```
#=============================================================================
#  file:           c:\Projects\CBP\Rcourse\MdCoreTrend\MdCTstepsTrends.r
#  function:       try to linke magnitude of step changes to other variables
#
#  programmer:    Elgin S. Perry, Ph. D.
#
#  date:          7/31/2014
#
#  address:       2000 Kings Landing Rd.
#                 Huntingtown,  Md.  20639
#
#  voice phone:   (410)535-2949
#  email:         EPERRY@chesapeake.net
#=============================================================================

#install.packages()
#library(lattice) #Used for contour plots [contourplot()]
#library(nlme)    #used for gam Mixed model [gamm()]
#library(MASS)    #used for glm Mixed model [glmmPQL()]
library(mgcv)    #Wood's gam package
library(chron)   #date functions
#library(doBy)    # Allows "BY processing similar to SAS
#library(FitAR)   #AR package from McLeod and Zhang
#library(Hmisc)   #stat function by Frank Harrell
#library(cluster) #cluster analysis routines
options(stringsAsFactors = FALSE)
get.ind <- function(x,y)
  { # get index of match for x in y
  ind <- 1:length(y)
  get.ind <- ind[x==y]
  }
vec.strg <- function(x,sep=' ')
  # converts a vector to single string character
  {
  if(length(x) > 1)
    {s <- ""
     for (y in x)
       {s <- paste(s,y,sep=sep)
       }
    }else
    {print('argument not a vector in vec.strg')
     s <- paste(x)
    }
  vec.strg <- s
  } # end of vec.strg

source("C:/Projects/Rtp/dfsum.r")
source("C:/Projects/Rtp/RTF.r")
source("C:/Projects/Rtp/DistFunct.r")
doy <- function(date)
  { # compute day of calendar year for a date
  # date must be of class dates, use dates()
  yr <- years(date)
  fdc <- paste('01/01/',yr)
  fd <- chron(dates = fdc)
  doy <- date-fd+1
  }

#  be sure to change \ to /
ProjRoot <- 'c:/Projects/CBP/Rcourse/MdCoreTrend/'
setwd(ProjRoot);
# file for writing *.rtf results
RTFout <- paste(ProjRoot,"MdCtStep.rtf",sep='')
# file to temporarily store plots from RTFput.plt()
```

```
temp.plot <- paste(ProjRoot,"TempPng.png",sep='')

# check number of fields in data
datafile <- paste(ProjRoot,"CORETrend97_13.csv",sep='');
a <- count.fields(datafile, sep = ",", quote = "\"", skip = 1,
            blank.lines.skip = TRUE, comment.char = "#")
range(a)
#rbind(1:length(a),a)

# read data into dataframe
ct <- read.table(datafile, header=TRUE, sep=",", na.strings="NA", dec=".",
strip.white=TRUE,stringsAsFactors = FALSE)
dfsum(ct)

# [1] "STATION"   "REP_NUM"   "SDEPTH"    "YEAR"      "MONTH"     "DAY"       "TOC_G"
"TSS_G"     "TKNW_G"
#[10] "TP_G"      "TOC"       "TSS"       "TKNW"      "TP"        "NH4"       "NH4_G"
"NO23"      "NO23_G"
#[19] "NO2"       "NO2_G"     "PO4"       "PO4_G"     "DATE"      "LayerCode" "DOC_A"
"DOC_G"     "DOC"
#[28] "NH4_A"     "NO2_A"     "NO23_A"    "PC_A"      "PC_G"      "PC"        "PN_A"
"PN_G"      "PN"
#[37] "PO4_A"     "PP_A"      "PP_G"      "PP"        "TDN_A"     "TDN_G"     "TDN"
"TDP_A"     "TDP_G"
#[46] "TDP"       "TKNW_A"    "TKNType"   "TOC_A"     "TP_A"      "TSS_A"     "TN"
"TN_G"

# transform character date into r-date
ct$date <- as.POSIXct(strptime(ct$DATE, "%d%b%Y"))

# establish the date of lab change
lab.change <- as.POSIXct(strptime("2005-07-01","%Y-%m-%d"))
# create a binary variable to model step change due to lab
ct$step <- as.numeric(ct$date > lab.change)
# create a day of year variable for modeling seasonal effects
ct$doy <- doy(dates(paste(ct$date),format="Y-m-d"))
#  create a year variable for trend
ct$year <- as.numeric(ct$YEAR)

# make a vector of dependent variables for this analysis
deps <- c("TSS","TN","NH4","NO23","NO2","TP","PO4")
# originally had PN and PP in this list, but not PN data before lab change
# log transform the dependent variables
lndeps <- paste('ln',deps,sep='')
ct$lnTSS  <- log(ct$TSS+0.1)
ct$lnTN   <- log(ct$TN  )
ct$lnNH4  <- log(ct$NH4 )
ct$lnNO23 <- log(ct$NO23)
ct$lnNO2  <- log(ct$NO2 )
ct$lnTP   <- log(ct$TP  )
ct$lnPO4  <- log(ct$PO4 )


# make a vector of stations for this analysis
stats <- unique(ct$STATION)

mnct <- aggregate(ct[,lndeps],list(station=ct$STATION),mean,na.rm=TRUE)

mnctl <-
reshape(mnct,idvar='station',varying=lndeps,v.names="lndeps",times=lndeps,direction="long")
dfsum(mnctl)
names(mnctl) <- c('station','lndep','statmn')

load(file=paste(ProjRoot,"AllSteps.rdata",sep=''))
names(all.step)[2] <- 'lndep'
```

```
aswide <-
reshape(all.step,idvar='station',varying=lndeps,v.names="lndeps",times=lndeps,direction="wide
")
steps <- merge(mnctl,all.step,by=c('station','lndep'))

all.step$dir <- ifelse(all.step$step.est > 0,'Positive','Negative')

table(all.step$lndep,all.step$step.sig)
addmargins(table(all.step$lndep,all.step$step.sig,all.step$dir))

step.plot <- function(x,y)
  {
#  x<- 'lnTSS'; y <- "lnPO4"
  mnx <- mnct[,c('station',x)]
  stepy <- all.step[all.step$lndep==y,]
  step <- merge(stepy,mnx,by='station')
  step$step.sig.col <- ifelse(step$step.sig,'red','blue')

plot(step[,x],step$step.est,xlab=paste('mean',x),ylab=paste('step',y),col=step$step.sig.col,p
ch=19)
  }


step.plot('lnTSS','lnPO4')
step.plot('lnPO4','lnPO4')

for (x in lndeps)
  {
  for (y in lndeps)
    {
    step.plot(x,y)
    readline('hit enter to continue ')
    }
  }


y <- 'lnPO4'
for (x in lndeps)
  {
  step.plot(x,y)
  readline('hit enter to continue ')
  }

y <- 'lnNH4'
for (x in lndeps)
  {
  step.plot(x,y)
  readline('hit enter to continue ')
  }
```