2/3rds of the average project

$$x = (x_0, x_1, \ldots x_{100})$$

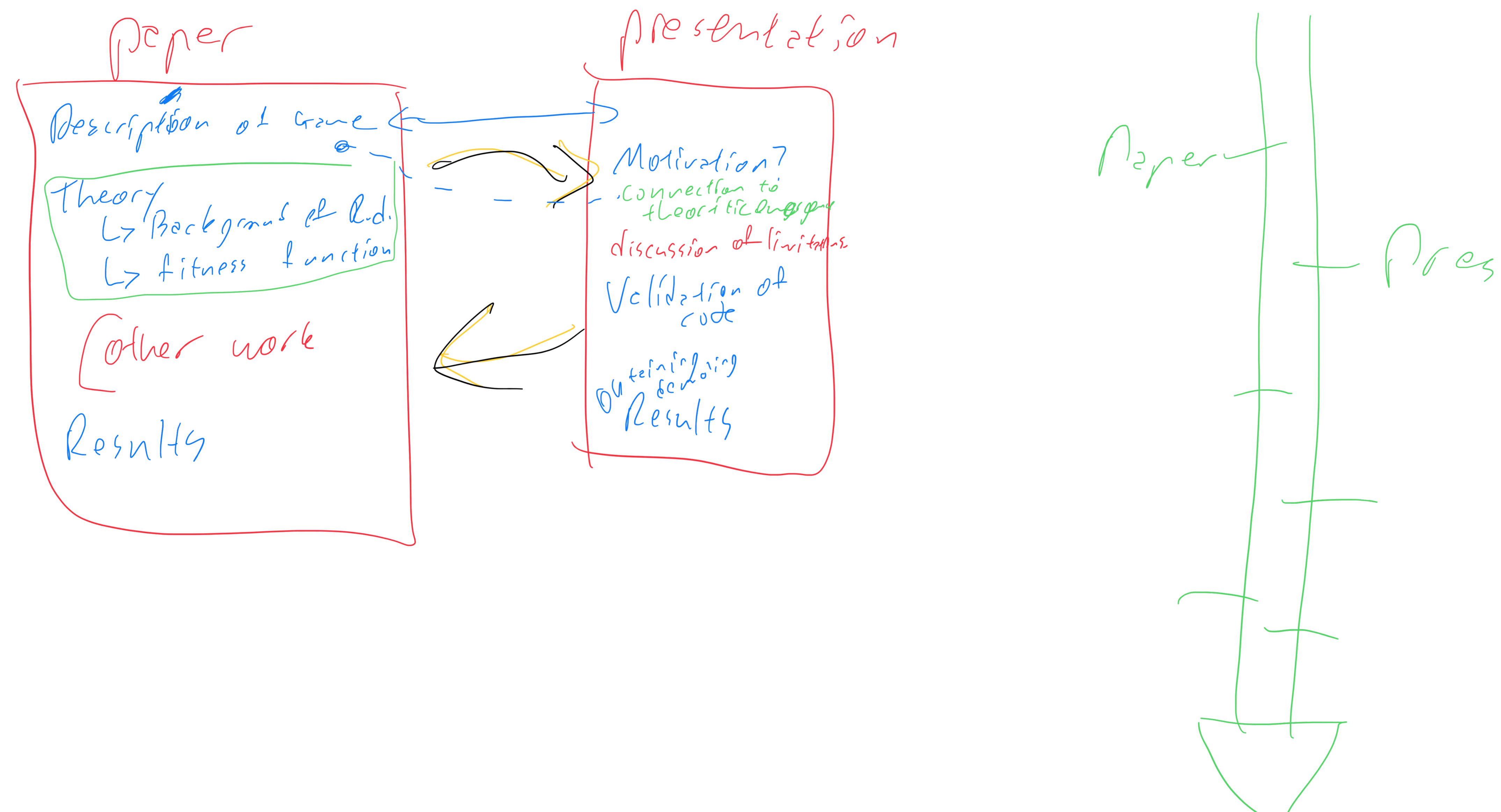
$$f = (f_0, f_1, \ldots f_{100})$$

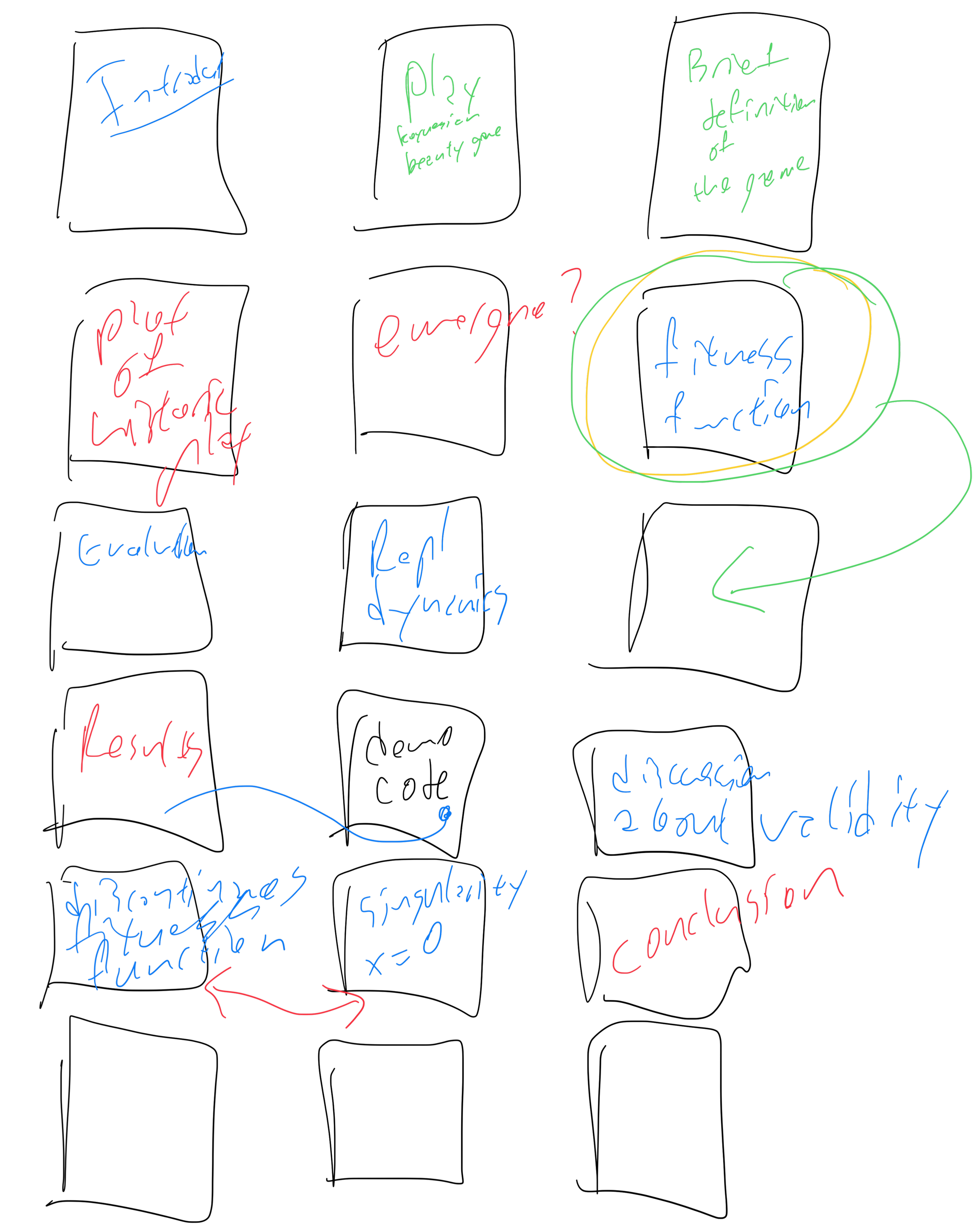
$$x = (1, 0, 0, \ldots 0)$$

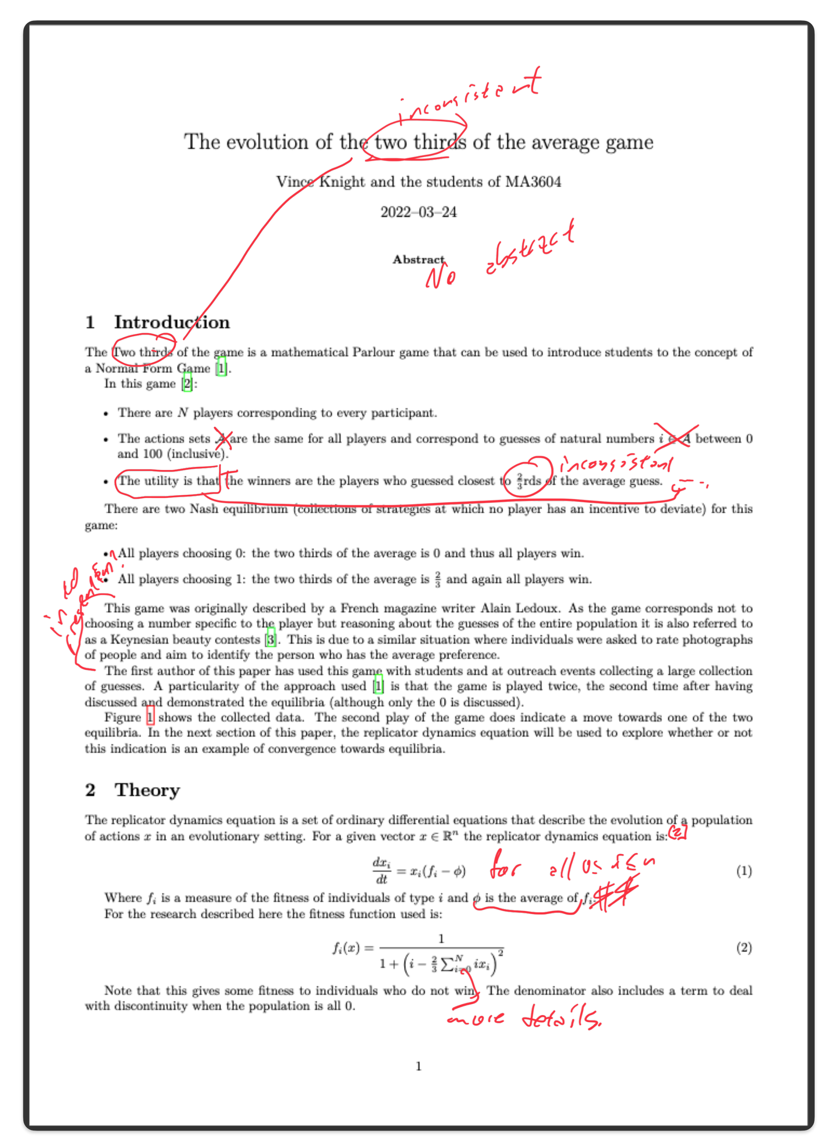
$$f_0 = \text{Big}$$

$$f_i = 0 \text{ if } 0$$

$$\frac{2}{3} \sum x_i i$$







The evolution of the two thirds of the average game

Vince Knight and the students of MA3604

2022-03-24

Abstract

1 Introduction

The Two thirds of the game is a mathematical Parlour game that can be used to introduce students to the concept of a Normal Form Game [1].

In this game [2]:

- There are N players corresponding to every participant.
- The actions sets A are the same for all players and correspond to guesses of natural numbers $i \in A$ between 0 and 100 (inclusive).
- The utility is that the winners are the players who guessed closest to $\frac{2}{3}$rds of the average guess.

There are two Nash equilibrium (collections of strategies at which no player has an incentive to deviate) for this game:

- All players choosing 0: the two thirds of the average is 0 and thus all players win.
- All players choosing 1: the two thirds of the average is $\frac{2}{3}$ and again all players win.

This game was originally described by a French magazine writer Alain Ledoux. As the game corresponds not to choosing a number specific to the player but reasoning about the guesses of the entire population it is also referred to as a Keynesian beauty contests [3]. This is due to a similar situation where individuals were asked to rate photographs of people and aim to identify the person who has the average preference.

The first author of this paper has used this game with students and at outreach events collecting a large collection of guesses. A particularity of the approach used [1] is that the game is played twice, the second time after having discussed and demonstrated the equilibria (although only the 0 is discussed).

Figure 1 shows the collected data. The second play of the game does indicate a move towards one of the two equilibria. In the next section of this paper, the replicator dynamics equation will be used to explore whether or not this indication is an example of convergence towards equilibria.

2 Theory

The replicator dynamics equation is a set of ordinary differential equations that describe the evolution of a population of actions x in an evolutionary setting. For a given vector $x \in \mathbb{R}^n$ the replicator dynamics equation is:

$$\frac{dx_i}{dt} = x_i(f_i - \phi) \quad (1)$$

Where f_i is a measure of the fitness of individuals of type i and ϕ is the average of f_i.

For the research described here the fitness function used is:

$$f_i(x) = \frac{1}{1 + \left(i - \frac{2}{3}\sum_{i=0}^{N} i x_i\right)^2} \quad (2)$$

Note that this gives some fitness to individuals who do not win. The denominator also includes a term to deal with discontinuity when the population is all 0.

(Annotations: inconsistent; No abstract; inconsistent; in the references?; for all $0 \le i \le n$; more details.)

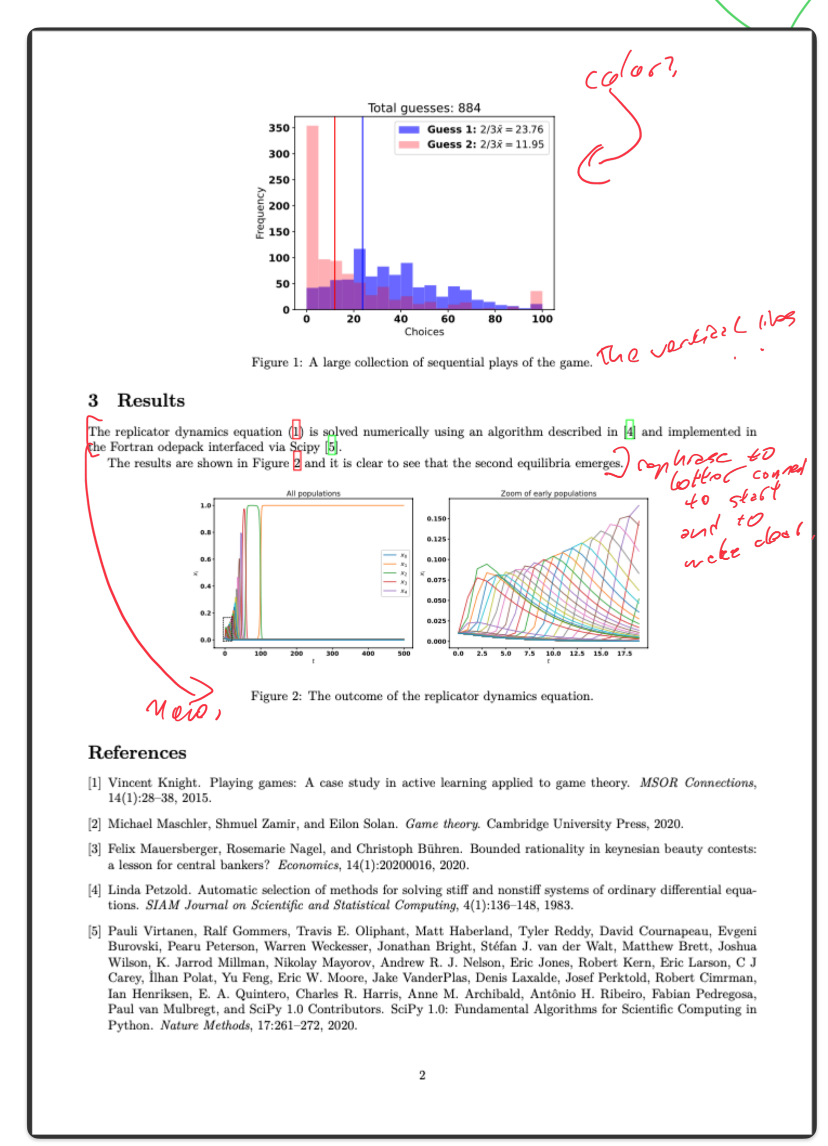


Figure 1: A large collection of sequential plays of the game.

3 Results

The replicator dynamics equation (1) is solved numerically using an algorithm described in [4] and implemented in the Fortran odepack interfaced via Scipy [5].

The results are shown in Figure 2 and it is clear to see that the second equilibria emerges.

Figure 2: The outcome of the replicator dynamics equation.

References

[1] Vincent Knight. Playing games: A case study in active learning applied to game theory. *MSOR Connections*, 14(1):28–38, 2015.

[2] Michael Maschler, Shmuel Zamir, and Eilon Solan. *Game theory*. Cambridge University Press, 2020.

[3] Felix Mauersberger, Rosemarie Nagel, and Christoph Bühren. Bounded rationality in keynesian beauty contests: a lesson for central bankers? *Economics*, 14(1):20200016, 2020.

[4] Linda Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM Journal on Scientific and Statistical Computing*, 4(1):136–148, 1983.

[5] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

(Annotations: colors; The vertical lines...; rephrase to better connect to start and to make clear; Merge.)