In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df = pd.read_csv('Reviews.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Su
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Dc
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Ad
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"J fa

In [4]:
```
df.shape
```

Out[4]:
```
(568454, 10)
```

In [5]:
```
df.info()
```

Out[5]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
 #   Column                  Non-Null Count   Dtype 
---  ------                  --------------   ----- 
 0   Id                      568454 non-null  int64 
 1   ProductId               568454 non-null  object
 2   UserId                  568454 non-null  object
 3   ProfileName             568438 non-null  object
 4   HelpfulnessNumerator    568454 non-null  int64 
 5   HelpfulnessDenominator  568454 non-null  int64 
 6   Score                   568454 non-null  int64 
 7   Time                    568454 non-null  int64 
 8   Summary                 568427 non-null  object
 9   Text                    568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

In [6]:
```
df.Text.head()
```

Out[6]:
```
0    I have bought several of the Vitality canned d...
1    Product arrived labeled as Jumbo Salted Peanut...
2    This is a confection that has been around a fe...
3    If you are looking for the secret ingredient i...
4    Great taffy at a great price.  There was a wid...
Name: Text, dtype: object
```

In [7]:
```
df.Summary.head()
```

Out[7]:
```
0    Good Quality Dog Food
1        Not as Advertised
2    "Delight" says it all
3           Cough Medicine
4              Great taffy
Name: Summary, dtype: object
```

In [8]:
```
!pip install vaderSentiment
```

```
Requirement already satisfied: vaderSentiment in c:\users\hp\anaconda3\lib\site-packages (3.3.2)
Requirement already satisfied: requests in c:\users\hp\anaconda3\lib\site-packages (from vaderSentiment) (2.24.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests->vaderSentiment) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests->vaderSentiment) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\hp\anaconda3\lib\site-packages (from requests->vaderSentiment) (3.0.4)
```

In [9]:
```
import seaborn as sns
import re
import os
import sys
import ast
plt.style.use('fivethirtyeight')
# Function for getting the sentiment
cp = sns.color_palette()
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()
```

In [10]:
```
# Generating sentiment for all the sentence present in the dataset
emptylines=[]
for row in df['Text']:
    vs=analyzer.polarity_scores(row)
    emptylines.append(vs)
# Creating new dataframe with sentiments
df_sentiments=pd.DataFrame(emptylines)
df_sentiments.head()
```

Out[10]:

	neg	neu	pos	compound
0	0.000	0.711	0.289	0.8441
1	0.138	0.862	0.000	-0.5664
2	0.085	0.771	0.144	0.8139
3	0.000	0.932	0.068	0.4404
4	0.000	0.599	0.401	0.9468

In [11]:
```
df_c = pd.concat([df.reset_index(drop=True), df_sentiments], axis=1)
df_c.head(5)
```

Out[11]:

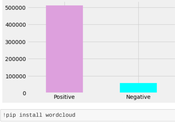
	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Su
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Dc
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Ad
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"J fa
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	W
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Gr

In [12]:
```
# Convert scores into positive and negetive sentiments using some threshold
df_c['Sentiment'] = np.where(df_c['compound'] >= 0 , 'Positive','Negative')
df_c.head(5)
```

Out[12]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Su
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Dc
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Ad
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"J fa
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	W
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Gr

In [13]:
```
result=df_c['Sentiment'].value_counts()
result.plot(kind='bar', rot=0, color=['plum','cyan']);
```



In [14]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\hp\anaconda3\lib\site-packages (1.8.1)
Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\site-packages (from wordcloud) (3.2.2)
Requirement already satisfied: pillow in c:\users\hp\anaconda3\lib\site-packages (from wordcloud) (7.2.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\hp\anaconda3\lib\site-packages (from wordcloud) (1.18.5)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.1)
```

In [17]:
```
from wordcloud import WordCloud
from wordcloud import STOPWORDS
```

In [18]:
```
df.columns
```

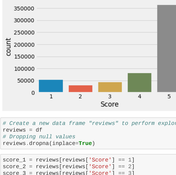
Out[18]:
```
Index(['Id', 'ProductId', 'UserId', 'ProfileName', 'HelpfulnessNumerator',
       'HelpfulnessDenominator', 'Score', 'Time', 'Summary', 'Text'],
      dtype='object')
```

In [19]:
```
df.Score.value_counts()
```

Out[19]:
```
5    363122
4     80655
1     52268
3     42640
2     29769
Name: Score, dtype: int64
```

In [20]:
```
sns.countplot(data = df, x = 'Score')
```

Out[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e90e101b60>
```



In [23]:
```
# Create a new data frame "reviews" to perform exploratory data analysis upon that
reviews = df
# Dropping null values
reviews.dropna(inplace=True)
```

In [24]:
```
score_1 = reviews[reviews['Score'] == 1]
score_2 = reviews[reviews['Score'] == 2]
score_3 = reviews[reviews['Score'] == 3]
score_4 = reviews[reviews['Score'] == 4]
score_5 = reviews[reviews['Score'] == 5]
```

In [25]:
```
reviews_sample = pd.concat([score_1,score_2,score_3,score_4,score_5],axis=0)
reviews_sample.reset_index(drop=True,inplace=True)
```

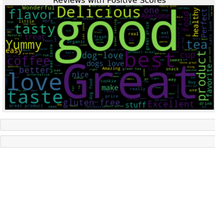
In [26]:
```
#Wordcloud function's input needs to be a single string of text.
# concatenating all Summaries into a single string.
# similarly you can build for Text column
reviews_str = reviews_sample.Summary.str.cat()
wordcloud = WordCloud(background_color='white').generate(reviews_str)
plt.figure(figsize=(10,10))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
plt.show()
```

In [27]:
```
# Now let's split the data into Negative (Score is 1 or 2) and Positive (4 or 5) Reviews.
negative_reviews = reviews_sample[reviews_sample['Score'].isin([1,2])]
positive_reviews = reviews_sample[reviews_sample['Score'].isin([4,5])]
# Transform to single string
negative_reviews_str = negative_reviews.Summary.str.cat()
positive_reviews_str = positive_reviews.Summary.str.cat()
```

In [28]:
```
wordcloud_negative = WordCloud(background_color='white').generate(negative_reviews_str)
wordcloud_positive = WordCloud(background_color='black').generate(positive_reviews_str)
# Plot
fig = plt.figure(figsize=(10,10))
ax1 = fig.add_subplot(211)
ax1.imshow(wordcloud_negative,interpolation='bilinear')
ax1.axis('off')
ax1.set_title('Reviews with Negative Scores',fontsize=20)
```

Out[28]:
```
Text(0.5, 1.0, 'Reviews with Negative Scores')
```


In [29]:
```
fig = plt.figure(figsize=(10,10))
ax2 = fig.add_subplot(212)
ax2.imshow(wordcloud_positive,interpolation='bilinear')
ax2.axis('off')
ax2.set_title('Reviews with Positive Scores',fontsize=20)
plt.show()
```



In [ ]:
```
```

In [ ]:
```
```