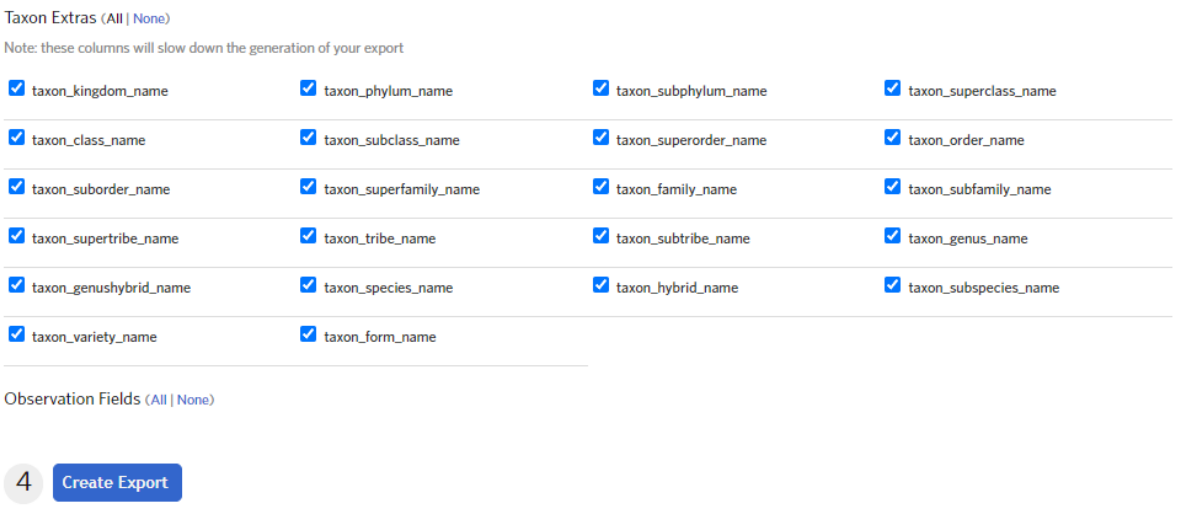
Using Python Code to Create Point Shapefiles from iNaturalist Observation Data Tables

Obtaining the Data Table:

1. If you've already done this, you can skip to the next section. Otherwise, continue. It is important when using the code to generate point shapefiles that the table you use is organized the same way as the one used when the code was made. Otherwise the code will likely not work. Go to the iNatualist project at https://www.inaturalist.org/projects/van-cortlandt-park-biodiversity-project?tab=observations and click on the "Export Observations" button.
2. Leave the first sections as default.
3. Scroll down to the "Taxon Extras" section and make sure to select all options like in image 1. Then select "Create Export."

Image 1:



4. It may take a very long time sometimes for the table to be exported. Select the "receive an email with your data" button and allow the export to load until it is ready.
5. When the export is ready in iNaturalist or has been emailed to you, download the table and save it as a .csv file. This table has data on all of the current VCP iNaturalist Biodiversity Project species observations since the date of exporting. This process will need to be repeated to update the table with new observations in the future.

Opening Code in Python:

6. Open ArcMap.

7. Select “Geoprocessing” in the top menu bar. Then click on “Python” to open a small, blank window for the program to be put into.
8. Right click on the white space in the new window and select “Load…” Then find the saved code you are using and open it. This will fill the new window with the code you selected.

Changing File Locations in the Code:

9. You will only need to do these steps once for each new computer the code is used on. If you’ve already done this once on your computer, you can skip to the next section.
10. You need to change the file locations and file names for the items needed for this process since they will likely be different from the ones on the code maker’s computer. This can be done in the python window in ArcMap.
11. Image 2 is the python window expanded. What is highlighted in green are the file locations in the code that need to be changed. There are three that need to be changed. The first one should be changed to the file location where you have the csv table downloaded from iNaturalist saved. The second one should be changed to the file location where you wish to save the new point shapefile the code will create. Lastly, the third one needs to be changed to the file location where the point shapefile was saved (same as the second highlighted file location).
12. Make sure everything that comes before and after the changes you made stays exactly the same. This means there should still be an r’ in the front and a \\ at the end of every new file location.
13. Save the code so you don’t have to change these every time. Do this by right clicking on anywhere in the white space of the python window. Then click “Save as…” Choose where you want to save this code to use again in the future. I don’t recommend overwriting just incase a mistake was made and you need to start over.

Changing Variables in the Code:

14. You now need to change the variables in the beginning of the code. These are used to tell the program what species/taxon group you want it to make a shapefile for.
15. Image 2 shows what needs to change here in yellow highlighting.
16. The yellow highlights in the first line need to be changed to what data in the iNaturalist table you want to make the point shapefile for. The first highlight in this line is for the column of the table. You will need to change the highlighted portion to the correct taxon column. So if you are interested in an order of organisms, this will be changed to “order”. In the other highlighted part of the first line, you need to change it to the name of the organism/group that is found in the data under the table column of interest. This will be what the points in the shapefile are showing. So this could be changed to something like “Hymenoptera”. So for example, this first line may be changed to read:

taxon_query = "\"taxon_order_name\" = 'Hymenoptera"

This will make a point shapefile showing all observations of Order Hymenoptera. Every change in this part needs to match exactly what is in the table columns and data, including capitalization, symbols, spelling, etc. If not, the code will not work.

17. The second line with yellow highlighting is what names the new shapefile. Here you can change the highlighted portion to what you want the shapefile to be named in your computer.
18. The third line with yellow highlighting tells the program what the name of the iNaturalist table is that you are using. You will not need to change this every time. You only need to change it if it is your first time using the code, or the name of the table has changed. Change the highlighted portion to be the name of the table you want to use from your computer. Again, make sure this table is saved as a csv.

Running the code:

19. You can now run the code to make the shapefile. Scroll down to the bottom of the python window and click the cursor on the end of the last line of the code. Hit the enter key and the code will start to make the shapefile. This will take a moment.
20. You should now have a point shapefile of the organism/s specified in the variables.
21. The code may take a while to set up and get used to, but will speed up the process a lot.

Image 2:

```
Python
>>> # variables
... taxon_query = "\"taxon_kingdom_name\" = 'Fungi'"
... shapefile_name = "Kingdom_Fungi2"
... observation_table = "observations_Nov2021"
... # code
... mxd = arcpy.mapping.MapDocument("CURRENT")
... df  = arcpy.mapping.ListDataFrames(mxd, "Layers")[0]
... addtable = arcpy.mapping.TableView(r'C:\Users\Joshua\Documents\Lehman College
\VanCortlandt_Biodiversity_Project\Inaturalist_Data\\'+ observation_table +".csv")
... arcpy.mapping.AddTableView(df, addtable)
... arcpy.RefreshTOC()
... arcpy.MakeXYEventLayer_management(observation_table, 'longitude', 'latitude', 'XY_Layer', "GEOGCS
['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROID['GRS_
1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]];-400 -400
1000000000;-100000 10000;-100000 10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision", '#')
... arcpy.MakeFeatureLayer_management('XY_Layer', 'Taxon_Layer', taxon_query, '#', 'id id VISIBLE
NONE;observed_on_string observed_on_string VISIBLE NONE;observed_on observed_on VISIBLE
NONE;time_observed_at time_observed_at VISIBLE NONE;time_zone time_zone VISIBLE NONE;user_id user_id
VISIBLE NONE;user_login user_login VISIBLE NONE;created_at created_at VISIBLE NONE;updated_at
updated_at VISIBLE NONE;quality_grade quality_grade VISIBLE NONE;license license VISIBLE NONE;url
url VISIBLE NONE;image_url image_url VISIBLE NONE;sound_url sound_url VISIBLE NONE;tag_list tag_list
VISIBLE NONE;description description VISIBLE NONE;num_identification_agreements
num_identification_agreements VISIBLE NONE;num_identification_disagreements
num_identification_disagreements VISIBLE NONE;captive_cultivated captive_cultivated VISIBLE
NONE;oauth_application_id oauth_application_id VISIBLE NONE;place_guess place_guess VISIBLE
NONE;latitude latitude VISIBLE NONE;longitude longitude VISIBLE NONE;positional_accuracy
positional_accuracy VISIBLE NONE;public_positional_accuracy public_positional_accuracy VISIBLE
NONE;geoprivacy geoprivacy VISIBLE NONE;taxon_geoprivacy taxon_geoprivacy VISIBLE
NONE;coordinates_obscured coordinates_obscured VISIBLE NONE;positioning_method positioning_method
VISIBLE NONE;positioning_device positioning_device VISIBLE NONE;species_guess species_guess VISIBLE
NONE;scientific_name scientific_name VISIBLE NONE;common_name common_name VISIBLE
NONE;iconic_taxon_name iconic_taxon_name VISIBLE NONE;taxon_id taxon_id VISIBLE
NONE;taxon_kingdom_name taxon_kingdom_name VISIBLE NONE;taxon_phylum_name taxon_phylum_name VISIBLE
NONE;taxon_subphylum_name taxon_subphylum_name VISIBLE NONE;taxon_superclass_name
taxon_superclass_name VISIBLE NONE;taxon_class_name taxon_class_name VISIBLE
NONE;taxon_subclass_name taxon_subclass_name VISIBLE NONE;taxon_superorder_name
taxon_superorder_name VISIBLE NONE;taxon_order_name taxon_order_name VISIBLE
NONE;taxon_suborder_name taxon_suborder_name VISIBLE NONE;taxon_superfamily_name
taxon_superfamily_name VISIBLE NONE;taxon_family_name taxon_family_name VISIBLE
NONE;taxon_subfamily_name taxon_subfamily_name VISIBLE NONE;taxon_supertribe_name
taxon_supertribe_name VISIBLE NONE;taxon_tribe_name taxon_tribe_name VISIBLE
NONE;taxon_subtribe_name taxon_subtribe_name VISIBLE NONE;taxon_genus_name taxon_genus_name VISIBLE
NONE;taxon_genushybrid_name taxon_genushybrid_name VISIBLE NONE;taxon_species_name
taxon_species_name VISIBLE NONE;taxon_hybrid_name taxon_hybrid_name VISIBLE
NONE;taxon_subspecies_name taxon_subspecies_name VISIBLE NONE;taxon_variety_name taxon_variety_name
VISIBLE NONE;taxon_form_name taxon_form_name VISIBLE NONE;Shape Shape VISIBLE NONE')
... arcpy.Project_management('Taxon_Layer', r'C:\Users\Joshua\Documents\Lehman College
\VanCortlandt_Biodiversity_Project\Shapefiles\Observation_Points\\'+ shapefile_name, "PROJCS['NAD_
1983_StatePlane_New_York_Long_Island_FIPS_3104_Feet',GEOGCS['GCS_North_American_1983',DATUM
['D_North_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT
['Degree',0.0174532925199433]],PROJECTION['Lambert_Conformal_Conic'],PARAMETER
['False_Easting',984250.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',-
74.0],PARAMETER['Standard_Parallel_1',40.66666666666666],PARAMETER['Standard_Parallel_
2',41.03333333333333],PARAMETER['Latitude_Of_Origin',40.16666666666666],UNIT
['Foot_US',0.3048006096012192]]", '#', "GEOGCS['GCS_North_American_1983',DATUM['D_North_American_
1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT
['Degree',0.0174532925199433]]", 'NO_PRESERVE_SHAPE', '#', 'NO_VERTICAL')
... arcpy.Delete_management('XY_Layer', '#')
... arcpy.Delete_management('Taxon_Layer', '#')
... mxd = arcpy.mapping.MapDocument("CURRENT")
... df2  = arcpy.mapping.ListDataFrames(mxd)[0]
... addshp = arcpy.mapping.Layer(r'C:\Users\Joshua\Documents\Lehman College
\VanCortlandt_Biodiversity_Project\Shapefiles\Observation_Points\\'+ shapefile_name +'.shp')
... arcpy.mapping.AddLayer(df2, addshp, "BOTTOM")
...
...
```