EDA of Spam & Non-Spam Emails

Authored by Kshitiz Gupta.

Contents

Introduction

This is an Exploratory Data Analysis of the `email` dataset from the `openintro` package.

Description

These data represent incoming emails into David Diez's Gmail Account for the first three months of 2012. All personally identifiable information has been removed. The dataset has 3921 observations on the following 21 variables:

`spam` Indicator for whether the email was spam.

`to_multiple` Indicator for whether the email was addressed to more than one recipient.

`from` Whether the message was listed as from anyone (this is usually set by default for regular outgoing email).

`cc` Indicator for whether anyone was CCed.

`sent_email` Indicator for whether the sender had been sent an email in the last 30 days.

`time` Time at which email was sent.

`image` The number of images attached.

`attach` The number of attached files.

`dollar` The number of times a dollar sign or the word "dollar" appeared in the email.

`winner` Indicates whether "winner" appeared in the email.

`inherit` The number of times "inherit" (or an extension, such as "inheritance") appeared in the email.

`viagra` The number of times "viagra" appeared in the email.

`password` The number of times "password" appeared in the email.

`num_char` The number of characters in the email, in thousands.

`line_breaks` The number of line breaks in the email (does not count text wrapping).

`format` Indicates whether the email was written using HTML (e.g. may have included bolding or active links).

`re_subj` Whether the subject started with "Re:", "RE:", "re:", or "rE:"

`exclaim_subj` Whether there was an exclamation point in the subject.

`urgent_subj` Whether the word "urgent" was in the email subject.

`exclaim_mess` The number of exclamation points in the email message.

`number` Factor variable saying whether there was no number, a small number (under 1 million), or a big number.

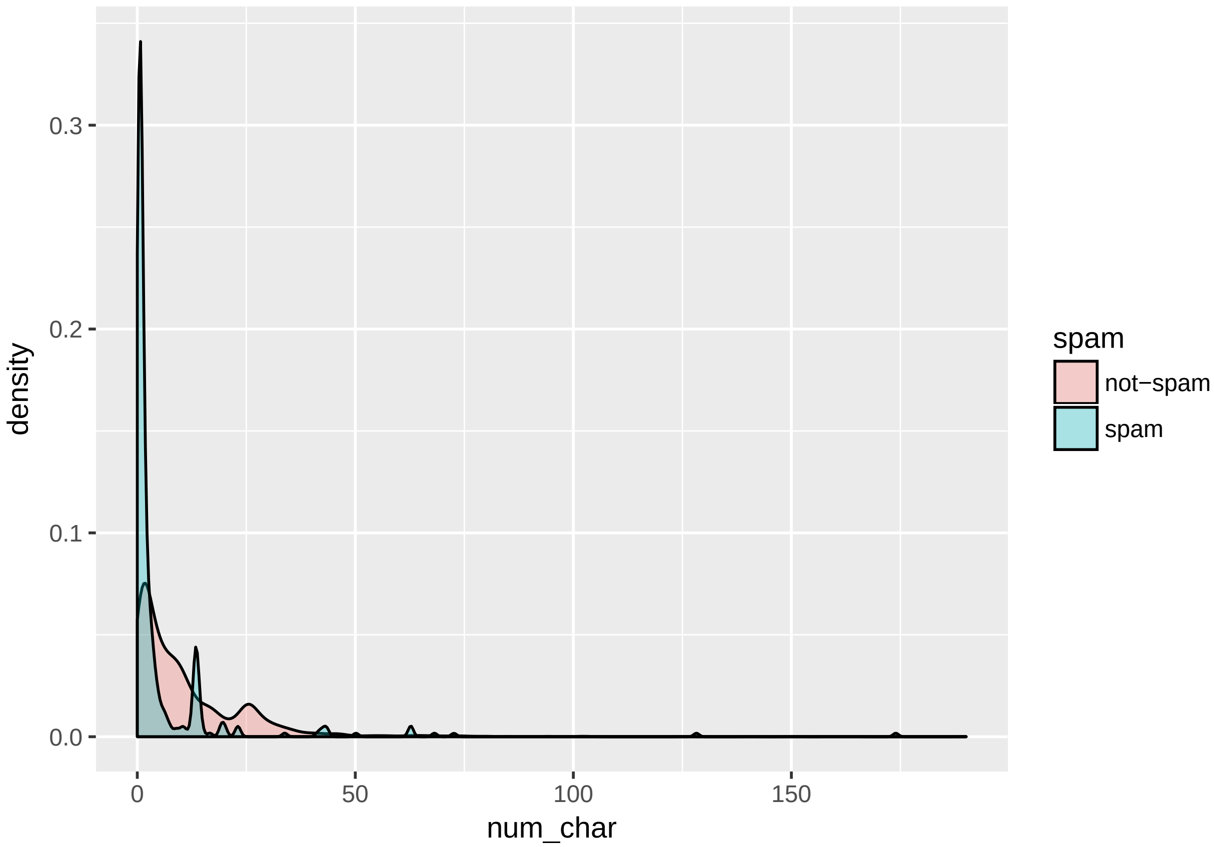
Loading the Necessary Packages

```
library(openintro); library(tidyverse); library(magrittr); library(corrplot); library(lubridate); libra
```

Spam and num of characters

Is there an association between spam and the length of an email? I would expect spam emails to be shorter and display less variability than real emails.

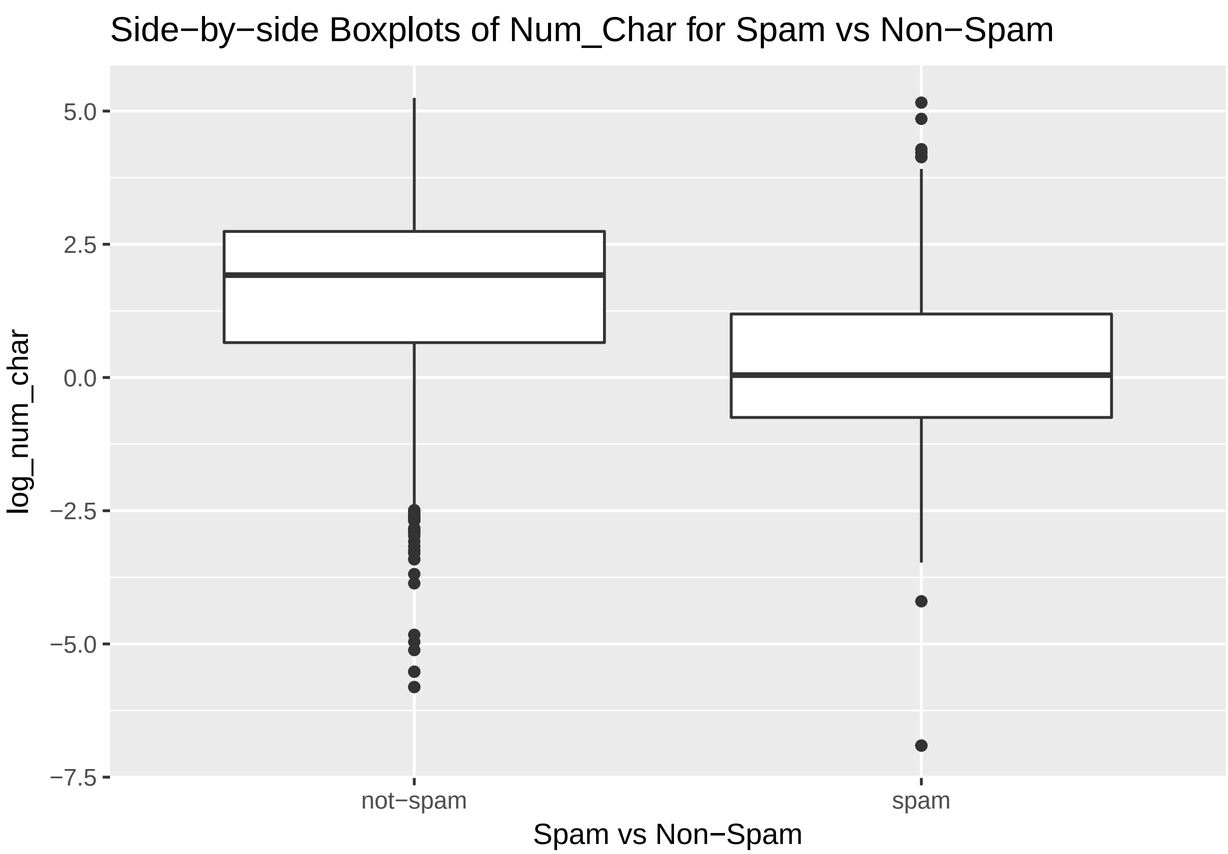
```
# Making the spam column more descriptive
email$spam[email$spam == 0] <- "not-spam"
email$spam[email$spam == 1] <- "spam"


# Making Density Plots
email %>%
  group_by(spam) %>% ggplot() + geom_density(aes(x = num_char, fill = spam), alpha = 0.3)
```




```
# The density plots show that the distributions are very right skewed. A log transformation may be appr

# Compute summary statistics

email %>%
  group_by(spam) %>%
  summarize(avg_len = mean(num_char), median_len = median(num_char), iqr = IQR(num_char), sd = sd(num_c
```

```
## # A tibble: 2 x 5
##   spam     avg_len median_len   iqr    sd
##   <chr>      <dbl>      <dbl> <dbl> <dbl>
## 1 not-spam   11.3        6.83 13.6   14.5
## 2 spam        5.44       1.05  2.82  14.9
```

```
# Create boxplots of log transformed num_char for spam vs non-spam
email %>%
  mutate(log_num_char = log(num_char)) %>%
  ggplot() + geom_boxplot(aes(x = as.factor(spam), y = log_num_char)) + xlab("Spam vs Non-Spam") + ggti
```



Here, we see that spam emails are indeed typically shorter than real ones.

Spam and !!!

Let's look at a more obvious indicator of spam: exclamation marks. `exclaim_mess` in the `emails` dataset contains the number of exclamation marks in each message. Using summary statistics and visualization, we can find out if there is a relationship between this variable and whether or not a message is spam.

First thing to consider is the relative number of spam and non-spam emails in this dataset

```
table(email$spam)/nrow(email)
```

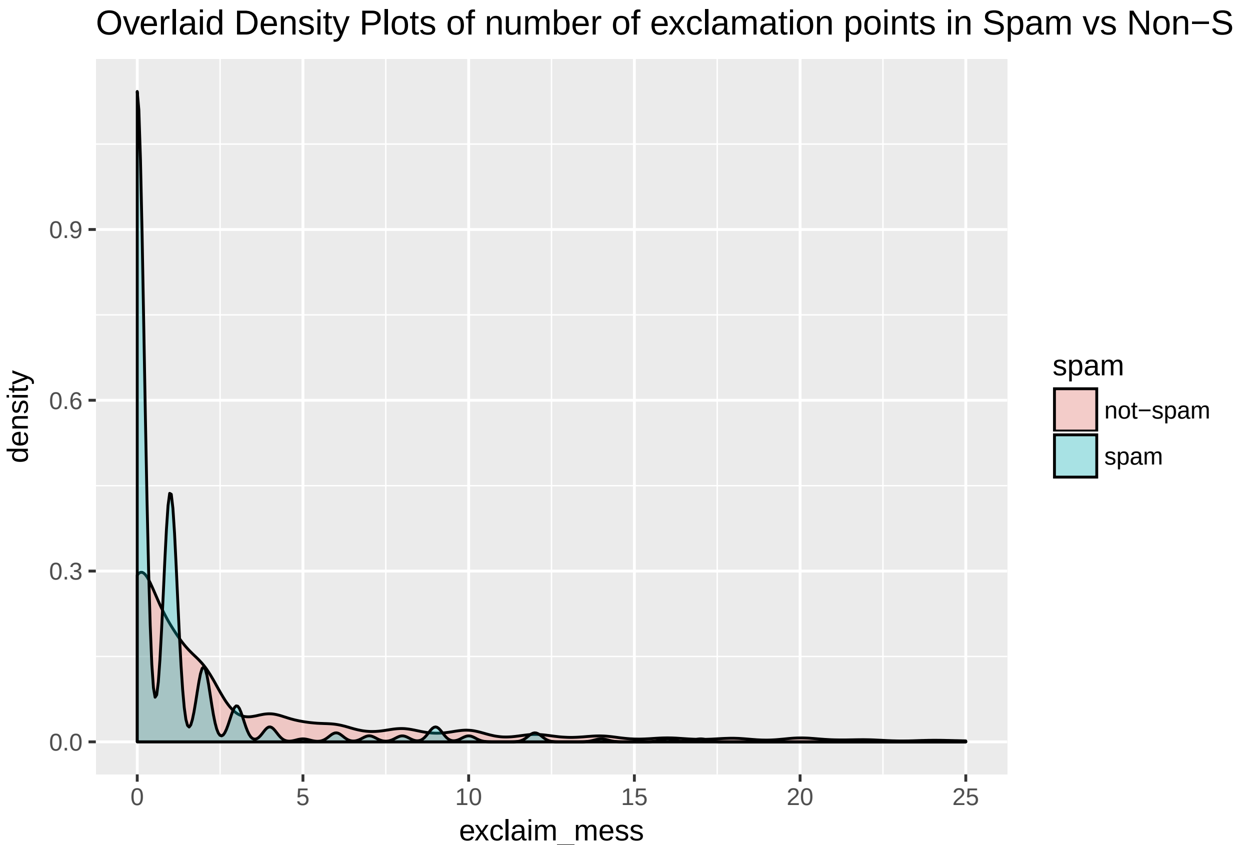
```
##
##   not-spam       spam
## 0.90640143 0.09359857
```

We see that about 90% of emails are not-spam and remaining 10% are spam. Keeping the relative number of these emails in mind we calculate the total number of exclamation points in spam and non-spam emails

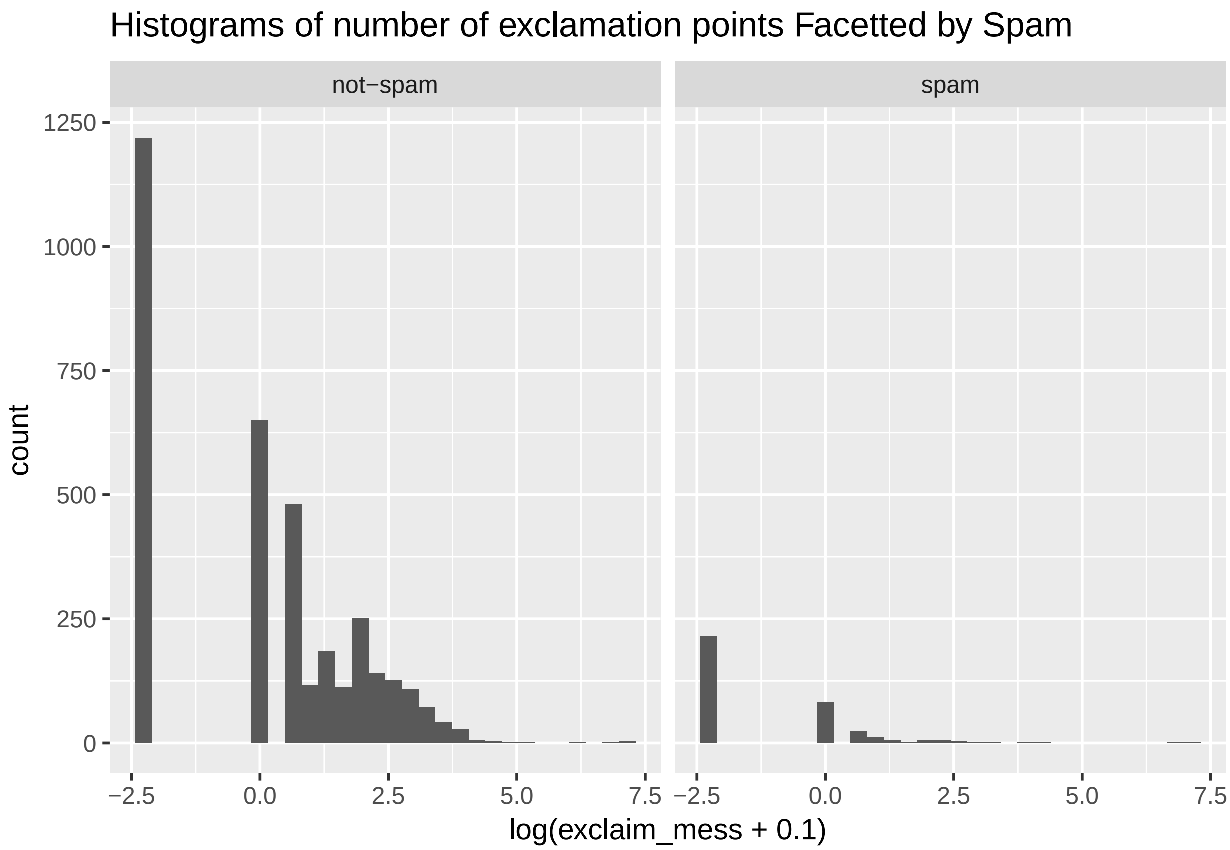
```
email %>% group_by(spam) %>% summarise(total_exclaim_points = sum(exclaim_mess))
```

```
## # A tibble: 2 x 2
##   spam     total_exclaim_points
##   <chr>                   <dbl>
## 1 not-spam                23130
## 2 spam                     2687
```

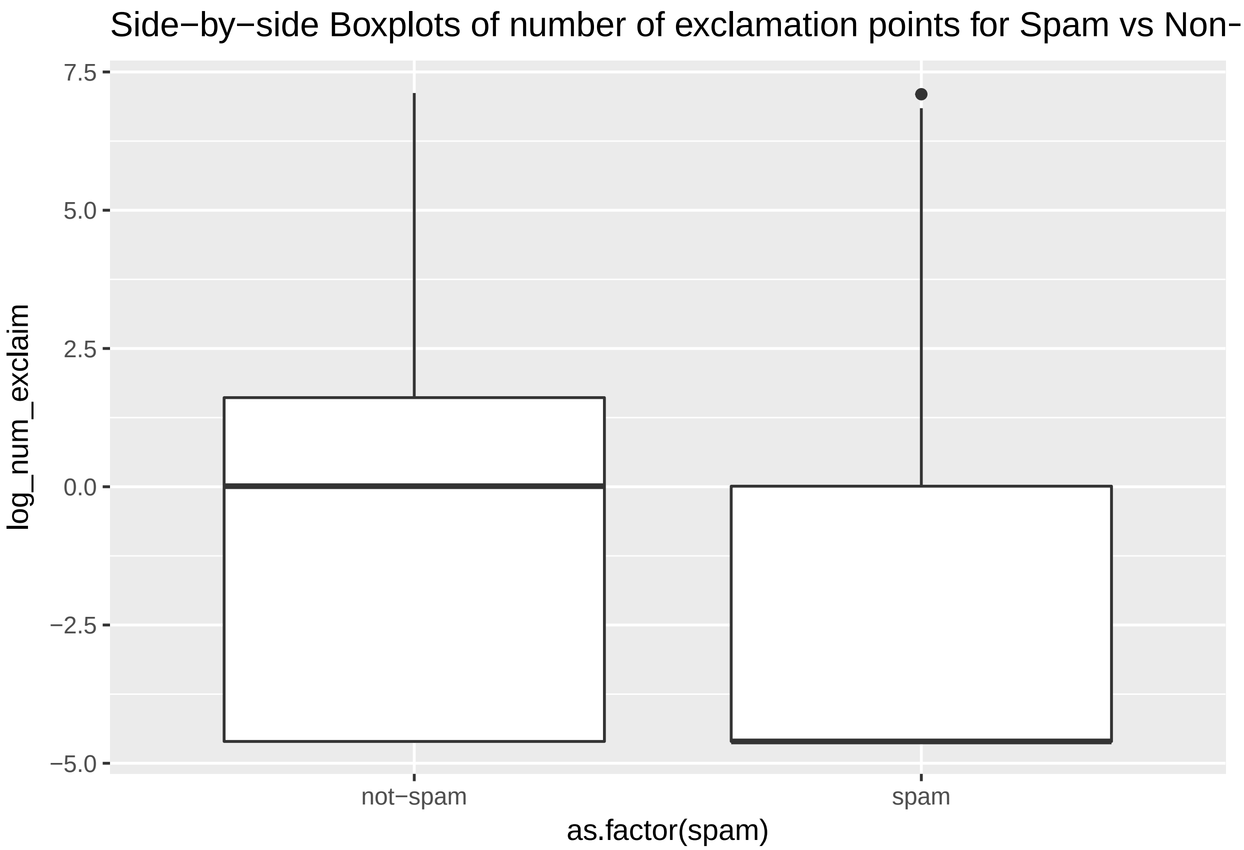
```r
# Generating Overlaid Density Plots
email %>% group_by(spam) %>% ggplot() + geom_density(aes(x = exclaim_mess, fill = spam), alpha = 0.3) +
```




```r
# The distributions are very right skewed here. Median and IQR seem to better summary statistics in thi


# Generating Faceted log_histograms
email %>% group_by(spam) %>% ggplot() + geom_histogram(aes(x = log(exclaim_mess + 0.1))) + facet_wrap(~
```




```
# Generating log transformed BoxPlots
email %>% mutate(log_num_exclaim = log(exclaim_mess+0.01)) %>% group_by(spam) %>% ggplot() + geom_boxpl
```




```
# Compute summary statistics
email %>%
  group_by(spam) %>%
  summarize(median(exclaim_mess), IQR(exclaim_mess))
```

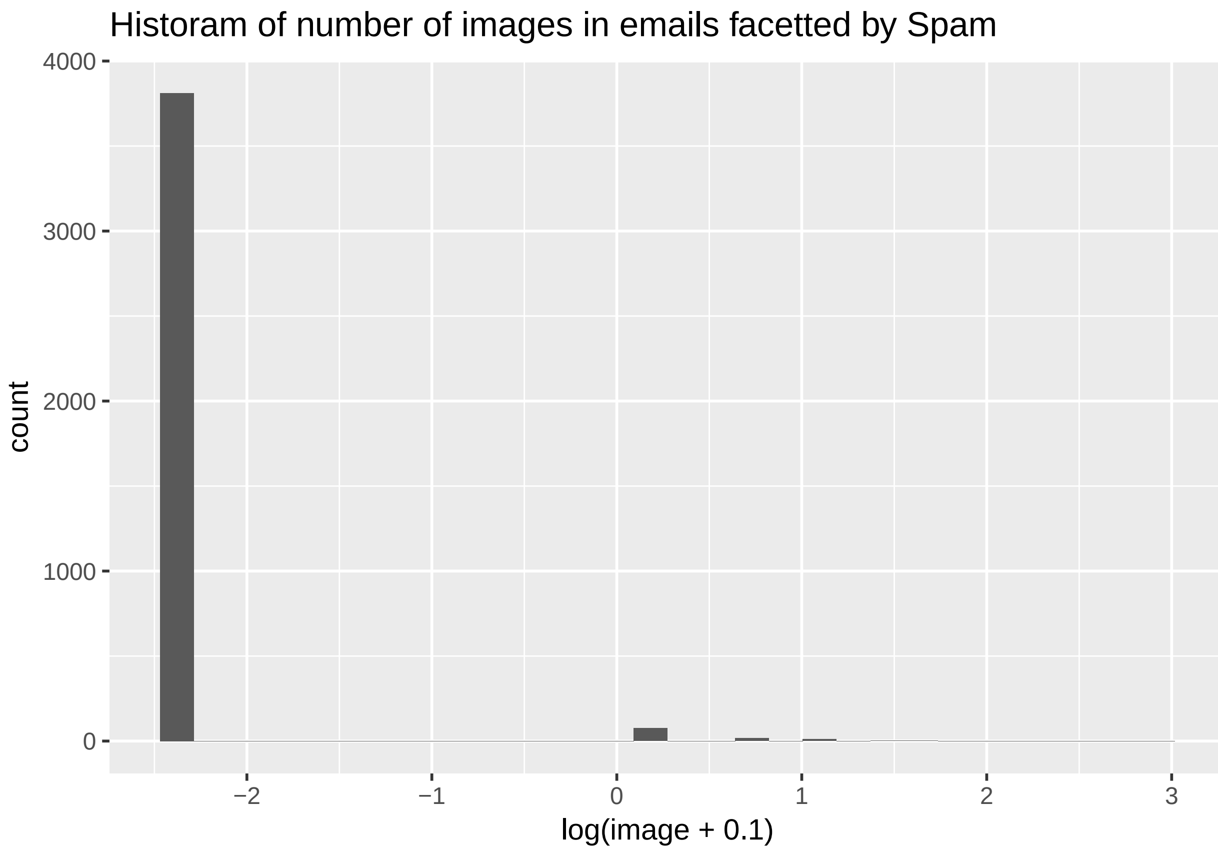
```
## # A tibble: 2 x 3
##   spam     `median(exclaim_mess)` `IQR(exclaim_mess)`
##   <chr>                     <dbl>               <dbl>
## 1 not-spam                      1                   5
## 2 spam                          0                   1
```

Here, we infer that the most common value of exclaim_mess in both classes of email is zero (a log(exclaim_mess) of -4.6 after adding .01) and the typical number of exclamations in the not-spam group appears to be slightly higher than in the spam group.

Spam and image attachments

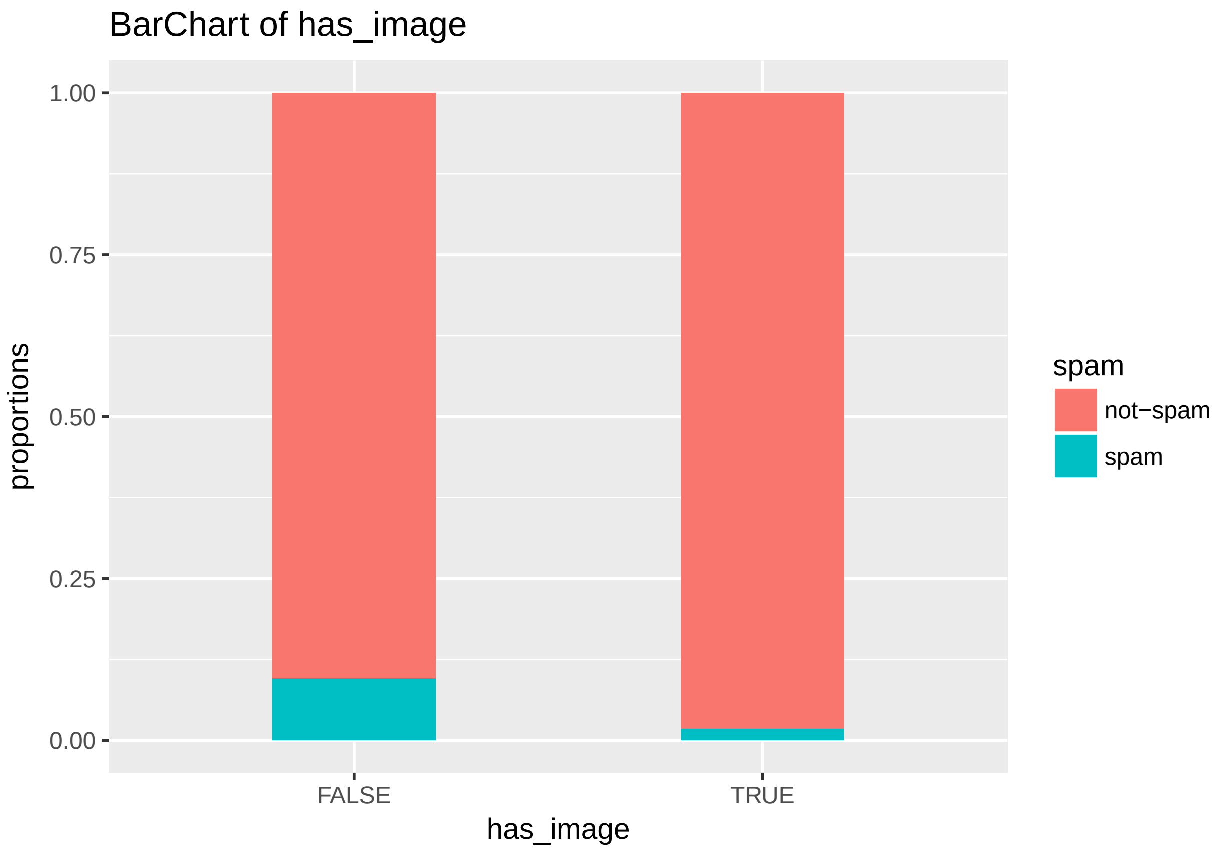
First let's explore the relationship between whether an email is spam or not and the number of images attached to it through a facetted histogram

```
email %>% ggplot() +  geom_histogram(aes(x = log(image+0.1))) + ggtitle("Historam of number of images i
```



We notice zero inflation i.e. most of the emails have zero images attached to them. To further explore this zero inflation, we create a new variable called has_image that is TRUE where the number of images is greater than zero and FALSE otherwise and generate an appropriate barchart to visualize the relationship between has_image and spam

```
email %>% mutate(has_image = image > 0) %>% group_by(spam) %>% ggplot() + geom_bar(aes(x = has_image, fi
```



Answering other questions

We can also anwser other questions like:

Q1. Within non-spam emails, is the typical length of emails shorter for those that were sent to multiple people?

```
email %>% filter(spam == "spam") %>% group_by(to_multiple) %>% summarize(median(num_char))
```

```
## # A tibble: 2 x 2
##   to_multiple `median(num_char)`
##         <dbl>              <dbl>
## 1           0              1.11
## 2           1              0.447
```

The answer seems to be yes, the typical length of non-spam sent to multiple people is a bit lower than those sent to only one person.

Q2. For emails containing the word "dollar", does the typical spam email contain a greater number of occurrences of the word than the typical non-spam email?

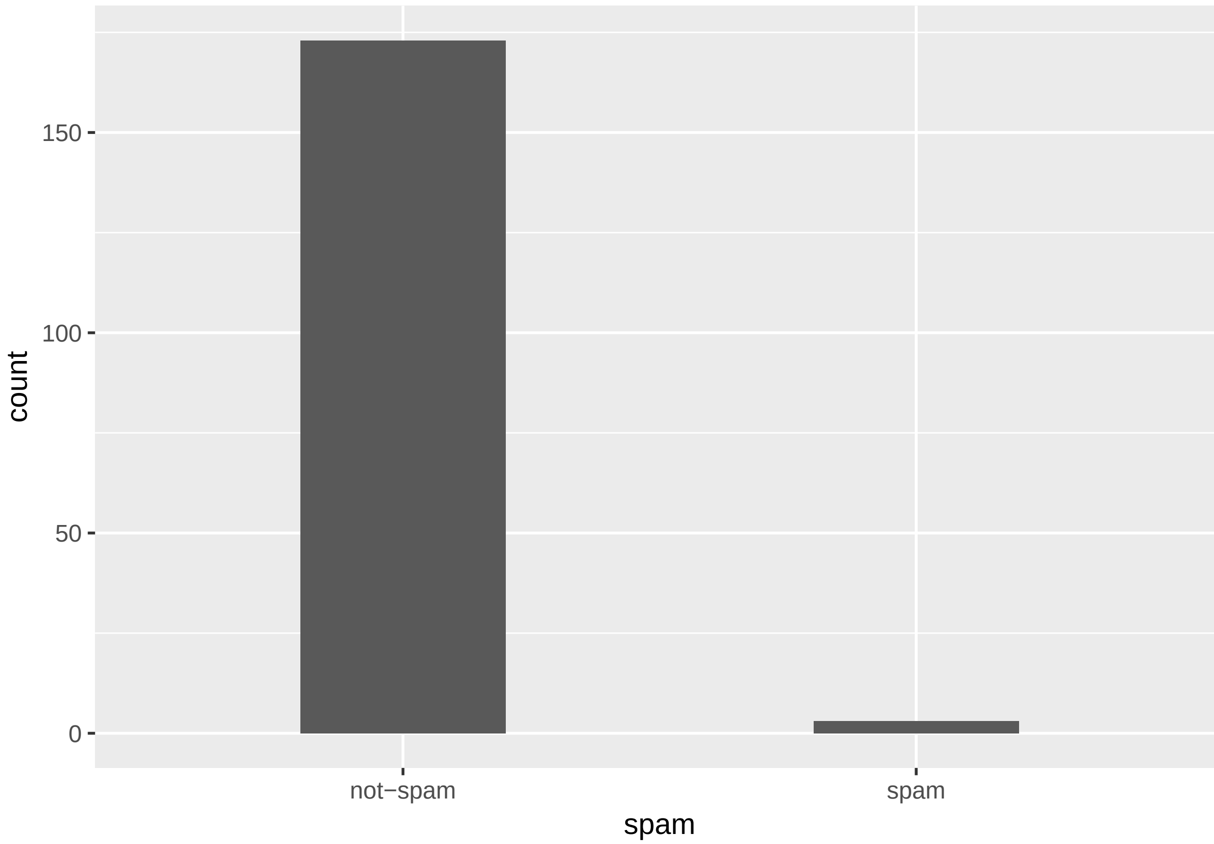
```
email %>% filter(dollar != 0) %>% group_by(spam) %>% summarise(median(dollar))
```

```
## # A tibble: 2 x 2
##   spam     `median(dollar)`
##   <chr>               <dbl>
## 1 not-spam                4
## 2 spam                    2
```

Here, the answer is no and a typical spam email contains less dollar word occurences than a real one.

Q3. If we encounter an email with greater than 10 occurrences of the word "dollar", is it more likely to be spam or not-spam?

```
email %>% filter(dollar > 10) %>% ggplot() + geom_bar(aes(spam), width = 0.4)
```

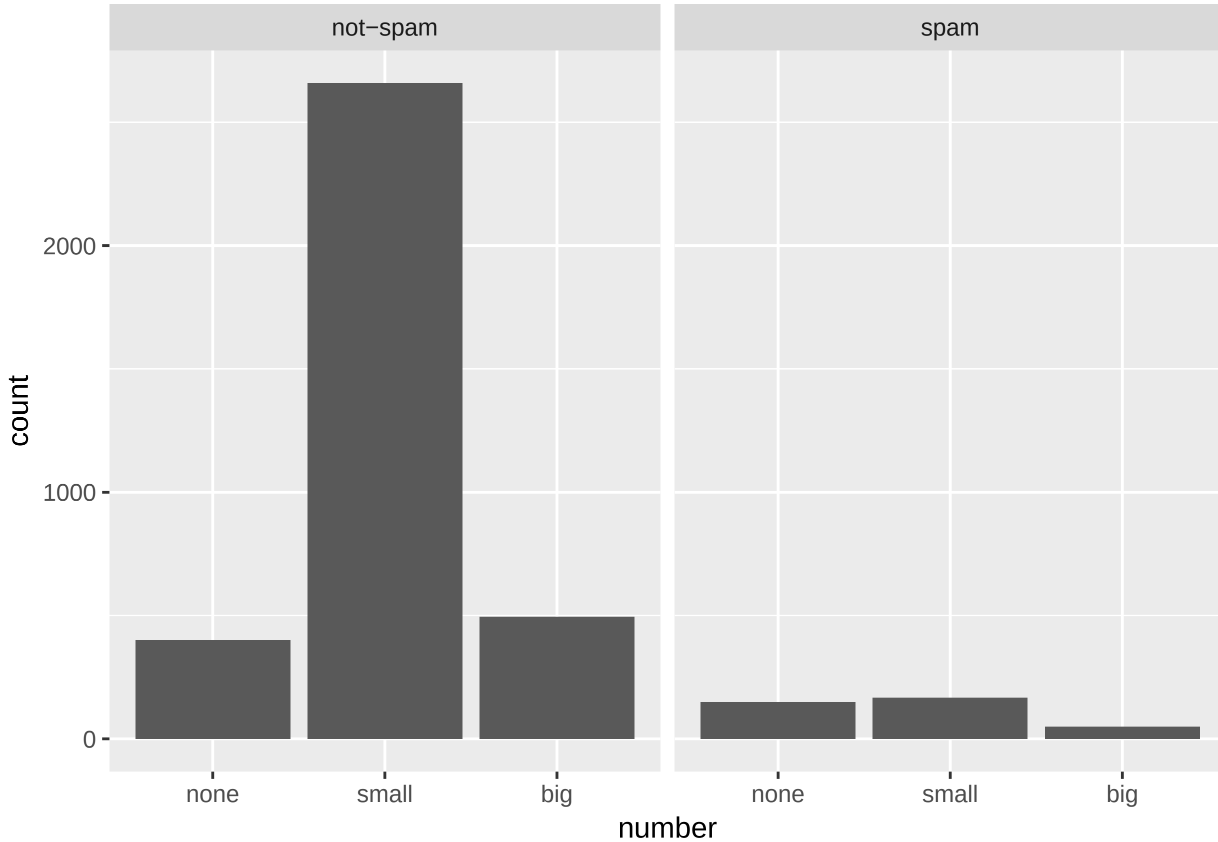


The email is more likely to be not-spam.

The dataset also contains a factor variable number describing whether a number was present in the email or not and if it was present what the size of the number was. `number` factor values tell us if there was no number, a small number (under 1 million), or a big number.

Q4. What is the association between `number` and spam?

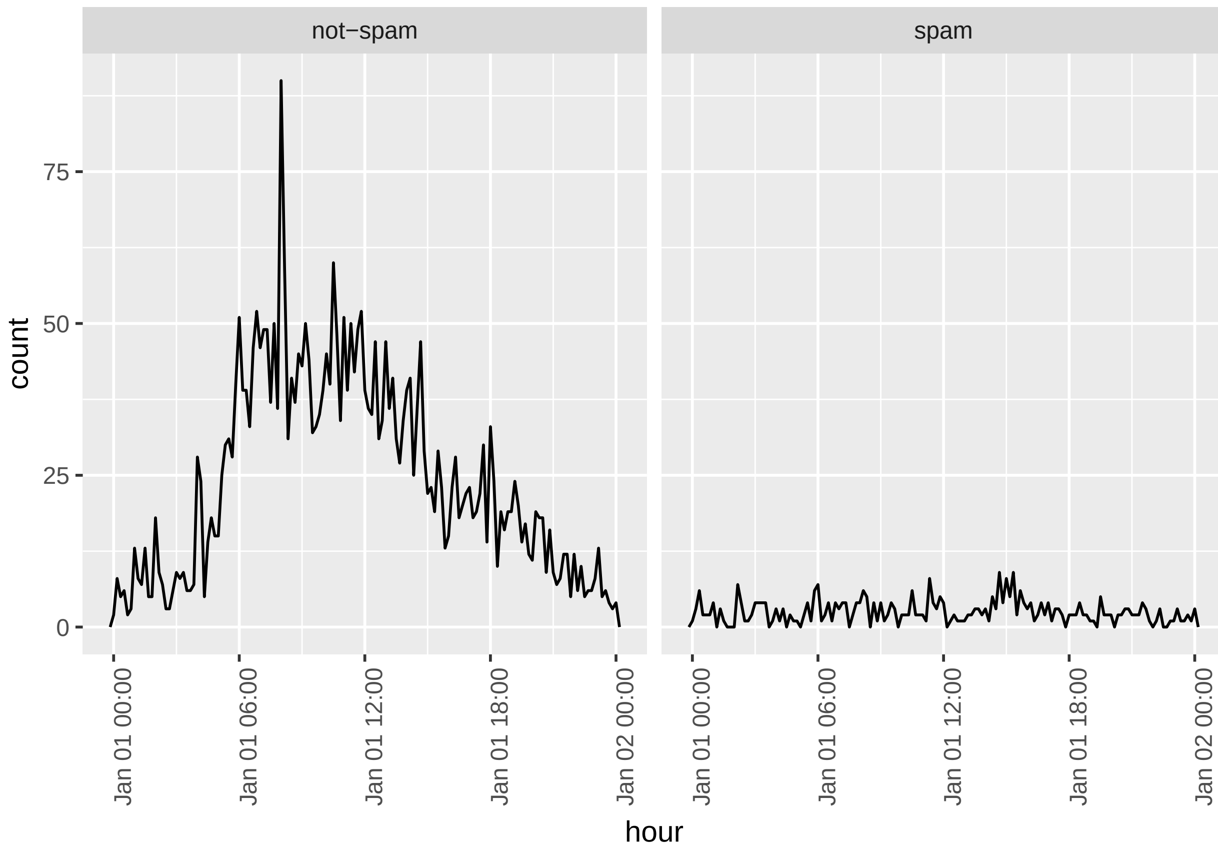
```
# Reorder levels
email$number <- factor(email$number, levels = c("none", "small","big"))

# Construct plot of number
email %>% ggplot()+ geom_bar(aes(x = number)) + facet_grid(~spam)
```



Surprisingly, the spam emails in this dataset contain less number of big numbers and also seem less likely to contain numbers compared to non-spam emails.

Q5. How is the time of the day at which the email was sent associated with spam i.e. is there a particular time of the day when more spam emails are sent?

```
email %>%
  ggplot(aes(hour)) +
  geom_freqpoly(binwidth = 600) + facet_grid(~ spam) + theme(axis.text.x = element_text(angle = 90))
```



Unlike non-spam emails which peak at certain times during the day we notice that the voulme of spam emails remains relatively constant throughout the day

Correlogram of keywords

Let's see which variable is correlated with which variable and then construct two Correlograms for spam and non-spam emails

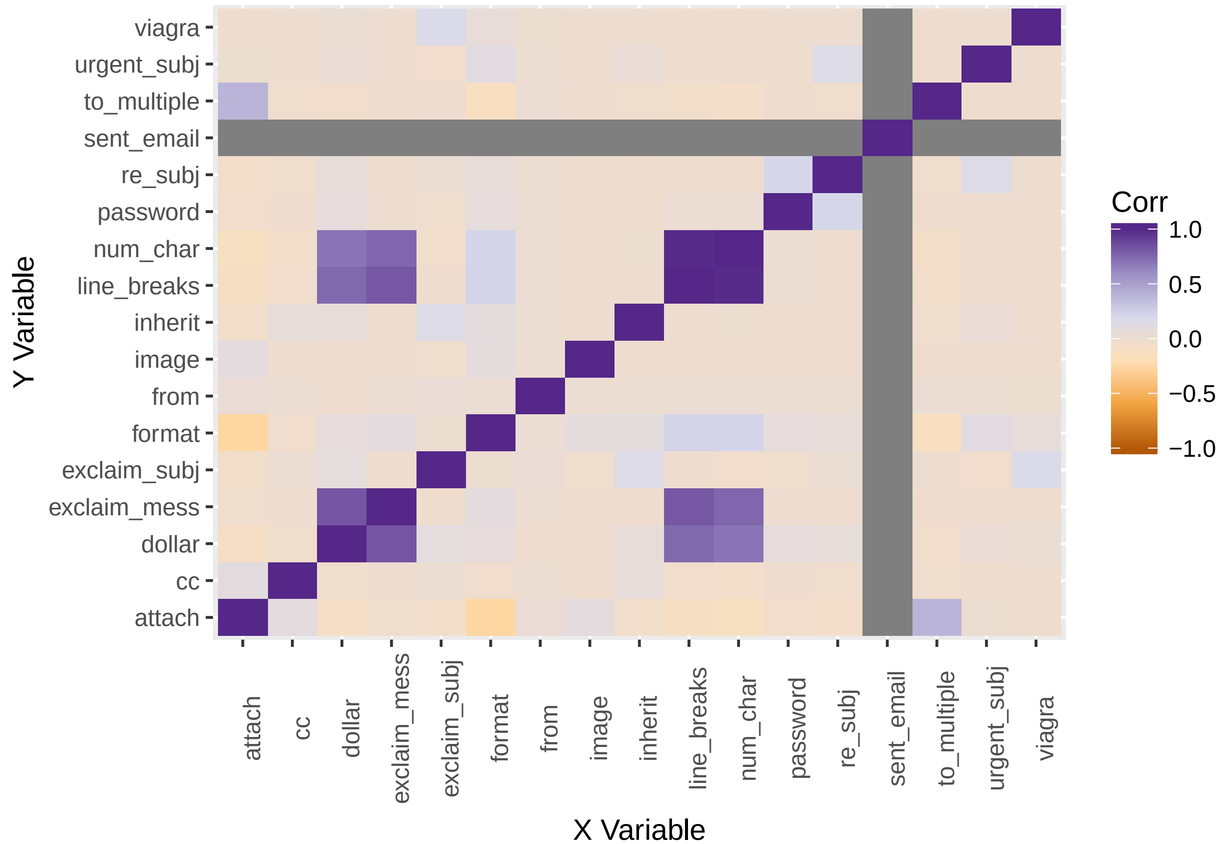
```
corrlgm <- function(df) {
  df %>% cor %>% as.data.frame %>% gather(xVar, Corr) %>% mutate(yVar = rep(names(df), times = length(d
    ggplot(aes(xVar, yVar, fill = Corr)) + geom_tile() + scale_fill_distiller(type = "div", palette = 4
    labs(x = "X Variable", y = "Y Variable") + theme(axis.text.x = element_text(angle = 90))
}
find_high_cor <- function(df, corr = .6) {
  cor_df <- df %>% cor %>% as.data.frame %>% gather(xVar, Corr) %>% mutate(yVar = rep(names(df), times =
    filter(Corr >= corr | Corr <= -corr, Corr != 1) %>% arrange(desc(Corr))
  return(c(cor_df$xVar, cor_df$yVar) %>% unique)
}

#Spam
find_high_cor(email %>% filter(spam == "spam") %>% select_if(is.numeric))
```

```
## Warning in cor(.): the standard deviation is zero
```

```
## [1] "num_char"     "line_breaks"  "dollar"       "exclaim_mess"
```

```r
corrlgm(email %>%filter(spam == "spam") %>% select_if(is.numeric))
```

```
## Warning in cor(.): the standard deviation is zero
```



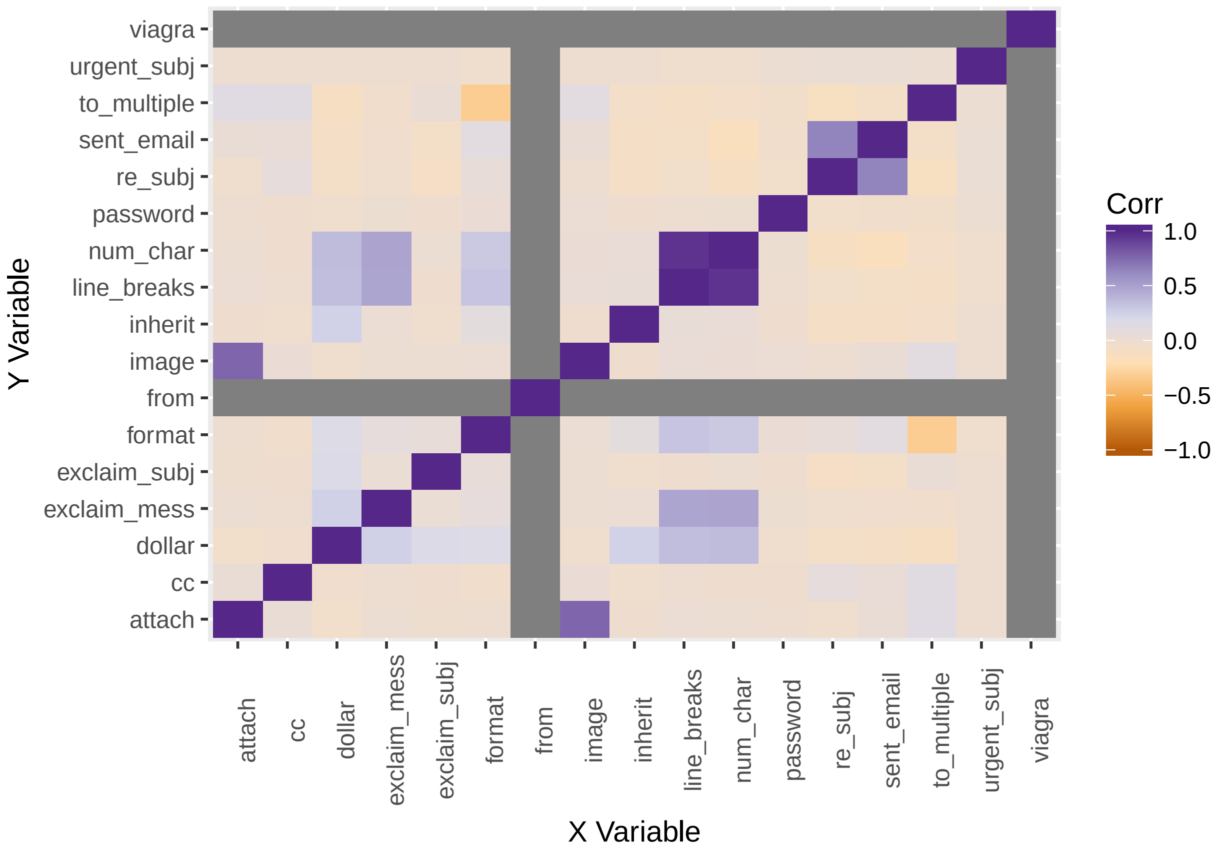

```r
#Not-Spam
find_high_cor(email %>% filter(spam == "not-spam") %>% select_if(is.numeric))
```

```
## Warning in cor(.): the standard deviation is zero
```

```
## [1] "num_char"    "line_breaks" "image"       "attach"      "sent_email"
## [6] "re_subj"
```

```r
corrlgm(email %>% filter(spam == "not-spam") %>% select_if(is.numeric))
```

```
## Warning in cor(.): the standard deviation is zero
```



viagra
urgent_subj
to_multiple
sent_email
re_subj
password
num_char
line_breaks
inherit
image
from
format
exclaim_subj
exclaim_mess
dollar
cc
attach
Y Variable
attach
cc
dollar
exclaim_mess
exclaim_subj
format
from
image
inherit
line_breaks
num_char
password
re_subj
sent_email
to_multiple
urgent_subj
viagra
X Variable
Corr
1.0
0.5
0.0
−0.5
−1.0