assignment

# Question 1

## a & b

mod<-y~x  
step<-function(x){ stepwise(mod,   
 direction = c("backward/forward", "forward/backward", "backward",  
 "forward","lasso"),   
 criterion = c("BIC", "AIC"))  
}  
  
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.0.5

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(leaps)

## Warning: package 'leaps' was built under R version 4.0.5

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.0.5

## Loading required package: Matrix

## Loaded glmnet 4.1-2

Hitters %>%  
 select(Salary) %>%  
 is.na() %>%  
 sum()

## [1] 59

Credit %>%  
 select(Balance) %>%  
 is.na() %>%  
 sum()

## [1] 0

Smarket %>%  
 select(Volume) %>%  
 is.na() %>%  
 sum()

## [1] 0

## c)

### i.For Hitters dataset

# Full Selection  
full <- regsubsets(Salary~., data = Hitters, nvmax = 19)  
summary(full)

## Subset selection object  
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19)  
## 19 Variables (and intercept)  
## Forced in Forced out  
## AtBat FALSE FALSE  
## Hits FALSE FALSE  
## HmRun FALSE FALSE  
## Runs FALSE FALSE  
## RBI FALSE FALSE  
## Walks FALSE FALSE  
## Years FALSE FALSE  
## CAtBat FALSE FALSE  
## CHits FALSE FALSE  
## CHmRun FALSE FALSE  
## CRuns FALSE FALSE  
## CRBI FALSE FALSE  
## CWalks FALSE FALSE  
## LeagueN FALSE FALSE  
## DivisionW FALSE FALSE  
## PutOuts FALSE FALSE  
## Assists FALSE FALSE  
## Errors FALSE FALSE  
## NewLeagueN FALSE FALSE  
## 1 subsets of each size up to 19  
## Selection Algorithm: exhaustive  
## AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI  
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " " " " "\*"   
## 2 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 3 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 4 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 5 ( 1 ) "\*" "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 6 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " " " "\*"   
## 7 ( 1 ) " " "\*" " " " " " " "\*" " " "\*" "\*" "\*" " " " "   
## 8 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " "\*" "\*" " "   
## 9 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 10 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 11 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 12 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 13 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 14 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" "\*" " " "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" " " "\*" "\*"   
## 17 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" " " "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN  
## 1 ( 1 ) " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " "\*" " " " " " "   
## 4 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 5 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 6 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 7 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 8 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 9 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 10 ( 1 ) "\*" " " "\*" "\*" "\*" " " " "   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## 12 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 14 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 15 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 17 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"

# Forward selection  
fwd<- regsubsets(Salary~., data = Hitters, nvmax = 19, method = "forward")  
summary(fwd)

## Subset selection object  
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19, method = "forward")  
## 19 Variables (and intercept)  
## Forced in Forced out  
## AtBat FALSE FALSE  
## Hits FALSE FALSE  
## HmRun FALSE FALSE  
## Runs FALSE FALSE  
## RBI FALSE FALSE  
## Walks FALSE FALSE  
## Years FALSE FALSE  
## CAtBat FALSE FALSE  
## CHits FALSE FALSE  
## CHmRun FALSE FALSE  
## CRuns FALSE FALSE  
## CRBI FALSE FALSE  
## CWalks FALSE FALSE  
## LeagueN FALSE FALSE  
## DivisionW FALSE FALSE  
## PutOuts FALSE FALSE  
## Assists FALSE FALSE  
## Errors FALSE FALSE  
## NewLeagueN FALSE FALSE  
## 1 subsets of each size up to 19  
## Selection Algorithm: forward  
## AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI  
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " " " " "\*"   
## 2 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 3 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 4 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 5 ( 1 ) "\*" "\*" " " " " " " " " " " " " " " " " " " "\*"   
## 6 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " " " "\*"   
## 7 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " " " "\*"   
## 8 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " "\*" "\*"   
## 9 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 10 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 11 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 12 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 13 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 14 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" "\*" " " "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" " " "\*" "\*"   
## 17 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" " " "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN  
## 1 ( 1 ) " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " "\*" " " " " " "   
## 4 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 5 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 6 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 7 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 8 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 9 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 10 ( 1 ) "\*" " " "\*" "\*" "\*" " " " "   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## 12 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 14 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 15 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 17 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"

# Backward selection  
bwd<-regsubsets(Salary~., data = Hitters, nvmax = 19, method = "backward")  
summary(bwd)

## Subset selection object  
## Call: regsubsets.formula(Salary ~ ., data = Hitters, nvmax = 19, method = "backward")  
## 19 Variables (and intercept)  
## Forced in Forced out  
## AtBat FALSE FALSE  
## Hits FALSE FALSE  
## HmRun FALSE FALSE  
## Runs FALSE FALSE  
## RBI FALSE FALSE  
## Walks FALSE FALSE  
## Years FALSE FALSE  
## CAtBat FALSE FALSE  
## CHits FALSE FALSE  
## CHmRun FALSE FALSE  
## CRuns FALSE FALSE  
## CRBI FALSE FALSE  
## CWalks FALSE FALSE  
## LeagueN FALSE FALSE  
## DivisionW FALSE FALSE  
## PutOuts FALSE FALSE  
## Assists FALSE FALSE  
## Errors FALSE FALSE  
## NewLeagueN FALSE FALSE  
## 1 subsets of each size up to 19  
## Selection Algorithm: backward  
## AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI  
## 1 ( 1 ) " " " " " " " " " " " " " " " " " " " " "\*" " "   
## 2 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " "\*" " "   
## 3 ( 1 ) " " "\*" " " " " " " " " " " " " " " " " "\*" " "   
## 4 ( 1 ) "\*" "\*" " " " " " " " " " " " " " " " " "\*" " "   
## 5 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " "\*" " "   
## 6 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " "\*" " "   
## 7 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " "\*" " "   
## 8 ( 1 ) "\*" "\*" " " " " " " "\*" " " " " " " " " "\*" "\*"   
## 9 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 10 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 11 ( 1 ) "\*" "\*" " " " " " " "\*" " " "\*" " " " " "\*" "\*"   
## 12 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 13 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 14 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" " " " " "\*" "\*"   
## 15 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" "\*" " " "\*" "\*"   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" " " "\*" "\*"   
## 17 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" " " "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN  
## 1 ( 1 ) " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " "\*" " " " " " "   
## 4 ( 1 ) " " " " " " "\*" " " " " " "   
## 5 ( 1 ) " " " " " " "\*" " " " " " "   
## 6 ( 1 ) " " " " "\*" "\*" " " " " " "   
## 7 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 8 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 9 ( 1 ) "\*" " " "\*" "\*" " " " " " "   
## 10 ( 1 ) "\*" " " "\*" "\*" "\*" " " " "   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## 12 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 14 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 15 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 16 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " "   
## 17 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 18 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 19 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*"

# Compare to see whether the forward and backward selections have improved the model  
# if they are any differences in the model  
coef(full, 2)

## (Intercept) Hits CRBI   
## -47.9559022 3.3008446 0.6898994

coef(fwd, 2)

## (Intercept) Hits CRBI   
## -47.9559022 3.3008446 0.6898994

coef(bwd, 2)

## (Intercept) Hits CRuns   
## -50.8174029 3.2257212 0.6614168

#### Conclussion: compairing the coofficients

* There is no significant difference between the full selection and the forward selection in the model so model remains unchanged.
* But the backward selection has a significant difference in the model as it removes the insignificant values hence brings out a bets fit for our model.

### ii. For Credit dataset

# Full Selection  
full1 <- regsubsets(Balance~., data = Credit, nvmax = 11)  
summary(full1)

## Subset selection object  
## Call: regsubsets.formula(Balance ~ ., data = Credit, nvmax = 11)  
## 12 Variables (and intercept)  
## Forced in Forced out  
## ID FALSE FALSE  
## Income FALSE FALSE  
## Limit FALSE FALSE  
## Rating FALSE FALSE  
## Cards FALSE FALSE  
## Age FALSE FALSE  
## Education FALSE FALSE  
## GenderFemale FALSE FALSE  
## StudentYes FALSE FALSE  
## MarriedYes FALSE FALSE  
## EthnicityAsian FALSE FALSE  
## EthnicityCaucasian FALSE FALSE  
## 1 subsets of each size up to 11  
## Selection Algorithm: exhaustive  
## ID Income Limit Rating Cards Age Education GenderFemale StudentYes  
## 1 ( 1 ) " " " " " " "\*" " " " " " " " " " "   
## 2 ( 1 ) " " "\*" " " "\*" " " " " " " " " " "   
## 3 ( 1 ) " " "\*" " " "\*" " " " " " " " " "\*"   
## 4 ( 1 ) " " "\*" "\*" " " "\*" " " " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*" "\*" "\*" " " " " " " "\*"   
## 6 ( 1 ) " " "\*" "\*" "\*" "\*" "\*" " " " " "\*"   
## 7 ( 1 ) " " "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## MarriedYes EthnicityAsian EthnicityCaucasian  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " " "   
## 5 ( 1 ) " " " " " "   
## 6 ( 1 ) " " " " " "   
## 7 ( 1 ) " " " " " "   
## 8 ( 1 ) " " " " " "   
## 9 ( 1 ) " " "\*" " "   
## 10 ( 1 ) "\*" "\*" " "   
## 11 ( 1 ) "\*" "\*" "\*"

fwd1<- regsubsets(Balance ~., data = Credit, nvmax = 11, method = "forward")  
summary(fwd1)

## Subset selection object  
## Call: regsubsets.formula(Balance ~ ., data = Credit, nvmax = 11, method = "forward")  
## 12 Variables (and intercept)  
## Forced in Forced out  
## ID FALSE FALSE  
## Income FALSE FALSE  
## Limit FALSE FALSE  
## Rating FALSE FALSE  
## Cards FALSE FALSE  
## Age FALSE FALSE  
## Education FALSE FALSE  
## GenderFemale FALSE FALSE  
## StudentYes FALSE FALSE  
## MarriedYes FALSE FALSE  
## EthnicityAsian FALSE FALSE  
## EthnicityCaucasian FALSE FALSE  
## 1 subsets of each size up to 11  
## Selection Algorithm: forward  
## ID Income Limit Rating Cards Age Education GenderFemale StudentYes  
## 1 ( 1 ) " " " " " " "\*" " " " " " " " " " "   
## 2 ( 1 ) " " "\*" " " "\*" " " " " " " " " " "   
## 3 ( 1 ) " " "\*" " " "\*" " " " " " " " " "\*"   
## 4 ( 1 ) " " "\*" "\*" "\*" " " " " " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*" "\*" "\*" " " " " " " "\*"   
## 6 ( 1 ) " " "\*" "\*" "\*" "\*" "\*" " " " " "\*"   
## 7 ( 1 ) " " "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## MarriedYes EthnicityAsian EthnicityCaucasian  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " " "   
## 5 ( 1 ) " " " " " "   
## 6 ( 1 ) " " " " " "   
## 7 ( 1 ) " " " " " "   
## 8 ( 1 ) " " " " " "   
## 9 ( 1 ) " " "\*" " "   
## 10 ( 1 ) "\*" "\*" " "   
## 11 ( 1 ) "\*" "\*" "\*"

# Backward selection  
bwd1<-regsubsets(Balance~., data = Credit, nvmax = 11, method = "backward")  
summary(bwd1)

## Subset selection object  
## Call: regsubsets.formula(Balance ~ ., data = Credit, nvmax = 11, method = "backward")  
## 12 Variables (and intercept)  
## Forced in Forced out  
## ID FALSE FALSE  
## Income FALSE FALSE  
## Limit FALSE FALSE  
## Rating FALSE FALSE  
## Cards FALSE FALSE  
## Age FALSE FALSE  
## Education FALSE FALSE  
## GenderFemale FALSE FALSE  
## StudentYes FALSE FALSE  
## MarriedYes FALSE FALSE  
## EthnicityAsian FALSE FALSE  
## EthnicityCaucasian FALSE FALSE  
## 1 subsets of each size up to 11  
## Selection Algorithm: backward  
## ID Income Limit Rating Cards Age Education GenderFemale StudentYes  
## 1 ( 1 ) " " " " "\*" " " " " " " " " " " " "   
## 2 ( 1 ) " " "\*" "\*" " " " " " " " " " " " "   
## 3 ( 1 ) " " "\*" "\*" " " " " " " " " " " "\*"   
## 4 ( 1 ) " " "\*" "\*" " " "\*" " " " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*" "\*" "\*" " " " " " " "\*"   
## 6 ( 1 ) " " "\*" "\*" "\*" "\*" "\*" " " " " "\*"   
## 7 ( 1 ) " " "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## MarriedYes EthnicityAsian EthnicityCaucasian  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " " "   
## 5 ( 1 ) " " " " " "   
## 6 ( 1 ) " " " " " "   
## 7 ( 1 ) " " " " " "   
## 8 ( 1 ) " " " " " "   
## 9 ( 1 ) " " "\*" " "   
## 10 ( 1 ) "\*" "\*" " "   
## 11 ( 1 ) "\*" "\*" "\*"

# Compare to see whether the forward and backward selections have improved the model  
# if they are any differences in the model  
coef(full1, 3)

## (Intercept) Income Rating StudentYes   
## -581.078888 -7.874931 3.987472 418.760284

coef(fwd1, 3)

## (Intercept) Income Rating StudentYes   
## -581.078888 -7.874931 3.987472 418.760284

coef(bwd1, 3)

## (Intercept) Income Limit StudentYes   
## -432.3374179 -7.9016203 0.2675379 427.0232667

#### Conclussion: compairing the coofficients

* Looking at the intercepts, there is no significant difference in the model for the full selection and the forward selection algorithm.
* But there is a significant difference in the backward selection hence bringing out the best fit for our model since it removes the insignificant values from our model.

### iii. For Smarket dataset

# Full Selection  
full2 <- regsubsets (Volume~., data = Smarket, nvmax = 8)  
summary(full2)

## Subset selection object  
## Call: regsubsets.formula(Volume ~ ., data = Smarket, nvmax = 8)  
## 8 Variables (and intercept)  
## Forced in Forced out  
## Year FALSE FALSE  
## Lag1 FALSE FALSE  
## Lag2 FALSE FALSE  
## Lag3 FALSE FALSE  
## Lag4 FALSE FALSE  
## Lag5 FALSE FALSE  
## Today FALSE FALSE  
## DirectionUp FALSE FALSE  
## 1 subsets of each size up to 8  
## Selection Algorithm: exhaustive  
## Year Lag1 Lag2 Lag3 Lag4 Lag5 Today DirectionUp  
## 1 ( 1 ) "\*" " " " " " " " " " " " " " "   
## 2 ( 1 ) "\*" " " " " " " "\*" " " " " " "   
## 3 ( 1 ) "\*" " " " " "\*" "\*" " " " " " "   
## 4 ( 1 ) "\*" " " "\*" "\*" "\*" " " " " " "   
## 5 ( 1 ) "\*" " " "\*" "\*" "\*" "\*" " " " "   
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

# forward selection  
fwd2<- regsubsets(Volume ~., data = Smarket, nvmax = 8, method = "forward")  
summary(fwd2)

## Subset selection object  
## Call: regsubsets.formula(Volume ~ ., data = Smarket, nvmax = 8, method = "forward")  
## 8 Variables (and intercept)  
## Forced in Forced out  
## Year FALSE FALSE  
## Lag1 FALSE FALSE  
## Lag2 FALSE FALSE  
## Lag3 FALSE FALSE  
## Lag4 FALSE FALSE  
## Lag5 FALSE FALSE  
## Today FALSE FALSE  
## DirectionUp FALSE FALSE  
## 1 subsets of each size up to 8  
## Selection Algorithm: forward  
## Year Lag1 Lag2 Lag3 Lag4 Lag5 Today DirectionUp  
## 1 ( 1 ) "\*" " " " " " " " " " " " " " "   
## 2 ( 1 ) "\*" " " " " " " "\*" " " " " " "   
## 3 ( 1 ) "\*" " " " " "\*" "\*" " " " " " "   
## 4 ( 1 ) "\*" " " "\*" "\*" "\*" " " " " " "   
## 5 ( 1 ) "\*" " " "\*" "\*" "\*" "\*" " " " "   
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

# Backward selection  
bwd2<-regsubsets(Volume~., data = Smarket, nvmax = 8, method = "backward")  
summary(bwd2)

## Subset selection object  
## Call: regsubsets.formula(Volume ~ ., data = Smarket, nvmax = 8, method = "backward")  
## 8 Variables (and intercept)  
## Forced in Forced out  
## Year FALSE FALSE  
## Lag1 FALSE FALSE  
## Lag2 FALSE FALSE  
## Lag3 FALSE FALSE  
## Lag4 FALSE FALSE  
## Lag5 FALSE FALSE  
## Today FALSE FALSE  
## DirectionUp FALSE FALSE  
## 1 subsets of each size up to 8  
## Selection Algorithm: backward  
## Year Lag1 Lag2 Lag3 Lag4 Lag5 Today DirectionUp  
## 1 ( 1 ) "\*" " " " " " " " " " " " " " "   
## 2 ( 1 ) "\*" " " " " " " "\*" " " " " " "   
## 3 ( 1 ) "\*" " " " " "\*" "\*" " " " " " "   
## 4 ( 1 ) "\*" " " "\*" "\*" "\*" " " " " " "   
## 5 ( 1 ) "\*" " " "\*" "\*" "\*" "\*" " " " "   
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

# Compare to see whether the forward and backward selections have improved the model  
# if they are any differences in the model  
coef(full2, 4)

## (Intercept) Year Lag2 Lag3 Lag4   
## -277.97536944 0.13951652 -0.01981232 -0.02001162 -0.02217697

coef(fwd2, 4)

## (Intercept) Year Lag2 Lag3 Lag4   
## -277.97536944 0.13951652 -0.01981232 -0.02001162 -0.02217697

coef(bwd2, 4)

## (Intercept) Year Lag2 Lag3 Lag4   
## -277.97536944 0.13951652 -0.01981232 -0.02001162 -0.02217697

#### Conclussion

Basing on intercepts, both the forward and backward selection methods have a no significant impact on the model as it remains the same.

Exercise 2. (5 points) For this exercise, the only extra package allowed is ISLR. a) Write an R function that as input takes a matrix X and as output returns the standarized matrix, by which we mean each column of X must be standarized so it would have a sample average equal be 0 and a sample standard deviation equal to 1. (That is, for each column, subtract its sample average from it and then divide it by its standard deviations. This will result in the new transformed column having a sample average equal be 0 and a sample standard deviation equal to 1.)\*\*

answer

mat<-function(x){  
 for (j in 1:ncol(x)) {  
 smean<-mean(x[,j])  
 ssd<-sqrt(var(x[,j]))  
 for (i in 1:nrow(x)) {  
 x[i,j]=x[i,j]+smean/ssd  
   
 }  
   
 }  
 return(x)  
}

1. Constuct a matrix from the Smarket dataset that contains all the variables except for Year and Direction. Apply your function to that matrix and then compute summary statistics and the covariance matrix of the output of your function.

library(ISLR)  
data=Smarket[,-c(1,9)]  
result=mat(data)  
head(result)

## Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today  
## 1 0.3843745 -0.1885509 -2.622493 -1.0535634 5.0148883 5.293634 0.9617619  
## 2 0.9623745 0.3844491 -0.190493 -2.6225634 -1.0501117 5.398834 1.0347619  
## 3 1.0353745 0.9624491 0.382507 -0.1905634 -2.6191117 5.513534 -0.6202381  
## 4 -0.6196255 1.0354491 0.960507 0.3824366 -0.1871117 5.378334 0.6167619  
## 5 0.6173745 -0.6195509 1.033507 0.9604366 0.3858883 5.308034 0.2157619  
## 6 0.2163745 0.6174491 -0.621493 1.0334366 0.9638883 5.451434 1.3947619

summary(result)

## Lag1 Lag2 Lag3   
## Min. :-4.918626 Min. :-4.918551 Min. :-4.920493   
## 1st Qu.:-0.636126 1st Qu.:-0.636051 1st Qu.:-0.638493   
## Median : 0.042374 Median : 0.042449 Median : 0.040007   
## Mean : 0.007209 Mean : 0.007368 Mean : 0.003223   
## 3rd Qu.: 0.600124 3rd Qu.: 0.600199 3rd Qu.: 0.598257   
## Max. : 5.736374 Max. : 5.736449 Max. : 5.734507   
## Lag4 Lag5 Volume Today   
## Min. :-4.920563 Min. :-4.91711 Min. :4.458 Min. :-4.91924   
## 1st Qu.:-0.638563 1st Qu.:-0.63511 1st Qu.:5.360 1st Qu.:-0.63674   
## Median : 0.039937 Median : 0.04339 Median :5.525 Median : 0.04126   
## Mean : 0.003073 Mean : 0.01050 Mean :5.581 Mean : 0.00590   
## 3rd Qu.: 0.598187 3rd Qu.: 0.60189 3rd Qu.:5.744 3rd Qu.: 0.59951   
## Max. : 5.734437 Max. : 5.73789 Max. :7.255 Max. : 5.73576

cov(result)

## Lag1 Lag2 Lag3 Lag4 Lag5  
## Lag1 1.291175062 -0.033950025 -0.013978596 -0.003863730 -0.007399459  
## Lag2 -0.033950025 1.291132819 -0.033507330 -0.014044104 -0.004639348  
## Lag3 -0.013978596 -0.033507330 1.296644424 -0.031187581 -0.024577209  
## Lag4 -0.003863730 -0.014044104 -0.031187581 1.296805622 -0.035392903  
## Lag5 -0.007399459 -0.004639348 -0.024577209 -0.035392903 1.316871484  
## Volume 0.016751517 -0.017763979 -0.017161918 -0.019867521 -0.009098570  
## Today -0.033771790 -0.013234781 -0.003167126 -0.008928177 -0.045457562  
## Volume Today  
## Lag1 0.016751517 -0.033771790  
## Lag2 -0.017763979 -0.013234781  
## Lag3 -0.017161918 -0.003167126  
## Lag4 -0.019867521 -0.008928177  
## Lag5 -0.009098570 -0.045457562  
## Volume 0.129857232 0.005975148  
## Today 0.005975148 1.291255147

## Exercise 3. (5 points)

For this exercise, no extra packages are allowed. We are interested in computing the integral

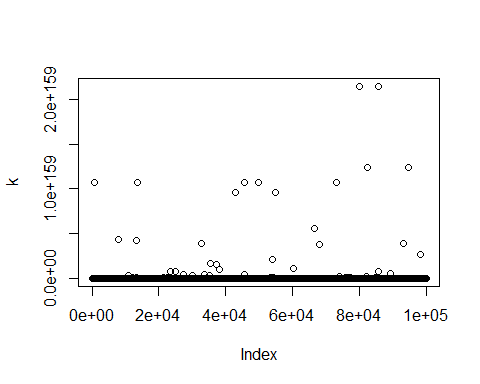
1. Write an R function that computes an approximation to the value of the area under the function f between 0 and 1. This function should take as input argument a number M, which corresponds to the number of Monte Carlo experiments desired for the approximation. This function should return the numerical approximation as the output argument

f <- function(M) {(M^2)\*exp(-M^2)}  
integrate(f, lower =0, upper = 1)

## 0.1894723 with absolute error < 2.1e-15

1. Design a Monte Carlo experiment (using 1000 replications) that computes the accuracy of the previous procedure by attaching a standard error to the approximation. Use then your experiment to compute the standard errors for 12 dierent values of M (100; 1000; 10000; 20000; 30000; :::; 100000). Create a plot where dierent values for M are on the x-axis and where the standard errors are on the y-axis. To compare, plot on the same figure the function f(M)=0.13/ sqrt(M) and comment the plot.

se <- function(M) sd(M) / sqrt(length(M))   
m1<- rnorm(100000,mean=1,sd=10)  
f1<-f(m1)  
se1<-se(m1)  
  
#  
k=0.13/sqrt(f1)  
plot(k)



Conclusion

Since all the points from the plot are scattered, the runs have a uniform distributiuon hence equal chance of occurring.