FDA_Countries_analysis

September 22, 2018

0.0.1 Content

- Clinical trials by country distribution (pie chart)
- Clinical trials by country distribution (world map)

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

```
In [2]: countries_full_df = pd.read_csv(
            r"c:\Dev\4. Python\03. XML converter of FDA list\goodDB\04. 2018Sep17_13-28-24\FDA
        # countries_full_df.head()
```

```
In [3]: #Data cleaning
        countries_full_df['country'][countries_full_df['country'] == "United States"] = "Unite
        countries_full_df['country'][countries_full_df['country'] == "Congo, The Democratic Rep
        countries_full_df['country'][countries_full_df['country'] == "Czech Republic"] = "Czec
```

```
In [4]: counted_countries = countries_full_df.groupby('country').count()
```

```
In [5]: %run -i create_uniform_buckets.py
        bucketsArray = createUniformBucketsFromSeries(pd.Series(counted_countries.nct_id), numb
```

```
Max value:114420
Min value:1
Series Length = 203
Ideal bucket len:25
Bucket value counts:[27, 25, 24, 25, 24, 24, 24, 24, 6]
```

```
In [6]: sorted_counted_countries = pd.DataFrame(counted_countries.sort_values(by="nct_id", asce
        #Have to make index correct - so it is not resolved to United States Minor Outlying Is
        #In dataset it is just "United States"
        #http://cmdlinetips.com/2018/03/how-to-change-column-names-and-row-indexes-in-pandas/
        first_index = sorted_counted_countries.index[0]
        print(first_index)
        sorted_counted_countries.rename(index={first_index:"United States of America"}, inplac

        maxCountryCount = sorted_counted_countries.nct_id.max()
        print("max:"+str(maxCountryCount))
        minCountryCount = sorted_counted_countries.nct_id.min()
        print("min:"+str(minCountryCount))

        sorted_counted_countries.head()
```

```
United States of America
max:114420
min:1
```

```
Out[6]:                           nct_id
        country
        United States of America  114420
        **None**                   33652
        France                     20464
        Canada                     18965
        Germany                    17464
```

```
In [7]: print(str(len(sorted_counted_countries.groupby("nct_id"))))
        sorted_counted_countries.head()
```

```
144
```

```
Out[7]:                           nct_id
        country
        United States of America  114420
        **None**                   33652
        France                     20464
        Canada                     18965
        Germany                    17464
```

```
In [8]: FILTER_THRESHOLD = 10000


        sum_of_all_countries = sorted_counted_countries.nct_id.sum()
        print("Sum of all countries:{}".format(sum_of_all_countries))

        sum_of_none = sorted_counted_countries.loc["**None**", "nct_id"]
        print("Sum of none:{}".format(sum_of_none))
        sum_of_all_countries_below_threshold = sorted_counted_countries[sorted_counted_countrie
        # print("Sum of all countries-None:{}".format(sum_of_all_countries_minus_none))
        # sum_of_all_countries_minus_none.head()
        print("sum_of_all_countries_below_threshold:{}".format(sum_of_all_countries_below_thres

        other_name = "Other(<"+str(FILTER_THRESHOLD)+" studies per country)"
        countryListWithNone = pd.DataFrame(sorted_counted_countries[sorted_counted_countries.nc
        countryListWithNone.loc[other_name] = sum_of_all_countries - countryListWithNone.nct_i
        # countryListWithNone.to_clipboard()
        # countryListWithNone.head(20)


        countryListWithoutNone = pd.DataFrame(countryListWithNone[countryListWithNone.index !=
        # countryListWithoutNone.head(20)

        countryListWithoutNoneWithouOther = pd.DataFrame(countryListWithoutNone[countryListWit
        # countryListWithoutNoneWithouOther.head(20)
```

```
Sum of all countries:419279
Sum of none:33652
sum_of_all_countries_below_threshold:164178
```

```
In [9]: # filtered_sorted_counted_countries = pd.DataFrame(sorted_counted_countries[sorted_cou
        # other_countries_len = len(sorted_counted_countries)-len(filtered_sorted_counted_coun
        # # print(sorted_counted_countries.loc["**None**", "nct_id"])

        # filtered_sorted_counted_countries.loc[other_name] = \
        #                          other_countries_len+sorted_counted_countries.loc["**None**",
        # filtered_sorted_counted_countries = pd.DataFrame(filtered_sorted_counted_countries.d
        # print(str(len(filtered_sorted_counted_countries)))
        # filtered_sorted_counted_countries.head()
```

```
In [10]: # print("Number of bins in filtered list:%i"%len(filtered_sorted_counted_countries.gr


         # countries_without_none = pd.DataFrame(sorted_counted_countries[sorted_counted_count
         # count_with_none = len(countries_without_none)
         # # print("Count before dropping None:{}".format(count_with_none))
         # countries_without_none.drop(["**None**"], inplace=True)

         # countries_without_none.loc[other_name] = 23
         # # print("Count without None:{}".format(countries_without_none.loc[other_name]))
         # # countries_without_none.head()
         # # filtered_sorted_counted_countries.head()
```

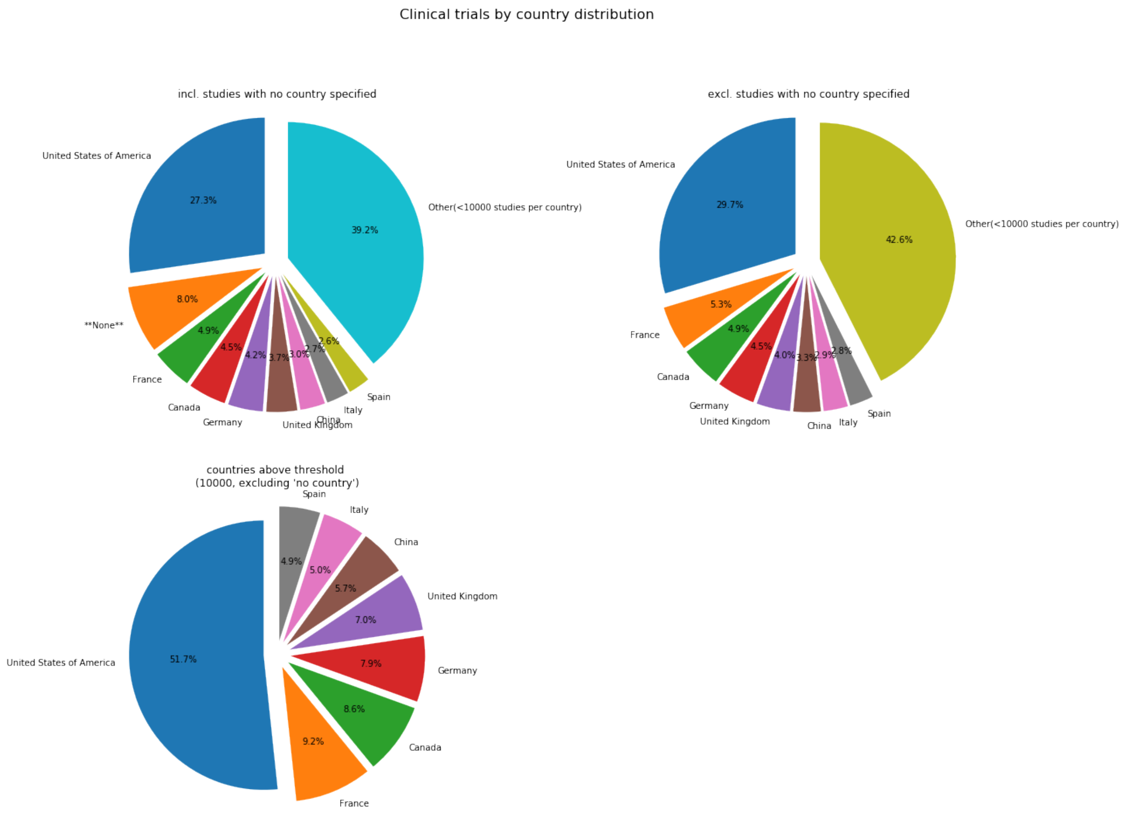
```
In [11]: # explode_arr = list(np.zeros(len(filtered_sorted_counted_countries.groupby("nct_id")
         # explode_arr.append(0.15)
```

1 Clinical trials by country distribution

Table of content

```
In [12]: # unfiltered_countries = pd.DataFrame(filtered_sorted_counted_countries.drop(["**None


         fig = plt.figure(figsize=(20, 15))
```

```
fig.suptitle("Clinical trials by country distribution", fontsize=16)
plt.subplot(221)
plt.pie(countryListWithNone,
        explode = list(np.zeros(len(countryListWithNone))+0.1),
        labels=countryListWithNone.index,
        autopct='%1.1f%%',
        shadow=False, startangle=90)

# plt.title("Clinical trials by country distribution")
plt.title("incl. studies with no country specified")
plt.axis('equal')


plt.subplot(222)
plt.title("excl. studies with no country specified")
plt.pie(countryListWithoutNone,
        explode = list(np.zeros(len(countryListWithoutNone))+0.1),
        labels=countryListWithoutNone.index,
        autopct='%1.1f%%',
        shadow=False, startangle=90)
plt.axis('equal')



plt.subplot(223)
plt.pie(countryListWithoutNoneWithouOther,
        explode = list(np.zeros(len(countryListWithoutNoneWithouOther))+0.1),
        labels=countryListWithoutNoneWithouOther.index,
        autopct='%1.1f%%',
        shadow=False, startangle=90)
plt.axis('equal')

plt.title("countries above threshold \n({}, excluding 'no country')".format(FILTER_TH

plt.show()
```



```
In [13]: %run country_codes_conversion.py

         for row in sorted_counted_countries.index:
             iso2digitCode = get2ISOcodeFromCountryName(row)
             sorted_counted_countries.loc[row, "iso2"] = str(iso2digitCode)

In [14]: sorted_counted_countries.loc[sorted_counted_countries.iso2=='CZ']

Out[14]: Empty DataFrame
         Columns: [nct_id, iso2]
         Index: []

In [15]: iso2_check_for_duplicates = pd.DataFrame(sorted_counted_countries.groupby(by="iso2").c
         #All should be 1s (no 2s - those are duplicates)
         iso2_check_for_duplicates.head()

Out[15]:                                       nct_id
         iso2
         ad                                         1
         no_code_former serbia and montenegro       1
         my                                         1
         mz                                         1
         na                                         1
```

```
In [16]: exportDF = pd.DataFrame(sorted_counted_countries)
         #CLEANING BEFORE EXPORTING TO JSON
         exportDF["full_country_name"] = exportDF.index
         # exportDF.index = exportDF.iso2
         # exportDF.drop(columns="iso2", inplace=True)
         exportDF.rename(columns={"nct_id":"nct_id_count"}, inplace=True)

         # exportDF.head()
```

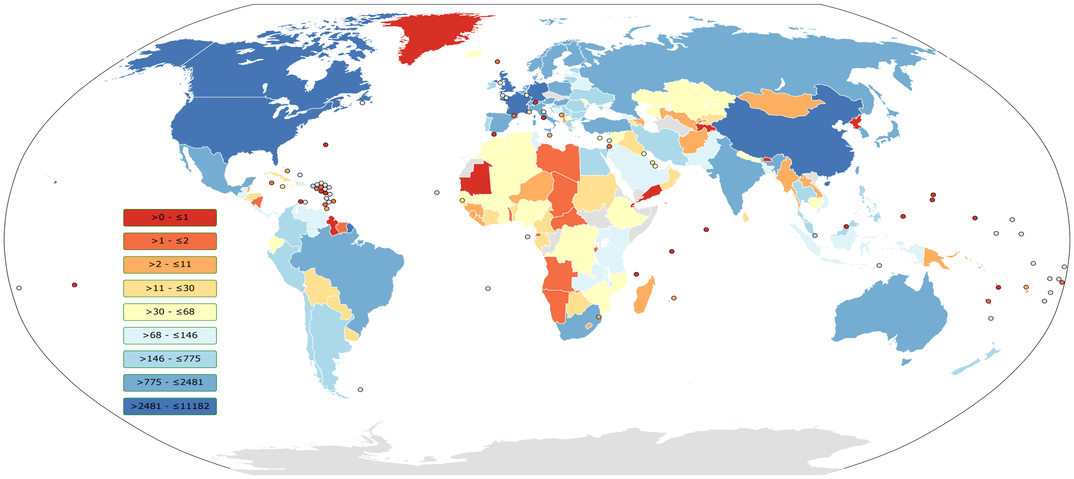
```
In [17]: %run transformSVG.py

         svgConverted = transformCountrySVG(exportDF, bucketsArray,  \
                                          [ "#d73027", "#f46d43", "#fdae61", "#fee090", "#fff
                                            "#abd9e9", "#74add1", "#4575b4"]  \
                          )

         # SVG(svgConverted)
         #str(len(svgConverted))+"     /n"+svgConverted[:1000]

         from IPython.core.display import display, HTML, DisplayObject
         from IPython.display import SVG
```

2 Clinical trials by country (Sep 2018)

Table of content

```
In [18]: SVG(svgConverted)
```

Out[18]:


>0 - ≤1
>1 - ≤2
>2 - ≤11
>11 - ≤30
>30 - ≤68
>68 - ≤146
>146 - ≤775
>775 - ≤2481
>2481 - ≤11182