Module 8

Week 4 Monday

Simulating population growth

Given an initial population, a rate of growth, and a maximum capacity how to model population growth?

We need a growth formula:

```
growth = rate*(1-population/capacity)*population
```

We can run the simulation 500 times and observe how the population changes over time.

Simulating population growth

```
def number(initial, rate, capacity):
    num = initial
    for i in range(500):
        growth = rate*(1-num/capacity)*num
        num += growth
        print(i, num)
```

We can call this function with different values: `number(10, .2, 100)`
Copy and paste the population numbers into a spreadsheet and plot a graph of the growth. (`i` on the x-axis and `num` on the y-axis).

Lotka-Volterra Models

- In the great war, there was no fishing in the Adriatic Sea
- When fisherman could fish again, they expected abundant catches
- However, this turned out to be false

- Search for an explanation led to the Lotka Volterra model for two species

Lotka Volterra

- Predator – Prey Model:
  - x – number of prey
  - y – number of predators

- Population develops:
  - For prey:
    - few predators: gain population (no threats)
    - many predators: loss of population (get eaten)
  - For predator:
    - few prey: loss of population (starvation)
    - much prey: gain of population (feeding frenzy)

Lotka Volterra

$$x_{n+1} = ax_n - bx_n y_n$$

$$y_{n+1} = cy_n + dx_n y_n$$

- *x* – prey at time *n*
- *y* – predators at time *n*
- *a* natural growth rate of prey
- *b* predation rate
- *c* efficiency of turning prey into predators
- *d* natural death rate of predator

Implementing Lotka Volterra

- Define a function with parameters, including initial values x0 and y0
- Use a for loop in order to model 500 time periods

```
def lotka_volterra(x0, y0, a, b, c, d):
    x = x0
    y = y0
    for i in range(500):
       …
         print(i,x,y)
```

Implementing Lotka Volterra

- Updating x and y according to the formula:
  - Need to use old value of x until both new values are set

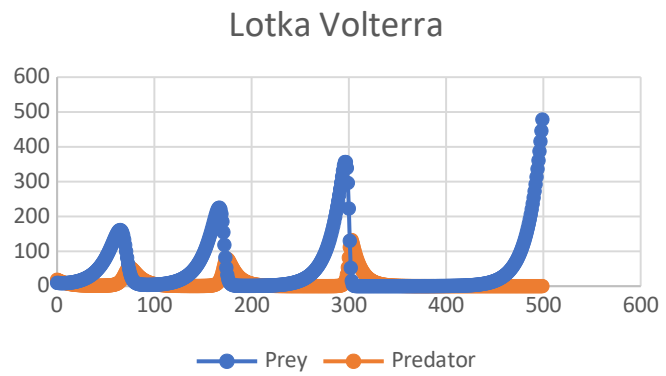
```
def lotka_volterra(x0, y0, a, b, c, d):
    x = x0
    y = y0
    for i in range(500):
        xnew = a*x-b*x*y
        ynew = c*y+d*x*y
        x = xnew
        y = ynew
        print(i, x, y)
```

Running Lotka Volterra

- Parameters are difficult to pick.
  - These ones work well

```
lotka_volterra(x0=10, y0=20, a=1.073, b=0.006, c=0.9, d=0.0021)
```

  - Obtain numbers and paste into a spreadsheet
  - Then make a table and see how the solution develops



Running Lotka Volterra

- Periodic, but not stable
  - Population sizes are increasing
- Problem with Modeling
  - This is a cute problem
  - Suffers from the problem of the atto-tiger
    - quintillionth of a tiger
    - The predator population can become very small
- But still <u>first</u> successful population model that explained reality