In [1]:
```
#Q.1) 1. Perform exploratory data analysis (EDA) with datasets like email data s
#different insights from the data.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Load dataset
file_path = "large_email_dataset_with_cc_bcc.csv"
df = pd.read_csv(file_path)
df.head()
```

Out[1]:

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

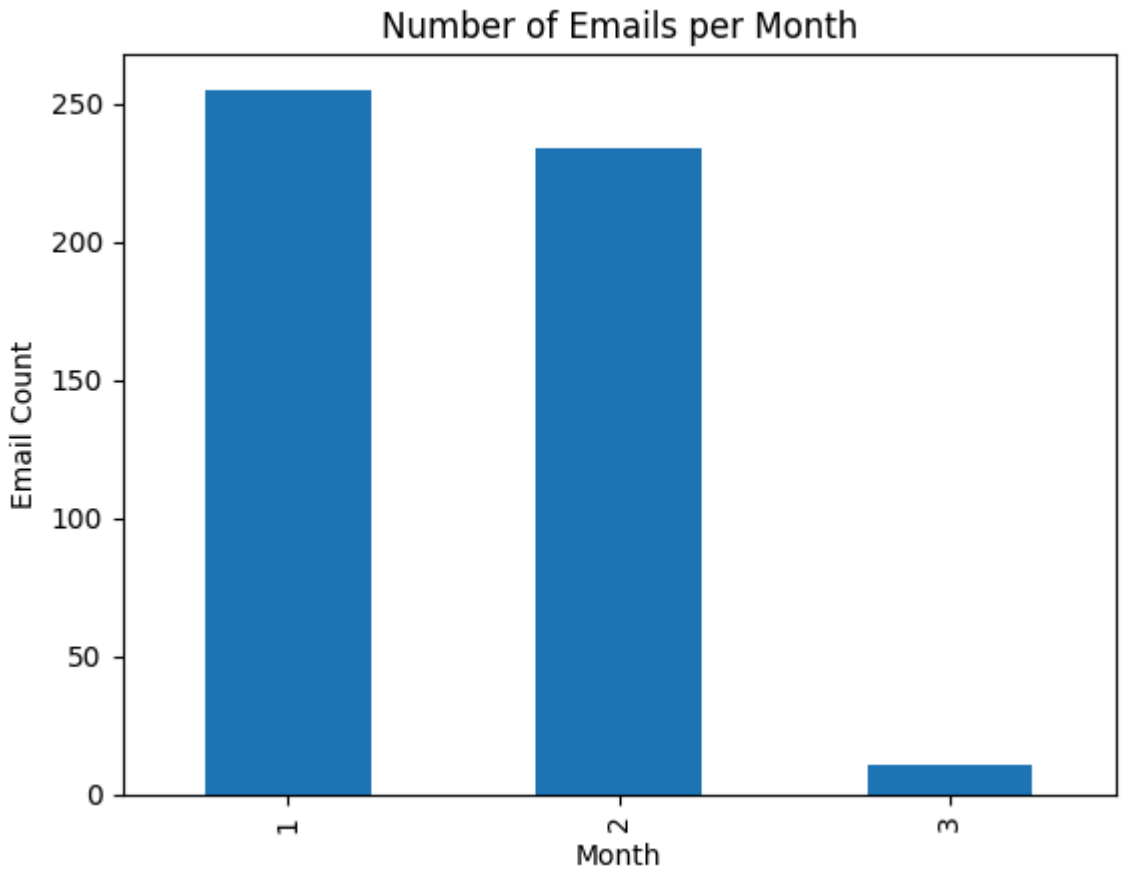
In [2]:
```
print("Dataset Shape:", df.shape)
df.info()
df.isnull().sum()
```

```
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   From     500 non-null    object
 2   To       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
```

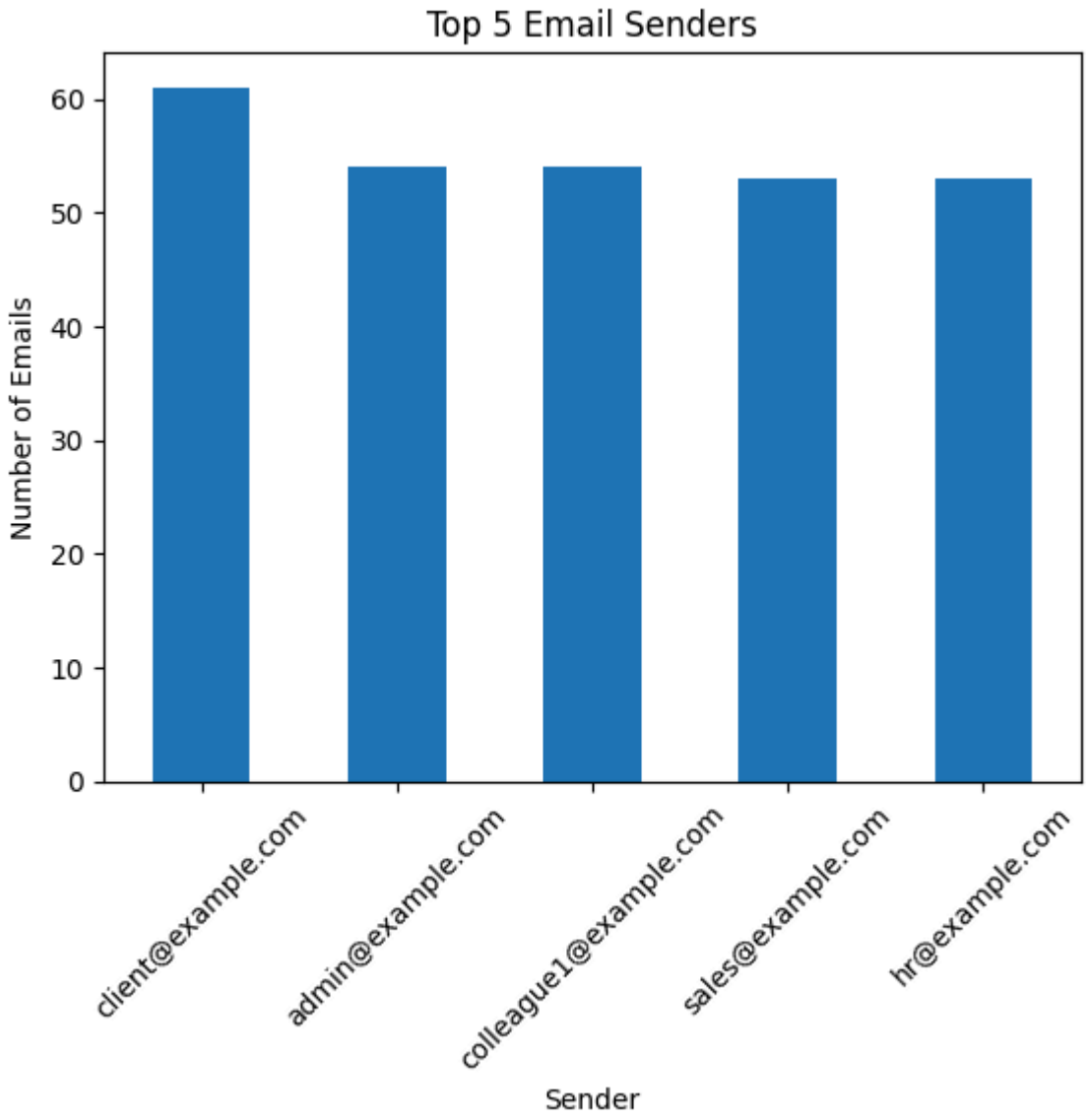
Out[2]:
```
Date          0
From          0
To            0
Subject       0
Body          0
Cc          143
Bcc         255
dtype: int64
```

In [4]:
```
df['Date'] = pd.to_datetime(df['Date'])
df['Year'] = df['Date'].dt.year
df['Month'] = df['Date'].dt.month
df['Day'] = df['Date'].dt.day
df['Hour'] = df['Date'].dt.hour
df['Email_Length'] = df['Body'].apply(len)
```

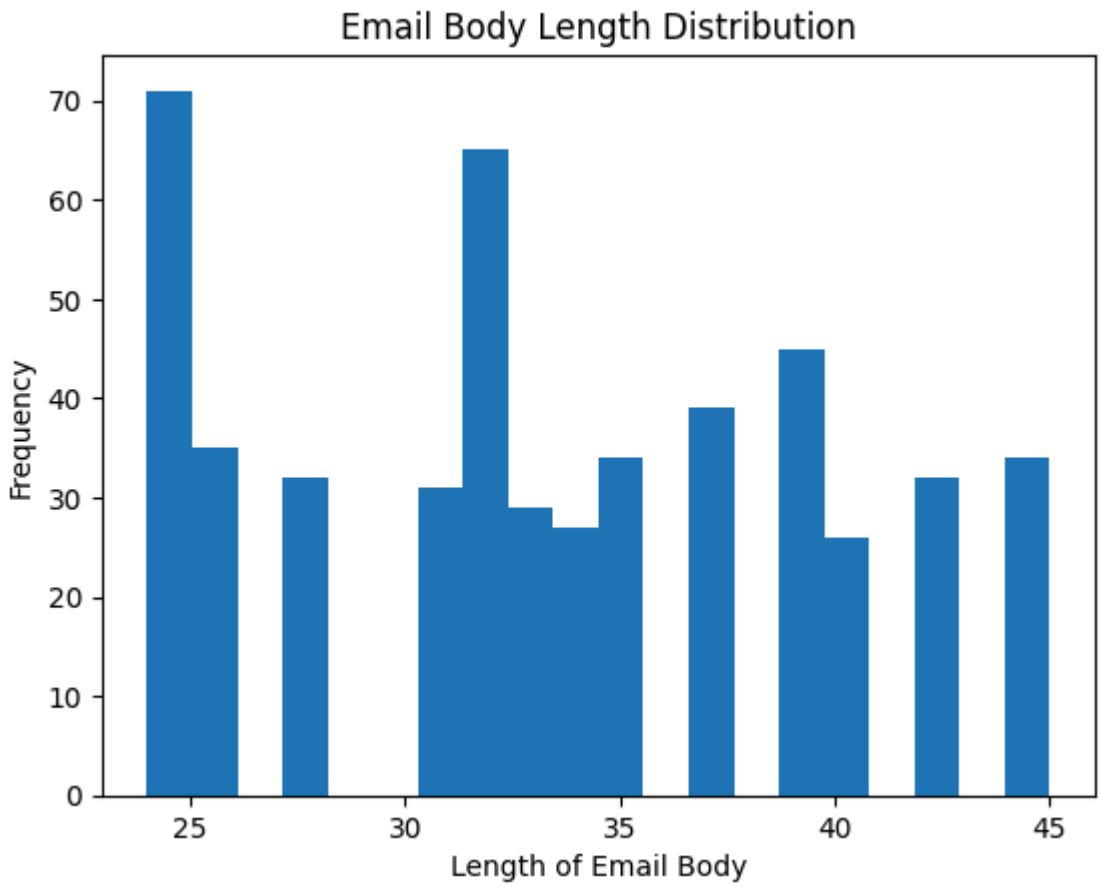
In [5]:
```
plt.figure()
df['Month'].value_counts().sort_index().plot(kind='bar')
plt.title("Number of Emails per Month")
plt.xlabel("Month")
plt.ylabel("Email Count")
plt.show()
```



In [6]:
```
plt.figure()
df['From'].value_counts().head(5).plot(kind='bar')
plt.title("Top 5 Email Senders")
plt.xlabel("Sender")
plt.ylabel("Number of Emails")
plt.xticks(rotation=45)
plt.show()
```



In [7]:
```
plt.figure()
plt.hist(df['Email_Length'], bins=20)
plt.title("Email Body Length Distribution")
plt.xlabel("Length of Email Body")
plt.ylabel("Frequency")
plt.show()
```



In [8]:
```
# Count emails with CC and BCC
cc_count = df['Cc'].notnull().sum()
bcc_count = df['Bcc'].notnull().sum()

print("Emails with CC:", cc_count)
print("Emails with BCC:", bcc_count)
```

```
Emails with CC: 357
Emails with BCC: 245
```

In [9]:
```
df['Subject'].value_counts().head(10)
```

Out[9]:
```
Subject
Code Review Request        40
Meeting Minutes            39
Invoice Reminder           38
Workshop Invite            37
Project Update             36
Team Outing                34
Client Feedback            34
Weekly Standup             33
Deadline Reminder          33
New Joiner Introduction    32
Name: count, dtype: int64
```

In [ ]: