Calculate VTR Area

Data Preparation

Data Collection

This block handles the data collection from the Oracle Database. It ensures that the correct R packages are installed/loaded before execution. Prerequisite to execution on Windows, the Open Database Connectivity (ODBC) drivers must be installed and a Data Source Name (DSN) must be set up in the ODBC Connection Manager. Access permissions must also be granted to view the VTR and PERMIT tables on the SOLE Database. A convention attempted in this script, variables written in all capital letters indicate variables pulled directly from the oracle database while variables written in lower camelCase are calculated within the script.

The data is first collected as two separate tables, one collecting trip information from the VTR table and the other collecting vessel information from the PERMIT table. The PERMIT data is then cleaned by removing permits that had issue dates occurring after the cancelation dates and removing permits that were not present in the VTR data.

Downloads runs ~5 minutes per year given.

```r
# Loads and/or installs the RODBC package to connect to
# oracle and data.table/rstudioapi packages
tryCatch(expr = require("RODBC"), warning = function(w) {
  install.packages("RODBC")
  require("RODBC")
})
tryCatch(expr = require("rstudioapi"), warning = function(w) {
  install.packages("rstudioapi")
  require("rstudioapi")
})
tryCatch(expr = require("data.table"), warning = function(w) {
  install.packages("data.table")
  require("data.table")
})

# Specifies username and target years for data to
# download
username <- "cquartararo"
years <- 2003:2006

# Connects to Oracle database where VTR data is stored.
# Asks for password when run
soleConnection <- odbcConnect(dsn = "SOLE", uid = username,
  pwd = rstudioapi::askForPassword("SOLE Database password"))

# Specifies the fields from the Oracle database to
# collect
soleQuery <- "SELECT t.permit permit, t.tripid, t.tripcatg, t.datesail, t.timesail,
  t.datelnd1 datelnd1, t.hullnum, t.port, t.crew,
  g.gearid, g.gearcode gear, g.mesh, g.gearqty,
  g.gearsize, g.nhaul, g.soakhrs, g.soakmin,
  g.depth, g.clatdeg, g.clatmin, g.clatsec,
 1"
g.clondeg, g.clonmin, g.clonsec, g.qdsq, 
g.cnemarea nemarea, g.tenmsq, 
s.sppcode, s.tripid, s.qtykept keptlb, s.qtydisc disclb, 
s.datesold, s.dealnum, 
a.nespp4, substr(nespp4,1,3) nespp3 
FROM VTR.VESLOG%sT t, VTR.VESLOG%sG g, VTR.VESLOG%sS s, VTR.VLSPPTBL a 
WHERE t.tripid=g.tripid and t.tripid=s.tripid and 
s.gearid=g.gearid and t.tripcatg in (1,4) 
and a.sppcode = s.sppcode"

# Loops through the years given above and collects all 
data into one table
soleData <- data.frame()  # Initializes the dataframe
for (year in years) {
  queryformat <- sprintf(soleQuery, year, year, year, year)  # Creates query for current year
  print(paste0("Querying ", year))
  soleData <- rbind(soleData, sqlQuery(soleConnection, 
    queryformat))  # Adds data to dataframe
  print(paste0("Finished ", year, "."))
}

# Specifies permit information from the Sole database to 
collect
vesselQuery <- "SELECT vp_num permit, hport, hpst, pport, ppst, ap_year, ves_name, 
gtons, vhp, len, date_issued issue, date_canceled cancel 
FROM permit.vps_vessel 
WHERE ap_year = %s"

# Connects to Oracle database where vessel permit data 
is stored. Asks for password when run. Server is the 
same as above, but connection may need to be refreshed 
after previous download
soleConnection <- odbcConnect(dsn = "SOLE", uid = username, 
pwd = rstudioapi::askForPassword("SOLE Database password"))

# Loops through years given above and collects a data 
frame for merging with SOLE data
vesselData <- data.frame() 
for (year in years) {
  queryformat <- sprintf(vesselQuery, year)  # Creates query for year
  print(paste0("Querying ", year))
  vesselData <- rbind(vesselData, sqlQuery(soleConnection, 
    queryformat))  # Adds data to dataframe
  print(paste0("Finished ", year, "."))
}

# PERMIT DATA CLEANING

# Filters out permits where the cancelation date is 
before the issue date
vesselData <- vesselData[which(vesselData$CANCEL > vesselData$ISSUE),
]
# Filters out duplicate rows
vesselData <- unique(vesselData)
# filters out permits that are not present in the
# soleData pulled for that year
vesselData <- vesselData[vesselData$PERMIT %in% unique(soleData$PERMIT), ]

Data Cleaning

The PERMIT and VTR tables are then merged together. Due to the erratic nature of the data and the
method of the function used, this merge produces a large dataset and could be an area where efficiency of
the process can be increased. Certain gear types are then dropped from the table since a Swept Area (SA)
calculation would be inappropriate. Duplicate Gear IDs are then sequenced and combined into one row to
create a dataset with one unique Gear ID per row. Those examining the table should note that the column
label for the respective character species codes, 3 digit codes, and quantities fished will all end in the same
number.

# Merges vesselData to soleData by permit number
data <- merge(soleData, vesselData, by = "PERMIT", all.x = TRUE, 
               all.y = FALSE)

# MAIN DATA CLEANNING

# Filters for data where the date of sail per trip is
# between or on the issue and cancel date of the permit
dataCleaned <- data[which((data$DATESAIL > data$ISSUE &
data$DATESAIL < data$CANCEL) | (data$DATESAIL == data$ISSUE |
data$DATESAIL == data$CANCEL)), ]

# Filters out gears not utilized in calculations
dataCleaned <- dataCleaned[-which(dataCleaned$GEAR %in%
                          c("HND", "DRO", "GNT", "OTM", "HRP", "DIV", "OTH", "GND",
                          "SES", "SED", "OTB", "LLP", "PUR", "PTM")), ]

# Removes the Issue and Cancel date from the dataset for
# cleanliness
dataCleaned[, c("ISSUE", "CANCEL")] <- list(NULL)

# Sets the cleaned dataset as a datatable
setDT(dataCleaned)

# Filters out duplicate rows
dataCleaned <- unique(dataCleaned)

# Compiles multiple entries of the same GEARID with
# duplicate SPP codes
BY <- dataCleaned[, names(dataCleaned)[-which(names(dataCleaned) %in%
                          c("KEPTLB", "DISCLB", "DEALNUM"))]]
dataCleaned <- dataCleaned[, list(KEPTLB = sum(KEPTLB),
                                 DISCLB = sum(DISCLB)), by = BY]

# Adds a column to number the same Gear IDs
dataCleaned <- transform(dataCleaned, sweepSeq = ave(GEARID,
                                          GEARID, FUN = seq_along))

# Collects gearID's of squid trips
squidGearIDs <- unique(dataCleaned$GEARID[which(dataCleaned$NESPP3 %in%
                                       c("801", "802", "803"))])

# Sets the cleaned dataset as a data table and combines
# rows
dataCleaned <- dcast(dataCleaned, PERMIT + TRIPID + TRIPCATG +
    DATESAIL + TIMESAIL + DATELND1 + HULLNUM + PORT + GEARID +
    GEAR + MESH + GEARQTY + GEARSIZE + SOAKHRS + SOAKMIN +
    DEPTH + CLATDEG + CLATMIN + CLATSEC + CLONDEG + CLONMIN +
    CLONSEC + QDSQ + NEMAREA + TENMSQ + VHP + GTONS + LEN +
    NHAUL ~ sweepSeq, value.var = c("SPPCODE", "NESPP3",
    "KEPTLB", "DISCLB"), drop = T)

data <- dataCleaned

**Addend Data**

Columns are then added to the cleaned table to illustrate decimal latitude/longitude, numerical
day/month/year for both sailing and landing, and the soak time of the gear in hours.

# Creates decimal lat and long
data$lat <- data$CLATDEG + (data$CLATMIN/60) + (data$CLATSEC/60 * 0.01)
data$lon <- -(data$CLONDEG + (data$CLONMIN/60) + (data$CLONSEC/60 * 0.04))

# Creates numeric columns for sailing day of month, month, year
data$day <- as.numeric(format(data$DATESAIL, "%d"))
data$month <- as.numeric(format(data$DATESAIL, "%m"))
data$year <- as.numeric(format(data$DATESAIL, "%Y"))

# Creates numeric columns for landing day of month, month, year
data$lday <- as.numeric(format(data$DATELND1, "%d"))
data$lmonth <- as.numeric(format(data$DATELND1, "%m"))
data$lyear <- as.numeric(format(data$DATELND1, "%Y"))

# Calculates the soak time by adding soak hours to soak minutes Sets all NAs to 0
# so that stupid R can do math properly
data$SOAKHRS[is.na(data$SOAKHRS)] <- 0
data$SOAKMIN[is.na(data$SOAKMIN)] <- 0
data$soak_time <- data$SOAKHRS + (data$SOAKMIN/60)

**Define Trip Types**

This block assigns trip types to each GearID to use in the calculations of Area. the function at the top of
the block is utilized when calculating the percentage of particular species to help assign trip types that may
utilize the same gear.

Assumptions that a trawl that catches greater than or equal to 75% squid is a squid trip. Assumptions that
a trawl after the year 2002 that catches more than 50% groundfish in areas 513/514 from July-November or
in areas 514/521 from September-December is a raised gear trip. Trips that last 1 day and keep less than
400lbs of target species using scallop dredge gear are labelled as Scallop General Catch (sca-gc).

# GENERAL MANAGEMENT

# Collects column names where SPP3 codes and KEPTLBs are # stored
neSpp3Cols <- names(data)[which(grepl(x = names(data), pattern = "NESPP3"))]  
keptLbCols <- names(data)[which(grepl(x = names(data), pattern = "KEPTLB"))]

# Calculates the total catch of a GearID row 
# data$total_kept <- apply(data[, ..keptLbCols], 1, sum, na.rm = TRUE)

calcP <- function(x, spps, neSpp3Cols, keptLbCols) {
  if (any(x[neSpp3Cols] %in% spps)) {
    return(sum(as.numeric(x[keptLbCols][which(x[neSpp3Cols] %in% spps)]), na.rm = TRUE)/as.numeric(x["total_kept"]))
  } else {
    return(0)
  }
}

# FIXED GEAR

# Gillnet 
# data$trip_type[data$GEAR %in% c("GNR", "GNS") ] <- "gillne"

# Longline 
# data$trip_type[data$GEAR %in% c("LLB") ] <- "longli"

# Traps and pots 
# data$trip_type[data$GEAR %in% c("TRP", "PTC", "PTE", "PTH", "PTL", "PTX", "PTW", "PTF", "PTS", "PTO") ] <- "trap"

# TRAWLS

# Otter Bottom Sets all trawling gears specified to 'otter'
# trip type. Following statements will overwrite to the 
# appropriate trip type 
# data$trip_type[data$GEAR %in% c("OTF", "OTS", "OTC", "OTO", "MIX", "OHS", "OTR", "PTB") ] <- "otter"

# Shrimp Bottom 
# data$trip_type[data$GEAR %in% c("OTS") & data$NESPP3 %in% 734:739 ] <- "shrimp"

# Squid Bottom Calculates the percentage of squid caught on a 
# GearID Row 
# data$squid_P <- apply(data, 1, calcP, c("800", "801", "802"), 
# neSpp3Cols, keptLbCols)

# Sets trip types for squid 
# data$trip_type[data$GEAR %in% c("OTF", "OTO") & data$squid_P >= 0.75 ] <- "squid"

# Raised Bottom Calculates the percentage of hake, whiting, 
# etc... caught on a GearID row 
# data$raised_P <- apply(data, 1, calcP, c("152", "508", "509"), 
# neSpp3Cols, keptLbCols)

# Sets trip types of OTF/MIX gears for years after 2002, 
# jul-nov in areas 514/513, sep-dec in areas 514/521 
# data$trip_type[data$GEAR %in% c("OTF", "MIX") & data$raised_P > 0.5 & data$year > 2002 & ((data$month %in% 7:11 & data$NEMAREA %in% c("514", "513")) | (data$month > 8 & data$NEMAREA %in% c("514", "521")))) ] <- "raised"
## Swept Area Calculations

All swept area calculations are done by calculating the distance towed and multiplying that by the Linear Effective Width of the gear type. These were created as functions that return a separate data frame to easily allow for replication and subset processing of a large cleaned data source. Calculations are vectorized as much as possible, avoiding loops to decrease the execution time.

### Trawls

This block calculates the Area swept (kM²) for all trawls.

A 40° angle of attack (α) is used as an intermediate value for trawl doors. Door weight is estimated based on a linear relationship based on vessel weight and horsepower. Door width is estimated based on a linear relationship based on door weight.

A 15° angle of attack (α) is used as an intermediate value for ground cables. Length of the cables was estimated using a linear relationship based on the vessel length.

The trawl sweep is estimated to be 43% of the nominal headrope width. Assumptions about the Soak Time of the dredges are made based on year and trip type.

Tow speed was estimated by calculating the mean speed per year from observer data. Due to similarities between averages, a constant speed of 3 knots was used across all years.

```r
calc_swept_area_OtterTrawl <- function(data) {
  # ADJUSTS SOAK TIMES Generic Otter Trawls
  data$soak_time[is.na(data$soak_time) & data$trip_type == "otter" & data$year < 2002] <- 2.6
  data$soak_time[is.na(data$soak_time) & data$trip_type == "otter" & data$year >= 2002] <- 2.1
  # Raised
  data$soak_time[is.na(data$soak_time) & data$trip_type == "raised"] <- 1.9
  # Shrimp
  data$soak_time[is.na(data$soak_time) & data$trip_type == "shrimp" & data$year < 2002] <- 2.48
  data$soak_time[is.na(data$soak_time) & data$trip_type == "shrimp" & data$year >= 2002] <- 2.05
  # Squid
  data$soak_time[is.na(data$soak_time) & data$trip_type == "squid"] <- 2.2
}
```
# DISTANCE
data$towspeed <- 3  # Towspeed (Nautical Miles/Hr)
data$hours_fished <- data$soak_time * data$NHAUL  # Total hours fished
data$d_nm <- data$towspeed * data$hours_fished  # Distance (Nautical Miles)
data$d_t <- data$d_nm * 1.852  # Distance towed (kilometers)

# BOARDS Weight of otter trawl doors estimated by regression
data$weight <- 70.848 + (1.844 * data$GTONS) + (0.5344 * data$VHP)
# Width trawl doors (meters) estimated by regression
data$width_o <- 0.001 * (1223 + (data$weight * 0.8333))
# Angle of attack (radians)
data$rad_o <- (2 * pi * 40)/360
# Effective Width of otter board (meters)
data$w_o <- data$width_o * sin(data$rad_o)
# Contact index for otter boards is constant
data$c_o <- 1

# CABLES (contact index set below) Width of ground cable (meters) based on regression
data$width_c <- 0.3048 * (137.54 + (data$LEN * 1.823))
# Adjust cable lengths
data$width_c[data$trip_type == "shrimp" & data$width_c > 27.432] <- 27.432
# Angle of attack (radians) of ground cable
data$rad_c <- (2 * pi * 15)/360
# Effective width of ground cable (meters)
data$w_c <- data$width_c * sin(data$rad_c)

# SWEEP (contact index set below)
data$raw_sweep <- data$GEARSIZE
data$w_s <- 0.43 * data$raw_sweep  # Effective width of sweep (meters)

# CONTACT INDECIES - CABLES Based on trip type
data$c_c[data$trip_type == "otter"] <- 0.95
data$c_c[data$trip_type == "raised"] <- 0.95
data$c_c[data$trip_type == "shrimp"] <- 0.9
data$c_c[data$trip_type == "squid"] <- 0.95

# CONTACT INDECIES - SWEEP Based on trip type
data$c_s[data$trip_type == "otter"] <- 0.9
data$c_s[data$trip_type == "raised"] <- 0.05
data$c_s[data$trip_type == "shrimp"] <- 0.95
data$c_s[data$trip_type == "squid"] <- 0.5

data$boards <- 2 * data$w_o * data$c_o  # Width of both boards (meters)
data$cables <- 2 * data$w_c * data$c_c  # Width of both cables (meters)
data$sweep <- data$w_s * data$c_s  # Width of the sweep (meters)
data$lin_eff_width <- data$boards + data$cables + data$sweep
data$A <- data$d_t * (0.001 * data$lin_eff_width)  # SA without sensitivity (km)

# Format and output data.frame with Gear ID, location,
Scallop Dredges

This block calculates the Area swept (kM²) for scallop dredges.

Soak time is assumed to be 1.14 hours if it is documented as greater than 3 hours or not documented. If the number of hauls is not documented or is greater than 200, the number of hauls is estimated based on the linear relationship to how many recorded days at sea per GearID. Gear size is assumed to be documented in feet if documented as less than 30 inches and converted before calculation. If none is documented, a gear size of 120 inches is assumed. If no gear quantity is documented, 1 dredge is assumed to be carried. If greater than 2 dredges are documented, the gear quantity is assumed to be 2.

Tow speed was estimated by averaging observer speed data per year. The same process was used to estimate tow duration in hours with data from NOVA’s OBDBS data, OBGEAR and OBHAU.

calc_swept_area_Dredge <- function(data) {
  # ADJUST PARAMETERS SPECIFIC TO SCALLOPS
  # Adjust soak times
  data$soak_time[data$soak_time == 0 | data$soak_time > 3] <- 1.14
  # Adds days at sea (da) and adjust number of hauls
  data$da <- data$DATELND1 - data$DATESAIL
  data$da[data == 0] <- 1
  data$da[data > 25] <- 25
  data$NHAUL[data$NHAUL == 0 | is.na(data$NHAUL) | data$NHAUL > 200] <- (-2.553 + da * 11.353)
  # Adjust gear size to inches
  data$GEARSIZE[data$GEARSIZE < 30] <- (data$GEARSIZE * 12)
  data$GEARSIZE[data$GEARSIZE == 0 | is.na(data$GEARSIZE)] <- 120
  # Adjust gear quantity
  data$GEARQTY[data$GEARQTY > 2] <- 2
  data$GEARQTY[data$GEARQTY == 0 | is.na(data$GEARQTY)] <- 1
  # DISTANCE Towspeed based on year (Nautical Miles/Hr)
  data$towspeed[data$year < 2004] <- 4.4
  data$towspeed[data$year == 2004] <- 4.5
  data$towspeed[data$year %in% 2005:2007] <- 4.6
  data$towspeed[data$year >= 2008] <- 4.7
  data$hours_fished <- data$soak_time * data$NHAUL  # Total hours fished
  data$d_nm <- data$towspeed * data$hours_fished  # Distance (Nautical Miles)
  data$d_t <- data$d_nm * 1.852  # Distance towed (kilometers)
  # GEAR WIDTH
  data$w_d <- 0.3048 * (data$GEARSIZE/12)  # Width of dredge converted from inches (meters)
  # CONTACT INEX - DREDGE
data$c_d <- 1

data$lin_eff_width <- data$GEARQTY * data$w_d * data$c_d
data$A = data$d_t * (0.001 * data$lin_eff_width)  # SASI without sensitivity (kilometers)

result_columns <- c("year", "NEMAREA", "GEAR", "GEARID", "trip_type", "lat", "lon", "A", "hours_fished")
result_data <- data[, ..result_columns]
return(result_data)

Hydraulic Clam Dredge

These SA values follow the same process as above, however averages for gear size and other parameters differ.
Towspeed is assumed to be 2 knots for all years based on observer data stored in NOVA’s OBDBS data, OBGEAR and OBHAU.
calc_swept_area_Dredge <- function(data) {

## Specific Clam Dredge information #### needs to be
## looked at

gear <- "DRC"
trip_type <- "hy_drg"

## Distance
data$towspeed <- 2  # Towspeed based on year (Nautical Miles/Hr)
data$hours_fished <- data$soak_time * data$NHAUL  # Total hours fished
data$d_nm <- data$towspeed * data$hours_fished  # Distance (Nautical Miles)
data$d_t <- data$d_nm * 1.852  # Distance towed (kilometers)

## Gear size estimated in meters from VTR Data
data$GEARSIZE[data$year %in% 2000:2003] <- 1.9
data$GEARSIZE[data$year == 2004] <- 2
$data$GEARSIZE[data$year %in% 2005:2006] <- 2.7
data$GEARSIZE[data$year == 2007] <- 2.84
data$GEARSIZE[data$year >= 2008] <- 2.85
data$w_d <- data$GEARSIZE  # Width of dredge converted from inches (meters)

## Contact index for dredge
data$c_d <- 1

data$lin_eff_width <- data$GEARQTY * data$w_d * data$c_d
data$A = data$d_t * (0.001 * data$lin_eff_width)  # SASI without sensitivity (kilometers)

result_columns <- c("year", "NEMAREA", "GEAR", "GEARID", "trip_type", "lat", "lon", "A", "hours_fished")
result_data <- data[, ..result_columns]
return(result_data)
}
Fixed Gears

Contains code that cleans the given data set by replacing empty values (.) of nhauls, gearqty, and gearsize with the means of each respectively. Empty soak_times will be replaced with a value according to type and year. gear_length is then added based on gear type.

Metrics for distance (d), hours fished (hours_fished), and area swept (A) are then calculated based on the clean data given.

calc_FixedGear <- function(data) {

    # Calculates the means of values and replaces '.' values
    # Number of hauls
    nhaulAve <- mean(data$NHAUL[data$NHAUL != 0], na.rm = T)
    data$NHAUL[is.na(data$NHAUL) | data$NHAUL == 0] <- nhaulAve
    # Gear quantity
    gearqtyAve <- mean(data$GEARQTY[data$GEARQTY != 0], na.rm = T)
    data$GEARQTY[is.na(data$GEARQTY) | data$GEARQTY == 0] <- gearqtyAve
    # Gear size
    gearsizeAve <- mean(data$GEARSIZE[data$GEARSIZE != 0], na.rm = T)
    data$GEARSIZE[is.na(data$GEARSIZE) | data$GEARSIZE == 0] <- gearsizeAve

    # Replace missing soak time values based on gear type
    # and year SOAKTIME UNUSED IN THIS GEARTYPE Trap Soak
    # times
    data$soak_time[is.na(data$soak_time) & data$trip_type == "trap"] <- 80
    # Longline soak times
    data$soak_time[is.na(data$soak_time) & data$trip_type == "longli" & data$year %in% c(1996:2003, 2010)] <- 3
    data$soak_time[is.na(data$soak_time) & data$trip_type == "longli" & data$year %in% c(2004)] <- 3.5
    data$soak_time[is.na(data$soak_time) & data$trip_type == "longli" & data$year %in% c(2005:2007)] <- 4.35
    data$soak_time[is.na(data$soak_time) & data$trip_type == "longli" & data$year %in% c(2008)] <- 4
    data$soak_time[is.na(data$soak_time) & data$trip_type == "longli" & data$year %in% c(2009)] <- 2.8
    data$soak_time[is.na(data$soak_time) & data$trip_type == "longli" & data$year %in% c(2011)] <- 3.9
    # Gillnet soak times
    data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(1996:2001)] <- 38
    data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2002)] <- 44
    data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2003:2004)] <- 46
    data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2005)] <- 56
    data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2006)] <- 50
}
data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2007)] <- 53
data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2008:2009)] <- 43
data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2010)] <- 40
data$soak_time[is.na(data$soak_time) & data$trip_type == "gillne" & data$year %in% c(2011)] <- 56

# Hours fished HOURS FISHED UNUSED IN THESE CALCULATIONS
data$hours_fished <- as.numeric(data$soak_time) * as.numeric(data$NHAUL)

# Sets gear length based on trip type (Implies gear) in # kM
setGearLength <- function(row) {
  # Trap length based on 1mi average string length
  if (row["trip_type"] == "trap") {
    return(1.609344)
  }
  # "gillne" gear length calculation
  if (row["trip_type"] == "gillne") {
    return(as.numeric(row["GEARQTY"])*as.numeric(row["GEARSIZE"])*0.0003048) # ft to kM
  }
  # Longline gear length calculation
  if (row["trip_type"] == "longli") {
    return(as.numeric(row["GEARSIZE"])*1.852) # 1/10 Nautical Mile to kM
  }
}
data$gear_length <- apply(data, 1, FUN = setGearLength) # Applies length function to data by row

# Contact index for fixed gear
data$c_d <- 1

# Distance
data$d <- as.numeric(data$NHAUL) * as.numeric(data$gear_length)

# Linear effective width of gear in kilometers
lin_eff_width <- 0.001

# Calculate SASI without sensitivity
data$A <- as.numeric(data$d) * lin_eff_width

# Returns final data
result_columns <- c("year", "NEMAREA", "GEAR", "GEARID", "trip_type", "lat", "lon", "A", "hours_fished")
result_data <- data[, ..result_columns]
return(result_data)

Executing Swept Area Calculations

This block will execute the functions described above. They do not subset by year, but do subset the cleaned data into data tables based on trip type that can be joined post hoc.
# Calculate SA for Trawls
trawlSA <- calc_swept_area_OtterTrawl(data[which(data$trip_type %in%
c("otter", "raised", "shrimp", "squid")), ])

# Calculate SA for Dredges
dredgeSA <- calc_swept_area_Dredge(data[which(data$trip_type %in%
c("sca-la", "sca-gc")), ])

# Calculate SA for Hydraulic Dredges
hyDredgSA <- calc_swept_area_Dredge(data[which(data$trip_type %in%
c("hy_drg")), ])

# Calculate SA for Fixed Gears
fixedSA <- calc_FixedGear(data[which(data$trip_type %in%
c("gillne", "longli", "trap")), ])