Check the 2020 data and incorporate it into gfiphc

Andrew Edwards

In 2020 only the first 20 hooks were evaluated, so those data are not easily imported into GFbio. Going to incorporate into gfiphc here. Likely need this as a template for future years (resave this file with new year, and change all 2020's to the subsequent year – and go through the code somewhat manually to check the output as you go along). This code includes some manual checks to make sure the data look okay.

For comparison first look at 2013 data included in gfiphc:

```
load_all()
> Loading gfiphc
> Warning: replacing previous import 'vctrs::data_frame' by 'tibble::data_frame'
> when loading 'dplyr'
setData2013
> # A tibble: 170 x 8
>     year station   lat   lon avgDepth effSkateIPHC E_it20 usable
>    <int> <chr>   <dbl> <dbl>    <int>        <dbl>  <dbl> <chr>
>  1  2013 2001     48.3 -126.       76         5.96   1.19 Y
>  2  2013 2002     48.3 -126.       93         5.90   1.19 Y
>  3  2013 2003     48.5 -125.       79         5.90   1.19 Y
>  4  2013 2004     48.5 -126.       56         5.96   1.20 Y
>  5  2013 2005     48.5 -126.       58         6.02   1.20 Y
>  6  2013 2006     48.5 -126.      110         5.78   1.16 Y
>  7  2013 2007     48.7 -125.       35         5.96   1.20 Y
>  8  2013 2008     48.7 -125.       35         5.90   1.20 Y
>  9  2013 2009     48.7 -126.       67         5.90   1.19 Y
> 10  2013 2010     48.7 -126.       41         5.96   1.20 Y
> # ... with 160 more rows
countData2013
> # A tibble: 1,304 x 4
>     year station spNameIPHC           specCount
>    <int> <chr>   <chr>                    <int>
>  1  2013 2001    Spiny Dogfish               61
>  2  2013 2001    Empty Hook                  57
>  3  2013 2001    Pacific Halibut              2
>  4  2013 2002    Spiny Dogfish               59
>  5  2013 2002    Empty Hook                  56
```

```
>  6  2013 2002    Pacific Halibut              5
>  7  2013 2003    Sablefish (Blackcod)         1
>  8  2013 2003    Longnose Skate               4
>  9  2013 2003    Arrowtooth Flounder          7
> 10  2013 2003    Spiny Dogfish               13
> # ... with 1,294 more rows
```

We want to get the new data into the same format as those (columns with same names and classes, even though in retrospect some classes aren't ideally chosen, but also retaining retrieved and observed hooks for the set data). Two data sets are needed because later gfiphc code summarises catches of a particular species at the station level, and needs to create counts of zeros for the species of interest (and such zeros are not included in IPHC output).

Set-level information

Maria was sent the file `2020 IPHCtoDFO_dataExtraction-Maria.xls` for set details, but this is multiple sheets and more complex than needed. Will try extracting directly from IPHC website (which they want us to do in the future anyway), using the following instructions:

Go to https://www.iphc.int/data/fiss-data-query and select the following options:

1. Year Range – 2020 to 2020.
2. Area 2B
3. Purpose Codes – All
4. IPHC Charter Regions – All
5. Maps – Nothing
6. Select non-Pacific halibut species – deselect All.

Download tab on bottom right (see instructions above question 4), and select CrossTab. Select "Set and Pacific Halibut data" and .xlsx format (I tried .csv format but it didn't save with commas, strangely). Save in this folder as `set-and-halibut-data-2020.xlsx`. Open in Excel and Export as .csv, `set-and-halibut-data-2020.csv`, and when trying to quit Excel say no to save changes (not sure if that matters).

Repeat but with all non-halibut data (select All in number 6), and save as `non-halibut-data-2020.xlsx` and export as .csv in Excel, `non-halibut-data-2020.csv`. Importantly, this file (but not the first one) contains the numbers of observed hooks, needed in our calculations.

```
sets_raw <- readr::read_csv("set-and-halibut-data-2020.csv") %>%
  dplyr::mutate_if(is.character, factor)
> Parsed with column specification:
> cols(
>   .default = col_double(),
```

```
>   `Vessel code` = col_character(),
>   `IPHC Reg Area` = col_character(),
>   `IPHC Charter Region` = col_character(),
>   Purpose = col_character(),
>   Date = col_character(),
>   Eff = col_character(),
>   Ineffcde = col_logical(),
>   `032 Pacific halibut weight` = col_number(),
>   `U32 Pacific halibut weight` = col_number()
> )
> See spec(...) for full column specifications.
sets_raw
> # A tibble: 198 x 33
>    `Row number`  Year Stlkey `Vessel code` Station Setno `IPHC Reg Area`
>           <dbl> <dbl>  <dbl> <fct>           <dbl> <dbl> <fct>
>  1            1  2020 2.02e7 BDP              2059     1 2B
>  2            2  2020 2.02e7 BDP              2052     2 2B
>  3            3  2020 2.02e7 BDP              2051     3 2B
>  4            4  2020 2.02e7 BDP              2079     4 2B
>  5            5  2020 2.02e7 BDP              2078     5 2B
>  6            6  2020 2.02e7 BDP              2073     6 2B
>  7            7  2020 2.02e7 BDP              2268     7 2B
>  8            8  2020 2.02e7 BDP              2267     8 2B
>  9            9  2020 2.02e7 BDP              2270     9 2B
> 10           10  2020 2.02e7 BDP              2272    10 2B
> # ... with 188 more rows, and 26 more variables: `IPHC Stat Area` <dbl>, `IPHC
> #   Charter Region` <fct>, Purpose <fct>, Date <fct>, Eff <fct>,
> #   Ineffcde <lgl>, BeginLat <dbl>, BeginLon <dbl>, `BeginDepth (fm)` <dbl>,
> #   EndLat <dbl>, EndLon <dbl>, `EndDepth (fm)` <dbl>, `MidLat fished` <dbl>,
> #   `MidLon fished` <dbl>, `AvgDepth (fm)` <dbl>, `Lat - Grid target` <dbl>,
> #   `Lon - Grid target` <dbl>, `032 Pacific halibut count` <dbl>, `U32 Pacific
> #   halibut count` <dbl>, `032 Pacific halibut weight` <dbl>, `U32 Pacific
> #   halibut weight` <dbl>, `No. skates set` <dbl>, `No. skates hauled` <dbl>,
> #   `Avg no. hook/skate` <dbl>, `Effective skates hauled` <dbl>, `Soak time
> #   (min.)` <dbl>
summary(sets_raw)
>    Row number          Year          Stlkey         Vessel code    Station
>  Min.   :  1.00   Min.   :2020   Min.   :20200071   BDP:139     Min.   :2044
>  1st Qu.: 50.25   1st Qu.:2020   1st Qu.:20200262   HAN:  1     1st Qu.:2094
>  Median : 99.50   Median :2020   Median :20200324   VNI: 58     Median :2144
>  Mean   : 99.50   Mean   :2020   Mean   :20200496               Mean   :2176
>  3rd Qu.:148.75   3rd Qu.:2020   3rd Qu.:20200859               3rd Qu.:2282
>  Max.   :198.00   Max.   :2020   Max.   :20200954               Max.   :2343
>
```

```
>      Setno         IPHC Reg Area IPHC Stat Area  IPHC Charter Region
>  Min.   :  1.00   2B:198        Min.   : 90.0   Charlotte:83
>  1st Qu.: 25.00                 1st Qu.:102.0   Goose Is.:56
>  Median : 49.50                 Median :112.0   St. James:59
>  Mean   : 57.80                 Mean   :112.1
>  3rd Qu.: 89.75                 3rd Qu.:121.0
>  Max.   :139.00                 Max.   :135.0
>
>                Purpose           Date      Eff      Ineffcde          BeginLat
>  Deep expansion   :  3   06-Sep-20:  6   Y:198   Mode:logical   Min.   :50.82
>  Shallow expansion: 30   07-Sep-20:  6           NA's:198       1st Qu.:51.83
>  Standard grid    :165   14-Jul-20:  6                          Median :52.66
>                          01-Jul-20:  5                          Mean   :52.70
>                          02-Jul-20:  5                          3rd Qu.:53.52
>                          02-Sep-20:  5                          Max.   :55.00
>                          (Other)  :165
>     BeginLon      BeginDepth (fm)      EndLat          EndLon
>  Min.   :-133.7   Min.   :  9.00   Min.   :50.80   Min.   :-133.7
>  1st Qu.:-131.1   1st Qu.: 39.25   1st Qu.:51.83   1st Qu.:-131.1
>  Median :-130.3   Median : 72.00   Median :52.66   Median :-130.3
>  Mean   :-130.3   Mean   : 86.25   Mean   :52.70   Mean   :-130.3
>  3rd Qu.:-129.3   3rd Qu.:122.00   3rd Qu.:53.52   3rd Qu.:-129.3
>  Max.   :-126.9   Max.   :281.00   Max.   :55.00   Max.   :-126.8
>
>  EndDepth (fm)   MidLat fished   MidLon fished    AvgDepth (fm)
>  Min.   : 10.0   Min.   :50.82   Min.   :-133.7   Min.   :  9.00
>  1st Qu.: 42.0   1st Qu.:51.83   1st Qu.:-131.1   1st Qu.: 45.25
>  Median : 81.5   Median :52.66   Median :-130.3   Median : 79.00
>  Mean   : 91.9   Mean   :52.70   Mean   :-130.3   Mean   : 87.90
>  3rd Qu.:124.5   3rd Qu.:53.51   3rd Qu.:-129.3   3rd Qu.:118.75
>  Max.   :390.0   Max.   :55.00   Max.   :-126.8   Max.   :287.00
>
>  Lat - Grid target Lon - Grid target O32 Pacific halibut count
>  Min.   :50.83     Min.   :-133.7    Min.   :  0.00
>  1st Qu.:51.83     1st Qu.:-131.1    1st Qu.:  8.00
>  Median :52.67     Median :-130.3    Median : 19.00
>  Mean   :52.70     Mean   :-130.3    Mean   : 28.71
>  3rd Qu.:53.50     3rd Qu.:-129.3    3rd Qu.: 39.75
>  Max.   :55.00     Max.   :-126.8    Max.   :173.00
>
>  U32 Pacific halibut count O32 Pacific halibut weight
>  Min.   :  0.00            Min.   :   0.0
>  1st Qu.:  1.25            1st Qu.: 224.8
>  Median :  9.50            Median : 491.0
```

```
>  Mean   : 18.96            Mean   : 716.4
>  3rd Qu.: 23.00            3rd Qu.: 864.2
>  Max.   :211.00            Max.   :5401.0
>
>  U32 Pacific halibut weight No. skates set No. skates hauled Avg no. hook/skate
>  Min.   :   0.0             Min.   :8      Min.   :7.00      Min.   : 97.00
>  1st Qu.:  11.0             1st Qu.:8      1st Qu.:8.00      1st Qu.: 99.00
>  Median :  78.5             Median :8      Median :8.00      Median : 99.00
>  Mean   : 150.6             Mean   :8      Mean   :7.99      Mean   : 98.97
>  3rd Qu.: 191.5             3rd Qu.:8      3rd Qu.:8.00      3rd Qu.: 99.00
>  Max.   :1628.0             Max.   :8      Max.   :8.00      Max.   :100.00
>
>  Effective skates hauled Soak time (min.)
>  Min.   :6.890           Min.   :394.0
>  1st Qu.:7.950           1st Qu.:442.2
>  Median :7.950           Median :528.5
>  Mean   :7.936           Mean   :540.7
>  3rd Qu.:7.950           3rd Qu.:609.0
>  Max.   :8.030           Max.   :952.0
>
testthat::expect_equal(unique(sets_raw$"IPHC Reg Area"), as.factor("2B")) # Check just B
testthat::expect_equal(unique(sets_raw$Year), 2020)
#testthat::expect_equal(length(unique(sets_raw$Station)),
#                       length(sets_raw$Station))
```

Understand any issues raised above

Uncomment those three `testthat` commands when looking at new data each year. If any of fail then have to comment it out and figure out what it means here.

This is for 2020 (check for future years), to look for station(s) that was fished twice:

```
length(unique(sets_raw$Station))
> [1] 197
length(sets_raw$Station)
> [1] 198
dplyr::count(sets_raw, Station) %>% dplyr::filter(n > 1)
> # A tibble: 1 x 2
>   Station     n
>     <dbl> <int>
> 1    2104     2
twice_fished <- dplyr::count(sets_raw, Station) %>%
  dplyr::filter(n > 1) %>%
  select(Station) %>%
```

```
  as.numeric()
# If there's more than a single station then adapt later code
as.data.frame(dplyr::filter(sets_raw,
                            Station == twice_fished))
>   Row number Year   Stlkey Vessel code Station Setno IPHC Reg Area
> 1         42 2020 20200254         VNI    2104    25            2B
> 2        108 2020 20200335         HAN    2104     4            2B
>   IPHC Stat Area IPHC Charter Region       Purpose      Date Eff Ineffcde
> 1            102           St. James Standard grid 12-Jul-20   Y       NA
> 2            102           St. James Standard grid 18-Jul-20   Y       NA
>   BeginLat  BeginLon BeginDepth (fm)  EndLat    EndLon EndDepth (fm)
> 1  52.1655 -130.0752             107 52.1665 -130.0110            87
> 2  52.1668 -130.0803             107 52.1663 -130.0142            87
>   MidLat fished MidLon fished AvgDepth (fm) Lat - Grid target Lon - Grid target
> 1       52.1660     -130.0432            93            52.167           -130.05
> 2       52.1667     -130.0472            94            52.167           -130.05
>   O32 Pacific halibut count U32 Pacific halibut count
> 1                        77                        60
> 2                        42                        49
>   O32 Pacific halibut weight U32 Pacific halibut weight No. skates set
> 1                       1321                        520              8
> 2                        723                        394              8
>   No. skates hauled Avg no. hook/skate Effective skates hauled Soak time (min.)
> 1                 8                  99                    7.95              553
> 2                 8                  98                    7.87              445
```

So Station 2104 had two vessels fishing the same station (which the code below originally caused a total of four rows for that station, explaining the 200 rows I had in original `setData2020` before fixing the issue). Interestingly the halibut catches were almost double for one vessel than the other (but were 6 days apart):

Note that one of those entries has 'Vessel code' HAN, but HAN only appears once in the whole data set (as seen in `summary(sets_raw)` above:

```
summary(sets_raw$"Vessel code")
> BDP HAN VNI
> 139   1  58
```

So given we want to exclude one of the duplicates, makes sense to exclude HAN. (Also, Dana mentioned some gear comparison studies for 2020). Do this and simplify down to what's needed and rename, based on `iphc2013data.Rnw` (need to include the 'purpose' column, unlike 2013):

```
sets_simp <- dplyr::filter(sets_raw, `Vessel code` != "HAN") %>%
  dplyr::select(year = Year,
                station = Station,
```

```
                lat = "MidLat fished",
                lon = "MidLon fished",
                avgDepth = "AvgDepth (fm)",
                skatesHauled = "No. skates hauled",
                effSkateIPHC = "Effective skates hauled",
                soakTimeMinutes = "Soak time (min.)",  # Joe might want
                usable = Eff,
                purpose = Purpose,
                U32halibut = "U32 Pacific halibut count",
                O32halibut = "O32 Pacific halibut count") %>%
  arrange(station) %>%
  dplyr::mutate(year = as.integer(year),
                station = as.character(station),
                avgDepth = as.integer(avgDepth),
                usable = as.character(usable))
sets_simp
> # A tibble: 197 x 12
>     year station   lat   lon avgDepth skatesHauled effSkateIPHC soakTimeMinutes
>    <int> <chr>   <dbl> <dbl>    <int>        <dbl>        <dbl>            <dbl>
>  1  2020 2044     50.8 -129.       50            8         7.95              679
>  2  2020 2045     50.8 -129.       68            8         7.95              571
>  3  2020 2046     51.0 -128.       53            8         7.95              633
>  4  2020 2047     51.0 -129.       39            8         7.95              541
>  5  2020 2048     51.0 -129.       47            8         7.95              768
>  6  2020 2049     51.0 -129.       86            8         7.95              654
>  7  2020 2050     51   -130.      131            8         7.95              551
>  8  2020 2051     51.2 -128.       58            8         7.95              642
>  9  2020 2052     51.2 -128.      106            8         7.95              546
> 10  2020 2053     51.2 -129.       56            8         7.95              446
> # ... with 187 more rows, and 4 more variables: usable <chr>, purpose <fct>,
> #   U32halibut <dbl>, O32halibut <dbl>
```

For future years check the HAN issue and remove that first line if necessary (especially if HAN is used in the survey).

Then change purpose to `standard` (`Y/N`) to match 2018 data (`Y` for the standard grid). Here `purpose` takes three values, and we need to convert to `standard`:

```
summary(sets_simp$purpose)
>    Deep expansion Shallow expansion     Standard grid
>                 3                30               164

sets_simp_std <- dplyr::mutate(sets_simp,
                               standard_tmp = (purpose == "Standard grid"))
```

```
standard <- as.character(sets_simp_std$standard_tmp)  # to get the right length
standard[sets_simp_std$standard_tmp] = "Y"
standard[!sets_simp_std$standard_tmp] = "N"
length(standard)
> [1] 197

sets_simp_std <- cbind(sets_simp_std,
                       standard) %>%
  as_tibble() %>%
  dplyr::select(-c("standard_tmp"))
summary(sets_simp_std)
>       year        station               lat             lon
>  Min.   :2020   Length:197         Min.   :50.82   Min.   :-133.7
>  1st Qu.:2020   Class :character   1st Qu.:51.83   1st Qu.:-131.1
>  Median :2020   Mode  :character   Median :52.66   Median :-130.3
>  Mean   :2020                      Mean   :52.70   Mean   :-130.3
>  3rd Qu.:2020                      3rd Qu.:53.51   3rd Qu.:-129.3
>  Max.   :2020                      Max.   :55.00   Max.   :-126.8
>     avgDepth       skatesHauled   effSkateIPHC   soakTimeMinutes
>  Min.   :  9.00   Min.   :7.00   Min.   :6.890   Min.   :394.0
>  1st Qu.: 45.00   1st Qu.:8.00   1st Qu.:7.950   1st Qu.:442.0
>  Median : 79.00   Median :8.00   Median :7.950   Median :529.0
>  Mean   : 87.87   Mean   :7.99   Mean   :7.936   Mean   :541.2
>  3rd Qu.:119.00   3rd Qu.:8.00   3rd Qu.:7.950   3rd Qu.:609.0
>  Max.   :287.00   Max.   :8.00   Max.   :8.030   Max.   :952.0
>     usable                        purpose      U32halibut       O32halibut
>  Length:197         Deep expansion   :  3   Min.   :  0.00   Min.   :  0.00
>  Class :character   Shallow expansion: 30   1st Qu.:  1.00   1st Qu.:  8.00
>  Mode  :character   Standard grid    :164   Median :  9.00   Median : 19.00
>                                             Mean   : 18.81   Mean   : 28.64
>                                             3rd Qu.: 23.00   3rd Qu.: 39.00
>                                             Max.   :211.00   Max.   :173.00
>    standard
>  Length:197
>  Class :character
>  Mode  :character
>
>
>
```

Look at data and show map to understand changing definition of standard station from 2018 to 2020.

It seems that the definition of 'standard grid' has changed from 2018 (when first needed due to the expanded grid) to 2020. Simply equating them as above is not sufficient.

This section figures out the two problems and corrects them. However, to replicate the original analysis we need to revert the second correction (the correction changes station 2343 to be non-standard, even though it originally was classed as standard – see later):

```
hooks_with_bait_revert <- hooks_with_bait

# This should be commented out for 2021 survey analysis in iphc-2021-data.Rmd,
#  since the problem is fixed:
hooks_with_bait_revert$set_counts[hooks_with_bait_revert$set_counts$year == 2018 &
                            hooks_with_bait_revert$set_counts$station == 2343,
                            ]$standard = "Y"

filter(hooks_with_bait$set_counts, year == 2018, station == 2343) %>%
  as.data.frame()      # saved version
>   year station    lat       lon  E_it N_it     C_it E_it20 N_it20   C_it20
> 1 2018    2343 53.984 -131.4333 7.027  110 15.65391 1.4054     16 11.38466
>   usable standard
> 1      Y        N
filter(hooks_with_bait_revert$set_counts, year == 2018, station == 2343) %>%
  as.data.frame()      # reverted version
>   year station    lat       lon  E_it N_it     C_it E_it20 N_it20   C_it20
> 1 2018    2343 53.984 -131.4333 7.027  110 15.65391 1.4054     16 11.38466
>   usable standard
> 1      Y        Y
```

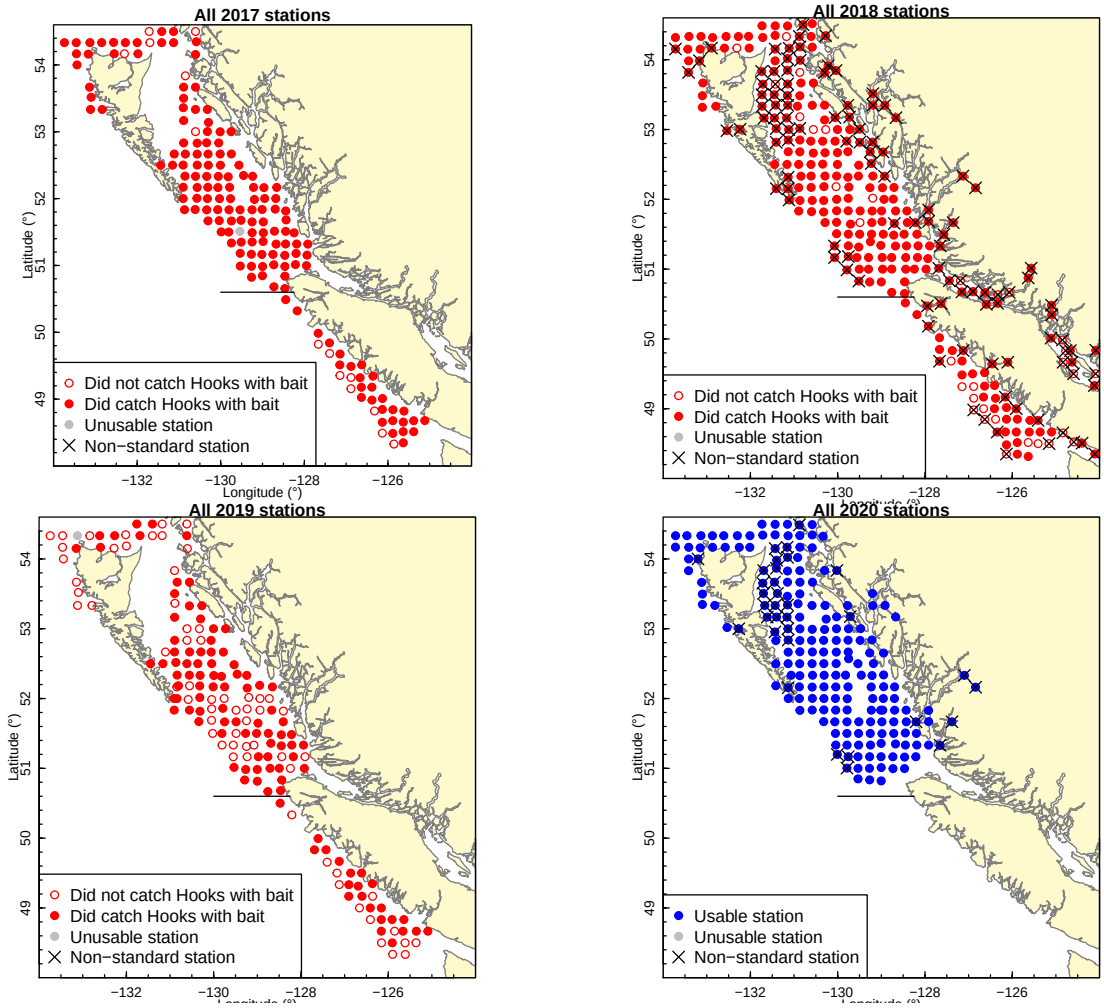
Plotting four years, with crosses showing 'non-standard'. (2020 is coloured different since no hooks with bait data yet, but the important bit is the crosses).

```
sets_2020 <- dplyr::select(sets_simp_std,
                           -c(U32halibut, O32halibut))
                           # not the same structure as sets_2018, below

plot_iphc_map_panel(hooks_with_bait_revert$set_counts,
                sp = "Hooks with bait",
                years = 2017:2019,
                indicate_standard = TRUE)

plot_iphc_map(sets_2020,
              sp = NULL,
              years = 2020,
```

```
                indicate_standard = TRUE)
```



Clearly 2020 has a few less stations just north of Vancouver Island, but not enough to worry about greatly. The curious ones are the ones way in in the inlets that are no longer flagged as non-standard.

Need to look and plot values:

```
sets_2018 <- filter(hooks_with_bait_revert$set_counts,
                    year == 2018)
not_std_2018 <- filter(sets_2018,
                       standard == "N")$station


not_std_2020 <- filter(sets_2020,
                       standard == "N")$station

# Not standard in both:
not_std_2018_and_2020 <- intersect(not_std_2018, not_std_2020)
```

```
not_std_2018_and_2020
>  [1] "2261" "2263" "2262" "2266" "2264" "2272" "2275" "2268" "2290" "2323"
> [11] "2320" "2316" "2312" "2314" "2308" "2309" "2304" "2302" "2295" "2296"
> [21] "2297" "2315" "2333" "2329" "2328" "2322" "2305" "2287" "2288" "2289"
> [31] "2273" "2298"

length(not_std_2018)
> [1] 130
length(not_std_2020)
> [1] 33
length(not_std_2018_and_2020)
> [1] 32

# 2018 has some east of the map, all non-standard:
filter(hooks_with_bait_revert$set_counts, year == 2018, lon > -124)$standard
>  [1] N N N N N N N N N N N N N N
> Levels: Y N
nrow(filter(hooks_with_bait_revert$set_counts, year == 2018, lon > -124))
> [1] 14

std_in_2018_but_not_std_in_2020 <- intersect(filter(sets_2018,
                                                    standard == "Y")$station,
                                             not_std_2020)
std_in_2018_but_not_std_in_2020
> [1] "2343"

not_std_in_2018_but_std_in_2020 <- intersect(not_std_2018,
                                             filter(sets_2020,
                                                    standard == "Y")$station)
not_std_in_2018_but_std_in_2020
>  [1] "2258" "2265" "2269" "2270" "2267" "2293" "2321" "2326" "2330" "2331"
> [11] "2299" "2317" "2334" "2335" "2332" "2327" "2324" "2318" "2285" "2311"
> [21] "2313" "2292" "2276" "2278" "2271" "2274" "2277" "2279" "2283" "2284"
> [31] "2280" "2307" "2303" "2301" "2294" "2306" "2291" "2286"

# setdiff(x, y) - elements in x but not in y
# setdiff(not_std_2018, not_std_2020) - but 2020 fewer coverage so misleading
```

Plot stations not standard in 2018 but standard in 2020, and vice versa, using each years' lats and lons (to verify that they all still agree – i.e., that station numbers have consistent lats and lons), and show 2019 data to check no 'usual' stations are non-standard in 2018 or 2020:

```
plot_BC()
points(lat~lon,
       data = filter(sets_2018,
```

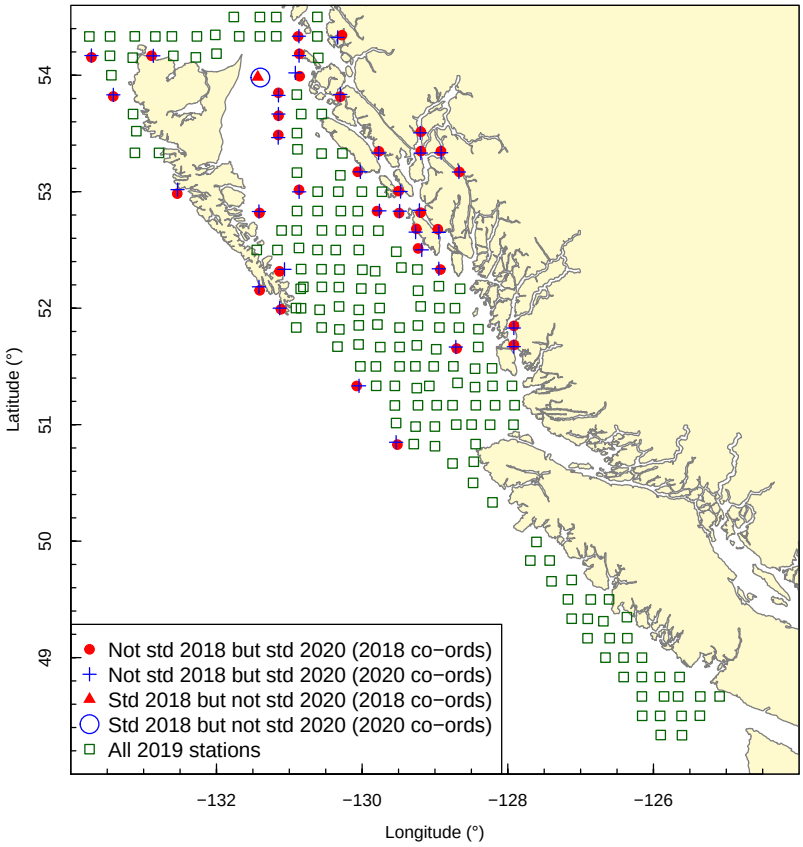
```
                    station %in% not_std_in_2018_but_std_in_2020),
       col="red",
       pch = 19)

# Do the same but using 2020 station co-ordinates - should overlap:
points(lat~lon,
       data = filter(sets_2020,
                    station %in% not_std_in_2018_but_std_in_2020),
       col="blue",
       pch = 3)

# And show the single station std in 2018 but not 2020:
points(lat~lon,
       data = filter(sets_2018,
                    station %in% std_in_2018_but_not_std_in_2020),
       col="red",
       pch = 17)
points(lat~lon,
       data = filter(sets_2020,
                    station %in% std_in_2018_but_not_std_in_2020),
       col="blue",
       pch = 1,
       cex = 2)

# Now show all 2019 stations:
points(lat~lon,
       data = filter(hooks_with_bait_revert$set_counts,
                    year == 2019),
       col="darkgreen",
       pch = 0)

legend("bottomleft",
       legend = c("Not std 2018 but std 2020 (2018 co-ords)",
                  "Not std 2018 but std 2020 (2020 co-ords)",
                  "Std 2018 but not std 2020 (2018 co-ords)",
                  "Std 2018 but not std 2020 (2020 co-ords)",
                  "All 2019 stations"),
              pch = c(19, 3, 17, 1, 0),
              pt.cex = c(1, 1, 1, 2, 1),
              col = c("red", "blue", "red", "blue", "darkgreen"))
```



Check if the one standard station in 2018 but not in 2020 (not fished at all in 2019) appears in any earlier years:

```
filter(hooks_with_bait_revert$set_counts,
       station == std_in_2018_but_not_std_in_2020) %>%
  as.data.frame()
>   year station     lat       lon  E_it N_it     C_it   E_it20 N_it20    C_it20
> 1 2018    2343 53.9840 -131.4333 7.027  110 15.65391 1.405400     16 11.384659
> 2 2020    2343 53.9802 -131.3977    NA   NA       NA 1.606061      8  4.981132
>   usable standard
> 1      Y        Y
> 2      Y        N
```

Was only fished in 2018 and 2020, so we should define it as non-standard.

So, the conclusions from this section are that we should:

1. Retain the 2018 definitions of standard stations, not the new ones defined in 2020:

```
sets_simp_std_corrected <- sets_simp_std

sets_simp_std_corrected$standard[sets_simp_std_corrected$station %in%
                                   not_std_in_2018_but_std_in_2020] <- "N"
# cbind(sets_simp_std$standard, sets_simp_std_corrected$standard) # to check
```

2. Define station 2343 as non-standard (over-riding original 2018 designation).

So check which functions that need changing, since they create a 'standard' column. These do not need changing: `get_iphc_hooks()` and `get_iphc_skates_info`.

Then `get_iphc_sets_info()` does return `standard`, but the `standard` designation is not saved in GFbio it is saved in `setDataExpansion` in `gfiphc`. So just need to add a line in `IPHC-stations-expanded.R` and then re-save all .rda files. Fixed that, now recreating all .rda files, as per the README.

Species counts

First, get the species counts into the desired format (to match `countData2013` shown earlier):

```
counts_raw <- readr::read_csv("non-halibut-data-2020.csv") %>%
  dplyr::mutate_if(is.character, factor)
> Parsed with column specification:
> cols(
>   `Row number` = col_number(),
>   Year = col_double(),
>   Stlkey = col_double(),
>   Station = col_double(),
>   Setno = col_double(),
>   `IPHC Species Code` = col_double(),
>   `Scientific Name` = col_character(),
>   `Species Name` = col_character(),
>   SampleType = col_character(),
>   HooksFished = col_double(),
>   HooksRetrieved = col_double(),
>   HooksObserved = col_double(),
>   `Number Observed` = col_double()
> )

counts_raw
> # A tibble: 1,441 x 13
>    `Row number`  Year Stlkey Station Setno `IPHC Species C~ `Scientific Nam~
>           <dbl> <dbl>  <dbl>   <dbl> <dbl>            <dbl> <fct>
>  1            1  2020 2.02e7    2059     1                2 Atheresthes sto~
>  2            2  2020 2.02e7    2059     1               27 Anoplopoma fimb~
```

```
>  3           3  2020 2.02e7    2059     1              54 Squalus suckleyi
>  4           4  2020 2.02e7    2059     1             143 Raja rhina
>  5           5  2020 2.02e7    2059     1             303 <NA>
>  6           6  2020 2.02e7    2059     1             304 <NA>
>  7           7  2020 2.02e7    2059     1             305 <NA>
>  8           8  2020 2.02e7    2052     2               2 Atheresthes sto~
>  9           9  2020 2.02e7    2052     2              27 Anoplopoma fimb~
> 10          10  2020 2.02e7    2052     2              54 Squalus suckleyi
> # ... with 1,431 more rows, and 6 more variables: `Species Name` <fct>,
> #   SampleType <fct>, HooksFished <dbl>, HooksRetrieved <dbl>,
> #   HooksObserved <dbl>, `Number Observed` <dbl>
summary(counts_raw)
>    Row number        Year          Stlkey            Station
>  Min.   :   1   Min.   :2020   Min.   :20200071   Min.   :2044
>  1st Qu.: 361   1st Qu.:2020   1st Qu.:20200262   1st Qu.:2097
>  Median : 721   Median :2020   Median :20200322   Median :2144
>  Mean   : 721   Mean   :2020   Mean   :20200496   Mean   :2176
>  3rd Qu.:1081   3rd Qu.:2020   3rd Qu.:20200858   3rd Qu.:2279
>  Max.   :1441   Max.   :2020   Max.   :20200954   Max.   :2343
>
>      Setno        IPHC Species Code            Scientific Name
>  Min.   :  1.00   Min.   :  2.0     Squalus suckleyi   :161
>  1st Qu.: 25.00   1st Qu.: 54.0     Anoplopoma fimbria :103
>  Median : 49.00   Median :143.0     Raja rhina         : 92
>  Mean   : 56.86   Mean   :179.2     Sebastes babcocki  : 57
>  3rd Qu.: 89.00   3rd Qu.:304.0     Sebastes ruberrimus: 52
>  Max.   :139.00   Max.   :307.0     (Other)            :317
>                                     NA's               :659
>                Species Name  SampleType    HooksFished    HooksRetrieved
>  Empty Hook          :198   20Hook:1441   Min.   :776.0   Min.   :686.0
>  Hook with Skin      :194                 1st Qu.:792.0   1st Qu.:792.0
>  Hook with Bait      :186                 Median :792.0   Median :792.0
>  Spiny Dogfish       :161                 Mean   :791.8   Mean   :790.5
>  Sablefish (Blackcod):103                 3rd Qu.:792.0   3rd Qu.:792.0
>  Longnose Skate      : 92                 Max.   :800.0   Max.   :800.0
>  (Other)             :507
>  HooksObserved   Number Observed
>  Min.   :140.0   Min.   :  1.00
>  1st Qu.:160.0   1st Qu.:  2.00
>  Median :160.0   Median :  6.00
>  Mean   :159.7   Mean   : 20.56
>  3rd Qu.:160.0   3rd Qu.: 25.00
>  Max.   :160.0   Max.   :151.00
>
```

```
testthat::expect_equal(unique(counts_raw$Year), 2020)
testthat::expect_equal(unique(counts_raw$SampleType), as.factor("20Hook"))

# This mismatches for 2020:
# testthat::expect_equal(length(unique(counts_raw$Station)),
#                        length(sets_raw$Station))

unique(counts_raw$"Species Name")
>  [1] Arrowtooth Flounder         Sablefish (Blackcod)
>  [3] Spiny Dogfish               Longnose Skate
>  [5] Hook with Skin              Empty Hook
>  [7] Hook with Bait              Yelloweye Rockfish
>  [9] Fish-eating Star            Bent/Broken/Missing
> [11] Silvergray Rockfish         Big Skate
> [13] Redbanded Rockfish          Quillback Rockfish
> [15] Blackspotted Rockfish       Sand Dab
> [17] Oregon Rock Crab            Spotted Ratfish
> [19] Lingcod                     Pacific Cod
> [21] Petrale Sole                Blue Shark
> [23] Rougheye Rockfish           Brittle Star
> [25] Yellowmouth Rockfish        Shortspine Thornyhead
> [27] Shortraker Rockfish         Bocaccio
> [29] Copper Rockfish             Glass Sponge
> [31] Sea Urchin                  Canary Rockfish
> [33] unident. thornyhead (Idiot) Sleeper Shark
> [35] Aleutian Skate              unident. Sponge
> [37] unident. Coral              Octopus
> [39] Sea Anemone                 Soupfin Shark
> [41] unident. Starfish           Wolf-Eel
> [43] Gastropod                   Tiger Rockfish
> [45] unident. Sculpin            Basketstar
> [47] Red Tree Coral              Sunflower Sea Star
> [49] Giant Pacific Octopus       Sea Pen
> 50 Levels: Aleutian Skate Arrowtooth Flounder ... Yellowmouth Rockfish
```

Note that halibut are not included in these counts:

```
dplyr::filter(counts_raw, "Species Name" == "Pacific Halibut")
> # A tibble: 0 x 13
> # ... with 13 variables: `Row number` <dbl>, Year <dbl>, Stlkey <dbl>,
> #   Station <dbl>, Setno <dbl>, `IPHC Species Code` <dbl>, `Scientific
> #   Name` <fct>, `Species Name` <fct>, SampleType <fct>, HooksFished <dbl>,
> #   HooksRetrieved <dbl>, HooksObserved <dbl>, `Number Observed` <dbl>
```

which I presume explains why total number of counts for a station does not add up to

`HooksObserved`. See later for halibut calculations.

Need to remove the HAN records for the twice-fished station, which turns out to be set number 4 for station 2104:

```
dplyr::filter(counts_raw, Station == twice_fished) %>%
  dplyr::select(c("Station", "Setno", "Species Name",
                  "Number Observed")) %>%
    as.data.frame()
>    Station Setno          Species Name Number Observed
> 1     2104    25           Pacific Cod               1
> 2     2104    25         Spiny Dogfish              19
> 3     2104    25        Hook with Skin              11
> 4     2104    25            Empty Hook              62
> 5     2104    25        Hook with Bait              46
> 6     2104     4 Sablefish (Blackcod)               1
> 7     2104     4         Spiny Dogfish              17
> 8     2104     4  Silvergray Rockfish               1
> 9     2104     4            Blue Shark               1
> 10    2104     4        Hook with Skin               4
> 11    2104     4            Empty Hook              73
> 12    2104     4        Hook with Bait              43

dplyr::filter(sets_raw, Station == twice_fished)
> # A tibble: 2 x 33
>   `Row number`  Year Stlkey `Vessel code` Station Setno `IPHC Reg Area`
>          <dbl> <dbl>  <dbl> <fct>           <dbl> <dbl> <fct>
> 1           42  2020 2.02e7 VNI              2104    25 2B
> 2          108  2020 2.02e7 HAN              2104     4 2B
> # ... with 26 more variables: `IPHC Stat Area` <dbl>, `IPHC Charter
> #   Region` <fct>, Purpose <fct>, Date <fct>, Eff <fct>, Ineffcde <lgl>,
> #   BeginLat <dbl>, BeginLon <dbl>, `BeginDepth (fm)` <dbl>, EndLat <dbl>,
> #   EndLon <dbl>, `EndDepth (fm)` <dbl>, `MidLat fished` <dbl>, `MidLon
> #   fished` <dbl>, `AvgDepth (fm)` <dbl>, `Lat - Grid target` <dbl>, `Lon -
> #   Grid target` <dbl>, `032 Pacific halibut count` <dbl>, `U32 Pacific halibut
> #   count` <dbl>, `032 Pacific halibut weight` <dbl>, `U32 Pacific halibut
> #   weight` <dbl>, `No. skates set` <dbl>, `No. skates hauled` <dbl>, `Avg no.
> #   hook/skate` <dbl>, `Effective skates hauled` <dbl>, `Soak time
> #   (min.)` <dbl>
```

So use that here to remove the species counts for that vessel (note that vessel code is not in `counts_raw`)

```
dplyr::filter(counts_raw,
              Station == twice_fished & Setno == 4)
> # A tibble: 7 x 13
```

```
>   `Row number`  Year Stlkey Station Setno `IPHC Species C~ `Scientific Nam~
>          <dbl> <dbl>  <dbl>   <dbl> <dbl>            <dbl> <fct>
> 1          786  2020 2.02e7    2104     4               27 Anoplopoma fimb~
> 2          787  2020 2.02e7    2104     4               54 Squalus suckleyi
> 3          788  2020 2.02e7    2104     4               60 Sebastes brevis~
> 4          789  2020 2.02e7    2104     4              126 Prionace glauca
> 5          790  2020 2.02e7    2104     4              303 <NA>
> 6          791  2020 2.02e7    2104     4              304 <NA>
> 7          792  2020 2.02e7    2104     4              305 <NA>
> # ... with 6 more variables: `Species Name` <fct>, SampleType <fct>,
> #   HooksFished <dbl>, HooksRetrieved <dbl>, HooksObserved <dbl>, `Number
> #   Observed` <dbl>

# So just keep these:
dplyr::filter(counts_raw,
              !(Station == twice_fished & Setno == 4))
> # A tibble: 1,434 x 13
>    `Row number`  Year Stlkey Station Setno `IPHC Species C~ `Scientific Nam~
>           <dbl> <dbl>  <dbl>   <dbl> <dbl>            <dbl> <fct>
>  1            1  2020 2.02e7    2059     1                2 Atheresthes sto~
>  2            2  2020 2.02e7    2059     1               27 Anoplopoma fimb~
>  3            3  2020 2.02e7    2059     1               54 Squalus suckleyi
>  4            4  2020 2.02e7    2059     1              143 Raja rhina
>  5            5  2020 2.02e7    2059     1              303 <NA>
>  6            6  2020 2.02e7    2059     1              304 <NA>
>  7            7  2020 2.02e7    2059     1              305 <NA>
>  8            8  2020 2.02e7    2052     2                2 Atheresthes sto~
>  9            9  2020 2.02e7    2052     2               27 Anoplopoma fimb~
> 10           10  2020 2.02e7    2052     2               54 Squalus suckleyi
> # ... with 1,424 more rows, and 6 more variables: `Species Name` <fct>,
> #   SampleType <fct>, HooksFished <dbl>, HooksRetrieved <dbl>,
> #   HooksObserved <dbl>, `Number Observed` <dbl>

countData2020_no_halibut <- dplyr::filter(counts_raw,
                                  !(Station == twice_fished & Setno == 4)) %>%
  dplyr::select(year = Year,
                station = Station,
                spNameIPHC = "Species Name",
                specCount = "Number Observed") %>%
  arrange(station) %>%
  dplyr::mutate(year = as.integer(year),
                station = as.character(station),
                spNameIPHC = as.character(spNameIPHC),
                specCount = as.integer(specCount))
```

```
testthat::expect_equal(names(countData2013), names(countData2020_no_halibut))
countData2020_no_halibut
> # A tibble: 1,434 x 4
>     year station spNameIPHC          specCount
>    <int> <chr>   <chr>                   <int>
>  1  2020 2044    Lingcod                     1
>  2  2020 2044    Spiny Dogfish               1
>  3  2020 2044    Yelloweye Rockfish          2
>  4  2020 2044    Hook with Skin             44
>  5  2020 2044    Empty Hook                 63
>  6  2020 2044    Hook with Bait             38
>  7  2020 2044    Bent/Broken/Missing         3
>  8  2020 2045    Petrale Sole                1
>  9  2020 2045    Lingcod                     2
> 10  2020 2045    Yelloweye Rockfish          6
> # ... with 1,424 more rows
summary(countData2020_no_halibut)
>       year        station           spNameIPHC          specCount
>  Min.   :2020   Length:1434        Length:1434        Min.   :  1.00
>  1st Qu.:2020   Class :character   Class :character   1st Qu.:  2.00
>  Median :2020   Mode  :character   Mode  :character   Median :  6.00
>  Mean   :2020                                         Mean   : 20.56
>  3rd Qu.:2020                                         3rd Qu.: 25.00
>  Max.   :2020                                         Max.   :151.00
```

Hooks observed and retrieved

Now, obtain the numbers of hooks observed and retrieved from `counts_raw`, to then merge into the set details:

```
hook_details <- dplyr::filter(counts_raw,
                              !(Station == twice_fished & Setno == 4)) %>%
  dplyr::group_by(Station) %>%
  dplyr::summarise(year = unique(Year),
                   hooksRetr = unique(HooksRetrieved),
                   hooksObs = unique(HooksObserved)) %>%
  dplyr::rename(station = Station) %>%
  dplyr::ungroup() %>%
  arrange(station) %>%
  dplyr::mutate(year = as.integer(year),
                station = as.character(station))
> `summarise()` ungrouping output (override with `.groups` argument)
```

```
hook_details
> # A tibble: 197 x 4
>    station  year hooksRetr hooksObs
>    <chr>   <int>     <dbl>    <dbl>
>  1 2044     2020       792      160
>  2 2045     2020       792      160
>  3 2046     2020       792      160
>  4 2047     2020       792      160
>  5 2048     2020       792      160
>  6 2049     2020       792      160
>  7 2050     2020       792      160
>  8 2051     2020       792      160
>  9 2052     2020       792      160
> 10 2053     2020       792      160
> # ... with 187 more rows

expect_equal(sets_simp_std_corrected$station, hook_details$station)
```

So now need to get the hook details into the set details, and keep columns as for `setData2013` but also with `standard`, and may as well keep `hooksRetr` and `hooksObs`:

```
setData2020 <- dplyr::left_join(sets_simp_std_corrected,
                                hook_details,
                                by = c("year", "station")) %>%
  dplyr::mutate(E_it20 = effSkateIPHC * hooksObs / hooksRetr) %>%
  dplyr::select(year,
                station,
                lat,
                lon,
                avgDepth,
                effSkateIPHC,
                E_it20,
                usable,
                standard,
                hooksRetr,
                hooksObs) %>%
  dplyr::mutate(year = as.integer(year),
                station = as.character(station),
                avgDepth = as.integer(avgDepth),
                usable = as.character(usable),
                standard = as.factor(standard))
setData2020
> # A tibble: 197 x 11
>     year station   lat   lon avgDepth effSkateIPHC E_it20 usable standard
>    <int> <chr>   <dbl> <dbl>    <int>        <dbl>  <dbl> <chr>  <fct>
```

```
>  1  2020 2044     50.8 -129.       50         7.95   1.61 Y      Y
>  2  2020 2045     50.8 -129.       68         7.95   1.61 Y      Y
>  3  2020 2046     51.0 -128.       53         7.95   1.61 Y      Y
>  4  2020 2047     51.0 -129.       39         7.95   1.61 Y      Y
>  5  2020 2048     51.0 -129.       47         7.95   1.61 Y      Y
>  6  2020 2049     51.0 -129.       86         7.95   1.61 Y      Y
>  7  2020 2050     51   -130.      131         7.95   1.61 Y      Y
>  8  2020 2051     51.2 -128.       58         7.95   1.61 Y      Y
>  9  2020 2052     51.2 -128.      106         7.95   1.61 Y      Y
> 10  2020 2053     51.2 -129.       56         7.95   1.61 Y      Y
> # ... with 187 more rows, and 2 more variables: hooksRetr <dbl>, hooksObs <dbl>
testthat::expect_equal(names(setData2013), names(setData2020)[1:ncol(setData2013)])
summary(setData2020)
>       year        station               lat             lon
>  Min.   :2020   Length:197         Min.   :50.82   Min.   :-133.7
>  1st Qu.:2020   Class :character   1st Qu.:51.83   1st Qu.:-131.1
>  Median :2020   Mode  :character   Median :52.66   Median :-130.3
>  Mean   :2020                      Mean   :52.70   Mean   :-130.3
>  3rd Qu.:2020                      3rd Qu.:53.51   3rd Qu.:-129.3
>  Max.   :2020                      Max.   :55.00   Max.   :-126.8
>     avgDepth       effSkateIPHC       E_it20          usable           standard
>  Min.   :  9.00   Min.   :6.890   Min.   :1.405   Length:197         N: 71
>  1st Qu.: 45.00   1st Qu.:7.950   1st Qu.:1.606   Class :character   Y:126
>  Median : 79.00   Median :7.950   Median :1.606   Mode  :character
>  Mean   : 87.87   Mean   :7.936   Mean   :1.603
>  3rd Qu.:119.00   3rd Qu.:7.950   3rd Qu.:1.606
>  Max.   :287.00   Max.   :8.030   Max.   :1.606
>    hooksRetr        hooksObs
>  Min.   :686.0   Min.   :140.0
>  1st Qu.:792.0   1st Qu.:160.0
>  Median :792.0   Median :160.0
>  Mean   :790.6   Mean   :159.7
>  3rd Qu.:792.0   3rd Qu.:160.0
>  Max.   :800.0   Max.   :160.0
```

Pacific Halibut counts

As noted above, the data extraction for the counts is for all non-halibut species. We still want the halibut counts for just the first 20 hooks – the `data_for_all_species` vignette (for data up to 2019) shows that the 20-hook and full hook counts (Series A and B) are very similar when rescaled, and the rescaling is miniscule with $G_A/G_B = 1.005$. So this justifies sticking with 20-hook counts for halibut, even though the full data are available for all sets, given it is a halibut survey. (Using all hooks for all years could be done, but would be a lot

of new code).

There are two options for getting halibut counts for the first 20 hooks (given we don't have hook-by-hook data, though it could probably be obtained just not from the IPHC website).

Option 1.

Take the halibut counts for all the hooks and create `N_it20_halibut_est = E_it20 / E_it * N_it`, or equivalently just `N_it20_halibut_est = hooksObs / hooksRetr * N_it`. Note that observed refers to observed for non-halibut species (presumably `hooksRetr` works for halibut). Not strictly the first 20 hooks, but is a rescaling. But will not guarantee integer values.

```
setData2020_and_halibut <-
  dplyr::left_join(setData2020,
                   dplyr::select(sets_simp_std_corrected,
                                 c(station,
                                   U32halibut,
                                   O32halibut)),
                   by = "station") %>%
  dplyr::mutate(N_it_halibut = U32halibut + O32halibut,
                N_it20_halibut_opt_1 = hooksObs / hooksRetr * N_it_halibut)
setData2020_and_halibut %>% dplyr::select(station,
                                          N_it_halibut,
                                          N_it20_halibut_opt_1)
> # A tibble: 197 x 3
>    station N_it_halibut N_it20_halibut_opt_1
>    <chr>          <dbl>                <dbl>
>  1 2044              60                12.1
>  2 2045              47                 9.49
>  3 2046              17                 3.43
>  4 2047              52                10.5
>  5 2048              12                 2.42
>  6 2049              31                 6.26
>  7 2050              46                 9.29
>  8 2051              13                 2.63
>  9 2052               9                 1.82
> 10 2053              91                18.4
> # ... with 187 more rows
```

Option 2.

Add the counts for each set (which include `Hook with Skin` etc.) and compare with `hooksObs`. I think the latter is higher, and the difference is halibut (as the only `non non-halibut` species).

Compare with the results from option 1. If close then use option 2, since it will be just be halibut counts and gives an integer number.

Add counts for each set:

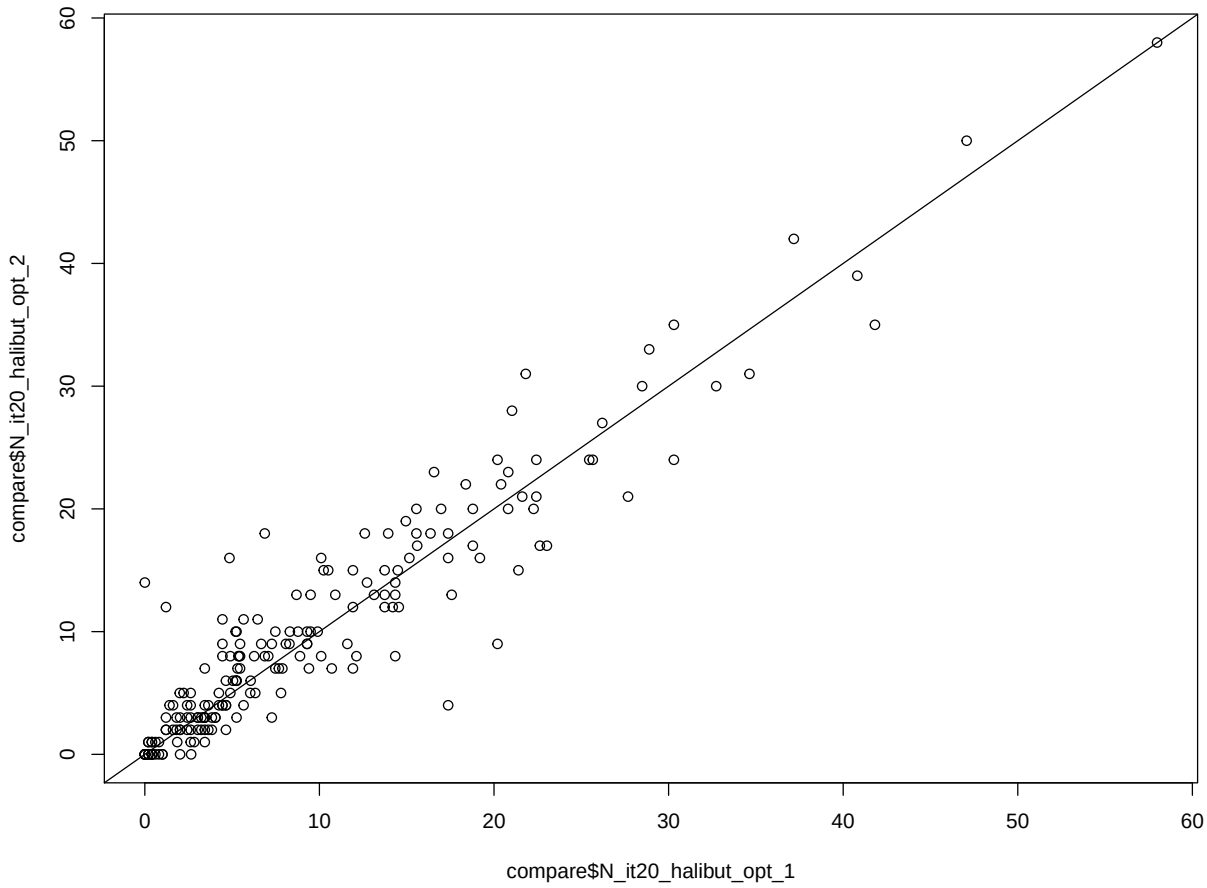
```
counts_20 <- countData2020_no_halibut %>%
  dplyr::group_by(station) %>%
  dplyr::summarise(non_halibut = sum(specCount)) %>%
  dplyr::ungroup()
> `summarise()` ungrouping output (override with `.groups` argument)
counts_20
> # A tibble: 197 x 2
>    station non_halibut
>    <chr>         <int>
>  1 2044            152
>  2 2045            147
>  3 2046            157
>  4 2047            145
>  5 2048            157
>  6 2049            152
>  7 2050            151
>  8 2051            159
>  9 2052            157
> 10 2053            138
> # ... with 187 more rows
```

Now join the two options together to calculate `N_it20_halibut_opt_2` and compare the two estimates of `N_it20_halibut`:

```
compare <-
  dplyr::left_join(setData2020_and_halibut,
                   counts_20,
                   by = "station") %>%
  dplyr::mutate(N_it20_halibut_opt_2 = hooksObs - non_halibut,
                N_it20_opt_1_over_opt_2 = N_it20_halibut_opt_1 / N_it20_halibut_opt_2) %
  dplyr::select(year,
                station,
                usable,
                N_it20_halibut_opt_1,
                N_it20_halibut_opt_2,
                N_it20_opt_1_over_opt_2)
compare$spNameIPHC <- "Pacific Halibut"
compare
> # A tibble: 197 x 7
>     year station usable N_it20_halibut_~ N_it20_halibut_~ N_it20_opt_1_ov~
>    <int> <chr>   <chr>             <dbl>            <dbl>            <dbl>
```

```
>  1  2020 2044    Y                12.1                 8              1.52
>  2  2020 2045    Y                 9.49               13              0.730
>  3  2020 2046    Y                 3.43                3              1.14
>  4  2020 2047    Y                10.5                15              0.700
>  5  2020 2048    Y                 2.42                3              0.808
>  6  2020 2049    Y                 6.26                8              0.783
>  7  2020 2050    Y                 9.29                9              1.03
>  8  2020 2051    Y                 2.63                1              2.63
>  9  2020 2052    Y                 1.82                3              0.606
> 10  2020 2053    Y                18.4                 22             0.836
> # ... with 187 more rows, and 1 more variable: spNameIPHC <chr>

plot(compare$N_it20_halibut_opt_1, compare$N_it20_halibut_opt_2)
abline(a = 0, b = 1)
```



So this is the right approach, though numbers not quite as close as may have thought. But these data are used for aggregating across all stations in a year (and any further analyses on halibut should be done using the raw data anyway). And the means aren't too bad:

```
mean(compare$N_it20_halibut_opt_1)
> [1] 9.593958
mean(compare$N_it20_halibut_opt_2)
> [1] 10.04569
```

So either of these would work. So use option 2 since gives an integer count:

```
compare$N_it20_halibut_opt_2
>   [1]  8 13  3 15  3  8  9  1  3 22 18  7  4  2  6  3 17  2  6 16  7  6  0  3  7
>  [26] 10 50 12  5 10  5  2  9  4  4 10 24 13  4  5 35 12  0  4  6  0 12 16  8 27
>  [51] 17 58  5 10 31 16  3  5 21  0  4 10 15 21 28  3  1 18  9 15 24  2  0 14 20
>  [76] 15  2  3  8  0  3  7  8 20  1 15  8  4  9  3 24  4 18 20 14  2 35 13 18  2
> [101] 11  7 30  3 39  9  7 24 17 20  4 13 30 13 42  5 13  4  2  2 16 18  8  6 31
> [126]  4 15  1  0  9  0 20  3  7  5  4  7 18  2 11  9  1 10 23 10  7  3 17  3  8
> [151] 33  1  3  0  9  6  8 22  4  0 14  9  3 19  8  0  8  0  0  2  2  0  0  5  0
> [176]  1  0  1  1 12  8  0 23  2 24 13  3 12 10  0 21  1  9  0 11 16  2
countData2020_halibut <- dplyr::select(compare,
                                       year,
                                       station,
                                       spNameIPHC,
                                       specCount = N_it20_halibut_opt_2) %>%
  dplyr::mutate(specCount = as.integer(specCount))
countData2020 <- rbind(countData2020_no_halibut,
                       countData2020_halibut) %>%
  dplyr::arrange(station)
# First time running, called the above countData2020_NEW to check remaining data didn'
#expect_equal(countData2020, filter(countData2020_NEW, spNameIPHC !=
#                                                      "Pacific Halibut"))
```

Check species names

The file `inst/extdata/iphc-spp-names.csv` contains species common names (as used for gfsynopsis, and a few extra like `unidentified skate`) and the IPHC common name.

These are IPHC names that are not given in `iphc-spp-names.csv`.

```
check_iphc_spp_name()
>  [1] "Unidentified Shark"          "Unident. Rockfish"
>  [3] "unident. thornyhead (Idiot)" "Grenadier (Rattails)"
>  [5] "Miscellaneous Shark"         "Eelpout"
>  [7] "unident. Roundfish"          "unident. Sculpin"
>  [9] "Unident. Flatfish"           "Greenland Turbot"
> [11] "unident. Hagfish"            "Starry Skate"
> [13] "Black Skate"                 "Brittle Star"
> [15] "Glass Sponge"                "Basketstar"
```

```
> [17] "Blackspotted Rockfish"
```

These are the ones just for the new 2020 data:

```
check_iphc_spp_name(countData2020)
> [1] "unident. thornyhead (Idiot)" "Brittle Star"
> [3] "Glass Sponge"                "Basketstar"
> [5] "Blackspotted Rockfish"       "unident. Sculpin"
```

Of these, only the Thornyhead and Blackspotted Rockfish are likely of interest (Issues #17 and #18). And maybe the sharks.

Save data sets

```
usethis::use_data(countData2020,
                  overwrite = TRUE)
> v Setting active project to 'C:/andy18/github/gfiphc'
> v Saving 'countData2020' to 'data/countData2020.rda'
> * Document your data (see 'https://r-pkgs.org/data.html')

usethis::use_data(setData2020,
                  overwrite = TRUE)
> v Saving 'setData2020' to 'data/setData2020.rda'
> * Document your data (see 'https://r-pkgs.org/data.html')
```