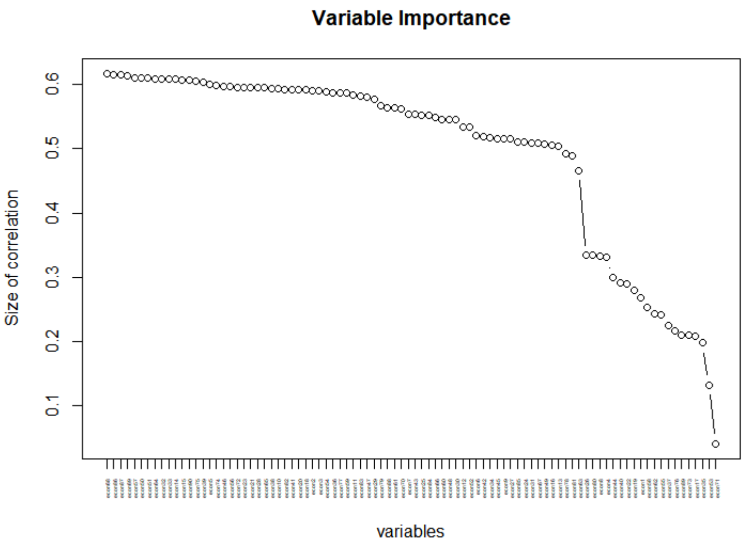
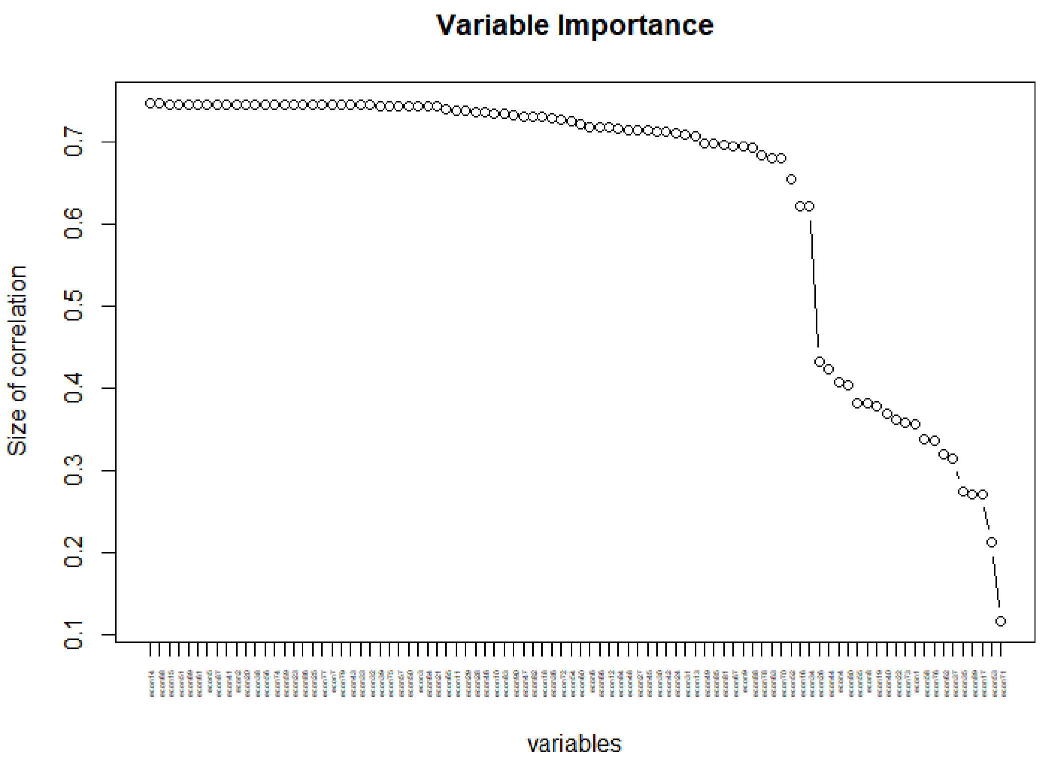
```
> ###########################################
> # Statistical Modelling & Machine Learning #
> #             R Example4                   #
> ###########################################
>
> options(warn = -1)  # Turn off warning message
>
> ######### Variable Importance: Regression problem ##########
>
> # Building data
> # 90 economic variables and sales variable (output)
> dat = read.table('building.csv', sep=',', header=T)
>
> # Correlation coefficient --------------------------
> VI = cor(dat)[,'price']
> SVI = sort(abs(VI), decreasing = T)[-1]
> SVI
    econ68     econ86     econ87     econ69     econ57     econ50     econ51
0.61701220 0.61541704 0.61473457 0.61350277 0.61131788 0.61128637 0.61118485
    econ64     econ32     econ33     econ14     econ15     econ90     econ75
0.60952319 0.60877521 0.60861991 0.60833342 0.60725207 0.60657069 0.60607852
    econ39      econ5     econ74     econ46     econ56     econ72     econ23
0.60453730 0.60115498 0.59873700 0.59763375 0.59725241 0.59584429 0.59554659
    econ21     econ28     econ65     econ38     econ10     econ82     econ41
0.59517838 0.59486492 0.59482798 0.59426435 0.59390329 0.59294998 0.59285014
    econ20     econ18      econ2      econ3     econ54     econ36     econ77
0.59162525 0.59154972 0.59073885 0.59021348 0.58960077 0.58789128 0.58763764
    econ59     econ11     econ83     econ47     econ29     econ79     econ88
0.58749289 0.58404324 0.58271133 0.58018114 0.57743482 0.56642843 0.56411320
    econ61     econ70      econ7     econ43     econ25     econ84     econ66
0.56404887 0.56210776 0.55472247 0.55350511 0.55298111 0.55210583 0.54885358
    econ60     econ48     econ30     econ12     econ52      econ6     econ42
0.54632685 0.54615987 0.54526351 0.53387115 0.53314601 0.52120916 0.51865266
    econ34     econ45      econ9     econ27     econ85     econ24     econ31
0.51656070 0.51511175 0.51504086 0.51490753 0.51001640 0.50990224 0.50903718
    econ67     econ49     econ16     econ13     econ78     econ81     econ63
0.50847776 0.50816509 0.50577267 0.50324426 0.49319746 0.48875751 0.46591618
    econ26     econ80      econ8      econ4     econ44     econ40     econ22
0.33372804 0.33361651 0.33291222 0.33024927 0.29966462 0.29119779 0.29012961
    econ19      econ1     econ58     econ62     econ55     econ37     econ76
0.27901459 0.26739827 0.25257770 0.24261738 0.24170153 0.22526512 0.21588431
    econ89     econ73     econ17     econ35     econ53     econ71
0.20988269 0.20928031 0.20824238 0.19716845 0.13182179 0.04036955
>
> plot(1:length(SVI),SVI, type='b', ylab='Size of correlation',
+     xlab ='variables', main='Variable Importance', xaxt='n')
> axis(side=1, at=1:length(SVI), labels=names(SVI), cex.axis=0.3,las=2)
```

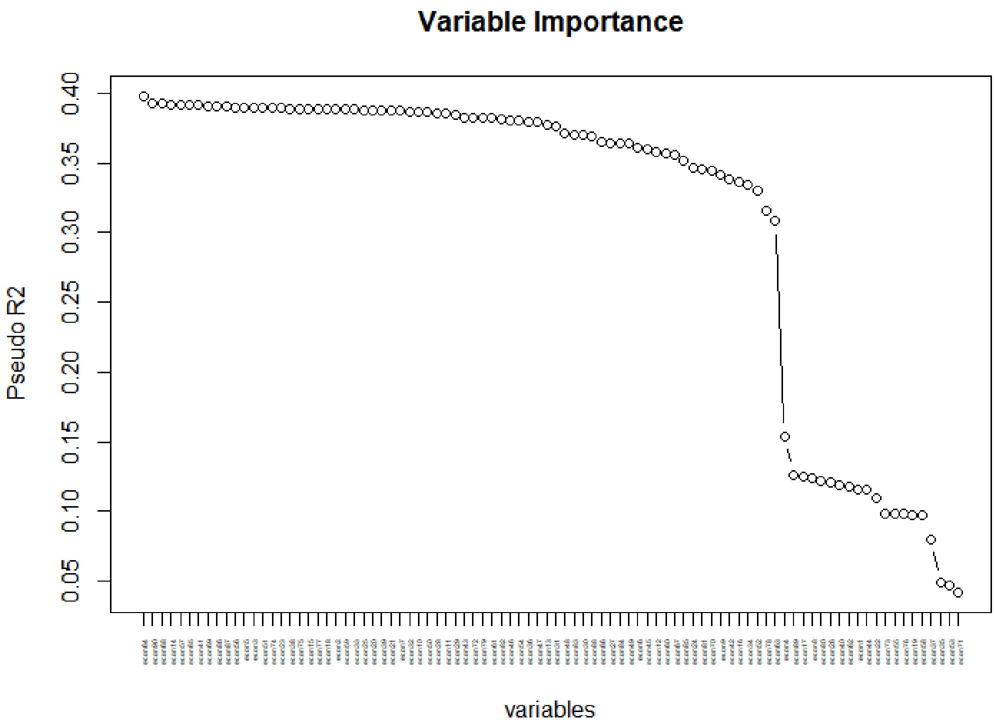


```
> # Spearman rank correlation coefficient -------------
> SP = cor(dat, method='spearman')[,'price']
> SPI = sort(abs(SP), decreasing = T)[-1]
> SPI
   econ14    econ68    econ15    econ51    econ69    econ61     econ5    econ87 
0.7480006 0.7478559 0.7470943 0.7470746 0.7470564 0.7470406 0.7470166 0.7470160 
   econ41     econ2    econ20    econ38    econ56    econ74    econ59    econ23 
0.7468722 0.7468707 0.7468707 0.7468707 0.7468707 0.7468707 0.7468269 0.7468192 
   econ86    econ25    econ77     econ7    econ79    econ43    econ33    econ32 
0.7466633 0.7466291 0.7465841 0.7463867 0.7463699 0.7463527 0.7461416 0.7458426 
   econ39    econ75    econ57    econ50     econ3    econ64    econ21    econ65 
0.7453043 0.7451003 0.7449772 0.7449110 0.7446253 0.7440360 0.7440314 0.7417415 
   econ11    econ29    econ28    econ46    econ10    econ83    econ90    econ47 
0.7385228 0.7382788 0.7371699 0.7365226 0.7360915 0.7346095 0.7331846 0.7322964 
   econ82    econ18    econ36    econ72    econ54    econ60     econ6    econ66 
0.7322654 0.7316920 0.7304120 0.7279187 0.7262652 0.7234224 0.7194596 0.7182935 
   econ12    econ84    econ48    econ27    econ45    econ30    econ42    econ24 
0.7182107 0.7169109 0.7161513 0.7159929 0.7152081 0.7142345 0.7136541 0.7114739 
   econ31    econ13    econ49    econ85    econ81    econ67     econ9    econ88 
0.7093104 0.7088550 0.6993933 0.6990538 0.6977191 0.6955061 0.6951678 0.6939592 
   econ78    econ63    econ70    econ52    econ16    econ34    econ26    econ44 
0.6842120 0.6815127 0.6812766 0.6544439 0.6224187 0.6220537 0.4331914 0.4234760 
    econ4    econ80    econ55     econ8    econ19    econ40    econ22    econ73 
0.4083398 0.4047152 0.3820735 0.3817524 0.3786940 0.3695891 0.3623909 0.3580080 
    econ1    econ58    econ76    econ62    econ37    econ35    econ89    econ17 
0.3566338 0.3392029 0.3360683 0.3204578 0.3156516 0.2742122 0.2716997 0.2713140 
   econ53    econ71 
0.2130441 0.1170807 
> 
> plot(1:length(SPI),SPI, type='b', ylab='Size of correlation',
+      xlab ='variables', main='Variable Importance', xaxt='n')
> axis(side=1, at=1:length(SPI), labels=names(SPI), cex.axis=0.3,las=2)
```

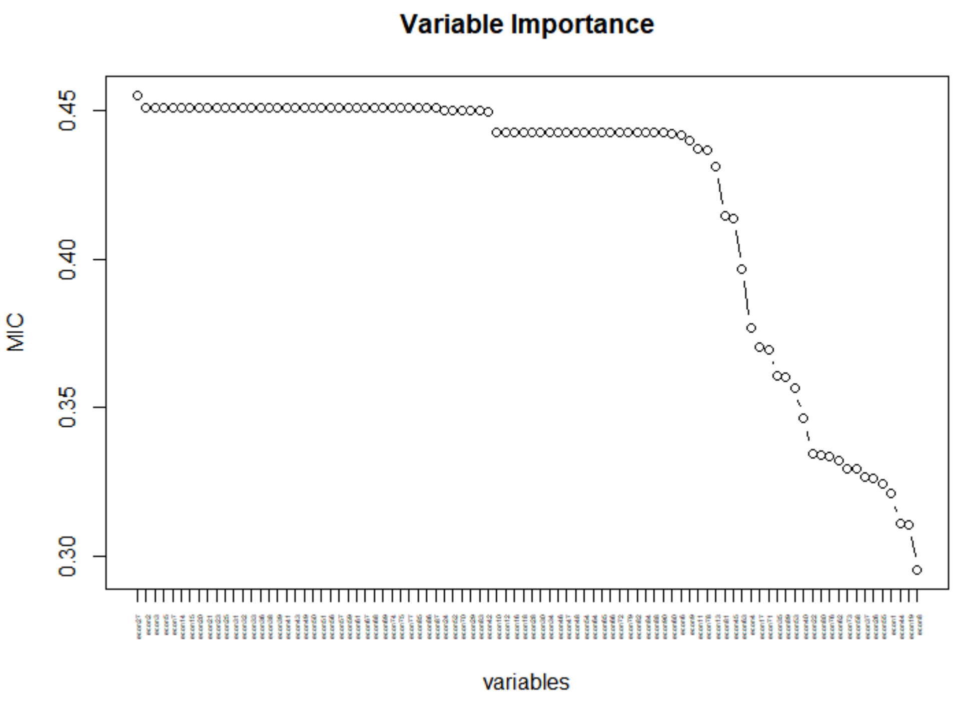



```
> # Pseudo R^2 ----------------------------------------
> p = 90
> PR2 = numeric(p)
> names(PR2) = colnames(dat[,-91])
> for (j in 1:p)
```

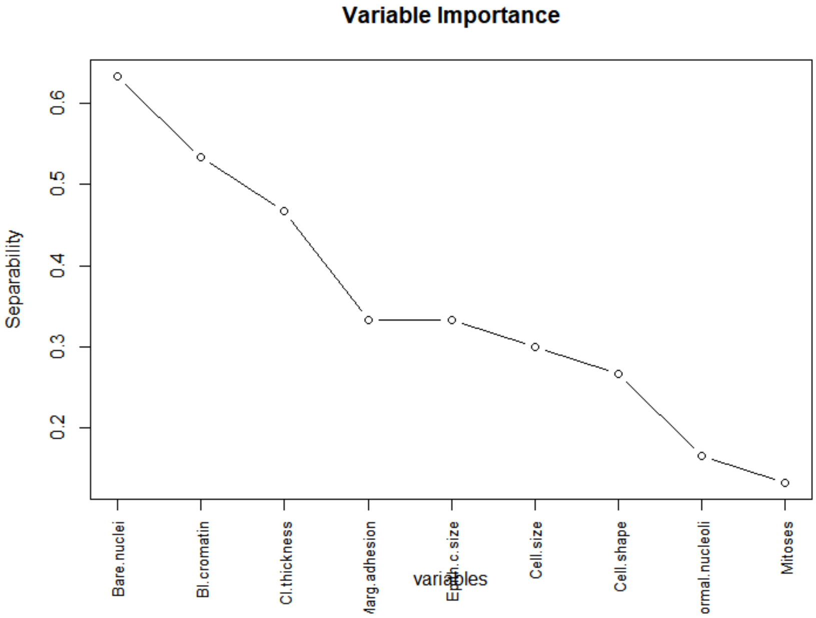
```
+ {
+   fit = loess(price ~ dat[,j], data=dat)  # Local linear regression
+   yhat = predict(fit, dat[,j])
+   PR2[j] = 1-(sum((dat$price - yhat)^2)/sum((dat$price - mean(dat$price))^2))
+ }
>
> SPR2 = sort(PR2, decreasing = T)
> SPR2
    econ64     econ90     econ68     econ14     econ57     econ65     econ41
0.39782881 0.39311868 0.39293505 0.39187266 0.39184300 0.39143906 0.39129056
    econ69     econ86     econ87     econ56      econ5      econ3     econ51
0.39116206 0.39103442 0.39027272 0.38992729 0.38985100 0.38983392 0.38954506
    econ74     econ23     econ38     econ75     econ15     econ77     econ18
0.38946368 0.38928179 0.38909528 0.38900633 0.38894807 0.38892545 0.38878446
     econ2     econ59     econ33     econ25     econ20     econ39     econ21
0.38876378 0.38865585 0.38817614 0.38810713 0.38806374 0.38804960 0.38764551
     econ7     econ32     econ10     econ50     econ28     econ11     econ29
0.38751223 0.38704344 0.38689628 0.38613038 0.38563787 0.38514273 0.38406873
    econ43     econ72     econ79     econ61     econ82     econ46     econ54
0.38303032 0.38278444 0.38257175 0.38232618 0.38100270 0.38049542 0.38002283
    econ36     econ47     econ13     econ31     econ48     econ83     econ30
0.37981754 0.37924352 0.37712482 0.37679239 0.37171271 0.37038657 0.37004289
    econ88     econ66     econ27     econ84     econ49      econ6     econ45
0.36955338 0.36535822 0.36434124 0.36424892 0.36378332 0.36095139 0.35953716
    econ12     econ60     econ67     econ85     econ24     econ81     econ70
0.35780713 0.35636359 0.35548371 0.35142339 0.34675822 0.34543402 0.34496508
     econ9     econ42     econ16     econ34     econ52     econ78     econ63
0.34106047 0.33880395 0.33584800 0.33398428 0.33018205 0.31556932 0.30832680
     econ4     econ89     econ17      econ8     econ80     econ26     econ40
0.15375108 0.12623539 0.12503377 0.12394045 0.12216669 0.12037664 0.11881972
    econ62      econ1     econ44     econ22     econ73     econ55     econ76
0.11775807 0.11559373 0.11535676 0.10943468 0.09876735 0.09865478 0.09847815
    econ19     econ58     econ37     econ35     econ53     econ71
0.09743884 0.09718496 0.07970306 0.04856789 0.04745673 0.04211173
>
> plot(1:length(SPR2),SPR2, type='b', ylab='Pseudo R2',
+      xlab ='variables', main='Variable Importance', xaxt='n')
> axis(side=1, at=1:length(SPR2), labels=names(SPR2), cex.axis=0.3,las=2)
```



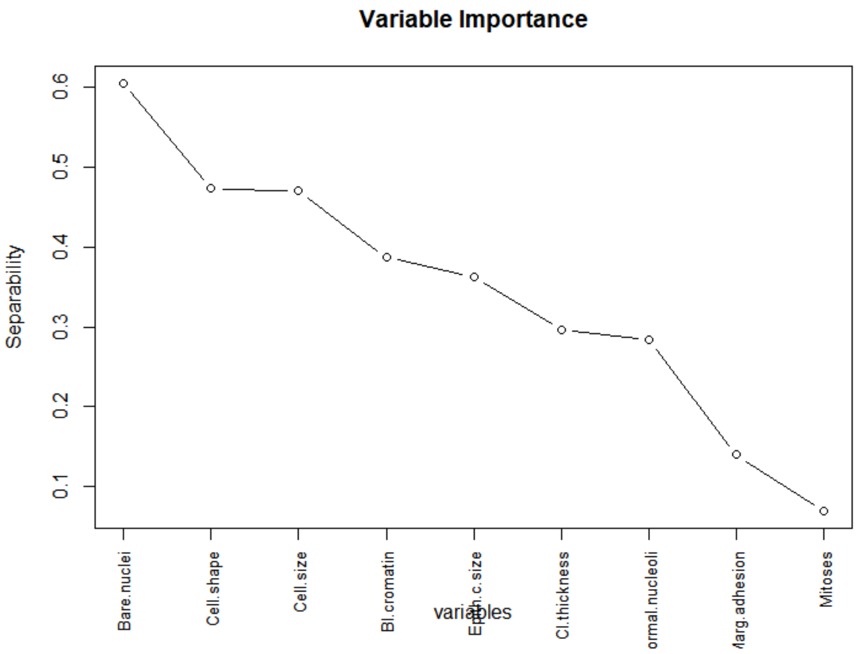
```
> # Maximal information coefficient (MIC) -------------
>
> install.packages('minerva')
> library(minerva)
>
> MIC = mine(dat)
> MIC = MIC$MIC[,'price']
>
> SMIC = sort(MIC, decreasing = T)[-1]
> SMIC
   econ27     econ2     econ3     econ5     econ7    econ14    econ15    econ20
0.4550555 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285
   econ21    econ23    econ25    econ31    econ32    econ33    econ36    econ38
0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285
   econ39    econ41    econ43    econ49    econ50    econ51    econ56    econ57
0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285
   econ59    econ61    econ67    econ68    econ69    econ74    econ75    econ77
0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285 0.4508285
   econ85    econ86    econ87    econ24    econ52    econ70    econ29    econ83
0.4508285 0.4508285 0.4508285 0.4502464 0.4502464 0.4502464 0.4499103 0.4499103
   econ42    econ10    econ12    econ16    econ18    econ28    econ30    econ34
0.4497651 0.4424887 0.4424887 0.4424887 0.4424887 0.4424887 0.4424887 0.4424887
   econ46    econ47    econ48    econ54    econ64    econ65    econ66    econ72
0.4424887 0.4424887 0.4424887 0.4424887 0.4424887 0.4424887 0.4424887 0.4424887
   econ79    econ82    econ84    econ88    econ90    econ60     econ6     econ9
0.4424887 0.4424887 0.4424887 0.4424887 0.4424887 0.4423206 0.4416032 0.4398774
   econ11    econ78    econ13    econ81    econ45    econ63     econ4    econ17
0.4371542 0.4367981 0.4311837 0.4144450 0.4136798 0.3965958 0.3769089 0.3703032
   econ71    econ35    econ89    econ53    econ40    econ22    econ80    econ76
0.3696456 0.3607008 0.3601444 0.3566271 0.3462738 0.3345459 0.3338895 0.3334517
   econ62    econ73    econ58    econ37    econ26    econ55     econ1    econ44
0.3322830 0.3294336 0.3292661 0.3265118 0.3262451 0.3244009 0.3209021 0.3109958
   econ19     econ8
0.3106083 0.2953973
>
> plot(1:length(SMIC),SMIC, type='b', ylab='MIC',
+      xlab ='variables', main='Variable Importance', xaxt='n')
> axis(side=1, at=1:length(SMIC), labels=names(SMIC), cex.axis=0.3,las=2)
```



```
> ######### Variable Importance: Classification problem ##########
>
> # Data
> install.packages('mlbench')
> library(mlbench)
> data(BreastCancer)
> dat = BreastCancer[,-1]
>
> # Relief algorithm ----------------------------
>
> install.packages('CORElearn')
> library(CORElearn)
>
> # Relief algorithm
> RE = attrEval(Class ~ ., data=dat, estimator='Relief',
+              ReliefIterations=30)
>
> SRE = sort(RE, decreasing = T)
> SRE
  Bare.nuclei    Bl.cromatin   Cl.thickness   Marg.adhesion    Epith.c.size
    0.6330396      0.5333333      0.4666667       0.3333333       0.3333333
    Cell.size     Cell.shape Normal.nucleoli        Mitoses
    0.3000000      0.2666667      0.1666667       0.1333333
>
> plot(1:length(SRE),SRE, type='b', ylab='Separability',
+     xlab ='variables', main='Variable Importance', xaxt='n')
> axis(side=1, at=1:length(SRE), labels=names(SRE), cex.axis=0.8,las=2)
```




```
> # ReliefF algorithm
> REF = attrEval(Class ~ ., data=dat, estimator='ReliefFequalK',
+               ReliefIterations=30)
>
> SREF = sort(REF, decreasing = T)
> SREF
  Bare.nuclei     Cell.shape      Cell.size     Bl.cromatin    Epith.c.size
    0.6047962      0.4733333      0.4700000       0.3866667       0.3633333
 Cl.thickness Normal.nucleoli  Marg.adhesion        Mitoses
    0.2966667      0.2833333      0.1400000       0.0700000
>
> plot(1:length(SREF),SREF, type='b', ylab='Separability',
+     xlab ='variables', main='Variable Importance', xaxt='n')
> axis(side=1, at=1:length(SREF), labels=names(SREF), cex.axis=0.8,las=2)
```




```
> ########## Variable Selection: Simulated Annealing ##########
>
> install.packages('mvtnorm')
> library(mvtnorm)
>
> # Data generation
> set.seed(10)
>
> n = 500
> p = 20
> S = matrix(0.3, nrow=p, ncol=p)
> diag(S) = 1
> X = rmvnorm(n, mean=rep(0,p), sigma=S)
>
> XN = NULL
> for (j in 1:p) XN = c(XN,paste('X',j,sep=''))
> colnames(X) = XN
>
> Y = 2 + 0.5*X[,1] - 0.3*X[,2] + 1.2*X[,3] + rnorm(n,sd=0.1)
>
> # Simulated Annealing ---------------------------
>
> install.packages('caret')
> library(caret)
>
> ctrl = safsControl(functions=caretSA, method='cv', number=5)
>
> obj = safs(x=X, y=Y, iters=20, safsControl=ctrl, method='lm')
> obj

Simulated Annealing Feature Selection

500 samples
20 predictors

Maximum search iterations: 20

Internal performance values: RMSE, Rsquared, MAE
Subset selection driven to minimize internal RMSE

External performance values: RMSE, Rsquared, MAE
Best iteration chose by minimizing external RMSE
External resampling method: Cross-Validated (5 fold)
```

```
During resampling, no variables were selected.

In the final search using the entire training set:
   * 14 features selected at iteration 20 including:
     X1 ...
   * external performance at this iteration is

       RMSE   Rsquared        MAE
     0.4314     0.8335     0.3257

> #################### ISIS ####################
>
> install.packages('SIS')
Error in install.packages : Updating loaded packages
> library(SIS)
>
> ?SIS
>
> # Data generation
> set.seed(0)
> n = 400; p = 50; rho = 0.5
> corrmat = diag(rep(1-rho, p)) + matrix(rho, p, p)
> corrmat[,4] = sqrt(rho)
> corrmat[4, ] = sqrt(rho)
> corrmat[4,4] = 1
> corrmat[,5] = 0
> corrmat[5, ] = 0
> corrmat[5,5] = 1
> cholmat = chol(corrmat)
> x = matrix(rnorm(n*p, mean=0, sd=1), n, p)
> x = x%*%cholmat
>
> # Linear regression
> set.seed(1)
> b = c(4,4,4,-6*sqrt(2),4/3)
> y=x[, 1:5]%*%b + rnorm(n)
>
>
> # ISIS with regularization
> model11=SIS(x, y, family='gaussian', tune='bic')
Iter 1 , screening:  1 2 3 4 5 6 7 8 9 10 11 18 21 22 24 25 26 29 30 31 32 33 34 35 36
41 42 43 44 46 47 48 50
Iter 1 , selection:  1 2 3 4 5
Iter 1 , conditional-screening:  6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 2 , screening:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 2 , selection:  1 2 3 4 5
Model already selected
> model11$ix
[1] 1 2 3 4 5
>
> model12=SIS(x, y, family='gaussian', tune='bic', varISIS='aggr', seed=11)
Iter 1 , screening:  1 2 3 5 6 7 9 10 20 23 24 27 28 29 38 40 41 42 43 45 47 48
Iter 1 , selection:  1 2 3 5 6 7 9 10 20 23 24 27 28 29 38 40 41 42 43 45 47 48
Iter 1 , conditional-screening:  4 8 11 12 13 14 15 16 17 18 19 21 22 25 26 30 31 32 33
34 35 36 37 39 44 46 49 50
Iter 2 , screening:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 2 , selection:  1 2 3 4 5
Iter 2 , conditional-screening:  6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 3 , screening:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 3 , selection:  1 2 3 4 5
Model already selected
> model12$ix
[1] 1 2 3 4 5
```

```
>
> # logistic regression
> set.seed(2)
> feta = x[, 1:5]%*%b; fprob = exp(feta)/(1+exp(feta))
> y = rbinom(n, 1, fprob)
>
> # ISIS with regularization
> model21=SIS(x, y, family='binomial', tune='bic', penalty='SCAD', perm=T, q=0.9)
Iter 1 , screening:  1 2 3 5 29
Iter 1 , selection:  1 2 3 5 29
Iter 1 , conditional-screening:  4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 2 , screening:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 2 , selection:  1 2 3 4 5
Iter 2 , conditional-screening:  7 11 17 26 27 41 46
Iter 3 , screening:  1 2 3 4 5 7 11 17 26 27 41 46
Iter 3 , selection:  1 2 3 4 5
Model already selected
> model21$ix
[1] 1 2 3 4 5
>
> model22=SIS(x, y, family='binomial', tune='bic', varISIS='aggr', seed=21)
Iter 1 , screening:  1 2 3 5 8 9 12 16 21 24 25 26 28 29 31 35 38 39 42 45 49 50
Iter 1 , selection:  1 2 3 5 8 9 12 21 24 25 26 28 31 35 38 39 50
Iter 1 , conditional-screening:  4 6 7 10 11 13 14 15 16 17 18 19 20 22 23 27 29 30 32
33 34 36 37 40 41 42 43 44 45 46 47 48 49
Iter 2 , screening:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 2 , selection:  1 2 3 4 5
Iter 2 , conditional-screening:  6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 3 , screening:  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
Iter 3 , selection:  1 2 3 4 5
Model already selected
> model22$ix
[1] 1 2 3 4 5
```