Identifying Patterns in Tourism Data to Enhance travel Industry Decision Making

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.preprocessing import LabelEncoder
from statsmodels.tsa.seasonal import seasonal_decompose
from prophet import Prophet

# Step 1: Load Data
df = pd.read_csv("tourism_data.csv")
print(df.head())

# Step 2: Data Preprocessing
# Convert date column
df['date'] = pd.to_datetime(df['date'])

# Handle missing values
df.fillna(method='ffill', inplace=True)

# Encode categorical variables
label_encoder = LabelEncoder()
```

```
df['country_encoded'] = label_encoder.fit_transform(df['country'])

# Step 3: EDA (Exploratory Data Analysis)
# Tourist inflow trend
monthly_data = df.groupby(df['date'].dt.to_period('M')).agg({'tourist_count':
'sum'}).reset_index()
monthly_data['date'] = monthly_data['date'].dt.to_timestamp()

plt.figure(figsize=(12,6))
sns.lineplot(data=monthly_data, x='date', y='tourist_count')
plt.title('Monthly Tourist Inflow')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Pie chart of tourists by country
country_data = df['country'].value_counts().head(10)
country_data.plot(kind='pie', autopct='%1.1f%%', title='Top 10 Countries by
Tourist Count')
plt.ylabel('')
plt.tight_layout()
plt.show()

# Step 4: Time Series Forecasting
# Prepare data for Prophet
prophet_df = monthly_data.rename(columns={'date': 'ds', 'tourist_count': 'y'})
model = Prophet()
```

```
model.fit(prophet_df)

# Make future predictions
future = model.make_future_dataframe(periods=12, freq='M')
forecast = model.predict(future)

# Plot forecast
model.plot(forecast)
plt.title('Tourist Forecast for Next Year')
plt.show()

# Plot forecast components
model.plot_components(forecast)
plt.show()
```