In [115]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# pd.set_option('display.max_columns',100)
# pd.set_option('display.max_rows',100)
sns.set(rc={'figure.figsize':(10,8)})
```

In [116]:

```
df = pd.read_csv("Data.csv",index_col=0)
```

In [117]:

```
df.columns
```

Out[117]:

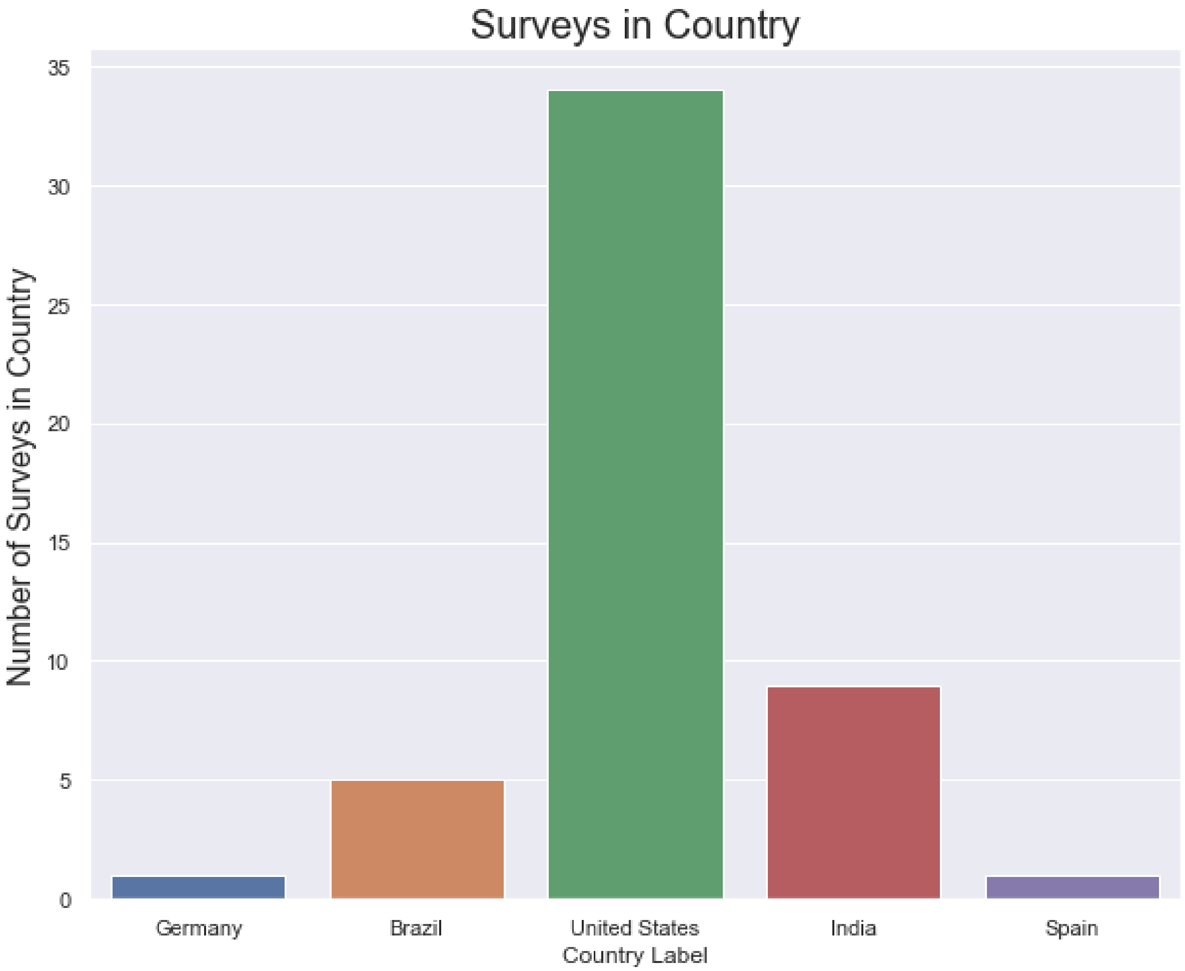
```
Index(['Survey Id', 'Keyword Id', 'Keyword Content', 'Country Of Origin Id',
       'Country Of Origin Label', 'Country Id', 'Country Label', 'Province I
d',
       'Province Label', 'Age Group Id', 'Age Group Label', 'Gender Label',
       'Share Device', 'Mobile Type', 'Degree', 'Rank Pandemic',
       'Source Information Id', 'Source Information Label', 'Result Item Ran
k',
       'Result Item Title', 'Result Item Metadesc', 'Result Item Created At',
       'Result Item Full Url', 'Result Item Full Domain',
       'Google Tracked Country', 'Google Tracked Address',
       'Result Item Openrank', 'Keyword Openrank Average',
       'Survey Openrank Average', 'Html File'],
      dtype='object')
```

Visuaization

Country

In [121]:

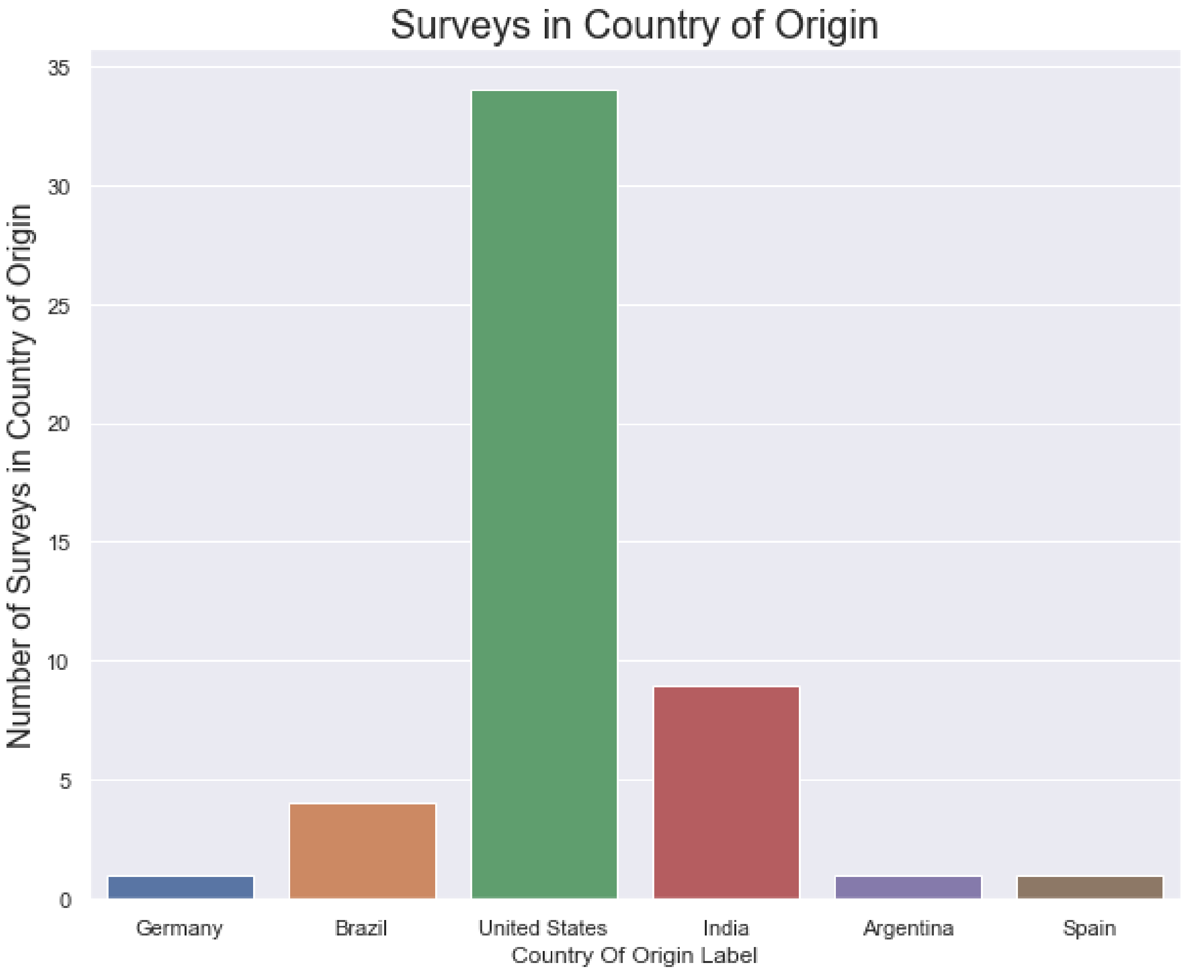
```
df1 = df.groupby("Survey Id")[["Country Label"]].agg("max")
sns.countplot(df1["Country Label"])
plt.title("Surveys in Country",fontsize=20)
plt.ylabel("Number of Surveys in Country",fontsize=16)
plt.show()
```



Country Of Origin

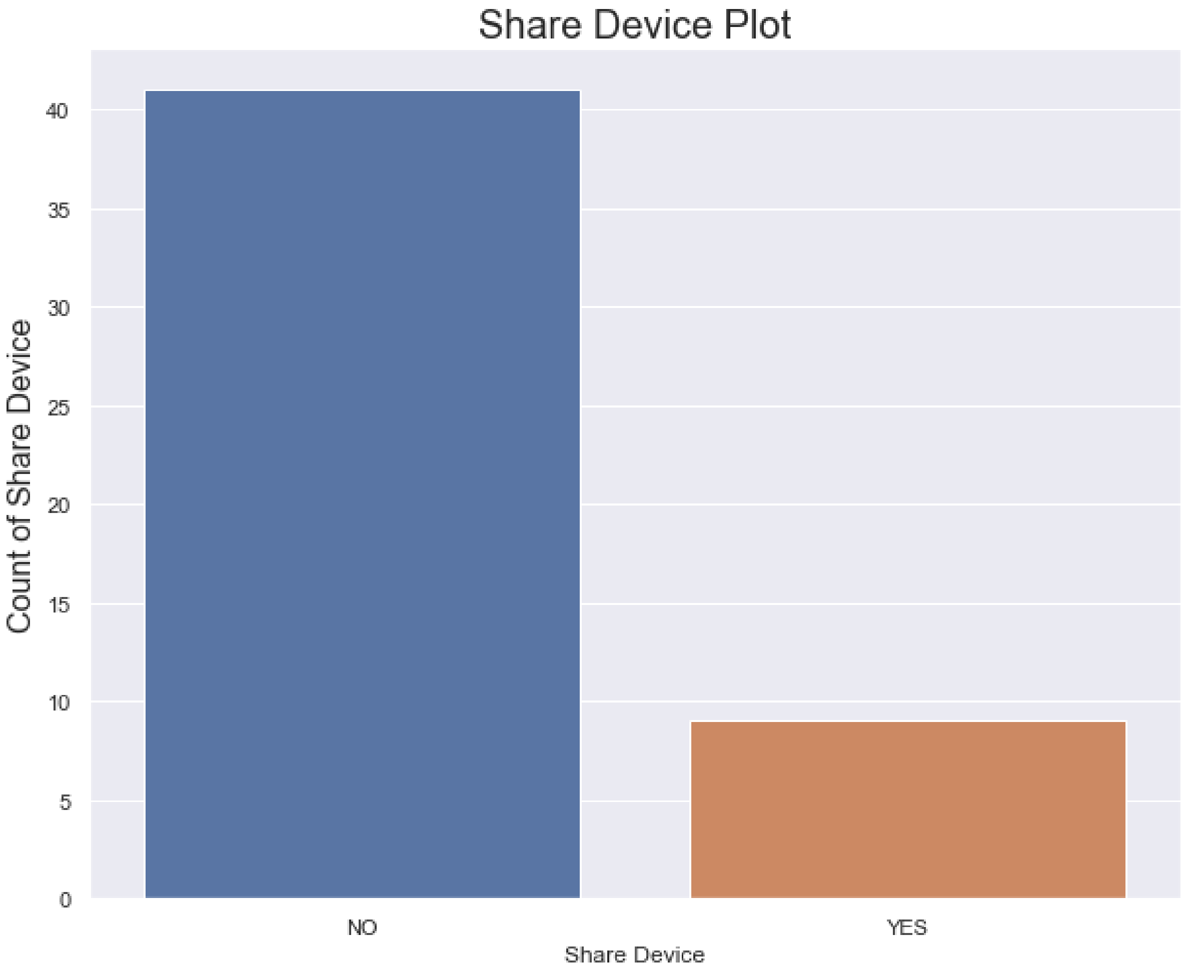
In [122]:

```
df2 = df.groupby("Survey Id")[["Country Of Origin Label"]].agg("max")
sns.countplot(df2["Country Of Origin Label"])
plt.title("Surveys in Country of Origin",fontsize=20)
plt.ylabel("Number of Surveys in Country of Origin",fontsize=16)
plt.show()
```



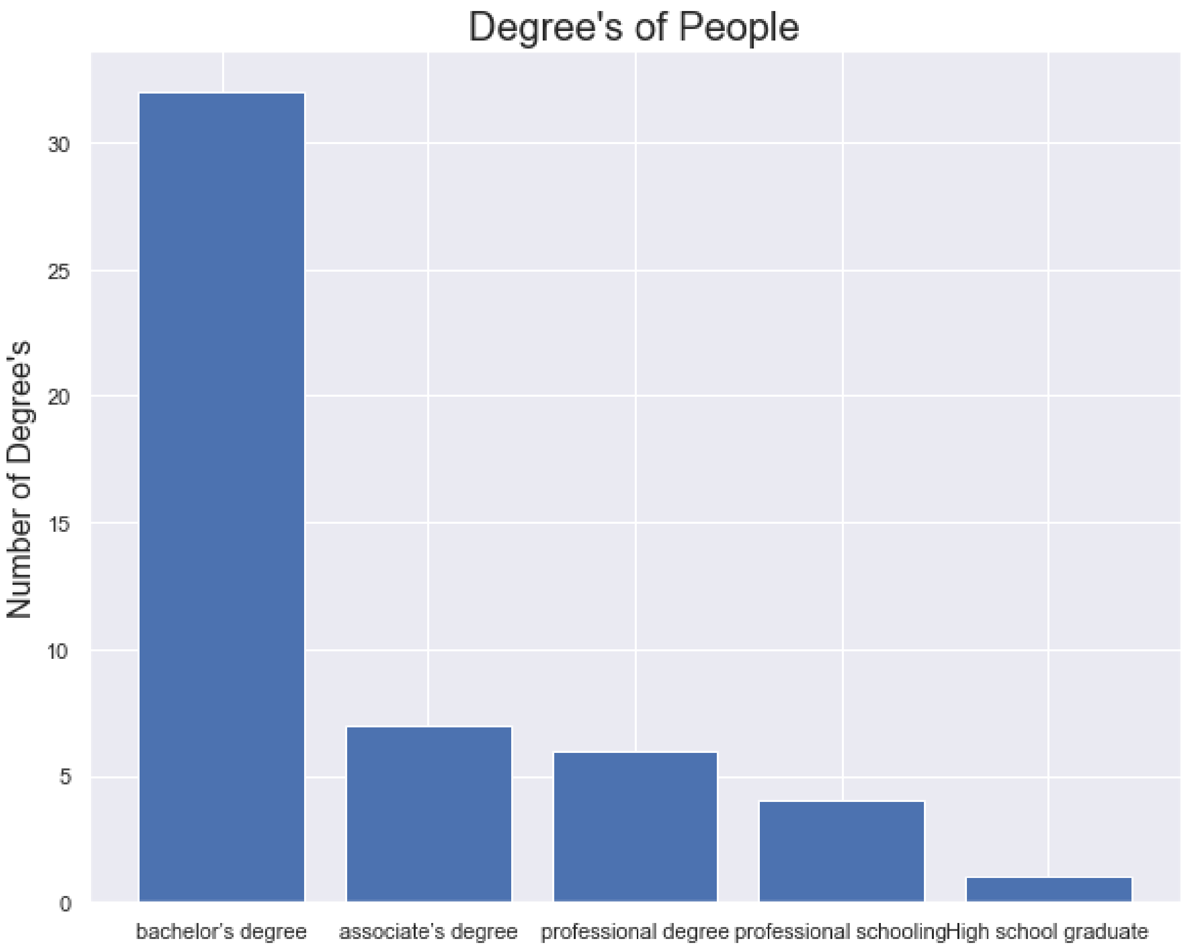
Share Device

```
In [123]: df3 = df.groupby("Survey Id")[["Share Device"]].agg("max")
          sns.countplot(df3["Share Device"])
          plt.title("Share Device Plot",fontsize=20)
          plt.ylabel("Count of Share Device",fontsize=16)
          plt.show()
```



Degree

```
In [127]: df4 = df.groupby("Survey Id")[["Degree"]].agg("max")
          s = df4["Degree"].value_counts()
          s.index = ['bachelor’s degree',"associate’s degree","professional degree","professional schooling","High school graduate"]
          plt.bar(s.index,s.values)
          plt.title("Degree's of People",fontsize=20)
          plt.ylabel("Number of Degree's",fontsize=16)
          plt.show()
```



Data Analysis

Is negative keywords has higher openrank average than positive keyword?

In [128]:

```
df["Keyword Content"].value_counts()
pos = ["should i get tested for covid",
      "should i get flu shot",
      "should i get vaccinated",
      "should i wear facemask",
      "is hydroxychloroquine effective for covid"]

neg = ["should i not get tested for covid",
      "should i not get flu shot",
      "should i avoid get vaccinated",
      "should i not wear facemask",
      "is hydroxychloroquine ineffective for covid"]

df2 = df.copy()[["Survey Id","Keyword Content","Keyword Openrank Average"]]
df2["Pos Keyword"] = df2["Keyword Content"].apply(lambda x: "Yes" if (x in pos
) else 'No')
df2 = df2.groupby(["Survey Id","Pos Keyword"]).agg('mean')
df2 = df2.reset_index(level=["Pos Keyword"])
```

In [129]:

```
pos_mean = df2[df2["Pos Keyword"] == "Yes"]["Keyword Openrank Average"].mean()
neg_mean = df2[df2["Pos Keyword"] == "No"]["Keyword Openrank Average"].mean()
```

In [130]:

```
print(neg_mean,pos_mean)
```

```
71.24749659863946 72.59403501696465
```

The Keyword Openrank Average of Negative keywords for all surveys is lower than Positive keywords.

Computing RBO value for each survey

In [134]:

```
import rbo
pairs = [["should i get tested for covid","should i not get tested for covid"
],
         ["should i get flu shot","should i not get flu shot"],
         ["should i get vaccinated","should i avoid get vaccinated"],
         ["should i wear facemask","should i not wear facemask"],
         ["is hydroxychloroquine effective for covid","is hydroxychloroquine i
neffective for covid"]]
dff = df.copy()
dff['rbo'] = None
```

In [135]:

```
for i in dff.groupby("Survey Id"):
    dumy = i[1]
    for p in pairs:
        d1 = dumy[dumy["Keyword Content"] == p[0]]
        d2 = dumy[dumy["Keyword Content"] == p[1]]
        l1 = d1["Result Item Full Url"].to_list()
        l2 = d2["Result Item Full Url"].to_list()
        res = rbo.RankingSimilarity(l1, l2).rbo()
        dff.loc[((dff["Survey Id"] == i[0])&(dff["Keyword Content"] == p[0])),
'rbo'] = res
        dff.loc[((dff["Survey Id"] == i[0])&(dff["Keyword Content"] == p[1])),
'rbo'] = res
```

In [136]:

```
dff['rbo'] = dff['rbo'].astype('float')
df3 = dff.groupby(["Survey Id","Keyword Content"])[["rbo"]].agg("mean")
df3 = df3.reset_index("Keyword Content")
df3 = df3.groupby(level=0)[['rbo']].agg('mean')
df3
```

Out[136]:

	rbo
Survey Id	
4	0.189158
5	0.200712
6	0.288238
10	0.230556
16	0.280199
17	0.305014
18	0.260459
19	0.291340
20	0.272690
22	0.234259
24	0.285833
26	0.305014
29	0.281368
30	0.253933
31	0.217687
36	0.261778
38	0.283547
40	0.265121
41	0.266319
42	0.269288
45	0.188093
47	0.280728
48	0.336808
50	0.278753
53	0.302699
55	0.190562
62	0.249912
67	0.271113
71	0.244356
74	0.280019
81	0.190562
84	0.248490
88	0.261778
89	0.190562

	rbo
Survey Id	
94	0.270292
95	0.282272
103	0.221246
105	0.283198
106	0.277284
119	0.258951
123	0.234259
130	0.316876
135	0.294343
139	0.266537
140	0.277741
141	0.274977
145	0.310660
146	0.271247
147	0.249924
148	0.317377

Above dataframe shows the RBO of each survey by averaging the calculated RBO values for each pair of positive and negative keyword.

In [137]:

```
rbo_dict = df3.to_dict()['rbo']
dff['rbo'] = dff["Survey Id"].apply(lambda x: rbo_dict[x])
```

Average RBO of people using Internet as Source of Information vs Other

In [138]:

```
df3 = dff.groupby("Survey Id")[["Source Information Label"]].agg('max')
people_internet = df3[df3["Source Information Label"] == 'Internet'].index
people_other = df3[df3["Source Information Label"] != 'Internet'].index
```

In [139]:

```
internet_mean = dff.set_index("Survey Id").loc[people_internet].groupby(level=
0)['rbo'].agg('mean').mean()
other_mean = dff.set_index("Survey Id").loc[people_other].groupby(level=0)['rb
o'].agg('mean').mean()
print("Internet Source Average :",internet_mean)
print("Other Sources Average :",other_mean)
```

```
Internet Source Average : 0.2671466642924976
Other Sources Average : 0.25909656084656074
```

Average RBO of people from 24 to 34 vs Other

In [140]:

```
df4 = dff.groupby("Survey Id")[["Age Group Label"]].agg('max')
age_2434 = df4[df4["Age Group Label"] == '25-34'].index
age_other = df4[df4["Age Group Label"] != '25-34'].index

age_2434_average = dff.set_index("Survey Id").loc[age_2434].groupby(level=0)[
'rbo'].agg('mean').mean()
age_other_average = dff.set_index("Survey Id").loc[age_other].groupby(level=0)
['rbo'].agg('mean').mean()
print("People with Age Between 24 to 34:",age_2434_average)
print("Other Age Groups :",age_other_average)
```

```
People with Age Between 24 to 34: 0.26134402177218263
Other Age Groups : 0.26595971907281435
```

Average RBO of people in United States vs Other Countries

In [141]:

```
df5 = dff.groupby("Survey Id")[["Country Label"]].agg('max')
us_people = df5[df5["Country Label"] == 'United States'].index
other_people = df5[df5["Country Label"] != 'United States'].index

us_average = dff.set_index("Survey Id").loc[us_people].groupby(level=0)['rbo']
.agg('mean').mean()
other_average = dff.set_index("Survey Id").loc[other_people].groupby(level=0)[
'rbo'].agg('mean').mean()
print("Average RBO of United States :",us_average)
print("Average RBO of Other Countries :",other_average)
```

```
Average RBO of United States : 0.2747509012864404
Average RBO of Other Countries : 0.2389125055114638
```

Average RBO of people with Bachelor's Degree vs Associate's Degree

In [142]:

```
df6 = dff.groupby("Survey Id")[["Degree"]].agg('max')
bachelor = df6[df6["Degree"] == 'College degree/bachelor’s degree'].index
associate = df6[df6["Degree"] == 'Some college (some community college, associ
ate’s degree)'].index

bachelor_average = dff.set_index("Survey Id").loc[bachelor].groupby(level=0)[
'rbo'].agg('mean').mean()
associate_average = dff.set_index("Survey Id").loc[associate].groupby(level=0)
['rbo'].agg('mean').mean()
print("Bachlors Degree holders RBO :",bachelor_average)
print("Associates Degree holders RBO :",associate_average)
```

```
Bachlors Degree holders RBO : 0.26594408619929444
Associates Degree holders RBO : 0.260319570420761
```

Average RBO of Male and Female

In [143]:

```
df7 = dff.groupby("Survey Id")[["Gender Label"]].agg('max')
male = df7[df7["Gender Label"] == 'male'].index
female = df7[df7["Gender Label"] == 'female'].index

male_average = dff.set_index("Survey Id").loc[male].groupby(level=0)['rbo'].agg('mean').mean()
female_average = dff.set_index("Survey Id").loc[female].groupby(level=0)['rbo'].agg('mean').mean()
print("Male Average RBO :",male_average)
print("Female Average RBO :",female_average)
```

```
Male Average RBO : 0.27426763668430326
Female Average RBO : 0.25281735008818346
```

Counting the number of times .gov appeared in Positive and Negative Keywords

In [144]:

```
pos = ["should i get tested for covid",
      "should i get flu shot",
      "should i get vaccinated",
      "should i wear facemask",
      "is hydroxychloroquine effective for covid"]

dff["gov domain"] = dff["Result Item Full Domain"].apply(lambda x: 1 if ".gov" in x else 0)
df8 = dff.copy()
df8["Keyword Type"] = df8["Keyword Content"].apply(lambda x: 'pos' if x in pos else 'neg')
df8 = df8.groupby(["Survey Id","Keyword Type"],as_index=False)[["gov domain"]].agg("sum")
df8.groupby("Keyword Type")[['gov domain']].agg('sum')
```

Out[144]:

	gov domain
Keyword Type	
neg	699
pos	707

Above Dataframe shows the number of times government website appeared in positive and negative keyword.

Counting .gov websites for people of different countries

In [146]:

```
df9 = dff.groupby(["Survey Id","Country Label"],as_index=False)[["gov domain"
]].agg('sum')
df9 = df9.groupby("Country Label")[["gov domain"]].agg(['sum','count'])
df9
```

Out[146]:

	gov domain	
	sum	count
Country Label		
Brazil	109	5
Germany	25	1
India	150	9
Spain	27	1
United States	1095	34

Above Dataframe shows the total number of times government website appreared in that country along with the total number of times the surveys taken in that country.

In [147]:

```
df9["Percentage of Gov Website"] = df9["gov domain"]['sum'] / df9["gov domain"
]['count']
df9
```

Out[147]:

	gov domain		Percentage of Gov Website
	sum	count	
Country Label			
Brazil	109	5	21.800000
Germany	25	1	25.000000
India	150	9	16.666667
Spain	27	1	27.000000
United States	1095	34	32.205882

Above Dataframe shows the percentage of Government websites appeared in results of each country.

In [ ]: