CCFRP Derived Data Table Code for Length and Effort Data

Rachel Brooks, Shelby Zeigler, Ryan Fields

05/09/2022

The following code is used to extract and compile the CCFRP effort table, which provides an estimate of fish abundance by calculating species-specific CPUE (catch per angler hour) and BPUE (biomass per angler hour). Additionally, this code is used to extract and compile the CCFRP species length table.

Open packages needed for analysis

Load 'Raw' formats of the tables. These are .csv versions of the Access database tables. These are exactly as they are exported from Access - i.e. Column names are not changed until it gets to R. The following .csv data tables used for analysis can be accessed through the CCFRP GIT HUB:

CCFRP GIT HUB LINK HERE

Use the raw drift data to merge fields with the 'Caught Fishes' table. This allows us to see all fish ever caught, even those excluded from CPUE calculations.

```
#1-Trip Information table
trip.data.raw = read.csv("CSV_Files/1-Trip Information.csv")

#3-Drift Information Table
#Included bit to make any blanks change to 'NA'
drift.data.raw = read.csv("CSV_Files/3-Drift Information.csv", na = c("", "NA"))

#4-Caught Fishes Table
fishes.caught.raw = read.csv("CSV_Files/4-Caught Fishes.csv")


#Fish codes to match species code with common name
fish.codes = read.csv("CSV_Files/Fish Species.csv")

#Length weight relationships
length.weight = read.csv("CSV_Files/CCFRP_Biomass_Conversion_20220206.csv")

#OPC MPA Designation Info
CCFRP_location_table <- read.csv("CSV_Files/CCFRP_location_table.csv")
```

Now filter data to be used

Trip Data:

1) Select columns of interest

2) Rename columns as needed

3) Filtering out 'Areas' for this analysis. Current version has all 'Areas' listed to be filtered. User can select specific 'Areas' of interest here (i.e., Select AN, PL, PB, BL for ONLY central CA data)

4) Using mutate() to convert 'Area' to a factor with the full names as labels

```r
trip.data = trip.data.raw %>%
  select(c(Trip.ID, Area, Month, Day, Year.Automatic)) %>%
  dplyr::rename(Year = Year.Automatic) %>%
  mutate(Area = factor(Area, levels = c('TD','CM','TM','SP',
                                        'FN','BH','AN','PL',
                                        'BL','PB','PC','CP',
                                        'AI','LB','SW','LJ'),
                       labels = c('Trinidad','Cape Mendocino','Ten Mile',
                                  'Stewarts Point','Farallon Islands',
                                  'Bodega Head','Ano Nuevo', 'Point Lobos',
                                  'Piedras Blancas','Point Buchon',
                                  'Point Conception','Carrington Point',
                                  'Anacapa Island','Laguna Beach',
                                  'Swamis','South La Jolla'))) %>%

  droplevels()

#Define cells to be excluded from analysis
excluded.cells = c("TDRR", "CMMM", "CMRR", "TMMM", "TMRR",
                   "FNMM","FNRR","SPMM", "SPRR", "BHMM", "BHRR",
                   "ANMM", "ANRR","PLMM","PLMN", "PLMO","PLRR",
                   "BLMM", "BLRR","PBMM", "PBRR",
                   "PCMM", "PCRR", "CPMM", "CPRR", "AIMM", "AIRR",
                   "LBMM", "LBRR", "SWMM", "SMRR", "LJMM", "LJRR")
```

Drift Data:

This should include drifts with no fish caught. I've included the column 'Total.Fishes.Caught' to visually verify that 0-fish drifts are included. This will be important for the fishing effort calculation.

1) Get rid of drifts we have marked for exclusion (e.g. drifted outside cell, overlapped with other cell, etc.)

2) Select columns of interest

3) Rename columns as needed

4) Merge with trip data: This is an inner-join by default and will only select drifts that are contained in the filtered trip data above (i.e. no need to select particular 'Areas' again)

5) Filter out the excluded cells

6) Filter any drift that was 2 min or less in length (See CCFRP SOP document for this rule)

```r
drift.data = drift.data.raw %>%

  filter(is.na(Excluded.Drift.Comment)) %>%
  #Drop excluded drifts - no text allowed in this cell

  select(c(Drift.ID, Trip.ID, Grid.Cell.ID, ID.Cell.per.Trip,
         Site..MPA..REF., Drift.Time..hrs., Total.Fishes.Caught,
         Total.Angler.Hrs )) %>%   #Select columns that are relevant

  dplyr::rename(Site = Site..MPA..REF.,
```

```
              Drift.Time.Hrs = Drift.Time..hrs.) %>%

   merge(., trip.data, by = intersect("Trip.ID", "Trip.ID")) %>%
   #Merge with trip.data to get Trip-level info attached to each drift

   filter(!Grid.Cell.ID %in% excluded.cells) %>%
   #don't sample from excluded cells

    filter(Drift.Time.Hrs > (2/60)) %>%
   #drift must be 2 min long at least to keep

   droplevels()
```

Effort Calculation:

This calculates 'Total Angler Hours Fished' per each 'ID.Cell.per.Trip' field.

'ID.Cell.per.Trip' is our Sample unit and is the amount of fish caught in a cell on a given day. This includes drifts that were greater than 2min and did not have fish caught.

We are not filtering total time just yet ('ID.Cell.per.Trip' time >2hrs). We will do that after merge with data below.

```
effort.total = drift.data %>%
  group_by(Area, Site, Year, ID.Cell.per.Trip,Grid.Cell.ID)%>%
  #!!Keep Grid.Cell.ID to merge drifts with 0 fish caught.
  #Drop later before calculating BPUE
  summarise(Total.Angler.Hours = sum(Total.Angler.Hrs))
```

```
## `summarise()` has grouped output by 'Area', 'Site', 'Year', 'ID.Cell.per.Trip'.
## You can override using the `.groups` argument.
```

```
#ddply needs name to be changed (I'm using 'Hours' instead of 'Hrs')
```

Fish Data:

'fishes.caught' will be the full data set of fish caught regardless of whether they were caught within CCFRP cells. This data set will be used for gear type comparisons, but not for CPUE.

1) Merge with raw drift data (so that no fishes are excluded)

2) Merge with 'fish.codes' to get Common Names

3) Select columns of interest

4) Rename columns for later use

```
fishes.caught.all = fishes.caught.raw %>%
 merge(., drift.data.raw, by = intersect('Drift.ID','Drift.ID'), all = F) %>%
  # 'all = F' excludes drift information for 225 drifts that did not catch fish
 merge(.,fish.codes, by = intersect('Species.Code','Species.Code'), all = F) %>%
  #merge common name; Drop unused Species Codes

  select(c(ID.Cell.per.Trip,Drift.ID, Grid.Cell.ID, Excluded.Drift.Comment,
```

```
          Drift.Time..hrs.,Gear.Type, Station..,Species.Code, Common.Name,
          Length..cm.)) %>%   #Choose columns of interest
  dplyr::rename(Exclude = Excluded.Drift.Comment,
                Drift.time.Hrs = Drift.Time..hrs.,
                Station = Station..,
                Length.cm = Length..cm.)
```

CPUE calculations:

We need to apply all the same filters as we did with 'drift.data'

1) Exclude fishes caught on excluded drifts

2) Exclude fishes caught in excluded cells

3) Exclude fishes caught on drifts less than 2 min long

4) Select columns of interest for later analysis

```
fishes.caught.cpue = fishes.caught.all %>%
  filter(is.na(Exclude)) %>% #drop excluded drifts
  filter( !Grid.Cell.ID %in% excluded.cells) %>%
  # Get rid of excluded cell codes - often for sampling outside of the cells
  filter(Drift.time.Hrs > (2/60)) %>%
  #Also get rid of fish from drifts less than 2 min
  select(c(ID.Cell.per.Trip, Drift.ID, Grid.Cell.ID, Species.Code,
          Common.Name, Length.cm)) %>%
  droplevels()
```

Calculate CPUE and Total Counts:

Next we need to aggregate the data like we would in a pivot table in Excel

1) Use reshape package. Much like a pivot table. This is using the 'length' function to count fishes.

2) Define first species column

3) Next we merge the 'pivot-table' of species counts by 'IDCell.per.Trip' with the effort data so that we can divide catch by total adjusted angler
   hours. Here set 'all.x = T' to keep Cells that we sampled but did not catch any fishes. These represent entire cells that were fished for enough time to be included, but caught no fish 3a) Use mutate() to create a Total CPUE variable (Sum of all species) 3b) Filter out ID.Cell.per.Trip's that were less than 2 Total angler hours (See CCFRP SOP document)

4) Replace NA's with Zeros:these are the cells that did not catch fishes NOTE: Ryan F confirmed that these are valid data points through 2016

5) Divide the counts by 'Total.Angler.Hours' to get fish/angler hour or CPUE

```
recast.counts = dcast(fishes.caught.cpue, ID.Cell.per.Trip + Grid.Cell.ID ~
                      Common.Name, length, value.var = "Species.Code")

first.spp = colnames(recast.counts)[3]

recast.counts$Grid.Cell.ID <- NULL
```

```r
Counts.per.IDcell = merge(x=effort.total, y=recast.counts, by =
          intersect("ID.Cell.per.Trip", "ID.Cell.per.Trip"), all.x = TRUE) %>%
  mutate(Total = rowSums(.[which(colnames(.) == first.spp):length(.)])) %>%
  filter(Total.Angler.Hours > 2) #Need over 2 angler hours per Cell to retain

#Define the columns with species (e.g. columns Barred Sand Bass:Total)
spp.cols = colnames(Counts.per.IDcell)[which(colnames(Counts.per.IDcell) ==
                                  first.spp):length(Counts.per.IDcell)]

#Now use these columns to select relevant data and replace NA with 0
Counts.per.IDcell[spp.cols] <- replace(Counts.per.IDcell[spp.cols],
                          is.na(Counts.per.IDcell[spp.cols]), 0)

#Table Reformatting for Species Count Data
Counts.per.IDcell$Total <- NULL
Count_long_format <- gather(Counts.per.IDcell, Common_Name, Count,
                "Barred Sand Bass":"Yellowtail Rockfish", factor_key=TRUE)
Count_long_format$Month <- substr(Count_long_format$ID.Cell.per.Trip,4,5)
Count_long_format$Day <- substr(Count_long_format$ID.Cell.per.Trip,6,7)
Count_long_format$Date <- paste(Count_long_format$Year,Count_long_format$
                          Month,Count_long_format$Day,sep="-")
Count_long_format <- merge(x = Count_long_format, y =
                    CCFRP_location_table[,c("CA_MPA_name_short",
                    "Grid_Cell_ID", "LTM_project_short_code",
                    "Monitoring_Group")], by.x = "Grid.Cell.ID",
                    by.y="Grid_Cell_ID",all.x = TRUE )
Count_long_format_reorder <- Count_long_format[,c(13,14,12,3,4,11,
                                        5,9,10,2,1,7,8,6)]
names(Count_long_format_reorder)[names(Count_long_format_reorder)==
                          "Grid.Cell.ID"]<-"Grid_Cell_ID"
names(Count_long_format_reorder)[names(Count_long_format_reorder)==
                          "ID.Cell.per.Trip"]<-"ID_Cell_per_Trip"

#Define the columns with species (e.g. columns Barred Sand Bass:Total)
spp.cols = colnames(Counts.per.IDcell)[which(colnames(Counts.per.IDcell) ==
                                  first.spp):length(Counts.per.IDcell)]

#This step divides the counts by effort to get CPUE
CPUE.per.IDcell = Counts.per.IDcell
CPUE.per.IDcell[spp.cols] = (Counts.per.IDcell[, spp.cols]/
                          Counts.per.IDcell[, "Total.Angler.Hours"])

#Table Reformatting for Species CPUE Data
CPUE_long_format <- gather(CPUE.per.IDcell, Common_Name, CPUE,
                "Barred Sand Bass":"Yellowtail Rockfish", factor_key=TRUE)
CPUE_long_format$Month <- substr(CPUE_long_format$ID.Cell.per.Trip,4,5)
CPUE_long_format$Day <- substr(CPUE_long_format$ID.Cell.per.Trip,6,7)
CPUE_long_format$Date <- paste(CPUE_long_format$Year,CPUE_long_format$Month,
                          CPUE_long_format$Day,sep="-")
CPUE_long_format <- merge(x = CPUE_long_format, y =
                    CCFRP_location_table[,c("CA_MPA_name_short",
                    "Grid_Cell_ID", "LTM_project_short_code",
                    "Monitoring_Group")], by.x = "Grid.Cell.ID",
```

```
                          by.y="Grid_Cell_ID",all.x = TRUE )
CPUE_long_format_reorder <- CPUE_long_format[,c(13,14,12,3,4,11,5,
                                     9,10,2,1,6,7,8)]
names(CPUE_long_format_reorder)[names(CPUE_long_format_reorder)==
                     "Grid.Cell.ID"]<-"Grid_Cell_ID"
names(CPUE_long_format_reorder)[names(CPUE_long_format_reorder)==
                     "ID.Cell.per.Trip"]<-"ID_Cell_per_Trip"
names(CPUE_long_format_reorder)[names(CPUE_long_format_reorder)==
                     "CPUE"]<-"CPUE_catch_per_angler_hour"
CPUE_long_format_reorder$Total.Angler.Hours <- NULL
```

Length Data

1) Merge the site/area/year information with the length information,

2) Filter out any NA value in the 'Length.cm' column since we only want fishes with size data. NA's also cause problems later when calculating average size.

```
#!!Use a holder of fishes.caught.cpue and drop the Grid.Cell.ID column from the
#fishes.caught.cpue.holder to ensure that Grid.Cell.ID values come from effort
#total table. Not dropped from fishes.caught.cpue as Grid.Cell.ID column is used
#later in a different merge function
fishes.caught.cpue.holder<- fishes.caught.cpue #!!
fishes.caught.cpue.holder$Grid.Cell.ID <- NULL #!!

length.data = merge(x = effort.total, y = fishes.caught.cpue.holder,
  by = intersect("ID.Cell.per.Trip", "ID.Cell.per.Trip"),all.x=TRUE) %>%
  filter(!is.na(Length.cm)) %>%  #get rid of fish without lengths
  droplevels()

levels(length.data$Area)
```

```
##  [1] "Trinidad"         "Cape Mendocino"  "Ten Mile"         "Stewarts Point"
##  [5] "Farallon Islands" "Bodega Head"     "Ano Nuevo"        "Point Lobos"
##  [9] "Piedras Blancas"  "Point Buchon"    "Point Conception" "Carrington Point"
## [13] "Anacapa Island"   "Laguna Beach"    "Swamis"           "South La Jolla"
```

```
length.data$Month <- substr(length.data$ID.Cell.per.Trip,4,5)
length.data$Day <- substr(length.data$ID.Cell.per.Trip,6,7)
length.data$Date <- paste(length.data$Year,length.data$
                          Month,length.data$Day,sep="-")
length.data <- merge(x = length.data, y =
               CCFRP_location_table[,c("CA_MPA_name_short","Grid_Cell_ID",
               "LTM_project_short_code", "Monitoring_Group")],
               by.x = "Grid.Cell.ID", by.y="Grid_Cell_ID",all.x = TRUE )
length.data_reorder <- length.data[,c(15,16,14,3,4,13,5,11,12,2,1,9,10,8,6)]

#Rename Column Headers
names(length.data_reorder)[names(length.data_reorder)=="Grid.Cell.ID"]<-
                                     "Grid_Cell_ID"
names(length.data_reorder)[names(length.data_reorder)=="ID.Cell.per.Trip"]<-
                                     "ID_Cell_per_Trip"
```

```r
names(length.data_reorder)[names(length.data_reorder)=="Common.Name"]<-
                                                "Common_Name"
names(length.data_reorder)[names(length.data_reorder)=="Length.cm"]<-
                                                "Length_cm"
names(length.data_reorder)[names(length.data_reorder)=="monitoring_group"]<-
                                                "Monitoring_Group"
length.data_reorder$Total.Angler.Hours <- NULL
length.data_reorder$Species.Code <- NULL

### Table export for species length data
write.csv(length.data_reorder, file = "2007-2021_CCFRP_derived_length_table.csv",
                                        row.names = FALSE)
```

BPUE Calculations

1) Start with the subset of fishes 'fishes.caught.cpue'

2) Caculate biomass in Kg by using length weight relationships: compilation of data from VRG and PISCO tables -> source listed in table (Updated by Rachel & Shelby 02/2022)

3) This will exclude species without length/weight conversions

Note: As of the 2017 season, there were 164 fishes without lengths that will not be counted which may lead to slight underestimates of biomass.

```r
fishes.caught.bpue = fishes.caught.cpue %>%
  na.omit() %>%
  #Get rid of fishes without lengths - this will change results
  #slightly, but can't be avoided
  merge(., length.weight, by = 'Common.Name') %>%
  mutate(
    biomass.kg =ifelse(WL_L_units=='cm' & WL_input_length=='TL'
                  & WL_W_units=='kg', WL_a*((Length.cm)^WL_b),
              ifelse(WL_L_units=='mm' & WL_input_length=='TL'
                  & WL_W_units=='kg', WL_a*((Length.cm*10)^WL_b),
              ifelse(WL_L_units=='cm' & WL_input_length=='TL'
                  & WL_W_units=='g', WL_a*((Length.cm)^WL_b)/1000,
              ifelse(WL_L_units=='mm' & WL_input_length=='TL'
                  & WL_W_units=='g', WL_a*((Length.cm*10)^WL_b)/1000,
              ifelse(WL_L_units=='mm' & WL_input_length=='SL'
                  & WL_W_units=='g' & LC_type_for_WL=='TYPICAL',
                  WL_a*((LC_a*(Length.cm*10)+LC_b)^WL_b)/1000,
              ifelse(WL_L_units=='mm' & WL_input_length=='SL'
                  & WL_W_units=='g' & LC_type_for_WL=='REVERSE',
                  WL_a*((((Length.cm*10)-LC_b)/LC_a)^WL_b)/1000,NA)))))))%>%
  droplevels()


recast.biomass = dcast(fishes.caught.bpue, ID.Cell.per.Trip + Grid.Cell.ID ~
                    Common.Name, sum, value.var = "biomass.kg")

############################################################################
```

```r
first.spp.biomass = colnames(recast.biomass)[3]

#!!Use a holder of effort.total and drop the Grid.Cell.ID column from the
#effort.total.holder to ensure that Grid.Cell.ID values come from recast biomass
#table. Not dropped from original effort.total as Grid.Cell.ID column
effort.total.holder<- effort.total #!!
effort.total.holder$Grid.Cell.ID <- NULL

Biomass.per.IDcell = merge(effort.total.holder, recast.biomass, by =
                            intersect("ID.Cell.per.Trip", "ID.Cell.per.Trip"),
                            all.x = T) %>%
  mutate(Total = rowSums(.[which(colnames(.) ==
                            first.spp.biomass):length(.)])) %>%
  filter( Total.Angler.Hours > 2) #Need over 2 angler hours per Cell to retain

#Define the columns with species (e.g. columns Barred Sand Bass:Total)
spp.cols.biomass =
  colnames(Biomass.per.IDcell)[which(colnames(Biomass.per.IDcell) ==
                            first.spp.biomass):length(Biomass.per.IDcell)]

#Now use these columns to select relevant data and replace NA with 0
Biomass.per.IDcell[spp.cols.biomass]<-
      replace(Biomass.per.IDcell[spp.cols.biomass],
      is.na(Biomass.per.IDcell[spp.cols.biomass]), NA)


#This step divides biomass by effort to get BPUE
BPUE.per.IDcell = Biomass.per.IDcell
BPUE.per.IDcell[spp.cols.biomass] = (Biomass.per.IDcell[, spp.cols.biomass]
                            /Biomass.per.IDcell[, "Total.Angler.Hours"])

#Table Reformatting for Species BPUE Data
BPUE.per.IDcell$Total <- NULL
BPUE_long_format <- gather(BPUE.per.IDcell, Common_Name, BPUE,
                  "Barred Sand Bass":"Yellowtail Rockfish", factor_key=TRUE)
BPUE_long_format$Month <- substr(BPUE_long_format$ID.Cell.per.Trip,4,5)
BPUE_long_format$Day <- substr(BPUE_long_format$ID.Cell.per.Trip,6,7)
BPUE_long_format$Date <- paste(BPUE_long_format$Year,BPUE_long_format$
                            Month,BPUE_long_format$Day,sep="-")
BPUE_long_format <- merge(x = BPUE_long_format,
                      y = CCFRP_location_table[,c("CA_MPA_name_short",
                            "Grid_Cell_ID", "LTM_project_short_code",
                            "Monitoring_Group")], by.x = "Grid.Cell.ID",
                             by.y="Grid_Cell_ID",all.x = TRUE )
BPUE_long_format_reorder <- BPUE_long_format[,c(13,14,12,3,4,11,5,
                                    9,10,2,1,7,8,6)]
names(BPUE_long_format_reorder)[names(BPUE_long_format_reorder)==
                            "Grid.Cell.ID"]<-"Grid_Cell_ID"
names(BPUE_long_format_reorder)[names(BPUE_long_format_reorder)==
                            "ID.Cell.per.Trip"]<-"ID_Cell_per_Trip"
names(BPUE_long_format_reorder)[names(BPUE_long_format_reorder)==
                            "BPUE"]<-"BPUE_biomass(kg)_per_angler_hour"
BPUE_long_format_reorder$Total.Angler.Hours <- NULL
```

Now we will merge Counts, CPUE and BPUE together to form the 'Effort Table".

We will create a column that combines both 'ID_cell_per_Trip' and 'Common Name' for merging

```r
#Rename data frames for merging purposes and create column for merging

#BPUE
CPUEmerge <- CPUE_long_format_reorder%>%
  group_by(LTM_project_short_code, Monitoring_Group,
           CA_MPA_name_short, Area, Site, Date, Year, Month, Day,
           ID_Cell_per_Trip, Grid_Cell_ID,  Common_Name)
CPUEmerge$combo<- paste(CPUEmerge$ID_Cell_per_Trip,CPUEmerge$
                                        Common_Name, sep=".")

#CPUE
BPUEmerge <- BPUE_long_format_reorder
BPUEmerge$combo<- paste(BPUEmerge$ID_Cell_per_Trip,
                        BPUEmerge$Common_Name, sep=".")

#Drop all variables except combo and CPUE columns
BPUEsub<-BPUEmerge[,c(13,14)]

#Species counts
Countmerge <- Count_long_format_reorder
Countmerge$combo<- paste(Countmerge$ID_Cell_per_Trip,
                         Countmerge$Common_Name, sep=".")
Countsub<-Countmerge[,c(13:15)]

#Merge BPUE and CPUE dataframes first
BCCPUE<- merge(CPUEmerge, BPUEsub, by=c("combo"))
AllFishVars<-merge(BCCPUE, Countsub, by=c("combo"))

#Drop column with combo used for merge
AllFishVars<-AllFishVars[,-1]

#If CPUE values are zero then BPUE values should be zero not NA
AllFishVars$`BPUE_biomass(kg)_per_angler_hour`<-
  ifelse(AllFishVars$CPUE_catch_per_angler_hour==0,0,
         AllFishVars$`BPUE_biomass(kg)_per_angler_hour`)

names(AllFishVars)[names(AllFishVars)=="Total.Angler.Hours"]<-
                                        "Total_Angler_Hours"
names(AllFishVars)[names(AllFishVars)=="Site"]<-"MPA_Status"


#Export out CSV file with fish data prior to adding in environmental data
write.csv(AllFishVars, "CCFRP_AllFishVariables.csv")
```

Now we need to gather the data for the environmental variables for each sampling unit or cell 'ID_cell_per_Trip'.

We will pull in the data from the CCFRP database for the following environmental data:

1. Surface water temperature (recorded in degrees Celsius)

2. Water temperature at depth (recorded in degrees Celsius),

3. Vessel sea water temperature (recorded in degrees Fahrenheit, code will convert to Celsius),

4. Bottom Relief (recorded in 0-3 categories),

5. Start depth (depth at start of drift; recorded in feet, code will convert to meters),

6. End depth (depth at end of drift; ; recorded in feet, code will convert to meters),

7. Wind speed in knots,

8. Swell height (recorded in feet, code will convert to meters)

```
drift.data.env<-drift.data.raw %>%
   select(c('ID.Cell.per.Trip', 'Surface.T..instrument..C.',
            'Depth.T...instrument..C.', 'SWT..vessel..F.', 'Relief..1.3.',
            'Start.Depth..ft.', 'End.Depth..ft.', 'Obs.Wind.Speed..kt.',
            'Obs.Swell.Height..ft.'))%>%
    dplyr::rename(ID_Cell_per_Trip = 'ID.Cell.per.Trip',
                  Surface_Water_Temp1 = 'Surface.T..instrument..C.',
                  Depth_Water_Temp1 =  'Depth.T...instrument..C.',
                  Vessel_Water_Temp1 = 'SWT..vessel..F.',
                  Relief1 = 'Relief..1.3.',
                  Start_Depth1 = 'Start.Depth..ft.',
                  End_Depth1 = 'End.Depth..ft.',
                  Wind_Speed1 = 'Obs.Wind.Speed..kt.',
                  Swell_Height1 = 'Obs.Swell.Height..ft.')%>%
  group_by(ID_Cell_per_Trip)%>%
  summarize(Surface_Water_Temp = mean(Surface_Water_Temp1),
            Depth_Water_Temp =  mean(Depth_Water_Temp1),
            Vessel_Water_Temp = mean(Vessel_Water_Temp1),
            Relief = mean(Relief1),
            Start_Depth = mean(Start_Depth1),
            End_Depth = mean(End_Depth1),
            Wind_Speed = mean(Wind_Speed1),
            Swell_Height = mean(Swell_Height1))%>%
  droplevels()

#Convert vessel water temperature from Fahrenheit to Celsius
drift.data.env$Vessel_Water_Temp<-((drift.data.env$Vessel_Water_Temp-32)/1.8)

#Convert depths and swell heights from feet to meters?
drift.data.env$Start_Depth<-(drift.data.env$Start_Depth/3.281)
drift.data.env$End_Depth<-(drift.data.env$End_Depth/3.281)
drift.data.env$Swell_Height<-(drift.data.env$Swell_Height/3.281)

#Match environmental data with the BPUE, CPUE and Count data.

#Create all CCFRP data frame to match the environmental data for each Cell ID
#since these are different length we have to match each variable independently
AllCCFRP<-AllFishVars

#Match surface water temp to ID cell per trip
AllCCFRP$Surface_Water_Temp<-drift.data.env$Surface_Water_Temp[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]
```

```r
#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="Surface_Water_Temp"]<-"Surface_Water_Temp_C"

#Match depth water temp  to ID cell per trip
AllCCFRP$Depth_Water_Temp<-drift.data.env$Depth_Water_Temp[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]
#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="Depth_Water_Temp"]<-"Depth_Water_Temp_C"

#Mach Vessel water temp to ID cell per trip
AllCCFRP$Vessel_Water_Temp<-drift.data.env$Vessel_Water_Temp[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]
#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="Vessel_Water_Temp"]<-"Vessel_Water_Temp_C"

#Match Relief to ID cell per trip
AllCCFRP$Relief<-drift.data.env$Relief[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]

#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="Relief"]<-"Relief_(1-3)"

#Match start depth to ID cell per trip
AllCCFRP$Start_Depth<-drift.data.env$Start_Depth[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]
#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="Start_Depth"]<-"Start_Depth_m"

#Match End depth to ID cell per trip
AllCCFRP$End_Depth<-drift.data.env$End_Depth[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]
#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="End_Depth"]<-"End_Depth_m"

#Match Wind speed to ID cell per trip
AllCCFRP$Wind_Speed<-drift.data.env$Wind_Speed[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]
#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="Wind_Speed"]<-"Wind_Speed_kt"

#Match Swell height to ID cell per trip
AllCCFRP$Swell_Height<-drift.data.env$Swell_Height[match(AllCCFRP$
                ID_Cell_per_Trip, drift.data.env$ID_Cell_per_Trip)]
#Add units to column header
names(AllCCFRP)[names(AllCCFRP)=="Swell_Height"]<-"Swell_Height_m"
```

Reorder the columns of the data set and export final csv

```r
#reorder column to have all trip and environmental data before fish data.
AllCCFRP <- AllCCFRP[,c(1:11,17:24,16,12:15)]

#Export final dataframe out
write.csv(AllCCFRP, "2007-2021_CCFRP_derived_effort_table.csv",row.names=FALSE)
```