Hw6

Bijesh Mishra

4/5/2021

Machine Learning Chapter 6: Model Selection:

```
rm(list = ls())
setwd("~/Dropbox/OSU/PhD/SemVISp2021/STAT5063ML/Homeworks/hw6")
library(MASS)
library(boot)
library(ISLR)
library(class)
library(readxl)
library(carData, warn.conflicts = F)
library(car, warn.conflicts = F)
library(leaps, warn.conflicts = F) #  All-subset Regression.
library(pls, warn.conflicts = F) # Parital Least Square, Principle Component Regression.
library(Matrix, warn.conflicts = F, logical.return = F)
require(glmnet, quietly = T, warn.conflicts = F) #Ridge & LASSO (Penalized) Regression.
```

```
## Loaded glmnet 4.1-1
```

```
studentdata2019 = read_excel("//Users//bmishra//Dropbox//OSU//PhD//SemVISp2021//STAT5063ML//Data//Studer
hw6.data = setNames(studentdata2019,
                    tolower(names(studentdata2019))) #lower case names.
attach(hw6.data, pos = 2L, warn.conflicts = F)
# names(hw6.data)
as.data.frame(hw6.data[c(4,36),])
```

```
##   gender    class hsclass txtsent txtrec fbtime pinterest snapchat introvert
## 1      F STAT2023     123      18     28     45         N        Y         4
## 2      M STAT5063      36      10     20      5         Y        N         4
##   year
## 1 2016
## 2 2019
```

Q1.A: Subset Selection Model:

```
# Step 1: Get a model fit with all variables.
fit.q1a = regsubsets(introvert ~ .,
                     data = hw6.data,
                     nvmax = 9) # Maximum size of variables.
summary.fitq1a =  summary(fit.q1a)
summary.fitq1a
```

```
## Subset selection object
## Call: regsubsets.formula(introvert ~ ., data = hw6.data, nvmax = 9)
## 9 Variables  (and intercept)
```

```
##               Forced in Forced out
## genderM           FALSE      FALSE
## classSTAT5063     FALSE      FALSE
## hsclass           FALSE      FALSE
## txtsent           FALSE      FALSE
## txtrec            FALSE      FALSE
## fbtime            FALSE      FALSE
## pinterestY        FALSE      FALSE
## snapchatY         FALSE      FALSE
## year              FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: exhaustive
##          genderM classSTAT5063 hsclass txtsent txtrec fbtime pinterestY
## 1  ( 1 ) " "     " "           " "     "*"     " "    " "    " "
## 2  ( 1 ) " "     " "           " "     "*"     " "    " "    " "
## 3  ( 1 ) " "     " "           " "     "*"     " "    " "    " "
## 4  ( 1 ) " "     "*"           " "     "*"     " "    " "    "*"
## 5  ( 1 ) " "     "*"           " "     "*"     "*"    " "    "*"
## 6  ( 1 ) "*"     "*"           " "     "*"     "*"    " "    "*"
## 7  ( 1 ) "*"     "*"           " "     "*"     "*"    "*"    "*"
## 8  ( 1 ) "*"     "*"           "*"     "*"     "*"    "*"    "*"
## 9  ( 1 ) "*"     "*"           "*"     "*"     "*"    "*"    "*"
##          snapchatY year
## 1  ( 1 ) " "       " "
## 2  ( 1 ) " "       "*"
## 3  ( 1 ) "*"       "*"
## 4  ( 1 ) " "       "*"
## 5  ( 1 ) " "       "*"
## 6  ( 1 ) " "       "*"
## 7  ( 1 ) " "       "*"
## 8  ( 1 ) " "       "*"
## 9  ( 1 ) "*"       "*"
```

```
# Step 2: Identify Selection Criteria.
attach(summary.fitq1a)

which.min(bic) # Minimum BIC.
```

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

```
which.min(cp) # Minimum Mallow Cp.
```

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

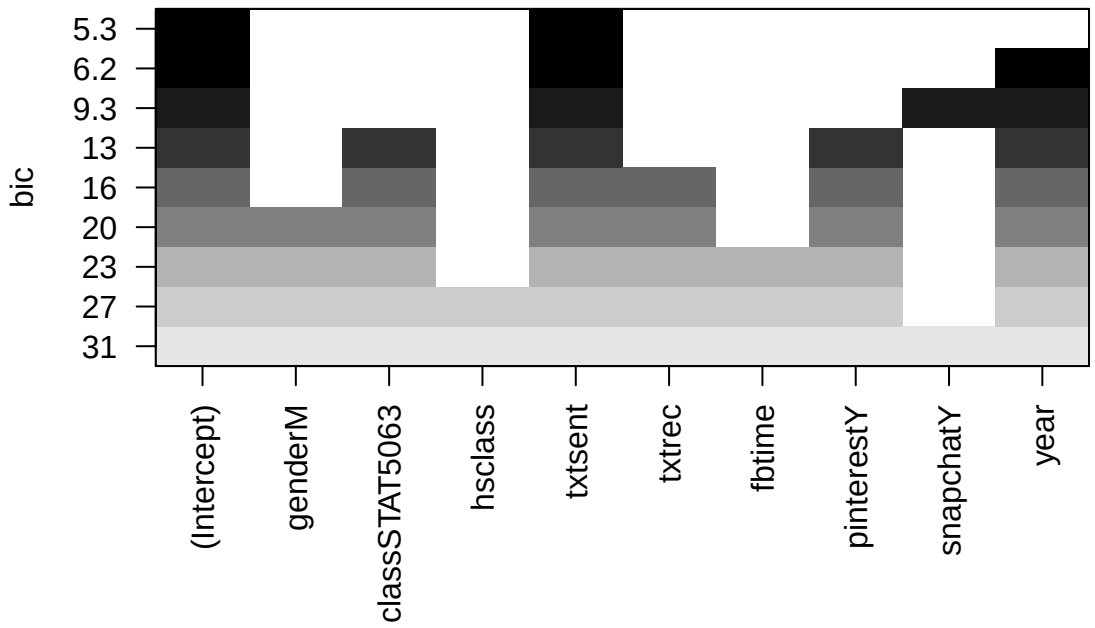
```
which.max(adjr2) #Maximum Adjusted R-Squared.
```

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

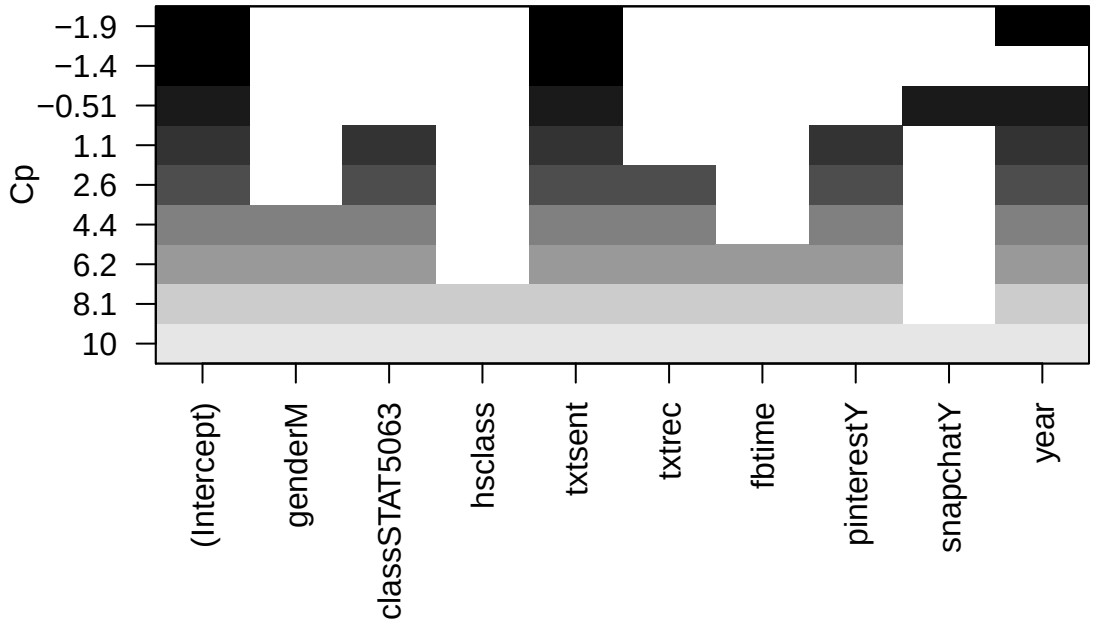
```
cat("Parameter to be included on the model based on smallest value of BIC is textsent and based
    on Adjusted R-Squared (maximum value) and Mallow CP. (Smallest Value) are txtsent and year.")
```

```
## Parameter to be included on the model based on smallest value of BIC is textsent and based
##     on Adjusted R-Squared (maximum value) and Mallow CP. (Smallest Value) are txtsent and year.
```

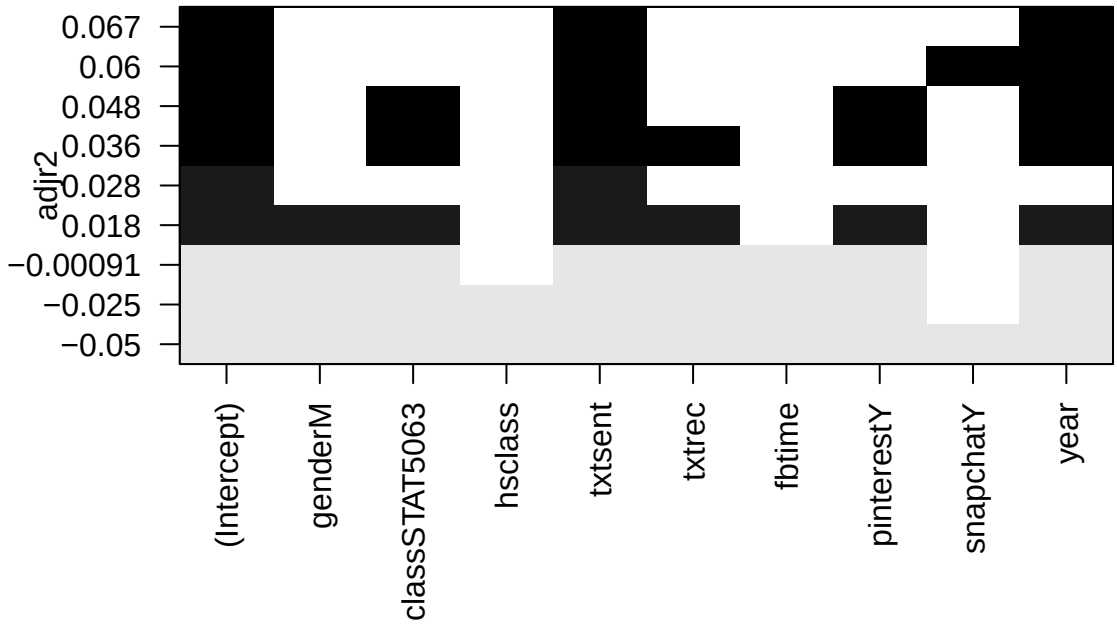
```
# Plotting Models:
plot(fit.q1a, scale = "bic",
     main = "BIC based Model Fit")
```



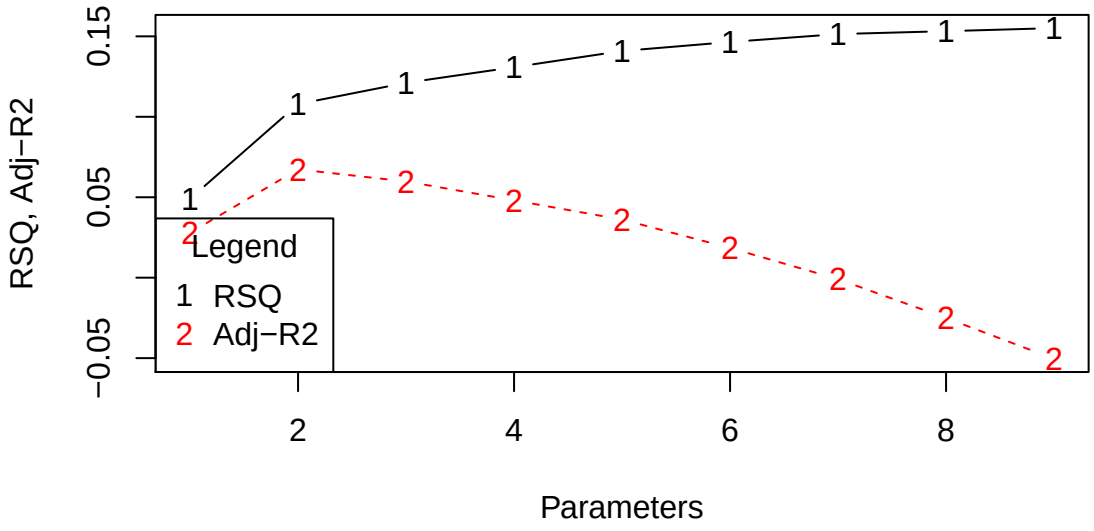

```
plot(fit.q1a, scale = "Cp",
     main = "Mallow Cp. based Model Fit")
```



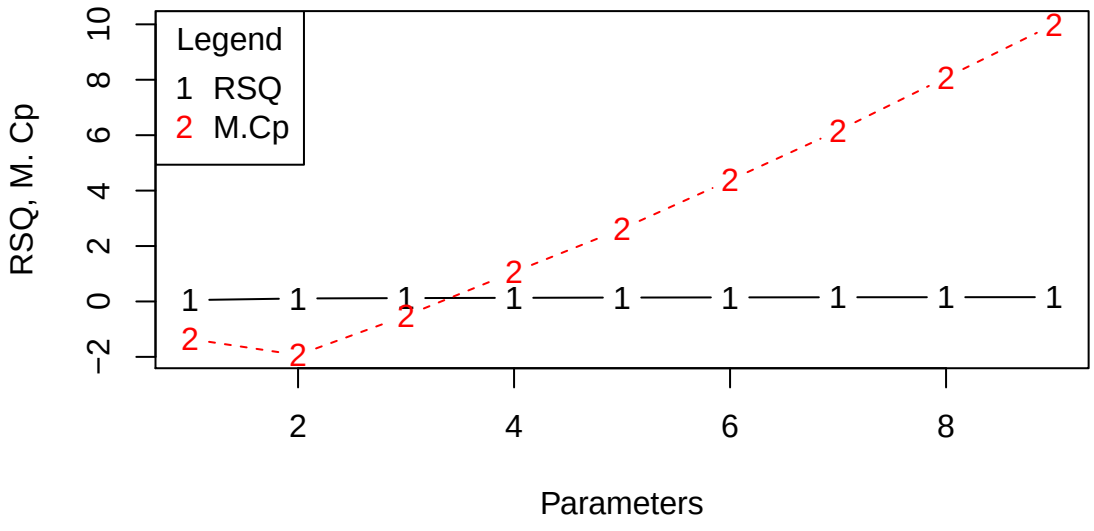

```
plot(fit.q1a, scale = "adjr2",
     main = "Adj-R2 based Model Fit")
```




```
# Matplot:
matplot(1:9, cbind(rsq, adjr2), type = "b",
        xlab = "Parameters", ylab = "RSQ, Adj-R2",
        main = "Parameters Vs. RSQ, & Adj-R2")
legend("bottomleft",
       col = c(1, 2),
       title = "Legend",
       pch = c("1", "2"),
       legend = c("RSQ",
                  "Adj-R2"))
```

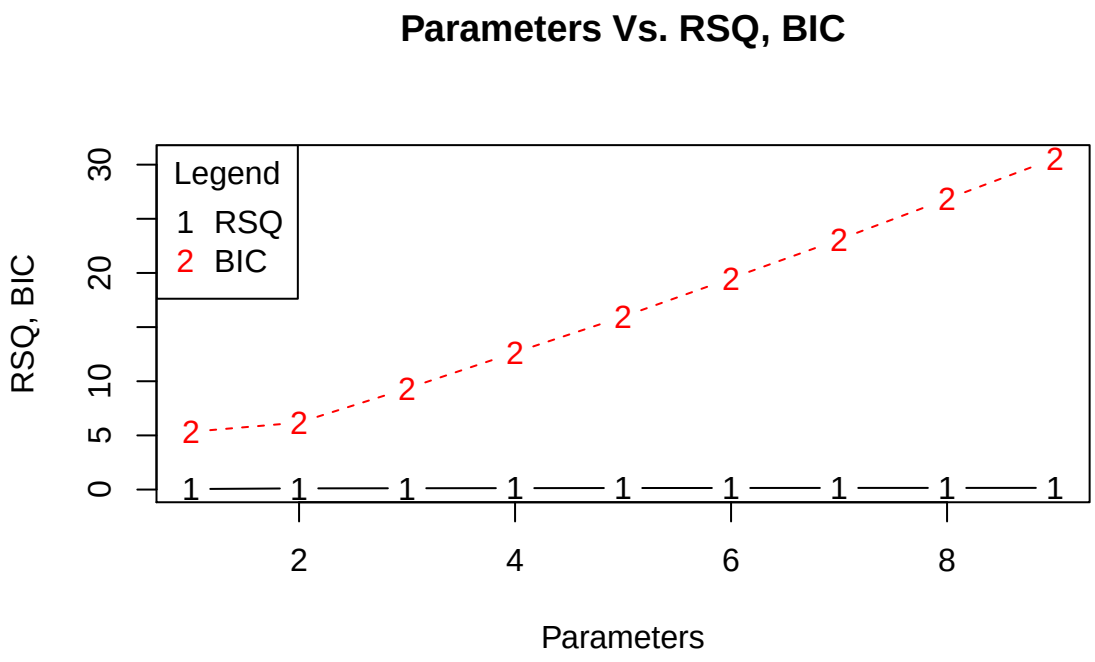



```
matplot(1:9, cbind(rsq, cp), type = "b",
        xlab = "Parameters", ylab = "RSQ, M. Cp",
        main = "Parameters Vs. RSQ, M. Cp")
legend("topleft",
       col = c(1, 2),
       title = "Legend",
       pch = c("1", "2"),
       legend = c("RSQ",
                  "M.Cp"))
```




```
matplot(1:9, cbind(rsq, bic), type = "b",
        xlab = "Parameters", ylab = "RSQ, BIC",
        main = "Parameters Vs. RSQ, BIC")
```

```
legend("topleft",
       col = c(1, 2),
       title = "Legend",
       pch = c("1", "2"),
       legend = c("RSQ",
                  "BIC"))
```




```
## Answer: Also as reflected in each graphs above besides calculated values previously,

## In Parameters Vs. RSQ, & Adj-R2 graph, The Adj-R2 value is highest for model with 2 parameters.

## In Parameters Vs. RSQ, M. Cp graph, The M. Cp value is lowest for model with 2 parameters.

## In Parameters Vs. RSQ, BIC graph, The BIC value is lowest for model with 1 parameters.

## All of the charts above show minimized Residual Sum of Squares (RSS).

## These answers can also be visualized in the charts below as well: Eg. In BIC and M. Cp charts,
## we can see model with text sent and year has lowest BIC, and M.Cp values.
## But in Adj-R2 Chart, model with textsent has highest Adj-R2 value.
```

Q1.B: Report and interpret the R squared for any identified models above.

```
lm.q1b1 = lm(introvert ~ txtsent, data = hw6.data)
summary(lm.q1b1)
```

```
##
## Call:
## lm(formula = introvert ~ txtsent, data = hw6.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9608 -1.4246  0.0774  1.9455  3.4685
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.008477   0.402798  12.434 3.71e-16 ***
```

```
## txtsent     -0.009540   0.006261  -1.524    0.135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.086 on 45 degrees of freedom
## Multiple R-squared:  0.04906,    Adjusted R-squared:  0.02793
## F-statistic: 2.322 on 1 and 45 DF,  p-value: 0.1346
```

```
cat("Ans Q1.B: The R-squared value value of model with one (ie. textsent) predictor is 0.04906
    implies that 4.91 % variations in the model is explained by given predictor in the model. ")
```

```
## Ans Q1.B: The R-squared value value of model with one (ie. textsent) predictor is 0.04906
##     implies that 4.91 % variations in the model is explained by given predictor in the model.
```

```
lm.q1b2 = lm(introvert ~ txtsent + year, data = hw6.data)
summary(lm.q1b2)
```

```
##
## Call:
## lm(formula = introvert ~ txtsent + year, data = hw6.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3442 -1.7381 -0.0647  1.7663  3.9616
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 804.307220 469.243393   1.714   0.0936 .
## txtsent      -0.011690   0.006261  -1.867   0.0686 .
## year         -0.396282   0.232645  -1.703   0.0956 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.044 on 44 degrees of freedom
## Multiple R-squared:  0.1079, Adjusted R-squared:  0.06734
## F-statistic: 2.661 on 2 and 44 DF,  p-value: 0.08114
```

```
cat("Ans Q1.B: The R-squared value value of model with two (ie. textsent and year) predictors is
0.1079 implies that 10.79 % variation in the model is explained by given predictors in the model. ")
```

```
## Ans Q1.B: The R-squared value value of model with two (ie. textsent and year) predictors is
## 0.1079 implies that 10.79 % variation in the model is explained by given predictors in the model.
```

Q2: Dimension Reduction:

Q2.A: Principle Component Regression:

```
q2.data = (hw6.data[,c(3, 4, 5, 6, 9, 10)]) # Unscaled Data.
attach(q2.data, warn.conflicts = F)
as.data.frame(q2.data[1,])
```

```
##   hsclass txtsent txtrec fbtime introvert year
## 1       1       1      1     30         8 2016
```

```
# Step 1: Get PCA on centered and scaled data.
set.seed(1)
pcr.q2a = pcr(introvert ~ ., # Formula
              data = q2.data, # Unscaled Data
```

```
              scale = TRUE, # Scale the data
              validation = "LOO") # LOOCV Method.
summary(pcr.q2a)
```

```
## Data:    X dimension: 47 5
##  Y dimension: 47 1
## Fit method: svdpc
## Number of components considered: 5
##
## VALIDATION: RMSEP
## Cross-validated using 47 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
## CV           2.139    2.155    2.155    2.206    2.282    2.265
## adjCV        2.139    2.154    2.153    2.203    2.279    2.262
##
## TRAINING: % variance explained
##            1 comps  2 comps  3 comps  4 comps  5 comps
## X           40.694   66.846    82.79    95.75   100.00
## introvert    1.678    6.716     9.96    10.10    12.09
```

Based on Principle Component Regression (PCR), the leave one out cross validation error (LOOCV), as given by CV above, is lowest for the model with only intercept (models without any predictor variables).Thus, the final model is the model with only intercept.

Principle Component Regression (PCR): The R-squared value for predicting introver with one and five PCs are equal to 1.678% and 12.09 % respectively which implies that the variation explained by models with one PC and five PCs are 1.678% and 12.09% respectively.

Q2.B: Partial Least Square Regression:

```
# Maximize R-Squared values ie. correlation between y and yhat.
set.seed(1)
plsr.q2b = plsr(introvert ~ ., # Formula
              data = q2.data, # Unscaled Data
              scale = TRUE, # Scale the data
              validation = "LOO") # LOOCV Method.
summary(plsr.q2b)
```

```
## Data:    X dimension: 47 5
##  Y dimension: 47 1
## Fit method: kernelpls
## Number of components considered: 5
##
## VALIDATION: RMSEP
## Cross-validated using 47 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
## CV           2.139    2.237    2.282    2.299    2.275    2.265
## adjCV        2.139    2.234    2.279    2.295    2.271    2.262
##
## TRAINING: % variance explained
##            1 comps  2 comps  3 comps  4 comps  5 comps
## X           30.499    59.83    78.09    87.02   100.00
## introvert    9.658    11.05    11.69    12.09    12.09
```

Based on Partial Leaset Squares Regression (PLSR), the leave one out cross validation error (LOOCV), as given by CV above, is lowest for the model with only intercept (models without any predictor variables).Thus, the final model is the model with only intercept.

Partial Leaset Squares Regression (PLSR): The R-squared value for predicting introver with one and five PCs are equal to 9.658% and 12.09% respectively which implies that the variation explained by models with one PC and five PCs are 9.658% and 12.09% respectively. Note: Training: % variance explained in dependent variable row (introvert) give R-squared value of each model with one to five PCs.

Q2.C: Multiple regression model predicting introvert with 5 predictors.

```
lmfit.q2c = lm(introvert ~ ., # Formula
               data = q2.data)
summary(lmfit.q2c)
```

```
##
## Call:
## lm(formula = introvert ~ ., data = q2.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5803 -1.6491 -0.0939  1.9110  3.7323
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.160e+02  5.326e+02   1.532    0.133
## hsclass      7.139e-04  1.237e-03   0.577    0.567
## txtsent     -1.377e-02  1.030e-02  -1.338    0.188
## txtrec       8.790e-04  4.777e-03   0.184    0.855
## fbtime      -5.706e-03  1.044e-02  -0.547    0.588
## year        -4.021e-01  2.640e-01  -1.523    0.135
##
## Residual standard error: 2.101 on 41 degrees of freedom
## Multiple R-squared:  0.1209, Adjusted R-squared:  0.01369
## F-statistic: 1.128 on 5 and 41 DF,  p-value: 0.3611
```

P-value for H_o: $\beta_1 = \ldots = \beta_5 = 0$ is 0.3611, implies that we fail to reject H_o. Thus none of the variables is responsible for explaining the variation in the model. Yes, this is consistent with previous answers because based on both PCR and PSLR models, the smallest CV is for model with intercept only. The varation in the model explained by given set of five predictors is 12.09%.

Q3: LASSO.

Q3.A:

```
moma.q3 = model.matrix(snapchat ~ . -1, hw6.data) # Design Matrix w/o intercept.
y.q3 = hw6.data$snapchat # Dependent variable: Snapchat (1/0).
row1.momaq3 = moma.q3[1,] # First Row of Design Matrix.
row1.momaq3 # First Row of Design Matrix.
```

```
##       genderF       genderM classSTAT5063       hsclass       txtsent
##             0             1             0             1             1
##        txtrec        fbtime    pinterestY     introvert          year
##             1            30             0             8          2016
```

```
## Q3A Answer: Using the first row of design matrix,  the person was not enrolled in
##      STAT 5063 and is not a pinterest user. Thus I predict a person who is not enrolled in
##      STAT5063 is not a pinterest user
```

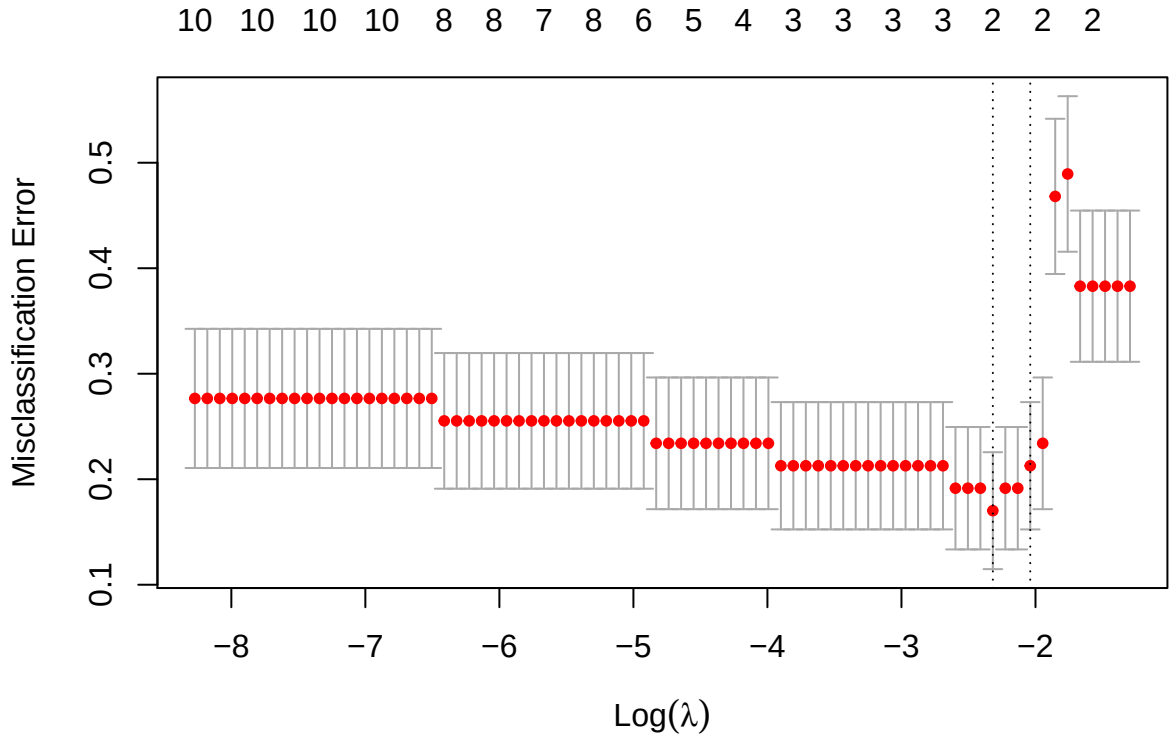
Q3.B:

```
# Step 1: Get lambda estimate, Cross Validation (CV) w/ Test Set.
set.seed(1)
```

```
q3.cv.glmnet = cv.glmnet(x = moma.q3, #Dependent Variables.
                         y = y.q3, # Snapchat: 1/0.
                         family = "binomial",  # Logistic Regression.
                         type.measure = "class", # Compute Classification Error.
                         alpha = 1, # 1 = LASSO ; 0  = Ridge Regression.
                         nfolds = length(y.q3)) # K-fold = # Obs. for LOOCV.
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```
plot(q3.cv.glmnet)
```




```
# Step 2: Get Minimum Lambda (Tuning Parameter) to tune the model.
lambda.hat.q3 = q3.cv.glmnet$lambda.min # Optimum Lambda/Tuining Parameter.
round(lambda.hat.q3, 3) # The LOOCV estimate for optimum lambda that minimize LOOCV test error
```

```
## [1] 0.098
```

```
## Answer: The LOOCV estimate for lambda is 0.098 .
```

Q3.C: Final Prediction Equation for Pr(Snapchat):

```
# Step 3: Use "lambda.min" to get LASSO (Or Ridge Regression) estimates:
set.seed(1)
q3.glmnet = glmnet(x = moma.q3, # Independent Variables
                   y = y.q3, # Dependent Variable # Snapchat: 1/0.
                   family = "binomial", # Logistic Regression.
                   alpha = 1, # 1 = LASSO ; 0  = Ridge Regression.
                   nfolds = length(y.q3), # K-fold = # Obs for LOOCV ?
                   lambda = lambda.hat.q3) # Optimum Lambda.
coef(q3.glmnet) # Coefficients.
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                         s0
## (Intercept)    0.77152625
## genderF        .
## genderM        .
## classSTAT5063 -1.21984502
## hsclass        .
## txtsent        0.01168128
## txtrec         .
## fbtime         .
## pinterestY     .
## introvert      .
## year           .
```

```
## Q3.C Answer: logit(Pr(Snapchat) = 0.772 + (0 x genderF) + (0 x GenderM) + ( -1.22 x ClassSTAT5063)
##      + (0 x hsclass) + ( 0.012 x txtsent) + (0 x txtrec) + (0 x fbtime) + (0 x pinterestY) +
##      (0 x year)
```

Q3.D: Estimated Probability of Student Having Snapchat Account.

```
newstu.q3d = data.matrix(cbind( genderF = 0, genderM = 1, classSTAT5063 = 1,
                                hasclass = 200, txtsent = 100, txtrec = 100,
                                fbtime  = 60, pinterestY= 1, introvert = 5,
                                year = 2021))

Snap.Pr.q3d = q3.glmnet$a0 + q3.glmnet$beta[1] * newstu.q3d[1] +
  q3.glmnet$beta[2] * newstu.q3d[2] + q3.glmnet$beta[3] * newstu.q3d[3] +
  q3.glmnet$beta[4] * newstu.q3d[4] + q3.glmnet$beta[5] * newstu.q3d[5] +
  q3.glmnet$beta[6] * newstu.q3d[6] + q3.glmnet$beta[7] * newstu.q3d[7] +
  q3.glmnet$beta[8] * newstu.q3d[8] + q3.glmnet$beta[9] * newstu.q3d[9] +
  q3.glmnet$beta[10] * newstu.q3d[10]
est.prob.q3d = exp(Snap.Pr.q3d)/(1 + exp(Snap.Pr.q3d))
est.prob.q3d
```

```
##       s0
## 0.672565
```

```
## Q3.D Answer: Estimated probability of a student having a Snapchat account given that
## they are: male, having pinterest account, are enrolled in STAT 5063, HSClass is 200,
## txtsent is 100, txtrec is 100, Fbtime is 60, introvert is 5 and year in 2021 = 0.673 .
```

Q4: Ridge Regression:

Q4.A: LOOCV Estimate for λ:

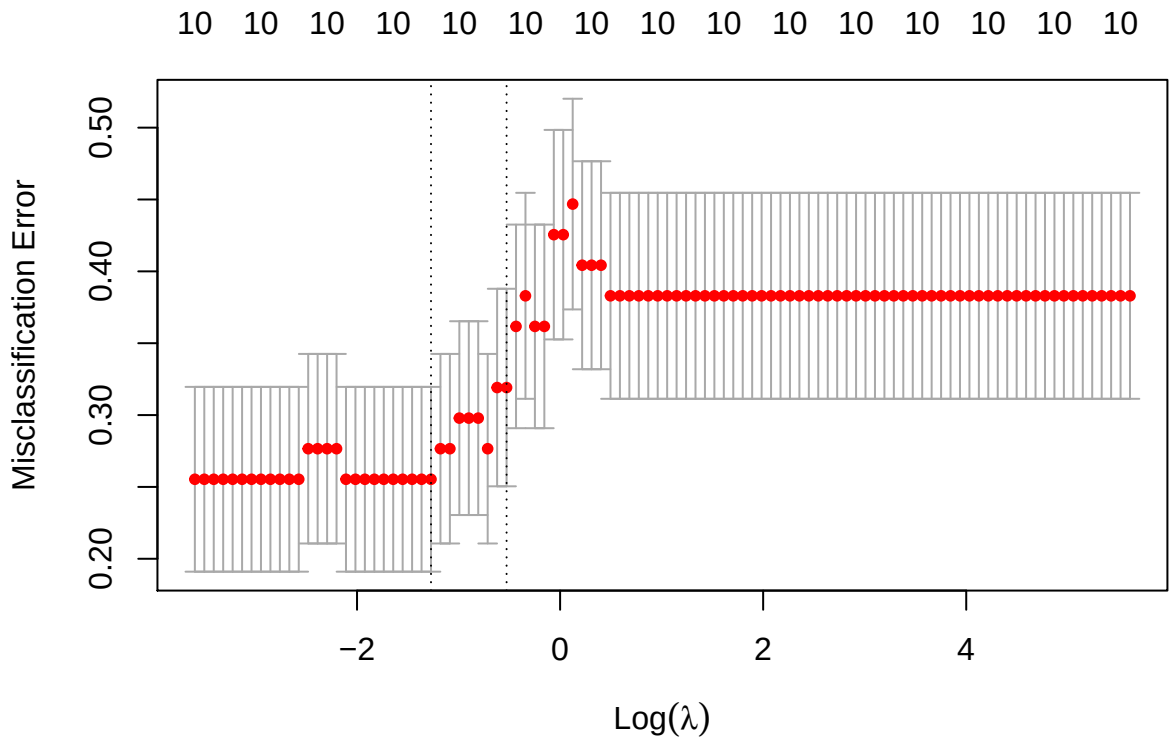
```
# moma.q3a is Model Design Matrix X and y.q4 is dependent variable.
moma.q4 = model.matrix(snapchat ~ . -1, hw6.data)
y.q4 = hw6.data$snapchat

# Step 1: Get lambda estimate, Cross Validation (CV) w/ Test Set.
set.seed(1)
q4.cv.glmnet = cv.glmnet(x = moma.q4,
                  y = y.q4, # Snapchat: 1/0.
                  family = "binomial",  # Logistic Regression.
                  alpha = 0, # Ridge Regression. 1 = LASSO.
                  type.measure = "class", #Compute classification error
                  nfolds = length(y.q4)) # K = 10 Fold CV.
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
```

```
## fold
```

```
plot(q4.cv.glmnet)
```




```
# Step 2: Get Minimum Lambda (Tuning Parameter) to tune the model.
lambda.hat.q4 = q4.cv.glmnet$lambda.min # Lambda.hat
lambda.hat.q4
```

```
## [1] 0.2804328
```

```
log(lambda.hat.q4)
```

```
## [1] -1.271421
```

```
cat("Q4.A Answer: The LOOCV Estimate for Lambda = ",
    round(lambda.hat.q4, 3))
```

```
## Q4.A Answer: The LOOCV Estimate for Lambda =  0.28
```

```
# Step 3: Use (tuining Parameter) to get LASSO estimates:
set.seed(1)
q4.glmnet = glmnet(x = moma.q4,
                   y = y.q4, # Snapchat: 1/0.
                   family = "binomial", # Logistic Regression
                   alpha = 0, # Ridge regression. 1 for LASSO.
                   nfolds = length(y.q4), # LOOCV
                   lambda = lambda.hat.q4)  # Minimize Lagrangean Multiplier to maximize constraints.
coef(q4.glmnet)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                              s0
## (Intercept)    1.156512e+02
## genderF        6.194048e-03
## genderM       -6.201608e-03
```

```
## classSTAT5063 -8.689216e-01
## hsclass        4.848951e-04
## txtsent        7.083973e-03
## txtrec         2.321192e-03
## fbtime         1.213837e-03
## pinterestY     3.565190e-01
## introvert      8.467651e-03
## year          -5.720309e-02
```

Q4.B: Estimated Probability Pr(Snapchat).

```
newst.q4 = data.matrix(cbind(genderF = 0, genderM = 1,
                             classSTAT5063 = 1, hasclass = 200,txtsent = 100,
                             txtrec = 100, fbtime  = 60, pinterestY= 1,
                             introvert = 5, year = 2021))

Snap.Pr.q4b = q4.glmnet$a0 + q4.glmnet$beta[1]*newst.q4[1] +
  q4.glmnet$beta[2]*newst.q4[2] + q4.glmnet$beta[3]*newst.q4[3] +
  q4.glmnet$beta[4]*newst.q4[4] + q4.glmnet$beta[5]*newst.q4[5] +
  q4.glmnet$beta[6]*newst.q4[6] + q4.glmnet$beta[7]*newst.q4[7] +
  q4.glmnet$beta[8]*newst.q4[8] + q4.glmnet$beta[9]*newst.q4[9] +
  q4.glmnet$beta[10]*newst.q4[10]
est.prob.q4b = exp(Snap.Pr.q4b)/(1 + exp(Snap.Pr.q4b))
est.prob.q4b
```

```
##        s0
## 0.6632595
```

```
## Q4.B Answer: Estimated probability of a student having a Snapchat account given that
## they are: male, having pinterest account, are enrolled in STAT 5063, HSClass is 200,
## txtsent is 100, txtrec is 100, Fbtime is 60, introvert is 5 and year in 2021 = 0.663 .
```

Q5.:

```
get.MSE.i = function(X, y, alpha = 0, i = 1)
{fit = cv.glmnet(x = X[-i,], y = y[-i],
                 alpha = alpha, nfolds = nrow(X[-i,]))
  lambda.hat = fit$lambda.min
  fit = glmnet(x=X[-i,],y=y[-i], alpha=alpha, lambda = lambda.hat)
  yhat = predict(fit,newx=X)[i]
  return((yhat-y[i])^2)}

f = function(X,y,alpha=0){
  MSE = rep(0,nrow(X))
for(i in 1:nrow(X)){
  MSE[i] = get.MSE.i(X,y,alpha,i)
}
  return(mean(MSE))
  }

moma.q5 = model.matrix(introvert ~ . -1, hw6.data) # Model Matrix
y.q5 = hw6.data$introvert # Introvert

Ridge.regression.MSE = f(moma.q5, y.q5, 0) # Ridge Regression
Ridge.regression.MSE
```

```
## [1] 4.577002
```

```r
cat("Ridge Regression MSE = ", round(Ridge.regression.MSE, 3))
```

```
## Ridge Regression MSE =  4.577
```

```r
LASSO.MSE = f(moma.q5, y.q5, 1) # LASSO
LASSO.MSE
```

```
## [1] 4.906606
```

```r
cat("LASSO MSE = ", round(LASSO.MSE, 3) )
```

```
## LASSO MSE =  4.907
```

Q5.A:function f is getting the LOOCV test error for a procedure that selects lambda using LOOCV. Cv.glmnet technically reports the training error since all the data is used to select lambda.

Q5.B: The test MSE estimate for LASSO (4.90) is than geater than that for Ridge Regression (4.58). So, we should use Ridge Regression to predict introvert with rest of the variables.

Q5.C: For categorical variable, we need to add family = "binomial", and type.measure = "class" to obtain misclassification error. We should look for "Binomial Deviance" instead of test MSE as we did in Q4.

Q5.D: If we do nfold = 10, it divides entire datasets into 10 different groups and allocate 9 datasets as training dataset and remaining 1 group as test dataset. Since there are more than one ways to allocate datasets into 10 groups, it is virtually impossible to reproduce same result everytime if seed is not set. set.seed() tells R to start picking random value from the same location using same random number algorithm which makes our work replicable.