STA 521 HW 2

William Tirone

! The honor code

(a) Please state the names of people who you worked with for this homework. You can also provide your comments about the homework here.
Alonso Guerrero, Eli Gnesin, Medhavi Darshan, Natalie Smith, Sanskriti Purohit, Tommy Misikoff.
Feedback: Question 5 was obscenely difficult. I would have appreciated some helpful tips or something. It seems not practically useful to have to work something out like question 5 part 6. The gradient of the EM was also very, very difficult.
(b) Please type/write the following sentences yourself and sign at the end. We want to make it extra clear that nobody cheats even unintentionally.
I hereby state that all of my solutions were entirely in my words and were written by me. I have not looked at another student's solutions and I have fairly credited all external sources in this write up.

```
library(mvtnorm)
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.3.6      v purrr   0.3.4
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.1      v stringr 1.4.1
v readr   2.1.3      v forcats 0.5.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(mclust)
```

```
Package 'mclust' version 5.4.10
Type 'citation("mclust")' for citing this R package in publications.

Attaching package: 'mclust'

The following object is masked from 'package:purrr':

    map

The following object is masked from 'package:mvtnorm':

    dmvnorm
```

1

a)

TRUE. It is unbiased, since $E(\hat{\beta}) - \beta = 0$

b)

TRUE. This can be verified with the formula for MSE for the ridge estimator, and since it is biased, we see that increasing λ increases the bias and decreases the variance. Hence, this is the bias-variance trade off.

c)

TRUE. $\lambda x = Ax = A^2x = A\lambda x = \lambda Ax = \lambda^2 x$
Then, we have $\lambda(1-\lambda)x = 0$ so $\lambda \in \{0, 1\}$

d)

TRUE.
It is a projection matrix, and thus idempotent. $H = H^T$ so it is symmetric, and it is PSD since every $\lambda \geq 0$.

e)

FALSE. $tr(I - H) = tr(I) - tr(H) = n - p$ since the trace of an idempotent matrix equals it's rank, and H has rank = p.

f)

FALSE. We saw an example in class of outliers that had low leverage scores.

g)

FALSE. From L9 p. 17 it looks like MSE will increase as p predictors are added.

h)

TRUE. I consulted wikipedia and this is true!

2

a)

projection = $[x_1, 0, 0]^T$

b)

projection = $[x_1, x_2, 0]^T$ - just did this visually.

c)

$$\text{part A: } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{part B: } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

d)

```r
# proj matrices
part_A = matrix(c(1,0,0,0,0,0,0,0,0),3,3)
part_B = matrix(c(1,0,0,0,1,0,0,0,0),3,3,byrow=TRUE)

# making random unif vectors
set.seed(345)
v1 = matrix(runif(3,0,1))
v2 = matrix(runif(3,0,1))

proj1 = part_A %*% v1
proj2 = part_B %*% v2

proj1
```

```
          [,1]
[1,] 0.2162537
[2,] 0.0000000
[3,] 0.0000000
```

```r
proj2
```

```
          [,1]
[1,] 0.6557397
[2,] 0.4358664
[3,] 0.0000000
```

e)

$$\frac{a^T x}{a^T a} a$$

and the matrix is:

$$P = \frac{a a^T}{a^T a}$$

f)

```r
proj = function(x,a) {
  proj_matrix = (a %*% t(a)) / as.double((t(a) %*% a))
```

```
  x_onto_a = proj_matrix %*% x
  x_onto_a
}

proj(
  x=matrix(c(3,2,-1)),
  a=matrix(c(1,0,1))
)
```

```
     [,1]
[1,]    1
[2,]    0
[3,]    1
```

g)

$$P = A(A^T A)^{-1} A^T$$

Where the columns of A are a_1, a_2.

h)

As hinted by the problem, we can use Gram-Schmidt to find an orthonormal basis that spans the same subspace given by a_1, a_2. With A = QR, and Q being the orthonormal basis, we can construct a projection matrix:

$$P = Q(Q^T Q)^{-1} Q^T$$

i)

```
A = matrix(c(1,0,1,1,-1,0),3,2)
qr_decomp = qr(A)
Q = qr.Q(qr_decomp)

x = matrix(c(3,2,-1))

projection_matrix = Q %*% solve(t(Q) %*% Q) %*% t(Q)
projection_matrix %*% x
```

```
     [,1]
```

```
[1,] 1.000000e+00
[2,] 1.665335e-16
[3,] 1.000000e+00
```

```
t(Q) %*% Q
```

```
              [,1]          [,2]
[1,] 1.000000e+00 7.024919e-17
[2,] 7.024919e-17 1.000000e+00
```

j)

I believe this is the same as my answer in h) (and looking ahead to part k) we can construct a projection matrix onto the k-dimensional subspace spanned by $a_1, ..., a_k$ by computing the QR decomposition, taking the Q, and constructing:

$$P = Q(Q^TQ)^{-1}Q^T$$

k)

I am not clear on what this part is asking - I've read the Ed discussion post but this isn't explicitly asking anything other than stating a fact so not sure how to answer.

l)

The answer below looks the same as part i).

```
A %*% solve(t(A)%*% A) %*% t(A) %*% x
```

```
              [,1]
[1,]  1.000000e+00
[2,] -5.551115e-17
[3,]  1.000000e+00
```

m)

$$P = Q(Q^TQ)^{-1}Q^T = Q(I)Q^T = QQ^Tx$$

3

a)

Joint Distributions:

$$P(X=x,Z=1;\theta)=\frac{1}{\sqrt{2}\pi}exp\{-\frac{1}{2}(x-\mu_1)^2\}\cdot wP(X=x,Z=2;\theta)=\frac{1}{\sqrt{2}\pi}exp\{-\frac{1}{2}(x-\mu_2)^2\}\cdot(1-w)$$

Marginal likelihood:

$$p(X=x;\theta)=w\cdot N(u_1,1)+(1-w)N(\mu_2,1)$$

log-likelihood:

$$\sum_{i=1}^{N}log(w\cdot N(u_1,1)+(1-w)N(\mu_2,1))$$

b)

E-step of the above computations:

$$P(Z=1|X=x_i;\theta)=\frac{w^{(t)}N(\mu_1^{(t)},1)}{w^{(t)}N(\mu_1^{(t)},1)+(1-w^{(t)})N(\mu_2^{(t)},1)}$$

$$P(Z=2|X=x_i;\theta)=\frac{(1-w^{(t)})N(\mu_2^{(t)},1)}{w^{(t)}N(\mu_1^{(t)},1)+(1-w^{(t)})N(\mu_2^{(t)},1)}$$

Now, we can find the conditional expectation $E_{Z|x_i,\theta_n}[log(p_\theta(x_i,Z)$, so for each Z=j we will have

$$E_{Z|X,\theta^t}[log(p_\theta(x_i,Z)$$

$$E_{Z|X,\theta^t}[\sum_{i=1}^{n}log(p_\theta(x_i,Z)]$$

$$\sum_{i=1}^{n}\sum_{j=1}^{2}p(Z=j|X=x;\theta^t)log(p_\theta(x_i,Z)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{2}q_i^{t+1}[log(\pi_j)-\frac{1}{2}(x_i-\mu_j)^T(x_i-\mu_j)-log(2\pi)]$$

c)

(used some results from this wiki article)

$$w^{t+1} = \frac{\sum_n q_i^{t+1}}{\sum_n w + (1-w)} = \frac{\sum_n q_i^{t+1}}{n}$$

Consulted the matrix cookbook for the gradients, and taking the gradient with respect to μ_1 we get:

$$argmax(\mu_1) \sum q^{t+1}[\frac{1}{2}(x_i - \mu_j)^T(x_i - \mu_j)]\mu_1^{t+1} = \frac{\sum_n q_i^{t+1} x_i}{\sum_n q_i^{t+1}}$$

and since we just have two weights, μ_2 is very similar:

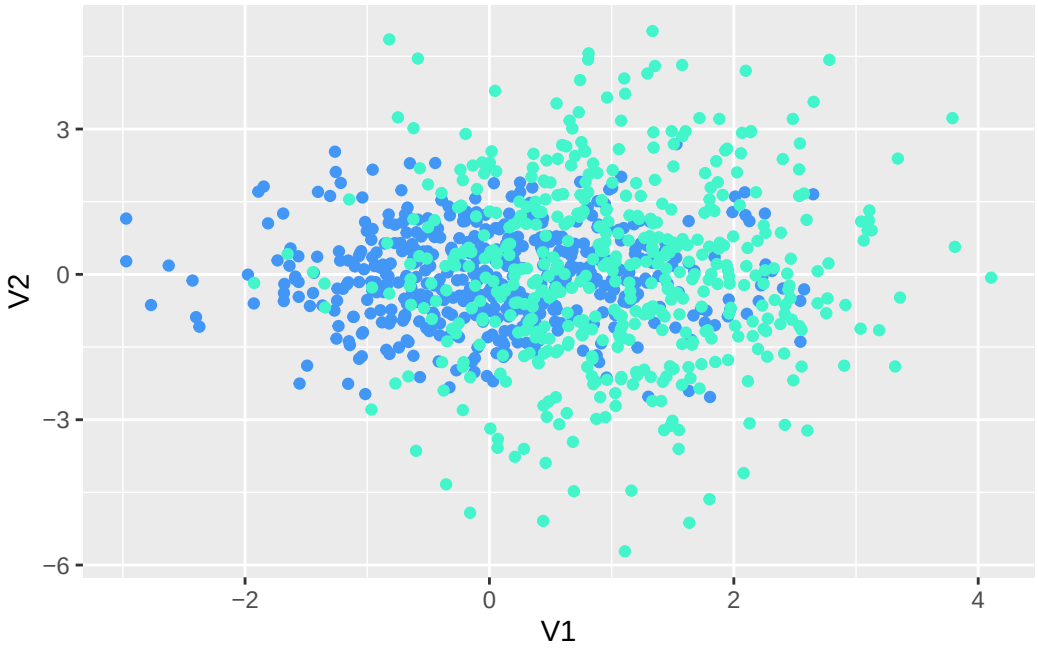
$$\mu_1^{t+1} = \frac{\sum_n (1 - q_i^{t+1}) x_i}{\sum_n (1 - q_i^{t+1})}$$

d)

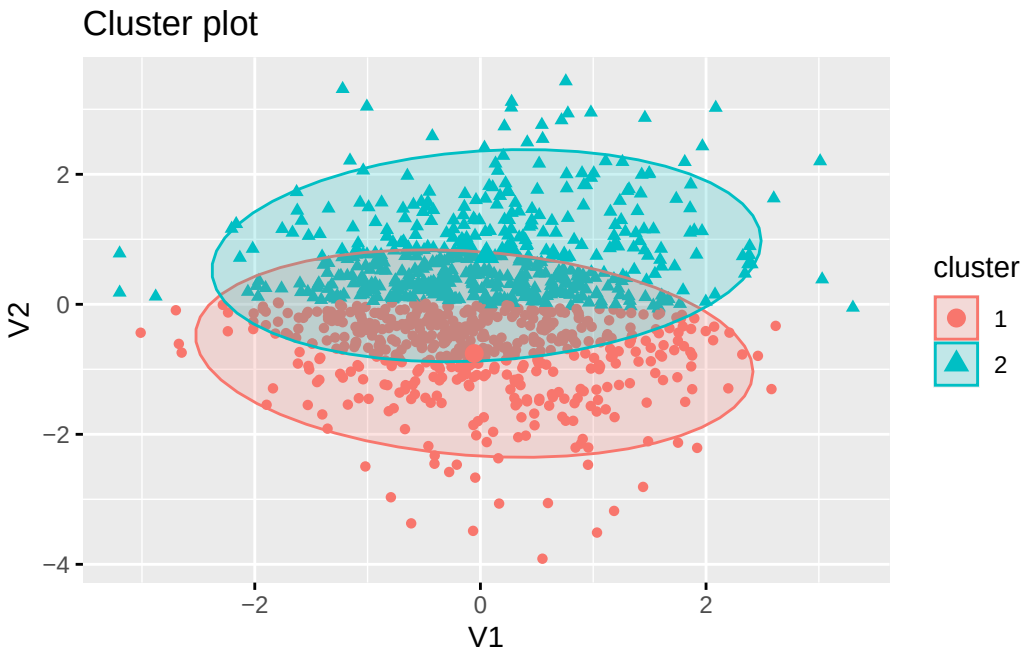
I believe the centers are different because K Means is trying to group the data according to distance to some centroid and the GMM is doing this based on the assumption that the underlying data comes from a normal model.

```
#code borrowed from this article: https://medium.com/mlearning-ai/drawing-and-plotting-obs

Z = rbernoulli(1000, 0.5) + 1
Z = rbernoulli(1000, 0.5) + 1
counts = as.data.frame(table(Z))

N1 = rmvnorm(n=counts[1,2], # this is the number sampled from Z where Z == 1
       matrix(c(0,0)),
       diag(1,2,2))

N2 = rmvnorm(n=counts[2,2], # number of Z==2 sampled from Z
       matrix(c(1,0)),
       matrix(c(1,0,0,4),2,2,byrow = TRUE))

N1 = as.data.frame(N1)
N2 = as.data.frame(N2)


ggplot() + geom_point(data=N1,aes(x=V1,y=V2),col='#4296f5') +
```

```
geom_point(data=N2,aes(x=V1,y=V2),col='#42f5cb')
```



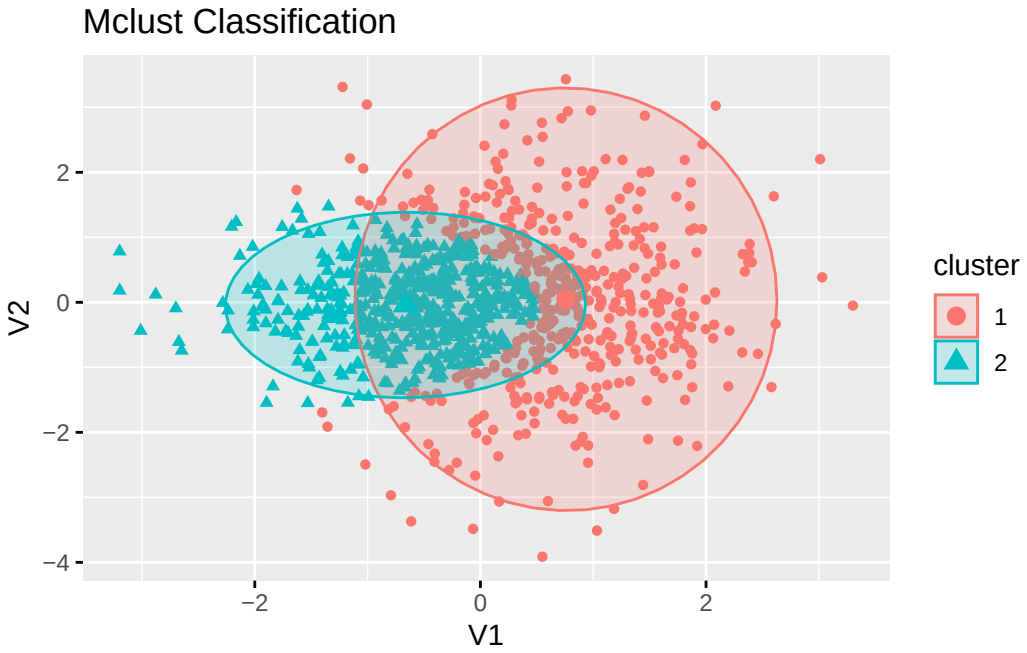

```
N1$ber = 1
N2$ber = 2
total_data = rbind(N1,N2)

set.seed(123)
km.result = kmeans(total_data[,1:2], centers=2)
total_data$cluster = km.result$cluster

fviz_cluster(km.result,total_data[,1:2],ellipse.type='norm',geom='point')
```




```
gmm.model = Mclust(total_data[,1:2],2)
gmm.model$parameters$mean
```

```
         [,1]        [,2]
V1 1.03399742 -0.03221819
V2 0.04001842 -0.02854677
```

```
fviz_cluster(gmm.model,
             what = "classification",
             main = "Mclust Classification",
             geom='point',
             ellipse.type='norm')
```



4

1.

We know the distribution of $\hat{\beta}$ must be normal because we assumed the errors are normally distributed. (Why?) Then it is sufficient to find the mean and variance of $\hat{\beta}$

$$\begin{aligned} E(\hat{\beta}) &= E[(X^TX)^{-1}X^Ty] \\ &= (X^TX)^{-1}X^TE[x\beta^* + \epsilon] \rightarrow \text{since X is constant} \\ &= (X^TX)^{-1}X^TXE(\beta^*) \\ &= \beta^* \end{aligned}$$

(Note for self: below mirrors form of $Var(X) = E(X - E(X))^2$

$$\begin{aligned} Var(B^*) &= E[(\hat{\beta} - E(\hat{\beta}))(\hat{\beta} - E(\hat{\beta}))^T] \\ &= E[(\hat{\beta} - (X^TX)^{-1}X^T\epsilon)(\hat{\beta} - (X^TX)^{-1}X^T\epsilon)^T] \\ &\text{next line follows since} E(\hat{\beta}) = \beta \text{and} X^TX \text{is symmetric} \\ &= E[(X^TX)^{-1}X^T\epsilon\epsilon^TX(X^TX)^{-1}] \\ &= (X^TX)^{-1}X^TE(\epsilon\epsilon^T)X(X^TX)^{-1} \\ &= \sigma^2(X^TX)^{-1} \end{aligned}$$

Noting that $E(\epsilon\epsilon^T)$ is the covariance matrix of ϵ with mean = 0.
Thus we have the mean and the variance of $\hat{\beta}$ and it is distributed $\sim N(\beta^*, \sigma^2(X^TX)^{-1})$
Now, since the data X is constant, $E(X\hat{\beta}) = E(X)E(\hat{\beta}) = X\beta^*$

2.

$$
\begin{aligned}
E||e||_2^2 &= ETr(ee^T) \\
&= ETr((I_n - X(X^TX)^{-1}X^T)\epsilon\epsilon^T(I_n - X(X^TX)^{-1}X^T)) \\
&= \sigma^2 Tr(I_n - X(X^TX)^{-1}X^T) \\
&\text{below using the cyclic property of trace} \\
&= \sigma^2 Tr(I_n) - Tr(X^TX(X^TX)^{-1}) \\
&= \sigma^2 Tr(I_n) - Tr(I_p) \\
&= \sigma^2(n-p)
\end{aligned}
$$

5

1.

$$
\begin{aligned}
\min_\theta ||X\theta - y||_2^2 + \lambda||\theta||_2^2 \\
gradient: 2X^T(X\theta - y) + 2\lambda\theta \\
\text{now setting this equal to 0 and solving for lambda} \\
2X^T(X\theta - y) + 2\lambda\theta = 0 \\
X^TX\theta - X^Ty + \lambda\theta = 0 \\
(X^TX + \lambda I_p)\theta = X^Ty \\
\hat{\theta}^{RR} = (X^TX + \lambda I_p)^{-1}X^Ty
\end{aligned}
$$

Gradient above found using the matrix cookbook. For each λ the solution above will yield a unique solution.

2.

Since we have assumed normally distributed errors, we know the ridge estimate will also be distributed normally. So we just need to find the mean and variance. First, we know how theta is distributed and will use these facts in the derivation:

$$\hat{\theta} \sim N(\theta^*, \sigma^2(X^TX)^{-1})$$

Then:

$$\begin{aligned} E(\hat{\theta}_\lambda) &= E[(X^TX + \lambda I_p)^{-1}X^T(X\theta + \epsilon)] \\ &= E[(X^TX + \lambda I_p)^{-1}X^TX\theta + (X^TX + \lambda I_p)^{-1}X^T\epsilon] \\ &= W_\lambda \theta^* \end{aligned}$$

Above I used that the expectation is linear, and $E(\epsilon) = 0$. Now noting that $\hat{\theta}_\lambda = W_\lambda \hat{\theta}$ which I have taken from the optional reading material on ridge regression.

$$\begin{aligned} Var(\hat{\theta}_\lambda) &= Var(W_\lambda \hat{\theta}) \\ &= W_\lambda Var(\hat{\theta}) W_\lambda^T \\ &= \sigma^2 W_\lambda (X^TX)^{-1} W_\lambda^T \\ &= \sigma^2 (X^TX + \lambda I_p)^{-1} X^TX (X^TX)^{-1} [(X^TX + \lambda I_p)^{-1} X^TX]^T \\ &= \sigma^2 (X^TX + \lambda I_p)^{-1} X^TX (X^TX + \lambda I_p)^{-1 \cdot T} \\ &= \sigma^2 W_\lambda (X^TX + \lambda I_p)^{-1} \end{aligned}$$

Above, on the second to last line, the transposed term is a symmetric matrix, and after grouping terms, we end up with the last line. This also borrows heavily from p.12 of the linked resource above but I tried to be a little more explicit. So we have

$$\hat{\theta}_\lambda \sim N(W_\lambda \theta^*, \sigma^2 W_\lambda (X^TX + \lambda I_p)^{-1})$$

3.

a)

$$\begin{aligned} &||E(\hat{\theta}_\lambda - \theta^*||_2^2 \\ &||(X^TX + \lambda I_d)^{-1} X^TX\theta^* - \theta^*||_2^2 \\ &||(UDU^T + \lambda I_d)^{-1} UDU^T\theta^* - \theta^*||_2^2 \\ &||U(D + \lambda I_d)^{-1} DU^T\theta^* - \theta^*||_2^2 \end{aligned}$$

b)

$$\begin{aligned} &E||\hat{\theta}_\lambda - E[\hat{\theta}_\lambda]||_2^2 \\ &E||(X^TX + \lambda I_p)^{-1} X^Ty - (X^TX + \lambda I_d)^{-1} X^TX\theta^*||_2^2 \\ &E||(U(D + \lambda I_p)^{-1} U^TX^Ty - U(D + \lambda I_d)^{-1} DU^T\theta^*||_2^2 \end{aligned}$$

Unsure how to complete these

4.

If $\lambda = 0$, the ridge regression problem just takes the form of OLS, which is unbiased and has variance $\sigma^2(X^TX)^{-1}$. As λ increases, we saw from Lecture 9 slides p. 36 that the bias-variance trade off starts to appear - bias will increase to ∞ and variance will go to 0.

5.

$$E||\hat{\theta}_\lambda - \theta^*||_2^2 = \sum_{i=1}^{p} E[(\hat{\theta}_{\lambda_i} - \theta_i^*)(\hat{\theta}_{\lambda_i} - \theta_i^*)]$$

Now, we can check the matrix $M(\lambda)$

$$\text{let} d_i = (\hat{\theta}_{\lambda_i} - \theta_i^*)M(\lambda) = E\begin{bmatrix} d_1d_1 & \dots & d_1d_p \\ \vdots & \ddots & \\ d_pd_1 & & d_pd_p \end{bmatrix}$$

So then by comparing the MSE to the sum of the diagonal elements of $M(\lambda)$ we can see they are the same.

6.

$MSE(\hat{\theta}_\lambda) < MSE(\theta^{OLS})0 < MSE(\theta^{OLS}) - MSE(\hat{\theta}_\lambda)$if we can prove this is positive definite we will have ou

We know the respective MSEs - I have taken the MSE for ridge from the additional reading material on ridge regression.

$$MSE(0) = \sigma^2(X^TX)^{-1}MSE(\hat{\theta}_\lambda) = \sigma^2W_\lambda(X^TX)^{-1}W_\lambda^T - (W_\lambda - I_p)\beta\beta^T(W_\lambda - I_p)^T$$

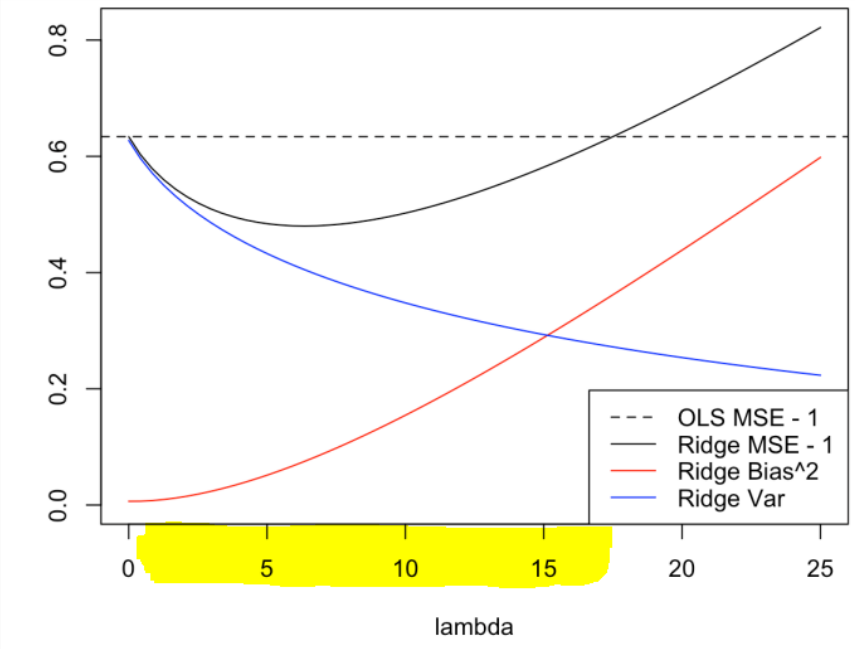
Then we have

$$M(0) - M(\hat{\theta}_\lambda) = \lambda(X^TX + \lambda I_p)^{-1}[2\sigma^2I_p + \lambda\sigma^2(X^TX)^{-1} - \lambda\beta\beta^T]([X^TX + \lambda I_p]^{-1})^T$$

Again the above result was borrowed from the "Lecture notes on ridge regression" by Wessel N. Van Wieringen. I'm not sure I can borrow a result like that for homework purposes, but I would've found the proof virtually impossible otherwise. Now, this is positive definite, as the reading shows, if $2\sigma^2I_p - \lambda\beta\beta^T > 0$ which occurs at $2\sigma^2(\beta^T\beta)^{-1}$. Thus, that is the range of lambdas we are after. To further emphasize my point, the highlighted regions below of lambdas was the point we emphasized in class and proved above.

Bias and variance of ridge regression in numerical experiment

Plot the Bias^2 and Variance for expected test error



31