**TECHNICAL REPORT ON APPLICATION OF SPATIAL DATA TO IDENTIFY ROAD RISKSIN NUWARA-ELIYA, AMBAGAMUWA AREA, SRI LANKA**

**Introduction**

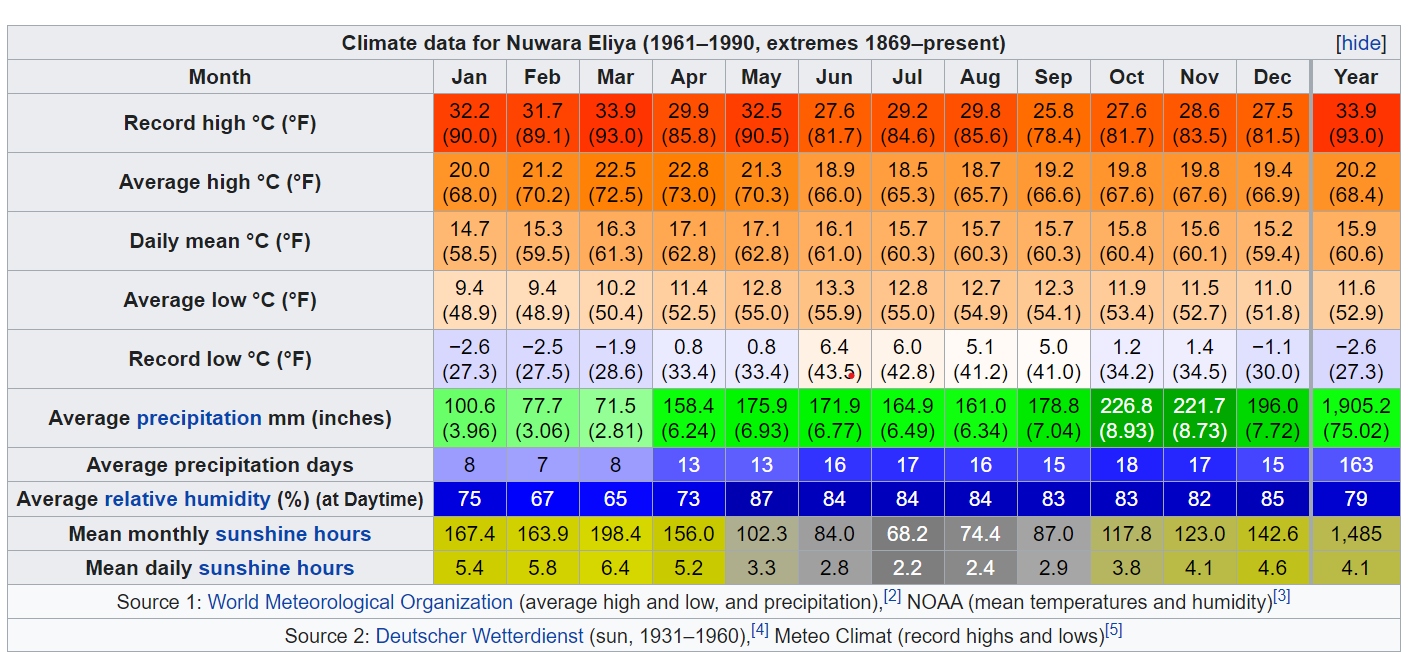
Based on the surrounding conditions, the Extreme Floods are regarded as one of the worst natural disasters. The damaging effects of the floods have a tremendous influence on human life. The harsh environmental conditions make it impossible for anybody to foresee the extreme event of flood risk in this region. The GIS role in disasters is frequently significant in vital life-saving strategies that are employed by contemporary developed nations worldwide. Diverse models and architectures for disaster management combining GIS and a hydraulic sensor-based approach have been proposed by researchers, scientists, and engineers from around the world in urgent emergency situations.

Due to pressure changes in the Bay of Bengal and strong winds, which are caused by Sri Lanka's location between the Gulf of Mannar and the Bay of Bengal in the Indian Ocean, unexpectedly severe rains can occur. Additionally, there are two inter-monsoonal rains and two monsoonal rains in Sri Lanka per year. The Disaster Management Center (DMC) serves as the coordinating organization for Disaster Release Managing (DRM) activities in Sri Lanka. The Ministry of Disaster Management was established in 2006 under its aegis. Flood Hazard Mapping has been one of the top priorities for DMC to complete because flooding is the most frequent natural disaster. It is debatable whether the current river system morphology can withstand these frequent and protracted high floods. The other concern is how the expanding human population is affecting and changing river systems' floodplains. All of these elements have a part in the rising risks and losses brought on by floods.

The mapping of flood zones using GIS technology will make it simple to develop non-structural solutions that lower the risks and damages associated with flooding. Implementing a flood management program that includes flood forecasting and mapping flood danger and vulnerability will be extremely beneficial to the populace. In order to lessen the risk of flooding in this area, the benefits of this project are very relevant for the residents of the Nuwara-Eliya, Ambagamuwa area, Sri Lanka and governmental agencies.

Methodology

Study Area

This study was conducted in Nuwara-Eleya which is a city in the hill country of the [Central Province, Sri Lanka](https://en.wikipedia.org/wiki/Central_Province,_Sri_Lanka). Its name means "city on the plain (table land)" or "city of light". The city is the administrative capital of [Nuwara Eliya District](https://en.wikipedia.org/wiki/Nuwara_Eliya_District), with a picturesque landscape and temperate climate. It is at an altitude of 1,868 m (6,128 ft) and is considered to be the most important location for [tea production in Sri Lanka](https://en.wikipedia.org/wiki/Tea_production_in_Sri_Lanka). The city is overlooked by [Pidurutalagala](https://en.wikipedia.org/wiki/Pidurutalagala" \o "Pidurutalagala), the tallest mountain in Sri Lanka. Nuwara Eliya is known for its temperate, cool climate – the coolest area in Sri Lanka.( [Nuwara Eliya - Wikipedia](https://en.wikipedia.org/wiki/Nuwara_Eliya))

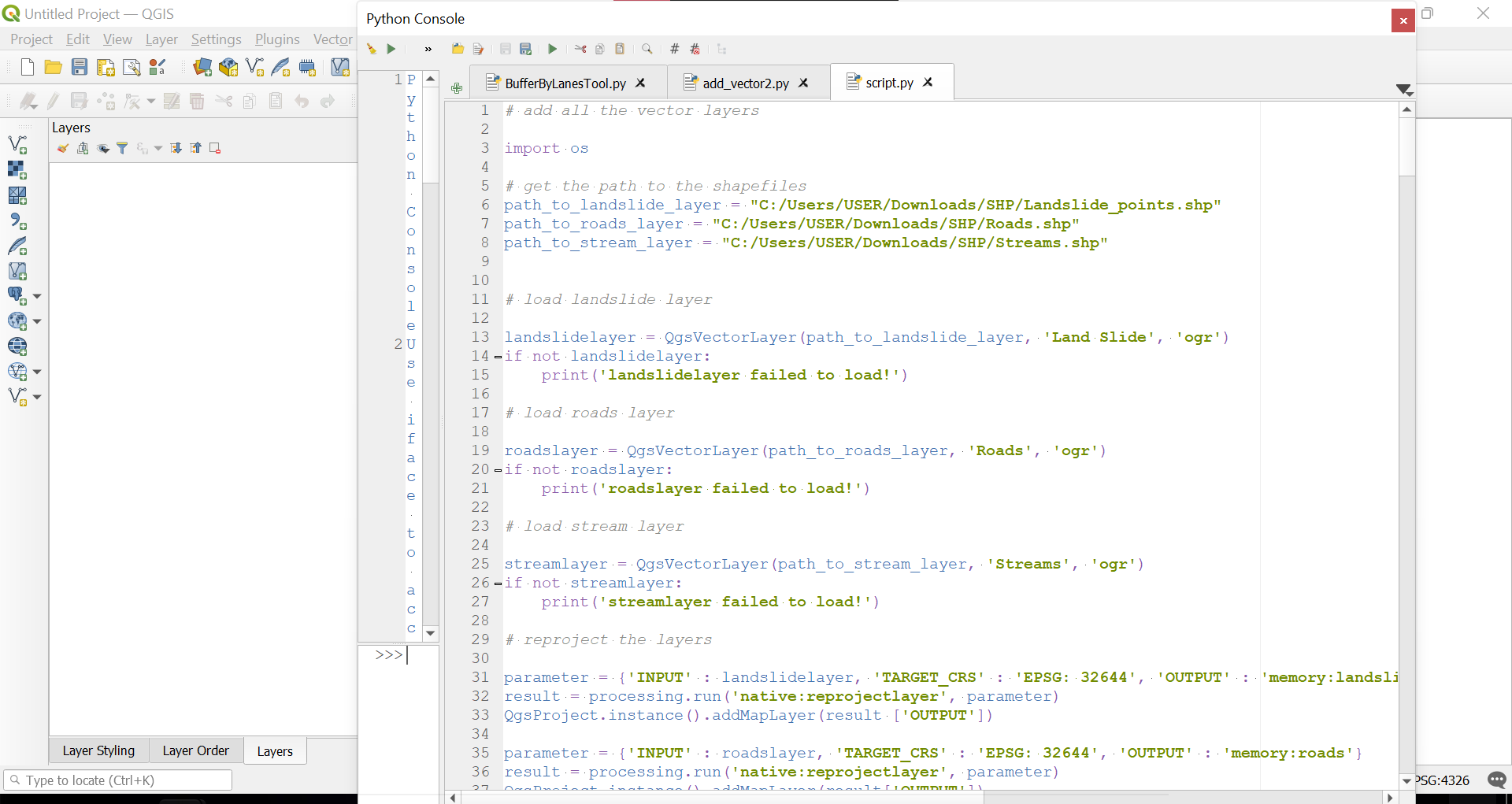
Source: [World Meteorological Organization - Wikipedia](https://en.wikipedia.org/wiki/World_Meteorological_Organization)

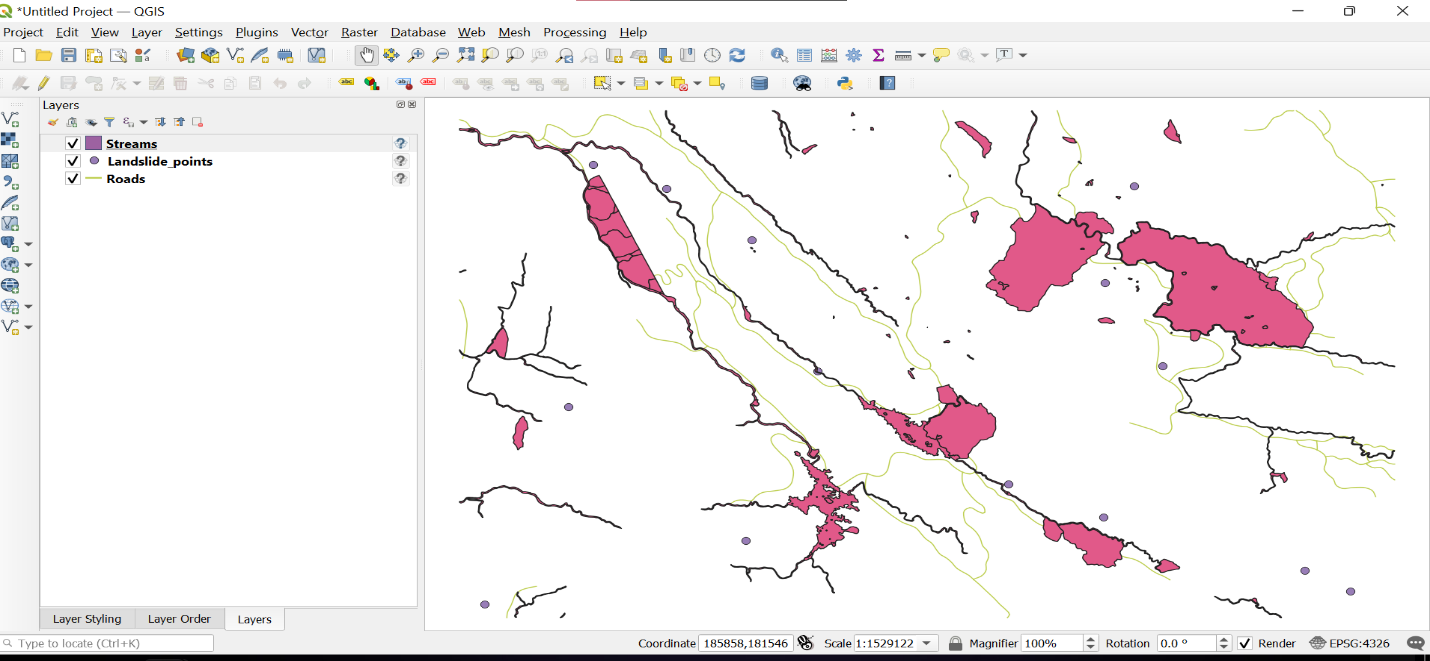
Nuwara Eliya has a subtropical highland climate (Köppen climate classification Cfb) due to its highland location, with no pronounced dry season, a monsoon-like cloudy season, and a mean annual temperature of 16 °C (61 °F). There may be frost at night during the winter months, but due to the high sun angle, it warms up quickly during the day.

Processing

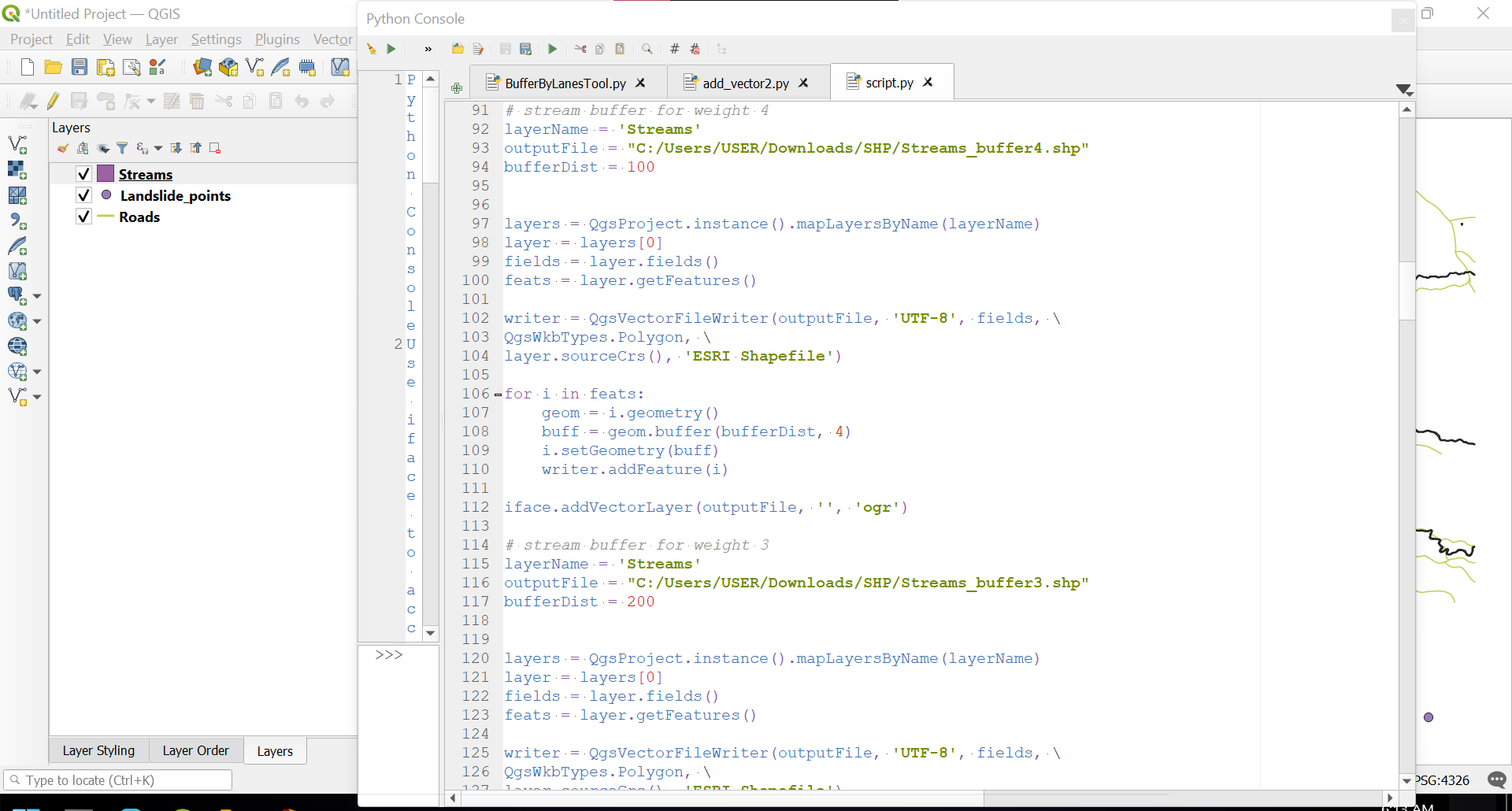
Python QGIS (Pyqgis) was used during this study. PyQGIS is the Python environment within QGIS that includes a set of QGIS libraries as well as Python tools and the ability to run other powerful libraries such as Pandas, Numpy, and Scikit-learn. PyQGIS can be used to perform many tasks that QGIS software can perform with the help of Python algorithms.

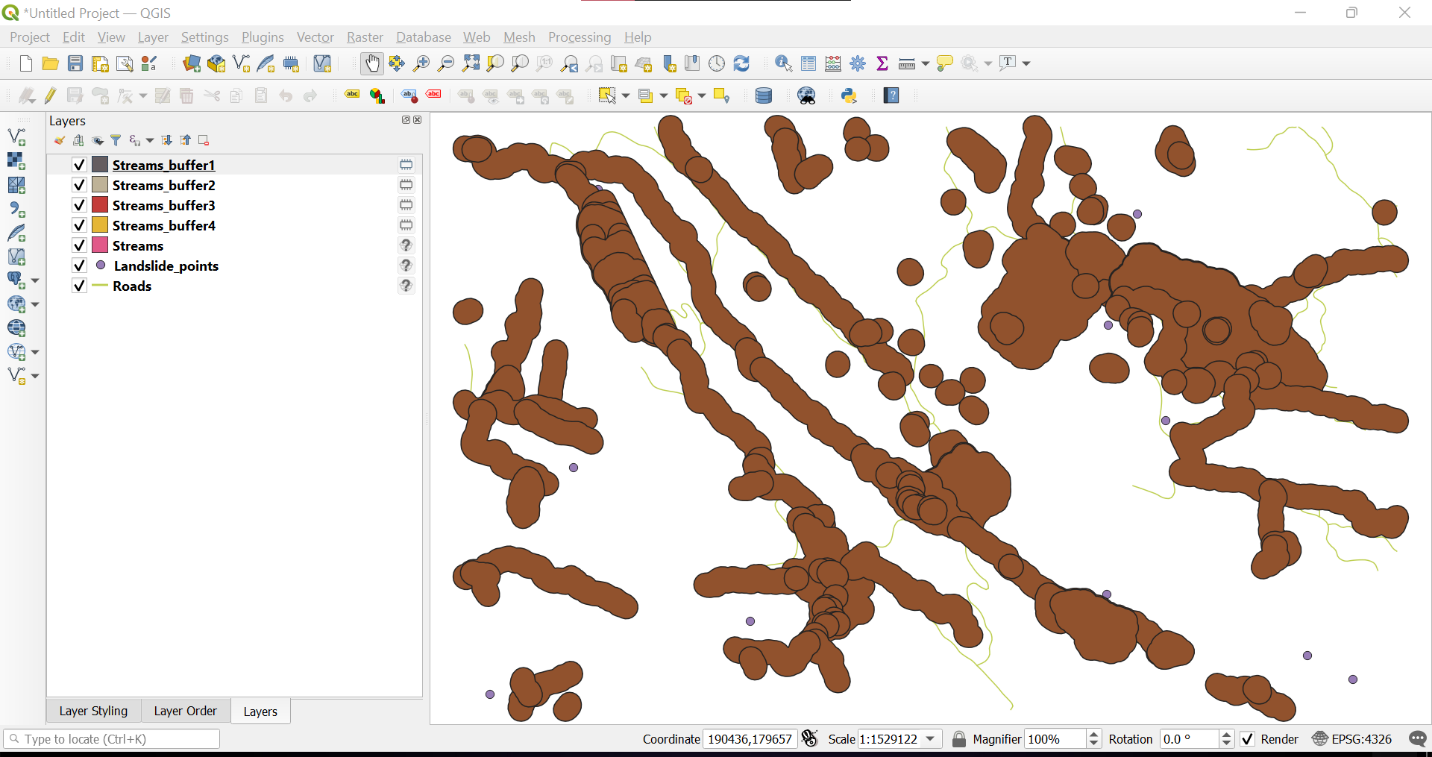
The processes are as follows:

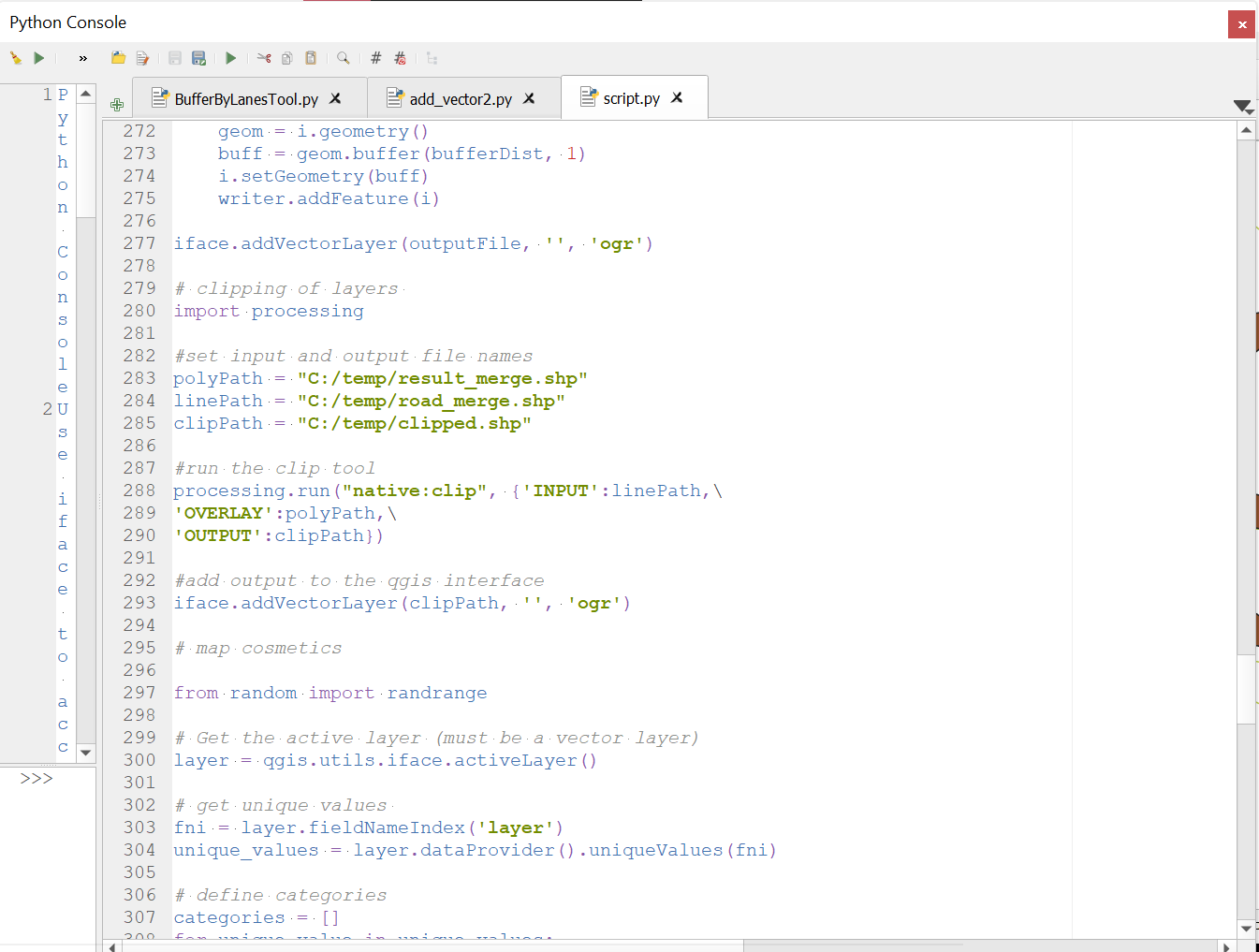
1. Import the data into QGIS environment using python snippet as shown below

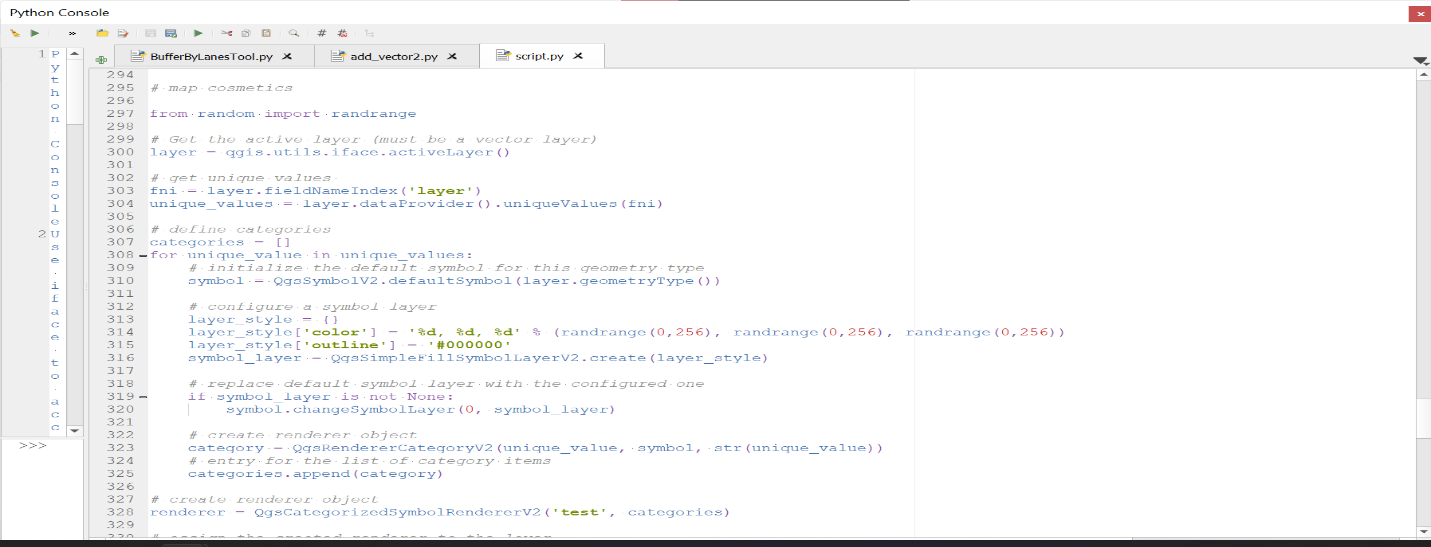


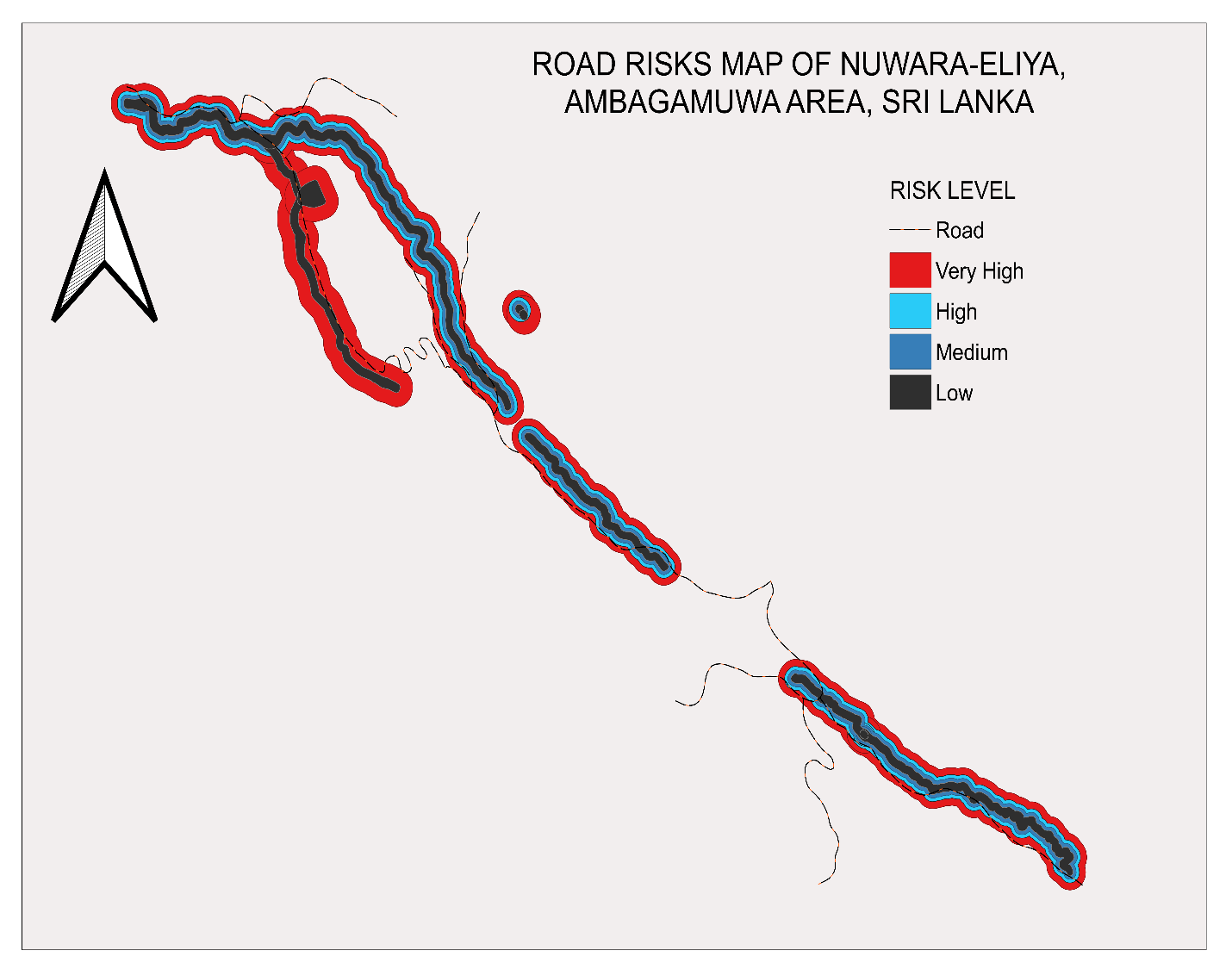
The result the algorithm when run.

1. The next procedure is to buffer the Landslide points and stream. The algorithm used is shown below with the resulting buffer



1. The next step is to clip the corresponding weight assigned to each layer. The algorithm is as shown below.
2. The next step is map cosmetics, in order to design the cartographically.

The code snippet is shown below

After running the code, the map generated is shown below:

Algorithm

# add all the vector layers

import os

# get the path to the shapefiles

path\_to\_landslide\_layer = "C:/Users/USER/Downloads/SHP/Landslide\_points.shp"

path\_to\_roads\_layer = "C:/Users/USER/Downloads/SHP/Roads.shp"

path\_to\_stream\_layer = "C:/Users/USER/Downloads/SHP/Streams.shp"

# load landslide layer

landslidelayer = QgsVectorLayer(path\_to\_landslide\_layer, 'Land Slide', 'ogr')

if not landslidelayer:

print('landslidelayer failed to load!')

# load roads layer

roadslayer = QgsVectorLayer(path\_to\_roads\_layer, 'Roads', 'ogr')

if not roadslayer:

print('roadslayer failed to load!')

# load stream layer

streamlayer = QgsVectorLayer(path\_to\_stream\_layer, 'Streams', 'ogr')

if not streamlayer:

print('streamlayer failed to load!')

# reproject the layers

parameter = {'INPUT' : landslidelayer, 'TARGET\_CRS' : 'EPSG: 32644', 'OUTPUT' : 'memory:landslide'}

result = processing.run('native:reprojectlayer', parameter)

QgsProject.instance().addMapLayer(result ['OUTPUT'])

parameter = {'INPUT' : roadslayer, 'TARGET\_CRS' : 'EPSG: 32644', 'OUTPUT' : 'memory:roads'}

result = processing.run('native:reprojectlayer', parameter)

QgsProject.instance().addMapLayer(result['OUTPUT'])

parameter = {'INPUT' : streamlayer, 'TARGET\_CRS' : 'EPSG: 32644', 'OUTPUT' : 'memory:streamlayer'}

result = processing.run('native:reprojectlayer', parameter)

QgsProject.instance().addMapLayer(result['OUTPUT'])

# create buffers for each layers according to the weights

# roads buffer with weight 2

layerName = 'Roads'

outputFile = "C:/Users/USER/Downloads/SHP/Roads\_buffer2.shp"

bufferDist = 50

layers = QgsProject.instance().mapLayersByName(layerName)

layer = layers[0]

fields = layer.fields()

feats = layer.getFeatures()

writer = QgsVectorFileWriter(outputFile, 'UTF-8', fields, \

QgsWkbTypes.LineString, \

layer.sourceCrs(), 'ESRI Shapefile')

for i in feats:

geom = i.geometry()

buff = geom.buffer(bufferDist, 2)

i.setGeometry(buff)

writer.addFeature(i)

iface.addVectorLayer(outputFile, '', 'ogr')

# roads buffer with weight 1

layerName = 'Roads'

outputFile = "C:/Users/USER/Downloads/SHP/Roads\_buffer1.shp"

bufferDist = 100

layers = QgsProject.instance().mapLayersByName(layerName)

layer = layers[0]

fields = layer.fields()

feats = layer.getFeatures()

writer = QgsVectorFileWriter(outputFile, 'UTF-8', fields, \

QgsWkbTypes.LineString, \

layer.sourceCrs(), 'ESRI Shapefile')

for i in feats:

geom = i.geometry()

buff = geom.buffer(bufferDist, 1)

i.setGeometry(buff)

writer.addFeature(i)

iface.addVectorLayer(outputFile, '', 'ogr')

# stream buffer for weight 4

layerName = 'Streams'

outputFile = "C:/Users/USER/Downloads/SHP/Streams\_buffer4.shp"

bufferDist = 100

layers = QgsProject.instance().mapLayersByName(layerName)

layer = layers[0]

fields = layer.fields()

feats = layer.getFeatures()

writer = QgsVectorFileWriter(outputFile, 'UTF-8', fields, \

QgsWkbTypes.Polygon, \

layer.sourceCrs(), 'ESRI Shapefile')

for i in feats:

geom = i.geometry()

buff = geom.buffer(bufferDist, 4)

i.setGeometry(buff)

writer.addFeature(i)

iface.addVectorLayer(outputFile, '', 'ogr')

# stream buffer for weight 3

layerName = 'Streams'

outputFile = "C:/Users/USER/Downloads/SHP/Streams\_buffer3.shp"

bufferDist = 200

layers = QgsProject.instance().mapLayersByName(layerName)

layer = layers[0]

fields = layer.fields()

feats = layer.getFeatures()

writer = QgsVectorFileWriter(outputFile, 'UTF-8', fields, \

QgsWkbTypes.Polygon, \

layer.sourceCrs(), 'ESRI Shapefile')