

```

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

# 1. Simulate International Market Data
regions = ['North America', 'Europe', 'Asia']
customers_per_region = 500

data = []

for region in regions:
    for _ in range(customers_per_region):
        age = random.randint(18, 65)
        spending_score = random.randint(1, 100)
        interest_level = random.choices(['Low', 'Medium', 'High'], weights=[0.2, 0.5, 0.3])[0]
        competitor_preference = random.choice(['Us', 'Competitor A', 'Competitor B'])
        converted = random.choices([0, 1], weights=[0.8, 0.2])[0] # before strategy
        data.append([region, age, spending_score, interest_level, competitor_preference, converted])

df = pd.DataFrame(data, columns=['Region', 'Age', 'SpendingScore', 'InterestLevel', 'CompetitorPreference', 'Converted'])

# 2. Market Insights and Trends
plt.figure(figsize=(10, 5))
sns.boxplot(data=df, x='Region', y='SpendingScore', palette='Set3')
plt.title("Spending Score Distribution by Region")
plt.show()

interest_counts = df.groupby(['Region', 'InterestLevel']).size().unstack()
interest_counts.plot(kind='bar', stacked=True, figsize=(10, 5), colormap='coolwarm')
plt.title("Customer Interest Levels Across Regions")
plt.ylabel("Number of Customers")
plt.show()

# 3. Competitor Strategy Insights
comp_pref = df.groupby(['Region', 'CompetitorPreference']).size().unstack()
comp_pref.plot(kind='bar', stacked=True, figsize=(10, 5), colormap='Pastel1')
plt.title("Customer Competitor Preferences by Region")
plt.ylabel("Number of Customers")
plt.show()

# 4. Qualified Leads Before Strategy
leads_before = df['Converted'].sum()
print(f"✅ Qualified Leads Before Strategy: {leads_before}")

# 5. Simulate Post-Strategy Impact (15% improvement)
df['Converted_Post'] = df['Converted']
improve_indices = df[df['Converted'] == 0].sample(frac=0.15, random_state=42).index
df.loc[improve_indices, 'Converted_Post'] = 1

leads_after = df['Converted_Post'].sum()
improvement = leads_after - leads_before

print(f"🔄 Qualified Leads After Strategy: {leads_after}")
print(f"📈 Increase in Leads: {improvement} ({(improvement / leads_before * 100):.2f}%)")

# 6. Regional Expansion Opportunity
conversion_by_region = df.groupby("Region")[['Converted', 'Converted_Post']].sum()
conversion_by_region["Increase"] = conversion_by_region["Converted_Post"] - conversion_by_region["Converted"]

print("\n📍 Regional Lead Improvement:")
print(conversion_by_region)

conversion_by_region[["Converted", "Converted_Post']].plot(kind='bar', figsize=(10, 5))
plt.title("Regional Lead Conversions Before vs After Strategy")
plt.ylabel("Leads")
plt.show()

```

```
↻ /tmp/ipython-input-1-3227894087.py:26: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `leg
sns.boxplot(data=df, x='Region', y='SpendingScore', palette='Set3')
```

