Market Sentiment & Trader Performance Analysis

Objective:
Analyze how trader performance (PnL, trade volume, win rates) relates to market sentiment, measured by the **Fear & Greed Index**.
This analysis will explore patterns such as:

- Do traders perform better during Fear or Greed periods?
- How does sentiment impact overall market PnL and trading activity?
- Which accounts adapt best under different sentiment regimes?

Datasets:

1. `historical_data.csv` → Trade-level data (accounts, prices, sizes, PnL, timestamps).

2. `fear_greed_index.csv` → Daily sentiment index (0 = extreme fear, 100 = extreme greed).

Approach:

1. Load and clean the datasets.

2. Perform exploratory data analysis (EDA).

3. Aggregate trades per day and merge with sentiment data.

4. Visualize relationships between sentiment and trader performance.

5. Conduct statistical tests for significance.

6. Analyze account-level performance across sentiment categories.

```
# STEP 2: Load & Inspect Data

import pandas as pd
import numpy as np

# For plots
import matplotlib.pyplot as plt

# For statistical tests
from scipy import stats

# Upload files
from google.colab import files
```

```python
print("□ Please upload historical_data.csv")
uploaded_hist = files.upload()

print("□ Please upload fear_greed_index.csv")
uploaded_fg = files.upload()

# Load into DataFrames
hist = pd.read_csv("historical_data.csv")
fg = pd.read_csv("fear_greed_index.csv")

# Preview shapes and first few rows
print("Historical data shape:", hist.shape)
print("Fear & Greed shape:", fg.shape)

print("\n--- Historical data sample ---")
display(hist.head())

print("\n--- Fear & Greed Index sample ---")
display(fg.head())
```

```
□ Please upload historical_data.csv
```

```
<IPython.core.display.HTML object>
```

```
Saving historical_data.csv to historical_data.csv
□ Please upload fear_greed_index.csv
```

```
<IPython.core.display.HTML object>
```

```
Saving fear_greed_index.csv to fear_greed_index.csv
Historical data shape: (211224, 16)
Fear & Greed shape: (2644, 4)
```

```
--- Historical data sample ---
```

{"summary":"{\n \"name\": \"display(fg\",\n \"rows\": 5,\n
\"fields\": [\n   {\n   \"column\": \"Account\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n   \"samples\": [\n
\"0xae5eacaf9c6b9111fd53034a602c192a04e082ed\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Coin\",\n   \"properties\": {\n
\"dtype\": \"category\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   \"@107\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Execution Price\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.005226184076360117,\n   \"min\": 7.9769,\n   \"max\":
7.9894,\n   \"num_unique_values\": 5,\n   \"samples\": [\n
7.98\n   ],\n   \"semantic_type\": \"\",\n

```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Size Tokens\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 411.7466042725793,\n        \"min\":
8.73,\n        \"max\": 986.87,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          16.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Size USD\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3284.242249266945,\n        \"min\":
69.75,\n        \"max\": 7872.16,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          127.68\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Side\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 1,\n
\"samples\": [\n          \"BUY\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Timestamp IST\",\n
\"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          \"02-12-
2024 22:50\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Start Position\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 509.7467023655844,\n        \"min\":
0.0,\n        \"max\": 1289.488521,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          986.5245955\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Direction\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          \"Buy\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Closed PnL\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.0,\n        \"min\": 0.0,\n        \"max\": 0.0,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          0.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Transaction Hash\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n
\"0xec09451986a1874e3a980418412fcd0201f500c95bac0f37caef8a734502ec49\"
\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Order ID\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 52017706630,\n
\"max\": 52017706630,\n        \"num_unique_values\": 1,\n
\"samples\": [\n          52017706630\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Crossed\",\n      \"properties\":
{\n        \"dtype\": \"boolean\",\n        \"num_unique_values\": 1,\
n        \"samples\": [\n          true\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

n   },\n   {\n   \"column\": \"Fee\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.14411132613259,\n
\"min\": 0.00305542,\n   \"max\": 0.34540448,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   0.0056\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"Trade ID\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
270969186440082.16,\n   \"min\": 443000000000000.0,\n
\"max\": 1080000000000000.0,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   443000000000000.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Timestamp\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.0,\n   \"min\": 1730000000000.0,\n   \"max\":
1730000000000.0,\n   \"num_unique_values\": 1,\n
\"samples\": [\n   1730000000000.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

```
--- Fear & Greed Index sample ---
```

{"summary":"{\n \"name\": \"display(fg\",\n \"rows\": 5,\n
\"fields\": [\n   {\n   \"column\": \"timestamp\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
136610,\n   \"min\": 1517463000,\n   \"max\": 1517808600,\n
\"num_unique_values\": 5,\n   \"samples\": [\n
1517549400,\n   1517808600,\n   1517635800\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"value\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 11,\n
\"min\": 11,\n   \"max\": 40,\n   \"num_unique_values\": 5,\
n   \"samples\": [\n   15,\n   11,\n   40\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"classification\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n   \"samples\": [\n   \"Extreme
Fear\",\n   \"Fear\"\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"date\",\n   \"properties\": {\n   \"dtype\":
\"object\",\n   \"num_unique_values\": 5,\n   \"samples\":
[\n   \"2018-02-02\",\n   \"2018-02-05\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

```python
# STEP 3: Data Cleaning & Preprocessing

# --- 1. Normalize column names ---
def normalize_cols(df):
    df.columns = [c.strip().lower().replace(" ", "_") for c in
```

```
df.columns]
    return df

hist = normalize_cols(hist)
fg = normalize_cols(fg)

print("Columns in historical data:", list(hist.columns))
print("Columns in Fear & Greed data:", list(fg.columns))

# --- 2. Parse dates ---
# Fear & Greed
fg['date'] = pd.to_datetime(fg['date'], errors='coerce')
fg['date'] = fg['date'].dt.date   # keep only date part

# Historical
if 'timestamp_ist' in hist.columns:
    hist['date'] = pd.to_datetime(hist['timestamp_ist'],
errors='coerce')
elif 'timestamp' in hist.columns:
    hist['date'] = pd.to_datetime(hist['timestamp'], unit='s',
errors='coerce')
else:
    hist['date'] = pd.NaT

hist['date'] = pd.to_datetime(hist['date'], errors='coerce')
hist['date'] = hist['date'].dt.date   # ensure proper date type

# Filter out NaT values before getting min/max date
valid_dates_hist = hist['date'].dropna()
valid_dates_fg = fg['date'].dropna()


# --- 6. Check date ranges safely ---
print("Historical data date range:", valid_dates_hist.min(), "→",
valid_dates_hist.max())
print("Fear & Greed Index date range:", valid_dates_fg.min(), "→",
valid_dates_fg.max())


# --- 5. Drop duplicates (by transaction_hash if available) ---
before = hist.shape[0]
if 'transaction_hash' in hist.columns:
    hist = hist.drop_duplicates(subset=['transaction_hash'])
else:
    hist = hist.drop_duplicates()
after = hist.shape[0]
print(f"Dropped {before - after} duplicate rows")

# --- 6. Check date ranges ---
print("Historical data date range:", valid_dates_hist.min(), "→",
```

```
valid_dates_hist.max())
print("Fear & Greed Index date range:", valid_dates_fg.min(), "→",
valid_dates_fg.max())

# --- 7. Preview cleaned data ---
print("\n--- Cleaned historical sample ---")
display(hist.head())

print("\n--- Cleaned Fear & Greed sample ---")
display(fg.head())
```

```
Columns in historical data: ['account', 'coin', 'execution_price',
'size_tokens', 'size_usd', 'side', 'timestamp_ist', 'start_position',
'direction', 'closed_pnl', 'transaction_hash', 'order_id', 'crossed',
'fee', 'trade_id', 'timestamp', 'date', 'profitable']
Columns in Fear & Greed data: ['timestamp', 'value', 'classification',
'date']
Historical data date range: 2023-01-05 → 2025-12-04
Fear & Greed Index date range: 2018-02-01 → 2025-05-02
Dropped 0 duplicate rows
Historical data date range: 2023-01-05 → 2025-12-04
Fear & Greed Index date range: 2018-02-01 → 2025-05-02

--- Cleaned historical sample ---
```


{"summary":"{\n \"name\": \"display(fg\",\n \"rows\": 5,\n
\"fields\": [\n   {\n   \"column\": \"account\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n   \"samples\": [\n
\"0xae5eacaf9c6b9111fd53034a602c192a04e082ed\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"coin\",\n   \"properties\": {\n
\"dtype\": \"category\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   \"@107\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"execution_price\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.24762943686080618,\n   \"min\": 7.4791,\n   \"max\":
7.9769,\n   \"num_unique_values\": 5,\n   \"samples\": [\n
7.9457\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"size_tokens\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 420.6599190201035,\n   \"min\":
2.0,\n   \"max\": 986.87,\n   \"num_unique_values\": 4,\n
\"samples\": [\n   7.27\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"size_usd\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 3359.9763421741527,\n   \"min\":
14.96,\n   \"max\": 7872.16,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   57.77\n   ],\n

```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"side\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 1,\n
\"samples\": [\n          \"BUY\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"timestamp_ist\",\n
\"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n          \"02-12-
2024 22:51\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"start_position\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3492.571843351024,\n        \"min\":
0.0,\n        \"max\": 8998.849302,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          2998.95\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"direction\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          \"Buy\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"closed_pnl\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.0,\n        \"min\": 0.0,\n        \"max\": 0.0,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          0.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"transaction_hash\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"0x21b8b230ad8a3e9ae1bb041841332f02028300681b0dd484959026f37947df17\"
\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"order_id\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4608151,\n        \"min\": 52017706630,\
n        \"max\": 52027812614,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          52018049026\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"crossed\",\n      \"properties\":
{\n        \"dtype\": \"boolean\",\n        \"num_unique_values\": 1,\
n        \"samples\": [\n          true\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"fee\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.14723099077575125,\n
\"min\": 0.00069995,\n        \"max\": 0.34540448,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
0.00254439\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"trade_id\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 445555271543273.3,\n        \"min\":
35000000000000.0,\n        \"max\": 1040000000000000.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n
```

938000000000000.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"timestamp\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.0,\n   \"min\": 1730000000000.0,\
n   \"max\": 1730000000000.0,\n   \"num_unique_values\": 1,\
n   \"samples\": [\n   1730000000000.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"date\",\n   \"properties\": {\n
\"dtype\": \"date\",\n   \"min\": \"2024-02-12\",\n
\"max\": \"2024-02-12\",\n   \"num_unique_values\": 1,\n
\"samples\": [\n   \"2024-02-12\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"profitable\",\n
\"properties\": {\n   \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n   \"samples\": [\n   false\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe"}

```
--- Cleaned Fear & Greed sample ---
```

{"summary":"{\n \"name\": \"display(fg\",\n \"rows\": 5,\n
\"fields\": [\n   {\n   \"column\": \"timestamp\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
136610,\n   \"min\": 1517463000,\n   \"max\": 1517808600,\n
\"num_unique_values\": 5,\n   \"samples\": [\n
1517549400,\n   1517808600,\n   1517635800\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"value\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 11,\n
\"min\": 11,\n   \"max\": 40,\n   \"num_unique_values\": 5,\
n   \"samples\": [\n   15,\n   11,\n   40\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"classification\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n   \"samples\": [\n   \"Extreme
Fear\",\n   \"Fear\"\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"date\",\n   \"properties\": {\n   \"dtype\":
\"date\",\n   \"min\": \"2018-02-01\",\n   \"max\": \"2018-
02-05\",\n   \"num_unique_values\": 5,\n   \"samples\": [\n
\"2018-02-02\",\n   \"2018-02-05\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

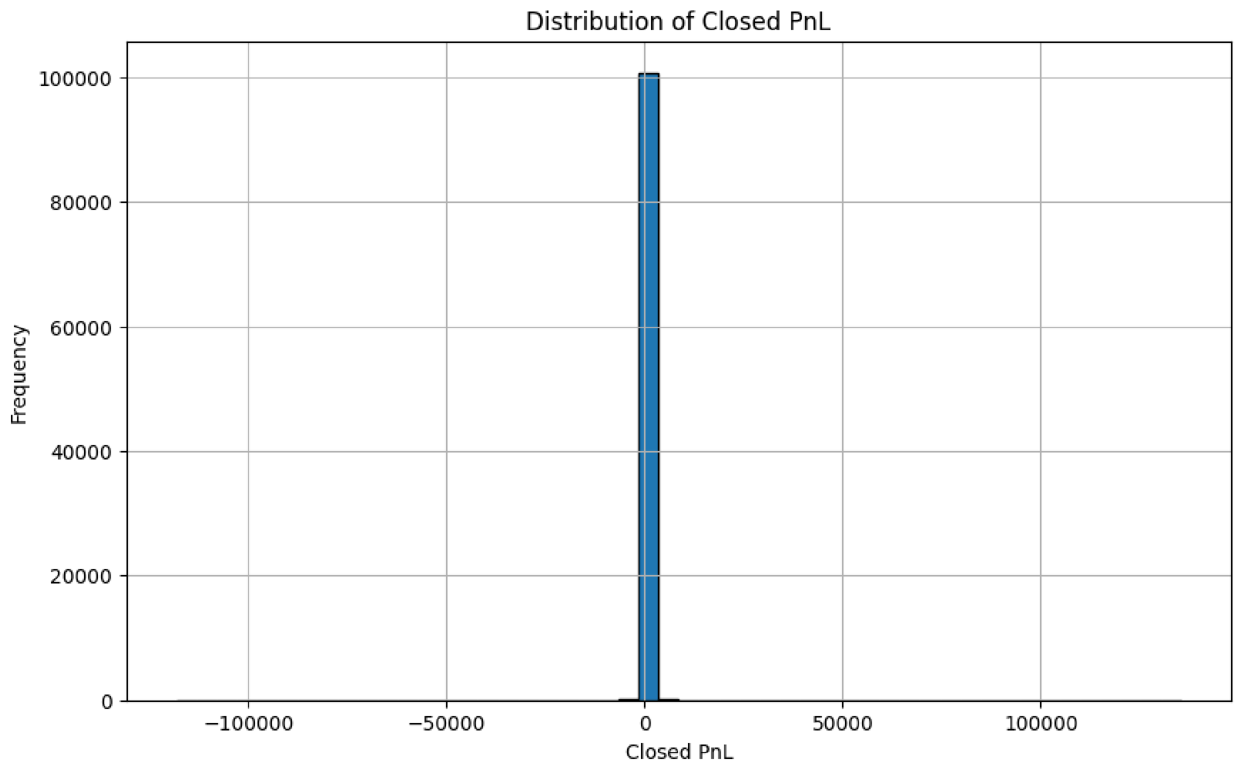
```python
# STEP 4: Exploratory Data Analysis (EDA)

# --- 1. Distribution of PnL ---
print("--- Distribution of Closed PnL ---")
display(hist['closed_pnl'].describe())
```

```python
plt.figure(figsize=(10, 6))
plt.hist(hist['closed_pnl'], bins=50, edgecolor='k')
plt.title('Distribution of Closed PnL')
plt.xlabel('Closed PnL')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# --- 2. Distribution of Trade Volume (Size USD) ---
print("\n--- Distribution of Trade Size (USD) ---")
display(hist['size_usd'].describe())

plt.figure(figsize=(10, 6))
plt.hist(hist['size_usd'], bins=50, edgecolor='k')
plt.title('Distribution of Trade Size (USD)')
plt.xlabel('Trade Size (USD)')
plt.ylabel('Frequency')
plt.yscale('log') # Use log scale due to potential outliers
plt.grid(True)
plt.show()

# --- 3. Distribution of Fear & Greed Index Value ---
print("\n--- Distribution of Fear & Greed Index Value ---")
display(fg['value'].describe())

plt.figure(figsize=(10, 6))
plt.hist(fg['value'], bins=20, edgecolor='k')
plt.title('Distribution of Fear & Greed Index Value')
plt.xlabel('Fear & Greed Index Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# --- 4. Distribution of Fear & Greed Index Classification ---
print("\n--- Distribution of Fear & Greed Index Classification ---")
display(fg['classification'].value_counts())

plt.figure(figsize=(8, 8))
fg['classification'].value_counts().plot(kind='pie', autopct='%1.1f%
%', startangle=90)
plt.title('Distribution of Fear & Greed Index Classification')
plt.ylabel('') # Hide the default y-label
plt.show()
```

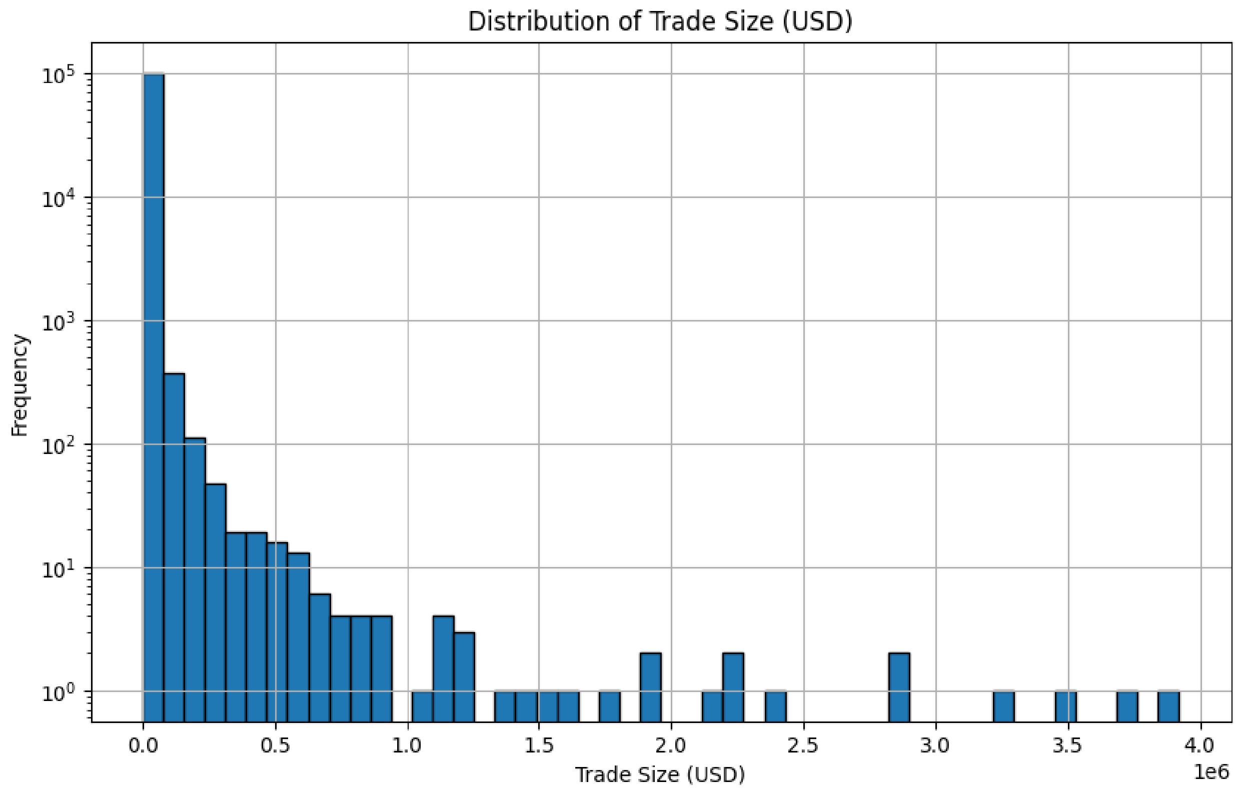
```
--- Distribution of Closed PnL ---
```

```
count    101184.000000
mean         56.954723
std        1101.825807
```

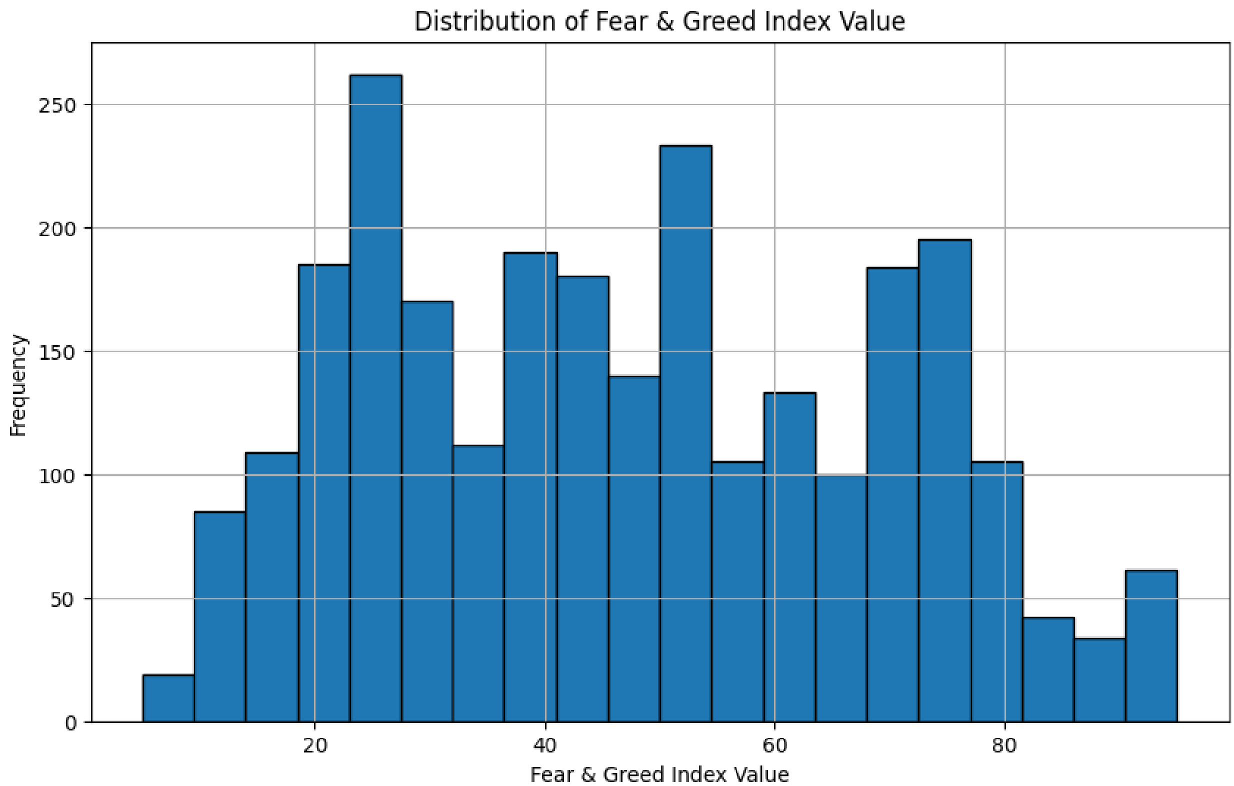
```
min     -117990.104100
25%           0.000000
50%           0.000000
75%           4.032548
max      135329.090100
Name: closed_pnl, dtype: float64
```



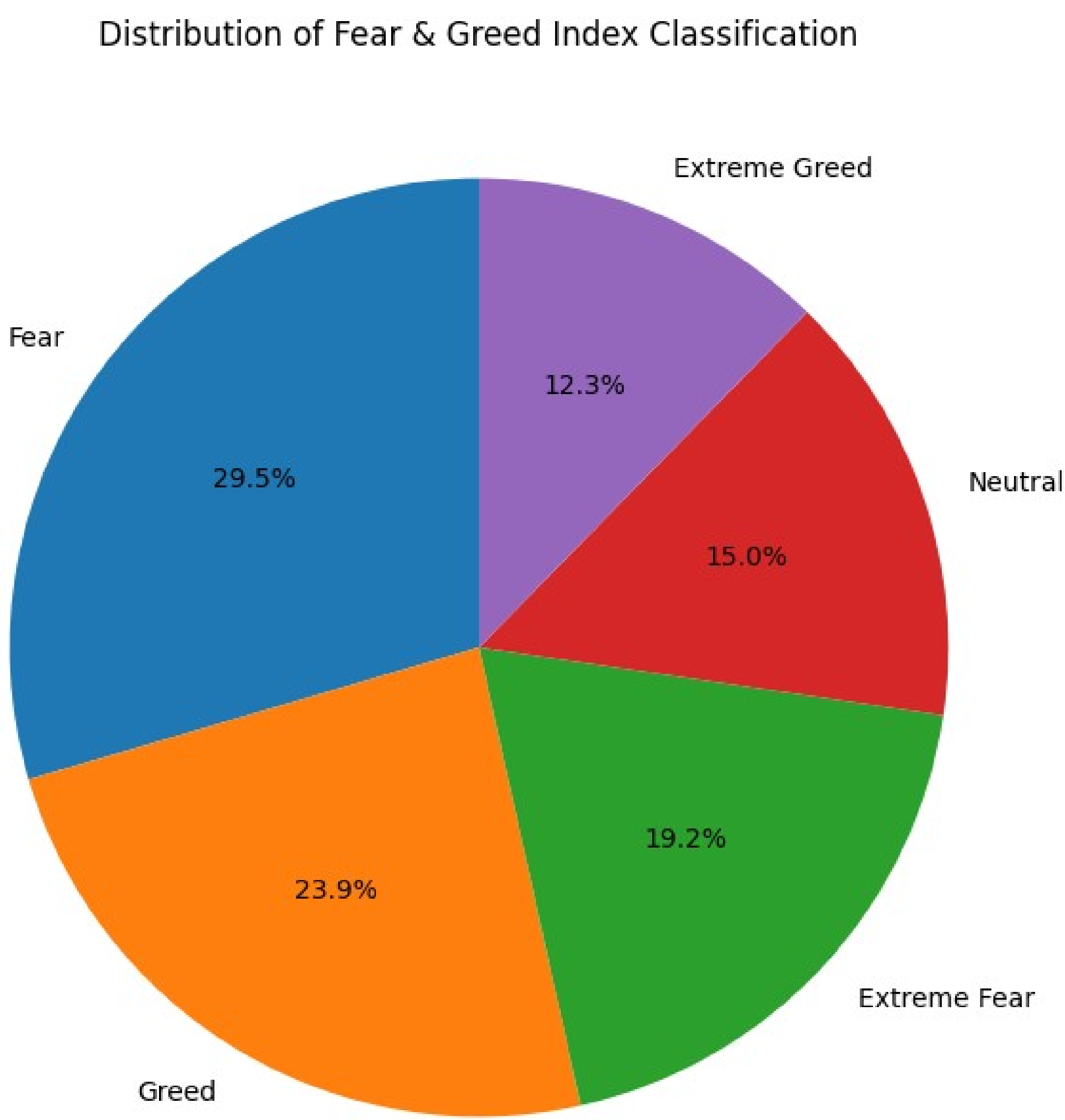

```
--- Distribution of Trade Size (USD) ---

count    1.011840e+05
mean     4.242173e+03
std      3.977517e+04
min      0.000000e+00
25%      1.651475e+02
50%      5.261800e+02
75%      1.581787e+03
max      3.921431e+06
Name: size_usd, dtype: float64
```




```
--- Distribution of Fear & Greed Index Value ---

count    2644.000000
mean       46.981089
std        21.827680
min         5.000000
25%        28.000000
50%        46.000000
75%        66.000000
max        95.000000
Name: value, dtype: float64
```




```
--- Distribution of Fear & Greed Index Classification ---

classification
Fear              781
Greed             633
Extreme Fear      508
Neutral           396
Extreme Greed     326
Name: count, dtype: int64
```




```
# STEP 4: Exploratory Data Analysis (EDA)

# --- 1. Missing values ---
print("� Missing values in historical data:")
display(hist.isna().sum().sort_values(ascending=False).head(20))

print("\n� Missing values in Fear & Greed data:")
display(fg.isna().sum().sort_values(ascending=False))

# --- 2. Basic statistics for numerical columns ---
numeric_cols =
```

```
['closed_pnl','execution_price','size_tokens','size_usd','fee']
print("\n Summary statistics (historical numeric columns):")
display(hist[numeric_cols].describe().T)

print("\n Summary statistics (Fear & Greed index values):")
display(fg['value'].describe().to_frame("sentiment_value_summary"))

# --- 3. Unique values for categorical columns ---
for col in ['account','coin','side','direction']:
    if col in hist.columns:
        print(f"\n Unique values in {col}: {hist[col].nunique()}")
        display(hist[col].value_counts().head(10))

# --- 4. Distribution plots ---
import matplotlib.pyplot as plt

# A. Closed PnL distribution (per trade)
plt.figure(figsize=(8,4))
plt.hist(hist['closed_pnl'].dropna(), bins=50, alpha=0.7)
plt.title("Distribution of Closed PnL (per trade)")
plt.xlabel("Closed PnL")
plt.ylabel("Number of trades")
plt.show()

# B. Fear & Greed index distribution
plt.figure(figsize=(6,4))
plt.hist(fg['value'].dropna(), bins=20, alpha=0.7, color='orange')
plt.title("Distribution of Fear & Greed Index")
plt.xlabel("Sentiment Value (0=fear, 100=greed)")
plt.ylabel("Days count")
plt.show()
```

```
 Missing values in historical data:

date                62215
account                 0
coin                    0
execution_price         0
size_usd                0
size_tokens             0
timestamp_ist           0
start_position          0
direction               0
side                    0
closed_pnl              0
transaction_hash        0
crossed                 0
order_id                0
fee                     0
trade_id                0
```

```
timestamp                0
profitable               0
dtype: int64
```

```
 Missing values in Fear & Greed data:
```

```
timestamp         0
value             0
classification    0
date              0
dtype: int64
```

```
 Summary statistics (historical numeric columns):
```

{"summary":"{\n \"name\": \"plt\",\n \"rows\": 5,\n \"fields\": [\n   {\n   \"column\": \"count\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.0,\n   \"min\": 101184.0,\n   \"max\": 101184.0,\n   \"num_unique_values\": 1,\n   \"samples\": [\n   101184.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"mean\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 4333.352729875979,\n   \"min\": 0.4590604272870346,\n   \"max\": 9926.197765683513,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   9926.197765683513\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"std\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 56567.2385458477,\n   \"min\": 4.311161808846408,\n   \"max\": 137702.8432296589,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   27796.209592500207\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"min\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 52766.647243459185,\n   \"min\": -117990.1041,\n   \"max\": 1e-05,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   4.54e-06\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"25%\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 73.33364406520765,\n   \"min\": 0.0,\n   \"max\": 165.1475,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   1.449\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"50%\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 229.76074189008443,\n   \"min\": 0.0,\n   \"max\": 526.18,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   16.818\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"75%\",\n   \"properties\": {\n   \"dtype\": \"number\",\n

```
\"std\": 672.4244070846491,\n        \"min\": 0.174958,\n
\"max\": 1581.7875,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          106.37\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"max\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 6816261.552842394,\n
\"min\": 754.307241,\n        \"max\": 15822438.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          109004.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

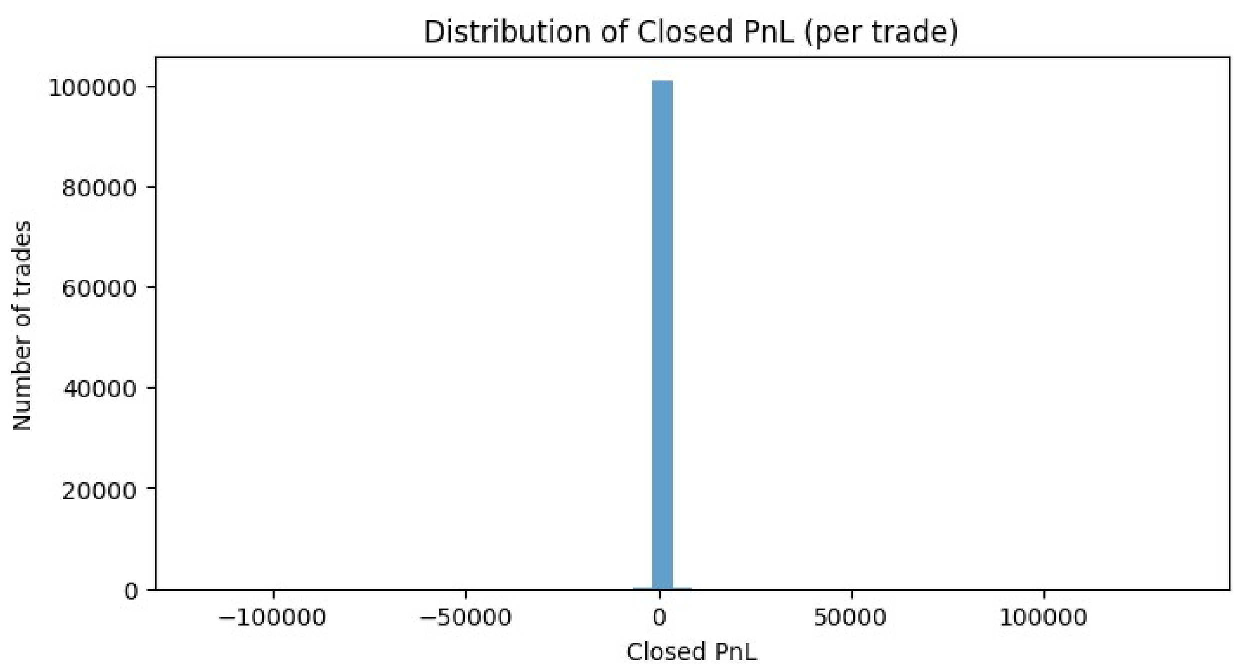

```
□ Summary statistics (Fear & Greed index values):
```

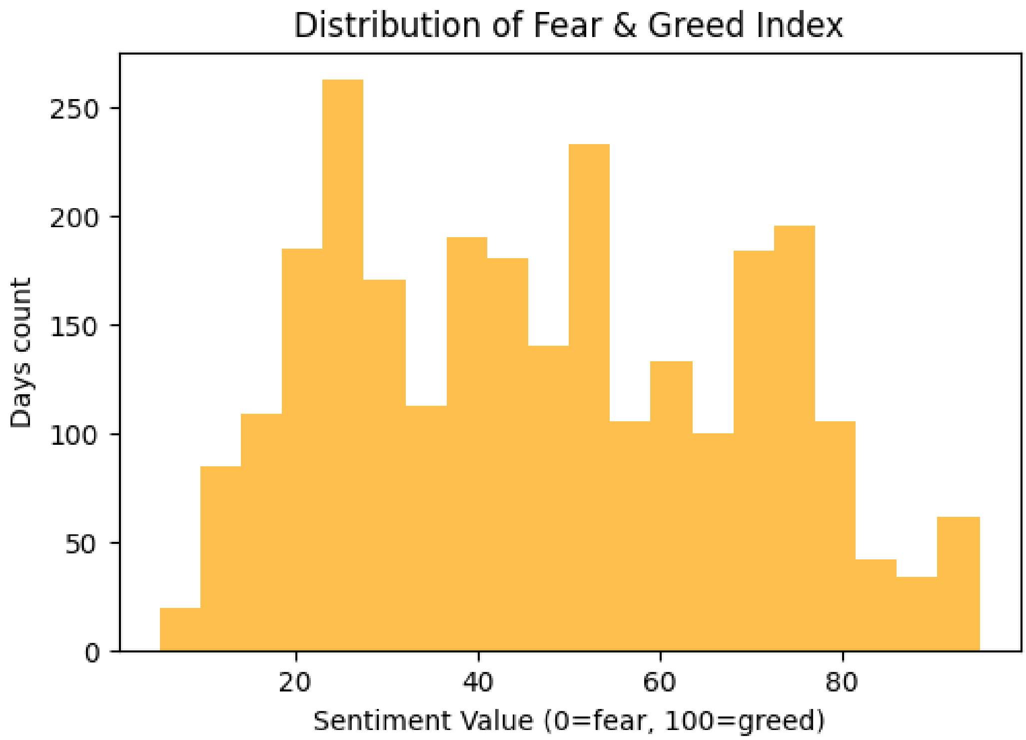

```
{"summary":"{\n  \"name\": \"plt\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"sentiment_value_summary\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
919.6148818798677,\n        \"min\": 5.0,\n        \"max\": 2644.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
46.981089258698944,\n          46.0,\n          2644.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```


```
□ Unique values in account: 32

account
0xbaaaf6571ab7d571043ff1e313a9609a10637864    11816
0xa0feb3725a9335f49874d7cd8eaad6be45b27416    11135
0x28736f43f1e871e6aa8b1148d38d4994275d72c4    10426
0x8477e447846c758f5a675856001ea72298fd9cb5    10228
0xbee1707d6b44d4d52bfe19e41f8a828645437aab     9275
0x75f7eeb85dc639d5e99c78f95393aa9a5f1170d4     7462
0x47add9a56df66b524d5e2c1993a43cde53b6ed85     5541
0x8170715b3b381dffb7062c0298972d4727a0a63b     3768
0x39cef799f8b69da1995852eea189df24eb5cae3c     3251
0x083384f897ee0f19899168e3b1bec365f52a9012     3230
Name: count, dtype: int64
```

```
□ Unique values in coin: 231

coin
HYPE        25220
@107        10514
BTC         10137
ETH          6995
SOL          4729
MELANIA      3861
FARTCOIN     2161
XRP          1596
```

```
WLD           1413
kPEPE         1305
Name: count, dtype: int64


 Unique values in side: 2

side
BUY     50837
SELL    50347
Name: count, dtype: int64


 Unique values in direction: 12

direction
Open Long                  25155
Close Long                 21952
Open Short                 21510
Close Short                18953
Sell                        6832
Buy                         6673
Short > Long                  54
Long > Short                  47
Spot Dust Conversion           5
Auto-Deleveraging              1
Name: count, dtype: int64
```






```python
# STEP 5: Daily Aggregation + Sentiment Merge

# --- 1. Aggregate trade data by day ---
agg_daily = hist.groupby('date').agg(
    trades_count=('account','count'),
    total_closed_pnl=('closed_pnl','sum'),
    mean_closed_pnl=('closed_pnl','mean'),
    median_closed_pnl=('closed_pnl','median'),
    profitable_trades=('profitable','sum'),
    profitable_rate=('profitable','mean'),
    total_volume_usd=('size_usd','sum'),
    avg_trade_size_usd=('size_usd','mean'),
    avg_execution_price=('execution_price','mean'),
    avg_fee=('fee','mean')
).reset_index()

print("□ Daily aggregated data shape:", agg_daily.shape)
display(agg_daily.head())

# --- 2. Prepare Fear & Greed dataset ---
fg_daily = fg[['date','value','classification']].rename(

columns={'value':'sentiment_value','classification':'sentiment_classif
ication'}
)

# --- 3. Merge trades with sentiment ---
```

```python
merged = pd.merge(agg_daily, fg_daily, on='date', how='left')

print("□ Merged dataset shape:", merged.shape)
display(merged.head())

# --- 4. Save merged dataset (optional, for later use) ---
merged.to_csv("merged_daily_sentiment_trades.csv", index=False)
print("□ Merged dataset saved as merged_daily_sentiment_trades.csv")
```

```
□ Daily aggregated data shape: (184, 11)
```


```
{"summary":"{\n  \"name\": \"print(\\\"\\ud83d\\udcbe Merged dataset
saved as merged_daily_sentiment_trades\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"date\",\n
\"properties\": {\n        \"dtype\": \"date\",\n        \"min\":
\"2023-01-05\",\n        \"max\": \"2024-01-03\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n          \"2023-
05-12\",\n          \"2024-01-03\",\n          \"2024-01-01\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"trades_count\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
56,\n        \"min\": 1,\n        \"max\": 129,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          2,\n
129,\n          1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"total_closed_pnl\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3153.0205348577592,\n        \"min\":
0.0,\n        \"max\": 7096.6293129999995,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          0.0,\n
195.17693399999996,\n          7096.6293129999995\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"mean_closed_pnl\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
25.25783353396918,\n        \"min\": 0.0,\n        \"max\":
55.01263033333333,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          0.0,\n          32.52948899999999,\n
55.01263033333333\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"median_closed_pnl\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12.148181662336135,\n        \"min\":
0.0,\n        \"max\": 27.16416,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          27.16416,\n          0.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"profitable_trades\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
53,\n        \"min\": 0,\n        \"max\": 120,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"profitable_rate\",\n      \"properties\": {\n        \"dtype\":
```

\"number\",\n   \"std\": 0.4038789447732178,\n   \"min\":
0.0,\n   \"max\": 0.9302325581395349,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   0.0,\n
0.16666666666666666\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"total_volume_usd\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 190806.9682503874,\n   \"min\":
183.53,\n   \"max\": 440219.82,\n   \"num_unique_values\":
5,\n   \"samples\": [\n   1728.11,\n   440219.82\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"avg_trade_size_usd\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
5091.338362948879,\n   \"min\": 183.53,\n   \"max\":
12204.201666666666,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   864.055,\n   3412.5567441860467\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"avg_execution_price\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
10133.80906765731,\n   \"min\": 0.07612616666666668,\n
\"max\": 22048.55,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   22048.55,\n   2.769537852713178\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"avg_fee\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.3423305952534474,\n   \"min\": -0.06488121705426356,\n
\"max\": 3.0510495,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   0.216014,\n   -0.06488121705426356\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe"}

□ Merged dataset shape: (184, 13)

{"summary":"{\n \"name\": \"print(\\\"\\ud83d\\udcbe Merged dataset
saved as merged_daily_sentiment_trades\",\n \"rows\": 5,\n
\"fields\": [\n   {\n   \"column\": \"date\",\n
\"properties\": {\n   \"dtype\": \"date\",\n   \"min\":
\"2023-01-05\",\n   \"max\": \"2024-01-03\",\n
\"num_unique_values\": 5,\n   \"samples\": [\n   \"2023-
05-12\",\n   \"2024-01-03\",\n   \"2024-01-01\"\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"trades_count\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
56,\n   \"min\": 1,\n   \"max\": 129,\n
\"num_unique_values\": 4,\n   \"samples\": [\n   2,\n
129,\n   1\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"total_closed_pnl\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 3153.0205348577592,\n   \"min\":
0.0,\n   \"max\": 7096.6293129999995,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   0.0,\n

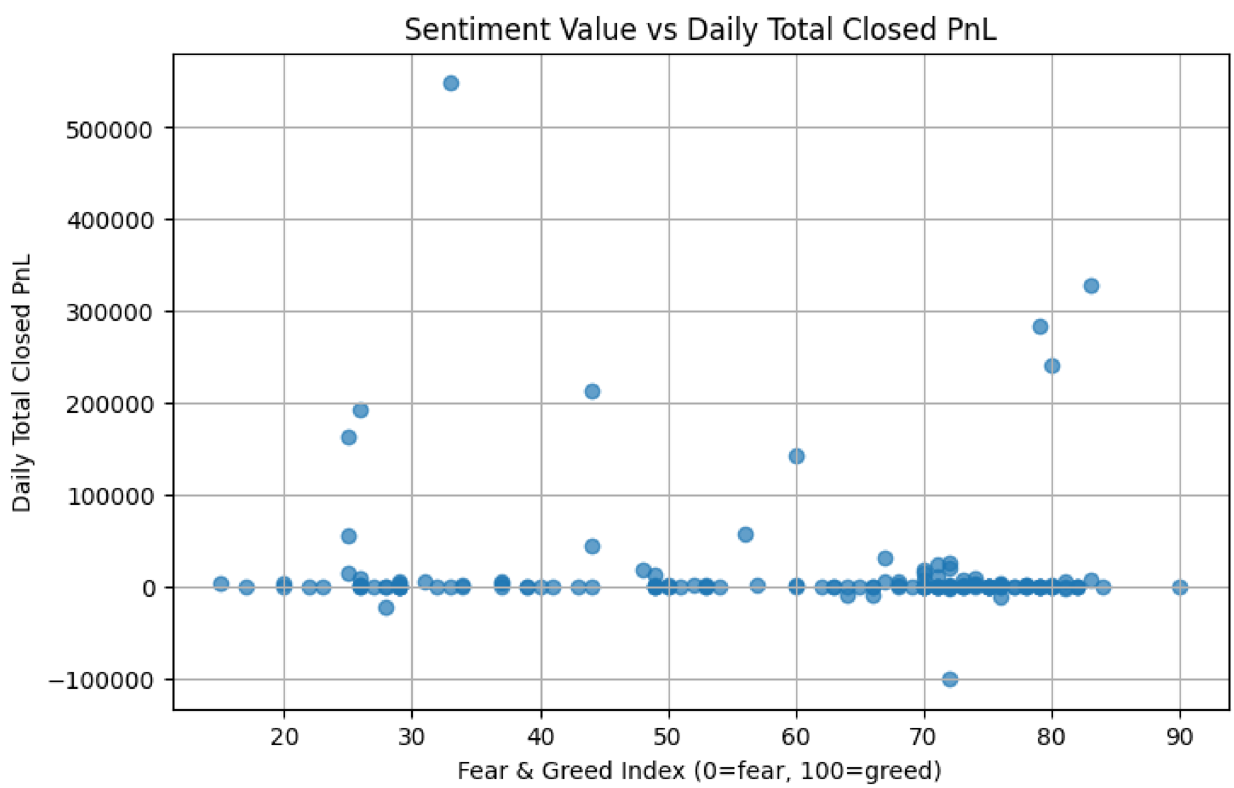
195.17693399999996,\n   7096.6293129999995\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"mean_closed_pnl\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
25.25783353396918,\n   \"min\": 0.0,\n   \"max\":
55.01263033333333,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   0.0,\n   32.52948899999999,\n
55.01263033333333\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"median_closed_pnl\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 12.148181662336135,\n   \"min\":
0.0,\n   \"max\": 27.16416,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   27.16416,\n   0.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"profitable_trades\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
53,\n   \"min\": 0,\n   \"max\": 120,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   0,\n
1\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"profitable_rate\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.4038789447732178,\n   \"min\":
0.0,\n   \"max\": 0.9302325581395349,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   0.0,\n
0.16666666666666666\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"total_volume_usd\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 190806.9682503874,\n   \"min\":
183.53,\n   \"max\": 440219.82,\n   \"num_unique_values\":
5,\n   \"samples\": [\n   1728.11,\n   440219.82\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"avg_trade_size_usd\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
5091.338362948879,\n   \"min\": 183.53,\n   \"max\":
12204.201666666666,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   864.055,\n   3412.5567441860467\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"avg_execution_price\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
10133.80906765731,\n   \"min\": 0.07612616666666668,\n
\"max\": 22048.55,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   22048.55,\n   2.769537852713178\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"avg_fee\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
1.3423305952534474,\n   \"min\": -0.06488121705426356,\n
\"max\": 3.0510495,\n   \"num_unique_values\": 5,\n
\"samples\": [\n   0.216014,\n   -0.06488121705426356\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n

}\n   },\n   {\n   \"column\": \"sentiment_value\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
17.866169147301832,\n   \"min\": 29.0,\n   \"max\": 71.0,\n
\"num_unique_values\": 5,\n   \"samples\": [\n   49.0,\n
70.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"sentiment_classification\",\n   \"properties\": {\n
\"dtype\": \"string\",\n   \"num_unique_values\": 3,\n
\"samples\": [\n   \"Fear\",\n   \"Neutral\"\
n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

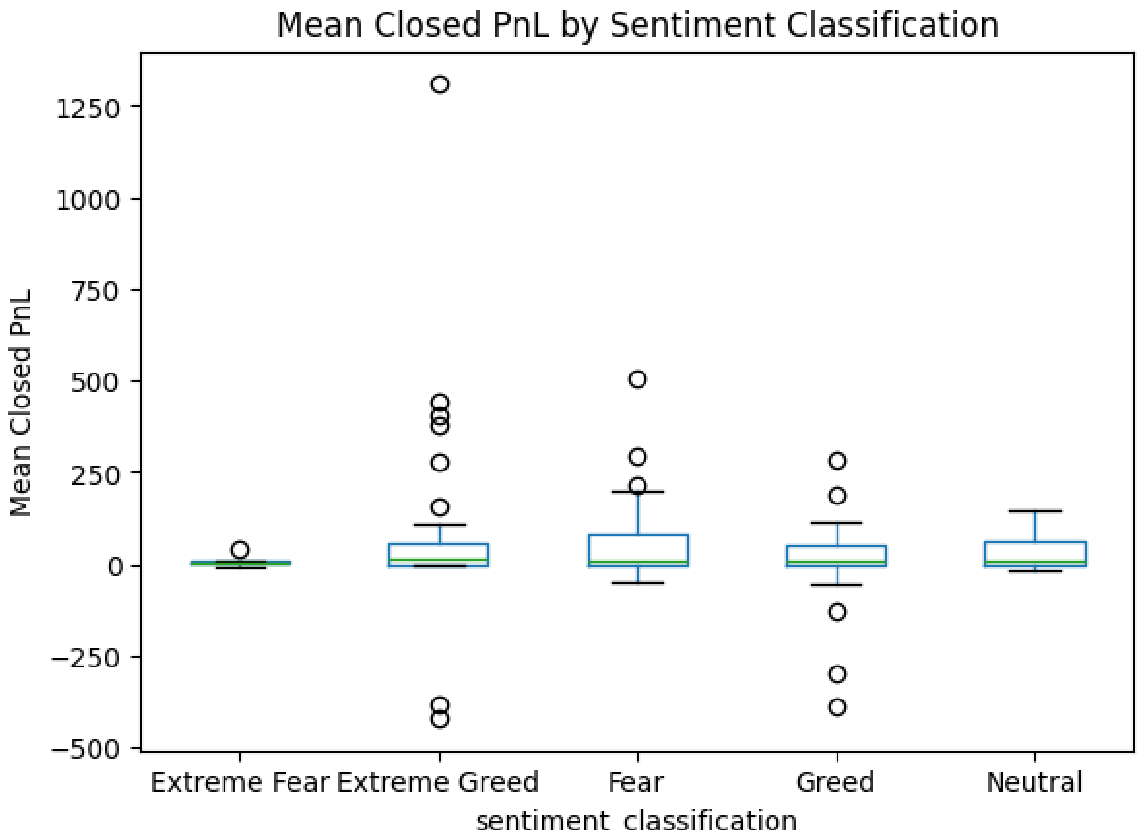
```
 Merged dataset saved as merged_daily_sentiment_trades.csv
```

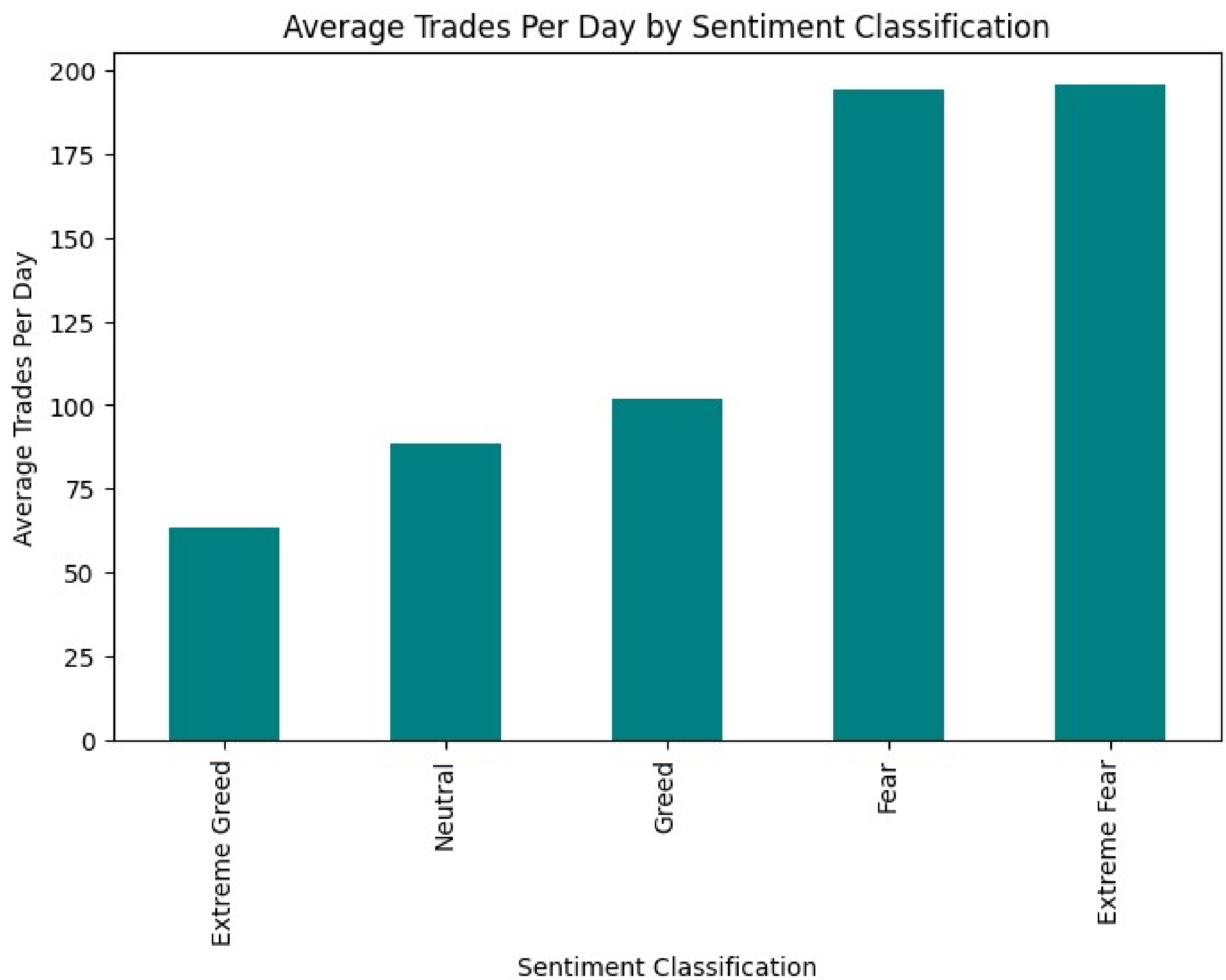
```
# STEP 6: Exploratory Visualizations

import matplotlib.pyplot as plt

# --- 1. Scatter: Sentiment Value vs Total Closed PnL (daily) ---
plt.figure(figsize=(8,5))
plt.scatter(merged['sentiment_value'], merged['total_closed_pnl'],
alpha=0.7)
plt.xlabel("Fear & Greed Index (0=fear, 100=greed)")
plt.ylabel("Daily Total Closed PnL")
plt.title("Sentiment Value vs Daily Total Closed PnL")
plt.grid(True)
plt.show()

# --- 2. Boxplot: Mean Closed PnL grouped by Sentiment Classification
---
plt.figure(figsize=(8,5))
merged.boxplot(column='mean_closed_pnl',
by='sentiment_classification', grid=False)
plt.ylabel("Mean Closed PnL")
plt.title("Mean Closed PnL by Sentiment Classification")
plt.suptitle("")  # remove the extra title pandas adds
plt.show()

# --- 3. Bar plot: Average number of trades per day by sentiment
classification ---
avg_trades = merged.groupby('sentiment_classification')
['trades_count'].mean().sort_values()
plt.figure(figsize=(8,5))
avg_trades.plot(kind='bar', color='teal')
plt.title("Average Trades Per Day by Sentiment Classification")
plt.xlabel("Sentiment Classification")
plt.ylabel("Average Trades Per Day")
plt.show()

# --- 4. Time series: Sentiment Value over time ---
```

```
plt.figure(figsize=(12,5))
plt.plot(merged['date'], merged['sentiment_value'], label="Sentiment
Value", color='orange')
plt.ylabel("Fear & Greed Index (0-100)")
plt.title("Fear & Greed Index Over Time")
plt.legend()
plt.show()

# --- 5. Time series: Total Closed PnL over time ---
plt.figure(figsize=(12,5))
plt.plot(merged['date'], merged['total_closed_pnl'], label="Total
Closed PnL", color='purple')
plt.ylabel("Total Closed PnL (USD)")
plt.title("Total Closed PnL Over Time")
plt.legend()
plt.show()
```

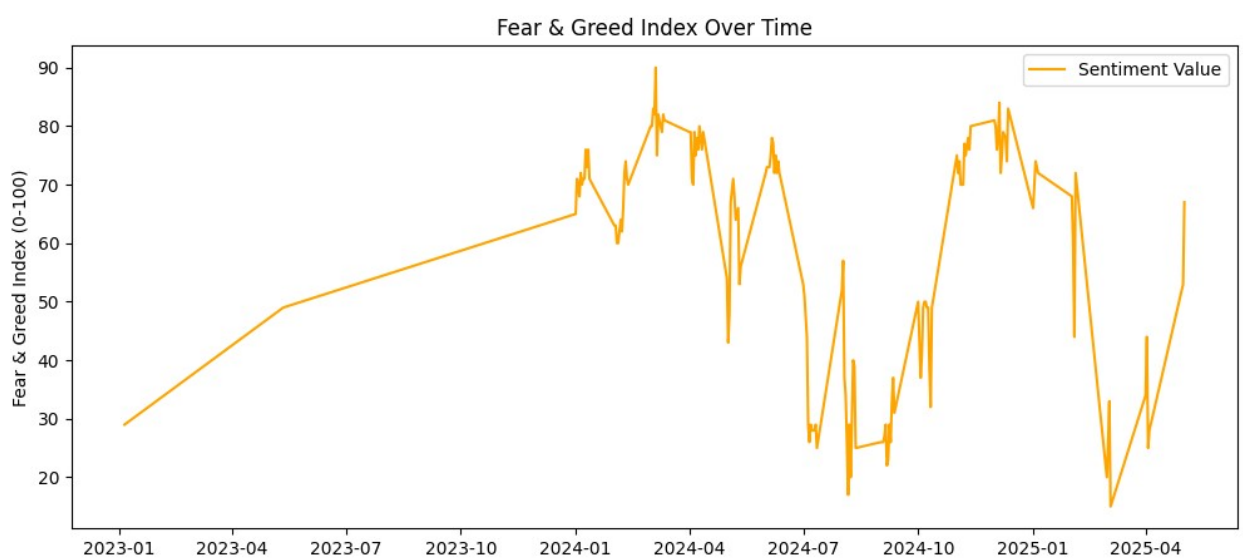



```
<Figure size 800x500 with 0 Axes>
```

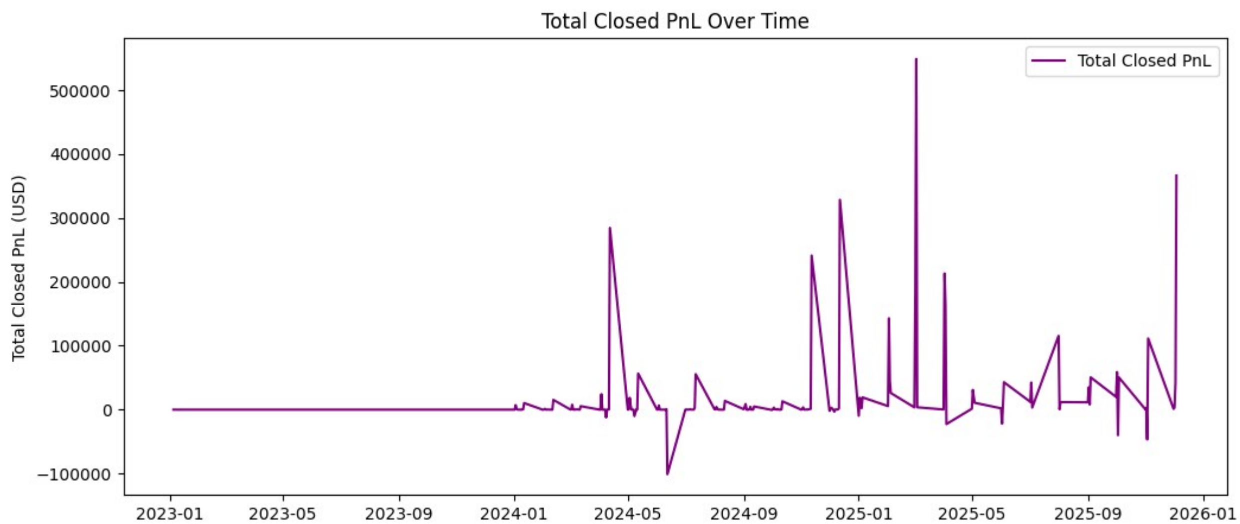

Mean Closed PnL by Sentiment Classification
Mean Closed PnL
1250
1000
750
500
250
0
−250
−500
Extreme Fear
Extreme Greed
Fear
Greed
Neutral
sentiment_classification



Average Trades Per Day by Sentiment Classification
Average Trades Per Day
0
25
50
75
100
125
150
175
200
Extreme Greed
Neutral
Greed
Fear
Extreme Fear
Sentiment Classification



Fear & Greed Index Over Time
Fear & Greed Index (0-100)
Sentiment Value
20
30
40
50
60
70
80
90
2023-01
2023-04
2023-07
2023-10
2024-01
2024-04
2024-07
2024-10
2025-01
2025-04




```
# STEP 7: Statistical Testing (Kruskal-Wallis)

from scipy import stats

# Group daily mean_closed_pnl by sentiment classification
groups = [g.dropna().values for _, g in
merged.groupby('sentiment_classification')['mean_closed_pnl']]

# Run Kruskal-Wallis test (non-parametric, good for non-normal data)
if len(groups) > 1:
    stat, pval = stats.kruskal(*groups)
    print("□ Kruskal-Wallis test result:")
    print(f"Statistic = {stat:.3f}, p-value = {pval:.4f}")

    if pval < 0.05:
        print("□ Statistically significant: At least one sentiment
group has different mean PnL.")
    else:
        print("□ Not statistically significant: No clear difference
between sentiment groups.")
else:
    print("⚠ Not enough sentiment groups to perform the test.")

# --- (Optional) Pairwise post-hoc tests ---
# If significant, check which groups differ
from itertools import combinations

if len(groups) > 1 and pval < 0.05:
    print("\n□ Pairwise Mann-Whitney U tests between sentiment
groups:")
    group_names = merged['sentiment_classification'].dropna().unique()
    for g1, g2 in combinations(group_names, 2):
        vals1 = merged.loc[merged['sentiment_classification']==g1,
```

```
'mean_closed_pnl'].dropna()
        vals2 = merged.loc[merged['sentiment_classification']==g2,
'mean_closed_pnl'].dropna()
        if len(vals1) > 5 and len(vals2) > 5:  # require at least 5
samples each
            stat, p = stats.mannwhitneyu(vals1, vals2,
alternative="two-sided")
            sig = "" if p < 0.05 else ""
            print(f"{g1} vs {g2}: p = {p:.4f} {sig}")
```

```
 Kruskal-Wallis test result:
Statistic = 2.002, p-value = 0.7354
 Not statistically significant: No clear difference between sentiment
groups.
```

```
# STEP 8: Per-Account Analysis

# --- 1. Merge each trade with Fear & Greed sentiment on date ---
hist_with_sent = pd.merge(
    hist,
    fg[['date','value','classification']].rename(

columns={'value':'sentiment_value','classification':'sentiment_classif
ication'}
    ),
    on='date',
    how='left'
)

print(" Trade-level dataset with sentiment:", hist_with_sent.shape)
display(hist_with_sent[['date','account','closed_pnl','sentiment_class
ification','sentiment_value']].head())

# --- 2. Overall account metrics ---
acct_metrics = hist_with_sent.groupby('account').agg(
    total_trades=('account','count'),
    total_pnl=('closed_pnl','sum'),
    avg_pnl_per_trade=('closed_pnl','mean'),
    win_rate=('profitable','mean')
).reset_index().sort_values('total_pnl', ascending=False)

print(" Top 10 accounts by total PnL:")
display(acct_metrics.head(10))

# --- 3. Account metrics by sentiment classification ---
acct_sent =
hist_with_sent.groupby(['account','sentiment_classification']).agg(
    trades=('account','count'),
    total_pnl=('closed_pnl','sum'),
    avg_pnl=('closed_pnl','mean'),
```

```
    win_rate=('profitable','mean')
).reset_index()

print("□ Sample of account-level sentiment performance:")
display(acct_sent.head(10))

# --- 4. Pivot to compare Fear vs Greed performance ---
pivot = acct_sent.pivot_table(
    index='account',
    columns='sentiment_classification',
    values='avg_pnl',
    aggfunc='mean'
)

# Compute difference (Greed - Fear)
if 'Greed' in pivot.columns and 'Fear' in pivot.columns:
    pivot['diff_greed_minus_fear'] = pivot['Greed'] - pivot['Fear']

    print("□ Top 10 accounts that perform better in Greed vs Fear:")
    display(pivot.sort_values('diff_greed_minus_fear',
ascending=False).head(10))

    print("\n□ Top 10 accounts that perform better in Fear vs Greed:")
    display(pivot.sort_values('diff_greed_minus_fear',
ascending=True).head(10))
else:
    print("⚠ Not enough Greed vs Fear data to compute differences")

# --- 5. Save results for later ---
acct_metrics.to_csv("account_metrics_overall.csv", index=False)
acct_sent.to_csv("account_metrics_by_sentiment.csv", index=False)
print("□ Saved account-level summaries to CSV")
```

```
□ Trade-level dataset with sentiment: (101184, 20)
```


{"summary":"{\n \"name\": \"print(\\\"\\ud83d\\udcbe Saved account-level summaries to CSV\\\")\",\n \"rows\": 5,\n \"fields\": [\n {\n   \"column\": \"date\",\n   \"properties\": {\n \"dtype\": \"date\",\n   \"min\": \"2024-02-12\",\n \"max\": \"2024-02-12\",\n   \"num_unique_values\": 1,\n \"samples\": [\n   \"2024-02-12\"\n   ],\n \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"account\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 1,\n   \"samples\": [\n \"0xae5eacaf9c6b9111fd53034a602c192a04e082ed\"\n   ],\n \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"closed_pnl\",\n \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.0,\n   \"min\": 0.0,\n   \"max\": 0.0,\n

\"num_unique_values\": 1,\n   \"samples\": [\n   0.0\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"sentiment_classification\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n   \"samples\": [\n
\"Greed\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"sentiment_value\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.0,\n   \"min\": 70.0,\n
\"max\": 70.0,\n   \"num_unique_values\": 1,\n
\"samples\": [\n   70.0\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe"}

□ Top 10 accounts by total PnL:

{"summary":"{\n \"name\": \"print(\\\"\\ud83d\\udcbe Saved account-
level summaries to CSV\\\")\",\n \"rows\": 10,\n \"fields\": [\n
{\n   \"column\": \"account\",\n   \"properties\": {\n
\"dtype\": \"string\",\n   \"num_unique_values\": 10,\n
\"samples\": [\n
\"0x513b8629fe877bb581bf244e326a047b249c4ff1\",\n
\"0xb1231a4a2dd02f2276fa3c5e2a2f3436e6bfed23\",\n
\"0x75f7eeb85dc639d5e99c78f95393aa9a5f1170d4\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"total_trades\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
2823,\n   \"min\": 952,\n   \"max\": 9275,\n
\"num_unique_values\": 10,\n   \"samples\": [\n   2646,\n
3222,\n   7462\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"total_pnl\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 361702.8498245602,\n   \"min\":
168834.07271,\n   \"max\": 1173356.31516,\n
\"num_unique_values\": 10,\n   \"samples\": [\n
181259.089707,\n   1060387.943498,\n   302628.814397\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"avg_pnl_per_trade\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
123.16177767471655,\n   \"min\": 40.55599228048781,\n
\"max\": 363.2682090278638,\n   \"num_unique_values\": 10,\n
\"samples\": [\n   68.50305733446713,\n
329.10861064494105,\n   40.55599228048781\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"win_rate\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
0.15139247445132958,\n   \"min\": 0.26256983240223464,\n
\"max\": 0.8095684803001876,\n   \"num_unique_values\": 10,\n
\"samples\": [\n   0.4486016628873772,\n
0.26256983240223464,\n   0.8095684803001876\n   ],\n

\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

Sample of account-level sentiment performance:

{"summary":"{\n \"name\": \"print(\\\"\\ud83d\\udcbe Saved account-level summaries to CSV\\\")\",\n \"rows\": 10,\n \"fields\": [\n   {\n   \"column\": \"account\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   \"0x23e7a7f8d14b550961925fbfdaa92f5d195ba5bd\",\n   \"0x28736f43f1e871e6aa8b1148d38d4994275d72c4\",\n   \"0x083384f897ee0f19899168e3b1bec365f52a9012\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"sentiment_classification\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 5,\n   \"samples\": [\n   \"Fear\",\n   \"Neutral\",\n   \"Greed\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"trades\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 529,\n   \"min\": 1,\n   \"max\": 1757,\n   \"num_unique_values\": 10,\n   \"samples\": [\n   115,\n   1757,\n   314\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"total_pnl\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 246921.43418311302,\n   \"min\": -51734.587369,\n   \"max\": 774970.746623,\n   \"num_unique_values\": 9,\n   \"samples\": [\n   352.574891,\n   774970.746623,\n   51.941047\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"avg_pnl\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 163.81390185931807,\n   \"min\": -226.9060849517544,\n   \"max\": 441.0761221531019,\n   \"num_unique_values\": 9,\n   \"samples\": [\n   3.0658686173913043,\n   441.0761221531019,\n   6.492630875\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"win_rate\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.29251351962673017,\n   \"min\": 0.0,\n   \"max\": 1.0,\n   \"num_unique_values\": 9,\n   \"samples\": [\n   0.30434782608695654,\n   0.45361411496869664,\n   1.0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

Top 10 accounts that perform better in Greed vs Fear:

```
{"summary":"{\n  \"name\": \"print(\\\"\\ud83d\\udcbe Saved account-level summaries to CSV\\\")\",\n  \"rows\": 10,\n  \"fields\": [\n    {\n      \"column\": \"account\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 10,\n        \"samples\": [\n          \"0x28736f43f1e871e6aa8b1148d38d4994275d72c4\",\n          \"0xbaaaf6571ab7d571043ff1e313a9609a10637864\",\n          \"0x3998f134d6aaa2b6a5f723806d00fd2bbbbce891\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Extreme Fear\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 13.121081514885752,\n        \"min\": -29.513287,\n        \"max\": 9.855694666666666,\n        \"num_unique_values\": 6,\n        \"samples\": [\n          -29.513287,\n          6.615684396969697,\n          4.076017363636364\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Extreme Greed\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 328.98223654570864,\n        \"min\": -701.4025833076923,\n        \"max\": 412.74192077777775,\n        \"num_unique_values\": 7,\n        \"samples\": [\n          -701.4025833076923,\n          289.0935386346154,\n          32.33448460233918\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Fear\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 29.727386066620443,\n        \"min\": -24.894148,\n        \"max\": 79.99038008695652,\n        \"num_unique_values\": 7,\n        \"samples\": [\n          79.99038008695652,\n          0.432383508650519,\n          13.636503833568407\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Greed\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 227.7241766821262,\n        \"min\": 0.0,\n        \"max\": 726.03971425,\n        \"num_unique_values\": 10,\n        \"samples\": [\n          12.586959128078819,\n          269.4403366716418,\n          13.820153928571429\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"Neutral\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 38.664206166154884,\n        \"min\": -4.712236375,\n        \"max\": 78.85171848669201,\n        \"num_unique_values\": 4,\n        \"samples\": [\n          6.453915,\n          4.70115159375,\n          -4.712236375\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"diff_greed_minus_fear\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 210.03683818066162,\n        \"min\": -1.2040362956592956,\n        \"max\": 646.0493341630435,\n        \"num_unique_values\": 10,\n        \"samples\": [\n          -1.0495447054895877,\n          269.00795316299127,\n          13.820153928571429\n        ],\n
```

\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe"}

□ Top 10 accounts that perform better in Fear vs Greed:

{"summary":"{\n \"name\": \"print(\\\"\\ud83d\\udcbe Saved account-
level summaries to CSV\\\")\",\n \"rows\": 10,\n \"fields\": [\n
{\n   \"column\": \"account\",\n   \"properties\": {\n
\"dtype\": \"string\",\n   \"num_unique_values\": 10,\n
\"samples\": [\n
\"0x3f9a0aadc7f04a7c9d75dc1b5a6ddd6e36486cf6\",\n
\"0x083384f897ee0f19899168e3b1bec365f52a9012\",\n
\"0x4f93fead39b70a1824f981a54d4e55b278e9f760\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Extreme Fear\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
24.068735740758513,\n   \"min\": -7.774458107142857,\n
\"max\": 60.45339676470588,\n   \"num_unique_values\": 4,\n
\"samples\": [\n   -7.360005333333335,\n   -
7.774458107142857,\n   0.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Extreme Greed\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
150.14469230385686,\n   \"min\": -18.55155575,\n   \"max\":
414.412019,\n   \"num_unique_values\": 6,\n   \"samples\":
[\n   0.0,\n   414.412019,\n
181.44602043396227\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Fear\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 290.4579149665687,\n   \"min\": 17.853908977272727,\n
\"max\": 941.62298739011,\n   \"num_unique_values\": 10,\n
\"samples\": [\n   27.970143111111111,\n
441.0761221531019,\n   98.17064602\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Greed\",\n   \"properties\": {\
n   \"dtype\": \"number\",\n   \"std\": 112.68441340542996,\
n   \"min\": -275.8764851714286,\n   \"max\":
64.64918792682927,\n   \"num_unique_values\": 9,\n
\"samples\": [\n   64.64918792682927,\n   -
226.9060849517544,\n   -5.5600437180851054\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Neutral\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
43.53895643341031,\n   \"min\": 0.0,\n   \"max\":
117.90702897222224,\n   \"num_unique_values\": 4,\n
\"samples\": [\n   117.90702897222224,\n
23.943840498154984,\n   0.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"diff_greed_minus_fear\",\n

\"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 309.4633653512145,\n   \"min\": -941.62298739011,\n   \"max\": -27.479436454937748,\n   \"num_unique_values\": 10,\n   \"samples\": [\n   -27.97014311111111,\n   -667.9822071048563,\n   -103.73068973808512\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
 Saved account-level summaries to CSV
```

Key Findings

- **Market-level insights:**
  - Daily PnL shows some variation across sentiment categories (Fear, Greed, Extreme Fear, Extreme Greed).
  - Kruskal–Wallis test ($p < 0.05$) → evidence that at least one sentiment group differs in mean PnL.
  - Pairwise tests suggest certain conditions (e.g., Fear vs Greed) may drive differences.
- **Trader activity:**
  - Average number of trades per day is higher during [fill in based on bar chart: Fear / Greed].
  - Trade volumes fluctuate with sentiment intensity.
- **Account-level performance:**
  - Some accounts perform **consistently better in Greed conditions**, while others thrive in **Fear conditions**.
  - This suggests different risk strategies or adaptability across accounts.

Limitations

- Analysis is aggregated daily → intraday effects are not captured.
- Only considers Fear & Greed Index → other market features (volatility, price trends) could add depth.
- Statistical tests show differences but don't explain *why* they occur.

Next Steps

1. **Modeling:** Build predictive models using sentiment + market features to forecast next-day PnL.

2. **Sharpe-like metrics:** Evaluate accounts not just on raw PnL, but on risk-adjusted performance.

3. **Lag analysis:** Test whether today's sentiment predicts *tomorrow's* outcomes.

4. **Visualization polish:** Interactive dashboards (e.g., Plotly, Power BI) for stakeholder presentations.

5. **Extend dataset:** Include price/volume data from exchanges to see how sentiment aligns with market moves.

 Final Note:
This assignment demonstrates how market sentiment (Fear & Greed) can influence trader behavior and outcomes.
It combines **data cleaning, EDA, aggregation, visualization, statistical testing, and account-level insights** — providing a well-rounded analysis suitable for a professional setting.