Oregon Bee Project Data Pipeline Script

Summary

These scripts retrieve observation records from iNaturalist.org, reformat them, and create bee specimen labels from them.

There are two ways to run the scripts:
1. **Full Pipeline Mode** - the program runs the full process sequentially:
    1. Pulling data from iNaturalist.org
    2. Formatting the data
    3. Merging the data with an existing dataset and indexing it
    4. Creating labels from the data (optional)
2. **Labels Only Mode** - the program creates labels from a given formatted dataset
    1. This option exists because creating labels is time-consuming, so the user may wish to run it as a separate process.

These processes can be run by executing (double-clicking) Full_Process.bat or Labels_Process.bat, respectively (Full_Process.sh and Labels_Process.sh on MacOS). See the corresponding sections below for details.

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.

- o 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.

```
Command Prompt                                                    —    □    ×

Microsoft Windows [Version 10.0.22621.1992]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Myles Scholz>python --version
Python was not found; run without arguments to install from the Microsoft Store, or disable this shortcut from Settings
> Manage App Execution Aliases.

C:\Users\Myles Scholz>
```
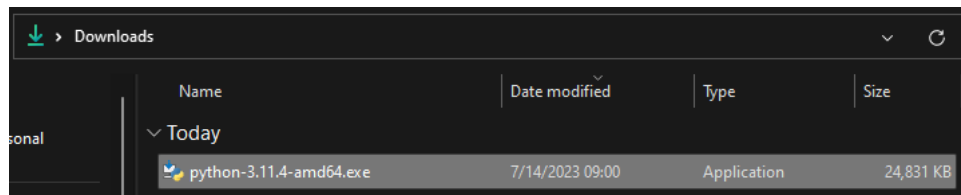
3. Go to https://www.python.org/downloads/ and click "Download Python 3.*.*".
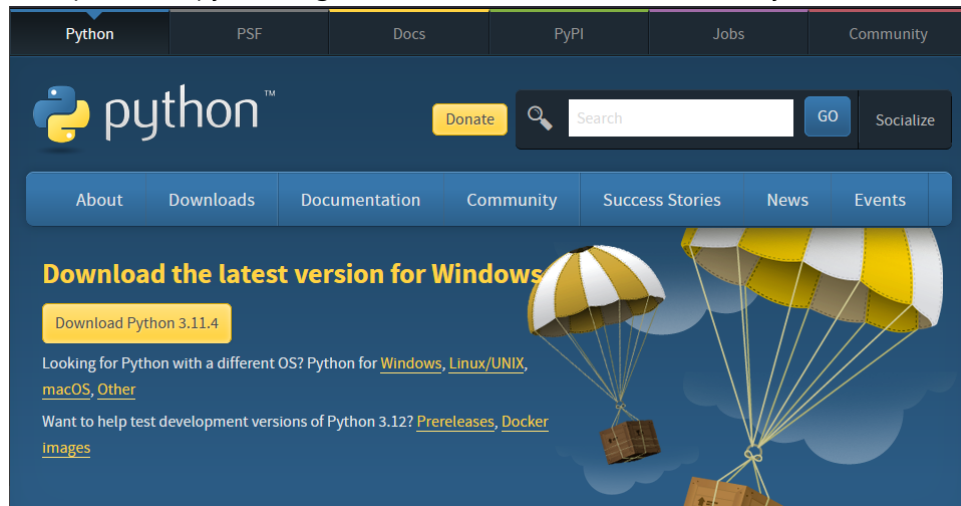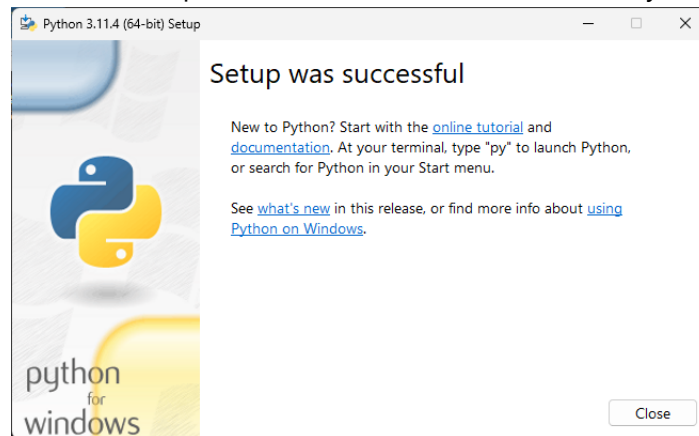

4. Go to the folder where the EXE file downloaded and execute (double-click) it.

- o Follow the installation instructions.
- o Check the box marked "Add python.exe to PATH variable" when prompted.
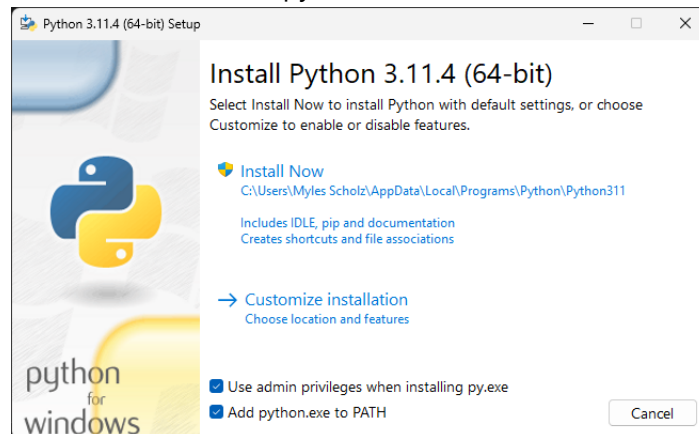
5. When the installation is complete, continue to the next section ("Python Libraries").
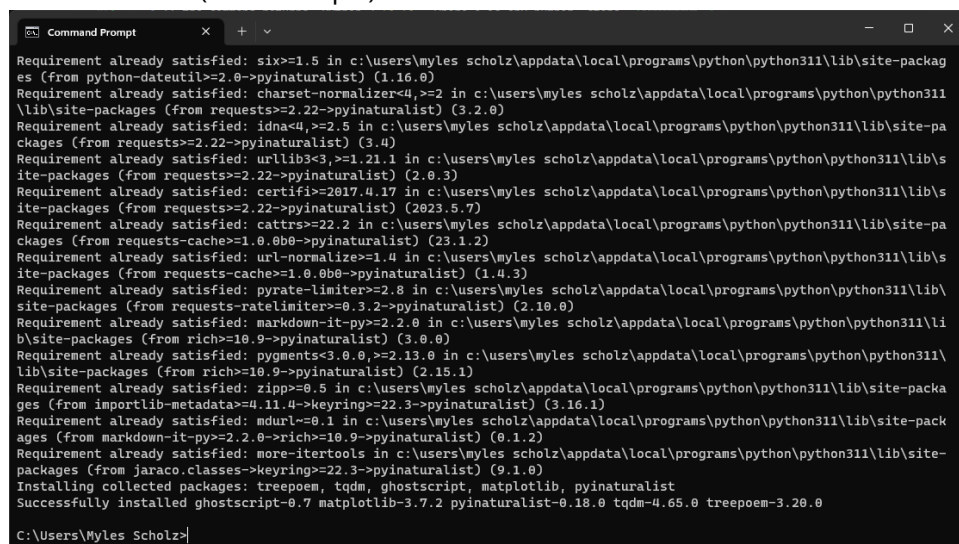
Python Libraries

Next, the scripts need some external Python libraries installed.

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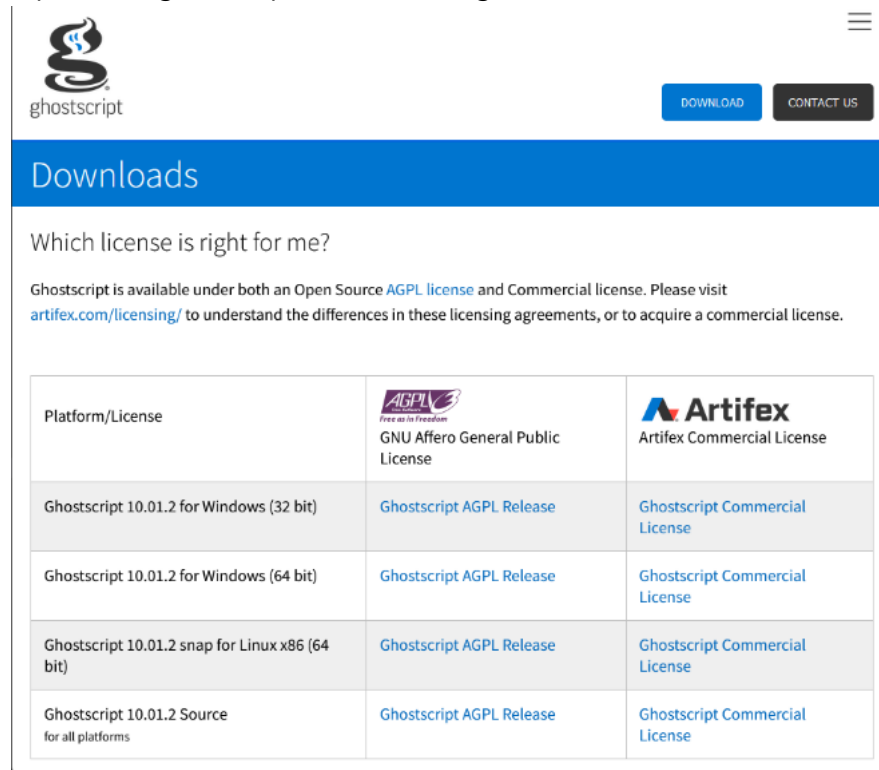2. Type "pip install pyinaturalist matplotlib treepoem ghostscript tqdm" and press Enter.

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.
   - 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.
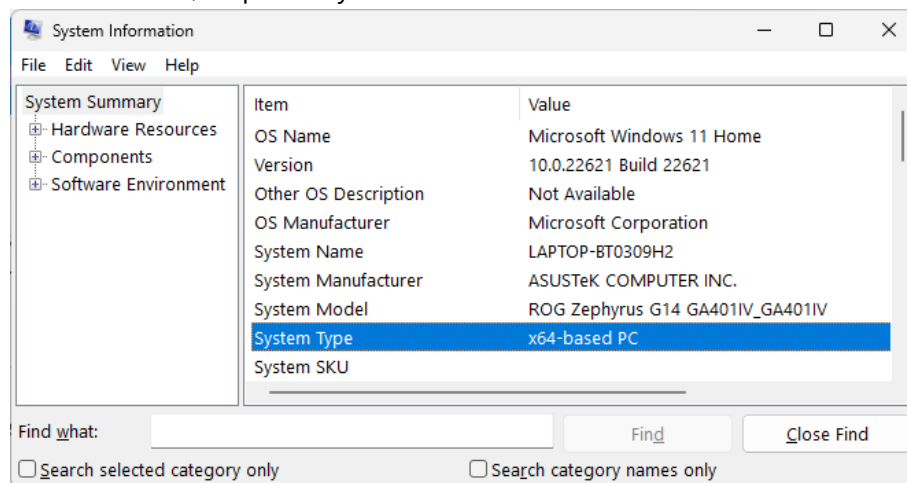
3. Go to the folder where the EXE file downloaded and execute (double-click) it.
   o This will require administrator privileges on the computer.
   o Follow the installation instructions without changing any of the options.
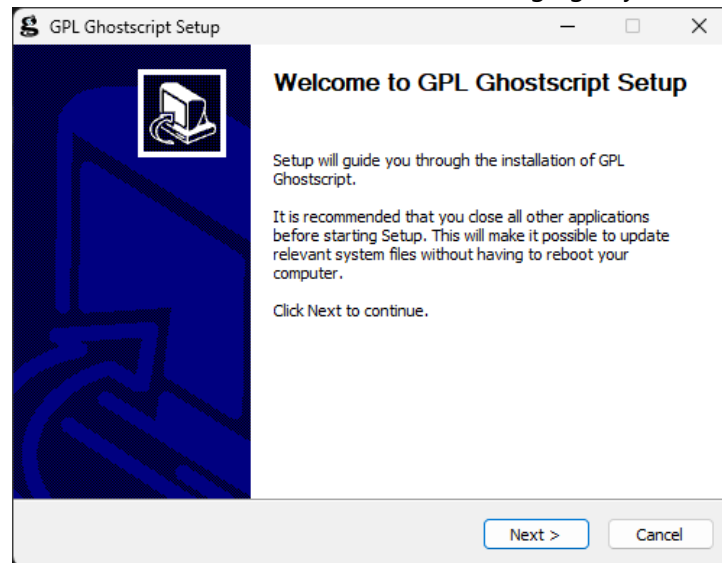
4. Wait until installation finishes.
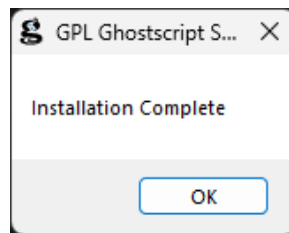
After completing the above steps, the scripts should be able to run on the computer.

Full Pipeline Mode

This script (Full_Process.bat/Full_Process.sh) executes the full data pipeline in four steps:
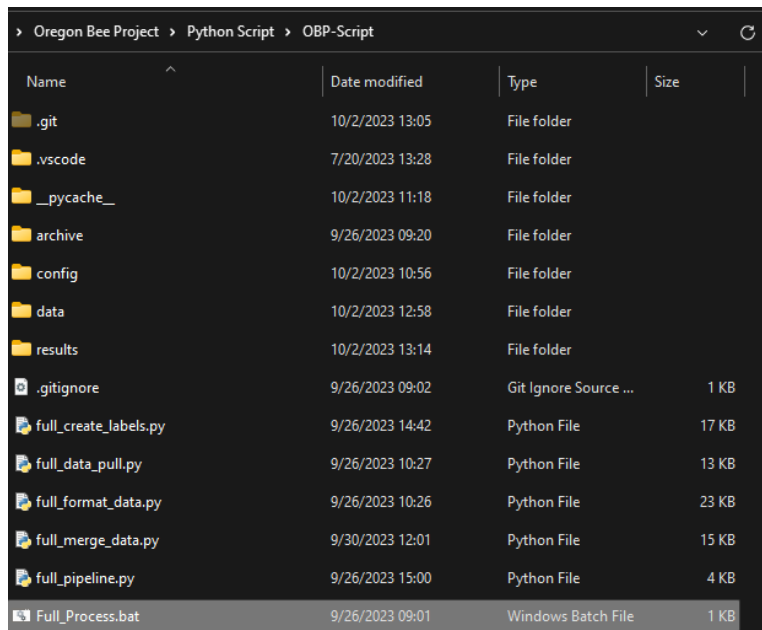1. Pulling data
2. Formatting data
3. Merging data
4. Creating labels

The first three steps always run in Full Pipeline Mode and cannot be paused. The user has the option to run the fourth (label creation) step or end the process after the first three steps. No information will be lost if the user ends the process before creating labels.

Running the Process

0. Check that the script is configured properly. Each step has a configuration file in OBP-Script/config/. See the respective section below for details on how to configure them.

1. Execute (double-click) Full_Process.bat (Full_Process.sh on MacOS).

2. A terminal will open and the program will run. When prompted, enter the requested information. See each step's section below for details on answering the prompts.

3. The locations of the output files for each step are detailed in the corresponding section below.

Step 1: Pulling Data from iNaturalist.org

The pipeline begins by querying 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.csv (see "Data Pulling Configuration" below).

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

Data Pulling Configuration

The script pulls data from a list of iNaturalist projects specified in OBP-Script/config/sources.csv. The file is in a CSV format with three columns. They are, in order:

1. Name: the name of the source as it will appear in the terminal.
2. ID: the iNaturalist project ID; this is essential for pulling data from the correct sources.
3. Abbreviation: the abbreviation of the source's name; this must be unique.
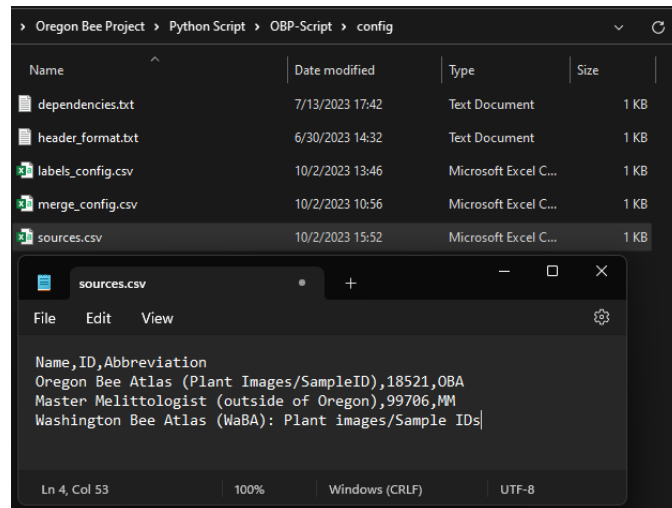
To add a source, do the following:

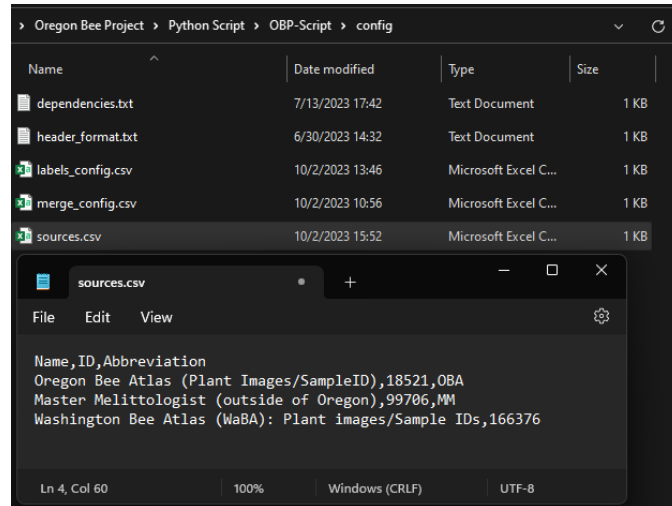1. Open OBP-Script/config/sources.csv in a text editor or Excel.
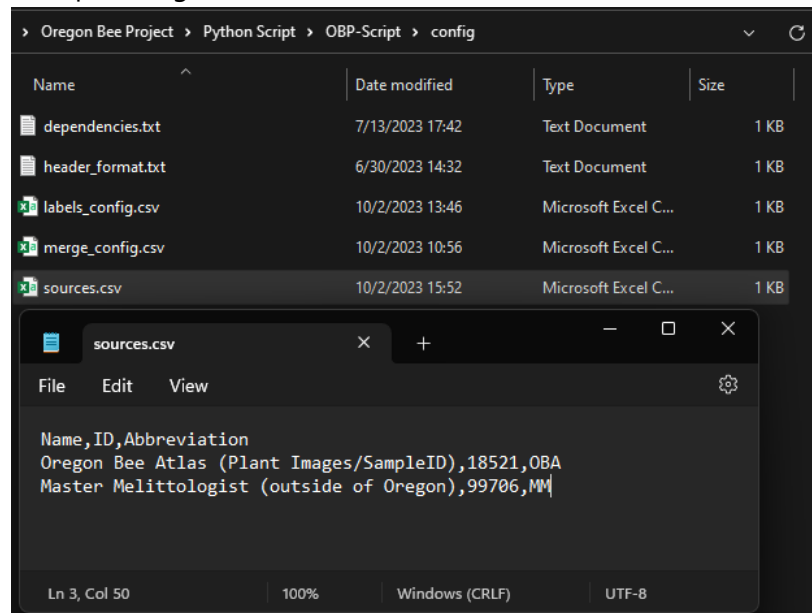
2. Add a line at the end of the file, or select the next blank cell in the A column if using Excel.
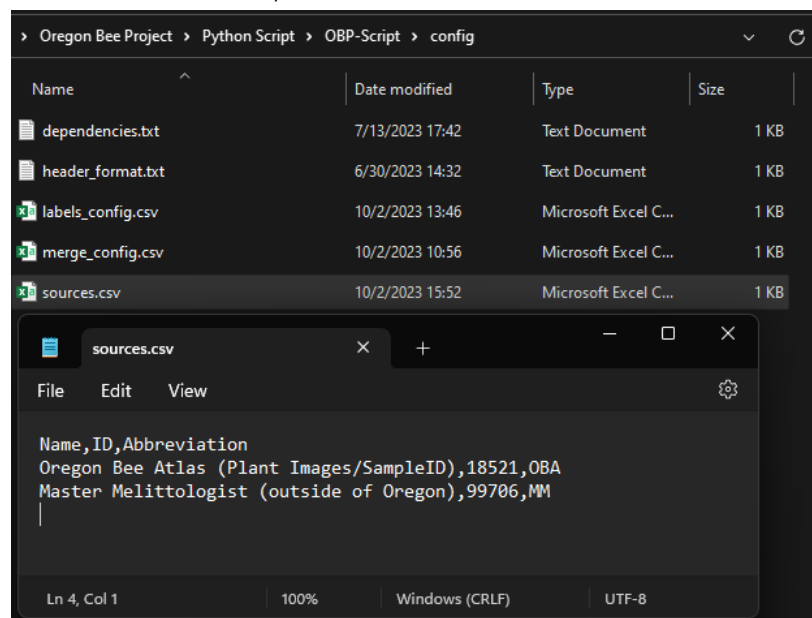
3. Type a name for the source. The name cannot contain commas, but all other keyboard characters are allowed.
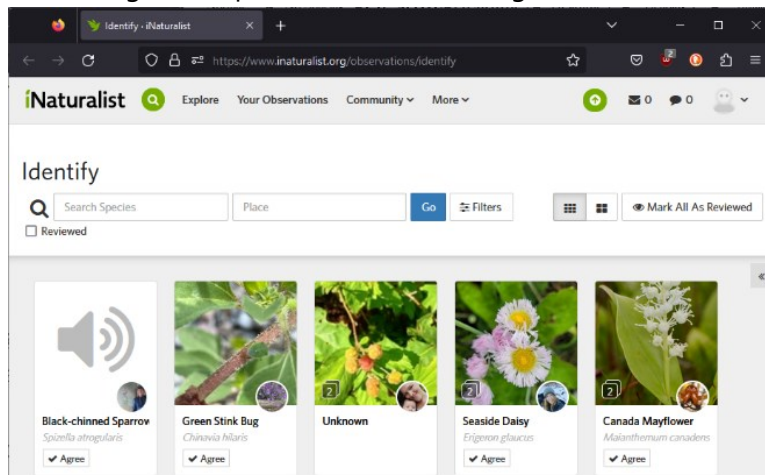

4. Type a comma (or move to B column), and then type the iNaturalist project ID.

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

1. In a browser, go to https://www.inaturalist.org/observations/identify.

2. Click the "Filters" button.

3. Click the "More Filters" button.

4. Type the name of the project in the "Project" field.
   - Be sure to select the project from the results drop-down menu. The full project name should appear in green.

5. The project ID will appear as a string of about five numbers at the end of the browser's URL bar.
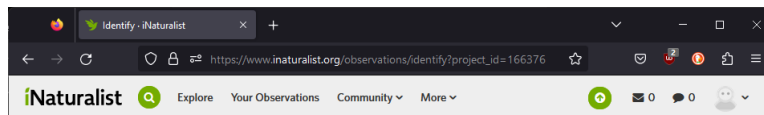
5. Type a comma (or move to the C column), and then type a unique abbreviation for the source.
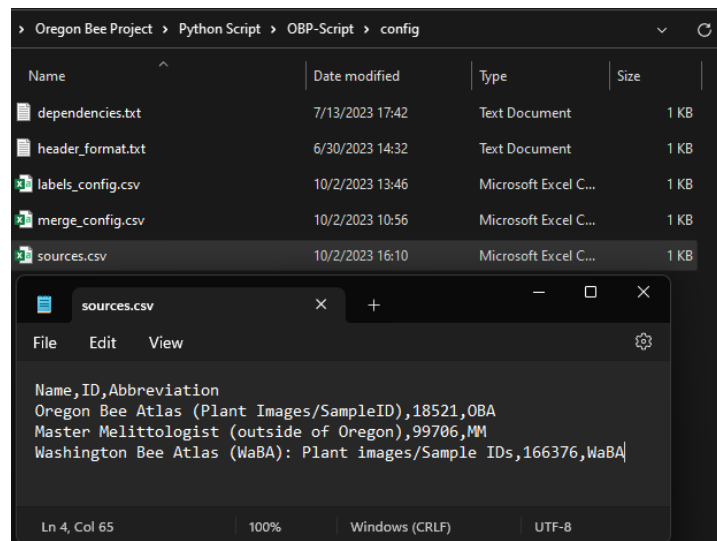
6. Save and close sources.csv.

To remove a source, do the following:
1. Open OBP-Script/config/sources.csv in a text editor or Excel.

2. Select and delete the line (or row) containing the source to be removed.

3. Save and close sources.csv.

Data Pulling Prompts

At the beginning of this step, the program will prompt the user for a year with which to query iNaturalist.org. It accepts four-digit years less than or equal to the current year. Type a number of this format and hit Enter to continue. There are no other prompts in this step.
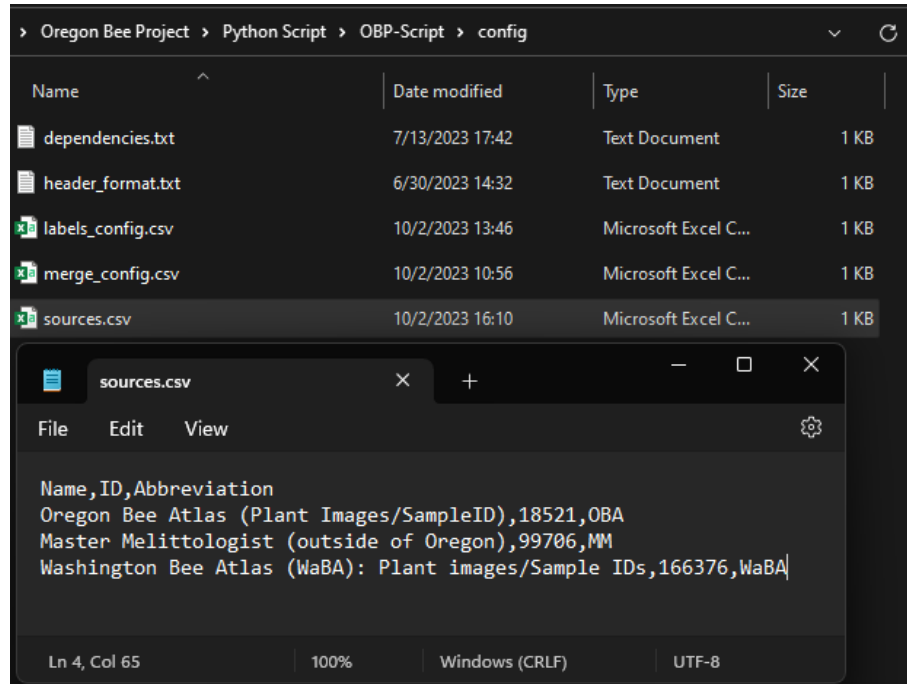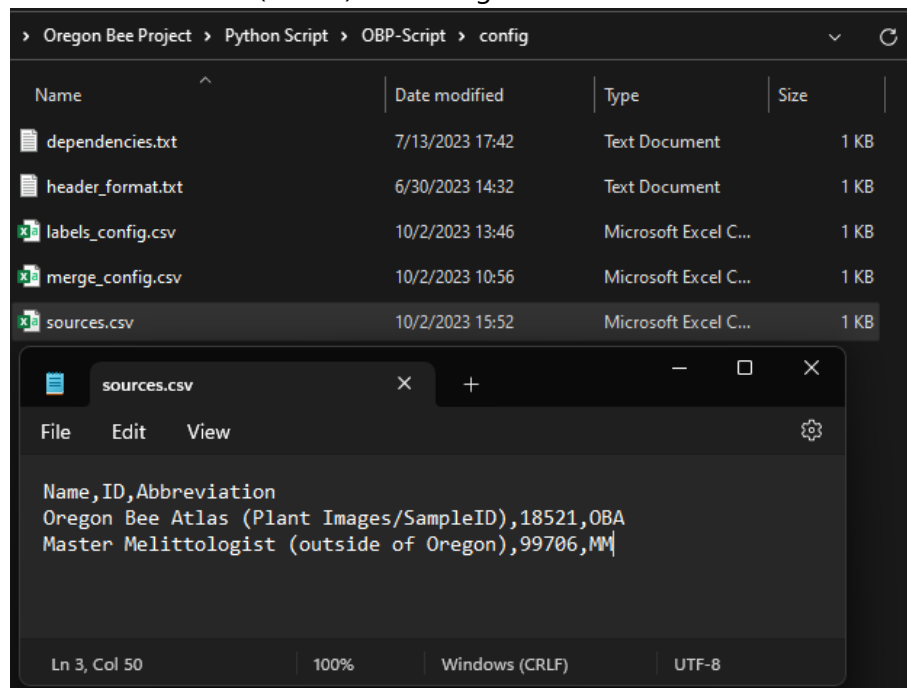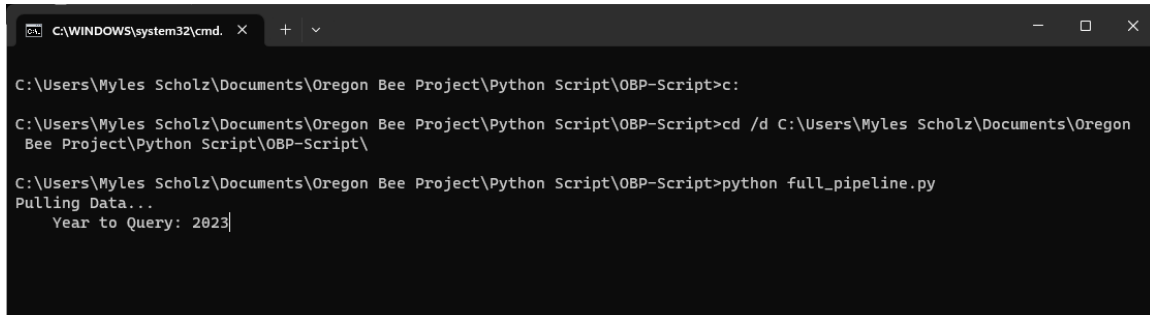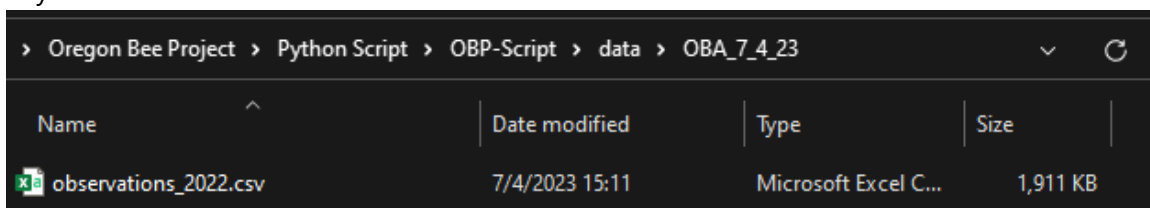

Data Pulling Output

The data pulling step outputs a minimally formatted CSV file in a source-specific folder under OBP-Script/data/.

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 4th, 2023 would appear in OBP-Script/data/OBA_7_4_23/.

The resulting CSV file will be named according to the format observations_YYYY.csv, where YYYY is the year that was queried. For example, querying Oregon Bee Atlas data from 2022 would produce a file named observations_2022.csv.

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

Step 2: Formatting Data

The second step of the pipeline is to format the data that was pulled from iNaturalist.org previously. This step is entirely fixed without modifying the code, so it does not need user input or configuration. Nonetheless, this step's configuration file, header_format.txt, is explained below.

Data Formatting Configuration

The program formats the data into a CSV file with column names specified in OBP-Script/config/header_format.txt. The name of each column appears in order on each line of the file. For the merging step to work, these column names must match those of the input dataset for that step exactly.
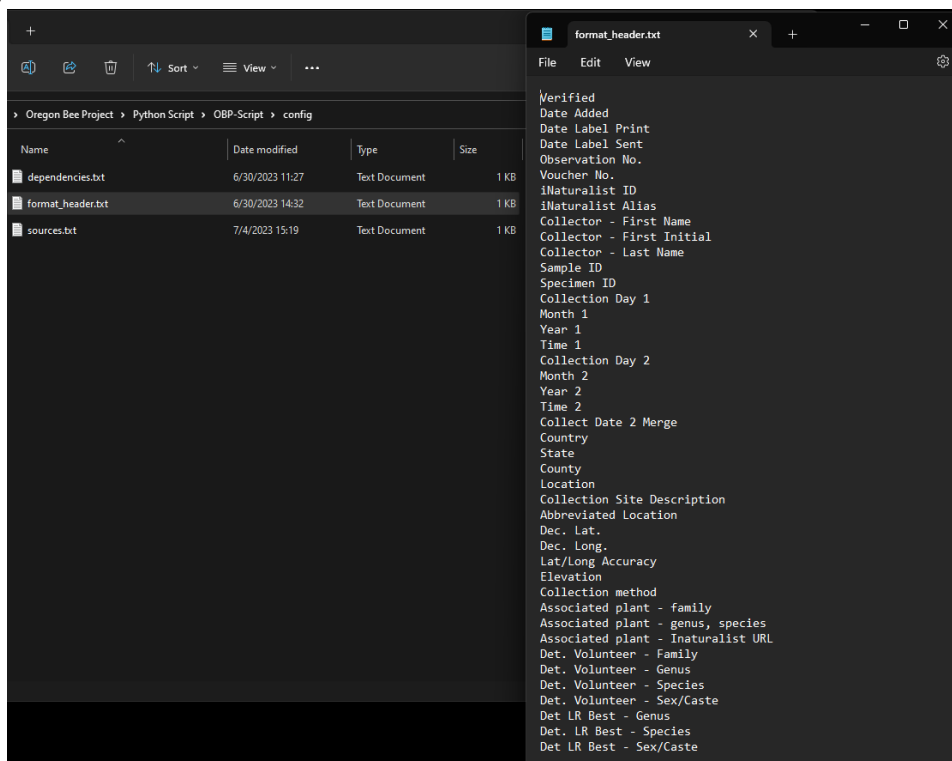
Data Formatting Prompts
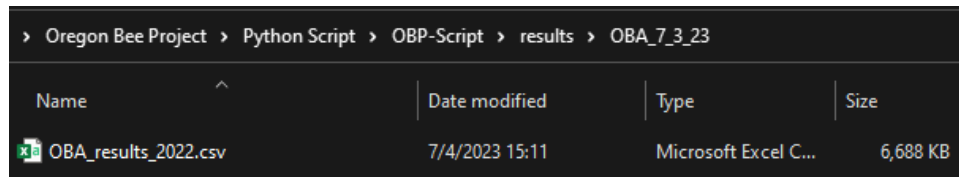
There are no user prompts for this step.

Data Formatting Output

The data formatting step outputs a CSV file in a source-specific folder 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 formatting 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, formatting 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 with the same query year, the output file will be entirely overwritten with new data.


Step 3: Merging and Indexing Formatted Data

In this step, the program will combine formatted data from the previous step with an existing dataset of the same format. It will detect duplicate entries and sort and index the output dataset. The input and output file paths for this step are defined in OBP-Script/config/merge_config.csv (see "Data Merging Configuration" below). These files must be CSV files, and the input file must have the exact header specified in OBP-Script/config/header_format.txt. There are special options to prompt the user for these files through a file select dialog.

Data Merging Configuration

The input and output file paths for the data merging step are specified in OBP-Script/config/merge_config.csv. This file is in a CSV format with two columns:
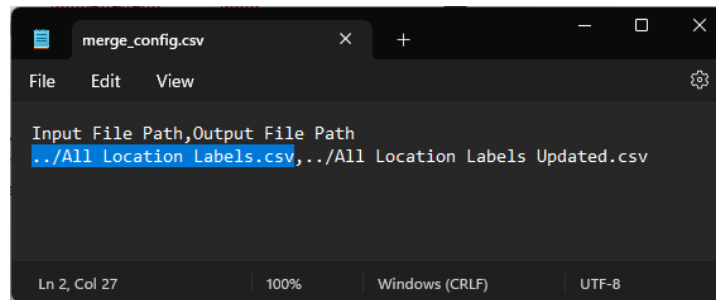
1. Input File Path: the relative or absolute file path of a formatted dataset to merge new data into. This must be a CSV file and must have the exact column names listed in OBP-Script/config/header_format.txt.
2. Output File Path: a relative or absolute file path where the resulting merged dataset will be saved. This must be a CSV file and can be the same as the Input File Path, but the original dataset will be overwritten.

To set the input and output file paths, do the following:
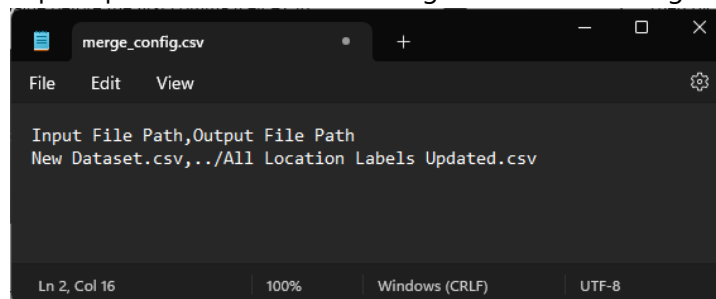1. Open OBP-Script/config/merge_config.csv in a text editor or Excel.
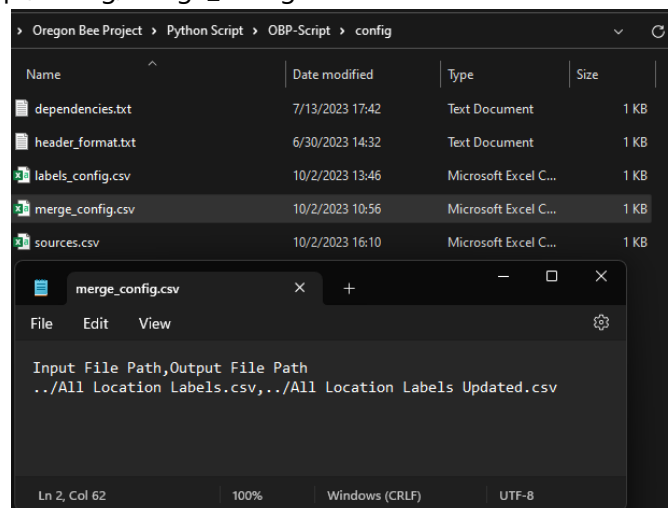
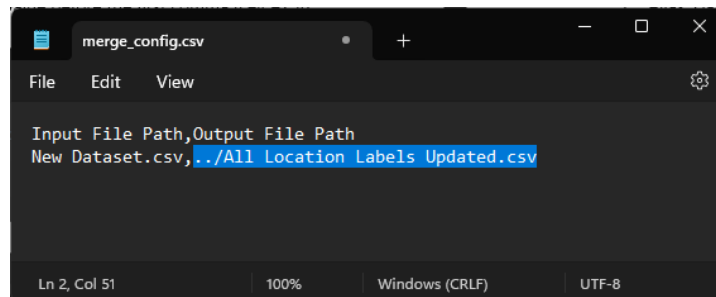2. On the first line below the column names, select the value before the first comma (cell A2 in Excel).

3. Type the new value for the Input File Path. The value "Select" (not case-sensitive) will cause the program to prompt the user for a file through a file select dialog.

4. On the same line, select the value after the first comma (cell B2).

5. Type the new value for the Output File Path. The value "Select" (not case-sensitive) will cause the program to prompt the user for a file through a file select dialog.
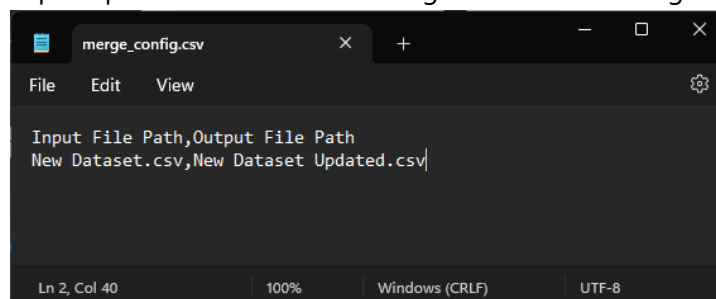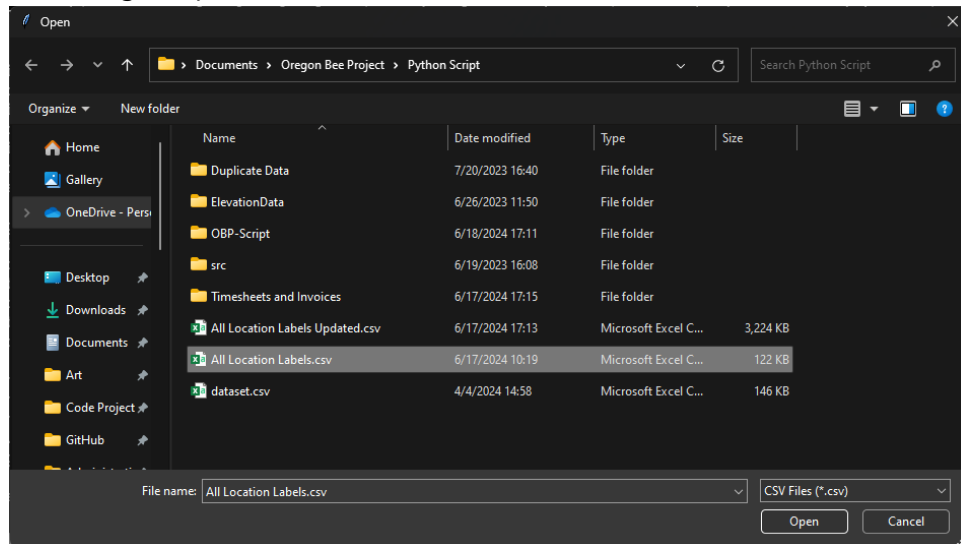

6. Save and close merge_config.csv.

See "Help and Tips - File Paths" for information about file paths.

Data Merging Prompts

If the Input File Path is set to "Select", the program will prompt the user at the beginning of this step through an "Open" file select dialog. Navigate to the desired file and click "Open" to choose the input file for the merge step.

If the Output File Path is set to "Select", the program will prompt the user through a "Save As" file select dialog. Navigate to the desired directory, type the name of the output file in the "File name" box, and click "Save".
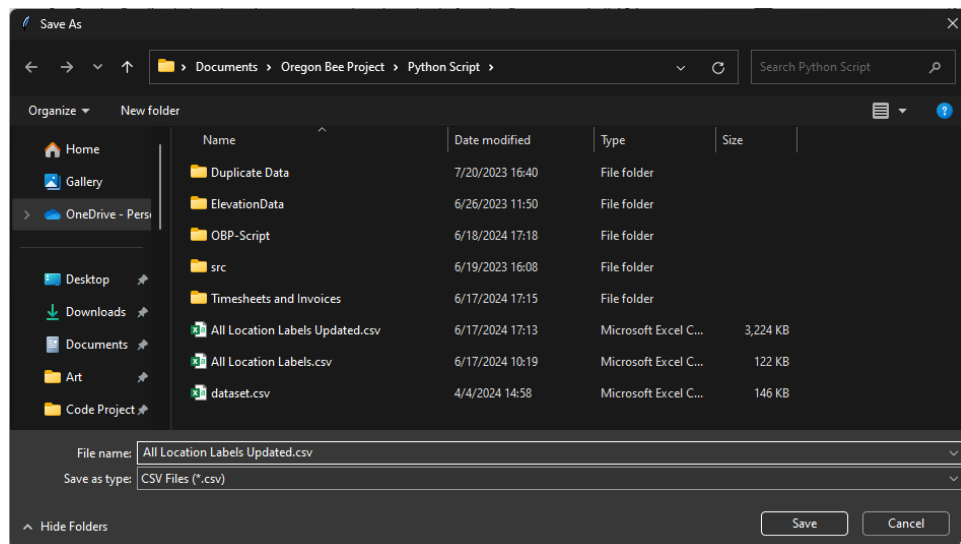
The program will automatically save these values as relative file paths in OBP-Script/config/merge_config.csv for future executions.

Data Merging Output

The data merging step outputs the resulting merged dataset to the file path specified in its configuration. The program checks the data for duplicate entries and adds indices to new data.

It checks for duplicate entries by comparing each entry in the appending data to each entry in the base dataset. 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 program 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
1. "Observation No."
2. Then by, "Collector - Last Name"
3. Then by, "Collector - First Name"
4. Then by, "Month 1"
5. Then by, "Collection Day 1"
6. Then by, "Sample ID"
7. 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.

Step 4: Creating Labels from Formatted Data

This step will create a PDF of labels with formatted data from the previous steps or from a CSV file, depending on how it was executed. If this step is reached in Full Pipeline Mode, the user will have the opportunity to check the data before the program begins creating labels. If this step is reached in Labels Only Mode, the user will need to provide a formatted CSV dataset as input. See "Label Creation Prompts" below for details.

Label Creation Configuration

The label creation step will output the labels to a PDF file path specified in OBP-Script/config/labels_config.csv. The configuration file also specifies the default row number in the input dataset from which to create labels. This value is set by the merging process and is optional (see "Label Creation Prompts" below), so editing it is not recommended.
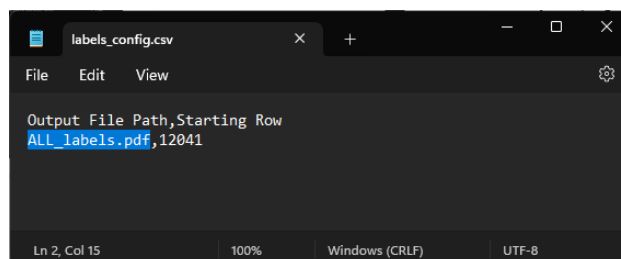
1. Output File Path: a relative or absolute file path where the labels will be output. This must be a PDF file.
2. Starting Row: the zero-indexed row number of the input dataset from which to create labels. It is not recommended for the user to change this manually.

To set the Output File Path, do the following:

1. Open OBP-Script/config/labels_config.csv in a text editor or Excel.
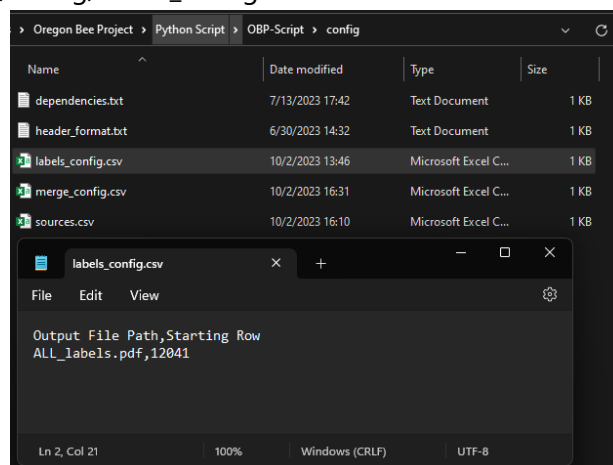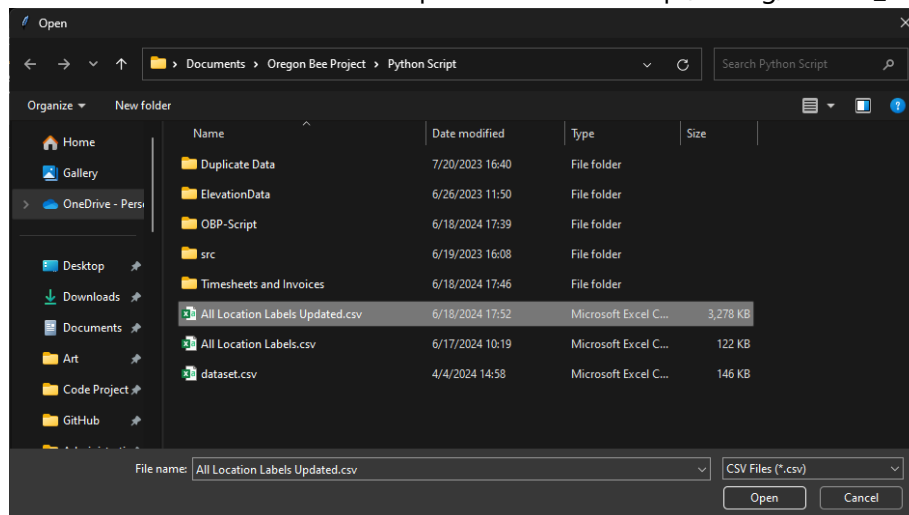

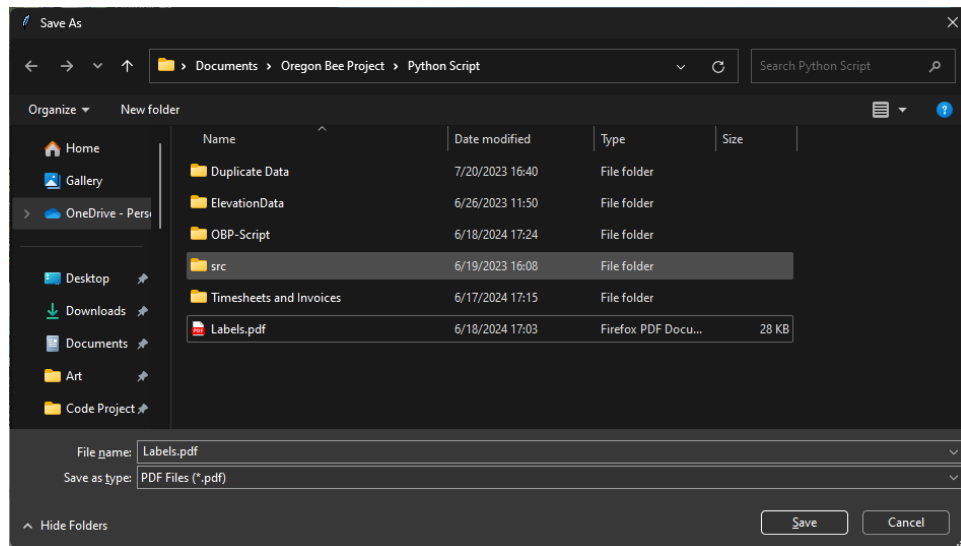2. On the first line after the column names, select the value before the first comma (cell A2 in Excel).

3. Type the new value for the Output File Path. The value "Select" (not case-sensitive) will cause the program to prompt the user for a file through a file select dialog.

4. Save and close labels_config.csv.

Label Creation Prompts

The prompts for the label creation step begin differently depending on whether the program is in Full Pipeline Mode or Labels Only Mode.

In Full Pipeline Mode, after the merging step is complete, the user will be prompted to check the dataset and respond affirmatively to proceed with label creation. It is recommended to look over the data in the output file of the merging step (located at the Output File Path in OBP-Script/config/merge_config.csv). When ready, type Y or y and Enter to continue. If the user wishes to end the program, they may type any other character (or no characters) and Enter to do so.

In Labels Only Mode, when the user runs the script, an "Open" file select dialog will appear to choose an input dataset. Navigate to the desired file and click "Open" to select it. The input dataset should be a formatted CSV file with the exact header specified in OBP-Script/config/header_format.txt.

If the Output File Path is set to "Select", the program will next prompt the user through a "Save As" file select dialog. Navigate to the desired directory, type the name of the output file in the "File name" box, and click "Save".

After the above prompts, Full Pipeline Mode and Labels Only Mode have the same prompts.

The program will next prompt the user for a starting row. This is an integer value representing the first row of the input dataset from which labels will be made. This allows the labels to be made from only part of the input dataset to avoid redundancy and save time. The user may type a number between 0 and the total length of the dataset and press Enter to specify a starting row.

If the user presses Enter without typing a number, the default value will be used. The default value for the starting row is the first new row. It is set by the data merging step and stored in OBP-Script/config/labels_config.csv, so in Labels Only Mode, it may not actually represent new data.

Lastly, the program will prompt the user for an ending row. This is an integer value that represents the last row of the input dataset from which labels will be made. The user may type a number between the starting row and the total length of the dataset and press Enter to specify an ending row.

If the user presses Enter without typing a number, the default value will be used. The default value for the ending row is the last row of the input dataset.

Label Creation Output

The label creation step produces a PDF file at the file path specified in its configuration. The resulting PDF will include one label for each entry of the input data within the given starting and ending row range. 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 full_create_labels.py.

Help and Tips
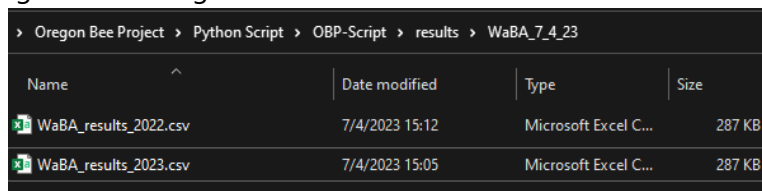
- **File Paths**
  A file path is a string of characters representing the unique location of a file on a computer. It contains a sequence of slash-separated folders followed by a file name and its extension. This program accepts absolute and relative file paths and will display given file paths relative 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:
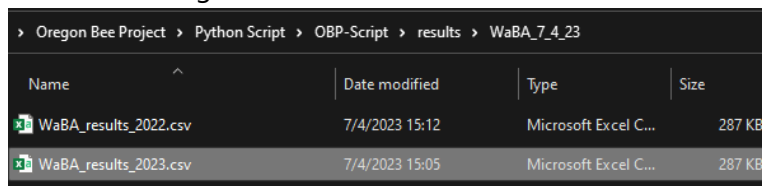
  On Windows 11 and some versions of Windows 10,
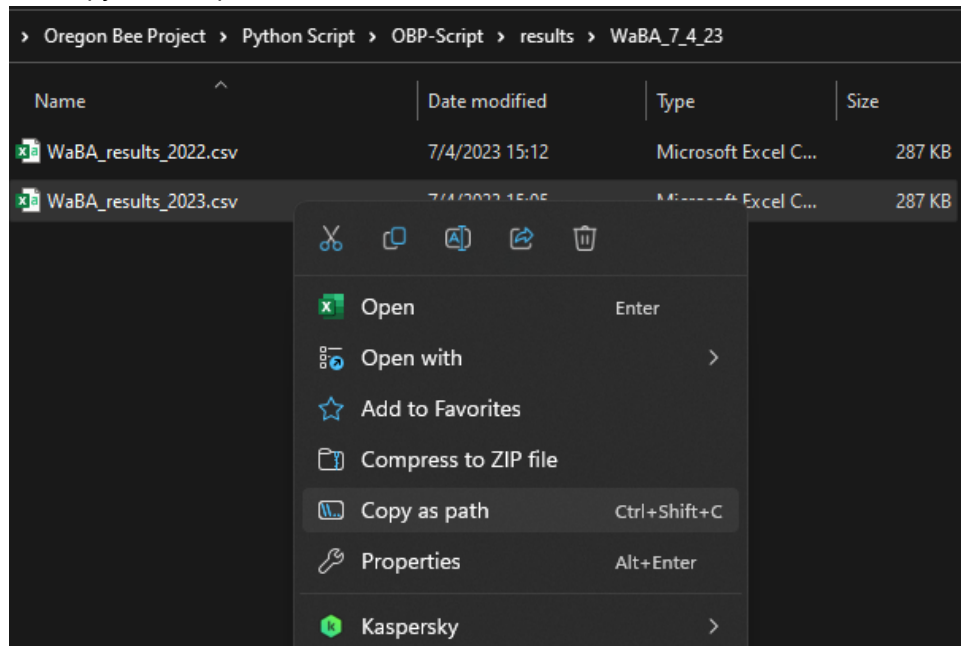   1. Navigate to the target file.

    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.
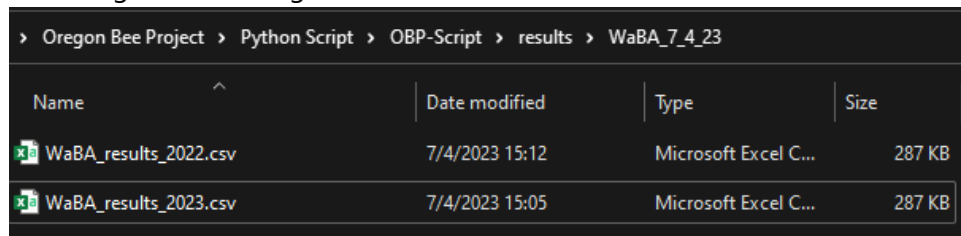

4. Click the terminal window in which the script is running.
5. When prompted for a file path, right click, or press Ctrl + V, to paste the file path.
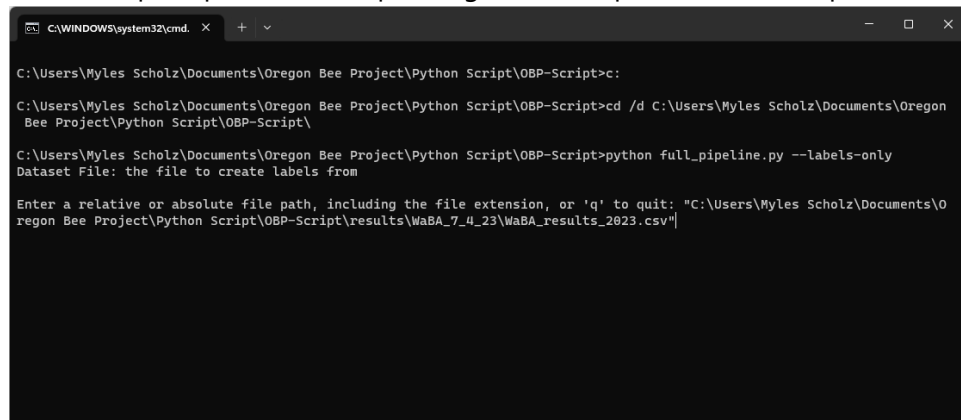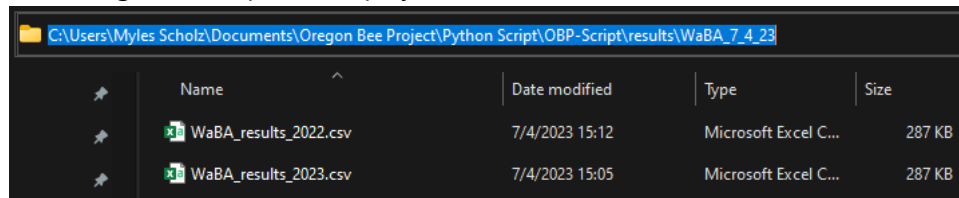

6. Press Enter to submit the file path.

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 the script 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.

13. Press Enter to submit the file path.