

중요사항

데이터 시각화

데이터 시각화(data visualization)는 점이나 선, 막대 그래프등의 시각적 이미지를 사용하여 데이터를 화면에 표시하는 기술

인간이 통찰을 통해 데이터에 내재하는 패턴을 알아내는 데도 유용하다.

교육이나 홍보 등의 분야에서도 중요한 의사 소통의 도구로 간주된다.

matplotlib

가장 널리 사용되는 시각화 도구.

matplotlib 함수

`import matplotlib.pyplot as plt` - pyplot 모듈을 불러와서 plt라는 별칭으로 지정한다.

`plt.plot(x축 값, y축 값, color='색', marker='마커의 모양', linestyle='선 두께')`: plt를 만들기 위한 함수, 그래프의 색깔과 두께, 마커의 모양 등을 설정할 수 있다. (---는 점선, -는 실선, :는 짧은 점선)

`plt.title("제목")`

`plt.xlabel('x축 이름')`

`plt.ylabel('y축 이름')`

`plt.show()`: 화면에 차트를 그리기 위해 필수적인 함수.

`plt.savefig('파일이름.png')`: 그림을 저장하기 위한 함수.

`plt.axis([x축 시작값, x축 끝값, y축 시작값, y축 끝값])`: 그림을 그릴 영역을 지정한다.

`plt.legend()`: 디폴트 위치에 범례를 표시한다.

`plt.bar(x축, y축)`: 막대모양 그래프를 그린다.

`plt.xticks(x축의 길이, 바꿀 값)`: x축 범위의 눈금값을 지정한다.

산포도 플롯(scatter plot)

산포도 플롯은 개별 데이터 포인트를 그리는 차트

`scatter()`함수 사용

파이 차트(pie chart)

데이터의 값에 따라서 원형 비율로 나누어져 있는 차트

`pie()`함수 사용

히스토그램(histogram)

주어진 자료를 몇 개의 구간으로 나누고 각 구간의 도수를 조사하여 나타낸 막대 그래프

`hist()`함수 이용, bin = 사용할 막대 수, alpha = 투명도 정도

상자 차트(box chart)

데이터의 최대, 최소, 중간값과 사분위 수 등을 효율적으로 가시화할 수 있는 방법
 boxplot() 함수 2개 이상의 상자 차트를 각각 그릴때 사용

한 화면에 여러 그래프 그리기

subplots(행의 개수, 열의 개수)
 ax[행 번호, 열 번호]

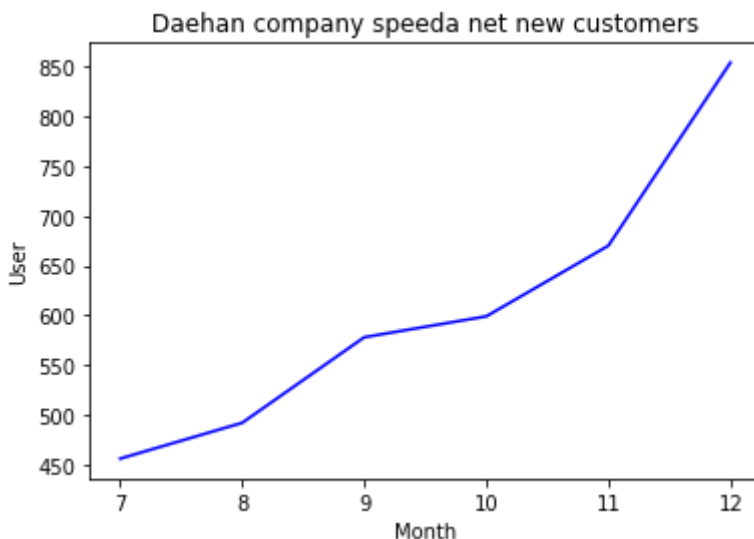
자기점검

1.2 문제에서 각 차트마다 제목을 붙이는 방법을 모르겠다. 1.3 문제에서 랜덤 함수를 사용해서 산포도 그래프를 그리는 방법을 모르겠다. 랜덤 수로 2차원 좌표쌍을 만드는 법을 모르겠다.

심화문제 풀이

11-1

```
In [14]: import matplotlib.pyplot as plt
user = [456, 492, 578, 599, 670, 854]
month = [7, 8, 9, 10, 11, 12]
plt.plot(month, user, color = 'blue')
plt.title('Daehan company speeda net new customers')
plt.ylabel('User')
plt.xlabel('Month')
plt.show()
```

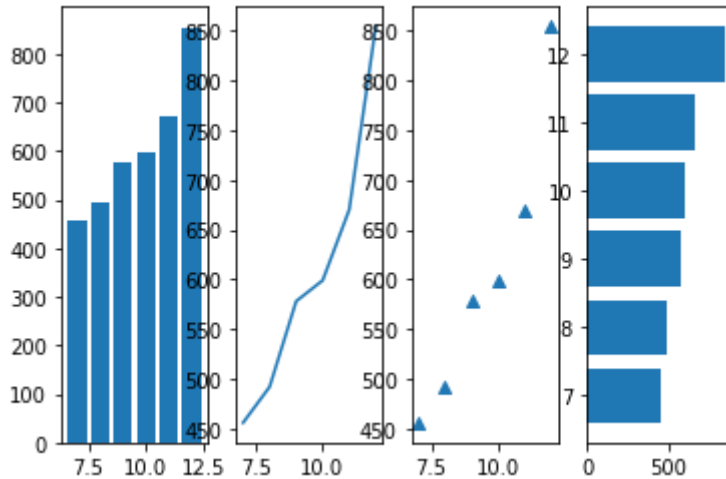


```
In [35]: import matplotlib.pyplot as plt
user = [456, 492, 578, 599, 670, 854]
month = [7, 8, 9, 10, 11, 12]

fig, ax = plt.subplots(1,4)

ax[0].bar(month, user)
ax[1].plot(month, user)
```

```
ax[2].scatter(month, user, marker = '^')
ax[3].barh(month, user)
plt.show()
```

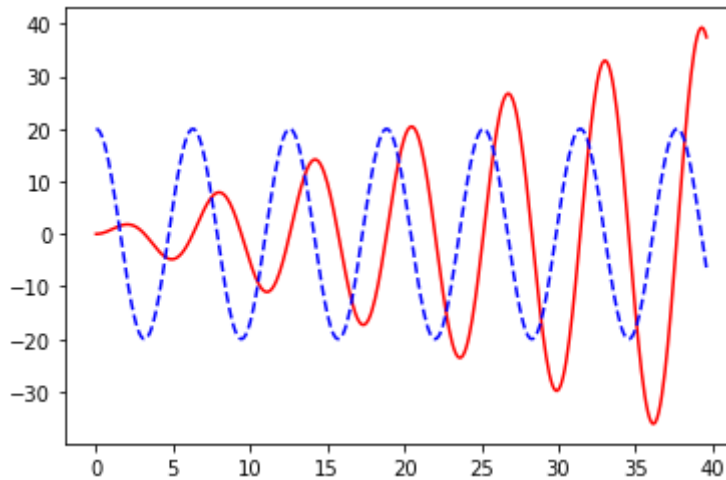


11-2

In [13]:

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

a = np.arange(0, (2*np.pi)*6+2, 0.1)
plt.plot(a, a*np.sin(a), '-r')
plt.plot(a, 20*np.cos(a), '--b')
plt.show()
```



11-3

In [43]:

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

np.random.seed(1000)
data = np.random.randint(0,50)

plt.scatter(data[0], data[1])
plt.show()
```

TypeError Traceback (most recent call last)
 ~\AppData\Local\Temp\ipykernel_22924\570613234.py in <module>

```

6
7
----> 8 plt.scatter(data[0], data[1])

```

TypeError: 'int' object is not subscriptable

11-4

In [44]:

```

import matplotlib.pyplot as plt
import numpy as np

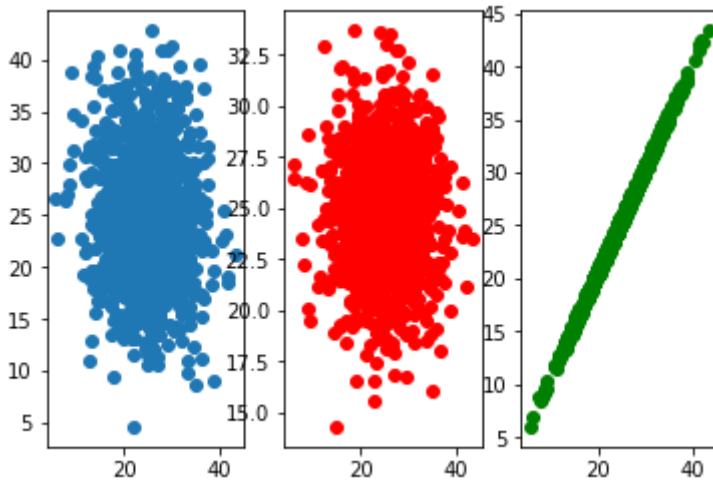
mu = 25
sigma1 = 6
sigma2 = 3

x = [mu + sigma1 * np.random.randn(1000)]
y1 = [mu + sigma1 * np.random.randn(1000)]
y2 = [mu + sigma2 * np.random.randn(1000)]
y3 = [x + np.random.rand(1000)]

fig, ax = plt.subplots(1, 3)

ax[0].scatter(x, y1)
ax[1].scatter(x, y2, color = 'red')
ax[2].scatter(x, y3, color = 'green')
plt.show()

```



11-5

In [51]:

```

import matplotlib.pyplot as plt
import numpy as np

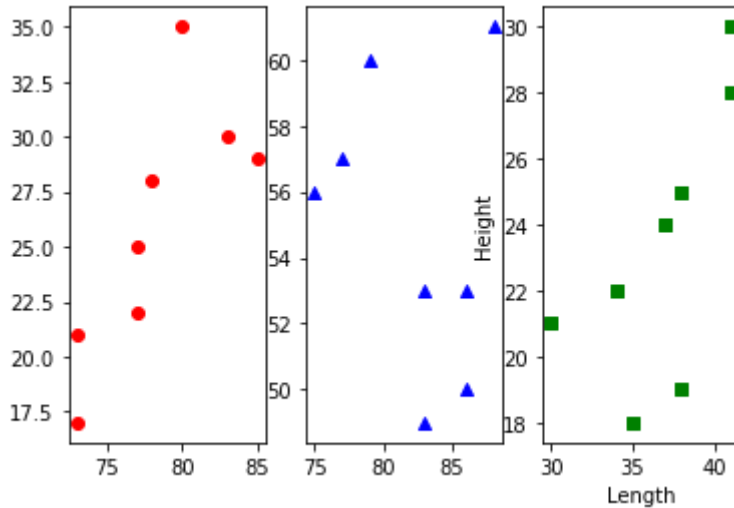
d_length = [77, 78, 85, 83, 73, 77, 73, 80]
d_height = [25, 28, 29, 30, 21, 22, 17, 35]
s_length = [75, 77, 86, 86, 79, 83, 83, 88]
s_height = [56, 57, 50, 53, 60, 53, 49, 61]
m_length = [34, 38, 38, 41, 30, 37, 41, 35]
m_height = [22, 25, 19, 30, 21, 24, 28, 18]

fig, ax = plt.subplots(1,3)

ax[0].scatter(d_length, d_height, color = 'red', marker = 'o')
ax[1].scatter(s_length, s_height, color = 'blue', marker = '^')
ax[2].scatter(m_length, m_height, color = 'green', marker = 's')

```

```
plt.xlabel('Length')
plt.ylabel('Height')
plt.show()
```



In [54]:

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

d_length = [77, 78, 85, 83, 73, 77, 73, 80]
d_height = [25, 28, 29, 30, 21, 22, 17, 35]
s_length = [75, 77, 86, 86, 79, 83, 83, 88]
s_height = [56, 57, 50, 53, 60, 53, 49, 61]
m_length = [34, 38, 38, 41, 30, 37, 41, 35]
m_height = [22, 25, 19, 30, 21, 24, 28, 18]

plt.scatter(d_length, d_height, label = 'Dachshund', color = 'red', marker = 'o')
plt.scatter(s_length, s_height, label = 'Samoyed', color = 'blue', marker = '^')
plt.scatter(m_length, m_height, label = 'Maltese', color = 'green', marker = 's')

plt.xlabel('Length')
plt.ylabel('Height')
plt.title('dog size')
plt.legend()
plt.show()
```

