In [59]:

```
# Basic libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from collections import Counter
from wordcloud import WordCloud
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS

# Configure plot style
sns.set(style="whitegrid", palette="muted", font_scale=1.1)
%matplotlib inline
```

In [60]:

```
# Load Amazon Fine Food Reviews dataset
df = pd.read_csv("Reviews.csv")

# Select relevant columns
df = df[["Text", "Score", "Time", "Summary", "ProductId", "UserId"]].dropna()


# Show first 5 rows
df.head()
```

Out[60]:

	Text	Score	Time	Summary	ProductId	UserId
0	I have bought several of the Vitality canned d...	5	1303862400	Good Quality Dog Food	B001E4KFG0	A3SGXH7AUHU8GW
1	Product arrived labeled as Jumbo Salted Peanut...	1	1346976000	Not as Advertised	B00813GRG4	A1D87F6ZCVE5NK
2	This is a confection that has been around a fe...	4	1219017600	"Delight" says it all	B000LQOCH0	ABXLMWJIXXAIN
3	If you are looking for the secret ingredient i...	2	1307923200	Cough Medicine	B000UA0QIQ	A395BORC6FGVXV
4	Great taffy at a great price. There was a wid...	5	1350777600	Great taffy	B006K2ZZ7K	A1UQRSCLF8GW1T

In [61]:

```
# Dataset info
df.info()

# Basic statistics
df.describe()

# Check for missing values
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 568427 entries, 0 to 568453
Data columns (total 6 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   Text       568427 non-null  object
 1   Score      568427 non-null  int64 
 2   Time       568427 non-null  int64 
 3   Summary    568427 non-null  object
 4   ProductId  568427 non-null  object
 5   UserId     568427 non-null  object
dtypes: int64(2), object(4)
memory usage: 30.4+ MB
```

Out[61]:

```
Text         0
Score        0
Time         0
Summary      0
ProductId    0
UserId       0
dtype: int64
```

In [62]:

```
# Clean text: lowercase, remove special characters, numbers, extra spaces
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r"[^a-z\s]", " ", text)  # remove non-alpha
    text = re.sub(r"\s+", " ", text).strip()
    return text

df["clean_text"] = df["Text"].apply(clean_text)

# Review length (number of words)
df["review_length"] = df["clean_text"].apply(lambda x: len(x.split()))

df.head()
```

Out[62]:

	Text	Score	Time	Summary	ProductId	UserId	clean_text	review_length
0	I have bought several of the Vitality canned d...	5	1303862400	Good Quality Dog Food	B001E4KFG0	A3SGXH7AUHU8GW	i have bought several of the vitality canned d...	48
1	Product arrived labeled as Jumbo Salted Peanut...	1	1346976000	Not as Advertised	B00813GRG4	A1D87F6ZCVE5NK	product arrived labeled as jumbo salted peanut...	32
2	This is a confection that has been around a fe...	4	1219017600	"Delight" says it all	B000LQOCH0	ABXLMWJIXXAIN	this is a confection that has been around a fe...	93
3	If you are looking for the secret ingredient i...	2	1307923200	Cough Medicine	B000UA0QIQ	A395BORC6FGVXV	if you are looking for the secret ingredient i...	41
4	Great taffy at a great price. There was a wid...	5	1350777600	Great taffy	B006K2ZZ7K	A1UQRSCLF8GW1T	great taffy at a great price there was a wide ...	27

In [63]:

```
# Map score to sentiment
def score_to_sentiment(score):
    if score <= 2:
        return "Negative"
    elif score == 3:
        return "Neutral"
    else:
        return "Positive"

df["Sentiment"] = df["Score"].apply(score_to_sentiment)

# Count of each sentiment
df["Sentiment"].value_counts()
```

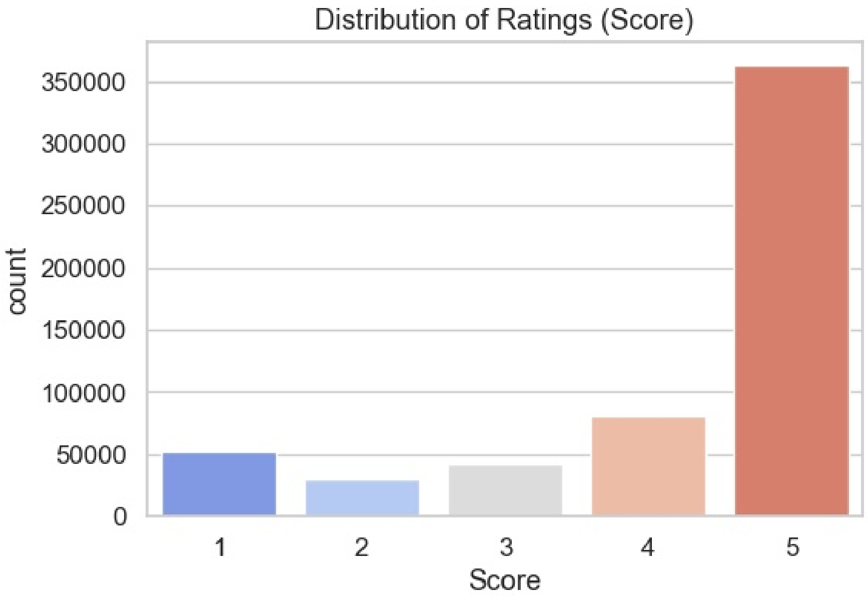
Out[63]:

```
Sentiment
Positive    443777
Negative     82012
Neutral      42638
Name: count, dtype: int64
```

In [64]:

```
plt.figure(figsize=(6,4))
sns.countplot(x="Score", data=df, palette="coolwarm")
plt.title("Distribution of Ratings (Score)")
plt.show()
```

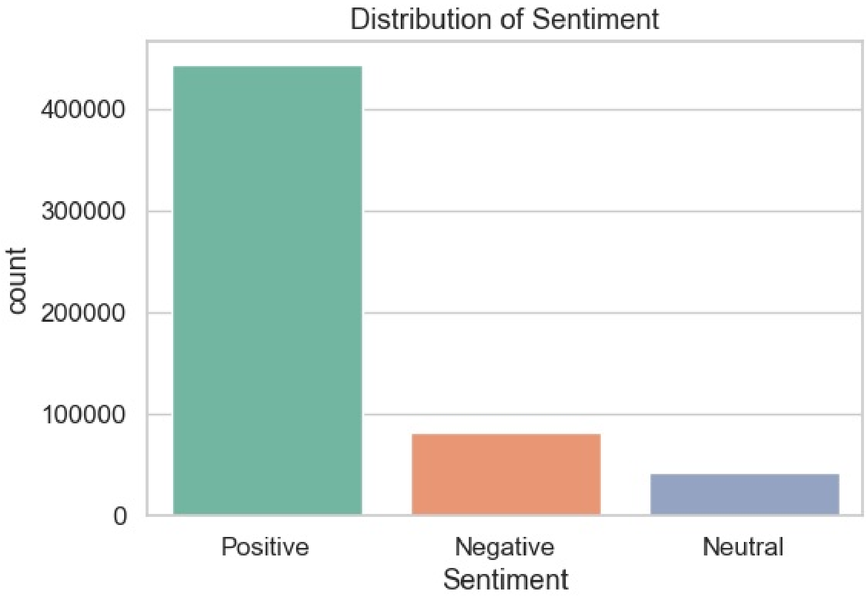
```
C:\Users\Vedant\AppData\Local\Temp\ipykernel_2548\1139816463.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.countplot(x="Score", data=df, palette="coolwarm")
```



In [65]:

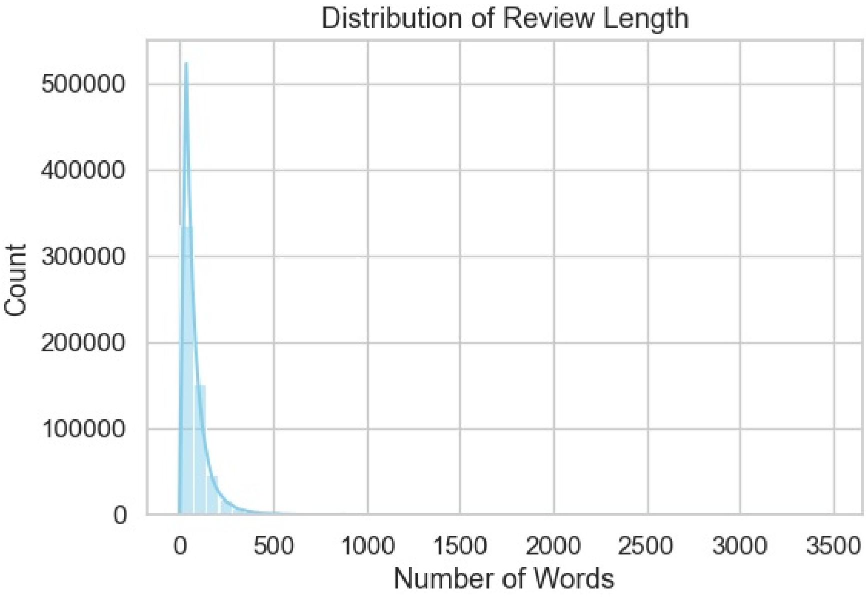
```
plt.figure(figsize=(6,4))
sns.countplot(x="Sentiment", data=df, palette="Set2")
plt.title("Distribution of Sentiment")
plt.show()
```

```
C:\Users\Vedant\AppData\Local\Temp\ipykernel_2548\952997905.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.countplot(x="Sentiment", data=df, palette="Set2")
```



In [66]:

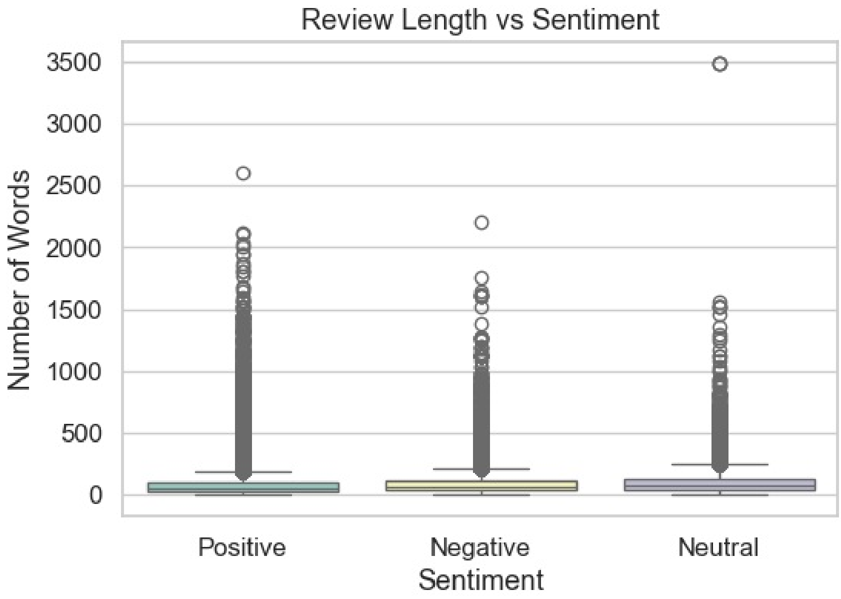
```
plt.figure(figsize=(6,4))
sns.histplot(df["review_length"], bins=50, kde=True, color="skyblue")
plt.title("Distribution of Review Length")
plt.xlabel("Number of Words")
plt.show()
```



In [67]:

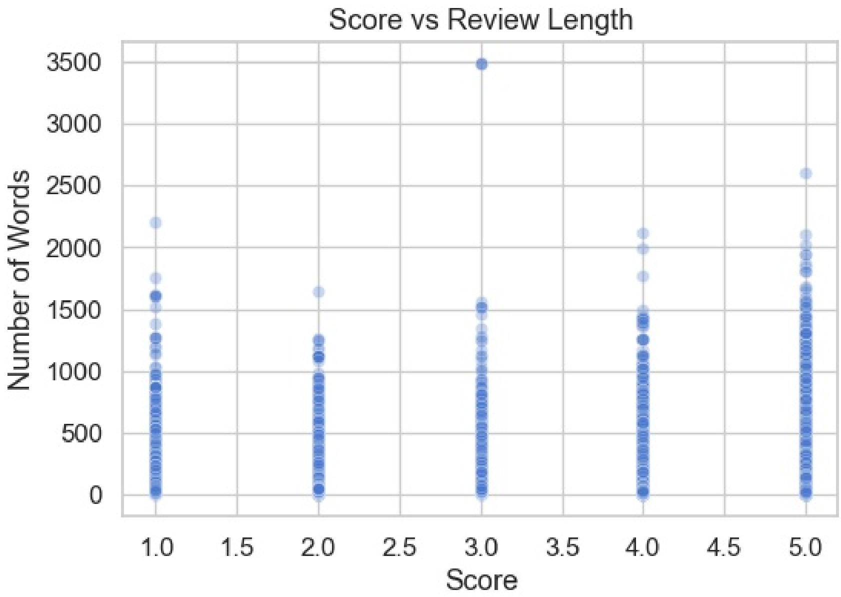
```
plt.figure(figsize=(6,4))
sns.boxplot(x="Sentiment", y="review_length", data=df, palette="Set3")
plt.title("Review Length vs Sentiment")
plt.ylabel("Number of Words")
plt.show()
```

```
C:\Users\Vedant\AppData\Local\Temp\ipykernel_2548\2116653056.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.boxplot(x="Sentiment", y="review_length", data=df, palette="Set3")
```



In [68]:

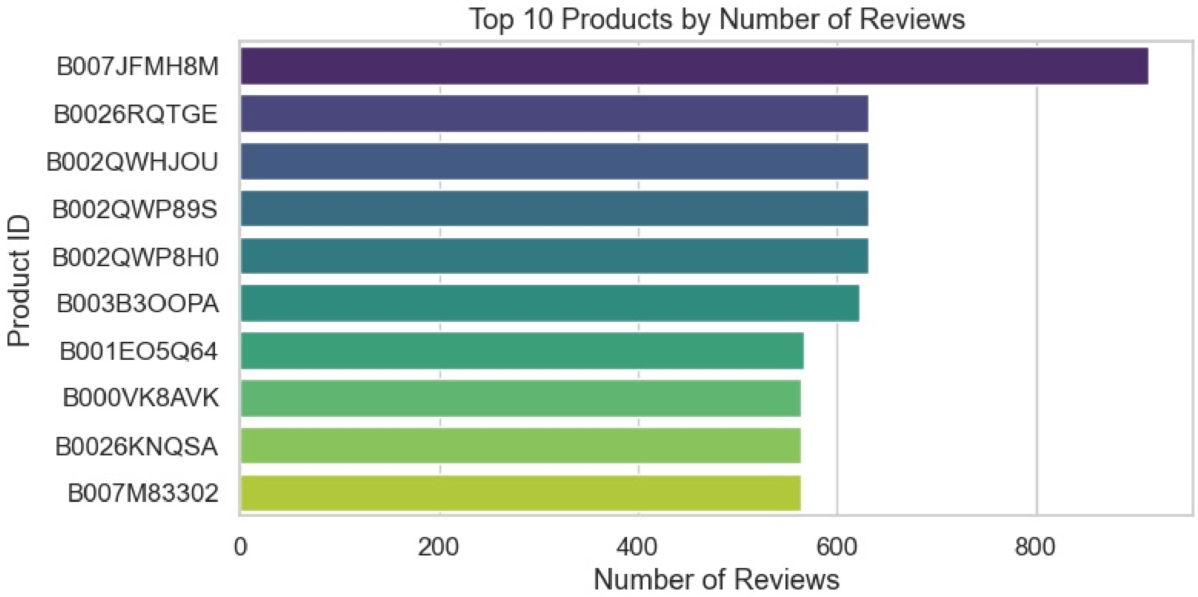
```
plt.figure(figsize=(6,4))
sns.scatterplot(x="Score", y="review_length", data=df, alpha=0.3)
plt.title("Score vs Review Length")
plt.ylabel("Number of Words")
plt.show()
```



In [69]:

```
top_products = df["ProductId"].value_counts().head(10)

plt.figure(figsize=(8,4))
sns.barplot(x=top_products.values, y=top_products.index, palette="viridis")
plt.xlabel("Number of Reviews")
plt.ylabel("Product ID")
plt.title("Top 10 Products by Number of Reviews")
plt.show()
```

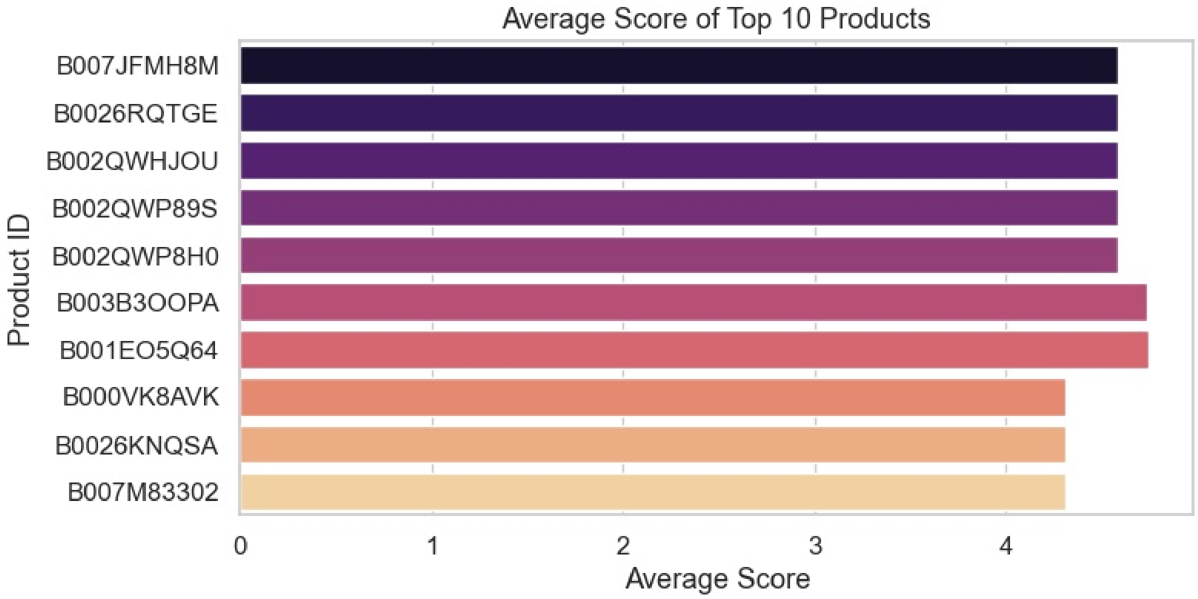
```
C:\Users\Vedant\AppData\Local\Temp\ipykernel_2548\3143326071.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_products.values, y=top_products.index, palette="viridis")
```



In [70]:

```
avg_score_products = df.groupby("ProductId")["Score"].mean().loc[top_products.index]

plt.figure(figsize=(8,4))
sns.barplot(x=avg_score_products.values, y=avg_score_products.index, palette="magma")
plt.xlabel("Average Score")
plt.ylabel("Product ID")
plt.title("Average Score of Top 10 Products")
plt.show()
```

```
C:\Users\Vedant\AppData\Local\Temp\ipykernel_2548\2800932052.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=avg_score_products.values, y=avg_score_products.index, palette="magma")
```



In [71]:

```
# Clean text: lowercase, remove HTML tags, special characters, numbers, extra spaces
def clean_text(text):
    text = str(text).lower()
    # remove HTML tags like <br />
    text = re.sub(r"<.*?>", " ", text)
    # remove special chars and numbers
    text = re.sub(r"[^a-z\s]", " ", text)
    # remove extra spaces
    text = re.sub(r"\s+", " ", text).strip()
    return text

df["clean_text"] = df["Text"].apply(clean_text)

# Review length (number of words)
df["review_length"] = df["clean_text"].apply(lambda x: len(x.split()))
```

```
df.head()
```

Out[71]:

	Text	Score	Time	Summary	ProductId	UserId	clean_text	review_length	Sentiment
0	I have bought several of the Vitality canned d...	5	1303862400	Good Quality Dog Food	B001E4KFG0	A3SGXH7AUHU8GW	i have bought several of the vitality canned d...	48	Positive
1	Product arrived labeled as Jumbo Salted Peanut...	1	1346976000	Not as Advertised	B00813GRG4	A1D87F6ZCVE5NK	product arrived labeled as jumbo salted peanut...	32	Negative
2	This is a confection that has been around a fe...	4	1219017600	"Delight" says it all	B000LQOCH0	ABXLMWJIXXAIN	this is a confection that has been around a fe...	93	Positive
3	If you are looking for the secret ingredient i...	2	1307923200	Cough Medicine	B000UA0QIQ	A395BORC6FGVXV	if you are looking for the secret ingredient i...	41	Negative
4	Great taffy at a great price. There was a wid...	5	1350777600	Great taffy	B006K2ZZ7K	A1UQRSCLF8GW1T	great taffy at a great price there was a wide ...	27	Positive

In [72]:

```
# Remove stopwords and single letters
def remove_stopwords_and_shortwords(text):
    words = text.split()
    words = [w for w in words if w not in ENGLISH_STOP_WORDS and len(w) > 2]
    return words

df["tokens"] = df["clean_text"].apply(remove_stopwords_and_shortwords)

# Flatten list of all tokens
all_tokens = [word for tokens in df["tokens"] for word in tokens]

# Count frequency
word_counts = Counter(all_tokens)

# Top 20 words
most_common_words = word_counts.most_common(20)
most_common_words
```

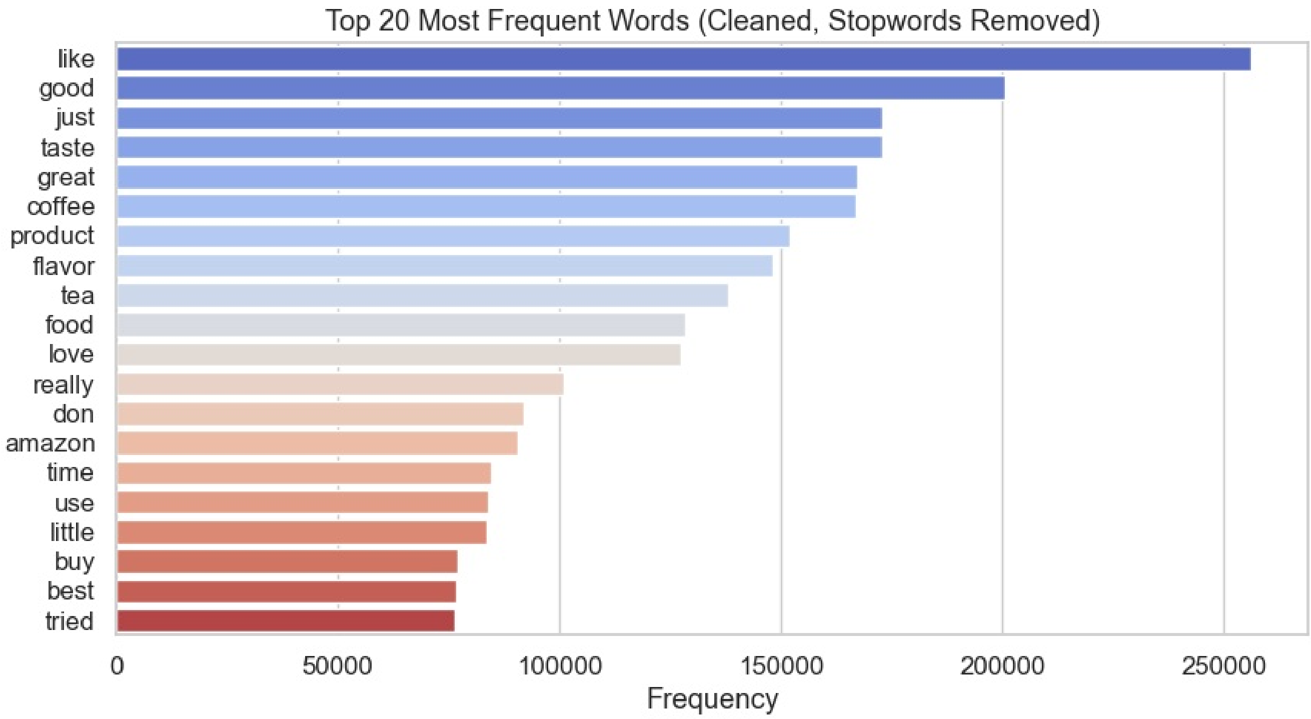
Out[72]:

```
[('like', 256215),
 ('good', 200644),
 ('just', 172973),
 ('taste', 172831),
 ('great', 167175),
 ('coffee', 166784),
 ('product', 151884),
 ('flavor', 148028),
 ('tea', 138153),
 ('food', 128525),
 ('love', 127520),
 ('really', 101076),
 ('don', 91874),
 ('amazon', 90504),
 ('time', 84769),
 ('use', 83908),
 ('little', 83499),
 ('buy', 76916),
 ('best', 76837),
 ('tried', 76486)]
```

In [73]:

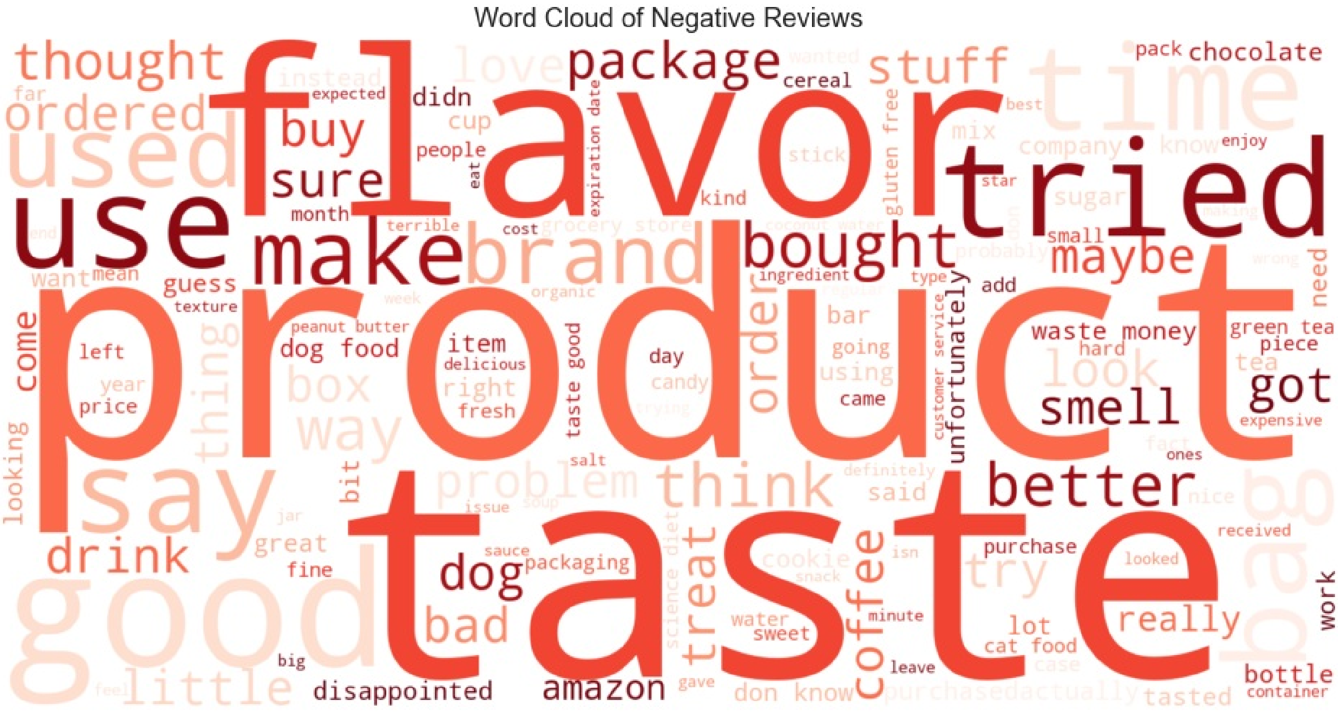
```
words, counts = zip(*most_common_words)

plt.figure(figsize=(10,5))
sns.barplot(x=list(counts), y=list(words), palette="coolwarm")
plt.title("Top 20 Most Frequent Words (Cleaned, Stopwords Removed)")
plt.xlabel("Frequency")
plt.show()
```

```
C:\Users\Vedant\AppData\Local\Temp\ipykernel_2548\3736262562.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=list(counts), y=list(words), palette="coolwarm")
```



In [74]:

```python
# Negative Reviews Word Cloud
neg_tokens = [w for tokens in df[df["Sentiment"]=="Negative"]["tokens"] for w in tokens]
neg_text = " ".join(neg_tokens)

wc_neg = WordCloud(width=1600, height=800, background_color="white",
                   max_words=150, colormap="Reds").generate(neg_text)

plt.figure(figsize=(12,6))
plt.imshow(wc_neg, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud of Negative Reviews")
plt.show()
```



In [75]:

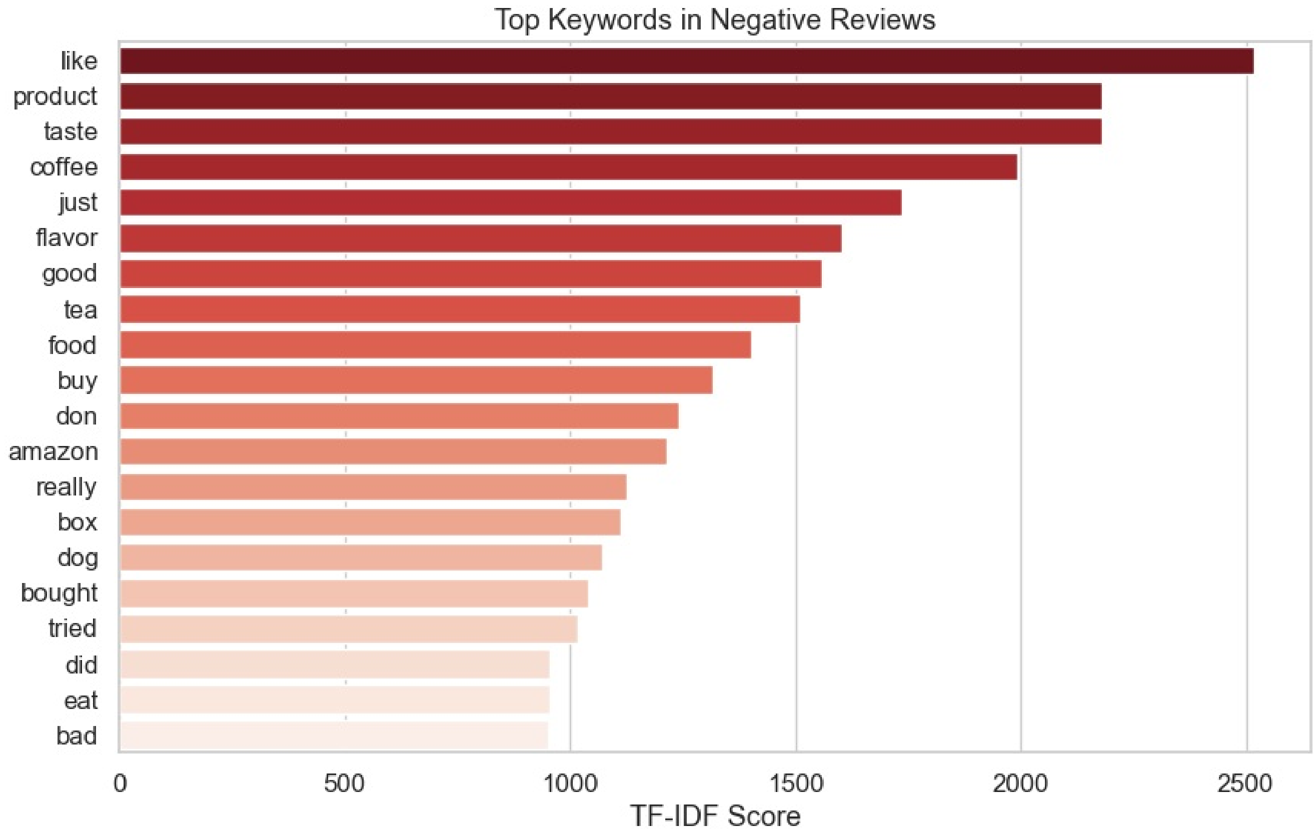
```python
from sklearn.feature_extraction.text import TfidfVectorizer

# Function to get top n keywords for a set of documents
def get_top_keywords(corpus, n=20):
    vectorizer = TfidfVectorizer(max_features=5000, stop_words='english', ngram_range=(1,2))
    X = vectorizer.fit_transform(corpus)
    feature_names = np.array(vectorizer.get_feature_names_out())
    # Sum TF-IDF scores across all documents
    scores = X.sum(axis=0).A1
    top_indices = scores.argsort()[-n:][::-1]
    top_features = feature_names[top_indices]
    top_scores = scores[top_indices]
    return list(zip(top_features, top_scores))

# Prepare text by joining tokens back
df["processed_text"] = df["tokens"].apply(lambda x: " ".join(x))


negative_corpus = df[df["Sentiment"]=="Negative"]["processed_text"].tolist()


top_negative_keywords = get_top_keywords(negative_corpus, n=20)


print("\nTop Negative Keywords:\n", top_negative_keywords)
```

```
Top Negative Keywords:
 [('like', 2518.952694463791), ('product', 2179.958248342551), ('taste', 2179.891925612207), ('coffee', 1993.978
713269519), ('just', 1736.7172527856949), ('flavor', 1605.3815041938315), ('good', 1559.8365575581413), ('tea',
1510.4296645448028), ('food', 1401.423855418517), ('buy', 1315.9309286582904), ('don', 1242.202368882589), ('ama
zon', 1216.0404654025765), ('really', 1126.777196284686), ('box', 1111.5371631866772), ('dog', 1071.988871884558
9), ('bought', 1039.9068516424252), ('tried', 1015.2221749030888), ('did', 954.3797367764944), ('eat', 954.01499
79351871), ('bad', 951.6739031326804)]
```

In [76]:

```
def plot_top_keywords(keywords, title, color="blue"):
    words, scores = zip(*keywords)
    plt.figure(figsize=(10,6))
    sns.barplot(x=list(scores), y=list(words), palette=color)
    plt.xlabel("TF-IDF Score")
    plt.title(title)
    plt.show()


# Negative
plot_top_keywords(top_negative_keywords, "Top Keywords in Negative Reviews", color="Reds_r")
```

```
C:\Users\Vedant\AppData\Local\Temp\ipykernel_2548\2597827445.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=list(scores), y=list(words), palette=color)
```



In [ ]: