eds(Theory Activity No. 1)

NAME: SAMADHAN SANJAY KENDHALE

ROLL no: cs2-38

BATCH: C22

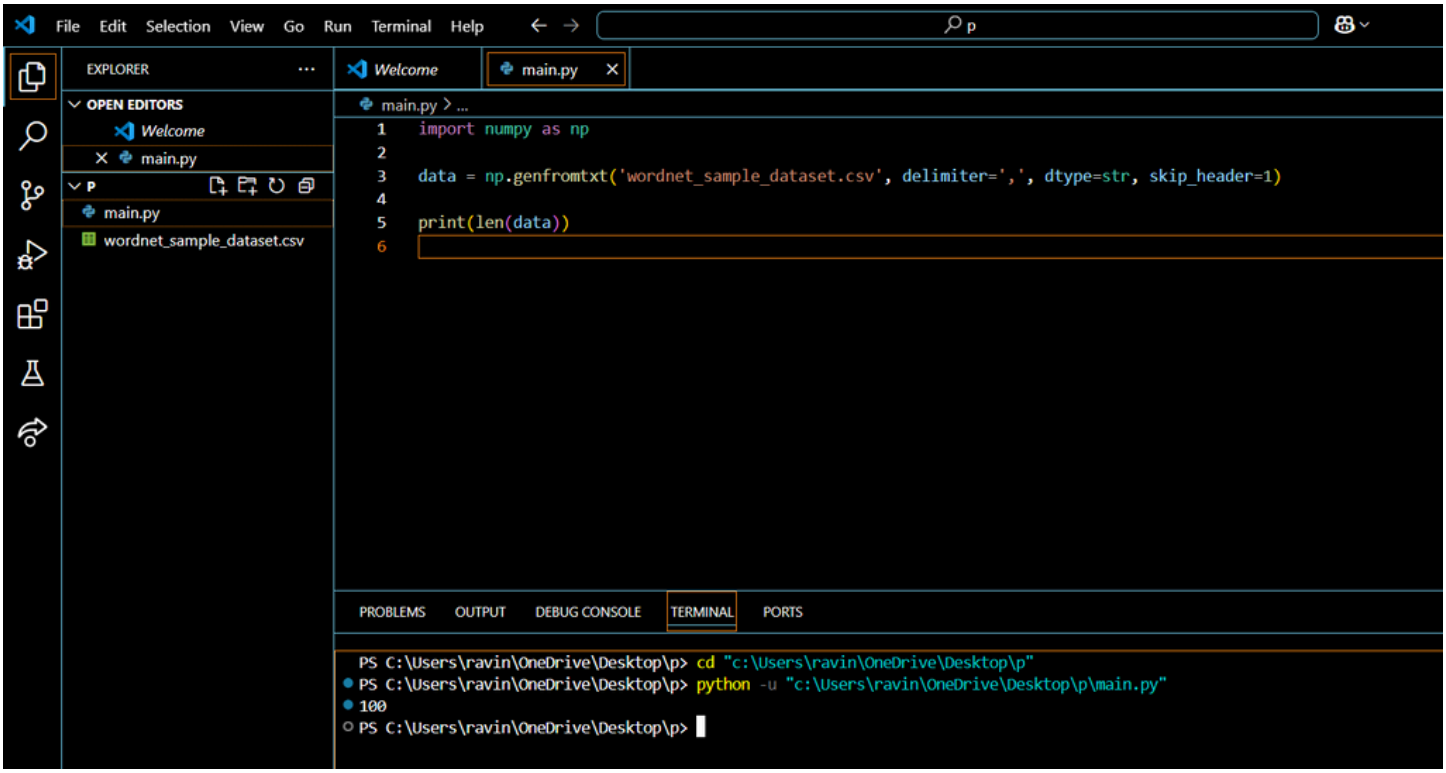
PRN: 202401040243

dataset: wordnet

numpy:

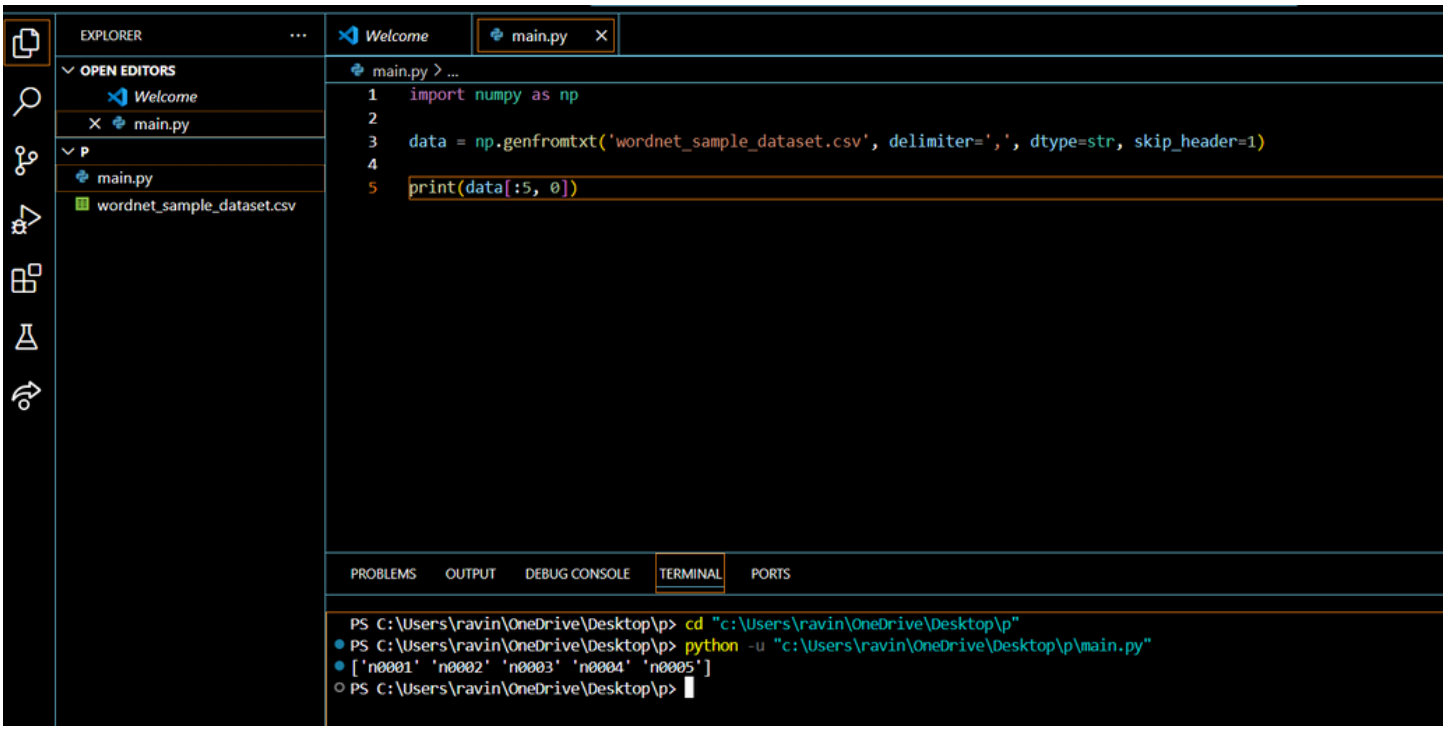
1. How many rows are there in the dataset?

solution:



2. Print the first 5 Synset Names.

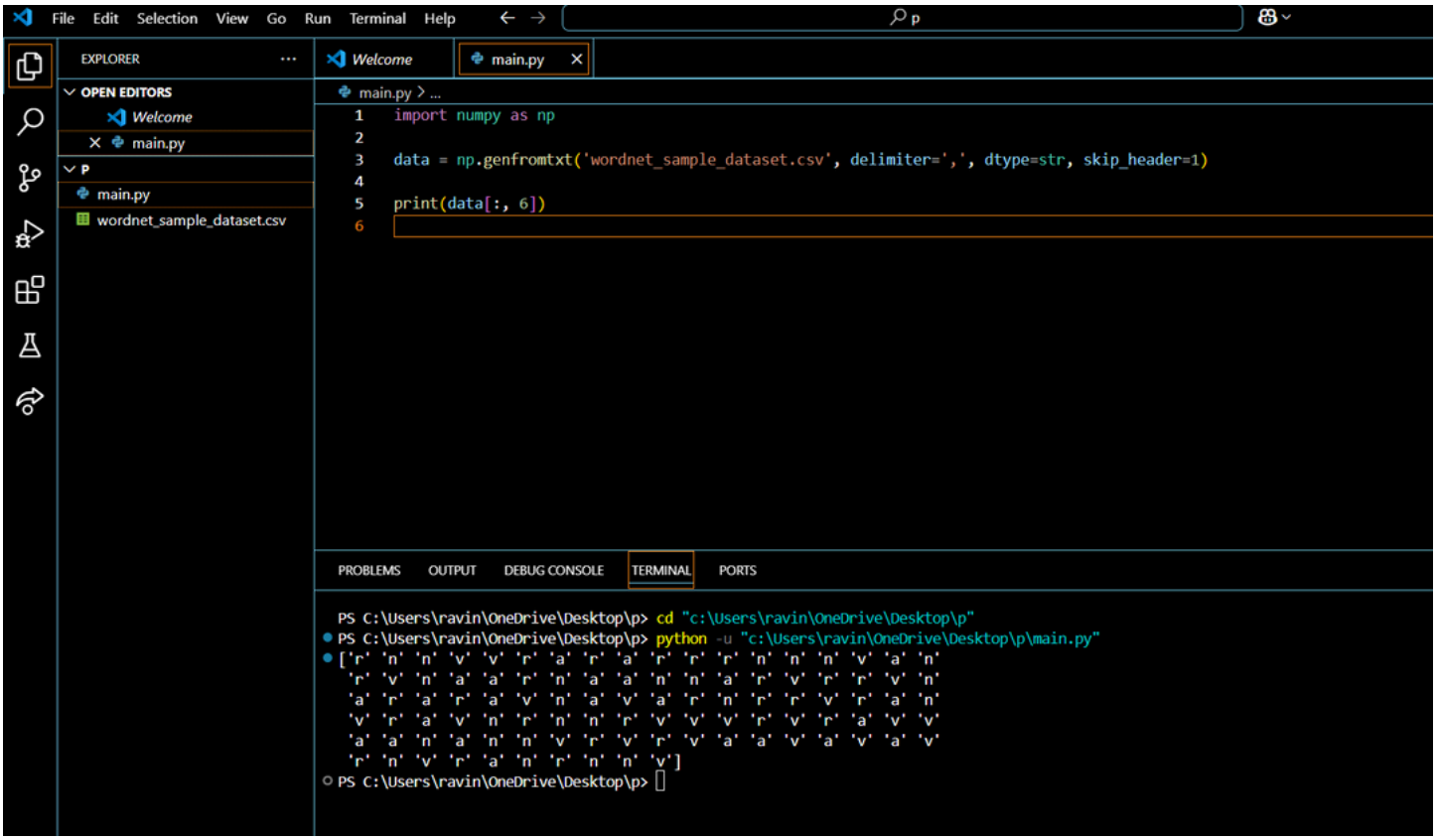
solution



print(data[:5, 0])

3. Print all POS tags.

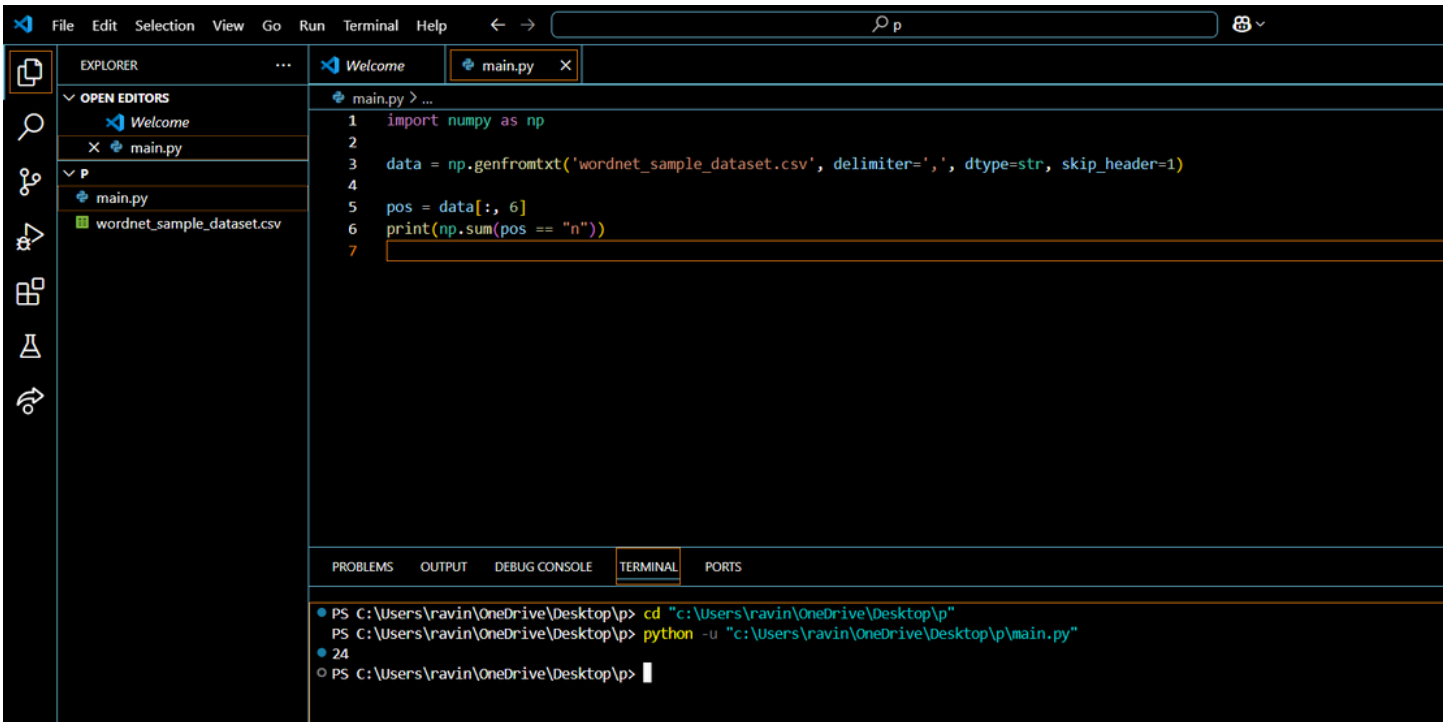
solution:



print(data[:, 6])

4. Find how many times "n" appears in the POS column.

solution:

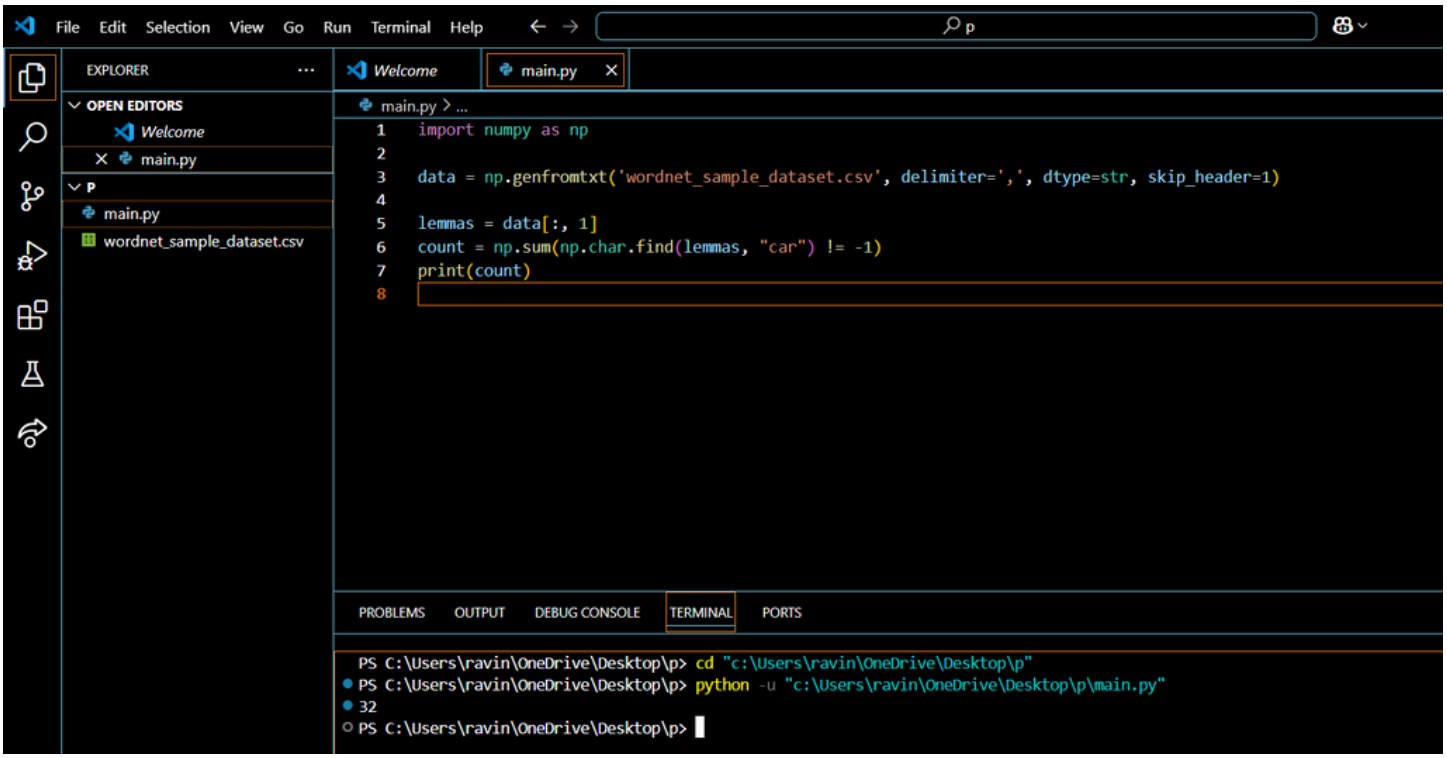


pos = data[:, 6]

print(np.sum(pos == "n"))

5. Find how many Lemmas contain the word "car".

solution:



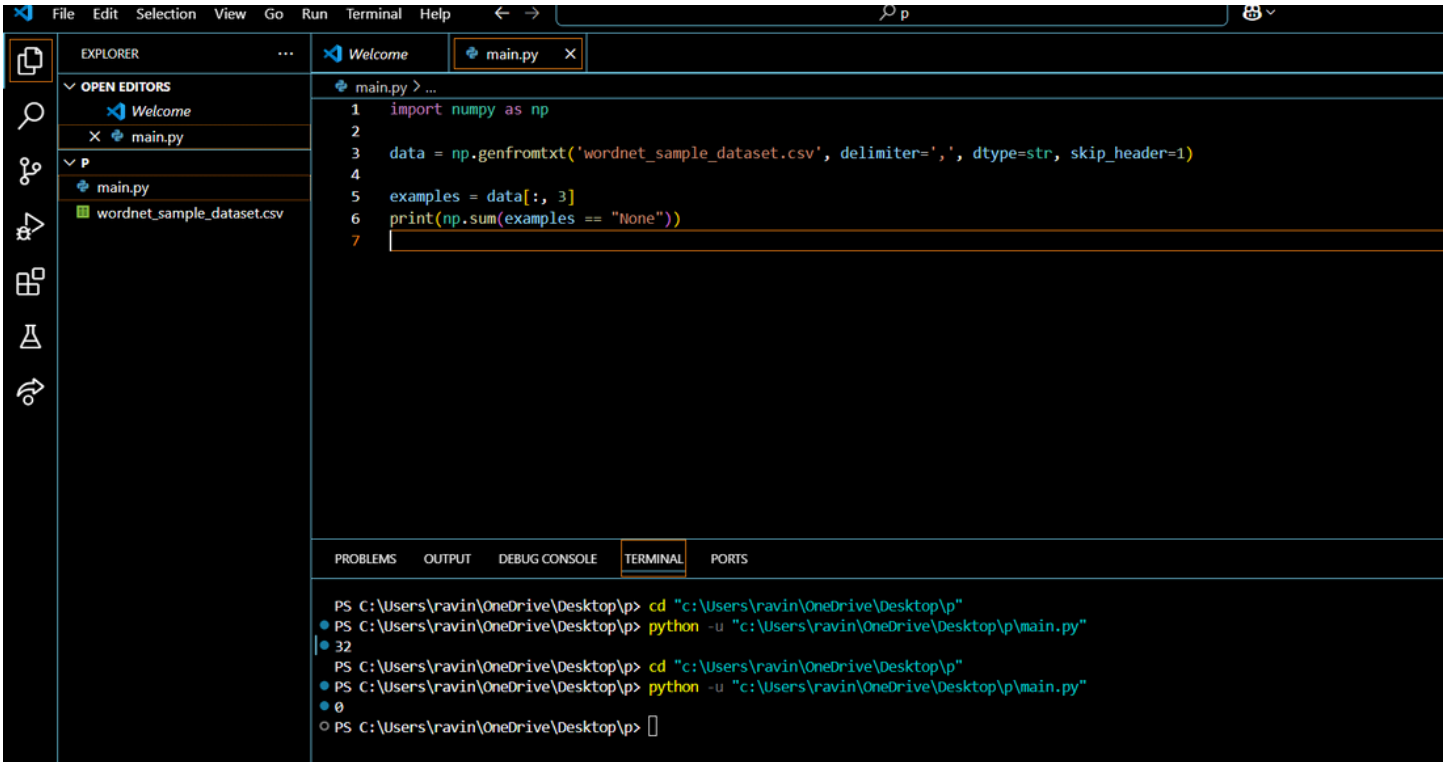
lemmas = data[:, 1]

count = np.sum(np.char.find(lemmas, "car") != -1)

print(count)

6. Find how many rows have "None" in Examples.

solution:

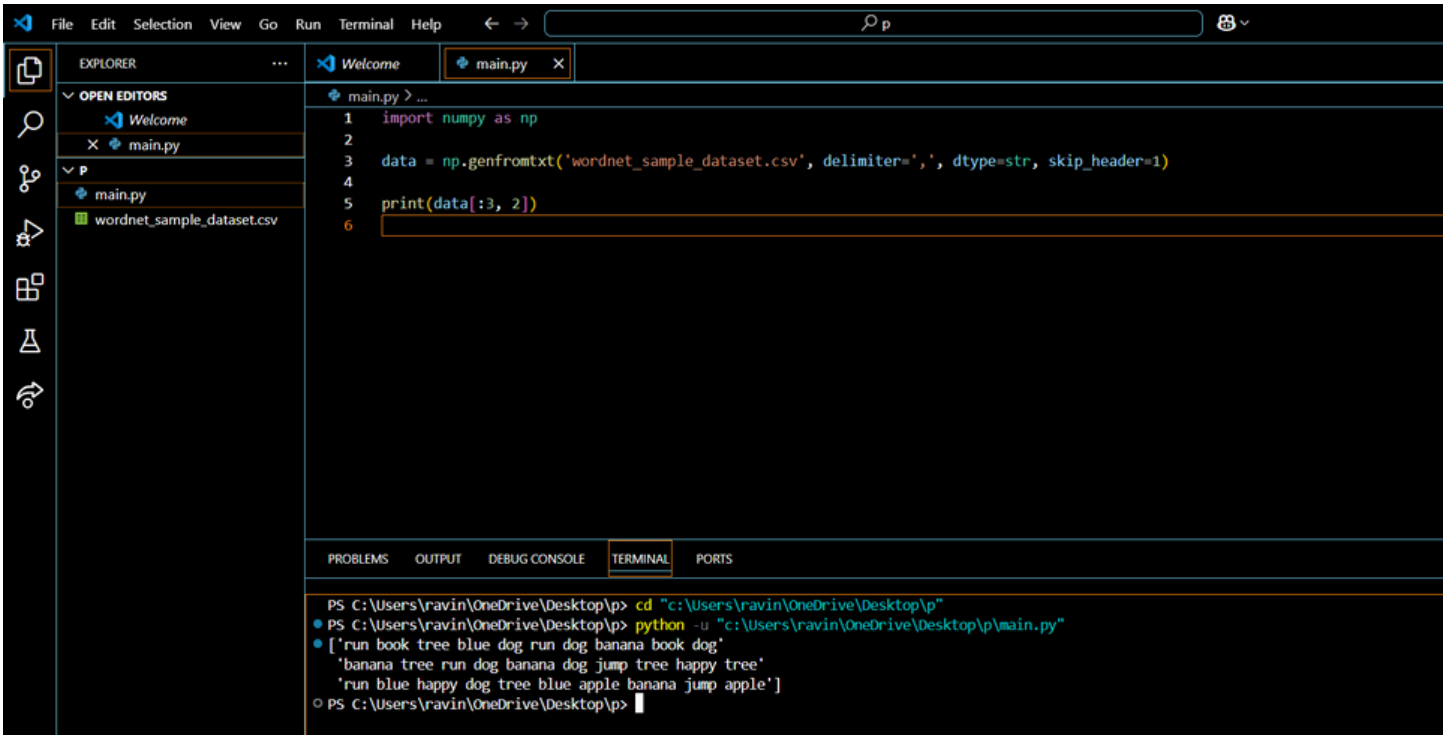


examples = data[:, 3]

print(np.sum(examples == "None"))

7. Print the first 3 Definitions.

solution:



print(data[:3, 2])

8. Print all unique POS tags.

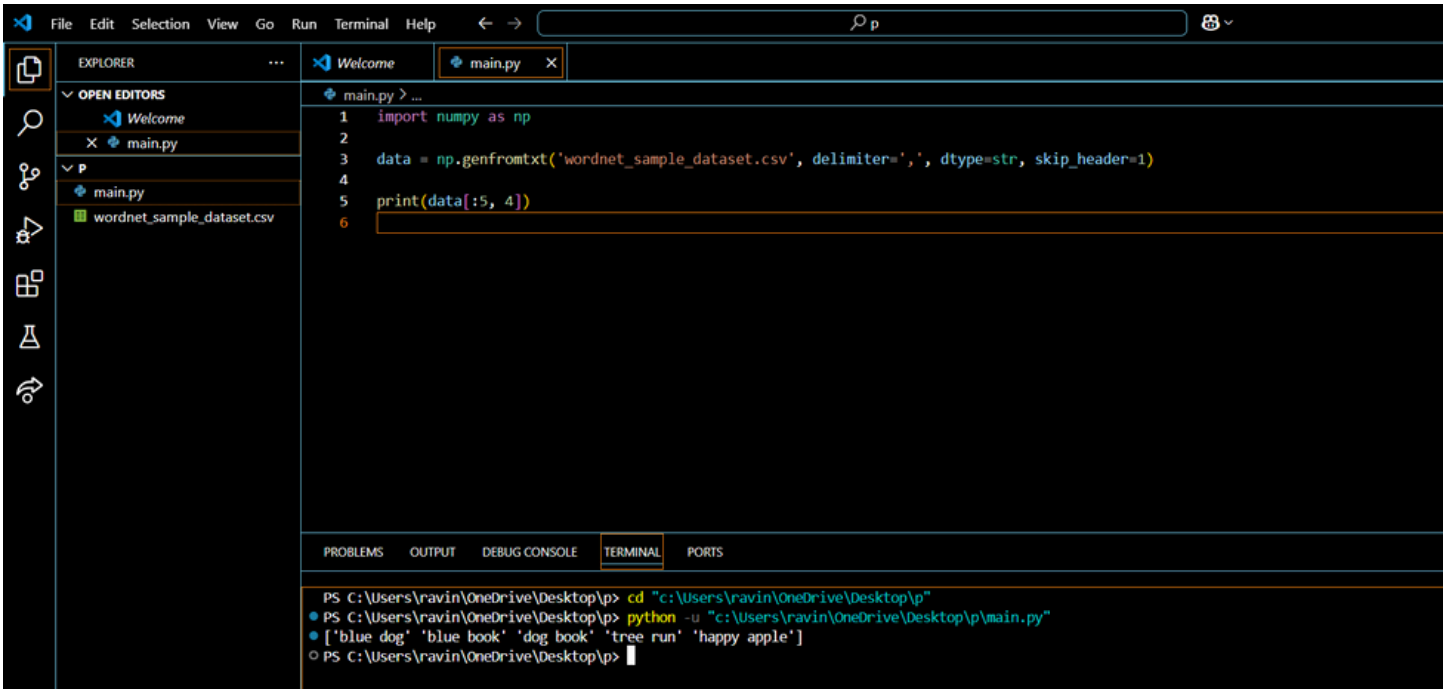
solution:

unique_tags = np.unique(data[:, 6])

print(unique_tags)

9. Print the first 5 Hypernyms.

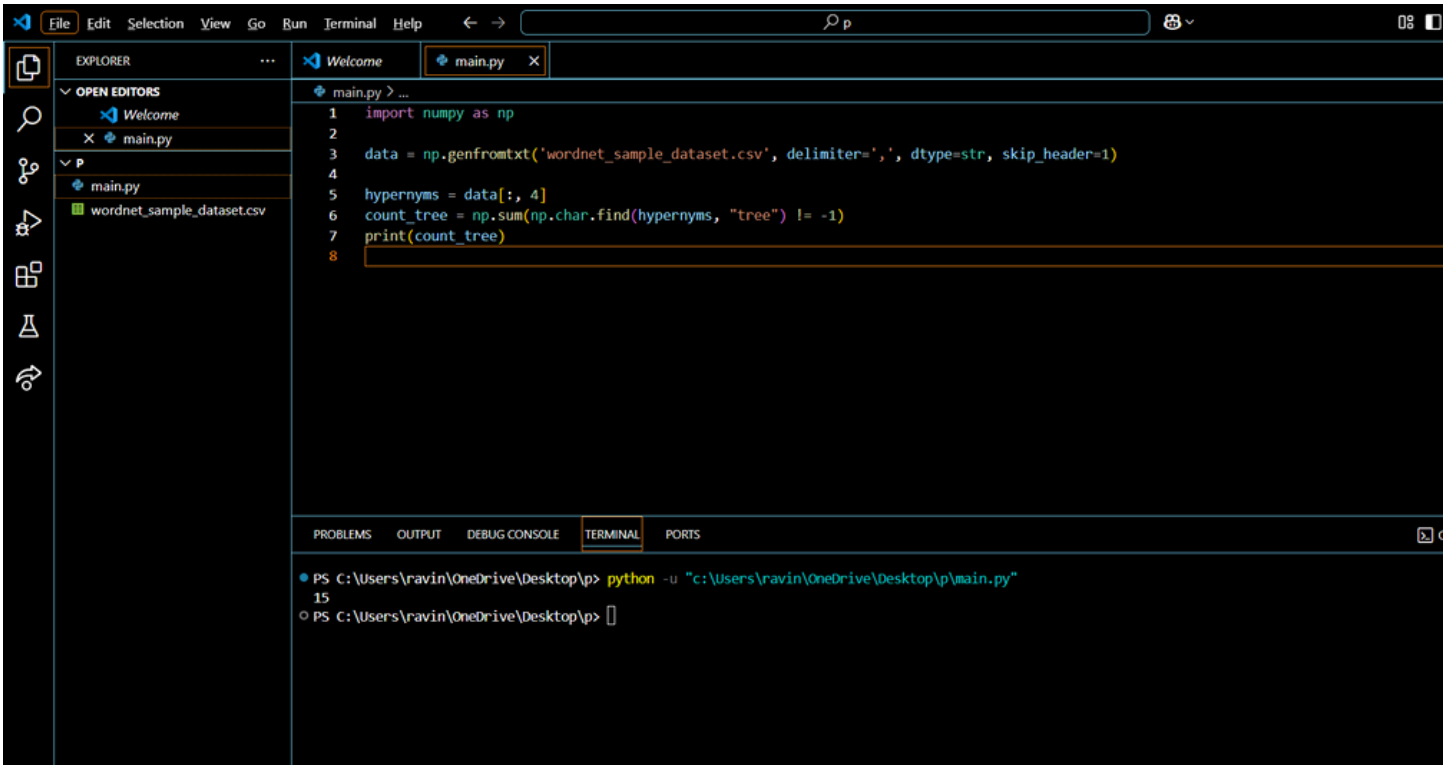
solution:



print(data[:5, 4])

10. How many rows have the word "tree" anywhere in Hypernyms?

solution:



hypernyms = data[:, 4]

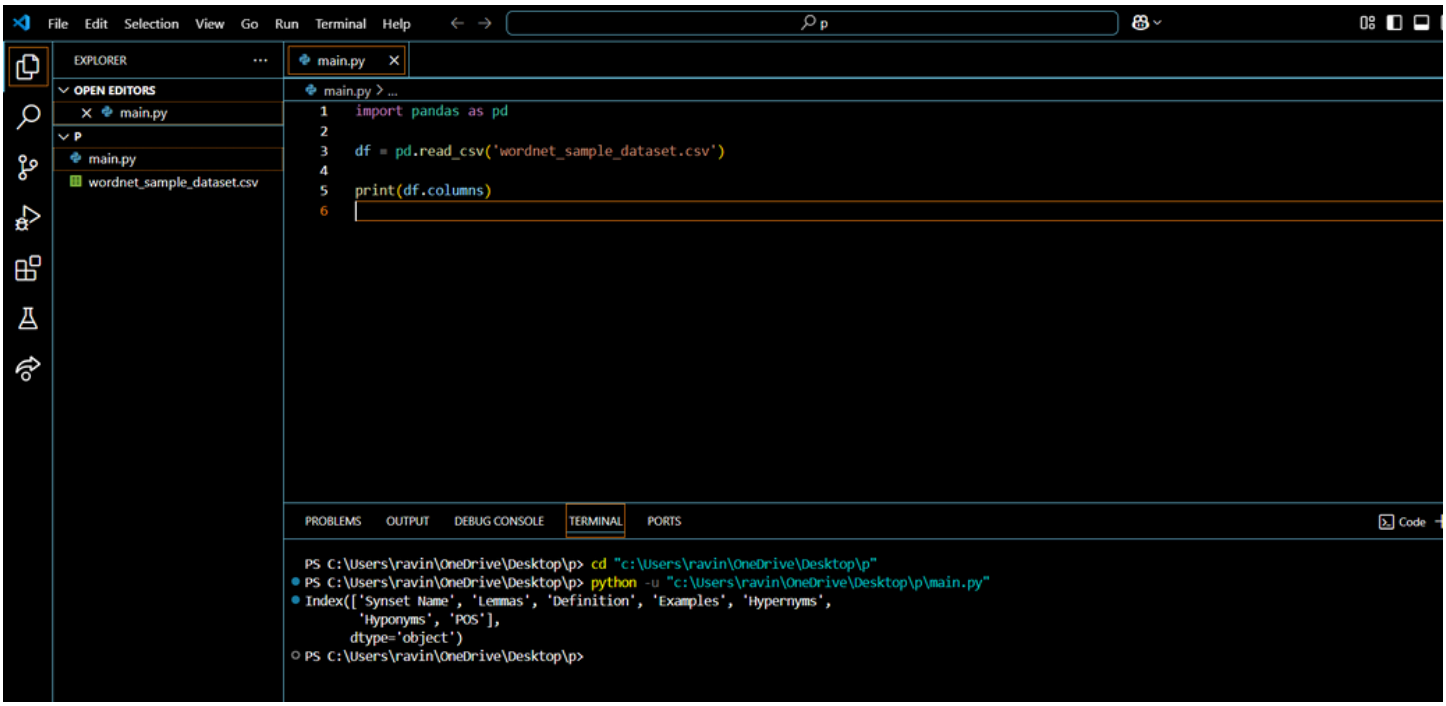
count_tree = np.sum(np.char.find(hypernyms, "tree") != -1)

print(count_tree)

pandas:

1. Show the column names of the dataset.

solution:

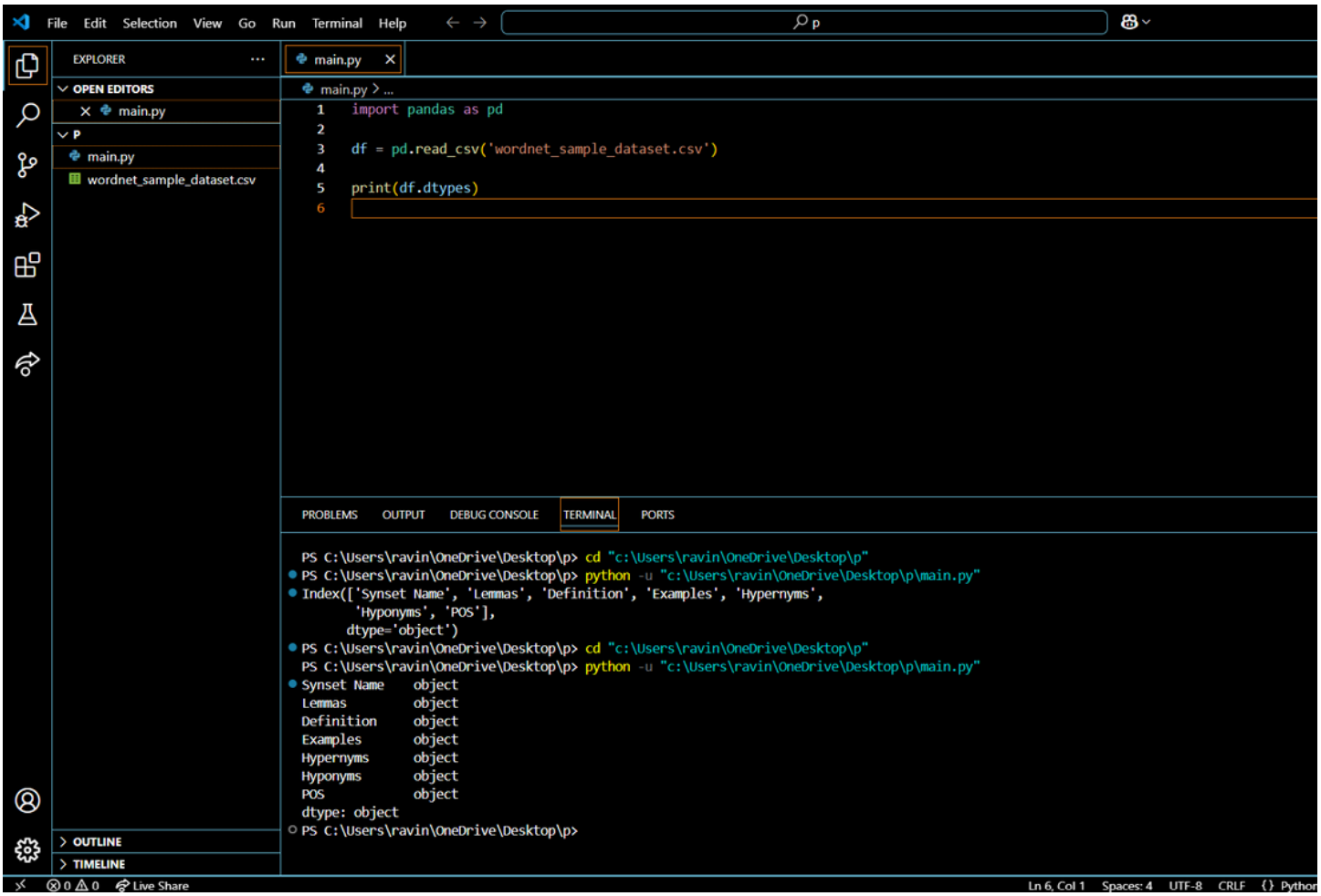


df = pd.read_csv('wordnet_sample_dataset.csv')

print(df.columns)

2. Show the datatype (dtype) of each column.

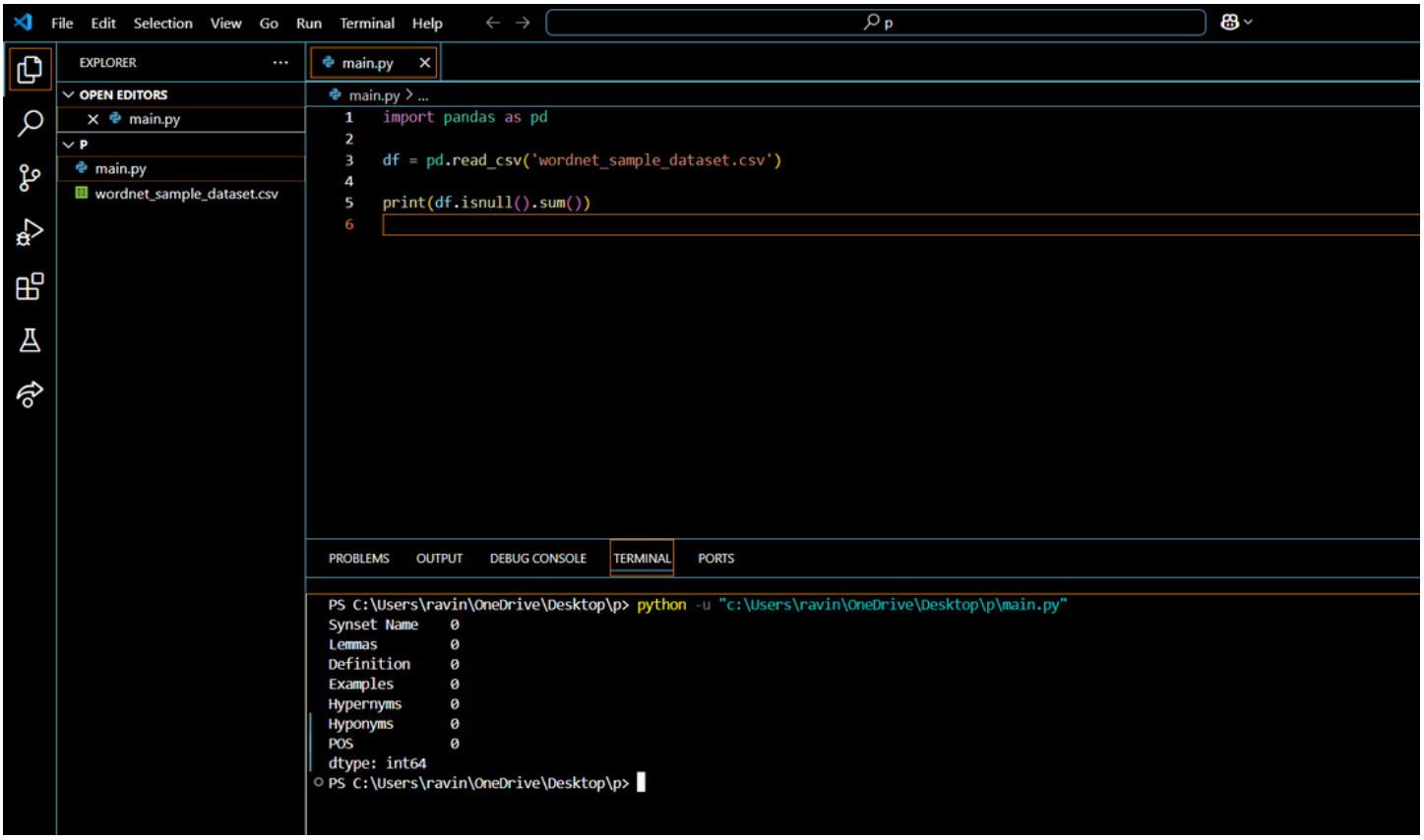
solution:



print(df.dtypes)

3. Find the number of missing (empty) values in each column.

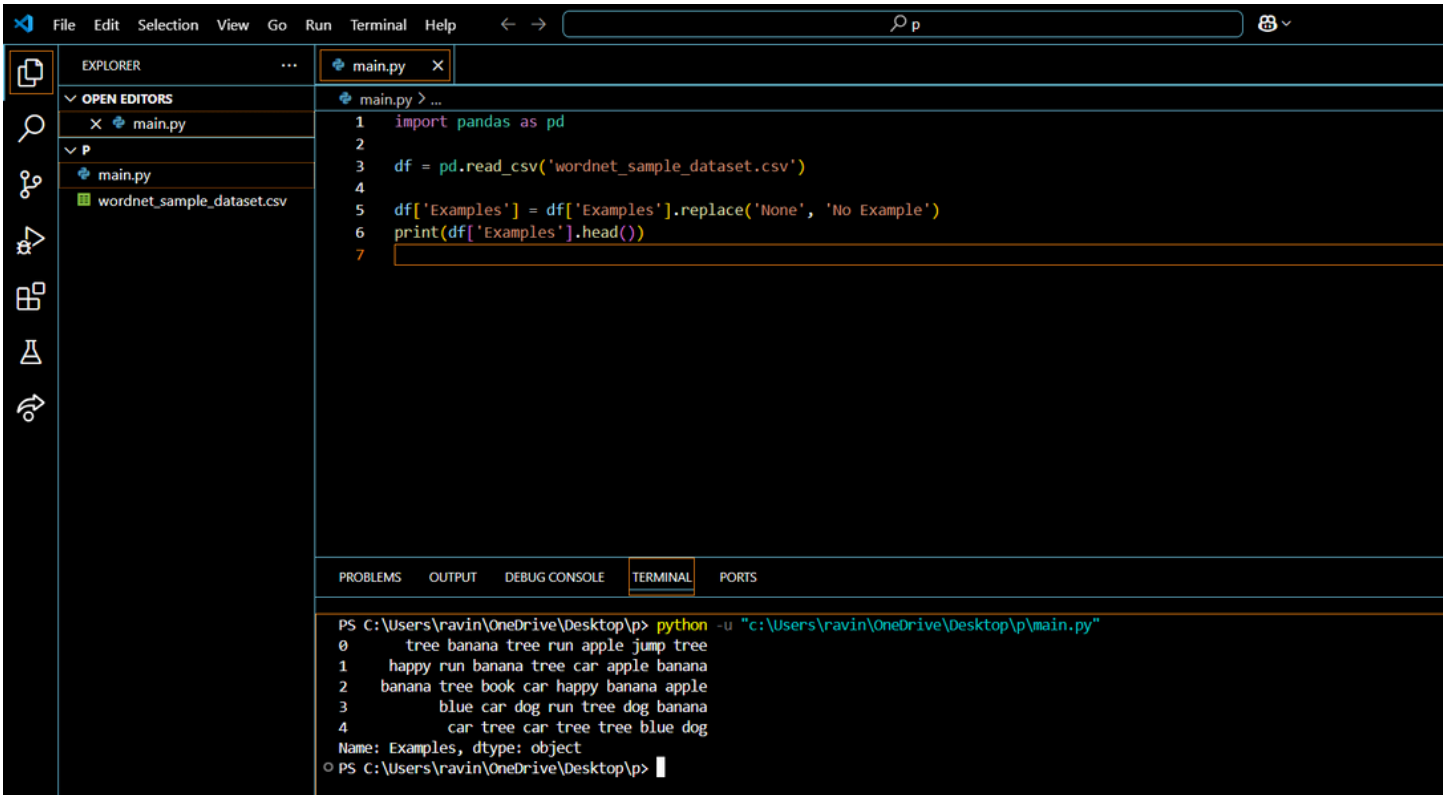
solution:



print(df.isnull().sum())

4. Replace all "None" values in Examples with "No Example".

solution:

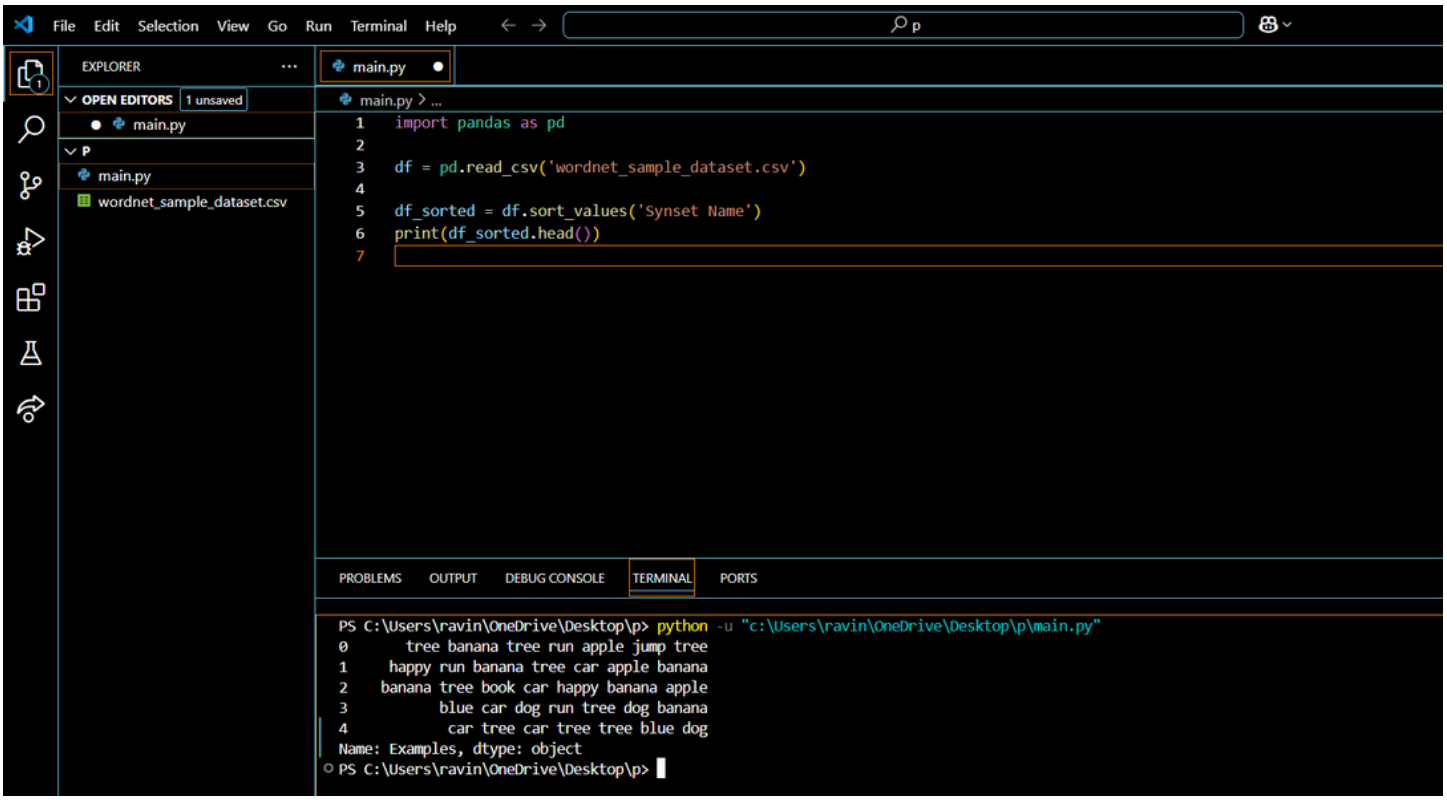


df['Examples'] = df['Examples'].replace('None', 'No Example')

print(df['Examples'].head())

5. Sort the dataset by "Synset Name" alphabetically.

solution:

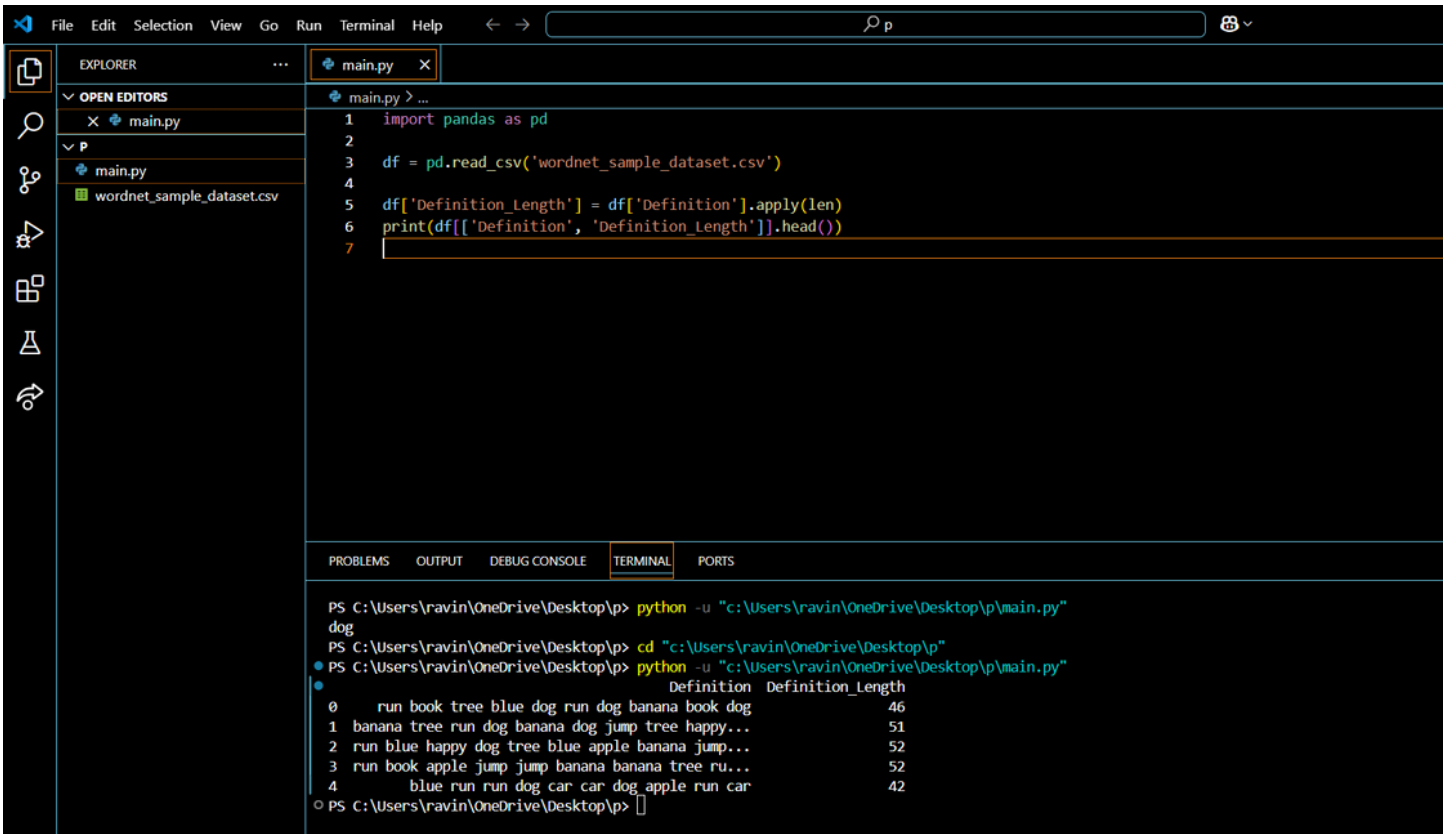


df_sorted = df.sort_values('Synset Name')

print(df_sorted.head())

6. Create a new column called "Definition_Length" that stores the number of characters in each definition.

solution:

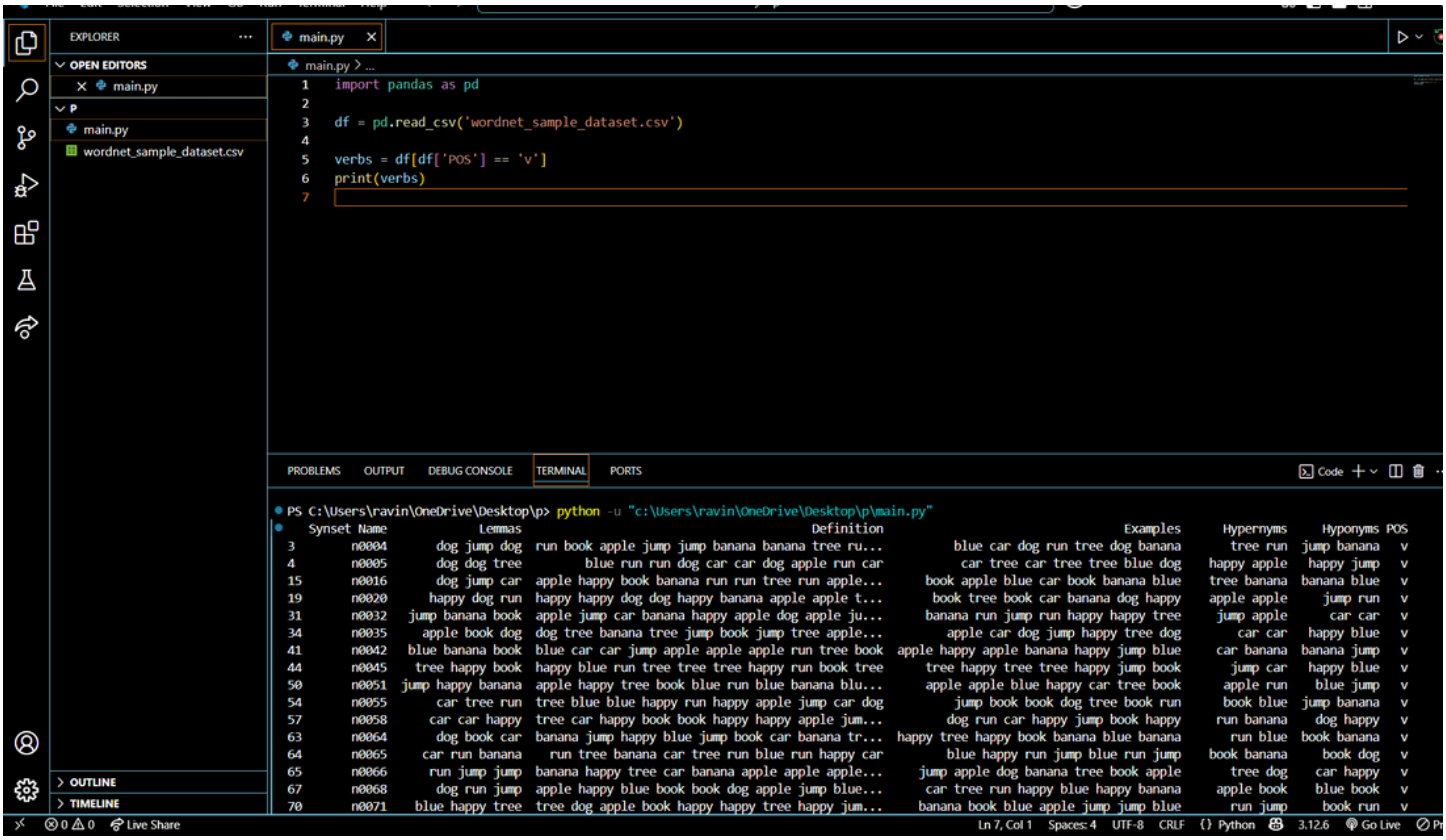


df['Definition_Length'] = df['Definition'].apply(len)

print(df[['Definition', 'Definition_Length']].head())

7. Filter and show all rows where POS is "v" (verb).

solution:

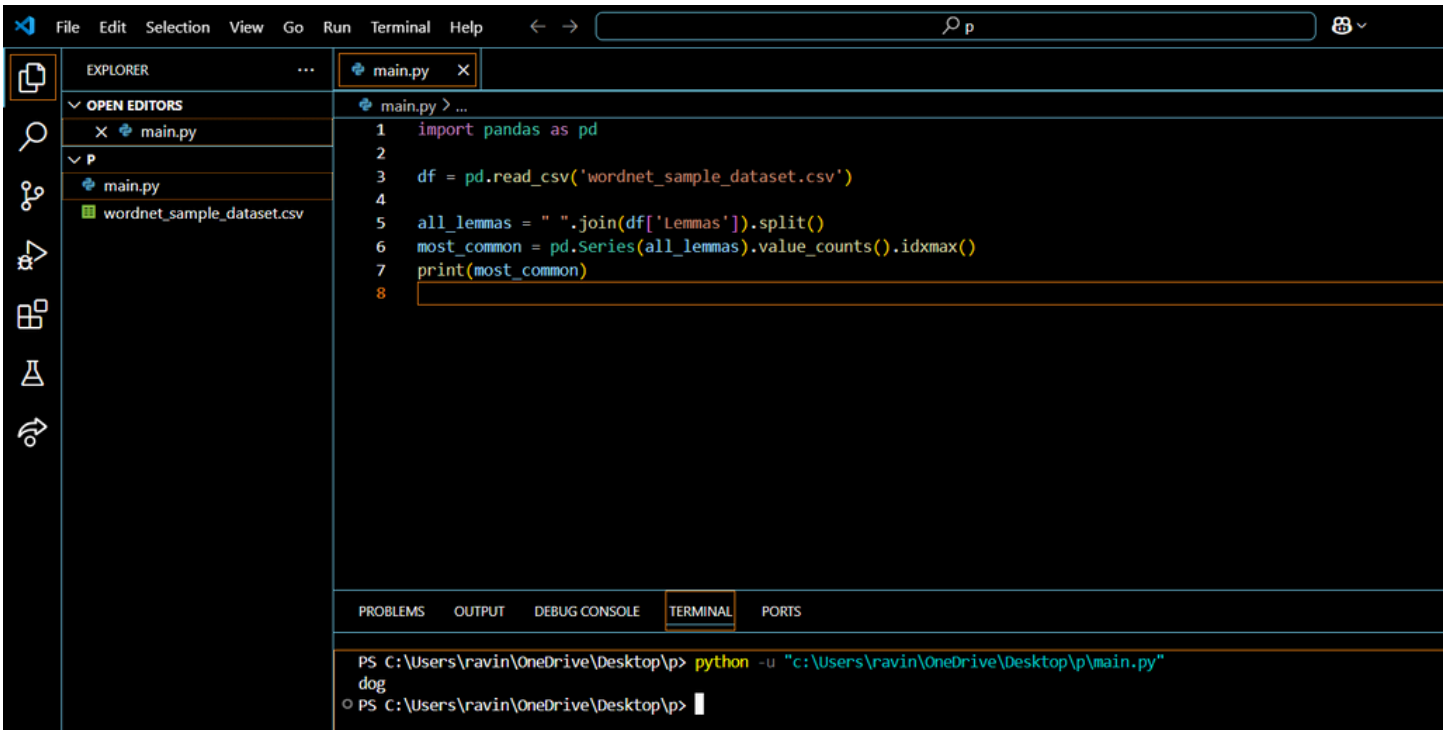


verbs = df[df['POS'] == 'v']

print(verbs)

8. Find the most common word appearing in Lemmas.

solution:



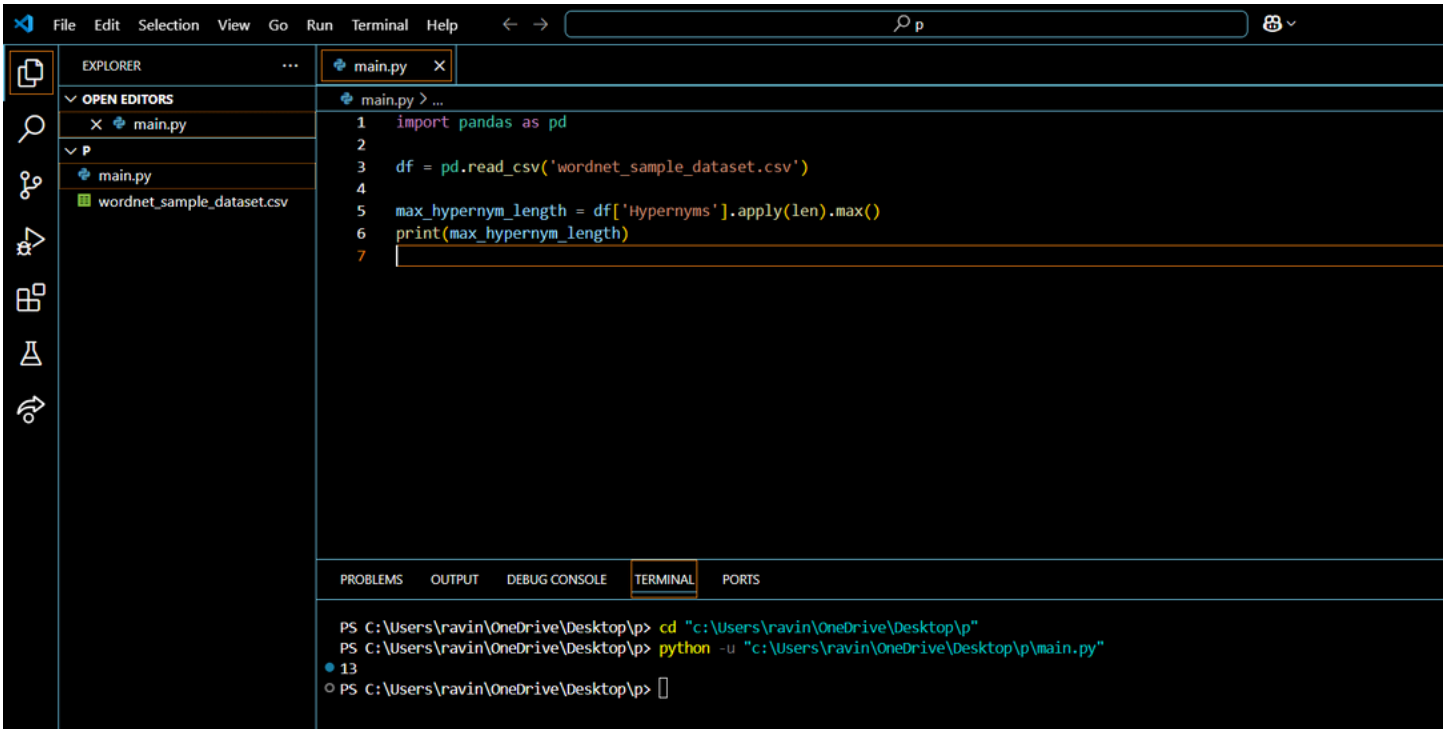
all_lemmas = " ".join(df['Lemmas']).split()

most_common = pd.Series(all_lemmas).value_counts().idxmax()

print(most_common)

9. Find the maximum length of the Hypernyms field.

solution:

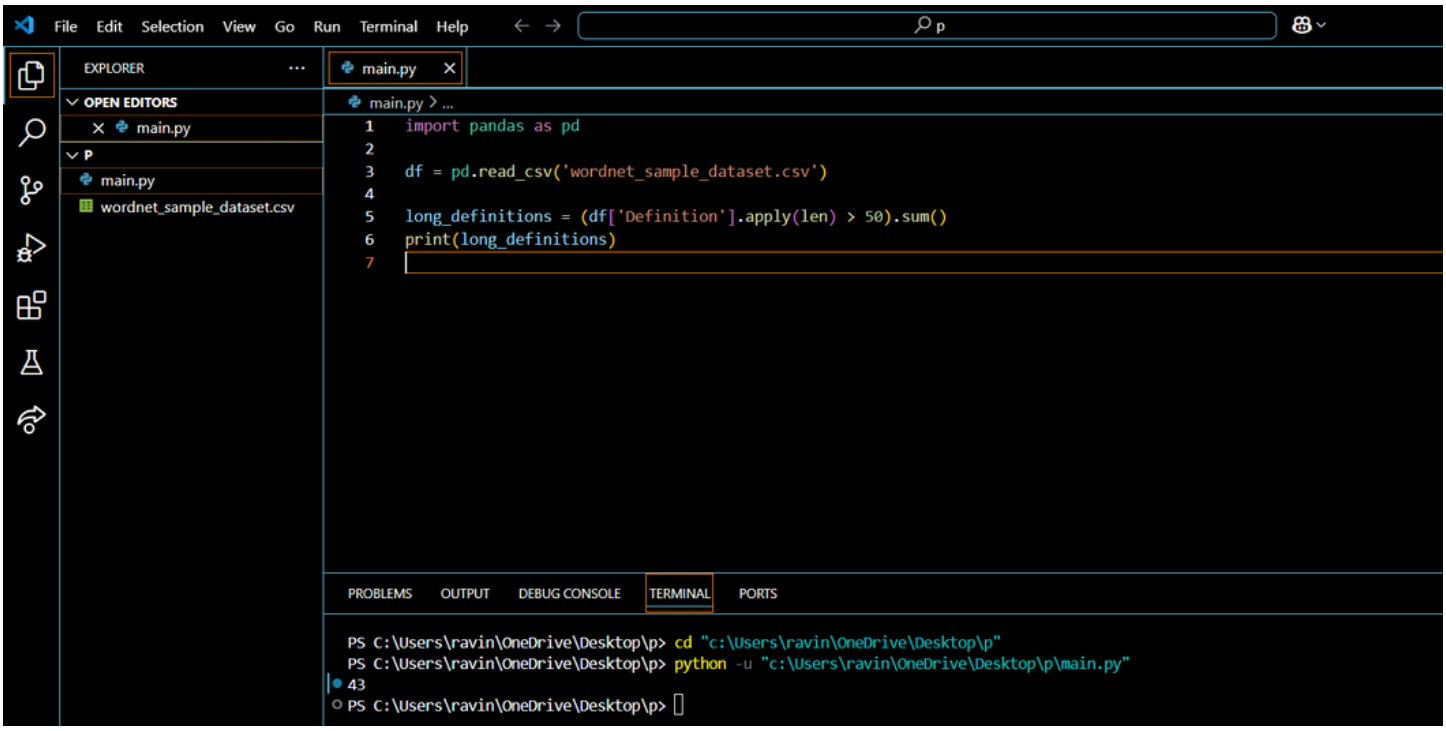


max_hypernym_length = df['Hypernyms'].apply(len).max()

print(max_hypernym_length)

10. Find how many Definitions have more than 50 characters.

solution:



long_definitions = (df['Definition'].apply(len) > 50).sum()

print(long_definitions)

---end---