In [1]:
```
#Q.3 Perform Time Series Analysis and apply the various visualization techniques
import pandas as pd
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]:
```
# Convert Date to datetime
df['Date'] = pd.to_datetime(df['Date'])

# Set Date as index
df.set_index('Date', inplace=True)

df.head()
```

Out[2]:

Date	From	To	Subject	Body	Cc	Bcc
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
2025-02-24 07:38:00	support@example.com	me@example.com	Invoice Reminder	Invoice is due next week.	team@example.com	NaN
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
2025-02-08 14:22:00	admin@example.com	me@example.com	New Joiner Introduction	Review the latest changes in the codebase.	NaN	NaN
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 [3]:
```
daily_emails = df.resample('D').size()
daily_emails.head()
```

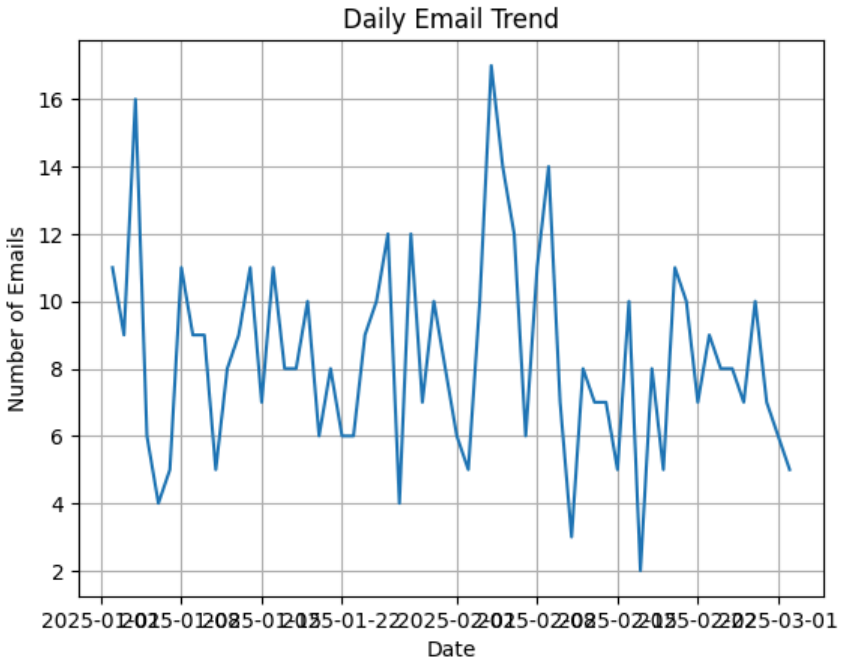
Out[3]:
```
Date
2025-01-02    11
2025-01-03     9
2025-01-04    16
2025-01-05     6
2025-01-06     4
Freq: D, dtype: int64
```

In [4]:
```
monthly_emails = df.resample('M').size()
monthly_emails.head()
```

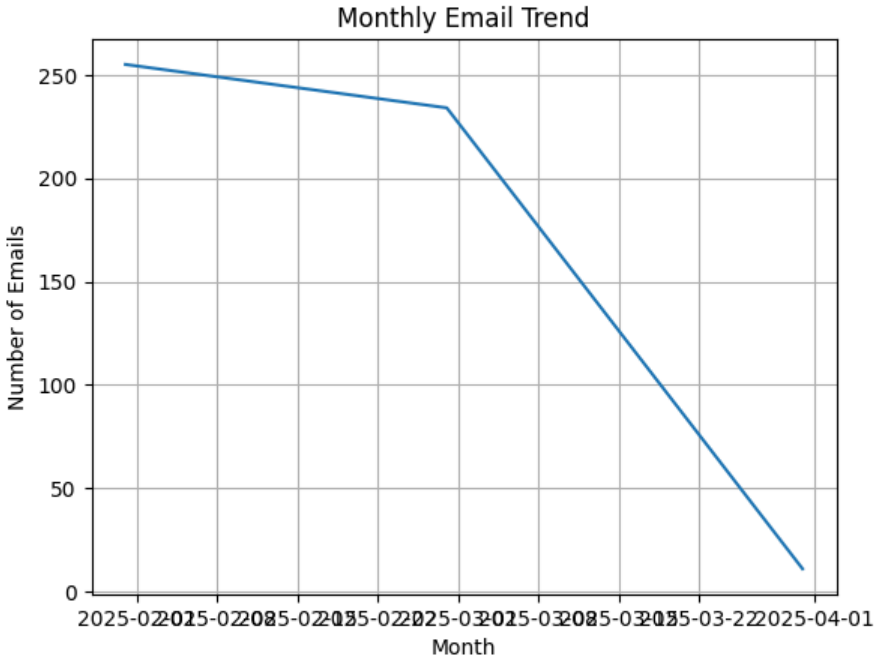
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_16000\1154504962.py:1: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead.
  monthly_emails = df.resample('M').size()
```

Out[4]:
```
Date
2025-01-31    255
2025-02-28    234
2025-03-31     11
Freq: ME, dtype: int64
```

In [5]:
```
plt.figure()
plt.plot(daily_emails)
plt.title("Daily Email Trend")
plt.xlabel("Date")
plt.ylabel("Number of Emails")
plt.grid(True)
plt.show()
```

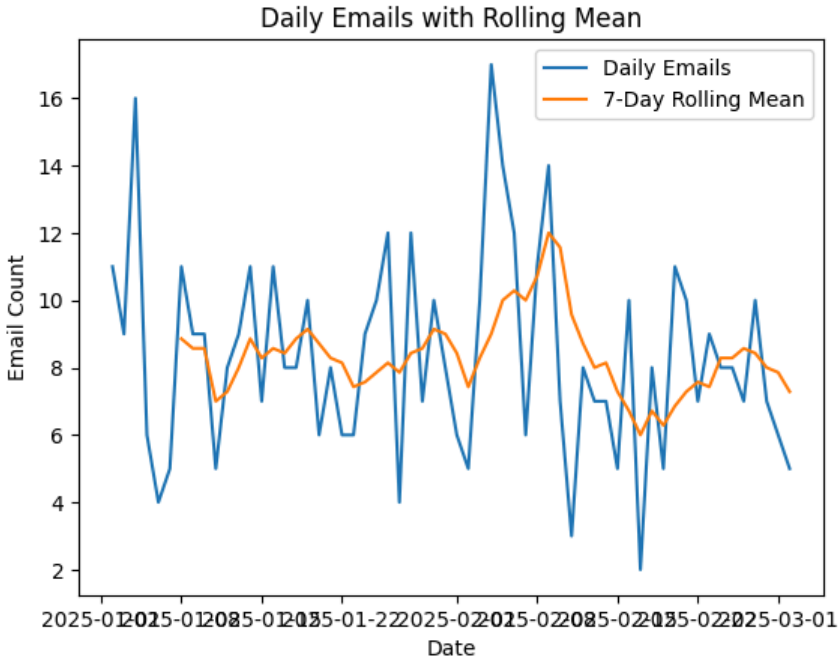


In [6]:
```
plt.figure()
plt.plot(monthly_emails)
plt.title("Monthly Email Trend")
plt.xlabel("Month")
plt.ylabel("Number of Emails")
plt.grid(True)
plt.show()
```

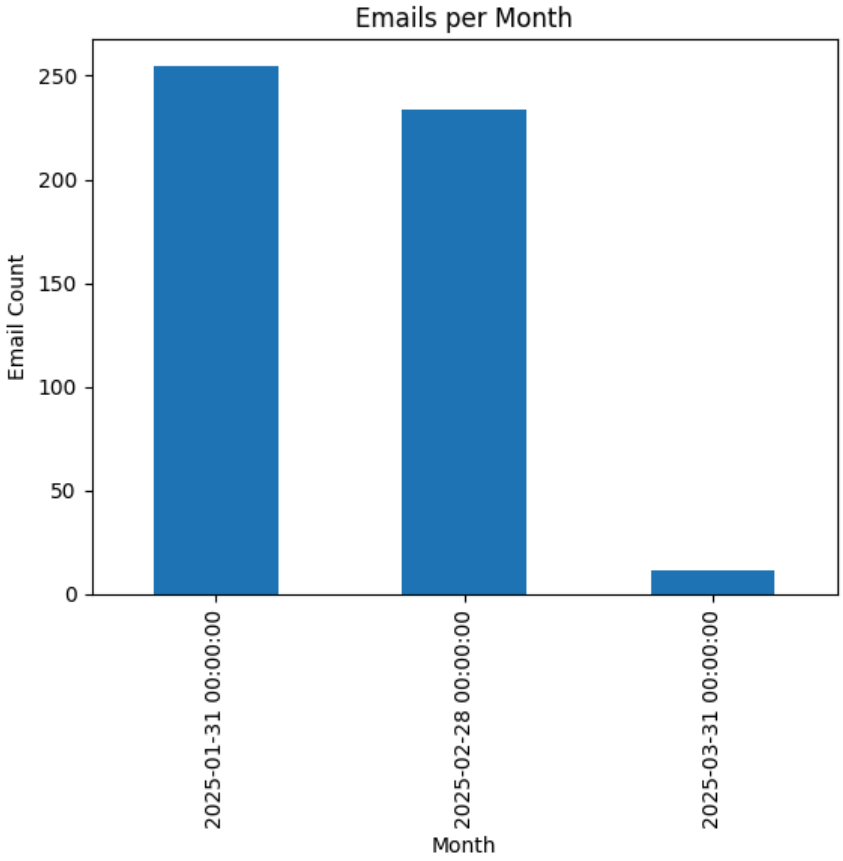


In [7]:
```
rolling_mean = daily_emails.rolling(window=7).mean()

plt.figure()
plt.plot(daily_emails, label='Daily Emails')
plt.plot(rolling_mean, label='7-Day Rolling Mean')
plt.title("Daily Emails with Rolling Mean")
plt.xlabel("Date")
plt.ylabel("Email Count")
plt.legend()
plt.show()
```



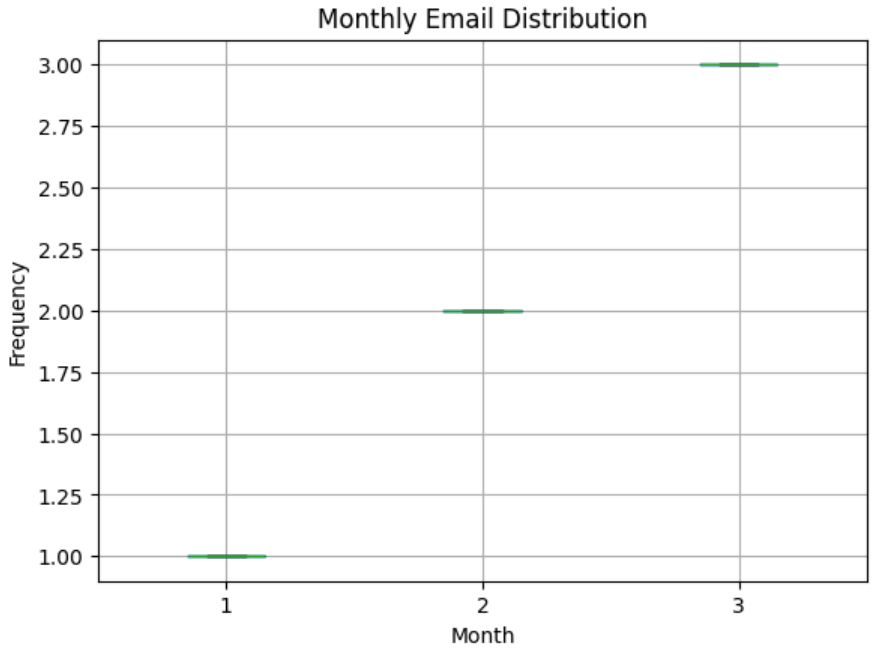
In [8]:
```
plt.figure()
monthly_emails.plot(kind='bar')
plt.title("Emails per Month")
plt.xlabel("Month")
plt.ylabel("Email Count")
plt.show()
```



In [9]:
```
df['Month'] = df.index.month

plt.figure()
df.boxplot(column='Month', by='Month')
plt.title("Monthly Email Distribution")
plt.suptitle("")
plt.xlabel("Month")
plt.ylabel("Frequency")
plt.show()
```

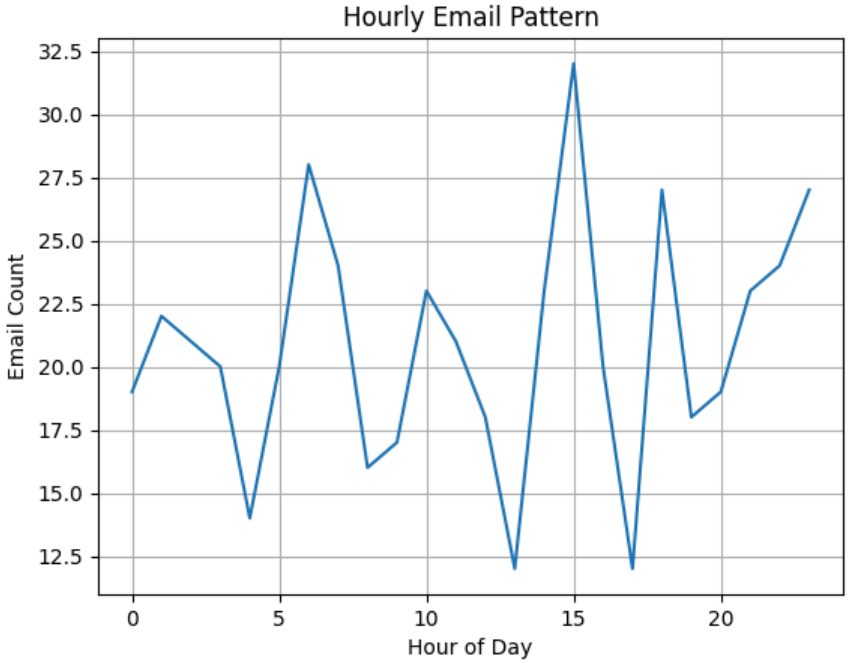
```
<Figure size 640x480 with 0 Axes>
```



In [10]:
```
df['Hour'] = df.index.hour
hourly_emails = df.groupby('Hour').size()

plt.figure()
hourly_emails.plot(kind='line')
plt.title("Hourly Email Pattern")
```

```
plt.xlabel("Hour of Day")
plt.ylabel("Email Count")
plt.grid(True)
plt.show()
```



In [ ]:

In [ ]: