Homework 06

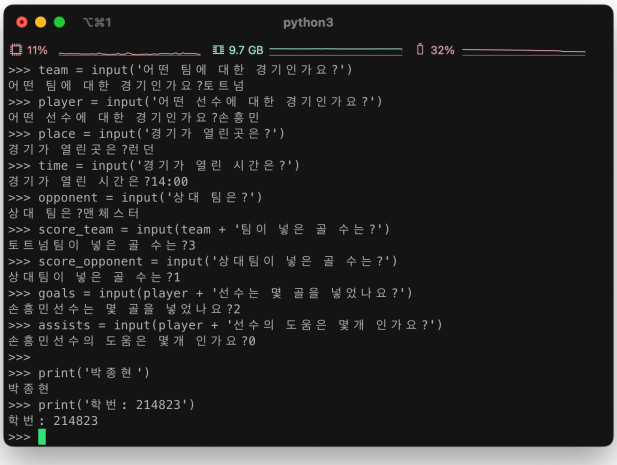
이름: 박종현
학번: 214823

Caution

1. 수업 자료에서 사용하는 라이브러리의 버전과 실행 환경에서 사용하는 라이브러리의 버전이 달라, 자료의 코드를 조금 수정하여 대응하였습니다.
   - 익명 Q&A 게시판 21번 게시물, “과제 6 수행 중 발생하는 문제 및 해결책 정리: keras_model.h5, Image.ANTIALIAS, df.append 관련” https://sel.jnu.ac.kr/mod/ubboard/article.php?id=1134310&bwid=700874
2. 스크린샷에 이름과 학번이 포함되어야한다는 조건을 충족하기 위해, 터미널 타이틀 바에 이름과 학번이 나오도록 설정하였습니다. (2번 과제부터 적용)

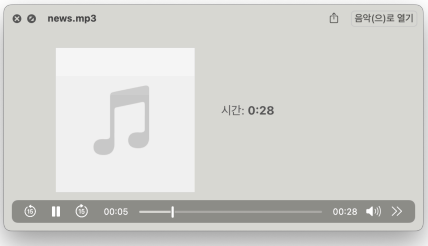
Content

[1] 12-1. 5 10 쪽



```
⌥⌘1                    python3
12%        9.8 GB        32%
>>> news = '\n[프리미어 리그 속보]\n'
>>> news += ' ' + player + ' 선수는 ' + place + '에서 ' + time + '에 열린 경기에
 출전하였습니다.'
>>> news += '상대 팀은 ' + opponent + '였습니다.'
>>>
>>> news += player + ' 선수의 팀이 ' + score_team + '골을 넣어'
>>> news += score_opponent + '골을 넣은'
>>>
>>> _score_team = int(score_team)
>>> _score_opponent = int(score_opponent)
>>>
>>> if _score_team > _score_opponent:
...     news += '상대팀을 이겼습니다.'
... elif _score_team < _score_opponent:
...     news += '상대팀에게 졌습니다.'
... else:
...     news += '상대팀과 비겼습니다.'
...
>>> _goals = int(goals)
>>> _assists = int(assists)
>>>
>>> if _goals > 0 and _assists > 0:
...     news += player + ' 선수는 ' + goals + ' 골에 ' + assists + ' 도움으로 크
게 활약하였습니다.'
... elif _goals > 0 and _assists == 0:
...     news += player + ' 선수는 ' + goals + ' 골을 넣었습니다.'
... elif _goals == 0 and _assists > 0:
...     news += player + ' 선수는 골은 없지만 ' + assists + ' 도움으로 팀에 공헌
하였습니다.'
... else:
...     news += '아쉽게도 이번 경기에서 ' + player + '의 발끝은 침묵을 지켰습니
다.'
...
>>> print(news)

[프리미어 리그 속보]
 손흥민 선수는 런던에서 14:00에 열린 경기에 출전하였습니다.상대 팀은 맨체스터였
습니다.손흥민 선수의 팀이 3골을 넣어1골을 넣은상대팀을 이겼습니다.손흥민 선수는
2 골을 넣었습니다.
>>> print('박종현')
박종현
>>> print('학번 : 214823')
학번 : 214823
>>>
```

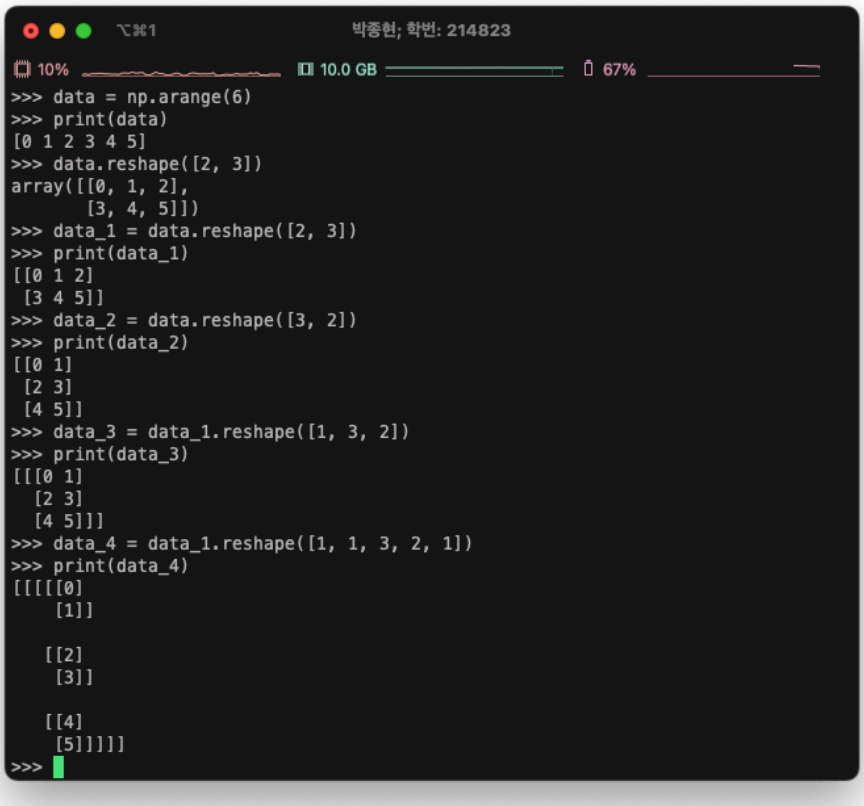


[2] 12-1. 27 29 쪽

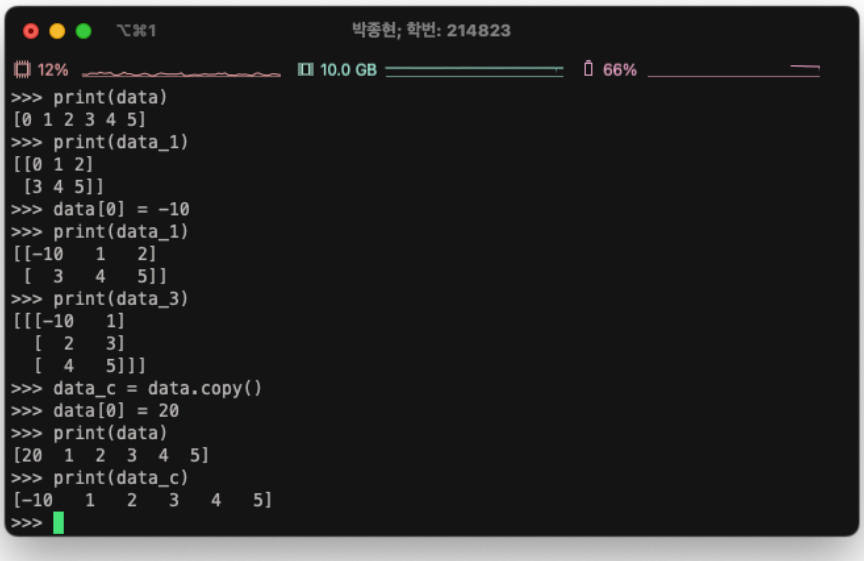
```
⌥⌘1                    박종현; 학번: 214823
14%        9.9 GB        67%
>>> from keras.models import load_model
>>> from PIL import Image, ImageOps
>>> import numpy as np
>>>
>>> model = load_model('keras_model.h5')
WARNING:tensorflow:No training configuration found in the save file, so the mode
l was *not* compiled. Compile it manually.
>>> model.compile()
>>>
>>> data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
>>> image = Image.open('고양이.jpg')
>>>
>>> size = (224, 224)
>>> image = ImageOps.fit(image, size, Image.LANCZOS)
>>>
>>> image_array = np.asarray(image)
>>> normalized_image_array = (image_array.astype(np.float32) / 127.) - 1
>>> data[0] = normalized_image_array
>>>
>>> prediction = model.predict(data)
1/1 [==============================] - 0s 265ms/step
>>> print(prediction)
[[9.9993551e-01 5.2111591e-05 1.2365789e-05]]
>>>
```

```
⌥⌘1                    박종현; 학번: 214823
9%         10.0 GB       67%
>>> image = Image.open('노란텀블러.jpg')
>>> image = ImageOps.fit(image,size, Image.LANCZOS)
>>> image_array = np.asarray(image)
>>> normalized_image_array = (image_array.astype(np.float32) / 127.) - 1
>>> data[0] = normalized_image_array
>>> prediction = model.predict(data)
1/1 [==============================] - 0s 19ms/step
>>> print(prediction)
[[1.4815846e-04 7.7639292e-05 9.9977416e-01]]
>>>
>>> image = Image.open('녹색텀블러.jpg')
>>> image = ImageOps.fit(image,size, Image.LANCZOS)
>>> image_array = np.asarray(image)
>>> normalized_image_array = (image_array.astype(np.float32) / 127.) - 1
>>> data[0] = normalized_image_array
>>> prediction = model.predict(data)
1/1 [==============================] - 0s 18ms/step
>>> print(prediction)
[[4.9207614e-05 9.9989891e-01 5.1825606e-05]]
>>>
```

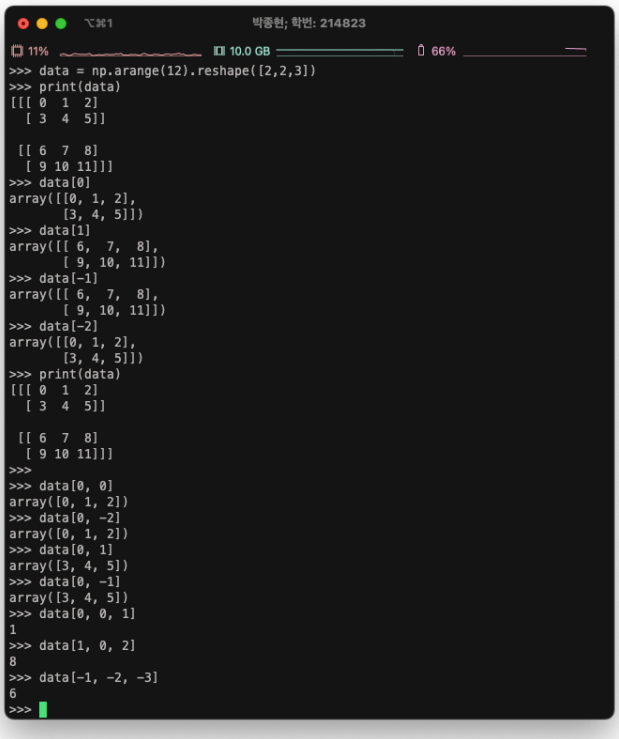
[3] 12-3. 14 쪽



[4] 12-3. 16 쪽

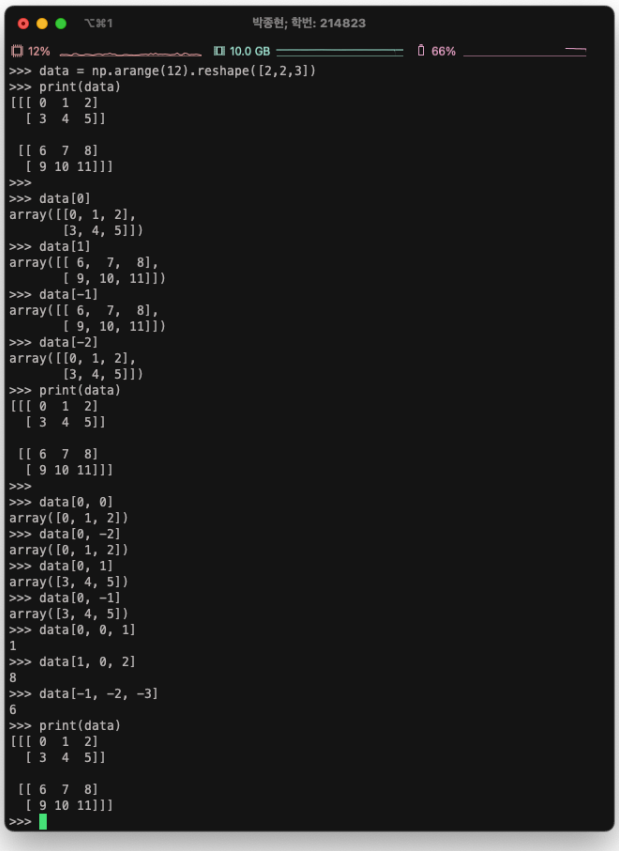


[5] 12-3. 18 20 쪽


박종현; 학번: 214823
11%
10.0 GB
66%
>>> data = np.arange(12).reshape([2,2,3])
>>> print(data)
[[[0 1 2]
  [3 4 5]]

 [[6 7 8]
  [9 10 11]]]
>>> data[0]
array([[0, 1, 2],
       [3, 4, 5]])
>>> data[1]
array([[6, 7, 8],
       [9, 10, 11]])
>>> data[-1]
array([[6, 7, 8],
       [9, 10, 11]])
>>> data[-2]
array([[0, 1, 2],
       [3, 4, 5]])
>>> print(data)
[[[0 1 2]
  [3 4 5]]

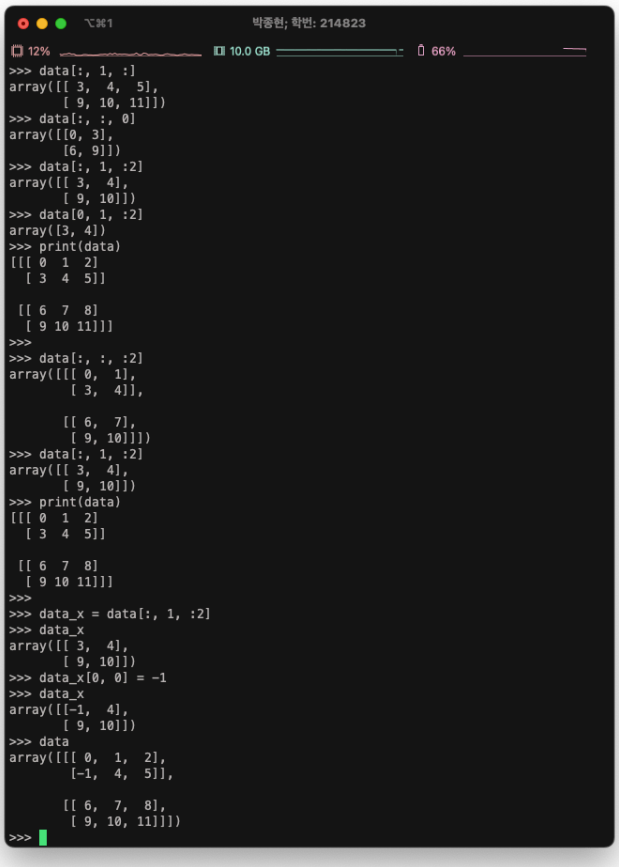
 [[6 7 8]
  [9 10 11]]]
>>>
>>> data[0, 0]
array([0, 1, 2])
>>> data[0, -2]
array([0, 1, 2])
>>> data[0, 1]
array([3, 4, 5])
>>> data[0, -1]
array([3, 4, 5])
>>> data[0, 0, 1]
1
>>> data[1, 0, 2]
8
>>> data[-1, -2, -3]
6
>>>


박종현; 학번: 214823
12%
10.0 GB
66%
>>> data = np.arange(12).reshape([2,2,3])
>>> print(data)
[[[0 1 2]
  [3 4 5]]

 [[6 7 8]
  [9 10 11]]]
>>>
>>> data[0]
array([[0, 1, 2],
       [3, 4, 5]])
>>> data[1]
array([[6, 7, 8],
       [9, 10, 11]])
>>> data[-1]
array([[6, 7, 8],
       [9, 10, 11]])
>>> data[-2]
array([[0, 1, 2],
       [3, 4, 5]])
>>> print(data)
[[[0 1 2]
  [3 4 5]]

 [[6 7 8]
  [9 10 11]]]
>>>
>>> data[0, 0]
array([0, 1, 2])
>>> data[0, -2]
array([0, 1, 2])
>>> data[0, 1]
array([3, 4, 5])
>>> data[0, -1]
array([3, 4, 5])
>>> data[0, 0, 1]
1
>>> data[1, 0, 2]
8
>>> data[-1, -2, -3]
6
>>> print(data)
[[[0 1 2]
  [3 4 5]]

 [[6 7 8]
  [9 10 11]]]
>>>

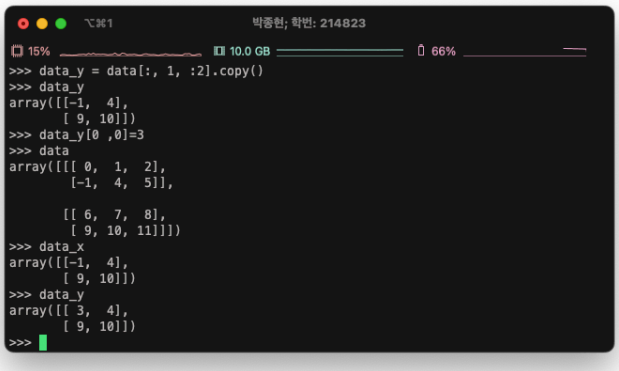

박종현; 학번: 214823
12%
10.0 GB
66%
>>> data[:, 1, :]
array([[3, 4, 5],
       [9, 10, 11]])
>>> data[:, :, 0]
array([[0, 3],
       [6, 9]])
>>> data[:, 1, :2]
array([[3, 4],
       [9, 10]])
>>> data[0, 1, :2]
array([3, 4])
>>> print(data)
[[[0 1 2]
  [3 4 5]]

 [[6 7 8]
  [9 10 11]]]
>>>
>>> data[:, :, :2]
array([[[0, 1],
        [3, 4]],

       [[6, 7],
        [9, 10]]])
>>> data[:, 1, :2]
array([[3, 4],
       [9, 10]])
>>> print(data)
[[[0 1 2]
  [3 4 5]]

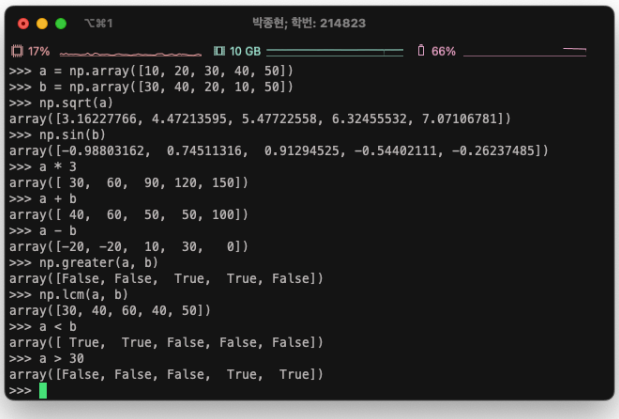
 [[6 7 8]
  [9 10 11]]]
>>>
>>> data_x = data[:, 1, :2]
>>> data_x
array([[3, 4],
       [9, 10]])
>>> data_x[0, 0] = -1
>>> data_x
array([[-1, 4],
       [9, 10]])
>>> data
array([[[0, 1, 2],
        [-1, 4, 5]],

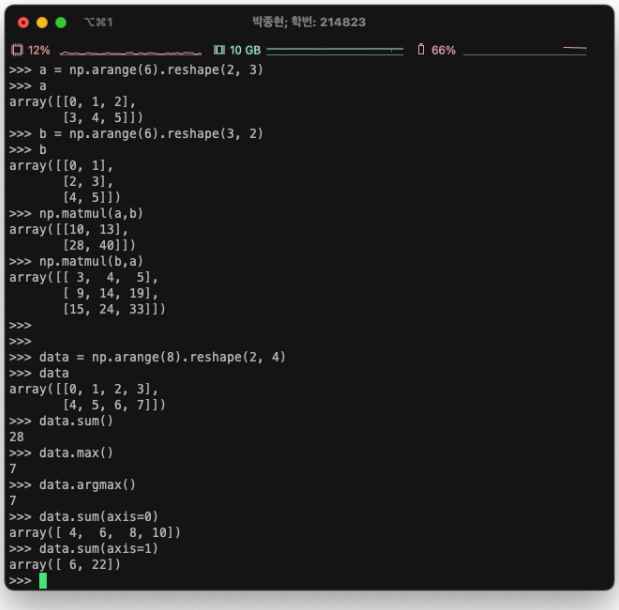
       [[6, 7, 8],
        [9, 10, 11]]])
>>>

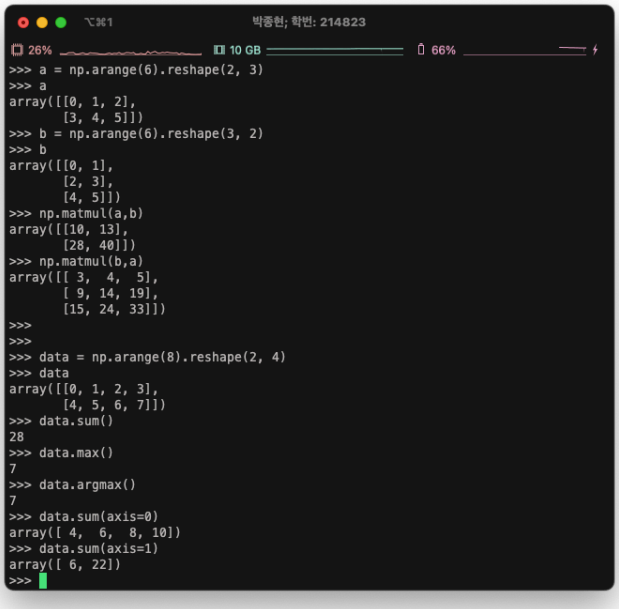

박종현; 학번: 214823
15%
10.0 GB
66%
>>> data_y = data[:, 1, :2].copy()
>>> data_y
array([[-1, 4],
       [9, 10]])
>>> data_y[0 ,0]=3
>>> data
array([[[0, 1, 2],
        [-1, 4, 5]],

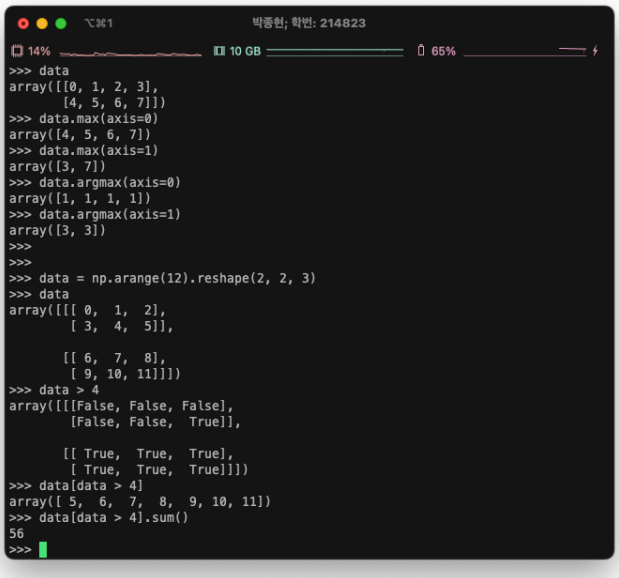
       [[6, 7, 8],
        [9, 10, 11]]])
>>> data_x
array([[-1, 4],
       [9, 10]])
>>> data_y
array([[3, 4],
       [9, 10]])
>>>

[6] 12-3. 24 28 쪽


박종현; 학번: 214823
17%
10 GB
66%
>>> a = np.array([10, 20, 30, 40, 50])
>>> b = np.array([30, 40, 20, 10, 50])
>>> np.sqrt(a)
array([3.16227766, 4.47213595, 5.47722558, 6.32455532, 7.07106781])
>>> np.sin(b)
array([-0.98803162, 0.74511316, 0.91294525, -0.54402111, -0.26237485])
>>> a * 3
array([30, 60, 90, 120, 150])
>>> a + b
array([40, 60, 50, 50, 100])
>>> a - b
array([-20, -20, 10, 30,   0])
>>> np.greater(a, b)
array([False, False, True, True, False])
>>> np.lcm(a, b)
array([30, 40, 60, 40, 50])
>>> a < b
array([True, True, False, False, False])
>>> a > 30
array([False, False, False, True, True])
>>>


박종현; 학번: 214823
12%
10 GB
66%
>>> a = np.arange(6).reshape(2, 3)
>>> a
array([[0, 1, 2],
       [3, 4, 5]])
>>> b = np.arange(6).reshape(3, 2)
>>> b
array([[0, 1],
       [2, 3],
       [4, 5]])
>>> np.matmul(a,b)
array([[10, 13],
       [28, 40]])
>>> np.matmul(b,a)
array([[3, 4, 5],
       [9, 14, 19],
       [15, 24, 33]])
>>>
>>>
>>> data = np.arange(8).reshape(2, 4)
>>> data
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
>>> data.sum()
28
>>> data.max()
7
>>> data.argmax()
7
>>> data.sum(axis=0)
array([4, 6, 8, 10])
>>> data.sum(axis=1)
array([6, 22])
>>>

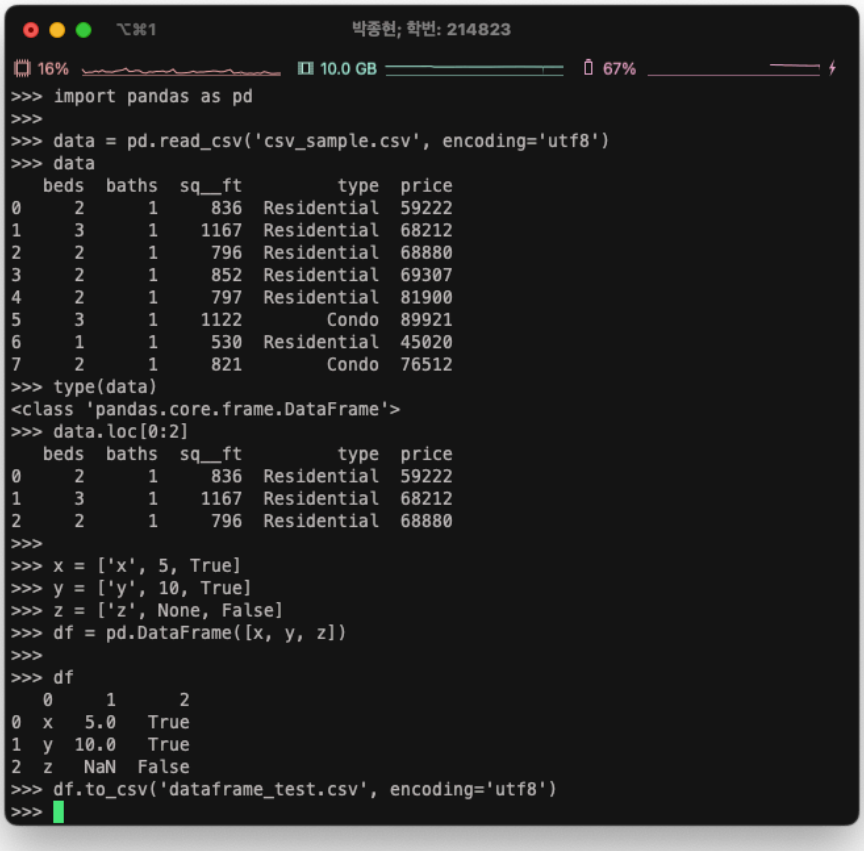

박종현; 학번: 214823
26%
10 GB
66%
>>> a = np.arange(6).reshape(2, 3)
>>> a
array([[0, 1, 2],
       [3, 4, 5]])
>>> b = np.arange(6).reshape(3, 2)
>>> b
array([[0, 1],
       [2, 3],
       [4, 5]])
>>> np.matmul(a,b)
array([[10, 13],
       [28, 40]])
>>> np.matmul(b,a)
array([[3, 4, 5],
       [9, 14, 19],
       [15, 24, 33]])
>>>
>>>
>>> data = np.arange(8).reshape(2, 4)
>>> data
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
>>> data.sum()
28
>>> data.max()
7
>>> data.argmax()
7
>>> data.sum(axis=0)
array([4, 6, 8, 10])
>>> data.sum(axis=1)
array([6, 22])
>>>


박종현; 학번: 214823
14%
10 GB
65%
>>> data
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
>>> data.max(axis=0)
array([4, 5, 6, 7])
>>> data.max(axis=1)
array([3, 7])
>>> data.argmax(axis=0)
array([1, 1, 1, 1])
>>> data.argmax(axis=1)
array([3, 3])
>>>
>>>
>>> data = np.arange(12).reshape(2, 2, 3)
>>> data
array([[[0, 1, 2],
        [3, 4, 5]],

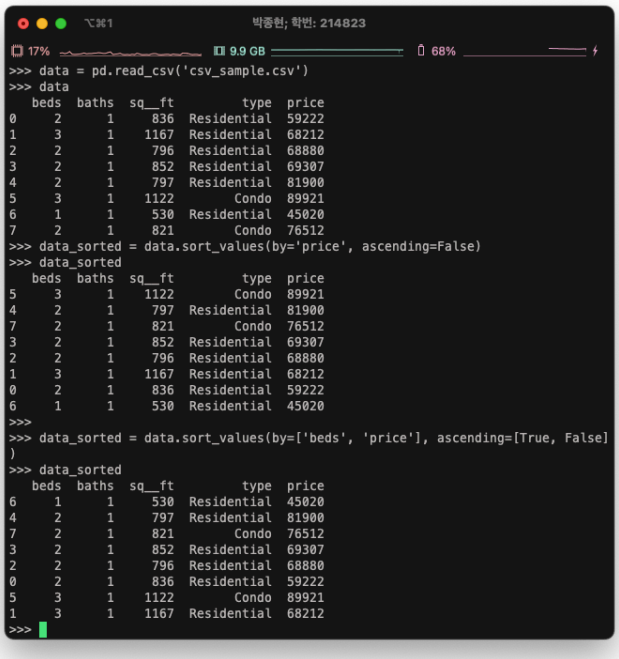
       [[6, 7, 8],
        [9, 10, 11]]])
>>> data > 4
array([[[False, False, False],
        [False, False, True]],

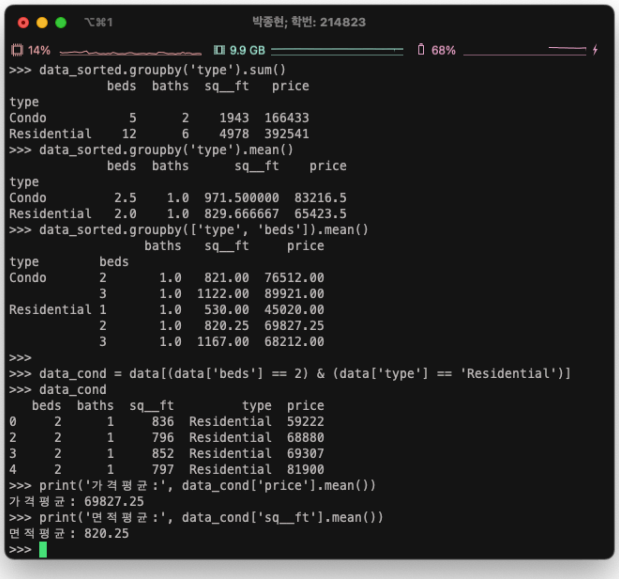
       [[True, True, True],
        [True, True, True]]])
>>> data[data > 4]
array([5, 6, 7, 8, 9, 10, 11])
>>> data[data > 4].sum()
56
>>>

[7] 13-1. 15 16 쪽

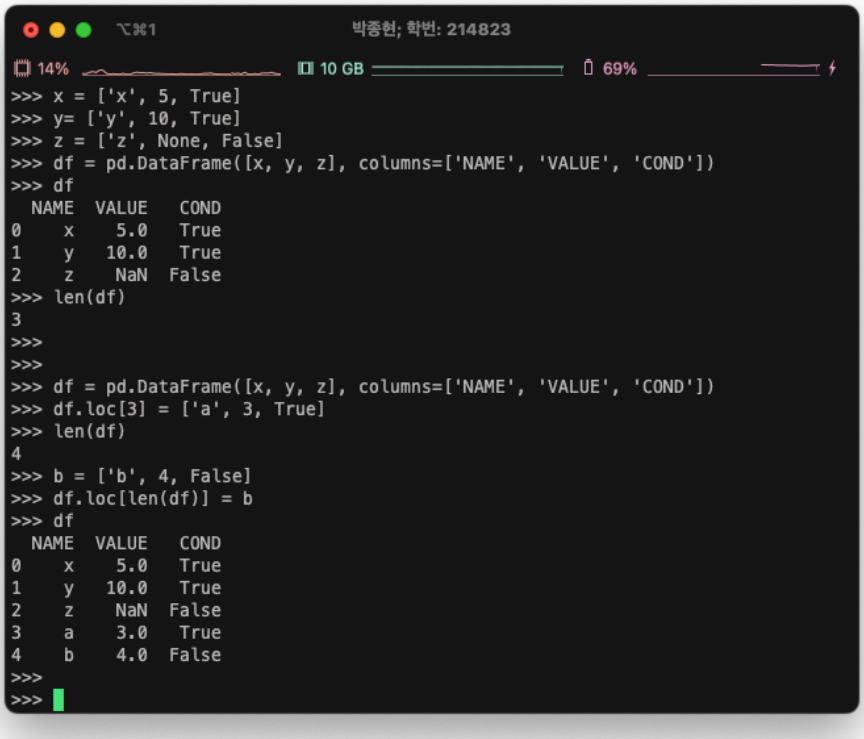


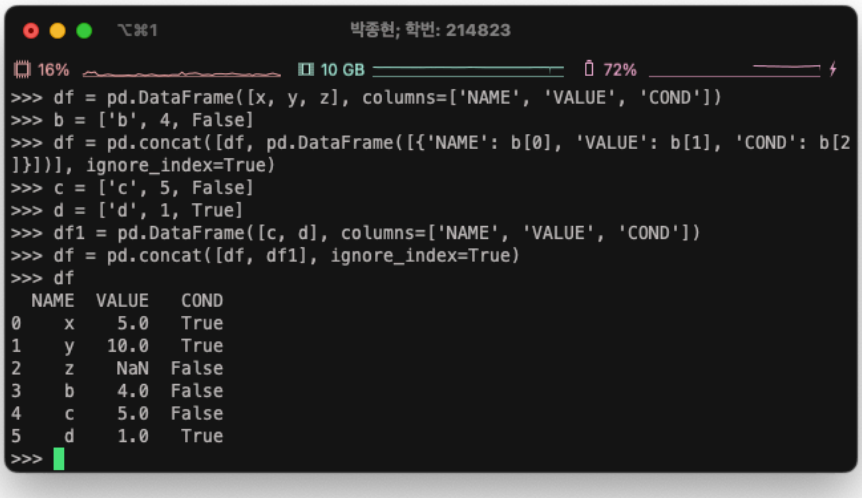
[8] 13-1. 19 23 쪽

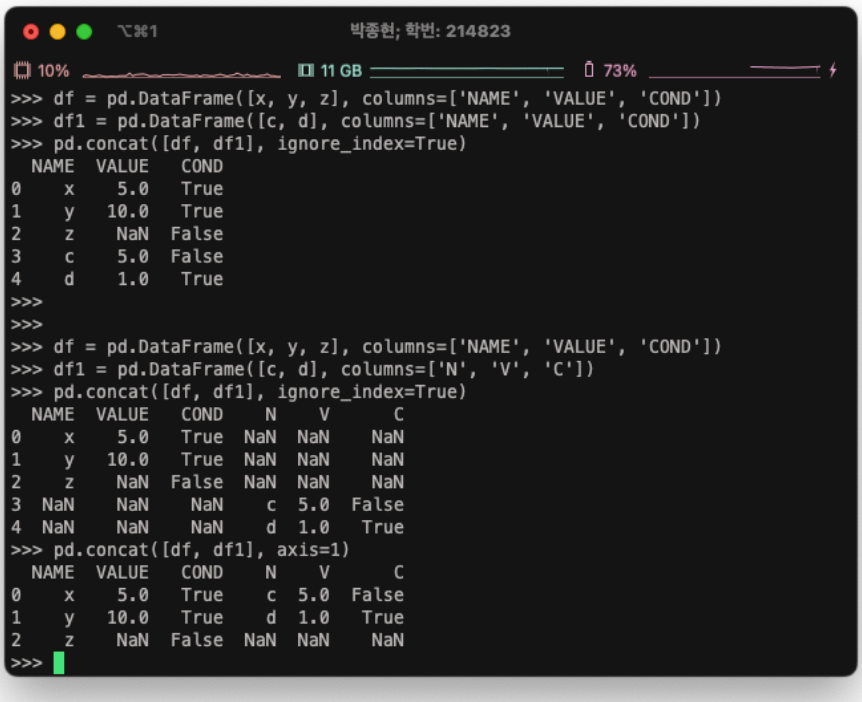




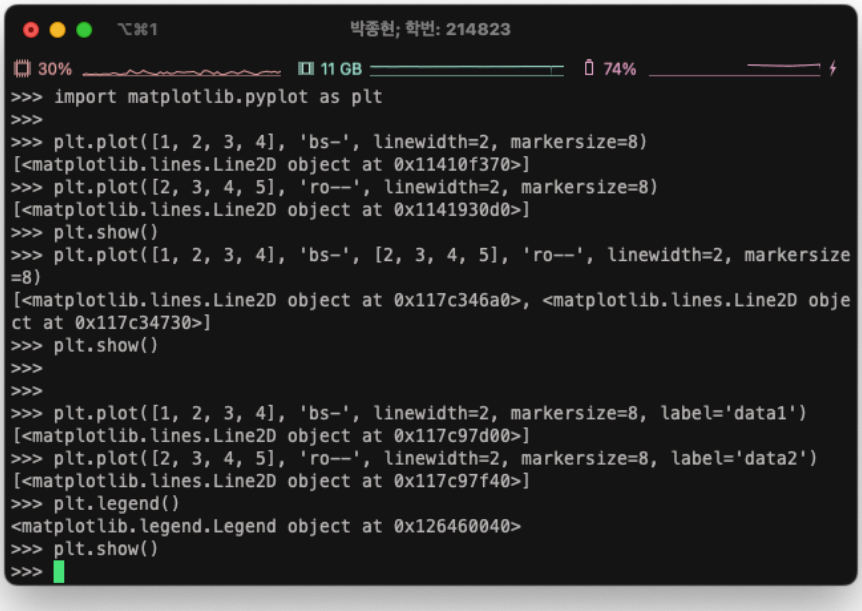
[9] 13-1. 28 32 쪽


⌥⌘1
박종현; 학번: 214823
14%
10 GB
69%
>>> x = ['x', 5, True]
>>> y= ['y', 10, True]
>>> z = ['z', None, False]
>>> df = pd.DataFrame([x, y, z], columns=['NAME', 'VALUE', 'COND'])
>>> df
  NAME VALUE   COND
0   x   5.0   True
1   y   10.0   True
2   z   NaN False
>>> len(df)
3
>>>
>>>
>>> df = pd.DataFrame([x, y, z], columns=['NAME', 'VALUE', 'COND'])
>>> df.loc[3] = ['a', 3, True]
>>> len(df)
4
>>> b = ['b', 4, False]
>>> df.loc[len(df)] = b
>>> df
  NAME VALUE   COND
0   x   5.0   True
1   y   10.0   True
2   z   NaN False
3   a   3.0   True
4   b   4.0 False
>>>
>>>


⌥⌘1
박종현; 학번: 214823
16%
10 GB
72%
>>> df = pd.DataFrame([x, y, z], columns=['NAME', 'VALUE', 'COND'])
>>> b = ['b', 4, False]
>>> df = pd.concat([df, pd.DataFrame([{'NAME': b[0], 'VALUE': b[1], 'COND': b[2
]}])], ignore_index=True)
>>> c = ['c', 5, False]
>>> d = ['d', 1, True]
>>> df1 = pd.DataFrame([c, d], columns=['NAME', 'VALUE', 'COND'])
>>> df = pd.concat([df, df1], ignore_index=True)
>>> df
  NAME VALUE   COND
0   x   5.0   True
1   y   10.0   True
2   z   NaN False
3   b   4.0 False
4   c   5.0 False
5   d   1.0   True
>>>


⌥⌘1
박종현; 학번: 214823
10%
11 GB
73%
>>> df = pd.DataFrame([x, y, z], columns=['NAME', 'VALUE', 'COND'])
>>> df1 = pd.DataFrame([c, d], columns=['NAME', 'VALUE', 'COND'])
>>> pd.concat([df, df1], ignore_index=True)
NAME VALUE COND
0 x 5.0 True
1 y 10.0 True
2 z NaN False
3 c 5.0 False
4 d 1.0 True
>>>
>>>
>>> df = pd.DataFrame([x, y, z], columns=['NAME', 'VALUE', 'COND'])
>>> df1 = pd.DataFrame([c, d], columns=['N', 'V', 'C'])
>>> pd.concat([df, df1], ignore_index=True)
NAME VALUE COND N V C
0 x 5.0 True NaN NaN NaN
1 y 10.0 True NaN NaN NaN
2 z NaN False NaN NaN NaN
3 NaN NaN NaN c 5.0 False
4 NaN NaN NaN d 1.0 True
>>> pd.concat([df, df1], axis=1)
NAME VALUE COND N V C
0 x 5.0 True c 5.0 False
1 y 10.0 True d 1.0 True
2 z NaN False NaN NaN NaN
>>>

[10] 13-2. 11 12 쪽


⌥⌘1
박종현; 학번: 214823
30%
11 GB
74%
>>> import matplotlib.pyplot as plt
>>>
>>> plt.plot([1, 2, 3, 4], 'bs-', linewidth=2, markersize=8)
[<matplotlib.lines.Line2D object at 0x11410f370>]
>>> plt.plot([2, 3, 4, 5], 'ro--', linewidth=2, markersize=8)
[<matplotlib.lines.Line2D object at 0x1141930d0>]
>>> plt.show()
>>> plt.plot([1, 2, 3, 4], 'bs-', [2, 3, 4, 5], 'ro--', linewidth=2, markersize
=8)
[<matplotlib.lines.Line2D object at 0x117c346a0>, <matplotlib.lines.Line2D obje
ct at 0x117c34730>]
>>> plt.show()
>>>
>>>
>>> plt.plot([1, 2, 3, 4], 'bs-', linewidth=2, markersize=8, label='data1')
[<matplotlib.lines.Line2D object at 0x117c97d00>]
>>> plt.plot([2, 3, 4, 5], 'ro--', linewidth=2, markersize=8, label='data2')
[<matplotlib.lines.Line2D object at 0x117c97f40>]
>>> plt.legend()
<matplotlib.legend.Legend object at 0x126460040>
>>> plt.show()
>>>

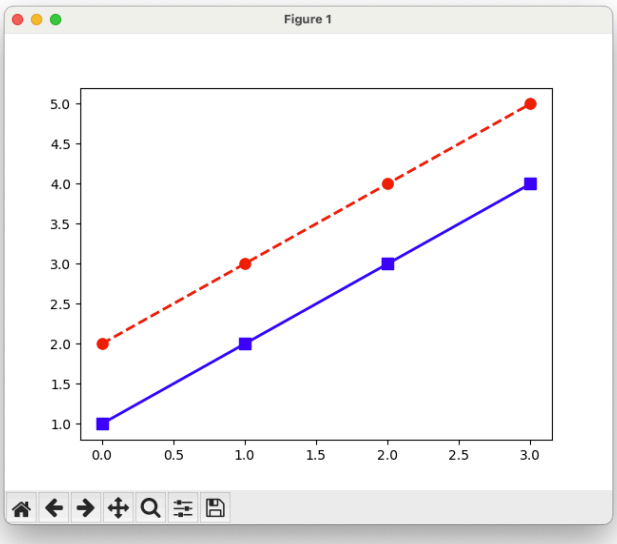


Figure 1
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

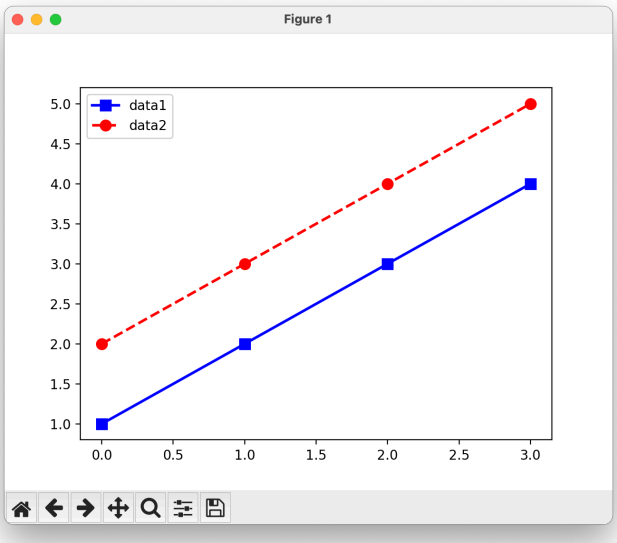
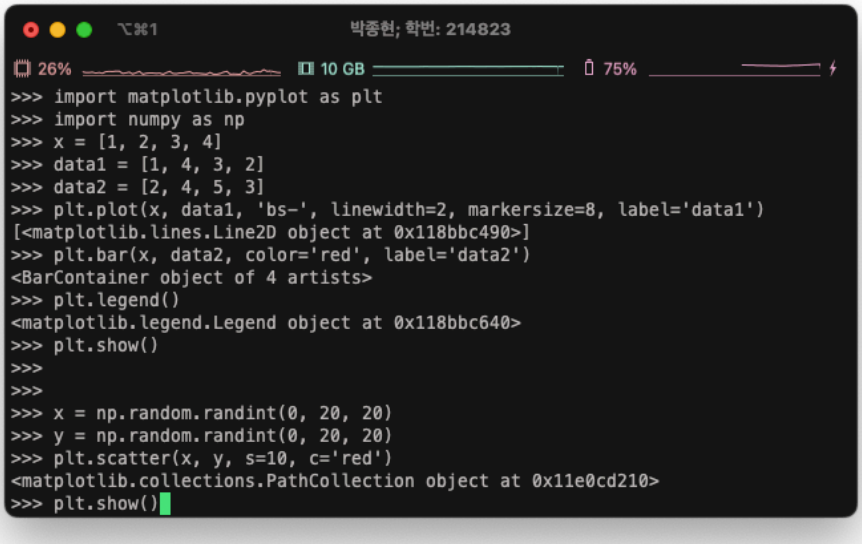


Figure 1
data1
data2
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0

[11] 13-2. 14 19 쪽


⌥⌘1
박종현; 학번: 214823
26%
10 GB
75%
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x = [1, 2, 3, 4]
>>> data1 = [1, 4, 3, 2]
>>> data2 = [2, 4, 5, 3]
>>> plt.plot(x, data1, 'bs-', linewidth=2, markersize=8, label='data1')
[<matplotlib.lines.Line2D object at 0x118bbc490>]
>>> plt.bar(x, data2, color='red', label='data2')
<BarContainer object of 4 artists>
>>> plt.legend()
<matplotlib.legend.Legend object at 0x118bbc640>
>>> plt.show()
>>>
>>>
>>> x = np.random.randint(0, 20, 20)
>>> y = np.random.randint(0, 20, 20)
>>> plt.scatter(x, y, s=10, c='red')
<matplotlib.collections.PathCollection object at 0x11e0cd210>
>>> plt.show()

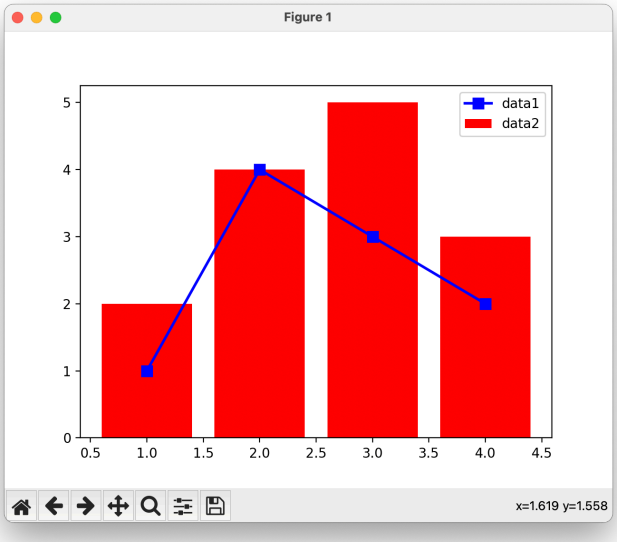


Figure 1
data1
data2
5
4
3
2
1
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
x=1.619 y=1.558

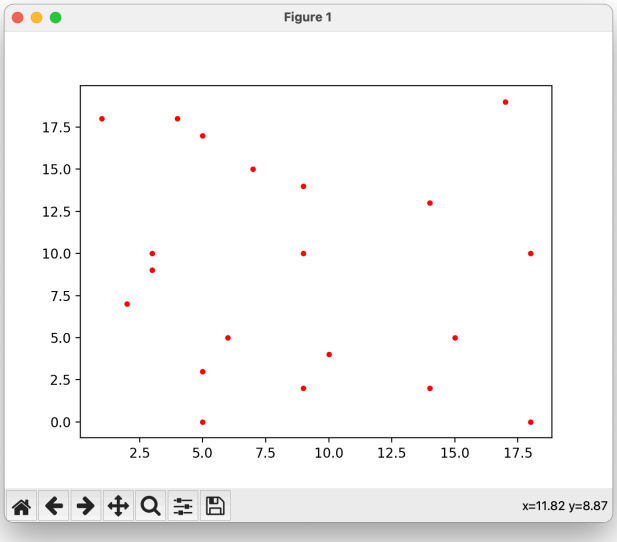
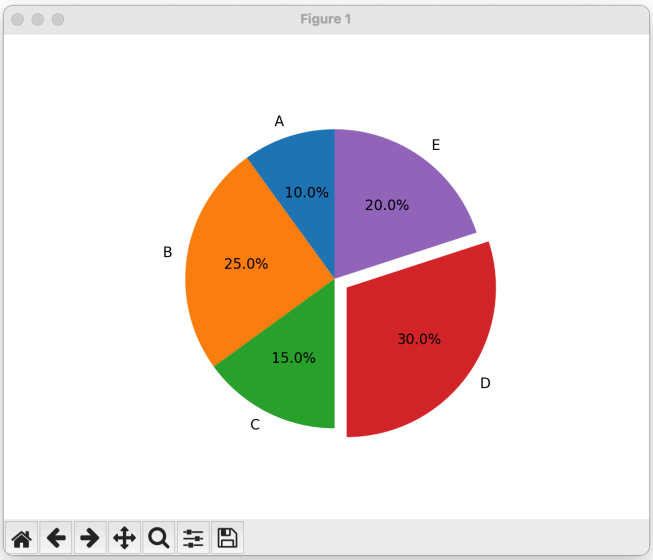
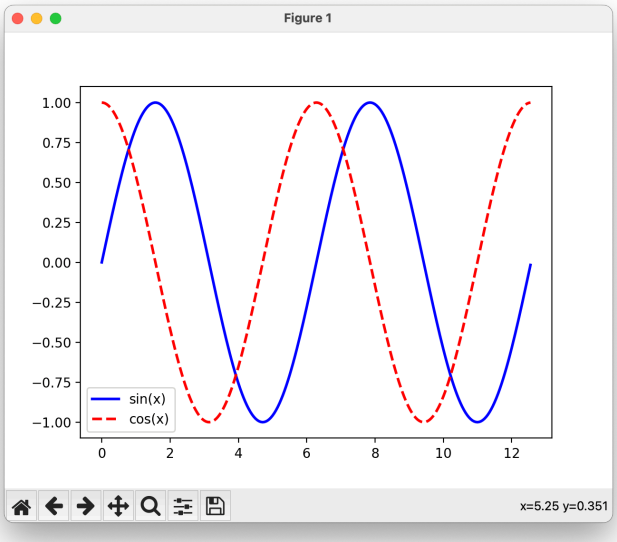


Figure 1
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
x=11.82 y=8.87

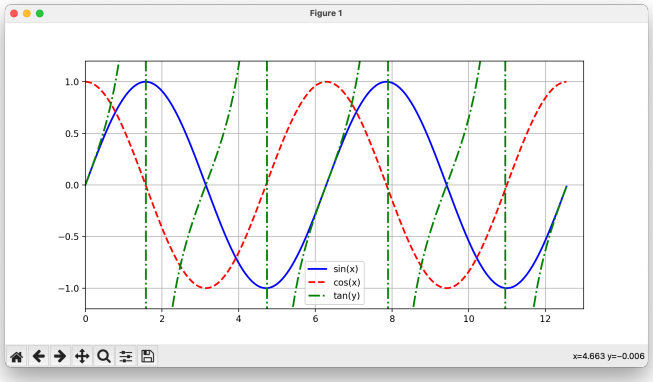
```
⌥⌘1                    박종현; 학번: 214823
16%  10 GB  76%
>>> plt.pie(
...     [10, 25, 15, 30, 20],
...     labels=['A', 'B', 'C', 'D', 'E'],
...     explode=[0, 0, 0, .1, 0],
...     autopct='%.1f%%',
...     startangle=90
... )
([<matplotlib.patches.Wedge object at 0x11eee7f10>, <matplotlib.patches.Wedge object at 0x11eee7e50>, <matplotlib.patches.Wedge object at 0x11ef18c40>, <matplotlib.patches.Wedge object at 0x11ede4460>, <matplotlib.patches.Wedge object at 0x11ef19510>], [Text(-0.3399186987098806, 1.0461621663333949, 'A'), Text(-1.0866457176265761, 0.1720779013721081, 'B'), Text(-0.4993895221842246, -0.9801071906340714, 'C'), Text(0.9708204526849543, -0.7053422209456772, 'D'), Text(0.6465636608734332, 0.8899187785623721, 'E')], [Text(-0.18541019929629848, 0.5706339089091244, '10.0%'), Text(-0.5926130052358697, 0.09386067347569531, '25.0%'), Text(-0.27239428482775885, -0.5346039221640388, '15.0%'), Text(0.56631193073289, -0.4114496288849784, '30.0%'), Text(0.35267108774914535, 0.4854102428522029, '20.0%')])
>>> plt.show()
```



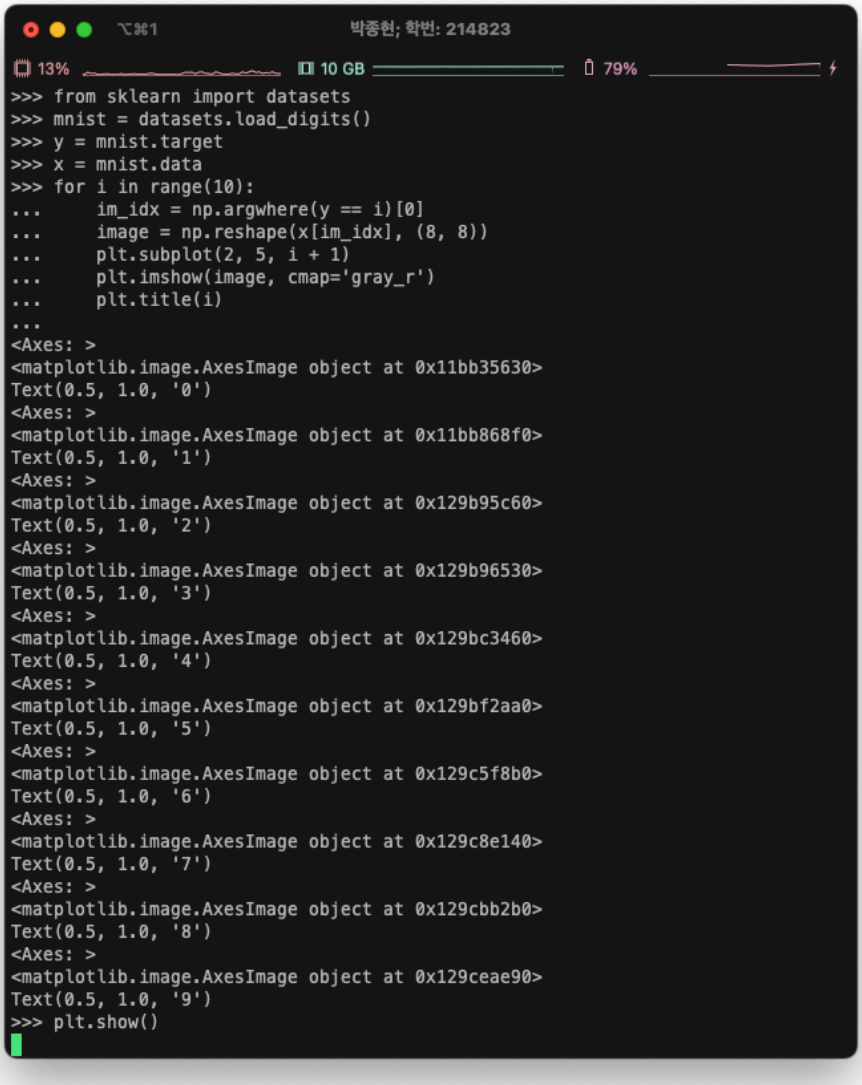

```
⌥⌘1                    박종현; 학번: 214823
24%  10 GB  76%
>>> import numpy as np
>>> x = np.arange(0, 4 * np.pi, .05)
>>> y1 = np.sin(x)
>>> y2 = np.cos(x)
>>> plt.plot(x, y1, 'b-', linewidth=2, label='sin(x)')
[<matplotlib.lines.Line2D object at 0x11ef4baf0>]
>>> plt.plot(x, y2, 'r--', linewidth=2, label='cos(x)')
[<matplotlib.lines.Line2D object at 0x11ef4bd90>]
>>> plt.legend()
<matplotlib.legend.Legend object at 0x11ef4be20>
>>> plt.show()
>>>
```

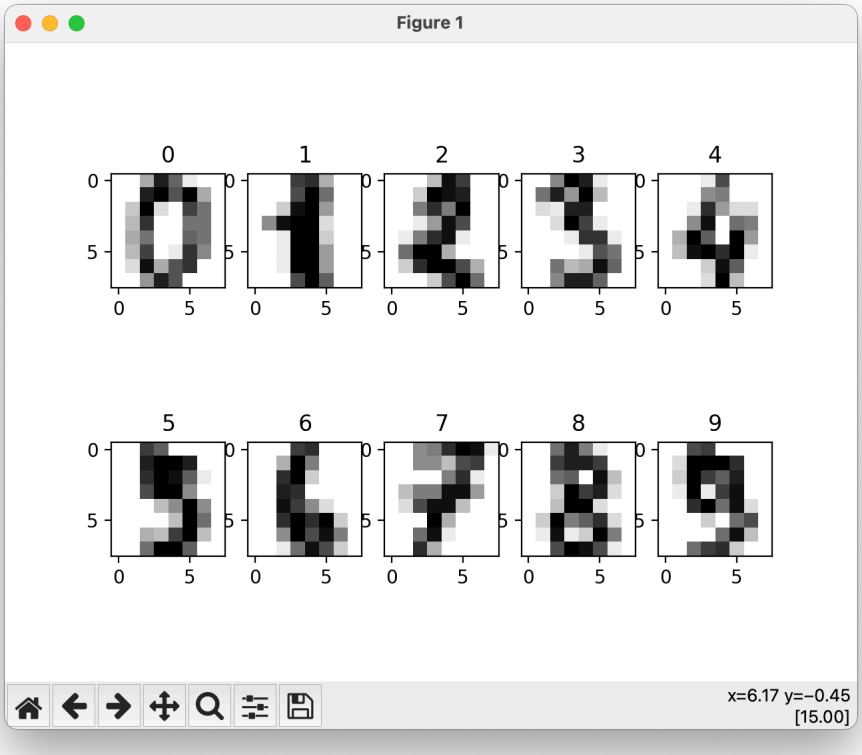



```
⌥⌘1                    박종현; 학번: 214823
24%  10 GB  77%
>>> x = np.arange(0, 4 * np.pi, .05)
>>> y1 = np.sin(x)
>>> y2 = np.cos(x)
>>> y3 = np.tan(x)
>>>
>>> plt.figure(figsize=(10, 5))
<Figure size 2000x1000 with 0 Axes>
>>> plt.plot(x, y1, 'b-', linewidth=2, label='sin(x)')
[<matplotlib.lines.Line2D object at 0x11f12b880>]
>>> plt.plot(x, y2, 'r--', linewidth=2, label='cos(x)')
[<matplotlib.lines.Line2D object at 0x11f12bdf0>]
>>> plt.plot(x, y3, 'g-.', linewidth=2, label='tan(y)')
[<matplotlib.lines.Line2D object at 0x11f12bbe0>]
>>> plt.axis([0, 13, -1.2, 1.2])
(0.0, 13.0, -1.2, 1.2)
>>> plt.legend()
<matplotlib.legend.Legend object at 0x11f12bee0>
>>> plt.grid()
>>> plt.show()
>>>
```

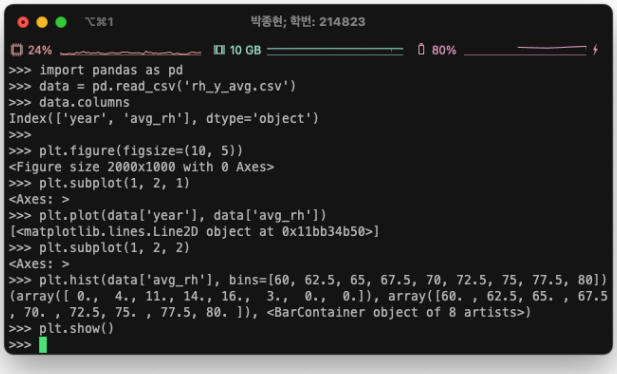


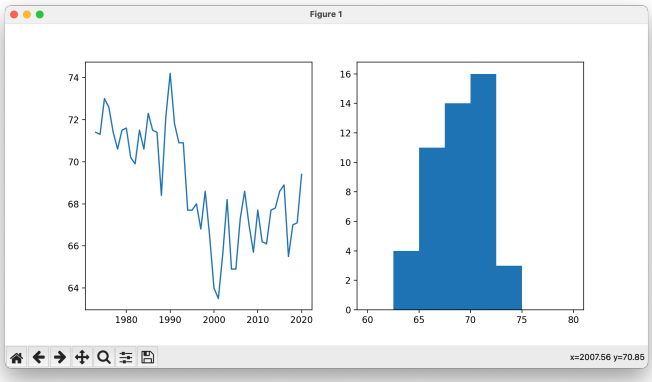
[12] 13-2. 25 26 쪽


⌥⌘1
박종현; 학번: 214823
13%
10 GB
79%
>>> from sklearn import datasets
>>> mnist = datasets.load_digits()
>>> y = mnist.target
>>> x = mnist.data
>>> for i in range(10):
...   im_idx = np.argwhere(y == i)[0]
...   image = np.reshape(x[im_idx], (8, 8))
...   plt.subplot(2, 5, i + 1)
...   plt.imshow(image, cmap='gray_r')
...   plt.title(i)
...
<Axes: >
<matplotlib.image.AxesImage object at 0x11bb35630>
Text(0.5, 1.0, '0')
<Axes: >
<matplotlib.image.AxesImage object at 0x11bb868f0>
Text(0.5, 1.0, '1')
<Axes: >
<matplotlib.image.AxesImage object at 0x129b95c60>
Text(0.5, 1.0, '2')
<Axes: >
<matplotlib.image.AxesImage object at 0x129b96530>
Text(0.5, 1.0, '3')
<Axes: >
<matplotlib.image.AxesImage object at 0x129bc3460>
Text(0.5, 1.0, '4')
<Axes: >
<matplotlib.image.AxesImage object at 0x129bf2aa0>
Text(0.5, 1.0, '5')
<Axes: >
<matplotlib.image.AxesImage object at 0x129c5f8b0>
Text(0.5, 1.0, '6')
<Axes: >
<matplotlib.image.AxesImage object at 0x129c8e140>
Text(0.5, 1.0, '7')
<Axes: >
<matplotlib.image.AxesImage object at 0x129cbb2b0>
Text(0.5, 1.0, '8')
<Axes: >
<matplotlib.image.AxesImage object at 0x129ceae90>
Text(0.5, 1.0, '9')
>>> plt.show()

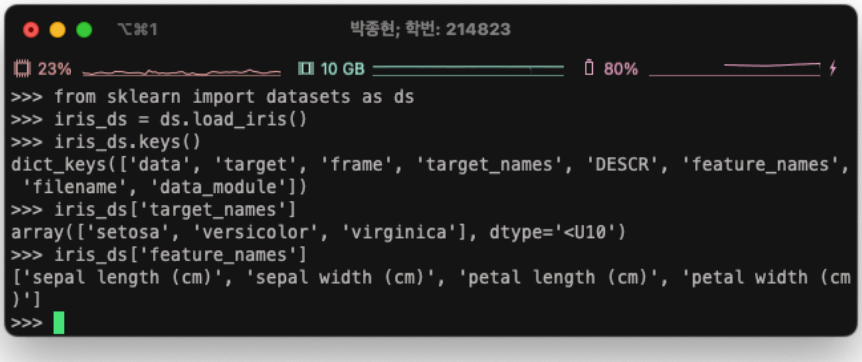


[13] 13-2. 28 쪽





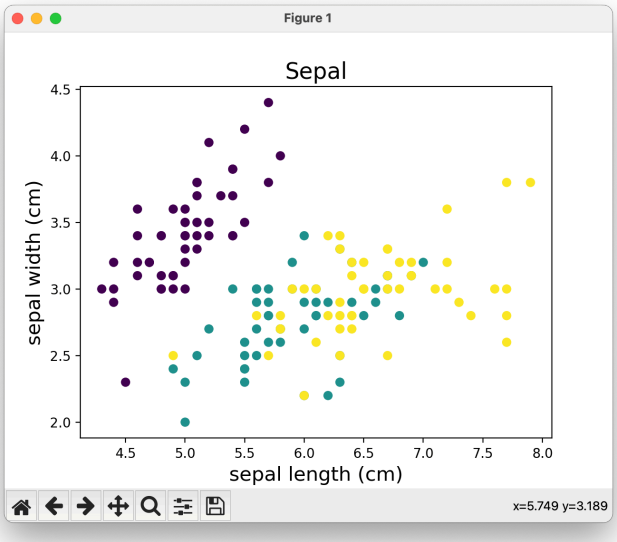
[14] 13-3. 18 23 쪽



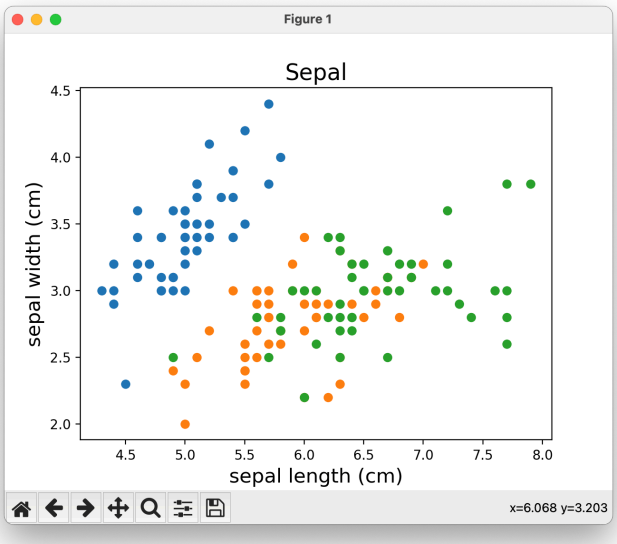
```
박종현; 학번: 214823
24%  10 GB  80%
>>> iris_data = iris_ds['data']
>>> type(iris_data)
<class 'numpy.ndarray'>
>>> iris_data.shape
(150, 4)
>>> iris_data[:4]
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2]])
>>>
```

```
박종현; 학번: 214823
24%  10 GB  80%
>>> iris_target = iris_ds['target']
>>> type(iris_target)
<class 'numpy.ndarray'>
>>> iris_target.shape
(150,)
>>> iris_target[:4]
array([0, 0, 0, 0])
>>> iris_target[146:]
array([2, 2, 2, 2])
>>>
```

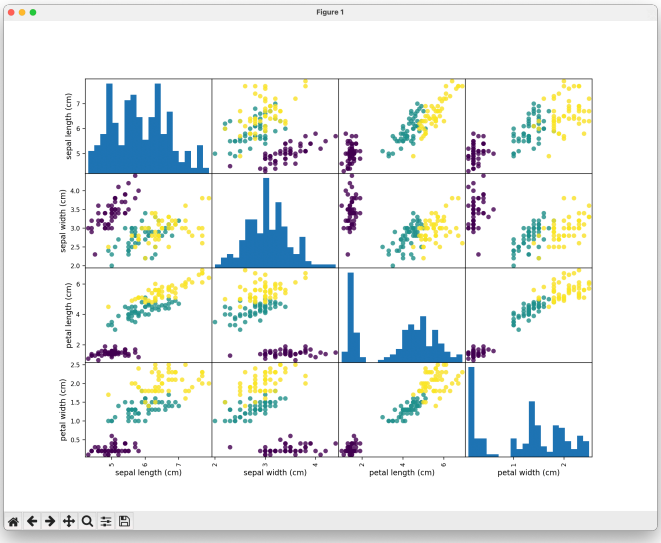
```
박종현; 학번: 214823
25%  10 GB  80%
>>> plt.scatter(iris_data[:, 0], iris_data[:, 1], c=iris_target)
<matplotlib.collections.PathCollection object at 0x1587460b0>
>>> plt.title('Sepal', fontsize=17)
Text(0.5, 1.0, 'Sepal')
>>> plt.xlabel(iris_ds['feature_names'][0], fontsize=15)
Text(0.5, 47.04444444444444, 'sepal length (cm)')
>>> plt.ylabel(iris_ds['feature_names'][1], fontsize=15)
Text(85.31944444444443, 0.5, 'sepal width (cm)')
>>> plt.show()
>>>
```



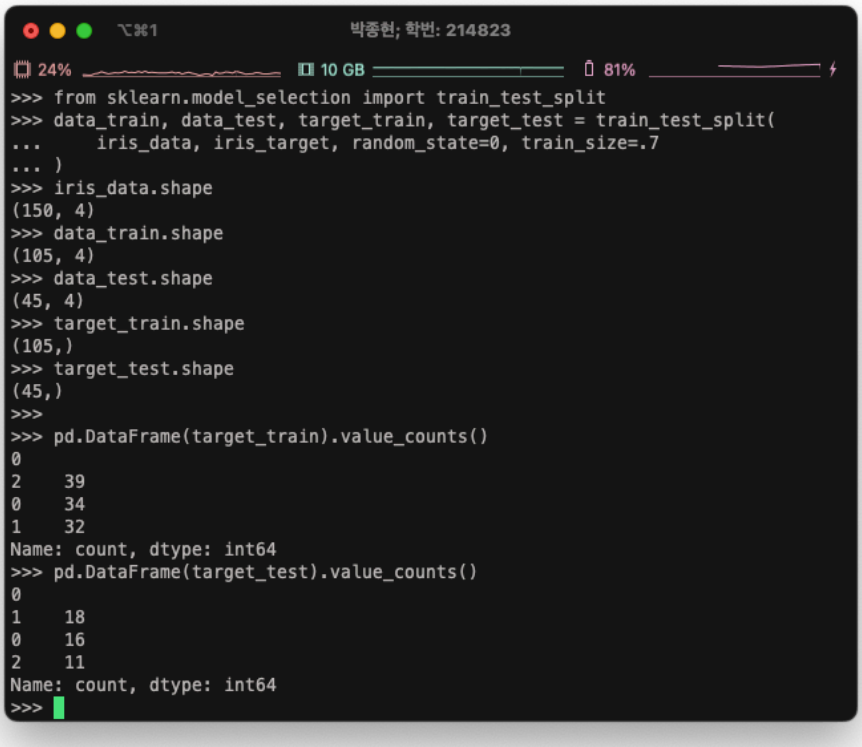
```
⌥⌘1                    박종현; 학번: 214823
26%  10 GB  81%
>>> for i in np.unique(iris_target):
...     plt.scatter(iris_data[iris_target == i, 0],
...         iris_data[iris_target == i, 1],
...         label=iris_ds['target_names'][i])
...
<matplotlib.collections.PathCollection object at 0x1587b0220>
<matplotlib.collections.PathCollection object at 0x1587b0670>
<matplotlib.collections.PathCollection object at 0x1587b0910>
>>> plt.title('Sepal', fontsize=17)
Text(0.5, 1.0, 'Sepal')
>>> plt.xlabel(iris_ds['feature_names'][0], fontsize=15)
Text(0.5, 47.04444444444444, 'sepal length (cm)')
>>> plt.ylabel(iris_ds['feature_names'][1], fontsize=15)
Text(85.31944444444443, 0.5, 'sepal width (cm)')
>>> plt.show()
>>>
```

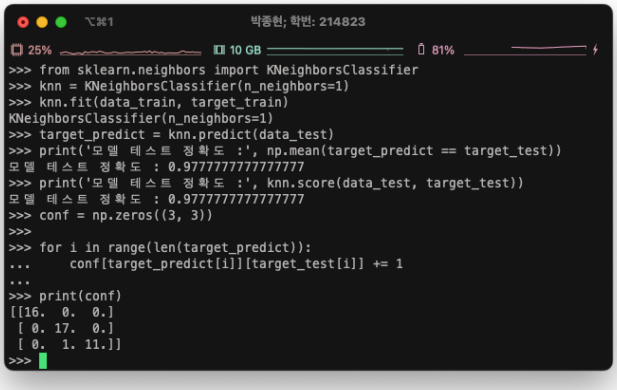


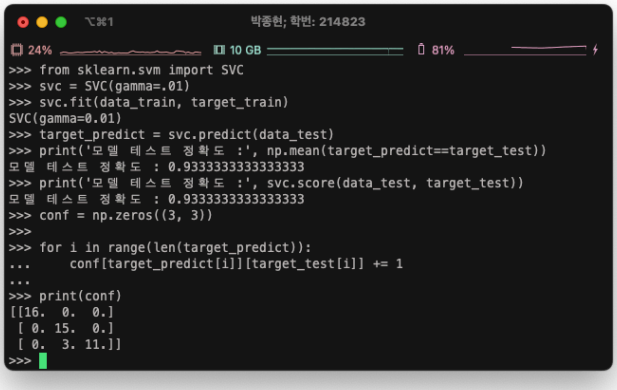

```
⌥⌘1                    박종현; 학번: 214823
15%  10 GB  81%
>>> iris_df = pd.DataFrame(iris_data, columns=iris_ds['feature_names'])
>>> pd.plotting.scatter_matrix(
...     iris_df,
...     c=iris_target,
...     figsize=(12, 9),
...     marker='o',
...     hist_kwds={'bins': 20},
...     alpha=.8
... )
array([[<Axes: xlabel='sepal length (cm)', ylabel='sepal length (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='sepal length (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='sepal length (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='sepal length (cm)'>],
       [<Axes: xlabel='sepal length (cm)', ylabel='sepal width (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='sepal width (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='sepal width (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='sepal width (cm)'>],
       [<Axes: xlabel='sepal length (cm)', ylabel='petal length (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='petal length (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='petal length (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='petal length (cm)'>],
       [<Axes: xlabel='sepal length (cm)', ylabel='petal width (cm)'>,
        <Axes: xlabel='sepal width (cm)', ylabel='petal width (cm)'>,
        <Axes: xlabel='petal length (cm)', ylabel='petal width (cm)'>,
        <Axes: xlabel='petal width (cm)', ylabel='petal width (cm)'>]],
      dtype=object)
>>> plt.show()
```



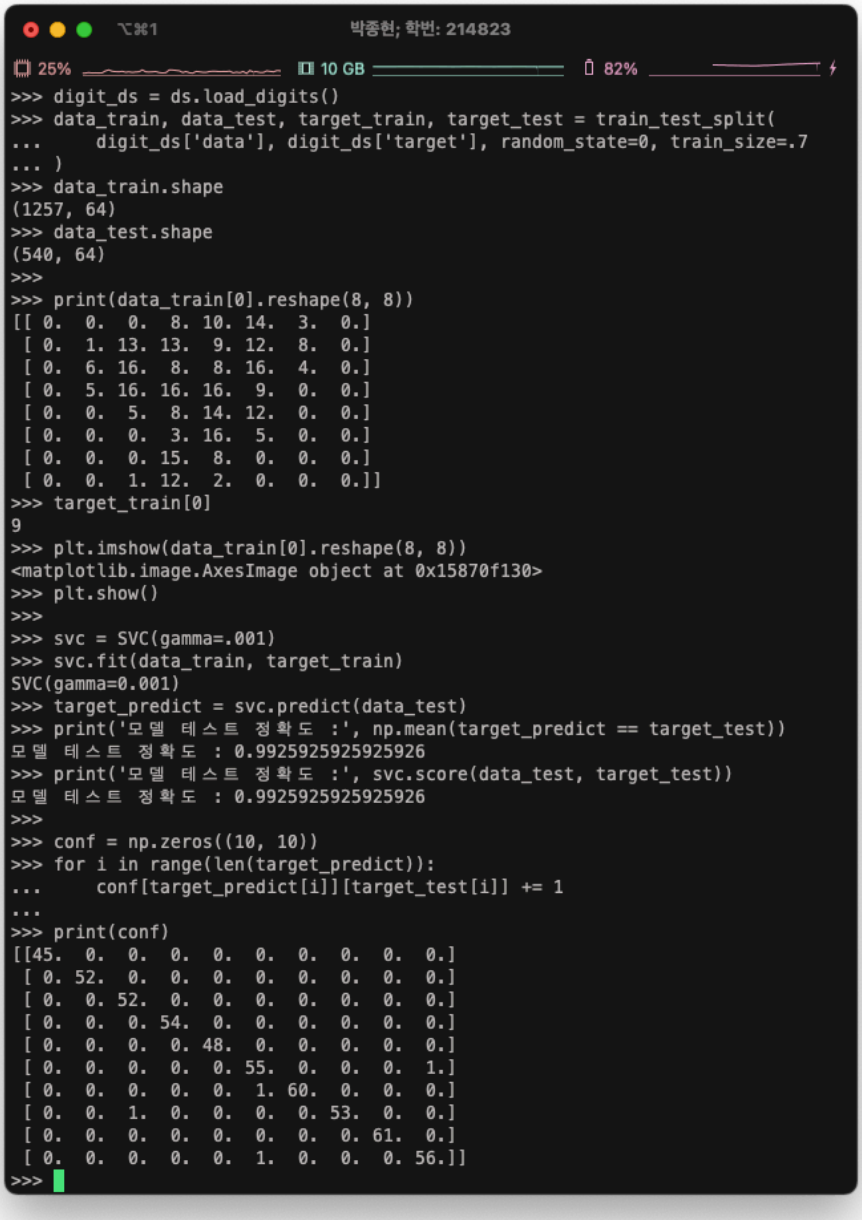
[15] 13-3. 25 28 쪽


⌥⌘1
박종현; 학번: 214823
24%
10 GB
81%
>>> from sklearn.model_selection import train_test_split
>>> data_train, data_test, target_train, target_test = train_test_split(
...   iris_data, iris_target, random_state=0, train_size=.7
...)
>>> iris_data.shape
(150, 4)
>>> data_train.shape
(105, 4)
>>> data_test.shape
(45, 4)
>>> target_train.shape
(105,)
>>> target_test.shape
(45,)
>>>
>>> pd.DataFrame(target_train).value_counts()
0
2   39
0   34
1   32
Name: count, dtype: int64
>>> pd.DataFrame(target_test).value_counts()
0
1   18
0   16
2   11
Name: count, dtype: int64
>>>


⌥⌘1
박종현; 학번: 214823
25%
10 GB
81%
>>> from sklearn.neighbors import KNeighborsClassifier
>>> knn = KNeighborsClassifier(n_neighbors=1)
>>> knn.fit(data_train, target_train)
KNeighborsClassifier(n_neighbors=1)
>>> target_predict = knn.predict(data_test)
>>> print('모델 테스트 정확도 :', np.mean(target_predict == target_test))
모델 테스트 정확도 : 0.9777777777777777
>>> print('모델 테스트 정확도 :', knn.score(data_test, target_test))
모델 테스트 정확도 : 0.9777777777777777
>>> conf = np.zeros((3, 3))
>>>
>>> for i in range(len(target_predict)):
...   conf[target_predict[i]][target_test[i]] += 1
...
>>> print(conf)
[[16. 0. 0.]
 [0. 17. 0.]
 [0. 1. 11.]]
>>>


⌥⌘1
박종현; 학번: 214823
24%
10 GB
81%
>>> from sklearn.svm import SVC
>>> svc = SVC(gamma=.01)
>>> svc.fit(data_train, target_train)
SVC(gamma=0.01)
>>> target_predict = svc.predict(data_test)
>>> print('모델 테스트 정확도 :', np.mean(target_predict==target_test))
모델 테스트 정확도 : 0.9333333333333333
>>> print('모델 테스트 정확도 :', svc.score(data_test, target_test))
모델 테스트 정확도 : 0.9333333333333333
>>> conf = np.zeros((3, 3))
>>>
>>> for i in range(len(target_predict)):
...   conf[target_predict[i]][target_test[i]] += 1
...
>>> print(conf)
[[16. 0. 0.]
 [0. 15. 0.]
 [0. 3. 11.]]
>>>

[16] 13-3. 29 31 쪽


⌥⌘1
박종현; 학번: 214823
25%
10 GB
82%
>>> digit_ds = ds.load_digits()
>>> data_train, data_test, target_train, target_test = train_test_split(
...   digit_ds['data'], digit_ds['target'], random_state=0, train_size=.7
...)
>>> data_train.shape
(1257, 64)
>>> data_test.shape
(540, 64)
>>>
>>> print(data_train[0].reshape(8, 8))
[[0. 0. 0. 8. 10. 14. 3. 0.]
 [0. 1. 13. 13. 9. 12. 8. 0.]
 [0. 6. 16. 8. 8. 16. 4. 0.]
 [0. 5. 16. 16. 16. 9. 0. 0.]
 [0. 0. 5. 8. 14. 12. 0. 0.]
 [0. 0. 0. 3. 16. 5. 0. 0.]
 [0. 0. 0. 15. 8. 0. 0. 0.]
 [0. 0. 1. 12. 2. 0. 0. 0.]]
>>> target_train[0]
9
>>> plt.imshow(data_train[0].reshape(8, 8))
<matplotlib.image.AxesImage object at 0x15870f130>
>>> plt.show()
>>>
>>> svc = SVC(gamma=.001)
>>> svc.fit(data_train, target_train)
SVC(gamma=0.001)
>>> target_predict = svc.predict(data_test)
>>> print('모델 테스트 정확도 :', np.mean(target_predict == target_test))
모델 테스트 정확도 : 0.9925925925925926
>>> print('모델 테스트 정확도 :', svc.score(data_test, target_test))
모델 테스트 정확도 : 0.9925925925925926
>>>
>>> conf = np.zeros((10, 10))
>>> for i in range(len(target_predict)):
...   conf[target_predict[i]][target_test[i]] += 1
...
>>> print(conf)
[[45. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 52. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 52. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 54. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 48. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 55. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 1. 60. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 53. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 61. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 56.]]
>>>

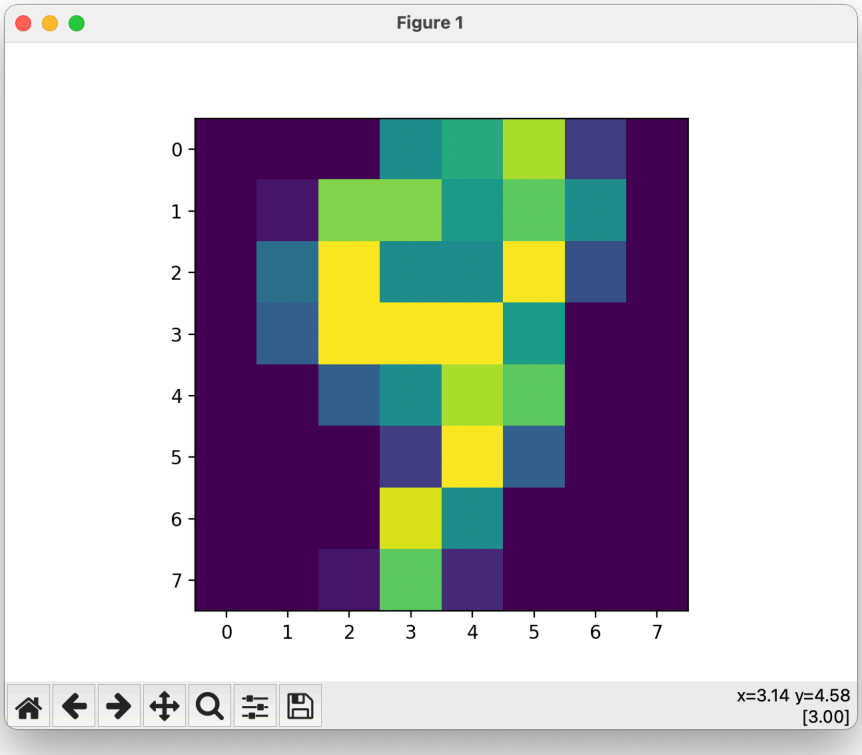


Figure 1
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
x=3.14 y=4.58
[3.00]