Oregon Bee Project Data Retrieval and Formatting Script

Summary

These scripts retrieve observation data from iNaturalist.org and reformat it for printing bee specimen labels.

There are two main processes:
1. Pulling and formatting data from iNaturalist.org
2. Merging and indexing formatted data

These processes can be run by executing (double-clicking) iNaturalist_Process.bat or Merge_Process.bat, respectively.

Installation

These scripts depend on a several pieces of third-party software to function. For developers' convenience, there is a list in OBP-Script/config/dependencies.txt. This section will provide instructions on how to install each in order on a Windows computer.

Python

Firstly, the scripts require a version of Python 3 installed.
1. Open a Command Prompt terminal.
2. Type "python --version" and press enter.
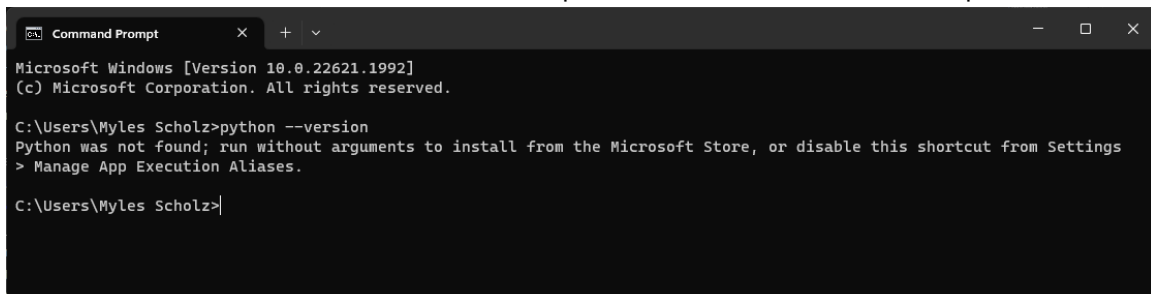   - If the command returns "Python 3.*.*", where * is any number, steps 3-5 are optional but recommended.

   - If the command returns a "Python was not found" error or a version of Python less than 3, close the Command Prompt window and continue with steps 3-5.
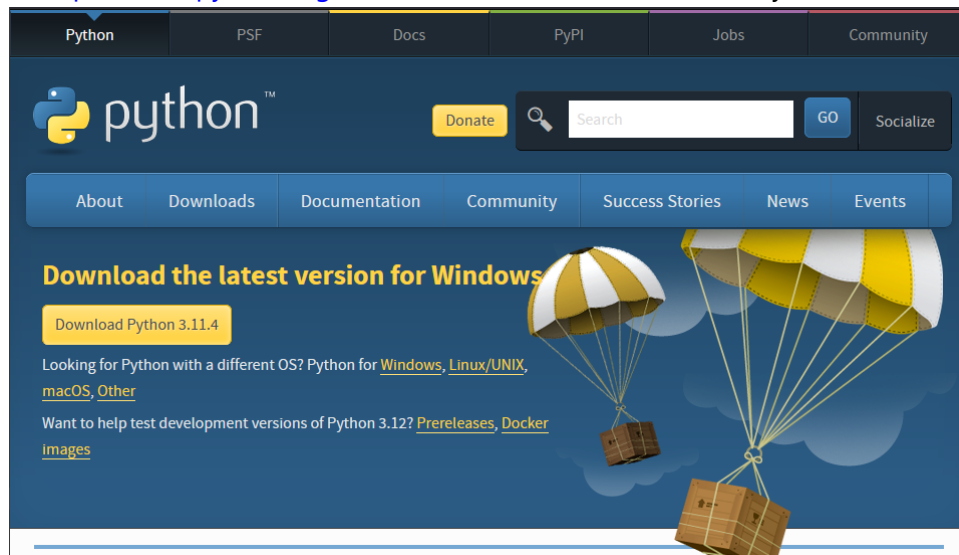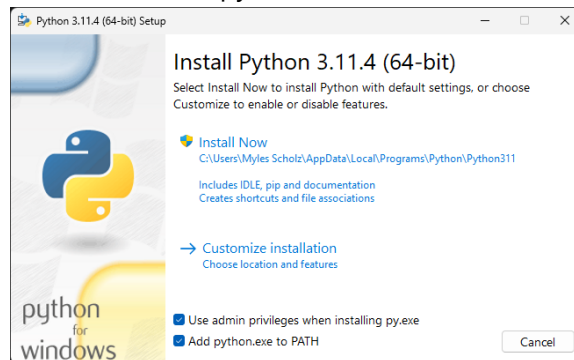
3. Go to https://www.python.org/downloads/ and click "Download Python 3.*.*".
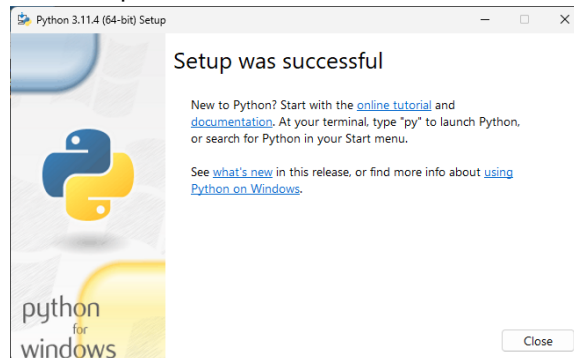

4. Go to the folder where the EXE file downloaded and execute (double-click) it.
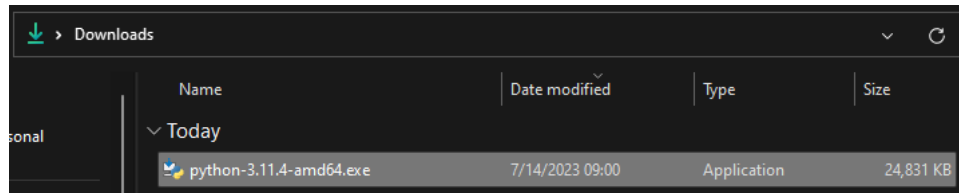
    o   Follow the installation instructions.
    o   Check the box marked "Add python.exe to PATH variable" when prompted.

5. When the installation is complete, continue to the next section ("Python Libraries").

Python Libraries

Next, the scripts need some external Python libraries installed.

1. Open a Command Prompt terminal.
2. Type "pip install pyinaturalist matplotlib treepoem ghostscript tqdm" and press Enter.

3. When installation is complete (the cursor is flashing next to a line ending with ">"), continue to the next section ("Ghostscript").

Ghostscript

Finally, some of the Python libraries (treepoem and ghostscript) require a version of the software Ghostscript installed.

1. Go to https://www.ghostscript.com/releases/gsdnld.html.

2. Click the link "Ghostscript AGPL release" next to either "Ghostscript 10.01.2 for Windows (64 bit)" or "Ghostscript 10.01.2 for Windows (32 bit)", depending on whether the computer is a 64-bit or 32-bit architecture.
    o To check the computer's architecture, open the System Information application. The "System Type" field will contain either x64 or x32, corresponding to 64-bit and 32-bit architectures, respectively.
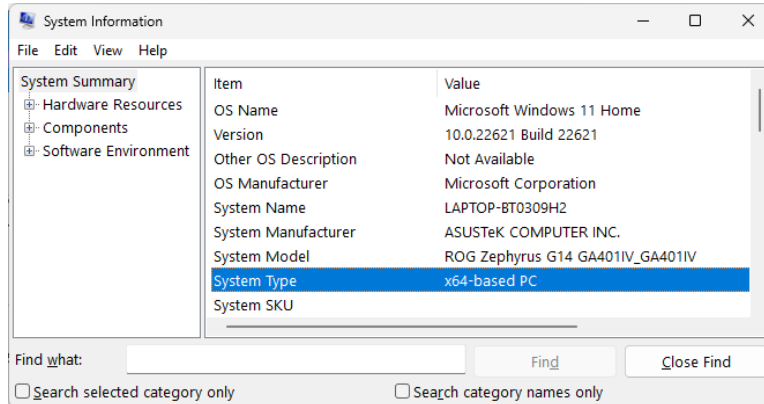
3. Go to the folder where the EXE file downloaded and execute (double-click) it.
    o This will require administrator privileges on the computer.
    o Follow the installation instructions without changing any of the options.
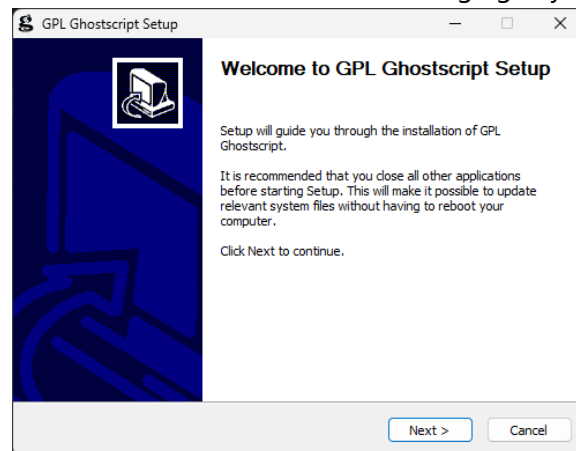
4. Wait until installation finishes.
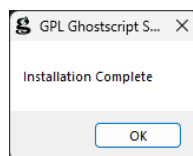
After completing the above steps, the scripts should be able to run on the computer.

Pulling and Formatting Data from iNaturalist.org

This script (iNaturalist_Process.bat) will query iNaturalist.org for observation data from a given year and from a given list of iNaturalist projects. When a user runs the script, they will be prompted to type a year to query. The list of iNaturalist projects to query is stored in OBP-Script/config/sources.txt (see "Changing Sources" below).

As of July 2023, the script pulls from the following projects:
- Oregon Bee Atlas (OBA)
- Master Melittologist (MM)
- Washington Bee Atlas (WaBA)

Running the Process
- Check that the script is set to pull data from the desired projects (OBP-Script/config/sources.txt).
  - If not, see "Changing Sources" below.
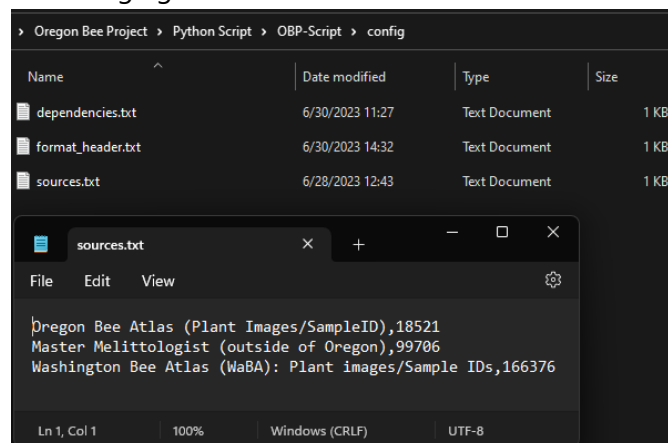
- Execute (double-click) iNaturalist_Process.bat.

o A terminal will open. When prompted, type a year to query and press Enter.

o Wait while the program fetches the data (this may take several minutes). The program will pull and format each source's data sequentially.

o The resulting formatted CSV files will appear in source-specific folders under OBP-Script/results/.

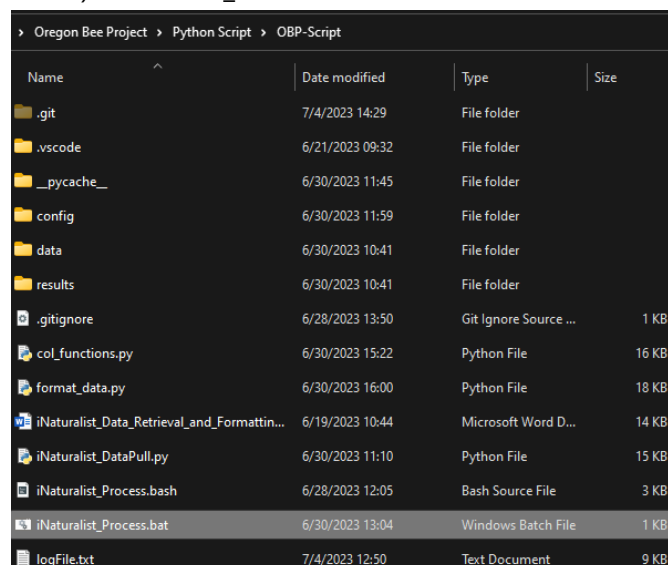Each source-specific folder will be named according to the format Abbr_M_D_YY, where Abbr is an abbreviation of the source and M_D_YY is the date when the program ran. For example, the results of fetching data from the Oregon Bee Atlas on July 3rd, 2023 would appear in OBP-Script/results/OBA_7_3_23/.

The resulting CSV file will be named according to the format Abbr_results_YYYY.csv, where Abbr is an abbreviation of the source and YYYY is the year that was queried. For example, querying Oregon Bee Atlas data from 2022 would produce a file named OBA_results_2022.csv.

If the program is run a second time on the same day, the results file will be entirely overwritten with new data.

Changing Sources

The script pulls data from a list of iNaturalist projects in OBP-Script/config/sources.txt. This file has a specific format that must be maintained for the script to run properly. Each source has its own line, and each line has two comma-separated fields: the name of the source and the iNaturalist project ID. The source name is not essential for the script to function; it is only used for improving the readability of the program's output. The iNaturalist project ID is essential for pulling data from the correct sources.

To add a source, do the following:
1. Open OBP-Script/config/sources.txt.
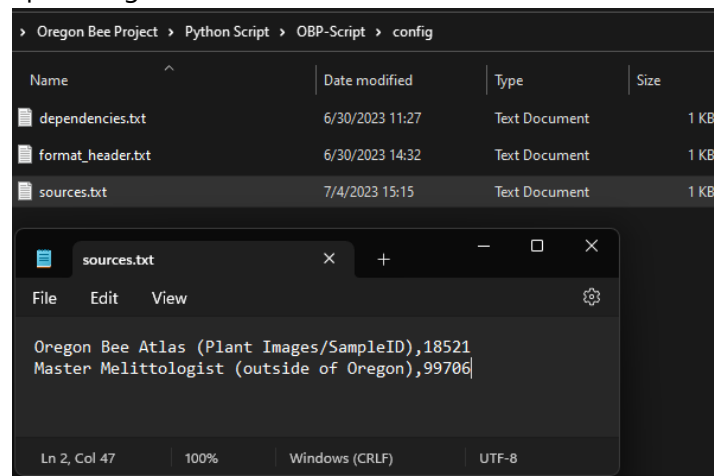
2. Add a line at the end of the file.
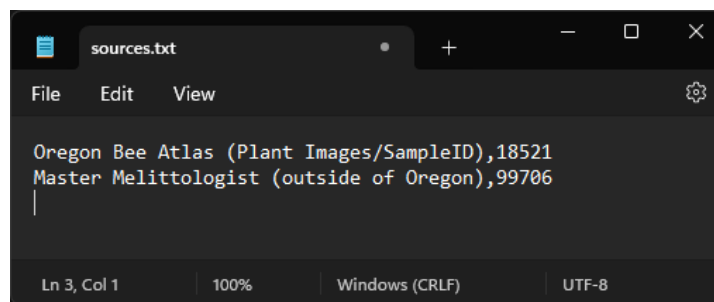
3. Type a name for the source.
   o The name cannot contain commas, but all other keyboard characters are allowed.
   o New sources will not have an abbreviation unless the code is modified. Output files and folders will use the project ID instead.
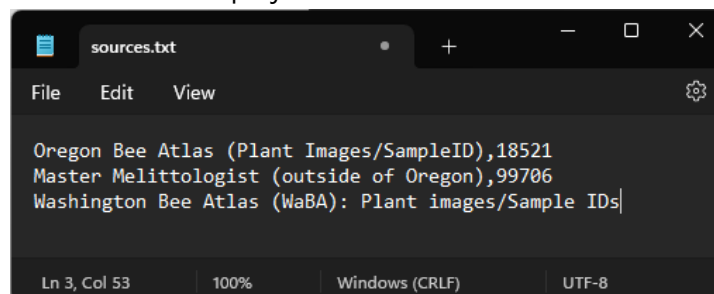
4. Type a comma and then the iNaturalist project ID.

The iNaturalist project ID can be found by doing the following:

- In a browser, go to https://www.inaturalist.org/observations/identify.
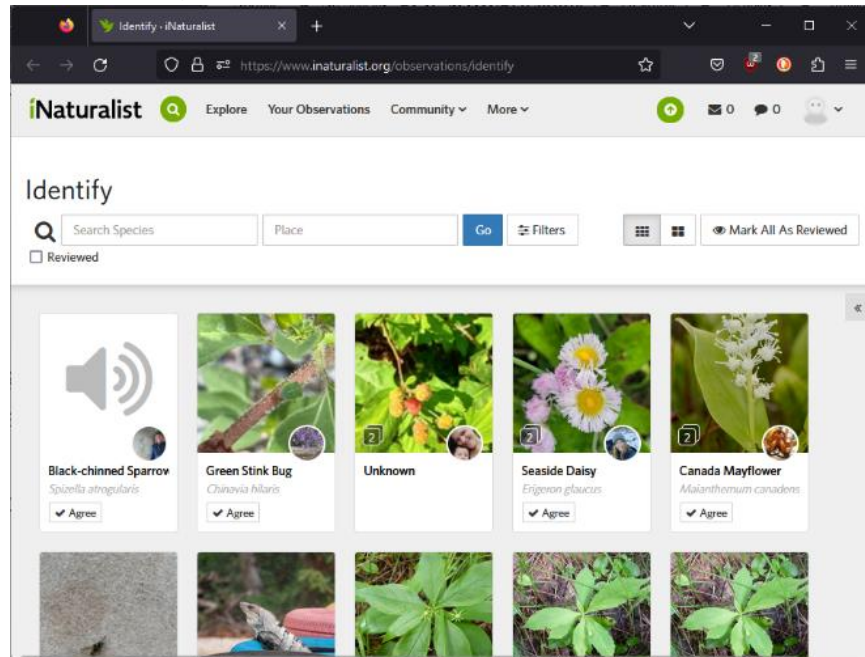
- Click the "Filters" button.
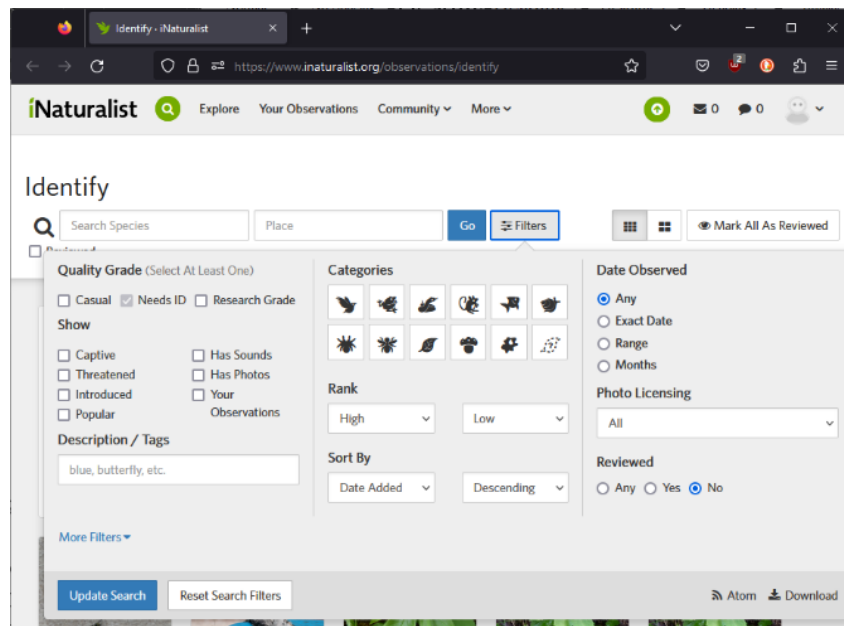
o Click the "More Filters" button.

o Type the name of the project in the "Project" field.
  ▪ Be sure to select the project from the results. The full project name should appear in green.

- o The project ID will appear as a string of about five numbers at the end of the browser's URL bar.
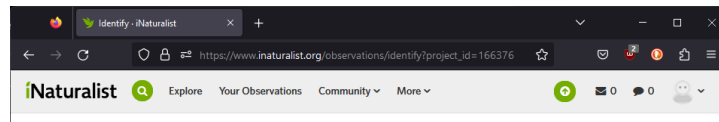
5. Save and close sources.txt.

To remove a source, do the following:
1. Open OBP-Script/config/sources.txt.
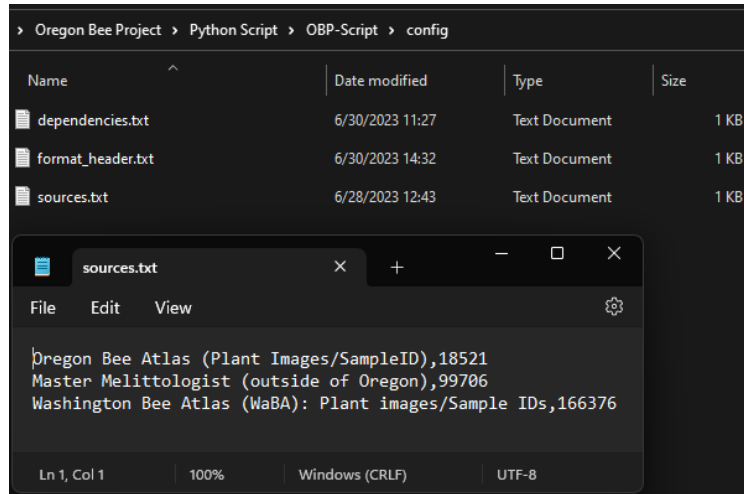
2. Select and delete the line containing the source to be removed.
   - o Do not leave the line empty. Press backspace on the empty line to remove it.
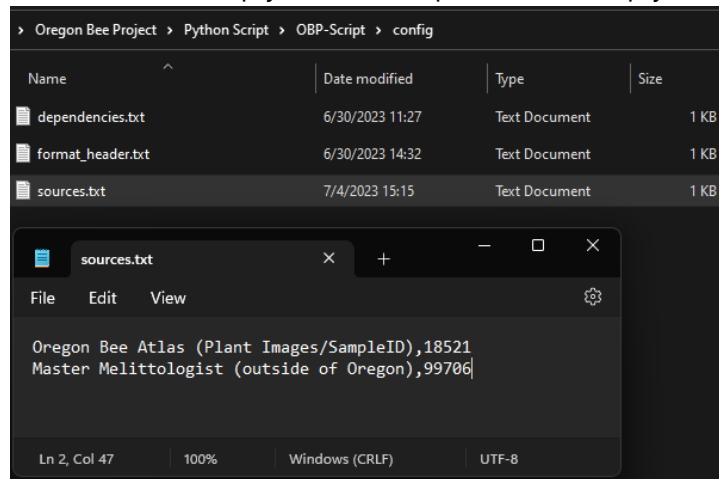
3. Save and close sources.txt.

Merging and Indexing Formatted Data

This script (Merge_Process.bat) will combine, sort, and index two formatted data files, such as the "results" files that are output from the data pulling process. When the user runs the script, they will be prompted for three file paths: the base file, the file to append to the base file, and the output file. These files must be CSV files and must have the exact header specified in OBP-Script/config/format_header.txt (see "Help and Tips" below).

Running the Process
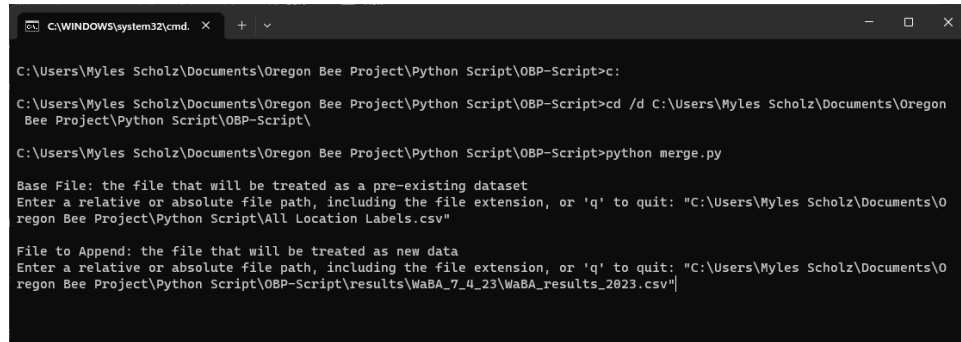
1. Execute (double-click) Merge_Process.bat.
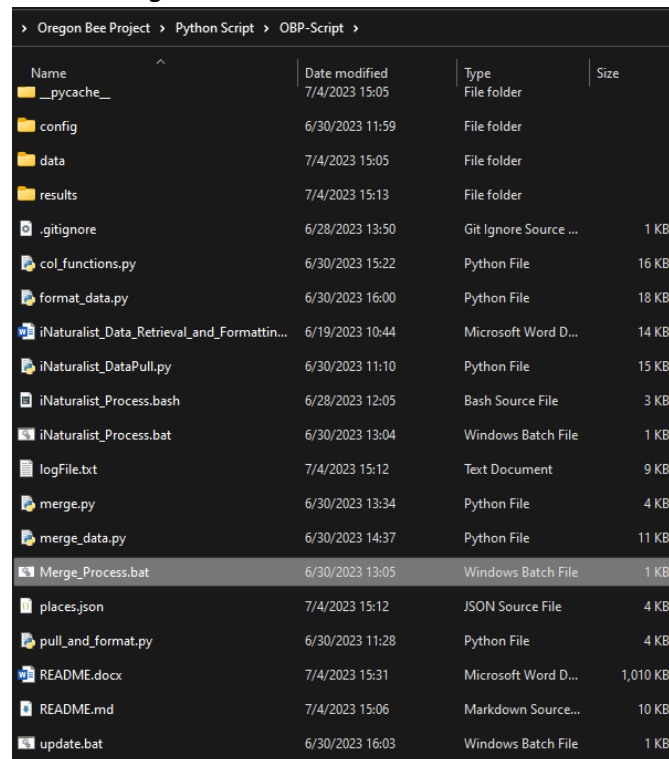
2. When prompted, type or paste the file path of the base file.
   - The program automatically trims quotation marks from the ends of the file path, so users may paste file paths copied directly from the File Explorer on Windows (see "Help and Tips" below).
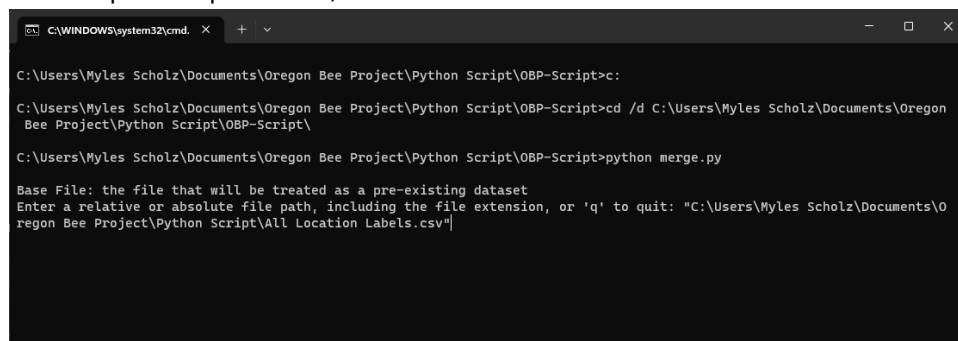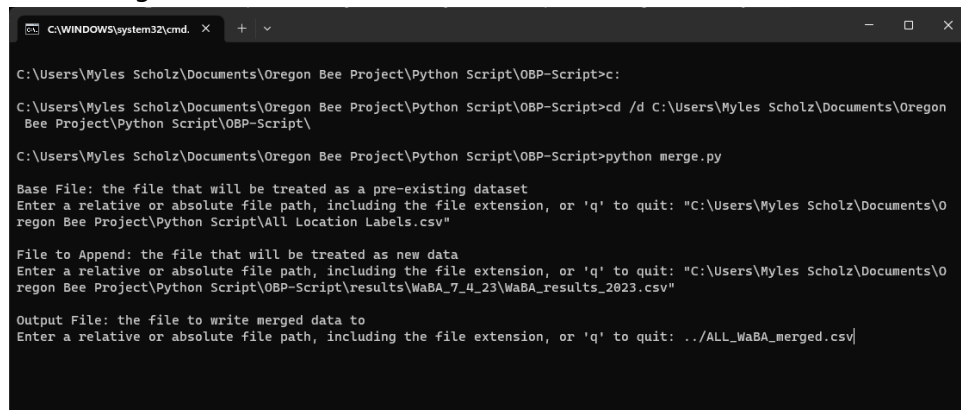
3. When prompted, type or paste the file path of the file to append to the base file.
   o The program automatically trims quotation marks from the ends of the file path, so users may paste file paths copied directly from the File Explorer on Windows (see "Help and Tips" below).

4. When prompted, type or paste a file path to output the merged data to.
   This may be a new file or an existing file but be careful because the program will overwrite all data in the target file.

5. Wait while the program merges the data (this may take a minute, depending on the size of the input files).

6. The resulting merged data will appear in the specified output file path.


```
C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>c:

C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>cd /d C:\Users\Myles Scholz\Documents\Oregon
 Bee Project\Python Script\OBP-Script\

C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>python merge.py

Base File: the file that will be treated as a pre-existing dataset
Enter a relative or absolute file path, including the file extension, or 'q' to quit: "C:\Users\Myles Scholz\Documents\O
regon Bee Project\Python Script\All Location Labels.csv"

File to Append: the file that will be treated as new data
Enter a relative or absolute file path, including the file extension, or 'q' to quit: "C:\Users\Myles Scholz\Documents\O
regon Bee Project\Python Script\OBP-Script\results\WaBA_7_4_23\WaBA_results_2023.csv"

Output File: the file to write merged data to
Enter a relative or absolute file path, including the file extension, or 'q' to quit: ../ALL_WaBA_merged.csv
Merging '..\All Location Labels.csv' and 'results\WaBA_7_4_23\WaBA_results_2023.csv' into '../ALL_WaBA_merged.csv'...
Merging '..\All Location Labels.csv' and 'results\WaBA_7_4_23\WaBA_results_2023.csv' into '../ALL_WaBA_merged.csv' => Do
ne!


Press any key to continue...
```
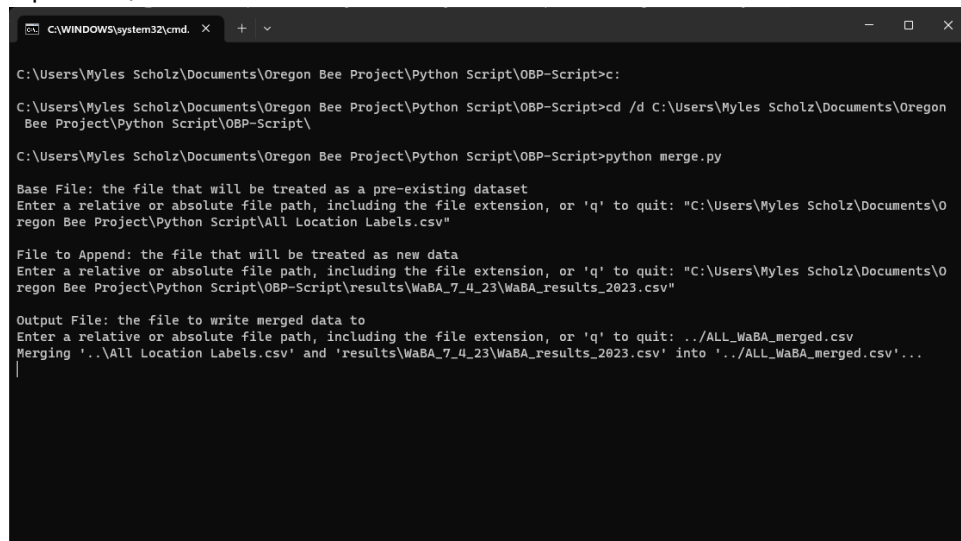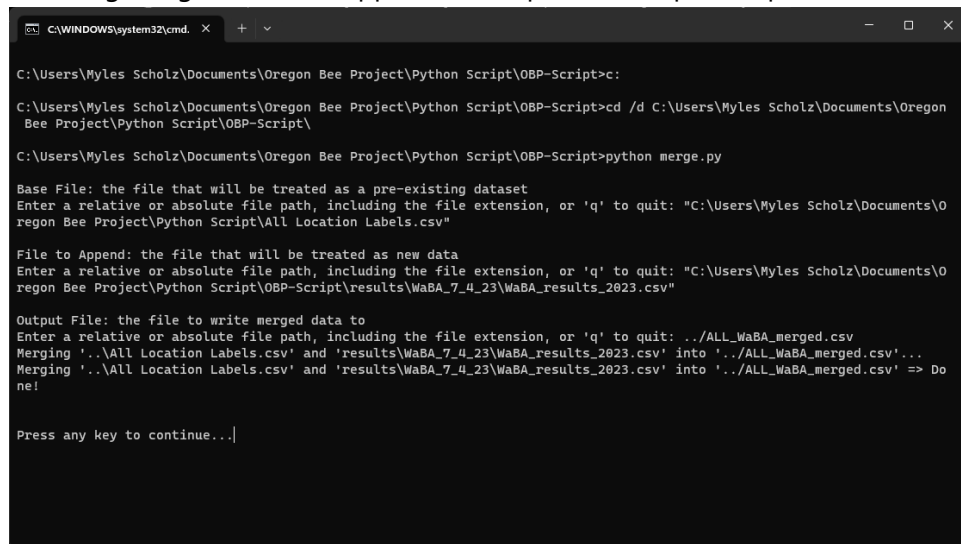
The script checks for duplicate entries by comparing each entry in the appending data to each entry in the base data. If an entry in the appending data matches in any of the following ways (checked in order), it will not be added to the merged file.
1. Observation No.: If the "Observation No." values are not empty, two entries match if their "Observation No." fields match.
2. URL, Sample ID, and Specimen ID: If the "Associated plant - Inaturalist URL" values are not empty, two entries match if their "Associated plant - Inaturalist URL", "Sample ID", and "Specimen ID" fields all match.
3. Alias, Date, Sample ID, and Specimen ID: In all other cases, two entries match if their "iNaturalist Alias", "Collection Day 1", "Month 1", "Year 1", "Sample ID", and "Specimen ID" fields all match.

Entries with empty "Dec. Lat." or "Dec. Long." fields will also not be added to the merged file.

The script will index (assign a unique number to) the data in the "Observation No." field using the format YY#####, where YY is the two-digit abbreviation of the year when the script ran and ##### are five sequentially assigned digits. The script assigns these numbers by sorting the data by
- o "Observation No."
- o Then by, "Collector - Last Name"
- o Then by, "Collector - First Name"
- o Then by, "Month 1"
- o Then by, "Collection Day 1"
- o Then by, "Sample ID"
- o Then by, "Specimen ID"

in ascending order, with blank values being put at the end. If all "Observation No." fields are blank in the merged data, the indexing will start at YY00000. Otherwise, the indexing will start at the previous largest "Observation No." plus one.

Help and Tips

- **File Paths**

  A file path is a string of characters representing the unique location of a file on a computer. It contains a sequence of slash-separated folders followed by a file name and its extension. This program accepts absolute and relative file paths and will display given file paths relative to the directory in which the program is running.

  This script accepts file paths pasted directly from File Explorer. To copy a file path in File Explorer, do the following:

  On Windows 11 and some versions of Windows 10,
  1. Navigate to the target file.
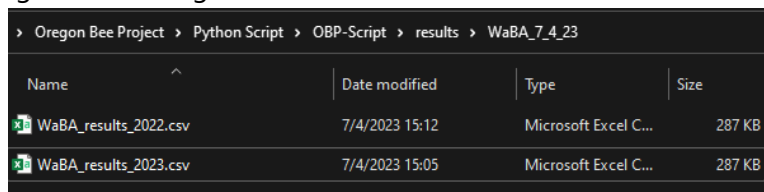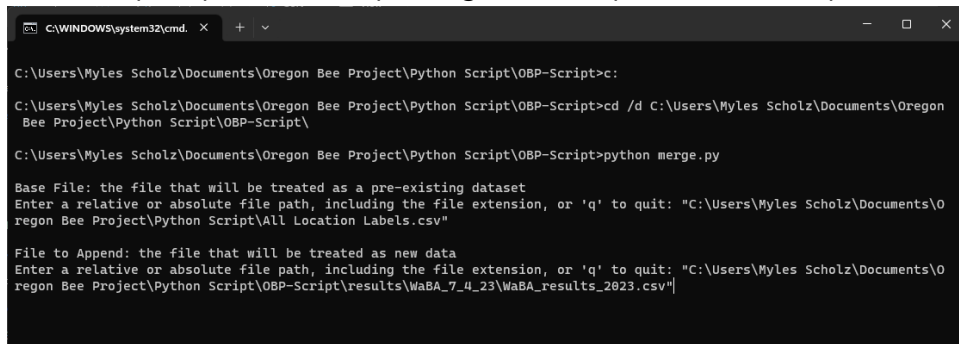
  2. Left click on the target file to select it.
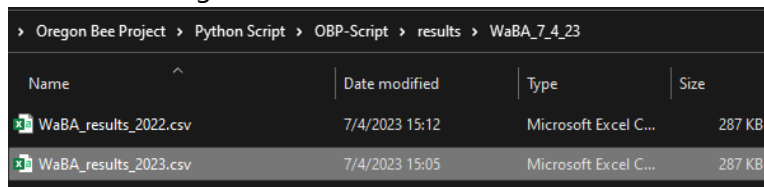
  3. Right click on the target file and click "Copy as Path", or press Ctrl + Shift + C, to copy the file path.
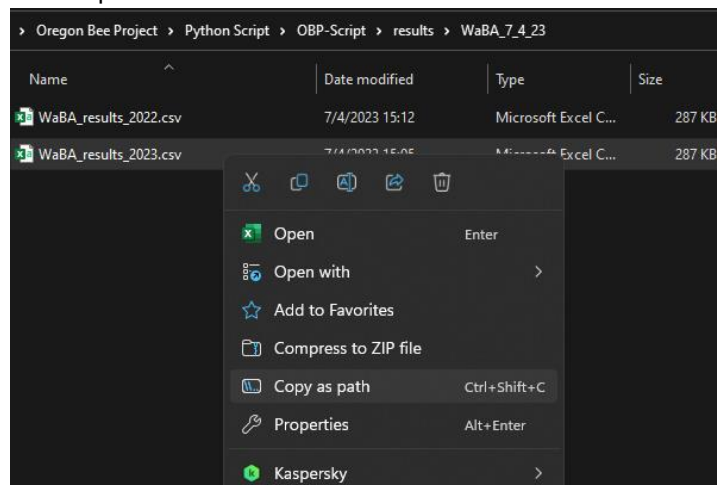
4. Click the terminal window in which Merge_Process.bat is running.
5. When prompted for a file path, right click, or press Ctrl + V, to paste the file path.

6. Press Enter to submit the file path.

On Windows 10,

7. Navigate to the target file's folder.

8. Click on the blank space in the file path bar near the top of the window. This should change the file path's display to text, which should be selected.
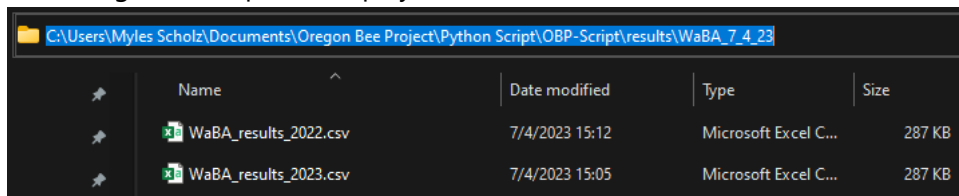
9. Press Ctrl + C, or right click and click "Copy", to copy the file path. This will only copy up to target file's parent folder.
10. Click the terminal window in which Merge_Process.bat is running.
11. When prompted for a file path, right click, or press Ctrl + V, to paste the folder path.

12. Type slash and the target file's name and extension.


```
C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>c:

C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>cd /d C:\Users\Myles Scholz\Documents\Oregon
 Bee Project\Python Script\OBP-Script\

C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>python merge.py

Base File: the file that will be treated as a pre-existing dataset
Enter a relative or absolute file path, including the file extension, or 'q' to quit: "C:\Users\Myles Scholz\Documents\O
regon Bee Project\Python Script\All Location Labels.csv"

File to Append: the file that will be treated as new data
Enter a relative or absolute file path, including the file extension, or 'q' to quit: C:\Users\Myles Scholz\Documents\Or
egon Bee Project\Python Script\OBP-Script\results\WaBA_7_4_23\WaBA_results_2023.csv
```
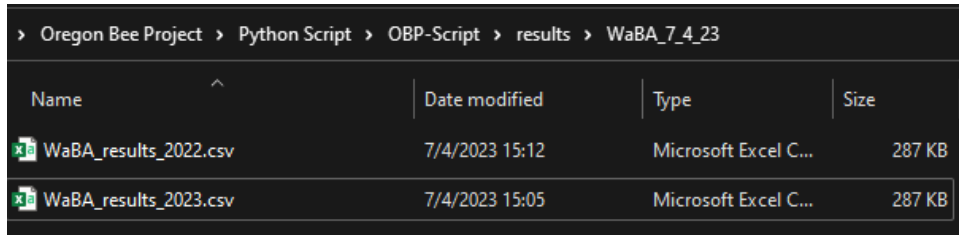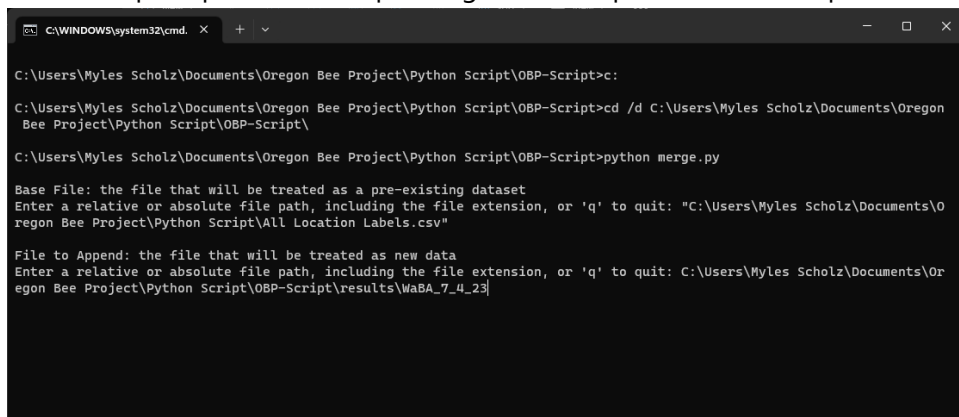
13. Press Enter to submit the file path.

- **Resolving Non-matching Header Errors (format_header.txt)**
  The "base and append file headers do not match" error is easy to trigger with this script because the program requires that the headers of its input files match *exactly*. Resolving this issue depends on which files are being used as input and whether their headers can be modified.

  If this error occurs when merging with "results" files from the data pulling and formatting process, the fix may be relatively simple. The "results" files use the header specified in OBP-Script/config/format_header.txt. This file lists the column names for the data, one on each line. To resolve the issue,
    1. Check whether the number of columns and the names of the columns match between format_header.txt and the other input file.
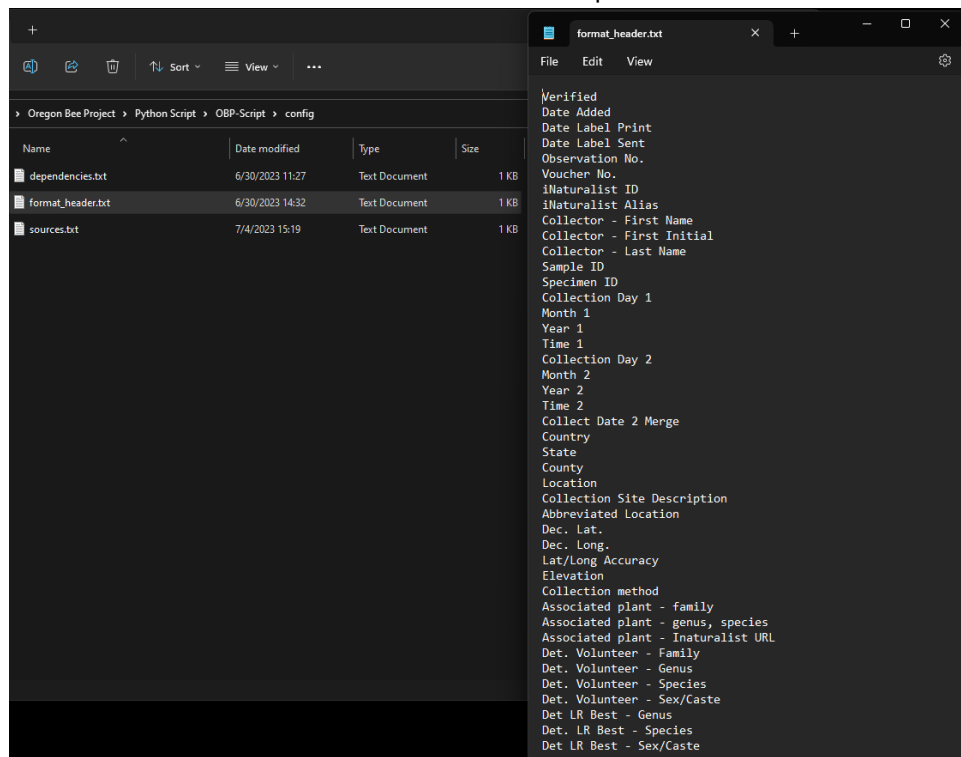
2. If the headers have the same number of columns and the columns roughly correspond but they do not match in spelling, capitalization, or spacing,
    1. Modify the column names in either format_header.txt or the other input file so the two match exactly.
    2. If format_header.txt was changed, save it, and run the data pulling and formatting process again.
    3. Run the merging process again (with the newly pulled data, if applicable).
3. If the number of columns does not match or the columns do not correspond in order, there are a couple options:
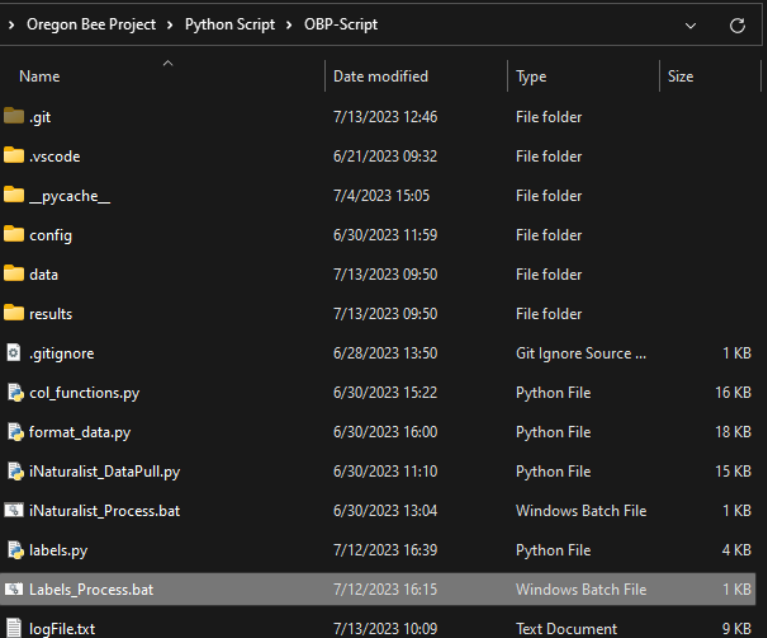   1. Manually modify the columns of either input file to match exactly. If reordering or renaming the columns, make sure that the data is reordered or rewritten in the same way.
    2. Modify the code (format_data.py) to change how data is formatted. Run the data pulling and formatting process again.
    3. After either, run the merging process again with the reformatted data.

Making Sheets of Labels from Formatted Data

This script (Labels_Process.bat) will create a PDF of labels with data from a formatted CSV file. When the user runs the script, they will be prompted for an input file path and an output file path. The input file path is required and must be a CSV file with a specific header (see "Input File Specifications" below). The output file is also required and must be a PDF file.

Running the Process
1. Execute (double-click) Labels_Process.bat.

2. When prompted, type or paste the file path of the input CSV file.
   o The program automatically trims quotation marks from the ends of the file path, so users may paste file paths copied directly from the File Explorer on Windows (see "Help and Tips" above).

3. When prompted, type or paste the file path of the output PDF file.
   o The program automatically trims quotation marks from the ends of the file path, so users may paste file paths copied directly from the File Explorer on Windows (see "Help and Tips" above).

4. Wait while the program creates the labels sheets. This may take several minutes, depending on the amount of input data.

5. The resulting PDF will appear at the specified output file path.

```
C:\WINDOWS\system32\cmd.    ×    +    ∨                                          —    □    ×

C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>c:

C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>cd /d C:\Users\Myles Scholz\Documents\Oregon
 Bee Project\Python Script\OBP-Script\

C:\Users\Myles Scholz\Documents\Oregon Bee Project\Python Script\OBP-Script>python labels.py

Input File: a formatted CSV file from which labels will be made
Enter a relative or absolute file path, including the file extension, or 'q' to quit: "C:\Users\Myles Scholz\Documents\O
regon Bee Project\Python Script\OBP-Script\test_csv.csv"

Output File: a PDF file to which labels will be written
Enter a relative or absolute file path, including the file extension, or 'q' to quit: test.pdf
Making labels from "test_csv.csv" into "test.pdf"... (This may take several minutes)
Page 1/2
100%|██████████████████████████████████████████████████████| 250/250 [04:36<00:00,  1.10s/it]
Page 2/2
100%|██████████████████████████████████████████████████████| 17/17 [00:19<00:00,  1.16s/it]
Making labels from "test_csv.csv" into "test.pdf" => Done!


Press any key to continue...
```

Input File Specifications

This script expects that the input CSV file has certain column names, which must match the following list *exactly*, excluding the quotation marks. The column names are defined by constants in the first lines of make_labels.py.

- o "Country"
- o "State"
- o "County"
- o "Abbreviated Location"
- o "Dec. Lat."
- o "Dec. Long."
- o "Elevation"
- o "Collection Day 1"
- o "Month 1"
- o "Year 1"
- o "Sample ID"
- o "Specimen ID"
- o "Collector - First Initial"
- o "Collector - Last Name"
- o "Collection method"
- o "Observation No."

Output File Specifications

The resulting PDF will include one label for each entry of the input data. The pages have the following layout:

US Letter size paper (8.5" x 11")
Portrait orientation
0.25" horizontal margins
0.5" vertical margins

25 rows of labels
10 columns of labels
Equal horizontal and vertical spacing
0.666" label width
0.311" label height

All layout values are defined as constants in the first lines of make_labels.py.