توظيف البيانات مفتوحة المصدر في تمثيل خرائط الكثافة السكانية في العراق بأستخدام البرمجة بلغة python

أ.م.د. زينب محمد أمين

الجامعة العراقية – كلية الاداب – قسم الجغرافية ونظم المعلومات الجغرافية

zainab.m.ameen@aliraqia.edu.iq

Mob.009647702631627

Employing open source data to represent population density map In Iraq using programming in Python

Asst.prof.Dr. Zainab Muhammad Amin

zainab.m.ameen@aliraqia.edu.iq

Iraqi University / College of Arts / Department of Geography

المستخلص

ان فكرة البيانات مفتوحة المصدر تأتي مترادفة مع مفردات في ذات السياق كالكيانات المادية المفتوحة، المحتوى المفتوح، التعليم المفتوح، مصادر التعليم المفتوحة، المعرفة المفتوحة، إتاحة وفتح إمكانية الوصول، العلوم المفتوحة، والشبكة المفتوحة، وكلها تصب في اتجاه واحد وهو تحقيق الفائدة المتبادلة انطلاقاً من مبدأ ان العلم يبدأ من حيث انتهى الاخرون.

في ظل الثورة المعلوماتية التي يشهدها العالم اصبحت هنالك حاجة اكبر الى تعلم مهارة الحصول على البيانات ومن ثم التعامل مع هذا الكم الهائل من البيانات من اجل انتقاء المناسب منها وتوظيفه لخدمة مجال البحث الذي يرغب الباحث انجازه ، وبالتالي فإن الخطوة الاولى لأنجاز اي بحث هي الاجابة عن مجموعة من الاسئلة:-

ماهي البيانات التي تخدم البحث ؟ هل هنالك امكانية الحصول على هذه البيانات؟ هل يمكن الاستفادة من البيانات التي نحصل عليها بصورة مباشرة من مصدرها وكما هي؟ ام بإجراء بعض التغييرات عليها وهو ما يطلق عليه اصطلاحاً (تنظيف البيانات).

في بحثنا قمنا بأختيار منطقة البحث متمثلة بالعراق وحسب الاقضية (القضاء هو المستوى الاداري الاقل من المحافظة) في محاولة منا للافادة من البيانات مفتوحة المصدر ومن اكثر من موقع وسحبها عن طريق البرمجة بلغة بايثون ومن ثم الدمج بين هذه المصادر بعد سلسلة من العمليات (تنظيف البيانات) وبالتالي توضيحها في خرائط للكثافة السكانية بأعتماد عدد من مكتبات بايثون. الخرائط التي تم انتاجها هنا قد تحتاج الى بعض الاضافات اولمسات الحس الفني التي تختلف من باحث الى اخر لكننا في المجمل حاولنا انتاج خرائط اعتماداً على بيانات كان لا يمكن تحميلها من المواقع الخاصة بها بصورة مباشرة وذلك باستخدام بايثون.

كلمات مفتاحية:- البيانات مفتوحة المصدر – مكتبات بايثون – الحلقات التكرارية (for loop)

Abstract

The idea of open source data is synonymous with terms in the same context, such as open physical entities, open content, open education, open educational resources, open knowledge, open access, open science, and the open web. They all aim in one direction, which is achieving mutual benefit based on the principle that science begins where others left off.

In light of the information revolution that the world is witnessing, there has become a greater need to learn the skill of obtaining data and then dealing with this huge amount of data in order to select the appropriate ones and employ them to serve the field of research that the researcher wishes to accomplish. Therefore, the first step in completing any research is to answer the questions. A set of questions-:

What data serves the research? Is there a possibility to obtain this data? Can we benefit from the data we obtain directly from its source and as it is? Or by making some changes to it, which is called (data cleaning).

In our research, we chose the research area represented by Iraq and according to the district (the district is the administrative level lower than the governorate) in an attempt to benefit from open source data from more than one site and pull it through programming in the Python language and then combine these sources after a series of operations (data cleaning). and thus clarified in population density maps by relying on a number of Python libraries. The maps produced here may need some additions or touches of artistic sense that differ from one researcher to another, but in general we tried to produce maps based on data that could not be downloaded directly from their respective sites using Python.

Keywords:- Open source data - Python libraries - for loop

المقدمة

تنظيف البيانات هو عملية إصلاح أو إزالة البيانات غير الصحيحة أو التالفة أو المنسقة بشكل غير صحيح أو المكررة أو غير المكتملة أو غير المطلوبة في البحث ضمن مجموعة البيانات، ذلك انه عند الجمع بين مصادر بيانات متعددة ، هناك العديد من الفرص لتكرار البيانات أو تسميتها بشكل خاطئ، فإذا كانت البيانات غير صحيحة ، فإن النتائج والخوارزميات لا يمكن الاعتماد عليها ، على الرغم من أنها قد تبدو صحيحة. ثم انه لا توجد طريقة واحدة مطلقة لوصف الخطوات الدقيقة في عملية تنظيف البيانات لأن العمليات ستختلف من مجموعة بيانات إلى مجموعة بيانات. ولكن من الأهمية بمكان إنشاء نموذج لعملية تنظيف البيانات الخاصة بنا حتى نعرف أننا تقوم بذلك بالطريقة الصحيحة في كل مرة.[1]

البيانات التي تم اعتمادها في البحث هي نوعين:-

1 – بيانات غير مكانية أو بيانات وصفية أو خصائص attribute or non spatial data –:تضم أية بيانات اضافية عن كل كائن (بخلاف موقعه علي سطح الأرض) و تخضع لعمليات التحليل الاحصائي

2 – يانات مكانية Spatial data :- تحدد الموقع الجغرافي/المكاني للكائن علي سطح الأرض و تخضع لعمليات التحليل المكاني[2]

في ورقتنا البحثية هذه سيتم اعتماد المصادر التالية للبيانات

1 – البيانات المكانية الوصفية / https://www.citypopulation.de/en/iraq

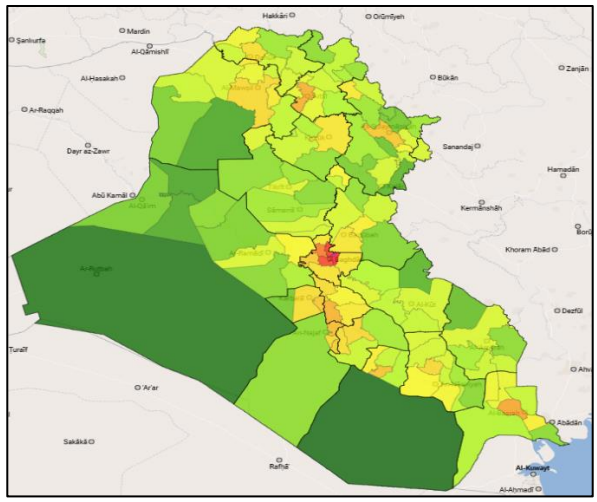
هذا الموقع مختص بالبيانات السكانية لأي دولة في العالم ولسنوات تختلف من دولة لأخرى (شكل 1) وللعراق تتوفر للاعوام 2009 و 2018، في بحثنا سنقوم بالتعامل مع بيانات اعداد سكان العراق وحسب الاقضية البالغ عددها (101) قضاء وللعام 2018.

2 – البيانات المكانية :- http://www.diva-gis.org/gdata هذه البيانات نحصل عليها من DIVA-GIS هو برنامج وموقع بيانات مكانية مجاني ومفتوح المصدر

لنظم المعلومات الجغرافية (GIS) وهو مفيد بشكل خاص للمهتمين برسم خرائط مشاريع الحفاظ على التنوع البيولوجي. ويسمح بتحميل البيانات المكانية لأي منطقة في العالم بصيغة shape file وبصيغ اخرى نتمكن من معالجتها والتعامل معها في برامج نظم المعلومات الجغرافية.

لتحقيق اقصى فائدة من بيات الموقعين التاليين سيتم التعامل مع بيانات اعداد السكان (البيانات الوصفية) بتحويلها عن طريق اكواد بايثون الى ملف اكسل قابل للحفظ والقراءة ، ذلك ان هذا الموقع لا يسمح بتحميل البيانات بصورة مباشرة ، الخطوة التالية هي الربط بين البيانت الوصفية والبيانات المكانية للحصول على خرائط الكثافة السكانية للعراق (حسب الاقضية وللاعوام 2009 – 2018) باستخدام البرمجة بلغة بايثون (مفتوحة المصدر).

شكل (1) منطقة الدراسة



المصدر/ https://www.citypopulation.de/en/iraq/admin

معالجة وتحليل النتائج

سيتم معالجة وتحليل نتائج البحث من خلال مبحثين

المبحث الاول:- الجانب النظري

أولاً: اهمية البرمجة بلغة Python في نظم المعلومات الجغرافية:-

تعد لغة بايثون من اهم الادوات المستخدمة في بناء وادوات برامج نظم المعلومات الجغرافية

و يمكن تلخيص اهمية البرمجة بلغة بايثون في نظم المعلومات الجغرافية بالنقاط التالية:-

1. أتمتة المهام المتكررة
2. صنع أدوا ت المعالجات الجغرافية الخاصة بالمستخدم
3. إضافة معالجة جغرافية لتطبيقات الويب
4. تخصيص تطبيقات سطح المكتب
5. توسيع امكانيات نظم المعلومات الجغرافية

وباتالي اصبح لابد لكل من يرغب في تطوير ادوات Arc GIS ان يكون ملما في البرمجة بلغة بايثون اولا ومن ثم توظيف هذا الالمام في مجال نظم المعلومات الجغرافية.

ان من ابرز مميزات العمل البحثي في المرحلة الراهنة هي العمل لساعات اقل وبجهد اقل ، هذا الجهد لا تظهر اهميته الا اذا تم بذله بذكاء اكبر وبالتالي صار لابد من اختصار الجهد. وهذه هي تحديدا اهم ميزة في استخدام لغة بايثون في عمل الاكواد التي تنفذ الاوامر المختلفة.

ثانياً: اهم المكتبات المستخدمة في نظم المعلومات الجغرافية والبحث:-

المكتبات (modules) :- هي مكتبات مستخدمة في بايثون والتي قام بعض المبرمجين بتجميع اكواد داخل ملفات ، فالـــ modules هي ملفات بايثون مكتوب بداخلها classes تم تجميعها من المبرمجين حيث يستطيع المستخدم استيرادها واستخدامها في أي وقت ، ومن الممكن ان يقوم المستخدم بعمل module خاصة به فيقوم بإنشاء ملف يضع داخله classes الخاصة به[3]

يميز بايثون إضافةً إلى الكود السهل والمبسط الذي يمكنك كتابته أو ما يسمى (Syntax)هو وجود مجموعة كبيرة من المكتبات التي يمكنك تضمينها والاستفادة

منها ليصبح تطبيقك البرمجي او الكود أكثر فعالية وقدرة على تأدية مجموعة من المهام الإضافية التي تقدمها تلك المكتبات.

وفيما يلي اهم المكتبات المستخدمة في نظم المعلومات الجغرافية واهمية كل منها :-

1. Arcpy:- هي احدى مكتبات بايثون التي تستخدم للتفاعل مع أدوات ArcGIS، والتي يتم تنفيذها بعد ذلك داخل برامج ArcGIS في تنسيق التعليمات البرمجية الداخلية ، وقاعدة التعليمات البرمجية هذه هي التي تسمح بالتحكم الإضافي في التحليلات الجغرافية المكانية وإنتاج الخرائط، وتستخدم مكتبة ArcPy للتحكم في الأدوات في ArcToolbox ، ولكن لم تتم إعادة كتابة الأدوات في Python ، للأدوات الموجودة مسبقا في Arc GIS ، هذه المكتبة تعطي القدرة على التحكم في مستندات خريطة بامتداد (MXD) وجميع الكائنات التي تتضمنها MXD من وسائل الايضاح والعناوين والصور والطبقات وعرض الخريطة نفسه، كما وتحتوي ArcPy أيضاً على أدوات غير متوفرة في ArcToolbox.[4]
2. Geopandas:- هي مكتبة مفتوحة المصدر تسهل التعامل مع البيانات الجغرافية المكانية في لغة بايثون .تسمح GeoPandas مع مكتبة Pandas بالتعامل مع العمليات المكانية على العمليات الهندسية (Geometric operations)حيث يتم تنفيذ العمليات الهندسية بشكل سلس، و تعتمد Geopandasأيضاً على fiona للوصول إلى الملفات و descartes و matplotlibلعمل المخططات (الاشكال البيانية)[5].
3. GDAL/OGR:- تُستخدم للترجمة بين تنسيقات وامتدادات GIS. تستخدمه QGIS و ArcGIS و ERDAS و ENVI و GRASS GIS وتقريباً جميع برامج GIS
4. RSGISLib :- عبارة عن مجموعة من أدوات الاستشعار عن بعد لمعالجة البيانات Raster وتحليلها.

5. PyProj:- الغرض الرئيسي من مكتبة PyProj هو كيفية عملها مع أنظمة الإسناد المكاني. يمكنه عرض الإحداثيات وتحويلها باستخدام مجموعة من النظم المرجعية الجغرافية. يمكن لـــ PyProj أيضًا إجراء الحسابات والمسافات الجيوديسية لأي مسند معين.

كذلك لدينا مجموعة المكتبات المستخدمة في نظم المعلومات الجغرافية وهي ذات الصلة بعلوم البيانات، ومنها:-

6. NumPy:- تأخذ لغة Python الرقمية (مكتبة NumPy) جدول السمات الخاص بك وتضعه في مصفوفة منظمة. بمجرد أن تصبح في مصفوفة منظمة ، تصبح أسرع بكثير لأي حوسبة علمية. وهذه المكتبة مهمة جدا في التعامل مع البيانات الضخمة.

7. Pandas:- مكتبة تحضى باستخدام واسع جدا لأهميتها في العمليات الاحصائية المختلفة في بيئة نظم المعلومات الجغرافية

8. Matplotlib:- عندما تعمل مع الآلاف من نقاط البيانات ، فإن أفضل شيء قد تفعله أحيانًا هو رسم كل شيء، ويكون ذلك باستخدام مكتبة Matplotlib. يستخدم الإحصائيون مكتبة matplotlib للعرض المرئي، فهي تستخدم في تمثيل الرسوم البيانية والمخططات والخرائط، وتكمن اهميتها في التعامل بيانيا مع البيانات الضخمة.

9. Re (regular expressions):- هي مكتبة التصفية المثلى، عندما يكون هناك سلسلة معينة من البيانات تريد البحث عنها في جدول ، فهذه هي مكتبة الافضل ولكن يمكن أن نأخذ الأمر إلى أبعد من ذلك مثل اكتشاف واستخراج واستبدال مطابقة الأنماط المتشابهة.

10. ReportLab:- واحدة من افضل المكتبات في اعداد التقارير ، إذ تفتقر نظم المعلومات الجغرافية الى انتاج التقارير التفصيلية وبالتالي فإن هذه المكتبة تدعم هذا الجانب.

11. Ipyleaflet:- مكتبة مهمة جدا في انتاج الخرائط التفاعلية وتحميل خرائط الاساس .

12. Folium:- هذه المكتبة كالسابقة تتيح لنا التعامل مع الخرائط التفاعلية، ومن ثم تمنح القدرة على معالجة بياناتك في Python ، ثم يمكنك تصورها باستخدام مكتبة JavaScript مفتوحة المصدر .

13. Geemap:- مكتبة متخصصة لتحليل البيانات في Google Earth وهي واسعة الاستخدام في تحليل بيانات الاستشعار عن بعد.

14. LiDAR:- الغرض منها هو معالجة وتصور بيانات (LiDAR). على سبيل المثال لا الحصر فهي تتضمن أدوات معالجة الخصائص الطوبولوجية وتصفيتها واستخراجها من بيانات نماذج الارتفاع الرقمية (DEM).

15. Scikit:- هذه المكتبة تتيح التعلم الآلي. وهي تعمل مع مكتبة NumPy و SciPy و Matplotlib. لذلك ، فإذا ما كنا نرغب في أي تنقيب عن البيانات أو تصنيفها أو توقعها، فإن مكتبة Scikit تعد خياراً جيداً[6]

المكتبات التي سنستخدمها هنا هي ثلاث مكتبات (pandas – geopandas – matplotlib)

المبحث الثاني:- الجانب العملي

اولاً:- الحصول على البيانات وتنظيفها في Python

الحصول على البيانات وتنظيفها تتمثل ببرنامج نصي نعمل من خلاله على استرداد بيانات العراق السكانية من صفحة https://www.citypopulation.de/en/iraq/admin وتحويلها الى ملف اكسل وحفظه في الحاسوب ومن ثم تنزيل البيانات المكانية الخاصة بالعراق وعلى مستوى الاقضية من موقع http://www.diva-gis.org/gdata عن طريق برنامج نصي ايضا باستخدام محرر Spyder

بفتح المحرر نقوم بعمل ملف العمل وليكن بأسم pop او اي اسم ثم اول ما نقوم به بعد انشاء ملف العمل هو استيراد المكتبات التالية:-

```
import pandas as pd
import geopandas as gpd
```

```
import matplotlib.pyplot as plt
```

بمجرد الضغط على F5 سنلاحظ اشتغال الكود بدون مشاكل

بعد ذلك اقوم بانشاء سطر لاستيراد بيانات اعداد السكان من موقع الويب الخاص بها عن طريق متغير بأسم data هذا المتغير يقرأها عن طريق مكتبة pandas التي اصبح التعامل معها باسم pd

```
data = pd.read_html('https://citypopulation.de/en/iraq/admin/')
```

بهذه الطريقة فإن البيانات لم يتم حفظها في pc الخاص بنا ، بل حفظ فقط رابط البيانات ، ولأجل حفظ البيانات كملف في pc فأننا نحتاج الى كود نصي يقوم بهذه المهمة

```
for population_data in data:
    print(population_data)
```

بهذا نكون قد قمنا بطباعة بيانات السكان الموجودة في صفحة الويب داخل المحرر بدون الحفظ في pc الخاص بنا ، مع ملاحظة ان هذه البيانات تحتاج الى تنظيف . فالبيانات التي في صفحة الويب تتكون من 5 أعمدة

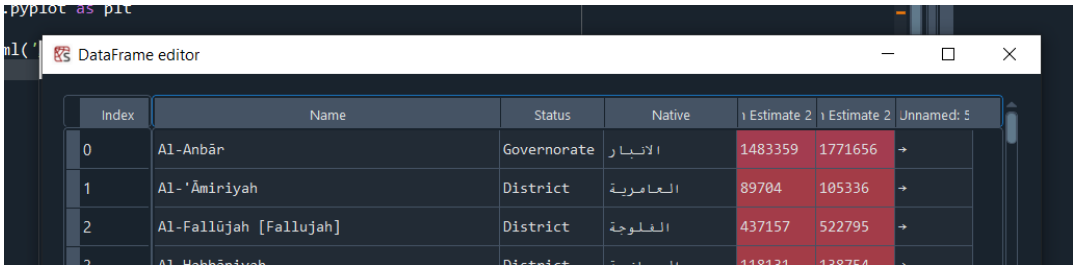
شكل 2 (البيانات كما هي في صفحة الويب)

Name	Status	Native	Population Estimate 2009-10-17	Population Estimate 2018-07-01
Al-Anbār	**Governorate**	**الانبار**	**1,483,359**	**1,771,656**
Al-'Āmiriyah	District	العامرية	89,704	105,336
Al-Fallūjah [Fallujah]	District	الفلوجة	437,157	522,795
Al-Ḥabbāniyah	District	الحبانية	118,131	138,754
Al-Qā'im	District	القائم	150,146	179,192
'Anah	District	عنه	26,162	31,575
Ar-Ramādī [Ramadi]	District	الرمادي	373,569	446,519
Ar-Ruṭbah	District	الرطبة	39,000	47,040
Ḥadīṯha [Haditha]	District	حديثة	87,219	105,710
Hīt	District	هيت	142,535	170,776
Rāwah	District	راوه	19,736	23,959
Al-Baṣrah [Basra]	**Governorate**	**البصرة**	**2,405,434**	**2,908,491**
Abū al-Khaṣīb	District	أبو الخصيب	183,285	223,675
Al-Baṣrah [Basra]	District	البصرة	1,175,868	1,436,253
Al-Fāw	District	الفاو	34,840	42,252
Al-Midaīnah	District	المدينة	199,891	234,641
Al-Qurnah	District	القرنة	241,490	286,073

المصدر https://citypopulation.de/en/iraq/admin/

لكن عندما استدعيناها في محرر بايثون (Spyder) تم اضافة عمود سادس في الاخير وهو فارغ فضلا عن وجود تسميات محلية غير مرغوب بها لأن اللغة الافضل في بايثون هي اللغة الانكليزية، ذلك ان انظمة التشغيل هي باللغة الانكليزية ، وفي ذات السياق يفضل ان توضع ملفات العمل في امتدادات لا تحتوي مسمياتها على اي حرف من غير احرف اللغة الانكليزية.

شكل3 (البيانات كما هي في محرر Spyder)



هذا من جانب ، ومن جانب اخر فإنني اريد حفظ هذه البيانات على pc لغرض تمثيلها بخرائط الكثافة ولا يمكنني ان ارجع في كل مرة لموقع الويب من اجل التعامل مع هذه البيانات ، وبذلك نقوم بوضع مسار لحفظ ملف الاكسل الذي سنخزن البيانات فيه مع وضع اسم الملف وامتداده وهو هنا pop.xlsx، مع ملاحظة انه من الضروري ان نقوم بوضع الحرف r كي يفهم ان التالي هو مسار وليس جزءً من الكود.

```
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
data = pd.read_html('https://citypopulation.de/en/iraq/admin/')
for population_data in data:
    print(population_data)
population_data.to_excel(r'C:\Users\Dr. Zinab\Desktop\test\pop.xlsx')
```

وبمجرد عمل run سنلاحظ انه تم حفظ البيانات في ملف اكسل بالامتداد الذي تم اعطاؤه في الكود.

الان لم نعد بحاجة الى الى قراءة البيانات عن طريق امتداد الويب وسنعمل على الفتح من pc عليه سنقوم بعمل # للاسطر السابقة ليتم التعامل معها كتعليق وليس كجزء من الكود (طبعا عدا الاسطر الثلاثة الاولى الخاصة باستيراد المكتبات)

```
population_data=pd.read_excel(r'C:\Users\Dr.Zinab\Desktop\PopulationDensity\PopulationDensity\pop.xlsx')
```

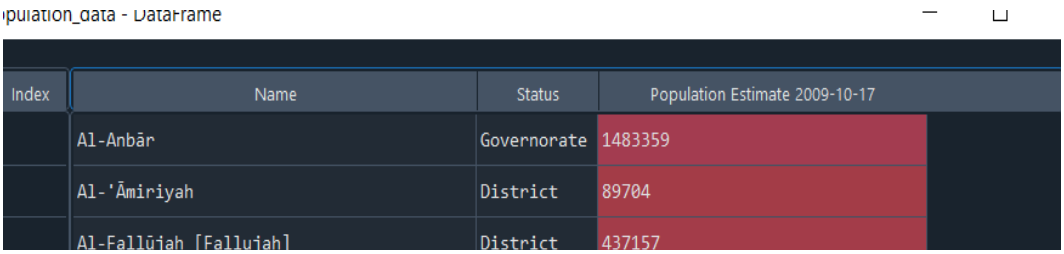
الان اريد عرض الاعمدة الثلاثة التي تعنيني الان في عرض الكثافة السكانية لعام 2009 وهي :-

'Name', 'Status', 'Population Estimate 2009-10-17'

وسيكون ذلك عن طريق الكود

```
population_data=population_data[['Name', 'Status', 'Population Estimate 2009-10-17']]
```

والنتيجة اننا اظهرنا فقط الاعمدة الثلاثة المطلوبة



نلاحظ ان الحقل الثالث الخاص بسنة التعداد هو حقل طويل جدا وكي نختصره يجب ان نقوم بأعاد تسميته من خلال الكود :-

```
rename(columns={'Population Estimate 2009-10-17': 'Population'}, inplace=True)
```

وبذلك استبدلت التسمية من 'Population Estimate 2009-10-17' الى Population

من حقل status هنالك بعض القيم التي تعطي مجموع المحافظة بالكامل فضلا عما تعطيه الاقضية ، فإذا ما اردنا ان نحتسب الكثافة فمن غير الممكن ان نحسب الكثافة للمحافظة والقضاء في وقت واحد ، عليه لابد من اعطاء ايعاز بأن المطلوب هو حساب القيم District فقط وتسميتها ب status وسيكون ذلك من الكود التالي:-

```
population_data=population_data.loc[population_data['Status'] == 'District']
```

الان سنقوم بانشاء عمود فارغ بأسم (Districts2) في الجدول وذلك لغرض وضع تسمية للوحدات الادارية (الاقضية بعد استبعاد التسمية المحلية ، فبعض الاقضية وضعت لها تسميتين كما في الفلوجة مثلا:-

Al-Fallūjah [Fallujah]

هذا العمود الفارغ سيتم عن طريق السطر البرمجي التالي:-

```
population_data['Districts2'] = ' '
```

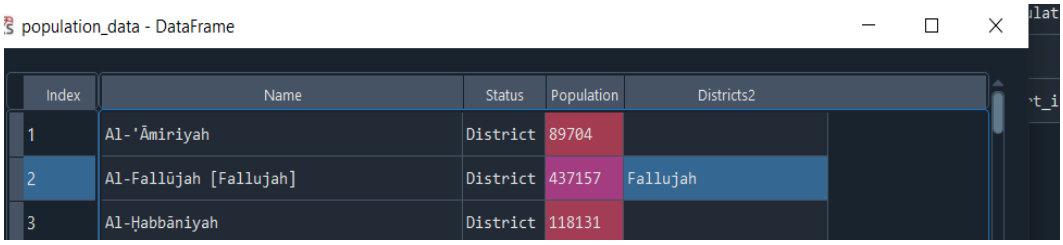
وهذا الامر مهم اثناء دمج بيانات السكان مع البيانات المكانية التي سنحصل عليها من موقع DIVA GIS والتي يجب ان تكون بذات التنسيق. ففي مثالنا عن الفلوجة سنقوم بنسخ الاسم الذي بين القوسين فقط ووضعه في العمود الجديد مع استبعاد التسمية المحلية والتي تعبر عن نطق ابناء البلد لهذه التسمية، والاقضية التي لها نفس التسمية ستنقل هذه التسمية الى الحقل الجديد.

ليس من المعقول ان اقوم بالوقوف على كل وحدة ادارية لها تسميتين (محلية و عالمية) لأقوم بنسخ العالمية ووضعها في الحقل الجديد، فهذا الامر يكون غير عملي خصوصا اذا ما كانت هنالك وحدات ادارية كثيرة، هذا من جانب من جانب اخر فأن أحد اهداف البرمجة بلغة بايثون هو اختصار الجهد والوقت ، هذه العملية ستتم من خلال الحلقات التكرارية في بايثون for loop ، فالغاية البرمجة هي اتمتة المهام واجراء المهام التكرارية لكيلا نحتاج الى القيام بها يدويا ، واحدى طرائق تكرار المهام المتشابهة هي استخدام حلقات التكرار Loop[7] ، فيبحث عن كل تسمية موجودة داخل قوسين [ ] ويضعها في العمود الجديد ، عن طريق الكود التالي:-

```
for index, row in population_data.iterrows():
if '[' and ']' in row['Name']:
start_index = row['Name'].find('[')
end_index = row['Name'].find(']')
```

```
population_data.loc[index, 'Districts2'] = population_data.loc[index]['Name'][start_index+1 : end_index]
```

نرجع لمثال الفلوجة لنلاحظ كيف اصبحت التسمية في العمود الجديد. وهذا الامر انطبق على كل الحالات المشابهة



اما الاقضية التي لها تسمية واحدة فبقيت الخلايا التي امامها فارغة في العمود الجديد وبالتالي فإنه مطلوب منا ان نعطي الامر بنقل تسميتها الوحيدة الى العمود الجديد وسيكون ذلك بغلق حلقة for loop عن طريق الامر else

```
else:
population_data.loc[index, 'Districts2'] = population_data.loc[index]['Name']
```

بعد عمل run سنلاحظ ان الحقل الجديد امتلأ بالكامل بتسميات الاقضية المنسقة وفق المطلوب

الان نعمل مرة اخرى على تنظيف البيانات ذلك انه لدينا اعمدة لا نحتاج اليها وهي (Name , Status) فلابد من حذفها وهذا سيكون من خلال:-

```
population_data = population_data[['Population', 'Districts2']]
```

وبما اننا حذفنا عمود Districts الاول (ولم يعد موجوداً) وابقينا على عمود Districts2 فسنغير تسمية العمود الاخير ليأخذ تسمية العمود الاول (Districts) وذلك عن طريق:-

```
population_data.rename(columns={'Districts2': 'District'}, inplace=True)
```

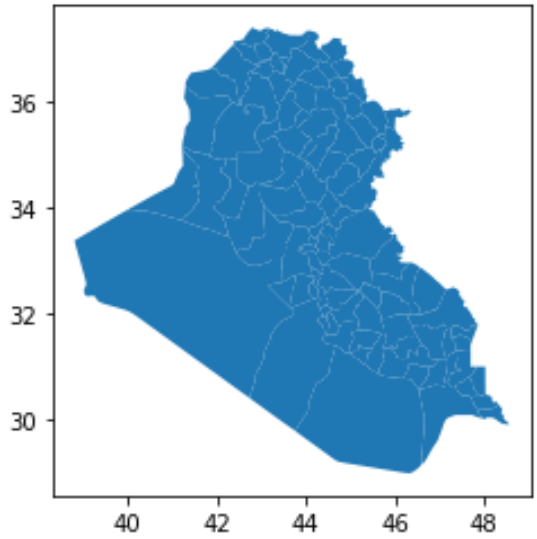
الجزء الاخر من البيانات مفتوحة المصدر هو تحميل Shapefile الخاص بالعراق وبالتقسيم الاداري نفسه (الاقضية) وذلك من موقع DIVA-GIS

بعد تحميل هذه البيانات على PC سنقوم باستيرادها الى واجهة بايثون باستخدام مكتبة Geopandas ، اي اننا سنقوم بأعطاء امر لقراءة بيانات Shapefile في محرر Spyder عبر السطر البرمجي التالي:-

```
irq_districts=gpd.read_file(r'C:\Users\Dr.Zinab\Desktop\PopulationDensity\PopulationDensity\IRQ_adm\IRQ_adm2.shp')
```

نستطيع من نافذة console ضمن محرر Spyder ان نعرض نوع الملف وعرض اول خمس صفوف ومن ثم انشاء الرسم البياني (plot) الذي يمثل هنا (العراق بأقضيته) وذلك من خلال الاسطر التالية:-

```
In[2]: type(irq_districts)
Out[2]: geopandas.geodataframe.GeoDataFrame
irq_districts.head()
Out[3]:
   ID_0 ISO  ... VARNAME_2                                   geometry
0   108  IRQ  ...       NaN  POLYGON ((44.28727 33.18921, 0
                                                44.28334 33.18531...
1   108  IRQ  ...       NaN  POLYGON ((44.10008 32.87399, 1
                                                44.10274 32.87026...
2   108  IRQ  ...       NaN  POLYGON ((42.12704 33.82630, 2
                                                42.12239 33.82910...
3   108  IRQ  ...       NaN  POLYGON ((41.22286 35.05230, 3
                                                41.25406 35.05267...
4   108  IRQ  ...       NaN  POLYGON ((41.22286 35.05230, 4
                                                41.22281 35.05319...
[5 rows x 12 columns]
()In[4]: irq_districts.plot
```



التحكم في حجم الشكل(الخريطة) من خلال نافذة cosole وذلك عبر السطر التالي:-

irq_districts.plot(figsize=(9,9))

كما ويمكن تغيير الارقام بدلا من 9 ان نجعلها اقل او اكثر ، هذا التكبير يعطي ظهور اكثر وضوحا للوحدات الادارية الصغيرة لأنه يرفع من الدقة المكانية للوحدات لادارية.

ثانياً: انشاء خريطة الكثافة السكانية في Python

ان جدول السمات (attribute table) فيه الكثير من الاعمدة التي لا نحتاجها ، فبالتالي سنعمد الى تضمين هذا الجدول للاعمدة التي نحتاجها فقط مع الاستغناء عن باقي الاعمدة وسيكون ذلك من خلا لاسطر البرمجي التالي:-

```
irq_districts = irq_districts[['NAME_2', 'geometry']]
```

والذي يجعل اعمدة الجدول كالتالي:-

Index	NAME_2	geometry
0	Abu Ghraib	POLYGON ((44.287273406982536 33.18921279907226…

الان لعمل خريطة للكثافة لابد من توحيد عمود الوحدات الادارية (الاقضية) وهو في بيانات اعداد السكان بأسم District وفي طبقة الخريطة بأسم 'NAME_2 عليه سنغير في الاخيرة عن طريق السطر البرمجي التالي:-

```
irq_districts.rename(columns={'NAME_2':  'District'},
                                    inplace=True)
```

من نافذة console اذا ما قمنا بكتابة الكود irq_districts.crs فأنه سيعطيني النظام الاحداثي للخريطة والنتيجة ستكون:

< Geographic 2D CRS: EPSG:3857>

Name: WGS 84

المشكلة هنا ان هذا يمثل نظام احداثيات جغرافي corrodent system وكي اقوم بعمل خريطة كثافة يجب ان تكون لديَّ مساحات الاقضية (Districts) بالامتار ، عن طريق السطر البرمجي التالي:-

```
irq_districts.to_crs(epsg=3857 , inplace = True)
```

بهذا نكون قمنا بحساب مساحة الاقضية بالامتار والنتيجة ستظهر في جدول السمات في عمود geometry

مسألة مهمة هنا هي ان عدد الصفوف في ملف اعداد السكان(149) يجب ان يكون متطابق مع عدد صفوف ملف الاقضية Districts (102) وبغض النظر عن ايهما الاصح ، فنحن هنا يجب ان نوحد هذه الصفوف لا بد من ان نعمل مطابقة هذه المطابقة ستكون عن طريق الحلقات التكرارية في بايثون for loop بحيث سيعتمد على بيانات Districts سيأخذ كل وحدة ادارية منها ويبحث عن عدد السكان الذي يقابلها في في ملف اعداد السكان ثم ينتقل الى القضاء الثاني ويعمل الشيء نفسه ويستبعد اعداد السكان الذين لا يقابلهم ملف مكاني وسيكون ذلك عن طريق السطر البرمجي التالي:-

```
for index, row in irq_districts['District'].items():
    if row in population_data['District'].tolist():
        pass
    else:
        print('the dustricts',row,'is not in the population_data list')
```

لكن هنا (في نافذة console) ممكن ان تظهر لنا حتى الوحدات موجودة في الملفين لكن باختلافات املائية ، فمثلاً قضاء الشيخان في طبقة GIS يكتب Al Shikhan لكن في طبقة بيانات السكان يكتب Shikhan اي بدون فواصل وبالتالي اعتبرهما كلمتان مختلفتان

هنا سنعطي امر باستبدال بيانات population_data التي اخذناها من موقع الويب بالبيانات التي تم اخذها من موقع DIVA GIS وسنعمل اولا على تطبيق ذلك على قضاء Al Shikhan بالسطر التالي:

```
population_data.replace("Al Shikhan","Shekhan", inplace = True)
```

ويطبق الامر نفسه على باقي الاقضية

الان بالعودة الى تشغيل الحلقات التكرارية في بايثون for loop عن طريق :-

```
for index, row in irq_districts['District'].items():
    if row in population_data['District'].tolist():
        pass
    else:
        print('the districts',row,'is NOT in the population_data list')
```

بتشغيل الاسطر البرمجية في حلقة for loop نلاحظ انه لا يظهر لنا اي رد مما يعني انه تمت عملية الاستبدال وتمت المطابقة بالكامل.

المطلوب الان هو القيام بانشاء عمود في ملف GeoDataFrame لحساب المساحة بالكيلومترات المربعة تحت مسمى area عن طريق السطر التالي:-

```
irq_districts['area'] = irq_districts.area/1000000
```

وسيضاف عمود جديد في ملف area بالمسمى المطلوب وفيه المساحة بالكيلومترات المربعة

الخطوة التالية هي عملية الدمج (merge) بين بيانات الملفين لعمل عمود جديد في ملف GeoDataFrame تحت مسمى population بحيث كل وحدة ادارية يقابلها geometry الخاص بها والمساحة وعدد السكان (وهنا سيتم استخدام مكتبة Pandas عن طريق:-

```
irq_districts = irq_districts.merge(population_data, on = 'District')
```

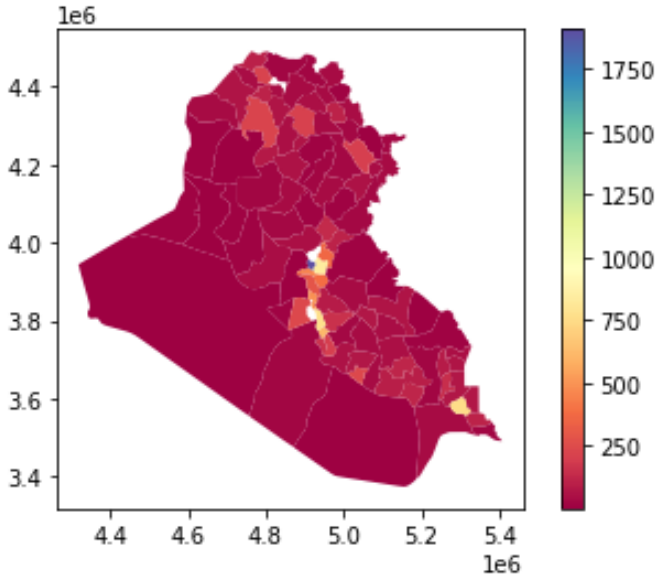
اصبحت الان مسألة انشاء عمود خاص بالكثافة السكانية اسهل ع طريق قسمة عمود population على عمود area من خلال السطر البرمجي:-

```
irq_districts['pop_den (people/sq. km)'] = irq_districts['Population']/irq_districts['area']
```

هنا تم احتساب الكثافة السكانية عن طريق عمود جديد تحت مسمى ('pop_den (people/sq. km')

الان نأتي الى المسألة الاهم كوننا جغرافيون وهي تمثيل هذه الكثافة على خرائط سكانية وذلك باستخدام مكتبة plotting

```
# Plotting
irq_districts.plot(column = 'pop_den (people/sq. km)', cmap = 'Spectral', legend = True)
plt.savefig('population_density_iraq2009.jpg')
```



لحفظ الخريطة في ملف Excel

```
Save the population me to an Excel file #
irq_districts.to_excel('population2018.xlsx', index=False)
```

ولطباعة الخريطة

```
Print a message to confirm that the Excel file was created #
print('sample_data.xlsx file has been created successfully!')
```

الاستنتاجات

1 . تسهم البرمجة بلغة بايثون من الاستفادة من البيانات مفتوحة المصدر

2 . على الباحث الذي يعمل مع البيانات مفتوحة المصدر ان يجد الطريقة الملائمة لتنظيف البيانات الخاصة ببحثه

3. تسهم البرمجة بلغة البايثون من هذه البيانات بعد ايجاد طريقة لتحميلها اولا ومن ثم تنظيفها وحفظها بشكل ملفات اكسل ومن ثم تمثيلها بخرائط كثافة سكانية وحفظها بهيئة صور

4 . اضافة اوامر وميزات وادوات غير موجودة في برامج نظم المعلومات الجغرافية لا يمكن ان يتم بمعزى عن معرفة واضحة بالبرمجة بلغات ابرزها لغة بايثون.

المصادر:-

1. https://www.tableau.com/learn/articles/what-is-data-cleaning#definition

2. جمعة محمد داود ، انواع البيانات في نظم المعلومات الجغرافية، دورة نظرية على الانترنت ، الحلقة الثالثة، 2019، الموقع على الانترنت:- https://www.youtube.com/playlist?list=PL-2sBQtgS7Y7rnp3fNM2ndmyNEJxpyj-B

3. http://www.diva-gis.org/gdata

4. رشا صابر نوفل، أسس البرمجة المكانية بلغتي Python و R في نظم المعلومات الجغرافية ، 2021 . ص73.

5. Silas Toms, ArcPy and ArcGIS – Geospatial Analysis with Python, Packt Publishing Ltd, Birmingham B3 2PB, UK.2015,P22.

6. Kelsey Jordahl, GeoPandas Documentation Release 0.2.0.dev, May 17, 2016,p1.

7. ليزا تاغليفيري، البرمجة بلغة بايثون ، ترجمة محمد بغات وعبد اللطيف ايمش، اكاديمية حسوب،النسخة الاولى ، 2020.ص263.

8. https://www.citypopulation.de/en/iraq