Python — Analýza a vizualizace finančních sankcí EU

LovelyData.cz

Obsah

1. Úvod

Určitě jste zaznamenali zprávy o finančních sankcích uvalovaných na osoby nebo organizace. Pokud si chcete udělat přesnější představu o tom kdy a na koho byly sankce uvaleny nebo jak se vyvíjela politická situace, můžete využít dostupná data zveřejňovaná EU.

2. Co se naučíte

Pro analýzu a vizualizaci nám dobře poslouží Python a jeho knihovny Pandas, Matplotlib a Seaborn.

V tomto článku si ukážeme jak na to. Probereme následující úkoly:

- Efektivní načtení zdrojových dat (jen to co potřebujeme)
- Odstranění nepotřebných dat
- Vytvoření nových sloupců
- Spojení data (Inner join)
- Vytvoření kontingenční tabulky
- Vizualizaci sankcí pomocí Heatmap
- Jak nastavit velikost v pixelech (standardně je v palcích)

3. IDE pro datovou vědu

Pro datového analytika je dnes k dispozici spousta nástrojů. Mnoho z nich je zdarma, což přispělo k demokratizaci a rychlému rozšíření datové vědy.

Mezi nejpoužívanější patří Jupyter Notebook. Tomu poslední dobou zdatně sekunduje Visual Studio Code, který nabízí mnoho rozšíření. *A to nejen pro práci v Pythonu a Jupyter Notebooku.*

- Jupyter
- Visual Studio Code

4. Zdrojová data

Pro naše potřeby si vystačíme pouze s 2 datovými sadami:

- Finanční sankce EU
- Seznam zemí a odpovídajících ISO kódů

5. Import knihoven a nastavení

Pro práci s daty budeme potřebovat univerzální knihovnu Pandas. Seaborn je knihovna pro vizualizaci dat, která je založená na knihovně Matplotlib. Knihovnu Matplotlib můžeme používat za *pramáti* pythonovské vizualizace, protože ji využívá spousta dalších projektů.

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

pd.set_option('display.max_rows', None)  # Chceme vidět všechny řádky
pd.set_option('display.max_columns', None)  # Chceme vidět všechny
sloupce
```

6. Načtení dat - sankce

Základní informace o sankcích najdete na [této adrese](#).

Pro naše potřeby si stáhneme seznam konsolidovaných finančních sankcí ve formátu CSV. Protože tato sada obsahuje mnoho sloupců, které nevyužijeme, omezíme načtení jen na vybrané.

```
sankce = pd.read_csv
('https://webgate.ec.europa.eu/fsd/fsf/public/files/csvFullSanctionsList
_1_1/content?token=dG9rZW4tMjAxNw',
  sep=';',
  usecols=('Entity_LogicalId', 'Entity_Regulation_EntryIntoForceDate',
    'Citizenship_CountryIso2Code',
'Entity_SubjectType_ClassificationCode'),  # Jen vybrané sloupce
  parse_dates=['Entity_Regulation_EntryIntoForceDate',]  # Tento sloupec
načteme jako datum
)
sankce.head()
```

```
|   | Entity_LogicalId | Entity_SubjectType_ClassificationCode |
Entity_Regulation_EntryIntoForceDate | Citizenship_CountryIso2Code |
|---|------------------|---------------------------------------
|--------------------------------------|-----------------------------|
| 0 |               13 | person                                 | 2003-
07-07 00:00:00                  | nan                           |
| 1 |               13 | person                                 | 2003-
07-07 00:00:00                  | nan                           |
| 2 |               13 | person                                 | 2003-
07-07 00:00:00                  | nan                           |
| 3 |               13 | person                                 | 2003-
07-07 00:00:00                  | nan                           |
| 4 |               13 | person                                 | 2003-
07-07 00:00:00                  | IQ                            |
```

7. Načtení dat - kódy zemí

Data o sankcích obsahují pouze dvoupísmenné označení země. Protože je jednodušší mít k dispozici běžný název země, potřebujeme data obohatit. K tomu dobře poslouží seznam kódů a zemí, který najdeme například zde.

```
zeme = pd.read_csv('https://datahub.io/core/country-list/r/data.csv')
zeme.head()
```

```
|   | Name           | Code |
|---|----------------|------|
| 0 | Afghanistan    | AF   |
| 1 | Åland Islands  | AX   |
| 2 | Albania        | AL   |
| 3 | Algeria        | DZ   |
| 4 | American Samoa | AS   |
```

8. Vymaž řádky, kde chybí Citizenship_CountryIso2Code

Budeme zkoumat sankce podle jednotlivých zemí a proto zahodíme řádky, kde chybí kód země.

```
sankce.dropna(
  subset=['Citizenship_CountryIso2Code'],
  inplace=True
)
```

9. Vymaž duplicity pro sloupec Entity_LogicalId

Některé osoby jsou v seznamu několikrát a vyskytují se v různých zemích. Pro naše účely tyto duplicity odstraníme a ponecháme pouze 1. záznam, pokud se taková duplicita vyskytuje.

```
sankce.drop_duplicates(
  subset='Entity_LogicalId',
  keep='first',  # ponech pouze 1. výskyt
  inplace=True
)
```

10. Vytvoř sloupec Rok (kdy vstoupila sankce v platnost)

Pro naši vizualizaci potřebujeme pouze rok uvalení sankce.

```
sankce['Year'] = sankce['Entity_Regulation_EntryIntoForceDate'].dt.year
```

11. Přidej ke kódům názvy zemí

Propojíme obě datové sady, abychom z kódu země získali její název.

```
sankce_zeme = pd.merge(sankce, zeme,
  how='inner',  # Inner join
  left_on='Citizenship_CountryIso2Code',
  right_on='Code'
)
```

12. Kontingenční tabulka

Pro vizualizaci budeme potřebovat seskupená data. K tomu využijeme knihovnu Pandas, která *umí* kontingenční tabulky na jedničku.

Spočítáme počet osob pro jednotlivé země a roky.

```
cols = ['Name', 'Year', 'Entity_LogicalId']
flt1 = sankce_zeme['Entity_SubjectType_ClassificationCode'] == 'person'
# Pouze řádky, týkající se osob

kont_tabulka = pd.pivot_table(sankce_zeme.loc[flt1, cols],
  index=cols[0:2],  # První dva sloupce = Name, Year
  values='Entity_LogicalId',
  fill_value=0,  # Prázdné hodnoty nahraď nulou
  aggfunc='count',  # Použij agregační funkci Count
).reset_index()  # Udělej z indexů sloupce

kont_tabulka.head()
```

```
|   | Name        | Year | Entity_LogicalId |
|---|-------------|------|------------------|
| 0 | Afghanistan | 2007 |                3 |
| 1 | Afghanistan | 2011 |                2 |
| 2 | Afghanistan | 2012 |                1 |
| 3 | Afghanistan | 2013 |                1 |
| 4 | Afghanistan | 2017 |              117 |
```

13. Vizualizace - Heatmap

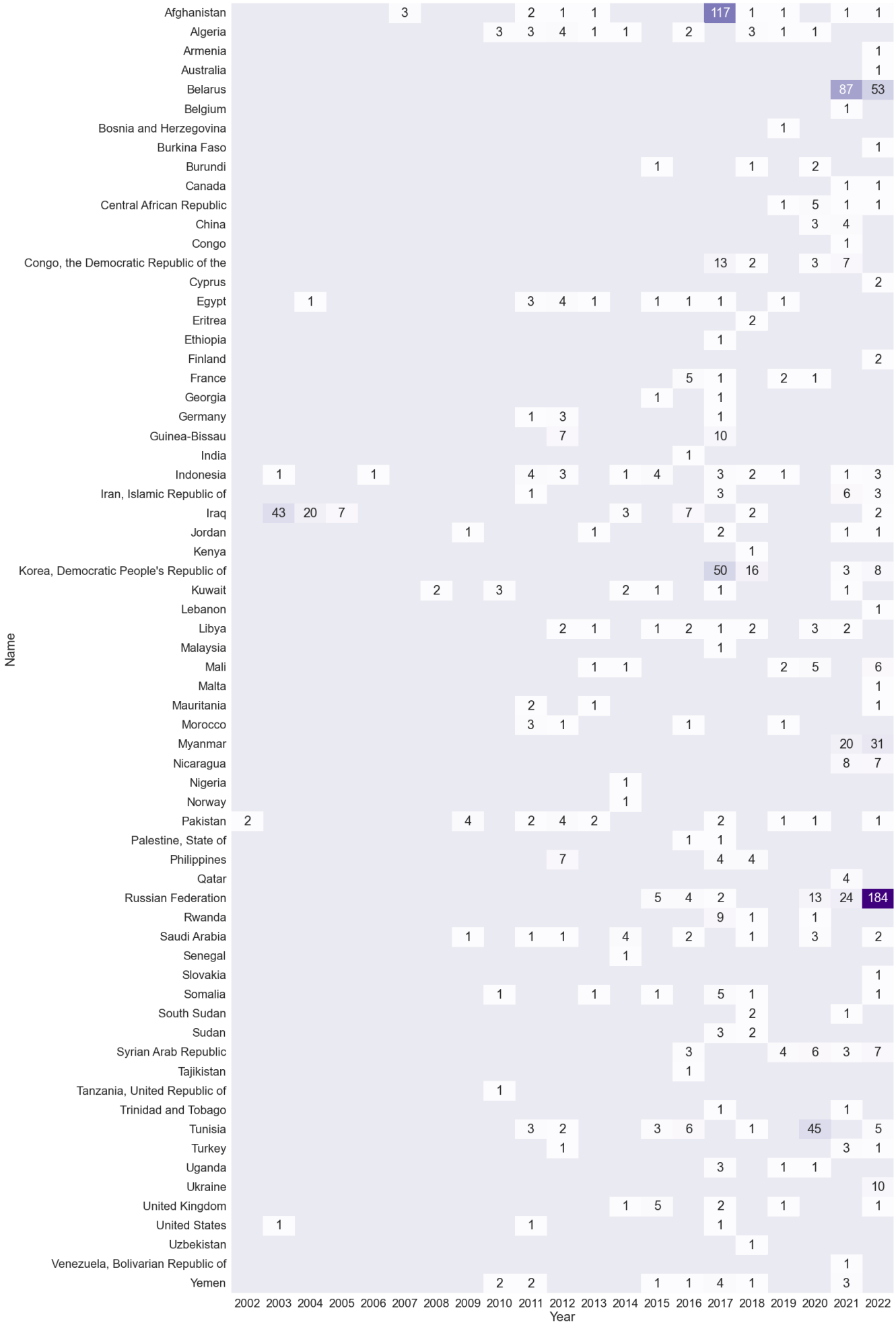
Možností, jak vizualizovat data je mnoho. Úkolem datového analytika je vybrat takovou, která je přehledná a která umožní snadnou a rychlou orientaci.

Pro přehled sankcí jsme vybrali Heatmap, která se občas v češtině překládá jako *tepelná mapa*. My se ale budeme držet anglického a běžně používaného názvu *Heatmap*.

Heatmap je vlastně tabulka, kde jsou hodnoty barevně odlišeny. Pro účely naší datové sady je tato vizualizace ideální, protože se v ní snadno bude orientovat i běžný uživatel.

```
sns.set(font_scale=2)  # Zvětšíme velikost písma
px = 1/plt.rcParams['figure.dpi']  # zjistíme velikost pixelu v palcích
f, ax = plt.subplots(figsize=(1600*px, 3200*px))  # Nadefinujeme
velikost grafu v pixelech

sns.heatmap(kont_tabulka.pivot('Name', 'Year', 'Entity_LogicalId'),
  cmap='Purples',  # Barevné schéma vhodné pro kvantitativní data
  annot=True,  # Chceme zobrazit hodnoty
  fmt='.0f',  # Formát čísla
  cbar=False,  # Nechceme zobrazit barevný pruh na straně
  ax=ax  # Použijeme nastavení pro vizualizaci
)
```


Name
Year
2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022
Afghanistan 3 2 1 1 117 1 1 1 1
Algeria 3 3 4 1 1 2 3 1 1
Armenia 1
Australia 1
Belarus 87 53
Belgium 1
Bosnia and Herzegovina 1
Burkina Faso 1
Burundi 1 1 2
Canada 1 1
Central African Republic 1 5 1 1
China 3 4
Congo 1
Congo, the Democratic Republic of the 13 2 3 7
Cyprus 2
Egypt 1 3 4 1 1 1 1 1
Eritrea 2
Ethiopia 1
Finland 2
France 5 1 2 1
Georgia 1 1
Germany 1 3 1
Guinea-Bissau 7 10
India 1
Indonesia 1 1 4 3 1 4 3 2 1 1 3
Iran, Islamic Republic of 1 3 6 3
Iraq 43 20 7 3 7 2
Jordan 1 1 2 1 1
Kenya 1
Korea, Democratic People's Republic of 50 16 3 8
Kuwait 2 3 2 1 1 1
Lebanon 1
Libya 2 1 1 2 1 2 3 2
Malaysia 1
Mali 1 1 2 5 6
Malta 1
Mauritania 2 1 1
Morocco 3 1 1 1
Myanmar 20 31
Nicaragua 8 7
Nigeria 1
Norway 1
Pakistan 2 4 2 4 2 2 1 1 1
Palestine, State of 1 1
Philippines 7 4 4
Qatar 4
Russian Federation 5 4 2 13 24 184
Rwanda 9 1 1
Saudi Arabia 1 1 1 4 2 1 3 2
Senegal 1
Slovakia 1
Somalia 1 1 1 5 1 1
South Sudan 2 1
Sudan 3 2
Syrian Arab Republic 3 4 6 3 7
Tajikistan 1
Tanzania, United Republic of 1
Trinidad and Tobago 1 1
Tunisia 3 2 3 6 1 45 5
Turkey 1 3 1
Uganda 3 1 1
Ukraine 10
United Kingdom 1 5 2 1 1
United States 1 1 1 1
Uzbekistan 1
Venezuela, Bolivarian Republic of 1
Yemen 2 2 1 1 4 1 3

14. Ke stažení

Neradi píšete kód? Stáhněte si okomentované řešení:

- Jupyter Notebook
- Python