Laboratory 3

Ex. 1

The task is to generate design matrix $X_{500\times 450}$ such that its elements are i.i.d random variables from $N(0, \frac{1}{\sqrt{500}})$, then generate response variable according to model:

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

where $\epsilon \sim 2N(0, I)$, $\beta_i = 10$ for $i = \{1, \ldots, k\}$ and $\beta_i = 0$ otherwise, $k \in \{5, 20, 50\}$. For 100 replications of the above model we need to estimate the regression coefficients and indentify important variables using: least squares, ridge regression, LASSO (with cross-validation), ridge with knockoffs (1.2), LASSO with knockoffs (1.2), adaptive LASSO 1 and 2 (1.3 and 1.4), adaptive SLOPE (1.5).

	$MSE(\beta)$	$MSE(\mu)$	FWER	FDR	power
OLS	18668.560	1808.461	-	-	-
ridge	420.620	348.158	-	-	-
LASSO	135.531	128.205	1.000	0.762	0.996
ridge_kf	491.932	478.167	0.330	0.113	0.478
LASSO_kf	223.318	216.512	0.300	0.094	0.716
ALASSO1	379.352	399.052	1.000	0.758	0.996
ALASSO2	89.550	86.859	0.590	0.162	0.956
ASLOPE	84.110	81.428	0.710	0.224	0.968

Table 1.1: Results for $k = 5$.

	$MSE(\beta)$	$MSE(\mu)$	FWER	FDR	power
OLS	17677.110	1795.061	-	-	-
ridge	1208.273	762.603	-	-	-
LASSO	422.184	348.191	1.000	0.741	0.999
ridge_kf	1947.546	1843.774	0.810	0.158	0.664
LASSO_kf	462.333	406.367	0.890	0.158	0.958
ALASSO1	781.928	714.615	1.000	0.727	0.999
ALASSO2	333.520	297.140	0.960	0.145	0.973
ASLOPE	288.249	253.189	1.000	0.244	0.987

Table 1.2: Results for $k = 20$.

	$MSE(\beta)$	$MSE(\mu)$	FWER	FDR	power
OLS	18618.186	1808.578	-	-	-
ridge	2297.097	1053.605	-	-	-
LASSO	1027.404	675.742	1.000	0.684	0.998
ridge_kf	4913.854	4565.632	0.810	0.154	0.354
LASSO_kf	1381.364	1020.936	1.000	0.180	0.897
ALASSO1	1294.903	905.940	1.000	0.630	0.997
ALASSO2	1010.192	762.780	1.000	0.149	0.958
ASLOPE	821.232	606.148	1.000	0.269	0.983

Table 1.3: Results for $k = 50$.

The results of those experiments are presented in tables 1.1, 1.2, 1.3. Clearly ordinary least squares is the worst method and it's because variances of β_i are elements of the diagonal of matrix from inverse Wishart distribution, which become very large when p approaches n. Ridge regression improves upon OLS. LASSO is better than last 2 methods in every case, it always discovers every important variable (almost 100% power), but there are still things to improve – it passes many false discoveries (high FDR and following it 100% of false diiscovery – FWER). Knockoff procedure helps Ridge regression and LASSO with controling FDR level, but trades it off for MSE.

Adaptive LASSO 1 is an example of wrong usage of weights – in theory it's great idea, but it's not taking into consideration noise (σ) and makes it worse than regular LASSO. Adaptive LASSO 2 uses estimation of σ and because of that it improves on every aspect of LASSO (most imporantly holds FDR). Adaptivie SLOPE also uses estimation of σ (almost) holds FDR and has the lowest MSE's of all methods.

More non-zero variables improves FDR of those methods, but also MSE increases.

Ex. 2

The task is to repeat previous experiment but with rows $X_i \sim N(0, \frac{1}{n}\Sigma)$, where $\Sigma_{ii} = 1$ and $\Sigma_{ij} = 0.5$ for $i \neq j$.

	$MSE(\beta)$	$MSE(\mu)$	FWER	FDR	power
OLS	36486.243	1803.638	-	-	-
ridge	453.402	206.620	-	-	-
LASSO	225.434	112.461	1.000	0.826	0.958
ridge_kf	497.043	1633.112	0.300	0.138	0.338
LASSO_kf	375.013	1078.811	0.350	0.156	0.358
ALASSO1	446.544	225.016	0.980	0.628	0.922
ALASSO2	203.098	110.738	0.950	0.373	0.822
ASLOPE	199.253	106.411	0.970	0.425	0.844

Table 2.1: Results for $k = 5$.

	$MSE(\beta)$	$MSE(\mu)$	FWER	FDR	power
OLS	36658.228	1814.806	-	-	-
ridge	1386.452	584.031	-	-	-
LASSO	600.285	277.389	1.000	0.717	0.985
ridge_kf	1943.187	22098.152	0.960	0.378	0.700
LASSO_kf	661.776	2926.246	0.910	0.221	0.844
ALASSO1	748.226	337.512	1.000	0.484	0.945
ALASSO2	654.814	302.291	1.000	0.263	0.886
ASLOPE	606.821	277.922	1.000	0.335	0.911

Table 2.2: Results for $k = 20$.

	$MSE(\beta)$	$MSE(\mu)$	FWER	FDR	power
OLS	37184.652	1801.736	-	-	-
ridge	3073.642	1274.958	-	-	-
LASSO	1232.902	493.939	1	0.605	0.989
ridge_kf	4819.604	130287.244	1	0.390	0.401
LASSO_kf	1514.382	12937.342	1	0.164	0.843
ALASSO1	1387.004	542.011	1	0.382	0.956
ALASSO2	1592.466	633.441	1	0.203	0.901
ASLOPE	1363.305	543.261	1	0.278	0.932

Table 2.3: Results for $k = 50$.

The results of those experiments are presented in tables 2.1, 2.2, 2.3. As expected, all methods are performing worse compared to the situation in previous task. Not all of them are able to hold FDR. Special construction of design matrix makes LASSO (and ridge in one case) hold FDR. More non-zero elements decreases the chance of false discovery.

Appendices

Ex. 1

Listing 1.1: Generating data and additional functions.

```
norm2 <- function(X,Y) sum((X-Y)^2)

generate_design <- function(n=500, p=450, correlated=FALSE, rho=.5) {
  if(!correlated)
    X = matrix(rnorm(n*p, 0, 1/sqrt(n)), n, p)
  else{
    Sigma = matrix(rho,p,p)
    diag(Sigma) = 1
    X = matrix(mvrnorm(n, rep(0,p), Sigma/n), n, p) #sqrt n?
  }
  return(X)
}
generate_response <- function(X, beta=10, nonzero=5, rdist=rnorm) {
  betas = c(rep(beta,nonzero), rep(0,ncol(X)-nonzero))
  X%*%betas + 2*rdist(nrow(X))
}

fdp <- function(tests, k, p) {
  d = sum(tests!=0)
  fd = sum(tests[(k+1):p]!=0)
  return(fd/max(d,1))
}

ifFalseDiscovery <- function(tests, k, p) {
  fd = sum(tests[(k+1):p]!=0)
  return(fd>0)
}
```

Listing 1.2: Knockoff selection.

```
W <- function(beta_hat, p=length(beta_hat)/2) abs(beta_hat[1:p])-abs(beta_
    hat[(p+1):(2*p)]) #LASSO coefficient difference stat

threshold_W <- function(w, q=.2) {
  ord = order(abs(w),decreasing=TRUE)
  fd = cumsum(w[ord]<0)
  nd = cumsum(w[ord]>0)
  fdr = (fd+1)/nd
  t_min = Inf
  if(sum(fdr<=q)>0){
    t_ind = ord[max(which(fdr<=q))]
    t_min = abs(w[t_ind])
  }
  return(t_min)
}

knockoff_select <- function(beta_hat, q=.2) {
  p = length(beta_hat)/2
  w = W(beta_hat)
  t_min = threshold_W(w, q)
  if(t_min!=Inf){
    sel_id = which(w > t_min) #>=?
```

```
    beta_hat[-sel_id] = 0
  }
  else
    beta_hat[1:p] = rep(0,p)
  return(beta_hat[1:p])
}
```

Listing 1.3: Adaptive LASSO 1.

```
adaptive_lasso1_select <- function(X, Y, beta_hat) {
  nonzero_id = which(beta_hat!=0)
  X_nonzero = X[,nonzero_id]
  beta_nonzero =  beta_hat[nonzero_id]

 W = 1/abs(beta_nonzero)
  X_nonzero = sweep(X_nonzero,2,W, '/') #*

  ad_lasso = cv.glmnet(X_nonzero, Y, intercept=FALSE, standardize=FALSE)
  beta_nonzero = rep(0, ncol(X))
  beta_nonzero[nonzero_id] = coef(ad_lasso, s='lambda.min')[-1,1] / W #*

  return(beta_nonzero)
}
```

Listing 1.4: Adaptive LASSO 2.

```
adaptive_lasso2_select <- function(X, Y, beta_hat, q=.2) {
  n = nrow(X); p = ncol(X); lambda_alasso = qnorm(1-q/2/p) #CONSTANTS

  RSS = sum((Y- X%*%beta_hat)^2)
  nonzero_id = which(beta_hat!=0)
  X_nonzero = X[,nonzero_id]
  beta_nonzero =  beta_hat[nonzero_id]
  l = length(nonzero_id)
  sigma_lassoCV = sqrt(RSS/(n-l))

 W = sigma_lassoCV/abs(beta_nonzero)
  X_nonzero = sweep(X_nonzero,2,W, '/')

  ad_lasso = glmnet(X_nonzero, Y, intercept=FALSE, standardize=FALSE,lambda
      =sigma_lassoCV*lambda_alasso/n)
  beta_nonzero = rep(0,p)
  beta_nonzero[nonzero_id] = coef(ad_lasso)[-1,1] / W
  return(beta_nonzero)
}
```

Listing 1.5: Adaptive SLOPE.

```
adaptive_slope_select <- function(X, Y, beta_hat, q=.2) {
  n = nrow(X); p = ncol(X); #CONSTANTS

  RSS = sum((Y- X%*%beta_hat)^2)
  nonzero_id = which(beta_hat!=0)
  l = length(nonzero_id)
  sigma_lassoCV = sqrt(RSS/(n-l))

 W = sigma_lassoCV/abs(beta_hat+0.000001) # beta_nonzero
  X = sweep(X,2,W, '/')

  ad_slope = SLOPE(X,Y,q=q, alpha=1/n*sigma_lassoCV, lambda='bh', solver='
      admm',max_passes=100, scale='none', intercept = FALSE)
```

```
  return(coef(ad_slope) / W)
}
```

Listing 1.6: Comparison.

```
ex1 <- function(n=500, p=450, beta=10, nonzero=c(5,20,50), rep=100, q=.2,
    correlated=FALSE, rho=.5) {
  OLS = matrix(0,rep,p); ridge = matrix(0,rep,p); LASSO = matrix(0,rep,p);
      ridge_kf = matrix(0,rep,p);
  LASSO_kf = matrix(0,rep,p); ALASSO1 = matrix(0,rep,p); ALASSO2 = matrix
      (0,rep,p); ASLOPE = matrix(0,rep,p);
  results_list = list()
  X = generate_design(n,p,correlated,rho)

  if(!correlated)
    X_aug = cbind(X, generate_design(n,p,correlated,rho))
  else{
    Sigma = matrix(rho,p,p)
    diag(Sigma) = 1
    s = min(eigen(Sigma)$values)
    s = min(2*s,1)
    sseq = rep(s,p)
    V = 2*diag(sseq)-diag(sseq)%*%solve(Sigma)%*%diag(sseq)
    mu = X-X%*%solve(Sigma)%*%diag(sseq)
    X_aug = cbind(X, mu+mvrnorm(n,rep(0,p),V)/sqrt(n))
  }

  for (j in 1:length(nonzero)) {
    k = nonzero[j]
    betas = c(rep(beta,k), rep(0,p-k))
    for (r in 1:rep) {
      Y = generate_response(X,beta,k)
      OLS[r,] = coef(lm(Y~X-1))
      ridge[r,] = coef(cv.glmnet(X,Y,alpha=0,intercept=FALSE, standardize=
          FALSE), s='lambda.min')[-1,1]
      LASSO[r,] = coef(cv.glmnet(X,Y,intercept=FALSE, standardize=FALSE), s
          ='lambda.min')[-1,1]

      ridge_aug = coef(cv.glmnet(X_aug,Y,alpha=0,intercept=FALSE,
          standardize=FALSE), s='lambda.min')[-1,1]
      LASSO_aug = coef(cv.glmnet(X_aug,Y,intercept=FALSE, standardize=FALSE
          ), s='lambda.min')[-1,1]
      ridge_kf[r,]  = knockoff_select(ridge_aug,q)
      LASSO_kf[r,]  = knockoff_select(LASSO_aug,q)

      ALASSO1[r,] = adaptive_lasso1_select(X,Y,LASSO[r,])
      ALASSO2[r,] = adaptive_lasso2_select(X,Y,LASSO[r,],q)
      ASLOPE[r,] = adaptive_slope_select(X,Y,LASSO[r,],q)
      cat("k= ",k,"rep= ", r, "\n")
    }

    methods_coef = list(OLS, ridge, LASSO, ridge_kf, LASSO_kf, ALASSO1,
        ALASSO2, ASLOPE)
    methods_names = c("OLS", "ridge", "LASSO", "ridge_kf", "LASSO_kf", "
        ALASSO1", "ALASSO2", "ASLOPE")
    methods_stats = matrix(-1, length(methods_coef), 5)
    colnames(methods_stats) = c("MSE_beta", "MSE_mu", "FWER", "FDR", "power
        ")
    rownames(methods_stats) = methods_names
```

```
    for (i in 1:length(methods_coef)) {
      coeff = methods_coef[[i]]
      methods_stats[i,1] = mean( apply(coeff, 1, function(beta_hat)norm2(
          beta_hat, betas)) ) #mse_beta
      methods_stats[i,2] = mean( apply(coeff, 1, function(beta_hat)norm2(X%
          *%beta_hat,X%*%betas)) ) #mse_mu

      if(!(methods_names[i] %in% c("OLS", "ridge") )){
        methods_stats[i,3] = mean( apply(coeff, 1, function(beta_hat)
            ifFalseDiscovery(beta_hat, k, p) )  ) #fwer
        methods_stats[i,4] = mean( apply(coeff, 1, function(beta_hat)fdp(
            beta_hat, k, p) )  ) #fdr
        methods_stats[i,5] = mean( apply(coeff, 1, function(beta_hat)(sum(
            beta_hat[1:k]!=0)/k)  ) ) #power
      }
    }

    results_list[[j]] = methods_stats
    save(list=c("methods_stats"), file=paste0("SL_lab3_ex1_nonzero",k,".
        RData"))
    printTable(methods_stats, paste("Results for k=",k), paste0("SL_lab3_
        ex1_nonzero",k))
  }

  names(results_list) = paste(nonzero)
  return(results_list)
}
```

Ex. 2

Listing 2.1: Comparison.

```
ex1(correlated = TRUE, rep=10)
```