week2-code

January 30, 2025

1 Week 1 Ingesting and Exploring the Dataset

[1]:
```
# install wordcloud
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (1.9.4)
Requirement already satisfied: numpy>=1.6.1 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from wordcloud)
(1.24.4)
Requirement already satisfied: pillow in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from wordcloud)
(10.2.0)
Requirement already satisfied: matplotlib in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from wordcloud)
(3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from
matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from
matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from
matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from
matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from
matplotlib->wordcloud) (23.1)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from
matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from
matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in
```

```
c:\users\geean\appdata\local\anaconda3\lib\site-packages (from python-
dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
```

[2]:
```
# import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

[7]:
```
# change working directory
import os
os.getcwd()  # Get current working directory
os.chdir('..')  # Move up one directory level from notebooks
print(os.getcwd())
#os.chdir('../data') # change to the data folder
```

```
C:\Users\geean\Documents\GitHub\analyticsproject
```

[11]:
```
# load the data
df = pd.read_csv('data/Combined Data.csv', index_col=0)

# make a copy and get rid of the missing values
df1 = df.copy()
df1.dropna(inplace = True)
df1.head()
```

[11]:

	statement	status
0	oh my gosh	Anxiety
1	trouble sleeping, confused mind, restless hear…	Anxiety
2	All wrong, back off dear, forward doubt. Stay …	Anxiety
3	I've shifted my focus to something else but I'…	Anxiety
4	I'm restless and restless, it's been a month n…	Anxiety

[13]:
```
# number of missing values
missing_values = df.isnull().sum()

print(missing_values)
```

```
statement    362
status         0
dtype: int64
```

[15]:
```
# get the rows and columns of the data
rows,columns = df1.shape
print(f"Number of rows: {rows}")
```

```
print(f"Number of columns: {columns}")
```

```
Number of rows: 52681
Number of columns: 2
```

[17]:
```
# create a new column that gives the length of each statement
df1['statement_len'] = df1['statement'].apply(lambda x: len(x.split(' ')))
df1.head()
```

[17]:

	statement	status	statement_len
0	oh my gosh	Anxiety	3
1	trouble sleeping, confused mind, restless hear…	Anxiety	10
2	All wrong, back off dear, forward doubt. Stay …	Anxiety	14
3	I've shifted my focus to something else but I'…	Anxiety	11
4	I'm restless and restless, it's been a month n…	Anxiety	14

[19]:
```
# information about the dataset
'''The class type of the DataFrame.
The range of the index.
The number of columns and their names.
The count of non-null values in each column.
The data type of each column.
The memory usage of the DataFrame.'''

print(df1.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 52681 entries, 0 to 53042
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   statement      52681 non-null  object
 1   status         52681 non-null  object
 2   statement_len  52681 non-null  int64 
dtypes: int64(1), object(2)
memory usage: 1.6+ MB
None
```

[21]:
```
# descriptive statistics
'''count is the number of non-null entries.
unique is the number of unique values.
top is the most frequent value.
freq is the frequency of the most frequent value.'''
df1.describe(include='object').T
```

[21]:

	count	unique	top	freq
statement	52681	51073	what do you mean?	22
status	52681	7	Normal	16343

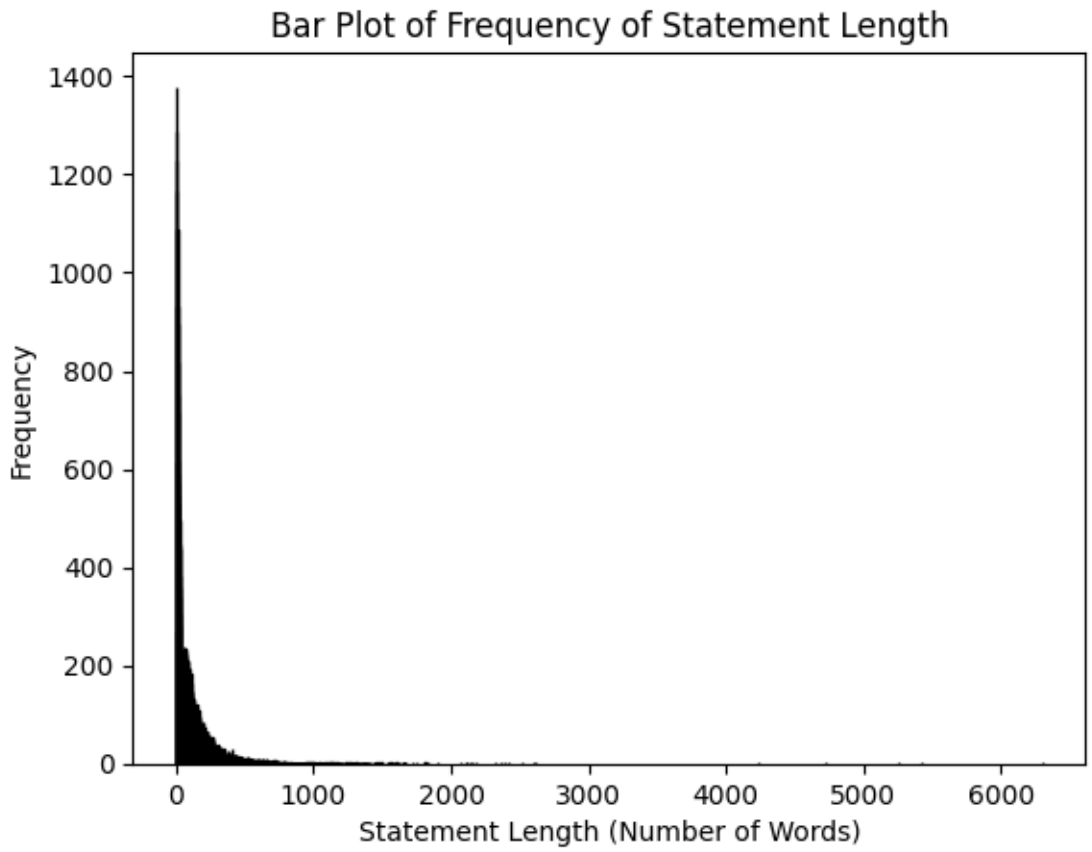
[23]:
```
# Get summary statistics for the 'statement_len' column
summary_statistics = df1['statement_len'].describe()
print(summary_statistics)
```

```
count    52681.000000
mean       113.035914
std        163.501877
min          1.000000
25%         15.000000
50%         62.000000
75%        148.000000
max       6300.000000
Name: statement_len, dtype: float64
```

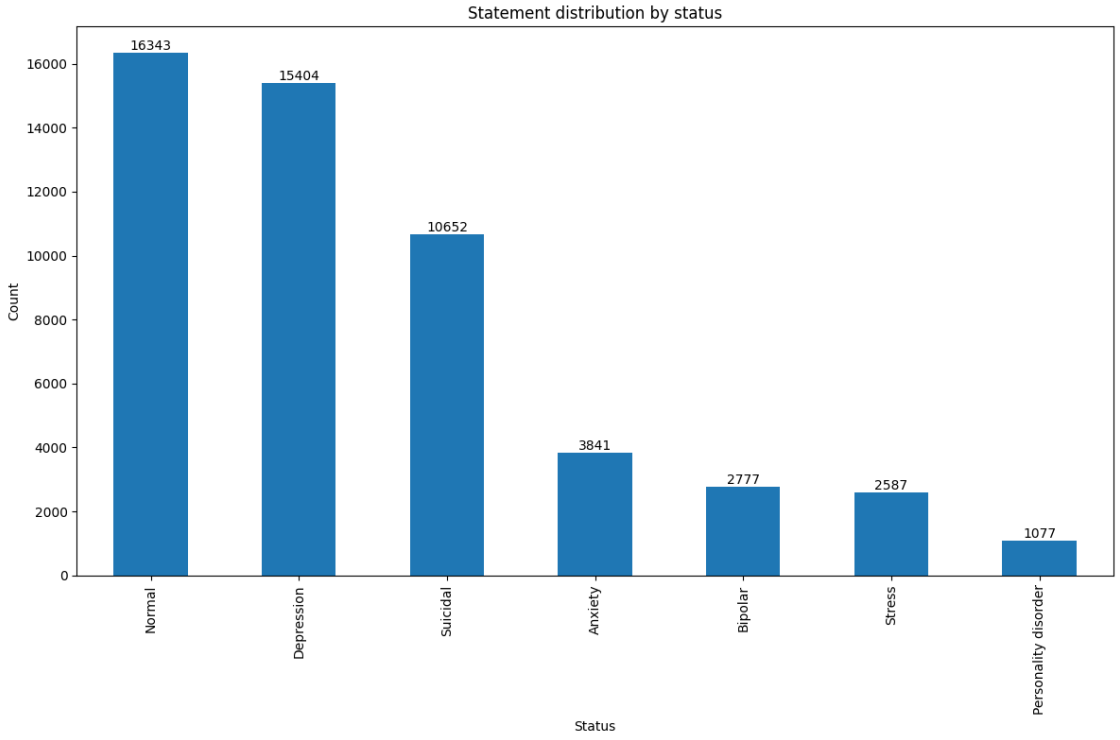
[25]:
```
# Calculate the mode of the 'statement_len' column
mode_value = df1['statement_len'].mode()[0]

print(f"The mode of the 'statement_len' column is: {mode_value}")
```

```
The mode of the 'statement_len' column is: 5
```

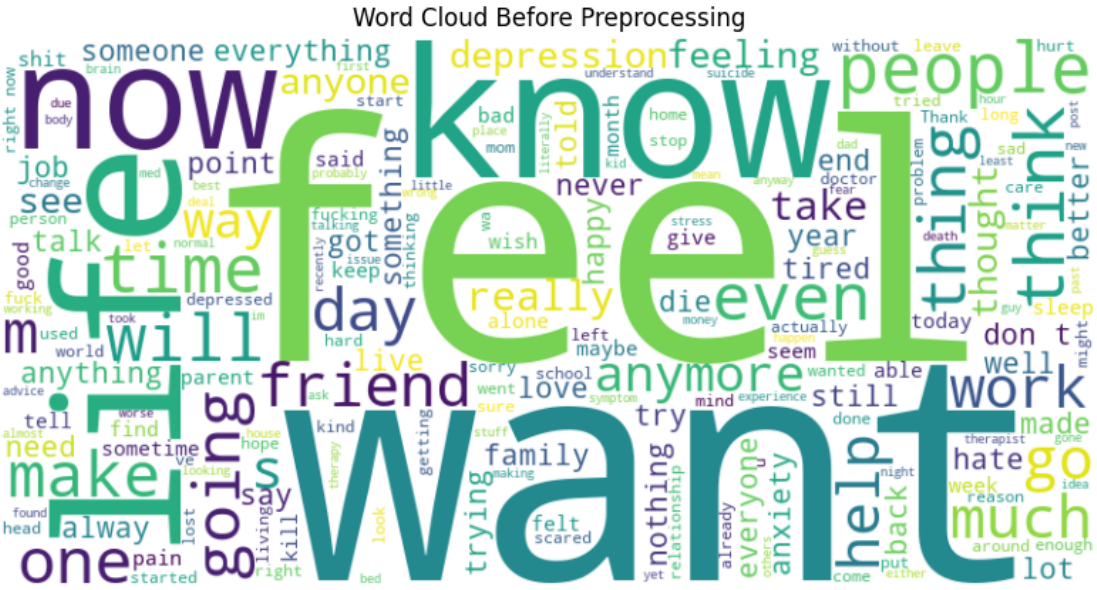
[27]:
```
# Create a bar plot of the frequency of the 'statement_len' column
statement_len_counts = df1['statement_len'].value_counts()

plt.bar(statement_len_counts.index, statement_len_counts.values,␣
 ↪edgecolor='black')
plt.xlabel('Statement Length (Number of Words)')
plt.ylabel('Frequency')
plt.title('Bar Plot of Frequency of Statement Length')
plt.show()
```



[28]:
```
# Histogram of Frequency of Statements by Status
plt.figure(figsize=(12,8))

# get the unique status values and their counts
status_counts = df1['status'].value_counts()

# create the bar plot
ax = status_counts.plot(kind='bar')

# add the count labels on top of each bar
for i, v in enumerate(status_counts):
    ax.text(i, v, str(v), ha='center', va='bottom')

plt.title('Statement distribution by status')
plt.xlabel('Status')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```



[31]:
```
# Word Cloud Before Preprocessing
# Combine all statements into a single string
text = ' '.join(df1['statement'].dropna())

# Create a word cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').
 ↪generate(text)

# Display the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud Before Preprocessing')
plt.show()
```


Word Cloud Before Preprocessing
feel
want
know
now
life
people
time
thing
will
friend
anymore
work
help
much
make
go
depression
feeling
everything
someone
anyone
depressed
anything
parent
family
nothing
anxiety
relationship
everyone
trying
felt
scared
alone
really
even
die
tired
year
take
end
doctor
never
said
point
way
job
see
talk
good
fuck
m
one
pain
started
alway
need
tell
sometime
find
hope
kind
sorry
went
love
school
still
try
mind
wanted
able
seem
actually
today
sleep
don t
well
made
hate
lot
around
enough
back
come
put
either
already
yet
left
maybe
hard
live
wish
happy
told
bad
month
home
stop
mom
place
literally
understand
suicide
without
leave
hurt
tried
Thank
long
hour
post
sad
least
new
better
thought
care
problem
death
matter
guess
stress
give
fucking
talking
got
keep
something
thinking
recently
issue
wa
little
wrong
probably
kid
mean
anyway
fear
person
change
med
best
deal
normal
let
working
took
im
used
world
advice
ask
went
sure
symptom
experience
done
therapist
gone
idea
reason
week
night
hour
lost
found
head
almost
worse
ve
looking
house
hope
stuff
making
therapy
going
kill
living
right
look
bed
shit
brain
first
start
due
body
right now