Annex IV - STREAM: Auxiliary scripts for the conversion from RCG Sampling and Landing tables to COST format

E. Mantzouni

Wed Jul 17 10:16:34 2019

Tools

R, Rstudio and packages.

```
#R general option:
options(stringsAsFactors = FALSE)

#chunk option
knitr::opts_chunk$set(cache=TRUE,echo=TRUE, warning=FALSE,
    message=FALSE, fig.height=6,progress=FALSE,verbose=FALSE,
        include=TRUE,dev='png',autodep=FALSE)

#load packages
library(reshape2)
library(reshape)
library(dplyr)
library(knitr)
library(pander)
library(data.table)
library(COSTcore)
library(COSTdbe)
library(COSTeda)

#pander options
panderOptions('table.split.table', 60)
panderOptions('table.style', 'grid')
panderOptions('table.split.cells', 10)
panderOptions('table.alignment.default', 'left')
panderOptions('table.alignment.rownames', 'right')
panderOptions('decimal.mark', ',')
panderOptions('graph.fontsize', '10')
```

script 01: *RCG_to_CS_COST_object*

This script allow to convert the RCG Data call format into the SDEF format as a CS COST object.

The RCG Data call format allows to provide the data according the following two modalities: 1) aggregated data related to sex, maturity and age by length class (leaving empty fish ID and

individual weight) 2) individual data related to length, sex, maturity, age and weight (necessarily providing the Fish ID to avoid duplicated records)

Settings

```
# set the working directory


myWD <- paste("C:\\Users\\Bitetto Isabella\\OneDrive - Coispa Tecnologia & Ricerca
S.C.A.R.L\\MARE22\\STREAM\\FINAL REVISION OF DELIVERABLES\\RCG_to_COST", sep="")

setwd(myWD)

log_varFilePath <- paste(getwd(), "/log_CS.csv", sep="")
path.data <- getwd()

  log_var  <- TRUE
  error <- FALSE

  fpKey <- function(tab, colIndex, sep = ":-:") {
    key <- tab[, colIndex]
    key <- apply(key, 1, paste, collapse = sep)
    key <- gsub("[[:space:]]", "", key)
    return(key)
  }

# CS can be in both above-mentioned modalities

CS <- read.csv(file="DPS_GSA99.csv",
               stringsAsFactors=FALSE,  sep=";")

#  CS <- read.csv(file="SDEF_STREAM_final_HKE_examples_NOid.csv",
# stringsAsFactors=FALSE,  sep=";")

dataset_proj <- "STREAM project"
```

Input Data

RCG Datacall format

Table continues below

Sampling.type	Flag.country	Year	Trip.code
S	COUNTRY1	1900	01_18_2017
S	COUNTRY1	1900	01_18_2017
S	COUNTRY1	1900	01_18_2017
S	COUNTRY1	1900	01_18_2017

S	COUNTRY1	1900	01_18_2017
S	COUNTRY1	1900	01_18_2017

Table continues below

Harbour	Number.of.sets…hauls.on.trip	Days.at.sea
ITBCE	5	1
ITBCE	5	1
ITBCE	5	1
ITBCE	5	1
ITBCE	5	1
ITBCE	5	1

Table continues below

Sampling.method	Aggregation.level	Station.number
SelfSampling	TRUE	999
SelfSampling	TRUE	999
SelfSampling	TRUE	999
SelfSampling	TRUE	999
SelfSampling	TRUE	999
SelfSampling	TRUE	999

Table continues below

Duration.of.fishing.operation	Initial.latitude
1200	NA
1200	NA
1200	NA
1200	NA
1200	NA
1200	NA

Table continues below

Initial.longitude	Final.latitude	Final.longitude
NA	NA	NA

NA	NA	NA
NA	NA	NA
NA	NA	NA
NA	NA	NA
NA	NA	NA

Table continues below

Depth.of.fishing.operation	Water.depth
65	NA
65	NA
65	NA
65	NA
65	NA
65	NA

Table continues below

Catch.registration	Species.registration	Date
Lan	All	08/01/1900
Lan	All	08/01/1900
Lan	All	08/01/1900
Lan	All	08/01/1900
Lan	All	08/01/1900
Lan	All	08/01/1900

Table continues below

Area	Fishing.activity.category.National
GSA99	OTB_shelf
GSA99	OTB_shelf
GSA99	OTB_shelf
GSA99	OTB_shelf
GSA99	OTB_shelf

GSA99 OTB_shelf

Table continues below

Fishing.activity.category.European.lvl.6	Species
OTB_DEF_>=40_0_0	Parapenaeus longirostris
OTB_DEF_>=40_0_0	Parapenaeus longirostris
OTB_DEF_>=40_0_0	Parapenaeus longirostris
OTB_DEF_>=40_0_0	Parapenaeus longirostris
OTB_DEF_>=40_0_0	Parapenaeus longirostris
OTB_DEF_>=40_0_0	Parapenaeus longirostris

Table continues below

Catch.category	Weight	Subsample.weight	Sex
Lan	6100	3050	M
Lan	6100	3050	F
Lan	6100	3050	F
Lan	6100	3050	M
Lan	6100	3050	F
Lan	6100	3050	F

Table continues below

Maturity.method	Maturity.Scale	Maturity.Stage
Macr	Medits scale	2d
Macr	Medits scale	2e
Macr	Medits scale	2e
Macr	Medits scale	2d
Macr	Medits scale	2e
Macr	Medits scale	2b

Table continues below

Ageing.method	Age	Length.code	Length.class
NA	NA	mm	18
NA	NA	mm	19

NA	NA	mm	20
NA	NA	mm	21
NA	NA	mm	21
NA	NA	mm	22

Table continues below

Number.at.length	Commercial.size.category.scale
1	COUNTRY1
1	COUNTRY1
2	COUNTRY1
1	COUNTRY1
1	COUNTRY1
1	COUNTRY1

Commercial.size.category	fish.ID	Individual.weight
1	NA	NA
1	NA	NA
1	NA	NA
1	NA	NA
1	NA	NA
1	NA	NA

Processing tables

Using some data in the DG MARE Med&BS format :

```
bad.rm <- FALSE

if (log_var) {
  log_varCs <- CS
  log_varCs$duplicated_TR <- FALSE
  log_varCs$duplicated_HH <- FALSE
  log_varCs$duplicated_SL <- FALSE
  log_varCs$duplicated_HL <- FALSE
  log_varMsg <- ""
}  else {
  log_varMsg <- " Consider using the log_var=TRUE parameter."
}
```

```r
names(CS)[which(tolower(names(CS)) == "sampling.type")] <- "sampType"
names(CS)[which(tolower(names(CS)) == "sampling_type")] <- "sampType"
names(CS)[which(tolower(names(CS)) == "samplingtype")] <- "sampType"
names(CS)[which(tolower(names(CS)) == "flag.country")] <- "vslFlgCtry"
names(CS)[which(tolower(names(CS)) == "flag_country")] <- "vslFlgCtry"
names(CS)[which(tolower(names(CS)) == "flagcountry")] <- "vslFlgCtry"
names(CS)[which(tolower(names(CS)) == "year")] <- "year"
names(CS)[which(tolower(names(CS)) == "trip.code")] <- "trpCode"
names(CS)[which(tolower(names(CS)) == "trip_code")] <- "trpCode"
names(CS)[which(tolower(names(CS)) == "tripcode")] <- "trpCode"
names(CS)[which(tolower(names(CS)) == "number.of.sets...hauls.on.trip")] <- "foNum
"
names(CS)[which(tolower(names(CS)) == "number_of_setshauls")] <- "foNum"
names(CS)[which(tolower(names(CS)) == "nsets")] <- "foNum"
names(CS)[which(tolower(names(CS)) == "days.at.sea")] <- "daysAtSea"
names(CS)[which(tolower(names(CS)) == "days_at_sea")] <- "daysAtSea"
names(CS)[which(tolower(names(CS)) == "daysatsea")] <- "daysAtSea"
names(CS)[which(tolower(names(CS)) == "sampling.method")] <- "sampMeth"
names(CS)[which(tolower(names(CS)) == "sampling_method")] <- "sampMeth"
names(CS)[which(tolower(names(CS)) == "aggregation.level")] <- "aggLev"
names(CS)[which(tolower(names(CS)) == "aggregation_level")] <- "aggLev"
names(CS)[which(tolower(names(CS)) == "aggregationlevel")] <- "aggLev"
names(CS)[which(tolower(names(CS)) == "station.number")] <- "staNum"
names(CS)[which(tolower(names(CS)) == "station_number")] <- "staNum"
names(CS)[which(tolower(names(CS)) == "stationnumber")] <- "staNum"
names(CS)[which(tolower(names(CS)) == "catch.registration")] <- "catReg"
names(CS)[which(tolower(names(CS)) == "catch_registration")] <- "catReg"
names(CS)[which(tolower(names(CS)) == "catchregistration")] <- "catReg"
names(CS)[which(tolower(names(CS)) == "species.registration")] <- "sppReg"
names(CS)[which(tolower(names(CS)) == "species_registration")] <- "sppReg"
names(CS)[which(tolower(names(CS)) == "speciesregistration")] <- "sppReg"
names(CS)[which(tolower(names(CS)) == "date")] <- "date"
names(CS)[which(tolower(names(CS)) == "area")] <- "area"
names(CS)[which(tolower(names(CS)) ==
                  "fishing.activity.category.national")] <- "foCatNat"
names(CS)[which(tolower(names(CS)) ==
                  "fishingactivitycategorynational")] <- "foCatNat"
names(CS)[which(tolower(names(CS)) == "fac_national")] <- "foCatNat"
names(CS)[which(tolower(names(CS)) ==
                  "fishing.activity.category.european.lvl.6")] <- "foCatEu6"
names(CS)[which(tolower(names(CS)) == "metier")] <- "foCatEu6"
names(CS)[which(tolower(names(CS)) == "fac_ec_lvl6")] <- "foCatEu6"
names(CS)[which(tolower(names(CS)) == "species")] <- "spp"
names(CS)[which(tolower(names(CS)) == "catch.category")] <- "catchCat"
names(CS)[which(tolower(names(CS)) == "catch_category")] <- "catchCat"
names(CS)[which(tolower(names(CS)) == "catchcategory")] <- "catchCat"
names(CS)[which(tolower(names(CS)) == "commercial.size.category")] <- "commCat"
names(CS)[which(tolower(names(CS)) == "commercial_size_category")] <- "commCat"
```

```r
names(CS)[which(tolower(names(CS)) == "weight")] <- "wt"
names(CS)[which(tolower(names(CS)) == "subsample.weight")] <- "subSampWt"
names(CS)[which(tolower(names(CS)) == "subsample_weight")] <- "subSampWt"
names(CS)[which(tolower(names(CS)) == "subsampleweight")] <- "subSampWt"
names(CS)[which(tolower(names(CS)) == "length.code")] <- "lenCode"
names(CS)[which(tolower(names(CS)) == "length_code")] <- "lenCode"
names(CS)[which(tolower(names(CS)) == "lengthcode")] <- "lenCode"
names(CS)[which(tolower(names(CS)) == "length.class")] <- "lenCls"
names(CS)[which(tolower(names(CS)) == "length_class")] <- "lenCls"
names(CS)[which(tolower(names(CS)) == "lengthclass")] <- "lenCls"
names(CS)[which(tolower(names(CS)) == "number.at.length")] <- "lenNum"
names(CS)[which(tolower(names(CS)) == "number_at_length")] <- "lenNum"
names(CS)[which(tolower(names(CS)) == "numberatlength")] <- "lenNum"


# new Fields RCG 2018 ------------------------------------------------


names(CS)[which(tolower(names(CS)) == "commercial.size.category.scale")] <- "commC
atScl"
names(CS)[which(tolower(names(CS)) == "commercial_size_category_scale")] <- "commC
atScl"
names(CS)[which(tolower(names(CS)) == "commercialsizecategoryscale")] <- "commCatS
cl"

names(CS)[which(tolower(names(CS)) == "fish.id")] <- "fishId"
names(CS)[which(tolower(names(CS)) == "fish_id")] <- "fishId"
names(CS)[which(tolower(names(CS)) == "fishid")] <- "fishId"

names(CS)[which(tolower(names(CS)) == "individual.weight")] <- "indWt"
names(CS)[which(tolower(names(CS)) == "individual_weight")] <- "indWt"
names(CS)[which(tolower(names(CS)) == "individualweight")] <- "indWt"

names(CS)[which(tolower(names(CS)) == "harbour")] <- "harbour"

names(CS)[which(tolower(names(CS)) == "duration.of.fishing.operation")] <- "foDur"
names(CS)[which(tolower(names(CS)) == "duration_of_fishing_operation")] <- "foDur"
names(CS)[which(tolower(names(CS)) == "durationoffishing_operation")] <- "foDur"

names(CS)[which(tolower(names(CS)) == "initial.latitude")] <- "latIni"
names(CS)[which(tolower(names(CS)) == "initial_latitude")] <- "latIni"
names(CS)[which(tolower(names(CS)) == "initiallatitude")] <- "latIni"

names(CS)[which(tolower(names(CS)) == "initial.longitude")] <- "lonIni"
names(CS)[which(tolower(names(CS)) == "initial_longitude")] <- "lonIni"

names(CS)[which(tolower(names(CS)) == "final.latitude")] <- "latFin"
names(CS)[which(tolower(names(CS)) == "final_latitude")] <- "latFin"
names(CS)[which(tolower(names(CS)) == "finallatitude")] <- "latFin"
```

```r
names(CS)[which(tolower(names(CS)) == "final.longitude")] <- "lonFin"
names(CS)[which(tolower(names(CS)) == "final_longitude")] <- "lonFin"
names(CS)[which(tolower(names(CS)) == "finallongitude")] <- "lonFin"

names(CS)[which(tolower(names(CS)) == "depth.of.fishing.operation")] <- "foDep"
names(CS)[which(tolower(names(CS)) == "depth_of_fishing_operation")] <- "foDep"
names(CS)[which(tolower(names(CS)) == "depthoffishingoperation")] <- "foDep"

names(CS)[which(tolower(names(CS)) == "water.depth")] <- "waterDep"
names(CS)[which(tolower(names(CS)) == "water_depth")] <- "waterDep"
names(CS)[which(tolower(names(CS)) == "waterdepth")] <- "waterDep"

names(CS)[which(tolower(names(CS)) == "sex")] <- "sex"

names(CS)[which(tolower(names(CS)) == "maturity.method")] <- "matMeth"
names(CS)[which(tolower(names(CS)) == "maturity_method")] <- "matMeth"
names(CS)[which(tolower(names(CS)) == "maturitymethod")] <- "matMeth"

names(CS)[which(tolower(names(CS)) == "maturity.scale")] <- "matScale"
names(CS)[which(tolower(names(CS)) == "maturity_scale")] <- "matScale"
names(CS)[which(tolower(names(CS)) == "maturityscale")] <- "matScale"


names(CS)[which(tolower(names(CS)) == "maturity.stage")] <- "matStage"
names(CS)[which(tolower(names(CS)) == "maturity_stage")] <- "matStage"
names(CS)[which(tolower(names(CS)) == "maturitystage")] <- "matStage"

names(CS)[which(tolower(names(CS)) == "ageing.method")] <- "ageMeth"
names(CS)[which(tolower(names(CS)) == "ageing_method")] <- "ageMeth"
names(CS)[which(tolower(names(CS)) == "ageingmethod")] <- "ageMeth"

names(CS)[which(tolower(names(CS)) == "age")] <- "age"


CS[is.na(CS[,])] <- -1


CS$aggLev <- as.character(CS$aggLev)
CS$aggLev[ CS$aggLev %in% c("t","TRUE",TRUE)] <- "T" ##!!

CS$spp <- unlist(lapply(CS$spp, function(x) paste(toupper(substring(x,1, 1)),
                                        tolower(substring(x, 2)), sep = "")))

CS <-CS %>% mutate(proj=dataset_proj,landCtry=vslFlgCtry) ##!!


trPk <- c("sampType", "vslFlgCtry", "year", "trpCode","proj","landCtry") ##!!
trOther <- c("foNum", "daysAtSea", "sampMeth",
             "harbour" ) # new : harbour
```

```r
hhPk <- c(trPk, "staNum")
hhOther <- c("aggLev", "catReg", "sppReg", "date", "area",
             "foCatNat", "foCatEu6",
             "foDur", "latIni", "lonIni", "latFin", "lonFin", "foDep", "waterDep")
# new

# modified respect to the previous version
slPk <- c(hhPk, "spp", "catchCat", "commCat", "commCatScl","sex") # new
slOther <- c("wt", "subSampWt", "lenCode")

#slPk <- c(hhPk, "spp", "catchCat", "commCat", "commCatScl","sex", "wt", "subSampW
t") # new
#slOther <- c( "lenCode")

hlPk <- c(slPk, "lenCls")
hlOther <- c("lenNum")

# modified respect to the previous version
#caPk <- c(hlPk, "age","area","fishId")
#caOther <- c( "matMeth",  "matScale" ,"matStage", "ageMeth" , "indWt", "lenCode")

caPk <- c(hlPk, "age","area","fishId", "matStage")
caOther <- c( "matMeth",  "matScale" , "ageMeth" , "indWt", "lenCode")

allFields <- c(caPk, trOther, hhOther, slOther, hlOther,caOther)
missingFields <- allFields[!allFields %in% names(CS)]

# check if all fields are used

# names(CS)[!names(CS) %in% allFields]

if (length(missingFields) > 0) {
  stop("Missing fields : ", paste(missingFields, collapse = ", ",
                                  sep = ""))
}

csTr <- unique(CS[, c(trPk, trOther)])

trPkV <- fpKey(csTr, trPk)

trPkVDup <- trPkV %in% trPkV[duplicated(trPkV)]
if (any(trPkVDup)) {
  print(trPkVDup)
  if (log_var) {
    log_varCs$duplicated_TR <- fpKey(CS, trPk) %in% trPkV[trPkVDup]
  }
  if (bad.rm) {
    message("Integrity problem for CS/TR, ", sum(trPkVDup),
            " row(s) removed.", log_varMsg)
```

```
    print("Removed following row(s):", quote = F)
    print(csTr[trPkVDup, ])
    CS <- merge(CS, csTr[!trPkVDup, ])
  }    else {
    error <- TRUE
    message("Integrity problem for CS/TR, ", sum(trPkVDup),
            " row(s) concerned.", log_varMsg)
  }
}
print(paste("No rows TR =",  nrow(csTr)), quote=F)

## [1] No rows TR = 69

csHh <- unique(CS[, c(hhPk, hhOther)])
hhPkV <- fpKey(csHh, hhPk)
hhPkVDup <- hhPkV %in% hhPkV[duplicated(hhPkV)]
if (any(hhPkVDup)) {
  if (log_var) {
    log_varCs$duplicated_HH <- fpKey(CS, hhPk) %in% hhPkV[hhPkVDup]
  }
  if (bad.rm) {
    message("Integrity problem for CS/HH, ", sum(hhPkVDup),
            " row(s) removed.", log_varMsg)
    print("Removed following row(s):", quote = F)
    print(csHh[hhPkVDup, ])
    CS <- merge(CS, csHh[!hhPkVDup, ])
  }
  else {
    error <- TRUE
    message("Integrity problem for CS/HH, ", sum(hhPkVDup),
            " row(s) concerned.", log_varMsg)
  }
}
print(paste("No rows HH =",  nrow(csHh)), quote=F)

## [1] No rows HH = 69

csSl <- unique(CS[, c(slPk, slOther)])
slPkV <- fpKey(csSl, slPk)
slPkVDup <- slPkV %in% slPkV[duplicated(slPkV)]
if (any(slPkVDup)) {
  if (log_var) {
    log_varCs$duplicated_SL <- fpKey(CS, slPk) %in% slPkV[slPkVDup]
  }
  if (bad.rm) {
    message("Integrity problem for CS/SL, ", sum(slPkVDup),
            " row(s) removed.", log_varMsg)
    print("Removed following row(s):", quote = F)
    print(csSl[slPkVDup, ])
    CS <- merge(CS, csSl[!slPkVDup, ])
  }
  else {
```

```
    error <- TRUE
    message("Integrity problem for CS/SL, ", sum(slPkVDup),
            " row(s) concerned.", log_varMsg)
  }
}
print(paste("No rows SL =",  nrow(csSl)), quote=F)

## [1] No rows SL = 413

 CS_aggregated_by_length <-  aggregate(CS$lenNum, by=list(CS$sampType , CS$vslFlgC
try ,
                               CS$year,  CS$trpCode, CS$proj, CS$landCtry, CS$harbou
r,
                               CS$foNum, CS$daysAtSea ,  CS$sampMeth, CS$aggLev, CS$
staNum,
                               CS$foDur, CS$foDep, CS$catReg, CS$sppReg, CS$date, CS
$area,
                               CS$foCatNat, CS$foCatEu6, CS$spp,  CS$catchCat, CS$wt
,
                               CS$subSampWt, CS$sex, CS$lenCode, CS$lenCls,
                               CS$commCatScl, CS$commCat), FUN="sum")

 colnames(CS_aggregated_by_length) <- c("sampType" , "vslFlgCtry" , "year" , "trpC
ode" ,
                              "proj" , "landCtry","harbour" , "foNum" , "daysAtSea"
,
                              "sampMeth" ,  "aggLev" , "staNum" , "foDur" , "foDep"
,
                              "catReg", "sppReg" , "date" , "area" , "foCatNat" ,
                              "foCatEu6",  "spp" , "catchCat" , "wt" , "subSampWt" ,
                              "sex" , "lenCode" , "lenCls" , "commCatScl" , "commCat
",
                               "lenNum")

# sum(CS$lenNum)
# sum(CS_aggregated_by_length$lenNum)

csHl <- unique(CS_aggregated_by_length[, c(hlPk, hlOther)])
hlPkV <- fpKey(csHl, hlPk)
hlPkVDup <- hlPkV %in% hlPkV[duplicated(hlPkV)]
if (any(hlPkVDup)) {
  if (log_var) {
    log_varCs$duplicated_HL <- fpKey(CS, hlPk) %in% hlPkV[hlPkVDup]
  }
  else {
    error <- TRUE
    message("Integrity problem for CS/HL, ", sum(hlPkVDup),
            " row(s) concerned.", log_varMsg)
    print("Check the following row(s):", quote = F)
    csHl[hlPkVDup, ]
  }
```

```
}
print(paste("No rows HL =",  nrow(csHl)), quote=F)

## [1] No rows HL = 4383

##

#   # check CA ------------------------------------------------------------

csCa <- unique(CS[, c(caPk, caOther)])
caPkV <- fpKey(csCa, caPk)
caPkVDup <- caPkV %in% caPkV[duplicated(caPkV)]
if (any(caPkVDup)) {
  if (log_var) {
    log_varCs$duplicated_CA <- fpKey(CS, caPk) %in% caPkV[caPkVDup]
  }
  if (bad.rm) {
    message("Integrity problem for CS/CA, ", sum(caPkVDup),
            " row(s) removed.", log_varMsg)
    print("Removed following row(s):", quote = F)
    print(csCa[caPkVDup, ])
    CS <- merge(CS, csCa[!caPkVDup, ])
  } else {
    error <- TRUE
    message("Integrity problem for CS/CA, ", sum(caPkVDup),
            " row(s) concerned.", log_varMsg)
    print("Check the following row(s):", quote = F)
    csCa[caPkVDup, ]
  }
}

##

if (log_var) {
  if (missing(log_varFilePath)) {
    log_varFilePath <- tempfile(fileext = ".csv")
  }
  write.table(log_varCs, file = log_varFilePath, row.names = FALSE,
              sep = ";")
}
if (error) {
  print("See errors on log_var file!", quote = F)
  message("log_var file: ", log_varFilePath)
}



# All CS tables names -------------------------------------------------

##!! Simpler way to define names for the CS tables
obj <- new("csData")
```

```
tr0 <- obj@tr
hh0 <- obj@hh
sl0 <- obj@sl
hl0 <- obj@hl
ca0 <- obj@ca

TR.col <- names(tr0)
HH.col <- names(hh0)
SL.col <- names(sl0)
HL.col <- names(hl0)
CA.col <- names(ca0)

## end names ##!!

if (!error) {

  missTR<-TR.col[!TR.col %in% names(csTr)]
  costCS.TR <- csTr
  costCS.TR[ ,missTR] <- NA
  costCS.TR<-  costCS.TR[,TR.col]

  write.table(costCS.TR, file.path(path.data, "SDEF CS-TR data.csv"),
              sep = ";", row.names = FALSE)

  missHH<-HH.col[!HH.col %in% names(csHh)]
  costCS.HH <- csHh
  costCS.HH[ , missHH] <- NA

  costCS.HH<-  costCS.HH[,HH.col]

  write.table(costCS.HH, file.path(path.data, "SDEF CS-HH data.csv"),
              sep = ";", row.names = FALSE)


  missSL<-SL.col[!SL.col %in% names(csSl)]
  costCS.SL <- csSl
  costCS.SL[ , missSL] <- NA

  costCS.SL<-  costCS.SL[,SL.col]

  write.table(costCS.SL, file.path(path.data, "SDEF CS-SL data.csv"),
              sep = ";", row.names = FALSE)


  missHL<-HL.col[!HL.col %in% names(csHl)]
  costCS.HL <- csHl
  costCS.HL[ , missHL] <- NA

  costCS.HL <-  costCS.HL[,HL.col]
```

```
  write.table(costCS.HL, file.path(path.data, "SDEF CS-HL data.csv"),
              sep = ";", row.names = FALSE)

  if (all(CS$fishId == -1)) {

    ## add subSampWt (SL) and lenNum (HL) to CA

    ca.sl.hl.COL<-c(caPk, caOther,"subSampWt","lenNum")
    ca.sl.hl <- unique(CS[, ca.sl.hl.COL])

    # indWt
    ca.sl.hl<-ca.sl.hl %>% mutate(indWt=subSampWt/lenNum)
    ca.sl.hl1 <- ca.sl.hl[rep(row.names(ca.sl.hl), ca.sl.hl$lenNum), 1:ncol(ca.sl.
hl)]
    ca.sl.hl1$fishId <- 1:nrow(ca.sl.hl1)
    costCS.CA <- ca.sl.hl1[, names(  ca.sl.hl1) %in% CA.col]

    missCA<-CA.col[!CA.col %in% names(costCS.CA)]
    costCS.CA[ , missCA] <- NA
    costCS.CA<-  costCS.CA[,CA.col]
    costCS.CA$indWt <- -1

  } else {

    missCA <- CA.col[!CA.col %in% names(csCa)]
    costCS.CA <- csCa
    costCS.CA[ , missCA] <- NA

    costCS.CA$proj<- dataset_proj
    costCS.CA$landCtry<-costCS.CA$vslFlgCtry
    costCS.CA<-  costCS.CA[,CA.col]

    costCS.CA$fishId=seq(1,dim(costCS.CA)[1],by=1)

  }

  print(paste("No rows CA =",  nrow(costCS.CA)), quote=F)

  write.table(costCS.CA, file.path(path.data, "SDEF CS-CA data.csv"),
              sep = ";", row.names = FALSE)

  costCS = csData(tr =costCS.TR, hh = costCS.HH, sl = costCS.SL,
                  hl = costCS.HL, ca=costCS.CA)

  saveRDS(costCS, "costCS.rds")

} else {

  print("An error occurred in the trasformation.
```

```
        Impossible to save the CS COST object!")
}

## [1] No rows CA = 80632
```

Output

CS COST object

```
## An object of class "csData"
## Slot "desc":
## [1] "Unknown stock"
##
## Slot "tr":
##     sampType landCtry vslFlgCtry year           proj     trpCode vslLen
## 1          S COUNTRY1   COUNTRY1 1900 STREAM project  01_18_2017     NA
## 83         S COUNTRY1   COUNTRY1 1900 STREAM project  02_18_2017     NA
## 204        S COUNTRY1   COUNTRY1 1900 STREAM project  03_18_2017     NA
## 357        S COUNTRY1   COUNTRY1 1900 STREAM project  09_18_2017     NA
## 391        S COUNTRY1   COUNTRY1 1900 STREAM project 101_18_2017     NA
## 530        S COUNTRY1   COUNTRY1 1900 STREAM project 110_18_2017     NA
##     vslPwr vslSize vslType harbour foNum daysAtSea vslId sampCtry
## 1       NA      NA    <NA>   ITBCE     5         1    NA     <NA>
## 83      NA      NA    <NA>   ITBDS     2         1    NA     <NA>
## 204     NA      NA    <NA>   ITBDS     4         1    NA     <NA>
## 357     NA      NA    <NA>   ITMFR    11         2    NA     <NA>
## 391     NA      NA    <NA>   ITMNP     4         1    NA     <NA>
## 530     NA      NA    <NA>   ITMFR    11         2    NA     <NA>
##         sampMeth
## 1   SelfSampling
## 83      Observer
## 204     Observer
## 357 SelfSampling
## 391     Observer
## 530 SelfSampling
##
## Slot "hh":
##     sampType landCtry vslFlgCtry year           proj     trpCode staNum
## 1          S COUNTRY1   COUNTRY1 1900 STREAM project  01_18_2017    999
## 83         S COUNTRY1   COUNTRY1 1900 STREAM project  02_18_2017    999
## 204        S COUNTRY1   COUNTRY1 1900 STREAM project  03_18_2017    999
## 357        S COUNTRY1   COUNTRY1 1900 STREAM project  09_18_2017    999
## 391        S COUNTRY1   COUNTRY1 1900 STREAM project 101_18_2017    999
## 530        S COUNTRY1   COUNTRY1 1900 STREAM project 110_18_2017    999
##     foVal aggLev catReg sppReg       date time foDur latIni lonIni latFin
## 1    <NA>      T    Lan    All 08/01/1900 <NA>  1200     -1     -1     -1
## 83   <NA>      T    All    All 21/01/1900 <NA>   510     -1     -1     -1
## 204  <NA>      T    All    All 31/01/1900 <NA>  1155     -1     -1     -1
## 357  <NA>      T    Lan    All 29/01/1900 <NA>  2880     -1     -1     -1
## 391  <NA>      T    All    All 17/01/1900 <NA>   960     -1     -1     -1
## 530  <NA>      T    Lan    All 22/01/1900 <NA>  1980     -1     -1     -1
##     lonFin  area rect subRect foDep waterDep  foCatNat foCatEu5
```

```
## 1       -1 GSA99 <NA>    <NA>    65       -1 OTB_shelf     <NA>
## 83      -1 GSA99 <NA>    <NA>   188       -1 OTB_shelf     <NA>
## 204     -1 GSA99 <NA>    <NA>   115       -1 OTB_shelf     <NA>
## 357     -1 GSA99 <NA>    <NA>   148       -1 OTB_shelf     <NA>
## 391     -1 GSA99 <NA>    <NA>    81       -1 OTB_shelf     <NA>
## 530     -1 GSA99 <NA>    <NA>   104       -1 OTB_shelf     <NA>
##             foCatEu6 meshSize selDev meshSizeSelDev
## 1   OTB_DEF_>=40_0_0       NA   <NA>             NA
## 83  OTB_DEF_>=40_0_0       NA   <NA>             NA
## 204 OTB_DEF_>=40_0_0       NA   <NA>             NA
## 357 OTB_DEF_>=40_0_0       NA   <NA>             NA
## 391 OTB_DEF_>=40_0_0       NA   <NA>             NA
## 530 OTB_DEF_>=40_0_0       NA   <NA>             NA
##
## Slot "sl":
##    sampType landCtry vslFlgCtry year           proj    trpCode staNum
## 1         S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 2         S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 33        S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 36        S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 60        S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 61        S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
##                         spp catchCat landCat commCatScl commCat subSampCat
## 1  Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
## 2  Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
## 33 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       2       <NA>
## 36 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       2       <NA>
## 60 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       3       <NA>
## 61 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       3       <NA>
##    sex    wt subSampWt lenCode
## 1    M  6100      3050      mm
## 2    F  6100      3050      mm
## 33   M 38080      2720      mm
## 36   F 38080      2720      mm
## 60   F  5600      1400      mm
## 61   M  5600      1400      mm
##
## Slot "hl":
##   sampType landCtry vslFlgCtry year           proj     trpCode staNum
## 1        S COUNTRY1   COUNTRY1 1900 STREAM project  14_19_2017    999
## 2        S COUNTRY1   COUNTRY1 1900 STREAM project  02_19_2017    999
## 3        S COUNTRY1   COUNTRY1 1900 STREAM project  38_19_2017    999
## 4        S COUNTRY1   COUNTRY1 1900 STREAM project  02_19_2017    999
## 5        S COUNTRY1   COUNTRY1 1900 STREAM project 125_18_2017    999
## 6        S COUNTRY1   COUNTRY1 1900 STREAM project  14_19_2017    999
##                         spp catchCat landCat commCatScl commCat subSampCat
## 1 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
## 2 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
## 3 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
## 4 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
## 5 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
```

```
## 6 Parapenaeus longirostris      Lan    <NA>   COUNTRY1       1       <NA>
##   sex lenCls lenNum
## 1   F     11      6
## 2   F     11      3
## 3   M     11      1
## 4   F     12      8
## 5   F     12      1
## 6   M     12      2
##
## Slot "ca":
##     sampType landCtry vslFlgCtry year           proj    trpCode staNum
## 1          S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 2          S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 3          S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 3.1        S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 4          S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
## 5          S COUNTRY1   COUNTRY1 1900 STREAM project 01_18_2017    999
##     quarter month                      spp sex catchCat landCat commCatScl
## 1        NA    NA Parapenaeus longirostris   M      Lan    <NA>   COUNTRY1
## 2        NA    NA Parapenaeus longirostris   F      Lan    <NA>   COUNTRY1
## 3        NA    NA Parapenaeus longirostris   F      Lan    <NA>   COUNTRY1
## 3.1      NA    NA Parapenaeus longirostris   F      Lan    <NA>   COUNTRY1
## 4        NA    NA Parapenaeus longirostris   M      Lan    <NA>   COUNTRY1
## 5        NA    NA Parapenaeus longirostris   F      Lan    <NA>   COUNTRY1
##     commCat stock  area rect subRect lenCls age fishId lenCode ageMeth
## 1         1  <NA> GSA99 <NA>    <NA>     18  -1      1      mm      -1
## 2         1  <NA> GSA99 <NA>    <NA>     19  -1      2      mm      -1
## 3         1  <NA> GSA99 <NA>    <NA>     20  -1      3      mm      -1
## 3.1       1  <NA> GSA99 <NA>    <NA>     20  -1      4      mm      -1
## 4         1  <NA> GSA99 <NA>    <NA>     21  -1      5      mm      -1
## 5         1  <NA> GSA99 <NA>    <NA>     21  -1      6      mm      -1
##     plusGrp otoWt otoSide indWt matMeth     matScale matStage
## 1      <NA>    NA    <NA>    -1    Macr Medits scale       2d
## 2      <NA>    NA    <NA>    -1    Macr Medits scale       2e
## 3      <NA>    NA    <NA>    -1    Macr Medits scale       2e
## 3.1    <NA>    NA    <NA>    -1    Macr Medits scale       2e
## 4      <NA>    NA    <NA>    -1    Macr Medits scale       2d
## 5      <NA>    NA    <NA>    -1    Macr Medits scale       2e
```

script 02: *RCG_CL_to_COST_CL*

This script allow to convert the RCG Landing table into CL COST object

Settings

```
# set the working directory
myWD <- paste("C:\\Users\\Bitetto Isabella\\OneDrive - Coispa Tecnologia & Ricerca
S.C.A.R.L\\MARE22\\STREAM\\FINAL REVISION OF DELIVERABLES\\RCG_to_COST", sep="")

logFilePath <- paste(getwd(), "/log_CL.csv", sep="")
path.data <- getwd()
```

```
setwd(myWD)

log=TRUE

CL <- read.csv(file="DPS_GSA99_LANDINGS.csv", stringsAsFactors=FALSE,  sep=";")

dataset_proj <- "STREAM project"


 fpKey <- function(tab, colIndex, sep = ":-:") {
    key <- tab[, colIndex]
    key <- apply(key, 1, paste, collapse = sep)
    key <- gsub("[[:space:]]", "", key)
    return(key)
 }
```

Input Data

RCG Datacall format

Table continues below

landCtry	vslFlgCtry	year	quarter	month
NA	COUNTRY1	1900	1	1
NA	COUNTRY1	1900	1	2
NA	COUNTRY1	1900	1	3
NA	COUNTRY1	1900	2	4
NA	COUNTRY1	1900	2	5
NA	COUNTRY1	1900	2	6

Table continues below

area	rect	subRect	taxon	landCat
GSA99	NA	NA	Parapenaeus longirostris	NA
GSA99	NA	NA	Parapenaeus longirostris	NA
GSA99	NA	NA	Parapenaeus longirostris	NA
GSA99	NA	NA	Parapenaeus longirostris	NA
GSA99	NA	NA	Parapenaeus longirostris	NA
GSA99	NA	NA	Parapenaeus longirostris	NA

Table continues below

commCatScl	commCat	foCatNat	foCatEu5
NA	NA	OTB_shelf	NA
NA	NA	OTB_shelf	NA
NA	NA	OTB_shelf	NA
NA	NA	OTB_shelf	NA
NA	NA	OTB_shelf	NA
NA	NA	OTB_shelf	NA

Table continues below

foCatEu6	harbour	vslLenCat
OTB_DEF_>=40_0_0	Port	NA
OTB_DEF_>=40_0_0	Port	NA
OTB_DEF_>=40_0_0	Port	NA
OTB_DEF_>=40_0_0	Port	NA
OTB_DEF_>=40_0_0	Port	NA
OTB_DEF_>=40_0_0	Port	NA

Table continues below

unallocCatchWt	misRepCatchWt	landWt	landMult
NA	NA	73453	NA
NA	NA	78742	NA
NA	NA	82021	NA
NA	NA	89023	NA
NA	NA	103912	NA
NA	NA	102987	NA

landValue
372659
392665
460281
433524
494103

498299

Processing tables

Using some data in the DG MARE Med&BS format :

```
 if (log) {
logCl <- CL
logCl$duplicated <- FALSE
logMsg <- ""
} else {
logMsg <- " Consider using the log=TRUE parameter."
}
error <- FALSE


names(CL)[which(tolower(names(CL)) == "flag.country")] <- "vslFlgCtry"
names(CL)[which(tolower(names(CL)) == "flag_country")] <- "vslFlgCtry"

names(CL)[which(tolower(names(CL)) == "year")] <- "year"
names(CL)[which(tolower(names(CL)) == "quarter")] <- "quarter"
names(CL)[which(tolower(names(CL)) == "month")] <- "month"
names(CL)[which(tolower(names(CL)) == "area")] <- "area"
names(CL)[which(tolower(names(CL)) == "species")] <- "taxon"

names(CL)[which(tolower(names(CL)) ==
                  "fishing.activity.category.national")] <- "foCatNat"
names(CL)[which(tolower(names(CL)) == "fac_national")] <- "foCatNat"

names(CL)[which(tolower(names(CL)) ==
                  "fishing.activity.category.european.lvl.6")] <- "foCatEu6"
names(CL)[which(tolower(names(CL)) == "fac_ec_lvl6")] <- "foCatEu6"

names(CL)[which(tolower(names(CL)) == "harbour")] <- "harbour"

names(CL)[which(tolower(names(CL)) == "official.landings.weight")] <- "landWt"
names(CL)[which(tolower(names(CL)) == "official_landings_weight")] <- "landWt"

names(CL)[which(tolower(names(CL)) == "official.landings.value")] <- "landValue"
names(CL)[which(tolower(names(CL)) == "official_landings_value")] <- "landValue"


## primary keys & fields

clPk <- c("vslFlgCtry", "year", "quarter", "month", "area", "taxon",
          "foCatNat", "foCatEu6")
clOther <- c("landWt", "landValue")

# check fields
allFields <- c(clPk, clOther)
```

```
missingFields <- allFields[! allFields %in% names(CL)]
if (length(missingFields) > 0) {
stop("Missing fields : ", paste(missingFields, collapse = ", ", sep=""))
}

 clPkV <- fpKey(CL, clPk)
clPkVDup <- clPkV %in% clPkV[duplicated(clPkV)]

# test integrity
if (any(clPkVDup)) {
if (log) {
logCl$duplicated <-  fpKey(CL, clPk)
}


if (bad.rm) {
message("Integrity problem for CL, ", sum(clPkVDup),
" row(s) removed.", logMsg)
CL <- CL[! clPkVDup,]
} else {
error <- TRUE
message("Integrity problem for CL, ", sum(clPkVDup),
" row(s) concerned.", logMsg)
}
}




if (log) {
if (missing(logFilePath)) {
logFilePath <- tempfile(fileext = ".csv")
}
write.table(logCl, file=logFilePath, row.names = FALSE, sep=";")
message("Log file: ", logFilePath)
}
if (error) {
stop("Stop on reported errors.")
}


# formating

CL$taxon <- unlist(lapply(CL$taxon, function(x)
paste(toupper(substring(x, 1, 1)), tolower(substring(x, 2)), sep="")))


## df
clDf <- data.frame(
landCtry=NA,
vslFlgCtry=CL$vslFlgCtry,
```

```
year=CL$year,
quarter=CL$quarter,
month=CL$month,
area=CL$area,
rect=NA,
subRect=NA,
taxon=CL$taxon,
landCat=NA,
commCatScl=NA,
commCat=NA,
foCatNat=CL$foCatNat,
foCatEu5=NA,
foCatEu6=CL$foCatEu6,
harbour=CL$harbour,
vslLenCat=NA,
unallocCatchWt=NA,
misRepCatchWt=NA,
landWt=CL$landWt,
landMult=NA,
landValue=CL$landValue,
stringsAsFactors=FALSE)

write.table(clDf, file.path(path.data,  "SDEF CL data.csv"), sep=";", row.names =
FALSE)

costCL = clData(cl=clDf)
saveRDS(costCL, "costCL.rds")
```

Output

CL COST object

```
## An object of class "clData"
## Slot "desc":
## [1] "Unknown stock"
##
## Slot "cl":
##   landCtry vslFlgCtry year quarter month  area rect subRect
## 1     <NA>   COUNTRY1 1900       1     1 GSA99 <NA>    <NA>
## 2     <NA>   COUNTRY1 1900       1     2 GSA99 <NA>    <NA>
## 3     <NA>   COUNTRY1 1900       1     3 GSA99 <NA>    <NA>
## 4     <NA>   COUNTRY1 1900       2     4 GSA99 <NA>    <NA>
## 5     <NA>   COUNTRY1 1900       2     5 GSA99 <NA>    <NA>
## 6     <NA>   COUNTRY1 1900       2     6 GSA99 <NA>    <NA>
##                      taxon landCat commCatScl commCat  foCatNat foCatEu5
## 1 Parapenaeus longirostris    <NA>       <NA>    <NA> OTB_shelf     <NA>
## 2 Parapenaeus longirostris    <NA>       <NA>    <NA> OTB_shelf     <NA>
## 3 Parapenaeus longirostris    <NA>       <NA>    <NA> OTB_shelf     <NA>
## 4 Parapenaeus longirostris    <NA>       <NA>    <NA> OTB_shelf     <NA>
## 5 Parapenaeus longirostris    <NA>       <NA>    <NA> OTB_shelf     <NA>
## 6 Parapenaeus longirostris    <NA>       <NA>    <NA> OTB_shelf     <NA>
```

```
##           foCatEu6 harbour vslLenCat unallocCatchWt misRepCatchWt
## 1 OTB_DEF_>=40_0_0    Port      <NA>             NA             NA
## 2 OTB_DEF_>=40_0_0    Port      <NA>             NA             NA
## 3 OTB_DEF_>=40_0_0    Port      <NA>             NA             NA
## 4 OTB_DEF_>=40_0_0    Port      <NA>             NA             NA
## 5 OTB_DEF_>=40_0_0    Port      <NA>             NA             NA
## 6 OTB_DEF_>=40_0_0    Port      <NA>             NA             NA
##      landWt landMult landValue
## 1  73452.53       NA  372658.8
## 2  78741.52       NA  392665.2
## 3  82021.10       NA  460280.8
## 4  89022.59       NA  433524.2
## 5 103911.92       NA  494103.4
## 6 102987.30       NA  498298.8
```