# Hydrological data retrieval in R

Louise Slater
*University of Oxford*

🏠 louisejslater.com

🐦 DrLouiseSlater

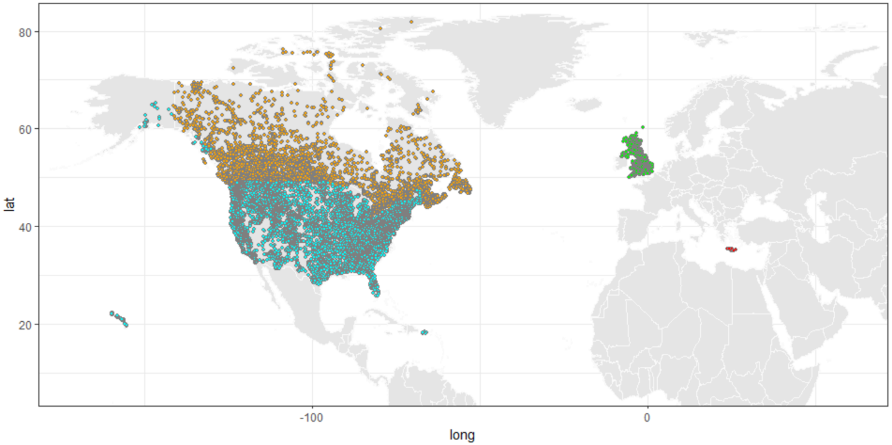
Using R
In
Hydrology

vEGU 2021

# Overview of hydrometric data sources

Streamflow data can be download for several countries using R packages:

- UK -> rnrfa package by Vitolo et al. (2021)
- USA -> dataRetrieval package by DeCicco et al. (2021)
- Canada -> tidyhydat package by Albers et al. (2020)
- Greece -> hydroscoper package by Vantas et al. (2021)



Other data sources we will discuss include the CAMELS datasets (for USA, GB, Australia, Brazil, Chile); the African Database of Hydrometric Indices (ADHI); the Global Runoff Data Centre (GRDC); the Global Streamflow Indices and Metadata Archive (GSIM); and the European floods database (Hall et al. 2015).

# Before starting

# Install and load R packages

Install packages:

```r
install.packages(tidyverse) # for data science functions
install.packages(ggplot2) # for nice plotting
install.packages(dataRetrieval) # USA
install.packages(rnrfa) # UK
install.packages(tidyhydat) # Canada
install.packages(hydroscoper) # Greece
```

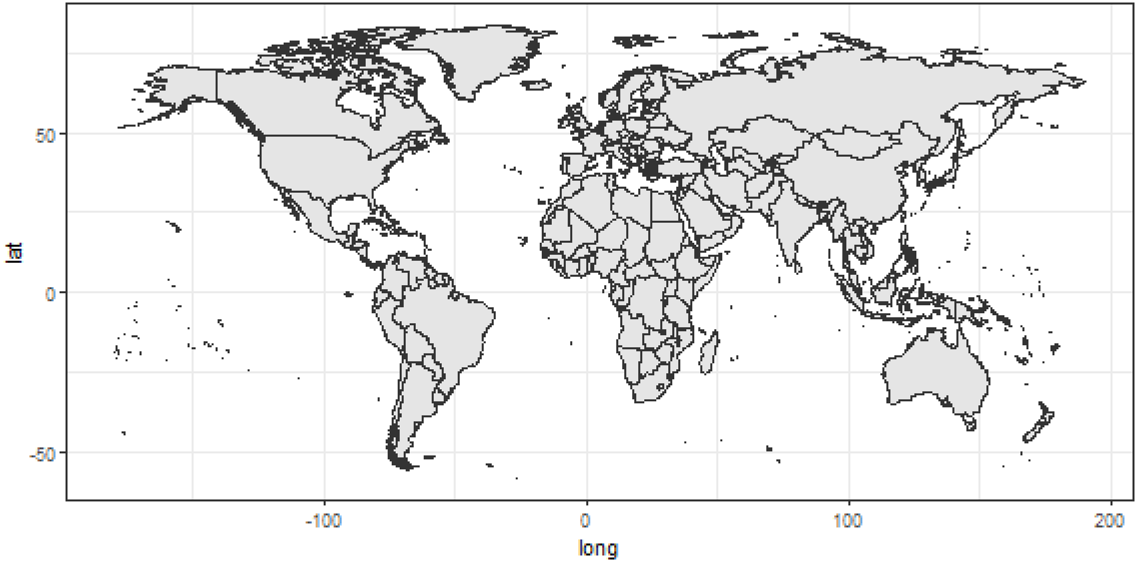
Load them:

```r
library(tidyverse)
library(ggplot2)
library(dataRetrieval)
library(rnrfa)
library(tidyhydat)
library(hydroscoper)
```

# Global borders

To plot the sites, we will need a shapefile of global borders:

```
world <- map_data("world") %>%
  filter(region != "Antarctica")

ggplot()+
  geom_polygon(data = world, aes(long, lat, group = group), size=0.5,
               fill = "grey90", color = "gray20")+
  theme_bw()
```

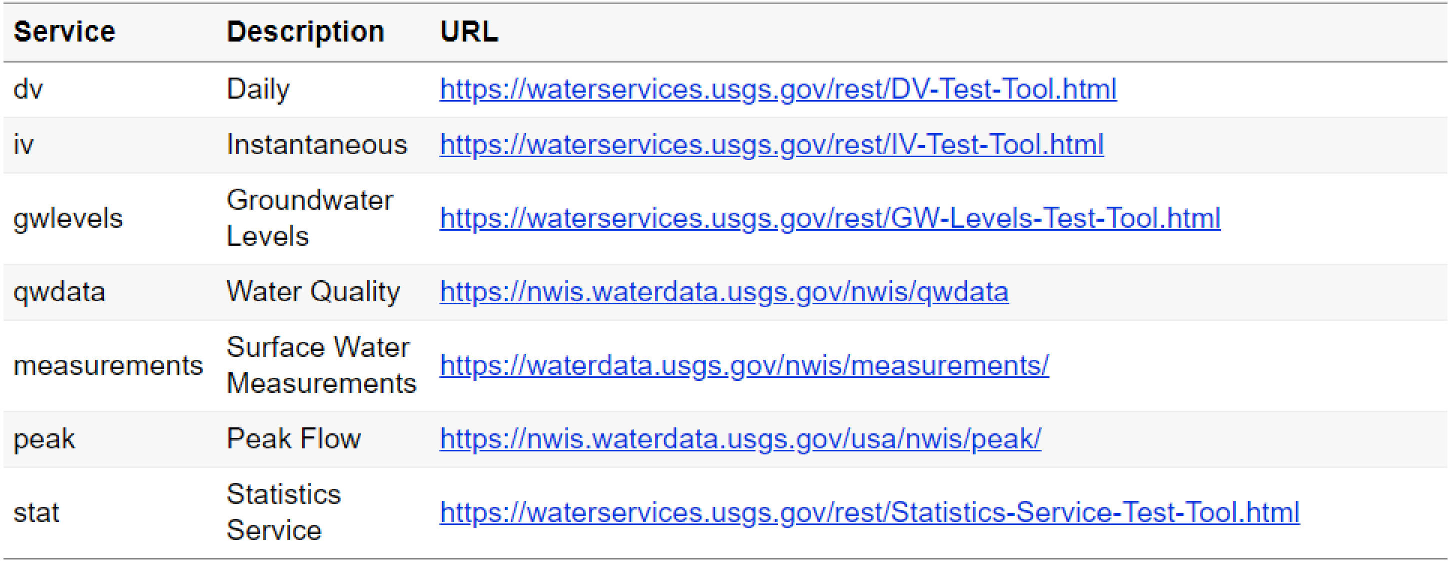


# United States

# United States: dataRetrieval package

We will use the dataRetrieval package by DeCicco et al. (2021). Useful tutorials include Laura DeCicco's slides and blogpost.

What data are available?

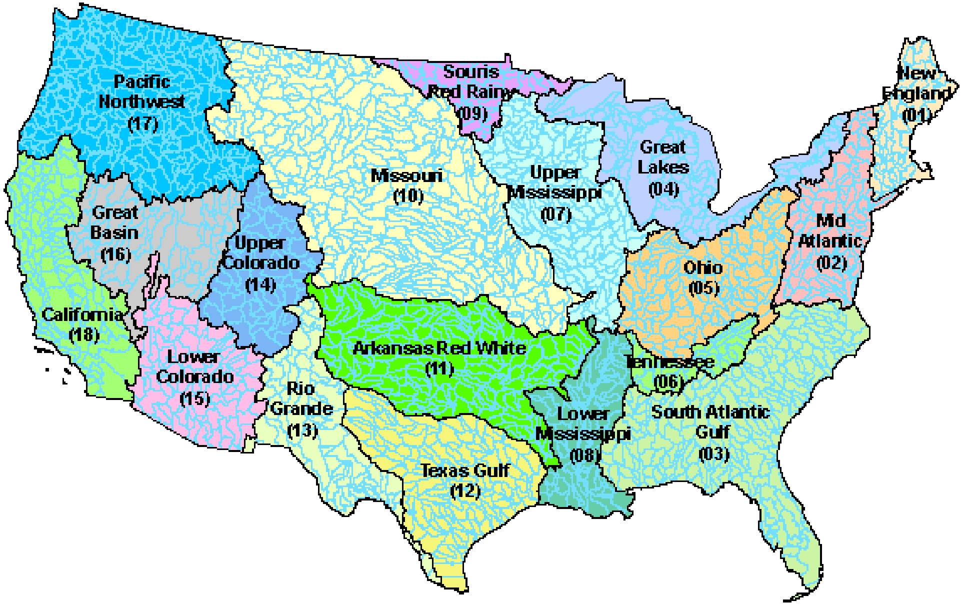


| Service | Description | URL |
|---|---|---|
| dv | Daily | https://waterservices.usgs.gov/rest/DV-Test-Tool.html |
| iv | Instantaneous | https://waterservices.usgs.gov/rest/IV-Test-Tool.html |
| gwlevels | Groundwater Levels | https://waterservices.usgs.gov/rest/GW-Levels-Test-Tool.html |
| qwdata | Water Quality | https://nwis.waterdata.usgs.gov/nwis/qwdata |
| measurements | Surface Water Measurements | https://waterdata.usgs.gov/nwis/measurements/ |
| peak | Peak Flow | https://nwis.waterdata.usgs.gov/usa/nwis/peak/ |
| stat | Statistics Service | https://waterservices.usgs.gov/rest/Statistics-Service-Test-Tool.html |

# United States

Let's assume we want to download streamflow data for the **entire USA**: we first need to identify the **sites** (stream gauges).

Every multiple site query requires a major **filter** (a list of sites, stateCd, huc, bBox, or countyCd). We choose **hydrologic units**:



# United States

We download data for each HUC (01-21), and repeat this for all HUCs to retrieve the whole USA, e.g.:

```r
library(dataRetrieval)
USsites01 <- whatNWISdata(huc="01",parameterCd="00060")
USsites02 <- whatNWISdata(huc="02",parameterCd="00060")
USsites03 <- whatNWISdata(huc="03",parameterCd="00060")
USsites04 <- whatNWISdata(huc="04",parameterCd="00060")
USsites05 <- whatNWISdata(huc="05",parameterCd="00060")
USsites06 <- whatNWISdata(huc="06",parameterCd="00060")
USsites07 <- whatNWISdata(huc="07",parameterCd="00060")
USsites08 <- whatNWISdata(huc="08",parameterCd="00060")
... etc.
```

# United States

Let's make a large database for all the HUCs with all the site-information:

```r
# A long but easy way of binding all HUCs
# (because you need to type out 21 objects):
# USsites <- rbind(USsites01, USsites02....)

# Quicker approach:
hucs <- paste0("USsites",sprintf('%0.2d', 1:21))
USsites <-  `row.names<-`(do.call(rbind,mget(hucs)), NULL)
```

# United States

Check the dataset -- it has 56,991 sites!

```
head(USsites)[1:3]
```

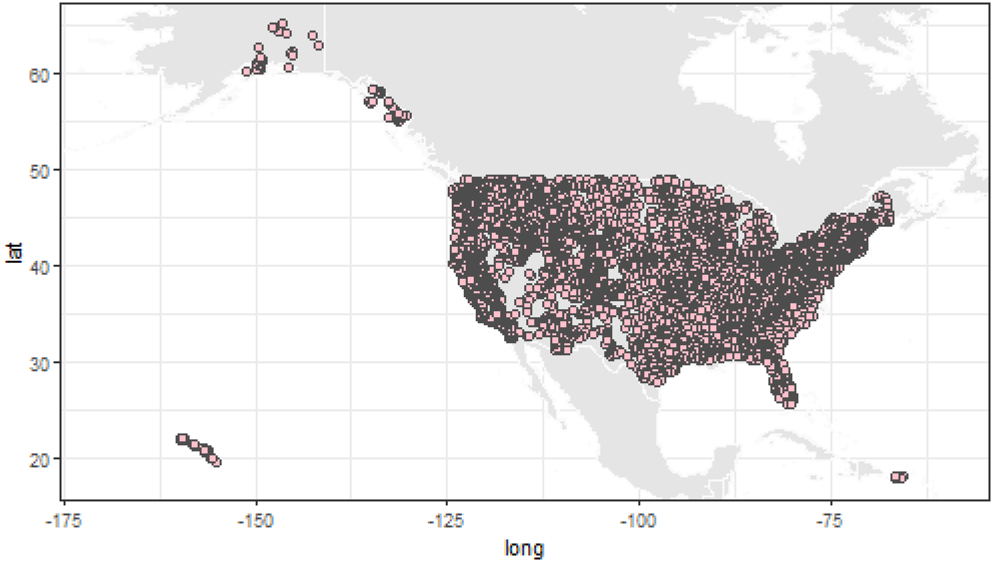
```
##   agency_cd  site_no                                      station_nm
## 1      USGS 01010000        St. John River at Ninemile Bridge, Maine
## 2      USGS 01010000        St. John River at Ninemile Bridge, Maine
## 3      USGS 01010000        St. John River at Ninemile Bridge, Maine
## 4      USGS 01010070           Big Black River near Depot Mtn, Maine
## 5      USGS 01010070           Big Black River near Depot Mtn, Maine
## 6      USGS 01010100 Shields Br Big Black River nr Seven Islands, ME
```

Let's reduce the dataset to 9,057 sites:

```
USsites <- USsites[USsites$begin_date < as.Date("1950-01-01"),]
```

# United States: site location

```
ggplot()+
  geom_polygon(data = world, aes(long, lat, group = group), size=0.5,
               fill = "gray90", color = "gray98") +
  coord_cartesian(xlim=c(-170,-60), ylim=c(18,65))+
  geom_point(data = USsites, aes(x=dec_long_va,y=dec_lat_va),
             fill="pink", col="grey30", size=2, pch=21)+
  theme_bw()
```



# United States: time series

How do we retrieve the actual time series? Let's select just one record from our database: USGS site **05420500**:

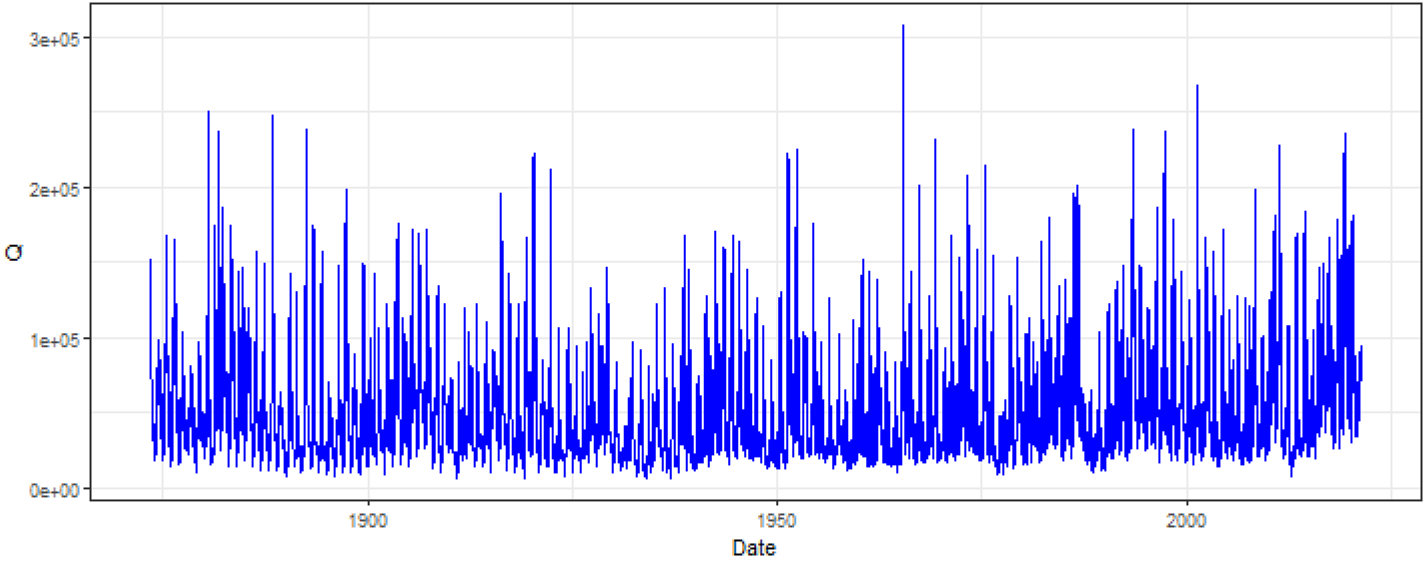
```
dfUS <- dataRetrieval::readNWISdv("05420500","00060","","")
#Rename the streamflow variable:
names(dfUS)[names(dfUS) == 'X_00060_00003'] <- 'Q'
head(dfUS)
```

```
##   agency_cd  site_no       Date      Q X_00060_00003_cd
## 1      USGS 05420500 1873-06-02  88800                A
## 2      USGS 05420500 1873-06-03  88800                A
## 3      USGS 05420500 1873-06-04  92000                A
## 4      USGS 05420500 1873-06-05  96800                A
## 5      USGS 05420500 1873-06-06 102000                A
## 6      USGS 05420500 1873-06-07 109000                A
```

# United States: time series

It's always worth plotting data to check for errors

```
ggplot(dfUS)+
  geom_line(aes(x=Date, y=Q), col="blue")+
  theme_bw()
```



# United Kingdom

# United Kingdom: rnrfa package

We will use the **rnrfa** package by Vitolo et al. (2021). Check out Claudia Vitolo's vignette.

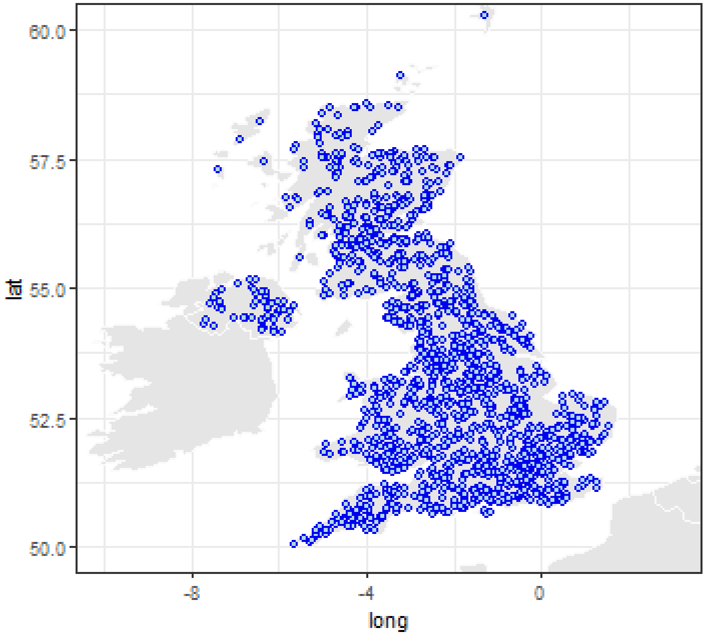
Obtain list of sites:

```r
library(rnrfa)
UKsites <- rnrfa::catalogue()
UKsites <- data.frame(UKsites)
# unique(UKsites$id) # list of sites
head(UKsites)[1:3]
```

```
##     id                    name catchment.area
## 1 1001         Wick at Tarroul          161.9
## 2 2001  Helmsdale at Kilphedir          551.4
## 3 2002    Brora at Bruachrobie          434.4
## 4 3001           Shin at Lairg          494.6
## 5 3002    Carron at Sgodachail          241.1
## 6 3003 Oykel at Easter Turnaig          330.7
```

# United Kingdom: site location

```
ggplot()+ theme_bw()+
  geom_polygon(data = world, aes(long, lat, group = group),
               size=0.5, fill = "grey90", color = "gray98") +
  coord_cartesian(xlim=c(-10,3), ylim=c(50,60))+
  geom_point(data = UKsites, aes(x=longitude,y=latitude),
             pch=21, color="blue2", fill="lightblue")
```



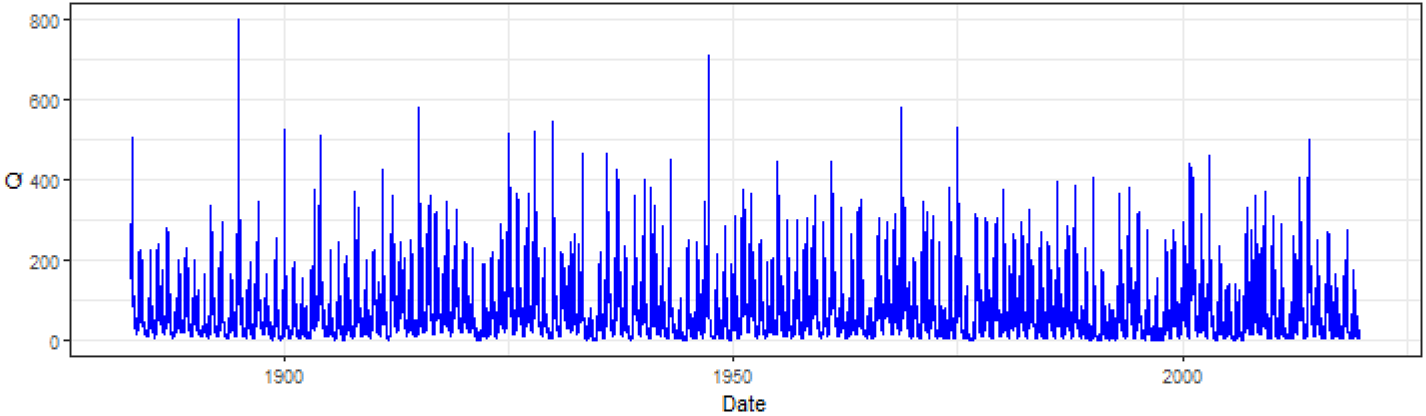
# United Kingdom: time series

Download just one site: e.g. the River Thames at Kingston, site 39001

```
df <- as.data.frame(gdf(id=39001, metadata = TRUE))
df$Date <- as.Date(row.names(df))
names(df)[names(df) == 'gdf'] <- 'Q'
```

Time series:

```
ggplot(df)+
  geom_line(aes(x=Date, y=Q), col="blue")+
  theme_bw()
```



# Canada

# Canada: hydat package

Below we use the tidyhydat package by Albers et al. (2020). Check out Sam Albers's vignettes: intro and examples.

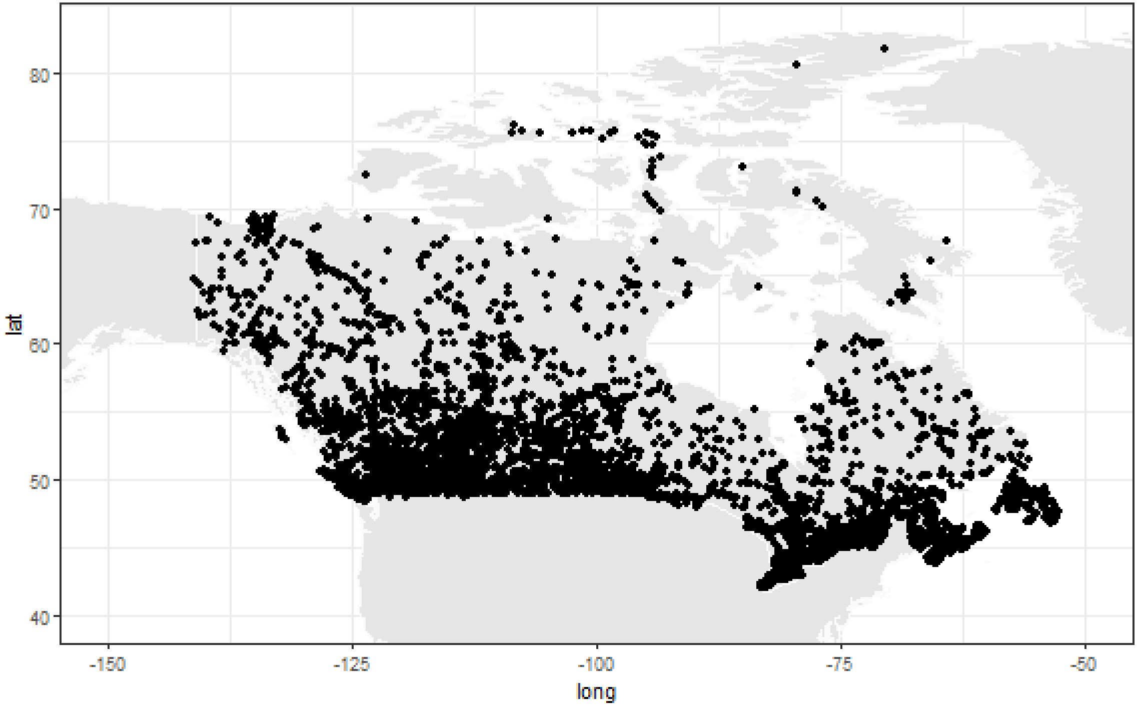
First, as before, retrieve list of sites:

```r
library(tidyhydat)

# download_hydat() # this takes about 10 minutes
CAsites <- hy_stations()

# retrieve list of sites
sites <- unique(CAsites$STATION_NUMBER)
sites[1:3] # first three
```

```
## [1] "01AA002" "01AD001" "01AD002"
```

# Canada: site location

```
ggplot()+
  geom_polygon(data = world, aes(long, lat, group = group),  size=0.5
               fill = "grey90", color = "gray98") +
  coord_cartesian(xlim=c(-150,-50), ylim=c(40,83))+
  geom_point(data = CAsites, aes(x=LONGITUDE,y=LATITUDE))+
  theme_bw()
```



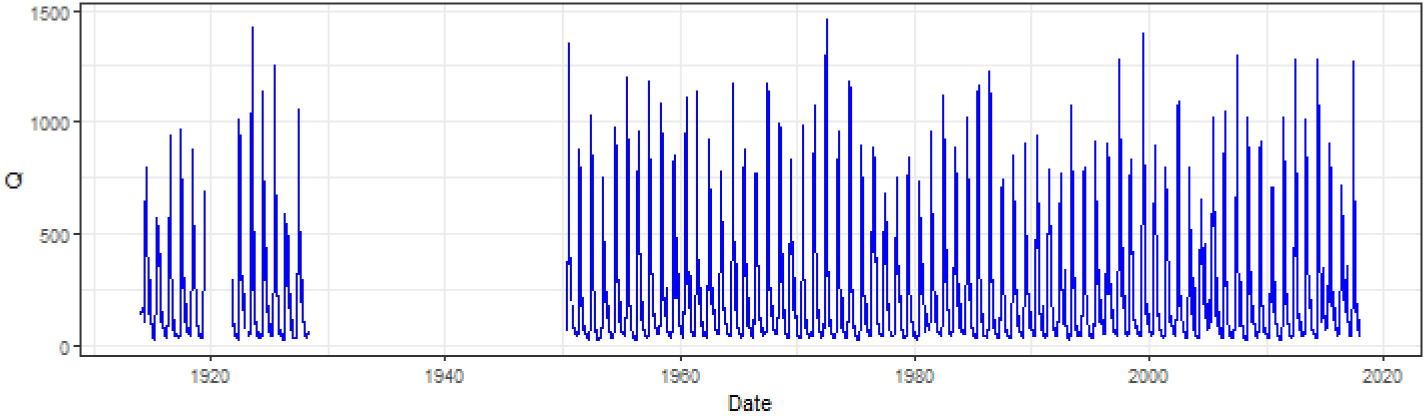
# Canada: time series

To download one site:

```
dfC <- hy_daily_flows(station_number = "08LA001")
names(dfC)[names(dfC) == "Value"] <- "Q"
```

Time series:

```
ggplot(dfC)+
  geom_line(aes(x=Date, y=Q), col="blue")+
  theme_bw()
```



# Greece

# Greece: hydroscoper package

We will use the hydroscoper package by Vantas et al. (2021). See Konstantinos Vantas's blogpost and vignette: an introduction to hydroscoper.

Retrieve list of sites:

```r
library(hydroscoper)

# load full data catalogue
data("stations")
GRcatalogue <- subset(stations,
                      subdomain =  c("kyy", "ypaat", "emy", "deh"),)
```

# Greece: variables

Multiple variables are available:

```
data("timeseries")
unique(timeseries$variable)[1:10]
```

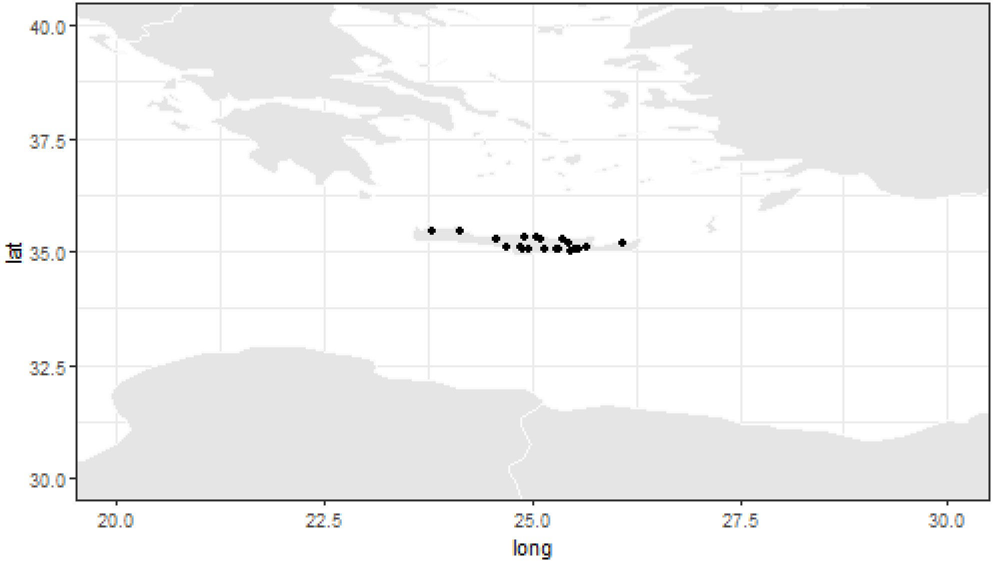
```
##  [1] "temperature_max"        "wind_direction"
##  [3] "temperature_min"        "flow"
##  [5] "snow"                   "wind_speed"
##  [7] "wind_speed_average"     "precipitation"
##  [9] "evaporation_estimation" "evaporation_present"
```

We only want streamflow:

```
timeseries <- subset(timeseries, variable=="flow")
# Merge in the lat/lon
GRsites <- merge(timeseries, GRcatalogue, all.x=TRUE)
```

# Greece: site location

```
ggplot()+
  geom_polygon(data = world, aes(long, lat, group = group),
               size=0.5, fill = "grey90", color = "gray98") +
  coord_cartesian(xlim=c(20,30), ylim=c(30,40))+
  geom_point(data = GRsites, aes(x=longitude,y=latitude))+
  theme_bw()
```



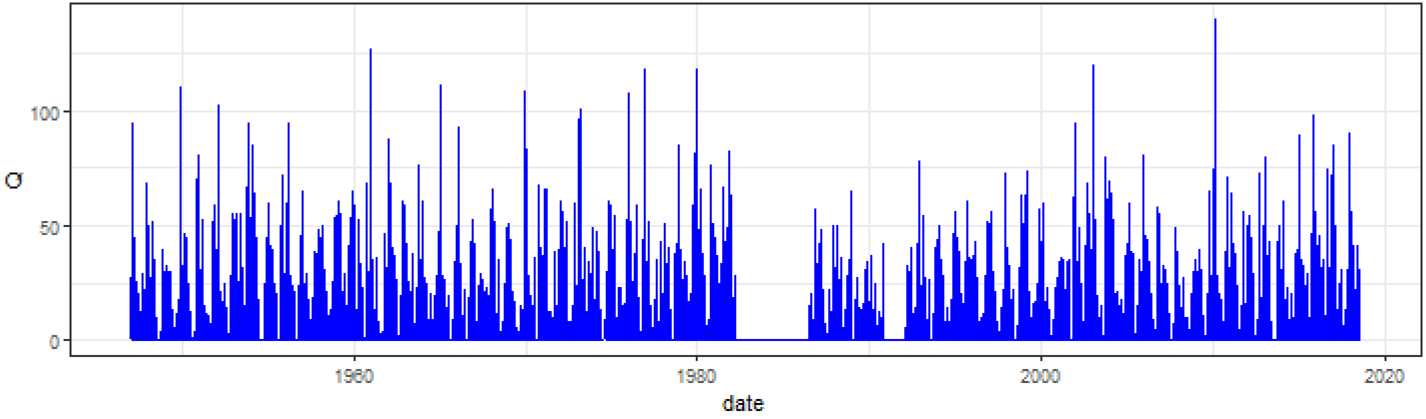
# Greece: time series

Select one site using the time_id from the dataset (GRsites)

```
dfG <- get_data(subdomain = "kyy", time_id = 753)
names(dfG)[names(dfG) == "value"] <- "Q"
```

Time series:

```
ggplot(dfG)+
  geom_line(aes(x=date, y=Q), col="blue")+
  theme_bw()
```



# All 4 countries

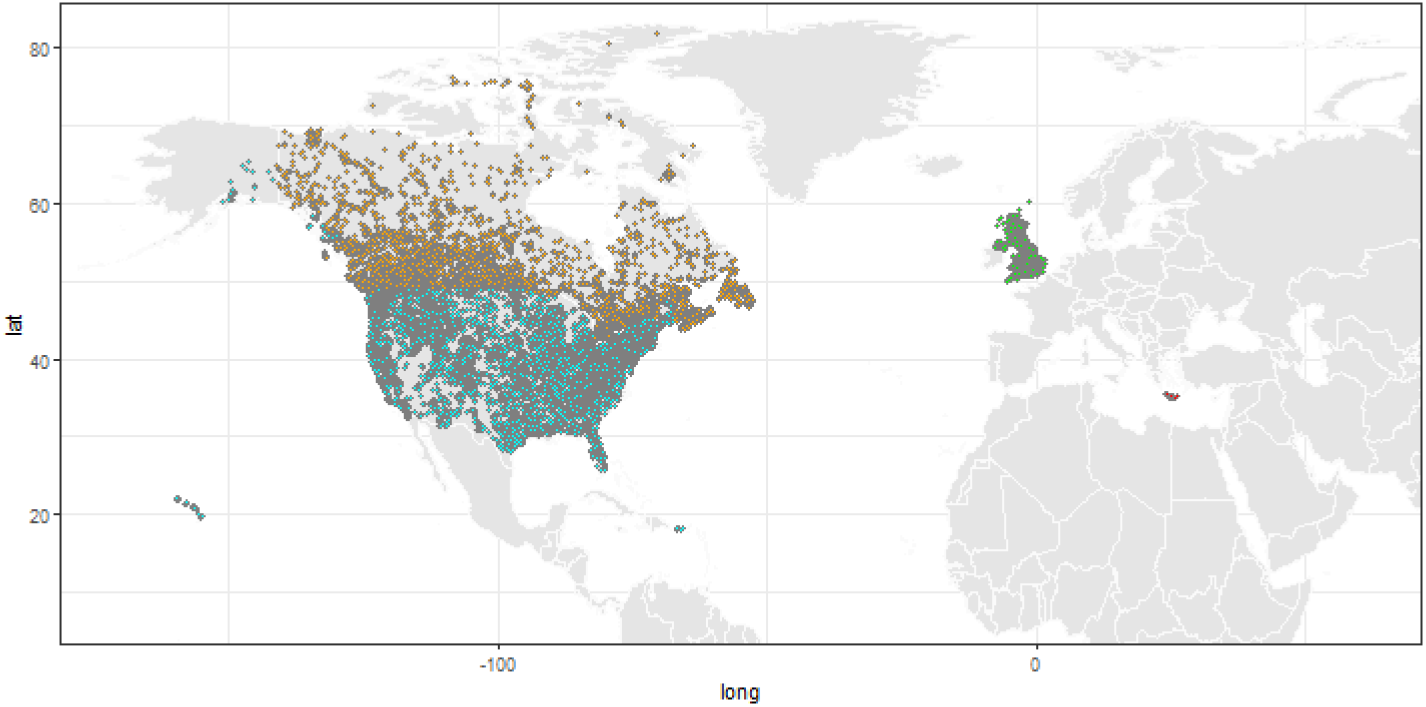
# All countries

Let's add together the different datasets we obtained

```
ggplot()+
  geom_polygon(data = world, aes(long, lat, group = group),  size=0.5
               fill = "grey90", color = "gray98") +
  geom_point(data = GRsites, aes(x=longitude, y=latitude),
             pch=21, size=1,col="grey50",fill="red")+
  geom_point(data = CAsites, aes(x=LONGITUDE, y=LATITUDE),
             pch=21, size=1,col="grey50",fill="orange")+
  geom_point(data = UKsites, aes(x=longitude, y=latitude),
             pch=21, size=1,col="grey50",fill="green")+
  geom_point(data = USsites, aes(x=dec_long_va, y=dec_lat_va),
             pch=21, size=1,col="grey50",fill="cyan")+
  coord_cartesian(xlim=c(-170,60), ylim=c(7,82))+
  theme_bw()
```

# All countries

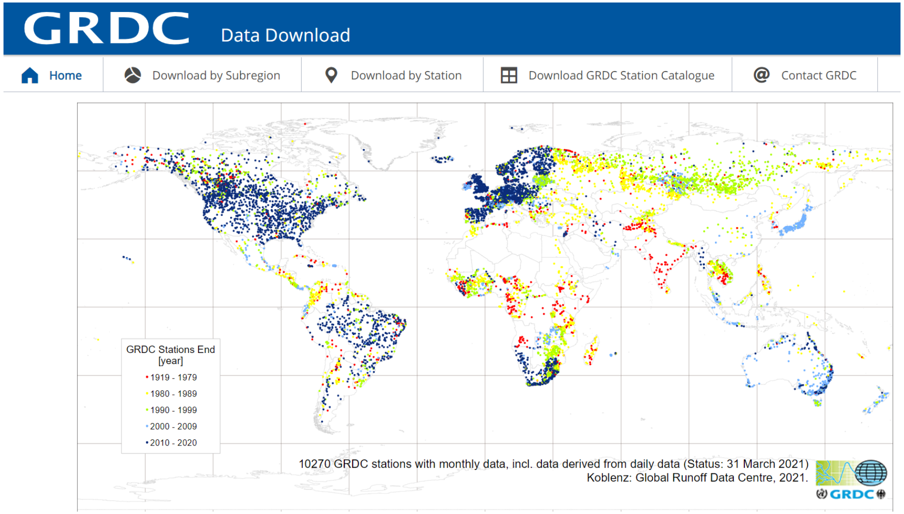
Let's add together the different datasets we obtained



# Additional datasets worth exploring!

# Global Runoff Data Centre (GRDC)

Global data can be obtained from the Global Runoff Data Centre (GRDC) -- see the portal here.



10270 GRDC stations with monthly data, incl. data derived from daily data (Status: 31 March 2021)
Koblenz: Global Runoff Data Centre, 2021.

For instance, in this paper we combined multiple real-time datasets with the GRDC dataset: Slater et al (2021). Global Changes in 20-year, 50-year and 100-year River Floods. *Geophysical Research Letters*, e2020GL091824

# CAMELS datasets

The CAMELS (catchment attributes and meteorology for large-sample studies) datasets provide large integrated hydrologic datasets for regions of the world. CAMELS datasets already exist for:

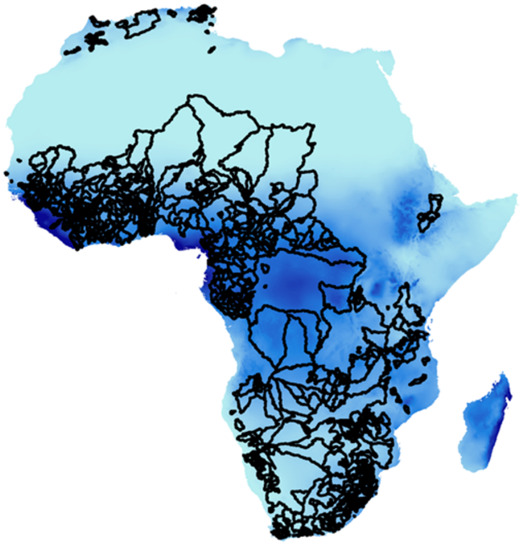
- USA (Addor et al. 2017)
- GB (Coxon et al. 2020)
- Australia (Fowler et al. 2021)
- Brazil (Chagas et al. 2020)
- Chile (Alvarez et al. 2018).

They usually include both the daily **time series** and catchment **attributes** (including topography, climate, hydrology, land cover, soils, and hydrogeology), and so are an extremely valuable resource.

# Africa: the ADHI

The African Database of Hydrometric Indices (ADHI) by Tramblay & Rouché 2020 contains catchment boundaries + time series for multiple stations of:

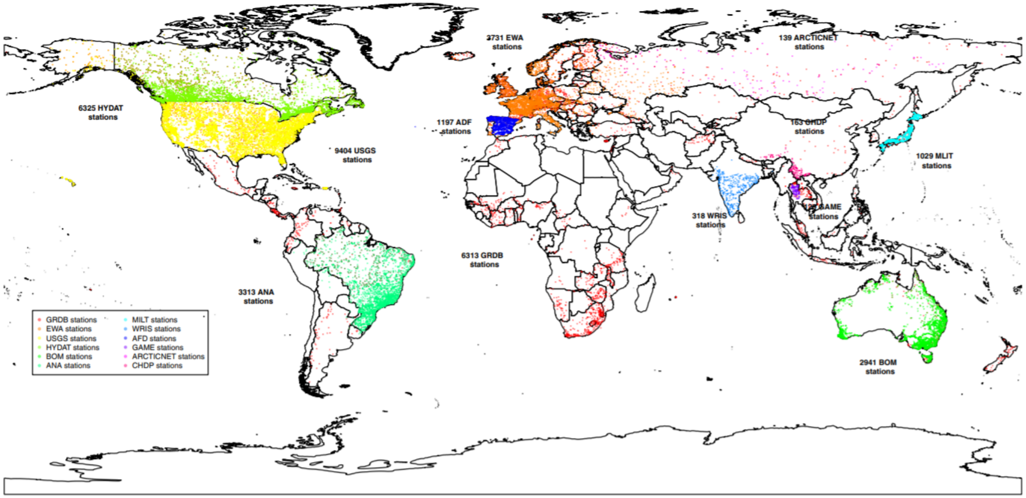
- annual minimum of 7-day discharge
- annual maximum runoff
- mean annual runoff
- streamflow percentiles (…& more)



# GSIM

The Global Streamflow Indices and Metadata Archive (GSIM) contains **indices and metadata**. It includes:

- A metadata catalogue;
- Catchment boundaries;
- Catchment metadata, from 12 gridded global data products (e.g. land cover type, soil type, and climate and topographic characteristics).



# European Floods Database

The European Flood Database was described in Hall et al. 2015. Annual time series for multiple sites (1960-2010) were shared as follows:

- The **dates** of annual maximum streamflows or water levels (daily or instantaneous values) for each calendar year, for 4,062 catchments (used in Blöschl et al. 2017), available here.
- The **annual maximum specific discharge** (m^3/s)/km^2 for each year (used in Blöschl et al. 2019), available here.

# Conclusions

I hope you have found this useful. It is straightforward to parallelise the download for many sites at once (see the presentation from the 2019 short course)!

Keep an eye on CRAN and the Hydrology task force for any new packages.

And please email me if you discover any other datasets or packages, so we can update this community resource in future years!
(`louise.slater@ouce.ox.ac.uk`)