NBA 4920/6921 Lecture 12

Linear Model Selection In-class Exercise

Murat Unal

10/07/2021

```r
rm(list=ls())
options(digits = 3, scipen = 999)
library(tidyverse)
library(ISLR)
library(cowplot)
library(ggcorrplot)
library(stargazer)
library(corrr)
library(lmtest)
library(sandwich)
library(MASS)
library(car)
library(jtools)
library(caret)
library(leaps)
library(future.apply)
set.seed(2)
```

Agenda

- Mid-term Survey
- In-class Exercise

Mid-term Survey

- Go to Canvas -> Quizzes
- Complete Mid-term survey
- Completely anonymized
- Get the regular 1 quiz point for completing the survey
- I appreciate your feedback

In-class exercise

- Go to Canvas -> Assignments -> In-class Exercise
- Download the train and test `Boston` data sets
- Your task is to build regression models to predict `crim` - crime rate in the neighborhoods in Boston.

- Build 3 regression models using `best subset selection`, `forward stepwise selection` and `backward stepwise selection`.
- Validate these models using two approaches for each `validation set` and `K-fold cross validation`
- Decide on your final model and make predictions on unseen `test` cases.

- Create a final plot that shows the test error estimates for each model and each validation approach.
- Report your MSE on the test data
- Submission achieves you an additional 5% on your final exam score
- The top 3 MSEs will get an additional 5%, 4% and 3%, respectively
- Submission open until tomorrow (10.08) 11:59PM

```
data_test <- read.csv("boston_test.csv")
data_train <- read.csv("boston_train.csv")
dim(data_train)
```

```
[1] 405  14
```

```
names(data_train)
```

```
 [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"
 [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lst
```

Best subset selection

```
# Draw validation set
validation_data = data_train %>% sample_frac(size = 0.3)
# Create the remaining training set
training_data = setdiff(data_train, validation_data)
```

```
nvars = 13
regfit.best=regsubsets(crim ~ .,data=training_data,
                                              nvmax=nvars)
```

Validation set approach

```
validation.mat=model.matrix(crim~.,
                            data=validation_data)

val.errors = numeric(nvars)
for(each in 1:nvars){
    coefi = coef(regfit.best,id=each)
    pred = validation.mat[,names(coefi)]%*%coefi
    val.errors[each]=
      mean((validation_data$crim-pred)^2)
}

best.subset.val.model <- which.min(val.errors)
best.subset.val.model
```

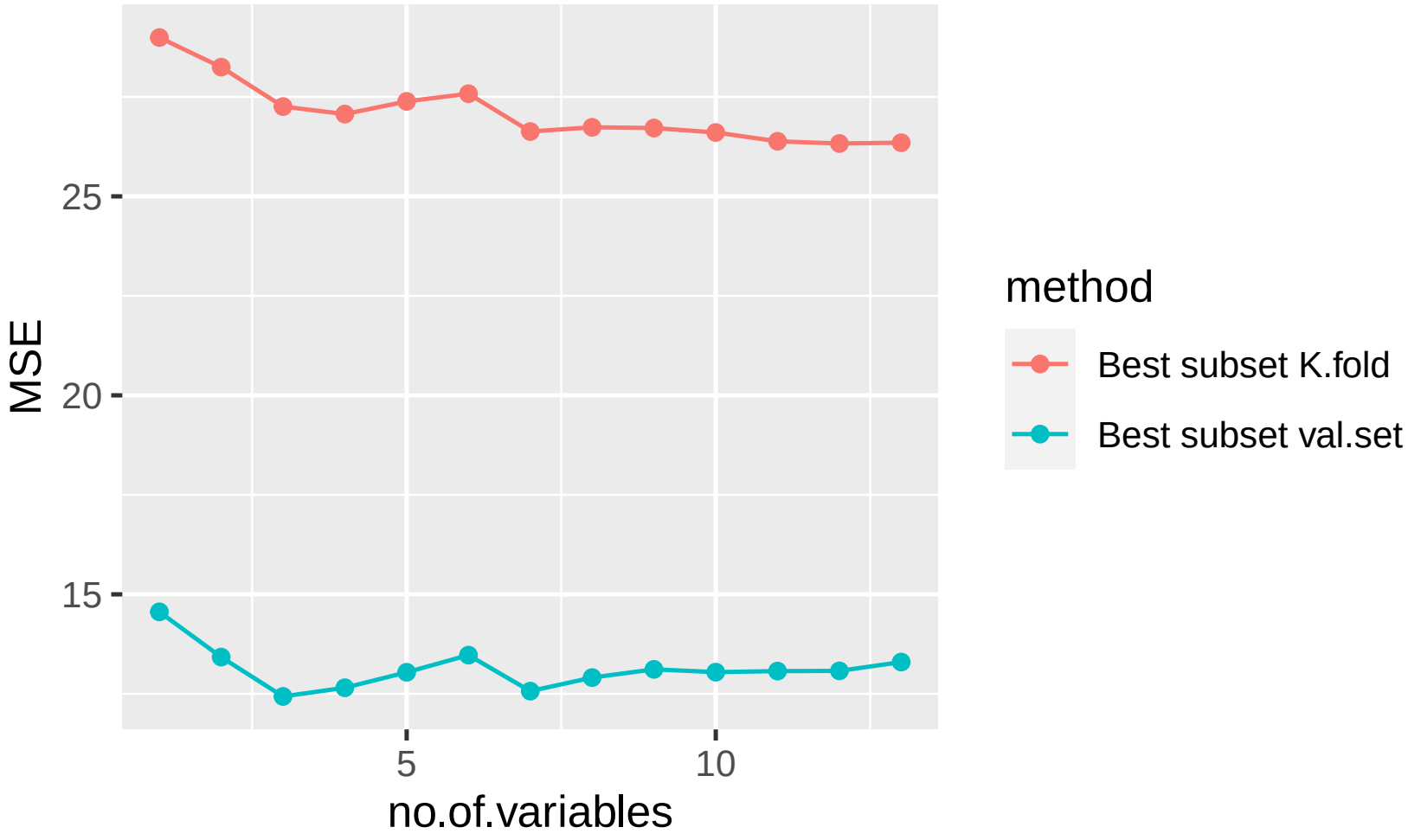
```
[1] 3
```

K-fold cross validation

```
nvars = 13
nfold = 10
# Create folds
fold.list <- createFolds(rownames(data_train),nfold)
# Empty vector to store the resulting MSEs
cv.errors =matrix(0,nfold,nvars,
                  dimnames =list(NULL,paste (1:nvars)))

for(each in 1:nfold){
 train <- data_train[-fold.list[[each]],]
 validate <- data_train[fold.list[[each]],]

 best.fit=regsubsets(crim~.,data=train,nvmax =19)
 validation.mat=model.matrix(crim~.,data=validate)
 for(i in 1:nvars){
   coefi = coef(best.fit,id=i)
   pred = validation.mat[,names(coefi)]%*%coefi
   cv.errors[each,i] = mean( (validate$crim-pred)^2)
   }
```

```
mean.cv.errors=apply(cv.errors ,2, mean)
best.subset.cv.model <- which.min(mean.cv.errors)
best.subset.cv.model
```

```
12
12
```


MSE
25
20
15
5
10
no.of.variables
method
Best subset K.fold
Best subset val.set

Predictions on test data

To obtain the final model we perform best subset selection on the full data set and obtain the 3-variable model and the 12-variable model.

```
test.mat <- model.matrix(crim~.,data=data_test)
best.fit=regsubsets(crim~.,data=data_train,nvmax =13)

val.coef <- coef(best.fit,best.subset.val.model)
pred = test.mat[,names(val.coef)]%*%val.coef
best.subset.val.mse = mean((data_test$crim-pred)^2)

cv.coef <- coef(best.fit,best.subset.cv.model)
pred = test.mat[,names(cv.coef)]%*%cv.coef
best.subset.cv.mse = mean((data_test$crim-pred)^2)

test.mse.data <- numeric(6)
test.mse.data[1] <- best.subset.val.mse
test.mse.data[2] <- best.subset.cv.mse
```

Forward Stepwise Selection

Validation set approach

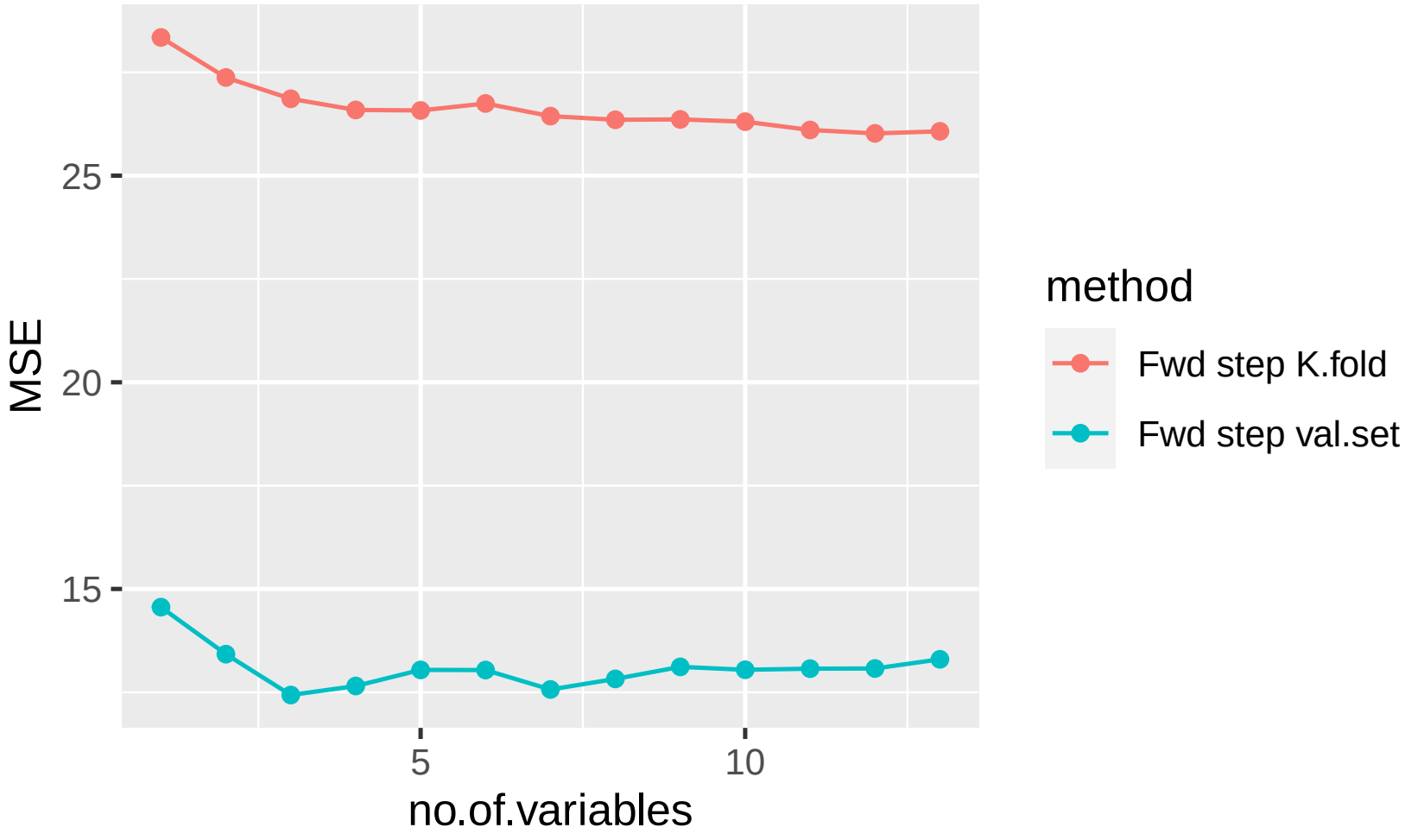
```
nvars=13
regfit.fwd=regsubsets(crim~.,data=training_data,
                      nvmax=nvars,method="forward")

validation.mat=model.matrix(crim~.,
                      data=validation_data)

fwd.val.errors = numeric(nvars)
for(each in 1:nvars){
    coefi = coef(regfit.fwd,id=each)
    pred = validation.mat[,names(coefi)]%*%coefi
    fwd.val.errors[each]=
      mean((validation_data$crim-pred)^2)
}

fwd.val.model <- which.min(fwd.val.errors)
```

K-fold cross validation

```
nvars = 13
nfold = 10
# Create folds
fold.list <- createFolds(rownames(data_train),nfold)
# Empty vector to store the resulting MSEs
fwd.cv.errors =matrix(0,nfold,nvars,
                  dimnames =list(NULL,paste (1:nvars)))

for(each in 1:nfold){
 train <- data_train[-fold.list[[each]],]
 validate <- data_train[fold.list[[each]],]

 best.fit=regsubsets(crim~.,data=train,nvmax =13,
                     method = "forward")
 validation.mat=model.matrix(crim~.,data=validate)
 for(i in 1:nvars){
   coefi = coef(best.fit,id=i)
   pred = validation.mat[,names(coefi)]%*%coefi
   fwd.cv.errors[each,i] = mean( (validate$crim-pred)^2)
```

```
mean.fwd.cv.errors=apply(fwd.cv.errors ,2, mean)
fwd.cv.model <- which.min(mean.fwd.cv.errors)
fwd.cv.model
```

```
12
12
```


MSE
25
20
15
5
10
no.of.variables
method
Fwd step K.fold
Fwd step val.set

Predictions on test data

```
test.mat <- model.matrix(crim~.,data=data_test)
fwd.fit=regsubsets(crim~.,data=data_train,nvmax =13,
                   method = "forward")

fwd.val.coef <- coef(fwd.fit,fwd.val.model)
pred = test.mat[,names(fwd.val.coef)]%*%fwd.val.coef
fwd.val.mse = mean((data_test$crim-pred)^2)

fwd.cv.coef <- coef(fwd.fit,fwd.cv.model)
pred = test.mat[,names(fwd.cv.coef)]%*%fwd.cv.coef
fwd.cv.mse = mean((data_test$crim-pred)^2)

test.mse.data[3] <- fwd.val.mse
test.mse.data[4] <- fwd.cv.mse
```

Backward Stepwise Selection

Validation set approach

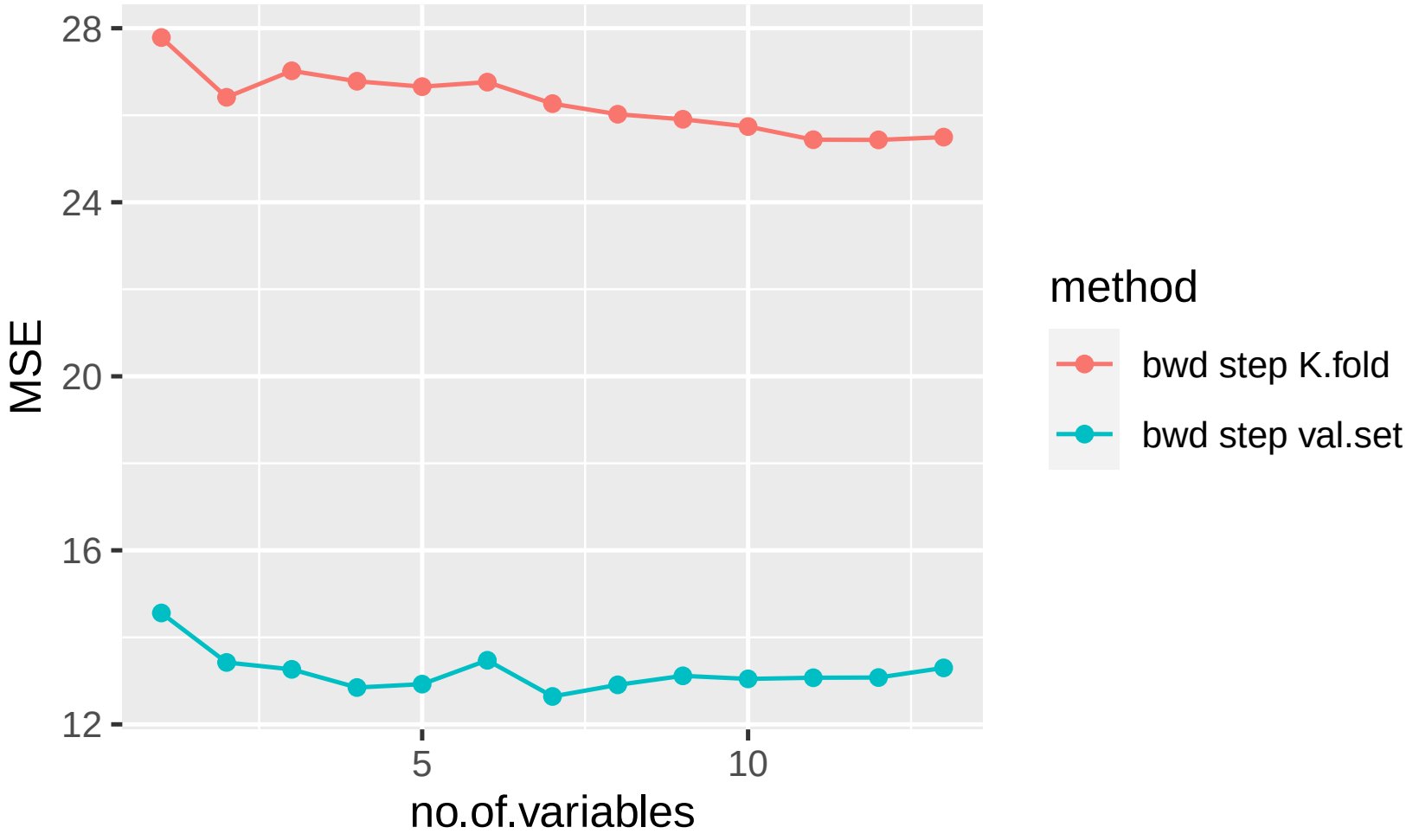
```
nvars=13
regfit.bwd=regsubsets(crim~.,data=training_data,
                      nvmax=nvars,method="backward")

validation.mat=model.matrix(crim~.,
                      data=validation_data)

bwd.val.errors = numeric(nvars)
for(each in 1:nvars){
    coefi = coef(regfit.bwd,id=each)
    pred = validation.mat[,names(coefi)]%*%coefi
    bwd.val.errors[each]=
      mean((validation_data$crim-pred)^2)
}

bwd.val.model <- which.min(bwd.val.errors)
```

K-fold cross validation

```
nvars = 13
nfold = 10
# Create folds
fold.list <- createFolds(rownames(data_train),nfold)
# Empty vector to store the resulting MSEs
bwd.cv.errors =matrix(0,nfold,nvars,
                  dimnames =list(NULL,paste (1:nvars)))

for(each in 1:nfold){
 train <- data_train[-fold.list[[each]],]
 validate <- data_train[fold.list[[each]],]

 best.fit=regsubsets(crim~.,data=train,nvmax =13,
                     method = "backward")
 validation.mat=model.matrix(crim~.,data=validate)
 for(i in 1:nvars){
   coefi = coef(best.fit,id=i)
   pred = validation.mat[,names(coefi)]%*%coefi
   bwd.cv.errors[each,i] = mean( (validate$crim-pred)^2)
```

```
mean.bwd.cv.errors=apply(bwd.cv.errors ,2, mean)
bwd.cv.model <- which.min(mean.bwd.cv.errors)
bwd.cv.model
```

```
12
12
```


28
24
20
16
12
MSE
5
10
no.of.variables
method
bwd step K.fold
bwd step val.set

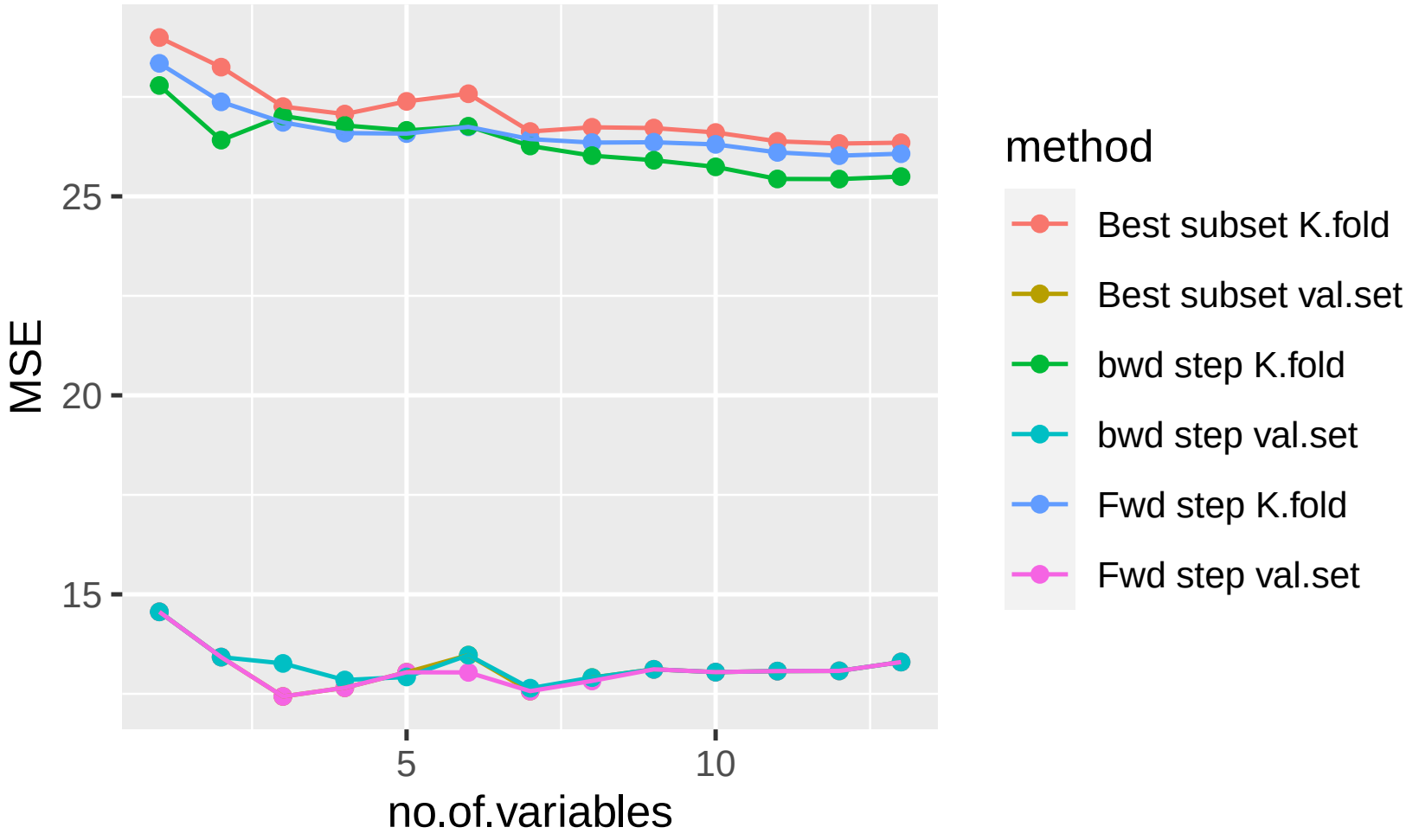
Predictions on test data

```
test.mat <- model.matrix(crim~.,data=data_test)
bwd.fit=regsubsets(crim~.,data=data_train,nvmax =13,
                   method = "backward")

bwd.val.coef <- coef(bwd.fit,bwd.val.model)
pred = test.mat[,names(bwd.val.coef)]%*%bwd.val.coef
bwd.val.mse = mean((data_test$crim-pred)^2)

bwd.cv.coef <- coef(bwd.fit,bwd.cv.model)
pred = test.mat[,names(bwd.cv.coef)]%*%bwd.cv.coef
bwd.cv.mse = mean((data_test$crim-pred)^2)

test.mse.data[5] <- bwd.val.mse
test.mse.data[6] <- bwd.cv.mse
```

Let's compare the test error estimates from all approaches

```
test.mse.data
```

```
[1] 112 110 112 110 109 110
```


MSE
25
20
15
5
10
no.of.variables
method
Best subset K.fold
Best subset val.set
bwd step K.fold
bwd step val.set
Fwd step K.fold
Fwd step val.set