

**비주얼 프로그래밍**

**김승태 교수님**

**과제 4**

**20176359 신수현**

## [문제 1]

득표율 그래프 그리기

### 1. 문제의 해결 방안

xlsx 파일을 가져오기 위해 xlrd를 import 한다. 이후 파일을 열고, 열을 labels에 for문을 통해 담는다. 마찬가지로 frequency는 (득표율) 값으로, for문을 통해 list에 담는다. pie chart를 그릴때 숫자가 겹치지 않기 위해서 sort 함수를 이용하여 비율이 큰 값과 작은 값이 섞일 수 있도록 정렬해준다. 나머지는 pie chart를 예쁘게 그리기 위한 소스코드를 검색하여 코딩한 것이다. 매 10초마다 그림이 그려지고 지워질 수 있도록 plt.draw(), plt.pause(10), plt.clf() 을 실행해준다. xlsx파일을 10초안에 업데이트하면 pie chart가 달라지는 것을 볼 수 있다.

### 2. 소스코드

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

isFirst = True

# make a square figure and axes
# pie chart looks best in square figures
# otherwise it looks like ellipses

while True:
    f = xlrd.open_workbook(filename = 'elec.xlsx')

    colors = ['lightblue', 'pink', 'purple', 'burlywood', 'lightcoral']
    ws = f.sheet_by_index(0)

    labels = []
    for i in range(5):
        labels.append(ws.row_values(0)[i])

    frequency = []
    for i in range(5):
```

```
frequency.append(int(ws.row_values(1)[i]))
```

```
#숫자가 겹치지 않게 sort
```

```
frequency.sort()
```

```
tempsmall = frequency[0]
```

```
tempbig = frequency[- 1]
```

```
frequency[0] = tempbig
```

```
frequency[- 1] = tempsmall
```

```
if isFirst:
```

```
    isFirst = False
```

```
    fig = plt.figure(figsize=(8,8)) ## 캔버스 생성
```

```
fig.set_facecolor('white') ## 캔버스 배경색을 하얀색으로 설정
```

```
ax = fig.add_subplot() ## 프레임 생성
```

```
pie = ax.pie(frequency, ## 파이차트 출력
```

```
    startangle=90, ## 시작점을 90도(degree)로 지정
```

```
    counterclock=False, ## 시계방향으로 그려짐
```

```
    colors = colors, ## 색상 지정
```

```
)
```

```
total =0
```

```
for i in range(len(frequency)):
```

```
    total += frequency[i]
```

```
threshold = 5 ## 상한선 비율
```

```
sum_pct = 0 ## 퍼센티지
```

```
bbox_props = dict(boxstyle='square',fc='w',ec='w',alpha=0) ## annotation 박스 스타일
```

```
## annotation 설정
```

```

config = dict(arrowprops=dict(arrowstyle='-',),bbox=bbox_props,va='center')

for i,l in enumerate(labels):
    ang1, ang2 = ax.patches[i].theta1, ax.patches[i].theta2 ## 파이의 시작 각도와 끝 각도
    center, r = ax.patches[i].center, ax.patches[i].r ## 원의 중심 좌표와 반지름길이

    if i < len(labels) - 1:
        sum_pct += float(f'{frequency[i]/total*100:.2f}')
        text = f'{frequency[i]/total*100:.2f}%'
    else: ## 마지막 파이 조각은 퍼센티지의 합이 100이 되도록 비율을 조절
        text = f'{100- sum_pct:.2f}%'

    ## 비율 상한선보다 작은 것들은 Annotation으로 만든다.
    if frequency[i]/total*100 < threshold:
        ang = (ang1+ang2)/2 ## 중심각
        x = np.cos(np.deg2rad(ang)) ## Annotation의 끝점에 해당하는 x좌표
        y = np.sin(np.deg2rad(ang)) ## Annotation의 끝점에 해당하는 y좌표

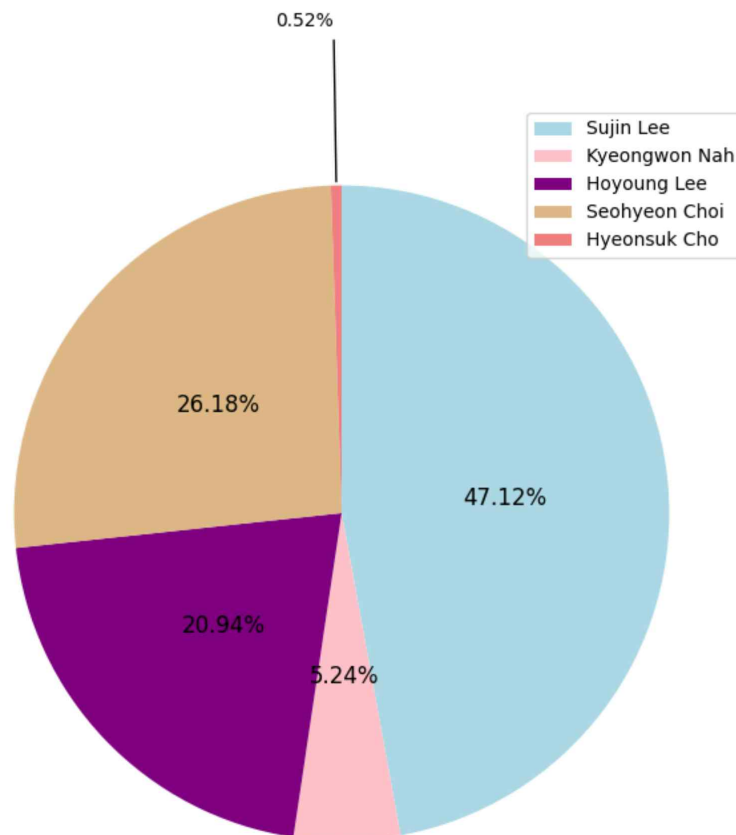
        ## x좌표가 양수이면 즉 y축을 중심으로 오른쪽에 있으면 왼쪽 정렬
        ## x좌표가 음수이면 즉 y축을 중심으로 왼쪽에 있으면 오른쪽 정렬
        horizontalalignment = {- 1: "right", 1: "left"}[int(np.sign(x))]
        connectionstyle = "angle,angleA=0,angleB={}".format(ang) ## 시작점과 끝점 연결 스타일
        config["arrowprops"].update({"connectionstyle": connectionstyle})
        ax.annotate(text, xy=(x, y), xytext=(1.6*x, 1.5*y),
                    horizontalalignment=horizontalalignment, **config)
    else:
        x = (r/2)*np.cos(np.pi/180*((ang1+ang2)/2)) + center[0] ## 텍스트 x좌표
        y = (r/2)*np.sin(np.pi/180*((ang1+ang2)/2)) + center[1] ## 텍스트 y좌표
        ax.text(x,y, text, ha='center', va='center', fontsize=12)

plt.legend(pie[0],labels,loc='upper right') ## 범례
plt.draw()
plt.pause(10)
plt.clf()

```

### 3.결과 화면 캡처

Figure 1



### [문제 2]

정규분포의 랜덤값 생성하기

#### 1.문제 해결 방안

irandom을 이용하여 1000개의 0~100까지의 숫자를 뽑기 위해서 for문을 이용한다. histogram을 이용하여 각 숫자별 빈도수가 나오게끔 그려준다.

numpy를 이용할때는, np.random.normal로 값을 뽑아주고, 0~100 사이 숫자만 나오게끔 bins와 xlim을 이용하여 숫자 범위를 조절해준다.

## 2. 소스 코드

```
import numpy as np
import matplotlib.pyplot as plt
import random
import pandas as pd
```

#random 으로 값 뽑기

```
num_list = []
for i in range(1000):
    num_list.append(random.randint(0, 100))
```

```
plt.hist(num_list, bins=100)
plt.draw()
plt.pause(3)
plt.clf()
```

#np.random.normal로 값 뽑기

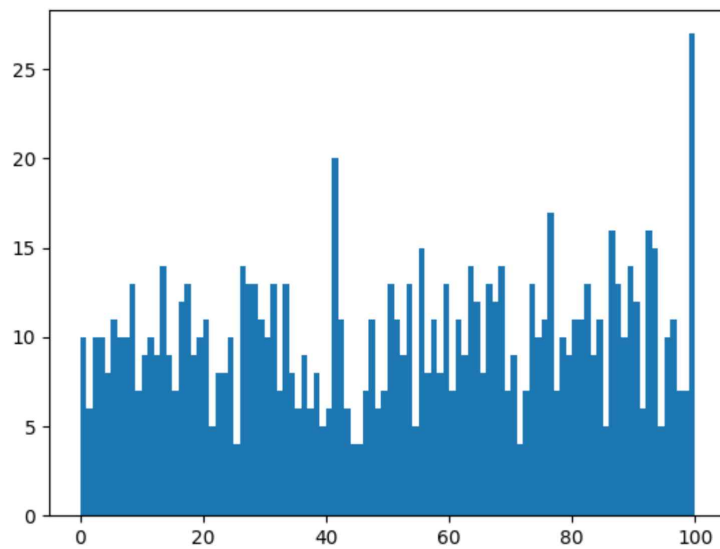
```
a= np.random.normal(50, 20, 1000)
plt.xlim([0, 100])
plt.hist(a, bins=100)
plt.draw()
plt.show()
```

## 3. 정규분포에 의한 랜덤값 생성 방법을 언제 쓰면 좋을지 설명하라.

숫자들을 랜덤하게 뽑을 때, 빈도수가 경향성을 지니도록 하고 싶을 때 이용하면 좋을 것 같다.

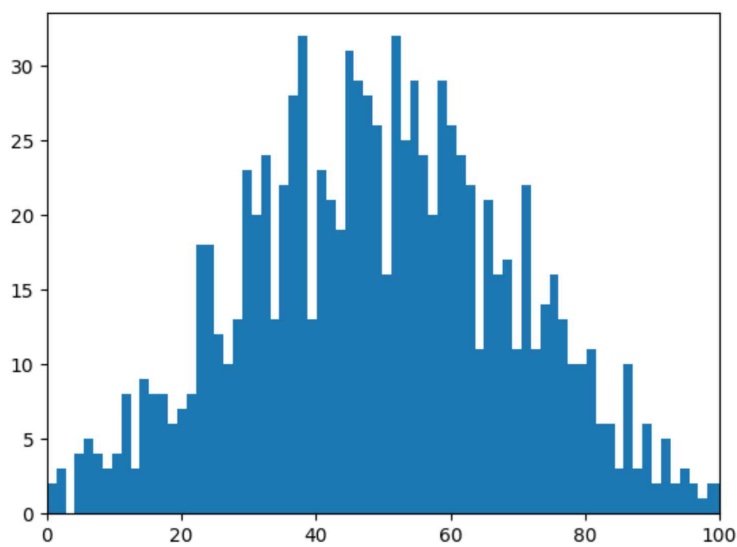
#### 4. 결과 화면 캡처와 특징 설명

Figure 1



0~100까지의 숫자를 1000개 랜덤하게 뽑을 때, 각 숫자마다의 빈도수에 따른 규칙성이 전혀 보이지 않는다.

Figure 1



0~100까지의 숫자를 1000개 뽑았을때, 50을 기준으로 0이나 100으로 다가가면 빈도수가 줄어드는 경향성을 띈다.