Consider the `Cars93` dataframe from `library(MASS)`. It is of interest to predict the city mileage of a car based on the following predictors

- x_1: number of cylinders
- x_2: engine size
- x_3: horse power
- x_4: RPM
- x_5: number of passengers
- x_6: weight

Find the best regression models as follows

1. Use `regsubsets()` from library `leaps` to find the best set of predictors suggested by adjusted-R^2. Use a model with these set of predictors to predict the city mileage of a 4-cylinder car with 2.3 engine size, 5500 RPM, 2950 pounds, 4 passengers, and 200 horse power.
2. Use `set.seed(12)` to divide the data set into a training and a test set (50%). Compare the MSPE of the full model and the model with the (adjusted-R^2) best predictors
3. Find the best set of predictors suggested by MSPE.
4. Find the best set of predictors suggested by AIC. Compare its MSPE with that of the best adjusted-R^2 model
5. Find the best set of predictors suggested by BIC. Compare its MSPE with that of other models.

```
# cars93b.r

library(PASWR2)   # checking.plots()
library(MASS)     # Cars93()
d0 = Cars93

# create data set
d1 = Cars93[,c(7,11,12,13,14,18,25)]    # or
d1 = subset(d0,select=c(MPG.city, Cylinders, EngineSize, Horsepower, RPM, Passengers, Weight))
d1$Cylinders = as.numeric(d1$Cylinders)

# Best set of predictors
#==================================================================
library(leaps)    # regsubsets()

# Select predictors
models=regsubsets(MPG.city~.,d1,nvmax=12)
summary(models)
# Selection Algorithm: exhaustive
#          Cylinders EngineSize Horsepower RPM Passengers Weight
# 1  ( 1 ) " "       " "        " "        " " " "        "*"
# 2  ( 1 ) "*"       " "        " "        " " " "        "*"
# 3  ( 1 ) "*"       "*"        " "        " " " "        "*"
# 4  ( 1 ) " "       "*"        "*"        "*" " "        "*"
# 5  ( 1 ) "*"       "*"        "*"        "*" " "        "*"
# 6  ( 1 ) "*"       "*"        "*"        "*" "*"        "*"

# best predictor is Weight
# worst predictor is n. of Passengers

summary(models)$adjr2
# [1] 0.7077055 0.7133132 0.7123930 0.7166129 0.7157038 0.7126693
a=summary(models)$adjr2
which.max(a)   #  4

# best model is in row 4
# best model includes   EngineSize, Horsepower, RPM, Weight
# these variables are highly correlated with MPG.city

# prediction with model 4
#==================================================================
m0 = lm(MPG.city~EngineSize+Horsepower+RPM+Weight,d1)
newval=data.frame(Cylinders=4,EngineSize=2.3,Horsepower=200,RPM=5500,Passengers=4,Weight=2950)
predict(m0,newval)
# 21.06858
```

```
# function predict.regsubsets() p249

predict.regsubsets <- function(object, newdata, id, ...)
{
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi = coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars]%*%coefi
}

newval=data.frame(MPG.city=100,Cylinders=4,EngineSize=2.3,Horsepower=200,RPM=5500,Passengers=4,Weight
predict.regsubsets(models,newval,id = 4)

#          [,1]
# [1,] 21.06858

# predict.regsubsets() identifies model with id=4
# it requires response MPG.city (with any value) in newval dataframe
# predict() function does not

# Test and training sets
#===================================================================

set.seed(12)
n = nrow(d1)    # 93
train = sample(1:n,47)       # train row numbers

d1train = d1[train,]
d1test  = d1[-train,]
dim(d1train)     # [1] 47  7
dim(d1test)      # [1] 46  7

# full model
m1 = lm(MPG.city~.,d1train)
yhat1 = predict(m1,d1test)
y = d1test$MPG.city
# mspe
mean((yhat1-y)^2)       # 9.736857

# best adjR2 model
m2 = lm(MPG.city~EngineSize+Horsepower+RPM+Weight,d1train)
yhat2  = predict(m2,d1test)
# mspe
mean((yhat2-y)^2)       # 9.389211
```

```
# If summary(m1) is compared against summary(m2)
# comparisons are for training performance

# regsubsets() with train set
#===================================================================

models=regsubsets(MPG.city~.,d1train,nvmax=12)
summary(models)

mspe = rep(0, 6)
for(i in 1:6)
{
  yhat = predict.regsubsets(models, d1test, id = i)
  mspe[i] = mean((y - yhat)^2)
}

mspe
# 9.122425 9.695262 9.819361 9.389211 9.608725 9.736857

# best model is model 1
# second best model is model 4

# 9) stepAIC
#=================================================================

step1 = stepAIC(m1)
coef(step1)
#  (Intercept)   Horsepower          RPM       Weight
# 35.280443949 -0.026385811  0.001349840 -0.005293273

# AIC model
m3 = lm(MPG.city~Horsepower+RPM+Weight,d1train)
yhat3  = predict(m3,d1test)
# mspe
mean((yhat3-y)^2)       # 9.602604

# adjR2 suggested an MSPE-better model than AIC
```

```
# BIC
n2 = nrow(d1train)
step2 = stepAIC(m1,k=log(n2))
coef(step2)
# (Intercept)      Weight
# 45.62537480 -0.00760907

m4 = lm(MPG.city~Weight,d1train)
yhat4  = predict(m4,d1test)
# mspe
mean((yhat4-y)^2)
# 9.122425

# agrees with regsubsets() best MSPE model
```