Game Theory
Programming Assignment – Cournot Adjustment and Multi-agent Learning

Cournout Adjustment Dynamics (Extra Credit):

Write a script that simulates Cournot adjustment dynamics for 2 player matrix games. The input/output syntax (if you choose to use MATLAB) should be of the form:

```
[A1,A2] = cournot(M1,M2,a1,a2,N)
```

where

- `M1` and `M2` are reward matrices for players 1 and 2, respectively.

  A 2×2 example is (your script should work for general matrix games):

	1	2
1	a, A	b, B
2	c, C	d, D

  $$\texttt{M1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \texttt{M2} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

- `a1` and `a2` are initial actions for players 1 and 2, respectively.
- `N` is the number of stages
- `A1` and `A2` are vectors with the history of actions for player 1 and 2, respectively.
- In the case of a non-unique best response, choose a best response at random.

1. Run your code on the four examples posted on Canvas. Include a screenshot of the results of your code being executed on each pair of matrices, and upload your code to Gradescope. Please do not print out your code and submit it with your homework. Note that if you choose to use a different programming language, you will need to figure out how to generate these arrays on your own (if you use Python, you may want to convert them to NumPy arrays using, for example, `scipy.io.loadmat`).
2. In any of the games, did the Cournot adjustment process converge to a Nash equilibrium? In any games, did it not converge at all? If so, why do you think it never converged?

Regret Matching:

Write a script that simulates regret matching for 2 player matrix games. The input/output syntax (if you choose to use MATLAB) should be of the form:

```
[A1,A2, R1, R2] = regretmatching(M1,M2,a1,a2,N)
```

where

- `M1` and `M2` are reward matrices for players 1 and 2, respectively.

  A 2×2 example is (your script should work for general matrix games):

	1	2
1	a, A	b, B
2	c, C	d, D

  $$\texttt{M1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \texttt{M2} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

- `a1` and `a2` are initial actions for players 1 and 2, respectively.
- `N` is the number of stages
- `A1` and `A2` are vectors of length N that encode the history of actions for player 1 and 2, respectively.
- `R1` and `R2` are vectors of length N that encode the max regret of player 1 and player 2 at each time instance.

1. Run your code on the zero-sum game example from lecture

	L	C	R
T	-1	0	1
M	1	-1	0
B	0	1	-1

   For several initial actions and with $N = 500$, plot the max regret of each player versus the iteration number. Verify that the max regret goes to 0.

2. Plot the distribution of the players' action profiles over time. Do you observe any patterns? What does the empirical frequency of play converge to as N increases, and how does this compare with Theorem 5.1 in Lecture 13?

Fictitious Play:

Write a script that simulates fictitious play for 2 player matrix games. The input/output syntax (if you choose to use MATLAB) should be of the form:

```
[A1,A2] = fictitiousplay(M1,M2,a1,a2,N)
```

where

- `M1` and `M2` are reward matrices for players 1 and 2, respectively.

  A 2×2 example is (your script should work for general matrix games):

	1	2
1	a, A	b, B
2	c, C	d, D

$$\texttt{M1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \texttt{M2} = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

- `a1` and `a2` are initial actions for players 1 and 2, respectively.
- `N` is the number of stages
- `A1` and `A2` are vectors of length N that encode the history of actions for player 1 and 2, respectively.

1. Run your code on the zero-sum game example from lecture

	L	C	R
T	-1	0	1
M	1	-1	0
B	0	1	-1

   For several initial actions and with $N = 500$, plot the running average utility for each player, i.e., $\bar{U}_1(t)$ versus t. Does this number converge? If so, how does this number compare to the value of the game?

2. Consider the Colonel Blotto game given in Question #3 on Lecture #9. Run your fictitious play algorithm on this example.

   (a) Do the average payoffs converge? If so, how do they relate to the value of the game?

   (b) Plot out the empirical frequency of play for players 1 and 2, i.e., $q_1(t)$ and $q_2(t)$. Do these empirical frequencies of play converge? If so, how do they relate to the security strategies of the players?

Write-up Template

Cournout Adjustment Dynamics (Extra Credit):

1. Run your code on the four examples posted on Canvas. Include a screenshot of the results of your code being executed on each pair of matrices.

2. In any of the games, did the Cournot adjustment process converge to a Nash equilibrium? In any of the games, did it not converge at all? If so, why do you think it never converged?

Regret Matching:

1. Run your code on the zero-sum game example from lecture

	L	C	R
T	-1	0	1
M	1	-1	0
B	0	1	-1

   For several initial actions and with $N = 500$, plot the max regret of each player versus the iteration number. Verify that the max regret goes to 0.

2. Plot the distribution of the players' action profiles over time. Do you observe any patterns? What does the empirical frequency of play converge to as N increases, and how does this compare with Theorem 5.1 in Lecture 13?

Fictitious Play:

1. Run your code on the zero-sum game example from lecture

	L	C	R
T	-1	0	1
M	1	-1	0
B	0	1	-1

For several initial actions and with $N = 500$, plot the running average utility for each player, i.e., $\bar{U}_1(t)$ versus t. Does this number converge? If so, how does this number compare to the value of the game?

2. Consider the Colonel Blotto game given in Question #3 on Lecture #9. Run your fictitious play algorithm on this example.

   (a) Do the average payoffs converge? If so, how do they relate to the value of the game?

(b) Plot out the empirical frequency of play for players 1 and 2, i.e., $q_1(t)$ and $q_2(t)$. Do these empirical frequencies of play converge? If so, how do they relate to the security strategies of the players?