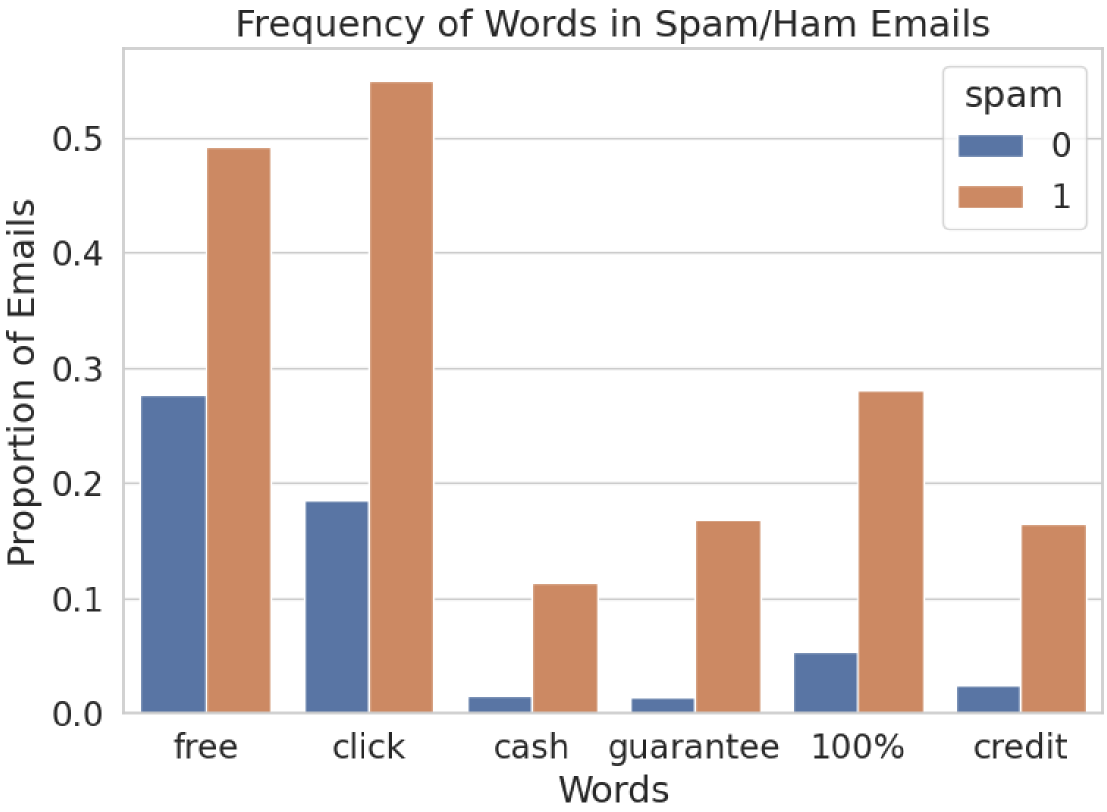
0.1 Question 1

Discuss one attribute or characteristic you notice that is different between the two emails that may allow you to uniquely identify a spam email.

One of the main differences is formatting type. The first ham is in a plain text format while the identified spam is in HTML format. Another difference is the the links contained in the email. The ham has friendly urls while the spam has a url pointing to an IP address. The spam email contains language that is more fishy, like it wants an action related to either clicking a link or paying for something. The ham email body is legibly formatted in plaintext with a date and some embedded urls.

Create your bar chart in the following cell:

```
In [13]: train = train.reset_index(drop=True) # We must do this in order to preserve the ordering of em
         plt.figure(figsize=(8,6))
         selected_words = ['free', 'click', 'cash', 'guarantee', '100%', 'credit']
         by_word = pd.DataFrame(words_in_texts(selected_words, train['email']))
         by_word = by_word.rename(columns=dict([(k,v) for k,v in zip([0,1,2,3,4,5], selected_words)]))
         by_word['spam'] = train['spam']
         by_word = by_word.melt('spam')
         sns.barplot(x=by_word['variable'], y=by_word['value'], hue=by_word['spam'], errorbar=None)
         plt.title('Frequency of Words in Spam/Ham Emails')
         plt.ylabel('Proportion of Emails')
         plt.xlabel('Words')
         #by_word.head()
         #train.head()
         plt.tight_layout()
         plt.show()
```



0.2 Question 6c

Explain your results in `q6a` and `q6b`. How did you know what to assign to `zero_predictor_fp`, `zero_predictor_fn`, `zero_predictor_acc`, and `zero_predictor_recall`?

The zero_predictor function sends all emails, regardless of any feature, to the inbox. In other words, all emails, whether spam or ham, will get classified as ham. The zero_predictor catches absolutely no spam emails. Thus, the number of true positives (spam classified as spam) and false positives (ham classified as spam) are both zero. The number of true negatives (ham classified as ham) and false negatives (spam classified as ham) will simply be the number of ham emails and the number of spam emails, respectively, since all ham and all spam get classified as ham. Finally, these four values help explain accuracy and recall of the zero_predictor model. Accuracy refers to the proportion of emails classified correctly: TP + TN / number of emails. No spam is caught (TP), so we get back the proportion of ham emails in the training data. Recall refers to of all predicted positive, the proportion classified correctly: TP / TP + FN. Since we catch no spam (TP), the recall of this model is 0.

0.3 Question 6f

How does the accuracy of the logistic regression classifier `my_model` compare to the accuracy of the zero predictor?

The training accuracy of my logistic regression classifier (0.7576) is slightly higher than the zero predictor (0.7447). This suggests that my_model outperforms the zero predictor, even though the recall (0.11) and precision (0.64) for my_model are lower than ideal. This indicates that my_model is still better than a naive model that makes no real predictions.

0.4 Question 6g

Given the word features provided in Question 4, discuss why the logistic regression classifier `my_model` may be performing poorly.

Hint: Think about how prevalent these words are in the email set.

There may be class imbalance, where the ham emails are more prevalent than the spam. We are getting a lot of false negatives, so our model is overfitting to the majority class most likely, even though the words we chose may be more associated with the spam. This imbalance means that the model might predict ham for many emails because they are more common and our model may be more conservative. To add, our choice of words may be prevalent in both classes (not exclusive), so this can explain why our model may be incorrectly predicting. Some of these words can definitely appear in spam. However, many of these words can also be used in ham emails. For example, a spam email containing the word "bank" could be an email fishing for a user's bank account information. A ham email containing the word "bank" could be an email from a user's actual bank signed off with "Chase Bank". Additionally, if the dataset has a large vocabulary and many of the words only appear in a small number of emails, the model might not have enough data to generalize well. The model might not be using the most discriminative words for the task at hand.

0.5 Question 6h

Would you prefer to use the logistic regression classifier `my_model` or the zero predictor classifier for a spam filter? Why? Describe your reasoning and relate it to at least one of the evaluation metrics you have computed so far.

I prefer the logistic regression classifier. The zero predictor classifier sends all emails to inbox. The false positives and true positives are effectively zero, making the accuracy low. It is not filtering. The logistic regression classifier has already shown some improvement with the accuracy score over the zero predictor accuracy. Our logistic regression model makes an attempt to classify spam emails at least, and is still better than not catcing any spam emails at all (recall of 0.11 vs recall of 0). Further, the precision for my_model is 0.64, meaning that 64% of the emails predicted as spam are actually spam. While not ideal, this is a reasonable precision considering the low recall. On the other hand, the zero predictor classifier has a precision of 0, since it never predicts spam and does not deal with false positives in the spam class (but it does suffer from a high number of false negatives). With better features, I believe the logistic regression classifier will be able to better catch more false positives, making accuracy better since it is filtering.