TRADERS@MIT

2024 Fall Competition

Case Packet

Email: traders@mit.edu
Website: traders.mit.edu
Instagram: @tradersatmit

November 9-10, 2024

Thank you to our sponsors!

CITADEL | CITADEL | Securities

Jane Street

SUSQUEHANNA DRW

jumptrading FIVE RINGS HRT hudson river trading

Midpoint Markets imc TRADING OLD MISSION

D E Shaw & Co Tower RESEARCH CAPITAL BRIDGEWATER TWO SIGMA

Contents

1 Introduction

Traders@MIT welcomes you to our 17th Annual Intercollegiate Trading Competition! We are excited to be holding the competition again this year and hope that it will provide a rich and challenging learning experience to competitors of all backgrounds. The competition will be held on Saturday, November 9 and Sunday, November 10 in the Multi-Purpose, Silverman Skyline, and Winter Garden Rooms at MIT Building E14.

This trading competition would not be possible without the generous support of our valued sponsors:

- **Diamond:** Citadel | Citadel Securities, Jane Street
- **Platinum:** DRW, Five Rings, Hudson River Trading, Jump Trading, Susquehanna
- **Gold:** Flow Traders, IMC Trading, Old Mission Capital
- **Silver:** Bridgewater Associates, The D. E. Shaw Group, Tower Research Capital, Two Sigma

Our competition consists of electronic trading. Teams will be ranked overall based on their combined rankings from each of the two cases. Between rounds, you will also have several opportunities to speak with our attending sponsors, as well as participate in events hosted by our diamond sponsors. The two cases featured are the **Five Rings Capital Market Making Case** and the **DRW Data Science Case**. Further details are provided in the following sections of the case packet.

The market-making case will focus on click trading and price discovery for the sum of rolls of weighted dice. Competitors will not only trade with each other and bots, but also impact the weighting of these dice. We recommend familiarizing yourself with market-making and strategy development by reading through the case description, as well as preparing any programs or spreadsheets that may help inform your trades throughout the case.

The data science case will involve quantitative modeling of a set of price signals. Relatively little advance preparation is expected, but we'd recommend familiarizing yourself with statistical analysis packages in Python and installing relevant packages on the laptop you'll be using in the competition—more information can be found in the case description.

2 Five Rings Market Making Case Brief

2.1 Overview

You will participate in a market-making simulation where your task is to manually click to trade several contracts. The game will run in multiple trading rounds and will require input from the user. During each trading round, you can submit *bid*[1] prices and/or *ask* prices, as well as a *size* on each side, for each available contract. Other teams will do the same. Your goal is to maximize your profits by trading against other competitors and bots that are active on the exchange.

At the end of each trading round, your positions will be *liquidated* according to a specified fair value and your *PnL* will be evaluated.

2.2 Details

2.2.1 Exchange

You will be click trading on our custom electronic exchange. At all times, you will have access to an anonymous order book, and be able to place orders at any price within a specified range. If any orders placed by teams or bots *cross*, trades will occur to uncross the market, with the trades happening at the price of the best available resting order. If two orders are placed at the same price, either by teams or by bots, the order that arrives at the exchange first will be given top priority, so that the first order will be completely filled before the second order begins trading.

The size of any individual order is limited to 10 shares. Other than that, there are no limits imposed by the exchange. You may acquire positions of any size on any of the contracts, including short positions, and may place as many orders as you wish.

2.2.2 Participants

Besides competing teams, there are two types of bots on each exchange.

[1]Some key terms are defined in a glossary at the end of this brief

1. One buyer bot and one seller bot that, over the course of a round, slowly buy or sell at the best available price. The total number of contracts each bot trades is independently and uniformly chosen in the range $[12000, 48000]$.
2. A passive market-maker bot who submits a wide market offering infinite size. The fair value of the contract you trade is guaranteed to be within the market-maker bot's market.

2.2.3 Scoring

Your score will be equal to your ranking for each **round**, with lower numbers indicating a better score. For example, first place would have score 1, second place score 2, etc. The **overall rankings** for this case will be determined by the geometric mean of the scores of each team on each of the 13 rounds.

2.3 Scenarios

In this game, you will be trading on the sum of a certain number of rolls of weighted six-sided dice (with face labels $1, 2, 3, 4, 5, 6$). The competitors will decide on the weights of each die. In every scenario, you will be trading up to three contracts: $SUMA$, $SUMB$, and $SPRD$.

Each round will involve (at most) two dice, which we denote by die A and die B. Let ΔA be the distribution of weights on die A, where ΔA_i is the weight on face i for each $i = 1, 2, 3, 4, 5, 6$. Similarly, let ΔB be the distribution of weights on die B.

To determine the weights of the dice for each round, you will be prompted to enter an integer between 1 and 6, inclusive. The weight assigned to each face value will be proportional to the number of participants selecting that value. For example, if die A is prompted with 50 teams, of which 5 teams select 1, 10 teams select 2, 5 teams select 3, 10 teams select 4, 8 teams select 5, and 12 teams select 6, we have

$$\Delta A_1 = \frac{1}{10},\ \Delta A_2 = \frac{1}{5},\ \Delta A_3 = \frac{1}{10},\ \Delta A_4 = \frac{1}{5},\ \Delta A_5 = \frac{4}{25},\ \Delta A_6 = \frac{6}{25}.$$

Depending on the round, you may be tasked with deciding the weights for only die A, only die B, or both. Furthermore, the timing of these decisions may vary, with selections occuring either before or during the round.

At the end of each round, your positions will be *liquidated* according to the final value for each contract.

2.3.1 Scenario 1: Introduction

In the first scenario, trading will focus exclusively on the $SUMA$ contract, which represents the sum of rolling a weighted die A 13 times. *There are 3 rounds in this scenario, each of which is identical and independent.*

Rounds 1 - 3: Each round will take 10 minutes. In the first 2 minutes, competitors will input an integer between 1 and 6, inclusive to determine ΔA. No trading will occur during this period. After 2 minutes, ΔA is fixed and 5 rolls of weighted die A are immediately revealed, allowing competitors to begin click trading. For the next 8 minutes, one new roll will be revealed at the start of each minute.

Note that your goal is to trade on the sum of rolling a weighted die A 13 times after the weights are decided by the competitors. Additionally, every round in this scenario is independent.

2.3.2 Scenario 2: Multiple Contracts

In the second scenario, you will trade on the following two contracts:

1. $SUMA$: Sum of weighted die A 13 times. [Same as Scenario 1 (2.3.1)]
2. $SUMB$: Sum of weighted die B 13 times.

There are 3 rounds in this scenario, each of which is identical and independent.

Rounds 4 - 6: In each round, $\Delta A = \Delta B$; that is, the weights of A and B are identical, but the rolls may not be. In the first 2 minutes of each 10-minute round, competitors will input one integer between 1 and 6, inclusive to determine ΔA and ΔB. Similar to Scenario 1 (2.3.1), for each round, $\Delta A = \Delta B$ is fixed. At the end of 2 minutes, 5 rolls of weighted die A and 5 rolls of weighted die B are immediately revealed, allowing competitors to begin click trading. For the next 8 minutes, one new roll will be revealed at the start of each minute.

2.3.3 Scenario 3: Spread

In the third scenario, you will still trade on the same contracts as Scenario 2 (2.3.2): $SUMA, SUMB$. However, $\Delta A \neq \Delta B$. However, you will additionally trade on a third contract:

1. $SPRD$: The difference between $SUMA$ and $SUMB$ ($SUMA - SUMB$)

Moreover, $\Delta A \neq \Delta B$ in this round.

There are 3 rounds in this scenario. Note that these rounds are not independent.

Round 7: For the first 2 minutes of the first 10-minute round, competitors will input two integers between 1 and 6, inclusive to determine ΔA and ΔB, respectively. Unlike Scenario 2 (2.3.2), it is highly likely that $\Delta A \neq \Delta B$. Afterwards, 5 rolls of weighted die A and 5 rolls of weighted die B are revealed, allowing competitors to begin click trading on $SUMA, SUMB$, and $SPRD$. For the next 8 minutes, one new roll of each die will be revealed each minute.

Round 8 - 9: Each round will take 10 minutes. For the first 2 minutes of each round, the two selected integers from the previous round remain, competitors may change up to one of the two values. Afterward, 5 rolls of weighted die A and 5 rolls of weighted die B are revealed, allowing competitors to begin click trading on $SUMA$, $SUMB$, and $SPRD$. For the next 8 minutes, one new roll will be revealed each minute.

2.3.4 Scenario 4: Chaos

In the fourth and final scenario, we will be trading on the same three contracts as Scenario 2 (2.3.2) and Scenario 3 (2.3.3): $SUMA$, $SUMB$, and $SPRD$. There are 4 rounds in this scenario. In this scenario, ΔA and ΔB are likely to change each minute.

Round 10: Similar to Round 7 in Scenario 3 (2.3.3), for the first 2 minutes of the first 10-minute round, competitors will input two integers between 1 and 6, inclusive to determine ΔA and ΔB, respectively. Afterward, 5 rolls of weighted die A and 5 rolls of weighted die B are revealed, allowing competitors to begin click trading on $SUMA$, $SUMB$, and $SPRD$. For the next 8 minutes, one new roll will be revealed each minute. However, before each reveal, competitors may change up to one of their two values. As a result, ΔA and ΔB will be different each minute.

Round 11 - 13: Each round will be 10 minutes. During the first 2 minutes of each round, no action will be required and no trading will occur. The numbers selected by competitors will stay fixed from the previous round. Afterward, 5 rolls of weighted die A and 5 rolls of weighted die B are revealed, allowing competitors to begin click trading on $SUMA$, $SUMB$, and $SPRD$. For the next 8 minutes, one new roll will be revealed each minute. Like Round 10, before each reveal, competitors may change up to one of their two values.

2.4 Case Timeline

Each round will consist of a 2-minute period for selecting new dice weights, during which no trading will occur, followed by 8 minutes of click trading (and selecting new weights in Scenario 4). Note that between each scenario, there will be a break at the host's discretion.

Let a be the selected value that affects the weighting of Die A, and b be the selected value that affects the weighting of Die B.

Round #	Scenario	Contracts	2-min User Inputs	8-min User Inputs
1	Introduction	$SUMA$	a	Trades
2	Introduction	$SUMA$	a	Trades
3	Introduction	$SUMA$	a	Trades
4	Multiple Contracts	$SUMA, SUMB$	$a (= b)$	Trades
5	Multiple Contracts	$SUMA, SUMB$	$a (= b)$	Trades
6	Multiple Contracts	$SUMA, SUMB$	$a (= b)$	Trades
7	Spread	All	$a \wedge b$	Trades
8	Spread	All	$a \oplus b$	Trades
9	Spread	All	$a \oplus b$	Trades
10	Chaos	All	$a \wedge b$	Trades, $a \oplus b$
11	Chaos	All	None	Trades, $a \oplus b$
12	Chaos	All	None	Trades, $a \oplus b$
13	Chaos	All	None	Trades, $a \oplus b$

2.5 Hints

For competitors new to market-making, think about the following questions to better understand how to start approaching this case.

- Given your private information (your chosen values a, b) and the publicly revealed die rolls, how will you estimate the true sum of all of the die rolls?
- Your goal is to maximize your PnL. Where is your PnL coming from in this case?
- Based on the first two considerations, how large do you want your orders to be, and at what price do you want to place them?

2.6 Glossary

Term	Definition
ask	an order placed to sell an asset
bid	an order placed to buy an asset
cross	referring to a buy and a sell order, when the buy order price is higher than a sell order price and a trade should occur
liquidate	to close out a position entirely so that at the end a market participant is holding only cash and no other assets
PnL	profit and loss, the overall value of your position evaluated against the fair value of the contract
size	the number of shares of an asset you would like to trade

3 DRW Data Science Case Brief

3.1 Overview

You are a member of the fictional trading arm of Traders@MIT, and you are asked to present evidence that some hand-crafted signals can be used to create a profitable trading strategy. To do this, you will construct a trading strategy in the form of a python function. You will be given a small amount of data to train your models, and you will be evaluated on a large hidden dataset. Your objective is to maximize the t-statistic[1] of your trading strategy.

Coming up with a successful strategy for this case might involve:

1. Performing statistical analysis to deduce the relationships between the signals and stock returns.
2. Determining the most profitable way to trade on that information.
3. Determining ways to trade in a way that reduces your variance.
4. Investigating the relationship between your trade sizing and t-statistic.

[1]: https://en.wikipedia.org/wiki/T-statistic. Note that maximizing this quantity involves increasing value and also reducing variance.

3.2 Case Information

3.2.1 Technical Details

You are given a training dataset ("train.csv") which is a DataFrame with 200 rows, each representing a different moment in time. Each row has some number of columns labeled "regressor{i}" or "target{i}" for some integer i. For each row, "regressor{i}" represents the value of the i'th signal for the given tick. "target{i}" represents the return of the i'th stock, meaning if your position in stock i is $\$X$ for the given tick, your PNL for that tick is $\$X \cdot$ target{i}. You can use this dataset to learn how to predict the stock movements given your feature set, and also estimate the performance of any potential trading strategies.

You will write a python function called **strategy** which takes as input only the regressors in a row of a DataFrame, and returns as output a list-like object with the same length as the number of stocks. The output represents the dollar positions you wish to have in each stock for the given tick, and they can be any real number. You may assume that you have no market impact, no trading costs, etc. Your submitted function will run on every row of the hidden DataFrame, and your score will be the t-statistic of your PNL over all the ticks. A starter notebook will be provided on the day of the competition with an exact copy of the code used to calculate your score and a template **strategy** function.

It is guaranteed that each tick is an independent generation of the same statistical process.

3.2.2 Round Details

There will be 2 rounds, each taking up one half of the allotted time. All the data and symbols will be available for each round. For each round, you will make a submission that contributes equally to your total score.

3.2.3 Technical Requirements

A fully-functional Google Colab link will be sent on the day of the competition. Note that a free-tier Google Colab session might time out during the competition. If you wish to use the notebook on your personal computer, you must minimally have **numpy**, **pandas**, and **sklearn** installed, though we also optionally recommend:

- matplotlib
- seaborn
- statsmodels
- xgboost

Your submission will be run on our AWS server using Python 3 and will be able to import **numpy** and **pandas**. You are welcome to use any package you like to model and explore the data, but your submission should only use **numpy** and **pandas**.

Attempts at abusing our system will result in immediate disqualification. This includes any attempts at reading or writing files.

Upon code submission, you will be provided feedback on your in-sample score, as well as any errors your code incurred in-sample. You should write your code to be robust to out-of-sample data.

3.2.4 Submission

Your python function will be submitted at a link that will be given on the day of the competition.