

EDS Activity:-

Name: Chinmay Ghag

Roll no.:CS7-28

PRN:202401080073

The screenshot shows the Visual Studio Code interface with a Python file named `import pandas as pd.py` open. The code imports `pandas` as `pd` and `numpy` as `np`, and imports `Counter` from `collections`. It then loads a CSV file from `C:/Users/user/Downloads/archive/mail_data.csv` into a DataFrame `df`. The script prints basic information about the dataset, including the number of rows and columns, and displays the first 5 rows of the data.

```
C:\Users\user> import pandas as pd.py > ...
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Display basic information
9 print("Basic Information about the Dataset:")
10 print(df.info())
11 print("\nfirst 5 rows:")
12 print(df.head())
```

The terminal output shows the result of `df.info()`, indicating 5572 non-null objects in a DataFrame with 2 dtypes and a memory usage of 87.2+ KB. The first 5 rows of the data are displayed, showing columns for `Category` and `Message`.

| | Category | Message |
|---|----------|---|
| 0 | ham | Go until jurong point, crazy.. Available only ... |
| 1 | ham | Ok lar... Joking wif u oni... |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 3 | ham | U dun say so early hor... U c already then say... |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... |

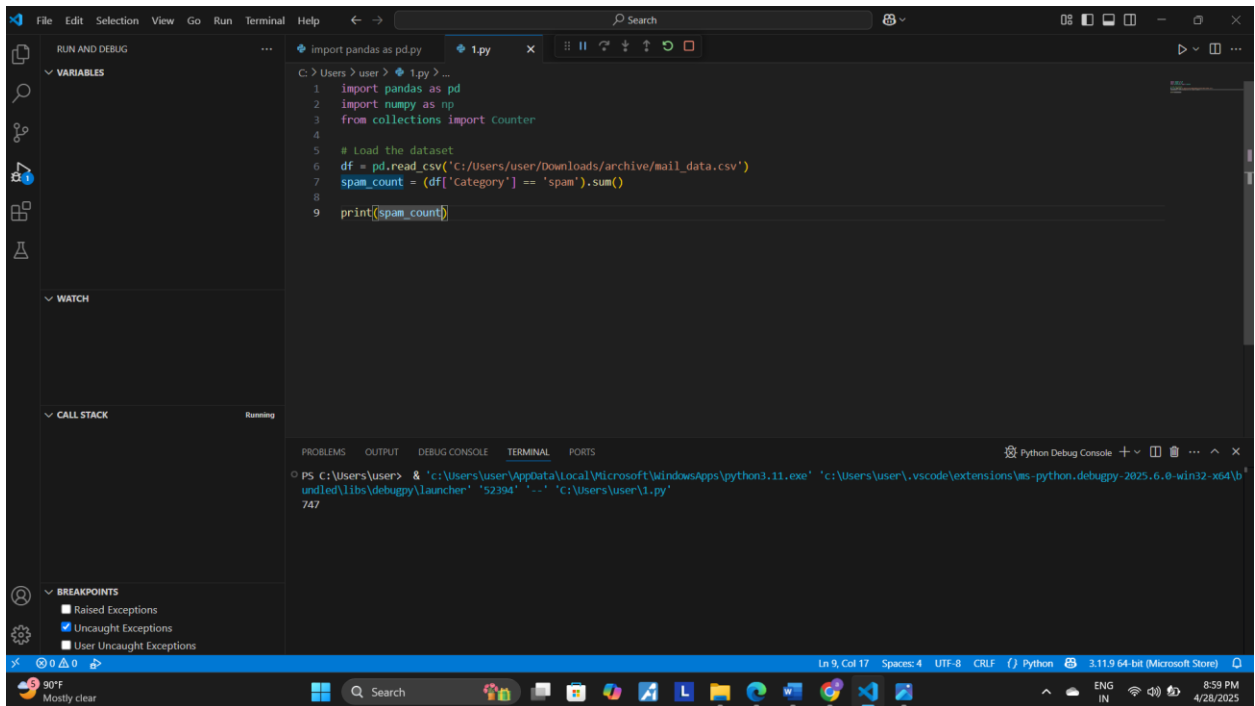
1. How many total messages are there?

The screenshot shows the Visual Studio Code interface with a Python file named `1.py` open. The code imports `pandas` as `pd` and `numpy` as `np`, and imports `Counter` from `collections`. It then loads a CSV file from `C:/Users/user/Downloads/archive/mail_data.csv` into a DataFrame `df`. The script calculates the total number of messages using `len(df)` and prints the result.

```
C:\Users\user> 1.py > ...
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7 total_messages = len(df)
8 print(total_messages)
```

The terminal output shows the command prompt running the script, which outputs the number 5572, representing the total number of messages in the dataset.

2. How many messages are spam?



The screenshot shows the Visual Studio Code interface with a Python file named `1.py` open. The code is as follows:

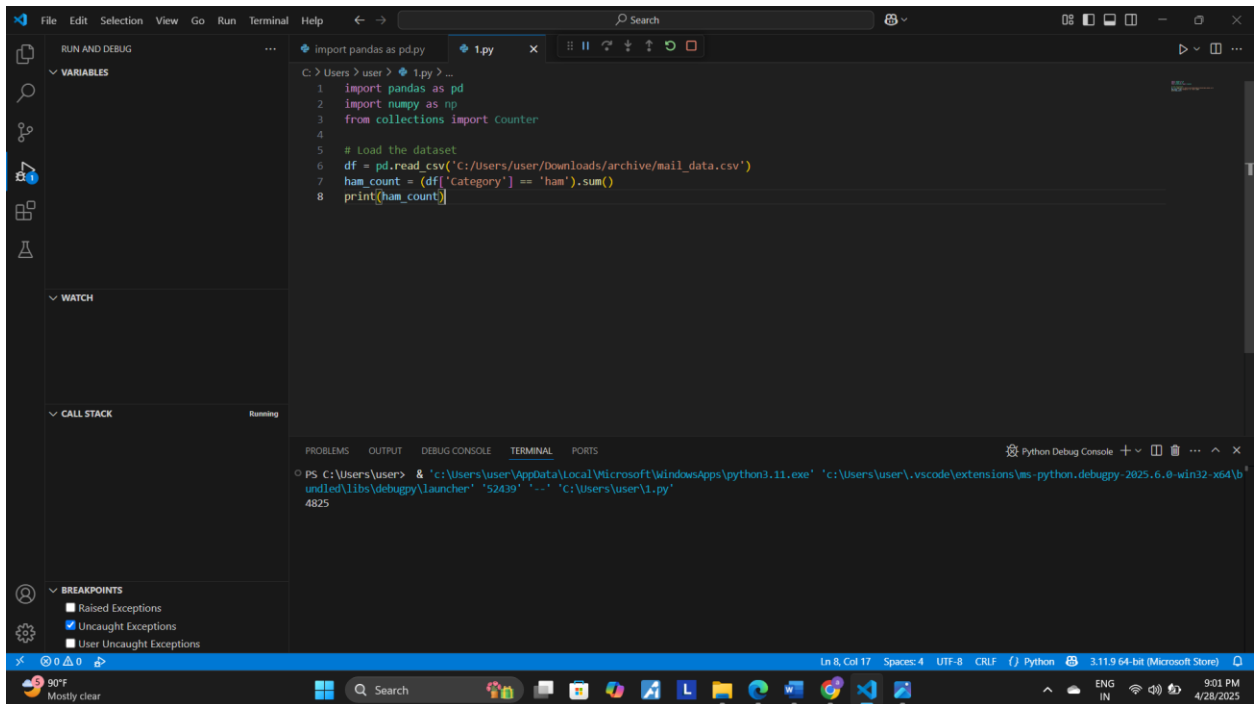
```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7 spam_count = (df['Category'] == 'spam').sum()
8
9 print(spam_count)
```

The left sidebar shows the 'RUN AND DEBUG' panel with 'VARIABLES', 'WATCH', 'CALL STACK', and 'BREAKPOINTS' sections. The 'BREAKPOINTS' section is expanded, showing 'Uncaught Exceptions' checked. The bottom status bar indicates 'Ln 9, Col 17'.

The terminal at the bottom shows the command to run the script:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy\launcher' '52394' '-' 'c:\Users\user\1.py'
```

3. How many messages are ham?



The screenshot shows the Visual Studio Code interface with a Python file named `1.py` open. The code is as follows:

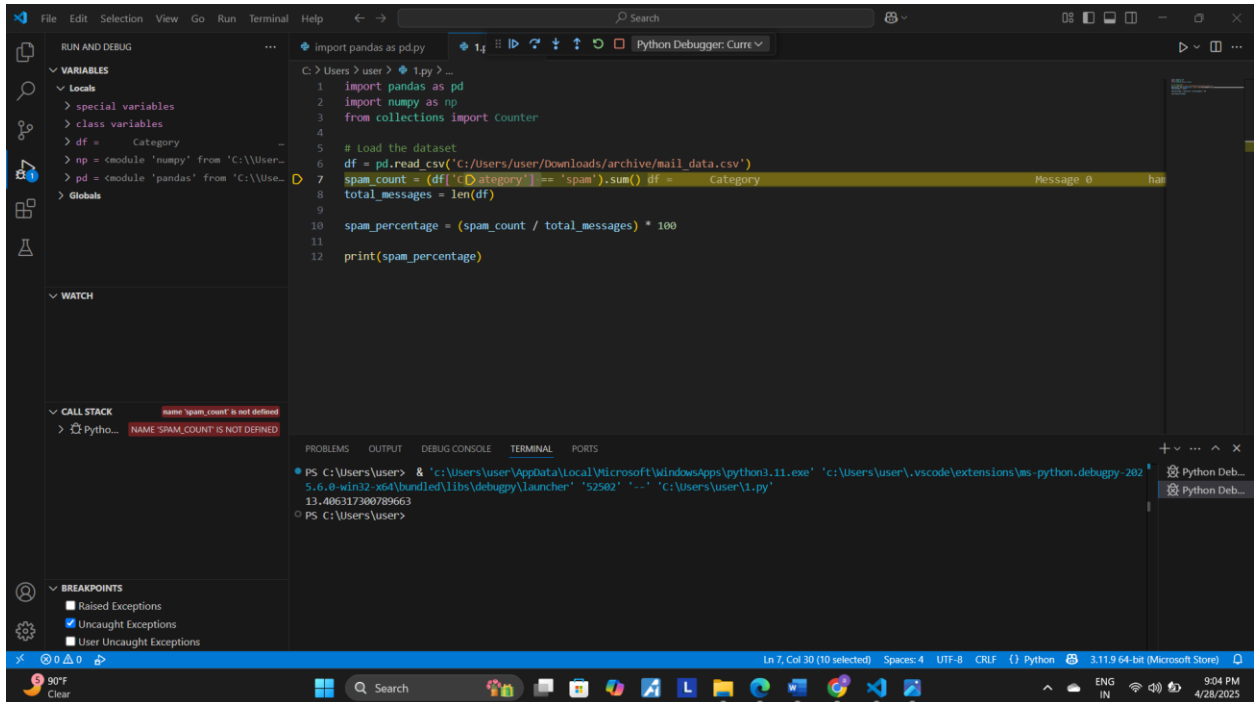
```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7 ham_count = (df['Category'] == 'ham').sum()
8
9 print(ham_count)
```

The left sidebar shows the 'RUN AND DEBUG' panel with 'VARIABLES', 'WATCH', 'CALL STACK', and 'BREAKPOINTS' sections. The 'BREAKPOINTS' section is expanded, showing 'Uncaught Exceptions' checked. The bottom status bar indicates 'Ln 8, Col 17'.

The terminal at the bottom shows the command to run the script:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy\launcher' '52439' '-' 'c:\Users\user\1.py'
```

4. What is the percentage of spam messages?



The screenshot shows the Visual Studio Code editor with a Python file named `1.py`. The script imports `pandas` and `numpy`, reads a CSV file from `C:/Users/user/Downloads/archive/mail_data.csv`, and calculates the percentage of spam messages. The `DEBUG CONSOLE` shows the execution of the script, and the `TERMINAL` shows the command prompt output.

```
import pandas as pd
import numpy as np
from collections import Counter

# Load the dataset
df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
spam_count = (df['category'] == 'spam').sum()
total_messages = len(df)

spam_percentage = (spam_count / total_messages) * 100
print(spam_percentage)
```

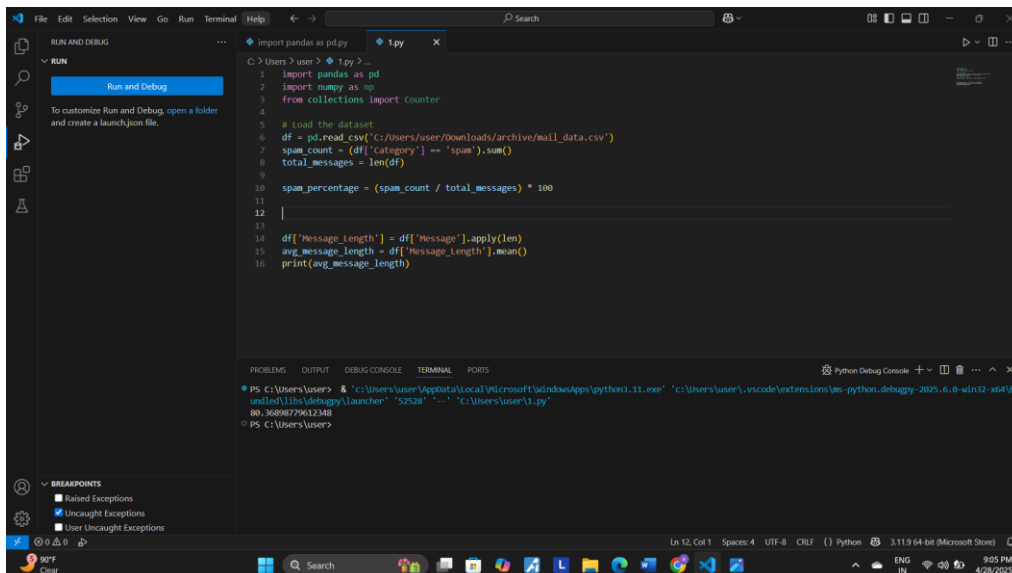
DEBUG CONSOLE:

```
Python Debugger: C:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe
5.6.0-win32-x64\lib\debugpy\launcher '52502' ... 'C:\Users\user\1.py'
13.406317300789663
```

TERMINAL:

```
PS C:\Users\user> & 'C:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\lib\debugpy\launcher' '52502' '...' 'C:\Users\user\1.py'
PS C:\Users\user>
```

5. What is the average length of message?



The screenshot shows the Visual Studio Code editor with a Python file named `1.py`. The script imports `pandas` and `numpy`, reads a CSV file from `C:/Users/user/Downloads/archive/mail_data.csv`, and calculates the average length of messages. The `DEBUG CONSOLE` shows the execution of the script, and the `TERMINAL` shows the command prompt output.

```
import pandas as pd
import numpy as np
from collections import Counter

# Load the dataset
df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
spam_count = (df['category'] == 'spam').sum()
total_messages = len(df)

spam_percentage = (spam_count / total_messages) * 100
print(spam_percentage)

df['message_length'] = df['message'].apply(len)
avg_message_length = df['message_length'].mean()
print(avg_message_length)
```

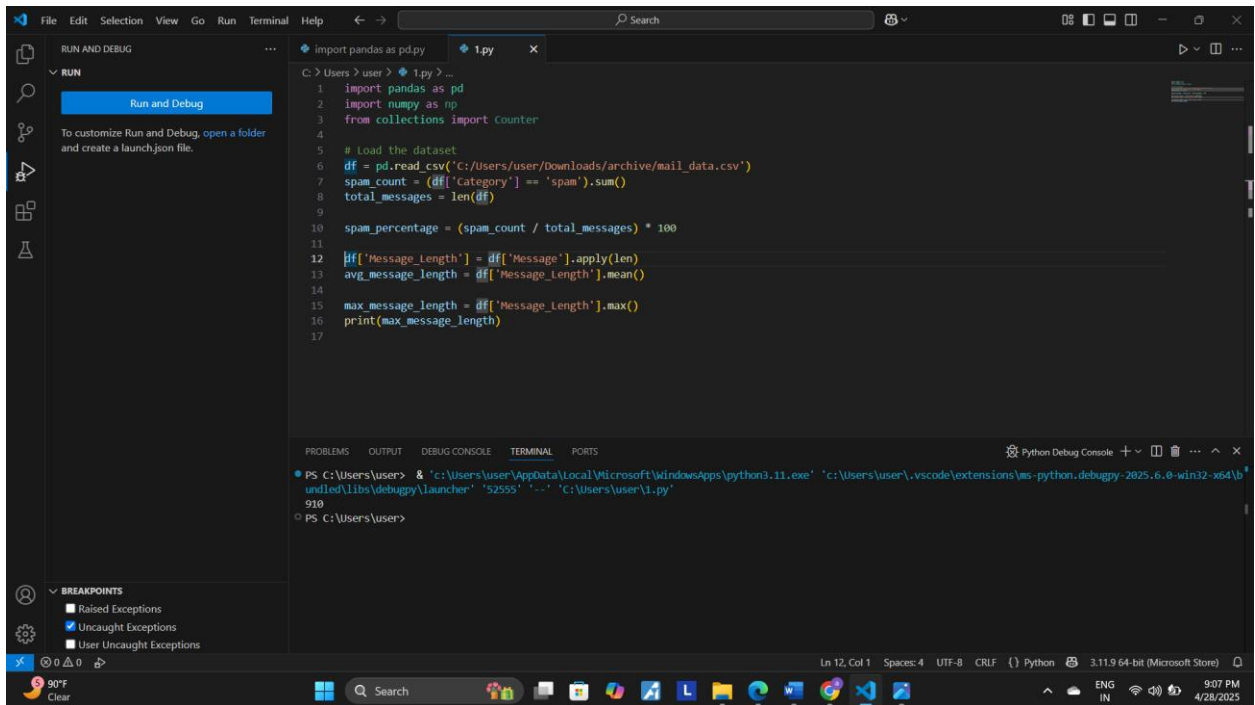
DEBUG CONSOLE:

```
Python Debugger: C:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe
5.6.0-win32-x64\lib\debugpy\launcher '52502' ... 'C:\Users\user\1.py'
13.406317300789663
```

TERMINAL:

```
PS C:\Users\user> & 'C:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\lib\debugpy\launcher' '52502' '...' 'C:\Users\user\1.py'
PS C:\Users\user>
```

6. What is the maximum message length?



The screenshot shows a Visual Studio Code editor window with a Python file named `1.py`. The script reads a CSV file, filters for spam messages, and calculates the maximum message length. The Run and Debug panel on the left shows the 'Run and Debug' button. The bottom panel shows the terminal output of the script execution.

```
import pandas as pd
import numpy as np
from collections import Counter

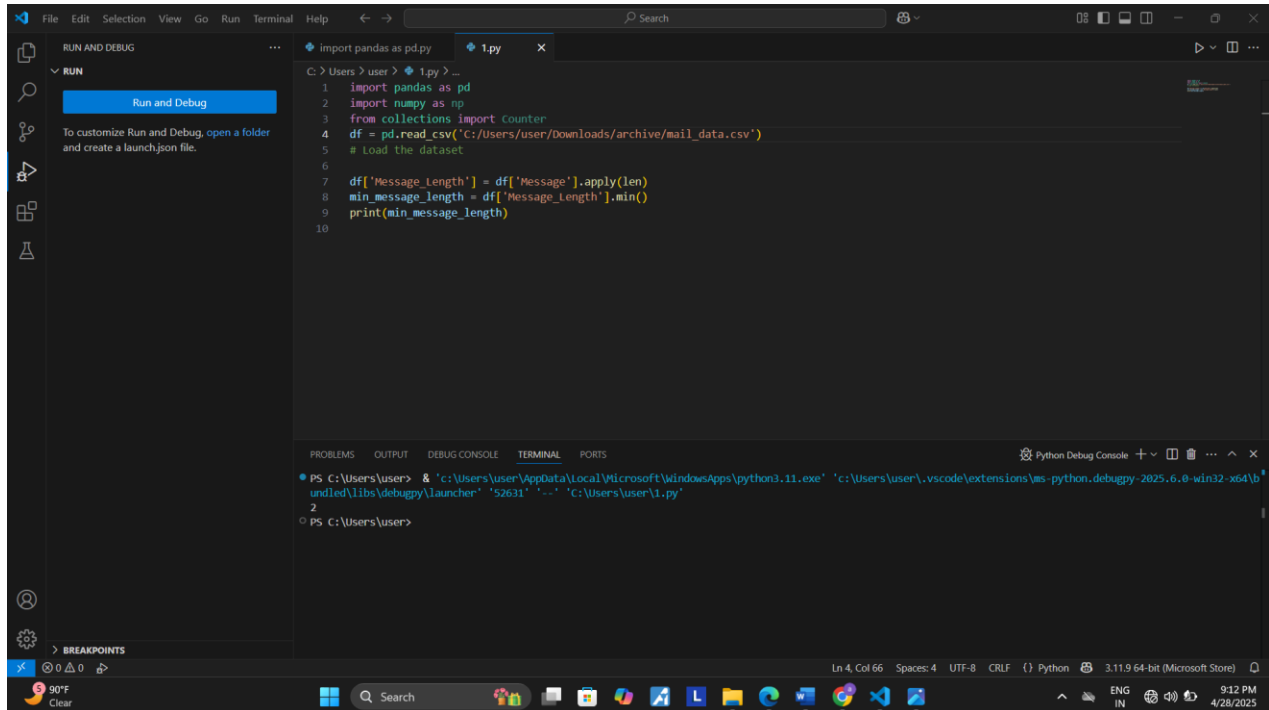
# Load the dataset
df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
spam_count = (df['category'] == 'spam').sum()
total_messages = len(df)

spam_percentage = (spam_count / total_messages) * 100

df['Message_Length'] = df['Message'].apply(len)
avg_message_length = df['Message_Length'].mean()
max_message_length = df['Message_Length'].max()
print(max_message_length)
```

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy_launcher' '52555' '-.' 'C:\Users\user\1.py'
910
PS C:\Users\user>
```

7. What is the minimum message length?



The screenshot shows the Visual Studio Code interface. The editor window displays a Python script named `1.py` with the following code:

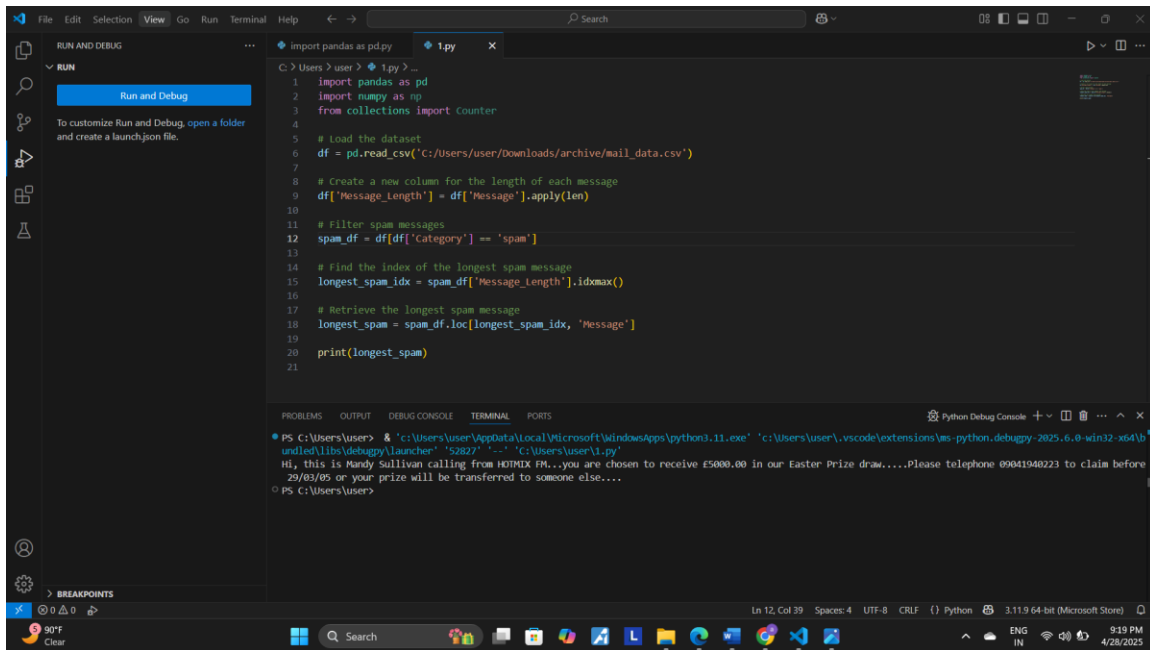
```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4 df = pd.read_csv("C:/Users/user/Downloads/archive/mail_data.csv")
5 # load the dataset
6
7 df['Message_length'] = df['Message'].apply(len)
8 min_message_length = df['Message_length'].min()
9 print(min_message_length)
10
```

The `TERMINAL` panel at the bottom shows the execution output:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy_launcher' '52631' '--' 'C:\Users\user\1.py'
2
PS C:\Users\user>
```

The status bar at the bottom indicates the file is at line 4, column 66, with 4 spaces, UTF-8 encoding, CRLF line endings, and is a Python 3.11.9 64-bit file from the Microsoft Store.

8. What is the longest spam message?



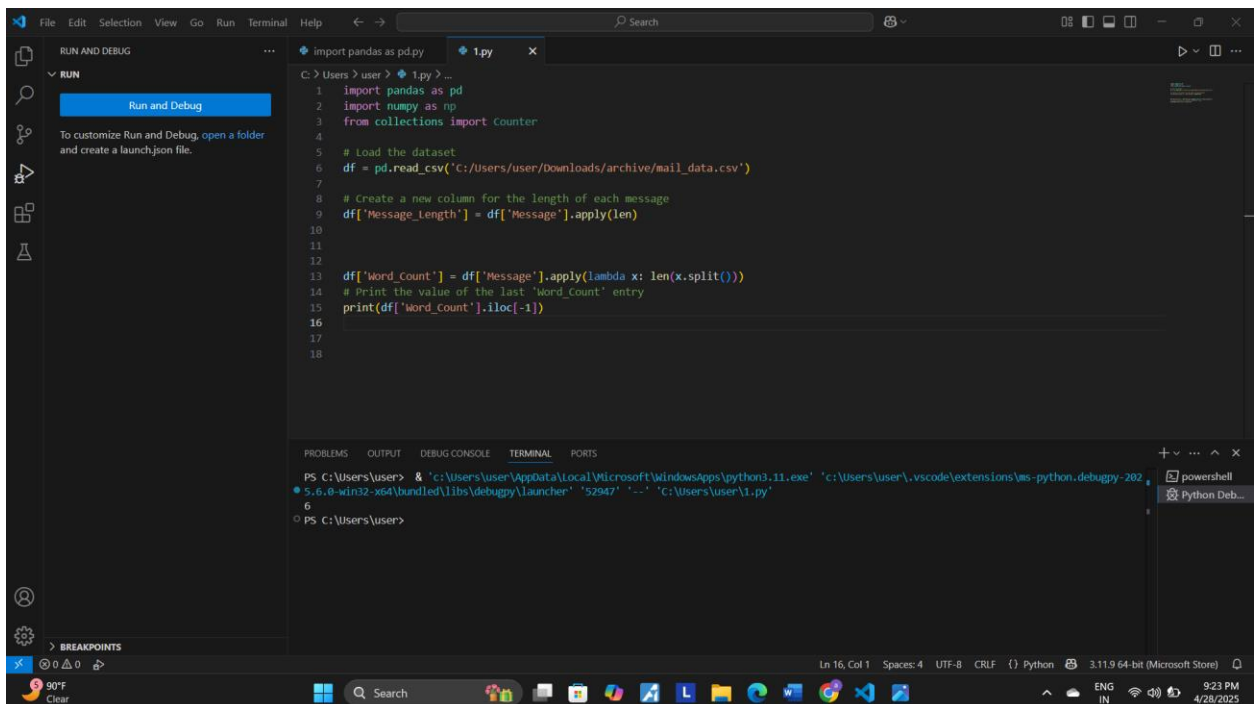
The screenshot shows a VS Code editor with a Python script named '1.py'. The script imports pandas and numpy, reads a CSV file 'mail_data.csv', and finds the longest spam message. The terminal output shows the execution of the script, displaying a long spam message.

```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_Length'] = df['Message'].apply(len)
10
11 # Filter spam messages
12 spam_df = df[df['Category'] == 'spam']
13
14 # Find the index of the longest spam message
15 longest_spam_idx = spam_df['Message_Length'].idxmax()
16
17 # Retrieve the longest spam message
18 longest_spam = spam_df.loc[longest_spam_idx, 'Message']
19
20 print(longest_spam)
```

Terminal Output:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\python\python.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\python\python.exe' 'c:\Users\user\1.py'
idx: this is Mandy Sullivan calling from ROMIX PH...you are chosen to receive £5000.00 in our Easter Prize draw....Please telephone 09041940223 to claim before 29/03/05 or your prize will be transferred to someone else....
PS C:\Users\user>
```

10. Create a column with word counts for each message.



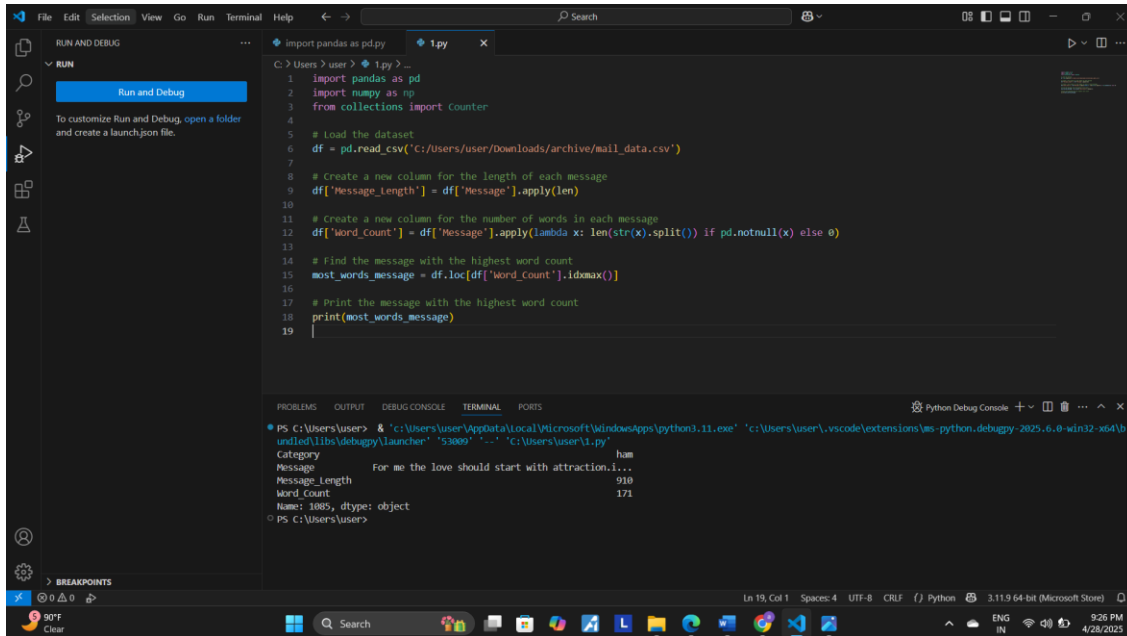
The screenshot shows a VS Code editor with a Python script named '1.py'. The script imports pandas and numpy, reads a CSV file 'mail_data.csv', and creates a new column 'Word_Count' for each message. The terminal output shows the execution of the script, displaying the word count for the last message.

```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_Length'] = df['Message'].apply(len)
10
11
12
13 df['Word_Count'] = df['Message'].apply(lambda x: len(x.split()))
14 # Print the value of the last 'Word_Count' entry
15 print(df['Word_Count'].iloc[-1])
16
17
18
```

Terminal Output:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\python\python.exe' 'c:\Users\user\1.py'
6
PS C:\Users\user>
```

11. Find the message with the most words.



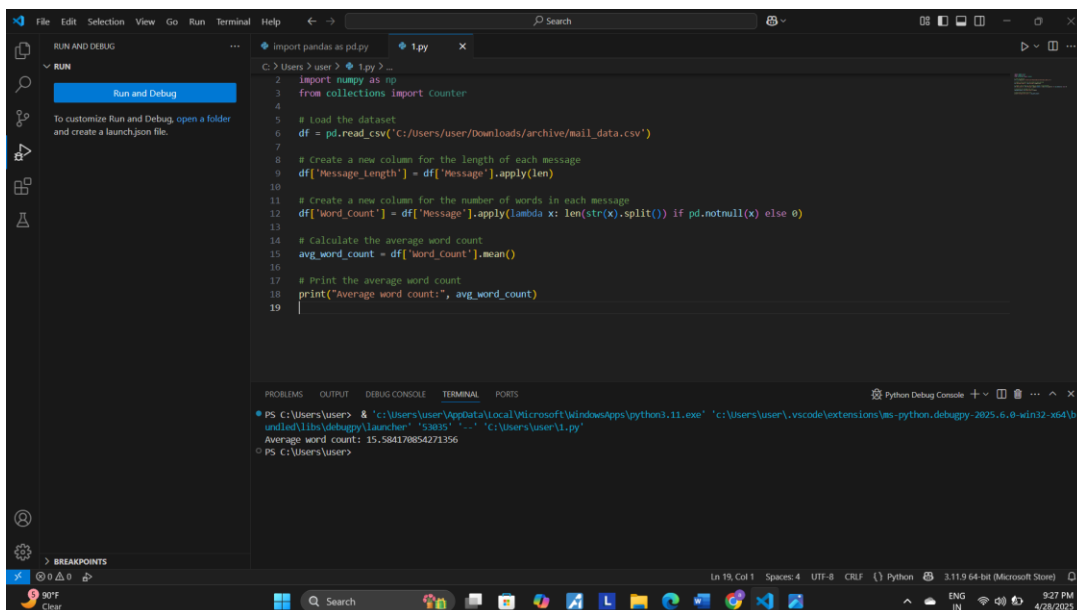
The screenshot shows a VS Code editor with a Python file named `1.py`. The script imports `pandas` and `numpy`, reads a CSV file from `C:\Users\user\Downloads\archive\mail_data.csv`, and creates two new columns: `Message_Length` (length of each message) and `Word_Count` (number of words in each message). It then finds the message with the highest word count and prints it.

```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_Length'] = df['Message'].apply(len)
10
11 # Create a new column for the number of words in each message
12 df['Word_Count'] = df['Message'].apply(lambda x: len(str(x).split()) if pd.notnull(x) else 0)
13
14 # Find the message with the highest word count
15 most_words_message = df.loc[df['Word_Count'].idxmax()]
16
17 # Print the message with the highest word count
18 print(most_words_message)
19
```

The terminal output shows the execution of the script, displaying the message with the highest word count:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\python\python.exe' 'c:\Users\user\Downloads\archive\mail_data.csv'
Category
Message
Message_Length
Word_Count
Name: 1005, dtype: object
PS C:\Users\user>
```

12. What is the average word count per message?



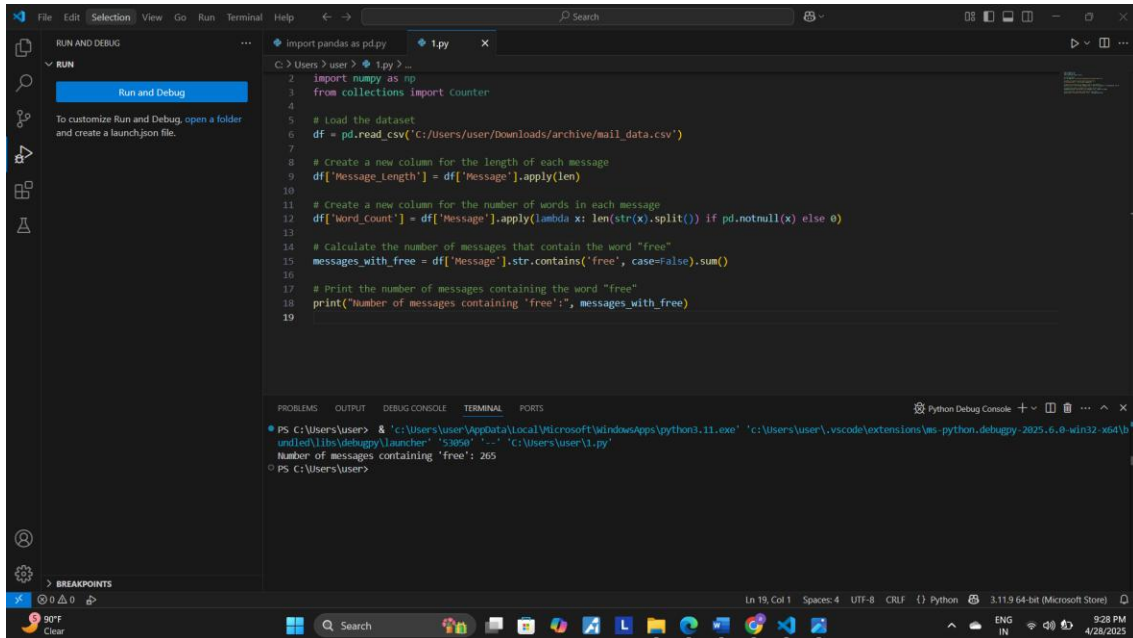
The screenshot shows a VS Code editor with a Python file named `1.py`. The script imports `pandas` and `numpy`, reads a CSV file from `C:\Users\user\Downloads\archive\mail_data.csv`, and creates two new columns: `Message_Length` (length of each message) and `Word_Count` (number of words in each message). It then calculates the average word count and prints it.

```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_Length'] = df['Message'].apply(len)
10
11 # Create a new column for the number of words in each message
12 df['Word_Count'] = df['Message'].apply(lambda x: len(str(x).split()) if pd.notnull(x) else 0)
13
14 # Calculate the average word count
15 avg_word_count = df['Word_Count'].mean()
16
17 # Print the average word count
18 print("Average word count:", avg_word_count)
19
```

The terminal output shows the execution of the script, displaying the average word count:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\python\python.exe' 'c:\Users\user\Downloads\archive\mail_data.csv'
Average word count: 15.584178854271356
PS C:\Users\user>
```


13. Find how many messages contain the word "free".

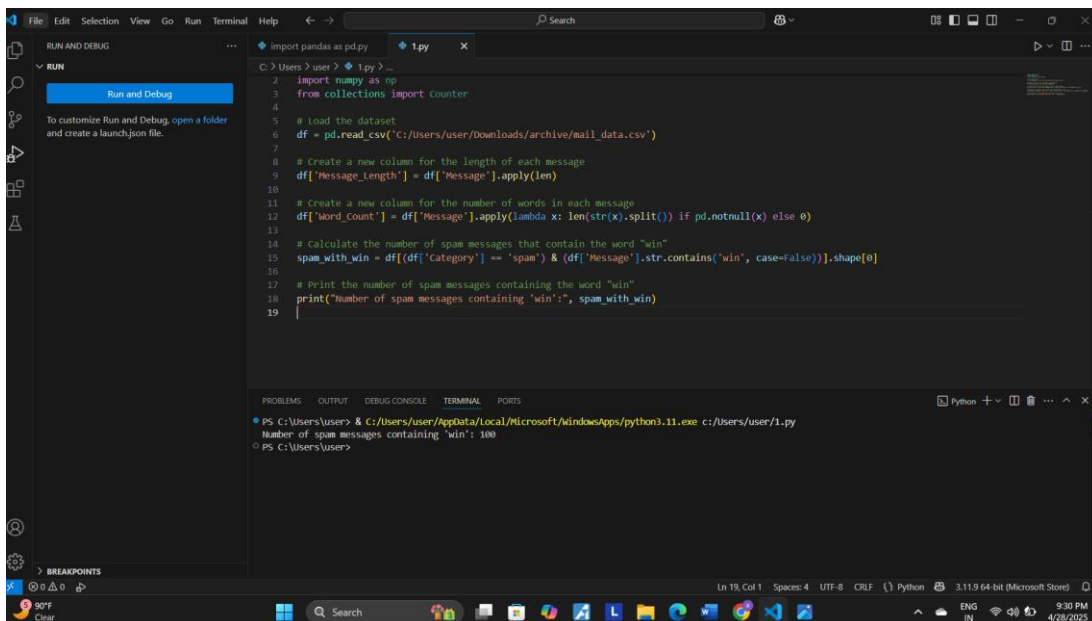


The screenshot shows a VS Code editor with a Python file named `1.py`. The script uses pandas to load a CSV file, calculate message lengths, and count the number of words in each message. It then filters for messages containing the word "free" and prints the count.

```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_length'] = df['Message'].apply(len)
10
11 # Create a new column for the number of words in each message
12 df['word_Count'] = df['Message'].apply(lambda x: len(str(x).split()) if pd.notnull(x) else 0)
13
14 # Calculate the number of messages that contain the word "free"
15 messages_with_free = df['Message'].str.contains('free', case=False).sum()
16
17 # Print the number of messages containing the word "free"
18 print("Number of messages containing 'free':", messages_with_free)
19
```

The terminal output shows the execution of the script, resulting in the message: "Number of messages containing 'free': 265".

14. Find how many spam messages contain the word "win".

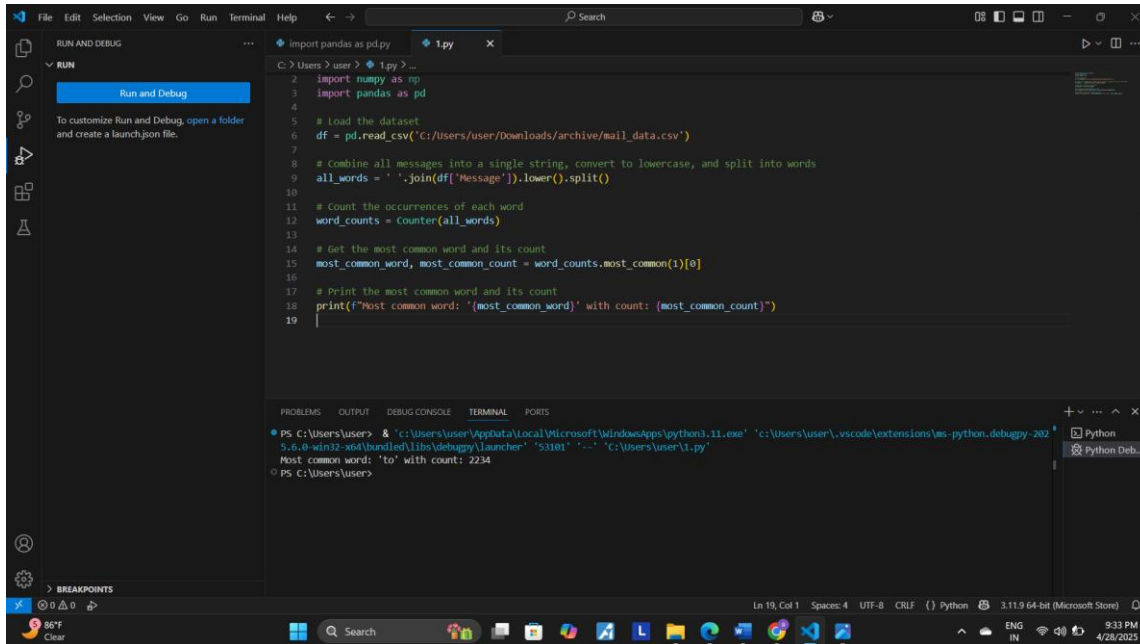


The screenshot shows a VS Code editor with a Python file named `1.py`. The script uses pandas to load a CSV file, calculate message lengths, and count the number of words in each message. It then filters for spam messages containing the word "win" and prints the count.

```
1 import pandas as pd
2 import numpy as np
3 from collections import Counter
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_length'] = df['Message'].apply(len)
10
11 # Create a new column for the number of words in each message
12 df['word_Count'] = df['Message'].apply(lambda x: len(str(x).split()) if pd.notnull(x) else 0)
13
14 # Calculate the number of spam messages that contain the word "win"
15 spam_with_win = df[(df['Category'] == 'spam') & (df['Message'].str.contains('win', case=False))].shape[0]
16
17 # Print the number of spam messages containing the word "win"
18 print("Number of spam messages containing 'win':", spam_with_win)
19
```

The terminal output shows the execution of the script, resulting in the message: "Number of spam messages containing 'win': 100".

15. What is the most common word in the dataset?



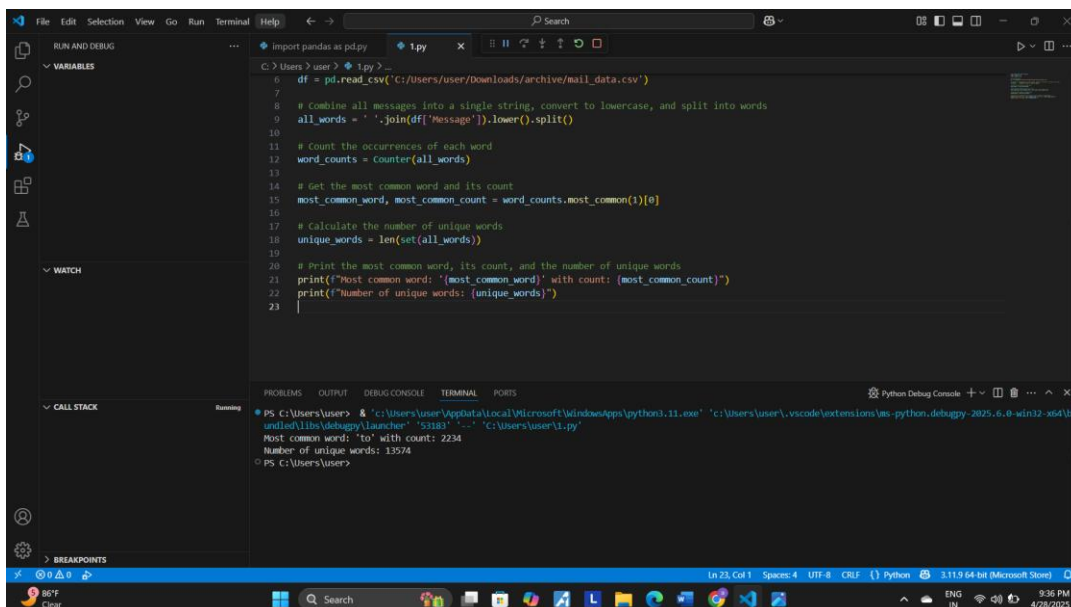
The screenshot shows a VS Code editor with a Python file named '1.py'. The script reads a CSV file, processes the messages, and finds the most common word. The terminal output shows the result: 'Most common word: 'to' with count: 2234'.

```
1 import pandas as pd
2 import numpy as np
3 import pandas as pd
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Combine all messages into a single string, convert to lowercase, and split into words
9 all_words = ' '.join(df['Message']).lower().split()
10
11 # Count the occurrences of each word
12 word_counts = Counter(all_words)
13
14 # Get the most common word and its count
15 most_common_word, most_common_count = word_counts.most_common(1)[0]
16
17 # Print the most common word and its count
18 print(f"Most common word: '{most_common_word}' with count: {most_common_count}")
19
```

Terminal Output:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\lib\debugpy\launcher' '53101' '-.' 'C:\Users\user\1.py'
Most common word: 'to' with count: 2234
PS C:\Users\user>
```

16. How many unique words are there across all messages?



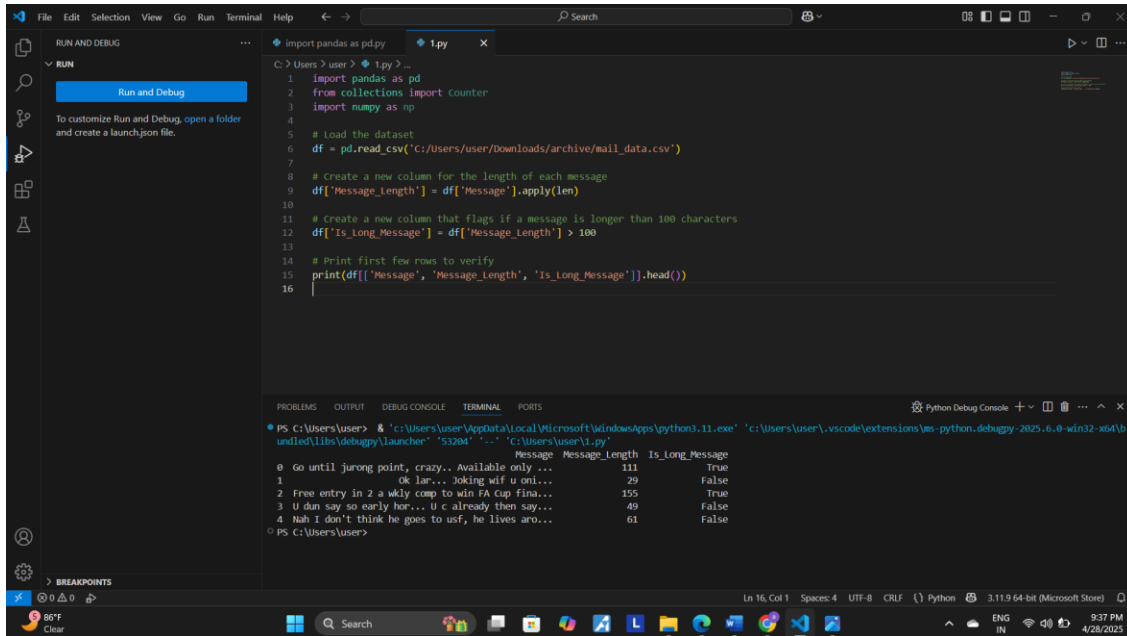
The screenshot shows a VS Code editor with a Python file named '1.py'. The script reads a CSV file, processes the messages, and finds the most common word and the number of unique words. The terminal output shows the results: 'Most common word: 'to' with count: 2234' and 'Number of unique words: 13574'.

```
1 import pandas as pd
2 import numpy as np
3 import pandas as pd
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Combine all messages into a single string, convert to lowercase, and split into words
9 all_words = ' '.join(df['Message']).lower().split()
10
11 # Count the occurrences of each word
12 word_counts = Counter(all_words)
13
14 # Get the most common word and its count
15 most_common_word, most_common_count = word_counts.most_common(1)[0]
16
17 # Calculate the number of unique words
18 unique_words = len(set(all_words))
19
20 # Print the most common word, its count, and the number of unique words
21 print(f"Most common word: '{most_common_word}' with count: {most_common_count}")
22 print(f"Number of unique words: {unique_words}")
23
```

Terminal Output:

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\lib\debugpy\launcher' '53183' '-.' 'C:\Users\user\1.py'
Most common word: 'to' with count: 2234
Number of unique words: 13574
PS C:\Users\user>
```

17. Create a boolean column: Is the message length > 100?

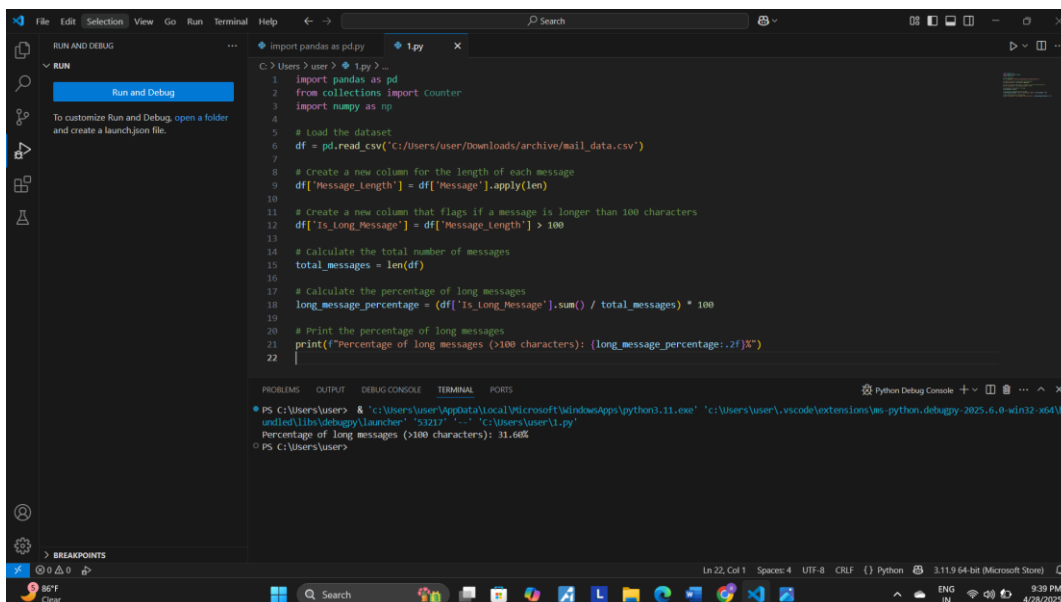


```
File Edit Selection View Go Run Terminal Help
import pandas as pd.py 1.py x
C:\Users\user> python 1.py ...
1 import pandas as pd
2 from collections import Counter
3 import numpy as np
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_Length'] = df['Message'].apply(len)
10
11 # Create a new column that flags if a message is longer than 100 characters
12 df['Is_Long_Message'] = df['Message_Length'] > 100
13
14 # Print first few rows to verify
15 print(df[['Message', 'Message_Length', 'Is_Long_Message']].head())
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy_launcher' "53204" ...
Message Message_Length Is_Long_Message
0 Go until jurong point, crazy.. Available only ... 111 True
1 Ok lar... Joking wif u oni... 29 False
2 Free entry in 2 a wily comp to win FA Cup fina... 155 True
3 U dun say so early hor... U c already then say... 49 False
4 Nah I don't think he goes to usf, he lives aro... 61 False
PS C:\Users\user>
```

18. Find the percentage of long messages (>100 characters).

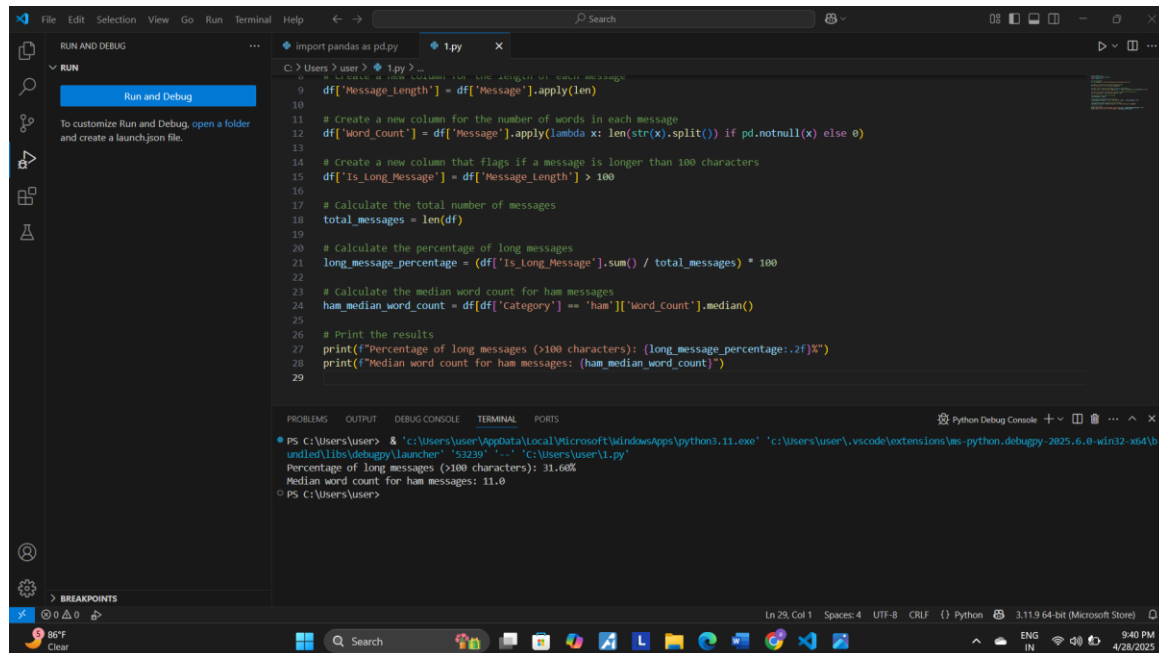


```
File Edit Selection View Go Run Terminal Help
import pandas as pd.py 1.py x
C:\Users\user> python 1.py ...
1 import pandas as pd
2 from collections import Counter
3 import numpy as np
4
5 # Load the dataset
6 df = pd.read_csv('C:/Users/user/Downloads/archive/mail_data.csv')
7
8 # Create a new column for the length of each message
9 df['Message_Length'] = df['Message'].apply(len)
10
11 # Create a new column that flags if a message is longer than 100 characters
12 df['Is_Long_Message'] = df['Message_Length'] > 100
13
14 # Calculate the total number of messages
15 total_messages = len(df)
16
17 # Calculate the percentage of long messages
18 long_message_percentage = (df['Is_Long_Message'].sum() / total_messages) * 100
19
20 # Print the percentage of long messages
21 print(f"Percentage of long messages (>100 characters): {long_message_percentage:.2f}%")
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\user> & 'c:\Users\user\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\user\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy_launcher' "53217" ...
Percentage of long messages (>100 characters): 31.60%
PS C:\Users\user>
```

19. What is the median word count of ham messages?



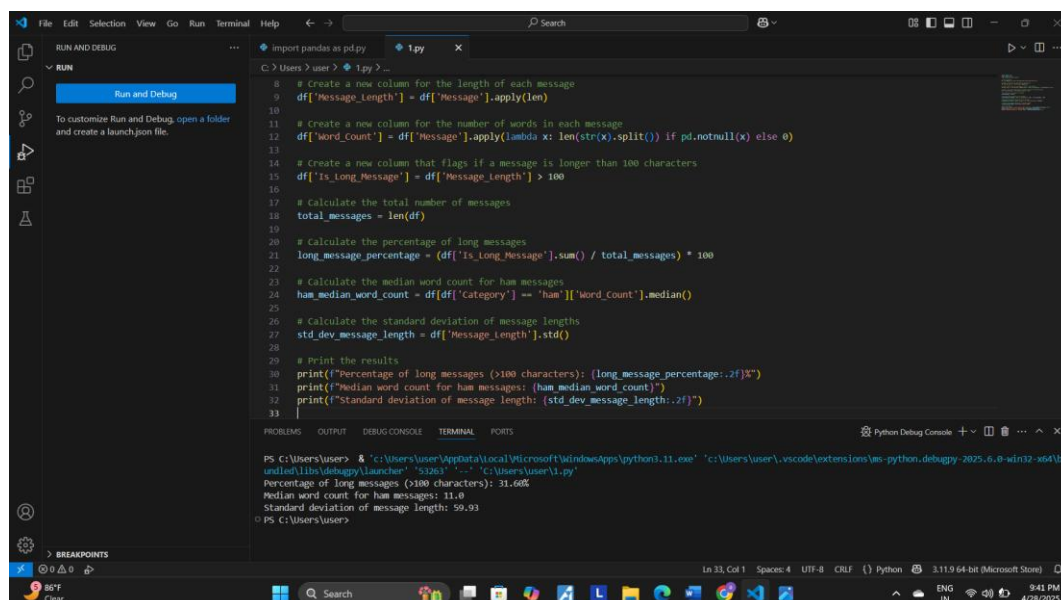
The screenshot shows a Python script in VS Code. The script imports pandas as pd, creates a DataFrame 'df' with columns 'Message_Length' and 'Word_Count', and calculates the median word count for ham messages. The output in the terminal shows the median word count for ham messages is 11.0.

```
import pandas as pd
df = pd.DataFrame({'Message_Length': [100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380, 390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490, 500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 610, 620, 630, 640, 650, 660, 670, 680, 690, 700, 710, 720, 730, 740, 750, 760, 770, 780, 790, 800, 810, 820, 830, 840, 850, 860, 870, 880, 890, 900, 910, 920, 930, 940, 950, 960, 970, 980, 990, 1000], 'Word_Count': [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]})
df['Message_Length'] = df['Message_Length'].apply(len)
df['Word_Count'] = df['Message'].apply(lambda x: len(str(x).split()) if pd.notnull(x) else 0)
df['Is_Long_Message'] = df['Message_Length'] > 100
total_messages = len(df)
long_message_percentage = (df['Is_Long_Message'].sum() / total_messages) * 100
ham_median_word_count = df[df['Category'] == 'ham']['Word_Count'].median()
print(f"Percentage of long messages (>100 characters): {long_message_percentage:.2f}%")
print(f"Median word count for ham messages: {ham_median_word_count}")
```

Python Debug Console output:

```
PS C:\Users\User> & 'c:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\User\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy_launcher' '53239' '-.' 'C:\Users\User\1.py'
Percentage of long messages (>100 characters): 31.60%
Median word count for ham messages: 11.0
PS C:\Users\User>
```

20. Find the standard deviation of message lengths.



The screenshot shows a Python script in VS Code. The script imports pandas as pd, creates a DataFrame 'df' with columns 'Message_Length' and 'Word_Count', and calculates the standard deviation of message lengths. The output in the terminal shows the standard deviation of message lengths is 59.93.

```
import pandas as pd
df = pd.DataFrame({'Message_Length': [100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370, 380, 390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490, 500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 610, 620, 630, 640, 650, 660, 670, 680, 690, 700, 710, 720, 730, 740, 750, 760, 770, 780, 790, 800, 810, 820, 830, 840, 850, 860, 870, 880, 890, 900, 910, 920, 930, 940, 950, 960, 970, 980, 990, 1000], 'Word_Count': [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]})
df['Message_Length'] = df['Message_Length'].apply(len)
df['Word_Count'] = df['Message'].apply(lambda x: len(str(x).split()) if pd.notnull(x) else 0)
df['Is_Long_Message'] = df['Message_Length'] > 100
total_messages = len(df)
long_message_percentage = (df['Is_Long_Message'].sum() / total_messages) * 100
ham_median_word_count = df[df['Category'] == 'ham']['Word_Count'].median()
std_dev_message_length = df['Message_Length'].std()
print(f"Percentage of long messages (>100 characters): {long_message_percentage:.2f}%")
print(f"Median word count for ham messages: {ham_median_word_count}")
print(f"Standard deviation of message length: {std_dev_message_length:.2f}")
```

Python Debug Console output:

```
PS C:\Users\User> & 'c:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\User\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bin\debugpy_launcher' '53263' '-.' 'C:\Users\User\1.py'
Percentage of long messages (>100 characters): 31.60%
Median word count for ham messages: 11.0
Standard deviation of message length: 59.93
PS C:\Users\User>
```