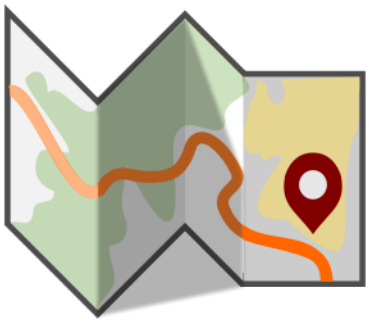
Find Neighbor Polygons in a Layer

QGIS Tutorials and Tips



Author

Ujaval Gandhi

http://google.com/+UjavalGandhi

Translations by

SongHyun Choi

인접 폴리곤 찾기

레이어에서 각 폴리곤과 인접한 폴리곤이 어디에 있는지 모두 찾고자 하는 경우가 있습니다. QGIS에서 간단한 파이썬 스크립트를 사용해서 이 것을 할 수 있고 더 많은 것들을 수행할 수 있습니다. 여기 예제 스크립트는 레이어에서 각 폴리곤과 경계를 공유하는 모든 폴리곤을 찾을 수 있고 또한 그것들의 이름을 속성 테이블에 추가할 수 있습니다. 추가적으로 스크립트는 또한 모든 이웃한 폴리곤으로부터 선택한 속성을 요약합니다.

과업 개요

어떻게 스크립트가 작동하는지 살펴보기 위해 국가의 폴리곤 레이어를 사용하고 경계를 공유한 나라를 찾을 것입니다. 또한 이웃한 나라의 전체 인구수를 계산하고자 합니다.

데이터 획득

Natural Earth의 Admin 0 – Countries 데이터셋을 사용할 것입니다.

Download the Admin 0 – countries shapefile..

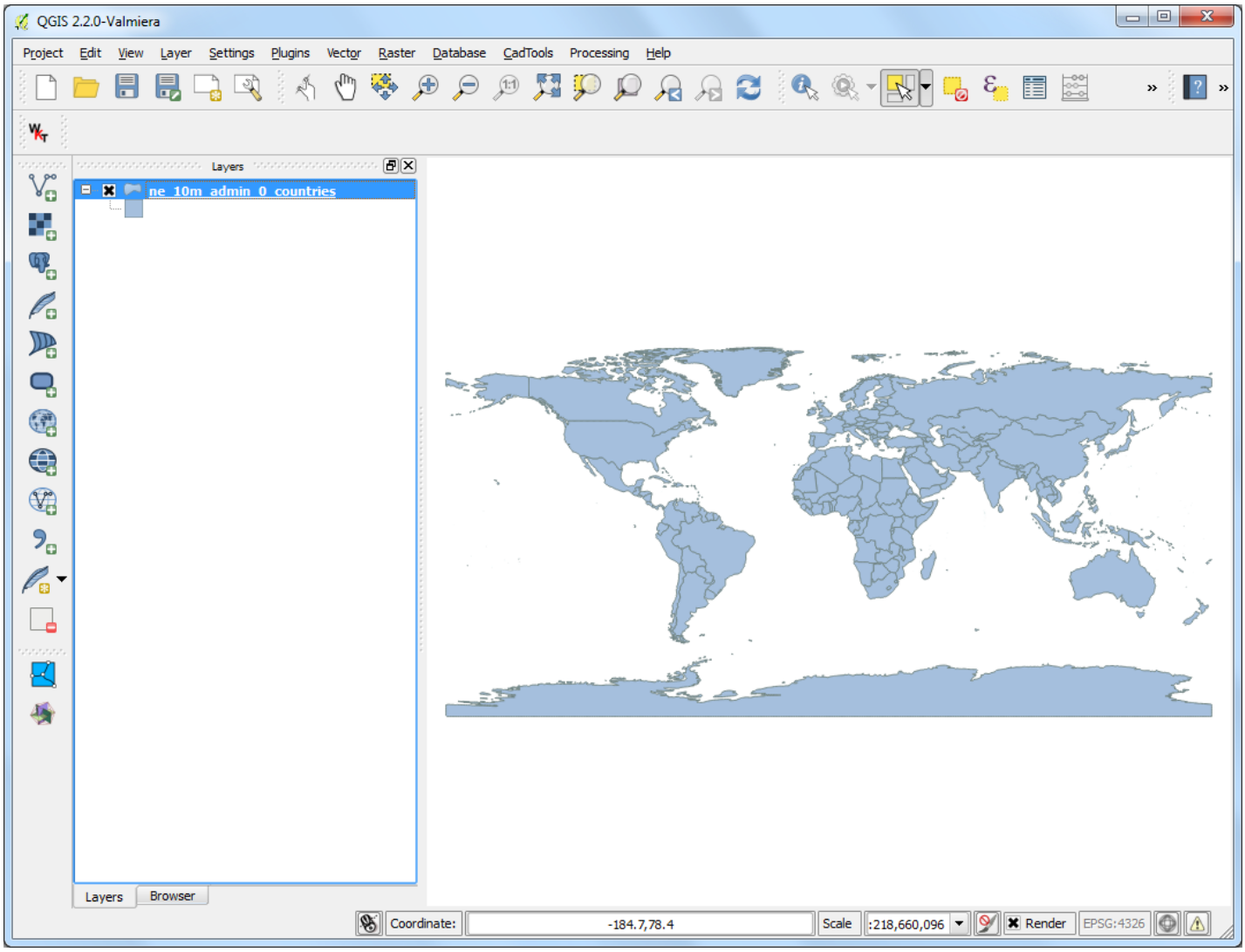
데이터 출처 : [NATURALEARTH]

스크립트 얻기

Download the ***`neighbors.py script`*** and save it to your disk.

과정

1. 메뉴 레이어 --> 벡터 데이터 추가 ``ne_10m_admin_0_countries``로 가서 ``ne_10m_admin_0_countries``레이어를 불러옵니다.

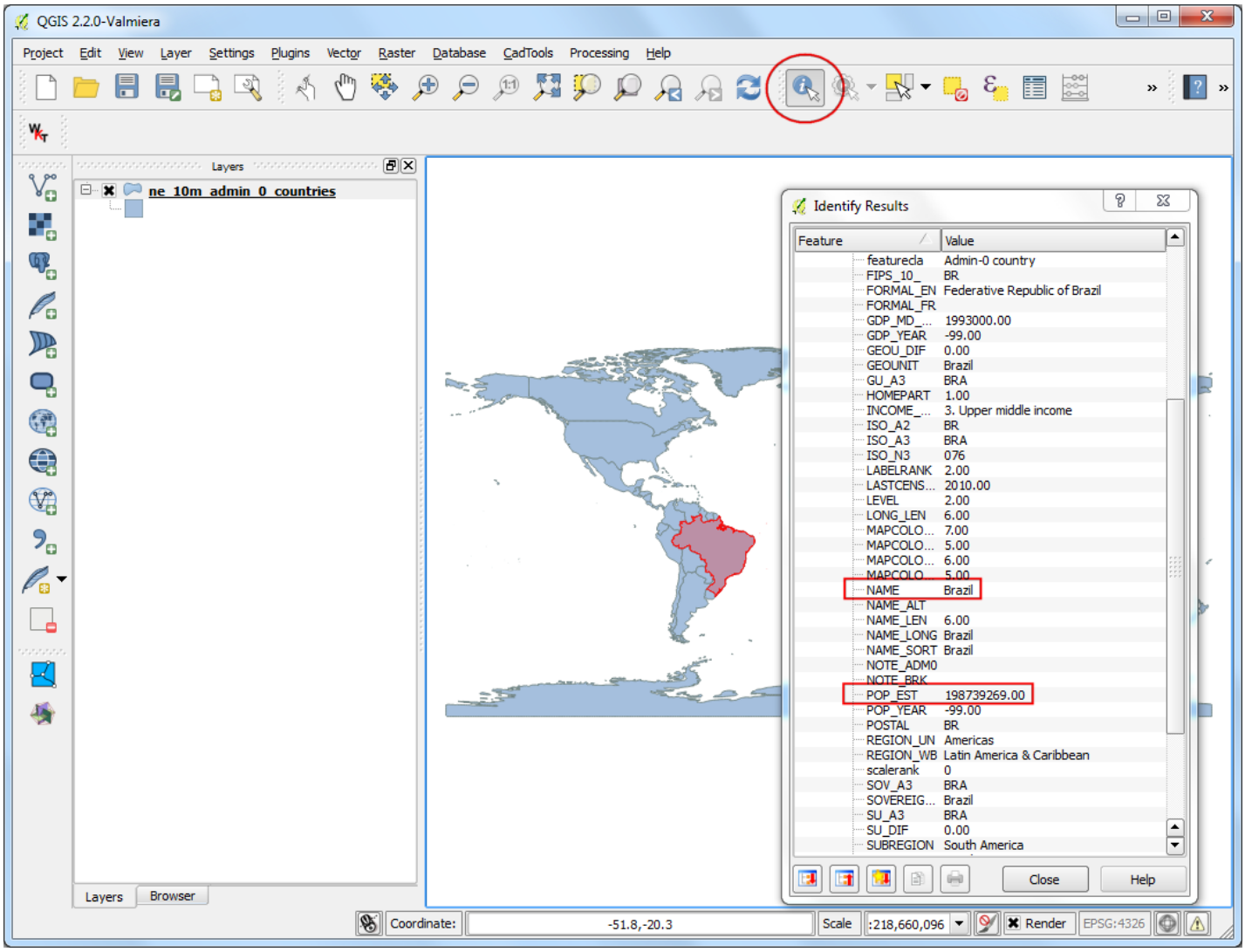


Coordinate: -184.7,78.4

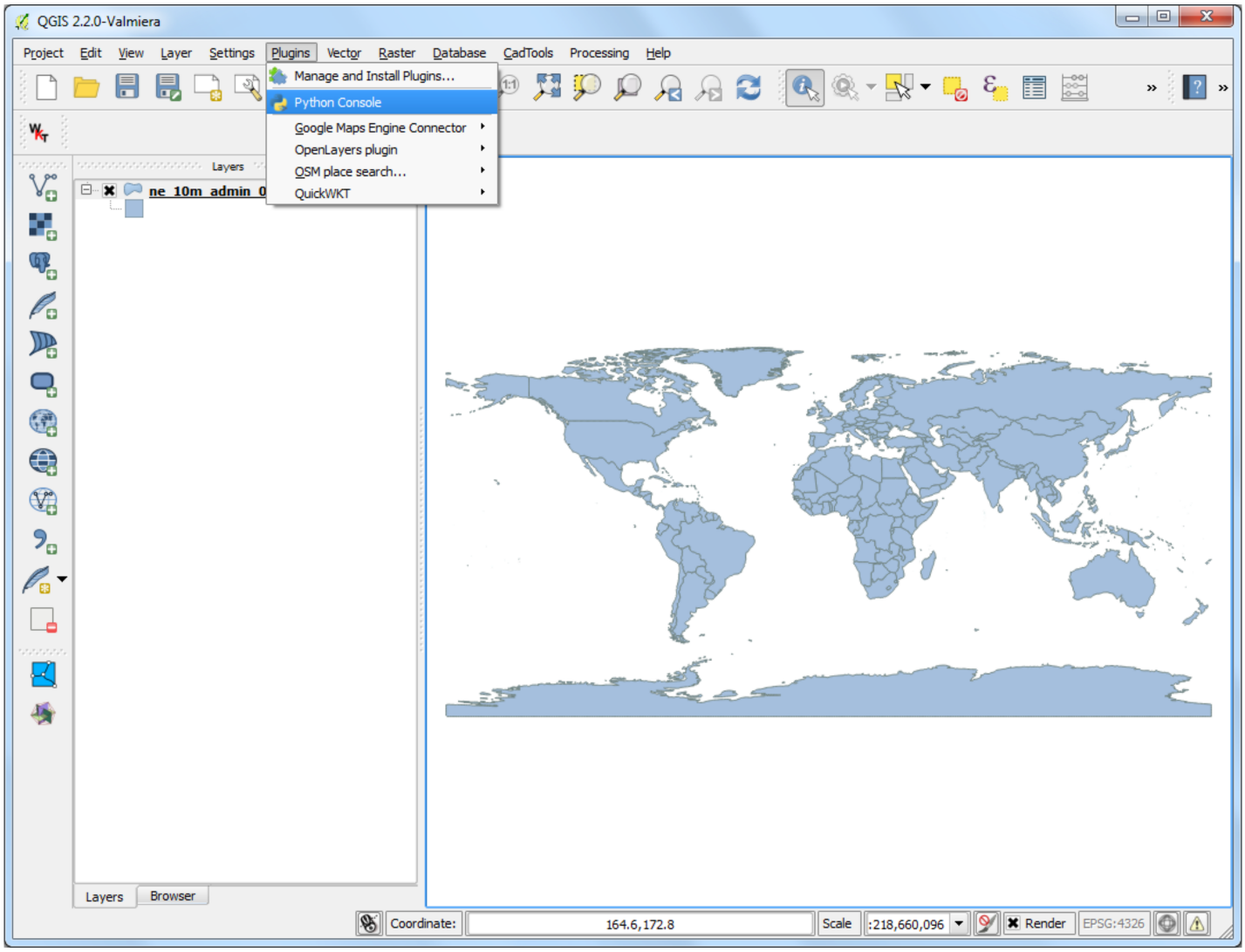
Scale :218,660,096

EPSG:4326

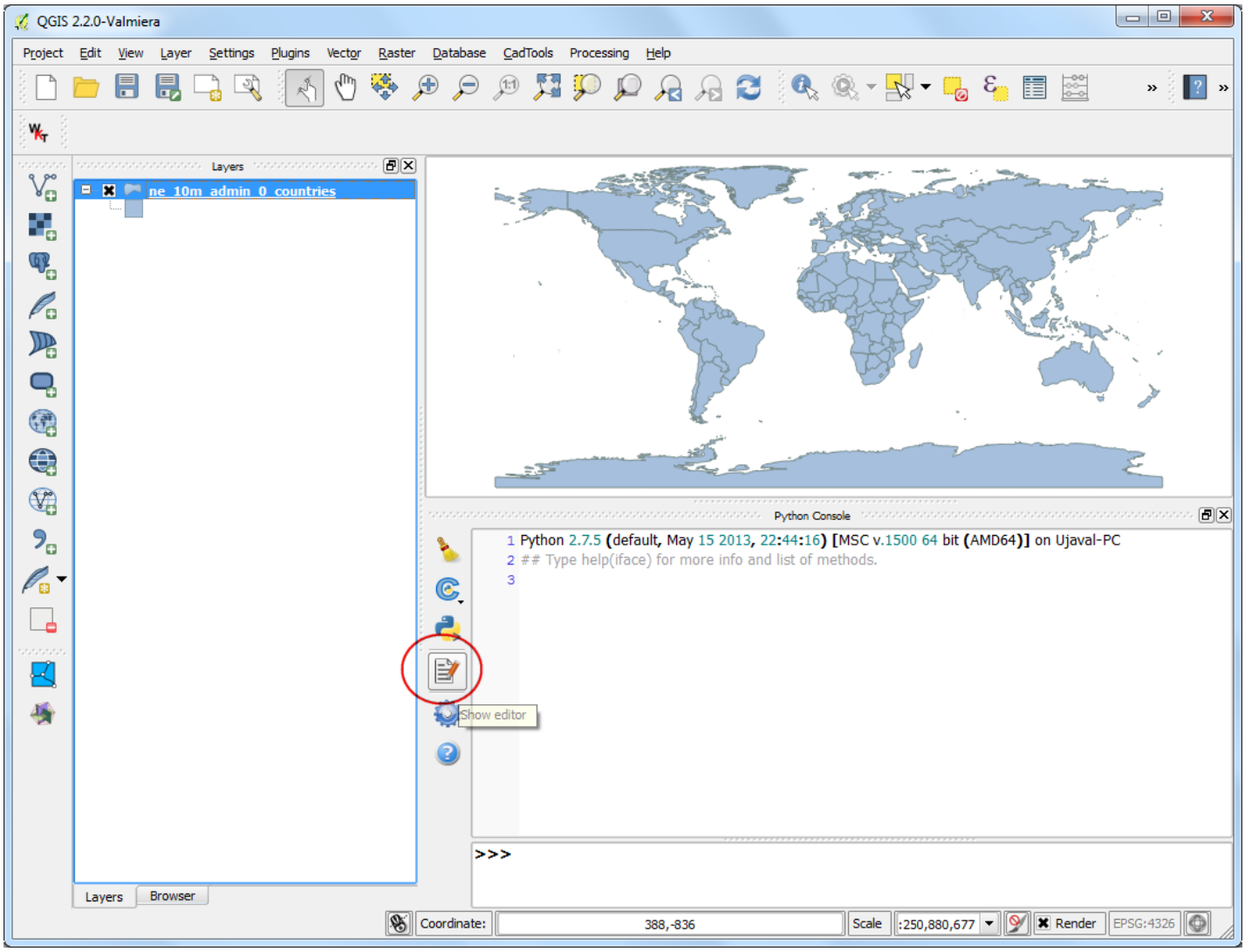
2. 실행을 하기 위해서 스크립트는 2개의 필드를 사용합니다. name 필드와 총합을 하려는 필드. 객체 확인 Identify 툴을 사용하여 아무 객체나 누릅니다. 그리고 속성을 검사합니다. 이 경우에는 name 필드가 **NAME**이고 인구수를 총계하고자 하는 것은 **POP_EST**필드입니다.



3. □□ □□□□ --> Phtyon □□ :menuselection:`Plugins --> Python Console`□ □□□□.

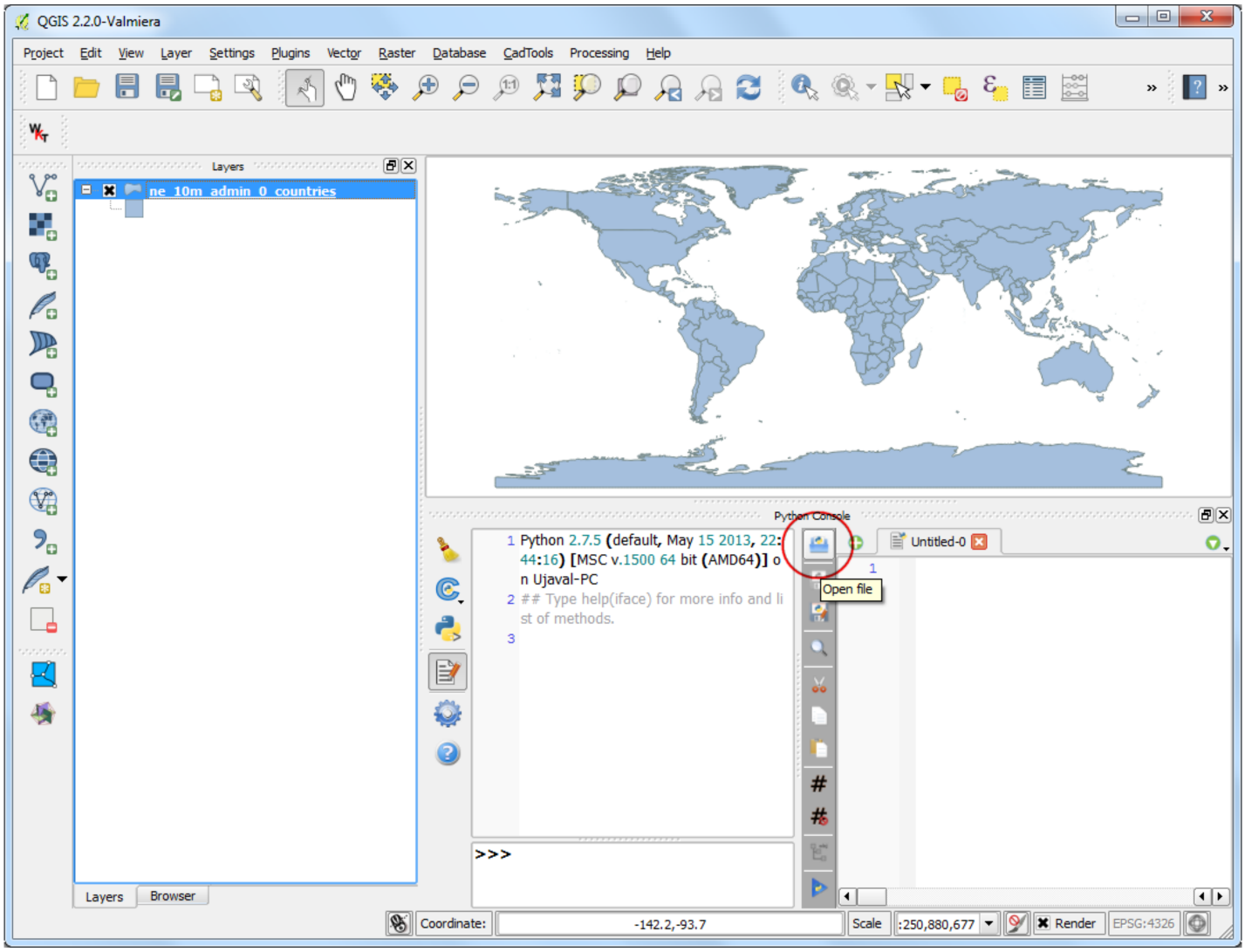


4. Phthon 콘솔 창에서 편집기 보이기 Show Editor 단추를 누릅니다 .

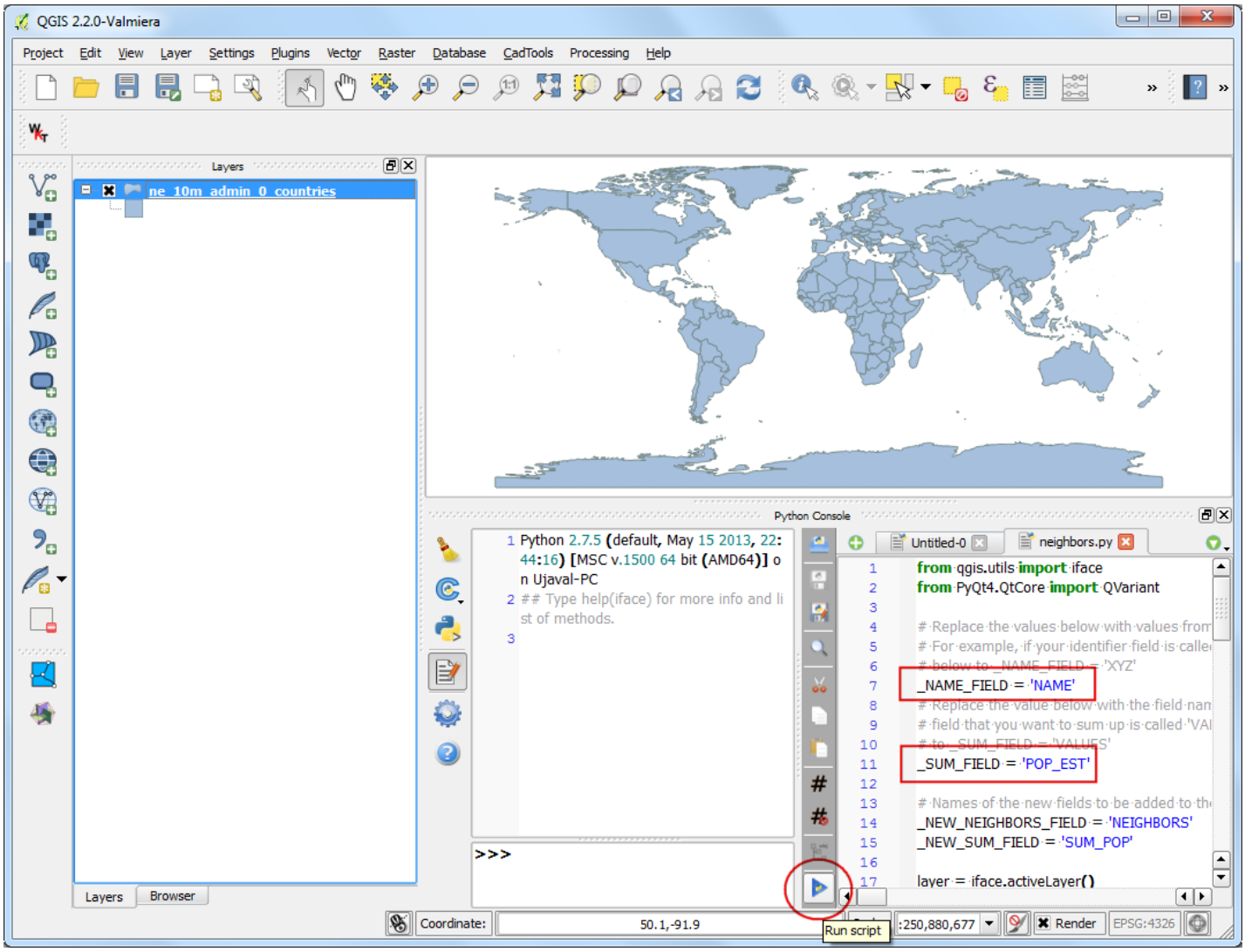


```
Python 2.7.5 (default, May 15 2013, 22:44:16) [MSC v.1500 64 bit (AMD64)] on Ujaval-PC
## Type help(iface) for more info and list of methods.

>>>
```

5. 편집기 Editor` 패널에서 파일 열기 :guilabel:`Open file 단추를 누르고 내려받은 ***neighbors.py*** 스크립트는 찾아서 열기 :guilabel:`Open`를 누릅니다.



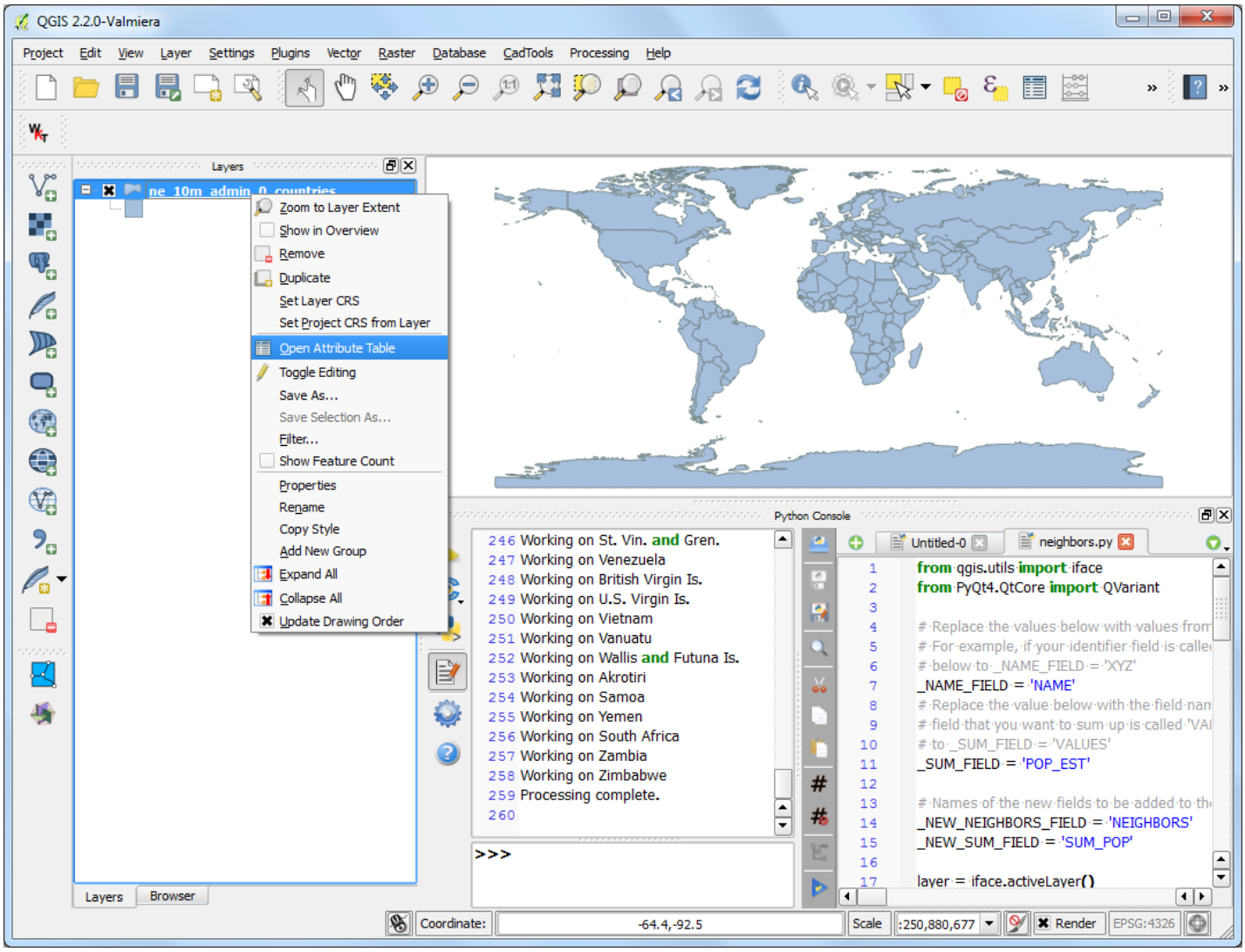
6. 일단 스크립트가 불러들여지면 , 가지고 있는 레이어에서 속성에 맞는 ***_NAME_FIELD``■ ``_SUM_FIELD*** 값을 변경합니다 . 만약 ``ne_10m_admin_0_countries`` 레이어로 작업을 하고자 한다면 그 레이어를 있는 그대로 두십시오 .만약 어떤 것을 변경했다면 편집기 Editor`패널에서 저장 :guilabel:`Save`을 누릅니다 . 이제 스크립트를 실행하기 위해서 스크립트 실행 :guilabel:`Run script 단추를 누릅니다 .



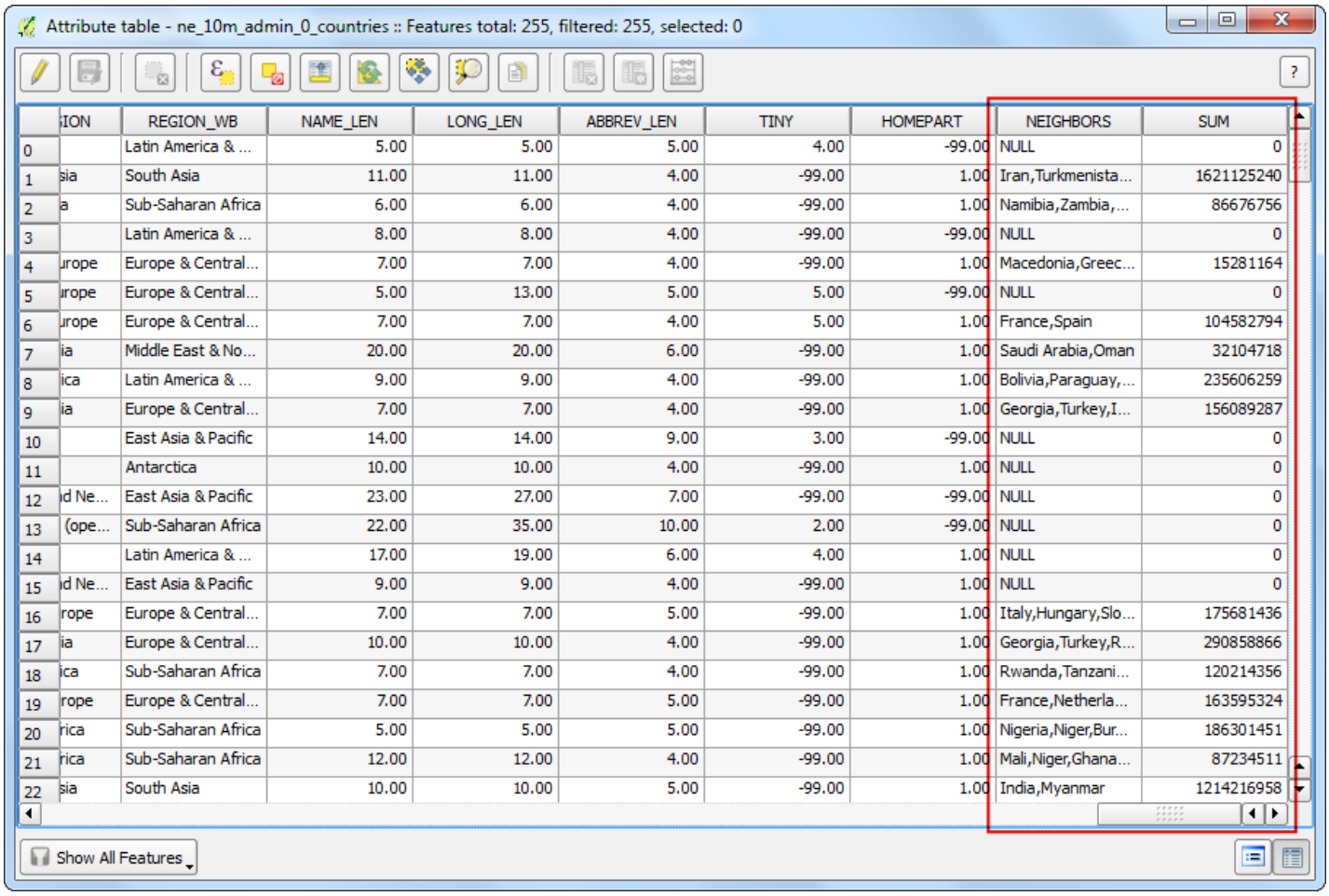
```
Python 2.7.5 (default, May 15 2013, 22:44:16) [MSC v.1500 64 bit (AMD64)] on Ujaval-PC
## Type help(iface) for more info and list of methods.
>>>
```

```
from qgis.utils import iface
from PyQt4.QtCore import QVariant

# Replace the values below with values from
# For example, if your identifier field is calle
# below to _NAME_FIELD = 'XYZ'
_NAME_FIELD = 'NAME'
# Replace the value below with the field nam
# field that you want to sum up is called 'VA
# to _SUM_FIELD = 'VALUES'
_SUM_FIELD = 'POP_EST'

# Names of the new fields to be added to th
_NEW_NEIGHBORS_FIELD = 'NEIGHBORS'
_NEW_SUM_FIELD = 'SUM_POP'

layer = iface.activeLayer()
```

7. 일단 스크립트가 종료되면 ``ne_10m_admin_0_countries`` 레이어를 우측클릭하고 속성 테이블 열기 :guilabel:`Open Attribute Table`를 선택합니다.



8. You will notice 2 new attributes called ***NEIGHBORS*** and ***SUM***. These were added by the script.



	ION	REGION_WB	NAME_LEN	LONG_LEN	ABBREV_LEN	TINY	HOMEPART	NEIGHBORS	SUM
0		Latin America & ...	5.00	5.00	5.00	4.00	-99.00	NULL	0
1	sia	South Asia	11.00	11.00	4.00	-99.00	1.00	Iran,Turkmenista...	1621125240
2	a	Sub-Saharan Africa	6.00	6.00	4.00	-99.00	1.00	Namibia,Zambia,...	86676756
3		Latin America & ...	8.00	8.00	4.00	-99.00	-99.00	NULL	0
4	urope	Europe & Central...	7.00	7.00	4.00	-99.00	1.00	Macedonia,Greec...	15281164
5	urope	Europe & Central...	5.00	13.00	5.00	5.00	-99.00	NULL	0
6	urope	Europe & Central...	7.00	7.00	4.00	5.00	1.00	France,Spain	104582794
7	ia	Middle East & No...	20.00	20.00	6.00	-99.00	1.00	Saudi Arabia,Oman	32104718
8	ica	Latin America & ...	9.00	9.00	4.00	-99.00	1.00	Bolivia,Paraguay,...	235606259
9	ia	Europe & Central...	7.00	7.00	4.00	-99.00	1.00	Georgia,Turkey,I...	156089287
10		East Asia & Pacific	14.00	14.00	9.00	3.00	-99.00	NULL	0
11		Antarctica	10.00	10.00	4.00	-99.00	1.00	NULL	0
12	d Ne...	East Asia & Pacific	23.00	27.00	7.00	-99.00	-99.00	NULL	0
13	(ope...	Sub-Saharan Africa	22.00	35.00	10.00	2.00	-99.00	NULL	0
14		Latin America & ...	17.00	19.00	6.00	4.00	1.00	NULL	0
15	d Ne...	East Asia & Pacific	9.00	9.00	4.00	-99.00	1.00	NULL	0
16	rope	Europe & Central...	7.00	7.00	5.00	-99.00	1.00	Italy,Hungary,Slo...	175681436
17	ia	Europe & Central...	10.00	10.00	4.00	-99.00	1.00	Georgia,Turkey,R...	290858866
18	ica	Sub-Saharan Africa	7.00	7.00	4.00	-99.00	1.00	Rwanda,Tanzani...	120214356
19	rope	Europe & Central...	7.00	7.00	5.00	-99.00	1.00	France,Netherla...	163595324
20	rica	Sub-Saharan Africa	5.00	5.00	5.00	-99.00	1.00	Nigeria,Niger,Bur...	186301451
21	rica	Sub-Saharan Africa	12.00	12.00	4.00	-99.00	1.00	Mali,Niger,Ghana...	87234511
22	sia	South Asia	10.00	10.00	5.00	-99.00	1.00	India,Myanmar	1214216958

□□□ □□ □□□□□□□□. □□□ □□ □□□ □ □□□□.

```
#############################################################################
# Copyright 2014 Ujaval Gandhi
#
#This program is free software; you can redistribute it and/or
#modify it under the terms of the GNU General Public License
#as published by the Free Software Foundation; either version 2
#of the License, or (at your option) any later version.
#
#This program is distributed in the hope that it will be useful,
#but WITHOUT ANY WARRANTY; without even the implied warranty of
#MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#GNU General Public License for more details.
#
#You should have received a copy of the GNU General Public License
#along with this program; if not, write to the Free Software
#Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
#
#############################################################################
from qgis.utils import iface
from PyQt4.QtCore import QVariant

# Replace the values below with values from your layer.
# For example, if your identifier field is called 'XYZ', then change the line
# below to _NAME_FIELD = 'XYZ'
```

```
_NAME_FIELD = 'NAME'
# Replace the value below with the field name that you want to sum up.
# For example, if the # field that you want to sum up is called 'VALUES', then
# change the line below to _SUM_FIELD = 'VALUES'
_SUM_FIELD = 'POP_EST'

# Names of the new fields to be added to the layer
_NEW_NEIGHBORS_FIELD = 'NEIGHBORS'
_NEW_SUM_FIELD = 'SUM'

layer = iface.activeLayer()

# Create 2 new fields in the layer that will hold the list of neighbors and sum
# of the chosen field.
layer.startEditing()
layer.dataProvider().addAttributes(
        [QgsField(_NEW_NEIGHBORS_FIELD, QVariant.String),
         QgsField(_NEW_SUM_FIELD, QVariant.Int)])
layer.updateFields()
# Create a dictionary of all features
feature_dict = {f.id(): f for f in layer.getFeatures()}

# Build a spatial index
index = QgsSpatialIndex()
for f in feature_dict.values():
    index.insertFeature(f)

# Loop through all features and find features that touch each feature
for f in feature_dict.values():
    print 'Working on %s' % f[_NAME_FIELD]
    geom = f.geometry()
    # Find all features that intersect the bounding box of the current feature.
    # We use spatial index to find the features intersecting the bounding box
    # of the current feature. This will narrow down the features that we need
    # to check neighboring features.
    intersecting_ids = index.intersects(geom.boundingBox())
    # Initalize neighbors list and sum
    neighbors = []
    neighbors_sum = 0
    for intersecting_id in intersecting_ids:
        # Look up the feature from the dictionary
        intersecting_f = feature_dict[intersecting_id]

        # For our purpose we consider a feature as 'neighbor' if it touches or
        # intersects a feature. We use the 'disjoint' predicate to satisfy
        # these conditions. So if a feature is not disjoint, it is a neighbor.
        if (f != intersecting_f and
            not intersecting_f.geometry().disjoint(geom)):
            neighbors.append(intersecting_f[_NAME_FIELD])
            neighbors_sum += intersecting_f[_SUM_FIELD]
    f[_NEW_NEIGHBORS_FIELD] = ','.join(neighbors)
    f[_NEW_SUM_FIELD] = neighbors_sum
    # Update the layer with new attribute values.
    layer.updateFeature(f)

layer.commitChanges()
print 'Processing complete.'
```