

# Cyfrowe narzędzia w przekładoznawstwie

# Główne funkcje: oppose()

> **oppose()**

---

oppose

*Contrastive analysis of texts*

---

## Description

Function that performs a contrastive analysis between two given sets of texts. It generates a list of words significantly preferred by a tested author (or, a collection of authors), and another list containing the words significantly avoided by the former when compared to another set of texts. Some visualizations are available.

# Główne funkcje: oppose()

```
> oppose()
```

Pokazuje słownictwo „preferowane” oraz „unikane” przez wybrane zbiory tekstów.

# Główne funkcje: oppose()

```
> oppose()
```

**UWAGA:** Pamiętamy, że **oppose()** to funkcja działająca w ramach **biblioteki stylo!** Oznacza to, że aby ją wywołać, musimy załadować bibliotekę **library(stylo)**, jeśli nie robiliśmy tego wcześniej.

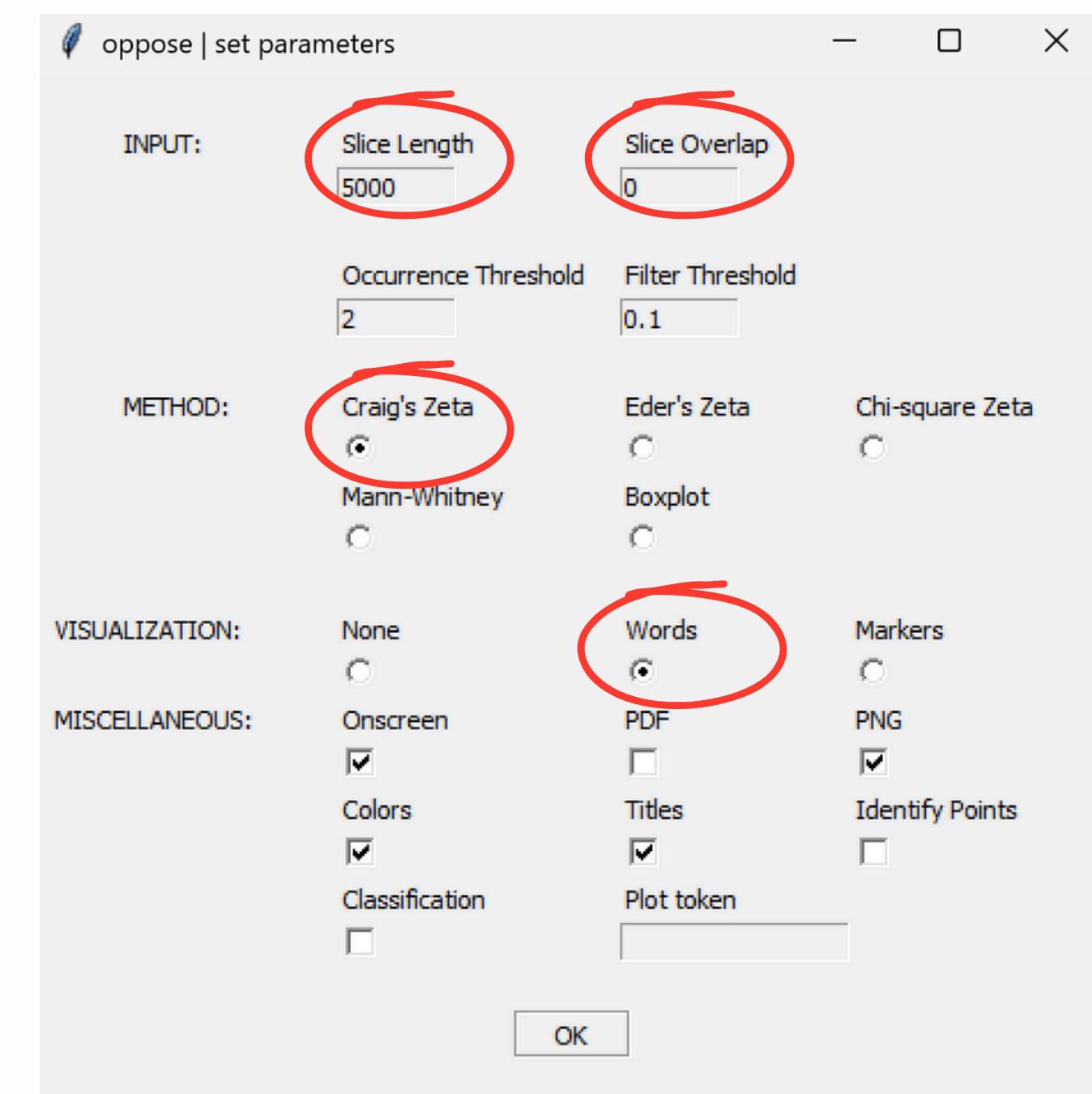
**OGÓLNA UWAGA:** Pamiętajmy, żeby po włączeniu konsoli R załadować bibliotekę **library(stylo)**, bo inaczej nie wywołamy żadnej funkcji tej biblioteki :)

# Główne funkcje: oppose()

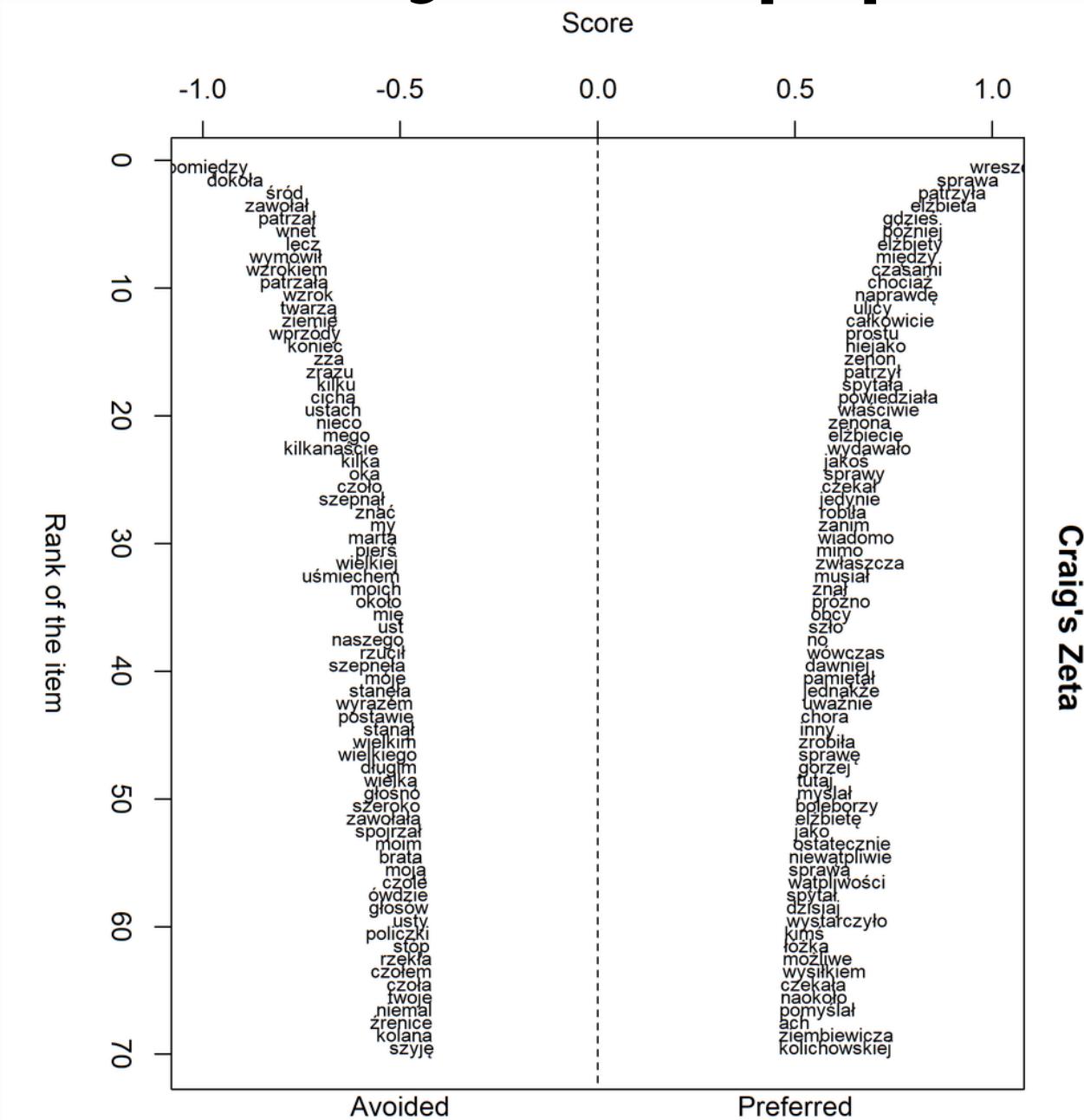
Aby wykonać analizę, musimy zmienić układ folderów. Zamiast jednego folderu **corpus**, potrzebujemy co najmniej dwóch folderów: **primary\_set** z tekstami odpowiadającymi jednej wybranej przez nas kategorii oraz **secondary\_set** z tekstami odpowiadającymi drugiej kategorii. Opcjonalnie dodajemy także folder **test\_set**.

📁 primary_set	18.01.2025 23:02	Folder plików
📁 secondary_set	18.01.2025 23:02	Folder plików
📁 test_set	18.01.2025 23:02	Folder plików

# Główne funkcje: oppose()



# Główne funkcje: oppose()



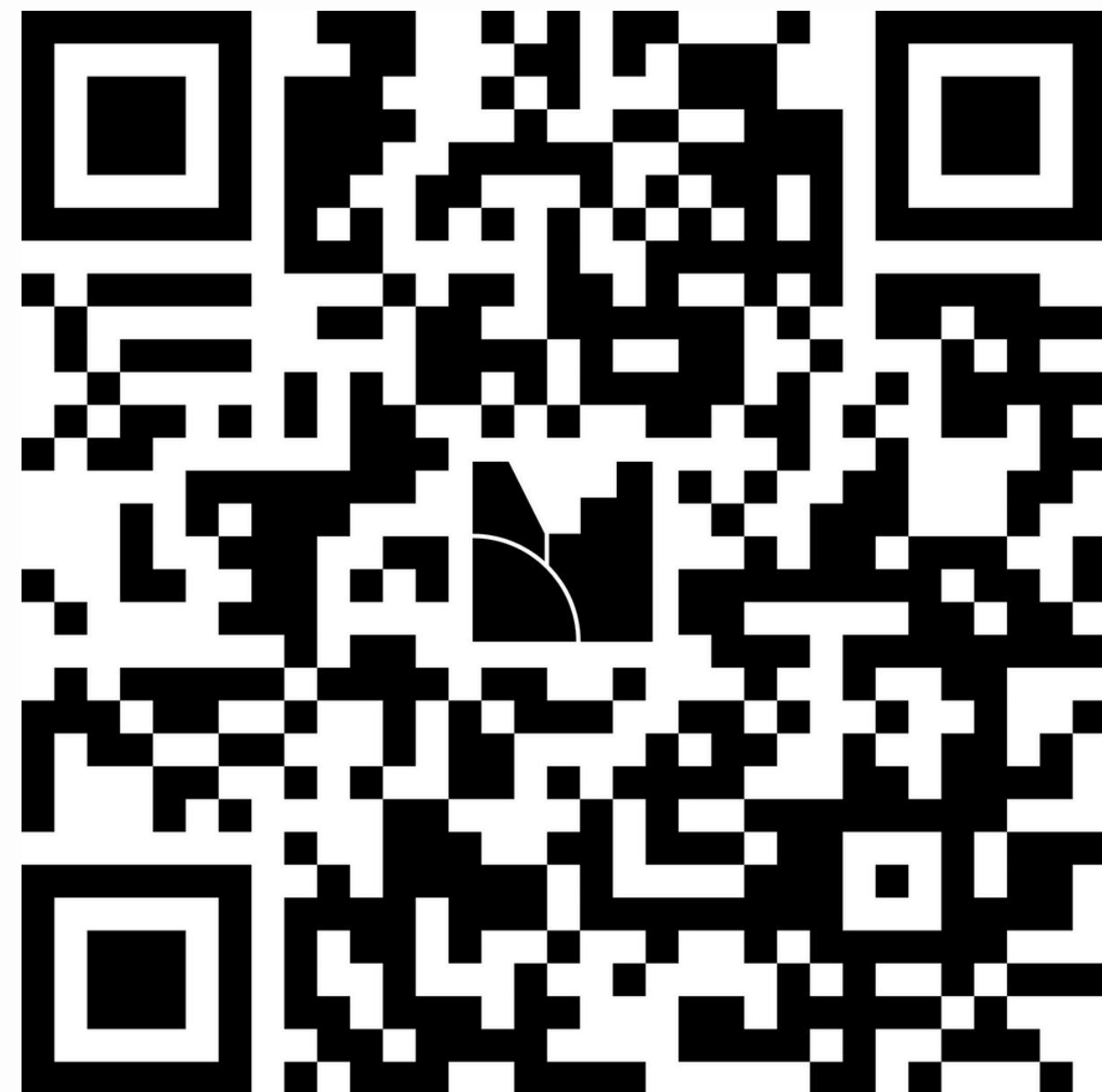
# Zadanie do funkcji oppose()

Chcemy dowiedzieć się, czy w tekstach zawartych w korpusie **Klasyka polska** zauważymy wyraźne różnice w „stylu” (czyli w naszym wypadku używanym słownictwie) w zależności od tego, czy teksty były pisane przez kobiety, czy przez mężczyzn.

1. W folderze Klasyka polska tworzymy foldery **primary\_set** oraz **secondary\_set**. Do pierwszego kopujemy teksty pisane przez polskie pisarki, do drugiego przez pisarzy.
2. Otwieramy konsolę R, ładowamy bibliotekę stylo, używamy funkcji **oppose()**, aby wygenerować wykres.
3. Jakie są wnioski z analizy?

# Zadanie do funkcji oppose()

Jakie inne analizy można przeprowadzić na tym korpusie za pomocą funkcji oppose()? Jakie kryteria możemy wziąć pod uwagę przy podziale tekstów na primary\_set i secondary\_set?



# Wyrażenia regularne

. - wildcard, zastępuje jakikolwiek znak, jak pusty kafelek w Scrabble

\ - następny znak jest tylko znakiem (np. \\$ to po prostu napisany \$, a nie koniec ciągu)

# Wyrażenia regularne

^ - początek stringu

\$ - koniec stringu

String - ciąg znaków

# Wyrażenia regularne

? - poprzedni znak może wystąpić,  
ale nie musi  
np. a?społeczny może oznaczać  
„społeczny” albo „aspółeczny”

| - albo to, co jest przed kreską,  
albo to, co za nią

# Wyrażenia regularne

+ - jeden albo więcej znaków, po których + występuje  
np. *a*+*a* może oznaczać „ana” albo „anna” albo „annna”...

\* - zero albo więcej znaków, po których \* występuje; domknięcie Kleenego  
np. *a*\**społeczny* może oznaczać „społeczny” albo „aspołeczny” albo „aaspółeczny”...

# Wyrażenia regularne

[ghi] - g albo h albo i (kolejność liter w nawiasach nie ma znaczenia)

[1-9] - od 1 do 9  
[a-z] - od a do z  
[A-Za-z] - od a do z i bierzemy pod uwagę każdą wielkość liter

# Wyrażenia regularne

{n} - ile razy poprzedni znak ma się powtórzyć  
np. an{2}a oznacza „anna”

{m,n} - conajmniej m powtórzeń i nie więcej niż n powtórzeń

# Wyrażenia regularne

() - zakres operatorów

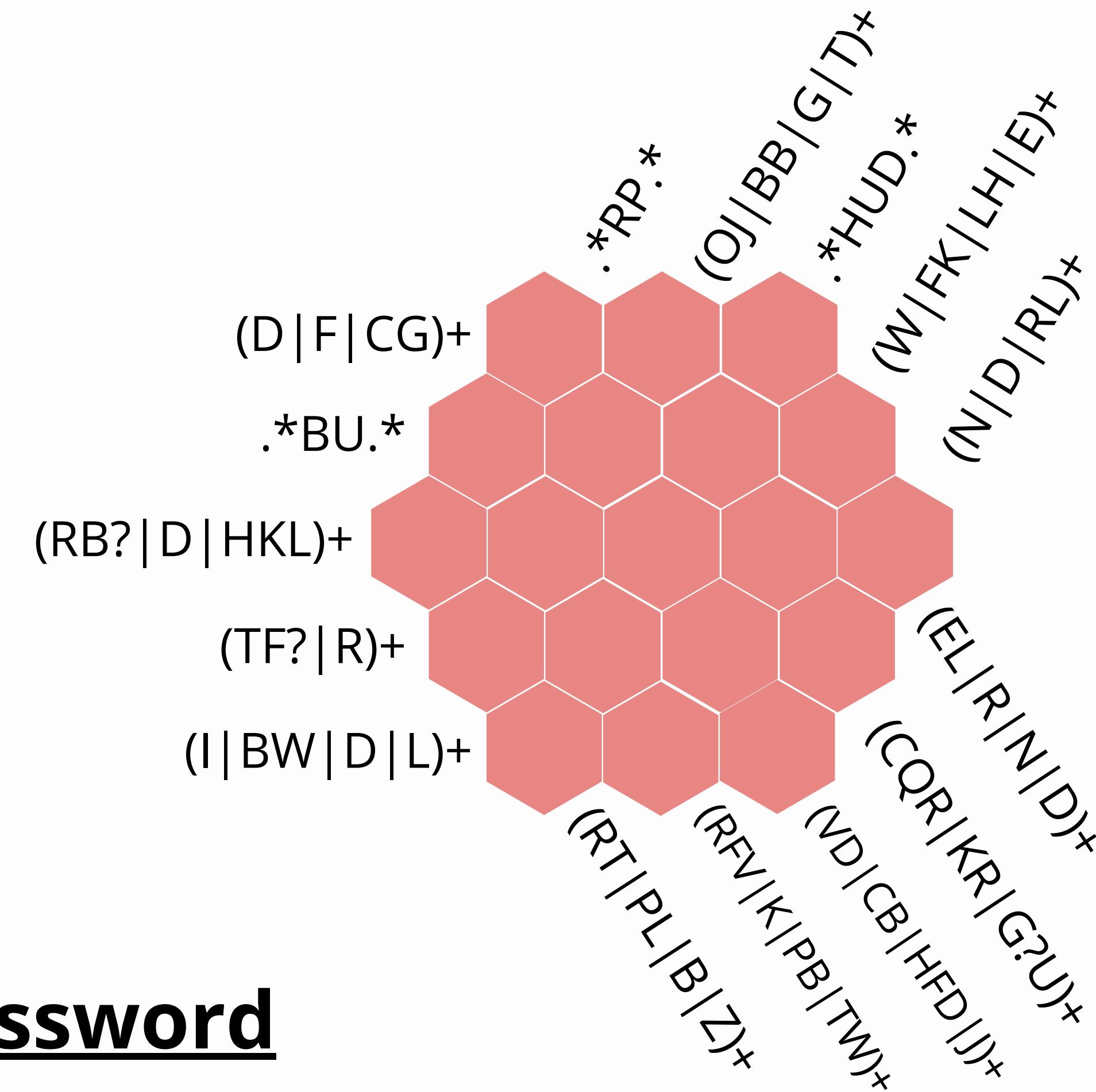
(ed|ing)\$ oznacza, że albo się skończy wyraz na -ed albo na -ing

# Wyrażenia regularne

- . - wildcard, zastępuje jakikolwiek znak
- \$ - koniec stringu
- ^ - początek stringu
- ? - poprzedni znak może wystąpić, ale nie musi
- [ghi] - g albo h albo i (kolejność liter w nawiasach nie ma znaczenia)
- [g-o] - od g do o
- + - jeden albo więcej znaków, po których + występuje
- \* - zero albo więcej znaków, po których \* występuje; domknięcie Kleenego
- \ - następny znak jest tylko znakiem (np. \\$ to po prostu napisany \$, a nie koniec ciągu)
- {n} - ile razy poprzedni znak ma się powtórzyć
- {m,n} - conajmniej m powtórzeń i nie więcej niż n powtórzeń
- | - albo to, co jest przed kreską, albo to, co za nią
- () - zakres operatorów
- (ed | ing)\$ - albo się skończy wyraz -ed albo -ing

# Wyrażenia regularne

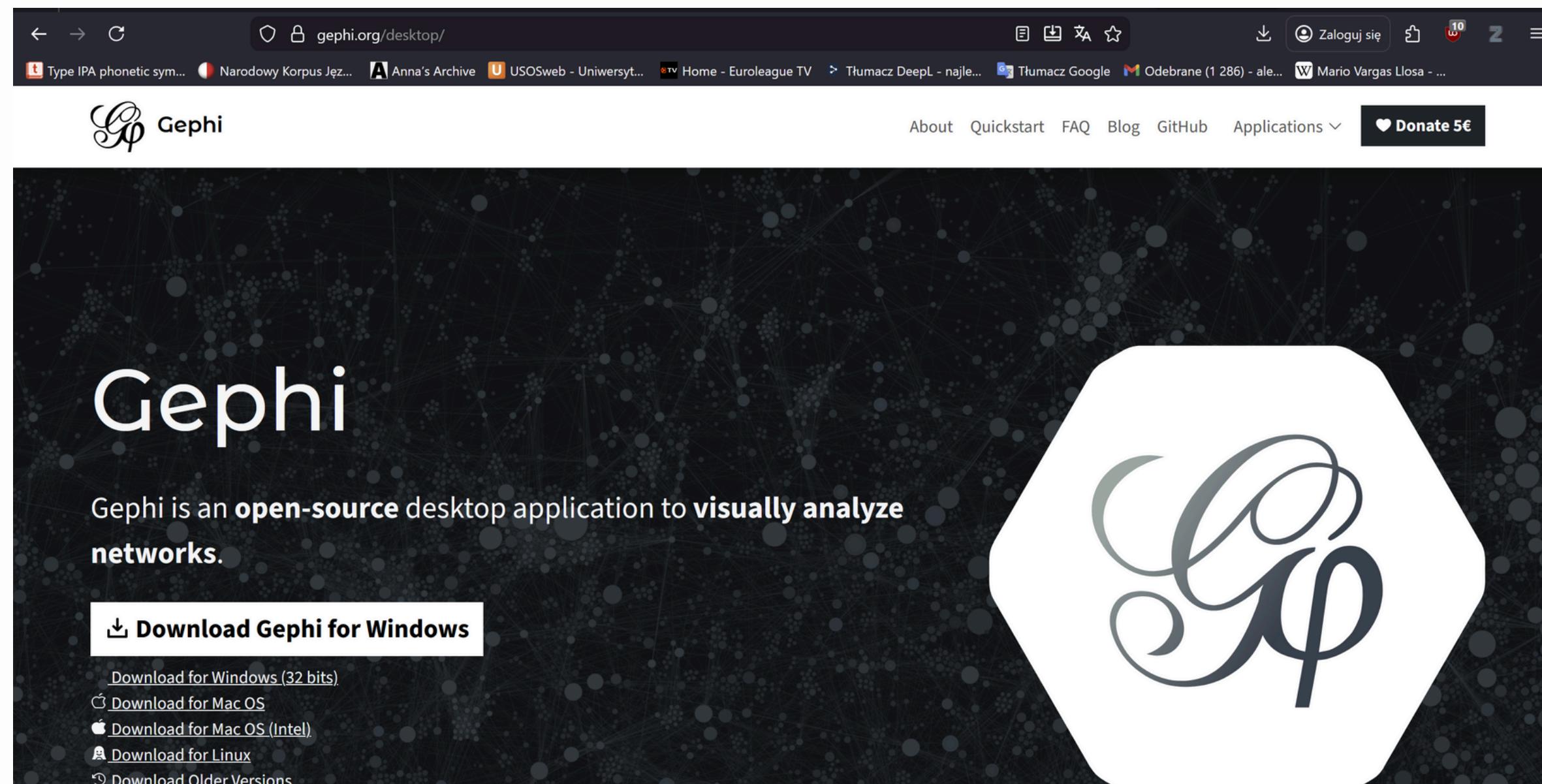
1. Jakim wyrażeniem regularnym wyszukamy wyrazy, które zaczynają się od prefiksów *pod-, ob-, nad-*?
2. Jakim wyrażeniem regularnym wyszukamy wyrazy, które kończą się na *-any, -ana, -ane*?
3. Jakim wyrażeniem regularnym wyszukamy wyrazy, które mają geminatę *-nn-*?
4. **Jakim wyrażeniem regularnym wyszukamy pierwszy wyraz w nazwach plików, które zostały utworzone według wzoru: Autor\_Tytuł\_Powieści\_Rok\_Wydania?**
5. Jakim wyrażeniem regularnym wyszukamy w nazwach plików utworzonych według schematu powyżej same lata wydania książek?



**Regexle**

**Regex crossword**

# Instalowanie Gephi



# Do zobaczenia za tydzień!

