Introduction to working with Canadian Water Data in R

Using tidyhydat and weathercan

Sam Albers
Digital Platforms and Data Division
Office of the Chief Information Officer
Ministry of Citizens' Services
Province of BC

CWRA Webinar

2019-09-25

Outline

- Who am I?
- Learning Outcomes
- Review R and RStudio and rationale behind using them
- Introduce packages:
  - `dplyr`
  - `tidyhydat`
  - `weathercan`
- Provide an example of using them together
- `tidyhydat` and `weathercan` development
- Where and how to get help in R
- Questions

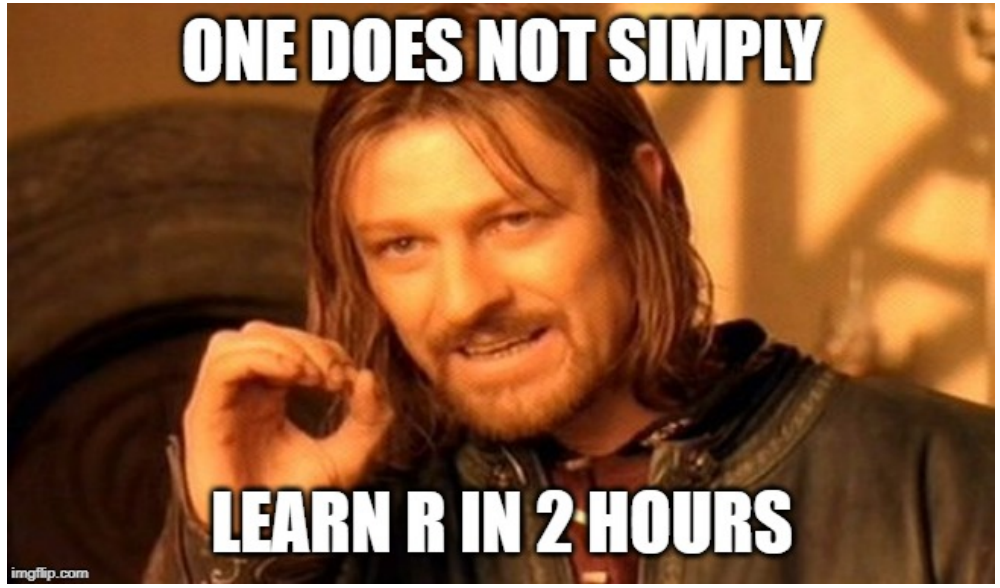
Sam Albers

- Data Scientist with BC government
- Environmental Scientist by training
- Been using R for 10 years
- Maintainer for `tidyhydat`, `rsoi`
- Contributor on many other packages including `weathercan`
- Maintainer of the Hydrology task view

@big_bad_sam
@boshek
sam.albers@gov.bc.ca

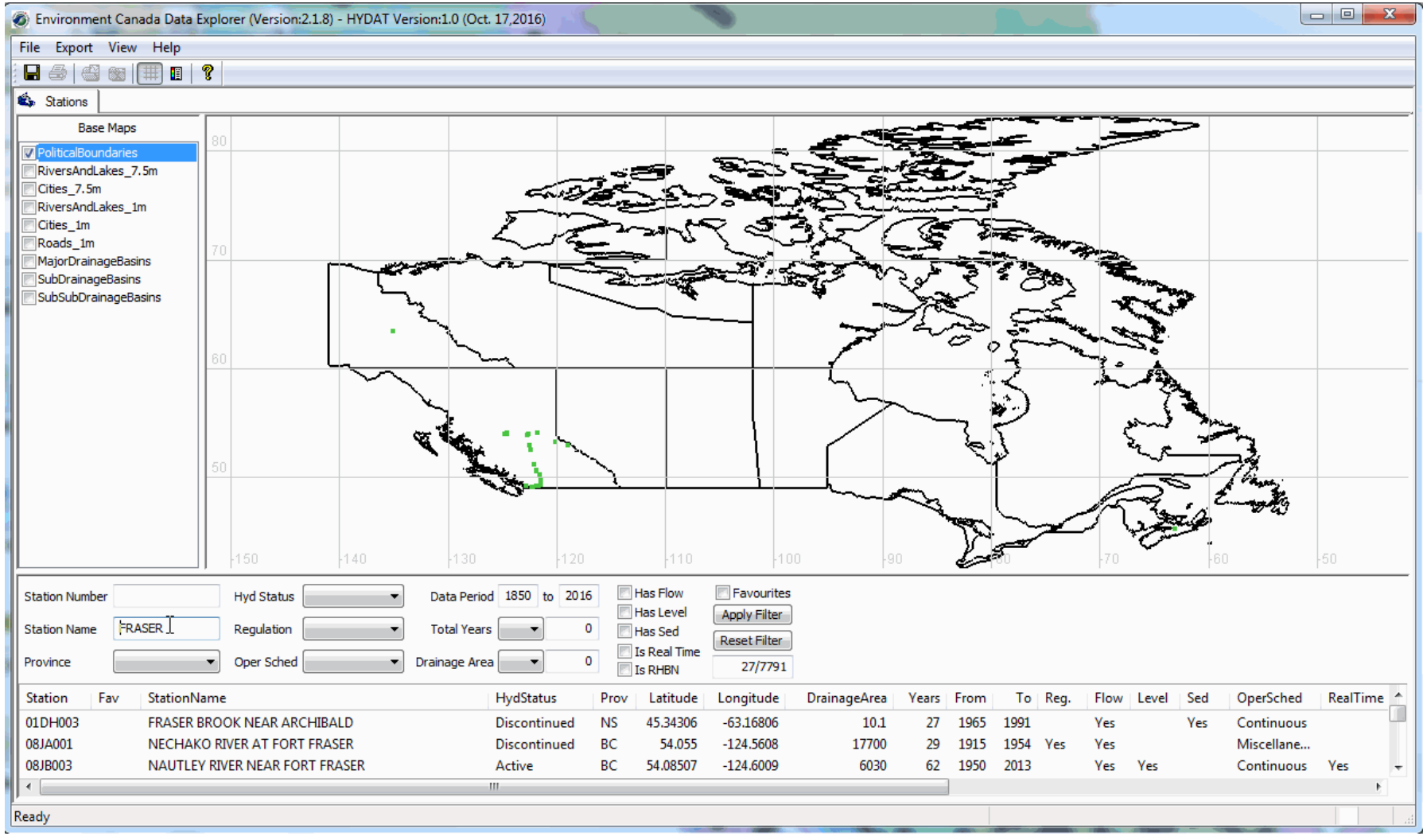
What are we hoping to learn?

- Describe visual elements of RStudio
- Define and assign data to variable
- Manage your workspaces and projects
- Call a function
- Understand the six main `dplyr` verbs
- Overview of tidyhydat and weathercan functions
- Describe usage of `tidyhydat` and `weathercan`
- How to ask for help in R



Common Analysis Problems

Accessing Environment and Climate Change Canada Data



Stakeholder/Manager: "Hey, this is a really cool analysis but we need to add five stations. Can you run it again?"

Make it reproducible!

Questions worth asking...

- Are your methods reproducible?
- What is your analysis recipe?
- Can you share it?

F Serves 4 V

Watershed flow correlations

A simple but elegant analysis.

Ingredients

- 100ml flow data
- 2 cups tidyhydat
- 1 cup Butter
- 4 tbsp data tidying
- 1 tsp correlation
- 3 cups plotted data

Instructions

Preheat the oven to Gas Mark 4, Electric 180°*C, Fan* 160°*C.*

1. Stir flow data in a bowl, add tidyhydat and the butter. When the mixture looks like breadcrumbs, mix in the data tidying. Lay the mixture on a shallow baking tray and bake for 25–30 minutes until golden brown. Leave on the side to cool. Mix together the correlation and plotted data and present analysis.

...Use R!

(or more generally any programmatic code based analysis approach...)

What is R?

- Free and open source
- Statistical programming language
- Publication quality graphics
- Much of the innovation occurs in contributed packages
- But definitely not intimidating...

Some example code

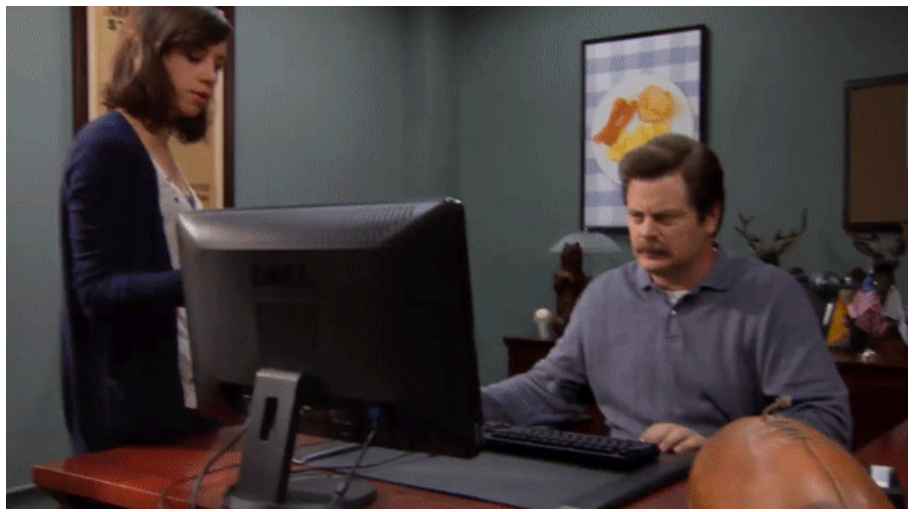
```
all_time_greats <- c(99, 66, 4, 9)
```

- `<-`: assignment operator
- `all_time_greats`: object
- `c`: function

What is RStudio?

- Provides a place to write and run code
- A means to organize projects
- Referred to as an IDE

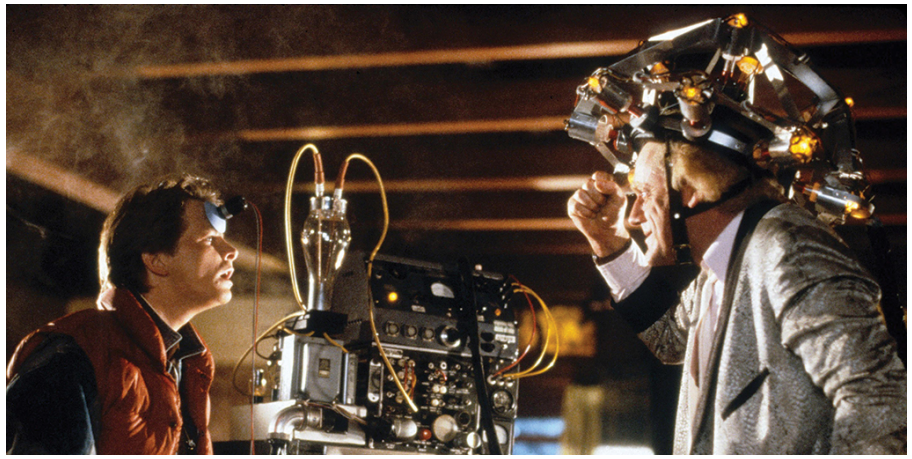
Not guaranteed to help with this...



R and RStudio

The Problem

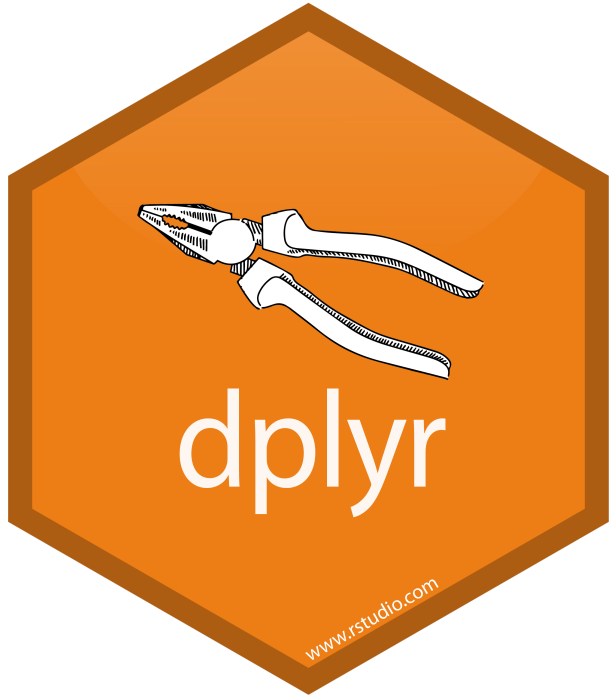
- Many tasks when analyzing environmental data are repetitive yet interactive
- Typically hydrologists/water professionals aren't computer scientists
- Helpful to abstract away unneeded complexity when possible
- A clean and easy to remember syntax reduces your cognitive load when doing analysis



Enter `dplyr`

> a consistent set of verbs that help you solve the most common data manipulation challenges

- Independent of the data source
- Designed for data science



`dplyr` verbs

Functions with English meanings that map directly to the action being taken when that function is called

Installation: `install.packages("dplyr")`

- `%>%` a special symbol to chain operations. Read it as "then"
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.
- `mutate()` adds new variables that are functions of existing variables

For a offline tutorial: http://swcarpentry.github.io/r-novice-gapminder/13-dplyr/index.html

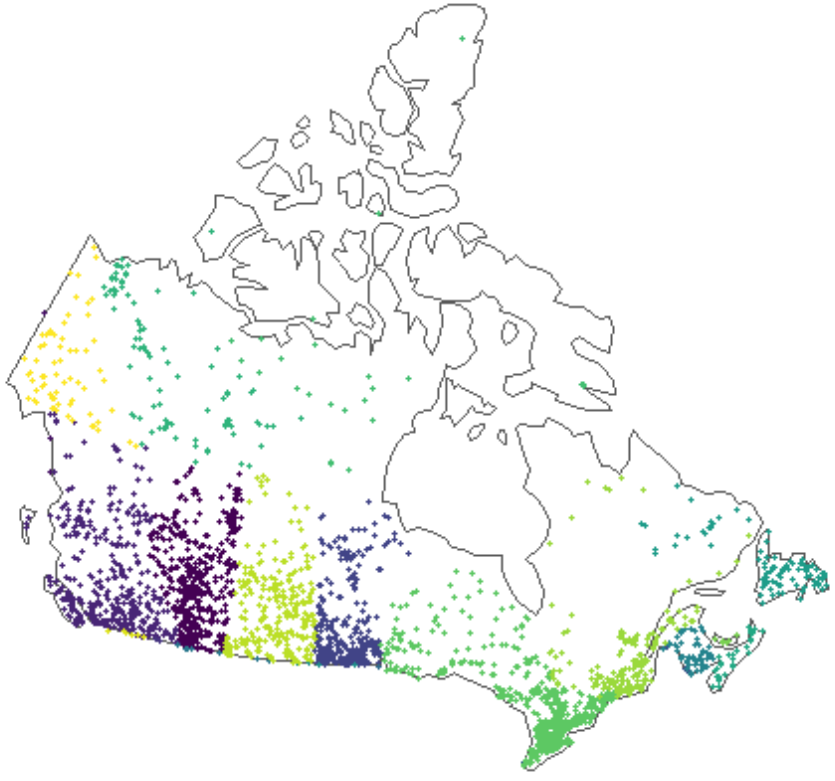
Artwork by @allison_horst

dplyr code break

> The objective of tidyhydat is to provide a standard method of accessing ECCC hydrometric data sources (historical and real time) using a consistent and easy to use interface that employs tidy data principles within the R project.

Installation: `install.packages("tidyhydat")`

hydat::Water Survey of Canada Network



1.1 GB

7842 stations in database

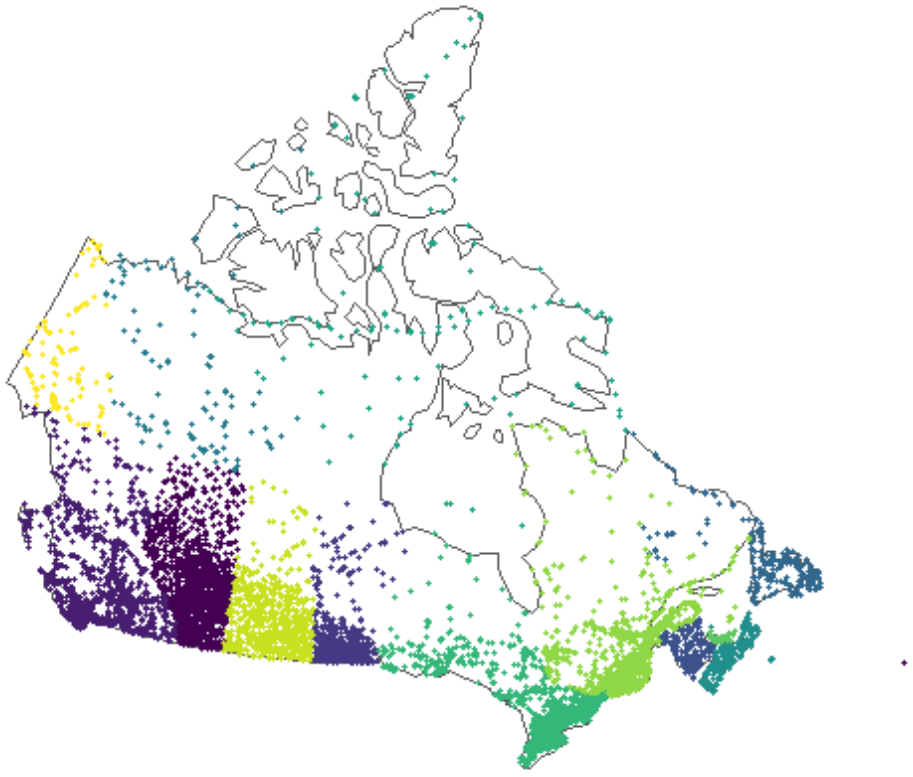
SQLite database

Self contained

> The objective of weathercan is to provide a standard method of accessing ECCC climate data sources using a consistent and easy to use interface that employs tidy data principles within the R project.

Installation: `install.packages("weathercan")`

weathercan::Climate Data



7935 stations

Available online

Looking closer at `tidyhydat` and `weathercan`

tidyhydat

Download the database:

```
download_hydat()
```

Access some flow data

```
flows_data <- hy_daily_flows(station_number = c("08MF005","09CD001","05KJ001","02KF005"))
```

- `<-`: assignment operator
- `flows_data`: object
- `hy_daily_flows`: function
- `station_number`: argument

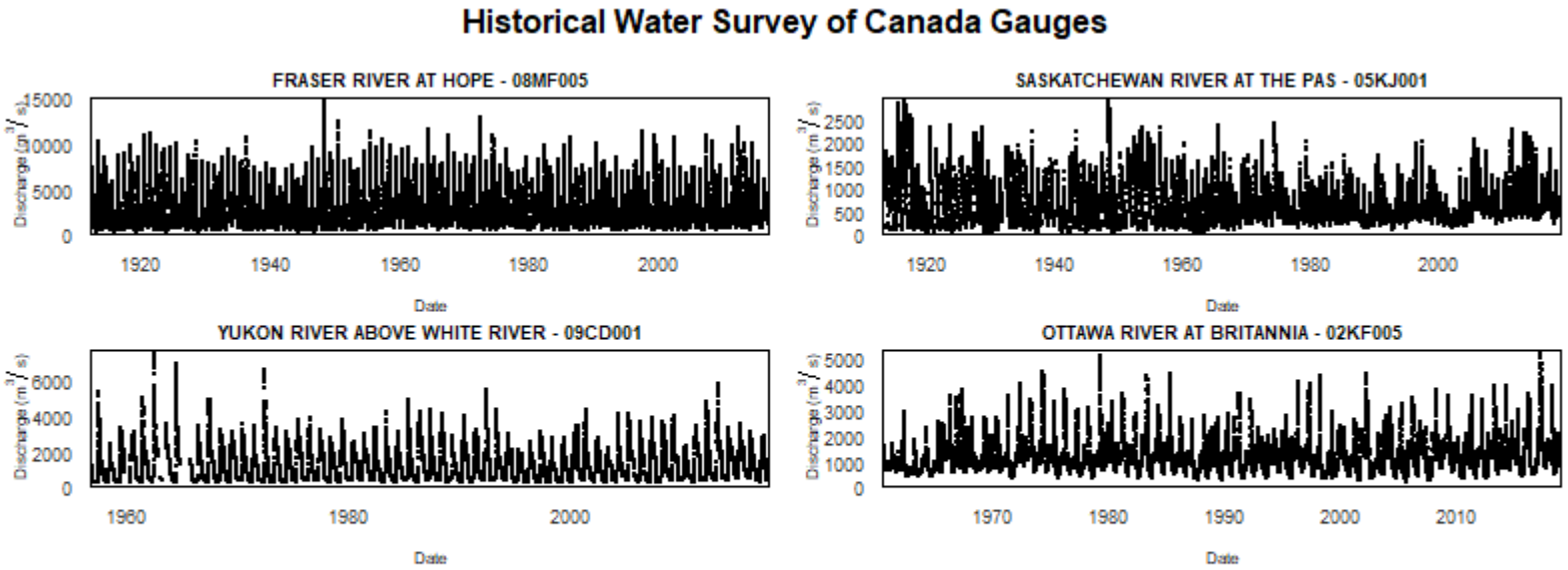
What else is available in `tidyhydat`?

All tables in HYDAT

- See `help(package = "tidyhydat")`
- Realtime data
- Instantaneous peaks
- Daily, monthly and yearly temporal summaries
- Discharge, level, sediment, particle size
- Data ranges
- Station metadata

What else is available in `tidyhydat`?

```
plot(flows_data)
```



What else is available in `tidyhydat`?

```
search_stn_name("fraser")
#> # A tibble: 31 x 5
#>    STATION_NUMBER STATION_NAME                    PROV_TERR_STATE_LOC LATITUDE LONGITUDE
#>    <chr>          <chr>                           <chr>                  <dbl>     <dbl>
#>  1 08JB003        NAUTLEY RIVER NEAR FORT FRASER  BC                      54.1     -125.
#>  2 08KA004        FRASER RIVER AT HANSARD         BC                      54.1     -122.
#>  3 08KA005        FRASER RIVER AT MCBRIDE         BC                      53.3     -120.
#>  4 08KA007        FRASER RIVER AT RED PASS        BC                      53.0     -119.
#>  5 08KB001        FRASER RIVER AT SHELLEY         BC                      54.0     -123.
#>  6 08KE018        FRASER RIVER AT SOUTH FORT GEORGE BC                    53.9     -123.
#>  7 08MC018        FRASER RIVER NEAR MARGUERITE    BC                      52.5     -122.
#>  8 08MD013        FRASER RIVER AT BIG BAR CREEK   BC                      51.2     -122.
#>  9 08MF005        FRASER RIVER AT HOPE            BC                      49.4     -121.
#> 10 08MF040        FRASER RIVER ABOVE TEXAS CREEK  BC                      50.6     -122.
#> # ... with 21 more rows
```

tidyhydat code break

weathercan

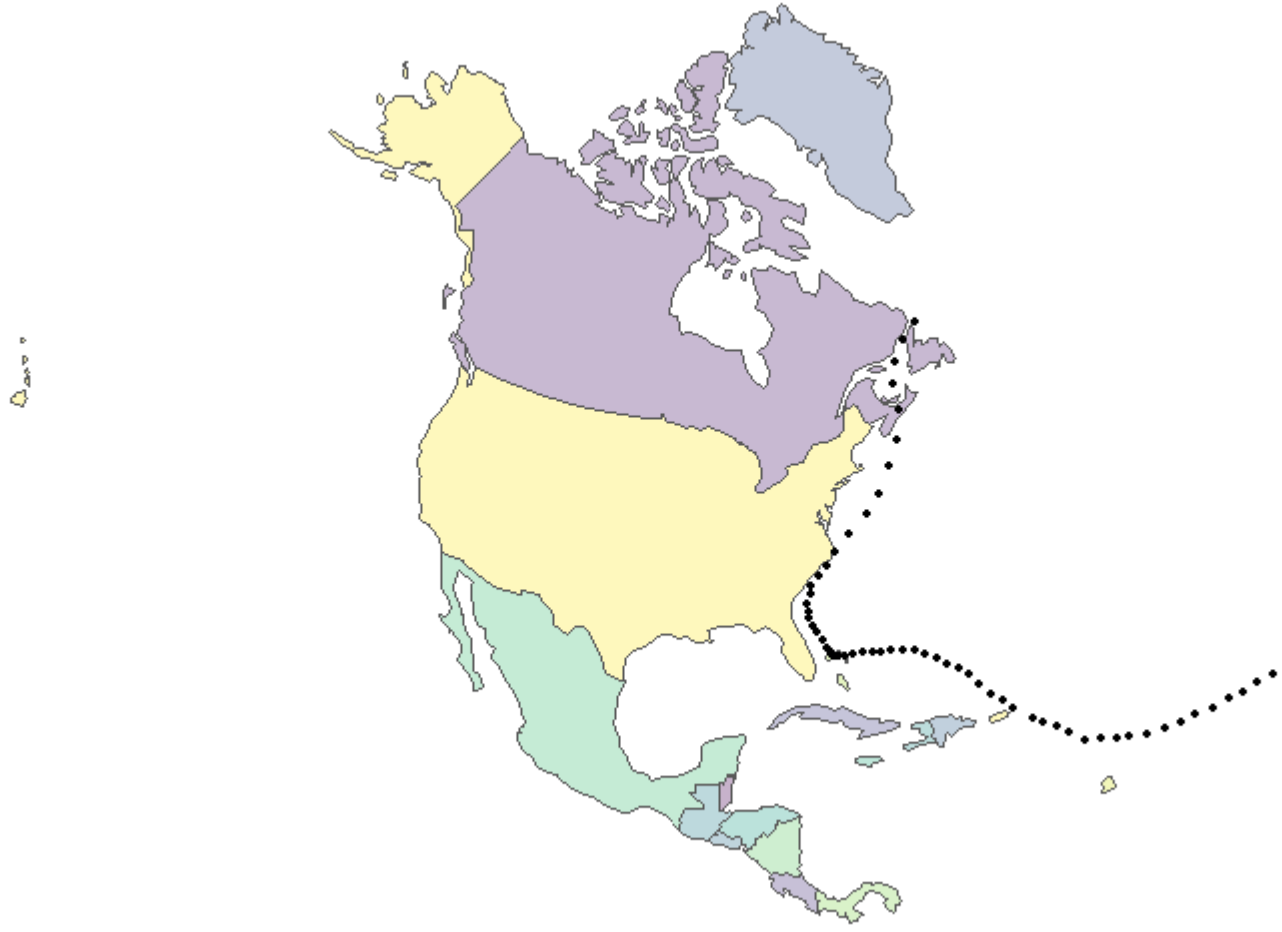
```
vic_gonzales <- weather_dl(station_ids = "114", interval = "day", start = "2019-01-01", end = "2(
vic_gonzales
#> # A tibble: 31 x 37
#>    station_name station_id station_operator prov    lat   lon  elev climate_id WMO_id
#>    <chr>        <chr>      <chr>            <chr> <dbl> <dbl> <dbl> <chr>      <chr>
#>  1 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  2 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  3 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  4 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  5 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  6 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  7 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  8 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#>  9 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#> 10 VICTORIA GO~ 114        Environment and~ BC     48.4 -123.    61 1018611    71200
#> # ... with 21 more rows, and 28 more variables: TC_id <chr>, date <date>, year <chr>,
#> #   month <chr>, day <chr>, qual <chr>, cool_deg_days <dbl>, cool_deg_days_flag <chr>,
#> #   dir_max_gust <dbl>, dir_max_gust_flag <chr>, heat_deg_days <dbl>,
#> #   heat_deg_days_flag <chr>, max_temp <dbl>, max_temp_flag <chr>, mean_temp <dbl>,
#> #   mean_temp_flag <chr>, min_temp <dbl>, min_temp_flag <chr>, snow_grnd <dbl>,
#> #   snow_grnd_flag <chr>, spd_max_gust <dbl>, spd_max_gust_flag <chr>,
#> #   total_precip <dbl>, total_precip_flag <chr>, total_rain <dbl>, total_rain_flag <chr>,
#> #   total_snow <dbl>, total_snow_flag <chr>
```

What else is available in `weathercan`?

- See `help(package = "weathercan")`
- Normals
- Climate normals measurements
- Station metadata

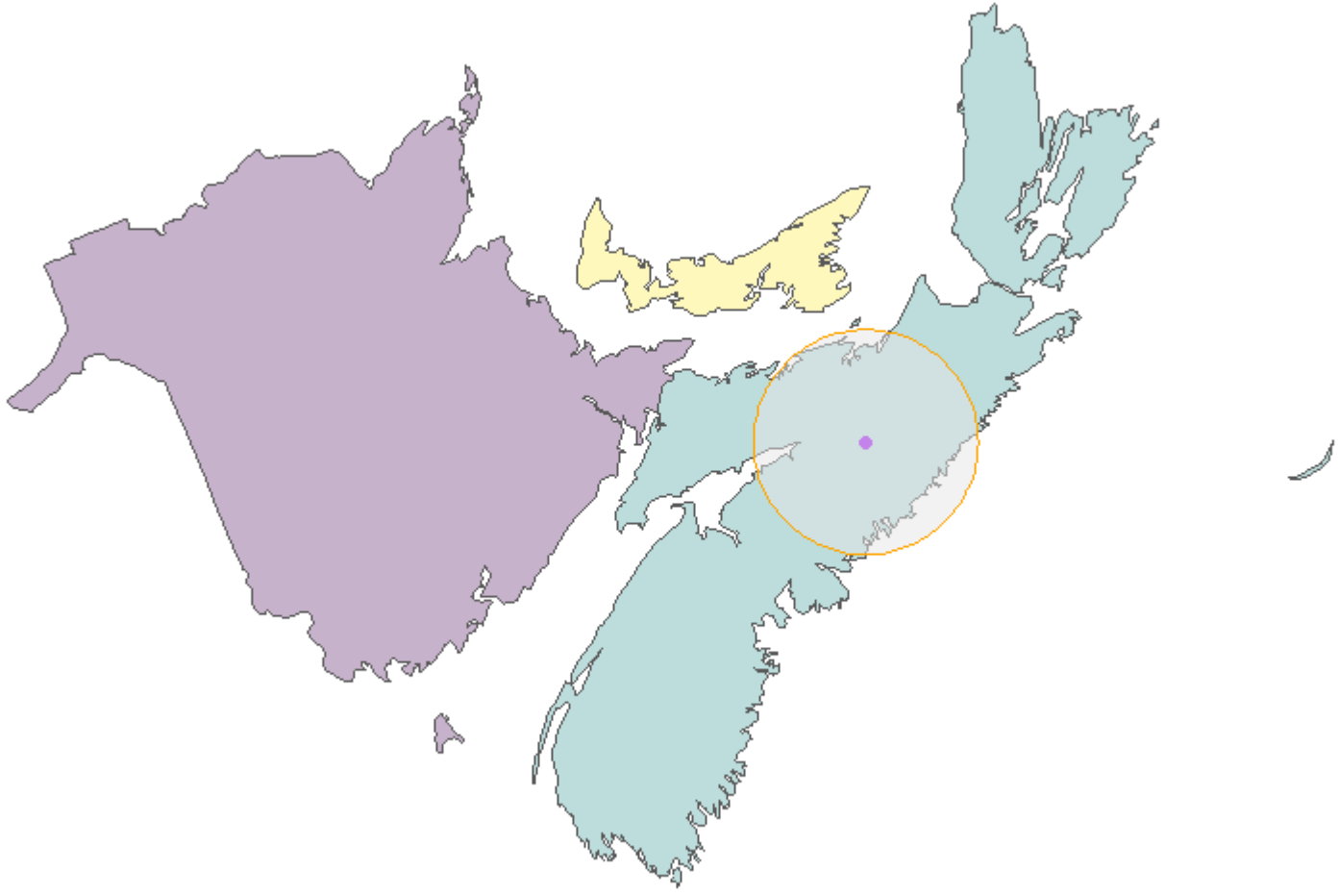
weathercan code break

Path of Hurricane Dorian

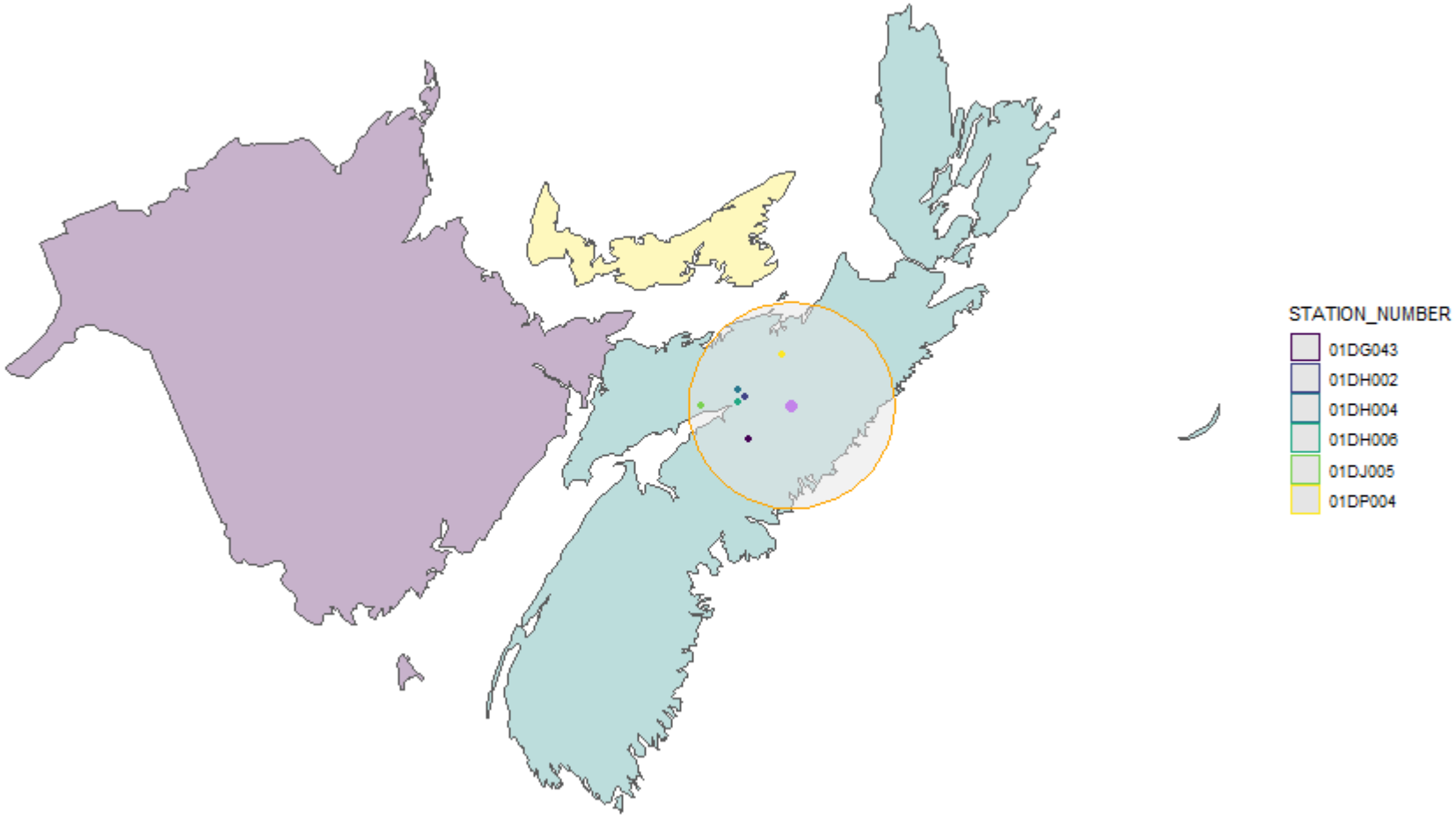


Point where Dorian is over Canadian land

Nova Scotia with buffer



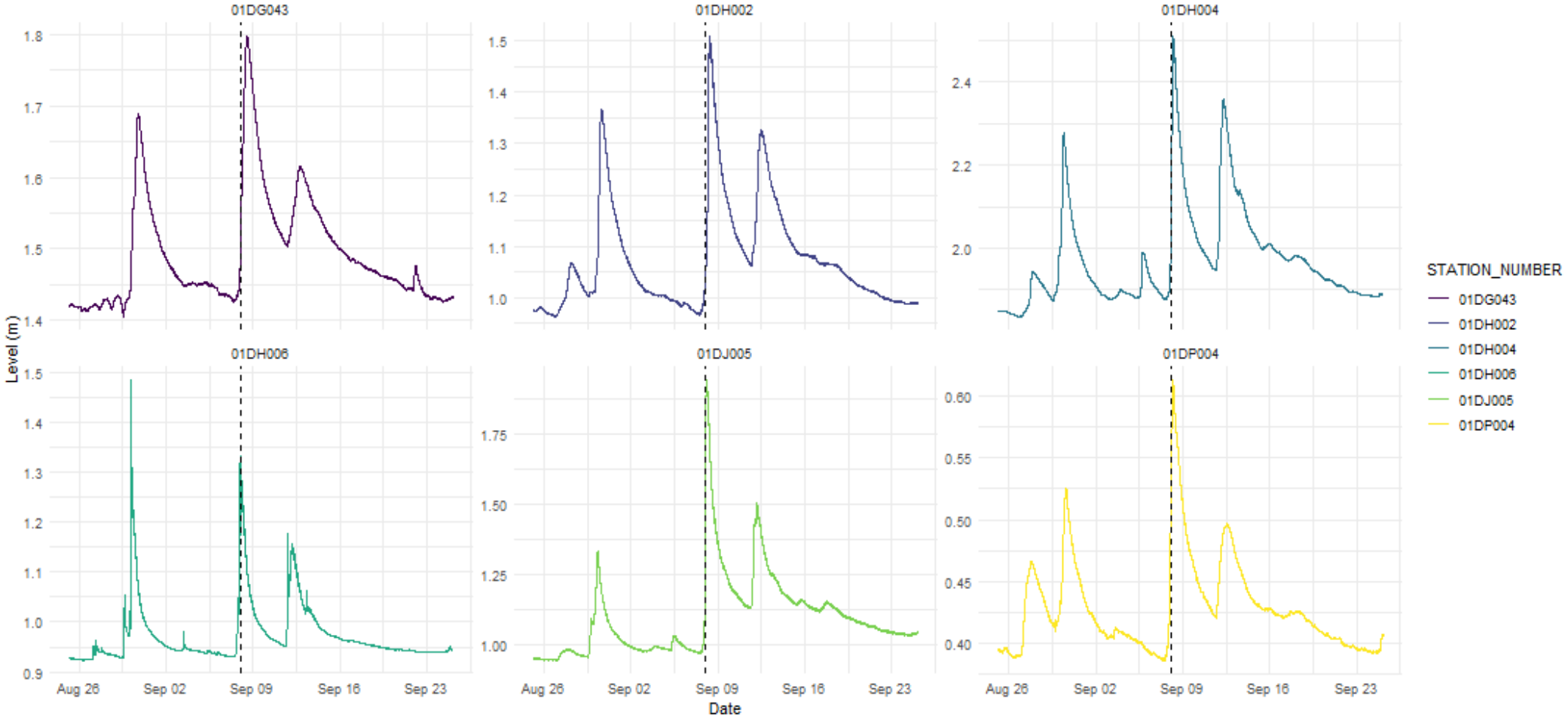
Hydrometric Stations



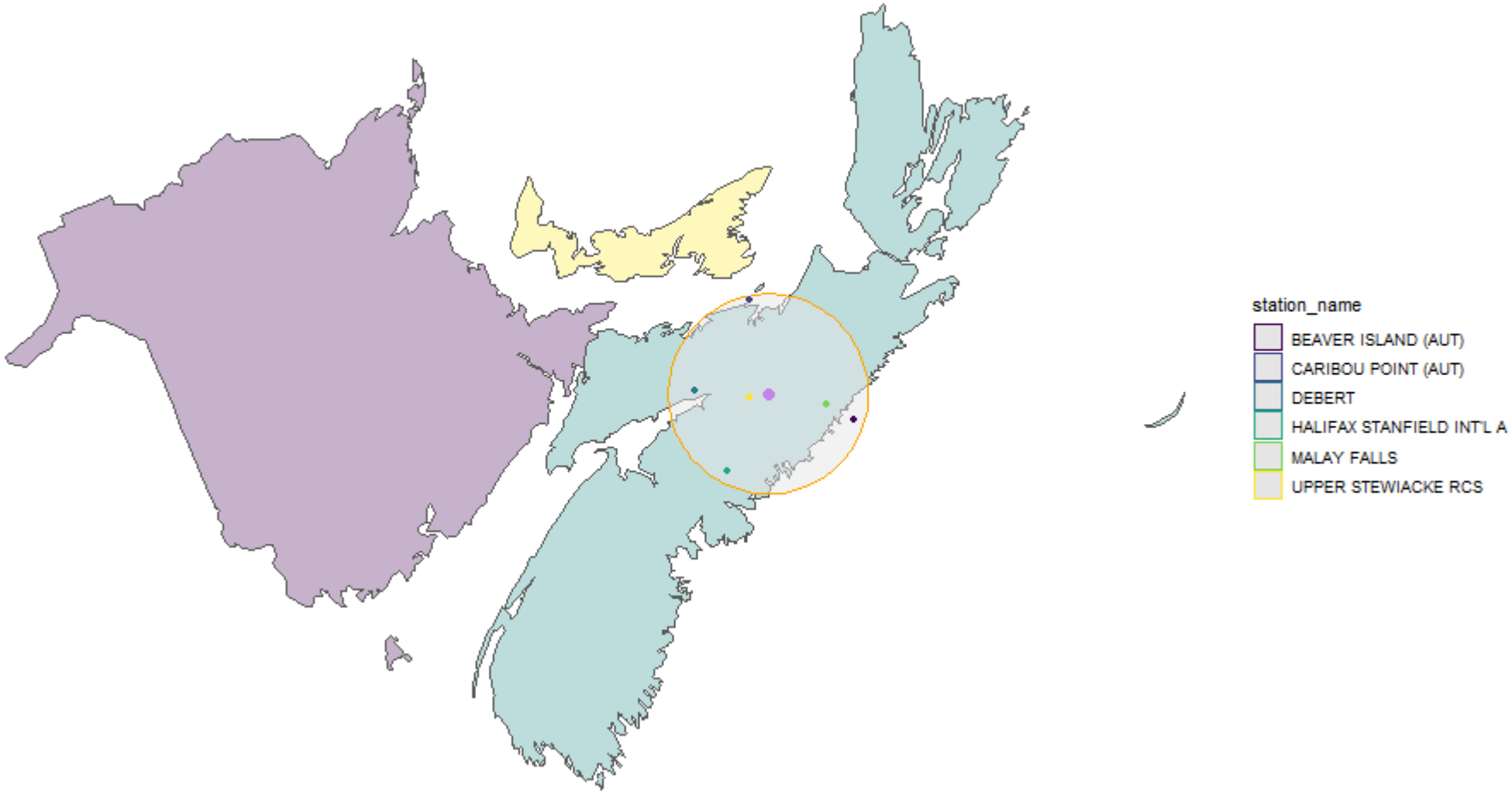
Hydro Data

```
hydro_dorian$STATION_NUMBER
#> [1] "01DG043" "01DH002" "01DH004" "01DH006" "01DJ005" "01DP004"
hydro_data <- realtime_dd(station_number = hydro_dorian$STATION_NUMBER) %>%
  filter(Parameter == "Level")

hydro_data
#>   Queried on: 2019-09-25 04:51:40 (UTC)
#>   Date range: 2019-08-25 to 2019-09-25
#> # A tibble: 52,796 x 8
#>    STATION_NUMBER PROV_TERR_STATE_~ Date                Parameter Value Grade Symbol Code
#>    <chr>          <chr>             <dttm>              <chr>     <dbl> <chr> <chr>  <chr>
#>  1 01DG043        NS                2019-08-25 04:00:00 Level      1.42 <NA>  <NA>   1
#>  2 01DG043        NS                2019-08-25 04:05:00 Level      1.42 <NA>  <NA>   1
#>  3 01DG043        NS                2019-08-25 04:10:00 Level      1.42 <NA>  <NA>   1
#>  4 01DG043        NS                2019-08-25 04:15:00 Level      1.42 <NA>  <NA>   1
#>  5 01DG043        NS                2019-08-25 04:20:00 Level      1.42 <NA>  <NA>   1
#>  6 01DG043        NS                2019-08-25 04:25:00 Level      1.42 <NA>  <NA>   1
#>  7 01DG043        NS                2019-08-25 04:30:00 Level      1.42 <NA>  <NA>   1
#>  8 01DG043        NS                2019-08-25 04:35:00 Level      1.42 <NA>  <NA>   1
#>  9 01DG043        NS                2019-08-25 04:40:00 Level      1.42 <NA>  <NA>   1
#> 10 01DG043        NS                2019-08-25 04:45:00 Level      1.42 <NA>  <NA>   1
#> # ... with 52,786 more rows
```

Hydro Data


01DG043
01DH002
01DH004
01DH006
01DJ005
01DP004
Level (m)
Date
Aug 26
Sep 02
Sep 09
Sep 16
Sep 23
STATION_NUMBER
01DG043
01DH002
01DH004
01DH006
01DJ005
01DP004

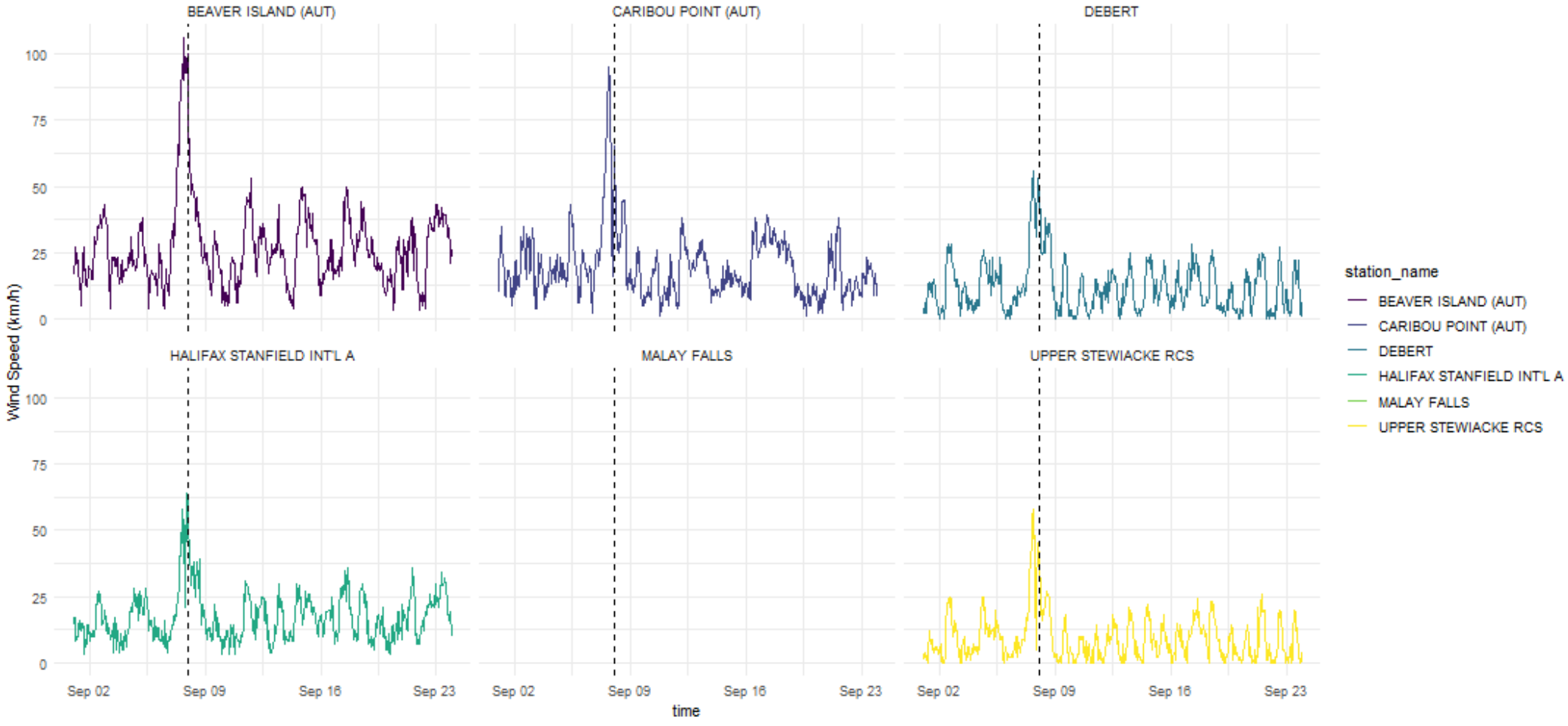
Climate Stations



Climate Data

```
climate_dorian$station_id
#> [1]  8990 10078 30668 42243 44363 50620 53938
climate_data <- weather_dl(station_ids = climate_dorian$station_id,
                           start = "2019-09-01", interval = "hour", quiet = TRUE)

climate_data
#> # A tibble: 3,312 x 35
#>    station_name station_id station_operator prov    lat   lon  elev climate_id WMO_id
#>    <chr>             <int> <chr>            <chr> <dbl> <dbl> <dbl> <chr>      <chr>
#>  1 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  2 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  3 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  4 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  5 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  6 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  7 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  8 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#>  9 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#> 10 CARIBOU POI~       8990 Environment and~ NS     45.8 -62.7   2.4 8200774    71415
#> # ... with 3,302 more rows, and 26 more variables: TC_id <chr>, date <date>, time <dttm>,
#> #   year <chr>, month <chr>, day <chr>, hour <chr>, weather <chr>, hmdx <dbl>,
#> #   hmdx_flag <chr>, pressure <dbl>, pressure_flag <chr>, rel_hum <dbl>,
#> #   rel_hum_flag <chr>, temp <dbl>, temp_dew <dbl>, temp_dew_flag <chr>, temp_flag <chr>,
```

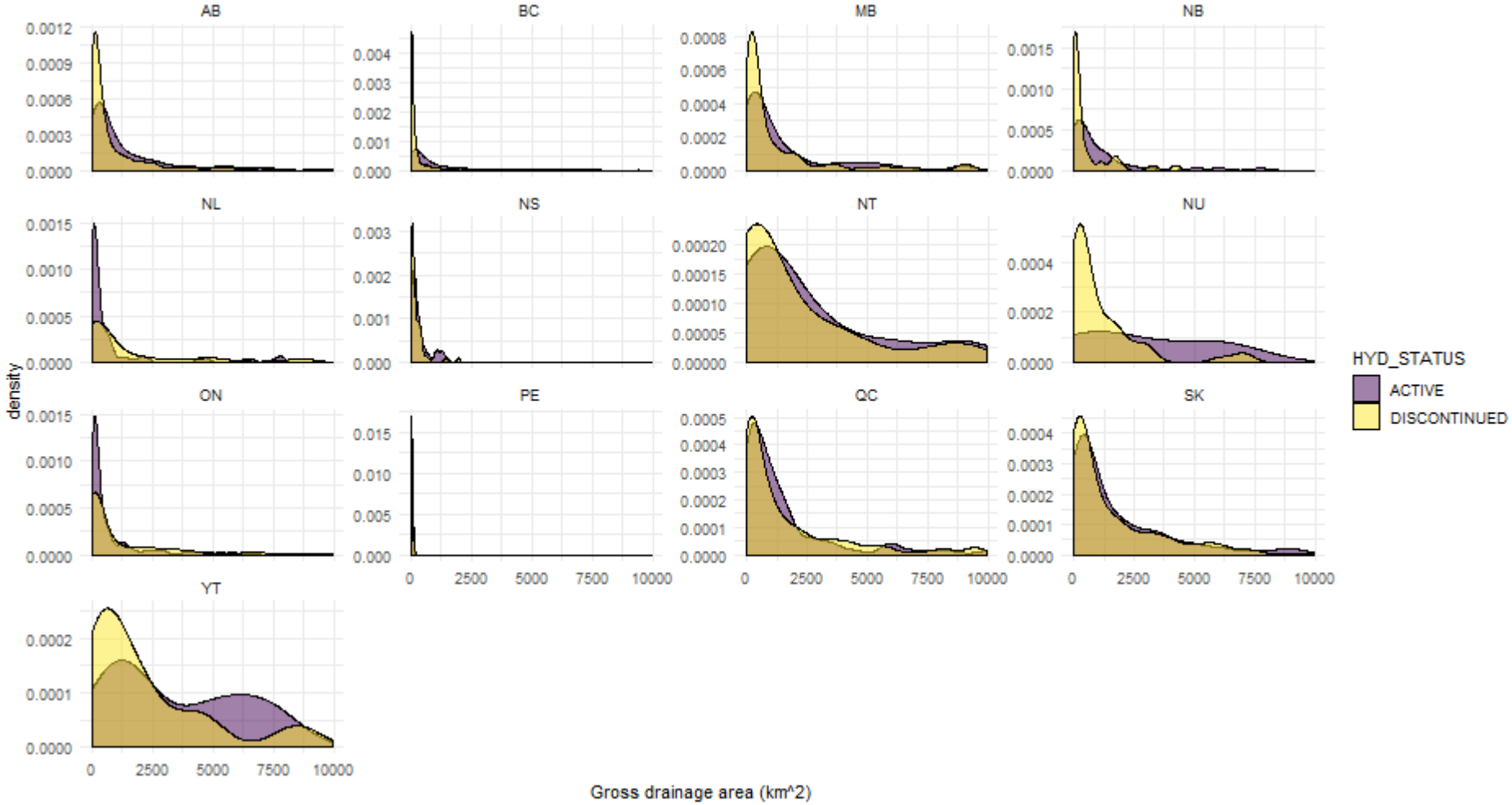
Climate Data



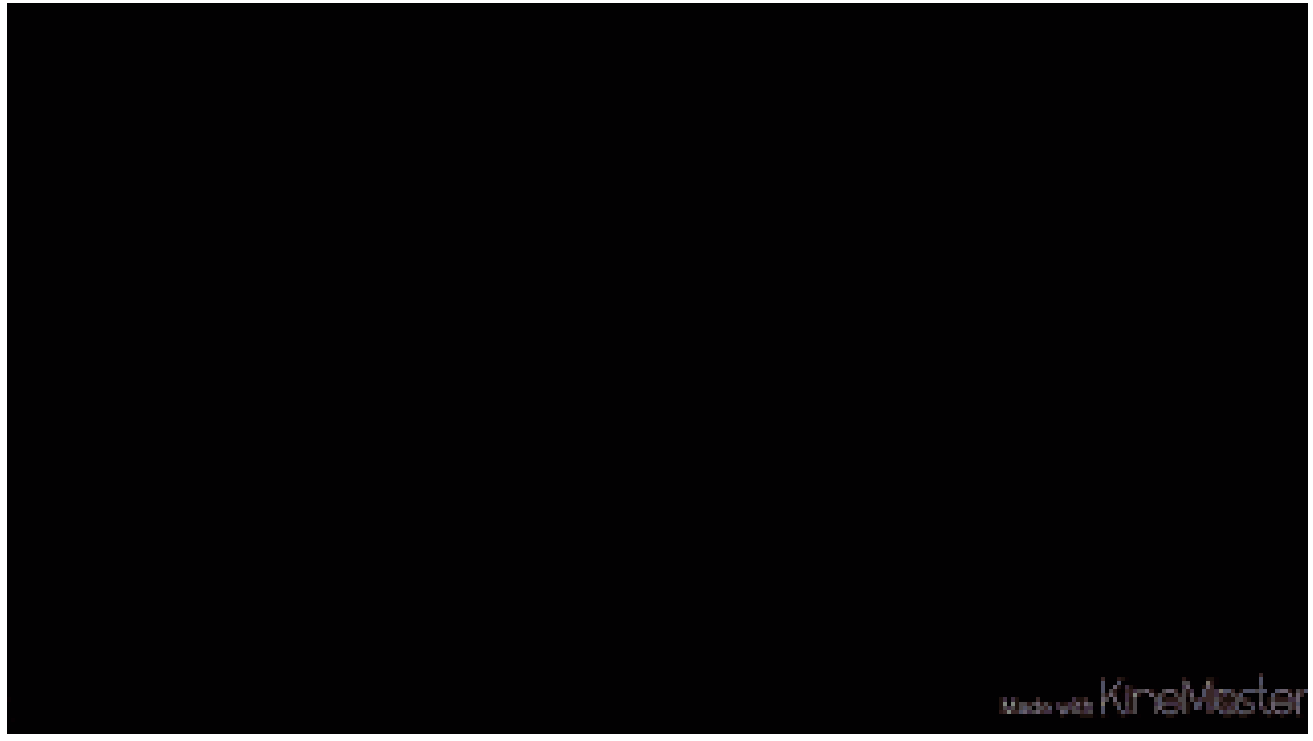
What else is available in R - ggplot2

```
library(ggplot2)

canada_stations <- hy_stations(prov_terr_state_loc = "CA") %>%
  filter(DRAINAGE_AREA_GROSS < 10000)

ggplot(canada_stations, aes(x = DRAINAGE_AREA_GROSS, fill = HYD_STATUS)) +
  geom_density(alpha = 0.5) +
  labs(x = "Mean long term annual discharge (m^3)", y = "Gross drainage area (km^2)") +
  theme_minimal() +
  facet_wrap(~PROV_TERR_STATE_LOC, scales = "free_y")
```

What else is available in R - ggplot2



It can be hard!



Resources for R

Reprex

- Prepare Reproducible Example Code via the Clipboard

Contribute to `tidyhydat` and `weathercan`

Openly developed on GitHub

https://github.com/ropensci/tidyhydat

https://github.com/ropensci/weathercan

Any contribution helps. You don't have to be an R programmer!

- Questions
- Ideas / Feature requests
- Bugs
- Bug-fixes
- Development

Ways to contribute

- Cite as you would with a paper
- Documentation - write a vignette!
- Use the package - find bugs

tidyhydat

- SQL code embedded to efficiently do analysis - leverage the database

weathercan

- Print and plot methods

Ways to cite

📝 Albers S (2017). “tidyhydat: Extract and Tidy Canadian Hydrometric Data.” *The Journal of Open Source Software*, **2**(20). doi: 10.21105/joss.00511, http://dx.doi.org/10.21105/joss.00511.

📝 LaZerte S, Albers S (2018). “weathercan: Download and format weather data from Environment and Climate Change Canada.” *The Journal of Open Source Software*, **3**(22), 571. http://joss.theoj.org/papers/10.21105/joss.00571.

Some Helpful Links

Intro R & RStudio: https://r4ds.had.co.nz

Getting started with `tidyhydat`: https://docs.ropensci.org/tidyhydat

Getting started with `weathercan`: https://ropensci.github.io/weathercan

Hydrology CRAN task view: https://CRAN.R-project.org/view=Hydrology

rOpenSci: https://ropensci.org

 But we all have to work in excel so read this:
https://www.tandfonline.com/doi/full/10.1080/00031305.2017.1375989

Questions?

Slides available from

https://github.com/ropensci/tidyhydat/blob/master/presentations/tidyhydat_weathercan/tidyhydat_weather.pdf
https://github.com/ropensci/tidyhydat/blob/master/presentations/tidyhydat_weathercan/tidyhydat_weather.Rmd

Contact sam.albers@gov.bc.ca