Stat 897 Spring 2017 Data Analysis Assignment 3

Penn State

Due February 5, 2017

1. Use the College data found in the ISLR library. It contains a number of variables for 777 different universities and colleges in the US. The list of variables and their full description can be found on p. 54 of the text.

(a) Load the dataset College. Objective is to predict the number of applications received using the other variables in the data set.

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.2.5
```

```
library(leaps)
```

(b) Split your data set into a training set containing 100 observations and a test set containing the rest of the observations. For reproducibility of results use set.seed().

```
set.seed(15359)
train = sample(seq(1:777), 100, replace = FALSE)
College_train = College[ train,]
College_test  = College[-train,]
```

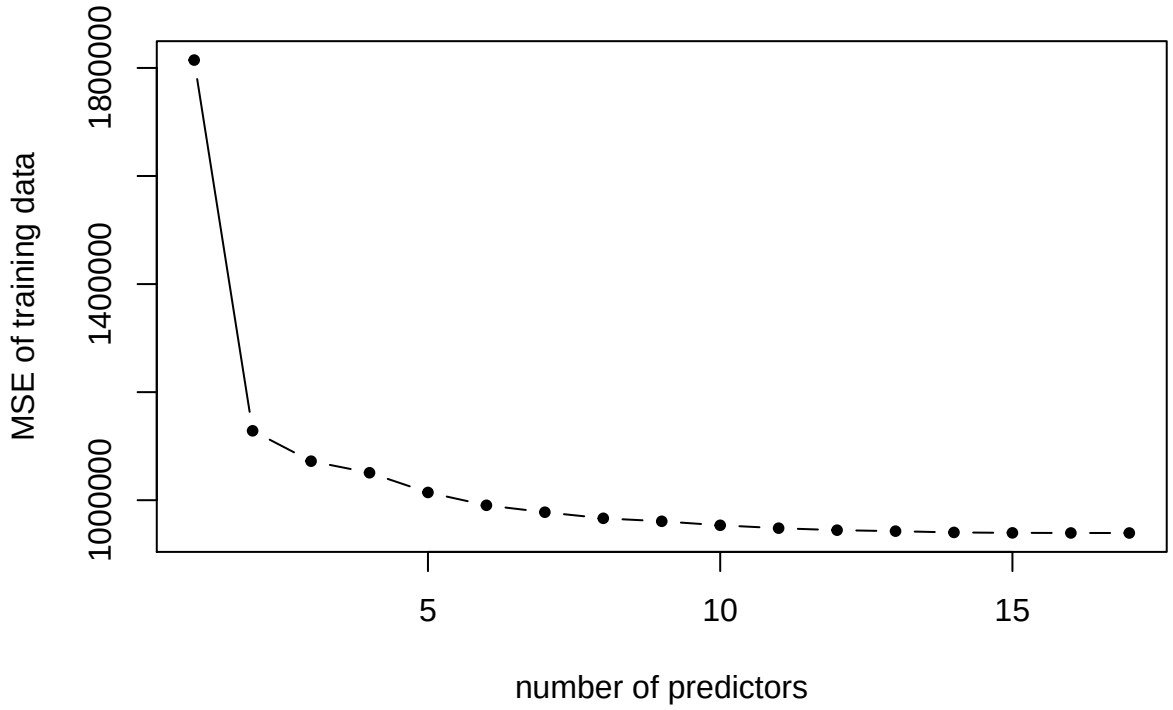
(c) Perform best subset selection on the training set, and plot the training set MSE associated with the best model of each size.

```
ntrain = nrow(College_train)
ntest  = nrow(College_test)
fit1   = regsubsets(Apps ~ ., data = College_train, nvmax = 17)

### predict function for regsubsets object
predict_regsubsets = function(object, newdata, id, ...){
  form   =  as.formula(~ .)
  mat    =  model.matrix(form, newdata)
  coefi  =  coef(object, id)
  xvars  =  names(coefi)
  return(mat[, xvars] %*% coefi)
}

mse_subset_train = rep(NA, 17)
for(i in 1:17){
  yhat_i = predict_regsubsets(fit1, newdata = College_train, id = i)
  mse_subset_train[i] = sum((College_train$Apps - yhat_i) ^ 2) / ntrain
}
```

```
plot(1:17, mse_subset_train, type = 'b', xlab = 'number of predictors',
     ylab = 'MSE of training data', pch = 20)
```

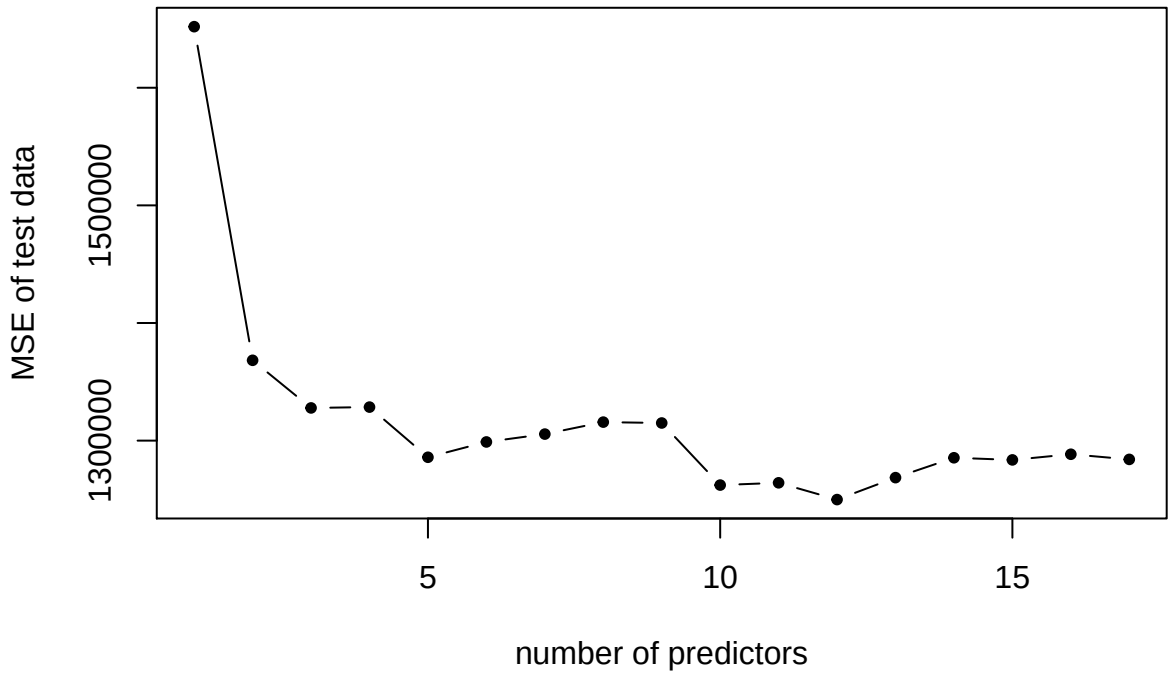


(d) Plot the test set MSE associated with the best model of each size.

```
mse_subset_test = rep(NA, 17)

for(i in 1:17){
  yhat_i = predict_regsubsets(fit1, newdata = College_test, id = i)
  mse_subset_test[i] = sum((College_test$Apps - yhat_i) ^ 2) / ntest
}

plot(1:17, mse_subset_test, type = 'b', xlab = 'number of predictors',
     ylab = 'MSE of test data', pch = 20)
```



(e) For which model size does the test set MSE take on its minimum value? Comment on your results.

```
best_modelsize = which.min(mse_subset_test)
best_modelsize
```

```
## [1] 12
```

Please check that none of the best models is the intercept only model, or the model with ALL predictors. If they are, try using a different seed value to avoid it.

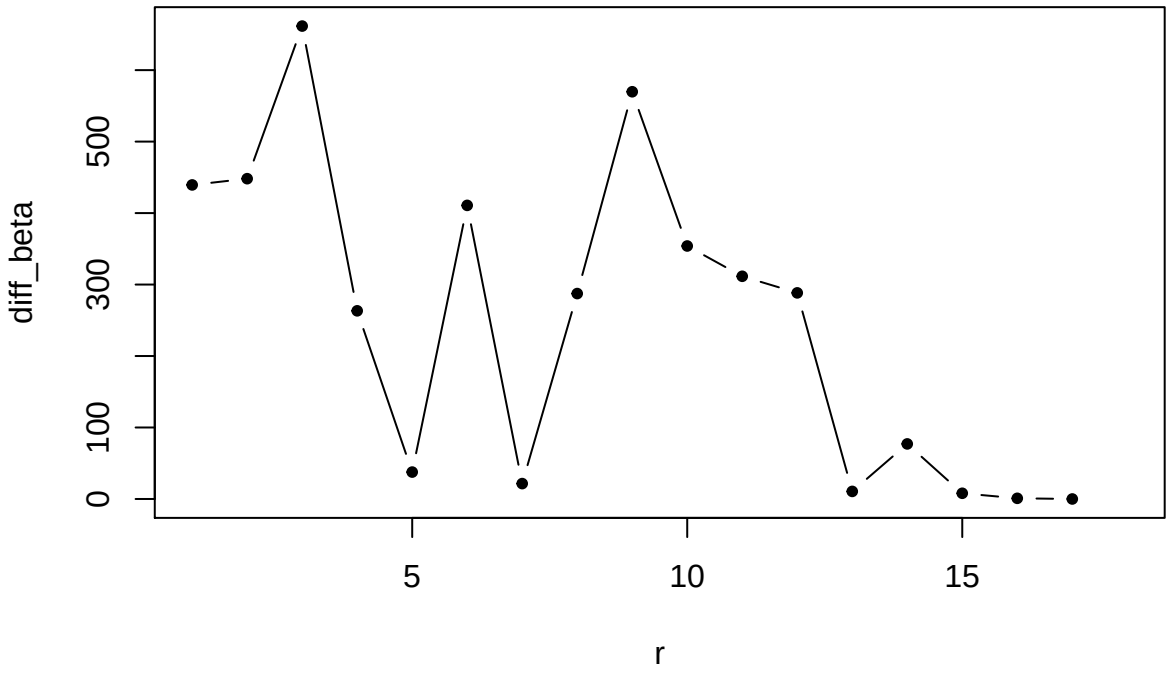
(f) Fit a regression model with all features to the full data containing 777 observations. Let the regression coefficients for this model be denoted by β_j. Let $\hat{\beta}_j^r$ be the estimated regression coefficient for the best model containing r features. Create a plot displaying

$$\sqrt{\sum_{j=1}(\beta_j - \hat{\beta}_j^r)^2}$$

for a range of values of r. Comment on what you observe. How does this plot compare to the test MSE plot from (d).

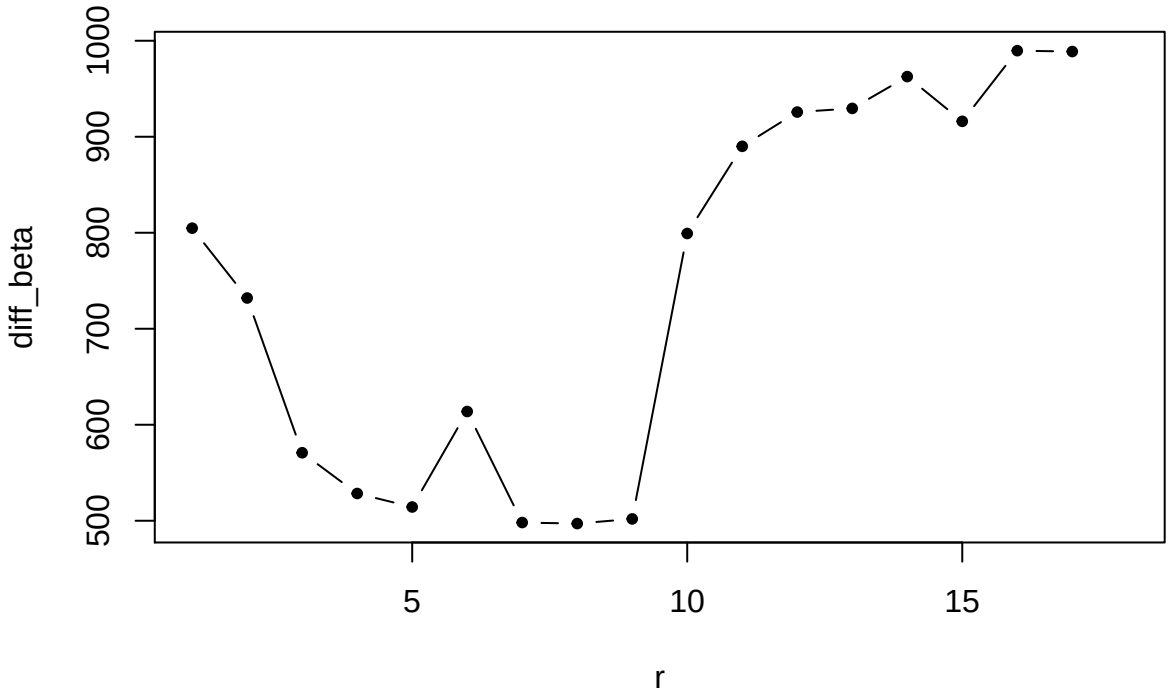
```
fit_full = lm(Apps ~., data = College)
fit2 = regsubsets(Apps ~., data = College, nvmax = 17)
betahat = fit_full$coefficients
diff_beta = rep(NA, 18)
for(i in 1:17){
  coefi = coef(fit2, id=i)
  diff_beta[i] = sqrt(sum((betahat[names(coefi)] - coefi) ^ 2))
```

```
}
plot(diff_beta, xlab = 'r', type = 'b', pch = 20)
```



There was some confusion on the wording of this question, so the following answer is also considered correct.

```
fit_full = lm(Apps ~., data = College)
fit2 = regsubsets(Apps ~., data = College, nvmax = 17)
betahat = fit_full$coefficients
diff_beta = rep(NA, 18)
coefi = matrix(0, nrow = 1, ncol = 18, dimnames = list(NULL, names(betahat)))
for(i in 1:17){
  coef_temp = coef(fit1, id=i)
  coefi[, names(coef_temp)] = coef_temp
  diff_beta[i] = sqrt(sum((betahat - coefi) ^ 2))
}
plot(diff_beta, xlab = 'r', type = 'b', pch = 20)
```



(g) Now use forward and backward stepwise selection with the BIC and AIC to select models (so you will get up to four best models). How do the results compare to what you obtained above in part (c) and (d)?

```
fit_null = lm(Apps ~ 1, data = College)
backward_aic = step(fit_full, direction = 'backward', k = 2, trace = 0)
backward_bic = step(fit_full, direction = 'backward', k = log(777), trace = 0)
forward_aic  = step(fit_null, direction = 'forward', k = 2,
                    scope = list(lower = fit_null, upper =  fit_full), trace = 0)
forward_bic  = step(fit_null, direction = 'forward', k = log(777),
                    scope = list(lower = fit_null, upper =  fit_full), trace = 0)

# variables selected by forward AIC
forward_aic$coefficients
```

```
##   (Intercept)        Accept     Top10perc        Expend      Outstate
## -157.28685883    1.58691470   50.41131660    0.07246655   -0.09017643
##        Enroll    Room.Board     Top25perc    PrivateYes           PhD
##   -0.88265385    0.14776586  -14.74735373 -511.78760196  -10.70502848
##     Grad.Rate   F.Undergrad   P.Undergrad
##    8.63961002    0.05945481    0.04593068
```

```
# variables selected by backward AIC
backward_aic$coefficients
```

```
##   (Intercept)    PrivateYes        Accept        Enroll     Top10perc
## -157.28685883 -511.78760196    1.58691470   -0.88265385   50.41131660
##     Top25perc   F.Undergrad   P.Undergrad      Outstate    Room.Board
##  -14.74735373    0.05945481    0.04593068   -0.09017643    0.14776586
##           PhD        Expend     Grad.Rate
##  -10.70502848    0.07246655    8.63961002
```

```
# variables selected by forward BIC
forward_bic$coefficients
```

```
##   (Intercept)        Accept     Top10perc        Expend      Outstate
## -100.51668243    1.58421887   49.13908916    0.07273776   -0.09466457
##        Enroll    Room.Board     Top25perc    PrivateYes           PhD
##   -0.56220848    0.16373674  -13.86531103 -575.07060789  -10.01608705
##     Grad.Rate
##    7.33268904
```

```
# variables selected by backward BIC
backward_bic$coefficients
```

```
##   (Intercept)    PrivateYes        Accept        Enroll     Top10perc
## -100.51668243 -575.07060789    1.58421887   -0.56220848   49.13908916
##     Top25perc      Outstate    Room.Board           PhD        Expend
##  -13.86531103   -0.09466457    0.16373674  -10.01608705    0.07273776
##     Grad.Rate
##    7.33268904
```