2:6]
y = mtcars_selected[,1]
dull = mypcr(X, y)
```


score_ratio_of_first_m
num_of_components