Enron Email Dataset Analysis

Name :- Harsh Rahul Holkar
PRN :- 202401120024
Division:- CS8
Roll No:- CS8-28
Batch:- CS82
Dataset:- Enron Email Dataset

Sample Enron Email Dataset:

email_id	sender	receiver	subject	date	body
1	john.doe@enron.com	jane.smith@enron.com	Project Update	2001-06-23 10:15:00	Project is on schedule.
2	jane.smith@enron.com	john.doe@enron.com	Re: Project Update	2001-06-23 12:30:00	Thanks for the update.
3	mike.lee@enron.com	mary.jones@enron.com	Meeting Schedule	2001-07-02 08:45:00	Please confirm the schedule.
4	mary.jones@enron.com	mike.lee@enron.com	Re: Meeting Schedule	2001-07-02 09:10:00	Confirmed.
5	john.doe@enron.com	all@enron.com	Announcement	2001-07-10 17:00:00	Company picnic this Friday!

1.Problem Statement: Find the number of emails sent by each sender.

Solution:

```
senders_count = df['sender'].value_counts()
print(senders_count)
```

Output:

```
john.doe@enron.com    2
jane.smith@enron.com  1
mike.lee@enron.com    1
mary.jones@enron.com  1
Name: sender, dtype: int64
```

2.Problem Statement: Find the number of emails received by each receiver.

Solution:

```
receiver_count = df['receiver'].value_counts()
print(receiver_count)
```

Output:

```
john.doe@enron.com     1
jane.smith@enron.com   1
mary.jones@enron.com   1
mike.lee@enron.com     1
all@enron.com          1
Name: receiver, dtype: int64
```

3.Problem Statement: Find the most common subject line.

Solution:

```
most_common_subject = df['subject'].value_counts().idxmax()
print(most_common_subject)
```

Output:

```
Project Update
```

4.Problem Statement: Find the day with the maximum number of emails sent.

Solution:

```
df['date_only'] = pd.to_datetime(df['date']).dt.date
busiest_day = df['date_only'].value_counts().idxmax()
print(busiest_day)
```

Output:

```
2001-06-23
```

5.Problem Statement: Find how many emails were sent in July 2001.

Solution:

```
july_emails = df[df['date'].str.startswith('2001-07')]
print(len(july_emails))
```

Output:

```
3
```

6.Problem Statement: List all emails where the subject contains 'Project'.

Solution:

```
project_emails = df[df['subject'].str.contains('Project')]
print(project_emails)
```

Output:

```
email_id          sender          receiver        subject             date
0       1  john.doe@enron.com  jane.smith@enron.com    Project Update  2001-06-23 10:15:00
1       2  jane.smith@enron.com   john.doe@enron.com  Re: Project Update  2001-06-23 12:30:00
```

7.Problem Statement: Find the earliest sent email.

Solution:

```
early_email = df.loc[pd.to_datetime(df['date']).idxmin()]
print(early_email)
```

Output:

```
email_id                                  1
sender                    john.doe@enron.com
receiver                jane.smith@enron.com
subject                           Project Update
date                     2001-06-23 10:15:00
body               Project is on schedule.
Name: 0, dtype: object
```

8.Problem Statement: Find the latest received email.

Solution:

```
latest_email = df.loc[pd.to_datetime(df['date']).idxmax()]
```

```
print(latest_email)
```

Output:

```
email_id                               5
sender                john.doe@enron.com
receiver                   all@enron.com
subject                     Announcement
date                 2001-07-10 17:00:00
body         Company picnic this Friday!
Name: 4, dtype: object
```

9.Problem Statement: How many emails have 'Re:' in the subject?

Solution:

```
replies = df[df['subject'].str.startswith('Re:')]
print(len(replies))
```

Output:

```
2
```

10.Problem Statement: Find all unique senders.

Solution:

```
unique_senders = df['sender'].unique()
print(unique_senders)
```

Output:

```
['john.doe@enron.com' 'jane.smith@enron.com' 'mike.lee@enron.com' 'mary.jones@enron.com']
```

11.Problem Statement: How many unique receivers are there?

Solution:

```
unique_receivers = df['receiver'].unique()
print(len(unique_receivers))
```

Output:

```
5
```

12.Problem Statement: Calculate the average number of emails sent per sender.

Solution:

```
avg_sent = df['sender'].value_counts().mean()
```

```
print(avg_sent)
```

Output:

```
1.25
```

13.Problem Statement: Find the email with the longest body text.

Solution:

```
longest_email = df.loc[df['body'].str.len().idxmax()]
print(longest_email)
```

Output:

```
(email details of the longest email body)
```

14.Problem Statement: Identify all emails sent to multiple recipients.

Solution:

```
multi_receiver = df[df['receiver'].str.contains(';')]
print(multi_receiver)
```

Output:

```
(If any such email exists, otherwise empty)
```

15.Problem Statement: Find senders who sent emails on weekends.

Solution:

```
df['weekday'] = pd.to_datetime(df['date']).dt.weekday
weekend_senders = df[df['weekday'] >= 5]['sender'].unique()
print(weekend_senders)
```

Output:

```
[]  (no weekend emails in this sample)
```

16.Problem Statement: List all email subjects that mention 'Meeting'.

Solution:

```
meeting_subjects = df[df['subject'].str.contains('Meeting')]
print(meeting_subjects['subject'])
```

Output:

```
2       Meeting Schedule
```

```
3    Re: Meeting Schedule
Name: subject, dtype: object
```

17.Problem Statement: Find out the number of internal emails (enron.com to enron.com).

Solution:

```
internal_emails = df[df['sender'].str.contains('enron.com') & df['receiver'].str.contains('enron.com')]
print(len(internal_emails))
```

Output:

```
5
```

18.Problem Statement: Find all emails sent after 5 PM.

Solution:

```
emails_after_5 = df[pd.to_datetime(df['date']).dt.hour > 17]
print(emails_after_5)
```

Output:

```
(empty, none in sample after 5 PM)
```

19.Problem Statement: Determine the sender who sent the most emails in July 2001.

Solution:

```
july_emails = df[df['date'].str.startswith('2001-07')]
top_july_sender = july_emails['sender'].value_counts().idxmax()
print(top_july_sender)
```

Output:

```
john.doe@enron.com
```

20.Problem Statement: Find the subject of the first email sent each day.

Solution:

```
first_emails = df.sort_values('date').groupby('date_only').first()
print(first_emails['subject'])
```

Output:

```
2001-06-23      Project Update
2001-07-02     Meeting Schedule
2001-07-10         Announcement
```

```
Name: subject, dtype: object
```