Notebook

April 20, 2020

0.0.1 Question 1c

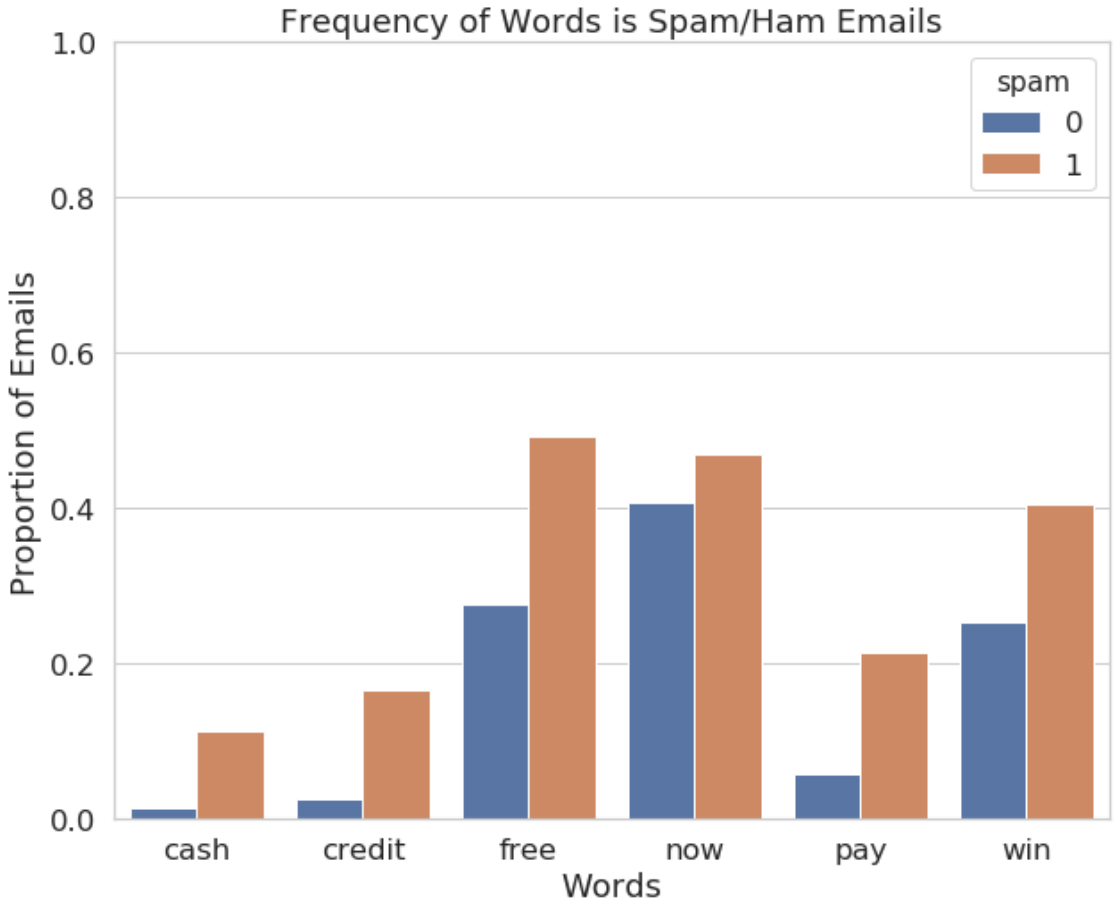
Discuss one thing you notice that is different between the two emails that might relate to the identification of spam.

One aspect that is different among the spam and 'ham' email is that the 'ham' email begins with an existing URL while the spam email begins with a webpage set up via HTML.

0.0.2 Question 3a

Create a bar chart like the one above comparing the proportion of spam and ham emails containing certain words. Choose a set of words that are different from the ones above, but also have different proportions for the two classes. Make sure to only consider emails from `train`.

In [103]:
```
train=train.reset_index(drop=True) # We must do this in order to preserve the ordering of ema
words = ['credit', 'win', 'pay', 'cash', 'now', 'free']

df_words = pd.DataFrame(words_in_texts(words, train['email']), columns = words)

df_words['spam'] = train['spam']

df_melt = df_words.melt("spam")

df_group_var = df_melt.groupby(['spam', 'variable']).apply(np.mean).drop(columns = ['spam']).

plt.figure(figsize=(10, 8))

sns.barplot(x= 'variable', y = 'value', hue = 'spam', data = df_group_var)

plt.xlabel('Words')
plt.ylabel('Proportion of Emails')
plt.title('Frequency of Words is Spam/Ham Emails')
plt.ylim((0.0, 1.0));
```



0.0.3 Question 3b

Create a *class conditional density plot* like the one above (using `sns.distplot`), comparing the distribution of the length of spam emails to the distribution of the length of ham emails in the training set. Set the x-axis limit from 0 to 50000.

In [123]:
```
train_0 = train[train['spam'] == 0]
train_1 = train[train['spam'] == 1]

train_0.loc[:, 'email_length'] = train_0['email'].apply(len).values
train_1.loc[:, 'email_length'] = train_1['email'].apply(len).values

sns.distplot(a = train_0['email'].apply(len).values, hist = False, label = 'Ham')
sns.distplot(a = train_1['email'].apply(len).values, hist = False, label = 'Spam')

plt.xlabel('Length of email body')
plt.ylabel('Distribution')
plt.xlim(0, 50000);
```