AreaDivider qGİS pLUGİN

Release 1.0.0

Ata

Jan 21, 2021

CONTENTS:

CHAPTER

ONE

SOURCES

1.1 save_attributes module

class save_attributes.**SaveAttributes**(*iface*)

Bases: object

add_action(*icon_path*, *text*, *callback*, *enabled_flag=True*, *add_to_menu=True*, *add_to_toolbar=True*, *status_tip=None*, *whats_this=None*, *parent=None*)

Parameters

- **icon_path** (*str*) – Path to the icon for this action. Can be a resource path (e.g. ':/plugins/foo/bar.png') or a normal file system path.
- **text** (*str*) – Text that should be shown in menu items for this action.
- **callback** (*function*) – Function to be called when the action is triggered.
- **enabled_flag** (*bool*) – A flag indicating if the action should be enabled by default. Defaults to True.
- **add_to_menu** (*bool*) – Flag indicating whether the action should also be added to the menu. Defaults to True.
- **add_to_toolbar** (*bool*) – Flag indicating whether the action should also be added to the toolbar. Defaults to True.
- **status_tip** (*str*) – Optional text to show in a popup when mouse pointer hovers over the action.
- **parent** (*QWidget*) – Parent widget for the new action. Defaults None.
- **whats_this** – Optional text to show in the status bar when the mouse pointer hovers over the action.

Returns The action that was created. Note that the action is also added to self.actions list.

Return type QAction

apply_graduated_symbology(*area_size*)

Creates Symbology for each value in range of values. Values are # of patients per zip code. Hard codes min value, max value, symbol (color), and label for each range of values. Then QgsSymbolRenderer takes field from attribute table and item from myRangeList and applies them to join_layer. Color values are hex codes, in a graduated fashion from light pink to black depending on intensity

createShp(*input_line*, *costs*, *out_shp*, *sr*)

error_msj(*uyari*)

initGui()
Create the menu entries and toolbar icons inside the QGIS GUI.

input_shp_file()

loadLayerList()

load_comboBox()
Load the fields into combobox when layers are changed

run()
Run method that performs all the real work

runAlgorithm()

select_output_file()

tr(*message*)
Get the translation for a string using Qt translation API.

We implement this ourselves since we do not inherit QObject.

Parameters **message** (*str, QString*) – String for translation.

Returns Translated version of message.

Return type QString

unload()
Removes the plugin menu item and icon from QGIS GUI.

CHAPTER

TWO

QGIS PLUGIN AREA DIVIDER

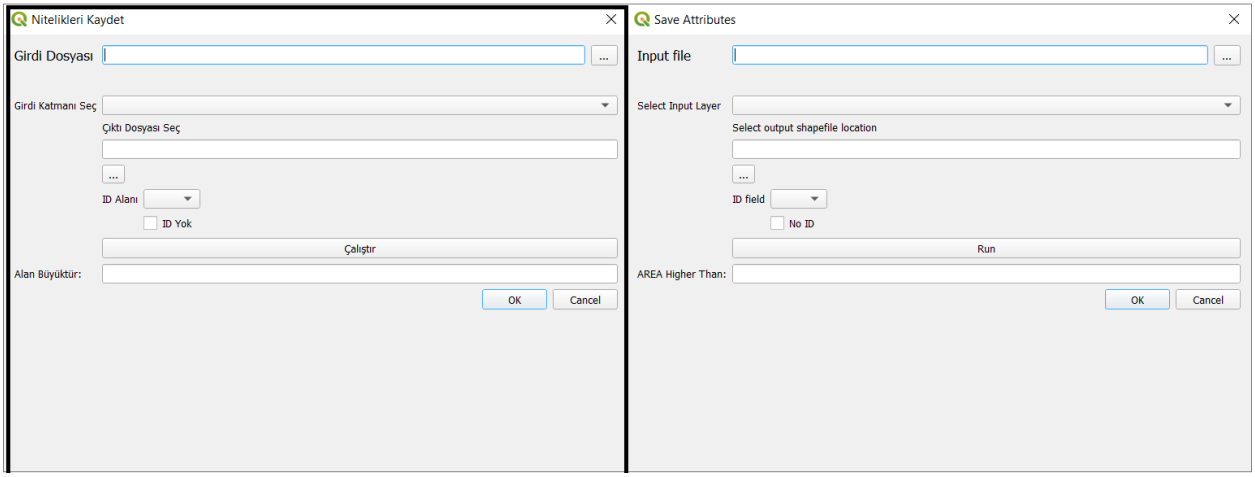
Authors:

Atahan Çelebi

Eda Karabaş

2.1 Plugin's User Interface

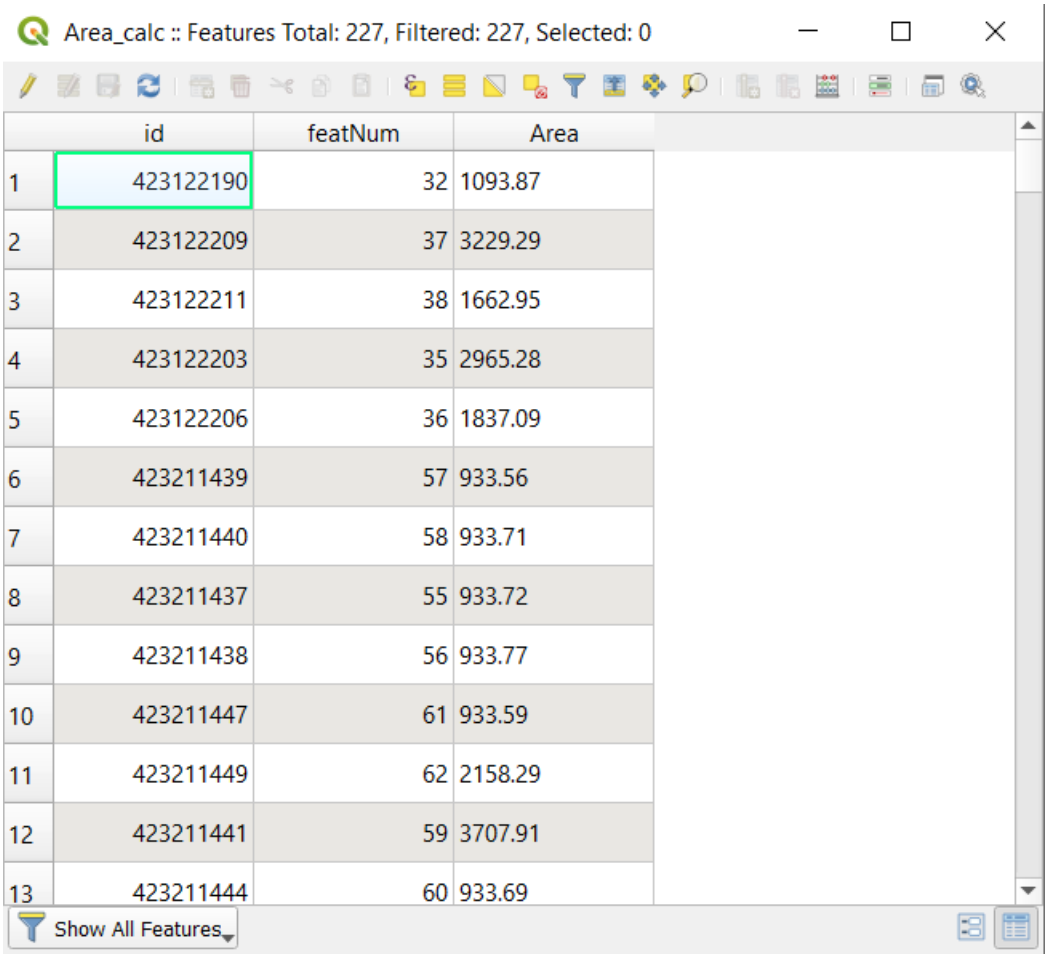
-Language Support (Eng-Tr)



We can see the lower and upper value according to user choice. After the selection of the area unit this part split the higher areas and the lower areas

2.2 Apply The Graduated Function

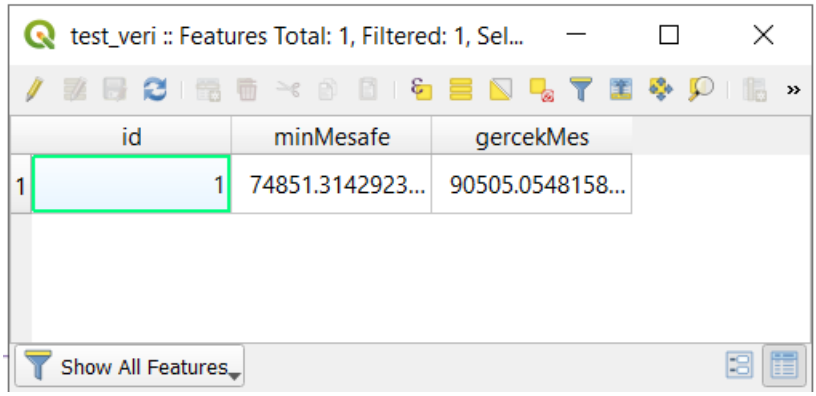
You can also look at the Area_calc's attribues



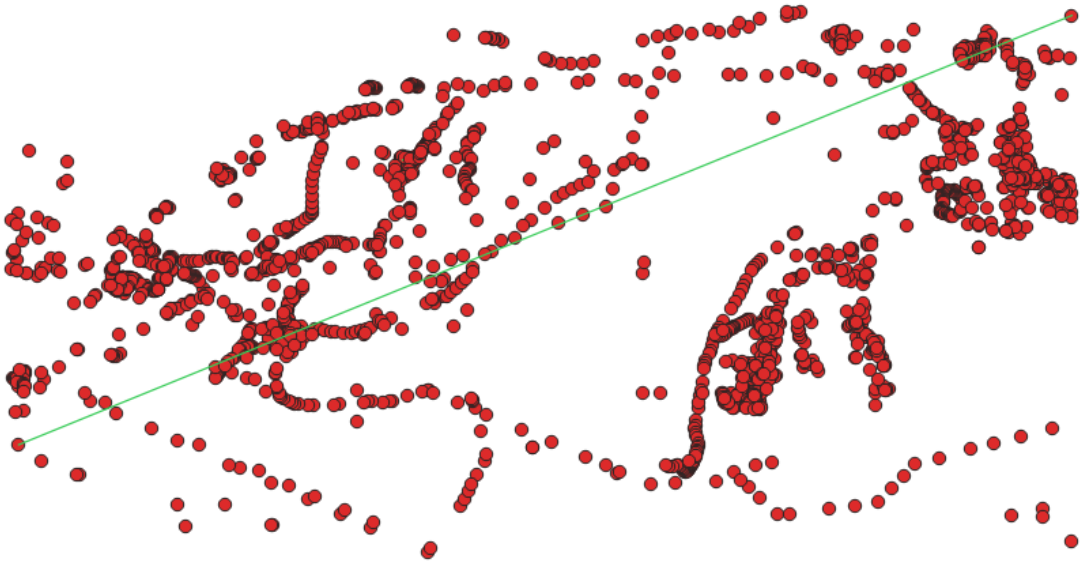
	id	featNum	Area
1	423122190	32	1093.87
2	423122209	37	3229.29
3	423122211	38	1662.95
4	423122203	35	2965.28
5	423122206	36	1837.09
6	423211439	57	933.56
7	423211440	58	933.71
8	423211437	55	933.72
9	423211438	56	933.77
10	423211447	61	933.59
11	423211449	62	2158.29
12	423211441	59	3707.91
13	423211444	60	933.69

2.3 Type of geometry is a point or line

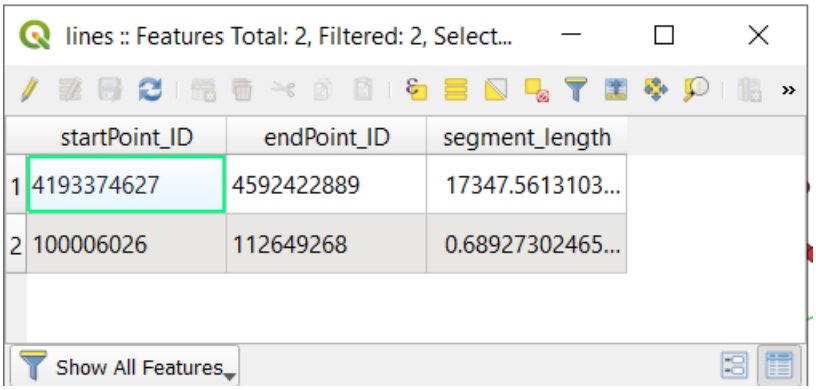
If we select a line shapefile it draws the minimum flight distance between the two points and the their real distance.



	id	minMesafe	gercekMes
1	1	74851.3142923...	90505.0548158...



Find the points that have minimum distance and the maximum distance



CHAPTER

THREE

QGIS PLUGIN AREA DIVIDER

This **unit test** work for the layer if it has been added or not and check its suitability

Test Layer Transaction -Qgis Project instance/Add map layer whether layer exist -try/except raise error if not exist could not pass unit test

Test Load Layer -Read the layer -If None could not pass unit test

Here is the all code that we implement:

```
class SaveAttributesTest(unittest.TestCase):
    """Test dialog works."""
    @classmethod
    def setUpClass(cls):
            cls.iface = get_iface()
    def setUp(self):
            """Runs before each test."""
            QgsProject.instance().clear()
            self.dialog = SaveAttributesDialog(None)
            self.attributes=SaveAttributes(get_iface())
            self.attributes.dlg=self.dialog
    def test_Layer(self):
            try:
                    vlayer = QgsVectorLayer("TEST_LINE","line","memory")
                    vlayer = QgsVectorLayer("TEST_LINE","point","memory")
                    vlayer = QgsVectorLayer("TEST_LINE","polygon","memory")

            except:
                    self.assertTrue(True==False)

    def test_Layer_Transaction(self):
            try:
                    vlayer = QgsVectorLayer("TEST_LINE","line","memory")
                    QgsProject.instance().addMapLayer(vlayer)

            except:
                    self.assertTrue(True==False)

    def test_load_layer(self):
            path=os.path.dirname(os.path.realpath(__file__)) + "../../save_attributes/
↪unittests/"
            ds = ogr.Open(path+"c_beytepe.shp",0)
            self.assertTrue(ds==None)

    def tearDown(self):
            """Runs after each test."""
```

(continues on next page)

(continued from previous page)

```
            self.dialog = None


    if __name__ == "__main__":
    suite = unittest.makeSuite(SaveAttributesDialogTest)

    logging.basicConfig(stream=sys.stderr, level=logging.DEBUG)
    runner = unittest.TextTestRunner(verbosity=2)
    runner.run(suite)
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX