In [63]:
```
import json
import pandas as pd
import matplotlib.pyplot as plt
```

## Accessing Developers Tools

Firefox and Chrome: Options --> More Tools --> Web Developer Tools

# Right click on the html code selected on the right side of the screen and Select: Copy --> Inner HTML

Paste the content of the clip paper into a text editor and save the content as txt file, for example.

I have the content saved as `contacts.txt`

# 1. Read the content of the text file

In [27]:
```
with open('contacts.txt' , 'r') as file:
    contacts = file.readlines()
```

## Hidden by privacy purposes

In [95]:
```
#contacts
```

## Now let's do some cleaning

Firstly, I have a long string with phone numbers, separated by comma. I will use `.split(', ')` to separate them.

In [29]:
```
phone_numbers = contacts[0].split(', ')
```

## Hidden by privacy purposes

In [96]:
```
#phone_numbers[:5]
```

Secondly, I will take out the country code for each number. They are usually the first group of numbers.

In [31]:
```
country_codes = [i.split(' ')[0] for i in phone_numbers]
country_codes[:10]
```

Out[31]:
```
['+91', '+234', '+234', '+234', '+234', '+254', '+234', '+234', '+234',
'+234']
```

## It looks like Nigeria and Ghana are in da house...

## Now let's make use of pandas to have a count for distinct country codes

In [32]:
```
country_codes_series = pd.Series(country_codes)
country_codes_series
```

Out[32]:
```
0       +91
1      +234
2      +234
3      +234
4      +234
       ...
178     +91
179     +91
180     +91
181     +91
182     +91
Length: 183, dtype: object
```

## And now, get the count of unique country codes

In [33]:
```
country_codes_count = country_codes_series.value_counts().sort_values(asc
country_codes_count
```

Out[33]:
```
+234    122
+233     14
+91      13
+254      7
+256      5
+44       4
+27       3
+263      3
+62       2
+60       1
+54       1
+55       1
+92       1
+1        1
+90       1
+255      1
+48       1
+7        1
+261      1
dtype: int64
```

## Now that we have the count values, we need to assign the country code to the country name

We need some help. I had to look up in the internet for some listing of the country names and country codes

## Load the country codes and country phone preffixes as a python dictionary, using a json file.

In [34]:
```
with open('phone_codes.json', 'r') as file:
    phone_codes = json.load(file)
```

In [10]:
```
#phone_codes
```

## Let's store it in a dataframe

In [35]:
```
country_codes_df = pd.DataFrame( {'country_code': phone_codes.keys(),'pre
```

In [36]:
```
country_codes_df.sample(5)
```

Out[36]:

| | country_code | preffix |
|---|---|---|
| **231** | IS | 354 |
| **40** | GP | 590 |
| **28** | TL | 670 |
| **146** | CI | 225 |
| **113** | MQ | 596 |

## Now load a different json file to obtain the country names for each country code

In [37]:
```
with open('country_names.json', 'r') as file:
    country_names = json.load(file)
```

In [14]:
```
#country_names
```

## Store the country names and codes to a pandas dataframe

In [38]:
```
country_names_df = pd.DataFrame( {'country_code': country_names.keys(),'c
country_names_df.sample(5)
```

Out[38]:

| | country_code | country_name |
|---|---|---|
| **148** | CO | Colombia |
| **207** | TT | Trinidad and Tobago |
| **41** | JP | Japan |
| **10** | BO | Bolivia |
| **1** | BE | Belgium |

## Now we need a combined dataframe where we can have `country_code`, `preffix` and `country_name`

I will use: `pd.merge(df1, df2, on='common_columnn')`

In [39]:
```
data = pd.merge(country_codes_df, country_names_df, on='country_code')
```

In [40]:
```
data.sample(5)
```

Out[40]:

| | country_code | preffix | country_name |
|---|---|---|---|
| **174** | KM | 269 | Comoros |
| **135** | VU | 678 | Vanuatu |
| **71** | IQ | 964 | Iraq |
| **110** | MT | 356 | Malta |
| **239** | AU | 61 | Australia |

A fine tunning: set preffix as index, so queries will be easier later.

In [41]:
```
data.set_index('preffix', inplace=True)
```

In [42]:
```
data.sample(5)
```

Out[42]:

| preffix | country_code | country_name |
|---|---|---|
| **687** | NC | New Caledonia |
| **355** | AL | Albania |
| **683** | NU | Niue |
| **686** | KI | Kiribati |
| **232** | SL | Sierra Leone |

## We're reaching the end of the process

Now let's get the list of country names for the preffixes of the whatsapp numbers

Do you remember the name of the variable where we stored the whatsapp preffixes? No? Me neither.
Let's use a trick to list all the variables we have declared so far: `dir()`

`dir()` returns also reserved variables (which start with underscore), so I will filter them and get only the ones we ourselves defined.

In [43]:
```
variables = [i for i in dir() if not i.startswith('_')]
variables
```

Out[43]:
```
['In',
 'Out',
 'contacts',
 'country_codes',
 'country_codes_count',
 'country_codes_df',
 'country_codes_series',
 'country_names',
 'country_names_df',
 'data',
 'exit',
 'file',
 'get_ipython',
 'json',
 'open',
 'pd',
 'phone_codes',
 'phone_numbers',
 'quit',
 'variables']
```

That's it. It was called `country_codes_count`

In [44]:
```
country_codes_count
```

Out[44]:
```
+234    122
+233     14
+91      13
+254      7
+256      5
+44       4
+27       3
+263      3
+62       2
+60       1
+54       1
+55       1
+92       1
+1        1
+90       1
+255      1
+48       1
+7        1
+261      1
dtype: int64
```

## Disclaimer: I'm a bit busy and don't have time to deal with some issues with country codes: 1 and 7, so I will drop them. Sorry if your number belongs to these countries.

In [45]:
```
country_codes_count.drop(['+1', '+7'], axis=0, inplace=True)  # by defaul
```

In [46]:
```
# make prettier the count series
country_codes_count = (country_codes_count.to_frame()
     .reset_index()
     .rename(columns = {'index' : 'preffix', 0 : 'count'})
)
```

In [47]:
```
country_codes_count
```

Out[47]:

| | preffix | count |
|---|---|---|
| 0 | +234 | 122 |
| 1 | +233 | 14 |
| 2 | +91 | 13 |
| 3 | +254 | 7 |
| 4 | +256 | 5 |
| 5 | +44 | 4 |
| 6 | +27 | 3 |
| 7 | +263 | 3 |
| 8 | +62 | 2 |
| 9 | +60 | 1 |
| 10 | +54 | 1 |
| 11 | +55 | 1 |
| 12 | +92 | 1 |
| 13 | +90 | 1 |
| 14 | +255 | 1 |
| 15 | +48 | 1 |
| 16 | +261 | 1 |

## Let's make the query

In [50]:
```
data.index
```

Out[50]:
```
Index(['880', '32', '226', '359', '387', '+1-246', '681', '590', '+1-441
',
       '673',
       ...
       '43', '297', '91', '+358-18', '994', '353', '62', '380', '974', '2
58'],
      dtype='object', name='preffix', length=250)
```

## The code below throws an error because `country_codes_count.preffix` has '+' in the their values. Let's fix it

In [51]:
```
#query = data.loc[country_codes_count.preffix]
#query
```

In [56]:
```
country_codes_count.preffix = country_codes_count.preffix.apply(lambda x
```

## Let's try again after the fix

In [58]:
```
query = data.loc[country_codes_count.preffix]
query
```

Out[58]:

| preffix | country_code | country_name |
|---|---|---|
| **234** | NG | Nigeria |
| **233** | GH | Ghana |
| **91** | IN | India |
| **254** | KE | Kenya |
| **256** | UG | Uganda |
| **44** | GB | United Kingdom |
| **27** | ZA | South Africa |
| **263** | ZW | Zimbabwe |
| **62** | ID | Indonesia |
| **60** | MY | Malaysia |
| **54** | AR | Argentina |
| **55** | BR | Brazil |
| **92** | PK | Pakistan |
| **90** | TR | Turkey |
| **255** | TZ | Tanzania |
| **48** | PL | Poland |
| **261** | MG | Madagascar |

## It set the index of the query as a column, so I can merge on the preffix column later

In [59]:
```
query = query.reset_index().rename(columns = {'index' : 'preffix'})
```

## Finally merge both the query result and the preffixes count

In [61]:
```
whatsapp_phones_study = pd.merge(query, country_codes_count, on='preffix'
```

In [62]:
```
whatsapp_phones_study
```

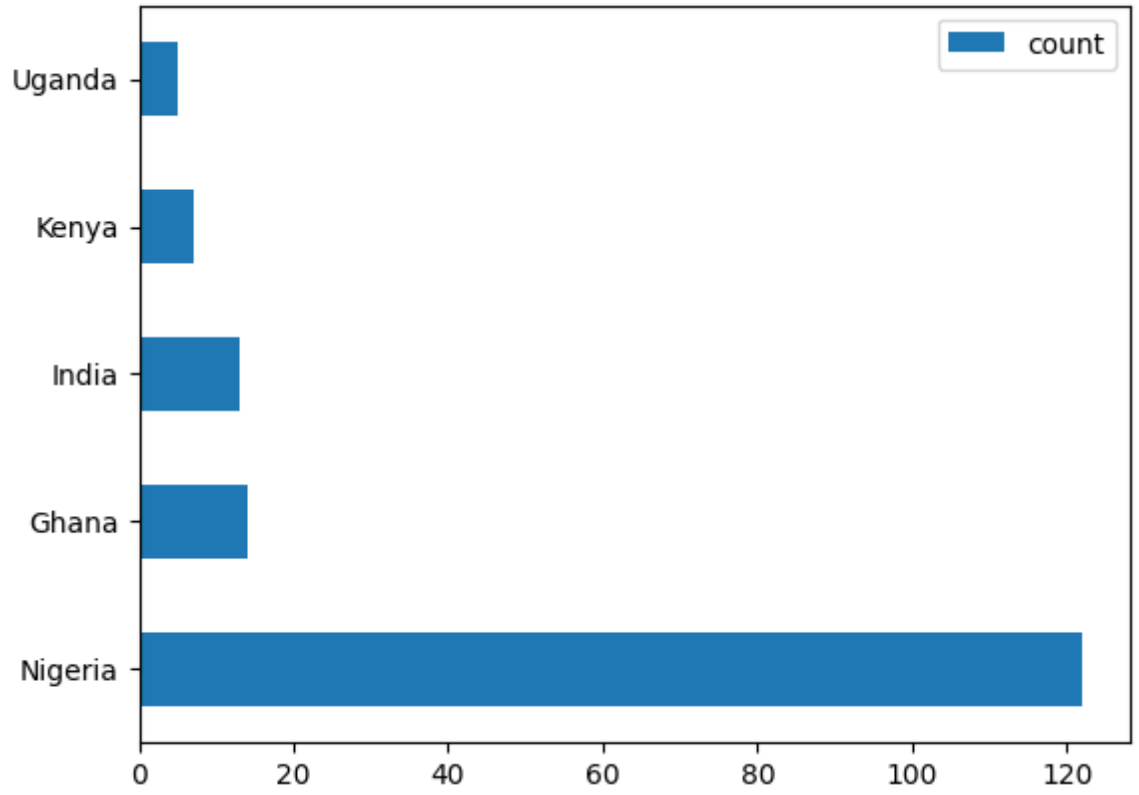
Out[62]:

| | preffix | country_code | country_name | count |
|---|---|---|---|---|
| **0** | 234 | NG | Nigeria | 122 |
| **1** | 233 | GH | Ghana | 14 |
| **2** | 91 | IN | India | 13 |
| **3** | 254 | KE | Kenya | 7 |
| **4** | 256 | UG | Uganda | 5 |
| **5** | 44 | GB | United Kingdom | 4 |
| **6** | 27 | ZA | South Africa | 3 |
| **7** | 263 | ZW | Zimbabwe | 3 |
| **8** | 62 | ID | Indonesia | 2 |
| **9** | 60 | MY | Malaysia | 1 |
| **10** | 54 | AR | Argentina | 1 |
| **11** | 55 | BR | Brazil | 1 |
| **12** | 92 | PK | Pakistan | 1 |
| **13** | 90 | TR | Turkey | 1 |
| **14** | 255 | TZ | Tanzania | 1 |
| **15** | 48 | PL | Poland | 1 |
| **16** | 261 | MG | Madagascar | 1 |

# Finally some graphs for top 5

In [91]:
```
top5 = whatsapp_phones_study.nlargest(5, 'count')
top5.index = top5.country_name.values
```

In [92]:
```
top5.plot.barh(y='count')
plt.show()
```




In [94]:
```
(top5.plot.pie(y='count', x='country_name' , ylabel='Number of participan
               legend=True, shadow=True, autopct='%1.1f%%', explode=[0.08
)
plt.show()
```