```
from google.colab import files
files.upload()
```

```
import os
os.listdir()
```

```
['.config',
 'fear_greed_index (1).csv',
 'historical_data (1).csv',
 'fear_greed_index.csv',
 'historical_data.csv',
 'sample_data']
```

load the data

```
import pandas as pd

trade_df = pd.read_csv('historical_data.csv')
sentiment_df = pd.read_csv('fear_greed_index.csv')

print(trade_df.head())

print(sentiment_df.head())
```

```
                                    Account  Coin  Execution Price  \
0  0xae5eacaf9c6b9111fd53034a602c192a04e082ed  @107           7.9769
1  0xae5eacaf9c6b9111fd53034a602c192a04e082ed  @107           7.9800
2  0xae5eacaf9c6b9111fd53034a602c192a04e082ed  @107           7.9855
3  0xae5eacaf9c6b9111fd53034a602c192a04e082ed  @107           7.9874
4  0xae5eacaf9c6b9111fd53034a602c192a04e082ed  @107           7.9894

   Size Tokens  Size USD Side     Timestamp IST  Start Position Direction  \
0       986.87   7872.16  BUY  02-12-2024 22:50        0.000000       Buy
1        16.00    127.68  BUY  02-12-2024 22:50      986.524596       Buy
2       144.09   1150.63  BUY  02-12-2024 22:50     1002.518996       Buy
3       142.98   1142.04  BUY  02-12-2024 22:50     1146.558564       Buy
4         8.73     69.75  BUY  02-12-2024 22:50     1289.488521       Buy

   Closed PnL                                  Transaction Hash     Order ID  \
0         0.0  0xec09451986a1874e3a980418412fcd0201f500c95bac...  52017706630
1         0.0  0xec09451986a1874e3a980418412fcd0201f500c95bac...  52017706630
2         0.0  0xec09451986a1874e3a980418412fcd0201f500c95bac...  52017706630
3         0.0  0xec09451986a1874e3a980418412fcd0201f500c95bac...  52017706630
4         0.0  0xec09451986a1874e3a980418412fcd0201f500c95bac...  52017706630

   Crossed       Fee      Trade ID     Timestamp
0     True  0.345404  8.950000e+14  1.730000e+12
1     True  0.005600  4.430000e+14  1.730000e+12
2     True  0.050431  6.600000e+14  1.730000e+12
3     True  0.050043  1.080000e+15  1.730000e+12
4     True  0.003055  1.050000e+15  1.730000e+12
    timestamp  value classification        date
0  1517463000     30           Fear  2018-02-01
1  1517549400     15   Extreme Fear  2018-02-02
2  1517635800     40           Fear  2018-02-03
3  1517722200     24   Extreme Fear  2018-02-04
4  1517808600     11   Extreme Fear  2018-02-05
```

checking data structure

```
# Trade
print("🔹 Trader Data Info:")
trade_df.info()
print("\n🔹 Null Values in Trade Data:")
print(trade_df.isnull().sum())

# Sentiment
print("\n🔹 Sentiment Data Info:")
sentiment_df.info()
print("\n🔹 Null Values in Sentiment Data:")
print(sentiment_df.isnull().sum())
```

```
 ---  ------            --------------   -----
 0   Account           211224 non-null  object
 1   Coin              211224 non-null  object
 2   Execution Price   211224 non-null  float64
 3   Size Tokens       211224 non-null  float64
 4   Size USD          211224 non-null  float64
 5   Side              211224 non-null  object
 6   Timestamp IST     211224 non-null  object
 7   Start Position    211224 non-null  float64
 8   Direction         211224 non-null  object
 9   Closed PnL        211224 non-null  float64
 10  Transaction Hash  211224 non-null  object
 11  Order ID          211224 non-null  int64
 12  Crossed           211224 non-null  bool
 13  Fee               211224 non-null  float64
 14  Trade ID          211224 non-null  float64
 15  Timestamp         211224 non-null  float64
dtypes: bool(1), float64(8), int64(1), object(6)
memory usage: 24.4+ MB

🔹 Null Values in Trade Data:
Account             0
Coin                0
Execution Price     0
Size Tokens         0
Size USD            0
Side                0
Timestamp IST       0
Start Position      0
Direction           0
Closed PnL          0
Transaction Hash    0
Order ID            0
Crossed             0
Fee                 0
Trade ID            0
Timestamp           0
dtype: int64

🔹 Sentiment Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2644 entries, 0 to 2643
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   timestamp       2644 non-null   int64
 1   value           2644 non-null   int64
 2   classification  2644 non-null   object
 3   date            2644 non-null   object
dtypes: int64(2), object(2)
memory usage: 82.8+ KB

🔹 Null Values in Sentiment Data:
timestamp         0
value             0
classification    0
date              0
dtype: int64
```

```
import pandas as pd

# 1. Converting 'Timestamp IST' in trade data to datetime
trade_df['Timestamp IST'] = pd.to_datetime(trade_df['Timestamp IST'], errors='coerce')
trade_df['date_only'] = trade_df['Timestamp IST'].dt.date

# 2. Convert 'date' in sentiment data to datetime and extract date
sentiment_df['date'] = pd.to_datetime(sentiment_df['date'], errors='coerce')
sentiment_df['date_only'] = sentiment_df['date'].dt.date

# 3. Merge the datasets on the date column
merged_df = pd.merge(trade_df, sentiment_df, on='date_only', how='inner')

# 4. Preview the merged data
display(merged_df.head())
```

	Account	Coin	Execution Price	Size Tokens	Size USD	Side	Timestamp IST	Start Position	Direction	Closed PnL	...	
0	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9769	986.87	7872.16	BUY	2024-02-12 22:50:00	0.000000	Buy	0.0	...	52(
1	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9800	16.00	127.68	BUY	2024-02-12 22:50:00	986.524596	Buy	0.0	...	52(
2	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9855	144.09	1150.63	BUY	2024-02-12 22:50:00	1002.518996	Buy	0.0	...	52(
3	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9874	142.98	1142.04	BUY	2024-02-12 22:50:00	1146.558564	Buy	0.0	...	52(
4	0xae5eacaf9c6b9111fd53034a602c192a04e082ed	@107	7.9894	8.73	69.75	BUY	2024-02-12 22:50:00	1289.488521	Buy	0.0	...	52(

5 rows × 21 columns

```
print("historical",trade_df.shape)
print("sentiment",sentiment_df.shape)
```

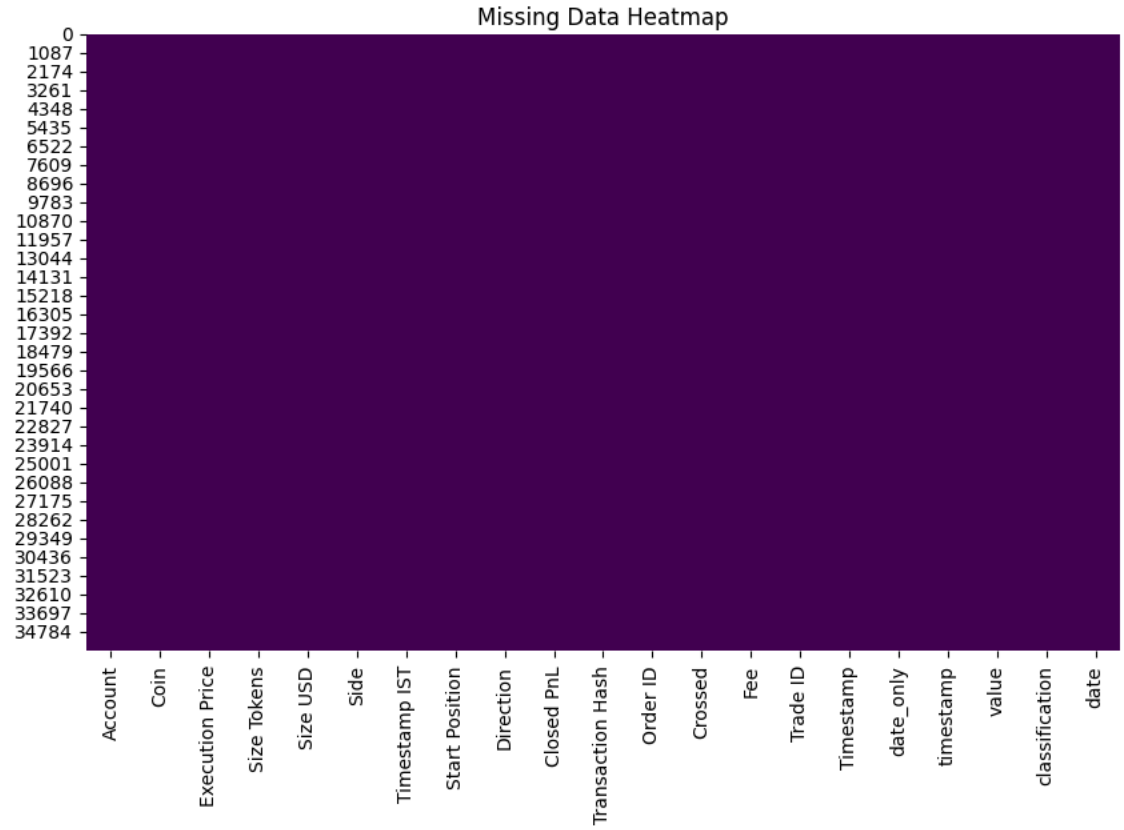
```
historical (211224, 17)
sentiment (2644, 5)
```

```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35864 entries, 0 to 35863
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Account           35864 non-null  object
 1   Coin              35864 non-null  object
 2   Execution Price   35864 non-null  float64
 3   Size Tokens       35864 non-null  float64
 4   Size USD          35864 non-null  float64
 5   Side              35864 non-null  object
 6   Timestamp IST     35864 non-null  datetime64[ns]
 7   Start Position    35864 non-null  float64
 8   Direction         35864 non-null  object
 9   Closed PnL        35864 non-null  float64
 10  Transaction Hash  35864 non-null  object
 11  Order ID          35864 non-null  int64
 12  Crossed           35864 non-null  bool
 13  Fee               35864 non-null  float64
 14  Trade ID          35864 non-null  float64
 15  Timestamp         35864 non-null  float64
 16  date_only         35864 non-null  object
 17  timestamp         35864 non-null  int64
 18  value             35864 non-null  int64
 19  classification    35864 non-null  object
 20  date              35864 non-null  datetime64[ns]
dtypes: bool(1), datetime64[ns](2), float64(8), int64(3), object(7)
memory usage: 5.5+ MB
```

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sns.heatmap(merged_df.isnull(), cbar=False, cmap='viridis')
plt.title("Missing Data Heatmap")
plt.show()
```




```
merged_df.describe()
```

	Execution Price	Size Tokens	Size USD	Timestamp IST	Start Position	Closed PnL	Order ID	Fee	Trade ID
count	35864.000000	3.586400e+04	3.586400e+04	35864	3.586400e+04	35864.000000	3.586400e+04	35864.000000	3.586400e+04
mean	7596.431745	1.004709e+04	4.920359e+03	2024-11-29 15:03:36.594914560	2.701678e+04	101.070948	5.844251e+10	1.028091	5.619946e+14
min	0.000005	5.630000e-06	0.000000e+00	2023-01-05 01:06:00	-1.050000e+07	-117990.104100	1.732711e+08	-1.175712	0.000000e+00
25%	3.245975	2.588700e+00	1.665300e+02	2024-09-12 12:47:00	-2.638983e+02	0.000000	5.240387e+10	0.012518	2.780000e+14
50%	16.390000	2.820000e+01	5.979050e+02	2025-01-05 07:06:00	7.010137e+01	0.000000	6.815193e+10	0.084384	5.620000e+14
75%	172.590000	1.990000e+02	2.220243e+03	2025-03-04 13:18:00	1.090053e+04	10.735428	7.641909e+10	0.398132	8.460000e+14
max	103265.000000	1.582244e+07	1.190250e+06	2025-05-02 23:59:00	3.050948e+07	71535.716740	9.014923e+10	212.298921	1.130000e+15
std	23547.203213	2.060973e+05	2.203314e+04	NaN	4.308246e+05	1364.610762	2.240666e+10	4.937482	3.262737e+14

```
merged_df['classification'].value_counts()
```

classification	count
Fear	13869
Greed	11292
Extreme Greed	5621
Neutral	2756
Extreme Fear	2326

dtype: int64

```
merged_df.groupby("classification")['Closed PnL'].mean().sort_values()
```

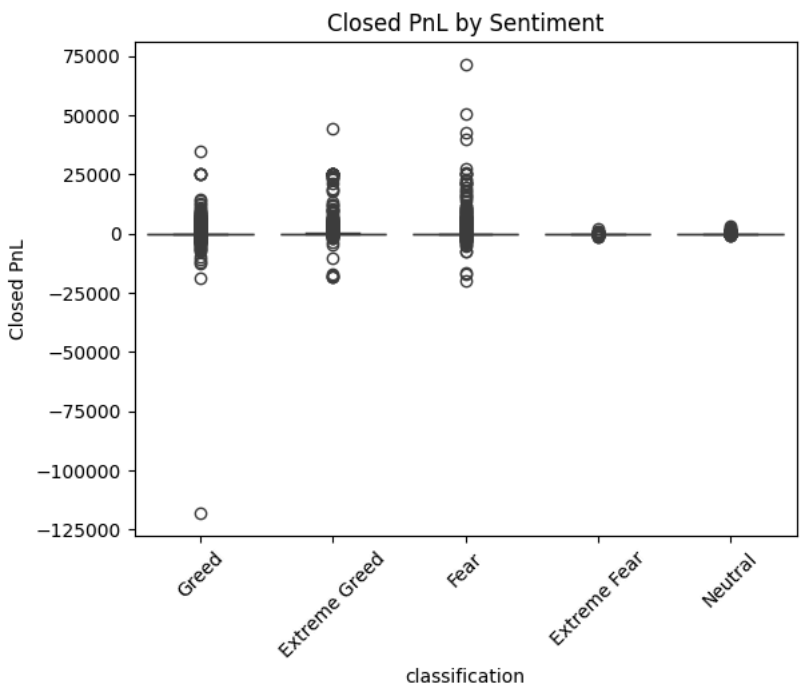
classification	Closed PnL
Extreme Fear	1.891632
Neutral	27.088803
Greed	53.988003
Fear	128.287950
Extreme Greed	205.816345

dtype: float64

```
merged_df['is_profitable'] = merged_df['Closed PnL'] > 0
win_rates = merged_df.groupby('classification')['is_profitable'].mean() * 100
print(win_rates)
```

```
classification
Extreme Fear     29.277730
Extreme Greed    55.328233
Fear             38.178672
Greed            43.570670
Neutral          49.492017
Name: is_profitable, dtype: float64
```

```
sns.boxplot(data=merged_df, x='classification', y='Closed PnL')
plt.title("Closed PnL by Sentiment")
plt.xticks(rotation=45)
plt.show()
```




```
merged_df['is_profitable'] = merged_df['Closed PnL'] > 0
```

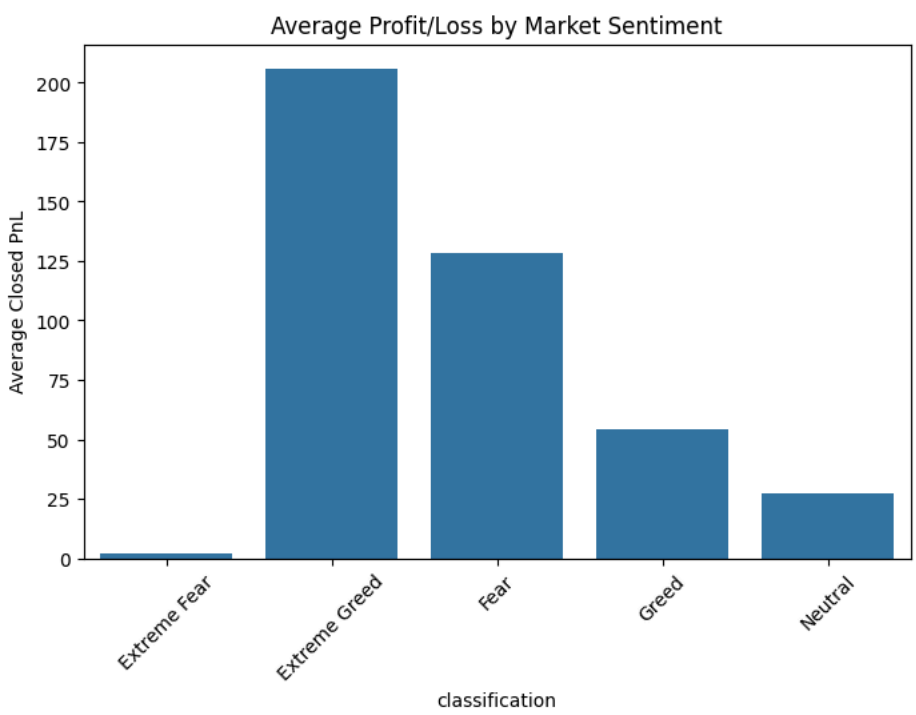
```
summary = merged_df.groupby('classification').agg(
    avg_profit = ('Closed PnL', 'mean'),
    median_profit = ('Closed PnL', 'median'),
    profit_std = ('Closed PnL', 'std'),
    min_profit = ('Closed PnL', 'min'),
    max_profit = ('Closed PnL', 'max'),
    avg_trade_size = ('Size USD', 'mean'),
    median_trade_size = ('Size USD', 'median'),
    win_rate = ('is_profitable', 'mean')
).reset_index()
```

```
summary['win_rate'] = summary['win_rate'] * 100
summary = summary.round(2)
print(summary)
```

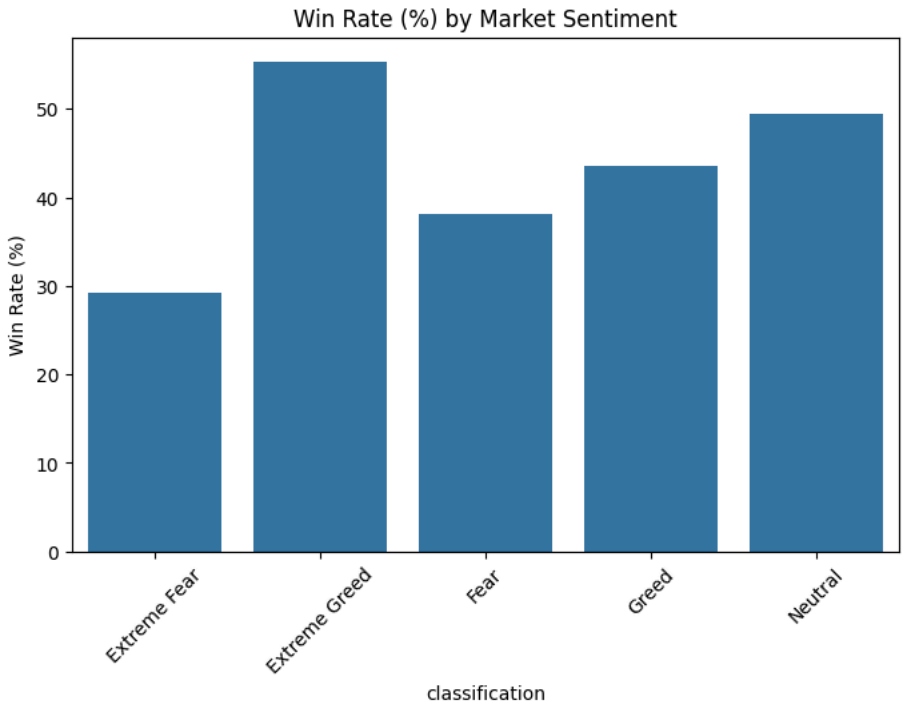
```
  classification  avg_profit  median_profit  profit_std  min_profit  \
0   Extreme Fear        1.89           0.00       76.73    -1430.89
1  Extreme Greed      205.82           0.96     1861.56   -18360.67
2           Fear      128.29           0.00     1342.35   -19841.24
3          Greed       53.99           0.00     1399.47  -117990.10
4        Neutral       27.09           0.00      142.95    -1032.98

   max_profit  avg_trade_size  median_trade_size  win_rate
0     2020.00         4118.76             599.12     29.28
1    44223.45         3242.09             365.00     55.33
2    71535.72         5744.78             703.88     38.18
3    34903.82         5051.88             675.08     43.57
4     2979.55         4332.20             411.81     49.49
```

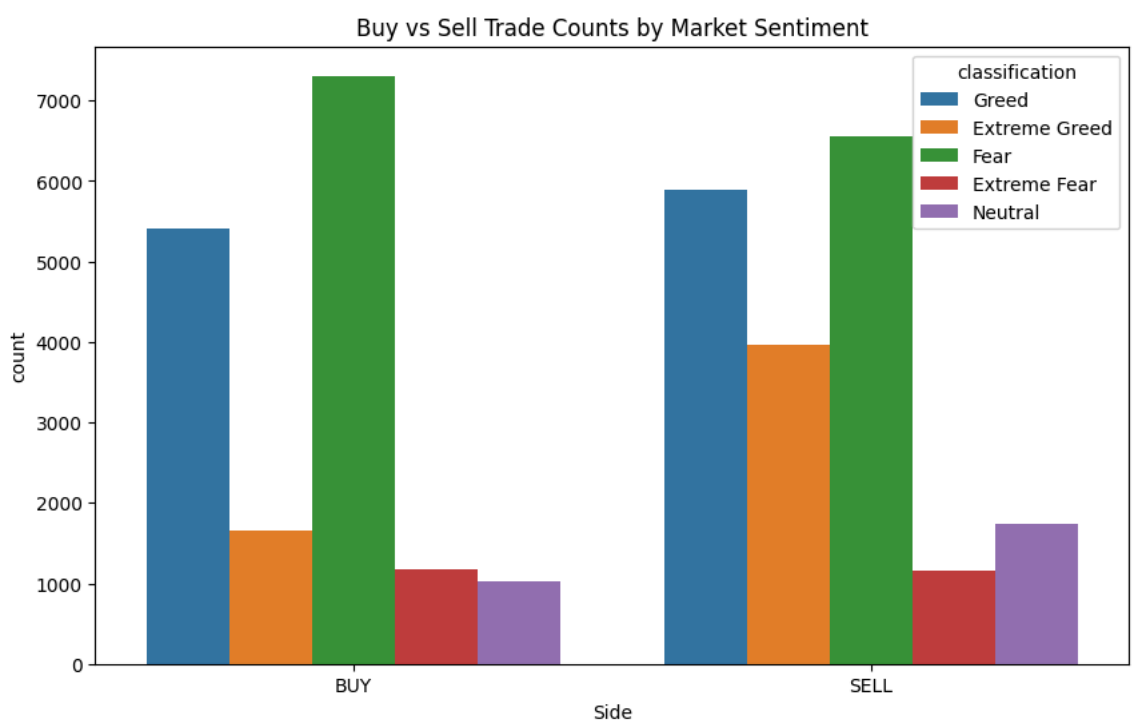
```
plt.figure(figsize=(8,5))
sns.barplot(data=summary, x='classification', y='avg_profit', order=summary['classification'])
plt.title('Average Profit/Loss by Market Sentiment')
plt.ylabel('Average Closed PnL')
plt.xticks(rotation=45)
plt.show()
```



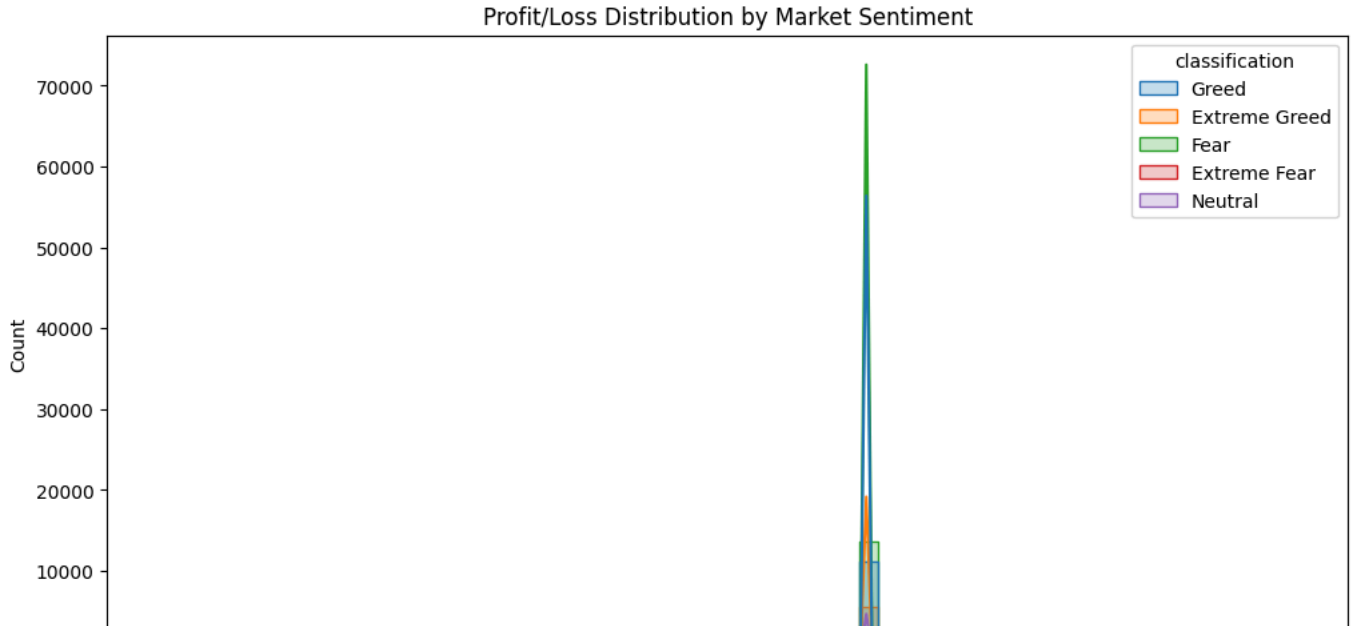

```
plt.figure(figsize=(8,5))
sns.barplot(data=summary, x='classification', y='win_rate', order=summary['classification'])
plt.title('Win Rate (%) by Market Sentiment')
plt.ylabel('Win Rate (%)')
plt.xticks(rotation=45)
plt.show()
```



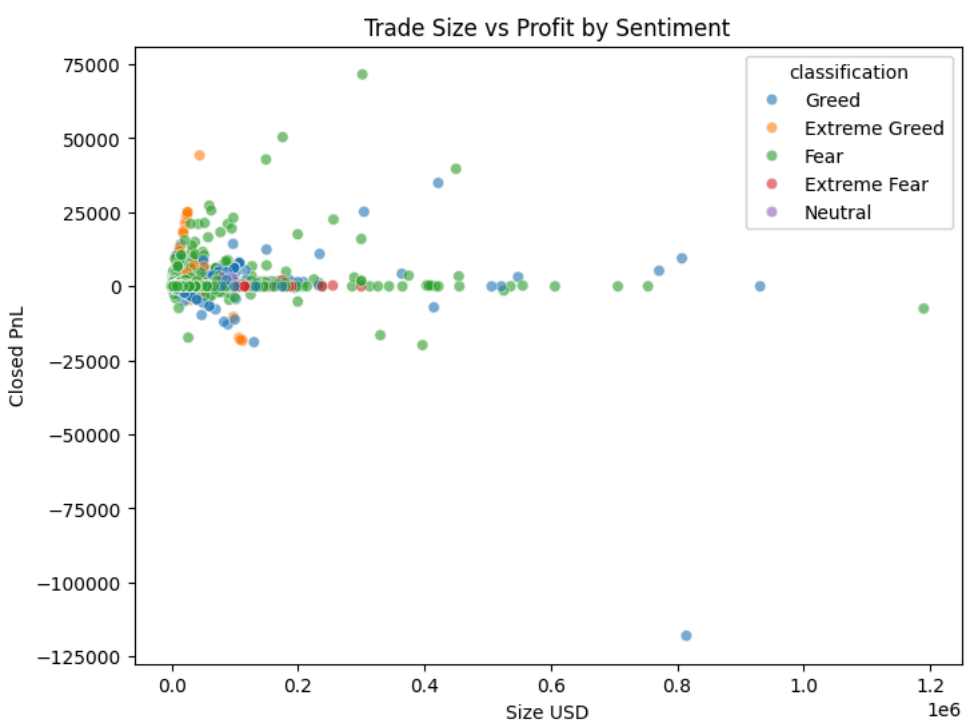

```
plt.figure(figsize=(10,6))
sns.countplot(data=merged_df, x='Side', hue='classification')
plt.title('Buy vs Sell Trade Counts by Market Sentiment')
plt.show()
```




```
plt.figure(figsize=(12,6))
sns.histplot(data=merged_df, x='Closed PnL', hue='classification', bins=60, kde=True, element='step')
plt.title('Profit/Loss Distribution by Market Sentiment')
plt.show()
```




```
plt.figure(figsize=(8,6))
sns.scatterplot(data=merged_df, x='Size USD', y='Closed PnL', hue='classification', alpha=0.6)
plt.title('Trade Size vs Profit by Sentiment')
plt.show()
```




```
trade_counts = merged_df['classification'].value_counts()
print("Number of trades per sentiment:")
print(trade_counts)
```

```
Number of trades per sentiment:
classification
Fear             13869
Greed            11292
Extreme Greed     5621
Neutral           2756
Extreme Fear      2326
Name: count, dtype: int64
```

```
print("Win rate (% profitable trades) by sentiment:")
```