Memahami Pesan di balik Data, Instalasi dan Analisis Data Sederhana Menggunakan Python untuk Windows 10

Apa yang terpikirkan di benak anda jika mendengar kata “data”? beberapa orang yang pernah saya tanyakan hal ini yang menjawab data adalah kumpulan angka, tidak salah hanya saja kurang tepat. Data tidak hanya berisikan angka saja melainkan gambar, teks, video, suara dll yang menggambarkan sebuah fakta atau peristiwa.

Lantas bagaimana caranya kita memahami pesan dari data? Hal yang harus dilakukan adalah mengolah data tersebut agar menjadi sebuah informasi. Akan sangat menyulitkan jika proses pengolahan data di lakukan secara manual. Saat ini banyak sekali *software* pengolah data yang dapat kita instal baik itu gratis maupun berbayar.

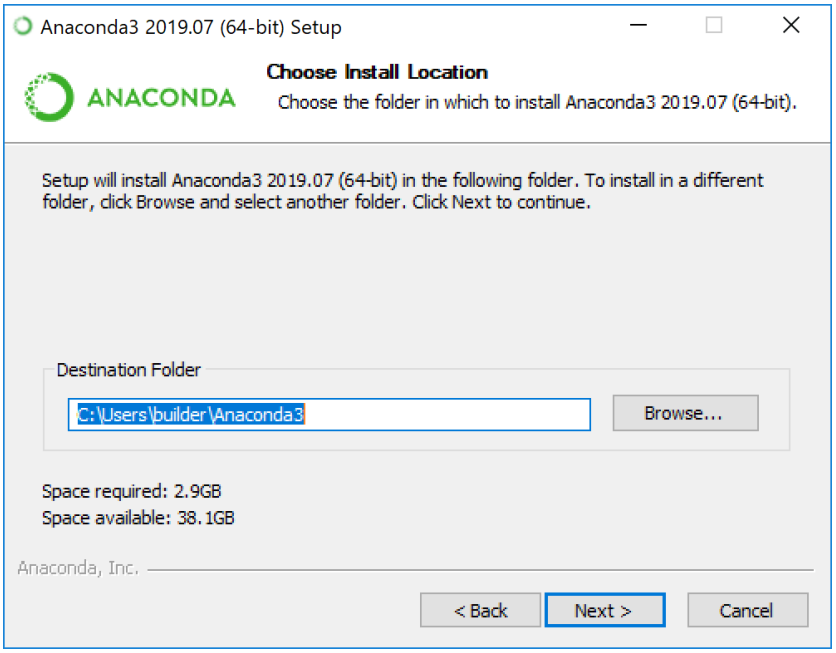
Salah satu *software* pengolahan data yang cukup populer dalam analisis data adalah python. Python merupakan *software open source* yang diciptakan oleh Guido van Rossem pada tahun 1991 dan lebih menekankan pada produktivitas juga proses pembacaan kode. Python sendiri tidak memiliki IDE (I*ntegrated Development Environment*) yang jelas seperti Spyder dan Jupyter Notebook. Python IDE yang saya sering gunakan adalah Jupyter Notebook. Jupyter Notebook adalah aplikasi web *open-source* yang memungkinkan kita membuat dan berbagi dokumen interaktif yang berisi kode *live*, persamaan, visualisasi, dan teks naratif yang kaya. Untuk menginstall Jupyter Notebook akan lebih mudah menggunakan Anaconda.

Instalasi Anaconda

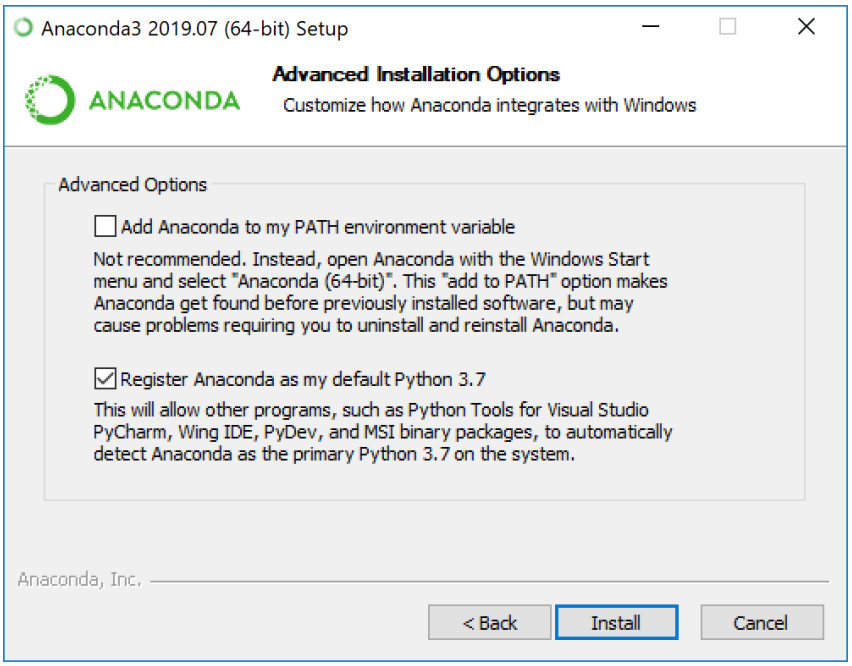
Sekarang kita akan mencoba instalasi anaconda di Windows 10.

1. Download installer anaconda disini. Pilih Python 3.7 version dan sesuai dengan windows kalian 32bit atau 64bit.
2. Setelah terdownload double klik pada installer, kemudian klik next.

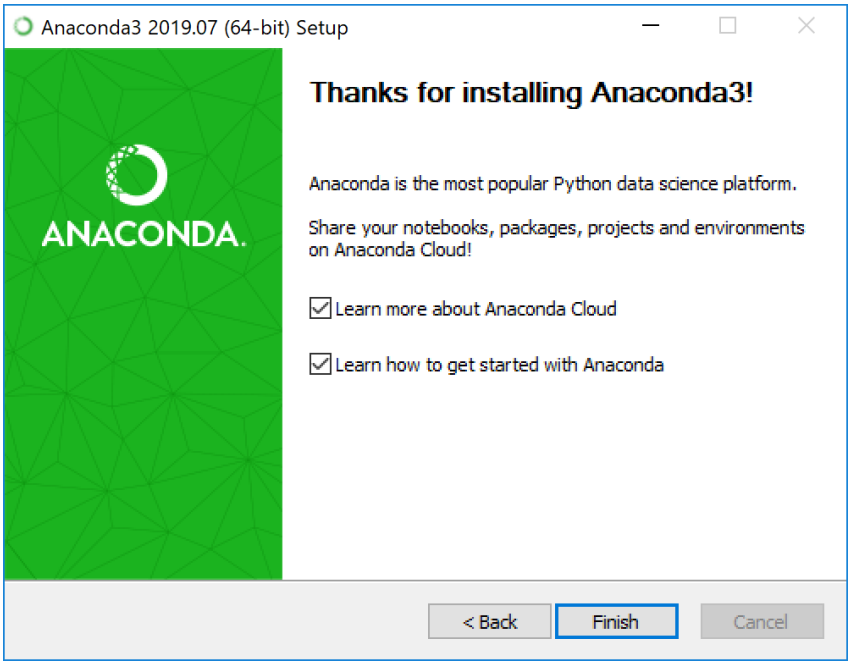
3. Klik ‘I Agree’ untuk melanjutkan, klik Just Me, kemudian klik next
4. Pilih lokasi yang diinginkan atau bisa dengan default lalu, klik next.



5. Centang Register Anaconda as my default, lalu klik install.



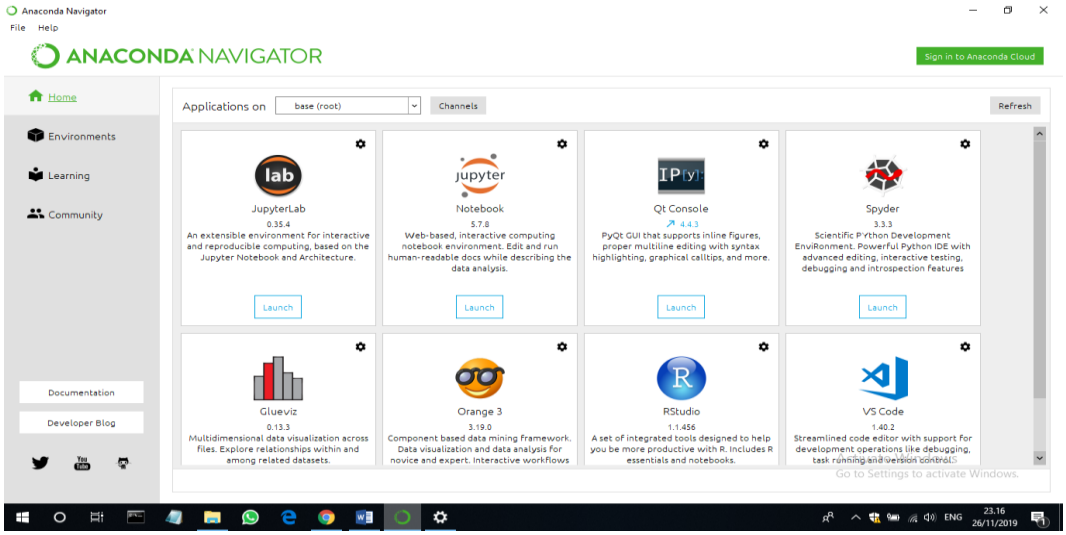
6. Tunggu proses instalasi sampai selesai lalu klik finish



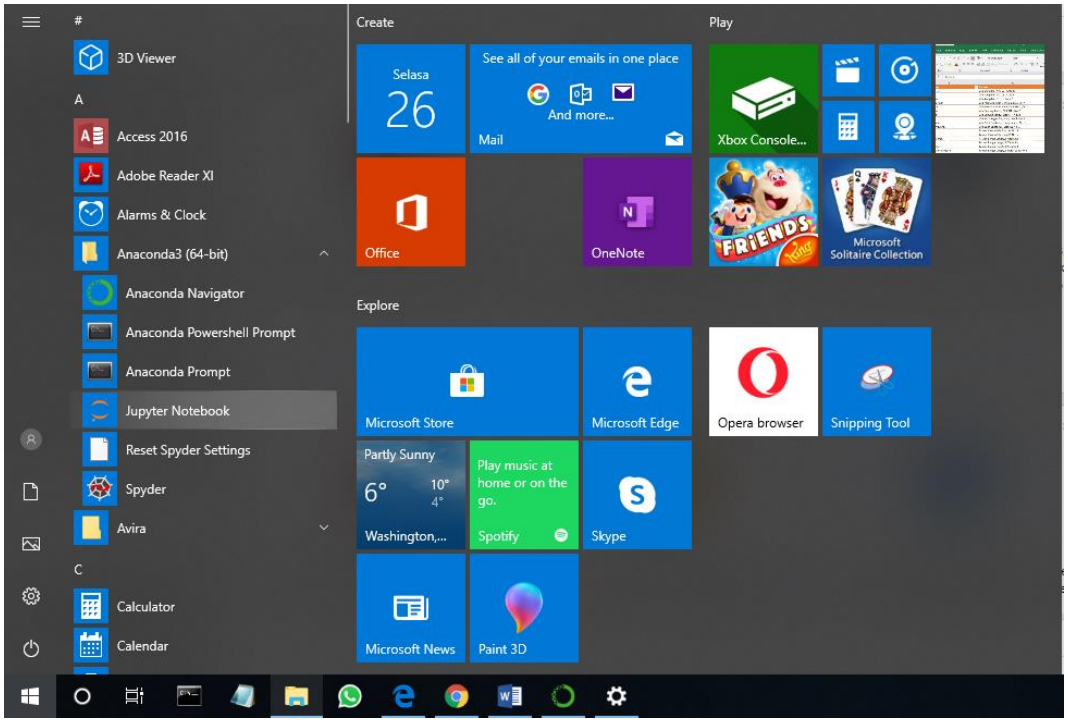
7. Setelah instalasi selesai, Jika terdapat kendala pada proses instalasi silahakan [kesini.]

8. Jalankan Anaconda Navigator. Klik Start dan klik folder Anaconda kemudian klik Anconda Navigator.

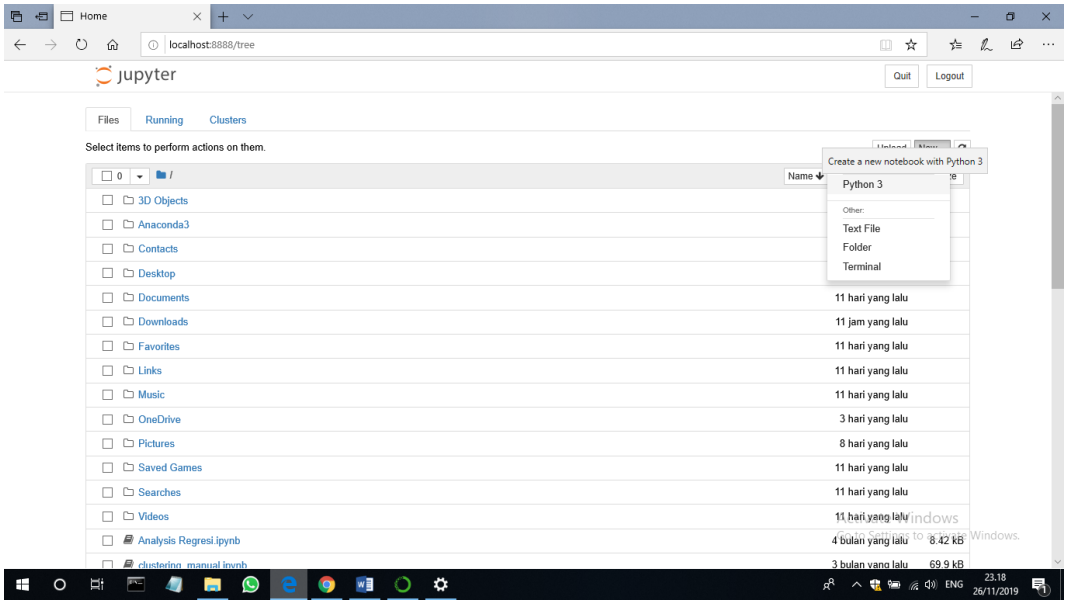
9. Tampilan Anconda akan terlihat seperti gambar dibawah ini. Kemudian Klik **Lunch** Jupyter Notebook.



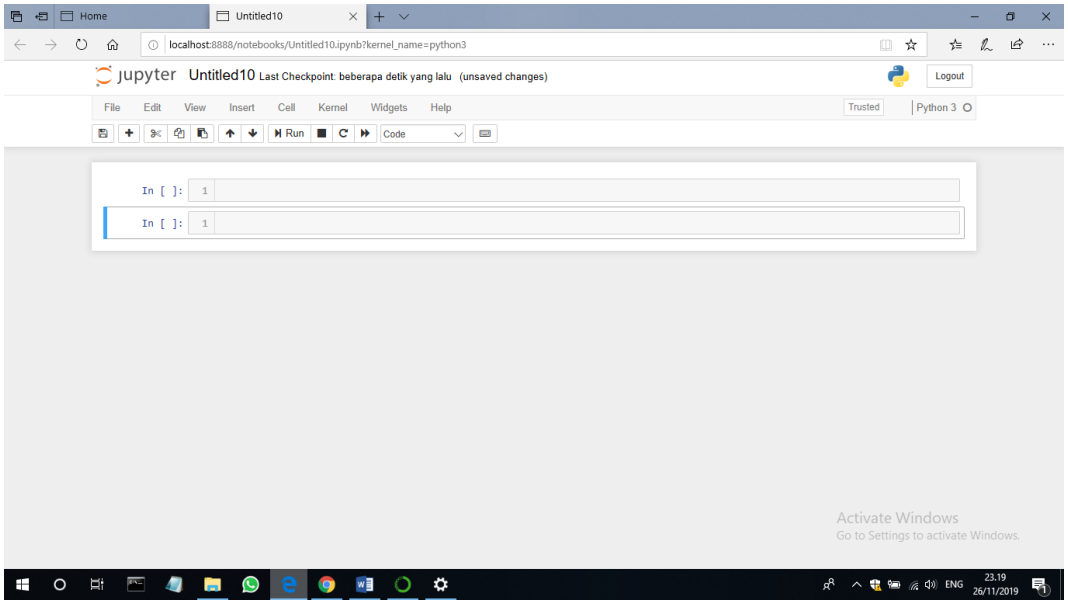
Atau lansung menjalankan Jupyter langsung tanpa harus membuka Anaconda Navigator. Seperti gambah di bawah ini.



10. Tunggulah sampai Jupyter terbuka di di browser kita, Klik **New** kemudian pilih **python 3.**



Jika langkah diatas dilakukan dengan benar maka tampilan Jupyter Notebook akan terlihat seperti gambar dibawah ini. Hore! Jupyter Notebook siap kita gunakan untuk analisis.



Analisis Data

Langkah pertama yang dilakukan dalam proses analisis data adalah eksplorasi data. Dalam statistik, eksplorasi data merupakan pendekatan untuk menganalisis data untuk merangkum karakteristik utama data. Pada tutorial ini kita akan menggunakan data WHO.csv. Kumpulan data ini merupakan statistik dari Organisasi Kesehatan Dunia (WHO) dari seluruh negara Berikut adalah detail dari variable dari data WHO :

- **Country**, nama negara,
- **Region**
- **Population**, Populasi per 1.000 penduduk,
- **Under15**, presentase populasi di bawah 15 tahun
- **Over 60**, presentase populasi lebih dari 60 tahun,
- **FertilityRate**, tingkat kesuburan atau jumlah rata-rata anak per wanita,
- **LifeExpectancy**, angka harapan hidup,
- **ChildMortality**, jumlah anak yang meninggal pada usia 5 tahun per 1.000 kelahiran,
- C**ellularSubribers** jumlah pelanggan seluler per 100 populasi,
- **LiteracyRate** adalah tingkat melek huruf di kalangan orang dewasa berusia lebih besar dari atau sama dengan 15,
- **GNI** adalah pendapatan nasional bruto per kapita,
- **PrimarySchoolEnrollMale** persentase anak laki-laki yang terdaftar di sekolah dasar, dan
- **PrimarySchoolEnrollfemale** adalah persentase anak perempuan terdaftar di sekolah dasar.)

Loading data set

Ketikan kode dibawah ini kemudian tekan Shift + Enter untuk run kode

In [1]:
```
import pandas as pd
df = pd.read_csv("D://Databank//WHO.csv")
#untuk melihat data teratas
df.head()
```

Out[1]:

	Country	Region	Population	Under15	Over60	FertilityRate	LifeExpectancy	ChildMortality	CellularSubscribers	LiteracyRate	GNI	PrimarySchoolE
0	Afghanistan	Eastern Mediterranean	29825	47.42	3.82	5.40	60	98.5	54.26	NaN	1140.0	
1	Albania	Europe	3162	21.33	14.93	1.75	74	16.7	96.39	NaN	8820.0	
2	Algeria	Africa	38482	27.42	7.17	2.83	73	20.0	98.99	NaN	8310.0	
3	Andorra	Europe	78	15.20	22.86	NaN	82	3.2	75.49	NaN	NaN	
4	Angola	Africa	20821	47.58	3.84	6.10	51	163.5	48.38	70.1	5230.0	

```
In [2]:
#untuk melihat data terbawah
df.tail()
```

Out[2]:

	Country	Region	Population	Under15	Over60	FertilityRate	LifeExpectancy	ChildMortality	CellularSubscribers	LiteracyRate	GNI	PrimarySch
189	Venezuela (Bolivarian Republic of)	Americas	29955	28.84	9.17	2.44	75	15.3	97.78	NaN	12430.0	
190	Viet Nam	Western Pacific	90796	22.87	9.32	1.79	75	23.0	143.39	93.2	3250.0	
191	Yemen	Eastern Mediterranean	23852	40.72	4.54	4.35	64	60.0	47.05	63.9	2170.0	
192	Zambia	Africa	14075	46.73	3.95	5.77	55	88.5	60.59	71.2	1490.0	
193	Zimbabwe	Africa	13724	40.24	5.68	3.64	54	89.8	72.13	92.2	NaN	

```
In [3]:
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 194 entries, 0 to 193
Data columns (total 13 columns):
Country                          194 non-null object
Region                           194 non-null object
Population                       194 non-null int64
Under15                          194 non-null float64
Over60                           194 non-null float64
FertilityRate                    183 non-null float64
LifeExpectancy                   194 non-null int64
ChildMortality                   194 non-null float64
CellularSubscribers              184 non-null float64
LiteracyRate                     103 non-null float64
GNI                              162 non-null float64
PrimarySchoolEnrollmentMale      101 non-null float64
PrimarySchoolEnrollmentFemale    101 non-null float64
dtypes: float64(9), int64(2), object(2)
memory usage: 19.8+ KB
```

Dengan info() kita dapat melihat struktur dataset, dataset WHO memiliki 194 observasi dan 13 variabel, dan didataset terdapat 3 tipe data yaitu object, int64 dan float64. Silahkan kesini untuk penjelasan lebih lanjut mengenai tipe data di python.

```
In [4]:
df.describe()
```

Out[4]:

	Population	Under15	Over60	FertilityRate	LifeExpectancy	ChildMortality	CellularSubscribers	LiteracyRate	GNI	PrimarySchoolEnrollm
count	1.940000e+02	194.000000	194.000000	183.000000	194.000000	194.000000	184.000000	103.000000	162.000000	101
mean	3.635997e+04	28.732423	11.163660	2.940656	70.010309	36.148969	93.641522	83.710680	13320.925926	9(
std	1.379031e+05	10.534573	7.149331	1.480984	9.259075	37.992935	41.400447	17.530645	15192.988650	1
min	1.000000e+00	13.120000	0.810000	1.260000	47.000000	2.200000	2.570000	31.100000	340.000000	3
25%	1.695750e+03	18.717500	5.200000	1.835000	64.000000	8.425000	63.567500	71.600000	2335.000000	87
50%	7.790000e+03	28.650000	8.530000	2.400000	72.500000	18.600000	97.745000	91.800000	7870.000000	9
75%	2.453525e+04	37.752500	16.687500	3.905000	76.000000	55.975000	120.805000	97.850000	17557.500000	9
max	1.390000e+06	49.990000	31.920000	7.580000	83.000000	181.600000	196.410000	99.800000	86440.000000	10

Syntax describe() menampilkan *summary* statistik terdapat **count** yang merupakan jumlah observasi/data, **mean** (rata-rata), **std** (standar deviasi), **min** (nilai terkecil data), **max** (nilai terbesar data), 25% (kuartil 1) , 50% (median) dan 75%(kuartil 3). Jika diperhatikan syntax ini hanya menghitung tipe data numerical/angka, terlihat variabel **Country** dan

Region tidak termasuk karena keduanya bukan data numerik/angka. Mari kita investigasi lebih lanjut!

```
In [5]: df['Region'].value_counts()
```

```
Out[5]: Europe                  53
        Africa                  46
        Americas                35
        Western Pacific         27
        Eastern Mediterranean   22
        South-East Asia         11
        Name: Region, dtype: int64
```

Dari output diatas kita dapat menarik kesimpulan bahwa negara terbanyak yaitu pada benua Eropa sebanyak 53 negara dan yang terkecil merupakan Asia Tenggara sebanyak 11 negara (dalam dataset yang kita miliki).

Kembali ke output describe(), untuk variabel **Population** (selanjutnya disebut populasi) didapat mean = 36359.974 per 1,000 penduduk atau sebanyak 3,6359,974 penduduk, dengan populasi terbesar sebanyak 1,390,000,000 penduduk, dan populasi terkecil sebanyak 1,000 penduduk. Dari sini kita sudah mendapatkan nilai **mean**,**min** dan **max** dari populasi, akan tetapi kita tidak mendapatkan informasi secara spesifik negara manakah memiliki nilai tersebut. maka dari itu kita perlu mengeksplor lebih dalam.

```
In [6]: df.loc[df['Population'].idxmax()]
```

```
Out[6]: Country                          China
        Region                  Western Pacific
        Population                   1390000
        Under15                        17.95
        Over60                         13.42
        FertilityRate                   1.66
        LifeExpectancy                    76
        ChildMortality                    14
        CellularSubscribers            73.19
        LiteracyRate                    94.3
        GNI                             8390
        PrimarySchoolEnrollmentMale      NaN
        PrimarySchoolEnrollmentFemale    NaN
        Name: 35, dtype: object
```

```
In [7]: df.loc[df['Population'].idxmin()]
```

```
Out[7]: Country                          Niue
        Region                  Western Pacific
        Population                           1
        Under15                          30.61
        Over60                            9.07
        FertilityRate                      NaN
        LifeExpectancy                      72
        ChildMortality                    25.1
        CellularSubscribers                NaN
        LiteracyRate                       NaN
        GNI                                NaN
        PrimarySchoolEnrollmentMale        NaN
        PrimarySchoolEnrollmentFemale      NaN
        Name: 125, dtype: object
```

Terlihat jelas bahwa negara China memiliki polulasi terbesar dan negara Niue memiliki populasi terkecil. Sejauh ini kita sudah mendapatkan informasi dari populasi, dengan syntax yang sama seperti diatas kita akan coba mengeksplor variabel **Under15** dan **Over 60**.

In [8]:
```
df['Under15'].describe()
```

Out[8]:
```
count    194.000000
mean      28.732423
std       10.534573
min       13.120000
25%       18.717500
50%       28.650000
75%       37.752500
max       49.990000
Name: Under15, dtype: float64
```

In [9]:
```
df.loc[df['Under15'].idxmin()]
```

Out[9]:
```
Country                              Japan
Region                     Western Pacific
Population                          127000
Under15                              13.12
Over60                               31.92
FertilityRate                         1.39
LifeExpectancy                          83
ChildMortality                           3
CellularSubscribers                 104.95
LiteracyRate                           NaN
GNI                                  35330
PrimarySchoolEnrollmentMale            NaN
PrimarySchoolEnrollmentFemale          NaN
Name: 85, dtype: object
```

In [10]:
```
df.loc[df['Under15'].idxmax()]
```

Out[10]:
```
Country                          Niger
Region                          Africa
Population                       17157
Under15                          49.99
Over60                            4.26
FertilityRate                     7.58
LifeExpectancy                      56
ChildMortality                   113.5
CellularSubscribers              29.52
LiteracyRate                       NaN
GNI                                720
PrimarySchoolEnrollmentMale       64.2
PrimarySchoolEnrollmentFemale       52
Name: 123, dtype: object
```

In [11]:
```
df['Over60'].describe()
```

Out[11]:
```
count    194.000000
mean      11.163660
std        7.149331
min        0.810000
25%        5.200000
50%        8.530000
75%       16.687500
max       31.920000
Name: Over60, dtype: float64
```

In [12]:
```
df.loc[df['Over60'].idxmax()]
```

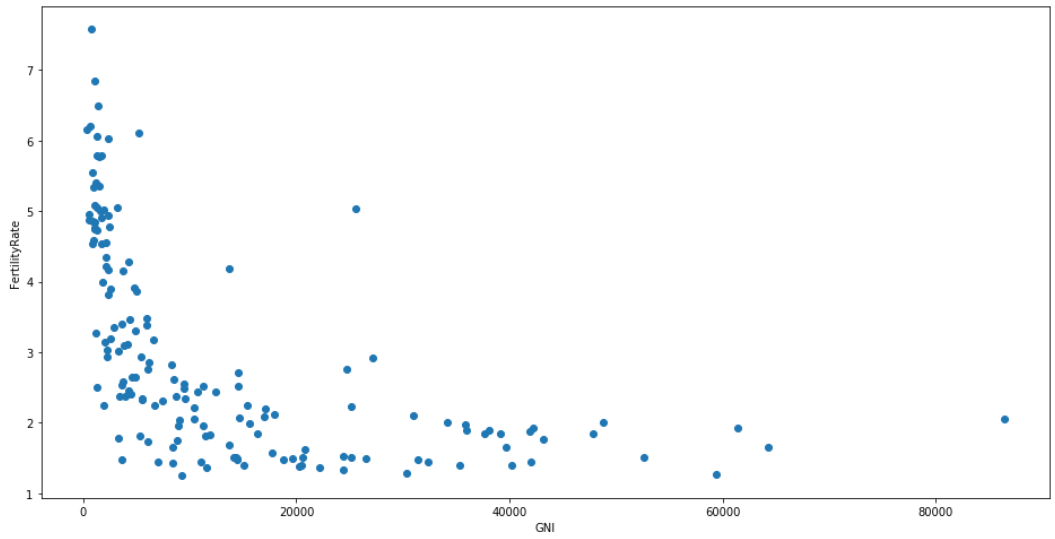
Out[12]:
```
Country                              Japan
Region                     Western Pacific
Population                          127000
Under15                              13.12
Over60                               31.92
FertilityRate                         1.39
LifeExpectancy                          83
ChildMortality                           3
CellularSubscribers                 104.95
LiteracyRate                           NaN
GNI                                  35330
PrimarySchoolEnrollmentMale            NaN
PrimarySchoolEnrollmentFemale          NaN
Name: 85, dtype: object
```

```
In [13]: 1 df.loc[df['Over60'].idxmin()]

Out[13]: Country                          United Arab Emirates
         Region                           Eastern Mediterranean
         Population                                        9206
         Under15                                          14.41
         Over60                                            0.81
         FertilityRate                                     1.84
         LifeExpectancy                                      76
         ChildMortality                                     8.4
         CellularSubscribers                             148.62
         LiteracyRate                                       NaN
         GNI                                              47890
         PrimarySchoolEnrollmentMale                        NaN
         PrimarySchoolEnrollmentFemale                      NaN
         Name: 182, dtype: object
```

Untuk populasi di bawah 15 tahun negara Jepang adalah negara yang memiliki presentase terkecil sebesar 13% dan negara Niger memiliki presentase sebesar 49.99 % dari populasi penduduk negaranya. Untuk Populasi lebih dari 60 tahun negara United Arab Emirates adalah negara yang memiliki presentase terkecil sebesar 1% dan negara Japan memiliki presentase sebesar 31.92% dari populasi penduduk negaranya.

Kita akan mencoba visualisasi di python sebelum itu silihkan install library matplotlib dengan mengetikan kode pip install matplotlib (pastikan terkoneksi ke internet)

```
In [14]: 1 import matplotlib.pyplot as plt

In [15]: 1 fig, ax = plt.subplots(figsize=(16,8))
         2 ax.scatter(df['GNI'], df['FertilityRate'])
         3 ax.set_xlabel('Pendapan Nasional Bruto')
         4 ax.set_ylabel('Tingkat Kesuburan')
```



Dari grafik terlihat banyak negara dengan GNI < 20,000 memiliki *fertility rate* yang cenderung tinggi.

Kesimpulan

Dari analsis sederhana yang telah dilakukan dapat kita tarik kesimpulan bahwa China merupakan negara dengan populasi terbesar dan negara Niue merupakan negara dengan populasi terkecil, lebih lengkapnya lihat tabel berikut

Negara	Populasi	Dibawah 15 tahun		Diatas 60 tahun		Usia Produktif	
		Populasi	presentase	populasi	presentase	populasi	presentase
Japan	127.000.000	16.662.400	13,12%	40.538.400	31,92%	69.799.200	54,96%
Niger	1.715.700	857.678	49,99%	73.089	4,26%	784.933	45,75%
United Arab Emirates	9.206.000	1.326.585	14,41%	74.569	0,81%	7.804.847	84,78%

Negara niger hampir setengah penduduknya berusia di bahwa 15 tahun, berbanding terbalik dengan jepang yang hanya 1/8 penduduknya yang berusia dibawah 15 tahun dan United Arab Emirates yang hampir seluruh penduduknya berusia produktif. Banyak negara dengan GNI yang rendah cenderung memiliki fertility rate yang tinggi.

Analisis sederhana ini hanya sebagian yang baru saja kita gali, masih banyak informasi yang bisa kita temukan dari dataset ini. Untuk analisis lebih lanjut akan dibahas di lain kesempatan. Terima Kasih.