In [6]:
```
sc
```

Out[6]:

SparkContext

Spark UI

Version	v4.0.1
Master	local[*]
AppName	PySparkShell

In [44]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
```

In [45]:
```
df = pd.read_csv("finaldataset_1.csv")
```

In [62]:
```
# --- 2.1 Data Cleaning & Exploration ---
print("\n--- 2.1 Data Cleaning & Exploration ---")
print("No. of records:", len(df))
n_records = len(df)

print("\nMissing values per column:")
print(df.isnull().sum())
```

```
--- 2.1 Data Cleaning & Exploration ---
No. of records: 39588

Missing values per column:
exporter                0
year                    0
product                 0
tradevalue              0
tradeshare           4116
expgrowth            1992
expgrowthTRIM        5726
BANK                  108
BANK_W3                 0
TWIN                 1745
RZ                   1477
FL                   1477
TANG                 1477
ofagdp              23731
pcrdbofgdp           1744
stmktcap             8447
RecessionAbroad         0
GDPgrAbroad             0
durables                0
loss                    0
loss2                   0
GDPcap                  0
developed               0
developing              0
blanguar            14853
liqsup              14853
forba               14853
forbb               14853
recaps              14853
debtrelief          14853
policytot           14853
recession               0
GDPgr                   0
INVSA                1330
CCC                  1330
RZyoung              2507
rznoncrisis          1477
caplab               1477
rd                   1477
homogeneity          9237
n                    2320
herf                 2320
intout               2320
contcrisis              0
dtype: int64
```

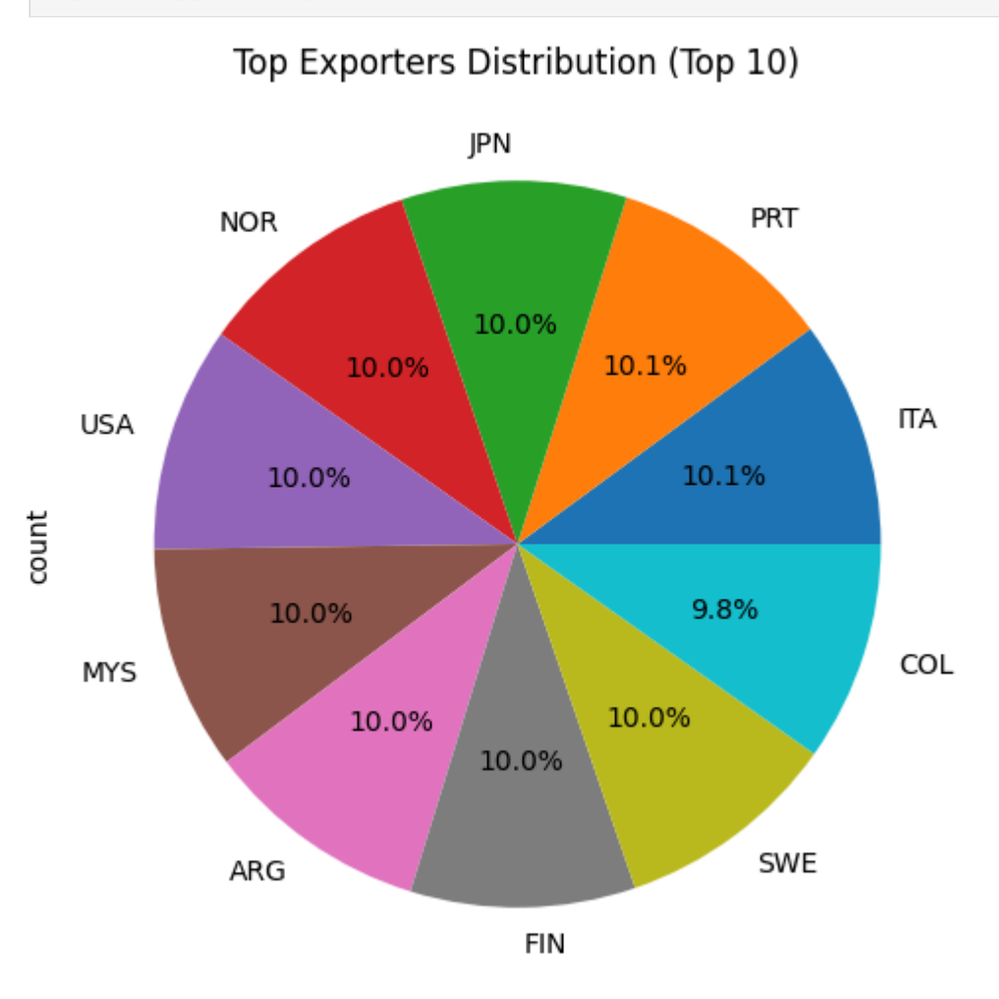
In [63]:
```
# Unique categorical columns
categorical_cols = df.select_dtypes(exclude=[np.number]).columns
print("\nUnique values in categorical columns:")
print({col: df[col].nunique() for col in categorical_cols})
```

```
Unique values in categorical columns:
{'exporter': 22}
```

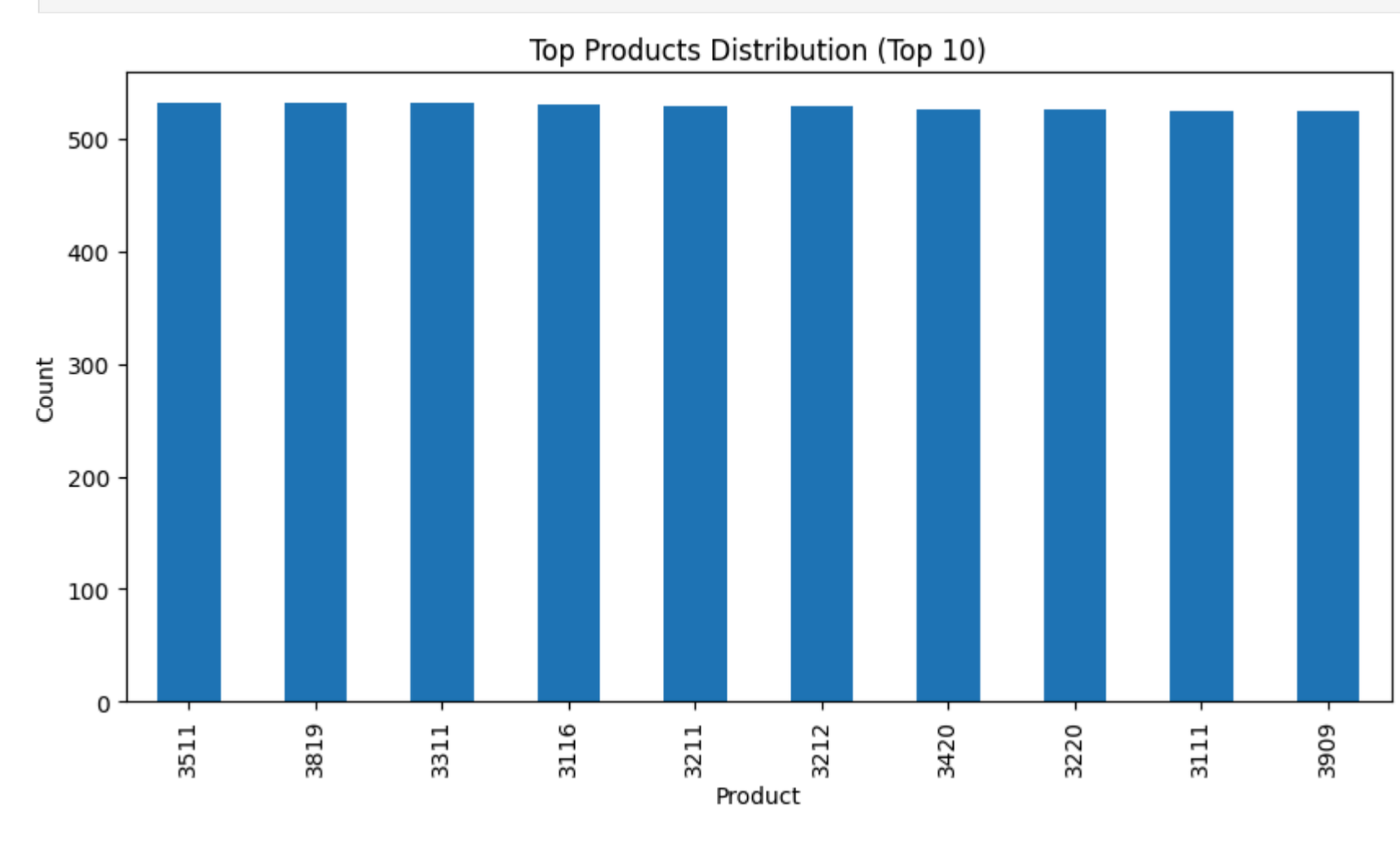
In [64]:
```
# Numeric summary
numeric_summary = df.select_dtypes(include=[np.number]).agg(
    ['mean','median','std',
     lambda x: np.percentile(x.dropna(),25),
     lambda x: np.percentile(x.dropna(),75)]
).T
numeric_summary.columns = ['mean','median','std','q1','q3']
print("\nNumerical summary (mean, median, std, Q1, Q3):")
print(numeric_summary)
```

```
Numerical summary (mean, median, std, Q1, Q3):
                          mean        median           std           q1  \
year               1993.528241   1994.000000  7.623218e+00  1987.000000
product            3508.390093   3523.000000  2.872312e+02  3215.000000
tradevalue       840004.572535  25400.830278  4.286242e+06  2180.100245
tradeshare            0.013819      0.002581  4.178049e-02     0.000587
expgrowth             0.104838      0.084501  7.865420e-01    -0.087494
expgrowthTRIM         0.098511      0.084501  3.389921e-01    -0.064466
BANK                  0.044200      0.000000  2.055409e-01     0.000000
BANK_W3               0.128397      0.000000  3.345361e-01     0.000000
TWIN                  0.002854      0.000000  5.334629e-02     0.000000
RZ                    0.308841      0.240000  3.264931e-01     0.140000
FL                    0.090295      0.088000  1.705702e-02     0.082000
TANG                  0.298723      0.280000  1.104495e-01     0.210000
ofagdp                0.302439      0.083422  3.797866e-01     0.037357
pcrdbofgdp            0.628470      0.551650  4.357287e-01     0.251799
stmktcap              0.440333      0.269242  4.905659e-01     0.111769
RecessionAbroad       0.148151      0.050467  2.045786e-01     0.001945
GDPgrAbroad           2.482578      2.476537  2.181723e+00     0.981274
durables              0.456729      0.000000  4.981304e-01     0.000000
loss                  0.005649      0.000000  5.271428e-02     0.000000
loss2                 0.000540      0.000000  3.952317e-02     0.000000
GDPcap            10456.790126   4079.520342  1.172864e+04  1392.541284
developed             0.383323      0.000000  4.862021e-01     0.000000
developing            0.059185      0.000000  2.359729e-01     0.000000
blanguar              0.025551      0.000000  1.577942e-01     0.000000
liqsup                0.022236      0.000000  1.474522e-01     0.000000
forba                 0.006226      0.000000  7.866055e-02     0.000000
forbb                 0.031898      0.000000  1.757324e-01     0.000000
recaps                0.012816      0.000000  1.124816e-01     0.000000
debtrelief            0.006266      0.000000  7.891393e-02     0.000000
policytot             0.104993      0.000000  5.741787e-01     0.000000
recession             0.221077      0.000000  4.149775e-01     0.000000
GDPgr                 3.430563      3.663312  3.492356e+00     1.774054
INVSA                 0.168396      0.158104  6.333888e-02     0.132553
CCC                   1.044697      0.977674  4.064324e-01     0.818567
RZyoung               0.660228      0.660000  6.024138e-01     0.570000
rznoncrisis           0.128036      0.050000  2.781763e-01    -0.010000
caplab               29.586375     21.780001  3.066396e+01    18.049999
rd                    0.027806      0.010000  6.811557e-02     0.010000
homogeneity           0.229943      0.000000  4.208029e-01     0.000000
n                     1.147987      1.179000  3.075209e-01     0.879000
herf                  0.805050      0.560000  5.564413e-01     0.497000
intout                1.026452      0.982000  1.550515e-01     0.931000
contcrisis            0.018970      0.000000  1.364221e-01     0.000000

                          q3
year               2000.000000
product            3823.000000
tradevalue       206898.675452
tradeshare            0.010035
expgrowth             0.273260
expgrowthTRIM         0.246487
BANK                  0.000000
BANK_W3               0.000000
TWIN                  0.000000
RZ                    0.450000
FL                    0.101000
TANG                  0.370000
ofagdp                0.499521
pcrdbofgdp            0.896425
stmktcap              0.586394
RecessionAbroad       0.221524
GDPgrAbroad           3.726201
durables              1.000000
loss                  0.000000
loss2                 0.000000
GDPcap            19465.136719
developed             1.000000
developing            0.000000
blanguar              0.000000
liqsup                0.000000
forba                 0.000000
forbb                 0.000000
recaps                0.000000
debtrelief            0.000000
policytot             0.000000
recession             0.000000
GDPgr                 5.300003
INVSA                 0.204679
CCC                   1.345260
RZyoung               0.870000
rznoncrisis           0.250000
caplab               31.080000
rd                    0.020000
homogeneity           0.000000
n                     1.333000
herf                  0.983000
intout                1.119000
contcrisis            0.000000
```

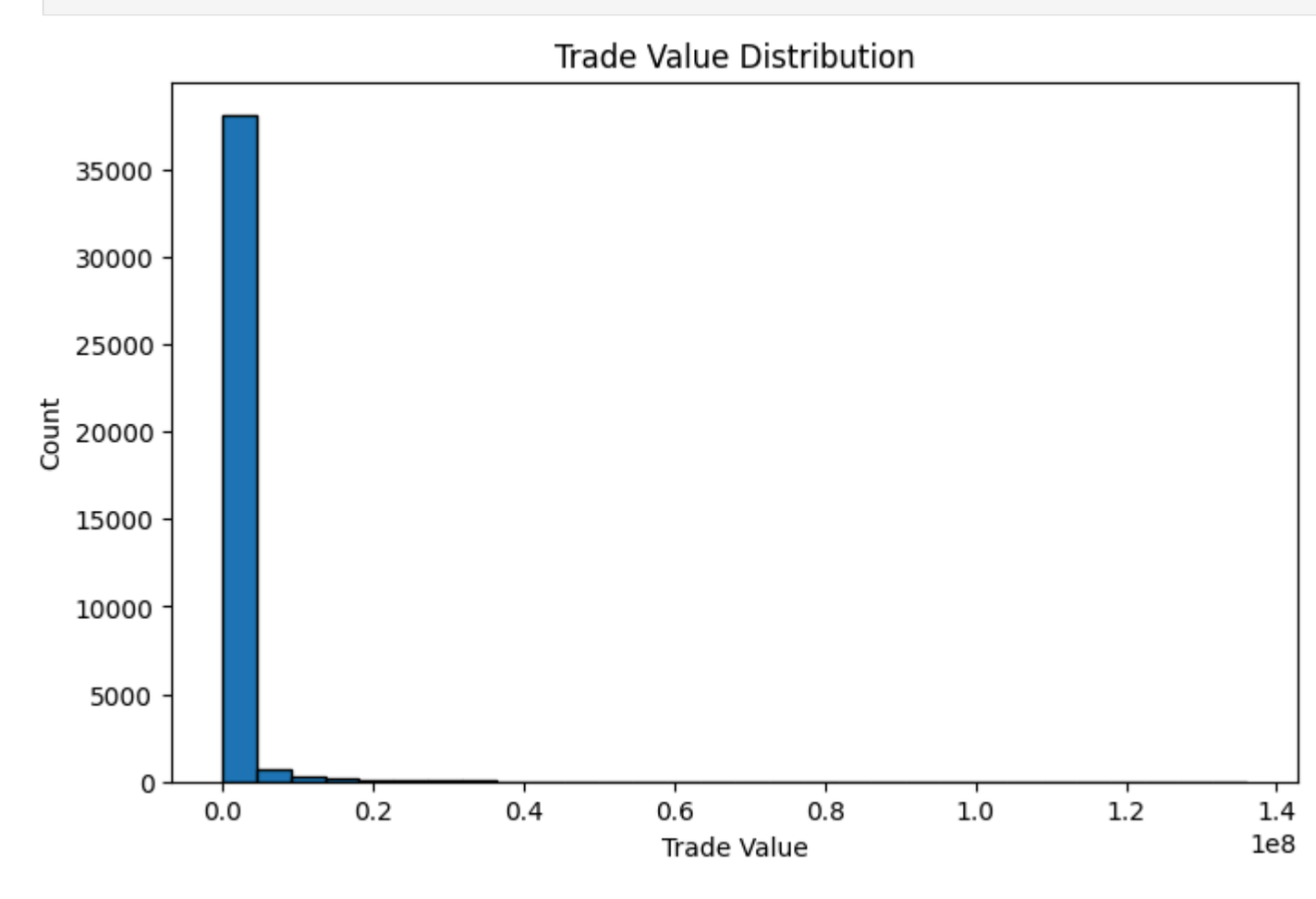
In [65]:
```
# --- 2.2 Descriptive Analytics ---
figures = []

# Exporter distribution
fig = plt.figure(figsize=(6,6))
df['exporter'].value_counts().head(10).plot.pie(autopct='%1.1f%%')
plt.title("Top Exporters Distribution (Top 10)")
figures.append(fig)
```



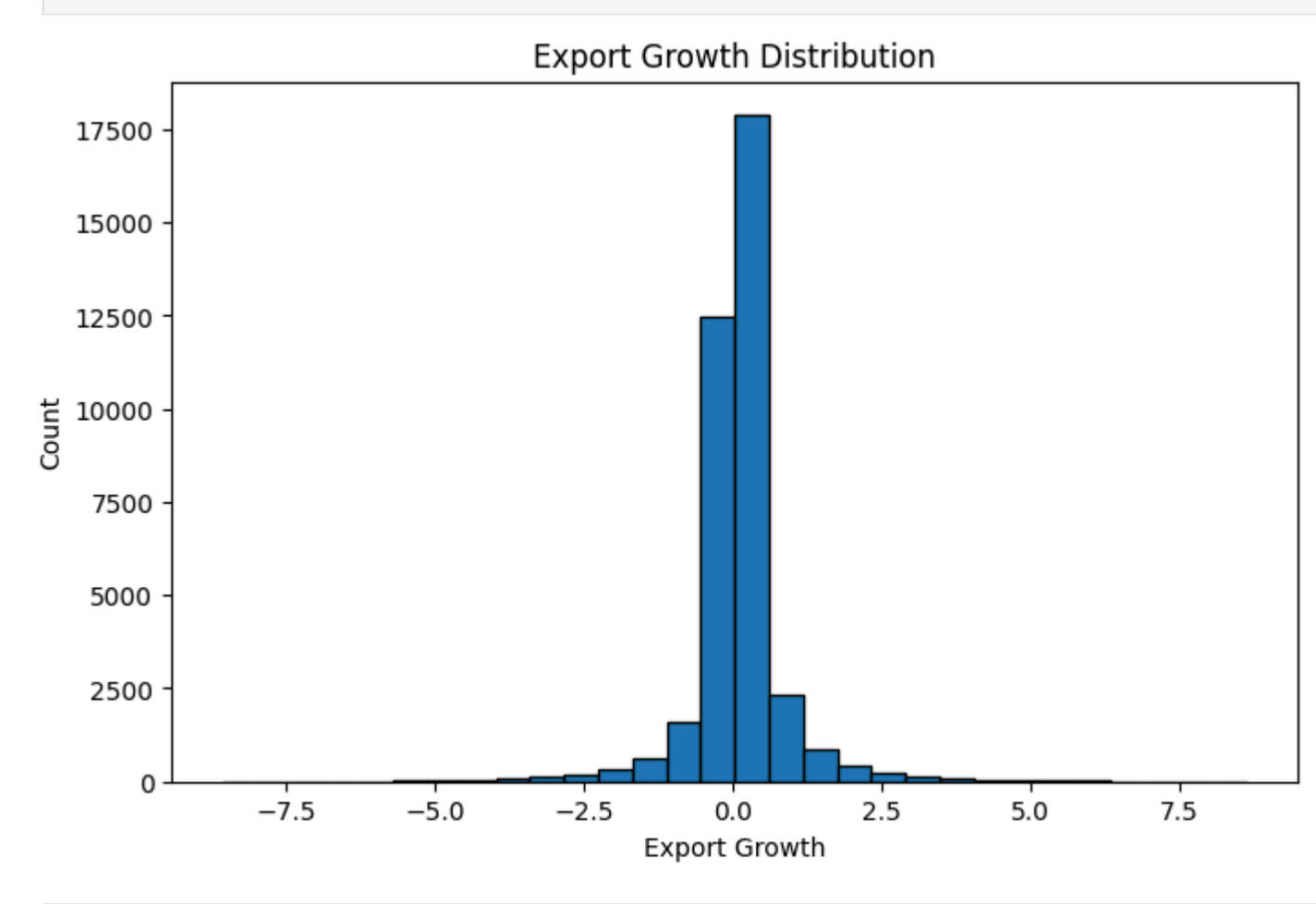
In [66]:
```
# Product distribution
fig = plt.figure(figsize=(10,5))
df['product'].value_counts().head(10).plot(kind='bar')
plt.title("Top Products Distribution (Top 10)")
plt.xlabel("Product"); plt.ylabel("Count")
figures.append(fig)
```



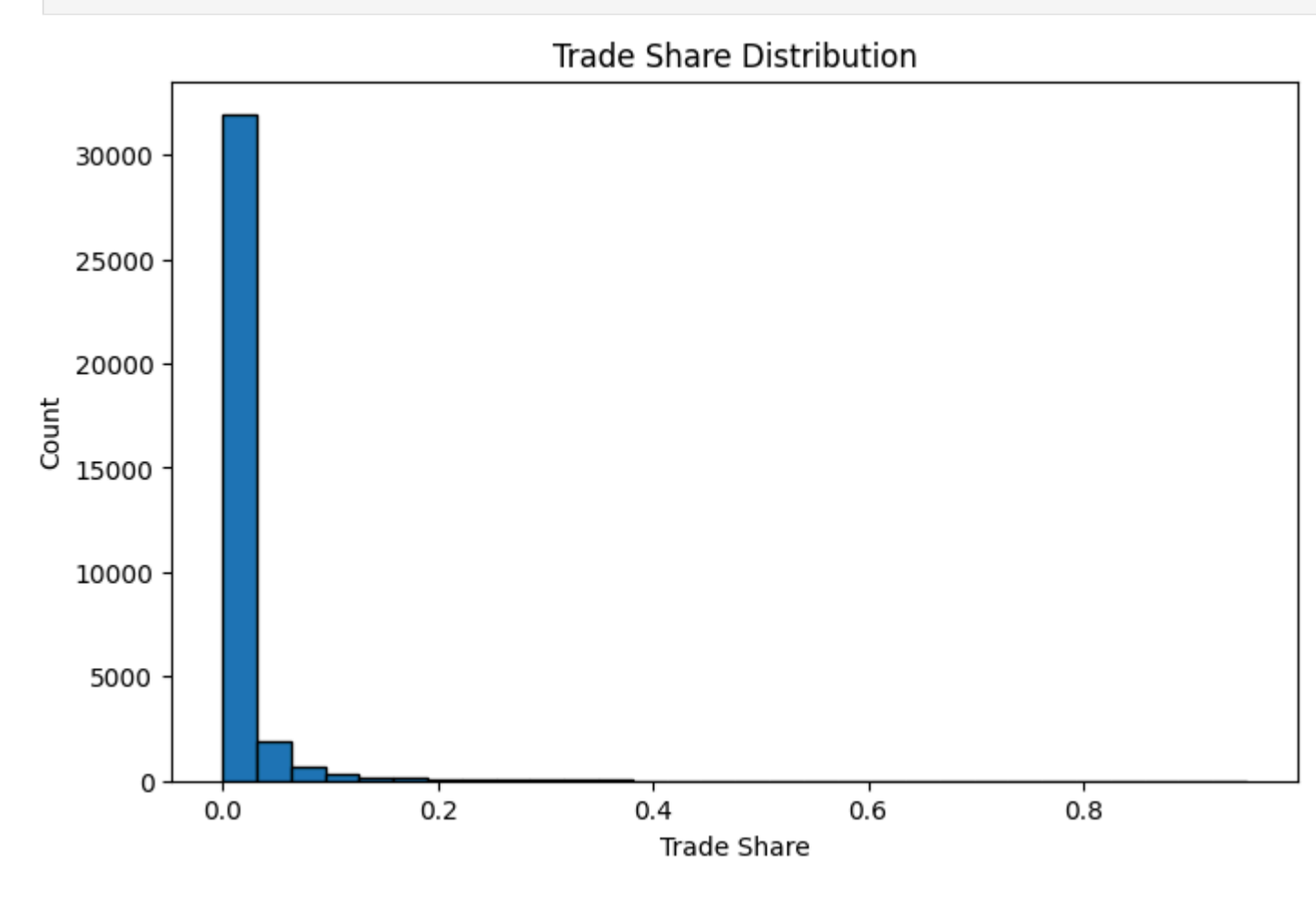
In [67]:
```
# Trade value distribution
fig = plt.figure(figsize=(8,5))
plt.hist(df['tradevalue'].dropna(), bins=30, edgecolor='black')
plt.title("Trade Value Distribution")
plt.xlabel("Trade Value"); plt.ylabel("Count")
figures.append(fig)
```



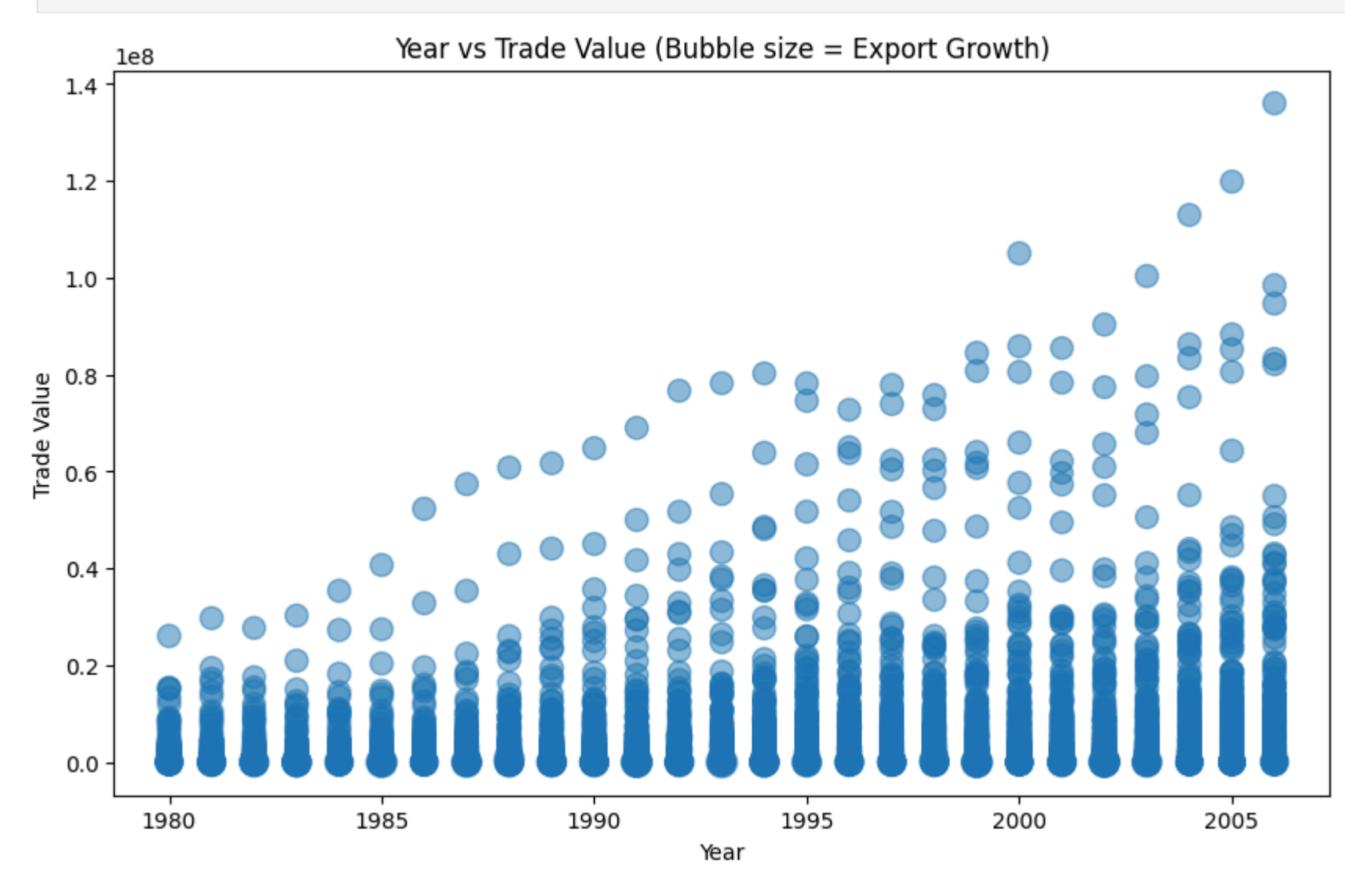
In [68]:
```
# Export growth distribution
fig = plt.figure(figsize=(8,5))
plt.hist(df['expgrowth'].dropna(), bins=30, edgecolor='black')
plt.title("Export Growth Distribution")
plt.xlabel("Export Growth"); plt.ylabel("Count")
figures.append(fig)
```



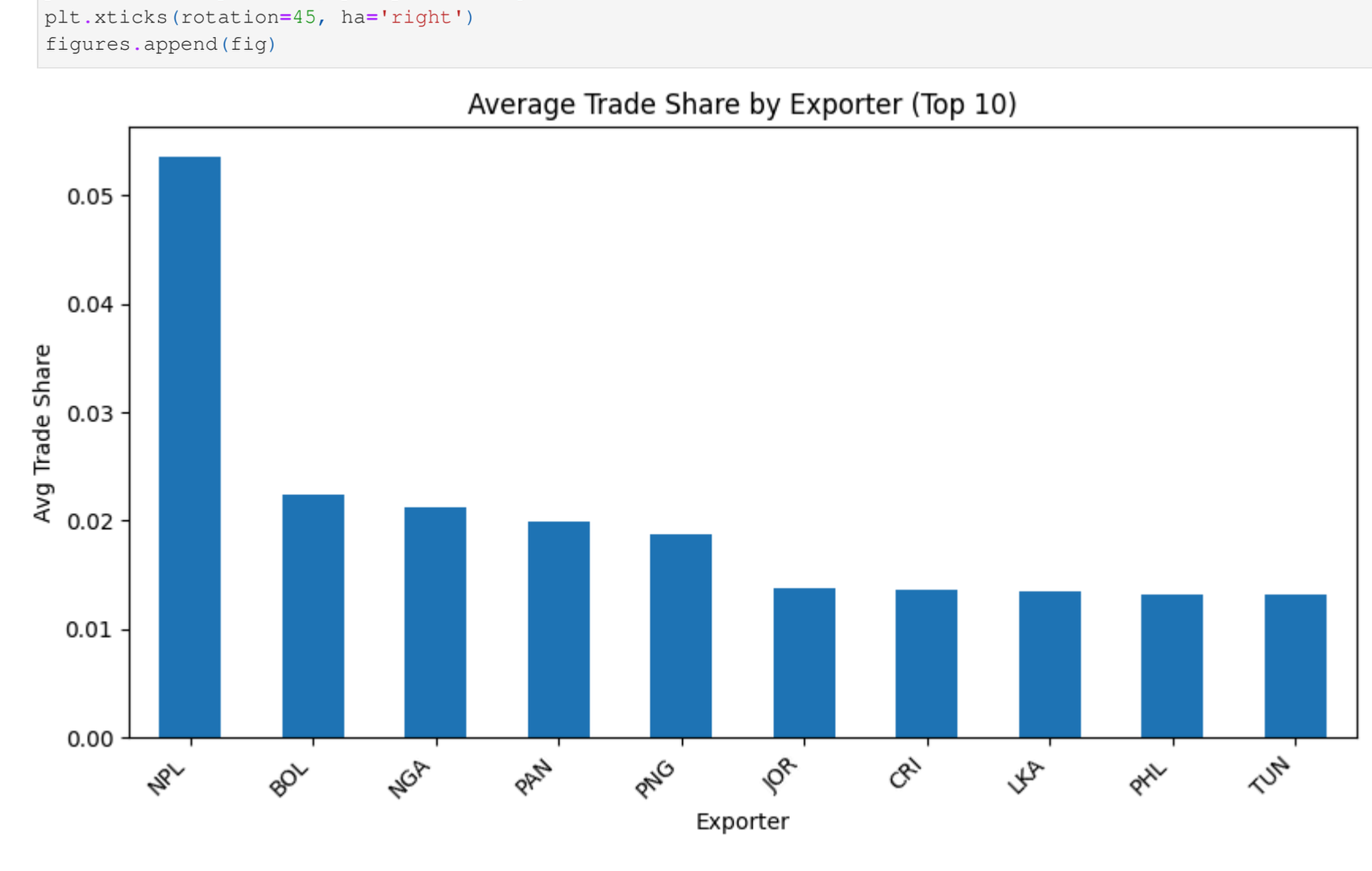
In [69]:
```
# Trade share distribution
fig = plt.figure(figsize=(8,5))
plt.hist(df['tradeshare'].dropna(), bins=30, edgecolor='black')
plt.title("Trade Share Distribution")
plt.xlabel("Trade Share"); plt.ylabel("Count")
figures.append(fig)
```



In [70]:
```
# --- 2.3 Relationship Analysis ---

# Year vs Trade Value (bubble size = export growth)
sub = df[['year','tradevalue','expgrowth']].dropna()
fig = plt.figure(figsize=(10,6))
sizes = (sub['expgrowth'] - sub['expgrowth'].min() + 1) / (sub['expgrowth'].max() - sub['expgrowth'].min() + 1) * 200
plt.scatter(sub['year'], sub['tradevalue'], s=sizes, alpha=0.5)
plt.title("Year vs Trade Value (Bubble size = Export Growth)")
plt.xlabel("Year"); plt.ylabel("Trade Value")
figures.append(fig)
```

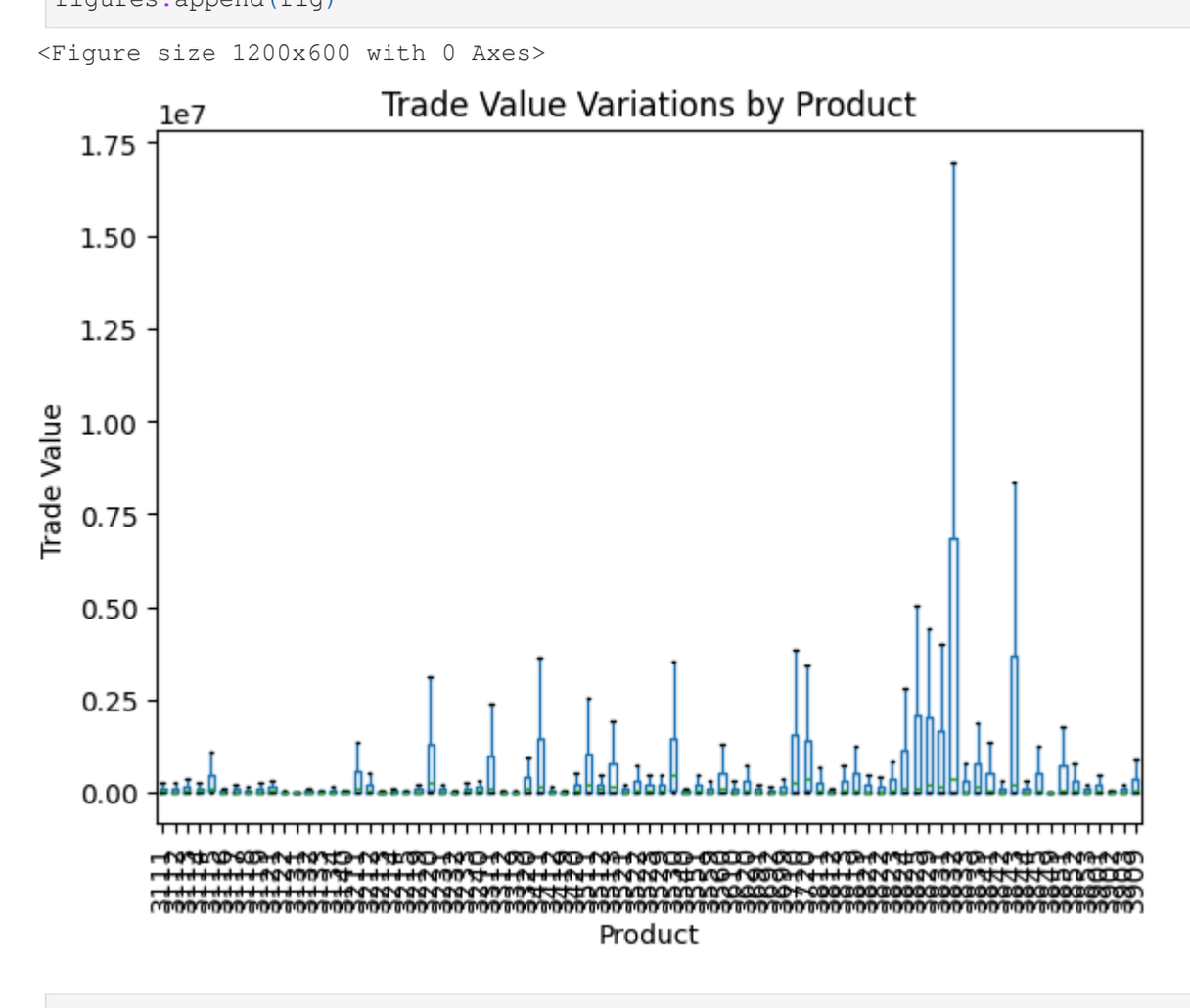


In [71]:
```
# Exporter vs Avg Trade Share
fig = plt.figure(figsize=(10,5))
df.groupby('exporter')['tradeshare'].mean().nlargest(10).plot(kind='bar')
plt.title("Average Trade Share by Exporter (Top 10)")
plt.xlabel("Exporter"); plt.ylabel("Avg Trade Share")
plt.xticks(rotation=45, ha='right')
figures.append(fig)
```

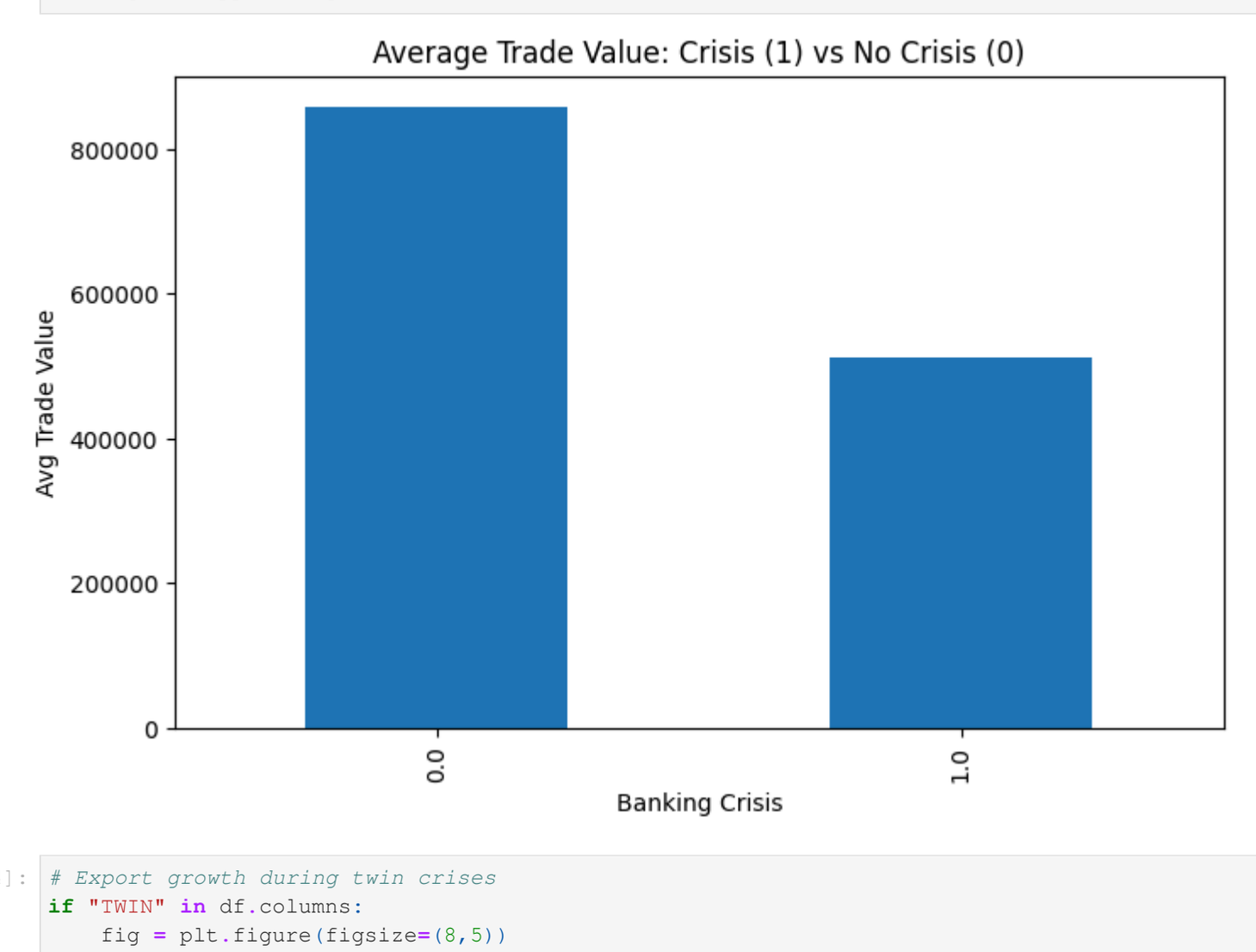


In [72]:
```
fig = plt.figure(figsize=(12,6))
df.boxplot(column='tradevalue', by='product', grid=False, showfliers=False)
plt.title("Trade Value Variations by Product")
plt.suptitle("")
plt.xlabel("Product"); plt.ylabel("Trade Value")
plt.xticks(rotation=90)
figures.append(fig)
```

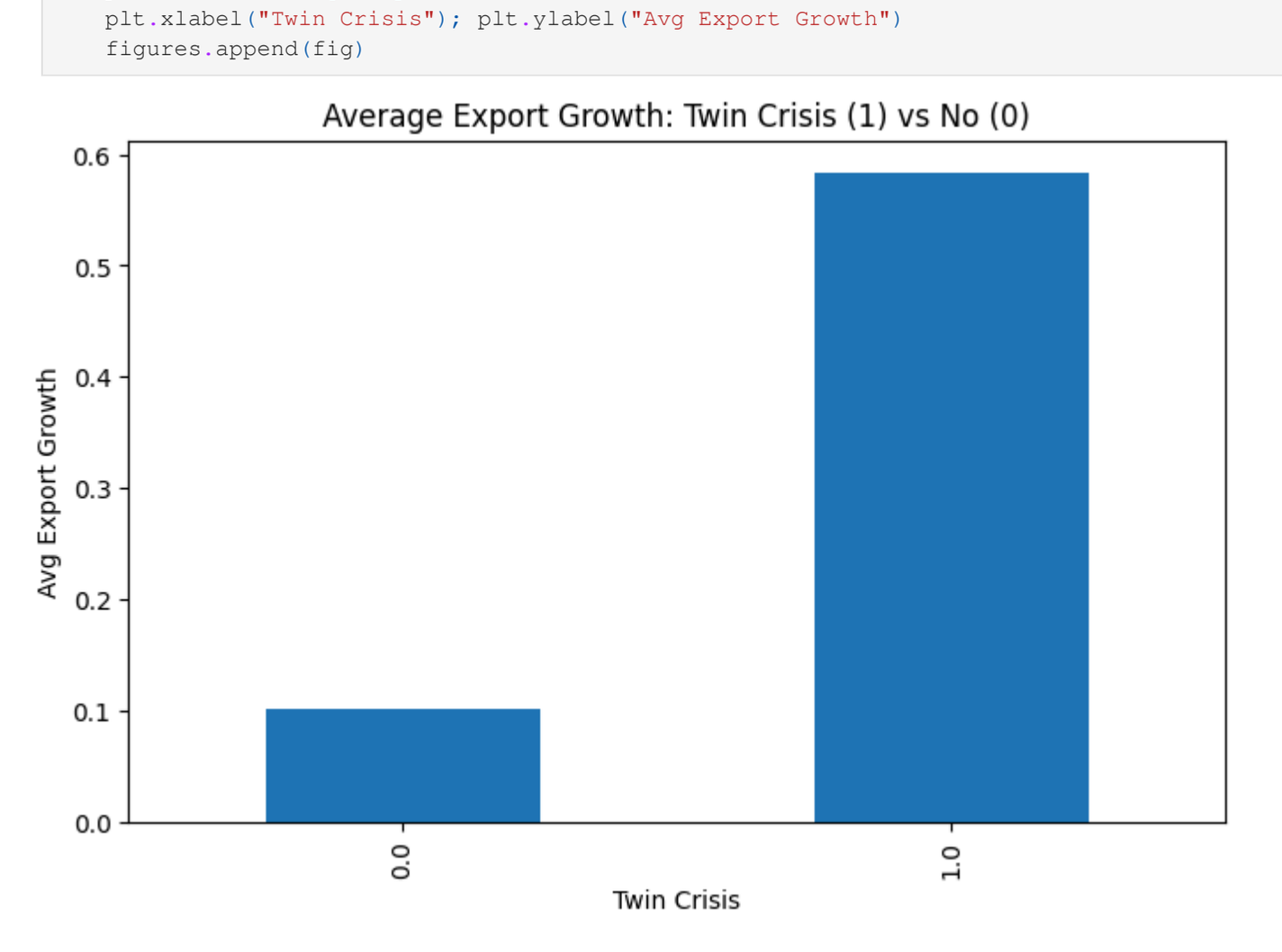
```
<Figure size 1200x600 with 0 Axes>
```



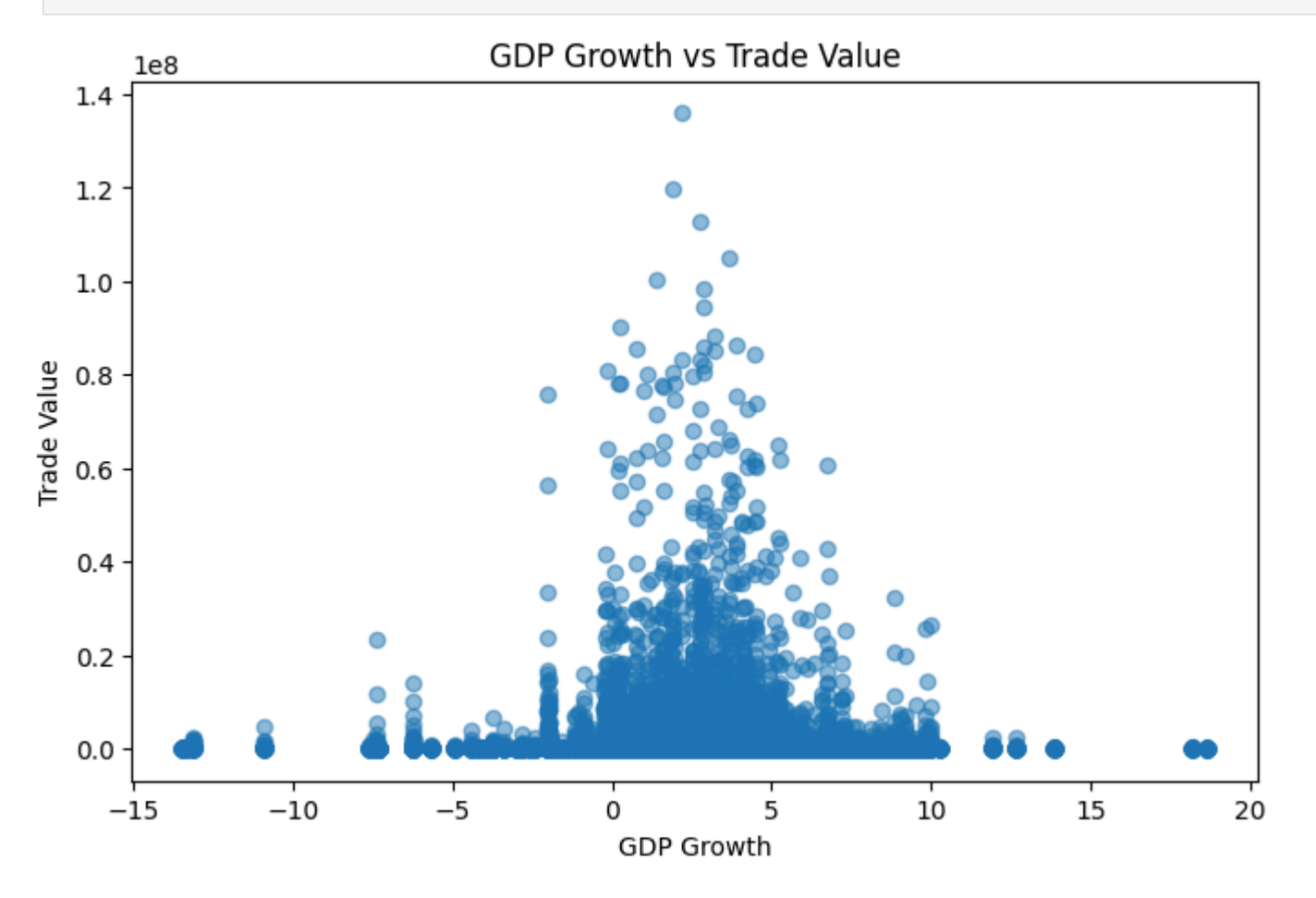
In [73]:
```
# --- Crisis-related analysis ---

# Trade value before vs during banking crises
if "BANK" in df.columns:
    fig = plt.figure(figsize=(8,5))
    df.groupby("BANK")['tradevalue'].mean().plot(kind='bar')
    plt.title("Average Trade Value: Crisis (1) vs No Crisis (0)")
    plt.xlabel("Banking Crisis"); plt.ylabel("Avg Trade Value")
    figures.append(fig)
```



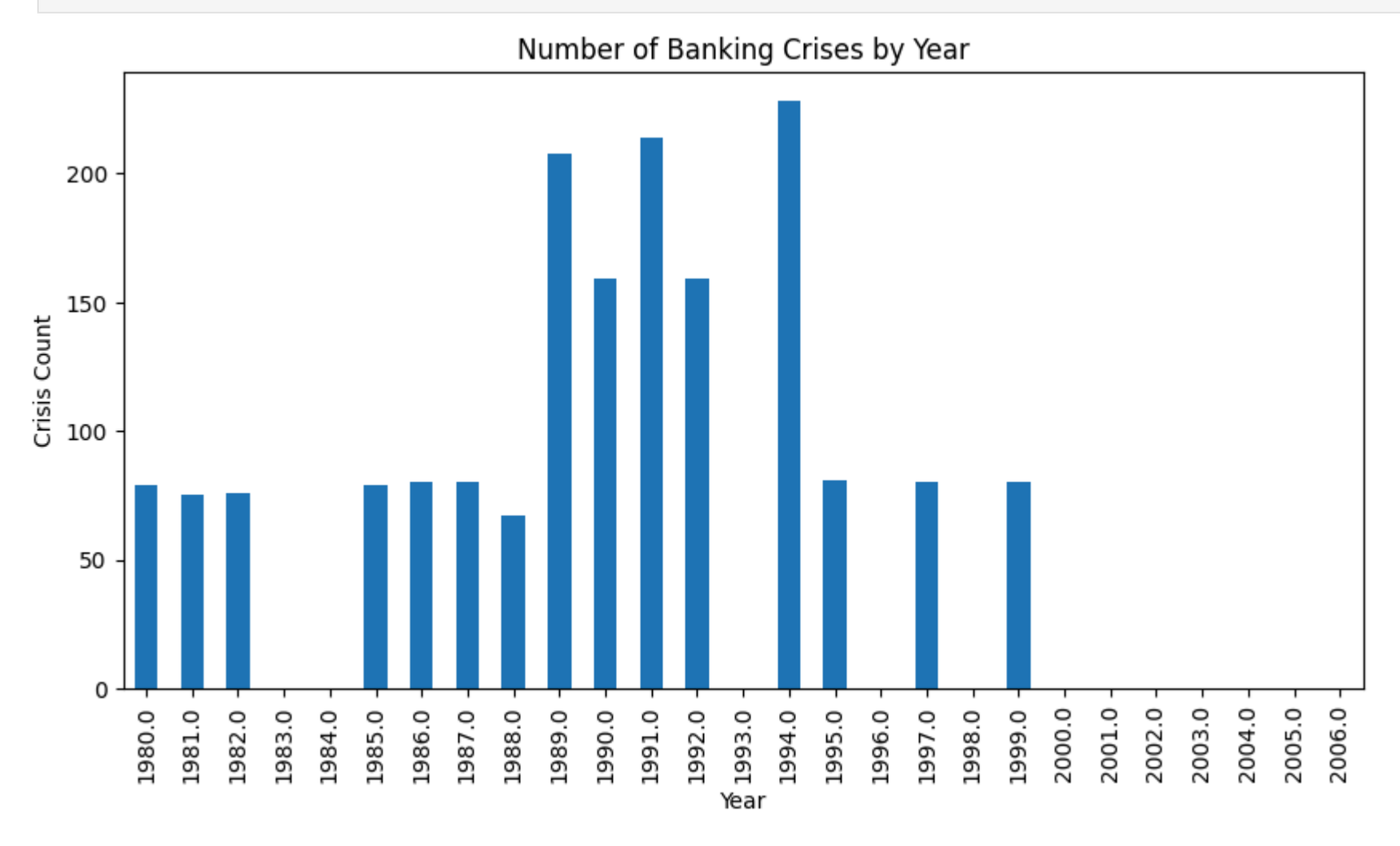
In [74]:
```
# Export growth during twin crises
if "TWIN" in df.columns:
    fig = plt.figure(figsize=(8,5))
    df.groupby("TWIN")['expgrowth'].mean().plot(kind='bar')
    plt.title("Average Export Growth: Twin Crisis (1) vs No (0)")
    plt.xlabel("Twin Crisis"); plt.ylabel("Avg Export Growth")
    figures.append(fig)
```



In [75]:
```
# GDP growth vs Trade Value
if "GDPgr" in df.columns:
    fig = plt.figure(figsize=(8,5))
    plt.scatter(df['GDPgr'], df['tradevalue'], alpha=0.5)
    plt.title("GDP Growth vs Trade Value")
    plt.xlabel("GDP Growth"); plt.ylabel("Trade Value")
    figures.append(fig)
```



In [76]:
```
# Crisis frequency over time
if "BANK" in df.columns:
    fig = plt.figure(figsize=(10,5))
    df.groupby("year")['BANK'].sum().plot(kind='bar')
    plt.title("Number of Banking Crises by Year")
    plt.xlabel("Year"); plt.ylabel("Crisis Count")
    figures.append(fig)
```



In [79]:
```
insights = [
    f"Dataset: Banking Crises and Exports",
    f"Total records: {n_records}",
    "",
    "Key Insights:",
    "- Export growth shows high volatility, with downturns during crisis years.",
    "- Trade value distribution is skewed, with a few exporters dominating trade.",
    "- Average trade value falls significantly during banking crises (BANK=1).",
    "- Export growth is lower in twin crises (TWIN=1) compared to non-crisis periods.",
    "- GDP growth shows positive correlation with trade values.",
    "",
    "Recommendations:",
    "- Strengthen financial safety nets during crises to stabilize trade flows.",
    "- Diversify export products to reduce vulnerability to shocks.",
    "- Monitor twin crises closely, as they amplify negative impacts on exports.",
    "- Encourage exporters in smaller economies to reduce concentration risks."
]
```

In [81]:
```
with PdfPages("BankingCrises_Exports_Report.pdf") as pdf:
    for fig in figures:
        pdf.savefig(fig)
        plt.close(fig)

print("\nFull analysis complete. Report saved as 'BankingCrises_Exports_Report.pdf'")
```

```
Full analysis complete. Report saved as 'BankingCrises_Exports_Report.pdf'
```

In [ ]: