In [5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('Reviews.csv', nrows=500)
df.head(3)
```

Out[5]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDen
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Id                      500 non-null    int64 
 1   ProductId               500 non-null    object
 2   UserId                  500 non-null    object
 3   ProfileName             500 non-null    object
 4   HelpfulnessNumerator    500 non-null    int64 
 5   HelpfulnessDenominator  500 non-null    int64 
 6   Score                   500 non-null    int64 
 7   Time                    500 non-null    int64 
 8   Summary                 500 non-null    object
 9   Text                    500 non-null    object
dtypes: int64(5), object(5)
memory usage: 39.2+ KB
```

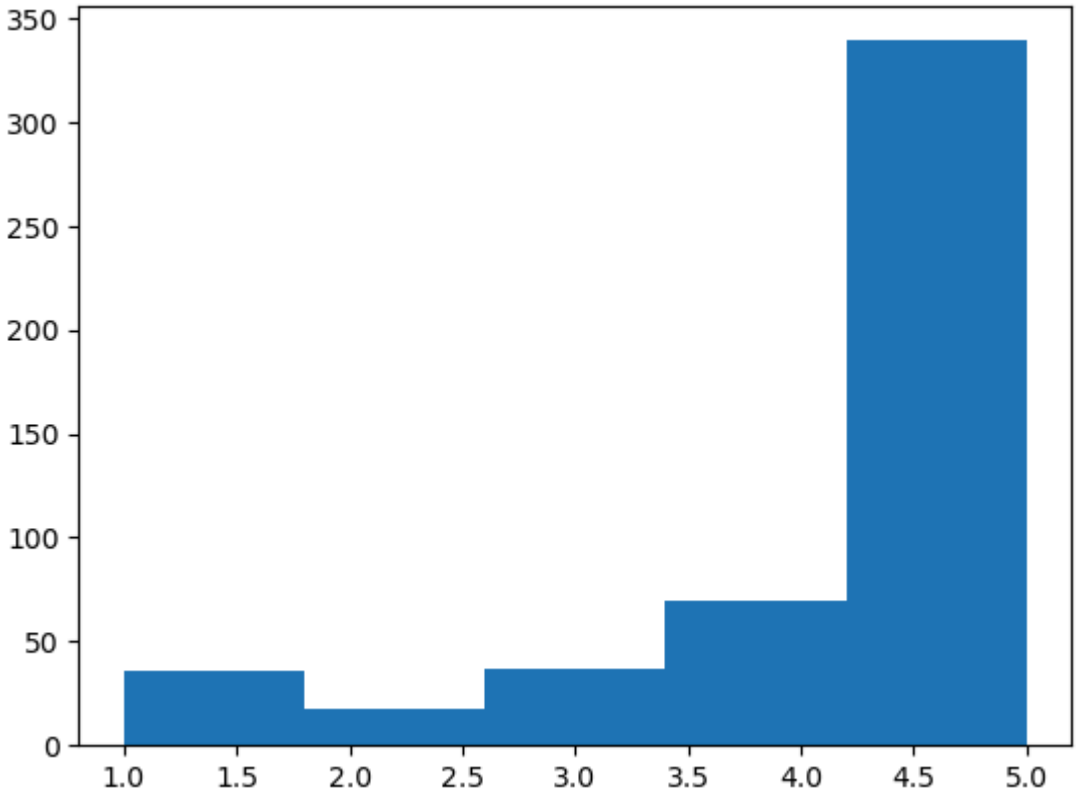
In [7]:
```
df.Summary.head()
```

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]:
```
df.Text.head()
```

Out[8]:
```
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 [12]:
```
import pandas as pd
import matplotlib.pyplot as plt
# Create a new data frame "reviews" to perform exploration
reviews = df
# Dropping null values
reviews.dropna(inplace=True)
# The histogram reveals this dataset is highly unbalanced
reviews.Score.hist(bins=5, grid=False)
plt.show()
print(reviews.groupby('Score').count().Id)
```




```
Score
1     36
2     18
3     37
4     70
5    339
Name: Id, dtype: int64
```

In [13]:
```
score_1 = reviews[reviews['Score'] == 1].sample(n=18)
score_2 = reviews[reviews['Score'] == 2].sample(n=18)
score_3 = reviews[reviews['Score'] == 3].sample(n=18)
score_4 = reviews[reviews['Score'] == 4].sample(n=18)
score_5 = reviews[reviews['Score'] == 5].sample(n=18)
```

In [16]:
```
# Here we recreate a 'balanced' dataset.
reviews_sample = pd.concat([score_1,score_2,score_3,score_4,score_5])
reviews_sample.reset_index(drop=True,inplace=True)
# Printing count by 'Score' to check dataset is now balanced.
print(reviews_sample.groupby('Score').count().Id)
```

```
Score
1    18
2    18
3    18
4    18
5    18
Name: Id, dtype: int64
```

In [17]:
```
from wordcloud import WordCloud
reviews_str = " ".join(reviews_sample["Summary"].to_numpy())
wordcloud = WordCloud(background_color='white').generate(reviews_str)
plt.figure(figsize=(10,10))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis("off")
plt.show()
```

In [18]:
```
# Splitting the data into Negative and Positive reviews
negative_reviews = reviews_sample[reviews_sample['Score'].isin([1,2])]
positive_reviews = reviews_sample[reviews_sample['Score'].isin([4,5])]
# Transforming the reviews into a single string
negative_reviews_str = negative_reviews.Summary.str.cat()
positive_reviews_str = positive_reviews.Summary.str.cat()
```

In [19]:
```
wordcloud_negative = WordCloud(background_color='white') \
 .generate(negative_reviews_str)
wordcloud_positive = WordCloud(background_color='white') \
 .generate(positive_reviews_str)
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)
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 [20]:
```
!pip install vaderSentiment
```

```
0:00:02
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
   ---------------------------------- ---- 112.6/126.0 kB 25.0 kB/s eta
0:00:01
```

In [22]:
```
import seaborn as sns
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
plt.style.use('fivethirtyeight')
# Function for getting the sentiment
cp = sns.color_palette()
analyzer = SentimentIntensityAnalyzer()
# Generating sentiment for all the sentence present in the dataset
emptyline=[]
for row in df['Text']:
 vs=analyzer.polarity_scores(row)
 emptyline.append(vs)
```

In [23]:
```
df_sentiments=pd.DataFrame(emptyline)
df_sentiments.head()
```

Out[23]:

	neg	neu	pos	compound
0	0.000	0.711	0.289	0.9441
1	0.138	0.862	0.000	-0.5664
2	0.085	0.771	0.144	0.8138
3	0.000	0.932	0.068	0.4404
4	0.000	0.599	0.401	0.9468

In [ ]: