

NCIFD R Package Demo

A. Powell Wheeler

March 24, 2022

Contents

Document Development	2
Introducton	2
R	2
RStudio	2
R Packages	2
What can a Division R Package Do for Us?	2
Data Sets	2
Functions	3
NCIFD Introduction	3
Install NCIFD	3
Load Library	3
Getting Package-wide Help	4
NCIFD Contents	4
Data Sets	4
Using Package Data Sets	5
Browsing a data set in R	5
Help for data sets	5
Data Sets Depend Each Other	6
Data Sets in the Source Code	6
Uses of NCIFD Data: Graphs	6
Functions	7
cleanBIODE()	7
standardWeight()	7
relativeWeight()	8
Z2A()	9
Data.frame viewing functions	9
ferret()	10
ordinalDate()	11

Document Development

This document was written in Rmarkdown (2.12) on R (4.1.2) and rendered on March 08, 2022 at 13:22 in RStudio (2021.9.0.351) using NCIFD (1.0.1) and knitr (1.37).

Introduciton

R

R is the free and open-source programming language that is becoming the standard tool for data analysis, not just in science, but across disciplines.

RStudio

RStudio is the free and open-source IDE (Integrated Desktop Environment) that is used by 99.99% of R users. It runs R inside of it and has lots of helpful bells and whistles that make using R much easier. RStudio is one of the reasons for the success of R.

Although they are a for-profit company, [RStudio is also Public Benefit Corporation which means they have legally-binding altruistic goals](#) including keeping RStudio free for individual users. [They make money selling and supporting enhanced versions of their free products for industry](#), which is a common open-source software business model. Thus, there is no rational basis to fear that, after investing your time in learning R in RStudio, you might have it taken away or somehow be forced to pay for it.

- RStudio GUI is actually is rendering a web page. Thus, it looks and works the same across platforms (Windows, MacOS, and Linux) and you could remotely access a powerful RStudio server through your web browser.
- Dark theme: tools -> global options -> appearance -> Editor Theme. I use Ambiance or Sky.
- Four panels can be resized or minimized by dragging their frames.
- Code editor panel can be dragged to a separate monitor.
- Tab autocomplete in console
- Up and Down Arrows to access previous commands

R Packages

R Packages allow R users to easily share functions and data sets with each other. [CRAN](#), the primary website for downloading R packages currently has [> 18k](#). R packages are most commonly used to share functions, but they are also useful for data distribution. The main goal of NCIFD is to clean-up and distribute NCWRC data sets among NCWRC staff, but it also stores some functions you may find useful.

What can a Division R Package Do for Us?

Data Sets

IFD has many relatively small data sets that are relevant outside of a single district but they can be difficult to find because there is no single source for their distribution. In addition, they are typically difficult to use because they are not arranged neatly in a single 2-dimensional data set and/or are very messy. For examples:

- Available through PAWS but exported data is very messy
 - MTSL (2011-present) - years are separate spreadsheets, difficult to piece together, waterbody names not standardized
 - WWSL (2011-present) - years are separate spreadsheets and difficult to piece together, waterbody names not standardized
 - NCARP - can only be downloaded by species groups, freedom units
 - Coldwater Stocking Trips: no county names, no waterbody names, freedom units, some bad waterbody and county codes, some wrong and missing PMTW classifications, some trips entered twice, water temps in mixed units
- Fisheries Research with no home on PAWS
 - Black Bass Genetics - plus each sample results are separate spreadsheets in differing formats
 - Wild Trout Distribution
 - Wild Trout Barriers
- No Standardized Distribution

- # Functions

We can make R functions to solve IFD problems (like cleaning-up the output from BIODE queries) or share our homebrewed functions inside the Divison.

NCIFD Introduction

NCIFD is an R package for the Inland Fisheries Division (IFD) of the North Carolina Wildlife Resources Commission. It contains functions and data sets authored by the staff. The first non-beta version Aggressive Alosid version 1.0.0 was released on 2022-02-08. A ‘patch’ version 1.0.1 was released recently. Minor releases with new and updated functions and data sets are expected every July (after the spring stockings), November (after the normal fall stockings), and February (after the new MTLs and WWSLs are finalized).

Install NCIFD

NCIFD is distributed from a MS Teams group for IFD R-users. E-mail [Powell Wheeler](#) if you want access to the shared folder. There are 5 files available after each release:

1. NCIFD_source.zip contains the source code. Its a zip-compressed directory structure that holds all the files that make the package.
2. IntroToNCIFD.pdf is a general overview of the package with installation procedures.
3. NEWS.pdf is the historical change log and some anticipated future changes.
4. NCIFD_X.X.X.zip installs NCIFD on MS Windows computers. To install the package in Windows, download and install it through RStudio's menus (Tools -> install.packages).
5. NCIFD_X.X.X_R_x86_64-pc-linux-gnu.tar.gz installs NCIFD on a Linux computer.

Load Library

```
library(NCIFD) #the library function makes NCIFD available in R

##
##   NCIFD - the R package of the Inland Fisheries Division of the NC Wildlife Resources Commission
##
##   Version:  1.0.1 (2022-02-25)    "Aggressive Alosid"
##
##                                     .',_
##                               _wnos;
##   _ssaaaaaajmmmmXov|                                     ._vl}
##   ._awumWWZXSSXSSVUUH##W#mqwaas,,. _auZnI>
##   =dZnQn2Z1)T*| | | | iiii{il*}***YY1YVXXSnoXXX1>~'
##   -11I!Yli| |i>| | +| +| +| +| =+++|=++| | | | |ilIIXos,
##   -^<i|iiiliiilviii| |i|ii| | | | +| | =| + ~"!Xmov|,
##   --~+| |i|l| |i| | | | +| +| +| +| =-+| | - ~!1ll=.
##   ---~--~+=+,. ^"
##   -
```

```
path.package('NCIFD') #NCIFD is stored in a directory on your hard drive. library() lets R access that directory.
```

```
## [1] "/usr/local/lib/R/site-library/NCIFD"
```

Getting Package-wide Help

NCIFD has package-wide help information that includes information about the project, as well as all the data sets and functions in the package. In RStudio, you can click links to view the specific help information for all the functions and data sets and there are hyperlinks for internet resources also. Everything in the package has useful help information.

```
help(NCIFD) #not run
```

- Demonstrate package-wide help(NCIFD) in RStudio
- Hyperlinks to internet resources
- Internal Links to detailed information on package functions and data sets.

NCIFD Contents

Data Sets

The versions in the package have has substantial clean-up and are instantly available in an R session. They are updated with the most recent data when new versions of the package are released.

- Admin
 - accountCodes. Contents of the PAWS database of Inland Fisheries Division publications.
 - staff. Work contact information for NCWRC employees.
- NC Information
 - counties. Code, district, and region information for NC counties.
 - waterbodies. Waterbody codes from Fish (1968).
 - fishes. Information on NC fishes including their official NCWRC abbreviations, common names, taxonomy, state and federal status, distribution, and ITIS numbers.
- Research Resources
 - afsFishes. Official list of fish names from the AFS.
 - missingReports. A list of NCWRC publications that are not stored on PAWS. Many are currently lost.
 - raleighLibrary. Inventory of the IFD and WMD libraries in the Raleigh Office.
 - reports. Contents of the PAWS database of Inland Fisheries Division publications.
- Research Projects
 - blackBassGenetics. Results of the on-going Black Bass Genetics Project.
 - troutBarriers. Fish movement barriers on NC wild trout streams.
 - troutDist. Distribution of wild trout in NC.
- Fishing
 - ncarp. NCARP awards data base records.
 - stateRecords. Current NC fishing records.
- Hatchery System
 - coldwaterStockingCoords. All the current trout stocking points.
 - coldwaterStockingTrips. All coldwater stocking records since July 1, 1991.
 - warmwaterStockingTrips. All warmwater stocking records since 2003 and some older ones back to 1972.
 - wwsl. All the warmwater fish stocking requests from 2011-present, plus some older entries back to 2005.
 - mtsl. All the Public Mountain Trout Water fish stocking requests beginning in 2001.
- Miscellaneous
 - townCoords. Average GPS coordinates of USA towns and cities.
 - zipCodeCoords. USPS ZIP Codes and their approximate GPS coordinates.

Using Package Data Sets

Browsing a data set in R

It's difficult to browse through data in R. This is often bewildering for those coming to R from Excel. Here are several ways to view your data:

1. Just type 'wvsl + ENTER' in the console window. If you type an object name and don't tell R what to do with it, R tries to print it out in the console window. This works for functions also.

```
wvsl #did not run. If you don't tell R what to do with something it will try to print to the screen.
```

```
head(wvsl) #only show top 6 rows. Notice it overflows the screen in this document.
```

```
##   year district    county      waterbody waterbodyCode designation priority spCode    commonName
## 1 2005         2   Beaufort   Tar/Pamlico River      TAR 1         MGT          1     SB   Striped Bass
## 2 2005         2   Carteret   Cedar Swamp Pond      MOT 2-15       Other          3     CC Channel Catfish
## 3 2005         2    Craven     Neuse River        NUS 1         MGT          1     SB   Striped Bass
## 4 2005         2 New Hanover   Cape Fear River      CFR 1         MGT          1     SB   Striped Bass
## 5 2005         2 New Hanover   NE Cape Fear River    NCF 1         MGT          1     SB   Striped Bass
## 6 2005         3   Halifax Roanoke Rapids Reservoir    RKE 1-32       MGT          1     SB   Striped Bass
##                                     genusSp fishSize requestN      stockLocation
## 1   Morone saxatilis    1-2"   100000      Washington
## 2 Ictalurus punctatus  10-12"    1000  Croatan National Forest
## 3   Morone saxatilis    1-2"  100000      WRC BAA: Bridgeton
## 4   Morone saxatilis    1-2"   50000      Wilmington
## 5   Morone saxatilis    1-2"   50000      WRC BAA: Castle Hayne
## 6   Morone saxatilis     6"   25000 WRC BAA: Thelma (SR 1422)
```

2. Use RStudio's Spreadsheet Interface

```
View(wvsl) #or open in IDE
```

3. Export the data set to a CSV file and view it in a spreadsheet program.

```
write.csv(wvsl, file = '~/Downloads/troutDist.csv', row.names = FALSE)
list.files("~/Downloads", 'trout.+csv', full.names = TRUE)
```

```
## [1] "/home/powell/Downloads/troutDist.csv"
```

- 4). NCIFD also has three functions that help browse data: dfScan(), dfCols(), dfSlim(), and ferret(). We'll look at those with the package functions.

Help for data sets

Help information for package data sets contains all the metadata that you need to understand the data.

```
help(troutDist) #not run
```

- format
 - data.frame: 2-dimensional data (like a spreadsheet)
 - number of rows and columns
 - column contents
- author - who did the R coding to get it in the package (Powell)
- source
 - person overseeing the data (Jake)
 - when updated
- examples - not very useful for data sets

Data Sets Depend Each Other

The data sets in {NCIFD} often integrate information from each other. For example, 'warmwaterStockingTrips' contains numeric codes for county names and uses 'counties' to decode them (e.g., 944 = Haywood). In addition, it uses 'fishes' to decode species codes into species names and 'waterbodies' and 'wwsl' to convert waterbody codes into waterbody names (Figure 1).

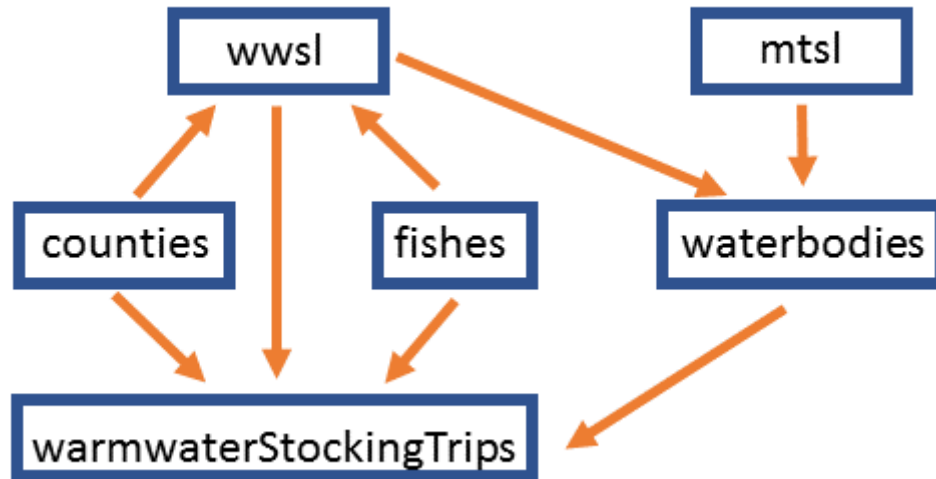


Figure 1: 'warmwaterStockingTrips' pulls information from 'wwsl', 'counties', 'fishes', and 'waterbodies'. In turn, 'wwsl' is built in part from 'fishes' and 'counties' and 'waterbodies' integrates information from 'wwsl' and 'mtsl'.

Data Sets in the Source Code

- Use coldwaterStockingTrips as an example
- NCIFD/data-raw
 - R scripts that clean-up the data and save it
- NCIFD/data
 - Data sets in R *.rda format
- NCIFD/R
 - Help Information coded in Roxygen2
- NCIFD/tests
 - Automated testing
 - Package has too many interconnected parts to check everything all the time
 - Sometimes find detect problems in the raw data, especially if hand-keyed

Uses of NCIFD Data: Graphs

- July PMTW Water Temperatures
 - temps.july.script.r
- Trout Stocking Requests and Results by Stream
 - stockingByWaterbodyByYear.R
- Trout Hatchery Requests and Output
 - totalHatcheryOutput.R

Functions

Some of the functions have specific IDF uses such as `cleanBIODE()`, `relativeWeight()`, `standardWeight()` and others were created to help with specific research projects. In addition, there are other functions (that are not listed here) which are used to help build the package.

- `cleanBIODE()` cleans-up MS Excel files produced when querying the BIODE database in PAWS.
- `flow()` calculates stream flow from interval velocity and depth measurements.
- `ordinalDate()` assigns ordinal dates (1:365).
- Basic Statistics
 - `movingAverage()` calculates moving average.
 - `quick2Sample()` two-sample inferential tests (t- and z-tests) for summarized data.
- Fisheries Statistics
 - `relativeWeight()` calculates fish relative weight.
 - `standardWeight()` calculates fish standard weight.
 - `Z2A()` converts instantaneous mortality rate (Z) to annual mortality rate (A) along with the SE and calculates CIs.
- Console Tools
 - `dfCols()` displays information about the columns in a data.frame including their class and address.
 - `dfScan()` views an evenly distributed subset of the rows in a data.frame.
 - `dfSlim()` views only as many data.frame columns as will fit cleanly across your console window.
 - `ferret()` finds rows in a data.frame which have a match for one or more search terms.
 - `fruitSalad()` creates a data.frame with information about fictitious salads.
 - `reveal()` reveals the libraries and objects that are active in your workspace.
- Unit Conversion Functions: `ac2ha()`, `c2f()`, `cubft2cubm()`, `cubft2gal()`, `cubm2cubft()`, `deg2rad()`, `f2c()`, `ft2m()`, `g2lb()`, `gal2cubft()`, `ha2ac()`, `in2mm()`, `kg2lb()`, `km2mi()`, `lb2g()`, `lb2kg()`, `m2ft()`, `mi2km()`, `mm2in()`, `rad2deg()`.
- NCIFD Development (Figure 2)
 - `addZeros()` adds zeros to the start or end of a string. Replaces zeros that were lost when spreadsheets treat an identifier, such as a PIT tag number, like a number and drop leading and trailing zeros.
 - `lake2End()` if a waterbody name starts with 'Lake', move it to the end of the name. So, 'Lake Fontana' becomes 'Fontana Lake'.
 - `stringCleaning()` corrects a universe of bad formatting often found in notes columns and decodes > 100 common IFD abbreviations.
 - `wordFreq()` finds the frequencies of words (by looking for spaces) in a vector or data.frame column. Its helps check the preformance of `stringCleaning()`.

cleanBIODE()

Cleans-up EXCEL files exported from BIODE queries.

```
cleanBIODE('~Downloads/BiodeQuery03-07-22.xlsx') #makes an object in your R Session
cleanBIODE('~Downloads/BiodeQuery03-07-22.xlsx', writeFiles=TRUE) #cleans-up and saves as CSV files on HDD
```

standardWeight()

Calculates standard weight of 47 NC fishes

- alias: `ws()`
- species info in hidden data.frame: `NCIFD:::wsLookup`
 - some species have multiple equations
 - * Defaults to most useful in NC (BKT) or most general (MKY)
- references in `help(standardWeight)`
- tested against published and known values - `NCIFD/tests/testthat/test_condition.R`

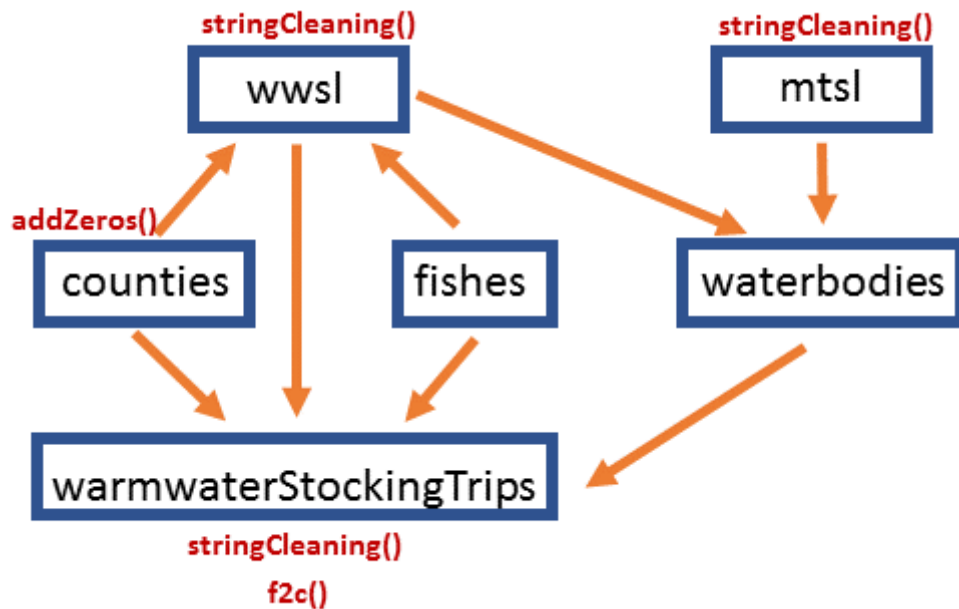


Figure 2: NCIFD functions that are used to build the package data sets are shown in red.

```
standardWeight('BKT', 150) #defaults to eq='A' (Harris et al. 2021); SABKT are skinnier
```

```
## [1] 32.7
```

```
standardWeight('BKT', 150, eq='B') #eq='B' selects Hyatt and Hubert (2001)
```

```
## [1] 36.85
```

```
standardWeight('BKT', 100) #Harris et al. (2021) works down to 80 mm TL
```

```
## [1] 9.55
```

```
standardWeight('BKT', 100, eq='B') #Hyatt and Hubert (2001) returns a NA when < 120 mm TL
```

```
## [1] NA
```

```
ws('BKT', 150) #ws() is an alias for standardWeight()
```

```
## [1] 32.7
```

relativeWeight()

Calculates relative weight of 47 NC fishes

- you can skip calculating W_s and get W_r directly
- uses `standardWeight()`
- alias: `wr()`
- tested against published and known values - NCIFD/tests/testthat/test_condition.R

```
relativeWeight('BKT', 150, 32.7) #relative weight also accepts fish TL. It calls standardWeight() internally.
```

```
## [1] 99.99
```



```
relativeWeight('BKT', 150, 32.7, eq='B')
```

```
## [1] 88.74
```

```
wr('BKT', 150, 32.7) #wr is an alias for relativeWeight()
```

```
## [1] 99.99
```

Z2A()

Quickly convert an instantaneous mortality rate (Z) to an annual mortality rate (A). Also, converts the standard error and bootstraps confidence intervals with the gamma distribution (Bolker 2008).

```
Z2A(0.69) #no SE specified
```

```
## $Instantaneous_Mortality
##      Z      SE(Z) Low95CI Up95CI
##    0.69         NA        NA      NA
##
## $Annual_Mortality
##    A_pct      SE(A) Low95CI Up95CI
##    49.8         NA        NA      NA
```

```
Z2A(0.69, 0.1) #with SE; defaults to 95% CIs
```

```
## $Instantaneous_Mortality
##      Z      SE(Z) Low95CI Up95CI
##    0.690    0.100    0.510    0.897
##
## $Annual_Mortality
##    A_pct      SE(A) Low95CI Up95CI
##    49.8       5.0     39.9    59.2
```

```
Z2A(0.69, 0.1, 0.99) #with SE and custom CI
```

```
## $Instantaneous_Mortality
##      Z      SE(Z) Low99CI Up99CI
##    0.690    0.100    0.457    0.970
##
## $Annual_Mortality
##    A_pct      SE(A) Low99CI Up99CI
##    49.8       5.0     36.7    62.1
```

Data.frame viewing functions

NCIFD contains several data sets, most of which are 2-dimensional data.frames. Because viewing data in R can be difficult, the package contains three functions that can help with that.

1. dfCols(): shows a summary of what is contained in the data.frame columns.

```
dfCols(zipCodeCoords) #no SE specified
```

```
##      column      mode      type      class length N_Obs N_NA N_Empty uniqueExamples
## 1      zipCode character character character    41873 41873    0      0 00501, 00544, 00601, 00602, 00603, ...
## 2  zipCodeType character character character    41873 41873    0      0 UNIQUE, STANDARD, PO BOX
## 3        town character character character    41873 41873    0      0 Holtsville, Adjuntas, Aguada, Aguad ...
## 4        state character character character    41873 41873    0      0 NY, PR, VI, MA, RI, NH, ME, VT, CT, ...
## 5 townAndState character character character    41873 41873    0      0 Holtsville, NY, Adjuntas, PR, Aguad ...
## 6          lat  numeric      double  numeric    41873 41873    0      0 40.81, 18.16, 18.38, 18.43, 18.18, ...
## 7          long  numeric      double  numeric    41873 41873    0      0 -73.04, -66.72, -67.18, -67.15, -66 ...
```

2. `dfScan()`: shows X number of evenly-spaced rows in a data.frame, including the first and last.

```
dfScan(afsFishes, 10) #no SE specified
```

##	commonName	genusSp	ref
## 1	Mud Lancelet	Branchiostoma bennetti	Boschung & Gunter 1966
## 431	Zabaleta Anchovy	Anchovia clupeioides	Swainson 1839
## 861	Greater Redhorse	Moxostoma valenciennesi	Jordan 1885
## 1290	Large-eye Silverside	Atherinella sallei	Regan 1903
## 1720	Speckled Scorpionfish	Pontinus sierra	Gilbert 1890
## 2150	Tiger Grouper	Mycteroperca tigris	Valenciennes 1833
## 2580	Yellowtail Jack	Seriola lalandi	Valenciennes 1833
## 3009	Socorro Wrasse	Halichoeres insularis	Allen & Robertson 1992
## 3439	Pallid Goby	Coryphopterus eidolon	Böhlke & Robins 1960
## 3869	Slender Mola	Ranzania laevis	Pennant 1776

3. `dfSlim()`: drops the right-side columns that won't fit on your screen to prevent wrap-around.

```
dfSlim(head(coldwaterStockingTrips)) #no SE specified
```

##	district	county	waterbody	waterbodyCode	pmtwClass	sizeCat	date	year	troutAll_n
## 1	9	Jackson	Balsam Lake	TUK 1-66-A	HS	C	1991-07-02	1991	700
## 2	9	Jackson	Tuckaseegee River	TUK 1	HS	C	1991-07-02	1991	1000
## 3	9	Macon	Cullasaja River	LTN 1-39	HS	C	1991-07-02	1991	1000
## 4	9	Macon	Ellijay Creek	LTN 1-39-4	HS	C	1991-07-02	1991	400
## 5	7	Wilkes	East Prong Roaring River	YAD 1-58-1	HS	C	1991-07-02	1991	250
## 6	7	Wilkes	Middle Fork Reddies River	YAD 1-63-2	HS	C	1991-07-02	1991	300

ferret()

Searches data.frames for strings and returns the rows that have matches. It is designed to quickly pull up information in NCIFD data.frames. With some package familiarity and practice, you can use it while talking to anglers on the phone. By default, it finds case-insensitive partial matches. For example, searching for 'WY' will match 'wy' and 'hwy'. However, if `exact=TRUE`, 'WY' will only match 'WY'. Here's some examples of how I've used it:

1. Is Charles D Owen still getting winter pond stockings?

```
ferret(coldwaterStockingTrips, 'charles')
```

2. You jerks only stocked Big Snowbird HS once all last year.

```
ferret(coldwaterStockingTrips, c('snowbird', '2021')) #search for two strings at once with c()
```

3. When are Striped Bass typically stocked in Lake Hiwassee?

```
ferret(warmwaterStockingTrips, c('sb', 'hiwassee')) #search for multiple strings at once with c()
```

4. Find all of David Yow's Creel Reports

```
ferret(reports, c('Yow', 'creel')) #search for multiple strings at once with c()
```

5. What months are Bear, Cedar Cliff, Wolf and Tanassee Stocked?

```
ferret(mts1, 'bear|cedar cliff|wolf|tanassee') #do an OR search with pipe symbol |; only works with exact=FALSE
```

6. What data do you have from Calderwood, Santeetlah, Cheoah, and Emory reservoirs?

```
ferret(reports, 'Calderwood|Santeetlah|Cheoah|Emory') #do an OR search with pipe symbol |; only works with exact=
```

7. When are Walleye normally stocked?

```
ferret(warmwaterStockingTrips, 'wy') #problem: finds 'wy' in notes "hwy"
#try exact matching
ferret(warmwaterStockingTrips, 'WY', exact = TRUE) #notice case sensitive
```

ordinalDate()

Convert dates into ordinal dates. For example, “2008-01-01” becomes 1 and “2000-01-02” becomes 2. This is useful for creating a common Jan-Dec x-axis when plotting multiple years of a time-series. R date format returns integers and POSIXct dates return decimal numbers.

```
#R Date Format
ordinalDate(as.Date(c("1/1/2000", "07/01/1999", "12/31/1970"), format = '%m/%d/%Y'))
```

```
## [1] 1 182 365
```

```
#POSIXct date and time format
dates <- c('2020-02-28 12:00:00', '2020-02-29 12:00:00', '2020-03-01 12:00:00')
ordinalDate(as.POSIXct(dates, tz = 'GMT+5'))
```

```
## [1] 59.5 NA 60.5
```

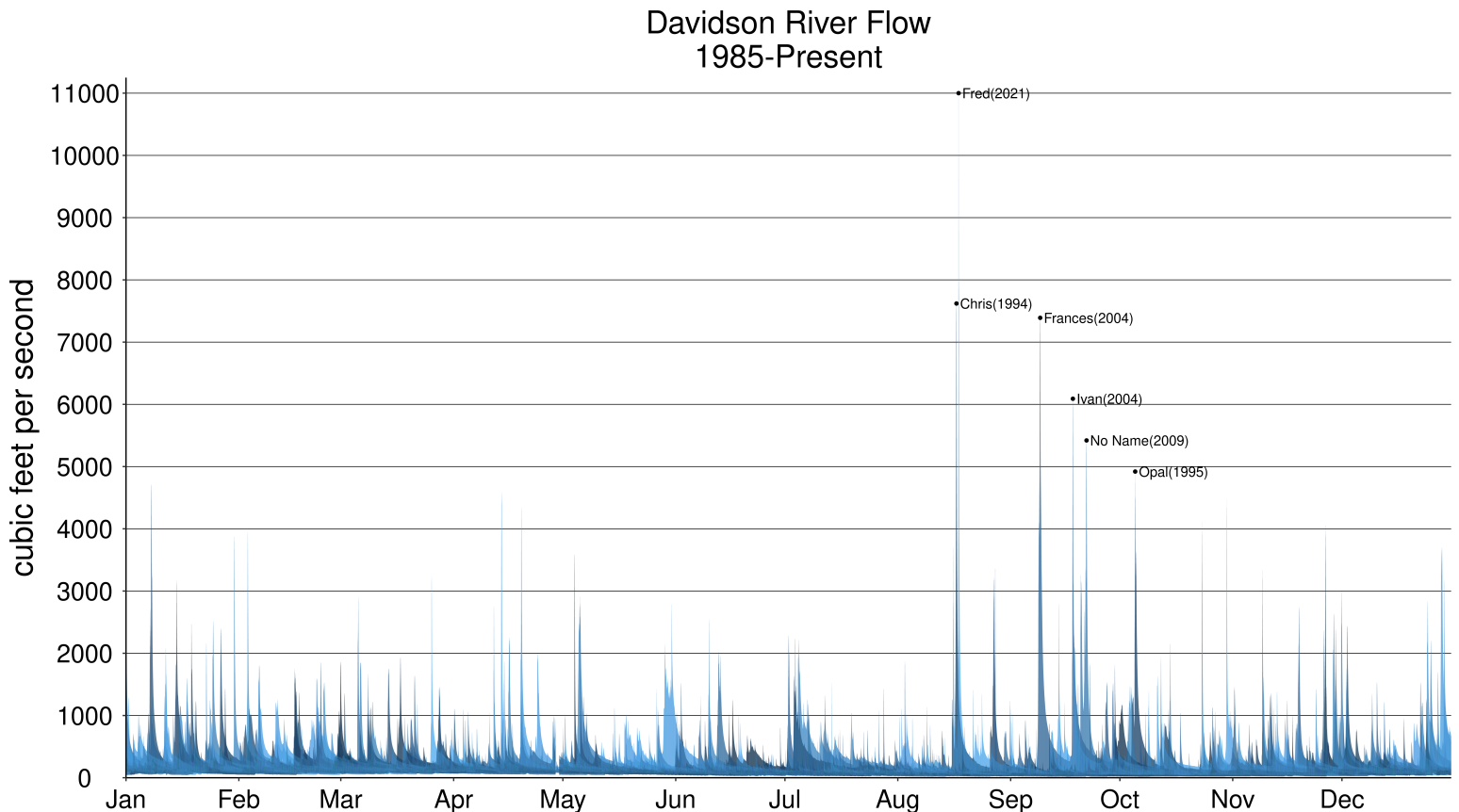


Figure 3: Daily flows on the Davidson River showing hurricane flows. `ordinalDate()` was used to create a common (1-365) x-axis for each year before the axis labels (months) were overlain.