



# Game Theory in Python

It's easier than you thought

Omer Nevo  
NeoWize

# The Prisoner's Dilemma

› Story time: Robbery

# The Prisoner's Dilemma

- › Story time: Robbery
- › Results

You \ Partner	Stay Silent	Confess
	Stay Silent	Confess
Stay Silent	2 years in prison	20 years in prison
Confess	1 year in prison	15 years in prison

# The Prisoner's Dilemma

- › Story time: Robbery
- › Results

You \ Partner	Stay Silent	Confess
	Stay Silent	Confess
Stay Silent	2 years in prison	20 years in prison
Confess	1 year in prison	15 years in prison

- › General Payoff Matrix

You \ Partner	Cooperate	Defect
	Cooperate	Defect
Cooperate	3 (Reward)	0 (Sucker)
Defect	5 (Temptation)	1 (Punishment)

# The Prisoner's Dilemma

- › Story time: Robbery
- › Results

You \ Partner	Stay Silent	Confess
	Stay Silent	Confess
Stay Silent	2 years in prison	20 years in prison
Confess	1 year in prison	15 years in prison

- › General Payoff Matrix
- › What would you do?

You \ Partner	Cooperate	Defect
	Cooperate	Defect
Cooperate	3 (Reward)	0 (Sucker)
Defect	5 (Temptation)	1 (Punishment)

## Why This Is Interesting

- › Prisoner's dilemma in real life
  - Arms races
  - Economics
  - Advertising & cigarettes

## Why This Is Interesting

- › Prisoner's dilemma in real life
  - Arms races
  - Economics
  - Advertising & cigarettes
- › Iterated prisoner's dilemma
  - Evolution & the three-spined stickleback

## Why This Is Interesting

- › Prisoner's dilemma in real life
  - Arms races
  - Economics
  - Advertising & cigarettes
- › Iterated prisoner's dilemma
  - Evolution & the three-spined stickleback
- › Questions: Which strategies work in different contexts?



# The Axelrod Package

```
import axelrod as axl
```

```
player1 = axl.Cooperator()
```

```
player2 = axl.Defector()
```

```
player3 = axl.TitForTat()
```

**Import and  
create players**

# The Axelrod Package

```
import axelrod as axl
```

```
player1 = axl.Cooperator(); player2 = axl.Defector(); player3 = axl.TitForTat()
```

```
match = axl.Match([player1,player2], 5)
```

```
match.play(), match.final_score()
```

```
((C, D), (C, D), (C, D), (C, D), (C, D)), (0, 25))
```

**Cooperator vs. Defector:  
Exploitation!**

# The Axelrod Package

```
import axelrod as axl
```

```
player1 = axl.Cooperator(); player2 = axl.Defector(); player3 = axl.TitForTat()
```

```
match = axl.Match([player1,player2], 5)
```

```
match.play(), match.final_score()
```

```
((C, D), (C, D), (C, D), (C, D), (C, D)), (0, 25))
```

```
match = axl.Match([player1,player3], 5)
```

```
match.play(), match.final_score()
```

```
((C, C), (C, C), (C, C), (C, C), (C, C)), (15, 15))
```

**Cooperator vs. TFT:  
Cooperation!**

# The Axelrod Package

```
import axelrod as axl
```

```
player1 = axl.Cooperator(); player2 = axl.Defector(); player3 = axl.TitForTat()
```

```
match = axl.Match([player1,player2], 5)
```

```
match.play(), match.final_score()
```

```
((C, D), (C, D), (C, D), (C, D), (C, D)), (0, 25))
```

```
match = axl.Match([player1,player3], 5)
```

```
match.play(), match.final_score()
```

```
((C, C), (C, C), (C, C), (C, C), (C, C)), (15, 15))
```

```
match = axl.Match([player2,player3], 5)
```

```
match.play(), match.final_score()
```

```
((D, C), (D, D), (D, D), (D, D), (D, D)), (9, 4))
```

**Defector vs. TFT:  
(Almost) Mutual defection!**

## More Than One Game

- › **Match:** Two players play PD multiple times
- › **Tournament:** Many players play, with each player playing against every other player
- › **Moran Process:** Playing multiple tournaments. At the end we duplicate players based on results and delete players randomly



## Storytime: Starting work at Random-Corp



## Storytime: Starting work at Random-Corp

- › A very strange company:
  - Monthly projects in pairs

## Storytime: Starting work at Random-Corp

- › A very strange company:
  - Monthly projects in pairs
  - Getting credit for your work



# Storytime: Starting work at Random-Corp

- › A very strange company:
  - Monthly projects in pairs
  - Getting credit for your work
  - Payoff Matrix (familiar?)

Partner You	C	D
C	3	0
D	5	1

# Storytime: Starting work at Random-Corp

› A very strange company:

- Monthly projects in pairs
- Getting credit for your work
- Payoff Matrix (familiar?)
- End of year reviews

Partner You	C	D
C	3	0
D	5	1

# Storytime: Starting work at Random-Corp

› A very strange company:

- Monthly projects in pairs
- Getting credit for your work
- Payoff Matrix (familiar?)
- End of year reviews

Partner You	C	D
C	3	0
D	5	1

› So, what's your strategy to succeed?

# Defectors & Cooperators

```
players = [axl.Cooperator() for i in range(5)] + [axl.Defector() for i in range(5)]  
mp = axl.MoranProcess(players, turns=12)  
populations = mp.play()
```

**Creating Players and  
the Moran Process**

# Defectors & Cooperators

```
players = [axl.Cooperator() for i in range(5)] + [axl.Defector() for i in range(5)]
```

```
mp = axl.MoranProcess(players, turns=12)
```

```
populations = mp.play()
```

```
populations[0]
```

```
Counter({'Cooperator': 5, 'Defector': 5})
```

**Starting with 50%-50%  
What will happen?**

# Defectors & Cooperators

```
players = [axl.Cooperator() for i in range(5)] + [axl.Defector() for i in range(5)]
```

```
mp = axl.MoranProcess(players, turns=12)
```

```
populations = mp.play()
```

```
populations[0]
```

```
Counter({'Cooperator': 5, 'Defector': 5})
```

```
mp.winning_strategy_name
```

```
'Defector'
```

```
[p['Defector'] for p in populations]
```

```
[5, 6, 6, 7, 7, 6, 6, 7, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10]
```

**Defectors win**  
**It takes time because of**  
**the randomness**

# Cooperators, Defectors and TFT

```
players = [axl.Cooperator() for i in range(9)] +  
           [axl.Defector() for i in range(3)] +  
           [axl.TitForTat() for i in range(3)]  
mp = axl.MoranProcess(players, turns=12)  
populations = mp.play()
```

**9 Cooperators, 3 Defectors, 3 TFT**  
**What will happen now?**

# Cooperators, Defectors and TFT

```
players = [axl.Cooperator() for i in range(9)] +  
           [axl.Defector() for i in range(3)] +  
           [axl.TitForTat() for i in range(3)]
```

```
mp = axl.MoranProcess(players, turns=12)
```

```
populations = mp.play()
```

```
mp.winning_strategy_name
```

```
'Tit For Tat'
```

```
pop3 = populations[:,3]
```

**Let's follow the results...**

```
[p['Defector'] for p in pop3];[p['Cooperator'] for p in pop3];[p['Tit For Tat'] for p in pop3]
```

```
D: [3, 5, 6, 7, 7, 10, 11, 10, 9, 7, 5, 4, 3, 3, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
C: [9, 9, 8, 6, 5, 3, 2, 2, 2, 1, 1, 1, 1, 1, 3, 4, 4, 5, 5, 4, 5, 6, 5, 6, 4, 5, 4, 4, 3, 4, 3, 2, 0]  
T: [3, 1, 1, 2, 3, 2, 2, 3, 4, 7, 9, 10, 11, 11, 11, 10, 10, 10, 10, 11, 10, 9, 10, 9, 11, 10, 11, 11, 12, 11, 12, 13, 15]
```





## More Options

- › More strategies
  - Over 200 included in the package
  - Strategy transformers
  - Human players

## More Options

### › More strategies

- Over 200 included in the package
- Strategy transformers
- Human players

### › More games

- Just insert values for the payoff matrix  
`chicken = axl.game.Game(r=0, s=-1, t=1, p=-10)`

## More Options

- › More strategies
  - Over 200 included in the package
  - Strategy transformers
  - Human players
- › More games
  - Just insert values for the payoff matrix  
`chicken = axl.game.Game(r=0, s=-1, t=1, p=-10)`
- › More cool stuff
  - Noise
  - Game length and player knowledge



# Questions?

Omer Nevo  
NeoWize