In [1]:
```
import pandas as pd
import numpy as np
import matplotlib. pyplot as plt
```

In [2]:
```
data = {'year': [2010, 2011, 2012, 2010, 2011, 2012, 2010, 2011, 2012]
        'team': ['FCBarcelona', 'FCBarcelona', 'FCBarcelona', 'RMadrid
                 'ValenciaCF', 'ValenciaCF'],
        'wins':   [30, 28, 32, 29, 32, 26, 21, 17, 19],
        'draws':  [6, 7, 4, 5, 4, 7, 8, 10, 8],
        'losses': [2, 3, 2, 4, 2, 5, 9, 11, 11]
        }
football = pd.DataFrame(
    data, columns=['year', 'team', 'wins', 'draws', 'losses'])
football
```

Out[2]:

	year	team	wins	draws	losses
0	2010	FCBarcelona	30	6	2
1	2011	FCBarcelona	28	7	3
2	2012	FCBarcelona	32	4	2
3	2010	RMadrid	29	5	4
4	2011	RMadrid	32	4	2
5	2012	RMadrid	26	7	5
6	2010	ValenciaCF	21	8	9
7	2011	ValenciaCF	17	10	11
8	2012	ValenciaCF	19	8	11

In [3]:
```
edu = pd.read_csv('educ_figdp_1_Data.csv',
                  na_values=':', usecols=['TIME', 'GEO', 'Value'])
edu
```

Out[3]:

	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95
...	...	...	...
379	2007	Finland	5.90
380	2008	Finland	6.10
381	2009	Finland	6.81
382	2010	Finland	6.85
383	2011	Finland	6.76

384 rows × 3 columns

In [4]:
```
edu.head()
```

Out[4]:

	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95

In [5]:
```
edu.tail()
```

Out[5]:

	TIME	GEO	Value
379	2007	Finland	5.90
380	2008	Finland	6.10
381	2009	Finland	6.81
382	2010	Finland	6.85
383	2011	Finland	6.76

In [6]:
```
edu.columns
```

Out[6]:
```
Index(['TIME', 'GEO', 'Value'], dtype='object')
```

In [7]:
```
edu.index
```

Out[7]:
```
RangeIndex(start=0, stop=384, step=1)
```

In [8]:
```
edu.values
```

Out[8]:
```
array([[2000, 'European Union (28 countries)', nan],
       [2001, 'European Union (28 countries)', nan],
       [2002, 'European Union (28 countries)', 5.0],
       ...,
       [2009, 'Finland', 6.81],
       [2010, 'Finland', 6.85],
       [2011, 'Finland', 6.76]], dtype=object)
```

In [9]:
```
edu.describe()
```

Out[9]:

	TIME	Value
count	384.000000	361.000000
mean	2005.500000	5.203989
std	3.456556	1.021694
min	2000.000000	2.880000
25%	2002.750000	4.620000
50%	2005.500000	5.060000
75%	2008.250000	5.660000
max	2011.000000	8.810000

In [10]:
```
edu['Value']
```

Out[10]:
```
0       NaN
1       NaN
2      5.00
3      5.03
4      4.95
       ...
379    5.90
380    6.10
381    6.81
382    6.85
383    6.76
Name: Value, Length: 384, dtype: float64
```

In [11]:
```
edu[10:14]
```

Out[11]:

	TIME	GEO	Value
10	2010	European Union (28 countries)	5.41
11	2011	European Union (28 countries)	5.25
12	2000	European Union (27 countries)	4.91
13	2001	European Union (27 countries)	4.99

In [12]:
```
edu.iloc[90:94,:]
```

Out[12]:

	TIME	GEO	Value
90	2006	Belgium	5.98
91	2007	Belgium	6.00
92	2008	Belgium	6.43
93	2009	Belgium	6.57

In [13]:
```
edu.sample(10,random_state=23)
```

Out[13]:

	TIME	GEO	Value
294	2006	Netherlands	5.50
343	2007	Romania	4.25
178	2010	Greece	NaN
73	2001	Euro area (13 countries)	4.97
284	2008	Malta	5.72
193	2001	France	5.95
236	2008	Latvia	5.71
205	2001	Italy	4.83
59	2011	Euro area (17 countries)	5.15
252	2000	Luxembourg	NaN

In [14]:
```
edu.sample(10,random_state=23).loc[73:59,:]
```

Out[14]:

	TIME	GEO	Value
73	2001	Euro area (13 countries)	4.97
284	2008	Malta	5.72
193	2001	France	5.95
236	2008	Latvia	5.71
205	2001	Italy	4.83
59	2011	Euro area (17 countries)	5.15

In [15]:
```
edu[edu['Value'] > 6.5].tail()
```

Out[15]:

	TIME	GEO	Value
286	2010	Malta	6.74
287	2011	Malta	7.96
381	2009	Finland	6.81
382	2010	Finland	6.85
383	2011	Finland	6.76

In [16]:
```
edu[edu['Value'].isnull()].head()
```

Out[16]:

	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
36	2000	Euro area (18 countries)	NaN
37	2001	Euro area (18 countries)	NaN
48	2000	Euro area (17 countries)	NaN

In [17]:
```
edu.max(axis=0)
```

Out[17]:
```
TIME       2011
GEO       Spain
Value      8.81
dtype: object
```

In [18]:
```
print('Pandas max function:', edu['Value'].max())
print('Python max function:', max(edu['Value']))
```

```
Pandas max function: 8.81
Python max function: nan
```

In [19]:
```
s = edu['Value'] / 100
s.head()
```

Out[19]:
```
0       NaN
1       NaN
2    0.0500
3    0.0503
4    0.0495
Name: Value, dtype: float64
```

In [20]:
```
s = edu['Value'].apply(np.sqrt)
s.head()
```

Out[20]:
```
0         NaN
1         NaN
2    2.236068
3    2.242766
4    2.224860
Name: Value, dtype: float64
```

In [21]:
```
s = edu['Value'].apply(lambda d: d**2)
s.head()
```

Out[21]:
```
0        NaN
1        NaN
2    25.0000
3    25.3009
4    24.5025
Name: Value, dtype: float64
```

In [22]:
```
edu['ValueNorm'] = edu['Value'] / edu['Value'].max()
edu.tail()
```

Out[22]:

	TIME	GEO	Value	ValueNorm
379	2007	Finland	5.90	0.669694
380	2008	Finland	6.10	0.692395
381	2009	Finland	6.81	0.772985
382	2010	Finland	6.85	0.777526
383	2011	Finland	6.76	0.767310

In [23]:
```
edu.drop('ValueNorm', axis=1, inplace=True)
edu.head()
```

Out[23]:

	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95

In [51]:
```
edu = edu.append({"TIME": 2000,"Value": 5.00,"GEO": 'a'} ,
                 ignore_index = True)
edu.tail()
```

```
---------------------------------------------------------------------------
------
AttributeError                            Traceback (most recent call
last)
~\AppData\Local\Temp\ipykernel_2484\2827960700.py in ?()
----> 1 edu = edu.append({"TIME": 2000,"Value": 5.00,"GEO": 'a'} ,
      2                  ignore_index = True)
      3 edu.tail()

~\anaconda3\Lib\site-packages\pandas\core\generic.py in ?(self, name)
   5985             and name not in self._accessors
   5986             and self._info_axis._can_hold_identifiers_and_hol
ds_name(name)
   5987         ):
   5988             return self[name]
-> 5989         return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'append'
```

In [25]:
```
edu.drop(max(edu.index), axis=0, inplace=True)
edu.tail()
```

Out[25]:

	TIME	GEO	Value
378	2006	Finland	6.18
379	2007	Finland	5.90
380	2008	Finland	6.10
381	2009	Finland	6.81
382	2010	Finland	6.85

In [26]:
```
eduDrop = edu.dropna(how='any', subset=['Value'], axis=0)
eduDrop.head()
```

Out[26]:

	TIME	GEO	Value
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95
5	2005	European Union (28 countries)	4.92
6	2006	European Union (28 countries)	4.91

In [27]:
```
eduFilled = edu.fillna(value={'Value': 0})
eduFilled.head()
```

Out[27]:

	TIME	GEO	Value
0	2000	European Union (28 countries)	0.00
1	2001	European Union (28 countries)	0.00
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95

In [28]:
```
edu.sort_values(by='Value', ascending=False, inplace=True)
edu.head()
```

Out[28]:

	TIME	GEO	Value
130	2010	Denmark	8.81
131	2011	Denmark	8.75
129	2009	Denmark	8.74
121	2001	Denmark	8.44
122	2002	Denmark	8.44

In [29]:
```
edu.sort_index(axis=0, ascending=True, inplace=True)
edu.head()
```

Out[29]:

	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95

In [30]:
```
group = edu[['GEO', 'Value']].groupby('GEO').mean()
group.head()
```

Out[30]:

GEO	Value
Austria	5.618333
Belgium	6.189091
Bulgaria	4.093333
Cyprus	7.023333
Czech Republic	4.168333

In [31]:
```
filtered_data = edu[edu['TIME'] > 2005]
pivedu = pd.pivot_table(filtered_data, values='Value',
                        index=['GEO'], columns=['TIME'])
pivedu.head()
```

Out[31]:

GEO	2006	2007	2008	2009	2010	2011
Austria	5.40	5.33	5.47	5.98	5.91	5.80
Belgium	5.98	6.00	6.43	6.57	6.58	6.55
Bulgaria	4.04	3.88	4.44	4.58	4.10	3.82
Cyprus	7.02	6.95	7.45	7.98	7.92	7.87
Czech Republic	4.42	4.05	3.92	4.36	4.25	4.51

In [32]:
```
pivedu.loc[['Spain', 'Portugal'], [2006, 2011]]
```

Out[32]:

GEO	2006	2011
Spain	4.26	4.82
Portugal	5.07	5.27

In [33]:
```
pivedu = pivedu.drop(['Euro area (13 countries)',
                      'Euro area (15 countries)',
                      'Euro area (17 countries)',
                      'Euro area (18 countries)',
                      'European Union (25 countries)',
                      'European Union (27 countries)',
                      'European Union (28 countries)'
                      ], axis=0)
pivedu = pivedu.rename(
    index={'Germany (until 1990 former territory of the FRG)': 'German
pivedu = pivedu.dropna()
pivedu.rank(ascending=False, method='first').head()
```

Out[33]:

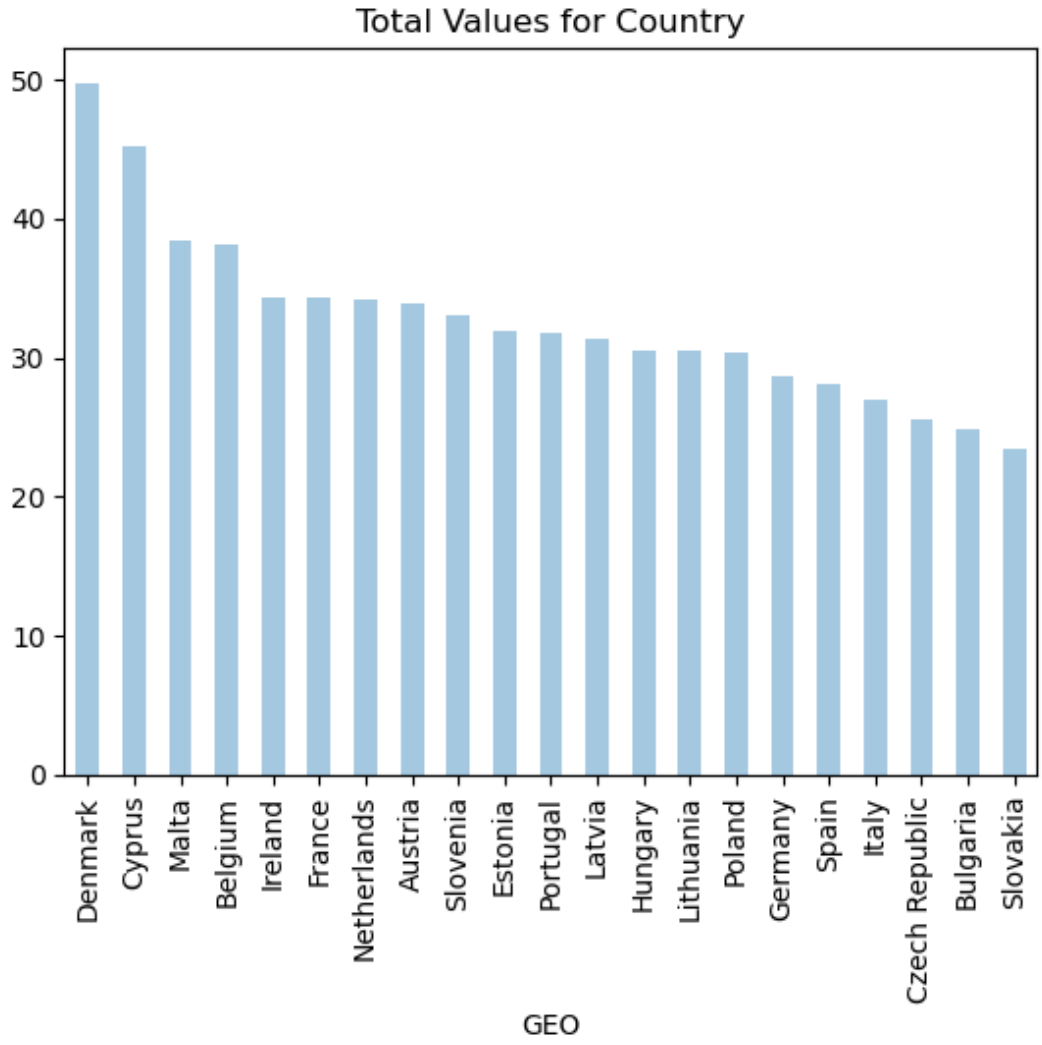
TIME	2006	2007	2008	2009	2010	2011
GEO						
Austria	9.0	6.0	10.0	6.0	7.0	7.0
Belgium	4.0	4.0	3.0	3.0	4.0	4.0
Bulgaria	20.0	20.0	19.0	19.0	21.0	21.0
Cyprus	2.0	2.0	2.0	2.0	2.0	3.0
Czech Republic	18.0	19.0	20.0	20.0	19.0	18.0

In [34]:
```
totalSum = pivedu.sum(axis=1)
totalSum.rank(ascending=False, method='dense').sort_values().head()
```

Out[34]:
```
GEO
Denmark    1.0
Cyprus     2.0
Malta      3.0
Belgium    4.0
Ireland    5.0
dtype: float64
```

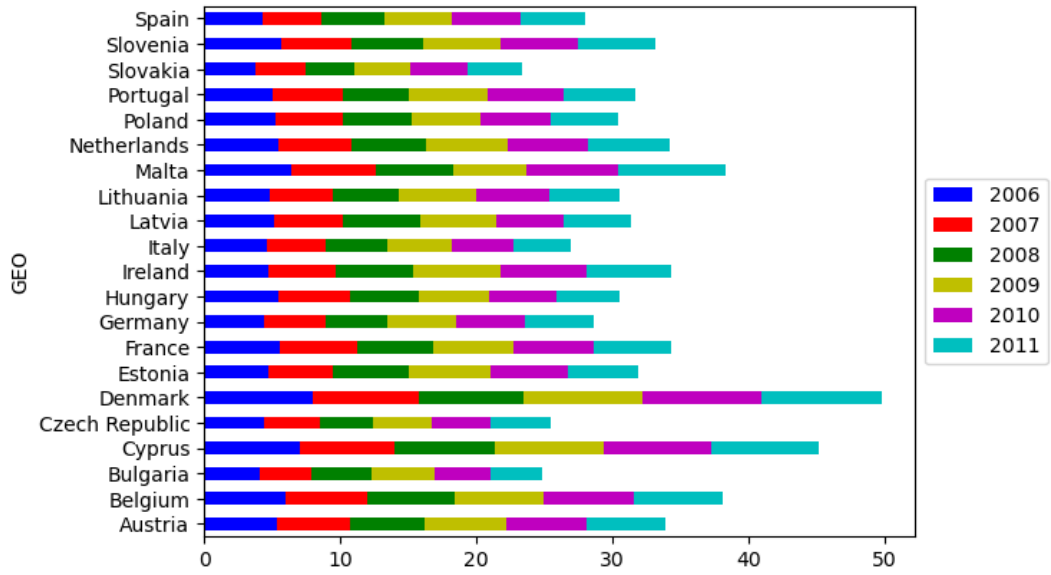
In [42]:
```
totalSum = pivedu. sum(axis = 1).sort_values(ascending = False)
totalSum. plot(kind = 'bar', style = 'b', alpha = 0.4,
title = "Total Values for Country")
```

Out[42]: <Axes: title={'center': 'Total Values for Country'}, xlabel='GEO'>



In [45]:
```
my_colors = ['b', 'r', 'g', 'y', 'm', 'c']
ax = pivedu. plot(kind = 'barh',
stacked = True ,
color = my_colors)
ax.legend(loc = 'center left', bbox_to_anchor = (1, .5))
```

Out[45]: <matplotlib.legend.Legend at 0x1b5027fb650>



In [ ]:
```
```