# Lab #7 Report

Nick Huo

2022-11-16

## Question 1

### 1.

```
dat <- read.csv("Lab7.csv")
y <- matrix(dat$Y, ncol=1)
X <- as.matrix(dat[,-1])
```

```
calc_vif <- function(mtx) {
    res <- c()
    for (i in 1:ncol(mtx)) {
        mod <- lm(mtx[,i] ~ mtx[,-i])
        rsq <- summary(mod)$r.squared
        res <- c(res, 1/(1-rsq))
    }
    return(res)
}
calc_vif(X)
```

```
## [1] 1.105097 1.094970 1.069188 1.098317 1.165354
```

We see that the VIFs for the columns are all close to 1, indicating that the variances are not inflated and there is no multicollinearity.

### 2.

```
bias_ridge <- function(X, lambda, beta){
  bias <- solve(t(X)%*%X + lambda*diag(1, ncol(t(X)%*%X)) ) %*% (t(X)%*%X) %*% beta - beta
  return(bias)
}

var_ridge <- function(X, lambda, sigma_sq){
  variance <- sigma_sq * solve(t(X)%*%X + lambda*diag(1, ncol(t(X)%*%X)) ) %*%
      (t(X)%*%X) %*% solve(t(X)%*%X + lambda*diag(1, ncol(t(X)%*%X)) )
  return(variance)
}
```

```
lbd <- 0
beta_ridge <- c(0.398, 0.004, -0.831, 0.292, 0.004)
sigmasq_ridge <- 0.40

bias_ridge(X, lbd, beta_ridge)
```

```
##             [,1]
## X1  0.000000e+00
## X2 -2.602085e-18
## X3  0.000000e+00
## X4  5.551115e-17
## X5  2.081668e-17
```

```
diag(var_ridge(X, lbd, sigmasq_ridge))
```

```
##          X1          X2          X3          X4          X5
## 0.004465040 0.004424122 0.004319952 0.004437645 0.004708500
```

When $\lambda = 0$, this is the same as doing Ordinary Least Squares regression. So we see that the bias is essentially zero, since we know the OLS estimator is unbiased. And we also see that the variances has values around 0.4, similar to our $\sigma^2 = 0.40$.

## 3.

```
lbd <- 2

bias_ridge(X, lbd, beta_ridge)
```

```
##            [,1]
## X1 -0.009293213
## X2 -0.001671519
## X3  0.019129715
## X4 -0.007562468
## X5 -0.005331197
```

```
diag(var_ridge(X, lbd, sigmasq_ridge))
```

```
##          X1          X2          X3          X4          X5
## 0.004253386 0.004220180 0.004127172 0.004231487 0.004467850
```

## 4.

When $\lambda = 0$, the bias is essentially zero. But when $\lambda = 2$, we now see that the bias is not zero any more, despite being a small bias. And looking at the variances, when $\lambda = 2$, the variances are also showing slightly smaller values, indicating that a smaller variance compared to when $\lambda = 0$. I think this is because that we are seeing the bias-variance trade off, where we achieved smaller variance at the cost of the predictor being a bit biased.

### 5.

```
lbds <- seq(0,10, length.out=1000)
biases_ridge <- matrix(NA, 1000, 5)
vars_ridge <- matrix(NA, 1000, 5)
```

### 6.

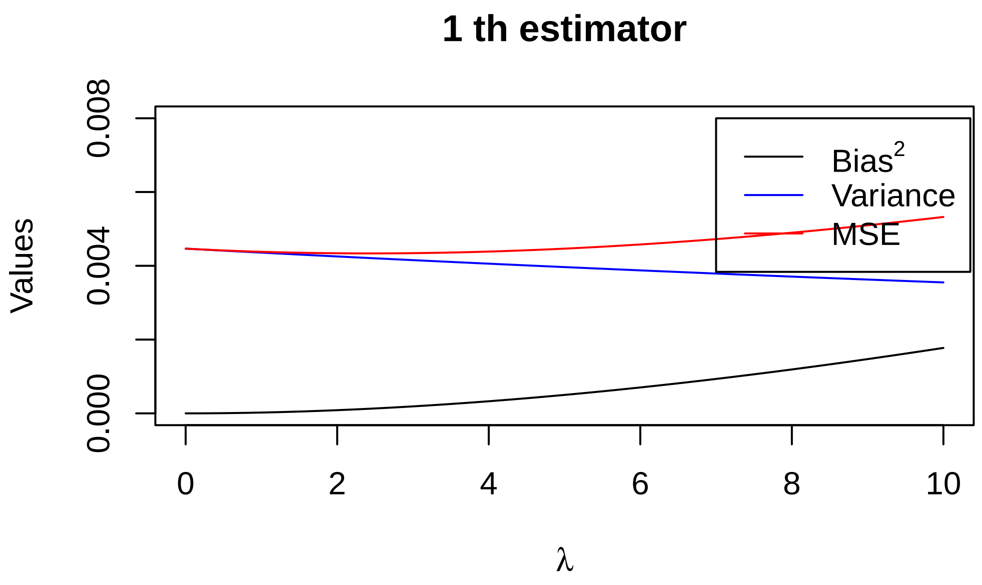
```
for (i in 1:1000) {
    this_lbd <- lbds[i]
    biases_ridge[i,] <- bias_ridge(X, this_lbd, beta_ridge)
    vars_ridge[i,] <- diag(var_ridge(X, this_lbd, sigmasq_ridge))
}
```

### 7.

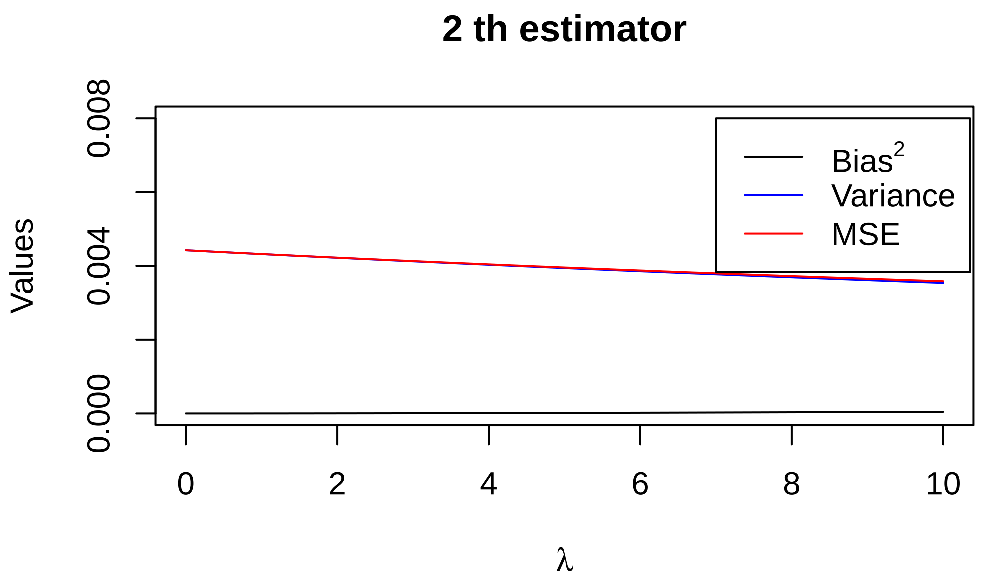
```
mse <- matrix(NA, 1000, 5)
for (i in 1:1000) {
    mse[i,] <- vars_ridge[i,] + (biases_ridge[i,])^2
}
```

### 8.

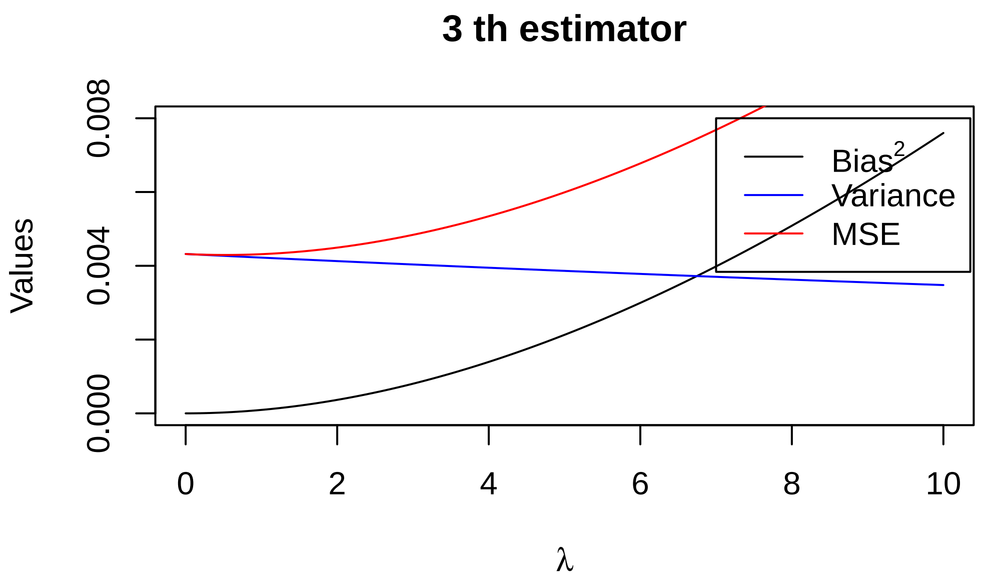
```
plot_est <- function(i) {
    plot(NA, NA, xlab = expression(lambda), ylab ="Values",
         main=paste(i,"th","estimator"),
         xlim=c(0,10), ylim =c(0,0.008))
    lines(lbds, biases_ridge[,i]^2)
    lines(lbds, vars_ridge[,i], col="blue")
    lines(lbds, mse[,i], col="red")
    legend(7, 0.008, legend=c(expression(Bias^2), "Variance", "MSE"),
           col=c("black","blue","red"), lty=1)
}

plot_est(1)
```



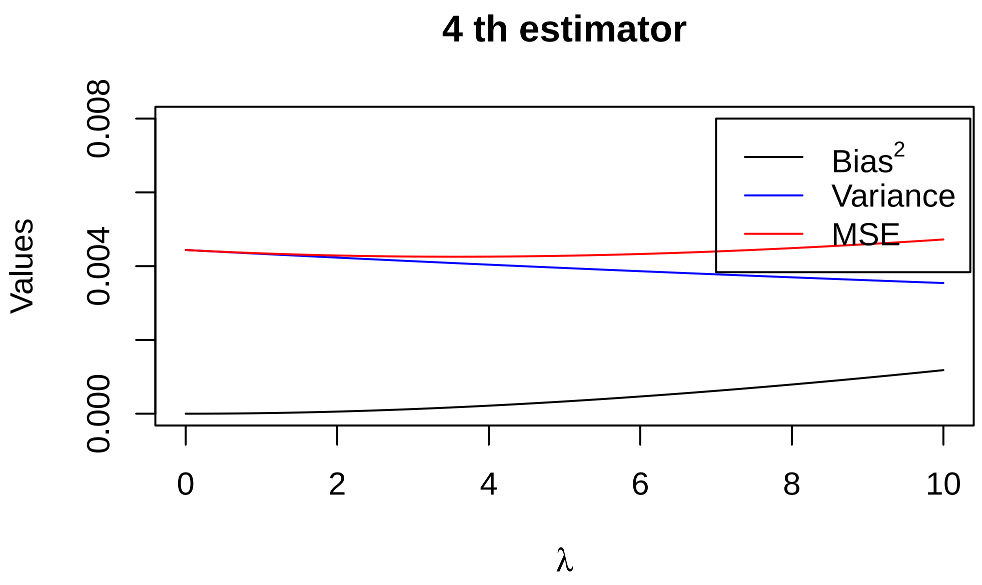
```
plot_est(2)
```



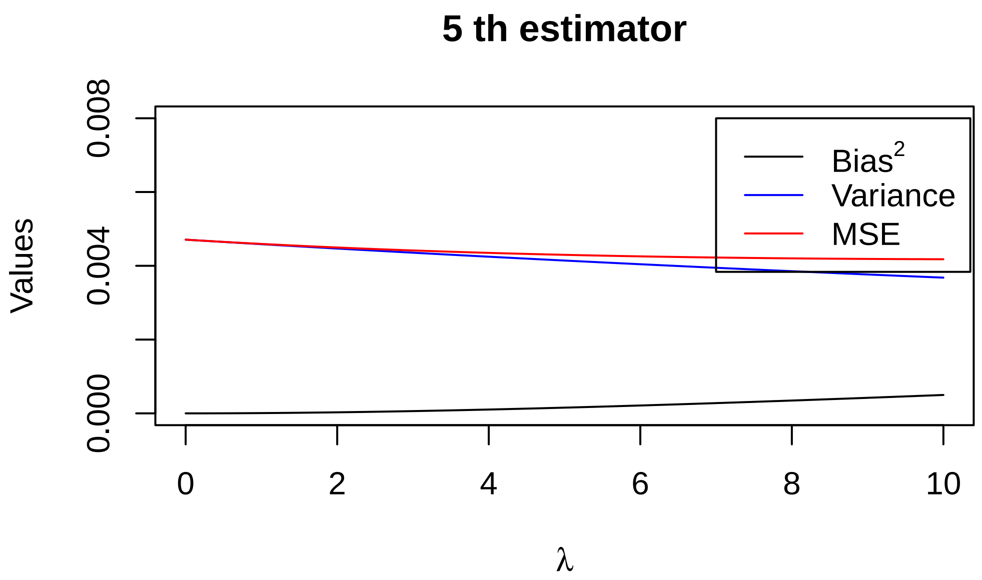
```
plot_est(3)
```



```
plot_est(4)
```



```
plot_est(5)
```



We see that for all the of estimators, as lambda increases, $Bias^2$ increases and $Variance$ decreases. Often times, for $\lambda$ values that are large, the bias will become too big that the decrease in variance is not big enough, and the MSE also increases drastically.

### 9.

It seems that $\lambda$ at around 2 works the best, since for most estimators, that is where the MSE is the lowest. So we are benefiting from the decrease in variance but not affected as much by the increase in bias.

### 10.

This procedure is highly subjective and is not precise. So we need some quantitative based method.

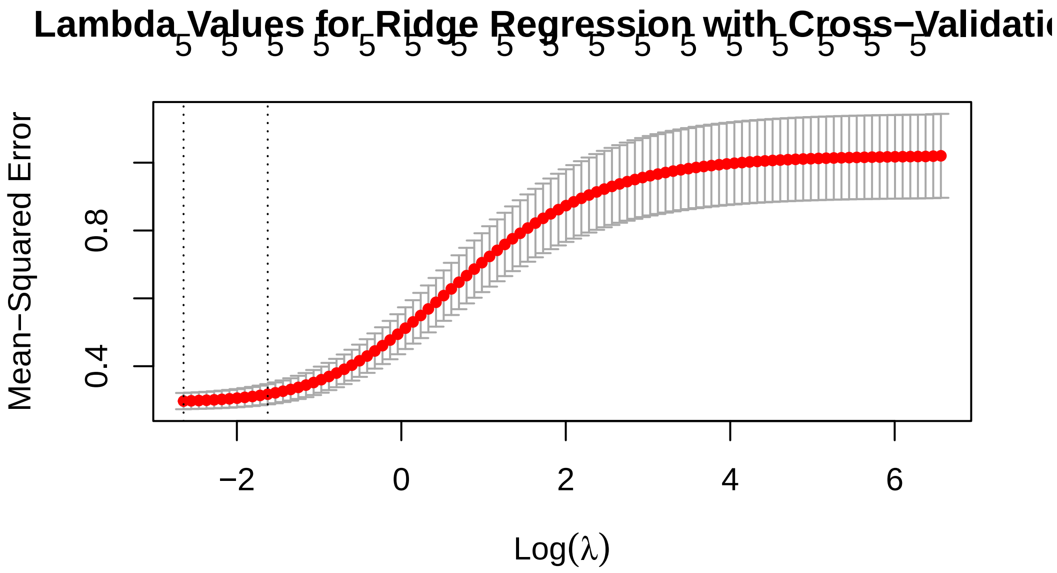
### 11.

```r
library(glmnet)
```

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

```
## Loaded glmnet 4.1-4
```

```r
ridge_cv <- cv.glmnet(X, y, alpha=0)
plot(ridge_cv, main="Lambda Values for Ridge Regression with Cross-Validation")
```



```
ridge_cv$lambda.min
```

```
## [1] 0.07077627
```

```
ridge_cv$lambda.1se
```

```
## [1] 0.1969392
```

### 12.

The $\lambda$ values we got from using cross-validation is a lot smaller than what I guessed to be the best $\lambda$ value in question 11.

### 13.

I think mainly we are learning to explore the bias-variance trade off. We see that when there is too much regulation ($\lambda$ is too high), we will get too much variance that produces large MSEs so it becomes even worse than using OLS. So this is why we need to pick $\lambda$ carefully, so that we are able to benefit from the bias-variance trade off and get lower MSEs with ridge regression. Just looking at the plots alone, it is unclear what $\lambda$ should we pick exactly. This is where cross-validation becomes useful and helps us to find the right $\lambda$ to use.