파이썬을 이용한 시각화 기본

학습 목표

- matplotlib를 활용한 시각화에 대해 알아봅니다.

학습 내용

- 5. 타이틀과 레이블
- 6. Grid
- 7. 여러개 그래프 전체 타이틀 표시 - SuperTitle
- 8. Scatter Plot(산점도)
- 9. Alpha(투명도)
- 10. Barplot(막대그래프)
- 11. Horizontal Bars(수평 막대그래프)
- 12. 히스토그램
- 13. pie chart(원그래프)
- 14. 3D Plots

목차

05. 타이틀과 레이블
06. Grid
07. 여러개 그래프 전체 타이틀 표시 - SuperTitle
08. Scatter Plot(산점도)
09. Alpha(투명도)
10. Barplot(막대그래프)
11. Horizontal Bars(수평 막대그래프)
12. 히스토그램
13. pie chart(원그래프)
14. 3D Plots

In [1]:

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

print(matplotlib.__version__)
```

```
3.3.2
```

05. 타이틀과 레이블

목차로

In [2]:

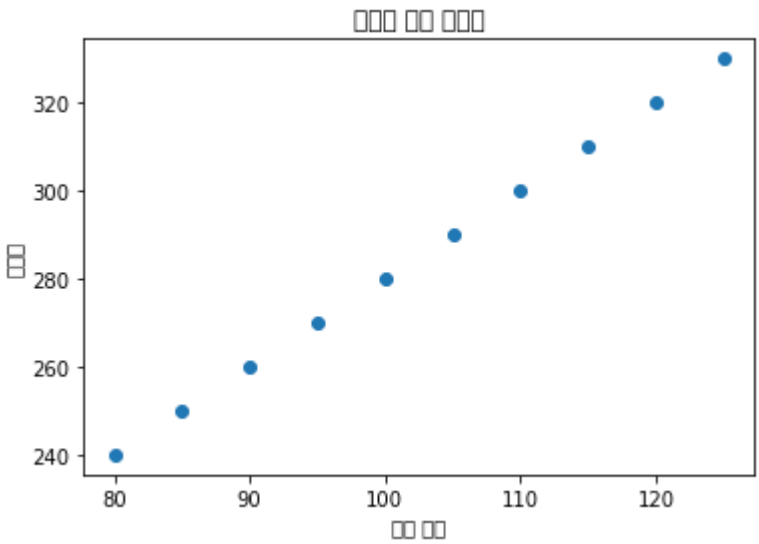
```
import os, warnings
warnings.filterwarnings(action='ignore')
```

In [3]:

```
x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.plot(x, y,'o')

plt.title("스포츠 시청 데이터")
plt.xlabel("평균 혈압")
plt.ylabel("칼로리")
```

Out[3]:

```
Text(0, 0.5, '칼로리')
```



한글 표기

In [4]:

```
from matplotlib import font_manager, rc
import platform
import matplotlib.pyplot as plt
```

In [5]:

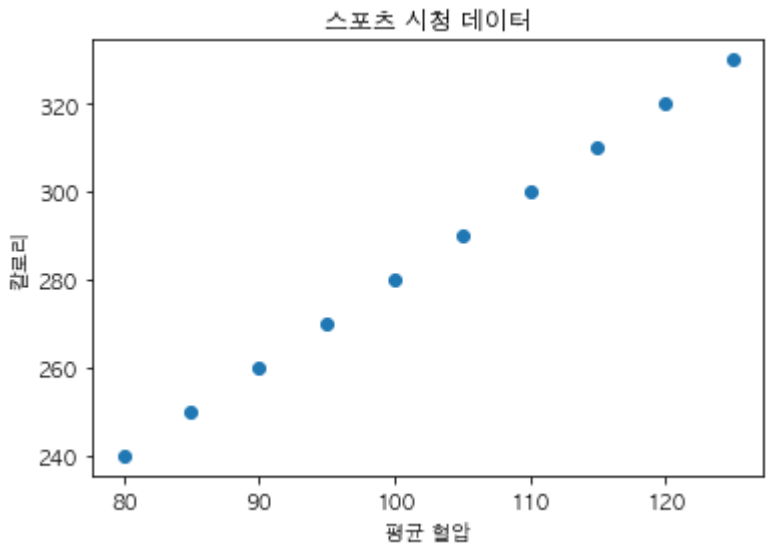
```
path = "C:/Windows/Fonts/malgun.ttf"
if platform.system() == "Windows":
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
elif platform.system()=="Darwin":
    rc('font', family='AppleGothic')
else:
    print("Unknown System")

matplotlib.rcParams['axes.unicode_minus'] = False
```

In [6]:

```
plt.plot(x, y,'o')

plt.title("스포츠 시청 데이터")
plt.xlabel("평균 혈압")
plt.ylabel("칼로리")
```

Out[6]:

```
Text(0, 0.5, '칼로리')
```



06. Grid

- 격자, 모눈이라는 뜻이다. 내용을 구성하는 데 사용되는 일련의 교차하는 직선 또는 곡선으로 구성된 구조.

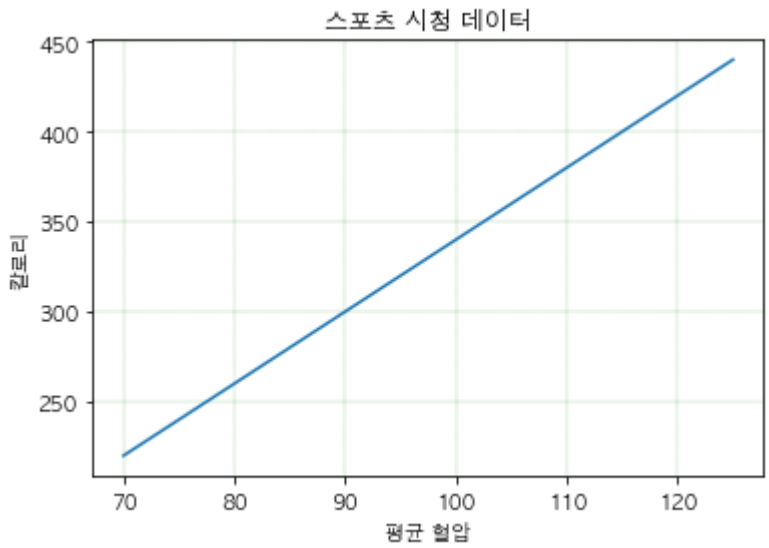
목차로

In [7]:

```
x = np.array([70, 75, 80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([220, 240, 260, 280, 300, 320, 340, 360, 380, 400, 420, 440])

plt.title("스포츠 시청 데이터")
plt.xlabel("평균 혈압")
plt.ylabel("칼로리")

plt.grid(color = 'green', linestyle = '--', linewidth = 0.2)

plt.plot(x, y)
```

Out[7]:

```
[<matplotlib.lines.Line2D at 0x7fdb06b1d340>]
```



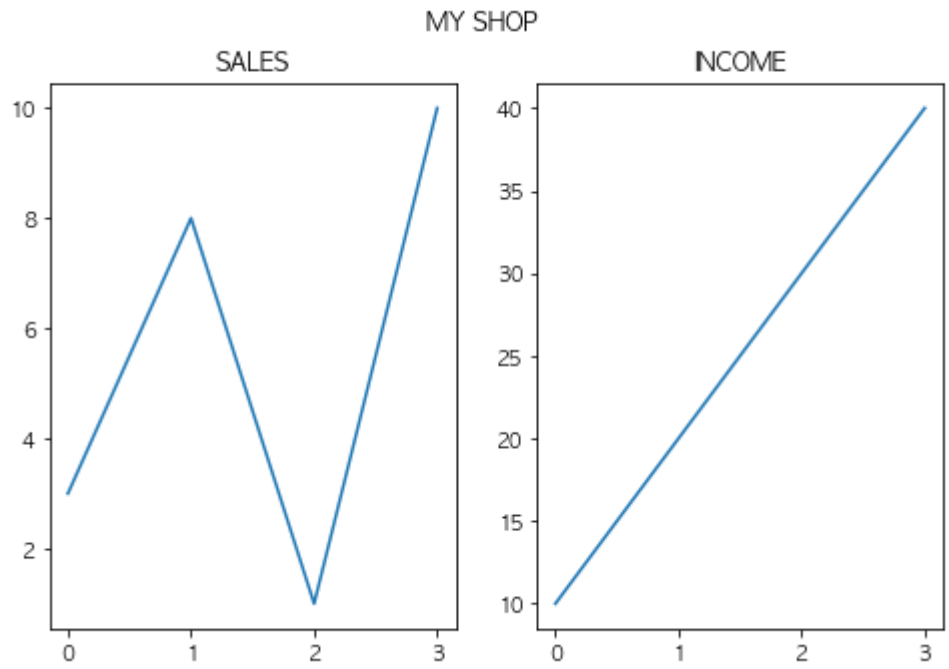
07. 여러개 그래프 전체 타이틀 표시 - SuperTitle

목차로

- 전체 이미지의 상위 타이틀을 subtitle()를 이용하여 제목을 추가할 수 있다.

In [8]:

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

plt.figure(figsize=(8,5))

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

# 1행 2열, 첫번째
plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SALES")

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

# 1행 2열, 두번째
plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("INCOME")

plt.suptitle("MY SHOP")
plt.show()
```

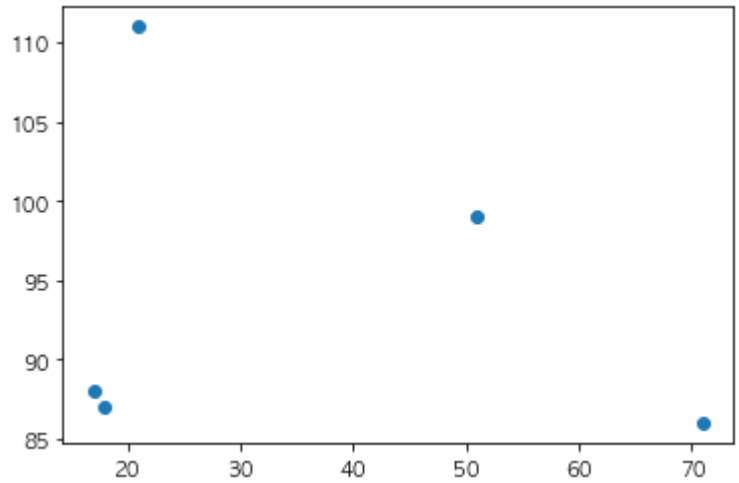


08. Scatter Plot(산점도)

- 직교 좌표계를 활용하여 좌표상의 점들을 표시.
- 두개 변수간의 관계를 나타낼 수 있다.

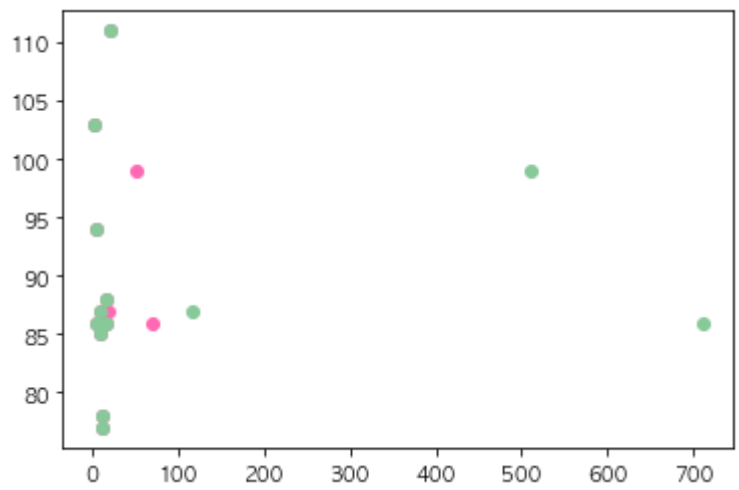
In [9]:

```
x = np.array([51,71,18,17,21])
y = np.array([99,86,87,88,111])

plt.scatter(x, y)
plt.show()
```



In [10]:

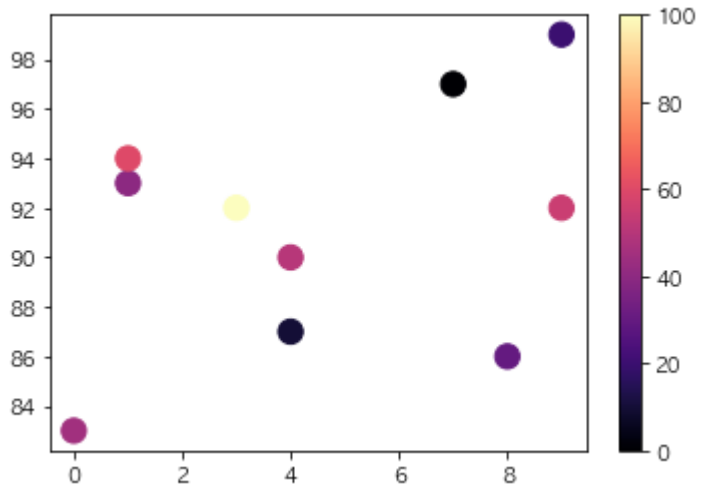
```
x = np.array([51,71,18,17,21,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y, color='hotpink')

x = np.array([511,711,118,17,21,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])

plt.scatter(x, y, color='#88c999')
plt.show()
```



In [11]:

```
x = np.random.randint(10, size=10)
y = np.random.randint(80,100, size=10)
print(x, y)
print( len(x), len(y) )

# 색에 숫자로 맵핑
colors = np.array([0, 10, 20, 30, 40, 45, 50, 55, 60, 100])

# 'viridis', 'plasma', 'inferno', 'magma', 'cividis'
plt.scatter(x, y, c=colors, cmap='magma', s=150)
plt.colorbar()   # colormap 을 포함할 수 있다.

plt.show()
```

```
[7 4 9 8 1 0 4 9 1 3] [97 87 99 86 93 83 90 92 94 92]
10 10
```



사용가능한 colorMaps

- https://www.w3schools.com/python/matplotlib_scatter.asp (https://www.w3schools.com/python/matplotlib_scatter.asp)

09. Alpha(투명도)

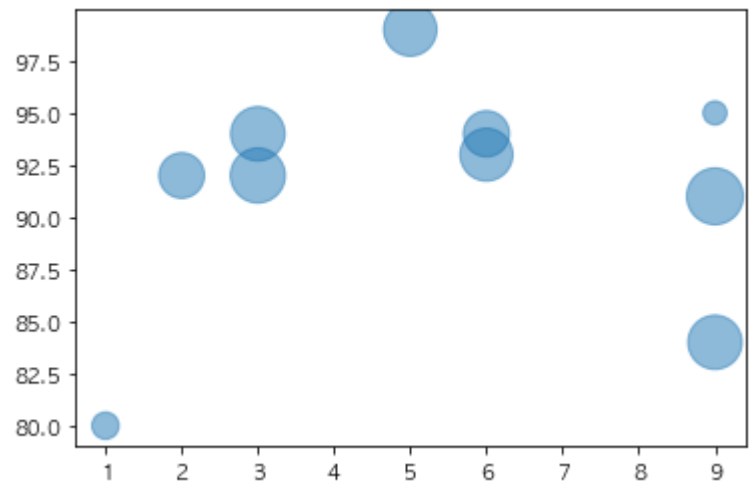
목차로

In [12]:

```
x = np.random.randint(10, size=10)
y = np.random.randint(80,100, size=10)

sizes = np.random.randint(20,800, size=10)
colors = np.random.randint(0,100, size=10)

plt.scatter(x, y, s=sizes, alpha=0.5)
```

Out[12]:

```
<matplotlib.collections.PathCollection at 0x7fdb07125a30>
```

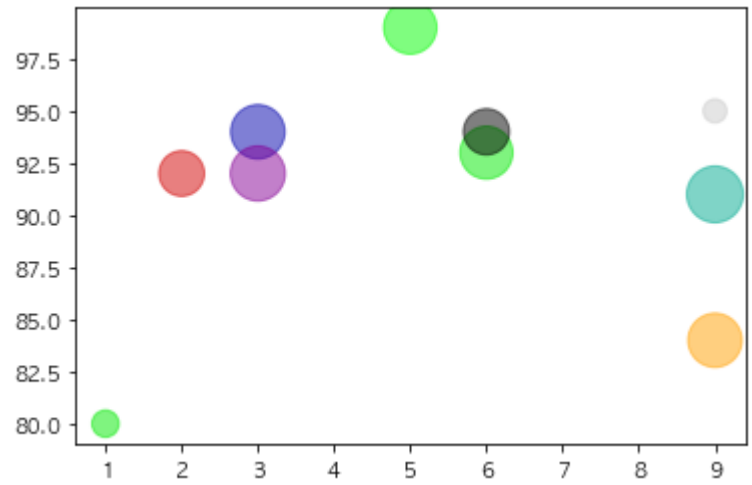


In [13]:

```
plt.scatter(x, y, c=colors, s=sizes, alpha=0.5, cmap='nipy_spectral')
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x7fdb0737b430>
```



10. Barplot(막대그래프)

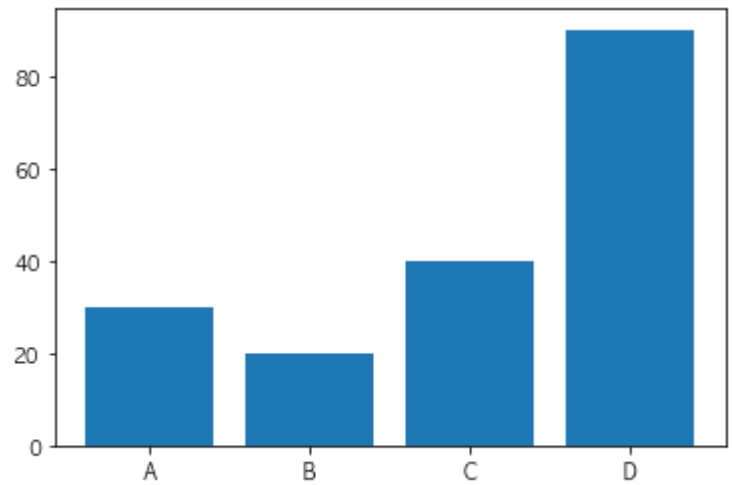
목차로

In [14]:

```
x = np.array(['A', 'B', 'C', 'D'])
y = np.array([30,20,40,90])

plt.bar(x, y)
```

Out[14]:

```
<BarContainer object of 4 artists>
```



11. Horizontal Bars(수평 막대그래프)

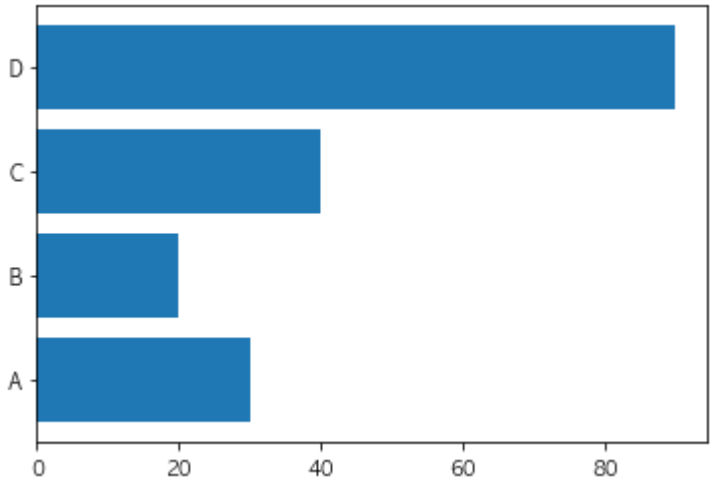
목차로

In [15]:

```
x = np.array(['A', 'B', 'C', 'D'])
y = np.array([30,20,40,90])

plt.barh(x, y)
```

Out[15]:

```
<BarContainer object of 4 artists>
```

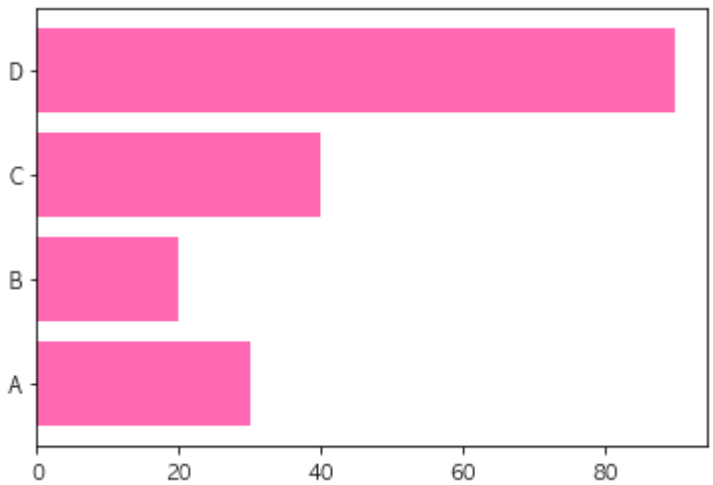


In [16]:

```
plt.barh(x, y, color = 'hotpink')
```

Out[16]:

```
<BarContainer object of 4 artists>
```

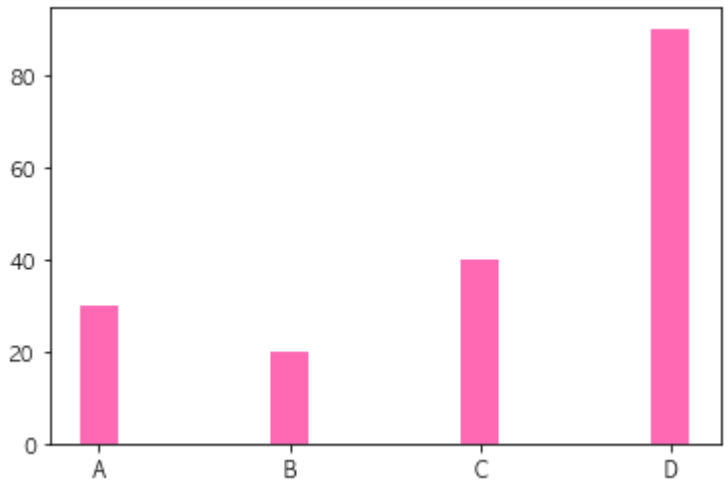


In [17]:

```
plt.bar(x, y, width=0.2, color = 'hotpink')
```

Out[17]:

```
<BarContainer object of 4 artists>
```

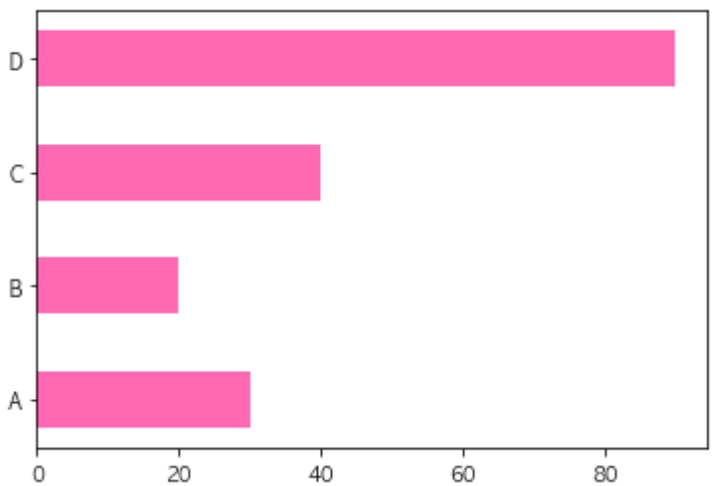


In [18]:

```
plt.barh(x, y, height=0.5, color = 'hotpink')
```

Out[18]:

```
<BarContainer object of 4 artists>
```



12. 히스토그램

- 연속형 값을 표시할 때 사용.
- 가로축이 계급, 세로축이 도수(구간의 값의 개수)를 의미

목차로

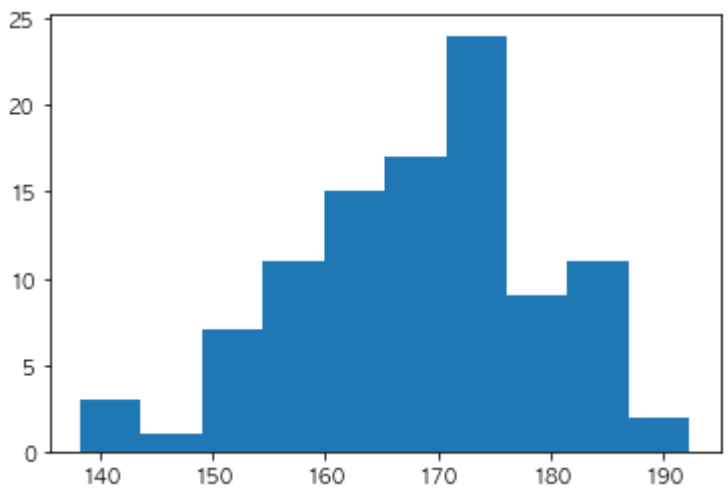
In [19]:

```
# 정규 분포를 따르는 값. 100개 생성
x = np.random.normal(loc=170, scale=10, size=100)  # 평균, 표준편차, 개수

print(x)
plt.hist(x)
```

```
[152.5406993  152.58811219 166.76959243 183.33838664 161.72215482
 164.31916632 174.41604908 183.95258972 183.34108577 185.17155697
 174.76831792 174.87329089 159.12135483 172.64163749 166.09896618
 165.44849559 167.98168884 156.35756634 179.88921524 180.88302276
 160.88313465 190.60350458 171.73780815 175.46629928 170.57714736
 150.64232003 158.93170333 170.97967374 157.06638055 175.66105824
 158.16527288 156.34482532 171.00803999 150.12516201 160.58923085
 167.99826649 167.44786372 154.69702435 192.31377421 166.41765357
 177.69179511 156.63098615 159.26920172 163.81686516 166.43436097
 164.15483121 169.90528477 185.7343065  162.16630254 176.63905717
 160.60979034 174.22197314 173.34943275 172.74195681 164.98158575
 138.2120435  173.80150855 183.65816382 169.86696501 182.8674672
 172.21729225 176.01960578 161.75488245 170.73485095 174.41720368
 154.33285304 169.41266861 180.68768713 184.60892983 184.26019092
 172.89703832 164.27676985 172.83733912 179.78924201 179.99114215
 167.75115666 171.82513025 168.07700456 164.25743337 167.19698076
 140.12876983 171.69165399 157.21956147 173.90478546 163.73584449
 179.74379252 147.60268239 165.80512893 178.96040905 163.29364409
 185.64139413 171.57458163 153.93947777 157.71097783 182.12725977
 170.85633773 165.48416812 161.96785633 143.38263269 152.10832488]
```

Out[19]:

```
(array([ 3.,  1.,  7., 11., 15., 17., 24.,  9., 11.,  2.]),
 array([138.2120435 , 143.62221657, 149.03238964, 154.44256271,
        159.85273578, 165.26290885, 170.67308192, 176.08325499,
        181.49342806, 186.90360114, 192.31377421]),
 <BarContainer object of 10 artists>)
```



13. pie chart(원그래프)

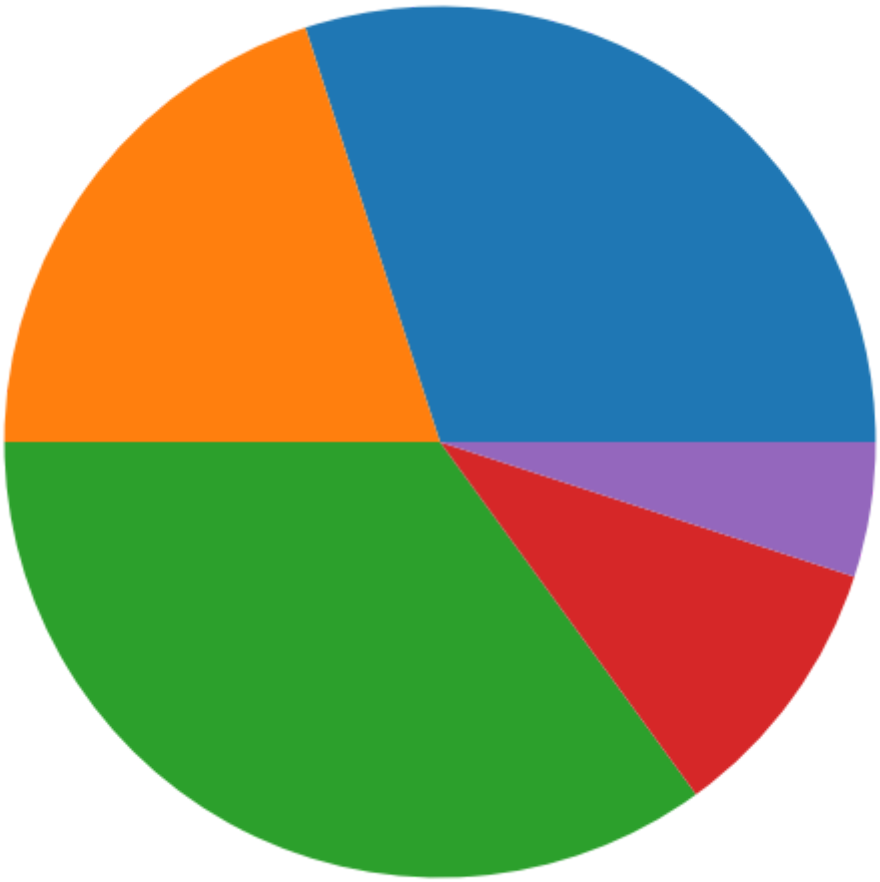
목차로

In [20]:

```
y = np.array([30,20,35,10, 5])
plt.figure(figsize=(10,10))
plt.pie(y)
```

Out[20]:

```
([<matplotlib.patches.Wedge at 0x7fdb07063550>,
  <matplotlib.patches.Wedge at 0x7fdb07063940>,
  <matplotlib.patches.Wedge at 0x7fdb070632b0>,
  <matplotlib.patches.Wedge at 0x7fdb0706f0a0>,
  <matplotlib.patches.Wedge at 0x7fdb0706f520>],
 [Text(0.6465637441936395, 0.8899187180267094, ''),
  Text(-0.8899187482945419, 0.6465637025335369, ''),
  Text(-0.49938947630209474, -0.9801072140121813, ''),
  Text(0.8899187331606258, -0.6465637233635886, ''),
  Text(1.0864571863351944, -0.1720778377961938, '')])
```

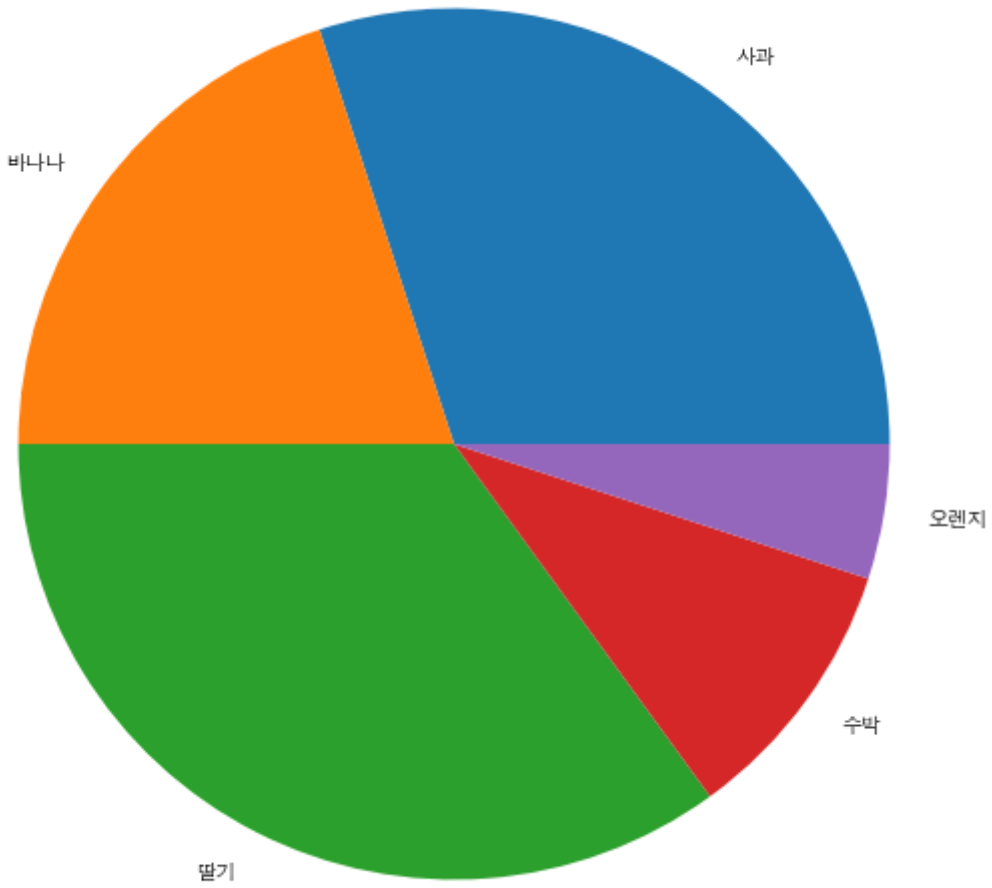


In [21]:

```
plt.figure(figsize=(10,10))

y = np.array([30,20,35,10, 5])
addlbl = ['사과', '바나나', '딸기', '수박', '오렌지']
plt.pie(y, labels=addlbl)
```

Out[21]:

```
([<matplotlib.patches.Wedge at 0x7fdb075abca0>,
  <matplotlib.patches.Wedge at 0x7fdb075b91c0>,
  <matplotlib.patches.Wedge at 0x7fdb075b9640>,
  <matplotlib.patches.Wedge at 0x7fdb075b9ac0>,
  <matplotlib.patches.Wedge at 0x7fdb075b9f40>],
 [Text(0.6465637441936395, 0.8899187180267094, '사과'),
  Text(-0.8899187482945419, 0.6465637025335369, '바나나'),
  Text(-0.49938947630209474, -0.9801072140121813, '딸기'),
  Text(0.8899187331606258, -0.6465637233635886, '수박'),
  Text(1.0864571863351944, -0.1720778377961938, '오렌지')])
```



In [22]:

```
y = np.array([30,20,35,10, 5])
addlbl = ['사과', '바나나', '딸기', '수박', '오렌지']
plt.pie(y, labels = addlbl, startangle = 90)
```

Out[22]:

```
([<matplotlib.patches.Wedge at 0x7fdb06cd0970>,
  <matplotlib.patches.Wedge at 0x7fdb06cd0e80>,
  <matplotlib.patches.Wedge at 0x7fdb06cb52b0>,
  <matplotlib.patches.Wedge at 0x7fdb06cb56d0>,
  <matplotlib.patches.Wedge at 0x7fdb06cb5b80>],
 [Text(-0.8899187180267095, 0.6465637441936395, '사과'),
  Text(-0.6465637025335373, -0.8899187482945414, '바나나'),
  Text(0.9801072140121813, -0.4993894763020948, '딸기'),
  Text(0.6465637233635887, 0.8899187331606258, '수박'),
  Text(0.17207783779619384, 1.0864571863351942, '오렌지')])
```

In [23]:

```
y = np.array([30,20,35,10, 5])
addlbl = ['사과', '바나나', '딸기', '수박', '오렌지']
myexplode = [0.2, 0, 0, 0, 0.3]

plt.pie(y, labels = addlbl, explode = myexplode)
```

Out[23]:

```
([<matplotlib.patches.Wedge at 0x7fdb07b31760>,
  <matplotlib.patches.Wedge at 0x7fdb07b31c40>,
  <matplotlib.patches.Wedge at 0x7fdb07b5e100>,
  <matplotlib.patches.Wedge at 0x7fdb07b5e430>,
  <matplotlib.patches.Wedge at 0x7fdb07b5e8b0>],
 [Text(0.764120788592483, 1.0517221213042929, '사과'),
  Text(-0.8899187482945419, 0.6465637025335369, '바나나'),
  Text(-0.4993894763020947 4, -0.9801072140121813, '딸기'),
  Text(0.8899187331606258, -0.6465637233635886, '수박'),
  Text(1.3827636916993382, -0.21900815719515573, '오렌지')])
```

In [24]:

```
y = np.array([30,20,35,10, 5])
addlbl = ['사과', '바나나', '딸기', '수박', '오렌지']
myexplode = [0.2, 0, 0, 0, 0.3]

plt.pie(y, labels = addlbl, explode = myexplode)
plt.legend()
```

Out[24]:

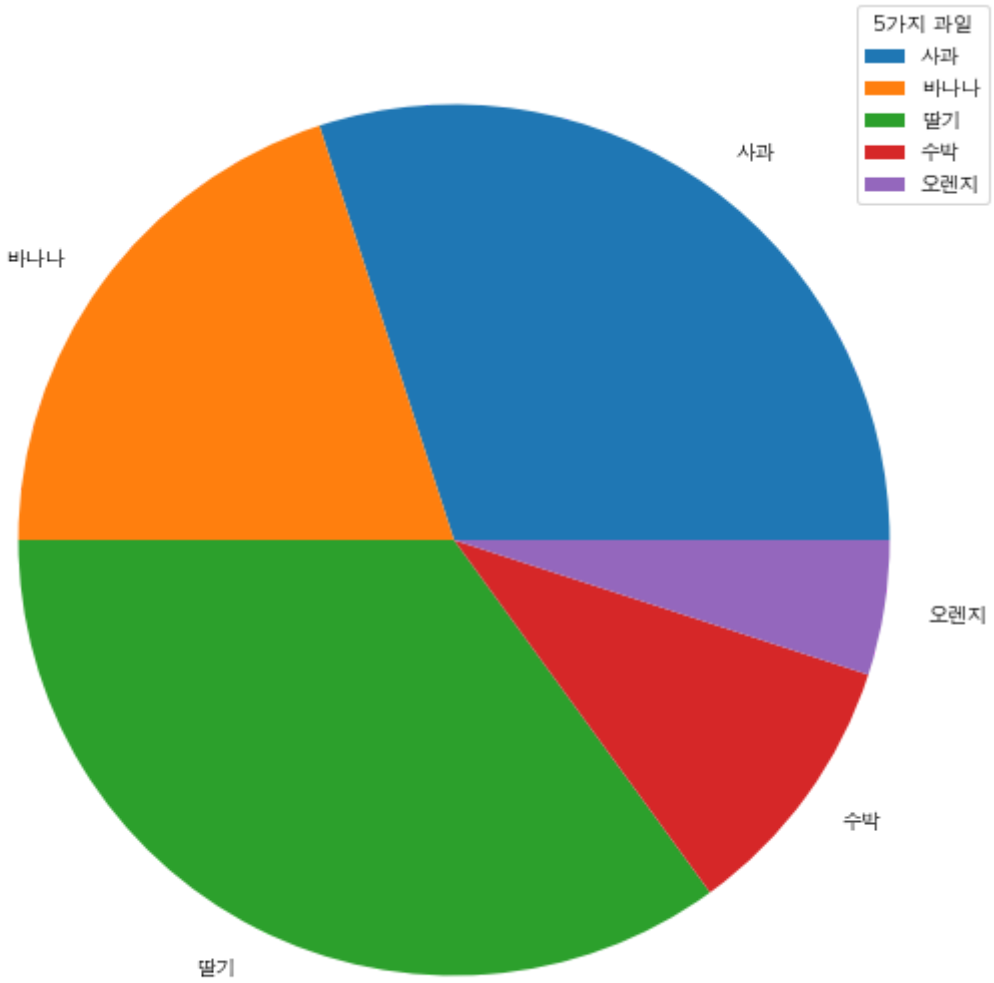
```
<matplotlib.legend.Legend at 0x7fdb07b75c10>
```

In [25]:

```
plt.figure(figsize=(10,10))

y = np.array([30,20,35,10, 5])
addlbl = ['사과', '바나나', '딸기', '수박', '오렌지']
myexplode = [0.2, 0, 0, 0, 0.3]

plt.pie(y, labels = addlbl)
plt.legend(title = "5가지 과일")
```

Out[25]:

```
<matplotlib.legend.Legend at 0x7fdb07b75be0>
```



14. 3D Plots

목차로

- 과거에서는 다음과 같이 썼지만, 이제는 통합되었다.

```
from mpl_toolkits.mplot3d import Axes3D

ax = Axes3D(fig)
```

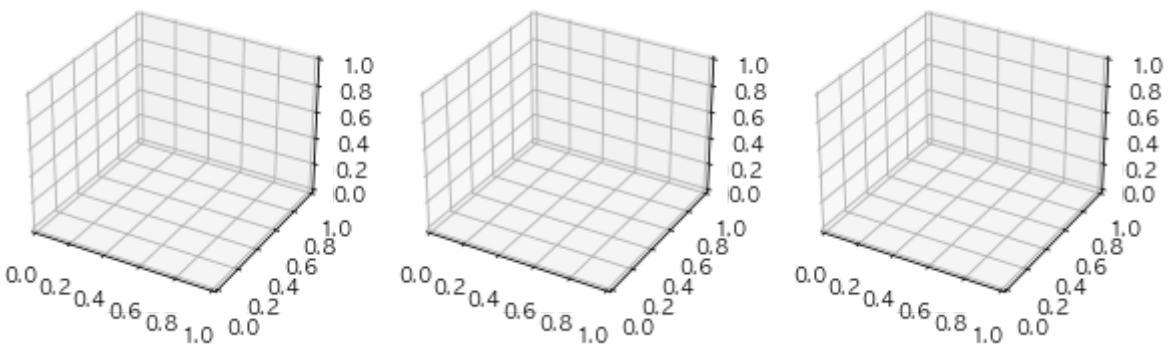
- 현재는 아래와 같이 써도 가능

    `fig.add_subplot(projection='3d')`만으로 `Axes3D`를 사용 가능.

1행 3열의 3D axes를 만들기

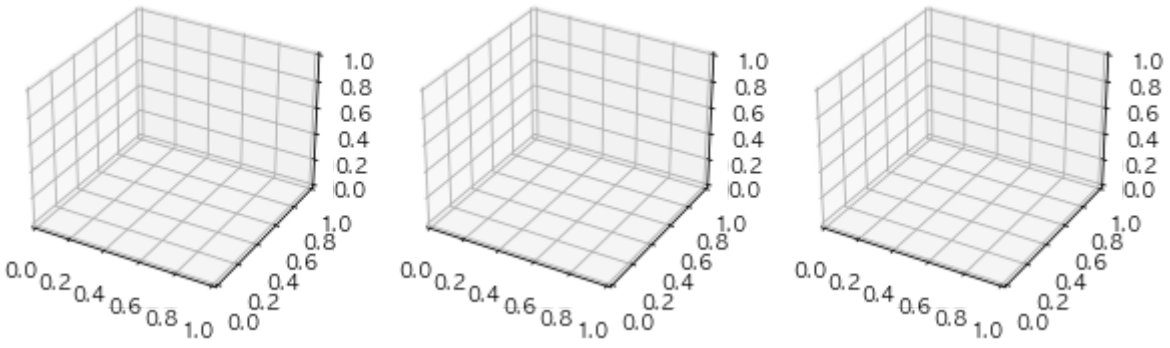
In [26]:

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

fig = plt.figure(figsize=(10, 3))
ax0 = fig.add_subplot(131, projection="3d")
ax1 = fig.add_subplot(132, projection="3d")
ax2 = fig.add_subplot(133, projection="3d")
```



In [27]:

```
# 다음과 같이 좀 더 간단히 가능
fig, axs = plt.subplots(ncols=3, figsize=(10, 3),
                        subplot_kw={"projection":"3d"})
```

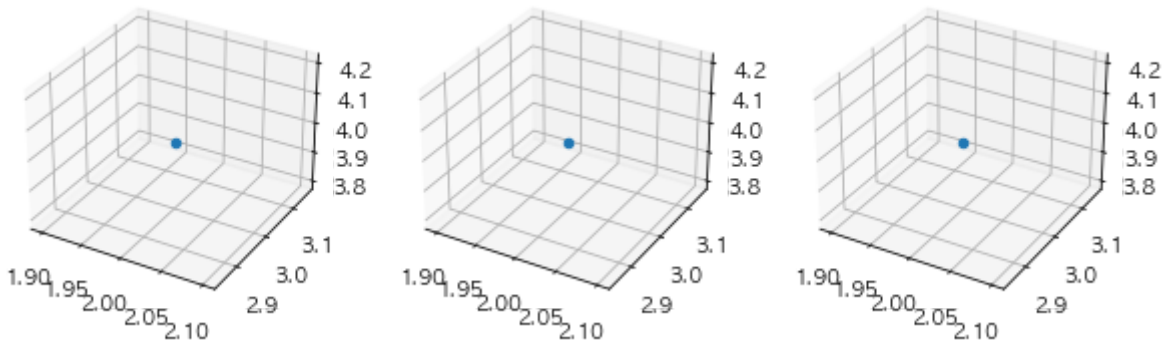


In [28]:

```
# 다음과 같이 좀 더 간단히 가능
fig, axs = plt.subplots(ncols=3, figsize=(10, 3),
                        subplot_kw={"projection":"3d"})

axs[0].scatter(2,3,4)
axs[1].scatter(2,3,4)
axs[2].scatter(2,3,4)
```

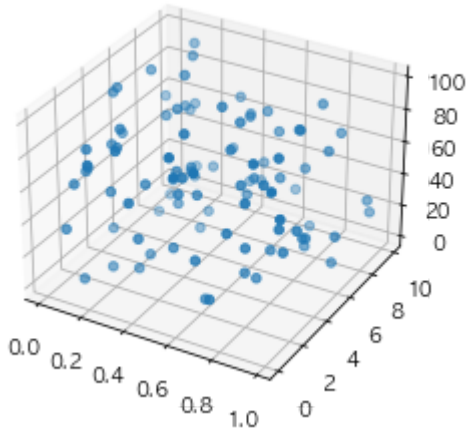
Out[28]:

```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7fdb08232df0>
```



In [29]:

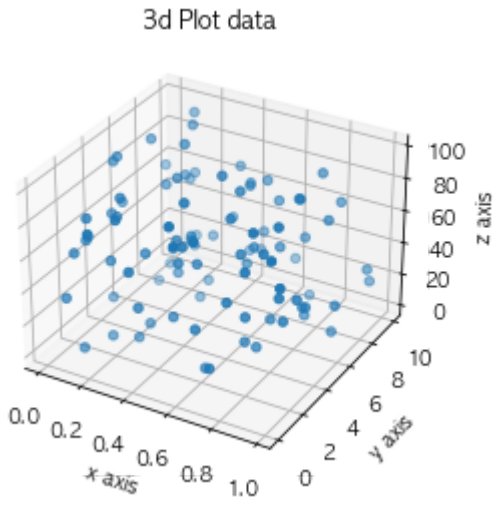
```
np.random.seed(42)

xs = np.random.random(100)
ys = np.random.random(100) * 10
zs = np.random.random(100) * 100

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(xs,ys,zs)
plt.show()
```



title, label 넣기

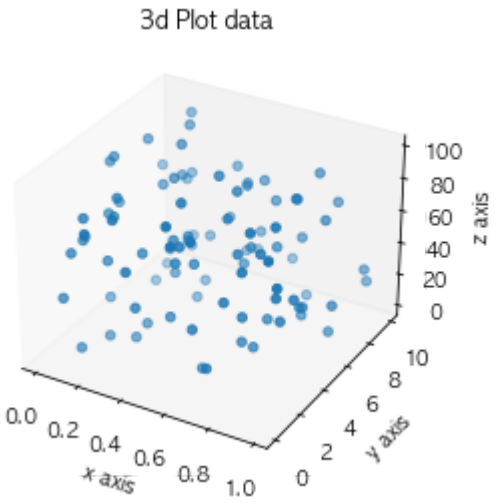
In [30]:

```
np.random.seed(42)

xs = np.random.random(100)
ys = np.random.random(100) * 10
zs = np.random.random(100) * 100

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(xs,ys,zs)

# 제목
ax.set_title("3d Plot data")

# 레이블
ax.set_xlabel("x axis")
ax.set_ylabel("y axis")
ax.set_zlabel("z axis")

plt.show()
```



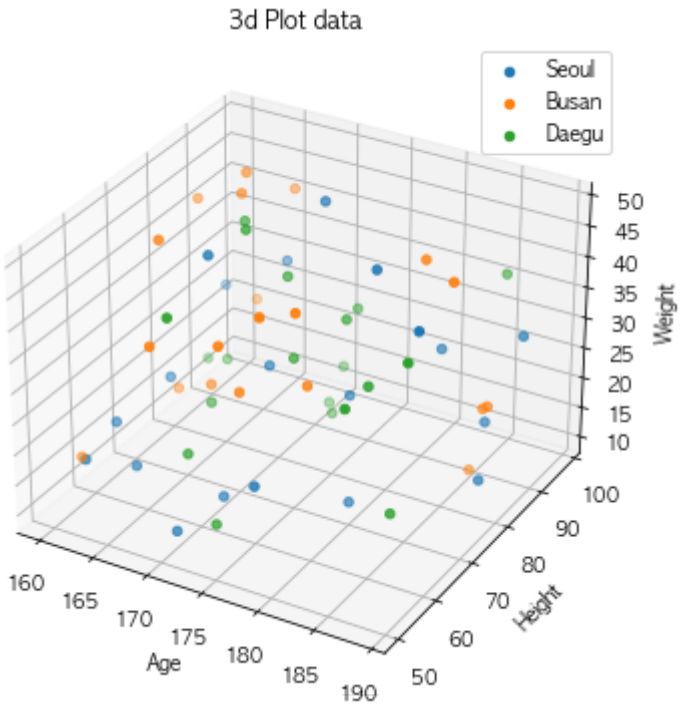
In [31]:

```
np.random.seed(42)

xs = np.random.random(100)
ys = np.random.random(100) * 10
zs = np.random.random(100) * 100

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(xs,ys,zs)

# 제목
ax.set_title("3d Plot data")
ax.grid(False)

# 레이블
ax.set_xlabel("x axis")
ax.set_ylabel("y axis")
ax.set_zlabel("z axis")

plt.show()
```



In [36]:

```
### 범주

labels = ["Seoul", "Busan", "Daegu"]

fig = plt.figure(figsize=(12,6))
ax = fig.add_subplot(111, projection='3d')

for one in labels:
    ages = np.random.randint(low=10, high=50, size=20)
    heights = np.random.randint(low=160, high=190, size=20)
    weights = np.random.randint(low=50, high=100, size=20)

    ax.scatter(xs=heights, ys=weights, zs=ages, label=one)

# 제목
ax.set_title("3d Plot data")

# 레이블
ax.set_xlabel("Age")
ax.set_ylabel("Height")
ax.set_zlabel("Weight")

ax.legend(loc="best")

plt.show()
```



Reference :

- https://www.w3schools.com/colors/colors_names.asp (https://www.w3schools.com/colors/colors_names.asp)
- matplotlib colormap : https://matplotlib.org/3.5.0/tutorials/colors/colormaps.html (https://matplotlib.org/3.5.0/tutorials/colors/colormaps.html)
- 3D Plot : https://likegeeks.com/3d-plotting-in-python/ (https://likegeeks.com/3d-plotting-in-python/)