In [1]:
```
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[1]:

	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 [2]:
```
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 [3]:
```
df.Summary.head()
```

Out[3]:
```
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 [4]:
```
df.Text.head()
```

Out[4]:
```
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 [5]:
```
from nltk.corpus import stopwords
from textblob import TextBlob
from textblob import Word

# Lower casing and removing punctuations
df['Text'] = df['Text'].apply(lambda x: " ".join(x.lower() for x in x.s
df['Text'] = df['Text'].str.replace('[^\w\s]', '')

# Removal of stop words
stop = stopwords.words('english')
df['Text'] = df['Text'].apply(lambda x: " ".join(x for x in x.split() i

# Spelling correction
df['Text'] = df['Text'].apply(lambda x: str(TextBlob(x).correct()))

# Lemmatization
df['Text'] = df['Text'].apply(lambda x: " ".join([Word(word).lemmatize(

df.Text.head()
```

```
C:\Users\nihar\AppData\Local\Temp\ipykernel_12368\1893040307.py:7: FutureW
arning: The default value of regex will change from True to False in a fut
ure version.
  df['Text'] = df['Text'].str.replace('[^\w\s]', '')
```

Out[5]:
```
0    bought several vitality canned dog food produc...
1    product arrived labelled lumbo halted peanutst...
2    connection around century light pillow city ge...
3    looking secret ingredient robitussin believe f...
4    great staff great price wide assortment mummy ...
Name: Text, dtype: object
```

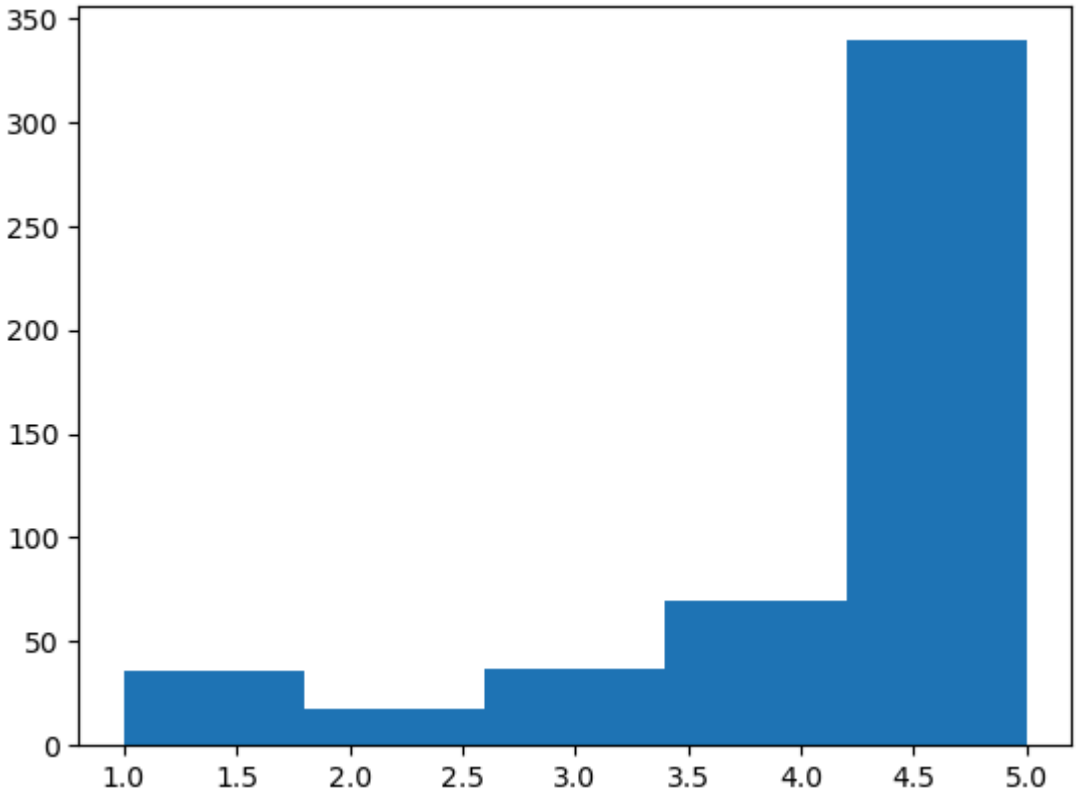
In [7]:
```
import pandas as pd
import matplotlib.pyplot as plt

# 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 [8]:
```
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 [9]:
```
# Here we recreate a 'balanced' dataset.
reviews_sample = pd.concat([score_1,score_2,score_3,score_4,score_5],ax
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 [10]:
```
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 [ ]:
```
1
```