Sitong Qian

STA141B – HW2

Report: Regular Expression -- Email

"I completed this assignment myself without consulting books or tutorials that specifically addressed this data set."

Sitong Qian

I. Task Description

The core for this assignment is to find a way to distinguish the difference between SPAM and HAM emails. For the raw dataset given, I have 6541 emails listed from five different files, easy_ham, easy_ham2, hard_ham, spam, spam_2. Since, I already knew if a particular email is spam or ham by tracking which directory it comes from. I tried to solve the core problem by testing different variables given in the prompt to see which are some variables that have dramatic differences in two categories.

First, I tried to get all email name, by using 'lapply' and 'list.files' function in each directory, and unlist them to have a flat list. The difficulty I met at this point is I need to give name of directory. Thus, I used 'full.names = T'

Then, after I got all the email name, I looked at the first line of each email to see if it is an actual email, and I tested by trying the most common starting expression, and find all the cases that not apply, and tried the second most common starting expression, and find the rest of the cases that not apply. By following these patterns, I found total seven possible starting expression and detected one file that is not a email at all. Then, I deleted that email from the email name list and got a new list which I named 'email_file_name_fix'.

The next step, I worked on dividing each email in the form of header, body and attachment. For the header part, I was informed by instructor that all header followed the pattern that it ends at the first blank space. So, I used regular expression to find the position of first empty space in the list and used that position number as a indicator and extracted everything from beginning of the email to that specific position number.

Then I worked on the attachment part. I found attachments are very messy in this assignment, they follow different patterns. I approached this by finding the most distinctive branch first, and filled out specific cases of each branch with subbranches, and filled out the next tier subbranches if necessary.

The first, or the most inclusive branch I found that applied to email is if content-type documented in the header. By doing this, I have the first layer of branches, with two direction, yes or no. Then I filled out the all subbranches for the yes by doing the following subbranches, namely if boundary is addressed in the content type. If it is, then extracting the unique string after boundary. Here, it goes to next difficulty point I met, I found even not all boundary appeared in the content type followed the same pattern. I investigated the particular email, and I found out, that is because some of them are from Apple email which had a special character in a different position than other regular one. So, I did a subbranches here too, to extract two different pattern and find position in each pattern. Here, I basically finished all the cases that boundary lines are detected in the content-type. The next step I fulfilled the path that mentioned detecting points are false, which return attachment as value NA. Also, in case, some of the attachments don't have an ending string, I added a path indicating that for this particular case, just return attachment as from the starting position to the end of email.

Then, If content-type is not documented in the header, it is possible that it just mentioned boundary in the email. Thus, I tested if there is ‘multipart boundary’ or something like this in the email, if it is, then find the position of this string appeared, and extract attachment. If there is no such multipart boundary pattern found, return attachment as NA.

For the body part, I just combined the situation happened in the attachment part, with header code. If there is attachment, then body is the whole email minus header minus attachment. If there is no attachment, then body is the whole email minus header.

After all this process, I combined functions I wrought above and made a huge list with 6540 sub lists, and for each 6540 sub lists, I had four further sub lists, named as email name, header, body, attachment. I wrote all following pattern for detecting variables based on this huge list.

When writing variables, I used a way, apply sapply function to all email, to have all the information printed to debug everything more efficient. Basically, for the logical value, I just return TRUE or FALSE based on the length of grep function. Since if the pattern is not detected by grep or agrep, it will have output as integer(0), and the length of integer(0) is 0. The difficulty I met at this point is for isYelling variable, since it asked for if string is composed by all capital letter, and I found it was really hard to write a perfect pattern since there are blanks involved in it too. I approached this by saying if there is a lower-case letter detectable in the string, then it is not yelling. Also, for hour sent, I found there are two different patterns for hour, one is 3, the other is 03, I wrote an if function here to process two different situations.

Then, I combined all the results I had for different variables to make a large data frame to work on the final analysis by applying ggplot2 and table function mainly.

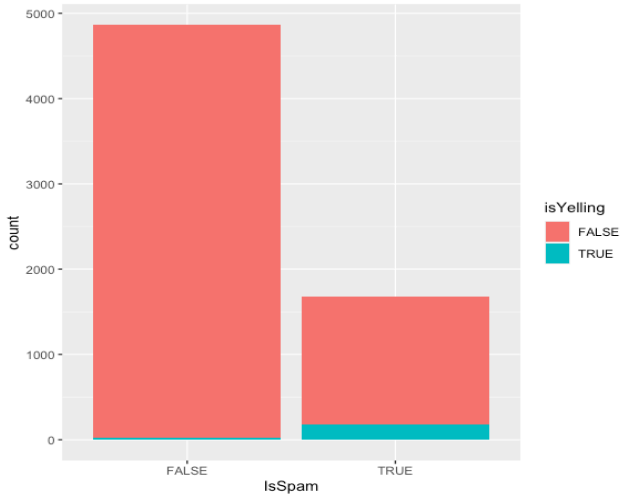
For the overall assignment, the must difficulty part was synthesized all the code I had into a complete function. Each time I did such combination process, I would get numerous error messages and I had to debug from the very part that the error message pop up and figured out how to revise my code in a way that avoid this particular error message and even beyond, find a pattern avoid future error.

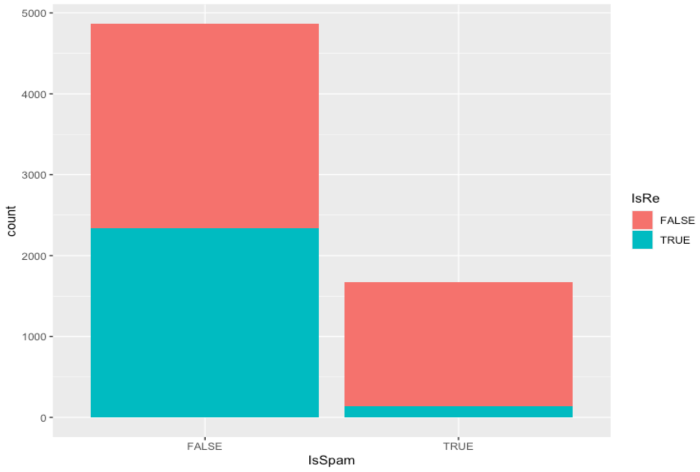
II. Data Analysis

After I had the data frame ready, I tested on several variable by using isSpam as the core variable to make comparison. For different variables, I made some guesses when evaluating the description of the variables on prompt, and it turned out some of the results did prove my thoughts. However, surprisingly, some of the variables I thought would work but it turned out in a opposite way.

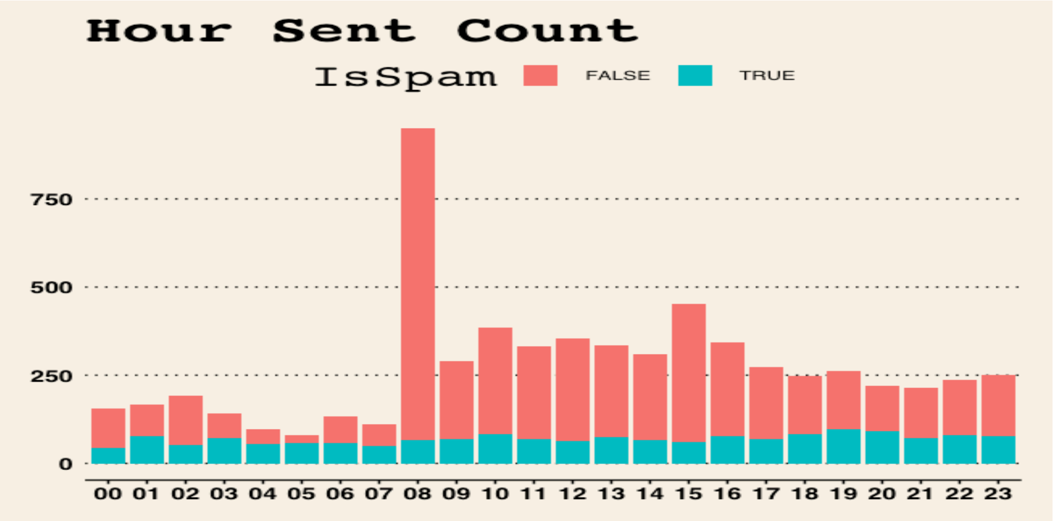
A. Powerful Indicators

Here, it can clearly be seen that for the spam email, it usually, don’t content Re in the subject line, So, Re can be a powerful indicator to tell if an email is spam or not. Initially, I assumed that most of the Spam are in a Yelling form in the header, but when I plot it, I found surprisingly, not that much of spam are in a Yelling form. However, there are much more spam email in a yelling form compared to ham email.

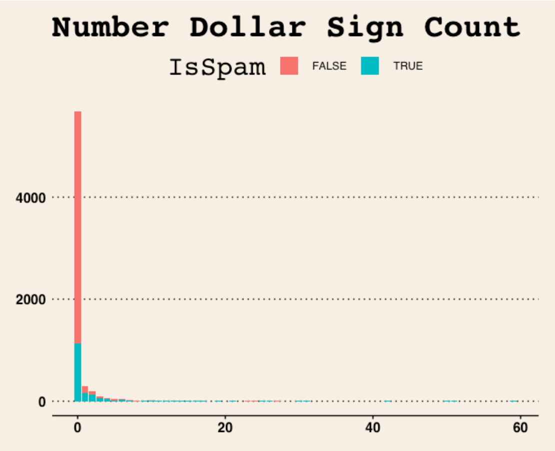


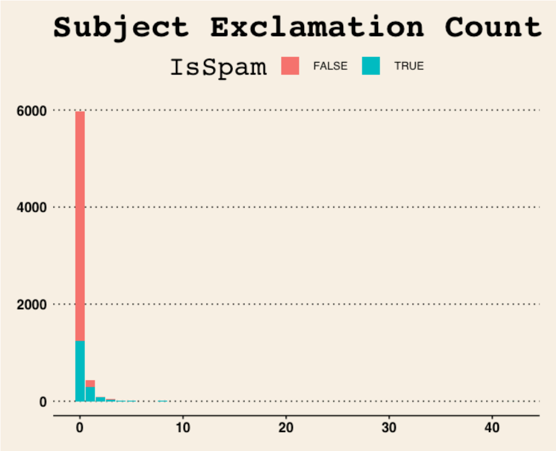


One of interesting features I found is when analyzing the time of hour sent. It can be seen that ham emails were generally sent in a working time, i.e, from 8am to 5pm, there is significant decline after working time. However, the time that spam emails sent are pretty standard, and somehow nearly uniformly distributed.

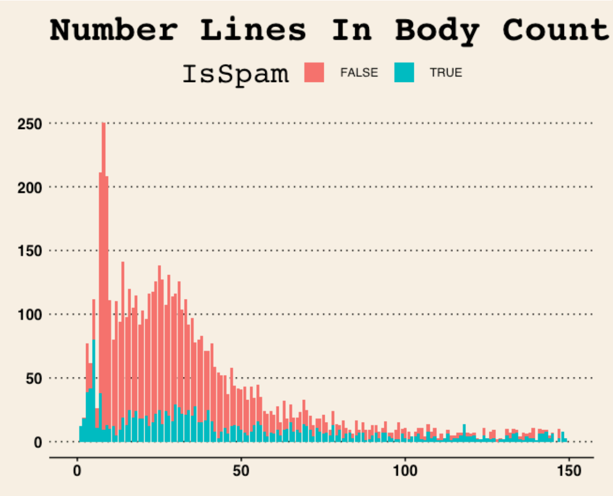


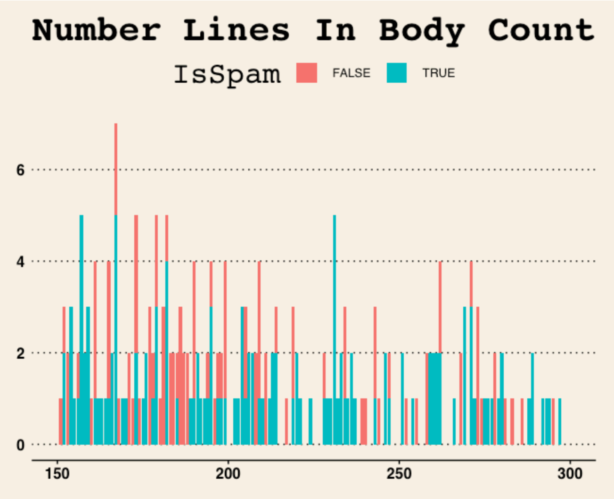
Another interesting features I found is related to number of dollar signs detected in the body. Initially, I assumed there should be more dollar signs in the spam emails, since spam emails are usually indicated about money and advertisement. It turns out the results actually prove my thoughts. By the following graph, it can be seen that spam emails tend to have more dollar signs than ham emails. Same distribution happened as for the subject exclamation count, spam emails tend to have more exclamation count. The Exclamation count is a very useful indicator to tell if the email is in a spam form or not. Since exclamation usually used in a very extreme writing environment which indicate a strong emotion.





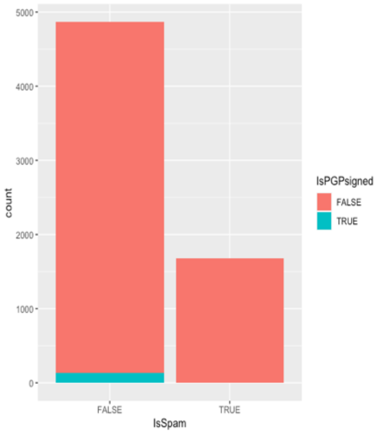
The count for number of lines in body is also very helpful, since the range for number of lines in body is too wide, I separated them into two major segments that have the most emails included.

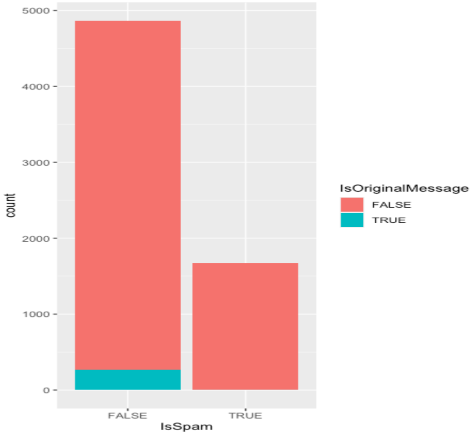


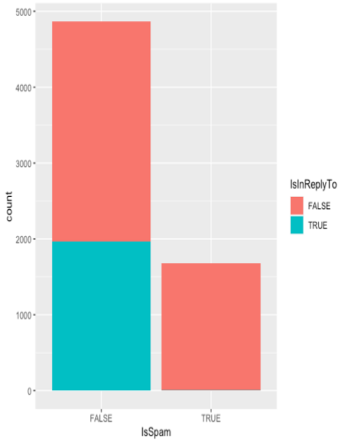


As it shown on the graph, the regular ham emails usually have less than 100 lines in their body part .The most abundant count for number of lines are between 0-50 lines in the body of emails. While, if the number of lines go beyond than 150 lines, it is a fair game for ham and spam emails, which means, it is not a useful indicator anymore.

There are three very powerful indicators which are IsPGPsigned, IsOriginalMessages, and IsInReplyto. From the following graph, it can be clearly indicated for all four indicators, they are highly likely or only addressed in the ham emails. Thus, it can work as reverse indicators for indicating if an email is not spam.

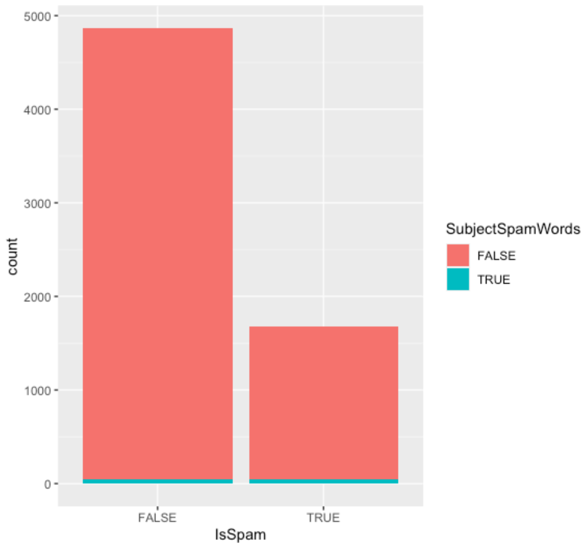


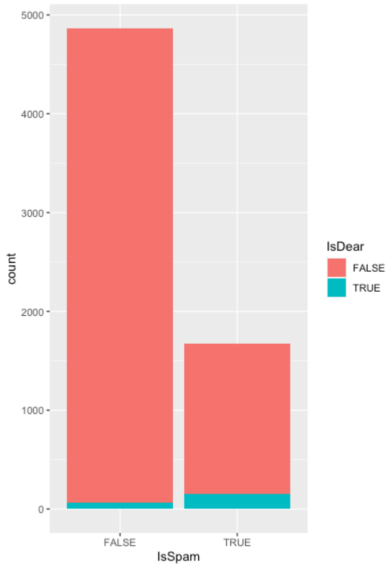




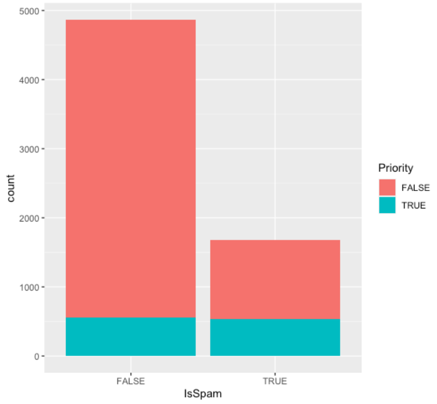
B. Ambiguous Indicators

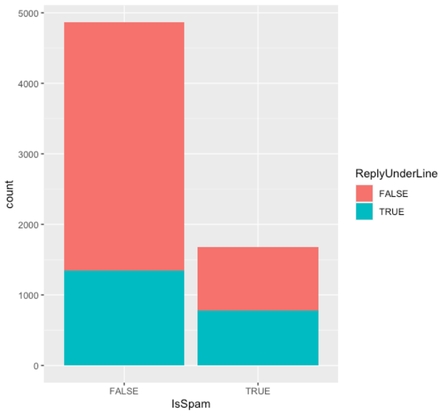
Also, there are some variables that have approximate same performance in both cases, or the differences are too subtle to be addressed or treated as a distinguish characteristic. At the beginning, I assumed spam word will be a powerful indicator, However, surprisingly, it is not. Based on the graph, there are actually roughly same amount spam word appeared in ham and spam emails. Another variable that I assume would work but turn out it didn't is IsDear Variable. From the graph, there is even more spam emails addressed dear than ham message.

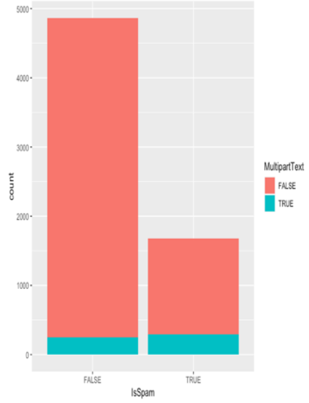




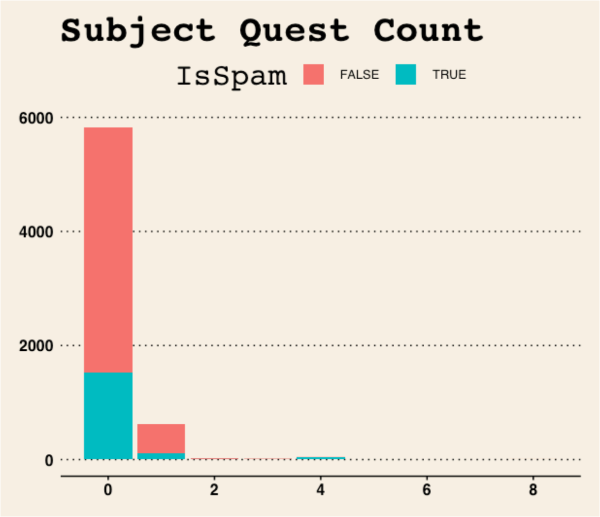
The distribution for X-priority showed that both spam and ham emails have roughly same amount of emails having X-priority or something similar. Such variables included if there is underline in Reply line and if the mail have Multipart text, which about half of spam emails have this characteristic, but it's hard to make an absolute conclusion based on this characteristic.







Unlike subject exclamation count and number dollar signs, it can be seen that quest count does not appear that often in the subject line for spam emails. Thus, it is not a powerful indicator.



III. Conclusion

It is hard to tell if an email is spam or ham from just one indicator. To make a better decision, it is recommended to evaluate several indicators to come up with a solid answer. In order to make a productive work, I should always start with some powerful indicators, then slowly move to indicators that have strong tendency to appear in either ham or spam emails, and finally move to some indicators that are more ambiguous and less determent.