

# Plotting in Python

Hyunjoong Kim

[soy.lovit@gmail.com](mailto:soy.lovit@gmail.com)

[github.com/lovit](https://github.com/lovit)

- 
- 데이터를 Scatter plot, Bar chart 등과 같은 plot 으로 시각화하기 위한 다양한 라이브러리들이 존재합니다.

- [matplotlib](#), [Bokeh](#), [Seaborn](#), [ggplot](#), etc ...

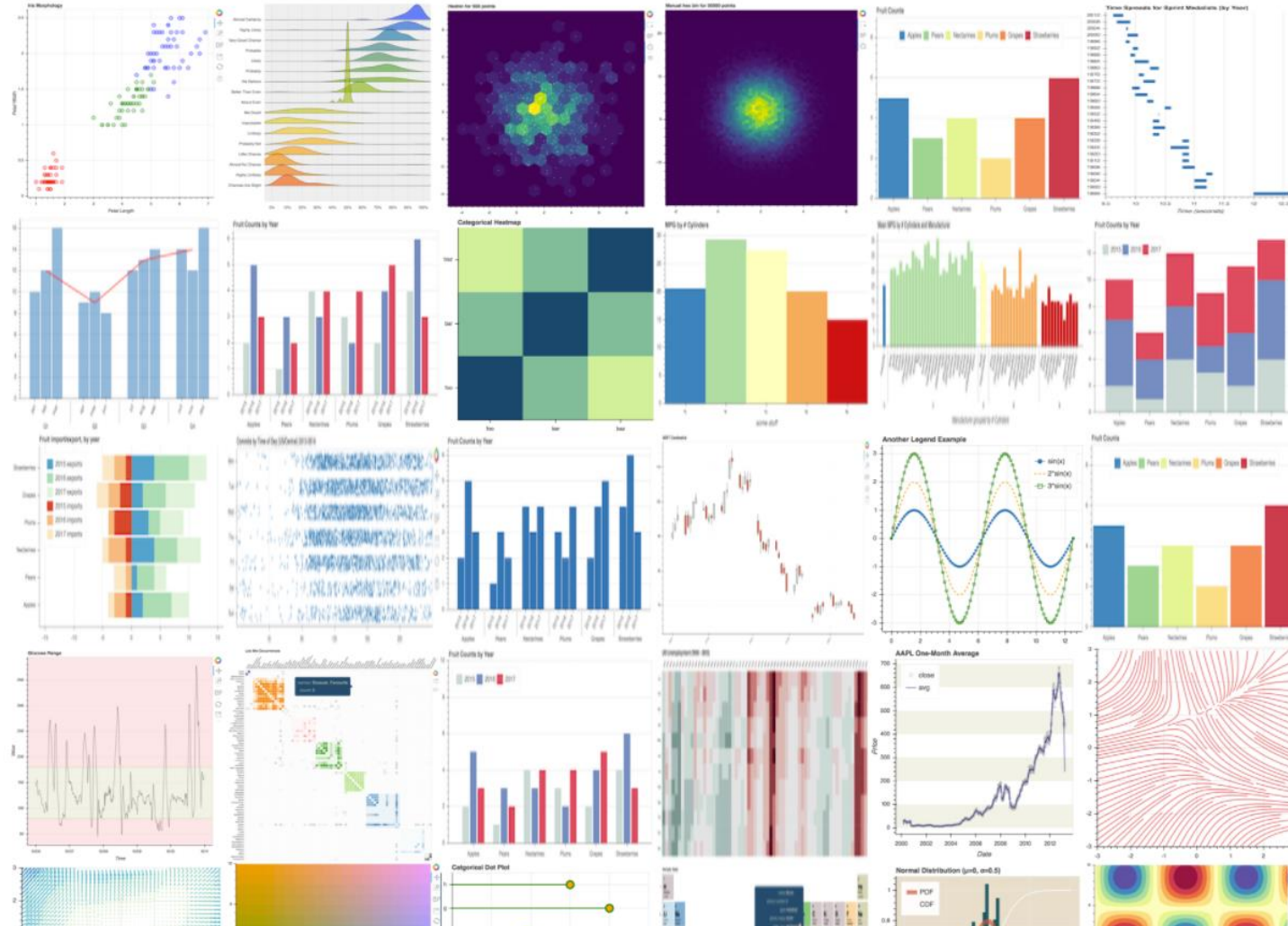
- 
- matplotlib 은 2004 년도부터 사용된 가장 오래된 plotting tool 입니다.
    - 여러 라이브러리들의 예시에 자주 등장합니다.
    - 문법이 복잡합니다.
  - 최근에는 JavaScript 를 이용하는 라이브러리들이 많이 제공되고 있습니다.
    - 최근에 만들어진 plotting tools 은 gallery 를 제공합니다.
    - Gallery 의 demo 만으로도 plotting 을 위한 작업을 할 수 있습니다.

# Bokeh gallery

## Standalone Examples

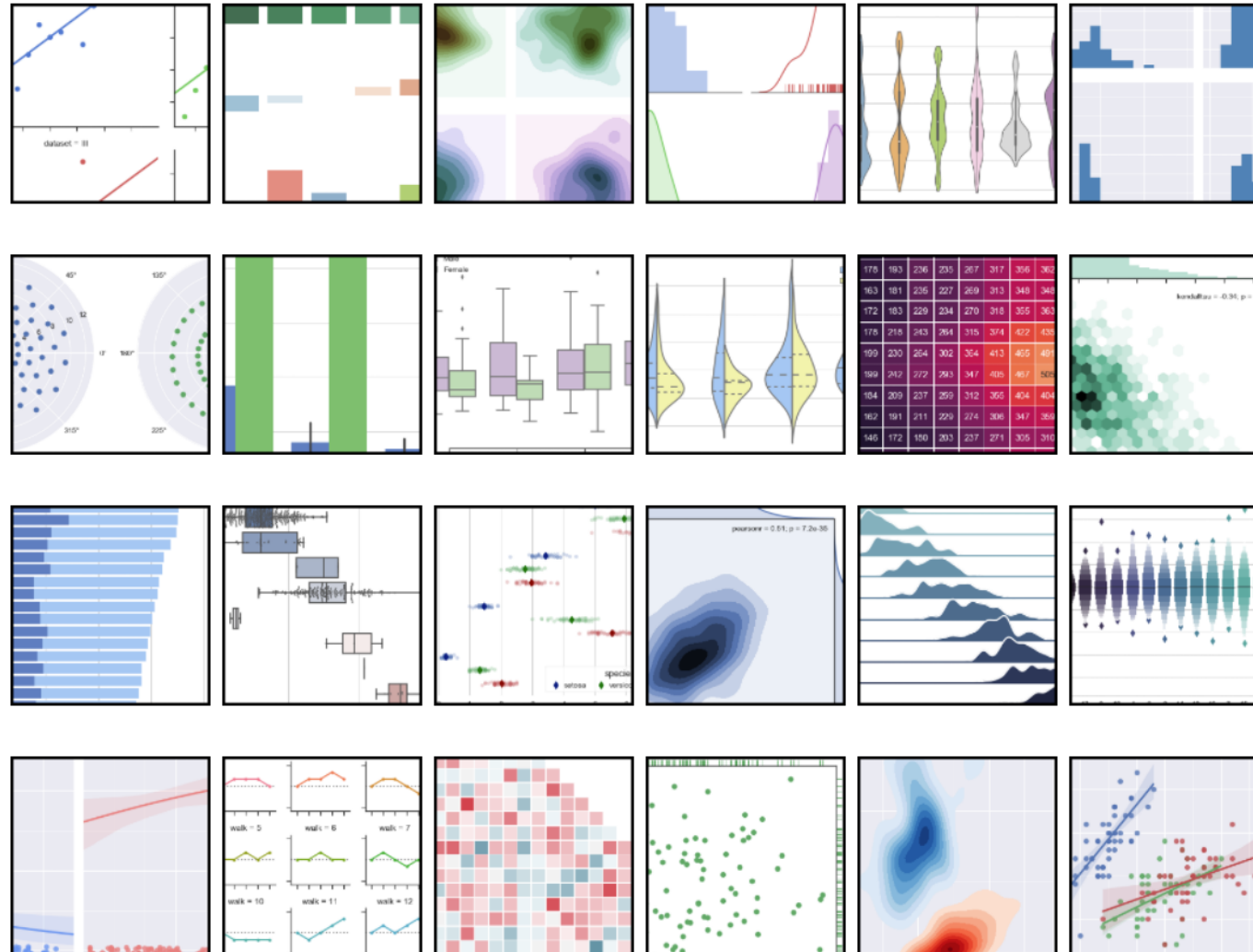
All of the examples below are located in the `examples` subdirectory of your Bokeh checkout. By “standalone” we mean that these examples make no use of the Bokeh server. These plots still have many interactive tools and features, including linked panning and brushing, and hover inspectors.

Click on an image below to see its code and interact with the live plot.



# Seaborn gallery

## Example gallery



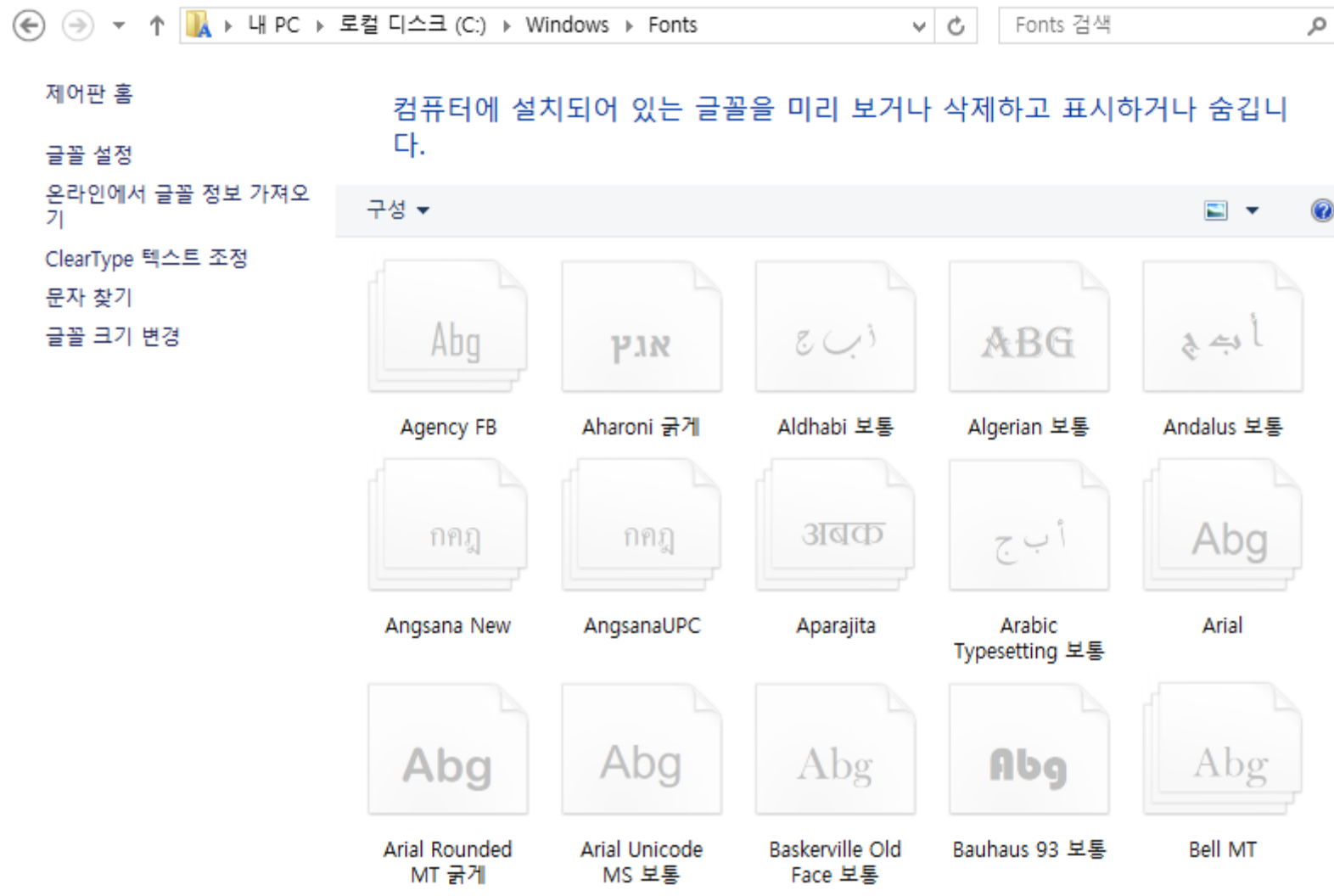
# 한글 폰트

---

- Plotting 과정 중 한글을 이용할 때에는 폰트 설정을 해야합니다.
  - 각 plotting tool 이 이용하는 기본 폰트가 한글을 지원하지 않는다면, 한글 폰트를 지정하면 됩니다.
  - 대부분의 plotting 라이브러리는 font path 를 설정하는 기능을 제공합니다.

# 한글 폰트

- 윈도우 OS 는 c:\windows\fonts 에 폰트가 있습니다.



# 한글 폰트

---

- Linux 에서는 /usr/share/fonts/ 에 폰트가 존재합니다.
- 공개되어 있는 한글 폰트를 다운로드 받은 뒤, 각 OS 의 폰트 폴더에 파일을 복사해 두세요.



# Jupyter notebook 에서 plotting 하기

---

- Jupyter notebook 에서 작업을 할 경우, 어떤 라이브러리들은 notebook output cell 에 plotting 을 출력하기 위한 설정들을 필요로 합니다.
  - 각 라이브러리 별로 jupyter notebook plotting 이라고 검색하면 아래와 같은 설정들을 찾을 수 있습니다.

```
# matplotlib
%matplotlib inline

# Bokeh
from bokeh.plotting import output_notebook
output_notebook()
```

# Scatter plot with matplotlib

---

```
import numpy as np

# generate data
data = np.random.random_sample((5, 2)) * 10

# set x, y
x = data[:, 0]
y = data[:, 1]
```

# Scatter plot with matplotlib

```
%matplotlib inline

import matplotlib

import matplotlib.pyplot as plt

krfont = {'family' : 'nanumgothic',
          'weight' : 'bold', 'size' : 10}

matplotlib.rc('font', **krfont)

# generate figure
fig = plt.figure(figsize=(10,10))
```

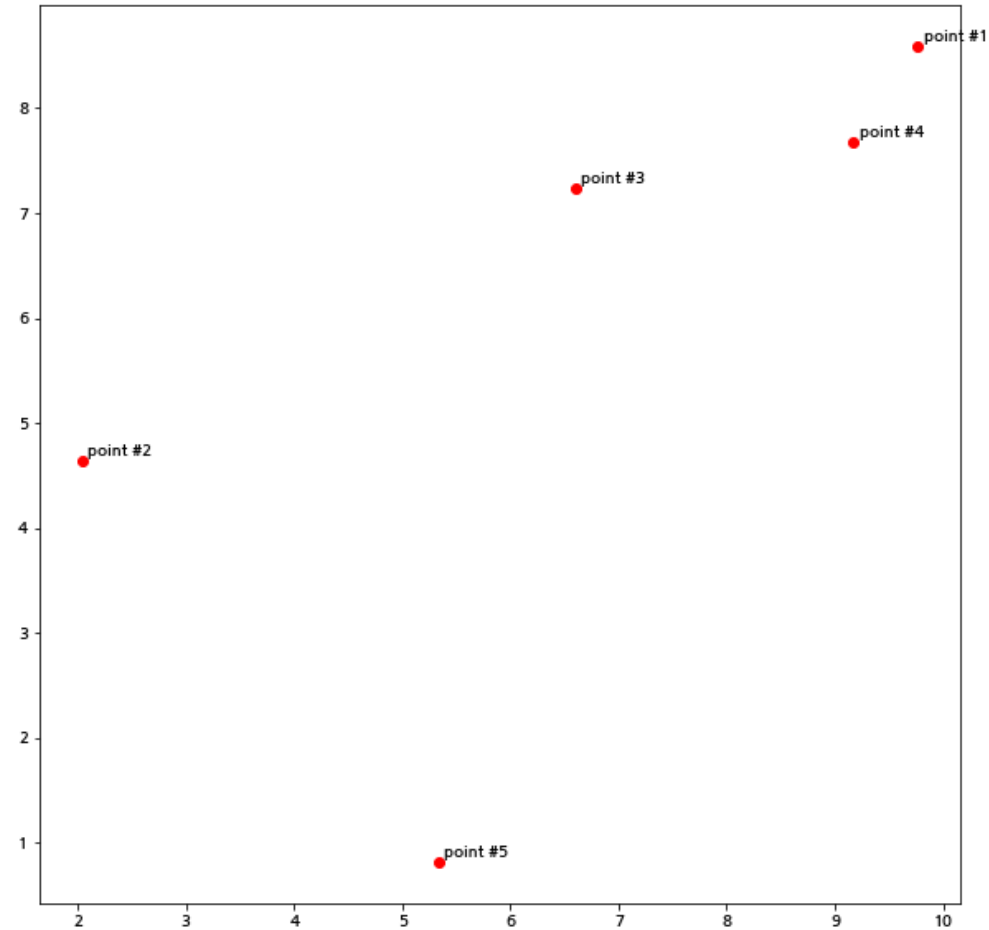
```
# scatter plot
plt.scatter(data[:,0], data[:,1], color='red')

# annotation
for idx in (data.shape[0]):
    x_ = data[idx,0] + 0.05
    y_ = data[idx,1] + 0.05
    text = 'point #{0}'.format(idx+1)
    plt.annotate(text, (x_, y_))

# show
plt.show()
```

# Scatter plot with matplotlib

---



# Bokeh

---

- pip install bokeh
  - Bokeh 는 JavaScript 를 이용하여 html 형식으로 plot 을 그리기 때문에 headless browser 같은 라이브러리들이 필요합니다.
  - 설치는 [공식 홈페이지](#)를 반드시 참고하세요.

# Bokeh

- Bokeh 는 figure 를 만든 뒤,
  - 그 위에 plot 을 그리고,
  - show 를 통하여 출력합니다.
- 
- Jupyter notebook 에서 이용할 경우에는 output\_notebook() 을 먼저 실행해야 합니다.

```
from bokeh.plotting import figure, output_file, output_notebook, show

x = [1, 2, 3, 5, 7]
y = [5, 1, 2, 7, 10]

# output to static HTML file
output_file("lines.html")

# output to notebook
output_notebook()

# create a new plot with a title and axis labels
p = figure(title="Title", x_axis_label='x', y_axis_label='y')

# add a line renderer with legend and line thickness
p.line(x, y, legend="variance", line_width=2)

# show the results
show(p)
```

# Bokeh

---

- JavaScript 를 이용하는 다른 여러 시각화 라이브러리들은 interactive 한 plot 을 그릴 수 있습니다.

# Scatter plot with Bokeh

---

```
from numpy.random import random from bokeh.plotting import figure, show, output_notebook
```

```
def mscatter(p, x, y, marker="circle", size=5,  
            line_color="navy", fill_color="orange", alpha=0.5):  
  
    p.scatter(x, y, marker=marker, size=size,  
             line_color= line_color, fill_color= fill_color, alpha=alpha)
```

```
x, y = [1, 2, 3], [4, 5, 6]
```

```
p = figure(title="Bokeh Markers", toolbar_location=None)  
p.grid.grid_line_color = None  
p.background_fill_color = "white"  
mscatter(p, x, y)
```

```
show(p)
```



# Scatter plot with Bokeh

---

