파이썬을 활용한 데이터 분석

강사: 차형준

- 웹 슬라이드쇼 페이지입니다.
- 우측 하단의 화살표 혹은 화살표 키로 조작합니다.
- w키로 슬라이드 오버뷰를 볼 수 있습니다.
- ?를 누르면 도움말을 볼 수 있습니다.

0. 들어가기 전에

0.1. 책 속에서

1. 많은 대학원생처럼 나도 방대한 데이터베이스를 구축하고서 컴퓨터 단추 하나만 누르면 일반적인 통계 분석을 수행할 수 있었다. 그러나 그 통계 분석이 어떻게 작동하는지를 깊이 살펴보는 법, 아니 살펴보는 법 자체를 결코 배운 적이 없었다. 그 통계 프로그램은 〈통계적으로 유의미하다〉고 간주하는 숫자를 달랑 하나 내뱉었을 뿐이었다. 불행히도 그 분석 결과는 거짓 긍정임이 거의 확실했다. 그 통계 검정이 내가 적용한 맥락에서 어떤 한계를 지니는지를 스스로 이해하지 못했기 때문이다.

> 2. 워싱턴 대학교에는 〈헛소리 판별하기*Calling Bullshit*〉라는 강좌가 있다*(*정식 강좌명은 *INFO 198/BIOL 106B)*.
>
> *(*데이비드 엡스타인*, 2020,* 『늦깎이 천재들의 비밀』*,* 열린책들*, p47/p261)*

0.2. 생각해보기

- 전세계에서 대략 100만건의 과학기술 논문이 나옴. 과학은 가치 중립적인가? 예) 밀양 송전탑, 사드의 인체 유해성. 코로나 백신은 위험한가? 등등
- 과학 연구자의 윤리(흔히 벌어지는 광경: 실험 데이터 끼워 맞추기, 자의적 결론 해석 등)
- 메커니즘이나 모델이 불분명한 경우가 많고 이때 의미있는 데이터를 추출해 데이터 분석을 함.
- 데이터 분석은 언제나 있어왔으며, 확률론에 기반한 통계가 활용됨.
- 즉, 빅데이터 기술이란 더 큰 데이터풀에서 유익한 데이터를 추출하는 기술.
- 유익한 데이터를 추출할 수 있는 안목이 필요. 자의적이기 때문에 왜곡 가능성도 크다.

1. 파이썬을 활용한 데이터 분석

> 참고자료: 하시모토 히로시/마키노 코오지, *2020*, 『데이터 사이언스 교과서』, 성안당.

1.1. 사용하는 라이브러리

- numpy : 수치, 행렬 계산.
  https://numpy.org/doc/stable/reference/index.html
- scipy : 각종 함수, 변수, 공식 계산.
- matplotlib.pyplot : 그래프 그리기.
  https://matplotlib.org/stable/api/index.html
- pandas : 데이터 분석용. 엑셀과 유사한 기능.

In [27]:

```
#matplotlib에 한글 폰트 적용하기
import matplotlib
# font_list = fm.findSystemFonts(fontpaths=None, fontext ='ttf')
# for i in range(200) : print(i,':', font_list[i])
matplotlib.rcParams['font.family'] = 'KoPubWorldDotum'

#OUTPUT 여러개 띄우기
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

In [28]:

```
#그래프 그리기 예시
import numpy as np          #np라는 이름으로 numpy 라이브러리 불러옴.
import matplotlib.pyplot as plt    #plt라는 이름으로 matplotlib의 pyplot 라이브러리 불러옴.
x = np.linspace(-4,4,40)    #등차수열로 -4에서 4까지 40등분한 배열.(-4, -3.8, -3.6, ... , 0, 3.8, 4)
y1 = x ** 2                 #배열 값마다 x^2 계산 (16, ...,9, ... , 16)
y2 = np.sin(x)              #배열 값마다 sin(x) 계산
y3 = x
y4 = np.cos(x)

#그래프 생성
fig = plt.subplots(figsize=(5,3))  #사이즈를 5인치*3인치로 그래프그리는 공간 생성
plt.plot(x,y1)  #(x, y1) 점별로 그려서 잇기
plt.plot(x,y2)  #(x, y2) 점별로 그려서 잇기

plt.grid()      #그리드 그리기
plt.xlabel('x') #x축을 라벨을 x로 붙여서 그리기
plt.ylabel('sin x') #y축을 라벨을 y로 붙여서 그리기
plt.title('테스트') #제목 붙이기


#그래프 따로 그리기
fig, axs = plt.subplots(nrows=2,ncols=2,figsize=(5,3)) #2행 2열로 그리기
axs[0,0].plot(x,y1)
axs[0,1].plot(x,y2)
axs[1,0].plot(x,y3)
axs[1,1].plot(x,y4)
'''
(0,0) (0,1)
(1,0) (1,1)
'''
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x13419822a90>]
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x134198226a0>]
```

Out[28]:

```
Text(0.5, 0, 'x')
```

Out[28]:

```
Text(0, 0.5, 'sin x')
```

Out[28]:

```
Text(0.5, 1.0, '테스트')
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x13419764160>]
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x13419770b80>]
```

Out[28]:

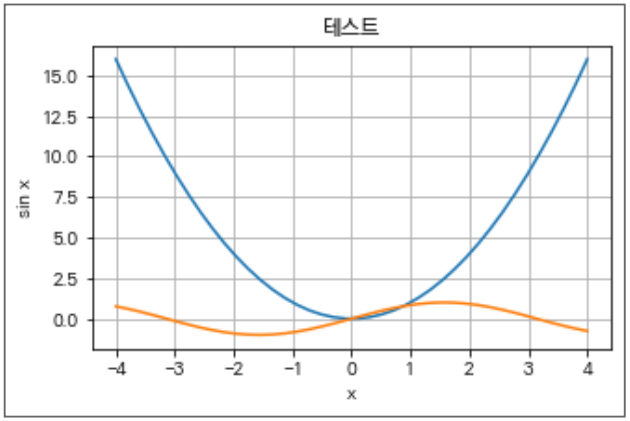
```
[<matplotlib.lines.Line2D at 0x13419770d30>]
```

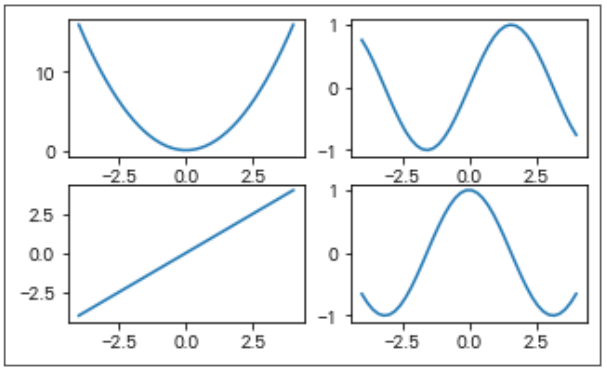
Out[28]:

```
[<matplotlib.lines.Line2D at 0x13419357040>]
```

Out[28]:

```
'₩n(0,0) (0,1)₩n(1,0) (1,1)₩n'
```





In [29]:

```
import pandas as pd #pandas를 pd로 불러오기
df = pd.read_csv('train.csv') #csv파일 불러오기
df.head(6) #표 그리기(앞의 n개 데이터만, 기본값=5개)
```

Out[29]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q

In [30]:

```
# df.tail(3) #뒤에서 3개
# df[4:7] #중간
df1 = pd.DataFrame(df, columns=['Survived','Age']) #특정 열만 선택해서 표 다시 만들기 1
df2 = df.loc[:5, ['Survived', 'Fare','Age']] #특정 열만 선택해서 표 다시 만들기 2
df.iloc[1:3,2:5] # 1~2번째행, 2~4번째열
df2[:2]
df1[0:5]
```

Out[30]:

	Pclass	Name	Sex
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	3	Heikkinen, Miss. Laina	female

Out[30]:

	Survived	Fare	Age
0	0	7.2500	22.0
1	1	71.2833	38.0

Out[30]:

	Survived	Age
0	0	22.0
1	1	38.0
2	1	26.0

	Survived	Age
3	1	35.0
4	0	35.0

In [31]:

```
df['Age'].fillna(df.Age.median(), inplace = True) #비어있는 값을 평균으로 채우기, inplace가 True면 df자체를 바꿈.
df.head(6)
```

Out[31]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

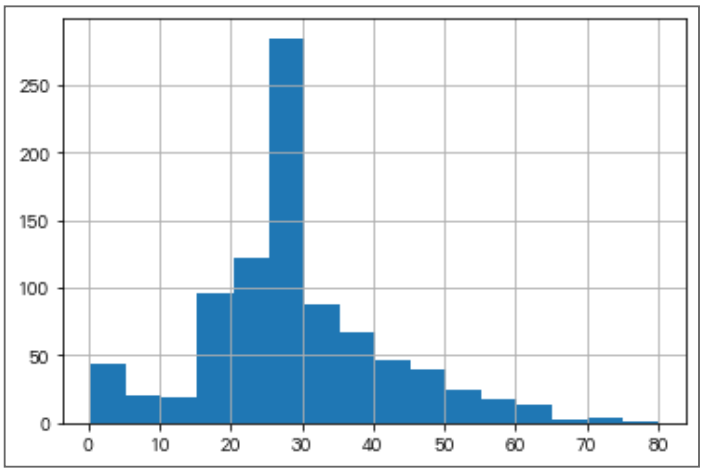
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	28.0	0	0	330877	8.4583	NaN	Q

In [32]:

```
df['Age'].hist(bins=16) #히스토그램, bins는 막대 갯수.
```

Out[32]:

<AxesSubplot:>



그 밖의 다양한 그래프 그리는 방법

- 표 형태 골라서 그리기

표이름.plot(kind='표형태')

- 세로/가로 막대 그래프, 축적 그래프

표이름.plot.bar() 표이름.plot.barh() 표이름.plot.bar(stacked=True) 표이름.plot.barh(stacked=True)

- 산점도

표이름.plot.scatter(x='열1', y='열2', color='색깔이름|RGB', label='라벨이름')

- 그 밖의 그래프 그리는 방법(pandas 메뉴얼 사이트)

https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

- Pandas 라이브러리 튜토리얼(한글)

https://wikidocs.net/70203

- 연습용 데이터 구하기

https://www.kaggle.com/

mask-use-by-county.csv 표로 실습해보기

뉴욕타임즈가 250000명 미국인 대상 군별(주 아래의 행정구역) 응답비율 36%로 "다른사람과 6피트 이내에 있을 것으로 예상되는 경우 공공장소에서 마스크를 얼마나 자주 착용합니까?"에 대한 응답 결과.

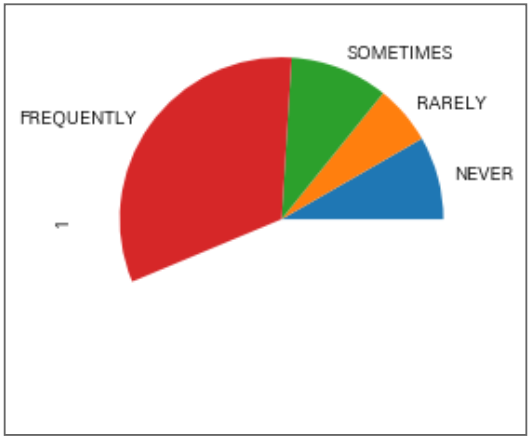
In [33]:

```
# 원형 그래프
table1 = pd.read_csv('mask-use-by-county.csv')
table1.iloc[1,1:5].plot(kind='pie')
```

```
C:₩Python3₩lib₩site-packages₩pandas₩plotting₩_matplotl
ib₩core.py:1583: MatplotlibDeprecationWarning: normali
ze=None does not normalize if the sum is less than 1 b
ut this behavior is deprecated since 3.3 until two min
or releases later. After the deprecation period the de
fault value will be normalize=True. To prevent normali
zation pass normalize=False
  results = ax.pie(y, labels=blabels, **kwds)
```

Out[33]:

```
<AxesSubplot:ylabel='1'>
```


SOMETIMES
RARELY
NEVER
FREQUENTLY
1

2. 확률의 기초

2.0. 용어들

- 확률, 이산확률 변수, 연속확률 변수
- 확률분포: 확률밀도 함수, 확률질량 함수
- 모집단, 표본
- 모수(parameter): 모집단의 특성을 나타내는 값
- 평균, 분산, 표준편차, 공분산 등

2.1. 정규분포

$N(m, \sigma^2)$

$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-m)^2}{2\sigma^2}}$

In [34]:

```
# -*- coding: utf-8 -*-
import scipy.stats
from scipy.stats import norm  # normal distribution, 정규분포

m = 5   # 평균
std = 2 # 표준편차
x = np.arange( -5, 15, 0.01)
y = norm.pdf(x, loc=m, scale=std)
#pdf (probability density function)
#y = (1 / np.sqrt(2 * np.pi * std*std ) ) * np.exp(-(x-m) ** 2 / (2 * std*std) ) 정규분포의 식에 대입

fig = plt.subplots(figsize=(8,3))
plt.plot(x,y)
plt.xlabel('x')
plt.ylabel('probabiity density')
plt.plot(x,y)
```

Out[34]:

```
[<matplotlib.lines.Line2D at 0x134197b2760>]
```

Out[34]:

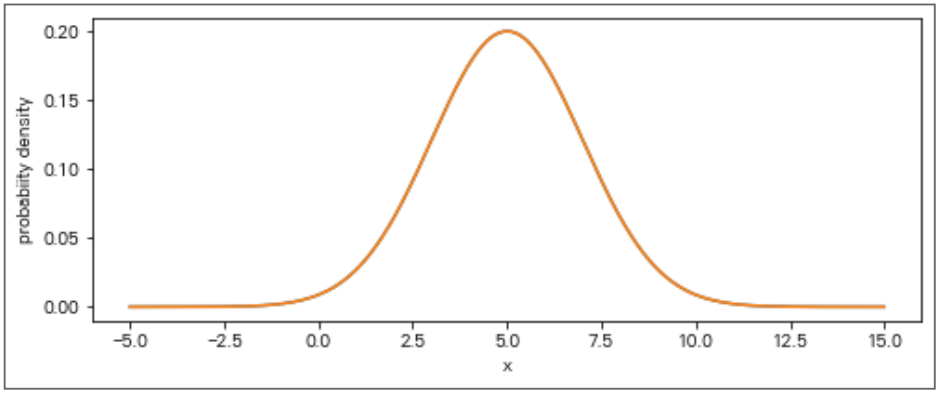
```
Text(0.5, 0, 'x')
```

Out[34]:

```
Text(0, 0.5, 'probabiity density')
```

Out[34]:

```
[<matplotlib.lines.Line2D at 0x134197b2a90>]
```



2.1.1. 표준 정규 분포

$z = \frac{x-m}{\sigma}$

$N(0,1)$

- norm.ppf(α) : $P(z < k) = \alpha$ 인 k를 구해준다.
- norm.isf(α) : $P(k < z) = \alpha$ 인 구간 $(-k, k)$를 구해준다.
- norm.interval(α) : $P(|z| < k) = \alpha$ 인 k를 구해준다.
- norm.cdf(k) : $P(z < k)$ 의 값을 구해준다.

In [35]:

```
m = 0
std = 1
alpha = 0.05
prob = 1 - alpha
z_0 = norm.ppf(prob, loc=m, scale=std)
z_1 = norm.isf(alpha, loc=m, scale=std)
p_0 = norm.cdf(z_0, loc=m, scale=std)
p_1 = norm.interval(0.95, loc=m, scale=std)
print('percent point =', z_0)
print('percent point =', z_1)
print('p=',p_0)
print('p=',p_1)
```

```
percent point = 1.6448536269514722
percent point = 1.6448536269514729
p= 0.95
p= (-1.959963984540054, 1.959963984540054)
```

2.1.2. 중심극한정리

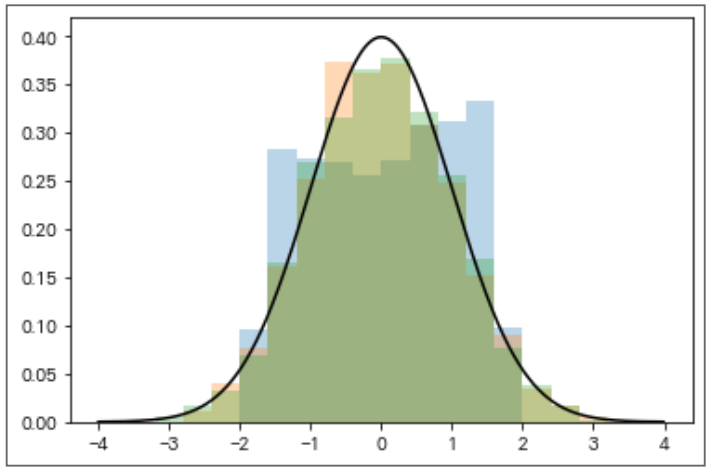
표본 $x_1, x_2, \cdots, x_n$ 평균 m, 표준편차 σ 인 확률분포를 따른다고 할 때
표본 평균을 확률 변수로 하는 $\bar{X}$는 표본의 크기 n이 커짐에 따라 평균 m, 표준편차 $\frac{\sigma}{\sqrt{n}}$ 인 정규분포에 가까워진다.
어떤 확률분포를 가지고 있든지 간에 표본 평균에 대한 확률분포(표본의 분포)가 정규분포에 가까워진다는 것.

In [36]:

```
import matplotlib.animation as animation

N = 2000
y= np.zeros(N)
for n in [1,5,500]:
    for i in range(N): #N번 반복
        x = scipy.stats.uniform.rvs(size=n) #[0,1]구간에서 균일하고 랜덤하게 n개의 변수를 뽑는다.(평균1/2, 표준편차 1/root(12))
        y[i] = (x.mean() - 1/2)/(np.sqrt(1/12)/np.sqrt(n)) #각각의 값 표준화 후 y에 저장
    fig = plt.hist(y, bins=20, range=(-4,4), density=True, alpha=0.3) #히스토그램 그리기
xx = np.arange(-4, 4, 0.01) #정규분포 x축
nrm = scipy.stats.norm.pdf(xx, loc=0.0, scale=1.0 ) #정규분포 y값
plt.plot(xx, nrm, c='k') #정규분포 그리기
plt.show()
```

Out[36]:

```
[<matplotlib.lines.Line2D at 0x13418c841c0>]
```



2.2. 다양한 확률분포(이산확률분포)

2.2.1. 베르누이 분포(0또는 1번 시행)

1. 시행의 결과는 성공 혹은 실패만(성공확률 p, 실패확률 1-p)
2. 각 시행은 독립(서로 영향을 미치지 않음)
3. 성공 확률은 항상 일정

scipy.stats.bernoulli

2.2.2. 이항 분포(n번의 시행횟수)

1. 해당 사건의 발생 여부를 확률로
2. 각 시행은 독립
3. 각 사건이 발생하는 확률은 p로 일정

$E(x) = np, V(x) = npq$

scipy.stats.bionom

예시: 동전 던지기, 주사위에서 특정 수가 나오는 사건.

2.2.3. 포아송 분포

시간 t 내에서 평균 λ 번 발생하는 사건이 k번 일어날 확률이면서 아래의 조건을 따를 때 갖는 분포.

1. 각 시행은 독립
2. 사건이 일어날 확률은 어느 시간대나 동일
3. 매우 짧은 시간동안 사건이 두 번 일어날 확률은 매우 작다.

$P(X=k) = \exp(-\lambda t)\frac{(\lambda t)^k}{k!}$

$E(x) = \lambda, V(x) = \lambda$

예시: 교통사고, 하루에 받는 카톡의 수, FIFA 리그의 득점 등.

In [37]:

```
from scipy.stats import poisson #포아송 분포 가져오기

fig = plt.subplots(figsize=(8,4))
k = np.arange(0,16)

for lamb in range(1,6):
    p = poisson.pmf(k, lamb)
    plt.plot(k, p, label='lambda='+str(lamb))

plt.xlabel('k')
plt.ylabel('Probability mass function')
plt.legend()    #범례 표시하기
plt.show()
```

Out[37]:

```
[<matplotlib.lines.Line2D at 0x134195c7040>]
```

Out[37]:

```
[<matplotlib.lines.Line2D at 0x134195c7190>]
```

Out[37]:

```
[<matplotlib.lines.Line2D at 0x134195c7550>]
```

Out[37]:

```
[<matplotlib.lines.Line2D at 0x134195c7850>]
```

Out[37]:

```
[<matplotlib.lines.Line2D at 0x134195c7e50>]
```

Out[37]:

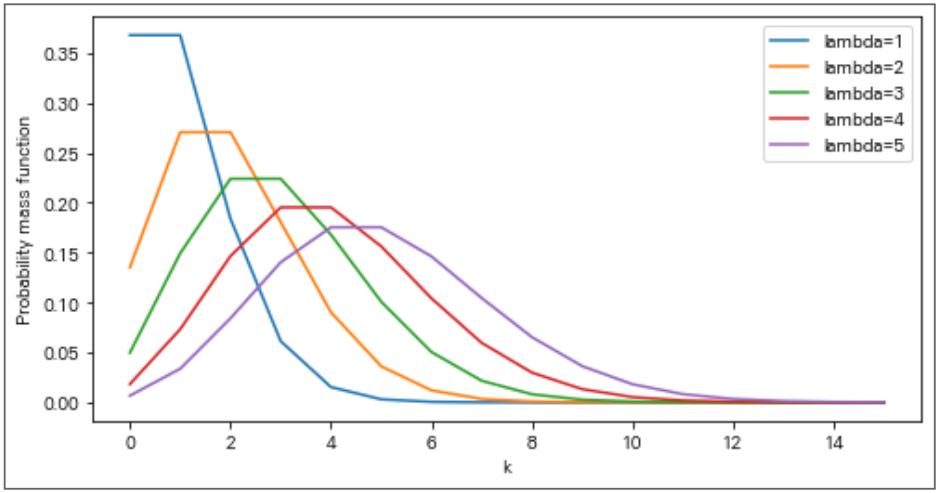
```
Text(0.5, 0, 'k')
```

Out[37]:

```
Text(0, 0.5, 'Probability mass function')
```

Out[37]:

```
<matplotlib.legend.Legend at 0x13419476760>
```


lambda=1
lambda=2
lambda=3
lambda=4
lambda=5
Probability mass function
k
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
12
14

예시1: 교통사고 평균 2.4건 / 일의 경우 , 교통사고가 2건/일 이하일 확률

In [38]:

```
_lambda = 2.4
psum = 0
for k in [0,1,2]:
    p = poisson.pmf(k, mu=_lambda) #mu:평균사건수, k번 일어날 확률
    psum = psum + p
print('sum of p =',psum)
```

```
sum of p = 0.5697087466575105
```

예시2: 한 축구 리그에서 A팀의 시합당 평균 득점 1, B팀의 시합당 평균 득점 2 일 때 A팀이 2:1로 이길 확률
이기는 케이스를 모두 구해서 더하면 이 경우 A팀이 이길 확률이 18.2%, 무승부일 확률 39.4%임을 알 수 있다.
득점력이 2배 차이나도 야구와 같이 득점력의 점수차가 큰 경우(평균3: 평균6) 이길확률과 무승부일 확률이 훨씬 낮아진다.

In [39]:

```
print(poisson.pmf(2, mu=1) * poisson.pmf(1, mu=2))
```

0.04978706836786394

2.2.4. 카이제곱(χ^2) 분포

표준정규분포를 따르는 서로 다른 변수들을 제곱해 더한 값을 새로운 확률 변수로 하는 분포.
가설, 신뢰구간 검정(카이제곱 검정, 프리드만 검정)에 사용.
scipy.stats.chi2

2.2.5. 지수 분포

편의점에 손님이 오는 시간 간격, 쇼크 후 사망할 때까지의 시간 간격 등 어떤 현상이 일어나는 주기의 확률.
포아송 분포와 같이 사용됨.
scipy.stats.expon

2.2.6. F 분포

카이제곱 분포를 따르는 서로 독립인 두 변수의 비가 따르는 분포.
scipy.stats.f

2.2.7. t 분포

검정에서 모분산을 알지 못할 때 사용되는 분포.
좌우 대칭인 분포형태. 자유도 값이 커질수록 정규분포에 가까워진다.
scipy.stats.t

2.2.8. 균일 분포

상수 함수와 같은 분포.
난수 발생, 전철 대기시간.

3. 통계의 기초

- 통계: 표본을 조사하여 모집단의 모수를 추정하고 검정하는 것.
- 모집단의 특징을 나타내는 값(모수): 모평균, 모분산 등
- 표본: 모집단에서 추출한 N 개의 데이터(N을 표본수(샘플수))
- 통계량: 표본으로부터 계산된 수치(표본평균, 표본분산, 표본표준편차)

3.1. 통계적 추정

- 표본으로 얻어진 통계량으로 모수가 존재하는 범위를 구하는 것

3.1.1. 점추정

- 모집단의 모수를 하나의 값으로 추정하는 방법.

3.1.2. 구간추정

- 표본평균이나 표본분산이 모평균이나 모분산이 어느정도 확률로 해당 분포의 구간에 들어가는지 추정하는 것.
- 신뢰구간: 해당 분포의 구간
- 신뢰도: 해당 분포에 들어갈 확률 #### 모분산을 아는 경우: 중심극한정리(정규분포)를 이용
- 중심극한정리: 각 표본들이 독립이고 표본수(표본의 크기)가 클 수록 표본평균 $\hat{m}$은 $N(m, \frac{\sigma^2}{n})$인 정규분포에 가까워짐.(모평균m, 모분산σ^2)

모분산을 모르는 경우: T분포를 이용

- 표본수가 작을 경우. 모분산과 표본분산은 차이가 난다.

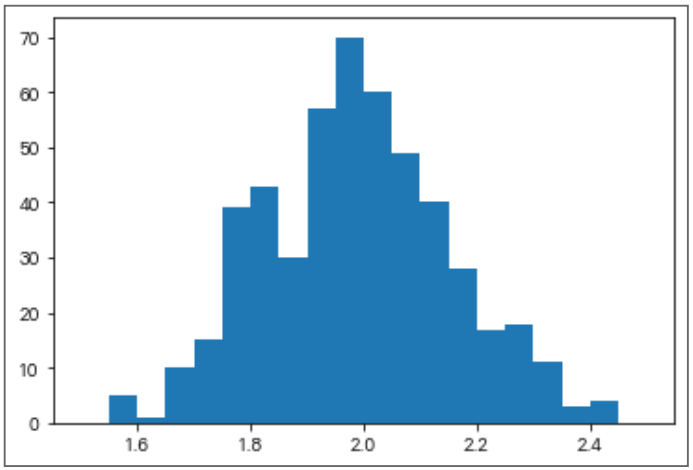
In [40]:

```
num = 500    #뽑은 표본의 개수
N = 10      #표본수
mean, std = 2, 0.5
mu = np.zeros(num)

for i in range(num):
    mu[i] = np.mean( norm.rvs(loc=mean, scale=std, size=N) )  # 평균2, 분산 0.5를 따르는 분포에서 랜덤하게 10개 뽑기

plt.hist(mu, bins=20, range=(1.5, 2.5))
print(mu.mean())    #표본평균들의 평균
print(mu.var())     #표본평균들의 분산
print(0.5**2/N)     #분산^2 / N
# 표본의 개수가 커질수록 정규분포에 가까워지는 것을 알 수 있다. (모든 경우의 수가 되면 아예 같아짐)
```

Out[40]:

```
(array([ 0.,  5.,  1., 10., 15., 39., 43., 30., 57., 70., 60., 49., 40.,
        28., 17., 18., 11.,  3.,  4.,  0.]),
 array([1.5 , 1.55, 1.6 , 1.65, 1.7 , 1.75, 1.8 , 1.85, 1.9 , 1.95, 2.  ,
        2.05, 2.1 , 2.15, 2.2 , 2.25, 2.3 , 2.35, 2.4 , 2.45, 2.5 ]),
 <BarContainer object of 20 artists>)
```

1.9881411660641124
0.027053796364880568
0.025



In [41]:

```
#모분산을 아는 경우
alp = 0.99  #신뢰도 99%
za, zb = norm.interval(alpha=alp, loc=0, scale=1) #신뢰구간 구하기
print('신뢰도 99%: ','za=',za, ' zb=',zb)
alp = 0.95  #신뢰도 95%
za, zb = norm.interval(alpha=alp, loc=0, scale=1)
print('신뢰도 95%: ','za=',za, ' zb=',zb)
```

```
신뢰도 99%:  za= -2.5758293035489004  zb= 2.5758293035489004
신뢰도 95%:  za= -1.959963984540054  zb= 1.959963984540054
```

In [42]:

```
#모분산을 모르는 경우
from scipy.stats import t       #t분포 라이브러리
N = 10
mu_hat = 145.2
std_hat = 23.7
t1 = t.interval( 0.99, df=N-1)
t2 = t.interval( 0.95, df=N-1)
t3 = t.interval( 0.90, df=N-1)
se = std_hat / np.sqrt(N)
print('신뢰도 99% 구간:', mu_hat + t1[0]*se, mu_hat + t1[1]*se)
print('신뢰도 95% 구간:', mu_hat + t2[0]*se, mu_hat + t2[1]*se)
print('신뢰도 90% 구간:', mu_hat + t3[0]*se, mu_hat + t3[1]*se)
```

```
신뢰도 99% 구간: 120.84378885405553 169.55621114594445
신뢰도 95% 구간: 128.24604132892404 162.15395867107594
신뢰도 90% 구간: 131.46155538107368 158.93844461892627
```

3.2. 가설검정 ★★★

3.2.1. 가설검정이란

- 어느 가설에 대해 그것이 옳은가의 여부를 통계학적으로 검증하는 수단.
- 귀무가설(영가설, null hypothesis) H_0: CO2농도는 지구 평균 온도에 영향을 주지 않는다.
- 대립가설(alternative hypothesis) H_1: CO2농도는 지구 평균 온도에 영향을 준다.
- 어떤 검정(검증)을 통해 H_0가 틀렸다는 것을 알았을 때, H_0를 기각한다고 한다. 동시에 H_1은 채택된다고 한다.(귀류법과 비슷)
- H_0를 기각하지 않았다고 해서 H_0가 옳다는 것을 보증해주지는 못한다. (모순이 없다고 해서 맞다고는 할 수 없다.)

3.2.2. 가설검정의 2가지 오류

1. 1종 오류: H_0가 옳은데 H_0을 기각. 필요한 데이터를 얻지 못함.(관측 실패 등)
   예: CO2 농도가 지구 평균 온도에 영향을 주지 않는데 가설을 기각한 경우.
2. 2종 오류: H_1가 옳은데 H_0을 기각하지 않음. 필요없는 데이터를 얻음.(노이즈 값, 관련 없는 자의적 해석 반영 등)
   예: CO2 농도가 지구 평균 온도에 영향을 주지 않는데 준다는 가설을 기각하지 않은 경우.

3.2.3. 유의수준

- 가설검정시 귀무가설 기각 여부를 판정하는 기준.
- 실험에서 오류가 날 확률, 우연히 일어날 확률 등을 고려해 유의수준으로 놓는다.
- 유의수준(=위험률=α)값으로 $0.05, 0.01, 0.001$을 주로 씀.
- $p < \alpha$ 일때 귀무가설을 기각하게 된다.

3.2.4. 단측검정, 양측검정

- 귀무가설이 등호로 설정된 경우 대립가설을 세우는 방식.
- 단측검정: 한 쪽 방향으로 검정.
- 양측검정: 큰 지 작은지는 고려하지 않고 검정.

예: 귀무가설 $\mu = 2.0$ 일 때

- 단측검정 대립가설 $\mu > 2.0$ 또는 $\mu < 2.0$ 둘 중 하나로 정함.
- 양측검정 대립가설 $\mu \neq 2.0$

3.2.5. 모평균의 검정(검정통계량 설정)

- 앞에서 했던 신뢰도, 신뢰구간과 같다.
- 모분산을 아는 경우: 정규분포를 이용한 z검정.
- 모분산을 모르는 경우: t분포를 이용한 t검정.

3.2.6. 가설 검정의 절차

1. 명제를 세운다.
2. 명제에 적당한 검정통계량 선택한다.
3. 귀무가설H_0 대립가설H_1을 세운다. 기각하려는 가설을 귀무가설로.
4. 유의수준 α를 정한다.
5. 이용된 검정통계량이 나타내는 확률분포로부터 확률 p를 구한다.
6. $p < \alpha$ 라면 H_0가 일어날 확률이 충분히 작다고 판단해 H_0를 기각하고 H_1을 채택한다.
7. $p > \alpha$ 라면 H_0를 기각하지 않는다. 표본수, 분석 방법 등을 다시 살펴보고 재검정을 실시할지 고려한다.

그 밖의 모분산의 검정(카이제곱 검정), 두 표본차의 검정(예: 두 체온계의 성능 검정) 등이 있다.

In [43]:

```
# 예시 : 어느 학습, 시험 평균점수와 보강의 효용성(단측검정)
# 표본이 작아서 모분산을 모름: t분포 활용
# 유의수준 5%
# 귀무가설: 보강의 효과는 없었다. m=0
# 대립가설: 보강 후 평균점수가 올랐다. m>0
from scipy.stats import t

data = np.array([1, -1, -2, 3, -1, 5, 4, 0, 7, -1])
m = np.average(data) # 표본평균
s = np.std(data, ddof=1) # 표본표준편차
N = len(data)   #표본수
m0 = 0  #귀무가설 값
t1 = (m-m0)/(s/np.sqrt(N)) #표준화
prob = t.cdf(t1,N-1) #t분포로 t검정
print('p value=',1-prob)
```

```
p value= 0.07800883831234118
```

3.2.7. 상관, 무상관의 검정

- 상관: 2가지 데이터를 대상으로 데이터 사이에 관계를 알아보는 방법.
- 상관계수: 두 데이터의 상관의 강약을 수치화 한 것. (양의 상관관계, 음의 상관관계, 무상관)
- 관련의 정도를 설정할 때 얼마일 때 강한지 약한지는 주관적이어서 상관 여부를 판정하는 데 어려움이 있다.
- 상관계수의 문제점을 보완하기 위해 무상관 검정을 사용한다.

예시: 아버지와 아들의 키에 대한 상관 검정. (회귀 분석 용어의 시초가 된 논문)

1. 귀무가설 H_0 : 상관계수 = 0
2. 대립가설 H_1 : 상관계수 $\neq$ 0

In [44]:

```
from scipy import stats

x = np.array([168, 172, 181, 179, 166, 185, 177, 176, 169, 161])
y = np.array([111, 125, 129, 120, 126, 133, 130, 116, 118, 115])
corr, pvalue = stats.pearsonr(x,y) #무상관 검정
print('corr. coef.=',corr, '  p value=',pvalue)
```

```
corr. coef.= 0.6342703173343619   p value= 0.048882990
19331422
```

In [45]:

```
plt.scatter(x,y) #산점도
plt.xlabel('Height of Faters')
plt.ylabel('Height of Sons')
```

Out[45]:

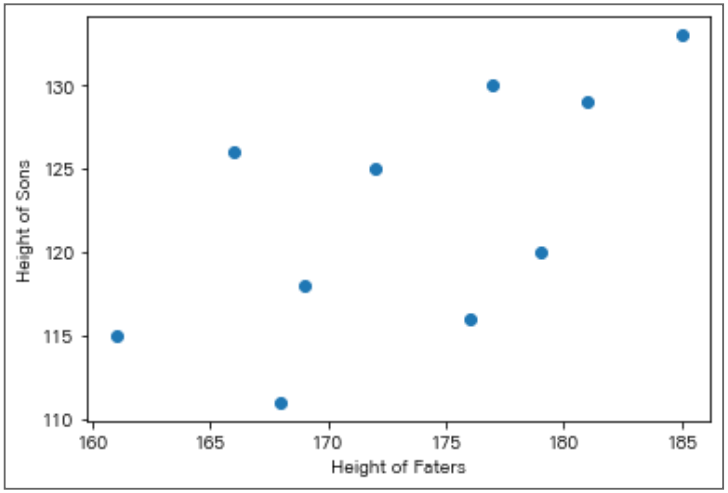
```
<matplotlib.collections.PathCollection at 0x1347e933f70>
```

Out[45]:

```
Text(0.5, 0, 'Height of Faters')
```

Out[45]:

```
Text(0, 0.5, 'Height of Sons')
```


Height of Sons
130
125
120
115
110
160
165
170
175
180
185
Height of Faters

4. 회귀분석?

- 통계를 이용해 미래를 예측하기.
- 두 데이터의 관계를 함수의 형태(다항식 등)로 만들어 내는 것.

직접 데이터를 찾아서 가공해보자.

가설 설정 -> 데이터 수집 -> 데이터 가공 -> 가설 검정 -> 시각화
https://docs.google.com/presentation/d/14mB7al6E4JD142NavkPi1cGmi9J2z usp=sharing
데이터 노다지

- 국가통계포털 **https://kosis.kr/index/index.do**
- 공공데이터포털 **https://www.data.go.kr/**
- 통합데이터지도 **https://www.bigdata-map.kr/**
- 마이크로데이터 **https://mdis.kostat.go.kr/index.do**
- 지역데이터 **https://www.localdata.go.kr/**
- AI허브(인공지능 학습용 데이터) **https://aihub.or.kr/**

예쁘게 꾸미기

- 인공지능 툰 서비스 **https://tooning.io/template-list/home**
- 미리캔버스 **https://www.miricanvas.com/**