```python
# airline_delay_analysis.py

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Simulate realistic flight delay data (since actual file not available)
np.random.seed(42)
n = 2000
airports = ['DEL', 'BOM', 'BLR', 'MAA', 'CCU']
months = np.random.choice(range(1, 13), n)
seasons = pd.cut(months, bins=[0, 3, 6, 9, 12], labels=['Winter', 'Summer', 'Monsoon', 'Post-Monsoon'])

data = pd.DataFrame({
    'FlightNumber': ['AI' + str(i).zfill(4) for i in range(n)],
    'Airport': np.random.choice(airports, n),
    'ScheduledDeparture': pd.date_range(start='2024-01-01', periods=n, freq='H'),
    'ActualDeparture': pd.date_range(start='2024-01-01', periods=n, freq='H') + pd.to_timedelta(np.random.randint(0, 180, n), unit='m'),
    'DelayReason': np.random.choice(['Weather', 'Carrier', 'ATC', 'Security', 'None'], n, p=[0.25, 0.35, 0.2, 0.1, 0.1]),
    'RouteLength_km': np.random.normal(1200, 300, n).astype(int),
    'FlightFreq_perDay': np.random.choice([1, 2, 3, 4], n),
    'Month': months,
    'Season': seasons
})

# 2. Feature Engineering
data['DelayMinutes'] = (data['ActualDeparture'] - data['ScheduledDeparture']).dt.total_seconds() / 60
data = data[data['DelayReason'] != 'None']  # remove
```

```python
non-delayed for focused analysis
data['DelayCategory'] = pd.cut(data['DelayMinutes'], bins=[0, 30, 90, np.inf], labels=['Short', 'Medium', 'Long'])

# 3. Exploratory Data Analysis
print("\n--- Average Delay by Airport ---")
print(data.groupby('Airport')['DelayMinutes'].mean())

print("\n--- Delay Reason Distribution ---")
print(data['DelayReason'].value_counts(normalize=True) * 100)

# Bar plot of delay reasons
plt.figure(figsize=(6, 4))
sns.countplot(data=data, x='DelayReason', order=data['DelayReason'].value_counts().index, palette='Set2')
plt.title("Delay Reason Distribution")
plt.ylabel("Number of Flights")
plt.xticks(rotation=30)
plt.tight_layout()
plt.show()

# Line plot of monthly delays
monthly_delay = data.groupby('Month')['DelayMinutes'].mean()
plt.figure(figsize=(7, 4))
monthly_delay.plot(marker='o')
plt.title("Average Monthly Delay (All Airports)")
plt.xlabel("Month")
plt.ylabel("Avg Delay (min)")
plt.grid()
plt.tight_layout()
plt.show()

# Heatmap of delay by airport & month
pivot = data.pivot_table(index='Airport', columns='Month', values='DelayMinutes', aggfunc='mean')
plt.figure(figsize=(8, 5))
sns.heatmap(pivot, annot=True, fmt=".1f", cmap='coolwarm')
```

```
plt.title("Avg Delay by Airport and Month")
plt.tight_layout()
plt.show()

# 4. Root Cause by Season
seasonal_cause = pd.crosstab(data['Season'], data['DelayReason'], normalize='index') * 100
seasonal_cause.plot(kind='bar', stacked=True, colormap='tab20', figsize=(8, 4))
plt.title("Root Cause Attribution by Season (%)")
plt.ylabel("Percentage of Delays")
plt.tight_layout()
plt.show()

# 5. Correlation Analysis
corr_data = data[['DelayMinutes', 'RouteLength_km', 'FlightFreq_perDay']]
corr = corr_data.corr()
sns.heatmap(corr, annot=True, cmap='Blues')
plt.title("Correlation Matrix")
plt.tight_layout()
plt.show()

# 6. Top 5 Most Delayed Flights
top5 = data.sort_values(by='DelayMinutes', ascending=False).head(5)
print("\n--- Top 5 Most Delayed Flights ---")
print(top5[['FlightNumber', 'Airport', 'DelayMinutes', 'DelayReason', 'ScheduledDeparture']])

```