nutrientModFunctions.R

gcn

Sat Mar 26 16:31:32 2016

@keywords utilities, nutrient data management functions title: “Functions to facilitate management of nutrient data” @name nutrientModFunctions.R @author Gerald C. Nelson, \email\protect\T1\textbraceleftnelson.gerald.c@@gmail.com}

```
# .onLoad <- function(libname, pkgname) {
#   op <- options()
#   op.devtools <- list(
#     devtools.path = "~/R-dev",
#     devtools.install.args = "",
#     devtools.name = "Gerald C. Nelson",
#     devtools.desc.author = 'person("Gerald", "Nelson",
#     "nelson.gerald.c@gmail.com", role = c("aut", "cre"))',
#     devtools.desc.license = "GPL-3",
#     devtools.desc.suggests = NULL,
#     devtools.desc = list()
#     )
#   toset <- !(names(op.devtools) %in% names(op))
#   if (any(toset))
#     options(op.devtools[toset])
#   invisible()
# }
```

Title getNewestVersion @param fileShortName The substantive (first) part of the file name. @return The most recent .RData file. @export

```
getNewestVersion <- function(fileShortName) {
  mData <- fileloc("mData")
  # see
  # http://stackoverflow.com/questions/7381641/regex-matching-beginning-and-end-strings
  # for an explanation of this regex expression
 # regExp <- paste("(?=^", fileShortName, ")(?=.*RData$)", sep = "")
  regExp <- paste("(?=^", fileShortName, ")(?=.*rds$)", sep = "")
  filesList <-
    grep(regExp,
         list.files(mData),
         value = TRUE,
         perl = TRUE)
  newestFile <- filesList[length(filesList)]
  #print(newestFile)
#  load(file = paste(mData, newestFile, sep = "/"))
  temp <- paste(mData, newestFile, sep = "/")
  return(readRDS(temp))
}
```

Title getNewestVersionIMPACT @description read in a .RData file that includes the file fileShortName from the data/IMPACT directory

@param fileShortName The substantive (first) part of the file name.

@return The most recent .RData file of IMPACT data @export

```
getNewestVersionIMPACT <- function(fileShortName) {
  iData <- fileloc("iData")
  # see
  # http://stackoverflow.com/questions/7381641/regex-matching-beginning-and-end-strings
  # for an explanation of this regex expression
  # regExp <- paste("(?=^", fileShortName, ")(?=.*RData$)", sep = "")
  regExp <- paste("(?=^", fileShortName, ")(?=.*rds$)", sep = "")
  filesList <-
    grep(regExp,
         list.files(iData),
         value = TRUE,
         perl = TRUE)
  newestFile <- filesList[length(filesList)]
#  return(load(file = paste(iData, newestFile, sep = "/")))
  return(readRDS(paste(iData, newestFile, sep = "/")))
}
```

Title removeOldVersionsIMPACT Remove old versions of a file with name fileShortName in the data/IMPACT directory @param fileShortName The substantive (first) part of the file name. @return nothing. Just deletes old files @export

```
removeOldVersionsIMPACT <- function(fileShortName) {
  # returns a list of all the [fileShortName] files in the iData
  # directory
  iData <- fileloc("iData")
#  regExp <- paste("(?=^", fileShortName, ")(?=.*RData$)", sep = "")
  regExp <- paste("(?=^", fileShortName, ")(?=.*rds$)", sep = "")
  oldVersionList <-
    grep(regExp,
         list.files(iData),
         value = TRUE,
         perl = TRUE)
  if (length(oldVersionList) > 0) {
    file.remove(paste(iData, oldVersionList, sep = "/"))
  }
}
```

Title removeOldVersions - removes old version of an RData file

@param fileShortName - short name of the file to be removed @export

```
removeOldVersions <- function(fileShortName) {
  mData <- fileloc("mData")
  # returns a list of all the [fileShortName] files in the mData
  # directory
#  regExp <- paste("(?=^", fileShortName, ")(?=.*RData$)", sep = "")
  regExp <- paste("(?=^", fileShortName, ")(?=.*rds$)", sep = "")
  oldVersionList <-
    grep(regExp,
         list.files(mData),
```

```
        value = TRUE,
        perl = TRUE)
  if (length(oldVersionList) > 0) {
    file.remove(paste(mData, oldVersionList, sep = "/"))
  }
}
```

Title removeOldVersions.xlsx - remove old xlsx versions in preparation for writing out new ones

@param fileShortName - short name of the files to be removed

@export

```
removeOldVersions.xlsx <- function(fileShortName) {
  mData <- fileloc("mData")
  # returns a list of all the [fileShortName] files in the mData
  # directory
  regExp <- paste("(?=^", fileShortName, ")(?=.*xlsx$)", sep = "")
  oldVersionList <-
    grep(regExp,
         list.files(mData),
         value = TRUE,
         perl = TRUE)
  if (length(oldVersionList) > 0) {
    file.remove(paste(mData, oldVersionList, sep = "/"))
  }
}
```

Title cleanup - remove old versions and save RData and xlsx versions of the file

@param inName - name of the data table or frame to be written out @param outName - short name of the file to be written out

```
cleanup <- function(inName, outName) {
  mData <- fileloc("mData")
  removeOldVersions(outName)
  removeOldVersions.xlsx(outName)
  temp <- eval(parse(text = inName), 1)
  # save(temp,
  #      file = paste(mData, "/", outName, ".", Sys.Date(), ".RData", sep = ""))
  saveRDS(temp,
       file = paste(mData, "/", outName, ".", Sys.Date(), ".rds", sep = ""))
  openxlsx::write.xlsx(
    x = temp,
    file = paste(mData, "/", outName, ".", Sys.Date(), ".xlsx", sep = ""),
    colWidths = "auto",
    colNames = TRUE
  )
}
```

Title fileloc directory locations for files @param variableName Name of variable holding a path @param RData - raw data directory @param mData - main data directory @param iData - directory with IMPACT data @param resultsDir - directory for results @param FBSData - directory where FBS data are kept @param SSPData - the path to the SSP data directory @param IMPACTData - the path to the raw IMPACT data

directory @param IMPACTDataClean - the path to the cleaned up IMPACT data directory @return Value of the variableName to be assigned in another script @export

```
fileloc <- function(variableName) {
  RData <- "data-raw"
  mData <- "data"
  iData <- "data/IMPACTData"
  resultsDir <- "results"
  FBSData <- paste(RData, "FBSData", sep = "/")
  SSPData <- paste(RData, "SSPData", sep = "/")
  IMPACTData <- paste(RData, "IMPACTData", sep = "/")
  IMPACTDataClean <- paste(mData, "IMPACTData", sep = "/")
  NutrientData <- paste(RData, "NutrientData", sep = "/")
  if (variableName == "list") {
    return(c(
      "RData",
      "mData",
      "iData",
      "resultsDir",
      "FBSData",
      "SSPData"
    ))
  } else {
    return(eval(parse(text = variableName)))
  }
}
```

Title keyVariable - Return a ky variable, or a list of all possibilities

@param keepYearList - list of scenario years to keep @param keepYearList.FBS - list of FBS years to keep @param FBSyearsToAverage - years to average over for base data set @param IMPACTfish_code- variable name list for fish consumption items for IMPACT @param IMPACTalcohol_code - variable name list for alcoholic beverages consumption for IMPACT @param IMPACTfoodCommodList - variable name lists for IMPACT food commodities @return list of key variables @export

```
keyVariable <- function(variableName) {
  keepYearList <-
    c(
      "X2005",
      "X2010",
      "X2015",
      "X2020",
      "X2025",
      "X2030",
      "X2035",
      "X2040",
      "X2045",
      "X2050"
    )

  keepYearList.FBS <-
    c(
      "X2000",
      "X2001",
```

```
    "X2002",
    "X2003",
    "X2004",
    "X2005",
    "X2006",
    "X2007",
    "X2008",
    "X2009",
    "X2010",
    "X2011"
  )

FBSyearsToAverage <- c("X2004", "X2005", "X2006")

#' note shrimp, tuna, and salmon are removed in dataManagement.fish.R
IMPACTfish_code <-
  c(
    "c_shrimp",
    "c_Crust",
    "c_Mllsc",
    "c_Salmon",
    "c_FrshD",
    "c_Tuna",
    "c_OPelag",
    "c_ODmrsl",
    "c_OMarn",
    "c_FshOil",
    "c_aqan",
    "c_aqpl"
  )

IMPACTalcohol_code <- c("c_wine", "c_beer", "c_spirits")
IMPACTfoodCommodList <- sort(
  c(
    "cbeef",
    "cpork",
    "clamb",
    "cpoul",
    "ceggs",
    "cmilk",
    "cbarl",
    "cmaiz",
    "cmill",
    "crice",
    "csorg",
    "cwhea",
    "cocer",
    "ccass",
    "cpota",
    "cswpt",
    "cyams",
    "corat",
    "cbean",
```

```
    "cchkp",
    "ccowp",
    "clent",
    "cpigp",
    "copul",
    "cbana",
    "cplnt",
    "csubf",
    "ctemf",
    "cvege",
    "csugr",
    "cgrnd",
    "cgdol",
    "crpsd",
    "crpol",
    "csoyb",
    "csbol",
    "csnfl",
    "csfol",
    "cplol",
    "cpkol",
    "ctols",
    "ctool",
    "ccoco",
    "ccafe",
    "cteas",
    "cothr"
  )
)

#' @param scenarioList - list of scenarios in the SSP data
scenarioList <-
  c(
    "SSP1_v9_130325",
    "SSP2_v9_130325",
    "SSP3_v9_130325",
    "SSP4_v9_130325",
    "SSP5_v9_130325"
  )
if (variableName == "list") {
  return(
    c(
      keepYearList,
      keepYearList.FBS,
      FBSyearsToAverage,
      IMPACTfish_code,
      IMPACTalcohol_code,
      IMPACTfoodCommodList,
      scenarioList
    )
  )
} else{
  return(eval(parse(text = variableName)))
```

```
  }
}

  # create metadata file -----
  metadata <- function() {
    metadata <-
      data.frame(
        file_name_location = character(1),
        file_description = character(1),
        stringsAsFactors = FALSE
      )
    metadata[(nrow(metadata) + 1), ] <-
      c(EARs, "data on nutrient requirements")
    metadata[(nrow(metadata) + 1), ] <-
      c(
        "http://www.nal.usda.gov/fnic/DRI/DRI_Tables/recommended_intakes_individuals.pdf",
        "Source of EARS"
      )
    metadata[(nrow(metadata) + 1), ] <-
      c(CSEs, "Consumer Surplus Equivalents for IMPACT commodities")
    metadata[(nrow(metadata) + 1), ] <-
      c(IMPACT3regions,
        "List of IMPACT regions; single countries and country aggregates")
    metadata[(nrow(metadata) + 1), ] <-
      c(IMPACTstdRegions,
        "List of the standard IMPACT large grouping of countries")
    metadata[(nrow(metadata) + 1), ] <-
      c(IMPACTgdx, "IMPACT demand data in gdx form")
    metadata[(nrow(metadata) + 1), ] <-
      c(
        R_GAMS_SYSDIR,
        "Location and name of GAMS program; needed for the gdx data import process"
      )
    metadata[(nrow(metadata) + 1), ] <-
      c(IMPACTfish, "data on fish from the IMPACT fish model")
    # nutrient data ------
    metadata[(nrow(metadata) + 1), ] <-
      c(nutrientLU, "nutrient lookup data for IMPACT commodities")
    metadata[(nrow(metadata) + 1), ] <-
      c(foodGroupLU, "commodity to food group lookup table")
    # SSP information ----
    metadata[(nrow(metadata) + 1), ] <-
      c(SSPdataZip, "zip file containing the SSP data")
    metadata[(nrow(metadata) + 1), ] <-
      c(SSPcsv, "csv file inside the SSP zip file")
    metadata[(nrow(metadata) + 1), ] <-
      c(modelListPop,
        "List of SSP models to extract population info from")
    metadata[(nrow(metadata) + 1), ] <-
      c(modelListGDP,
        "List of SSP models to extract population info from")
    metadata[(nrow(metadata) + 1), ] <-
      c(SSP_DRI_ageGroupLU,
```

```
      "lookup tables for SSP to DRI age and gender groups")
  # FBS information ----
  metadata[(nrow(metadata) + 1), ] <-
    c(FBSdataZip, "Zip file containing the FBS data")
  metadata[(nrow(metadata) + 1), ] <-
    c("FBS data creation date", createDate)
  metadata[(nrow(metadata) + 1), ] <-
    c("FBS lookup table", FBSlookupTableLink)
  metadata[(nrow(metadata) + 1), ] <-
    c(FBSCommodityInfo,
      "File in the FBS zip file containing the FBS data")
  metadata[(nrow(metadata) + 1), ] <-
    c(
      FAOCountryNameCodeLookup,
      "Lookup table for FAOSTAT and other country identification"
    )
  metadata[(nrow(metadata) + 1), ] <-
    c(ISOCodes,
      "List of all ISO 3 codes and the names of the countries they represent")
  openxlsx::write.xlsx(
    metadata,
    file = paste(mData, "/inputFileList.", Sys.Date(), ".xlsx", sep = ""),
    colWidths = "auto"
  )
  save(metadata,
       file = paste(mData, "/inputFileList.", Sys.Date(), ".RData", sep = ""))
}

#' Title fileNameList returns a list of filenames, with or without complete paths
#'
#' @param EARFileName - the name of the spreadsheet with the EAR data
#' @param EARs - the path to and the name of the EAR data file
#' @param CSEFileName - the name of the file with consumer support equivalents (CSEs)
#' @param CSEs - the path to and the file name for the CSE data
#' @param IMPACT3regionsFileName - the file name with the IMPACT3 regions names
#' @param IMPACT3regions - the path to and the file name for the IMPACT3 regions names
#' @param IMPACTstdRegionsFileName - file name with IMPACT standard global regions
#' @param IMPACTstdRegions - path and file name for the list of IMPACT standard regions
#' @param IMPACTgdxfileName- file name with IMPACT demand results
#' @param IMPACTgdx - name and path to IMPACT demand results gdx file
#' @param gdxLib - path to gdx library
#' @param R_GAMS_SYSDIR - path to gdx library
#' @param IMPACTfishInfo - file name with info in IMPACT fish elasticities and quantities
#' @param IMPACTfish - path and file name for IMPACT fish elasticities and quantities
#' @param nutrientFileName - file name for nutrient lookup data
#' @param nutrientLU - path and file name for nutrient lookup data
#' @param commodityFoodGroupLookupFileName - file name for the commodity to food group lookup spreads
#' @param foodGroupLU - path and file name for the commodity to food group lookup
#' @param SSPdataZipFile - file name of the SSP data in zip format
#' @param SSPdataZip - path and file name for the SSP data zip file
#' @param SSPcsv - name of the SSP data file in the zip file
#' @param modelListPop - list of models (currently only one) for the population data
#' @param modelListGDP - list of models (currently only one) for the GDP data
```

```
#' @param SSP_DRI_ageGroupLUFileName - lookup tables for SSP to DRI age and gender groups
#' @param SSP_DRI_ageGroupLU - lookup tables for SSP to DRI age and gender groups
#' @source \url{http://faostat3.fao.org/download/FB/FBS/E} Source of FBS data
#' @return Nothing
#' @export

fileNameList <- function(variableName) {
  IMPACTData <- fileloc("IMPACTData")
  NutrientData <- fileloc("NutrientData")
  SSPData <- fileloc("SSPData")
  EARFileName <- "DRI_IOM_V2.xlsx"
  EARs <- paste(NutrientData, EARFileName, sep = "/")
  # CSE - consumer support equivalent
  #Note: the price a consumer pays is Pc * (1-CSE)
  CSEFileName <- "CSEs20150824.xlsx"
  CSEs <- paste(IMPACTData, CSEFileName, sep = "/")
  IMPACT3regionsFileName <-
    "IMPACTRegionsFeb2016.xlsx" # this file includes Denmark plus (DNP) and Sudan plus (SDP)
  #' IMPACT3regionsFileName <- "IMPACTRegionsMay2015.csv" # this file includes Denmark plus (DNP) and
  #' #IMPACT3regionsFileName <- "IMPACTRegionsJan15tmp.csv" # this file removes Denmark plus (DNP) an
  IMPACT3regions <-
    paste(IMPACTData, IMPACT3regionsFileName, sep = "/")
  IMPACTstdRegionsFileName <- "IMPACT-agg-regionsFeb2016.xlsx"
  IMPACTstdRegions <-
    paste(IMPACTData, IMPACTstdRegionsFileName, sep = "/")
  IMPACTgdxfileName <- "Demand Results20150817.gdx"
  IMPACTgdx <- paste(IMPACTData, IMPACTgdxfileName, sep = "/")
  gdxLib <- "/Applications/GAMS/gams24.5_osx_x64_64_sfx"
  R_GAMS_SYSDIR <- "/Applications/GAMS/gams24.5_osx_x64_64_sfx"
  IMPACTfishInfo <- "Fish Elasticities and Quantities IMPACT.xlsx"
  IMPACTfish <- paste(IMPACTData, IMPACTfishInfo, sep = "/")
  # nutrient data ------
  nutrientFileName <- "USDA GFS IMPACT V15.xlsx"
  nutrientLU <- paste(NutrientData, nutrientFileName, sep = "/")
  commodityFoodGroupLookupFileName <-
    "food commodity to food group table V2.xlsx"
  foodGroupLU <-
    paste(NutrientData, commodityFoodGroupLookupFileName, sep = "/")
  # SSP information ----
  SSPdataZipFile <- "SspDb_country_data_2013-06-12.csv.zip"
  SSPdataZip <- paste(SSPData, SSPdataZipFile, sep = "/")
  #get the name of the file inside the zip. Assumes only 1
  temp <-
    unzip(SSPdataZip, list = TRUE)
  SSPcsv <- temp$Name[1]
  modelListPop <- "IIASA-WiC POP"
  modelListGDP <- "OECD Env-Growth"
  SSP_DRI_ageGroupLUFileName <-  "SSP_DRI_ageGroupLookUp.xlsx"
  SSP_DRI_ageGroupLU <-
    paste(NutrientData, SSP_DRI_ageGroupLUFileName, sep = "/")
  if (variableName == "list") {
    #list of variables that can be returned
    return(
```

```
      c(
        "EARFileName",
        "EARs",
        "CSEFileName",
        "CSEs",
        "IMPACT3regionsFileName",
        "IMPACT3regions",
        "IMPACTstdRegionsFileName",
        "IMPACTstdRegions",
        "IMPACTgdxfileName",
        "IMPACTgdx",
        "gdxLib",
        "R_GAMS_SYSDIR",
        "IMPACTfishInfo",
        "IMPACTfish",
        "nutrientFileName",
        "nutrientLU",
        "commodityFoodGroupLookupFileName",
        "foodGroupLU",
        "SSPdataZipFile",
        "SSPdataZip",
        "SSPcsv",
        "modelListPop",
        "modelListGDP",
        "SSP_DRI_ageGroupLUFileName",
        "SSP_DRI_ageGroupLU"
      )
    )
  } else {
    return(eval(parse(text = variableName)))
  }
}

# Food Balance Sheet Information information ----
#'
#' Title filelocFBS Returns a list of files and paths for FBS-related data
#' @source  \url{http://www.fao.org/countryprofiles/iso3list/en/}
#' @param FBSdataZipFile - file name for the Food Balance Sheet data in zip file
#' @param FBSdataZip - path and file name for the Food Balance Sheet zip file
#' @param FBScsv - name of the FBS csv file contained the FBS zip file
#' @param FBSCommodityInfoFileName - worksheet with the list of FBS food items by code, name, definit
#' @param FBSCommodityInfo - path and file name to the worksheet with the list of FBS food items by c
#' @param FAOCountryNameCodeLookupFile - file with lookup info for FAO codes and others including ISO
#' @param FAOCountryNameCodeLookup - path and file name for the lookup spreadsheet
#' @param ISOCodesFile - file name with ISO country codes
#' @param ISOCodes - path and file name for the ISO country codes
#' @param FBSregionsToDrop - countries that do not have enough information or are large regions
#' @return Nothing
#' @export
filelocFBS <- function(variableName) {
  FBSData <- fileloc("FBSData")
  RData <- fileloc("RData")
  #' FBS to ISO lookup table
```

```
  FBSlookupTableLink <-
    "http://www.fao.org/countryprofiles/iso3list/en/"
  FBSdataZipFile <- "FoodBalanceSheets_E_All_Data.zip"
  FBSdataZip <- paste(FBSData, FBSdataZipFile, sep = "/")
  list <- unzip(FBSdataZip, list = TRUE)
  createDate <- as.character(list$Date[1])
  temp <-
    unzip(FBSdataZip, list = TRUE) #get the name of the file inside the zip. Assumes only 1
  FBScsv <- temp$Name[1]
  FBSCommodityInfoFileName <- "FBStoIMPACTlookupV3.xlsx"
  FBSCommodityInfo <-
    paste(FBSData, FBSCommodityInfoFileName, sep = "/")
  FAOCountryNameCodeLookupFile <- "FAOCountryNameCodeLookup.xlsx"
  FAOCountryNameCodeLookup <-
    paste(FBSData, FAOCountryNameCodeLookupFile, sep = "/")
  ISOCodesFile <- "ISOCountrycodes.xlsx"
  ISOCodes <- paste(RData, ISOCodesFile, sep = "/")

  #These regions are reported as their individual member countries during the relevant
  # time period (e.g. after 1999 for Belgium-Luxembourg). Their data entries are all NA.
  # Although Ethiopia PDR doesn't have data, Ethiopia does.
  FBSregionsToDrop <-
    c(
      "Belgium-Luxembourg",
      "Czechoslovakia",
      "Ethiopia PDR",
      "Montenegro",
      "Serbia",
      "Serbia and Montenegro",
      "Yugoslav SFR",
      "Europe",
      "Eastern Europe",
      "Southern Europe",
      "Western Europe",
      "European Union",
      "USSR",
      "World",
      "Netherlands Antilles (former)",
      "Caribbean"
    )
  if (variableName == "list") {
    return(
      #list of variables that can be returned
      c(
        "FBSdataZipFile",
        "FBSdataZip",
        "FBScsv",
        "FBSCommodityInfoFileName",
        "FBSCommodityInfo",
        "FAOCountryNameCodeLookupFile",
        "FAOCountryNameCodeLookup",
        "ISOCodesFile",
        "ISOCodes",
```

```
        "FBSregionsToDrop"
      )
    )
  } else {
    return(eval(parse(text = variableName)))
  }
}

#' Title flagMissingFiles Prints a list of missing files and a hint of how to address
#'
#' @return Nothing
#' @export
flagMissingFiles <- function() {
  shortNameList = data.frame(
    name = c("FBS", "regions.all", "SSPPopClean"),
    script = c("dataPrep.FBS.R", "dataPrep.regions.R", "dataPrep.SSP.R")
  )
  mData <- fileloc("mData")
  for (i in length(shortNameList)) {
    filesList <-
      grep(shortNameList$name[i], list.files(mData), value = TRUE)
    if (length(filesList) == 0) {
      rowNumber <- which(grepl(shortNameList$name[i], shortNameList$name))
      print(paste("Missing data file", shortNameList$name[i]))
      print(paste(" run R/", shortNameList$script[rowNumber], sep = ""))
      return()
    }
  }
}
```