

```

# Function:    sortSKit
# Version:    1.5
# Date:       2010-01-17
# Author:     Gabor Meszaros < gabor.meszaros@boku.ac.at >
# Maintainer: Gabor Meszaros < gabor.meszaros@boku.ac.at >
# Depends:    R (>= 2.8.0)
# Description: Function sortSKit was primarily intended to create
#              an alternative option to sort data files for the Survival Kit
#              (Ducrocq and Soelkner, 1998), when using the COX program.
#              Later it was extended to sort text files with sorting criteria
#              up to three columns (both ascending and descending order).
# License:    GPL (version 3 or later)

```

```

sortSKit=function(inputFile,
                  outputFile="sorted.txt",
                  mode="default",
                  strata=0,
                  order1=0,
                  order2=0,
                  order3=0,
                  wdNew=" ",
                  wdSetBack=TRUE
                  )
{
# initial values
  wdCurr=getwd()
# changes working directory, if specified
  if (nchar(wdNew)>1){
    setwd(wdNew)
  }
#####
# does the sorting #
#####
  toSort = read.table(inputFile,header=FALSE)
  attach(toSort)
#
# the DEFAULT case
#

if (mode=="default") {

# converts negative values into positive values
  orderx=c(0,0,0)
  if (order1 < 0) {orderx[1]=sqrt(order1**2)}
  if (order2 < 0) {orderx[2]=sqrt(order2**2)}
  if (order3 < 0) {orderx[3]=sqrt(order3**2)}

# creates column names for ascending and descending sort
  if (order1>0) {stColO1=toSort[paste("V",order1,sep="")]
  } else if (order1<0) {stColO1=-toSort[paste("V",orderx[1],sep="")]
  }
  if (order2>0) {stColO2=toSort[paste("V",order2,sep="")]
  } else if (order2<0) {stColO2=-toSort[paste("V",orderx[2],sep="")]
  }
  if (order3>0) {stColO3=toSort[paste("V",order3,sep="")]
  } else if (order3<0) {stColO3=-toSort[paste("V",orderx[3],sep="")]
  }

# sorts the file
  if (order1!=0 & order2==0 & order3==0) {
    toSort.sorted=toSort[order(stColO1),]
  } else if (order1!=0 & order2!=0 & order3==0) {
    toSort.sorted=toSort[order(stColO1,stColO2),]
  } else if (order1!=0 & order2!=0 & order3!=0) {
    toSort.sorted=toSort[order(stColO1,stColO2,stColO3),]
  } else { stop("Problem with column numbers in order1, order2 or order3.")
  }
}

```

```

    }
# error message if strata is used in default setting
  if (strata!=0) {
    stop("The 'strata' option is not permitted in the 'default' mode!
    Use 'order1' instead to sort by a single column.")
  }
}
#
# in case of COX
#

if (mode=="cox"){
  if (strata == 0) {
    toSort.sorted=toSort[order(-V1, V2),]
  } else if (strata > 0) {
    stColStr=paste("V",strata,sep="")
    toSort.sorted=toSort[order(toSort[stColStr], -V1, V2),]
  } else { stop(" Only positive values allowed for 'strata' definition! ") }
# error message if strata is used in default setting
  if (order1!=0 | order2!=0 | order3!=0) {
    stop("The 'order1', 'order2', 'order3' options are not permitted in the 'cox' mode!")
  }
}
#
# any other case
#

if (mode!="default" & mode!="cox") {
  stop(
    "Improper keyword in the 'mode' statement.
    It should be: 'default' or 'cox' (lower case!)"
  )
# writes out the results to file
write.table(toSort.sorted,file=outputFile,row.names=FALSE,col.names=FALSE)
detach(toSort)
# back to initial values, if not specified otherwise
if (wdSetBack==TRUE){
  setwd(wdCurr)
}
# final information
paste("Sorting done. The output file is:",outputFile,"in directory",getwd())
}

```