

Extract to create cSV used to feed FGP/OGSL/ETC

Pablo Vergara

February 27 2024

This document is the narrative of how an Excel file called “sGSL-Northumberland-Survey-FGP.csv” was generated.

The DFO Gulf Region Northumberland survey follows a stratified random sampling design and covers Division 4T of the Northwest Atlantic Fisheries Organisation (NAFO).

```
`%nin%` = Negate(`%in%`) ## a useful operator
suppressPackageStartupMessages(library(gulf))
survey = "ns"
```

SETS

Using the function “read.card” from the DFO Gulf Region’s R package “gulf”, read in the set cards and exclude hydrographic stations and null sets, so as to keep only representative tows in strata 401 to 439.

```
yrs <- 1999:2023 #
projectBase = here::here()
x <- read.card(card.type="set", year=yrs, survey=survey)
```

Data is being retrieved from the PTRAN database

```
index <- x$experiment == 1
x <- x[which(index),]

valid.cruise<-c('0901','0024','0139','0241','0341','0434','0536','0637','0030','0022','0029',
               '0103','0129','0026','P126','P021','P521','P018','P024','P041','P140',
               'P150','P151','P152','P303')###only cruise numbers from July/August

index <- paste0(x$vessel.code, x$cruise.number) %in% valid.cruise
x = x[which(index),]
```

Add a few useful columns to the data frame containing the set card information (including depth and swept area as requested).

```
## add useful columns
x$unique.id <- paste(x$year, x$cruise.number, x$vessel.code, x$set.number, sep="-")
x$experiment.str <- experiment.str(x$experiment)
x$gear.str <- gear.str(x$gear)
```

```
## Data is being retrieved from the PTRAN database
```

```
x$longitude <- longitude(x)
x$latitude <- latitude(x)

ox <- order(x$year, x$month, x$day, x$start.hour)
x <- x[ox,] # reorder chronologically
```

Catch cards contain the total catch information for the species of interest. Here they are adjusted for distance towed, estimated diurnal effects and estimated vessel-gear effects.

```
### Catch card for all years requested (1999 - 2023)
y <- read.card(card.type="catch", year = yrs, survey = survey)
```

```
## Data is being retrieved from the PTRAN database
```

```
index <- y$experiment == 1
y <- y[which(index),]
index <- paste0(y$vessel.code, y$cruise.number) %in% valid.cruise
y = y[which(index),]
```

CATCH

```
y <- adjust(y, x)
```

```
## Data is being retrieved from the PTRAN database
```

```
## Tow catches for nephrops trawl adjusted to a standard tow distance of 0.125 nautical miles.
```

```
y$unique.id <- paste(y$year, y$cruise.number, y$vessel.code, y$set.number, sep="-")
y$english.name <- species.str(y$species, "english")
```

```
## Data is being retrieved from the PTRAN database
```

```
y$latin.name <- species.str(y$species, "latin")
```

```
## Data is being retrieved from the PTRAN database
```

```
y$french.name <- species.str(y$species, "french")
```

```
## Data is being retrieved from the PTRAN database
```

```
y[y$number.caught == 0,]$number.caught = NA # as per request by Natalie

z <- merge(y,x, all.x = TRUE, by = "unique.id",
          names = c("longitude","latitude", "gear.str"))
```

```

## CSV
## write catch cards to file
fn2 <- "sGSL-Northumberland-Survey-FGP.csv"
fp <- paste0(projectBase, "/stock_assessment_surveys/6d61f7b4 (ns_survey)/")

csv.fn2 <- file.path(fp, fn2)

ooz <- order(z$year, z$month, z$day, z$start.hour, z$start.minute, z$species)

#columns to keep
fvars2 <- c("cruise.number", "year", "month", "day", "start.hour", "start.minute",
            "latitude", "longitude", "gear.str", "species", "french.name", "english.name",
            "latin.name", "weight.caught", "number.caught")

zz= z[ooz, fvars2]

#proper headers for FGP
names(zz) = c("cruise_number__numero_de_croisiere", "year__annee", "month__mois", "day__jour", "start_hour__",
             "start_minute__minute_de_depart", "latitude", "longitude", "gear__equipement", "species__esp",
             "french_name__nom_francais", "english_name__nom_anglais", "latin_name__nom_latin",
             "weight_caught__poids_pris", "number_caught__quantite_attrape")

con<-file(csv.fn2, encoding="ISO-8859-1")
write.csv(zz, file=con, row.names=FALSE)

```