Version 0.6.0 of the [hrbrthemes[](https://git.sr.ht/~hrbrmstr/hrbrthemes)](https://git.sr.ht/~hrbrmstr/hrbrthemes) package should be hitting a CRAN mirror near you soon. Apart from some general documentation and code cleanup this release includes the dark theme folks have been seeing in blog posts and tweets over the past few months. It’s called theme\_ft\_rc() since it is an homage to the wonderful new chart theme developed by the [@ft\_data](http://twitter.com/ft_data) crew over at the Financial Times (you can see examples from their work [here](https://twitter.com/ftdata?lang=en)).

While there was nothing stopping folks from using the GitHub version, the CRAN release makes it more widely available. There are still intermittent issues with fonts for some folks which I’ll be working on for the next release.

Since you’ve already seen lots of examples of these charts I won’t just make a gratuitous example using the theme. I *will*, however, make some charts based on a new data package dubbed [iceout[](https://github.com/hrbrmstr/iceout)](https://github.com/hrbrmstr/iceout). The iceout package was originally conceived by Ben Tupper from the [Bigelow Laboratory for Ocean Sciences](https://www.bigelow.org/). I keep an eye on fellow Mainer repositories and I did not realize (but should have known) that researches keep track of when inland bodies of water freeze and thaw. The package name is derived from the term used for the thaw measurements (“ice-out” or “ice-off”).

Before becoming obsessed with this data and getting the package to the current state it is in, the original codebase worked off of a USGS [Lake Ice-Out Data for New England](https://me.water.usgs.gov/iceout.html) dataset that focused solely on New England and only went up to 2005. Some digging discovered that

* Maine’s Department of Agriculture and Forestry [maintains](https://www.maine.gov/dacf/parks/water_activities/boating/ice_out06.shtml) online records since 2003; and,
* Minnesota’s Department of Natural Resources [maintains](https://www.dnr.state.mn.us/ice_out/index.html?year=1843) a comprehensive database of records going back to the 1800’s.

*But* I hit the jackpot after discovering the U.S. National Snow & Ice Data Center’s [Global Lake and River Ice Phenology](http://nsidc.org/data/lake_river_ice/) dataset which:

*… contains freeze and breakup dates and other ice cover descriptive data for 865 lakes and rivers. Of the 542 water bodies that have records longer than 19 years, 370 are in North America and 172 are in Eurasia; 249 have records longer than 50 years; and 66 longer than 100 years. A few have data prior to 1845. These data, from water bodies distributed around the Northern Hemisphere, allow analysis of broad spatial patterns as well as long-term temporal patterns.*

So, I converted the original package to a data package containing all four of those datasets plus some interactive functions for pulling “live” data and a set of “builders” to regenerate the databases. Let’s take a quick look at what’s in the NSIDC data and the global coverage area:

library(iceout) # github/hrbrmstr/iceout

library(hrbrthemes)

library(ggplot2)

library(dplyr)

data("nsidc\_iceout")

glimpse(nsidc\_iceout)

## Observations: 35,918

## Variables: 37

## $ lakecode "ARAI1", "ARAI1", "ARAI1", "ARAI1", "ARAI1", "ARAI1", "ARAI1…

## $ lakename "Lake Suwa", "Lake Suwa", "Lake Suwa", "Lake Suwa", "Lake Su…

## $ lakeorriver "L", "L", "L", "L", "L", "L", "L", "L", "L", "L", "L", "L", …

## $ season "1443-44", "1444-45", "1445-46", "1446-47", "1447-48", "1448…

## $ iceon\_year 1443, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, …

## $ iceon\_month 12, 11, 12, 12, 11, 12, 12, 12, 12, 11, 12, 12, 12, 12, 12, …

## $ iceon\_day 8, 23, 1, 2, 30, 8, 13, 8, 23, 28, 3, 5, 1, 5, 6, 20, 10, 15…

## $ iceoff\_year NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ iceoff\_month NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ iceoff\_day NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ duration NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ latitude 36.15, 36.15, 36.15, 36.15, 36.15, 36.15, 36.15, 36.15, 36.1…

## $ longitude 138.08, 138.08, 138.08, 138.08, 138.08, 138.08, 138.08, 138.…

## $ country "Japan", "Japan", "Japan", "Japan", "Japan", "Japan", "Japan…

## $ froze TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, …

## $ obs\_comments "calendar correction for ice\_on: -30 days of original data; …

## $ area\_drained 531, 531, 531, 531, 531, 531, 531, 531, 531, 531, 531, 531, …

## $ bow\_comments NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ conductivity\_us NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ elevation 759, 759, 759, 759, 759, 759, 759, 759, 759, 759, 759, 759, …

## $ filename "ARAI", "ARAI", "ARAI", "ARAI", "ARAI", "ARAI", "ARAI", "ARA…

## $ initials "ARAI", "ARAI", "ARAI", "ARAI", "ARAI", "ARAI", "ARAI", "ARA…

## $ inlet\_streams "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", …

## $ landuse\_code "UAFO", "UAFO", "UAFO", "UAFO", "UAFO", "UAFO", "UAFO", "UAF…

## $ largest\_city\_population 52000, 52000, 52000, 52000, 52000, 52000, 52000, 52000, 5200…

## $ max\_depth 7.6, 7.6, 7.6, 7.6, 7.6, 7.6, 7.6, 7.6, 7.6, 7.6, 7.6, 7.6, …

## $ mean\_depth 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, 4.7, …

## $ median\_depth NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ power\_plant\_discharge NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ secchi\_depth NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

## $ shoreline 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, 18, …

## $ surface\_area 12.9, 12.9, 12.9, 12.9, 12.9, 12.9, 12.9, 12.9, 12.9, 12.9, …

## $ state "Nagano Prefecture", "Nagano Prefecture", "Nagano Prefecture…

## $ iceon\_date 1443-12-08, 1444-11-23, 1445-12-01, 1446-12-02, 1447-11-30,…

## $ iceon\_doy 342, 328, 335, 336, 334, 343, 347, 342, 357, 333, 337, 339, …

## $ iceout\_date NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…

## $ iceout\_doy NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …

maps::map("world", ".", exact = FALSE, plot = FALSE, fill = TRUE) %>%

fortify() -> wrld

ggplot() +

ggalt::geom\_cartogram(

data = wrld, map = wrld, aes(long, lat, map\_id=region),

fill="#3B454A", color = "white", size = 0.125

) +

geom\_point(

data = distinct(nsidc\_iceout, lakeorriver, longitude, latitude),

aes(longitude, latitude, fill = lakeorriver),

size = 1.5, color = "#2b2b2b", stroke = 0.125, shape = 21

) +

scale\_fill\_manual(

name = NULL, values = c("L"="#fdbf6f", "R"="#1f78b4"), labels=c("L" = "Lake", "R" = "River")

) +

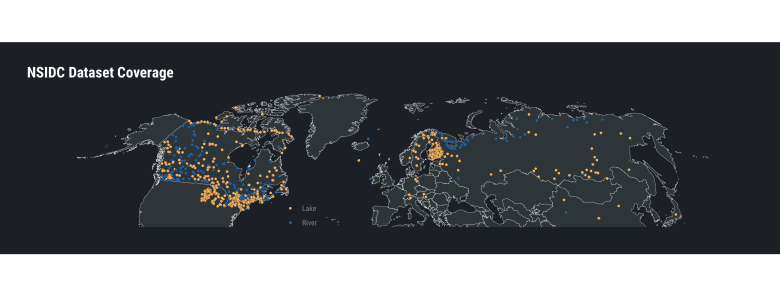
ggalt::coord\_proj("+proj=wintri", ylim = range(nsidc\_iceout$latitude, na.rm = TRUE)) +

labs(title = "NSIDC Dataset Coverage") +

theme\_ft\_rc(grid="") +

theme(legend.position = c(0.375, 0.1)) +

theme(axis.text = element\_blank(), axis.title = element\_blank())

[](https://rud.is/b/2019/01/21/hrbrthemes-0-6-0-on-cran-other-in-development-package-news/nc-idc-coverage/)

W00t! Lots of data (though not all of the extra features are populated for all readings/areas)!

I think the reason the ice-out data garnered my obsession was how it can be used as another indicator that we are indeed in the midst of a climate transformation. Let’s look at the historical ice-out information for Maine inland bodies of water:

filter(nsidc\_iceout, country == "United States", state == "ME") %>%

mutate(iceout\_date = as.Date(format(iceout\_date, "2020-%m-%d"))) %>% # we want the Y axis formatted as month-day so we choose a leap year to ensure we get leap dates (if any)

ggplot(aes(iceoff\_year, iceout\_date)) +

geom\_point(aes(color = lakename), size = 0.5, alpha=1/4) +

geom\_smooth(aes(color = lakename), se=FALSE, method = "loess", size=0.25) +

scale\_y\_date(date\_labels = "%b-%d") +

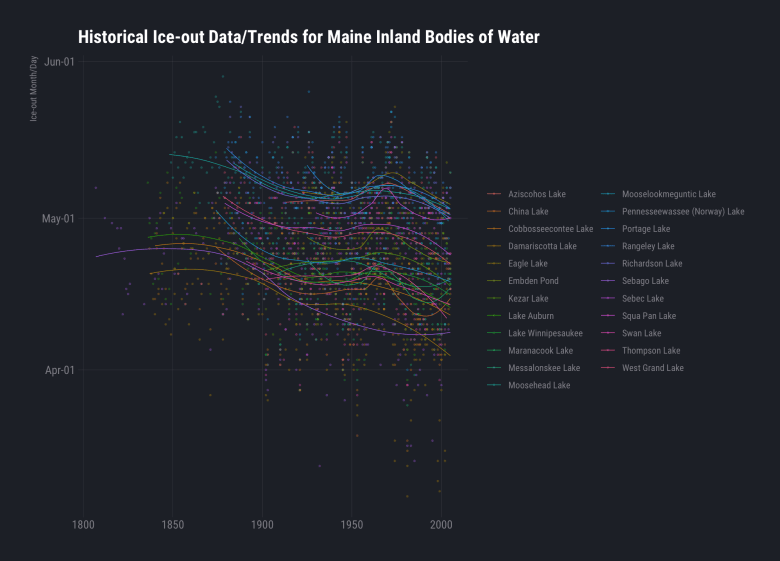
labs(

x = NULL, y = "Ice-out Month/Day", color = NULL,

title = "Historical Ice-out Data/Trends for Maine Inland Bodies of Water"

) +

theme\_ft\_rc(grid="XY")

[](https://rud.is/b/2019/01/21/hrbrthemes-0-6-0-on-cran-other-in-development-package-news/me-inland/)

You can follow that code-pattern to look at other states. It’s also fun to look at the ice-out date distributions by latitude grouping:

filter(nsidc\_iceout, !is.na(latitude) & !is.na(longitude) & !is.na(iceout\_date)) %>%

filter(country == "United States") %>%

mutate(iceout\_date = as.Date(format(iceout\_date, "2020-%m-%d"))) %>%

mutate(lat\_grp = cut(latitude, scales::pretty\_breaks(5)(latitude), ordered\_result = TRUE)) %>%

arrange(desc(iceoff\_year)) %>%

ggplot() +

ggbeeswarm::geom\_quasirandom(

aes(lat\_grp, iceout\_date, fill = iceoff\_year), groupOnX = TRUE,

shape = 21, size =1, color = "white", stroke = 0.125, alpha=1/2

) +

scale\_y\_date(date\_labels = "%b-%d") +

viridis::scale\_fill\_viridis(name = "Year", option = "magma") +

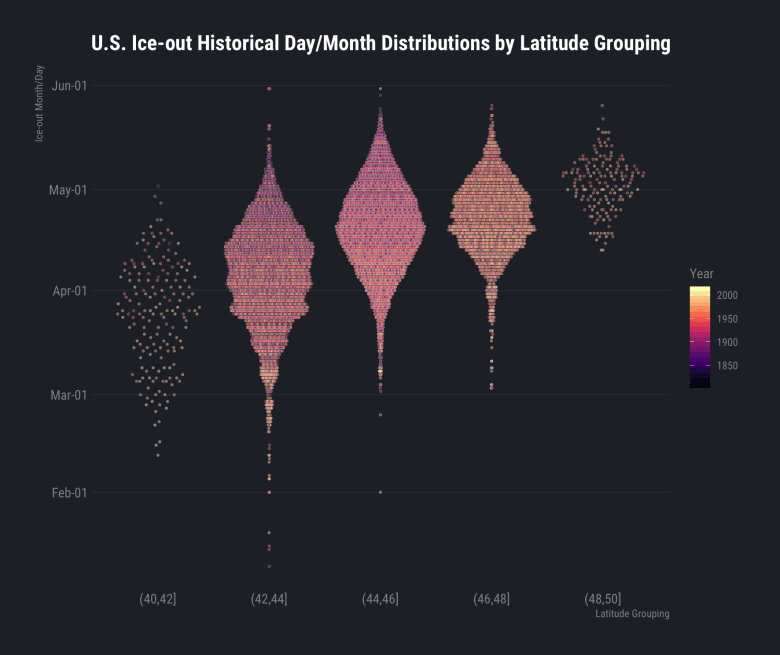
labs(

x = "Latitude Grouping", y = "Ice-out Month/Day",

title = "U.S. Ice-out Historical Day/Month Distributions by Latitude Grouping"

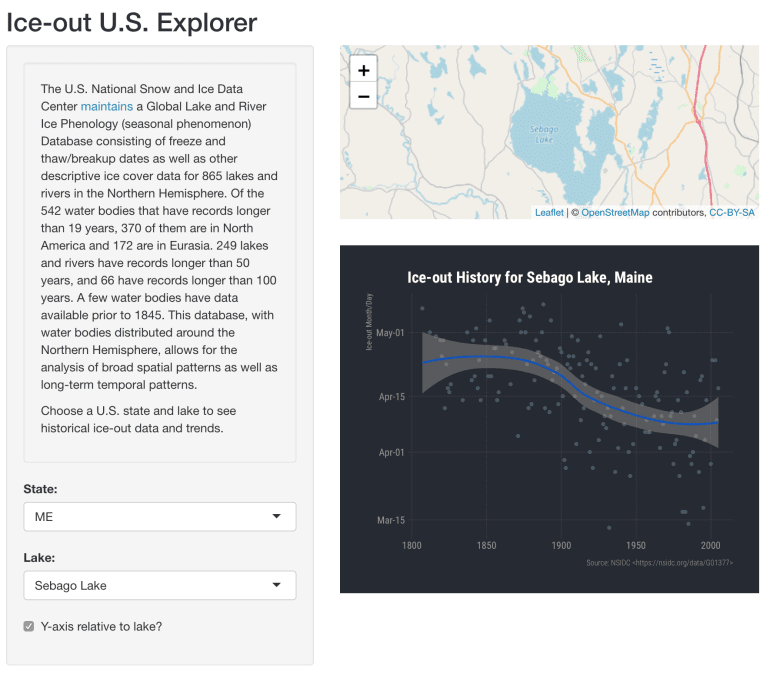
) +

theme\_ft\_rc(grid="Y")

[](https://rud.is/b/2019/01/21/hrbrthemes-0-6-0-on-cran-other-in-development-package-news/lat-grp/)

If you want to focus on individual lakes there’s a Shiny app for that (well one for the U.S. anyway).

After loading the package, just enter explore\_us() at an R console and you’ll see something like this:

[](https://rud.is/b/2019/01/21/hrbrthemes-0-6-0-on-cran-other-in-development-package-news/explore-us-shiny-02/)

The leaflet view will zoom to each new lake selected and the graph will be updated as well.

**Other Package News**

The [sergeant[](https://git.sr.ht/~hrbrmstr/sergeant)](https://git.sr.ht/~hrbrmstr/sergeant) package is reaching a stable point in the 0.8.0 branch (mostly due to David Severski’s tireless help finding bugs ) and should be headed to CRAN soon. Get your issues or PRs in if you want them CRANdied.

I’ve *finally* updated the Java library dependencies in [pdfboxjars[](https://git.sr.ht/~hrbrmstr/pdfboxjars)](https://git.sr.ht/~hrbrmstr/pdfboxjars) so [pdfbox[](https://git.sr.ht/~hrbrmstr/pdfbox)](https://git.sr.ht/~hrbrmstr/pdfbox) will no longer cause GitHub to tell you or I that it is insecure.

There’s a new package dubbed [reapr[](https://git.sr.ht/~hrbrmstr/reapr)](https://git.sr.ht/~hrbrmstr/reapr) that is aimed somewhere at the intersection of curl + httr + rvest. Fundamentally, it provides some coder-uplift when scraping data. The README has examples but here’s what you get on an initial scrape of this blog’s index page:

reapr::reap\_url("http://rud.is/b")

## Title: rud.is | "In God we trust. All others must bring data"

## Original URL: http://rud.is/b

## Final URL: https://rud.is/b/

## Crawl-Date: 2019-01-17 19:51:09

## Status: 200

## Content-Type: text/html; charset=UTF-8

## Size: 50 kB

## IP Address: 104.236.112.222

## Tags: body[1], center[1], form[1], h2[1], head[1], hgroup[1], html[1],

## label[1], noscript[1], section[1], title[1],

## aside[2], nav[2], ul[2], style[5], img[6],

## input[6], article[8], time[8], footer[9], h1[9],

## header[9], p[10], li[19], meta[20], div[31],

## script[40], span[49], link[53], a[94]

## # Comments: 17

## Total Request Time: 2.093s

The reap\_url() function:

* Uses httr::GET() to make web connections and retrieve content which enables it to behave more like an actual (non-javascript-enabled) browser. You can pass anything httr::GET() can handle to ... (e.g. httr::user\_agent()) to have as much granular control over the interaction as possible.
* Returns a richer set of data. After the httr::response object is obtained many tasks are performed including:
  + timestamping of the URL crawl
  + extraction of the asked-for URL and the final URL (in the case  
    of redirects)
  + extraction of the IP address of the target server
  + extraction of both plaintext and parsed (xml\_document) HTML
  + extraction of the plaintext webpage