# SANBI-GBIF training workshop in Data Management and Cleaning:

## Getting GBIF data in R

Vernon Visser

SEEC - Statistics in Ecology, Environment and Conservation
SANBI
Biodiversity for Life
South African National Biodiversity Institute

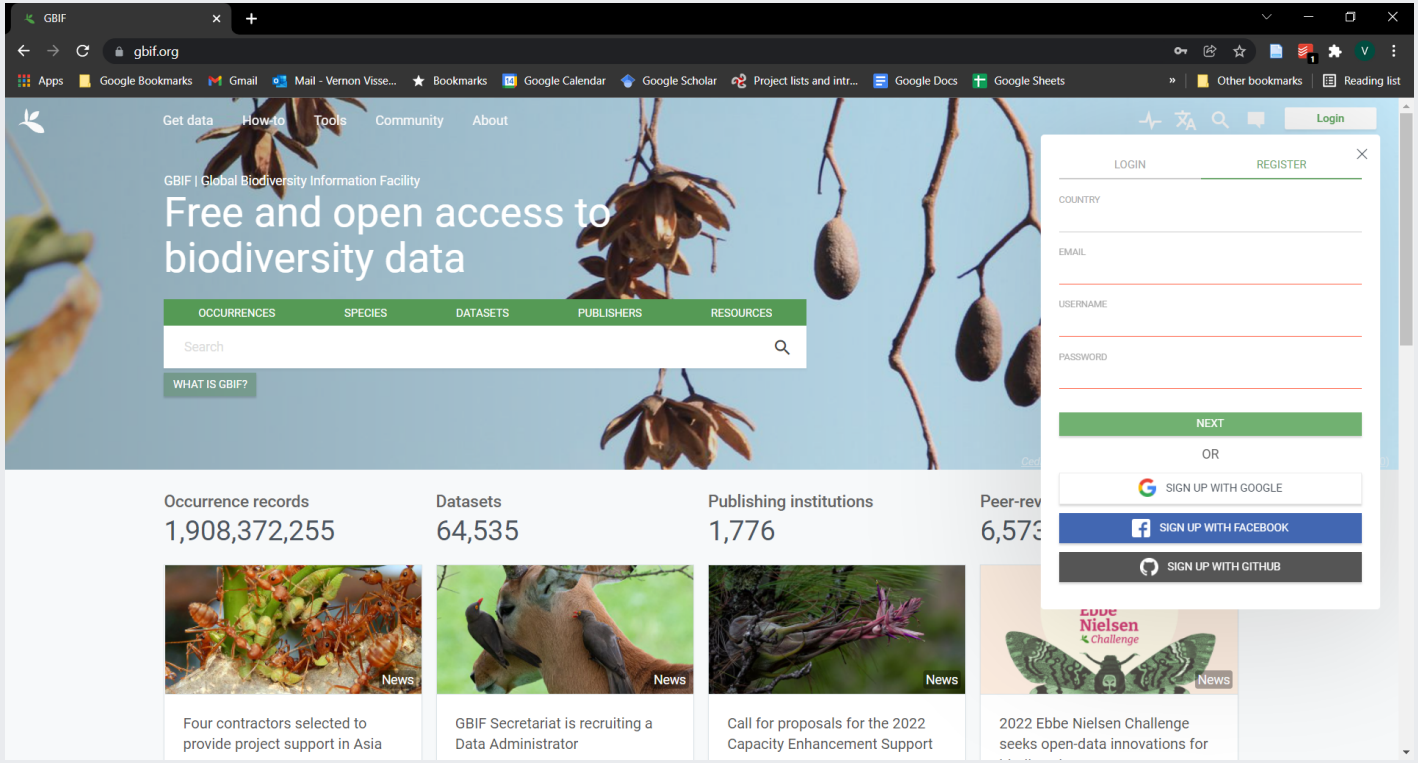
# Getting GBIF data using rgbif

We will look at how to obtain occurrence data from the Global Biodiversity Information Facility. As you will have learned from the previous session on GBIF, there are a number of ways of interrogating GBIF data (e.g. date, region, IUCN red list status...).

We will use an example of downloading data for a grass genus (*Pentameris*) in South Africa.

# Registering

The first step will be to create your own GBIF account. Once you have registered, log in to GBIF using the username and password you chose.



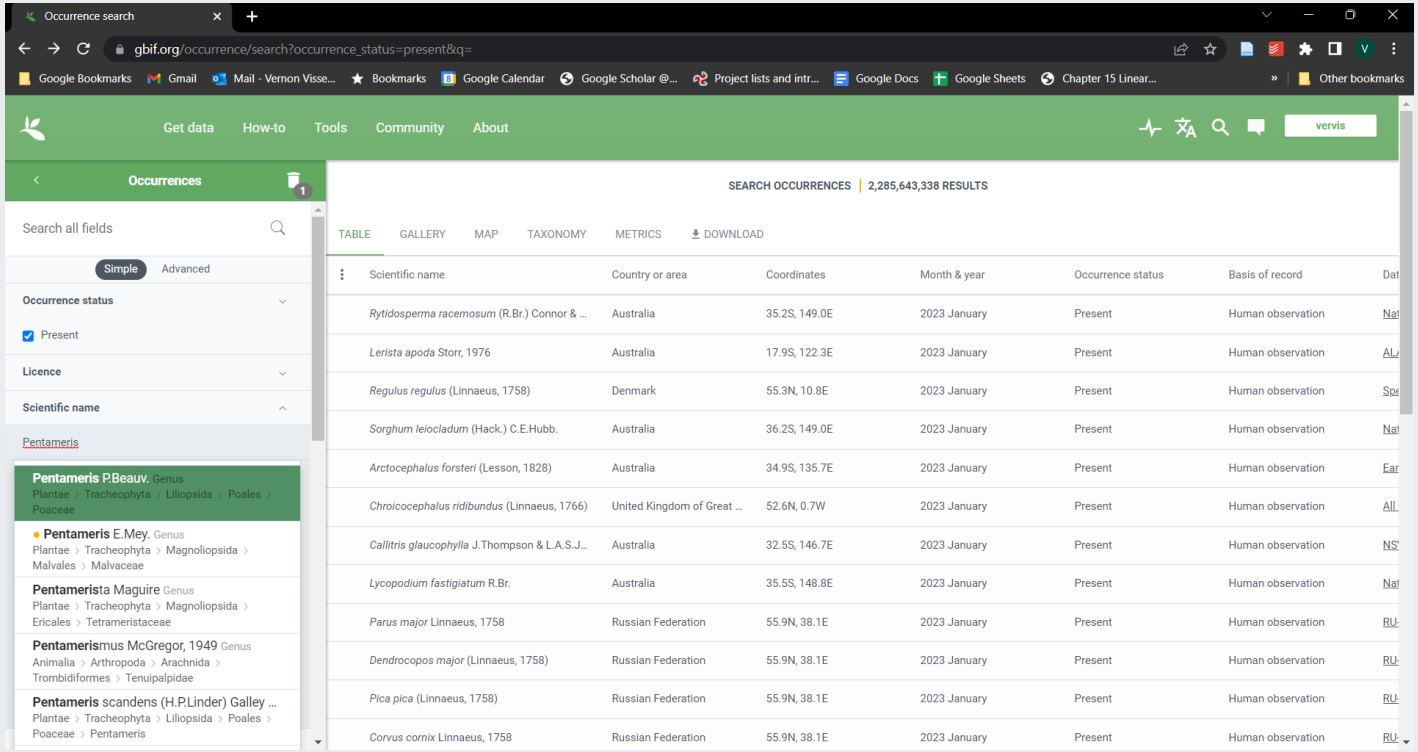
# Overview of what we will do

We will first look at how one downloads data directly from GBIF. This should help you understand the various options available on the portal.

We will then look at ways of **downloading data from GBIF in R**. We will use the `rgbif` package to do this.

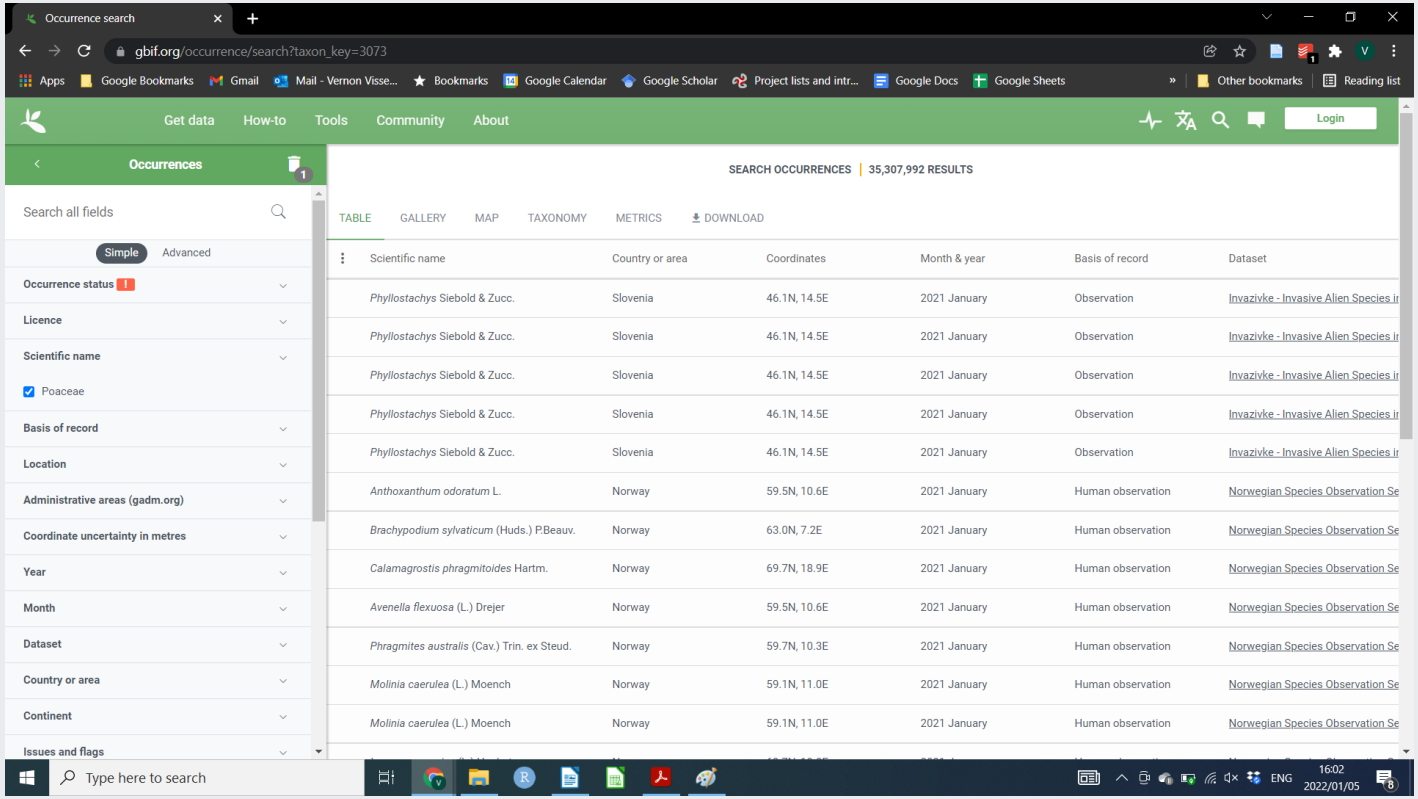
# First step - search

The first step will be to search for a particular taxon. Although this is possible from the GBIF home page, we are using the occurrences page because this is how you will search using filters, such as by country or publisher...



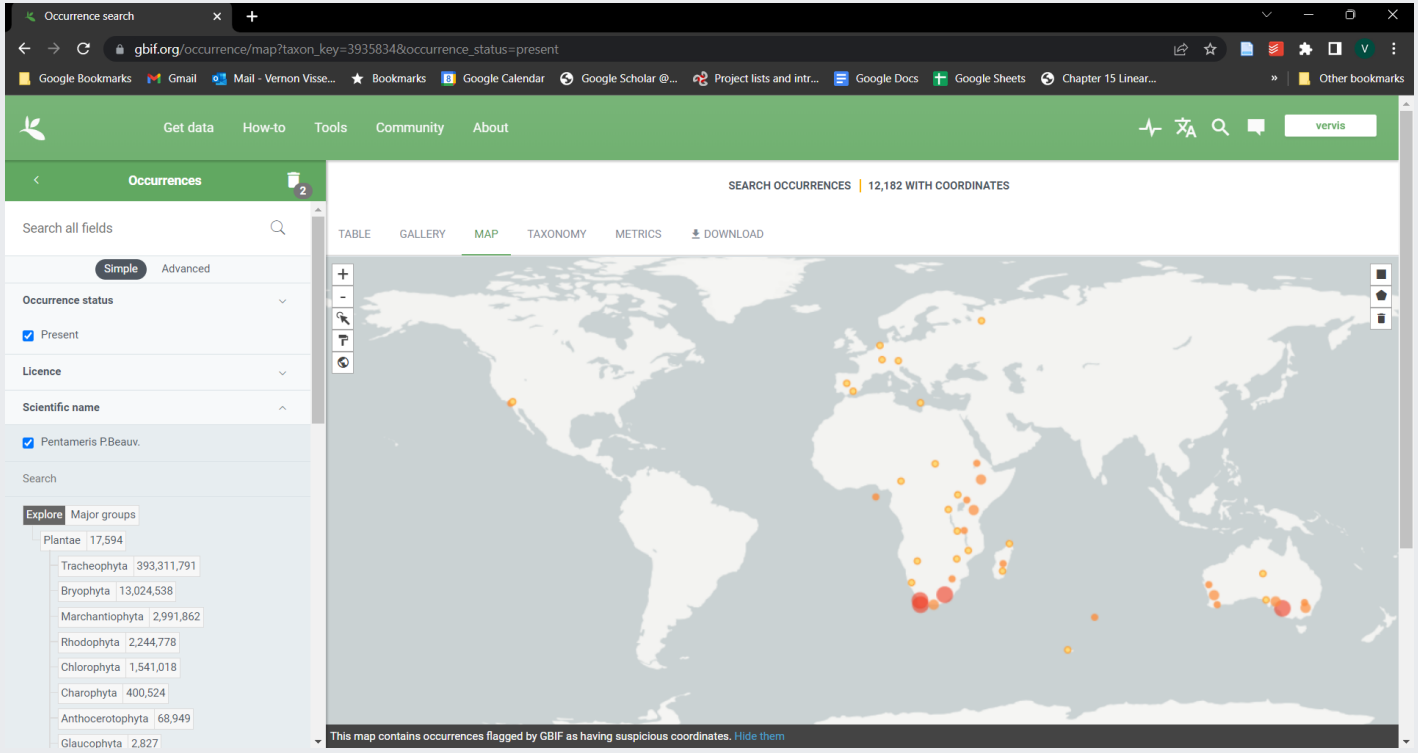
# Occurrences page

This is what the main occurrences search and filter page looks like. On the left you can see all the filter options available. Above the table of occurrences, you can see various tabs, which we will look at in the next few slides.



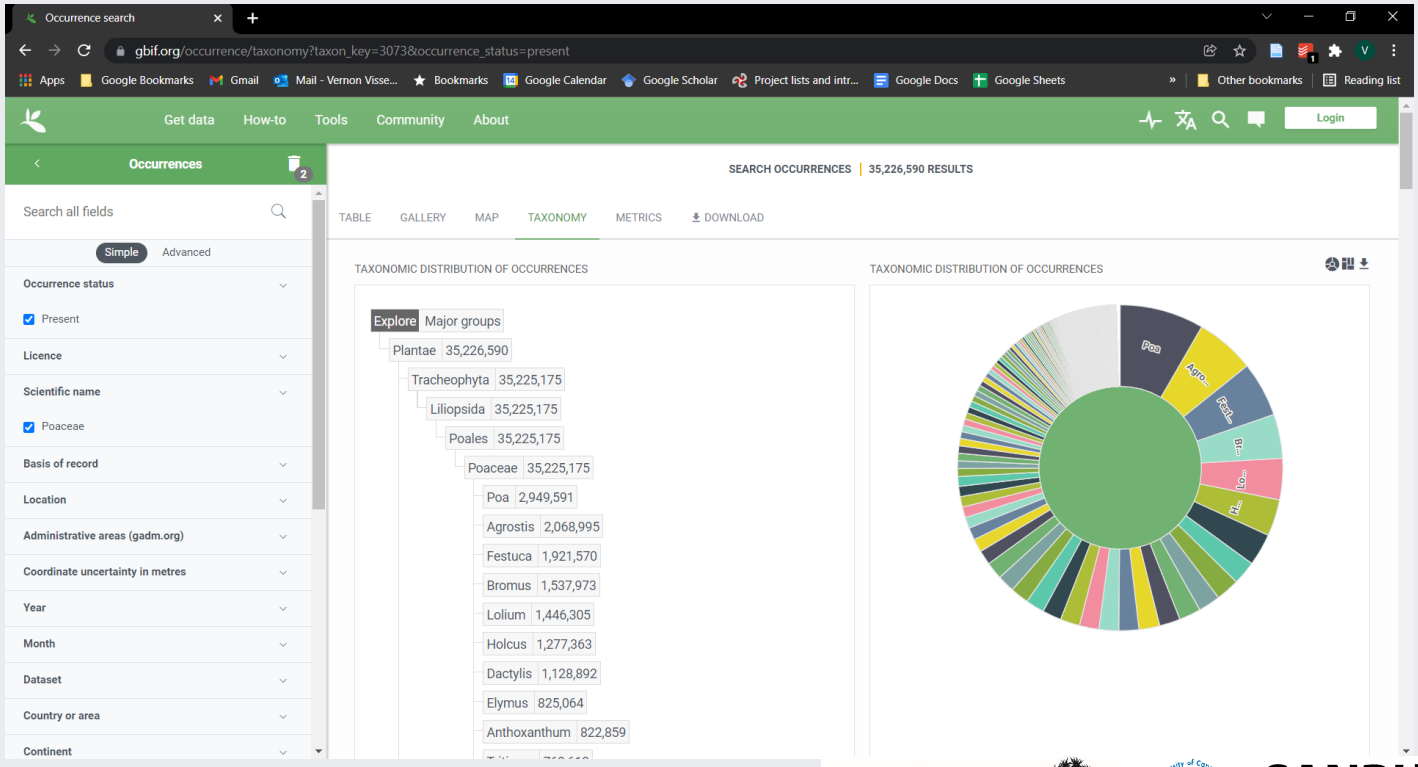
# Map of occurrences

In one of the occurrences tabs you can view the distribution (map) of the particular selection of occurrences you have made.



# Taxonomy of occurrences

In another of the tabs you can view the taxonomy of the particular selection of occurrences you have made. This is an important one to check to make sure you have selected the correct taxon and not used e.g., a taxon in a completely different group of animal or plants with the same name, an outdated name...

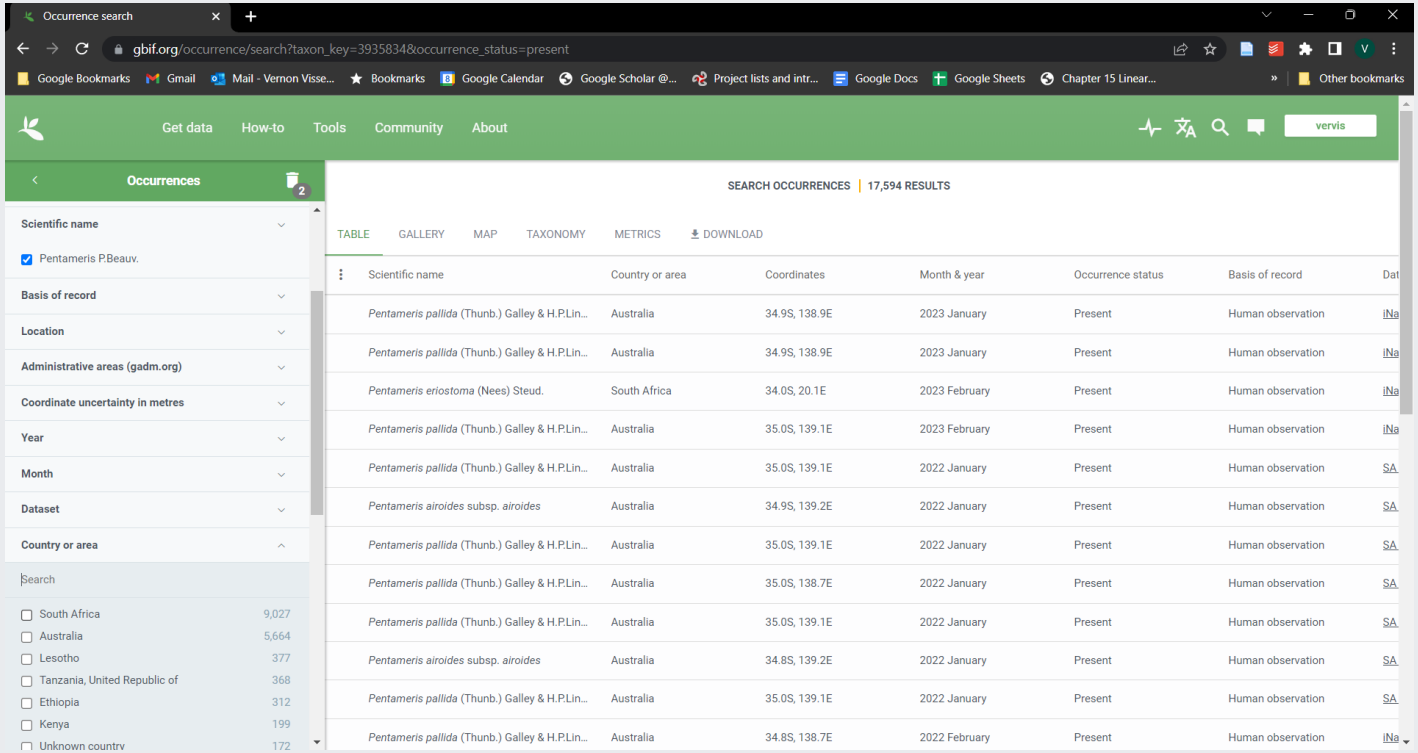



SEEC - Statistics in Ecology, Environment and Conservation

SANBI
Biodiversity for Life
South African National Biodiversity Institute

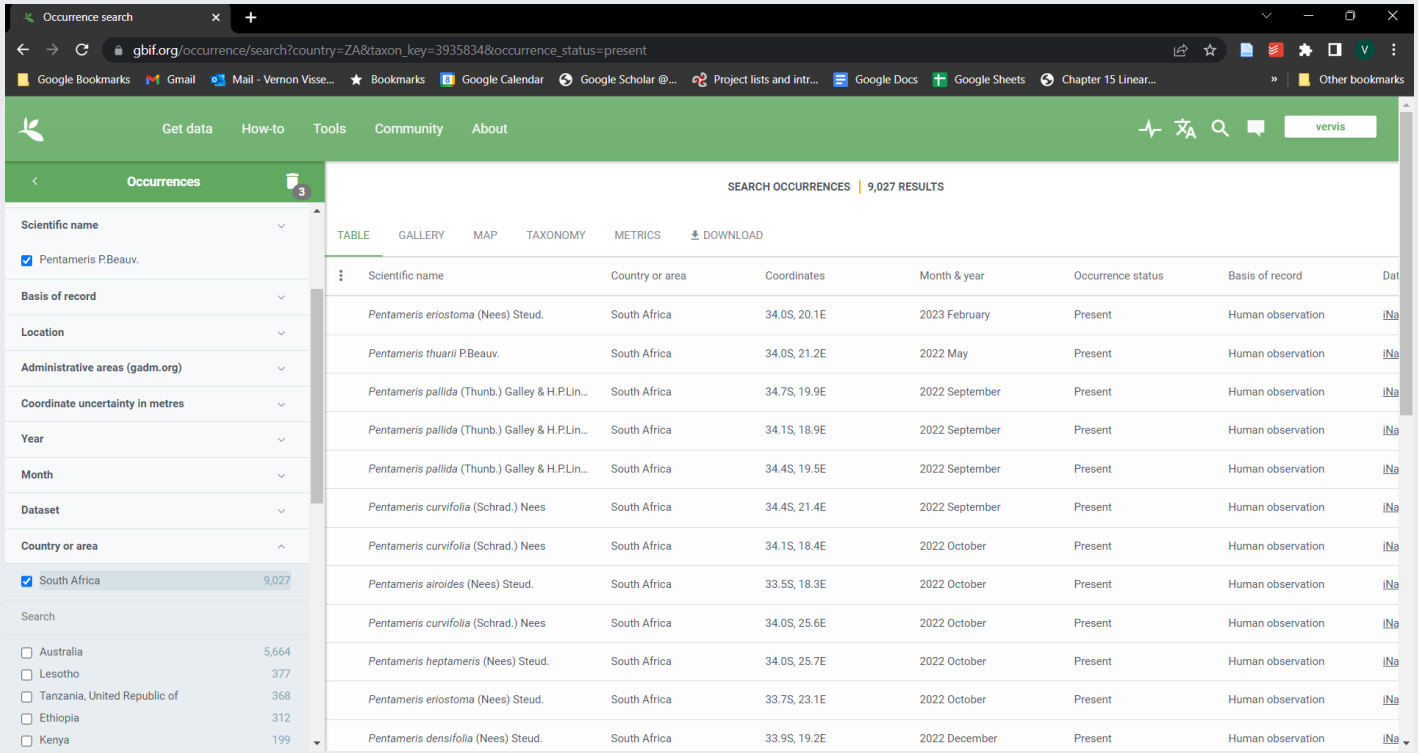
# Filtering by country

For our particular case we are going to filter *Pentameris* occurrences to just those in South Africa. You do this by selecting the "Country" filter on the left-hand side and searching for and selecting "South Africa".



# Pentameris occurrences in South Africa

You will see that the number of occurrences in the data table will drop by quite a bit.

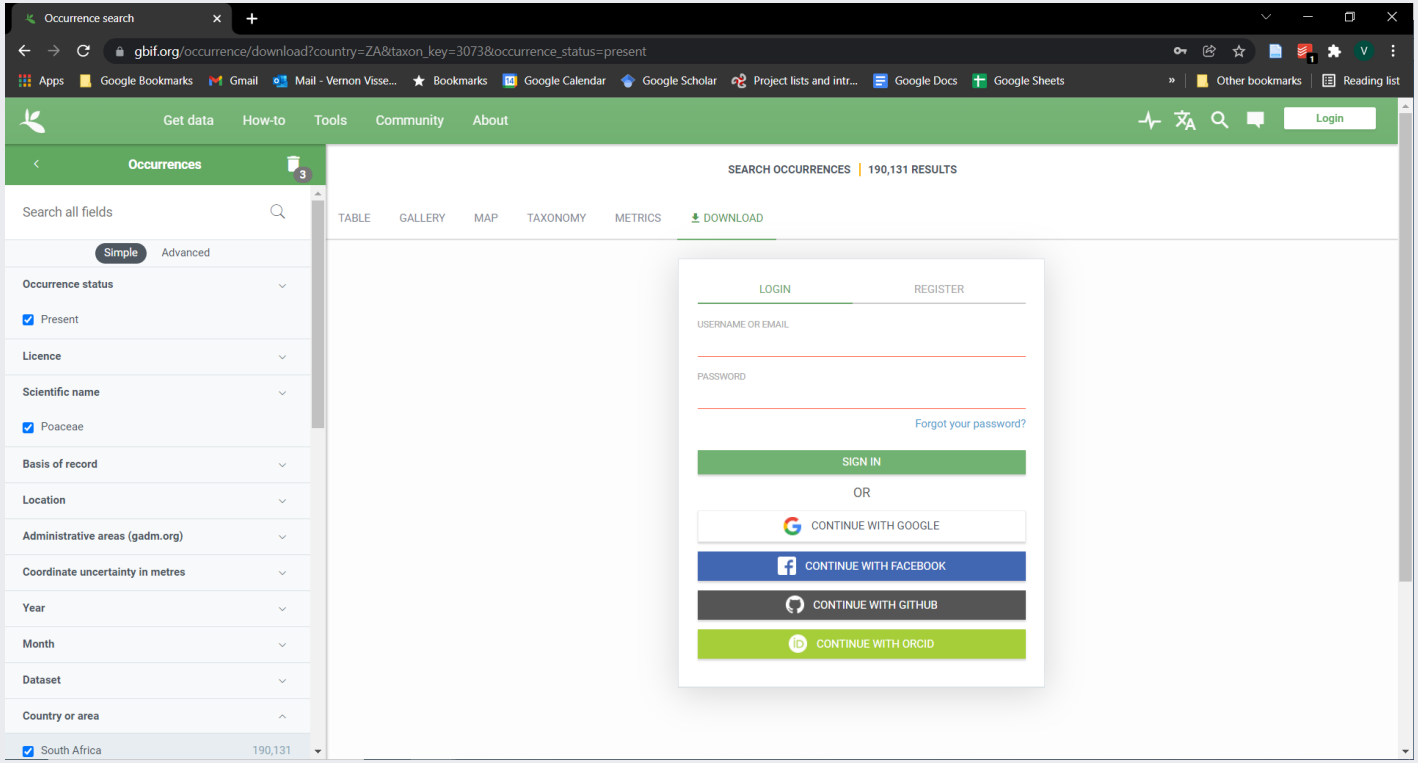



SEEC - Statistics in Ecology, Environment and Conservation

SANBI Biodiversity for Life
South African National Biodiversity Institute

# Downloading - log in

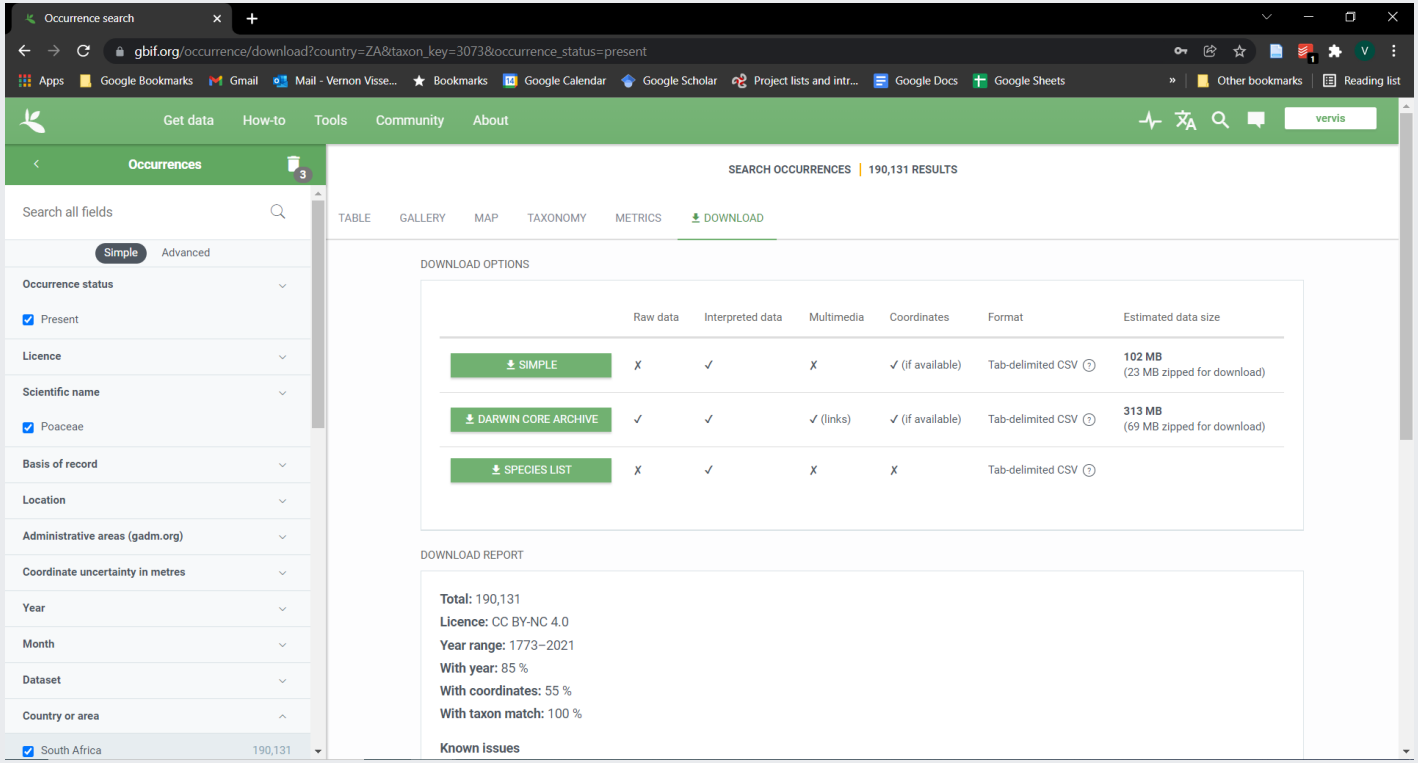
At this stage, if you have not already logged in, you will need to first log in to be able to download.



# Downloading - options

Once you are logged in, **click the download button** at the top right above the data table. You will get an option to download three different things. We will be using the **"Simple" CSV** option which provides a large comma-separated spreadsheet with occurrence records as rows and different metadata for each record as columns...

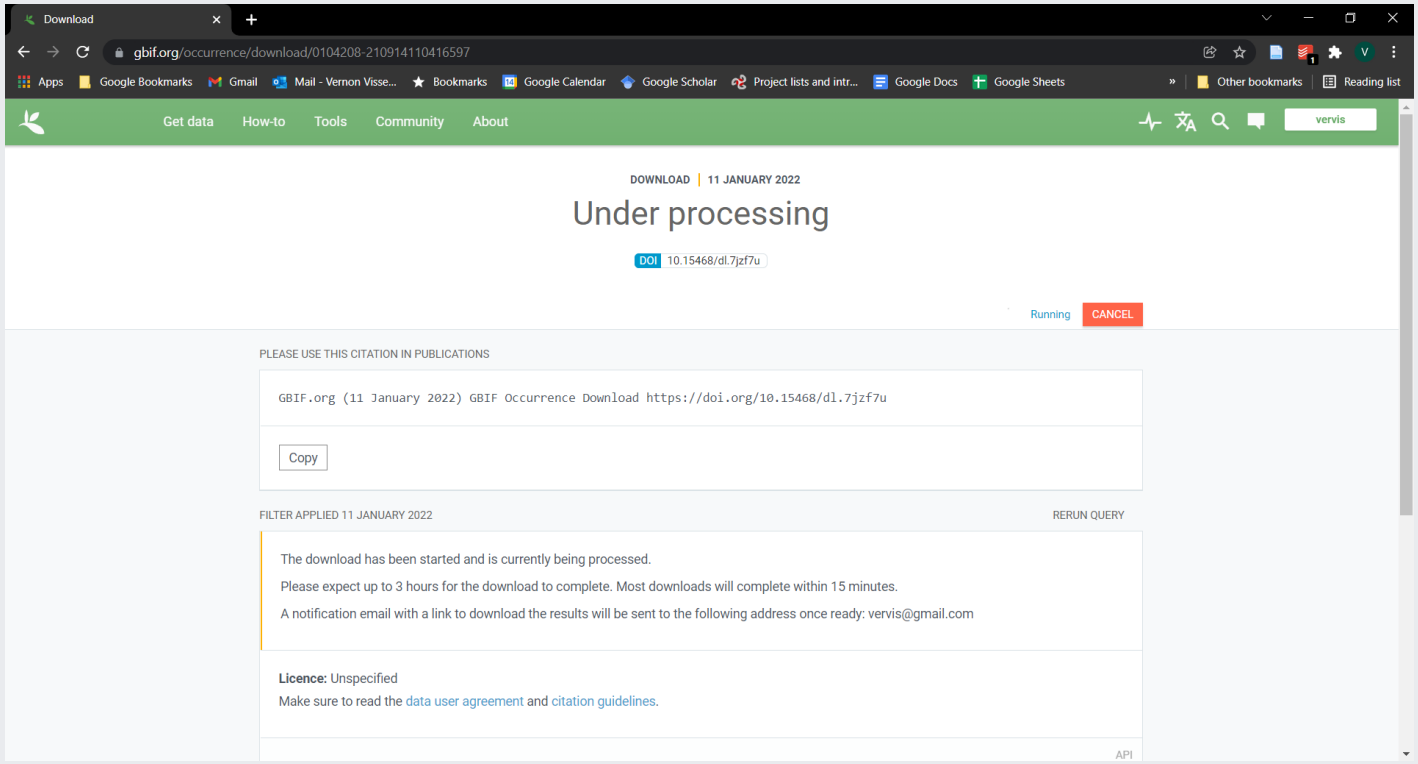
The "Darwin Core Archive" format you might remember from our earlier discussions on biodiversity data formats, but we will not be using this now. The "Species List" is exactly what it says and can be useful for checklists or for checking the taxonomy of your occurrences against different taxonomic databases, but this can also be achieved using the data we will be using.



# Download in process

Once you have selected to download the "Simple" CSV, your download will process and you will get a screen like the one on the next page. This is important for two reasons:

- You will get a unique DOI for your download. This needs to be used in any publications in which you use the data from GBIF you have just downloaded.
- If you refresh the page, it will give you an update on the status of your download. Usually though you just wait until you get an email informing you that your download is ready.




SEEC - Statistics in Ecology, Environment and Conservation

SANBI
Biodiversity for Life
South African National Biodiversity Institute

# Download

Once you receive an email saying your download is ready, you can download the data to your hard drive. It is good practice to have a set of folders for any project you are working on for raw data, like this, which I usually just call "Data" and another folder for edited data and results, which I usually call "Outputs".

# Using R to do GBIF downloads

Now that you have an idea of how to manually download data from GBIF, we are going to look at how to do this in R using the package `rgbif`. First we will install the package and then learn to how to search for a taxon to get a GBIF key, which is needed to do a download.

```
# Install the rgbif package. You need only do this once, i.e. the first time
# you run this code install.packages('rgbif')

# Load the packages we will need
library(rgbif)
library(dplyr)  #A useful package for data wrangling

# Use the name_suggest() function to find the best species name match for
# 'Pentameris barbata'
grassName = name_suggest(q = "Pentameris barbata", rank = "species")

# What is in our grassName object?
names(grassName)
```

```
## [1] "data"      "hierarchy"
```

```
# Take a look at the data
grassName$data
```

```
## # A tibble: 1 × 3
##       key canonicalName      rank
##     <int> <chr>              <chr>
## 1 4146807 Pentameris barbata SPECIES
```

# rgbif approaches to downloading

GBIF provides two ways to get occurrence data:

1. By searching for occurrences using `occ_search()`,
2. By downloading occurrences using the numerous `occ_download*()` functions. `occ_search()` is better for smaller data requests, while `occ_download*()` functions are better for larger data requests.

# Small downloads

Let's take a look at a small dataset query using `occ_search()`

```
# Let's do our search using the taxonKey we found using the name_suggest()
# function
P_barbata = occ_search(taxonKey = grassName$data$key)

# We can look at the metadata of this object using
P_barbata$meta

# We could achieve the same thing, but without first checking the taxon name,
# using:
P_barbata = occ_search(scientificName = "Pentameris barbata")

# To access the actual data you can use:
P_barbata$data
```

Some things to note about using `occ_search()` are (1) that you do not get a DOI for your download and (2) you are limited to a maximum of 100,000 records. You need a DOI if you want to publish using these data, but it is possible to get this using the `derived_dataset()` function. However, we will rather use the `occ_download*()` approach from here on in.

# Large downloads

Let's see how to do a large download using our earlier example of downloading all *Pentameris* records in South Africa.

```
# First get the taxon key
taxonKey = name_suggest(q = 'Pentameris', rank = "genus")

# Set up your username etc., which will be needed later
# user = "" # GBIF user name
# pwd = "" # GBIF password
# email = "" # Your email

# Set up the download. This query is sent to GBIF, just as you did manually and then we need to import the
download after this.
PentamerisInSADownload = occ_download(
  pred_in("taxonKey", taxonKey$data$key), # Set the taxonKey
  pred("hasCoordinate", TRUE), # Select only records with coordinates
  pred("country", "ZA"), # Select only records in South Africa
  format = "SIMPLE_CSV", # Get simple CSV format
  user = user, pwd = pwd, email = email # Give rgbif your credentials for GBIF
)

# We can now look at the download key
PentamerisInSADownload

# Let us take a look and see if our download is ready and see metadata associated with it
occ_download_meta(key = "0081229-230224095556074")

# Download the dataset from GBIF to a folder you choose and import it into R
PentamerisInSA = occ_download_get(key = "0081229-230224095556074", path = "../Data/", overwrite = TRUE) %>%
#Download the data
  occ_download_import(PentamerisInSA, na.strings = c("", NA), quote = "") #Import data into R
```

# Importing very large downloads

Using `occ_download_import()` works for large but not VERY large datasets. If your download is larger than say a million records, you might be better off just running the `occ_download_get()` part of the code and then reading the data in using a "large data" package, such as `data.table`.

Before you do this, you can unzip the GBIF download using e.g., 7zip, or directly from within R and then read the CSV file into R, like below.

```
unzip("../Data/0081229-230224095556074.zip", exdir="../Data")

#Read CSV data into R
# install.packages(data.table) #First install data.table, if you do not already have it
library(data.table)
dat = fread("../Data/0081229-230224095556074.csv", data.table = FALSE, fill = F, encoding = "UTF-8", quote="")
```

# Take a look at our data

```
names(dat)
```

```
##  [1] "gbifID"                           "datasetKey"
##  [3] "occurrenceID"                     "kingdom"
##  [5] "phylum"                           "class"
##  [7] "order"                            "family"
##  [9] "genus"                            "species"
## [11] "infraspecificEpithet"             "taxonRank"
## [13] "scientificName"                   "verbatimScientificName"
## [15] "verbatimScientificNameAuthorship" "countryCode"
## [17] "locality"                         "stateProvince"
## [19] "occurrenceStatus"                 "individualCount"
## [21] "publishingOrgKey"                 "decimalLatitude"
## [23] "decimalLongitude"                 "coordinateUncertaintyInMeters"
## [25] "coordinatePrecision"              "elevation"
## [27] "elevationAccuracy"                "depth"
## [29] "depthAccuracy"                    "eventDate"
## [31] "day"                              "month"
## [33] "year"                             "taxonKey"
## [35] "speciesKey"                       "basisOfRecord"
## [37] "institutionCode"                  "collectionCode"
## [39] "catalogNumber"                    "recordNumber"
## [41] "identifiedBy"                     "dateIdentified"
## [43] "license"                          "rightsHolder"
## [45] "recordedBy"                       "typeStatus"
## [47] "establishmentMeans"               "lastInterpreted"
## [49] "mediaType"                        "issue"
```

# Take a look at our data

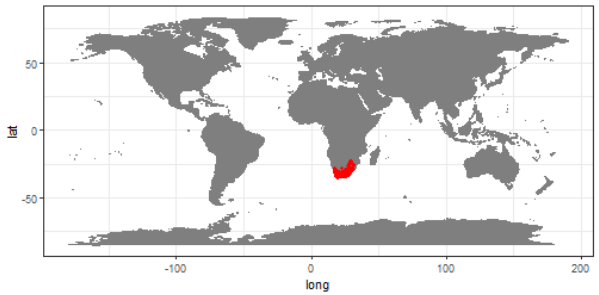
```
head(dat)
```

```
##      gbifID                              datasetKey
## 1 912558605 cd6e21c8-9e8a-493a-8a76-fbf7862069e5
## 2 912558604 cd6e21c8-9e8a-493a-8a76-fbf7862069e5
## 3 912558594 cd6e21c8-9e8a-493a-8a76-fbf7862069e5
## 4 912558592 cd6e21c8-9e8a-493a-8a76-fbf7862069e5
## 5 912558591 cd6e21c8-9e8a-493a-8a76-fbf7862069e5
## 6 912558587 cd6e21c8-9e8a-493a-8a76-fbf7862069e5
##                                    occurrenceID kingdom       phylum      class
## 1 http://specimens.kew.org/herbarium/K000719928 Plantae Tracheophyta Liliopsida
## 2 http://specimens.kew.org/herbarium/K000719938 Plantae Tracheophyta Liliopsida
## 3 http://specimens.kew.org/herbarium/K000719932 Plantae Tracheophyta Liliopsida
## 4 http://specimens.kew.org/herbarium/K000719940 Plantae Tracheophyta Liliopsida
## 5 http://specimens.kew.org/herbarium/K000719919 Plantae Tracheophyta Liliopsida
## 6 http://specimens.kew.org/herbarium/K000719925 Plantae Tracheophyta Liliopsida
##    order  family      genus              species infraspecificEpithet taxonRank
## 1 Poales Poaceae Pentameris  Pentameris dregeana                        SPECIES
## 2 Poales Poaceae Pentameris   Pentameris pusilla                        SPECIES
## 3 Poales Poaceae Pentameris Pentameris glacialis                        SPECIES
## 4 Poales Poaceae Pentameris   Pentameris pusilla                        SPECIES
## 5 Poales Poaceae Pentameris  Pentameris dregeana                        SPECIES
## 6 Poales Poaceae Pentameris  Pentameris dregeana                        SPECIES
##                    scientificName          verbatimScientificName
## 1       Pentameris dregeana Stapf       Pentameris dregeana Stapf
## 2 Poagrostis pusilla (Nees) Stapf Poagrostis pusilla (Nees) Stapf
## 3 Pentameris glacialis N.P.Barker Pentameris glacialis N.P.Barker
## 4 Poagrostis pusilla (Nees) Stapf Poagrostis pusilla (Nees) Stapf
## 5       Pentameris dregeana Stapf       Pentameris dregeana Stapf
## 6       Pentameris dregeana Stapf       Pentameris dregeana Stapf
##   verbatimScientificNameAuthorship countryCode
## 1                            Stapf          ZA
## 2                      (Nees) Stapf          ZA
## 3                        N.P.Barker          ZA
## 4                      (Nees) Stapf          ZA
## 5                            Stapf          ZA
```

# Make a map of our occurrences

```
# install.packages(ggplot2) #First install ggplot2, if you do not already have it
library(ggplot2) #Needed for making plots
#Get a world map that is just grey in colour
wm = borders("world", colour="gray50", fill="gray50")

#Plot dataset
ggplot(dat=dat) +
   coord_fixed() + #Specify coordinate system
   wm + #Add world map
   geom_point(aes(x = decimalLongitude, y = decimalLatitude), colour = 'red', size = 1) + #Add points for
occurrences
   theme_bw() #Specify a ggplot theme
```

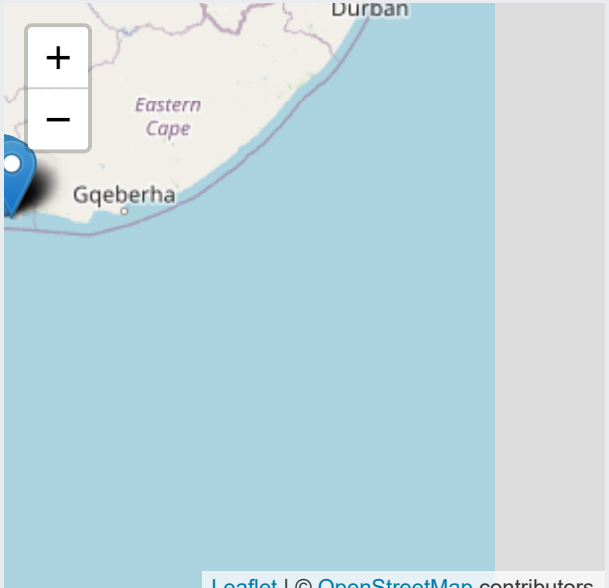


# Interactive map option

```
# install.packages(leaflet) #First install leaflet, if you do not already have it
library(leaflet) #Needed for making plots

P_barbata = dat %>%
  filter(species=="Pentameris barbata") %>% #Select only Pentameris barbata occurrences
  filter(!is.na(decimalLongitude)) #Remove occurrences with no coordinates

leafP_barbata = leaflet::leaflet() %>%
  leaflet::addProviderTiles(providers$OpenStreetMap) %>%
  leaflet::addMarkers(
    lng = P_barbata$decimalLongitude,
    lat = P_barbata$decimalLatitude,
    data = P_barbata,
    popup = ~as.character(P_barbata$gbifID)
  )
leafP_barbata
```



# Now you have data, but need to interrogate and clean it next...