Tarefa 1

Implemente um programa em Python (Python notebook) que constrói um dicionário de termos de um documento (mínimo 1000 palavras) com a frequência de cada termo permitindo confirmar a lei de Zipf para o documento selecionado.

Comente ao final os resultados obtidos.

Resolução

Para a resolução do exercício foi utilizado o seguinte texto: https://www.nexojornal.com.br/expresso/2022/03/08/A-flexibiliza%C3%A7%C3%A3o-do-uso-de-m%C3%A1scaras-no-Brasil-e-no-mundo

Escolhi esse texto por estar relacionado com a nossa situação pandêmica atual, aonde diversas cidades, estados e países estão flexibilizando o uso das máscaras.

O texto foi copiado para um arquivo .txt para que pudéssemos considerar apenas o texto em si, sem os demais elementos da página web. Em seguinda o arquivo foi disponibilizado no GitHub: https://github.com/BrunaKrasotaMatos/ParadigmasProgramacao/blob/b8d339c31e532fe89e54ca31e853a2b97a8a520f/Trilha1/Trilha1TextExercicio.txt

```
#imports
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

# Declaração de variáveis
texto = []
mydict = {}

# Abrindo arquivo
arquivo = open('/content/Trilha1TextExercicio.txt','r')
print(arquivo)

# Abrindo e lendo o arquivo
for line in arquivo:
    texto.append(line)

# Eliminando itens desnecessários no texto
for i in range(len(texto)):
    texto[i] = texto[i].lower() # para unicidade
    texto[i] = texto[i].replace('\n','')
    texto[i] = texto[i].replace('.','')
    texto[i] = texto[i].replace(',','')
```

```
    texto[i] = texto[i].replace('(','')
    texto[i] = texto[i].replace(')','')
    texto[i] = texto[i].replace('?','')
    texto[i] = texto[i].replace('\'','') # elimina ' e "

# Usando a lista 'texto' populada para formar o dicionário
for line in texto:
  line = line.lower()                  # converte para lower
  words = line.split()                 # separa cada palavra
  #print(words)

  for word in words:                     # para cada palavra em words
    if word not in mydict.keys():        # se palavra não está no dicionário
      mydict[word] = 1                   # acrescenta a word com o valor 1
    else:                                # se a entrada já existe
      mydict[word] =  mydict[word] + 1   # apenas soma 1 ao valor já existente

# Utilizando o dicionário populado anteriormente para apresentar os resutados em forma

df = pd.DataFrame(mydict.items(),  columns=['word', 'count']).sort_values('count',asce
df = df[df['count'] > 4] # somente termos com mais de 4 ocorrências
# df = df.iloc[ np.int(len(df)/2) - 10 : np.int(len(df)/2) + 10 ] # para livros ou tex

plt.figure(figsize=(24,10))
mpl.style.use(['seaborn'])
sns.barplot(x=df.word,y=df['count'])
plt.xticks(rotation=90)

plt.show()
```

Comentários

Analisando o resultado obtido foi possível comprovar a Lei de Zipf. Isso significa que as palavras que apareceram mais frequentemente no texto tratam-se de palavras sem muito significado para o tema. Somente quando percebemos uma estabilização na quantidade de uso das palavras é que elas passam a ter alguma relevância.