EDA_M-2

April 14, 2022

[ ]:
```
!pip install textblob
```

```
Collecting textblob
  Downloading textblob-0.17.1-py2.py3-none-any.whl (636 kB)
     |                  | 636 kB 246 kB/s eta 0:00:01
Requirement already satisfied: nltk>=3.1 in
/home/ada/anaconda3/lib/python3.8/site-packages (from textblob) (3.6.1)
Requirement already satisfied: click in /home/ada/anaconda3/lib/python3.8/site-
packages (from nltk>=3.1->textblob) (7.1.2)
Requirement already satisfied: tqdm in /home/ada/anaconda3/lib/python3.8/site-
packages (from nltk>=3.1->textblob) (4.62.3)
Requirement already satisfied: regex in /home/ada/anaconda3/lib/python3.8/site-
packages (from nltk>=3.1->textblob) (2021.4.4)
Requirement already satisfied: joblib in /home/ada/anaconda3/lib/python3.8/site-
packages (from nltk>=3.1->textblob) (1.0.1)
Installing collected packages: textblob
Successfully installed textblob-0.17.1
```

[ ]:
```
import pandas as pd
import numpy as np
import spacy
#from tqdm.auto import tqdm
import plotly.express as px
from wordcloud import WordCloud
from matplotlib import pyplot as plt
from datetime import datetime, timedelta
from collections import Counter
import itertools
from spacy import displacy
import textstat
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from nltk import ngrams
from langdetect import detect
import nltk
import swifter
from textblob import TextBlob
import copy
#from IPython.display import Image
```

```
#import kaleido
import warnings
warnings.filterwarnings('ignore')
from nltk import everygrams
from nltk.corpus import stopwords
import re
```

[6]:
```
! python -m spacy download en_core_web_sm
```

```
Collecting en_core_web_sm==2.2.5
  Downloading https://github.com/explosion/spacy-
models/releases/download/en_core_web_sm-2.2.5/en_core_web_sm-2.2.5.tar.gz (12.0
MB)
     |                  | 12.0 MB 12.8 MB/s
Requirement already satisfied: spacy>=2.2.2 in
/usr/local/lib/python3.7/dist-packages (from en_core_web_sm==2.2.5) (2.2.4)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (2.23.0)
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.7/dist-
packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (7.4.0)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.5)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (4.63.0)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.6)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (2.0.6)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.7/dist-
packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.21.5)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-
packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (57.4.0)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (1.1.3)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (3.0.6)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in
```

```
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (0.4.1)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in
/usr/local/lib/python3.7/dist-packages (from
spacy>=2.2.2->en_core_web_sm==2.2.5) (0.9.0)
Requirement already satisfied: importlib-metadata>=0.20 in
/usr/local/lib/python3.7/dist-packages (from
catalogue<1.1.0,>=0.0.7->spacy>=2.2.2->en_core_web_sm==2.2.5) (4.11.3)
Requirement already satisfied: typing-extensions>=3.6.4 in
/usr/local/lib/python3.7/dist-packages (from importlib-
metadata>=0.20->catalogue<1.1.0,>=0.0.7->spacy>=2.2.2->en_core_web_sm==2.2.5)
(3.10.0.2)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-
metadata>=0.20->catalogue<1.1.0,>=0.0.7->spacy>=2.2.2->en_core_web_sm==2.2.5)
(3.7.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from
requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from
requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from
requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5) (2021.10.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.5)
(2.10)
 Download and installation successful
You can now load the model via spacy.load('en_core_web_sm')
```

[ ]:
```
df1 = pd.read_csv("../scrapowanie/pr1.csv")
df2 = pd.read_csv("../scrapowanie/pr2.csv")
```

[ ]:
```
df = df1.append(df2, ignore_index=True)
```

[11]:
```
df
```

[11]:

	title \
0	Interoperability of messaging services - a gam…
1	EU market adjusting to lack of sunflower oil f…
2	S&Ds welcome the use of cohesion funds in EU r…
3	Roaming calls within the EU remain cheap and i…
4	S&Ds welcome the Strategic Compass as a big st…
…	…
12990	The world is learning about lab tests…
12991	Results of EUFORES' Inter-Parliamentary Meetin…

```
12992  CPME: Health services: Patients and Medical do…
12993  CPME: Health services: Patients and Medical do…
12994  Mobility: the panacea for the EU labour market…

                                                   text  \
0      Interoperability of messaging services is one …
1      Brussels, 24 March 2022 - A month after the in…
2      The Socialists and Democrats in the European P…
3      The European Parliament has adopted the new ru…
4      One month ago, Vladimir Putin gave Russian tro…
…                                                    …
12990  The Lab Tests Online community celebrated last…
12991  On January 29th 2008, the Inter-Parliamentary …
12992  EPF and CPME: a Health services Directive is n…
12993  You must have JavaScript enabled to use this f…
12994  Brussels, 21 January 2009: CEEP participated w…

                                          organisation  \
0      S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
1      FEDIOL - The EU Vegetable Oil and Proteinmeal …
2      S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
3      S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
4      S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
…                                                    …
12990  EDMA - European Diagnostic Manufacturers Assoc…
12991  EUFORES - European Forum for Renewable Energy …
12992  CPME - The Standing Committee of European Doctors
12993  CPME - The Standing Committee of European Doctors
12994  CEEP - European Centre of Employers and Enterp…

                           date
0      2022-03-25T00:00:00+01:00
1      2022-03-25T00:00:00+01:00
2      2022-03-25T00:00:00+01:00
3      2022-03-25T00:00:00+01:00
4      2022-03-25T00:00:00+01:00
…                            …
12990  2008-02-14T16:46:41+01:00
12991  2008-02-14T16:38:47+01:00
12992  2008-02-12T15:02:37+01:00
12993  2008-02-12T15:01:07+01:00
12994  2008-01-22T10:29:24+01:00

[12995 rows x 4 columns]
```

0.0.1 Dodanie kategorii

[16]:
```
categories = pd.read_csv('/content/drive/MyDrive/datasets/categories.csv',␣
↪index_col=0)
```

[ ]:
```
categories = pd.read_csv("categories.csv", index_col = 0)
```

[40]:
```
df['category'] = categories
```

[41]:
```
sum(categories.value_counts() >50)
```

[41]: 30

0.1 Statystyka opisowa

Swoje analizy przeprowadziliśmy na zbiorze danych utworzonym z dokumentów pobranych ze strony https://pr.euractiv.com/. Ramka danych zawiera: - `title` tytuł dokumentu, - `text` treść dokumentu, - `organisation` nazwę organizacji, która dany dokument opublikowała, - `data` datę publikacji. - `category` kategorię tematyczną dokumentu

[ ]:
```
print('Liczba obserwacji: ', df.shape[0])
print(df.dtypes) # nic odkrywczego
print(df.apply(pd.isna).sum())
```

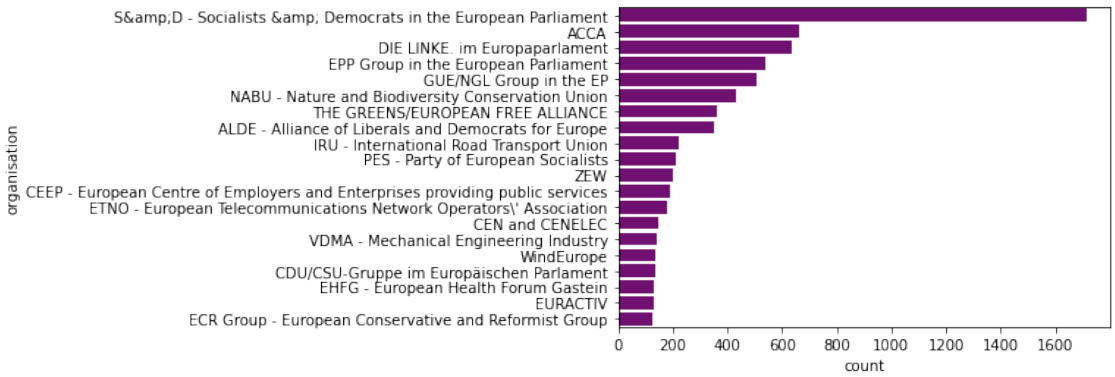
```
Liczba obserwacji:  12995
title           object
text            object
organisation    object
date            object
category        object
dtype: object
title             0
text              0
organisation      0
date              0
category        362
dtype: int64
```

362 braki w kolumnie ‘category’, czyli niektóre teksty nie mają przypisanej żadnej kategorii.

[ ]:
```
df.date = pd.to_datetime(df.date)
df['year'] = df.date
df.year = df.year.apply(lambda x: x.year)
print('Liczba organizacji: ', df.organisation.nunique())
print('Okres z którego pochodzą dokumenty: ', df.year.nunique(), 'lat (lata ',␣
↪df.year.min(),'-', df.year.max(), ')')
```

```
Liczba organizacji:  847
Okres z którego pochodzą dokumenty:  15 lat (lata  2008 - 2022 )
```

[ ]:
```
import seaborn as sns
ax = sns.countplot(y="organisation", data=df,
                   order=df.organisation.value_counts().iloc[:20].index,
                   color='purple')
```



[ ]:
```
pd.DataFrame(df.organisation.value_counts().iloc[60:80])
```

[ ]:

	organisation
EBAA - European Business Aviation Association	39
EFET - European Federation of Energy Traders	39
EuroACE -The European Alliance of Companies for…	37
CEMA - European Agricultural Machinery	36
EBF - European Banking Federation	35
PU Europe	35
ILGA Europe - European Region of the Internatio…	34
FEDIOL - Federation for European Oil and Protei…	34
EPIA - European Photovoltaic Industry Association	33
European Parliament	33
Eurochild	33
ACT	32
EIM - European Rail Infrastructure Managers	32
ECTA - European Competitive Telecommunications …	32
The Brewers of Europe	31
APPLiA - Home Appliance Europe	29
ETRMA - The European Tyre & Rubber Manufact…	29
COCIR	29
ChargeUp Europe	27
EDF - European Disability Forum	27

Organizacje, które opublikowały największą liczbę dokumentów: - `S@D Group` (Postępowy Sojusz Socjalistów i Demokratów w Parlamencie Europejskim) - `ACCA`, międzynarodowa organizacja zrzeszająca specjalistów z zakresu finansów, rachunkowości i zarządzania. Według stanu z marca 2013 ACCA ma na całym świecie w 173 krajach 162 tysięcy członków i 428 tysięcy studentów. - `DIE LINKE` (z niem. Lewica) - socjalizm demokratyczny - `EPP Group` (Europejska Partia Ludowa -

Chrześcijańscy Demokraci), największa grupa polityczna w PE

0.2 Analiza stowarzyszeń politycznych

https://pl.wikipedia.org/wiki/Parlament_Europejski#Sk%C5%82ad - obecni deputowani (w sumie 705 posłów) - rok powstania - liczba publikacji (odczytana z tabelki) - lewicowa/centrowa/prawicowa (-1/0/1)

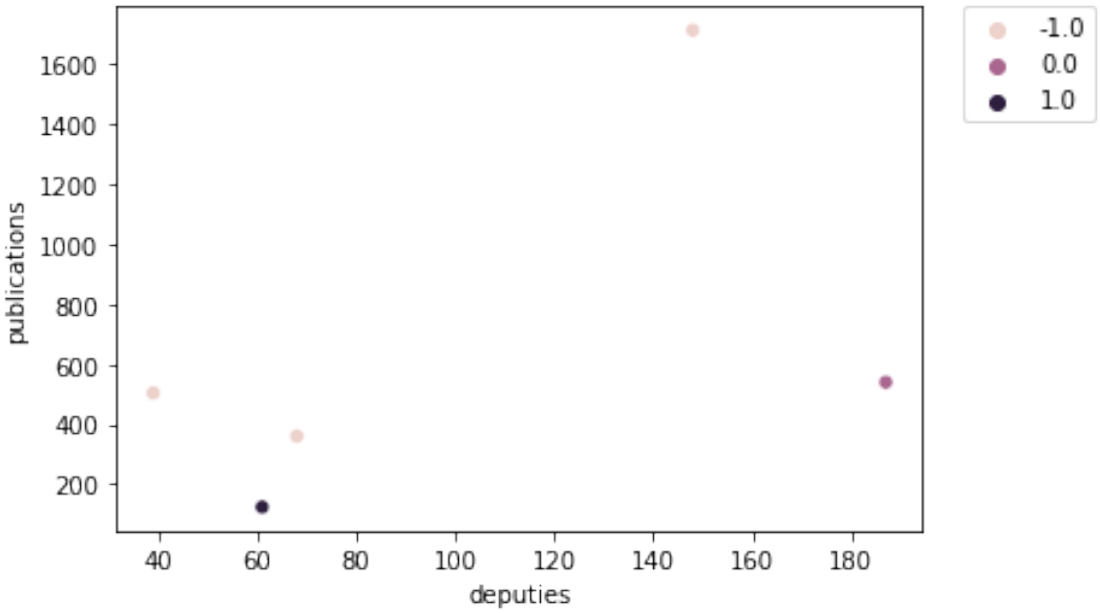
[ ]:
```
import pandas as pd
import numpy as np
```

[ ]:
```
parties_df = pd.DataFrame({
    'name': ['EPP', 'S@D', 'RE', 'ID', 'G/EFA', 'ECR', 'GUE/NGL',␣
 ↪'niezrzeszeni'],
    'deputies': [187, 148, 97, 76, 68, 61,39, 29],
    'origin_year': [1976, 2009, 2019, 2019, 1999, 2009, 1995, np.nan],
    'publications': [540, 1713, np.nan, np.nan , 360, 124, 504, np.nan],
    'c': [0,-1,0,1,-1,1,-1, np.nan]
}) # nie znalazłam 'publications' dla RE i ID :(
parties_df
```

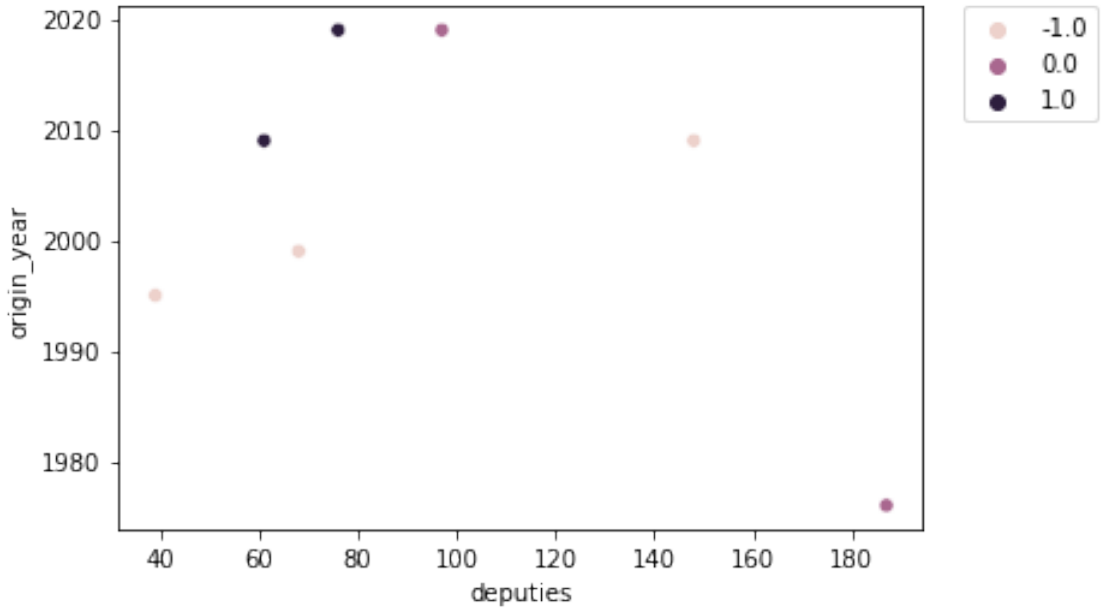
[ ]:

	name	deputies	origin_year	publications	c
0	EPP	187	1976.0	540.0	0.0
1	S@D	148	2009.0	1713.0	-1.0
2	RE	97	2019.0	NaN	0.0
3	ID	76	2019.0	NaN	1.0
4	G/EFA	68	1999.0	360.0	-1.0
5	ECR	61	2009.0	124.0	1.0
6	GUE/NGL	39	1995.0	504.0	-1.0
7	niezrzeszeni	29	NaN	NaN	NaN

[ ]:
```
sns.scatterplot(parties_df.deputies, parties_df.publications, hue=parties_df.c )
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```

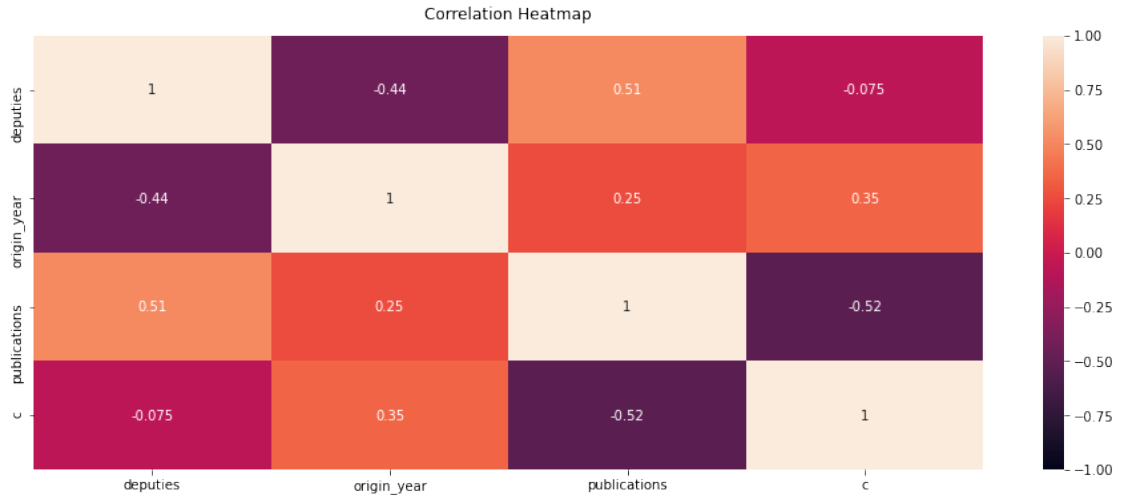


[ ]:
```
sns.scatterplot(parties_df.deputies, parties_df.origin_year, hue=parties_df.c )
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```



[ ]:
```
plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(parties_df.corr(), vmin=-1, vmax=1, annot=True)
```

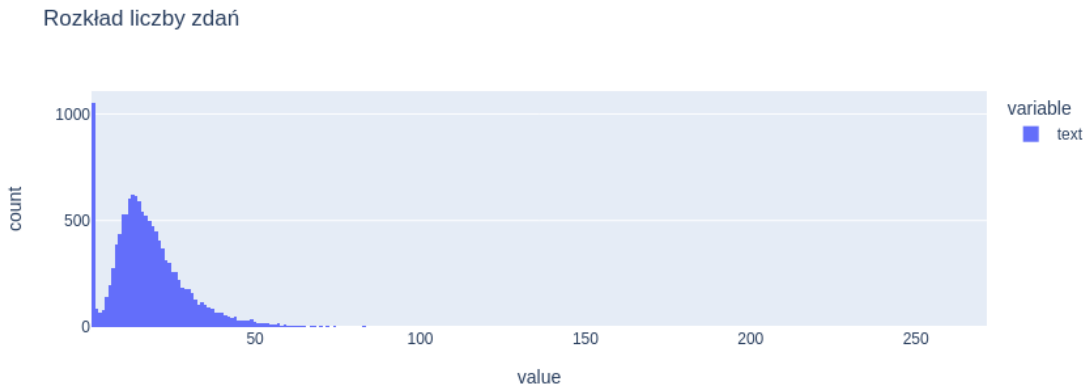
```
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':12}, pad=12);
```



Ogólne wnioski: - im starsza partia, tym więcej eurodeputowanych (-0.44) - im więcej deputowanych, tym więcej publikacji (0.51) - im starsza organizacja, tym więcej publikacji (0.25) Eksperyment ten był przeprowadzony na bardzo małej grupie badanych, więc wyników nie można przełożyć na ogół organizacji (co byłoby zresztą trudne ze względu na konieczność przypisywania danych ręcznie (rok powstania i nakierowanie lewicowe/prawicowe).

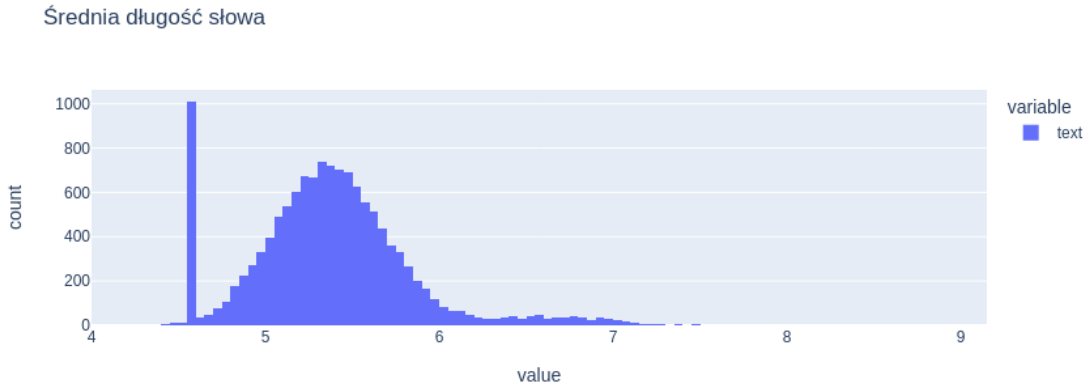
0.2.1 Histogram

[ ]:
```
a = df.text.apply(lambda x: textstat.sentence_count(x)) #złożyć te 3 w jedną␣
↪funkcję i po prostu wywołać
px.histogram(a, title = 'Rozkład liczby zdań')
```



Rozkład ten przypomina rozkład gamma.

```
[ ]: a = df.text.apply(lambda x: textstat.letter_count(x, ignore_spaces=True) /␣
      ↪textstat.lexicon_count(x, removepunct=True)) #średnia długość
     px.histogram(a, title = 'Średnia długość słowa')
```

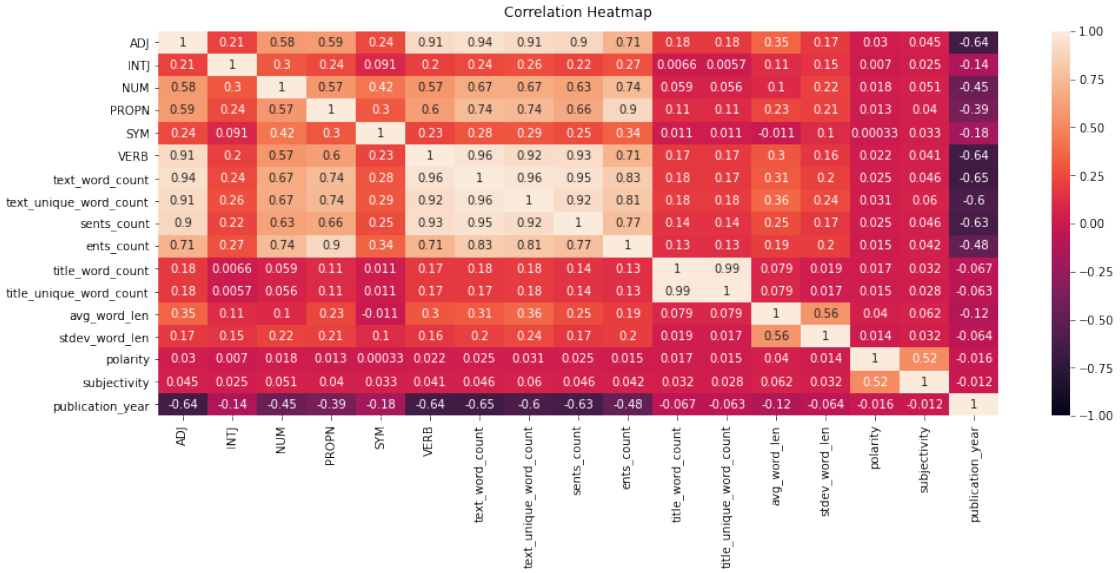


Rozkład normalny.

0.2.2 Korelacje

- ADJ: adjective, e.g. big, old, green, incomprehensible, first
- ADV: adverb, e.g. very, tomorrow, down, where, there
- INTJ: interjection, e.g. psst, ouch, bravo, hello
- NOUN: noun, e.g. girl, cat, tree, air, beauty
- NUM: numeral, e.g. 1, 2017, one, seventy-seven, IV, MMXIV
- PROPN: proper noun, e.g. Mary, John, London, NATO, HBO
- SYM: symbol, e.g. $, %, §, ©, +, −, ×, ÷, =, :),
- VERB: verb, e.g. run, runs, running, eat, ate, eating
- sents_count: liczba zdań
- ents_count: liczba nazw własnych
- polarity [-1,1]: -1 defines a negative sentiment and 1 defines a positive sentiment
- subjectivity [0,1]: quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information

```
[ ]: count_matrix = pd.read_table('dokorelacji-3.csv', sep=',', index_col = 0)
count_matrix.
 ↪drop(['CONJ','X','SPACE','ADV','AUX','CCONJ','DET','PART','PRON','PUNCT','SCONJ'],␣
 ↪inplace=True, axis=1)
count_matrix['publication_year'] = df.year
plt.figure(figsize=(16, 6))
heatmap = sns.heatmap(count_matrix.corr(), vmin=-1, vmax=1, annot=True)
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':12}, pad=12);
```



1. `ADJ` (przymiotnik):

- Słabo skorelowany z `title_word_count` (0.18) - wskazuje to na to, iż tytuły definiowane są bardzo rzeczowo, bez zbędnego wdawania się w opisy (przy pomocy przymiotników).

2. `polarity` (zabarwienie negatywne/pozytywne):

- Mocno skorelowany (0.52) z `subjectivity`- im więcej opinii zawiera tekst, tym jest oceniany na bardziej pozytywny.

3. `publication_year` (data publikacji dokumentu):

- Mocno skorelowany ujemnie (-0.65) z `text_word_count`, co wskazuje na to, że im później publikowany dokument, tymm mniej słów zawiera, czyli na przestrzeni lat dokumenty stają się coraz krótsze.

4. `avg_word_len` (średnia długość słowa):

- Mocno skorelowana (0.36) z `text_(unique_)word_count`, co świadczy o tym, że im więcej słów w tekście, tym też słowa te są dłuższe.

@TODO - wybrać podzbiór

0.2.3 Preprocessing

[42]:
```
df['lang']=df['text'].swifter.apply(detect)
```

```
Pandas Apply:   0%|          | 0/12995 [00:00<?, ?it/s]
```

Przy okazji plot ile jest notek w poszczególnych językach

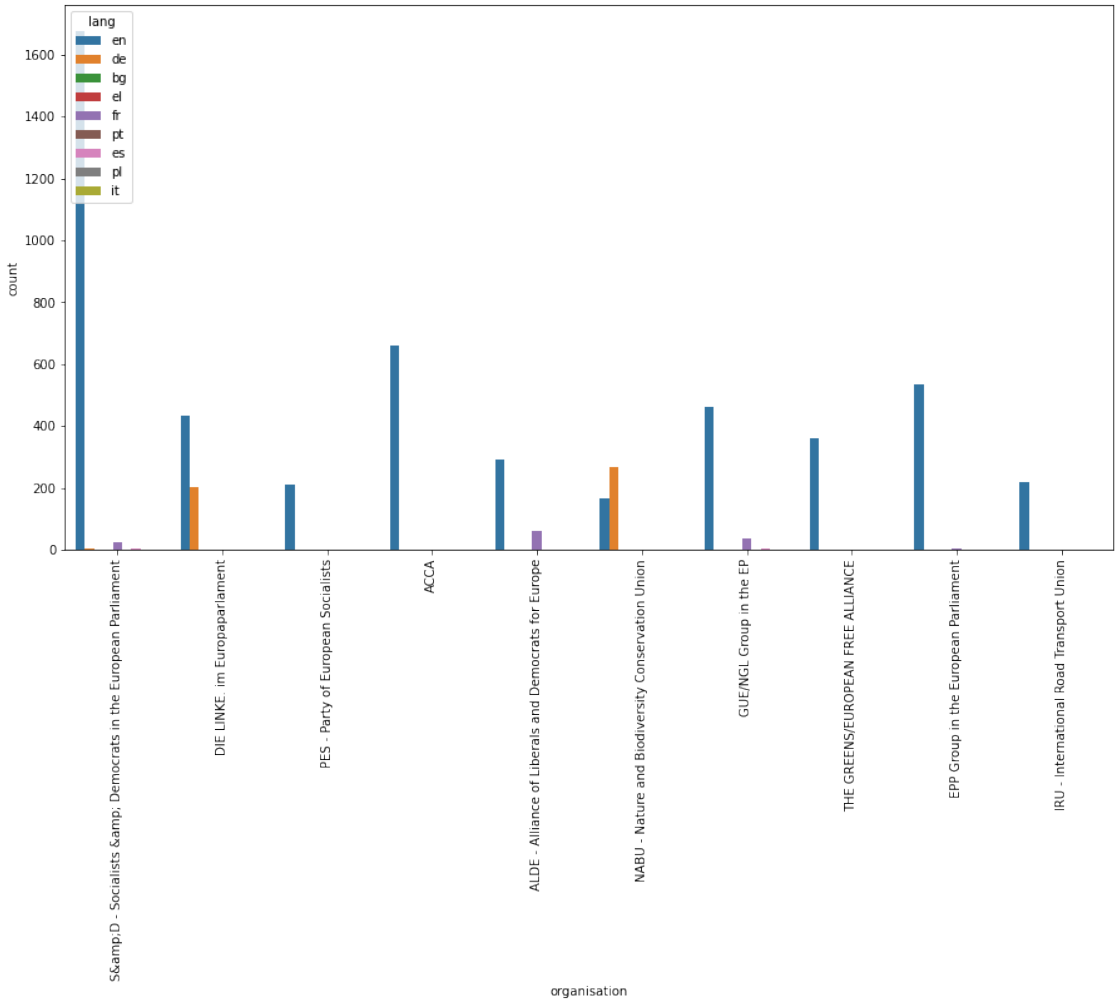
[43]:
```
df['lang'].hist()
```

Dla 10 najczęściej publikujący organizacji sprawdzamy w jakich językach pisały.

[112]:
```
import seaborn as sns
```

[147]:
```
plt.figure(figsize=(15,8))
sns.countplot(data=df_nawszelki.loc[df_nawszelki['organisation'].
 ↪isin(list(df_nawszelki['organisation'].value_counts().head(10).index))],
              x='organisation',
              hue='lang')
plt.xticks(rotation=90)
```

[147]:
```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 <a list of 10 Text major ticklabel objects>)
```



[45]:
```
df_pol = df.loc[df['lang']=='pl']
```

[46]:
```
df_nawszelki = df.copy()
```

[47]:
```
df = df.loc[df['lang']=='en']
```

[48]:
```
df = df.drop(['lang'], axis=1)
```

[36]:
```
import re
```

[50]:
```
df.date = pd.to_datetime(df.date)
df = df.drop_duplicates()
df.loc[:,'text'] = df.loc[:,'text'].apply(lambda x: x.replace('\r\n\r\n',''))
df.loc[:,'text'] = df.loc[:,'text'].apply(lambda x: re.sub('\w*\d\w*','', x)) #␣
 ↪usuwamy cyfry
df.loc[:,'text'] = df.loc[:,'text'].apply(lambda x: re.sub('http','', x))
```

```
df.loc[:,'text'] = df.loc[:,'text'].apply(lambda x: re.sub('www','', x))
df.loc[:,'text'] = df.loc[:,'text'].apply(lambda x: x.replace('+',''))
df.loc[:,'text'] = df.loc[:,'text'].apply(lambda x: x.replace('tel',''))
df.loc[:,'text'] = df.loc[:,'text'].apply(lambda x: x.replace('Tel',''))
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/indexing.py:1951:
SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[154]:
```
df.date = pd.to_datetime(df.date,utc=True)
```

[156]:
```
years = pd.DatetimeIndex(df['date']).year
```

[158]:
```
df = df.drop(['year'],axis=1)
```

Usunięcie wielokrotnie powtarzającego się tekstu

[51]:
```
df.nunique()
```

[51]:
```
title           11927
text            11026
organisation      824
date             6054
category          945
dtype: int64
```

[52]:
```
df = df.loc[df.text.apply(lambda x: x != 'You must have JavaScript enabled to␣
 ↪use this form.')]
```

Mamy przypadki identycznych tekstów z różnymi tytułami

[53]:
```
print(len(df))
print(df.nunique())
print("Liczba artukułów o tym samym tekście:",sum(list(map(lambda x: x > 1,␣
 ↪list(df.text.value_counts())))))
```

```
11047
title           10934
text            11025
organisation      804
date             5933
```

```
category          922
dtype: int64
Liczba artukułów o tym samym tekście: 21
```

Przykład identycznych tekstów

[54]:
```
sample_identical = df.text.value_counts().index[0]
df_identical = df.loc[df.text == sample_identical]
df_identical
```

[54]:
```
                                                  title  \
8496  Peace initiative on Kurdish issue aims to kick…
8642  Peace initiative on Kurdish issue aims to kick…
8644  Peace initiative on Kurdish issue aims to kick…

                                                   text  \
8496  Representatives of the Dalai Lama and Desmond …
8642  Representatives of the Dalai Lama and Desmond …
8644  Representatives of the Dalai Lama and Desmond …

                organisation                       date  \
8496  GUE/NGL Group in the EP  2013-02-06 08:31:28+01:00
8642  GUE/NGL Group in the EP  2012-12-04 08:32:54+01:00
8644  GUE/NGL Group in the EP  2012-12-03 15:46:32+01:00

                                              category
8496       EnlargementEU Priorities 2020Regional Policy
8642            EU Priorities 2020Global EuropeSecurity
8644  EU Priorities 2020Global EuropeJustice & Home …
```

Sugerowane wyjaśnienie:

Walcząc z nacjonalistycznym gniewem, Erdogan (turecki prezydent) wprowadził wstępne reformy dotyczące praw Kurdów, a w 2012 r. rozpoczął negocjacje, aby spróbować zakończyć partyzantkę PKK, która od 1984 r. zabiła 40 000 ludzi. Kruche zawieszenie broni obowiązywało od marca 2013 r. Być może dokument ten był publikowany dla przypomnienia w ważniejszych momentach tychże działań.

[ ]:
```
count_identical = pd.DataFrame(df.text.value_counts())
count_identical = count_identical.loc[count_identical.text > 1,:]
count_identical
df_identical = df.loc[df.text.apply(lambda x: x in count_identical.text)]
df_identical
```

[ ]:
```
                                                  title  \
22     CSRD: European Parliament’s vote promotes an o…
23     PRESS RELEASE - CSRD: European Parliament’s vo…
28               Ukrainians Facing Another Deadly Threat
29               Ukrainians Facing Another Deadly Threat
```

```
48     Joint statement IPA Europe - European Dairy As…
51     Joint statement IPA Europe - European Dairy As…
63     Wealthy Russians linked to Putin must be strip…
67     Wealthy Russians linked to Putin must be strip…
78     S&Ds in Hungary to promote democracy and socia…
85     S&Ds in Hungary to promote democracy and socia…
102    EU data confirms key contribution of biomethan…
155    EU data confirms key contribution of biomethan…
339    S&D MEPs denounce Ortega for turning Nicaragua…
372    S&D MEPs denounce Ortega for turning Nicaragua…
776    Time to act on digital taxation and a global m…
787    Time to act on digital taxation and a global m…
2698       Two Candidates from the People for the People
2700        European Left: TWO CANDIDATES FOR THE PEOPLE
4798   Pittella: Rule of law should not serve politic…
4801   Pittella: Rule of law should not serve politic…
5924   Business as usual - Juncker snubs environment …
5934   Business as usual - Juncker snubs environment …
7605   Troika is not the scapegoat - Parliament calls…
7606                     Europe steps up its IT security
8496   Peace initiative on Kurdish issue aims to kick…
8642   Peace initiative on Kurdish issue aims to kick…
8644   Peace initiative on Kurdish issue aims to kick…
8934   World Water Week 2012: « Water and Food Securi…
8935   World Water Week 2012: « Water and Food Securi…
9719   White Paper on Transport: Encourage a sustaina…
9727   White Paper on Transport: Encourage a sustaina…
10768                          Time for a safe fair ride
10775                          Time for a safe fair ride
12060  COPA-COGECA welcome agricultural council meeti…
12061  COPA-COGECA welcome agricultural council meeti…
12122                                       FP7 - Health
12127                                       FP7 - Health
12258  CEEP Statement on the (future) Review of EU Te…
12277  CEEP Statement on the (future) Review of EU Te…
12462  For a horizontal alignment of sectoral product…
12463  For a horizontal alignment of sectoral product…
12464    Cross border European accreditation in practice
12465    Cross border European accreditation in practice
12469         Cable Industry Revenues Exceed €18 billion
12470         Cable Industry Revenues Exceed €18 billion
12504  IRU City Trophy 2009 to reward municipal coach…
12505  IRU City Trophy 2009 to reward municipal coach…
12506  IRU to reward service quality and accessibilit…
12507  IRU to reward service quality and accessibilit…
12830          Does better health lead to better wealth?
12831          Does better health lead to better wealth?
```

```
12921  GUE/NGL voices its opposition to the return di…
12922  GUE/NGL voices its opposition to the return di…
12985  New energy and climate package for Europe: The…
12986  New energy and climate package for Europe: The…

                                                  text  \
22     Brussels,  March  - TIC Council welcomes the E…
23     Brussels,  March  - TIC Council welcomes the E…
28     Brussels,  March : Since the Russian invasion …
29     Brussels,  March : Since the Russian invasion …
48     The European Dairy Association (EDA) and the I…
51     The European Dairy Association (EDA) and the I…
63     The S&D Group in the European Parliament suppo…
67     The S&D Group in the European Parliament suppo…
78     As part of the Conference on the Future of Eur…
85     As part of the Conference on the Future of Eur…
102    The statistical office of the European Union (…
155    The statistical office of the European Union (…
339    The Socialists and Democrats in the European P…
372    The Socialists and Democrats in the European P…
776    The Socialists and Democrats in the European P…
787    The Socialists and Democrats in the European P…
2698   At the meeting of the Executive Board of the P…
2700   At the meeting of the Executive Board of the P…
4798   Following an exchange of views today with EU C…
4801   Following an exchange of views today with EU C…
5924   Commission Work Programme ignores businesses a…
5934   Commission Work Programme ignores businesses a…
7605   Binding security standards for IT networks are…
7606   Binding security standards for IT networks are…
8496   Representatives of the Dalai Lama and Desmond …
8642   Representatives of the Dalai Lama and Desmond …
8644   Representatives of the Dalai Lama and Desmond …
8934   Brussels,  August "There is no food security w…
8935   Brussels,  August "There is no food security w…
9719   The White Paper on Transport's promotion of In…
9727   The White Paper on Transport's promotion of In…
10768  ANEC, the European Consumer Voice in Standardi…
10775  ANEC, the European Consumer Voice in Standardi…
12060  Copa and Cogeca welcomed in Brussels today the…
12061  Copa and Cogeca welcomed in Brussels today the…
12122  New EU funding opportunities for health!Almost…
12127  New EU funding opportunities for health!Almost…
12258  In view of the upcoming negotiations on the fu…
12277  In view of the upcoming negotiations on the fu…
12462  The European engineering industry welcomes Com…
12463  The European engineering industry welcomes Com…
```

```
12464  IntroductionThe EC Regulation n°/ of  July  se…
12465  IntroductionThe EC Regulation n°/ of  July  se…
12469   million new subscriptions for digital TV, bro…
12470   million new subscriptions for digital TV, bro…
12504  IRU calls for candidates for its City Trophy A…
12505  IRU calls for candidates for its City Trophy A…
12506  IRU calls for candidates for the Eurochallenge…
12507  IRU calls for candidates for the Eurochallenge…
12830  It has long been accepted that greater wealth …
12831  It has long been accepted that greater wealth …
12921  "We will be participating in today's demonstra…
12922  "We will be participating in today's demonstra…
12985  It proposes a stable and flexible EU framework…
12986  It proposes a stable and flexible EU framework…

                                          organisation  \
22                                         TIC-Council
23                                         TIC Council
28                 ECO - European Cancer Organisation
29                       European Cancer Organisation
48                                          IPA Europe
51       The European Probiotic Association (IPA Europe)
63     S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
67     S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
78     S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
85     S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
102                  EBA - European Biogas Association
155                  EBA - European Biogas Association
339    S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
372    S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
776    S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
787    S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
2698                        Party of the European Left
2700                        Party of the European Left
4798   S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
4801   S&amp;amp;D - Socialists &amp;amp; Democrats in the Eu…
5924                             BirdLife International
5934                             BirdLife International
7605              EPP Group in the European Parliament
7606              EPP Group in the European Parliament
8496                           GUE/NGL Group in the EP
8642                           GUE/NGL Group in the EP
8644                           GUE/NGL Group in the EP
8934                                        EuropaBio
8935                                        EuropaBio
9719                              FIA EUROPEAN BUREAU
9727                              FIA EUROPEAN BUREAU
```

```
10768  ANEC - The European Consumer Voice in Standard…
10775  ANEC - The European Consumer Voice in Standard…
12060                                        Copa-Cogeca
12061                                        Copa-Cogeca
12122                                   Interface Europe
12127                                   Interface Europe
12258  CEEP - European Centre of Employers and Enterp…
12277  CEEP - European Centre of Employers and Enterp…
12462                                          Orgalime
12463                                          Orgalime
12464                                          Orgalime
12465                                          Orgalime
12469  Cable Europe - European Cable Communications A…
12470  Cable Europe - European Cable Communications A…
12504           IRU - International Road Transport Union
12505           IRU - International Road Transport Union
12506           IRU - International Road Transport Union
12507           IRU - International Road Transport Union
12830                                               WHO
12831                                               WHO
12921                           GUE/NGL Group in the EP
12922                           GUE/NGL Group in the EP
12985                                       WindEurope
12986                                       WindEurope

                           date  \
22     2022-03-18 00:00:00+01:00
23     2022-03-18 00:00:00+01:00
28     2022-03-16 00:00:00+01:00
29     2022-03-15 00:00:00+01:00
48     2022-03-09 00:00:00+01:00
51     2022-03-09 00:00:00+01:00
63     2022-03-08 00:00:00+01:00
67     2022-03-07 00:00:00+01:00
78     2022-03-04 00:00:00+01:00
85     2022-03-02 00:00:00+01:00
102    2022-03-01 00:00:00+01:00
155    2022-02-04 00:00:00+01:00
339    2021-11-12 00:00:00+01:00
372    2021-10-25 00:00:00+02:00
776    2021-05-03 00:00:00+02:00
787    2021-04-29 00:00:00+02:00
2698   2019-01-28 00:00:00+01:00
2700   2019-01-28 00:00:00+01:00
4798   2017-09-01 00:00:00+02:00
4801   2017-08-31 00:00:00+02:00
5924   2016-10-27 00:00:00+02:00
```

```
5934   2016-10-26 00:00:00+02:00
7605   2014-03-13 00:00:00+01:00
7606   2014-03-13 00:00:00+01:00
8496   2013-02-06 08:31:28+01:00
8642   2012-12-04 08:32:54+01:00
8644   2012-12-03 15:46:32+01:00
8934   2012-08-31 13:26:06+02:00
8935   2012-08-31 13:25:45+02:00
9719   2011-11-24 08:44:31+01:00
9727   2011-11-22 11:33:47+01:00
10768  2011-01-25 14:06:11+01:00
10775  2011-01-24 16:54:14+01:00
12060  2009-09-25 09:43:40+02:00
12061  2009-09-25 08:00:58+02:00
12122  2009-08-25 08:00:21+02:00
12127  2009-08-24 07:21:25+02:00
12258  2009-06-04 07:29:02+02:00
12277  2009-05-22 09:14:56+02:00
12462  2009-03-05 10:30:01+01:00
12463  2009-03-05 10:30:00+01:00
12464  2009-03-05 10:25:06+01:00
12465  2009-03-05 10:25:00+01:00
12469  2009-03-05 09:06:06+01:00
12470  2009-03-05 09:06:00+01:00
12504  2009-01-26 08:36:25+01:00
12505  2009-01-26 08:36:00+01:00
12506  2009-01-26 08:33:28+01:00
12507  2009-01-26 08:33:00+01:00
12830  2008-06-24 14:46:39+02:00
12831  2008-06-24 14:45:30+02:00
12921  2008-05-13 09:02:33+02:00
12922  2008-05-13 09:02:18+02:00
12985  2008-03-04 16:20:33+01:00
12986  2008-03-04 16:19:38+01:00
```

	category	year
22	Innovation & Enterprise	2022
23	Innovation & Enterprise	2022
28	Global Europe	2022
29	Global Europe	2022
48	Agriculture & Food	2022
51	Agriculture & Food	2022
63	Global Europe	2022
67	Global Europe	2022
78	Justice & Home Affairs	2022
85	Justice & Home Affairs	2022
102	Energy	2022

```
155                                                Energy  2022
339                                         Global Europe  2021
372                                         Global Europe  2021
776                                 Innovation & Enterprise  2021
787                                          Euro & Finance  2021
2698                                            InfoSociety  2019
2700                                        Global Europe  2019
4798                                        Global Europe  2017
4801                    Global EuropeJustice & Home Affairs  2017
5924               Agriculture & FoodClimate & Environment  2016
5934               Agriculture & FoodClimate & Environment  2016
7605                                         Euro & Finance  2014
7606                                            InfoSociety  2014
8496          EnlargementEU Priorities 2020Regional Policy  2013
8642               EU Priorities 2020Global EuropeSecurity  2012
8644   EU Priorities 2020Global EuropeJustice & Home …  2012
8934                                    Agriculture & Food  2012
8935                                    Agriculture & Food  2012
9719                                             Transport  2011
9727                          Health & ConsumersTransport  2011
10768                                   Health & Consumers  2011
10775                     Health & ConsumersTrade & Society  2011
12060                  Agriculture & FoodHealth & Consumers  2009
12061                  Agriculture & FoodHealth & Consumers  2009
12122                                   Health & Consumers  2009
12127                      Public AffairsHealth & Consumers  2009
12258                                Social Europe & Jobs  2009
12277                                Social Europe & Jobs  2009
12462                                                 NaN  2009
12463                Sustainable Dev.Climate & Environment  2009
12464                                                 NaN  2009
12465                Sustainable Dev.Climate & Environment  2009
12469                                                 NaN  2009
12470  Sustainable Dev.Innovation & EnterpriseClimate…  2009
12504                                  TransportInfoSociety  2009
12505                                  TransportInfoSociety  2009
12506                                  TransportInfoSociety  2009
12507                                  TransportInfoSociety  2009
12830  Social Europe & JobsInnovation & EnterpriseHea…  2008
12831  Social Europe & JobsInnovation & EnterpriseHea…  2008
12921  Justice & Home AffairsSocial Europe & JobsEU P…  2008
12922  Justice & Home AffairsSocial Europe & JobsEU P…  2008
12985                                                 NaN  2008
12986                                                 NaN  2008
```

Wnioski: - daty publikacji identycznych notek mogą być różne - tytuły są identyczne lub bardzo podobne - np. `CSRD: European Parliament's...`, `PRESS RELEASE - CSRD: European`

Parliament's... - podobnie organizacje - często różnią się rozwinięciem skrótu w nawiasach

Krótka analiza polskich tekstów

[ ]:
```
df_pol
```

[ ]:
```
                                                  title  \
5240  Silesia deserves autonomy and Poland should gr…
6834  "No one should face jail for abortion" say S&D…

                                                   text  \
5240  General Assembly of European Free Alliance-Eur…
6834  S&D Euro MPs are outraged by plans to impose a…

                                           organisation  \
5240                      EFA - European Free Alliance
6834  S&amp;D - Socialists &amp; Democrats in the Eu…

                           date  \
5240  2017-03-31 00:00:00+02:00
6834  2016-04-08 00:00:00+02:00

                                              category  year
5240  Global EuropeLanguages & CultureSocial Europe …  2017
6834                           Justice & Home Affairs  2016
```

[ ]:
```
df_pol
pol_doc = en(df_pol.iloc[1,1])
print("Tytuł artykułu:", df_pol.iloc[1,:].title)
print("Data artykułu:", df_pol.iloc[1,:].date)
print("Organizacja:", df_pol.iloc[1,:].organisation)
```

```
Tytuł artykułu: "No one should face jail for abortion" say S&D Euro MPs (PL
below)
Data artykułu: 2016-04-08 00:00:00+02:00
Organizacja: S&amp;D - Socialists &amp; Democrats in the European Parliament
```

[ ]:
```
spacy.displacy.render(pol_doc, style='ent',jupyter=True)
```

```
<IPython.core.display.HTML object>
```

Ważny wniosek: Teksty potrafią być w 2 wersjach językowych ##### Hipoteza: teksty w 2 wersjach językowych zawierają słowo 'below'

[ ]:
```
sum(df.title.apply(lambda x: 'below' in x))
```

[ ]:
```
1
```

Wniosek: Jednak tylko 1 taki dokument zawieraja w tytule słowo 'below'.

0.2.4 Usunięcie informacji kontaktowych

Niektóre z publikowanych notek zawierały opis organizacji. Zebraliśmy słowa typowe dla końca notki prasowej danej organizacji, aby odfiltrować zbędne informacje

[63]:
```
import copy
```

[60]:
```
def delete_contact(df):
    cpdf = copy.deepcopy(df)
    separators = ['About ACCA', 'For further information', 'Press Contact',␣
↪'Richard More O\'Ferrall', 'please contact', '* * *', 'please contact',␣
↪'please contact']
    orgs = ['ACCA', 'EPP Group in the European Parliament', 'GUE/NGL Group in␣
↪the EP', 'THE GREENS/EUROPEAN FREE ALLIANCE',
        'ALDE - Alliance of Liberals and Democrats for Europe', 'IRU -␣
↪International Road Transport Union', 'PES - Party of European Socialists',␣
↪'CEEP - European Centre of Employers and Enterprises providing public␣
↪services']
    sep2 = ['For media enquiries', 'For more information', 'For further␣
↪information']
    orgs2 = ['ACCA', 'ALDE - Alliance of Liberals and Democrats for Europe',␣
↪'CEEP - European Centre of Employers and Enterprises providing public␣
↪services']

    for i in range(len(separators)):
        df_acca_text = cpdf.loc[cpdf.organisation == orgs[i], 'text'].
↪apply(lambda x: x.split(separators[i], 1)[0])
        cpdf.loc[cpdf.organisation == orgs[i], 'text'] = df_acca_text
    for i in range(len(sep2)):
        df_acca_text = cpdf.loc[cpdf.organisation == orgs2[i], 'text'].
↪apply(lambda x: x.split(sep2[i], 1)[0])
        cpdf.loc[cpdf.organisation == orgs2[i], 'text'] = df_acca_text

    return cpdf
```

EPP Group - zbiasowane o 453

[64]:
```
df = delete_contact(df)
```

1 Eksploracja

[20]:
```
en = spacy.load("en_core_web_sm")
```

[65]:
```
docs = df['text'].swifter.apply(en)
```

```
Pandas Apply:   0%|          | 0/11047 [00:00<?, ?it/s]
```

[66]:
```
len(docs)
```

[66]:
```
11047
```

1.1 Wordlclouds

[67]:
```
lemmas = docs.swifter.apply(lambda doc: [token.lemma_ for token in doc if not␣
↪token.is_stop if not token.is_punct])
```

```
Pandas Apply:   0%|          | 0/11047 [00:00<?, ?it/s]
```

[73]:
```
from collections import Counter
import itertools
```

Wszystkie słowa

[176]:
```
def createCounterDeleteMostCommon(lemmas, n):
  word_counts = Counter(list(itertools.chain(*lemmas)))
  wordcountsDf = pd.DataFrame.from_dict(word_counts, orient='index').
↪reset_index()
  wordcountsDf.columns=['word','count']
  for x in list(wordcountsDf.sort_values('count',ascending=False).
↪head(n)['word']):
    del word_counts[x]
  return word_counts
```

[207]:
```
def viewMostCommon(lemmas,n):
  word_counts = Counter(list(itertools.chain(*lemmas)))
  wordcountsDf = pd.DataFrame.from_dict(word_counts, orient='index').
↪reset_index()
  wordcountsDf.columns=['word','count']
  print(wordcountsDf.sort_values('count',ascending=False).head(n))
```

[178]:
```
def createWc(word_counter):
  wc = WordCloud(width=800, height=400)
  wc.generate_from_frequencies(frequencies=word_counter)
  plt.figure(figsize=(10,8))
  plt.imshow(wc)
```

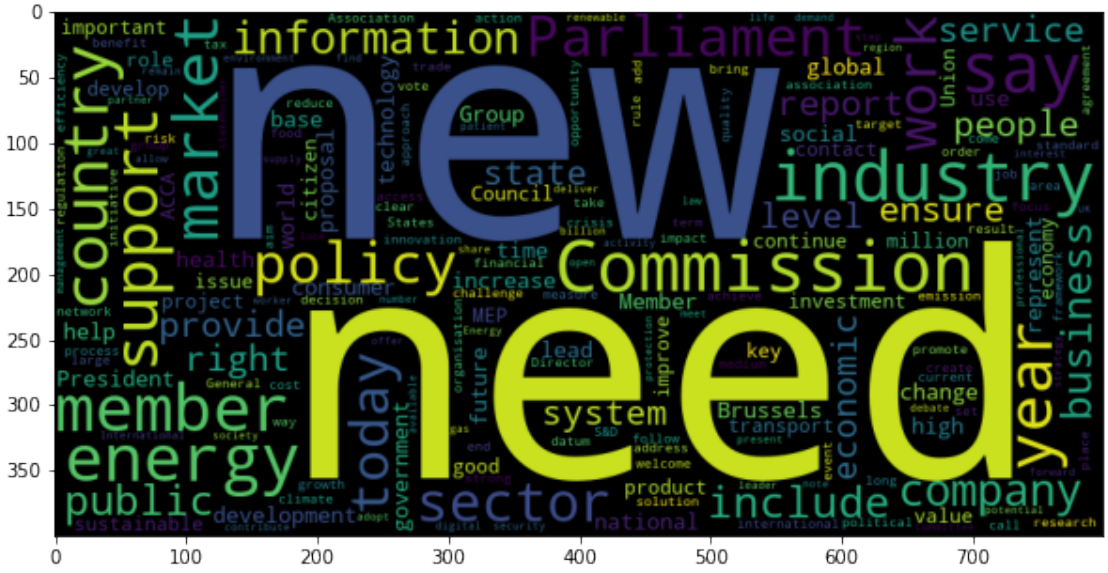
[208]:
```
viewMostCommon(lemmas,10)
```

```
          word   count
49          \n\n  123786
144               44691
```

```
41      European   28132
38            EU   27336
106       Europe   22819
259     european   15377
381          new   12039
20          need   11949
43    Commission   11275
54           say   10738
```

Usuwamy pierwsze 2 wartości i synonimy Europy

[179]:
```
createWc(createCounterDeleteMostCommon(lemmas,6))
```



Tylko rzeczowniki

[87]:
```
nouns = docs.swifter.apply(lambda doc: [token.lemma_
          for token in doc
          if not token.is_stop
          if not token.is_punct
          if token.pos_ == 'NOUN'])
```

```
Pandas Apply:   0%|          | 0/11047 [00:00<?, ?it/s]
```

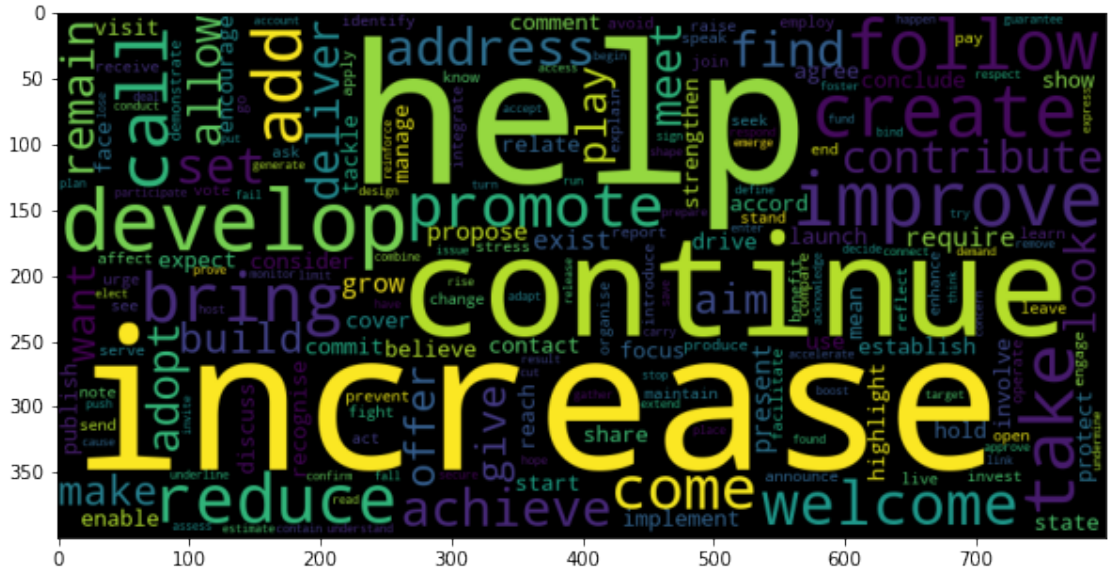
[180]:
```
createWc(createCounterDeleteMostCommon(nouns,10))
```

Czasowniki

[93]:
```
verbs = docs.swifter.apply(lambda doc: [token.lemma_
          for token in doc
          if not token.is_stop
          if not token.is_punct
          if token.pos_ == 'VERB'])
```

```
Pandas Apply:   0%|          | 0/11047 [00:00<?, ?it/s]
```

[181]:
```
createWc(createCounterDeleteMostCommon(verbs,10))
```

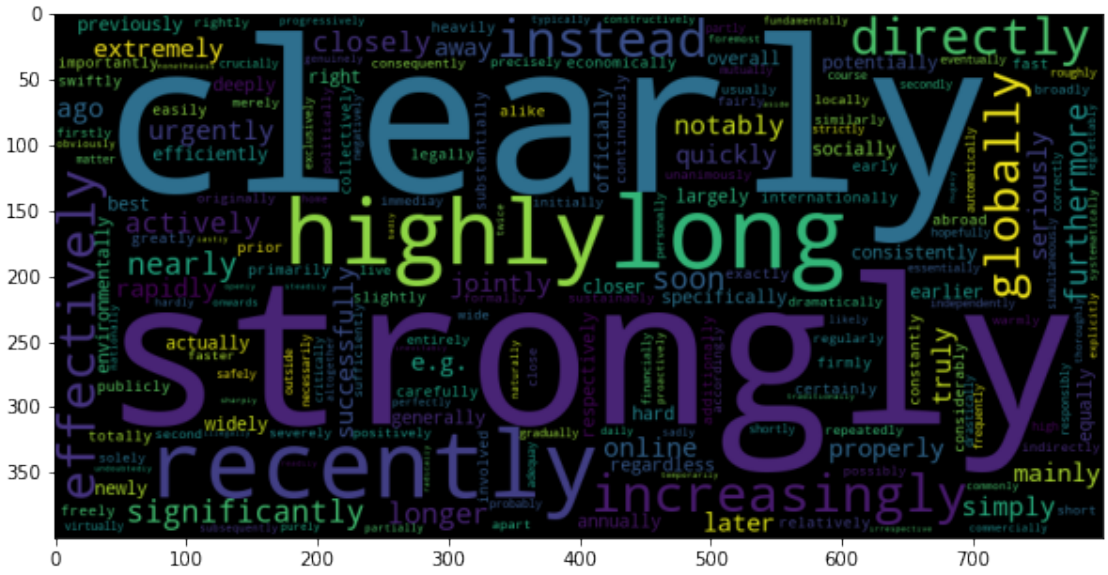


Przysłówki

[98]:
```
adverbs = docs.swifter.apply(lambda doc: [token.lemma_
          for token in doc
          if not token.is_stop
          if not token.is_punct
          if token.pos_ == 'ADV'])
```

```
Pandas Apply:   0%|          | 0/11047 [00:00<?, ?it/s]
```

[182]:
```
createWc(createCounterDeleteMostCommon(adverbs,10))
```



Jak się zmieniały słowa przez lata

[164]:
```
docs_year=pd.DataFrame(zip(docs,list(years)))
docs_year.columns=['doc', 'year']
```

[169]:
```
lemmas2022 = docs_year.loc[docs_year['year']==2021].iloc[:,0].swifter.
↪apply(lambda doc: [token.lemma_ for token in doc if not token.is_stop if not␣
↪token.is_punct])
```

```
Pandas Apply:   0%|          | 0/839 [00:00<?, ?it/s]
```

[170]:
```

```

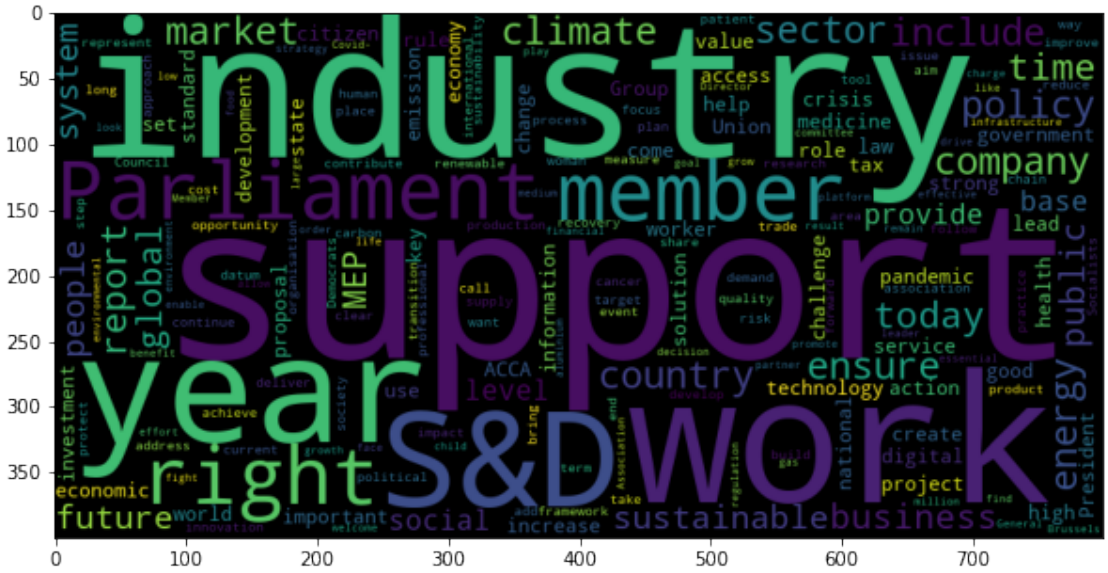
```
lemmas2008 = docs_year.loc[docs_year['year']==2008].iloc[:,0].swifter.
↪apply(lambda doc: [token.lemma_ for token in doc if not token.is_stop if not␣
↪token.is_punct])
```

```
Pandas Apply:   0%|          | 0/428 [00:00<?, ?it/s]
```

[210]:
```
viewMostCommon(lemmas2022,10)
```

	word	count
3	\n\n	7645
73		2918
374	EU	2564
843	European	2111
12	Europe	1942
776	need	1127
755	european	1107
105	new	983
810	say	949
844	Commission	943

[214]:
```
createWc(createCounterDeleteMostCommon(lemmas2022,10))
```

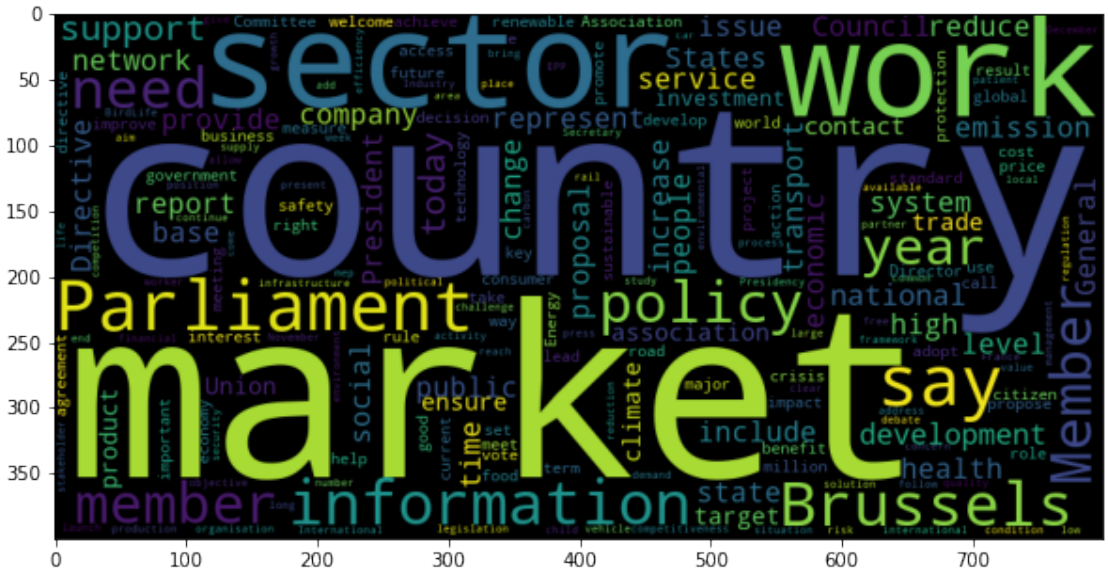


[211]:
```
viewMostCommon(lemmas2008,10)
```

	word	count
42	\n\n	4283
1		1273
102	European	847

```
11           EU    777
93       Europe    613
89     european    498
172  Commission    408
819    industry    359
5        energy    335
416         new    281
```

[215]:
```
createWc(createCounterDeleteMostCommon(lemmas2008,10))
```



1.2 Analiza n-gramów

[ ]:
```
def tokenize(df_text):
    all_tokens = []
    for doc in df_text:
        tokens = [token for token in doc if not token.is_punct and not token.
 ↪is_space]
        all_tokens.append(tokens)
    return all_tokens
```

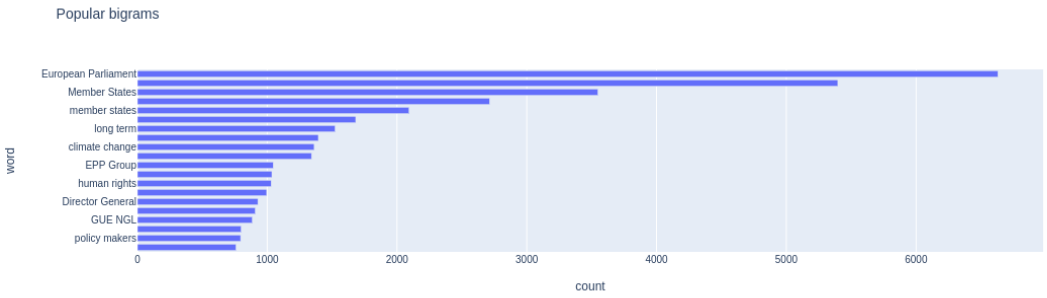
[ ]:
```
def create_bigrams_trigrams(all_tokens):
    all_tokens_flat = [s for S in all_tokens for s in S]
    bgrams = ngrams(all_tokens_flat, 2)
    tgrams = ngrams(all_tokens_flat, 3)
    bigrams = []
    trigrams = []
    for t1, t2 in bgrams:
```

```
        if not t1.is_stop and  not t2.is_stop:
            bigrams_1 = (str(t1), str(t2))
            bigrams.append(bigrams_1)

    for t1, t2, t3 in tgrams:
        if not t1.is_stop and  not t2.is_stop and not t3.is_stop:
            tigrams_1 = (str(t1), str(t2), str(t3))
            trigrams.append(tigrams_1)
    return bigrams, trigrams
```

[ ]:
```
all_tokens = tokenize(docs)
```

[ ]:
```
[bigrams, trigrams] = create_bigrams_trigrams(all_tokens)
```

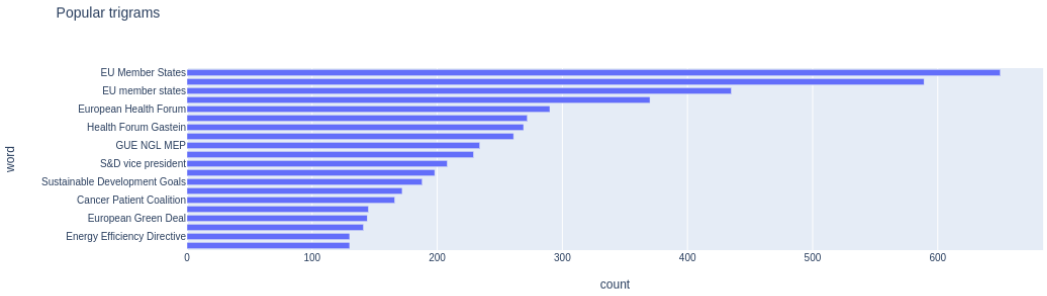
[ ]:
```
counts_bi = pd.DataFrame(Counter(bigrams).most_common(20), columns=['word',␣
↪'count'])
counts_bi.loc[:,'word'] = counts_bi.loc[:,'word'].apply(lambda x: str(x[0])+'␣
↪'+ str(x[1]))
fig = px.bar(counts_bi,orientation='h', y='word', x='count', title = 'Popular␣
↪bigrams')

fig['layout']['yaxis']['autorange'] = "reversed"
fig.update_layout(bargap=0.30, font={'size':10})
#img_bytes = fig.to_image(format="png")
#Image(img_bytes)
fig
```



Najpopularniejsze bigramy dotyczą Unii Europejskiej- parlamentu, komisji, krajów członkowskich oraz poszczególnych stanowisk. Pozostałe: - `S&D Group` - Postępowy Sojusz Socjalistów i Demokratów w Parlamencie Europejskim, socjaldemokratyczna grupa polityczna w Parlamencie Europejskim VII kadencji, powołana faktycznie 23 czerwca 2009 - `EPP Group` - Grupa Europejskiej Partii Ludowej (Chrześcijańscy Demokraci), Od 1999 (tj. od V kadencji) jest największą grupą polityczną w PE (parlamencie). (Na końcu każdej z notek organizacji znajduje się nazwa -`EPP Group`, zatem w rzeczywistości wystąpień tego słowa jest o ok. 450 mniej) - `long term` - praw-

dopodobnie w dokumentach dużo jest mowy o planach długoterminowych - `Secretary General` - sekretarz generalny - `climate change`, `energy efficiency` - kwestie klimatyczne i oszczędności energii najwyraźniej również były w dokumentach ważne - `human rights`

[ ]:
```
counts_tri = pd.DataFrame(Counter(trigrams).most_common(20), columns=['word',␣
↪'count'])
counts_tri.loc[:,'word'] = counts_tri.loc[:,'word'].apply(lambda x: str(x[0])+'␣
↪'+ str(x[1])+ ' '+ str(x[2]))
fig = px.bar(counts_tri,orientation='h', y='word', x='count', title = 'Popular␣
↪trigrams')

fig['layout']['yaxis']['autorange'] = "reversed"
fig.update_layout(bargap=0.30, font={'size':10})
#img_bytes = fig.to_image(format="png")
#Image(img_bytes)
fig
```

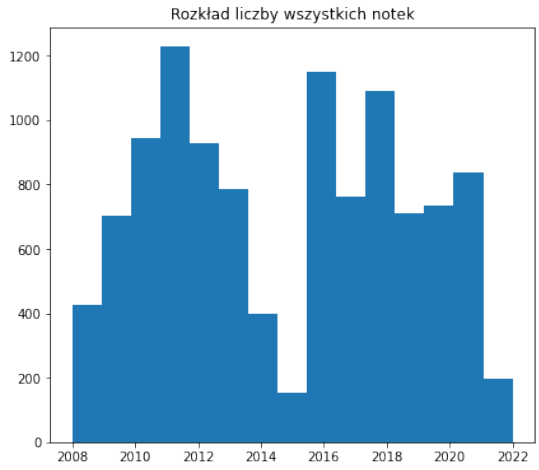


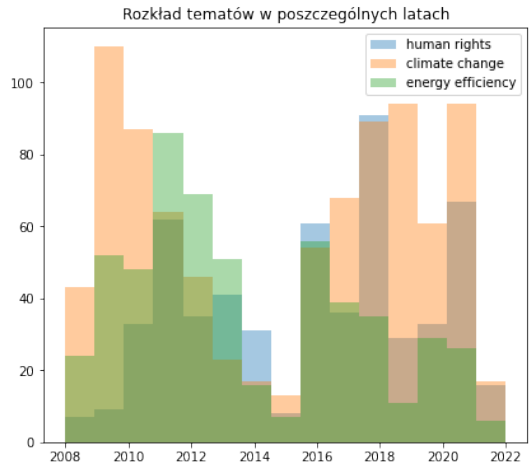
Najpopularniejsze trigramy: - `Chartered Certified Accountants`, zapewne od 'Association of Chartered Certified Accountants' - ACCA, czyli organizacji, która jest na II miejscu w liczebności publikowanych dokumentów - `EU member states`, jak wyżej - dużo mowy o sprawach UE - `European Health Forum Gastein` - wiodąca konferencja na temat polityki zdrowotnej w Europie

[ ]:
```
interesting = ['human rights', 'climate change', 'energy efficiency'] #␣
↪zliczamy dokumenty, w których te tematy wystąpiły

fig, ax = plt.subplots(1, 2, figsize = (15, 6))

ax[0].hist(df.date.apply(lambda x: x.year), bins = 15)
ax[0].set_title("Rozkład liczby wszystkich notek")

for i in range(len(interesting)):
    is_present = df.text.apply(lambda x: interesting[i] in x)
    years = df.date[is_present].apply(lambda x: x.year)
    plt.hist(years, alpha = 0.4, label = interesting[i], bins = 15)
```

```
    plt.legend(loc='upper right')
plt.title("Rozkład tematów w poszczególnych latach")
```

[ ]:
```
Text(0.5, 1.0, 'Rozkład tematów w poszczególnych latach')
```





W roku 2015 nastąpiło wyraźne obniżenie liczby publikowanych dokumentów. Może coś było nie tak ze stroną? (TODO). Obniżenie liczności nastąpiło również w roku 2022, ponieważ jesteśmy w jego trakcie. Możemy szacować, że skoro w ciągu dwóch pierwszych miesięcy roku 2022 opublikowanych zostało około 200 dokumentów, to w ciągu całego roku będzie to około 200*6=1200. Taki wynik byłby jednym z wyższych na przestrzni tych lat. Prawdopodobnie wynika to z trwającej wojny. Pik w roku 2011 mógł wynik z kilku ważnych wydarzeń owego roku: beatyfikacja Jana Pawła II, trzęsienie ziemi i katastrofa nuklearna w Japonii, zabicie bin Ladena, arabska wiosna, interwencja NATO w Libii, masakra na norweskiej wyspie Utoya, kryzys w straefie euro - to najważniejsze wydarzenia roku 2011 na świecie.

Na podstawie drugiego wykresu możemy wnioskować, że kluczowym tematem na przestrzeni lat były zmiany klimatyczne. Rok 2009 ?

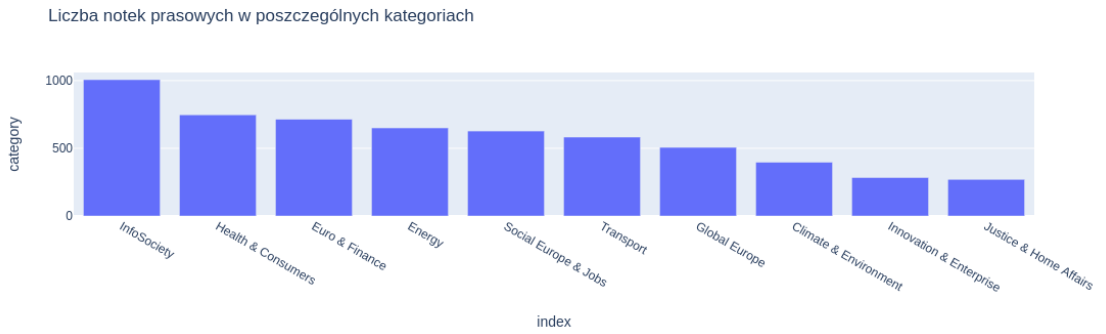
1.2.1 Analiza ngramów z podziałem na kategorie

[ ]:
```
popular_cat = list(df.category.value_counts().index)[0:10]
popular_cat
```

[ ]:
```
['InfoSociety',
 'Health & Consumers',
 'Euro & Finance',
 'Energy',
 'Social Europe & Jobs',
 'Transport',
 'Global Europe',
 'Climate & Environment',
```

```
 'Innovation & Enterprise',
 'Justice & Home Affairs']
```

[ ]:
```
is_popular_org = df.category.apply(lambda x: x in popular_cat)
pop_org_text = pd.DataFrame(np.array(df)[is_popular_org]).iloc[:,[1,4]]
pop_org_text.columns = ['text', 'category']
categories_counted = pop_org_text.category.value_counts().reset_index()
px.bar(categories_counted, x = 'index', y = 'category', title ='Liczba notek␣
↪prasowych w poszczególnych kategoriach')
```



Najczęściej poruszanym działem tematycznym jest `InfoSocieity`, czyli najwięcej jest dokumentów informacyjnych.

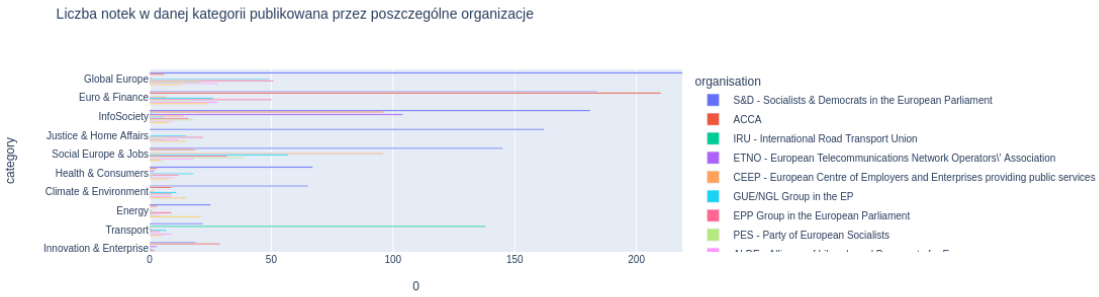
[ ]:
```
popular_orgs = list(df.organisation.value_counts().index)[0:10]
perc = round(len(df.loc[df.organisation.apply(lambda x: x in popular_orgs)]) /␣
↪len(df) * 100, 2)
print(f"Zauważmy, że ok {perc}% tekstów zostało opublikowaych przez 10␣
↪najpopularniejszych organizacji - w dalszej eksploracji skupimy się głównie␣
↪na nich.")
popular_orgs
```

```
Zauważmy, że ok 42.93% tekstów zostało opublikowaych przez 10
najpopularniejszych organizacji-w dalszej eksploracji skupimy się głównie na
nich.
```

[ ]:
```
['S&amp;D - Socialists &amp; Democrats in the European Parliament',
 'ACCA',
 'EPP Group in the European Parliament',
 'GUE/NGL Group in the EP',
 'THE GREENS/EUROPEAN FREE ALLIANCE',
 'ALDE - Alliance of Liberals and Democrats for Europe',
 'IRU - International Road Transport Union',
 'PES - Party of European Socialists',
 'CEEP - European Centre of Employers and Enterprises providing public
```

```
services',
 "ETNO - European Telecommunications Network Operators\\' Association"]
```

[ ]:
```
df_category_org = df.loc[df.category.apply(lambda x: x in popular_cat)]
df_category_org = df_category_org.loc[df_category_org.organisation.apply(lambda␣
 ↪x: x in popular_orgs)]
df_category_org = df_category_org.groupby(['category', 'organisation']).
 ↪value_counts().reset_index()
df_category_org = df_category_org.iloc[:,0:2]
df_category_org = pd.DataFrame(df_category_org.value_counts().reset_index())

fig = px.bar(df_category_org, x = 0, y = 'category', color='organisation',␣
 ↪barmode='group', width = 1015, title='Liczba notek w danej kategorii␣
 ↪publikowana przez poszczególne organizacje')
fig['layout']['yaxis']['autorange'] = "reversed"
fig.update_layout(bargap=0.30, font={'size':10})
fig
```



Obserwacje: 1. Swój największy udział w dziedzinach `Climate & Environment`, `Health & Consumers`, `Social Europe & Jobs`, `Justice & Home Affairs`, `InfoSociety` oraz `Global Europe` miała S&D - Socialists & Democrats in the European Parliament (partia lewicowa), co zgadzałoby się z tematami zainteresowań partii właśnie lewicowej.

Nie zauważyłam żadnych odstępstw - organizacje zdają się przestrzegać wcześniej powziętej misji zajmowania się konkretnymi tematami ze swojego programu.

[ ]:
```
text_by_categories = pd.DataFrame(pop_org_text.groupby('category').text.sum())
```

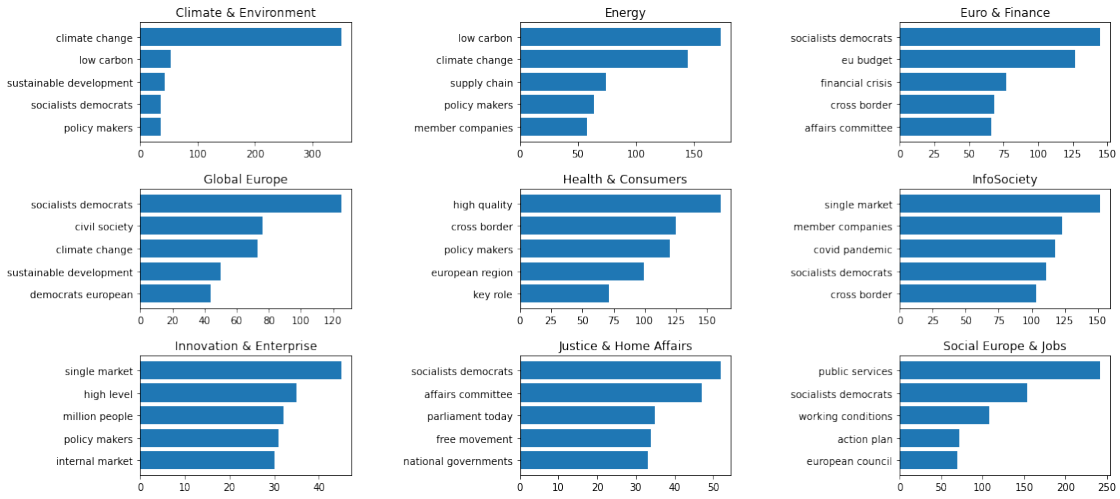
[ ]:
```
def count_vectorize(text, names = [], ngram_range = (2, 2)):
    count_vectorizer = CountVectorizer(min_df = 10, stop_words='english',␣
 ↪token_pattern = r"[a-zA-Z]{2,}", ngram_range=ngram_range)
    count_cat = count_vectorizer.fit_transform(text)
    count_feature_names = count_vectorizer.get_feature_names()
```

```
    df_count_org = pd.DataFrame(count_cat.toarray(),␣
↪columns=list(count_feature_names))
    if names is not []:
        df_count_org.index = names
    return df_count_org
```

[ ]:
```
def plot_ngrams_by_categories(df_org, drop_list =[]):
    fig, ax = plt.subplots(3, 3, figsize = (15, 7))

    categories = ['Climate & Environment', 'Energy', 'Euro & Finance', 'Global␣
↪Europe',
          'Health & Consumers','InfoSociety', 'Innovation & Enterprise',
          'Justice & Home Affairs', 'Social Europe & Jobs', 'Transport']
    a = 0
    for i in range(3):
        for j in range(3):
            df_count_org = copy.deepcopy(df_org)
            df_count_org_loc = df_count_org.loc[categories[a]]
            df_count_org_loc = df_count_org_loc.drop(drop_list)
            counted = df_count_org_loc.sort_values(ascending = False)[0:5]
            counted = counted.sort_values()
            ax[i, j].barh(counted.index, counted)
            ax[i, j].set_title(categories[a])
            a = a+1
    plt.subplots_adjust(left=0.0,bottom=0.1, right=0.9, top=1, wspace=0.8,␣
↪hspace=0.4)
```

[ ]:
```
bigrams_categories = count_vectorize(text_by_categories.text,␣
↪text_by_categories.index)
```

[ ]:
```
to_drop = ['member states', 'european commission', 'european parliament',␣
↪'european union', 'eu member', 'secretary general',
         'european countries', 'vice president', 'gue ngl', 'europa eu',␣
↪'member state', 'group european', 'long term', 'information contact']
plot_ngrams_by_categories(bigrams_categories, to_drop)
```



1. `Health & Consumers` - najczęściej pojawiającym się bigramem jest `high quality`. Prawdopodobnie dotyczy on opisu wysokiej jakości produktów i usług.

@TODO

1.3 Tf-idf + wizualizacja (wagi słów)

[ ]:
```
def tfidf_vectorize(text_to_vectorize, popular_org_names, ngram_range=(1,1),␣
↪min_df = 7, popular = True, drop_list = []):
    tfidf_vectorizer_org = TfidfVectorizer(min_df = min_df, use_idf=True,␣
↪stop_words='english', token_pattern = r"[a-zA-Z]{2,}",␣
↪ngram_range=ngram_range)
    tfidf_org_pop = tfidf_vectorizer_org.fit_transform(text_to_vectorize)
    tfidf_feature_names = tfidf_vectorizer_org.get_feature_names()
    df_tfidf_org = pd.DataFrame(tfidf_org_pop.toarray(),␣
↪columns=list(tfidf_feature_names))
    df_tfidf_org.index = popular_org_names
    if drop_list:
        df_tfidf_org = df_tfidf_org.drop(drop_list, axis = 1)
    if popular:
        popular_words = list(df_tfidf_org.max(axis=0).sort_values(ascending =␣
↪False)[0:20].index)
        df_tfidf_org = df_tfidf_org.loc[:, popular_words]

    return df_tfidf_org
```

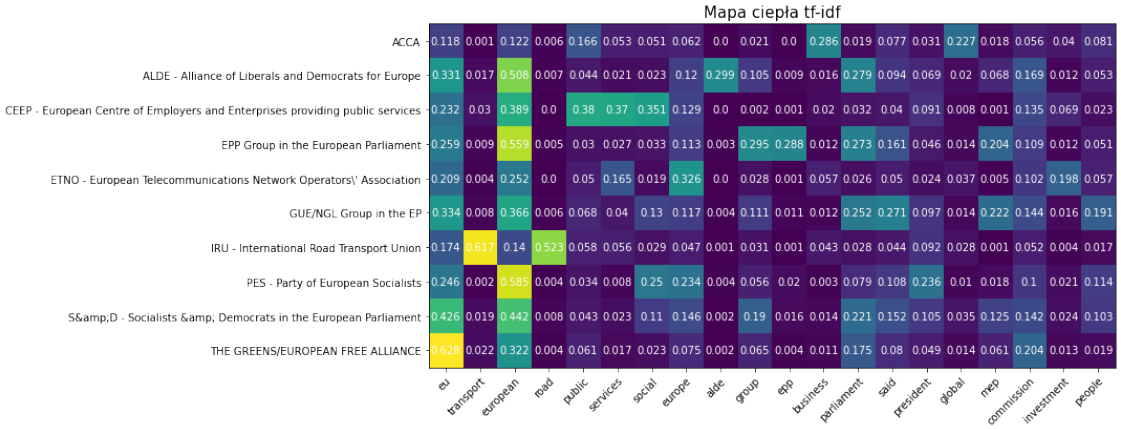
[ ]:
```
is_popular_org = df.organisation.apply(lambda x: x in popular_orgs)
pop_org_text = pd.DataFrame(np.array(df)[is_popular_org]).iloc[:,1:3]
pop_org_text.columns = ['text', 'organisation']
```

```
text_by_organisations = pd.DataFrame(pop_org_text.groupby('organisation').text.
↪sum())
```

[ ]:
```
df_tfidf_org = tfidf_vectorize(text_by_organisations.text,␣
↪list(text_by_organisations.index))
df_tfidf_bi = tfidf_vectorize(text_by_organisations.text,␣
↪list(text_by_organisations.index), ngram_range = (2, 2))
```

[ ]:
```
def plot_heatmap(df_tfidf_org, title = "Mapa ciepła tf-idf", fig_size = (15,␣
↪20)):
    orgs = df_tfidf_org.index
    words = df_tfidf_org.columns

    fig, ax = plt.subplots(figsize = fig_size)
    im = ax.imshow(df_tfidf_org)

    ax.set_xticks(np.arange(len(words)), labels=words)
    ax.set_yticks(np.arange(len(orgs)), labels=orgs)

    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
             rotation_mode="anchor")
    for i in range(len(orgs)):
        for j in range(len(words)):
            text = ax.text(j, i, round(df_tfidf_org.iloc[i, j],3),
                           ha="center", va="center", color="w")
    ax.set_title(title, fontsize = 15)
    fig.tight_layout()
    plt.show()
```

[ ]:
```
plot_heatmap(df_tfidf_org)
```

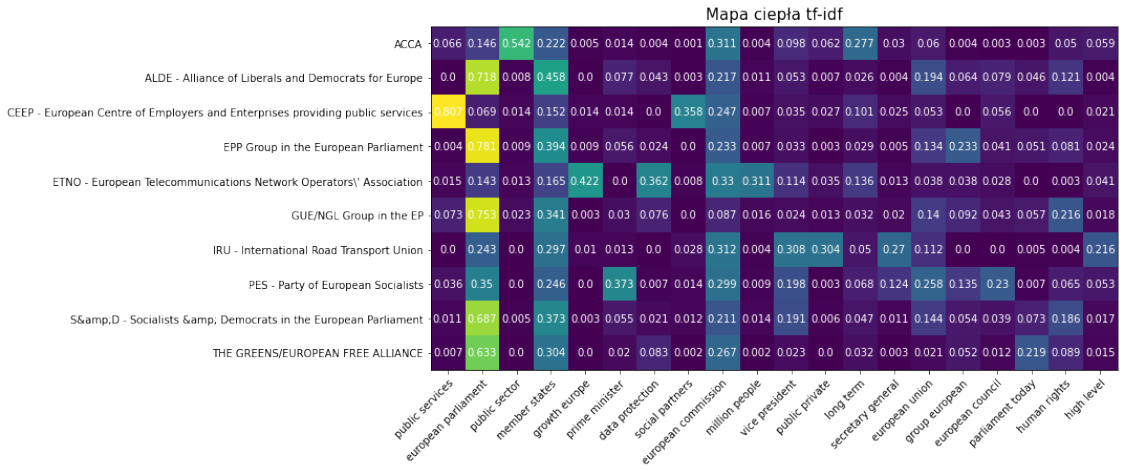


Charakterystyka słów dla: - `S&D Group` ważne słowa: `buisness` - `ACCA` ważne słowa: `eu`, `european`,

`ALDE`- Porozumienie Liberałów i Demokratów na rzecz Europy (frakcja w Parlamencie Europejskim) - `IRU` ważne słowa: `transport`, `road` - ma sens, bo to organizacja transportowa itd.

Generalnie najważniejsze dla większości organizacji były: `eu`, `european`.

[ ]:
```
plot_heatmap(df_tfidf_bi)
```



Ważkość bigramów: - `ACCA` ważne bigramy: `european parliament`, `information contact`

Generalnie najważniejsze bigramy to: `european parliament`, `member states`, `european commission`, co potwierdza uprzednio pokazaną liczność tychże słów w dokumentach.

TODO: tutaj i w poprzednim można by więcej opisać.

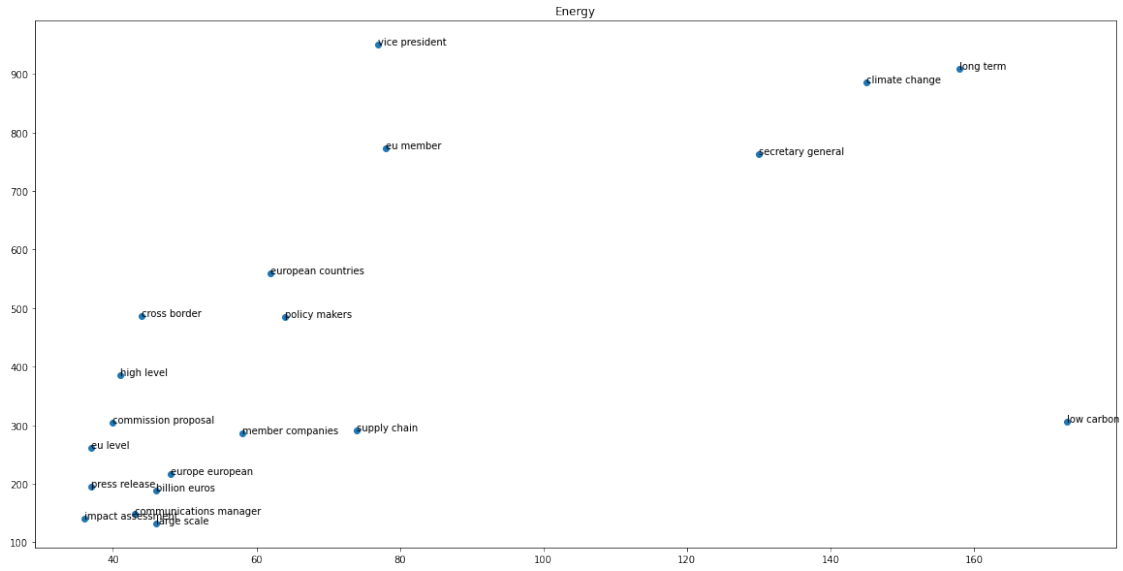
1.3.1 Analiza ze względu na lata i kategorie

Aby skupić się na analizie słów istotnych dla danego obszaru tematycznego, dodaliśmy wykres z liczbą wystąpień w danej kategorii na osi x oraz liczbą wystąpień ogółem na osi y.
Mapa ciepła (funkcja `tfidf_vectorize`) ma dodatkowy argument - `drop_list`, do której można dodawać wyrażenia, które są wspólne dla większości kategorii.

[ ]:
```
bigrams_categories = bigrams_categories.drop(['european parliament', 'member␣
↪states', 'european commission', 'european union', 'information contact'],␣
↪axis = 1)
```

[ ]:
```
def check_importance(df_count_org, num):
    """Kolejność alfabetyczna: 0-'Climate & Environment', 1-'Energy', 2-'Euro &␣
↪Finance', 3-'Global Europe', 4-'Health & Consumers'
    5-'InfoSociety', 6-'Innovation & Enterprise', 7-'Justice & Home Affairs',␣
↪8-'Social Europe & Jobs', 9-'Transport'"""

    text_freq = df_count_org.iloc[num,:].sort_values(ascending = False)[0:20]
    labels = list(text_freq.index)
```

```
    document_freq = df_count_org.sum(axis = 0).loc[labels]

    fig, ax = plt.subplots(figsize = (20, 10))
    plt.scatter(text_freq, document_freq)
    plt.title(df_count_org.index[num])

    for i, txt in enumerate(labels):
        ax.annotate(txt, (text_freq[i], document_freq[i]))

check_importance(bigrams_categories, 1)
```



Z wykresu można odczytać, że istotnie (jak to wynikało już z bigramów) dla kategorii **Energy** charakterystycznym wyrażeniem jest **low carbon**, czyli **niska emisja**. Świadczy to o podejmowaniu tematu oszczędności źródeł energii. Ma to sens ze względu na ograniczenia zasobów występujących na Ziemi.

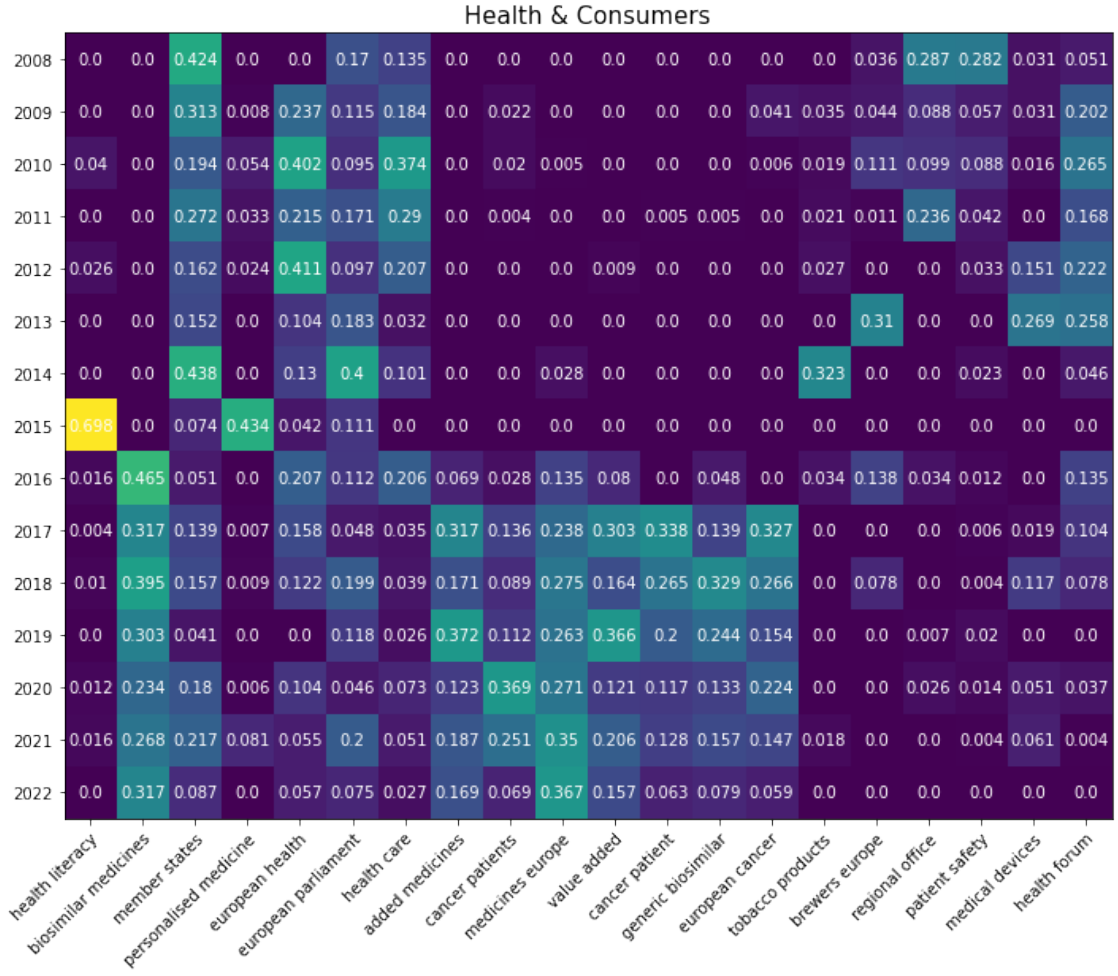
Jest również podejmowany temat zmian klimatycznych, jedakże **climate change** jest wyrażeniem charakterystycznym również wśród innych kategorii.

[ ]:
```
def tfidf_categories(number):
    df_one_cat = df.loc[df.category == popular_cat[number]]
    df_one_cat = df_one_cat.iloc[:,[1,5]]
    text_by_cat_years = pd.DataFrame(df_one_cat.groupby('year').text.sum())
    return text_by_cat_years
```

universal_drop_list = [‘european parliament’, ‘member states’, ‘european commission’, ‘european union’] text_by_cat_years = tfidf_categories(0) df_tfidf_cat = tfidf_vectorize(text_by_cat_years.text, list(text_by_cat_years.index), ngram_range = (2, 2), drop_list = universal_drop_list) plot_heatmap(df_tfidf_cat, title = popular_cat[0], fig_size

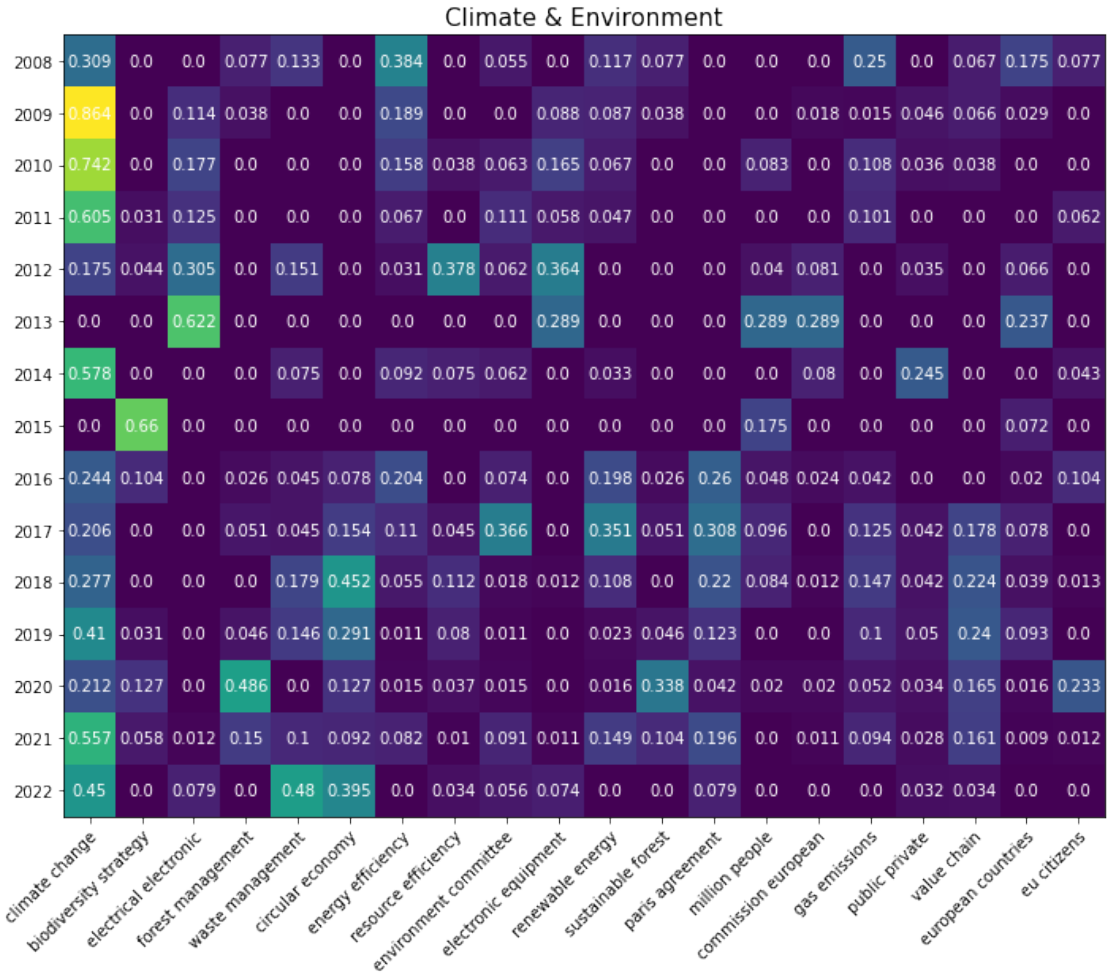
= (10, 10))

```
[ ]: text_by_cat_years = tfidf_categories(1)
df_tfidf_cat = tfidf_vectorize(text_by_cat_years.text, list(text_by_cat_years.
 ↪index), ngram_range = (2, 2))
plot_heatmap(df_tfidf_cat, title = popular_cat[1], fig_size = (10, 10))
```



Ciekawe: - Temat wiedzy o zdrowiu (`health literacy`) w latach 2008-2014 oraz 2016-2022 był praktycznie nieobecny w dokumentach, a w roku 2015 nagle odnotowano duży wzrost. Jak to wyjaśnić? Nie wiadomo. - O `biosimilar medicines` (biologic medical product that is almost an identical copy of an original product that is manufactured by a different company) zaczęto pisać od 2016 roku. Pierwszy biosimilar został zatwierdzony przez UE w 2015r. (jako jedyny w tymże roku), w następnych latach corocznie zatwierdzano ich co niemiara. - Wzmożone zainteresowanie `tobacco products` nastapiło w 2014 r. Wtedy też został wydany dokument regulujący prawa związane z wytwarzaniemm prezentowaniem i sprzedażą wyrobów tytoniowych i wyrobów pokrewnych wśród państw członkowskich. - ...

```
[ ]: drop_list = ['european parliament', 'member states', 'european commission',␣
     ↪'european union', 'europa eu', 'eu member']
     text_by_cat_years = tfidf_categories(7)
     df_tfidf_cat = tfidf_vectorize(text_by_cat_years.text, list(text_by_cat_years.
     ↪index), ngram_range = (2, 2), drop_list = drop_list)
     plot_heatmap(df_tfidf_cat, title = popular_cat[7], fig_size = (10, 10))
```



Wnioski: - w roku 2020 charakterystycznym tematem była **gospodarka leśna** - wtedy miał miejsce pożar Amazonii - temat **zmian klimatycznych** był najbardziej popularny w latach 2009-2011 oraz 2021-2022. - w roku 2016 popularna była strategia utrzymania bioróżnorodności - miało to związek z planem działań na lata 2016-2020. - **Porozumienie paryskie** – porozumienie wieńczące 21 Konferencję ONZ w sprawie zmian klimatu. Porozumienie zobowiązuje wszystkie kraje do przedstawienia do 2020 roku długoterminowych scenariuszy ograniczenia emisji gazów cieplarnianych zgodnie z metodologią przyjętą przez IPCC.

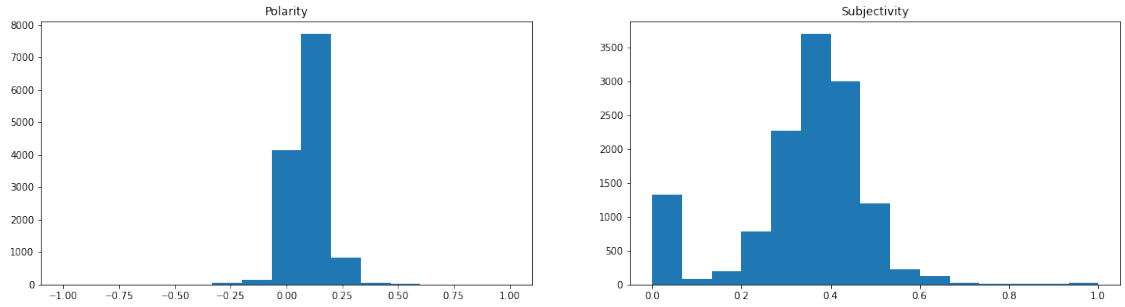
1.4 Analiza sentymentu

[ ]:
```
df_extended = copy.deepcopy(df)
```

[ ]:
```
def polarity(text):
    return TextBlob(text).sentiment.polarity

def subjectivity(text):
    return TextBlob(text).sentiment.subjectivity

df_extended['polarity'] = df_extended['text'].apply(lambda x : polarity(x))
df_extended['subjectivity'] = df_extended['text'].apply(lambda x :␣
↪subjectivity(x))
```

- polarity [-1,1]: -1 defines a negative sentiment and 1 defines a positive sentiment
- subjectivity [0,1]: quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information

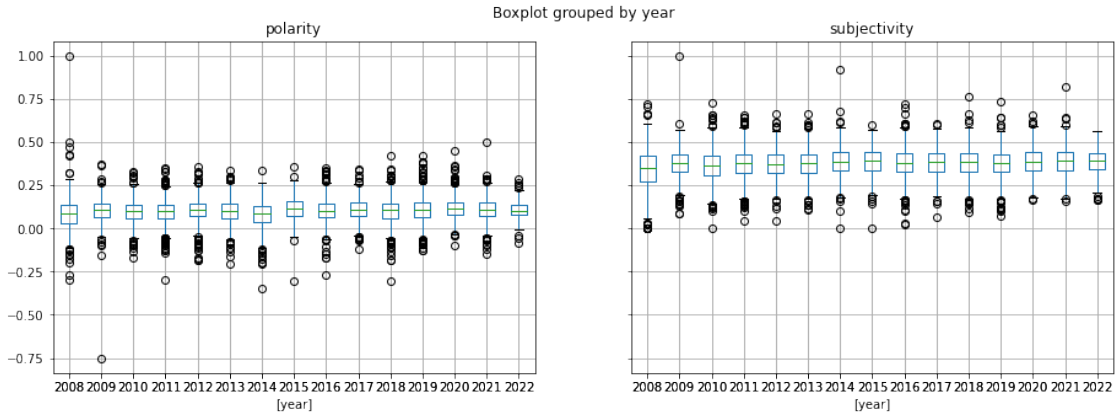
[ ]:
```
fig, ax = plt.subplots(1,2, figsize=(20, 5))
ax[0].hist(df_extended['polarity'], bins = 15)
ax[0].set_title("Polarity")
ax[1].hist(df_extended['subjectivity'], bins = 15)
ax[1].set_title("Subjectivity")
plt.show()
```



- Większość tekstów jest neutralna z lekkim przeważeniem w stronę pozytywną (na poziomie ok. 0.13).
- Najwięcej jest tekstów subiektywnych na poziomie ok. 0.4. To znaczy stosunek opinii do faktów występujących w tekstach jest mniej więcej na poziomie 0.4. To znaczy, że w tesktach możemy znaleźć średnio więcej faktów niż subiektywnej oceny, co dodaje tekstom wiarygodności.

[ ]:
```
df_extended['year'] = df_extended['date'].apply(lambda x: x.year)
```

[ ]:
```
df_extended.boxplot(by = 'year', figsize = (15, 5))
plt.show()
```



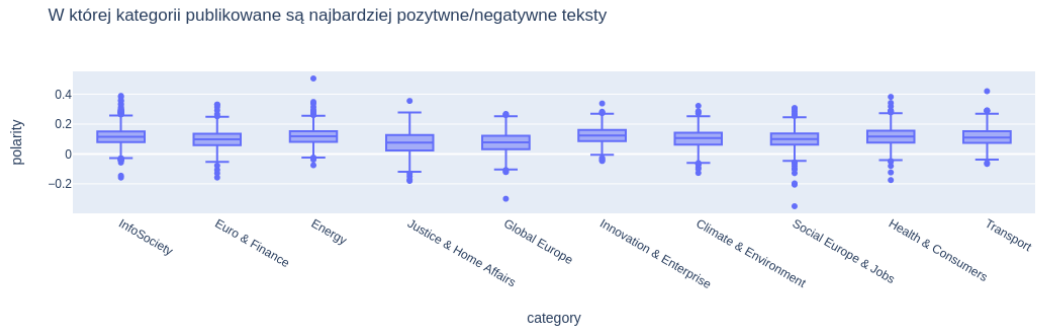
`Polarity` utrzymuje się na podobnym poziomie na przestrzeni lat. Wartości rozrzucone są blisko 0, ale tuż nad nim. Co roku również pojawiało się kilka-kilkanaście dokumentów jako obserwacji odstających - tzn. były albo wyjątkowo pozytywne, albo wyjątkowo negatywne. We wczesnych latach (2008-2009) pojawiły się 2 skrajne dokumenty (1.00 i -0.75), obecnie jednak zachowana jest konwencja pozytywności bliskiej neutralności.

`Subjecetivity` również utrzymuje się na podobnym poziomie. Co ciekawe, w przeszłości (2008, 2010, 2014,2015) pojawiały się dokumenty zupełnie pozbawione subiektywności (na poziomie 0.00) oraz w roku 2009 odkument w pełni subiektywny (na poziomie 1.00).

[ ]:
```
df_sentiment_grouped = df_extended.groupby('organisation').mean().
↪sort_values('subjectivity', ascending = False).reset_index()
```

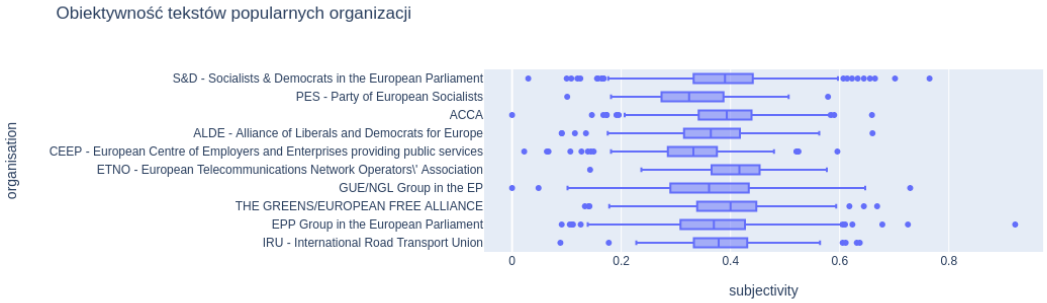
[ ]:
```
df_sentiment_grouped_loc = df_extended.loc[df_extended.category.apply(lambda x:␣
↪x in popular_cat)]
df_sentiment_grouped_loc = df_sentiment_grouped_loc.iloc[:,[4,6,7]]

df_sentiment_grouped_loc_org = df_extended.loc[df_extended.organisation.
↪apply(lambda x: x in popular_orgs)]
df_sentiment_grouped_loc_org = df_sentiment_grouped_loc_org.iloc[:,[2,6,7]]
```

[ ]:
```
fig = px.box(df_sentiment_grouped_loc, x="category", y="polarity", title = "W␣
↪której kategorii publikowane są najbardziej pozytwne/negatywne teksty")
fig
```



We wszystkich kategoriach poziom `polarity` jest bardzo podobny, choć po ręcznym zbadaniu median w każdej z nich minimalnie wyższe wyniki uzyskaliśmy dla kategorii `Innovation & Enterprise` - możemy ją traktować zatem jako pozytywną bardziej niż pozostałe. Minimalnie niższą medianę uzyskała kategoria `Justice & Home Affairs` - możemy ją więc traktować jako nieco bardziej negatywną. Eksperyment ten, mimo drobnych różnic w wartościach polarity, potwierdza intuicję mówiącą, że sądownictwo często wzbudza więcej negatywnych emocji.

[ ]:
```
fig = px.box(df_sentiment_grouped_loc_org, y="organisation", x="subjectivity",␣
↪title = 'Obiektywność tekstów popularnych organizacji')
fig['layout']['yaxis']['autorange'] = "reversed"
fig
```



Konwencja podobnych poziomów subiektywności jest zachowana również z podziałem na organizacje.

1.5 Nazwy własne - named entities

[ ]:
```
def tokenize_lemmatize(df_text):
    #standard_tokens = []
    standard_docs = []
    for text in df_text:
```

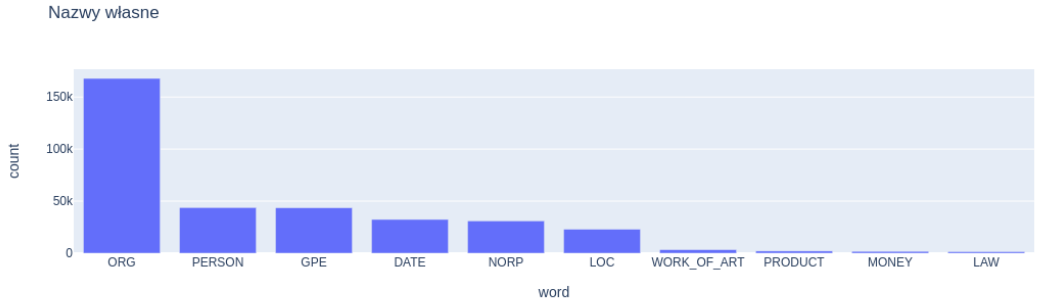
```
        doc = en(text)
        #tokens = [token for token in doc if not token.is_punct]
        #standard_tokens.append(tokens)
        standard_docs.append(doc)
    return standard_docs
```

[ ]:
```
def label_entities(docs_speeches):
    ent_labels = [] #lista zawierająca etykiety nazw własnych
    all_ents = [] #lista nazw własnych
    for doc in docs_speeches:
        entities = doc.ents
        ent_label = [ent.label_ for ent in entities if not ent.label_ ==␣
 ↪'ORDINAL' and not ent.label_ == 'CARDINAL']
        entity = [ent for ent in entities if not ent.label_ == 'ORDINAL' and␣
 ↪not ent.label_ == 'CARDINAL']
        ent_labels.append(ent_label)
        all_ents.append(entity)
    return all_ents, ent_labels
```

[ ]:
```
[all_ents, ent_labels] = label_entities(docs)
```

- ORG organizacje
- PERSON osobistości
- GPE kraje, miasta, stany
- DATE daty
- NORPnarodowości, religie, grupy polityczne
- LOC góry, rzeki, kontynenty
- WORK_OF_ART
- PRODUCT
- MONEY waluty
- LAW dokumenty prawne

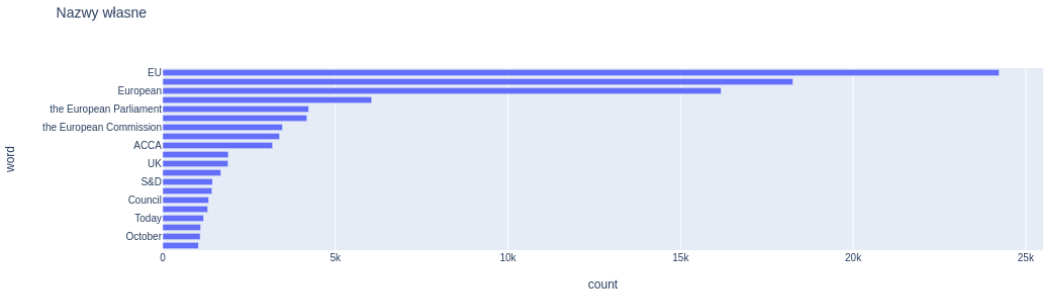
[ ]:
```
counted_ents = Counter(list(itertools.chain(*ent_labels))).most_common(10)
counts = pd.DataFrame(counted_ents, columns=['word', 'count']).
 ↪sort_values('count', ascending = False)
fig = px.bar(counts, x = 'word', y = 'count', title = 'Nazwy własne')
#img_bytes = fig.to_image(format="png")
#Image(img_bytes)
fig
```



Na tak dużą liczność nazw własnych organizacji `ORG` wpływ ma to, że teksty zawierają dużo odniesień do organizacji, przez które zostały napisane.

Zaskakująca może być liczność `LOC`, czyli nazw gór, rzek i kontynentów. W dokumentach prawdopodobnie nie ma za wiele słów dotyczących dwóch pierwszych obiektów, ale raczej dotyczą one kontynentów.

[ ]:
```
ents = [str(s) for S in all_ents for s in S]
counted_ents = Counter(ents).most_common(20)
counts = pd.DataFrame(counted_ents, columns=['word', 'count']).
 ↪sort_values('count', ascending = False)
fig = px.bar(counts, x = 'count', y = 'word', title = 'Nazwy własne',␣
 ↪orientation = 'h')

fig['layout']['yaxis']['autorange'] = "reversed"
fig.update_layout(bargap=0.30, font={'size':10})
#img_bytes = fig.to_image(format="png")
#Image(img_bytes)
fig
```



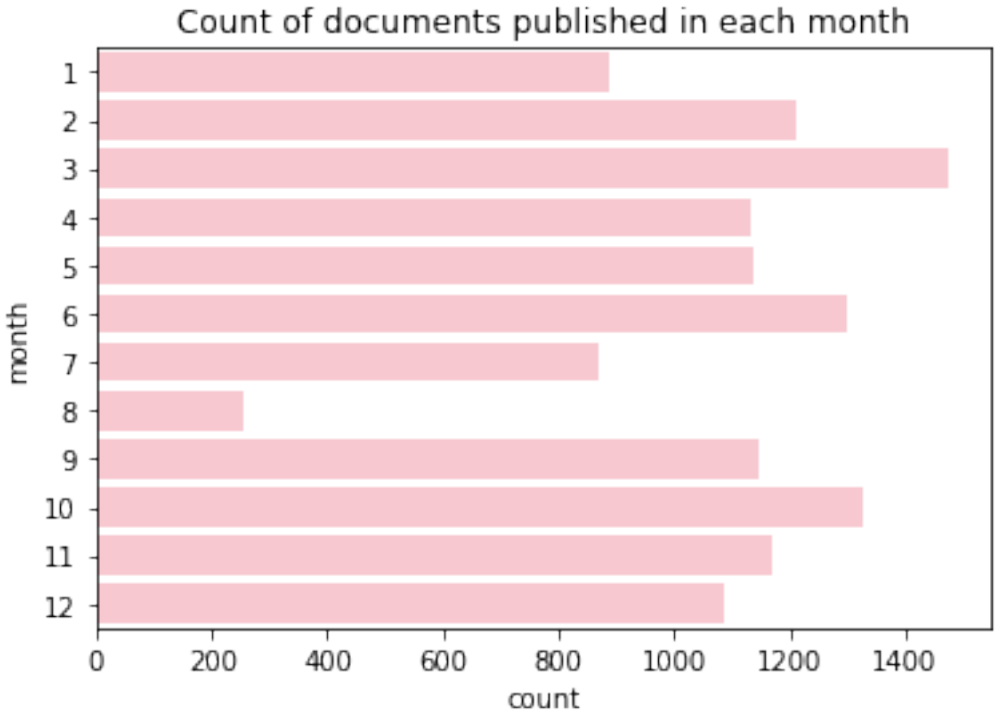
Wykres potwierdza wcześniejszą hipotezę nt. liczności `LOC` - drugą najczęstszą nazwą własną jest `Europe`, czyli nazwa kontynentu. Co dziwne, algorytm za nazwę własną uznał `today` - pewnie

zaliczył je do kategorii `DATE`.

Kontynentami/krajami/miastami, do których najczęściej odnoszą się dokumenty, są: Europa, Bruksela, UK, Germany, US, Francja. Świadczy to o ich ważkości wśród wszystkich innych.

Co ciekawe, na wykresie pojawiło się również słowo `October`, jakoby miało duże znaczenie, ale być może wynika to z tego, że w tymże miesiącu publikowane jest najwięcej dokumentów i w ich treści pojawia się to słowo jako miesiąc publikacji. Sprawdźmy to.

[ ]:
```
df['month'] = df.date
df.month = df.month.apply(lambda x: x.month)
sns.countplot(y=df.month, data=df, color='pink')
plt.title('Count of documents published in each month')
plt.show()
```

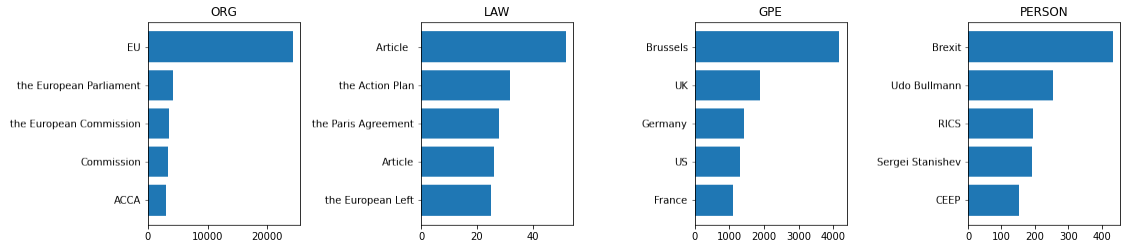


Rzeczywiście **październik** (10) jest miesiącem, w którym opublikowanych zostało wyjątkowo dużo artykułów - znajduje się na II miejscu pod tym względem. Nie wyjaśnia to niestety dlaczego w takim razie częściej występującym miesiącem (jako nazwa własna) nie jest **marzec**, który jest zdecydowanym liderem pod kątem liczby publikacji.

1.5.1 Nazwy własne z podziałem na organizacje, akty prawne, miejsca oraz osoby

[ ]:
```
def plot_entities(all_ents, ent_labels):
    ent_names = ['ORG','LAW','GPE','PERSON']
    fig, ax = plt.subplots(1, 4)
    fig.set_size_inches(15, 3.5)

    for j in range(4):
        is_org = list(map(lambda x: x == ent_names[j], list(itertools.
↪chain(*ent_labels))))
        org_list = list(np.array(list(itertools.chain(*all_ents)))[is_org])
        org_list = list(map(lambda x: str(x), org_list))
        counted_orgs = Counter(org_list).most_common(5)
        counts = pd.DataFrame(counted_orgs, columns=['word', 'count']).
↪sort_values('count')
        ax[j].barh(counts.iloc[:,0], counts.iloc[:, 1])
        ax[j].set_title(ent_names[j])

    plt.subplots_adjust(left=0.0,bottom=0.1, right=0.9, top=0.9, wspace=0.8,␣
↪hspace=0.4)
```

[ ]:
```
plot_entities(all_ents, ent_labels)
```



`GPE` - najczęściej odnosi się do `Brukseli`, nic dziwnego - jest to siedziba instytucji UE

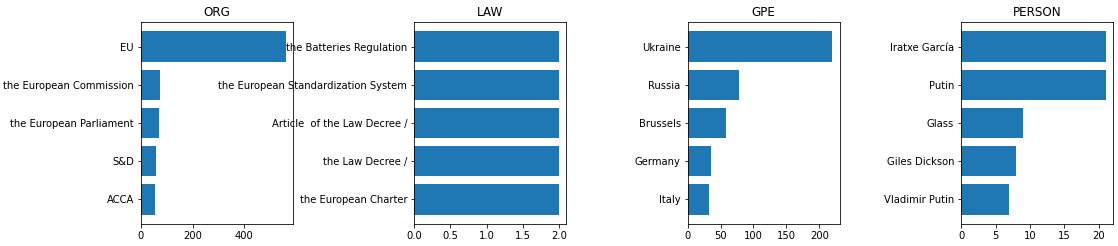
Zabawna obserwacja: algorytm potraktował `Brexit` jako osobę. Analiza pozostałych „osób'': - `Udo Bullmann` - niemiecki polityk i nauczyciel akademicki, poseł do Parlamentu Europejskiego; członek Socjaldemokratycznej Partii Niemiec; w roku 2018 został nowym przewodniczącym frakcji `S@D` - `RICS` (The Royal Institution of Chartered Surveyors) - brytyjska organizacja zawodowa dla geodetów, działa na poziomie międzyrządowym i ma na celu promowanie i egzekwowanie najwyższych międzynarodowych standardów w zakresie wyceny, zarządzania i zagospodarowania gruntów, nieruchomości, budownictwa i infrastruktury - kolejna pomyłka - `Sergei Stanishev` - bułgarski polityk i dziennikarz, przewodniczący Bułgarskiej Partii Socjalistycznej i Partii Europejskich Socjalistów (PSE), w latach 2005–2009 premier Bułgarii

`LAW` - tutaj też pomyłka: `the European Left` to partia polityczna...

Generalnie słaby ten algorytm.

1.5.2 Nazwy własne z podziałem na organizacje, akty prawne, miejsca oraz osoby w 2022r

[ ]:
```
doc_2022 = np.array(docs)[df.date.apply(lambda x: x.year == 2022)]
[ents_2022, labels_2022] = label_entities(doc_2022)
plot_entities(ents_2022, labels_2022)
```



`ORG` - z grubsza bez zmian w porównaniu z dokumentami ze wszystkich lat `LAW` - widać duże różnice, pojawiły się wyszczególnione zagadnienia prawne, któr okazały się ważne w roku 2022 (a raczej w ciągu kilku pierwszych miesięcy tegoż roku). `GPE` - na pierwszy plan wyszła `Ukraina` i `Rosja` - co nie dziwi ze względu na haniebny atak tej drugiej. Zaskakująca jest z kolei obecność Włoch. ? `PERSON` - duża zmiana: - `Iratxe Garcia` - hiszpańska polityk, posłanka do Kongresu Deputowanych, deputowana do Parlamentu Europejskiego, przewodnicząca frakcji Postępowego Sojuszu Socjalistów i Demokratów w Parlamencie Europejskim (S@D) - `Glass` - raczej pomyłka, nie ma takiej osobistości - `Giles Dickson` - UK Permanent Representation to the EU, jego motto: 'Leading the promotion of wind energy across Europe' - `Putin` - bez komentarza

2 Krótkie podsumowanie

Zbiór danych, który analizowaliśmy jest tak obszerny, że nie sposób przeanalizować wszystkie ciekawe obserwacje. Udało nam się jednak opisać część z nich.