Formatting Data for Morphology CURE

Alex Krohn

7/20/2020

Introduction

This document leads you through the process of finding, downloading and subsampling one dataset used in the Plant Morphology CURE. The dataset includes the images and metadata of any photographed herbarium specimens of Purple Loosestrife (**Lythrum salicaria**).

First, we must find the records with photos on iDigBio and download the data about those records. Second, we must subset those data to a manageable chunk (1000 random images here, but this may differ for you). Third, we must create a new spreadsheet with the information that is relevant to our projects: the location of the specimen, the date it was collected, and the link to the image. Fourth, and finally, we must download the images of those records, label each record uniquely so that it may be matched back to the correct metadata.

Finding and Downloading Data from iDigBio

To start, navigate in your browser to https://idigbio.org. Click "Search the Portal." A heat map of millions of specimens should appear. At left, under the "Filters" tab, type Lythrum salicaria into the Scientific Name box. Near the top, check the box indicating that the records "Must have media".

As of July 16th, 2020, 3,257 ocurrence records of **Lythrum salicaria** had 3,340 images on iDigBio. Your numbers may vary.

Click on the gray Download tab. Enter your email, and download the CSV. (You may have to wait a few minutes for the download to build and be ready to download.)

Once complete, you should have a zip file containing the records. Unzip the file to reveal a folder with a long name containing various TXT, XML and CSV files. We're going to assume you put these files into a folder called "CURE."

Subsetting the Images

First, set your working directory to the CURE folder so R knows where to find the CSVs.

```
setwd("CURE/")
```

Then, load the data on each specimen occurence, so you can decide which specimens that your class will analyze. Here, we'll randomly subset 1000 specimens, and then only keep relevant information. Colons (":") are a special character in R. R will not treat it as special if the name is within ". We will change these names to make them easier to deal with. If you'd like to select more columns, add them to select().

```
# Load the proper libraries
library(tidyverse)
library(glue)

# Load the data
```

```
occ.data <- read_csv("occurrence_raw.csv")

# Randomly choose 1000 records to work with, then keep only the columns with locality information, a un
occ.data.subset <-
  occ.data %>%
  slice_sample(n = 1000) %>%
  select(coreid, coll.date = `dwc:eventDate`, country = `dwc:country`, state = `dwc:stateProvince`, cou
```

You could, instead, subset by another parameter that's of interest to your class. Subsetting by geography might be best accomplished using iDigBio's filtering parameters on their website. Otherwise, you could use filter() as below. Note: not all data in this dataset have coordinates, not all data with coordinates have states/provinces/countries given, and not all data with states/provinces/countries given have coordinates. Filter carefully!

```
# Filter to only records from New Jersey
newjersey.data <-
  occ.data %>%
  filter(`dwc:stateProvince` == "New Jersey")
```

Add the Image URL to the Occurrence Data

From the multimedia.csv file, we'll use the unique specimen identifier (coreid) to find the correct URL. Then, we'll add this column of URLs to the occurence data spreadsheet.

```
# Import the multimedia data, but only the URL and the coreID
multimedia <-
  read_csv("~/Documents/UCSC/ucsc_online/Norris-Center/BCEENET/data_cleaning_tutorial/43b68f3d-bb1f-46c
  select(coreid, url = `ac:accessURI`)

# Add the URL to the occ.data.subset dataframe
occ.data.and.urls <-
  occ.data.subset %>%
  left_join(y = multimedia, by = "coreid")
```

Note, there are 27 occurrences with repeated coreids. These are specimens with multiple images.

Download the Images and Label them

Next, we'll download the images to your working directory, and rename them with the coreid.

First, create a folder within your CURE folder called images.

```
mkdir images
```

Next, create a vector of names for each image to be saved.

```
image.names <-
  occ.data.and.urls %>%
  select(coreid) %>%
  glue_data("{coreid}.jpg")
```

Next, create a vector of URLs to download the images from.

```
urls <- occ.data.and.urls$url
```

Finally, use the download.file() function to download each file from the URL, label it according to the coreID and store it in the images folder. This may take a while, depending on the number of images.

```
setwd("images/")

# Skip over any broken links
safe_download <- safely(~ download.file(.x, .y, mode = "wb"))

# Execute
walk2(urls, image.names, safe_download)
```