Problem 3

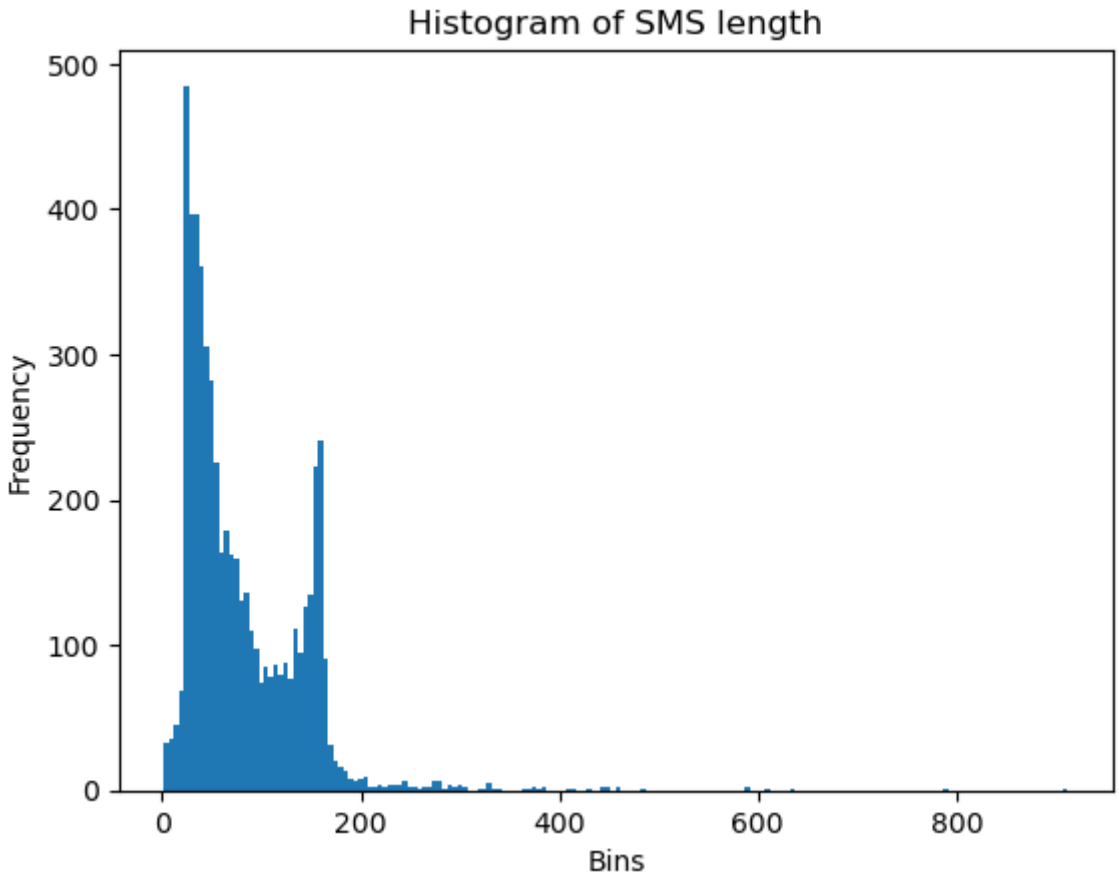
The dataset contains 5,574 messages tagged according to ham (legitimate) or spam. In this experiment, we will learn about text features, how to convert them into matrix form, and the Naive Bayes algorithm.

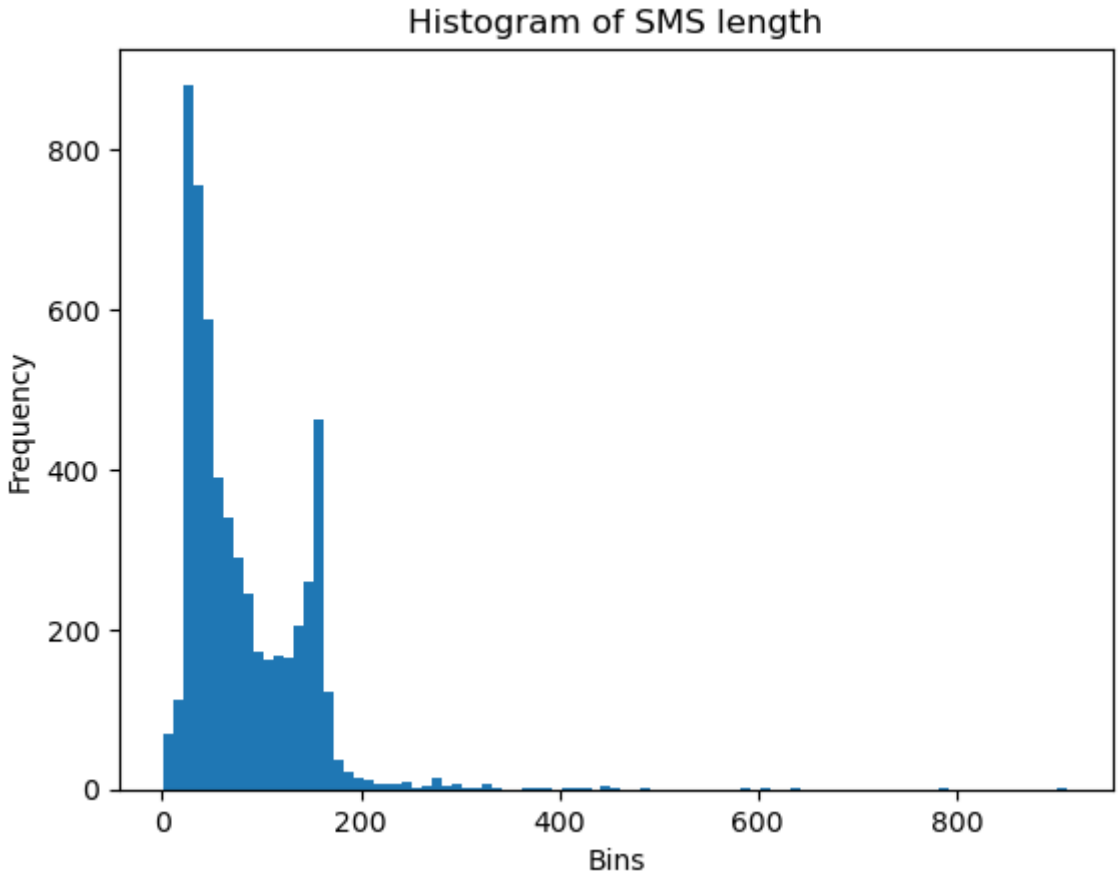
Answer the following questions:

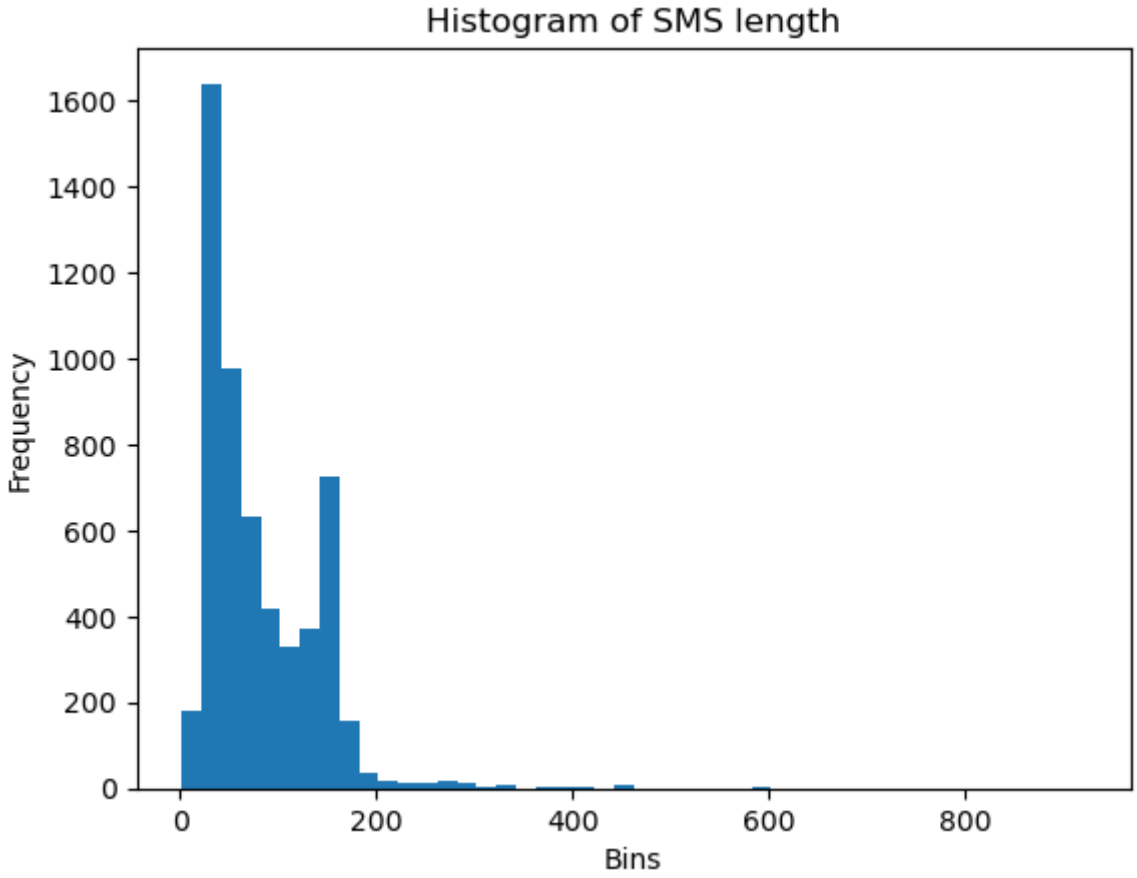
1. How many records are there? What's the distribution of the “label” class? Is it skewed?
   There are 5572 records in total. For the label class, there are 4825 ham and 747 spam. It is a little skewed.

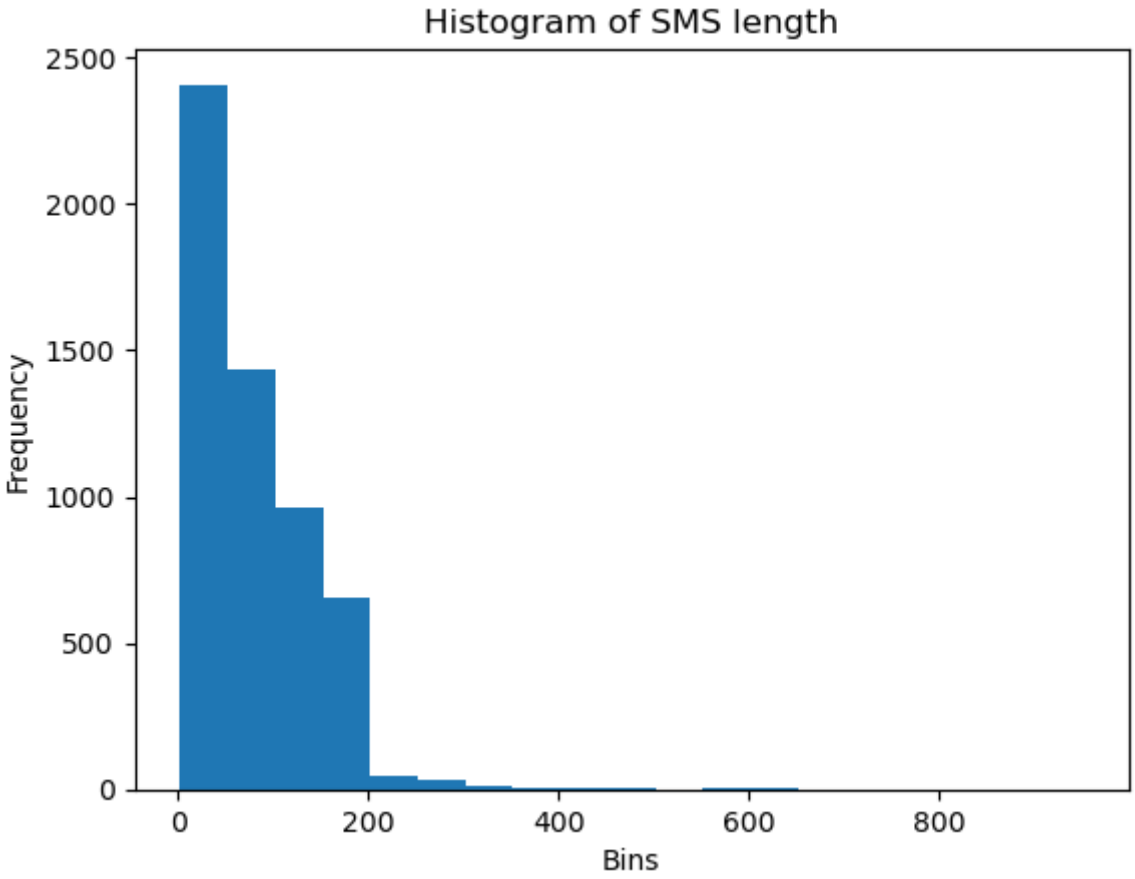
2. How many unique SMS messages are there in the dataset? What is the SMS message that occurs most frequently, and what is its frequency?
   There are 5169 unique SMS messages in the dataset.
   The SMS message that occurs most frequently is: Sorry, I'll call later.
   Its frequency is 30.

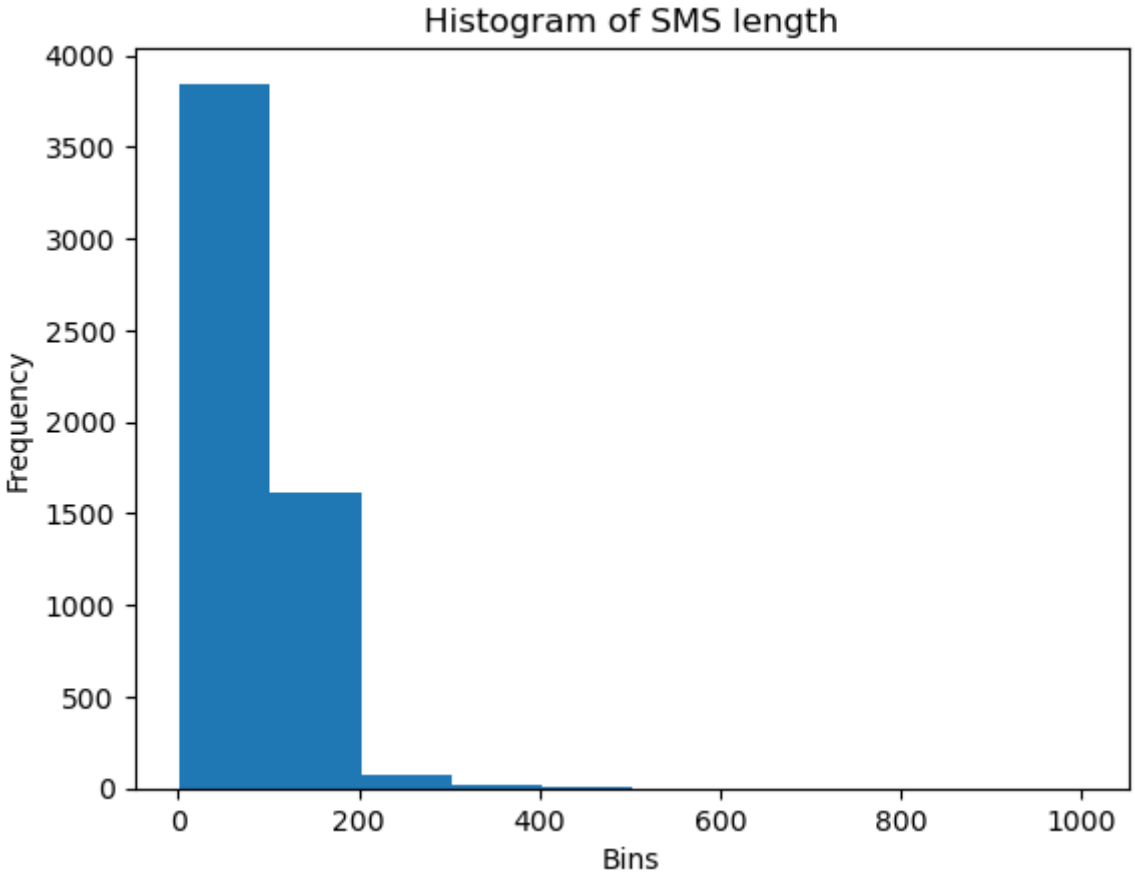
3. What is the maximum and minimum length of SMS messages present in the dataset? Plot the histogram of the length of SMS messages with bin sizes 5,10,20,50,100,200. What do you perceive after examining the plots?
   Maximum length of SMS is 910. Minimim length of SMS is 2.

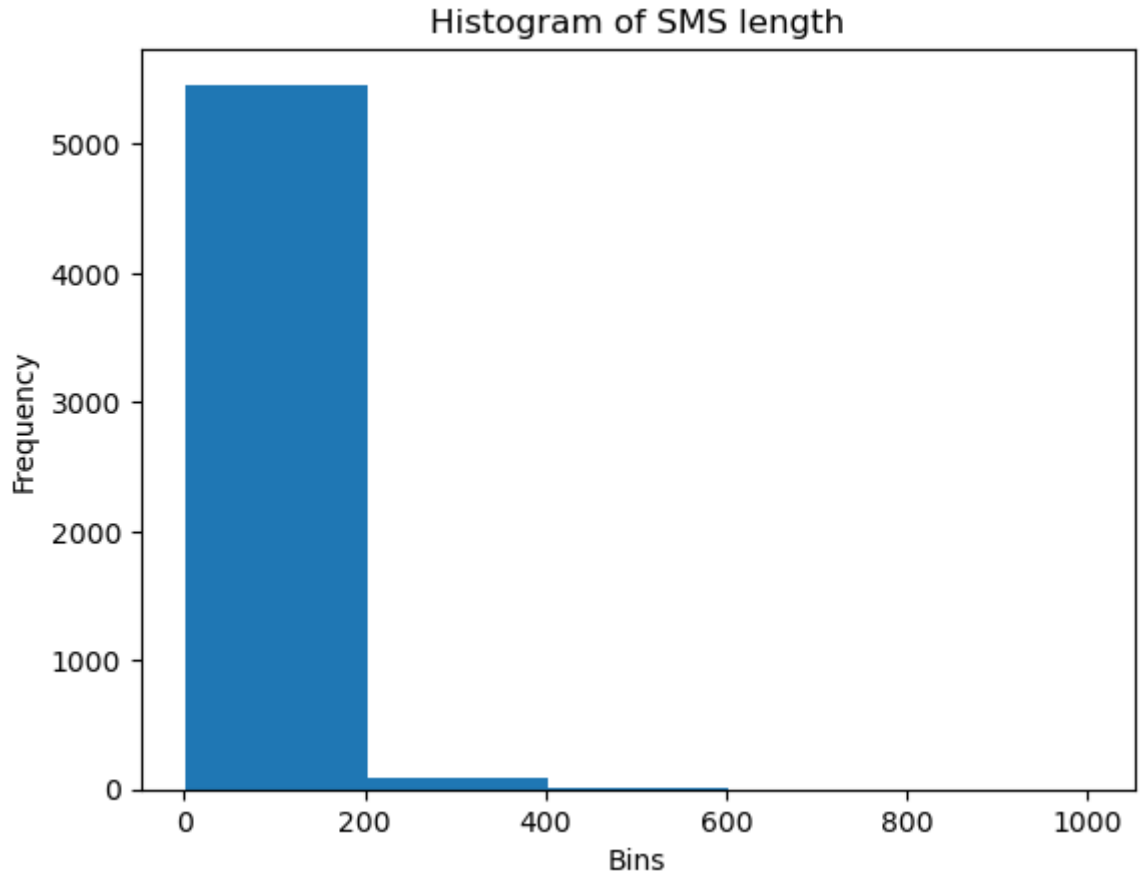



Histogram of SMS length
Frequency
0
200
400
600
800
Bins


Histogram of SMS length
Frequency
0
200
400
600
800
1000
1200
1400
1600
Bins

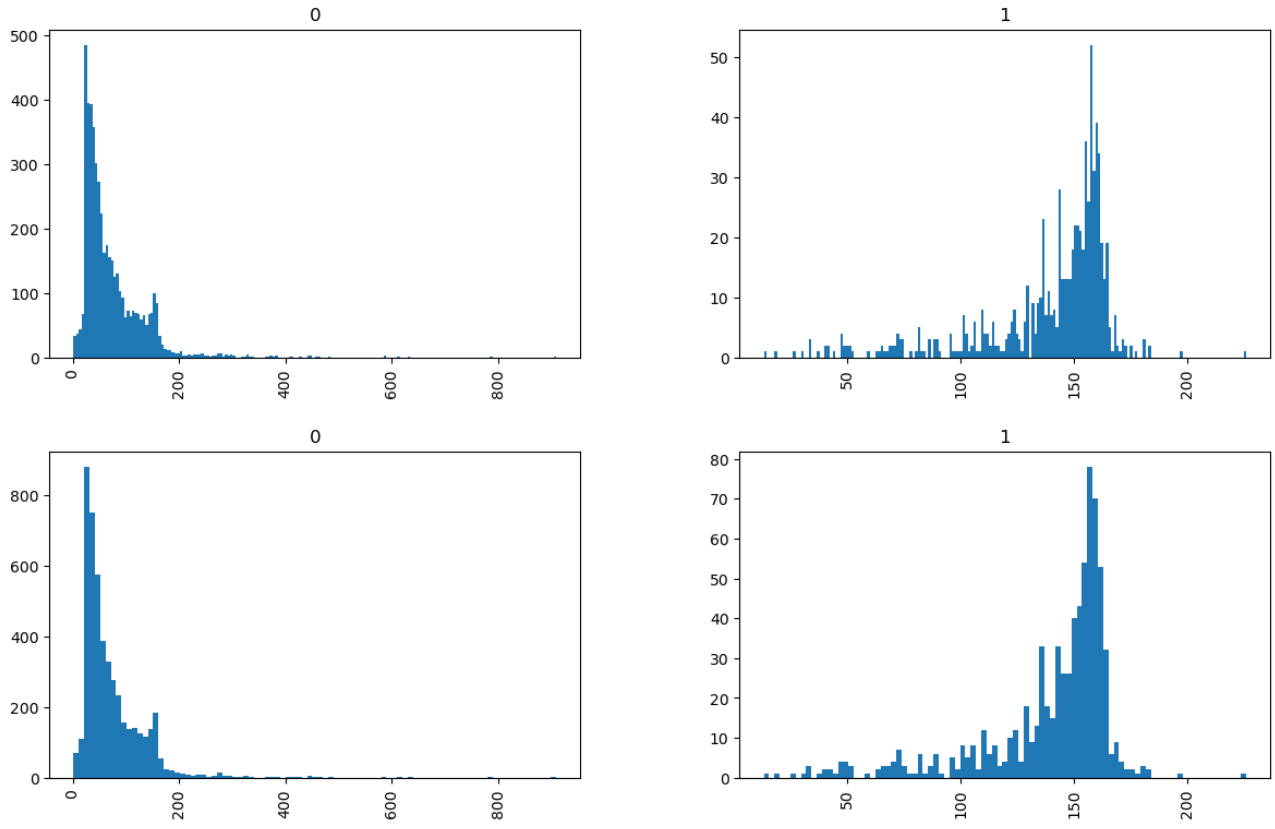

Histogram of SMS length
Frequency
Bins
0
500
1000
1500
2000
2500
0
200
400
600
800

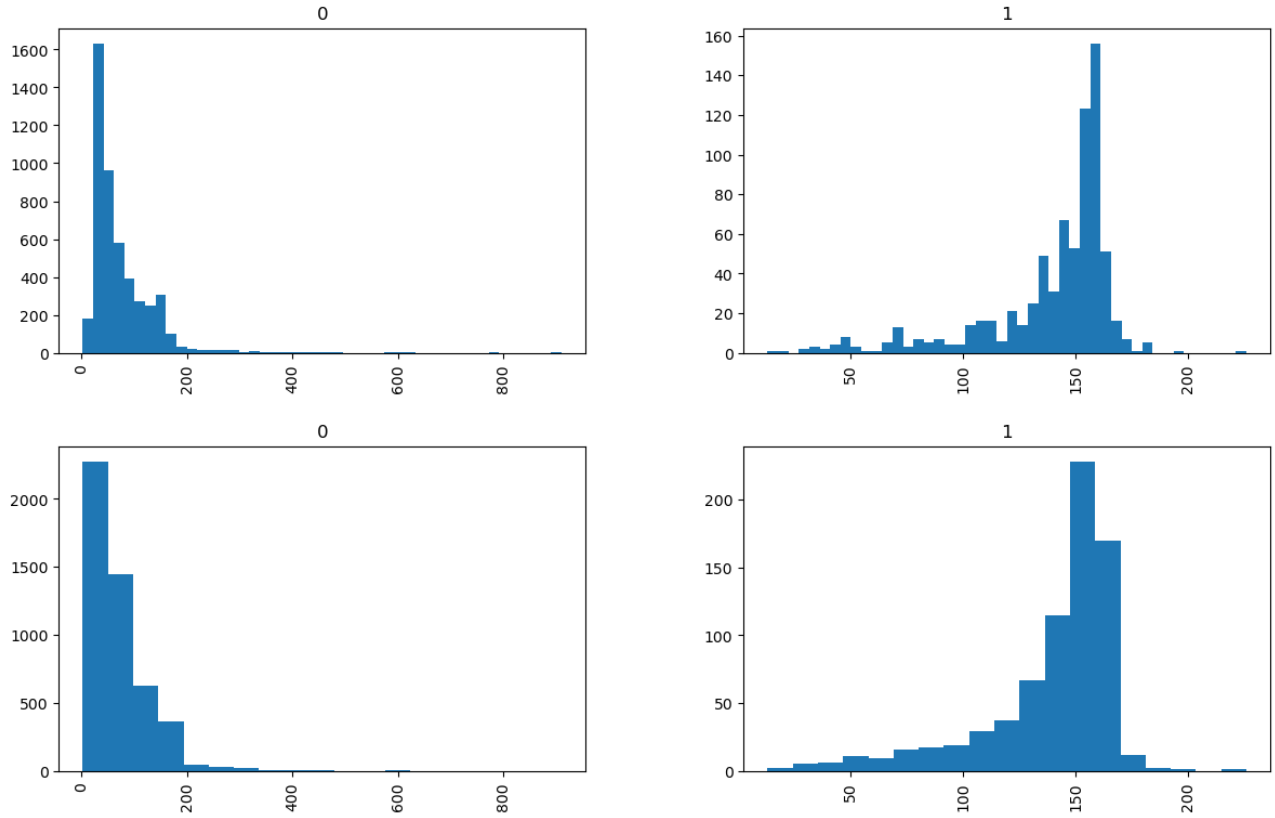

Histogram of SMS length
Frequency
Bins
0
500
1000
1500
2000
2500
3000
3500
4000
0
200
400
600
800
1000



It is a random distribution, which is a type of distribution that has several peaks. We can see one peark around 20-40, another peak around 100-140 and it lacks an apparent pattern. There can be a scenario where it combines various data properties. Hence, we should analyze the data separately.

4. Plot the histogram of the length of SMS messages for each label separately with bin sizes 5,10,20,50 i.e., histogram of the length of all ham SMS messages and histogram of the length of all spam SMS messages. What do you perceive after examining the plots?





- For histogram of the length of all **ham** SMS messages: we can note that the graph is biased towards the left side, and hence this is a sign of distribution, which is right-skewed distribution. A large number of data values occur on the left side (length between 20-50) and fewer data on the right side (length between 100-900).
- For histogram of the length of all **spam** SMS messages: we can note that the graph is biased towards the right side, and hence this is a sign of distribution, which is left-skewed distribution. A large number of data values occur on the right side (length between 150-170) and fewer data on the left side (length between 0-100).

5. In the Bag of words approach, we convert all strings into lower cases. Why did we do that, and why is it important? Can we convert all strings into the upper case and still fulfill our original goal?

   - Why did we do that, and why is it important?
     We convert all strings into lower cases so that a same word with different cases will be counted. For example, "win" and "Win" are counted as two occurances of "win", instead of "win": 1, "Win": 1.
   - Can we convert all strings into the upper case and still fulfill our original goal?
     Yes, we can convert all strings into the upper case.

6. What does CountVectorizer achieve? What will happen if we set stop words = "english"? Give five examples of stop-words in English.

   - What does CountVectorizer achieve?
     Convert a collection of text documents to a matrix of token counts.
   - What will happen if we set stop words = "english"?
     If ‘english’, a built-in stop word list for English is used.
   - Give five examples of stop-words in English.
     ['a', 'about', 'above', 'across', 'after']

7. Given a dataset, how do we generate a document-term matrix? Do we first generate a document-term matrix and then separate the matrix into train/test, or first separate the data into train/test and then generate a document-term matrix based on the training dataset and afterward generate a matrix for the test set? Explain your reasoning.
   Before generating document-term matrix, we should separate the SMS data into train and test.
   If you included the test data in the document-term matrix, you obtained a matrix that not just represent the training data.

8. Using the bag of words approach, convert documents = ['Hi, how are you?', 'Win money, win from home. Call now.', 'Hi., Call you now or tomorrow?'] to its document-term matrix.

	are	call	from	hi	home	how	money	now	or	tomorrow	win	you
0	1	0	0	1	0	1	0	0	0	0	0	1
1	0	1	1	0	1	0	1	1	0	0	2	0
2	0	1	0	1	0	0	0	1	1	1	0	1

9. How many features are created while making a document-term matrix for the SMS dataset? Can you think of a method to reduce the number of features? List the pros and cons of the method.
   7777 features.
   We could add more stop_words when we instatiate the CountVectorizer. Stop words are words like “and”, “the”, “him”, which are presumed to be uninformative in representing the content of a text, and which may be removed to avoid them being construed as signal for prediction. It also serves our purpose of reducing the number of features. Sometimes, however, similar words are useful for prediction, we might lose important information if we elimiate wrong words.
10. For our input dataset, which Naive Bayes model should we use, Gaussian Naive Bayes or Multinomial Naive Bayes? Explain your reasoning. Report accuracy, precision, recall, and F1 score for the spam class after applying the Naive Bayes algorithm.
    Specifically, we use **multinomial Naive Bayes** implementation. This particular classifier is suitable for classification with discrete features (such as in our case, word counts for text classification). It takes in integer word counts as its input. On the other hand Gaussian Naive Bayes is better suited for continuous data as it assumes that the input data has a Gaussian(normal) distribution.
    Accuracy score: 0.9847533632286996
    Precision score: 0.9420289855072463
    Recall score: 0.935251798561151
    F1 score: 0.9386281588447652