nutrientModFunctions.R

gcn

Mon May 16 08:06:41 2016

Nutrient Modeling Functions

@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()
# }
```

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 resData - directory with results @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"
  resData <- "results"
  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",
      "resData",
      "resultsDir",
      "FBSData",
      "SSPData"
    ))
  } else {
    return(eval(parse(text = variableName)))
  }
}
```

getNewestVersion

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

@export

```
getNewestVersion <- function(fileShortName, directory) {
  if (missing(directory)) {mData <- fileloc("mData")} else {mData <- directory}
  # 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 .rds file that includes the file fileShortName from the data/IMPACTData directory @param fileShortName The substantive (first) part of the file name. @return The most recent .rds 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)]
  print(newestFile)
  #  return(load(file = paste(iData, newestFile, sep = "/")))
  return(readRDS(paste(iData, newestFile, sep = "/")))
}
```

Title removeOldVersions - removes old version of an RData file

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

```
removeOldVersions <- function(fileShortName,dir) {
  #  regExp <- paste("(?=^", fileShortName, ")(?=.*RData$)", sep = "")
  regExp <- paste("(?=^", fileShortName, ")(?=.*rds$)", sep = "")
  oldVersionList <-
    grep(regExp,
         list.files(dir),
         value = TRUE,
         perl = TRUE)
  if (length(oldVersionList) > 0) {
    file.remove(paste(dir, 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,dir) {
  #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(dir),
         value = TRUE,
         perl = TRUE)
  if (length(oldVersionList) > 0) {
    file.remove(paste(dir, oldVersionList, sep = "/"))
  }
}
```

Title cleanup - remove old versions and save rds and xlsx versions of the file @param inDT - name of the data table or frame to be written out @param outName - short name of the file to be written out @param dir - directory where the cleanup takes place

```
cleanup <- function(inDT, outName,dir, writeFiles) {

  #mData <- fileloc("mData")
  #convert inDT to a standard order
  print(paste("started cleanup for ", outName, sep = ""))
```

```r
print(proc.time())
flush.console()
oldOrder <- names(inDT)
startOrder <- c("scenario",keyVariable("region"),"year")
if (all(startOrder %in% oldOrder)) {
  remainder <- oldOrder[!oldOrder %in% startOrder]
  data.table::setcolorder(inDT,c(startOrder,remainder))
  data.table::setorderv(inDT,c(startOrder,remainder))
}
print(paste("removing old versions of ", outName, sep = ""))
print(proc.time())
flush.console()
removeOldVersions(outName,dir)
removeOldVersions.xlsx(outName,dir)
# save(inDT,
#   file = paste(dir, "/", outName, ".", Sys.Date(), ".RData", sep = ""))
print(paste("writing the rds for ", outName, " to ",dir, sep = ""))
print(proc.time())
flush.console()
saveRDS(inDT,
    file = paste(dir, "/", outName, ".", Sys.Date(), ".rds", sep = ""))

print(proc.time())
flush.console()
if (missing(writeFiles)) {writeFiles = "xlsx"}
if ("csv"  %in% writeFiles) {
    print(paste("write the csv for ", outName, " to ",dir, sep = ""))
    write.csv(inDT,file = paste(dir, "/", outName, ".", Sys.Date(), ".csv", sep = ""))
}
if (nrow(inDT) > 50000) {
    print (paste("number of rows in the data, ", nrow(inDT), ", greater than 50,000. Not writing xlsx
    writeFiles <- writeFiles[!writeFiles %in% "xlsx"]
}
if (!"xlsx"  %in% writeFiles) {
    print("not writing out xlsx file")
}
else {
  print(paste("write the xlsx for ", outName, " to ",dir, sep = ""))
  wbGeneral <- openxlsx::createWorkbook()
openxlsx::addWorksheet(wb = wbGeneral, sheetName = outName)

openxlsx::writeDataTable(
  wbGeneral,
  inDT,
  sheet = outName,
  startRow = 1,
  startCol = 1,
  rowNames = FALSE,
  colNames = TRUE,
  withFilter = TRUE
)

openxlsx::setColWidths(
    wbGeneral,
    sheet = outName,
    cols = 1:ncol(inDT),
    widths = "auto"
  )

  numStyle <- openxlsx::createStyle(numFmt = "0.00")
  openxlsx::addStyle(
    wbGeneral,
    sheet = outName,
    style = numStyle,
    rows = 1:nrow(inDT),
    cols = 2:ncol(inDT),
    gridExpand = TRUE
  )

  xcelOutFileName = paste(dir, "/", outName, ".", Sys.Date(), ".xlsx", sep = "")
  openxlsx::saveWorkbook(wbGeneral, xcelOutFileName, overwrite = TRUE)
  print(paste("done writing the xlsx for ", outName, sep = ""))
  print(proc.time())
  flush.console()
  }
}
```

Title keyVariable - Return a key 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 @param scenarioListSSP - list of scenarios in the SSP data @param DinY - number of days in a year @param reqSSP - nutrient requirements by SSP age groups @param ctyDeleteList @param useCookingRetnValues - apply the cooking retention values to the nutrient content @param userName - Name of person running the scripts and generating results @param region - Aggregation scheme from individual countries to regions @param commonList - names of the lists of nutrient names common to the nutrient lookup table and the requirements @return list of key variables @export

```
keyVariable <- function(variableName) {
  region <- "region_code.IMPACT3"
  keepYearList <-
    c(
      "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", IMPACTfish_code,
                                 IMPACTalcohol_code))

  scenarioListSSP <- c("SSP1_v9_130325", "SSP2_v9_130325", "SSP3_v9_130325",
                       "SSP4_v9_130325", "SSP5_v9_130325")

  DinY <-
    365 #see http://stackoverflow.com/questions/9465817/count-days-per-year for a way to deal with leap
  #' #' countries to remove because of poor data
  #' FSM - Micronesia, Federated States of
  #' GRD - Grenada
  #' PRK - Korea, Democratic People's Republic of

  reqSSP <- c("req.EAR.ssp", "req.RDA.vits.ssp","req.RDA.minrls.ssp", "req.RDA.macro.ssp","req.UL.vits.s
  commonList <- c( "common.EAR", "common.RDA.vits", "common.RDA.minrls", "common.RDA.macro", "common.UL
  ctyDeleteList <- c("FSM", "GRD", "PRK")
  useCookingRetnValues <- "yes"
  userName <- "Gerald C. Nelson"
  if (variableName == "list") {
    return(
      c(
        "region",
        "keepYearList",
        "keepYearList.FBS",
        "FBSyearsToAverage",
        "IMPACTfish_code",
        "IMPACTalcohol_code",
        "IMPACTfoodCommodList",
        "scenarioListSSP",
        "scenarioListIMPACT",
        "DinY",
        "reqSSP",
        "ctyDeleteList",
        "useCookingRetnValues",
        "commonList",
        "userName"
      )
    )
  } else{
    return(eval(parse(text = variableName)))
  }
}
```

```r
metadata <- function() {
  metadata <-
    data.frame(
      file_name_location = character(1),
      file_description = character(1),
      stringsAsFactors = FALSE
    )
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("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(fileNameList("CSEs"), "Consumer Surplus Equivalents for IMPACT commodities")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("IMPACT3regions"),
      "List of IMPACT regions; single countries and country aggregates")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("IMPACTstdRegions"),
      "List of the standard IMPACT large grouping of countries")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("IMPACTgdx"), "IMPACT demand data in gdx form")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("R_GAMS_SYSDIR"),
      "Location and name of GAMS program; needed for the gdx data import process"
    )
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("IMPACTfish"), "data on fish from the IMPACT fish model")
  # nutrient data -------
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("nutrientLU"), "nutrient lookup data for IMPACT commodities")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("foodGroupLU"), "commodity to food group lookup table")
  # SSP information ----
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("SSPdataZip"), "zip file containing the SSP data")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("SSPcsv"), "csv file inside the SSP zip file")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("modelListPop"),
      "List of SSP models to extract population info from")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("modelListGDP"),
      "List of SSP models to extract population info from")
  metadata[(nrow(metadata) + 1), ] <-
    c(fileNameList("SSP_DRI_ageGroupLU"),
      "lookup tables for SSP to DRI age and gender groups")
  # FBS information ----
  metadata[(nrow(metadata) + 1), ] <-
    c(filelocFBS("FBSdataZip"), "Zip file containing the FBS data")
  metadata[(nrow(metadata) + 1), ] <-
```

```
    c("FBS data creation date", filelocFBS("createDate"))
  metadata[(nrow(metadata) + 1), ] <-
    c("FBS lookup table", filelocFBS("FBSlookupTableLink"))
  metadata[(nrow(metadata) + 1), ] <-
    c(filelocFBS("FBSCommodityInfo"),
      "File in the FBS zip file containing the FBS data")
  metadata[(nrow(metadata) + 1), ] <-
    c(filelocFBS("FAOCountryNameCodeLookup"),
      "Lookup table for FAOSTAT and other country identification"
    )
  metadata[(nrow(metadata) + 1), ] <-
    c(filelocFBS("ISOCodes"),
      "List of all ISO 3 codes and the names of the countries they represent")

  inDT <- metadata
  outName <- "metaData"
  cleanup(inDT,outName,fileloc("resData"))
}
```

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 IMPACTalcoholInfo - file name with info in IMPACT alcohol elasticities @param IMPACTalcohol - path and file name for IMPACT alcohol elasticities and quantities @param IMPACTfood - path and file name for IMPACT food results @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 spreadsheet @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 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_V3.xlsx"
  mData <- fileloc("mData")
  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 S
#' #IMPACT3regionsFileName <- "IMPACTRegionsJan15tmp.csv" # this file removes Denmark plus (DNP) and
IMPACT3regions <-
  paste(IMPACTData, IMPACT3regionsFileName, sep = "/")
IMPACTstdRegionsFileName <- "IMPACT-agg-regionsFeb2016.xlsx"
IMPACTstdRegions <-
  paste(IMPACTData, IMPACTstdRegionsFileName, sep = "/")
# IMPACTgdxfileName <- "Micronutrient-Inputs20160404.gdx"  #- new larger gdx
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 = "/")
IMPACTalcoholInfo    <- "Alcohol Elasticities and Quantities IMPACT.xlsx"
IMPACTalcohol        <- paste(IMPACTData, IMPACTalcoholInfo, sep = "/")
IMPACTfoodFileName <- "dt.IMPACTfood"
IMPACTfoodFileInfo <-  paste(mData,"/IMPACTData/",IMPACTfoodFileName,sep="")
# nutrient data ------
nutrientFileName <- "USDA GFS IMPACT V16.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",
        "IMPACTfood",
        "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 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, definition, and IMPACT commodity code @param FBSCommodityInfo - path and file name to the worksheet with the list of FBS food items by code, name, definition, and IMPACT commodity code @param FAOCountryNameCodeLookupFile - file with lookup info for FAO codes and others including ISO3 @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 The content of the variable name. @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 plusCnst @param region_code.IMPACT - 3 letter code for the new region @param lst - list of countries that go in the region @param region_title - name of the region (eg., Denmark plus) @return data from with the information for an IMPACT region with multiple countries

```
# plusCnst <- function(region_code.IMPACT, ISO3_list, region_title) {
#   data.frame(region_code.IMPACT3, ISO3_lst, region_name.IMPACT3, stringsAsFactors = FALSE)}

plusCnst <- function(region_code.IMPACT3, ISO3_lst, region_name) {
  data.frame(region_code.IMPACT3, ISO3_lst, region_name, stringsAsFactors = FALSE)
}
```

Title createIMPACT3Regions @return regions.IMPACT3 @export

```
createIMPACT3Regions <- function() {
  #' regions.IMPACT3.plus is all the regions larger than a single political unit (as defined by an ISO3
  #' and what political units are included
  regions.IMPACT3.plus <- data.frame(
    region_code.IMPACT3 = character(0),
    region_members = character(0),
    region_name.IMPACT3 = character(0),
    stringsAsFactors = FALSE
  )
  #' @param region_code.IMPACT3 - temporary variable to hold countries that make up a region
  region_code.IMPACT3 <- "BLT"
  ISO3_lst <- c("EST", "LTU", "LVA")
  region_name.IMPACT3 <- "Baltic States"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('BLT Baltic States is Estonia EST, Lithuania LTU, Latvia
  # LVA')

  region_code.IMPACT3 <- "BLX"
  ISO3_lst <- c("BEL", "LUX")
  region_name.IMPACT3 <- "Belgium-Luxembourg"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('BLX Belgium-Luxembourg is Belgium BEL, Luxembourg LUX')

  region_code.IMPACT3 <- "CHM"
  ISO3_lst <- c("CHN", "HKG", "MAC", "TWN")
  region_name.IMPACT3 <- "China plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('CHM China plus is China CHN, Hong Kong HKG, Macao MAC,
  # Taiwan TWN')

  region_code.IMPACT3 <- "CHP"
  ISO3_lst <- c("CHE", "LIE")
  region_name.IMPACT3 <- "Switzerland plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('CHP Switzerland plus is Switzerland CHE Liechtenstein LIE')

  region_code.IMPACT3 <- "CRB"
  ISO3_lst <- c("ABW", "AIA", "ATG", "BES", "BHS", "BLM", "BRB", "CUW", "CYM",
                "DMA", "GLP", "GRD", "KNA", "LCA", "MAF", "MSR", "MTQ", "PRI", "SXM",
                "TCA", "TTO", "VCT", "VGB", "VIR")
  region_name.IMPACT3 <- "Other Caribbean"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('CRB Other Caribbean is Aruba ABW, Anguilla AIA, Netherlands
  # Antilles (obsolete) ANT, Antigua ATG Bonaire, Sint Eustatius, and
  # Saba BES, Bahamas BHS, St,Barthélemy BLM, Barbados BRB, Curacao CUW,
  # Cayman Islands CYM Dominica DMA, Guadeloupe GLP, Grenada GRD,
  # St,Kitts and Nevis KNA, St,Lucia LCA, Saint Martin MAF Montserrat
  # MSR, Martinique MTQ, Puerto Rico PRI, Sint Maarten SXM, Turks and
  # Caicos Islands TCA Trinidad and Tobago TTO, St,Vincent and Grenadines
  # VCT, British Virgin Islands VGB, U.S,Virgin Islands VIR') ANT dropped
  # from this list

  #DNP is commented out because the latest version of the IMPACT regions has Denmark and Greenland sepa
  # region_code.IMPACT3 <- "DNP"
  # ISO3_lst <- c("DNK", "GRL")
  # region_name.IMPACT3 <- "Denmark plus"
  # regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, plusCnst(region_code.IMPACT3,
  #                                                                    ISO3_list, region_name.IMPACT3))
  # txt <- c('DNP Denmark plus is DNK Denmark GRL Greenland')

  region_code.IMPACT3 <- "FNP"
  ISO3_lst <- c("ALA", "FIN")
  region_name.IMPACT3 <- "Finland plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('FNP Finland plus is Aland Islands ALA Finland FIN')

  region_code.IMPACT3 <- "FRP"
  ISO3_lst <- c("FRA", "MCO")
  region_name.IMPACT3 <- "France plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('FRP France plus is France FRA Monaco MCO')

  region_code.IMPACT3 <- "GSA"
  ISO3_lst <- c("GUF", "GUY", "SUR")
  region_name.IMPACT3 <- "Guyanas"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('GSA Guyanas is South America French Guiana GUF Guyana GUY
  # Suriname SUR')

  region_code.IMPACT3 <- "ITP"
  ISO3_lst <- c("ITA", "MLT", "SMR", "VAT")
  region_name.IMPACT3 <- "Italy plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('ITP Italy plus is Italy ITA Malta MLT San Marino SMR
  # Vatican City VAT')

  region_code.IMPACT3 <- "MOR"
  ISO3_lst <- c("MAR", "ESH")
  region_name.IMPACT3 <- "Morocco plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('MOR Morocco plus is Morocco MAR Western Sahara ESH')

  region_code.IMPACT3 <- "OAO"
  # Antartic (ATA) added to this list
  ISO3_lst <- c("ATA", "BMU", "BVT", "CPV", "FLK", "FRO", "SGS", "SHN", "SJM",
                "SPM", "STP")
  region_name.IMPACT3 <- "Other Atlantic Ocean"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('OAO Other Atlantic Ocean is Bermuda BMU Bouvet Island BVT
  # Cape Verde CPV Falkland Islands FLK Faroe Islands FRO South Georgia
  # and South Sandwich Islands SGS Saint Helena, Ascension, and Tristan
  # de Cunha SHN Svalbard and Jan Mayen SJM Saint Pierre and Miquelon SPM
  # Sao Tome and Principe STP')

  region_code.IMPACT3 <- "OBN"
  ISO3_lst <- c("BIH", "MKD", "MNE", "SRB")
  region_name.IMPACT3 <- "Other Balkans"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('OBN Other Balkans is Bosnia-Herzegovina BIH Macedonia (FYR)
  # MKD Montenegro MNE Serbia SRB')

  region_code.IMPACT3 <- "OIO"
  ISO3_lst <- c("ATF", "CCK", "COM", "CXR", "HMD", "IOT", "MDV", "MUS", "MYT",
                "REU", "SYC")
  region_name.IMPACT3 <- "Other Indian Ocean"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('OIO Other Indian Ocean is Southern Territories ATF Keeling
  # Islands CCK Comoros COM Christmas Island CXR Heard and McDonald
  # Islands HMD British Indian Ocean Territory IOT Maldives MDV Mauritius
  # MUS Mayotte MYT Réunion REU Seychelles SYC') CXR deleted from this
  # list

  region_code.IMPACT3 <- "OPO"
  ISO3_lst <- c("ASM", "COK", "FSM", "GUM", "KIR", "MHL", "MNP", "NCL", "NFK",
                "NIU", "NRU", "PCN", "PLW", "PYF", "TKL", "TON", "TUV", "UMI", "WLF",
                "WSM")
  region_name.IMPACT3 <- "Other Pacific Ocean"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('OPO Other Pacific Ocean is American Samoa ASM Cook Islands
  # COK Micronesia FSM Guam GUM Kiribati KIR Marshall Islands MHL
  # Northern Mariana Islands MNP New Caledonia NCL Norfolk Island NFK
  # Niue NIU Nauru NRU Pitcairn PCN Palau PLW French Polynesia PYF
  # Tokelau TKL Tonga TON Tuvalu TUV Minor Outlying Islands UMI Wallis
  # and Futuna WLF Samoa WSM')

  region_code.IMPACT3 <- "OSA"
  ISO3_lst <- c("BRN", "SGP")
  region_name.IMPACT3 <- "Other Southeast Asia"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('OSA OtherSoutheast Asia is Brunei BRN Singapore SGP')

  region_code.IMPACT3 <- "RAP"
  ISO3_lst <- c("ARE", "BHR", "KWT", "OMN", "QAT")
  region_name.IMPACT3 <- "Rest of Arab Peninsula"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('RAP Rest of Arab Peninsula is United Arab Emirates ARE
  # Bahrain BHR Kuwait KWT Oman OMN Qatar QAT')

  region_code.IMPACT3 <- "SDP"
```

```r
  ISO3_lst <- c("SSD", "SDN")
  region_name.IMPACT3 <- "Sudan plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('SDP Sudan plus is SSD Sudan SDN South Sudan')

  region_code.IMPACT3 <- "SPP"
  ISO3_lst <- c("AND", "ESP", "GIB")
  region_name.IMPACT3 <- "Spain plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('SPP Spain plus is Andorra AND Spain ESP Gibraltar GIB')

  region_code.IMPACT3 <- "UKP"
  ISO3_lst <- c("GBR", "GGY", "IMN")
  region_name.IMPACT3 <- "Great Britain plus"
  temp <- plusCnst(region_code.IMPACT3,ISO3_lst,region_name.IMPACT3)
  regions.IMPACT3.plus <- rbind(regions.IMPACT3.plus, temp)
  # txt <- c('UKP Great Britain plus is Great Britain GBR Guernsey GGY
  # Isle of Man IMN Jersey JEY')

  colnames(regions.IMPACT3.plus) <-
    c("region_code.IMPACT3", "ISO_code", "region_name.IMPACT3")

  # Create regions.IMPACT3 ----
  # The next lines of code get a list of IMPACT 3 regions that are not in IMPACT3.plus
  IMPACT3regions <- fileNameList("IMPACT3regions")
  regions.IMPACT3 <- openxlsx::read.xlsx(IMPACT3regions)
  colnames(regions.IMPACT3) <-
    c("region_code.IMPACT3", "region_name.IMPACT3")
  #' @param regions.IMPACT3.region_name.IMPACT3 regions in IMPACT3 that are only one country
  regions.IMPACT3.cty <-
    regions.IMPACT3[!regions.IMPACT3$region_code.IMPACT3 %in% regions.IMPACT3.plus$region_code.IMPACT3,]
  regions.IMPACT3.cty$ISO_code <-
    regions.IMPACT3.cty$region_code.IMPACT3
  regions.IMPACT3 <- rbind(regions.IMPACT3.cty, regions.IMPACT3.plus)
  regions.IMPACT3 <-
    regions.IMPACT3[order(regions.IMPACT3$ISO_code), ]
  temp <- regions.IMPACT3
  return(temp)
}
```

Title flagMissingFiles Prints a list of missing files and a hint of how to address

@return Nothing @export

```r
flagMissingFiles <- function() {
  shortNameList = data.frame(
    name = c("FBS", "df.regions.all", "dt.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()
    }
  }
}

# these functions return the maximum or minimum in every row
colMax <- function(dataIn) {
  lapply(dataIn, max, na.rm = TRUE)
}
colMin <- function(dataIn) {
  lapply(dataIn, min, na.rm = TRUE)
}
```