```
# functions for distribution description
lnpp <- function(y,xlab='expected',const=0,ylab='observed',main='lognormal probability plot')
{
  #y <- exp(rnorm(100));xlab<-'expected';ylab<-'observed';main<-'lognormal probability
plot';const=0
  ys <- sort(y)
  lnys <- log(y+const)
  n <- length(y)
  yr <- 1:n
  er <- (yr-0.5)/n
  elny <- qnorm(er)
  ey <- exp(elny)
  mnlny = mean(lnys) - const
  sdlny = sd(lnys)
  xlim <- c(ey[1],ey[n])
  eylim <- c(exp(mnlny + elny[1]*sdlny),exp(mnlny + elny[n]*sdlny))
  ylim <- c(min(ys[1],eylim[1]),max(ys[n],eylim[2]))
  plot(ey,ys,xlab=xlab,ylab=ylab,main=main,xlim=xlim,ylim=ylim)
  lines(xlim,eylim)
  } #end of lnpp function

skew <- function(y) {
  #y <- 1:10
  cy <- y - mean(y)
  m3 <- sum(cy*cy*cy)/length(y)
  sdy <- sd(y)
  skew <- m3/(sdy^3)
  } #end skew
kurt <- function(y) {
  y <- 1:10
  cy <- y - mean(y)
  m4 <- sum(cy*cy*cy*cy)/length(y)
  m2 <- var(y)
  kurt <- m4/(m2^2) - 3
  } #end skew

npp <- function(y,xlab='expected',ylab='observed',main='normal probability plot') {
  #y <- rnorm(100);xlab<-'expected';ylab<-'observed';main<-'normal probability plot'
  ys <- sort(y)
  n <- length(y)
  yr <- 1:n
  er <- (yr-0.5)/n
  ey <- qnorm(er)
  mny = mean(y)
  sdy = sd(y)
  xlim <- c(ey[1],ey[n])
  eylim <- c(mny + ey[1]*sdy,mny + ey[n]*sdy)
  ylim <- c(min(ys[1],mny + ey[1]*sdy),max(ys[n],mny + ey[n]*sdy))
  plot(ey,ys,xlab=xlab,ylab=ylab,main=main,xlim=xlim,ylim=ylim)
  lines(xlim,eylim)
  # boxplot(y,add=TRUE,at=min(ey),pch=8)

  } #end of npp function

DistPlot <- function(y,vlab='observed',main='distribution plots') {
  # does normal probability plot, boxplot, histogram, and density plot for y
  # needs npp() above
  #y <- swmR38$TSS; #debug
  #vlab <- 'TSS'
  par.def <- par(no.readonly = TRUE) # save default settings
  par(mfrow=c(2,2)) # sets up for multiple plots per page
  npp(y,ylab=vlab)
  mtext(text=main,side=3,line=-1,outer=TRUE,font=2)
  boxplot(y,pch=8,main = 'boxplot',ylab=vlab)
```

```
  points(1,mean(y),pch=15)
  hist(y,xlab=vlab,main='histogram')
  a <- density(y)
  plot(a$x,a$y,xlab=vlab,ylab='density',type='l',main='density plot')
  ytic <- runif(length(y),min=0,max=0.25*max(a$y))
  points(y,ytic)
  par(par.def)
  par(mfrow=c(1,1)) # sets back to 1 plot per page
  }  #end DistPlot

DistPlot2v <- function(y1,y2,vlab1='observed',vlab2='observed',main='distribution plots') {
  # compare distribution plots for two variables
  #y1<-lh$fec; y2<-lh$lnfec; vlab1 <- 'test v1'; vlab2 <- 'test v2'; main <- 'test' #debug
line
  par(mfrow=c(2,2)) # sets up for multiple plots per page
  npp(y1,ylab=vlab1)
  mtext(text=main,side=3,line=-1,outer=TRUE,font=2)
  a <- density(y1)
  plot(a$x,a$y,xlab=vlab,ylab='density',type='l',main='density plot')
  ytic <- runif(length(y1),min=0,max=0.25*max(a$y))
  points(y1,ytic)

  npp(y2,ylab=vlab2)
  mtext(text=main,side=3,line=-1,outer=TRUE,font=2)
  a <- density(y2)
  plot(a$x,a$y,xlab=vlab,ylab='density',type='l',main='density plot')
  ytic <- runif(length(y2),min=0,max=0.25*max(a$y))
  points(y2,ytic)
  }  #end DistPlot2v

skew <- function(y) {
  #y <- 1:10
  cy <- y - mean(y)
  m3 <- sum(cy*cy*cy)/length(y)
  sdy <- sd(y)
  skew <- m3/(sdy^3)
  } #end skew
kurt <- function(y) {
  #y <- 1:10
  cy <- y - mean(y)
  m4 <- sum(cy*cy*cy*cy)/length(y)
  m2 <- var(y)
  kurt <- m4/(m2^2) - 3
  } #end skew
DistSum <- function(y,vlab='observed'){
  #y <- 1:10
  DistTmp <- data.frame(c('sample size','mean','standard dev','variance', 'skewness', 'excess
kurtosis','minimum','q25','median','q75','maximum'),

c(length(y),mean(y),sd(y),var(y),skew(y),kurt(y),min(y),quantile(y,0.25),median(y),quantile(y
,0.75),max(y)))
  names(DistTmp) <- c('statistic','estimate')
  DistSum <- DistTmp
  } #end of DistSum
pltedf <- function(x,main,xlab,xmn,xmx,clr)
  { # plot edf with this, overlay edf with addedf()
# x <- rnorm(50);  main <- 'test'; xlab<-'x'; xmn <- -4;xmx<-4; clr='red'
#   x<- Medway[(Medway$month==5),'meanflow']; xmn<-2500; xmx<-14000
  lx <- length(x)
  x <- sort(x)
  edfx <- ((1:lx)-0.5)/(lx+1)
  plot(x,edfx,xlim=c(xmn,xmx),main=main,xlab=xlab,type='s',col=clr)
  }
addedf <- function(x,clr)
  {
```

```
    lx <- length(x)
    x <- sort(x)
    edfx <- ((1:lx)-0.5)/(lx+1)
    lines(x,edfx,type='s',col=clr)
    }
```