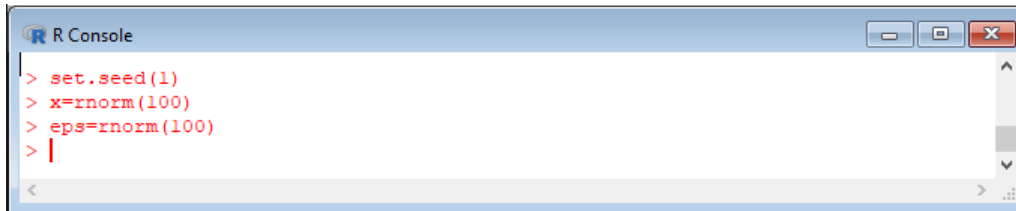


Звіт  
до індивідуального завдання №5  
з предмету Моделі статистичного навчання

Роботу виконала:  
**Мерцало Ірина Ігорівна,**  
студентка групи ПМІМ-11

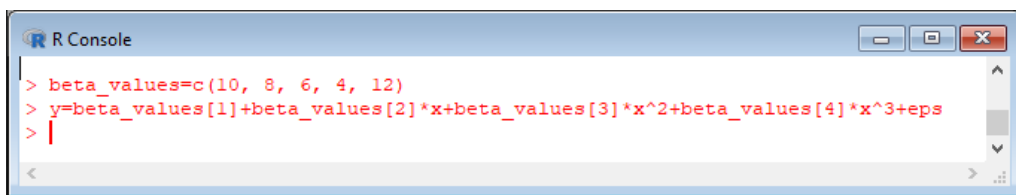
## Завдання 1

1.1 Використовуючи функцію `rnorm()` згенерувала предиктор  $X$  довжиною  $n = 100$ , та вектор залишків  $\varepsilon$  такої ж довжини  $n = 100$ :



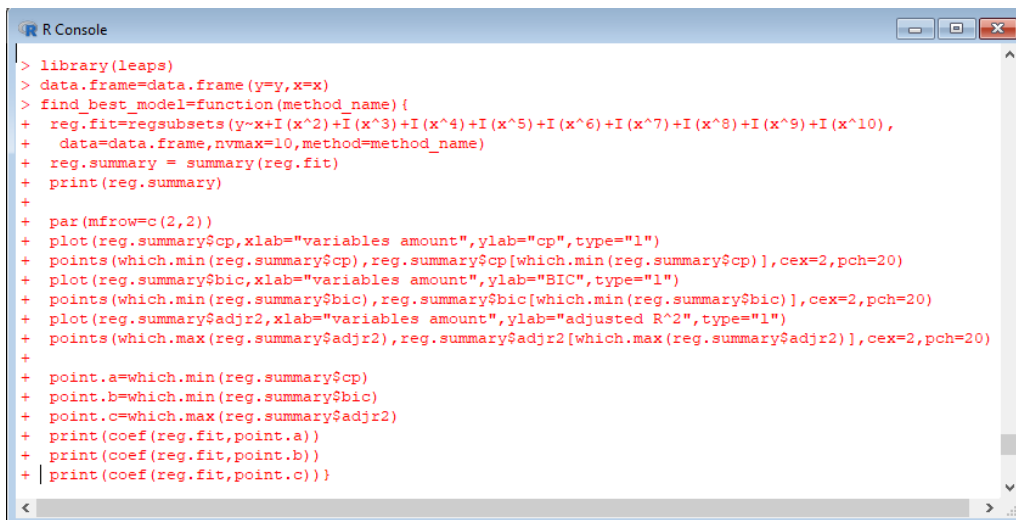
```
> set.seed(1)
> x=rnorm(100)
> eps=rnorm(100)
> |
```

1.2 Згенерувала вектор залежних змінних  $Y$  довжини  $n = 100$  відповідно до моделі  $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$ , де константи  $\beta_0, \beta_1, \beta_2$  і  $\beta_3$  дорівнюють 10, 8, 6, 4, 12 відповідно:



```
> beta_values=c(10, 8, 6, 4, 12)
> y=beta_values[1]+beta_values[2]*x+beta_values[3]*x^2+beta_values[4]*x^3+eps
> |
```

1.3 Використовуючи функцію `regsubsets()` вибрала найкращу модель методом вибору найкращої підмножини з множини предикторів  $X, X^2, \dots, X^{10}$ .



```
> library(leaps)
> data.frame=data.frame(y=y,x=x)
> find_best_model=function(method_name){
+   reg.fit=regsubsets(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10),
+   +   data=data.frame,nvmax=10,method=method_name)
+   reg.summary = summary(reg.fit)
+   print(reg.summary)
+ }
+ par(mfrow=c(2,2))
+ plot(reg.summary$cp,xlab="variables amount",ylab="cp",type="l")
+ points(which.min(reg.summary$cp),reg.summary$cp[which.min(reg.summary$cp)],cex=2,pch=20)
+ plot(reg.summary$bic,xlab="variables amount",ylab="BIC",type="l")
+ points(which.min(reg.summary$bic),reg.summary$bic[which.min(reg.summary$bic)],cex=2,pch=20)
+ plot(reg.summary$adjr2,xlab="variables amount",ylab="adjusted R^2",type="l")
+ points(which.max(reg.summary$adjr2),reg.summary$adjr2[which.max(reg.summary$adjr2)],cex=2,pch=20)
+ point.a=which.min(reg.summary$cp)
+ point.b=which.min(reg.summary$bic)
+ point.c=which.max(reg.summary$adjr2)
+ print(coef(reg.fit,point.a))
+ print(coef(reg.fit,point.b))
+ print(coef(reg.fit,point.c)) }
```

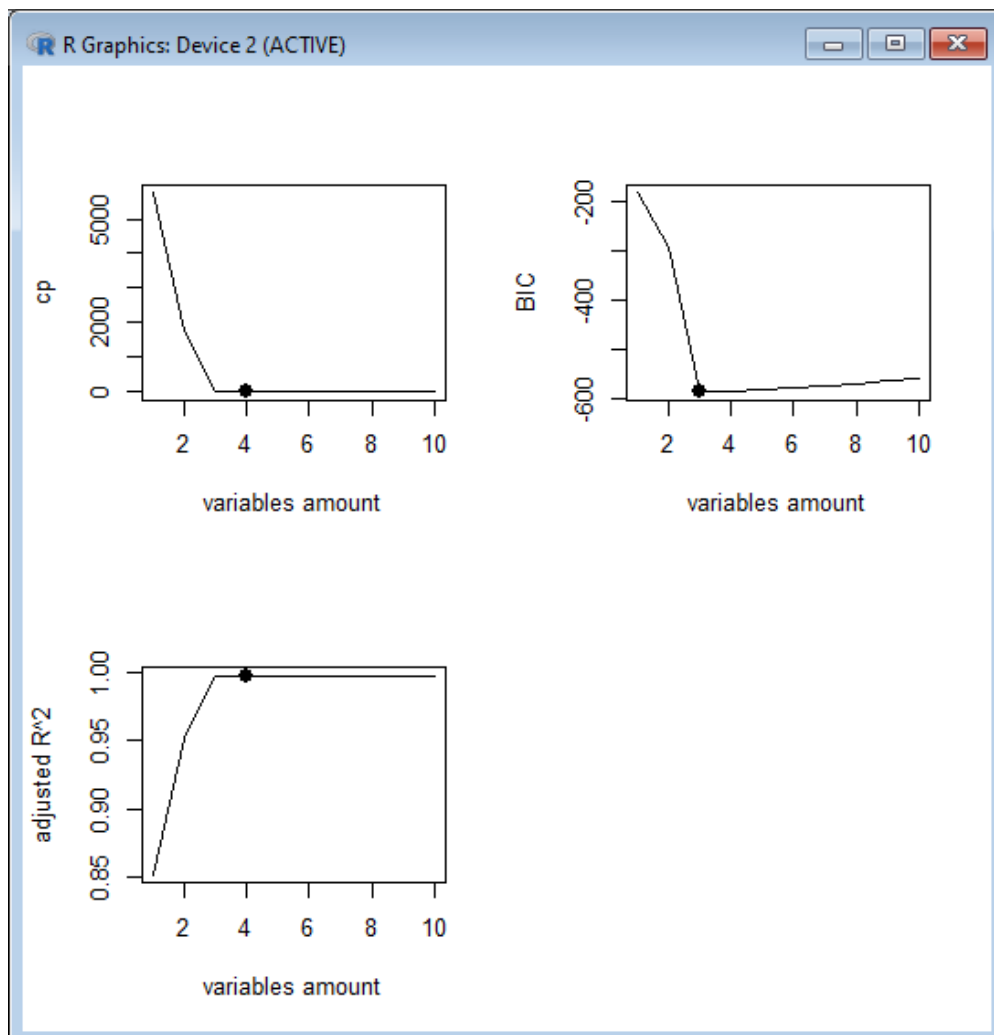
```

R Console
> find_best_model("exhaustive")
Subset selection object
Call: find_best_model("exhaustive")
10 Variables (and intercept)
Forced in Forced out
x          FALSE      FALSE
I(x^2)      FALSE      FALSE
I(x^3)      FALSE      FALSE
I(x^4)      FALSE      FALSE
I(x^5)      FALSE      FALSE
I(x^6)      FALSE      FALSE
I(x^7)      FALSE      FALSE
I(x^8)      FALSE      FALSE
I(x^9)      FALSE      FALSE
I(x^10)     FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: exhaustive
x  I(x^2) I(x^3) I(x^4) I(x^5) I(x^6) I(x^7) I(x^8) I(x^9) I(x^10)
1 ( 1 ) " " " " " " " " " " " "
2 ( 1 ) " " " " " " " " " " " "
3 ( 1 ) " " " " " " " " " " " "
4 ( 1 ) " " " " " " " " " " " "
5 ( 1 ) " " " " " " " " " " " "
6 ( 1 ) " " " " " " " " " " " "
7 ( 1 ) " " " " " " " " " " " "
8 ( 1 ) " " " " " " " " " " " "
9 ( 1 ) " " " " " " " " " " " "
10 ( 1 ) " " " " " " " " " " " "
(Intercept) x I(x^2) I(x^3) I(x^5)
10.07200775 8.38745596 5.84575641 3.55797426 0.08072292
(Intercept) x I(x^2) I(x^3)
10.061507 7.975280 5.876209 4.017639
(Intercept) x I(x^2) I(x^3) I(x^5)
10.07200775 8.38745596 5.84575641 3.55797426 0.08072292

```

Можна побачити, що зірочками позначені змінні, які формують найкращу модель за показниками  $C_p$ ,  $BIC$  і скорегований  $R^2$  для кожної розмірності.

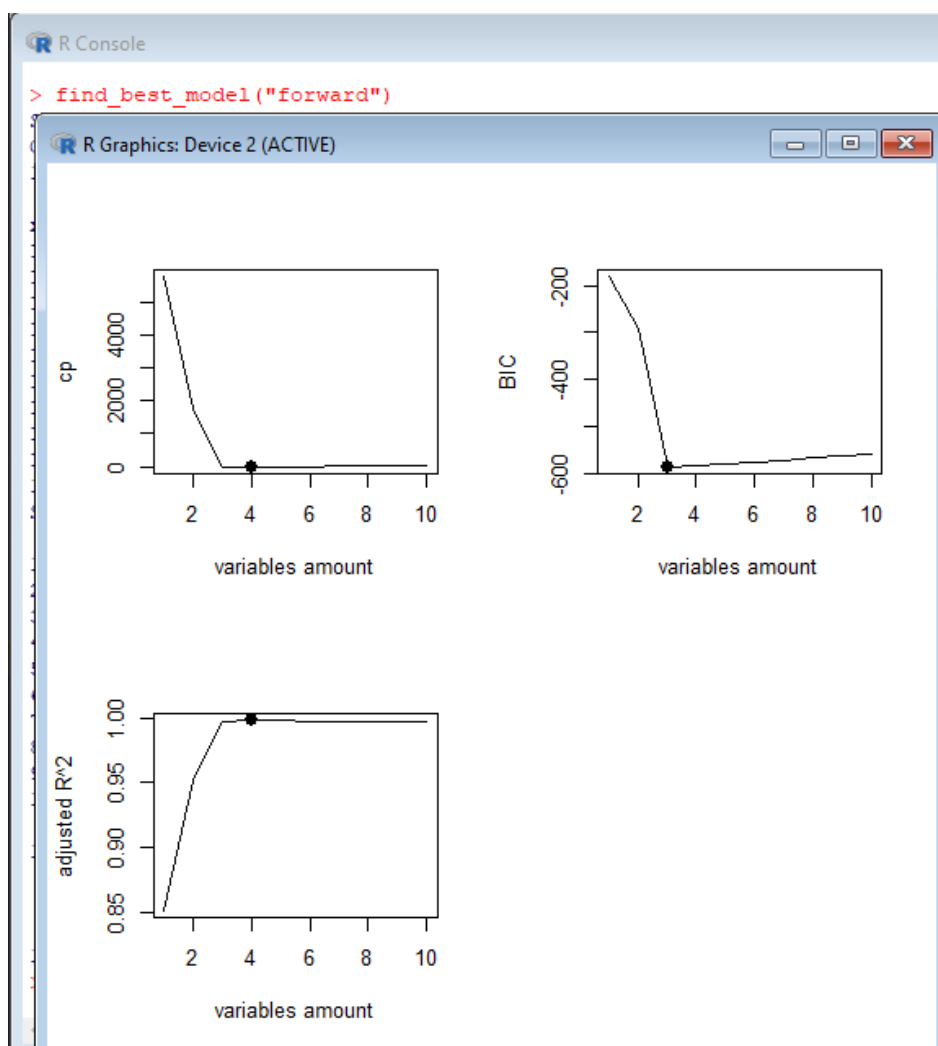
Навела декілька графіків на підтвердження своєї відповіді:



За показниками  $C_p$  і скорегований  $R^2$  можна побачити, що найкраща модель - з кількістю змінних 4, тобто  $x, x^2, x^3, x^5$ , а за показником  $BIC$  – з кількістю змінних 3, тобто  $x, x^2, x^3$ .

1.4 Повторила 1.3, використовуючи методи покрокового вибору вперед та назад:

```
R Console
> find_best_model("forward")
Subset selection object
Call: find_best_model("forward")
10 Variables (and intercept)
Forced in Forced out
x FALSE FALSE
I(x^2) FALSE FALSE
I(x^3) FALSE FALSE
I(x^4) FALSE FALSE
I(x^5) FALSE FALSE
I(x^6) FALSE FALSE
I(x^7) FALSE FALSE
I(x^8) FALSE FALSE
I(x^9) FALSE FALSE
I(x^10) FALSE FALSE
1 subsets of each size up to 10
Selection Algorithm: forward
  x I(x^2) I(x^3) I(x^4) I(x^5) I(x^6) I(x^7) I(x^8) I(x^9) I(x^10)
1 ( 1 ) " " " " " " " " " " "
2 ( 1 ) " " " " " " " " " " "
3 ( 1 ) " " " " " " " " " " "
4 ( 1 ) " " " " " " " " " " "
5 ( 1 ) " " " " " " " " " " "
6 ( 1 ) " " " " " " " " " " "
7 ( 1 ) " " " " " " " " " " "
8 ( 1 ) " " " " " " " " " " "
9 ( 1 ) " " " " " " " " " " "
10 ( 1 ) " " " " " " " " " " "
(Intercept) x I(x^2) I(x^3) I(x^5)
10.07200775 8.38745596 5.84575641 3.55797426 0.08072292
(Intercept) x I(x^2) I(x^3)
10.061507 7.975280 5.876209 4.017639
(Intercept) x I(x^2) I(x^3) I(x^5)
10.07200775 8.38745596 5.84575641 3.55797426 0.08072292
```

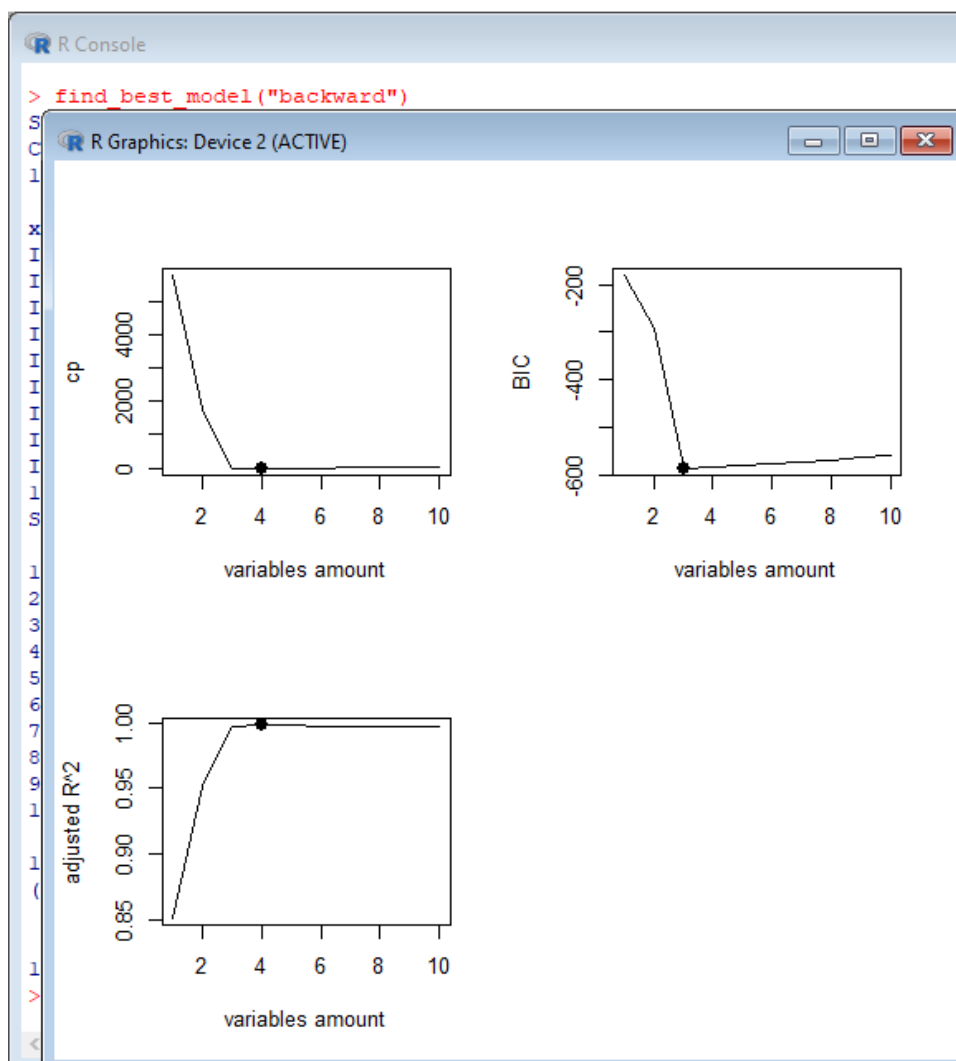


З допомогою методу покрокового вибору вперед можна побачити, що за показниками  $C_p$  і скорегований  $R^2$  найкраща модель - з кількістю змінних 4, тобто  $x, x^2, x^3, x^5$ , а за показником  $BIC$  - з кількістю змінних 3, тобто  $x, x^2, x^3$ .

```

R Console
> find_best_model("backward")
Subset selection object
Call: find_best_model("backward")
10 Variables (and intercept)
Forced in Forced out
x FALSE FALSE
I(x^2) FALSE FALSE
I(x^3) FALSE FALSE
I(x^4) FALSE FALSE
I(x^5) FALSE FALSE
I(x^6) FALSE FALSE
I(x^7) FALSE FALSE
I(x^8) FALSE FALSE
I(x^9) FALSE FALSE
I(x^10) FALSE FALSE
1 subsets of each size up to 10
Selection Algorithm: backward
x I(x^2) I(x^3) I(x^4) I(x^5) I(x^6) I(x^7) I(x^8) I(x^9) I(x^10)
1 ( 1 ) " " " " " " " " " " " "
2 ( 1 ) " " " " " " " " " " " "
3 ( 1 ) " " " " " " " " " " " "
4 ( 1 ) " " " " " " " " " " " "
5 ( 1 ) " " " " " " " " " " " "
6 ( 1 ) " " " " " " " " " " " "
7 ( 1 ) " " " " " " " " " " " "
8 ( 1 ) " " " " " " " " " " " "
9 ( 1 ) " " " " " " " " " " " "
10 ( 1 ) " " " " " " " " " " " "
(Intercept) x I(x^2) I(x^3) I(x^9)
10.079236362 8.231905828 5.833494180 3.819555807 0.001290827
(Intercept) x I(x^2) I(x^3)
10.061507 7.975280 5.876209 4.017639
(Intercept) x I(x^2) I(x^3) I(x^9)
10.079236362 8.231905828 5.833494180 3.819555807 0.001290827
>

```

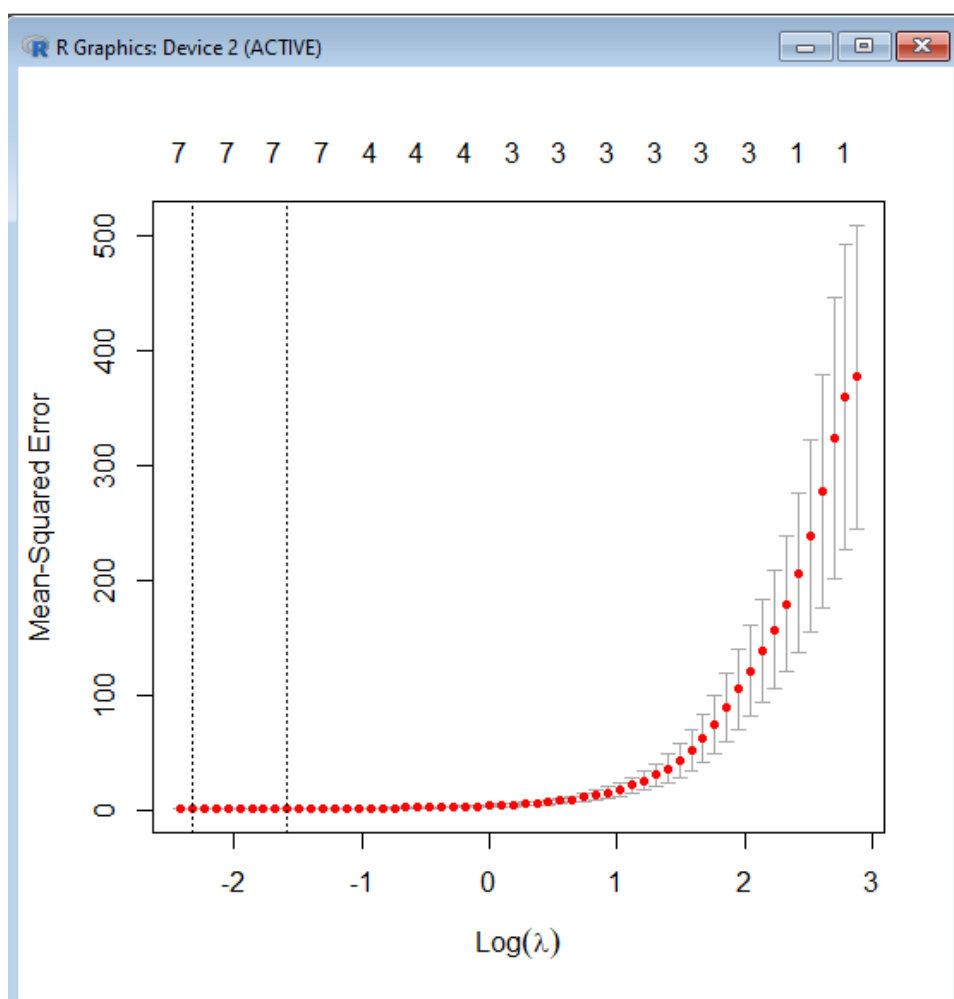


З допомогою методу покрокового вибору назад можна побачити, що за показниками  $C_p$  і скорегований  $R^2$  найкраща модель - з кількістю змінних 4, тобто  $x, x^2, x^3, x^5$ , а за показником  $BIC$  - з кількістю змінних 3, тобто  $x, x^2, x^3$ .

1.5 Пристосувала ласо модель до згенерованих даних, використовуючи  $X, X^2, \dots, X^{10}$  як предиктори. Використала перехресну перевірку для вибору значення  $\lambda$ . Побудувала графіки помилки перехресної перевірки як функції від  $\lambda$ .

```
R Console
> library(glmnet)
> lasso_model=function() {
+   x_m=model.matrix(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9)+I(x^10),
+   data=data.frame)[,~1]
+   cv.out=cv.glmnet(x_m,y,alpha=1)
+   par(mfrow=c(1,1))
+   best_lambda=cv.out$lambda.min
+   print(paste('min lambda:',best_lambda))
+   plot(cv.out)
+   out=glmnet(x_m,y,alpha=1)
+   lasso.coefs=predict(out,s=best_lambda,type="coefficients")[1:11,]
+   print(lasso.coefs[lasso.coefs != 0])
+ }
> lasso_model()
[1] "min lambda: 0.0977150162044969"
      (Intercept)          x          I(x^2)          I(x^3)          I(x^4)          I(x^5)          I(x^7)          I(x^9)
1.020411e+01  8.059718e+00  5.592374e+00  3.876805e+00  4.636993e-02  3.169856e-04  3.653814e-03  7.082794e-05
```

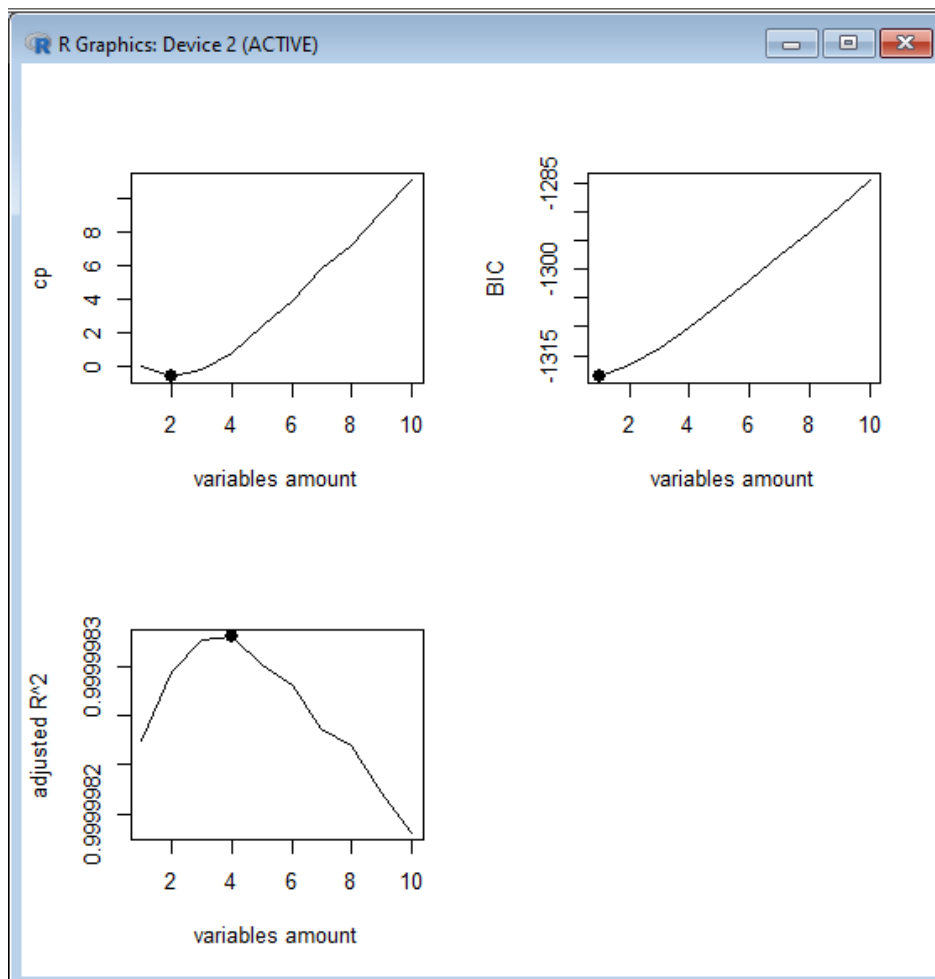
Значення  $\lambda$  з найменшою помилкою дорівнює 0,098. Найкраща модель - з кількістю змінних 7, тобто  $x, x^2, x^3, x^4, x^5, x^7, x^9$ .



Коли значення  $\lambda$  додатне, помилка зростає.

1.6 Згенерувала вектор залежних змінних  $Y$  відповідно до моделі  $Y = \beta_0 + \beta_7 X^7 + \varepsilon$ , і застосувала метод найкращого вибору підмножини і ласо:

```
R Console
> find_best_model("exhaustive")
Subset selection object
Call: find_best_model("exhaustive")
10 Variables (and intercept)
    Forced in Forced out
x          FALSE      FALSE
I(x^2)      FALSE      FALSE
I(x^3)      FALSE      FALSE
I(x^4)      FALSE      FALSE
I(x^5)      FALSE      FALSE
I(x^6)      FALSE      FALSE
I(x^7)      FALSE      FALSE
I(x^8)      FALSE      FALSE
I(x^9)      FALSE      FALSE
I(x^10)     FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: exhaustive
  x  I(x^2) I(x^3) I(x^4) I(x^5) I(x^6) I(x^7) I(x^8) I(x^9) I(x^10)
1 ( 1 )    "    "    "    "    "    "    "    "    "
2 ( 1 )    "    "    "    "    "    "    "    "    "
3 ( 1 )    "    "    "    "    "    "    "    "    "
4 ( 1 )    "    "    "    "    "    "    "    "    "
5 ( 1 )    "    "    "    "    "    "    "    "    "
6 ( 1 )    "    "    "    "    "    "    "    "    "
7 ( 1 )    "    "    "    "    "    "    "    "    "
8 ( 1 )    "    "    "    "    "    "    "    "    "
9 ( 1 )    "    "    "    "    "    "    "    "    "
10 ( 1 )   "    "    "    "    "    "    "    "    "
(Intercept)      I(x^2)      I(x^7)
10.0704904 -0.1417084 12.0015552
(Intercept)      I(x^7)
9.95894 12.00077
(Intercept)      x      I(x^2)      I(x^3)      I(x^7)
10.0762524 0.2914016 -0.1617671 -0.2526527 12.0091338
```

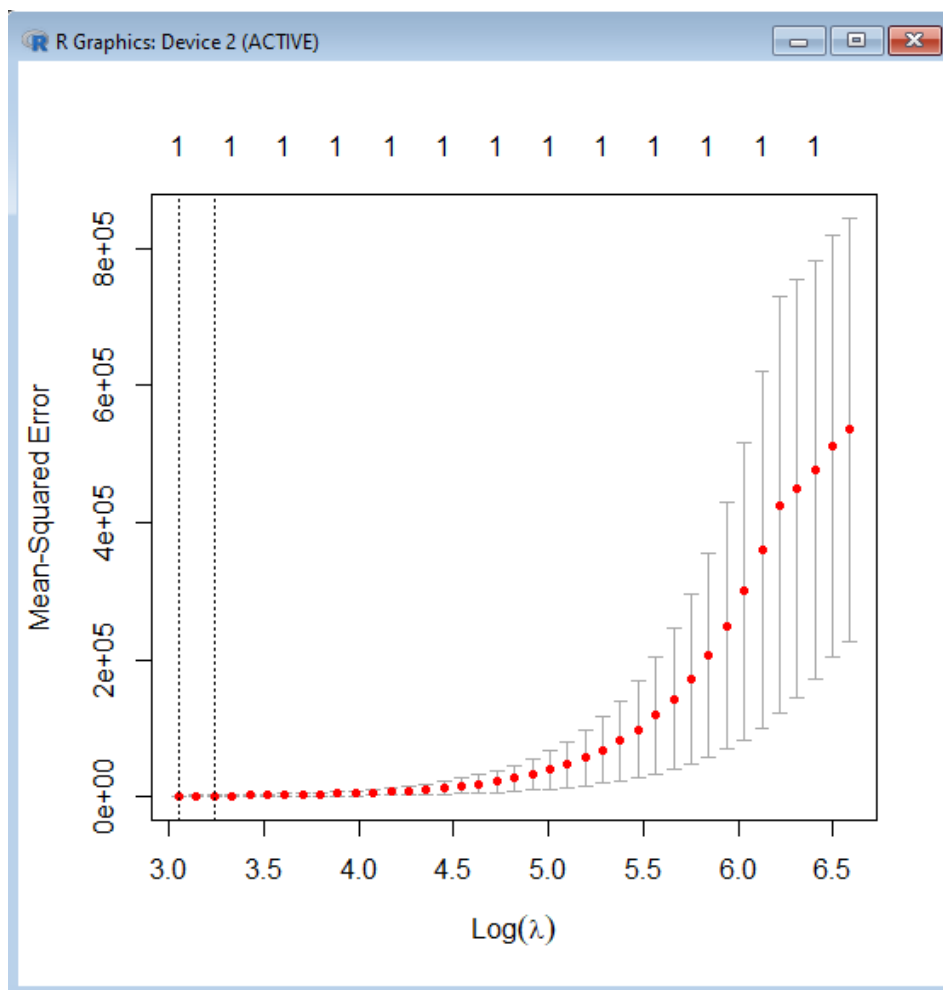


З допомогою методу покрокового вибору можна побачити, що за показником  $C_p$  найкраща модель - з кількістю змінних 2, тобто  $x^2$ ,  $x^7$ , за показником  $BIC$  - з кількістю змінних 1, тобто  $x^7$ , за показником скорегований  $R^2$  - з кількістю змінних 4, тобто  $x$ ,  $x^2$ ,  $x^3$ ,  $x^7$ .

```

R Console
> lasso_model()
[1] "min lambda: 21.2027490613505"
(Intercept)      I(x^7)
    11.43534      11.65094
>

```



Значення  $\lambda$  з найменшою помилкою дорівнює 11,651. Найкраща модель - з кількістю змінних 1, тобто  $x^7$ .

**Завдання 2.** На основі даних College передбачити кількість отриманих заяв.



R Console

```
> library(ISLR)
> fix(College)
```

Data Editor

	row.names	Private	Apps	Accept
1	Abilene Christian University	Yes	1660	1232
2	Adelphi University	Yes	2186	1924
3	Adrian College	Yes	1428	1097
4	Agnes Scott College	Yes	417	349
5	Alaska Pacific University	Yes	193	146
6	Albertson College	Yes	587	479
7	Albertus Magnus College	Yes	353	340
8	Albion College	Yes	1899	1720
9	Albright College	Yes	1038	839
10	Alderson-Broadus College	Yes	582	498
11	Alfred University	Yes	1732	1425
12	Allegheny College	Yes	2652	1900
13	Allentown Coll. of St. Francis de Sales	Yes	1179	780
14	Alma College	Yes	1267	1080
15	Alverno College	Yes	494	313
16	American International College	Yes	1420	1093
17	Amherst College	Yes	4302	992
18	Anderson University	Yes	1216	908
19	Andrews University	Yes	1130	704

2.1 Розбила набір даних на навчальний та тестовий набори:

R Console

```
> set.seed(1)
> train=sample(1:dim(College)[1],0.5*dim(College)[1])
> College.train=College[train,]
> College.test=College[-train, ]
```

2.2 Оцінила лінійну модель, використовуючи метод найменших квадратів на навчальному наборі, та обчислила тестову помилку:

R Console

```
> fit.lm=lm(Apps~.,data=College.train)
> pred.lm=predict(fit.lm,College.test)
> round(mean((pred.lm-College.test$Apps)^2),2)
[1] 1135758
> |
```

2.3 Пристосувала модель гребеневої регресії до тренувального набору, вибравши  $\lambda$  шляхом перехресної перевірки. Обчислила тестову помилку.

R Console

```
> library(glmnet)
> train.m=model.matrix(Apps~.,data=College.train)
> test.m=model.matrix(Apps~.,data=College.test)
> grid=10^seq(10,-2,length=100)
> fit.ridge=glmnet(train.m,College.train$Apps,alpha=0,lambda=grid)
> cv.ridge=cv.glmnet(train.m,College.train$Apps,alpha=0,lambda=grid)
> dim(coef(fit.ridge))
[1] 19 100
> best_lambda=cv.ridge$lambda.min
> print(best_lambda)
[1] 0.01
> pred.ridge=predict(fit.ridge,s=best_lambda,newx=test.m)
> round(mean((pred.ridge-College.test$Apps)^2),2)
[1] 1134677
> |
```

2.4 Пристосувала модель ласо до тренувального набору, вибравши  $\lambda$  шляхом перехресної перевірки. Обчислила тестову помилку.

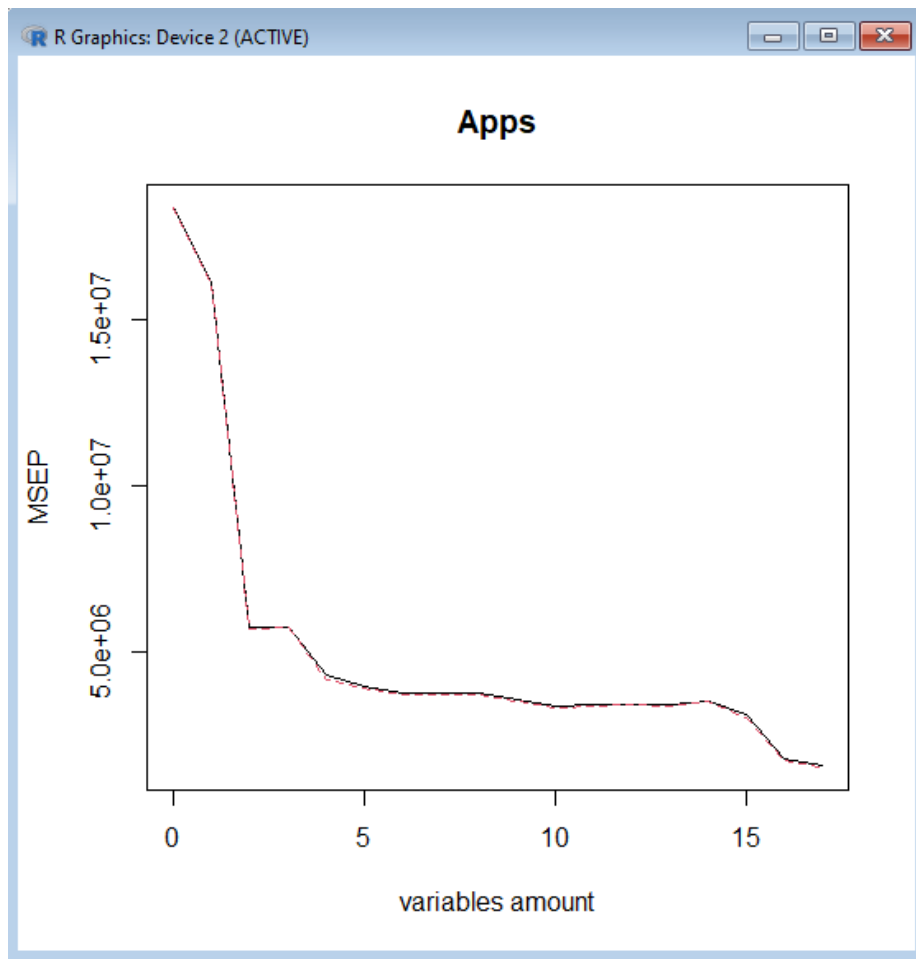
```
R Console
> fit.lasso=glmnet(train.m,College.train$Apps,alpha=1,lambda=grid)
> cv.lasso=cv.glmnet(train.m,College.train$Apps,alpha=1,lambda = grid)
> best_lambda=cv.lasso$lambda.min
> print(best_lambda)
[1] 0.01
> pred.lasso=predict(fit.lasso,s=best_lambda,newx=test.m)
> round(mean((pred.lasso-College.test$Apps)^2),2)
[1] 1133422
> lasso_coefs=predict(fit.lasso,s=best_lambda,type="coefficients")
> print(lasso_coefs)
19 x 1 sparse Matrix of class "dgCMatrix"
      s1
(Intercept) -7.931498e+02
(Intercept) .
PrivateYes -3.078903e+02
Accept 1.777242e+00
Enroll -1.450532e+00
Top10perc 6.659456e+01
Top25perc -2.221506e+01
F.Undergrad 8.983869e-02
P.Undergrad 1.005260e-02
Outstate -1.082871e-01
Room.Board 2.118762e-01
Books 2.922508e-01
Personal 6.234085e-03
PhD -1.542914e+01
Terminal 6.364841e+00
S.F.Ratio 2.284667e+01
perc.alumni 1.114025e+00
Expend 4.861825e-02
Grad.Rate 7.466015e+00
> |
```

В результаті можна побачити, що всі коефіцієнти ненульові.

2.5 Пристосувала модель PCR до тренувального набору, причому М вибрала шляхом перехресної перевірки. Обчислила отриману помилку тесту.

```
R Console
> library(pls)
> fit.pcr=pcr(Apps~.,data=College.train,scale=TRUE,validation="CV")
> validationplot(fit.pcr, val.type = "MSEP", xlab = "variables amount")
> print(which.min(fit.pcr$validation$adj))
[1] 17
> pred.pcr = predict(fit.pcr, College.test, ncomp = which.min(fit.pcr$validation$
[1] 1135758
> |
```

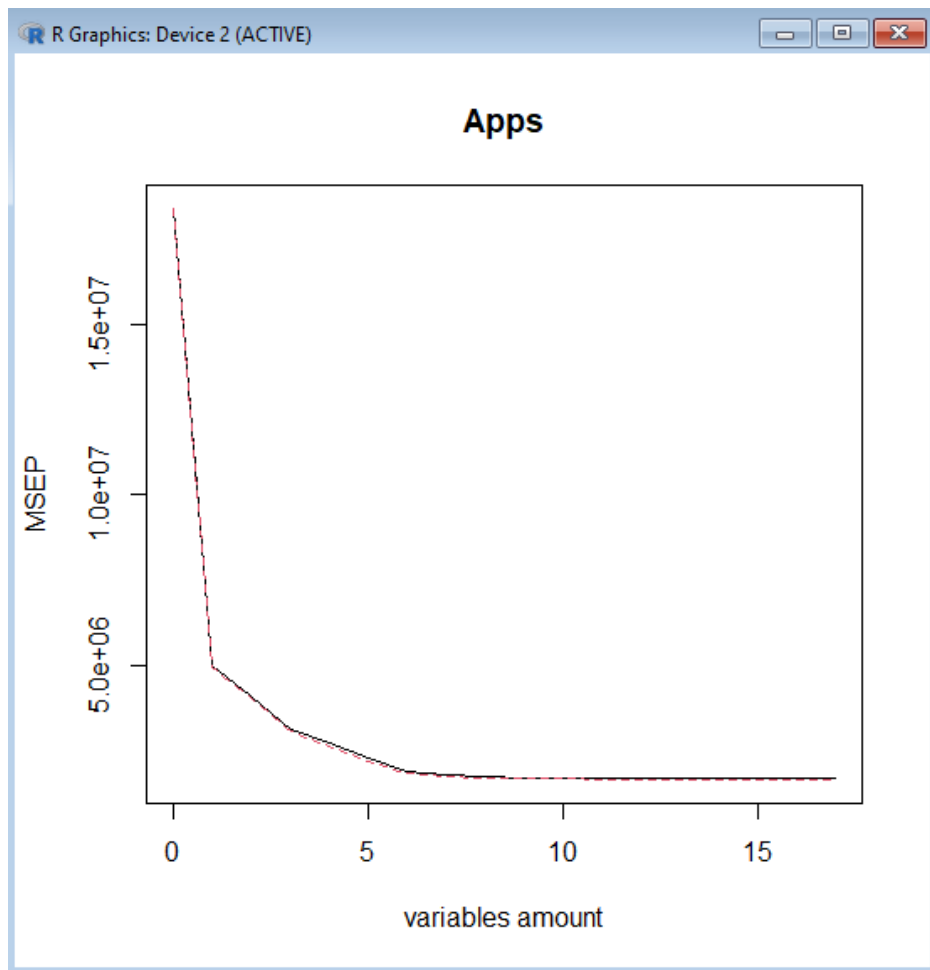
Отримане значення М дорівнює 17.



2.6 Пристосувала модель PLS до тренувального набору, причому М вибрала шляхом перехресної перевірки. Обчислила отриману помилку тесту.

```
R Console
> fit.pls=plsr(Apps~.,data=College.train,scale=TRUE,validation = "CV")
> validationplot(fit.pls,val.type="MSEP",xlab="variables amount")
> print(which.min(fit.pls$validation$adj))
[1] 17
> pred.pls=predict(fit.pls,College.test,ncomp=which.min(fit.pls$validation$adj))
> round(mean((pred.pls-College.test$Apps)^2),2)
[1] 1135758
>
```

Отримане значення М дорівнює 17.



2.7 Визначила, наскільки точно ми можемо передбачити кількість отриманих заявок на коледж кожним з підходів:

```

R Console
> test.mean=mean(College.test$Apps)
> methods.list=c(pred.lm,pred.ridge,pred.lasso,pred.pcr,pred.pls)
> test.mean=mean(College.test$Apps)
> methods.list=c(pred.lm,pred.ridge,pred.lasso,pred.pcr,pred.pls)
> RSqr=function(m) {
+   r_res=1-mean((m-College.test$Apps)^2)/mean((test.mean-College.test$Apps)^2)
+   return(round(r_res,7)*100)}
> print(RSqr(pred.lm))
[1] 90.15413
> print(RSqr(pred.ridge))
[1] 90.16351
> print(RSqr(pred.lasso))
[1] 90.17438
> print(RSqr(pred.pcr))
[1] 90.15413
> print(RSqr(pred.pls))
[1] 90.15413
>

```

Різниця між тестовими помилками, що виникають внаслідок розглянутих п'яти підходів, не велика.

### Завдання 3

3.1 Сформувала набір даних з  $p = 20$  ознаками,  $n = 1000$  спостереженнями, і пов'язаний з ним вектор залежних змінних відповідно до моделі  $Y = X\beta + \varepsilon$ , де вектор  $\beta$  має деякі елементи, які точно дорівнюють нулю.

```
R Console
> set.seed(1)
> x=matrix(rnorm(1000*20),1000,20)
> eps=rnorm(1000)
> beta_values=runif(20,min=-2.72,max=3.14)
> zero_val_quantity=sample(3:10,1)
> for(i in sample(1:length(beta_values),zero_val_quantity)){
+   beta_values[i]=0
+   print(beta_values)
+ }
[1] 1.5350075 0.0000000 -2.7034295 2.7980721 0.6320459 2.4600726 0.0000000
[8] -1.5060529 0.0000000 0.0000000 2.4158184 1.6200996 0.7906195 -2.1023079
[15] -1.9238998 -0.7934472 1.0609377 0.2005645 -2.5021053 0.0000000
> y=x%*%beta_values+eps
> |
```

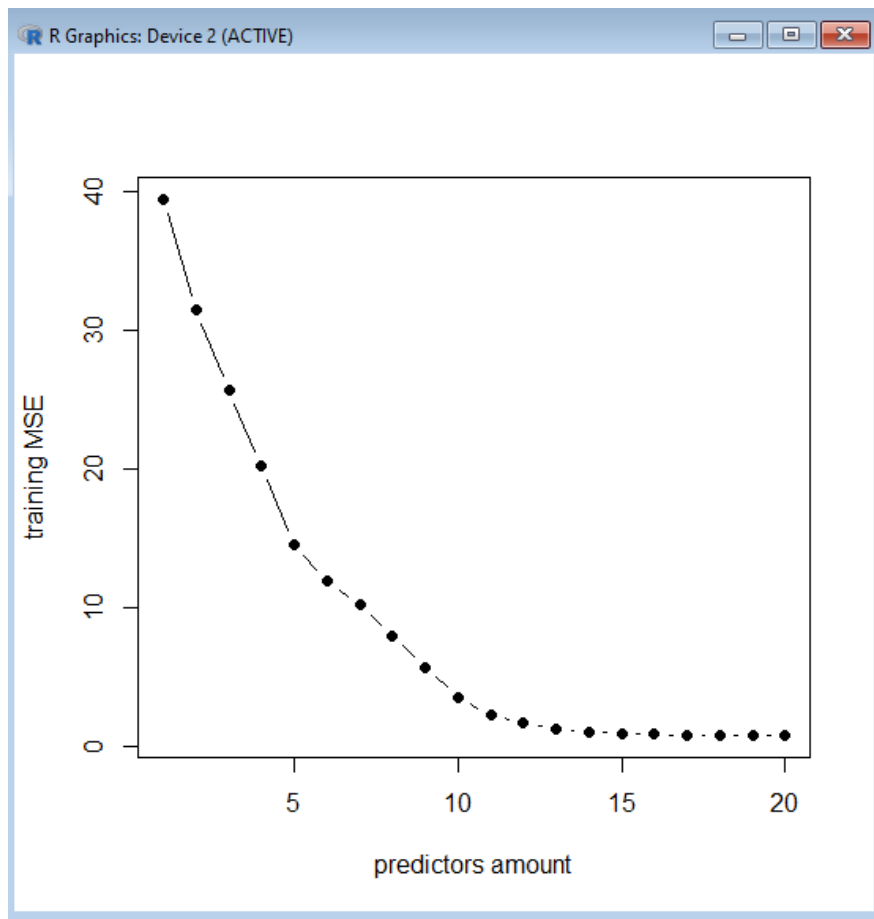
3.2 Розділила свій набір даних на навчальний набір, що містить 100 спостережень та тестовий набір, що містить 900 спостережень:

```
R Console
> train=sample(1:length(eps),100)
> x.train=x[train,]
> y.train=y[train]
> x.test=x[-train,]
> y.test=y[-train]
> |
```

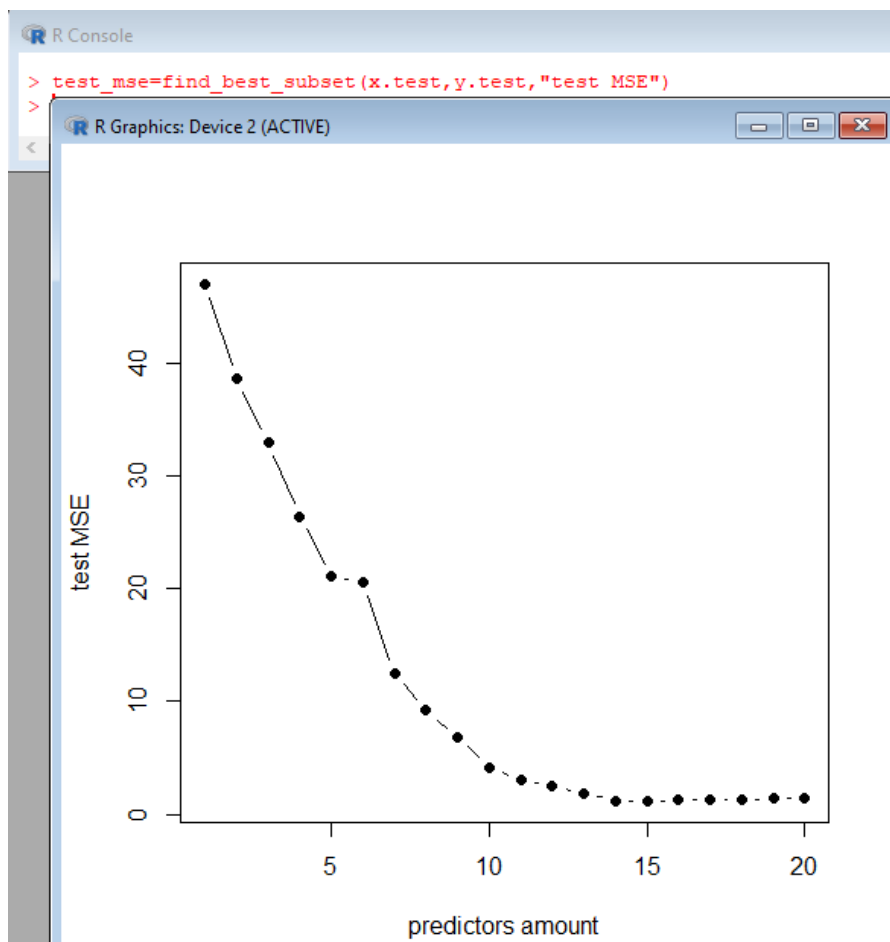
3.3 Використала метод вибору найкращої підмножини на навчальному наборі:

```
R Console
> library(leaps)
> reg.fit=regsubsets(y~.,data=data.frame(y=y.train,x=x.train),nvmax=20)
> find_best_subset=function(x,y,ylab){
+   local_data=data.frame(y=y,x=x)
+   m=model.matrix(y~.,data=local_data,nvmax = 20)
+   val.errors=rep(0,20)
+   for(i in 1:20){
+     coef_i=coef(reg.fit,id= i)
+     pred=m[,names(coef_i)]%*%coef_i
+     val.errors[i]=mean((pred-y)^2)}
+   plot(val.errors,xlab="predictors amount",ylab=ylab,pch=19,type="b")
+   return(val.errors)}
> find_best_subset(x.train,y.train,"training MSE")
[1] 39.5019992 31.4815121 25.6894467 20.2829630 14.6205946 11.9391628 10.2216451
[8] 8.0384360 5.7718576 3.5877601 2.2857950 1.7702461 1.3107050 1.0733538
[15] 0.9809518 0.9116284 0.8680321 0.8530347 0.8459438 0.8459334
> |
```

Побудувала графік навчального MSE, який відповідає найкращій моделі кожного розміру:



3.4 Побудувала графік тестового MSE, який відповідає найкращій моделі кожного розміру:



3.5 Визначила, для якого розміру моделі тестовий MSE приймає мінімальне значення:

```
R Console
> print(which.min(test_mse))
[1] 15
> print(min(test_mse))
[1] 1.170253
> |
```

Для моделі розміром 15 (MSE буде дорівнювати 1.17).

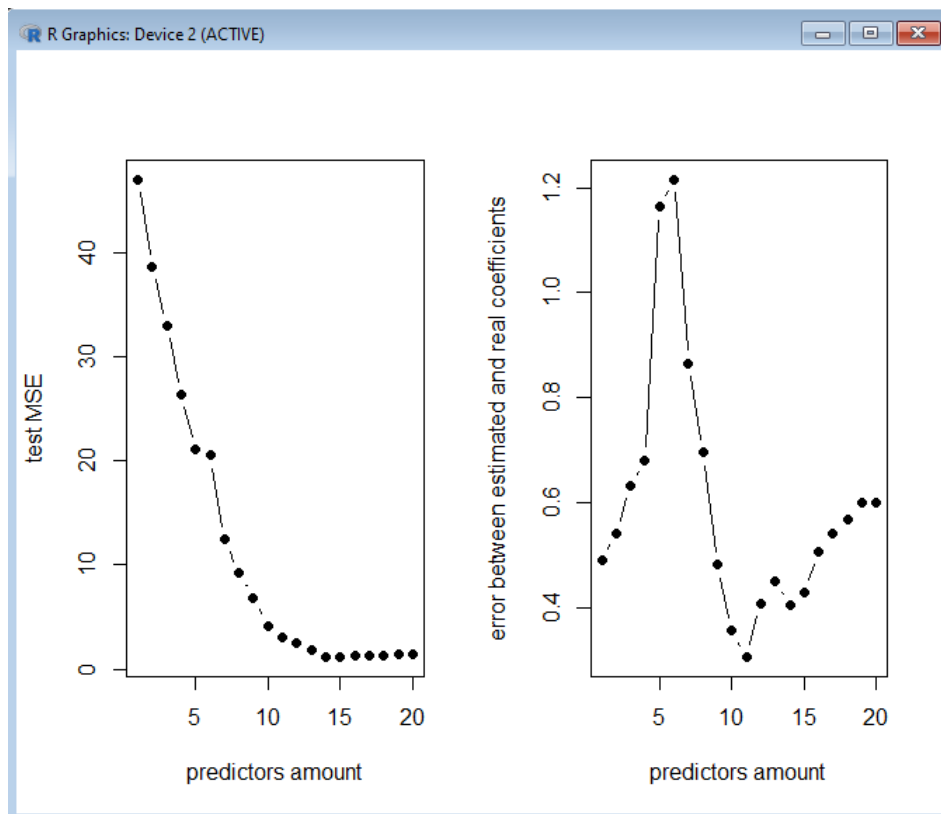
3.6 Розглянула, як співвідносяться модель, що мінімізує тестовий MSE та справжня модель, яка використовувалася для генерації даних

```
R Console
> coef(reg.fit,which.min(test_mse))
(Intercept)      x.1      x.3      x.4      x.5      x.6      x.8
0.0483791    1.6249734   -2.6446191   2.7726741   0.8276008   2.4792278  -1.6001762
      x.11      x.12      x.13      x.14      x.15      x.16      x.17
2.3008815    1.6403113    0.5978145  -2.1394335  -1.8138665  -0.7183958   1.2395690
      x.18      x.19
0.3297070   -2.5767892
> |
```

Можна побачити, що нульові коефіцієнти виключені з моделі.

3.7 Побудувала графік для відображення величини  $\sqrt{\sum_{j=1}^p (\beta_j - \hat{\beta}_j^r)^2}$  для всіх значень  $r$ , де  $\hat{\beta}_j^r$  - оцінка  $j$ -ого коефіцієнта для найкращої моделі, що містить  $r$  коефіцієнтів. Порівняла отриманий графік з графіком тестового MSE з 3.4.

```
R Console
> x_cols=colnames(x,do.NULL=FALSE,prefix="x.")
> for(i in 1:20){
+   coef_i=coef(reg.fit,id=i)
+   val.errors[i]=sqrt(sum((beta_values[x_cols %in% names(coef_i)]-coef_i[names(coef_i) %in% x_cols])^2))}
> par(mfrow=c(1,2))
> find_best_subset(x.test,y.test,"test MSE")
[1] 47.019131 38.633316 32.987160 26.381954 21.113270 20.547826 12.463378  9.192976
[9]  6.868330  4.145006  3.033259  2.512932  1.887346  1.230968  1.170253  1.261337
[17]  1.299058  1.325936  1.371570  1.371136
> plot(val.errors,xlab="predictors amount",ylab="error between estimated and real coefficients",pch=19,type="b")
> |
```



Можна побачити, що мінімальна помилка між оціночними і реальними значеннями коефіцієнтів є у моделі, в якій кількість предикторів дорівнює 11.

**Завдання 4.** Передбачити рівень злочинності на основі набору даних Boston.

R Console

```
> library(MASS)
> fix(Boston)
```

R Data Editor

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2
6	0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9
8	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1
9	0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311	15.2	386.63	29.93	16.5

4.1 Застосувати методи вибору моделі регресії, розглянуті раніше, такі як вибір найкращої підмножини, ласо, гребенева регресія та PCR.

Застосувала метод вибору найкращої підмножини:

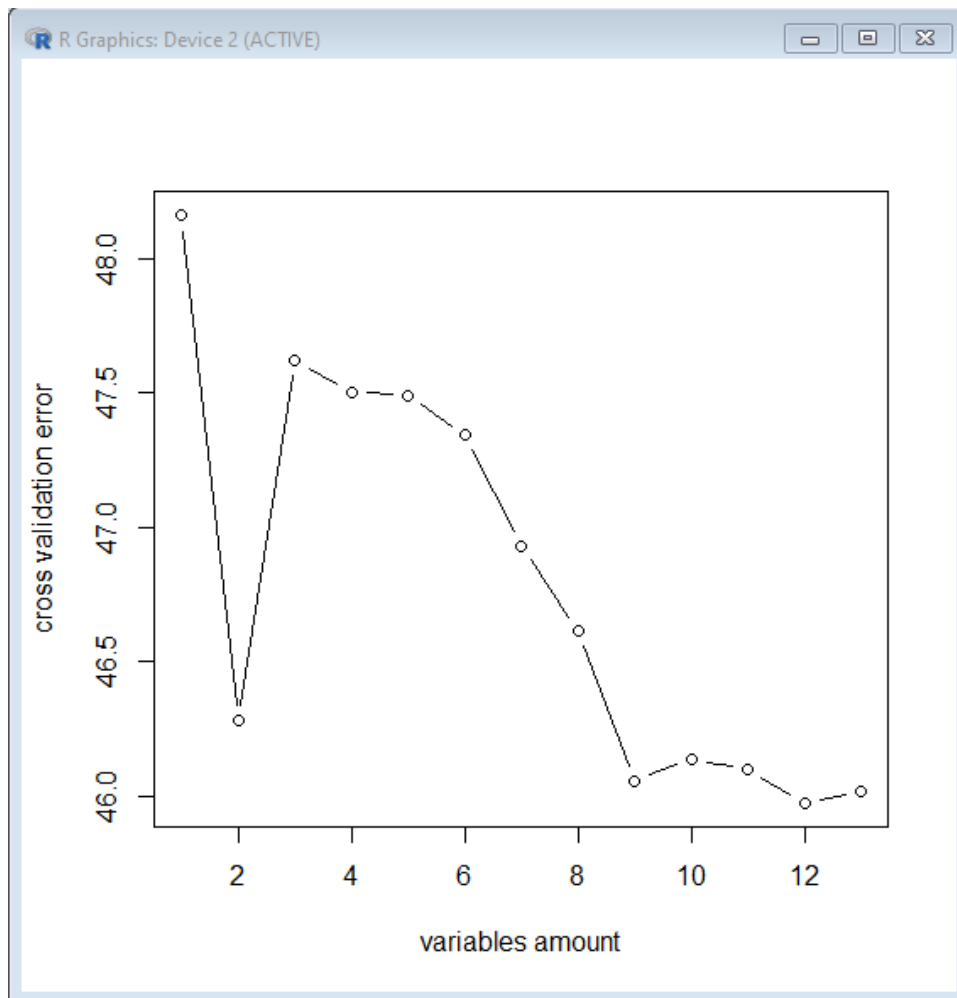


```

> library(leaps)
> predict.regsubsets=function(object,newdata,id,...){
+   form=as.formula(object$call[[2]])
+   m=model.matrix(form,newdata)
+   coef_i=coef(object,id=id)
+   xvars=names(coef_i)
+   m[,xvars]%=coef_i}
> k=10
> folds=sample(1:k,nrow(Boston),replace=TRUE)
> cv.errors=matrix(0,k,13)
> for(j in 1:k){
+   best.fit=regsubsets(crim~.,data=Boston[folds!=j,],nvmax=13)
+   for(i in 1:13){
+     pred=predict.regsubsets(best.fit,Boston[folds==j,],id=i)
+     cv.errors[j, i]=mean((Boston$crim[folds==j]-pred)^2) }}
> mean.cv.errors=rep(0,13)
> for(i in 1:13){
+   mean.cv.errors[i]=mean(cv.errors[,i])}
> print(which.min(mean.cv.errors))
[1] 12
> print(min(mean.cv.errors))
[1] 45.97504
> plot(mean.cv.errors,xlab="variables amount",ylab="cross validation error",type="b")
>

```

Значення помилки дорівнює 45.98



Застосувала метод ласо:

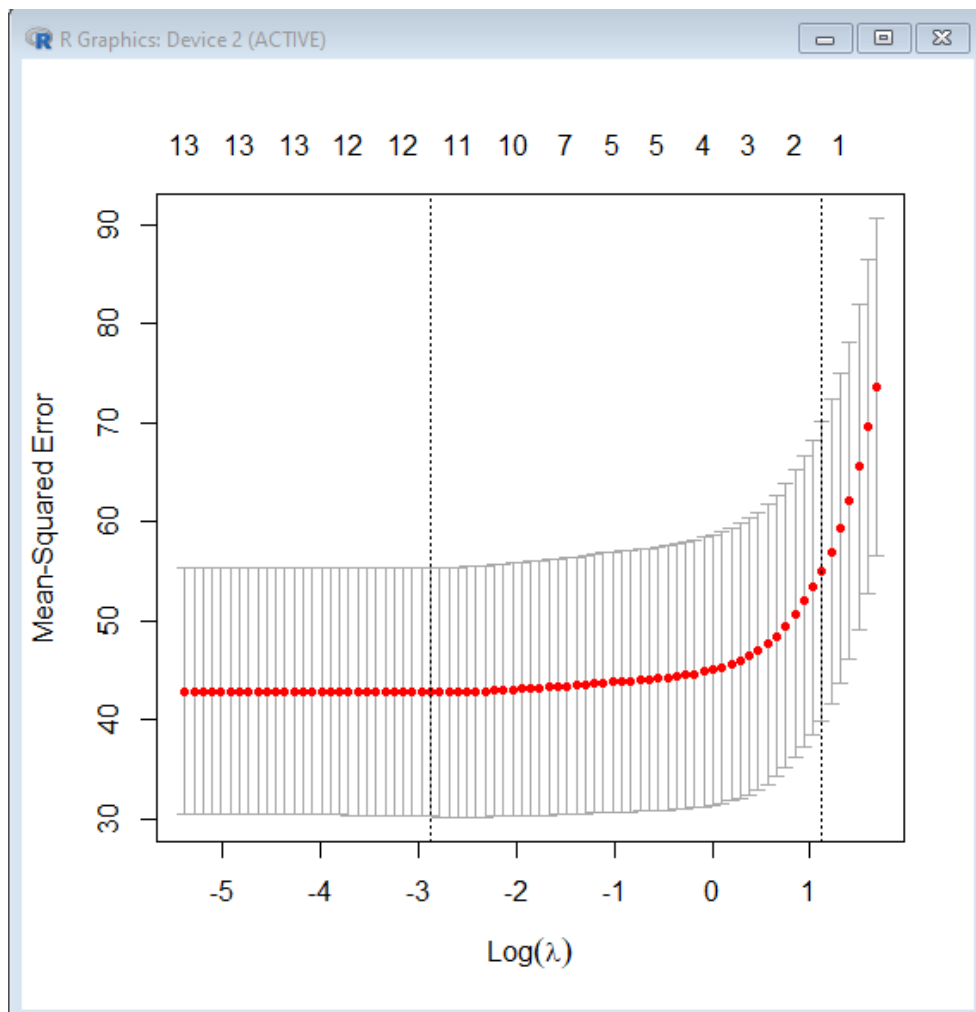
```

R Console
> library(glmnet)
> x=model.matrix(crim~.,Boston)[-1]
> y=Boston$crim
> cv.lasso=cv.glmnet(x,y,alpha=1,type.measure="mse")
> print(cv.lasso$lambda.min)
[1] 0.05630926
> print(min(cv.lasso$cvm))
[1] 42.78045
> plot(cv.lasso)

```

Значення помилки дорівнює 42.78

Побудувала графік помилки перехресної перевірки як функції від  $\lambda$ :



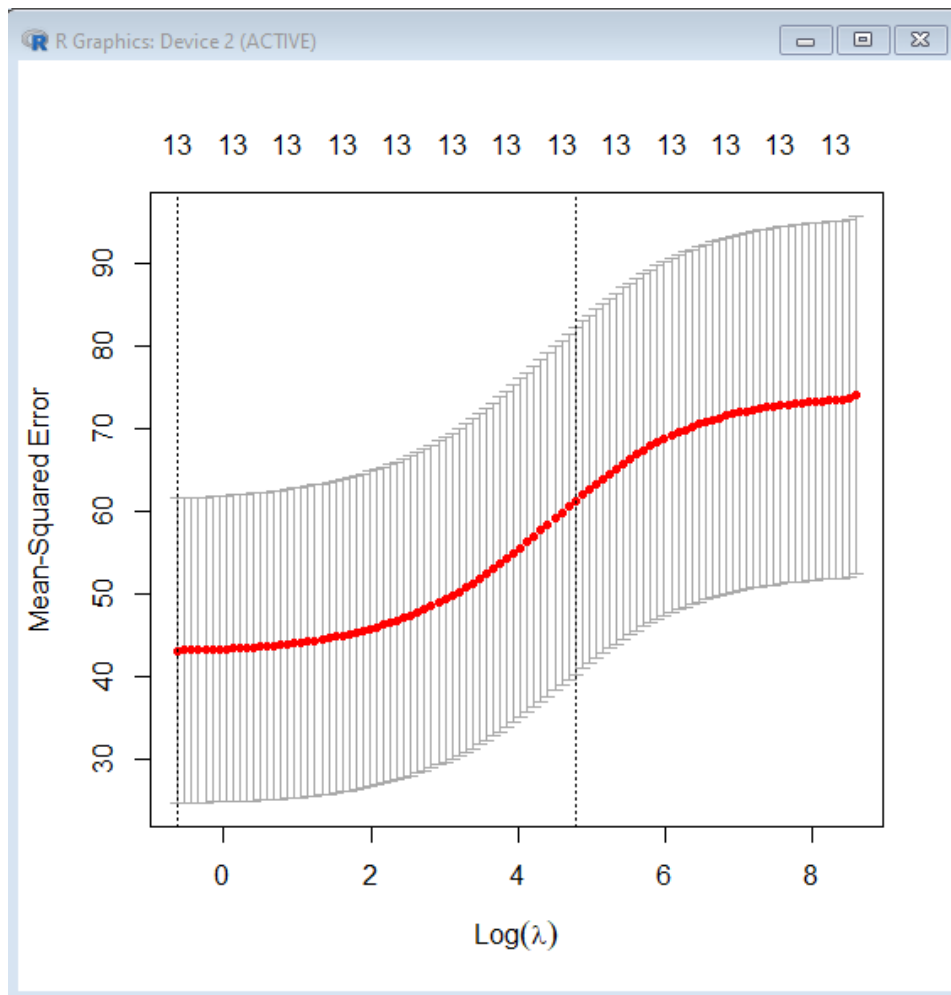
Застосувала метод гребеневої регресії:

```

R Console
> cv.ridge=cv.glmnet(x,y,alpha=0,type.measure="mse")
> print(cv.ridge$lambda.min)
[1] 0.5374992
> print(min(cv.ridge$cvm))
[1] 43.19782
> plot(cv.ridge)
>

```

Значення помилки дорівнює 43.2



Застосувала метод PCR:

```

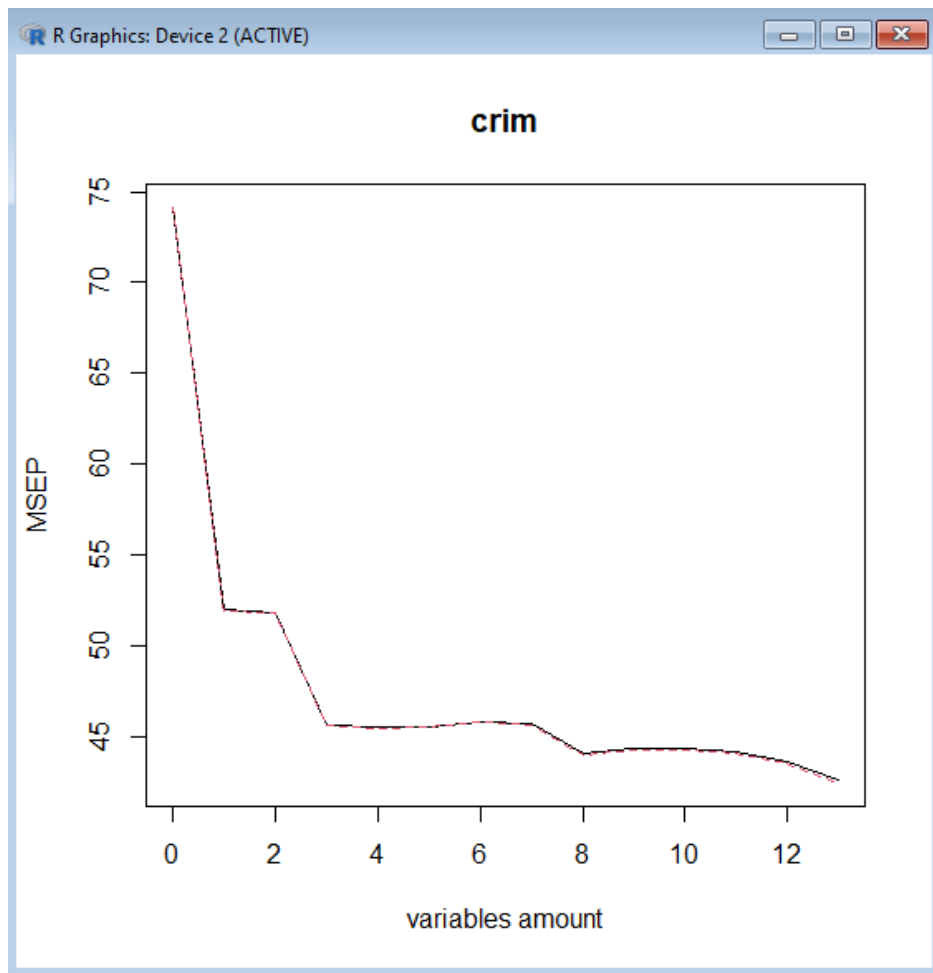
R Console
> library(pls)
> fit.pcr=pcr(crim~.,data=Boston,scale=TRUE,validation="CV")
> summary(fit.pcr)
Data:   X dimension: 506 13
        Y dimension: 506 1
Fit method: svdpc
Number of components considered: 13

VALIDATION: RMSEP
Cross-validated using 10 random segments.
      (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
CV           8.61    7.208    7.199    6.755    6.745    6.750    6.770    6.762
adjCV        8.61    7.205    7.196    6.751    6.738    6.746    6.764    6.756
      8 comps  9 comps 10 comps 11 comps 12 comps 13 comps
CV        6.641  6.661  6.657  6.644  6.605  6.525
adjCV      6.633  6.654  6.649  6.636  6.595  6.515

TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
X       47.70   60.36   69.67   76.45   82.99   88.00   91.14   93.45   95.40
crim    30.69   30.87   39.27   39.61   39.61   39.86   40.14   42.47   42.55
      10 comps 11 comps 12 comps 13 comps
X       97.04   98.46   99.52   100.0
crim    42.78   43.04   44.13   45.4
> print(which.min(fit.pcr$validation$adj))
[1] 13
> print(min(fit.pcr$validation$adj))
[1] 40.43851
> validationplot(fit.pcr,val.type="MSEP",xlab="variables amount")
>

```

Значення помилки дорівнює 40.44



4.2 Можна побачити, що з моделей вище, найменше значення помилки отримано за допомогою методу PCR (40.44), а найбільше – за допомогою методу вибору найкращої підмножини (45.98).

4.3 Модель PCR містить всі 13 предикторів, завдяки цьому і досягається найменше значення помилки.