

스마트폰(iOS) 센서데이터와 Python 데이터 분석 환경 구축

오픈 프론티어 2기
강명구

Table Of Contents

- 센서들의 종류와 데이터 특징
- Python 데이터 분석 패키지
 - Numpy, SciPy, Pandas, matplotlib
- Jupyter(iPython) 사용하기

iPhone Sensors



Touch ID

Proximity sensor

Ambient light sensor

Three-axis gyro

Accelerometer

Barometer

iPhone Sensors

Touch ID

지문 센서

Proximity sensor

근접 센서

Ambient light sensor

조도 센서

Three-axis gyro

자이로 센서

Accelerometer

가속도계

Barometer

기압계

정전방식



Touch ID 인식 원리

광방식



습도, 오염에 민감

칩 손상 위험

얇음

지문 채취 도용 불가

습도, 오염에 강함

외부 충격, 굵힘 강함

두꺼움

지문 채취로 도용 가능

에어리어

Touch ID 정전 방식

스와이프



지문 채취 도용 불가

360도 인식 가능

여러번 스캔 필요

공간 필요

지문 채취 도용 불가

정해진 방향만 인식

작은 스캔 필요

작은 공간만을 사용



■ 항문 인식 시스템(Anus Recognition System)



ARS를 장착한 ATM에 항문 인식으로 사용자 인증을 받는 장면



Touch ID 지문 센서

광방식, 정전방식

스와이프, 에어리어(360도)

지문 정보는 이미지가 아니다

외부 접근 불가(애플이라도)

Ring: Ready and Reading

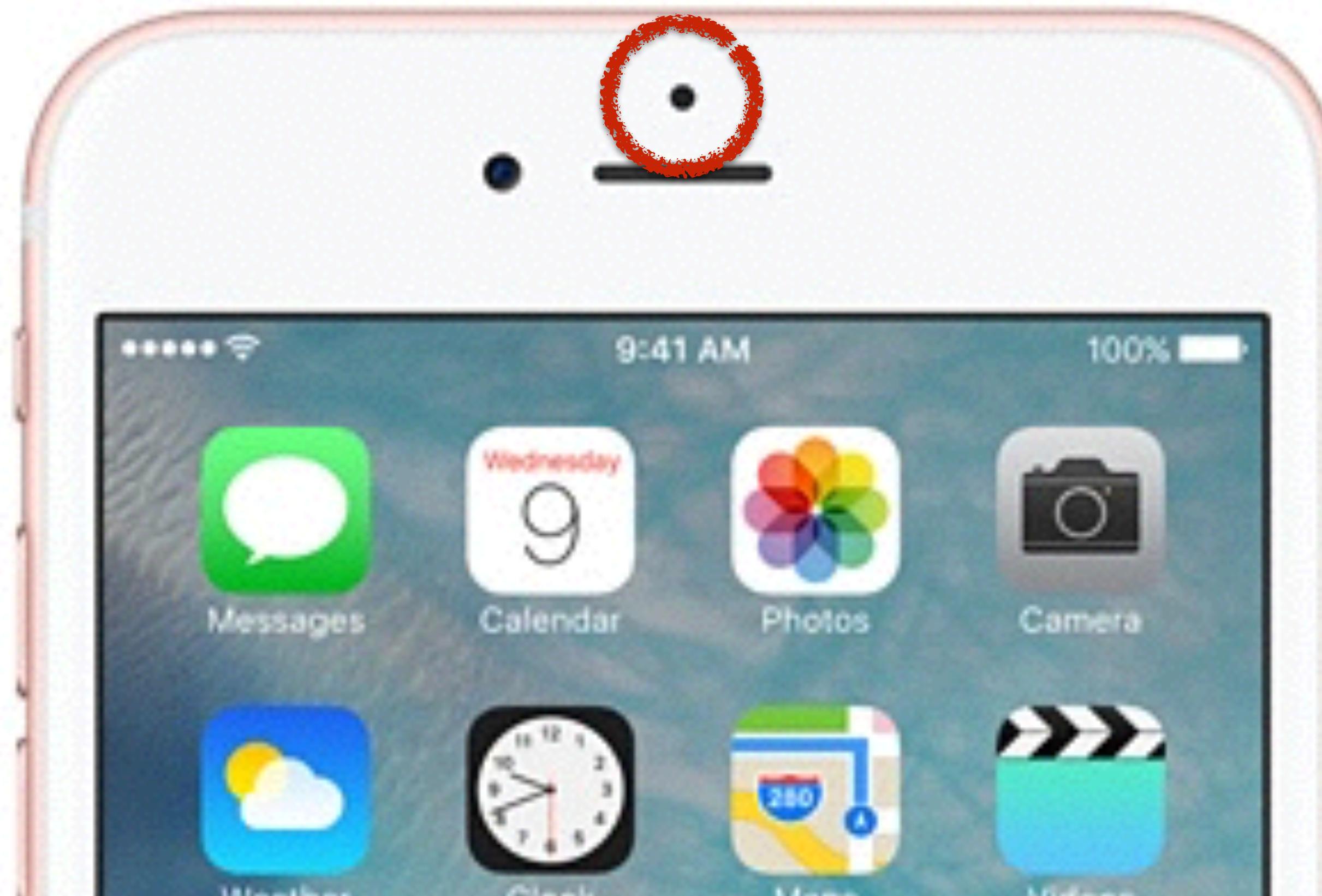
Proximity sensor 근접 센서

UIDevice Class

Using the Proximity Sensor

`proximityMonitoringEnabled` Property

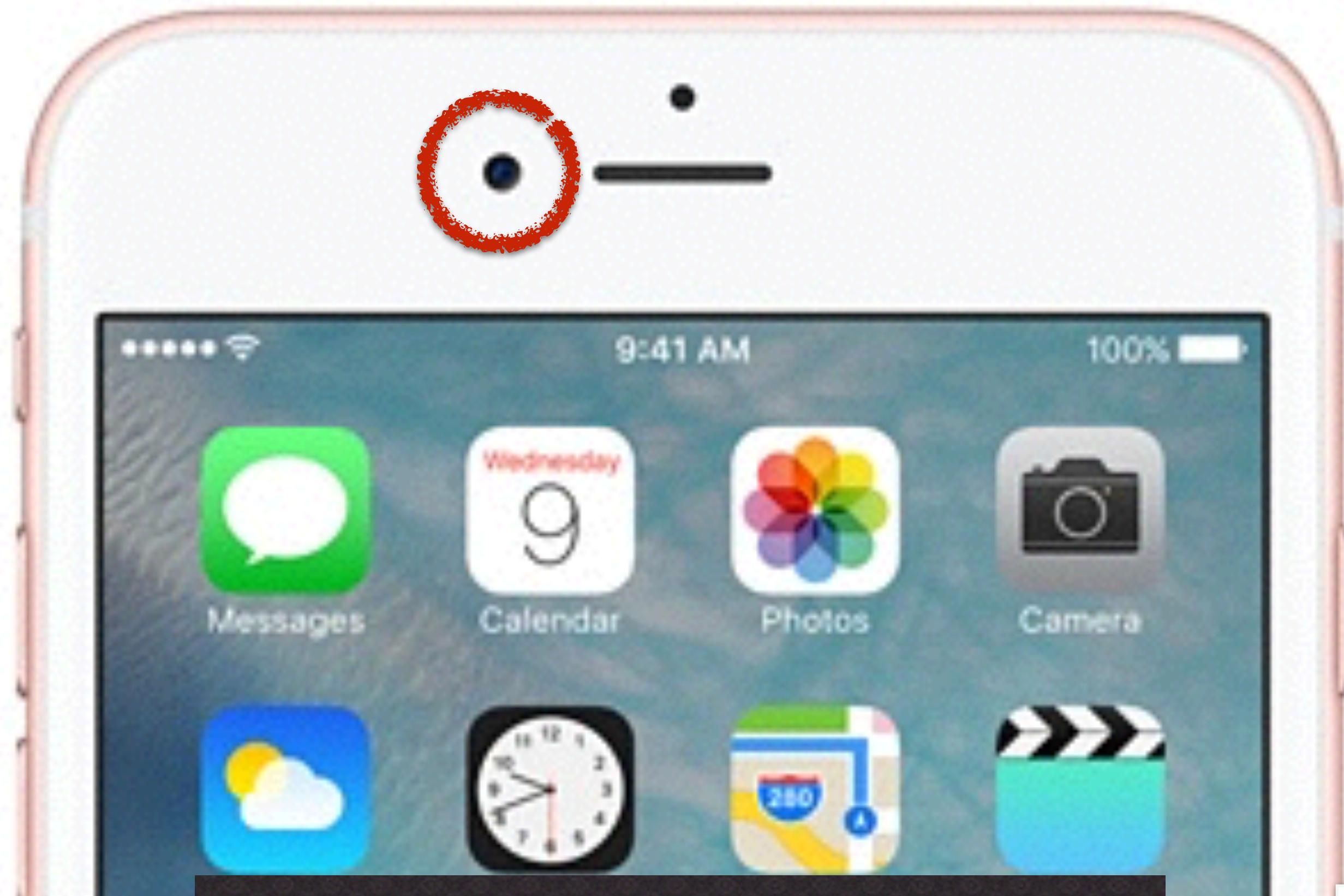
`proximityState` Property



Ambient light sensor 조도 센서

빛의 양을 감지

아이폰에선 사용 불가



Ambient light sensor 조도 센서

빛의 양을 감지

아이폰에선 사용 불가

안드로이드는 됩니다!!



Sensor

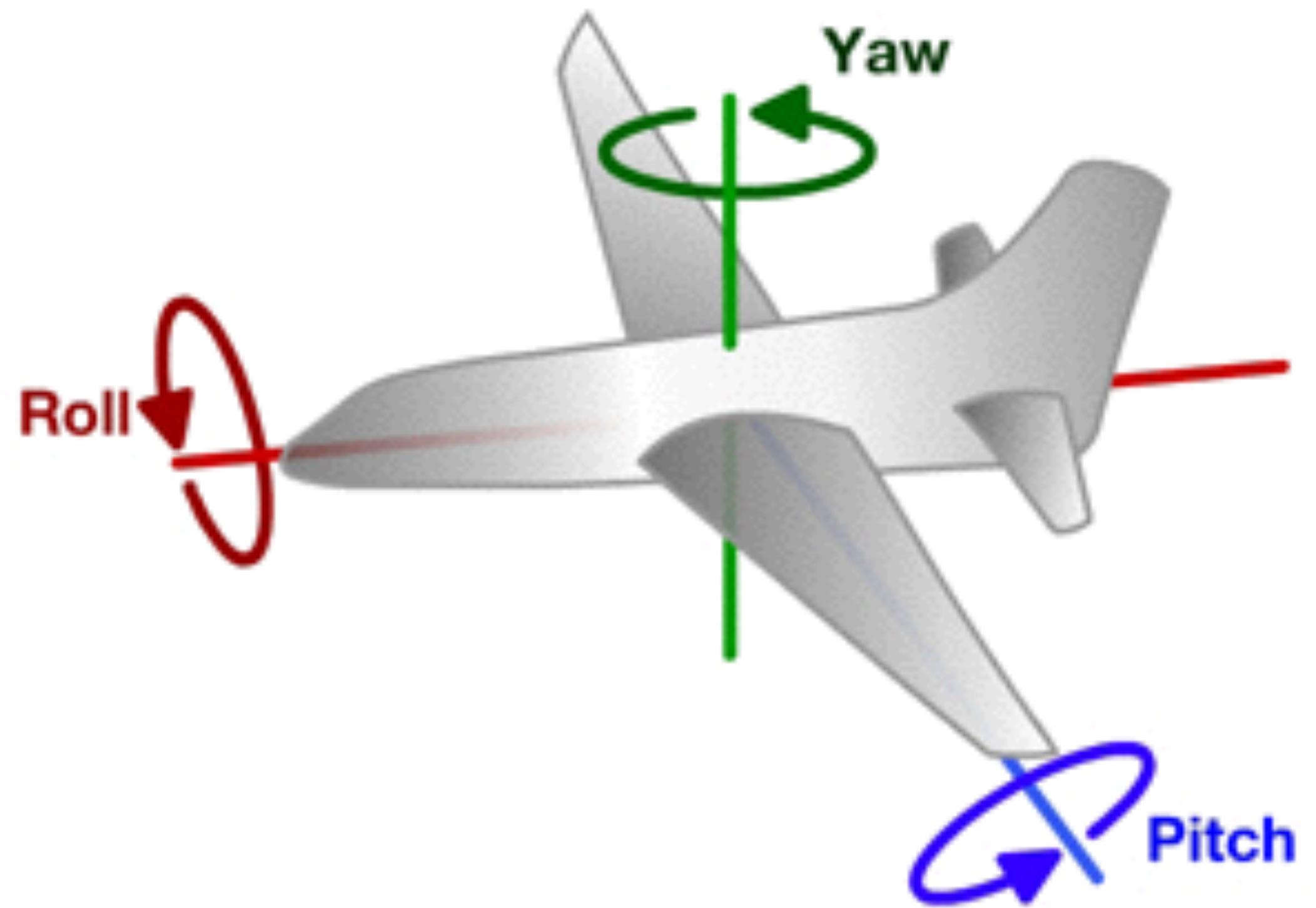
extends [Object](#)

[java.lang.Object](#)
↳ [android.hardware.Sensor](#)

Class Overview

Class representing a sensor. Use [getSensorList\(int\)](#) to get the list of available Sensors.

Gyro 자이로



Rotation Rate Data

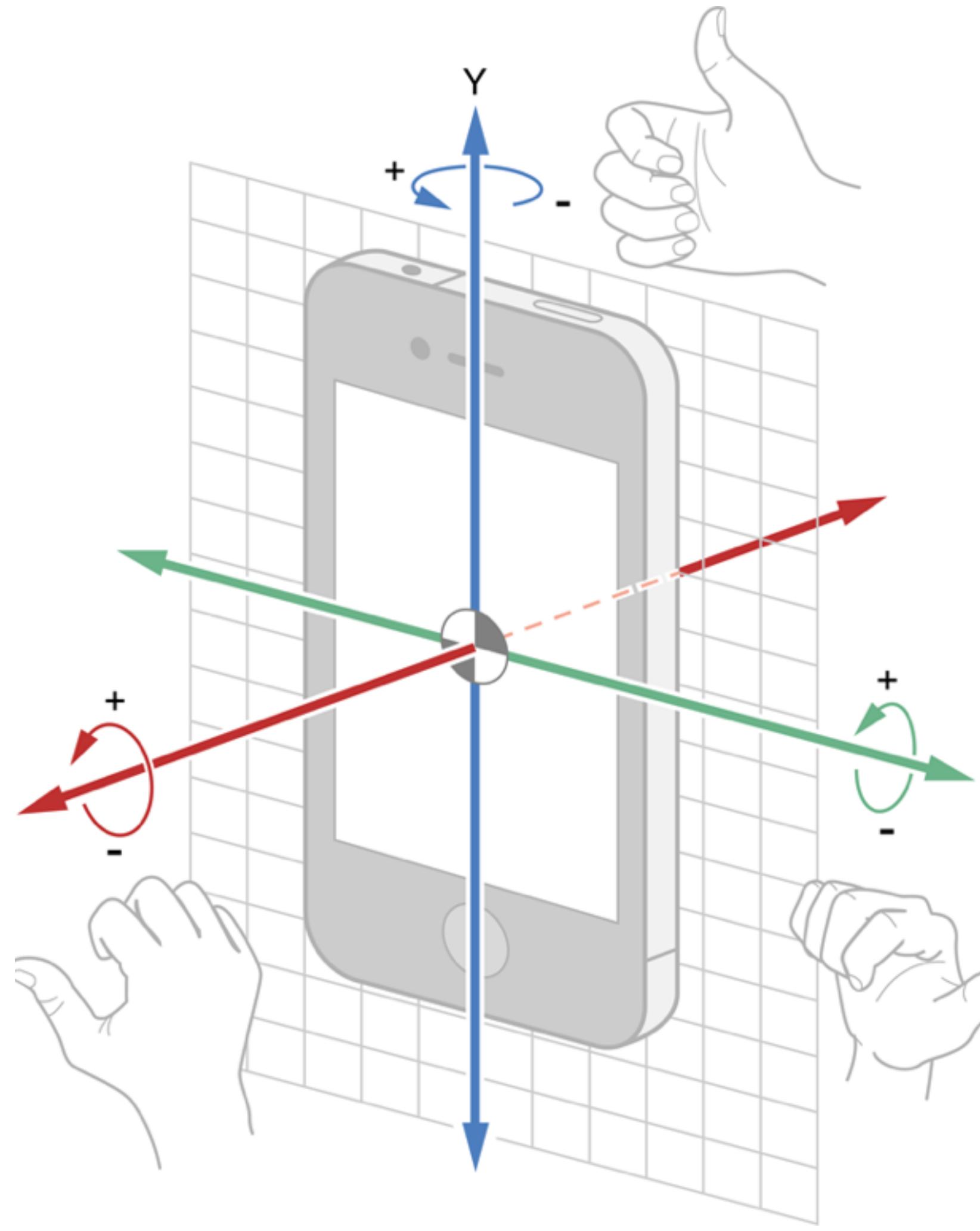
3-axis: x, y, z

x: roll

y: pitch

z: yaw

Gyro 자이로



X	Y	Z
-2.233626	-0.107776	3.660564
-2.233626	-0.107776	3.660564
-1.876707	0.446855	3.508412
-1.353126	0.757885	2.967723
-0.406958	1.060845	2.425728
-0.114053	1.199294	2.267979
-0.070082	1.309804	2.201038
-0.070082	1.309804	2.201038
0.110778	1.424789	2.0422
0.433992	1.058436	1.421444
0.61349	0.130493	0.692136
0.487422	-0.044504	0.407559

Rotation Rate Data

3-axis: x, y, z

x: roll

y: pitch

z: yaw

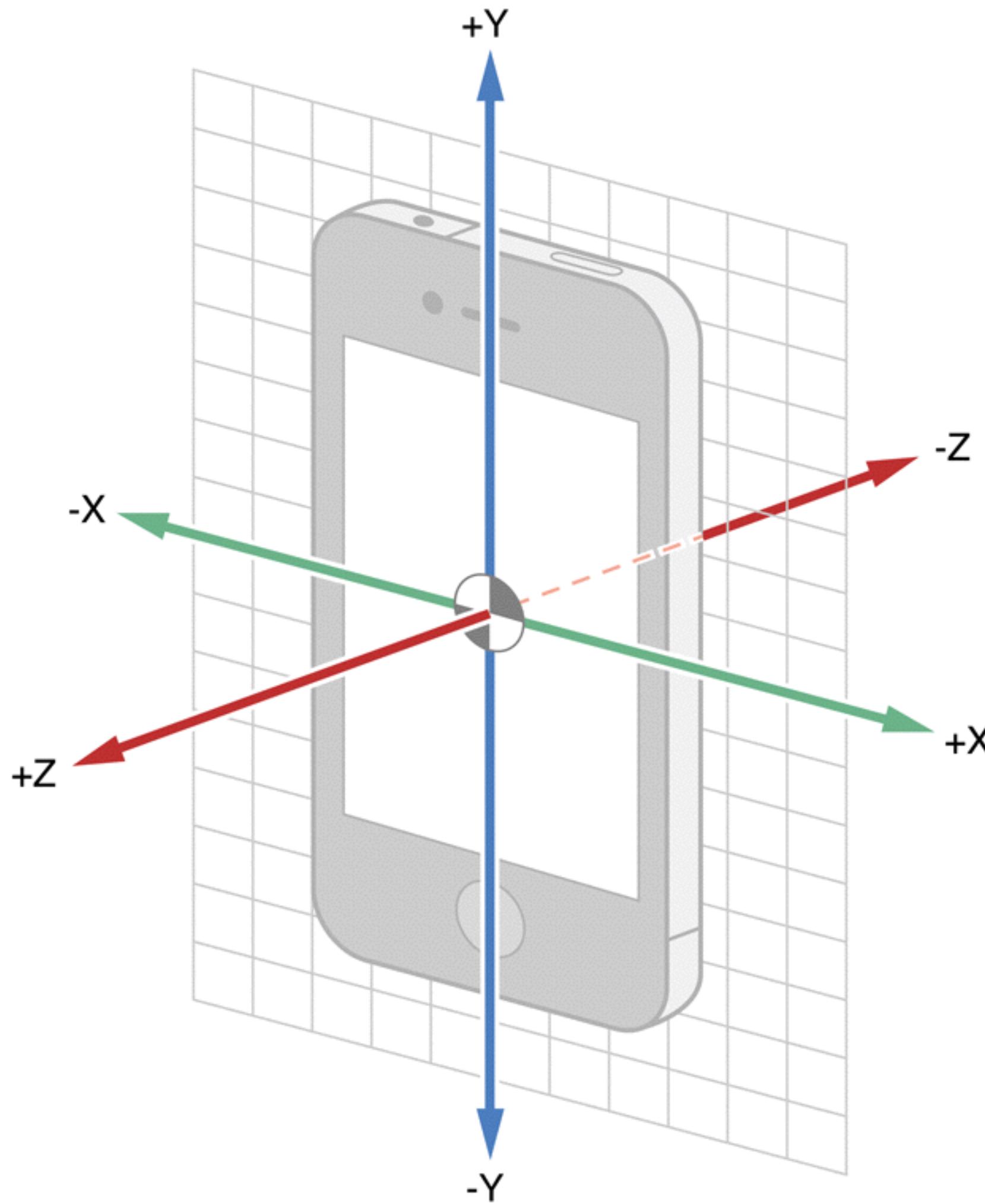
Gyro 자이로

CMMotionManager

Gyroscope. Set the `gyroUpdateInterval` property to specify an update interval. Call the `startGyroUpdatesToQueue:withHandler:` method, passing in a block of type `CMGyroHandler`. Rotation-rate data is passed into the block as `CMGyroData` objects.

단위: Radians per second
Right hand rule
3.14 radian = 180 degree

Accelerometer 가속도계



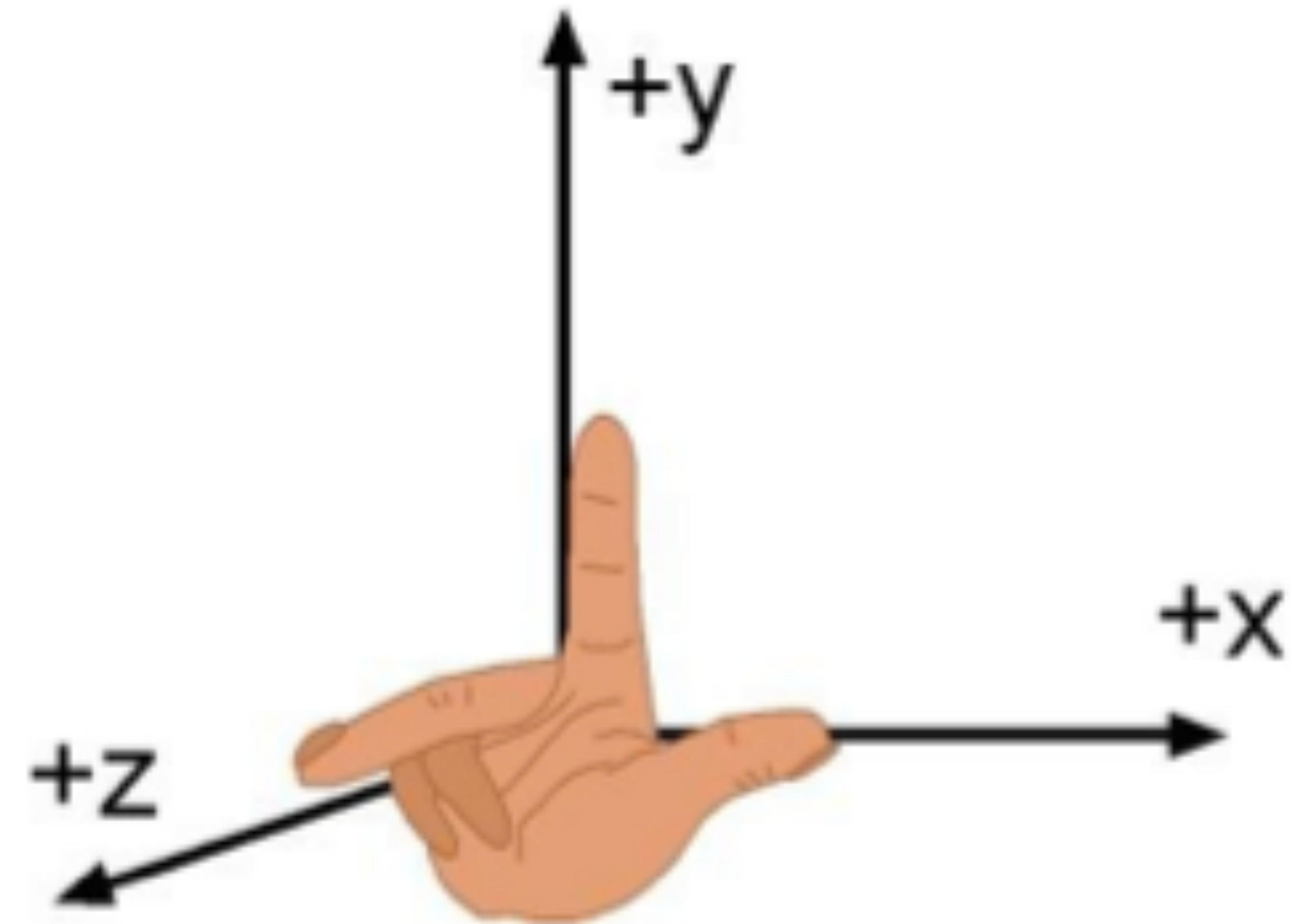
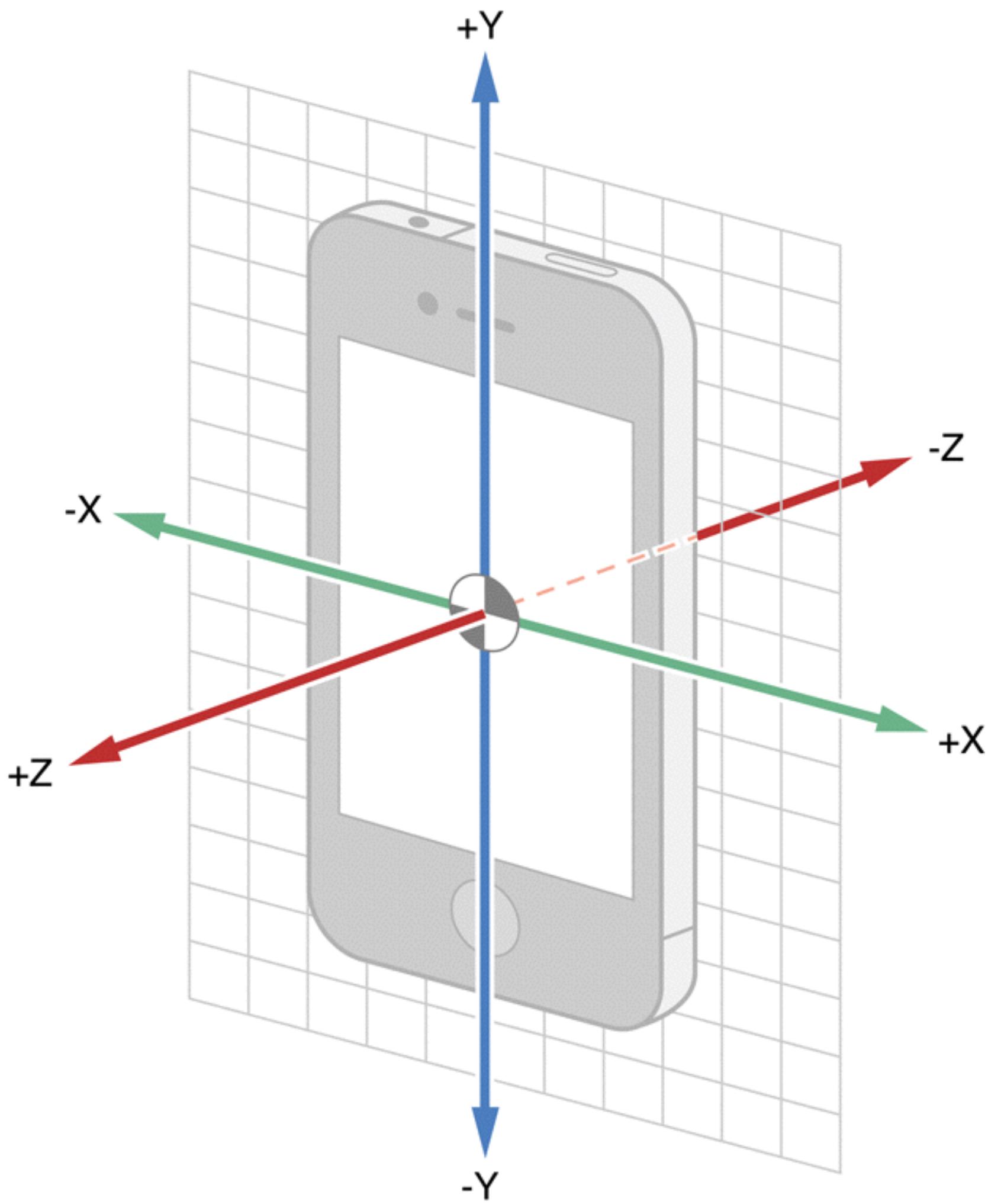
3-axis: x, y, z

$F = ma$ 의 그 a

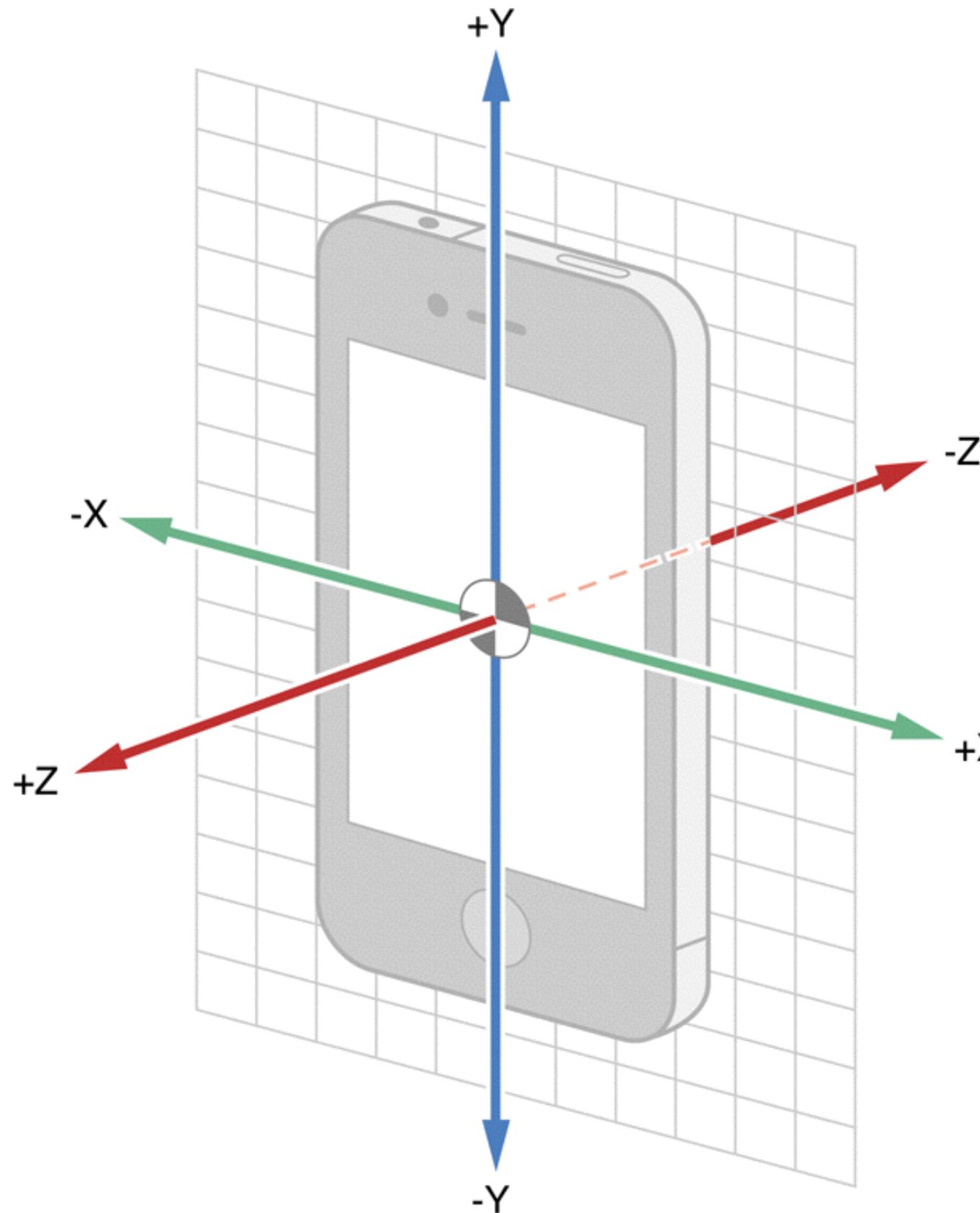
Gravity=9.80665 m/s²

Orientation에 사용

Accelerometer 가속도계



Accelerometer 가속도계

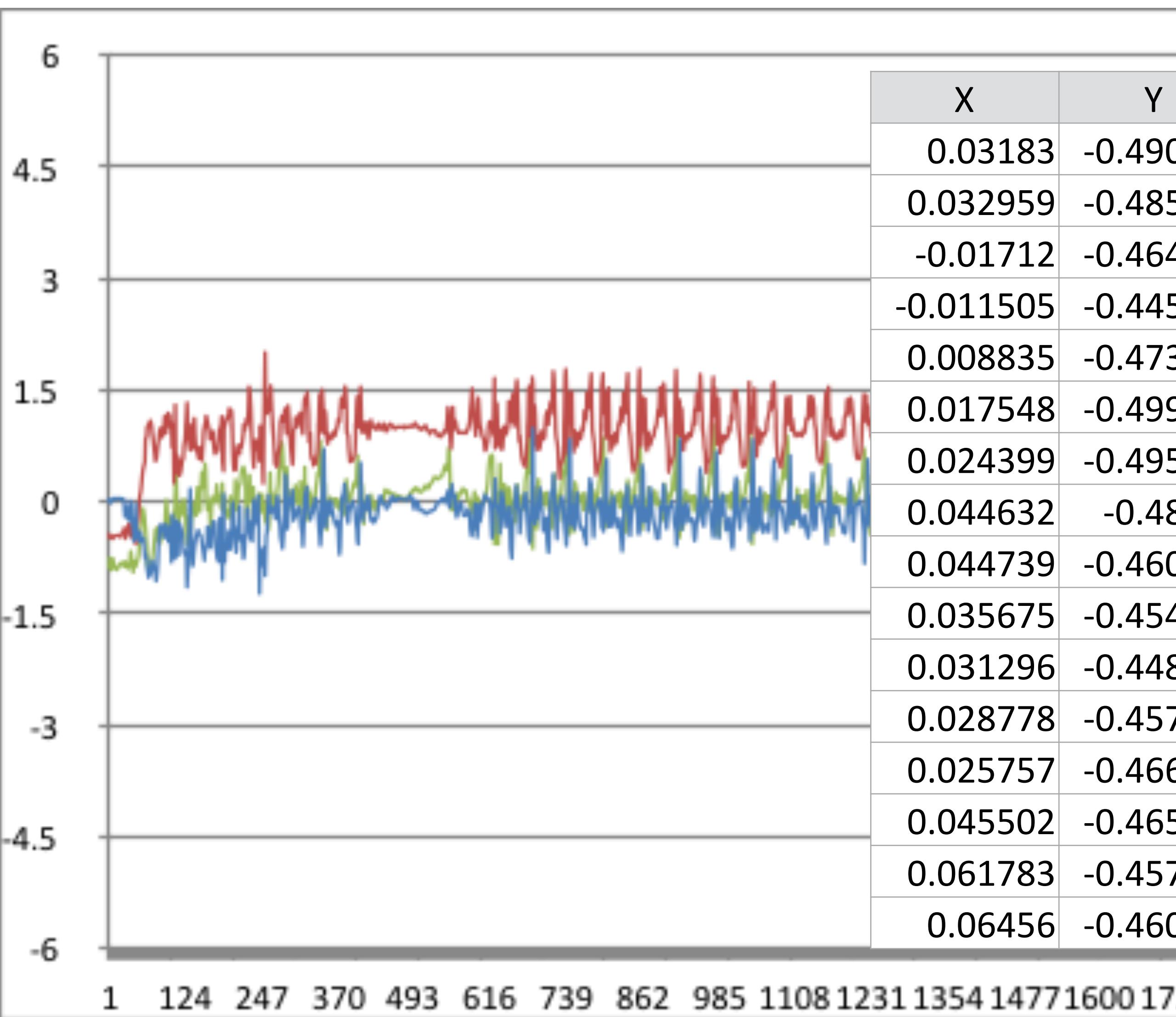


X	Y	Z
0.03183	-0.490067	-0.758514
0.032959	-0.485825	-0.828629
-0.01712	-0.464615	-0.920547
-0.011505	-0.445114	-0.902847
0.008835	-0.473251	-0.825668
0.017548	-0.499802	-0.765472
0.024399	-0.495178	-0.787216
0.044632	-0.48378	-0.837448
0.044739	-0.460266	-0.86821
0.035675	-0.454086	-0.905334
0.031296	-0.448135	-0.919601
0.028778	-0.457001	-0.921387
0.025757	-0.466766	-0.918411
0.045502	-0.465118	-0.920654
0.061783	-0.457001	-0.897217
0.06456	-0.460693	-0.862122

$1G = 9.80665 \text{ m/s}^2$

수평일 경우 Z = -1

Accelerometer 가속도계



수평일 경우 Z = -1

$$V = V_0 + at$$

$$S = vt - \frac{1}{2}at^2$$

오차가 심하다.

Accelerometer 가속도계

CMMotionManager

Accelerometer. Set the `accelerometerUpdateInterval` property to specify an update interval.

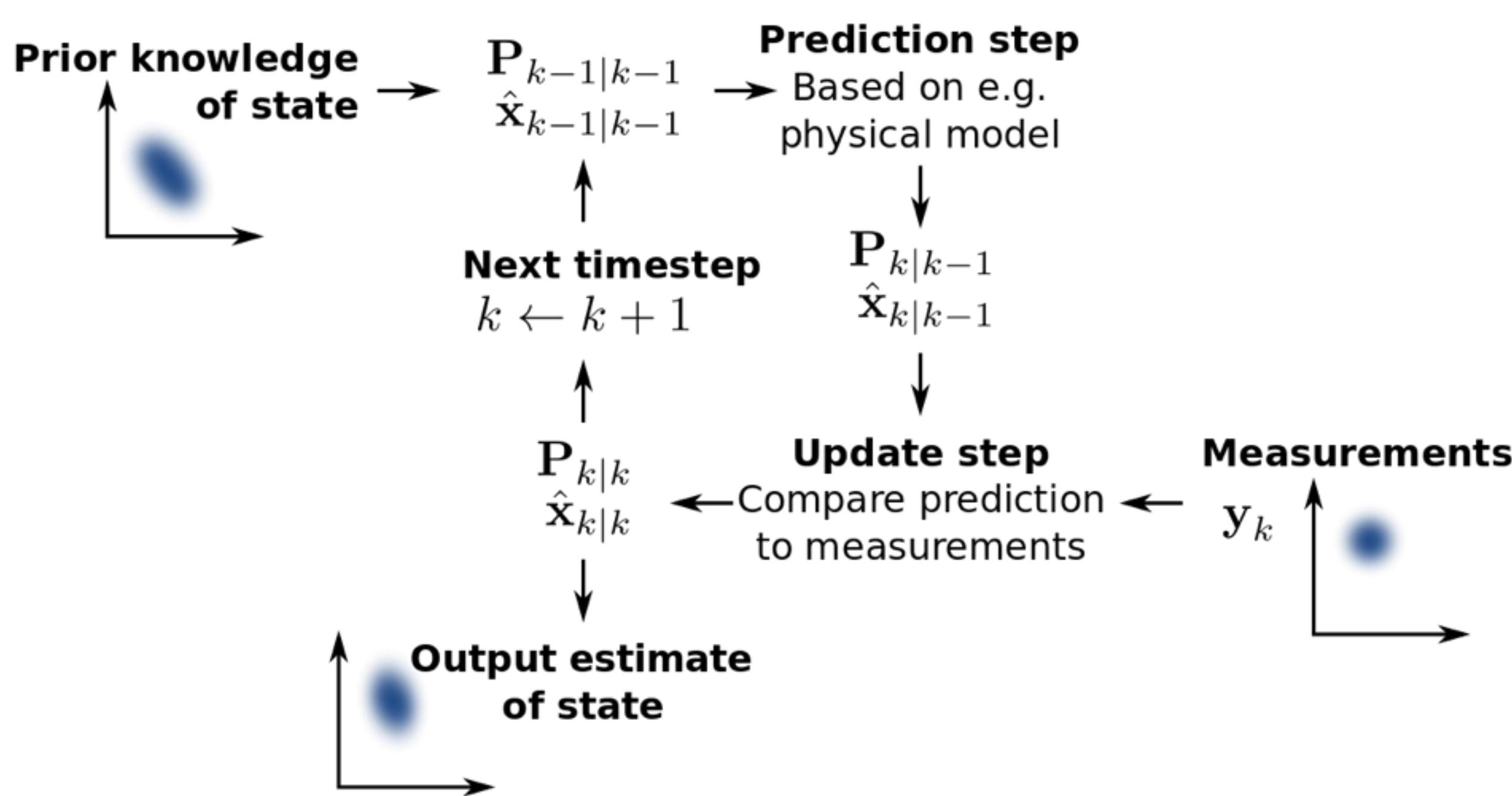
Call the `startAccelerometerUpdatesToQueue:withHandler:` method, passing in a block of type `CMAccelerometerHandler`. Accelerometer data is passed into the block as `CMAccelerometerData` objects.

단위:G

$1G = 9.80665 \text{ m/s}^2$

CMDeviceMotion

Accelerometer + Gyro



$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$
$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$
$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$
$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

CMDeviceMotion

Accelerometer + Gyro

CMMotionManager

Device motion. Set the `deviceMotionUpdateInterval` property to specify an update interval. Call the `startDeviceMotionUpdatesUsingReferenceFrame:` or `startDeviceMotionUpdatesUsingReferenceFrame:toQueue:withHandler:` or `startDeviceMotionUpdatesToQueue:withHandler:` method, passing in a block of type `CMDeviceMotionHandler`. With the former method (new in iOS 5.0), you can specify a reference frame to be used for the attitude estimates. Rotation-rate data is passed into the block as `CMDeviceMotion` objects.

가속도계+자이로 센서+필터
지구 중력을 뺀 가속도값
디바이스의 자세(Attitude) 등
일반적인 용도로는 이것을 사용

Barometer 기압계



기압계(고도계와 다르다)

공기의 압력

높을수록 낮아짐

iOS: kPa(kilopascals)

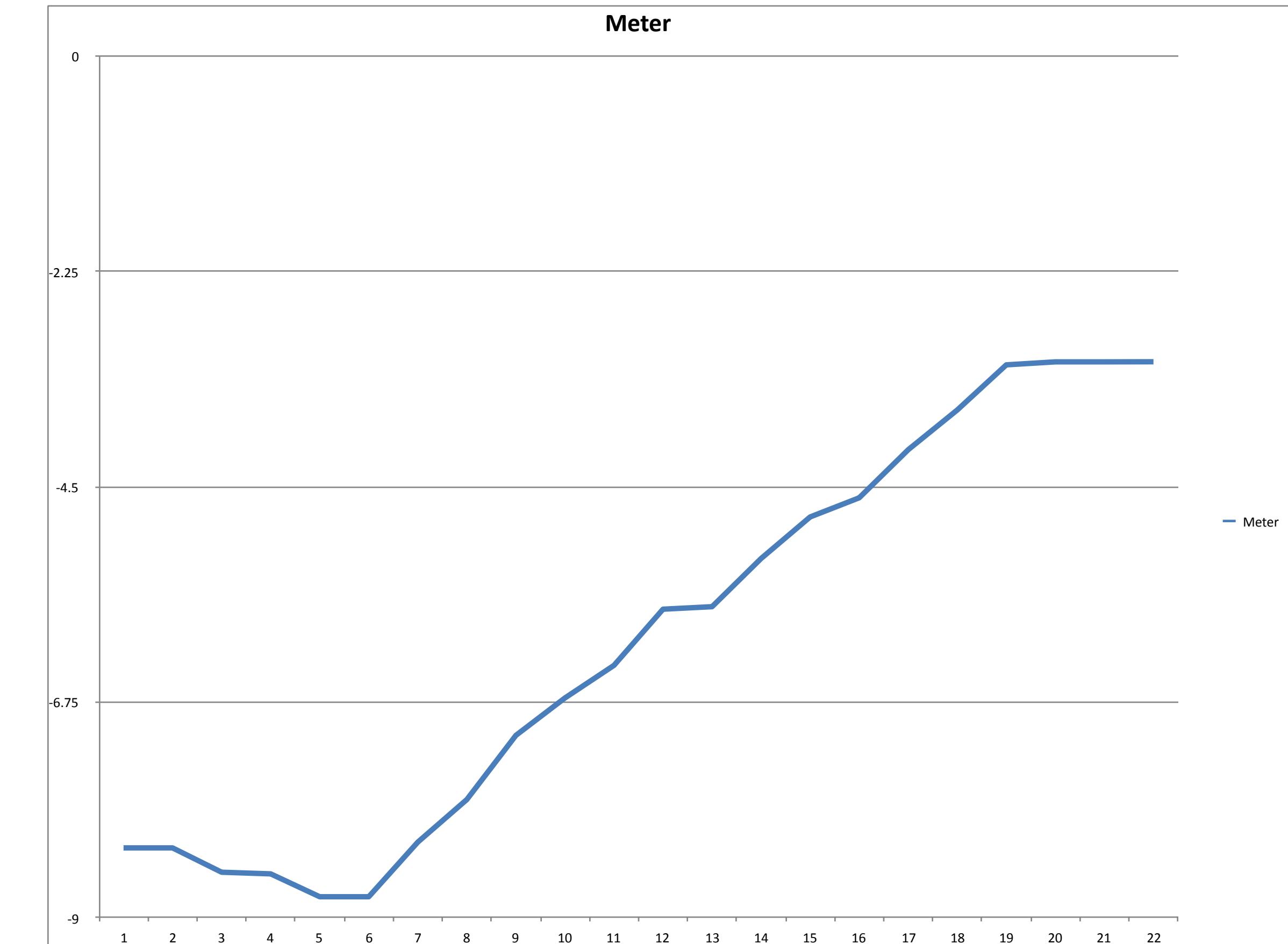
Barometer 기압계



Barometer 기압계

Pressure(kPa)	Meter
101.483543395996	-8.27086448669433
101.483543395996	-8.27086448669433
101.486602783203	-8.52492427825927
101.486808776855	-8.54199504852294
101.489685058593	-8.78098583221435
101.489685058593	-8.78098583221435
101.482818603515	-8.2106122970581
...	...
101.433540344238	-4.11264514923095
101.42854309082	-3.6977367401123
101.422889709472	-3.22654914855957
101.422508239746	-3.19540309906005
101.422508239746	-3.19540309906005
101.422500610351	-3.19439888000488

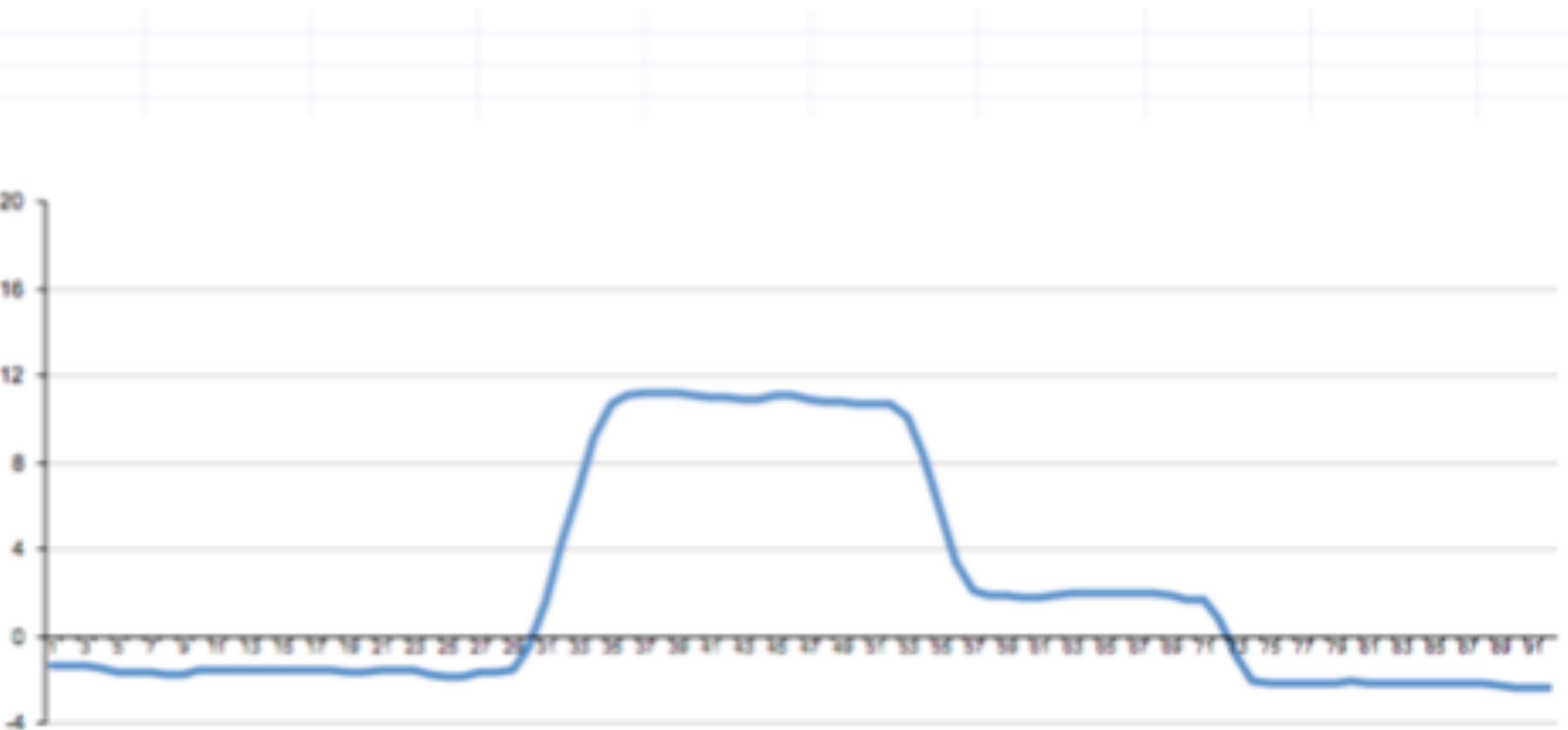
1kPa = 10hPa = 1000Pa



계단 올라가기

Barometer 기압계

-29.468313	100.8	-1.59602356
-30.79002	100.8	-1.72678375
-32.114185	100.8	-1.83078384
-33.429889	100.8	-1.83078384
-34.747306	100.8	-1.70860672
-36.067434	100.8	-1.63994598
-37.386097	100.8	-1.58340073
-38.706482	100.78	-0.589763641
-40.031438	100.76	1.549
-41.34833	100.73	4.235
-42.661902	100.7	6.708
-43.984342	100.67	9.126
-45.302844	100.65	10.66
-46.622748	100.64	11.097
-47.947819	100.64	11.188
-49.26358	100.64	11.188
-50.587298	100.64	11.155
-51.903116	100.64	11.139
-53.221763	100.65	10.988
-54.545499	100.65	10.959
-55.861761	100.65	10.936
-57.181707	100.65	10.936
-58.498221	100.64	11.049
-59.81557	100.64	11.049



엘레베이터

Apple Watch

WatchOS 2.0



Heart Rate

Accelerometer

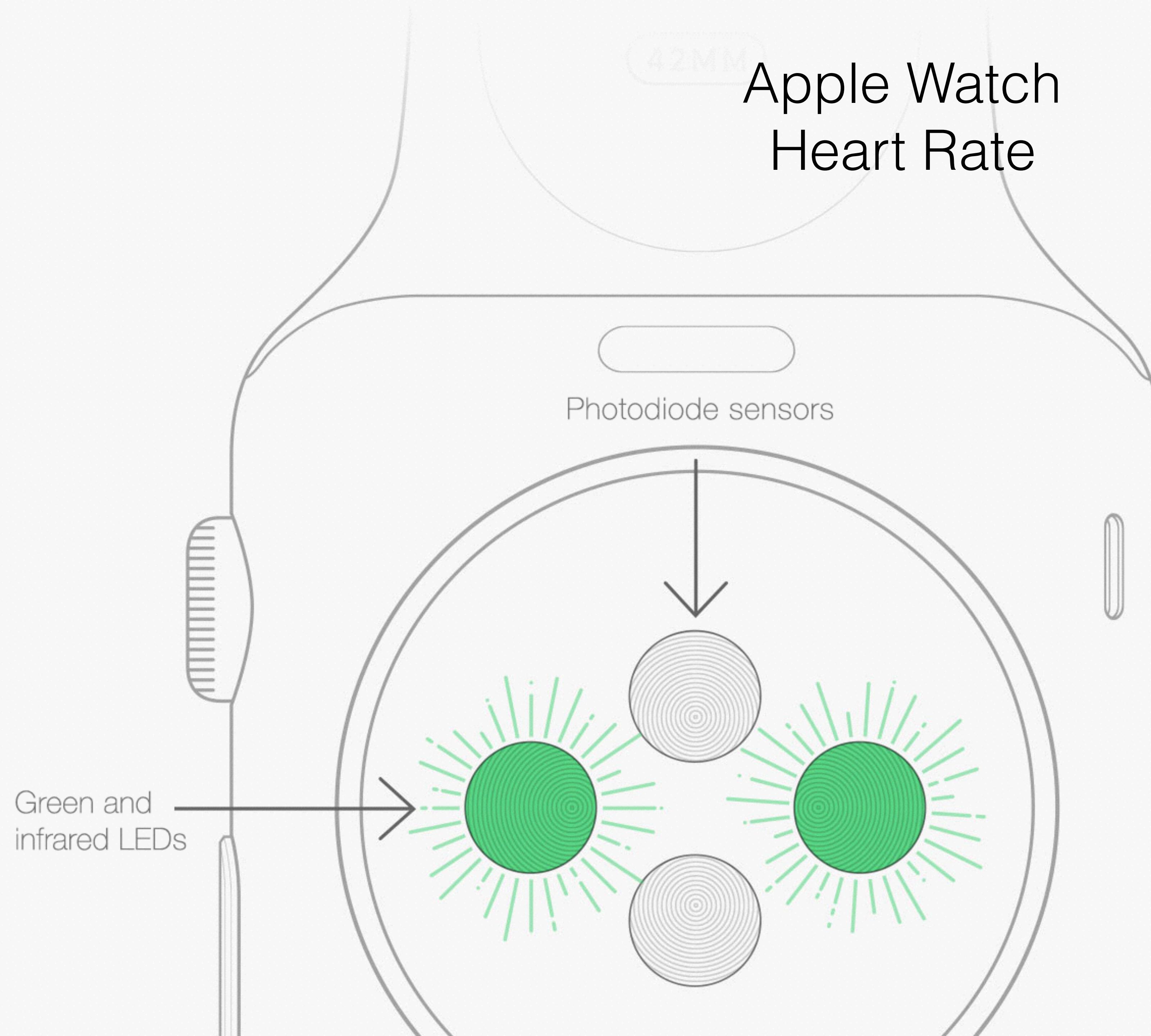
Gyro(접근 불가)

Apple Watch Heart Rate



photoplethysmography
(광혈류락측정)

Apple Watch Heart Rate



photoplethysmography
(광혈류량측정)

Green LED

Photodiode: 반사된 빛

심장박동사: 혈류량 많아짐

심장박동사이: 혈류량 적어짐

센서 응용 분야



Python으로 Matlab 대신하기



Matlab이나 파이썬을 잘 몰라요...

수단이라 생각하세요.





알맞게 설치하세요

<https://www.python.org>

Anaconda

데이터 분석 혹은 공학, 수학 등을 위한 파이썬 통합패키지



pandas
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

matplotlib



알맞게 설치하세요

<https://www.continuum.io>



웹 인터프리터

```
$ conda install jupyter
```

jupyter Pyplot_Tutorial Last Checkpoint: 12/09/2015 (autosaved)

File Edit View Insert Cell Kernel Help Python 3

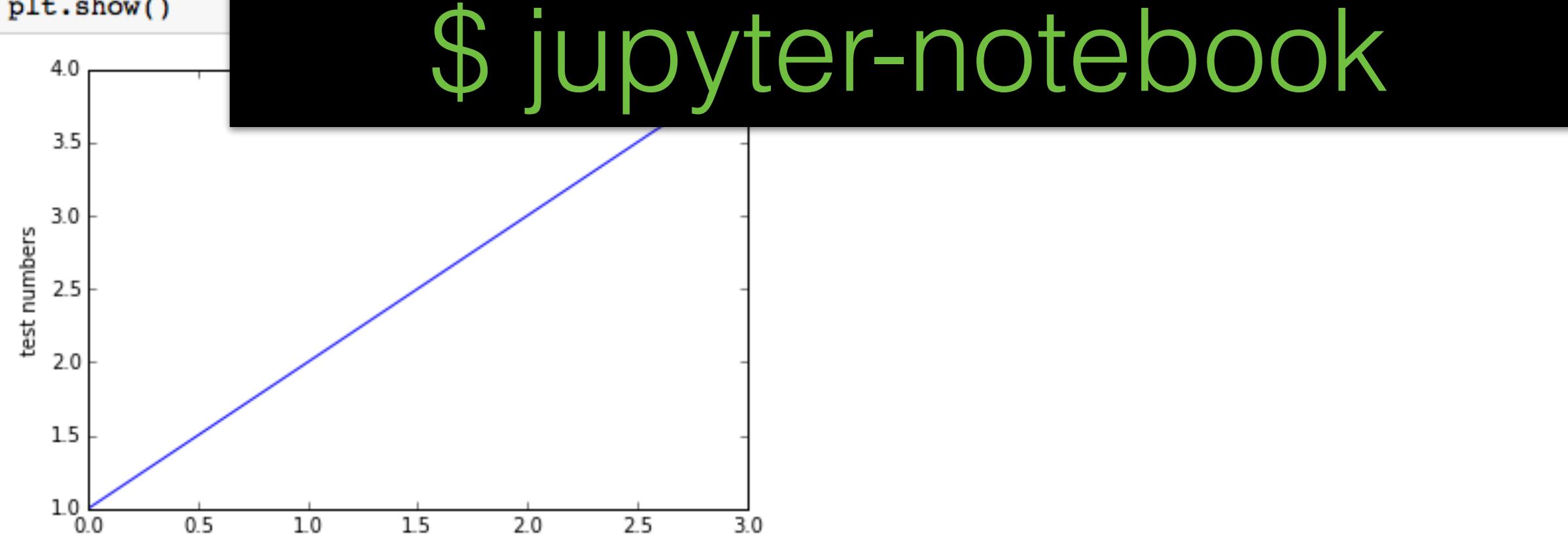
Cell Toolbar: Markdown

Pyplot tutorial

이 문서는 [original](#)을 기준으로 학습하며 써내려간 튜토리얼이다. 애매한 용어를 그대로 사용했다.

`matplotlib.pyplot`은 간단히 말해서 python에서 MATLAB과 같은 그래프 동작을 수행하게 해주는 command style functions의 콜렉션이다. 이것은 function calls을 넘겨서 상태를 유지하므로 현재 figure와 plot부분을 기록하여(tracking) 동작된다. (MATLAB과 같은 방식이다. 가령 figure에서 데이터 입력->x축 설정->y축 설정->그래프 그리기->x축 변경 등이 가능하다.

```
In [1]: # jupyter notebook 내부에 그래프를 표시해준다.  
%matplotlib inline  
import matplotlib.pyplot as plt  
plt.plot([1,2,3,4])  
plt.ylabel('test numbers')  
plt.show()
```



x축의 범위는 따로 입력하지 않을 경우 0부터 시작하여 y값의 변화량과 맞춰지게 된다. 즉 x 데이터는 [0, 1, 2, 3]으로 자동 생성된다.

`plot()`은 범용적(versatile)인 커맨드이므로 임의의 숫자들을 인자들로 받을 수 있다. 예로, x 축에 대응하는 y축을 입력할 경우에는 아래의 커맨드를 참조하자.

```
In [2]: x = [1, 2, 3, 4]  
y = [0, 3, 5, 6]  
plt.plot(x, y)  
plt.show()
```

MarkDown

각종 언어 사용 가능

기록 누적 가능

Github 지원

Numpy, SciPy, Pandas

NUMeric Python

N-dimensional Array
Object

데이터들의 다차원 배열

다양한 수학 라이브러리

Broadcasting

SCientific Python

Based on Numpy

수학적 알고리즘 모음

필요한 거만 골라 쓰면 됨

Pandas

Based on Numpy

퍼포먼스 차이 X

Dictionary처럼 사용가능

High lv Data 다룸

NumPy Broadcasting

```
x = np.array([[1,2,3], [4,5,6], [7,8,9]])
```

```
y = np.array([1,3,5])
```

```
z = x + y
```

x: [[1 2 3]
 [4 5 6]
 [7 8 9]]

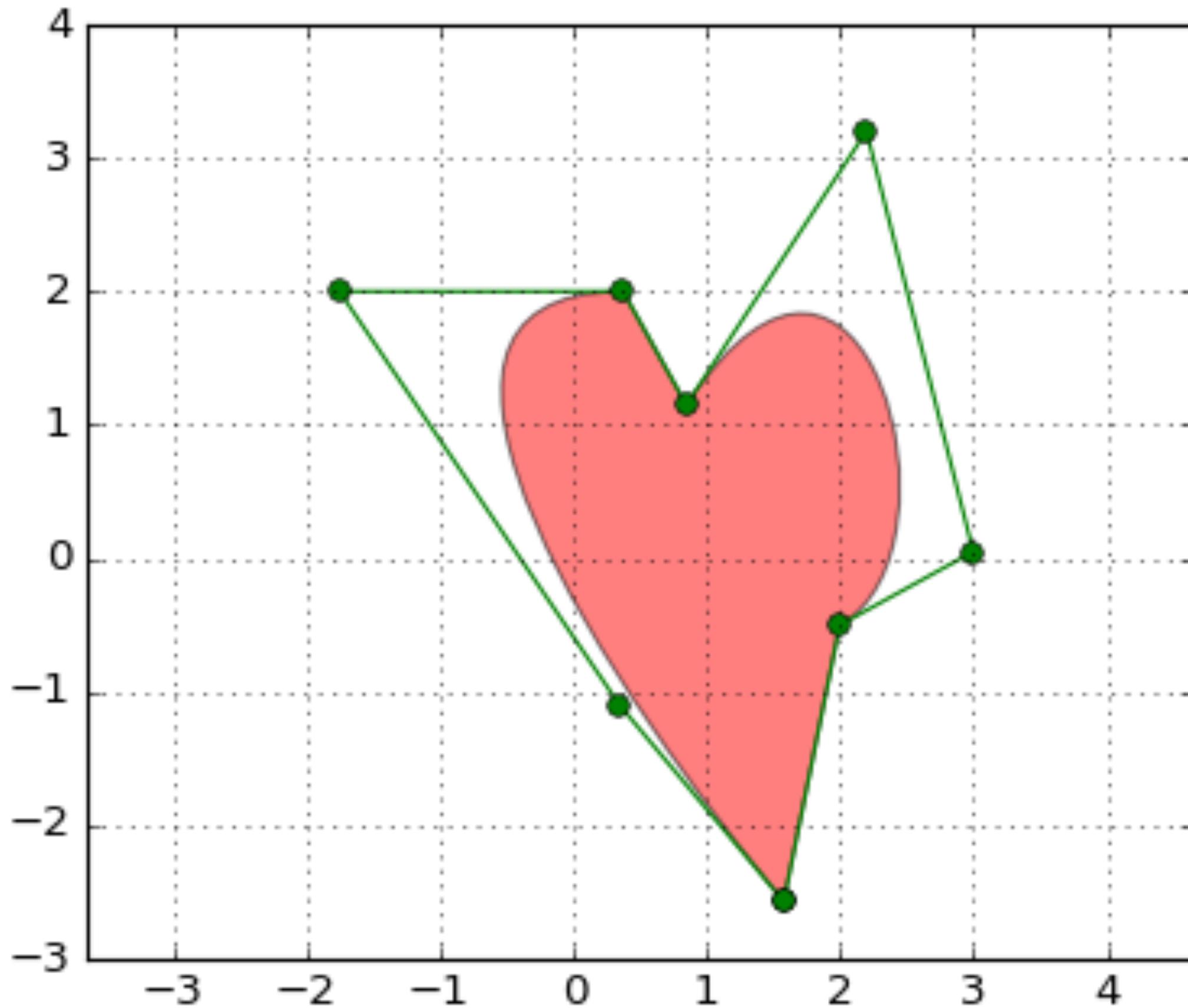
+

y: [1 3 5]

=

x+y:
[[2 5 8]
 [5 8 11]
 [8 11 14]]

matplotlib.pyplot

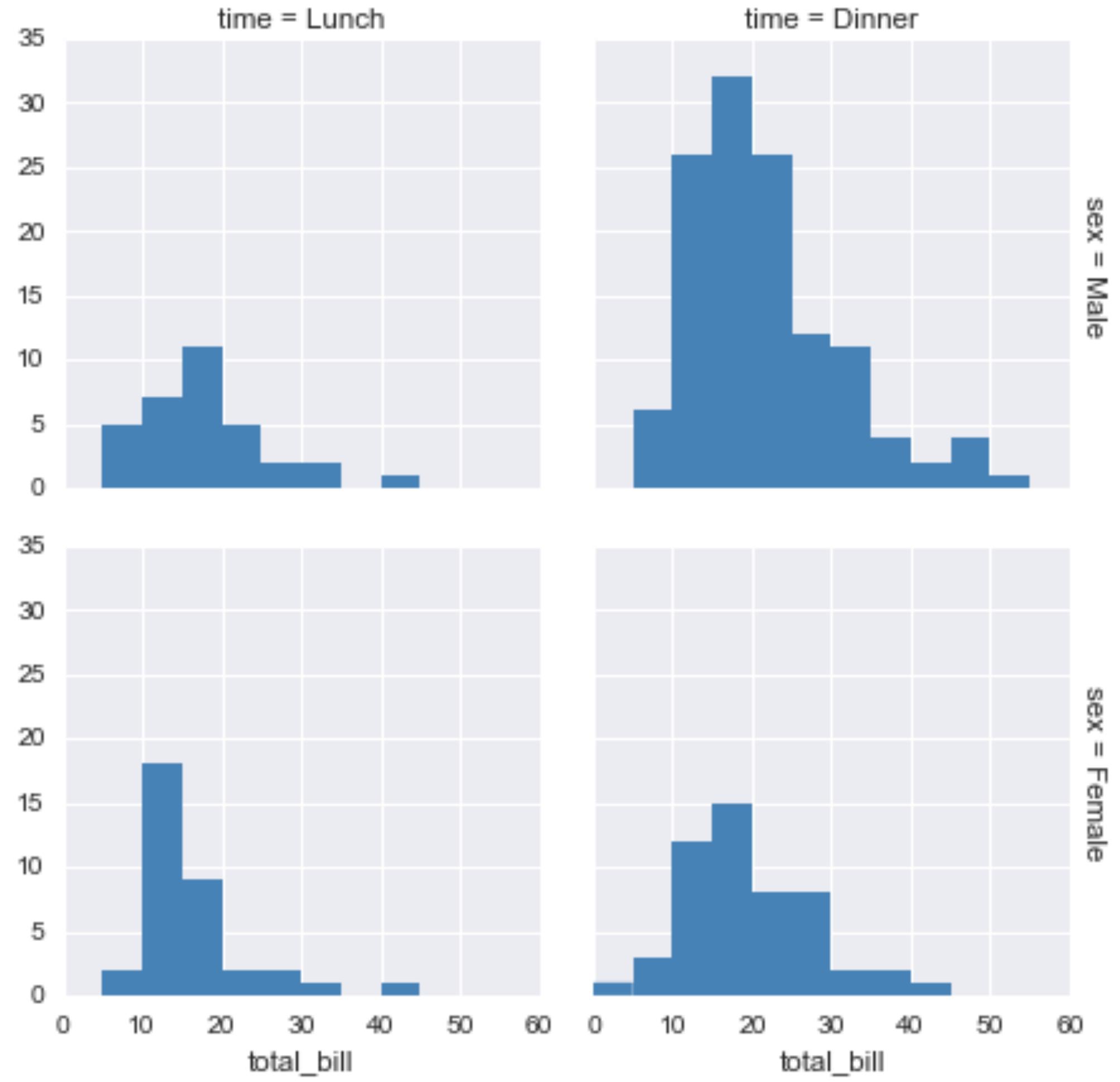


Python Graph Library

Figure, Plot Tracking

Matlab과 같은 방식

Seaborn



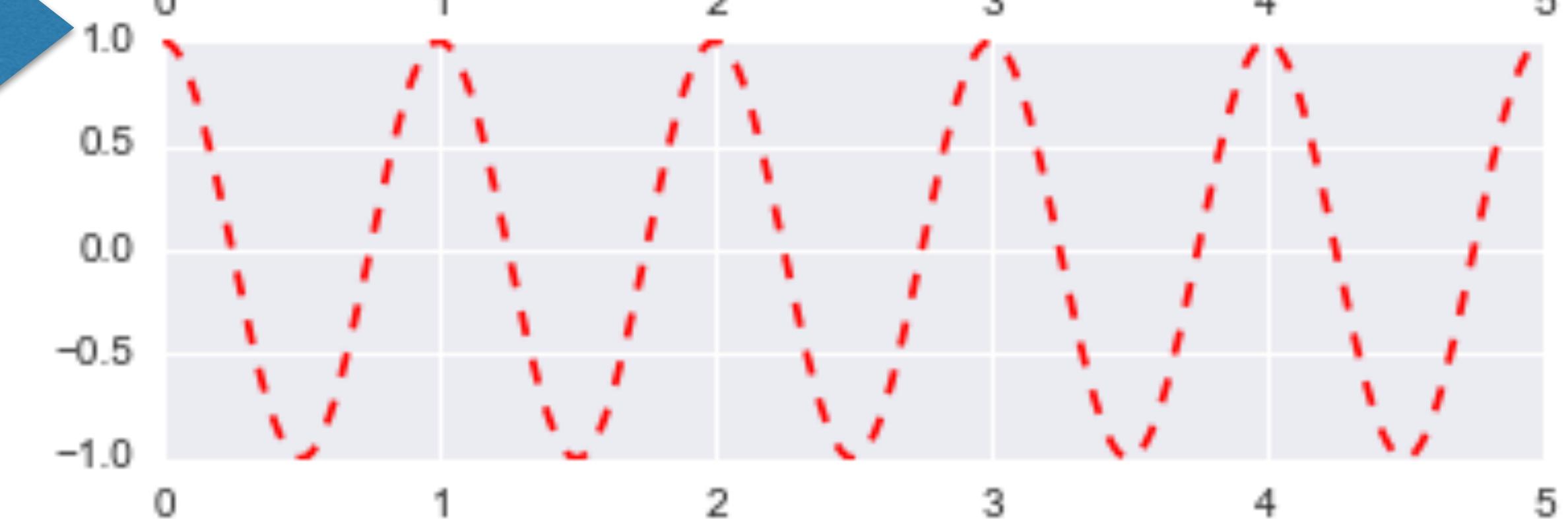
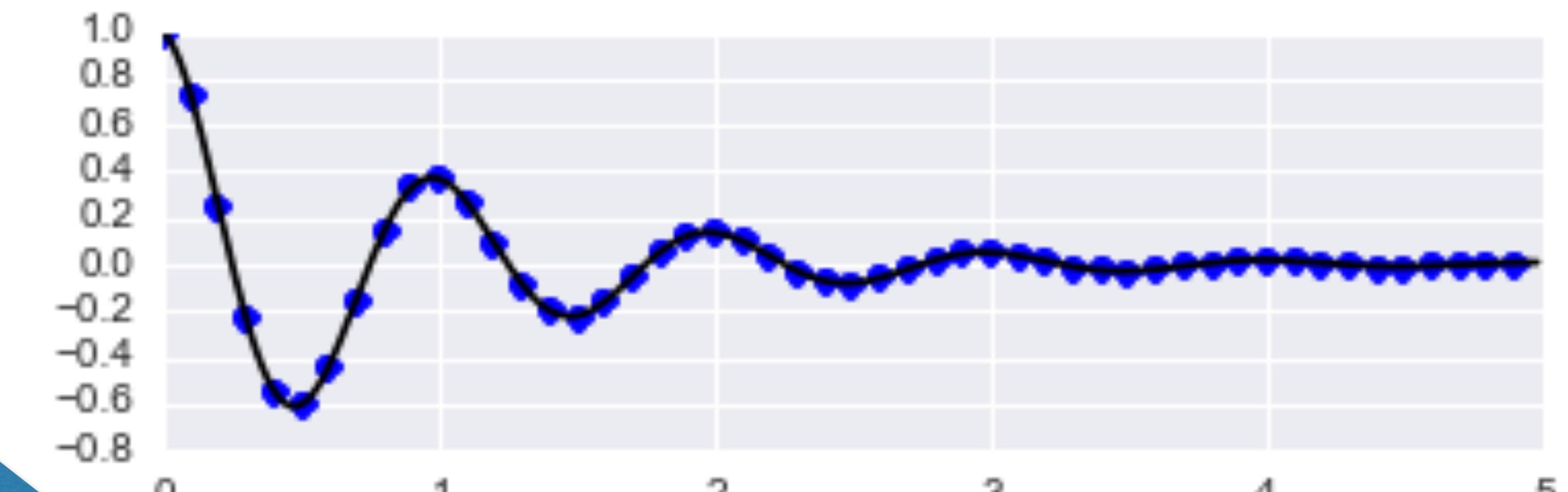
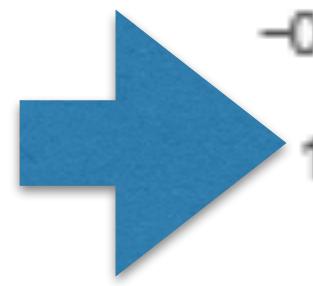
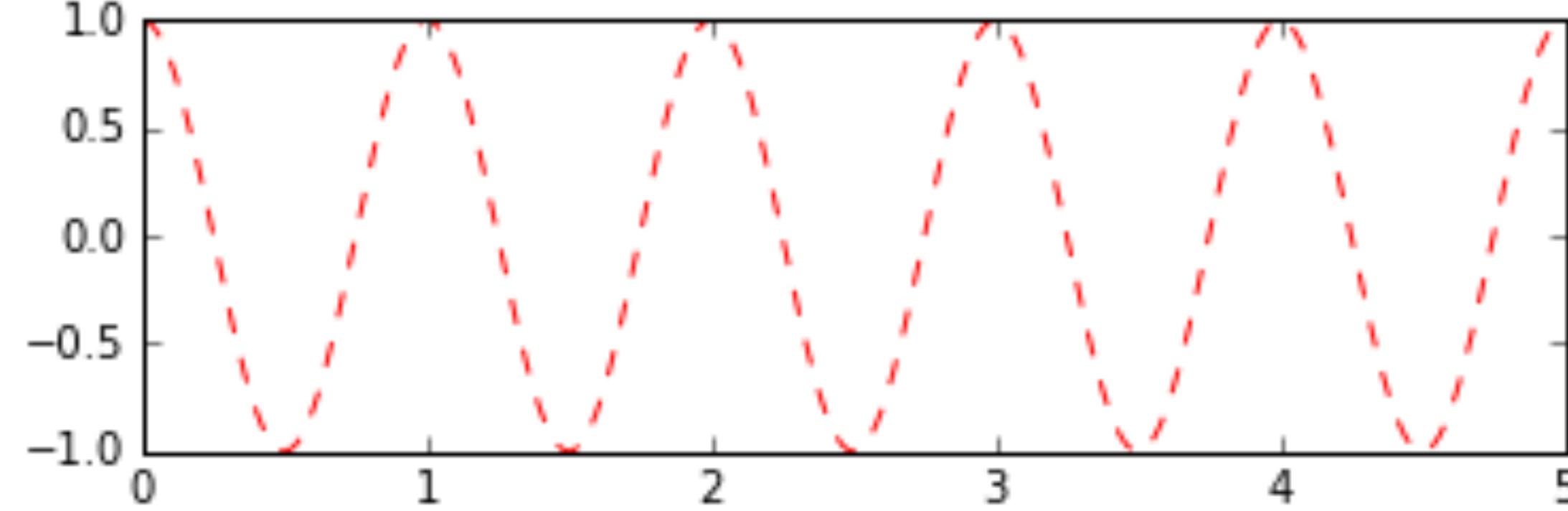
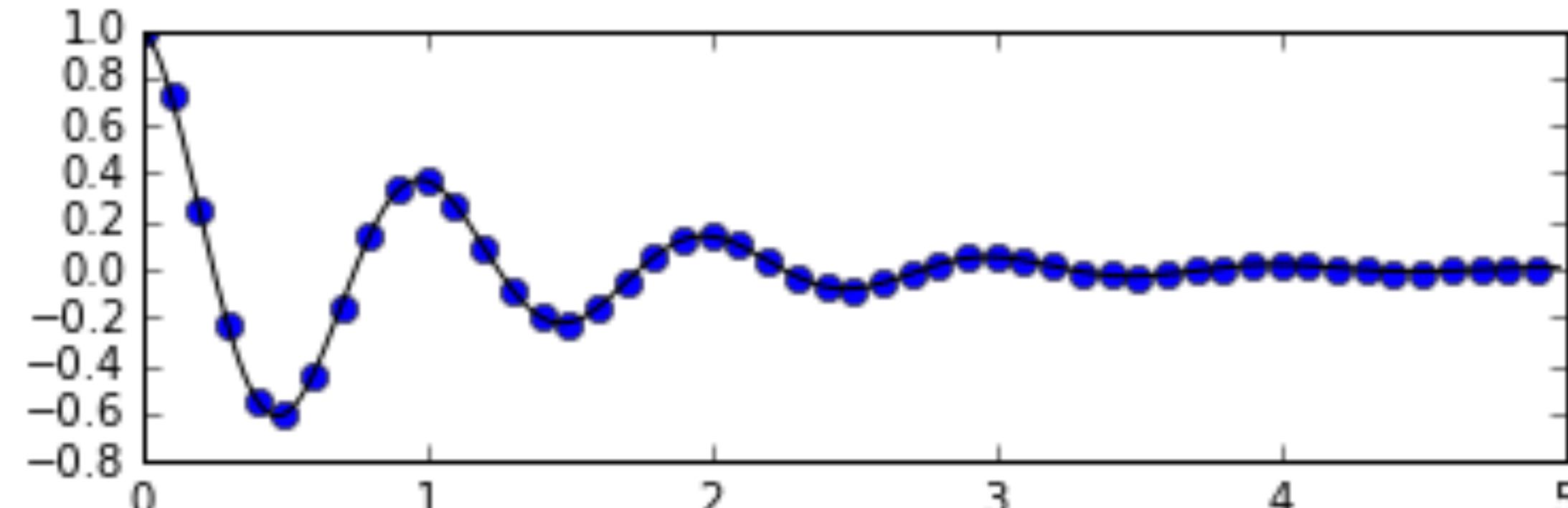
```
$ conda install seaborn
```

matplotlib.pyplot 기반

이쁜 그래프

Seaborn

```
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn
```



Jupyter 상에서 데이터 다루기

```
In [34]: %matplotlib inline

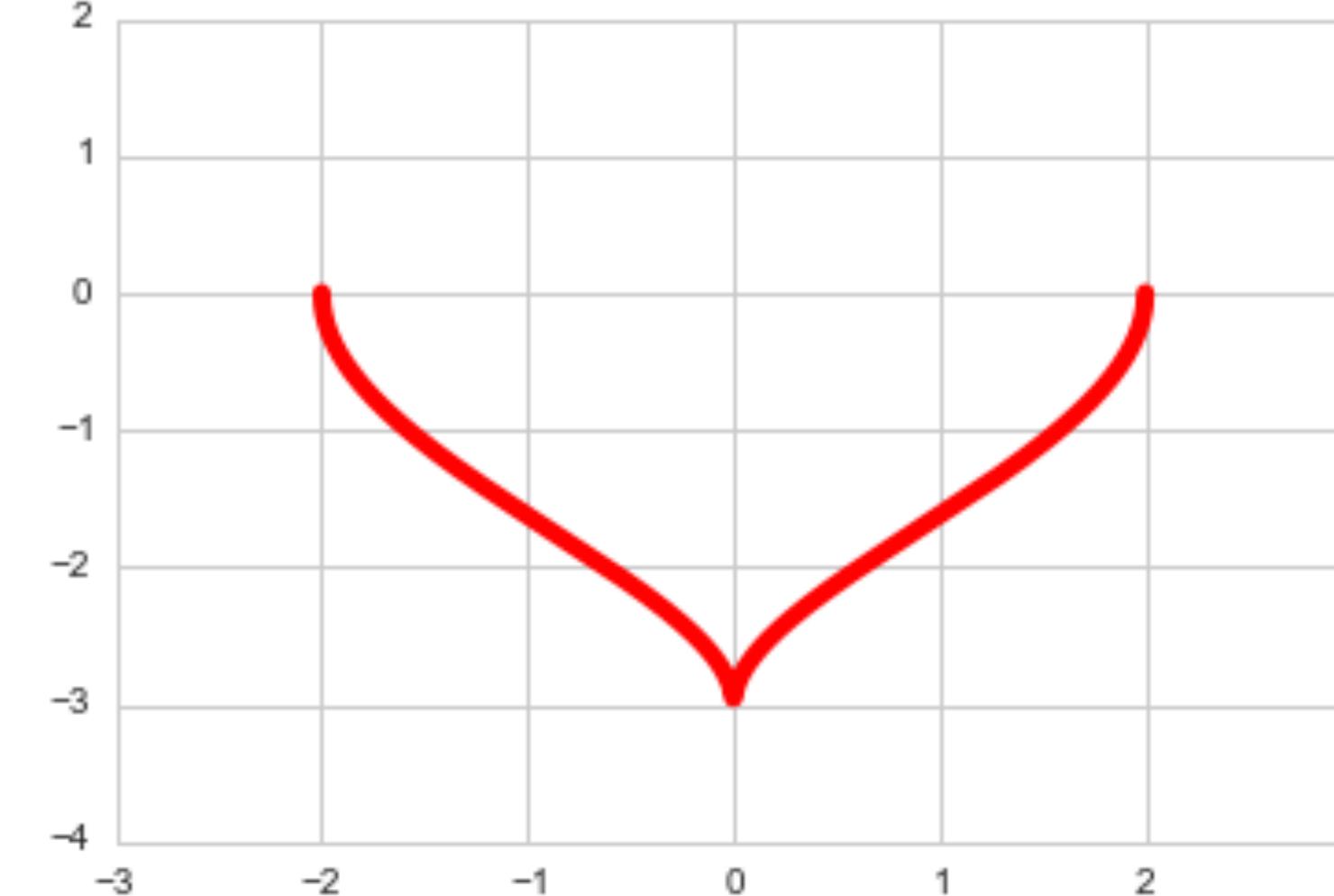
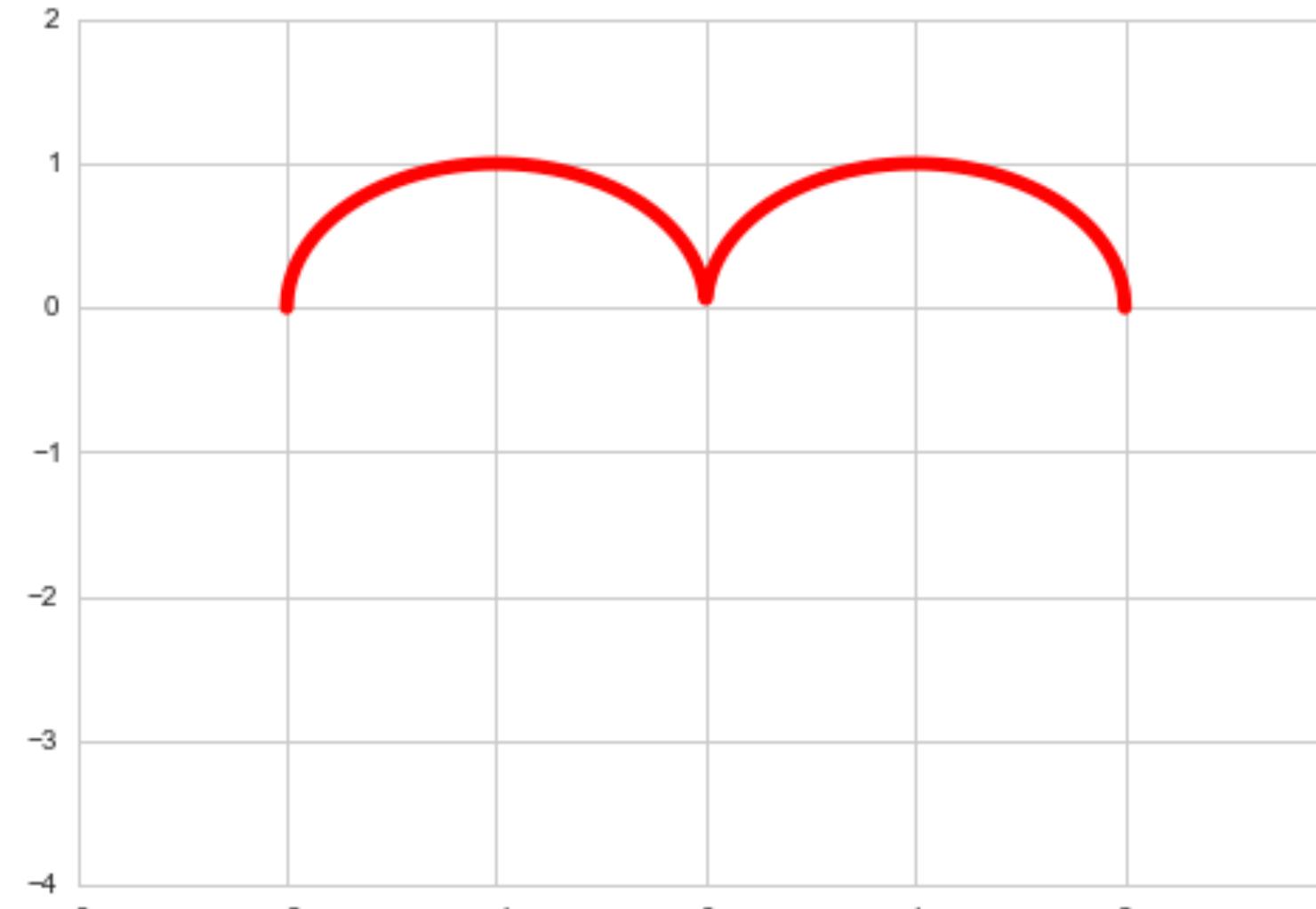
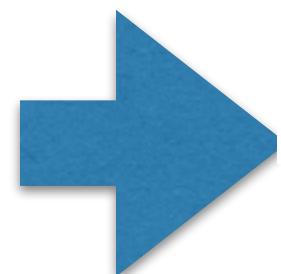
import scipy as sp
import matplotlib.pyplot as plt
import seaborn

x = sp.linspace(-2,2,1000)
y1 = sp.sqrt(1-(abs(x)-1)**2)
y2 = -3*sp.sqrt(1-(abs(x)/2)**0.5)

seaborn.set_style("whitegrid")
heart = plt.plot(x, y1, color="r", linewidth=5)
plt.plot(x, y2, color="r", linewidth=5)
plt.xlabel("Love you", fontsize=25, color="r")

axes = plt.gca()
axes.set_xlim([-3,3])
axes.set_ylim([-4,2])

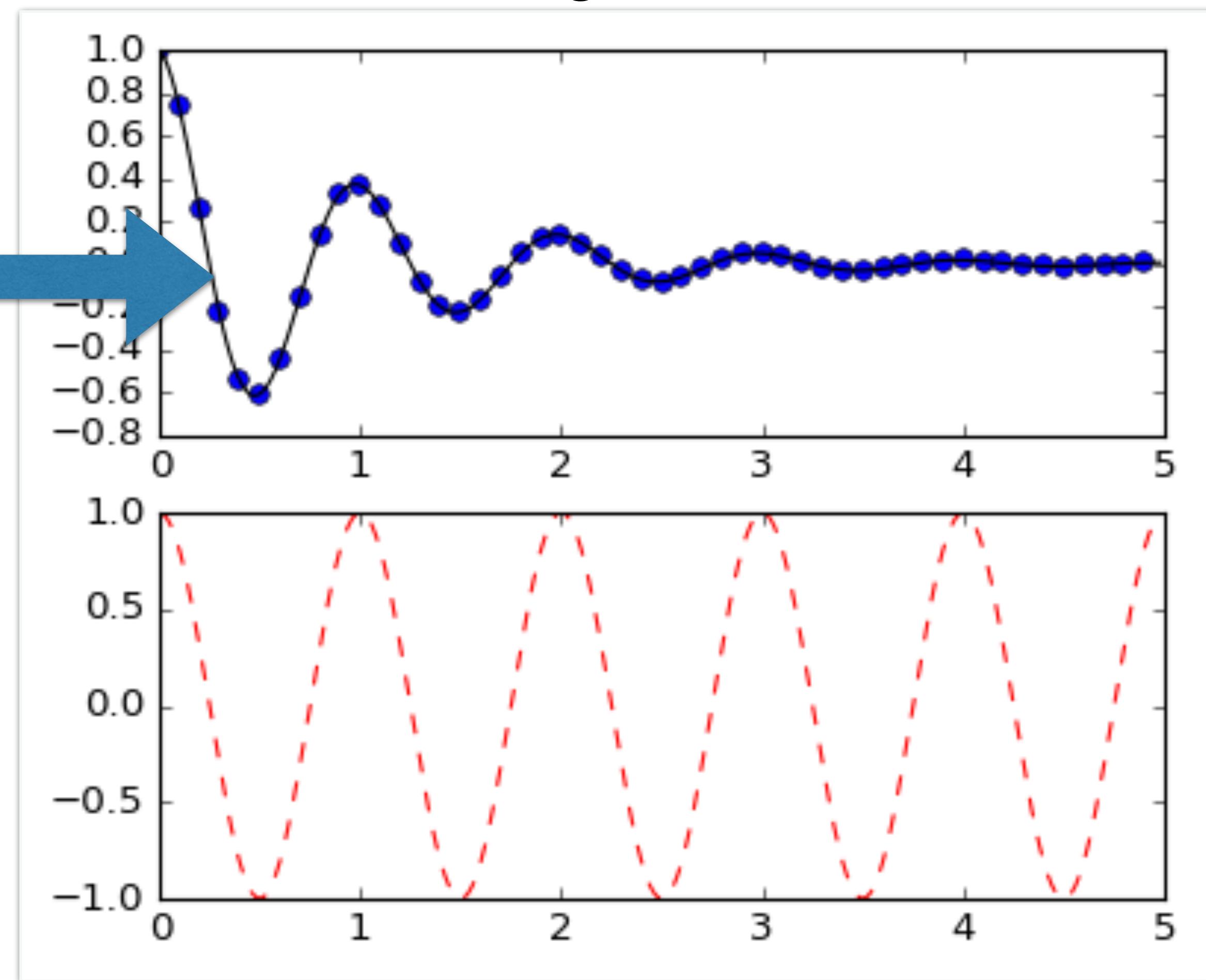
plt.show()
```



Jupyter 상에서 데이터 다루기

Figure1

Line



subplot1

subplot2

Jupyter 상에서 데이터 다루기

```
In [34]: %matplotlib inline

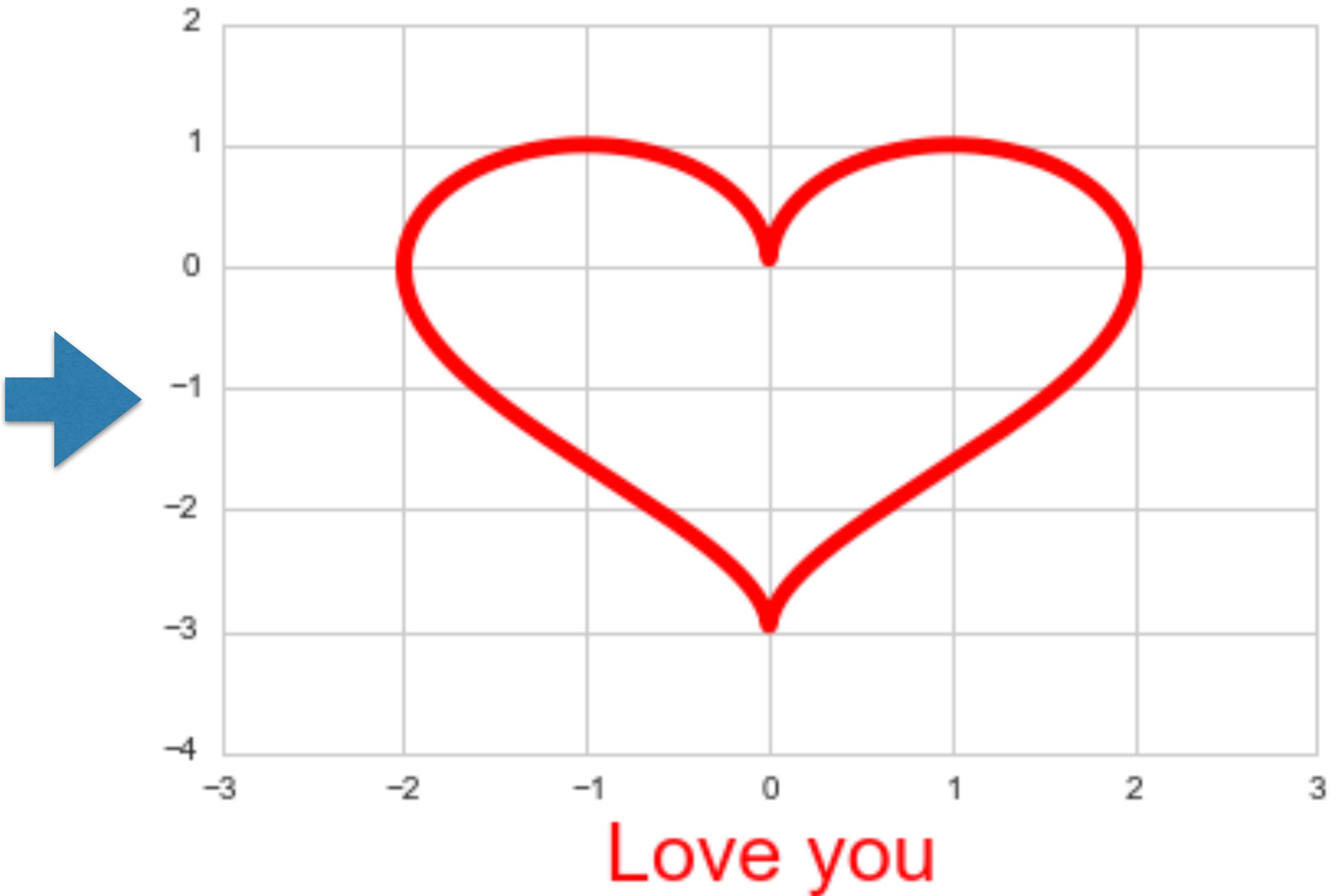
import scipy as sp
import matplotlib.pyplot as plt
import seaborn

x = sp.linspace(-2,2,1000)
y1 = sp.sqrt(1-(abs(x)-1)**2)
y2 = -3*sp.sqrt(1-(abs(x)/2)**0.5)

seaborn.set_style("whitegrid")
heart = plt.plot(x, y1, color="r", linewidth=5)
plt.plot(x, y2, color="r", linewidth=5)
plt.xlabel("Love you", fontsize=25, color="r")

axes = plt.gca()
axes.set_xlim([-3,3])
axes.set_ylim([-4,2])

plt.show()
```





jeffgukang@gmail.com

출처

- 지문 인식 광전식: SUPREMA
- 애플 먹는 안드로이드
<http://www.hdwallper.co/android-eating-apple-cool-free-wallpapers-for-desktop.html/android-eating-apple-cool-free-wallpapers-for-desktop-2>
- Kalman Filter
https://en.wikipedia.org/wiki/Kalman_filter#/media/File:Basic_concept_of_Kalman_filtering.svg
- Right Hands Rule
<https://share.ehs.uen.org/sites/default/files/images/unit5l1handXYZ.png>
- 기압계
<http://jmic.co.kr/index.php?incPage=partnerProduct&page=41&pageSet=24&scrSeq=&cate1=&cate2=&cate3=&searchWord=>
- 에베레스트
<http://extmovie.maxmovie.com/xe/everest/8100410>
- 항문인식
www.xenofreaks.com
- 애플와치 심박센서
<http://watch.camp/2015/05/apple-watch-mio-alpha-wahoo-blue-garmin-polar/>
- pyplot 하트 그래프
<http://stanford.edu/~mwaskom/software/seaborn/>