In [16]:
```
!pip install beautifulsoup4 pandas numpy seaborn matplotlib textblob requests wordcloud
```
```
Requirement already satisfied: beautifulsoup4 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (4.13.4)
Requirement already satisfied: pandas in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (2.3.2)
Requirement already satisfied: numpy in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (2.3.2)
Requirement already satisfied: seaborn in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (0.13.2)
Requirement already satisfied: matplotlib in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (3.10.6)
Requirement already satisfied: textblob in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (0.19.0)
Requirement already satisfied: requests in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (2.32.4)
Collecting wordcloud
  Downloading wordcloud-1.9.4-cp313-cp313-win_amd64.whl.metadata (3.5 kB)
Requirement already satisfied: soupsieve>1.2 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from beautifulsoup4) (2.7)
Requirement already satisfied: typing-extensions>=4.0.0 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from beautifulsoup4) (4.14.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (4.59.2)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from matplotlib) (3.2.3)
Requirement already satisfied: nltk>=3.9 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from textblob) (3.9.2)
Requirement already satisfied: charset_normalizer<4,>=2 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from requests) (3.4.3)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from requests) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from requests) (2025.8.3)
Requirement already satisfied: click in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from nltk>=3.9->textblob) (8.3.0)
Requirement already satisfied: joblib in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from nltk>=3.9->textblob) (1.5.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from nltk>=3.9->textblob) (2025.9.18)
Requirement already satisfied: tqdm in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from nltk>=3.9->textblob) (4.67.1)
Requirement already satisfied: six>=1.5 in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: colorama in c:\users\dell\appdata\local\programs\python\python313\lib\site-packages (from click->nltk>=3.9->textblob) (0.4.6)
Downloading wordcloud-1.9.4-cp313-cp313-win_amd64.whl (300 kB)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.4
```

In [4]:
```python
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from textblob import TextBlob
import requests

#create lists to store data within.

Names = []
Cities = []
Posted_on = []
Ratings = []
Reviews = []
Occasion = []
```

In [5]:
```python
#fetch the data from site(FlowerAura)

url = "https://www.floweraura.com/reviews/p/6617/10-red-roses-bouquet?page="

for i in range(1,51):
    cnp = url+str(i)
    url_new = cnp

    r = requests.get(url_new)

    soup = BeautifulSoup(r.text, "html.parser")
    soup


    main = soup.find("div",{"class":"review-left-container"})
    sub = main.find_all("div",{"class":"new-review-card-container"})

    for i in sub:
        name= i.find("span",{"class":"review-author-name"})
        Names.append(name.text.title())

        city= i.find_all("span",{"class":"review-meta-details"})
        Cities.append(city[0].text.title())

        try:
            Occasion.append(city[1].text.title())
        except:
            Occasion.append(np.nan)

        date = i.find_all("span")
        try:
            Posted_on.append(date[4].text)
        except:
            Posted_on.append(np.nan)

        rating = i.find("span",{"class":"star-count-container"})
        Ratings.append(rating.text)

        review = i.find_all("div")
        Reviews.append(review[-1].text)
```

In [6]:
```python
# Raw dataframe

df = pd.DataFrame({"Names":Names,"Cities":Cities,"Posted_on":Posted_on,"Occasion":Occasion,"Reviews":Reviews,"Ratings":Ratings})
df
```

Out[6]:

	Names	Cities	Posted_on	Occasion	Reviews	Ratings
0	Suraj Chaunal	Noida	Posted On : 23rd Sep 2025	Occassion : Birthday	the flowers were fresh and colorful. TBH they ...	5
1	Bakiyalakshmi	Bangalore	Posted On : 22nd Sep 2025	Occassion : Anniversary	Very fresh flowers, delivered on time with bea...	5
2	Pulak.Pal77	Kolkata	Posted On : 19th Sep 2025	Occassion : Birthday	Roses quality is very nice.	5
3	Inaya	Udaipur	Posted On : 31st Aug 2025	NaN	Very beautifull on tome delivered.	5
4	Nallapandiyan D	Coimbatore	Posted On : 11th Jul 2025	Occassion : Birthday	Thanks for the timely delivery.	4
...	...	...	...	...	...	...
495	Shweta	Goa	Posted On : 6th Sep 2023	Occassion : Love & Romance	Tnx... it was on time n perfect	5
496	Arun	Hyderabad	Posted On : 9th Sep 2023	NaN	Thanks for the flowers	4
497	Prernasaini	Gurgaon	Posted On : 8th Sep 2023	NaN	Thank you so much	5
498	Jhanvi Jaiswal	Mumbai	Posted On : 7th Sep 2023	NaN	Thankyou somuch for helping and making my surp...	5
499	Shashikala	Varanasi	Posted On : 5th Sep 2023	Occassion : Birthday	Loved it	5

500 rows × 6 columns

In [7]:
```python
#lets get dates as date's standard format:-

a = "Posted On : 31st Aug 2025"
x = a.index(":")
print(a[x+2:])
```
```
31st Aug 2025
```

In [8]:
```python
#create function which will extract all dates & occasions within the DF in there standard format:-
def extract(value):
    try:
        x = value.index(":")
        return value[x+2:]
    except:
        return np.nan
df["Posted_on"]= df["Posted_on"].apply(extract)
df["Occasion"]= df["Occasion"].apply(extract)
df
```

Out[8]:

	Names	Cities	Posted_on	Occasion	Reviews	Ratings
0	Suraj Chaunal	Noida	23rd Sep 2025	Birthday	the flowers were fresh and colorful. TBH they ...	5
1	Bakiyalakshmi	Bangalore	22nd Sep 2025	Anniversary	Very fresh flowers, delivered on time with bea...	5
2	Pulak.Pal77	Kolkata	19th Sep 2025	Birthday	Roses quality is very nice.	5
3	Inaya	Udaipur	31st Aug 2025	NaN	Very beautifull on tome delivered.	5
4	Nallapandiyan D	Coimbatore	11th Jul 2025	Birthday	Thanks for the timely delivery.	4
...	...	...	...	...	...	...
495	Shweta	Goa	6th Sep 2023	Love & Romance	Tnx... it was on time n perfect	5
496	Arun	Hyderabad	9th Sep 2023	NaN	Thanks for the flowers	4
497	Prernasaini	Gurgaon	8th Sep 2023	NaN	Thank you so much	5
498	Jhanvi Jaiswal	Mumbai	7th Sep 2023	NaN	Thankyou somuch for helping and making my surp...	5
499	Shashikala	Varanasi	5th Sep 2023	Birthday	Loved it	5

500 rows × 6 columns

In [9]:
```python
#removing all suffix:-

rep = ["th","st","rd","nd"]
for i in rep:
    df["Posted_on"]= df["Posted_on"].str.replace(i,"")
df
```

Out[9]:

	Names	Cities	Posted_on	Occasion	Reviews	Ratings
0	Suraj Chaunal	Noida	23 Sep 2025	Birthday	the flowers were fresh and colorful. TBH they ...	5
1	Bakiyalakshmi	Bangalore	22 Sep 2025	Anniversary	Very fresh flowers, delivered on time with bea...	5
2	Pulak.Pal77	Kolkata	19 Sep 2025	Birthday	Roses quality is very nice.	5
3	Inaya	Udaipur	31 Aug 2025	NaN	Very beautifull on tome delivered.	5
4	Nallapandiyan D	Coimbatore	11 Jul 2025	Birthday	Thanks for the timely delivery.	4
...	...	...	...	...	...	...
495	Shweta	Goa	6 Sep 2023	Love & Romance	Tnx... it was on time n perfect	5
496	Arun	Hyderabad	9 Sep 2023	NaN	Thanks for the flowers	4
497	Prernasaini	Gurgaon	8 Sep 2023	NaN	Thank you so much	5
498	Jhanvi Jaiswal	Mumbai	7 Sep 2023	NaN	Thankyou somuch for helping and making my surp...	5
499	Shashikala	Varanasi	5 Sep 2023	Birthday	Loved it	5

500 rows × 6 columns

In [10]:
```python
df.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Names      500 non-null    object
 1   Cities     500 non-null    object
 2   Posted_on  500 non-null    object
 3   Occasion   388 non-null    object
 4   Reviews    500 non-null    object
 5   Ratings    500 non-null    object
dtypes: object(6)
memory usage: 23.6+ KB
```

In [11]:
```python
df["Posted_on"] = pd.to_datetime(df["Posted_on"])
df["Ratings"] = df["Ratings"].astype("float")
df["Polarity"] = [TextBlob(i).sentiment.polarity for i in df["Reviews"]]
df["Subjectivity"] = [TextBlob(i).subjectivity for i in df["Reviews"]]
df
```

Out[11]:

	Names	Cities	Posted_on	Occasion	Reviews	Ratings	Polarity	Subjectivity
0	Suraj Chaunal	Noida	2025-09-23	Birthday	the flowers were fresh and colorful. TBH they ...	5.0	0.262500	0.400000
1	Bakiyalakshmi	Bangalore	2025-09-22	Anniversary	Very fresh flowers, delivered on time with bea...	5.0	0.532381	0.740476
2	Pulak.Pal77	Kolkata	2025-09-19	Birthday	Roses quality is very nice.	5.0	0.780000	1.000000
3	Inaya	Udaipur	2025-08-31	NaN	Very beautifull on tome delivered.	5.0	0.200000	0.300000
4	Nallapandiyan D	Coimbatore	2025-07-11	Birthday	Thanks for the timely delivery.	4.0	0.200000	0.200000
...	...	...	...	...	...	...	...	...
495	Shweta	Goa	2023-09-06	Love & Romance	Tnx... it was on time n perfect	5.0	1.000000	1.000000
496	Arun	Hyderabad	2023-09-09	NaN	Thanks for the flowers	4.0	0.200000	0.200000
497	Prernasaini	Gurgaon	2023-09-08	NaN	Thank you so much	5.0	0.200000	0.200000
498	Jhanvi Jaiswal	Mumbai	2023-09-07	NaN	Thankyou somuch for helping and making my surp...	5.0	0.408333	0.758333
499	Shashikala	Varanasi	2023-09-05	Birthday	Loved it	5.0	0.700000	0.800000

500 rows × 8 columns

In [12]:
```python
#adding Polarity to know how it has been performing:-

p = df["Polarity"].mean()
if p <= 0:
    print("Negative")
else:
    print("Positive")
```
```
Positive
```

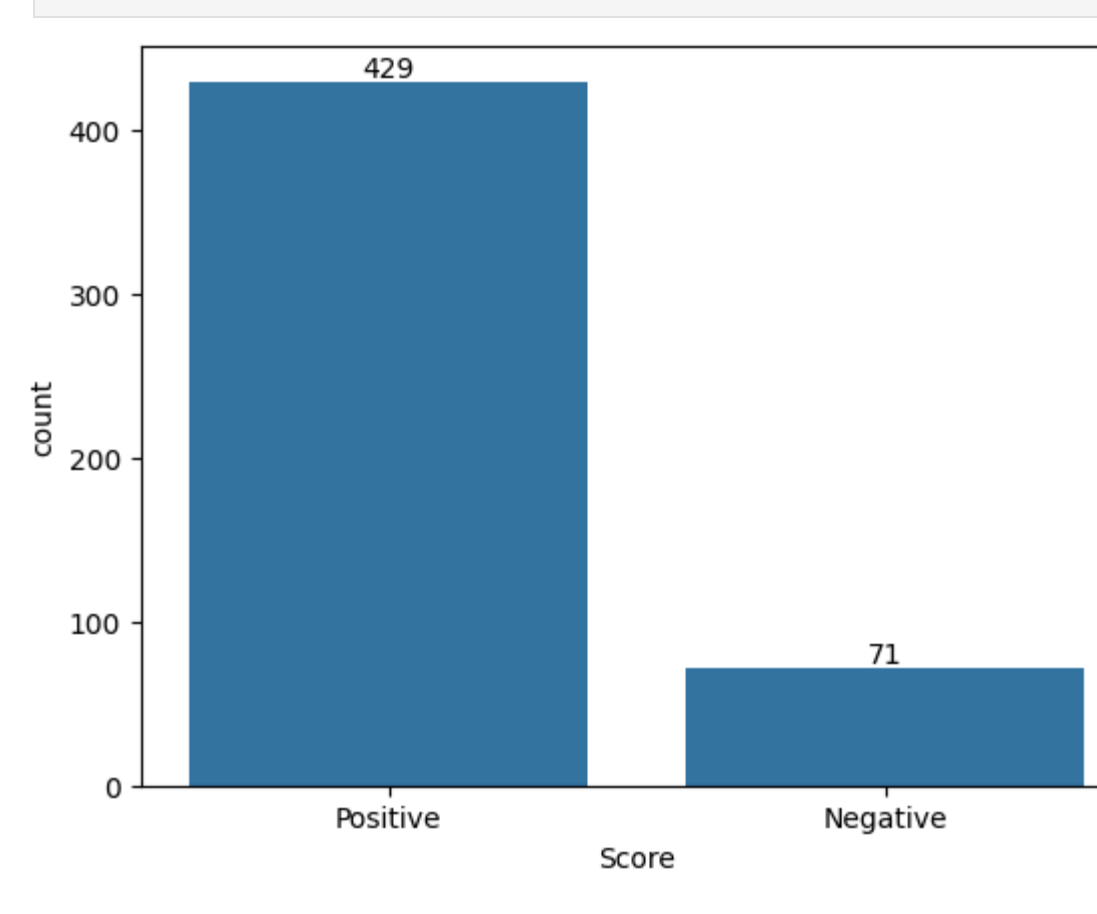
In [13]:
```python
def score(value):
    if value <= 0:
        return"Negative"
    else:
        return"Positive"
df["Score"]= df["Polarity"].apply(score)
df
```

Out[13]:

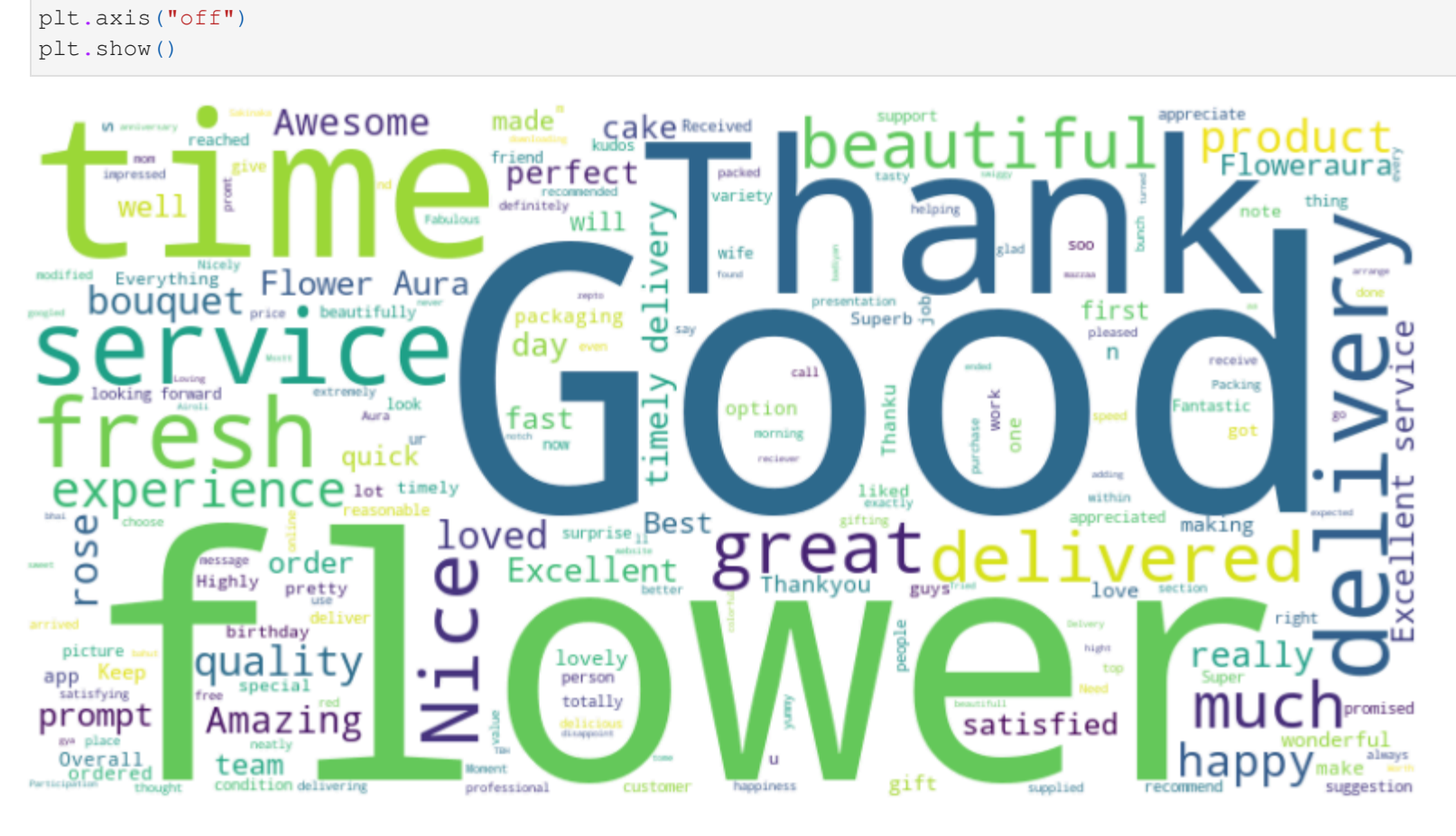
	Names	Cities	Posted_on	Occasion	Reviews	Ratings	Polarity	Subjectivity	Score
0	Suraj Chaunal	Noida	2025-09-23	Birthday	the flowers were fresh and colorful. TBH they ...	5.0	0.262500	0.400000	Positive
1	Bakiyalakshmi	Bangalore	2025-09-22	Anniversary	Very fresh flowers, delivered on time with bea...	5.0	0.532381	0.740476	Positive
2	Pulak.Pal77	Kolkata	2025-09-19	Birthday	Roses quality is very nice.	5.0	0.780000	1.000000	Positive
3	Inaya	Udaipur	2025-08-31	NaN	Very beautifull on tome delivered.	5.0	0.200000	0.300000	Positive
4	Nallapandiyan D	Coimbatore	2025-07-11	Birthday	Thanks for the timely delivery.	4.0	0.200000	0.200000	Positive
...	...	...	...	...	...	...	...	...	...
495	Shweta	Goa	2023-09-06	Love & Romance	Tnx... it was on time n perfect	5.0	1.000000	1.000000	Positive
496	Arun	Hyderabad	2023-09-09	NaN	Thanks for the flowers	4.0	0.200000	0.200000	Positive
497	Prernasaini	Gurgaon	2023-09-08	NaN	Thank you so much	5.0	0.200000	0.200000	Positive
498	Jhanvi Jaiswal	Mumbai	2023-09-07	NaN	Thankyou somuch for helping and making my surp...	5.0	0.408333	0.758333	Positive
499	Shashikala	Varanasi	2023-09-05	Birthday	Loved it	5.0	0.700000	0.800000	Positive

500 rows × 9 columns

In [14]:
```python
#adding barplot to get the idea of positive and negative reviews:-

ax = sns.countplot(x=df["Score"], data = df)

ax.bar_label(container = ax.containers[0])
plt.show()
```



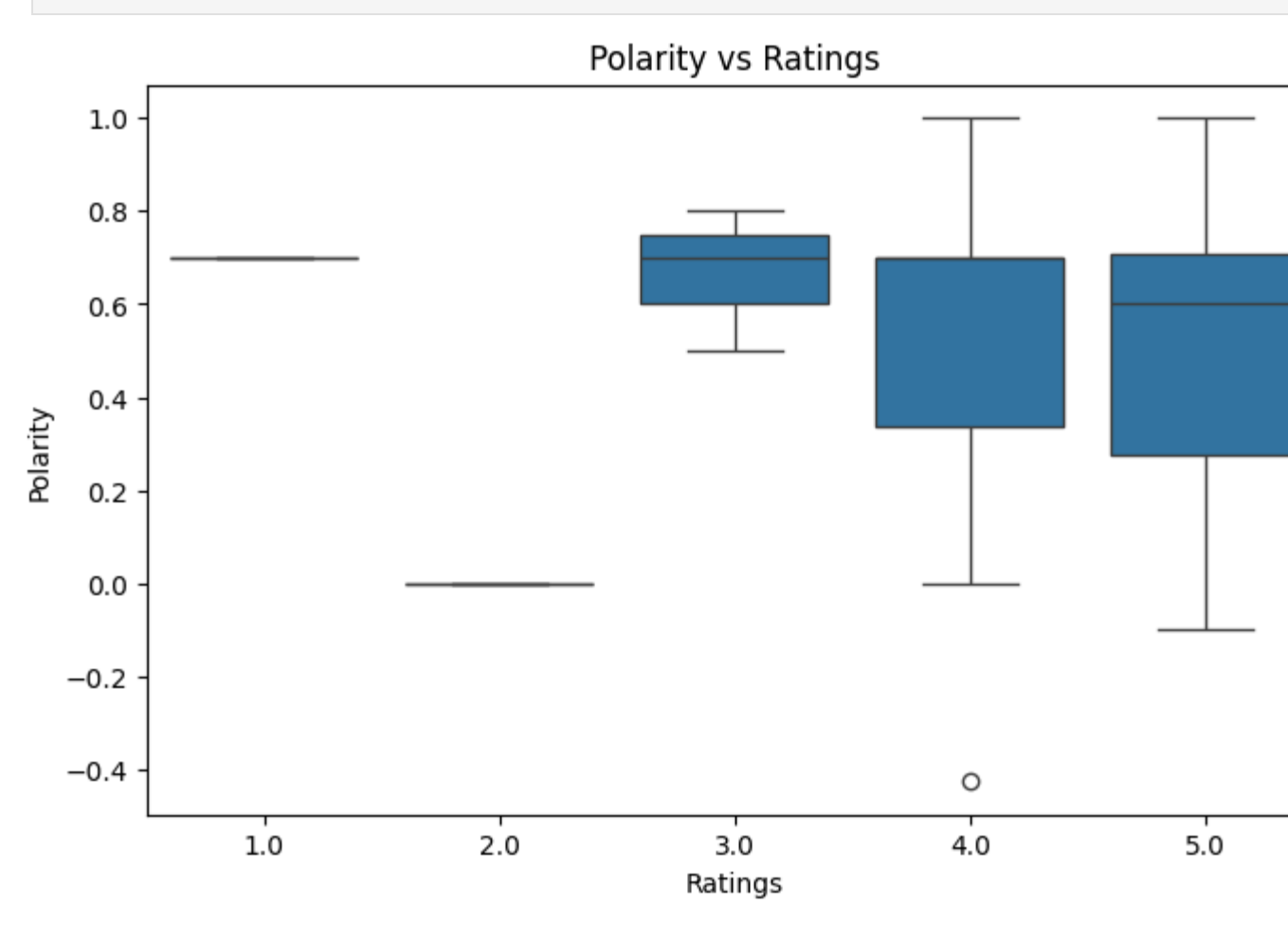
In [17]:
```python
from wordcloud import WordCloud , STOPWORDS
import matplotlib.pyplot as plt

#Two DataFrame created for postive and negative wordcloud

df_pos = df.loc[(df["Score"]=="Positive")]
df_neg = df.loc[(df["Score"]=="Negative")]

# For Positive wordcloud

text = " ".join(text for text in df_pos["Reviews"])
wordcloud = WordCloud(width=800, height=400,
                      background_color='white').generate(text)

plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```



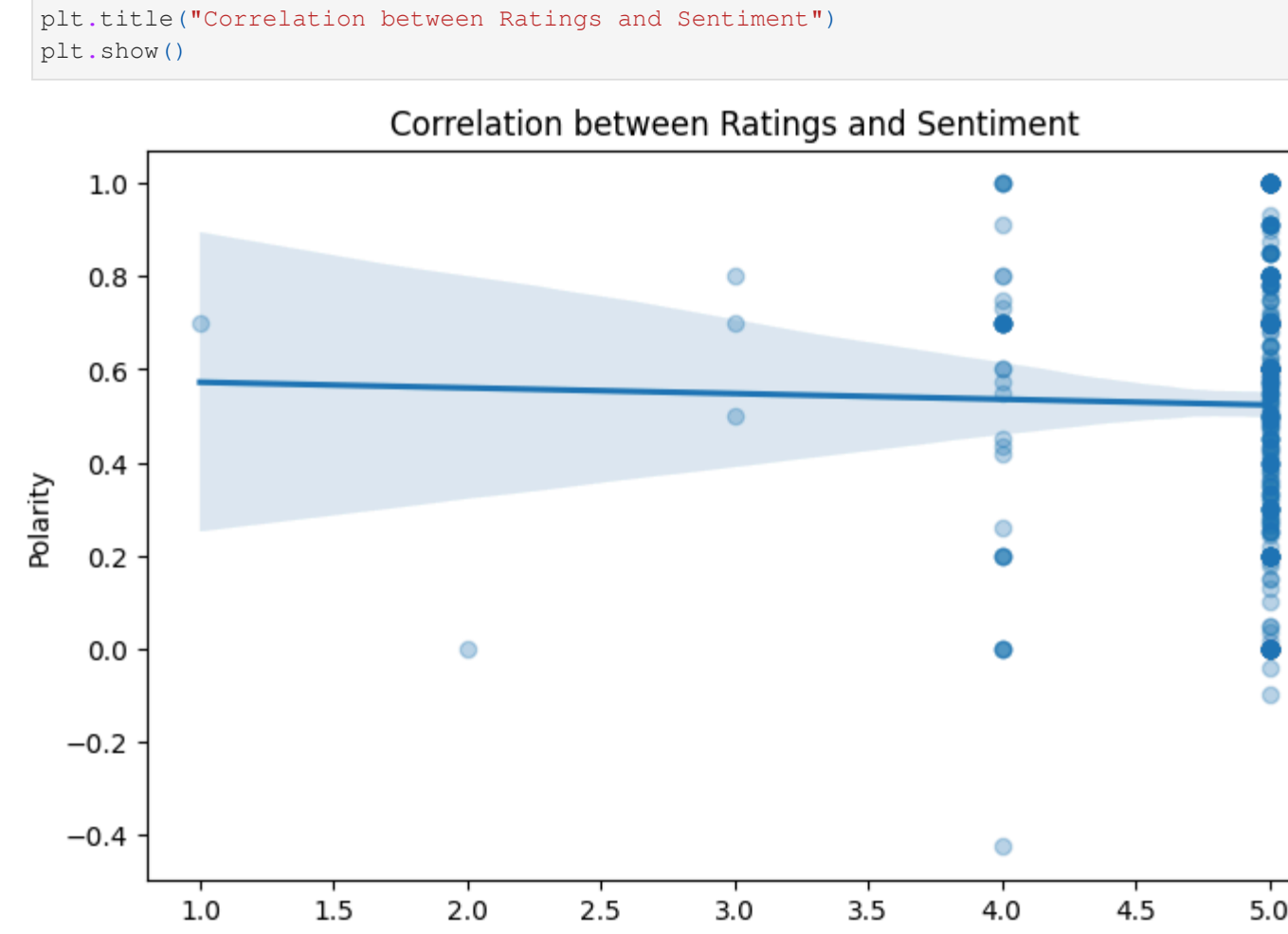
In [18]:
```python
#For negative wordcloud


text = " ".join(text for text in df_neg["Reviews"])
wordcloud = WordCloud(width=800, height=400,
                      background_color='white').generate(text)

plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

In [19]:
```python
# finding correlation between ratings and polarity:-

correlation = df["Ratings"].corr(df["Polarity"])
print("Correlation between Ratings and Polarity:", correlation)
```
```
Correlation between Ratings and Polarity: -0.014541348587567443
```

In [20]:
```python
plt.figure(figsize=(8,5))
sns.boxplot(x="Ratings", y="Polarity", data=df)
plt.title("Polarity vs Ratings")
plt.show()
```



In [21]:
```python
plt.figure(figsize=(8,5))
sns.regplot(x="Ratings", y="Polarity", data=df, scatter_kws={"alpha":0.3})
plt.title("Correlation between Ratings and Sentiment")
plt.show()
```

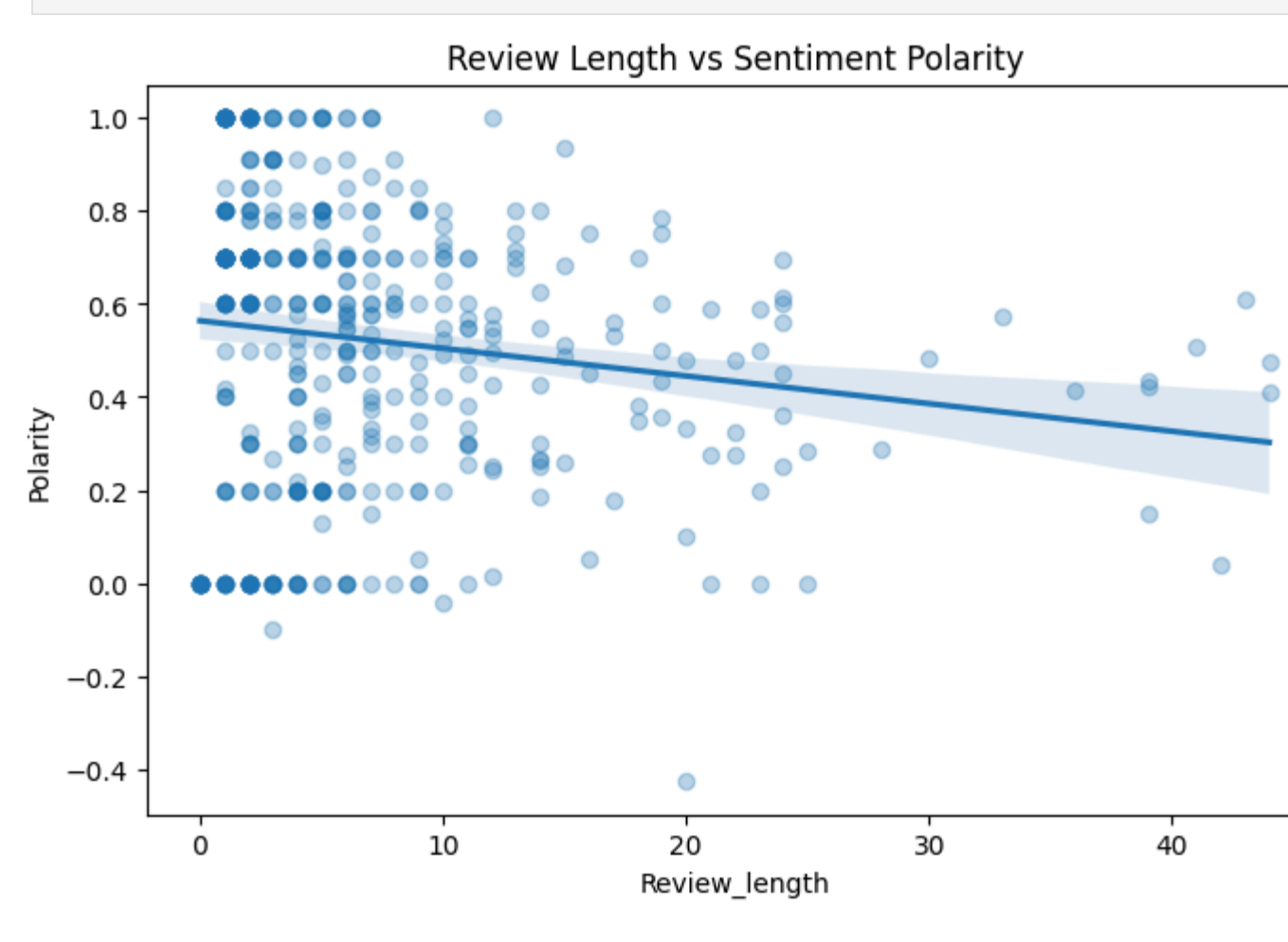


In [22]:
```python
#calculatig review length written by customers whether it is Positive or negative:-

df["Review_length"] = df["Reviews"].apply(lambda x: len(str(x).split()))  # word count
df["Review_length"]
```

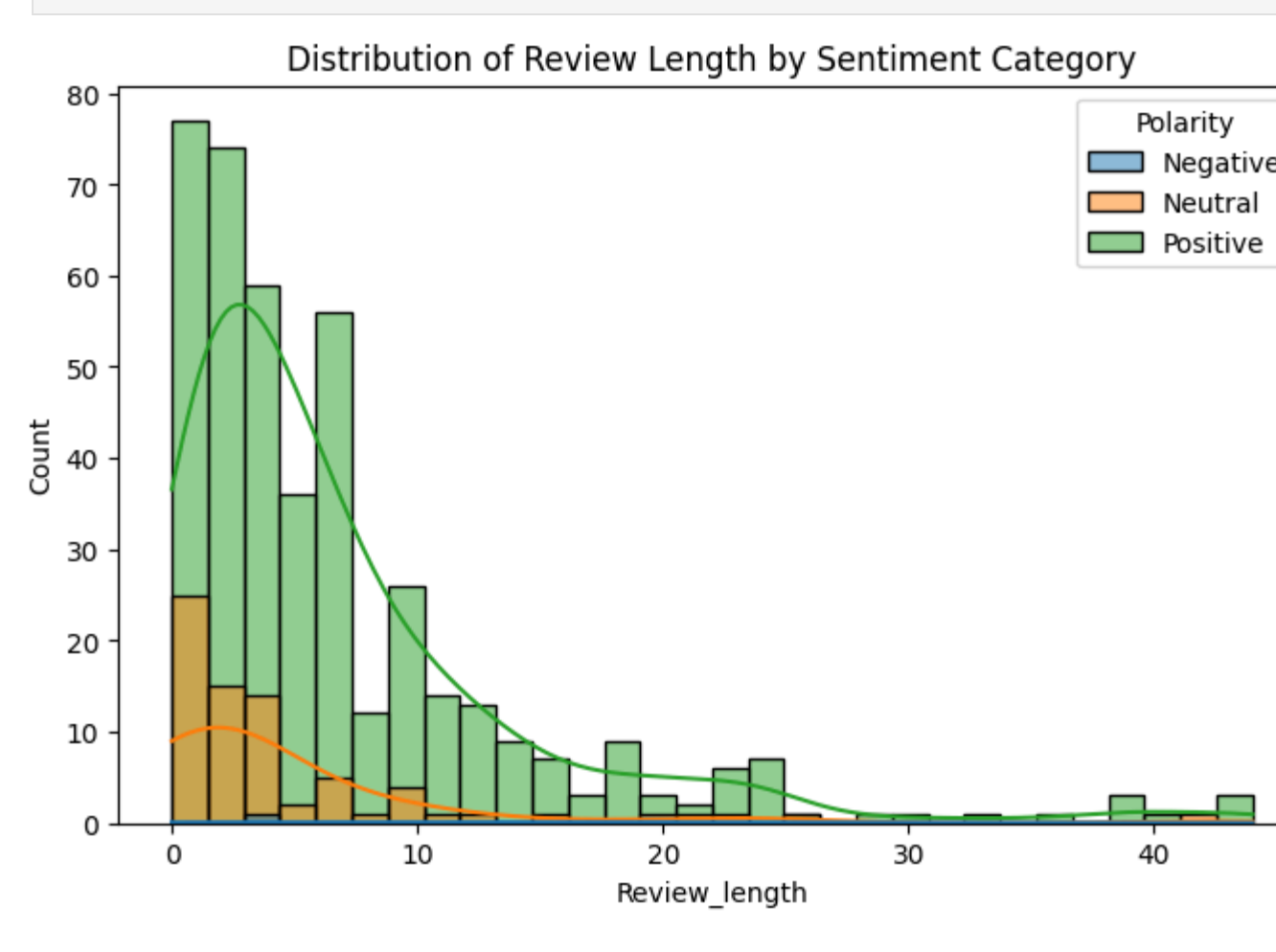
Out[22]:
```
0      14
1      17
2       5
3       5
4       5
       ..
495     7
496     4
497     4
498    44
499     2
Name: Review_length, Length: 500, dtype: int64
```

In [23]:
```python
corr = df["Review_length"].corr(df["Polarity"])
print("Correlation between Review Length and Sentiment Polarity:", corr)
```
```
Correlation between Review Length and Sentiment Polarity: -0.14094743322237901
```

In [24]:
```python
plt.figure(figsize=(8,5))
sns.regplot(x="Review_length", y="Polarity", data=df, scatter_kws={"alpha":0.3})
plt.title("Review Length vs Sentiment Polarity")
plt.show()
```



In [25]:
```python
plt.figure(figsize=(8,5))
sns.histplot(data=df, x="Review_length", hue=pd.cut(df["Polarity"], bins=[-1,-0.1,0.1,1], labels=["Negative","Neutral","Positive"]), kde=True)
plt.title("Distribution of Review Length by Sentiment Category")
plt.show()
```



In [ ]:
```
🌹✨
#Sentiment Analysis Report – 10 Red Roses Bouquet (FlowerAura)

📊 1. Data Overview

We collected and cleaned customer reviews (handled missing values, standardized text).
Additionally, we engineered two features:

🧭 Sentiment Polarity → captures positivity/negativity strength

✍️ Review Length → helps analyze expression patterns in customer opinions

💡 2. Sentiment Analysis Results

📈 Review Distribution → Majority of reviews were positive, but a fair share of neutral/negative ones exist.

⭐ Ratings vs Sentiment → 4–5★ ratings aligned strongly with positive sentiments, while 1–2★ ratings leaned negative.

📊 Average Sentiment per Rating → Clear upward trend → higher rating = more positive sentiment.

📝 Review Length → Longer reviews reflected stronger emotions (either highly satisfied 🌟 or quite unhappy 😠).

🌼 3. Key Insights

✅ What Customers Loved

Freshness 🌿

Fragrance 🌸

Elegant packaging 🎁

Quick & smooth delivery 🚚💨

Beautiful presentation 💐

❌ Common Complaints

Late deliveries ⏳

Inconsistent flower quality ⚖️

Slightly higher pricing 💰

🔎 Patterns Observed

Happy customers → write longer, descriptive reviews 🖊️

Unhappy customers → leave short, sharp complaints ⚡

🚀 4. Recommendations

🕒 Fix Delivery Issues → Streamline logistics to ensure timely deliveries.

🌺 Maintain Quality → Consistency in freshness & flower quality will reduce negatives.

📣 Leverage Strengths → Highlight fragrance, freshness & premium packaging in ads/social media.

📷 Engage Customers → Encourage satisfied buyers to share reviews & photos online for organic promotion.

💎 Price Positioning → If priced higher, market the bouquet as a premium, luxury experience.


👉 In a nutshell: Customers adore the freshness, fragrance & presentation of the Red Roses 🌹,
   but addressing delivery speed & quality consistency will elevate FlowerAura's bouquet into an undisputed favorite ✨💖.
```