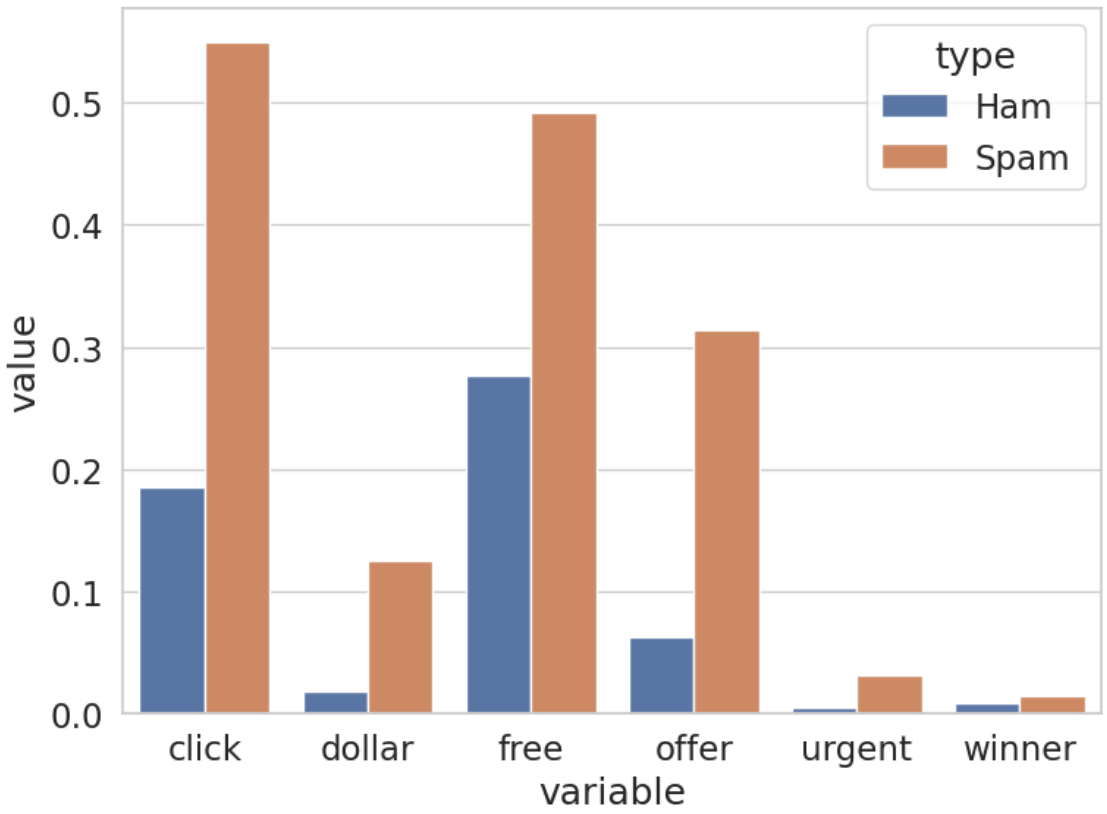

0.1 Question 1

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

The spam email has a bunch of html tags, and uses a lot more promotional languauge, and has an unreadable url. The ham email has a lot more normaly spoken text and readable less suspicious urls.

Create your bar chart in the following cell:

```
In [23]: train = train.reset_index(drop=True) # We must do this in order to preserve the ordering of em
         plt.figure(figsize=(8,6))

         words = ['free', 'click', 'winner', 'offer', 'dollar', 'urgent']
         word_array = words_in_texts(words, train['email'])

         word_df = pd.DataFrame(word_array, columns = words)
         word_df['type'] = train['spam'].map({0: 'Ham', 1: 'Spam'})

         melted = word_df.melt(id_vars= 'type')
         grouped = melted.groupby(['type', 'variable'])['value'].mean().reset_index()

         sns.barplot(data=grouped, x='variable', y='value', hue='type')

         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`?

zero_predictor_fp - happens when we incorrectly predict ham. Since the zero predictor never predicts anything as spam, it can't make false positives, so fp=0.

zero_predictor_fn - happens when a spam email is mislabeled as ham. Since every ham is labeled 0, the number of false negatives is the sum of all the training data.

zero_predictor_acc - accurace is the number of correct predictions / total predictions made. This translates to the number of ham emails or (len(Y_train) - sum(Y_train)) / the total number of emails or len(Y_train)

zero_predictor_recall - recall is equal to the number of true positives divided by the total total number of spam emails. Since the predictor never identifies anything as spam, this number is 0 / total emails = 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 accuracy of my_model tries to distinguish between spam and ham based on word presence. While my_model makes false positives and false negatives, it usually establishes a better balance between precision and recall with higher accuracy. While the zero predictor always predicts 0, so its accuracy is just the proportion of ham emails in the training data. The zero predictor may have decent accuracy when most emails are ham, but it completely fails in identifying spam, meaning it has zero recall and several false negatives.

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.

my_model was likely performing poorly with the features provided in question 4 because those words only apply in such a small number of emails, that the rows in its feature matrix are just 0 which makes it difficult to find patterns. Also, some of the words like 'private' and 'memo' are pretty likely to appear in both spam and ham emails, as they are just common email topics in general.

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 would rather use the logistic regregression classifier. Even though the zero predictor can have decent accuracy, its complete inability to predict spam makes it useless as a spam filter. my_model will detect some spam meaning it can succesfully stop people from recieving spam.