```
#================= Simulated Metabolomics Biomarkers R Code ================================
#Program description: R code to perform feature selection on simulated metabolomics data in two sections. First section
#uses LASSO and Elastic Net and second one uses VIP and Student t feature selection methods. The BioMark package
# simulates data according to different study parameters (N,P,Q,delta,correlation structures). The four feature selection
#methods are used on each simulated data set and the pAROC is calculated. Output is two csv files containing the
# pAROC values according to different parameters of the simulation, one for each section.
#
#Running first half of code to generate Elastic Net and Lasso results takes about 5.6 minutes per simulation on my Dell
#with a 64-bit processor and an Intel(R) Core(TM) i7-3770OS CPU @ 3.10 GHz with 8.00 GM RAM. Running the
#second half for the Student t and VIP takes about 1hour per simulation, primarily because of the resampling to obtain p-
values for the VIP method.

#Authors:  written by Danny Lu, with edits by Karen Kopciuk, Calgary

# Project name: Simulated Metabolomics Biomarkers R Code
# Project home page: https://people.ucalgary.ca/~kakopciu
# Operating system(s): Unix (including FreeBSD and Linux),Windows, MacOS
# Programming language: R 3.0 and higher
# Other requirements: R contributed packages BioMark (will load glmnet
# License: GNU GPL
# Any restrictions to use by non-academics: none

# ----------Setting directory, installing packages ----------#
# set working directory where source code can be found and where output will be placed
# setwd("C://Documents and Settings")

# can run program by sourcing code or by copying and pasting code directly into R
# source("C:\\Documents and Settings\\SimulationRcode.txt")
# install 2 contributed packages from CRAN
install.packages("BioMark")
#then attach library or just attach if packages already installed in a local repository
library(BioMark)

# remove ALL objects and functions to ensure only new results and variables being captured
# CAUTION – will remove everything in directory !
rm(list=ls())

#----------      setting options and parameters for simulation     ----------#
# Setting the minimal fraction of times a variable should be in the top list to be considered as a potential biomarker.
biom.options(min.present=0.5)
#Setting the number of "top" coefficients taken into account in stability-based biomarker identification
biom.options(ntop = 10)
biom.options(lasso=list(lambda.min.ratio = 0.01, lambda = seq(0.003,0.3, 0.0025), alpha = c(0.5,1)) )

seed<-set.seed(94783)
Numbermets<-c(50,200,1000)  # Number of features, P
Ncontrol<-c(25,50)          # Number of observations N/2 in each of the two classes(control/treated)
Numbersims<-5 #200          # Number of simulations
delta<-c(0.2,0.4,0.8)       # Mean differences between groups for biomarkers
alp<-c(0.5,1)               # alpha parameter used in Elastic Net
rho<- 0.5                   # use this parameter for AR(1) correlation setting


startT<-Sys.time() # can use to estimate total run time for large number of simulations
```

```
#================  ENet and LASSO  ==========================================#
mm<-matrix(ncol=6+Numbersims)    #creating an empty matrix to store pAUC values

#-------------  independent correlation structure ----------------------------------------------------------------------#
for(l in Ncontrol){     # Loops through different sample sizes of N
          for(i in Numbermets){        # Loops through different sizes for P
               NumberSignmets<-as.integer(c(i*0.1,i*0.2,i*0.3))   #Percentage of significant mets, Q, 10%,20%,and 30% of
#the total number of mets
               # Creating independent correlation matrix
                    mycorind <- matrix(0, i, i)
                    diag(mycorind) <-  1
               for( j in NumberSignmets){     # Loops through different sizes for Q
                    for(k in delta){           # Loops through different values for delta
                         #Creating empty vectors for pAROC values
                           lassovec<-c()
                           enet1<-c()
                           enet2<-c()
                              simdata <- gen.data(ncontrol = l, nvar = i, nbiom = j,cormat = mycorind,nsimul =
Numbersims,means=rep(0,i),group.diff = k)
                                   for(a in alp) {
            biom.options(lasso=list(lambda.min.ratio = 0.01, lambda = seq(0.003,0.3, 0.0025), alpha = a))
                                        for(m in 1:Numbersims) {
set.seed(94783+m)
                                             studt.coef <- get.biom(X = simdata$X[,,m], Y = simdata$Y, fmethod
="lasso",ncomp=2, scale.p="none", type = "stab")
                                        # Checking lambdas until q mets are selected
                                             for(e in 1:length(studt.coef$info$lasso$lambda)){
                                                  if(length(studt.coef$lasso[[e]]$biom.indices)-j >= 0){
                                                            fracsel<-studt.coef$lasso[[e]]$fraction.selected
                                                            break
                                                   }
                                                  else fracsel<-
studt.coef$lasso[[length(studt.coef$info$lasso$lambda)]]$fraction.selected
                                                   }
                                        # Creating ROC cuves from the two methods
                                             true.biom <- (1:ncol(simdata$X[,,m]) %in% c(1:j))
                                             vip.roc<-ROC(fracsel,true.biom)
                                             #placing the pAROC(FDR 0.2) values into the vectors
                                                  if(a == 0.5){
                                                            enet1<-append(enet1,AUC(vip.roc,max.mspec=0.2))
                                                   }
                                                  if(a == 1){
                                                            lassovec<-append(lassovec,AUC(vip.roc,max.mspec=0.2))
                                                   }
                                        }#end m loop
                                   }#end a loop)
                         # Creating a vector with the parameters and the pAROC values
                         v0<-c('enet alpha=0.5',l*2,i,j,k,"Independent",enet1)
                         v1<-c('Lasso',l*2,i,j,k,"Independent",lassovec)
                         # Adding the vector to the matrix
                         mm<-rbind(mm,v0,v1)
                    } # end k loop
               } #end j loop
          } #end i loop
  } #end l loop
```

```
#------------  Low correlation  ----------------------------------------------------------------------------------------------------#
for(l in Ncontrol) { #Loops through different sample sizes of N
        for(i in Numbermets) { # Loops through different sizes for P
            NumberSignmets<-as.integer(c(i*0.1,i*0.2,i*0.3))  #Percentage of significant mets, Q, 10%,20%,and 30% of
the total number of mets
                for( j in NumberSignmets) { # Loops through different sizes for Q
                    for(k in delta) { # Loops through different values for delta
                        #Creating empty vectors for pAROC values
                        lassovec<-c()
                        enet1<-c()
                        enet2<-c()
                        mycovlowcor <- matrix(0, i, i)
                        mycovlowcor[row(mycovlowcor) <= j & col(mycovlowcor) <= j] <- 0.7
                        mycovlowcor[row(mycovlowcor) > j & col(mycovlowcor) >j] <- 0.1
                        diag(mycovlowcor) <- 1
                            simdata <- gen.data(ncontrol = l, nvar = i, nbiom = j,cormat = mycovlowcor,nsimul =
Numbersims,means=rep(0,i),group.diff = k)
                                for(a in alp){
biom.options(lasso=list(lambda.min.ratio = 0.01, lambda = seq(0.003,0.3, 0.0025), alpha = a))
                                    for(m in 1:Numbersims){
set.seed(94783+m)
studt.coef <- get.biom(X = simdata$X[,,m], Y = simdata$Y, fmethod ="lasso",ncomp=2, scale.p="none", type = "stab")
                                        #checking lambdas until q mets are selected
                                    for(e in 1:length(studt.coef$info$lasso$lambda)){
                                        if(length(studt.coef$lasso[[e]]$biom.indices)-j >= 0){
                                            fracsel<-studt.coef$lasso[[e]]$fraction.selected
                                                break
                                        }
                                        else fracsel<-
studt.coef$lasso[[length(studt.coef$info$lasso$lambda)]]$fraction.selected
                                        }
                                #Creating ROC cuves from the two methods
                                    true.biom <- (1:ncol(simdata$X[,,m]) %in% c(1:j))
                                    vip.roc<-ROC(fracsel,true.biom)
                                    #placing the pAROC(FDR 0.2) values into the vectors
                                        if(a == 0.5){
                                            enet1<-append(enet1,AUC(vip.roc,max.mspec=0.2))
                                                }
                                        if(a == 1){
                                            lassovec<-append(lassovec,AUC(vip.roc,max.mspec=0.2))
                                         }
                                    }     #end m loop
                                 }  #end a loop)
                            # Creating a vector with the parameters and the pAROC values
                            v2<-c('enet alpha=0.5',l*2,i,j,k,"Low Correlation",enet1)
                            v3<-c('Lasso',l*2,i,j,k,"Low Correlation",lassovec)
                            # Adding the vector to the matrix
                            mm<-rbind(mm,v2,v3)
                    } # end k loop
                } #end j loop
            } #end i loop
    } #end l loop


# ------------  AR(1) structure  ----------------------------------------------------------------------------------------------------#
```

```
for(l in Ncontrol) { # Loops through different sample sizes of N
        for(i in Numbermets){# Loops through different sizes for P
            NumberSignmets<-as.integer(c(i*0.1,i*0.2,i*0.3)) #Percentage of significant mets, Q, 10%,20%,and 30% of
the total number of mets
                for( j in NumberSignmets) { # Loops through different sizes for Q
                    for(k in delta) { # Loops through different values for delta
                        #creating empty vectors for pAROC values
                            lassovec<-c()
                            enet1<-c()
                            enet2<-c()
                        #Creating AR(1) correlation matrix
                            mycovAR <- rho^(as.matrix(dist(1:i)))
                            diag(mycovAR) <- 1
                                simdata <- gen.data(ncontrol = l, nvar = i, nbiom = j,cormat = mycovAR,nsimul =
Numbersims,means=rep(0,i),group.diff = k)
                                    for(a in alp){
biom.options(lasso=list(lambda.min.ratio = 0.01, lambda = seq(0.003,0.3, 0.0025), alpha = a))
                                        for(m in 1:Numbersims){
set.seed(94783+m)
  studt.coef <- get.biom(X = simdata$X[,m], Y = simdata$Y, fmethod ="lasso",ncomp=2, scale.p="none", type = "stab")
                                            # Checking lambdas until q mets are selected
                                            for(e in 1:length(studt.coef$info$lasso$lambda)){
                                                if(length(studt.coef$lasso[[e]]$biom.indices)-j >= 0){
                                                    fracsel<-studt.coef$lasso[[e]]$fraction.selected
                                                    break
                                                }
                                                else fracsel<-
studt.coef$lasso[[length(studt.coef$info$lasso$lambda)]]$fraction.selected
                                                }
                                    # Creating ROC cuves from the two methods
                                        true.biom <- (1:ncol(simdata$X[,m]) %in% c(1:j))
                                        vip.roc<-ROC(fracsel,true.biom)
                                        # Placing the pAROC(FDR 0.2) values into the vectors
                                            if(a == 0.5){
                                                enet1<-append(enet1,AUC(vip.roc,max.mspec=0.2))
                                                    }
                                            if(a == 1){
                                                lassovec<-append(lassovec,AUC(vip.roc,max.mspec=0.2))
                                             }
                                        }     #end m loop
                                     }  #end a loop)
                                # Creating a vector with the parameters and the pAROC values
                                v4<-c('enet alpha=0.5',l*2,i,j,k,"AR(1)",enet1)
                                v5<-c('Lasso',l*2,i,j,k,"AR(1)",lassovec)
                                # Adding the vector to the matrix
                                mm<-rbind(mm,v4,v5)
                    } # end k loop
                } #end j loop
            } #end i loop
  } #end l loop

#----------------------     create output .csv file  for  ENet and LASSO Results----------------------------------------
ff<-data.frame(mm[-1,])
#  Naming the columns. After sigma, each column represents the pAROC values estimated in each simulation
names<-c('method','n','p','q','delta','sigma',rep(1:Numbersims))
colnames(ff)<-names
```

```
#   Writing output to csv file
  write.csv(ff,"lassoelasticNet.csv")
Sys.time() - startT

#========================== VIP and Student t ========================================== #
startT<-Sys.time()
# remove storage matrices with same names as in previous part
rm("ff")
rm("mm")
mm<-matrix(ncol=6+Numbersims)#creating an empty matrix

#-------------  independent correlation structure ---------------------------------------------------------------------------------#
for(l in Ncontrol) { # Loops through different sample sizes of N
          for(i in Numbermets) { # Loops through different sizes for P
               NumberSignmets<-as.integer(c(i*0.1,i*0.2,i*0.3)) # Percentage of significant mets, Q, 10%,20%,and 30% of
the total number of mets
               # Creating independent correlation matrix
                    mycorind <- matrix(0, i, i)
                    diag(mycorind) <-  1
               for( j in NumberSignmets){# Loops through different sizes for  Q
                    for(k in delta){# Loops through different values for delta
                              # Creating empty vectors for pAROC values
                                   studtvec<-c()
                                   vipvec<-c()
                                   studtHCvec<-c()
                                   vipHCvec<-c()
                                   studtHCadjvec<-c()
                                   vipHCadjvec<-c()
                    for(m in 1:Numbersims){# Loops through all simulated datasets
set.seed(94783+m)
                                        # Simulating metabolomics data with different parameters in each loop
                                          simdata <- gen.data(ncontrol = l, nvar = i, nbiom = j,cormat = mycorind,
                                                nsimul = Numbersims,means=rep(0,i),group.diff = k)
                                        # Using Student T and VIP methods on the simulated data
                                          studt.coef <- get.biom(X = simdata$X[„m], Y = simdata$Y, fmethod
=c("studentt","vip") ,ncomp=2, scale.p="none", type = "stab")
                                        # Creating a vector of true/false, where true represents the true biomarkers
                                          true.biom <- (1:ncol(simdata$X[,,m]) %in% c(1:j))

# now get pvalues and adjusted pvalues for Student t and vip using HC
studt.HC <- get.biom(X = simdata$X[,,m], Y = simdata$Y, fmethod =c("studentt","vip") ,ncomp=2, scale.p="none", type
= "HC")
                                        # Creating ROC cuves from the two methods
                              vip.roc<-ROC(studt.coef$vip$`2`$fraction.selected,true.biom)
                              stab.roc <- ROC(studt.coef$studentt[[1]]$fraction.selected,true.biom)

                              vipHC.roc<-ROC( (1/(studt.HC$ vip [[1]]$pvals)), true.biom)
                              vipHCadj.roc<-ROC( (1/(p.adjust(studt.HC$ vip [[1]]$pvals, method="fdr"))) ,true.biom)
                              studtHC.roc <- ROC((1/(studt.HC$ studentt [[1]]$pvals)) ,true.biom)
                              studtHCadj.roc <- ROC((1/(p.adjust(studt.HC$ studentt [[1]]$pvals, method="fdr"))) ,true.biom)
                                        # Placing the pAROC(FDR 0.2) values into the vectors
                                          studtvec<-append(studtvec,AUC(stab.roc, max.mspec = .2))
                                          vipvec<-append(vipvec,AUC(vip.roc, max.mspec = .2))
                                          studtHCvec<-append(studtHCvec, AUC(studtHC.roc, max.mspec = .2))
                                          vipHCvec<-append(vipHCvec, AUC(vipHC.roc, max.mspec = .2))
                                          studtHCadjvec<-append(studtHCadjvec,AUC(studtHCadj.roc, max.mspec = .2))
```

```
                                        vipHCadjvec<-append(vipHCadjvec,AUC(vipHCadj.roc,  max.mspec = .2))
                }    #end m loop

                        # Creating a vector with the parameters and the pAROC values
                        va2<-c('Student T',l*2,i,j,k,"Independent",studtvec)
                        va3<-c('vip',l*2,i,j,k,"Independent",vipvec)
                        vb1<-c('Student T HC',l*2,i,j,k,"Independent",studtHCvec)
                        vb2<-c('Student T HC adj',l*2,i,j,k,"Independent",studtHCadjvec)
                        vb3<-c('vip HC',l*2,i,j,k,"Independent",vipHCvec)
                        vb4<-c('vip HC adj',l*2,i,j,k,"Independent",vipHCadjvec)
                        # Adding the vector to the matrix
                        mm<-rbind(mm,va2,va3,vb1,vb2,vb3,vb4)
                    } # end k loop
                } #end j loop
            } #end i loop
        } #end l loop

#-------------  Low correlation  ----------------------------------------------------------------------------------------------------#
for(l in Ncontrol){# Loops through different sample sizes
        for(i in Numbermets){# Loops through different sizes for p
                NumberSignmets<-as.integer(c(i*0.1,i*0.2,i*0.3))# Percentage of significant mets, 10%,20%,and 30% of the total number of mets
                for( j in NumberSignmets){# Loops through different sizes for q
                        for(k in delta){# Loops through different values for delta
                        # Creating empty vectors for pAROC values
                                studtvec<-c()
                                vipvec<-c()
                                studtHCvec<-c()
                                vipHCvec<-c()
                                studtHCadjvec<-c()
                                vipHCadjvec<-c()
                        # Creating low correlation matrix
                        mycovlowcor <- matrix(0, i, i)
                        mycovlowcor[row(mycovlowcor) <= j & col(mycovlowcor) <= j] <- 0.7
                        mycovlowcor[row(mycovlowcor) > j & col(mycovlowcor) >j] <- 0.1
                        diag(mycovlowcor) <- 1
                            for(m in 1:Numbersims){#loops through all simulated datasets
set.seed(94783+m)
                                #Simulating metabolomics data with different parameters in each loop
                                    simdata <- gen.data(ncontrol = l, nvar = i, nbiom = j,cormat = mycovlowcor, nsimul = Numbersims,means=rep(0,i),group.diff = k)
                                # Using Student T and VIP methods on the simulated data
                                    studt.coef <- get.biom(X = simdata$X[,,m], Y = simdata$Y, fmethod =c("studentt","vip") ,ncomp=2, scale.p="none", type = "stab"  )
                                # Creating a vector of true/false, where true represents the true biomarkers
                                    true.biom <- (1:ncol(simdata$X[,,m]) %in% c(1:j))
                                # Creating ROC cuves from the two methods
                                    vip.roc<-ROC(studt.coef$vip$`2`$fraction.selected,true.biom)
                                    stab.roc <- ROC(studt.coef$studentt[[1]]$fraction.selected, true.biom)

# now get pvalues and adjusted pvalues for student t and vip using HC
                                  studt.HC <- get.biom(X = simdata$X[,,m], Y = simdata$Y, fmethod =c("studentt","vip") ,ncomp=2, scale.p="none", type = "HC")
                                # Creating ROC cuves from the two methods
                                  vipHC.roc<-ROC( (1/(studt.HC$ vip [[1]]$pvals)), true.biom)
                                  vipHCadj.roc<-ROC( (1/(p.adjust(studt.HC$ vip [[1]]$pvals, method="fdr"))) ,true.biom)
```

```
                                studtHC.roc <- ROC((1/(studt.HC$ studentt [[1]]$pvals)) ,true.biom)
                                studtHCadj.roc <- ROC((1/(p.adjust(studt.HC$ studentt [[1]]$pvals, method="fdr")))
,true.biom)

                              # Placing the pAROC(FDR 0.2) values into the vectors
                                studtvec<-append(studtvec,AUC(stab.roc, max.mspec = .2))
                                vipvec<-append(vipvec,AUC(vip.roc, max.mspec = .2))

                                   studtHCvec<-append(studtHCvec, AUC(studtHC.roc, max.mspec = .2))
                                   vipHCvec<-append(vipHCvec, AUC(vipHC.roc, max.mspec = .2))
                                   studtHCadjvec<-append(studtHCadjvec,AUC(studtHCadj.roc, max.mspec = .2))
                                   vipHCadjvec<-append(vipHCadjvec,AUC(vipHCadj.roc, max.mspec = .2))
                                 }    #end m loop
                            # Creating a vector with the parameters and the pAROC values
                            va4<-c('Student T',l*2,i,j,k,"Low correlation",studtvec)
                            va5<-c('vip',l*2,i,j,k,"Low correlation",vipvec)
                            vb5<-c('Student T HC',l*2,i,j,k,"Low correlation",studtHCvec)
                            vb6<-c('Student T HC adj',l*2,i,j,k,"Low correlation",studtHCadjvec)
                            vb7<-c('vip HC',l*2,i,j,k,"Low correlation",vipHCvec)
                            vb8<-c('vip HC adj',l*2,i,j,k,"Low correlation",vipHCadjvec)
                            # Adding the vector to the matrix
                            mm<-rbind(mm,va4,va5,vb5,vb6,vb7,vb8)
                        } # end k loop
                    } #end j loop
                } #end i loop
            } #end l loop
# ------------  AR(1) structure  ----------------------------------------------------------------------------------------------------------------#
for(l in Ncontrol){# Loops through different sample sizes
            for(i in Numbermets){# Loops through different sizes for p
                NumberSignmets<-as.integer(c(i*0.1,i*0.2,i*0.3))# Percentage of significant mets, 10%,20%,and 30% of the
total number of mets
                    for( j in NumberSignmets){# Loops through different sizes for q
                        for(k in delta){# Loops through different values for delta
                            # Creating empty vectors for pAROC values
                              studtvec<-c()
                              vipvec<-c()
                         studtHCvec<-c()
                         vipHCvec<-c()
                         studtHCadjvec<-c()
                         vipHCadjvec<-c()
                                # Creating AR(1) correlation matrix
                                  mycovAR <- rho^(as.matrix(dist(1:i)))
                                  diag(mycovAR) <- 1
                                       for(m in 1:Numbersims){# Loops through all simulated datasets
set.seed(94783+m)
                                            # Simulating metabolomics data with different parameters in each loop
                                                simdata <- gen.data(ncontrol = l, nvar = i, nbiom = j,cormat = mycovAR, nsimul
= Numbersims,means=rep(0,i),group.diff = k)
                                            #Using Student T and VIP methods on the simulated data
                                                studt.coef <- get.biom(X = simdata$X[,,m], Y = simdata$Y, fmethod
=c("studentt","vip") ,ncomp=2, scale.p="none", type = "stab"  )
                                            # Creating a vector of true/false, where true represents the true biomarkers
                                                true.biom <- (1:ncol(simdata$X[,,m]) %in% c(1:j))
                                            # Creating ROC cuves from the two methods
                                                studt.roc <- ROC(studt.coef$studentt[[1]]$fraction.selected, true.biom)
                                                vip.roc<-ROC(studt.coef$vip$`2`$fraction.selected,true.biom)
```

```
# now get pvalues and adjusted pvalues for student t and vip using HC
                                        studt.HC <- get.biom(X = simdata$X[,,m], Y = simdata$Y, fmethod
=c("studentt","vip") ,ncomp=2, scale.p="none", type = "HC")
                                    # Creating ROC cuves from the two methods
                                      vipHC.roc<-ROC( (1/(studt.HC$ vip [[1]]$pvals)), true.biom)
                                      vipHCadj.roc<-ROC( (1/(p.adjust(studt.HC$ vip [[1]]$pvals, method="fdr")))
,true.biom)
                                      studtHC.roc <- ROC((1/(studt.HC$ studentt [[1]]$pvals)) ,true.biom)
                                      studtHCadj.roc <- ROC((1/(p.adjust(studt.HC$ studentt [[1]]$pvals,
method="fdr"))) ,true.biom)
                                    # Placing the pAROC(FDR 0.2) values into the vectors
                                       studtvec<-append(studtvec,AUC(stab.roc, max.mspec = .2))
                                       vipvec<-append(vipvec,AUC(vip.roc, max.mspec = .2))
                            studtHCvec<-append(studtHCvec, AUC(studtHC.roc, max.mspec = .2))
                                          vipHCvec<-append(vipHCvec, AUC(vipHC.roc, max.mspec = .2))
                                          studtHCadjvec<-append(studtHCadjvec,AUC(studtHCadj.roc, max.mspec = .2))
                                          vipHCadjvec<-append(vipHCadjvec,AUC(vipHCadj.roc, max.mspec = .2))
                                    }    #end m loop
                            # Creating a vector with the parameters and the pAROC values
                            va6<-c('Student T',l*2,i,j,k,"AR(1)",studtvec)
                            va7<-c('vip',l*2,i,j,k,"AR(1)",vipvec)
                            vb9<-c('Student T HC',l*2,i,j,k,"AR(1)",studtHCvec)
                            vb10<-c('Student T HC adj',l*2,i,j,k,"AR(1)",studtHCadjvec)
                            vb11<-c('vip HC',l*2,i,j,k,"AR(1)",vipHCvec)
                            vb12<-c('vip HC adj',l*2,i,j,k,"AR(1)",vipHCadjvec)
                            # Adding the vector to the matrix
                            mm<-rbind(mm,va6,va7,vb9,vb10,vb11,vb12)
                        } # end k loop
                } #end j loop
            } #end i loop
    } #end l loop

#----------------------    create output .csv file   ----------------------------------------------------------------------#
#Changing the matrix into a dataframe
ff<-data.frame(mm[-1,])
#Naming the columns. After sigma, each column represents the pAROC values estimated  in each simulation
names<-c('method','n','p','q','delta','sigma',rep(1:Numbersims))
colnames(ff)<-names
#Writing output to csv file
write.csv(ff,"studentT_VIP.csv")
Sys.time() - startT

#======================= END PROGRAM ========================================#
```