Laboratory 1

Ex. 1

The task is to estimate prediction errors using different methods, based on the design matrix with random variables from $N(0, \sigma = \frac{1}{1000})$ and response vector of a form:

$$Y = X\beta + \epsilon,$$

where $\beta = (3, 3, 3, 3, 3, \ldots)^T$ and $\epsilon \sim N(0, I)$. Analysis will be performed on models with $p = 5, 10, 20, 100, 500, 950$ first variables and 3 PE estimiators:

a) Using β estimated by least squares 1.1:

$$PE = E||X(\beta - \hat{\beta}) + \epsilon^*||^2,$$

where $\epsilon^* \sim N(0, I)$ is independent on the training sample.

b) Using the residual sum of squares assuming that σ is known 1.2:

$$\hat{PE} = RSS + 2\sigma^2 p,$$

and unknown:

$$\hat{\sigma^2} = \frac{RSS}{n}, \quad \hat{PE} = RSS + 2\frac{RSS}{n}p,$$

c) Using leave-one-out crossvalidation (given the formula) 1.3:

$$CV = \sum_{i=1}^{n} \left(\frac{Y_i - \hat{Y}_i}{1 - M_{ii}} \right)^2, \quad M = X(X^T X)^{-1} X^T,$$

	PE_LS	PE_sigma	PE_RSS	PE_CV	AIC
5	1007.371	995.856	995.764	995.468	6903.510
10	980.015	1002.116	1001.957	1001.959	6909.710
20	1021.175	1013.548	1013.284	1014.053	6920.947
100	1039.200	1063.374	1055.235	1066.607	6960.848
500	1463.936	1459.407	1378.222	1832.755	7129.937
950	1982.490	1929.182	1138.103	12013.812	5273.557

Table 1.1: Estimators of PE 1.5.

Table 1.1 presents calculation of PE using least squares and estimators using RSS with known σ, unknown σ, crossvalidation and AIC with unknown σ respectively, for every number of p variables. When number of variables is closer to the true number those estimators are close to the real value of prediction error. CV method is getting bigger with more variables.

Estimator *PE_sigma* is equivalent to AIC with known σ. Based on values of AIC (for known and unknown σ) in the table we can select best model – 5 first variables in each case (We are not taking full model into account, because we are looking for local minimum). This model is minimizing the AIC.

	PE_LS	PE_sigma	PE_RSS	PE_CV	AIC
5	1001.801	1002.180	1002.152	1002.171	6908.893
10	1003.339	1007.305	1007.250	1007.315	6913.976
20	1026.101	1018.261	1018.190	1018.602	6924.732
100	1093.751	1100.151	1100.185	1111.728	7001.391
500	1503.254	1498.823	1496.470	1999.963	7210.238
950	1948.240	1950.634	1974.720	21087.037	5807.887

Table 1.2: Mean of estimators of PE 1.7.

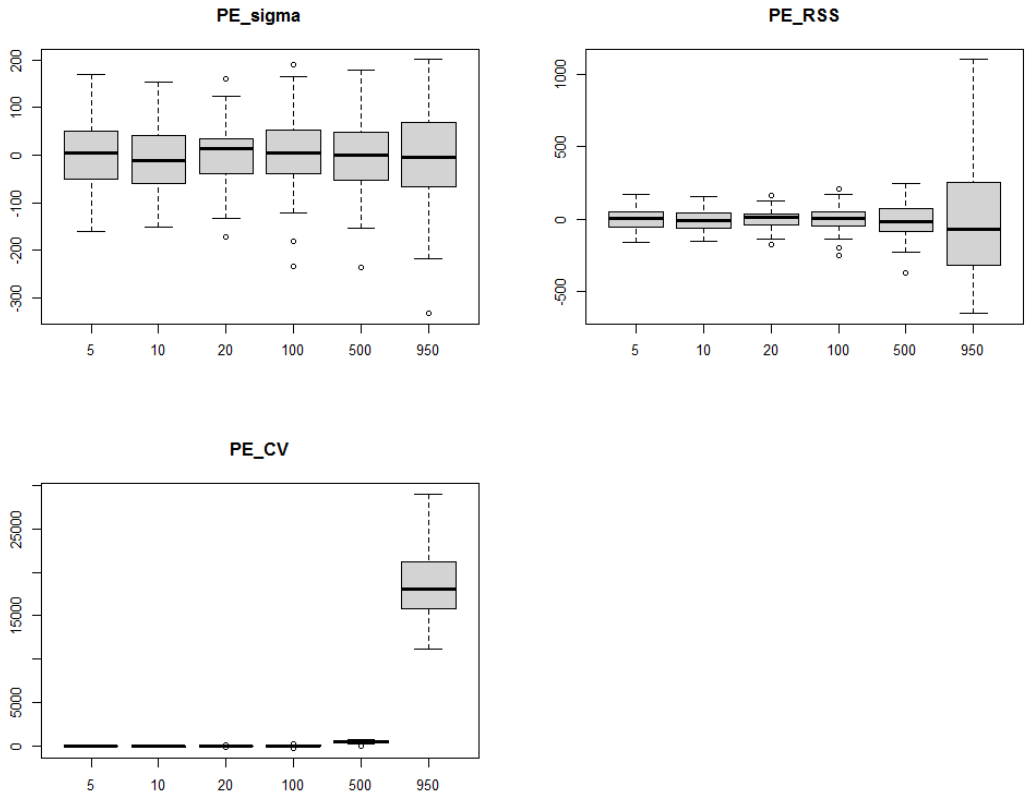

Figure 1.1: Boxplots of differences of three estimators and true PE 1.7.

Boxplots in 1.1 show the difference between estimators of PE and PE obtained via least squares estimator. Both methods that use RSS are close to real PE. Method with known σ has constant variance relative to the number of variables. Method with unknown σ has increasing error relative to the number of variables. CV method has even bigger error when number of variables is close to the number of data.

Ex. 2

The task is to use different information criteria to identify important covariates, when the search is performed over the data base consistinig of p first variables.

a) Number of false and true discoveries and mean square error (2.1):

	MSE	FD	TD	D	FDR	power
20	1.06	3	5	8	0.38	1
100	1.02	14	5	19	0.74	1
500	0.84	65	5	70	0.93	1
950	0.84	65	5	70	0.93	1

(a) Statistics for aic.

	MSE	FD	TD	D	FDR	power
20	1.08	0	3	3	0.00	0.60
100	1.08	0	3	3	0.00	0.60
500	1.03	7	3	10	0.70	0.60
950	0.99	11	3	14	0.79	0.60

(b) Statistics for bic.

	MSE	FD	TD	D	FDR	power
20	1.11	0	1	1	0	0.20
100	1.12	0	0	0	0	0.00
500	1.12	0	0	0	0	0.00
950	1.12	0	0	0	0	0.00

(c) Statistics for mbic.

	MSE	FD	TD	D	FDR	power
20	1.08	0	3	3	0	0.60
100	1.12	0	0	0	0	0.00
500	1.12	0	0	0	0	0.00
950	1.12	0	0	0	0	0.00

(d) Statistics for mbic2.

	MSE	FD	TD	D	FDR	power
20	1.08	0	3	3	0	0.60
100	1.11	0	1	1	0	0.20
500	1.12	0	0	0	0	0.00
950	1.12	0	0	0	0	0.00

(e) Statistics for ric

Table 2.1: Table of MSE, false (FD) and true (TD) discoveries (D=TD+FD), false discovery rate and power (2.1).

b) Repeating point a) 100 times and estimating power, FDR and MSE (2.2).

	MSE	FD	TD	D	FDR	power
20	0.99	1.94	4.76	6.70	0.26	0.95
100	0.95	14.48	4.76	19.24	0.74	0.95
500	0.73	82.64	4.76	87.40	0.95	0.95
950	0.53	166.75	4.76	171.51	0.97	0.95

(a) Statistics for aic.

	MSE	FD	TD	D	FDR	power
20	1.01	0.07	3.34	3.41	0.01	0.67
100	1.00	0.71	3.34	4.05	0.16	0.67
500	0.97	4.58	3.34	7.92	0.56	0.67
950	0.93	8.55	3.34	11.89	0.71	0.67

(b) Statistics for bic.

	MSE	FD	TD	D	FDR	power
20	1.01	0.01	2.41	2.42	0.00	0.48
100	1.03	0.05	1.31	1.36	0.03	0.26
500	1.04	0.04	0.62	0.66	0.03	0.12
950	1.04	0.05	0.49	0.54	0.03	0.10

(c) Statistics for mbic.

	MSE	FD	TD	D	FDR	power
20	1.01	0.05	2.78	2.83	0.01	0.56
100	1.02	0.08	1.46	1.54	0.04	0.29
500	1.04	0.07	0.67	0.74	0.04	0.13
950	1.04	0.06	0.52	0.58	0.04	0.10

(d) Statistics for mbic2.

	MSE	FD	TD	D	FDR	power
20	1.00	0.10	3.63	3.73	0.02	0.73
100	1.01	0.16	2.64	2.80	0.06	0.53
500	1.02	0.28	1.50	1.78	0.10	0.30
950	1.02	0.27	1.19	1.46	0.13	0.24

(e) Statistics for ric.

Table 2.2: Table of average MSE, false (FD) and true (TD) discoveries (D=TD+FD), false discovery rate and power (2.2).

Table 2.2 presents results of 100 repetitions and averaging over them. We can conclude, that in terms of power AIC is vastly superior to other IC's. It almost always chooses the right covariates to include in models, but because of this "less restrictive" selection it also takes a lot of other false variables. That's why FDR of AIC is very high. Other IC try to hold FDR on lower level (0.05), but because of that, they never discover every true covariate. MSE is case of every IC is close to 1.

We know from theory, that AIC makes a false discovery with probability of 0.16, in the case of our setup it gives $2.4, 15.2, 79.2, 151.2$ false discoveries respectively. We can say the same about the BIC, which makes a false discovery with probability $2(1 - \Phi(\sqrt{\log 1000})) = 0.009$, in our case:

$0.13, 0.82, 4.25, 8.11$. Modified BIC's control FDR at level $1000^{-1/2} = 0.032$. The results are close to the theoretical values.

Ex. 3

The task is to compare $RIC, mBIC, mBIC2$ using example vi) of problem 1 when the vector of true regression coefficients contains 50 nonzero entries.

	MSE	FD	TD	D	FDR	power
950	1.39	0.01	1.84	1.85	0.01	0.04

(a) Statistics for mbic.

	MSE	FD	TD	D	FDR	power
950	1.37	0.09	2.75	2.84	0.02	0.05

(b) Statistics for mbic2.

	MSE	FD	TD	D	FDR	power
950	1.31	0.26	5.67	5.93	0.04	0.11

(c) Statistics for ric.

Table 3.1: Table of average MSE, false (FD) and true (TD) discoveries (D=TD+FD), false discovery rate and power (3.1).

Table 3.1 presents results of 100 repetitions and averaging over them. We can conclude that these IC's always control the FDR but they are probably to strict and also do not include important variables, so the power is also low. Power is lower than in previous task, because there were more β_i to discover, while those methods are still strict and choose 1-2 variables.

Ex. 4

The task is to generate the vector of the response variable according to model $Y = X\beta + \varepsilon$, where $\varepsilon \sim Exp(1)$ or $\varepsilon \sim Cauchy(0, 1)$ and report estimated FDR, power and estimate coefficients with the robust regression. Repeating experiment 100 times and estimating FDR and power:

	MSE	FD	TD	D	FDR	power
950	1.11	0.04	28.02	28.06	0.00	0.93

(a) Statistics for mbic.

	MSE	FD	TD	D	FDR	power
950	1.03	0.94	28.92	29.86	0.03	0.96

(b) Statistics for mbic2.

	MSE	FD	TD	D	FDR	power
950	24020	0.04	28.06	28.10	0.00	0.94

(c) Statistics for rbic.

	MSE	FD	TD	D	FDR	power
950	22413	1.15	28.89	30.04	0.04	0.96

(d) Statistics for rbic2.

Table 4.1: Table of average MSE, false (FD) and true (TD) discoveries (D=TD+FD), false discovery rate and power for $\varepsilon \sim Exp(1)$ (4.1).

The exponential error term presented in fig. 4.1 seems to not affect that much the selection, methods using ranks are not needed and they give the same results.

	MSE	FD	TD	D	FDR	power
950	38809	0.03	0	0.03	0.03	0

(a) Statistics for mbic.

	MSE	FD	TD	D	FDR	power
950	38810	0.03	0	0.03	0.03	0

(b) Statistics for mbic2.

	MSE	FD	TD	D	FDR	power
950	73709	0.01	5.37	5.38	0.00	0.18

(c) Statistics for rbic.

	MSE	FD	TD	D	FDR	power
950	69809	0.28	8.55	8.83	0.03	0.28

(d) Statistics for rbic2.

Table 4.2: Table of average MSE, false (FD) and true (TD) discoveries (D=TD+FD), false discovery rate and power for $\varepsilon \sim Cauchy(0, 1)$ (4.1).

The cauchy error presented in fig. 4.2 is making original methods useless, so the help of ranking observations is needed. While those do not work well, they are improving the selection.

Using variables selected by $rBIC2$ and fitting model using least squares, huber and Bi-square robust regression we can estimate mean square error of estimation of coefficients:

	LS	Huber	Bi-square
$Exp(1)$	3.47	1.05	1.11
$Cauchy(0, 1)$	915.78	6.62	5.69

Table 4.3: MSE of estimation of regression coefficients in different methods (4.2).

MSE of estimations of coefficients presented in fig. 4.3 are done better by robust regression in comparision to least squares. In case of error term from Cauchy distribution the least squares method is estimating coefficients from wide range, what makes the MSE large.

Ex. 5

The task is to generate the binary response variable according to the logistic regression model and use information criterion to identify important covariates.

Calculating the MSE of estimation of coefficients:

	bic	mbic	mbic2
950	78.98	61.16	66.42

Table 5.1: MSE of estimates of coefficients (5.3).

Repeating experiment 100 times and estimating FDR, power and MSE (of model):

	MSE	FD	TD	D	FDR	power
950	0.09	168.70	29.47	198.17	0.85	0.98

(a) Statistics for aic.

	ACC	FD	TD	D	FDR	power
950	0.78	9.29	24.70	33.99	0.27	0.82

(b) Statistics for bic.

	ACC	FD	TD	D	FDR	power
950	0.64	0.01	5.55	5.56	0.00	0.18

(c) Statistics for mbic.

	ACC	FD	TD	D	FDR	power
950	0.67	0.25	8.71	8.96	0.03	0.29

(d) Statistics for mbic2.

Table 5.2: Table of average accuracy, false (FD) and true (TD) discoveries (D=TD+FD), false discovery rate and power (5.2).

Table 5.2 presents results of repeating the experiment. The AIC was not calculated using *logistic* regression, because it was taking almost every variable to model. It's statistics are calculated using *linear* regression. The other IC's have either low power, but hold FDR or have high power and introduce more false discoveries.

Table 5.1 shows mean squared error of estimation of coefficients. AIC with linear regression has even bigger error than ever other. The best criterion in this regard is $mBIC$.

Appendices

Ex. 1

```
generate_design <- function(n=1000, p=950, beta=3, nonzero=5) {
  X = matrix(rnorm(n*p, 0, 1/sqrt(n)), n, p)
  betas = c(rep(beta,nonzero), rep(0,p-nonzero))
  colnames(X) = betas
  return(X)
}

generate_response <- function(X, rdist=rnorm) {
  betas = as.numeric(colnames(X))
  X%*%betas + rdist(nrow(X))
}
```

Listing 1.1: LS method

```
PE_LS <- function(X, betas_hat) {
  betas = as.numeric(colnames(X))
  mean(sum( (X%*%(betas-betas_hat)+rnorm(nrow(X)))^2 ))
}
```

Listing 1.2: RSS method

```
PE_RSS <- function(RSS, sigma=NULL, n, p) {
  if(is.null(sigma))
    RSS + 2*(RSS/(n-p))*p
  else
    RSS + 2*sigma^2*p
}
```

Listing 1.3: CV method

```
PE_CV <- function(X, Y, Y_hat) {
  H = X%*%solve(t(X)%*%X)%*%t(X)
  sum( ((Y-Y_hat) / (1-diag(H)) )^2);
}
```

Listing 1.4: AIC

```
AIC <- function(RSS, n, p) n*log(RSS) +2*p
```

Listing 1.5: PE estimators

```
PE_estimators <- function(n=1000, P = c(5,10,20,100,500,950), beta=3,
    nonzero=5) {
  data = data.frame()
  X = generate_design(n,max(P),beta,nonzero)
  Y = generate_response(X)

  for (p in P) {
    model = lm(Y~X[,1:p]-1)
    RSS = sum(model$residuals^2)
    data = rbind(data, c(PE_LS(X[,1:p], model$coefficients), PE_RSS(RSS,1,n
        ,p), PE_RSS(RSS,NULL,n,p), #1/sqrt(n)?
                              PE_CV(X[,1:p],Y,predict(model)), AIC(RSS,n,p) ) )
  }
  colnames(data) = c( "PE_LS", "PE_sigma", "PE_RSS", "PE_CV", "AIC")
  rownames(data) = P
```

```
  #printTable(data, "Estimators of PE.", "SL_lab1_ex1_PE_est")
  return(data)
}
```

Listing 1.6: Model selection

```
select_model <- function(PE_data) {
  best_known = as.numeric(rownames(PE_data)[which.min(PE_data$`PE_sigma`)])
  best_unknown = as.numeric(rownames(PE_data)[which.min(PE_data$`AIC`)])
  data.frame(PE_sigma=best_known, AIC=best_unknown)
}
```

Listing 1.7: PE comparison

```
PE_comparison <- function(n=1000, P = c(5,10,20,100,500,950), beta=3,
    nonzero=5, rep=100) {
  data = matrix(0, length(P), 5)
  SIG = data.frame(); RSS = data.frame(); CV = data.frame(); best_models=
      data.frame();
  for (i in 1:rep) {
    est = PE_estimators(n,P,beta,nonzero)
    data = data + est
    SIG = rbind(SIG, est$`PE_sigma`-est$`PE_LS`)
    RSS = rbind(RSS, est$`PE_RSS`-est$`PE_LS`)
    CV = rbind(CV, est$`PE_CV`-est$`PE_LS`)
    #best_models = rbind(best_models, select_model(est[-6,]))
  }
  data = data/rep
  colnames(SIG) = paste(P)
  colnames(RSS) = paste(P)
  colnames(CV) = paste(P)

  #FP = data.frame(PE_sigma=best_models$`PE_sigma`-nonzero, AIC=best_models
      $`AIC`-nonzero)
  #FN = data.frame(PE_sigma=best_models$`PE_sigma`-nonzero, AIC=best_models
      $`AIC`-nonzero)

  par(mfrow=c(2,2))
  boxplot( SIG, main="PE_sigma" )
  boxplot( RSS, main="PE_RSS" )
  boxplot( CV, main="PE_CV" )
  #par(mfrow=c(2,2))
  #hist(FP$`PE_sigma`, main="FP PE_sigma", xlim = c(0,1000))
  #hist(FP$`AIC`, main="FP AIC", xlim = c(0,1000))
  #hist(FN$`PE_sigma`, main="FN PE_sigma", xlim = c(0,1000))
  #hist(FN$`AIC`, main="FN AIC", xlim = c(0,1000))
  par(mfrow=c(1,1))

  save(list=c("data"), file=paste("SL_lab1_ex1_PE_comp",".RData", sep=""))
  #printTable(data, "Mean of estimators of PE.", "SL_lab1_ex1_PE_comp")
  return(data)
```

Ex. 2

Listing 2.1: Covariates identification

```
covariates_identification <- function(n=1000, P=c(20,100,500,950), beta=3,
    nonzero=5, methods = c("bic", "mbic", "mbic2", "aic", "ric"), rdist=
    rnorm, type="linear") {
```

```
  X = generate_design(n,max(P),beta,nonzero)
  Y = generate_response(X, rdist)
  methods_stats = list()
  for (crit_met in methods) {
    MSE=NULL; FD=NULL; TD=NULL; data=data.frame()

    for (p in P) {
      if(crit_met=="ric")
        FF_obj =  fast_forward(prepare_data(Y,unname(X[,1:p]), type=type),
            crit=ric, maxf = p)
      else if(crit_met=="rbic")
        FF_obj =  fast_forward(prepare_data(rank(Y),unname(X[,1:p]), type=
            type), crit="mbic", maxf = p)
      else if(crit_met=="rbic2")
        FF_obj =  fast_forward(prepare_data(rank(Y),unname(X[,1:p]), type=
            type), crit="mbic2", maxf = p)
      else
        FF_obj =  fast_forward(prepare_data(Y,unname(X[,1:p]), type=type),
            crit=crit_met, maxf = p)

      if(is.null(FF_obj$model)){
        FD = 0
        TD = 0
        MSE = sum(lm(Y~1)$residuals^2)/n
      }else{
        FD = sum(!(as.numeric(FF_obj$model) %in% 1:nonzero))
        TD = sum(as.numeric(FF_obj$model) %in% 1:nonzero)
        MSE = FF_obj$metric_v
      }
      data = rbind(data, c(MSE, FD, TD, TD+FD, FD/max(TD+FD,1), TD/nonzero)
          )
    }
    rownames(data) = paste(P)
    colnames(data) = c("MSE", "FD", "TD", "D", "FDR", "power")
    methods_stats[[crit_met]] = data
  }

  return(methods_stats)
}
```

Listing 2.2: Estimating covariates identification

```
estimate_cov_identification <- function(n=1000, P=c(20,100,500,950), beta
    =3, nonzero=5, methods = c("bic", "mbic", "mbic2", "aic","ric"), rep
    =100,  rdist=rnorm, type="linear") {
  methods_stats = covariates_identification(n, P, beta, nonzero, methods,
      rdist, type)
  for (i in 1:(rep-1)) {
    tmp = covariates_identification(n, P, beta, nonzero, methods, rdist,
        type)
    for (j in 1:length(methods))
      methods_stats[[j]] =  methods_stats[[j]] + tmp[[j]]
  }

  for (i in 1:length(methods_stats)) {
    methods_stats[[i]] = methods_stats[[i]]/rep
  }

  save(list=c("methods_stats"), file=paste("SL_lab1_ex5_est_covariates",".
```

```
      RData", sep=""))
  for (i in 1:length(methods_stats)) {
    printTable(methods_stats[[i]], paste("Statistics for", methods[i]),
        paste("SL_lab1_ex5_est_covariates_",methods[i],sep=""))
  }#SUBTABLES AND 2 DIGITS!!!!

  return(methods_stats)
}
```

Ex. 3

Listing 3.1: Estimating covariates identification

```
estimate_cov_identification(P=c(950), nonzero = 50, methods = c("mbic", "
    mbic2", "ric"))
```

Ex. 4

Listing 4.1: Estimating covariates identification

```
estimate_cov_identification(P=c(950), beta = 10, nonzero = 30, methods = c(
    "mbic", "mbic2", "rbic", "rbic2"), rdist=rexp, rep=100)
estimate_cov_identification(P=c(950), beta = 10, nonzero = 30, methods = c(
    "mbic", "mbic2", "rbic", "rbic2"), rdist=rcauchy, rep=100)
```

Listing 4.2: MSE of estimate of coefficients

```
X = generate_design(1000,950,10,30)
Y = generate_response(X, rcauchy)
FF_obj =  fast_forward(prepare_data(rank(Y),unname(X)), crit="mbic2", maxf
    = 950)
model = as.numeric(FF_obj$model)
library(MASS)
LS = lm(Y~unname(X[,model])-1)$coefficients
hub = rlm(X[,model],Y, psi=psi.huber)$coefficients
bisq = rlm(X[,model],Y, psi=psi.bisquare)$coefficients
mean((LS- 10)^2)
```

Ex. 5

Listing 5.1: Generating response

```
generate_response <- function(X, rdist=rnorm) {
  betas = as.numeric(colnames(X))
  logit = X%*%betas
  p = exp(logit)/(1+exp(logit))
  rbinom(nrow(X), 1, p)
}
```

Listing 5.2: Estimating covariates identification

```
estimate_cov_identification(P=c(950), beta = 10, nonzero = 30, methods = c(
    "bic", "mbic", "mbic2"), rep=100, type = "logistic")
```

Listing 5.3: MSE of estimate of coefficients

```
estimate_coefficients <- function(n=1000, P=c(950), beta = 10, nonzero =
    30, methods = c("bic", "mbic", "mbic2"), rep=100, type = "logistic") {
  MSE = matrix(0, length(P), length(methods))
  for (r in 1:rep) {
```

```
    X = generate_design(n,max(P),beta,nonzero)
    Y = generate_response(X)
    mse = matrix(0, length(P), length(methods))
    for (i in 1:length(methods)) {
      crit_met = methods[i]
      for (j in 1:length(P)) {
        p = P[j]
        FF_obj =  fast_forward(prepare_data(Y,unname(X[,1:p]), type=type),
            crit=crit_met, maxf = p)
        model = as.numeric(FF_obj$model)
        ls = lm(Y~X[,model]-1)$coefficients
        mse[j,i] = mean((ls - beta)^2)
      }
    }
    MSE = MSE + mse
  }

  MSE = data.frame(MSE/rep)
  colnames(MSE) = methods
  rownames(MSE) = P

  printTable(MSE, "MSE of estimates of coefficients.", "SL_lab1_ex5_mse")
  return(MSE)
}
estimate_coefficients(rep=10)
```