평균과 표준편차

- 출처 (https://direction-f.tistory.com/2)
- 자료를 다룸에 있어, 몇몇 대표 지표를 통해 자료를 해석하고 설명하는 것이 도움이 될 때가 많습니다. 특히 이러한 대표 지표 중에서도 가장 대표적으로 활용 되는 것들이 바로 평균과 표준편차 이며, 평균과 표준편차는 단순 실무에서도 많이 적용되고 있는 지표입니다. 해당 글에서는 표본의 평균, 표준편차에 대해서 설명하도록 하겠습니다.

평균

- 평균은 자료의 중심위치를 나타내는 지표중에서도 가장 많이 활용되고 있는 지표입니다. 평균도 산술평균, 기하평균, 조화평균 등 평균도 다양하게 나뉘어질 수 있지만, 우리가 흔히 알고 가장 많이 활용하고 있는 평균은 산술평균입니다.
- 어떤 표본들의 분포가 정규분포라고 가정해보면, 평균과 가까운 표본이 나올 가능성이 평균과 먼 표본이 나올 가능성보다 높다는 것을 유추해볼 수 있습니다.(확률보다는 가능성이 더 적절한 용어라고 판단했습니다.) 따라서 대칭적인 분포에서 평균은 충분히 우리의 자료를 대표할 수 있는 지표로 볼 수 있을 것입니다.
- 우리가 자주 접한 것처럼 표본들이 주어졌을 때, 평균은 아래와 같이 구할 수 있습니다.

$$\overline{x} = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n}$$

- 만약 표본이 {2,3,4}가 주어졌다면 해당 표본 평균은 3이 될 것입니다.
- 평균은 구하기 쉽고 대체적으로 자료를 나타내는 대표 지표지만, 특이값에 의해 민감하게 반응하기도 합니다. 따라서 이러한 특이값이 많은 표본의 경우에는 중앙값이나 최빈값을 평균 대체 지표롤 활용하기도 합니다.
- 중앙값과 최빈값은 가볍게 말씀을 드리면, 중앙값은 전체 표본들을 크기 순으로 배열했을 때, 가운데 위치한 값입니다. 최빈값은 표본들 중에 가장 자주 나오는 값을 뜻하며, 이 최빈값은 연속적이지 않은 이산형 변수일때 적용하며, 표본이 연속형 자료일 때는 원자료에 적용하는 것은 적절하지 않을 것입니다.

표준편차

- 평균과 더불어서 자료를 설명하기 위해 가장 많이 활용하는 표본편차를 살펴보도록 하겠습니다. 먼저 편차에 대해 이야기를 하고 가는 것이 좋을 것 같습니다.
- 편차란 산술평균값을 우리 자료의 중심값을 나타내는 지표로써 사용할때, 각 표본값과 평균값의 차이로 정의하게 됩니다. 다만 저희가 편차의 합을 표본의 "퍼진 정도"를 나타내는 지표로 활용하지 않은 이유는 값의 왜곡이 발생하기 때문입니다.
- 예를 들어 앞서와 같이 평균은 3, 표본은 {2,3,4} 라고 해보겠습니다. 그렇다면 편차의 합은 아래와 같이 계산됩니다.

$$(2 - 3) + (3 - 3) + (4 - 3) = -1 + 0 + 1 = 0$$

- 표본들은 실제로 값이 퍼져있음에도 불구하고 편차의 합은 0이 되기 때문에, 편차는 "퍼진 정도"를 나타내는 값으로 부적절합니다.
- 따라서 분산과 표준편차의 개념이 나오게 됩니다.
- 먼저 분산은 편차의 제곱합을 (표본의 수(n)-1)로 나누게 됩니다.

$$s^2 = \frac{\Sigma(x_i - \overline{x})^2}{n - 1}$$

- 여기서 표본의 수(n)을 활용하지 않고 n-1을 활용하는 이유는 자유도(degree of freedom)라는 개념때문입니다. 우리가 표본의 표준편차를 구하기 위해서는 표본의 평균을 활용해야 합니다. 그런데 표본의 평균은 이미 표본을 나타내는 값이기 때문에(모집단에서 뽑는 표본에 의해 달라지는 값) 1을 빼주게 된 것입니다.(해당 내용에 대해서 좀 더 학습하고 싶으신 분은 MLE(Maximum Likelihood Estimator)관점에서 Unbaised Estimator를 증명한 자료 (https://dawenl.github.io/files/mle_biased.pdf)를 참조하시면 좋을 것 같습니다.)
- 표준편차는 위의 분산의 값에 제곱근을 한 값입니다.

$$s = \sqrt{(s^2)}$$

- 다시 예를 들어보도록 하겠습니다. 표본이 {2,3,4}일때 분산은 아래와 같습니다.

$$s^2 = \frac{(2-3)^2 + (3-3)^2 + (4-3)^2}{3-1} = 1$$

- 표준편차의 정의에 따라 표준편차의 값도 1이 될 것입니다.
- 평균과 표준편차에 대해 간단히 Python을 활용하여 구하는 방법입니다.

In [1]:

```
import numpy as np

sample_ = np.array([2,3,4]) # Sample 생성
np.mean(sample_) # 평균= 2
np.var(sample_, ddof=1) #분산=1 , ddof=1 -> 불편추정량으로 계산(n-1)
np.std(sample_, ddof=1) #표준편차=1 , ddof=1 -> 불편추정량으로 계산(n-1)
```

Out[1]:

```
1.0
```

산점도, 공분산, 상관계수

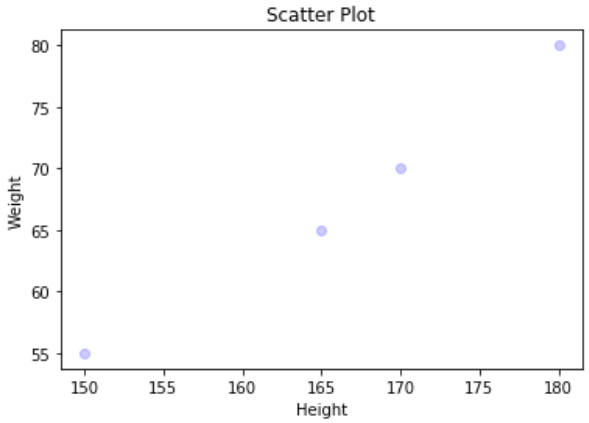
- 출처 (https://direction-f.tistory.com/3)
- 평균과 표준편차는 하나의 표본을 설명하는 대표적인 지표로써 활용되고 있습니다. 그렇다면 두 변수와의 관계를 나타낼 수 있는 방법은 무엇이 있을까요? 바로 대표적으로 산점도, 공분산, 상관계수가 있습니다.

산점도

- 만약 변수 x와 y에 대해 (x,y)가 짝을 이루고 있다고 가정해보겠습니다. 각 변수 x, y에만 관심이 있다면 x의 평균/표준편차, y의 평균/표준편차를 이용하여 x와 y의 특징을 나타낼 수 있을 것입니다. 하지만 저희는 x와 y의 관계를 알고 싶기 때문에 x와 y를 동시에 고려해야 합니다. 이 때 가장 쉽게 적용할 수 있는 방안이 산점도를 활용하는 것입니다.
- 산점도는 변수 x를 수평축에 놓고 변수 y를 수직축에 놓고 각 관측값의 짝을 표시하는 것입니다.
- 아래와 같은 자료가 주어졌다고 해보겠습니다.

	키	체중
A	170	70
B	180	80
C	165	65
D	150	55

- 위의 자료를 산점도로 나타내면 아래와 같이 나타낼 수 있습니다.



- 산점도로 표현하니 키와 체중이 강한 선형관계가 있음을 판단할 수 있습니다.

공분산

- 공분산은 두 개의 변수의 관계를 보여주는 값입니다. 다시 말하면, 변수 x와 y에 대해 x가 변할 때 y가 어떻게 변동하는지를 나타냅니다. 공분산은 각 변수 x, y의 편차의 기대값(평균)으로 나타낼 수 있습니다.

\newline Cov(X,Y)= E((X-\mu_x)(Y-\mu_y)) \newline \newline =\sum_{x}\sum_{y}(X-\mu_x)(Y-\mu_y)f(x,y) \newline \new

- 위 식에서 f(x,y)는 결합확률분포입니다. 변수 x와 y가 쌍으로 관찰될 확률정도로 이해해주시면 될 것 같습니다.
- 여기서 중요한 것은 위의 식에서 보는 것과 같이 x가 평균보다 커질 때 y도 평균보다 커진다면 0보다 큰 값을 가지게 될 것이고 x가 평균보다 작은 값을 가질 때 y는 평균보다 큰 값을 가지게 된다면 0보다 작은 값을 가지게 될 것입니다.
- 이와 같이 공분산이 음수인지, 양수인지에 따라 x와 y가 함께 움직이는지 다르게 움직이는지 판단할 수 있을 것입니다.

$$Cov(X, Y) > 0$$

- X가 증가 할 때 Y도 증가

$$Cov(X, Y) < 0$$

- X가 증가 할 때 Y는 감소

$$Cov(X, Y) = 0$$

- 두 변수간에는 아무런 선형관계가 없음 (두 변수가 독립적이라면 공분산은 0이 되지만, 공분산이 0이라고 해서 항상 독립적이라고 할 수 없음) 출처 (https://destrudo.tistory.com/15)

상관계수

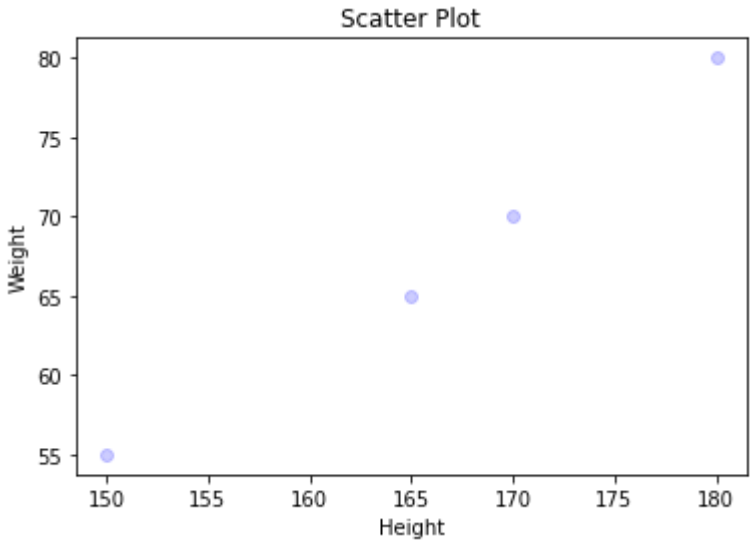
- 공분산을 활용해서 변수 x와 y가 함께 변동하는지를 판단하는 척도로 활용할 수 있지만, "얼마나" 함께 움직이는지를 판단하기는 어렵습니다. 그 이유는 공분산은 변수 x, y의 단위에 따라 값의 변동이 크기 때문입니다.
- 예를 들어 x = (1, 2, 4), y = (5, 10, 15)라고 해보겠습니다. 이 때 공분산은 7.5가 됩니다. 만약 x = (10, 20, 40), y = (50, 100, 150)이라면 공분산은 어떻게 될까요? 750이 됩니다. 이와 같이 공분산은 x,y의 단위에 따라 값의 변동이 커져 "얼마나" 함께 움직이는지를 판단하기 어렵습니다.
- 따라서 상관계수를 활용해서 우리는 경향성의 정도를 파악하게 됩니다.

$$\newline\rho = \frac{Cov(X,Y)}{\sqrt(Var(X)Var(Y))},\;-1\leq \rho\leq1$$

- 위의 수식에서 확인할 수 있듯이 x와 y 분산의 곱의 제곱근을 나눠줌으로써 단위의 영향을 줄여주게 됩니다. 이 때 상관계수는 1에 가까울 수록 양의 관계를, -1에 가까울 수록 음의 관계를 나타냅니다.
- 다시 위의 예로 돌아가 상관계수를 도출해보도록 하겠습니다. x: (1,2,4) y:(5,10,15) 일 때 x와 y의 상관계수는 0.98이 됩니다. x:(10,20,40), y:(50,100,150)일때는 어떻게 될까요? 이 때도 마찬가지로 상관계수가 0.98이 됩니다.
- 이처럼 상관계수를 활용하여 효과적으로 상관성이 얼마나 큰지 판단할 수 있게 됩니다.
- 아래는 이번 글에서 수행한 실습코드를 첨부하였습니다.

In [2]:

```
## Data 입력
import pandas as pd

dict_table = {"Height":pd.Series([170, 180, 165, 150], index=["A","B","C","D"]),
       "Weight":pd.Series([70,80,65,55], index=["A","B","C","D"])}

table_ = pd.DataFrame(dict_table)
table_

## 산점도 그리기

import matplotlib.pyplot as plt

plt.plot("Height","Weight", data= table_, linestyle="none",marker="o", color ="b
lue", alpha =0.2)
plt.title("Scatter Plot")
plt.xlabel("Height", fontsize = 10)
plt.ylabel("Weight", fontsize = 10)
plt.show()

## 공분산, 상관계수 구하기

import numpy as np

list1 = ([1,2,4])
list2 = ([5,10,15])
list3 = ([10,20,40])
list4 = ([50,100,150])

print("공분산-list1 vs list2",np.cov(list1, list2)[0][1])
print("공분산-list3 vs list4",np.cov(list3, list4)[0][1])
print("상관계수- list1 vs list2",np.corrcoef(list1,list2)[0][1])
print("상관계수- list3 vs list4",np.corrcoef(list3,list4)[0][1])
```




```
공분산-list1 vs list2 7.5
공분산-list3 vs list4 750.0
상관계수- list1 vs list2 0.9819805060619657
상관계수- list3 vs list4 0.9819805060619657
```

확률의 이해

- 출처 (https://direction-f.tistory.com/4)
- 일반적으로 우리는 어떤 일이 일어날 가능성를 나타내는데 확률의 개념을 활용합니다. 확률의 개념은 저희가 무언가를 판단할 때 중요한 기준으로 작용하고 있습니다. 쉬운 예로 투자를 하는데, 돈을 잃을 확률이 높다고 여겨지면 투자를 하지 않을 것입니다. 그렇다면 통계학에서 확률은 어떻게 이야기 되고 있을까요?
- 통계적으로 확률을 정의하는데 앞서, 먼저 실험, 표본 공간(Sample space), 사건(event)을 먼저 정리하도록 하겠습니다.

실험, 표본공간, 사건

- 여기에서의 실험은 우리가 그 실험의 결과를 사전에 정확하게 예측할 수 없는 확률실험(Random experiment)를 뜻합니다. 실험을 다시 정의해보면, 어떤 결과 나올지 모르는 상황에서 어떤 결과를 유발하는 과정을 말합니다. 표본공간은 실험에서 유발된 실현가능한 모든 기본 결과들을 모아놓은 집합입니다. 즉, 실험에서 실현가능한 모든 결과들을 모아 놓은 것입니다. 사건은 위의 표본공간의 부분집합을 나타냅니다. 예를 통해 이해하는 것이 가장 좋을 것 같습니다.
- [예시]
  - 실험 : 주사위를 던진다
  - 표본공간 : 주사위에서 나올 수 있는 모든 결과({1,2,3,4,5,6})
  - 사건: "짝수값" 관찰 사건 ({2,4,6}), "홀수값" 관찰 사건 ({1,3,5}), "숫자 1" 관찰 사건({1}) 등등 표본공간으로 이루어진 부분집합
- 위의 예처럼 표본공간이 {1,2,3,4,5,6}으로 이루어 진다면, 이 각각의 숫자를 근원사건이라고 하게 됩니다. 이때 각 근원사건들은 상호배타적(mutually exlusive)으로 구성되어 있으며(주사위를 한 번 던졌을 때 1와 2가 동시에 나올수가 없습니다.), 이 상호배타적인 특성으로 인해 각 특정 근원사건을 얻을 확률은 1/(전체 근원사건들의 수) 입니다.

확률의 개념

- 실험, 표본공간, 사건에 대해서 살펴보았습니다. 그렇다면 위에서 언급된 개념들을 이용하여 확률의 개념을 정리하고자 합니다. 표본공간 S를 아래와 같이 구성 되어 있고 e는 표본공간을 이루는 사건이라고 생각해 보겠습니다.

$$S = \{e_1, e_2, e_3, \cdots, e_n\}$$

- 이 때 아래와 같은 3가지 조건을 만족해야 확률로 정의 할 수 있습니다.

\newline1)\,\,\,\,\,\,\,\,\,\, 0\leq P(e) \leq 1 \newline \newline2)\,\,\,\,\,\,\,\,\,\, P(S)=1 \newline \newline3)\,\,\,\,\,\,\,\,\,\, \sum_

- 위 처럼 확률은 여러 가지 사건이 나타날 수 있을 때 특정한 사건이 일어날 가능성을 수치로 나타낸 것이고 0부터 1사이의 비율로 나타나집니다.

In [4]:

```
import numpy as np

S = np.array([1,2,3,4,5,6]) ## 표본공간

# 짝수가 나올 확률
even = S[S%2==0]
even_prob = len(even)/len(S)
print("짝수가 나올 확률:", even_prob) ## 0.5

#2보다 큰 수가 나올 확률

above_2 = S[S>2]
above_2_prob = len(above_2)/len(S)
print("2보다 큰수가 나올 확률:", round(above_2_prob,2)) ## 0.67

# 1과 3이 나올 확률 = 1이 나올 확률 + 3이 나올 확률

onethree= S[(S==1) | (S==3)]
onethree_prob = len(onethree)/len(S)
one = S[S==1]
three =S[S==3]
one_prob = len(one)/len(S)
three_prob = len(three)/len(S)
print(onethree_prob==(one_prob+three_prob)) ## True
```

```
짝수가 나올 확률: 0.5
2보다 큰수가 나올 확률: 0.67
True
```

확률의 기본 연산

- 출처 (https://direction-f.tistory.com/5)
- 우리가 실제 어떤 사건의 확률을 계산할 때는 여러 관계 있는 사건들을 활용하는 것이 효율적인 경우가 많습니다.
- 예를 들어 주사위를 한 번 던졌을 때, 짝수면서 3보다 이하인 숫자가 나올 확률을 구해보는 문제가 있다고 해보겠습니다.
- 위의 문제는 "짝수인 사건" 과 "3보다 이하인 숫자가 나온 사건"을 활용하여 쉽게 확률을 도출해 볼 수 있습니다.
- 이러한 효율적인 계산을 위해서 사건들의 기본 연산인 여사건, 합사건, 곱사건에 대해서 살펴보겠습니다.

여사건, 합사건, 곱사건

- 여사건은 특정 사건 A가 있을 때 A에 포함되지 않은 근원사건들의 모임으로 나타냅니다. 따라서 특정 사건 A와 특정 사건 A의 여집합의 확률의 합은 1이 됩니다. 따라서 여사건의 확률법칙은 아래와 같습니다.
- 곱사건은 사건 A와 사건 B가 있을 때 사건 A와 사건 B에 모두 포함되는 근원사건들을 나타내게 되며 곱사건의 확률은 아래와 같이 표현됩니다.
- 합사건은 사건 A와 사건 B가 하나의 표본공간에 있다고 가정한다면, 사건 A 혹은 B에 모두 포함되는 근원사건들의 모임입니다. 이 때 단순히 사건 A에 속한 근원사건들과 사건 B에 속한 근원사건들을 더해주면 중복되는 부분이 중복되어 합해지기 때문에 중복되는 근원사건들은 제외해줘야 합니다. 따라서 합사건의 확률법칙은 아래와 같습니다.
- 이 때, 사건 A와 사건 B가 공유하는 근원사건이 없다면 배반사건(Disjoint event)라고 합니다.
- 이제 예시를 통해 좀 더 직관적으로 이해해 보겠습니다.

> 실험 : 주사위를 던진다.
>
> 표본공간 : {1,2,3,4,5,6}
>
> 사건 A : 짝수가 나오는 사건 : {2,4,6}
>
> 사건 B : 3보다 이하인 수가 나오는 사건 : {1,2,3}
>
> 사건 C: 3보다 큰 수가 나오는 사건 : {4,5,6}
>
> (1) 사건 A의 여집합은 무엇인가? -> {1,3,5} (표본공간에서 {2,4,6} 제거)
>
> (2) 사건 A와 사건 B의 교집합은 무엇인가? -> {2} (사건 A와 사건 B가 공통적으로 가지고 있는 근원사건)
>
> (3) 사건 A와 사건 B의 합집합은 무엇인가? -> {1,2,3,4,6} (사건 A와 사건 B를 더한 후 중복되는 항목은 한 번만 표시)
>
> (4) 사건 B와 사건 C는 배반사건인가? -> 공유하는 근원사건이 없기 때문에 사건 B와 C는 배반산건

- 위의 예를 간단히 파이썬으로 구현한 코드입니다.

In [5]:

```
Set_ = set([1,2,3,4,5,6]) # 표본공간
Set_even =set([2,4,6]) # 사건 A- 짝수가 나오는 사건
Set_below3 = set([1,2,3]) #사건 B- 3보다 이하인 수가 나오는 사건
Set_above3 = set([4,5,6]) #사건 C- 3보다 큰 수가 나오는 사건

# 사건 A의 여집합
print(Set_.difference(Set_even)) #{1, 3, 5}
print(Set_ - Set_even) #{1, 3, 5}

# 사건 A와 사건 B의 교집합
print(Set_even.intersection(Set_below3)) #{2}
print(Set_even & Set_below3) #{2}

# 사건 A와 사건 B의 합집합
print(Set_even.union(Set_below3)) #{1, 2, 3, 4, 6}
print(Set_even | Set_below3) #{1, 2, 3, 4, 6}

# 사건 B와 사건 C는 배반사건인가?
print(Set_below3.intersection(Set_above3)) #set(): empty set
print(Set_below3 & Set_above3) #set(): empty set
```

```
{1, 3, 5}
{1, 3, 5}
{2}
{2}
{1, 2, 3, 4, 6}
{1, 2, 3, 4, 6}
set()
set()
```

조건부 확률과 베이즈 정리

- 출처 (https://direction-f.tistory.com/9)
- 두 개 이상의 사건이 있을 때 한 사건이 다른 사건의 확률에 영향을 미치는 경우를 본 적이 있으실 겁니다. 예를 들어 성인 남성과 남자 아동이 함께 있는 집단에서 임의로 한 사람을 뽑았을 때 그 사람이 성인 남성인 사건을 A라고 키가 180cm 이상일 사건을 B라고 해보겠습니다.
- 그렇다면 전체 집단에서 임의적으로 특정 인원을 뽑았을 때 성인 남성일 확률과 키가 180cm이상인 사람이 뽑혔을 때 성인 남성일 확률이 상이할 것이라는 것을 우리는 직관적으로 알 수 있습니다.

조건부 확률

- 위의 예와 같이 사건 B와 관련된 정보가 우선적으로(사전적으로) 주어졌을 때 사건 A의 변화된 확률을 "B가 주어졌을 때 사건 A의 조건부 확률"이라고 하며 P(A|B)로 표기 합니다. 조건부 확률을 계산하는 공식은 아래와 같습니다.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, P(B) > 0$$

- 위의 조건부 확률의 정의를 이용하면, 아래와 같은 식을 유도할 수 있습니다.

$$P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

- 만약 이 때 사건 A와 B의 교집합일 확률이 사건 A일 확률 P(A)와 사건 B일 확률 P(B)의 곱으로 표현된다면, 사건 A와 사건 B를 서로 독립이라고 할 수 있습니다.

$$P(A \cap B) = P(A) \cdot P(B)$$

- 만약 사건 A와 사건 B가 독립이라면 조건부 확률을 아래와 같이 표현됩니다.

$$P(A \cap B) = P(A) \; or \; P(B)$$

베이즈 정리

- 우리는 표본 공간이 교집합이 없는 사건(배반사건)들의 모임으로 구성될 수 있음을 알 수 있습니다.
- 예를 들어 직장인들을 "20대 미만", "20대 이상 30대 미만", "30대 이상"으로 교집합 없이 구성 할 수 있을 것입니다.
- 이와 같이 각 사건들이 배반사건들이면서 사건들의 합집합이 표본공간을 구성할 때, 이 때 각 사건을 표본공간의 분할이라고 부릅니다.
- 우리는 위에서 직장인을 "20대 미만", "20대 이상 30대 미만", "30대 이상"으로 구분했습니다. 이 사건들을 각각 A1, A2, A3라고 하겠습니다. 추가적으로 직장인들 중에서 담배를 피는 사건을 B라고 하겠습니다. A1, A2, A3는 교집합이 없기 때문에 직장인들 중 담배를 피는 사건은 아래와 같이 표현할 수 있습니다.

$$B = (B \cap A_1) \cup (B \cap A_2) \cup (B \cap A_3)$$

- 그렇다면 직장인들 중 담배를 필 확률은 아래와 같이 계산될 수 있습니다.

$$P(B) = (P(B|A_1)P(A)_1) + (P(B|A_2)P(A)_2) + (P(B|A_3)P(A)_3)$$

- 위와 같은 계산을 일반화 한 것이 총확률의 법칙이며, 아래와 같이 정의 됩니다.

$$P(B) = \sum_{i=1}^{n} P(A_i)P(B|A_i)$$

- 그렇다면 이제 우리가 지금까지 정리했던 조건부 확률에서 교집합을 정의한 식과 총확률의 법칙에서 정의된 식을 활용하여 베이즈정리를 아래와 같이 정의할 수 있습니다.

$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)} = \frac{P(A_i)P(B|A_i)}{P(A_1)P(B|A_1) + P(A_2)P(B|A_2) + \cdots + P(A_n)P(B|A_n)}$$

- 파이썬을 활용해서 공식을 통해 도출한 값과 실제로 손으로 푸는 값이 같은 결과를 보는지 확인해 보았습니다.

In [7]:

```
import pandas as pd


## 가상 데이터 생성
smoking_group = {"ten":{"non":10, "smoking" :0},
                 "twenty":{"non":5, "smoking" :5},
                 "thirty":{"non":2, "smoking" :8}}

table = pd.DataFrame(smoking_group).transpose()
display(table)

## 각각 확률 계산
P_ten = table.loc["ten"].sum()/table.values.sum()
P_twenty = table.loc["twenty"].sum()/table.values.sum()
P_thirty= table.loc["thirty"].sum()/table.values.sum()
P_smoking = table["smoking"].sum()/table.values.sum()
P_smoking_given_ten = table.loc["ten"]["smoking"]/table.loc["ten"].sum()
P_smoking_given_twenty = table.loc["twenty"]["smoking"]/table.loc["twenty"].sum
()
P_smoking_given_thirty = table.loc["thirty"]["smoking"]/table.loc["thirty"].sum
()



P_twenty_given_smoking = P_twenty*P_smoking_given_twenty/((P_ten*P_smoking_given
_ten)+(P_twenty*P_smoking_given_twenty)+(P_thirty*P_smoking_given_thirty))
print("P_twenty_given_smoking:{}".format(round(P_twenty_given_smoking,3))) ##0.3
85
print("담배피는 사람일 때 20대일 확률:{}".format(round(5/13,3)))  ##0.385
```

	non	smoking
ten	10	0
twenty	5	5
thirty	2	8

```
P_twenty_given_smoking:0.385
담배피는 사람일 때 20대일 확률:0.385
```

확률변수, 확률분포

- 출처 (https://direction-f.tistory.com/11)
- 실험을 통해 일어날 수 있는 모든 사건들의 집합인 표본공간은 사건들의 집합으로 표현할 수 있었습니다. 예를 들어 동전을 두번 던지는 실험을 했다고 가정하면 표본공간은 {HH, HT, TH, TT}으로 표현 할 수 있습니다.(H: 앞면, T: 뒷면)
- 이 때 우리는 앞면의 나온 수로 각 근원 사건을 표현할 수 있습니다.({2,1,1,0}) 이와 같이 표본공간의 사건들을 특정 수치로 표현할 수 있습니다.

확률변수

- 이처럼 각 사건에 수치를 대응시키는 것을 확률변수(Random variable)라고 합니다. 즉, 확률변수는 각각의 사건들에 실수값을 대응시키는 함수라고 정의 할 수 있습니다.
- 예를 들어보겠습니다. 세 사람이 있고 세 사람은 아이폰이나 갤럭시 중 하나를 가지고 있다고 가정해보겠습니다. 그렇다면 이 때 아이폰을 가지고 있는 근원 사건은 아래와 같이 총 8가지로 표현할 수 있습니다.

A	B	C
아이폰	아이폰	아이폰
아이폰	아이폰	갤럭시
아이폰	갤럭시	갤럭시
아이폰	갤럭시	아이폰
갤럭시	갤럭시	갤럭시
갤럭시	갤럭시	아이폰
갤럭시	아이폰	아이폰
갤럭시	아이폰	갤럭시

- 근원사건과 아이폰을 가지고 있는 사람 수(X)를 대응해보겠습니다.

A	B	C	X
아이폰	아이폰	아이폰	3
아이폰	아이폰	갤럭시	2
아이폰	갤럭시	갤럭시	1
아이폰	갤럭시	아이폰	2
갤럭시	갤럭시	갤럭시	0
갤럭시	갤럭시	아이폰	1
갤럭시	아이폰	아이폰	2
갤럭시	아이폰	갤럭시	1

- 이 때 우리는 아이폰을 가지고 있는 사람 수(X)는 근원사건을 특정 숫자로 대응하게 해주는 확률변수이며, 확률 값을 가지게 됩니다.
- 확률변수가 위의 예처럼 셀 수 있는 경우라면 "이산확률변수", 셀 수 없이 구간에서 연속인 확률변수는 "연속확률변수"라고 정의 할 수 있습니다.

확률분포

- 확률변수는 각각 확률 값을 가지게 됩니다. 확률변수가 가질 확률을 정해주는 관계를 확률분포(Probability distribution)이라고 부르며, 확률변수가 가지는 값과 그 확률변수에 대응하는 확률값을 나타내는 것입니다.
- 위의 예(아이폰을 가진 사람 수)를 활용하여 확률분포를 자세히 알아보겠습니다.

근원사건	확률변수	확률
(아이폰,아이폰,아이폰)	3	1/8
(아이폰,아이폰,갤럭시)/(아이폰,갤럭시,아이폰)/(갤럭시,아이폰,아이폰)	2	3/8
(갤럭시,갤럭시,아이폰)/(갤럭시,아이폰,갤럭시)/(아이폰,갤럭시,갤럭시)	1	3/8
(갤럭시,갤럭시,갤럭시)	0	1/8

- 위의 표에서 확인 할 수 있듯이 각 확률변수에 대응하는 확률로 분포는 표현됩니다. 이 때 확률분포의 값은 1보다 작아야 하며, 확률의 합은 1이 되어야 합니다.

확률분포의 기댓값(평균), 표준편차

- 출처 (https://direction-f.tistory.com/12)

확률변수의 기대값(평균)

- 표본자료에서 평균은 자료의 중심을 나타내는 대표적인 지표임과 동시에 그 자료를 설명하는 가장 대표적인 지표였습니다. 예를 들어 어떤 퀴즈 대회에서 상금으로 10,000원, 100,000원, 1,000,000원 10,000,000원을 지급한다고 하면 상금의 평균은 각 상금의 합을 4로 나눈 2,777,500원이 될 것이며, 퀴즈 대회에 참여한 사람들은 평균적으로 2,777,500원을 얻을 수 있을 것이라고 생각 할 수 있을 것입니다.
- 만약 10,000원, 100,000원, 1,000,000원 10,000,000원의 상금을 탈 확률이 다르다면 어떻게 될까요? 아마 우리가 퀴즈를 통해 평균적으로 얻을 수 있다고 생각하는 상금은 달라질 것입니다. 각 상금을 탈 확률이 아래의 표와 같다고 가정해보겠습니다.

상금	확률
10,000원	5/10
100,000원	3/10
1,000,000원	2/10
10,000,000원	1/10

- 이 때 우리가 퀴즈 대회를 통해 얻을 수 있는 상금의 기대 값은 아래와 같이 1,235,000원일 것입니다.

$$10,000 \cdot \frac{5}{10} + 100,000 \cdot \frac{3}{10} + 1,000,000 \cdot \frac{2}{10} + 10,000,000 \cdot \frac{1}{10} = 1,235,000$$

- 따라서 우리는 확률분포의 기대값을 아래와 같이 정의 할 수 있습니다.

$$E(X) = \mu = \sum(\text{확률변수가 취하는 값}) \times (\text{그 값을 가질 확률})$$

- 가질 수 있는 확률변수 값을 x, 그 값을 가질 확률을 $f(x)$로 표현하면 기대값은 다시 아래와 같이 표현 할 수 있습니다.

$$\sum x_i \cdot f(x)$$

확률변수의 표준편차

- 표준편차는 자료에서 얼마나 퍼져있는지를 나타내는 지표이며 아래와 같이 계산을 통하여 모집단의 분산과 표준편차 s를 도출했습니다.

$$s^2 = \frac{\Sigma(x_i - \bar{x})^2}{n}$$

- 확률변수의 표준편차도 위의 식과 거의 동일합니다, 다만 모집단의 자료 수(n)로 편차의 제곱의 합을 나눠주는 것이 아니라 그 값을 가질 확률을 곱해주는 것이 차이점입니다.
- 따라서, 확률 변수 X의 분산과 표준편차는 아래와 같이 정의 됩니다.

$$Var(X) = \sum(\text{편차})^2 \times \text{확률}$$
$$Sd(X) = \sqrt{Var(X)}$$

- 가질 수 있는 확률변수 값을 x, 그 값을 가질 확률을 $f(x)$로 표현하면 표준편차는 다시 아래와 같이 표현 할 수 있습니다.

$$Var(X) = E(X-\mu)^2 = \sum(x_i - \mu)^2 \cdot f(x)$$

$$Sd(X) = \sqrt{Var(X)}$$

- 분산을 손쉽게 계산하기 위한 간편식은 아래와 같습니다.

$$Var(X) = E(X^2) = (E(X))^2 = \sum x_i^2 f(x_i) - \mu^2$$

베르누이 시행과 이항 분포

- 실험을 통해 얻을 수 있는 결과가 두 가지만 있다고 생각해보겠습니다. 예를 들어 동전을 던지는 실험을 했을 때 우리가 얻을 수 있는 결과는 앞면(H)와 뒷면(T) 두 가지 뿐입니다. 예와 같이 두 가지의 결과만 반복해서 나오며, 아래와 같은 조건을 만족하는 경우 이를 베르누이 시행이라고 부릅니다.

> 1) 각 시행은 성공(S), 실패(F)의 두 결과만을 갖는다(우리가 흔히 사용하는 성공의, 실패의 의미와는 무관, 결과가 두 개 뿐임을 강조)
>
> 2) 각 시행에서 성공할 확률 $P(S) = p$, 실패활 확률 $P(F) = q = 1 - p$로 그 값이 일정함
>
> 3) 각 시행은 서로 독립으로 각 시행의 결과가 다른 시행의 결과에 영향을 미치지 않음

이항분포(Binomial distribution)

- 위와 같은 조건을 만족하는 베르누이 시행을 반복할 때에 일어나는 성공의 횟수를 X라고 하면 X는 확률변수가 됩니다. 이 때 확률변수 X가 따라는 확률분포를 모수가 (n, p)인 이항분포라고 하며, $X\ B(n, p)$로 표현하게 됩니다.
- 이항분포의 모수와 확률변수 X는 아래와 같이 정의 됩니다.

> n: 베르누이 시행의 반복 횟수
>
> p: 각 시행에서의 성공확률
>
> X: n번의 시행 중에서 성공의 횟수

- 만약 확률변수 X가 B(n,p)를 따르면, 성공횟수가 x일 때 확률함수는 아래와 같이 정의 됩니다.

$$P(X = x) = \binom{n}{x} p^n q^{n-x}$$

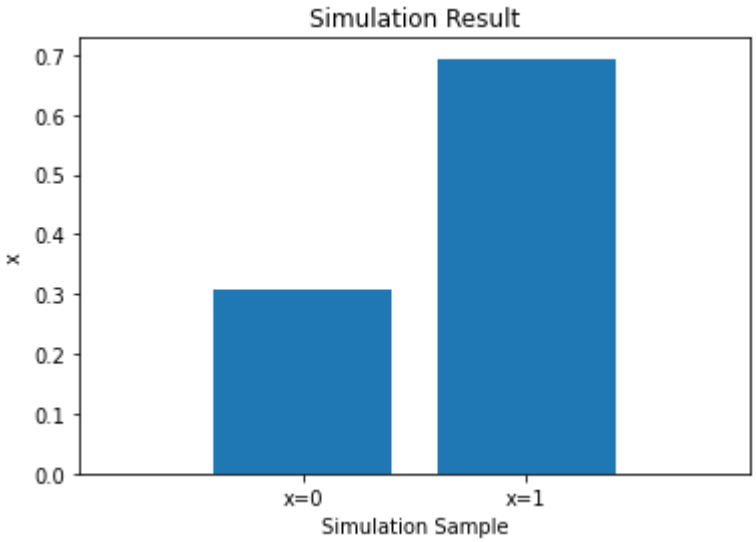
- 성공횟수가 x일 때 순서에 상관없이 성공횟수가 x이기만 하면 되기 때문에 성공횟수가 x회가 될 수 있는 조합의 경우의 수를 곱해주게 됩니다.
- 예를 들어 휘어진 동전이 있다고 가정해보겠습니다. 해당 동전은 앞면이 나올 확률이 0.7, 뒷면이 나올 확률이 0.3입니다. 그렇다면 동전을 10번 던져 앞면이 7번 나올 확률은 아래와 같습니다.

$$P(X = 7) = \binom{10}{7} 0.7^7 0.3^3 \approx 0.267$$

- 10번 던져 7번 앞면이 나올 확률은 약 26.7%정도입니다.
- Python으로 간단히 베르누이 시행을 시뮬레이션 해보도록하겠습니다.

In [9]:

```
## 필요 Moule Import
from scipy.stats import bernoulli
import matplotlib.pyplot as plt

## scipy 모듈을 활용한 이항분포 시뮬레이션
data_bern = bernoulli.rvs(size=1000, p=0.7) #(베르누이 1000번 시행, 성공할 확률 0.7)

## 시뮬레이션 결과 성공/실패 횟수 저장
s=len(data_bern[data_bern==1])
f=len(data_bern[data_bern==0])

## 시뮬레이션 결과 시각화

fail_or_sucess = [0, 1]
plt.bar(fail_or_sucess, height = [f/(s+f),s/(s+f)])
plt.xlim(-1, 2)
plt.xticks([0, 1], ["x=0", "x=1"])
plt.xlabel("Simulation Sample")
plt.ylabel("x")
plt.title("Simulation Result")
plt.show()
```



포아송 분포

- 출처 (https://direction-f.tistory.com/15)
- 만약 우리가 금은방을 운영하고 있다고 가정해보겠습니다. 퇴근 후 한 시간에 도둑이이 10명 올 확률은 어떻게 될까요? 100명이 올 확률은 어떻게 될까요?(도둑이 오면 안되겠지만요...) 이와 같이 일정 기간 동안에 확률이 낮은 특정 사건이 일어날 확률을 나타내기 위해 활용하는 것이 포아송 분포입니다.
- 다시 위의 예를 좀 더 깊게 들여보다면 저 확률을 이항분포로 나타낼 수 있지 않을까? 하는 생각이 드실 수도 있습니다. 다시 말하면 1분에 도둑이 올 확률이 0.01 오지 않을 확률이 0.99라면 이는 결과가 두 가지 뿐인 베르누이 시행으로 간주할 수 있습니다. 따라서 한 시간(60분)은 베르누이 시행을 60번 시행했다고 볼 수 있을 것입니다.
- 하지만 도둑이 1분에 한명만 오는 것이라고 한정할 수 없습니다. 즉 1초에 한명만 오는 시행으로 볼 수도 있고 0.01초에 한명만 오는 시행으로 볼 수도 있을 것입니다. 즉 베르누이 시행의 횟수를 무한대로 나타낼 수 있다는 것입니다.
- 따라서 포아송 분포는 베르누이 시행을 무한대로 시행한 경우로 나타내며, 아래와 같이 정의 할 수 있습니다.

$$P(X = x) = \lim_{x \to \infty} \binom{n}{x} p^x (1-p)^{n-x} = \frac{e^{-\lambda} \lambda^x}{x!}$$

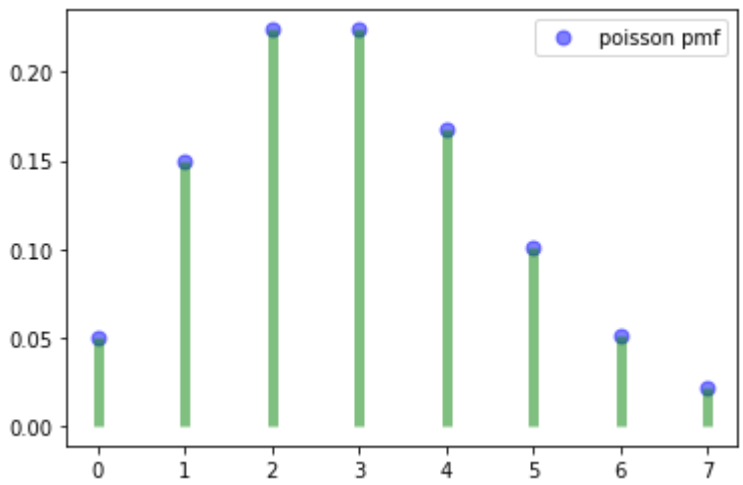
- 이 때 람다(λ)는 확률변수 X(도둑의 수)의 평균값이며, 도둑의 예를 들면 한시간동안 오는 평균적인 도둑 수입니다. (** 만약 성공할 확률 p가 0에 가깝지 않고, n이 무한대는 아니나 충분히 큰 수이면, 베르누이 시행을 n번 시행했을 때 성공 횟수는 정규분포에 가깝게 근사가 됩니다. 포아송 분포는 베르누이 시행해서 p가 아주 작고 n이 무한대로 표현될 수 있을 때, 근사된 분포입니다.)
- 다시 정리하면, 포아송 분포는 매 순간 사건 발생이 가능하나, 순간의 발생 확률이 아주 작은 경우에 활용합니다. 따라서 포아송 분포는 연속적인 시간에서 매 순간의 발생확률 대신, 특정 기간에 발생하리라 생각되는 평균 발생 횟수를 이용하여 실제로 발생하는 사건의 횟수의 관한 문제를 다루는 모형입니다.
- 실제로 포아송 분포를 적용하기 위해서는 다음과 같은 세 가지의 가정을 만족해야 합니다.

> 1) 사건의 평균 발생횟수는 구간의 길이에만 영향을 받는다.
>
> 2) 한 순간에 2회 이상의 사건이 발생할 확률은 거의 0에 가깝다.
>
> 3) 한 구간에서 발생한 사건은 다른 구간에서 발생한 사건에 영향을 주지 않는다.

Python으로 포아송 분포를 시뮬레이션 해보겠습니다.

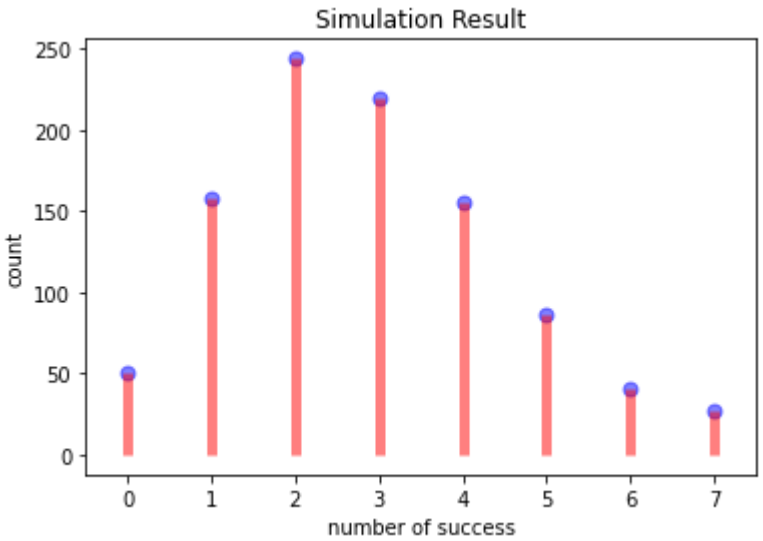
In [13]:

```
## 필요 Module Import
from scipy.stats import poisson
import matplotlib.pyplot as plt
import numpy as np
from collections import Counter
import pandas as pd

### 평균 3가정 하여 분포 추정
mu = 3
poisson.stats(mu, moments ="mvsk") ##mean, var, skewness, kurtosis 확인

## 추정한 분포 시각화
fig, ax = plt.subplots(1, 1)
x = np.arange(poisson.ppf(0.01, mu), poisson.ppf(0.99, mu))## ppf -> 누적 분포 함수
 확률이 0.01~0.99사이인 x 값들 반환
ax.plot(x, poisson.pmf(x, mu), 'bo', ms=7,alpha=0.5,label='poisson pmf') ## pmf
 - > 확률 질량 함수
ax.vlines(x, 0, poisson.pmf(x, mu), colors='g', lw=5, alpha=0.5) ## 수직선
ax.legend()
plt.show()
```



In [14]:

```
## 시뮬레이션 수행
poisson_ = poisson(mu) # 분포 추정
data_poisson=poisson_.rvs(1000) #추정한 분포 기반 Sample 1000개 생성
##(data_poisson=np.random.poisson(3, 1000) ## numpy 기반 Sample 1000개 생성)

## 성공횟수(x)별 횟수 도출 및 Dataframe 저장
count = Counter(data_poisson)
count_data = pd.DataFrame.from_dict(count, orient="index").reset_index()
count_data =count_data.rename(columns={"index":"number", 0:"count"})

## 성공횟수(x)별 정렬
count_data=count_data.sort_values(by=["number"], ascending=True)

## 시각화
plt.plot(count_data["number"], count_data["count"],'bo', ms=7,alpha=0.5)
plt.vlines(count_data["number"],0, count_data["count"],colors='r', lw=5, alpha=
0.5)
plt.xlabel("number of success")
plt.ylabel("count")
plt.title("Simulation Result")
plt.xlim(-0.5, 7.5)
plt.show()
```



연속확률분포, 정규분포, 표준정규분포

- 우리가 셀 수 있는 확률변수들의 분포를 이산확률분포라고 불렀습니다. 이러한 이상확률분포 중에는 대표적으로 이항분포와 포아송 분포가 있었습니다.
- 이산확률분포와는 다르게, 정규분포는 0과 1사이의 임의의 실수처럼 셀 수 없는 연속적인 값을 가지는 연속확률분포입니다. 정규분포는 연속적인 값을 가지는 확률변수의 분포를 나타내는데 가장 많이 적용되고 있는 분포입니다. 정규분포 외에도 데이터에 따라 더 적합한 분포들도 많이 제안되어 왔지만, 여전히 가장 강력하고 일상적으로 적용되고 있는 분포입니다.

연속확률분포

- 본격적으로 정규분포에 대해서 알아보기 전에, 연속확률분포에 대해서 알아보겠습니다. 연속확률분포는 이산확률분포처럼 발생가능한 모든 값에 확률을 대응/나열하여 표현하기가 어렵습니다. 발생 가능한 값이 무한대이기 때문입니다.
- 따라서 연속확률분포는 주어진 구간에서 확률이 어떻게 분포하는지를 판단하게 되고, 어느 구간의 확률이 작은지, 큰지를 나타낼 수 있는 함수를 이용하게 됩니다.
- 연속확률변수 X의 확률분포는 확률의 밀도를 나타내는 확률밀도함수에 의해 결정되며, 확률밀도함수는 아래와 같은 조건을 만족해야합니다.

$$1) \forall x,\ f(x) \geq 0$$

$$2) P(a \leq X \leq b) = \int_a^b f(x)dx$$

$$3) P(-\infty < X < \infty) = \int_{-\infty}^{\infty} f(x)dx = 1$$

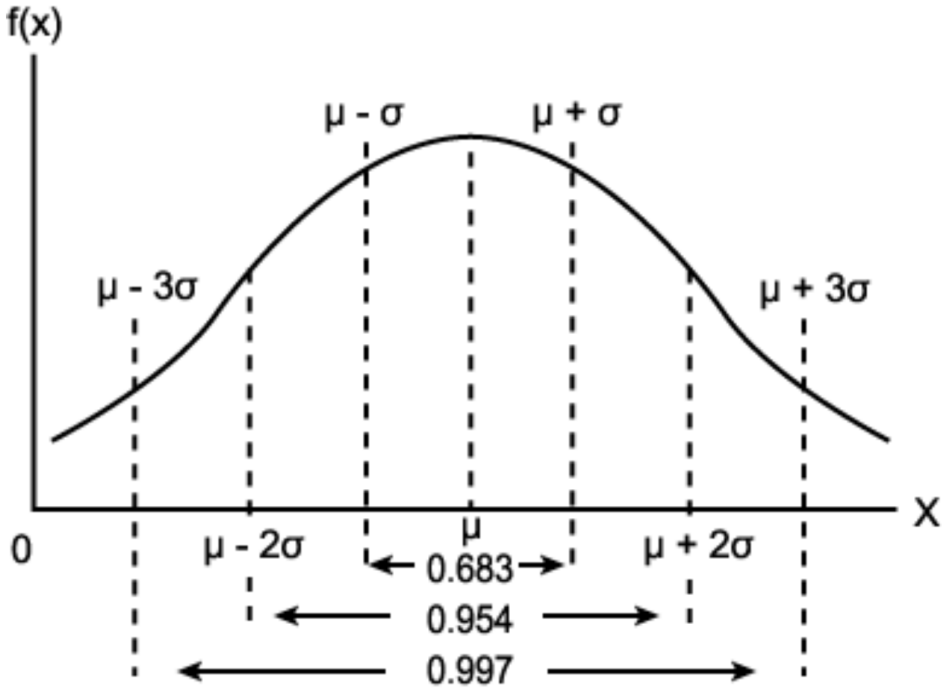
- 연속확률분포에서는 정의 2) 에서 보는것과 같이 특정값을 가질 확률은 0입니다. 아래의 식처럼 구간의 크기가 0이기 때문입니다.

$$P(3 < X < 3) = \int_3^3 f(x)dx = 0$$

- 따라석 연속확률분포에서 엄밀하게 확률의 값을 정의하기 위해서는 "특정 구간"이 주어져야 합니다.(확률밀도함수의 값이 확률이 아님을 주의해야 합니다.)

정규분포(normal distribution)

- 정규분포는 "평균", "분산"에 의해서 그 분포가 확정됩니다. 그 확률밀도함수의 대략적인 특성은 다음과 같습니다.

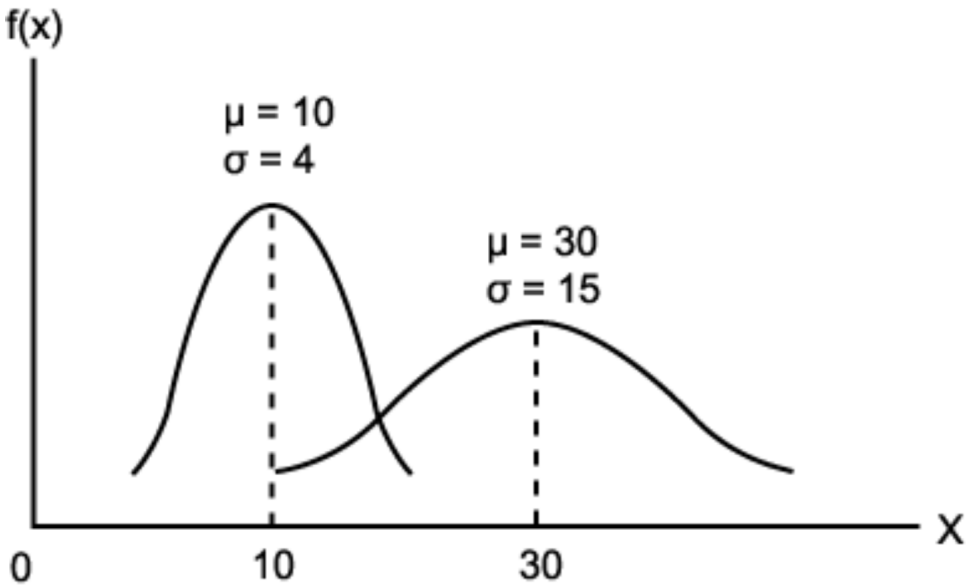


출처 (https://sites.google.com/site/nmrstudy/statistics/normalization)

- 위의 사진에서 보는 것같이 확률 변수 X가 평균으로부터 1표준편차, 2표준편차, 3표준편차 사이에 있을 확률은 아래와 같습니다.

$$P(\mu - \sigma \le X \le \mu + \sigma) = 68.27\%$$
$$P(\mu - 2\sigma \le X \le \mu + 2\sigma) = 95.45\%$$
$$P(\mu - 3\sigma \le X \le \mu + 3\sigma) = 99.73\%$$

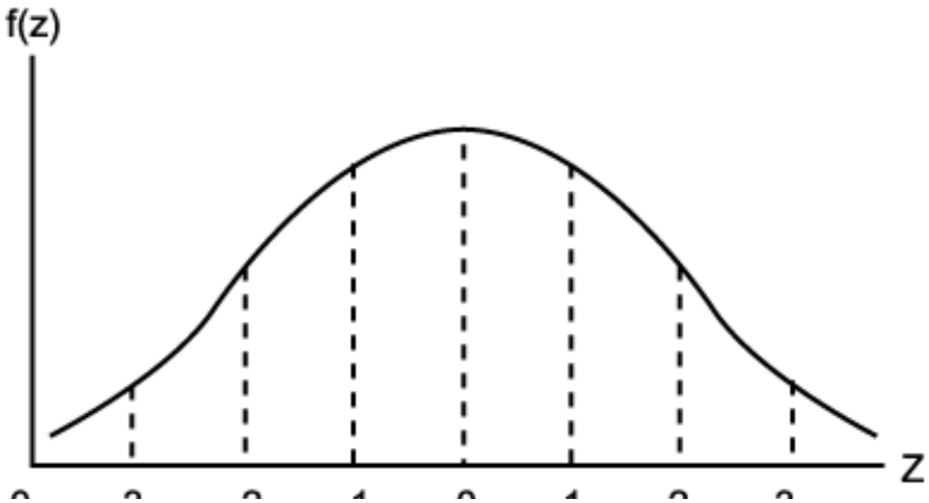
- 확률변수 X가 평균 μ, 표준편차 σ인 정규분포를 따를 때 주로 $N(\mu, \sigma^2)$로 표현하게 됩니다. 표준편차 σ가 커질수록 정규분포의 모형은 옆으로 퍼지게 됩니다. 즉 퍼진정도가 커지게 됩니다.



출처 (https://sites.google.com/site/nmrstudy/statistics/normalization)

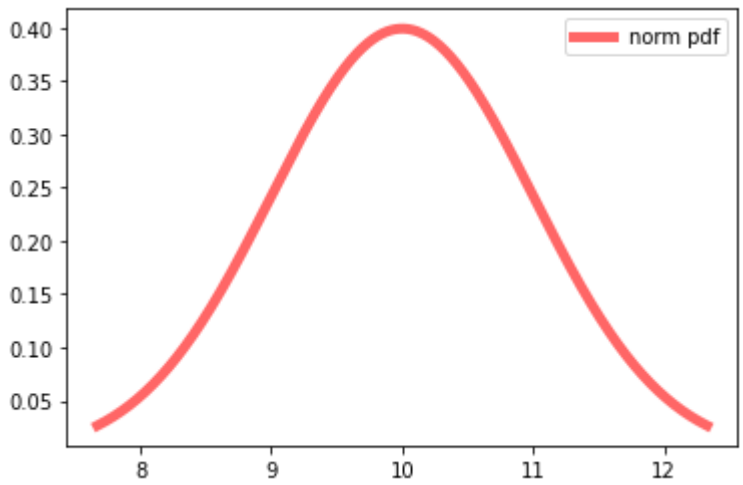
표준정규분포(Standard normal distribution)

- 표준정규분포는 평균이 0이고 표준편차가 1인 정규분포를 나타냅니다. 표준정규분포를 가지는 확률변수를 Z라고 표현했을 때 확률변슈 Z는 0을 중심으로 대칭인 분포를 가지게 됩니다.



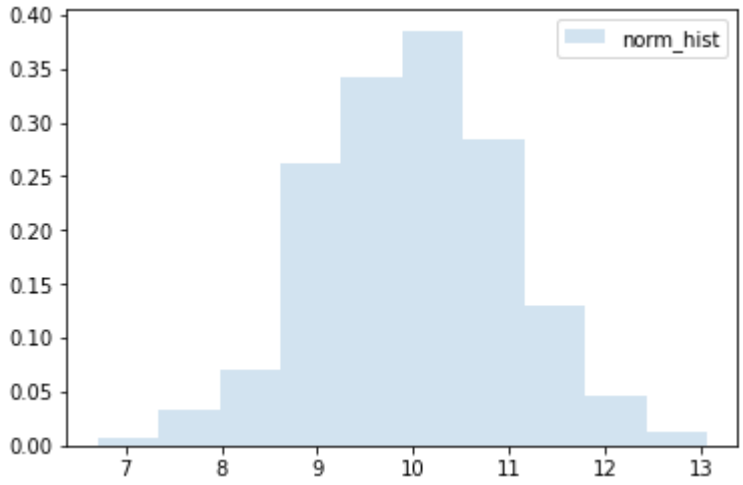
In [15]:

```
## 필요 Module Import

from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

mu = 10
std = 1
norm_ = norm(mu,std) ## 평균 10, 표준편차 1인 정규분포 생성

## 시각화
x = np.linspace(norm_.ppf(0.01), norm_.ppf(0.99),1000) ##  1%일때 x값, 99%일때 x값
 사이의 x값 생성
plt.plot(x, norm_.pdf(x),'r-', lw=5, alpha=0.6, label='norm pdf')
plt.legend()
plt.show()
```



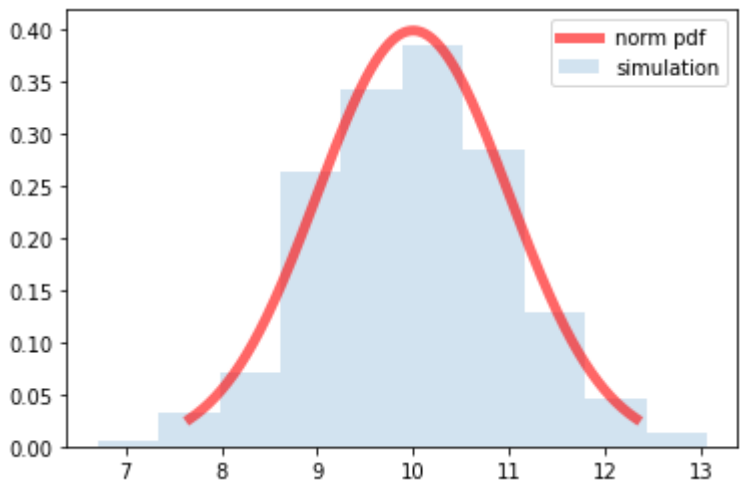
In [16]:

```
data_norm = norm_.rvs(1000) ## 평균 10, 표준편차 1의 정규분포를 따르는 데이터 생성
plt.hist(data_norm, density =True, histtype = "stepfilled", alpha=0.2, label =
"norm_hist")
plt.legend()
plt.show()
```



In [17]:

```
## 동시에 시각화 표현
fig, ax = plt.subplots(1, 1)
ax.plot(x, norm_.pdf(x),'r-', lw=5, alpha=0.6, label='norm pdf')
ax.hist(data_norm, density =True, histtype = "stepfilled", alpha=0.2, label = "s
imulation")
ax.legend(loc="upper right")
plt.show()
```

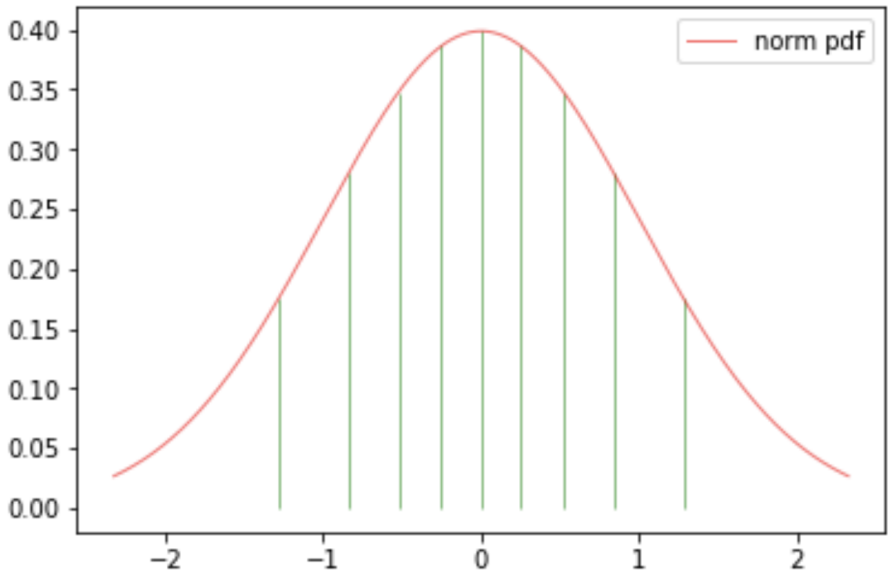


정규점수, 정규확률그림

- 출처 (https://direction-f.tistory.com/20)
- 우리가 표본을 추출하였을 때, 정규분포를 따른다고 가정이 맞는지 잘 못 됐는지 어떻게 판단 할 수 있을까요? 해당 가정을 쉽게 추정해 볼 수 있는 방법으로 정규점수그림 또는 정규확률그림이란 것이 있습니다.

정규점수

- 정규점수라는 것은 표준정규분포(평균 0, 표준편차 1)에서의 이상적인 표본을 말합니다. 다시 말하면, 표준정규분포의 확률밀도함수를 등확률 구간으로 나누어 주는 경계값(z값)을 의미합니다. 만약 우리가 표본이 있다고 가정한다면, 평균 근처에 값들의 빈도가 높아야 정규분포에 가깝다고 판단 할 수 있을 것입니다.



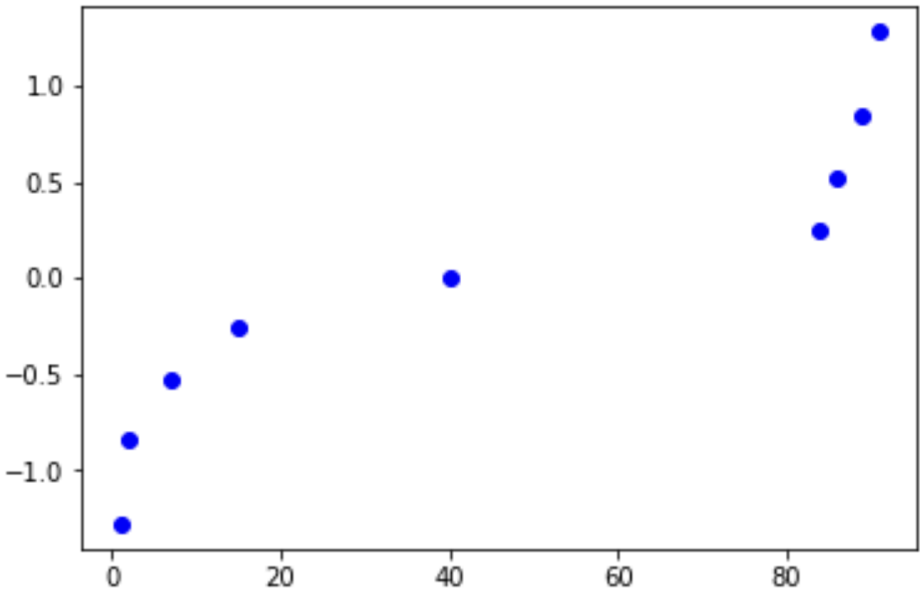
- 위에 그림을 보시면 초록색 줄 사이에 값들은 등확률입니다. 즉 줄 사이에 넓이들이 같습니다. 만약 우리가 9개의 표본을 가지고 있다면, 초록색 줄과 x축이 만나는 점(여기가 정규점수 입니다)에 9개의 표본을 정규화 한 값이 위치해있으면, 우리의 표본 9개가 정규분포를 따른다고 볼 수 있겠습니다.

정규확률그림

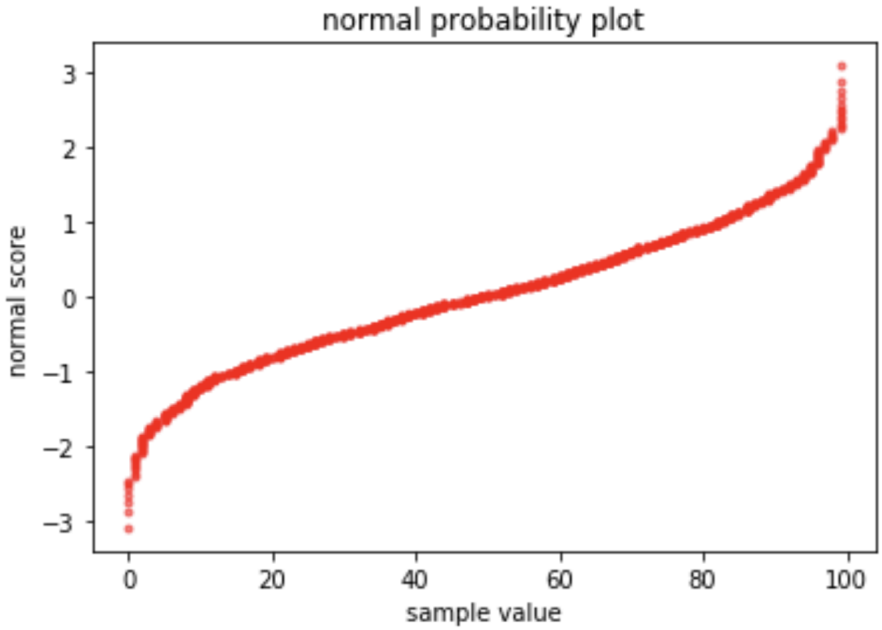
- 정규확률그림은 위의 정규점수와 실제 표본을 보기 쉽게 표현한 그래프입니다. 정규확률그림을 그리는 순서는 아래와 같습니다.

> (1) 표본을 작은 것부터 크기순으로 나열합니다.
>
> (2) 각 자료에(순서)에 해당하는 정규점수를 계산합니다.
>
> (3) 같은 순선의 자료와 정규점수를 2차원 그래프로 나타냅니다.

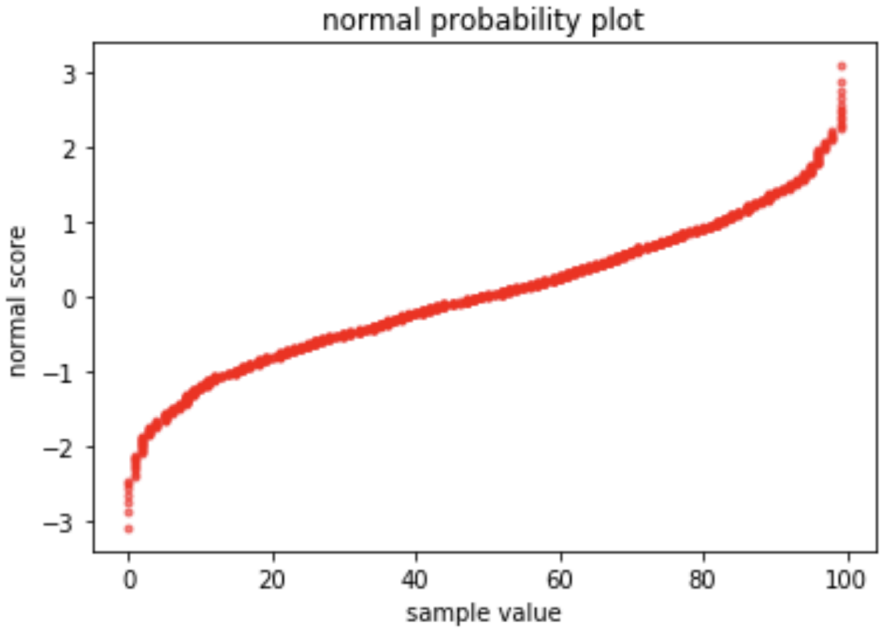
- 예를 들어 1,2,7,15,40,84,86,89,91 의 표본을 가지고 있다고 가정해보겠습니다. 그렇다면 확률을 10등분하는 기준점, 정규점수 9개는 아래와 같습니다. -1.28, -0.84, -0.52, -0.25, 0, 0.25, 0.52, 0.82, 1.28 이를 이용해서 Plot을 해보겠습니다.



- 그림을 확인하시면 직선 형태가 이루어 지지 않는 다는 것을 확인 하 실 수 있습니다. 저 점들의 형태가 직선에 가까울 수록 정규성을 만족한다고 판단합니다. 사실 9개 정도의 데이터로 정규성을 만족하는 것은 쉽지 않기 때문에, 약 1000개정도의 Sample을 생성하여, 정규확률그림을 그려보겠습니다.



- 위에 그림은 Random으로 생성하여 그려본 것입니다. 극단값에 갈 수록 곡선의 모양을 보이면서 정규분포와 멀어지는 것 같습니다.
- 이제 표준정규분포를 가정하고 Sample을 생성하여 그림을 그려 보겠습니다.

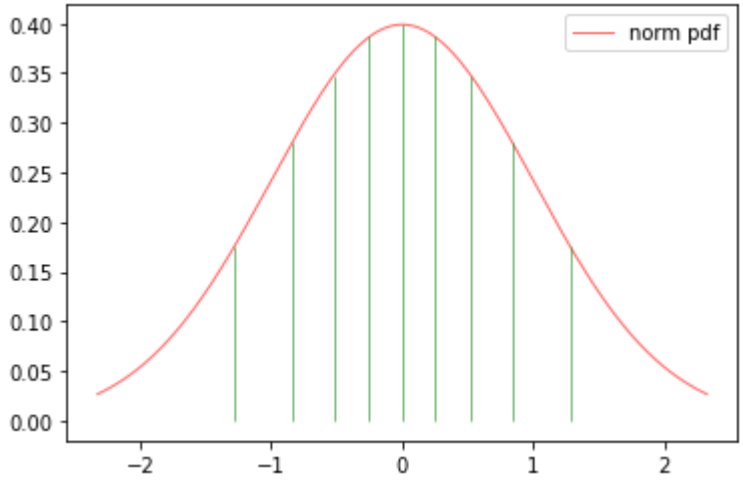


- 거의 직선에 가까운 것을 확인 할 수 있습니다. 해당 표본 자료는 정규성을 만족한다고 볼 수 있을 것입니다.
- 아래는 정규 확률 그림을 그리기 위한 Python 코드 입니다.

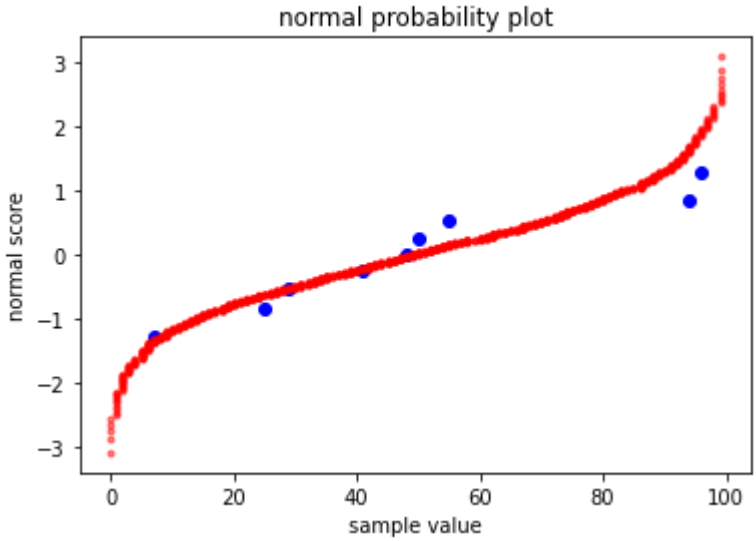
In [18]:

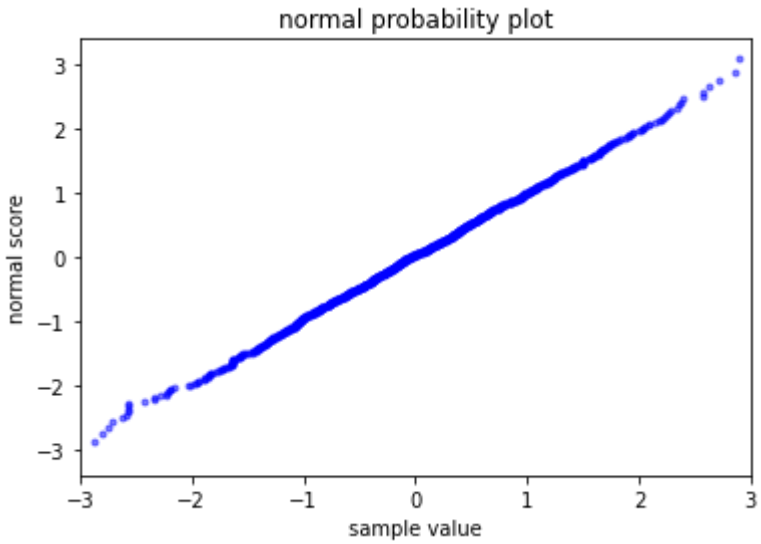
```
# module
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# 표준정규분포 생성
mu = 0
std = 1
norm_ =norm(mu,std)

## Sample이 9개일때 정규점수 계산
x_list =[]
for i in range(9):
    a = (i+1)/10
    x_list.append(norm_.ppf(a))

## 정규점수 시각화
x = np.linspace(norm_.ppf(0.01), norm_.ppf(0.99),1000)
plt.plot(x, norm_.pdf(x),'r-', lw=1, alpha=0.6, label='norm pdf')
plt.vlines(x_list, 0, norm_.pdf(x_list),colors='g', lw=1, alpha=0.5)
plt.legend()
plt.show()

## Sample이 9개 일때 정규확률 그림
data = np.random.randint(100, size = 9)
data=np.sort(data)
print(data)
plt.plot(data, x_list, "bo")

## Random Sample 999개일때 정규확률 그림
x_list =[]
for i in range(999):
    a = (i+1)/999
    x_list.append(norm_.ppf(a))

data = np.random.randint(100, size = 999)
data=np.sort(data)
plt.plot(data,x_list, "ro" ,alpha=0.5, ms= 3)
plt.title("normal probability plot")
plt.ylabel("normal score")
plt.xlabel("sample value")
plt.show()


## 표준정규분포로 Sample 999개 생성후 정규확률 그림
data_norm = norm_.rvs(999)
data_norm = np.sort(data_norm)
plt.plot(data_norm, x_list, "bo" ,alpha=0.5, ms= 3)
plt.xlim(-3,3)
plt.title("normal probability plot")
plt.ylabel("normal score")
plt.xlabel("sample value")
plt.show()


## Scipy 모듈을 활용한 정규확률 그림
```

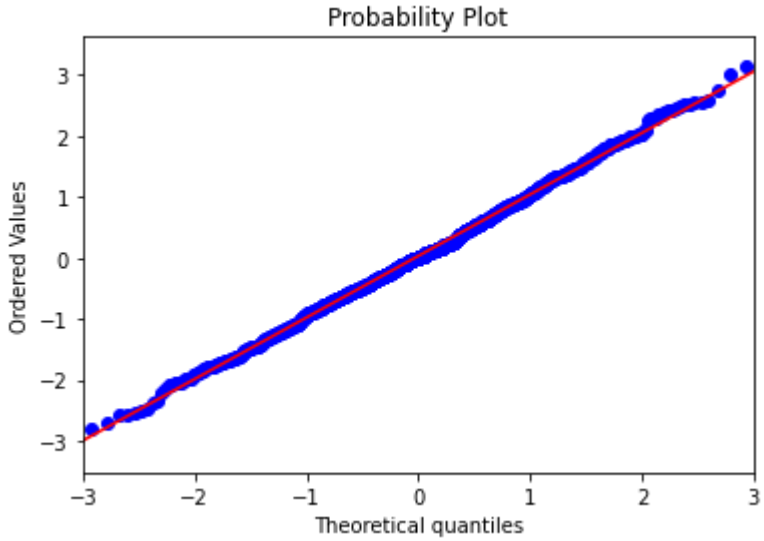
```
from scipy import stats

x= norm_.rvs(999)
res=stats.probplot(x, plot=plt)
plt.xlim(-3,3)
plt.show()

## Scipy 모듈을 경유 x축이 이론적인 값(normal score) y축이 sample value
```




```
[ 7 25 29 41 48 50 55 94 96]
```






Probability Plot
Ordered Values
Theoretical quantiles
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

표본평균의 분포와 중심극한

- 출처 (https://direction-f.tistory.com/21)
- 주어진 표본으로부터 모집단의 특성을 파악하는 것을 "추론"이라고 하며 어떻게 보면 통계학의 가장 중심이 되는 것이라고 볼 수 있습니다.
- 이 때, 모집단의 특성을 수치적으로 표현하는 것을 모수(Parameter)라고 합니다. 이러한 모수를 추정하기 위해서는 모집단 전체를 다 조사해야합니다. 하지만 모집단을 전부 조사하는 것은 일반적으로 어려운 일입니다.
- 따라서 제한된 표본으로부터 표본에서 적절한 양을 계산하여 활용하게 되는데, 이를 통계량(Statistic)이라고 부릅니다. 통계량은 표본의 관측값들에 의해 정의되는 양을 뜻합니다.
- 그렇다면 통계량은 모집단이 동일하더라도 표본이 바뀔때마다 바뀌는 양이 되게 됩니다. 그러므로 여러번의 표본을 뽑으면 통계량도 특정 확률분포를 갖게 됩니다. 이때 이 확률분포를 표집분포(Sampling distribution)이라고 합니다.

표본평균의 분포

- 예를 들어서 어느 모집단이 {1,2,3}으로 이루어졌으며 각 수치가 발생할 확률은 1/3으로 동일하다고 가정해보겠습니다. 그 다음 두 개의 값(X1, X2)을 복원추출하여 표본을 구성하고 해당 표본의 평균값의 분포를 추정해보겠습니다. 이 때 각 X1, X2는 집단 {1,2,3}을 가지고 각 수치가 발생할 확률이 1/3으로 모집단의 분포와 동일합니다.
- 모집단이 3개이고 복원추출이기 때문에 나올 수 있는 경우의 수는 3*3으로 총 9개이며, 아래와 같은 분포를 가지게 됩니다.

조합	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
표본평균	1	1.5	2	1.5	2	2.5	2	2.5	3
확률	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9

- 위와 같이 우리는 표본평균의 표집분포를 구성할 수 있음을 확인 할 수 있었습니다.
- 마찬가지로, 평균이 μ이고 표준편차가 σ인 모집단으로부터 크기가 n인 표본($X_1, X_2, \cdots, X_n$)을 추출 했을 때 표본평균의 기대값과 분산도 계산될 수 있으며, 아래와 같습니다. (각 표본 값 X_1, X_2는 확률변수로써 모집단의 분포를 따릅니다.)

$$E(\bar{X}) = \frac{1}{n}[E(X_1) + E(X_2) + \cdots + E(X_n)] = \mu$$

$$Var(\bar{X}) = \frac{1}{n^2}[Var(X_1) + Var(X_2) + \cdots + Var(X_n)] = \frac{\sigma^2}{n}$$

$$sd((\bar{X})) = \sqrt{Var} = \frac{\sigma}{\sqrt{n}}$$

중심극한정리

- 위의 식에서 확인 할 수 있듯이, 표본평균의 분포의 평균은 모집단의 중심 μ와 일치합니다. 만약 모집단이 정규분포가 아닌경우 표본평균의 분포는 모집단의 분포에 따라 다르게 나타나기도 합니다. 하지만, 표본의 크기 n이 큰 경우에는 표본평균의 분포는 모집단의 분포와 무관하게 근사적으로 정규분포를 따르게 됩니다. 이것이 중심극한정리입니다.
- 다시 정리하면
- 모집단의 평균이 μ이고 표준편차가 σ일 때, 임의로 추출한 표본의 표본평균은 표본의 크기 n이 충분히 클 때 (통상 적으로 30 이상) 근사적으로 정규분포를 따르게 되며, 그 때 평균은 μ이고 분산은 σ/√n이 됩니다. 이를 표준화 하여 나타내면 아래와 같이 표현됩니다.

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1)$$

통계적 추론(점 추정)

- 출처 (https://direction-f.tistory.com/24)
- 통계적 추론이란 우리가 가지고 있는 표본으로부터 모집단의 특성을 추론하는 것을 말합니다. 다시 말하면, 통계적 추론은 표본으로부터 모집단의 특성을 유도하고 그 특성이 옳은지 그른지를 판단하는 것입니다.
- 모집단의 특성을 추정하는 것에는 점 추정(Point Estimiation)과 구간 추정(Interval Estimation)이 있습니다. 점추정은 모집단의 특성을 나타내리라 생각하는 하나의 값을 추정하는 것이고, 구간 추정은 하나의 값만을 추정하는 것이 아니라 모수를 포함하리라 생각하는 적절한 구간을 추정하는 것입니다.

점 추정

- 모수를 추정하기 위해 모집단에서 크기가 n인 표본을 추출한다고 가정해보겠습니다. 그렇다면, 표본의 평균은 아래와 같을 것입니다.

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n}$$

- 이때 표본의 평균은 추정량(estimator)가 됩니다. 즉, 표본이 어떻게 구성되었는지에 따라서 값이 달라지는 것입니다. 이제 추정량을 이용하여 추론한 모집단 특성값을 추정치(estimate)라고 합니다.
- 위에서 말한 것처럼 추정량은 표본에 따라 변하는 값입니다. 이러한 수치들의 변화의 정도는 추정량의 정확도와 관련이 있는데, 이 정확도를 추정하는 하나의 도구로 표준오차(standard error)라는 것을 활용합니다.
- 표준오차(standard error)는 추정량의 표준편차 입니다.
- 전 포스팅 (https://direction-f.tistory.com/21)에서 표본평균의 표준편차는 평균이 μ이고 표준편차가 σ인 모집단이 주어졌을 때 아래와 같았습니다.

$$sd(\bar{X}) = \frac{\sigma}{\sqrt{n}}$$

- 표준오차는 표본평균의 표준편차와 같습니다. 다만 모집단의 표준편차 σ가 주어지지 않은 경우가 대부분이기 때문에 이 때는 표본자료의 표준편차를 σ대신하여 활용합니다. 표본자료의 표준편차(s) (https://direction-f.tistory.com/2)는 아래와 같이 도출했습니다. (표본자료의 표준편차와 표본평균의 표준편차는 다른 것임을 유의해야 합니다.)

$$s^2 = \frac{\Sigma(x_i - \bar{x})^2}{n-1}$$
$$s = \sqrt{s^2}$$

- 예제를 통해 확인해보겠습니다.
- {2,3,4}라는 표본이 주어졌다고 가정해보겠습니다. 그렇다면 표본의 평균은 3이고 표본의 표준편차는 아래와 같은 계산을 통해 1이 됩니다.

$$s^2 = \frac{(2-3)^2 + (3-3)^2 + (4-3)^2}{3-1} = 1$$

- 이제 이 값을 모집단의 표준편차를 대신하여 표준오차를 계산합니다, 그렇다면 표준오차는 표본의크기의 제곱근을 나눠 1/√3 이 됩니다. (만약 모집단의 표준편차 σ를 알고 있다면 모집단의 표준편차를 활용하여 계산하면 됩니다.)

통계적 추론(구간 추정)

- 출처 (https://direction-f.tistory.com/25)
- 점 추정은 모집단의 특성을 나타내는 하나의 값을 추정하는 것이었습니다. 반면 구간 추정(Interval Estimation)은 추정량(Estimator)의 분포를 활용하여 모집단의 특성을 나타내는 값을 포함하리라고 생각되는 구간을 추정하는 것입니다.

구간 추정

- 우리가 구간을 추정을 통해 모수 값(모집단의 특성을 나타내는 값)을 포함하는 구간을 추정하는데, 이 구간을 신뢰구간(Confidence Interval)이라고 부릅니다.
- 신뢰구간은 상한과 하한이 있고 (L,U)형태로 가지게 됩니다. 이 때 L이 -∞이고 U가 ∞라면 모수 값이 어떻게 되더라도 신뢰구간에 포함되게 될 것입니다.
- 따라서 우리는 상한과 하한값을 제한 할 필요가 있습니다. 이 필요로 인해서 우리가 흔히 들어본 95% 신뢰구간, 90% 신뢰구간이란 용어가 나오게 됩니다.
- 이와 같이 구간을 제한하게 되면 모수 값이 신뢰구간에 들어갈 확률이 100%가 아니라 95%, 90%로 제한되게 됩니다. 이 때 95%와 90%같은 수를 우리는 신뢰수준(level of confidence)이라고 부릅니다.
- 만약 평균이 μ이고 표준편차가 σ인 모집단이 주어졌을 때 표본평균의 분포는 평균이 μ이고 표준편차가 σ/√n을 따르게 됩니다. 우리는 이제 표본평균의 평균과 표준편차를 활용하여 표준화된 표본평균의 분포를 구할 수 있게 되고 표준화된 표본평균의 분포는 아래와 같습니다.

$$\frac{\bar{X}-\mu}{\sigma/\sqrt{\mu}} \sim N(0,1)$$

- 그렇다면 특정 값 α가 주어졌을 때(일반적으로 α값은 0.05, 0.1등으로 주어지게 됩니다.) 아래의 식을 만족하게 됩니다.

$$P(|\frac{\bar{X}-\mu}{\sigma/\sqrt{\mu}}| < Z_{\alpha/2}) < 1-\alpha$$
$$P(-Z_{\alpha/2}\cdot\frac{\alpha}{\sqrt{n}} < \bar{X}-\mu < Z_{\alpha/2}\cdot\frac{\alpha}{\sqrt{n}}) = 1-\alpha$$
$$P(\bar{X}-Z_{\alpha/2}\cdot\frac{\alpha}{\sqrt{n}} < \mu < \bar{X}+Z_{\alpha/2}\cdot\frac{\alpha}{\sqrt{n}}) = 1-\alpha$$

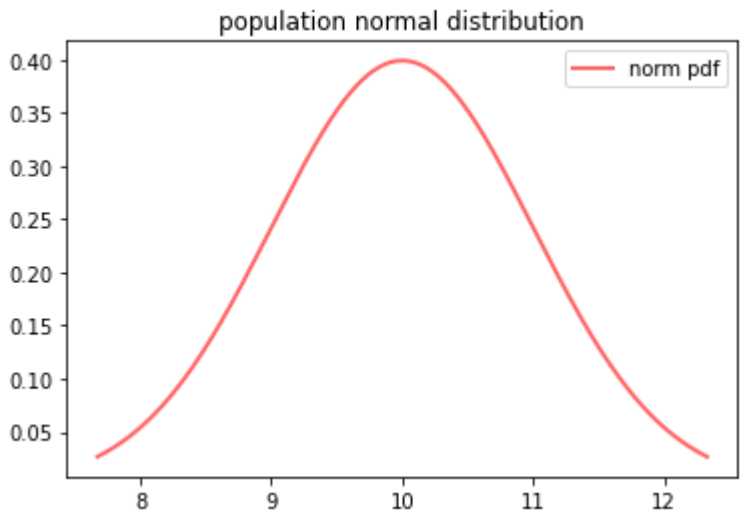
- 위에 식에서 확인할 수 있듯이, 모집단의 표준편차 σ가 주어지게 되면, 모집단 평균 μ에 대해서 100(1-α)% 신뢰구간은 아래와 같이 됩니다.

$$(\bar{X}-Z_{\alpha/2}\cdot\frac{\alpha}{\sqrt{n}}, \bar{X}+Z_{\alpha/2}\cdot\frac{\alpha}{\sqrt{n}})$$

- 이때 모집단의 표준편차가 주어지지 않으면 우리는 표본자료의 표준편차를 이용하여 신뢰구간을 추정하게 됩니다.(표본자료의 표준편차는 링크 (https://direction-f.tistory.com/2)를 참조하시면 됩니다.)
- 신뢰구간을 추정할 때 특이할 점은, 신뢰수준을 높일수록 구간이 길어진다는 것이고 표본의 크기 n이 커질수록 구간이 짧아져 모집단 평균 μ에 대해서 더 정확한 정보를 얻을 수 있다는 것입니다.
- 주의할 점은 95%신뢰구간이라고 한다고 하여, 모수가 95%확률 포함된다는 개념은 아닌 것입니다. 신뢰구간은 표본이 어떻게 추출되었냐에 따라 변하는 가변적인 구간입니다. 따라서 우리는 해석에 주의를 해야합니다.
- 따라서 신뢰구간은 동일한 방법으로(같은 크기로) 100번의 표본을 추출했을 때, 계산되는 100개의 신뢰구간 중 모평균을 포함한 신뢰구간들의 숫자가 95개 정도 된다라고 해석하면 됩니다.출처 (https://m.blog.naver.com/PostView.nhn?blogId=vnf3751&logNo=220823007712&proxyReferer=https:%2F%2Fwww.google.com%2F)
- Python을 활용 신뢰구간을 추정해보도록 하겠습니다.

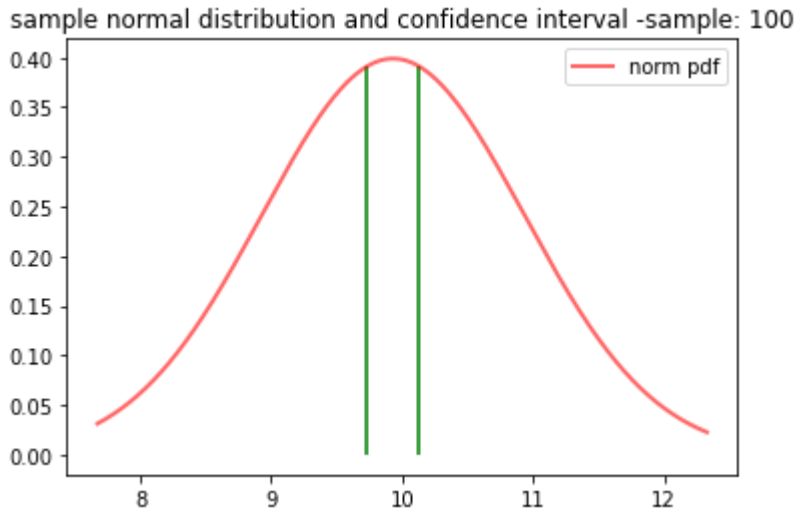
In [19]:

```
## 필요 Module import
import scipy.stats
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np


## 평균 10, 표준편차 1인 정규 분포 생성 및 시각화

mu = 10
std = 1
norm_ = norm(mu,std) ## 평균 100, 표준편차 10인 정규분포 생성

## 시각화
x = np.linspace(norm_.ppf(0.01), norm_.ppf(0.99),1000) ##  1%일때 x값, 99%일때 x값
 사이의 x값 생성
plt.plot(x, norm_.pdf(x),'r-', lw=2, alpha=0.6, label='norm pdf')
plt.title("population normal distribution")
plt.legend()
plt.show()
```



In [20]:

```
## sample 수 : 100개라 가정
data_norm = norm_.rvs(100) ## 앞에서 생성한 정규분포에서 100개 Sampling

## 표본의 통계량(estimator) 추정
mu_sample = np.mean(data_norm)
std_sample = np.std(data_norm, ddof=1)

## 상한, 하한 추정(95% 구간 z=1.96)
L_ = mu_sample - 1.96*(std_sample/np.sqrt(100))
U_ = mu_sample + 1.96*(std_sample/np.sqrt(100))

new_norm = norm(mu_sample,std_sample) ## 시각화를 위한 Sample 정보를 이용한 정규분포 생성

##시각화
plt.vlines(L_, 0, new_norm.pdf(L_), colors="g")
plt.vlines(U_, 0, new_norm.pdf(U_), colors="g")
plt.plot(x, new_norm.pdf(x),'r-', lw=2, alpha=0.6, label='norm pdf')
plt.title("sample normal distribution and confidence interval -sample: 100")
plt.legend()
plt.show()
print(L_, mu_sample, U_) ## 결과 : 9.884552789207058 10.094701914191354 10.304851
03917565
```




```
9.734216075260212 9.930259015190048 10.126301955119883
```

In [21]:

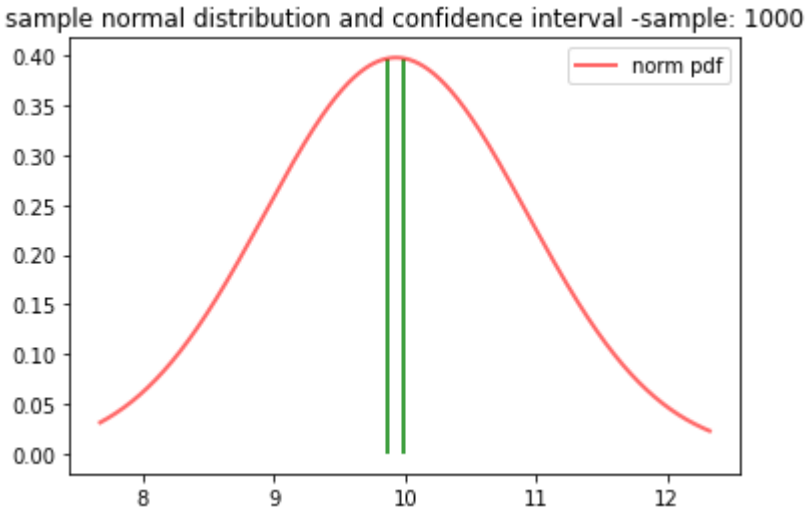
```
## Confidence Interval을 계산하기 위한 함수 생성

mu2 = 0
std2 = 1
standard_norm = norm(mu2,std2)

def mean_confidence_interval(data, confidence=0.95):
    a = np.array(data)
    m, se = np.mean(a), scipy.stats.sem(a) ## 표본의 표준편차를 활용하여 표준오차 계산
    h = se * standard_norm.ppf((1+confidence)/2) ## 신뢰구간이 95%이고 α=0.05일 때, 누적확률이 97.5%인 z값 계산
    return  m-h,m, m+h

mean_confidence_interval(data_norm) ## 결과 (9.884556650746545, 10.094701914191354, 10.304847177636164)

##위에서 손으로 계산한 값과 일치함을 확인
```

Out[21]:

```
(9.734219677595238, 9.930259015190048, 10.126298352784858)
```

In [22]:

```
## sample 수 : 10000개라 가정
data_norm = norm_.rvs(1000) ## 앞에서 생성한 정규분포에서 1000개 Sampling

## 표본의 통계량(estimator) 추정
mu_sample = np.mean(data_norm)
std_sample = np.std(data_norm, ddof=1)

## 상한, 하한 추정(95% 구간 z=1.96)
L_ = mu_sample - 1.96*(std_sample/np.sqrt(1000))
U_ = mu_sample + 1.96*(std_sample/np.sqrt(1000))

new_norm = norm(mu_sample,std_sample) ## 시각화를 위한 Sample 정보를 이용한 정규분포 생성

##시각화
plt.vlines(L_, 0, new_norm.pdf(L_), colors="g")
plt.vlines(U_, 0, new_norm.pdf(U_), colors="g")
plt.plot(x, new_norm.pdf(x),'r-', lw=2, alpha=0.6, label='norm pdf')
plt.title("sample normal distribution and confidence interval -sample: 1000")
plt.legend()
plt.show()
print(L_, mu_sample, U_)  ##결과 9.979423508565548 10.041601247611059 10.10377898
665657

## 신뢰구간이 좁아짐을 확인
```




```
9.869948211111891 9.932036558699012 9.994124906286133
```

가설검정

- 출처 (https://direction-f.tistory.com/26)
- 가설검정(Testing statistical hypotheses)은 모수에 대한 가설이 적합한지를 추출한 표본으로부터 판단하는 것을 나타내는 것입니다.
- 전국의 초등학교에서 6학년의 비만인 남자 학생(모집단)의 체중 평균이 70이고 표준편차가 7라고 가정해보겠습니다. 동시에 교육부에서 특정 다이어트 프로그램을 대대적으로 홍보했다고 가정해보겠습니다.
- 그럼 이러한 다이어트 프로그램 수행을 통해서 실제로 평균이 낮아졌는지를 판단하기 위해서 어떠한 과정을 거쳐야 할까요? 이러한 질문에 답을 하기 위해서 필요한 것이 가설검정입니다.
- 다이어트 프로그램 수행 후의 비만인 남자 학생들의 평균 몸무게를 μ라고 해보겠습니다. 우리는 전국의 비만인 남자 학생들을 다 조사할 수 없기 때문에 표본을 추출하여 표본의 평균을 구하는 것이 현실적인 방안입니다.(여기서는 임의로 49명을 추출한다고 하겠습니다.) 다만 단순히 49명의 표본평균이 모집단의 평균인 70보다 작다고 하여, 다이어트 프로그램 후의 모집단의 평균(μ)이 다이어트 프로그램 전의 모집단의 평균보다 낮아졌는지 확신하기 어렵습니다.
- 하지만 만약 다이어트 프로그램 전의 모집단 평균이 70이라고 할 때, 도저히 나오기 어려울 정도로 작은 값(c)이 다이어트 프로그램 후의 표본평균이라면 다이어트 프로그램이 모집단의 평균을 낮췄다고 판단할 수 있을 것입니다. 그렇다면 도저히 나오기 어려울 정도의 작은 값(c)의 기준은 어떻게 도출하게 될까요?
- 예를 들어 0.05의 확률을 거의 나오지 않을 정도의 확률이라고 하겠습니다. 그렇다면 아래와 같은 식을 만족하는 c값을 찾을 수 있다면, 우리는 다이어트 프로그램 후의 표본평균이 c보다 작다면 충분히 작은 값으로 판단하고 모집단의 평균을 낮춰다고 주장할 수 있을 것입니다.

$$P(\bar{X} \leq c) = 0.05$$

- 이에 답하기 위하여 우리는 표본평균의 분포를 활용해야 합니다. 모집단의 평균이 70, 표준편차가 7일 때 표본평균의 분포는 평균이 70이고 표준편차가 1(=7/√49)인 정규분포를 따르게 되고 이를 표준화하면 평균이 0이고 표준편차가 1인 표준정규분포를 따르게 됩니다.

$$Z = \frac{\bar{X} - 70}{7/\sqrt{49}} \sim N(0, 1)$$

- 표준정규분포표를 활용하면 $P(Z \leq 1.645) = 0.05$이고 우리는 이 1.645의 값을 활용하여, c값을 추정할 수 있습니다. 추정하는 식은 아래와 같이 됩니다.

$$0.05 = P(\frac{\bar{X} - 70}{7/\sqrt{49}} \leq -1.645) = P(\bar{X} \leq 70 - 1.645 \times 7/\sqrt{49})$$

- 즉 c는 68.355가됩니다. 그렇기 때문에 만약 표본평균이 68.335보다 작으면 다이어트 프로그램이 효과가 있다고 주장할 수 있을 것입니다.

가설검정 > 가설, 검정통계량

- 출처 (https://direction-f.tistory.com/28)
- 가설검정에는 두 개의 가설이 있습니다. 하나는 우리가 주장하고자 하는 가설이고, 다른 하나는 그 주장을 입증할 수 없을 때 주장을 무효화하면서 받아들여야 하는 가설입니다.

가설의 종류

- 이 때 우리가 주장하는 가설을 대립가설(Alternative Hypothesis, H_1)라고 하며, 주장을 입증할 수 없을 때 받아들여야 하는 가설을 귀무가설(Null Hypothesis, H_0)라고 합니다.
- 앞선 포스팅 (https://direction-f.tistory.com/26)에서 언급한 초등학생 대상 다이어트 프로그램이 효과에 대해 대립가설과 귀무가설을 세우면 아래와 같이 됩니다.

> 귀무가설(H_0) : 다이어트 프로그램은 평균 몸무에게 영향을 미치지 못했다.(μ = 70)
>
> 대립가설(H_1) : 다이어트 프로그램은 초등학생들의 평균 몸무게를 줄였을 것이다 (μ < 70)

오류의 종류

- 가설검정을 하게 되면 우리가 가질 수 있는 결과는 아래와 같이 두 가지로 나타날 수 있습니다.

> (1) 귀무가설을 기각하고 대립가설을 채택한다.
>
> (2) 대립가설을 기각하고 귀무가설을 채택한다.

- 이러한 결과를 판단하는데, 오류가 발생할 가능성은 항상 존재하게 됩니다. 실제로 귀무가설(H_0)이 맞는데 귀무가설(H_0)을 기각하거나 실제로 귀무가설(H_0)이 틀린데 귀무가설(H_0)을 기각하지 못 할 수 있습니다. 이 두 종류의 오류를 각각 제 1종 오류(Type 1 Error), 제 2종 오류(Type 2 Error)라고 정의하게됩니다.
- 다이어트 프로그램 예를 들면, 다이어트 프로그램이 실제로 효과가 없는데 효과가 있다고 판단한 경우가 제 1종 오류(Type 1 Error)이며, 다이어트 프로그램이 실제로 효과가 있는데 효과가 없다고 판단한 경우가 제 2종 오류(Type 2 Error)입니다.
- 다시 정리하면 아래와 같이 됩니다.

> (1) 제 1종 오류: 귀무가설이 맞을 때, 귀무가설을 기각하는 오류
>
> (2) 제 2종 오류: 귀무가설이 틀릴 때, 귀무가설을 채택하는 오류

가설검정 > 검정통계량과 기각역

- 출처 (https://direction-f.tistory.com/30)
- 우리는 앞선 포스팅 (https://direction-f.tistory.com/28)에서 두 가지의 가설을 정의하였습니다. 하나는 우리가 주장하고자 하는 가설(대립가설, H_1)이고, 다른 하나는 그 주장을 입증할 수 없을 때 주장을 무효화하면서 받아들여야 하는 가설(귀무가설, H_0)입니다.
- 그렇다면 우리는 어떤 가설을 받아들여야 하는지 어떻게 결정할 수 있을까요? 이 때 활용하는 것이 검정통계량(Test statistic)입니다.

검정통계량

- 다시 앞의 포스팅의 예를 들어보도록 하겠습니다. 우리는 아래와 같이 가설을 수립했습니다.

> 귀무가설(H_0) : 다이어트 프로그램은 평균 몸무에게 영향을 미치지 못했다.(μ = 70)
>
> 대립가설(H_1) : 다이어트 프로그램은 초등학생들의 평균 몸무게를 줄였을 것이다 (μ < 70)

- 대립가설을 채택하기 위해서는 다이어트 후 남자 초등학생의 평균 몸무게(표본평균)가 70에 비해 상당히 적어야 합니다. 즉, 어떤 적당한 값에 대해 그 값보다 적을 때 귀무가설을 기각하게 됩니다.
- 이렇게 표본평균이 취하는 구간중에 H_0를 기각하게 하는 구간을 기각역(critical region)이라고 합니다. 이를 아래와 같이 표현합니다. 중요한 것은 적절한 값 c를 정하는 것입니다.

$$R : \bar{X} \leq c$$

- 우리는 표본만을 관측하여 가설에 대해 검정하기 때문에 오류가 발생할 확률이 항상 존재합니다. 1종 오류를 범할 확률을 α, 2종 오류를 범할 확률을 β라고 하겠습니다. 두가지 종류의 오류를 함께 최소화하는 기각역을 구하면 가장 바람직하겠지만, α와 β는 반대로 움직입니다. 즉 α를 줄이면 β가 커지게 되고, α가 커지게 되면 β는 줄어들게 됩니다.
- 여기서 α와 β는 아래와 같이 표현됩니다.

$$\alpha = P(\bar{X} \leq c),\ \beta = P(\bar{X} \geq c)$$

- 보통의 경우 1종 오류가 2종 오류보다 심각한 것이기 때문에 1종 오류를 범할 확률을 작게 가져갑니다. 그래서 일반적으로 α값을 0.05나 0.1로 정하게 되고 이 때 정한 α값을 유의수준(Sigmificance level)이라고 합니다. 그리고 이 유의수준에 맞춰 c를 결정하게 됩니다.

Z 검정

- 표본평균은 아래와 같은 방법을 이용하여 표준화 시킬 수 있습니다.(만약 모집단의 표준편차를 모르면, 표본의 표준편차를 사용해도 무방합니다.)

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1)$$

- 이 때 유의수준 α에 대하여 기각역 R은 아래와 같습니다.

$$P(Z \leq -z_\alpha) = \alpha \rightarrow R : \bar{X} \leq \mu_0 - z_\alpha \cdot \frac{s}{\sqrt{n}}$$

- 일반적으로는, 표본으로부터 직접 Z를 구하여 기각역을 구하는 것이 흔합니다. 따라서 기각역을 다시 아래와 같이 표현합니다.

$$R : Z \leq -z_\alpha$$

- 위와 같은 검정를 Z-검정(Z-test)라고 합니다.

- 이제까지는 대립가설이 μ<x 경우에만 고려했습니다. 우리는 μ>x 인 경우, 그리고 μ≠x 인 경우 모두 고려할 수 있습니다. 검증형태는 아래와 같이 정리될 수 있습니다.
- 표본의 크기가 클 때 모평균 μ에 대한 가설 H_0: μ=x를 검정하기 위한 검정통계량은 아래와 같습니다.

$$Z = \frac{\bar{X} - \mu}{s/\sqrt{n}}$$

- 각 기각역은 아래와 같습니다.

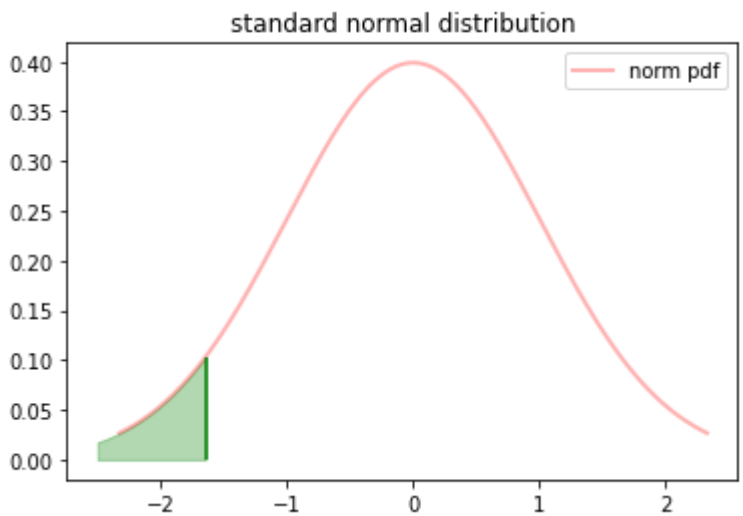
> 단측검정(μ<x) : $H_1 : \mu < x \rightarrow R : Z \leq -z_\alpha$
>
> 단측검정(μ>x) : $H_1 : \mu > x \rightarrow R : Z \geq -z_\alpha$
>
> 양측검정 : $H_1 : \mu \neq x \rightarrow R : |Z| \geq -z_\alpha$

- 파이썬을 활용해서 기각여부를 판단해보겠습니다.

In [23]:

```
from scipy.stats import norm
import matplotlib.pyplot as plt
import numpy as np

## 다이어트 전 비만 남자 초등학생의 몸무게 평균이 70, 표준편차가 3라고 가정
## 대립가설(H1) : 다이어트 프로그램은 초등학생들의 평균 몸무게를 줄였을 것이다(μ<70)
## 귀무가설(H0) : 다이어트 프로그램은 평균 몸무에게 영향을 미치지 못했다.(μ=70)

##  30명만 추출하여 다이어트 후의 효과를 검증하고자 함(임의로 Random sampling)
sample_data=np.random.randint(60,75,30)##low, high, size
sample_mean = np.mean(sample_data)
sample_std = np.std(sample_data, ddof=1)

## 검정을 위한 통계량 계산
Z_statistic = (sample_mean -70)/(sample_std/np.sqrt(30))

## -z_0.5 계산
mu2 = 0
std2 = 1
standard_norm = norm(mu2,std2)
z_05=-standard_norm.ppf(0.95) ## α=0.05일때

## 기각여부 판단
if Z_statistic <= z_05:
    print("H0 기각")
else :
    print("H0 채택")

 ## 기각역 시각화
x = np.linspace(standard_norm.ppf(0.01), standard_norm.ppf(0.99),1000)
z = np.linspace(-2.5,z_05,1000)
plt.plot(x, standard_norm.pdf(x),'r-', lw=2, alpha=0.3, label='norm pdf')
plt.vlines(z_05, 0, standard_norm.pdf(z_05), colors="g")
plt.fill_between(z, 0,standard_norm.pdf(z), color ="g", alpha=0.3)
plt.title("standard normal distribution")
plt.legend()
plt.show()
```

H0 기각



가설검정 > P-value(유의확률)

- 출처 (https://direction-f.tistory.com/31)
- 아마도 데이터를 분석할 때 검증 방법론을 조금이라도 활용해보신 분은 P-value란 개념을 많이 들어보셨을 것입니다. 우리가 어떤 통계적인 검증을 수행할 때, 우리가 수립했던 가설을 채택할지 하지 않을지 결정할 때 P-value를 많이 활용합니다.

P-value

- 먼저 P-value에 대해서 정의를 하고 설명을 하는 것이 좋을 것 같습니다.
- P-value(유의확률)이란 주어진 검정통계량 관측치로부터 귀무가설(H_0)을 기각하게 하는 최소의 유의수준을 말합니다.
- 정의만 보면 상당히 난해한 것 같습니다.
- 우리는 앞선 포스팅 (https://direction-f.tistory.com/30)에서 기각역을 정의하고 기각역안에 검정통계량이 포함되어야 귀무가설(H_0)를 기각함을 알았습니다.
- 예를 들어 [H_0 : μ = x/H_1 : μ < x]인 귀무가설과 대립가설이 있다고 가정하겠습니다. 그렇다면 이 때 유의수준 α에 대하여 기각역 R은 아래와 같습니다.

$$H_1 : \mu < x \rightarrow R : Z \leq -z_\alpha$$

- 위의 기각역에서 볼 수 있는 것처럼 검정통계량 Z가 -z_a보다 작아야 우리는 H_0를 기각하게됩니다. 만약 유의수준 α를 점점 키우면 어떻게 될까요? z_a값이 점점 작아짐에따라 -z_a는 커져 0에 가까워질 것입니다.(유의수준이 커짐에 따라 정규분포표에서 차지하는 영역이 커집니다.)
- 위에서 확인한 것과 같이 유의수준에 따라 우리는 기각여부가 달라짐을 알 수 있습니다. 그렇다면 표본으로부터 얻어진 검정 통계량(Z)를 가지고 귀무가설을 기각할 수 있게 하는 최소의 유의수준은 무엇일까요?
- 예를 통해 알아보겠습니다. 만약 검정통계량이 -1.96으로 구해졌다고 해보겠습니다. 만약 -z_a가 -2라면 -1.96은 기각역에 포함되지 않기때문에 H0을 기각할 수 없습니다. 즉, 적어도 -z_a가 -1.96보다 크거나 같아야 기각할 수 있게 됩니다. 자연스럽게 최소한의 기각역은 -1.96이 되고 P(Z≤-1.96)=0.025가 바로 유의확률, P-value가 됩니다.
- P-value가 크다는 것은 그만큼 기각을 위한 최소한의 유의수준 α가 크다는 것이고, 이 최소한의 유의수준 α를 검증의 기준으로 삼는다면 1종 오류를 범할 위험이 크다는 것을 뜻합니다.
- 따라서 우리는 통계적 검증을 할 때는 0.05, 0.1과 같은 기준을 두고 P-value가 이것보다 작으면 대립가설을 채택하게 됩니다.
- P-value를 구하는 식을 정리하면 다음과 같습니다.

$$R : Z \leq z \rightarrow P - value : P(Z \leq z)$$
$$R : Z \geq z \rightarrow P - value : P(Z \geq z)$$
$$R : |Z| \geq z \rightarrow P - value : P(|Z| \geq z)$$

두 모집단의 비교(표본이 클 때)

- 출처 (https://direction-f.tistory.com/35)
- 지금까지는 하나의 표본에 대해서 가설검정, 신뢰구간을 추정했습니다. 이번 포스팅에서는 두 표본집단에 대해서 비교를 하는 통계추론 방안에 대해서 정리해보도록 하겠습니다.
- 예를 들어 A지역 사람들의 평균소득과 B지역 사람들의 평균소득을 비교하는 문제와 같은 것입니다.
- 아래는 두 모집단으로부터 추출된 두 개의 표본과 그로부터 계산되는 통계량을 정리한 것입니다.(이전에 했던 하나의 표본으로 할때와 같습니다. 다만 그것을 두 번하는 것 뿐입니다.)
- 평균이 μ_1 이고 분산이 σ_1 인 모집단으로부터 추출된 표본(표본의 개수 n_1):

$$\bar{X} = \frac{1}{n_1}\sum X_i,\ s_1^2 = \frac{1}{n_1 - 1}\sum (X_i - \bar{X})^2$$

- 평균이 μ_2 이고 분산이 σ_2 인 모집단으로부터 추출된 표본(표본의 개수 n_2):

$$\bar{Y} = \frac{1}{n_2}\sum Y_i,\ s_2^2 = \frac{1}{n_2 - 1}\sum (Y_i - \bar{Y})^2$$

- 여기서 우리의 관심사는 $\mu_1 - \mu_2$ 에 대한 추론입니다.

두 모평균의 차이 추론(표본의 크기가 충분할 때)

- 두 모평균의 차 $(\mu_1 - \mu_2)$ 에 대한 추론을 위해서 두 표본평균의 차를 활용하는 것이 일반적입니다. 두 표본의 크기 n_1 , n_2 가 충분히 클 때 두 표본평균은 아래와 같이 정규분포로 근사되게 됩니다.

$$\bar{X} \sim N(\mu_1, \frac{\sigma_1^2}{n_1}),\ \bar{Y} \sim N(\mu_2, \frac{\sigma_2^2}{n_2})$$

- 그렇다면, 정규분포의 성질에 따라 표본평균의 차이는 다음과 같은 정규분포를 따르게 됩니다.

$$\bar{X} - \bar{Y} \sim N(\mu_1 - \mu_2, \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2})$$

- 표본평균의 차이도 정규분포를 따른다면, 정규화를 시켜줄 수 있습니다. 이 때 표본이 크기가 충분히 클때 모표준편차 σ는 표본표준편차 s로 대체하여 표현할 수 있습니다.

$$Z = \frac{(\bar{X} - \bar{Y}) - (\mu_1 - \mu_2)}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$

- 위와같이 정규화한뒤 표준정규분포를 활용하면, 앞서 단독 표본일 때 신뢰구간을 추정하는 방법과 동일하게 두 표본의 표본평균차이에 대한 신뢰구간을 추정할 수 있습니다.
- 최종적으로, $(\mu_1 - \mu_2)$ 에 대한 신뢰구간 및 검정은 아래와 같이 정리 할 수 있습니다
- (1) 표본의 크기 n_1 , n_2 가 모두 30보다 클 때 $100(1-\alpha)\%$ 신뢰구간:

$$((\bar{X} - \bar{Y}) - z_{\alpha/2} \times \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}},\ (\bar{X} - \bar{Y}) + z_{\alpha/2} \times \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}})$$

- (2) 표본의 크기 n_1 , n_2 가 모두 30보다 클 때 H0: $(\mu_1 - \mu_2) = \sigma_0$ 에 대한 검정:

$$Z = \frac{(\bar{X} - \bar{Y}) - \sigma_0}{\sqrt{s_1^2/n_1 + s_2^2/n_2}}$$

$$H_1 : \mu_1 - \mu_2 < \sigma_0 \rightarrow R : Z \leq z_\alpha$$

$$H_1 : \mu_1 - \mu_2 > \sigma_0 \rightarrow R : Z \geq z_\alpha$$

$$H_1 : \mu_1 - \mu_2 \neq \sigma_0 \rightarrow R : |Z| \geq z_{\sim}$$

두 모집단의 비교(표본이 작을 때, T-Test)

- 출처 (https://direction-f.tistory.com/36)
- 이전 포스팅 (https://direction-f.tistory.com/35)에서 표본이 충분히 클 때 두 집단의 차이를 보는 통계적 추론 방안을 다루었습니다. 이번 포스팅에서는 표본의 크기가 충분히 크지 못할 때(30개 미만일 때) 어떻게 통계적 추론을 할지에 대해서 정리해보도록 하겠습니다.
- 독립 표본의 경우에 우리는 표본의 크기가 충분히 크지 않을 때 표준정규분포 대신 t-distribution을 활용하였습니다. 두 집단간의 차이비교도 마찬가지로 우리는 t-분포를 활용합니다. 다만 표본의 수가 충분히 클 때와 다른 점은, 공통분산을 추정한다는 것입니다.

두 모평균의 차이 추론(표본이 작을 때)

- 표본이 크기가 작을 때, 두 집단의 차이에 대해서 통계적 추론을 하기 위해서는 아래와 같은 두 가지의 가정이 필요합니다.

> 가정(1): 두 모집단이 모두 정규분포를 따른다.
>
> 가정(2): 두 모집단의 표준편차가 일치한다.

- 공통분산은 가정(2)를 만족하기 위해 필요한 개념입니다. 즉, 공통분산을 구하여야만이 우리는 표본이 적을때도 t-분포를 적용하여 두 집단간의 차이를 검증할 수 있습니다. 두 모집단의 표준편차가 일치한다고 가정한다면 표본평균들의 차이는 다음과 같은 정규분포를 따르게 됩니다.

$$(\bar{X} - \bar{Y}) \sim N(\mu_1 - \mu_2, \sigma(\frac{1}{n_1} + \frac{1}{n_2}))$$

- 공통분산의 추정량은 각 표본의 편차제곱합을 더하여 두 표본의 자유도를 더한 값을 나누어주며, 아래와 같이 정의됩니다.

$$s_p^2 = \frac{\Sigma(X_i - \bar{X})^2 - \Sigma(Y_i - \bar{Y})^2}{n_1 + n_2 - 2} = \frac{(n_1 - 1)s_1^2 (n_2 - 1)s_2^2}{n_1 + n_2 + 2}$$

- 구한 공통분산을 활용하여 $(\mu_1 - \mu_2)$에 대한 검정을 위한 확률변수는 아래와 같이 정의되면 자유도가 $(n_1 + n_2 - 2)$인 t-분포를 따릅니다.

$$t = \frac{(\bar{X} - \bar{Y}) - (\mu_1 - \mu_2)}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t(n_1 + n_2 - 2)$$

- 최종적으로, 표본이 작을 때 $(\mu_1 - \mu_2)$에 대한 신뢰구간 및 검정은 아래와 같이 정리 할 수 있습니다.
- (1) $100(1 - \alpha)\%$ 신뢰구간:

$$(\bar{X} - \bar{Y}) \pm t_{\alpha/2}(n_1 + n_2 - 2) \times s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

- (2) H_0: $(\mu_1 - \mu_2) = \sigma_0$에 대한 검정:

$$t = \frac{(\bar{X} - \bar{Y}) - \sigma_0}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$H_1 : \mu_1 - \mu_2 < \sigma_0 \rightarrow R : t \leq t_\alpha(n_1 + n_2 - 2)$$
$$H_1 : \mu_1 - \mu_2 > \sigma_0 \rightarrow R : t \geq t_\alpha(n_1 + n_2 - 2)$$
$$H_1 : \mu_1 - \mu_2 \neq \sigma_0 \rightarrow R : |t| \geq t_\alpha(n_1 + n_2 - 2)$$

- H_0: $(\mu_1 - \mu_2) = \sigma_0$, 두 집단간의 차이가 있는지 검증하는 방안을 우리는 통상적으로 T-test라고 부릅니다. 즉 집단간의 차이가 유의한지 Test를 하는 것입니다.

- 지금까지 각 표본의 모집단의 표준편차가 동일하다고 가정할 때, 검정을 수행하는 것입니다. 만약 명확히 모집단의 표준편차가 다르면 어떻게 될까요? 이 때는 근사적으로 t-분포를 따른다고 보고 추론을 합니다. 이 때 확률변수 t는 아래와 같이 정의되며, $(n_1 - 1)$이나 $(n_2 - 1)$중에 적은 값을 자유도로 가집니다.

$$t = \frac{(\bar{X} - \bar{Y}) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \sim t(n_1 - 1) \, or \, t(n_2 - 1)$$

- 파이썬으로 실습을 진행해보겠습니다.

In [24]:

```
from scipy.stats import t
import matplotlib.pyplot as plt
import numpy as np

## 임의적으로 Random Sample 생성
sample_data1=np.random.randint(60,75,30)##low, high, size
sample_data2=np.random.randint(60,75,25)##low, high, size

#Sample1 정보
n1 = 30
mean_1 = sample_data1.mean()
var_1 = sample_data1.var(ddof=1)
std_1 = sample_data1.std(ddof=1)

#Sample2 정보
n2 = 25
mean_2 = sample_data2.mean()
var_2 = sample_data2.var(ddof=1)
std_2 = sample_data2.std(ddof=1)

#공통분산, 표준편차 계산
equal_var = ((n1-1)*var_1+(n2-1)*var_2)/(n1+n2-2)
equal_std = np.sqrt(equal_var)

## 95%신뢰구간
df = n1+n2-2
t_ = t(df)
t_025 = t_.ppf(0.975)
L_= round((mean_1-mean_2) - t_025*(equal_std*np.sqrt((1/n1)+(1/n2))),2)
U_= round((mean_1-mean_2) + t_025*(equal_std*np.sqrt((1/n1)+(1/n2))),2)
print("{} < {} < {}".format(L_, (mean_1-mean_2), U_)) ## 결과 0.39 < 2.7 < 5.01

## 가설검증 수행
## H1: (mean_1-mean_2) !=0
t_statistic = (mean_1-mean_2)/(equal_std*np.sqrt((1/n1)+(1/n2)))

if t_statistic <= t_025 or t_statistic >= t_025:
    print("H0 기각 (menq_1과 mean_2는 다르다)")
else :
    print("H0 채택 (menq_1과 mean_2는 다르지 않다)")

## 결과 H0 기각 (menq_1과 mean_2는 다르다)

## p-value(양측검정이기 때문에 2배)
p_val=2*min(t_.cdf(t_statistic),(1-t_.cdf(t_statistic)))

print(t_statistic) #2.3457492956179795
print(p_val) #0.022763016309548245

## scipy 패키지로 T-test 수행
from scipy import stats
print(stats.ttest_ind(sample_data1,sample_data2))
#결과 Ttest_indResult(statistic=2.34574929561798, pvalue=0.0227630163095481 2)

## 모집단의 표준편차가 같다는 가정을 적용하지 않을 때
print(stats.ttest_ind(sample_data1,sample_data2, equal_var=False))
# 결과 Ttest_indResult(statistic=2.3519948200767145, pvalue=0.02251635206666634)
```

```
-4.35 < -1.9533333333333331 < 0.44
H0 기각 (menq_1과 mean_2는 다르다)
-1.6355919851290301
0.1078508835026033
Ttest_indResult(statistic=-1.6355919851290301, pvalue=0.107850883502
6033)
Ttest_indResult(statistic=-1.6291472462493644, pvalue=0.109520562095
4587)
```

단순회귀분석

- 출처 (https://direction-f.tistory.com/37)
- 우리는 회귀분석을 활용하여, 아래와 같은 질문에 답을 할 수 있습니다.

> (1) 변수들은 서로 관련이 있는가?
>
> (2) 얼마나 밀접하게 관련이 있는가?
>
> (3) 관련이 있다면, 다른 변수를 가지고 관심있는 변수를 예측할 수 있는가?

- 위와 같이 회귀분석은 변수들의 관계를 규명하는데 활용되며, 회귀분석에 활용되는 변수는 독립변수(Independent variable)과 종속변수(dependent variable)가 있습니다. 독립변수는 설명변수(explanatory variable)이라고도 불리며, 연구자가 통제하는 변수입니다. 종속변수는 독립변수에 의해 결정/변화되는 변수로 주로 연구자가 관심을 가지는 변수입니다.

단순회귀분석

- 단순회귀분석은 종속변수가 1개이고, 독립변수도 1개인 가장 간단한 회귀분석입니다. 예를 들어 아버지의 키(독립변수)와 자식의 키(종속변수)간의 관계와 같은 것입니다.
- 단순회귀모형은 독립변수(x)와 종속변수(Y)간의 직선전인 관계를 나타내며, 아래와 같이 표현할 수 있습니다.

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

- 여기서 β_0과 β_1은 추정돼야 하는 미지의 회귀계수입니다. 오차 εi들은 서로 독립이며, 평균이 0, 표준편차가 σ인 정규분포를 따르는 확률변수입니다.
- 이 때 종속변수 Y_i는 아래와 같은 분포를 따르게 됩니다.

$$Y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

- 위 분포로부터 우리는 종속변수의 실제 관측값은 직선상의 특정한 값 $\beta_0 + \beta_1 x$가 정규분포를 따르는 오차 ϵ에 의하여 변동된 것으로 볼 수 있습니다.

최소제곱법(least squares method)

- 최소제곱법에 들어가기 앞서 편차를 정의하는 것이 필요합니다. 편차는 종속 변수의 실제 관측값과 예측값과의 차이입니다.

$$d_i = y_i - (\beta_0 + \beta_1 x_i)$$

- 최소제곱법은 위와 같은 편차들의 제곱합을 최소로하는 β_0과 β_1를 추정하는 방법입니다. 편차제곱합은 아래와 같이 정의 됩니다.

$$D = \sum d_i^2 = \sum (y_i - (\beta_0 + \beta_1 x_i))^2$$

- 이 때 최소제곱추정량(least squares estimator)는 아래와 같이 표기하겠습니다.

$$\hat{\beta}_0, \hat{\beta}_1$$

- 그 다음으로, 최소제곱추정량을 정의하기에 앞서, 계산의 편의성을 위해서 몇가지 기호를 정의하겠습니다.

$$\bar{x} = \frac{1}{n}\sum x_i, \ \bar{y} = \frac{1}{n}\sum y_i$$

$$S_{xx} = \sum (x_i - \bar{x})^2 = \sum x_i^2 - n\bar{x}^2 = \sum x_i^2 - \frac{(\sum x_i)^2}{n}$$

$$S_{yy} = \sum (y_i - \bar{y})^2 = \sum y_i^2 - n\bar{y}^2 = \sum y_i^2 - \frac{(\sum y_i)^2}{n}$$

$$S_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}) = \sum x_i y_i - n\bar{x}\bar{y} = \sum x_i y_i - \frac{(\sum x_i)(\sum y_i)}{n}$$

- 최소제곱추정량은 편차제곱합 D를 β_0과 β_1로 편미분하여 얻을 수 있습니다.참고 (https://are.berkeley.edu/courses/EEP118/current/derive_ols.pdf)

- 최소제곱추정량 및 추정회귀직선은 아래와 같습니다.

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$