10 Countries:

1. China
2. Japan
3. Sweden
4. Russia
5. Argentina
6. Ukraine
7. Netherlands
8. Spain
9. Denmark
10. Germany

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
```

In [2]:
```
#title = % Population Over 65
O65P = pd.DataFrame()
O65P['Country Name'] = ['China', 'Japan', 'Sweden', 'Russia', 'Argentin
a', 'Ukraine', 'Netherlands', 'Spain', 'Denmark', 'Germany']
O65P['%'] = [10.6, 27, 19.9, 14.2, 11.2, 16.5, 18.8, 19.4, 19.7, 21.5]
O65P
```

Out[2]:

	Country Name	%
0	China	10.6
1	Japan	27.0
2	Sweden	19.9
3	Russia	14.2
4	Argentina	11.2
5	Ukraine	16.5
6	Netherlands	18.8
7	Spain	19.4
8	Denmark	19.7
9	Germany	21.5

In [3]:
```
Data = pd.read_excel('API_SH.MED.BEDS.ZS_DS2_en_excel_v2_1495658.xls')
```

In [4]:
```
Columns = Data.loc[Data['Data Source'] == 'Country Name'].values
newArr = Columns.reshape(65)
List = newArr.tolist()
List
```

Out[4]:
```
['Country Name',
 'Country Code',
 'Indicator Name',
 'Indicator Code',
 1960.0,
 1961.0,
 1962.0,
 1963.0,
 1964.0,
 1965.0,
 1966.0,
 1967.0,
 1968.0,
 1969.0,
 1970.0,
 1971.0,
 1972.0,
 1973.0,
 1974.0,
 1975.0,
 1976.0,
 1977.0,
 1978.0,
 1979.0,
 1980.0,
 1981.0,
 1982.0,
 1983.0,
 1984.0,
 1985.0,
 1986.0,
 1987.0,
 1988.0,
 1989.0,
 1990.0,
 1991.0,
 1992.0,
 1993.0,
 1994.0,
 1995.0,
 1996.0,
 1997.0,
 1998.0,
 1999.0,
 2000.0,
 2001.0,
 2002.0,
 2003.0,
 2004.0,
 2005.0,
 2006.0,
 2007.0,
 2008.0,
 2009.0,
 2010.0,
 2011.0,
```

```
2012.0,
2013.0,
2014.0,
2015.0,
2016.0,
2017.0,
2018.0,
2019.0,
2020.0]
```

In [5]:
```
Data.columns = List
```

In [6]:
```
Data
```

Out[6]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960.0	1961.0	1962.0	1963.0
0	Last Updated Date	2020-10-13 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Country Name	Country Code	Indicator Name	Indicator Code	1960.000000	1961.0	1962.0	1963.0
3	Aruba	ABW	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN
4	Afghanistan	AFG	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	0.170627	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
262	Kosovo	XKX	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN
263	Yemen, Rep.	YEM	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	0.454545	NaN	NaN	NaN
264	South Africa	ZAF	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN
265	Zambia	ZMB	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	2.801656	NaN	NaN	NaN
266	Zimbabwe	ZWE	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	3.876646	NaN	NaN	NaN

267 rows × 65 columns

In [7]:
```
Data = Data.drop([0,1,2])
Data
```

Out[7]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960.0	1961.0	1962.0	1963.0	1964.
3	Aruba	ABW	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	Nal
4	Afghanistan	AFG	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	0.170627	NaN	NaN	NaN	Nal
5	Angola	AGO	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	2.061462	NaN	NaN	NaN	Nal
6	Albania	ALB	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	5.102676	NaN	NaN	NaN	Nal
7	Andorra	AND	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	Nal
...	...	...	...	...	...	...	...	...	.
262	Kosovo	XKX	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	Nal
263	Yemen, Rep.	YEM	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	0.454545	NaN	NaN	NaN	Nal
264	South Africa	ZAF	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	Nal
265	Zambia	ZMB	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	2.801656	NaN	NaN	NaN	Nal
266	Zimbabwe	ZWE	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	3.876646	NaN	NaN	NaN	Nal

264 rows × 65 columns

In [8]:
```
Data = Data.reset_index(drop = True)
```

In [9]:
```
Data.columns
```

Out[9]:
```
Index([  'Country Name',   'Country Code', 'Indicator Name', 'Indicat
or Code',
                  1960.0,           1961.0,           1962.0,
1963.0,
                  1964.0,           1965.0,           1966.0,
1967.0,
                  1968.0,           1969.0,           1970.0,
1971.0,
                  1972.0,           1973.0,           1974.0,
1975.0,
                  1976.0,           1977.0,           1978.0,
1979.0,
                  1980.0,           1981.0,           1982.0,
1983.0,
                  1984.0,           1985.0,           1986.0,
1987.0,
                  1988.0,           1989.0,           1990.0,
1991.0,
                  1992.0,           1993.0,           1994.0,
1995.0,
                  1996.0,           1997.0,           1998.0,
1999.0,
                  2000.0,           2001.0,           2002.0,
2003.0,
                  2004.0,           2005.0,           2006.0,
2007.0,
                  2008.0,           2009.0,           2010.0,
2011.0,
                  2012.0,           2013.0,           2014.0,
2015.0,
                  2016.0,           2017.0,           2018.0,
2019.0,
                  2020.0],
      dtype='object')
```

In [10]:
```
df = pd.DataFrame()
China = Data.loc[Data['Country Name'] == 'China']
Japan = Data.loc[Data['Country Name'] == 'Japan']
Sweden = Data.loc[Data['Country Name'] == 'Sweden']
Russia = Data.loc[Data['Country Name'] == 'Russian Federation']
Argentina = Data.loc[Data['Country Name'] == 'Argentina']
Ukraine = Data.loc[Data['Country Name'] == 'Ukraine']
Netherlands = Data.loc[Data['Country Name'] == 'Netherlands']
Spain = Data.loc[Data['Country Name'] == 'Spain']
Denmark = Data.loc[Data['Country Name'] == 'Denmark']
Germany = Data.loc[Data['Country Name'] == 'Germany']
df = df.append(China)
df = df.append(Japan)
df = df.append(Sweden)
df = df.append(Russia)
df = df.append(Argentina)
df = df.append(Ukraine)
df = df.append(Netherlands)
df = df.append(Spain)
df = df.append(Denmark)
df = df.append(Germany)
df
```

Out[10]:

	Country Name	Country Code	Indicator Name	Indicator Code	1960.0	1961.0	1962.0	1963.0	196
38	China	CHN	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	N
117	Japan	JPN	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	9.000000	NaN	NaN	NaN	N
221	Sweden	SWE	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	14.200000	NaN	NaN	NaN	N
200	Russian Federation	RUS	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	N
7	Argentina	ARG	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	6.352251	NaN	NaN	NaN	N
246	Ukraine	UKR	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	N
174	Netherlands	NLD	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	N
68	Spain	ESP	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	N
56	Denmark	DNK	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	NaN	NaN	NaN	NaN	N
53	Germany	DEU	Hospital beds (per 1,000 people)	SH.MED.BEDS.ZS	10.500000	NaN	NaN	NaN	N

10 rows × 65 columns

In [11]:
```
beds = pd.DataFrame()
```

In [12]:
```
beds['Country Name'] = df['Country Name']
```

In [13]:
```
noBed = [4.2, 13.4, 2.6, 8.2, 5, 8.8, 4.7, 3, 2.5, 8.3]
```

In [14]:
```
beds['beds1k'] = noBed
beds = beds.reset_index(drop = True)
```

In [15]:
```
case = pd.DataFrame()
case['Country Names'] = beds['Country Name']
response = [0.34, 1.36, 58.26, 17.91, 64.94, 14.71, 41.25, 74.38, 12.1
1, 12.13]
case['Death/100kPop'] = response
case
```

Out[15]:

	Country Names	Death/100kPop
0	China	0.34
1	Japan	1.36
2	Sweden	58.26
3	Russian Federation	17.91
4	Argentina	64.94
5	Ukraine	14.71
6	Netherlands	41.25
7	Spain	74.38
8	Denmark	12.11
9	Germany	12.13

In [16]:
```
features = pd.DataFrame()
features['Country Names'] = O65P['Country Name']
features['beds1k'] = beds['beds1k']
features['%Over65'] = O65P['%']
features['Death/100kPop'] = case['Death/100kPop']
features
```

Out[16]:

	Country Names	beds1k	%Over65	Death/100kPop
0	China	4.2	10.6	0.34
1	Japan	13.4	27.0	1.36
2	Sweden	2.6	19.9	58.26
3	Russia	8.2	14.2	17.91
4	Argentina	5.0	11.2	64.94
5	Ukraine	8.8	16.5	14.71
6	Netherlands	4.7	18.8	41.25
7	Spain	3.0	19.4	74.38
8	Denmark	2.5	19.7	12.11
9	Germany	8.3	21.5	12.13

USA response rate aka Death/100kPop = 68.84

In [17]:
```
target = pd.DataFrame()
target['response'] = features['Death/100kPop']
```

In [18]:
```
X = features[['beds1k', '%Over65']]
y = target['response']
```

In [19]:
```
reg = LinearRegression().fit(X,y)
reg.score(X,y)
```

Out[19]:
```
0.3179821673226729
```

In [20]:
```
reg.coef_
```

Out[20]:
```
array([-4.69042678,  0.58105536])
```

In [21]:
```
reg.intercept_
```

Out[21]:
```
47.82062077488726
```

In [22]:
```
pred = reg.predict(X)
print(pred)
```

```
[34.28001508  0.65739655 47.18851276 17.61010724 30.87630687 16.13227
849
 36.69945562 45.02181437 47.54134436 21.38276867]
```

In [23]:
```
reg.predict(np.array([[5, 11.2]]))
```

Out[23]:
```
array([30.87630687])
```

In [24]:
```
Usa = pd.DataFrame()
Usa['Country Names'] = ['USA']
Usa['beds1k'] = [2.9]
Usa['%Over65'] = [15.4]
Usa['Death/100kPop'] = [68.84]
Usa
```

Out[24]:

	Country Names	beds1k	%Over65	Death/100kPop
0	USA	2.9	15.4	68.84

In [25]:
```
reg.predict(np.array([[2.9, 15.4]]))
```

Out[25]:
```
array([43.16663561])
```