Name : Pranav Tanaji Lamkhade

Roll no : 69

Prn : 202401120062

Batch : CS84

```
import pandas as pd
import numpy as np

# 1. Assume 'emails.csv' is the loaded dataset
emails = pd.read_csv('emails.csv')

# 2. Unique senders
unique_senders = emails['sender'].nunique()

# 3. Unique receivers
unique_receivers = emails['receiver'].nunique()

# 4. Sender with the most emails
top_sender = emails['sender'].value_counts().idxmax()

# 5. Receiver with most emails
top_receiver = emails['receiver'].value_counts().idxmax()

# 6. Average length of email subjects
emails['subject_length'] = emails['subject'].fillna('').apply(len)
avg_subject_length = emails['subject_length'].mean()

# 7. Percentage of emails with empty subjects
```

```
empty_subjects = emails['subject'].isnull().mean() * 100

# 8. Day with most emails
emails['date'] = pd.to_datetime(emails['date'])
busiest_day = emails['date'].dt.day_name().value_counts().idxmax()

# 9. Month with highest email activity
busiest_month = emails['date'].dt.month_name().value_counts().idxmax()

# 10. Top 5 email domains among senders
emails['domain'] = emails['sender'].str.split('@').str[-1]
top_5_domains = emails['domain'].value_counts().head(5)

# 11. Emails between internal employees
internal_emails = emails[emails['sender'].str.contains('enron.com') &
emails['receiver'].str.contains('enron.com')]
internal_email_count = len(internal_emails)

# 12. Average number of recipients per email
emails['recipient_count'] = emails['receiver'].apply(lambda x: len(str(x).split(',')))
avg_recipients = emails['recipient_count'].mean()

# 13. Top 10 keywords in email subjects
from collections import Counter

keywords = ' '.join(emails['subject'].dropna()).lower().split()
common_keywords = Counter(keywords).most_common(10)

# 14. Number of duplicate emails
duplicate_emails = emails.duplicated().sum()
```

```
# 15. Earliest and latest email timestamp
earliest_email = emails['date'].min()
latest_email = emails['date'].max()

# 16. Emails sent outside working hours
emails['hour'] = emails['date'].dt.hour
night_emails = emails[(emails['hour'] >= 21) | (emails['hour'] <= 6)]
outside_hours_count = len(night_emails)

# 17. Correlation between email length and number of recipients
emails['body_length'] = emails['body'].fillna('').apply(len)
correlation = emails['body_length'].corr(emails['recipient_count'])

# 18. Emails without body text
empty_body_count = emails['body'].isnull().sum()

# 19. Group emails by year and plot trend
yearly_email_count = emails['date'].dt.year.value_counts().sort_index()

# 20. Total number of emails
total_emails = len(emails)


print("Solutions Computed Successfully!")
```