

```

# file name Cordef&reg.
# one can insert their own data after one run of the program
# data for example (N(0,1), and correlation coefficient rho) to set up the routines
# please change rho below to observe other conditions
rho <- -0.6
n <- 51
x <- rnorm(n)
y <- x *rho + sqrt(1-rho^2) *rnorm(n)
# run following line for Cauchy data with same rho
#conv <- abs(rnorm(n));y <-y/conv ; x <- x/conv
# the GDCC location function is Cg
Cg <- function(x) { n <- length(x); xs <- sort(x)
k1 <- (n+1)%/% 3 ; k2 <- (n+3)%/% 3
k3 <- (2*n+2) %/% 3 ; k4 <- (2*n+4)%/%3
xs <- sort(x)
Cg <- (xs[k1] + xs[k2] + xs[k3] + xs[k4])/4
Cg }
Cg(x)
# end location fcn
# (0) fxyrk is a function run on x,y to break ties so that unique ranks are
# chosen to maximize (rky1) and minimize (rky2) correlation
# w will contain these two sets of ranks
fxyrk <- function(x,y) {
n <- length(x)
xt <- x[order(y,x)] # x ordered by y with y ties ordered by x
rky <- 1:n
rky1 <- rky[order(xt,rky)] # ranks of y ordered by x, most positive or max
# now most negative or min
xrr <- n + 1 -rank(x) #reverse ranks on the x
xt <- x[order(y,xrr)] # x ordered by y with y ties ordered by rev(x)
rky2 <- order(xt,n:1) # ranks of y ordered by x with y ties ordered by rev(y)
w <- matrix(c(rky1,rky2),n,2,byrow=FALSE )
}
# (1) Greatest Deviation Correlation Coefficient (GDCC)
GDave <- function(x,y) { w <- fxyrk(x,y)
n <- length(w[,1]) ; n1 <- n-1 ; k <- 1; cc <- NULL
if(sum(abs(w[,1] - w[,2])) == 0 ) nave <- 1 else nave <- 2 #no ties if 1
while(k <= nave) {rky <- w[,k]
ryr <- n +1 -rky
dy <- NULL; dyn <- NULL
for (i in 1:n1) {
dy <- c(dy,sum(rky[1:i]-i >0))
dyn <- c(dyn,sum(ryr[1:i]-i>0)) }
mdyr <- max(dyn)
mdy <- max(dy)
cc[k] <- (mdyr - mdy)/(n%/%2)
cc[2] <- cc[k]
k <- k + 1 }
GDcor <- (cc[1]+cc[2])/2
GDcor }
# (2) Kendall's Correlation Coefficient (Tau)
KENtau <- function(x,y) {w <- fxyrk(x,y)
n <- length(x) ; n1 <- n-1
rky <- w[,1] ; rky2 <- w[,2]
dy <- 0 ;dy2 <- 0
for(i in 1:n1) {i1 <- i+1
dy <- dy + sum(rky[i] < rky[i1:n])
dy2 <- dy2 + sum(rky2[i] < rky2[i1:n]) }
KT <- ((dy+dy2)/choose(n,2))-1
KT }
# (3) Gini or Modified Footrule CC
Gini <- function(x,y) {w <- fxyrk(x,y)
n <- length(x) ; ident <- 1:n
rky <- w[,1] ; rky2 <- w[,2]
dpnc <- sum(abs(n+1-rky -ident))
dppc <- sum(abs(rky - ident))
dpnc2 <- sum(abs(n+1-rky2 -ident))
dppc2 <- sum(abs(rky2 - ident))
den <- n^2%/%2
Gcor <- ((dpnc+dpnc2) - (dppc+dppc2))/(den*2)

```

```

Gcor }
#
# (4) absolute value CC, the continuous version of Gini
abscor <- function(x,y){
ym <- mean(y); xm <- mean(x)
SAx <- sum(abs(x-xm)); SAy <- sum(abs(y-ym))
dpnc <- sum(abs((x-xm)/SAx + (y-ym)/SAy))
dppc <- sum(abs((x-xm)/SAx - (y-ym)/SAy))
rav <- (dpnc-dppc)/2
rav
}
#
# (5)MAD CCs, and the covariance form , 3 definitions
# median CCs can be, for very high correlation, occasionally greater than 1
MADcor <- function(x,y){xm <- median(x)
ym <- median(y)
madx <- median(abs(x-xm))
mady <- median(abs(y-ym))
dpnc <- median(abs((x-xm)/madx + (y-ym)/mady))
dppc <- median(abs((x-xm)/madx - (y-ym)/mady))
rmad <- (dpnc-dppc)/2
rmad
}
# (6) now a MAD covarince definition
CMAD <- function(x,y){ xm <- median(x)
ym <- median(y)
xmc <- x-xm ; ymc <- y - ym
tplus <- median(abs(xmc+ymc))
tminus <- median(abs(xmc-ymc))
covmad <- (tplus^2 - tminus^2)/4
covmad }
# As is done for the scale estimate MAD it is also necessary for CMAD
# i.e., must divide by .6745 squared for normal distribution data
# but by 1 if Cauchy distribution data
# (7) below is a another definition of a median CC, CORMAD
CORMAD <- function(x,y) { mcor <- CMAD(x,y)/(sqrt(CMAD(x,x)*CMAD(y,y)))
mcor } # new MAD corr and note the form is like classicl definition
#
# now come the population inverses for elliptical contoured distributions,
# except for CORMAD
GinI <- function(y) {tan(pi*y/4)*sqrt(1+2*cos(pi*y/2)) } # Gini or MF inv
GDI <- function(y) sin(pi*y/2) #GD or Kendall inverse
MADI <- function(y) y*sqrt(2-y^2) # MAD or abs CC inverses
c1 <- GDave(x,y); c2 <- KENTau(x,y); c3 <- Gini(x,y)
c4 <- abscor(x,y); c5 <- MADcor(x,y); c6 <- CORMAD(x,y)

cor(x,y,use="all.obs",method=c("kendall")) # check Kendall value
c11 <- GDI(c1); c22 <- GDI(c2) ; c33 <- GinI(c3)
c44 <- MADI(c4); c55 <- MADI(c5)

# below are the regressions
#
GDfcn <- function(b,x,y) { GDave(x,y-b*x) }
GDslp <- uniroot(GDfcn,c(-10,10),x=x,y=y)$root
Kenfcn <- function(b,x,y) { KENTau(x,y-b*x) }
Kenslp <- uniroot(Kenfcn,c(-2,+2),x=x,y=y)$root
Ginfcn <- function(b,x,y) { Gini(x,y-b*x) }
Ginslp <- uniroot(Ginfcn,c(-2,+2),x=x,y=y)$root
absfcn <- function(b,x,y) {abscor(x,y-b*x) }
absslp <- uniroot(absfcn,c(-2,+2),x=x,y=y)$root
madfcn <- function(b,x,y) {MADcor(x,y-b*x) }
madslp <- uniroot(madfcn,c(-5,7.0),x=x,y=y)$root
mad2fcn <- function(b,x,y) {CMAD(x,y-b*x) }
mad2slp <- uniroot(mad2fcn,c(-5,7),x=x,y=y)$root
Pfcn <- function(b,x,y) {cor(x,y-b*x) }
Pslp <- uniroot(Pfcn,c(-5,7.0),x=x,y=y)$root
# output below
c7 <- cor(x,y)
cat(" " "GDCC"," Kendall"," Gini"," Absolute"," MADcor"," CORMAD"," Pearson","
" corr",c1,c2,c3,c4,c5,c6,c7,"\n",

```

```

"trans",c11,c22,c33,c44,c55,"unknown",c7,"\n",
"slope",GDslp,Kenslp,Ginslp,absslp,madslp,mad2slp,Pslp,"\n")

# on graph put in regression line for GDCC
plot(x,y)
title("CES SLR zero correlation and minimization methods")
res <- y-GDslp*x
int <- median(res)
abline(int,GDslp)

# minimization method below, all with GDCC scale and location estimates
# for b slope between -5 and 5
ysr <- sort(y)
rtest <- function(b,x,y) {y1 <- sort(y - b*x)
s <- uniroot(GDfcn,c(-5,5),x= ysr, y = y1)$root
return(s) }
out <- optimize(rtest,c(-5,5), x=x, y=y)
out$min; out$obj
res <- y - out$min *x
int <- Cg(res) # GDCC location statistic
abline(int,out$min)
c(int,out$min) # intercept and slope
1- (out$obj)^2 # estimate of coefficient of determination
# COD with GDCC or MADCC can get < 0, don't know about Kendall CC
sqrt(1- (out$obj)^2) # estimate of absolute value of correlation
# end minimization process
r1 <- signif(c(c1,c2,c3,c4,c5,c6,c7),digits=5)
r2 <- signif(c(c11,c22,c33,c44,c55,NA,c7),digits=5)
r3 <- signif(c(GDslp,Kenslp,Ginslp,absslp,madslp,mad2slp,Pslp),digits=5)
matcc<- matrix(c(r1,r2,r3),3,7,byrow=TRUE)
rownames(matcc) <- c("corr","trans","slope")
colnames(matcc) <- c("GDCC"," Kendall"," Gini"," Absolute"," MADcor"," CORMAD"," Pearson")
matcc

```