```
#-------------------------------------------------------------------------------
#  File Name:
#     quaternary_tree_statistics.R
#
#  Purpose:
#
#     R script file for quaternary search tree statistics
#
#  Parameters:
#
#
#  Discussion:
#
#
#  Licensing:
#
#    This code is distributed under the GNU LGPL license.
#
#  Modified:
#
#    2018.11.12 Mon
#
#  Author:
#
#    Young Won Lim
#
#-------------------------------------------------------------------------------
R <<- 4
level_1 <- 2
level_2 <- 9
ngroup <- R^level_1
nleaf <- R^level_2
nnode <- nleaf/ngroup


#-------------------------------------------------------------------------
read_th_vec <- function( fname, nelem ) {
  to.read = file(fname, "rb")

  th_vec = c()
  ch = c()
  for (i in 1:nelem) {
    # double theta   -- Be careful for data alignment !!!
    # int    branch  -- (position, order ...)
    # int    depth
    # int    id
    # int    child[4]
    # int    parent

    th <- readBin(to.read, "double", n=1, size=8)
    br <- readBin(to.read, "int",    n=1, size=4)
    dp <- readBin(to.read, "int",    n=1, size=4)

    id <- readBin(to.read, "int",    n=1, size=4)
    ch[1:R] <- readBin(to.read, "int",    n=R, size=4)
    pa <- readBin(to.read, "int",    n=1, size=4)

    if ((2+3+R+1) %% 2) readBin(to.read, "int",    n=1, size=4)

    s1<- sprintf("th=%+10.6f, br=%2d, dp=%2d, id=%2d, ", th, br, dp, id)
    s2<- sprintf("ch[1:%d]= %s, ", R, paste(ch, collapse=", "))
    s3<- sprintf("pa=%2d", pa)
    print(paste(s1, s2, s3))

    th_vec = append(th_vec, th)

    # print(th_vec)
  }

  close(to.read)
```

```
  #t <- order(th_vec, decreasing=TRUE)

  return(th_vec)

}

#-----------------------------------------------------------------------
read_data <- function( level, group, ngroup ) {

  nogroup <- missing(group) & missing(ngroup)

  if (nogroup) {
    fname <- sprintf("quaternary_tree_L%02d.dat", level)
    nelem <- R^level
  } else {
    fname <- sprintf("quaternary_tree_L%02d.G%02d.dat", level, group)
    nelem <- R^level / ngroup
  }

  th_vec <- read_th_vec(fname,nelem)

  return(th_vec)
}


#-----------------------------------------------------------------------
plot_ordering <- function(level, x, group, ngroup) {

  nogroup <- missing(group) & missing(ngroup)

  if (nogroup) {
    fname <- sprintf("quaternary_tree_L%02d.dat", level)
    plist <- list("level"=level, "fname"=fname)
    titles <- "Theta angle ordering - group"
  } else {
    fname <- sprintf("quaternary_tree_L%02d.G%02d.dat", level, group)
    plist <- list("level"=level, "group"=group-1, "ngroup"=ngroup)
    plist <- append(plist, list("fname"=fname))
    titles <- "Theta angle ordering - leaf"
  }

  local({
  rk.header (titles, parameters=plist)

  rk.graph.on ()
  try ({
      dotchart(x)
  })
  rk.graph.off ()
  })
}


#-----------------------------------------------------------------------
print_diff_statistics <- function(level, x_diff, group, ngroup) {

  nogroup <- missing(group) & missing(ngroup)

  if (nogroup) {
    fname <- sprintf("quaternary_tree_L%02d.dat", level)
    plist <- list("level"=level, "fname"=fname)
    titles <- "Theta angle difference statistics - leaf"
  } else {
    fname <- sprintf("quaternary_tree_L%02d.G%02d.dat", level, group)
    plist <- list("level"=level, "group"=group-1, "ngroup"=ngroup)
    plist <- append(plist, list("fname"=fname))
    titles <- "Theta angle difference statistics - group"
  }
```

```
  local({
  ## Compute
  vars <- rk.list (x_diff)
  results <- data.frame ("Variable Name"=I(names (vars)), check.names=FALSE)
  for (i in 1:length (vars)) {
      var <- vars[[i]]
      results[i, "Mean"]     <- mean(var,na.rm=TRUE)
      results[i, "Variance"] <- var(var,na.rm=TRUE)
      results[i, "sd"]       <- sd(var,na.rm=TRUE)
      results[i, "Minimum"]  <- min(var,na.rm=TRUE)
      results[i, "Maximum"]  <- max(var,na.rm=TRUE)
      results[i, "Median"]   <- median(var,na.rm=TRUE)
      results[i, "Inter Quartile Range"] <- IQR(var,na.rm=TRUE)
      temp <- quantile (var,na.rm=TRUE)
      results[i, "Quartiles"] <- paste (names (temp), format (temp), sep=": ", collapse=" ")

      # robust statistics
  }

  ## Print result
  rk.header (titles, parameters=plist)

  rk.results (results)
  })

}


#------------------------------------------------------------------------
print_range_statistics <- function(level, x, group, ngroup) {

  nogroup <- missing(group) & missing(ngroup)

  if (nogroup) {
    fname <- sprintf("quaternary_tree_L%02d.dat", level)
    plist <- list("level"=level, "fname"=fname)
    titles <- "Theta angle range statistics - leaf"
  } else {
    fname <- sprintf("quaternary_tree_L%02d.G%02d.dat", level, group)
    plist <- list("level"=level, "group"=group-1, "ngroup"=ngroup)
    plist <- append(plist, list("fname"=fname))
    titles <- "Theta angle range statistics - group"
  }

  local({
  ## Compute
  vars <- rk.list (x)
  results <- data.frame ("Variable Name"=I(names (vars)), check.names=FALSE)
  for (i in 1:length (vars)) {
      var <- vars[[i]]
      results[i, "Minimum"]  <- min(var,na.rm=TRUE)
      results[i, "Maximum"]  <- max(var,na.rm=TRUE)
      results[i, "Mean"]     <- mean(var,na.rm=TRUE)
      results[i, "Variance"] <- var(var,na.rm=TRUE)
      results[i, "sd"]       <- sd(var,na.rm=TRUE)
      results[i, "Median"]   <- median(var,na.rm=TRUE)
      # robust statistics
  }

  ## Print result
  rk.header (titles, parameters=plist)

  rk.results (results)
  })

}
```

```r
#.....................................................................
th_vec <- read_data(level_2)
th_diff <- c(diff(th_vec), 0)
th_ord <- order(th_vec, decreasing=TRUE)


G <- matrix(ncol=ngroup, nrow=nnode)
G_diff <- matrix(ncol=ngroup, nrow=nnode)
G_ord <- matrix(ncol=ngroup, nrow=nnode)
for (i in 1:ngroup) {
  G[,i] <- read_data(level_2, i-1, ngroup)
  G_diff[,i] <- c(diff(G[,i]), 0)
  G_ord[,i] <- order(G[,i], decreasing=TRUE)
}


#.....................................................................
plot_ordering(level_2, th_ord)


for (i in 1:ngroup) {
  plot_ordering(level_2, G_ord[,i], i, ngroup)
}


#.....................................................................
print_diff_statistics(level_2, th_diff)


for (i in 1:ngroup) {
  print_diff_statistics(level_2, G_diff[,i], i, ngroup)
}


#.....................................................................
print_range_statistics(level_2, th_vec)


for (i in 1:ngroup) {
  print_range_statistics(level_2, G[,i], i, ngroup)
}
```