```
# source("C:/data/program/R/NMMP_TimeLine.R")
# works with C:/data/program/R/NMMP_WMA_web.R
Version <- function() {
  Copyright.statement <- "Please acknowledge source: Resource Quality Services, Department of Water Affairs"
  Author <- c("Michael Silberbauer")
  # File.description comment
  program.purpose <-
    "Plot _Escherichia coli_ and turbidity data from the national microbiological monitoring programme"
  program.inputs <- "The national water quality database, WMS"
  program.outputs <- "A time series plot for each monitoring point, showing E. coli and turbidity"
  Version <- 0.1 #  First version, including several Google R Style conventions; note that the WMS database uses
  # underscores, so these are retained in variable names, for convenience, rather than dots.
  # Also, monitoring programme names are all upper case, e.g. NMMP, for clarity (Michael Silberbauer 2012-03-08)
  Version <- 0.2 #  Loop through all sites in NMMP (Michael Silberbauer 2012-03-12)
  Version <- 0.3 #  Improve graph grid (Michael Silberbauer 2012-03-14)
  Version <- 0.4 #  Fix ToTitle reaction to spaces next to the symbols at, comma and point (@,.) (Michael Silberbauer 2
  Version <- 0.5 #  Fix risk levels (Michael Silberbauer 2012-10-05)
  Version <- 1.0 # Use faecal coliform data too (Michael SIlberbauer 2013-06-21)
  Version <- 1.1 # Italics for E. coli in legend (Michael Silberbauer 2013-06-21)
  Version <- 1.11 # require(stringr) (Michael Silberbauer 2013-11-15)
  Version <- 1.12 # change "pathogenic bacteria" to "microbial pathogens" (Michael Silberbauer 2014-12-17)
  return(Version)

  # TODO :  Use standard site names from nms.
}
# No source() statements

#---------------------------------------------------------------
DateStamp <- function() {
  DateStamp<-
    paste("Data from the NMMP programme. ",
          "Plotted at Resource Quality Services, Department of Water Affairs, on ",
          format(Sys.time(),"%Y-%m-%d %H:%M:%S")," using NMMP_TimeLine.R v", Version(),
          " under ", R.version.string, sep="")
  return(DateStamp)
}
#---------------------------------------------------------------
ClassText <- function(guide.class.name,guide.range,ecoli) {
  #  Describe the water quality class based on the counts of coli per 100mL
  #  Args:
  #   guide.class.name:  standard descriptions of user risk
  #   guide.range:         cutpoints between risk classes
  #  Returns:              appropriate text for the risk class
  ClassText <- paste("<=10",guide.class.name[1])
  for(i in 1:length(guide.range)) {
    #print(paste(i,guide.range[i],guide.class.name[i])) #  debug
    if(i%%2==0) {sepa="\n"} else {sepa="; "}
    if(ecoli>=guide.range[i]) ClassText <- paste(ClassText,sepa,">",guide.range[i]," ",guide.class.name[i+1],sep="")
  }
  return(ClassText)
}
#-----------------------------------------------------Main program -

require(RODBC)
require(stringr)

if (!exists ("toTitle", mode="function") ) source ("C:/data/program/R/toTitle.R")
```

```
database <- "wmsdb"
channel <- odbcConnect(database)
database.unlock <- "set isolation dirty read"
sqlQuery(channel, database.unlock) #  Tip from Deon to allow reading a locked table

vars <- sqlQuery(channel,"select * from monitorng_variable")
vars$mon_variable_name <- str_trim(vars$mon_variable_name) # remove trailing blanks
mv.c <- c("ESCHERICHIA COLI", "FAECAL COLIFORM COUNT")
mv.e <- c("ESCHERICHIA COLI")
mv.f <- c("FAECAL COLIFORM COUNT")
k.var.ccoli <- vars$mon_variable_id[vars$mon_variable_name %in% mv.c]
k.var.ecoli <- vars$mon_variable_id[vars$mon_variable_name %in% mv.e]
k.var.fcoli <- vars$mon_variable_id[vars$mon_variable_name %in% mv.f]
#k.var.ecoli <- 77
k.var.turbidity <- 66
k.nmmp.id <- 135
guide.col.pastel <- c("#C1CDC1", "#FFFACD", "#FFE1FF", "#FFB6C1")
guide.class.name <- c(
  "low-med risk",
  "high risk when drinking untreated water",
  "high risk from full or partial contact",
  "high risk when irrigating crops that are eaten raw",
  "high risk when drinking after only limited treatment")
guide.range <- c(10,400,4000,20000)

query.SQL<-paste("SELECT UNIQUE mon_feature_id ",
                 "FROM programme_sample ",
                 "WHERE mon_program_id =", k.nmmp.id)
sites<-sqlQuery(channel, query.SQL)

for (nSite in 1:nrow(sites)) {
  featid <- sites$mon_feature_id[nSite]
  query.SQL<-paste("SELECT UNIQUE mon_feature_name ",
                   "FROM monfeat_sample_sum  ",
                   "WHERE mon_feature_id = ", featid)
  name.site <- toTitle(sqlQuery(channel, query.SQL)$mon_feature_name)

  print(paste(nSite, featid))
  #  query.SQL <- paste("SELECT mon_feature_id, sample_begin_date, sample_begin_time,",
  #         " result_num_value FROM released_result WHERE mon_variable_id = ", k.var.ecoli,
  #         " AND sample_begin_depth = 0",
  #         " AND mon_feature_id = ", featid,
  #         sep="")
  query.SQL <- paste("SELECT mon_feature_id, sample_begin_date, sample_begin_time,",
                     " sample_begin_depth, result_num_value FROM released_result",
                     " WHERE mon_variable_id = ", k.var.ecoli,
                     " AND mon_feature_id = ", featid,
                     sep="")
  ecolis <- sqlQuery(channel, query.SQL)
  ecolis$sample_begin_date <- as.Date(ecolis$sample_begin_date,"%Y-%m-%d")
  names(ecolis)[5] <- "E.coli"

  query.SQL <- paste("SELECT mon_feature_id, sample_begin_date, sample_begin_time,",
                     " sample_begin_depth, result_num_value FROM released_result",
                     " WHERE mon_variable_id = ", k.var.fcoli,
                     " AND mon_feature_id = ", featid,
                     sep="")
  fcolis <- sqlQuery(channel, query.SQL)
```

```
fcolis$sample_begin_date <- as.Date(fcolis$sample_begin_date,"%Y-%m-%d")
names(fcolis)[5] <- "f.coli"

efcolis <- merge(ecolis, fcolis,
              by=c("mon_feature_id", "sample_begin_date",
                  "sample_begin_time", "sample_begin_depth"
              ),
              all=TRUE)

query.SQL <- paste("SELECT mon_feature_id, sample_begin_date, sample_begin_time,",
               " sample_begin_depth, result_num_value FROM released_result",
               " WHERE mon_variable_id = ", k.var.turbidity,
               " AND mon_feature_id = ", featid,
               sep="")
turbidity <- sqlQuery(channel, query.SQL)
turbidity$sample_begin_date <- as.Date(turbidity$sample_begin_date, "%Y-%m-%d")
names(turbidity)[5] <- "Turbidity"

if(nrow(efcolis) < 1 && nrow(turbidity) < 1) {
  print(paste(nSite, featid, "***No Data***", name.site))
}
else {
  print(paste(nSite, featid, "plotting...", name.site))

  # Decide which data to use and put into a new column, ccoli -
  # prefer E. coli, arbitrarily, and replace missing E. coli with faecal coliforms
  # - add a ctype column with the symbol for plotting
  if(nrow(efcolis) > 0) {
    efcolis$c.coli <- as.numeric(efcolis$E.coli)
    efcolis$c.type <- with(efcolis, 16)
    for(ci in 1:nrow(efcolis)) {
      if(is.na(efcolis$c.coli[ci])) {
        efcolis$c.coli[ci] <- as.numeric(efcolis$f.coli[ci])
        efcolis$c.type[ci] <- 1
        if(is.na(efcolis$c.coli[ci])) efcolis$c.type[ci] <- 4
      }
    }
  }

  df.nmmp <- merge(efcolis, turbidity,
               by=c("mon_feature_id","sample_begin_date",
                   "sample_begin_time", "sample_begin_depth"),
               all=TRUE)
  year.1 <- as.numeric(substr(min(df.nmmp$sample_begin_date), 1, 4))
  year.2 <- as.numeric(substr(max(df.nmmp$sample_begin_date),1 , 4))
  Date.1 <- as.Date(paste(year.1, "-01-01", sep=""))
  Date.2 <- as.Date(paste(year.2, "-12-31", sep=""))
  # e <- ggplot(df.nmmp, aes(sample_begin_date,E.coli)) #  Nice idea, but not sufficiently controllable
  # e + geom_point(pch=16) + scale_y_log10()


  # plot coli and turbidity data
  PNGfile<-paste("C:/tmp/nmmp/Chart_nmmp_",featid,".png",sep="")
  png(filename=PNGfile, width=1024, height=600)
  layout(1:2)

  # E. coli
  #if(length(df.nmmp$E.Coli) < 0) {
```

```
if(length(df.nmmp$c.coli) < 1) {
  plot (c(Date.1, Date.2), c(0, 1),
        xlim=c(as.numeric(Date.1), as.numeric(Date.2)), type="n", xlab="", ylab="",
        #               main=expression("No "*italic("Eschericia coli")*" data for NMMP site"),
        main=expression("No bacterial pathogen indicator data for monitoring site"),
        xaxt="n", yaxt="n", col="grey60"
  )
  title(main=featid,line=+0.5)
}
else {
  if(min(df.nmmp$c.coli, na.rm=TRUE) < 0.5) {
    df.nmmp$c.coli[df.nmmp$c.coli < 0.5] <- 0.5
  }
  c.coli.max <- max(df.nmmp$c.coli, na.rm=TRUE)
  plot(df.nmmp$sample_begin_date, df.nmmp$c.coli,
       xlim=c(as.numeric(Date.1), as.numeric(Date.2)),
       ylim=c(0.1, c.coli.max),
       xlab="", ylab="/100mL", xaxt="n", yaxt="n", type="n", log="y")

  title(main=paste ("Full record for NMMP site ", featid, ",
  showing high risk guidelines for indicators of pathogenic bacteria.", sep=""))
  title( sub=paste(Date.1, "to", Date.2, "\n", name.site, "\n",
                   ClassText(guide.class.name, guide.range, c.coli.max)),
         cex.sub=0.8, line=4.0)
  #axis.Date(1, at=seq(Date.1, Date.2, "days"), labels=FALSE, tcl=-0.1)
  #axis.Date(1, at=seq(Date.1, Date.2, "weeks"), tcl=-0.2, cex.axis=0.8, padj=-1.6)
  axis.Date(1, at=seq(Date.1, Date.2, "months"), labels=FALSE, tcl=-0.1)
  axis.Date(1, at=seq(Date.1, Date.2, "years"), labels=TRUE, tcl=-0.25,
            cex.axis=0.8, padj=-1.6, col="grey25")

  # draw a border, background class shading and grid
  rect(par("usr")[1], 10^par("usr")[3], par("usr")[2], 10^par("usr")[4], col="lightsteelblue3")
  rect(par("usr")[1], guide.range[1], par("usr")[2], guide.range[2],    border=NA, col=guide.col.pastel[1])
  rect(par("usr")[1], guide.range[2], par("usr")[2], guide.range[3],    border=NA, col=guide.col.pastel[2])
  rect(par("usr")[1], guide.range[3], par("usr")[2], guide.range[4],    border=NA, col=guide.col.pastel[3])
  rect(par("usr")[1], guide.range[4], par("usr")[2], 10^par("usr")[4], border=NA, col=guide.col.pastel[4])

  for (ny in c(0:10)) segments(par("usr")[1],  10^ny,  par("usr")[2],  10^ny,  col="white")
  for (nx in c((year.1-1):(year.2+1))) {
    if((year.2 - year.1) < 9) {
      for (nmn in c(2:12)) {
        dseg <- as.Date(paste(nx, "-", nmn, "-01", sep=""), "%Y-%m-%d")
        segments(dseg, 10^par("usr")[3], dseg, 10^par("usr")[4], col="white")
      }
    }
    dseg <- as.Date(paste(nx, "-07-01", sep=""), "%Y-%m-%d")
    segments(dseg, 10^par("usr")[3], dseg, 10^par("usr")[4], col="white",  lwd=2)
    dseg <- as.Date(paste(nx, "-01-01", sep=""), "%Y-%m-%d")
    segments(dseg, 10^par("usr")[3], dseg, 10^par("usr")[4], col="white",  lwd=3)
  }
  #for (nx in c(year.1-1:year.2+1)) segments(nx,10^par("usr")[3],nx,10^par("usr")[4],col="white",lwd=2)
  lines(df.nmmp$sample_begin_date, df.nmmp$c.coli, col="grey90", lwd=2)
  points(df.nmmp$sample_begin_date, df.nmmp$c.coli, pch=df.nmmp$c.type, col="#333333AA", cex=1.5)
  if(c.coli.max < 10000) {
    axis(2, c(1, 10, 100, 1000), c("1","10","100", "1000"), cex.axis=0.75, las=1)
  }   else axis(2, c(1, 10, 1000, 100000, 10000000), c("1", "10", "1000", "100000", "10000000"),
                cex.axis=0.75, las=1)
  legend("bottomright",
```

```
        legend=c(expression(italic("Eschericia coli")), "faecal coliforms"),
        bg="#ffffffaa", col="#333333AA", pch=c(16, 1),
        inset=c(0.005, 0.01), box.col="grey", horiz=TRUE, cex=0.8, pt.cex=1.5)

  rect(par("usr")[1], 10^par("usr")[3], par("usr")[2], 10^par("usr")[4], border="grey60")
 }

 # Turbidity

 #if(length(df.nmmp$Turbidity) < 0) {
 if(nrow(turbidity) < 1) {
  plot (c(Date.1,Date.2),c(0,1),
       xlim=c(as.numeric(Date.1),as.numeric(Date.2)), ylim=c(0,1), type="n", xlab="", ylab="",
       main=paste("No turbidity data for", featid),
       sub=DateStamp(), cex.sub=0.7,
       xaxt="n", yaxt="n", col="grey60"
  )
 }
 else {
  turbidity.max <- max(df.nmmp$Turbidity, na.rm=TRUE)
  plot(df.nmmp$sample_begin_date, df.nmmp$Turbidity,
       xlim=c(as.numeric(Date.1),as.numeric(Date.2)),
       main="Turbidity",
       sub=DateStamp(), cex.sub=0.7,
       xlab="", xaxt="n", ylab="NTU", type="n", pch=16, las=1, col="grey60",
       ylim=c(0, turbidity.max)
  )
  axis.Date(1, at=seq(Date.1, Date.2, "months"), labels=FALSE, tcl=-0.1)
  axis.Date(1, at=seq(Date.1, Date.2, "years"), labels=TRUE, tcl=-0.25, cex.axis=0.8, padj=-1.6, col="grey25")
  rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], col="lightsteelblue3") # grey85
  rect(par("usr")[1],10,par("usr")[2], par("usr")[4],border=NA,col=guide.col.pastel[3])
  for (nx in c((year.1-1):(year.2+1))) {
   if((year.2 - year.1) < 9) {
    for (nmn in c(2:12)) {
     dseg <- as.Date(paste(nx,"-",nmn,"-01",sep=""),"%Y-%m-%d")
     segments(dseg,0,dseg,turbidity.max,col="white")
    }
   }
   dseg <- as.Date(paste(nx,"-07-01",sep=""),"%Y-%m-%d")
   segments(dseg,0,dseg,turbidity.max,col="white", lwd=2)
   dseg <- as.Date(paste(nx,"-01-01",sep=""),"%Y-%m-%d")
   segments(dseg,0,dseg,turbidity.max,col="white", lwd=3)
  }
  grid(nx=NA, ny=NULL, col="white", lty=1)
  lines(df.nmmp$sample_begin_date, df.nmmp$Turbidity, col="grey90", lwd=2)
  points(df.nmmp$sample_begin_date, df.nmmp$Turbidity, pch=16, col="#333333AA", cex=1.5)
 }
 rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], border="grey60")
 dev.off()
 }
}
```