---

title: "Lab 4"

author: "Feiyi Ma"

output: pdf_document

date: "11:59PM March 10, 2021"

Load up the famous iris dataset. We are going to do a different prediction problem. Imagine the only input x is Species and you are trying to predict y which is Petal.Length. A reasonable prediction is the average petal length within each Species. Prove that this is the OLS model by fitting an appropriate `lm` and then using the predict function to verify.

```{r}
data(iris)
mod=lm(Petal.Length~Species,iris)
mean(iris$Petal.Length[iris$Species=="setosa"])
mean(iris$Petal.Length[iris$Species=="versicolor"])
mean(iris$Petal.Length[iris$Species=="virginica"])

predict(mod,data.frame(Species = c("setosa")))
predict(mod,data.frame(Species = c("versicolor")))
predict(mod,data.frame(Species = c("virginica")))

#pacman::p_load(ggplot2)
#ggplot(iris) + geom_boxplot(aes(x = Species, y = Petal.Length))
#dataset give all other condition the same, like sunshine, water... limited effect on others   /// geom_boxplot(aes(x = Species, y = Petal.Length))
#geom_boxplot(aes(x = 1, y = Petal.Length))
```

Construct the design matrix with an intercept, X, without using `model.matrix`. #matrix for each data in x

```{r}
#X <- cbind(1,iris$Species)
#head(X)  #view the species as continuous variable, make setosa=1 versicolor=2  virginica=3, which is wrong
X <- cbind(1,iris$Species=="versicolor",iris$Species=="virginica")
head(X)
```

Find the hat matrix H for this regression.

```{r}
H = X %*% solve(t(X)%*% X) %*% t(X)
Matrix::rankMatrix(H)#find independent column, linearly independent column REM (remainer)
#head(H)
#rank(H) # linear of combination of x, of each 3 in x Rank =3
```

Verify this hat matrix is symmetric using the `expect_equal` function in the package `testthat`.

```{r}
pacman::p_load(testthat)
expect_equal(H,t(H))
```

Verify this hat matrix is idempotent using the `expect_equal` function in the package `testthat`.

```{r}

expect_equal(H,H%*%H)
```

Using the `diag` function, find the trace of the hat matrix.

```{r}
sum(diag(H))  #diag only print out the diag vaue of that 0.02
#sum of agenv alue = 3
```

It turns out the trace of a hat matrix is the same as its rank! But we don't have time to prove these interesting and useful facts..

For masters students: create a matrix $X_\perp$.

```{r}
#TO-DO
```

Using the hat matrix, compute the $\hat{y}$ vector and using the projection onto the residual space, compute the e vector and verify they are orthogonal to each other.

```{r}
y = iris$Petal.Length
y
y_hat = H %*% y
e = (diag(nrow(iris))-H)%*% y
```

```
#rank of I-H is 147 (150-3)
head(e)
Matrix::rankMatrix(diag(nrow(iris))-H)
t(e)%*%y_hat
```

Compute SST, SSR and SSE and R^2 and then show that SST = SSR + SSE.

```{r}
SSE = t(e) %*% e
SSE
y_bar = mean(y)
#SSE = e %*% t(e) the 147*147 of rank 1
SST = t(y-y_bar) %*% (y-y_bar)
Rsq = 1-SSE/SST
Rsq #
SSR = t(y_hat - y_bar) %*% (y_hat - y_bar)
SSR

expect_equal(SST,SSE+SSR)
#var(y)
#var(e) #3.11 vs 0.18 error is low compare to y.
```

Find the angle θ between y - $\bar{y}1$ and $\hat{y} - \bar{y}1$ and then verify that its cosine squared is the same as the R^2 from the previous problem.

```{r}
theta = acos (t(y-y_bar) %*% (y_hat - y_bar) / sqrt(SST * SSR))
```

theta * (180/pi) #u*v/UVcos(theta)

expect_equal(cos(theta)^2,Rsq)

```

Project the $y$ vector onto each column of the $X$ matrix and test if the sum of these projections is the same as yhat.

```{r}
proj1 = (X[,1]%*% t(X[,1]) / as.numeric((t(X[,1]) %*% X[,1]))) %*% y   #v1
proj2 = (X[,2]%*% t(X[,2]) / as.numeric((t(X[,2]) %*% X[,2])))%*% y
proj3 = (X[,3]%*% t(X[,3]) / as.numeric((t(X[,3]) %*% X[,3]))) %*% y
expect_equal(proj1+proj2+proj3, y_hat, tol =1e4) # not the orthnormal but not orthogonal
```

Construct the design matrix without an intercept, X, without using `model.matrix`.

```{r}
x_matrix = cbind(as.numeric(iris$Species == "setosa"), as.numeric(iris$Species == "versicolor"), as.numeric(iris$Species =="virgincia"))

x_matrix
```

Find the OLS estimates using this design matrix. It should be the sample averages of the petal lengths within species.

```{r}
y=iris$Petal.Length

H_1 = x_matrix %*%solve(t(x_matrix)%*%x_matrix)%*%t(x_matrix)

y_hat = H_1 * y

mean(iris$Petal.Length[iris$Species =="setosa"])

mean(iris$Petal.Length[iris$Species =="versicolor"])

mean(iris$Petal.Length[iris$Species =="virgincia"])

```

Verify the hat matrix constructed from this design matrix is the same as the hat matrix constructed from the design matrix with the intercept. (Fact: orthogonal projection matrices are unique).

```{r}

expect_equal(H_1, H)

```

Project the y vector onto each column of the X matrix and test if the sum of these projections is the same as yhat.

```{r}

proj1 = (x_matrix[,1] %*% t(x_matrix[,1])/as.numeric(t(x_matrix[,1])%*%x_matrix[,1]))%*% y

proj2 = (x_matrix[,2] %*% t(x_matrix[,2])/as.numeric(t(x_matrix[,2])%*%x_matrix[,2]))%*% y

proj3 = (x_matrix[,3] %*% t(x_matrix[,3])/as.numeric(t(x_matrix[,3])%*%x_matrix[,3]))%*% y

 expect_equal((proj1-proj2-proj3),y_hat)

```

Convert this design matrix into Q, an orthonormal matrix.
```

```{r}
Q = qr.Q(qr(x_matrix))
sum(Q[,1]^2) #normal length is 1
sum(Q[,2]^2)
sum(Q[,3]^2)
Q[,1]%*%Q[,2]#result 0 for orthogonal
Q[,1]%*%Q[,3]
Q[,2]%*%Q[,3]
```

Project the y vector onto each column of the Q matrix and test if the sum of these projections is the same as yhat.

```{r}
proj_y1 = (Q[,1] %*% t(Q[,1])/as.numeric(t(Q[,1])%*%Q[,1]))%*% y
proj_y2 = (Q[,2] %*% t(Q[,2])/as.numeric(t(Q[,2])%*%Q[,2]))%*% y
proj_y3 = (Q[,3] %*% t(Q[,3])/as.numeric(t(Q[,3])%*%Q[,3]))%*% y

 expect_equal((proj_y1 - proj_y2 - proj_y3),y_hat)
```

Find the $p=3$ linear OLS estimates if Q is used as the design matrix using the `lm` method. Is the OLS solution the same as the OLS solution for X?

```{r}
Q = lm(Q~1)
coef(Q)
Q$fitted.values
```

expect_equal(Q$fitted.values, x_matrix)

```

Use the predict function and ensure that the predicted values are the same for both linear models: the one created with $X$ as its design matrix and the one created with $Q$ as its design matrix.

```{r}
predict(object = Q,
        newdata =data.frame(x_matrix))
```

Clear the workspace and load the boston housing data and extract X and y. The dimensions are $n=506$ and $p=13$. Create a matrix that is $(p + 1) \times (p + 1)$ full of NA's. Label the columns the same columns as X. Do not label the rows. For the first row, find the OLS estimate of the y regressed on the first column only and put that in the first entry. For the second row, find the OLS estimates of the y regressed on the first and second columns of X only and put them in the first and second entries. For the third row, find the OLS estimates of the y regressed on the first, second and third columns of X only and put them in the first, second and third entries, etc. For the last row, fill it with the full OLS estimates.

```{r}
y = MASS::Boston[, 14]
X = as.matrix(cbind(1, MASS::Boston[, 1 : 13]))
p_plus_one = ncol(X)
n = nrow(X)
y_bar = (mean(y))
H = X %*% solve(t(X) %*% X) %*% t(X)
dim(H)
y_hat = H %*% y
SSR =sum((y_hat-y_bar)^2)
SST = sum((y-y_bar)^2)
```
```

```
Rsq = (SSR/SST)
Rsq

orthogonal_projection = function(a, v){
  H = v%*%t(v) / norm_vec(v)^2  #dimension is
  a_parallel = H%*%a
  a_perpendicular = a - a_parallel



  list(a_parallel = a_parallel, a_perpendicular = a_perpendicular)
}

na = matrix(NA, nrow = 14, ncol = p_plus_one)
na[ , 1] = X[ , 1]
for (j in 2:p_plus_one){
  na[,j]= X[,j]
  for (k in 1:(j-1)){
    na[,j]=na[,j] - orthogonal_projection(X[,j], na[,k])$a_parallel
  }
}

```

Why are the estimates changing from row to row as you add in more predictors?

#to-do Add more dimantion and projection that will increasing the % of explained data.

Create a vector of length $p+1$ and compute the R^2 values for each of the above models.

````
```{r}
#TO-DO
m = rep(NA,length=p_plus_one)
m
SSR =sum((y_hat-y_bar)^2)
SST = sum((y-y_bar)^2)


Rsq = (SSR/SST)
```
````

Is R^2 monotonically increasing? Why?

#TO-DO yes, because more orthogonal projection repesent to the regress, increase the % of expained part of the regression.