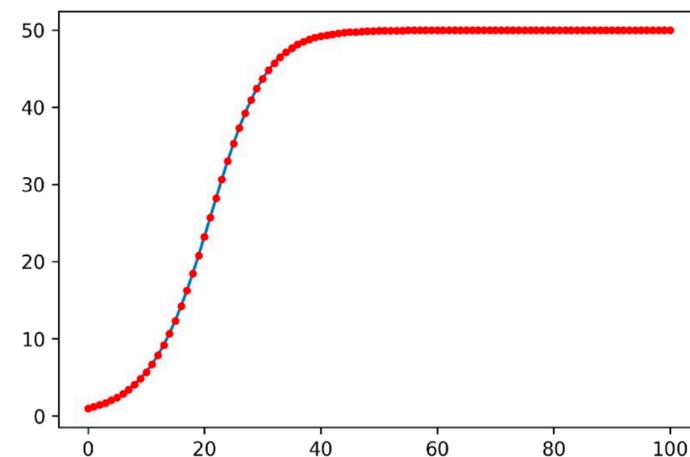
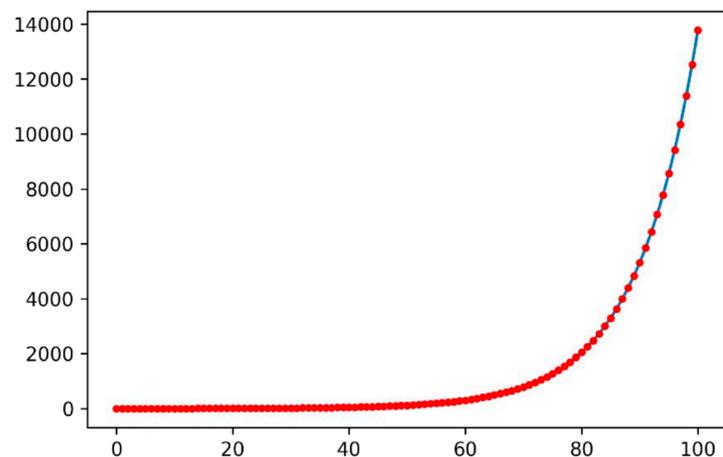


数理生物学演習

第3回 個体群動態の数理モデル（1）：離散ロジスティック成長モデル



野下 浩司 (Noshita, Koji)

✉ noshita@morphometrics.jp

🏠 <https://koji.noshita.net>

理学研究院 数理生物学研究室

第3回：個体群動態の数理モデル（1）： 離散指数増殖，離散ロジスティック成長モデル

今回の目標

- 差分方程式を解く
- 時間発展をプロットする

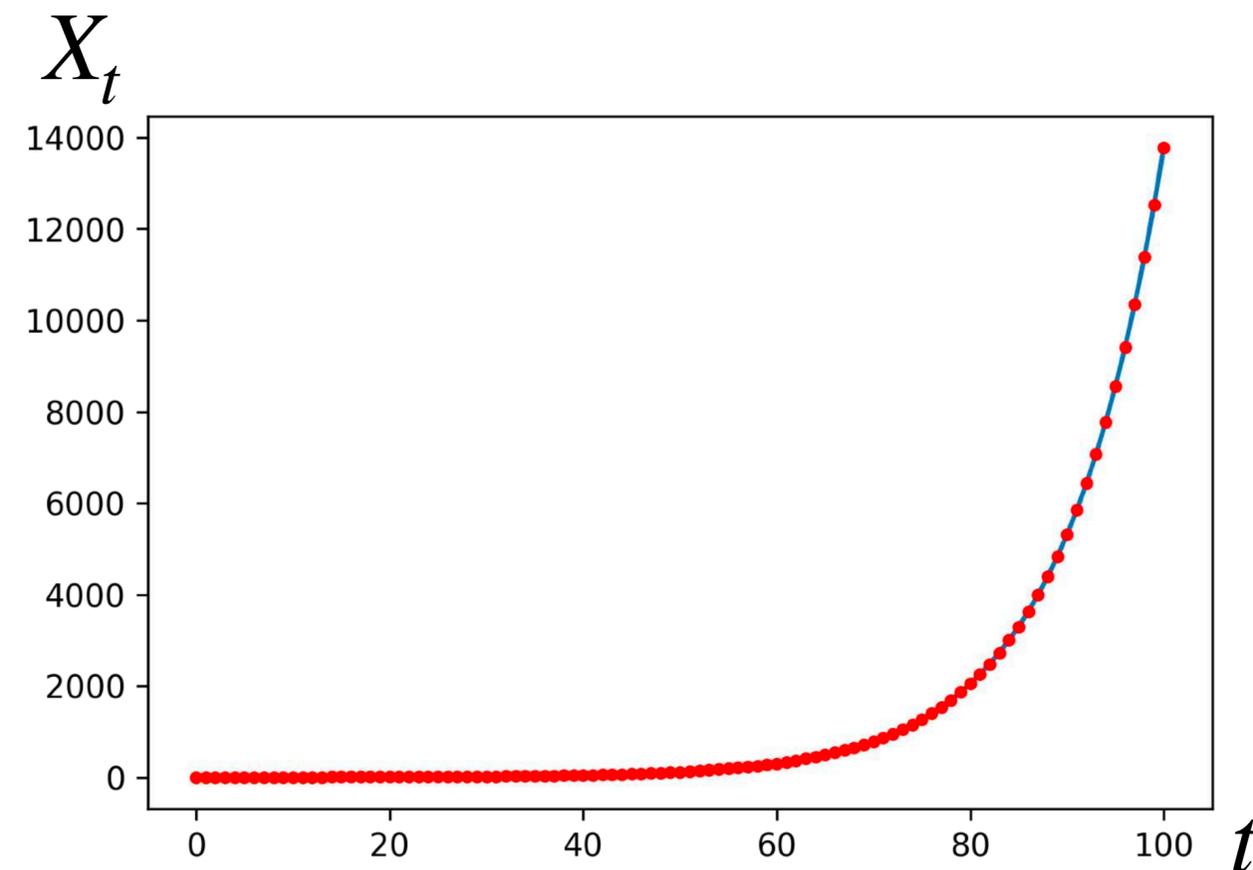
個体群動態（離散時間）

指数増殖・ロジスティック成長モデル

離散時間指数増殖モデル

$$X_{t+1} = X_t + aX_t$$

a : マルサス係数. 1世代あたりの増殖率. $a \geq 0$.

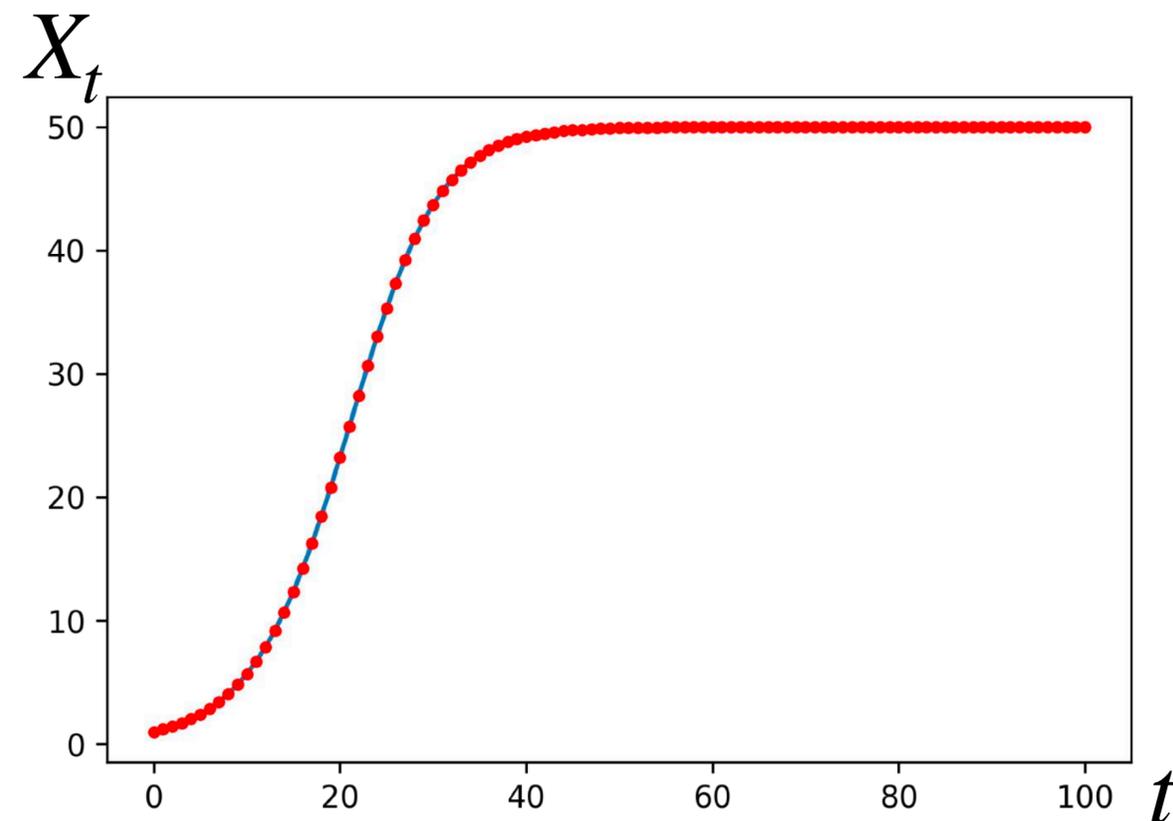


離散時間ロジスティック成長モデル

$$X_{t+1} = X_t + r \left(1 - \frac{X_t}{K} \right) X_t$$

r : 内的自然増加率. 個体数が十分小さい場合 ($X \approx 0$) の1世代あたりの増殖率. $r \geq 0$.

K : 環境収容力. ある環境で維持されうる個体数, $K > 0$.



平衡点と局所安定性

時間的に変化しない（釣り合いが取れている）状態を見つけ、その安定性を調べる

平衡点： $X_{t+1} = X_t = \bar{X}$ となるような \bar{X}

例. 離散ロジスティック成長モデル

$$X_{t+1} = X_t + r \left(1 - \frac{X_t}{K} \right) X_t$$

モデルの式に代入すれば, $r \left(1 - \frac{\bar{X}}{K} \right) \bar{X} = 0$.

これを満たすが \bar{X} が平衡点になる.

よって, 平衡点は $\bar{X} = 0, K$ の2つ存在する.

局所安定性：平衡点からの微小なずれが生じた際に
どうなるか

$X_{t+1} = f(X_t)$ とし, 微小なずれ n_t を考えテイラー展開する

$$\begin{aligned} \text{と, } \quad \bar{X} + n_{t+1} &= f(\bar{X} + n_t) \\ &= f(\bar{X}) + \frac{df}{dX} n_t + \frac{1}{2} \frac{d^2f}{dX^2} n_t^2 + \dots \end{aligned}$$

\bar{X} は平衡点なので $\bar{X} = f(\bar{X})$. また, n_t は十分小さいので
2次以降の項を無視すると,

$$n_{t+1} \approx \frac{df}{dX} n_t$$

よって, $\left| \frac{df}{dX} \right| < 1$ で安定, $\left| \frac{df}{dX} \right| > 1$ で不安定.

離散ロジスティック成長モデルの場合に2つの平衡点
について計算してみよう!

平衡点と局所安定性

時間的に変化しない（釣り合いが取れている）状態を見つけ、その安定性を調べる

例. 離散ロジスティック成長モデル

$$X_{t+1} = X_t + r \left(1 - \frac{X_t}{K} \right) X_t$$

安定性

$$g(\bar{X}) = r \left(1 - \frac{\bar{X}}{K} \right) \bar{X} \quad \text{とする.}$$

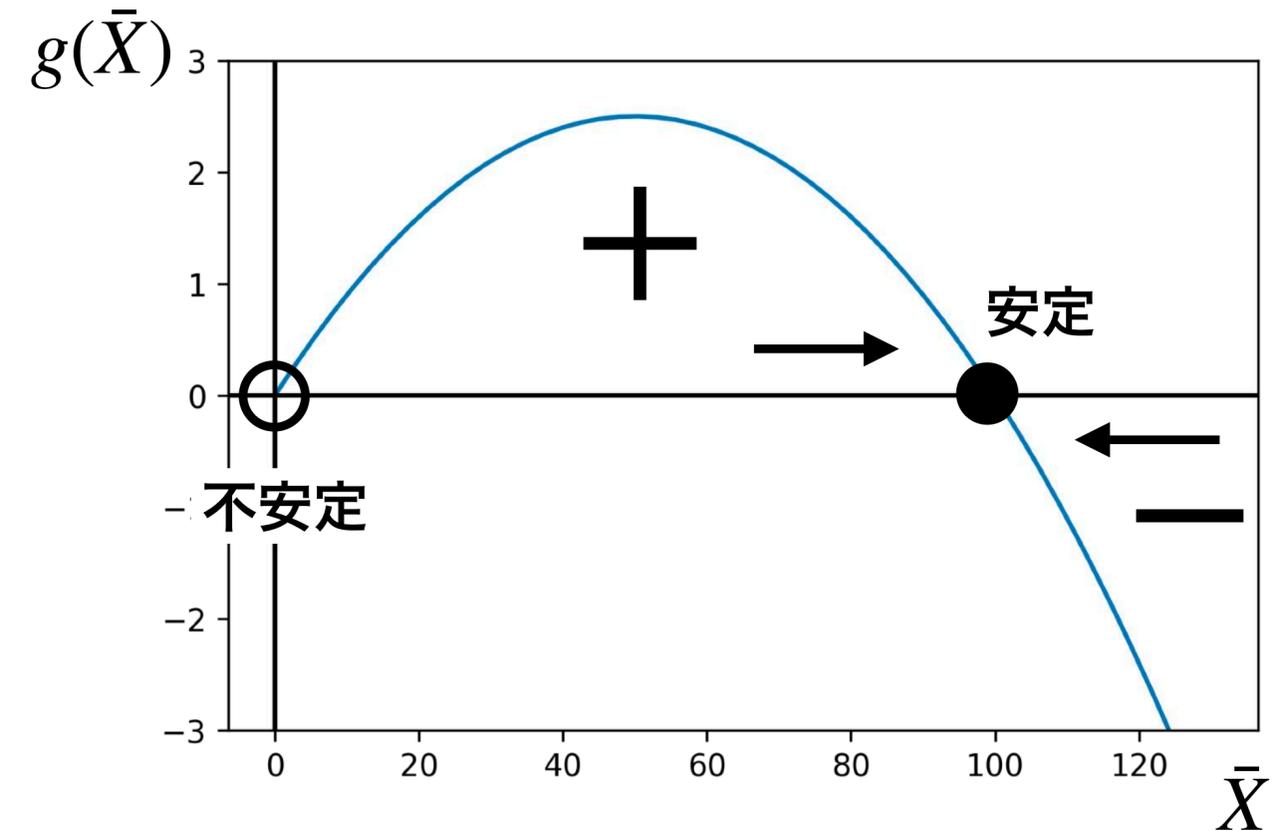
平衡点 ($X_{t+1} = X_t = \bar{X}$ となるような \bar{X})

モデルの式に代入すれば,

$$r \left(1 - \frac{\bar{X}}{K} \right) \bar{X} = 0.$$

これを満たすが \bar{X} が平衡点になる.

よって, 平衡点は $\bar{X} = 0, K$.



離散指数増殖モデルの数値計算と プロット

離散指数増殖モデル (1)

```
# 01-01. 離散指数増殖モデル (1)
import matplotlib.pyplot as plt

a = 0.1
x = 1
t = 0

t_list = [t]
x_list = [x]
for i in range(100):
    t = t + 1
    x = x + a * x

    t_list.append(t)
    x_list.append(x)

plt.plot(t_list, x_list)
```

同じ高さのインデントにより
forループのブロックを表現

パラメータ・初期値の設定

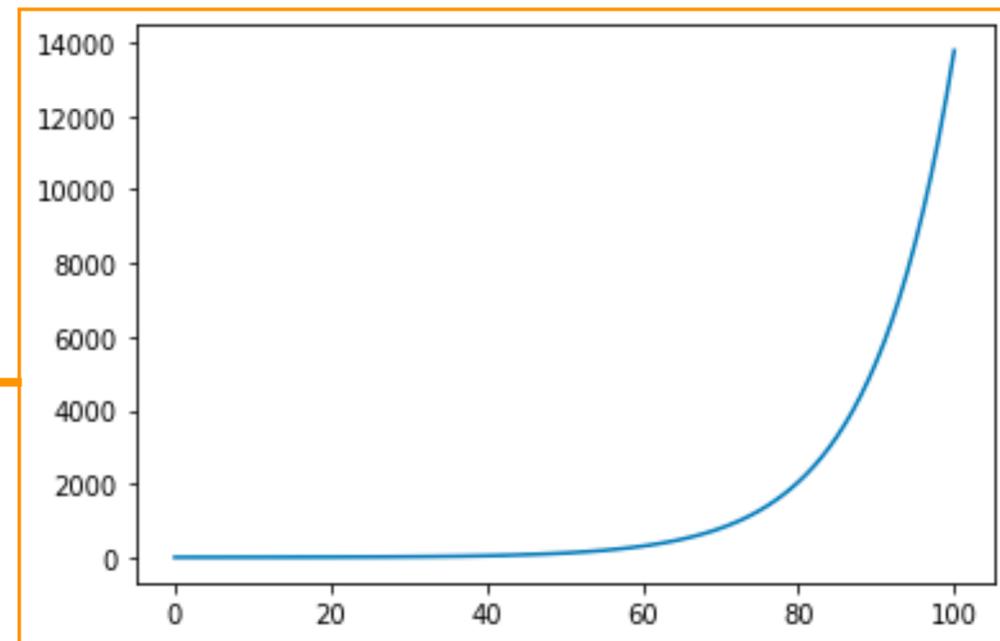
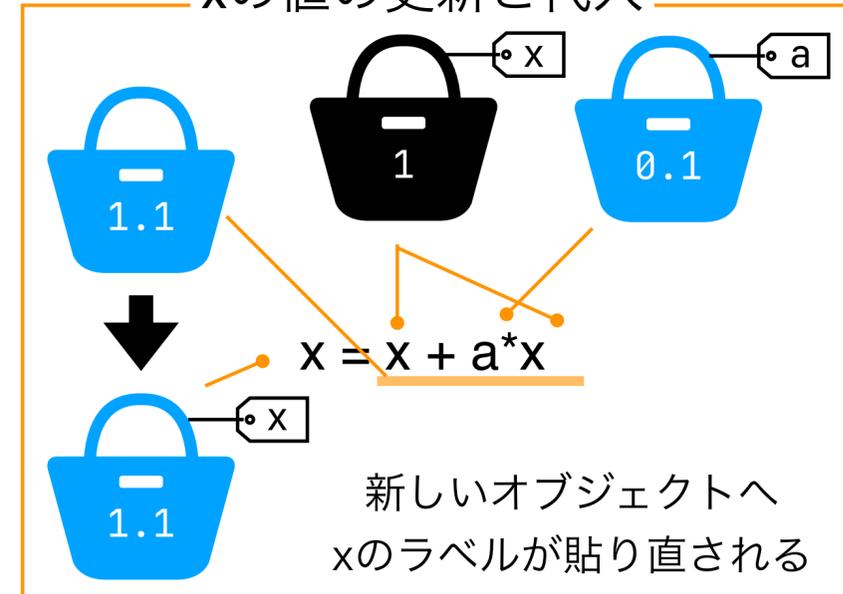
時刻 (t) と個体数 (x) を記録するリスト

100世代のループ

$$X_{t+1} = X_t + aX_t$$

更新された値を
リストに格納

xの値の更新と代入

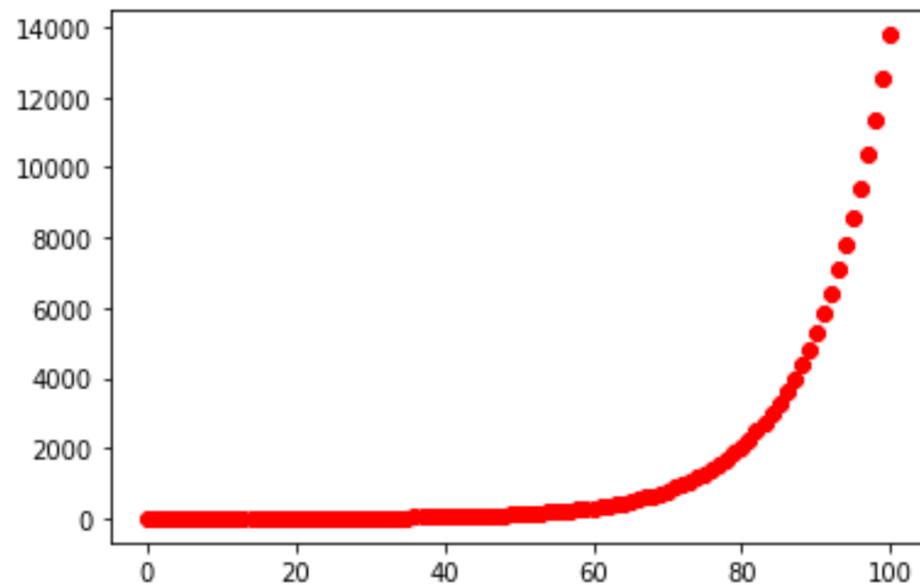


プロットを鍛える (1)

matplotlib.pyplot

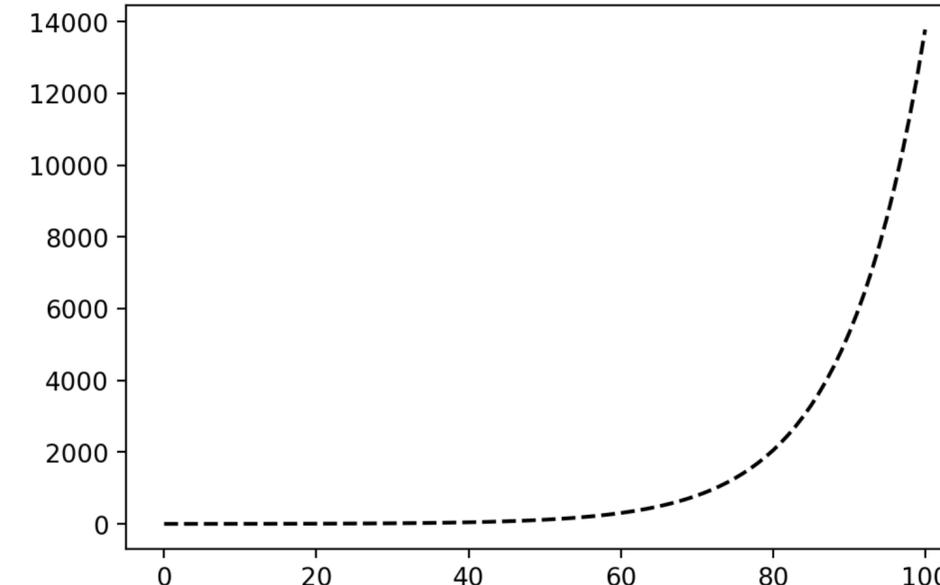
- plot(横軸値リスト, 縦軸値リスト, フォーマット)
(横軸値, 縦軸値)で与えられる座標値を指定されたフォーマットでプロットする

```
# 01-02. フォーマットの変更1  
plt.plot(t_list, x_list, "ro")
```



フォーマットの例
r: 赤, o: circleマーカー

```
# 01-03. フォーマットの変更2  
plt.plot(t_list, x_list, "k--")
```



フォーマットの例
k: 黒, -: ダッシュライン

指定できるフォーマットの詳細は公式ドキュメントを参照.

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html

プロットを鍛える (2)

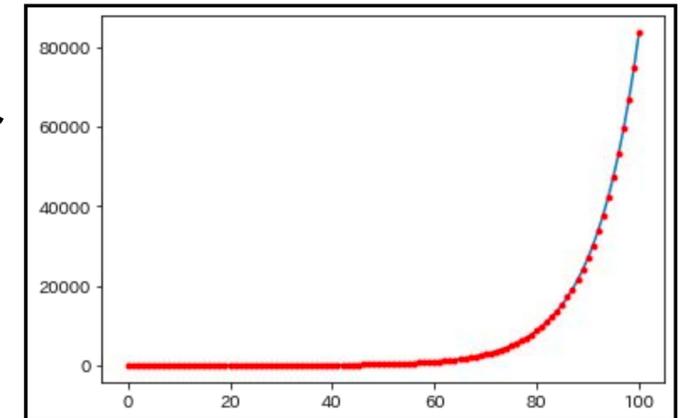
matplotlib.pyplot

- plot(横軸値リスト1, 縦軸値リスト1, フォーマット1, 横軸値リスト2, 縦軸値リスト2, フォーマット2, ...)
(横軸値, 縦軸値)で与えられる座標値を指定されたフォーマットでプロットする。
フォーマットを指定しなくても良い。

01-04. 複数のデータのプロット1

```
plt.plot(t_list, x_list, "-", t_list, x_list, "r.")
```

01-04. 同じ結果を異なる
フォーマットで重ねてプロット



01-05. 複数のデータのプロット2

```
x_list_list = []
```

```
for a in [0.1, 0.11, 0.12]:
```

```
    x = 1
```

```
    t = 0
```

```
    t_list = [t]
```

```
    x_list = [x]
```

```
    for i in range(100):
```

```
        t = t + 1
```

```
        x = x + a * x
```

```
        t_list.append(t)
```

```
        x_list.append(x)
```

```
    x_list_list.append(x_list)
```

```
plt.plot(t_list, x_list_list[0], t_list, x_list_list[1], t_list, x_list_list[2])
```

個体数 (x) を記録するリスト
(x_list) を記録するリスト

3通りのaの値について
ループを回す

それぞれのaの値について
100世代分のループを回す

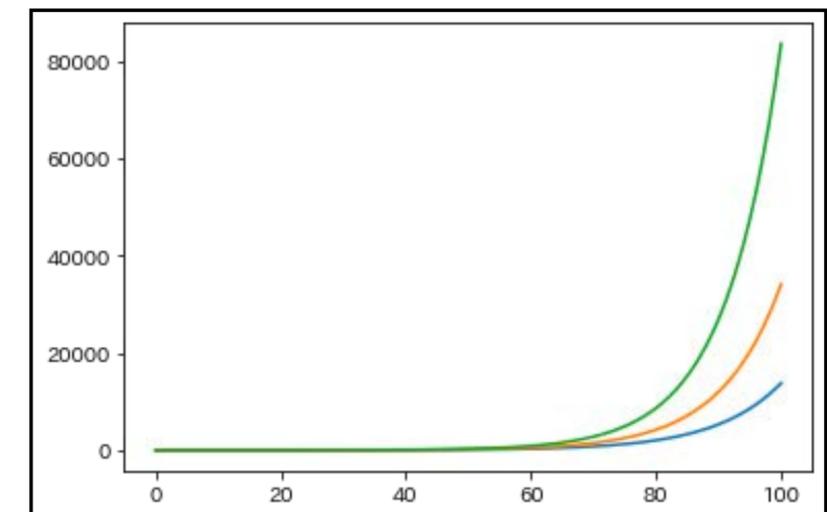
forのネスト (2重ループ)

a=0.1の結果

a=0.11の結果

a=0.12の結果

01-05. 異なるaに対する
結果をまとめてプロット



プロットを鍛える (3)

matplotlib.pyplot

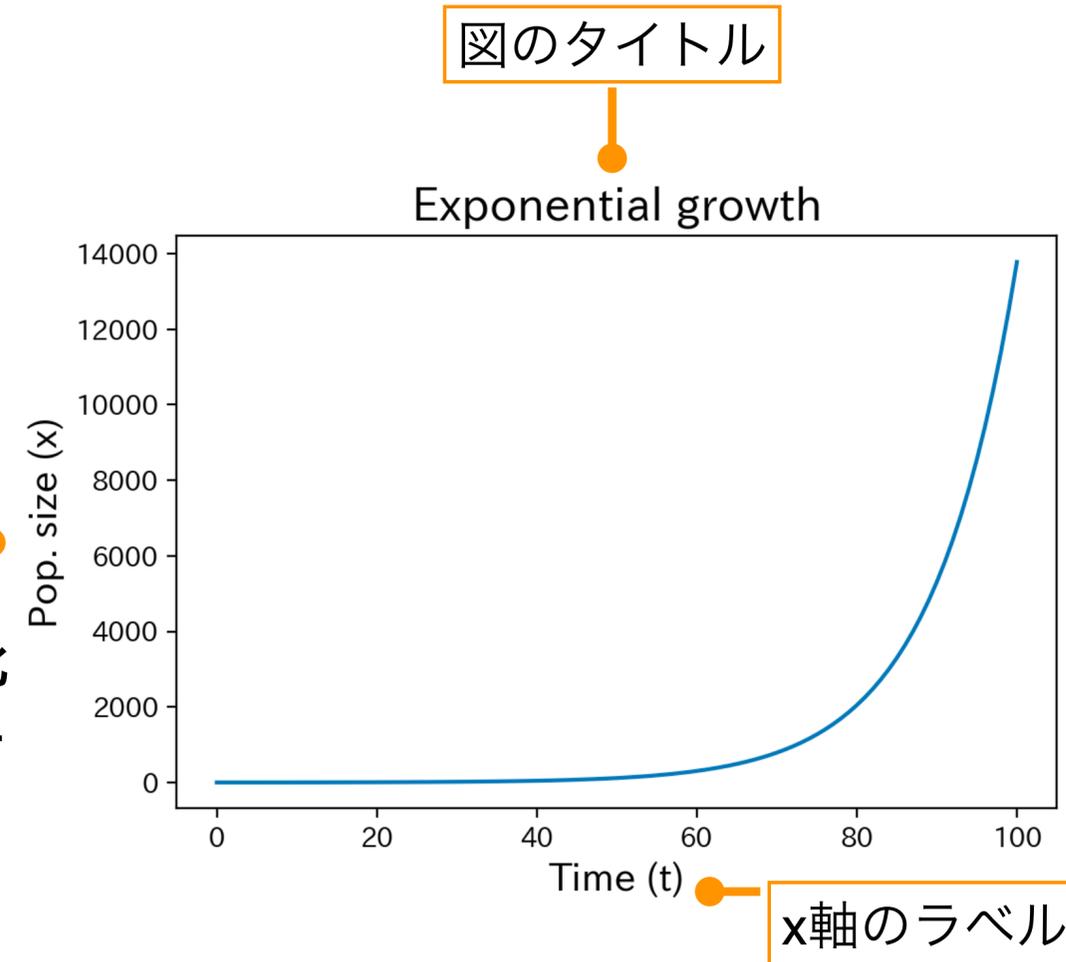
- title(タイトル) : プロットにタイトルをつける.
- xlabel(x軸ラベル), ylabel(y軸ラベル) : 各軸にラベルをつける.

```
# 01-06. タイトル・軸ラベル1
plt.plot(t_list, x_list)
plt.title("Exponential growth")
plt.xlabel("Time (t)")
plt.ylabel("Pop. size (x)")
```

y軸のラベル

注意: デフォルトだと, 日本語フォントをタイトルやラベルに使うと豆腐化する (日本語フォントが□になってしまう). 基本はアルファベットがおすすめ. どうしても日本語化したい人は後述の日本語化を試してみよう.

```
# 01-07. タイトル・軸ラベル2
plt.plot(t_list, x_list)
plt.title("Exponential growth", fontsize="xx-large")
plt.xlabel("Time (t)", fontsize="x-large")
plt.ylabel("Pop. size (x)", fontsize="x-large")
```



フォントサイズを指定できる.

整数もしくは 'xx-small', 'x-small', 'small', 'medium', 'large', 'x-large', 'xx-large' のいずれか.

プロットを鍛える (3) : おまけ

matplotlibで日本語を使う

```
# matplotlibで日本語フォントを使う準備1  
# matplotlib-fontjaのインストール  
  
!pip install matplotlib-fontja
```

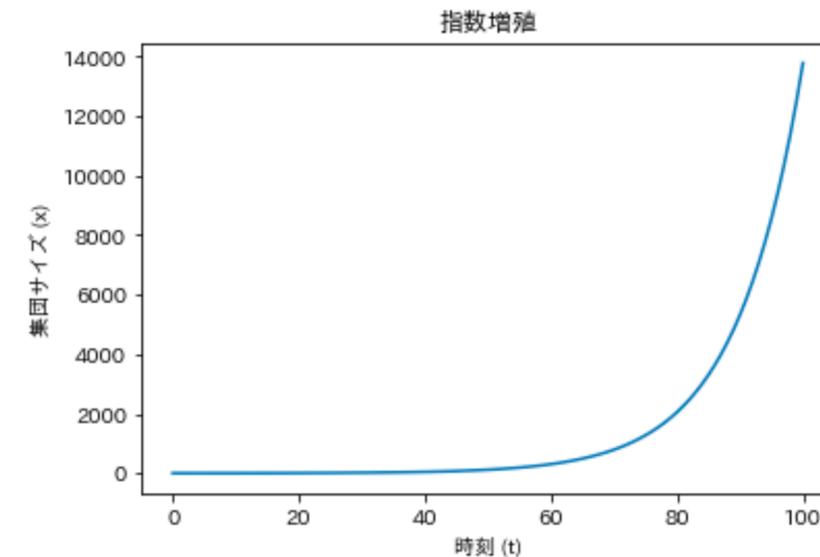
matplotlibで日本語表示できるようにするパッケージmatplotlib-fontja

- <https://github.com/ciffelia/matplotlib-fontja>
- <https://pypi.org/project/matplotlib-fontja/>をインストール. !はOS上でシェルコマンドを実行するためのマジックコマンド

```
# matplotlibで日本語フォントを使う準備2  
# matplotlibとともにjapanize_matplotlibを読み込む  
  
import matplotlib.pyplot as plt  
import matplotlib_fontja
```

matplotlibを読み込んだ後で, matplotlib_fontjaをインポートする. これによりmatplotlibのフォント設定を弄り, 日本語フォントを利用可能な状態にしている.

```
# タイトル・軸ラベル (日本語)  
plt.plot(t_list, x_list)  
plt.title("指数増殖")  
plt.xlabel("時刻 (t)")  
plt.ylabel("集団サイズ (x)")
```



おまけ
matplotlib-fontjaはmatplotlibを日本語表示に対応させるためのパッケージjapanize-matplotlibをフォークし, 最新版のPythonへも対応を施したパッケージ.

matplotlib-fontjaのインストールはインスタンスが再起動された場合には, 再度行う必要あり!

プロットを鍛える (4)

matplotlib.pyplot

- `figure(dpi=解像度)`: 図の解像度を指定する. デフォルトは100 (dpi).
- `figure(figsize=[幅, 高さ])`: 図のサイズ (幅と高さ) をインチで指定する. デフォルトは[6.4, 4.8]

```
###01-08. 解像度の変更  
plt.figure(dpi = 200)  
plt.plot(t_list, x_list)
```

```
# 01-09. プロットサイズの変更  
plt.figure(figsize = [5, 7])  
plt.plot(t_list, x_list)
```

他にも様々な調整ができるので, 詳しく知りたい人は

- 公式ドキュメント <https://matplotlib.org/>
- DataCampチュートリアル Matplotlib Tutorial: Python Plotting <https://www.datacamp.com/community/tutorials/matplotlib-tutorial-python>

などを参照. また, matplotlib以外やmatplotlibを拡張するプロット用のライブラリがあるので, 興味のある人は探してみよう.

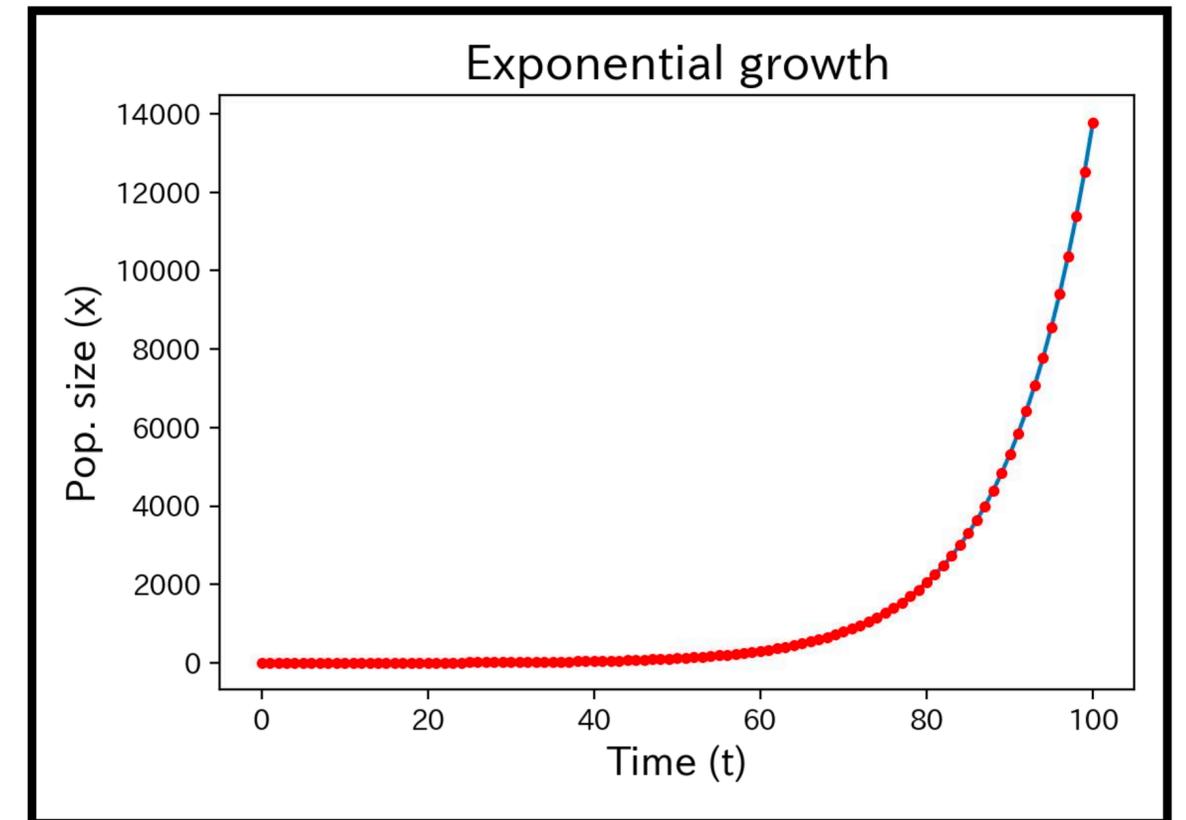
離散指数増殖モデル (2)

```
# 01-10. 離散指数増殖モデル (2)
a = 0.1
x = 1
t = 0

t_list = [t]
x_list = [x]
for i in range(100):
    t = t + 1
    x = x + a * x

    t_list.append(t)
    x_list.append(x)

plt.figure(dpi=200)
plt.plot(t_list, x_list, "-", t_list, x_list, "r.")
plt.title("Exponential growth", fontsize="xx-large")
plt.xlabel("Time (t)", fontsize="x-large")
plt.ylabel("Pop. size (x)", fontsize="x-large")
```



離散ロジスティック成長モデルの 数値計算とプロット

課題 ノーマル

プログラムの流れ 離散ロジスティックモデルの時間発展

必要なパッケージ (matplotlib) の読み込み



初期値 (x0) ・パラメータ (r, K) の定義



時間tと個体数xのリスト (t_list, x_list) を作成



forループ

100世代のループ

差分方程式

$$X_{t+1} = X_t + r \left(1 - \frac{X_t}{K} \right) X_t$$

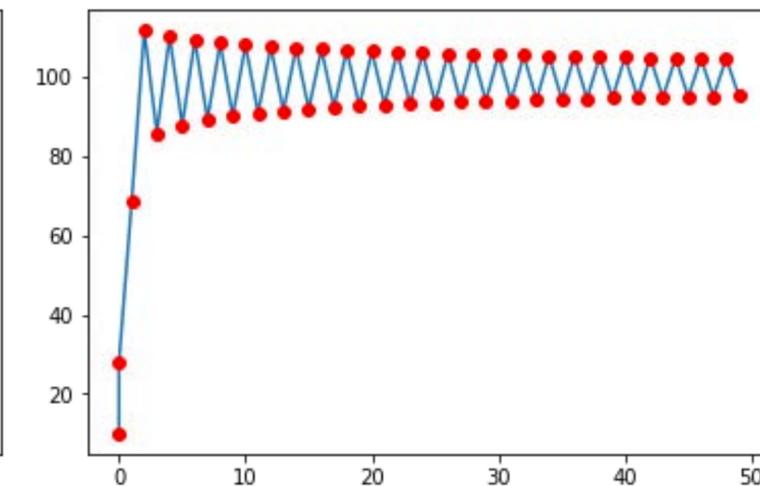
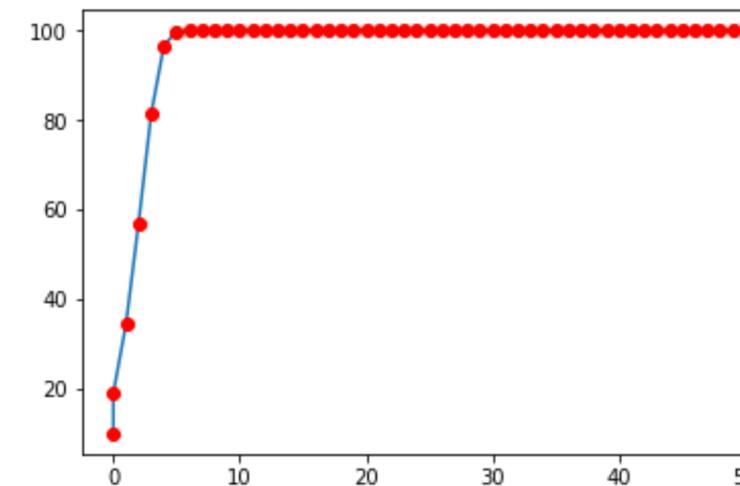
を使い, t+1ステップ目を計算する.

$$x = x + r * (1 - x / K) * x$$

結果をリストに追加



プロット



局所安定性解析の結果から推察される
様々な挙動が再現できるとGood!

離散指数増殖モデルを参考にしてみよう

```
# 01-10. 離散指数増殖モデル (2)
import matplotlib.pyplot as plt

a = 0.1
x = 1
t = 0

t_list = [t]
x_list = [x]
for i in range(100):
    t = t + 1
    x = x + a * x

    t_list.append(t)
    x_list.append(x)

plt.figure(dpi=200)
plt.plot(t_list, x_list, "-", t_list, x_list, "r.")
plt.title("Exponential growth", fontsize="xx-large")
plt.xlabel("Time (t)", fontsize="x-large")
plt.ylabel("Pop. size (x)", fontsize="x-large")
```

必要なパッケージ (matplotlib) の読み込み



初期値 (x0) ・ パラメータ (a) の定義



時間tと個体数xのリスト (t_list, x_list) を作成



forループ

100世代のループ

差分方程式

$$X_{t+1} = X_t + aX_t$$

を使い, t+1ステップ目を計算する.

$$x = x + r * (1 - x / K) * x$$

結果をリストに追加



プロット

離散指数増殖モデルを参考にしてみよう

```
# 01-10. 離散指数増殖モデル (2)
import matplotlib.pyplot as plt

a = 0.1
x = 1
t = 0

t_list = [t]
x_list = [x]
for i in range(100):
    t = t + 1
    x = x + a * x

    t_list.append(t)
    x_list.append(x)

plt.figure(dpi=200)
plt.plot(t_list, x_list, "-", t_list, x_list, "r.")
plt.title("Exponential growth", fontsize="xx-large")
plt.xlabel("Time (t)", fontsize="x-large")
plt.ylabel("Pop. size (x)", fontsize="x-large")
```

必要なパッケージ (matplotlib) の読み込み



初期値 (x0) ・ パラメータ (a) の定義



時間tと個体数xのリスト (t_list, x_list) を作成



forループ

100世代のループ

差分方程式
$$X_{t+1} = X_t + aX_t$$

を使い, t+1ステップ目を計算する.
$$x = x + r * (1 - x / K) * x$$

この辺を変更すれば良さそう!
+プロット時のタイトルも

結果をリストに追加



プロット

第3回 課題 ノーマル

1. 離散ロジスティックモデルの平衡点を求めよ。また、その安定性を調べよ。
2. 離散ロジスティックモデルの時間発展を様々な r に対してプロットせよ ($0.5 \leq r \leq 3$ ぐらいの範囲がおすすめ)。
3. 質問, 意見, 要望等をどうぞ。

課題をGitHub Classroomにて提出すること

第3回 課題 ハード

1. 離散ロジスティックモデルの分岐図を描け.

挑戦したい人は、課題をGitHub Classroomにて提出すること

次回予告

第4回：指数成長・ロジスティック成長

5月8日

復習推奨

- 指数成長
- ロジスティック成長
- 微分方程式（変数分離型）の解法

発展

分岐図 bifurcation diagram

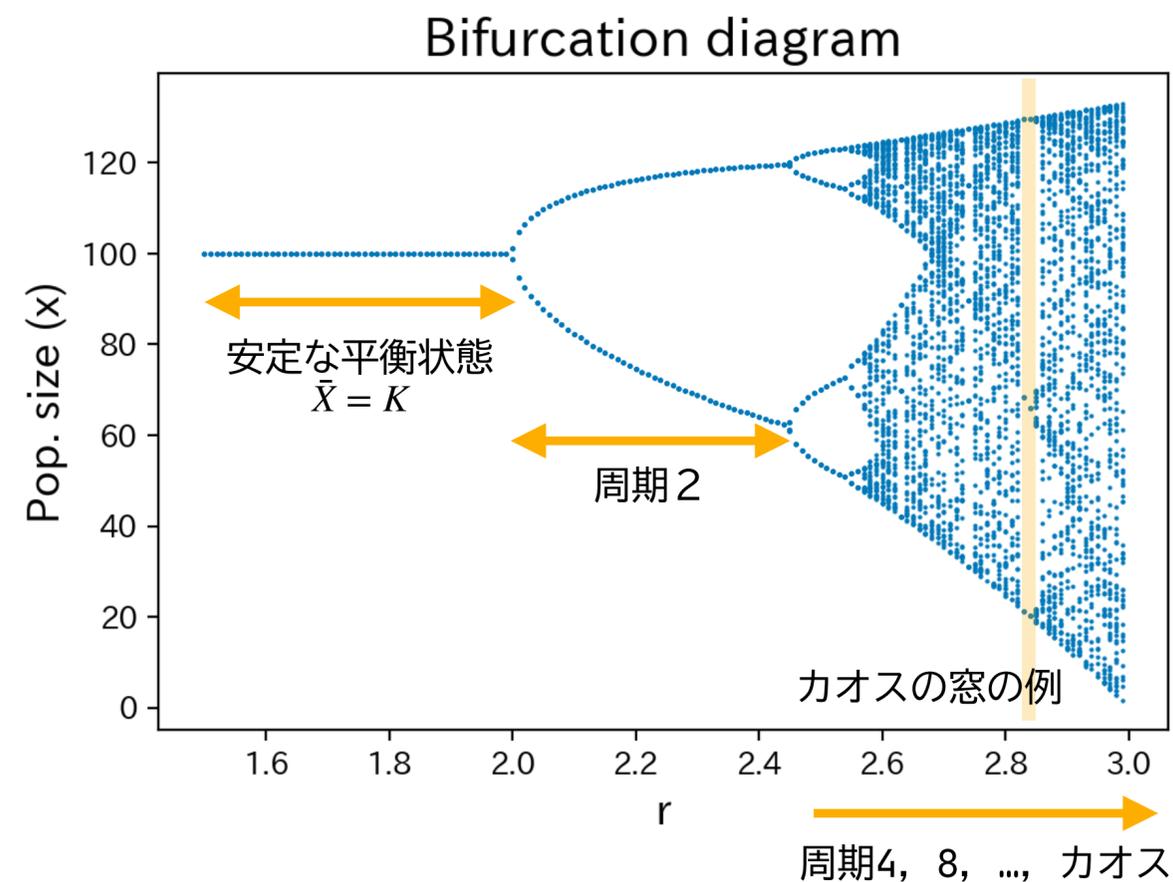
ロジスティック成長モデル

$$X_{t+1} = X_t + r \left(1 - \frac{X_t}{K} \right) X_t$$

r : 内的自然増加率. 個体数が十分小さい場合 ($x \approx 0$) の1世代あたりの増殖率. $r \geq 0$.

K : 環境収容力. ある環境で維持されうる個体数, $K > 0$.

- 内的自然増加率 (r) が大きくなると平衡状態が不安定になる.
- $r > 2$ で周期2の安定な振動を観察できる. さらに r が大きくなると, 周期4, 8, ... と分岐する.
- その後, カオス軌道が観察される. 時折, カオス軌道から特定の周期に変わるカオスの窓と呼ばれる空白地帯が出現する.



プログラムの流れ
分岐図

課題 ハード 1

必要なパッケージ (matplotlib) の読み込み



パラメータ (K) の定義



パラメータrと個体数xのリスト (r_list, x_list) を作成



forループ

初期値 (x0) ・ パラメータ (r) の設定 $1.5 \leq r \leq 3$ の範囲

forループ

t=0からt<1000まで

差分方程式

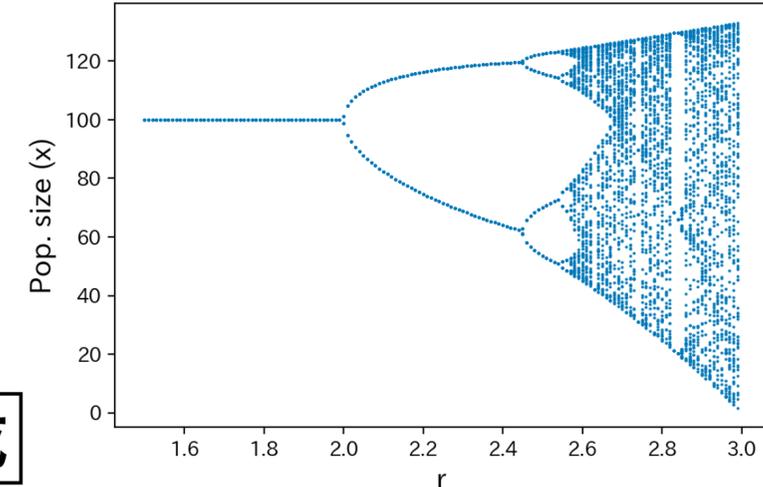
$$X_{t+1} = X_t + r \left(1 - \frac{X_t}{K} \right) X_t$$

を使い, t+1ステップ目を計算する.

$$x = x + r * (1 - x/K) * x$$

最後の100ステップをリスト (r_list, x_list) に追加

Bifurcation diagram



最終的にこんなのを
プロットしたい

→ プロット