

In [1]:

```
# coding: utf-8

# # ONE: Import necessary files and libraries
#

# In[1]:

import pandas as pd
import numpy as np
```

In [4]:

```
# Declare variable for each file
cell= pd.read_csv(r"C:\Users\DELL 5440\Downloads\cell_phones.csv")
population= pd.read_csv(r"C:\Users\DELL 5440\Downloads\population.csv")
geocountrycode = pd.read_csv(r"C:\Users\DELL 5440\Downloads\country.csv",header=0)
```

In [5]:

```
# # TWO: Join data

# In[2]:

datajoin=pd.merge(cell,population, on=["Country", "year"],how="inner")
print(datajoin)
datajoin.shape
```

	Country	year	cell_phones_total	population
0	Afghanistan	1960	0.0	9000000
1	Afghanistan	1961	0.0	9170000
2	Afghanistan	1962	0.0	9350000
3	Afghanistan	1963	0.0	9540000
4	Afghanistan	1964	0.0	9740000
...
11635	Zimbabwe	2015	12800000.0	13800000
11636	Zimbabwe	2016	12900000.0	14000000
11637	Zimbabwe	2017	14100000.0	14200000
11638	Zimbabwe	2018	12900000.0	14400000
11639	Zimbabwe	2019	13200000.0	14600000

[11640 rows x 4 columns]

Out[5]:

(11640, 4)

In [6]:

```
# In[3]:

# renaming columns
geocountrycode=geocountrycode.rename(columns={'country': 'geo', 'name': 'Country'})
geocountrycode
```

Out[6]:

	geo	g77_and_oecd_countries	income_3groups	income_groups	is--country	iso3166_1_alpha2	iso3166_1_alpha3
0	abkh	others	NaN	NaN	True	NaN	NaN
1	abw	others	high_income	high_income	True	AW	ABW
2	afg	g77	low_income	low_income	True	AF	AFG

3	g77	g77_and_oecd_countries	g77	middle_income	lower_middle_income	True	iso3166_1_alpha2	iso3166_1_alpha3
4	aia	others		NaN	NaN	True	AI	ALA
...
268	yem_south	others		NaN	NaN	True	NaN	NaN
269	yug	others		NaN	NaN	True	NaN	NaN
270	zaf	g77	middle_income	upper_middle_income		True	ZA	ZAF
271	zmb	g77	middle_income	lower_middle_income		True	ZM	ZMB
272	zwe	g77	middle_income	lower_middle_income		True	ZW	ZWE

273 rows x 21 columns



In [7]:

```
# In[9]:

newdat=datajoin.merge(geocountrycode[['geo','Country']])
newdat.shape
newdat
```

Out[7]:

	Country	year	cell_phones_total	population	geo
0	Afghanistan	1960	0.0	9000000	afg
1	Afghanistan	1961	0.0	9170000	afg
2	Afghanistan	1962	0.0	9350000	afg
3	Afghanistan	1963	0.0	9540000	afg
4	Afghanistan	1964	0.0	9740000	afg
...
11635	Zimbabwe	2015	12800000.0	13800000	zwe
11636	Zimbabwe	2016	12900000.0	14000000	zwe
11637	Zimbabwe	2017	14100000.0	14200000	zwe
11638	Zimbabwe	2018	12900000.0	14400000	zwe
11639	Zimbabwe	2019	13200000.0	14600000	zwe

11640 rows x 5 columns

In [105]:

```
# # part Three

# In[17]:

#calculate the number of cell phones per person and add this column to the dataframe
cellsum =newdat ['cell_phones_total']/newdat ['population']
cellsum
```

Out[105]:

0	0.000000
1	0.000000
2	0.000000
3	0.000000
4	0.000000

```

...
11635    0.927536
11636    0.921429
11637    0.992958
11638    0.895833
11639    0.904110
Length: 11640, dtype: float64

```

In [111]:

```

#adding this column onto your dataframe
newdat['cellsum'] = cellsum

```

In [109]:

```
print (newdat)
```

	Country	year	cell_phones_total	population	geo	cellsum
0	Afghanistan	1960	0.0	9000000	afg	0.000000
1	Afghanistan	1961	0.0	9170000	afg	0.000000
2	Afghanistan	1962	0.0	9350000	afg	0.000000
3	Afghanistan	1963	0.0	9540000	afg	0.000000
4	Afghanistan	1964	0.0	9740000	afg	0.000000
...
11635	Zimbabwe	2015	12800000.0	13800000	zwe	0.927536
11636	Zimbabwe	2016	12900000.0	14000000	zwe	0.921429
11637	Zimbabwe	2017	14100000.0	14200000	zwe	0.992958
11638	Zimbabwe	2018	12900000.0	14400000	zwe	0.895833
11639	Zimbabwe	2019	13200000.0	14600000	zwe	0.904110

[11640 rows x 6 columns]

In [9]:

```

def Average(cellsum):
    return sum(cellsum) / len(cellsum)

```

In [28]:

```

lst = [cellsum]
average = Average(lst)

print("Average of the list =", round(average, 50))

```

```

Average of the list = 0          0.000000
1          0.000000
2          0.000000
3          0.000000
4          0.000000
...
11635    0.927536
11636    0.921429
11637    0.992958
11638    0.895833
11639    0.904110
Length: 11640, dtype: float64

```

In [45]:

```
geocountrycode = pd.Series('geo')
```

In [68]:

```
geocountrycode.str.upper()
```

Out[68]:

```

0      GEO
dtype: object

```

In [55]:

```
# An overview of the dataframe's contents
newdat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11640 entries, 0 to 11639
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Country                11640 non-null  object
1   year                   11640 non-null  int64
2   cell_phones_total      11640 non-null  float64
3   population              11640 non-null  int64
4   geo                    11640 non-null  object
dtypes: float64(1), int64(2), object(2)
memory usage: 545.6+ KB
```

In [57]:

```
# statistical data
newdat.describe()
```

Out[57]:

	year	cell_phones_total	population
count	11640.000000	1.164000e+04	1.164000e+04
mean	1989.500000	7.875722e+06	2.710307e+07
std	17.318846	5.572496e+07	1.086187e+08
min	1960.000000	0.000000e+00	4.380000e+03
25%	1974.750000	0.000000e+00	1.117500e+06
50%	1989.500000	0.000000e+00	5.100000e+06
75%	2004.250000	5.615000e+05	1.590000e+07
max	2019.000000	1.730000e+09	1.430000e+09

In [64]:

```
#dataframe's numeric columns
newdat.select_dtypes(include=['float64'])
```

Out[64]:

	cell_phones_total
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
...	...
11635	12800000.0
11636	12900000.0
11637	14100000.0
11638	12900000.0
11639	13200000.0

11640 rows x 1 columns

In [65]:

```
# non-numeric columns
```

```
# non-numeric columns.
newdat.select_dtypes(exclude=['int64'])
```

Out[65]:

	Country	cell_phones_total	geo
0	Afghanistan	0.0	afg
1	Afghanistan	0.0	afg
2	Afghanistan	0.0	afg
3	Afghanistan	0.0	afg
4	Afghanistan	0.0	afg
...
11635	Zimbabwe	12800000.0	zwe
11636	Zimbabwe	12900000.0	zwe
11637	Zimbabwe	14100000.0	zwe
11638	Zimbabwe	12900000.0	zwe
11639	Zimbabwe	13200000.0	zwe

11640 rows x 3 columns

In [72]:

```
#top country with the most cell phones total
newdat.max()
```

Out[72]:

Country Zimbabwe
year 2019
cell_phones_total 1.73e+09
population 1430000000
geo zwe
dtype: object

In [92]:

```
#top 5 countries with the most cell phones total
newdat.max(axis=None, skipna=None, level = 0, numeric_only=None)
```

Out[92]:

	Country	year	cell_phones_total	population	geo
0	Afghanistan	1960	0.0	9000000	afg
1	Afghanistan	1961	0.0	9170000	afg
2	Afghanistan	1962	0.0	9350000	afg
3	Afghanistan	1963	0.0	9540000	afg
4	Afghanistan	1964	0.0	9740000	afg
...
11635	Zimbabwe	2015	12800000.0	13800000	zwe
11636	Zimbabwe	2016	12900000.0	14000000	zwe
11637	Zimbabwe	2017	14100000.0	14200000	zwe
11638	Zimbabwe	2018	12900000.0	14400000	zwe
11639	Zimbabwe	2019	13200000.0	14600000	zwe

11640 rows x 5 columns

In []:

