from qgis.core import (

QgsProject,

QgsPointXY,

QgsVectorLayer,

QgsField,

QgsGeometry,

QgsFeature,

QgsCoordinateReferenceSystem

)

from qgis.PyQt.QtCore import QVariant

from qgis.utils import iface

import csv

# Specify the path to the CSV file

csv\_file\_path = "/Users/ismailsa/Downloads/0.AaaaGIS/SabahNew/dataCase/dummyCase1.csv"

# Create a new point vector layer

layer\_name = "CSV Points"

crs = QgsCoordinateReferenceSystem("EPSG:4326") # Assuming WGS84 coordinates

# Read the CSV file and add features to the layer

with open(csv\_file\_path, "r") as file:

csv\_data = csv.reader(file)

header = next(csv\_data) # Read and store the header row

# Create QgsField objects for the additional columns

additional\_fields = [QgsField(header[i], QVariant.String) for i in range(3, len(header))]

# Create the layer with all necessary fields

fields = [

QgsField("ID", QVariant.Int),

QgsField("X", QVariant.Double),

QgsField("Y", QVariant.Double),

QgsField("Gender", QVariant.String),

QgsField("Year", QVariant.Int),

QgsField("Week", QVariant.Int),

QgsField("Child", QVariant.Int),

QgsField("Income", QVariant.String),

QgsField("House", QVariant.String),

QgsField("Education", QVariant.String)

] + additional\_fields

layer = QgsVectorLayer("Point?crs={}".format(crs.toWkt()), layer\_name, "memory")

provider = layer.dataProvider()

provider.addAttributes(fields)

layer.updateFields()

for row in csv\_data:

id = int(row[0])

x, y = float(row[1]), float(row[2])

gender = row[3]

year = int(row[4])

week = int(row[5])

child = int(row[6])

income = row[7]

house = row[8]

education = row[9]

point = QgsPointXY(x, y)

geom = QgsGeometry.fromPointXY(point)

feature = QgsFeature()

feature.setGeometry(geom)

# Extract the additional column values and add them as attributes

additional\_values = [row[i] for i in range(3, len(row))]

attributes = [id, x, y, gender, year, week, child, income, house, education] + additional\_values

feature.setAttributes(attributes)

provider.addFeature(feature)

# Add the layer to the map

QgsProject.instance().addMapLayer(layer)

# Specify the path to save the Shapefile

shapefile\_path = "/Users/ismailsa/Downloads/0.AaaaGIS/SabahNew/dataCase/dummyCase1.shp"

# Save the layer as a Shapefile

QgsVectorFileWriter.writeAsVectorFormat(layer, shapefile\_path, "utf-8", layer.crs(), "ESRI Shapefile")

# Load the saved Shapefile as a vector layer

saved\_layer = QgsVectorLayer(shapefile\_path, "dummyCase1", "ogr")

if not saved\_layer.isValid():

print("Failed to load the Shapefile.")

else:

# Add the saved layer to the map

QgsProject.instance().addMapLayer(saved\_layer)

# Remove the temporary CSV Points layer

QgsProject.instance().removeMapLayer(layer)

# Path to the shapefile

shapefilePoly\_path = '/Users/ismailsa/Downloads/0.AaaaGIS/SabahNew/dataCase/sabahDummy.shp'

shapefilePoly\_layer\_name = 'Shapefile'

shapefilePoly\_provider\_type = 'ogr'

# Load the shapefile

shapefilePoly\_layer = QgsVectorLayer(shapefilePoly\_path, shapefilePoly\_layer\_name, shapefilePoly\_provider\_type)

# Check if the shapefile layer was loaded successfully

if not shapefilePoly\_layer.isValid():

print("Failed to load the shapefile layer!")

exit()

# Add the shapefile layer to the QGIS workspace

iface.addVectorLayer(shapefilePoly\_path, shapefilePoly\_layer\_name, shapefilePoly\_provider\_type)

# Set the active layer in the QGIS GUI to the shapefile layer

iface.setActiveLayer(shapefilePoly\_layer)

# Zoom to the extent of the shapefile layer

iface.zoomToActiveLayer()

# Run the "Count Points in Polygon" algorithm

result = processing.run("qgis:countpointsinpolygon", {

'POLYGONS': shapefilePoly\_layer,

'POINTS': saved\_layer,

'FIELD\_PREFIX': 'count\_',

'OUTPUT': '/Users/ismailsa/Downloads/0.AaaaGIS/SabahNew/dataCase/caseFile.shp'

})

# Check if the algorithm ran successfully

if result['OUTPUT']:

count\_layer = QgsVectorLayer(result['OUTPUT'], "caseFile", "ogr")

if not count\_layer.isValid():

print("Failed to load the count layer.")

else:

# Add the count layer to the QGIS workspace

QgsProject.instance().addMapLayer(count\_layer)

else:

print("Failed to run the 'Count Points in Polygon' algorithm.")

# Remove the 'Shapefile.shp' layer from the QGIS workspace

shapefile\_layer = QgsProject.instance().mapLayersByName('Shapefile')

if len(shapefile\_layer) > 0:

QgsProject.instance().removeMapLayer(shapefile\_layer[0])