

Autor: Piotr Puskarski

Przedmiot: Przetwarzanie języka naturalnego

Laboratorium: 1

1. Wydostań wszystkie **wartości pieniężne wyrażone w złotych** pojawiające się w tekstach orzeczeń określonego roku, znormalizuj je i przedstaw ich rozkład w postaci histogramu.
2. Jak w punkcie 1. ale zrób osobny wykres dla wartości **do 1 mln zł.** oraz **powyżej 1 mln zł.**

### Rozwiązanie

W celu znalezienia wszystkich wartości użyłem następującego wyrażenia regularnego:

```
found =
re.findall(r'((\b\d+[\d,]*\b) | (\b\d+\s?[\d.]*,?\d{2}?\b) | (\b\d+[\d\s]+,?\d{2}?\b))\s?'
r'((\bml\d\b\s?) | (\bmln\b\s?) | (\btys\b\s?) | (\b(?:star|w{1,3})\b)\b\s?) | (\b(?:w\s
\s]+\b)\b\s?))\s?')
r'((\bzłot|w{1,3})\b) | (\bzł|.|?)\b)', text['textContent'])
for matched in found:
    value = parse_value(matched)
    money.append(value)
```

```
((\b\d+[\d,]*\b) | (\b\d+\s?[\d.]*,?\d{2}?\b) | (\b\d+[\d\s]+,?\d{2}?\b))\s?
```

Odpowiada za znalezienie wartości w postaci takiej jak: 254546 lub 23.455.655,54 lub 80 760593,43

```
((\bml\d\b\s?) | (\bmln\b\s?) | (\btys\b\s?) | (\b(?:star|w{1,3})\b)\b\s?) | (\b(?:w\s
\s]+\b)\b\s?))\s?')
```

Po wartości liczbowej może wystąpić mln|tys|mld. Dodatkowo słowo „stare” lub „starych” w nawiasach lub nie. Także może pojawić się kwota słownie w nawiasach „(dwieście dziesięć”. Może być np. „(dwieście) mln starych” dlatego {0,3}.

```
((\bzłot|w{1,3})\b) | (\bzł|.|?)\b)
```

na samym końcu znajduje się określenie jednostki, czyli: „złote”, „złotych” lub „zł”.

czyli regex dopasuje się do wyrażenie np. takiego:

270,23 (dwieście siedemdziesiąt złotych i 23 grosze) zł

Następnie wartość jest zamieniana na float:

```
value = parse_value(matched)
```

Sprawdzanie, czy jest z roku 2008:

```
date = text['judgmentDate']
date = re.split(r'[-.\\/]', date)
if '2008' in date:
```

```
    return True
else:
    return False
```

## Wyniki

ilość znalezionych wartości: 16242

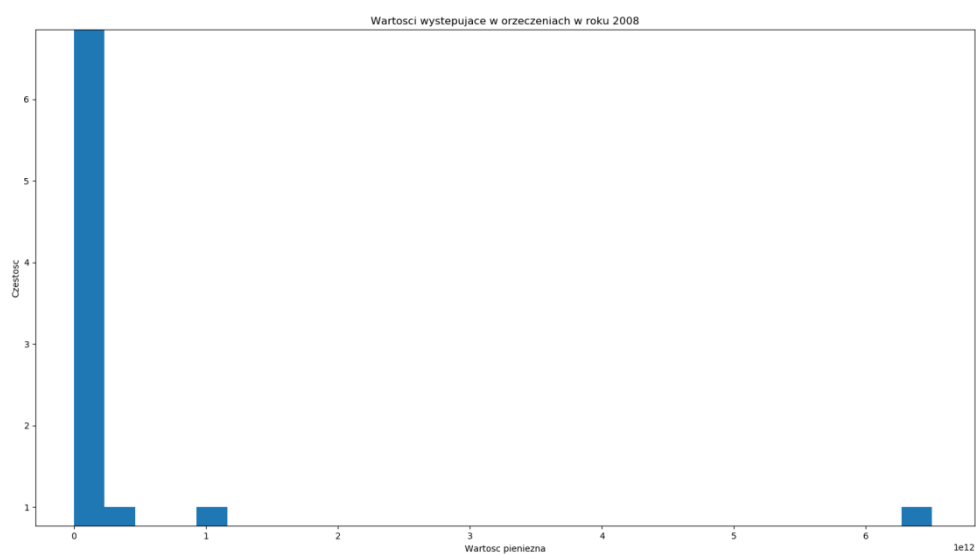
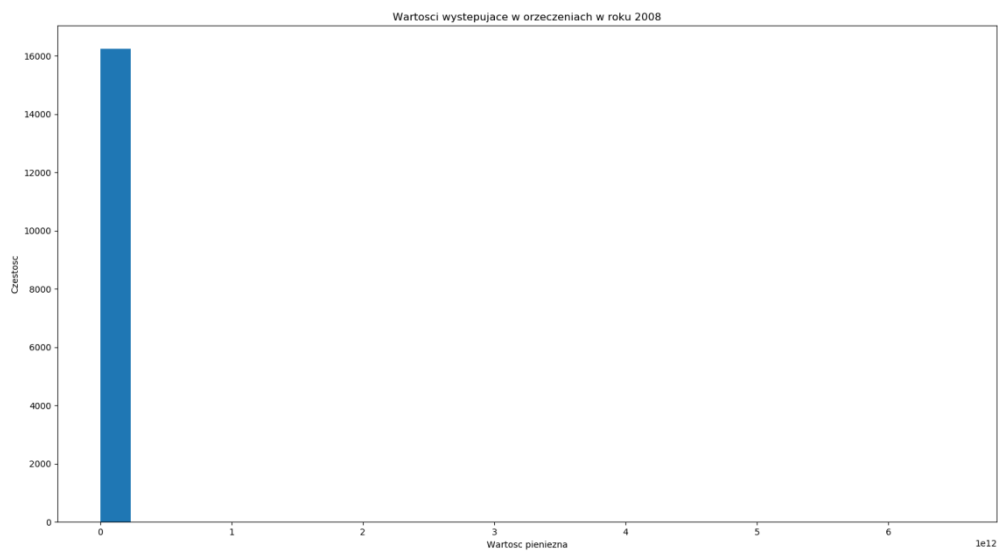
ilość wartości powyżej miliona: 2131

poniżej: 14111

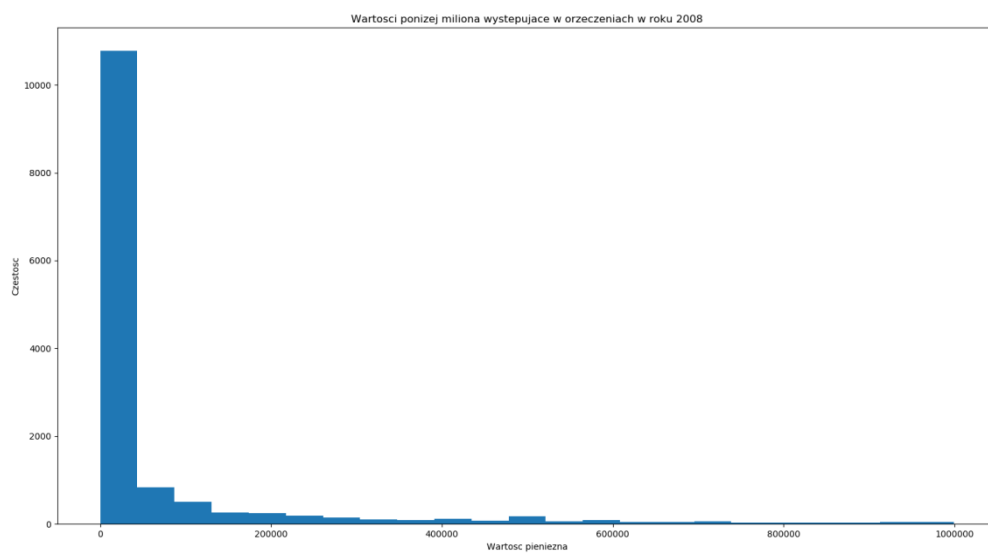
czas wykonania: 250 sekund

największa liczba: 6 500 000 000 000

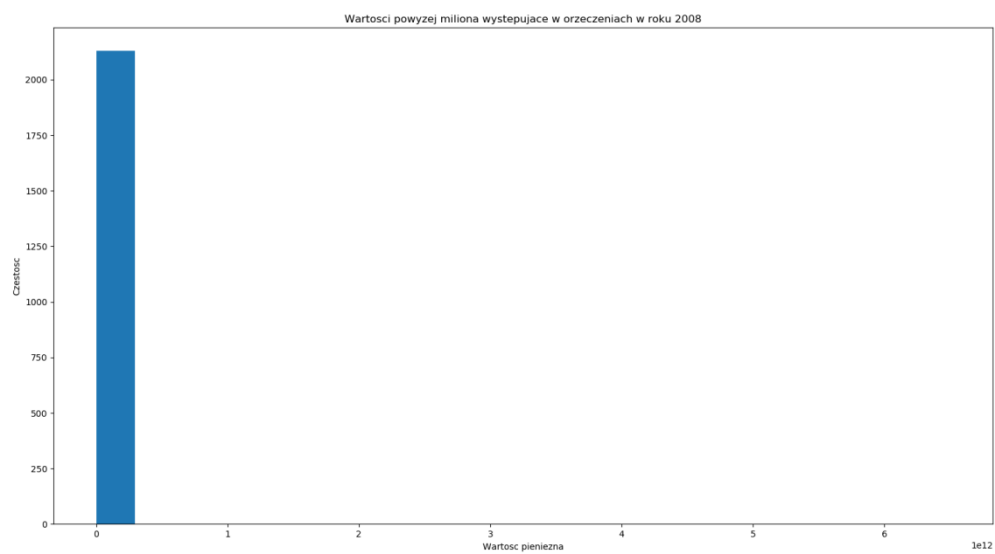
Następujący histogram przedstawia wszystkie wartości pieniężne występujące w orzeczeniach w roku 2008. Oś y to ilość, a x to wartość. Kolejny jest przybliżeniem w celu zobaczenia tych rzadziej występujących.

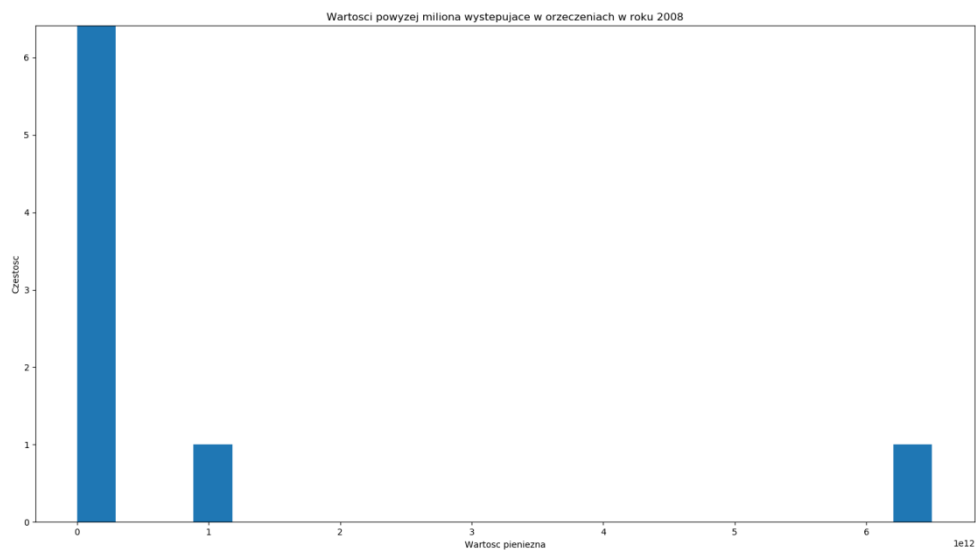


Kolejny histogram przedstawia wartości poniżej miliona zł:



Ostatnie dwa przedstawiaj powyżej miliona:





3. Określ liczbę orzeczeń odwołujących się w określonym roku do **artykułu 445 Ustawy z dnia 23 kwietnia 1964 r. - Kodeks cywilny**.

#### Rozwiązanie

```
for reference in item['referencedRegulations']:
    found =
re.findall(r'(\bUstawa\b\s+\b23\b\s+\bkwietnia\b\s+\b1964\b\s+r\.\s+\bKodeks\b\s+\bcywilny\b) '
           r'[\S\s]+(\bart.\s+\b445\b)', reference['text'])
f.extend(found)
```

Pierwsza grupa to "Ustawa z dnia 23 kwietnia 1964 r. – Kodeks cywilny". Następnie dowolne znaki aż do napotkania art. 445

#### Wyniki

39 znalezionych

Czas wykonania: 266 sekund

4. Określ liczbę orzeczeń w określonym roku, które zawierają słowo **szkoda** w dowolnej formie fleksyjnej. Wynik ten nie może obejmować innych słów, które mają wspólny prefiks ze słowem szkoda, np. *szkodzić, szkodzący*, itp.

#### Rozwiązanie

Najpierw znalazłem słowa zawierające „szkod” lub „szkód”. Następnie sprawdziłem czy końcówka znalezionej słowa należy do jakiegokolwiek końcówki formy fleksyjnej „szkoda”.

```
found = re.findall(r'\b[Šš]zk[oó]d\w{0,3}\b', text['textContent'])

for word in found:
    if proper(word):
        words.append(word)
        break
```

Sprawdzanie końcówki:

```
szkoda_ends = ['a', 'y', 'zie', 'ę', 'ą', 'o', 'om', 'ami', 'ach']
words = []

def proper(word):
    if word.lower() == 'szkód':
        return True
    elif word[5:] in szkoda_ends:
        return True
    return False
```

## Wynik

829 orzeczeń

Czas wykonania: 280 sekund