Homework #2

ARINJAY JAIN
A20447307

1- F-statistic is equivalent by dropping a single coefficient from a model is equal to the corresponding Z-score.

Solution $\Rightarrow$ we know that F-statistic is

$$F = \frac{\frac{RSS_0 - RSS_1}{p_1 - p_0}}{\frac{RSS_1}{N - p_1 - 1}}$$

$F = z^2$ where $z = $ Z-score

$RSS_0 \sim \chi^2 (N - p_0 - 1)$
and $RSS_1 \sim \chi^2 (N - p_1 - 1)$

$$RSS_0 - RSS_1 \sim \chi^2 (N - p_0 - 1) - \chi^2 (N - p_1 - 1)$$

$$\sim \chi^2 (N - p_0 - 1 - N + p_1 + 1)$$

$$\Rightarrow RSS_0 - RSS_1 \sim \chi^2 (p_1 - p_0)$$

So when single coefficient is dropped $p_1 = p_0 + 1$
Thus,

$$RSS_0 - RSS_1 \sim \chi^2 (p_0 + 1 - p_0)$$

~~So, when single coefficent is dropped $p_1 = p_0 + 1$~~
~~Thus,~~

$$RSS_0 - RSS_1 \sim \chi^2 (p_0 + 1 - p_0)$$
$$\Rightarrow RSS_0 - RSS_1 \sim \chi^2, \text{ with dof} = 1$$

And,

$$\frac{RSS_1}{N - p_1 - 1} = \hat{\sigma}^2$$

So $\hat{\sigma}^2 \sim \chi^2_{N - p_1 - 1}$

And, we know that Chi-squared distribution is sum of squared Gaussian random variable

if $S = \sum_{i=1}^{v} Z_i^2$, where Z_i is Gaussian Random Variable $Z_i \sim N(0,1)$ then

$$S \sim \chi_v^2$$

Here $v=1$ as $p_1 - p_0 = 1$
Hence,

$$F = \frac{\chi_1^2}{\chi^2_{N-p_1-1}} = \frac{Z_1^2}{\hat{\sigma}^2}$$

$$= Z_j^2, \text{ where } z_j \text{ is } Z\text{-Score.}$$

Thus square of Z-Score is identical to F-distribution when one ~~Coef~~ coefficient is dropped.

Q2 Show that the ridge regression estimate is the mean (and mode) of the posterior distribution, under a Gaussian prior $\beta \sim N(0, \tau I)$ and Gaussian sampling model $y \sim N(X\beta, \sigma^2 I)$. Find the relationship between the regularization parameter λ in the ridge formula, and the variances τ and σ^2.

Solution, $\Rightarrow$

$$y \sim N(X\beta, \sigma^2 I) \text{ ~~so~~}$$

In term of probability,

$$Pr[y \mid \beta, \sigma] = \prod_{i=1}^{n} N(y_i \mid X\beta, \sigma^2 I)$$

Livelihood is

$$l(\beta, \sigma^2 | y_1 \ldots y_n) = Pr[Y | \beta, x]$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta^T x_i)^2\right)$$

where $x_i = \begin{pmatrix} x_{i,1} \\ \vdots \\ x_{i,p} \end{pmatrix}$ and $\beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}$

$$= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2\right)$$

to make it centered. we ar taking log !

$$L(\beta, \sigma^2 | y_1, \ldots y_n) = \log l(\beta, \sigma^2 | y_1 \ldots y_n)$$

$$= \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left(\frac{-1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2\right)\right]$$

$$= -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta^T x_i)^2$$

$$Pr[\beta] = \prod_{j=1}^{p} N(\beta_j | 0, \tau I)$$

$$= \prod_{j=1}^{p} \frac{1}{(2\pi)^{1/2}\tau} \exp\left(-\frac{1}{2\tau} \beta_j^2\right)$$

$$= \frac{1}{(2\pi)^{p/2} \tau^p} \exp\left(-\frac{1}{2\tau} \sum_{j=1}^{p} \beta_j^2\right)$$

$$\log Pr[\beta] = -\frac{p}{2} \log 2\pi - p \log \tau - \frac{1}{2\tau} \sum_{j=1}^{p} \beta_j^2$$

We are considering the feature are not dependent of each other.

Now, the posterior,

$$Pr[\beta|x,y] = \frac{Pr[y|\beta,x]\,Pr[\beta]}{Pr[y]}$$

$$\propto Pr[y|\beta,x]\,Pr[\beta]$$

We can ignore, $Pr[y]$ because is the depender on β or we can say ~~its~~ it will be constent w.r.t β.

taking log both side. ⇒

$$\log Pr[\beta|x,y] \propto \log Pr[y|\beta,x] + \log(Pr[\beta])$$

So, maximum aposterior is

$$MAP = \arg\max_{\beta}\left[\log Pr[y|\beta,x] + \log Pr[\beta]\right]$$

$$= \arg\max_{\beta}\left[-\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\beta^T x_i)^2 - \frac{P}{2}\log 2\pi\right.$$

$$\left.-P\log\tau - \frac{1}{2\tau}\sum_{j=1}^{P}\beta_j^2\right]$$

$$= \arg\max_{\beta}\left[-\underline{\text{Constent}} - \left(\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\beta^T x_i)^2 + \frac{1}{2\tau}\sum_{j=1}^{P}\beta_j^2\right)\right]$$

$$= \arg\max_{\beta} - \left[\left(\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\beta^T x_i)^2 + \frac{1}{2\tau}\sum_{j=1}^{P}\beta_j^2\right)\right]$$

Now we look for min.
and taking $\frac{1}{2\sigma^2}$ out side

$$= \arg\min_{\beta} \frac{1}{2\sigma^2}\left(\sum_{i=1}^{n}(y_i - \beta^T x_i)^2 + \frac{\sigma^2}{T}\sum_{j=1}^{p}\beta_j^2\right)$$

$$J(\beta) = \frac{1}{2\sigma^2}\left(\sum_{i=1}^{n}(y_i - \beta^T x_i)^2 + \frac{\sigma^2}{T}\|\beta\|^2\right)$$

$$= \frac{1}{2\sigma^2}\left((y - X\beta)^T(y - X\beta) + \frac{\sigma^2}{T}\beta^T\beta\right)$$

$$J(\beta) = \frac{1}{2\sigma^2}\left(y^Ty - 2\beta^T X^T y + \beta^T X^T X\beta + \frac{\sigma^2}{T}\beta^T\beta\right)$$

Now, taking partial derivation w.r.t β to get the β^{ridge},

$$\frac{\partial J(\beta)}{\partial(\beta)} = \frac{1}{2\sigma^2}\left(0 - 2X^Ty + 2X^TX\beta + 2\left(\frac{\sigma^2}{T}\right)\beta\right) = 0$$

$$X^Ty = \left(X^TX + \frac{\sigma^2}{T}I_p\right)\beta$$

$$\left\{\hat{\beta}^{ridge} = \left(X^TX + \frac{\sigma^2}{T}I_p\right)^{-1}X^Ty\right\}$$

It is clear the $Pr(\beta|y, x)$ is Gaussian and its mean or mode coincide,

$$Pr[\beta|x, y] \sim N(\mu_\beta, \Sigma)$$

$$\log Pr[\beta|x, y] = \frac{1}{2}(\beta - \mu_\beta)^T\Sigma^{-1}(\beta - \mu_\beta)$$

$$= 1/2\left(\beta^T\Sigma^{-1}\beta - \beta^T\Sigma^{-1}\mu_\beta - \mu_\beta^T\Sigma^{-1}\beta + \mu_\beta^T\Sigma^{-1}\mu_\beta\right)$$

$$= 1/2\left(\beta^T\Sigma^{-1}\beta - 2\beta^T\Sigma^{-1}\mu_\beta + \mu_\beta^T\Sigma^{-1}\mu_\beta\right)$$

by comparision of posterior deduced using prior and likelihood $\Rightarrow$

$$\frac{1}{\sigma^2}\left(\beta^T X^T X\beta + \frac{\sigma^2}{\tau}\beta^T\beta\right) = \beta^T\Sigma^{-1}\beta$$

$$\therefore \frac{1}{\sigma^2}\left(X^T X + \frac{\sigma^2}{\tau} I_p\right) = \Sigma^{-1}$$

and

$$\frac{1}{\sigma^2}\left(\beta^T X^T y\right) = \beta^T\Sigma^{-1}\mu_\beta$$

$$= \beta^T \frac{1}{\sigma^2}\left(X^T X + \frac{\sigma^2}{\tau} I_p\right)\mu_\beta$$

$$X^T y = \left(X^T X + \frac{\sigma^2}{\tau} I_p\right)\mu_\beta$$

$$\Rightarrow \mu_\beta = \left(X^T X + \frac{\sigma^2}{\tau} I_p\right)^{-1} X^T \cdot y$$

Ridge regression estimate is the mean (mode) of the posterior distribution under a Gaussian prior $\beta \sim N(0, \tau I)$, and Gaussian sampling model $y \sim N(X\beta, \sigma^2 I)$,

letting $\left\{\lambda = \frac{\sigma^2}{\tau}\right\}$ //

Homework#2

Arinjay Jain

```r
library(class)
library(formatR)

data <- read.table(file = "C:/Arinjay_Personal/Statistical Learning/Homework#2/Grocery.txt",
    header = FALSE, sep = "\t")


dataFrame <- data.frame(data)
names(dataFrame) <- c("Y", "X1", "X2", "X3")

fitModel <- lm(Y ~ X1 + X2 + factor(X3), data = dataFrame)
summary(fitModel)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + factor(X3), data = dataFrame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -264.05 -110.73  -22.52   79.29  295.75
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.150e+03  1.956e+02  21.220  < 2e-16 ***
## X1           7.871e-04  3.646e-04   2.159   0.0359 *
## X2          -1.317e+01  2.309e+01  -0.570   0.5712
## factor(X3)1  6.236e+02  6.264e+01   9.954 2.94e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 143.3 on 48 degrees of freedom
## Multiple R-squared:  0.6883, Adjusted R-squared:  0.6689
## F-statistic: 35.34 on 3 and 48 DF,  p-value: 3.316e-12
```

```r
coefficients <- fitModel$coefficients
std_Dev <- coef(summary(fitModel))[, "Std. Error"]
z_Score <- coef(summary(fitModel))[, "t value"]
p_Values <- coef(summary(fitModel))[, "Pr(>|t|)"]
fitModel_Table <- cbind(coefficients, std_Dev, z_Score, p_Values)
print(fitModel_Table)
```

```
##             coefficients      std_Dev    z_Score     p_Values
## (Intercept)  4.149887e+03 1.955654e+02 21.2199453 4.902653e-26
```

```
## X1           7.870804e-04 3.645540e-04  2.1590228 3.587650e-02
## X2          -1.316602e+01 2.309173e+01 -0.5701616 5.712274e-01
## factor(X3)1  6.235545e+02 6.264095e+01  9.9544230 2.940869e-13
```

```r
estimation_SigmaSquare <- (sum((fitModel$residuals)^2))/fitModel$df.residual

cat("estimation sigma_SigmaSquare:", estimation_SigmaSquare)
```

```
## estimation sigma_SigmaSquare: 20531.87
```

```r
y_Hat <- predict(fitModel)
```

```r
#Stepwise

library(olsrr)
```

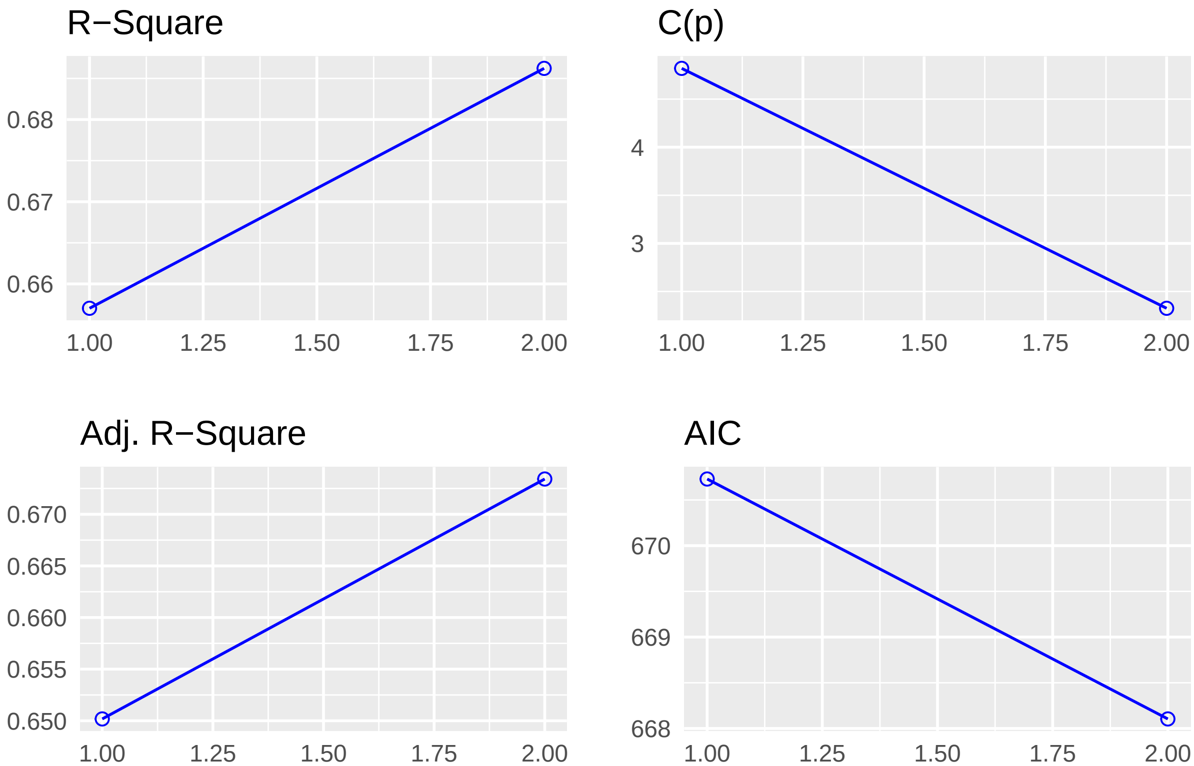
```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':
##
##     rivers
```

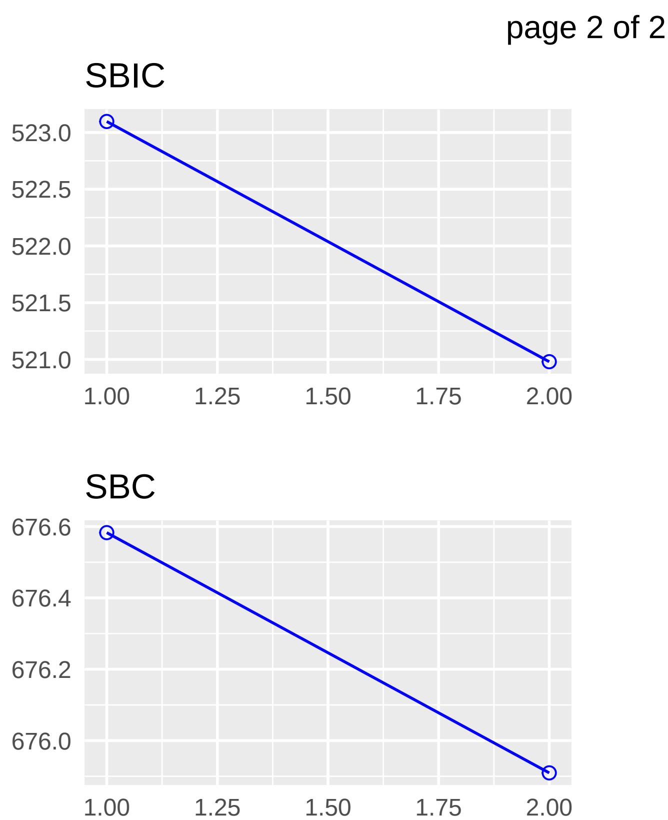
```r
forward_Step<-ols_step_forward_p(fitModel)
print(forward_Step)
```

```
##
## Selection Summary
## ---------------------------------------------------------------------------
## Variable Adj.
## Step Entered R-Square R-Square C(p) AIC RMSE
## ---------------------------------------------------------------------------
## 1 factor(X3) 0.6570 0.6502 4.8198 670.7292 147.2745
## 2 X1 0.6862 0.6734 2.3251 668.1045 142.2992
## ---------------------------------------------------------------------------
```

```r
plot(ols_step_forward_p(fitModel))
```

page 1 of 2


R−Square
0.68
0.67
0.66
1.00
1.25
1.50
1.75
2.00
C(p)
4
3
1.00
1.25
1.50
1.75
2.00
Adj. R−Square
0.670
0.665
0.660
0.655
0.650
1.00
1.25
1.50
1.75
2.00
AIC
670
669
668
1.00
1.25
1.50
1.75
2.00




```
back_Step<-ols_step_backward_p(fitModel)
print(back_Step)
```

```
##
##
## Elimination Summary
## ---------------------------------------------------------------------------
## Variable Adj.
## Step Removed R-Square R-Square C(p) AIC RMSE
## ---------------------------------------------------------------------------
## 1 X2 0.6862 0.6734 2.3251 668.1045 142.2992
## ---------------------------------------------------------------------------
```

```
print("From Forward and Backward both approaches giving same results. In our final model, we will keep
```

```
## [1] "From Forward and Backward both approaches giving same results. In our
final model, we will keep X1, X3 and remove X2"
```

```
finalModel<- lm(Y~X1+factor(X3), data=dataFrame)
summary(finalModel)
```

```
##
## Call:
## lm(formula = Y ~ X1 + factor(X3), data = dataFrame)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -286.249  -99.650   -9.251   70.746  292.311
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.058e+03  1.109e+02  36.592  < 2e-16 ***
## X1          7.704e-04  3.609e-04   2.135   0.0378 *
## factor(X3)1 6.196e+02  6.183e+01  10.021 1.88e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.3 on 49 degrees of freedom
## Multiple R-squared:  0.6862, Adjusted R-squared:  0.6734
## F-statistic: 53.58 on 2 and 49 DF,  p-value: 4.647e-13
```

```r
estimation_SigmaSquare_finalModel<- (sum((finalModel$residuals)^2))/finalModel$df.residual

cat("estimation sigma_SigmaSquare_finalModel:", estimation_SigmaSquare_finalModel)
```

```
## estimation sigma_SigmaSquare_finalModel: 20249.07
```

```r
## Bestsubset using Cp Criteria
library(leaps)

models <- regsubsets(Y~., data = dataFrame, nvmax = 3)

modelSummary <- summary(models)

CP = which.min(modelSummary$cp)

#best model will have below predictors:
modelSummary$which[CP,]
```

```
## (Intercept)          X1          X2          X3
##        TRUE        TRUE       FALSE        TRUE
```

```r
print("Checking the p-values in both small model and full model for the F-test to see the significance l
```

```
## [1] "Checking the p-values in both small model and full model for the F-test
to see the significance level:"
```

```r
#From part b: FinalModel #From part a: Fit model
com <- anova(finalModel,fitModel,test='F')

cat("F test value", com$F[2])
```

```
## F test value 0.3250843
```

```r
cat("P-value value", com$`Pr(>F)`[2])
```

```
## P-value value 0.5712274
```

```r
## Using F test formula

rSS_0 <- sum((finalModel$residuals)^2)
rSS_1 <- sum((fitModel$residuals)^2)

f_test = (rSS_0-rSS_1)*(fitModel$df.residual)/rSS_1
f_test
```

```
## [1] 0.3250843
```

```r
# F critical value
f_critical <- qf(p = 0.95, df1 = 1, df2 = 48)
f_critical
```

```
## [1] 4.042652
```

```r
if (f_test < f_critical){
  print("The null hypothesis is accepted")
}
```

```
## [1] "The null hypothesis is accepted"
```

```r
print("Here we can see in the small model (final model) both (x1 and x3) predictors have very significa
```

```
## [1] "Here we can see in the small model (final model) both (x1 and x3)
predictors have very significant (less then alpha{0.05}) p-value but in the
full model we have X2 with non-significan p-value. Hence, we will go with small
model(final model) as it keeps the model simpler with features being
statistically more significant "
```