Untitled4

July 4, 2025

[5]:

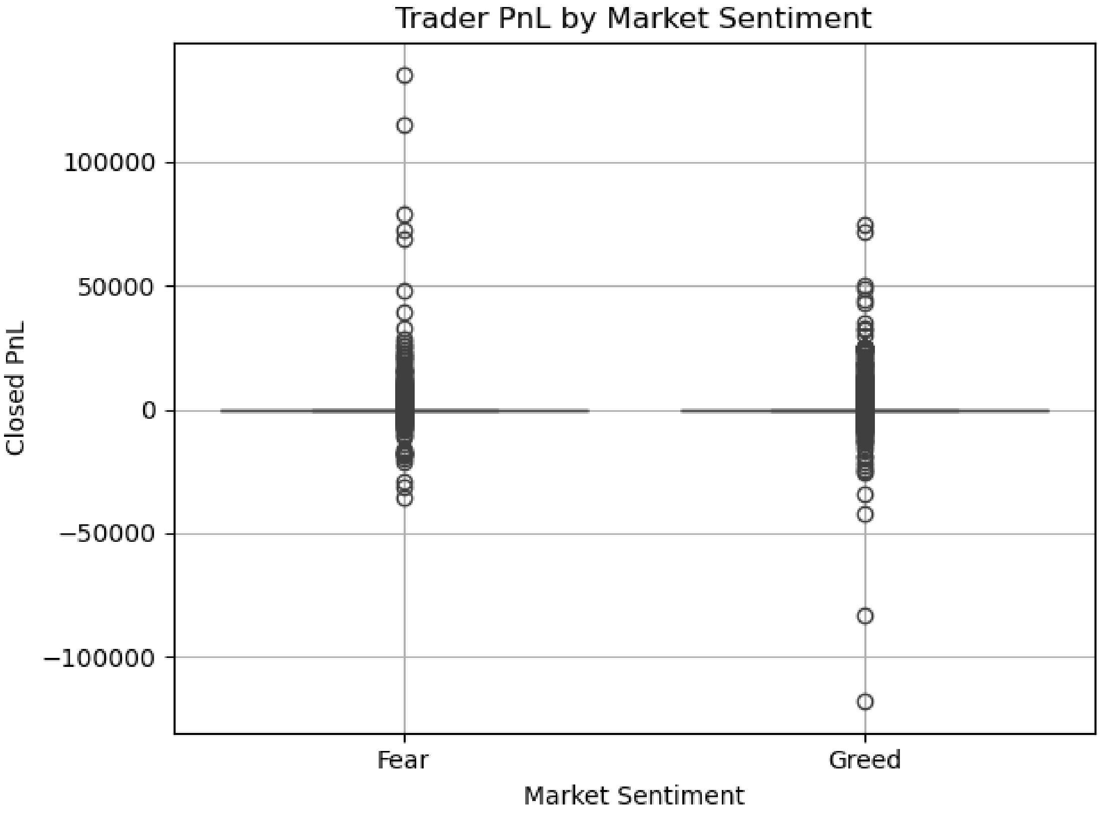
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# === Step 1: Load Datasets ===

# Load historical trader data
trader_df = pd.read_csv(r"C:\Users\HP\Desktop\CGS Assignment\historical_data.
 ↪csv")

# Load fear-greed index data
sentiment_df = pd.read_csv(r"C:\Users\HP\Desktop\CGS␣
 ↪Assignment\fear_greed_index.csv")
```

[4]:

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

# === Step 1: Load Data from your system ===
trader_df = pd.read_csv(r"C:\Users\HP\Desktop\CGS Assignment\historical_data.
 ↪csv")
sentiment_df = pd.read_csv(r"C:\Users\HP\Desktop\CGS␣
 ↪Assignment\fear_greed_index.csv")

# === Step 2: Clean and Prepare Trader Data ===
# Convert 'Timestamp IST' to datetime using correct format
trader_df['Timestamp IST'] = pd.to_datetime(trader_df['Timestamp IST'],␣
 ↪format="%d-%m-%Y %H:%M")
trader_df['Date'] = trader_df['Timestamp IST'].dt.normalize()  # drop time part

# === Step 3: Clean and Prepare Sentiment Data ===
sentiment_df['date'] = pd.to_datetime(sentiment_df['date'])
sentiment_df['Sentiment'] = sentiment_df['classification'].apply(lambda x:␣
 ↪'Fear' if 'Fear' in x else 'Greed')
sentiment_df['Sentiment'] = sentiment_df['Sentiment'].map({'Fear': 0, 'Greed':␣
 ↪1})
```

```python
# === Step 4: Merge Datasets ===
merged_df = pd.merge(trader_df, sentiment_df[['date', 'Sentiment']],␣
 ↪left_on='Date', right_on='date', how='inner')

# === Step 5: Add Win Column ===
merged_df['win'] = merged_df['Closed PnL'] > 0

# === Step 6: Summary Statistics ===
print("\n  Average Closed PnL by Sentiment:")
print(merged_df.groupby('Sentiment')['Closed PnL'].mean())

print("\n  Win Rate by Sentiment:")
print(pd.crosstab(merged_df['Sentiment'], merged_df['win'], normalize='index'))

# === Step 7: Visualization ===

# Boxplot of Closed PnL
sns.boxplot(data=merged_df, x='Sentiment', y='Closed PnL')
plt.xticks([0, 1], ['Fear', 'Greed'])
plt.title("Trader PnL by Market Sentiment")
plt.xlabel("Market Sentiment")
plt.ylabel("Closed PnL")
plt.grid(True)
plt.tight_layout()
plt.show()

# Optional: Leverage histogram if you add a 'leverage' column later
```

```
  Average Closed PnL by Sentiment:
Sentiment
0    49.212077
1    48.118246
Name: Closed PnL, dtype: float64

  Win Rate by Sentiment:
win           False     True
Sentiment
0          0.592129  0.407871
1          0.586556  0.413444
```



[6]:

```
merged_df.to_csv("merged_trader_sentiment.csv", index=False)
```

[9]:

```
from IPython.display import display, Markdown

display(Markdown("""
**On average, traders earned more during Fear days ( 49.21) than Greed days ( 48.
↪12).**

Win rates were higher on Greed days (41.34%) compared to Fear days (40.78%).

Closed PnL showed greater variability under Fear sentiment (standard deviation␣
↪ 990.88 vs  867.31 for Greed).

Visualizations reveal sentiment does impact performance, though the differences␣
↪are small - indicating subtle behavioral or strategic shifts in trading.
"""))
```

On average, traders earned more during Fear days (49.21) than Greed days (48.12).

Win rates were higher on Greed days (41.34%) compared to Fear days (40.78%).

Closed PnL showed greater variability under Fear sentiment (standard deviation 990.88 vs 867.31

for Greed).

Visualizations reveal sentiment does impact performance, though the differences are small — indicating subtle behavioral or strategic shifts in trading.

[ ]: