Name: Harsh Bhanarkar

Class: CS5

Roll No: CS5-36

PRN: 202401100075

Subject: EDS

Submission: Theory Activity 01

```
import numpy as np
import pandas as pd

df = pd.read_csv('/content/spam.csv')
```

```
df
```

	ID	Message	Label	Sender	Time Sent	Language
0	1	Congratulations! You've won $1000!	Spam	1234567890	4/28/2025 10:05	English
1	2	Hey, are we meeting today?	Ham	1987654321	4/28/2025 9:30	English
2	3	Free entry in 2 a wkly comp to win!	Spam	1122334455	4/28/2025 11:00	English
3	4	Call me when you get a chance.	Ham	1222333444	4/28/2025 12:45	English
4	5	Urgent! Claim your prize now.	Spam	1333444555	4/28/2025 8:50	English
5	6	Let's catch up later.	Ham	1444555666	4/28/2025 14:20	English

Next steps: Generate code with df View recommended plots New interactive sheet

```
#What is the total number of messages?

total_messages = df.shape[0]
total_messages
```

```
6
```

```
#How many spam and ham messages are there?

label_counts = df['Label'].value_counts()
label_counts
```

	count
Label	
Spam	3
Ham	3

```
dtype: int64
```

```
 #What is the percentage of spam messages?

spam_percentage = (label_counts.get('spam',0) / total_messages) * 100
spam_percentage
```

```
0.0
```

```
#What is the percentage of ham messages?

ham_percentage = (label_counts.get('ham',0) / total_messages) * 100
ham_percentage
```

```
0.0
```

```
 #Find the average number of characters in all messages.

avg_length = df['Message'].apply(len).mean()
avg_length
```

```
np.float64(29.166666666666668)
```

```
#Find the message with the maximum characters.

max_length_message = df.iloc[df['Message'].apply(len).idxmax()]
max_length_message
```

```
                                       2
      ID                               3
 Message   Free entry in 2 a wkly comp to win!
   Label                            Spam
  Sender                      1122334455
Time Sent                4/28/2025 11:00
Language                         English

dtype: object
```

```
#Find the message with the minimum characters.

min_length_message = df.iloc[df['Message'].apply(len).idxmin()]
min_length_message
```

```
                             5
      ID                     6
 Message   Let's catch up later.
   Label                   Ham
  Sender            1444555666
Time Sent      4/28/2025 14:20
Language               English

dtype: object
```

```
#What is the average length of spam messages only?

avg_spam_length = df[df['Label'] == 'spam']['Message'].apply(len).mean()
avg_spam_length
```

```
nan
```

```
#What is the average length of ham messages only?

avg_ham_length = df[df['Label'] == 'ham']['Message'].apply(len).mean()
avg_ham_length
```

```
nan
```

```
#Add a new column "Message_Length" that stores number of characters.

df['Message_Length'] = df['Message'].apply(len)
df.head()
```

```
   ID                              Message  Label      Sender        Time Sent  Language  Message_Length
0   1  Congratulations! You've won $1000!   Spam  1234567890  4/28/2025 10:05   English              34
1   2           Hey, are we meeting today?    Ham  1987654321   4/28/2025 9:30   English              26
2   3  Free entry in 2 a wkly comp to win!   Spam  1122334455  4/28/2025 11:00   English              35
3   4          Call me when you get a chance.   Ham  1222333444  4/28/2025 12:45   English              30
4   5            Urgent! Claim your prize now.  Spam  1333444555   4/28/2025 8:50   English              29
```

Next steps: Generate code with df View recommended plots New interactive sheet

```
#Find how many messages have more than 100 characters.

long_messages = df[df['Message_Length'] > 100].shape[0]
long_messages
```

```
0
```

```
 #Find the proportion of long messages (>100 characters).

long_message_proportion = (long_messages / total_messages) * 100
long_message_proportion
```

```
0.0
```

```
# Create a new column "Word_Count" that stores number of words in each message.

df['Word_Count'] = df['Message'].apply(lambda x: len(x.split()))
df.head()
```

	ID	Message	Label	Sender	Time Sent	Language	Message_Length	Word_Count
0	1	Congratulations! You've won $1000!	Spam	1234567890	4/28/2025 10:05	English	34	4
1	2	Hey, are we meeting today?	Ham	1987654321	4/28/2025 9:30	English	26	5
2	3	Free entry in 2 a wkly comp to win!	Spam	1122334455	4/28/2025 11:00	English	35	9
3	4	Call me when you get a chance.	Ham	1222333444	4/28/2025 12:45	English	30	7
4	5	Urgent! Claim your prize now.	Spam	1333444555	4/28/2025 8:50	English	29	5

Next steps: Generate code with df View recommended plots New interactive sheet

```
#Find the average number of words per message.

avg_words = df['Word_Count'].mean()
avg_words
```

```
np.float64(5.666666666666667)
```

```
 #Find the message with the highest word count.

max_word_count_message = df.iloc[df['Word_Count'].idxmax()]
max_word_count_message
```

	2
ID	3
Message	Free entry in 2 a wkly comp to win!
Label	Spam
Sender	1122334455
Time Sent	4/28/2025 11:00
Language	English
Message_Length	35
Word_Count	9

dtype: object

```
# Find how many spam messages have word count greater than 20.
```

```
spam_long_word_messages = df[(df['Label'] == 'spam') & (df['Word_Count'] > 20)].shape[0]
spam_long_word_messages
```

```
0
```

```
#Find the number of spam messages that contain "win" (case-insensitive).

spam_win_messages = df[(df['Label'] == 'spam') & (df['Message'].str.contains('win', case=False))].shape[0]
spam_win_messages
```

```
0
```

```
#Replace 'ham' and 'spam' labels with 0 and 1 respectively.

df['Label_Num'] = df['Label'].map({'ham': 0, 'spam': 1})
df.head()
```

	ID	Message	Label	Sender	Time Sent	Language	Message_Length	Word_Count	Label_Num
0	1	Congratulations! You've won $1000!	Spam	1234567890	4/28/2025 10:05	English	34	4	NaN
1	2	Hey, are we meeting today?	Ham	1987654321	4/28/2025 9:30	English	26	5	NaN
2	3	Free entry in 2 a wkly comp to win!	Spam	1122334455	4/28/2025 11:00	English	35	9	NaN
3	4	Call me when you get a chance.	Ham	1222333444	4/28/2025 12:45	English	30	7	NaN
4	5	Urgent! Claim your prize now.	Spam	1333444555	4/28/2025 8:50	English	29	5	NaN

Next steps: Generate code with df View recommended plots New interactive sheet

```
#Calculate the correlation between Message_Length and Label_Num.

correlation = df[['Message_Length', 'Label_Num']].corr()
correlation
```

	Message_Length	Label_Num
Message_Length	1.0	NaN
Label_Num	NaN	NaN

Next steps: Generate code with correlation View recommended plots New interactive sheet

Start coding or generate with AI.