제 2장. 모집단과 표본

기술 통계학 (descriptive statistics)
: 자료의 특성을 표, 그림, 통계량 등을 사용하여 쉽게 파악할 수 있도록 정리, 요약하는 방법을 다루는 분야

2.1 패키지의 사용

파이썬 코드 작성 시 기본적인 내장 함수 (built-in function)를 사용 할 수 있지만 내장함수만으로는 다양한 분석이 불가하다. 따라서 파이썬의 다양한 모듈(module)과 패키지(package)를 사용해야 한다.
모듈(module)이란 변수, 함수 혹은 클래스를 모아 놓은 파일이다. 즉, 특정 기능에 대해 만들어놓은 파일을 의미하며 우리는 모듈을 통해 다른 사람이 만든 함수를 사용하거나, 자신이 만든 함수를 공유할 수 있다. 자신이 만든 함수를 저장해 놓을 수도 있다.
패키지(package)란 특정 기능과 관련된 여러 모듈을 묶어놓은 것을 말한다.

Anaconda는 자주 쓰이는 중요한 패키지들을 기본적으로 제공한다. 주요 패키지에 대한 간략한 설명은 다음과 같다.

패키지명	설명
Numpy	높은 수준의 계산을 위한 데이터 패키지이다. 주요 기능으로는 벡터 연산, 다차원배열, 선형대수, 난수 생성 등이 있다
Pandas	구조화된 데이터나 표 형식의 데이터를 빠르고 쉽게 다루게 한다. 특히 데이터프레임 구조를 다룰 때 유용하다.
Scipy	수치해석기능을 제공하는 파이썬 패키지로 여러 가지 서브 패키지를 가지고 있다. 그 중 stats 서브 패키지는 다양한 통계적 기능을 제공한다.
Matplotlib	그래프나 2차원 데이터 시각화를 생성하는 패키지로 가장 많이 사용되는 시각화 패키지이다.

패키지를 사용하기 위해서는 import 명령어를 사용한다.

```
import [패키지] as [예약어]
```

예약어(약자)를 사용하여 패키지를 불러온 후 그 기능을 사용하는 경우에는 약자를 접두어로 붙여 사용한다. 다음의 간단한 예를 확인해보자.

```
import numpy as np

a = [1,2,3,4,5]

np.sum(a)
```

```
15
```

numpy 패키지를 약자 np를 사용하여 import 하였다. 그리고 리스트로 작성된 변수 a에 대하여 numpy에 포함된 기본 통계 함수(메소드)인 sum을 사용하여 변수의 합을 구할 수 있다.

패키지 안에 있는 모듈 속 함수를 사용하는 방법은 다양하다. 예를 들어 데이터를 시각화 하는 패키지인 matplotlib 안에 있는 pyplot 모듈은 히스토그램, 막대그래프, 산점도 등을 그리는 다양한 함수를 갖고 있다, 그 중 히스토그램을 그리는 hist 함수를 사용할 수 있는 방법은 다음과 같이 다양하다.

```
1.import matplotlib → matplotlib.pyplot.hist(score)
2.Import matplotlib as plt → plt.pyplot.hist(score)
3.import matplotlib.pyplot as plt → plt.hist(score)
4.from matplotlib.pyplot import hist
```

앞으로 다양한 패키지의 사용법은 예제를 다루며 하나씩 살펴보기로 하자.

2.2 일변량 자료의 요약 - 그래프를 이용한 요약

다음은 18명의 학생들의 성별과 점수에 대한 자료이다.

성별	M	M	M	M	M	M	M	M	F	F	F	F	F	F	F	F	F	F
점수	98	90	96	54	43	87	88	90	94	92	81	79	85	91	79	88	89	83

다음과 같이 자료를 입력한다.

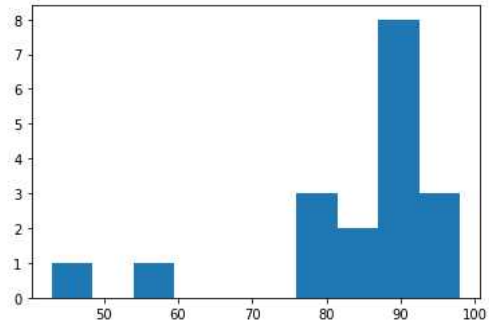
```
gender=["M","M","M","M","M","M","M","M","F","F","F","F","F","F","F","F","F","F"]
score=[98,90,96,54,43,87,88,90,94,92,81,79,85,91,79,88,89,83]
```

2.2.1 히스토그램

일변량 자료의 분포를 알아보는데 유용한 그래프는 히스토그램이다. 데이터 시각화에 사용할 Matplotlib.pyplot 모듈을 약칭 plt 로 import 하여 다음과 같이 그릴 수 있다. plt.show()는 모든 플롯을 시각화 할 때 반드시 들어가야 하는 명령어이다.

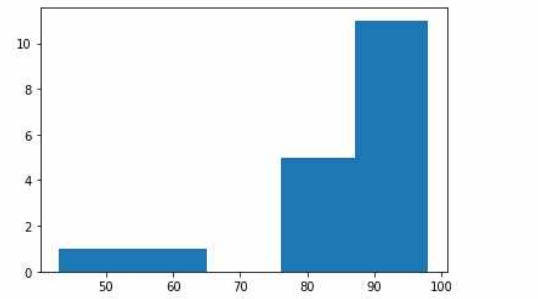
```
import matplotlib.pyplot as plt

plt.hist(score)
plt.show()
```

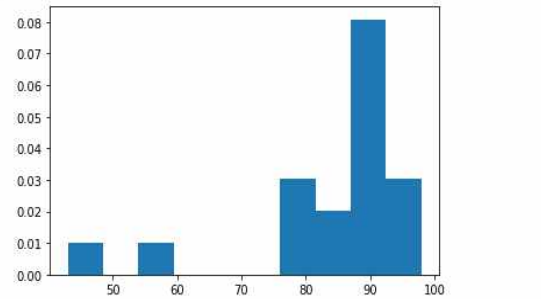


일반적으로 패키지에 구현된 함수는 매우 다양한 옵션을 제공하며 이를 적절히 지정할 수 있다. 예를 들어. plt.hist( ) 함수의 인자 중 bins는 히스토그램의 구간의 개수를 지정하는 옵션으로 원하는 구간의 개수를 숫자로 지정한다. 그리고 density 옵션은 논리값을 사용하여 지정할 수 있는데 'True'로 지정하는 경우 히스토그램의 Y축을 빈도수가 아닌 각 구간의 확률밀도로 나타낼 수 있다. 옵션값을 지정하지 않는 경우에는 각 옵션별 기본값을 사용한다. 다음의 결과를 확인해보자.

```
plt.hist(score, bins=5)
plt.show()
```



```
plt.hist(score, density=True)
plt.show()
```

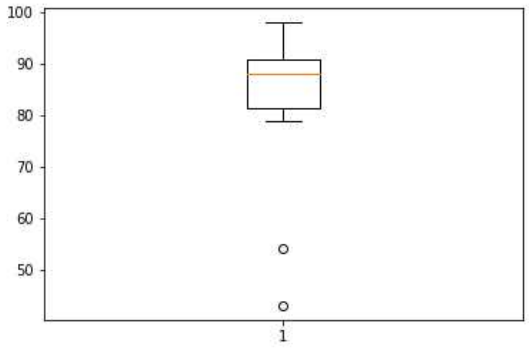


함수와 함수에 속한 각 옵션에 대한 설명은 도움말을 통해 확인할 수 있다. 파이썬에서 도움말을 보기 위해서는 help()를 사용할 수 있다.

2.2.2 상자그림

상자그림은 데이터의 분포를 보여주는 그림으로 가운데 상자는 제 1사분위수, 중앙값, 제 3사분위수를 보여준다. 상자그림은 plt.boxplot( )로 그릴 수 있다.

```
plt.boxplot(score)
plt.show()
```



2.3 일변량 자료의 요약 - 수치를 이용한 요약

통계량 : 표본으로부터 계산되는 표본의 특성값

- 중심위치의 측도 : 평균, 중앙값
- 산포의 측도 : 분산, 표준편차, 사분위수범위

2.3.1 범주형 자료의 요약

범주형 자료의 요약은 분할표를 이용할 수 있다. 분할표의 작성은 pandas 패키지의 crosstab()함수를 사용한다. 단, pandas 패키지는 시리즈(Series)와 데이터 프레임(DataFrame)이라는 구조화 된 데이터 형식을 사용하기 때문에 주어진 자료를 적절한 형태로 먼저 변환해야 한다. 시리즈는 데이터가 순차적으로 나열된 1차원 배열, 데이터 프레임은 행(row)과 열(column)이 있는 2차원 배열이라고 생각하면 된다.

먼저 현재 주어진 성별(gender) 변수를 사용하여 빈도표(frequency table)를 작성해보자.

```
pandas.crosstab(index, colums)
```

```
import pandas as pd

gender_sr = pd.Series(gender)
pd.crosstab(index=gender_sr, columns="count")
```

col_0	count
row_0	
F	10
M	8

처음 우리가 성별(gender) 변수를 리스트의 형태로 저장했으므로 pd.Series( ) 함수를 사용하여 1차원 배열의 형태인 시리즈로 바꿔주도록 한다. pd.crosstab( )의 index는 행, columns는 열의 값을 지정한다.

또는 collections 모듈의 Counter를 사용하면 리스트의 형태에서도 간단히 데이터의 개수를 셀 수 있다.

```
from collections import Counter
Counter(gender)
```

```
Counter({'M': 8, 'F': 10})
```

2.3.2 숫자형 자료의 요약

숫자형 자료의 수치적 요약은 다음과 같은 다양한 통계량을 사용할 수 있다.

```
import numpy as np
print(np.mean(score)) #평균
print(np.std(score)) #표준편차
print(np.var(score)) #분산
print(np.median(score)) #중앙값
print(np.percentile(score,50)) #분위수
print(np.sum(score)) #총 합
print(np.min(score)) # 최소값
print(np.max(score)) # 최대값
```

```
83.7222222222
13.6047604232
185.089506173
88.0
88.0
1507
43
98
```

2.4 이변량 자료의 요약

다음은 어느 고등학교에서 랜덤하게 추출된 10명의 수학, 물리 성적이다.

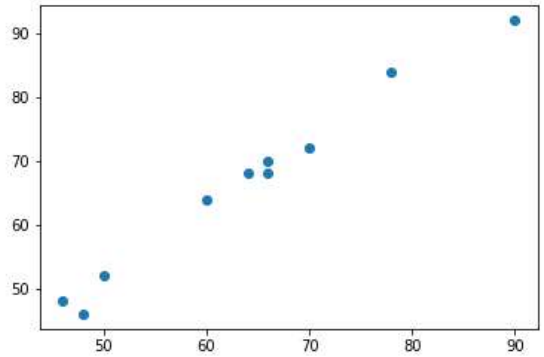
수학	66	64	48	46	78	60	90	50	66	70
물리	70	68	46	48	84	64	92	52	68	72

```
math=[66,64,48,46,78,60,90,50,66,70]
phy=[70,68,46,48,84,64,92,52,68,72]
```

2.4.1 그래프를 이용한 요약

그래프를 이용한 이변량 자료의 요약은 산점도(scatter plot)를 이용할 수 있으며, matplotlib.pyplot 모듈의 scatter()를 사용한다.

```
plt.scatter(math,phy)
plt.show()
```

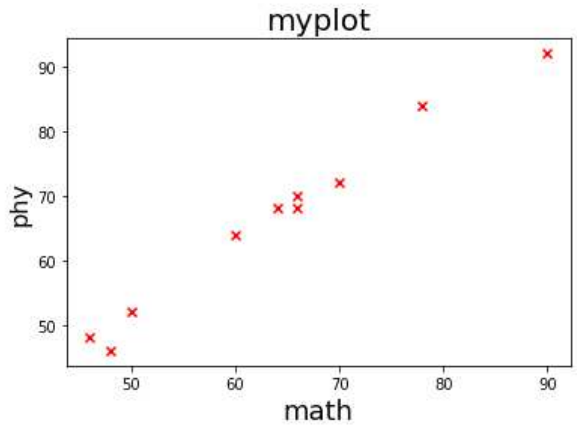


plt.scatter( )함수 역시 다양한 옵션을 지정할 수 있는데, 다음은 몇 가지 주요 선택사항들에 대한 설명이다.

plt.scatter(x , y)	산점도의 x축 변수와 y축 변수를 지정한다
plt.title()	차트 제목 추가
plt.xlabel()	x축 이름
plt.ylabel()	y축 이름
fontsize	글씨크기 (숫자로 지정)
c	그래프 색상
marker	marker style

```
plt.scatter(math,phy, c='red', marker='x')  # 산점도 지정
plt.title('myplot', fontsize=20) # 제목 추가
plt.xlabel('math', fontsize=18) # x축 이름
plt.ylabel('phy', fontsize=16) # y축 이름

plt.show()
```



2.4.2 상관계수를 이용한 요약

두 변수의 상관계수는 numpy의 corrcoef( )를 사용한다.

numpy.corrcoef(변수1, 변수2)

```
np.corrcoef(math,phy)
```

```
array([[ 1.       ,  0.9918056],
       [ 0.9918056,  1.       ]])
```

2.5 자료를 이용한 예제 (cdc.txt)

행동위험요인 감시시스템(The Behavioral Risk Factor Surveillance System)은 매년 미국에서 시행되는 대규모 전화 설문 조사이다. 이 조사에서는 응답자들의 현재 건강 상태 및 그들의 건강과 관련된 생활 습관 등을 조사한다. 이 조사에 관한 자세한 내용은 BRFSS 의 웹사이트에서 확인할 수 있다. (http://www.cdc.gov/brfss)

주어진 자료는 2000년도에 시행된 20,000명의 BRFSS 조사 데이터의 일부이며 전체 200개 이상의 항목 중에서 간추린 9개의 항목을 포함하고 있다. 각 변수에 대한 설명은 다음과 같다.

- genhlth : 범주형 자료, 전반적인 건강상태 (excellent/very good/good/fair/poor)
- exerany : 범주형 자료, 지난달의 운동 여부 (1=yes, 0=no)
- hlthplan : 범주형 자료, 건강보험 가입 여부 (1=yes, 0=no)
- smoke100 : 범주형 자료, 현재까지 최소 100개피 이상의 담배 흡연 여부 (1=yes, 0=no)
- height : 숫자형 자료, 신장 (inch)
- weight : 숫자형 자료, 체중 (pound)
- wtdesire : 숫자형 자료, 응답자가 생각하는 본인의 이상적인 체중 (pound)
- age : 숫자형 자료, 나이 (year)
- gender : 범주형 자료, 성별 (m=남성, f=여성)

주어진 자료는 텍스트 형태의 자료이다. Pandas는 다양한 형태의 외부 파일을 읽어와서 행과 열이 있는 데이터 프레임으로 변환하는 함수를 제공한다. 어떤 파일이든 Pandas 객체인 데이터프레임으로 변환되고 나면 Pandas의 모든 함수와 기능을 자유롭게 사용할 수 있다.

먼저, 주어진 자료를 파이썬으로 읽어보자. 명령어는 다음과 같다.

pandas.read_csv("파일 경로")

```
import pandas as pd
df=pd.read_csv("D:/cdc.txt", sep=" ")
```

주어진 텍스트 데이터를 df 라는 이름의 데이터 프레임으로 저장하였다. 여기서 sep=" " 옵션은 텍스트 데이터가 공백(" ")으로 구분되어 있음을 나타낸다. 파일에 따라서 쉼표(,) 또는 탭(\t)으로 구분되어 있을 수도 있으니 그게 맞는 적절한 옵션을 지정해주어야 한다.

이렇게 데이터를 불러온 후, 분석을 실행하기 전 다음과 같은 명령어를 사용하여 데이터를 살펴볼 수 있다,

dataframe 이름.head(n)	데이터의 처음 n개 행을 보여줌. (default n=5)
dataframe 이름.tail(n)	데이터의 마지막 n개 행을 보여줌. (default n=5)
dataframe 이름.shape( )	데이터 프레임의 (행,열)의 개수
dataframe 이름.info( )	데이터 프레임의 기본 정보 출력
dataframe 이름.describe( )	수치형 데이터에 대한 기술통계량 출력

```
df.head()
```

	genhlth	exerany	hlthplan	smoke100	height	weight	wtdesire	age	gender
1	good	0	1	0	70	175	175	77	m
2	good	0	1	1	64	125	115	33	f
3	good	1	1	1	60	105	105	49	f
4	good	1	1	0	66	132	124	42	f
5	very good	0	1	0	61	150	130	55	f

자료의 각 열(column)은 각각의 변수를 나타낸다. 데이터 프레임의 각 열을 사용하기 위해서는 데이터프레임.변수명을 사용한다. 예를 들어 df 데이터의 첫 번째 열인 genhlth 변수를 사용하기 위해서는 df.genhlth를 입력한다.

예제 1. genhlth 변수에 대해 적절한 방법을 이용하여 요약해보자. 범주형 자료의 경우에는 어떠한 요약 방법을 사용할 수 있는가?
예제 2. weight 변수에 대한 수치적 요약 값을 구해보자. 전체 응답자의 평균 몸무게는 얼마인가?
예제 3. weight 변수와 wtdesire 변수의 산점도를 그려보자. 두 변수 사이에는 어떠한 관계가 존재한다고 보여지는가? 두 변수의 상관계수는 무엇은 나타내고 있는가?
예제 4. wtdesire 변수와 weight 변수의 차를 계산하여 새로운 변수 wdiff 를 만들어보자. wdiff 의 분포는 어떠한가? 수치적 요약과 그래프 요약을 통해 살펴보자. 이것이 의미하는 바는 무엇인가?
예제 5. age 변수를 이용하여 히스토그램을 그려보자. 그리고 구간의 수를 50, 100으로 바꿔가며 동일한 히스토그램을 그린 후 비교해보자.
(참고) 히스토그램은 자료의 형태를 파악하기 위한 쉬운 방법이지만 구간의 수가 달라짐에 따라 그 모양이 조금씩 달라질 수 있다.