**Purpose:** This function takes output from **logs\_and\_fishery\_data.r** and calculates the annual CPUE for particular banks along with an estimate of the variability of the CPUE data.

Notes: I believe there are several problems with the current version of this function,

* Data pulled from the transition period in late 1990’s BB and GB from one to two banks appears to miss important components of the catch and effort during that period.
* The Gin option would give incorrect values unless the first year of the function call was aligned with the table (1981), any other option would be incorrect
* When using “survyr” some of the functions appear to be pulling from the wrong year and giving somewhat incorrect results.
* Assumptions about the proportion caught on GBa and GBb before 1977 (and likely into the early 80’s) appear to be incorrect as they ignore a component of our data (we have information on Canadian catches in US waters in our data that was previously overlooked).

**Version Control:** Likely many versions of this function

**Libraries**:, splacns

**Function Arguments Summary**

1. **fishery.data**: Fishery data from **logs\_and\_fishery\_data.r**.
2. **yr**: The years of interest. Default is to use all years from 1955 to current year. Note that since this is using output from **logs\_and\_fishery\_data.r** ensure that the years in the call to that function include the years used in this function.
3. **bk**: Which bank to select from using abbreviated bank id. Default is GBa (Georges Bank)

The options are:

* + "GBa": Georges a
  + "GBb": Georges b
  + "BBn": Browns north
  + "BBs": Browns south
  + “Ger": German
  + "Mid": Middle
  + "Sab": Sable
  + "Ban": Banquereau
  + "SPB": St. Pierre Bank

1. **nafo.div**: Used if interested in a particular NAFO division within the bank chosen. This will

always just look at CPUE in NAFO division on the bank chosen. There are currently 42

possible NAFO areas to choose from (any combination of the 42 is acceptable just

ensure they align with the bank):

* + 4VN 4VS 4VSB 4VSC 4VSE 4VSV 4W 4WD 4WE 4WF 4WG 4WH 4WJ 4WK 4WL 4WM 4W 4X 4XL 4XM 4XN 4XO 4XP 4XQ 4XR 4XS 4XX 5Y 5YB 5YC 5YD 5YE 5YF 5ZC 5ZE 5ZEG 5ZEH 5ZEJ 5ZEM 5ZEN 5ZEO 5ZWQ

1. **Gin**: Use the table gbcatch which includes historic calculations of the actual catch on Georges

Bank. (T/F) default is F.

1. **export**: Export the annual CPUE table. (T/F) default is F
2. **method**: How are we calculating the annual CPUE + variance, options are “avgdaily” and

“jackknife”. Default option is “avgdaily’ which is simply calculating the mean for the year + variance using every data point available. The preferred, yet slower if using large datasets, option is to use ‘jackknife’ function based on Smith(1980) to obtain an unbiased estimate of CPUE and its variance.

1. **model.out**: Specifiy the ‘out’ object to have a specific format. (T/F) default is F.
2. **period**: Are we looking at the calendar year or the survey year. Calendar year = 'calyr' while survey year ='survyr'
3. **surv:** If period = ‘survyr’ do you want to split the data using the spring ‘May” or summer

“August” survey.

1. **direct:** Set working directory. default = "Y:/Offshore scallop/Assessment/Assessment\_fns/"

**Section 1**

Within this section the data are subdivided by year. The year division depends on whether we are looking at the survey year or calendar year. If looking at calendar year it is relatively straightforward, if looking at survey year your brain may start to melt later in the function.

* + The month of the survey (May or August depending which you are interested in) is the last month of the survey year.
    1. For example: if looking at the spring survey (May) a year is defined as June-May, the year tagged to this survey year is the larger of the two. Thus June 1992 – May 1993 is considered the survey year 1993 for later calculations.
  + If sub-dividing by survey year some data will “disappear” as it is incomplete.
    1. For example, if looking at years 1992-2009 spring survey year. The data from January-May 1992 is orphaned (to be a real survey year the June-December 1991 data is needed), thus it is discarded.
    2. Conversely, in the most recent year the data from June-December 2009 would orphaned and would represent an incomplete survey year, thus it is discarded. The survey year 2009 would exist but as date from June 2008 – May 2009.
    3. If interested in the survey years it is recommended you select an extra year earlier and later (if possible) to ensure complete coverage of the years of interest.

**Section 2**

This part of the function largely deals with obtaining the Bank specific annual Catch and effort data. For all banks after 1999 this is straightforward. For the years before 1999 it is more complex for GB and BB as they were not yet subdivided. To get an estimate of the total catch these sub-banks from 1998 and earlier the proportion of the data we have available that was found to be on a particular sub-bank (say GBa on GB) was calculated. From that proportion we determine how much of the total catch was obtained on that bank. Problematically, this excludes varying amounts of catch that was obtained in the US portion of Georges Bank from before 1977 and even into the early 1980’s. If we are interested in getting this correct that US Georges bank information should be included in some way in these calculations. This is not a problem on Browns bank as data found to have come from US waters was originally sourced as GB.

**Section 3**

In this section the CPUE is calculated. There are two options, to simply use the mean CPUE calculated from the data, or to use the Smith(1980) **jackknife** function. The mean CPUE estimates are biased (because it is a ratio) and we should be using the annual estimates from **jackknife** whenever possible, using all the GBa data (from 1955-present) will take about 10 minutes with this function, while smaller banks run in a matter of seconds.. Also, in the past we have used the “err” variance calculated from the **jackknife** function incorrectly. See the jackknife documentation for details, but quite simply use err = ‘se’ and use the square root of these values as your standard error, this option also appears to run slightly more efficiently (about 20% quicker).

There is also an optional section here in which the catch on Georges Bank can be pro-rated from the historically reported catch data found in the file *‘gbcatch.txt’*. The difference between the default method and this option appear minimal. It isn’t clear if this is still actively used option. The remainder of the function deals with exporting tables and/or objects to other functions.

**Function Index**

%in%

any

as.Date

as.numeric

c

data.frame

dimnames

for

function

If

is.na

length

merge

min

max

paste

print

rbind

read.table

require

return

rowSums

source

subset

stop

sum

tapply

unique

unlist

which

with

write.table