

코딩으로 공부하는 과학이야기

1차시: 프로그래밍 기초 (2)

서지범 (서울대학교 과학교육과 물리전공)
*jabam1264@snu.ac.kr



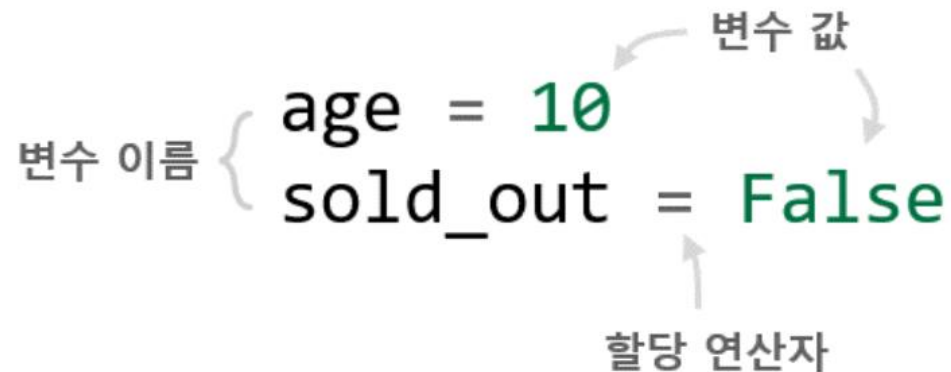
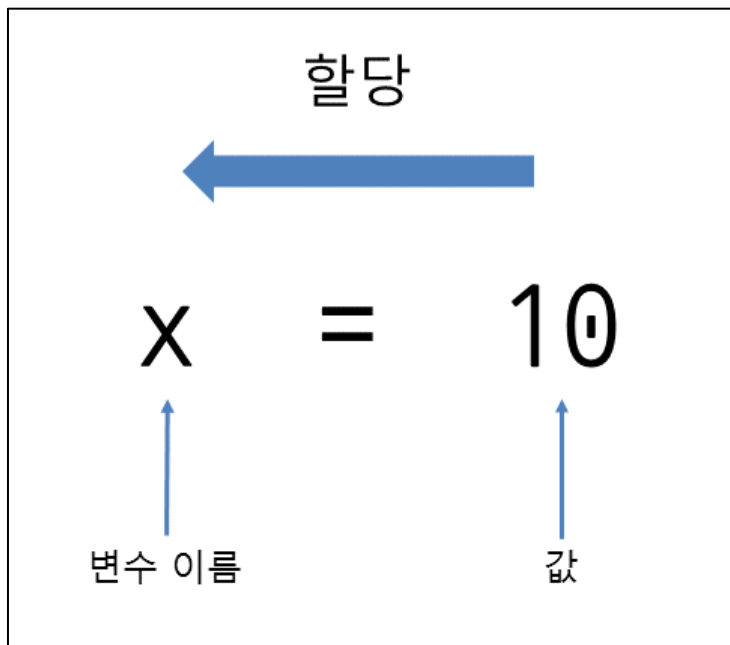
SEOUL NATIONAL UNIVERSITY

1. 기초 파이썬 문법

1. 기초 파이썬 문법

Python 변수 (Variables)

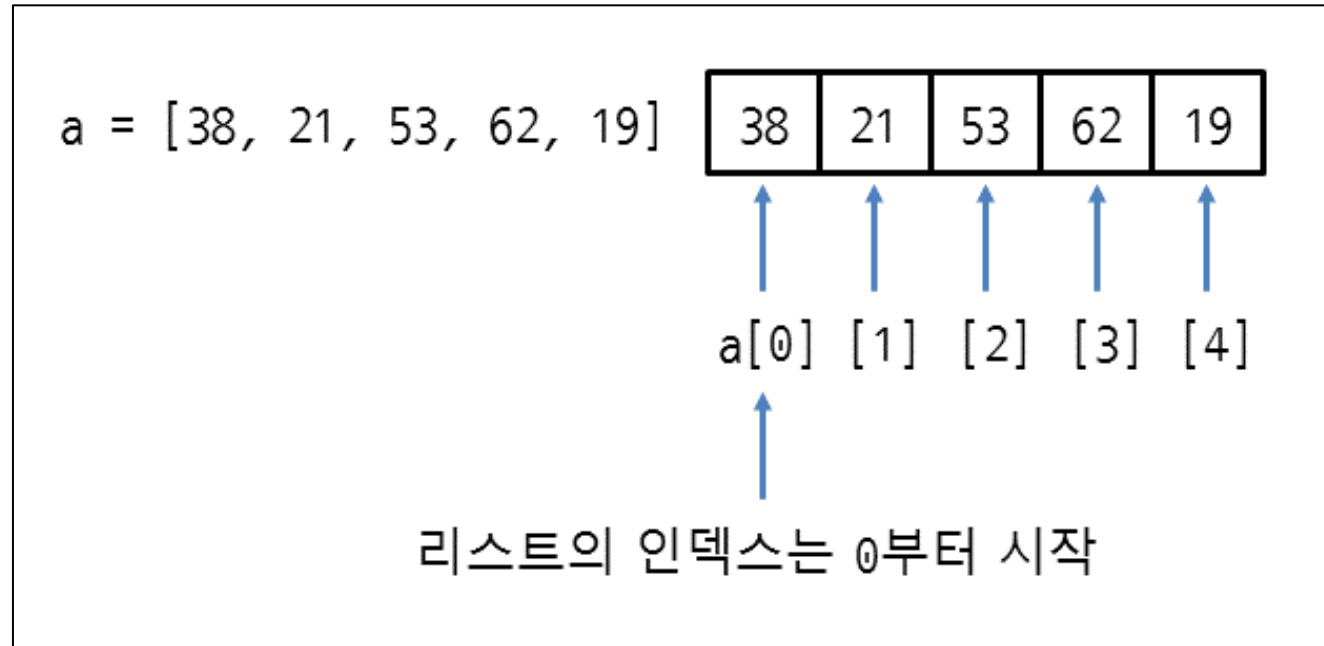
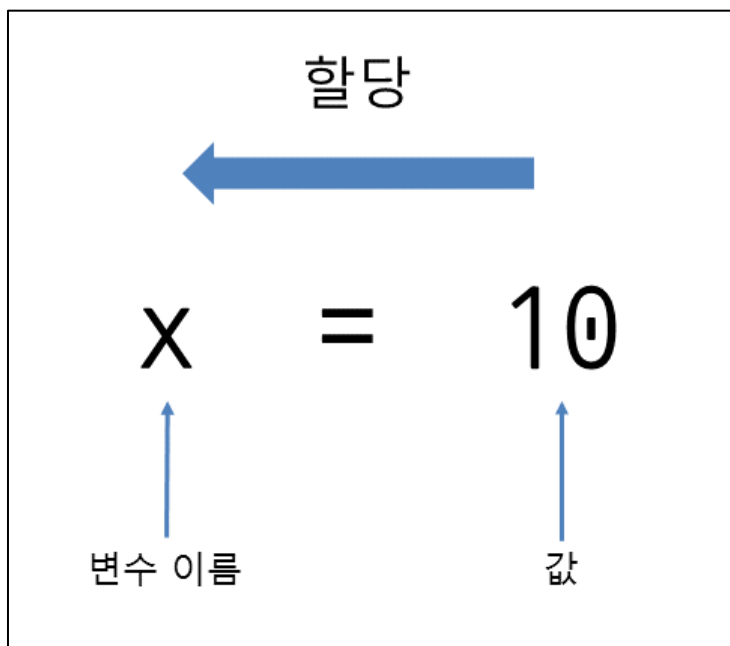
변수 (Variable)는 메모리에 데이터를 저장하는데 사용되는 공간의 이름입니다.



1. 기초 파이썬 문법

Python 변수 (Variables)

변수 (Variable)는 메모리에 데이터를 저장하는데 사용되는 공간의 이름입니다.



1. 기초 파이썬 문법

① 반복 대상에서 변수로 값 가져오기

for 변수 in 반복 대상:

실행할 명령1

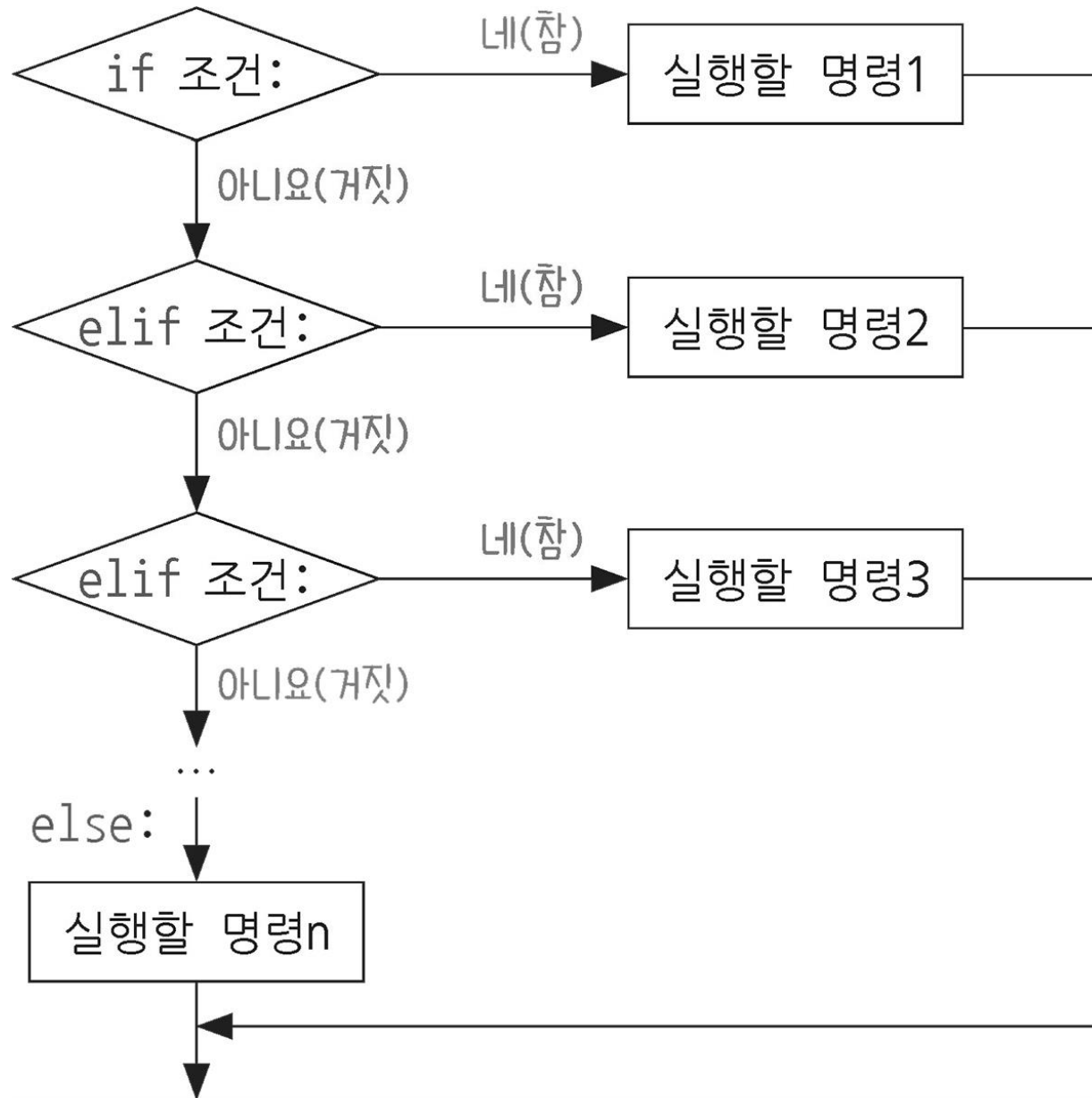
실행할 명령2

...

② 실행

③ 반복 대상으로 돌아가기

1. 기초 파이썬 문법



2. matplotlib

2. matplotlib

- Matplotlib는 Python 프로그래밍 언어 및 수학적 확장 NumPy 라이브러리를 활용한 플로팅 라이브러리이다.

쉽게 설명하면, 그림 그리는 데 필요한 라이브러리



2. matplotlib



Quick start

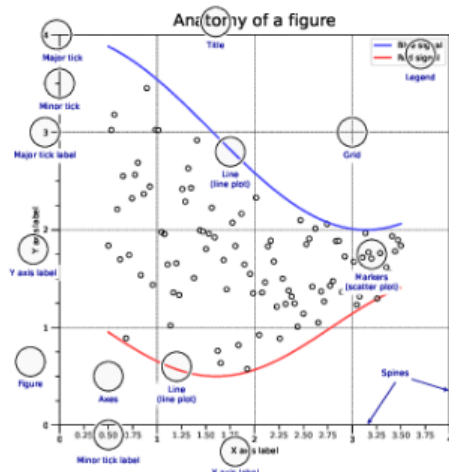
```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
X = np.linspace(0, 2*np.pi, 100)
Y = np.cos(X)
```

```
fig, ax = plt.subplots()
ax.plot(X, Y, color='green')
```

```
fig.savefig("figure.pdf")
plt.show()
```

Anatomy of a figure



Subplots layout

Basic plots

```
plot([X], Y, [fmt], ...)
X, Y, fmt, color, marker, linestyle
```

```
scatter(X, Y, ...)
X, Y, [s]izes, [c]olors, marker, cmap
```

```
bar[h](x, height, ...)
x, height, width, bottom, align, color
```

```
imshow(Z, ...)
Z, cmap, interpolation, extent, origin
```

```
contour[f](X, Y, Z, ...)
X, Y, Z, levels, colors, extent, origin
```

```
pcolormesh([X], [Y], Z, ...)
X, Y, Z, vmin, vmax, cmap
```

```
quiver([X], [Y], U, V, ...)
X, Y, U, V, C, units, angles
```

```
pie(X, ...)
Z, explode, labels, colors, radius
```

```
text(x, y, text, ...)
x, y, text, va, ha, size, weight, transform
```

```
fill[_between](x)(...)
X, Y1, Y2, color, where
```

Advanced plots

```
step(X, Y, [fmt], ...)
X, Y, fmt, color, marker, where
```

```
boxplot(X, ...)
```

Scales

```
ax.set_[xy]scale(scale, ...)
linear any values
log values > 0
symlog any values
logit 0 < values < 1
```

Projections

```
subplot(..., projection=p)
p='polar'
p='3d'
```

```
p=ccrs.Orthographic()
import cartopy.crs as ccrs
```

Lines

```
linestyle or ls
capstyle or dash_capstyle
"butt" "round" "projecting"
```

Markers

Markers: circle, square, triangle, diamond, etc.

markevery: 10, [0, -1], (25, 5), [0, 25, -1]

Colors

Colors: 'C0', 'C1', 'C2', 'C3', 'C4', 'C5', 'C6', 'C7', 'C8', 'C9', 'darkblue', 'firebrick', 'crimson', 'indianred', 'salmon', 'name'

Tick locators

```
from matplotlib import ticker
ax.[xy]axis.set_[minor|major]_locator(locator)
```

```
ticker.NullLocator()
ticker.MultipleLocator(8.5)
ticker.FixedLocator([0, 1, 5])
ticker.LinearLocator(numticks=3)
ticker.IndexLocator(base=8.5, offset=0.25)
ticker.AutoLocator()
ticker.MaxNLocator(n=4)
ticker.LogLocator(base=10, numticks=15)
```

Tick formatters

```
from matplotlib import ticker
ax.[xy]axis.set_[minor|major]_formatter(formatter)
```

```
ticker.NullFormatter()
ticker.FixedFormatter(['zero', 'one', 'two', ...])
ticker.FuncFormatter(lambda x, pos: "[%2f]" % x)
ticker.FormatStrFormatter('>%d')
ticker.ScalarFormatter()
ticker.StrMethodFormatter('x')
ticker.PercentFormatter(xmax=5)
```

Ornaments

```
ax.legend(...)
handles, labels, loc, title, frameon
```

Legend: handles, labels, loc, title, frameon

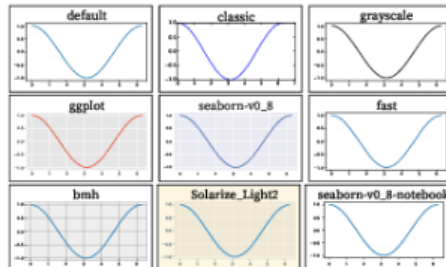
Animation

```
import matplotlib.animation as mpla
```

```
T = np.linspace(0, 2*np.pi, 100)
S = np.sin(T)
line, = plt.plot(T, S)
def animate(i):
    line.set_ydata(np.sin(T+i/50))
anim = mpla.FuncAnimation(
    plt.gcf(), animate, interval=5)
plt.show()
```

Styles

```
plt.style.use(style)
```



Quick reminder

```
ax.grid()
ax.set_[xy]lim(vmin, vmax)
ax.set_[xy]label(label)
ax.set_[xy]ticks(ticks, [labels])
ax.set_[xy]ticklabels(labels)
ax.set_title(title)
ax.tick_params(width=10, ...)
ax.set_axis_[on|off]()
```

```
fig.suptitle(title)
fig.tight_layout()
plt.gcf(), plt.gca()
mpl.rcParams['axes', linewidth=1, ...]
[fig|ax].patch.set_alpha(0)
text=r'$\frac{-e^{i\pi}}{2^n}$'
```

3. 그래프 소개

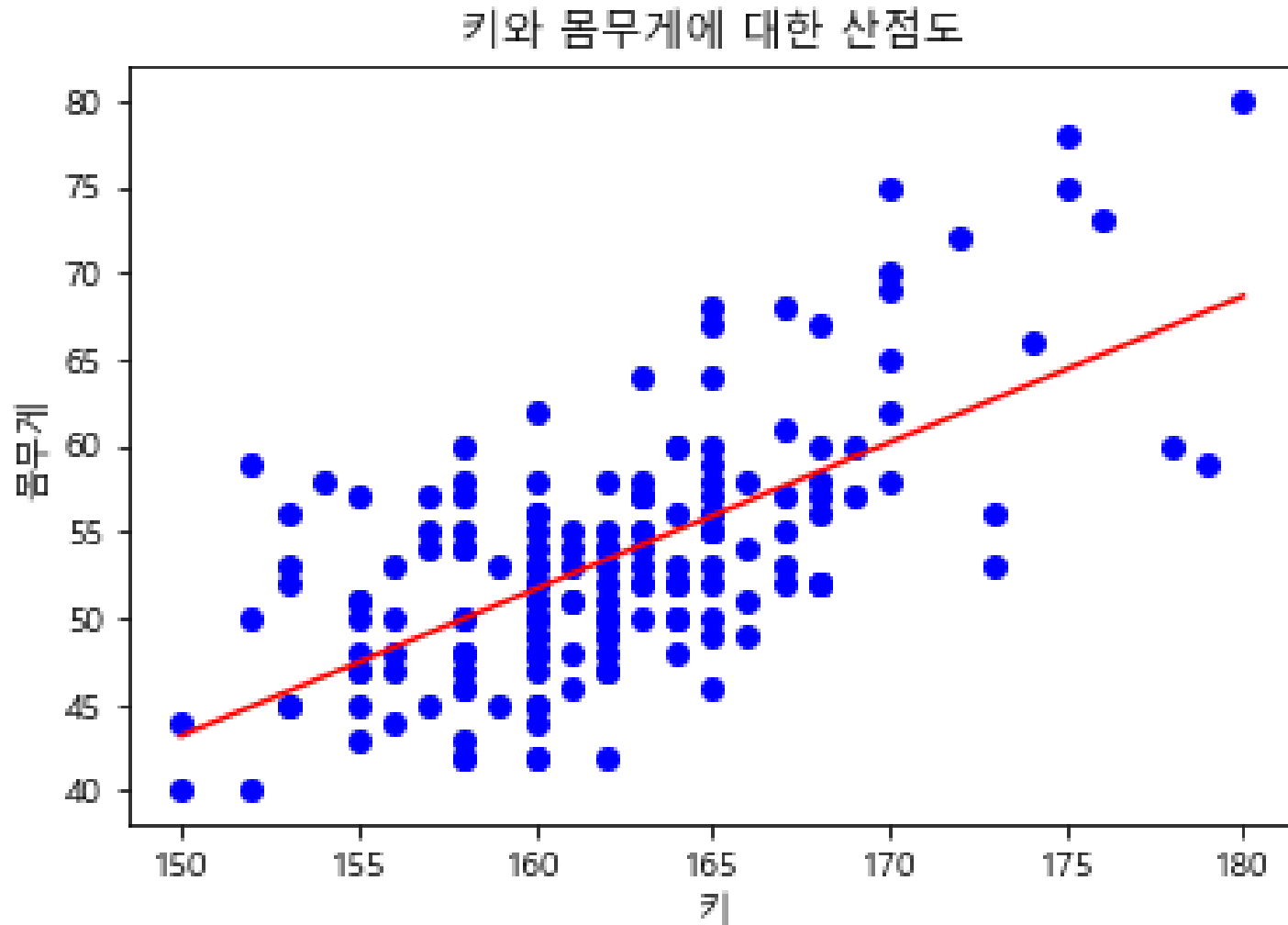
3. 그래프 소개

- 선 그래프 (Line plot): 시간에 따른 데이터 변화 확인 가능



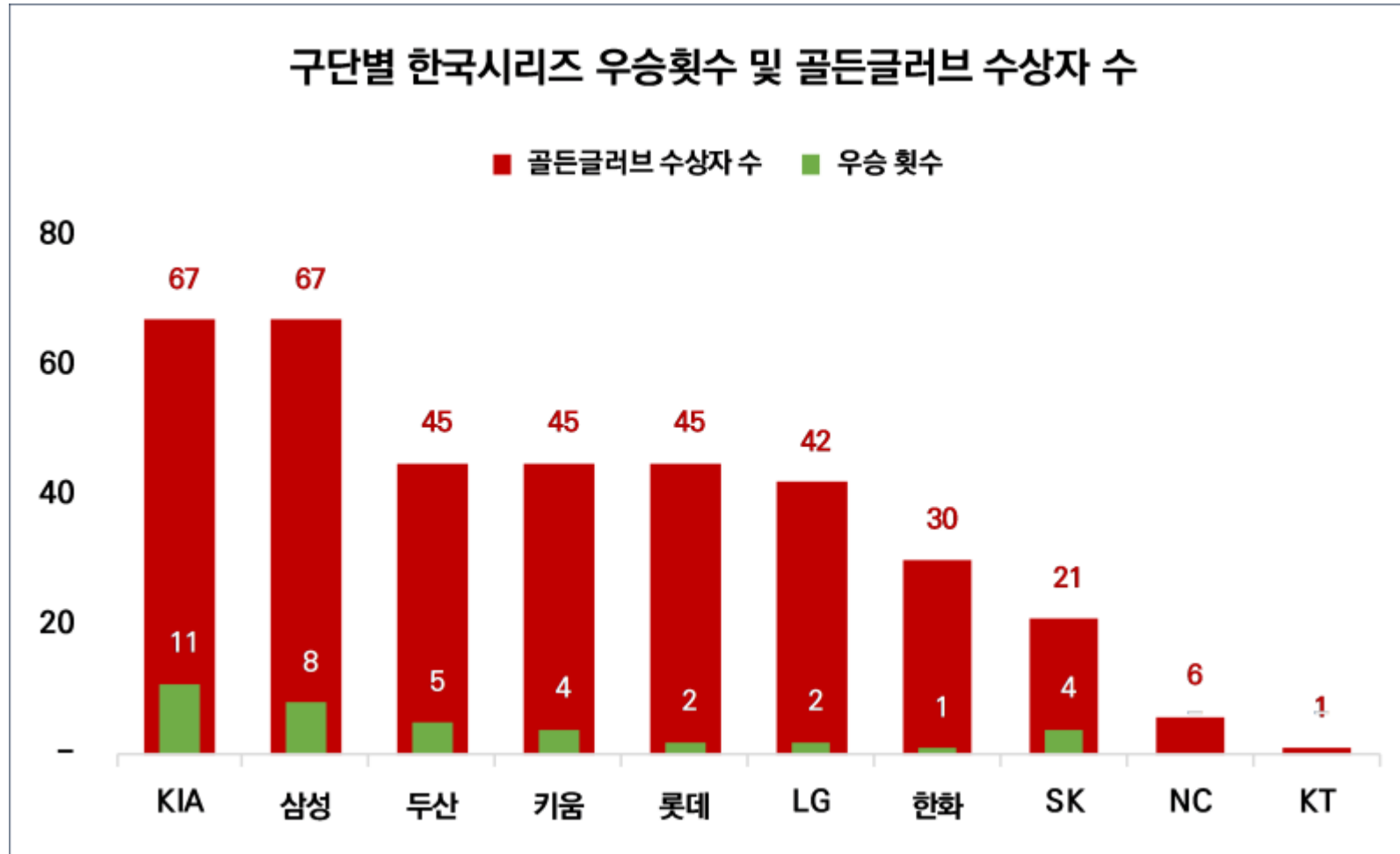
3. 그래프 소개

- 산점도 (Scatter plot): 두 변수 간 관계 탐색



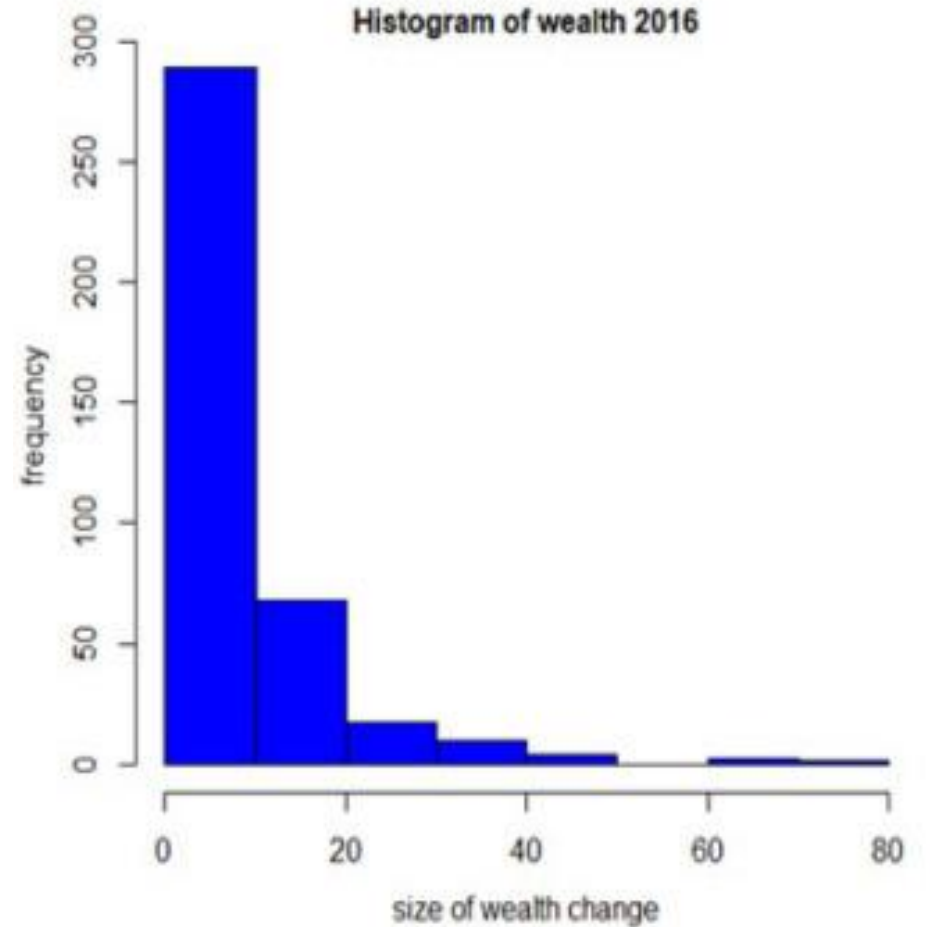
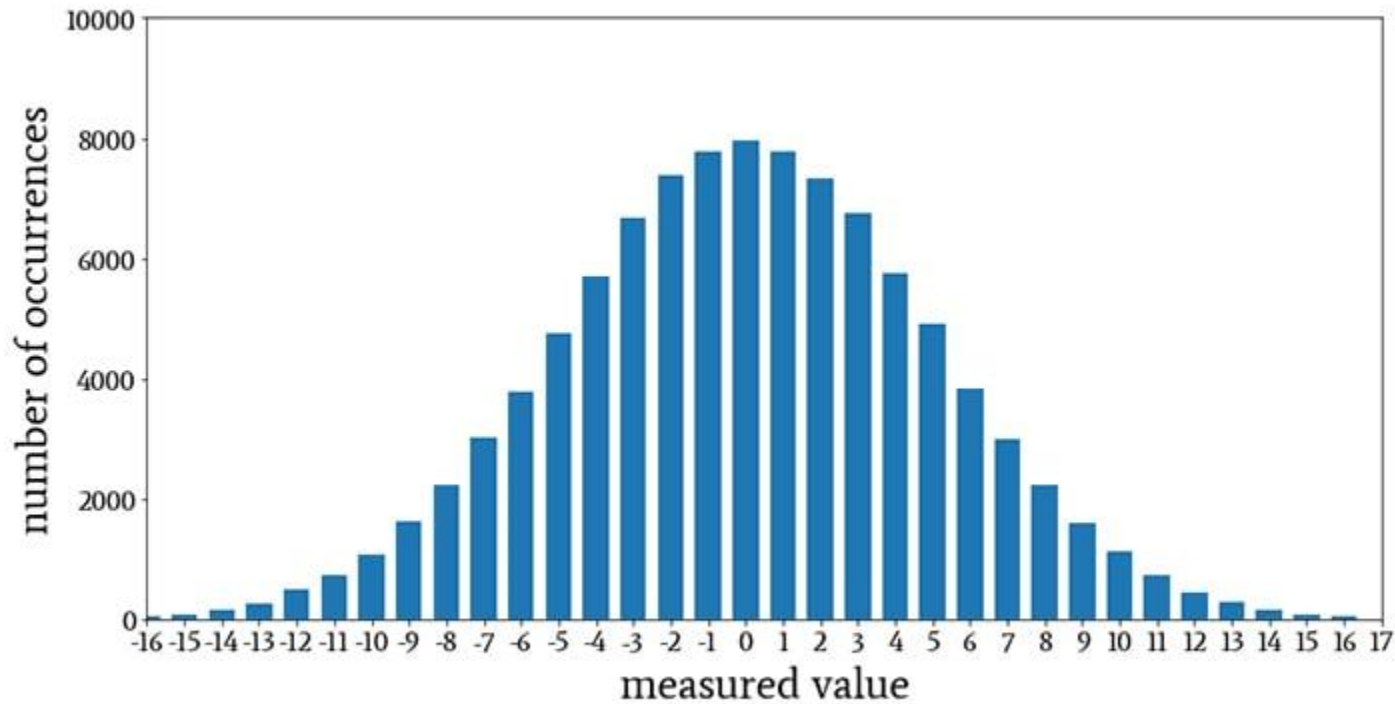
3. 그래프 소개

- 막대 그래프 (Bar Chart): 카테고리별 비교



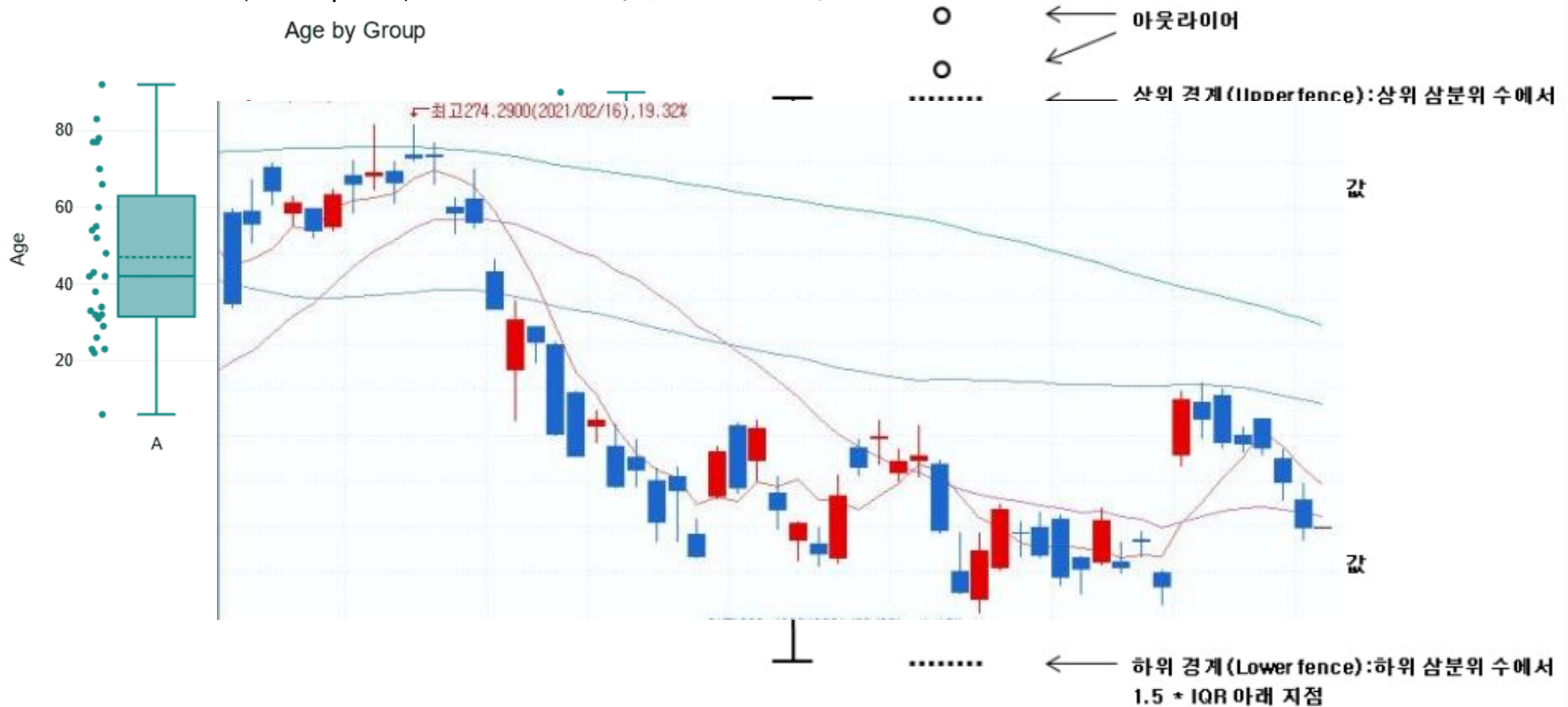
3. 그래프 소개

- 히스토그램 (Histogram): 데이터 분포 확인



3. 그래프 소개

- 상자 그림 (Box plot): 분포 요약 및 이상치 확인





Thank you