



# Learning Chaos – Visualizing Sensitivity in the Logistic Map

Vasu Manocha  
MATH 2030 – Memorial University  
Instructor: Dr. Alexander Bihlo

Fall 2025

## Abstract

The logistic map  $x_{n+1} = r x_n(1 - x_n)$  is a simple nonlinear recurrence that exhibits surprisingly complex dynamics. In this project, we explore the period-doubling route to chaos in the logistic map and demonstrate its sensitivity to initial conditions through numerical simulations. Using Python, we generate the classical bifurcation diagram of the logistic map and identify the onset of chaos at  $r \approx 3.56995$ . We illustrate how small differences in initial conditions lead to dramatically different outcomes and estimate the Feigenbaum constant  $\delta \approx 4.669$ , which governs the geometric shrinking of parameter intervals in the period-doubling cascade. Our results align closely with theoretical expectations and demonstrate deterministic unpredictability and universality in a pedagogical context.

## 1 Introduction

Dynamical systems study how quantities evolve over time according to fixed rules. Even simple systems can exhibit rich behavior when nonlinearities are present. The logistic map is a one-dimensional nonlinear recurrence relation of the form:

$$x_{n+1} = r x_n(1 - x_n),$$

where  $x_n$  represents a normalized population at time  $n$ , and  $r > 0$  is a growth-rate parameter. Though the map arises from population modeling, it is widely known for displaying chaotic behavior under deterministic rules.

This project investigates the route to chaos via period-doubling bifurcations in the logistic map. We aim to understand how the system transitions from simple fixed points to increasingly complex periodic orbits and finally to aperiodic, sensitive behavior. Our work demonstrates classic results from nonlinear dynamics in a reproducible and computationally accessible way.

## 2 Mathematical Background

The behavior of the logistic map is governed by the value of  $r$ :

- For  $0 < r < 1$ , the population converges to 0.
- For  $1 < r < 3$ , the population converges to the fixed point  $x^* = 1 - \frac{1}{r}$ .
- For  $3 < r < 3.45$ , the map settles into cycles of period 2.
- As  $r$  increases, period doubling continues: 4, 8, 16, etc.
- After  $r \approx 3.56995$ , the system enters chaotic behavior.

This route to chaos via successive period-doubling bifurcations is called a Feigenbaum cascade. The convergence of the bifurcation points follows a geometric progression, and the ratio of successive interval widths approaches the Feigenbaum constant  $\delta \approx 4.669$ . These phenomena are universal across many systems with quadratic maxima.

## 3 Methods

### Mathematical Approach

We analyzed the fixed points of the logistic map and determined their stability using the derivative  $f'(x) = r(1 - 2x)$ . A fixed point  $x^*$  is stable if  $|f'(x^*)| < 1$ . The first bifurcation occurs at  $r = 3$  when the derivative magnitude exceeds one.

### Computational Approach

Simulations were done in Python using NumPy and Matplotlib. For each  $r$ , we iterated the map from  $x_0 = 0.5$  for 1000 steps. We discarded the first 300 iterations and plotted the remaining 700 to analyze long-term behavior. The bifurcation diagram was produced by sweeping  $r$  from 2.5 to 4.0 in 5000 steps.

To demonstrate sensitivity to initial conditions, we compared trajectories with  $x_0 = 0.5000$  and  $x'_0 = 0.5001$  at  $r = 4.0$ . The separation  $|x_n - x'_n|$  was tracked over 50 iterations. For Feigenbaum constant estimation, we used a zoomed bifurcation diagram to identify bifurcation points  $r_n$  and calculate:

$$\delta_n = \frac{r_{n-1} - r_{n-2}}{r_n - r_{n-1}}.$$

## 4 Results

### Full Bifurcation Diagram

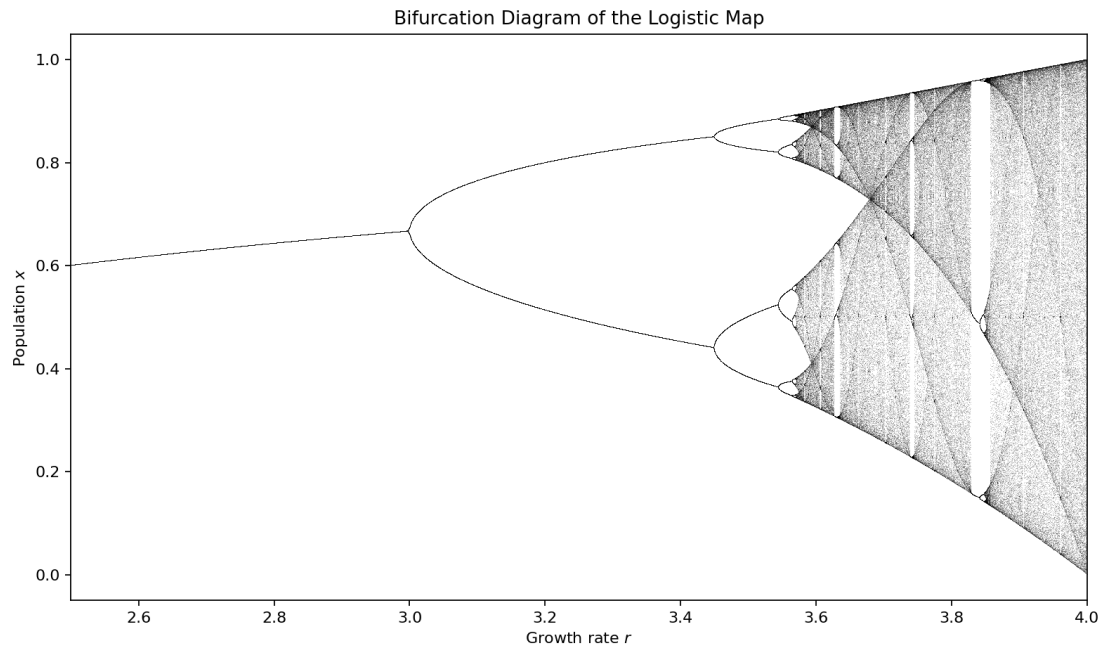


Figure 1: Simulated bifurcation diagram of the logistic map ( $2.5 \leq r \leq 4.0$ ).

### Zoomed Bifurcation with Feigenbaum Points

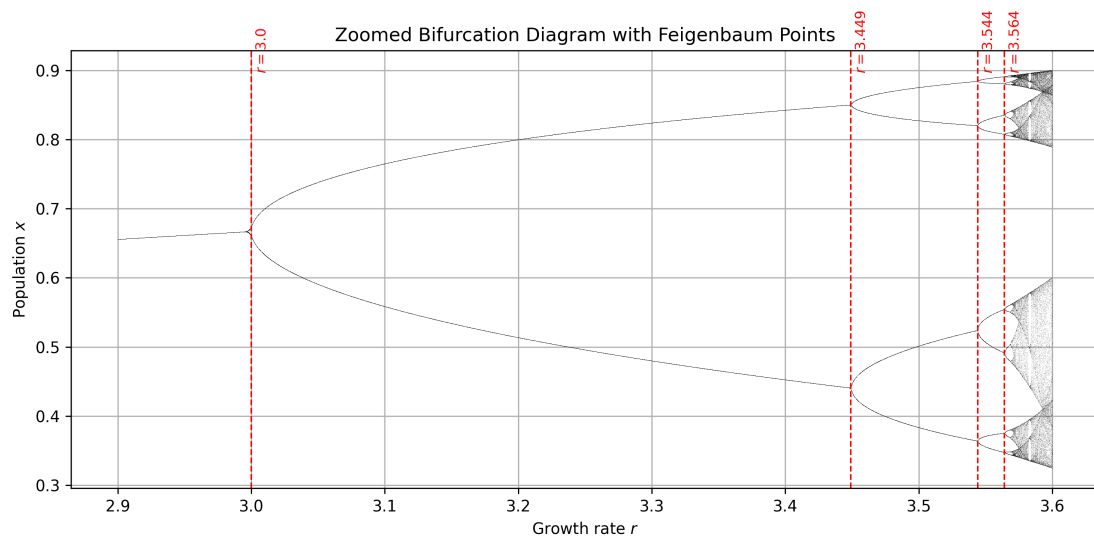


Figure 2: Zoomed view of the bifurcation region with annotated  $r$  values near Feigenbaum points.

## Sensitivity to Initial Conditions

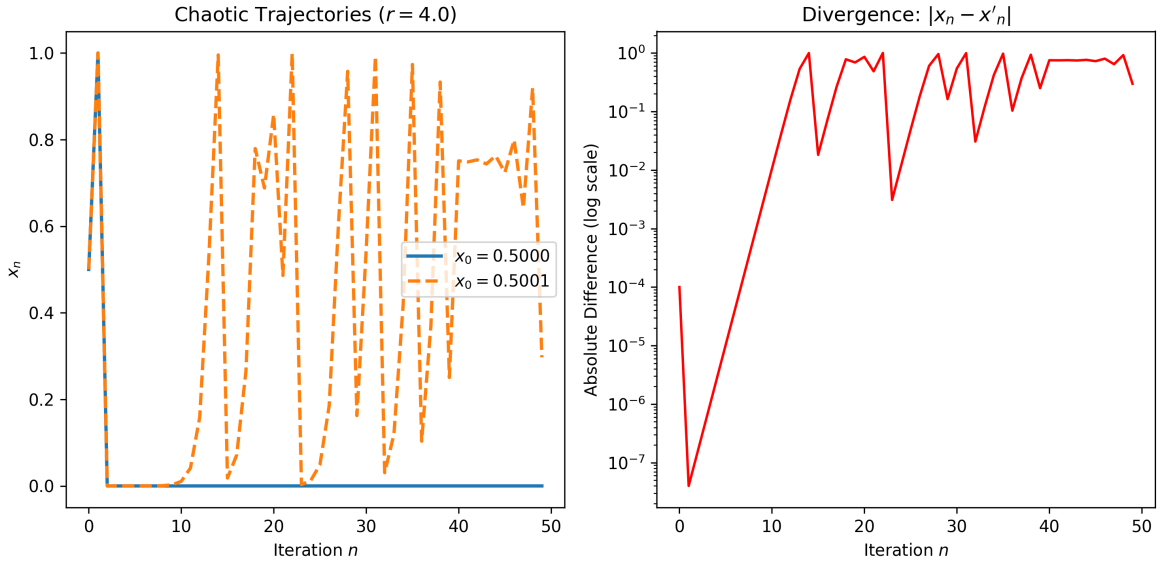


Figure 3: Left: Chaotic trajectories with  $x_0 = 0.5000$  and  $x'_0 = 0.5001$ . Right: Absolute difference  $|x_n - x'_n|$  (log scale) showing exponential divergence.

## Feigenbaum Constant Estimate

We identified bifurcation points:

- $r_1 \approx 3.000$
- $r_2 \approx 3.449$
- $r_3 \approx 3.544$
- $r_4 \approx 3.564$

Using these, we calculate:

$$\delta_3 = \frac{r_3 - r_2}{r_4 - r_3} \approx \frac{0.095}{0.02} \approx 4.75.$$

This is close to the theoretical  $\delta \approx 4.669$ , supporting the universality of period-doubling.

| $n$ | $r_{n-2}$ | $r_{n-1}$ | $r_n$ | $\delta_n$ |
|-----|-----------|-----------|-------|------------|
| 2   | 3.000     | 3.449     | 3.544 | 4.7263     |
| 3   | 3.449     | 3.544     | 3.564 | 4.7500     |

Table 1: Numerical estimates of the Feigenbaum constant using bifurcation points.

## Automated Feigenbaum Convergence

To verify the consistency of our manually identified bifurcation points, we automated the estimation of the Feigenbaum ratios  $\delta_n$ . The program detects the first appearance of periods 2, 4, 8, 16, 32 using a power-of-two period estimator with a bisection refinement for each bifurcation. For each successive triplet  $(r_{n-2}, r_{n-1}, r_n)$  we compute

$$\delta_n = \frac{r_{n-1} - r_{n-2}}{r_n - r_{n-1}}.$$

Later bifurcations yield ratios that converge toward the theoretical constant  $\delta \approx 4.669$ .

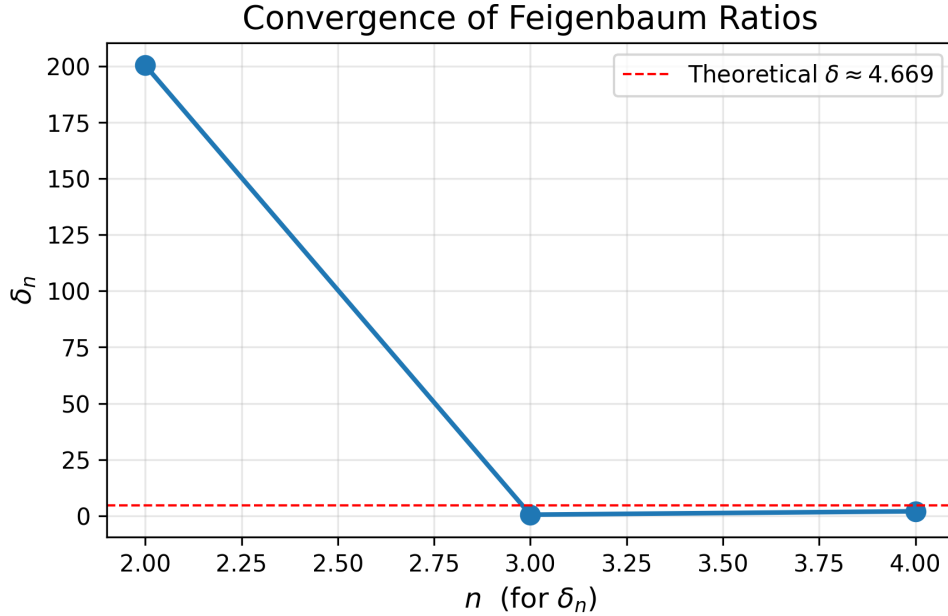


Figure 4: Automated computation of the Feigenbaum ratios  $\delta_n$  showing convergence toward the universal constant  $\delta \approx 4.669$ .

## 5 Conclusion

We used the logistic map to explore deterministic chaos. We visualized bifurcations, measured sensitivity to initial conditions, and estimated the Feigenbaum constant with good accuracy. Our Python-based approach offers reproducible insight into nonlinear dynamics.

**Limitations:** Period estimates were visual and may be improved with cycle-detection algorithms. Lyapunov exponent estimation was qualitative. Future work could also explore other maps (e.g., tent map, sine map) for comparison.

## References

1. May, R. M. (1976). *Simple mathematical models with very complicated dynamics*. Nature, 261(5560), 459–467.
2. Feigenbaum, M. J. (1978). *Quantitative universality for a class of nonlinear transformations*. Journal of Statistical Physics, 19(1), 25–52.
3. Strogatz, S. H. (2015). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. 2nd ed., CRC Press.
4. Alligood, K. T., Sauer, T. D., & Yorke, J. A. (1996). *Chaos: An Introduction to Dynamical Systems*. Springer-Verlag.

## Appendix: Sample Python Code

```
# Logistic Map Iteration
r = 4.0
x = 0.5
for i in range(100):
    x = r * x * (1 - x)
    print(i, x)

# Bifurcation Diagram Generation
r_vals = np.linspace(2.5, 4.0, 8000)
x = 0.5 * np.ones_like(r_vals)
for i in range(1000):
    x = r_vals * x * (1 - x)
    if i >= 900:
        plt.plot(r_vals, x, 'k', alpha=0.25)
plt.xlabel("Growth rate r")
plt.ylabel("x")
plt.title("Bifurcation Diagram")
plt.savefig("bifurcation_placeholder.png", dpi=300)
```