K-Digital Training

codingOn

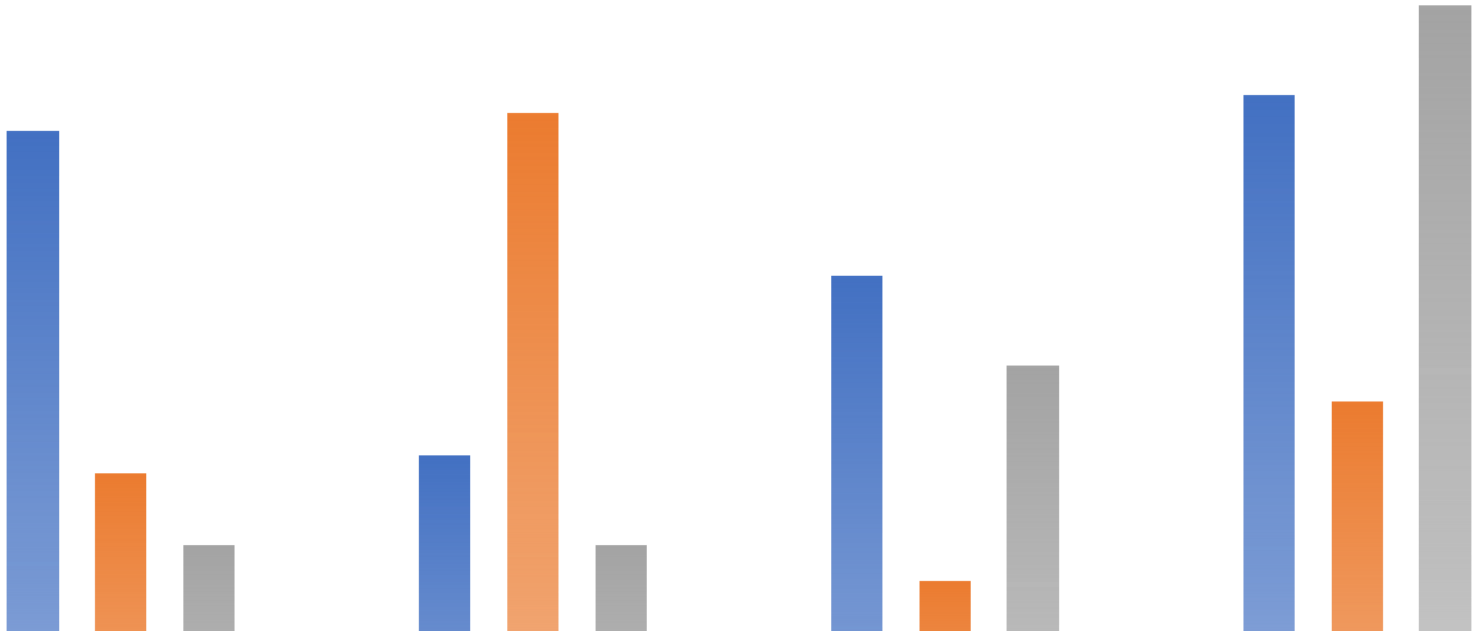
Matplotlib

codingOn

데이터 시각화

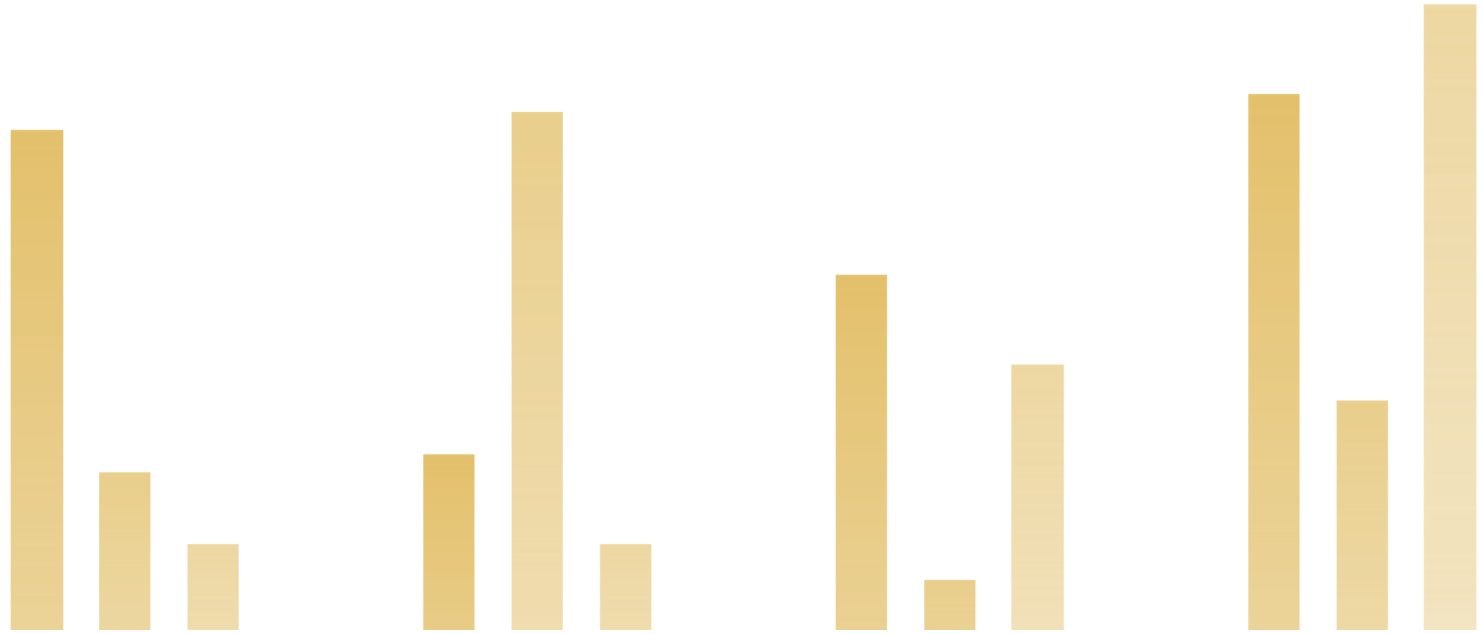
데이터 시각화

- 그래프, 차트, 다이어그램 등 다양한 시각화 도구를 사용하여 데이터를 시각적으로 표현한 것
- 데이터 시각화는 복잡한 데이터 집합을 직관적이고 이해하기 쉬운 형태로 변환하여 데이터의 패턴, 추세, 상관 관계 등을 파악하는데 유리함



데이터 시각화의 목적

- 데이터의 패턴과 추세 파악: 시각화를 통해 데이터의 패턴, 추세, 이상치 등을 식별하고 분석
- 데이터 간 관계 이해: 다양한 변수 간의 관계와 상관 관계를 시각적으로 이해
- 인사이트 도출: 시각화를 통해 데이터에서 의미 있는 인사이트와 통찰력을 도출

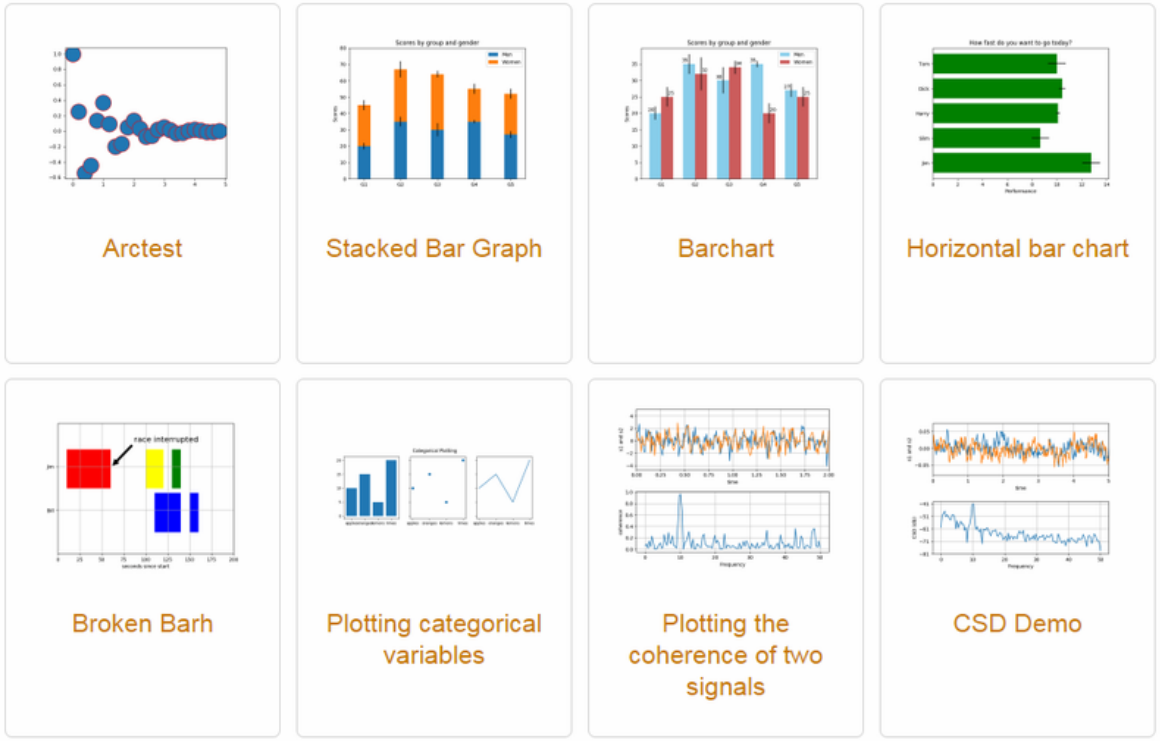


codingOn

Matplotlib

Matplotlib

- 파이썬에서 가장 널리 사용되는 데이터 시각화 라이브러리
- 그래프, 차트, 플롯 등 다양한 시각화 요소를 생성하고 데이터를 시각적으로 나타낼 수 있음
- [공식홈페이지]



출처: wikidocs

기본 그래프 그리기

codingOn

- import matplotlib.pyplot as plt
- pyplot? 맷플롯립의 기능을 간편하게 사용할 수 있는 함수제공
- 그래프 그리기
  - plot(x좌표, y좌표, [옵션 매개변수])
- 그래프 표시
  - show() : 그래프 표시

```
import matplotlib.pyplot as plt

# 데이터 정의
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# 선 그래프 생성
plt.plot(x, y)

# 그래프 출력
plt.show()
```

한글 설정

- 맷플롯립은 기본적으로 한글을 사용할 수 없음
- 한글을 사용하려면 폰트 설정을 해줘야함

```
import matplotlib.pyplot as plt
from matplotlib import font_manager

# 설치된 폰트 확인
# 모든 폰트 경로 출력
font_list = font_manager.findSystemFonts(fontpaths=None, fontext='ttf')
# 특정 폴더 폰트 출력
font_list = font_manager.findSystemFonts(fontpaths=["/path/to/fonts"], fontext='ttf')
print(font_list)

path = "/Users/Library/Fonts/GmarketSansTTFMedium.ttf"
font = font_manager.FontProperties(fname=path).get_name()
plt.rc('font', family=font)
```

기본 그래프 그리기

codingOn

- 그래프 꾸미기
- color : 그래프 색상 (예: color="red" or "r")
- linestyle : '-', '--' 등 선 스타일, 스타일 더보기
- linewidth : 선 두께 (예: linewidth=2)

```
plt.plot(x, y, color="#f3f3f3", linestyle='--', linewidth=2 )
```

기본 그래프 그리기

codingOn

- 범례 표시하기
- label : 범례 표시 명 (예 : label = “그래프”)
- legend() 함수 추가
- legend() 함수의 매개변수
  - loc : 위치 (예: loc=“upper center”)
  - title : 범례명
  - frameon : 박스표시여부
  - fontsize : 범례 텍스트 크기

```
plt.plot(x, y, label="그래프")

plt.legend(loc="upper center", fontsize=15, title="범례명", frameon=False)
```

기본 그래프 그리기

codingOn

- 마커 표시 하기
- marker : 마커생성 (예: ‘o’, ‘s’, ‘*’) 등
- markersize : 마커크기 변경
- markerfacecolor : 마커 내부 색상 변경
- markeredgecolor : 마커 외곽선 색상 변경

```
plt.plot(x, y, marker="*", markersize=30, markerfacecolor="white", markeredgecolor="red")
```

기본 그래프 그리기

codingOn

- 그래프 제목 넣기
- title(“타이틀명”, [옵션])
- 매개변수
  - pad : 간격지정
  - fontdict : 폰트 꾸미기(딕셔너리형태)

```
plt.title('Matplotlib 그래프 그리기', fontsize=20,  pad=20, backgroundcolor="red", color="#ffffff")
```

기본 그래프 그리기

codingOn

- 축 레이블 생성
- xlabel("축이름", [옵션]) : x축
- ylabel("축이름", [옵션]) : y축
- 매개변수
  - 폰트 설정과 동일
  - labelpad : 간격지정

```
plt.xlabel('X-축', fontsize=20, color="red")
plt.ylabel('Y-축', labelpad=20, backgroundcolor="blue")
```

기본 그래프 그리기

codingOn

- 그리드 생성
- grid(True, axis, [옵션])
- axis : ‘x’, ‘y’, ‘both’
- 매개변수
  - 라인스타일지정
  - color

```
plt.grid(True, axis="both", color='gray', linestyle='--', linewidth=0.5)
```

codingOn

Colors

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

Line Styles

character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style

Markers

character	description
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

축 범위 설정

- xlim([xmin, xmax]) : x축 범위 설정
- ylim([ymin, ymax]) : y축 범위 설정
- axis([xmin, xmax, ymin, ymax]) : 축범위 설정
- axis(“equal”) : x축과 y축을 동일하게 유지
- axis(“scaled”) : 데이터의 비율에 따라 축 스케일 설정
- axis(“tight”) : 데이터가 그래프의 모든영역을 채우도록 설정
- axis(“auto”) : 자동을 축설정

그래프 여러개 표시

- subplot(nrows, ncols, index)
  - nrows: 전체 플롯이 배치될 행(row) 개수
  - ncols: 전체 플롯이 배치될 열(column) 개수
  - index: 플롯이 배치될 위치 (왼쪽에서 오른쪽, 위에서 아래로 번호 매김)
- tight_layout() : 간격 조정
- suptitle() : 전체 그래프의 제목

막대 그래프 그리기

codingOn

- bar(x축값, y축값, [옵션])
- 매개변수
  - width: 막대의 너비를 설정합니다. 기본값은 0.8
  - bottom: 막대가 시작하는 y축 위치를 설정
  - align: 막대의 정렬 방식. 'center' or 'edge'
  - color: 막대의 색상을 설정. 단일 색상 문자열이나 색상 배열을 사용
  - edgecolor: 막대 테두리의 색상을 설정합니다.
  - label: 범례에 사용할 텍스트를 지정합니다.

```
categories = ['A', 'B', 'C']
values = [10, 15, 7]

plt.bar(categories, values, width=0.5, align="edge", color=['r', 'g', 'b'])
```

막대 그래프 그리기

codingOn

- 눈금표시하기
- xticks(x값, 눈금값)
- yticks(y값, 눈금값)

```
plt.xticks(categories, ['2023', '2024', '2025'])
```

수평 막대 그래프 그리기

codingOn

- barh(y축값, x축값, [옵션])

```
categories = ['A', 'B', 'C']
values = [10, 15, 7]

plt.barh(categories, values, color='skyblue', edgecolor='gray')
```

히스토그램

- 데이터의 분포를 시각적으로 분석할 때 사용
- 특정 값들이 얼마나 자주 발생하는지(빈도)를 구간별로 나타내는 데 적합
- 예)
- 통계 분석: 시험 점수, 매출 데이터 분포
- 의료 데이터 분석: 환자의 혈압이나 체온 분포
- 경제 데이터 분석: 주택 가격, 소득 분포
- 환경 데이터 분석: 온도, 습도 등의 측정값 분포

히스토그램 그래프 그리기

codingOn

- hist(배열, [옵션값])
- 매개변수
  - bins: 데이터를 나눌 구간 수 또는 구간 경계를 지정(정수:기본10), 배열: 구간 경계를 명시적으로 지정
  - (예: bins=5 또는 bins=[1, 2, 3, 4, 5])
  - range: 히스토그램 구간의 최소값과 최대값을 지정 (예: range=(1, 5))
  - density: True로 설정하면 빈도 대신 확률 밀도를 표시. 확률 밀도는 각 구간의 면적 합이 1이 되도록 조정된 값
  - cumulative: True로 설정하면 누적 히스토그램을 생성 (예: 각 구간까지의 합계)
  - histtype: 히스토그램의 스타일을 지정합니다.

히스토그램 그래프 그리기

codingOn

- 매개변수
  - histtype: 히스토그램의 스타일을 지정
    - 'bar': 기본 막대 그래프
    - 'barstacked': 스택된 막대 그래프
    - 'step': 계단 그래프
    - 'stepfilled': 채워진 계단 그래프

```
data = [1, 2, 2, 3, 3, 3, 4, 5, 5, 5, 5, 6]
plt.hist(data, bins=5, histtype="step", color='skyblue', alpha=0.7)
```

산점도 그리기

codingOn

- scatter(x, y, s=None, c=None, marker=None, cmap=None, alpha=None)
- 매개변수
  - s: size(기본값20)
  - c: color(단일색상 또는 배열사용)
  - cmap: 색상 배열을 사용할 때 컬러맵을 지정
    - viridis, plasma, inferno, coolwarm

```
# 산점도 그리기
plt.scatter(x, y, s=sizes, c=colors, cmap='viridis', alpha=0.7, edgecolors='black', linewidths=1.5)
plt.colorbar(label='Color Scale')  # 색상 바 추가
```

파이차트 그리기

codingOn

- pie(x, [옵션])
- 매개변수
  - labels : 각 섹션의 이름을 정의. ['A', 'B', 'C']처럼 문자열 리스트
  - colors: 각 섹션의 색상을 지정
  - autopct: 섹션의 비율을 차트에 표시. '%1.1f%%' 형식으로 전달하면 소수점 한 자리까지 비율을 표시함
  - startangle: 시작 각도를 설정.기본값은 0도. (예를 들어 90을 주면 위쪽에서 시작)
  - explode: 특정 섹션을 강조하기 위해 섹션을 중심에서 떨어뜨림. [0.1, 0, 0, 0]처럼 배열로 전달합니다.

파이차트 그리기

codingOn

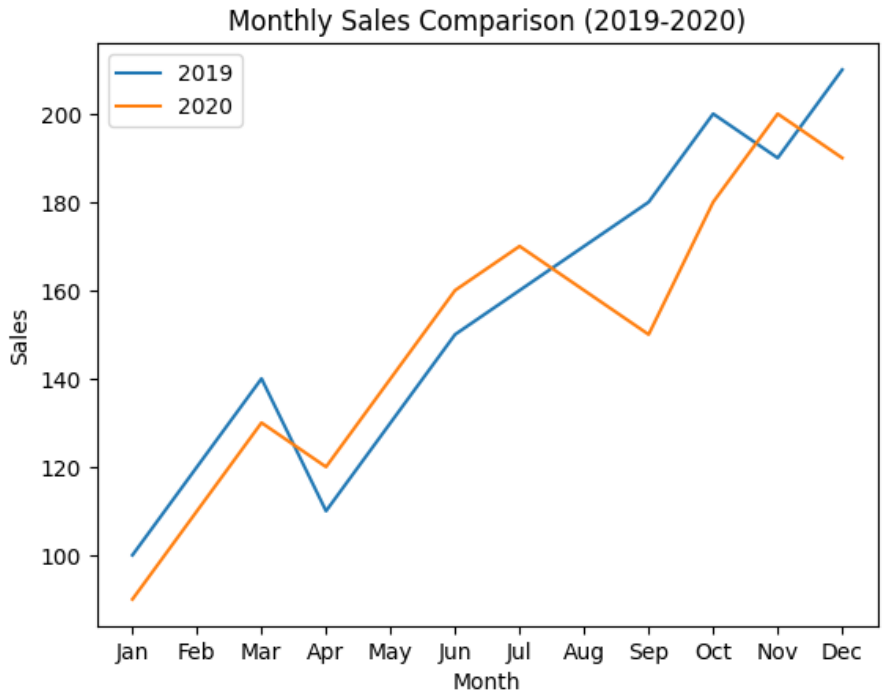
- 매개변수
  - shadow: 파이 차트에 그림자를 추가. (True 또는 False)
  - radius: 파이 차트의 반지름 크기를 설정. (기본값은 1.0)
  - textprops: 텍스트 스타일을 설정하는 데 사용. (예: {'fontsize': 12, 'color': 'blue'})
  - counterclock: 섹션이 시계 반대 방향으로 그려질지 결정. (True 또는 False)
  - wedgeprops: 각 섹션의 테두리 속성을 설정. (예: {'edgecolor': 'black', 'linewidth': 2})

```
sizes = [25, 35, 20, 20]
labels = ['A', 'B', 'C', 'D']

plt.pie(sizes, labels=labels)
plt.show()
```

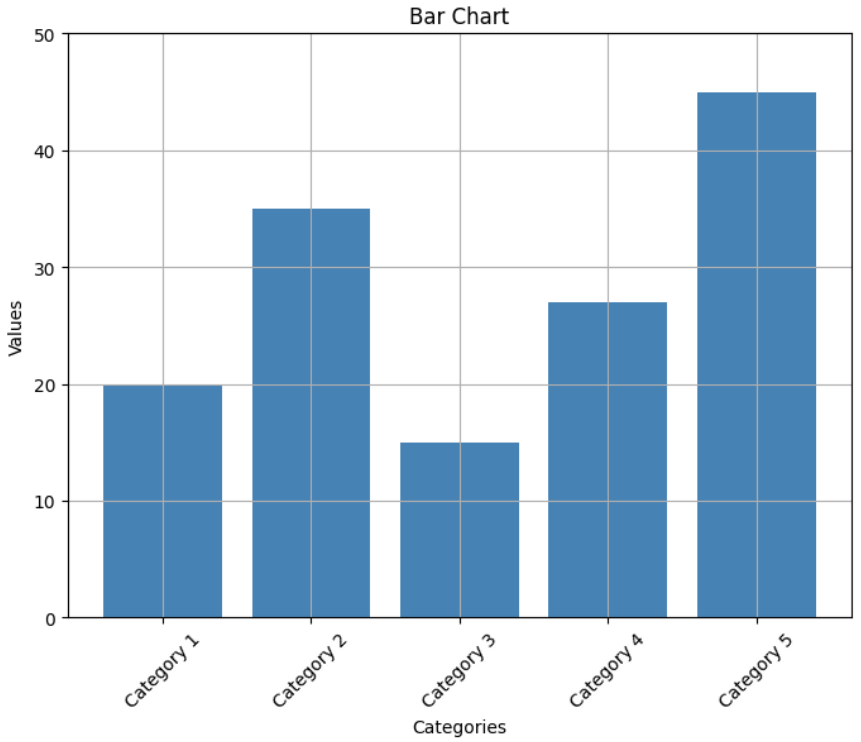
실습1. 그래프 그리기

```
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
sales_2019 = [100, 120, 140, 110, 130, 150, 160, 170, 180, 200, 190, 210]
sales_2020 = [90, 110, 130, 120, 140, 160, 170, 160, 150, 180, 200, 190]
```

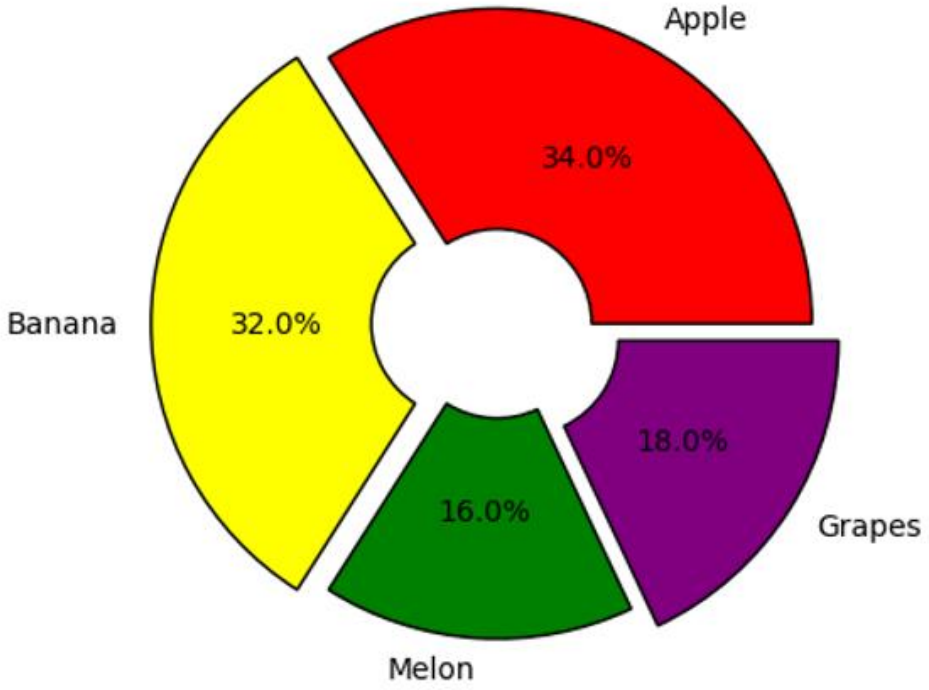


실습2. 그래프 그리기

```
categories = ['Category 1', 'Category 2', 'Category 3', 'Category 4', 'Category 5']
data = [20, 35, 15, 27, 45]
```



실습3. 그래프 그리기



복.습.철.저

수고하셨습니다