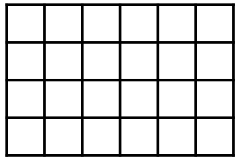
Note) Slides and example codes are available:
https://github.com/mint-lab/prog_meets_math

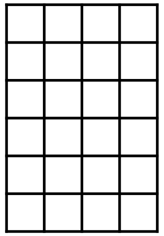
Python Meets Mathematics: Calculus

Sunglok Choi, Assistant Professor, Ph.D.
Computer Science and Engineering Department, SEOULTECH
sunglok@seoultech.ac.kr | https://mint-lab.github.io/

Building Mathematical Intuition

1. **Imagine** (or visualize)
   - e.g. *Numberblocks* (by BBC) [YouTube]
   - e.g. Why 4 x 6 == 6 x 4? (for kids without knowledge of a multiplication table and commutativity)





2. **Ask why** (or think about its applications)
   - e.g. Why did I learn about a matrix?

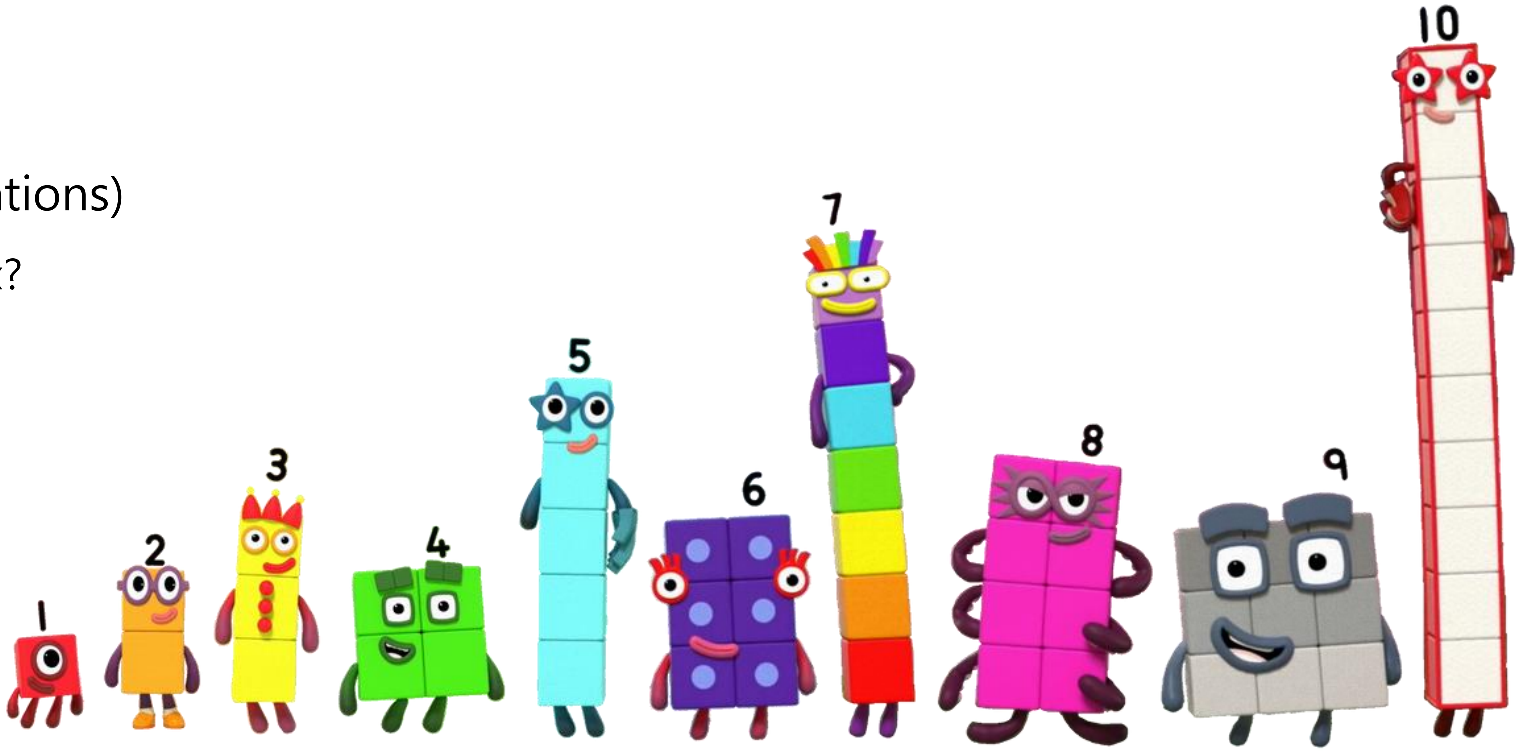


Image: Pinterest

Tools

- SciPy
  - A python-based open-source ecosystem for mathematics, science, and engineering.
    - An open-source computing tool against the commercial software such as MATLAB
    - Included in Anaconda by default
  - Major components

NumPy Base N-dimensional array package	SciPy library Fundamental library for scientific computing	Matplotlib Comprehensive 2-D plotting
IP[y]: IPython IPython Enhanced interactive console	SymPy Symbolic mathematics	pandas Data structures & analysis

- Online references: MINT Lab's Know-where
  - Please refer *Programming (Python)* category, especially about *P3. NumPy, SciPy, and Matplotlib* section.

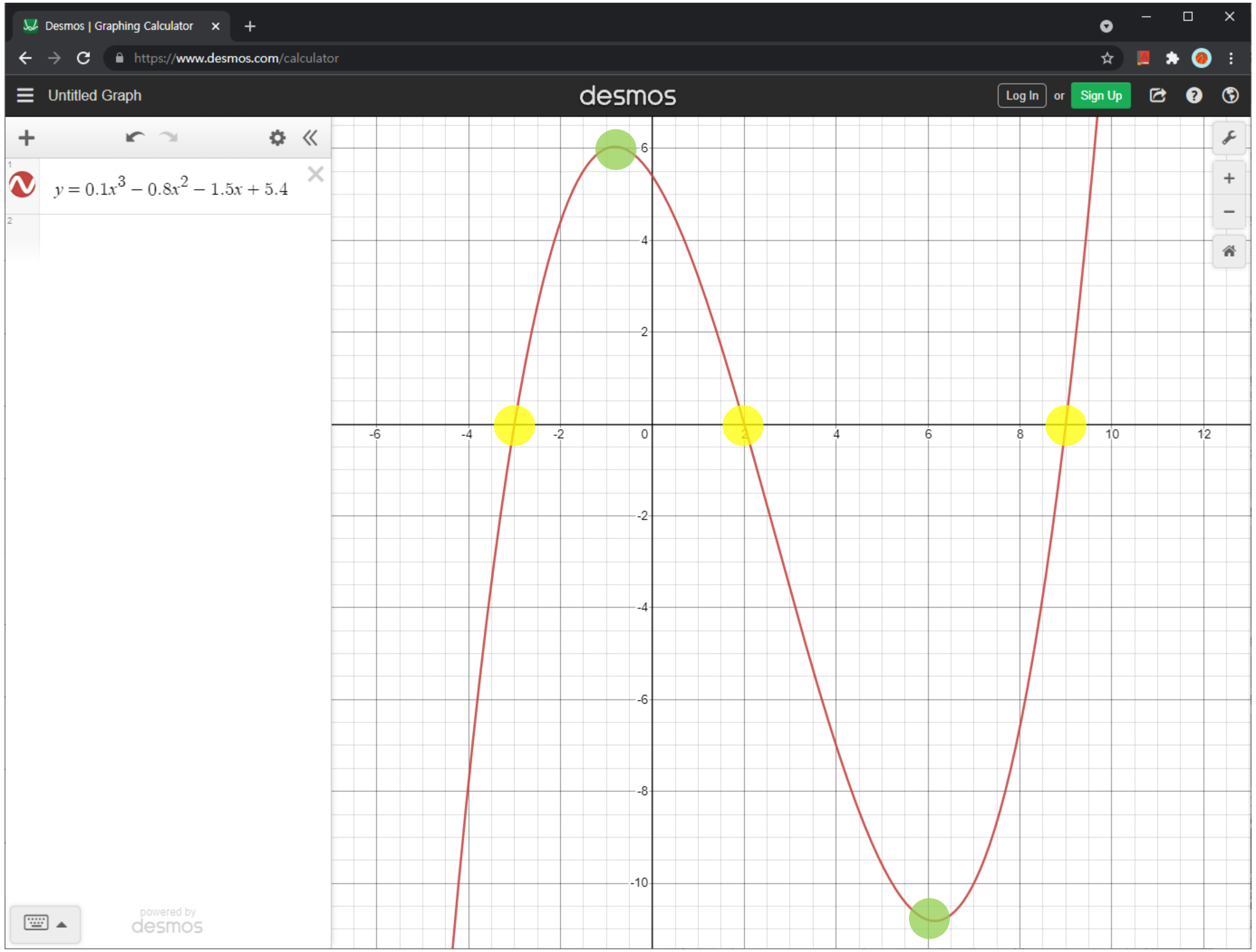
Image: SciPy

~~Programming~~ Python with SciPy meets Mathematics

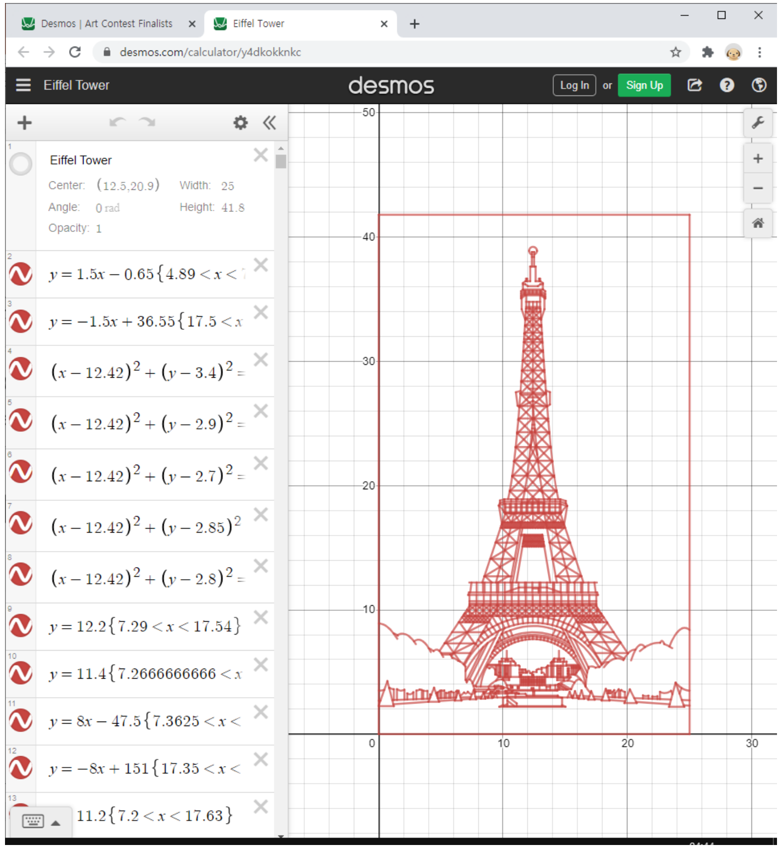
- **Calculus**
- Linear Algebra
- Optimization
- Probability
- Information Theory

Getting Started from Drawing an Equation

- Example) Drawing $y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$
- Visualization with Desmos Graphing Calculator



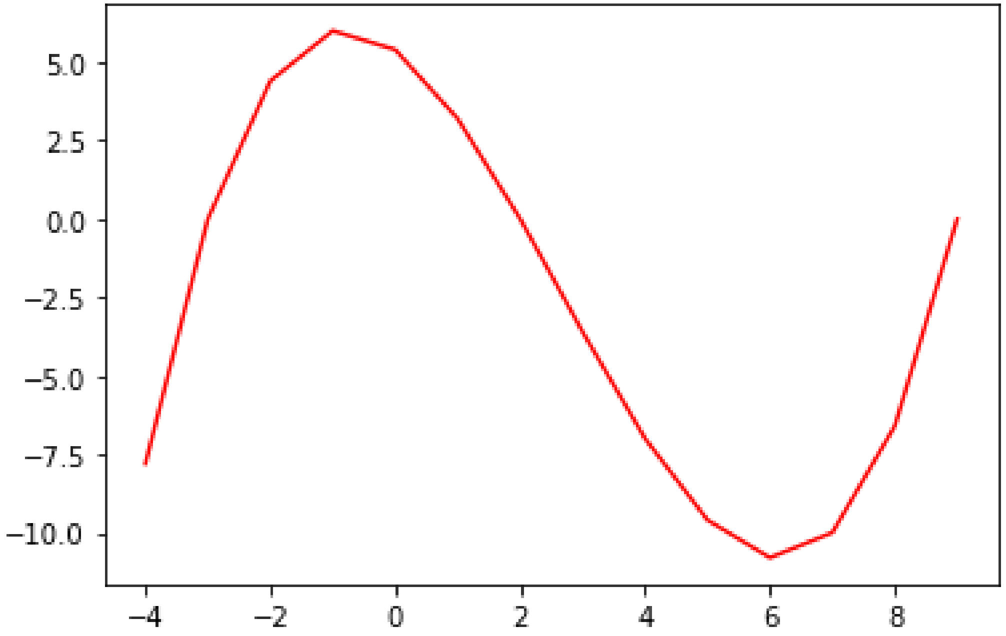
Desmos Global Math Art Contest (Age 13-14)



Getting Started from Drawing an Equation

- Example) Drawing $y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$
- Visualization with [Matplotlib]

```
import matplotlib.pyplot as plt

xs = [x for x in range(-4, 10)]
ys = [0.1*x**3 - 0.8*x**2 - 1.5*x + 5.4 for x in xs]

plt.plot(xs, ys, 'r-')
plt.show()
```



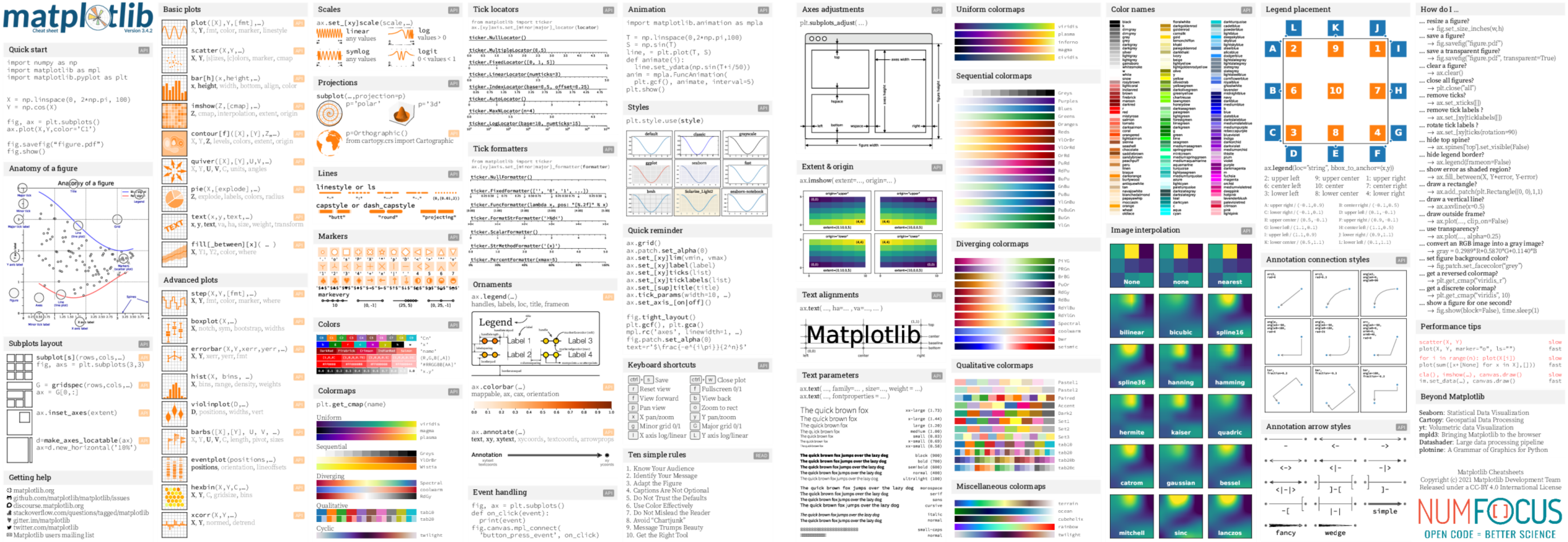
Further improvement)

- How to make the graph smooth?
- How to draw the grid?
- How to display labels on X and Y axes?
- How to make its aspect ratio equal?

Image: Matplotlib cheatsheets

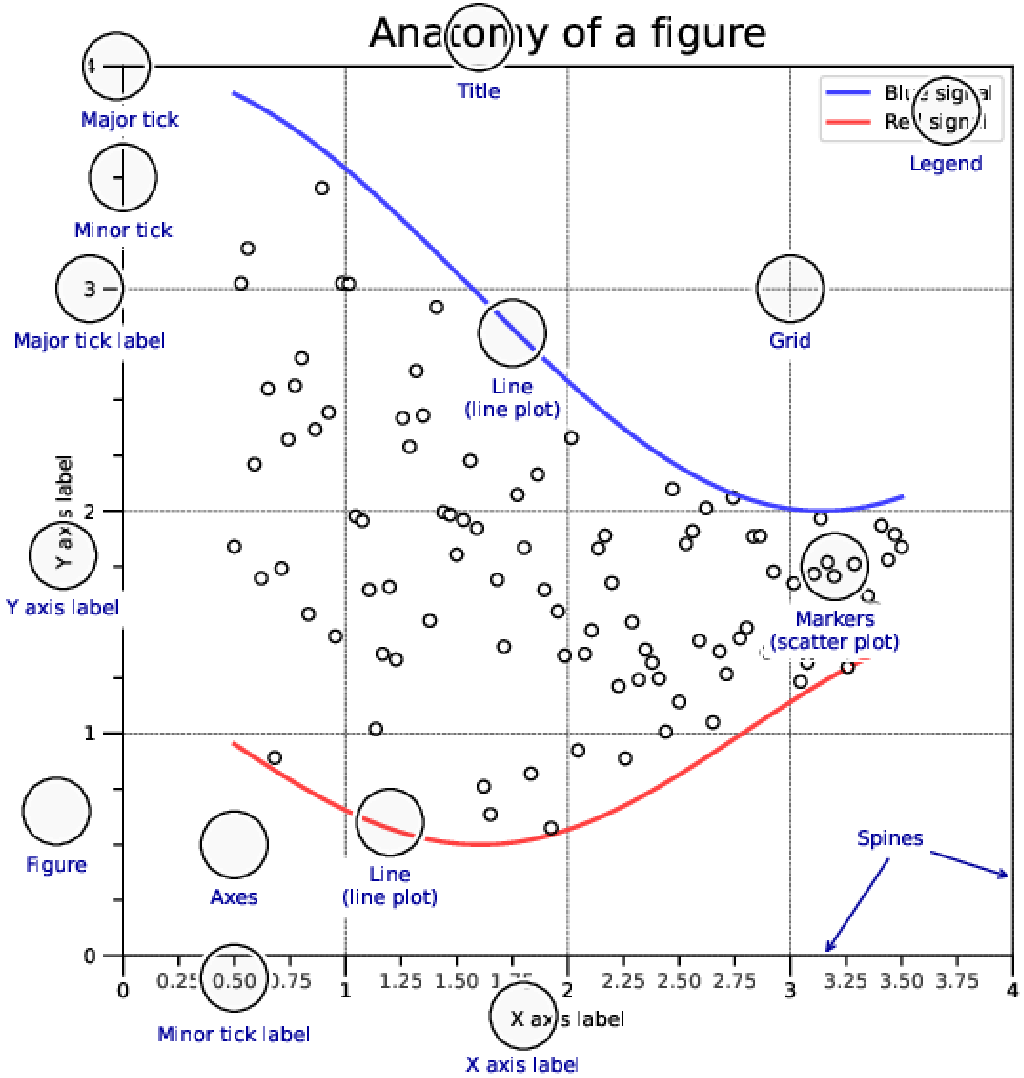
Matplotlib: A Plotting Library in Python

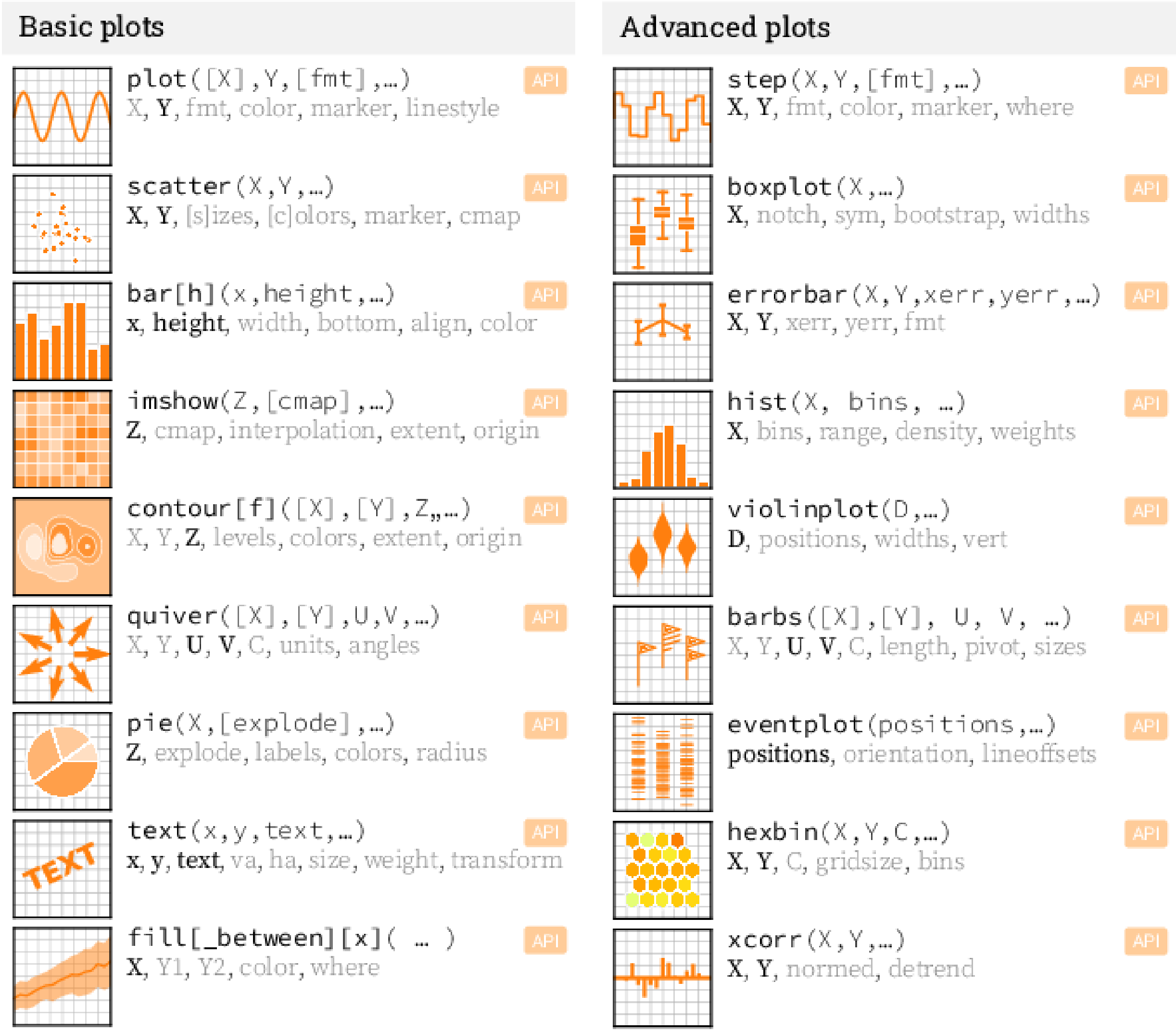
- Matplotlib is a plotting library for creating static, animated, and interactive visualization in Python.
- References: Documentation, Examples (Gallery), Tutorials, and **Cheatsheets** (made by Matplotlib)



[Matplotlib]: A Plotting Library in Python

- [Matplotlib] is a plotting library for creating static, animated, and interactive visualization in Python.
- References: [Documentation], [Examples] (Gallery), [Tutorials], and [**Cheatsheets**] (made by Matplotlib)





Matplotlib: A Plotting Library in Python

- Matplotlib is a plotting library for creating static, animated, and interactive visualization in Python.
- References: Documentation, Examples (Gallery), Tutorials, and **Cheatsheets** (made by Matplotlib)
- API examples @ **`matplotlib.pyplot` module** (a state-based interface; ~ MATLAB)
  - `plot([x], y, [fmt], …)`: Plot `y` versus `x` as lines and/or markers
  - `hist(x, bins=None, range=None, …)`: Plot a histogram
  - `contour([X, Y,] Z, [levels], …)`: Plot contour lines
  - `imshow(X, cmap=None, …)`: Display data as an image (a 2D regular raster)

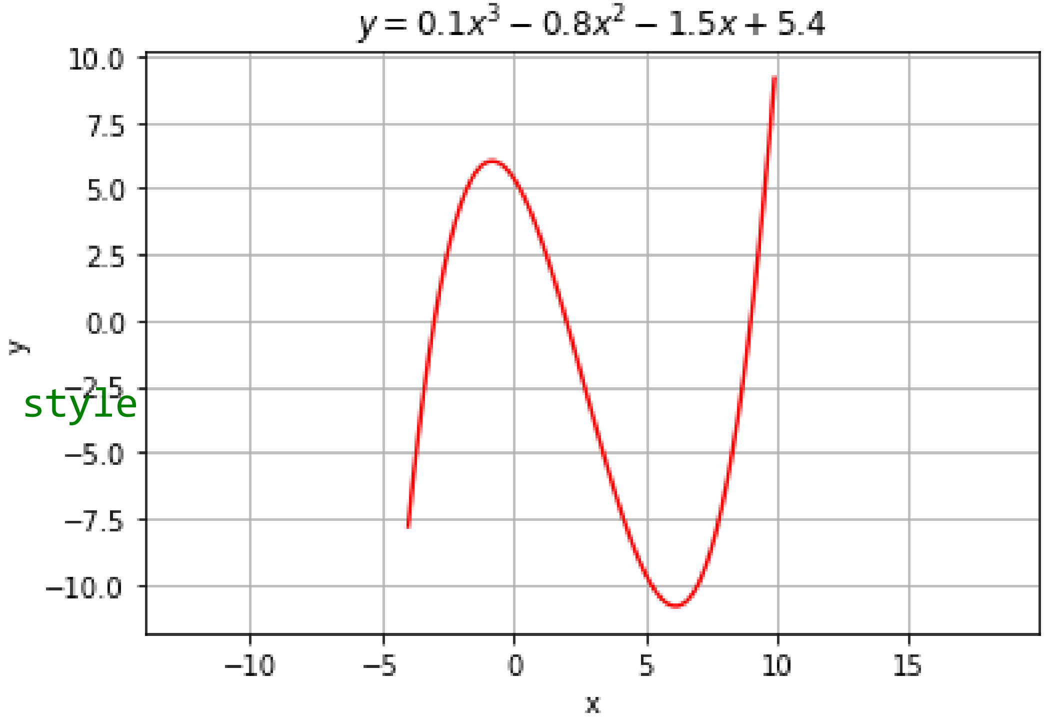
  - `title(label, …)`: Set a title for the `Axes`
  - `axis(…)`: Get or set axis properties (e.g. `'on'`/`'off'`, `'equal'`, and range of X/Y axes)
  - `legend(…)`: Place a legend on the `Axes`
  - `grid(…)`: Configure the grid lines
  - `xlabel(label, …)`, `xlim(left, right)`, …: Set the label and limit for the x-axis

  - `figure(num=None, …)`: Create a new figure or activate an existing figure
  - `show(…)`: Display all open figures
  - `savefig(filename, …)`: Save the current figure

Matplotlib: A Plotting Library in Python

- Usage example) Drawing $y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$

```
import matplotlib.pyplot as plt

scale = 10
xs = [x/scale for x in range(-4*scale, 10*scale)]
ys = [0.1*x**3 - 0.8*x**2 - 1.5*x + 5.4 for x in xs]

plt.title('$y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$') # LaTeX style
plt.plot(xs, ys, 'r-')
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
plt.axis('equal')
plt.show()
```



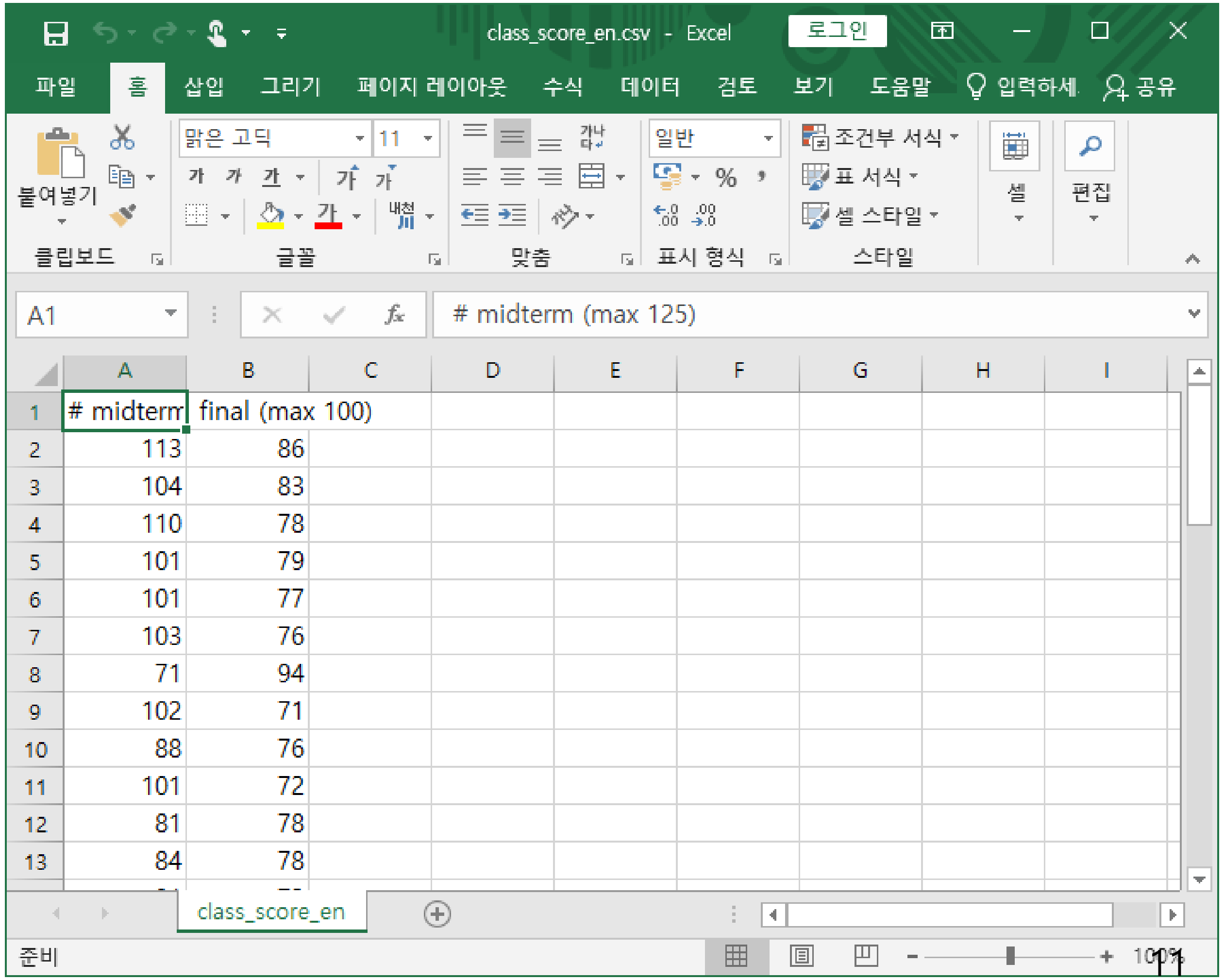
Further improvement)

- ✓ How to make the graph smooth?
- ✓ How to draw the grid?
- ✓ How to display labels on X and Y axes?
- ✓ How to make its aspect ratio equal?

Matplotlib: A Plotting Library in Python

- Example) Plotting midterm and final scores
  - The given data (file: `data/class_score_en.csv`)

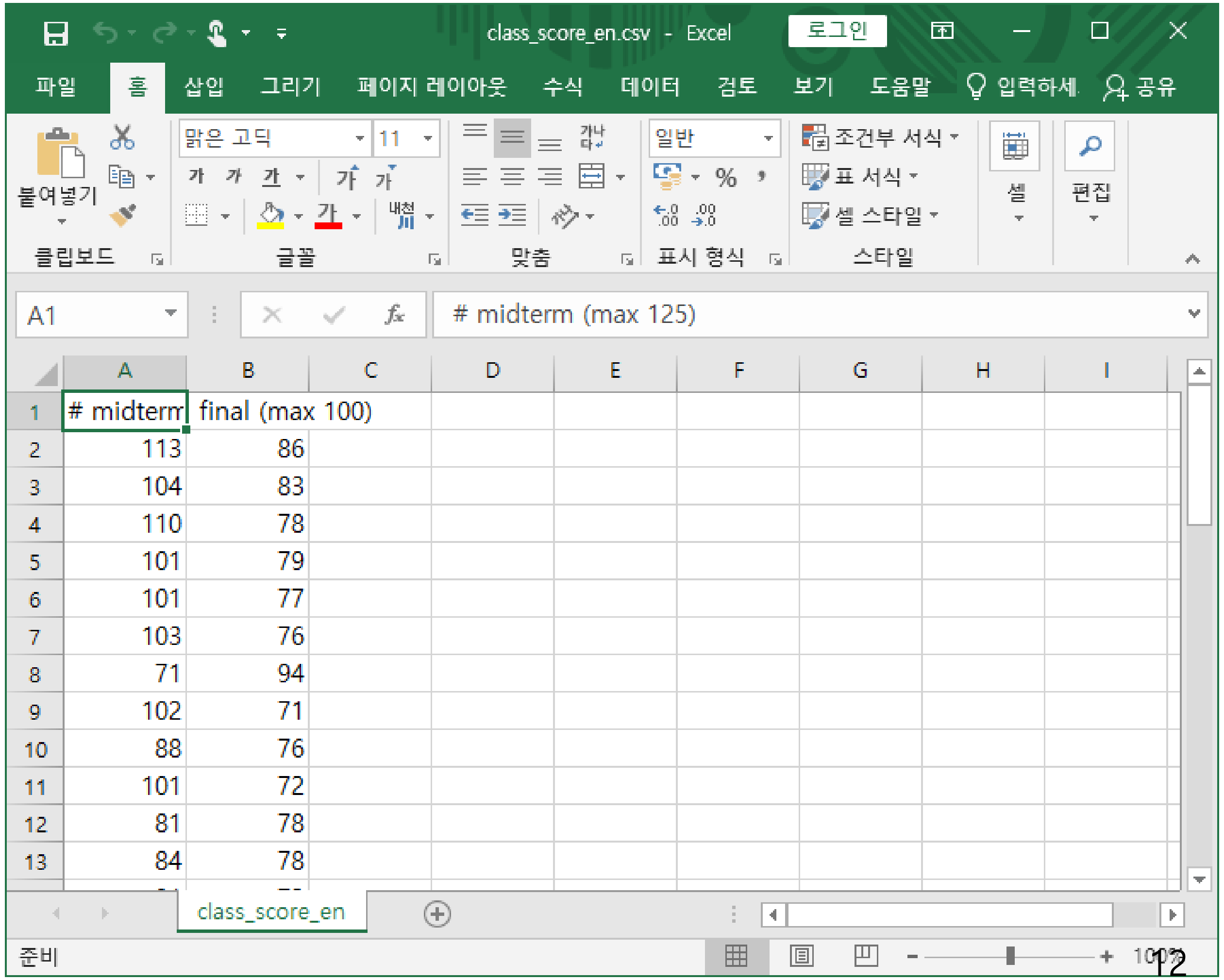
```
# midterm (max 125), final (max 100)
113, 86
104, 83
110, 78
101, 79
101, 77
103, 76
71, 94
102, 71
88, 76
101, 72
81, 78
84, 78
. . .
```



	A	B
1	# midterm	final (max 100)
2	113	86
3	104	83
4	110	78
5	101	79
6	101	77
7	103	76
8	71	94
9	102	71
10	88	76
11	101	72
12	81	78
13	84	78

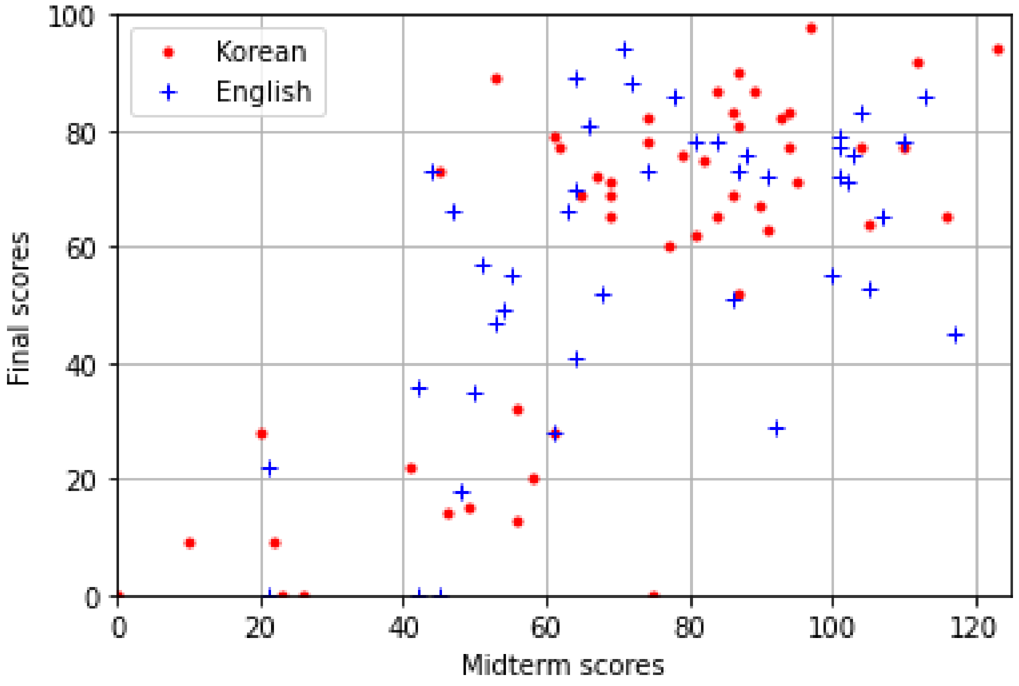
[Matplotlib](): A Plotting Library in Python

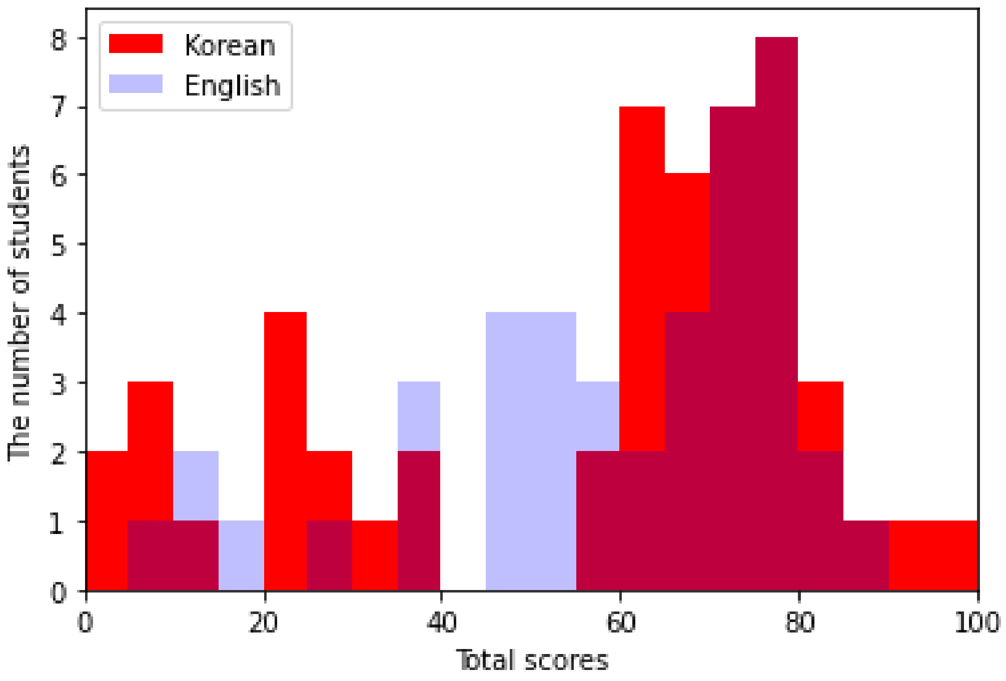
- Example) Plotting midterm and final scores
  - Analysis using representative values (e.g. mean, median, …; [대표값]() in Korean)
    - Midterm
      - Mean: **74.209**
      - Variance: 632.817
      - Median: **72.000**
      - Min/Max: (21.000, 117.000)
    - Final
      - Mean: **58.674**
      - Variance: 618.545
      - Median: **66.000**
      - Min/Max: (0.000, 94.000)
    - Total
      - Mean: **58.952**
      - Variance: 423.546
      - Median: **65.000**
      - Min/Max: (6.720, 87.760)



[Matplotlib](): A Plotting Library in Python

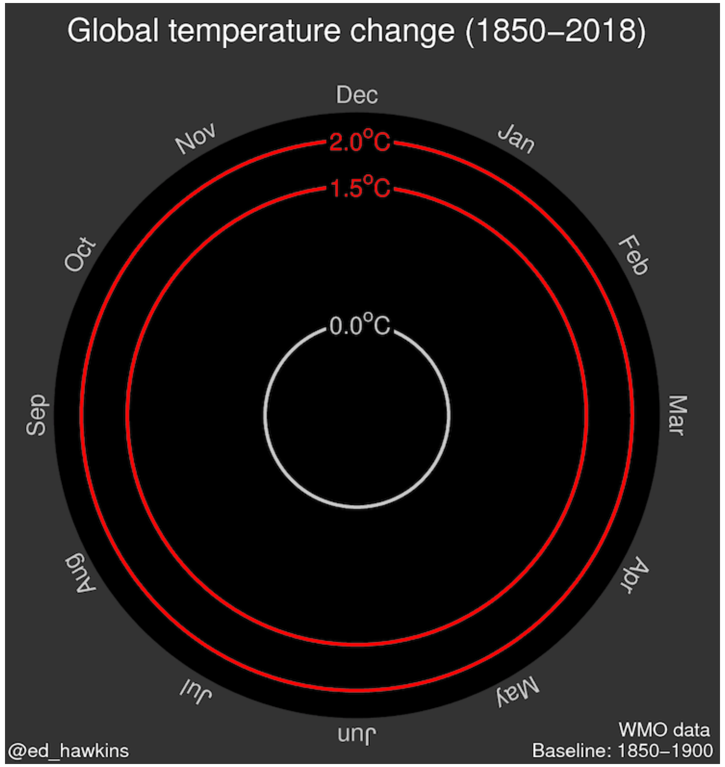
- Example) Plotting midterm and final scores
  - Practice) Plot the [scatter plot]() of midterm and final scores
  - Practice) Plot the [histogram]() of total scores
  - What can you discover from two graphs?





Visualization

- **Visualization is very important** not only for your **data** but also for your **programs**, **systems**, and **society.**



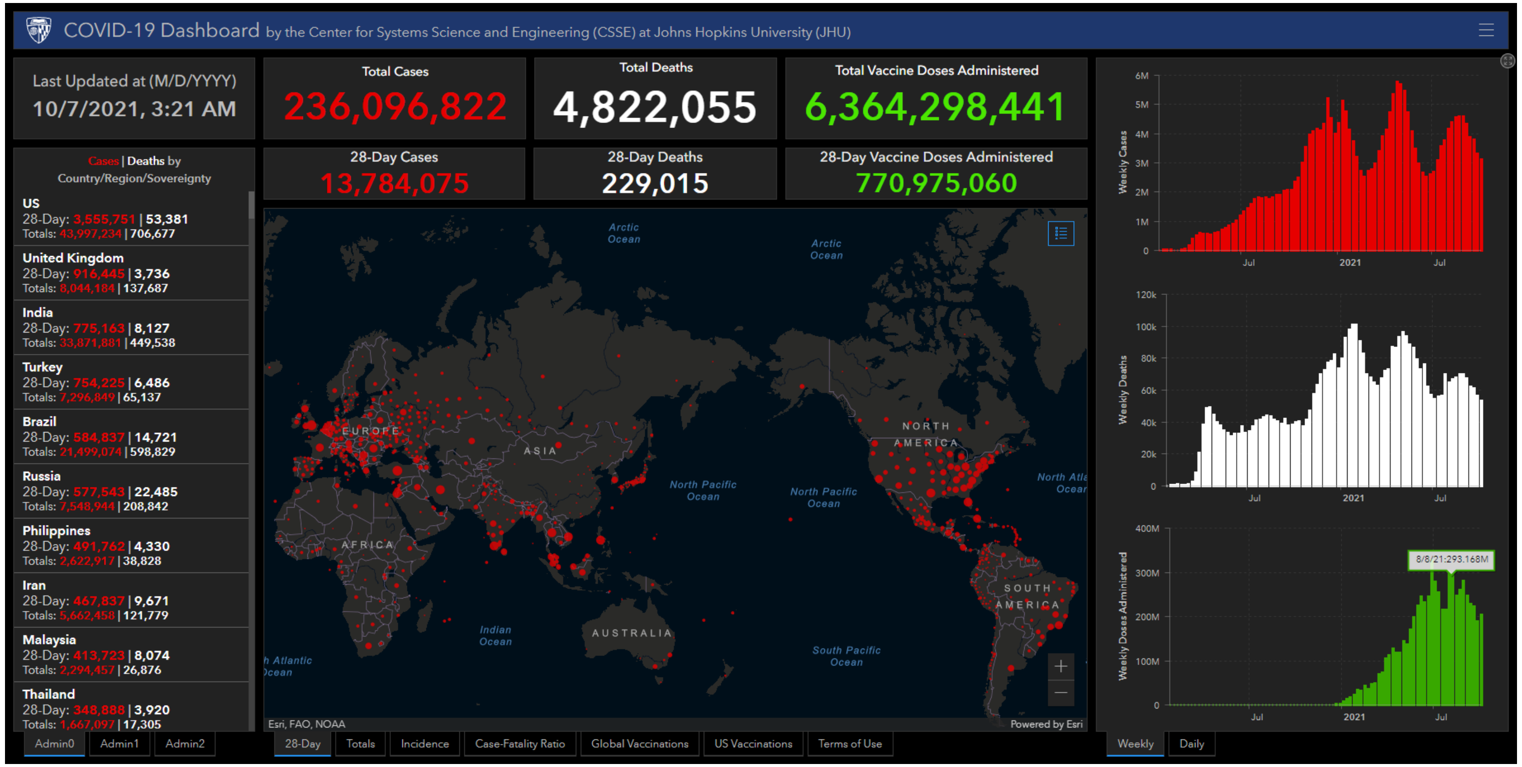


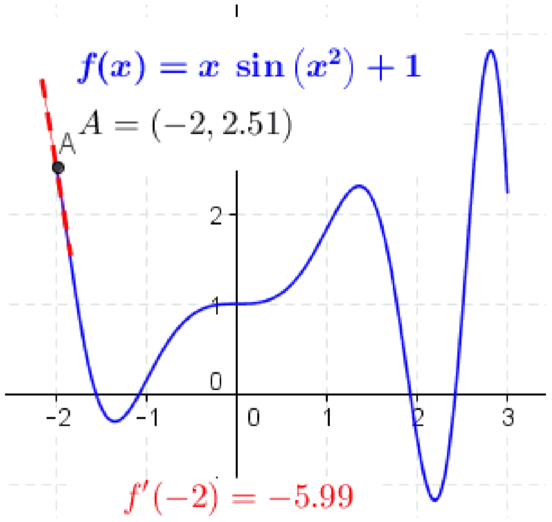
Image: Wikipedia, COVID-19 Dashboard (by JHU)

Calculus

- **Calculus** (originally called *infinitesimal calculus*, 미적분학 in Korean) is the mathematical study of continuous change.
- Two major tools

Differentiation:

Finding a derivative of a function



Integration:

Calculating numbers to a function

by combining infinitesimal increments

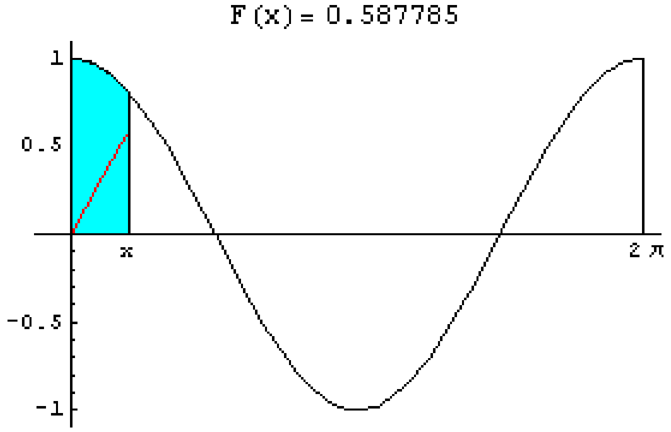
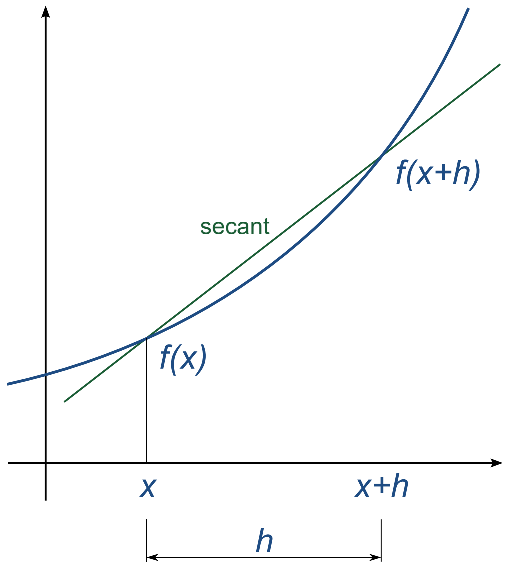


Image: Wikipedia, Docencia con Mathematica

~~Calculus~~ Differentiation

- **Differentiation** is the process of finding a **derivative** of a function.
- **Derivative** [Wikipedia]
  - Definition
    - Change of the function value (**output value**) with respect to a change in its argument (**input value**)

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



  - Meaning
    - The **slope** of a tangent line (plane) to the given function
    - The tangent line ~ **linear approximation** (Note: 1st-order Tayor series)
  - Notation
    - Leibniz's notation: $\frac{df}{dx}(x)$, $\frac{d^2f}{dx^2}(x)$
    - Largrange's notation: $f'(x)$, $f''(x)$
    - Newton's notation: $\dot{y}$, $\ddot{y}$ where $y = f(x)$
    - Euler's notation: $D_x f(x)$, $D_x^2 f(x)$

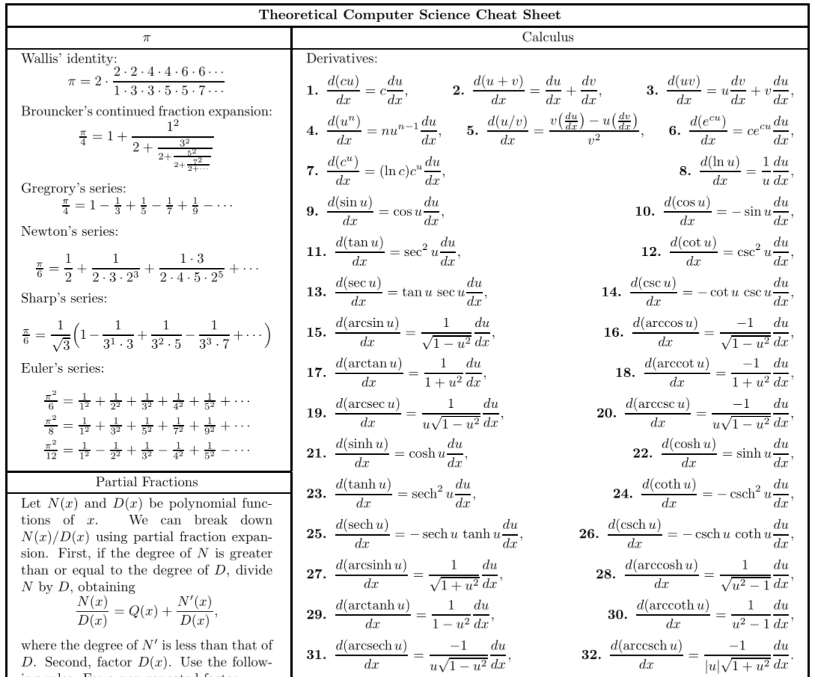
Image: Wikipedia

[Theoretical CS Mathematical Cheat Sheet]

Differentiation

- Rules of computing a derivative [Wikipedia]
  - The polynomial functions such as $f(x) = x^n$
    - $f'(x) = nx^{n-1}$
  - Logarithmic and exponential functions
    - $\frac{d}{dx}\ln x = \frac{1}{x}$
    - $\frac{d}{dx}(e^x) = e^x$ (Note: e - Euler's number, 자연상수 in Korean)
  - Too many functions …

  - **Differentiation is linear.**
    - $(f(x)+g(x))' = f'(x)+g'(x)$ and $(af(x))' = af'(x)$
  - The **product rule** of $h(x) = f(x)g(x)$
    - $h'(x) = f'(x)g(x) + f(x)g'(x)$
  - The **chain rule** of $h(x) = f(g(x))$
    - $h'(x) = \frac{dh(x)}{dx} = \frac{df(g(x))}{dx} = f'\big(g(x)\big)\cdot g'(x) = \frac{df(g(x))}{dg(x)}\cdot\frac{dg(x)}{dx}$



Note) The quotient rule of $h(x) = \frac{f(x)}{g(x)} = f(x)g(x)^{-1}$

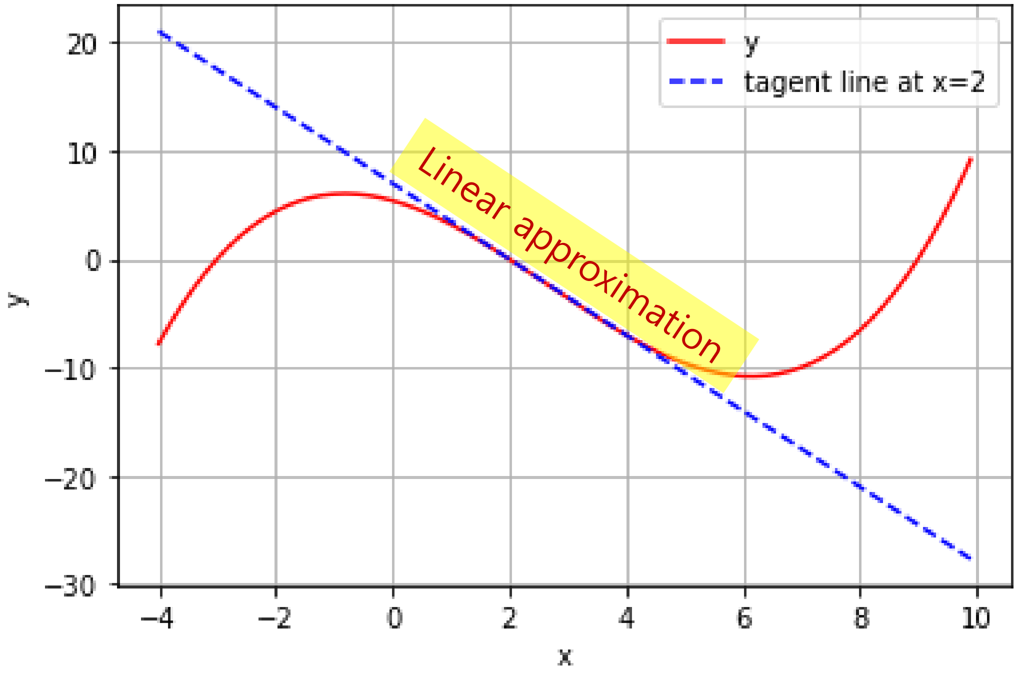
- $h'(x) = \frac{f'(x)g(x) - g'(x)f(x)}{g^2(x)}$

Differentiation

- Example) Finding the slope and tangent line of $y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$ at $x = 2$ $(y = 0)$
  - Derivative function: $\frac{dy}{dx}(x) = 0.3x^2 - 1.6x - 1.5$
  - Derivative value at $x = 2$ $(y = 0)$: $\frac{dy}{dx}(2) = -3.5$
  - Tangent line at $x = 2$ $(y = 0)$: $\mathrm{y} - 0 = -3.5(x - 2)$ $\rightarrow$ $\mathrm{y} = -3.5x + 7$

- Example) Plotting the above results

```
import matplotlib.pyplot as plt

scale = 10
xs = [x/scale for x in range(-4*scale, 10*scale)]
ys = [0.1*x**3 - 0.8*x**2 - 1.5*x + 5.4 for x in xs]
yt = [-3.5*x + 7 for x in xs]

plt.plot(xs, ys, 'r-',  label='y')
plt.plot(xs, yt, 'b--', label='tangent line at x=2')
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
plt.legend()
plt.show()
```

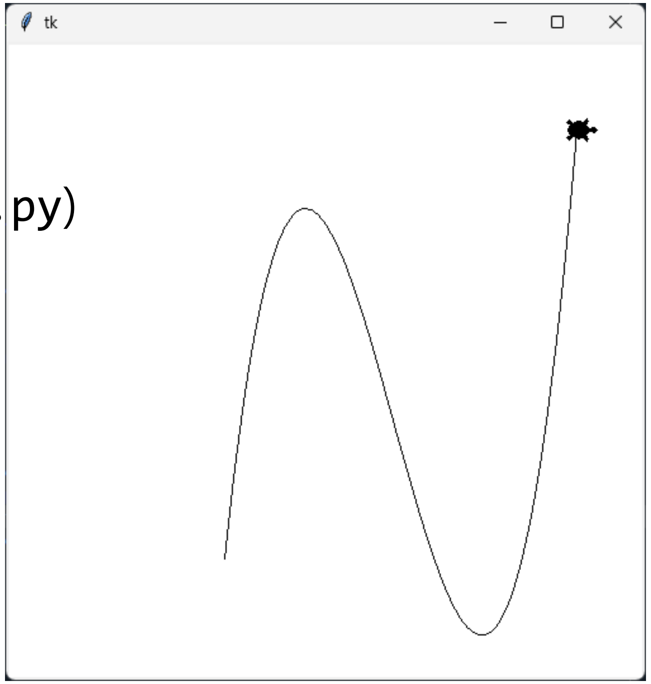


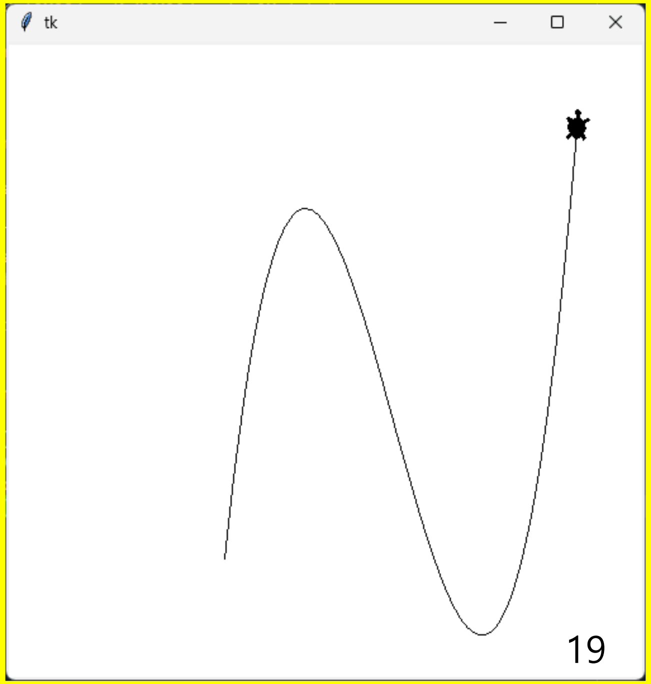
Differentiation

- Example) A turtle moving on $y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$ (`turtle_animation.py`)

```
import tkinter as tk
import turtle, math

# Generate 'y = 0.1*x**3 - 0.8*x**2 - 1.5*x + 5.4'
scale = 10
xs = [x/scale for x in range(-4*scale, 10*scale)]
ys = [0.1*x**3 - 0.8*x**2 - 1.5*x + 5.4 for x in xs]
yd = [0.3*x**2 - 1.6*x - 1.5 for x in xs]

# Prepare 'screen' and 'actor'
root = tk.Tk()
canvas = tk.Canvas(root, width=500, height=500)
canvas.pack()
screen = turtle.TurtleScreen(canvas)
actor = turtle.RawTurtle(screen, 'turtle')
actor.radians() # Use radian unit for angle and rotation

# Draw the function
zoom = 20
actor.penup()
actor.setpos(zoom*xs[0], zoom*ys[0]) # Put at the initial point
actor.pendown()
for (x, y, slope) in zip(xs, ys, yd):
    actor.setpos(zoom*x, zoom*y)
    actor.setheading(math.atan2(slope, 1))
screen.mainloop()
```

How to derive a derivative function of a complex function?





Differentiation

- Example) Finding the derivative function of $y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$ using SymPy

```
import sympy as sp

x, y = sp.symbols('x y')
y = 0.1*x**3 - 0.8*x**2 - 1.5*x + 5.4

yd = sp.diff(y, x)
print(yd)                          # 0.3*x**2 - 1.6*x - 1.5
print(float(yd.subs({x: 2})))      # -3.5 / Casting from Float object to float
```

Note) More Examples with SymPy

- Example) Solve the above equation (find x when $y = 0$) using SymPy

```
roots = sp.solveset(y, x)
print(roots)                       # FiniteSet(-3.0, 2.0, 9.0)
r0 = float(roots.args[0])          # -3.0 / Casting from Float object to float
```

- Example) Factorize the above equation using SymPy

```
sp.pprint(sp.factor(y))            # 5.4·(0.11...·x - 1.0)·(0.33...·x + 1.0)·(0.5·x - 1.0)

y = (x**3 - 8*x**2 - 15*x + 54)/10 # Make the coefficients rational numbers
sp.pprint(sp.factor(y))            # (x - 9)·(x - 2)·(x + 3)
                                   # ───────────────────────
                                   #           10
print(sp.factor(y))                # (x - 9)*(x - 2)*(x + 3)/10
```

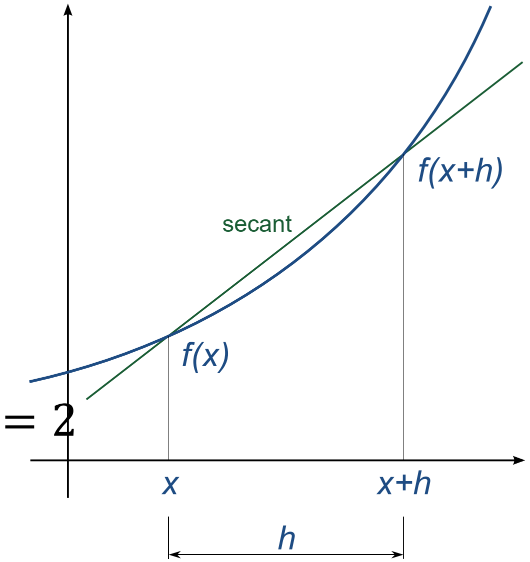
Differentiation

- Example) Finding the derivative function of a composite function using [SymPy]

```
import sympy as sp
from random import random

perceptron = lambda x, w, b: sp.tanh(w * x + b)
f = lambda x: perceptron(x, random(), random())

x, y = sp.symbols('x y')
y = f(f(f(f(f(f(f(x)))))))

yd = sp.diff(y, x)
print(yd)
#0.005*
#(1 - tanh(0.12*x + 0.81)**2)*
#(1 - tanh(0.97*tanh(0.12*x + 0.81) + 0.62)**2)*
#(1 - tanh(0.75*tanh(0.97*tanh(0.12*x + 0.81) + 0.62) + 0.97)**2)*
#(1 - tanh(0.68*tanh(0.75*tanh(0.97*tanh(0.12*x + 0.81) + 0.62) + 0.97) + 0.38)**2)*
#(1 - tanh(0.99*tanh(0.68*tanh(0.75*tanh(0.97*tanh(0.12*x + 0.81) + 0.62) + 0.97) + 0.38) + 0.91)**2)*
#(1 - tanh(0.49*tanh(0.99*tanh(0.68*tanh(0.75*tanh(0.97*tanh(0.12*x + 0.81) + 0.62) + 0.97) + 0.38) + 0
#(1 - tanh(0.16*tanh(0.49*tanh(0.99*tanh(0.68*tanh(0.75*tanh(0.97*tanh(0.12*x + 0.81) + 0.62) + 0.97) +
```

Differentiation

- Numerical differentiation is an algorithm to approximate a derivative value of a function using values of the function. (Note: It does not derive a derivative function.)
  - The definition of derivative: $f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$
  - Two-point estimation: $f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h}$ or $f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h}$



- Example) Finding the slope and tangent line of $y = 0.1x^3 - 0.8x^2 - 1.5x + 5.4$ at $x = 2$
  - Derivative at $x = 2$ $(y = 0)$: -3.5

```
y = lambda x: 0.1*x**3 - 0.8*x**2 - 1.5*x + 5.4
x0 = 2
h = 0.001

d1 = (y(x0+h) - y(x0)) / h       # -3.5002
d2 = (y(x0+h/2) - y(x0-h/2)) / h # -3.5000
```

Image: Wikipedia

Summary

- Building **mathematical intuition**: 1) **Imagine**, 2) **Ask why**
- **Visualization**
  - Matplotlib: A plotting library in Python
  - **Visualization is very important** not only for your **data** but also for your **programs**, **systems**, and **society.**
- ~~Calculus~~ **Differentiation**
  - Derivative definition: $f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x)-f(x)}{\Delta x}$
  - Meaning
    - The **slope** of a tangent line (plane) to the given function
    - The tangent line ~ **linear approximation**
  - How to get a derivative
    - ~~Your hand-on derivation using rules~~
    - Analytical differentiation using SymPy (a symbolic mathematical engine in Python)
    - Numerical differentiation (two-point estimation)