Fisheries Dependent Information data call 2019

Notice explicative

Mathieu Depetris

13/08/2019

Contents

1 Introduction et présentation **1**

2 Explication du script d'importations et de manipulations des données **2**

2.1 Initialisation des packages R et de l'environnement de travail 2

2.2 Table A: Catch summary 3

2.3 Table D: Discards length data 7

2.4 Table F: Landings length data 7

2.5 Table G: Effort summary 11

2.6 Table H: Landings by rectangle 12

2.7 Table I: Effort by rectangle 14

2.8 Table J: Capacity and fleet segment effort 14

3 Script de verification des données **15**

4 Explications du script d'exportation des tables **16**

1 Introduction et présentation

L'appel à données "Fisheries Dependent Information" (FDI ou New-FDI) a pour but l'incrémentation d'une base de données, afin d'apporter un support dans la gestion et la coordination des pêcheries. En 2018, un appel à données similaire avait été lancé par la Commission Européenne (via le biais du STECF, "Scientific, Technical and Economic Committee for Fisheries"). Chaque année, un groupe de travail apporte des modifications à ce format, en rapport avec les retours d'expériences des années précédentes. Actuellement, le format n'est pas validé et des modifications sont encore apportées chaque année. C'est d'ailleurs pour cette raison que les données des années déjà récupérées précédemment sont encore demandées durant l'appel à données en cours. A terme, uniquement la dernière année sera intégrée avec une vocation à recréer des séries historiques (et donc à fournir des données anciennes). A titre d'information, le prochain groupe de travail sur cet appel à données (EWG 19-11) aura lieu le 16-20 septembre 2019.

Concernant l'appel à données de 2019, la Commission Européenne demande aux etats membres de fourir des données en rapport avec les années 2015, 2016, 2017 et 2018. Toutes les informations et les détails sont disponibles sur la plateforme de l'appel à données à l'adresse suivante https://datacollection.jrc.ec.europa.eu/dc/fdi. Le format et la spécification des données sont décrits dans le document "Annex 1" disponible sur la plateforme.

2 Explication du script d'importations et de manipulations des données

L'objectif de cette section est d'apporte une aide dans la compréhension des procédures ayant permis la livraison des données. Il est important de noté de certains points (marqué dans le document par "**Point de vigilance**") devront presenter une attention tout particulière pour l'appel à données de l'année prochaine. Ces points peuvent porter des processus à améliorer, à corriger ou à verifier.

2.1 Initialisation des packages R et de l'environnement de travail

```
# Setup ----
library(devtools)
install_github("https://github.com/OB7-IRD/furdeb")
library(furdeb)
library(tidyverse)
library(RPostgreSQL)
```

Le package furdeb (FUnctions for R DEvelopment and Beyond!) regroupe des fonctions diverses utilisables sous R. Ce package est en accès libre et disponible à l'adresse suivante : https://github.com/OB7-IRD/furdeb.

Point de vigilance : il est conseillé de mettre à jour les packages précédemment cités (notamment tidyverse) afin de bénéficier des dernières versions en vigueur. A titre d'exemple, il a été noté une instabilité avec certaines versions antérieures du package "dyplr" (chargé avec la librairie "tidyverse") s'il n'était pas mise à jour. Par ailleurs, il faut toujours vérifier que le package "furdeb" ne dispose pas d'une version plus récente disponible. L'utilisation de la fonction "install.github" devrait automatiquement lancer une mise à jour si elle est disponible.

```
config_file <- configuration_file(new_configtype = F,
    path_configtype = "...\\configfile_fdi2019.csv")

fao_area <- rgdal::readOGR(dsn = system.file("fao_area",
    "FAO_AREAS.shp", package = "furdeb"),
    verbose = FALSE)
```

Point de vigilance : l'utilisation de la fonction "configuration_file" (issu du package "furdeb") permet de créer un fichier de configuration regroupant toutes les informations nécessaires à charger l'environnement de travail (emplacement de java et mémoire allouée à celui-ci, localisation du répertoire de travail, identifiants permettant de se connecter aux bases de données, ...). Lors du premier lancement, la création de se fichier de configuration se fait à l'aide la fonction "configuration_file()". Regarder l'aide de la fonction (?configuration_file) pour en savoir plus.

```
# Databases connections ----
balbaya_con <- db_connection(db_user = config_file[["balbaya_user"]],
    db_password = config_file[["balbaya_password"]],
    db_dbname = config_file[["balbaya_dbname"]],
    db_host = config_file[["balbaya_host"]],
    db_port = config_file[["balbaya_port"]])

sardara_con <- db_connection(db_user = config_file[["sardara_user"]],
    db_password = config_file[["sardara_password"]],
```

```
    db_dbname = config_file[["sardara_dbname"]],
    db_host = config_file[["sardara_host"]],
    db_port = config_file[["sardara_port"]])
```

Point de vigilance : Il est nécessaire de lancer le script dans l'ordre d'appartion des commandes. Certaines tables ont besoin des sorties précédentes pour être crées.

2.2 Table A: Catch summary

```
balbaya_landing_query <- paste(readLines(con = file.path(config_file[["queries_loc"]],
    "balbaya_landing.sql", fsep = "\\")),
    collapse = "\n")

balbaya_landing <- dbGetQuery(balbaya_con,
    balbaya_landing_query)

observe_bycatch <- NULL
for (file_name in list.files(path = file.path(config_file[["work_path"]],
    "3-Data\\by_catch", fsep = "\\"))) {
    observe_bycatch <- rbind(observe_bycatch,
        read.csv2(file.path(config_file[["work_path"]],
            "3-Data\\by_catch", file_name,
            fsep = "\\"), stringsAsFactors = FALSE))
}
rm(file_name)
```

Point de vigilance : Cette année, les données provenant de la base ObServe ont été récupéré par Philippe S. via des fichiers csv (en rapport avec les appels à données fournis aux ORGPs et surtout afin d'appliquer les mêmes processus de manipulation de données). Pour l'année suivant, il est vivement recommandé de récupérer les données brutes directement depuis la base de données ObServe et de retracer et documenter les manipulations effectuées sur les données.

```
# By-catch design
observe_bycatch <- cwp_to_center(data = observe_bycatch,
    cwp_name = "cwp11", cwp_length = 1)

observe_bycatch <- fao_area_overlay(data = observe_bycatch,
    overlay_level = "division", longitude_name = "longitude_dec",
    latitude_name = "latitude_dec")

observe_bycatch <- fao_area_overlay_unassociated(data = observe_bycatch[is.na(observe_bycatch$f_subarea
    ], overlay_level = "division", longitude_name = "longitude_dec",
    latitude_name = "latitude_dec") %>% mutate(f_area = f_area_near,
    f_subarea = f_subarea_near, f_division = f_division_near) %>%
    select(-f_area_near, -f_subarea_near,
        -f_division_near) %>% rbind(observe_bycatch[!is.na(observe_bycatch$f_subarea),
    ])

if (dim(as.data.frame(observe_bycatch[observe_bycatch$f_subarea ==
    "far_away", ]))[1] != 0) {
    stop("You have NA(s) value(s)", "\n",
```

```
        "Checking data")
}

observe_bycatch_fao_area <- unique(observe_bycatch[,
    c("f_area", "f_subarea", "f_division")]) %>%
    rowwise() %>% mutate(f_division_final = ifelse(f_subarea ==
    "34.2", "34.2.0", f_division), f_subarea_final = ifelse(f_area ==
    "34", f_division_final, f_subarea), eez_indicator = ifelse(f_area %in%
    c("41", "47", "51", "57"), "NA", ifelse(f_area ==
    "34", ifelse(f_subarea_final == "34.1.1",
    "COAST", ifelse(f_subarea_final %in%
        c("34.1.2", "34.1.3", "34.2.0"),
        "RFMO", "NA")), "eez_not_defined")))

if (dim(as.data.frame(observe_bycatch_fao_area[observe_bycatch_fao_area$eez_indicator ==
    "eez_not_defined", ]))[1] != 0) {
    stop("You have NA(s) value(s)", "\n",
        "Checking data")
}

observe_bycatch <- observe_bycatch %>% inner_join(observe_bycatch_fao_area,
    by = c("f_area", "f_subarea", "f_division")) %>%
    mutate(sub_region = f_subarea_final,
        country = "FRA", vessel_length = "VL40XX",
        fishing_tech = "PS", gear_type = "PS",
        target_assemblage = "LPF", mesh_size_range = "NK",
        metier = "PS_LPF_0_0_0", supra_region = "OFR",
        geo_indicator = "IWE", specon_tech = "NA",
        deep = "NA", species = as.character(fao_code),
        retained_tons = ifelse(is.na(retained_tons),
            0, retained_tons))

observe_bycatch_retained <- observe_bycatch[observe_bycatch$retained_tons !=
    0, ]

observe_bycatch$metier_1 <- str_extract(string = observe_bycatch$metier,
    pattern = regex(pattern = "[^_]*"))
observe_bycatch$metier_2 <- str_extract(string = observe_bycatch$metier,
    pattern = regex(pattern = "(?<=_)\\w+"))
observe_bycatch$metier_3 <- str_c(observe_bycatch$metier_1,
    observe_bycatch$school_type, sep = "-")
observe_bycatch$metier_4 <- str_c(observe_bycatch$metier_3,
    observe_bycatch$metier_2, sep = "_")

observe_bycatch$domain_discards <- str_c(observe_bycatch$country,
    observe_bycatch$quarter, observe_bycatch$sub_region,
    observe_bycatch$metier_4, observe_bycatch$vessel_length,
    observe_bycatch$species, "NA", sep = "_")

observe_bycatch <- select(.data = observe_bycatch,
    -cwp, -quadrat, -latitude_dec, -longitude_dec,
    -f_area, -f_subarea, -f_division, -metier_1,
    -metier_2, -metier_3, -metier_4, -fao_code) %>%
```

```
    rename(discards = discarded_tons) %>%
    mutate(fishing_mode = as.character(school_type)) %>%
    group_by(country, year, quarter, vessel_length,
        fishing_tech, gear_type, target_assemblage,
        mesh_size_range, metier, fishing_mode,
        domain_discards, supra_region, sub_region,
        eez_indicator, geo_indicator, specon_tech,
        deep, species) %>% summarise(retained_tons = sum(retained_tons),
    discards = sum(discards)) %>% ungroup()

observe_bycatch <- observe_bycatch[!(observe_bycatch$retained_tons ==
    0 & observe_bycatch$discards == 0), ]

# Landings design
balbaya_landing <- fao_area_overlay(data = balbaya_landing,
    overlay_level = "division", longitude_name = "longitude",
    latitude_name = "latitude")

balbaya_landing <- fao_area_overlay_unassociated(data = balbaya_landing[is.na(balbaya_landing$f_subarea
    ], overlay_level = "division", longitude_name = "longitude",
    latitude_name = "latitude") %>% mutate(f_area = f_area_near,
    f_subarea = f_subarea_near, f_division = f_division_near) %>%
    select(-f_area_near, -f_subarea_near,
        -f_division_near) %>% rbind(balbaya_landing[!is.na(balbaya_landing$f_subarea),
    ])

balbaya_landing_fao_area <- unique(balbaya_landing[,
    c("f_area", "f_subarea", "f_division")]) %>%
    rowwise() %>% mutate(f_division_final = ifelse(f_subarea ==
    "34.2", "34.2.0", f_division), f_subarea_final = ifelse(f_area ==
    "34", f_division_final, f_subarea), eez_indicator = ifelse(f_area %in%
    c("41", "47", "51", "57"), "NA", ifelse(f_area ==
    "34", ifelse(f_subarea_final == "34.1.1",
    "COAST", ifelse(f_subarea_final %in%
        c("34.1.2", "34.1.3", "34.2.0"),
        "RFMO", "NA")), "eez_not_defined")))

balbaya_landing <- balbaya_landing %>% inner_join(balbaya_landing_fao_area,
    by = c("f_area", "f_subarea", "f_division")) %>%
    mutate(sub_region = f_subarea_final)

if (dim(as.data.frame(balbaya_landing[is.na(balbaya_landing$sub_region),
    ]))[1] != 0) {
    stop("You have NA(s) value(s)", "\n",
        "Checking data")
} else {
    if ("eez_not_defined" %in% unique(balbaya_landing$eez_indicator)) {
        stop("You have at least one eez not defined",
            "\n", "Checking data and code above")
    } else {
        balbaya_landing_rectangle <- select(.data = balbaya_landing,
            -f_area, -f_subarea, -f_division,
            -fishing_mode, -f_division_final,
```

```
            -f_subarea_final)
        balbaya_landing <- select(.data = balbaya_landing,
            -latitude, -longitude, -f_area,
            -f_subarea, -f_division, -f_division_final,
            -f_subarea_final)
    }
}

balbaya_landing$metier_1 <- str_extract(string = balbaya_landing$metier,
    pattern = regex(pattern = "[^_]*"))
balbaya_landing$metier_2 <- str_extract(string = balbaya_landing$metier,
    pattern = regex(pattern = "(?<=_)\\w+"))
balbaya_landing$metier_3 <- str_c(balbaya_landing$metier_1,
    balbaya_landing$fishing_mode, sep = "-")
balbaya_landing$metier_4 <- str_c(balbaya_landing$metier_3,
    balbaya_landing$metier_2, sep = "_")
balbaya_landing$domain_landings <- str_c(balbaya_landing$country,
    balbaya_landing$quarter, balbaya_landing$sub_region,
    balbaya_landing$metier_4, balbaya_landing$vessel_length,
    balbaya_landing$species, "NA", sep = "_")

balbaya_landing <- balbaya_landing %>% group_by(country,
    year, quarter, vessel_length, fishing_tech,
    gear_type, target_assemblage, mesh_size_range,
    metier, fishing_mode, domain_landings,
    supra_region, sub_region, eez_indicator,
    geo_indicator, specon_tech, deep, species) %>%
    summarise(totwghtlandg = sum(totwghtlandg)) %>%
    ungroup()

# Final design
tablea_final <- balbaya_landing %>% full_join(observe_bycatch,
    by = c("country", "year", "quarter",
        "vessel_length", "fishing_tech",
        "gear_type", "target_assemblage",
        "mesh_size_range", "metier", "fishing_mode",
        "supra_region", "sub_region", "eez_indicator",
        "geo_indicator", "specon_tech", "deep",
        "species")) %>% mutate(discards = ifelse(fishing_tech ==
    "HOK", "NK", ifelse(is.na(discards),
    0, round(discards, 3))), retained_tons = ifelse(is.na(retained_tons),
    0, retained_tons), totwghtlandg = ifelse(is.na(totwghtlandg),
    0, totwghtlandg), totwghtlandg = round(totwghtlandg +
    retained_tons, 3), domain_landings = ifelse(is.na(domain_landings),
    domain_discards, domain_landings), domain_discards = ifelse(discards ==
    0, domain_landings, ifelse(discards ==
    "NK", "NK", domain_discards)), confidential = ifelse(fishing_tech ==
    "HOK", "Y", "N"), totvallandg = ifelse(totwghtlandg ==
    0, 0, "NK")) %>% select(country, year,
    quarter, vessel_length, fishing_tech,
    gear_type, target_assemblage, mesh_size_range,
    metier, domain_discards, domain_landings,
    supra_region, sub_region, eez_indicator,
```

```
    geo_indicator, specon_tech, deep, species,
    totwghtlandg, totvallandg, discards,
    confidential)

names(tablea_final) <- toupper(names(tablea_final))
```

2.3 Table D: Discards length data

```
observer_discard <- NULL
for (file_name in list.files(path = file.path(config_file[["work_path"]],
    "3-Data\\discards", fsep = "\\"))) {
    observer_discard <- rbind(observer_discard,
        read.csv2(file.path(config_file[["work_path"]],
            "3-Data\\discards", file_name,
            fsep = "\\"), stringsAsFactors = FALSE))
}
rm(file_name)

tabled_final <- observer_discard %>% select(-totwghtlandg,
    -discards) %>% inner_join(tablea_final[,
    c("COUNTRY", "YEAR", "DOMAIN_DISCARDS",
        "SPECIES", "TOTWGHTLANDG", "DISCARDS")],
    by = c(country = "COUNTRY", year = "YEAR",
        domain_discards = "DOMAIN_DISCARDS",
        species = "SPECIES")) %>% select(country,
    year, domain_discards, species, TOTWGHTLANDG,
    DISCARDS, no_samples, no_length_measurements,
    length_unit, min_length, max_length,
    length, no_length)

names(tabled_final) <- toupper(names(tabled_final))
```

Point de vigilance : ici on devrait lors de la jointure des deux tables, on devrait avoir aucune perte du point de vue des rejets observés. Cependant on peut constater que certains rejets ne sont associés à aucune données de la table A (d'ou l'utilisation d'un "inner_join"). L'année prochaine, il faudra étudier cela de plus prés et voir si le problème est toujours présent lors de l'utilisation des données depuis la base de données ObServe (sans passer par l'intermédiaire des extractions de Philippe S.).

2.4 Table F: Landings length data

```
# By-catch retained data from table A
observe_bycatch_retained_tabled <- observe_bycatch_retained %>%
    group_by(country, year, quarter, sub_region,
        metier, school_type, vessel_length,
        species) %>% summarise(retained_tons = sum(retained_tons)) %>%
    ungroup() %>% rename(fishing_mode = school_type) %>%
    mutate(fishing_mode = as.character(fishing_mode))

# Landings
```

```
balbaya_landing_cwp_query <- paste(readLines(con = file.path(config_file[["queries_loc"]],
    "balbaya_landing_cwp.sql", fsep = "\\")),
    collapse = "\n")

balbaya_landing_cwp <- dbGetQuery(balbaya_con,
    balbaya_landing_cwp_query)

balbaya_landing_cwp <- cwp_to_center(data = balbaya_landing_cwp,
    cwp_name = "cwp", cwp_length = 5) %>%
    fao_area_overlay(overlay_level = "division",
        longitude_name = "longitude_dec",
        latitude_name = "latitude_dec")

balbaya_landing_cwp <- fao_area_overlay_unassociated(data = balbaya_landing_cwp[is.na(balbaya_landing_cw
    ], overlay_level = "division", longitude_name = "longitude_dec",
    latitude_name = "latitude_dec", tolerance = 250) %>%
    mutate(f_area = f_area_near, f_subarea = f_subarea_near,
        f_division = f_division_near) %>%
    select(-f_area_near, -f_subarea_near,
        -f_division_near) %>% rbind(balbaya_landing_cwp[!is.na(balbaya_landing_cwp$f_subarea),
    ])

balbaya_landing_cwp_fao_area <- unique(balbaya_landing_cwp[,
    c("f_area", "f_subarea", "f_division")]) %>%
    rowwise() %>% mutate(f_division_final = ifelse(f_subarea ==
    "34.2", "34.2.0", f_division), f_subarea_final = ifelse(f_area ==
    "34", f_division_final, f_subarea))

balbaya_landing_cwp <- balbaya_landing_cwp %>%
    inner_join(balbaya_landing_cwp_fao_area,
        by = c("f_area", "f_subarea", "f_division")) %>%
    mutate(sub_region = f_subarea_final) %>%
    group_by(country, year, quarter, sub_region,
        metier, fishing_mode, vessel_length,
        species) %>% summarise(totwghtlandg = sum(totwghtlandg)) %>%
    ungroup()

balbaya_landing_tablef <- balbaya_landing_cwp %>%
    full_join(observe_bycatch_retained_tabled,
        by = c("country", "year", "quarter",
            "sub_region", "metier", "fishing_mode",
            "vessel_length", "species")) %>%
    mutate(totwghtlandg = ifelse(is.na(totwghtlandg),
        0, totwghtlandg), retained_tons = ifelse(is.na(retained_tons),
        0, retained_tons), totwghtlandg = round(totwghtlandg +
        retained_tons, 3)) %>% rowwise() %>%
    mutate(domain_landings = paste(country,
        quarter, sub_region, paste0(unlist(strsplit(x = metier,
            split = "_"))[1], "-", fishing_mode,
            "_", paste(unlist(strsplit(x = metier,
                split = "_"))[-1], collapse = "_")),
        vessel_length, species, "NA", sep = "_"),
        no_samples = 1) %>% select(-quarter,
```

```
    -sub_region, -metier, -fishing_mode,
    -vessel_length, -retained_tons)

# CAS from sardara
sardara_cas_query <- paste(readLines(con = file.path(config_file[["queries_loc"]],
    "sardara_cas.sql", fsep = "\\")), collapse = "\n")
sardara_cas <- dbGetQuery(sardara_con, sardara_cas_query)

sardara_cas <- cwp_to_center(data = sardara_cas,
    cwp_name = "cwp", cwp_length = 5) %>%
    fao_area_overlay(overlay_level = "division",
        longitude_name = "longitude_dec",
        latitude_name = "latitude_dec")

sardara_cas <- fao_area_overlay_unassociated(data = sardara_cas[is.na(sardara_cas$f_subarea),
    ], overlay_level = "division", longitude_name = "longitude_dec",
    latitude_name = "latitude_dec", tolerance = 250) %>%
    mutate(f_area = f_area_near, f_subarea = f_subarea_near,
        f_division = f_division_near) %>%
    select(-f_area_near, -f_subarea_near,
        -f_division_near) %>% rbind(sardara_cas[!is.na(sardara_cas$f_subarea),
    ])

if (dim(as.data.frame(sardara_cas[is.na(sardara_cas$f_subarea),
    ]))[1] != 0) {
    stop("You have NA(s) value(s)", "\n",
        "Checking data")
}

sardara_cas_fao_area <- unique(sardara_cas[,
    c("f_area", "f_subarea", "f_division")]) %>%
    rowwise() %>% mutate(f_division_final = ifelse(f_subarea ==
    "34.2", "34.2.0", f_division), f_subarea_final = ifelse(f_area ==
    "34", f_division_final, f_subarea), eez_indicator = ifelse(f_area %in%
    c("41", "47", "51", "57"), "NA", ifelse(f_area ==
    "34", ifelse(f_subarea_final == "34.1.1",
    "COAST", ifelse(f_subarea_final %in%
        c("34.1.2", "34.1.3", "34.2.0"),
        "RFMO", "NA")), "eez_not_defined")))

if (dim(as.data.frame(sardara_cas_fao_area[sardara_cas_fao_area$eez_indicator ==
    "eez_not_defined", ]))[1] != 0) {
    stop("You have NA(s) value(s)", "\n",
        "Checking data")
}

sardara_cas <- sardara_cas %>% inner_join(sardara_cas_fao_area,
    by = c("f_area", "f_subarea", "f_division"))

sardara_cas$metier_1 <- str_extract(string = sardara_cas$metier,
    pattern = regex(pattern = "[^_]*"))
sardara_cas$metier_2 <- str_extract(string = sardara_cas$metier,
    pattern = regex(pattern = "(?<=_)\\w+"))
```

```
sardara_cas$metier_3 <- str_c(sardara_cas$metier_1,
    sardara_cas$fishing_mode, sep = "-")
sardara_cas$metier_4 <- str_c(sardara_cas$metier_3,
    sardara_cas$metier_2, sep = "_")
sardara_cas$domain_landings <- str_c(sardara_cas$country,
    sardara_cas$quarter, sardara_cas$f_subarea_final,
    sardara_cas$metier_4, sardara_cas$vessel_length,
    sardara_cas$species, "NA", sep = "_")

sardara_cas <- sardara_cas %>% group_by(country,
    year, domain_landings, species, length) %>%
    summarise(no_length = sum(no_length)) %>%
    ungroup() %>% group_by(country, year,
    domain_landings, species) %>% mutate(min_length = min(length),
    max_length = max(length), no_length_measurements = n())

# Merge for final table
tablef_final <- balbaya_landing_tablef %>%
    inner_join(sardara_cas, by = c("country",
        "year", "species", "domain_landings")) %>%
    mutate(length_unit = "cm") %>% select(country,
    year, domain_landings, species, totwghtlandg,
    no_samples, no_length_measurements, length_unit,
    min_length, max_length, length, no_length) %>%
    mutate(id_verif = paste0(country, year,
        domain_landings))

names(tablef_final) <- toupper(names(tablef_final))

# Consistency with table A
cons_tablea <- setdiff(unique(tablef_final[,
    c("COUNTRY", "YEAR", "DOMAIN_LANDINGS")]),
    unique(tablea_final[, c("COUNTRY", "YEAR",
        "DOMAIN_LANDINGS")])) %>% mutate(id_verif = paste0(COUNTRY,
    YEAR, DOMAIN_LANDINGS)) %>% select(id_verif)

cons_tablea = cons_tablea$id_verif

tablef_final <- tablef_final[!tablef_final$ID_VERIF %in%
    cons_tablea, ]
```

Point de vigilance : Ici nous avons été obligé de récupérer les données de débarquement depuis la base de données Balbaya en utilisant un autre niveau d'agrégation que celui utilisé pour la table A. En effet les "Catch-at-size" (CAS) de la base de données Sardara sont exprimé en carrée de 5 degrée. Pour pouvoir les regrouper avec les débarquements il est donc nécessaire de représenter les débarquements par carrée de 5 degrée (et non plus par latitude et longitude). Par ailleurs, lors de la jointure des deux tables finales on perd des données. Cela veut dont dire que certains CAS ne sont affectées à aucun débarquement et vis versa. L'an prochain il faudra regarder de plus près cela et si possible retracer le processus de calcul des CAS (actuellement calculé via le processus T3) avec les agrégations demandées par l'appel à données.

2.5 Table G: Effort summary

```
balbaya_effort_query <- paste(readLines(con = file.path(config_file[["queries_loc"]],
    "balbaya_effort.sql", fsep = "\\")),
    collapse = "\n")

balbaya_effort <- dbGetQuery(balbaya_con,
    balbaya_effort_query)

# Classification according FDI spatial
# area
balbaya_effort <- fao_area_overlay(data = balbaya_effort,
    overlay_level = "division", longitude_name = "longitude",
    latitude_name = "latitude")

balbaya_effort <- fao_area_overlay_unassociated(data = balbaya_effort[is.na(balbaya_effort$f_subarea),
    ], overlay_level = "division", longitude_name = "longitude",
    latitude_name = "latitude") %>% mutate(f_area = f_area_near,
    f_subarea = f_subarea_near, f_division = f_division_near) %>%
    select(-f_area_near, -f_subarea_near,
        -f_division_near) %>% rbind(balbaya_effort[!is.na(balbaya_effort$f_subarea),
    ]) %>% mutate(f_area = ifelse(f_area ==
    "47" & f_subarea %in% c("34.3", "34.4"),
    "34", f_area))

if (dim(as.data.frame(balbaya_effort[is.na(balbaya_effort$f_subarea),
    ]))[1] != 0) {
    stop("You have NA(s) value(s)", "\n",
        "Checking data")
}

balbaya_effort_fao_area <- unique(balbaya_effort[,
    c("f_area", "f_subarea", "f_division")]) %>%
    rowwise() %>% mutate(f_division_final = ifelse(f_subarea ==
    "34.2", "34.2.0", f_division), f_subarea_final = ifelse(f_area ==
    "34" | f_area == "27", f_division_final,
    f_subarea), eez_indicator = ifelse(f_area %in%
    c("41", "47", "51", "57"), "NA", ifelse(f_area ==
    "34", ifelse(f_subarea_final == "34.1.1",
    "COAST", ifelse(f_subarea_final %in%
        c("34.1.2", "34.1.3", "34.2.0"),
        "RFMO", "NA")), ifelse(f_area ==
    "27", ifelse(f_subarea_final %in% c("27.9.a",
    "27.8.a", "27.8.c"), "NA", "RFMO"), "eez_not_defined"))))

if (dim(as.data.frame(balbaya_effort_fao_area[balbaya_effort_fao_area$eez_indicator ==
    "eez_not_defined", ]))[1] != 0) {
    stop("You have NA(s) value(s)", "\n",
        "Checking data")
}

balbaya_effort <- balbaya_effort %>% inner_join(balbaya_effort_fao_area,
    by = c("f_area", "f_subarea", "f_division")) %>%
```

```
    mutate(sub_region = f_subarea_final)

balbaya_effort_rectangle <- balbaya_effort %>%
    select(-vessel_id, -totseadays, -totkwdaysatsea,
        -totgtdaysatsea, -totkwfishdays,
        -totgtfishdays, -hrsea, -kwhrsea,
        -gthrsea, -f_area, -f_subarea, -f_division,
        -f_division_final, -f_subarea_final)

balbaya_effort_nb_vessel <- balbaya_effort %>%
    group_by(country, year, quarter, vessel_length,
        fishing_tech, gear_type, target_assemblage,
        mesh_size_range, metier, supra_region,
        sub_region, eez_indicator, geo_indicator,
        specon_tech) %>% summarise(totves = n_distinct(vessel_id)) %>%
    ungroup()

balbaya_effort <- balbaya_effort %>% group_by(country,
    year, quarter, vessel_length, fishing_tech,
    gear_type, target_assemblage, mesh_size_range,
    metier, supra_region, sub_region, eez_indicator,
    geo_indicator, specon_tech, deep) %>%
    summarise(totseadays = sum(totseadays),
        totkwdaysatsea = sum(totkwdaysatsea),
        totgtdaysatsea = sum(totgtdaysatsea),
        totfishdays = sum(totfishdays), totkwfishdays = sum(totkwfishdays),
        totgtfishdays = sum(totgtfishdays),
        hrsea = sum(hrsea), kwhrsea = sum(kwhrsea),
        gthrsea = sum(gthrsea)) %>% ungroup() %>%
    mutate(totkwdaysatsea = ifelse(is.na(totkwdaysatsea),
        "NK", totkwdaysatsea), totkwfishdays = ifelse(is.na(totkwfishdays),
        "NK", totkwfishdays), kwhrsea = ifelse(is.na(kwhrsea),
        "NK", kwhrsea)) %>% inner_join(balbaya_effort_nb_vessel,
    by = c("country", "year", "quarter",
        "vessel_length", "fishing_tech",
        "gear_type", "target_assemblage",
        "mesh_size_range", "metier", "supra_region",
        "sub_region", "eez_indicator", "geo_indicator",
        "specon_tech")) %>% mutate(confidential = ifelse(fishing_tech ==
    "HOK", "Y", "N"))

names(balbaya_effort) <- toupper(names(balbaya_effort))
```

2.6 Table H: Landings by rectangle

```
observe_bycatch_retained_tableh <- lat_long_to_csquare(data = observe_bycatch_retained,
    grid_square = 0.5, latitude_name = "latitude_dec",
    longitude_name = "longitude_dec") %>%
    select(country, year, quarter, vessel_length,
        fishing_tech, gear_type, target_assemblage,
        mesh_size_range, metier, supra_region,
```

```
        geo_indicator, specon_tech, deep,
        species, retained_tons, eez_indicator,
        sub_region, grid_square_0.5) %>%
    rename(c_square = grid_square_0.5) %>%
    mutate(rectangle_type = "NA", rectangle_lat = "NA",
        rectangle_lon = "NA") %>% group_by(country,
    year, quarter, vessel_length, fishing_tech,
    gear_type, target_assemblage, mesh_size_range,
    metier, supra_region, geo_indicator,
    specon_tech, deep, species, sub_region,
    eez_indicator, c_square, rectangle_type,
    rectangle_lat, rectangle_lon) %>% summarise(retained_tons = sum(retained_tons)) %>%
    ungroup()

balbaya_landing_rectangle <- lat_long_to_csquare(data = balbaya_landing_rectangle,
    grid_square = 0.5, latitude_name = "latitude",
    longitude_name = "longitude") %>% mutate(rectangle_type = "NA",
    rectangle_lat = "NA", rectangle_lon = "NA") %>%
    rename(c_square = grid_square_0.5) %>%
    select(-latitude, -longitude) %>% group_by(country,
    year, quarter, vessel_length, fishing_tech,
    gear_type, target_assemblage, mesh_size_range,
    metier, supra_region, geo_indicator,
    specon_tech, deep, species, sub_region,
    eez_indicator, c_square, rectangle_type,
    rectangle_lat, rectangle_lon) %>% summarise(totwghtlandg = sum(totwghtlandg)) %>%
    ungroup()

tableh_final <- balbaya_landing_rectangle %>%
    full_join(observe_bycatch_retained_tableh,
        by = c("country", "year", "quarter",
            "vessel_length", "fishing_tech",
            "gear_type", "target_assemblage",
            "mesh_size_range", "metier",
            "supra_region", "geo_indicator",
            "specon_tech", "deep", "species",
            "sub_region", "eez_indicator",
            "c_square", "rectangle_type",
            "rectangle_lat", "rectangle_lon")) %>%
    mutate(totwghtlandg = ifelse(is.na(totwghtlandg),
        0, totwghtlandg), retained_tons = ifelse(is.na(retained_tons),
        0, retained_tons), totwghtlandg = round(totwghtlandg +
        retained_tons, 3), totvallandg = "NK",
        confidential = ifelse(fishing_tech ==
            "HOK", "Y", "N")) %>% select(country,
    year, quarter, vessel_length, fishing_tech,
    gear_type, target_assemblage, mesh_size_range,
    metier, supra_region, sub_region, eez_indicator,
    geo_indicator, specon_tech, deep, rectangle_type,
    rectangle_lat, rectangle_lon, c_square,
    species, totwghtlandg, totvallandg, confidential)

names(tableh_final) <- toupper(names(tableh_final))
```

2.7 Table I: Effort by rectangle

```
balbaya_effort_rectangle <- lat_long_to_csquare(data = balbaya_effort_rectangle,
    grid_square = 0.5, latitude_name = "latitude",
    longitude_name = "longitude") %>% mutate(rectangle_type = "NA",
    rectangle_lat = "NA", rectangle_lon = "NA") %>%
    rename(c_square = grid_square_0.5) %>%
    select(-latitude, -longitude) %>% group_by(country,
    year, quarter, vessel_length, fishing_tech,
    gear_type, target_assemblage, mesh_size_range,
    metier, supra_region, geo_indicator,
    specon_tech, deep, sub_region, eez_indicator,
    c_square, rectangle_type, rectangle_lat,
    rectangle_lon) %>% summarise(totfishdays = sum(totfishdays)) %>%
    ungroup() %>% select(country, year, quarter,
    vessel_length, fishing_tech, gear_type,
    target_assemblage, mesh_size_range, metier,
    supra_region, sub_region, eez_indicator,
    geo_indicator, specon_tech, deep, rectangle_type,
    rectangle_lat, rectangle_lon, c_square,
    totfishdays) %>% mutate(confidential = ifelse(fishing_tech ==
    "HOK", "Y", "N"))

names(balbaya_effort_rectangle) <- toupper(names(balbaya_effort_rectangle))
```

2.8 Table J: Capacity and fleet segment effort

```
balbaya_capacity_query <- paste(readLines(con = file.path(config_file[["queries_loc"]],
    "balbaya_capacity.sql", fsep = "\\")),
    collapse = "\n")
balbaya_capacity <- dbGetQuery(balbaya_con,
    balbaya_capacity_query)

balbaya_maxseadays_query <- paste(readLines(con = file.path(config_file[["queries_loc"]],
    "balbaya_maxseadays.sql", fsep = "\\")),
    collapse = "\n")
balbaya_maxseadays <- dbGetQuery(balbaya_con,
    balbaya_maxseadays_query)

balbaya_capacity_final <- balbaya_capacity %>%
    inner_join(balbaya_maxseadays, by = c("country",
        "year", "vessel_length", "fishing_tech",
        "supra_region", "geo_indicator")) %>%
    mutate(totkw = ifelse(is.na(totkw), "NK",
        totkw))

names(balbaya_capacity_final) <- toupper(names(balbaya_capacity_final))
```

3 Script de verification des données

```
# Total landings
check_landing_tablea <- tablea_final %>%
    group_by(COUNTRY, YEAR) %>% summarise(totwghtlandg_tablea = sum(TOTWGHTLANDG)) %>%
    ungroup()

check_landing_tablef <- unique(tablef_final[,
    c("COUNTRY", "YEAR", "DOMAIN_LANDINGS",
        "TOTWGHTLANDG")]) %>% group_by(COUNTRY,
    YEAR) %>% summarise(totwghtlandg_tablef = sum(TOTWGHTLANDG)) %>%
    ungroup()

check_landing_tableh <- tableh_final %>%
    group_by(COUNTRY, YEAR) %>% summarise(totwghtlandg_tableh = sum(TOTWGHTLANDG)) %>%
    ungroup()

check_landing <- check_landing_tablea %>%
    full_join(check_landing_tablef, by = c("COUNTRY",
        "YEAR")) %>% full_join(check_landing_tableh,
    by = c("COUNTRY", "YEAR"))

# Total discard
check_discard_tablea <- tablea_final %>%
    mutate(DISCARDS = ifelse(DISCARDS ==
        "NK", NA, DISCARDS), DISCARDS = as.numeric(DISCARDS)) %>%
    group_by(COUNTRY, YEAR) %>% summarise(discard_tablea = sum(DISCARDS,
    na.rm = TRUE)) %>% ungroup()

check_observe_bycatch <- observe_bycatch %>%
    group_by(country, year) %>% summarise(discard_observe_bycatch = sum(discards)) %>%
    ungroup()

check_discard_tabled <- unique(tabled_final[,
    c("COUNTRY", "YEAR", "DOMAIN_DISCARDS",
        "DISCARDS")]) %>% mutate(DISCARDS = as.numeric(DISCARDS)) %>%
    group_by(COUNTRY, YEAR) %>% summarise(discard_tabled = sum(DISCARDS)) %>%
    ungroup()

check_discards <- check_discard_tablea %>%
    full_join(check_observe_bycatch, by = c(COUNTRY = "country",
        YEAR = "year")) %>% full_join(check_discard_tabled,
    by = c("COUNTRY", "YEAR"))

# Effort
check_effort_tableg <- balbaya_effort %>%
    group_by(COUNTRY, YEAR, FISHING_TECH) %>%
    summarise(totfishdays = sum(TOTFISHDAYS)) %>%
    ungroup()

check_effort_tablei <- balbaya_effort_rectangle %>%
    group_by(COUNTRY, YEAR, FISHING_TECH) %>%
    summarise(totfishdays = sum(TOTFISHDAYS)) %>%
```

```
    ungroup()

check_effort <- check_effort_tableg %>% full_join(check_effort_tablei,
    by = c("COUNTRY", "YEAR", "FISHING_TECH"))
```

4 Explications du script d'exportation des tables

```
# Table A
write.csv2(x = as.data.frame(tablea_final),
    file = file.path(config_file[["output_loc"]],
        paste0("TABLE_A_IRD_", format(as.POSIXct(Sys.time()),
            "%Y%m%d_%H%M%S"), ".csv"), fsep = "\\"),
    row.names = FALSE)

# Table D
write.csv2(x = as.data.frame(tabled_final),
    file = file.path(config_file[["output_loc"]],
        paste0("TABLE_D_NAO_OFR_IRD_", format(as.POSIXct(Sys.time()),
            "%Y%m%d_%H%M%S"), ".csv"), fsep = "\\"),
    row.names = FALSE)

# Table F
write.csv2(x = tablef_final, file = file.path(config_file[["output_loc"]],
    paste0("TABLE_F_NAO_OFR_IRD_", format(as.POSIXct(Sys.time()),
        "%Y%m%d_%H%M%S"), ".csv"), fsep = "\\"),
    row.names = FALSE)

# Table G
write.csv2(x = as.data.frame(balbaya_effort),
    file = file.path(config_file[["output_loc"]],
        paste0("TABLE_G_IRD_", format(as.POSIXct(Sys.time()),
            "%Y%m%d_%H%M%S"), ".csv"), fsep = "\\"),
    row.names = FALSE)

# Table H
write.csv2(x = as.data.frame(tableh_final),
    file = file.path(config_file[["output_loc"]],
        paste0("TABLE_H_IRD_", format(as.POSIXct(Sys.time()),
            "%Y%m%d_%H%M%S"), ".csv"), fsep = "\\"),
    row.names = FALSE)

# Table I
write.csv2(x = as.data.frame(balbaya_effort_rectangle),
    file = file.path(config_file[["output_loc"]],
        paste0("TABLE_I_IRD_", format(as.POSIXct(Sys.time()),
            "%Y%m%d_%H%M%S"), ".csv"), fsep = "\\"),
    row.names = FALSE)

# Table J
write.csv2(x = as.data.frame(balbaya_capacity_final),
    file = file.path(config_file[["output_loc"]],
```

```
        paste0("TABLE_J_IRD_", format(as.POSIXct(Sys.time()),
            "%Y%m%d_%H%M%S"), ".csv"), fsep = "\\"),
    row.names = FALSE)
```

Point de vigilance : ici les tables sont exportées en csv plutot que en xlsx (demandé dans les spécifications de l'appel à données). Cela s'explique car la fonction d'exportation du package "xlsx" n'est pas du tout optimisée pour les gros jeux de données. Il est beaucoup plus rapide d'exporter directement les données en csv puis de convertir le fichier en xlsx via Excel.