STA4241 Midterm Writeup

Problem 1

i. Estimator 1: $\hat{\lambda}_1^2 = \frac{1}{2n}\sum_{i=1}^{n} X_i^2$

$$E(\hat{\lambda}_1^2) = E(\frac{1}{2n}\sum_{i=1}^{n} X_i^2) = \frac{1}{2n}\sum_{i=1}^{n} E(X_i^2) = \frac{1}{2n}\sum_{i=1}^{n} 2\lambda^2 = \frac{1}{2n} \times 2n\lambda^2 = \lambda^2$$

To show that the above is an unbiased estimator, the expectation of the estimator must be the original statistic (lambda squared). Using the second moment of the exponential distribution and linearity of expectation, the equation ends up as lambda squared, proving the estimator to be unbiased.

Estimator 2: $\hat{\lambda}_2^2 = \frac{n}{n+1}\overline{X}^2$

$$E(\overline{X}) = E(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = \frac{1}{n} \times n\lambda = \lambda$$

$$E(\overline{X}^2) = Var(\overline{X}) + (E(\overline{X}))^2 = \frac{\lambda^2}{n} + \lambda^2 = \lambda^2(\frac{1}{n} + 1) = \lambda^2(\frac{n+1}{n})$$

$$E(\hat{\lambda}_2^2) = E(\frac{n}{n+1}\overline{X}^2) = \frac{n}{n+1} E(\overline{X}^2) = \frac{n}{n+1} (\lambda^2(\frac{n+1}{n})) = \lambda^2$$

To find the expectation of the second estimator, first find the expectation of x-bar: use the definition of the sample mean, linearity of expectation, and definition of distribution to find that the expectation is lambda. Next, find the expectation of x-bar squared by rearranging the definition of the variance of x-bar. Finally, find the expectation of the whole estimator by plugging the values in; as the expectation is equal to lambda squared, the estimator is unbiased.

ii. Variance of lambda squared

$$Var(\hat{\lambda}^2) = Var(\overline{X}^2) = E(\overline{X}^4) - (E(\overline{X}^2))^2$$

$$E(Y^4) = \theta^4 k(k+1)(k+2)(k+3) = (\theta^4)(k^4 + 6k^3 + 11k^2 + 6k)$$

$$E(\overline{X}^4) = (\frac{\lambda}{n})^4 n(n+1)(n+2)(n+3) = \frac{\lambda^4(n+1)(n+2)(n+3)}{n^3}$$

$$E(\overline{X}^4) - \left(E(\overline{X}^2)\right)^2 = \frac{\lambda^4(n+1)(n+2)(n+3)}{n^3} - \frac{\lambda^4(n+1)^2}{n^2}$$

$$= \lambda^4 \frac{(n+1)((n+2)(n+3) - n(n+1))}{n^3} = \lambda^4 \frac{(n+1)(4n+6)}{n^3}$$

To find the variance of lambda squared, use the 4th moment and the definition of the sample mean to get the variance.

iii. First, you must estimate the unknown parameter lambda by using the given definition in the problem. To estimate, sum the sample data that comes from the exponential distribution, divide this by the number of samples, and square that term to estimate lambda squared.

$$\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} X_i \text{ so } \hat{\lambda}^2 = (\frac{1}{n}\sum_{i=1}^{n} X_i)^2$$

Next, for each iteration of the bootstrap, generate samples by pulling from the exponential distribution with the estimated parameters. Sample from 1 through n for each 1 through B bootstrap runs.

$$X_i^{(b)} \sim \text{Exponential}(\hat{\lambda}) \text{ for } i = 1, \ldots, n$$

For each sample, calculate the bootstrap estimate of lambda squared.

$$\hat{\lambda}^2_{(b)} = (\frac{1}{n}\sum_{i=1}^{n} X_i^{(b)})^2 \text{ for } \hat{\lambda}^2_{(1)}, \hat{\lambda}^2_{(2)}, \ldots, \hat{\lambda}^2_{(B)}$$

You can store these estimates to create confidence intervals (using quantile function in R to find 95% confidence interval).

iv. To answer this problem, we can use the law of total variance.

$$\text{Var}(\hat{\lambda}^2_{(b)}) = \text{E}[\text{Var}(\hat{\lambda}^2_{(b)}|X)] + \text{Var}(\text{E}[\hat{\lambda}^2_{(b)}|X])$$

$$\text{E}[\hat{\lambda}^2_{(b)}|X] = \text{E}[\overline{X}^{(b)}|X]^2 + \text{Var}(\overline{X}^{(b)}|X) = \hat{\lambda}^2 + \frac{\hat{\lambda}^2}{n} = \hat{\lambda}^2(1 + 1/n)$$

$$E[\hat{\lambda}^2_{(b)}{}^2|X] = E[(\overline{X}^{(b)})^4] = (\frac{\hat{\lambda}^2}{n})^2 + 6(\frac{\hat{\lambda}^2}{n})\hat{\lambda}^2 + \hat{\lambda}^4 = \hat{\lambda}^4(\frac{1}{n^2} + \frac{6}{n} + 1)$$

$$\text{Var}(\hat{\lambda}^2_{(b)}|X) = \text{E}[\hat{\lambda}^2_{(b)}{}^2|X] - (\text{E}[\hat{\lambda}^2_{(b)}|X])^2 = \hat{\lambda}^4(\frac{1}{n^2} + \frac{6}{n} + 1) - (\hat{\lambda}^2(1 + 1/n))^2 = \hat{\lambda}^4(\frac{6}{n} - \frac{2}{n})$$

$$= \frac{4\hat{\lambda}^4}{n}$$

v. The expectation is being taken with respect to the random distribution of values x-bar from the exponential distribution.

$$E[Var(\hat{\lambda}^2_{(b)}|X)] = \frac{4}{n}E[\hat{\lambda}^4] = \frac{4\hat{\lambda}^4(n+1)(n+2)(n+3)}{n^3}$$

The expected value of the variance comes from the formula for the 4th moment.

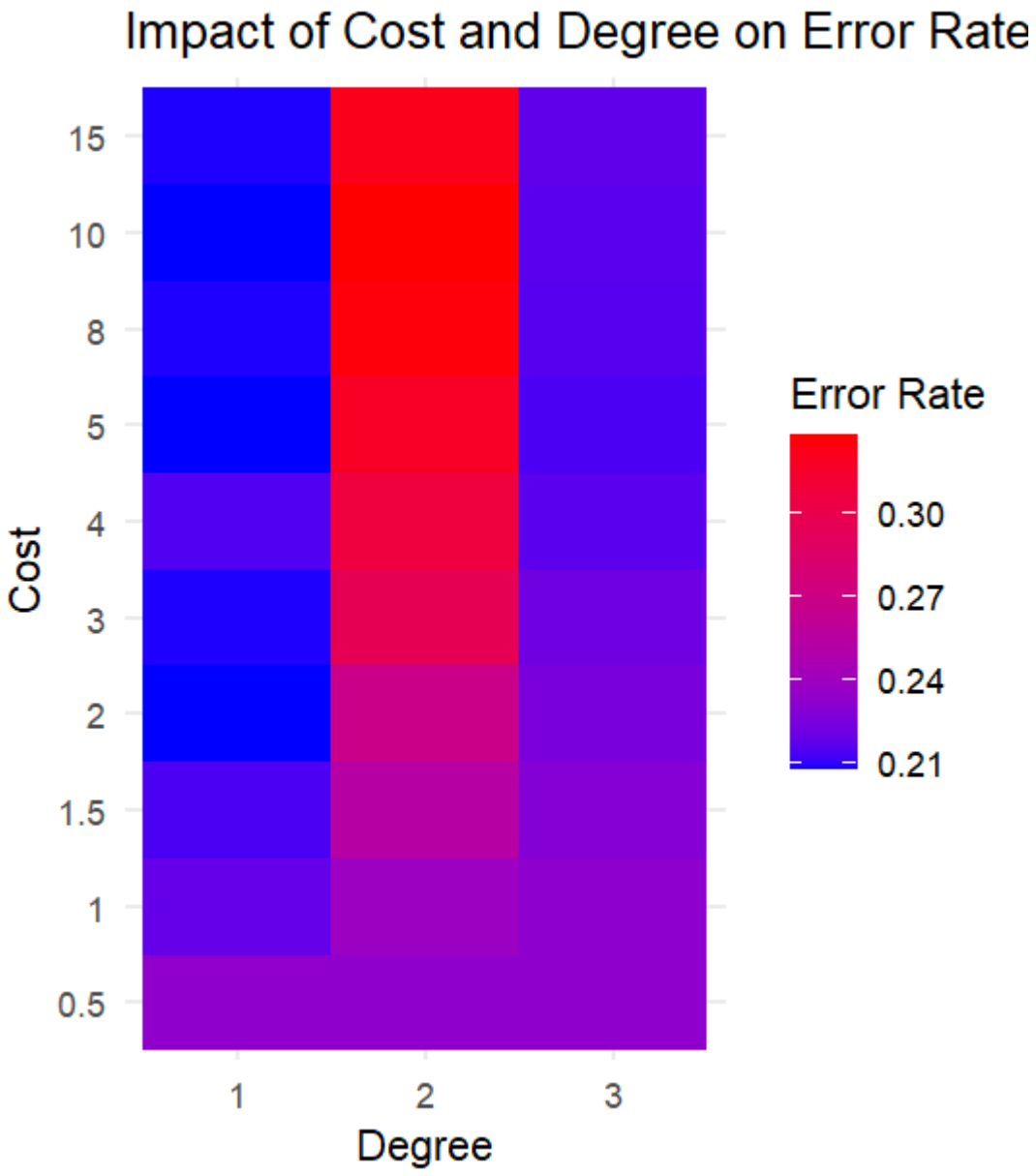
vi. The variance of the lambda squared estimator is 16.2, the expectation using the parametric bootstrap is 12.549, and the expectation using the nonparametric bootstrap is 13.282.

vii. The parametric and nonparametric bootstrap methods both underestimate the true variance, though the nonparametric bootstrap's variance is closer. Neither statistic was substantially far off, and it makes sense that the nonparametric bootstrap is closer as it does not assume any underlying distribution.

Problem 2

i. On the testing data, the error rate for logistic regression is 0.216. The error rate for linear discriminant analysis is 0.218. The error rate for a support vector machine with a radial kernel is 0.227.

ii. The optimal choice of cost is 10, the optimal choice of degree is 1, and the error rate under these optimal choices is 0.208. Below is the heatmap showing cross validation error as a function of cost and degree.

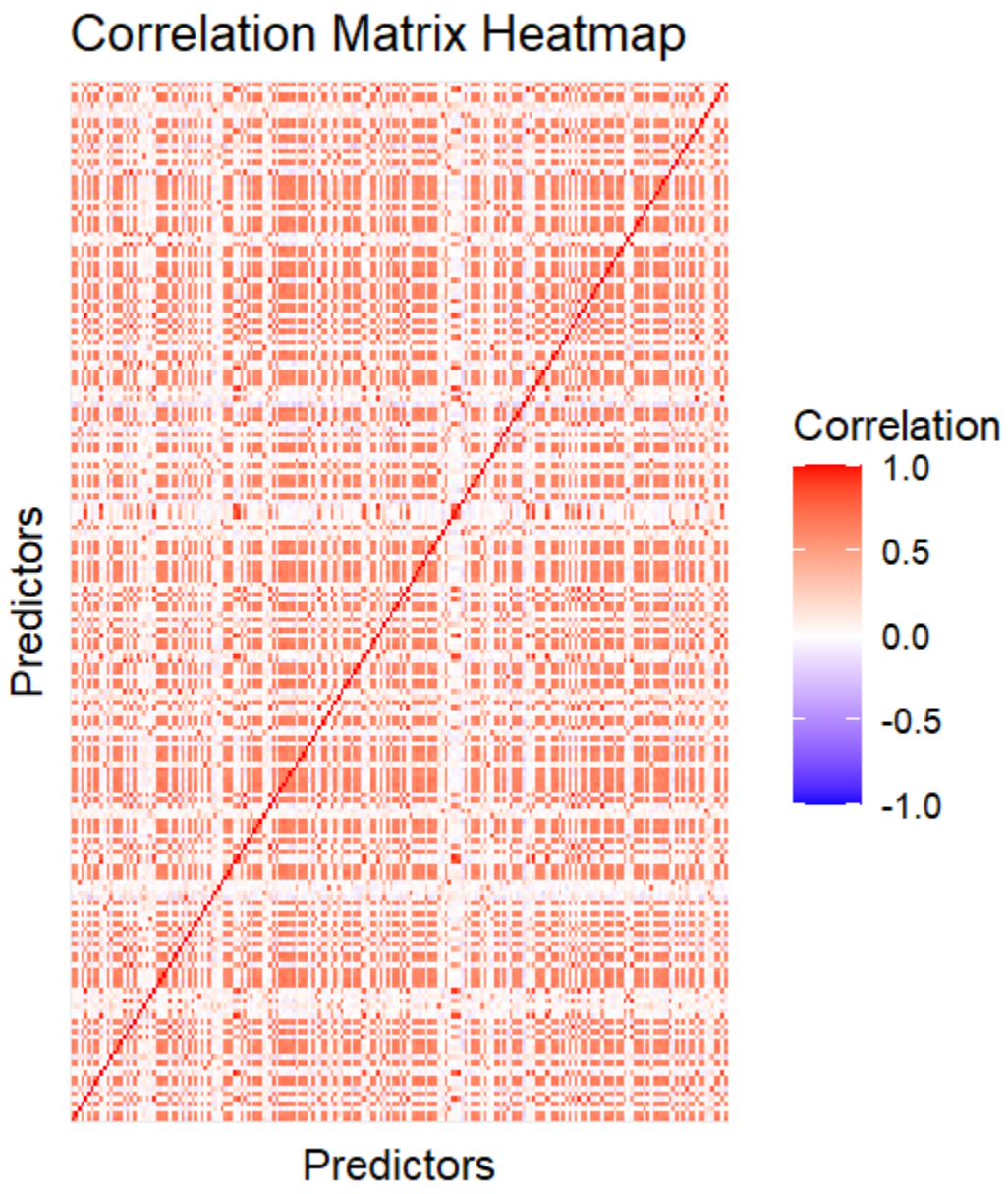


iii. To implement forward stepwise regression using qda, start by building the null model – using none of the predictors, effectively just guessing. Perform QDA on the model, noting the error rate. From there, try adding each predictor, eventually adding the predictor that lowers the error rate the most, also removing this predictor from the original list. Continue to add the best predictor that reduces the error rate, stopping this process when adding more predictors does not reduce the error rate anymore. Now, you have used forward stepwise regression to select the best predictors to minimize the error rate.

The covariates in the final model are [X.11, X.1, X.33, X.6, X.13, X.32, X.26, X.29, X.2, X.8, X.7, X.23, X.36, X.20] and the error rate is 0.191.

Problem 3

i. Looking at the correlation matrix heatmap, the significant amount of red/dark red (indicating higher positive correlation) indicates that a method like PCR to reduce the dimensions of the data to avoid collinearity.



ii. The mean square error rates: lasso is 1.117, ridge is 1.156, partial least squares is 1.451, and principal components is 1.222.

iii.a.

```
 [1] "(Intercept)" "X.1"         "X.2"         "X.4"         "X.5"         "X.12"        "X.14"
 [8] "X.19"        "X.23"        "X.25"        "X.30"        "X.36"        "X.38"        "X.41"
[15] "X.47"        "X.49"        "X.53"        "X.56"        "X.60"        "X.61"        "X.62"
[22] "X.63"        "X.65"        "X.66"        "X.67"        "X.77"        "X.83"        "X.87"
[29] "X.90"        "X.96"        "X.100"       "X.101"       "X.102"       "X.106"       "X.112"
[36] "X.115"       "X.116"       "X.117"       "X.120"       "X.122"       "X.124"       "X.129"
[43] "X.132"       "X.134"       "X.138"       "X.141"       "X.143"       "X.145"       "X.146"
[50] "X.150"       "X.154"       "X.157"       "X.163"       "X.167"       "X.173"       "X.178"
[57] "X.183"       "X.184"       "X.195"       "X.199"
```

```
 [1] "(Intercept)" "X.1"         "X.4"         "X.14"        "X.19"        "X.24"        "X.30"
 [8] "X.38"        "X.41"        "X.47"        "X.49"        "X.53"        "X.56"        "X.60"
[15] "X.61"        "X.62"        "X.65"        "X.66"        "X.77"        "X.78"        "X.87"
[22] "X.90"        "X.93"        "X.100"       "X.101"       "X.102"       "X.112"       "X.115"
[29] "X.116"       "X.117"       "X.120"       "X.122"       "X.132"       "X.134"       "X.138"
[36] "X.141"       "X.143"       "X.145"       "X.151"       "X.154"       "X.157"       "X.163"
[43] "X.173"       "X.178"       "X.183"       "X.199"
```

iii.b. Above, the parameters on top are the 60 chosen by the CV error minimization technique, and the parameters on the bottom are the 46 chosen by the one SE technique. Using the first technique with more parameters may be better when trying to maximize predictive accuracy, where the one standard error method uses few predictors, creating a simpler model to avoid potentially overfitting.

iv.a. The derivation uses a modified version of the OLS estimator:

$$\hat{\beta}_\lambda = (X^TX + \lambda I)^{-1}X^Ty = (X^TX + \lambda I)^{-1}X^TX(X^TX)^{-1}X^Ty = (X^TX + \lambda I)^{-1}X^TX\hat{\beta}$$

$$\begin{aligned}\mathrm{Var}(\hat{\beta}_\lambda) &= (X^TX + \lambda I)^{-1}X^TX\,\mathrm{Var}(\hat{\beta}_\lambda)\,[(X^TX + \lambda I)^{-1}X^TX]^T = \\ &= (X^TX + \lambda I)^{-1}X^TX\,\mathrm{Var}(\hat{\beta}_\lambda)X^TX\,(X^TX + \lambda I)^{-1} \\ &= (X^TX + \lambda I)^{-1}X^TX\sigma^2(X^TX)^{-1}X^TX\,(X^TX + \lambda I)^{-1} \\ &= \sigma^2(X^TX + \lambda I)^{-1}X^TX\,(X^TX + \lambda I)^{-1}\end{aligned}$$

iv.b. I would not expect this to lead to a confidence interval with the correct properties. The formula does not account for the biases introduced by ridge regression, so the standard error (function of variance) would be underestimated.

iv.c.

$$\mathrm{Cov}(\hat{\beta}_2, \hat{\beta}_{92}) = \sigma^2(X^TX + \lambda I)^{-1}X^TX\,(X^TX + \lambda I)^{-1} = (X^TX + 2I)^{-1}X^TX\,(X^TX + 2I)^{-1}$$

v.a. For the sparse scenario, after setting the number of samples and isolating the covariates, create a vector that is only 20% of the number of covariates and generate random data from a normal distribution, setting the other coefficients to zero (this is for beta). Make a covariate matrix (to act as X) using multivariate normal data, ultimately finding the outcome variable by multiplying the X matrix by the beta values, adding normally distributed random noise between 0 and 1. For the dense scenario, all the coefficients in the beta matrix are randomly generated from a normal distribution. From here, follow the same logic as sparse: create multivariate

normal data for the X matrix based on the input matrix and use the formula to find the outcome variable.

v.b.

```
> resultsSparse
  Method CoeffError  PredError
1  Lasso   14.37291   73.90764
2  Ridge   47.17794 1618.96412
3    PCR   56.03867 1230.66338
> resultsDense
  Method CoeffError PredError
1  Lasso   198.3229  153.5803
2  Ridge   190.0368 5319.4580
3    PCR   218.2890 3801.7541
```

v.c. Lasso regression performs well in both sparse and dense scenarios for both statistics, especially in the sparse scenario. Given a sparse data set, I would be inclined to use lasso, as it seems ridge and PCR have too much noise created by other predictors.