2) a) $E(Err_{in}) = E\left\{E\left[n^{-1}\sum \{\tilde{y}_i - \hat{f}(x_i)\}^2 \mid T\right]\right\}$

$= E\left[n^{-1}\sum\{\tilde{y}_i - \hat{f}(x_i)\}^2\right] = E\left[n^{-1}\sum A_i\right]$

Let's take a look at

$E(A_i) = E\left[(\tilde{y}_i - \hat{f}(x_i))^2\right] = E\left[(f(x_i) - \hat{f}(x_i) + \epsilon_i)^2\right]$

$= E\left[(f(x_i) - \hat{f}(x_i))^2\right] + \underbrace{E(\epsilon_i^2)}_{\sigma_i^2} - 2E[f(x_i) - \hat{f}(x_i)]E(\epsilon_i)$ ⁰

$= E\left[(f(x_i) - \hat{f}(x_i))^2\right] + \sigma_i^2$

$= E\left[\{(f(x_i) - E(\hat{f}(x_i))) - (\hat{f}(x_i) - E(\hat{f}(x_i)))\}^2\right] + \sigma_i^2$

$= E\left[\{E(\hat{f}(x_i)) - f(x_i)\}^2\right] + E\left[\{\hat{f}(x_i) - E(\hat{f}(x_i))\}^2\right]$

$- 2E\left[\{f(x_i) - E(\hat{f}(x_i))\}\{\hat{f}(x_i) - E(\hat{f}(x_i))\}\right] + \sigma_i^2$

$= \{E(\hat{f}(x_i)) - f(x_i)\}^2 + E\left[\{\hat{f}(x_i) - E(\hat{f}(x_i))\}^2\right]$

$- 2(f(x_i) - E(\hat{f}(x_i)))\left[\underbrace{E(\hat{f}(x_i)) - E(\hat{f}(x_i))}_{0}\right] + \sigma^2$

$= [\text{bias}(\hat{f}(x_i))]^2 + \text{Var}(\hat{f}(x_i)) + \sigma^2$

$\Rightarrow E(Err_{in}) = E\left[n^{-1}\sum A_i\right] = \frac{1}{n}\sum_{i=1}^{n} E(A_i)$

$= \frac{1}{n}\sum_{i=1}^{n}\left([\text{bias}(\hat{f}(x_i))]^2 + \text{Var}(\hat{f}(x_i)) + \sigma^2\right)$

$= \frac{1}{n}\sum_{i=1}^{n}\left([\text{bias}(\hat{f}(x_i))]^2 + \text{Var}(\hat{f}(x_i))\right) + \frac{1}{n}\cdot n\sigma^2$

$= n^{-1}\sum_{i=1}^{n}\left([\text{bias}(\hat{f}(x_i))]^2 + \text{Var}(\hat{f}(x_i))\right) + \sigma^2$

Based on part of the proof, when we minimize

$E(Err_{in})$, we're basically minimizing

$$E\left[n^{-1}\sum A_i\right] = n^{-1}\sum E(A_i) = n^{-1}\sum E\{[f(x_i)-\hat{f}(x_i)]^2\} + n^{-1}\cdot n\sigma^2$$

$$= n^{-1}\sum_{i=1}^{n} MSE_i + \underbrace{\sigma^2}_{\text{fixed}}$$

$\Rightarrow$ We're minimizing $n^{-1}\sum_{i=1}^{n} MSE_i$

b) $E(w) = E(Err_{in} - \overline{err})$

$$= E\left(\frac{1}{n}\sum E[(\tilde{y}_i - \hat{f}(x_i))^2] - \frac{1}{n}\sum(y_i - \hat{f}(x_i))^2\right)$$

$$= \frac{1}{n}\sum\left\{E\left[E\left[\underbrace{(\tilde{y}_i)^2}\right] + \hat{f}^2(x_i) - 2\tilde{y}_i\hat{f}(x_i)\right] - (y_i^2 + \hat{f}^2(x_i) - 2y_i\hat{f}(x_i))\right]\right\}$$

$$= \frac{1}{n}\sum\left\{E(y_i^2) + E(\hat{f}^2(x_i)) - 2E(y_i)E(\hat{f}(x_i)) - E(y_i^2) - E(\hat{f}^2(x_i)) + 2E(y_i\hat{f}(x_i))\right.$$

$$= \frac{1}{n}\sum\left\{2E(y_i\hat{f}(x_i)) - 2E(y_i)E(\hat{f}(x_i))\right\}$$

$$= \frac{2}{n}\sum_{i=1}^{n}\left\{E(y_i\hat{f}(x_i)) - E(y_i)E(\hat{f}(x_i))\right\}$$

$$= \frac{2}{n}\sum_{i=1}^{n} Cov(y_i, \hat{f}(x_i))$$

- For a linear model, $\hat{f}(x_i) = \hat{y}_i = h_{ii}\, y_i$ (i^{th} diagonal entry of hat matrix)

$$\Rightarrow E(w) = \frac{2}{n}\sum Cov(y_i, h_{ii}y_i)$$

$$= \frac{2}{n}\sum Cov(y_i, y_i)\, h_{ii}$$

$$= \frac{2}{n}\sigma^2\sum_{i=1}^{n} h_{ii} = \frac{2}{n}\sigma^2 \cdot tr(H)$$

From HW5, $tr(H) = p > 0$

$\Rightarrow E(w) > 0 \Rightarrow$ test error > training error

$\Rightarrow$ Training error usually underestimates test error.

3) a) We know $X^T X$ is always a positive semidefinite matrix for any matrix $X \in \mathbb{R}^{n \times p}$

Proof: For $z \in \mathbb{R}^n$, $z^T(X^T X) z = (Xz)^T (Xz) = \|Xz\|_2^2$

and $\|Xz\|_2^2 \geqslant 0 \Rightarrow X^T X$ is a positive semidefinite matrix

$\Rightarrow$ If we have c as an eigen-value of $X^T X \Rightarrow c \geqslant 0$

Also, we will have $c + \lambda$ as an eigenvalue for $X^T X + \lambda I$

With $\lambda > 0 \Rightarrow c + \lambda > 0 \Rightarrow X^T X + \lambda I$ is a positive definite matrix

And all positive definite matrices are invertible

$\Rightarrow \quad (X^T X + \lambda I_p)$ is invertible for all $\lambda > 0$

b) (i) $\hat{\beta}^{(ridge)}(\lambda) = (X^T X + \lambda I_p)^{-1} X^T Y$

$\Rightarrow \hat{Y}_\lambda^{(ridge)} = X \hat{\beta}^{(ridge)} = [X (X^T X + \lambda I_p)^{-1} X^T] Y = H_\lambda Y$

$\Rightarrow \quad H_\lambda = X(X^T X + \lambda I_p)^{-1} X^T$

We have $df_\lambda = \frac{1}{\sigma^2} \sum Cov(\hat{f}(x_i)_\lambda, y_i)$

$= \frac{1}{\sigma^2} Tr(Cov(\hat{Y}_\lambda^{(ridge)}, Y))$

$= \frac{1}{\sigma^2} Tr(Cov(H_\lambda Y, Y))$

$= \frac{1}{\sigma^2} Tr(H_\lambda Var(Y))$

$\Rightarrow df_\lambda = \frac{1}{\sigma^2} \cdot Var(Y) Tr(H_\lambda) = \frac{\sigma^2}{\sigma^2} Tr(H_\lambda) = Tr(H_\lambda)$

(ii) Using SVD of $X = UDV^T$

$\Rightarrow \quad X^T X = VDU^T UDV^T = VD^2V^T$ is the eigendecomposition of $X^T X$

$\Rightarrow H_\lambda = X(X^T X + \lambda I_p)^{-1} X^T$

$= UDV^T(VD^2V^T + \lambda I_p)^{-1} VDU^T$

$= UD\cancel{V^T V}(D^2 + \lambda I_p)^{-1} \cancel{V^T V} DU^T$

$= UD(D^2 + \lambda I_p)^{-1} DU^T$

$\Rightarrow$ Eigenvalues of H_λ is $\underbrace{D(D^2 + \lambda I_p)^{-1} D}_{p \times p \text{ diagonal matrix}}$

Let d_j be the j^{th} diagonal entry of D

$\Rightarrow \quad \frac{d_j^2}{d_j^2 + \lambda}$ is the j^{th} diagonal entry of the matrix $D(D^2 + \lambda I_p)^{-1} D$

$\Rightarrow \quad tr(H_\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$

So for $\lambda_1 < \lambda_2 \Rightarrow tr(H_{\lambda_1}) > tr(H_{\lambda_2})$

$\Rightarrow \quad df_{\lambda_1} > df_{\lambda_2}$

Also, as $\lambda \to 0$, $tr(H_\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2} = p$

Recall from ~~mid~~ midterm and HW5 : $df_{OLS} = p = rank(X) = tr(H_{OLS})$

$\Rightarrow$ for $0 < \lambda_1 < \lambda_2$, $\text{rank}(X) > df_{\lambda_1} > df_{\lambda_2}$

c) $\hat{\beta}^{(ridge)}_{(-i)} = (X^T_{(-i)} X_{(-i)} + \lambda I)^{-1} X^T_{(-i)} Y_{(-i)}$

We have $X^T_{(-i)} Y_{(-i)} = X^T Y - x_i y_i$

Also $X^T_{(-i)} X_{(-i)} + \lambda I = X^T X - x_i x_i^T + \lambda I$

Using result from HW8, we have

$$(X^T_{(-i)} X_{(-i)} + \lambda I)^{-1} = (X^T X + \lambda I)^{-1} + \frac{(X^TX+\lambda I)^{-1} x_i x_i^T (X^TX+\lambda I)^{-1}}{1 - \underbrace{x_i^T (X^TX + \lambda I) x_i}_{h_{ii}^{(\lambda)}}}$$

$$\Rightarrow \hat{\beta}^{ridge}_{(-i)} = (X^TX+\lambda I)^{-1}(X^TY - x_i y_i) + \frac{(X^TX+\lambda I)^{-1} x_i x_i^T (X^TX+\lambda I)^{-1}}{1 - h_{ii}^{(\lambda)}}(X^TY - x_i y_i)$$

$$= \hat{\beta}^{ridge} - (X^TX+\lambda I)^{-1} x_i y_i + \frac{(X^TX+\lambda I)^{-1} x_i}{1 - h_{ii}^{(\lambda)}} \Big(\overbrace{x_i^T (X^TX+\lambda I)^{-1} X^T Y}^{\hat{y}_i^{(\lambda)}} - \underbrace{x_i^T (X^TX+\lambda I)^{-1} x_i}_{h_{ii}^{(\lambda)}} y_i\Big)$$

$$= \hat{\beta}_{ridge} - (X^TX+\lambda I)^{-1} x_i y_i + \frac{(X^TX+\lambda I)^{-1} x_i}{1 - h_{ii}^{(\lambda)}} (\hat{y}_i^{(\lambda)} - h_{ii}^{(\lambda)} y_i)$$

$$= \hat{\beta}_{ridge} + \frac{(X^TX+\lambda I)^{-1} x_i}{1 - h_{ii}^{(\lambda)}} (\hat{y}_i^{(\lambda)} - h_{ii}^{(\lambda)} y_i - (1 - h_{ii}^{(\lambda)}) y_i)$$

$$= \hat{\beta}_{ridge} + \frac{(X^TX+\lambda I)^{-1} x_i}{1 - h_{ii}^{(\lambda)}} (\hat{y}_i^{(\lambda)} - y_i)$$

$$\Rightarrow y_i - x_i^T \hat{\beta}^{(ridge)}_{(-i)} = y_i - x_i^T \Big(\hat{\beta}_{ridge} + \frac{(X^TX+\lambda I)^{-1} x_i}{1 - h_{ii}^{(\lambda)}} (\hat{y}_i^{(\lambda)} - y_i)\Big)$$

$$= y_i - \underbrace{x_i^T \hat{\beta}_{ridge}} - \frac{x_i^T (X^TX+\lambda I)^{-1} x_i}{1 - h_{ii}^{(\lambda)}} (\hat{y}_i^{(\lambda)} - y_i)$$

$$= y_i - \hat{y}_i^{(\lambda)} - \frac{h_{ii}^{(\lambda)}}{1 - h_{ii}^{(\lambda)}} (\hat{y}_i^{(\lambda)} - y_i)$$

$$= (y_i - \hat{y}_i^{(\lambda)}) \left(\frac{1 - h_{ii}^{(\lambda)} + h_{ii}^{(\lambda)}}{1 - h_{ii}^{(\lambda)}} \right)$$

$$= \frac{y_i - \hat{y}_i^{(\lambda)}}{1 - h_{ii}^{(\lambda)}}$$

$$\Rightarrow PRESS(\lambda) = \sum_{i=1}^{n} (y_i - x_i^T \hat{\beta}^{(ridge)}_{(-i)})^2 = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i^{(\lambda)}}{1 - h_{ii}^{(\lambda)}} \right)^2$$

d) The estimator for λ is $\hat{\lambda}_{cv}$ which is defined as

$$\hat{\lambda}_{cv} = \underset{\lambda>0}{\operatorname{argmin}} \left(\sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i(\lambda)}{1 - h_{ii}^{(\lambda)}} \right)^2 \right)$$

HW9

Giang Vu

11/8/2020

Problem 1

Forward & backward selection

With both forward selection & alpha = 0.1 and backward selection & alpha = 0.2, only temp and fat are included in the model.

```
#read data
dat91 <- read.delim("/Users/giangvu/Desktop/STAT 2131 - Applied Stat Methods 1/HW/hw9/steam_text-2.txt")
fit91 <- lm(steam~fat+glycerine+wind+frezday+temp,data = dat91)

#Foward selection with alpha = 0.1
alpha.1 <- 0.1
forward91 <- olsrr::ols_step_forward_p(fit91, penter = alpha.1)
forward91$predictors #temp & fat included
```

```
## [1] "temp" "fat"
```

```
#Backward selection with alpha = 0.2
alpha.2 <- 0.2
backward91 <- olsrr::ols_step_backward_p(fit91, penter = alpha.2)
backward91$removed #temp & fat not removed
```

```
## [1] "wind"      "glycerine" "frezday"
```

Best subset using AIC and BIC

With both best subset regression using AIC and best subset using BIC, again, only temp and fat are included in the model.

```
#best subset with AIC
best.subset91 <- olsrr::ols_step_best_subset(fit91)
which.min(best.subset91$aic) #model with only fat and temp selected
```

```
## [1] 2
```

```
#best subset with BIC
AIC <- best.subset91$aic
our.BIC <- AIC - 2*(1:11) + log(nrow(dat91))*(1:11)
#How does this compare to their BIC#
best.subset91$sbc - our.BIC
```

```
##  [1]  2.437752  2.437752  2.437752  2.437752  2.437752 -3.656627 -3.656627
##  [8] -3.656627 -3.656627 -3.656627 -9.751007
```

```r
which.min(our.BIC)
```

```
## [1] 2
```

```r
which.min(best.subset91$sbc)
```

```
## [1] 2
```

Problem 4

(a)
Simple linear model

```r
#read data
dat94 <- read.delim("/Users/giangvu/Desktop/STAT 2131 - Applied Stat Methods 1/HW/hw9/Fat.txt")

#divide into test & train sets
test94 <- dat94[seq(1, nrow(dat94), 10), ]
train94 <- anti_join(dat94,test94)
```

```
## Joining, by = c("siri", "age", "weight", "height", "adipos", "free", "neck", "chest", "abdom", "hip"
```

```r
#simple linear
fit94 <- lm(siri~.,data = train94)
fit94
```

```
##
## Call:
## lm(formula = siri ~ ., data = train94)
##
## Coefficients:
## (Intercept)          age       weight       height       adipos         free
##   -6.612054     0.004228     0.387944     0.033490    -0.470841    -0.573609
##        neck        chest        abdom          hip        thigh         knee
##   -0.023312     0.122950     0.105760    -0.004548     0.176306     0.025355
##       ankle       biceps      forearm        wrist
##    0.110958     0.138203     0.204817     0.164980
```

```r
sum94a <- summary(fit94)
```

(b)
Ridge regression results are given below, I did this with a range of lambda from 0 to 0.1, incrementing by 0.0001. The value of lambda that minimizes the generalized cross validation value is 0.0339, which can also be seen in the plot.

```r
#ridge with training set
fit.ridge94 <- lm.ridge(siri~., data=train94, lambda = seq(0, 0.1, 0.0001))
summary(fit.ridge94)
```

```
##        Length Class  Mode
## coef   15015  -none- numeric
## scales    15  -none- numeric
## Inter      1  -none- numeric
## lambda  1001  -none- numeric
## ym         1  -none- numeric
## xm        15  -none- numeric
## GCV     1001  -none- numeric
## kHKB       1  -none- numeric
## kLW        1  -none- numeric
```

```r
res94b <- data.frame(fit.ridge94$GCV)
colnames(res94b) <- "GCV"
res94b$lambda <- as.numeric(rownames(res94b))
res94b[which.min(res94b$GCV),]$lambda #lambda pf 0.0339 is the one that minimizes the GCV value
```

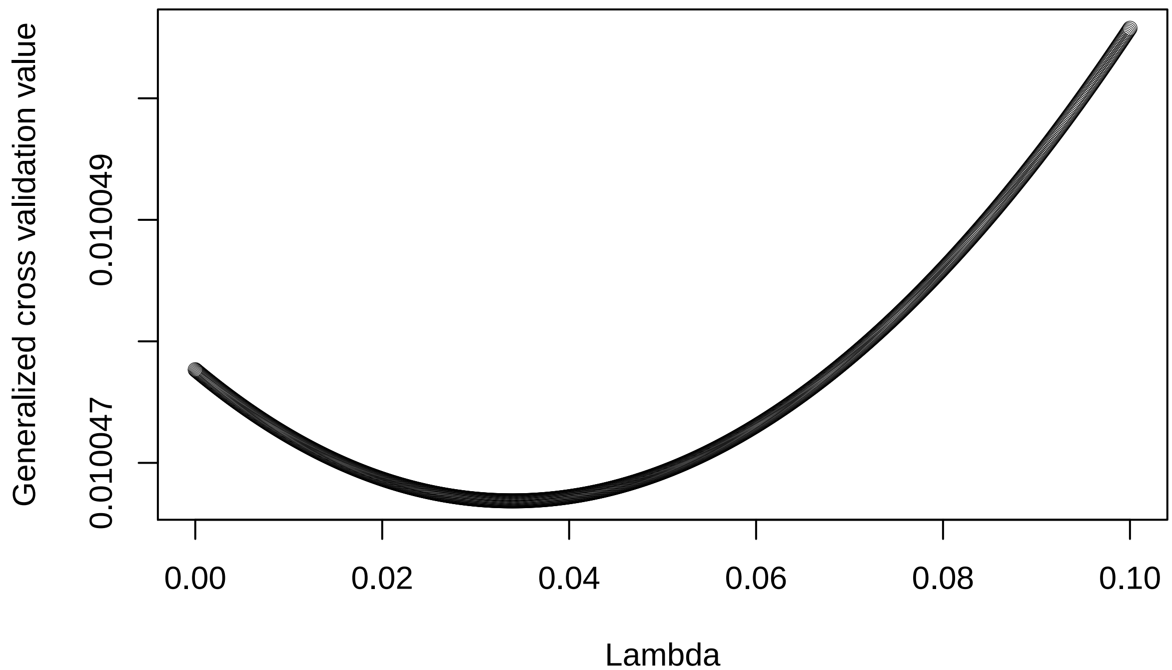
```
## [1] 0.0339
```

check this again, should be 0.046, use lm.ridge
The reason why I got 0.0339 bc i used obs #1,11,21,31,… as test set
to get 0.046, test set should be obs #10,20,30,..

```r
coef(fit.ridge94)[which.min(res94b$GCV),] #coefficient estimates for model with lambda = 0.0339
```

```
##                      age       weight       height       adipos         free
## -7.231046005  0.004091294  0.384519966  0.035212614 -0.466768517 -0.572024020
##         neck        chest        abdom          hip        thigh         knee
## -0.022049752  0.123745180  0.108432154 -0.002101477  0.176526808  0.027360916
##        ankle       biceps      forearm        wrist
##  0.113554272  0.139320345  0.205529015  0.162729489
```

```r
plot(x=res94b$lambda,y=res94b$GCV,xlab="Lambda",ylab="Generalized cross validation value",lwd=0.3)
```



(c)

The training error (MSE) of simple linear model in (a) and ridge regression model with lambda = 0.0339 in (b) are calculated below. We can see that the training error for model in (a) is smaller than that of model in (b). As already proven in question 2(b), for most linear models, training error tends to underestimate the prediction error, so it is a poor judge of how well the model will predictt future data.

```
#training error (MSE) for linear model
mse94a <- mean(sum94a$residuals^2)
mse94a #1.979365
```

```
## [1] 1.979365
```

```
#training error (MSE) for ridge model
coef94b <- coef(fit.ridge94)[which.min(res94b$GCV),] #coefficient estimates for model with lambda = 0.0
Xtrain <- model.matrix(siri~.,data=train94) #design matrix from training set
Yhat94b <- Xtrain%*%coef94b #fitted values for ridge model with lambda = 0.0339
mse94b <- mean((train94$siri-Yhat94b)^2)
mse94b #1.979579
```

```
## [1] 1.979579
```

```
#training error for model in a is smaller than training model for model in b
mse94a < mse94b
```

```
## [1] TRUE
```

(d)
For this part, I used squared loss as prediction error (test error), and found that the test error for model in (a) is higher than the test error for the model with ridge in (b), which is consistent with answer in part (c) where we discussed how the training error underestimates the test error. In this case, training error for model of (a) is smaller, but its test error is in fact larger than model in (b). So using training error to judge performance, model in (a) performs better, but using test error, we will have model in (b) performing better.

```
#test error for linear model
pred94a<-predict(fit94,newdata=test94[,-1],se=T)
testerr94a <- mean((test94$siri - pred94a$fit) ^ 2) #3.787006
testerr94a
```

```
## [1] 3.787006
```

```
#test error for ridge model
Xtest <- model.matrix(siri~.,data=test94) #design matrix from test set
Yhat94b_test <- Xtest%*%coef94b #fitted values for ridge model with lambda = 0.0339 with test dataset
testerr94b <- mean((test94$siri-Yhat94b_test)^2) #3.752729
testerr94b
```

```
## [1] 3.752729
```

```
#training error for model in a is smaller than training model for model in b
testerr94a < testerr94b
```

```
## [1] FALSE
```