
파이썬 기반
빅데이터 처리
및 분석기술
Crayon

이번 시간에는

3 다중 서브 플롯

- subplot() 함수 사용
- subplots() 함수 사용
- GridSpec() 함수 사용

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ 서브 플롯

- 서로 다른 그림이나 도표를 나란히 표시할 때 사용함
- 하나의 큰 그림 내부에 있는 작은 그림을 의미함

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.subplot 함수

```
%matplotlib inline

import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
import numpy as np

nrows, ncols = 3, 4

for i in range(nrows * ncols):
    plt.subplot(nrows, ncols, i+1)
    plt.text(0.5, 0.5, str((nrows, ncols, i+1)), fontsize=16, ha='center')
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

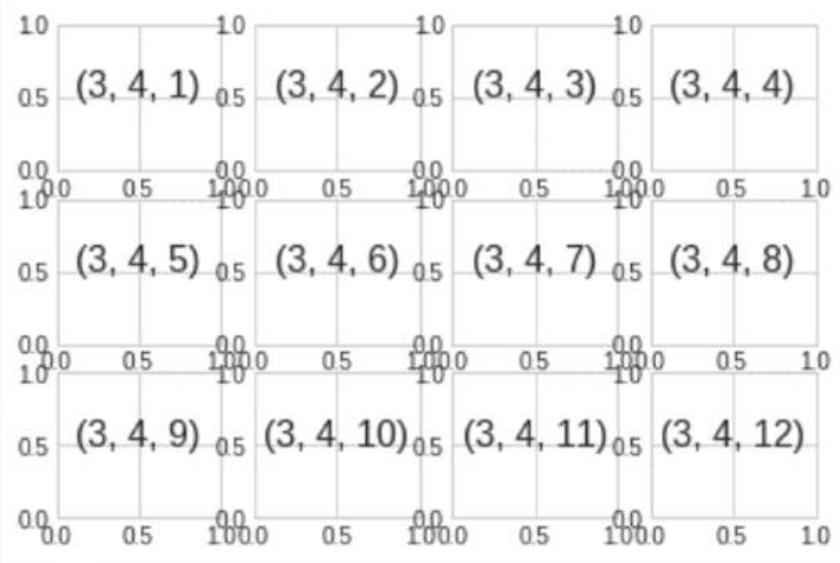
◆ plt.subplot 함수

```
%matplotlib inline

import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
import numpy as np

nrows, ncols = 3, 4

for i in range(nrows * ncols):
    plt.subplot(nrows, ncols, i+1)
    plt.text(0.5, 0.5, str((nrows, ncols, i+1)), fontsize=16, ha='center')
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

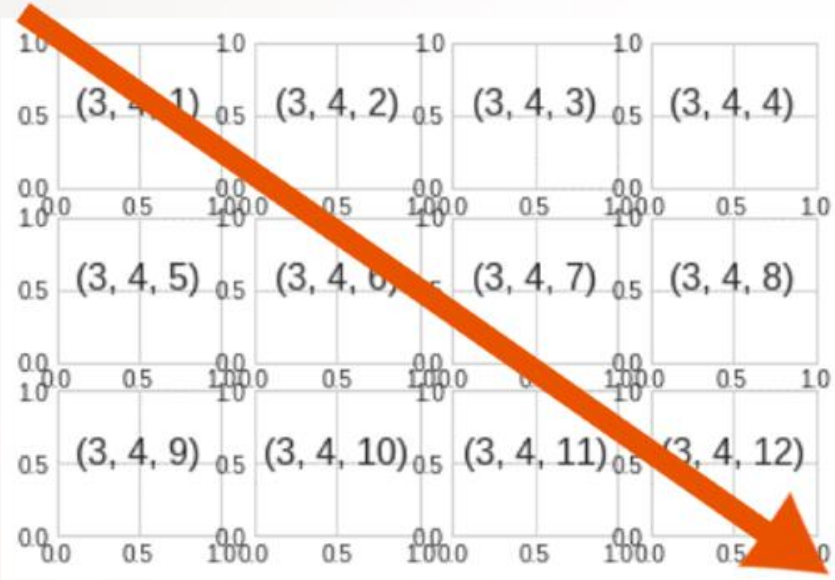
◆ plt.subplot 결과



1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.subplot 결과



너무 촘촘해서 축이 겹치는 문제 발생

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.subplot._adjust 함수

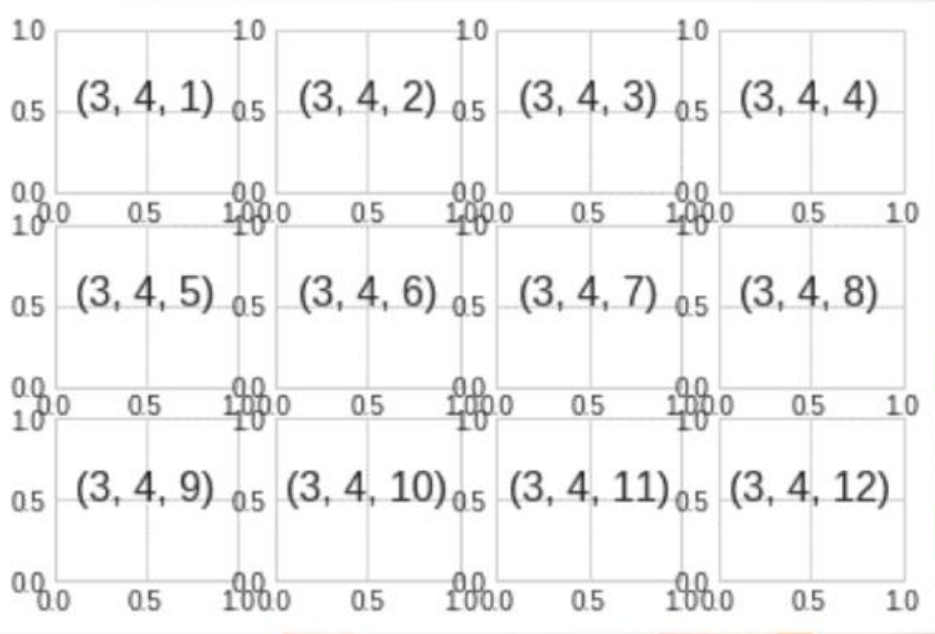
```
fig = plt.figure()
fig.subplots_adjust
(hspace=0.4, wspace=0.4)

nrows, ncols = 3, 4

 for i in range(nrows * ncols):
    plt.subplot(nrows, ncols, i+1)
    plt.text(0.5, 0.5, str((nrows, ncols, i+1)),
    fontsize=16, ha='center')
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.subplot._adjust 함수

```
fig = plt.figure()
fig.subplots_adjust
(hspace=0.4, wspace=0.4)

nrows, ncols = 3, 4

for i in range(nrows * ncols):
    plt.subplot(nrows, ncols, i+1)
    plt.text(0.5, 0.5, str((nrows, ncols, i+1)),
    fontsize=16, ha='center')
```



기본 간격의 40% 확대 설정하여 문제 해결

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.subplots 함수

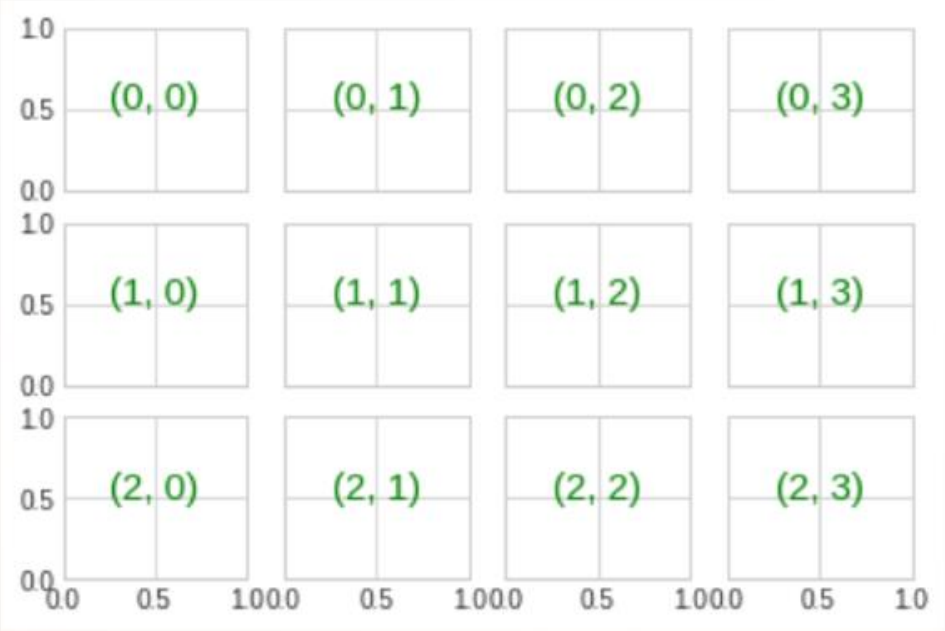
```
nrows, ncols = 3, 4

fig, ax = plt.subplots(nrows, ncols, sharex='col', sharey='row')

for i in range(nrows):
  for j in range(ncols):
    ax[i, j].text(0.5, 0.5, str((i, j)), fontsize=15, ha='center', color='g');
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.subplots 결과

- plt.subplot, plt.subplots : 격자형 다중 서브 플롯



1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
nrows, ncols = 3, 4

grid = plt.GridSpec(nro, 4, wspace=0.4, hspace=0.4)

ax1 = plt.subplot(grid[0, 0])
ax2 = plt.subplot(grid[0, 1:3])
ax3 = plt.subplot(grid[1:, :2])
ax4 = plt.subplot(grid[1:, 2])
ax5 = plt.subplot(grid[:, 3])
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
nrows, ncols = 3, 4

grid = plt.GridSpec(nro, 4, wspace=0.4, hspace=0.4)

ax1 = plt.subplot(grid[0, 0])
ax2 = plt.subplot(grid[0, 1:3])
ax3 = plt.subplot(grid[1:, :2])
ax4 = plt.subplot(grid[1:, 2])
ax5 = plt.subplot(grid[:, 3])
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

첫 번째, 두 번째 인수 : 텍스트가 시작하는 위치 설정

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

세 번째 인수 : 텍스트의 내용

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

네 번째 인수 : 텍스트의 크기

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

다섯 번째 인수 : 가로 방향으로 가운데 정렬

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 함수

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

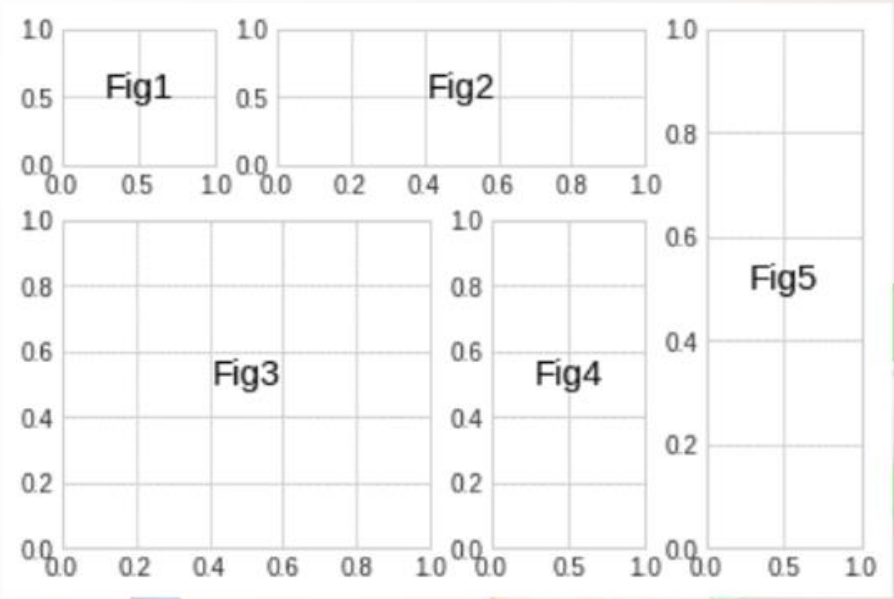
여섯 번째 인수 : 텍스트의 색상

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 결과

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

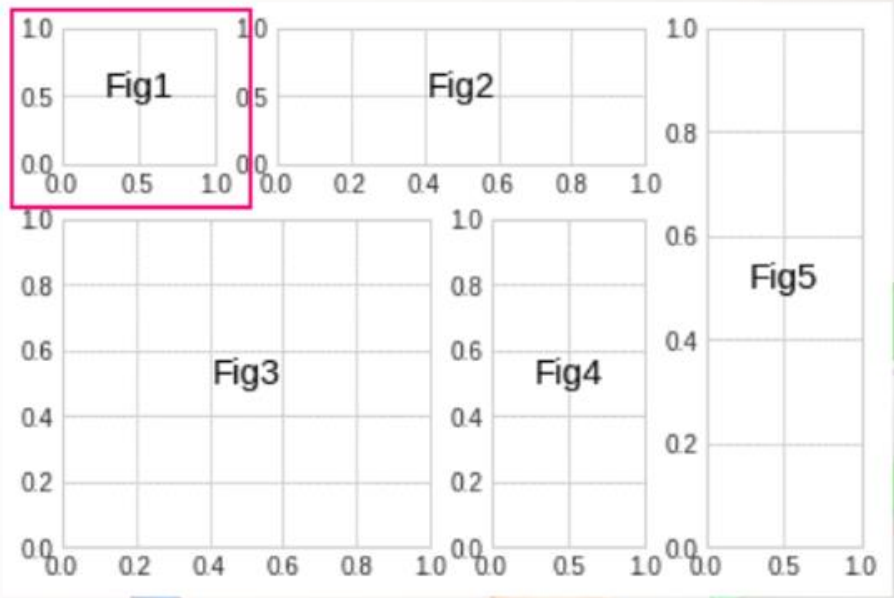


1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 결과

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

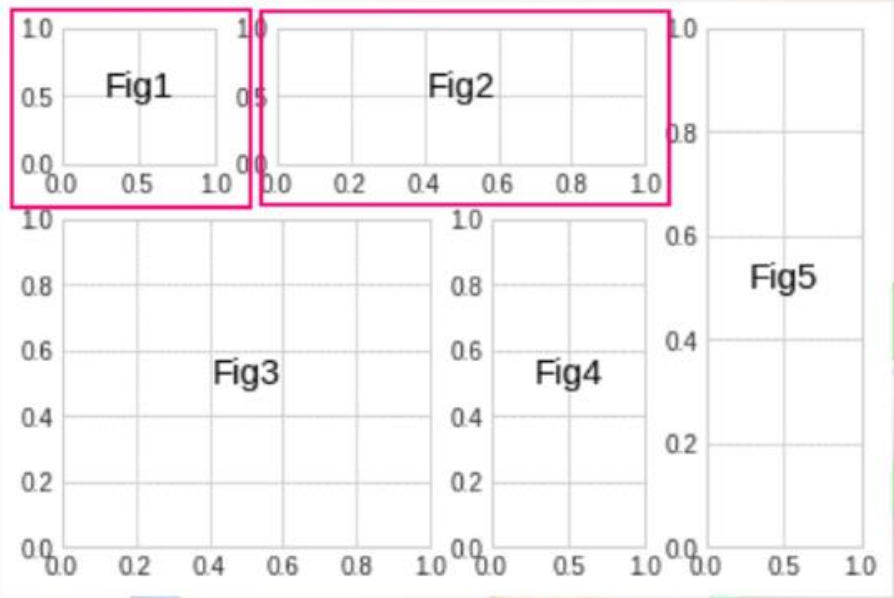


1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 결과

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

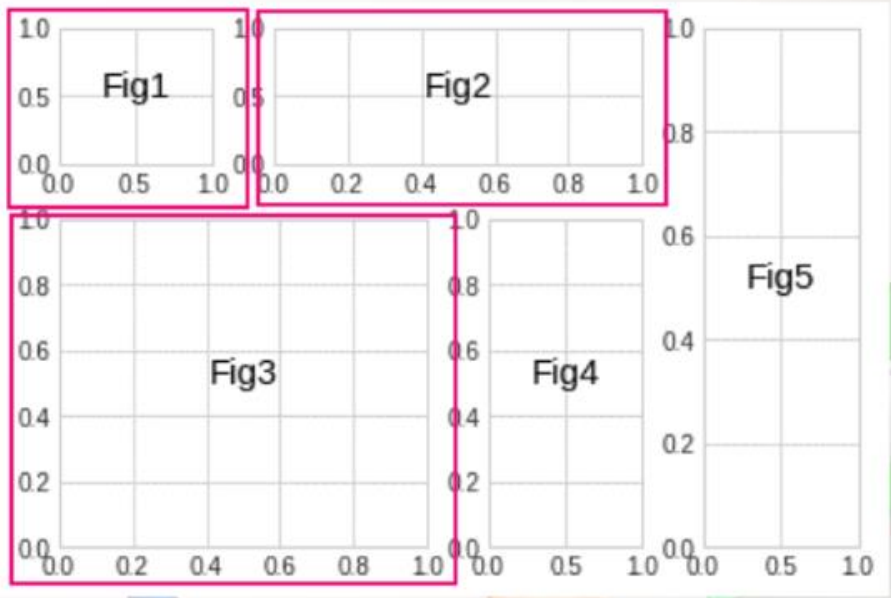


1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 결과

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```

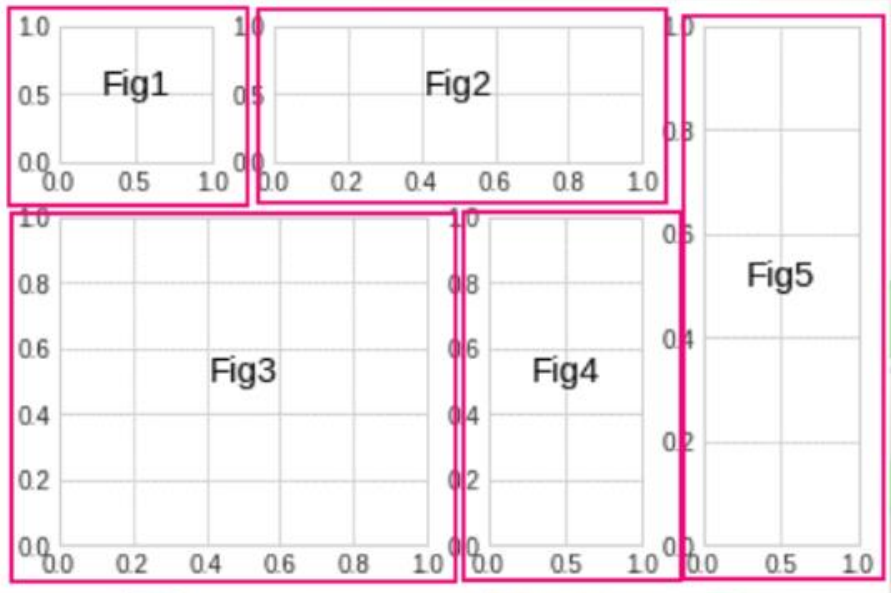


1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 결과

```
ax1.text(0.5, 0.5, 'Fig1', fontsize=15, ha='center', color='k')
ax2.text(0.5, 0.5, 'Fig2', fontsize=15, ha='center', color='k')
ax3.text(0.5, 0.5, 'Fig3', fontsize=15, ha='center', color='k')
ax4.text(0.5, 0.5, 'Fig4', fontsize=15, ha='center', color='k')
ax5.text(0.5, 0.5, 'Fig5', fontsize=15, ha='center', color='k')
```



1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 활용 사례

정규 분포 데이터 만들기

```
mean = [0, 0]
cov = [[1, 1], [1, 2]]
x, y = np.random.multivariate_normal(mean, cov, 5000).T
```

GridSpec으로 축 설정하기

```
fig = plt.figure(figsize=(6, 6))
grid = plt.GridSpec(4, 4, hspace=0.2, wspace=0.2)
main_ax = fig.add_subplot(grid[:-1, 1:])
y_hist = fig.add_subplot(grid[:-1, 0], xticklabels=[], sharey=main_ax)
x_hist = fig.add_subplot(grid[-1, 1:], yticklabels=[], sharex=main_ax)
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 활용 사례

메인 축에 점 산포하기

```
main_ax.plot(x, y, 'ok', markersize=3, alpha=0.2)
```

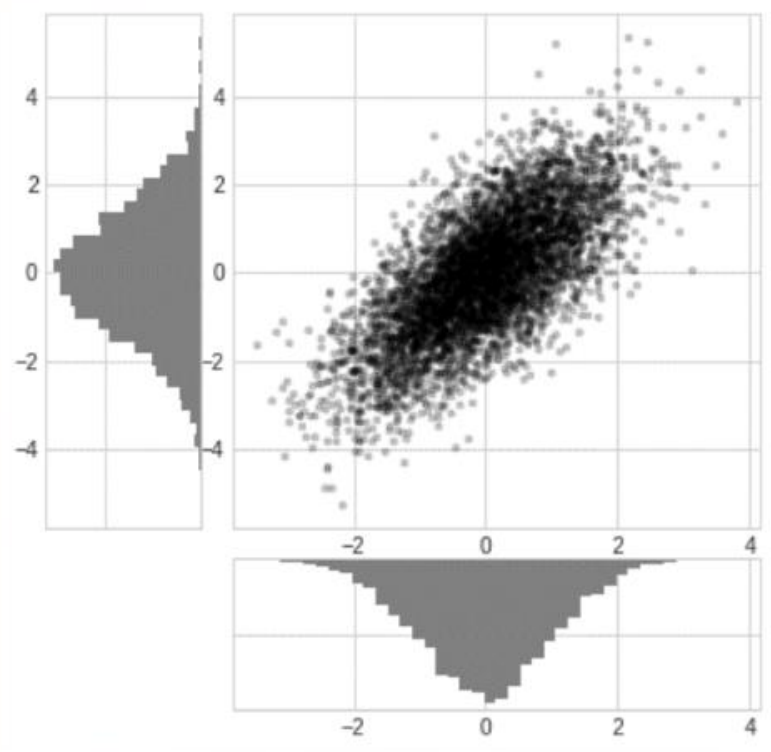
보조 축 상에 히스토그램 만들기

```
x_hist.hist(x, 40, histtype='stepfilled', orientation='vertical', color='gray')
x_hist.invert_yaxis()

y_hist.hist(y, 40, histtype='stepfilled', orientation='horizontal', color='gray')
y_hist.invert_xaxis()
```

1. Matplotlib 사용법

1) 다중 서브 플롯 작성

◆ plt.GridSpec 활용 사례



이번 시간에는

3 다중 서브 플롯

- subplot() 함수 사용
- subplots() 함수 사용
- GridSpec() 함수 사용

이번 시간에는

실습 참고 자료

- **Colab 노트북 파일**
- **Matplotlib 공식 사이트**
  - https://matplotlib.org/tutorials/index.html

이번 시간에는

과제 안내

- 과제 : 퀴즈
- 제출 방법 : 과제 게시판 제출 방법 안내 참조

질의 응답 게시판

- 학습 내용, 퀴즈, 과제 등에 대한 질의응답 게시판을 통한 질의응답

다음 시간에는

6 머신러닝 기법

- 머신러닝의 개념과 기본 절차
- Scikit-Learn API 사용법
- 회귀분석 심화