“QGIS PLUGIN DEVELOPMENT TO CLIP TWO LAYERS WITH GEOMETRY”

HAMZA SADAQAT [18-ARID-3301]

TAYYAB HANIF [18-ARID-3320]

SHAHZAD SOHAIL [18-ARID-3329]

FACULTY OF AGRICULTURAL ENGINEERING AND TECHNOLOGY
INSTITUTE OF GEO-INFORMATION & EARTH OBSERVATION

PIR MEHR ALI SHAH
ARID AGRICULTURE UNIVERSITY RAWALPINDI
PAKISTAN
2021

DEDICATION

This work is dedicated to our respected teachers and my beloved friends.

CONTENTS

Acknowledgements

First thing first, I am thankful to Allah who granted us courage and will to fulfill this task. Then, we want to mention our Teacher Sir Shahinsha, because of his motivation and skillful lectures we completed our project.

HAMZA SADAQAT ABBASI

SHAHZAD SOHAIL

TAYYAB HANIF

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND:-

QGIS has made provision to extend its functionality by writing Plugins in Python and C++. Plugins are categorized as core, stable and experimental. Several Plugins are quite extensively used such as “OpenLayers” and “Semi-Automatic Classification Plugin”. This tutorial will outline the process involved in setting up your development environment, designing the user interface for a plugin and writing code to interact with QGIS.

1.2 PURPOSE:

To build a clip tool which updates the geometry columns.

1.3 PROBLEM STATEMENT/LIMITATION:

Measurement of geometry in GIS is very important. In different spatial analysis for example snow cover assessment, flood damage assessment, cadastral mapping, overlay analysis or different other spatial/location based analysis, accurate geometry measurement is very important. So, the existing clip tool in QGIS does not update the geometry columns. So, we have to build a tool which updates geometry column of the clipped file as well. We shall try to clip Snow cover area of Swat with the district boundary of Swat, KPK, and notice the change in the geometry columns and then we will compare with the clipped file that will be created using our plugin.

1.4 PRE-REQUISITES:

Knowledge of GIS data is a must. Working knowledge of Python programming will be added advantage.

1.5 QGIS VERSION:

3.6.0-Noosa

1.6 ADDITIONAL SOFTWARE REQUIRED:

(1) Qt Designer (comes bundled with the standard QGIS installation)
(2)QGIS Plugin - Plugin Builder,
(3) QGIS Plugin - Reloader plugin and
(4) a beautiful editor - Sublime Text 2 or Notepad ++ or Python shell

CHAPTER 2

TUTORIAL

2.1 PROCEDURE:

For the ease of understanding, both the layers used are in the UTM 43N / WGS 84 "projected" coordinate system. The map units are in meters

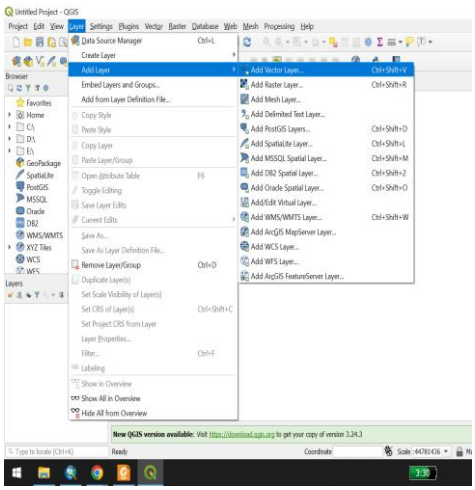
1. Add 2 overlapping vector layers. Click on Layer - Add Layer - Add Vector Layer. Navigate to the folder shared with you, press Ctrl and select the two files "Swat_snowcover.shp" and "Swat.shp"

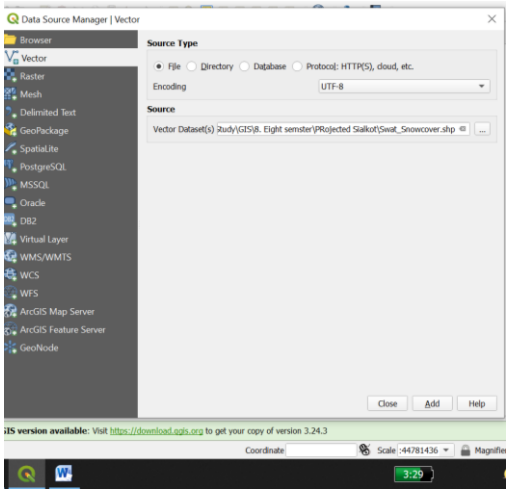
2. Click on Vector - Geometry tools - Add / Export Geometry columns. Select "Swat_snowcover.shp" in the drop and click Ok. Close the box, after process is over.

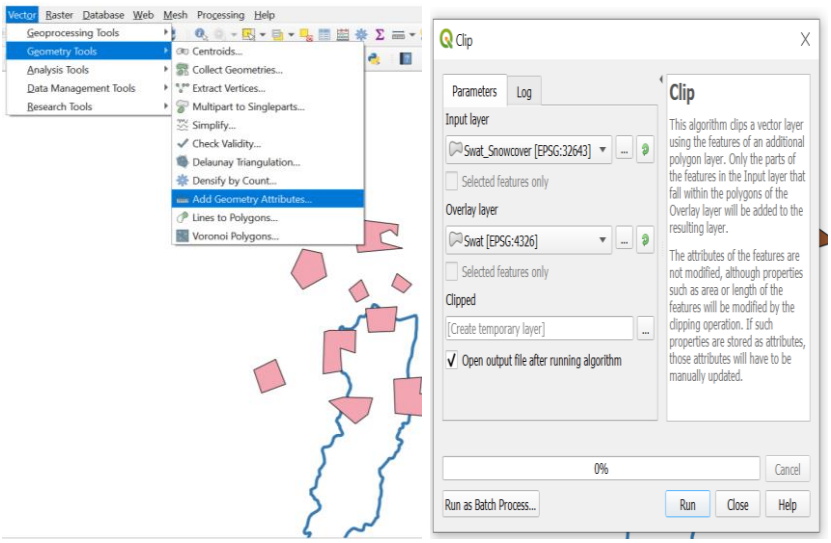
3. Now right click on the new layer "Added geom info", click on Open Attribute Table. Note the values of AREA and PERIMETER columns (esp., for features

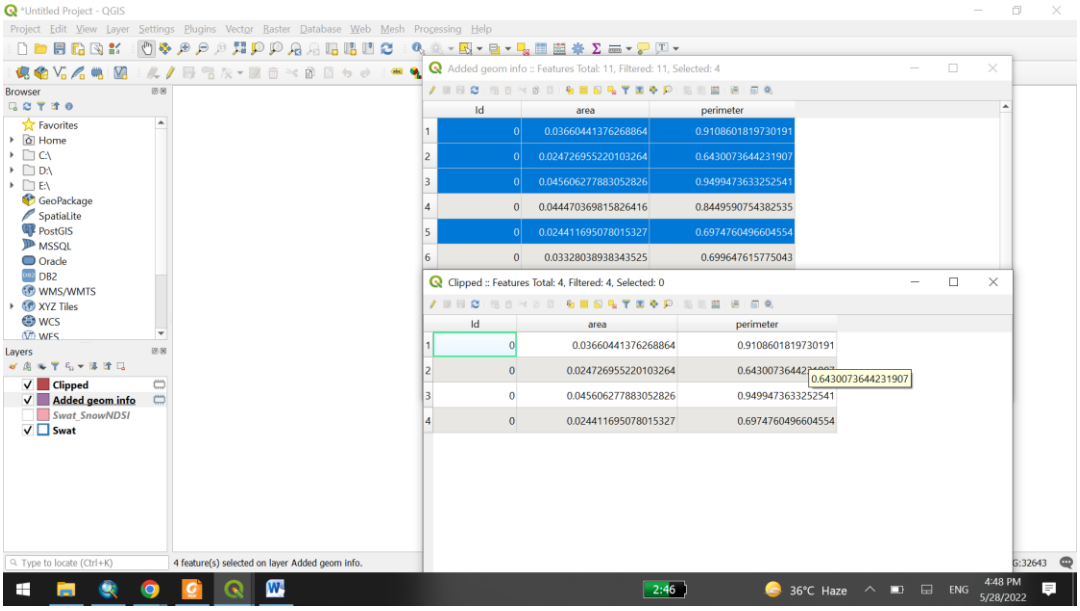
with POLY_ID). These are the area and perimeter values before the clip operation.

4. Now perform the clip operation and save the output file as “clip before”. Open the Attribute table of the “clip before.shp” file and compare the area and perimeter values noted down in the previous step. You will notice that although the features have changed, the area and perimeter values have not been updated.





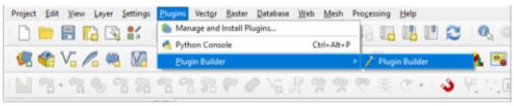




5. Now we shall try to create a tool to clip Snow cover area of Swat with the district boundary of Swat, KPK and update the geometry columns also.

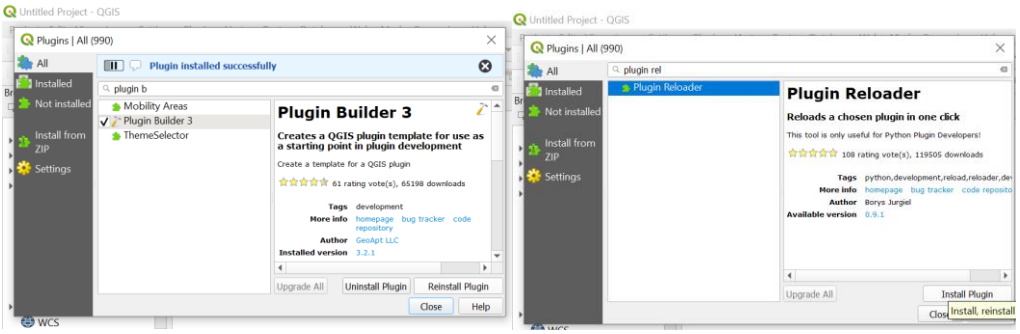
Procedure:

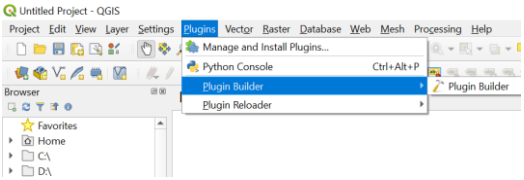
1. Open QGIS. Go to *Plugins - Plugin Builder - Plugin Builder*



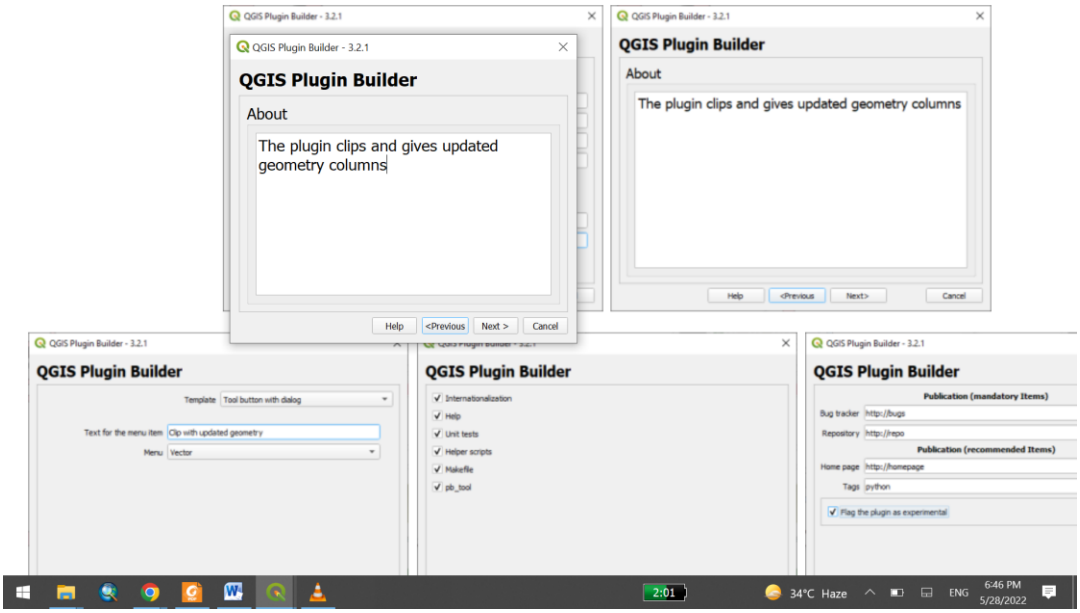
2. You will see the QGIS Plugin Builder dialog with a form. You can fill the form with details relating to our plugin. The Class name will be the name of the Python Class containing the logic of the plugin. This will also be the name of the folder containing all the plugin files. Enter ClipWithGeometry as the class name. The Plugin name is the name under which your plugin will appear in the Plugin Manager. Enter the name as “Clip With Geometry”. Add a description in the Description field. The Module name will be the name of the main python file for

the plugin. Enter it as clip_geometry. Leave the version numbers as they are. Enter your name and email address in the appropriate fields. Click on Next. Enter some text in the “About” box and click on Next. Select the “Tool button with dialog” from the Template selector. The Text for menu item value will be how he users will find your plugin in QGIS menu. Enter it as “Clip with Update Geometry”. The Menu field will decide where your plugin item is added in QGIS. Since our plugin is for vector data, select Vector. Click on Next. Plugin builder will prompt you for the type of files to generate. Keep the default selection and click Next. Check the Flag the plugin as experimental box at the bottom. Click Next

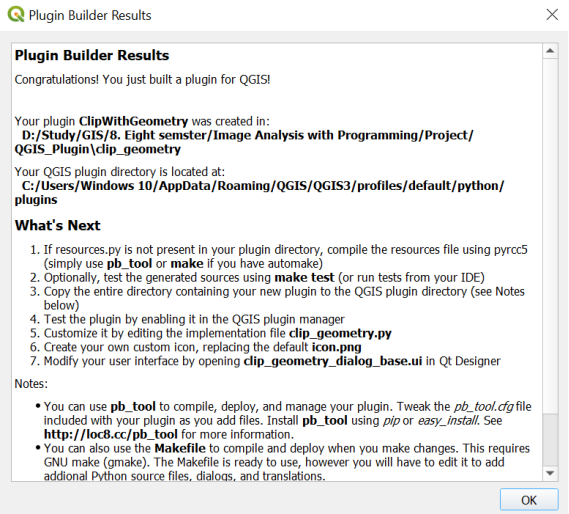


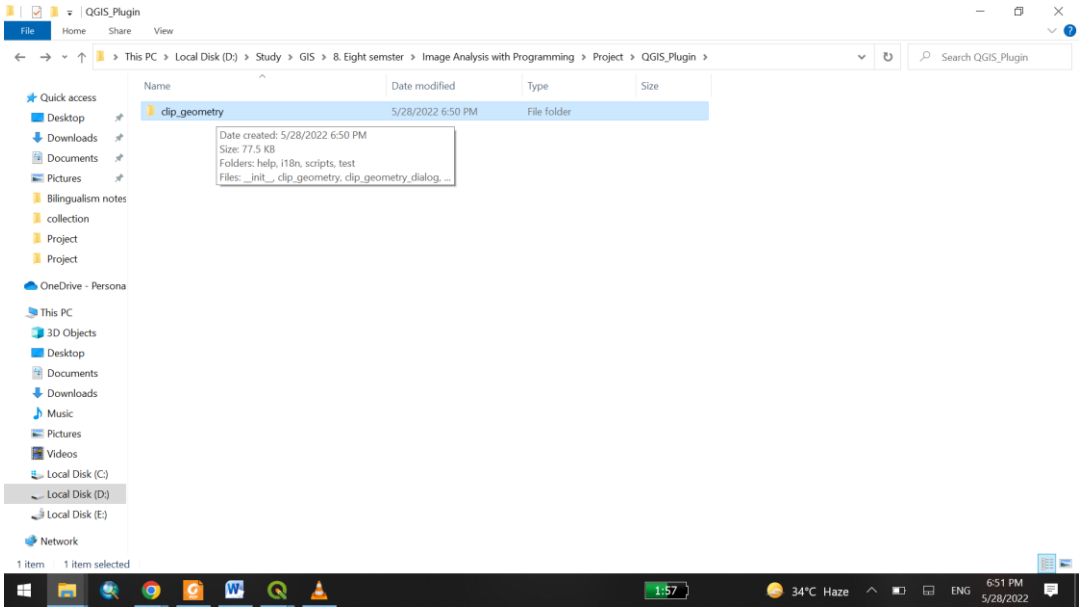


3. Next, you will be prompted to choose a directory for your plugin. You need to browse to the QGIS python plugin directory on your computer and select Select Folder. Provide the path of today’s folder: (Replace username with your login name). Click on Generate. You may get a warning saying pyrcc5 is not found in the path. You can ignore the message.

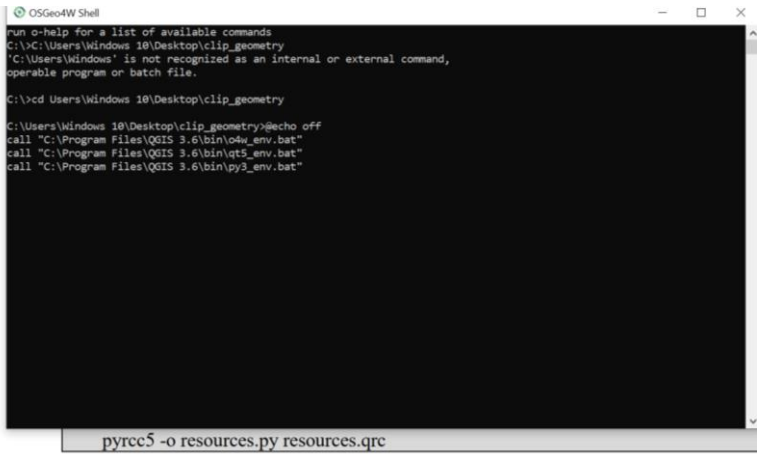


4. You will see a confirmation dialog once your plugin template is created. Note the path to the plugin folder

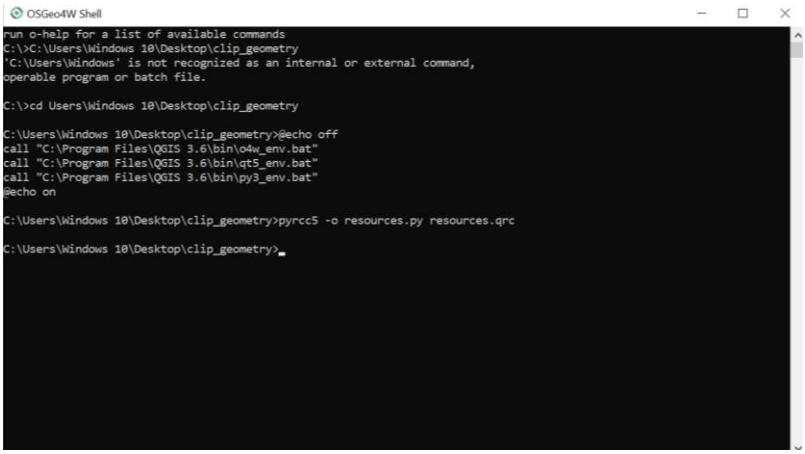




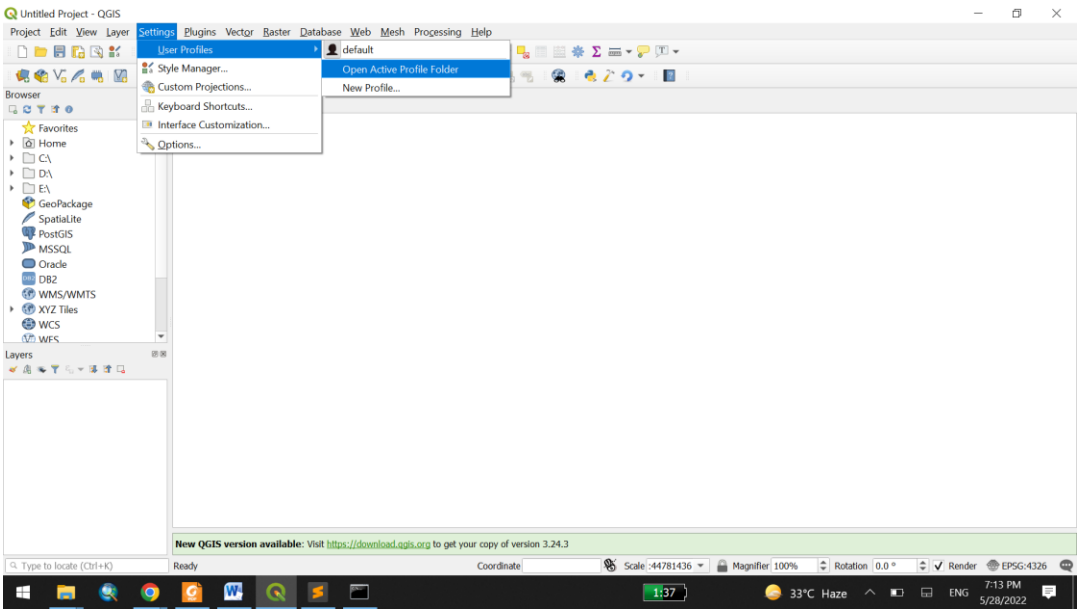
5. Before we can use the newly created plugin, we need to compile the resources.qrc file that was created by Plugin Builder. Launch the OSGeo4W Shell on windows start menu or running the file C:\Program Files\QGIS 3.6\OSGeo4W.bat.

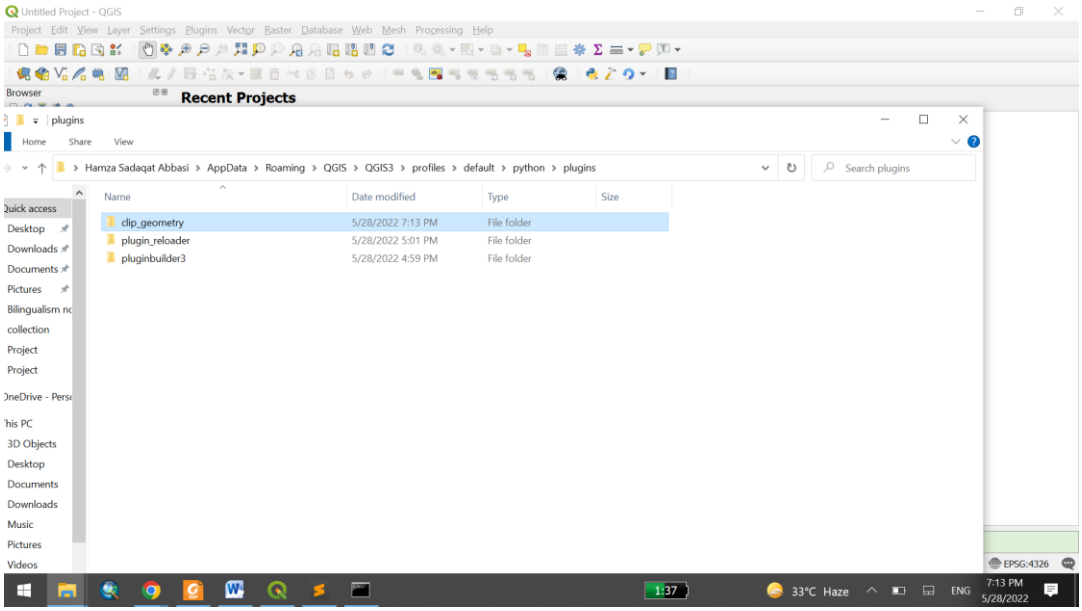


6. Once you are in the directory, type the following commands one by one. This will run the pyrcc4 command that we had installed as part of Qt bindings for Python

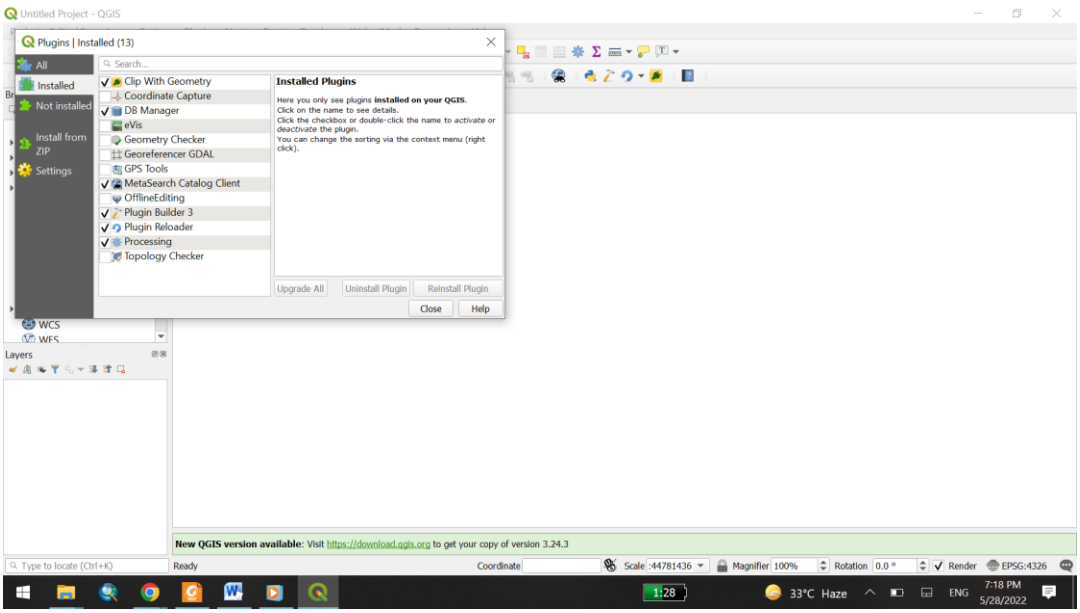


7. Plugins in QGIS are stored in a special folder. We must copy our plugin directory to that folder before it can be used. In QGIS, locate your current profile folder by going to Settings ‣ User Profiles ‣ Open Active Profile Folder. In the profile folder, copy the plugin folder to python ‣ plugins subfolder

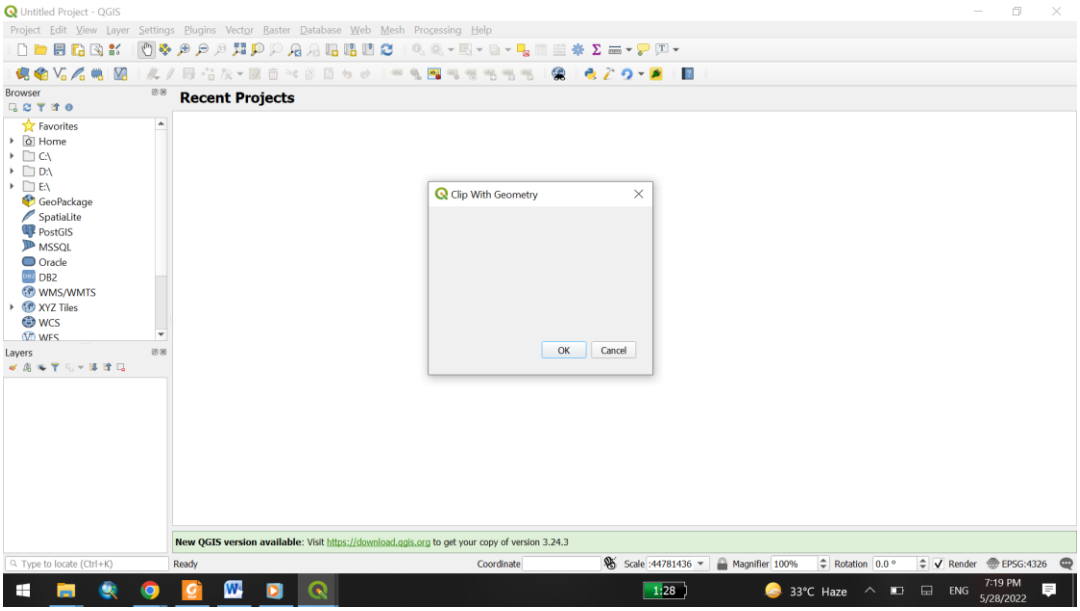




8. Now we are ready to have a first look at the brand new plugin we created. Close QGIS and launch it again. Go to Plugins - Manage and Install plugins and enable the “Clip With Geometry” plugin in the Installed tab. You will notice that there is a new icon in the toolbar and a new menu entry under Vector - Clip With Geometry - Clip with Update Geometry. Select it to launch the plugin dialog.

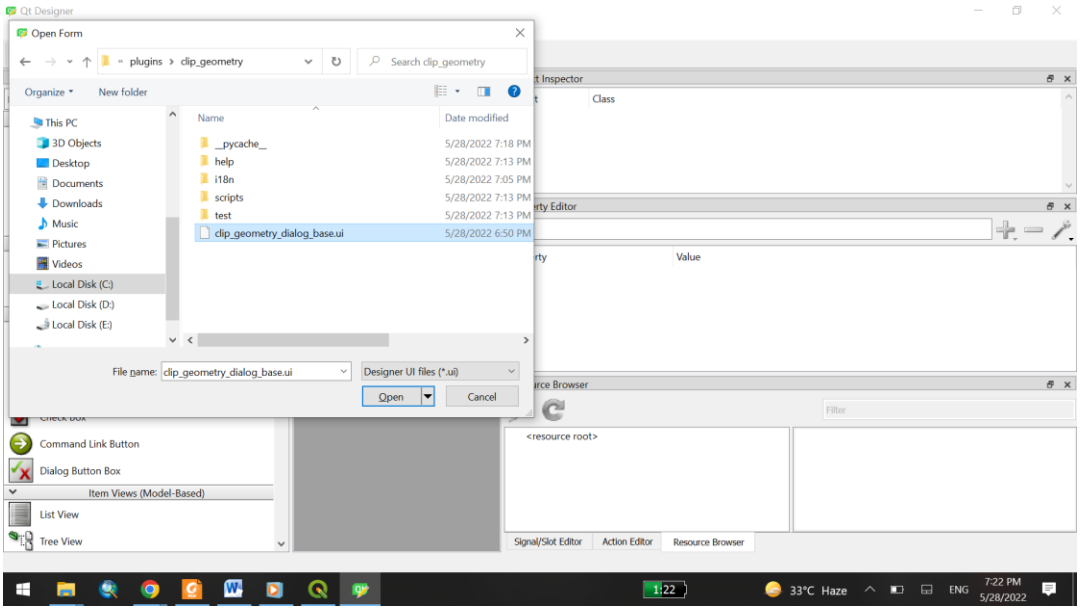


9. You will notice a new blank dialog named Clip With Geometry. Close this dialog

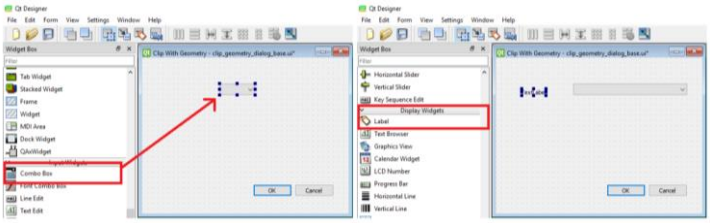


10. We will now design our dialog box and add some user interface elements to it. Open the QtDesigner program and to to File –> Open File or Project....

11. Browse to the plugin directory (in my case it is "C:\Users\prasun\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\clip_geometry") and select the clip_geometry_dialog_base.ui file. Click Open

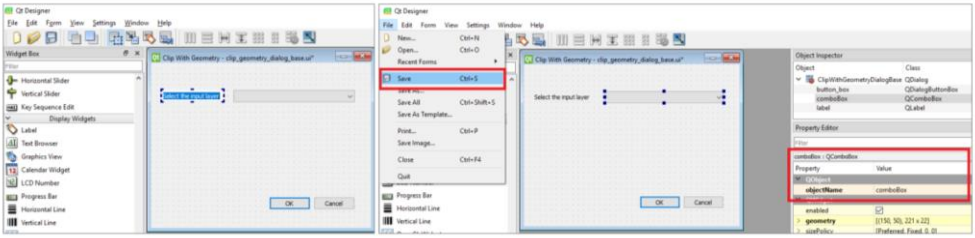


11. You will see the blank dialog from the plugin. You can drag-and-drop elements from the left-hand panel on the dialog. We will add a Combo Box type of Input Widget. Drag it to the plugin dialog



12. Resize the combo box and adjust its size. Now drag a Label type Display Widget on the dialog.

13. Click on the label text and enter “Select the input layer”.



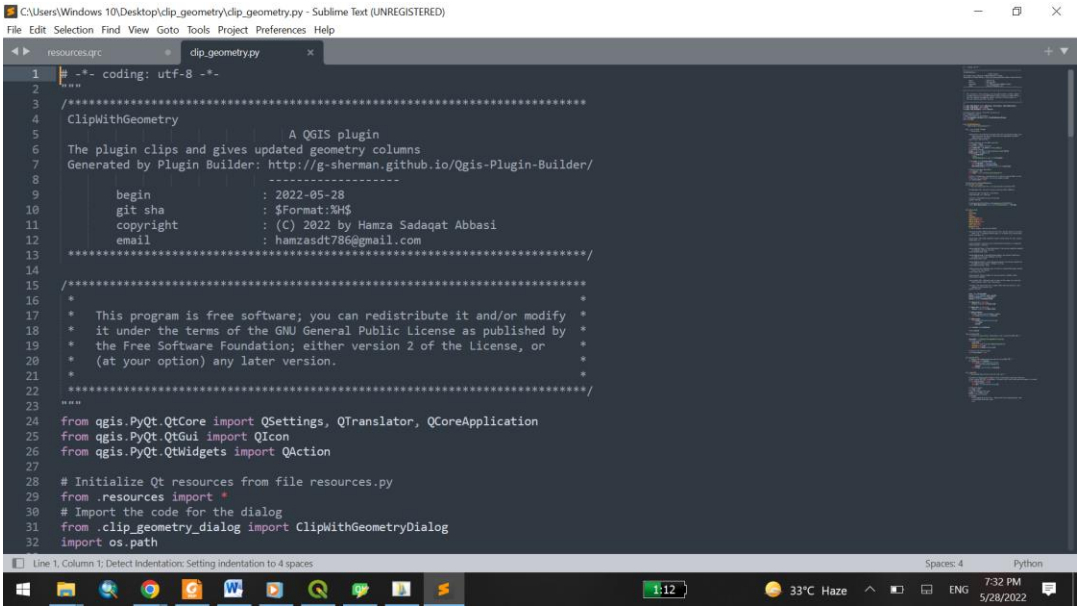
14. Save this file by going to File - Save “clip_geometry_dialog_base.ui”. Note the name of the combo box object is comboBox. To interact with this object using python code, we will have to refer to it by this name.

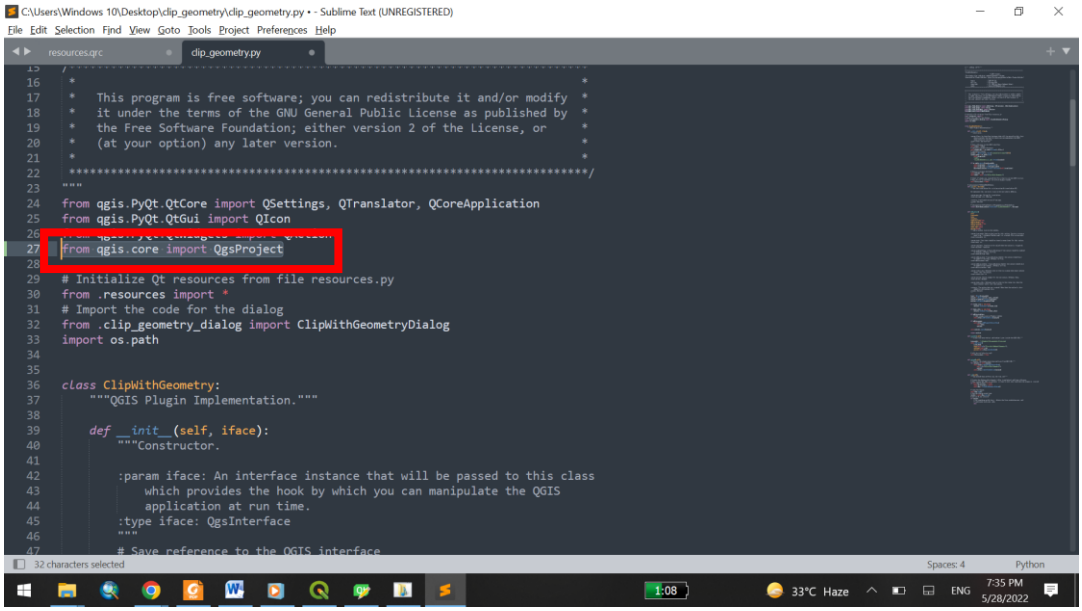
15. Let’s reload our plugin so we can see the changes in the dialog window. Go to Plugin – Plugin Reloader - Choose a plugin to be reloaded.

16. Select ClipWithGeometry in the Configure Plugin reloader dialog

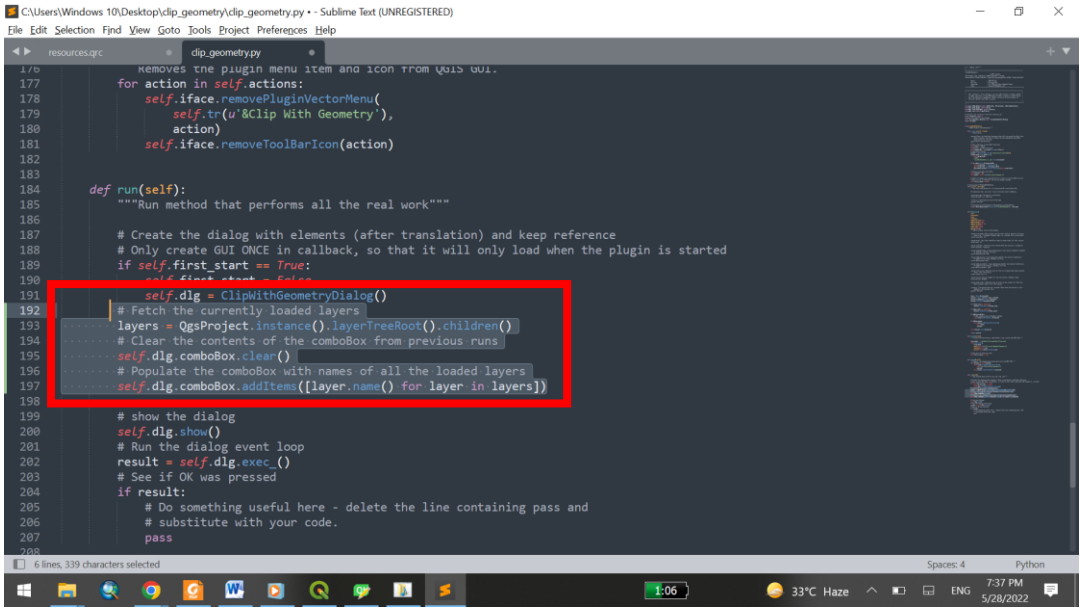
17. Now click the button. You will see the newly designed dialog box.

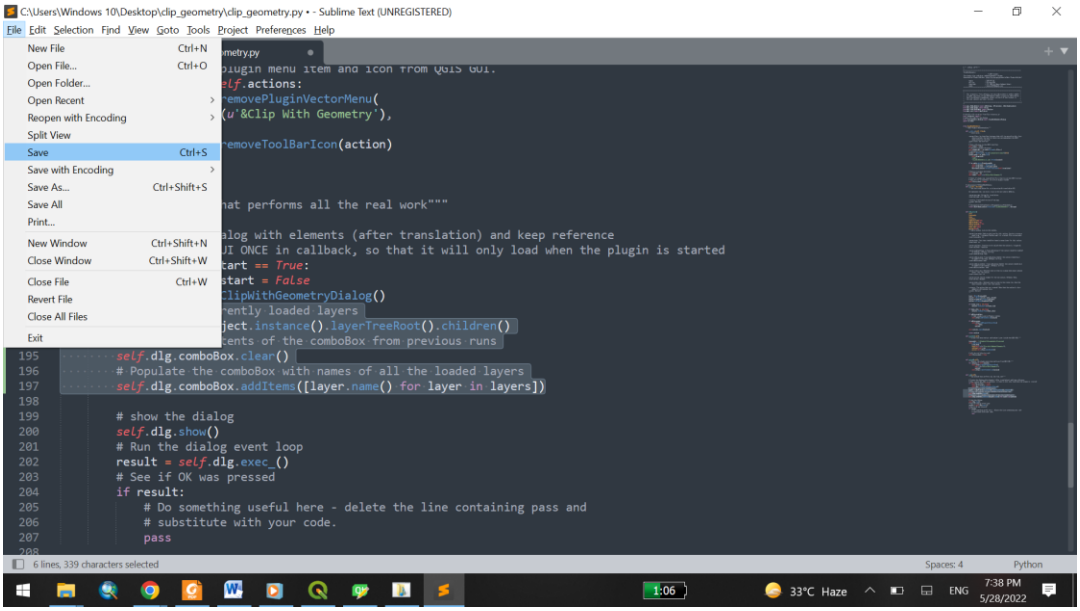
18. Let's add some logic to the plugin that will populate the combo box with the layers loaded in QGIS. Go to the plugin directory and load the file clip_geometry.py in a text editor. First, insert at the top of the file with the other imports





19. Scroll down and find the run(self) method. This method will be called when you click the toolbar button or select the plugin menu item. Add the following code at the beginning of the method. This code gets the layers loaded in QGIS and adds it to the comboBox object from the plugin dialog. Save the file





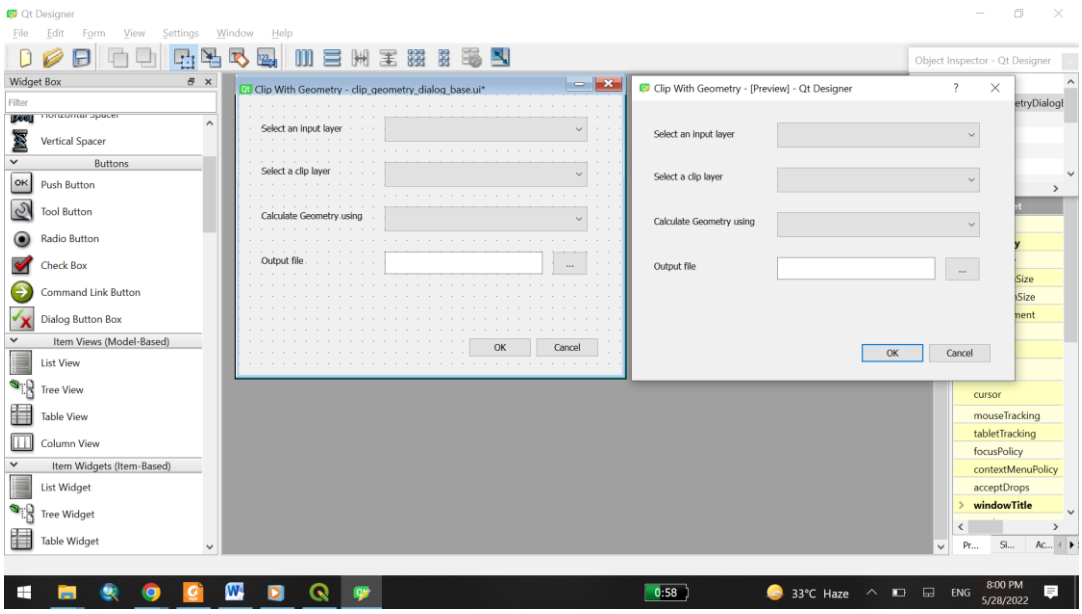
20. . Back in the main QGIS window, reload the plugin by going to Plugins – Plugin Reloader – Reload plugin: ClipWithGeometry. Alternatively, you can just press Ctrl+F5

21. To test this new functionality, we must load some layers in QGIS. Add the two layers provided (Swat_Snowcover and Swat).

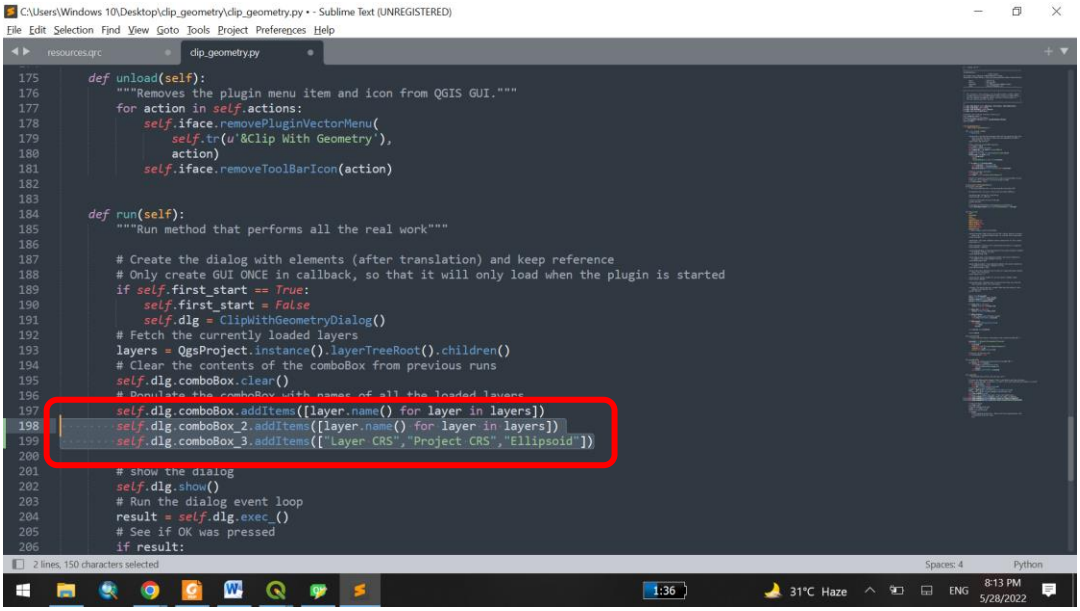
22. After you load the data, launch the plugin by going to Vector - Clip With Geometry - Clip with Updated Geometry. You will see that our combo box is now populated with the layer names that are loaded in QGIS

23. Let's add remaining user interface elements. Switch back to Qt Creator and load theclip_geometry.ui file.

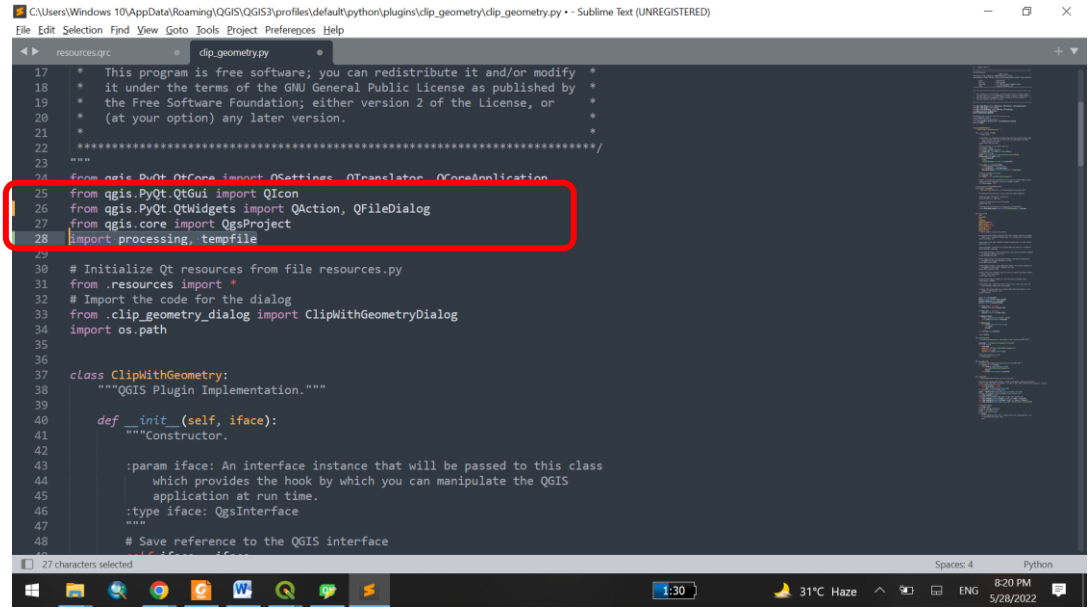
- Add a Label *Display Widget* and change the text to "Select the clip layer".
- Add another Combo Box type *Input Widget* that will show the clip layer (objectName =comboBox_2).
- Add a Label *Display Widget* and change the text to "Calculate Geometry using".
- Add another Combo Box type *Input Widget* that will show the method used to calculate the Geometry values (objectName = comboBox_3).
- Add a Label Display Widget and change the text to "Select the output file".
- Add a LineEdit type *Input Widget* that will show the output file path that the user has chosen (objectName = lineEdit).
- Next, add a Push Button type *Button* and change the button label to "...." Note the object names of the widgets that we will have to use to interact with them (objectName = pushButton). Save the file



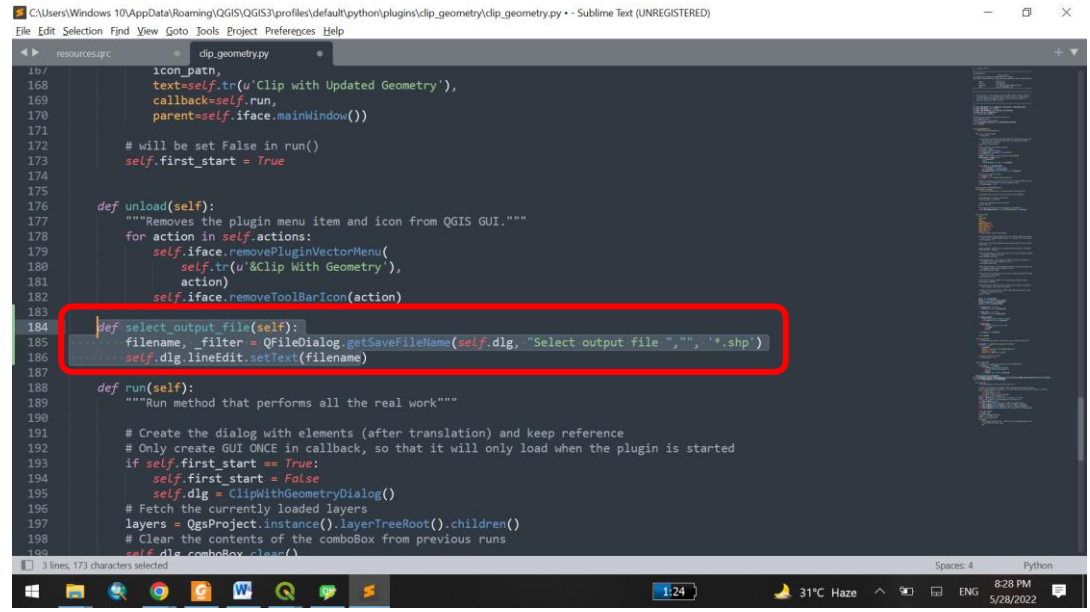
24. Open the clip_geometry.py file in a text editor. Scroll down and find the run(self) method. Add the following code lines as shown. The 1st line gets the layers loaded in QGIS and adds it to the comboBox_2 object from the plugin dialog; and the 2nd line indicates the options available to the user to Calculate Geometry value



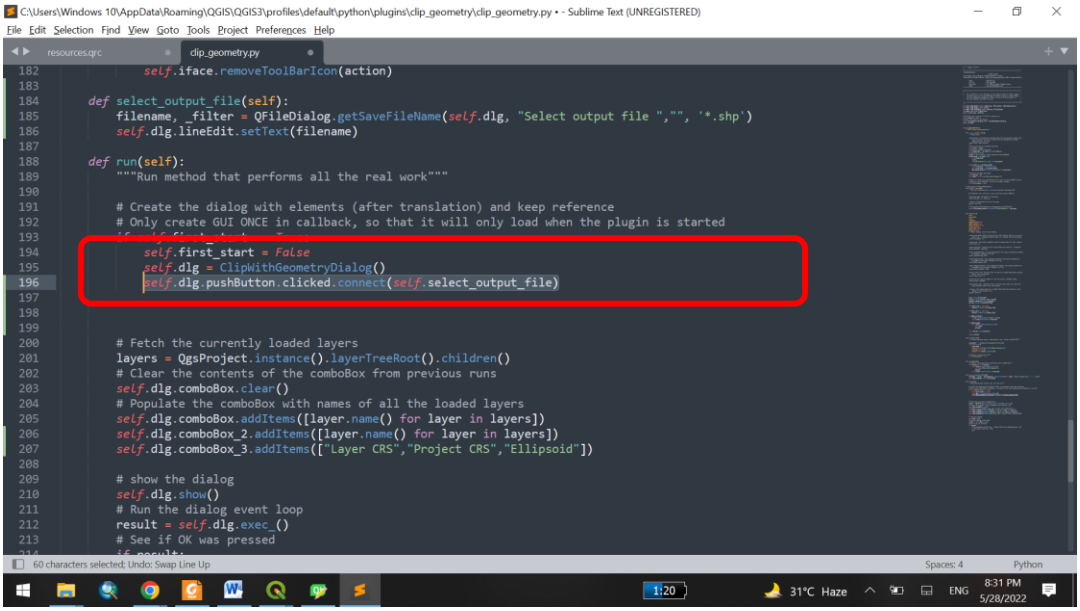
25. We will now add python code to open a file browser when the user clicks the "..." push button and show the select path in the line edit widget. Add QFileDialog, processing and tempfile to our list of imports at the top of the clip_geometry.py file



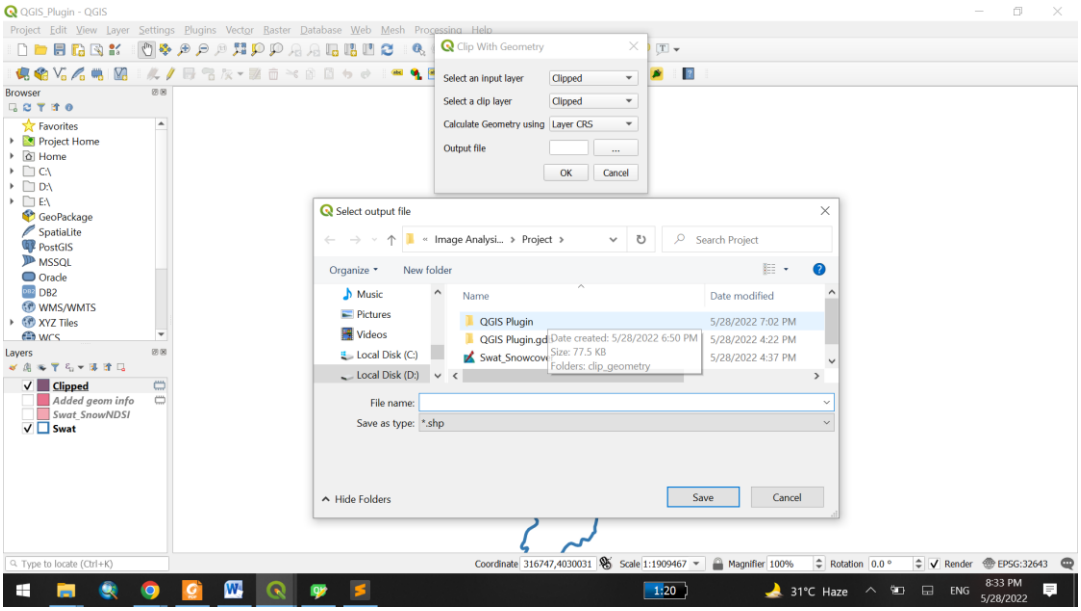
26. Add a new method called select_output_file with the following code. This code will open a file browser and populate the line edit widget with the path of the file that the user chose



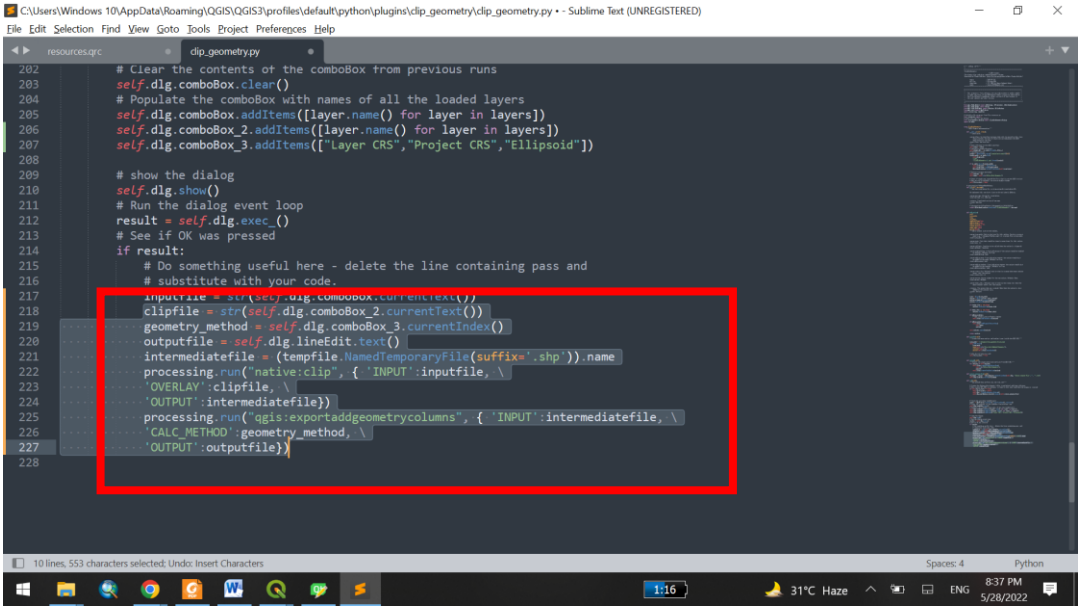
27. Now we need to add code so that when the ... button is clicked, select_output_file method is called. Scroll down to the run method and add the following line in the block where the dialog is initialized. This code will connect the select_output_file method to the clicked signal of the push button widget



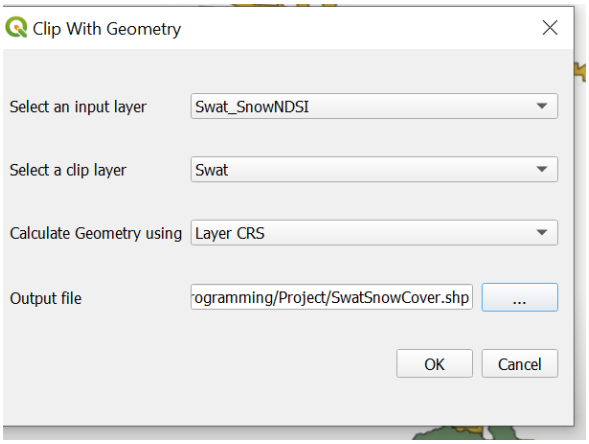
28. Back in QGIS, reload the plugin and open the Clip With Geometry dialog. If all went fine, you will be able to click the ... button and select an output text file from your disk



27. When you click OK on the plugin dialog, nothing happens. That is because we have not added the logic to pull attribute information from the layer and write it to the text file. We now have all the pieces in place to do just that. Find the place in the run method where it says pass. Replace it with the code below.

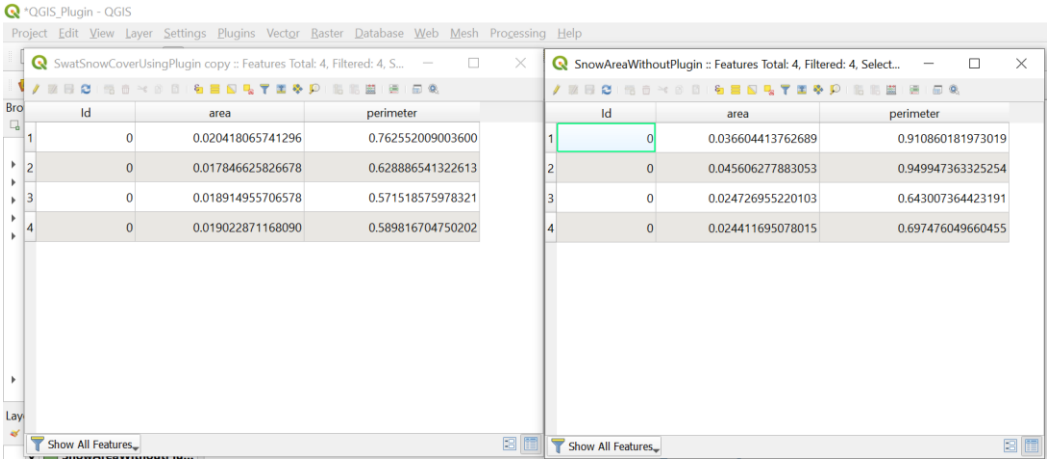


29. Now our plugin is ready. Reload the plugin and try it out. You will find that the output shape file will have the updated geometry columns after the clip operation.



3.1 RESULTS AND DICUSSION:

Here is the comparison of both shapefiles. On the right, there is attribute table of file that is generated using clip tool of the QGIS. While, on the left side this shapefile is clipped using GetGeometryPlugin. Now, we can clip updated geometry using this plugin.



DEVELOPED BY:

Hamza Sadaqat Abbasi & Tayyab Hanif & Shahzad Sohail, Institute of Geoinformatics and Earth Observation, PMAS-AAUR (hamzasdt786@gmail.com)

DATA SOURCES:

I have uploaded Plugin and tutorial on my Github account. Anyone can access to the plugin using following link and can download and install in QGIS.

https://github.com/Hamza-Sadaqat/QGIS-Plugin-Devlopment