1D-Discrete Dynamical Systems: Logistic Map Analysis (Assignment Sheet 5a)

Introduction to Chaos Applied to Systems, Processes, and Products (ETSIDI, UPM)

Alfonso Allen-Perkins, Juan Carlos Bueno, and Eduardo Faleiro

2025-03-20

Contents

Introduction

In this document, we explore the **logistic map** given by

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

where μ is a parameter in the range $[0, 4]$. We will:

1. **Iterate** the map for various initial conditions and values of μ to see how the orbits evolve.
2. **Plot cobweb diagrams** to observe the iteration steps graphically.
3. **Construct the bifurcation diagram** to visualize the long-term (asymptotic) values of x_n for a range of μ.

Before starting, load the necessary libraries:

```
knitr::opts_chunk$set(
  echo = TRUE,         # Show the code in the output
  message = FALSE,     # Hide library/package messages
  warning = FALSE      # Hide warnings
)
```

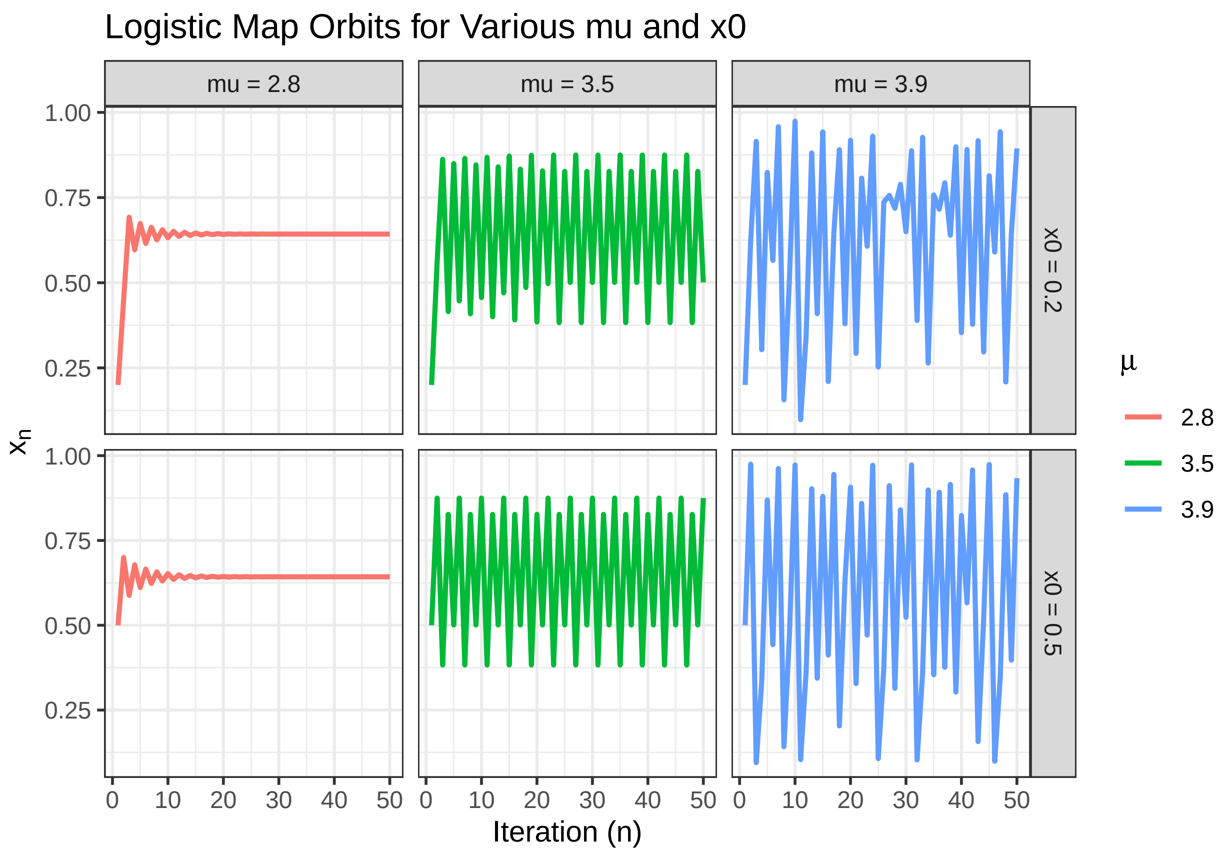
```r
# Load necessary libraries
library(ggplot2)       # For general plotting
library(tidyverse)     # For data manipulation and plotting

# Logistic map function
logistic_map <- function(x, mu) {
  mu * x * (1 - x)
}

# Generate a time series (orbit) of length n for a given mu and initial x0
iterate_logistic <- function(mu, x0, n) {
  x_values <- numeric(n)
  x_values[1] <- x0
  for (i in 2:n) {
    x_values[i] <- logistic_map(x_values[i - 1], mu)
  }
  # Return a data frame with iteration index and x-value
  data.frame(
    iteration = 1:n,
    x = x_values,
    mu = mu,
    x0 = x0
  )
}
```

1. Logistic Map Orbits

Here we iterate the logistic map for a few values of μ and different initial conditions x_0. We then plot the resulting orbits to see how the system evolves over time.

```r
# Parameters
n <- 50                       # Number of iterations
x0_values <- c(0.2, 0.5)     # Different initial conditions
mu_values <- c(2.8, 3.5, 3.9)  # Different mu's

# Generate orbits
orbit_data <- data.frame()
for (mu in mu_values) {
  for (x0 in x0_values) {
    orbit_data <- rbind(orbit_data, iterate_logistic(mu, x0, n))
  }
}

# Convert mu and x0 to labels for faceting
orbit_data$mu_label  <- paste0("mu = ", orbit_data$mu)
orbit_data$x0_label  <- paste0("x0 = ", orbit_data$x0)

ggplot(orbit_data, aes(x = iteration, y = x, color = as.factor(mu))) +
  geom_line(linewidth = 0.9) +
  facet_grid(x0_label ~ mu_label, scales = "free_y") +
  xlab("Iteration (n)") +
  ylab(expression(x[n])) +
  labs(color = expression(mu)) +
```

```r
  ggtitle("Logistic Map Orbits for Various mu and x0") +
  theme_bw()
```



Observations

- For some values of μ, the sequence rapidly converges to a fixed point.
- For larger μ, it might oscillate or show more complex behavior (e.g., periodic or even chaotic).

---

2. Cobweb Plots

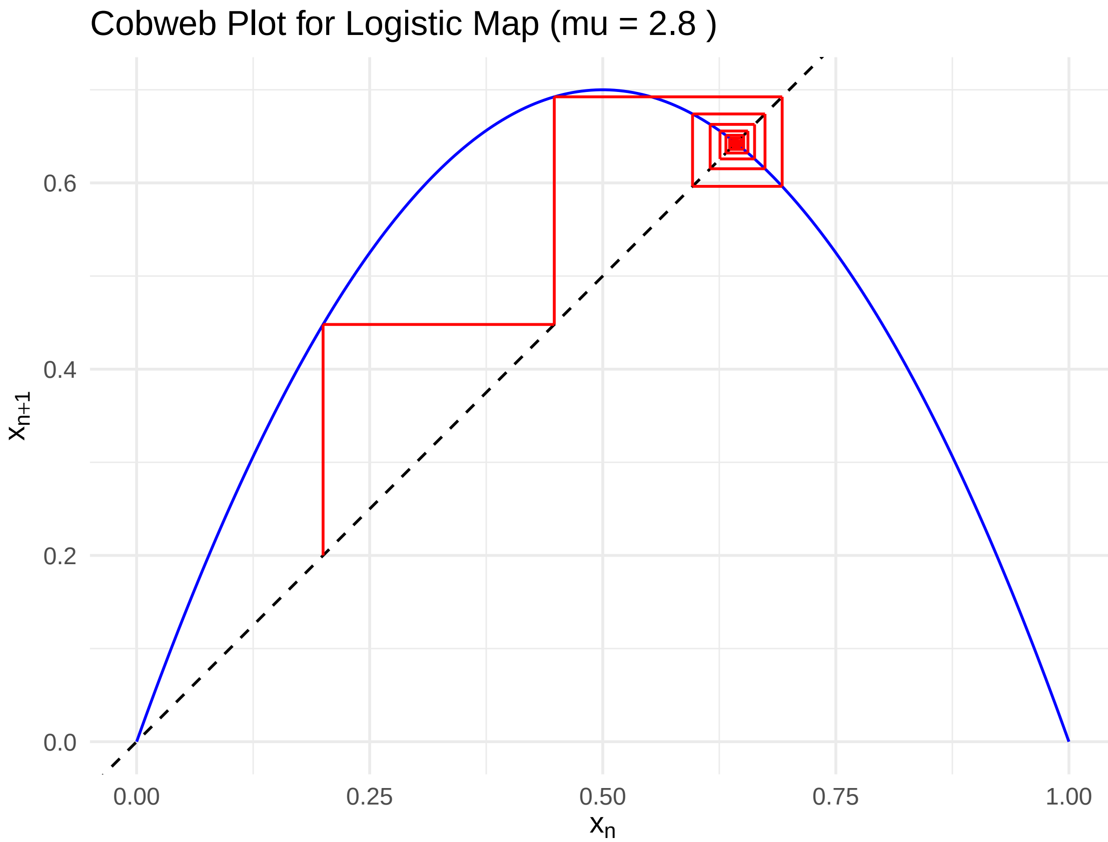
Cobweb plots visualize the iterative process: 1. We start at $(x_0, 0)$. 2. Move **vertically** to the curve $(x_0, f(x_0))$. 3. Then **horizontally** to $(f(x_0), f(x_0))$. 4. Repeat, producing a "cobweb" that reveals whether the iteration converges to a point, a cycle, or diverges.

Try different values of μ to see how the map's dynamics change.

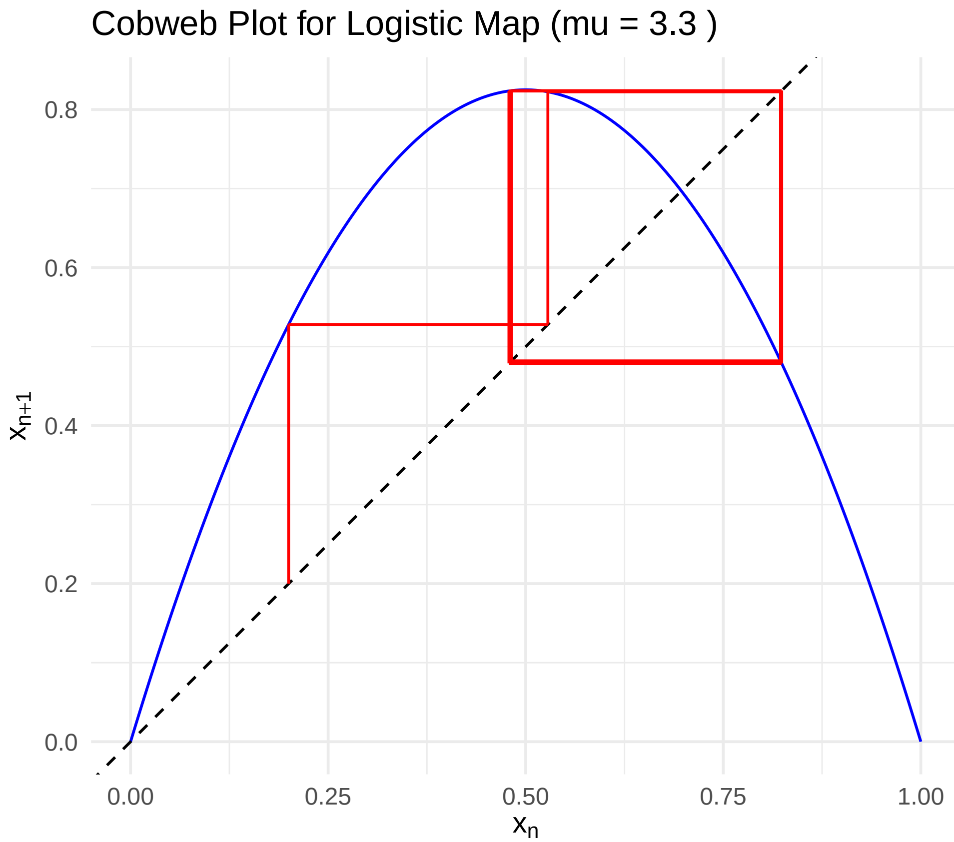
```r
# Cobweb plot function
cobweb_plot <- function(mu, x0 = 0.2, n_iter = 50) {

  # Generate iterative points
  x_vals <- numeric(n_iter)
  x_vals[1] <- x0
```

```r
  for (i in 2:n_iter) {
    x_vals[i] <- logistic_map(x_vals[i - 1], mu)
  }

  # Data for the logistic curve f(x) = mu*x*(1 - x)
  curve_data <- data.frame(
    x = seq(0, 1, length.out = 200)
  )
  curve_data$fx <- logistic_map(curve_data$x, mu)

  # Data for the cobweb lines
  cobweb_segments <- data.frame(
    x_start = x_vals[-length(x_vals)],
    x_end   = x_vals[-length(x_vals)],
    y_start = x_vals[-length(x_vals)],
    y_end   = x_vals[-1]
  )

  # We'll pair each vertical line with a horizontal line from (x_end, y_end).
  # The second set is for the horizontal segments:
  cobweb_segments2 <- data.frame(
    x_start = x_vals[-length(x_vals)],
    x_end   = x_vals[-1],
    y_start = x_vals[-1],
    y_end   = x_vals[-1]
  )

  # Plot
  ggplot() +
    geom_line(data = curve_data, aes(x = x, y = fx), color = "blue") +       # logistic curve
    geom_abline(slope = 1, intercept = 0, linetype = "dashed") +            # y = x line
    geom_segment(data = cobweb_segments,
                 aes(x = x_start, xend = x_end, y = y_start, yend = y_end),
                 color = "red") +                                           # vertical
    geom_segment(data = cobweb_segments2,
                 aes(x = x_start, xend = x_end, y = y_start, yend = y_end),
                 color = "red") +                                           # horizontal
    labs(title = paste("Cobweb Plot for Logistic Map (mu =", mu, ")"),
         x = expression(x[n]),
         y = expression(x[n+1])) +
    coord_fixed(ratio = 1) +
    theme_minimal()
}
```

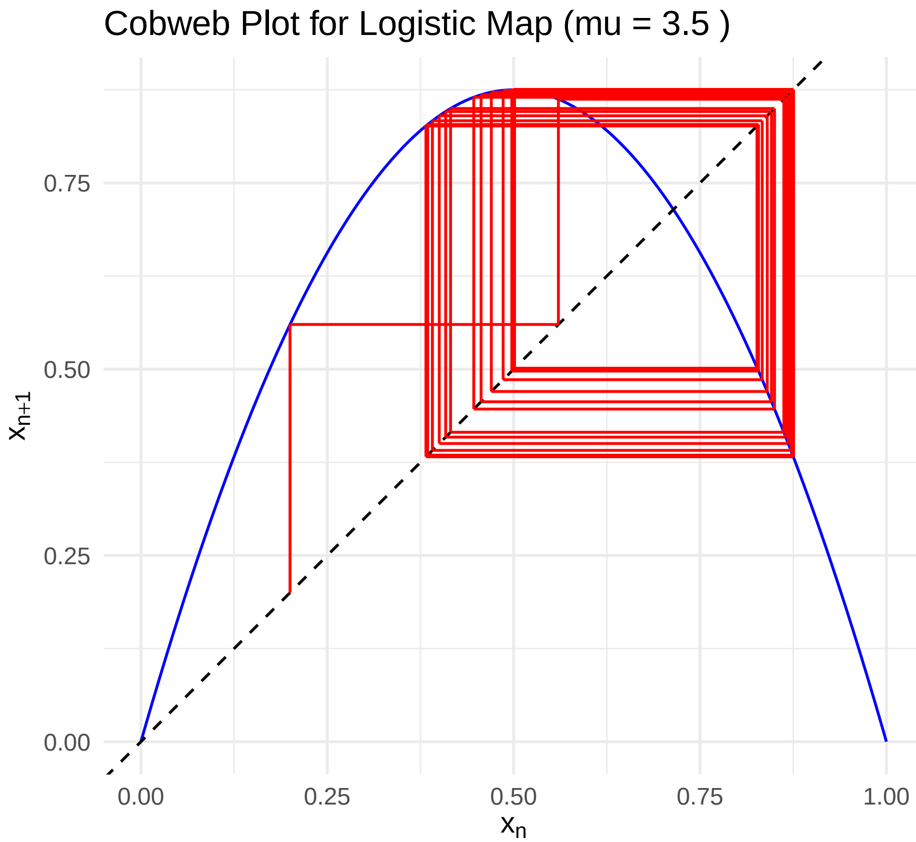
```r
# Example cobweb for mu = 2.8
cobweb_plot(mu = 2.8, x0 = 0.2, n_iter = 20)
```



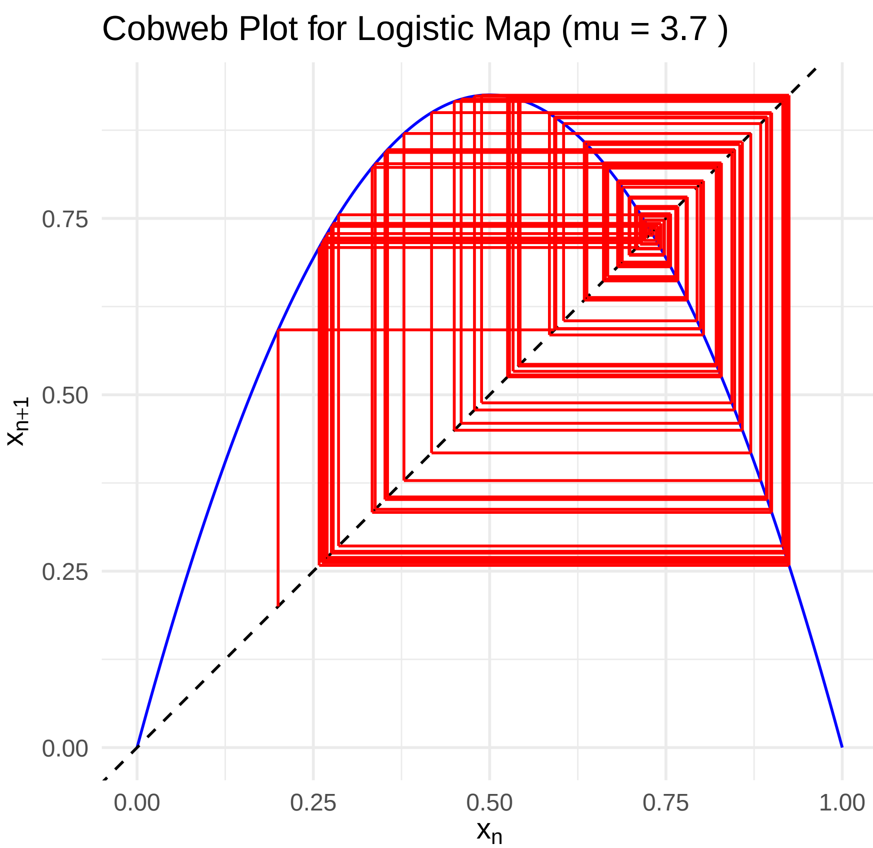

```
# Example cobweb for mu = 3.3
cobweb_plot(mu = 3.3, x0 = 0.2, n_iter = 50)
```




```r
# Example cobweb for mu = 3.5
cobweb_plot(mu = 3.5, x0 = 0.2, n_iter = 100)
```




```
# Example cobweb for mu = 3.7
cobweb_plot(mu = 3.7, x0 = 0.2, n_iter = 100)
```



Observations

- For smaller μ, the diagram typically shows convergence to a single fixed point.
- As μ increases, you might see 2-cycle, 4-cycle, or more complex chaotic behavior.

---

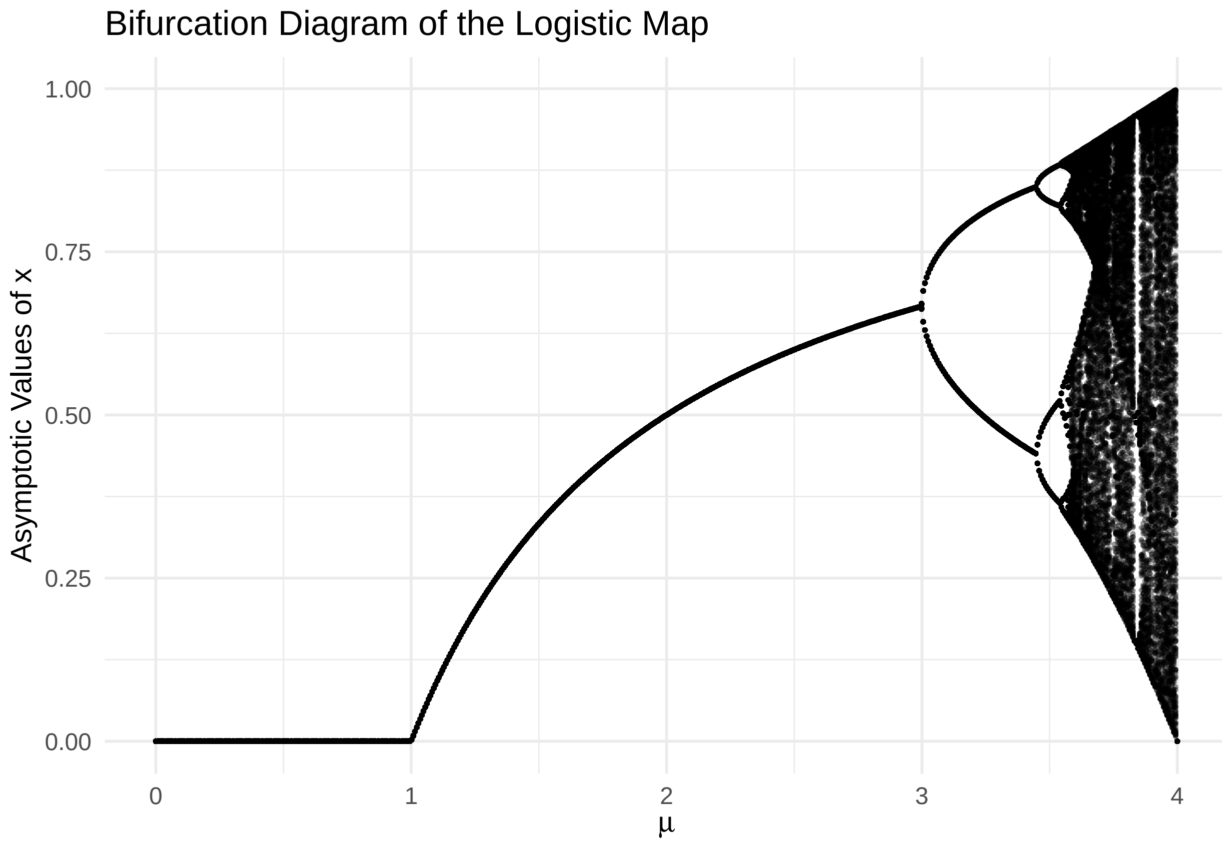
3. Bifurcation Diagram

We now generate the classic **bifurcation diagram** for the logistic map, plotting the long-term (asymptotic) values of x_n for a range of μ. We'll discard the initial transient iterations and plot only the last iterations, where the system is close to its attractor (fixed point, cycle, or chaotic set).

```
# Function to generate a bifurcation dataset
generate_bifurcation_data <- function(mu_min, mu_max, mu_steps,
                                      x0 = 0.5,  # initial condition
                                      n_iter = 1000,  # total iterations
                                      n_keep = 300) { # how many to keep (post-transient)
  mu_values <- seq(mu_min, mu_max, length.out = mu_steps)
  bif_data <- data.frame()

  for (mu in mu_values) {
    x <- numeric(n_iter)
    x[1] <- x0
```

```r
    # Iterate
    for (i in 2:n_iter) {
      x[i] <- logistic_map(x[i - 1], mu)
    }
    # Discard the first (n_iter - n_keep) points
    # Keep only the tail where the orbit "settled"
    keep_x <- tail(x, n_keep)

    temp_df <- data.frame(
      mu = rep(mu, n_keep),
      x = keep_x
    )
    bif_data <- rbind(bif_data, temp_df)
  }

  bif_data
}
```

```r
# Generate data
bif_data <- generate_bifurcation_data(mu_min = 0, mu_max = 4, mu_steps = 600,
                                      x0 = 0.5, n_iter = 1000, n_keep = 300)

# Plot
ggplot(bif_data, aes(x = mu, y = x)) +
  geom_point(size = 0.3, alpha = 0.4) +
  labs(title = "Bifurcation Diagram of the Logistic Map",
       x = expression(mu),
       y = "Asymptotic Values of x") +
  theme_minimal()
```



Observations

- Around $\mu \approx 3$, the orbit transitions from a stable fixed point to a stable 2-cycle.
- Further increasing μ leads to repeated **period doubling** until chaos begins.
- In the chaotic regime, windows of periodic behavior still appear.

Conclusions

1. **Orbits**: By iterating for specific μ-values, we can see that smaller μ typically settles into a single fixed point, while larger μ can exhibit periodic or chaotic behavior.
2. **Bifurcation Diagram**: Summarizes the long-term (asymptotic) behavior for a continuous range of μ. The logistic map is a classic example of a route to chaos through period doubling.