Merging JsonStat and GeoJson formatted data to create a GeoDataFrame, it's visualisation, and writing it to an Esri Shapefile

An example from Ireland using Python

May 2019

Kevin McCormack

Dr. Mary Smyth

Sinead Phelan

Contents **Page**

1. Introduction

In this tutorial we will discuss how to extract and join statistical data from the CSO's online database, StatBank, with geographic data, namely polygons, extracted from the OSi open data portal to create a GeoDataFrame, it's visualisation, and write it to an Esri Shapefile

2. Data Repository - Statbank

StatBank is the CSO's online database of Official Statistics . This database contains current and historical data series compiled from CSO statistical releases and is accessed at http://www.cso.ie/px/pxeirestat/statire/SelectTable/Omrade0.asp?Planguage=0

3. Geographic data repository – Open Data Portal

The OSi provide Open Geographic Data is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement that the source of the information is attributed to and accessed at https://data-osi.opendata.arcgis.com/.

4. JSON-stat format

The **JSON-stat format** is a simple lightweight JSON[1] format for data dissemination. It is based in a cube model that arises from the evidence that the most common form of data dissemination is the tabular form. In this cube model, **datasets** are organized in **dimensions**. Dimensions are organized in **categories**.

Data dissemination is not the business of a few anymore. Even though the **JSON-stat format** can be the perfect companion for the **open data initiatives** of National Statistical Offices, it is suitable for all kinds of data disseminators because it has been designed with simplicity in mind.

[1] **JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

5. GeoJSON

GeoJSON is a format for encoding a variety of geographic data structures. GeoJSON supports the following geometry types:

- Point,
- LineString,
- Polygon,
- MultiPoint,
- MultiLineString, and
- MultiPolygon.

Geometric objects with additional properties are Feature objects. Sets of features are contained by FeatureCollectionobjects.

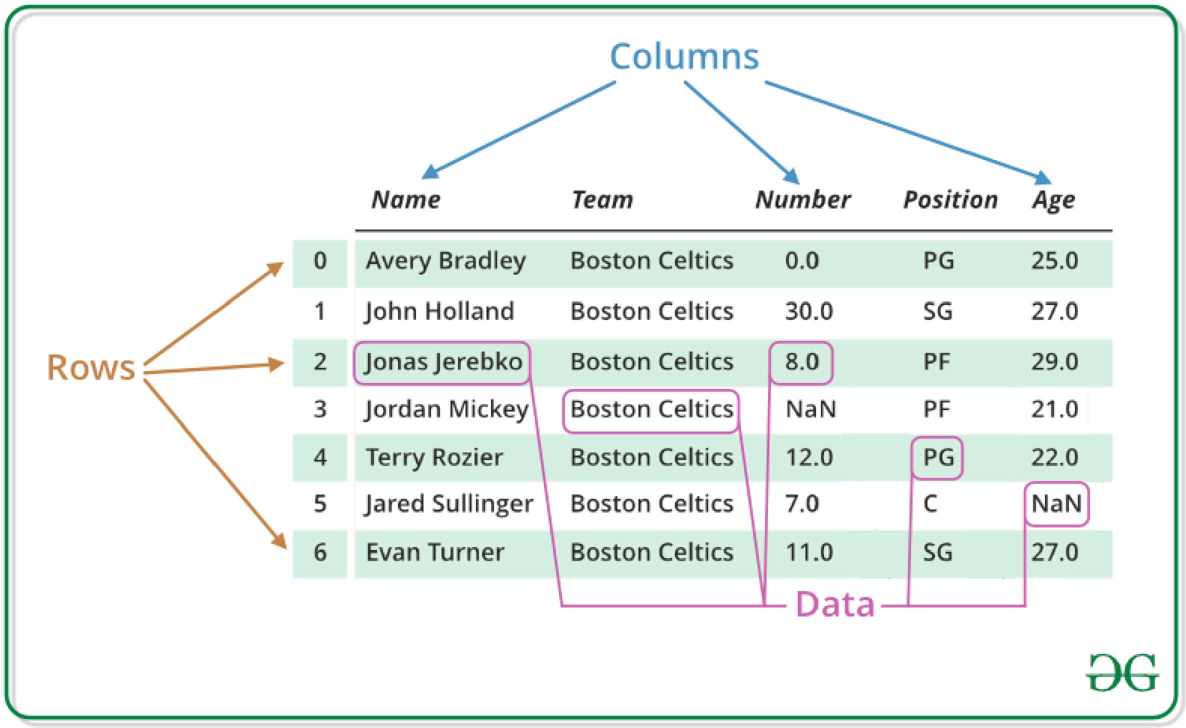
6. Python

Python is an interpreted, high-level, general-purpose programming language.

Python can be downloaded and installed from either
https://www.python.org/downloads/, or
https://www.anaconda.com/

6.1 Pandas DataFrame

A Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labelled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas DataFrame consists of three principal components, the **data**, **rows**, and **columns**.


Columns
Rows
Data

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Figure 1: Pandas DataFrame

6.2 GeoDataFrame

A GeoDataFrame object is a pandas.DataFrame that has a column with geometry.

6.3 Python libraries

Within Python there are numerous libraries which are collections of functions and methods that allows one to perform many actions without writing your own code.

In this tutorial several libraries are installed, imported and used.

7. Python code

There are several steps in the process.

7.1 Installing the libraries

In this tutorial the following libraries are installed.

```
!pip install jsonstat.py
!pip install pyjstat
```

It should be noted that when installing the geopandas library one has to install several dependency libraries. A number of these libraries they may not install directly and one may have to go to this website and download the *.whl file and manually load using Anaconda prompt. https://www.lfd.uci.edu/~gohlke/pythonlibs/#shapely

```
!pip install numpy
!pip install GDAL              # had to be manually loaded
!pip install pyshp
!pip install Shapley           # had to be manually loaded
!pip install Fiona             # had to be manually loaded
!pip install geopy
!pip install pyproj
!pip install geopandas         # had to be manually loaded
!pip install descartes
```

If one wishes to visualise a GeoDataFrame then install these libraries.

```
!pip install matplotlib
!pip install mplleaflet
```

7.2 Importing and initialising the main python libraries

To activate the libraries, one must import and initialise them.

```
import geopandas as gpd
import pandas as pd             # using panda to convert jsonstat data set to pandas dataframe
import os                       # The functions that the OS module provides allows you to
                                interface with the underlying operating system that Python is
                                running on – e.g. Windows, etc.

from pyjstat import pyjstat
import numpy as np
import shapefile as shp
import jsonstat                 # import jsonstat.py package
import matplotlib.pyplot as plt # for plotting
import requests
import seaborn as sns
```

8. Setting current directory and downloading JsonStat file from the CSO's Statbank API

The following code allows one to identify and set the current working directory

```
os.getcwd()                  # current working directory 'C:\\XXXX'
os.chdir('C:\\XXXX\\Python') # changing current directory
```

We will now download the JsonStat QLF07 file, also known as a collection, from CSO Statbank's API, which contains data for Persons aged 15 years and over in Employment by Sex, NACE Rev 2 Economic Sector, Region and Quarter obtained from the CSO's Quarterly Labour Force Survey

First, we will define a url and file_name as follows

```
url ='https://www.cso.ie/StatbankServices/StatbankServices.svc/jsonservice/responseinstance/QLF07'
file_name = "QLF07.json"
```

9. Extracting the JsonStatCollection

We will extract the JsonStatCollection and create a directory for data

```
jsonstat.from_url(url, file_name)
file_path = (os.path.join('C:\\XXXX', file_name))
```

10. Initialising the JsonStatCollection from the file

We will now Initialise JsonStatCollection from the file and print the list of dataset contained into the collection.

```
collection = jsonstat.from_file(file_path)
```

To identify the number of datasets contained in this collection

```
len(collection)
```

The value '1' is returned indicating that there is one dataset contained in this collection and this can be confirmed by running the command,

```
collection
```

which provides this output,

```
JsonstatCollection contains the following JsonStatDataSet:

+-----+-----------+
| pos | dataset   |
+-----+-----------+
| 0   | 'dataset' |
+-----+-----------+
```

11. Selecting and extract dataset contained into collection

We select, and extract the dataset contained in the collection and name it QLF07 as follows:

```
QLF07 = collection.dataset(0)
```

To understand what is contained in the dataset simply run the code,

```
QLF07
```

Which provides the following output.

```
name:    'dataset'
label:   'Persons aged 15 years and over in Employment  by  Sex, NACE Rev 2 Economic
         Sector,   Region and Quarter'
source:  'Persons aged 15 years and over in Employment  by  Sex, NACE Rev 2 Economic
         Sector,  Region and Quarter'
size:    13122

+-----+----------------------------------------+----------------------------------------+-------+--------+
| pos | id                                     | label                                  | size  | role   |
+-----+----------------------------------------+----------------------------------------+-------+--------+
| 0   | Sex                                    | Sex                                    |   3   |        |
| 1   | NACE Rev 2 Economic Sector             | NACE Rev 2 Economic Sector             |  18   |        |
| 2   | Region                                 | Region                                 |   9   |        |
| 3   | Quarter                                | Quarter                                |  27   | time   |
| 4   | Statistic                              | Statistic                              |   1   | metric |
+-----+----------------------------------------+----------------------------------------+-------+--------+
```

12. Exploring the dimensions of the JsonStatDataSet:

One can obtain some details about the dimensions, i.e. categories within a dimension as follows.

```
QLF07.dimension('dataset')
```

Which provides the following output with the name of the dimensions identified.

```
JsonStatException: "dataset 'dataset': unknown dimension 'dataset' know dimensions
                            ids are:
Sex, NACE Rev 2 Economic Sector, Region, Quarter, Statistic"
```

To investigate the dimension region, for example, use the following command.

```
QLF07.dimension('Region')
```

Which returns the following.

```
+-----+---------+-----------------+
| pos | idx     | label           |
+-----+---------+-----------------+
| 0   | '-'     | 'State'         |
| 1   | 'IE041' | 'Border'        |
| 2   | 'IE042' | 'West'          |
| 3   | 'IE051' | 'Mid-West'      |
| 4   | 'IE052' | 'South-East'    |
| 5   | 'IE053' | 'South-West'    |
| 6   | 'IE061' | 'Dublin'        |
| 7   | 'IE062' | 'Mid-East'      |
| 8   | 'IE063' | 'Midland'       |
+-----+---------+-----------------+
```

The following code provides details on the Nace Rev 2 Economic Sector dimension.

```
QLF07.dimension('NACE Rev 2 Economic Sector')
```

It can also be referenced as dimension 1, as follow:

```
QLF07.dimension(1)
```

Which returns the following:

```
# dimension 1
```

```
+-------------------------------------------------------------------------------------------------------+
| pos | idx      | label                                                                              |
+-----+----------+------------------------------------------------------------------------------------+
| 0   | '-'      | 'All NACE economic sectors'                                                        |
| 1   | 'A'      | 'Agriculture, forestry and fishing (A)'                                            |
| 2   | 'F'      | 'Construction (F)'                                                                 |
| 3   | 'G'      | 'Wholesale and retail trade, repair of motor vehicles and motorcycles (G)' |
| 4   | 'H'      | 'Transportation and storage (H)'                                                   |
| 5   | 'I'      | 'Accommodation and food service activities (I)'                                    |
| 6   | 'J'      | 'Information and communication (J)'                                                |
| 7   | 'M'      | 'Professional, scientific and technical activities (M)'                            |
| 8   | 'N'      | 'Administrative and support service activities (N)'                               |
| 9   | 'O'      | 'Public administration and defence, compulsory social security (O)'                |
| 10  | 'P'      | 'Education (P)'                                                                    |
| 11  | 'Q'      | 'Human health and social work activities (Q)'                                      |
| 12  | 'Y0900'  | 'Industry (B to E)'                                                                |
| 13  | 'Y1000'  | 'Industry and Construction (B to F)'                                               |
| 14  | 'Y3000'  | 'Services (G to U)'                                                                |
| 15  | 'Y3500'  | 'Financial, insurance and real estate activities (K,L)'                            |
| 16  | 'Y7500'  | 'Other NACE activities (R to U)'                                                   |
| 17  | 'ZXD400' | 'Not stated'                                                                       |
```

We continue checking the other dimensions, with

QLF07.dimension('Sex')

returning:

```
+-----+-----+----------------+
| pos | idx | label          |
+-----+-----+----------------+
| 0   | '-' | 'Both sexes' |
| 1   | '1' | 'Male'         |
| 2   | '2' | 'Female'       |
+-----+-----+----------------+
```

and

```
QLF07.dimension('Quarter')
```

returning:

```
#+-----+-------------+------------+  +-----+-------------+------------+
#| pos | idx          | label      |  | pos | idx          | label      |
#+-----+-------------+------------+  +-----+-------------+------------+
#| 0    | '2012Q1' | '2012Q1' |  | 14   | '2015Q3' | '2015Q3' |
#| 1    | '2012Q2' | '2012Q2' |  | 15   | '2015Q4' | '2015Q4' |
#| 2    | '2012Q3' | '2012Q3' |  | 16   | '2016Q1' | '2016Q1' |
#| 3    | '2012Q4' | '2012Q4' |  | 17   | '2016Q2' | '2016Q2' |
#| 4    | '2013Q1' | '2013Q1' |  | 18   | '2016Q3' | '2016Q3' |
#| 5    | '2013Q2' | '2013Q2' |  | 19   | '2016Q4' | '2016Q4' |
#| 6    | '2013Q3' | '2013Q3' |  | 20   | '2017Q1' | '2017Q1' |
#| 7    | '2013Q4' | '2013Q4' |  | 21   | '2017Q2' | '2017Q2' |
#| 8    | '2014Q1' | '2014Q1' |  | 22   | '2017Q3' | '2017Q3' |
#| 9    | '2014Q2' | '2014Q2' |  | 23   | '2017Q4' | '2017Q4' |
#| 10   | '2014Q3' | '2014Q3' |  | 24   | '2018Q1' | '2018Q1' |
#| 11   | '2014Q4' | '2014Q4' |  | 25   | '2018Q2' | '2018Q2' |
#| 12   | '2015Q1' | '2015Q1' |  | 26   | '2018Q3' | '2018Q3' |
#| 13   | '2015Q2' | '2015Q2' |
#+-----+-------------+----------+  +-------+------------+-------------+
```

And for the Statistic dimension, number 4,

```
QLF07.dimension(4)
```

returns

```
+-----+--------------+---------------------------------------------------------------------------------+
| pos | idx          | label                                                                           |
+-----+--------------+---------------------------------------------------------------------------------+
| 0   | 'QLF07C01' | 'Persons aged 15 years and over in Employment (Thousand)' |
```


13. Accessing a value in the JsonStatDataSet, "dataset"

To access a value in the dataset one references the dimension values such as:

```
QLF07.data(Region = 'IE041', Sex ='1', Quarter = '2018Q3' )
```

Which returns

```
JsonStatValue(idx=4427, value=97.5, status=None)
```

It should be noted that the dimension name NACE Rev 2 Economic Sector cannot be used as it contains spaces, simply use the dimension number instead.

14. Transforming the JsonStatDataSet, "dataset", into a pandas DataFrame

The following code transforms the JsonStatDataSet, "dataset", into a pandas DataFrame

```
df_QLF07 = QLF07.to_data_frame(content='idx')
```

Note that the *content* = *'idx'* is used to grab the geography category values, which we will call GEOGID, from the regional dimension

It is also possible to grab the label, by replacing content='idx' with content = 'label'

```
df_QLF07A = QLF07.to_data_frame(content='label')
```

Again, one can display the dataframe:

```
display(df_QLF07)
```

Which provides the following output:

```
   Sex NACE Rev 2 Economic Sector Region   Quarter  Statistic    Value
0    -                 -              -    2012Q1  QLF07C01  1863.2
1    -                 -              -    2012Q2  QLF07C01  1878.0
2    -                 -              -    2012Q3  QLF07C01  1887.0
3    -                 -              -    2012Q4  QLF07C01  1893.6
4    -                 -              -    2013Q1  QLF07C01  1892.0
5    -                 -              -    2013Q2  QLF07C01  1926.3
6    -                 -              -    2013Q3  QLF07C01  1961.8
7    -                 -              -    2013Q4  QLF07C01  1971.0
8    -                 -              -    2014Q1  QLF07C01  1950.7
9    -                 -              -    2014Q2  QLF07C01  1970.3
10   -                 -              -    2014Q3  QLF07C01  2008.9
11   -                 -              -    2014Q4  QLF07C01  2025.2
12   -                 -              -    2015Q1  QLF07C01  2014.4
13   -                 -              -    2015Q2  QLF07C01  2049.7
14   -                 -              -    2015Q3  QLF07C01  2079.9
15   -                 -              -    2015Q4  QLF07C01  2085.4
16   -                 -              -    2016Q1  QLF07C01  2080.8
17   -                 -              -    2016Q2  QLF07C01  2126.7
18   -                 -              -    2016Q3  QLF07C01  2158.0
19   -                 -              -    2016Q4  QLF07C01  2163.5
20   -                 -              -    2017Q1  QLF07C01  2158.4
21   -                 -              -    2017Q2  QLF07C01  2180.9
22   -                 -              -    2017Q3  QLF07C01  2206.5
23   -                 -              -    2017Q4  QLF07C01  2230.8
24   -                 -              -    2018Q1  QLF07C01  2220.7
25   -                 -              -    2018Q2  QLF07C01  2255.0
26   -                 -              -    2018Q3  QLF07C01  2273.2
27   -                 -          IE041    2012Q1  QLF07C01   145.1
28   -                 -          IE041    2012Q2  QLF07C01   146.3
29   -                 -          IE041    2012Q3  QLF07C01   145.3
```

One can display the dataframe with the labels.

```
display(df_QLF07A)
```

```
#        Sex     NACE Rev 2 Economic Sector  Region  Quarter                   Statistic                        Value
#0   Both sexes  All NACE economic sectors   State  2012Q1  Persons aged 15 years and over in Employment (...  1863.2
#1   Both sexes  All NACE economic sectors   State  2012Q2  Persons aged 15 years and over in Employment (...  1878.0
#2   Both sexes  All NACE economic sectors   State  2012Q3  Persons aged 15 years and over in Employment (...  1887.0
#3   Both sexes  All NACE economic sectors   State  2012Q4  Persons aged 15 years and over in Employment (...  1893.6
#4   Both sexes  All NACE economic sectors   State  2013Q1  Persons aged 15 years and over in Employment (...  1892.0
#5   Both sexes  All NACE economic sectors   State  2013Q2  Persons aged 15 years and over in Employment (...  1926.3
#6   Both sexes  All NACE economic sectors   State  2013Q3  Persons aged 15 years and over in Employment (...  1961.8
#7   Both sexes  All NACE economic sectors   State  2013Q4  Persons aged 15 years and over in Employment (...  1971.0
#8   Both sexes  All NACE economic sectors   State  2014Q1  Persons aged 15 years and over in Employment (...  1950.7
#9   Both sexes  All NACE economic sectors   State  2014Q2  Persons aged 15 years and over in Employment (...  1970.3
#10  Both sexes  All NACE economic sectors   State  2014Q3  Persons aged 15 years and over in Employment (...  2008.9
#11  Both sexes  All NACE economic sectors   State  2014Q4  Persons aged 15 years and over in Employment (...  2025.2
#12  Both sexes  All NACE economic sectors   State  2015Q1  Persons aged 15 years and over in Employment (...  2014.4
#13  Both sexes  All NACE economic sectors   State  2015Q2  Persons aged 15 years and over in Employment (...  2049.7
#14  Both sexes  All NACE economic sectors   State  2015Q3  Persons aged 15 years and over in Employment (...  2079.9
#15  Both sexes  All NACE economic sectors   State  2015Q4  Persons aged 15 years and over in Employment (...  2085.4
#16  Both sexes  All NACE economic sectors   State  2016Q1  Persons aged 15 years and over in Employment (...  2080.8
#17  Both sexes  All NACE economic sectors   State  2016Q2  Persons aged 15 years and over in Employment (...  2126.7
#18  Both sexes  All NACE economic sectors   State  2016Q3  Persons aged 15 years and over in Employment (...  2158.0
#19  Both sexes  All NACE economic sectors   State  2016Q4  Persons aged 15 years and over in Employment (...  2163.5
#20  Both sexes  All NACE economic sectors   State  2017Q1  Persons aged 15 years and over in Employment (...  2158.4
#21  Both sexes  All NACE economic sectors   State  2017Q2  Persons aged 15 years and over in Employment (...  2180.9
#22  Both sexes  All NACE economic sectors   State  2017Q3  Persons aged 15 years and over in Employment (...  2206.5
#23  Both sexes  All NACE economic sectors   State  2017Q4  Persons aged 15 years and over in Employment (...  2230.8
#24  Both sexes  All NACE economic sectors   State  2018Q1  Persons aged 15 years and over in Employment (...  2220.7
#25  Both sexes  All NACE economic sectors   State  2018Q2  Persons aged 15 years and over in Employment (...  2255.0
#26  Both sexes  All NACE economic sectors   State  2018Q3  Persons aged 15 years and over in Employment (...  2273.2
#27  Both sexes  All NACE economic sectors  Border  2012Q1  Persons aged 15 years and over in Employment (...   145.1
#28  Both sexes  All NACE economic sectors  Border  2012Q2  Persons aged 15 years and over in Employment (...   146.3
#29  Both sexes  All NACE economic sectors  Border  2012Q3  Persons aged 15 years and over in Employment (...   145.3
```

15. Renaming columns in a dataframe

To confirm the column list in the dataframes use the following code:

```
df_QLF07.columns
```

which returns,

```
Out[79]: Index(['Sex', 'NACE Rev 2 Economic Sector', 'Region', 'Quarter', 'Statistic',
               'Value'], dtype='object')
```

and,

```
df_QLF07A.columns
```

which returns

```
Out[80]: Index(['Sex', 'NACE Rev 2 Economic Sector', 'Region', 'Quarter', 'Statistic',
         'Value'], dtype='object')
```

We can now rename columns *Region* to *NUTS3* in df_QLF07 dataframe and to NUTS3NAME in df_QLF07A dataframe, in order to join with the GeoJson file below

```
df_QLF07.rename(columns={'Region':'NUTS3'}, inplace=True)
df_QLF07A.rename(columns={'Region':'NUTS3NAME'}, inplace=True)
```

We will now list the of columns in each dataframe to confirm the change of name.

```
df_QLF07.columns
```

returns,

```
Out[83]: Index(['Sex', 'NACE Rev 2 Economic Sector', 'NUTS3', 'Quarter', 'Statistic',
         'Value'], dtype='object')
```

and,

```
df_QLF07A.columns
```

returns,

```
Out[84]: Index(['Sex', 'NACE Rev 2 Economic Sector', 'NUTS3NAME', 'Quarter',
         'Statistic', 'Value'], dtype='object')
```

16. Creating a GeoDataFrame directly from GeoJson dataset on OSi's open portal

We will now create a dataframe directly from a GeoJson data for the NUTS3 Regions from the Ordnance Survey Ireland's (OSi) open data portal

Again we will define a url as follows:

```
url1 = "http://data-osi.opendata.arcgis.com/datasets/8e1da9ca81cb478d8146580b130abe08_2.geojson"
```

Then create the data frame.

```
dfa = gpd.read_file(url1)
```

To ensure that the a GeoDataFrame has been created, we can simply run the code:

```
type(dfa)
```

which returns,

```
geopandas.geodataframe.GeoDataFrame
```

We can list the columns in the GeoDataFrame,

```
dfa.columns
```

which returns,

```
Index(['OBJECTID', 'NUTS1', 'NUTS1NAME', 'NUTS2', 'NUTS2NAME', 'NUTS3', 'NUTS3NAME', 'GUID', 'Shape__Area', 'Shape__Length', 'geometry'], dtype='object')
```

We can explore this GeoDataFrame by referencing the columns.

```
dfa.GUID
```

returns,

```
0    B1A65D7C-1984-4A87-AD58-0E846812C992
1    A69CA800-8D87-4920-A7C1-50426A1D39B4
2    42C5C2A5-2D71-4BD1-BDB5-BD7D3198CD78
3    27C93D4E-AD0E-4B0C-8FA8-3566AEEDA5CC
4    604546A1-A856-4B9B-AD46-E88B27C27155
5    F97E459B-57ED-49C0-8A28-2BC1C7F08E88
6    8E4862CC-7E43-4BF5-A4EF-B2D5ECBA61EF
7    B26C8BAA-F3C5-49A9-B74E-D7FED1823E65
Name: GUID, dtype: object
```

With

```
dfa.NUTS3
```

returning,

```
0    IE041
1    IE042
2    IE051
3    IE052
4    IE053
5    IE061
6    IE062
7    IE063
Name: NUTS3, dtype: object
```

While,

```
dfa.NUTS3NAME
```

returns,

```
0    Border
1    West
```

```
2    Mid-West
3    South-East
4    South-West
5    Dublin
6    Mid-East
7    Midlands
Name: NUTS3NAME, dtype: object
```

17. Merging the Data and GeoData Frames

We will create a new GeoDataFrame by merging the Data and Geodata frames on the NUTS3 column.

```
dfb = dfa.merge(df_QLF07, on='NUTS3')
```

We will now list the of columns in the new GeoDataFrame to confirm the merge was successful.

```
dfb.columns
```

which returns,

```
Index(['OBJECTID', 'NUTS1', 'NUTS1NAME', 'NUTS2', 'NUTS2NAME', 'NUTS3', #
'NUTS3NAME', 'GUID', 'Shape__Area', 'Shape__Length', 'geometry', 'Sex', 'NACE
Rev 2 Economic Sector', 'Quarter', 'Statistic', 'Value'], dtype='object')
```

We can use head and tail commands to quickly view the new GeoDataFrame, ***dfb***.

The head command,

dfb.head()

will output the 1st five lines of the GeoDataFrame as follows, (not all columns are shown by default):

```
  OBJECTID NUTS1 NUTS1NAME NUTS2        NUTS2NAME  NUTS3 NUTS3NAME                                  GUID  Shape_Area  Shape_Length                                          geometry  -            Quarter  Statistic  Value
0        1    IE   Ireland  IE04  Northern and Western  IE041    Border  B1A65D7C-1984-4A87-AD58-0E846812C992    1.594889     31.557961  (POLYGON ((-7.25362176708799 55.4458025525653,...  -          -  2012Q1  QLF07C01  145.1
1        1    IE   Ireland  IE04  Northern and Western  IE041    Border  B1A65D7C-1984-4A87-AD58-0E846812C992    1.594889     31.557961  (POLYGON ((-7.25362176708799 55.4458025525653,...  -          -  2012Q2  QLF07C01  146.3
2        1    IE   Ireland  IE04  Northern and Western  IE041    Border  B1A65D7C-1984-4A87-AD58-0E846812C992    1.594889     31.557961  (POLYGON ((-7.25362176708799 55.4458025525653,...  -          -  2012Q3  QLF07C01  145.3
3        1    IE   Ireland  IE04  Northern and Western  IE041    Border  B1A65D7C-1984-4A87-AD58-0E846812C992    1.594889     31.557961  (POLYGON ((-7.25362176708799 55.4458025525653,...  -          -  2012Q4  QLF07C01  147.7
4        1    IE   Ireland  IE04  Northern and Western  IE041    Border  B1A65D7C-1984-4A87-AD58-0E846812C992    1.594889     31.557961  (POLYGON ((-7.25362176708799 55.4458025525653,...  -          -  2013Q1  QLF07C01  150.9
```

and, the tail command,

dfb.tail()

will output the last five lines of the GeoDataFrame as follows, (not all columns are shown by default):

```
      OBJECTID NUTS1 NUTS1NAME NUTS2        NUTS2NAME  NUTS3 NUTS3NAME                                  GUID  Shape_Area  Shape_Length                                          geometry  NACE Rev 2 Economic Sector  Quarter  Statistic  Value
11659        8    IE   Ireland  IE06  Eastern and Midland  IE063  Midlands  B26C8BAA-F3C5-49A9-B74E-D7FED1823E65    0.896836      7.564148  POLYGON ((-7.64673135464132 53.9415530759351, ...  2               ZXD400  2017Q3  QLF07C01    NaN
11660        8    IE   Ireland  IE06  Eastern and Midland  IE063  Midlands  B26C8BAA-F3C5-49A9-B74E-D7FED1823E65    0.896836      7.564148  POLYGON ((-7.64673135464132 53.9415530759351, ...  2               ZXD400  2017Q4  QLF07C01    NaN
11661        8    IE   Ireland  IE06  Eastern and Midland  IE063  Midlands  B26C8BAA-F3C5-49A9-B74E-D7FED1823E65    0.896836      7.564148  POLYGON ((-7.64673135464132 53.9415530759351, ...  2               ZXD400  2018Q1  QLF07C01    NaN
11662        8    IE   Ireland  IE06  Eastern and Midland  IE063  Midlands  B26C8BAA-F3C5-49A9-B74E-D7FED1823E65    0.896836      7.564148  POLYGON ((-7.64673135464132 53.9415530759351, ...  2               ZXD400  2018Q2  QLF07C01    NaN
11663        8    IE   Ireland  IE06  Eastern and Midland  IE063  Midlands  B26C8BAA-F3C5-49A9-B74E-D7FED1823E65    0.896836      7.564148  POLYGON ((-7.64673135464132 53.9415530759351, ...  2               ZXD400  2018Q3  QLF07C01    NaN
```

18. Creating additional GeoDataFrames from dfb

We will now create individual GeoDataFrames for males, females and total, by parsing the GeoDataFrame, dfb, where:

- Sex: 1 = Males, 2 = Females, - = Total,
- NACE Rev 2 Economic Sector = F, Construction, and
- Quarter = 2018Q3

```
dfbm = dfb.loc[(dfb['Sex'] == '1') & (dfb['NACE Rev 2 Economic Sector'] == 'F') & (dfb['Quarter'] == '2018Q3')]
dfbf = dfb.loc[(dfb['Sex'] == '2') & (dfb['NACE Rev 2 Economic Sector'] == 'F') & (dfb['Quarter'] == '2018Q3')]
dfbt = dfb.loc[(dfb['Sex'] == '-') & (dfb['NACE Rev 2 Economic Sector'] == 'F') & (dfb['Quarter'] == '2018Q3')]
```

We can output the 1st five lines of each GeoDataFrame, for male, female and total.

```
dfbm.head()
```

```
      OBJECTID  NUTS1  NAME NUTS2  NUTS2NAME                 NUTS3  NUTS3NAME   GUID                                  Shape__Area  Shape__Length   geometry                                              Sex  -  Quarter  Statistic  Value
566         1  IE  Ireland  IE04  Northern and Western  IE041  Border      B1A65D7C-1984-4A87-AD58-0E846812C992  1.594889     31.557961      (POLYGON ((-7.25362176708799 55.4458025525653,...   1      2018Q3  QLF07C01  13.9
2024        2  IE  Ireland  IE04  Northern and Western  IE042  West        A69CA800-8D87-4920-A7C1-50426A1D39B4  1.940981     38.200397      (POLYGON ((-9.524108481921839 53.06827438179, ...   1      2018Q3  QLF07C01  13.3
3482        3  IE  Ireland  IE05  Southern              IE051  Mid-West    42C5C2A5-2D71-4BD1-BDB5-BD7D3198CD78  1.396164     11.020233      (POLYGON ((-9.935623142260511 52.5611963634314...   1      2018Q3  QLF07C01  14.5
4940        4  IE  Ireland  IE05  Southern              IE052  South-East  27C93D4E-AD0E-4B0C-8FA8-3566AEEDA5CC  0.951352     11.412129      (POLYGON ((-7.76097866504068 51.9451840492602,...    1      2018Q3  QLF07C01  12.3
6398        5  IE  Ireland  IE05  Southern              IE053  South-West  604546A1-A856-4B9B-AD46-E88B27C27155  1.610451     32.378775      (POLYGON ((-9.472006032876481 51.4533061425, -...    1      2018Q3  QLF07C01  20.5
```

dfbf.head()

```
 OBJECTID NUTS1  NUTS1NAME NUTS2  NUTS2NAME          NUTS3  NUTS3NAME   GUID                                  Shape_Area  Shape_Length  geometry                                              Sex  -   Quarter  Statistic  Value
1052    1  IE  Ireland  IE04  Northern and Western  IE041  Border      B1A65D7C-1984-4A87-AD58-0E846812C992   1.594889   31.557961  (POLYGON ((-7.25362176708799 55.4458025525653,...  2        2018Q3  QLF07C01  NaN
2510    2  IE  Ireland  IE04  Northern and Western  IE042  West        A69CA800-8D87-4920-A7C1-50426A1D39B4   1.940981   38.200397  (POLYGON ((-9.524108481921839 53.06827438179, ...  2        2018Q3  QLF07C01  NaN
3968    3  IE  Ireland  IE05  Southern              IE051  Mid-West    42C5C2A5-2D71-4BD1-BDB5-BD7D3198CD78   1.396164   11.020233  (POLYGON ((-9.935623142260511 52.5611963634314...  2        2018Q3  QLF07C01  NaN
5426    4  IE  Ireland  IE05  Southern              IE052  South-East  27C93D4E-AD0E-4B0C-8FA8-3566AEEDA5CC   0.951352   11.412129  (POLYGON ((-7.76097866504068 51.9451840492602,...  2        2018Q3  QLF07C01  NaN
6884    5  IE  Ireland  IE05  Southern              IE053  South-West  604546A1-A856-4B9B-AD46-E88B27C27155   1.610451   32.378775  (POLYGON ((-9.472006032876481 51.4533061425, -...  2        2018Q3  QLF07C01  NaN
```

dfbt.head()

```
  OBJECTID NUTS1 NUTS1NAME NUTS2  NUTS2NAME         NUTS3  NUTS3NAME   GUID                                  Shape_Area  Shape_Length  geometry                                             Sex  -  Quarter  Statistic  Value
80      1  IE  Ireland  IE04  Northern and Western  IE041  Border      B1A65D7C-1984-4A87-AD58-0E846812C992   1.594889   31.557961   (POLYGON ((-7.25362176708799 55.4458025525653,...  -     2018Q3  QLF07C01  14.9
1538    2  IE  Ireland  IE04  Northern and Western  IE042  West        A69CA800-8D87-4920-A7C1-50426A1D39B4   1.940981   38.200397   (POLYGON ((-9.524108481921839 53.06827438179, ...  -     2018Q3  QLF07C01  14.4
2996    3  IE  Ireland  IE05  Southern              IE051  Mid-West    42C5C2A5-2D71-4BD1-BDB5-BD7D3198CD78   1.396164   11.020233   (POLYGON ((-9.935623142260511 52.5611963634314...  -     2018Q3  QLF07C01  15.2
4454    4  IE  Ireland  IE05  Southern              IE052  South-East  27C93D4E-AD0E-4B0C-8FA8-3566AEEDA5CC   0.951352   11.412129   (POLYGON ((-7.76097866504068 51.9451840492602,...  -     2018Q3  QLF07C01  12.7
5912    5  IE  Ireland  IE05  Southern              IE053  South-West  604546A1-A856-4B9B-AD46-E88B27C27155   1.610451   32.378775   (POLYGON ((-9.472006032876481 51.4533061425, -...  -     2018Q3  QLF07C01  21.8
```

19. Writing a GeoDataFrame to an Esri shape file

We will now write GeoDataFrame for containing the data for males, dfbm, to an Esri shape file for visualisation, as follows:

```
dfbm.to_file('NUTS3__Generalised_20mA', driver='ESRI Shapefile')
```

A *NUTS3__Generalised_20mA* folder is created in the working directory *'C:\\XXXX\\Python'*.

20. Visualisation with python

One can also visualise a GeoDataFrame in Python using *matplotlib*.

Import the library.

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

One can then preview the GeoDataFrame, *dfa,* as follows:

```
dfa.plot()
```

Which returns the NUTS3 outlines, as it contains no data.

```
Out[21]: <matplotlib.axes._subplots.AxesSubplo
```

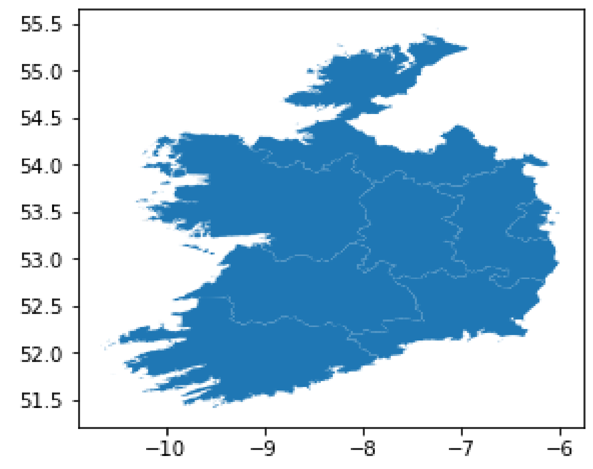


Fig 1: NUTS 3 outline

We can now visualise the GeoDataFrame, *dfbm*, which contains data on the number of Males in the Construction Industry for Quarter 3 of 2018..

We set the column that will be called to be visualised on the map.

```
variable = 'Value'
```

Next, we set up the figure and axis.

```
fig, ax = plt.subplots(1, figsize=(10,6))
```

We now ADD layer of polygons on the axis

```
dfbmpp = dfbm.plot(column=variable,
                   ax=ax,
                   alpha = 0.8,  #intensity of colour of a polygon
                   cmap=''Reds', #colour of the polygon
                   legend='True',
                   linewidth=0.3,
                   edgecolor='0.8',
                   label = 'Males')
```

We will also turn off the axis seen in **Figure 1**, as follows.

```
ax.axis('off')
```

A figure title is added as follows,

```
ax.set_title('Numbers of Males at work in the Construction Industry, Q3 2018')
```

and a legend title is set.

```
ax.legend(title='(000)')
```

Now we can view the map.

```
ax.figure
```

which outputs:

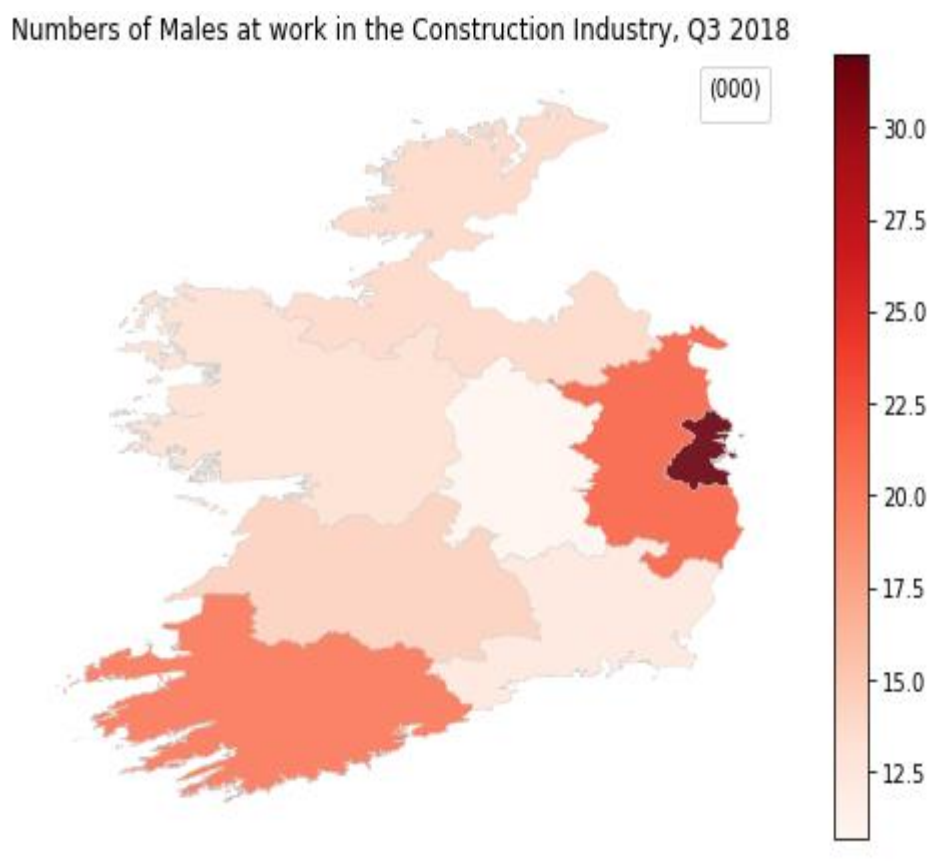

Numbers of Males at work in the Construction Industry, Q3 2018
(000)
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5

Fig 2: *'Numbers of Males at work in the Construction Industry, Q3 2018'*

21 Converting matplotlib plots from Python into interactive Leaflet web maps.

One can convert matplotlib plots from Python into interactive Leaflet web maps using the ***mplleaflet*** library:

Leaflet does not readily support the use of legends and requires very dark Choropleth map colours. We will return the code above leaving out the legend and changing the colour to 'Darkt2_r' to produce a new *ax.figure.*

```
variable = 'Value'
fig, ax = plt.subplots(1, figsize=(10,6))
dfbmpp = dfbm.plot(column=variable,
                   ax=ax,
                   alpha = 0.8,  #intensity of colour of a polygon
                   cmap=''Dark2_r, #colour of the polygon
                   legend='True',
                   linewidth=0.3,
                   edgecolor='0.8',
                   label = 'Males')
ax.axis('off')
ax.set_title('Numbers of Males at work in the Construction Industry, Q3 2018')
ax.figure
```

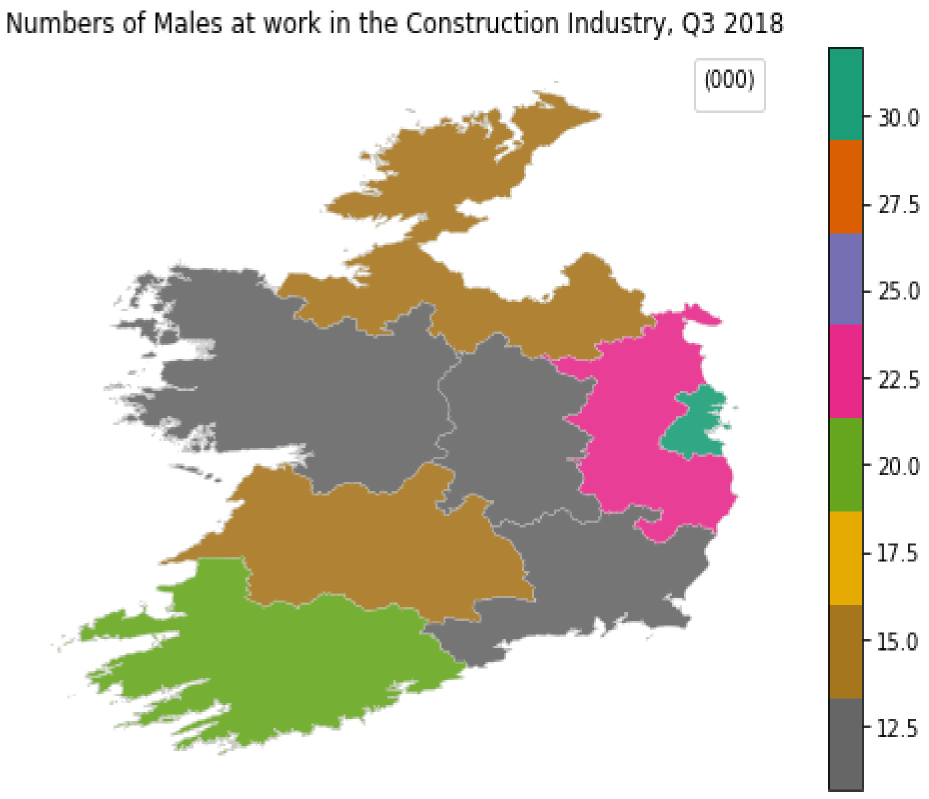

Numbers of Males at work in the Construction Industry, Q3 2018
(000)
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5

Fig 2: *'Numbers of Males at work in the Construction Industry, Q3 2018'*

We now Import the mplleaflet library,

```
import mplleaflet
```

and Output _map.html file and display it in your browser.

```
mplleaflet.show(fig=ax.figure)
```

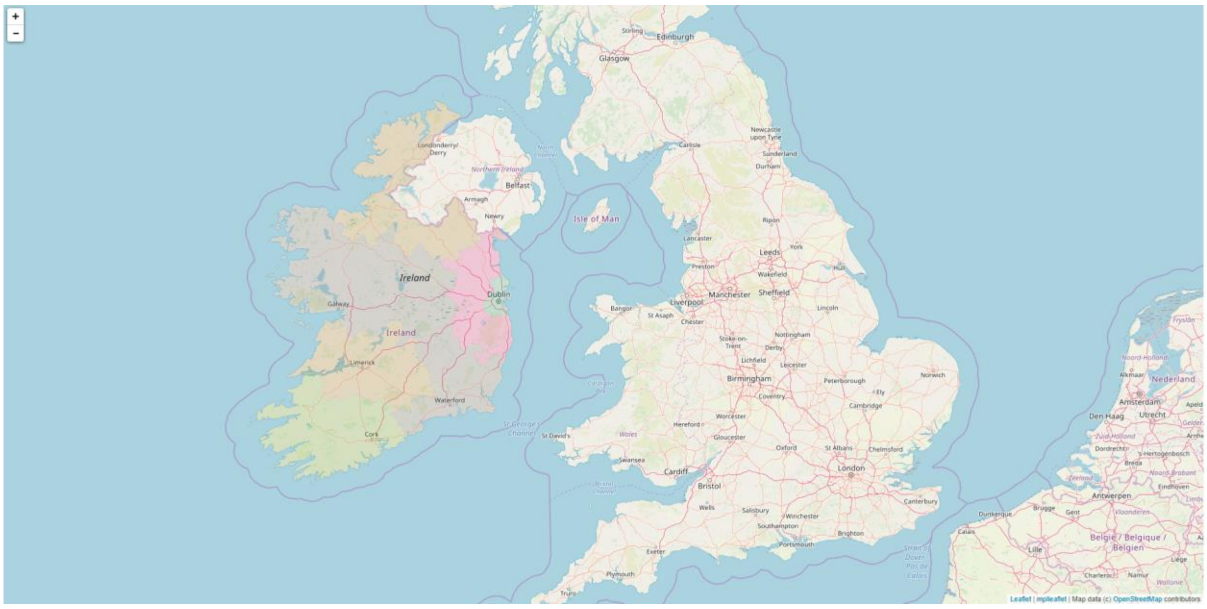


Fig 3: html map, displayed in browser.