LESSON PLAN

LESSON #9

Aim: Ranking and plotting data part 1

Objective: After a lesson on reading and selection of data, students will be able to use Python to perform the same procedures on two other datasets.

Do Now: Why is data science important? Statistics to make you think! [4 – 8:10 minutes]

https://www.youtube.com/watch?v=Qck5Ae3F3RY&list=RDUASWcDsH_nM&index=20

Standards: 9-12. IC.7 Computational Thinking, Algorithms, and Programming

9-12. CT.2 Computational Thinking, Data Analysis, and Visualization

9-12. CT.3 Computational Thinking, Data Analysis, and Visualization

9-12. DL.5 Digital Literacy, Digital Use

Mini-Lesson:

1. Ranking data
   a. Cleanup previous pivot table entries
   b. Drop the Euro area entries and shorten the Germany name entry by using the rename function
   c. Drop all rows containing the NaN by using dropna function.
   d. Use the rank function by setting ascending to false – highest to lowest
   e. To make global ranking across all years – sum all columns and rank the result

   "dense" – items that compare equals get same ranking and the next not equal item receives the immediately following ranking

2. Plotting Data
   a. Plotting the accumulated values for each country over the last six years by taking the Series obtained in the last example and plot it directly
   b. Plotting a data frame directly – treat columns as a separate Series by plotting the value for each year
3. Conclusion and summary

   Discussion:

   » What are some of the problems or challenges you encountered?

   » How did you resolve them?

   » What did you learn from this lesson?

   » Do you have any lingering questions on today's lesson or data science in general?

CODE AND OUTPUT

Ranking Data

```
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)’: ’Germany’})
pivedu = pivedu.dropna()
pivedu.rank(ascending = False, method = ’first’).head()
```

Output

TIME	2006	2007	2008	2009	2010	2011
GEO						
Austria	10	7	11	7	8	8
Belgium	5	4	3	4	5	5
Bulgaria	21	21	20	20	22	21
Cyprus	2	2	2	2	2	3
Czech Republic	19	20	21	21	20	18

To make global ranking across all years

Sum all columns and rank the result

```
totalSum = pivedu.sum(axis = 1)
totalSum.rank(ascending = False, method = ’dense’)
         .sort_values().head()
```

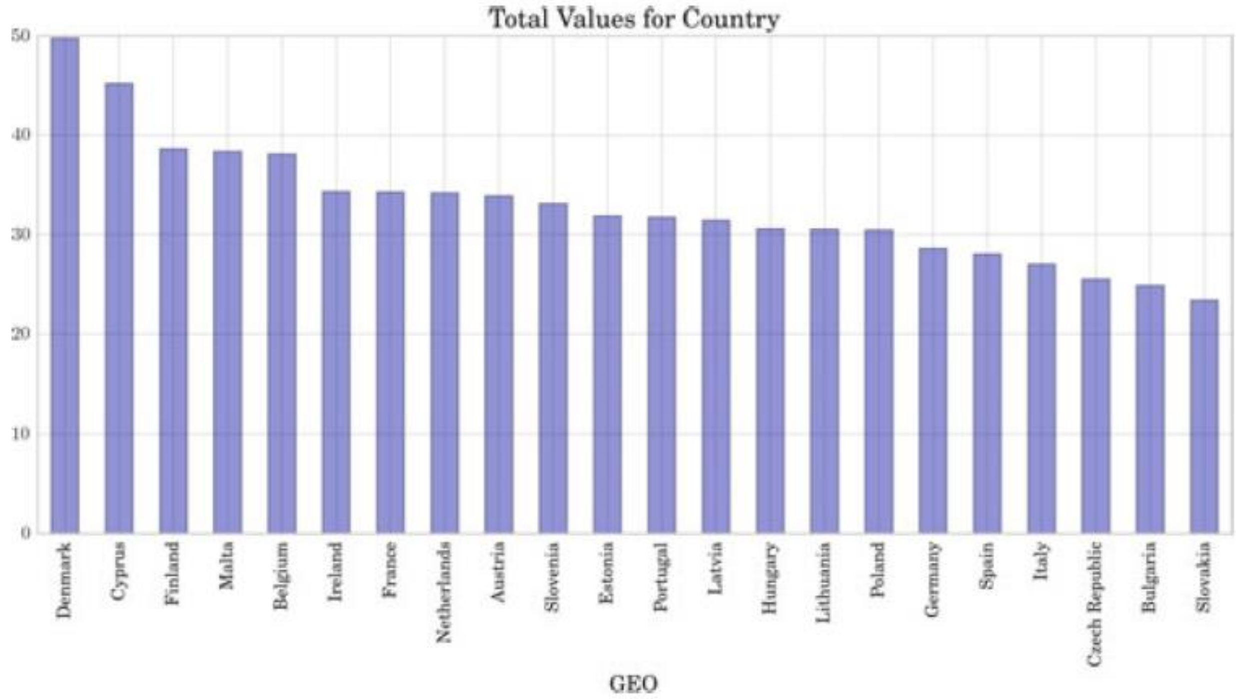
```
GEO
Denmark             1
Cyprus              2
Finland             3
Malta               4
Belgium             5
dtype: float64
```

Plotting

Plotting the accumulated values for each country over the last six years by taking the series obtained in the last example and plot it directly.

```
totalSum = pivedu.sum(axis = 1)
                 .sort_values(ascending = False)
totalSum.plot(kind = ’bar’, style = ’b’, alpha = 0.4,
              title = "Total Values for Country")
```

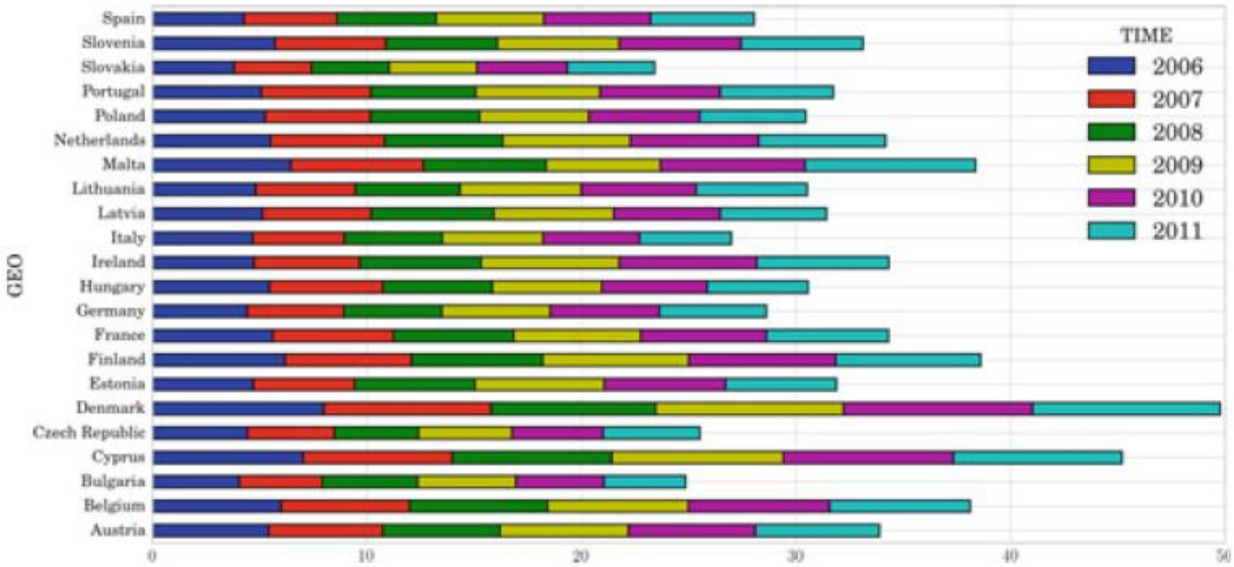
Output



Plotting a data frame directly

Treat columns as a separate series and plot the value for reach year

```
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))
```


GEO
Spain
Slovenia
Slovakia
Portugal
Poland
Netherlands
Malta
Lithuania
Latvia
Italy
Ireland
Hungary
Germany
France
Finland
Estonia
Denmark
Czech Republic
Cyprus
Bulgaria
Belgium
Austria
TIME