STATS 415 HW 6., Li Hsuan Lin

UM-ID: 49109112

Section: 002

Problem (1)

(a):

- First, we show $R_a^2 \leq 1$, or equivalently, $1-R_a^2 \geq 0$

From Lecture slide, we know R_a^2 is defined as

$$R_a^2 = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$, n is sample size, p is the number of predictors

Then, $1-R_a^2 = 1-\left(1-\frac{RSS/(n-p-1)}{TSS/(n-1)}\right)$

$$= \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

$$= \frac{RSS}{TSS} \cdot \frac{(n-1)}{(n-p-1)}$$

$$= (1-R^2) \cdot \frac{(n-1)}{(n-p-1)}$$

since we know $0 \leq R^2 \leq 1$, $1-R^2 \geq 0$ and the fact that $p \geq 0$ makes the statement that $\frac{n-1}{(n-p-1)} \geq 1$ is true.

it must be the case that $1-R_a^2 \geq 0$, or $R_a^2 \leq 1$,

---

- For lower bound, I will prove it is false by providing a counter example.

Consider the case when there is no predictor when conducting linear regression. We have the model $Y = B_0 + \varepsilon$.

Using the formula, we know $\hat{B_0} = \bar{y} - \hat{B_1}\bar{x}$. Since $\hat{B_1} = 0$, $\hat{B_0} = \bar{y}$.

we have $\hat{b_0} = \bar{y} = \hat{y_i}$, Thus,

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \sum_{i=1}^{n}(y_i - \bar{y})^2 = TSS$$

$$R_a^2 = 1 - \frac{RSS}{TSS}\left(\frac{n-1}{n-p-1}\right)$$

$$= 1 - \left(\frac{n-1}{n-p-1}\right) = \frac{-n}{n-p-1} < 0.$$

Thus, the lower bound of adjusted R^2 can be less than 0

1b) We have $p = 500$, $n = 501$ and $R_a^2 = \frac{RSS/(n-d-1)}{TSS/(n-1)} = 1-(1-R^2)\frac{(n-1)}{(n-d-1)}$

Let $R_a = 0$. we have $1-(1-0.5)\frac{500}{(501-d-1)} = 0$

$$0 = 1 - 0.5 \cdot \frac{500}{500-d}$$

$$0.5 \cdot 500 = 500 - d$$

$$250 = 500 - d$$

$$d = 250$$

---

2) NTS $AIC(\text{Linear Regression}) = n\{\log(2\pi) + \log(\sigma^2)\} + SSE/\sigma^2 + 2d$

Proof: $Y_i \sim N(u_i, \sigma^2)$ where $u_i = X_i^T\beta$ and for $i = 1 \dots n$, Y_i are independent

$L(\beta, \sigma^2 | Y, X) = \prod_{i=1}^{n} f(X_i; \beta, \sigma^2)$ (since independence)

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(Y_i - X_i^T\beta)^2}{2\sigma^2}}$$

$$\ell(\beta, \sigma^2 | Y, X) = \log(L(\beta, \sigma^2 | Y, X)) = \sum_{i=1}^{n}\left(-\tfrac{1}{2}\log(2\pi) - \tfrac{1}{2}\log(\sigma^2)\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - X_i^T\beta)^2$$

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

we know $\hat{\beta}_{MLE}$ satisfies $\frac{\partial}{\partial\beta}\ell(\beta, \sigma^2 | Y, X) = \frac{-1}{2\sigma^2}\left(\frac{\partial}{\partial\beta}\right)\left(\sum_{i=1}^{n}(Y_i - X_i^T\beta)^2\right) = 0$

which means $\hat{\beta}_{MLE}$ satisfies $\frac{\partial}{\partial\beta}\sum_{i=1}^{n}(Y_i - X_i^T\beta)^2 = 0$

$$\Rightarrow \hat{\beta}_{MLE} = \hat{\beta}_{OLS}$$

Thus, $\ell(\beta, \sigma^2 | Y, X) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - X_i^T\hat{\beta}_{OLS})^2$

$$= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2} SSE(M)$$

Using the definition of AIC in Lecture Slide

$$AIC(M) = -2\log L(M) + 2d$$

$$= \frac{n}{2}\log(2\pi) + n\log(\sigma^2) + \frac{1}{\sigma^2} SSE(M) + 2d$$

```r
library(ISLR)
library(tidyverse)
```

```
## — Attaching packages ——————————————————————— tidyverse 1.2.1 —
```

```
## ✔ ggplot2 3.2.1     ✔ purrr   0.3.3
## ✔ tibble  2.1.3     ✔ dplyr   0.8.3
## ✔ tidyr   1.0.0     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0
```

```
## — Conflicts ———————————————————————— tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

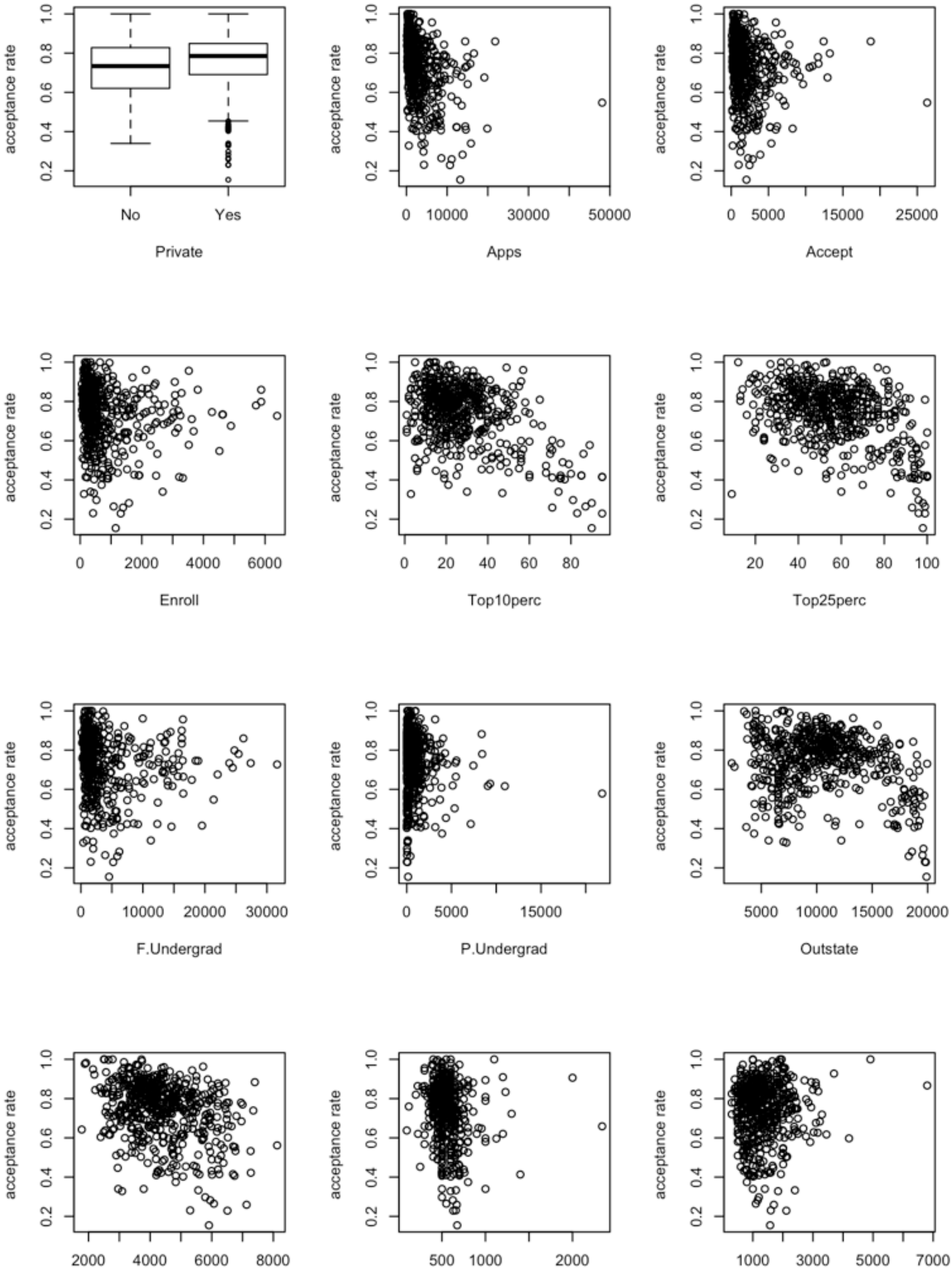
```r
library(SignifReg)
library(leaps)
library(boot)
```

Problem 3:

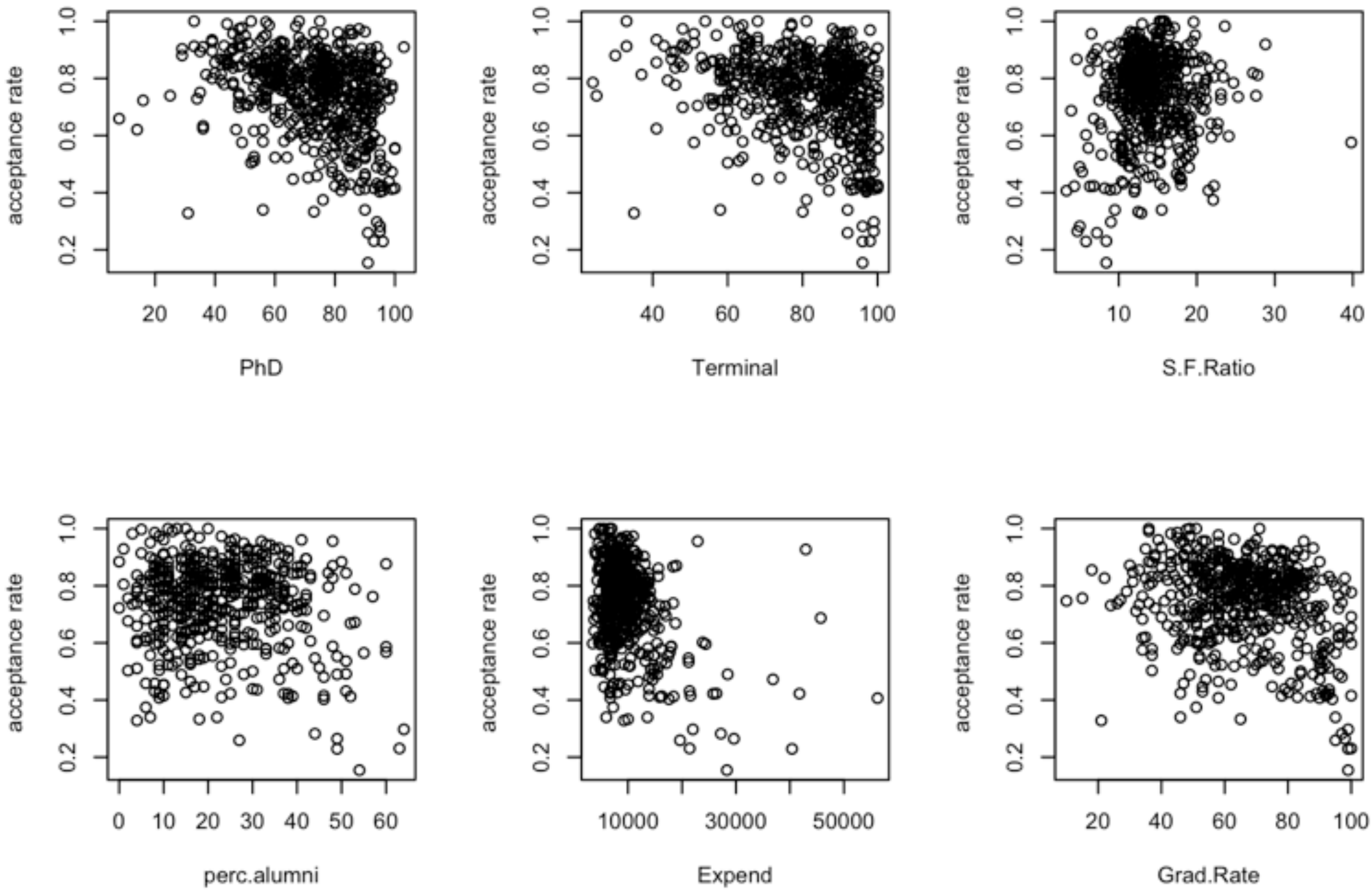
####(a) It seems that “Top10perc”,“Top25perc”,“Room.Board” and “S.F ratio” are predictive as they seem to have linear relationship with “acceptance rate”

```r
set.seed(234)
college_raw = College

#create new response variable
college = college_raw %>% mutate(accept_rate = Accept / Apps)


#split train/test dataset
RNGkind(sample.kind = "Rejection")
nrows_college = nrow(college)
test_id = sample(nrows_college, floor(0.3*nrows_college))
college_train = college[-test_id,]
college_test = college[test_id,]
```

```
par(mfrow = c(2,3))
for(i in 1:18){
plot(college_train[,"accept_rate"]~college_train[,i],
      xlab = names(college_train)[i],
      ylab = "acceptance rate ")
}
```


acceptance rate
Private
No
Yes
Apps
Accept
Enroll
Top10perc
Top25perc
F.Undergrad
P.Undergrad
Outstate

Room.Board Books Personal



(b)

The training mse is 0.009480855 and the test mse is 0.008436671.

```
mod_college_full = lm(accept_rate ~ ., data = college_train)

#train_mse
train_mse_full = mean((mod_college_full$residuals)^2) %>% print()
```

```
## [1] 0.009480855
```

```
yhat_test = predict(mod_college_full,newdata = college_test)

#test_mse
test_mse_full = mean((college_test$accept_rate -yhat_test)^2) %>% print()
```

```
## [1] 0.008436671
```

(c)

The forward selection selects the full model.

The backward selection selects the model consists of variable“PrivateYes”,“Apps”,“Accept”,“Enroll”,“Outstate”,“Books”,“Personal”.

```
#forward selection
college_full_forward = SignifReg(lm(accept_rate~., data = college_train), alpha =
0.05, direction = "forward", correction = "None", trace = FALSE)

college_full_forward
```

```
##
## Call:
## lm(formula = accept_rate ~ ., data = college_train)
##
## Coefficients:
## (Intercept)   PrivateYes         Apps       Accept       Enroll
##   1.033e+00    4.605e-02   -6.493e-05    1.043e-04   -1.983e-05
##   Top10perc    Top25perc  F.Undergrad  P.Undergrad     Outstate
##   1.220e-04   -1.094e-03    1.138e-06   -7.170e-06   -1.282e-06
##  Room.Board        Books     Personal          PhD     Terminal
##  -1.756e-05   -6.954e-05    9.101e-06   -7.073e-04    3.552e-04
##   S.F.Ratio  perc.alumni       Expend    Grad.Rate
##  -4.332e-03    7.061e-04   -2.248e-06   -8.594e-04
```

```
#backward selection
college_full_backward = SignifReg(lm(accept_rate~., data = college_train), alpha =
0.05, direction = "backward", correction = "None", trace = FALSE)

college_full_backward
```

```
##
## Call:
## lm(formula = accept_rate ~ Private + Apps + Accept + Enroll +
##     Outstate + Books + Personal, data = college_train)
##
## Coefficients:
## (Intercept)   PrivateYes         Apps       Accept       Enroll
##   8.257e-01    6.022e-02   -7.247e-05    1.128e-04   -2.794e-05
##    Outstate        Books     Personal
##  -8.033e-06   -8.146e-05    1.122e-05
```

```
train_mse_forward = mean((college_full_forward$residuals)^2) %>% print()
```

```
## [1] 0.009480855
```

```
yhat_test_forward = predict(college_full_forward,newdata = college_test)

test_mse_forward = mean((college_test$accept_rate -yhat_test_forward)^2) %>% print
()
```

```
## [1] 0.008436671
```

```
train_mse_backward = mean((college_full_backward$residuals)^2) %>% print()
```

```
## [1] 0.01045669
```

```
yhat_test_backward = predict(college_full_backward,newdata = college_test)

test_mse_backward = mean((college_test$accept_rate -yhat_test_backward)^2) %>% pri
nt()
```

```
## [1] 0.008853622
```

(d)

Adjusted R^2:
predictors: “Private”,“Apps”,“Accept”,“Enroll”,“Top10perc”,“F.Undergrad”,“Outstate”,“Books”
,“Personal”,“PhD”,“Terminal”,“perc.alumni”,“Expend”,“Grad.Rate”,“accept_rate”
train mse: 0.009975353
test mse: 0.008628698

AIC:
predictors:“Private”,“Apps”,“Accept”,“Enroll”,“F.Undergrad”,“Outstate”,“Books”,“Personal”
,“perc.alumni”,“Grad.Rate”,“accept_rate”
train mse: 0.01023833
test mse: 0.008740803

BIC:
predictors: “Private”,“Apps”,“Accept”,“Enroll”,“F.Undergrad”,“Books”
train mse: 0.01023833
test mse: 0.009128133

```
regfit_full = regsubsets(accept_rate~. , data = college_train,nvmax = NULL)
selection = summary(regfit_full)
names_var = names(college_train)
```

```
#adjusted R squared
best_adjust_r2 = which.max(selection$adjr2)
var_adjust_r2 = names_var[selection$which[best_adjust_r2,]] %>% print()
```

```
##  [1] "Private"     "Apps"        "Accept"      "Enroll"      "Top10perc"
##  [6] "F.Undergrad" "Outstate"    "Books"       "Personal"    "PhD"
## [11] "Terminal"    "perc.alumni" "Expend"      "Grad.Rate"   "accept_rate"
```

```
mod_adjust_r2 = lm(accept_rate ~., data =college_train[var_adjust_r2])

train_mse_adjust_r2 = mean((mod_adjust_r2 $residuals^2)) %>% print()
```

```
## [1] 0.009975353
```

```
test_mse_adjust_r2 =mean((college_test$accept_rate-predict(mod_adjust_r2,college_t
est))^2)%>% print()
```

```
## [1] 0.008628698
```

```
#AIC (equivalent to Cp in this case)
best_AIC = which.min(selection$cp)
var_AIC = names_var[selection$which[best_AIC,]] %>% print()
```

```
##  [1] "Private"     "Apps"        "Accept"      "Enroll"      "F.Undergrad"
##  [6] "Outstate"    "Books"       "Personal"    "perc.alumni" "Grad.Rate"
## [11] "accept_rate"
```

```
mod_AIC = lm(accept_rate ~., data =college_train[var_AIC])

train_mse_AIC = mean((mod_AIC$residuals^2)) %>% print()
```

```
## [1] 0.01023833
```

```
test_mse_AIC =mean((college_test$accept_rate-predict(mod_AIC,college_test))^2)%>%
print()
```

```
## [1] 0.008740803
```

```
#BIC
best_BIC = which.min(selection$bic)
var_BIC = names_var[selection$which[best_BIC,]] %>% print()
```

```
## [1] "Private"     "Apps"         "Accept"       "Enroll"       "F.Undergrad"
## [6] "Books"
```

```
mod_BIC = lm(accept_rate ~., data =college_train[c(var_BIC,"accept_rate")])

train_mse_BIC = mean((mod_AIC$residuals^2)) %>% print()
```

```
## [1] 0.01023833
```

```
test_mse_BIC =mean((college_test$accept_rate-predict(mod_BIC,college_test))^2)%>%
print()
```

```
## [1] 0.009128133
```

(e)

```
set.seed(234)
glm_mod_full = glm(accept_rate~.,data = college_train)
cv_mse_mod_full = cv.glm(college_train, glm_mod_full,K = 5)$delta[1] %>% print()
```

```
## [1] 0.01137196
```

```
glm_forward_select = glm(college_full_forward)
cv_mse_forward_select = cv.glm(college_train, glm_forward_select ,K = 5)$delta[1]
%>% print()
```

```
## [1] 0.01145405
```

```
glm_backward_select = glm(college_full_backward)
cv_mse_backward_select = cv.glm(college_train,glm_backward_select,K = 5)$delta[1]%
>% print()
```

```
## [1] 0.01218588
```

```
glm_adjust_r2 = glm(mod_adjust_r2)
cv_mse_adjust_r2 = cv.glm(college_train,glm_adjust_r2,K = 5)$delta[1] %>% print()
```

```
## [1] 0.01215397
```

```
glm_AIC = glm(mod_AIC)
cv_mse_AIC = cv.glm(college_train, glm_AIC,K = 5)$delta[1] %>% print()
```

```
## [1] 0.01176597
```

```
glm_BIC = glm(mod_BIC)
cv_mse_BIC = cv.glm(college_train, glm_BIC,K = 5)$delta[1] %>% print()
```

```
## [1] 0.01383613
```

First: train error, Second: test error, Third: CV error
Full model: 0.009480855,0.008436671, 0.01137196

Forward:0.009178469,0.009523646, 0.01145405

Backward:0.01064957,0.01023246, 0.01218588

Adjusted_R2: 0.009975353, 0.008628698, 0.01215397

AIC:0.01023833, 0.00874080, 0.01176597

BIC:0.01023833, 0.009128133, 0.01383613

As we can see, training error are bigger than testing error, and cv error are greater than training error and testing error.