{NCIFD}

An Inland Fisheries Division R Package

A. Powell Wheeler

March 24, 2022

Contents

R, RStudio, and R Packages **2**

R . . . 2

RStudio . . . 2

R Packages . . . 2

Division Data Problems **2**

Introduction to {NCIFD} **3**

Install . . . 3

Load Library . . . 3

Getting Help . . . 4

{NCIFD} Data Sets **4**

Browsing Data in R . . . 5

Help for Data Sets . . . 6

Data Sets are Interconnected . . . 6

Source Code for Data Sets . . . 7

Uses of NCIFD Data: Graphs . . . 7

{NCIFD} Functions **7**

Help for Functions . . . 8

Source Code for Functions . . . 9

Function Examples . . . 9

Conclusion **13**

Ready to Learn R? **13**

This document was written in R Markdown (2.13) on RStudio (2021.9.0.351) and rendered at March 10, 2022 at 16:02 using R (4.1.2), NCIFD (1.0.1), and knitr (1.37).

R, RStudio, and R Packages

R

R is a free and open-source programming language that has become a standard data analysis tool in sciences and other disciplines. R is fast, runs well on low-powered computers, can handle much larger data sets than spreadsheet programs, and incorporates reproducibility. R is the present and foreseeable-future of data analysis. When you learn R, you are learning the skills that will carry you through to the end of your career.

RStudio

RStudio is a company that makes a free and open-source program which is also called RStudio. RStudio is an Integrated Desktop Environment (IDE) that runs R inside of it, has a terrific code editor, and many other features that makes R easier to use. RStudio is used by about 99.99% of R users. In addition, RStudio (company) produces a series of add-on packages for R that are collectively known as the 'tidyverse' which also make R easier. RStudio and its products have contributed substantially to the success of R.

Although it is a for-profit company, RStudio is a Public Benefit Corporation with legally-binding altruistic goals including keeping RStudio free for individual users. They make money by selling and supporting enhanced versions of their free products for industry, which is a common open-source software business model. Open-source software users are often suspicious when big corporations get involved in open-source projects, but no one is worried about RStudio. Thus, currently there is no rational basis to fear that after investing your time learning R in RStudio, it might be taken away or you will be forced to pay for it.

- RStudio renders a web page on your screen. Thus, it looks and works the same across platforms (Windows, MacOS, and Linux) and you can use your web browser to access a remote RStudio server.
- Dark themes are available. tools -> global options -> appearance -> Editor Theme. I like Ambiance and Sky.
- RStudio splits your screen into four panels which can be resized or minimized by dragging their frames.
- Code editor panel can be dragged to a separate monitor.
- Tab auto-complete works in the console.
- Up and down arrows access the command history in the console.

R Packages

R packages allow R users to share two things with each other: 1) functions and 2) data sets. CRAN, the primary website for downloading R packages currently hosts over > 18k packages. Although R packages are typically downloaded from CRAN, they are stand-alone files and can be distributed in a variety of ways. R packages are most commonly used for sharing functions but they are also useful for data distribution.

Division Data Problems

The Inland Fisheries Division (IFD) has many relatively small (by R standards) data sets that are useful outside of a single district but are often difficult to find because they are scattered throughout the Division and Agency. In addition, the data sets are typically difficult to use because they are not arranged neatly in a single spreadsheet and/or are very messy. For examples:

- Some data is available through PAWS but the exported data is very messy and requires arduous clean-up before it is usable.
  - MTSL (2011–present) - years are separate spreadsheets and difficult to piece together, waterbody names are not standardized.
  - WWSL (2011–present) - years are separate spreadsheets and difficult to piece together, waterbody names are not standardized.
  - NCARP - query-able only by species groups, freedom units only.

  - 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.
  - Warmwater Stocking Trips - no county names, no waterbody names, freedom units, some bad waterbody and county codes, water temps in mixed units.
- Data from on-going fisheries research projects with no home on PAWS.
  - Black Bass Genetics - genetic results are spread across many spreadsheet files with inconsistant formatting.
  - Wild Trout Distribution
  - Wild Trout Barriers
- Other data sets have no distribution.
  - MTSL (2001–2010) - spreadsheets on Deaton's computer.
  - WWSL (2005–2010) - spreadsheets on Deaton's computer.
  - Waterbody Codes - full collection is only available in print (Fish 1968).
  - Coldwater Stocking Coordinates - spreadsheet on Scott Marsh's computer.
  - County Codes - PDF scan of an old piece of paper on Scott Marsh's computer.
  - Missing Reports - spreadsheet on Powell's computer.
  - Raleigh Library Contents - spreadsheet on Shauna Glover's computer.
  - Fish Species Codes - no definitive source. PDFs are found throughout the agency and some of the codes are used in PAWS. There are some inconsistencies among sources and not all fish species have a code.

Introduction to {NCIFD}

{NCIFD} is an R package for the Inland Fisheries Division of the North Carolina Wildlife Resources Commission (NCWRC). The main goal of {NCIFD} is to increase the availability and usefulness of Agency and Division data among IFD staff, but it stores some functions also. The first non-beta version 1.0.0 'Aggressive Alosid' was released on 2022-02-08 and a 'patch' version 1.0.1 was released on 2022-03-11. Releases with new and updated functions and data sets are expected three times per year at natural breaks in data collection:

1. February (after the MTLS and WWSL are finalized)
2. July (after spring stockings)
3. November (after fall stockings)

Install

{NCIFD} is only for Division use and distributed from a MS Teams group for IFD R-users. E-mail Powell Wheeler if you want access to the shared folder. There are five files available after each release:

1. NCIFD_source.zip contains the source code.
2. IntroToNCIFD.pdf is a general overview of the package.
3. NEWS.pdf is the historical change log with 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*}***YY1YVXXSnoXXXX1>~‘
##  -11I!Yli||i>||+|+|++|+|=+++=|++||||||ilIIXos,
##     -^<i|iiiliiiilviii||i|ii|||||+||=|+  ~"!Xmov|,
##         -~~+||i|l||i|||||++++|+=-+||-        ~!1ll=.
##                  --------=++.                    ^"
##                           -
```

```
path.package('NCIFD') #library() allowed R to access the NCIFD package directory.
```

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

Getting Help

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

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

Demonstration: package-wide help

- Hyperlinks to internet resources.
- Internal links to detailed information on package functions and data sets.

{NCIFD} Data Sets

Although the data sets in the package are often available elsewhere in the Agency, the versions in the package are substantially improved and are instantly available in an R session. The data sets are updated when new versions of the package are released. It only takes half a worker-day to update all the data, re-build the package, and release a new version. The data sets are all R data.frames which are 2-dimensional and have rows and columns like a spreadsheet.

- Admin
  - accountCodes; NCWRC-simplified version of the State of NC chart of accounts.
  - staff; publicly-available 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 AFS.
  - missingReports; known NCWRC publications that are not available on PAWS. Some missing reports are being acquired but others are currently lost.
  - raleighLibrary; the > 3,300 items available in the IFD and Wildlife Management Division libraries in the Raleigh Office.

  - reports; contents of the PAWS database of IFD 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 database records.
  - stateRecords; current NC fishing records.

- Hatchery System
  - coldwaterStockingCoords; current trout stocking points.
  - coldwaterStockingTrips; coldwater stocking records since July 1, 1991.
  - warmwaterStockingTrips; warmwater stocking records since 2003 and some older ones back to 1972.
  - wwsl; warmwater fish stocking requests since 2011, plus some older entries back to 2005.
  - mtsl; trout stocking requests beginning in 2001.

- Other
  - townCoords; GPS coordinates of USA towns and cities.
  - zipCodeCoords; USPS ZIP Codes and their approximate GPS coordinates.

Browsing Data in R

Browsing data is difficult in R and can be discomforting for those coming from Excel. Here are several ways to view data:

1. Type 'wwsl + ENTER' in the console window. If you enter a data set name without telling R what to do with it, R will print it to the screen.

```
wwsl #did not run.
```

```
head(wwsl) #only shows top 6 rows. Notice the columns overflow the screen and wrap around.
```

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

2. RStudio's Spreadsheet Interface

```
View(wwsl) #or open from the IDE
```

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

```
write.csv(wwsl, file = '~/Downloads/troutDist.csv', row.names = FALSE)
```

4). {NCIFD} has four functions that help browse data: dfScan(), dfCols(), dfSlim(), and ferret(). See Data.frame Viewing Functions below.

Help for Data Sets

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

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

- Description
- 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 in charge of the data (Jake)
  - Last update
  - Examples - not very useful for data sets

Data Sets are Interconnected

The data sets in {NCIFD} often aggregate information from each other (Figure 1). For example, 'warmaterStockingTrips' contains numeric codes for counties and uses the 'counties' data set to decode them (e.g., 944 = Haywood). 'counties' is also used to assign district numbers to the trips (Haywood = 9). In addition, 'warmaterStockingTrips' uses 'fishes' to decode species codes into species names and 'waterbodies' and 'wwsl' to convert waterbody codes into waterbody names. R's ability to merge data from other data sets is similar to databases and more powerful than lookup functions in Excel.

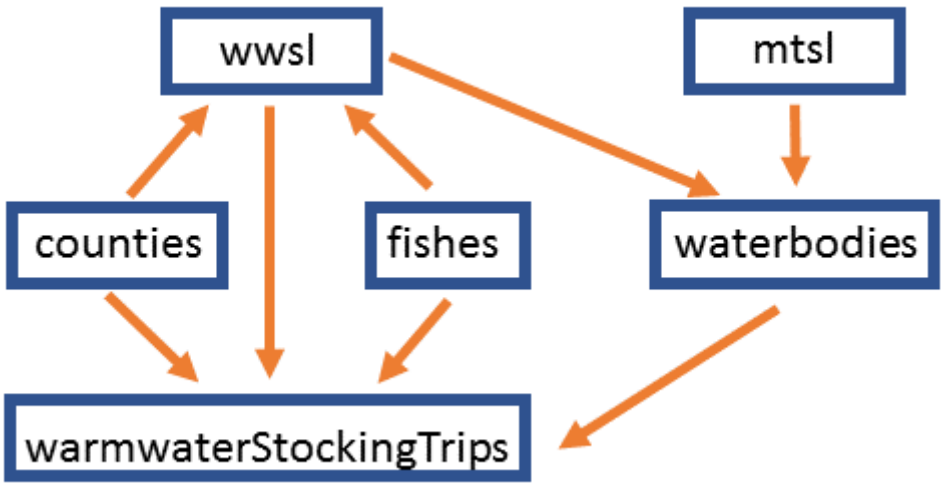

Figure 1: 'warmwaterStockingTrips' pulls information from 'wwsl', 'counties', 'fishes', and 'waterbodies'. In turn, 'wwsl' is built in part from 'fishes' and 'counties', and 'waterbodies' integrates information from 'wwsl' and 'mtsl'.

Source Code for Data Sets

Example: coldwaterStockingTrips components in the source code.

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

Uses of NCIFD Data: Graphs

Demonstration: graphs with coldwaterStockingTrips

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

{NCIFD} Functions

{NCIFD} contains a variety of functions. Some of the functions have specific IFD uses such as cleanBIODE(), relativeWeight(), standardWeight(), others were created to help with research projects, and others help build the package.

- IFD Standardized Sampling Database
  - cleanBIODE(); cleans-up MS Excel files produced when querying the BIODE database in PAWS.
- General 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.
- Other Research Tools
  - flow(); calculates stream flow from interval velocity and depth measurements.
  - ordinalDate(); assigns ordinal dates (1–365).

- Console Tools
  - dfCols(); displays information about the columns in a data.frame.
  - 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 screen.
  - ferret(); finds rows in a data.frame which have a match for one or more search terms.
  - fruitSalad(); creates a data.frame with X rows of randomized information about fictitious salads.
  - reveal(); reveals the libraries and objects that are active in your R session.
- Unit Conversion Functions
  - ac2ha(), c2f(), cubft2cubm(), cubft2gal(), cubm2cubft(), deg2rad(), f2c(),
  - ft2m(), g2lb(), gal2cubft(), ha2ac(), in2mm(), kg2lb(), km2mi(), lb2g(),
  - lb2kg(), m2ft(), mi2km(), mm2in(), and rad2deg().
- Functions for {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 IFD 'notes' and 'comments' columns and decodes > 100 common abbreviations.
  - wordFreq(); finds the frequencies of words data.frame column. It helps check the preformance of stringCleaning().

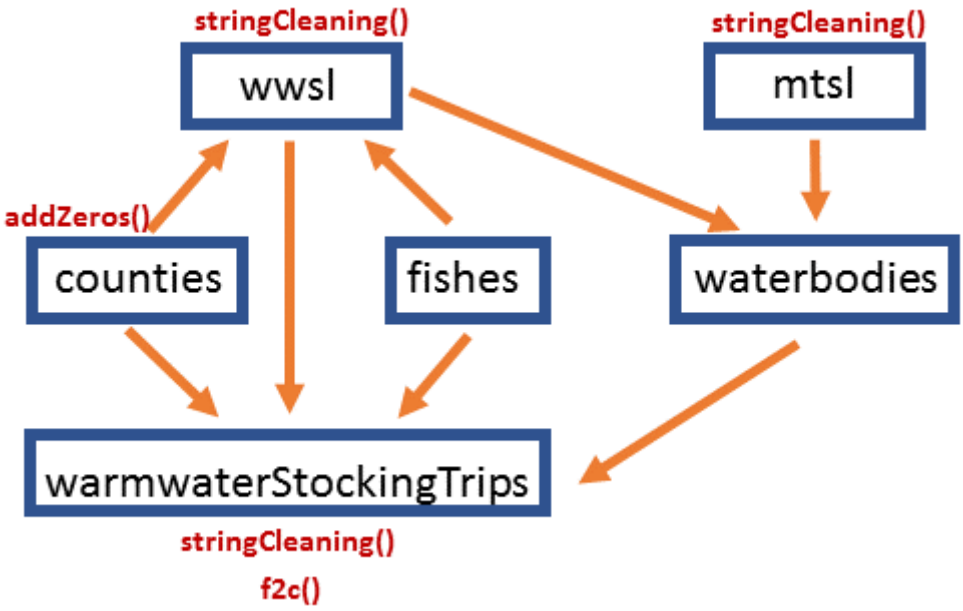

Figure 2: {NCIFD} functions help build the package data sets are shown in red.

Help for Functions

Help information contains all the information that you need to use the function.

```
help(standardWeight) #not run
```

- Description - what the function does
- Arguments - information the function needs
- Value - what the function outputs

- Author - who did the R coding
- Examples - how to use the function
- Other Info - anything we want to pass on to the user
  - Warning
  - Note
  - Fishes with Multiple Standard Weight Equations
  - References

Source Code for Functions

Example: standardWeight() in the source code.

- NCIFD/R/condition.R
  - Function and Help Information (coded in Roxygen2)
- NCIFD/tests/test_that/test_condition.R
  - Automated testing
  - Check against published and known values

Function Examples

cleanBIODE()

Cleans-up EXCEL files exported from PAWS 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 for 47 NC fishes.

- Species equation parameters are in a hidden data.frame: NCIFD:::wsLookup.
  - Some species have multiple equations.
    - Defaults to most useful in NC (BKT = SE BKT) or most general (MKY = both sexes).
- AFS-Format references for standard weight equations are in help(standardWeight).
- Tested against published and known values.
- Alias for less typing: ws().

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

```
## [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

- Skips calculating W_s and finds W_r directly
- Calls standardWeight() internally
- Tested against published and known values
  - NCIFD/tests/testthat/test_condition.R
- Alias for less typing: wr()

```
# make a data.frame with vectors rather than importing a CSV file
spCode <- rep('BKT', 5)
tl_mm  <- c(188, 197, 160, 170, 146)
wt_g   <- c(74, 73, 38, 43, 28)
Ws     <- c(66.5260, 77.2344, 39.9909, 48.3721, 30.0742)
Wr     <- c(111.2346, 94.5175, 95.0217, 88.8942, 93.1030) #DG reported known Wr's

# Build the data.frame - vectors become columns
SE_BKT <- data.frame(spCode = spCode, tl_mm = tl_mm , wt_g = wt_g, Ws = Ws, Wr = Wr)

#calculate Wr with NCIFD::relativeWeight()
SE_BKT$ncifd_Wr <- relativeWeight(SE_BKT$spCode, SE_BKT$tl_mm, SE_BKT$wt_g)
print(SE_BKT)
```

```
##   spCode tl_mm wt_g    Ws     Wr ncifd_Wr
## 1    BKT   188   74 66.53 111.23   111.23
## 2    BKT   197   73 77.23  94.52    94.52
## 3    BKT   160   38 39.99  95.02    95.02
## 4    BKT   170   43 48.37  88.89    88.89
## 5    BKT   146   28 30.07  93.10    93.10
```

Z2A()

Converts an instantaneous mortality rate (Z) to an annual mortality rate (A). Also, converts the standard error and bootstraps confidence intervals with the gamma distribution. Thanks to Kyle Rachels for adding the CIs!

```
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   0.895
##
## $Annual_Mortality
##   A_pct   SE(A) Low95CI  Up95CI
##    49.8     5.0    39.9    59.1
```

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

```
## $Instantaneous_Mortality
##       Z   SE(Z) Low99CI  Up99CI
##   0.690   0.100   0.455   0.976
##
## $Annual_Mortality
##   A_pct   SE(A) Low99CI  Up99CI
##    49.8     5.0    36.6    62.3
```

ordinalDate()

Converts dates into ordinal dates. For example, calling ordinalDate() on "2008-01-01" returns 1 and "2000-01-02" returns 2. This is useful for creating a common Jan–Dec x-axis when plotting multiple years of a time-series (Figure 3). R dates 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
```

Data.frame Viewing Functions

Because viewing data in R can be difficult, the package contains four functions to help.

1. dfCols(); shows a summary of data.frame columns. This mimics how tibbles print in RStudio. The width of the last column is user specified: defaults to 40, but is set to 30 in this case.

```
dfCols(zipCodeCoords, 30)
```

```
##         column      mode      type     class length N_Obs N_NA N_Empty                uniqueExamples
## 1      zipCode character character character  41873 41873    0       0 00501, 00544, 00601, 0060 ...
## 2  zipCodeType character character character  41873 41873    0       0      UNIQUE, STANDARD, PO BOX
## 3         town character character character  41873 41873    0       0 Holtsville, Adjuntas, Agu ...
## 4        state character character character  41873 41873    0       0 NY, PR, VI, MA, RI, NH, M ...
## 5 townAndState character character character  41873 41873    0       0 Holtsville, NY, Adjuntas, ...
## 6          lat   numeric    double   numeric  41873 41873    0       0 40.81, 18.16, 18.38, 18.4 ...
## 7         long   numeric    double   numeric  41873 41873    0       0 -73.04, -66.72, -67.18, - ...
```

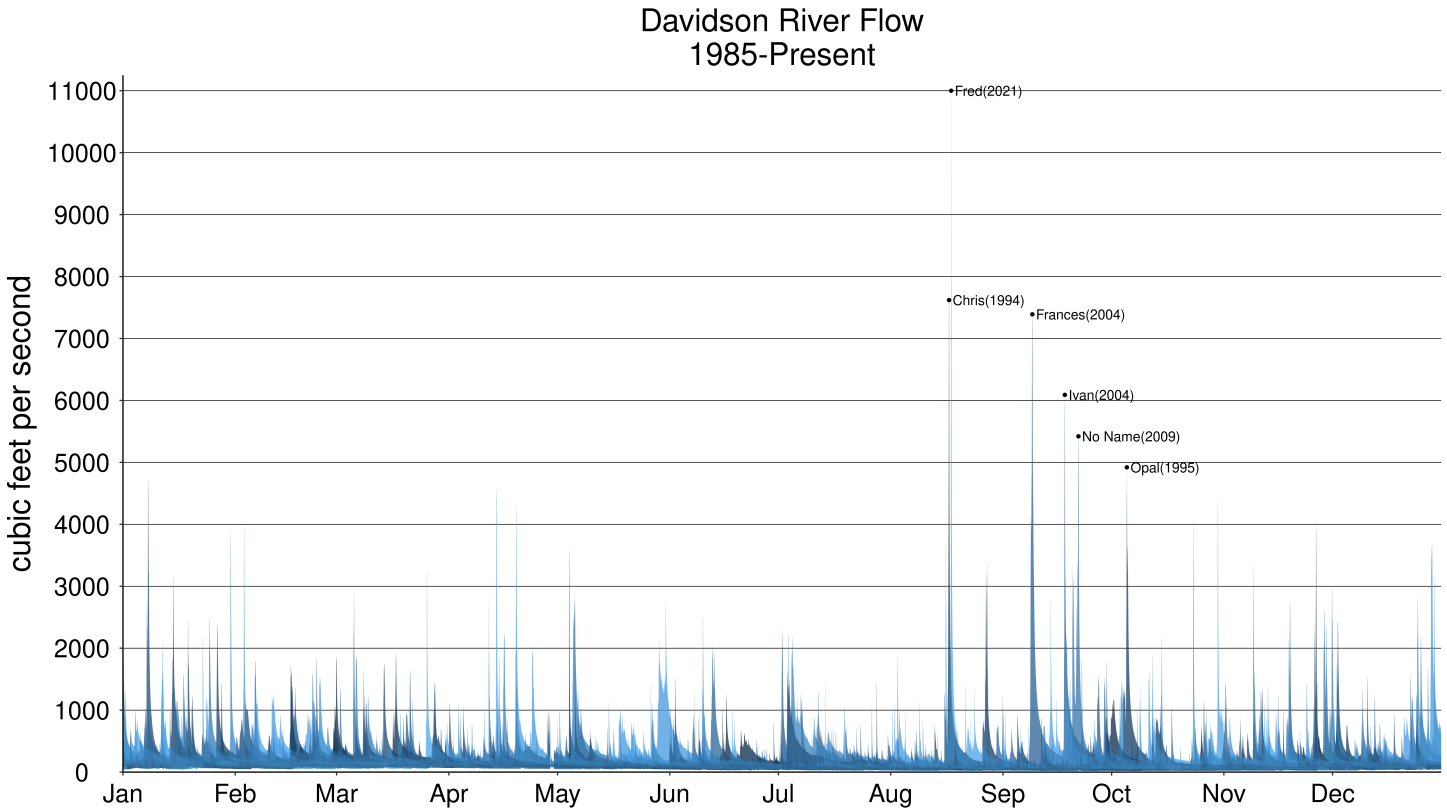

Figure 3: Flows, hurricanes, and tropical storms on the Davidson River, NC. ordinalDate() was used to create a common (1–365) x-axis for each year before the labels (months) were overlain.

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

```
dfScan(afsFishes, 10)
```

```
##                commonName                 genusSp                    ref
## 1            Mud Lancelet  Branchiostoma bennetti Boschung & Gunter 1966
## 431      Zabaleta Anchovy     Anchovia clupeoides          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 do not fit on your screen to prevent wrap-around. This functions works in RStudio, the Linux terminal, and R Markdown documents.

```
dfSlim(head(coldwaterStockingTrips)) #used head() to only print first six rows
```

```
##   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       Tuckaseigee 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
```

4. ferret(); searches a data.frame for strings and returns the rows that have matches. It was designed to quickly find 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 and partial matches. For example, searching for 'WY' will match 'wy', 'wY', and 'Wyoming'. However, if exact=TRUE, 'WY' will only match 'WY'. Here's some examples of how I've used it:

- Is Charles D. Owen Pond still getting winter pond trout stockings?

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

- You jerks only stocked Big Snowbird HS once all last year!

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

- When are Striped Bass typically stocked in Lake Hiwassee?

```
ferret(warmwaterStockingTrips, c('sb', 'hiwassee'))
```

- Find all of David Yow's creel reports

```
ferret(reports, c('Yow', 'creel'))
```

- What months are Bear, Cedar Cliff, Wolf and Tanassee stocked?

```
ferret(mtsl, 'bear|cedar cliff|wolf|tanassee') #if exact=FALSE do an OR search with pipe symbol
```

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

```
ferret(reports, 'Calderwood|Santeetlah|Cheoah|Emory')
```

- When are Walleye normally stocked?

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

Conclusion

R is exposing us, the IFD field staff, to a world of new tools which increase our ability to use and manage our own data. A Division R package is a tool that we can continue to build and develop to improve and share data internally. In addition, we can build functions to accomplish common tasks and share those in the package also. This relatively new technology allows us to handle some problems ourselves that previously required professional IT support. Ultimately, because {NCIFD} is developed by field staff, it may help us better cooperate and collaborate laterally across geographic divisions.

Ready to Learn R?

Start with the book 'R for Data Science' by Wickham and Grolemund. It's suprisingly inexpensive but you can also browse it for free online. 'R4DS' is the absolute gold-standard for getting started with the most modern approach to R programming.