```
#=====================================================================
#  file:          c:\Projects\CBP\Rcourse\FunctionRbit006.r
#  function:      show how to write user defined function
#
#  programmer:    Elgin S. Perry, Ph. D.
#
#  date:           6/22/2014
#
#  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)

add <- function(x1,x2) { add <- x1+x2}

# example of function for simple calculation
Euc.dist <- function(x1,y1,x2,y2)
  {
#  x1<-1; y1<-1; x2<-5; y2<-4
  xdist <- x2-x1
  ydist <- y2-y1
  Euc.dist <- sqrt(xdist^2+ydist^2)
  }  # end of Euc.dist

print(Euc.dist(1,1,7,9))

# passing back multiple results
Euc.dist <- function(x1,y1,x2,y2)
  {
#  x1<-1; y1<-1; x2<-5; y2<-4
  xdist <- x2-x1
  ydist <- y2-y1
  Euc.dist <- c(sqrt(xdist^2+ydist^2), xdist, ydist)
  }  # end of Euc.dist

print(Euc.dist(1,1,7,9))


# a more R-like version

Euc.dist <- function(x1,x2)
  {
#  x1<-c(1,1); x2<-c(5,4)
  xdist <- x2-x1
  Euc.dist <- sqrt(t(xdist)%*%xdist)
  }  # end of Euc.dist

print(Euc.dist(c(1,1),c(7,9)))
print(Euc.dist(c(1,1,1),c(7,9,10)))
print(Euc.dist(c(1,1,1,1),c(7,9,10,11)))
```

```
# load libraries
library(chron)
library(mgcv)
# set working directory
ProjRoot <- 'c:/Projects/CBP/Rcourse/'
setwd(ProjRoot);

# load some user defined functions
source("C:/Projects/Rtp/dfsum.r")
source("C:/Projects/Rtp/RTF.r")

# read data as usual
datafile <- paste(ProjRoot,"MAT_5day.csv",sep='');
mat <- read.table(datafile, header=TRUE, sep=",", na.strings="NA", dec=".",
strip.white=TRUE,stringsAsFactors = FALSE)
dfsum(mat)

# define a time variable (this uses chron library)
mat$time <- times(paste(mat$Time,'00',sep=':'))

# define an analysis function
diel.gam <- function(day)
  {
# day <- '3/21/2006'
  # select data for specified date
  tdta <- mat[day==mat$Date,]
  # fit gam model to selected data
  dogam <- gam(DO ~ s(time,bs='cc'),data=tdta)
  # get predicted values from gam
  tdta$pred <- predict(dogam)
  # plot data, label with day
  plot(DO~time,data=tdta,main=day)
  # overlay predicted line
  lines(pred~time,data=tdta,col='red',lwd=2)
  # get max and min predictions
  range.do <- range(tdta$pred)
  # locate times associated with max and min
  min.pt <- tdta[range.do[1]==tdta$pred,c('time','pred')]
  max.pt <- tdta[range.do[2]==tdta$pred,c('time','pred')]
  # label max and min on plot
  text(min.pt[1],min.pt[2],'min',cex=1.5,col='red',pos=1)
  text(max.pt[1],max.pt[2],'max',cex=1.5,col='red',pos=3)

  } #end diel.gam
# test for one date
diel.gam('3/22/2006')

# call function for each date in mat
daylist <- unique(mat$Date)
diel.gam(daylist[1])
diel.gam(daylist[2])
diel.gam(daylist[3])
diel.gam(daylist[4])
diel.gam(daylist[5])


# call function for each date in mat using loop and save graphics output to pdf
pdf(paste(ProjRoot,"AllDielplots",".pdf",sep=''),width=9,height=6.5)
for (day in daylist) { diel.gam(day) }
dev.off()
```