In [1]:
```
#Q.2) Explore various variable and row filters in Python for cleaning data. Apply various plot features in Pythor
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv("large_email_dataset_with_cc_bcc.csv")
df.head()
```

Out[1]:

	Date	From	To	Subject	Body	Cc	Bcc
0	2025-01-05 03:16:00	sales@example.com	me@example.com	Project Update	Invoice is due next week.	supervisor@example.com	audit@example.com
1	2025-02-24 07:38:00	support@example.com	me@example.com	Invoice Reminder	Invoice is due next week.	team@example.com	NaN
2	2025-02-02 07:43:00	client@example.com	me@example.com	Team Outing	Review the latest changes in the codebase.	manager@example.com	NaN
3	2025-02-08 14:22:00	admin@example.com	me@example.com	New Joiner Introduction	Review the latest changes in the codebase.	NaN	NaN
4	2025-03-02 01:37:00	client@example.com	me@example.com	Performance Feedback	Reminder: Submit your deliverables.	supervisor@example.com	admin@example.com

In [2]:
```
df_filtered_cols = df[['Date', 'From', 'Subject', 'Body']]
df_filtered_cols.head()
```

Out[2]:

	Date	From	Subject	Body
0	2025-01-05 03:16:00	sales@example.com	Project Update	Invoice is due next week.
1	2025-02-24 07:38:00	support@example.com	Invoice Reminder	Invoice is due next week.
2	2025-02-02 07:43:00	client@example.com	Team Outing	Review the latest changes in the codebase.
3	2025-02-08 14:22:00	admin@example.com	New Joiner Introduction	Review the latest changes in the codebase.
4	2025-03-02 01:37:00	client@example.com	Performance Feedback	Reminder: Submit your deliverables.

In [9]:
```
df_dropped = df.drop(columns=['Cc', 'Bcc'])
df.head()
```

Out[9]:

	Date	Sender	Receiver	Subject	Body	Cc	Bcc
0	2025-01-05 03:16:00	sales@example.com	me@example.com	Project Update	Invoice is due next week.	supervisor@example.com	audit@example.com
1	2025-02-24 07:38:00	support@example.com	me@example.com	Invoice Reminder	Invoice is due next week.	team@example.com	NaN
2	2025-02-02 07:43:00	client@example.com	me@example.com	Team Outing	Review the latest changes in the codebase.	manager@example.com	NaN
3	2025-02-08 14:22:00	admin@example.com	me@example.com	New Joiner Introduction	Review the latest changes in the codebase.	NaN	NaN
4	2025-03-02 01:37:00	client@example.com	me@example.com	Performance Feedback	Reminder: Submit your deliverables.	supervisor@example.com	admin@example.com

In [10]:
```
df.rename(columns={'From': 'Sender', 'To': 'Receiver'}, inplace=True)
df.head()
```

Out[10]:

	Date	Sender	Receiver	Subject	Body	Cc	Bcc
0	2025-01-05 03:16:00	sales@example.com	me@example.com	Project Update	Invoice is due next week.	supervisor@example.com	audit@example.com
1	2025-02-24 07:38:00	support@example.com	me@example.com	Invoice Reminder	Invoice is due next week.	team@example.com	NaN
2	2025-02-02 07:43:00	client@example.com	me@example.com	Team Outing	Review the latest changes in the codebase.	manager@example.com	NaN
3	2025-02-08 14:22:00	admin@example.com	me@example.com	New Joiner Introduction	Review the latest changes in the codebase.	NaN	NaN
4	2025-03-02 01:37:00	client@example.com	me@example.com	Performance Feedback	Reminder: Submit your deliverables.	supervisor@example.com	admin@example.com

In [11]:
```
# Emails sent by admin
admin_emails = df[df['Sender'] == 'admin@example.com']
df.head()
```

Out[11]:

	Date	Sender	Receiver	Subject	Body	Cc	Bcc
0	2025-01-05 03:16:00	sales@example.com	me@example.com	Project Update	Invoice is due next week.	supervisor@example.com	audit@example.com
1	2025-02-24 07:38:00	support@example.com	me@example.com	Invoice Reminder	Invoice is due next week.	team@example.com	NaN
2	2025-02-02 07:43:00	client@example.com	me@example.com	Team Outing	Review the latest changes in the codebase.	manager@example.com	NaN
3	2025-02-08 14:22:00	admin@example.com	me@example.com	New Joiner Introduction	Review the latest changes in the codebase.	NaN	NaN
4	2025-03-02 01:37:00	client@example.com	me@example.com	Performance Feedback	Reminder: Submit your deliverables.	supervisor@example.com	admin@example.com

In [12]:
```
invoice_emails = df[df['Subject'].str.contains("Invoice", case=False)]
df.head()
```

Out[12]:

	Date	Sender	Receiver	Subject	Body	Cc	Bcc
0	2025-01-05 03:16:00	sales@example.com	me@example.com	Project Update	Invoice is due next week.	supervisor@example.com	audit@example.com
1	2025-02-24 07:38:00	support@example.com	me@example.com	Invoice Reminder	Invoice is due next week.	team@example.com	NaN
2	2025-02-02 07:43:00	client@example.com	me@example.com	Team Outing	Review the latest changes in the codebase.	manager@example.com	NaN
3	2025-02-08 14:22:00	admin@example.com	me@example.com	New Joiner Introduction	Review the latest changes in the codebase.	NaN	NaN
4	2025-03-02 01:37:00	client@example.com	me@example.com	Performance Feedback	Reminder: Submit your deliverables.	supervisor@example.com	admin@example.com

In [13]:
```
print("Dataset Shape:", df.shape)
df.info()
df.isnull().sum()
```

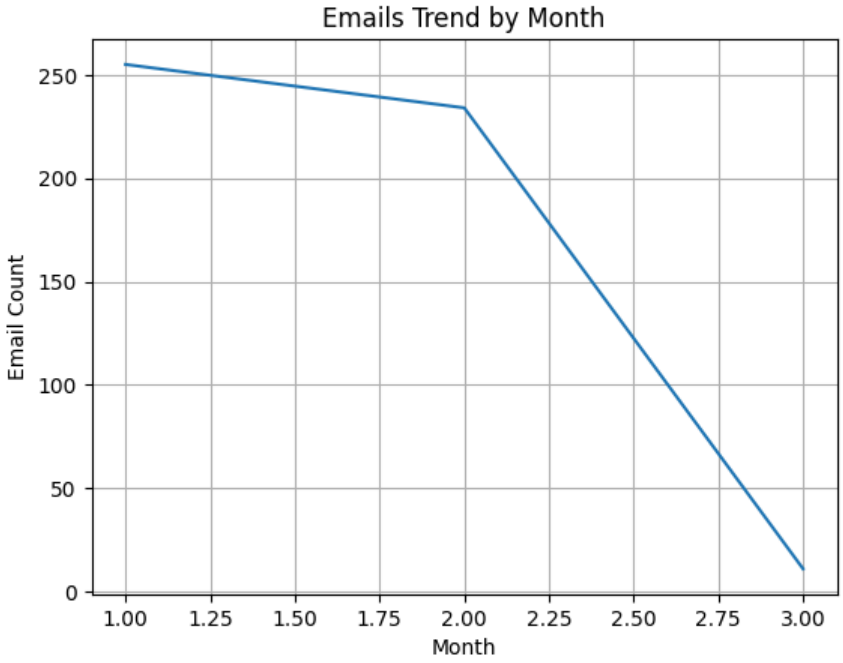
```
Dataset Shape: (500, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      500 non-null    object
 1   Sender    500 non-null    object
 2   Receiver  500 non-null    object
 3   Subject   500 non-null    object
 4   Body      500 non-null    object
 5   Cc        357 non-null    object
 6   Bcc       245 non-null    object
dtypes: object(7)
memory usage: 27.5+ KB
```

Out[13]:

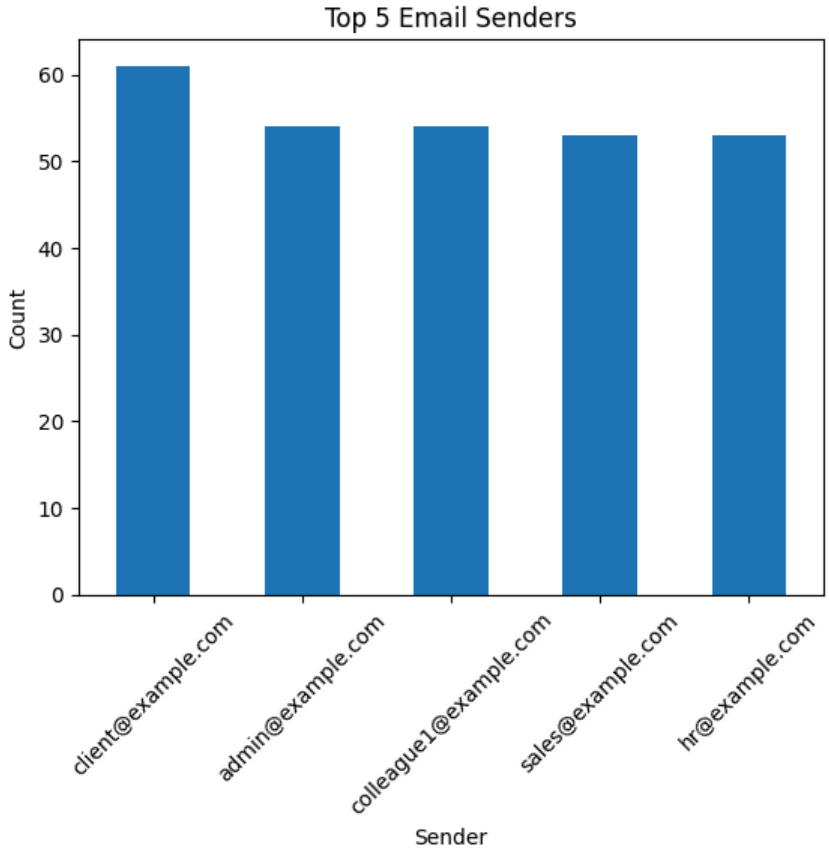
```
Date          0
Sender        0
Receiver      0
Subject       0
Body          0
Cc          143
Bcc         255
dtype: int64
```

In [14]:
```
# Convert Date
df['Date'] = pd.to_datetime(df['Date'])

# Extract features
df['Month'] = df['Date'].dt.month
df['Hour'] = df['Date'].dt.hour

# Email length
df['Email_Length'] = df['Body'].apply(len)
```

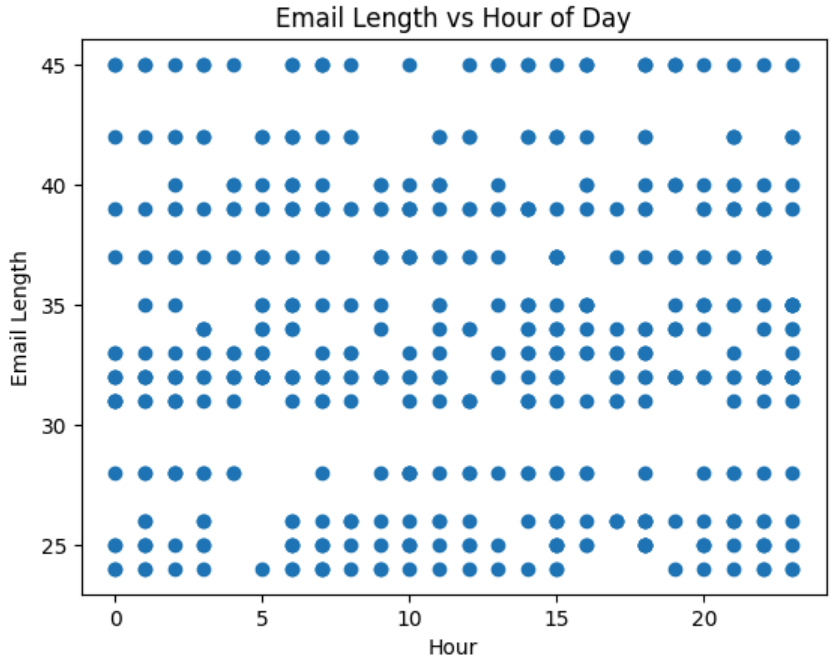
In [15]:
```
monthly_emails = df['Month'].value_counts().sort_index()

plt.figure()
plt.plot(monthly_emails)
plt.title("Emails Trend by Month")
plt.xlabel("Month")
plt.ylabel("Email Count")
plt.grid(True)
plt.show()
```



In [16]:
```
plt.figure()
df['Sender'].value_counts().head(5).plot(kind='bar')
plt.title("Top 5 Email Senders")
plt.xlabel("Sender")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.show()
```



In [17]:
```
plt.figure()
plt.scatter(df['Hour'], df['Email_Length'])
plt.title("Email Length vs Hour of Day")
plt.xlabel("Hour")
plt.ylabel("Email Length")
plt.show()
```


Email Length vs Hour of Day
Email Length
Hour
45
40
35
30
25
0
5
10
15
20

In [ ]: