Part 1: Regular expression warmup

In [ ]:

```
##### -- Imports -- #####
import re
import pandas as pd

##### -- Variables -- #####
data = '''1307026153
2308134469
1211004254
1517972564
151797-2564'''

##### -- Functions -- #####
def pandasDataFrame(n):
    df = pd.DataFrame(
        columns = ["DD", "MM", "YY", "IIII"]
    )
    pattern = re.compile('(^\d{2})(\d{2})(\d{2})-?(\d{4})')
    data_lines = n.split('\n')
    for data_line in data_lines:
        match = pattern.search(data_line)
        if match:
            newRow = [str(match.group(1)), (match.group(2)),
                      (match.group(3)), (match.group(4))]
            df.loc[len(df)] = newRow
    return(df)

def born(n):
    cprNumber = n
    pattern = re.compile('(^\d{2})(\d{2})(\d{2})-?(\d{4})')
    match = pattern.search(cprNumber)
    A = int(match.group(4))
    B = int(match.group(3))
    if (A < 4000 and B < 100 ): return(1900)
    elif (4000 <= A < 5000 and B <= 36): return(2000)
    elif (4000 <= A < 5000 and 37 <= B <= 99): return(1900)
    elif (5000 <= A < 9000 and 00 <= B <= 57): return(2000)
    elif (5000 <= A < 9000 and 58 <= B <= 99): return(1800)
    elif (9000 <= A < 10000 and 00 <= B <= 36): return(2000)
    elif (9000 <= A < 10000 and 37 <= B <= 99): return(1900)
    else: return ("Error")

##### -- Calls -- #####
print(pandasDataFrame(data))
cpr1 = '1517972564'
print("The person with cpr-number: "+ cpr1 + " is born in " + str(born(cpr1)))
```

```
   DD  MM  YY  IIII
0  13  07  02  6153
1  23  08  13  4469
2  12  11  00  4254
3  15  17  97  2564
4  15  17  97  2564
The person with cpr-number: 1517972564 is born in 1900
```

Part 2: Processing the FakeNewsCorpus data set

In [ ]:

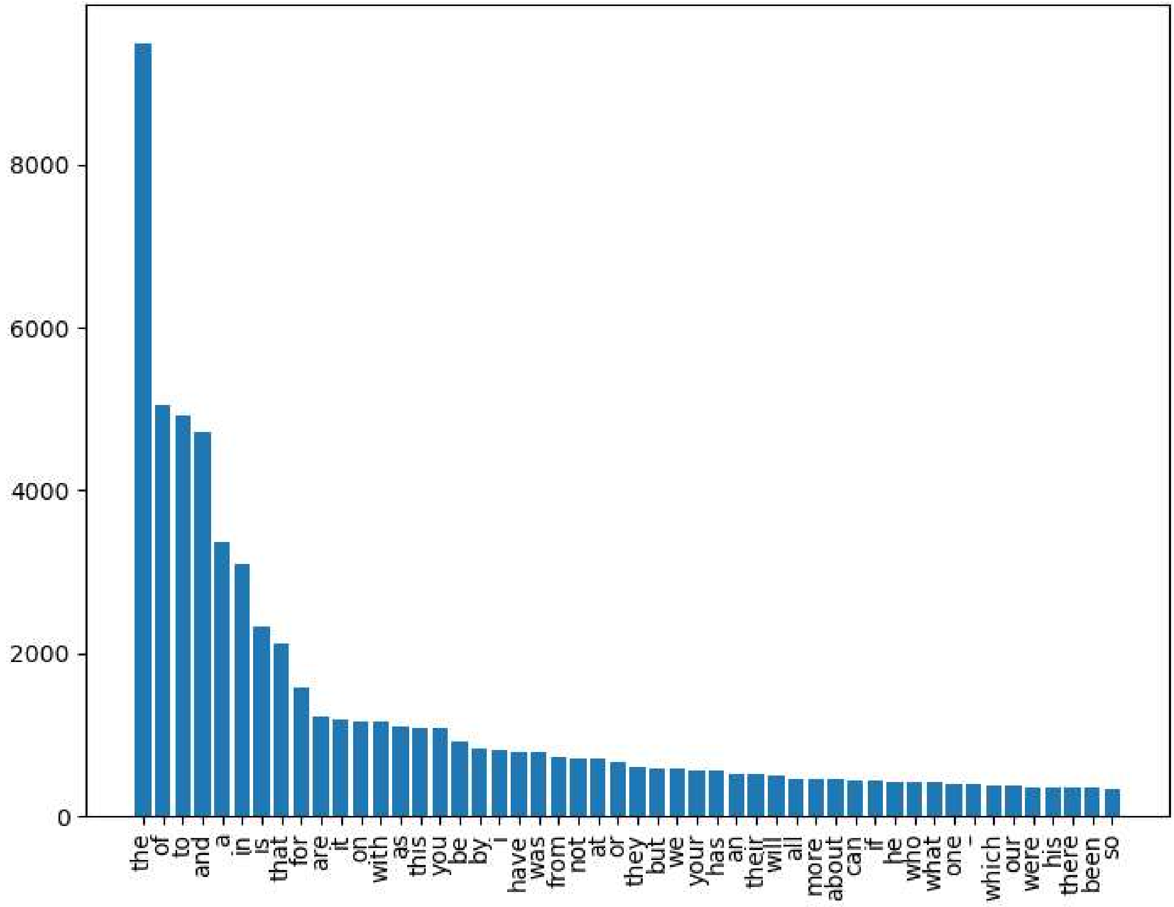
```
##### -- Imports -- #####
import csv
import re
import pandas as pd
from cleantext import clean

##### -- Variables -- #####
originalData =  'news_sample.csv'

##### -- Functions -- #####
def cleanCsv(data):
    with open(data, 'r') as csv_in, open('edited_sample.csv','w', newline = '') as
        csv_reader = csv.reader(csv_in)
        csv_writer = csv.writer(csv_out)
        for row in csv_reader :
            newRow = [cell.lower() for cell in row]
            newRow = [re.sub(r'\s+', ' ', cell) for cell in newRow] #White spaces
            newRow = [re.sub(r'[a-zA-Z0-9-_.]+@[a-zA-Z0-9-_.]+', '<MAIL>', cell) f
            newRow = [re.sub(r'\S+@S+', '<URL>', cell) for cell in newRow]  #URL p
            newRow = [re.sub(r'[a-zA-Z0-9-_.]+.com', '<URL>', cell) for cell in ne
            newRow = [re.sub(r'(\d{4}-?\d{2}-?\d{2})', '<DATE>', cell) for cell in
            newRow = [re.sub(r'[a-z]{3,9}\s\d{2},.\d{4}', '<DATE>', cell) for cell
            newRow = [re.sub(r'[a-z]{3,8}\s\d{2}\s[a-z]{3}\s\d{4}', '<DATE>', cell
            newRow = [re.sub(r'\d*\.?\d*$/', '<NUM>', cell) for cell in newRow] #N
            newRow = [re.sub(r'\d', '<NUM>', cell) for cell in newRow] #All number
            csv_writer.writerow(newRow)

def cleantextCsv(data):
    df = pd.read_csv(data)
    for col in df.columns:
        if df[col].dtype == 'object':
            df[col] = df[col].apply(lambda x: clean(x,
                fix_unicode=True,
                to_ascii=True,
                lower=True,
                no_line_breaks=True,
                no_urls=True,
                no_emails=True,
                no_phone_numbers=True,
                no_numbers=True,
                no_digits=True,
                no_currency_symbols=True,
                replace_with_url="<URL>",
                replace_with_email="<MAIL>",
                replace_with_phone_number="<PHONE>",
                replace_with_number="<NUM>",
                replace_with_digit="0",
                replace_with_currency_symbol="<CUR>",
                lang="en"))

    df.to_csv('clean_sample.csv')

##### -- Calls -- #####
cleanCsv(originalData)
cleantextCsv(originalData)
```

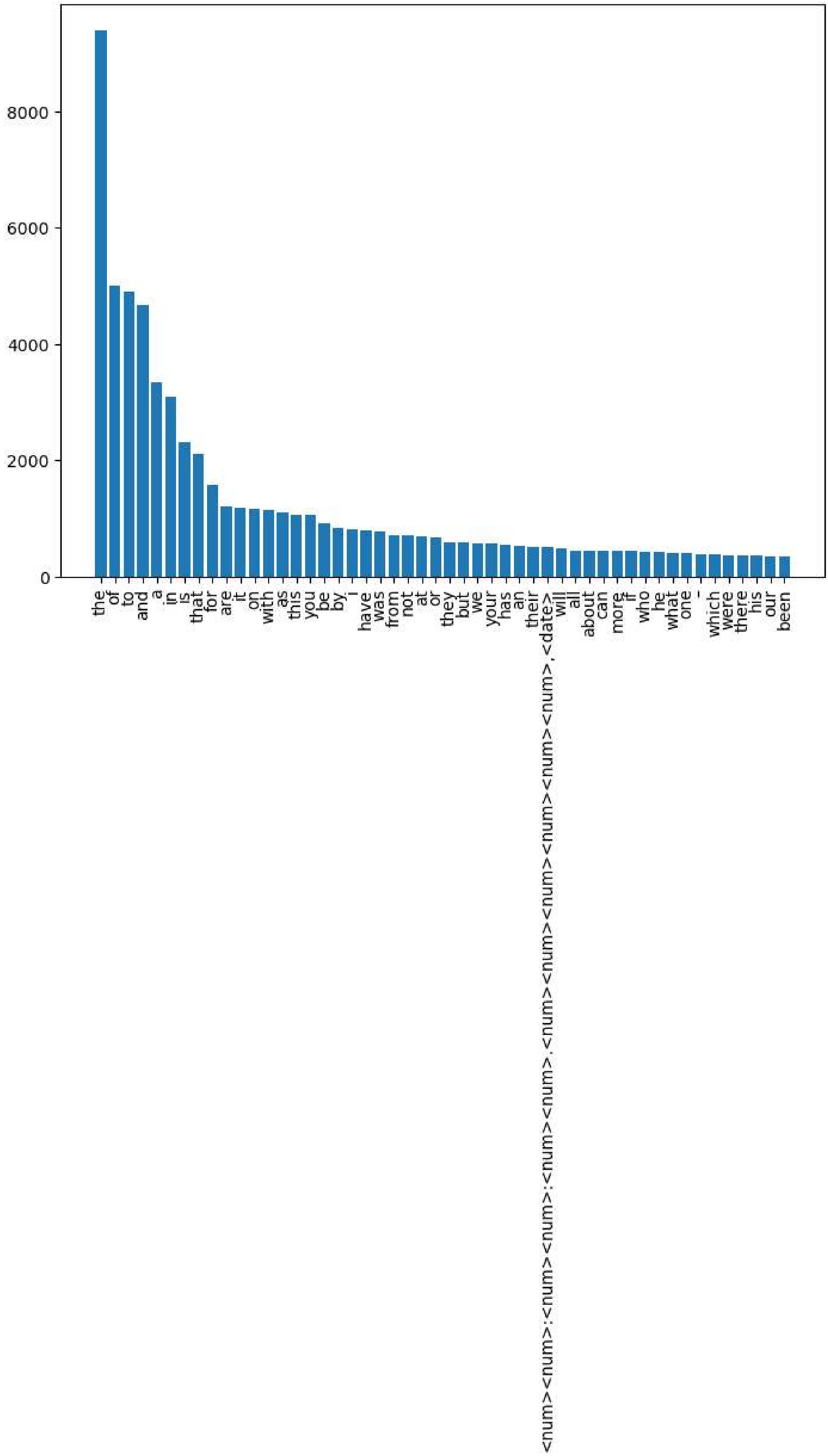
Part 3: Descriptive frequency analysis of the data

In [ ]:

```
##### -- Imports -- #####
import matplotlib.pyplot as plt
import itertools

##### -- Functions -- #####
def wordDic(data):
    file = open(data, 'r')
    read = file.read().lower()
    words = read.split()
    dictionary = {}
    for i in words:
        if i in dictionary:
            dictionary[i] += 1
        else:
            dictionary[i] = 1
    return dictionary

def plot (data):
    Sort = dict(sorted(wordDic(data).items(), key=lambda x: x[1], reverse=True))
    plotDictionary = dict(itertools.islice(Sort.items(), 50))
    axis = plt.figure().add_axes([0,0,1,1])
    x = list((plotDictionary).keys())
    y = list((plotDictionary).values())
    axis.bar(x, y)
    plt.xticks(rotation = 90)
    plt.show()

##### -- Calls -- #####
print ("Number of unique words (original): " + str(len(wordDic('news_sample.csv'))
print ("Number of unique words (cleaned) : " + str(len(wordDic('news_sample.csv'))
plot('news_sample.csv')
plot('edited_sample.csv')
```

```
Number of unique words (original): 28808
Number of unique words (cleaned) : 28808
```




0
2000
4000
6000
8000
the
of
to
and
a
in
is
that
for
are
it
on
with
as
this
you
be
by
i
have
was
from
not
at
or
they
but
we
your
has
an
their
<num><num>:<num><num>:<num><num>.<num><num><num><num><num><num><num>,<date>
will
all
about
can
more
if
who
he
what
one
-
which
were
there
his
our
been