ADP 실기대비 교재 - Python 편

통계이론반

슬기로운 통계생활

2023 | Version 1.0

차례

그림 차례

환영합니다!

- 본 교재는 ADP 실기 시험의 통계 문제를 정확하게 이해하고, 만점에 가까운 점수를 받아내기 위해서 제작된 **슬기로운 통계생활** 만의 노하우가 담긴 교재입니다. 통계는 하루 아침에 실력이 늘어나지 않는 분야이며, 동시에 거짓말을 하지 않는 분야이기도 합니다. 시험 합격을 위하여 같이 재미있게 공부해보시죠!

https://www.youtube.com/c/statisticsplaybook

제 1 장

통계적 검정의 근본 원리

ADP 시험을 완벽하게 준비하기 위해서 필요한 통계 지식들을 정리한 챕터입니다. 처음 공부하시는 분들은 힘들 수 있지만, 포기하지 마시고 이해하려고 노력해주세요!

1.1 표본추출 개념 이해하기

모집단과 표본집단

- 모집단: 특정한 연구, 조사에서 특성을 알아내고 싶은 전체 집단
  - 대통령 선거 후보 호불호 조사의 모집단: 특정 시점의 대한민국 선거권을 가진 모든 대한민국 국민
- 표본집단: (시간과 비용의 문제로) 모집단에서 선택된 부분 집단
  - 전화 설문 조사 3천명

통계적 추론은 표본집단에서부터 얻어낸 정보로 계산한 정량적인 수치들을 모집단 역시 동일하게 가질 것이라 추론 (간주) 함

- 좋은 표본이 가져야할 가장 중요한 특성은 모집단을 잘 반영해야한다는 점!

표본 추출의 대표적인 방법들

모집단의 특성을 반영하는 좋은 표본 집단을 만들기 위하여 여러가지 표본 추출법이 제안되어 왔음.

- **단순랜덤추출**: N개 중 n개를 임의로 선택하여 추출
- 계통추출법: 표본을 동일 집단으로 나눈 뒤 번호 부여, 일정 거리를 두고 추출
- 집락추출법: 클러스터링 후, 임의 집단을 선택.
  - 1 stage: 선택된 임의 집단 구성원 전부 추출
  - 2 stage: 선택 된 임의 집단에서 표본 임의 선택.
- **층화추출법**: 모집단의 특정 성질을 그대로 반영한 표본을 뽑는 방법. 계층을 형성 후 각 계층 별 표본을 추출한다.

그림로 보는 추출방법

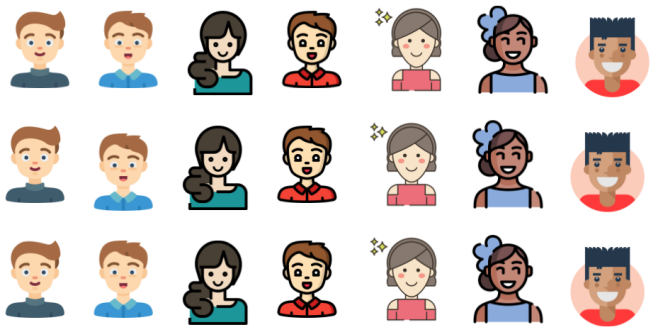


그림 1.1: 표본추출 방법 설명을 위한 그림

복원 추출 vs. 비복원 추출 개념 이해

현실에서는 거의 모든 표본 추출법은 **비복원 추출**이지만, 검정 이론에서는 복원 추출 가정이 대부분.

- 복원 추출: 한 번 뽑힌 표본도 다시 뽑힐 수 있도록 허용함.
  - 설문조사에서 한번 답변한 사람이 다시 답변하는 꼴.
- 비복원 추출: 한 번 뽑힌 표본은 다시 뽑지 않는다.

1.2 통계적 용어 짚고 넘어가기

ADP는 서술형 답안지를 제출하는 시험입니다. 여러분의 단어 선택, 문장 구성이 여러분의 통계 수준을 보여주는 가장 직접적인 근거가 됩니다. 그만큼 한 글자, 한 단어가 중요합니다.

랜덤 실험 (random experiment)

어떤 실험에서 나올 수 있는 결과의 경우의 수가 여러 개이고, 실험을 진행하기 전에 어떠한 결과가 나올지 예측 불가능하며, 각 결과가 나올 가능성을 수치화 할 수 있는 실험을 랜덤 실험이라고 이야기 합니다. 랜덤 실험의 예는 다음과 같습니다.

- 동전을 1번 던지는 행위
- 주사위를 2개 던지는 행위
- 특정 선수의 자유투를 던지는 행위
- 지나가는 유권자에게 특정 후보에 대한 호불호를 물어보는 설문

표본 공간 (sample space)

랜덤 실험으로부터 나올 수 있는 결과값 들의 **집합**이며 보통 S를 사용하여 나타냅니다. 표본 공간의 예는 다음과 같습니다.

- $\{Head, Tail\}$
- $\{(1,1), ..., (6,6)\}$

사건 (event)

사건이란 **표본공간**의 **부분집합**들을 의미합니다.

예제

앞에서 배운 동전을 한번 던지는 랜덤 실험의 사건들을 나열해 보세요.

- E_1:
- E_2:
- E_3:
- E_4:

"사건 E가 일어났다."의 뜻

어떤 실험을 했을 때, 그 결과값이 사건 E의 원소로 포함되어 있다면, 우리는 사건 E가 일어났다고 말합니다.

확률 함수 (probability function)

어떤 **사건**이 일어나는 가능성을 측정하는 함수를 **확률함수**라고 부르고, P라는 기호를 사용하여 표기합니다. 통계학에서는 사건의 확률의 계산하는 방법에 있어서 다음의 3가지를 약속하고 있습니다.

확률의 공리 (axiom of probability)

1. 모든 사건 E에 대하여 $P(E) \geq 0$. 즉, 모든 사건의 확률은 0보다 크거나 같다. (확률은 음수가 될 수 없습니다.)
2. 표본 공간 S에 대하여 $P(S) = 1$. 즉, 표본 공간에 대한 확률은 1이다.
3. 사건 $E_1, E_2, ...$를 생각하자. 만약 i와 j가 다를 때, $E_i \cap E_j = \phi$를 만족한다면, 각 사건들의 합 사건에 대한 확률은 각 사건의 확률을 더하여 계산한다.

$$P(E_1 \cup E_2 \cup ...) = P(E_1) + P(E_2) + ...$$

확률은 함수라는 사실

많이 지나치는 점이지만, 확률은 함수라는 사실을 기억합니다. 함수는 입력값과 출력값이 존재하죠.

- 확률함수의 입력값: 사건
- 확률함수의 출력값: 0과 1사이의 실수

확률변수 (random variable)

확률변수는 **표본공간**의 원소들(실험의 결과값)에 정의된 실수값을 갖는 함수입니다.

- 입력값: 표본 공간의 원소들
- 출력값: 실수 (많은 경우)

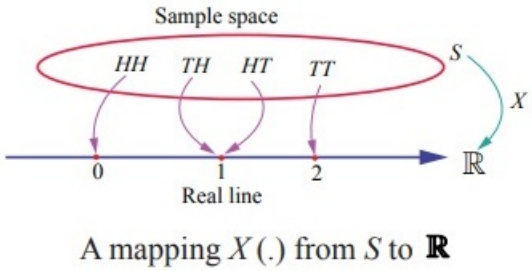


그림 1.2: 확률변수 개념 시각화

가장 기초적인 예

주사위를 두 번 던지는 실험에서 확률변수 X를 주사위를 2개 던져 나온 숫자의 합으로 정의하자.

- 질문 1. 표본공간은?
- 질문 2. 확률변수 X가 갖는 값은?
- 질문 3. 확률변수는 왜 쓸까?

1.3 확률 계산 과정의 이해

기본적인 확률 계산공식들

- $P(E^c) = 1 - P(E)$
- $P(\phi) = 0$

일어날 확률이 똑같은 사건들의 확률 계산

어떠한 실험의 결과값이 동일한 확률을 갖는다는 것을 미리 알고 있는 경우가 있습니다. 예를 들어 정육면체의 주사위를 던져 각 숫자가 나오는 결과값이 그런 경우이죠. 만약 표본공간 S의 원소들이 유한한 n개의 결과값으로 이루어져 있고,

$$S = \{s_1, s_2, ..., s_n\}$$

그 각각의 결과값이 일어날 확률이 같다는 것을 알고 있을 경우,

$$P(\{s_1\}) = P(\{s_2\}) = ... = P(\{s_n\})$$

각각의 확률을 다음과 같이 부여할 수 있다.

$$P(\{s_i\}) = \frac{1}{n}$$

예 1: 주사위 던지기

각 숫자가 나올 확률이 똑같은 완벽한 주사위가 있다고 가정. 주사위를 한 번 던지는 실험의 표본 공간은 다음과 같다.

$$S = \{1, 2, 3, 4, 5, 6\}$$

따라서, 각 숫자가 나올 확률은 다음과 같이 정할 수 있게 됩니다.

$$P(\{1\}) = ... = P(\{6\}) = \frac{1}{6}$$

예 2: 주사위 2개 던지기

주사위 2개 던지는 실험의 표본공간은 다음과 같다.

- 주사위를 두 번 던져서 얻을 수 있는 숫자들의 순서쌍 총 36개

$$S = \left\{ \begin{array}{llllll} (1,1), & (1,2), & (1,3), & (1,4), & (1,5), & (1,6) \\ (2,1), & (2,2), & (2,3), & (2,4), & (2,5), & (2,6) \\ (3,1), & (3,2), & (3,3), & (3,4), & (3,5), & (3,6) \\ (4,1), & (4,2), & (4,3), & (4,4), & (4,5), & (4,6) \\ (5,1), & (5,2), & (5,3), & (5,4), & (5,5), & (5,6) \\ (6,1), & (6,2), & (6,3), & (6,4), & (6,5), & (6,6) \end{array} \right\}$$

따라서, 표본 공간의 원소들에 대한 확률값은 다음과 같이 계산됩니다.

$$P(\{(1,1)\}) = ... = P(\{(6,6)\}) = \frac{1}{36}$$

확률의 덧셈법칙

일반적인 두 사건에 대하여 합사건의 확률은 다음과 같은 덧셈 법칙을 따릅니다.

- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

따라서, 교집합이 없는 두 사건에 대한 합사건의 확률은 각각 사건의 확률을 더하면 됩니다.

- $P(A \cup B) = P(A) + P(B) - P(A \cap B) = P(A) + P(B) - P(\phi) = P(A) + P(B)$

조건부 확률

사건 A가 일어났다는 전제하에 사건 B가 일어날 확률을 나타냅니다. (단, $P(A) > 0$ 일 때, 정의)

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

- 수학적 의미: 사건 B의 확률을 계산할 때 사건 A가 새로운 표본공간이 됨을 의미합니다.

예제: 조건부 확률과 교차표

다음은 한 학교에서 학생들의 성별과 그들이 선택한 과목에 대한 교차표입니다.

	수학	과학	영어	합계
남성	20	30	50	100
여성	30	40	30	100

	수학	과학	영어	합계
합계	50	70	80	200

1. 학생이 수학을 선택했을 때, 그 학생이 남성일 확률은 얼마인가요?
2. 학생이 여성일 때, 그 학생이 과학을 선택할 확률은 얼마인가요?

해결 방법

1. 학생이 수학을 선택했을 때, 그 학생이 남성일 확률은 다음과 같이 계산할 수 있습니다:

$$P(\text{남성}|\text{수학}) = \frac{P(\text{남성} \cap \text{수학})}{P(\text{수학})} = \frac{20}{50} = 0.4$$

2. 학생이 여성일 때, 그 학생이 과학을 선택 할 확률은 다음과 같이 계산할 수 있습니다:

$$P(\text{과학}|\text{여성}) = \frac{P(\text{과학} \cap \text{여성})}{P(\text{여성})} = \frac{40}{100} = 0.4$$

확률의 곱셈법칙

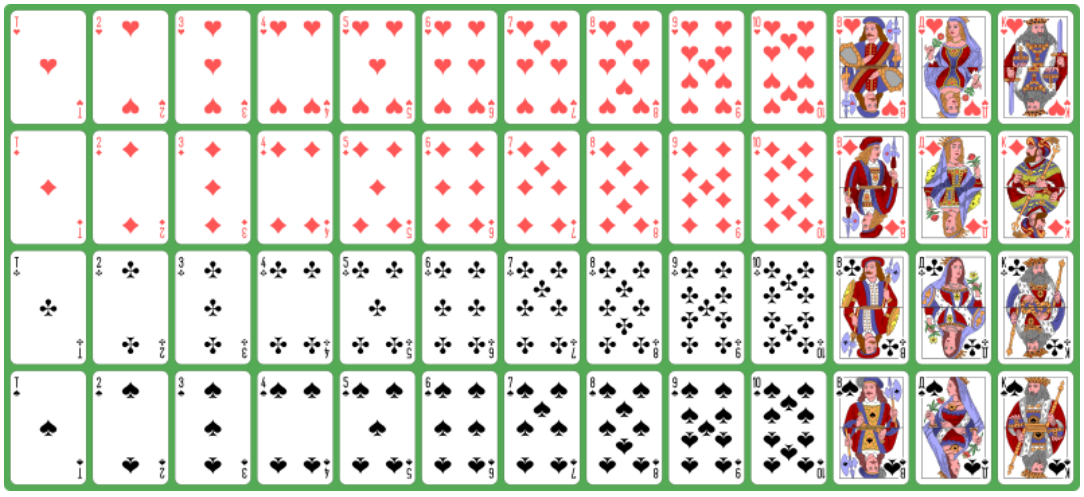
사건 A와 B가 동시에 일어날 확률은 다음과 같이 계산합니다.

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

여기서 $P(B|A)$는 조건부 확률을 의미합니다.

예제: 카드 뽑기

다음은 카드 한 세트(덱, Deck)를 나타낸 그림입니다. (조커는 제외)



- 2장의 카드를 덱에서 뽑았을때, 두 장 모두 빨간색 카드가 나올 확률을 구하세요. (단, 한번 뽑힌 카드는 다시 덱에 집어넣지 않음)

독립사건의 확률

두 사건 A와 B가 독립이면 다음이 성립합니다.

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

- 2장의 카드를 덱에서 뽑았을때, 두 장 모두 빨간색 카드가 나올 확률을 구하세요. (단, 한번 뽑힌 카드는 다시 덱에 집어넣음)

전확률정리 (law of total probability)

사건 $A_1, A_2, ...$이 표본공간 S를 **분할** 할 때, 사건 B가 일어날 확률은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} P(B) &= P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + ... \\ &= \sum_i P\left(B|A_i\right) P\left(A_i\right) \end{aligned}$$

여기서 분할의 의미는 다음과 같다.

1. $i \neq j$이면, $A_i \cap A_j = \phi$
2. $A_1 \cup A_2 \cup ... = S$

예제: 접시를 깰 확률

슬통 식당에는 수니, 젤리, 뭉이 종업원 3명이 번갈아 일한다. 수니는 한달 중 50%, 젤리는 30%, 뭉이는 나머지 날들을 일하고 있다. 가장 오랫동안 일 한 수니는 1%의 확률로 접시를 깨고, 젤리는 2%, 미숙한 뭉이는 3% 확률로 접시를 깬다. 슬통이가 가게를 방문한 어느 날, 접시가 깨질 확률을 구하시오.

- 정답: 0.017

슬통의 방문날 각 종업원이 일 할 확률은 $P(Soony) = 0.5, P(Jelly) = 0.3, P(Moong) = 0.2$과 같다. 접시가 깨지는 사건을 Break 라고 정의하면, 각자의 일하고 있는 날 사건 Break가 발생할 확률을 다음과 같이 나타낼 수 있다.

- $P(Break|Soony) = 0.01$
- $P(Break|Jelly) = 0.02$
- $P(Break|Moong) = 0.03$

따라서 전확률 정리에 의하여 슬통이가 방문한 날 접시가 깨질 확률은 다음과 같다.

$$\begin{aligned} P(Break) &= P(Soony \cap Break) + P(Jelly \cap Break) + P(Moong \cap Break) \\ &= P(Soony)P(Break|Soony) + P(Jelly)P(Break|Jelly) + P(Moong)P(Break|Moong) \end{aligned}$$

베이즈 정리

- 전확률 정리와 확률의 곱셈법칙의 응용

사건 $A_1, A_2, ...$ 들이 표본공간 S를 분할하고 있고, 모든 A_i's들과 사건 B의 확률값이 0보다 클 때, 다음이 성립합니다.

$$P(A_i \mid B) = \frac{P(B \mid A_i)\,P(A_i)}{P(B)} = \frac{P(B \mid A_i)\,P(A_i)}{\sum_i P(B|A_i)\,P(A_i)}$$

예제: 접시를 깰 확률 2

이전 문제에서 슬통이가 방문한 날 마침 주방에서 접시가 깨지는 소리가 들렸다. 그날 일하고 있는 종업원이 젤리일 확률을 구하세요.

- 정답: 35.3%

1.4 확률변수의 편리성

앞에서 살펴본 주사위를 2개 던지는 실험에서 나온 눈의 합을 확률 변수 X로 정의할 때, 다음의 집합이 나타내는 사건은 무엇이며, 의미는 무엇일까요?

$$\{X \leq 3\}$$

확률변수를 사용한 확률 계산

$\{X \leq 3\}$에 대응하는 실제 다음의 사건은 다음과 같습니다.

$$\{(1,1),(1,2),(2,1)\}$$

따라서 $P(X \leq 3)$는 앞에서 배운 확률의 공리에 의하여 다음과 같이 계산할 수 있습니다.

$$\begin{aligned} P(\{X \leq 3\}) &= P(\{(1,1),(1,2),(2,1)\}) \\ &= P(\{(1,1)\}) + P(\{(1,2)\}) + P(\{(2,1)\}) \\ &= \frac{1}{36} + \frac{1}{36} + \frac{1}{36} = \frac{1}{12} \end{aligned}$$

핵심 포인트는 확률변수를 사용해 표현된 조건을 만족하는 사건들을 역으로 찾아서 그것들의 확률값을 계산하는 메커니즘이라는 것입니다.

X의 확률분포표 작성하기

확률변수 X가 갖는 값에 대응하는 확률을 계산 후, 표를 채워 보세요.

x	2	3	4	5	6	7	8	9	10	11	12
Prob.											

확률질량함수를 사용해서 계산

위에서 계산한 $P(X \leq 3)$ 확률을 확률분포표를 사용하여 다시 계산하면 다음과 같습니다.

$$P(X \leq 3) = P(X = 2) + P(X = 3) = \frac{1}{36} + \frac{2}{36} = \frac{1}{12}$$

위에서 작성한 확률분포표를 입력값을 확률변수가 가질 수 있는 값으로, 출력값을 확률값으로 하는 함수로 볼 수 있는데, 이를 **확률질량함수**라고 부릅니다. 이산형 확률변수에는 그것에 대응하는 확률질량함수가 존재합니다.

확률변수 X의 확률분포 (probability distribution of X)

확률변수 X의 확률분포라는 용어는 확률변수 X의 값에 대응하는 확률이 어떻게 퍼져있는지는 나타내는 용어입니다.

확률분포

확률변수 X가 가질 수 있는 값 x에 대응하는 확률을 시각화 한 것이라 생각합시다.

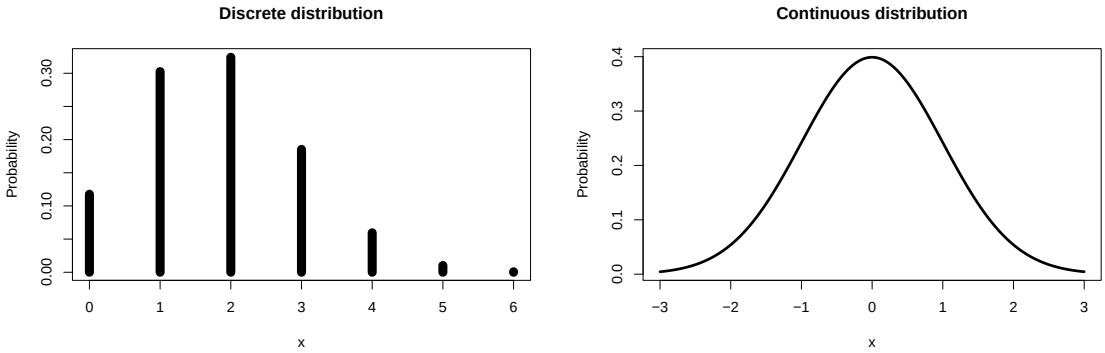


그림 1.3: 확률질량함수(왼쪽)와 확률밀도함수(오른쪽)

확률변수가 갖는 값의 성질(이산형, 연속형)에 따라 확률을 나타내는 방법이 달라집니다.

- (이산형) 확률질량함수: x에 대응하는 막대 높이가 확률을 나타냅니다.
- (연속형) 확률밀도함수: x축에 대응하는 범위에 해당하는 함수 아래의 넓이가 확률을 나타냅니다.

"확률변수가 확률분포 f를 따른다."는 의미

어떤 확률변수 X의 확률분포를 확률질량함수 혹은 확률밀도함수 f를 사용하여 나타낼 수 있을 경우,

1. f를 f_X로 나타내고,
2. 확률변수 X가 확률분포 f를 따른다고 말하며, 기호로 다음과 같이 나타냅니다.

$$X \sim f_X$$

X의 확률분포를 f_X를 사용하여 다음과 같이 계산 가능합니다.

$$P(a \leq X \leq b) = \sum_{a \leq x \leq b} f_X(x)$$

혹은

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

으로 나타낼 수 있습니다.

확률분포의 중심을 잡아내는 기대값

확률변수의 평균, 또는 기대값은 확률변수가 가질 수 있는 각각의 값에 그 값이 나타날 확률을 곱한 값들의 합입니다. 이는 **확률변수의 '평균적인' 값**이라고 볼 수 있고, 혹은 **확률 분포의 무게중심**을 잡아내는 지표라고 볼 수 있습니다.

이산 확률변수의 평균

이산 확률변수 X의 기대값은 다음과 같이 정의됩니다.

$$E(X) = \sum_{i=1}^{n} x_i P(x_i)$$

여기서 x_i는 확률변수 X가 가질 수 있는 각각의 값이고, $P(x_i)$는 그 값이 나타날 확률입니다.

연속 확률변수의 평균

연속 확률변수 X의 기대값은 다음과 같이 정의됩니다:

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx$$

여기서 $f(x)$는 확률밀도 함수입니다.

기대값의 성질

기대값은 다음과 같은 성질을 가집니다.
1. 상수 c에 대해, $E(c) = c$
2. 상수 a와 b, 확률변수 X에 대하여,

$$E(aX + b) = aE(X) + b$$

3. 확률변수 X와 Y에 대하여,

$$E(X + Y) = E(X) + E(Y)$$

예제: 이산 확률변수의 기대값 구하기

확률변수 X가 다음과 같은 확률질량함수를 가진다고 가정해봅시다.

x_i	$P(x_i)$
1	0.1
2	0.3
3	0.2
4	0.4

이 경우 확률변수 X의 기대값 $E(X)$는 다음과 같이 계산됩니다.

$$E(X) = \sum_{i=1}^{n} x_i P(x_i) = 1 \cdot 0.1 + 2 \cdot 0.3 + 3 \cdot 0.2 + 4 \cdot 0.4 = 3.0$$

따라서 주어진 확률변수 X의 기대값은 2.9입니다.

```
import numpy as np

values = np.array([1, 2, 3, 4])
probabilities = np.array([0.1, 0.3, 0.2, 0.4])
np.sum(values * probabilities).round(3)
```

```
## 2.9
```

예제: 연속 확률변수의 기대값 구하기

확률변수 X가 2에서 4까지의 균일분포를 가진다고 가정해봅시다. 2에서 4까지의 균일분포의 확률밀도함수는 다음과 같습니다.

$$f(x) = \begin{cases} \frac{1}{4-2} = \frac{1}{2} & \text{for } 2 \leq x \leq 4, \\ 0 & \text{otherwise} \end{cases}$$

이 경우 확률변수 X의 기대값 $E(X)$는 다음과 같이 계산됩니다.

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx = \int_{2}^{4} x \cdot \frac{1}{2} dx = \frac{1}{2} \cdot \frac{x^2}{2} \Big|_{2}^{4} = 3$$

따라서, 확률변수의 기대값은 3입니다. 이것은 균일분포의 확률밀도함수를 중심을 맞추기 위해서 손가락을 3에 올리면 무게가 딱 맞는 것과 일치합니다.

확률분포의 퍼짐을 잡아내는 분산

확률변수의 분산은 확률변수의 값이 그 평균값으로부터 얼마나 퍼져 있는지를 측정하는 값입니다. 분산이 크면 값들이 평균으로부터 많이 퍼져 있고, 분산이 작으면 값들이 평균에 가깝게 모여 있습니다.

이산 확률변수의 분산

이산 확률변수 X의 분산은 다음과 같이 정의됩니다.

$$Var(X) = E[(X - E(X))^2] = \sum_{i=1}^{n}(x_i - E(X))^2 P(x_i)$$

여기서 x_i는 확률변수 X가 가질 수 있는 각각의 값이고, $P(x_i)$는 그 값이 나타날 확률입니다.

연속 확률변수의 분산

연속 확률변수 X의 분산은 다음과 같이 정의됩니다:

$$Var(X) = E[(X - E(X))^2] = \int_{-\infty}^{\infty}(x - E(X))^2 f(x)dx$$

여기서 $f(x)$는 확률밀도 함수입니다.

분산의 성질

분산은 다음과 같은 성질을 가집니다.
1. 상수 c에 대해, $Var(c) = 0$
2. 상수 a와 b, 확률변수 X에 대하여,

$$Var(aX + b) = a^2 Var(X)$$

3. 독립인 확률변수 X와 Y에 대하여,

$$Var(X + Y) = Var(X) + Var(Y)$$

분산을 좀 더 편하게 계산하는 법

확률변수의 분산은 다음과 같이 계산할 수도 있습니다.

$$Var(X) = E[X^2] - (E[X])^2$$

이 식은 분산을 **평균의 제곱과 제곱의 평균의 차이**로 표현한 것입니다. 이 식을 사용하면 분산을 계산하는 데 필요한 계산량을 줄일 수 있습니다.

예제: 이산 확률변수의 분산 구하기

앞서 사용한 이산 확률변수 X의 확률질량함수를 다시 사용하겠습니다.

x_i	$P(x_i)$
1	0.1
2	0.3

x_i	$P(x_i)$
3	0.2
4	0.4

이 경우 확률변수 X의 분산 $Var(X)$는 다음과 같이 계산됩니다.

$$\begin{aligned} Var(X) &= \sum_{i=1}^{n}(x_i - E(X))^2 P(x_i) \\ &= (1-2.9)^2 \cdot 0.1 + (2-2.9)^2 \cdot 0.3 + (3-2.9)^2 \cdot 0.2 + (4-2.9)^2 \cdot 0.4 = 1.09 \end{aligned}$$

따라서 이 확률변수의 분산은 1.09입니다.

```
values = np.array([1, 2, 3, 4])
probabilities = np.array([0.1, 0.3, 0.2, 0.4])
np.sum((values - 2.9)**2 * probabilities).round(3)
```

```
## 1.09
```

예제: 연속 확률변수의 분산 구하기

앞서 사용한 2에서 4까지의 균일분포를 가진 연속 확률변수 X를 다시 사용하겠습니다. 이 경우 확률변수 X의 분산 $Var(X)$는 다음과 같이 계산됩니다.

$$Var(X) = \int_{-\infty}^{\infty}(x - E(X))^2 f(x)dx = \int_{2}^{4}(x-3)^2 \cdot \frac{1}{2}dx = \frac{1}{2} \cdot \frac{(x-3)^3}{3}\Big|_2^4 = \frac{1}{3}$$

따라서 이 확률변수의 분산은 $1/3$입니다.

확률누적분포 (Cumulative Distribution Function, CDF)

확률누적분포(Cumulative Distribution Function, CDF)는 확률변수가 특정 값보다 작거나 같을 확률을 나타내는 함수입니다. 확률밀도함수(PDF)를 특정 범위에서 적분하여 얻을 수 있습니다.

확률변수 X의 누적분포함수 $F(x)$는 다음과 같이 정의됩니다:

$$F(x) = P(X \leq x)$$

이는 확률변수 X의 값이 x보다 작거나 같을 확률을 나타냅니다.

연속형 확률변수의 CDF

연속형 확률변수의 경우, CDF는 확률밀도함수를 $-\infty$에서 x까지 적분한 값으로 정의됩니다.

$$F(x) = \int_{-\infty}^{x} f(t)\,dt$$

여기서 $f(t)$는 확률밀도함수입니다.

이산형 확률변수의 CDF

이산형 확률변수의 경우, CDF는 각 가능한 결과에 대한 확률을 더한 값으로 정의됩니다:

$$F(x) = \sum_{t \le x} P(X = t)$$

활용

CDF는 확률변수가 특정 범위에 속할 확률을 계산하는 데 사용됩니다. 예를 들어, 확률변수 X가 a보다 크고 b보다 작을 확률은 다음과 같이 계산할 수 있습니다:

$$P(a < X < b) = F(b) - F(a)$$

이는 $X = b$에서의 누적확률에서 $X = a$에서의 누적확률을 빼는 것으로, 이는 a와 b 사이의 확률밀도함수 아래의 영역의 넓이, 즉, 확률을 계산할 수 있습니다.

1.5 여러가지 확률분포

통계학에서는 미리 정해진 여러 유명한 확률분포가 존재합니다. 이러한 그 중 가장 기본이 되는 확률분포들에 대하여 알아봅시다.

파이썬에서 분포 관련 함수

Python의 SciPy 라이브러리는 다양한 확률분포에 대한 확률질량함수(`pmf`), 확률밀도함수 (`pdf`), 누적분포함수(`cdf`), 퀀타일 함수(Percent Point Function, `ppf`), 그리고 랜덤 샘플 생성 함수 (`rvs`)를 제공합니다.

SciPy 함수 이름 정리

파이썬의 SciPy 라이브러리의 분포관련 함수들의 이름은 다음의 규칙을 따릅니다.
* 확률질량함수 (pmf) / 확률밀도함수 (pdf)
* 누적분포함수 (cdf)
* 퀀타일 함수 (ppf)
* 랜덤샘플함수 (rvs)

확률변수의 모수 (parameter)

어떤 확률분포는 그 모양이 특정한 값에 따라 결정됩니다. 이러한 수를 **확률변수의 모수**라고 합니다. 분포의 종류가 정해지고, 대응하는 모수를 알면 확률분포를 그릴 수 있습니다. 앞으로 자주 사용 할 몇 가지 대표적인 분포에 대하여 공부합시다.

베르누이 분포 (p)

0과 1이 나오는 베르누이 확률변수의 결과값에 대응하는 확률분포를 그리기 위해서는 1이 나오는 확률 p만 알면 됩니다. 베르누이 분포의 확률질량함수는 다음과 같이 표현됩니다.

$$P(X = k) = p^k(1-p)^{1-k}$$

여기서 X는 베르누이 확률변수를 나타내며, k는 가능한 결과인 0 또는 1을 나타냅니다. p는 1이 나올 확률로, $0 \leq p \leq 1$의 값을 가집니다. 수식 내의 p^k는 1이 나오는 경우의 확률을, $(1-p)^{1-k}$는 0이 나오는 경우의 확률을 나타냅니다. 파이썬의 경우 SciPy 라이브러리의 `stats` 모듈에서 `bernoulli` 클래스를 사용하여 베르누이 분포에 대한 함수들을 사용할 수 있습니다.

```
from scipy.stats import bernoulli
bernoulli.pmf(k, p)
bernoulli.cdf(k, p)
bernoulli.ppf(q, p)
bernoulli.rvs(p, size, random_state)
```

이항 분포 (n, p)

이항 분포는 독립적인 베르누이 시행을 n번 반복하여 성공하는 횟수에 대한 분포입니다. 이때, 각 베르누이 시행의 성공 확률은 p로 주어집니다. 이항 분포의 확률질량함수는 다음과 같이 표현됩니다.

$$P(X = k) = \binom{n}{k} p^k(1-p)^{n-k}$$

여기서 k는 이항 확률변수를 나타내며, k는 성공 횟수를 나타냅니다. n은 시행 횟수를 나타내며, p는 각 시행에서 성공할 확률입니다. $\binom{n}{k}$는 이항 계수를 나타내며, 다음과 같이 계산됩니다:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

이항 계수는 n개의 원소 중에서 k개의 원소를 선택하는 경우의 수를 나타냅니다. 파이썬의 경우 SciPy 라이브러리의 `stats` 모듈에서 `binom` 클래스를 사용하여 베르누이 분포에 대한 함수들을 사용할 수 있습니다.

```
from scipy.stats import binom
binom.pmf(k, n, p)
binom.cdf(k, n, p)
```

```
binom.ppf(q, n, p)
binom.rvs(n, p, size=1, random_state=None)
```

포아송 분포 (λ)

포아송 분포는 이산 확률 분포 중 하나로, 일정한 시간 또는 공간에서 발생하는 이벤트의 횟수를 모델링하는 데 주로 사용됩니다. 포아송 분포의 확률질량함수는 다음과 같습니다.

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

여기서, k는 이벤트가 발생하는 횟수(0, 1, 2, ...)를 의미하고, λ는 단위 시간 또는 공간당 평균 이벤트 발생 횟수를 의미합니다. e는 자연상수(약 2.71828)이며, $k!$는 k의 팩토리얼을 의미합니다. 파이썬의 경우 SciPy 라이브러리의 `stats` 모듈에서 `poisson` 클래스를 사용하여 베르누이 분포에 대한 함수들을 사용할 수 있습니다.

```
from scipy.stats import poisson
poisson.pmf(k, mu)
poisson.cdf(k, mu)
poisson.ppf(q, mu)
poisson.rvs(mu, size=None, random_state=None)
```

균일분포 (a, b)

특정 구간의 값이 동일한 가능성을 갖는 균일확률변수의 경우, 확률변수가 가질 수 있는 시작점(a)와 끝점(b)을 알면, 누구나 같은 확률밀도함수를 그릴 수 있습니다. 균일분포의 확률밀도함수는 다음과 같습니다.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$$

파이썬의 경우 SciPy 라이브러리의 `stats` 모듈에서 `uniform` 클래스를 사용하여 베르누이 분포에 대한 함수들을 사용할 수 있습니다.

```
from scipy.stats import uniform
uniform.pdf(x, loc=0, scale=1)
uniform.cdf(x, loc=0, scale=1)
uniform.ppf(q, loc=0, scale=1)
uniform.rvs(loc=0, scale=1, size=None, random_state=None)
```

- 주의: `loc`은 구간시작점, `scale`은 구간 길이

지수분포 (λ)

지수분포는 연속형 확률변수를 나타내는 분포로, 사건 발생 간격이나 기다림 시간을 모델링하는 데 주로 사용됩니다. 지수분포의 확률밀도함수는 다음과 두 가지 형태를 사용합니다.

1. θ 형태:

$$f(x;\theta) = \frac{1}{\theta}e^{-\frac{x}{\theta}}, \quad x \geq 0, \theta > 0$$

2. λ 형태:

$$f(x;\lambda) = \lambda e^{-\lambda x}, \quad x \geq 0, \lambda > 0$$

파이썬의 경우 SciPy 라이브러리의 stats 모듈에서 expon 클래스를 사용하여 베르누이 분포에 대한 함수들을 사용할 수 있습니다.

```
from scipy.stats import expon
expon.pdf(x, scale)
expon.cdf(x, scale)
expon.ppf(q, scale)
expon.rvs(scale=1, size=None, random_state=None)
```

- 주의: λ 값이 주어질 경우, scale = 1/lambda로 설정

정규분포 (μ, σ^2)

확률변수 X가 정규분포를 따른다는 것을 다음과 같이 기호로 나타냅니다.

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

종모양 분포로 널리 알려져있는 정규분포의 확률밀도함수는 평균 (μ)과 분산 (σ^2)를 통하여 정의됩니다.

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

파이썬의 경우 SciPy 라이브러리의 stats 모듈에서 norm 클래스를 사용하여 정규분포에 대한 함수들을 사용할 수 있습니다.

```
from scipy.stats import norm
norm.pdf(x, loc=0, scale=1)
norm.cdf(x, loc=0, scale=1)
norm.ppf(q, loc=0, scale=1)
norm.rvs(loc=0, scale=1, size=None, random_state=None)
```

분포 관련 계산 연습하기

예제 (정규분포 관련 기본 계산)

- 확률밀도함수 (PDF)

평균이 2이고 표준편차가 3인 정규분포에서 $x = 1$에서의 확률밀도를 계산하는 코드는 다음과 같습니다.

```
from scipy.stats import norm
norm.pdf(1, loc=2, scale=3)
```

```
## 0.12579440923099774
```

- 누적분포함수 (CDF)

평균이 0이고 표준편차가 2인 정규분포에서 $P(X \leq 1)$의 확률을 계산하는 코드는 다음과 같습니다.

```
norm.cdf(1, loc=0, scale=2)
```

```
## 0.6914624612740131
```

- 퀀타일 함수 (PPF)

평균이 0이고 표준편차가 1인 정규분포에서 하위 95% 확률에 대응하는 확률변수의 값을 계산하는 코드는 다음과 같습니다.

```
norm.ppf(0.95, loc=0, scale=1)
```

```
## 1.6448536269514722
```

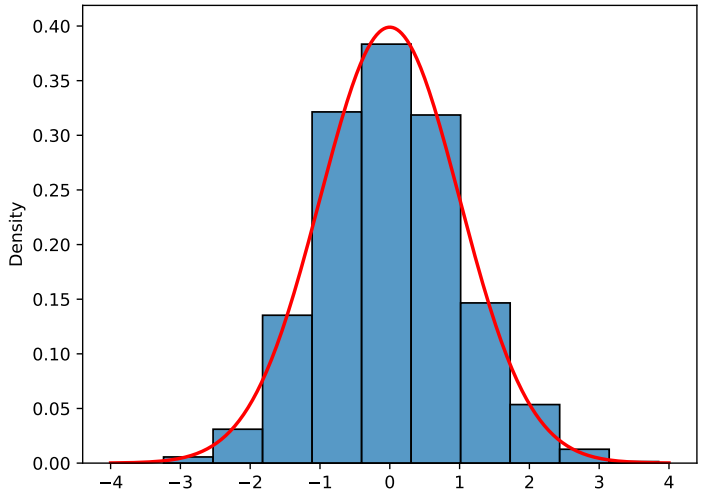
- 랜덤 샘플 생성 (RVS)

각 분포의 rvs 함수를 사용하여 랜덤 샘플을 생성할 수 있습니다. 예를 들어, 평균이 0이고 표준편차가 1인 정규분포에서 1000개의 랜덤 샘플을 생성하는 코드는 다음과 같습니다.

```
samples = norm.rvs(loc=0, scale=1, size=1000, random_state=42)
```

이렇게 뽑힌 표본들을 사용하여 히스토그램을 그리고, 정규분포의 확률밀도 함수를 겹쳐 그리도록 하겠습니다.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# bins: 막대수, kde: 커널 density, stat='density': 히스토그램 높이 조정
sns.histplot(samples, bins=10, kde=False, stat='density')

# 표준 정규 분포의 PDF를 그리기
x = np.linspace(-4, 4, 1000)
y = norm.pdf(x, loc=0, scale=1)
plt.plot(x, y, 'r', linewidth=2)
plt.show()
```



표본들이 앞에서 살펴본 종모양의 정규분포의 확률밀도함수를 닮아있는 것을 알 수 있습니다.

예제 (이산형 확률변수)

앞에서 구한 주사위 문제의 확률분포표를 사용하여 다음 확률을 구하세요.

$$P(3 < X \leq 7)$$

x	2	3	4	5	6	7	8	9	10	11	12
Prob.	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

예제 (이산형 확률변수 2)

한 웹사이트에는 평균적으로 하루에 10번의 트래픽이 발생합니다. 이 웹사이트의 트래픽은 포아송 분포를 따른다고 가정하겠습니다. 이때, 하루에 트래픽이 15번 이상 발생할 확률을 구하시오.

해답

하루에 발생하는 트래픽 수를 확률변수 X라고 설정하면, X는 $\lambda = 10$인 포아송 분포 따른다고 볼 수 있습니다. 따라서, 확률질량함수는 다음과 같습니다.

$$P(X = x) = \frac{10^x e^{-10}}{x!}$$

$P(X \geq 15)$를 구해야 하므로, 이는 $1 - P(X < 15)$로 계산할 수 있습니다. 즉, 0부터 14까지의 확률을 모두 더한 후 이를 1에서 빼면 됩니다.

$$P(X \geq 15) = 1 - \sum_{k=0}^{14} \frac{10^k e^{-10}}{k!}$$

이를 계산하면, 하루에 트래픽이 15번 이상 발생할 확률은 약 0.0835입니다. 위의 코드를 파이썬으로 그대로 계산하면 다음과 같습니다.

```python
from math import exp, factorial

# Parameters
lambda_ = 10
x_max = 14

# Calculate the probability
prob = 1 - sum([(lambda_**x * exp(-lambda_)) / factorial(x) for x in range(x_max + 1)])

print(f"P(X ⩾ 15) = 1 - P(X < 15) = {prob:.4f}")
```

```
## P(X ⩾ 15) = 1 - P(X < 15) = 0.0835
```

- SciPy 라이브러리 사용하는 법 (ADP 시험장에서 우리가 사용할 방법)

```python
from scipy.stats import poisson

# Calculate the probability
prob = 1 - poisson.cdf(14, 10)
print(f"P(X ⩾ 15) = 1 - P(X < 15) = {prob:.4f}")
```

```
## P(X ⩾ 15) = 1 - P(X < 15) = 0.0835
```

예제 (연속형 확률변수)

λ값이 0.5인 지수분포를 따르는 확률변수 X가 2보다 크고 5보다 작게 될 확률을 계산하시오.

해결 방법 1 (확률밀도함수 적분, 이해 만 할 것)

확률변수 X가 2보다 크고 5보다 작게 될 확률을 계산하는 과정은 다음과 같습니다:

1. 우선, 지수분포의 확률밀도함수 $f(x;\lambda) = \lambda e^{-\lambda x}$에서 $\lambda = 0.5$를 대입하면, $f(x;0.5) = 0.5e^{-0.5x}$가 됩니다.
2. 이제, $P(2 < X < 5)$를 계산하기 위해 $x = 2$에서 $x = 5$까지의 범위에 대해 위의 확률밀도함수를 적분합니다. 이를 수식으로 표현하면 다음과 같습니다:

$$P(2 < X < 5) = \int_2^5 0.5e^{-0.5x}\, dx$$

3. 이 적분은 지수함수의 적분 공식 $\int e^{ax}dx = \frac{1}{a}e^{ax}$를 사용하여 계산할 수 있습니다. 여기서 $a = -0.5$입니다. 따라서, 적분 결과는 다음과 같습니다:

$$\int_2^5 0.5e^{-0.5x}\, dx = [-e^{-0.5x}]_2^5$$

4. 이제, 적분 결과를 $x = 5$와 $x = 2$에 대입하여 계산합니다:

$$[-e^{-0.5x}]_2^5 = -e^{-0.5 \cdot 5} - (-e^{-0.5 \cdot 2}) = -e^{-2.5} + e^{-1} \approx 0.286$$

따라서, 확률변수 X가 2보다 크고 5보다 작게 될 확률은 약 0.286입니다.

해결 방법 2 (누적분포함수, 개념 꼭 알아둘 것!)

λ가 0.5인 지수분포의 확률변수 X에 대한 누적분포함수(CDF)는 다음과 같습니다.

$$P(X \leq x) = F(x) = 1 - e^{-0.5x}, \quad x \geq 0$$

따라서, 확률변수 X가 2보다 크고 5보다 작게 될 확률을 계산하려면, $X = 5$에서의 누적확률에서 $X = 2$에서의 누적확률을 빼면 됩니다. 즉, 다음과 같습니다.

$$P(2 < X < 5) = F(5) - F(2) = (1 - e^{-0.5 \cdot 5}) - (1 - e^{-0.5 \cdot 2})$$

이를 계산하면, 확률변수 X가 2보다 크고 5보다 작게 될 확률은 약 0.286입니다.

해결 방법 3 (SciPy, ADP 시험장에서 우리가 사용할 방법)

주어진 모수값을 `expon.cdf`에 넣을때, scale 옵션을 사용하여 역수로 넣어줘야 함에 주의합니다.

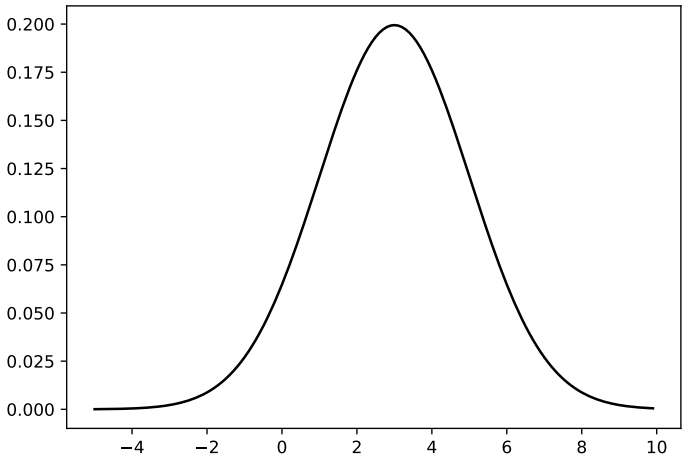
```
from scipy.stats import expon

# Parameters
lambda_ = 0.5

# Calculate the probability
prob = expon.cdf(5, scale=1/lambda_) - expon.cdf(2, scale=1/lambda_)

print(f"P(2 < X < 5) = {prob:.3f}")
```

```
## P(2 < X < 5) = 0.286
```

예제 (정규분포의 확률밀도함수)

평균 3, 분산이 4인 정규분포의 확률밀도함수는 다음과 같습니다.

```
from scipy.stats import norm     # scipy.stats에서 norm을 불러옴
import matplotlib.pyplot as plt  # 시각화 라이브러리
import numpy as np               # numpy 라이브러리를 np로 불러옴

x = np.arange(-5, 10, 0.1)
y = norm.pdf(x, loc=3, scale=2)  # x=확률변수, loc=평균, scale=표준편차
plt.plot(x, y, color='k')        # color옵션을 통해 색변경 가능
plt.show()
```



모수인 μ는 종모양 분포의 중심점을 나타내고, σ은 표준편차를 나타내고, 종모양 분포의 퍼짐을 나타내는 모수입니다. 위 분포를 따르는 확률변수 X에서 4.5보다 큰 숫자가 나올 확률 구하세요.

$$P(X \geq 4.5) = \int_{4.5}^{\infty} f_X(x)dx$$

해결 방법

```
from scipy.stats import norm

# Calculate the probability
prob = 1 - norm.cdf(4.5, 3, 2)

print(f"P(2 < X < 5) = {prob:.3f}")
```

```
## P(2 < X < 5) = 0.227
```

1.6 통계적 검정의 근본(fundamental)을 이해하자.

통계적 추정과 검정의 중요한 2가지 정리

평균이 μ이고, 분산이 σ^2인 확률변수 $X_i, i = 1, ..., n$에 대하여 다음이 성립한다.

표본평균의 수렴

$$\overline{X} \xrightarrow{p} \mu$$

- 점추정을 하는 근본 이론.

표본평균의 분포

표본평균은 다음과 같은 분포를 따르게 된다.

$$\overline{X} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$$

- 신뢰구간을 만들때 사용되는 근본 이론
- 표본의 크기가 크면 무조건 정규분포 수렴
- 작다면? X의 원분포가 정규분포일때는 잘 수렴. **그렇지 않는 경우**에는 잘 작동하지 않음.

위 두 원리는 무조건 이해하고 있어야 함

- https://onlinestatbook.com/stat_sim/sampling_dist/index.html

1. 표본평균 (파란색 블록)의 분포의 중심과 분산 체크하기
    - 분산과 검정색 블록 숫자와의 관계 이해하기
2. 모분포의 생김새와 **상관없이** 표본평균의 분포는 종모양 (정규분포) 을 이룸.

정규분포와 표본평균의 분포

평균이 30, 표준편차가 5인 분포를 따르는 확률변수 X에서 추출한 표본크기 30인 표본들의 표본평균의 확률밀도함수을 그려보세요.

```
import math

x = np.arange(25, 35, 0.1)
y = norm.pdf(x, loc=30,scale=5/math.sqrt(30))
plt.plot(x, y, color='k')
plt.xlabel('')
plt.ylabel('p.d.f')
plt.show()
```

통계적 추정 (statistical inference)

표본에 담겨있는 정보를 이용하여, 데이터를 발생시킨 확률변수의 모수를 예측하는 과정

- 예: 표본평균을 사용해서 모평균 μ를 예측
- 예: 표본분산을 사용해서 모분산 σ^2을 예측

점 추정 (point estimation)

모수의 값을 특정값 하나로 추정하는 방법

- 데이터 셋 A
  2.51, 5.21, 4.395, 4.439, 6.592
- 데이터 셋 B
  2.51, 5.21, 4.395, 4.439, 6.592, 1004.6292, -995.3708

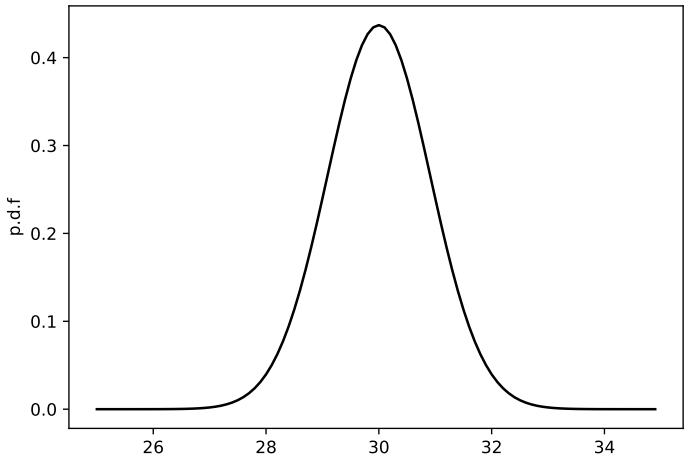


그림 1.4: 표본 평균의 분포는 표본의 크기가 커짐에 따라 정규분포를 따르게 된다.

한계점

- 모수값을 특정 값으로 추정: 맞출 확률은?
- 두 데이터 셋의 표본 평균값: 4.6292으로 같음
- 추정의 정확도를 나타내기가 어려움.

구간 추정 (interval estimation)

특정값을 사용하여 확률변수의 모수값를 예측하는 점추정의 대안으로 구간을 사용하여 추정

- 예: $(\overline{x} - \mu, \overline{x} + \mu)$ 구간은 68% 확률로 모수를 포함한다.

연습문제

농구 vs. 축구

슬통 마을의 많은 사람들이 농구와 축구 관람을 좋아한다고 한다. 마을의 40%는 농구 경기를 좋아하고, 70%는 축구 경기를 좋아하며, 농구와 축구 관람을 모두 좋아하는 비율이 20%라고 한다.

1) 마을 사람을 무작위로 한 명 선택했을 때, 그 사람이 농구와 축구 둘 다 좋아하지 않는 사람일 확률을 구하세요.
2) 마을 사람을 무작위로 한 명 선택해서, 농구를 좋아하냐고 물어보았다. 그 사람이 농구를 좋아하지 않는다고 대답했다면, 축구를 좋아할 확률을 구하세요.

상자 속의 색깔공

두 개의 상자 A, B가 놓여있다. A 상자에는 빨간색 공이 3개, 파란색 공이 3개 들어있고, B 상자에는 빨간색 공이 4개, 파란색 공이 6개 들어있다고 한다.

1) 각각의 상자에서 하나씩 공을 꺼낼 때, 두 공이 같은 색깔일 확률을 구하세요.

2) 이번에는 슬통이가 두 개 상자 중 하나에서 공을 하나 꺼내왔다고 한다. 뽑힌 공의 색깔을 보니 빨간색이었다. 슬통이가 이 공을 상자 A에서 꺼냈을 확률을 구하세요.

ADP 표본점수

2022년에 실시 된 ADP 실기 시험의 통계파트 표준점수는 평균이 30, 표준편차가 5인 정규분포를 따른다고 한다.

1) ADP 실기 시험의 통계파트 표준점수의 밀도함수를 그려보세요.

2) ADP 수험생을 임의로 1명을 선택하여 통계 점수를 조회했을때 45점 보다 높은 점수를 받았을 확률을 구하세요.

3) 슬통이는 상위 10%에 해당하는 점수를 얻었다고 한다면, 슬통이의 점수는 얼마인지 계산해보세요.

4) 슬기로운 통계생활의 해당 회차 수강생은 16명이었다고 한다. 16명의 통계 파트 점수를 평균내었을 때, 이 평균값이 따르는 분포의 확률밀도 함수를 1번의 그래프와 겹쳐 그려보세요.

5) 슬기로운 통계생활 ADP 반 수강생들의 통계점수를 평균내었다고 할 때, 이 값이 38점보다 높게 나올 확률을 구하세요.

Covid 19 발병률

Covid-19의 발병률은 1%라고 한다. 다음은 이번 코로나 사태로 인하여 코로나 의심 환자들 1,085명을 대상으로 슬통 회사의 "다잡아" 키트를 사용하여 양성 반응을 체크한 결과이다.

키트 \ 실제	양성	음성
양성	370	10
음성	15	690

1) 다잡아 키트가 코로나 바이러스에 걸린 사람을 양성으로 잡아낼 확률을 계산하세요.

2) 슬통 회사에서 다잡아 키트를 사용해 양성으로 나온 사람이 실제로는 코로나 바이러스에 걸려있을 확률을 97%라며, 키트의 우수성을 주장했다. 이 주장이 옳지 않은 이유를 서술하세요.

3) Covid-19 발병률을 사용하여, 키트의 결과값이 양성으로 나온 사람이 실제로 코로나 바이러스에 걸려있을 확률을 구하세요.

카이제곱분포와 표본분산

자유도가 k인 카이제곱분포를 따르는 확률변수 X를

$$X \sim \chi^2(k)$$

과 같이 나타내고, 이 확률변수의 확률밀도함수는 다음과 같습니다.

$$f_X(x;k) = \frac{1}{2^{k/2}\Gamma(k/2)}\, x^{k/2-1}e^{-x/2}$$

다음의 물음에 답하세요.

1) 자유도가 4인 카이제곱분포의 확률밀도함수를 그려보세요.

2) 다음의 확률을 구해보세요.

$$P(3 \leq X \leq 5)$$

3) 자유도가 4인 카이제곱분포에서 크기가 1000인 표본을 뽑은 후, 히스토그램을 그려보세요.

4) 자유도가 4인 카이제곱분포를 따르는 확률변수에서 나올 수 있는 값 중 상위 5%에 해당하는 값은 얼마인지 계산해보세요.

5) 3번에서 뽑힌 표본값들 중 상위 5%에 위치한 표본의 값은 얼마인가요?

6) 평균이 3, 표준편차가 2인 정규분포를 따르는 확률변수에서 크기가 20인 표본, $x_1, ..., x_{20}$,을 뽑은 후 표본분산을 계산한 것을 s_1^2이라 생각해보죠. 다음을 수행해보세요!

- 같은 방법으로 500개의 s^2들, $s_1^2, s_2^2, ..., s_{500}^2$ 발생시킵니다.
- 발생한 500개의 s^2들 각각에 4.75를 곱하고, 그것들의 히스토그램을 그려보세요. (히스토그램을 그릴 때 `probability = TRUE` 옵션을 사용해서 그릴 것)
- 위에서 그린 히스토그램에 자유도가 19인 카이제곱분포 확률밀도함수를 겹쳐그려보세요.

제 2 장

통계적 추정과 가설 검정 기본기 다지기

2.1 통계적 추정

통계적 추정은 데이터를 사용하여 데이터를 발생시킨 모수의 값을 예측하는 방법입니다. 현재 내가 가진 데이터를 발생시킨 분포의 모수를 값을 추정하는 방법에는 두 가지 방법이 존재합니다.

- 점추정
- 구간 추정

점 추정의 경우 ADP 시험에는 잘 나오지 않으며, **구간 추정의 경우 기출 문제로 나온 적이 있으므로, 정확하게 이해하고 있어야 합니다.** 이번 챕터에 사용되는 기본적인 파이썬 모듈은 다음과 같습니다.

```
import numpy as np
import scipy.stats as sp
import matplotlib.pyplot as plt
```

구간추정

모수가 존재할 것으로 예상되는 구간을 추정하는 방법을 구간 추정이라고 합니다. 이는 표본으로부터 얻은 정보를 바탕으로 구간을 계산하고, 모집단의 모수(평균, 비율 등)가 이 구간 안에 존재할 확률이 특정 수준 이상이라는 것을 보여주는 방법입니다.

구간 추정을 공부 할 때 가장 중요한 점은 신뢰구간은 확률변수라는 것을 이해하는 것 입니다.

위의 문장을 정확하게 이해했다면, 여러분의 통계 실력이 한층 높아지리라 생각합니다.

모평균에 대한 구간 추정

$X_1, X_2, ..., X_n$ 을 정규분포 μ, σ^2 를 따르는 확률변수라고 하고, 이 확률변수들에서 랜덤 표본들을 뽑았을 때, 모평균 μ에 대한 $100(1-\alpha)$ 신뢰구간은 다음과 같이 2가지 경우로 나눠 구할 수 있습니다.

모분산 σ^2 이 알려져 있는 경우

$$C.I. = \overline{X} \pm z_{\alpha/2}\frac{\sigma}{\sqrt{n}}$$

z_α는 표준정규분포에서의 임계값(critical value)이라고 부르며, $100(1-\alpha)$ 백분위수를 의미합니다.

모분산 σ^2이 알려져 있지 않은 경우

$$C.I. = \overline{X} \pm t_{\alpha/2,n-1}\frac{S_n}{\sqrt{n}}$$

$t_{\alpha,n}$는 자유도가 n인 t분포에서의 임계값(critical value)이라고 부르며, $100(1-\alpha)$ 백분위수를 의미합니다.

모평균 계산 예제 파이썬으로 구하기 (모분산 σ를 모르는 경우)

다음과 같이 데이터가 주어졌을 때, 모평균에 대한 95% 신뢰구간을 구해보도록 하겠습니다.

```
# 표본 데이터
data = [4.3, 4.1, 5.2, 4.9, 5.0, 4.5, 4.7, 4.8, 5.2, 4.6]
```

먼저, 우리는 주어진 데이터로부터 표본 평균($\overline{x}$)를 계산합니다.

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

여기서 n은 표본의 크기이고, x_i는 표본 데이터의 각 값입니다. 그 다음, 표본의 표준편차(s)를 계산합니다.

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

표본의 표준편차를 표본의 크기의 제곱근으로 나누어 준 표준 오차 (standard error)를 계산합니다.

$$s.e. = \frac{s}{\sqrt{n}}$$

마지막으로, 95% 신뢰구간은 표본 평균에서 $\pm$ 표준 오차의 $t_{0.025,9} = 2.26$배 떨어진 범위가 됩니다. 데이터를 통하여 모평균에 대한 구간 추정을 수행하는 파이썬 코드를 보겠습니다.

```
from scipy.stats import t

# 표본 평균
mean = np.mean(data)

# 표본 크기
n = len(data)

# 표준 오차
se = np.std(data, ddof=1) / np.sqrt(n)
```

```
# 95% 신뢰구간
round(mean - t.ppf(0.975, n-1) * se, 3); round(mean + t.ppf(0.975, n-1) * se, 3)
```

```
## 4.469
## 4.991
```


같은 신뢰구간을 t.interval()를 사용하여 구하는 방법은 다음과 같습니다.

```
# t.interval로 구하기
ci = t.interval(0.95, loc=mean, scale=se, df=n-1)
print("95% 신뢰구간: ", [round(i, 3) for i in ci])
```

```
## 95% 신뢰구간:  [4.469, 4.991]
```

참고로 ADP 25회 기출문제의 경우, $z_{0.05}$, $z_{0.025}$, $t_{0.05}$, 그리고 $t_{0.025}$의 값을 모두 제공하여, 시험자가 정확한 공식을 사용할 수 있는지 평가하였습니다.

2.2 통계적 검정

통계 검정은 "모집단 분포의 모수에 대한 가설의 참, 거짓 여부"를 관찰된 데이터가 우연인지 아니면 특정한 효과가 있는지를 판단하여 검증하는 과정입니다. 이를 위해 귀무가설(null hypothesis, H_0)과 대립가설(alternative hypothesis, H_a 또는 H_1)이라는 두 가지 가설을 설정하고, 귀무가설이 참이라는 가정하에 주어진 검정통계량 값보다 극단적인 값을 관찰 할 확률을 계산하여 결과를 도출합니다.

검정에서 사용되는 가설들

귀무가설 (H_0 : Null Hypothesis)

귀무가설은 기본적으로 전제가 되는 가정 혹은 **기존에 참이라고 받아들여지고 있는** 가정입니다.

$$H_0 : \mu = \mu_0$$

- 귀무가설은 위와 같이 항상 등호(=) 형식을 띄고 있습니다. 꼭 기억해 주세요!
- 귀무가설을 말로 표현 할 때, 보통 '0과 같다', **'차이가 없다'**, 혹은 '효과가 없다' 라고 하는 형태를 많이 띕니다. (Null의 의미)

귀무가설 예 업체 A에서 생산되는 베어링 평균 지름의 길이는 3 cm 이다.

$$H_0 : \mu = 3$$

한 가지 주목 할 점은 앞에서 귀무가설을 기호로 표현함에 있어서, μ_0 가 3으로 설정 된 상황입니다.

대립가설 (H_1 : Alternative Hypothesis)

대립가설은 현재까지 받아들여지고 있는 귀무가설과 대립되는 가설입니다. 따라서, 대립가설을 받아들이기 위해서는 확실한 증거 필요합니다.

- **귀무가설의 반대 상황**을 나타냅니다.
- 귀무가설이 기각 되었을 때 받아들여지게 된다.

대립가설 예 업체 A에서 생산되는 베어링 평균 지름의 길이는 3 cm 가 아니다.

$$H_A : \mu \neq 3$$

위와 같은 형태의 대립가설을 양측검정(two-tailed test)에 대응하는 대립 가설이라고 합니다. 하지만, 다음과 같이 경우에 따라서 특정 방향에만 우리의 관심이 있는 경우가 있습니다.

업체 A에서 생산되는 베어링 평균 지름의 길이는 3 cm 보다 작다.

$$H_A : \mu < 3$$

업체 A에서 생산되는 베어링 평균 지름의 길이는 3 cm 크다.

$$H_A : \mu > 3$$

위의 두 가지 형태의 대립가설을 단측검정(one-tailed test)에서의 대립가설이라고 부릅니다.

검정통계량이란?

검정통계량(test statistic)은 가설 검정에 사용되는 확률변수입니다. 데이터를 사용하여 검정통계량의 실현치를 계산하고, 그 값에 따라 귀무가설을 기각 할 지 말지에 대한 통계적 의사 결정을 내립니다.

Z 검정통계량 (Z-test statistic)

Z 검정통계량은 표본 평균이 정규분포를 따르는 경우에 모집단의 평균을 검정할 때 사용할 수 있습니다. 일반적으로, 모집단의 분산이 알려져 있고 표본 크기가 충분히 클 때 사용됩니다.

모분포의 분산 σ^2을 알고 있는 경우, n개의 표본으로 만든 표본평균 $\overline{X}$의 분포를 표준화시켜 가설검정을 할 수 있습니다. Z 검정통계량은 다음과 같습니다.

$$Z = \frac{\overline{X} - \mu_0}{\sigma/\sqrt{n}} \xrightarrow{n} \mathcal{N}(0, 1^2)$$

위에서 알 수 있듯, n이 커지면, Z 검정통계량의 분포는 표준정규분포로 수렴하게 됩니다. 따라서, Z 검정을 수행하기 전에 다음 가정 2가지가 충족되어야 합니다.

1. 모집단의 분산이 알려져 있어야 합니다.
2. 모집단이 정규분포를 따르거나, 표본 크기가 충분히 커야 합니다. (일반적으로 $n \geq 30$).

스튜던트 t 검정통계량

많은 경우 실무에서 기본적으로 사용하는 검정통계량이며, 스튜던트 정리에 기반하여 검정통계량의 분포가 유도되었습니다. 따라서, t 검정을 적용하기 위해서는 스튜던트 정리에서 가정하는 가정들을 만족해야만 검정의 결과를 믿을 수 있습니다.

따라서 t분포의 정의에 의하여 t 검정 통계량은 자유도가 $n-1$이 t분포를 따르게 됩니다.

$$T = \frac{\overline{X} - \mu_0}{S/\sqrt{n}} \sim t_{n-1}$$

여기서 $\overline{X}$는 표본 평균 확률변수, S은 표본 표준편차 확률변수, 마지막으로 n은 표본의 크기를 나타냅니다.

스튜던트 정리

정말 많이 사용되는 t 검정의 근간을 이루는 정리이므로 꼭 알아둬야 합니다. ADP 시험에서 t검정을 적용하기 전, 가정 체크를 하는 근본적인 이유가 됩니다.

스튜던트 정리

평균이 μ이고 분산이 σ^2인 정규분포를 따르는 독립인 $X_1, X_2, ..., X_n$에 대하여 다음이 성립한다.
1. $\overline{X}$는 정규분포 평균이 μ, 분산이 σ^2/n인 분포를 따른다.
2. $\overline{X}$과 S^2은 독립이다.
3. $\frac{(n-1)S^2}{\sigma^2}$은 자유도가 $n-1$인 카이제곱분포를 따른다.

위에서 사용된 $\overline{X}$와 S^2는 다음과 같이 정의한다.

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

$$S^2 = \frac{\sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2}{n-1}$$

t 분포 확률변수

자유도가 ν인 t 분포를 따르는 확률변수는 다음과 같이 정의 할 수 있습니다.

$$\frac{Z}{\sqrt{V/\nu}}$$

여기서 Z는 표준정규분포, V는 자유도가 ν인 카이제곱분포를 따르는 확률변수입니다.

t 검정통계량의 분포 유도

t 검정 통계량은 다음과 같이 2가지 파트로 생각해 볼 수 있습니다.

$$\frac{\overline{X}-\mu_0}{S/\sqrt{n}} = \frac{\frac{\overline{X}-\mu_0}{\sigma/\sqrt{n}}}{\frac{S}{\sigma}} = \frac{\frac{\overline{X}-\mu_0}{\sigma/\sqrt{n}}}{\sqrt{\frac{(n-1)S^2}{\sigma^2}/(n-1)}}$$

스튜던트 정리의 1번에 의하여

$$\frac{\overline{X}-\mu_0}{\sigma/\sqrt{n}} \sim \mathcal{N}(0,1)$$

이 성립합니다. 또한 스튜던트 정리의 3번

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2_{n-1}$$

에 의하여 S/σ는 자유도 $n-1$인 카이제곱분포를 따르는 확률변수를 그것의 자유도로 나누고 제곱근을 씌운 꼴로 볼 수 있다.

$$\frac{S}{\sigma} = \sqrt{\frac{(n-1)S^2}{\sigma^2}/(n-1)}$$

따라서 t분포의 정의에 의하여 t 검정 통계량은 자유도가 $n-1$이 t분포를 따르게 된다.

$$\frac{\overline{X}-\mu_0}{S/\sqrt{n}} \sim t_{n-1}$$

위의 정리에서와 마찬가지로

- $\overline{X}$는 표본 평균 확률변수
- S은 표본 표준편차 확률변수
- n은 표본의 크기

를 의미합니다.

t 검정통계량 계산

가장 기본적인 검정 형태인 1 표본 t 검정에 대하여 데이터를 사용하여 공부해봅시다. 기본 가정은 모분산 σ^2을 모른다는 것이며, 관찰값들이 하나의 동일한 분포에서 독립적으로 뽑혔다는 것을 가정하고 있습니다. 이러한 상태에서 모평균 μ이 특정값과 같은지, 같지 않는지에 대한 검정입니다.

목표

데이터가 주어졌을 때, 주어진 관찰값 (표본)들을 발생시킨 분포의 모평균이 7인지 검정하려고 합니다.

예제 데이터

다음 15개의 데이터가 있다고 가정합시다.

4.62, 4.09, 6.2, 8.24, 0.77, 5.55, 3.11, 11.97, 2.16, 3.24, 10.91, 11.36, 0.87, 9.93, 2.9

가설 설정

대응하는 귀무가설, 대립가설을 다음과 같이 설정할 수 있습니다.

$$H_0 : \mu = 7 \quad vs. \quad H_A : \mu \neq 7$$

이 경우, 검정은 양측검정에 속하게 됩니다.

검정통계량 값 계산

$$t = \frac{\overline{x} - \mu_0}{s/\sqrt{n}}$$

위의 데이터를 사용하여 계산된 t 검정 통계량 값은 약 -1.279가 나옵니다. 이 값은 스튜던트 정리에 의하여, 자유도가 14인 t 분포에서 뽑혀진 관찰값이라 생각할 수 있습니다.

```
x = [4.62, 4.09, 6.2, 8.24, 0.77, 5.55, 3.11,
     11.97, 2.16, 3.24, 10.91, 11.36, 0.87, 9.93, 2.9]
t_value = (np.mean(x)-7) / (np.std(x, ddof=1) / np.sqrt(len(x)))
round(t_value, 3)
```

```
## -1.279
```

기각역

기각역은 가설 검정에서 귀무가설이 기각시킬 수 있는 **영역**을 의미하고, 구체적으로 주어진 귀무가설이 참인 상황에서 잘 볼 수 없는 표본들이 위치한 영역을 의미합니다.

- 기각역은 유의수준 α와 가설의 모양에 따라서 정해집니다.

위의 양측 검정에서는 t 검정 통계량이 따르는 분포인 자유도가 $n-1$인 t분포의 양 끝단 (각 끝에서 확률을 $\alpha/2$씩 차지하는) 부분입니다. 다음은 유의수준 $\alpha = 0.05$에 해당하는 기각역을 시각화하는 파이썬 코드입니다.

```
from scipy.stats import t

n = len(x)
x = np.arange(-3, 3, 0.1)
y = t.pdf(x, df=n-1)
reject_range = t.ppf(0.975, df=n-1)

plt.plot(x, y, color='k')
plt.axhline(0, color='black', linewidth=0.5)
plt.fill_between(x, y,
    where=(x < -reject_range) | (x > reject_range),
    color='yellow')
plt.plot(t_value, 0, 'ro') # 검정통계량 표현
plt.show()
```

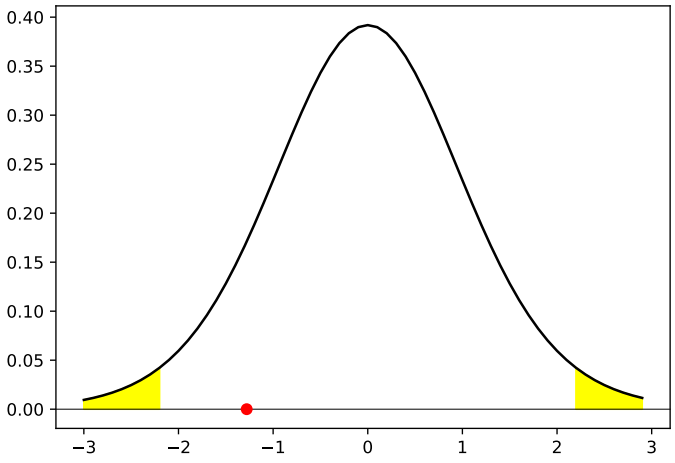


그림 2.1: 자유도가 14인 t분포의 양측검정 기각역 시각화

기각역에 검정 통계량 값이 속하게 되면 귀무가설을 기각하게 됩니다. 이번 예제에서는 검정통계량 값이 기각역에 속하지 않으므로, 귀무가설을 기각할 수 없습니다.

유의 확률 (p-value)

귀무가설이 참이라는 가정하에서 검정통계량 값이 주어진 관측값과 같거나, 더 극단적인 값을 얻게 될 확률을 의미합니다. 아래 그림의 어두운 부분에 해당하는 확률값이 양측 검정에서의 p-value를 표현한 것입니다.

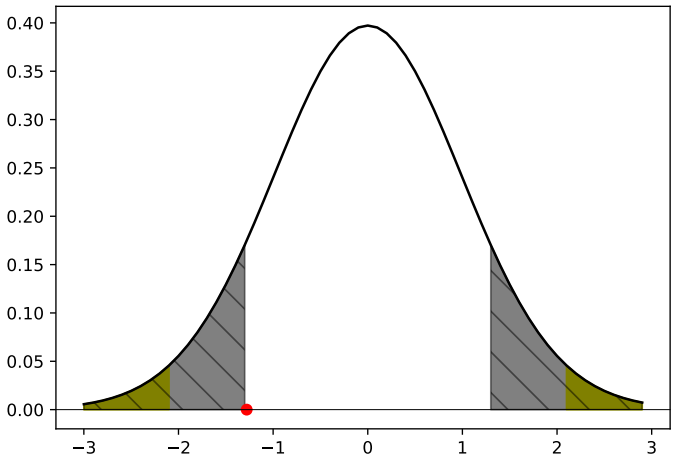


그림 2.2: 자유도가 14인 t 분포의 양측검정 p-value 시각화

가설 형태와 p-value 계산

가설의 모양에 따라서 p-value 계산 시, 양쪽을 고려할 지, 한쪽 만을 고려할 지 결정합니다.

- 양측 검정의 대립가설 형태는 양쪽을 모두 고려합니다.

$$H_a : \mu \neq \mu_0 \quad \text{인 경우} \quad 2P(T \geq |t|)$$

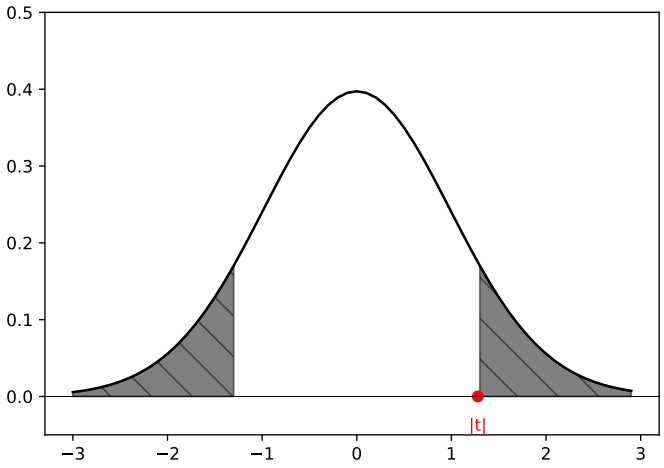


그림 2.3: 양측검정 p-value 시각화

양측검정 `p-value`의 경우, 검정통계량 값에 대응하는 반대편 영역의 확률까지 고려하기 때문에 확률의 2배로 계산된다는 것을 꼭 기억합시다.

- 단측검정의 대립가설 형태는 한쪽 만을 고려합니다.

$$H_a : \mu > \mu_0 \quad \text{인 경우} \quad P(T \geq t)$$

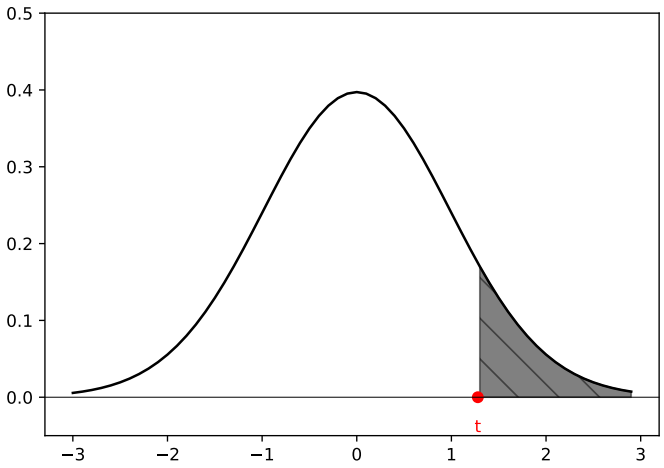


그림 2.4: 단측검정 p-value 시각화

$$H_a : \mu < \mu_0 \quad \text{인 경우} \quad P(T \leq t)$$

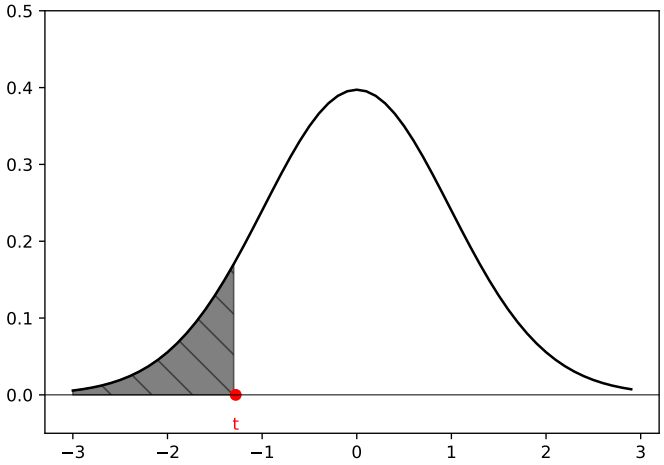


그림 2.5: 단측검정 p-value 시각화

검정력 (Test power)

귀무가설이 참이 아니라는 전제하에, 유의수준이 α인 검정이 귀무가설을 기각할 확률을 의미합니다.

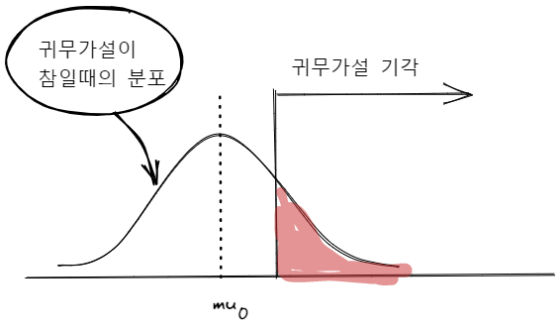
- test power가 높을 수록 좋은 검정 방법이라 말할 수 있습니다.
- 일반적으로 **검정력 0.8이상**인 검정을 좋은 검정이라고 말합니다.

유의수준과 검정력

유의수준 α는 검정 방법을 반복해서 시행했을 때, 얼마나 믿을 만 한가를 나타내는 측정 요소입니다.

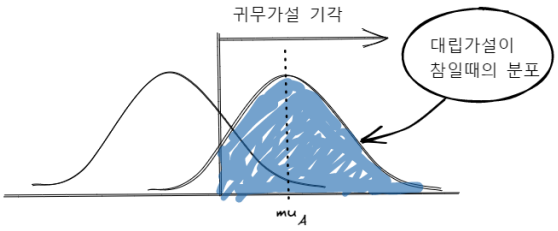
- 유의수준 0.05의 의미: 주어진 검정 방법을 계속해서 시행했을 경우, 귀무가설이 참이라는 전제하에 5% 정도 옳지 않는 판단 (귀무가설을 기각)을 내리게 됩니다.

유의수준은 다음 그림에서처럼 귀무가설이 참일때, 기각역에 해당하는 확률을 의미합니다.



반면, 검정의 검정력은 대립가설이 참이라는 가정하에서, 귀무가설이 기각될 확률을 의미합니다. 따라서, 대립가설이 참일때의 특정 모수값에 해당하는 분포에서 기각역에 해당하는 확률을 의미하게

된다.



t-test 는 언제 작동을 잘 할까?

ADP 자격증 시험 답안 작성 필수 팁

- 통계적 검정을 하기 전, 검정의 가정을 만족하는 상황인지 **필수적으로 체크** 해야 합니다.
- 이론적으로 t 검정은 자료가 꼭! 정규분포를 따른다는 가정이 만족해야만 잘 작동하는 통계적 검정[1]입니다. 따라서 검정을 시작하기 전에 정규분포를 따르는 것을 꼭 체크해줘야 합니다.
- 정규성을 만족을 하지 못한다면? 다른 검정을 시도해야 합니다.

표본의 크기가 작은 경우 ($n < 30$)

- 표본의 원 분포가 정규분포를 따르면 적용 할 수 있습니다.

표본의 크기가 큰 경우 ($n > 30$)

- 중심극한정리에 따라서 표본 평균이 정규분포를 따르는 것이 어느 정도 보장됩니다. 따라서, t 검정이 아닌 정규분포로 근사해서 사용할 수 있습니다. t 분포의 n 무한대인 경우가 표준정규분포가 된다는 사실도 알고 있으면 좋겠죠?

2.3 데이터가 특정 분포를 따르는지 체크하는 방법

t 검정 수행을 위하여 데이터가 정규분포를 따르는지 체크해보아야 합니다. 이 챕터에서는 좀 더 일반적으로 데이터가 특정분포 (정규분포 포함)를 따르는지 체크하는 방법을 배워봅시다.

Quantile-Quantile plot

- QQ Plot의 작동원리 직관: 이론적으로 예상하는 백분위수와 데이터를 사용하여 계산 된 백분위수를 비교

데이터에 대응하는 백분위수 구하기

분위수는 데이터를 일정 그룹으로 나누는 기준이 되는 수를 말하고, 이 중 데이터를 100개 그룹으로 나누는 기준이 수들을 백분위수 (percentile) 이라고 말합니다.

[1] 해당 문제는 위키피디아의 선형계획법 페이지에서 가져옴.

p-th percentile 구하는 방법

$$x_1, ..., x_n$$

와 같이 n개의 데이터가 존재할 때 p번째 백분위수는 다음과 같이 구합니다.

1. 데이터 순서대로 정렬: $x_{(1)}, ..., x_{(n)}$
    - $x_{(1)} \leq ... \leq x_{(n)}$
2. $(n+1) \times p/100$ 를 계산해서 정수부분과 소수 부분을 구분하기
    - 정수부분 j, 소수부분 h
3. p-th percentile 구하는 공식 적용
    - $(1-h)x_{(j)} + hx_{(j+1)}$

위의 과정을 간단한 숫자를 사용해서 구해봅시다. 생각보다 너무 쉬우니, 미리 겁먹지 마세요! 수식을 두려워하지 않는 연습 과정이라고 생각합시다.

$$155, 126, 27, 82, 115$$

위와 5개 숫자를 사용해서 50 백분위수를 구해봅시다.

- 중앙값 (50th percentile): 115
- 3사분위수 (75th percentile): 140.5

```
import numpy as np

x = np.array([155, 126, 27, 82, 115])

#numpy의 percentile()를 통해 분위수 구할 수 있음
q2 = np.percentile(x, 50) #중앙값 (2사분위수)
q3 = np.percentile(x[x>q2], 50) #3사분위수
q1 = np.percentile(x[x<q2], 50) #1사분위수
print('1사분위수:', q1)
```

```
## 1사분위수: 54.5
```

```
print('2사분위수:', q2)
```

```
## 2사분위수: 115.0
```

```
print('3사분위수:', q3)
```

```
## 3사분위수: 140.5
```

데이터에 대응하는 백분위를 한꺼번에 구하는 방법

이번에는 반대로, 데이터에 대응하는 백분위 (percent)를 구하는 방법입니다.

$$155, 126, 27, 82, 115$$

데이터가 주어졌을 때 대응하는 백분위를 한꺼번에 구하는 방법은 다음과 같습니다.

1. 데이터를 순서를 부여한다.
2. $n+1$로 나눠줌.


```
x = np.array([155, 126, 27, 82, 115])
np.sort(x)
```

```
## array([ 27,  82, 115, 126, 155])
```

```
ranks = sp.rankdata(sorted(x)) #sp.rankdata() 안에 정렬된 array 넣기
ranks / (len(x) + 1)
```

```
## array([0.16666667, 0.33333333, 0.5       , 0.66666667, 0.83333333])
```

- 순서와 상관없이 대응되는 백분위 구하기

```
x
```

```
## array([155, 126,  27,  82, 115])
```

```
sp.rankdata(x) / (len(x) + 1)
```

```
## array([0.83333333, 0.66666667, 0.16666667, 0.33333333, 0.5       ])
```

QQ plot을 위한 예제 데이터 다음 데이터에 대응하는 QQ plot을 그려보도록 합시다.

$$4.62, 4.09, 6.2, 8.24, 0.77, 5.55, 3.11, 11.97, 2.16, 3.24, 10.91, 11.36, 0.87$$

데이터에 대응하는 백분위 구하기 앞에서 살펴본 방식으로 백분위를 구합니다.

```
data_x = np.array([4.62, 4.09, 6.2, 8.24, 0.77, 5.55,
                   3.11, 11.97, 2.16, 3.24, 10.91, 11.36, 0.87])
data_percent = sp.rankdata(data_x) / (len(data_x) + 1)
print(data_percent[:6])
```

```
## [0.5        0.42857143 0.64285714 0.71428571 0.07142857 0.57142857]
```

이론적인 백분위수 (percentile) 구하기 위에서 구한 백분위에 대응하는 이론적인 백분위수를 구합니다. 특정 분포의 이론적인 백분위수는 그 분포의 누적분포함수의 역함수를 사용해서 구할 수 있으므로, Python에서 제공하는 `<dist.name>.ppf()` 함수를 사용합니다.

- 데이터의 백분위(%)에 대응하는 이론적인 정규분포의 백분위수(percentile) 구하기

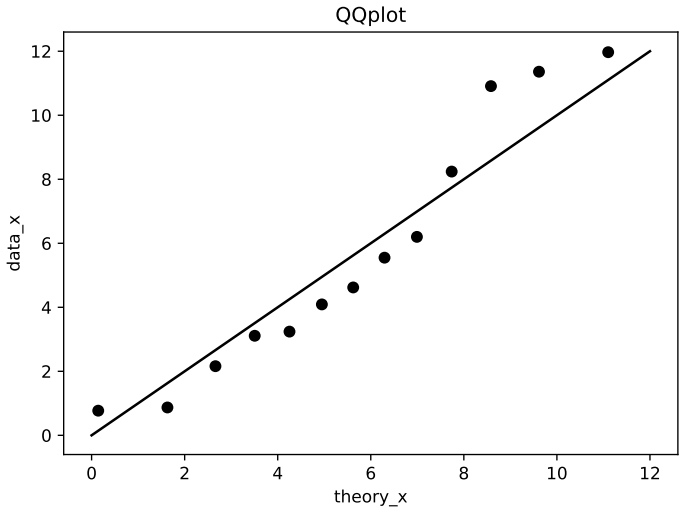
```
theory_x = sp.norm.ppf(data_percent, np.mean(data_x), np.std(data_x))
print(theory_x[:6])
```

```
## [5.62230769 4.94909284 6.99148158 7.73885752 0.14258901 6.29552254]
```

QQ plot 의 아이디어

만약 데이터가 정규분포를 따른다면 이론적으로 구한 값과 데이터에서 구해진 정보가 거의 비슷한 값을 가지고 있어야 할 것 입니다.

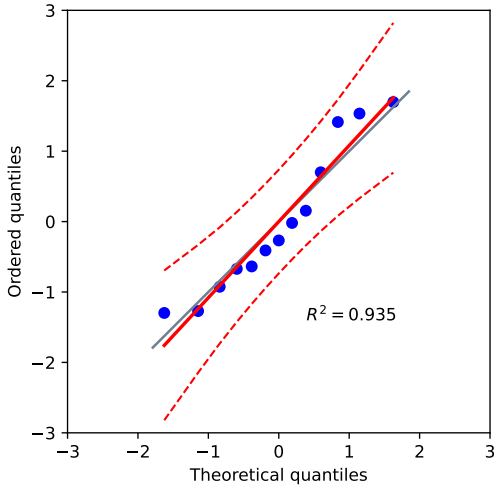
```
plt.scatter(theory_x, data_x, color='k')
plt.plot([0,12], [0,12], 'k')
plt.title('QQplot')
plt.xlabel('theory_x')
plt.ylabel('data_x')
plt.show()
```



따라서, 45도 대각선을 기준으로 점들이 찍혀있다면 데이터가 특정분포를 따르는 것이라고 판단할 수 있습니다. QQ plot은 주어진 데이터를 y 값으로, 앞에서 구한 이론적인 백분위수를 x 값으로하여, 산점도를 그립니다.

QQ plot 실제 시험장 코드 실제 시험장에서는 `pingouin` 패키지의 `qqplot()` 함수를 사용합니다. qqplot() 함수가 간혹 모든 점들을 표시하지 않는 경우가 있습니다. 이 경우, `ylim()`과 `xlim()`를 사용해서 조정해줍니다.

```
# pip install pingouin
import pingouin as pg
ax = pg.qqplot(data_x, dist='norm', confidence=0.95)
plt.ylim(-3, 3);
plt.xlim(-3, 3);
plt.show()
```



Shapiro-Wilk 검정

표본 크기가 50개가 안되는 작은 데이터들의 정규성 검정을 위하여 고안된 검정입니다. 좋은 검정력(power)을 가지고 있는 것 때문에 많이 사용합니다.

- **정규분포 전용** 검정: 모든 검정 대비 최고 검정력

귀무가설과 대립가설

- H_0: 데이터가 정규분포를 따른다.
- H_A: 데이터가 정규분포를 따르지 않는다.

검정통계량

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

SW 검정통계량의 분포를 유도할 수 없고, 필요도 없지만, 다음과 같은 특성들은 알아둬야 합니다.

- a_i 들은 미리 정해진 숫자들, $x_{(i)}$ 들은 순위 표본을 의미합니다.
- 귀무가설이 참 이면, 이론적으로 1이 나오도록 설계되어 있습니다.
- W 값은 0에서 1사이의 값을 갖습니다.
- W 값이 0과 가까우면 가까울 수록 정규분포와는 다르게 분포되어 있음을 뜻합니다.

- 단점: 너무 민감한 검정. 데이터의 분포가 타켓 분포와 조금만 달라도 p-value가 너무 작게 나와 귀무가설이 기각됩니다.
- 해결책: 시각화 기법과 같이 사용해서 보여주도록 합니다.
  - QQplot과 Density를 꼭 같이 사용

파이썬으로 Shapiro-Wilk 검정 하는 법

파이썬 `sp.shapiro(data)` 코드를 사용합니다.

```
x = [4.62, 4.09, 6.2, 8.24, 0.77, 5.55, 3.11,
     11.97, 2.16, 3.24, 10.91, 11.36, 0.87, 9.93, 2.9]
print(sp.shapiro(x))
```

```
## ShapiroResult(statistic=0.9116314053535461, pvalue=0.14343824982643127)
```

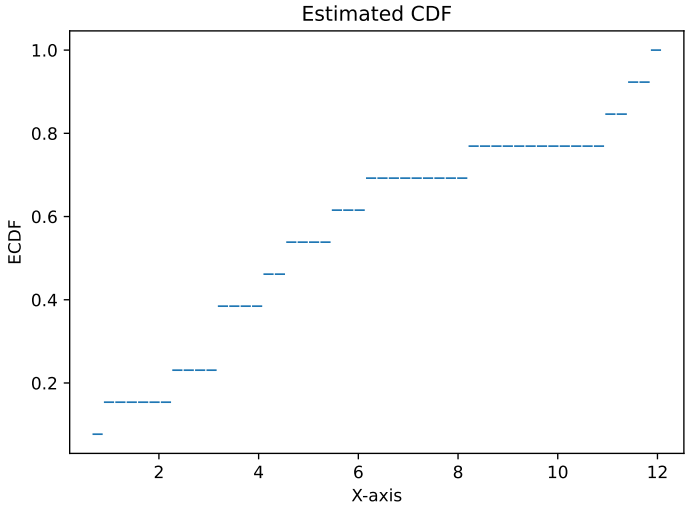
결과를 살펴보면 `p-value` 값이 상당히 크므로 유의수준 5% 하에서 귀무가설을 기각할 수 없습니다.

데이터를 사용하여 누적분포함수 그리기

데이터에 들어있는 분포의 정보를 사용하여 누적분포함수를 구하는 방법입니다. `statsmodels`에서 제공하는 `ECDF()` 함수를 사용하면 됩니다.

- 예제 데이터에 대응하는 누적분포함수 그리기

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.distributions.empirical_distribution import ECDF

data_x = [4.62, 4.09, 6.2, 8.24, 0.77, 5.55, 3.11,
          11.97, 2.16, 3.24, 10.91, 11.36, 0.87]
ecdf = ECDF(data_x)
x = np.linspace(min(data_x), max(data_x))
y = ecdf(x)
plt.plot(x,y,marker='_', linestyle='none')
plt.title("Estimated CDF")
plt.xlabel("X-axis")
plt.ylabel("ECDF")
plt.show()
```



Anderson-Darling test

이론적인 누적분포함수와 데이터에 들어있는 누적분포함수가 얼마나 비슷한 지 체크하여 검정하는 방법입니다. 이 검정의 귀무가설과 대립가설은 다음과 같습니다.

- H_0: 데이터가 target 분포를 따른다.
- H_A: 데이터가 target 분포를 따르지 않는다.

이론적인 CDF vs. Estimated CDF 거리

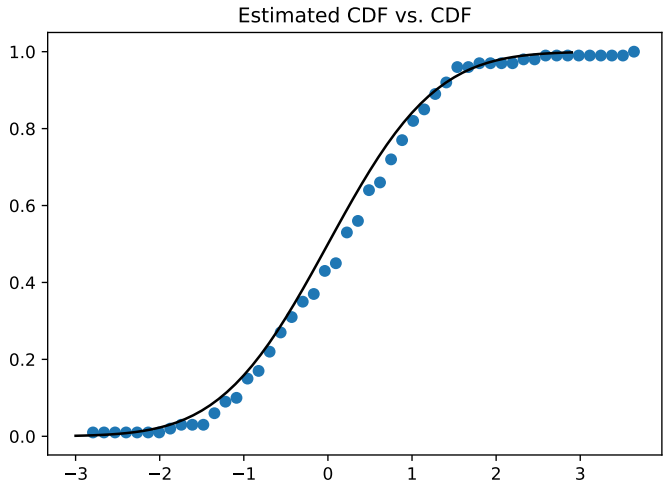
- AD 검정의 작동원리 직관: 귀무가설이 참이라면 이론적인 누적분포함수와 표본에서 구한 분포함수의 차이가 작아야지 않을까?

$$A^2 = n \int_{-\infty}^{\infty} \frac{(F_n(x) - F(x))^2}{F(x)\,(1 - F(x))}\, dF(x),$$

Anderson-Darling 검정은 정규분포 검정에도 특히 좋지만 다른 분포 검정에도 사용 가능합니다.

```
from scipy.stats import norm

np.random.seed(2021)
sample_data = norm.rvs(loc = 0, scale = 1, size = 100)

ecdf = ECDF(sample_data)
x = np.linspace(min(sample_data), max(sample_data))
y = ecdf(x)

plt.scatter(x, y)
plt.title("Estimated CDF vs. CDF")
```

```
# add Nomrmal CDF
k = np.arange(-3, 3, 0.1)
plt.plot(k, norm.cdf(k), color='k')
plt.show()
```



어떤 류의 검정을 할 수 있나? 이론적인 CDF를 알고있는 모든 분포들에 대하여 적용 가능합니다. 즉, 특정 분포를 따르는지에 대한 검정을 수행 가능합니다. 다만, 실무에서는 정규분포를 체크하는데 많이 쓰입니다.

```
from scipy.stats import anderson

ad_test = anderson(x) # p-value 꺼내는 법

print('검정통계량',ad_test[0])
```

```
## 검정통계량 0.534500239332317
```

```
print('임계값:',ad_test[1]) #p-value가 많이 다름
```

```
## 임계값: [0.538 0.613 0.736 0.858 1.021]
```

```
print('유의수준:',ad_test[2])
```

```
## 유의수준: [15.  10.   5.   2.5  1. ]
```

결과를 살펴보면, p-value 값이 Shapiro-Wilk와 마찬가지로 크게 나왔으므로, 유의수준 5하에서 H_0 기각 할 수 없습니다.

생각해보기

- AD 검정통계량의 분모에 들어있는 $F(x)(1-F(x))$ 부분은 어떤 의미일까요?
- Kolmogorov-Smirnov 검정통계량과 차이점은 무엇일까요?[2,3]

$$D_n = \sup_x |F_n(x) - F(x)|$$

연습문제

신뢰구간 구하기

다음은 한 고등학교의 3학년 학생들 중 16명을 무작위로 선별하여 몸무게를 측정한 데이터이다. 이 데이터를 이용하여 해당 고등학교 3학년 전체 남학생들의 몸무게 평균을 예측하고자 한다.

79.1, 68.8, 62.0, 74.4, 71.0, 60.6, 98.5, 86.4, 73.0, 40.8, 61.2, 68.7, 61.6, 67.7, 61.7, 66.8

단, 해당 고등학교 3학년 남학생들의 몸무게 분포는 정규분포를 따른다고 가정한다.

1) 모평균에 대한 95% 신뢰구간을 구하세요.
2) 작년 남학생 3학년 전체 분포의 표준편차는 6kg 이었다고 합니다. 이 정보를 이번 년도 남학생 분포의 표준편차로 대체하여 모평균에 대한 90% 신뢰구간을 구하세요.

신형 자동차의 에너지 소비효율 등급

슬통 자동자는 매해 출시되는 신형 자동차의 에너지 소비효율 등급을 1등급으로 유지하고 있다. 22년 개발된 신형 모델이 한국 자동차 평가원에서 설정한 에너지 소비 효율등급 1등급을 받을 수 있을지 검정하려한다. 평가원에 따르면 1등급의 기준은 평균 복합 에너지 소비효율이 16.0 이상인 경우 부여한다고 한다.

다음은 신형 자동차 15대의 복합 에너지소비효율 측정한 결과이다.

15.078, 15.752, 15.549, 15.56, 16.098, 13.277, 15.462, 16.116, 15.214, 16.93, 14.118, 14.927, 15.382, 16.709, 16.804

표본에 의하여 판단해볼때, 현대자동차의 신형 모델은 에너지 효율 1등급으로 판단할 수 있을지 판단해보시오. (유의수준 1%로 설정)

1. t 검정을 위한 가정체크를 진행하세요.
2. 검정을 위한 가설을 명확하게 서술하시오.
3. 검정통계량 계산하시오.
4. p-value을 구하세요.
5. 주어진 유의수준에 대응하는 기각역은 얼마인가요?
6. 현대자동차의 신형 모델의 평균 복합 에너지 소비효율에 대하여 95% 신뢰구간을 구해보세요.

[2] https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test

[3] Anderson-Darling test는 왜 tail에 웨이트를 주는 것인가? https://encyclopediaofmath.org/wiki/Anderson-Darling_statistic

7. 평균 복합에너지 소비효율이 15이라면, 위의 검정의 검정력을 구하세요. (단, 검정력 계산시, 모분포의 표준편차를 1라고 가정한다.)

검정력을 만족하는 표본 개수

위의 슬통 자동차 문제에서 주어진 표본들을 사용해서 신형 자동차의 평균 에너지 소비효율이 16인지 아닌지를 유의수준 5%하에서 검정을 진행하려고 한다. 단, 모분포의 표준편차를 2라고 가정한다.

1. 신형 자동차의 평균 에너지 소비효율이 15일때, 주어진 검정의 검정력을 구하세요.
2. 신형 자동차의 평균 에너지 소비효율이 15일때, 주어진 검정의 검정력을 80%로 만족시키도록 하는 표본갯수를 구해보세요. (단, 모분포의 표준편차는 모른다고 가정한다.)

IQR 과 상자그림

주어진 데이터를 사용하여 다음의 물음에 답하세요.

$$12, 15, 14, 10, 18, 20, 21, 15, 17, 19, 10, 13, 16, 22, 50, 70$$

1. 첫 번째와 세 번째 사분위수(Q1, Q3)를 계산하세요.
2. Interquartile Range (IQR)를 계산하세요.
3. 이상치를 식별하고, 이들을 데이터 세트에서 찾아내세요.
4. 데이터의 상자그림(Boxplot)을 그리세요.

제 3 장

t 검정 파헤치기

여러 상황에서의 t 검정

자료구조 파악하기

데이터가 어떻게 주어졌는지에 따라서 t 검정의 형태가 바뀝니다. 따라서 주어진 데이터가 어떠한 형태인지 파악하는 것이 중요합니다.

기본적인 자료형

t 검정의 기본적인 자료 형태는 데이터가 벡터 형태로 모든 표본이 같은 그룹으로 묶일 수 있는 형태입니다.

표 3.1: 학생들의 점수 조사표 (기본형태)

학생 ID	성적
1	9.76
2	11.10
3	10.70
4	10.72
5	11.80
6	6.15
7	10.52
8	14.83
9	13.03
10	16.46
11	10.84
12	12.45

다른 변수들이 추가 된 데이터

기본 데이터 형태에서 변형이 된 데이터의 경우 기본 자료형과 다르게 그룹을 나눌 수 있는 변수들이 들어있는 형태가 있습니다.

- 표 3.2의 경우 성별 변수로서 데이터가 두 그룹으로 나뉠 수 있습니다.

- 표 3.3의 구조는 전체 관찰 대상은 학생 6명으로 바라볼 수 있습니다.

표 3.2: 학생들의 성별 점수 조사 결과

학생 ID	성적	성별
1	9.76	female
2	11.10	female
3	10.70	female
4	10.72	female
5	11.80	female
6	6.15	female
7	10.52	female
8	14.83	male
9	13.03	male
10	16.46	male
11	10.84	male
12	12.45	male

표 3.3: 교육 프로그램 수료 후 점수 변화 조사 결과

학생 ID	성적	전/후
1	9.76	before
2	11.10	before
3	10.70	before
4	10.72	before
5	11.80	before
6	6.15	before
1	10.52	after
2	14.83	after
3	13.03	after
4	16.46	after
5	10.84	after
6	12.45	after

통계 검정 선택 시 판단 요소

t 검정의 형태를 결정 할 때, 다음 두 가지를 고려 후 판단합니다.

1. 그룹 변수가 존재하는가?
2. 표본들을 짝지을 수 있는 특정 변수가 존재하는가?

그룹 변수가 존재하는 경우, 데이터의 집단을 그룹 변수의 값에 따라서 2개로 나누어 생각할 수 있는지 판단하고, 그렇다면 2 표본 t 검정을 선택합니다. 특정 데이터의 경우 주어진 표본들이 짝을 지을 수 있는 경우가 존재하는데, 이 경우, 짝지을 수 있는 데이터들을 사용하여 데이터를 변형한 후, t 검정을 진행합니다.

검정 할 수 있는 형태들

위의 각 데이터는 다음과 같은 형태의 검정에 적합한 데이터이다.

- 학생들의 점수의 평균이 특정 값과 같은가?

$$H_0 : \mu = 10 \quad vs. \quad H_A : \mu \neq 10$$

- 남학생과 여학생 두 그룹의 평균은 같을까?

$$H_0 : \mu_M = \mu_F \quad vs. \quad H_A : \mu_M \neq \mu_F$$

- 교육 프로그램은 효과가 있었을까?

$$H_0 : \mu_{before} = \mu_{after} \quad H_A : \mu_{before} < \mu_{after}$$

t 검정을 위한 분산 가정 체크 방법

그룹이 2개 이상인 데이터를 t-검정하기 위해서는 한가지 가정을 추가적으로 체크해야 합니다.

- 그룹별 데이터가 동일한 분산을 갖는가?

F-test를 사용한 두 그룹의 등분산 체크하는 방법 (그룹 2개)

- 핵심 idea: F 검정은 추정한 분산의 비율로 두 그룹의 분산이 같은지 검정한다.

F 분포

확률변수 X가 자유도 d_1, d_2인 F 분포를 따른다고 하면, X는 다음과 같이 분수꼴로 표현되는 두 개의 확률변수Y_1, Y_2가 존재한다.

$$X = \frac{Y_1/d_1}{Y_2/d_2}$$

Y_1과 Y_2는 카이제곱분포 자유도 d_1과 d_2를 따르는 확률변수

스튜던트 정리 revisit

$$\frac{(n-1)\,S^2}{\sigma^2} \sim \chi^2_{(n-1)}$$

- 앞에서 배운 스튜던트 정리에 따르면 정규분포를 따르는 표본 크기 n로 구한 위의 통계량은 카이제곱분포 자유도 $n-1$을 따른다.
- 그룹 1 (표본 n개)과 그룹 2 (표본 m개) 데이터로 위 통계량을 구하게 된다면?

$$\frac{(n-1)\,S_1^2}{\sigma_1^2} \sim \chi^2_{(n-1)}$$

$$\frac{(m-1)\,S_2^2}{\sigma_2^2} \sim \chi^2_{(m-1)}$$

여기서 σ_1^2과 σ_2^2는 각 그룹 데이터의 모분산이다.

귀무가설이 참이라면?

두 그룹의 분산이 σ_0^2로 같다면 다음은 자유도 n, m인 F분포를 따르게 된다.

$$\frac{\frac{(n-1)S_1^2}{\sigma_0^2}/(n-1)}{\frac{(m-1)S_2^2}{\sigma_0^2}/(m-1)} = \frac{S_1^2}{S_2^2} \sim F_{n-1,m-1}$$

F 통계량 값에 따른 해석

- 두 그룹의 분산이 동일하다면, F값은 1이 나와야 함.
- 두 그룹의 분산이 다르다면, F값은 1보다 작거나 큰 값으로 나옴.

F-검정의 가설

$H_0: \sigma_A^2 = \sigma_B^2$ vs. $H_A: \sigma_A^2 \neq \sigma_B^2$

- 귀무가설: A와 B 그룹의 분산은 같다.
- 대립가설: A와 B 그룹의 분산은 같지 않다.

Python에서 F-test 하기

데이터가 데이터 프레임에 들어있는지, 벡터 형식으로 들어있는지에 대하여 구문이 달라진다.

데이터

총 관찰 데이터는 30개이며 자료 구조는 다음과 같다.

```
import pandas as pd
import numpy as np

# Set seed for reproducibility
mydata = pd.read_csv('./data/tooth_growth.csv')
mydata.head()
```

```
##     len supp  dose
## 0  11.2   VC   0.5
## 1   9.4   OJ   0.5
## 2  25.2   OJ   1.0
## 3  16.5   OJ   0.5
## 4  16.5   VC   1.0
```

Python code

```
oj = mydata[mydata['supp'] =='OJ']
s1 = oj['len'].std(ddof=1) #oj의 표본표준편차
```

```
vc = mydata[mydata['supp'] =='VC']
s2 = vc['len'].std(ddof=1) #vc의 표본표준편차

ratio_of_variances = s1**2/s2**2 # s1^2/s2^2

print('ratio_of_variances:',round(ratio_of_variances,4))
```

```
## ratio_of_variances: 0.6701
```


F 검정이 파이썬에서는 현재 지원되지 않는 상태이므로 다음과 같이 f_test() 함수를 정의하도록 하자.

```
import numpy as np
import scipy.stats as stats

def f_test(x, y):
    x = np.array(x)
    y = np.array(y)
    f = np.var(x, ddof=1) / np.var(y, ddof=1) # 검정통계량
    dfn = x.size-1
    dfd = y.size-1
    p = stats.f.cdf(f, dfn, dfd)
    p = 2*min(p, 1-p) # two sided p-value
    return f, p

# F 검정 수행하기
f_value, p_value = f_test(oj['len'], vc['len'])

print("Test statistic: ", f_value)
```

```
## Test statistic:  0.6701284713415635
```

```
print("p-value: ", p_value)
```

```
## p-value:  0.4438335690639984
```

t-검정 가정 체크 예시

앞에서 살펴 본 3가지 데이터 셋에 대하여, t 검정을 진행 할 경우, 가정 체크 과정에 대하여 알아보도록 하겠습니다.

첫번째 자료에 대한 가정 체크

주어진 데이터를 sample 변수에 저장하도록 하겠습니다.

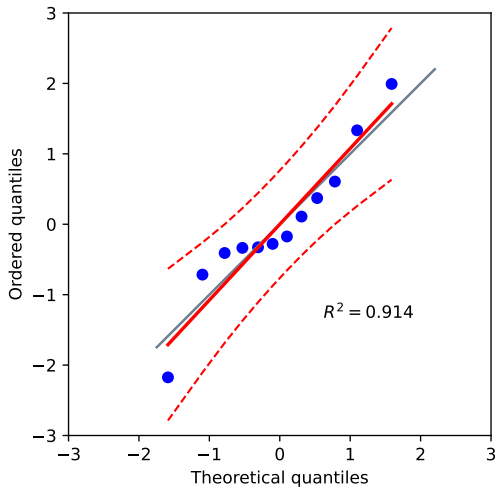
```
sample = [9.76, 11.1, 10.7, 10.72, 11.8,
          6.15, 10.52, 14.83, 13.03,
          16.46, 10.84, 12.45]
```

정규성 체크

데이터의 정규성을 체크하기 위하여, QQ plot과 Shapiro Wilk 검정을 수행합니다.

- Q-Q plot

```
import pingouin as pg
import matplotlib.pyplot as plt

ax = pg.qqplot(sample, dist='norm', confidence=0.95)
plt.ylim(-3, 3)

## (-3.0, 3.0)

plt.xlim(-3, 3)

## (-3.0, 3.0)

plt.show()
```



- Shapiro Wilk 검정

```
import scipy.stats as sp

print(sp.shapiro(sample))
```

```
## ShapiroResult(statistic=0.9387352466583252, pvalue=0.48186349868774414)
```

1표본 t-test 함수 옵션 설정

대립가설의 형태에 따라서 alternative 옵션을 조정해줘야 합니다.

- two.sided
- less
- greater

주어진 유의 수준에 따라서 conf.level 옵션을 조정해줘야 합니다.

- conf.level=0.95

```
from scipy.stats import ttest_1samp

t_statistic, p_value = ttest_1samp(sample, popmean=10, alternative='two-sided')

print("t-statistic:", t_statistic)
```

```
## t-statistic: 2.050833816777307
```

```
print("p-value:", p_value)
```

```
## p-value: 0.06488240727465693
```

두번째 자료에 대한 가정 체크

주어진 데이터를 판다스 데이터 프레임으로 만들어 봅시다.

```
import pandas as pd

sample = [9.76, 11.1, 10.7, 10.72, 11.8, 6.15, 10.52,
          14.83, 13.03, 16.46, 10.84, 12.45]
gender = ["Female"]*7 + ["Male"]*5

my_tab2 = pd.DataFrame({"score": sample, "gender": gender})
my_tab2
```

```
##     score  gender
## 0    9.76  Female
```

```
## 1   11.10  Female
## 2   10.70  Female
## 3   10.72  Female
## 4   11.80  Female
## 5    6.15  Female
## 6   10.52  Female
## 7   14.83    Male
## 8   13.03    Male
## 9   16.46    Male
## 10  10.84    Male
## 11  12.45    Male
```

두 집단 각각에 대하여 정규성을 체크하고, 등분산 가정 체크를 실시합니다.

정규성 체크 - 여학생 집단

- Q-Q plot

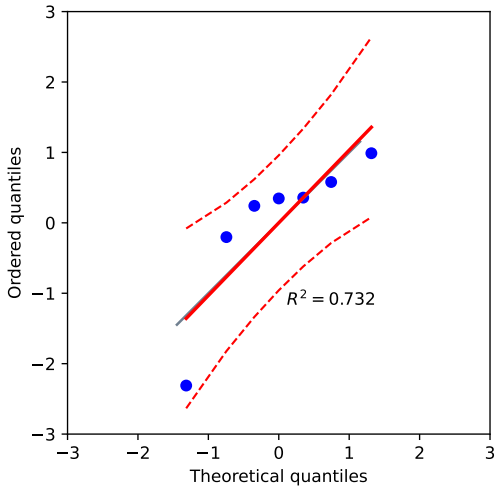
```
ax = pg.qqplot(sample[:7], dist='norm', confidence=.95)
plt.ylim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.xlim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.show()
```



- Shapiro Wilk 검정

```
result = sp.shapiro(sample[:7])
print(result)
```

```
## ShapiroResult(statistic=0.7618962526321411, pvalue=0.016863727942109108)
```

정규성 체크 - 남학생 집단

- Q-Q plot

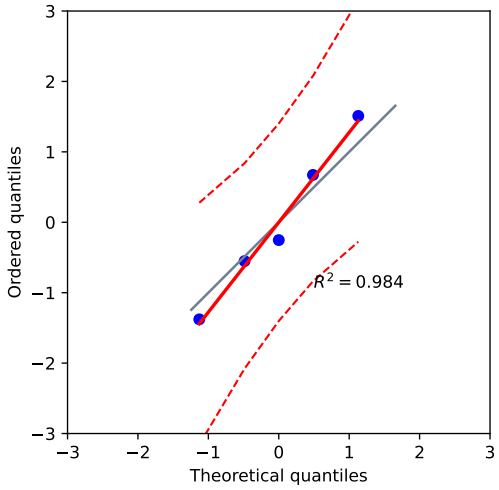
```
ax = pg.qqplot(sample[7:12], dist='norm', confidence=.95)
plt.ylim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.xlim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.show()
```



- Shapiro Wilk 검정

```
result = sp.shapiro(sample[7:12])
print(result)
```

```
## ShapiroResult(statistic=0.9813238382339478, pvalue=0.9415760636329651)
```

등분산성 체크

- F-test

```
# F 검정 수행하기
female = my_tab2[my_tab2['gender'] == 'Female']['score']
male = my_tab2[my_tab2['gender'] == 'Male']['score']

f_value, p_value = f_test(female, male)

print("Test statistic: ", f_value)
```

```
## Test statistic:  0.7230748344750079
```

```
print("p-value: ", p_value)
```

```
## p-value:  0.6870845918240329
```

두번째 자료에 대한 검정 실시

귀무가설 vs. 대립가설

귀무가설은 "두 그룹의 모평균이 동일하다" 이며, 대립가설은 "두 그룹의 모평균이 동일하지 않다" 라고 설정되어 있습니다.

- H_0: $\mu_{male} = \mu_{female}$
- H_A: $\mu_{male} \neq \mu_{female}$

alternative 옵션 설정하는 방법

- two.sided
- less
- greater

`alternative = "greater"` 로 설정한다는 의미는 첫 번째 입력값에 해당하는 그룹 (x 자리) 의 평균이 두 번째 입력값에 해당하는 그룹 (y 자리)의 평균보다 크다는 의미입니다.

Unpaired two sample t test

Unpaired two sample t test는 데이터의 정규성과 등분산성이 보장 된 경우 적용할 수 있는 t 검정입니다. `ttest_ind` 함수에 `equal_var=True`를 적용해 줍니다.

- 정규성 통과 등분산성 통과

```
from scipy.stats import ttest_ind

male = my_tab2[my_tab2['gender'] == 'Male']
```

```
female = my_tab2[my_tab2['gender'] == 'Female']

t_statistic, p_value = ttest_ind(female['score'], male['score'], equal_var=True)

print("t-statistic: ",t_statistic)
```

```
## t-statistic:  -2.9360367510416165
```

```
print("p-value: ",p_value)
```

```
## p-value:  0.01488614765791557
```

Welch (or Satterthwaite) two sample t test

Welch (or Satterthwaite) two sample t test는 데이터의 정규성은 보장되나, 등분산성이 보장 되지 않는 경우 적용할 수 있는 t 검정입니다. `ttest_ind` 함수에 `equal_var=False`을 적용해 줍니다.

- 정규성 통과 등분산성 불만족

```
from scipy.stats import ttest_ind

male = my_tab2[my_tab2['gender'] == 'Male']
female = my_tab2[my_tab2['gender'] == 'Female']

t_statistic, p_value = ttest_ind(female['score'], male['score'], equal_var=False)

print("t-statistic: ",t_statistic)
```

```
## t-statistic:  -2.850539711551644
```

```
print("p-value: ", p_value)
```

```
## p-value:  0.02199712977900471
```

무엇이 다를까?

등분산성 가정을 만족하는지 만족하지 않는지에 따라서 t 검정의 검정통계량 값을 계산 할 때, 분산을 추정하는 방법과 검정통계량이 따르는 t분포의 자유도 값이 달라집니다. 하지만, 검정통계량의 형태는 동일합니다.

https://academic.oup.com/beheco/article/17/4/688/215960?login=false

Unpaired two sample t test

- 검정 통계량

$$t = \frac{(\overline{X}_1 - \overline{X}_2) - (\mu_1 - \mu_2)_0}{s_p^2\sqrt{\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim t_{n_1+n_2-2}$$

  여기서 $\overline{x}$은 표본평균, n은 표본크기를 나타내고, $(\mu_1 - \mu_2)_0$은 귀무가설에서 주장하는 두 집단 모평균의 차이를 나타낸다. 보통은 0으로 설정 됨.

- 검정 통계량의 분포는 자유도가 $n_1 + n_2 - 2$인 t 분포를 따른다.
- s_p^2는 두 모집단의 분산을 추정하기 위한 통계량으로 쓰임.

$$s_p^2 = \frac{(n_1 - 1)\, S_1^2 + (n_2 - 1)\, S_2^2}{n_1 + n_2 - 2}$$

  n_1, s_1, n_2, s_2는 각각 그룹 1과 2의 표본 크기와 표본 표준편차를 나타낸다.

Welch's t-test statistic

- 검정 통계량

$$t = \frac{(\overline{X}_1 - \overline{X}_2) - (\mu_1 - \mu_2)_0}{\sqrt{\frac{S_2^2}{n_1} + \frac{S_2^2}{n_2}}} \sim t_\nu$$

  자유도 ν는 다음 Satterthwaite's approximation을 사용하여 계산한다.

$$\nu \approx \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{s_1^4}{n_1^2(n_1-1)} + \frac{s_2^4}{n_2^2(n_2-1)}}$$

세번째 자료에 대한 가설 체크

세번째 자료는 데이터가 짝지어 질 수 있는 데이터 (대응 표본) 였습니다. 주어진 자료를 입력 받은 후, 대응되는 표본들끼리의 차을 이용하여 하나의 그룹 데이터로 변형시켜 보겠습니다.

```
import pandas as pd

tab3 = pd.read_csv('./data/tab3.csv')
tab3_data = tab3.pivot_table(index='id',columns='group',values='score')

tab3_data['score_diff'] = tab3_data['after'] - tab3_data['before']
test3_data = tab3_data[['score_diff']]

test3_data
```

```
## group  score_diff
## id
```

```
## 1            0.76
## 2            3.73
## 3            2.33
## 4            5.74
## 5           -0.96
## 6            6.30
```

주어진 표본 크기가 반절로 줄어들었지만, 변형 결과 첫번째 유형의 데이터와 형태가 동일해 졌습니다. 따라서, 1표본 t-test 검정을 위한 가정 체크를 실시한다.

- Q-Q plot

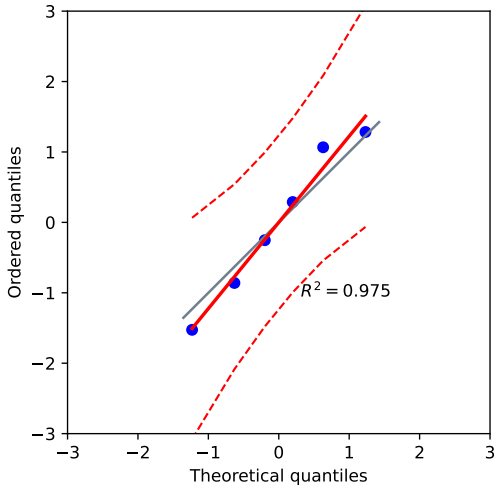
```
ax = pg.qqplot(test3_data, dist='norm',confidence=.95)
plt.ylim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.xlim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.show()
```



- Shapiro test

```
result = sp.shapiro(test3_data)
print(result)
```

```
## ShapiroResult(statistic=0.9559008479118347, pvalue=0.7876578569412231)
```

세번째 자료에 대한 검정 실시

- 교육 프로그램은 효과가 있었을까?

$$H_0 : \mu_{before} = \mu_{after} \quad vs. \quad H_A : \mu_{before} < \mu_{after}$$
$$H_0 : \mu_d \leq 0 \quad vs. \quad H_A : \mu_d > 0$$

- $\mu_d := \mu_{after} - \mu_{before}$
- 새로운 모수에 대한 표본 평균을 관찰하는 것으로 생각할 수 있음.

귀무가설 vs. 대립가설

귀무가설은 "교육 프로그램의 점수 향상 효과가 없음." 이며, 대립가설은 "교육 프로그램의 점수 향상 효과가 있음." 라고 설정.

- H_0: $\mu_d = 0$
- H_A: $\mu_d > 0$

Paired t test

- 검정통계량

$$t = \frac{\overline{D} - \mu_d^0}{S_d/\sqrt{n_d}} \sim t_{n_d - 1}$$

- 표본 크기 n_d가 원래 자료의 반으로 줄어든다는 것에 주의하자.
- 이후 내용은 1표본 t 검정과 동일.
- 데이터 프레임 형식일 때 1표본 t 검정하는 방법

```
from scipy.stats import ttest_1samp

t_statistic, p_value = ttest_1samp(test3_data, 0, alternative='greater')

print("t-statistic:", t_statistic)
```

```
## t-statistic: [2.58116143]
```

```
print("p-value:", p_value)
```

```
## p-value: [0.02468128]
```

앞에서는 두 그룹의 값의 차를 사용하여 `score_diff`를 계산하여 1 표본 t-test를 사용하였다. 하지만, `ttest_rel()`를 사용하면, 두 그룹의 값을 따로따로 입력하여 paired t test를 수행할 수 있다.

```
t_statistic, p_value = stats.ttest_rel(tab3_data['after'],
                                       tab3_data['before'],
                                       alternative='greater')
print("t-statistic:", t_statistic)
```

```
## t-statistic: 2.5811614301011883
```

```
print("p-value:", p_value)
```

```
## p-value: 0.02468128345546597
```

주의사항!

여기서 주의할 점은, alternative 옵션을 첫번째 그룹을 기준으로 대립가설의 부등호를 입력한다는 것이다. 따라서, 다음과 같이 순서를 바꿔서 입력하면 엉뚱한 p-value값이 도출 됨에 주의합니다.

```
t_statistic, p_value = stats.ttest_rel(tab3_data['before'],
                                       tab3_data['after'],
                                       alternative='greater')
print("t-statistic:", t_statistic)
```

```
## t-statistic: -2.5811614301011883
```

```
print("p-value:", p_value)
```

```
## p-value: 0.9753187165445341
```

3.1 참고 내용

데이터가 특정 분산값을 따르는지 체크하는 방법 (23회 기출문제)

정규분포를 따르는 확률변수에서 관찰한 표본들의 표본표준편차 S^2 은 다음과 같이 카이제곱분포를 따름.

- 앞에서 배운 스튜던트 정리와 관련이 있음.

$$\frac{(n-1)\,S^2}{\sigma^2} \sim \chi^2_{(n-1)}$$

귀무가설 vs. 대립가설

- H_0: 모분산 σ^2 이 σ_0^2 와 같다.
- H_A: 모분산 σ^2 이 σ_0^2 같지 않다.

검정순서

- 데이터가 정규성을 따르는지 체크
- 검정통계량 값을 계산해서 대응하는 카이제곱 분포에서 관찰할 수 있는 확률을 구함.

Python에서 구현하기

표 3.1 의 성적 데이터의 표준편차가 3보다 큰 지 검정하고자 한다면 다음과 같이 할 수 있다.

```
import numpy as np
from scipy.stats import chi2

sample_data = np.array([9.76, 11.10, 10.70, 10.72,
                        11.80, 6.15, 10.52, 14.83,
                        13.03, 16.46, 10.84, 12.45])

s = np.std(sample_data, ddof=1)
sigma = 3
n = 12
chi = ((n-1)*(s**2)) / sigma**2
p_value = 1 - chi2.cdf(chi, df=n-1)

print('Chi-Squared:',chi)
```

```
## Chi-Squared: 8.16306666666667
```

```
print('p_value:',p_value)
```

```
## p_value: 0.6986263615554091
```

연습문제

신약 효과 분석

새로 제안된 혈압약에 대한 효과 분석을 위하여 무작위로 배정된 두 그룹에 대한 혈압 측정 데이터입니다. Treated 그룹의 경우 혈압약을 일정기간 복용하였으며, Control 그룹은 평상시 활동을 그대로 유지하였습니다. 표 3.4는 두 그룹의 혈압을 측정한 데이터입니다.

혈압약의 사용이 혈압을 떨어뜨리는 효과가 있는지 유의수준 5%하에서 검정해보세요.

- 귀무가설 vs. 대립가설
- 가정 체크 및 검정 방법 설정

표 3.4: 혈압약 효과 측정 데이터

ID	Score	Group
1	3.81	control
2	4.47	control
3	4.81	control
4	4.79	control
5	4.25	control
6	3.93	treated
7	4.26	treated
8	3.74	treated
9	3.61	treated
10	3.93	treated
11	4.36	treated
12	3.93	treated
13	3.89	treated
14	4.03	treated
15	3.85	treated
16	4.06	treated

고양이 소변의 효과

고양의 소변이 식물의 성장을 저해한다는 명제를 확인하기 위하여 무작위로 선정된 두 식물 그룹을 가지고 실험을 진행하였습니다.

Treated 그룹의 경우 고양이 소변을 체취하여 일정 간격으로 뿌려주었으며, 이외 다른 조건은 Treated, Control 그룹 모두 동일하게 유지하였습니다. 표 3.5는 한달 후 두 그룹의 성장 높이 데이터를 기록한 것입니다.

고양의 소변이 식물의 성장을 저해한다는 것에 대한 검정을 유의수준 5%하에서 수행해보세요.

표 3.5: 식물 성장 측정 데이터

ID	Score	Group
1	45	control
2	87	control
3	123	control
4	120	control
5	70	control
6	51	treated
7	71	treated
8	42	treated
9	37	treated
10	51	treated
11	78	treated
12	51	treated
13	49	treated
14	56	treated
15	47	treated
16	58	treated

제 4 장

비모수 검정 친해지기

4.1 비모수 검정 이해를 위한 준비운동

이제까지의 검정들은 모두 스튜던트 정리에 기반한 모수 검정들이었습니다. 하지만 이 정리는 표본들이 정규분포에서 뽑혀져 나온다는 큰 가정에 기반하고 있습니다. 따라서 검정을 하기 전 가정 체크시 정규성을 위반한 경우는 적용할 수 없습니다. 데이터가 주어졌을때, 다음과 같은 가정 체크에 따라서 모수 검정과 비모수 검정 중 알맞은 검정을 선택하도록 합시다.

두 그룹 이하 평균 비교 검정의 고려 순서도

집단 1개

- 정규성 체크
  - Yes: t-test 시행
  - No: 대칭성 확인
    - Yes: Wilcoxon signed-rank test
    - No: Sign test

집단 2개

- 정규성 체크
  - Yes
    - 등분산 체크 (F-test)
      - Yes: 2 sample t-test 시행
      - No: Welch's t-test
  - No:
    - 등분산 체크 (Levene test)
      - Yes: Mann-Whitney-Wilcoxon test (= Wilcoxon rank-sum test)
      - No: Brunner-Munzel test

비모수 검정의 장점

일반적으로 비모수 검정은 다음과 같은 장점을 가지고 있습니다.

1. 정규성 가정이 필요하지 않습니다.

정규성 가정이 필요하지 않다는 것은 검정에 필요한 가정이 적어져, 검정을 적용할 수 있는 경우가 많다는 의미입니다.

2. 이상치에 민감하지 않습니다.

검정 통계량 값이 데이터에 따라 변동하는 것이 적어서 이상치가 있더라도 검정 결과에 큰 영향을 주지 않는다는 의미입니다.

3. 통계량이 직관적인 경우가 많습니다.

해당 검정 통계량을 이해하고 사용하기 쉽다는 의미입니다. 이로 인해 통계적 분석 결과를 직관적으로 이해하고 해석할 수 있습니다.

비모수 검정을 매번 사용하지 않는 이유

비모수 검정 방법은 자료가 정규분포를 따를 때에 비해 모수 검정 방법보다 파워(검정력)가 약할 수 있습니다.

- 검정력 (Power) - 귀무가설이 참이 아닐 때, 귀무가설을 기각하는 확률

비모수 검정시 유의 사항

앞에서 살펴본 검정들은 t 검정 관련 함수를 사용했던 것처럼, 이번에는 상황에 맞는 비모수 검정 함수를 사용합니다. 비모수 검정과 관련한 함수를 사용할 때 다음 사항들을 꼭 인지하고 있어야 합니다.

- 비모수 검정은 모수의 중심점을 평균이 아닌 **중앙값**으로 설정하고, 중앙값에 대한 검정을 수행
- 모 중앙값을 나타내는 그리스 문자로 η사용[1]: 귀무 가설과 대립가설 설정 시 꼭 변경

Levene 검정을 이용한 등분산 가정 체크

비모수 검정에서 등분산 가정을 체크할 때, Levene 검정을 일반적으로 많이 사용합니다. Levene 검정은 유료 통계프로그램 (SPSS etc.) 에서 분산 비교 시 사용하는 일반적인 검정 방법입니다.

F 검정과의 비교

이전 챕터에서 배운 F 검정의 경우, 스튜던트 정리의 2번째 사실인 카이제곱분포 확률변수를 분모꼴로 나타낸 검정통계량을 사용했었습니다. 따라서 데이터가 정규분포에서 뽑혀져 나왔다는 가정이 필요했습니다. Levene 검정은 F 검정과 비교하여 다음과 같은 특징이 있습니다.

- 2개 이상의 그룹에도 적용 가능 (F 검정은 2개 그룹에만 적용 가능)
- F-test는 태생이 정규분포와 궁합이 좋은 검정
- 이상치에 대하여 좀 더 robust 한 검정

[1] 해당 문제는 위키피디아의 선형계획법 페이지에서 가져옴.

귀무가설 vs. 대립가설

검정의 귀무가설과 대립가설은 다음과 같습니다.

- 귀무가설: "모든 그룹이 동일한 분산을 갖는다."
  - H_0: $\sigma_1^2 = \sigma_2^2 = ... = \sigma_k^2$
- 대립가설: "하나라도 분산이 다른 그룹이 존재한다."
  - H_A: $\sigma_i^2 \neq \sigma_j^2$ at least one pair

검정통계량

Levene 검정의 검정통계량은 다음과 같습니다.

$$W = \frac{(N-k)}{(k-1)} \cdot \frac{\sum_{i=1}^{k} N_i (Z_{i\cdot} - Z_{\cdot\cdot})^2}{\sum_{i=1}^{k} \sum_{j=1}^{N_i} (Z_{ij} - Z_{i\cdot})^2},$$

이 검정통계량의 특징은 원 데이터를 이용하는 것이 아니라 transformed 된 데이터들 Z_{ij}들을 사용하여 통계량을 구함

- k 그룹이 존재
- Z_{ij}를 구하는 방법을 3가지로 선택할 수 있음
  - $Z_{ij} = |Y_{ij} - \bar{Y}_{i\cdot}|$: 각 그룹의 평균에서의 deviation
  - $Z_{ij} = |Y_{ij} - \tilde{Y}_{i\cdot}|$: 각 그룹의 중앙값에서의 deviation
  - $Z_{ij} = |Y_{ij} - \bar{Y}'_{i\cdot}|$: 각 그룹의 10% 절단 평균에서의 deviation
- mean: 대칭, median: 치우침, trimmed mean: 두터운 꼬리

Python에서 Levene 검정 수행하기

Python에서 Levene 검정을 수행하는 방법은 다음과 같다.

예제 데이터 불러오기 총 관찰 데이터는 30개이며 자료 구조는 다음과 같다.

```
#Levene's test 관련 패키지 불러오기
from scipy.stats import levene
import pandas as pd
import numpy as np

mydata = pd.read_csv('./data/tooth_growth.csv')
mydata.head()
```

```
##     len supp  dose
## 0  11.2   VC   0.5
## 1   9.4   OJ   0.5
```

```
## 2  25.2   OJ   1.0
## 3  16.5   OJ   0.5
## 4  16.5   VC   1.0
```

levene() 함수 사용하기

두 그룹으로 분리 후, levene() 함수를 사용하여 수행합니다.

```
a = mydata[mydata['supp'] == 'OJ']['len']
b = mydata[mydata['supp'] == 'VC']['len']

levene(b, a, center='mean')
```

```
## LeveneResult(statistic=0.06815502357191881, pvalue=0.7959528904404206)
```

- Option 선택
  - center = "median" (기본) 혹은 "mean" 혹은 "trim"
  - proportiontocut = 0.05 (trim 선택 시 사용)

4.2 비모수 검정 방법

Wilcoxon signed rank test (1 표본 검정)

- 가정 1: 데이터가 연속 확률분포를 따른다. (체크 불필요)
- 가정 2: 데이터가 대칭 분포를 따른다. (체크 필요, 그래프)

One Sample 예제 데이터

그룹이 1개의 동일 그룹으로 이루어진 데이터에 대응하는 검정이다. 3장에서 사용한 학생들의 성별에 따른 점수 조사 데이터를 다시 사용하자.

표 4.1: 학생들의 점수 조사표 (기본형태)

학생 ID	성적
1	9.76
2	11.10
3	10.70
4	10.72
5	11.80
6	6.15
7	10.52
8	14.83
9	13.03
10	16.46
11	10.84
12	12.45

귀무가설 vs. 대립가설

앞에서 배웠던 t 검정의 귀무가설과 대립가설 종류와 동일합니다.

- $\eta = \eta_0$ vs. $\eta \neq \eta_0$
- $\eta \geq \eta_0$ vs. $\eta < \eta_0$
- $\eta \leq \eta_0$ vs. $\eta > \eta_0$

자료 입력

```
import numpy as np
sample = np.array([9.76, 11.1, 10.7, 10.72, 11.8, 6.15, 10.52,
          14.83, 13.03, 16.46, 10.84, 12.45])
```


검정 통계량

모중앙값 η_0에 대한 검정 통계량 W^+은 다음과 같습니다.

$$W^+ = \sum_{i=1}^{n} \psi\left(X_i - \eta_0\right) R_i$$

여기서 $\psi(x)$ 함수는 입력값 x가 0보다 크면 1, 0보다 작거나 같은 경우 0을 반환합니다.

- 코드로 직접 구해보기

```
import numpy as np
from scipy.stats import rankdata

sample_diff = abs(np.array(sample) - 10)
sample_rank = rankdata(sample_diff)

sample_sign = np.sign(np.array(sample) - 10)

sum(sample_rank[sample_sign > 0])
```

```
## 67.0
```

핵심 아이디어

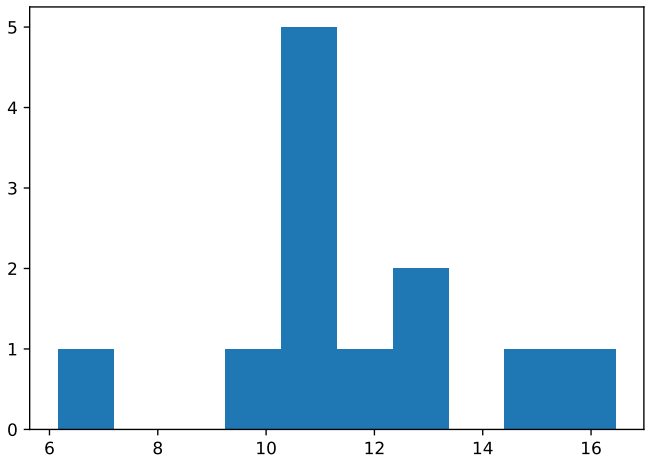
귀무가설이 참이라면 η_0를 중심으로 데이터가 균형을 이루며 퍼져있어야 할 것이다. 따라서 이상적인 W^+인 값은 전체 순위합의 절반인 $n(n+1)/4$가 되어야 한다. 이 값에서 멀어질 수록 (크거나 작게 된다면) 귀무가설을 기각하는 근거가 된다.

- W^+ 값의 이론적인 최대값은 전체 순위합인 $n(n+1)/2$가 된다.

가정체크 방법

검정의 가정이 데이터의 분포가 대칭이룬다는 가정이 있으므로 확인을 해봐야 한다. 하지만 분포의 대칭을 검정하는 방법은 현재 정확하게 정립된 것이 없는 것으로 보이므로 주어진 표본의 히스토그램을 그려서 확인하도록 하자.

```
import matplotlib.pyplot as plt
plt.hist(sample);
plt.show()
```



Python에서 검정하기

`scipy.stats`의 `wilcoxon()` 함수를 사용합니다. 함수에서 사용할 수 있는 옵션들은 3장에서 배웠던 t 검정 함수와 똑같으므로 설명을 생략한다. 다만, 주의점은 데이터의 입력을 귀무가설 하에서 주어진 η_0 값에서 빼준 데이터를 입력해줘야 한다는 것입니다.

```
from scipy.stats import wilcoxon

eta_0=10
statistics, pvalue = wilcoxon(sample-eta_0, alternative="two-sided")

print("Test statistic: ", statistics)
```

```
## Test statistic:  11.0
```

```
print("p-value: ", pvalue)
```

```
## p-value:  0.02685546875
```

5% 유의수준 하에서 p-value 값이 작으므로, 귀무가설을 기각합니다. 해석할 때 모평균이라는 단어 대신 **분포의 중앙**이라는 단어를 사용해야 함에 주의하세요!

참고사항 위의 파이썬 코드와 R에서의 wilcox.test()함수의 검정통계량 값이 다릅니다. 이유는 R에서는 양수인 RANK의 합을 반환하는 반면, 파이썬에서는 양수 쪽, 음수 쪽 SUM(RANK) 중 작은 것을 반환하기 때문입니다.

```
a = np.arange(start=1, stop=13)
print("R 결과와 같게끔 출력, Test statistic: ", sum(a)-statistics)
```

```
## R 결과와 같게끔 출력, Test statistic:  67.0
```

- R에서 결과 확인

```
sample <- c(9.76, 11.1, 10.7, 10.72, 11.8, 6.15,
            10.52, 14.83, 13.03, 16.46, 10.84, 12.45)
wilcox.test(sample, mu =10, alternative ="two.sided")
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  sample
## V = 67, p-value = 0.02686
## alternative hypothesis: true location is not equal to 10
```

Mann-Whitney-Wilcoxon test (2 표본 검정)

2 표본 독립 그룹을 가정

2개의 독립적인 그룹에서 추출된 데이터에 대응하는 검정 방법입니다. 또한, 각 그룹의 분산이 동일하다는 가정을 하고 있는 검정입니다. 따라서, Levene 검정을 사용하여 각 그룹의 등분산성을 체크한 후, 진행하도록 합시다.

예제 데이터

3장에서 사용된 학생들의 성별 점수 조사 데이터를 다시 사용합니다.

자료 입력

```
import pandas as pd

sample = [9.76, 11.1, 10.7, 10.72, 11.8, 6.15, 10.52,
          14.83, 13.03, 16.46, 10.84, 12.45]
gender = ['female']*7 + ['male']*5

data_wilcoxon = pd.DataFrame({'id': range(1,13),
                              'score': sample,
```

표 4.2: 학생들의 성별 점수 조사 결과

학생 ID	성적	성별
1	9.76	female
2	11.10	female
3	10.70	female
4	10.72	female
5	11.80	female
6	6.15	female
7	10.52	female
8	14.83	male
9	13.03	male
10	16.46	male
11	10.84	male
12	12.45	male

```
                                'gender':gender})
data_wilcoxon['gender'] = pd.Categorical(data_wilcoxon['gender'],
                                         categories=['male','female'])
data_wilcoxon
```

```
##     id  score  gender
## 0    1   9.76  female
## 1    2  11.10  female
## 2    3  10.70  female
## 3    4  10.72  female
## 4    5  11.80  female
## 5    6   6.15  female
## 6    7  10.52  female
## 7    8  14.83    male
## 8    9  13.03    male
## 9   10  16.46    male
## 10  11  10.84    male
## 11  12  12.45    male
```

귀무가설 vs. 대립가설

- $\eta_1 = \eta_2$ vs. $\eta_1 \neq \eta_2$
- $\eta_1 = \eta_2$ vs. $\eta_1 < \eta_2$
- $\eta_1 = \eta_2$ vs. $\eta_1 > \eta_2$

검정 통계량

검정 통계량 U는 다음과 같이 정의된다.

$$U = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \psi\left(Y_j - X_i\right)$$

- 코드로 직접 구해보기

```
import itertools
sample1 = sample[:7]
sample2 = sample[7:]
possible_comb = list(itertools.product(sample1, sample2))
U = sum([x[0] < x[1] for x in possible_comb])

print(U)
```

```
## 33
```

Wilcoxon rank sum 통계량 MWW 통계량 U는 X_i와 Y_j의 혼합표본에서 Y들의 순위 R_j들의 합을 계산한 윌콕슨 순위합 (Wilcoxon rank sum) 통계량 $W = \sum R_j$와 다음과 같은 관계가 있습니다.

$$W = U + \sum_{k=1}^{n} k$$

따라서 Mann-Whitney-Wilcoxon 검정과 Wilcoxon rank sum 검정의 동치 관계에 있다고 할 수 있습니다.

- 예제 데이터에서 W 통계량값과 U 통계량 관계 확인하기

```
U + sum(range(1,6))
```

```
## 48
```

```
from scipy.stats import rankdata

rank_sample = rankdata(sample)
sum(rank_sample[7:12])
```

```
## 48.0
```

MWW 검정의 핵심 아이디어

두 분포의 중앙이 같다면 각 그룹에서 하나씩을 뽑아 크기 비교를 하면 그중 절반은 한 그룹에서 뽑은 표본이 더 커야합니다. 즉, 귀무가설이 참이라면 다음의 값이 0.5와 비슷하게 나와야 합니다.

```
U / len(possible_comb)
```

```
## 0.9428571428571428
```

이 값이 0.5에서 멀어질 수록, 귀무가설을 기각하게 되는 근거가 됩니다.

Python에서 검정하기

2표본의 경우 mannwhitneyu 모듈의 wilcoxon() 함수를 사용하여 수행합니다.

```
from scipy.stats import mannwhitneyu

male = data_wilcoxon[data_wilcoxon['gender'] == 'male']['score']
female = data_wilcoxon[data_wilcoxon['gender'] == 'female']['score']

stat, pvalue = mannwhitneyu(male, female)

print("stat: ", stat.round(3))
```

```
## stat:  33.0
```

```
print("p-value: ", pvalue.round(3))
```

```
## p-value:  0.01
```

유의수준 5% 하에서 p-value 값이 작으므로 귀무가설을 기각합니다.

등분산 가정이 깨졌을 경우

Mann-Whitney-Wilcoxon의 등분산 가정을 표본이 만족하지 않는 경우, 일반화 MWW 검정이라 불리는 Brunner Munzel 검정을 적용할 수 있습니다. 검정 관련 함수는 scipy 패키지의 brunnermunzel()을 사용합니다.

```
from scipy.stats.mstats import brunnermunzel

female = data_wilcoxon[data_wilcoxon['gender'] == 'female']
male = data_wilcoxon[data_wilcoxon['gender'] == 'male']
brunnermunzel(male['score'],female['score'], alternative='two-sided')
```

```
## BrunnerMunzelResult(statistic=-6.511302390730245, pvalue=0.00029416889680564974)
```

Two-sample paired 예제 데이터

특정 변수를 사용하여 1개 그룹으로 변환 가능한 짝이 존재하는 (paired) 데이터에 대응하는 검정 방법입니다.

표 4.3: 교육 프로그램 수료 후 점수 변화 조사 결과

학생 ID	성적	전/후
1	9.76	before
2	11.10	before
3	10.70	before
4	10.72	before
5	11.80	before
6	6.15	before
1	10.52	after
2	14.83	after
3	13.03	after
4	16.46	after
5	10.84	after
6	12.45	after

자료 입력 및 변환

데이터를 입력하고, 짝이 있는 경우, 짝을 이용해서 데이터를 변환합니다.

- 자료입력

```
import pandas as pd

id = [1, 2, 3, 4, 5, 6]
before_after = ['before']*6 + ['after']*6

tab3 = pd.DataFrame({'id': id*2, 'score': sample,
                     'group':before_after})

tab3['group'] = pd.Categorical(tab3['group'],
                               categories=['before','after'])
tab3
```

```
##     id  score   group
## 0    1   9.76  before
## 1    2  11.10  before
## 2    3  10.70  before
## 3    4  10.72  before
## 4    5  11.80  before
## 5    6   6.15  before
## 6    1  10.52   after
## 7    2  14.83   after
## 8    3  13.03   after
## 9    4  16.46   after
```

```
## 10   5  10.84   after
## 11   6  12.45   after
```

- 자료변환

```
test3_data = tab3.pivot(index='id', columns='group', values='score')
test3_data['score_diff'] = test3_data['after'] - test3_data['before']
test3_data['score_diff']
```

```
## id
## 1    0.76
## 2    3.73
## 3    2.33
## 4    5.74
## 5   -0.96
## 6    6.30
## Name: score_diff, dtype: float64
```

귀무가설 vs. 대립가설

짝이 지어진 그룹간의 차이를 $\Delta := \eta_{pair_1} - \eta_{pair_2}$ 로 정의합니다.

- $\Delta = 0$ vs. $\Delta \neq 0$
- $\Delta = 0$ vs. $\Delta < 0$
- $\Delta = 0$ vs. $\Delta > 0$

검정 통계량

변환 된 데이터는 One sample 경우와 똑같다는 것을 알 수 있습니다.

```
sample_sign = np.sign(test3_data['score_diff'])
sum(rankdata(abs(test3_data['score_diff']))[sample_sign > 0])
```

```
## 19.0
```

Python에서 검정하기

`wilcoxon()` 함수에 대한 설명은 앞에서 설명한 One Sample 경우와 같으므로 설명은 생략합니다.

```
from scipy.stats import wilcoxon

wilcoxon(test3_data['score_diff'], alternative = 'greater')
```

```
## WilcoxonResult(statistic=19.0, pvalue=0.046875)
```

4.3 부호 검정 (Sign test)

앞에서는 부호(sign)와 순위(rank)를 사용하여 검정 통계량을 만들었습니다. Signed test 부호만을 가지고 검정하는 부호 검정 방법에 대하여 알아봅니다. 부호 검정은 부호만을 이용한 검정이므로 적용할 수 있는 범위는 넓지만, 그만큼 검정력은 떨어지는 검정입니다. 따라서, 부호 검정은 제일 마지막 선택지로 남겨두도록 하겠습니다.

귀무가설 vs. 대립가설

- $\eta = \eta_0$ vs. $\eta \neq \eta_0$
- $\eta \geq \eta_0$ vs. $\eta < \eta_0$
- $\eta \leq \eta_0$ vs. $\eta > \eta_0$

검정 통계량

모중앙값 η_0에 대한 검정 통계량 B은 다음과 같다.

$$B = \sum_{i=1}^{n} \psi\left(X_i - \eta_0\right)$$

윌콕슨 부호순위 검정 통계량에서 순위부분을 빼고 더한 값이 된다.

핵심 아이디어

만약 귀무가설이 참이라면 η_0를 분포의 중앙으로 데이터가 고르게 분포되어야 한다. 따라서 검정통계량 값 B는 이항분포 $B(n, 0.5)$를 따르게 된다. n은 표본의 크기를 의미한다.

- 이항분포 $B(n, 0.5)$에서 계산된 검정통계량 값을 관찰 할 확률로 p-value를 계산한다.
- 코드로 직접 구해보기

```
sample_sign = np.sign(np.array(sample) - 10)
sum(sample_sign > 0)
```

```
## 10
```

Python에서 검정하기

statsmodels 패키지에서 제공하는 sign_test()를 이용하도록 한다.

```
from statsmodels.stats.descriptivestats import sign_test

sample = np.array([9.76, 11.1, 10.7, 10.72, 11.8, 6.15, 10.52,
                   14.83, 13.03, 16.46, 10.84, 12.45])
sign_test(sample, mu0=10)
```

```
## (4.0, 0.03857421875)
```

다만, 이 경우 검정통계량 값이 다르게 나오는데, `sign_test()`의 반환값이 μ_0보다 큰 표본의 개수 $N(+)$와 작은 표본의 개수$N(-)$ 차를 2로 나눈 값을 반환하기 때문이다. 이는 일반적인 검정통계량은 아니며, 계산된 p-value는 동일하다.

유의수준 5% 하에서 p-value 값, 0.038이 작으므로 귀무가설을 기각한다.

p-value 값의 이해

주어진 표본의 갯수는 12이고, 귀무가설이 참인 경우 검정통계량 값 B는 이항분포 $B(12, 0.5)$를 따르게 되므로 p-value는 다음과 같이 계산할 수 있다.

```
from scipy.stats import binom
(1 - binom.cdf(9, 12, 0.5)) * 2
```

```
## 0.03857421875
```

이산형 분포의 경우 검정통계량 값보다 같거나 큰 경우를 계산함에 유의하자.

부호 검정에서의 표본 크기 계산하기

- 만약 귀무가설에서 고려하는 η_0의 값과 같은 표본이 존재하는 경우 **표본에서 제외**시킨다.
- 즉, 새로운 표본 갯수 n'는 전체 표본 n에서 η_0인 표본 갯수 k를 빼서 계산한다.

$$n' = n - k$$

연습문제

신제품 촉매제

슬통 회사에서는 이번에 출시한 새로운 촉매제의 효능을 검증하고 싶어한다. 신제품 촉매재는 기존 공정에서 사용되는 화학반응 속도를 혁신적으로 줄여주는 기능이 탑재되어 있다고 한다.

회사 제품 검증 부서에서는 기존 공정의 화학 반응속도와 촉매제를 넣은 후의 반응 속도를 측정하여 데이터를 만들었다.

1. 유의수준 5%하에서 신제품 촉매제가 기존의 화학 공정을 단축시킨다고 할 수 있는지에 대하여 검정하시오.
2. 촉매제로 인한 단축된 공정 시간에 대하여 90% 신뢰구간을 구하시오.

심장 질환 약 효능

슬통제약의 신약이 심장 질환 환자의 혈압을 낮출 수 있는지 검증하려고 한다. 표본으로 15명의 환자가 선택되었으며, 약을 복용하기 전과 복용한 후의 혈압을 측정하였다.

- 복용전: 130, 125, 120, 135, 140, 136, 129, 145, 150, 135, 128, 140, 139, 130, 145
- 복용후: 125, 120, 115, 130, 135, 134, 128, 140, 145, 134, 127, 140, 138, 129, 142

표 4.4: 촉매제 성능 비교 데이터

ID	Time	Treat
1	2.17	before
2	0.86	before
3	0.91	before
4	3.11	before
5	1.29	before
6	1.25	before
7	0.76	before
8	2.98	before
9	1.21	before
10	2.23	before
11	0.67	before
12	1.22	before
13	1.23	before
14	1.21	add_catalyst
15	1.71	add_catalyst
16	1.80	add_catalyst
17	1.41	add_catalyst
18	1.01	add_catalyst
19	0.82	add_catalyst
20	1.03	add_catalyst
21	2.03	add_catalyst
22	0.65	add_catalyst
23	1.01	add_catalyst
24	0.45	add_catalyst
25	0.98	add_catalyst
26	1.04	add_catalyst

약이 혈압을 실제로 낮추는 것인지 검증하기 위하여 부호 검정을 실시하라. 유의 수준은 0.05로 하고, 양측 검정을 수행하시오.

제 5 장

카이제곱 검정 친해지기

5.1 카이제곱 분포에 대하여

"확률변수 X가 자유도가 ν인 카이제곱분포를 따른다."라는 것은 다음과 같이 표현합니다.

$$X \sim \chi^2(\nu)$$

카이제곱 분포는 자유도 ν만 주어지면 확률밀도함수를 그릴 수 있습니다.

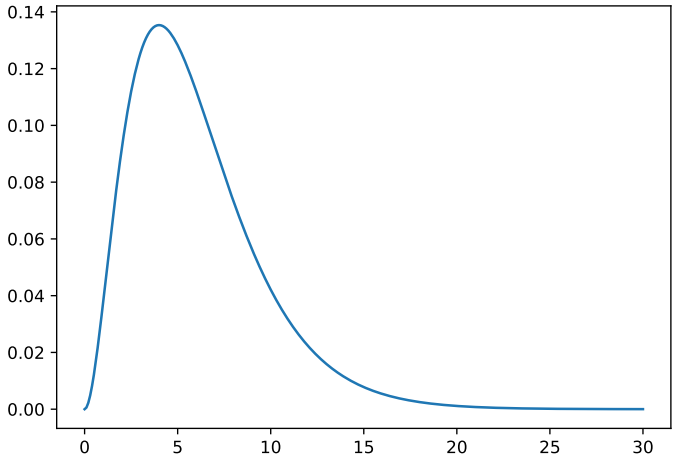
$$f(x;\nu) = \frac{1}{2^{k/2}\Gamma(k/2)}\, x^{k/2-1} e^{-x/2}$$

카이제곱 분포의 특징

카이제곱 분포는 확률밀도함수를 통해 그 특징을 파악할 수 있습니다. 첫 번째로, 카이제곱 분포의 확률변수는 항상 0 이상의 값을 가집니다. 이는 카이제곱 분포가 음수 값을 가지지 않는다는 것을 의미합니다. 두 번째로, 분포의 형태는 오른쪽으로 치우쳐져 있습니다.

- 항상 0보다 큰 값이 나오는 확률변수
- 오른쪽으로 치우진 분포

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2

k = np.arange(0,30.1,0.1)
y = chi2.pdf(k, 6)
plt.plot(k, y)
plt.show()
```



카이제곱분포와 표준정규분포의 관계

카이제곱 분포와 표준정규분포 사이의 관계는 통계학에서 중요한 개념 중 하나입니다. 특히, 표준정규분포의 확률변수 k개를 각각 제곱하여 합하면, 그 결과는 자유도가 k인 카이제곱분포를 따르게 됩니다.

$$Z_1^2 + Z_2^2 + \cdots + Z_k^2 \sim \chi^2(k)$$

여기서 $(Z_1, Z_2, \ldots, Z_k)$은 표준정규분포를 따르는 확률변수들이며, $\chi^2(k)$는 자유도가 k인 카이제곱분포를 나타냅니다.

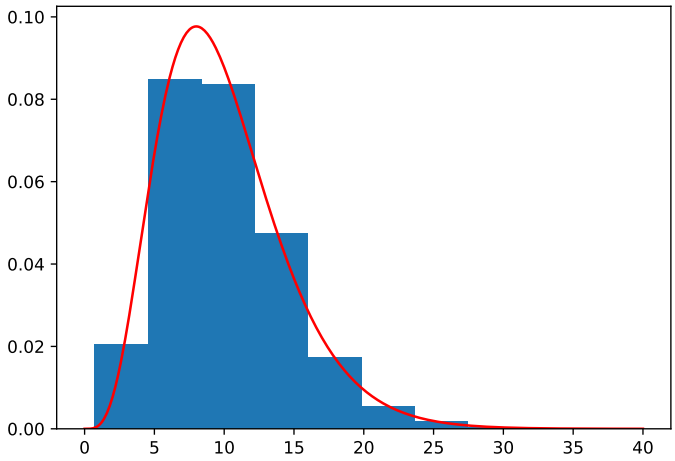
코드로 확인해보기

```
import numpy as np

# 시드 고정
np.random.seed(2023)

#랜덤샘플이므로 값이 매번 바뀜
sample = np.random.normal(0, 1, 10)
print(np.sum(np.square(sample))) # ~ chi(10)
```

```
## 21.859280277355396
```

```
result = []
for i in range(10000):
    sample = np.random.normal(0,1,10)
    result.append(np.sum(np.square(sample)))
```

```
import matplotlib.pyplot as plt
plt.hist(result, density = True)

from scipy.stats import chi2
k = np.arange(0,40.1,0.1)
density_k = chi2.pdf(k,10)
plt.plot(k, density_k, 'r')
```



카이제곱 분포와 관련된 검정들

카이제곱 분포는 여러 통계적 검정에 사용됩니다.

- 1 표본 분산 검정 (One-sample chi-squared test for variance):

  표본 분산이 특정 값과 같은지 여부를 검정합니다.

- 카이제곱 독립성 검정 (Chi-squared test for independence):

  두 카테고리컬 변수가 서로 독립적인지 아닌지를 검정합니다. 주어진 교차표 (contingency table)의 기대도수와 관측도수를 비교하여 두 변수 간의 관계를 평가합니다.

- 카이제곱 동질성 검정 (Chi-squared test for homogeneity):

  두 개 이상의 모집단에서 추출된 표본들의 분포 또는 비율이 동일한지 검정하는 데 사용됩니다. 이 검정은 카이제곱 독립성 검정과 매우 유사합니다. 실제로 **두 검정은 동일한 계산 방법을 사용**하지만, 연구의 문맥과 목적에 따라 다르게 해석됩니다.

- 카이제곱 적합도 검정 (Chi-squared goodness-of-fit test):

  관측된 빈도가 특정 이론적 분포 (예: 균등 분포)를 따르는지 검정합니다. 연속형 또는 이산형 분포에 모두 적용될 수 있습니다.

5.2 1 표본 분산 검정

데이터를 발생시키는 모분포의 분산이 특정값과 같은지 검정합니다.

귀무가설

- H_0: $\sigma^2 \leq 1.3$
- H_A: $\sigma^2 > 1.3$

카이제곱 검정 통계량

- 스튜던트 정리에 의하여 다음 검정통계량은 자유도가 $n-1$인 카이제곱분포를 따릅니다.

$$T = \frac{(n-1)S^2}{\sigma_0^2} \sim \chi^2_{n-1}$$

판단 및 주의점

- 검정통계량 T 값과 $\chi^2_{1-\alpha,n-1}$ 비교하거나 p-value를 계산한 후 유의수준과 비교
- 정규분포를 따르는 표본들에 한하여 적용합니다.

예제 (기출: 22회 통계파트 1번)

금속 성분 함유량 데이터 - 제품에 금속 재질 함유량의 분산이 1.3을 넘으면 불량이라고 보고 있는데, 제조사별로 차이가 난다고 제보를 받았다. 주어진 회사 제품의 분산에 대해 검정을 수행하시오. (유의확률 5%)

10.67, 9.92, 9.62, 9.53, 9.14, 9.74, 8.45, 12.65, 11.47, 8.62

`chi2.cdf()` 함수를 이용한 풀이

```
from scipy.stats import chi2

sample_data = [10.67, 9.92, 9.62, 9.53, 9.14, 9.74, 8.45,
               12.65, 11.47, 8.62]
n = len(sample_data)
sample_variance = np.var(sample_data,ddof=1) # ddof기본값이 0으로 지정
t = (n-1) * sample_variance / 1.3

print(t)
```

```
## 11.626530769230774
```

```
print('p-value:',1 - chi2.cdf(t, df = n-1))
```

```
## p-value: 0.23519165145589116
```

스튜던트 t 정리에 기반한 카이제곱 검정을 이용하여 풀기 때문에, 전제 조건인 데이터가 정규성을 만족한다는 것을 체크해야합니다. 따라서 **검정 전 정규성 검정을 선행합니다.**.

5.3 독립성 검정

표가 주어졌을 때, 두 카테고리컬 변수 간 상관성이 있는지 없는지를 분석합니다.

독립의 개념

통계에서 **독립의 개념은 그 적용 대상에 따라서** 정의가 달라진다.

- 두 사건이 독립이라는 의미

두 사건 A와 B가 독립이라는 의미는 특정 사건의 발생 유무가 다른 사건이 발생하는 확률에 영향이 없는 상태를 의미합니다.

$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B) \iff \mathrm{P}(A \mid B) = \frac{\mathrm{P}(A \cap B)}{\mathrm{P}(B)} = \mathrm{P}(A).$$

- 두 확률변수가 독립이라는 의미

두 확률변수 X, Y가 독립이라는 의미는 수학적으로 두 확률변수의 결합누적분포함수가 각각의 누적분포함수의 곱으로 표현이 되는 것을 의미합니다.

$$F_{X,Y}(x, y) = F_X(x)F_Y(y) \quad \text{for all } x, y$$

- 직관적으로 이해하기: 두 확률변수가 독립이 아닐 경우, 한 변수가 갖는 값을 알면, 다른 변수의 값을 예측하는데 도움이 됨.
- 예: 두 개 상자 A와 B가 있다. A에는 빨간공 10개, 파란공 10개, B에는 빨간공 5개와 파란공 5개가 들어있다. 만약 특정 상자에서 공을 꺼냈는데 빨간공이 나왔다면, 공을 꺼낸 상자는 A일까? B일까?

귀무가설 vs. 대립가설

- H_0: 변수들이 독립.
- H_A: 변수들이 독립이 아니다.

검정 통계량

$$\chi^2 = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}} \sim \chi^2_{(r-1)(c-1)}$$

- O_{ij}: 관찰빈도

- E_{ij}: 예상빈도
- i, j는 각 행, 열에 대한 인덱스
- r: 전체 행 갯수
- c: 전체 열 갯수
- **자유도 계산에 주의**하자.

예상 빈도

$$E_{ij} = \frac{i-th\ row\ total \times j-th\ column\ total}{table\ total}$$

- 주의할 점! 모든 칸의 예상 빈도가 5 이상인지 체크해야 합니다.

예제 문제

흡연 데이터

다음은 운동 선수 18명, 일반인 10명에 대하여 흡연 여부를 조사한 데이터이다. 운동 선수와 흡연 여부 간의 독립성 검정을 수행하시오.

표 5.1: 운동선수 흡연율

Info.	Non-smoker	Smoker	Total
Athlete	14	4	18
Non-athlete	0	10	10
Total	14	14	28

예상 빈도

주어진 데이터를 사용하여 예상 빈도를 계산한다.

표 5.2: 예상 빈도수

Info.	Non-smoker	Smoker
Athlete	9	9
Non-athlete	5	5

검정통계량 구하기

주어진 표를 사용하여 다음과 같은 그룹에 대응하는 값을 계산합니다.

- Athlete and Non-smoker: $\frac{(14-9)^2}{9} = 2.78$
- Non-athlete and Non-smoker: $\frac{(0-5)^2}{5} = 5$
- Athlete and Smoker: $\frac{(4-9)^2}{9} = 2.78$
- Non-athlete and Smoker: $\frac{(10-5)^2}{5} = 5$

$$T = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}} = 15.55$$

기각역은?

귀무가설을 기각하는 관찰값이 발생하는 범위는 카이제곱분포의 오른쪽 끝부분이 됩니다.

- 예상 빈도수와 차이나는 데이터들의 "제곱" 값이기 때문에 귀무가설과 벗어나는 표본들이 많을수록 검정통계량 값은 0에서 멀어지는 값을 갖게 됩니다.
- 유의수준 5%에 대응하는 기각역은 3.841을 기준으로 오른쪽 영역이 됩니다.

```
from scipy.stats import chi2

quantile = chi2.ppf(0.95, df=1)
print(quantile)
```

```
## 3.841458820694124
```

- p value는 어떻게 구할까?

```
pvalue = chi2.sf(15.55, df=1)
print(pvalue)
```

```
## 8.035164786841964e-05
```

시험 예상문제

주어진 자료를 테이블이 아닌 일반 데이터로 줄 수 있습니다. 주어진 데이터를 조건에 맞게 분류 후, 테이블 형식으로 작성하는 방법을 알아둬야 합니다.

표 5.3: 예상 데이터 구조

Petal.Length	Species
4.2	versicolor
5.1	virginica
5.1	virginica
1.5	setosa
3.0	versicolor
3.8	versicolor
5.9	virginica
5.5	virginica
4.6	versicolor
4.3	versicolor

데이터 전처리 과정

꽃잎의 길이를 구분하는 기준이 $4.3cm$이라고 한다. iris 데이터에서 꽃잎의 길이와 품종이 독립인지 검정하세요.

```
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['Species'] = iris.target
df['Species'] = df['Species'].map({0:"setosa", 1:"versicolor", 2:"virginica"})
df["size"] = df["petal length (cm)"].apply(lambda x: "Small" if x < 4.3 else "Big")

table = pd.crosstab(df["Species"], df["size"])
print(table)
```

```
## size        Big  Small
## Species
## setosa        0     50
## versicolor   27     23
## virginica    50      0
```

Python에서 검정하기

독립성 검정을 위해서 chi2_contingency() 함수를 이용합니다. chi2_contingency() 함수는 테이블을 입력값으로 받습니다.

- 함수의 결과를 사용하여, 데이터가 기대 빈도 조건 충족하는지 체크합니다.

```
from scipy.stats import chi2_contingency

chi2, p, df, expected = chi2_contingency(table)
print(expected)
```

```
## [[25.66666667 24.33333333]
##  [25.66666667 24.33333333]
##  [25.66666667 24.33333333]]
```

```
print('X-squared:', chi2.round(3), 'df:', 2, 'p-value:', p.round(3))
```

```
## X-squared: 100.285 df: 2 p-value: 0.0
```

Fisher's exact test 기대 빈도 가정이 무너졌을 경우에는 `fisher_exact()` 함수를 사용한다.

```
import pandas as pd

person = ["Athlete"]*9 + ["Non-athlete"]*5
smoke = ["Non-smoker"]*7 + ["Smoker"]*7
smoke_data = pd.DataFrame({"person":person, "smoke":smoke})

table = pd.crosstab(smoke_data['person'], smoke_data['smoke'])
print(table)
```

```
## smoke        Non-smoker  Smoker
## person
## Athlete               7       2
## Non-athlete           0       5
```

```
from scipy.stats import fisher_exact

chi2, p, dof, expected = chi2_contingency(table)
odds_ratio, p = fisher_exact(table, alternative = 'two-sided')

print(expected)
```

```
## [[4.5 4.5]
##  [2.5 2.5]]
```

```
print("P-value:", p.round(3))
```

```
## P-value: 0.021
```

Fisher's exact test의 핵심 아이디어는 초기하 분포를 이용하는 것입니다. 이 검정은 초기하 분포를 사용하여 p-value를 계산합니다.

초기하 분포 초기하 분포는 전체 모집단이 두 그룹 (A, B)로 이루어져있고, 전체 N개 중 M개가 A 그룹이 차지하고 있는 모집단에서 n개를 뽑았을때, 그 중 k개가 A 그룹에서 뽑혔을 확률을 의미한다.

$$p(k; N, K, n) = \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$$

따라서 주어진 데이터에서 p-value를 계산하는 방법은 다음과 같습니다. 9명이 운동선수 (그룹 A)이고, 전체 14개로 이루어진 모집단에서 7명을 뽑았을때, 7명이 모두 운동선수일 확률과

$$p = \frac{\binom{9}{7}\binom{5}{0}}{\binom{14}{7}}$$

14명 중 7명을 뽑았을 때 그 중 2명이 운동 선수, 5명이 운동 선수가 아닐 확률을 더해 줍니다.

$$p = \frac{\binom{9}{2}\binom{5}{5}}{\binom{14}{7}}$$

```
import scipy.special as sp

p1 = (sp.comb(9, 7) * sp.comb(5, 0) / sp.comb(14, 7))
p2 = (sp.comb(9, 2) * sp.comb(5, 5) / sp.comb(14, 7))

print(p1 + p2)
```

```
## 0.02097902097902098
```

```
print(p1 * 2)
```

```
## 0.02097902097902098
```

5.4 동질성 검정

두 개 이상의 모집단에서 추출된 표본들의 카테고리 분포 또는 비율이 동일한지 검정합니다. 데이터는 교차표 (contingency table) 형태로 제공되며, 각 셀에는 해당 카테고리에 속하는 관측치의 수가 포함됩니다.

- 귀무 가설 (H_0): 모든 모집단에서 카테고리별 비율은 동일하다.
- 대립 가설 (H_A): 적어도 한 모집단에서 카테고리별 비율이 다르다.

예제 문제

두 도시 (도시 X와 도시 Y)에서 선호하는 음료의 종류 (콜라, 사이다, 주스)를 조사하려고 합니다. 각 도시에서 100명씩 조사한 결과를 바탕으로 두 도시의 음료 선호도가 동일한지 카이제곱 동질성 검정을 사용하여 확인하려고 합니다.

	콜라	사이다	주스
도시 X	50	30	20
도시 Y	45	35	20

이러한 데이터를 바탕으로 카이제곱 동질성 검정을 수행하여 두 도시의 음료 선호도가 동일한지 검정할 수 있습니다. 여기서 카테고리는 음료가 되고, 표본집단은 도시가 됩니다.

귀무 가설 및 대립 가설 설정

- 귀무 가설 (H_0): 두 도시 (도시 X와 도시 Y)에서의 음료 선호도는 동일하다.
- 대립 가설 (H_A): 두 도시에서의 음료 선호도는 동일하지 않다.

기대 빈도 계산

두 도시에서의 음료 선호도가 동일하다고 가정하면, 각 음료의 기대 확률은 다음과 같습니다.

- 콜라: $\frac{50+45}{200} = 0.475$
- 사이다: $\frac{30+35}{200} = 0.325$
- 주스: $\frac{20+20}{200} = 0.2$

그 다음 각 도시의 총 사람수를 곱해서 콜라, 사이다, 주스를 좋아하는 사람들의 기대값을 구할 수 있습니다. 두 도시의 총 인원이 100명으로 동일하므로, 47.5, 32.5, 20으로 계산합니다.

주의사항

두 도시의 음료 선호도가 동일하다는 것이, 콜라, 사이다, 주스를 동일한 1/3 비율로 좋아한다는 의미가 아님에 주의합니다. 두 도시의 사람들이 콜라, 사이다, 주스를 좋아하는 분포가 동일한 것을 가정한다는 의미입니다.

카이제곱 통계량 계산

카이제곱 통계량은 각 카테고리별로 $\frac{(Obs-Exp)^2}{Exp}$ 의 합으로 계산됩니다.

$$Q = \sum_{i=1} \sum_{j=1} \frac{(y_{ij} - n_i \hat{p}_j)^2}{n_i \hat{p}_j}$$

검정통계량 Q는 자유도가 $k-1$인 카이제곱분포를 따른다고 알려져 있습니다. k는 카테고리 개수를 의미합니다. 또한 각 카테고리 선호도는 관측치 합과 각 표본들의 값을 분수꼴로 표현하여 추정할 수 있습니다.

$$\hat{p}_j = \frac{y_{1j} + y_{2j}}{n_1 + n_2}$$

자세한 내용은 다음 자료를 참고해주세요.

https://online.stat.psu.edu/stat415/lesson/17/17.1

p-value 계산 및 결론 도출

카이제곱 통계량 값을 카이제곱 분포에 대입하여 p-value를 계산합니다. p-value가 특정 유의 수준(예: 0.05)보다 작으면 귀무가설을 기각하고, 두 도시에서의 음료 선호도는 동일하지 않다고 결론을 내릴 수 있습니다. 그렇지 않으면 귀무가설을 기각할 수 없습니다.

```
import numpy as np
from scipy.stats import chi2_contingency

# 데이터 설정: 교차표
data = np.array([[50, 30, 20],  # 도시 X
                 [45, 35, 20]]) # 도시 Y

chi2, p, df, expected = chi2_contingency(data)

print(chi2.round(3), p.round(3))
```

```
## 0.648 0.723
```

5.5 적합도 검정

내가 관찰한 데이터가 이론적인 분포를 따르는지 검정합니다. 데이터는 역시 교차표 (contingency table) 형태로 제공되며, 각 셀에는 해당 카테고리에 속하는 관측치의 수가 포함됩니다.

- 귀무 가설 (H_0): 데이터의 모집단이 특정 분포를 따른다.
- 대립 가설 (H_A): 데이터의 모집단이 특정 분포를 따르지 않는다.

자료 및 검정 예시

다음은 140명의 신생아들의 태어난 날을 조사하여 작성한 표입니다. 카이제곱 적합도 검정을 이용하여 특정 요일에 아이들이 더 많이 태어나는지 검정할 수 있습니다.

표 5.5: 요일별 출생아 수

요일	월	화	수	목	금	토	일
출생수	13	23	24	20	27	18	15

귀무가설 vs 대립가설

- H_0: 요일별 신생아 출생 비율이 같다.
  - $p_1 = p_2 = \ldots = p_7 = \frac{1}{7}$
- H_A: 요일별 신생아 출생 비율이 모두 같지는 않다.
  - Not all $p_i = \frac{1}{7}$

검정통계량

- 앞에서 배운 카이제곱 검정 통계량과 같음. 자유도가 바뀜에 주의하자.

$$T = \sum_i \frac{(O_i - E_i)^2}{E_i} \sim \chi^2_{n-1}$$

위의 검정통계량 값을 앞에서 배운 `chisq.test()`를 사용해서 구할 수 있다. 먼저 데이터를 입력하도록 한다. `matrix()` 함수를 사용하여 데이터를 입력 후, 열과 행 이름을 부여한 후 테이블 오브젝트로 변환한다.

```
from scipy.stats import chisquare
import numpy as np

observed = np.array([13, 23, 24, 20, 27, 18, 15])
expected = np.repeat(20, 7)
statistic, p_value = chisquare(observed, f_exp=expected)

print("Test statistic: ", statistic.round(3))
```

```
## Test statistic:  7.6
```

```
print("p-value: ", p_value.round(3))
```

```
## p-value:  0.269
```

데이터 입력을 마쳤으므로, `chisquare()`를 사용하여 검정을 진행할 수 있다.

검정통계량에 대응하는 P-value 0.2688이 유의수준 5%보다 크므로, 귀무가설을 기각하지 못한다. 따라서 각 요일별 신생아 출생율은 모두 같다고 판단한다.

또한, 아래와 같이 각 셀의 기대빈도가 5보다 모두 크므로, 카이제곱검정의 결과를 신뢰할 수 있다고 판단한다.

```
print("Expected: ", expected)
```

```
## Expected:  [20 20 20 20 20 20 20]
```

5.6 비율 검정

비율 검정 역시 표본이 1개, 2개, 그리고 3개 이상인 경우에 따라서 달라집니다.

1표본 자료 및 검정 예시

다음은 어느 마을을 대상으로 원자력 발전소 건설에 찬성과 반대를 조사한 데이터이다. 마을 전체 사람들의 찬성 비율이 50%보다 높은지 검정하시오.

표 5.6: 원자력 발전소 설문

설문조사	찬성	반대
결과	45	37

귀무가설 vs. 대립가설

- $H_0: p = p_0$ vs. $H_A: p \neq p_0$
- $H_0: p \leq p_0$ vs. $H_A: p > p_0$
- $H_0: p \geq p_0$ vs. $H_A: p < p_0$

위의 선택지 중 2번째 형태의 가설을 설정해야 한다.

- $H_0: p \leq 0.5$ vs. $H_A: p > 0.5$
- H_0: 마을의 원자력 발전소 찬성 비율은 0.5 보다 낮거나 같다.
- H_A: 마을의 원자력 발전소 찬성 비율은 0.5 보다 높다.

검정통계량

귀무가설이 참이라는 전제하에 다음의 검정통계량 Z는 표준정규분포를 따른다.

$$Z = \frac{\hat{p} - p_0}{\sqrt{p_0 q_0 / n}} \sim \mathcal{N}(0, 1^2)$$

- 위 검정은 이항분포를 따르는 확률변수가 특정 조건을 만족할 때 정규분포로 근사시킬 수 있다는 사실에 기반한다.
- $X \sim B(n, p)$를 따를 때, $np \geq 10$이고, $n(1-p) \geq 10$ 조건을 만족하는 경우, 다음이 성립한다.

$$X \approx \mathcal{N}(np, np(1-p))$$

- 코드로 구해보기

```
p_hat = 45/82
z = (p_hat-0.5) / np.sqrt(0.5 * 0.5 / 82)
z
```

```
## 0.8834522085987732
```

기각역

대립 가설에 따라 기각역을 `norm.ppf()`을 사용해 계산할 수 있다. 유의수준 5%인 단측 검정에 해당하는 기각역은 다음과 같습니다.

```
from scipy.stats import norm

rounded_quantile = round(norm.ppf(0.975), 3)
print("{} 이상".format(rounded_quantile))
```

```
## 1.96 이상
```

```
print("{} 이하".format(-rounded_quantile))
```

```
## -1.96 이하
```

위에 주어진 검정통계량 0.883는 기각역에 속하지 않으므로, 5% 유의수준 하에서 귀무가설을 기각하지 못한다고 판단합니다.

Python에서 검정하기

주어진 상황에 맞게 `binom_test()`와 `proportions_ztest()` 함수 중 선택하여 사용한다.

- 조건: np_0와 $n(1-p_0)$ 모두 10 이상이다.

`proportions_ztest()` 사용하는 경우 조건을 만족하는 경우, 정규분포 근사가 가능하여 Z 검정통계량을 사용하여 검정할 수 있다. `proportions_ztest()` 함수는 카이제곱검정 통계량을 계산해주는데, 이는 앞에서 구한 Z 통계량을 제곱한 값이다.

```
from statsmodels.stats.proportion import proportions_ztest

z_score, p_value = proportions_ztest(45, 82, 0.5, alternative='larger')

print("x-sqaured:",z_score**2)
```

```
## x-sqaured: 0.7879879879879893
```

```
print("p-value:",p_value)
```

```
## p-value: 0.1873546039633795
```

```
print('sqrt_z-score',z_score)
```

```
## sqrt_z-score 0.8876868749666119
```

위 조건 불만족시 `binom_test()`를 사용한다.

binom_test() 사용하는 경우 정확하게 성공횟수가 이항분포를 따른다는 전제하에 검정통계량을 계산해준다.

```
from scipy.stats import binom_test

p_value = binom_test(x=45, n=82, p=0.5, alternative='greater')
print(p_value)
```

```
## 0.219852714592911
```

이항분포 $n = 82, p = 0.5$를 따르는 분포하에서 찬성 표 45이상을 받게 될 확률이 p-value값이 된다.

```
from scipy.stats import binom

prob_x_or_less = binom.cdf(44, 82, 0.5)
prob_x_or_more = 1 - prob_x_or_less
print(prob_x_or_more)
```

```
## 0.219852714592911
```

비율 추정 신뢰구간

유의수준 α 혹은 신뢰수준 $C = 1-\alpha$에 대응하는 모비율 p에 대한 신뢰구간을 구하는 공식은 다음과 같다.

$$\begin{aligned} & \hat{p} \pm z_{\alpha/2} S.E._{\hat{p}} \\ = & \hat{p} \pm z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \end{aligned}$$

위 식에서 $z_{\alpha/2}$는 표준정규분포에서 다음을 조건을 만족하는 임계값이며,

$$P\left(|Z| > z_{\alpha/2}\right) = 1 - C = \alpha$$

$S.E._{\hat{p}}$는 표본 비율 통계량의 표준오차를 의미한다. 또한, 임계값 $z_{\alpha/2}$와 표준오차 $S.E._{\hat{p}}$의 곱을 **margin of error (허용오차 혹은 표본 오차범위)** 라고 부른다.

표본 갯수 구하기

모비율의 구간추정 문제에서 추정의 허용오차를 특정 수준으로 맞추기 위하여, 필요한 최소한의 표본 갯수를 구하는 문제가 나오기도 한다. 이런 경우, $\hat{p}$ 값을 데이터에서 구한 값으로 대채하는 경우와 최대값인 0.5로 놓고 계산하는 경우 2가지로 나뉜다.

Type 1 다음은 어느 마을을 대상으로 원자력 발전소 건설에 찬성과 반대를 사전 조사한 데이터이다. 마을 전체 사람들의 찬성 비율 p에 대한 90% 신뢰구간의 허용오차가 2% 이하가 되도록하는 최소한의 표본을 조사하려 한다. 몇명의 사람들을 더 조사해야할까?

찬성	반대
450	370

다음을 만족하는 n의 최소 정수값을 구하면 된다.

$$z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq 0.02$$

따라서,

$$\frac{\hat{p}(1-\hat{p})}{\left(\frac{0.02}{z_{\alpha/2}}\right)^2} \leq n$$

```
from scipy.stats import norm

p_hat = 450 / (450+370)
p_hat * (1-p_hat) / (0.02 / norm.ppf(0.95))**2
```

```
## 1674.8698137525512
```

즉, 총 1675명이 필요하므로, 855명이 추가적으로 더 필요하다.

Type 2 어느 마을을 대상으로 원자력 발전소 건설에 찬성 비율을 추정하려고 한다. 마을 전체 사람들의 찬성 비율 p에 대한 95% 신뢰구간의 허용오차가 3% 이하가 되도록하는 최소한의 표본을 조사하려 한다. 몇명의 사람들을 조사해야할까?

```
p_hat = 0.5
p_hat * (1 - p_hat) / (0.03 / norm.ppf(0.975))**2
```

```
## 1067.0718946372572
```

총 1068명의 표본이 필요하다.

이론적 신뢰구간과 실제 구현된 신뢰구간의 차이점

`proportions_ztest()` 함수를 사용한 신뢰구간과 우리가 알고있는 계산식으로 구한 신뢰구간 값이 차이가 나는 것을 알 수 있다.

```
from statsmodels.stats.proportion import proportions_ztest, proportion_confint

z_score, p_value = proportions_ztest(45, 82, 0.5, alternative='two-sided')
conf_int = proportion_confint(45, 82, alpha=0.05)
print("confidence interval:",conf_int)
```

```
## confidence interval: (0.44107584336090966, 0.6564851322488465)
```

기초 통계 시간에 배우는 신뢰구간의 경우, 다음과 같이 구할 수 있다.

```
from scipy.stats import norm, chi2
import math

sqrt_qchisq = math.sqrt(chi2.ppf(0.95, 1))

print(sqrt_qchisq)
```

```
## 1.9599639845400538
```

```
print(p_hat - norm.ppf(0.975) * math.sqrt((p_hat * (1-p_hat)) / 82))
```

```
## 0.39177909306775244
```

```
print(p_hat + norm.ppf(0.975) * math.sqrt((p_hat * (1-p_hat)) / 82))
```

```
## 0.6082209069322475
```

결과을 비교해보면 비슷하게 나오는 것을 알 수 있지만, `proportions_ztest()`의 정확한 신뢰 구간을 구하기 위해서는 Wilson이 제안한 상당히 복잡한 수식의 신뢰구간을 사용한다.

$$\left(\frac{\hat{p} + \frac{z^2_{\alpha/2}}{2n} - z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{z^2_{\alpha/2}}{n}}, \frac{\hat{p} + \frac{z^2_{\alpha/2}}{2n} + z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{z^2_{\alpha/2}}{n}} \right)$$

z_α는 유의수준 α에 해당하는 임계값으로 다음을 만족하는 값이다.

$$P(Z > z_\alpha) = \alpha$$

2표본 자료 및 검정 예시

다은 서울과 제주에서 과속 여부에 대한 자료이다. 사람들이 두 지역에 따라 과속 비율이 달라지는지 여부에 대하여 검정하시오.

표 5.8: 지역별 과속 비율 조사

지역	과속	운전자
서울	145	300
제주	78	200

귀무가설 vs. 대립가설

위의 주어진 검정의 경우, 다음과 같이 귀무가설과 대립가설을 설정할 수 있습니다.

- H_0: 지역별 과속 비율$(p_1 : Seoul, p_2 : Jeju)$이 같다.
  - $p_1 = p_2$
- H_A: 지역별 과속 비율$(p_1 : Seoul, p_2 : Jeju)$은 같지 않다.
  - $p_1 \neq p_2$

검정통계량

귀무가설이 참이라는 전제 하에 다음의 검정통계량 Z는 표준정규분포를 따른다.

$$Z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim \mathcal{N}(0, 1^2)$$

```
x = [145, 78]
n = [300, 200]

z_score, p_value = proportions_ztest(count = x, nobs = n)
print("x-squared:", z_score**2)
```

```
## x-squared: 4.230679984674577
```

```
print("p-value:", p_value.round(3))
```

```
## p-value: 0.04
```

정말 위와 같은 검정통계량 공식을 사용하여 계산되는지 확인해보면 다음과 같이 구할 수 있다.

```
p1 = 145/300
p2 = 78/200
p_hat = (145+78) / 500
z = (p1-p2) / np.sqrt(p_hat*(1-p_hat)*(1/300 + 1/200))
z**2
```

```
## 4.230679984674577
```

5.7 연습문제

휴대전화 사용자들의 정치 성향은 다를까?

다음은 휴대전화와 유선전화를 모두 사용하는 사용자들과 유선전화만을 사용하는 사용자들의 정치 성향을 조사한 데이터이다. 정당 지지와 핸드폰 사용 유무 사이에 상관성을 검정해보세요.

표 5.9: 정치 성향 설문조사 결과

정당지지	핸드폰	유선전화
진보	49	47
중도	15	27
보수	32	30

여자아이 vs. 남자아이

다음은 4자녀를 둔 130가구를 조사하여 여자아이의 수를 조사한 자료이다. 여자 아이의 출생 비율이 50% 인지 유의수준 5%하에서 검정해보세요.

표 5.10: 4자녀 가정 여자아이 숫자 조사 결과

Girl	Frequency
0	10
1	31
2	44
3	34
4	11

지역별 대선 후보의 지지율

어느 도시에 있는 3개의 선거구에서 특정후보 A를 지지하는 유권자의 비율을 비교하기 위해 각 선거구에서 300명을 무작위를 추출하여 조사한 데이터이다. 주어진 데이터를 대상으로 후보A를 지지하는 비율이 3개 선거구 간에 차이가 있는지를 5% 유의수준에서 검정하라.

표 5.11: 지역별 대선 후보의 지지율

구분	선거구 1	선거구 2	선거구 3
지지함	176	193	159
지지하지 않음	124	107	141

데이터가 특정분포를 따를까?

다음의 데이터가 주어졌을때, 카이제곱 검정법을 사용하여 데이터가 모수가 2인 지수분포를 따르는지 유의수준 5% 하에서 검정해보세요.

0.211, 0.098, 0.736, 0.091, 0.756, 1.039, 0.391, 0.172, 2.113, 0.013, 0.073, 0.812, 0.132, 0.263, 0.124, 0.339, 0.092, 0.24, 0.438, 0.584, 0.722, 0.231, 0.033, 0.203, 0.177, 0.095, 0.352, 0.023

- 데이터는 3개 구간 (0, 0.2], (0.2, 0.4], (0.4, Inf)을 사용해서 검정하세요.

설문조사 환자 수

슬통 병원에서는 최근 도입한 진료 서비스에 대한 환자 만족도를 평가하고자 합니다. 병원은 환자들에게 만족도 설문지를 무작위로 분배하려고 합니다. 설문을 시작하기 전에, 병원은 적절한 표본 크기를 결정하려고 합니다.

설문지는 환자의 진료 서비스에 대한 만족도를 7점 척도로 묻습니다. 병원은 특히 환자 중 만족하거나 매우 만족하는 비율 p에 관심이 있습니다 (이는 7점 척도 중 상위 두 단계에 해당합니다).

병원은 만족도 추정의 신뢰 수준을 95%로 설정하려고 하며, 오차 범위는 3% 또는 0.03 이하로 원합니다. 보수적인 추정을 위하여 확률은 0.5로 가정하고자 합니다. 조건 만족을 위하여 설문에 필요한 최소 환자 수를 구해주세요.

설문조사 환자 수 2

다음은 슬통 병원에서 작년에 조사해놓은 환자들의 만족도 조사 결과 입니다.

7, 7, 6, 7, 6, 3, 6, 4, 5, 2, 7, 6, 6, 6, 6, 6, 6, 6, 7, 6

위의 데이터를 사용하여 설문에 필요한 최소 환자 수를 구해주세요. 병원은 만족도 추정의 신뢰 수준을 98%로 설정하려고 하며, 오차 범위는 2% 또는 0.02 이하로 원합니다.

유권자의 마음

슬통 신문사에서는 다음과 같은 42명의 시민들을 대상으로 지난 1월 A 대통령 후보를 지지하는 조사하였습니다. 다음은 대통령이 된 A 후보의 임기 시작 후 6개월이 지난 오늘, 다시 한번 동일 인원들에게 전화를 걸어 대통령 후보를 지지하는지 물어본 결과입니다.

	당선 후 지지여부	
당선 전 지지여부	지지함	지지하지 않음
지지함	17	7
지지하지 않음	5	13

사람들의 A 후보에 대한 지지율이 당선 전과 당선 후 변하였는지 검정해보세요.

제 6 장

ANOVA: Analysis of Variance

6.1 Analysis of Variance

분산 분석(ANOVA)은 이름에서 짐작할 수 있듯이 분산에 대한 검정처럼 들릴 수 있지만, 실제로는 그렇지 않습니다. ANOVA는 주로 3개 이상의 집단에서 평균의 차이를 검정하기 위해 사용되는 통계적 방법입니다.

그룹을 나누는 기준이 되는 변수의 개수에 따라서 ANOVA의 종류도 다양해집니다.

- One way ANOVA: 이 방법은 한 가지 관심 변수에 따라 그룹을 나누고 그룹 간의 평균 차이를 검정합니다. 예를 들어, 다양한 브랜드의 제품 효과를 비교할 때 사용될 수 있습니다.
- Two way ANOVA: 이 방법은 두 가지 관심 변수를 기준으로 그룹을 나누어 평균의 차이를 검정합니다. 이것은 두 가지 다른 변수가 결과에 어떤 영향을 미치는지를 동시에 알아보기 위해 사용됩니다. 예를 들면, 제품 브랜드와 사용자의 연령대를 기준으로 제품의 효과를 비교하는 경우에 적용될 수 있습니다.

6.2 One-way ANOVA

모델 가정

One-way ANOVA 모델은 하나의 독립 변수에 따라 그룹 간의 평균 차이를 검정하기 위해 사용됩니다. 이 모델에서는 데이터가 다음과 같은 원리로 발생한다고 가정합니다:

$$x_{ij} = \mu_i + \epsilon_{ij}$$

여기서: - x_{ij}는 i번째 집단의 j번째 관찰값을 나타냅니다. - μ_i는 i번째 집단의 평균을 나타냅니다. - ϵ_{ij}는 오차항으로, 각 관찰값에 더해지는 랜덤한 변동을 나타냅니다.

- i는 집단의 번호를 나타내며, $i = 1, ..., k$로서 총 k개의 집단이 있습니다.
- j는 각 집단 내에서의 관찰값의 번호를 나타내며, $j = 1, ..., n_i$로서 i번째 집단에는 총 n_i개의 관찰값이 있습니다.
- ϵ_{ij}의 분포는 정규 분포로, 평균이 0이며 분산이 σ^2입니다.

이러한 가정 하에서, 각 집단 i에서는 n_i 개의 표본이 관찰됩니다.

중요한 가정

One-way ANOVA 모델을 사용할 때 고려해야 하는 몇 가지 중요한 가정들이 있습니다.

정규성 모든 에러항은 정규 분포를 따라야 합니다.

모델 적합 후 에러항의 분포를 체크하는 것은 매우 중요합니다. 이를 통해 모델의 가정이 유효한지 확인할 수 있습니다.

등분산성 모든 그룹에서의 에러항의 분산은 동일해야 합니다.

독립성 각 집단 i의 관찰값은 서로 독립적이며, 각 집단은 정규분포 평균 μ_i와 분산 σ^2를 따라야 합니다.

귀무가설과 대립가설

- H_0: 모든 집단의 평균이 동일하다.
  - $\mu_1 = \mu_2 = ... = \mu_k$
- H_A: 평균이 다른 집단이 적어도 하나 존재한다.
  - Not all of $\mu_1, \mu_2, ...$ and μ_k are equal.

ANOVA의 핵심 아이디어

데이터 분해

ANOVA의 아이디어는 전체 데이터의 변동성을 분해하면 두 가지로 분리할 수 있다는 것에서부터 시작합니다.

그룹간 변동성과 그룹 안에서의 변동성 데이터의 변동성을 측정하는 도구로써 SS (Sums of squares)라는 개념을 사용합니다.

- SST (Sums of Squares for Total): 각 데이터와 전체 평균과의 변동성
- SSG (Sums of Squares for Groups): 각 그룹 평균과 전체 평균과의 변동성
- SSE (Sums of Squares for Error): 각 데이터와 대응하는 그룹 평균과의 변동성

$$\underbrace{\sum_{i=1}^{k}\sum_{j=1}^{n_j}\left(X_{ij}-\overline{X}_{..}\right)^2}_{SST} = \underbrace{\sum_{i=1}^{k} n_i\left(\overline{X}_{i.}-\overline{X}_{..}\right)^2}_{SSG} + \underbrace{\sum_{i=1}^{k}\sum_{j=1}^{n_j}\left(X_{ij}-\overline{X}_{i.}\right)^2}_{SSE}$$

- $n = \sum_{i=1}^{k} n_i$
- $\frac{SSG}{\sigma^2} \sim \chi^2_{k-1}$
- $\frac{SSE}{\sigma^2} \sim \chi^2_{n-k}$

자세한 증명은 Hogg 책 ANOVA 파트 참조.

검정 통계량

ANOVA에서 중요한 검정 통계량은 F 값입니다. 이는 그룹 간과 그룹 내의 분산을 비교하기 위해 사용되며, F 분포를 따릅니다.

$$F = \frac{MSG}{MSE} = \frac{\text{그룹 간 평균들의 분산}}{\text{그룹 안에서 분산}}$$

여기서,

- MSG는 그룹 간의 평균 제곱 오차로, 각 그룹의 평균과 전체 평균 간의 차이를 기반으로 계산됩니다.

$$MSG = \frac{n_1(\overline{X}_1 - \overline{X})^2 + ... + n_k(\overline{X}_k - \overline{X})^2}{k-1}$$

- MSE는 그룹 내의 평균 제곱 오차로, 각 그룹 내의 분산을 기반으로 계산됩니다.

$$MSE = \frac{(n_1-1)s_1^2 + ... + (n_k-1)s_k^2}{n-k}$$

F 값은 그룹 간의 평균 차이가 그룹 내의 변동에 비해 얼마나 큰지를 측정합니다. F 값이 크면 그룹 간의 차이가 유의미하다는 것을 나타냅니다.

MSE의 의미

MSE는 여러 그룹 내의 표본 분산을 통합하여 모분산을 추정하는 방법입니다. 이를 **pooled variance** 라고도 부릅니다.

- 특징: k개의 그룹 크기를 고려하여 가중 평균을 구함

이렇게 해서 각 그룹의 표본 수와 분산을 모두 반영하여 전체 분산을 추정하게 됩니다.

$$s_p^2 = \frac{(n_1-1)\,s_1^2 + (n_2-1)\,s_2^2 + ... + (n_k-1)\,s_k^2}{(n_1-1) + (n_2-1) + ... + (n_k-1)}$$

왜 작동할까?

ANOVA의 핵심 아이디어는 각 그룹의 평균이 전체 평균에서 얼마나 움직이는지와 그룹 내의 개별 표본이 그룹 평균에서 얼마나 움직이는지를 비교하는 것입니다.

- F 값이 작으면: 그룹 간의 차이가 그룹 내의 차이에 비해 작다는 것을 나타냅니다. 이는 **귀무가설이 참일 가능성이 높음**을 의미합니다.
- F 값이 크면: 그룹 간의 차이가 그룹 내의 차이에 비해 크다는 것을 나타냅니다. 이는 **귀무가설이 거짓일 가능성이 높음**을 의미합니다.

이 때, F 통계량은 $F_{k-1,n-k}$ 분포를 따릅니다. 여기서, $k-1$과 $n-k$는 각각 그룹 간과 그룹 내의 자유도를 나타냅니다.

ANOVA 예제

데이터 불러오기

데이터를 불러오기 위해 pandas 라이브러리를 사용하고, read_csv 함수로 데이터 파일을 불러옵니다. 그 후, head() 함수를 사용하여 데이터의 첫 5행을 확인합니다.

```
import pandas as pd

anova_data = pd.read_csv('./data/anova_example.csv')
anova_data.head()
```

```
##    Minutes      Odor
## 0       92  Lavender
## 1      126  Lavender
## 2      114  Lavender
## 3      106  Lavender
## 4       89  Lavender
```

데이터 살펴보기

데이터를 더 깊게 살펴보기 위해, 각 Odor 그룹에 대한 기술 통계량을 계산합니다. 이를 위해 groupby와 describe 함수를 사용하여 그룹별 평균, 표준편차, 최소값, 최대값 등의 통계량을 얻을 수 있습니다.

```
anova_data.groupby(['Odor']).describe()
```

```
##          Minutes
##            count        mean        std   min    25%    50%    75%    max
## Odor
## Lavender    14.0  107.142857  17.302099  76.0  94.25  105.5  121.5  137.0
## Lemon       14.0   88.642857  14.457402  63.0  75.75   88.5  100.0  112.0
## No_odor     15.0   87.466667  16.282623  68.0  72.50   85.0   97.5  121.0
```

ANOVA 실행하기

ANOVA 분석을 위해 statsmodels 라이브러리를 사용합니다. ols() 함수를 사용하여 선형 회귀 모델을 만들고, 이를 기반으로 anova_lm() 함수를 통해 ANOVA 분석을 수행합니다.

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols('Minutes ~ Odor', data=anova_data).fit()
result = sm.stats.anova_lm(model)
print(result)
```

```
##             df        sum_sq      mean_sq         F    PR(>F)
## Odor       2.0   3457.524142  1728.762071  6.700198  0.003093
## Residual  40.0  10320.661905   258.016548       NaN       NaN
```

위의 결과는 ANOVA 표라고 불리며, 검정통계량 F 값 6.7에 대응하는 p-value는 0.003이라는 것을 알 수 있습니다.

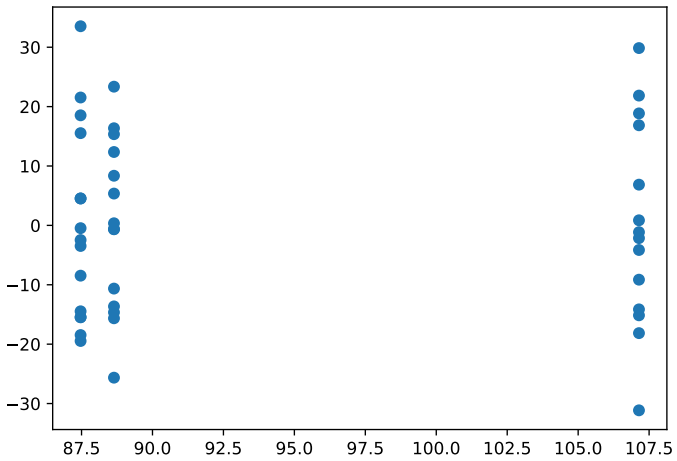
가정 체크

ANOVA에서 가정을 체크하는 것은 중요합니다. 가정이 만족되지 않으면 ANOVA의 결과를 신뢰하기 어렵습니다. 가정 중 하나는 잔차들의 정규성입니다.

잔차들의 정규성

모델의 잔차를 적합값에 대해 시각화하여 잔차의 분포와 패턴을 확인합니다. 주어진 그래프에서 잔차들은 0을 중심으로 분포되어 있으며, 어떤 특정한 패턴 없이 무작위로 분포되어 있어야 합니다.

```
import matplotlib.pyplot as plt

residuals = model.resid
fitted_values = model.fittedvalues

plt.scatter(fitted_values, residuals)
plt.show()
```



또한, 잔차들의 분포가 정규분포를 따르는지도 확인해야 합니다. `pingouin` 라이브러리를 사용하여 Q-Q 그래프를 그려 확인하도록 하겠습니다.

```
import pingouin as pg

residuals = model.resid
```

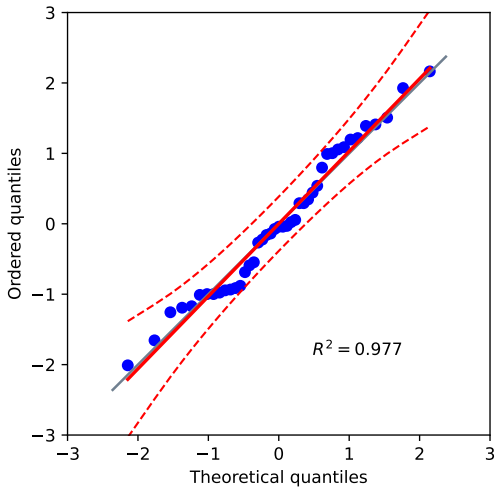
```
ax = pg.qqplot(residuals, dist='norm')
plt.ylim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.xlim(-3, 3)
```

```
## (-3.0, 3.0)
```

```
plt.show()
```



Shapiro-Wilk 검정을 통하여 잔차의 정규성을 통계적으로 검정합니다.

```
from scipy import stats

W, p = stats.shapiro(residuals)
print(round(W, 3), round(p, 3))
```

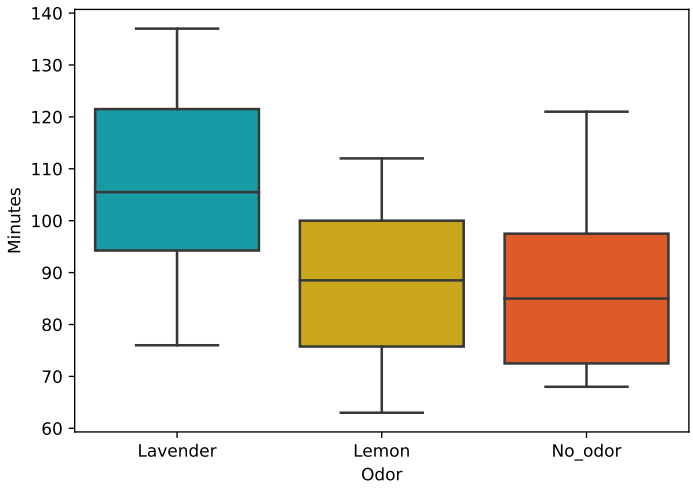
```
## 0.971 0.353
```

검정통계량 W값 0.971에 대응하는 p-value 0.353이 유의 수준 0.05보다 크므로, 잔차가 정규분포를 따른다는 귀무가설을 기각할 수 없습니다. 따라서 정규성 가정을 만족한다고 판단합니다.

등분산 가정 체크

- 각 그룹별 상자그림 그려보기

```
import seaborn as sns

sns.set_palette(["#00AFBB", "#E7B800", "#FC4E07"])
sns.boxplot(x = "Odor", y = "Minutes", data = anova_data, order = ["Lavender", "Lemon", "No_odor"])
plt.xlabel("Odor")
plt.ylabel("Minutes")
plt.show()
```



잔차들을 그룹별로 묶어서 Levene 검정을 실시할 수 있습니다.

```
from scipy.stats import levene

odor_groups = [anova_data[anova_data['Odor'] == odor_type]['Minutes'] for odor_type in anova_data['
W, p = levene(*odor_groups, center = 'mean')
print(W, p)
```

```
## 0.17643170248475967 0.8389048016208949
```

검정통계량 값 0.132에 대응하는 p-value가 0.87이므로 유의수준 0.05보다 큽니다. 따라서 각 그룹별 분산이 같다는 귀무가설을 기각하지 못하므로, 등분산 가정을 만족한다고 판단합니다.

사후 검정

ANOVA는 여러 그룹 간의 평균 차이가 유의미한지만을 판단합니다. 만약 ANOVA 결과에서 귀무가설이 기각된다면, 어떤 그룹 간에 평균 차이가 유의미하게 나타났는지를 알기 위해 사후 검정 (post-hoc test)을 수행해야 합니다.

- 귀무가설 H_0: $\mu_{Lavender} = \mu_{Lemon} = \mu_{No\ odor}$

평균이 다른 그룹이 존재한다는 것은 판명이 났지만, 어느 그룹이 다른지 모르는 상황!

추가 검정 가설

- $\mu_{Lavender} = \mu_{Lemon}$
- $\mu_{Lemon} = \mu_{No\ odor}$
- $\mu_{Lavender} = \mu_{No\ odor}$

위에서 제시한 세 개의 추가 검정 가설을 각각 t-test로 검정하려면, 여러 번의 t-test를 수행하는 것이므로 유의수준을 조정해야 합니다.

Bonferrorni

이를 위해 Bonferroni 보정 같은 방법을 사용할 수 있습니다. Bonferroni 보정은 원래의 유의수준 (예: 0.05)을 t-test의 수 (여기서는 3)로 나누어 각 t-test의 유의수준을 조정합니다.

$$\text{보정된 유의수준} = \frac{0.05}{3} = 0.0167$$

```
from scipy.stats import ttest_ind

lavender = anova_data[anova_data['Odor'] == 'Lavender']['Minutes']
lemon = anova_data[anova_data['Odor'] == 'Lemon']['Minutes']
no_odor = anova_data[anova_data['Odor'] == 'No_odor']['Minutes']

t_stat1, p_val1 = ttest_ind(lavender, lemon)
t_stat2, p_val2 = ttest_ind(lemon, no_odor)
t_stat3, p_val3 = ttest_ind(lavender, no_odor)

print("Lavender vs Lemon: t =", round(t_stat1, 3), ", p =", round(p_val1, 3))
```

```
## Lavender vs Lemon: t = 3.07 , p = 0.005
```

```
print("Lemon vs No odor: t =", round(t_stat2, 3), ", p =", round(p_val2, 3))
```

```
## Lemon vs No odor: t = 0.205 , p = 0.839
```

```
print("Lavender vs No odor: t =", round(t_stat3, 3), ", p =", round(p_val3, 3))
```

```
## Lavender vs No odor: t = 3.155 , p = 0.004
```

결과를 기반으로 사후 검정의 해석:

1. **Lavender vs Lemon**:
    - t 값: 3.07, p-value: 0.005
    - **해석**: p 값이 0.0167보다 작으므로, 라벤더와 레몬 그룹 간의 평균에는 유의미한 차이가 있다고 할 수 있습니다.

2. **Lemon vs No odor**:
    - t 값: 0.205, p-value: 0.839
    - **해석**: p 값이 0.0167보다 크므로, 레몬과 무향 그룹 간의 평균에는 유의미한 차이가 없다고 할 수 있습니다.
3. **Lavender vs No odor**:
    - t 값: 3.155, p-value: 0.004
    - **해석**: p 값이 0.0167보다 작으므로, 라벤더와 무향 그룹 간의 평균에는 유의미한 차이가 있다고 할 수 있습니다.

정리하면, 라벤더와 레몬 그룹 간 그리고 라벤더와 무향 그룹 간에는 평균 차이가 유의미하게 나타났습니다. 반면, 레몬과 무향 그룹 간에는 평균 차이가 유의미하지 않았습니다.

Tukey HSD

Tukey HSD 검정은 ANOVA의 사후검정 중 하나입니다. 이 검정은 모든 그룹 간의 평균 차이를 비교하며, 이로 인한 유형 1 오류를 제어합니다. `statsmodels` 라이브러리의 `pairwise_tukeyhsd()` 함수를 사용하여 Tukey HSD를 수행할 수 있습니다. 이러한 함수를 사용하면, 유의수준을 자동으로 보정해 주므로 편합니다.


```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Perform multiple comparisons
tukey_results = pairwise_tukeyhsd(anova_data['Minutes'], anova_data['Odor'])
print(tukey_results)
```

```
##   Multiple Comparison of Means - Tukey HSD, FWER=0.05
## =====================================================
##  group1   group2 meandiff p-adj   lower    upper  reject
## -----------------------------------------------------
## Lavender   Lemon    -18.5 0.0111 -33.2768 -3.7232   True
## Lavender No_odor -19.6762 0.0057 -34.2046 -5.1477   True
##    Lemon No_odor  -1.1762 0.9788 -15.7046 13.3523  False
## -----------------------------------------------------
```

Games-Howell test

Games-Howell 검정은 ANOVA의 등분산 가정이 충족되지 않을 때 선택하는 ANOVA (ANOVA with Welch's correction) 검정과 같이 사용되는 사후검정입니다. 이는 각 그룹 간의 분산이 다를 때 특히 유용합니다. `pingouin` 라이브러리를 사용하여 Games-Howell 검정을 수행할 수 있습니다.

```
import pingouin as pg

pg.pairwise_gameshowell(data=anova_data, dv='Minutes', between='Odor')
```

```
##           A        B     mean(A)  ...         df      pval    hedges
## 0  Lavender    Lemon  107.142857  ...  25.203999  0.013568  1.126563
## 1  Lavender  No_odor  107.142857  ...  26.537150  0.010881  1.139639
## 2     Lemon  No_odor   88.642857  ...  26.940005  0.976900  0.074087
##
## [3 rows x 10 columns]
```

주어진 유의수준 0.05 하에서 판단하면, 앞에서와 동일한 결과(Lemon - No_odor 그룹 만 유의미한 차이 없음)를 얻게 됩니다.

Kruskal-Wallis Test

Kruskal-Wallis 검정은 비모수적인 방법으로 ANOVA의 대안으로 사용됩니다. 특히 데이터가 정규분포를 따르지 않을 때 유용하게 사용됩니다. 이 검정은 **각 표본의 순위를 기반**으로 하며, 모든 그룹의 중앙값이 같은지를 검정합니다.

특징

- 정규성 가정이 필요하지 않습니다.
- 모든 그룹의 분산이 동일하다는 가정이 필요하지 않습니다.
- 크기가 다른 여러 그룹에 대해 중앙값의 차이를 검정합니다.

즉, 데이터가 정규분포를 따르지 않거나, 등분산성 가정을 충족하지 못할 때 사용 가능하며, 데이터에 이상치가 많은 경우에도 사용할 수 있습니다.

귀무가설 vs. 대립가설

비모수 검정이므로 모평균이 아닌 분포의 중앙(중앙값)에 대한 검정을 시행한다.

- 귀무가설 H_0: $\eta_1 = \eta_2 = ... = \eta_k$
- 대립가설 H_A: Not all of $\eta_1, \eta_2, ...$ and η_k are equal.

검정통계량

검정통계량 H는 주어진 공식에 따라 계산되며, 크기가 큰 표본에서는 카이제곱 분포를 따르게 됩니다.

$$H = \frac{12}{N\,(N+1)} \sum_{i=1}^{k} \frac{R_{.i}^2}{n_i} - 3\,(N+1)$$

여기서

- $R_{.i}$는 i번째 그룹의 순위합입니다.
- N는 전체 표본의 크기입니다.
- k는 그룹의 수입니다.
- n_i는 i번째 그룹의 표본 크기입니다.

주어진 검정통계량 H는 각 그룹의 최소 표본 크기가 커지면 커질수록 자유도 $k-1$인 카이제곱분포를 따르게 된다.

Python에서 검정하기

scipy.stats 패키지의 kruskal() 함수를 사용한다.

```
from scipy.stats import kruskal

odor_groups = [anova_data[anova_data['Odor'] == odor_type]['Minutes']
    for odor_type in anova_data['Odor'].unique()]

H, p = kruskal(*odor_groups)
print(H, p)
```

```
## 10.316640217637701 0.005751353222496428
```

검정통계량값 10.31에 대응하는 p-value가 0.0057이므로 유의수준 0.05에 비하여 작습니다. 따라서 귀무가설을 기각하며, 각 그룹의 중앙값이 다른 그룹이 존재한다고 판단할 수 있습니다. 위의 유의확률 값이 카이제곱 분포에서 계산된 것은 다음의 코드를 통하여 확인 할 수 있습니다.

```
from scipy.stats import chi2
1 - chi2.cdf(10.31, 2)
```

```
## 0.005770480075096951
```

Kruskal-Wallis Test의 사후 검정

statsmodels.stats 패키지의 multitest() 함수를 사용하여 사후 검정을 수행합니다.

```
from statsmodels.stats import multitest

odor_groups = [anova_data[anova_data['Odor'] == odor_type]['Minutes']
    for odor_type in anova_data['Odor'].unique()]

H, p = kruskal(*odor_groups)

reject, p_values_corrected, _, _ = multitest.multipletests(p, method='fdr_bh')

print(kruskal(*odor_groups))
```

```
## KruskalResult(statistic=10.316640217637701, pvalue=0.005751353222496428)
```

```
print(p_values_corrected)
```

```
## [0.00575135]
```

6.3 Two-way ANOVA

이원 분산분석(Two-way ANOVA)은 두 개의 독립변수가 종속변수에 미치는 영향을 동시에 분석하는 방법입니다. 이는 실험 설계에서 두 가지 요인(독립변수)의 **주효과** 및 그들 간의 **상호작용 효과**를 파악하기 위해 사용됩니다.

- 주효과 (Main Effects): 각각의 독립변수가 종속변수에 미치는 평균적인 영향을 의미합니다.
- 상호작용 효과 (Interaction Effect): 한 독립변수의 영향이 다른 독립변수의 수준에 따라 달라지는 경우, 이 두 변수 간에 상호작용이 있다고 말합니다.

Two-Way ANOVA의 모델 가정

이원 분산분석(Two-Way ANOVA)는 두 개의 범주형 독립변수와 하나의 연속형 종속변수 간의 관계를 분석하는 통계적 방법입니다. 이원 분산분석의 주요 가정은 다음과 같습니다.

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk} \overset{ind.}{\sim} \mathcal{N}(\mu_{ij}, \sigma^2)$$

- μ는 전체 평균을 나타냅니다.
- 두 변수가 각각 I와 J 레벨을 가지고 있다고 가정합니다.
- α_i는 첫 번째 범주형 변수의 i번째 수준의 효과를 나타냅니다. $i = 1, ..., I$
- β_j는 두 번째 범주형 변수의 j번째 수준의 효과를 나타냅니다. $j = 1, ..., J$
- $(\alpha\beta)_{ij}$는 두 변수의 i번째 및 j번째 수준 간의 상호작용 효과를 나타냅니다.
- ϵ_{ijk}는 i번째 및 j번째 수준에 속한 k번째 관찰값의 잡음효과라 생각하면 되며, 정규분포 $\mathcal{N}(0, \sigma^2)$를 따릅니다.

이제는 모델식을 보고 관련한 가정을 추정해 볼 수 있어야 합니다. 잔차들이 정규분포를 따르고 있으며, 분산이 σ^2로 같다는 가정을 만족해야 합니다.

귀무가설 및 대립가설

- 주효과에 대한 귀무가설: 각 독립변수의 모든 수준들 사이에는 차이가 없다.
- 상호작용에 대한 귀무가설: 두 독립변수의 상호작용 효과는 없다.

Two-Way ANOVA 예제

TV에서 광고를 방송할 때, 광고의 길이와 노출 횟수가 광고의 효과에 어떤 영향을 미치는지 알아보기 위해 실험을 설계했습니다.

- 참가자 200명을 모집하여 다양한 조건의 광고에 노출시켰습니다.
- 광고를 시청한 후에 참가자들은 물건에 대한 구매 의사를 0부터 100까지의 점수로 평가하는 설문조사를 완료했습니다.

2개 변수

- 광고 길이
- 노출 횟수

데이터 구조

표 6.1: 광고 노출횟수 효과 측정실험

광고길이	1회	3회	5회
30초	그룹1	그룹2	그룹3
90초	그룹4	그룹5	그룹6

검정 데이터에 따른 분류

Two-Way ANOVA를 수행할 때, 실험 설계와 데이터의 특성에 따라 분류할 수 있습니다. 우리는 다음과 같은 조건을 충족하고 있다고 생각합니다.

- Crossed design: 이 디자인에서는 모든 조합의 셀(예: 광고 길이와 노출 횟수의 조합)에 대한 정보를 모두 수집합니다.
- CRD 혹은 RBD를 만족합니다.
  - Completely Randomized Design (CRD)
    * 각 처리 조합에 대한 관찰값은 무작위로 선택됩니다.
    * 모든 셀에 대한 표본은 독립적입니다.
    * 모든 셀의 반응변수는 동일한 분산 을 가진 정규분포를 따릅니다.
  - Randomized Block Design (RBD)
    * 실험 단위를 유사한 블록으로 그룹화하고, 각 블록 내에서 처리를 무작위로 할당합니다.
    * 블록 내에서는 모든 셀에 대한 표본이 독립적입니다.
    * 각 셀의 반응변수는 동일한 분산 σ^2 을 가진 정규분포를 따릅니다.
- 균형 설계 (Balanced Design): 이 디자인에서는 각 셀에 동일한 수의 관찰값이 포함됩니다. 이러한 설계는 통계적 검정력을 높이고, 분석을 단순화하는 데 도움이 됩니다. 각 셀마다 같은 표본 갯수를 가진다. (최대한 노력)

해석 방법

Main Effects

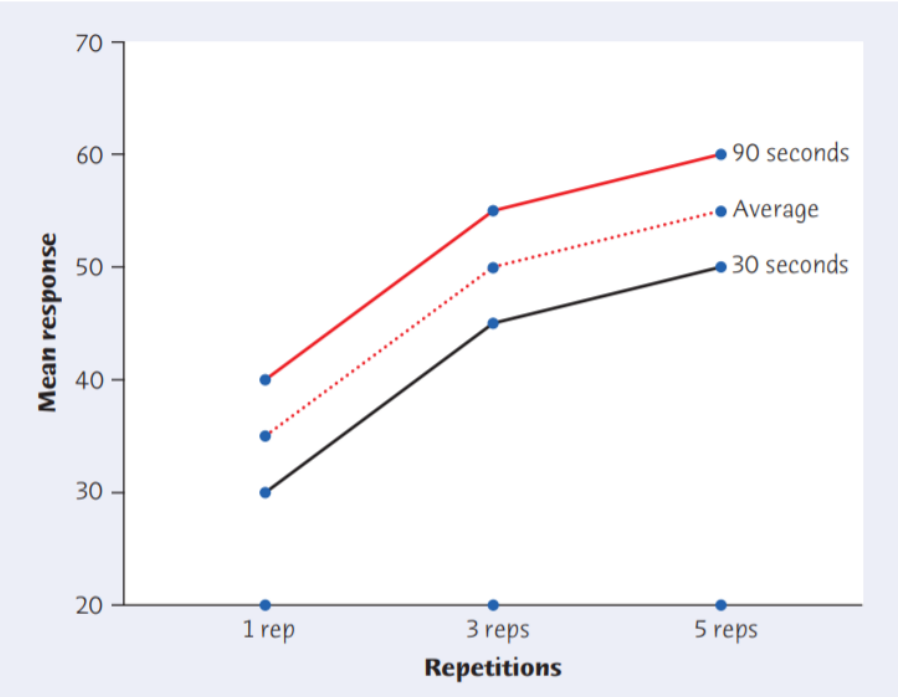
- 광고 길이 Main effect
  - 30 초 광고 평균 41.7
  - 90 초 광고 평균 51.7
- 노출 횟수 Main effect
  - 1회 평균 35
  - 3회 평균 50
  - 5회 평균 55

No interaction

- 평행 라인이 나왔을때

표 6.2:

광고길이	1회	3회	5회
30초	30	45	50
90초	40	55	60



Main Effects with Interactions

노출횟수 고려시 광고길이에 따른 Main effect 없음.

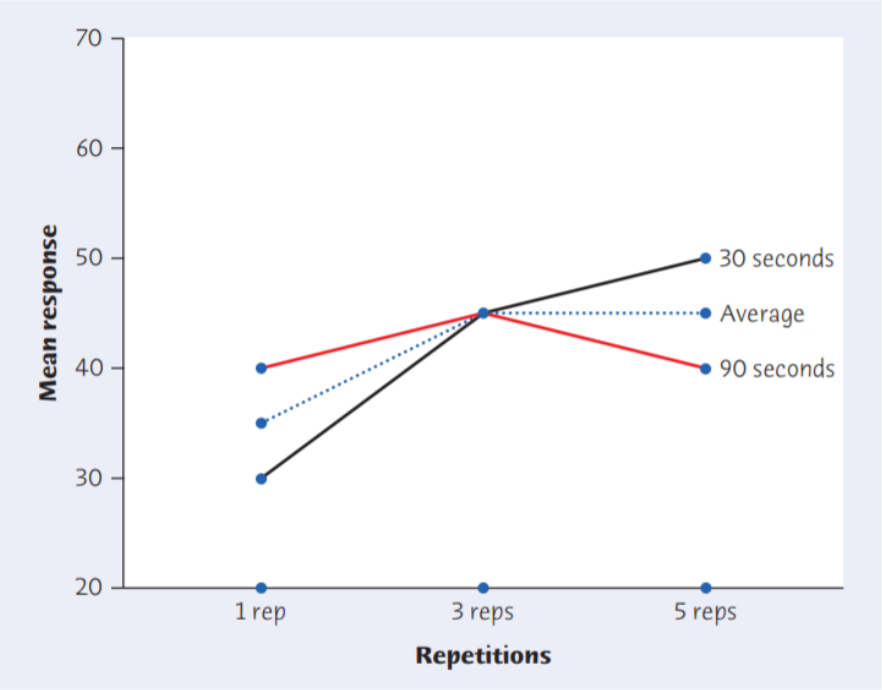
- Interaction 존재시 Main effects 효과가 줄어듦.
- 광고 길이 Main effect
  - 30 초 광고 평균 41.7
  - 90 초 광고 평균 41.7
- 노출 횟수 Main effect
  - 1회 평균 35
  - 3회 평균 45
  - 5회 평균 45

표 6.3:

광고길이	1회	3회	5회
30초	30	45	50
90초	40	45	40

Interaction

- 평행 라인이 안나옴

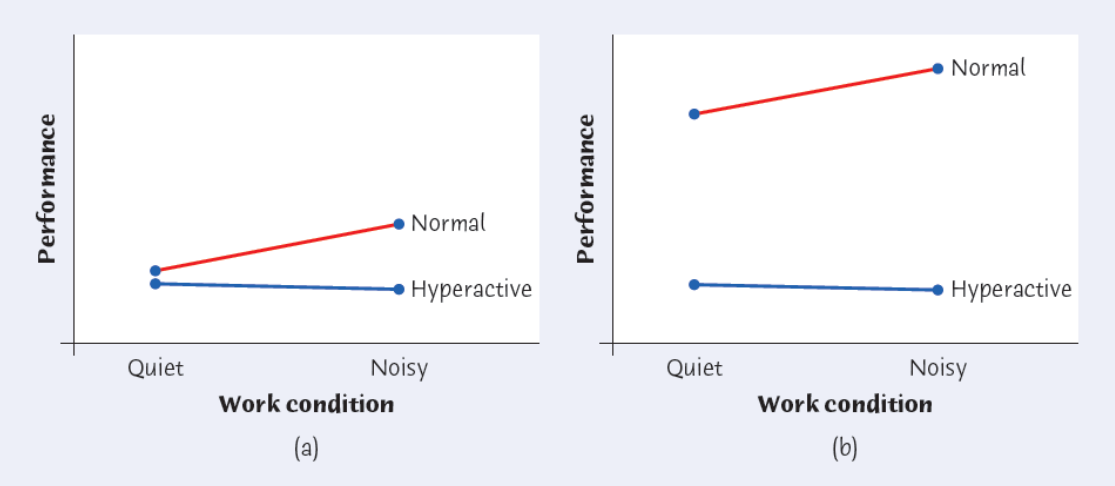


어떤 효과가 더 중요할까?

- 같은 기울기 다른 위치

Main effect vs. Interaction

- 왼쪽: Interaction이 상당히 중요
- 오른쪽: Main effect가 상당히 중요

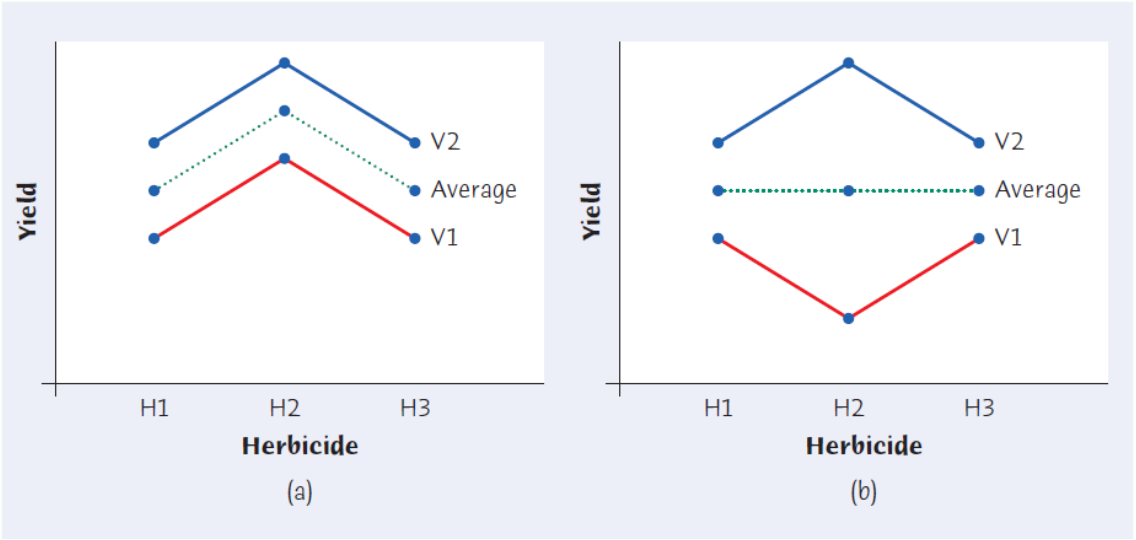


연습문제

- 각 그래프에 대하여 다음을 답하시오.

용어 구분하기

- Interaction 유무
- Main effect 존재 유무: H 변수, V 변수



Two-way ANOVA 분석 전체 과정

1. 그룹 평균과 분산을 구함
   - 그룹 평균 plot을 그린다.
   - ANOVA 가정 체크
2. ANOVA 분석 수행 및 해석
   - 3개의 F 테스트와 p-value가 주어짐.
3. 검정 질문들
   - Interaction이 통계적으로 유의한가?
   - 가로 변수에 대한 main effect가 통계적으로 유의한가?
   - 세로 변수에 대한 main effect가 통계적으로 유의한가?

연습문제

```
two_anova_data = pd.read_csv('./data/ad-two-way-anova.csv')
grouped = two_anova_data.groupby(['ad_count', 'ad_length']).mean()
grouped = grouped.unstack(level=-1)
grouped
```

```
##                response
```

```
## ad_length         30         90
## ad_count
## 1          12.083333  12.833333
## 3           9.750000   8.583333
```

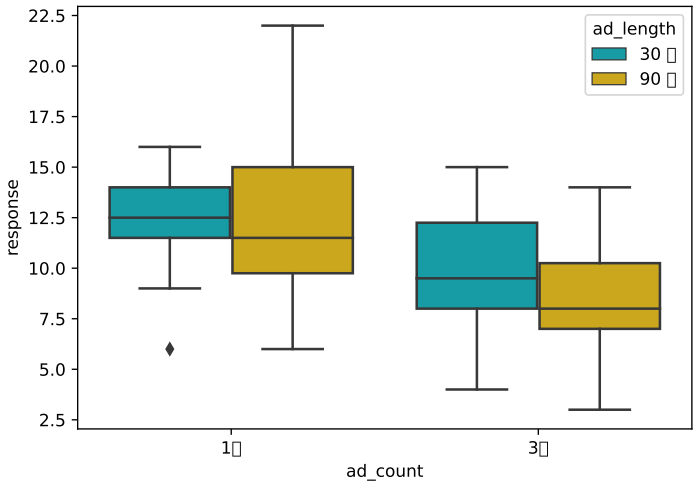
Visualization

- 가정 체크 해볼 것

```
import seaborn as sns

two_anova_data['ad_length'] = pd.Categorical(two_anova_data['ad_length'], categories=[30, 90], orde
two_anova_data['ad_length'] = two_anova_data['ad_length'].cat.rename_categories(["30 초", "90 초"])

two_anova_data['ad_count'] = pd.Categorical(two_anova_data['ad_count'], categories=[1, 3], ordered=
two_anova_data['ad_count'] = two_anova_data['ad_count'].cat.rename_categories(["1회", "3회"])

sns.boxplot(x='ad_count', y='response', data=two_anova_data, hue='ad_length')
```

```
## <AxesSubplot:xlabel='ad_count', ylabel='response'>
```

```
plt.show()
```



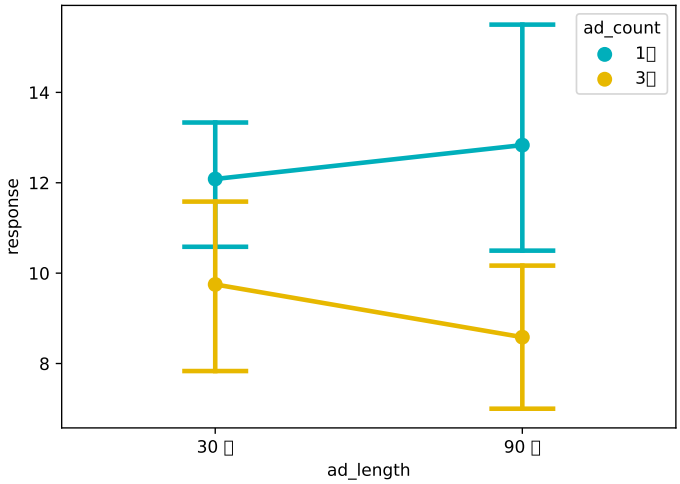
Interaction plot 해석하기

- Main effect 유무 체크
- Interaction 유무 체크

```
import seaborn as sns

sns.pointplot(x='ad_length', y='response',
    data=two_anova_data, hue='ad_count', capsize=.2)
```

```
## <AxesSubplot:xlabel='ad_length', ylabel='response'>
```

```
plt.show()
```



Two-way ANOVA 테이블

F 분포 검정

- 자유도는 해당 테이블

Main effects 모델

표 6.4: 상호작용 미포함 Two way ANOVA

Source	SS	df	Mean.Square	F
Factor A	SS(A)	(I-1)	SS(A)/(I-1)	MSA/MSE
Factor B	SS(B)	(J-1)	SS(B)/(J-1)	MSB/MSE
Error	SSE	(N-I-J+1)	SSE/(N-I-J+1)	
Total	SS(Total)	(N-1)		

- 자유도 예

$$MSA/MSE \sim F_{I-1,N-I-J+1}$$

표 6.5: 상호작용 포함 Two way ANOVA

Source	SS	df	Mean.Square	F
Factor A	SS(A)	(I-1)	SS(A)/(I-1)	MSA/MSE
Factor B	SS(B)	(J-1)	SS(B)/(J-1)	MSB/MSE
Interaction	SS(AB)	(I-1)(J-1)	SS(AB)/((I-1)(J-1))	MSAB/MSE
Error	SSE	(N-IJ)	SSE/(N-IJ)	
Total	SS(Total)	(N-1)		

Interaction term 모델

Python에서 Two-way ANOVA 수행하기

인터렉션이 존재하는 모델을 수식을 사용하여 넣는 방법을 주의해서 알아두자. 회귀분석이나 일반화 선형모형에서도 같은 방식이 사용된다.

- `response ~ ad_count + ad_length`
- `response ~ ad_count + ad_length + ad_count:ad_length`
- `response ~ ad_count * ad_length`

위의 2번째, 3번째 방법이 동일하게 인터렉션이 존재하는 모델을 나타낸다.

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols('response ~ ad_count*ad_length', data=two_anova_data).fit()
result = sm.stats.anova_lm(model)
print(result)
```

```
##                       df      sum_sq     mean_sq          F    PR(>F)
## ad_count             1.0  130.020833  130.020833  10.560068  0.002218
## ad_length            1.0    0.520833    0.520833   0.042301  0.837995
## ad_count:ad_length   1.0   11.020833   11.020833   0.895093  0.349268
## Residual            44.0  541.750000   12.312500        NaN       NaN
```

6.4 모델 검정 방법

등분산 가정 확인

```
import matplotlib.pyplot as plt

residuals = model.resid
fitted_values = model.fittedvalues

plt.scatter(fitted_values, residuals)
```

```
## <matplotlib.collections.PathCollection object at 0×000001AB986C0D88>
```

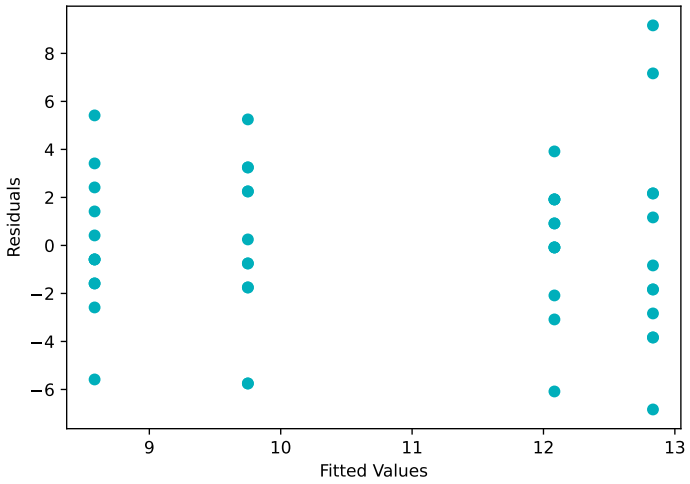
```
plt.xlabel('Fitted Values')
```

```
## Text(0.5, 0, 'Fitted Values')
```

```
plt.ylabel('Residuals')
```

```
## Text(0, 0.5, 'Residuals')
```

```
plt.show()
```




```
from scipy.stats import levene

two_anova_data['residuals'] = residuals
two_anova_data['fitted_values'] = fitted_values

x1 = two_anova_data.iloc[:12,3]
x2 = two_anova_data.iloc[12:24,3]
x3 = two_anova_data.iloc[24:36,3]
x4 = two_anova_data.iloc[36:,3]

levene(x1,x2,x3,x4)
```

```
## LeveneResult(statistic=1.1635486527826802, pvalue=0.33441312522437444)
```

Scale-Location plot

- 잔차들이 그룹 변수에 따라 고르게 분포 되었는가?

- 등분산 가정 체크: 빨간 수평선, 잔차들이 (패턴 없이) 무작위로 퍼져있어야 함.
- 잔차들을 표준화 후 기호를 없앰.

1번이랑 뭐가 다른가?

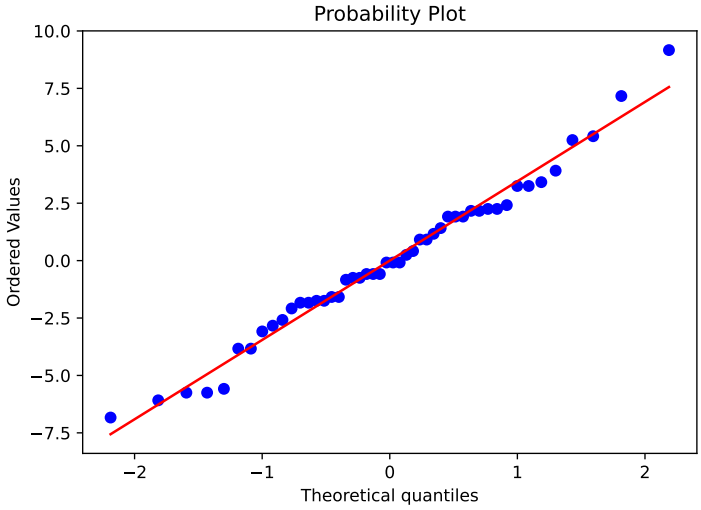
- 똑같음. 다만 잔차가 X 축을 따라서 불균형하게 분포 되어있을때, Residual vs. Fitted 보다 이상한 점을 쉬운 경우 존재.

정규성 가정 확인

```
from scipy import stats

residuals = model.resid
stats.probplot(residuals, plot=plt)
```

```
## ((array([-2.18794508, -1.81466696, -1.5940389 , -1.43152593, -1.29991017,
##         -1.18761792, -1.08858668, -0.99921942, -0.91719469, -0.84091983,
##         -0.76924975, -0.7013297 , -0.63650166, -0.57424545, -0.51414026,
##         -0.45583845, -0.39904728, -0.34351563, -0.28902427, -0.23537844,
##         -0.18240202, -0.12993297, -0.07781945, -0.02591656,  0.02591656,
##          0.07781945,  0.12993297,  0.18240202,  0.23537844,  0.28902427,
##          0.34351563,  0.39904728,  0.45583845,  0.51414026,  0.57424545,
##          0.63650166,  0.7013297 ,  0.76924975,  0.84091983,  0.91719469,
##          0.99921942,  1.08858668,  1.18761792,  1.29991017,  1.43152593,
##          1.5940389 ,  1.81466696,  2.18794508]), array([-6.83333333, -6.08333333, ↩
-5.75      , -5.75      , -5.58333333,
##         -3.83333333, -3.83333333, -3.08333333, -2.83333333, -2.58333333,
##         -2.08333333, -1.83333333, -1.83333333, -1.75      , -1.75      ,
##         -1.58333333, -1.58333333, -0.83333333, -0.75      , -0.75      ,
##         -0.58333333, -0.58333333, -0.58333333, -0.08333333, -0.08333333,
##         -0.08333333,  0.25      ,  0.41666667,  0.91666667,  0.91666667,
##          1.16666667,  1.41666667,  1.91666667,  1.91666667,  1.91666667,
##          2.16666667,  2.16666667,  2.25      ,  2.25      ,  2.41666667,
##          3.25      ,  3.25      ,  3.41666667,  3.91666667,  5.25      ,
##          5.41666667,  7.16666667,  9.16666667])), (3.454475594607066, ↩
-7.038151455032605e-15, 0.9904205983651462))
```

```
plt.show()
```




```
import scipy.stats as stats

residuals = model.resid
stats.shapiro(residuals)
```

```
## ShapiroResult(statistic=0.9820642471313477, pvalue=0.6665181517601013)
```

사후 분석 방법

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey = pairwise_tukeyhsd(endog=two_anova_data['response'],
                          groups=two_anova_data['ad_count'],
                          alpha=0.05)
print(tukey)
```

```
## Multiple Comparison of Means - Tukey HSD, FWER=0.05
## ====================================================
## group1 group2 meandiff p-adj   lower   upper  reject
## ----------------------------------------------------
##     1회     3회  -3.2917 0.0019 -5.3069 -1.2764   True
## ----------------------------------------------------
```

Unbalanced design Two-way ANOVA

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

```
model = ols('response ~ ad_count*ad_length', data=two_anova_data).fit()
table = sm.stats.anova_lm(model, typ=2)
print(table)
```

```
##                         sum_sq    df          F    PR(>F)
## ad_count            130.020833   1.0  10.560068  0.002218
## ad_length             0.520833   1.0   0.042301  0.837995
## ad_count:ad_length   11.020833   1.0   0.895093  0.349268
## Residual            541.750000  44.0        NaN       NaN
```

.footnote[Langsrud, Ø. (2003). ANOVA for unbalanced data: Use Type II instead of Type III sums of squares. Statistics and Computing, 13(2), 163-167.]

6.5 연습문제

자동차 연비 문제

mpg 데이터 셋에는 자동차 연비를 나타내는 mpg 변수가 있다. 데이터 안의 차들은 cylinders 변수를 기준으로 4개, 6개, 8개가 들어있는 그룹으로 분류할 수 있다. 자동차의 실린더 수가 늘어나면 보통 연비가 줄어든다는 이야기가 있다. 데이터에서 이러한 경향을 보이는지 유의수준 5% 하에서 검정해 보세요.

```
import seaborn as sns

mpg = sns.load_dataset('mpg')
selected_mpg = mpg[['mpg', 'cylinders']]

print(selected_mpg.head())
```

```
##     mpg  cylinders
## 0  18.0          8
## 1  15.0          8
## 2  18.0          8
## 3  16.0          8
## 4  17.0          8
```

펭귄의 부리길이

palmerpenguins 패키지의 penguins 데이터에는 펭귄 종류별 부리길이 (bill_length_mm) 정보가 들어있다. 펭귄의 종류에 따라서 부리길이가 다르다고 할 수 있는지 유의수준 1% 하에서 검정해보세요.

```
import seaborn as sns
penguins = sns.load_dataset('penguins')
selected_penguins = penguins[['species', 'bill_length_mm']]

print(selected_penguins.head())
```

```
##   species  bill_length_mm
## 0  Adelie            39.1
## 1  Adelie            39.5
## 2  Adelie            40.3
## 3  Adelie             NaN
## 4  Adelie            36.7
```

드릴 도구 검정 절차 (Drilling process)

`drilling-tool-exam.csv` 데이터를 사용하여 다음 물음에 답하세요.

슬통 철공 회사에서는 5개의 브랜드의 드릴 소재를 사용하여 철판에 구멍을 뚫은 작업을 하고 있다. 회사 제품은 2.5cm 직경을 가진 철판인데, 브랜드 별 사용하는 소재들이 미세하게 달라, 직경에 차이가 발생하는지 알아보고자 한다. 품질 팀은 제품 품질을 유지하기 위하여 온도의 영향도 조사하기로 하였다. 슬통이는 한 명의 품질 관리사에게 각기 다른 온도에서 브랜드별 드릴을 무작위로 20개씩 선택하여 뚫은 구멍의 직경을 측정하도록 하였다.

1) 데이터를 사용 각 브랜드별, 온도별 평균과 표준편차 정리 표를 만드세요.
2) 브랜드별, 온도별로 ANOVA main effect & interaction plot을 그려보세요.
3) Two-way ANOVA 분석을 진행해 주세요.

제 7 장

회귀분석의 이해

7.1 단순 선형회귀 (Simple linear regression)

데이터를 나타내는 기호들

- 반응변수 $y_i, i = 1, ..., n$
- 독립변수 $x_i, i = 1, ..., n$
- 잡음변수 $e_i, i = 1, ..., n$

$$\mathrm{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}, X\beta = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \cdots & \cdots \\ 1 & x_{n-1} \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, e = \begin{pmatrix} e_1 \\ e_2 \\ \cdots \\ e_n \end{pmatrix}$$

모델 가정

반응변수의 관찰값들은 다음과 같은 모델을 통해서 발생되었다고 가정한다.

$$\underline{y} \sim \mathcal{N}(X\beta, \sigma^2 I)$$

- 관찰값 y_i은 독립변수 x_i와 잡음 e_i의 선형결합으로 이루어져있다.

$$y_i = \beta_0 + \beta_1 x_i + e_i, \quad i = 1, ..., n$$

- 잡음 e_i들의 분포는 평균이 0이고 분산이 σ^2인 독립인 정규분포를 따른다고 가정
- 원래 잡음 분포에 대한 가정이 없었으나, 추후에 추가 됨.

Best fitting line

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T y$$

회귀분석 계수 추정을 하는 방법은 다음의 슬기로운 통계생활 영상을 참고하자.

- 영상1

- 영상2

```
# 파이썬 코드
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris
iris = load_iris()
iris = pd.DataFrame(data=iris.data, columns=iris.feature_names)
iris.columns = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width'] #컬럼명 변경시

x = iris['Petal_Width']
y = iris['Petal_Length']

data_X = np.column_stack((np.ones(len(x)), x))
beta = np.linalg.solve(np.dot(data_X.T, data_X), np.dot(data_X.T, y))

plt.scatter(x, y, color='blue')
```

```
## <matplotlib.collections.PathCollection object at 0×000001AB9878D248>
```

```
plt.xlabel("Width")
```

```
## Text(0.5, 0, 'Width')
```

```
plt.ylabel("Length")
```

```
## Text(0, 0.5, 'Length')
```

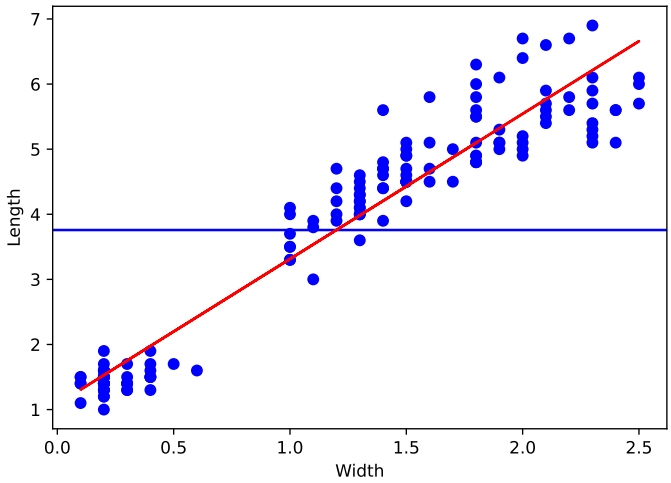
```
plt.axhline(y=np.mean(y), color='blue', linestyle='-')
```

```
## <matplotlib.lines.Line2D object at 0×000001ABA23EB508>
```

```
plt.plot(x, np.dot(data_X, beta), color='red', linestyle='-')
```

```
## [<matplotlib.lines.Line2D object at 0×000001ABA23EB448>]
```

```
plt.show()
```



- 파란색 선의 의미: 독립변수 X의 정보가 없었다면 반응 변수 y에 대한 최적 추정치는 y의 표본 평균을 사용해서 예측
- 빨간색 선의 의미: 독립변수 X의 정보가 추가된 최적의 예측 직선

파이썬에서 회귀분석

```
from statsmodels.formula.api import ols

model = ols("Petal_Length ~ Petal_Width", data=iris).fit()
model.summary()
```

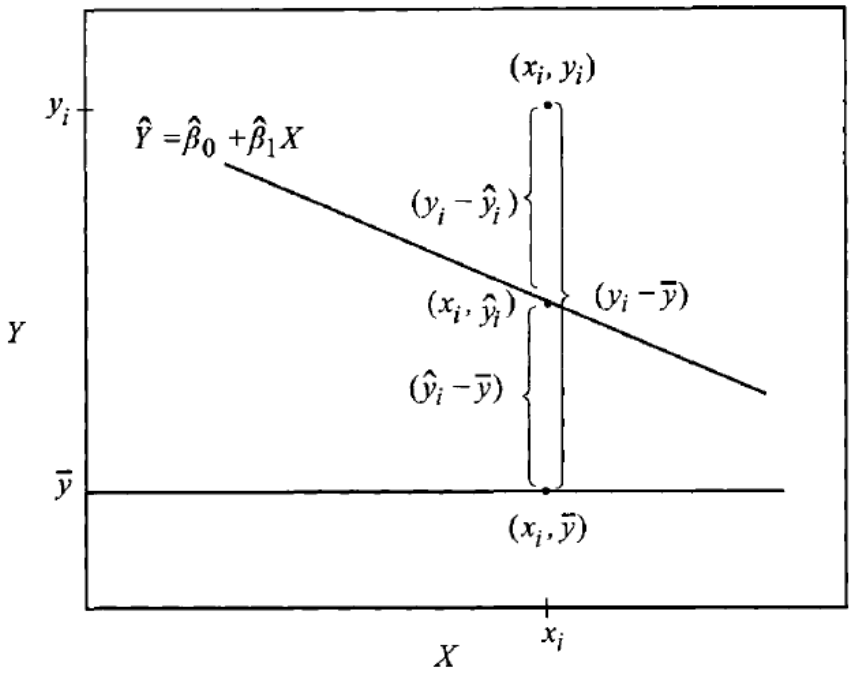
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:           Petal_Length   R-squared:                       0.927
## Model:                            OLS   Adj. R-squared:                  0.927
## Method:                 Least Squares   F-statistic:                     1882.
## Date:                토, 28 10 2023   Prob (F-statistic):          4.68e-86
## Time:                        14:01:32   Log-Likelihood:                -101.18
## No. Observations:                 150   AIC:                             206.4
## Df Residuals:                     148   BIC:                             212.4
## Df Model:                           1
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
```

```
## ------------------------------------------------------------------------------
## Intercept       1.0836      0.073     14.850      0.000       0.939       1.228
## Petal_Width     2.2299      0.051     43.387      0.000       2.128       2.332
## ==============================================================================
## Omnibus:                        2.438   Durbin-Watson:                   1.430
## Prob(Omnibus):                  0.295   Jarque-Bera (JB):                1.966
## Skew:                           0.211   Prob(JB):                        0.374
## Kurtosis:                       3.369   Cond. No.                         3.70
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
## """
```

계수에 따른 모델식

$$Petal\ Length = 1.08356 + 2.22994 \times Petal\ Width$$

회귀분석의 효용성 측정 지표들



SST, SSM, and SSE

붓꽃의 꽃잎 길이 **아무런 추가 정보가 없을 경우** 평균 ($\bar{y}$) 으로 예측할 것이다. 이렇게 반응변수 평균값 ($\bar{y}$) 에서 각 관측치들까지의 변동성의 제곱합은 다음과 같이 분해 할 수 있음.

$$SST = SSR + SSE$$

- 관측치들의 편차 제곱합 (SST): $\sum(y_i - \bar{y})^2$
  - 붓꽃의 꽃잎 너비 정보(X_1)를 사용하여 회귀모델을 수립하여 예측 ($\hat{y}_i$)
- 예측치의 오차들의 제곱합(SSE): $\sum(y_i - \hat{y}_i)^2$
- 향상된 예측력들의 제곱합(SSR): $\sum(\hat{y}_i - \bar{y})^2$
  - 회귀모델을 사용 예측함으로서 향상된 예측력 ($\hat{y}_i - \bar{y}$)

7.2 회귀분석 ANOVA

귀무가설 vs. 대립가설

- H_0: 모든 회귀계수들이 0이다. $\beta_1 = 0$
- H_A: 0이 아닌 회귀계수가 존재한다. $\beta_1 \neq 0$

검정통계량

$$F = \frac{SSR/1}{SSE/(n-2)} \sim F_{1,n-2}$$

- ANOVA에서 그룹별 평균이 다르다고 결론 내리는 논리와 동일함.
- 회귀분석을 통해서 향상된 예측 효과 (분자 부분)가 모델의 잡음보다 훨씬 크다면, 회귀분석 모델의 효과가 통계적으로 의미가 있다고 판단한다.
- 독립변수를 고려하는 것이 정말 효과가 있는지를 검정한다.
- 설명력이 많이 좋아졌는지를 체크!

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols("Petal_Length ~ Petal_Width", data=iris).fit()
sm.stats.anova_lm(model)
```

```
##                 df      sum_sq     mean_sq            F        PR(>F)
## Petal_Width    1.0  430.480647  430.480647  1882.452368  4.675004e-86
## Residual     148.0   33.844753    0.228681          NaN           NaN
```

- 유의수준 5% 하에서 F value (1882.5)와 대응하는 p-value 2.2^-16을 고려할 때, 너무 작으므로, 귀무가설을 기각한다.

알아두면 좋은 분포 정보

자유도가 a인 t 분포를 따르는 확률변수를 제곱하면 자유도가 $1, a$인 F 분포를 따르게 된다.

$$(t_{n-2})^2 = F_{1,n-2}$$

회귀분석의 성능측정 지표 - R^2 vs. adjusted R^2

R^2 - 회귀 직선의 성능은 얼마나 좋아?

- 회귀직선으로 인하여 향상된 예측력이 전체 관측치 변동성에서 차지하는 비율

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

- R^2 해석: R^2 만큼의 데이터 변동성이 독립변수에 의하여 설명됨
- 주의: Adjusted R^2는 R^2처럼 해석하면 안 됨.

Adjusted R^2 - 모델 복잡도를 고려한 지표

- 모델에 들어있는 독립 변수 갯수 p가 많아지면 R^2은 항상 높아지는 경향을 보인다.
- 변수 갯수가 다른 모델간 적합도를 비교하려 한다면 adjusted R^2를 사용하여 비교한다.

$$R_a^2 = 1 - \frac{SSE/(n-p-1)}{SST/(n-1)} = 1 - \frac{n-1}{n-p-1}(1-R^2)$$

R^2에 대한 오해

- 단순선형회귀에서만 R^2와 표본 상관계수 r이 같음.

$$r_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

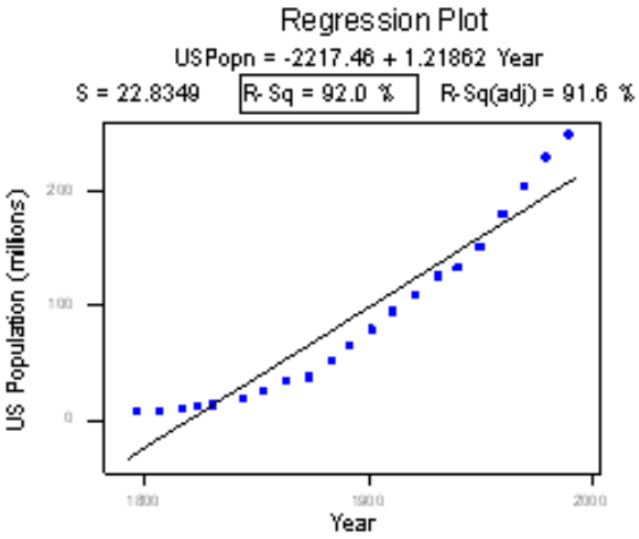
- R^2 값만을 가지고 판단할 수 없는 이유 (잔차 그래프)
- R^2 값은 선형적인 모델이 데이터 변동성에 대하여 얼마나 설명력을 가지고 있는가를 나타낸다.
- R^2 값이 높다는 것이 항상 회귀분석 모델이 데이터에 잘 적합 (fitting) 되어 있다는 의미가 아니다.

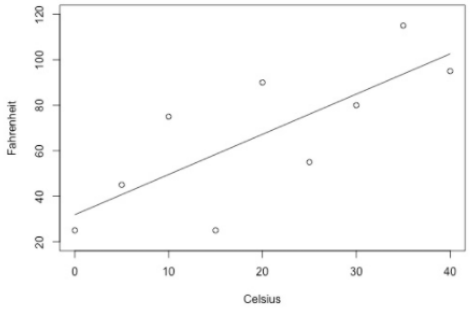
회귀모델 잡음의 분산 (Regression standard error)

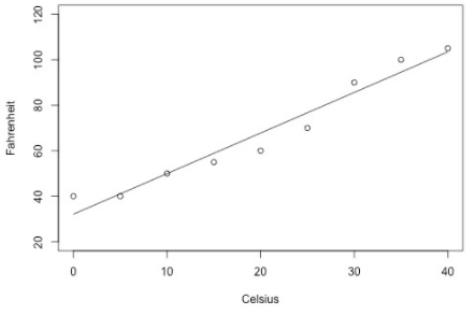
다음 중 어느 회귀 직선의 예측값이 더 정확할까?

- 잡음을 발생시키는 분포의 분산 (σ^2)을 다음의 값을 통하여 추정한다.

$$\hat{\sigma}^2 = MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n-p-1}$$







7.3 다중 선형회귀 (Multiple linear regression)

단순 선형 회귀분석에서 독립변수의 숫자가 p개로 늘어난 모형이다.

- 기본 가정은 동일함.

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + e_i, \quad i = 1, ..., n$$

위의 형태를 다음과 같이 벡터 형태로 표현 할 수 있음.

$$y_i = \mathrm{x}_i^T \beta + e_i, \quad i = 1, ..., n$$

$$X = \begin{bmatrix} 1 & x_{11} & ... & x_{1p} \\ 1 & x_{21} & ... & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & ... & x_{np} \end{bmatrix} \quad \beta = \begin{pmatrix} \beta_1 \\ ... \\ \beta_p \end{pmatrix}$$

R에서 다중회귀분석 실행하기

꽃잎의 길이를 세가지 독립변수들 (꽃잎 너비, 꽃받침 길이와 너비)을 사용하여 설명하는 회귀분석 모델을 만들어보자.

```
model2 = ols("Petal_Length ~ Petal_Width + Sepal_Length + Sepal_Width",
             data=iris).fit()
model2.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:           Petal_Length   R-squared:                       0.968
## Model:                            OLS   Adj. R-squared:                  0.967
## Method:                 Least Squares   F-statistic:                     1473.
## Date:                토, 28 10 2023   Prob (F-statistic):          6.98e-109
## Time:                        14:01:33   Log-Likelihood:                -39.408
## No. Observations:                 150   AIC:                             86.82
## Df Residuals:                     146   BIC:                             98.86
## Df Model:                           3
## Covariance Type:            nonrobust
## ===============================================================================
##                   coef    std err          t      P>|t|      [0.025      0.975]
## -------------------------------------------------------------------------------
## Intercept       -0.2627      0.297     -0.883      0.379      -0.850       0.325
## Petal_Width      1.4468      0.068     21.399      0.000       1.313       1.580
## Sepal_Length     0.7291      0.058     12.502      0.000       0.614       0.844
## Sepal_Width     -0.6460      0.068     -9.431      0.000      -0.781      -0.511
## ==============================================================================
## Omnibus:                        2.520   Durbin-Watson:                   1.783
## Prob(Omnibus):                  0.284   Jarque-Bera (JB):                2.391
## Skew:                           0.073   Prob(JB):                        0.303
## Kurtosis:                       3.601   Cond. No.                         79.3
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
## """
```

모델 비교하기 - Model 1 vs. Model 2

독립변수 2개 추가 효용성이 있는지를 검정해보자.

- 모델 1: 독립변수 1개 - `Petal.Width`
- 모델 2: 독립변수 3개 - `Petal.Width + Sepal.Length + Sepal.Width`

귀무가설 vs. 대립가설

H_0: Reduced Model이 알맞음.
H_A: Full Model이 알맞음.

Full model vs. Reduced model

- 같은 구조를 가지고 있는 모델 중 한 모델이 다른 모델을 포함하는 형식의 2 모델을 비교
  - Full model: `Petal.Width + Sepal.Length + Sepal.Width`
  - Reduced model: `Petal.Width`

F-검정

- 두 모델의 오차제곱합 차이를 비교

$$F = \frac{\left[\left(SSE\left(RM\right)\right) - SSE\left(FM\right)\right] / \left(p + 1 - k\right)}{SSE\left(FM\right) / \left(n - p - 1\right)} \sim F_{(p+1-k, n-p-1)}$$

- Full 모델의 독립변수 갯수 p
- Reduced 모델의 모수 갯수 (Intercept 포함) k

```
# 파이썬 코드
import statsmodels.api as sm
from statsmodels.formula.api import ols

model1 = ols('Petal_Length ~ Petal_Width', data=iris).fit() #mod1
model2 = ols('Petal_Length ~ Petal_Width + Sepal_Length + Sepal_Width',
             data=iris).fit() #mod2

table = sm.stats.anova_lm(model1, model2) #anova
print(table)
```

```
##    df_resid        ssr  df_diff    ss_diff          F        Pr(>F)
## 0     148.0  33.844753      0.0        NaN        NaN           NaN
## 1     146.0  14.852948      2.0  18.991805  93.341859  7.752746e-27
```

F 검정 통계량과 p-value로 미루어보아 귀무가설을 기각할 수 있으며, Full model을 선택할 수 있음.

```
model2.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:           Petal_Length   R-squared:                       0.968
## Model:                            OLS   Adj. R-squared:                  0.967
## Method:                 Least Squares   F-statistic:                     1473.
## Date:                토, 28 10 2023   Prob (F-statistic):          6.98e-109
## Time:                        14:01:35   Log-Likelihood:                -39.408
## No. Observations:                 150   AIC:                             86.82
## Df Residuals:                     146   BIC:                             98.86
## Df Model:                           3
## Covariance Type:            nonrobust
## ===============================================================================
##                   coef    std err          t      P>|t|      [0.025      0.975]
## -------------------------------------------------------------------------------
## Intercept       -0.2627      0.297     -0.883      0.379      -0.850       0.325
## Petal_Width      1.4468      0.068     21.399      0.000       1.313       1.580
## Sepal_Length     0.7291      0.058     12.502      0.000       0.614       0.844
## Sepal_Width     -0.6460      0.068     -9.431      0.000      -0.781      -0.511
## ==============================================================================
## Omnibus:                        2.520   Durbin-Watson:                   1.783
## Prob(Omnibus):                  0.284   Jarque-Bera (JB):                2.391
## Skew:                           0.073   Prob(JB):                        0.303
## Kurtosis:                       3.601   Cond. No.                         79.3
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
## """
```

각 독립변수들의 계수들이 0이 아닌 값을 갖는 것이 통계적으로 유의하다고 판단할 수 있음.

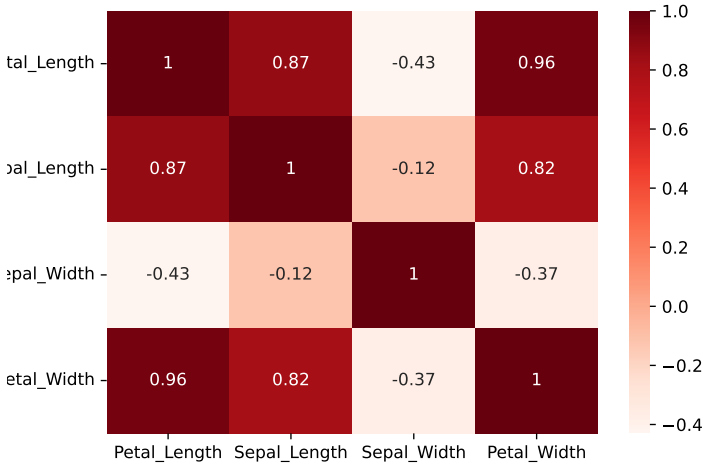
모델 진단하기

1변수 그래프: Histogram, Box plot

주요 체크 사항 - 각 변수들 중 skew한 분포가 없는지 확인

2변수 그래프: Correlation plot

```
# 파이썬 코드
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

cols = ["Petal_Length", "Sepal_Length", "Sepal_Width", "Petal_Width"]
corr_mat = iris[cols].corr().round(2)

sns.heatmap(corr_mat, annot=True, cmap=plt.cm.Reds);
plt.show()
```



잔차 그래프와 검정

잔차의 정규성과 잔차의 등분산성 체크

- 정규성
  - Anderson-Darling Test 혹은 Shapiro-Wilk Test를 실시한다.
- 잔차 등분산성 검정
  - 어떻게 체크할까? F test, Bartlett 검정 혹은 Levene 검정?
  - 잔차를 그룹을 나눠서 체크 할 카테고리 변수 존재하지 않음.

```
# 파이썬 코드
import scipy.stats as stats

residuals = model2.resid
fitted_values = model2.fittedvalues
```

```
plt.figure(figsize=(15,4))
```

```
## <Figure size 1500×400 with 0 Axes>
```

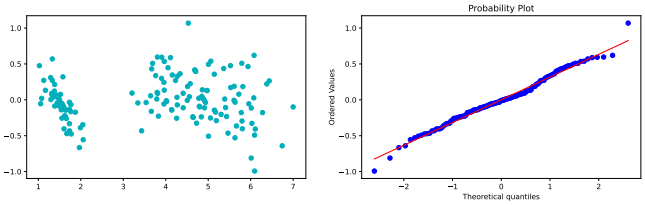
```
plt.subplot(1,2,1)
```

```
## <AxesSubplot:>
```

```
plt.scatter(fitted_values, residuals);

plt.subplot(1,2,2)
```

```
## <AxesSubplot:>
```

```
stats.probplot(residuals, plot=plt);
plt.show()
```



Breusch–Pagan / Cook–Weisberg 검정

- 아이디어: 우리의 잔차가 등분산을 갖는다는 의미는 독립변수에 의하여 설명이 안된다는 의미
- Breusch–Pagan 선형성
  - H_0: 모든 계수 0
  - H_A: 0이 아닌 계수 존재

귀무가설 하에서 검정통계량은 카이제곱분포 p를 따름.

$$\hat{r}_i^2/\hat{\sigma}^2 = \delta_0 + \delta_1 X_1 + ... + \delta_p X_p + noise$$

```
from statsmodels.stats.diagnostic import het_breuschpagan

model = ols('Petal_Length ~ Petal_Width + Sepal_Length + Sepal_Width', data=iris).fit()
bptest = het_breuschpagan(model.resid, model.model.exog)

print('BP-test statistics: ', bptest[0])
```

```
## BP-test statistics:  6.039114919618932
```

```
print('p-value: ', bptest[1])
```

```
## p-value:  0.10972262962330982
```

오차 독립성

특정 패턴을 띄지 않는지, 분산이 변하지는 않는지를 체크한다.

- Durbin-Watson test 실시
  - 귀무가설: 잔차들간의 상관성이 존재하지 않는다.
  - 대립가설: 잔차들간의 자기 상관성이 존재한다.

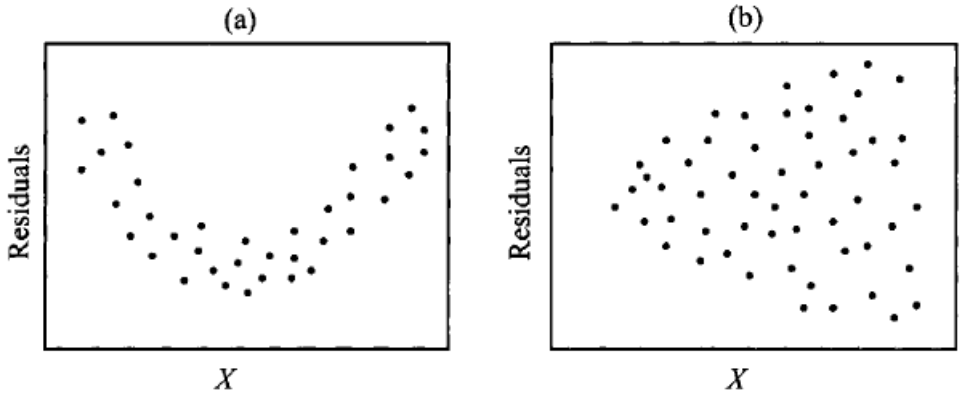


그림 7.1: (왼쪽) 비선형성을 설명할 수 있는 모형을 고려 vs. (오른쪽) 등분산성을 만족하지 못하는 경우

```
from statsmodels.stats.stattools import durbin_watson

dw_stat = durbin_watson(model2.resid)
print(dw_stat)
```

```
## 1.782966528592163
```

7.4 연습문제

`palmerpenguins` 패키지에는 남극 Palmer station에서 관측한 펭귄 정보들이 포함된 데이터이다.

```
import pandas as pd
import numpy as np
from palmerpenguins import load_penguins

penguins = load_penguins()
print(penguins.head())
```

```
##   species     island  bill_length_mm  ...  body_mass_g     sex  year
## 0  Adelie  Torgersen            39.1  ...       3750.0    male  2007
## 1  Adelie  Torgersen            39.5  ...       3800.0  female  2007
## 2  Adelie  Torgersen            40.3  ...       3250.0  female  2007
## 3  Adelie  Torgersen             NaN  ...          NaN     NaN  2007
## 4  Adelie  Torgersen            36.7  ...       3450.0  female  2007
##
## [5 rows x 8 columns]
```

```
np.random.seed(2022)
train_index = np.random.choice(penguins.shape[0], 200)
```

1) `train_index` 를 사용하여 펭귄 데이터에서 인덱스에 대응하는 표본들을 뽑아서 `train_data`를 만드세요. (단, 결측치가 있는 경우 제거)
2) `train_data`의 펭귄 부리길이 (bill_length_mm)를 부리 깊이 (bill_depth_mm)를 사용하여 산점도를 그려보세요.
3) 펭귄 부리길이 (bill_length_mm)를 부리 깊이 (bill_depth_mm)의 상관계수를 구하고, 두 변수 사이에 유의미한 상관성이 존재하는지 검정해보세요.
4) 펭귄 부리길이 (bill_length_mm)를 부리 깊이 (bill_depth_mm)를 사용하여 설명하는 회귀 모델을 적합시킨 후 2번의 산점도에 회귀 직선을 나타내 보세요. (모델 1)
5) 적합된 회귀 모델이 통계적으로 유의한지 판단해보세요.
6) R^2 값을 구한 후 의미를 해석해 보세요.
7) 적합된 회귀 모델의 계수를 해석해 보세요.
8) 1번에서 적합한 회귀 모델에 새로운 변수 (종 - species) 변수를 추가하려고 합니다. 성별 변수 정보를 사용하여 점 색깔을 다르게 시각화 한 후 적합된 모델의 회귀 직선을 시각화 해보세요. (모델 2)
9) 종 변수가 새로 추가된 모델 2가 모델 1 보다 더 좋은 모델이라는 근거를 제시하세요.
10) 모델 2의 계수에 대한 검정과 그 의미를 해석해 보세요.
11) 모델 2 에 잔차 그래프를 그리고, 회귀모델 가정을 만족하는지 검증을 수행해주세요.
12) 모델 2 의 잔차를 통하여 영향점, 혹은 이상치의 유무를 판단해보세요.

제 8 장

회귀분석 파고들기

8.1 Best 모델을 찾아서

Stepwise 방법 처럼 자동으로 베스트 모델을 찾아주는 알고리즘은 현재 통계학에서는 사용하지 않는 것을 추천하고 있다. 대표적이 이유는 다음과 같다.

- 결과 모델의 R-squared 값이 높게 측정되는 경향성을 띈다.
- 결과 모델의 예측에 대한 신뢰구간이 비정상적으로 좁게 나오는 모델이 선택된다.
- 결과 모델의 선택된 변수의 계수들이 비정상적으로 높게 나오는 경향을 띈다.
- 다중공선성이 문제가 되는 데이터의 경우 stepwise 방법은 잘 작동하지 않는다.

위의 문제에 대한 것은 다음의 참고 자료 를 찾아보자.

Stepwise 방법

- 각 단계별로 변수를 하나씩 추가해나가면서 최적의 모델을 찾는 방법
  - 모델 평가 지표들 (p-value, AIC, BIC, Adjusted R^2, Residual Mean Square) 중 하나를 선택하여 설정한 기준값을 비교하여 모델을 선택함.
  - 예를 들어 p-value를 기준으로 한다면 α_{enter} 와 α_{remove} 를 먼저 설정해 놓고, 각 변수의 p-value와 기준을 비교해서 모델에 넣을지 뺄지 결정함.
  - 보통 α_{enter} 와 α_{remove} 값으로 0.15를 선택한다. (R의 경우)

1 단계

- 각 변수별 회귀모델 설정 $\rightarrow$ 모델 중 α_{enter} 보다 낮은 p-value 중 가장 낮은 변수를 선택

2 단계

- 1차 선택 변수에 나머지 변수들 끼워서 회귀모델 설정
  - 같은 방법으로 2번째 변수 선택 $\rightarrow$ 조건 불만족시 stop.
  - 이미 선택된 변수 중 α_{remove} 보다 높은 p-value $\rightarrow$ 제거.

현재 파이썬의 경우 p-value 기반 stepwise는 구현된 것이 없고, AIC 기반 만 구현되어있다.

AIC, BIC base stepwise method

AIC, BIC는 무엇일까?

- Likelihood function 기반 모델 적합도 평가 지표

$$AIC = -2logL + 2p$$
$$BIC = -2logL + p\,log\,n$$

일반적으로 Likelihood 값은 높을 수록 모델의 적합도가 높다고 판단할 수 있다. 하지만, 모델에 사용되는 변수가 늘어날 수록 Likelihood 값이 높아지는 경향을 보인다.

- 모델에 사용되는 변수는 적으면 적을 수록 좋다. (같은 성능이면 모델 복잡성 낮은 모델을 선호)
- 특정 변수가 추가되었는데 늘어나는 Likelihood 값이 미미하다면 추가하지 않는 것이 좋다.
- AIC, BIC는 Likelihood 값에 음수가 붙어있으므로 같이 낮을 수록 좋음!

AIC stepwise in Python

Stepwise 방법은 가능한 모든 모델의 평가지표를 비교하는 방법이 아니다.

예를 들어 p개의 독립변수가 존재하면, 2^p개의 모델에 대한 AIC값을 다 비교해야 하지만 계산량이 너무 많아진다.

- Stepwise 방식은 최적의 모델을 효율적으로 찾아가는 방법이라고 이해하면 된다.
- 단, 단점은 stepwise의 결과 모델이 최적의 모델이라는 것을 장담할 수 없다.

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
import statsmodels.api as sm
from sklearn.datasets import load_iris

# Load iris data
iris = load_iris()
X = iris.data[:, [0, 1, 3]]
y = iris.data[:, 2]
names = np.array(iris.feature_names)[[0, 1, 3]]

# Define model
lr = LinearRegression()

# Define custom feature selector
def aic_score(estimator, X, y):
    X = sm.add_constant(X)
```

```
    model = sm.OLS(y, X).fit()
    print("Model AIC:", model.aic)
    return -model.aic

# Perform SFS
sfs = SFS(lr,
          k_features=(1,3),
          forward=True,
          scoring=aic_score,
          cv=0)
sfs.fit(X, y)

# 결과 출력
```

```
## Model AIC: 385.1349961458542
## Model AIC: 568.7536581517621
## Model AIC: 206.35386357991936
## Model AIC: 156.17922963502622
## Model AIC: 193.9940121409602
## Model AIC: 86.81602018114893
## SequentialFeatureSelector(cv=0, estimator=LinearRegression(), k_features=(1, 3),
##                           scoring=<function aic_score at 0×000001ABA3F9F0D8>)
```

```
print('Selected features:', np.array(names)[list(sfs.k_feature_idx_)])
```

```
## Selected features: ['sepal length (cm)' 'sepal width (cm)' 'petal width (cm)']
```

mlxtend패키지에서 제공되는 SequentialFeatureSelector 는 순차적으로 변수를 넣어가면서 모델을 평가하는 기능을 제공한다. 다만 평가 지표에 AIC나 BIC같은 지표를 제공하지 않기 때문에 사용자가 직접 정의하여 사용해야한다.

- 방향은 2방향이 가능: forward, backward
  - forward 옵션을 False로 설정할 경우 backward 방향이 작동한다.

Mallows's C_p statistic

Mallow's C_p의 값을 이해하기 위해서 모델의 전체 모수 갯수를 p, 그것들의 특정한 부분 집합인 k개의 변수들을 생각하자. 여기서 p는 각 독립변수에 딸려있는 계수 $p-1$개와 절편 1개를 포함한 값임에 주의하자.

Mallow's C_p 모델 평가 지표이며, p개의 모수를 사용한 모델로 적합된 값에 대한 MSE(mean square error)와 관련한 통계량이다.

$$C_k = \frac{SSE_k}{\hat{\sigma}^2} - (n - 2k)$$

여기서 $\hat{\sigma}^2$은 전체 모수 p개를 모두 사용하여 추정된 σ^2의 추정값이고, SSE_k는 모수 k개를 사용한 모델에서 계산된 SSE 값을 의미한다. 위의 Mallow의 통계량은 기댓값을 계산하기 위해서 다음의 사실을 받아들이도록 하자.

$$\mathbb{E}\left[SSE_k\right] = \sigma^2\left(n-k\right)$$

위의 사실을 이용하면, C_k의 기댓값은 k라는 것을 알 수 있으며, 이론적으로 C_k의 최소값은 k라는 것이 알려져있다. 따라서, 일반적으로 C_k 값은 작은 값이 좋다고 할 수 있으며, k 값 근처에 있는 모델을 좋은 모델 후보로 선정한다. 또한, Full 모델의 C_p 값은 언제나 p이 나오게 설계되어있다. 따라서 Full 모델을 C_p 값으로 평가해서는 안된다.

C_p 값이 작게 나오는 경우

위의 계산식에서 σ^2은 Full 모델의 MSE를 통하여 추정하게 되는데, 이 값이 잘 작동하려면 Full 모델의 계수들이 모두 유효해야한다. 만약 Full 모델에서의 계수들이 0 근처인 변수들이 많다면, $\hat{\sigma}^2$은 실제 값보다 크게 추정되는 경향을 보인다. 이 경우 C_p 값은 작아지게되고, 유용하지 않는 값이 된다.

Mallow's C_p in Python

Mallow's C_p를 Python에서 구현 하는 함수는 현재 존재하지 않는다. 다음과 같이 사용자 정의 함수를 사용하도록 하자.

```
import numpy as np
import pandas as pd
import itertools
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm

def calculate_mallows_cp(X, y):

    X = sm.add_constant(X)
    lr_full = LinearRegression().fit(X, y)
    y_pred_full = lr_full.predict(X)
    k = X.shape[1]
    n = X.shape[0]

    mse_full =  1/(n-k) * np.sum((y - y_pred_full) ** 2)
    cp_results = []

    # Loop over all possible combinations
    for p in range(1, k+1):
        for subset in itertools.combinations(range(k), p):
            X_subset = X[:, subset]
```

```
            # Define and fit subset model
            lr_subset = LinearRegression().fit(X_subset, y)
            y_pred_subset = lr_subset.predict(X_subset)

            # Calculate SSE of subset model
            sse_subset = np.sum((y - y_pred_subset) ** 2)

            # Calculate Mallow's Cp
            cp = (sse_subset / mse_full) - n + 2 * p

            # Store results
            cp_results.append((subset, cp))

    return cp_results
```

iris 데이터셋을 가져와 C_p 값을 계산해보도록 한다.

```
from sklearn.datasets import load_iris
iris = load_iris()

# 'Petal.Width', 'Sepal.Length', 'Sepal.Width'
X = iris.data[:, [0, 1, 3]]

# 'Petal.Length'
y = iris.data[:, [2]]
cp_results = calculate_mallows_cp(X, y)
df = pd.DataFrame(cp_results, columns=['Variables', 'Mallows Cp'])
df[df['Mallows Cp'] ⩽ 100]
```

```
##        Variables  Mallows Cp
## 8         (1, 3)   88.947328
## 11     (0, 1, 3)   90.947328
## 13     (1, 2, 3)    2.000000
## 14  (0, 1, 2, 3)    4.000000
```

예제 데이터와 모델 선택하기

다음은 시멘트 혼합물에 따른 발열 측정 자료이다.

- Y: heat evolved in calories per gram of cement
- X_1: tricalcium aluminate
- X_2: tricalcium silicate
- X_3: tetracalcium alumino ferrite

- X_4: dicalcium silicate

표 8.1: Hald cement data

Y	X1	X2	X3	X4
78.5	7	26	6	60
74.3	1	29	15	52
104.3	11	56	8	20
87.6	11	31	8	47
95.9	7	52	6	33
109.2	11	55	9	22
102.7	3	71	17	6
72.5	1	31	22	44
93.1	2	54	18	22
115.9	21	47	4	26
83.8	1	40	23	34
113.3	11	66	9	12
109.4	10	68	8	12

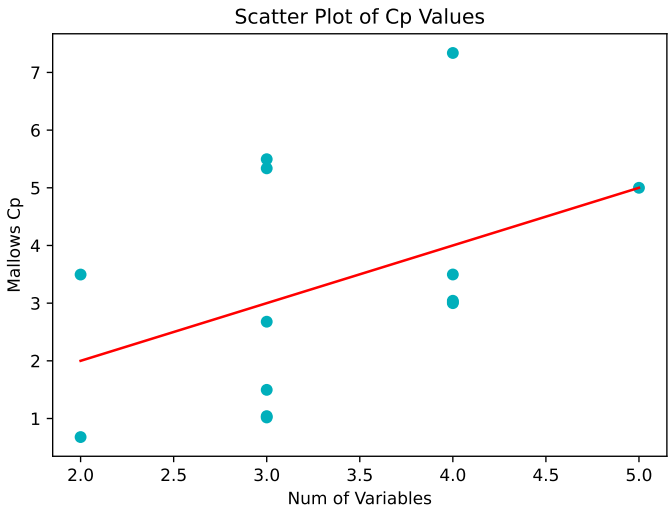
데이터를 `hald_cement_data` 변수에 입력한 후 Mallows C_p를 활용하여 모델 선택을 진행해보자.

```
y = np.array([78.5, 74.3, 104.3, 87.6, 95.9, 109.2, 102.7,
              72.5, 93.1, 115.9, 83.8, 113.3, 109.4])
X = np.array([[7, 26, 6, 60],
              [1, 29, 15, 52],
              [11, 56, 8, 20],
              [11, 31, 8, 47],
              [7, 52, 6, 33],
              [11, 55, 9, 22],
              [3, 71, 17, 6],
              [1, 31, 22, 44],
              [2, 54, 18, 22],
              [21, 47, 4, 26],
              [1, 40, 23, 34],
              [11, 66, 9, 12],
              [10, 68, 8, 12]])
cp_results = calculate_mallows_cp(X, y)
df = pd.DataFrame(cp_results, columns=['Variables', 'Mallows Cp'])
df[df['Mallows Cp'] ⩽ 20]
```

```
##           Variables  Mallows Cp
## 9            (1, 2)    0.678242
## 11           (1, 4)    3.495851
## 15        (0, 1, 2)    2.678242
## 17        (0, 1, 4)    5.495851
```

```
## 21        (1, 2, 3)    1.041280
## 22        (1, 2, 4)    1.018233
## 23        (1, 3, 4)    1.496824
## 24        (2, 3, 4)    5.337474
## 25     (0, 1, 2, 3)    3.041280
## 26     (0, 1, 2, 4)    3.018233
## 27     (0, 1, 3, 4)    3.496824
## 28     (0, 2, 3, 4)    7.337474
## 29     (1, 2, 3, 4)    3.000000
## 30  (0, 1, 2, 3, 4)    5.000000
```

```
import matplotlib.pyplot as plt

# Filter Cp values ⩽ 20
filtered_df = df[df['Mallows Cp'] ⩽ 20]
num_variables = [len(variables) for variables in filtered_df['Variables']]

# Scatter plot
plt.scatter(num_variables, filtered_df['Mallows Cp']);
plt.plot(num_variables, num_variables,
         color='red', label='y = x');
plt.xlabel('Num of Variables');
plt.ylabel('Mallows Cp');
plt.title('Scatter Plot of Cp Values');
plt.show()
```



Mallows's C_p 값이 p 직선과 가까운 모델들을 후보로 선정한다.

- $X_1 + X_2$ 모델
- $X_1 + X_2 + X_4$ 모델

- $X_1 + X_2 + X_3$ 모델
- $X_1 + X_3 + X_4$ 모델

이 모델들 중에서 가장 작은 C_p 값을 갖는 $X_1 + X_2$ 모델이 최적의 모델로 생각할 수 있다.

- Adjusted R^2 구하기

```
from sklearn.metrics import r2_score

def adjusted_r2_score(estimator, X, y):
    y_pred = estimator.predict(X)
    n = X.shape[0]
    p = X.shape[1]
    r2 = r2_score(y, y_pred)
    adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
    return adjusted_r2

# Perform SFS
sfs = SFS(lr,
          k_features=(1,4),
          forward=True,
          scoring=adjusted_r2_score,
          cv=0)
sfs.fit(X, y)

# 결과 출력
```

```
## SequentialFeatureSelector(cv=0, estimator=LinearRegression(), k_features=(1, 4),
##                           scoring=<function adjusted_r2_score at 0×000001ABA4075DC8>)
```

```
selected_indices = list(sfs.k_feature_idx_)
print('Selected features:', selected_indices)
```

```
## Selected features: [0, 1, 3]
```

Adjusted R^2 값을 기준으로 보았을 경우 최적의 모델은 $x_1 + x_2 + x_4$ 모델로 생각할 수 있다.

- stepwise 알고리즘을 적용한 모델 선택

stepwise 알고리즘으로 intercept만 존재하는 모델에서 시작해서 모든 변수 조합을 기준으로 최적 모델을 구해보면 다음과 같다.

```
# Perform SFS
sfs = SFS(lr,
          k_features=(1,4),
```

```
          forward=True,
          scoring=aic_score,
          cv=0)
sfs.fit(X, y)

# 결과 출력
```

```
## Model AIC: 100.41187201391966
## Model AIC: 96.0703964203777
## Model AIC: 105.95980439471697
## Model AIC: 95.74404477885616
## Model AIC: 65.63410626724041
## Model AIC: 97.52172751297775
## Model AIC: 76.74498580894803
## Model AIC: 61.86628547186261
## Model AIC: 62.619952232581554
## Model AIC: 63.83668979165169
## SequentialFeatureSelector(cv=0, estimator=LinearRegression(), k_features=(1, 4),
##                           scoring=<function aic_score at 0×000001ABA3F9F0D8>)
```



```
selected_indices = list(sfs.k_feature_idx_)
print('Selected features:', selected_indices)
```

```
## Selected features: [0, 1, 3]
```

이렇게 모델을 비교하다보면 각 모델의 측정 지표에 따라서 최적의 모델이 달라지는 경우가 있다. 이런 경우 분석가가 꼭 명심해야 하는 사실은 다음과 같다.

- 각 측정 지표를 통해서 뽑은 최적의 모델은 정답 모델이 아니다.

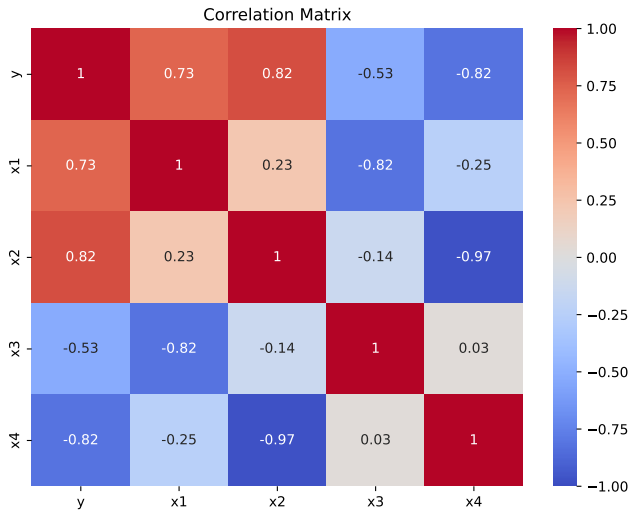
참고로 stepwise 방법은 현재 통계학에서는 거의 사용하지 않는 모델 선택 방법으로는 좋지 않은 방법 중 하나로 꼽힌다. 따라서 결정된 최적 모델들은 분석가가 다시 면밀하게 분석하고, 최종 모델을 결정해야 한다.

예를 들어, Mallows's C_p 기준으로 뽑힌 $x_1 + x_2$ 모델과 Adjusted R^2 기준과 stepwise 알고리즘으로 선택된 $x_1 + x_2 + x_4$ 모델을 판단 과정을 살펴보자.

분석 시 변수 간 상관관계를 계산해본다면 x_2와 x_4가 강한 상관관계를 갖는다는 사실을 알 수 있다.

```
import seaborn as sns

df = pd.DataFrame(np.concatenate((y.reshape(-1, 1), X), axis=1), columns=['y', 'x1', 'x2', 'x3', 'x
corr_mat = df.corr().round(2)

plt.figure(figsize=(8, 6));
```

```
sns.heatmap(corr_mat, annot=True, cmap='coolwarm', vmin=-1, vmax=1);
plt.title('Correlation Matrix');
plt.show()
```



위의 상관계수 테이블에 의하면, 두 변수가 같이 들어간 $x_1 + x_2 + x_4$ 모델의 경우 다중공선성(Multicollinearity)이 우려될 수 있으며, 이는 다음 시간에 배울 다중공선성 측정 지표인 VIF에 의하여도 잡아낼 수 도 있다.

이러한 사실을 바탕으로 $x_1 + x_2 + x_4$ 모델보다는 $x_1 + x_2$ 모델이 좀 더 좋은 판단이라고 분석가는 최종 결정을 내릴 수 있다.

8.2 회귀분석에서의 영향점 (influential point)

회귀 모델을 설정함에 있어서 영향력이 있는 표본들이 존재한다. 이들을 잡아낼 수 있는 지표들을 배워보자.

- Cook's distance
- 스튜던트화 잔차

Cook's distance

- **PRESS** 잔차: 만약 특정 포인트의 정보가 없어진다면 모델의 예측이 얼마나 바뀔 것 인가?

$$C_i = \frac{\sum_{j=1}^{n} \left(\hat{y}_j - \hat{y}_{j(i)} \right)}{\hat{\sigma}^2 \, (p+1)}$$

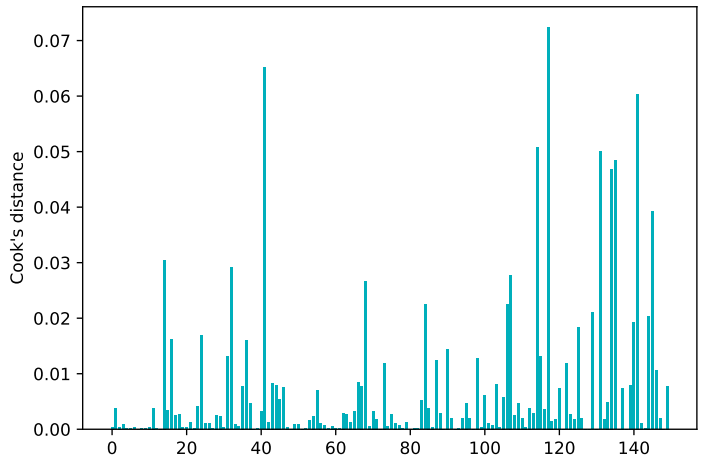
즉, 각 표본 별 Cook's distance가 존재함.

- C_i 값 $\geq F_{(p+1, n-p-1)}$ 분포의 중앙값을 사용하여 Influential point로 판정

```
from scipy.stats import f
f.ppf(0.5, 4, 150 - 5)
```

```
## 0.8431114434390823
```

```
from statsmodels.formula.api import ols
from statsmodels.stats.outliers_influence import OLSInfluence
from sklearn.datasets import load_iris

iris = load_iris()
iris = pd.DataFrame(data=iris.data, columns=iris.feature_names)

#컬럼명 변경
iris.columns = ['Sepal_Length','Sepal_Width','Petal_Length','Petal_Width']
model2 = ols('Petal_Length ~ Petal_Width + Sepal_Length + Sepal_Width', data=iris).fit()

influence = OLSInfluence(model2)
cooks_distance = influence.cooks_distance[0]

# Plot the Cook's distance
plt.bar(range(len(cooks_distance)), cooks_distance, width=0.8);
plt.ylabel("Cook's distance");
plt.show()
```



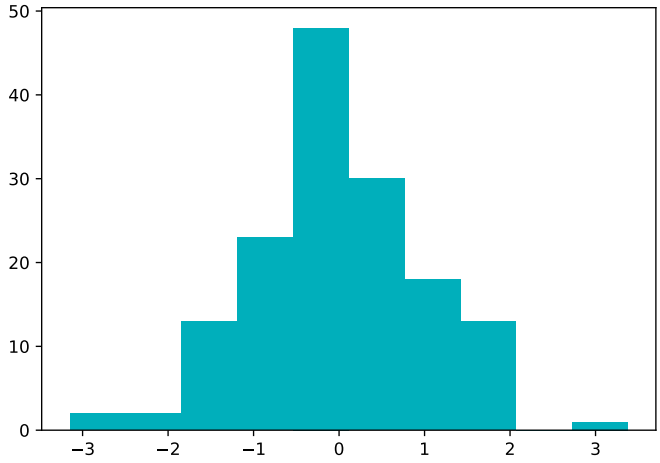
표준화된 잔차 (Standardized Residuals)

잔차들을 표준화 시키면 표준정규분포를 따를 것이다. 표준 정규분포의 66-95-99 규칙을 이용해서 3 표준편차 영역 밖의 표본들을 걸러낸다.

```
std_res = influence.resid_studentized_internal
print(std_res[np.abs(std_res) > 3])
```

```
## 134    3.379260
## 141   -3.151953
## dtype: float64
```

```
plt.hist(std_res);
plt.show()
```



스튜던트화 잔차 (Studentized Residuals)

그냥 잔차가 아닌 Cook's distance 계산시의 잔차들을 표준화 한 후, 3 표준편차 영역 밖에 위치한 표본들을 걸러낸다.

$$t_i = \frac{d_i}{std\left(d_i\right)}$$

where $d_i = y_i - \hat{y}_{(i)}$

- $\hat{y}_{(i)}$: i 번째 표본을 제외하고 회귀 모델을 구한 후 i 번째 표본에 대하여 예측한 값
- i 번째 표본이 influential point 라면 회귀직선이 표본 i 쪽으로 기울어져 있을 수 있다. 따라서 i 번째 표본을 제외 했을 때의 회귀직선을 사용해서 잔차를 구하는 것이 더 좋을 경우가 있다.

```
stud_res = influence.resid_studentized_external
stud_res[np.abs(stud_res) > 3]
```

```
## 134    3.507635
## 141   -3.253796
## dtype: float64
```

Outliers vs. High leverage point

영향점 (influential point)을 구성하는 표본들은 두 가지로 분류된다.

- outlier
- high leverage point

회귀분석에서의 Outliers

- 반응변수 y에 대한 outlier를 의미한다.
- Studentized __residual__을 사용한 판단

회귀분석에서의 High leverage point

- 독립변수 X에 대한 outlier를 의미한다.
- leverage값이 높은 관측치
- `hatvalues()` 함수
- 2(p+1)/n 보다 크면 high
- 잔차값은 높지 않게 나올 수 있음

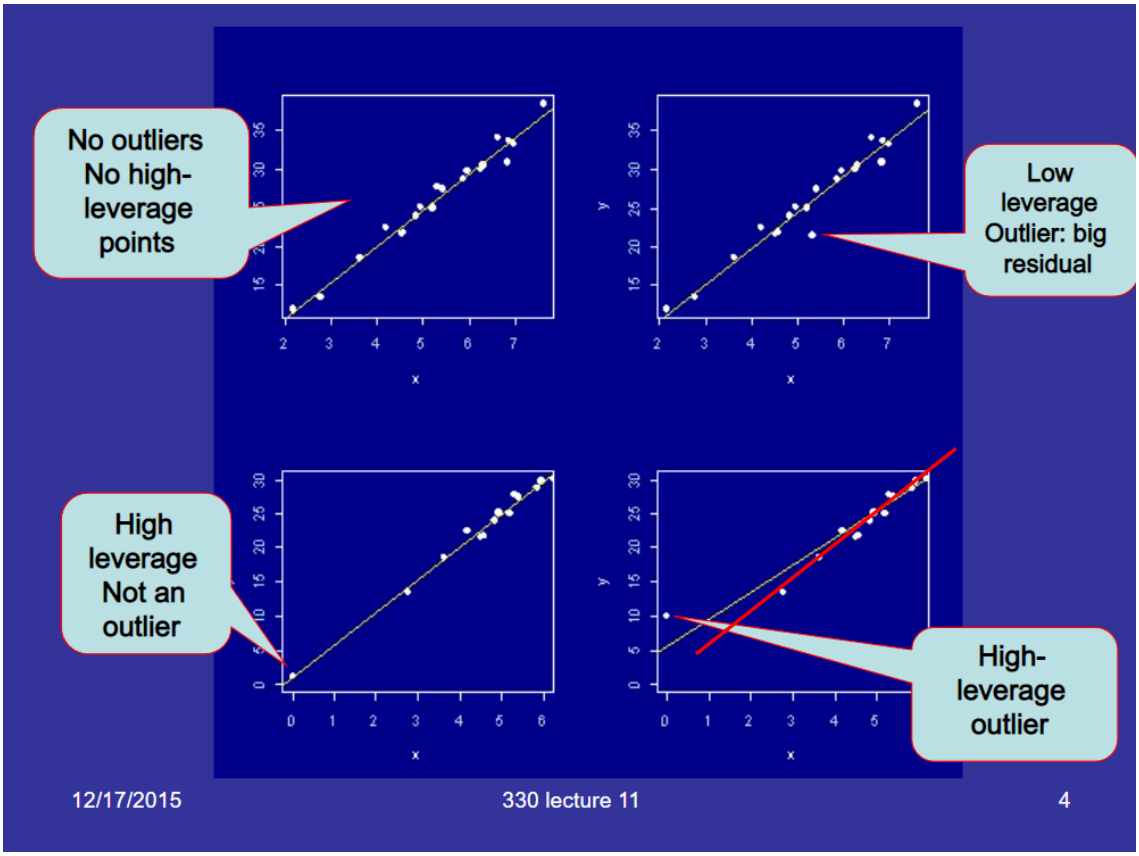


그림 8.1: Image from 'https://slideplayer.com/slide/8964135/'

8.3 영향있는 표본들, Outlier에 대처하는 방법

튀는 표본이 존재한다하여 무작정 삭제가 답이 아니다. 즉, Outlier Flag가 떳다고해서 꼭 Outlier라는 이야기가 아니다.

- 의미: 내가 가진 데이터의 패턴을 벗어난 자료라는 뜻

데이터를 자세히 살펴보고, 이상치로 판단하여 제거할 지, 영향력이 있는 표본으로 생각하여 분석에 포함시키는 문제는 분석가의 판단이다. (합리적인 근거를 마련한다.)

데이터 에러를 고려

- 데이터가 잘못 기록된 것인 아닌지 살펴본다.
- 표본이 연구 모집단을 반영하지 못한다고 생각하면 지운다. 다만, 삭제에 있어서 객관적인 이유가 존재해야 한다.
- Robust한 회귀분석 방법을 적용해본다.
  - 학습 시 weight를 변경: Weighted regression
  - 학습 시 손실함수를 절대거리 기반으로 변경: `ltsReg` or `lmrob`

```
from statsmodels.formula.api import ols
from statsmodels.robust.robust_linear_model import RLM

# Fit the robust linear model
robust_mod = RLM.from_formula(
    'Petal_Length ~ Petal_Width + Sepal_Length + Sepal_Width', data=iris).fit()
```

모델 에러를 고려

- 고려되는 현재 모델에 다른 중요한 변수가 빠진 것은 아닌지 체크한다.
  - 인터렉션 항 추가를 고려해본다.
  - 비선형 모델을 고려해본다.
  - 데이터 변환을 고려해본다.

Scottish Hill Racing 예제

스코틀랜드에 매해 열리는 언덕 달리기 경주 데이터를 사용하여 회귀분석 모델을 돌려보자.

데이터 불러오기

```
import pandas as pd

race_data = pd.read_csv('./data/scottish-hills-races.csv')
race_data.columns = ['hill_race', 'time', 'distance', 'climb']
race_data.head()
```

그림 8.2: Image from 'www.scottishdistancerunninghistory.scot'

```
##                     hill_race  time  distance  climb
## 0  Greenmantle New Year Dash     965       2.5    650
## 1                    Carnethy    2901       6.0   2500
## 2               Craig Dunain    2019       6.0    900
## 3                    Ben Rha    2736       7.5    800
## 4                  Ben Lomond    3736       8.0   3070
```

회귀분석 모델

마지막으로 완주한 경기 시간 time을 distance 변수와 climb 변수를 사용하여 설명하는 회귀분석 모델을 설정한다.

```
reg1 = ols('time ~ distance + climb', data=race_data).fit()
reg1.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                   time   R-squared:                       0.919
## Model:                            OLS   Adj. R-squared:                  0.914
## Method:                 Least Squares   F-statistic:                     181.7
## Date:                  토, 28 10 2023   Prob (F-statistic):           3.40e-18
## Time:                        14:01:49   Log-Likelihood:                -285.41
## No. Observations:                  35   AIC:                             576.8
## Df Residuals:                      32   BIC:                             581.5
## Df Model:                           2
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
```

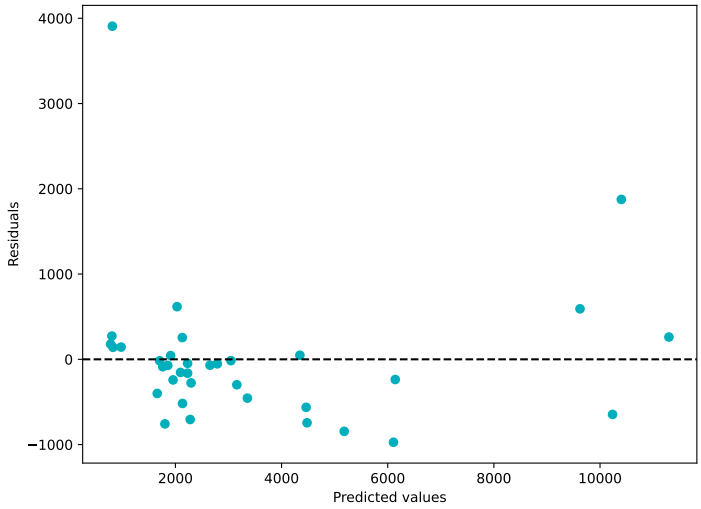
```
## Intercept   -539.4829    258.161     -2.090      0.045   -1065.339     -13.627
## distance     373.0727     36.068     10.343      0.000     299.604     446.542
## climb          0.6629      0.123      5.387      0.000       0.412       0.914
## ==============================================================================
## Omnibus:                       47.910   Durbin-Watson:                   2.249
## Prob(Omnibus):                  0.000   Jarque-Bera (JB):              233.983
## Skew:                           3.026   Prob(JB):                     1.55e-51
## Kurtosis:                      14.127   Cond. No.                     4.20e+03
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
## [2] The condition number is large, 4.2e+03. This might indicate that there are
## strong multicollinearity or other numerical problems.
## """
```

F-검정 통계량과 각 변수 t 검정 통계량값을 통하여 회귀분석 모델과 사용된 모든 변수가 통계적으로 유의한 것을 확인할 수 있다.

잔차 그래프 분석

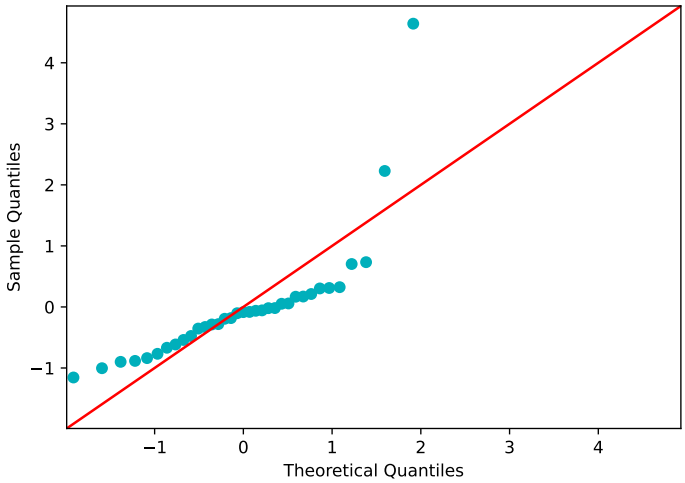
적합된 모델과 잔차를 확인해보자. (앞시간에 배운 정규성 검정과 등분산 검정, 독립성 검정은 다시 한번 연습해 볼 것.)

```
fig, ax = plt.subplots(figsize=(8,6))
ax.scatter(reg1.predict(), reg1.resid);
plt.axhline(y=0, color='black', linestyle='--');
plt.xlabel('Predicted values');
plt.ylabel('Residuals');
plt.show()
```



그래프로 보아 약간의 U 모양 그래프로 볼 수 있고, 또한 등분산 가정도 깨져보인다. (검정 필요함, 해보세요!)

```
sm.qqplot(reg1.resid, fit=True, line='45');
plt.show()
```

Normal QQ 그래프 경우 7번와 18번 표본을 제외하고는 다들 괜찮아 보인다.

8.4 다중공선성을 대하는 우리들의 자세

새로운 변수가 들어왔을 경우 가능한 4가지 상황

- 새로운 변수가 유의하지 않고, 기존 변수가 이전 값에 비해 크게 변하지 않음: 새로운 변수 추가 해서는 안됨.

- 새로운 변수가 **유의하고**, 기존 변수가 이전 값에 비해 크게 변하지 않음: 이상적 케이스
- 새로운 변수가 **유의하고**, 기존 변수가 이전 값에 비해 크게 변함: 새로운 변수 추가하고, collinearity 체크.
  - collinearity 증거가 없다. Add
  - collinearity 증거가 있다. Treat
- 새로운 변수가 유의하지 않고, 기존 변수가 이전 값에 비해 크게 변함: collinearity 확실한 증거
  - 변수 넣을지 말지에 대한 것 보다 treat 먼저 선행

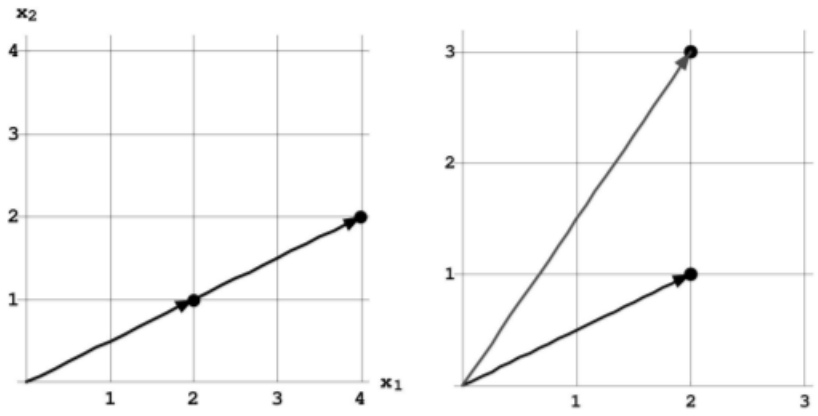
Collinearity (공선성)

- 많은 회귀분석의 경우 독립변수들은 orthogonality를 가정하지만 실제로 독립 변수들이 orthogonal 한 경우는 드물다. 이렇게 독립변수들끼리 correlation이 존재하는 상황을 공선성이 존재한다고 한다. 하지만 **독립변수 간 공선성의 존재는 실제로 분석에 영향을 줄 정도로 별 문제가 되지 않는다**.
- 가끔 너무 심하게 orthogonal 가정이 무너져서 분석에 영향을 줄 때가 있다.
  - 데이터가 살짝만 변해도 계수가 확 바뀜
  - 변수가 더하거나 지워질 때 계수가 확 바뀜

완전 linearly dependent한 경우

다음의 행렬의 열 들은 완벽하게 dependent한 경우이다. 즉, 각 열을 다른 두 열을 사용하여 상수배를 해서 더하고 빼면 만들어낼 수 있다.

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 3 & 6 & 9 \end{pmatrix}$$



공선성 처리 방법

다음 사항들을 머리속에 넣고, 읽어나가기 바란다.

- 공선성은 어떻게 회귀분석 추정과 예측에 영향을 미치는가
- 공선성을 어떻게 감지하고, 어떻게 해결할까?

다중공선성 감지 지표

VIF: Variance Inflation Factors

다중공선성이 심하다는 이야기는 특정 독립변수 X_j의 정보가 다른 독립 변수들의 정보로 모두 설명이 가능하다는 이야기가 된다. 따라서, X_j를 반응변수로 놓고, 다른 독립변수를 사용해서 회귀 모델을 적합해보면, R^2이 거의 1과 비슷하게 나올 것이다. 독립변수 X_j에 대한 VIF는 다음과 같이 계산한다.

$$VIF_j = \frac{1}{1 - R_j^2}$$

- R_j^2: j번째 독립 변수를 다른 독립변수들을 사용하여 회귀분석
- 독립변수 X_j가 다른 독립변수들과 선형적인 관계가 없는 이상적인 경우: VIF 값이 1
- 독립변수 X_j가 다른 독립변수들과 선형적인 관계가 심할 경우: VIF 값이 발산
- 판단 기준값: 10

왜 Variance Inflation Factors 인가?

- VIF가 계수의 변동성 증가분을 측정하기 때문

회귀분석에서 j 번째 변수의 계수의 분산 측정값은 다음과 같이 표현할 수 있음.

$$\widehat{\text{var}}(\hat{\beta}_j) = \frac{s^2}{(n-1)\widehat{\text{var}}(X_j)} \cdot \frac{1}{1 - R_j^2}$$

만약 j 번째 변수가 다른 변수들과 비교하여 선형 독립을 만족한다면, R^2 값은 0이 나옴. 따라서 VIF_j는 각 변수에 해당하는 계수의 분산값이 선형 독립인 경우 대비 얼마나 inflation 되었는지를 측정하는 지표로 볼 수 있다.

- $VIF_j = \frac{Var(\hat{\beta}_j)}{Var(\hat{\beta}_j)_{linearly\ indep.}}$

평균 VIF값의 의미

$$\frac{\sum_{j=1}^{p} VIF_j}{p} = \overline{VIF}$$

각 변수별 측정된 VIF값의 평균은 독립변수들이 선형 독립인 경우에 비하여 회귀분석 계수 추정치가 불안정한 정도를 나타내어 준다.

학생 성취도 데이터

학생 성취도 데이터를 통하여 이야기 해보자. 학생의 학업 성취도를 나타내는 `achv`를 가족환경 `fam`, 학교 친구들 `peer`, 그리고 학교 환경 `school`을 사용하여 설명하는 회귀 모델을 적합한다.

```
achieve_data = pd.read_csv('./data/student_achievement.csv')
reg2 = ols(formula='ACHV ~ FAM + PEER + SCHOOL', data=achieve_data).fit()
achieve_data.head()
```

```
##       ACHV      FAM     PEER   SCHOOL
## 0 -0.43148  0.60814  0.03509  0.16607
## 1  0.79969  0.79369  0.47924  0.53356
## 2 -0.92467 -0.82630 -0.61951 -0.78635
## 3 -2.19081 -1.25310 -1.21675 -1.04076
## 4 -2.84818  0.17399 -0.18517  0.14229
```

회귀분석 결과

회귀 모델이 유의한 지에 대한 F 검정통계량을 통하여 유의한 모델이라 말할 수 있다. 하지만 다음의 2가지 특징을 확인 할 수 있다.

- 각 변수들의 유의성 검정에서 p-value값이 높게 나오고, 모두 유의하지 않은 것을 알 수 있다.
- R^2 값이 0.2로 비교적 낮다고 할 수 있다.

```
reg2.summary()
```

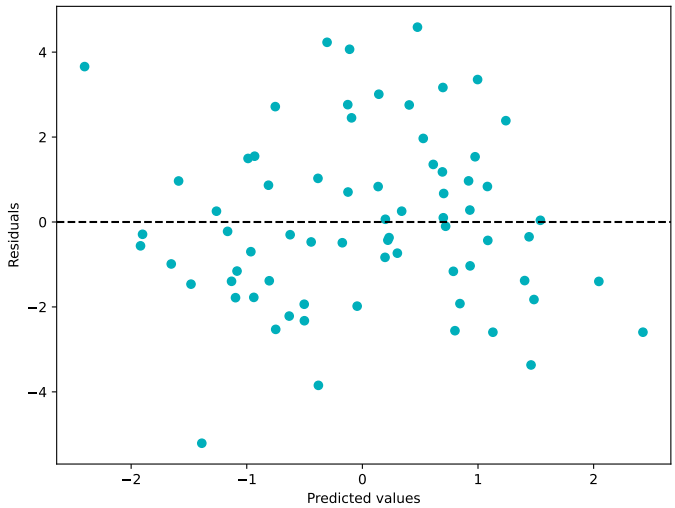
```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                   ACHV   R-squared:                       0.206
## Model:                            OLS   Adj. R-squared:                  0.170
## Method:                 Least Squares   F-statistic:                     5.717
## Date:                토, 28 10 2023   Prob (F-statistic):            0.00153
## Time:                        14:01:52   Log-Likelihood:                -148.20
## No. Observations:                  70   AIC:                             304.4
## Df Residuals:                      66   BIC:                             313.4
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept     -0.0700      0.251     -0.279      0.781      -0.570       0.430
## FAM            1.1013      1.411      0.781      0.438      -1.715       3.918
## PEER           2.3221      1.481      1.568      0.122      -0.635       5.280
## SCHOOL        -2.2810      2.220     -1.027      0.308      -6.714       2.152
## ==============================================================================
## Omnibus:                        0.558   Durbin-Watson:                   1.791
```

```
## Prob(Omnibus):                  0.756   Jarque-Bera (JB):                0.578
## Skew:                           0.203   Prob(JB):                        0.749
## Kurtosis:                       2.816   Cond. No.                         19.2
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
## """
```

잔차분석

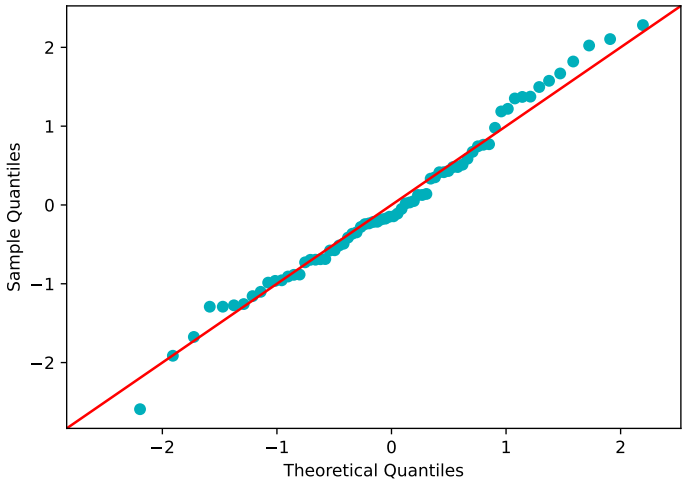
잔차 그래프를 통하여 분석을 해도 특이점이 나타나지 않을 수 있다.

```
fig, ax = plt.subplots(figsize=(8,6))
ax.scatter(reg2.predict(), reg2.resid);
plt.axhline(y=0, color='black', linestyle='--');
plt.xlabel('Predicted values');
plt.ylabel('Residuals');
plt.show()
```



QQ plot의 결과 역시 마찬가지 이다.

```
sm.qqplot(reg2.resid, fit=True, line='45');
plt.show()
```

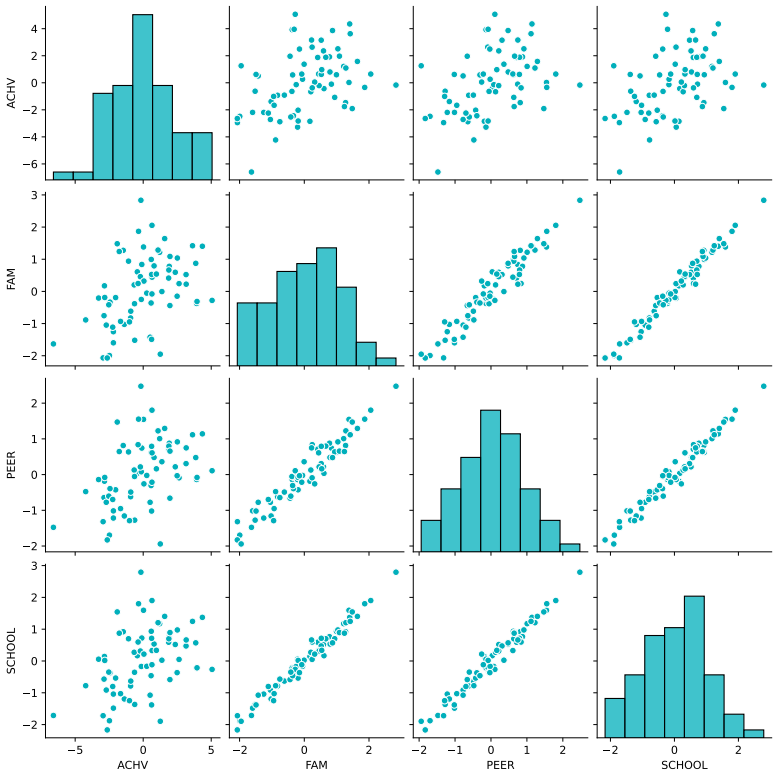


다중공선성 파악하기

각 변수들에 대하여 상관계수를 구해보도록 하자.

```
import seaborn as sns
sns.pairplot(achieve_data, kind="scatter", diag_kind="hist")
```

```
## <seaborn.axisgrid.PairGrid object at 0×000001ABA1F31EC8>
```



- 세 변수 fam, peer, 그리고 school의 산점도가 선형적인 패턴을 보이고 있다. 이는 전형적인 다중공선성 패턴이다.

```
achieve_data.corr()
```

```
##             ACHV       FAM      PEER    SCHOOL
## ACHV    1.000000  0.419459  0.439846  0.418101
## FAM     0.419459  1.000000  0.960081  0.985684
## PEER    0.439846  0.960081  1.000000  0.982160
## SCHOOL  0.418101  0.985684  0.982160  1.000000
```

- 각 변수의 VIF는 car 패키지의 vif() 함수를 사용한다.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

x = achieve_data.iloc[:,1:4]
vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(
    x.values, i) for i in range(x.shape[1])]
vif["features"] = x.columns
vif
```

```
##    VIF Factor features
## 0   37.139750      FAM
## 1   29.782169     PEER
## 2   81.339023   SCHOOL
```

결과에서 확인 할 수 있다시피 세 변수의 VIF 값은 모두 높다.

해결책

- 1 차원적 해결책은 VIF가 가장 높은 변수를 제외하고 회귀분석을 돌려보는 것이다.
  - VIF값이 안정될때까지 변수를 지워나감.
  - 지우는 것 만이 능사가 아님 - 지워진 변수에도 유용한 정보가 존재할 수 있기 때문이다.

다중 공선성의 해결책으로 언급되는 방법 중 대표적인 것은 다음과 같다.

- PCA (Principal Components Analysis)
- Penalized Regression

주성분 (Principal Components)의 이해

p개의 독립변수를 p개의 __선형 독립인 변수들__로 바꿔주는 방법이다.

준비단계 - 변수 표준화

주성분 분석을 적용하기 위해서 각 변수들을 scaling 해준다. (평균을 빼고, 표준편차로 나눠줌.)

- $X_1, X_2, ..., X_p$의 변수를 표준화시킨 변수를 $\tilde{X}, \tilde{X}_2, ..., \tilde{X}_p$라고 하자.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaled_data = scaler.fit_transform(achieve_data)
scaled_data = pd.DataFrame(scaled_data, columns=achieve_data.columns)
scaled_data.head(3)
```

```
##        ACHV       FAM      PEER    SCHOOL
## 0 -0.199734  0.519586 -0.012224  0.132025
## 1  0.345912  0.692129  0.471510  0.493656
## 2 -0.418312 -0.814296 -0.725165 -0.805211
```

각 변수들은 scaling 후 평균이 0, 표준편차가 1로 맞춰지게 된다.

- 평균 0

```
scaled_data.drop(columns=['ACHV']).mean(axis=0).round()
```

```
## FAM       0.0
## PEER      0.0
## SCHOOL    0.0
## dtype: float64
```

- 표준편차 1

```
scaled_data.drop(columns=['ACHV']).std(axis=0).round()
```

```
## FAM       1.0
## PEER      1.0
## SCHOOL    1.0
## dtype: float64
```

변환하기

표준화된 변수들을 사용하여 새로운 변수 $C_j, j = 1, ..., p$을 다음과 같이 만들어낸다.

$$C_j = v_{1j}\tilde{X}_1 + v_{2j}\tilde{X}_2 + ... + v_{pj}\tilde{X}_p$$

계수 역할을 하는 v들을 이리저리 조정하여 C_j들이 선형독립이 되도록 계수 v들을 결정하는 방법.

- 즉, C_j의 공분산 행렬은 대각 행렬이 됨.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=3)
scaled = scaled_data.drop(columns=['ACHV'])
pca_array = pca.fit_transform(scaled)

my_pca = pd.DataFrame(pca_array,
                      index = scaled.index,
                      columns=["pca1", "pca2", "pca3"])
```

결과값 해석

분산값 비교하기

- 원데이터 (Scaled)

```
scaled.head()
```

```
##         FAM      PEER    SCHOOL
## 0  0.519586 -0.012224  0.132025
## 1  0.692129  0.471510  0.493656
## 2 -0.814296 -0.725165 -0.805211
## 3 -1.211176 -1.375633 -1.055565
## 4  0.115871 -0.252115  0.108624
```

- 원 데이터의 공분산행렬

```
scaled_cov = scaled.cov()
scaled_cov
```

```
##              FAM      PEER    SCHOOL
## FAM     1.014493  0.973995  0.999969
## PEER    0.973995  1.014493  0.996394
## SCHOOL  0.999969  0.996394  1.014493
```

- 변환된 데이터

```
my_pca.head()
```

```
##        pca1      pca2      pca3
## 0  0.368955 -0.368726  0.124351
## 1  0.956635 -0.150422  0.085339
## 2 -1.353829  0.063735  0.019803
## 3 -2.102111 -0.129908 -0.194383
## 4 -0.015266 -0.268837 -0.127596
```

- 변환된 데이터의 공분산행렬

```
my_pca.cov().round(3)
```

```
##        pca1   pca2   pca3
## pca1  2.995 -0.000 -0.000
## pca2 -0.000  0.041  0.000
## pca3 -0.000  0.000  0.008
```

새로 만들어진 변수들의 분산값들은 pca.explained_variance_ 변수에 저장되어 있다.

```
pca.explained_variance_.round(3)
```

```
## array([2.995, 0.041, 0.008])
```

공분산 행렬의 분해

앞에서 살펴본 새로 만들어진 변수들의 분산 정보는 원 데이터 행렬의 공분산 행렬의 eigen values와 동일하다는 것을 알 수 있다.

```
from numpy import linalg
eig_values, eig_vectors = linalg.eig(scaled.cov())
eig_values
```

```
## array([2.99477567, 0.04062791, 0.00807469])
```

```
eig_vectors
```

```
## array([[-0.57613845, -0.67939712, -0.45440515],
##        [-0.5754361 ,  0.73197527, -0.36480886],
##        [-0.58046342, -0.05130072,  0.81266873]])
```

따라서 다음의 식이 성립한다.

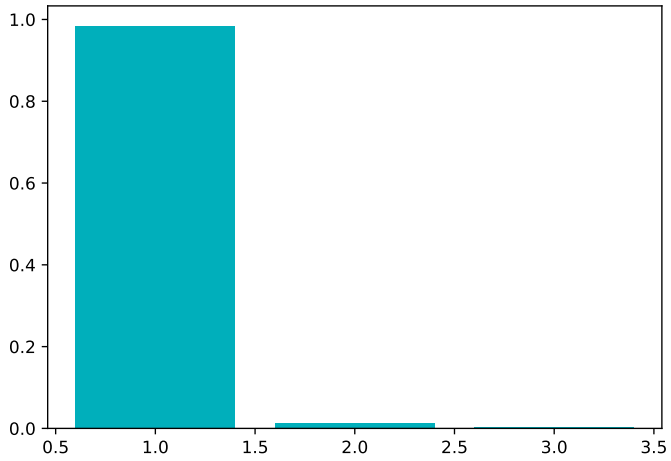
$$Var(C_j) = \lambda_j$$

λ_j는 원 데이터 행렬의 eigenvalues 중 j번째로 큰 값을 의미

주성분의 설명력 시각화

각 변수 별 분산을 순서대로 계산해서 바 그래프로 그려준다.

```
import matplotlib.pyplot as plt
plt.bar(range(1, 4), pca.explained_variance_ratio_);
plt.show()
```



- 각 주성분의 설명력을 시각화 해준다.

그래프를 해석해보면 첫번째 변수 C_1 가 전체 데이터 변동성의 90퍼센트가 넘는 부분을 설명하고 있다고 말할 수 있다.

다중공선성의 지표로서의 eigenvalue

Condition indices - κ_p

eigenvalue 들 중 유독 0에 가까운 작은 값이 존재한다면, 선형 독립화 과정에서 공선성이 존재하는 변수들의 정보가 하나의 성분으로 합쳐졌음을 의미한다.

- 이 사실을 이용해서 원 데이터에 다중 공선성이 존재하는지를 판단하는 기준을 만들 수 있다.
- 다중공선성이 존재해서 $lambda_p$ 값이 0과 아주 가까운 값을 갖게 된다면 왼쪽항이 발산하게 됨.

따라서, 우리가 가진 데이터에 다중공선성이 존재한다고 판단할 수 있다. 혹은 가장 큰 eigen value와 가장 작은 eigen value의 비율로서 측정할 수도 있는데, 이 경우 보통 15가 다중공선성이 있다고 판단하는 기준치가 된다.

$$\kappa_p = \sqrt{\frac{\lambda_1}{\lambda_p}} \geq 15$$

```
print(np.sqrt(eig_values[0] / eig_values[2]) >= 15)
```

```
## True
```

eigenvectors의 의미

앞선 공분산 행렬에 대한 eigen value decomposition의 결과에는 eigenvector들도 저장이 되어 있다.

```
eig_vectors
```

```
## array([[-0.57613845, -0.67939712, -0.45440515],
##        [-0.5754361 ,  0.73197527, -0.36480886],
##        [-0.58046342, -0.05130072,  0.81266873]])
```

이 정보는 주성분 분석으로 새로 만들어진 변수 C_j들이 어떻게 만들어졌는지에 대한 정보들을 담고 있다.

```
x_to_pc = pd.DataFrame(pca.components_,
                       columns=scaled.columns,
                       index=['pca1','pca2','pca3']).round(3)
x_to_pc
```

```
##         FAM   PEER  SCHOOL
## pca1  0.576  0.575   0.580
## pca2 -0.679  0.732  -0.051
## pca3  0.454  0.365  -0.813
```

단, pca 결과와와 eigen vector 행렬이 전치 (transpose) 관계, 그리고 부호가 다르다는 것을 유념하자. 이는 sklearn의 구현문제로 보이며, 결과적으로는 같은 행렬이어야만 한다. 즉, 위 행렬은 각 변수 C_j를 만들어내기 위해 사용된 표준화된 $\tilde{X}$에 부여된 다음과 같은 식의 가중치를 나타낸다.

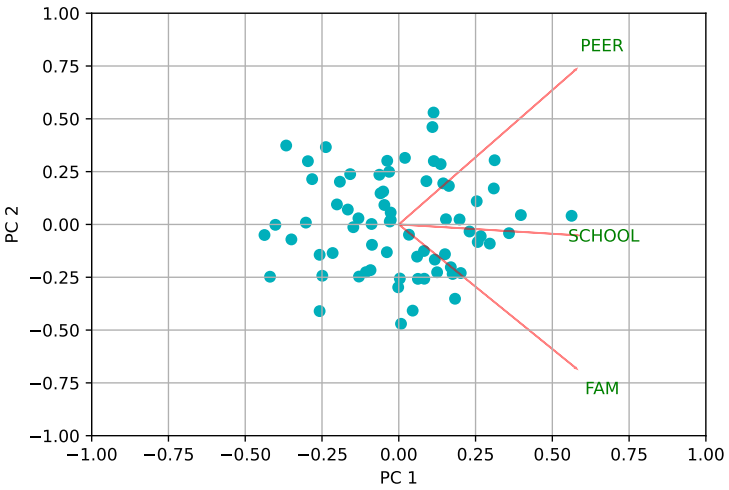
$$
\begin{aligned}
C_1 &= 0.576\tilde{X}_1 + 0.575\tilde{X}_2 + 0.580\tilde{X}_3 \\
C_2 &= -0.679\tilde{X}_1 + 0.732\tilde{X}_2 - 0.051\tilde{X}_3 \\
C_3 &= 0.454\tilde{X}_1 + 0.365\tilde{X}_2 - 0.813\tilde{X}_3
\end{aligned}
$$

8.5 Biplot을 사용한 변수 시각화

주성분 분석에서 각 변수의 기여도를 시각화 하는 biplot 그래프를 구현한 함수이다.

```
def biplot(score, coeff, pcax, pcay, labels=None):
  pca1=pcax-1
  pca2=pcay-1
  xs = score[:,pca1]
  ys = score[:,pca2]
  n=score.shape[1]
  scalex = 1.0/(xs.max()- xs.min())
  scaley = 1.0/(ys.max()- ys.min())
  plt.scatter(xs*scalex,ys*scaley)
  for i in range(n):
    plt.arrow(0, 0, coeff[pca1, i], coeff[pca2, i],color='r',alpha=0.5)
```

```
    if labels is None:
      plt.text(coeff[pca1, i]* 1.15, coeff[pca2, i] * 1.15,
               "Var"+str(i+1), color='g', ha='center', va='center')
    else:
      plt.text(coeff[pca1, i]* 1.15, coeff[pca2, i] * 1.15,
               labels[i], color='g', ha='center', va='center')
  plt.xlim(-1,1)
  plt.ylim(-1,1)
  plt.xlabel("PC 1")
  plt.ylabel("PC 2")
  plt.grid()

biplot(pca_array, pca.components_, 1, 2,
       labels=scaled.columns)
plt.show()
```



빨간선의 의미를 잘 해석할 수 있어야한다. 많이 헷갈려하고, 현직 교수님들도 헷갈려하는 부분이다. 잘 이해하고 있도록 한다.

- school 변수의 파란색 화살표가 가로축 (Dim 1)과 거의 일치하게 그려져있는 것을 볼 수 있는데, 이것은 rotation에 담겨있는 정보 중 school에 해당하는 3번째 줄 값들 중 PC1와 PC2에 해당하는 $0.580, -0.051$ 값을 나타내고 있다. 다음 행렬을 보면서, 각 화살표에 대한 해석 연습을 해보자.

```
x_to_pc
```

```
##          FAM   PEER  SCHOOL
## pca1   0.576  0.575   0.580
## pca2 -0.679  0.732  -0.051
## pca3   0.454  0.365  -0.813
```

즉, school 변수에 담긴 정보는 PC1을 만들때 상당한 기여를 했고, 반대로 PC2를 만들 때에는 기여도가 작았다고 해석할 수 있다.

Eigenvalue 값을 사용한 변수 관계 파악하기

알고 있어야 하는 상식 하나. 상수 (Constant) 의 분산은 0이다.

- 예: 숫자 3의 분산은 0이고, 기댓값은 3이다.
- $Var(C_j) = \lambda_j$, 만약 이 값이 0에 가깝다면? 상수라는 의미.
- $\mathbb{E}[C_j] = 0$, 왜냐하면 $\tilde{X}$ 들을 다 스케일링을 했기 때문이다.

```
min(pca.explained_variance_).round(3)
```

```
## 0.008
```

따라서, 다음과 같이 변수들 끼리의 관계를 파악해볼 수도 있다.

$$C_3 = 0.454\tilde{X}_1 + 0.365\tilde{X}_2 - 0.813\tilde{X}_3 \overset{set}{=} 0$$

주성분을 이용한 회귀분석

선형 독립을 강제로 만들었다면 회귀분석을 수행할 수 있다.

표준화된 회귀분석

표준화된 데이터 행렬을 사용해서 회귀분석을 수행했을 경우 구해지는 계수들을 θ라고 하자.

$$\tilde{Y} = \theta_1\tilde{X}_1 + \theta_2\tilde{X}_2 + \theta_3\tilde{X}_3 + \epsilon'$$

알파 계수 계산하기

선형 독립인 주성분들 PC1, PC2, PC3들을 사용해서 회귀분석을 돌렸을때 얻어지는 계수들을 $alpha$들이라고 하자.

$$\tilde{Y} = \alpha_1 C_1 + \alpha_2 C_2 + \alpha_3 C_3 + \epsilon'$$

우리 예제의 경우 α 계수들은 다음과 같이 구할 수 있다.

```
from sklearn.decomposition import PCA
from sklearn.linear_model import LinearRegression

reg3 = LinearRegression(fit_intercept=False)
reg3.fit(my_pca, scaled_data[['ACHV']])
```

```
## LinearRegression(fit_intercept=False)
```

```
reg3.coef_
```

```
## array([[0.2498183 , 0.38775709, 1.41806458]])
```

이렇게 구해진 PC1에 대한 계수들은 주성분들의 해석이 불가능하므로 사실상 해석이 불가능하게 된다. 물론, Biplot을 통하여 각 주성분에 대한 해석 아이디어를 얻을 수도 있겠지만, 엄밀하게 말하면 해석력이 떨어진다.

- 생각해 볼 거리: 왜 회귀분석 모델식에 `fit_intercept=False`은 왜 들어갔을까?

α 계수를 통해서 θ 계수 복구하기

- 구해진 α 계수들은 eigen vector들을 이용해서 θ 계수들로 복구 할 수 있다.

```
x_to_pc
```

```
##          FAM   PEER  SCHOOL
## pca1   0.576  0.575   0.580
## pca2 -0.679  0.732  -0.051
## pca3   0.454  0.365  -0.813
```

```
np.dot(reg3.coef_, x_to_pc)
```

```
## array([[ 0.52440959,  0.94507728, -1.0277675 ]])
```

θ 계수를 통해서 β 계수 복구하기

β 계수들은 표준화가 되기 전의 변수에 대한 계수들이다. 이 β 계수들 역시 다음의 식을 이용하여 구해진 θ 계수들로부터 다시 복구할 수 있다.

$$\hat{\beta}_j = \frac{s_y}{s_j}\hat{\theta}_j$$

$$\hat{\beta}_0 = \overline{y} - \sum_{j=1}^{3}\hat{\beta}_j$$

주성분 분석 복귀 예시

2개의 주성분 만을 이용하여 회귀분석 모델을 적합시켰다.

```
my_pca['y'] = scaled_data[['ACHV']]
my_pca.head()
```

```
##         pca1      pca2      pca3         y
## 0  0.368955 -0.368726  0.124351 -0.199734
## 1  0.956635 -0.150422  0.085339  0.345912
```

```
## 2 -1.353829  0.063735  0.019803 -0.418312
## 3 -2.102111 -0.129908 -0.194383 -0.979456
## 4 -0.015266 -0.268837 -0.127596 -1.270798
```

```
reg_pca = ols(formula='y ~ 0 + pca1 + pca2', data=my_pca).fit()
reg_pca.params.values
```

```
## array([0.2498183 , 0.38775709])
```

위에서 구해진 알파 계수들을 사용하여 θ 계수들을 구하자.

```
# theta coefficients
theta = np.dot(reg_pca.params.values, x_to_pc.iloc[:2,])
theta
```

```
## array([-0.11939173,  0.42748371,  0.125119  ])
```

$$\tilde{Y} = -0.12\tilde{X}_1 + 0.43\tilde{X}_2 + 0.12\tilde{X}_3 + \epsilon'$$

θ 계수들에서 β 계수들에 대한 값을 복구해보자.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

```
scale_info = np.array(scaler.scale_)
center_info = np.array(scaler.mean_)

beta = (scale_info[0] / scale_info[1:4]) * theta
beta_0 = center_info[0] - np.dot(center_info[1:4], beta)
beta = np.concatenate(([beta_0], beta))
beta
```

```
## array([-0.02595669, -0.2505051 ,  1.0505191 ,  0.27781196])
```

원래 데이터와 대응되는 회귀식은 다음과 같다.

$$\hat{y} = -0.026 - 0.25X_1 + 1.05X_2 + 0.28X_3$$

참고사항

현재 ADP 시험장에서 **제공되지 않는** `pca` 패키지를 사용하면, 다음과 같이 `biplot`을 그릴 수 있다. (단, 사용하기 위해서는 설치가 필요)

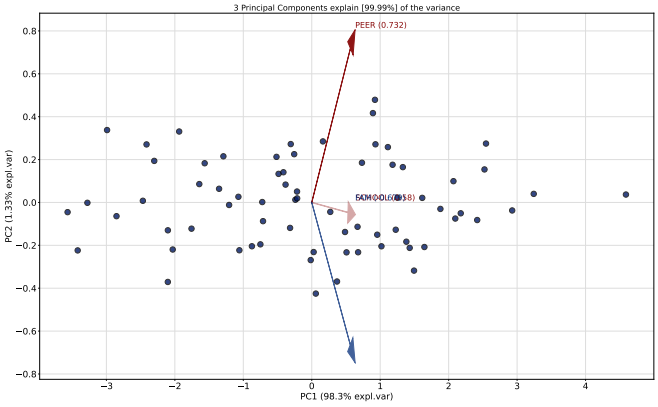
```
# 설치코드: !pip install pca
from pca import pca

model = pca(n_components=3, normalize=False);
model.fit_transform(scaled,
                    col_labels = scaled.columns);
```

```
## [pca] >Extracting row labels from dataframe.
## [pca] >The PCA reduction is performed on the [3] columns of the input dataframe.
## [pca] >Fit using PCA.
## [pca] >Compute loadings and PCs.
## [pca] >Compute explained variance.
## [pca] >Outlier detection using Hotelling T2 test with alpha=[0.05] and n_components=[3]
## [pca] >Multiple test correction applied for Hotelling T2 test: [fdr_bh]
## [pca] >Outlier detection using SPE/DmodX with n_std=[3]
```

```
model.biplot();
```

```
## [pca] >Plot PC1 vs PC2 with loadings.
##
## [scatterd] >INFO> Create scatterplot
```

```
plt.show()
```



- 위 코드에서 scaled 데이터를 사용하였기 때문에 normalize = False로 설정해주었다.

제 9 장

선형계획법 문제풀이

선형계획법 예제

슬통이는 두 가지 종류의 빵을 판매하는데, 초코빵을 만들기 위해서는 밀가루 100g과 초콜릿 10g이 필요하고 밀빵을 만들기 위해서는 밀가루 50g이 필요하다. 재료비를 제하고 초코빵을 팔면 100원이 남고 밀빵를 팔면 40원이 남는다. 오늘 슬통이는 밀가루 3000g과 초콜릿 100g을 재료로 갖고 있다. 만든 빵을 전부 팔 수 있고 더 이상 재료 공급을 받지 않는다고 가정한다면, 슬통이는 이익을 극대화하기 위해서 어떤 종류의 빵을 얼마나 만들어야 하는가?[1]

풀이

x_1 을 초코빵을 만드는 개수, x_2를 밀빵을 만드는 개수로 설정하자. 그렇다면, x_1, x_2는 정수값을 가져야하며, 다음과 같은 조건하에서 이익을 최대로 만드는 문제의 해가 된다.

$$
\begin{aligned}
Objective : &\max_{z} z = 100x_1 + 40x_2 \\
Constraints : &100x_1 + 50x_2 \leq 3000, \\
&10x_1 \leq 100, \\
&x_1, x_2 \geq 0.
\end{aligned}
$$

위의 문제를 다음과 같은 Python코드를 통하여 풀 수 있다.

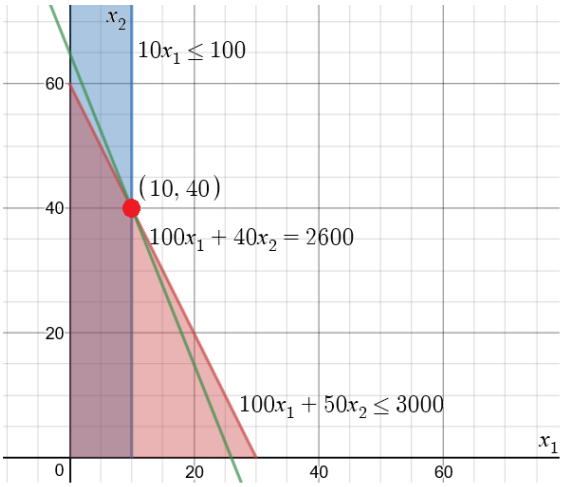
```
from scipy.optimize import linprog

# 목적함수 계수 (최소화를 위해 음수로 설정)
c = [-100, -40]

# 제약조건 계수
A = [[100, 50],
     [10, 0]]
```

[1] 해당 문제는 위키피디아의 선형계획법 페이지에서 가져옴.

```
# 제약조건 우변 값
b = [3000, 100]

# 변수의 하한
x0_bounds = (0, None)
x1_bounds = (0, None)

# 선형계획법 문제 풀이
res = linprog(c, A_ub=A, b_ub=b, bounds=[x0_bounds, x1_bounds], method='highs')

# 결과 출력
print('Optimal value:', -res.fun, '\nX:', res.x)
```

```
## Optimal value: 2600.0
## X: [10. 40.]
```

위의 문제를 그림으로 풀면 다음과 같다.



선형계획법 기출

순현가 가치를 최대화하는 투자 계획을 세우려고 한다. 정해진 예산은 1년차 50억, 2년차 60억, 3년차 80억을 넘지 않는 선에서 포트폴리오를 운영하려고 할 때, 현재 가능한 최대 NPV를 달성할 수 있는 최적의 투자안을 구하시오.

	1년차	2년차	3년차	NPV
투자안 1	23	23	15	30
투자안 2	15	15	12	20
투자안 3	17	25	12	31
투자안 4	16	12	13	42

	1년차	2년차	3년차	NPV
투자안 5	24	23	17	44

풀이

$$
\begin{aligned}
Objective: & \max_{z} z = 30x_1 + 20x_2 + 31x_3 + 42x_4 + 44x_5 \\
Constraints: & 23x_1 + 15x_2 + 17x_3 + 16x_4 + 24x_5 \leq 50, \\
& 23x_1 + 15x_2 + 25x_3 + 12x_4 + 23x_5 \leq 60, \\
& 15x_1 + 12x_2 + 12x_3 + 13x_4 + 17x_5 \leq 80, \\
& x_1, x_2, x_3, x_4, x_5 \geq 0.
\end{aligned}
$$

위의 문제 (해가 이진수) 를 직접적으로 푸는 파이썬 패키지는 ortools에 있으나, ADP의 기본적으로 제공되는 패키지에 속하지는 않는다.

```
import numpy as np
from ortools.linear_solver import pywraplp

# 투자안의 비용
costs = np.array([[23, 23, 15],
                  [15, 15, 12],
                  [17, 25, 12],
                  [16, 12, 13],
                  [24, 23, 17]])

# 각 투자안의 NPV
npv = np.array([30, 20, 31, 42, 44])

# 각 해에 대한 예산 제약조건
budgets = np.array([50, 60, 80])

# Solver 초기화
solver = pywraplp.Solver.CreateSolver('SCIP')

# 변수 설정
x = []
for i in range(5):  # 5개 투자안
    x.append(solver.IntVar(0.0, 1.0, 'x[%i]' % i))

# 제약조건 설정
for i in range(3):  # 3년차
    solver.Add(sum(costs[j][i] * x[j] for j in range(5)) <= budgets[i])
```

```
# 목적함수 설정
```

```
## <ortools.linear_solver.pywraplp.Constraint; proxy of <Swig Object of type ↩
'operations_research::MPConstraint *' at 0x000001ABA1FDCE10> >
## <ortools.linear_solver.pywraplp.Constraint; proxy of <Swig Object of type ↩
'operations_research::MPConstraint *' at 0x000001ABA1FF6900> >
## <ortools.linear_solver.pywraplp.Constraint; proxy of <Swig Object of type ↩
'operations_research::MPConstraint *' at 0x000001ABA1FB2630> >
```

```
objective = solver.Objective()
for i in range(5):
    objective.SetCoefficient(x[i], float(npv[i]))
objective.SetMaximization()

# Solver 실행
solver.Solve()

# 결과 출력
```

```
## 0
```

```
print('Maximum NPV: ', objective.Value())
```

```
## Maximum NPV:  93.0
```

```
print('Investment decisions for each project:')
```

```
## Investment decisions for each project:
```

```
for i in range(5):
    print('x[%i] = %i' % (i, x[i].solution_value()))
```

```
## x[0] = 0
## x[1] = 1
## x[2] = 1
## x[3] = 1
## x[4] = 0
```

위와 같은 문제는 다음의 코드처럼 주어진 모든 경우의 수를 구한 후, 제약에 걸리는 경우는 제외시키는 방법으로 구하는 게 훨씬 빠르다.

```
import numpy as np
from itertools import product

# 각 투자안의 연간 비용과 NPV
costs = np.array([[23, 23, 15],
                  [15, 15, 12],
                  [17, 25, 12],
                  [16, 12, 13],
                  [24, 23, 17]])
npv = np.array([30, 20, 31, 42, 44])

# 연간 예산
budgets = np.array([50, 60, 80])

# 가능한 모든 조합 생성
comb = list(product([0, 1], repeat=5))

# 조건을 만족하는 조합 및 그에 따른 NPV 계산
valid_comb = []
valid_npv = []
for c in comb:
    total_costs = np.dot(c, costs)
    if np.all(total_costs ⩽ budgets):
        valid_comb.append(c)
        valid_npv.append(np.dot(c, npv))

# 최대 NPV 찾기
max_npv_idx = np.argmax(valid_npv)
optimal_comb = valid_comb[max_npv_idx]
optimal_npv = valid_npv[max_npv_idx]

print('Optimal investment plan: ', optimal_comb)
```

```
## Optimal investment plan:  (0, 1, 1, 1, 0)
```

```
print('Maximum NPV: ', optimal_npv)
```

```
## Maximum NPV:  93
```

제 10 장

로지스틱 회귀분석

이항분포와 오즈

오즈(Odds)의 개념

로지스틱 회귀분석은 확률의 오즈를 선형모형으로 모델링하는 개념이다. 따라서 확률의 오즈가 무엇인지 먼저 알아보자.

확률의 오즈(odds)란 어떤 사건이 발생할 확률과 그 사건이 발생하지 않을 확률의 비율을 말한다. 즉,

$$\text{Odds of Event A} = \frac{P(A)}{P(A^c)} = \frac{P(A)}{1-P(A)}$$

와 같은 형태로 표현된다.

예를 들어, 동전 던지기에서 앞면이 나올 확률이 1/2이라면, 앞면이 나올 오즈는 1/1, 즉, 1이 된다.


```
import pandas as pd

admission_data = pd.read_csv("./data/admission.csv")
print(admission_data.shape)
```

대학교 입학 데이터

```
## (400, 5)
```

```
print(admission_data.head())
```

```
##    admit  gre   gpa  rank gender
## 0      0  380  3.61     3      M
## 1      1  660  3.67     3      F
## 2      1  800  4.00     1      F
```

```
## 3      1  640  3.19     4      M
## 4      0  520  2.93     4      M
```

다음은 학교 입학 데이터이다. 데이터에서 입학이 허가될 확률의 오즈를 구해보자.

```
p_hat = admission_data['admit'].mean()
p_hat / (1 - p_hat)
```

```
## 0.4652014652014652
```

입학할 확률에 대한 오즈는 0.465가 된다. 즉, 입학에 실패할 확률의 46%정도이다. 즉, 오즈가 1을 기준으로 낮을 경우, 발생하기 어렵다는 뜻이다.

범주형 변수를 사용한 오즈 계산 이 데이터에는 rank 변수가 존재한다. 각 범주별 입학에 대한 오즈를 계산할 수도 있다.

```
unique_ranks = admission_data['rank'].unique()
print(unique_ranks)
```

```
## [3 1 4 2]
```

1에서부터 4등급까지 존재하는 것을 확인했다. 각 등급별 입학에 대한 오즈를 구해보자.

```
grouped_data = admission_data.groupby('rank').agg(p_admit=('admit', 'mean'))
grouped_data['odds'] = grouped_data['p_admit'] / (1 - grouped_data['p_admit'])
print(grouped_data)
```

```
##        p_admit      odds
## rank
## 1     0.540984  1.178571
## 2     0.357616  0.556701
## 3     0.231405  0.301075
## 4     0.179104  0.218182
```

1등급 학생들이 입학에 성공할 확률은 입학에 실패할 확률보다 18% 더 높으며, 나머지 등급의 학생들은 입학할 확률이 입학에 실패할 확률보다 더 낮다는 것을 확인할 수 있다.

오즈(Odds)를 사용한 확률 역산 Odds가 주어졌을 때, 위의 관계를 사용하여 역으로 확률을 계산할 수 있다.

$$\hat{p} = \frac{Odds}{Odds + 1}$$

앞에서 살펴본 1등급 학생들의 오즈를 사용하여 입학할 확률을 계산해보자.

```
1.178 / (1.178 + 1)
```

```
## 0.5408631772268135
```

로그 오즈

일반 회귀에서는 종속변수 Y를 독립변수들의 선형결합으로 모델링하였다. 로지스틱 회귀에서는 확률을 독립변수들의 선형결합으로 모델링하고 싶다. 하지만, 확률이라는 것은 0과 1사이의 값을 가지게 되고, 모델링하는 선형결합은 $-\infty$에서 ∞까지의 값을 가지므로, 이것을 그대로 종속변수로 사용할 수는 없다. 로지스틱 회귀분석에서는 앞에서 배운 오즈에 로그를 씌운 로그 오즈를 이용하여 $-\infty$에서 ∞까지의 값으로 늘려준다.

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

확률값 p에 대하여 로그 오즈값의 그래프를 그리면 다음과 같다.

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

p = np.arange(0, 1.01, 0.01)
log_odds = np.log(p / (1 - p))

plt.plot(p, log_odds)
```

```
## [<matplotlib.lines.Line2D object at 0×000001ABA203F308>]
```

```
plt.xlabel('p')
```

```
## Text(0.5, 0, 'p')
```

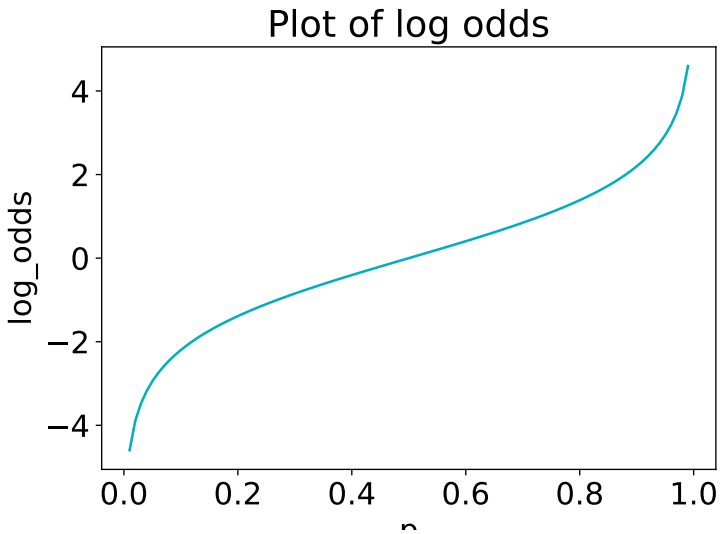
```
plt.ylabel('log_odds')
```

```
## Text(0, 0.5, 'log_odds')
```

```
plt.title('Plot of log odds')
```

```
## Text(0.5, 1.0, 'Plot of log odds')
```

```
plt.show()
```



로지스틱 회귀계수 예측 아이디어

로지스틱 회귀분석의 계수를 구하는 내용은 MLE를 사용하여 구하는 것이다. 구하는 방식에 대한 설명은 추후 슬기로운 통계생활 채널 영상으로 대체하겠다. 여기서는 변수의 레벨이 가장 간단한 성별 변수를 사용하여 계수를 예측해보자.

```
odds_data = admission_data.groupby('rank').agg(p_admit=('admit', 'mean')).reset_index()
odds_data['odds'] = odds_data['p_admit'] / (1 - odds_data['p_admit'])
odds_data['log_odds'] = np.log(odds_data['odds'])
print(odds_data)
```

```
##    rank   p_admit      odds  log_odds
## 0     1  0.540984  1.178571  0.164303
## 1     2  0.357616  0.556701 -0.585727
## 2     3  0.231405  0.301075 -1.200395
## 3     4  0.179104  0.218182 -1.522427
```

rank 변수가 범주형이긴 하지만 순서가 있는 변수이기 때문에 수치형 변수라고 생각하고 회귀직선을 구해보자. 로지스틱 회귀분석의 계수를 이런식으로 추정하는 것이 아니지만, 아이디어를 충분히 잡아내는 방식이라 생각한다.

```
import statsmodels.api as sm

model = sm.formula.ols("log_odds ~ rank", data=odds_data).fit()
print(model.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:               log_odds   R-squared:                       0.972
```

```
## Model:                            OLS   Adj. R-squared:                  0.957
## Method:                 Least Squares   F-statistic:                     68.47
## Date:                   토, 28 10 2023   Prob (F-statistic):             0.0143
## Time:                        14:02:22   Log-Likelihood:                 3.2107
## No. Observations:                   4   AIC:                            -2.421
## Df Residuals:                       2   BIC:                            -3.649
## Df Model:                           1
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept      0.6327      0.188      3.368      0.078      -0.175       1.441
## rank          -0.5675      0.069     -8.275      0.014      -0.863      -0.272
## ==============================================================================
## Omnibus:                        nan   Durbin-Watson:                   2.037
## Prob(Omnibus):                  nan   Jarque-Bera (JB):                0.602
## Skew:                        -0.062   Prob(JB):                        0.740
## Kurtosis:                     1.103   Cond. No.                         7.47
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
```


이 직선과 주어진 로그 오즈를 시각화해보자.

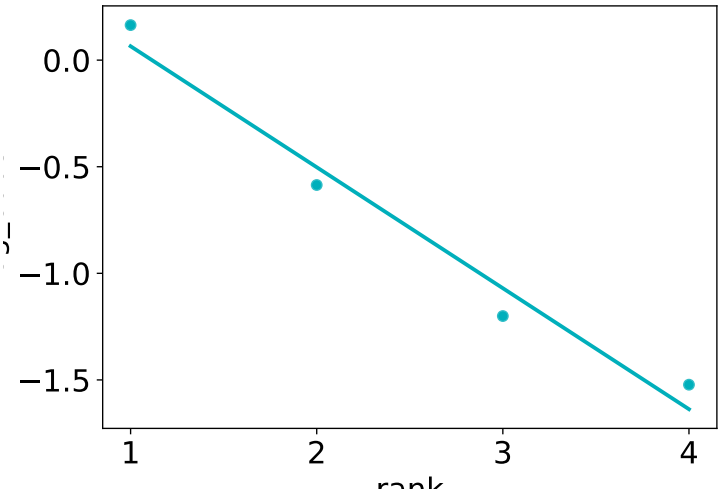
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(data=odds_data, x='rank', y='log_odds')
```

```
## <AxesSubplot:xlabel='rank', ylabel='log_odds'>
```

```
sns.regplot(data=odds_data, x='rank', y='log_odds', ci=None)
```

```
## <AxesSubplot:xlabel='rank', ylabel='log_odds'>
```

```
plt.show()
```



로지스틱 회귀계수 해석 아이디어

앞의 회귀모델의 경우 절편의 의미가 불분명한 모델이지만, 여기서 주의 깊게 봐야 할 것은 기울기 계수인 `-0.5675` 값을 어떻게 해석할 것인가이다. 회귀분석의 경우를 떠올리면 다음과 같이 해석할 수 있다.

rank가 1 단위 증가하면, y변수, 즉, 로그 오즈가 0.5675 만큼 "감소"한다. 하지만, 이러한 해석은 직관적으로 받아들이기가 어려우므로, 이 계수를 앞에서 살펴본 Odds를 구하는 방식으로 변형해보자.

$$log\left(\frac{p(x_{rank})}{1-p(x_{rank})}\right) = 0.6327 - 0.5675x_{rank}$$

양변에 지수를 취하여 왼쪽을 오즈로 만든다.

$$Odds(x_{rank}) = \frac{p(x_{rank})}{1-p(x_{rank})} = exp(0.6327 - 0.5675x_{rank})$$

이렇게 쓰게 되면 좋은 점이 하나있는데, 지수의 성질을 이용해서 -0.5675라는 계수값만 딱 떨어뜨려놓을 수 있게 된다.

오즈비 (Odds ratio) rank가 x일때의 오즈와, 한 단위 증가한 x+1일때의 오즈를 분수꼴로 놓아보자. 이러한 값을 오즈들의 비율이라는 의미로 오즈비 (Odds ratio)라고 부른다.

$$\frac{Odds(x_{rank}+1)}{Odds(x_{rank})} = \frac{exp(0.6327 - 0.5675(x_{rank}+1))}{exp(0.6327 - 0.5675x_{rank})} = exp(-0.5675) \approx 0.567$$

즉, rank가 한 단위 증가할 때마다, Odds가 이전 오즈의 약 절반 가량 (56%)으로 감소하는 경향을 보인다. 이것은 앞에서 계산했던 rank별 오즈의 경향성과 일치한다.

```
selected_data = odds_data[['rank', 'p_admit', 'odds']]
selected_data['odds_frac'] = selected_data['odds'] / selected_data['odds'].shift(1, fill_value=sele
```

```
print(selected_data)
```

```
##    rank   p_admit      odds  odds_frac
## 0     1  0.540984  1.178571   1.000000
## 1     2  0.357616  0.556701   0.472352
## 2     3  0.231405  0.301075   0.540820
## 3     4  0.179104  0.218182   0.724675
```

오즈를 이용한 확률 역산 앞에서 오즈를 알고있다면, 이를 이용해서 확률을 역산하는 방법을 알아보았다. 따라서, 오즈에 대한 식을 사용하여 확률 $p(x_{rank})$는 다음과 같이 쓸 수 있다.

$$p(x_{rank}) = \frac{exp(0.6327 - 0.5675x_{rank})}{1 + exp(0.6327 - 0.5675x_{rank})}$$

위 식을 이용하면 각 랭크별 입학 확률을 다음과 같이 계산할 수 있다.

```
rank_vec = np.array([1, 2, 3, 4])
result = np.exp(0.6327 - 0.5675 * rank_vec) / (1 + np.exp(0.6327 - 0.5675 * rank_vec))

print(result)
```

```
## [0.51629423 0.37700031 0.25544112 0.16283279]
```

그리고 이러한 추세를 rank변수가 $-\infty$에서 ∞까지 값을 갖는 연속형 변수라고 놓았을 때는 다음과 같이 추세선을 그려볼 수 있다.

Python에서 로지스틱 회귀분석 하기

앞에서 살펴본 admission 데이터를 사용하여 로지스틱 회귀분석 모델을 만들고 결과를 살펴보자.

```
import statsmodels.api as sm

admission_data['rank'] = admission_data['rank'].astype('category')
admission_data['gender'] = admission_data['gender'].astype('category')

model = sm.formula.logit("admit ~ gre + gpa + rank + gender", data=admission_data).fit()
```

```
## Optimization terminated successfully.
##          Current function value: 0.573066
##          Iterations 6
```

```
print(model.summary())
```

```
##                            Logit Regression Results
## ==============================================================================
## Dep. Variable:                  admit   No. Observations:                  400
## Model:                          Logit   Df Residuals:                      393
## Method:                           MLE   Df Model:                            6
## Date:                토, 28 10 2023   Pseudo R-squ.:                 0.08305
## Time:                        14:02:25   Log-Likelihood:                -229.23
## converged:                       True   LL-Null:                       -249.99
## Covariance Type:            nonrobust   LLR p-value:                 2.283e-07
## ==============================================================================
##                  coef    std err          z      P>|z|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept      -3.9536      1.149     -3.442      0.001      -6.205      -1.702
## rank[T.2]      -0.6723      0.317     -2.123      0.034      -1.293      -0.052
## rank[T.3]      -1.3422      0.345     -3.887      0.000      -2.019      -0.665
## rank[T.4]      -1.5529      0.418     -3.717      0.000      -2.372      -0.734
## gender[T.M]    -0.0578      0.228     -0.254      0.800      -0.504       0.388
## gre             0.0023      0.001      2.062      0.039       0.000       0.004
## gpa             0.8032      0.332      2.420      0.016       0.153       1.454
## ==============================================================================
```

절편에 대한 계수는 일반적으로 해석하지 않는다.

- `gre` (0.0023): GRE 점수가 1점 증가할 때마다 합격 로그 오즈가 0.0023만큼 증가합니다. 이는 GRE 점수가 1점 증가할 때마다 합격에 대한 오즈가 약 0.2% 증가합니다.
- `gpa` (0.8032): GPA가 1점 증가할 때마다 합격 로그 오즈가 0.8032만큼 증가합니다. 이는 GPA가 1점 증가할 때마다 합격에 대한 오즈가 약 123% 증가합니다

```
import math
math.exp(0.0023)
```

```
## 1.002302647029
```

```
math.exp(0.8032)
```

```
## 2.232674066397348
```

- `gender` (-0.0578): 성별이 남성인 학생은 여성 학생에 비해 합격 로그 오즈가 0.0578만큼 낮습니다. 이는 여학생 그룹과 남학생 그룹의 합격에 대한 오즈비가 0.943으로 1보다 작습니다. 그러나 p 값이 0.800으로, 이 변수의 계수는 통계적으로 유의하지 않다고 볼 수 있습니다. 즉, 이 데이터에서 성별이 합격 여부에 큰 영향을 미치지 않는 것으로 보입니다.

```
math.exp(-0.0578)
```

```
## 0.9438386963005431
```

각 계수 검정하기 with Wald test

귀무가설: $\beta_i = 0$ 을 검정하기 위한 검정통계량은 다음과 같다.

$$z = \frac{\hat{\beta}_i}{SE_{\hat{\beta}_i}} \sim \mathcal{N}(0, 1^2)$$

따라서 유의수준 5%하에서 gre의 계수가 0이라는 귀무가설을 기각할 수 있다.

```
import scipy.stats as stats

result1 = 0.002256 / 0.001094
result2 = 2 * (1 - stats.norm.cdf(result1))

print(result1)
```

```
## 2.0621572212065815
```

```
print(result2)
```

```
## 0.03919277001389343
```

각 Odds ratio에 대한 신뢰구간 구하기

앞에서 기울기 β에 대한 검정이 Wald 검정을 사용하는 것을 생각해보면, 기울기에 대한 신뢰구간을 다음과 같이 구할 수 있을 것이라 생각할 수 있다.

$$\beta_i \pm z^* SE_{b_1}$$

따라서 오즈비(Odds ratio)는 e^{β}이므로 신뢰구간 을 구할 때, 기울기에 대한 신뢰구간에 지수꼴을 취해주면 된다.

$$(e^{\beta_i - z^* SE_{b_1}}, e^{\beta_i + z^* SE_{b_1}})$$

따라서, gre 변수의 계수 0.002256에 대한 95% 신뢰구간은 다음과 같다.

```
import scipy.stats as stats

a = round(model.params[1] - stats.norm.ppf(0.975) * 0.001094, 3)
b = round(model.params[1] + stats.norm.ppf(0.975) * 0.001094, 3)
```

```
glue_str = f"({a}, {b})"
print(glue_str)
```

```
## (-0.674, -0.67)
```

추가적으로 오즈비에 대한 신뢰구간을 구해보자. 이미 계산이 다 되어있기 때문에 간단하게 구할 수 있다.

```
a = round(np.exp(a), 3)
b = round(np.exp(b), 3)

glue_str = f"({a}, {b})"
print(glue_str)
```

```
## (0.51, 0.512)
```

LR test: 회귀모델이 유의한가?

일반 선형회귀에서는 F 검정을 사용해서 모델의 유의성을 체크했는데, 로지스틱 회귀분석의 경우 Likelihood ratio 검정을 진행한다. 이 검정의 귀무가설과 대립가설은 다음과 같다.

귀무가설: 모든 베타 계수들이 0이다. 대립가설: 0이 아닌 베타 계수가 존재한다.

이 검정의 검정통계량은 다음과 같다.

$$\Lambda = -2(\ell(\hat{\beta})^{(0)} - \ell(\hat{\beta})) \sim \chi^2_{k-r}$$

위 식에서 $\ell(\hat{\beta})^{(0)}$ 부분은 귀무가설 하에서의 로그 우도함수 값을 나타낸다. R의 결과에서는 Null에서의 deviance 값과 Residual deviance 값을 보여주고 있다. 이것은 위 식에서 -2 곱하기 각 로그우도 함수값을 나타낸다. 따라서 위의 Λ값은 Null deviance에서 Residual deviance 값을 빼서 구할 수 있다. 따라서, 검정통계량 값을 계산해보면, 다음과 같다.

$$G^2 = 499.98 - 458.45$$

위의 검정통계량은 카이제곱분포 자유도가 두 모델의 자유도 차를 따르게 되므로, 다음과 같이 p-value를 구할 수 있다.

계산된 p-value 값, `2.286918e-07`으로 보아 주어진 로지스틱 회귀모델은 통계적으로 유의하다라고 판단한다.

참고자료

본 챕터의 데이터와 내용은 많은 부분 UCLA 로지스틱 회귀분석 자료를 참고하였음을 밝힙니다.

제 11 장

챕터별 연습문제 풀이

Chapter 1. 통계적 검정의 근본 원리

문제 1. 농구, 축구 경기 선호도 확률

슬통 마을의 많은 사람들이 농구와 축구 관람을 좋아한다고 한다. 마을의 40%는 농구 경기를 좋아하고, 70%는 축구 경기를 좋아하며, 농구와 축구 관람을 모두 좋아하는 비율이 20%라고 한다.

- 사건 Basket: 특정 사람이 농구를 좋아할 확률
- 사건 Soccer: 특정 사람이 축구를 좋아할 확률

$$P(Basket) = 0.4$$
$$P(Soccer) = 0.7$$
$$P(Basket \cap Soccer) = 0.2$$

1) 마을 사람을 무작위로 한 명 선택했을 때, 그 사람이 농구와 축구 둘 다 좋아하지 않는 사람일 확률을 구하세요.

구하고자 하는 것: $P(B^c \cap S^c)$

$$P(B^c \cap S^c) = P((B \cup S)^c) = 1 - P(B \cup S)$$

확률의 덧셈법칙에 의하여 다음이 성립한다.

$$P(B \cup S) = P(B) + P(S) - P(B \cap S) = 0.4 + 0.7 - 0.2 = 0.9$$

따라서 농구와 축구 둘 다 좋아하지 않는 사람일 확률은 $1 - 0.9 = 0.1$

2) 마을 사람을 무작위로 한 명 선택해서, 농구를 좋아하냐고 물어보았다. 그 사람이 농구를 좋아하지 않는다고 대답했다면, 축구를 좋아할 확률을 구하세요.

구하고자 하는 것: $P(S|B^c)$

위 확률은 조건부 확률에 의하여 다음과 같이 쓸 수 있다.

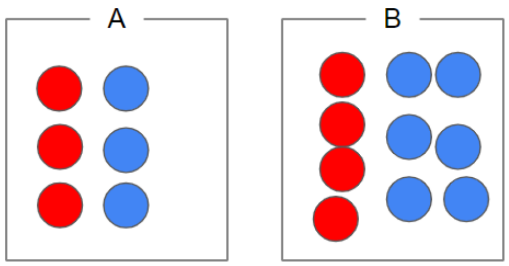
$$P(S|B^c) = \frac{P(S \cap B^c)}{P(B^c)} = \frac{P(S) - P(S \cap B)}{1 - P(B)} = \frac{0.7 - 0.2}{0.6} = \frac{5}{6}$$

```
round(5/6, 2)
```

```
## 0.83
```

문제 2. 빨간공, 파란공

두 개의 상자 A, B가 놓여있다. A 상자에는 빨간색 공이 3개, 파란색 공이 3개 들어있고, B 상자에는 빨간색 공이 4개, 파란색 공이 6개 들어있다고 한다.



1) 각각의 상자에서 하나씩 공을 꺼낼 때, 두 공이 같은 색깔일 확률을 구하세요.

- 사건 AR: 상자 A에서 빨간공을 꺼내는 사건
- 사건 AB: 상자 A에서 파란공을 꺼내는 사건
- 사건 BR: 상자 B에서 빨간공을 꺼내는 사건
- 사건 BB: 상자 B에서 파란공을 꺼내는 사건
- 두 공이 같은 색깔
  - 상자 A에서 빨간공을 꺼내고, 상자 B에서 빨간공을 꺼내거나
  - 상자 A에서 파란공을 꺼내고, 상자 B에서 파란공을 꺼내는 경우
- 구하고자 하는 것: $P((AR \cap BR) \cup (AB \cap BB))$

모두 빨간공을 꺼내는 사건과 모두 파란공을 꺼내는 사건은 동시에 일어날 수 없으므로 각각 사건의 확률의 합으로 나눠쓸 수 있음.

$$P((AR \cap BR) \cup (AB \cap BB)) = P(AR \cap BR) + P(AB \cap BB)$$

각 상자에서 공을 꺼내는 사건은 독립이므로 독립사건 곱셈법칙 적용

$$\begin{aligned} P(AR \cap BR) + P(AB \cap BB) &= P(AR)P(BR) + P(AB)P(BB) \\ &= 0.5 * 0.4 + 0.5 * 0.6 \end{aligned}$$

```
0.5 * 0.4 + 0.5 * 0.6
```

```
## 0.5
```

2) 이번에는 슬통이가 두 개 상자 중 하나에서 공을 하나 꺼내왔다고 한다. 뽑힌 공의 색깔을 보니 빨간색이었다. 슬통이가 이 공을 상자 A에서 꺼냈을 확률을 구하세요.

- 사건 A: 상자 A를 선택하는 사건
- 사건 B: 상자 B를 선택하는 사건
- 사건 R: 빨간공을 꺼내는 사건
- 구하는 것: $P(A|R)$

베이즈 정리에 의하여 다음이 성립한다.

$$\begin{aligned} P(A|R) &= \frac{P(A)P(R|A)}{P(R)} \\ &= \frac{P(A)P(R|A)}{P(A)P(R|A)+P(B)P(R|B)} \\ &= \frac{0.5*0.5}{0.5*0.5+0.5*0.4} \end{aligned}$$

```
p_a_bar_r = (0.5 * 0.5)/(0.5 * 0.5 + 0.5 * 0.4)
round(p_a_bar_r,3)
```

```
## 0.556
```

문제 3. ADP 실기시험 성적 분포

2022년에 실시 된 ADP 실기 시험의 통계파트 표준점수는 평균이 30, 표준편차가 5인 정규분포를 따른다고 한다.

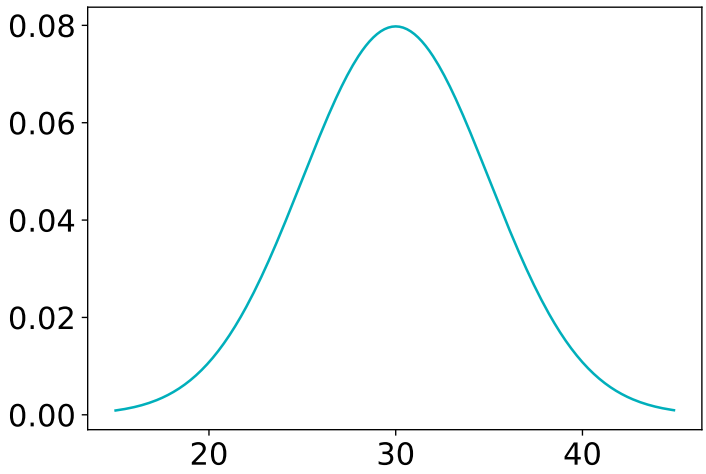
$$X \sim \mathcal{N}(30, 5^2)$$

1) ADP 실기 시험의 통계파트 표준점수의 밀도함수를 그려보세요.

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

x = np.arange(15,45,0.1)
y = norm.pdf(x, loc=30, scale=5)
plt.plot(x, y)
```

```
## [<matplotlib.lines.Line2D object at 0×000001ABA2504A88>]
```

```
plt.show()
```



2) ADP 수험생을 임의로 1명을 선택하여 통계 점수를 조회했을때 45점 보다 높은 점수를 받았을 확률을 구하세요.

구하는 것: $P(X > 45)$

```
1 - norm.cdf(45, 30, 5)
```

```
## 0.0013498980316301035
```

3) 슬통이는 상위 10%에 해당하는 점수를 얻었다고 한다면, 슬통이의 점수는 얼마인지 계산해보세요.

구하는 것: $P(X > k) = 0.1$ 에서 k 값

Quantile 을 구해주는 `qnorm` 함수를 사용하여 계산

```
norm.ppf(0.9, 30, 5)
```

```
## 36.407757827723
```

4) 슬기로운 통계생활의 해당 회차 수강생은 16명이었다고 한다. 16명의 통계 파트 점수를 평균내었을 때, 이 평균값이 따르는 분포의 확률밀도 함수를 1번의 그래프와 겹쳐 그려보세요.

중심극한정리에 의하여, 표본평균 확률변수는 다음과 같은 정규분포를 따른다.

$$\overline{X} \sim \mathcal{N}(30, \frac{5^2}{16})$$

따라서 개별 점수의 정규분포보다 훨씬 좁은 종모양의 분포를 띄게 된다.

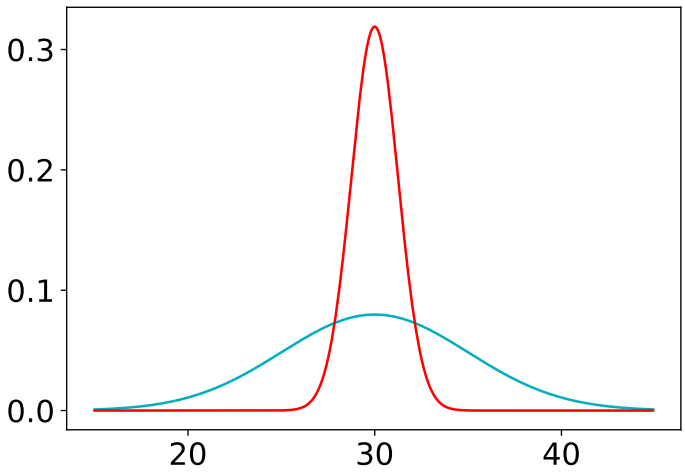
```
x = np.arange(15,45,0.1)
y = norm.pdf(x, loc=30, scale=5)
plt.plot(x, y)
```

```
## [<matplotlib.lines.Line2D object at 0×000001ABA26FD248>]
```

```
y2 = norm.pdf(x,loc=30,scale=5/np.sqrt(16))
plt.plot(x,y2,color='red')
```

```
## [<matplotlib.lines.Line2D object at 0×000001ABA2710CC8>]
```

```
plt.show()
```



5) 슬기로운 통계생활 ADP 반 수강생들의 통계점수를 평균내었다고 할 때, 이 값이 38점보다 높게 나올 확률을 구하세요.

구하고자 하는 것: $P(\overline{X} > 38)$ 여기서 $\overline{X} \sim \mathcal{N}(30, \frac{5^2}{16})$

```
1 - norm.cdf(38, 30, 5/np.sqrt(16))
```

```
## 7.76885222819601e-11
```

따라서 슬기로운 통계생활 ADP 반 수강생들의 통계점수를 평균내었다고 할 때, 이 값이 38점보다 높게 나올 사건은 거의 일어나기 불가능하다.

문제 4. 코비드 19 발병률

Covid-19의 발병률은 1%라고 한다. 다음은 이번 코로나 사태로 인하여 코로나 의심 환자들 1,085명을 대상으로 슬통 회사의 "다잡아" 키트를 사용하여 양성 반응을 체크한 결과이다.

키트 \ 실제	양성	음성	합계
양성	370	10	380
음성	15	690	705
———––	——	——	——
합계	385	700	1085

- 사건 DP: 키트가 양성으로 판단하는 사건
- 사건 TP: 실제상태가 양성인 사건
- 사건 DN: 키트가 음성으로 판단하는 사건
- 사건 TN: 실제상태가 음성인 사건

1) 다잡아 키트가 코로나 바이러스에 걸린 사람을 양성으로 잡아낼 확률을 계산하세요.

- 구하는 것: $P(DP|TP)$

$$P(DP|TP) = 370/385$$

```
370 / 385
```

```
## 0.961038961038961
```

2) 슬통 회사에서 다잡아 키트를 사용해 양성으로 나온 사람이 실제로는 코로나 바이러스에 걸려 있을 확률을 97%라며, 키트의 우수성을 주장했다. 이 주장이 옳지 않은 이유를 서술하세요.

표본으로 뽑힌 집단의 유병률과 모집단의 유병률의 차이가 크다.

3) Covid-19 발병률을 사용하여, 키트의 결과값이 양성으로 나온 사람이 실제로 코로나 바이러스에 걸려있을 확률을 구하세요.

- Covid-19 발병률의 의미: $P(TP) = 0.01$

구하고자 하는 것: $P(TP|DP)$

위 확률은 베이즈 정리에 의하여 다음과 같이 계산 할 수 있다.

$$\begin{aligned} P(TP|DP) &= \frac{P(TP \cap DP)}{P(DP)} \\ &= \frac{P(TP)P(DP|TP)}{P(TP)P(DP|TP) + P(TN)P(DP|TN)} \end{aligned}$$

테이블에서 얻을 수 있는 정보 중 $P(DP|TP)$와 $P(DP|TN)$의 정보는 상대적으로 믿을 수 있는 정보이므로, 발병률 정보와 같이 대입해서 계산한다.

$$\begin{aligned} &\frac{P(TP)P(DP|TP)}{P(TP)P(DP|TP) + P(TN)P(DP|TN)} \\ &= \frac{0.01 * (370/385)}{0.01 * (370/385) + 0.99 * (10/700)} \end{aligned}$$

```
sol = (0.01 * (370 / 385)) / (0.01 * (370 / 385) + 0.99 * (10 / 700))
round(sol,3)
```

```
## 0.405
```

따라서, 키트의 결과값이 양성으로 나온 사람이 실제로 코로나 바이러스에 걸려있을 확률은 40.5%로 추정하는게 합리적이다.

문제 5. 카이제곱분포와 표본분산

자유도가 k인 카이제곱분포를 따르는 확률변수 X를

$$X \sim \chi^2(k)$$

과 같이 나타내고, 이 확률변수의 확률밀도함수는 다음과 같습니다.

$$f_X(x;k) = \frac{1}{2^{k/2}\Gamma(k/2)}\, x^{k/2-1}e^{-x/2}$$

다음의 물음에 답하세요.

1) 자유도가 4인 카이제곱분포의 확률밀도함수를 그려보세요.

- 관련 패키지

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import chi2, norm

x = np.linspace(0, 20, 1000)
pdf = chi2.pdf(x, 4)

plt.plot(x, pdf)
```

```
## [<matplotlib.lines.Line2D object at 0×000001ABA26ED908>]
```

```
plt.xlabel("x")
```

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

```
plt.ylabel("f_X(x; k)")
```

```
## Text(0, 0.5, 'f_X(x; k)')
```

```
plt.show()
```

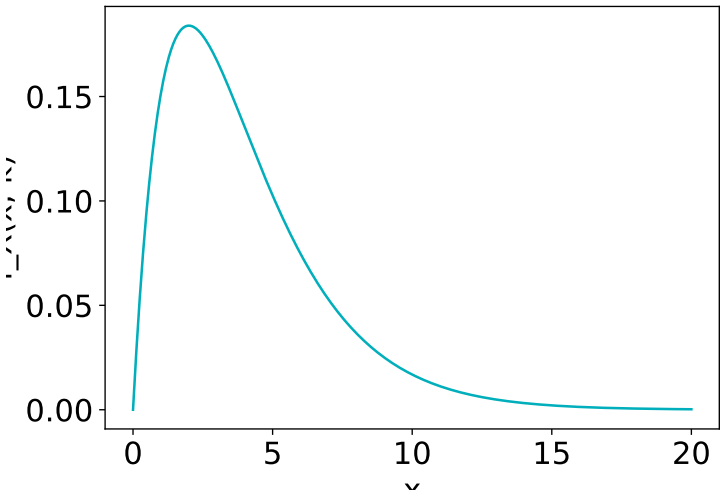


그림 11.1: 자유도 4인 카이제곱분포 pdf

2) 다음의 확률을 구해보세요.

$$P(3 \leq X \leq 5)$$

```
chi2.cdf(5, 4) - chi2.cdf(3, 4)
```

```
## 0.27052790518742903
```

3) 자유도가 4인 카이제곱분포에서 크기가 1000인 표본을 뽑은 후, 히스토그램을 그려보세요.

```
np.random.seed(2023)
sample_size = 1000
sample_data = chi2.rvs(4, size=sample_size)

plt.hist(sample_data, bins=50, density=True,
         color="lightblue", edgecolor="black");
plt.xlabel("sample")
```

```
## Text(0.5, 0, 'sample')
```

```
plt.ylabel("density")
```

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

```
plt.show()
```

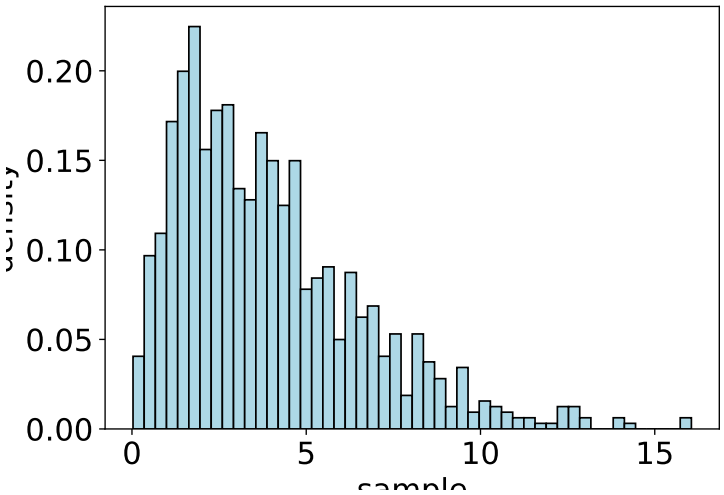


그림 11.2: 자유도가 4인 카이제곱분포에서 추출된 표본의 히스토그램

4) 자유도가 4인 카이제곱분포를 따르는 확률변수에서 나올 수 있는 값 중 상위 5%에 해당하는 값은 얼마인지 계산해보세요.

```
chi2.ppf(0.95, 4)
```

```
## 9.487729036781154
```

5) 3번에서 뽑힌 표본값들 중 상위 5%에 위치한 표본의 값은 얼마인가요?

```
np.percentile(sample_data, 95)
```

```
## 9.079466124811068
```

6) 평균이 3, 표준편차가 2인 정규분포를 따르는 확률변수에서 크기가 20인 표본, $x_1, ..., x_{20}$,을 뽑은 후 표본분산을 계산한 것을 s_1^2이라 생각해보죠. 다음을 수행해보세요!

- 같은 방법으로 500개의 s^2들, $s_1^2, s_2^2, ..., s_{500}^2$ 발생시킵니다.
- 발생한 500개의 s^2들 각각에 4.75를 곱하고, 그것들의 히스토그램을 그려보세요. (히스토그램을 그릴 때 `density = True` 옵션을 사용해서 그릴 것)
- 위에서 그린 히스토그램에 자유도가 19인 카이제곱분포 확률밀도함수를 겹쳐그려보세요.

```
np.random.seed(2023)
n = 20
num_samples = 500
var_samples = []
```

```
for i in range(num_samples):
    x = norm.rvs(3, 2, size=n)
    var_samples.append(np.var(x, ddof=1))

scaled_var_samples = np.array(var_samples) * 4.75

plt.hist(scaled_var_samples,
         bins=50, density=True, color="lightblue",
         edgecolor="black");
plt.xlabel("4.75 * s^2");
plt.ylabel("density");

x = np.linspace(0, max(scaled_var_samples), 1000)
pdf_chi19 = chi2.pdf(x, df=19)

plt.plot(x, pdf_chi19, 'r--', linewidth=2);
plt.legend(["histogram", "df 19 chisquare dist"], loc="upper right");
plt.show()
```

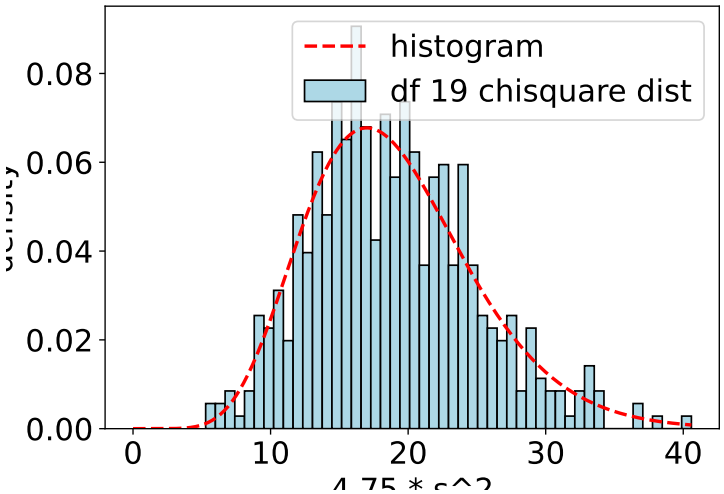


그림 11.3: 500개의 scale 된 표본분산의 히스토그램

Chapter 2. 통계적 검정의 근본 원리

문제 1. 신뢰구간 구하기

다음은 한 고등학교의 3학년 학생들 중 16명을 무작위로 선별하여 몸무게를 측정한 데이터이다. 이 데이터를 이용하여 해당 고등학교 3학년 전체 남학생들의 몸무게 평균을 예측하고자 한다.

79.1, 68.8, 62.0, 74.4, 71.0, 60.6, 98.5, 86.4, 73.0, 40.8, 61.2, 68.7, 61.6, 67.7, 61.7, 66.8

단, 해당 고등학교 3학년 남학생들의 몸무게 분포는 정규분포를 따른다고 가정한다.

1) 모평균에 대한 95% 신뢰구간을 구하세요.

2) 작년 남학생 3학년 전체 분포의 표준편차는 6kg 이었다고 합니다. 이 정보를 이번 년도 남학생 분포의 표준편차로 대체하여 모평균에 대한 90% 신뢰구간을 구하세요.

데이터 입력

```
import numpy as np
from scipy.stats import t, norm

sample_data = np.array([79.1, 68.8, 62.0, 74.4, 71.0, 60.6, 98.5, 86.4,
                        73.0, 40.8, 61.2, 68.7, 61.6, 67.7, 61.7, 66.8])
sample_data
```

```
## array([79.1, 68.8, 62. , 74.4, 71. , 60.6, 98.5, 86.4, 73. , 40.8, 61.2,
##        68.7, 61.6, 67.7, 61.7, 66.8])
```

표본에서의 정보 추출하기

다음은 표본 평균과 표본 표준편차를 계산하는 파이썬 코드 입니다.

```
sample_mean = np.mean(sample_data)
sample_sd = np.std(sample_data, ddof=1)  # ddof=1로 설정하여 표본 표준편차를 계산
n = len(sample_data)                     # 표본 크기
last_year_sd = 6                         # 작년 표준편차
```

1) 모평균에 대한 95% 신뢰구간을 구하기 위해, 표본 평균과 표본 표준편차를 사용하고, 표본 크기는 16명이므로 자유도는 15입니다.

```
# 모평균에 대한 95% 신뢰구간 계산 (작년의 표준편차를 사용)
ci_95 = t.interval(0.95, df=n-1, loc=sample_mean,
                   scale=sample_sd/np.sqrt(n))
print("모평균에 대한 95% 신뢰구간: ", np.round(ci_95, 2))
```

```
## 모평균에 대한 95% 신뢰구간:  [62.13 75.65]
```

위의 계산을 stats.t.interval()을 사용하지 않고, 구하면 다음과 같습니다.

```
t_critical = t.ppf(1 - 0.05/2, df=n-1)  # t 분포의 분위수 계산
mg_of_error = t_critical * sample_sd / np.sqrt(n) # margin of error 계산
ci_95 = (sample_mean - mg_of_error, sample_mean + mg_of_error)
np.round(ci_95, 2)
```

```
## array([62.13, 75.65])
```

2) 모평균에 대한 90% 신뢰구간을 구하기 위해, 표본 평균을 사용하고, 표준 편차는 작년의 값인 6kg을 사용합니다. 표본 크기는 여전히 16명이므로 z 분포의 90% 신뢰구간을 계산하겠습니다.

```
# 모평균에 대한 90% 신뢰구간 계산 (작년의 표준편차를 사용)
ci_90 = norm.interval(0.90, loc=sample_mean,
                      scale=last_year_sd/np.sqrt(n))
print("모평균에 대한 90% 신뢰구간 (작년의 표준편차를 사용): ", np.round(ci_90, 2))
```

```
## 모평균에 대한 90% 신뢰구간 (작년의 표준편차를 사용):  [66.43 71.36]
```

위의 계산을 stats.norm.interval()을 사용하지 않고, 구하면 다음과 같습니다.

```
z_critical = norm.ppf(1 - 0.10/2)  # z 분포의 분위수 계산
mg_of_error = z_critical * last_year_sd / np.sqrt(n)
ci_90 = (sample_mean - mg_of_error, sample_mean + mg_of_error)
np.round(ci_90, 2)
```

```
## array([66.43, 71.36])
```

문제 2. 신형 자동차의 에너지 소비효율 등급

슬통 자동자는 매해 출시되는 신형 자동차의 에너지 소비효율 등급을 1등급으로 유지하고 있다. 22년 개발된 신형 모델이 한국 자동차 평가원에서 설정한 에너지 소비 효율등급 1등급을 받을 수 있을지 검정하려한다. 평가원에 따르면 1등급의 기준은 평균 복합 에너지 소비효율이 16.0 이상인 경우 부여한다고 한다.

다음은 신형 자동차 15대의 복합 에너지소비효율 측정한 결과이다.

15.078, 15.752, 15.549, 15.56, 16.098, 13.277, 15.462, 16.116, 15.214, 16.93, 14.118, 14.927, 15.382, 16.709, 16.804

표본에 의하여 판단해볼때, 현대자동차의 신형 모델은 에너지 효율 1등급으로 판단할 수 있을지 유의수준 1% 하에서 판단해보시오.

데이터 입력

```
import numpy as np

sample_data = np.array([15.078, 15.752, 15.549, 15.56, 16.098,
13.277, 15.462, 16.116, 15.214, 16.93,
14.118, 14.927, 15.382, 16.709, 16.804])
sample_data
```

```
## array([15.078, 15.752, 15.549, 15.56 , 16.098, 13.277, 15.462, 16.116,
##        15.214, 16.93 , 14.118, 14.927, 15.382, 16.709, 16.804])
```

귀무가설 vs. 대립가설

귀무가설: 22년 개발된 현대 자동차 신형 모델의 평균 에너지 소비 효율등급은 1등급 기준을 만족한다.

$$\mu \geq 16$$

대립가설: 22년 개발된 현대 자동차 신형 모델의 평균 에너지 소비 효율등급은 1등급 기준을 만족하지 않는다.

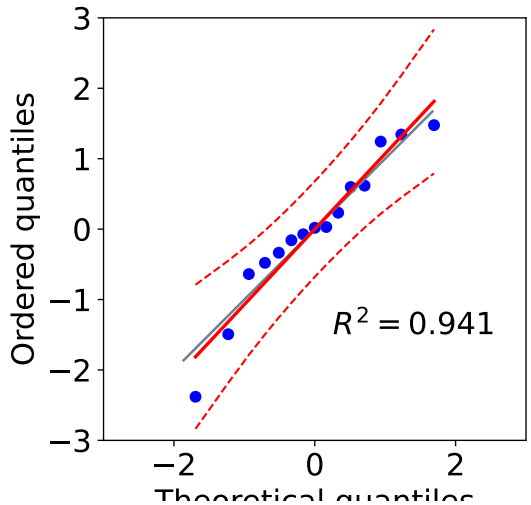
$$\mu < 16$$

가정 체크 및 검정 방법 설정

1표본 t 검정은 데이터가 정규성을 만족해야 한다는 전제를 만족해야 하므로 데이터의 정규성 가정을 체크한다.

정규성 검정 데이터에 대한 정규성 검정을 시각화 기법과 Shapiro-wilk 검정을 사용하여 시행한다.

```
import pingouin as pg
import matplotlib.pyplot as plt

ax = pg.qqplot(sample_data, dist='norm')
plt.ylim(-3, 3);
plt.xlim(-3, 3);
plt.show()
```



1 개의 표본을 제외하면정규성을 띄는 것으로 판단된다. 정확한 검정을 위하여 S-W 검정을 시행한다. S-W 검정의 유의수준은 정해진 값이 없으므로 5%를 기준으로 진행한다.

- Shapiro-Wilk 검정
  - 귀무가설: 데이터가 정규분포를 따른다.

- 대립가설: 데이터가 정규분포를 따르지 않는다.

```
from scipy.stats import shapiro
shapiro(sample_data)
```

```
## ShapiroResult(statistic=0.9448551535606384, pvalue=0.4473438858985901)
```

p-value 값 0.4473이 유의수준인 5%보다 크므로 귀무가설을 기각할 수 없다. 따라서 데이터는 정규성을 만족한다고 판단한다.

t 검정

표본이 정규성을 만족하므로 t-test를 실시한다.

```
from scipy.stats import ttest_1samp
ttest_1samp(sample_data, 16,alternative='less')
```

```
## Ttest_1sampResult(statistic=-1.8500447456376756, pvalue=0.042762417664207845)
```

표본의 평균 복합 에너지 효율은 15.53173이고, 검정통계량값은 -1.85이다. t 검정통계량에 대응하는 p-value값 0.04276가 유의수준인 1%보다 크므로, 귀무가설을 기각하지 못한다. 따라서 22년 현대 자동차 신형 모델 그룹의 평균 에너지 소비효율은 1등급을 만족한다고 판단한다.

기각역

t 통계량은 귀무가설하에서 자유도가 14인 t 분포를 따르며, 유의수준 1%에 해당하는 기각역은 다음을 만족하는 $\bar{x}$의 범위를 역산하면 된다.

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{\bar{x} - 16}{s/\sqrt{15}} \leq -2.624$$

```
result = 16 - 2.624 * (sample_data.std(ddof=1) / np.sqrt(15))
result.round(3)
```

```
## 15.336
```

따라서 기각역은 $\bar{x}$가 15.336보다 작은 영역이다.

검정력

검정력을 구하기 위해서 실제 평균 복합 에너지 평균값을 15이라고 가정할 때, 표본 평균이 15.335보다 크거나 같게 관찰될 확률을 구한다.

$$\begin{aligned} P\left(\bar{X} \leq 15.335 | \mu_a = 15\right) &= P\left(z \leq \frac{15.335 - 15}{1/\sqrt{15}}\right) \\ &= P\left(t \leq 1.324\right) \end{aligned}$$

따라서, 검정력은 다음과 같이 90.28%로 계산할 수 있다.

```
from scipy.stats import norm
norm.cdf(1.297449)
```

```
## 0.9027616289304574
```

문제 3. 검정력을 만족하는 표본 개수

귀무가설과 대립가설 설정

해당 검정을 위해 귀무가설과 대립가설을 다음과 같이 수립할 수 있다.

$$H_0 : \mu = 16 \quad vs. \quad H_a : \mu \neq 16$$

기각역 구하기

모 표준편차를 알고있다는 가정하에서 다음의 Z 통계량은 귀무가설하에서 표준정규분포를 따르며, 유의수준 5%에 해당하는 기각역은 다음을 만족하는 $\bar{x}$의 범위를 역산하면 된다.

$$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} = \frac{\bar{x} - 16}{2/\sqrt{15}} \geq 1.96$$

$$z = \frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} = \frac{\bar{x} - 16}{2/\sqrt{15}} \leq -1.96$$

따라서, R에서는 다음의 함수를 사용해서 구할 수 있다.

```
from scipy.stats import norm
norm.ppf(0.025, 16, 2/np.sqrt(15))
```

```
## 14.987878950494672
```

```
norm.ppf(0.975, 16, 2/np.sqrt(15))
```

```
## 17.01212104950533
```

즉, 기각역은 표본평균이 14.987보다 작은 영역과 17.012보다 큰 영역이다.

평균값 15일때의 검정력

검정력을 구하기 위해서 실제 평균 복합 에너지 평균값을 15이라고 가정할 때, 다음의 영역에 속할 확률을 구한다.

$$\begin{aligned} P\left(\bar{X} \geq 17.012 | \mu_a = 15\right) &= P\left(Z \geq \frac{17.012 - 15}{2/\sqrt{15}}\right) \\ &= P\left(Z \geq 3.896\right) \end{aligned}$$

$$P\left(\bar{X} \leq 14.987 | \mu_a = 15\right) = P\left(Z \leq \frac{14.987 - 15}{2/\sqrt{15}}\right)$$
$$= P\left(Z \leq -0.0251\right)$$

```
p1 = norm.cdf(-0.0251)
p2 = 1 - norm.cdf(3.896)

p1 + p2
```

```
## 0.49003649728580484
```

검정력을 만족하는 표본개수

다음의 코드를 통하여 유의수준 5%하에서 검정력 80%를 만족하는 표본크기를 구하면, 표본크기가 32 이상인 경우에 만족하는 것을 알 수 있다.

```
n = 32
a = 16 + norm.ppf(0.975, 0, 1) * (2/np.sqrt(n))
b = 16 - norm.ppf(0.975, 0, 1) * (2/np.sqrt(n))

left_a = (b-15) / (2/np.sqrt(n))
right_b = (a-15) / (2/np.sqrt(n))
norm.cdf(left_a) + (1-norm.cdf(right_b))
```

```
## 0.8074304194325567
```

문제 4. IQR 과 상자그림

주어진 데이터를 사용하여 다음의 물음에 답하세요.

$$12, 15, 14, 10, 18, 20, 21, 15, 17, 19, 10, 13, 16, 22, 50, 70$$

1. 첫 번째와 세 번째 사분위수(Q1, Q3)를 계산하세요.
2. Interquartile Range (IQR)를 계산하세요.
3. 이상치를 식별하고, 이들을 데이터 세트에서 찾아내세요.
4. 데이터의 상자그림(Boxplot)을 그리세요.

Chapter 3. t 검정 파헤치기

문제 1. 신약 효과 분석

새로 제안된 혈압약에 대한 효과 분석을 위하여 무작위로 배정된 두 그룹에 대한 혈압 측정 데이터이다. Treated 그룹의 경우 혈압약을 일정기간 복용하였으며, Control 그룹은 평상시 활동을 그대로 유지하였다. 표 11.2는 두 그룹의 혈압을 측정한 데이터이다.

혈압약의 사용이 혈압을 떨어뜨리는 효과가 있는지 유의수준 5%하에서 검정하시오.

표 11.2: 혈압약 효과 측정 데이터

ID	Score	Group
1	3.81	control
2	4.47	control
3	4.81	control
4	4.79	control
5	4.25	control
6	3.93	treated
7	4.26	treated
8	3.74	treated
9	3.61	treated
10	3.93	treated
11	4.36	treated
12	3.93	treated
13	3.89	treated
14	4.03	treated
15	3.85	treated
16	4.06	treated

데이터 입력

```
import pandas as pd
import numpy as np

prac_group = ["control"]*5 + ["treated"]*11
id = list(range(1,17))
score = np.array([3.81, 4.47, 4.81, 4.79, 4.25, 3.93, 4.26, 3.74,
                  3.61, 3.93, 4.36, 3.93, 3.89, 4.03, 3.85, 4.06])
prac1 = pd.DataFrame({"id":id, "score":score, "group":prac_group})
prac1
```

```
##     id  score    group
## 0    1   3.81  control
## 1    2   4.47  control
## 2    3   4.81  control
## 3    4   4.79  control
## 4    5   4.25  control
## 5    6   3.93  treated
## 6    7   4.26  treated
## 7    8   3.74  treated
## 8    9   3.61  treated
## 9   10   3.93  treated
```

```
## 10  11   4.36  treated
## 11  12   3.93  treated
## 12  13   3.89  treated
## 13  14   4.03  treated
## 14  15   3.85  treated
## 15  16   4.06  treated
```

```
control = prac1[prac1['group']=='control']['score']
treated = prac1[prac1['group']=='treated']['score']
```

귀무가설 vs. 대립가설

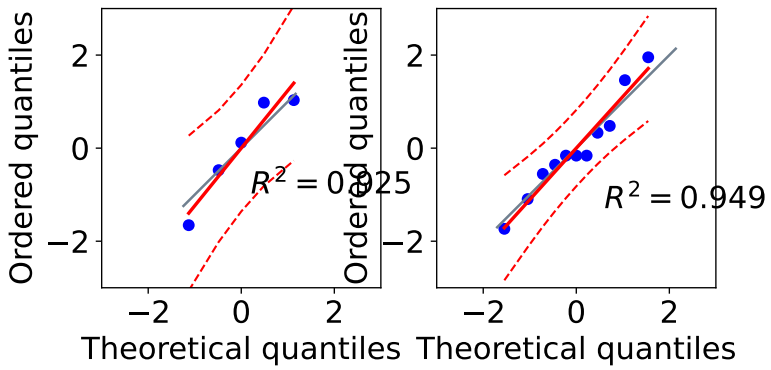
- 귀무가설: 혈압약의 사용은 혈압을 떨어뜨리는 효과가 없다. $\mu_{control} = \mu_{treated}$
- 대립가설: 혈압약의 사용은 혈압을 떨어뜨리는 효과가 있다. $\mu_{control} > \mu_{treated}$

가정 체크 및 검정 방법 설정

2표본 t 검정은 데이터가 정규성을 만족하고, 등분산성의 만족 유무에 따라서 선택할 수 있는 검정 방법이 달라지므로 검정의 가정을 체크한다.

정규성 검정 각 그룹에 대한 정규성 검정을 시각화 기법과 Shapiro-wilk 검정을 사용하여 시행한다.

```
import pingouin as pg
import matplotlib.pyplot as plt

plt.subplot(1,2,1);
ax = pg.qqplot(control, dist='norm')
plt.ylim(-3, 3);
plt.xlim(-3, 3);

plt.subplot(1,2,2);
ax = pg.qqplot(treated, dist='norm')
plt.ylim(-3, 3);
plt.xlim(-3, 3);

plt.show()
```



몇 개의 표본을 제외하면 두 그룹 모두 정규성을 띄는 것으로 판단된다. 정확한 검정을 위하여 유의수준 5% 하에서 Shapiro-Wilk 검정을 시행한다.

- 귀무가설: 데이터가 정규분포를 따른다.
- 대립가설: 데이터가 정규분포를 따르지 않는다.

```
from scipy.stats import shapiro
shapiro(prac1.iloc[:5,1])
```

```
## ShapiroResult(statistic=0.9128482341766357, pvalue=0.4848945438861847)
```

```
shapiro(prac1.iloc[5:16,1])
```

```
## ShapiroResult(statistic=0.9562556147575378, pvalue=0.7242791652679443)
```

두 그룹 데이터 모두 p-value 값이 유의수준인 5%보다 크므로 귀무가설을 기각할 수 없다. 따라서 두 그룹 데이터 모두 정규성을 만족한다고 판단한다.

등분산성 검정 두 그룹이 정규성을 만족한다는 것을 확인하였으므로, 유의수준 5%하에서 F 검정을 통하여 등분산성 가정을 확인한다.

- 귀무가설: 두 그룹의 모분산이 같다.
- 대립가설: 두 그룹의 모분산은 같지 않다.

```
import numpy as np
import scipy.stats as stats

def f_test(x, y):
    x = np.array(x)
    y = np.array(y)
```

```
    f = np.var(x, ddof=1) / np.var(y, ddof=1) # 검정통계량
    dfn = x.size-1
    dfd = y.size-1
    p = stats.f.cdf(f, dfn, dfd)
    p = 2*min(p, 1-p) # two sided p-value
    return f, p

group1 = prac1[prac1["group"] == "control"]["score"]
group2 = prac1[prac1["group"] == "treated"]["score"]

# Perform F-test
f_value, p_value = f_test(group1, group2)

print("Test statistic: {:.3f}".format(f_value))
```

```
## Test statistic: 3.800
```

```
print("p-value: {:.3f}".format(p_value))
```

```
## p-value: 0.079
```

F 검정통계량 값 3.8에 대응하는 p-value 0.079가 유의수준인 5% 보다 크므로 귀무가설을 기각할 수 없다. 따라서 두 그룹 데이터의 모분산은 동일하다고 판단한다.

t 검정

표본의 정규성과 등분산성을 만족하므로 독립 2표본 t-test를 실시한다.

```
from scipy.stats import ttest_ind
t_value, p_value = ttest_ind(group1, group2, equal_var = True, alternative='greater')

print("Test statistic: ", t_value.round(3))
```

```
## Test statistic:  3.0
```

```
print("p-value: ", p_value.round(3))
```

```
## p-value:  0.005
```

컨트롤 그룹의 표본 평균 혈압은 4.426이고, 혈압약을 복용한 그룹의 표본 평균 혈압은 3.96을 나타냈다.

t 검정의 p-value 0.005가 유의수준인 5%보다 작으므로, 귀무가설을 기각한다. 즉, 혈압약 복용/미복용 두 그룹의 평균 혈압 차는 통계적으로 유의미하다고 이야기 할 수 있다. 따라서 혈압약의 사용이 혈압을 떨어뜨리는 효과가 있다고 판단한다.

문제 2. 고양이 소변의 효과

고양이의 소변이 식물의 성장을 저해한다는 명제를 확인하기 위하여 무작위로 선정된 두 식물 그룹을 가지고 실험을 진행하였다.

Treated 그룹의 경우 고양이 소변을 체취하여 일정 간격으로 뿌려주었으며, 이외 다른 조건은 Treated, Control 그룹 모두 동일하게 유지하였다. 표 11.3는 한달 후 두 그룹의 성장 높이 데이터를 기록한 것이다.

고양이의 소변이 식물의 성장을 저해한다는 것에 대한 검정을 유의수준 5%하에서 수행해보시오.

표 11.3: 식물 성장 측정 데이터

ID	Score	Group
1	45	control
2	87	control
3	123	control
4	120	control
5	70	control
6	51	treated
7	71	treated
8	42	treated
9	37	treated
10	51	treated
11	78	treated
12	51	treated
13	49	treated
14	56	treated
15	47	treated
16	58	treated

데이터 입력

```
prac_group = ["control"]*5 + ["treated"]*11
id = list(range(1,17))
score = [45, 87, 123, 120, 70,51, 71, 42, 37, 51, 78,51, 49, 56, 47, 58]
prac2 = pd.DataFrame({"id":id, "score":score, "group":prac_group})
prac2
```

```
##     id  score    group
## 0    1     45  control
## 1    2     87  control
## 2    3    123  control
## 3    4    120  control
## 4    5     70  control
## 5    6     51  treated
## 6    7     71  treated
```

```
## 7    8     42  treated
## 8    9     37  treated
## 9   10     51  treated
## 10  11     78  treated
## 11  12     51  treated
## 12  13     49  treated
## 13  14     56  treated
## 14  15     47  treated
## 15  16     58  treated
```

```
control = prac2[prac2['group']=='control']['score']
treated = prac2[prac2['group']=='treated']['score']
```

귀무가설 vs. 대립가설

귀무가설: 고양이의 소변은 식물의 성장에 무해하다.

$$\mu_{control} = \mu_{treated}$$

대립가설: 고양이의 소변은 식물의 성장에 저해한다.

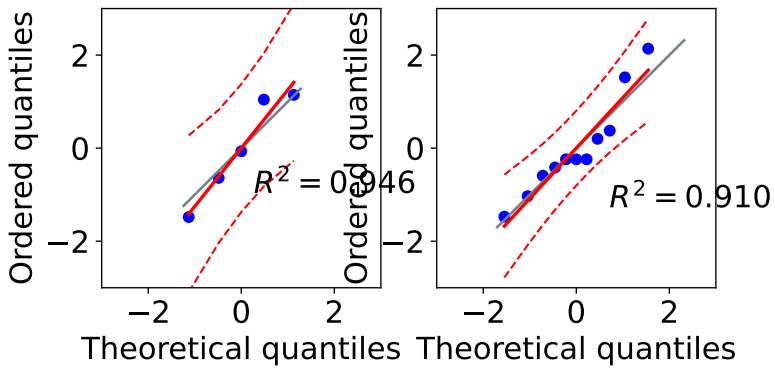
$$\mu_{control} > \mu_{treated}$$

가정 체크 및 검정 방법 설정

2표본 t 검정은 데이터가 정규성을 만족하고, 등분산성의 만족 유무에 따라서 선택할 수 있는 검정 방법이 달라지므로 검정의 가정을 체크한다.

정규성 검정 각 그룹에 대한 정규성 검정을 시각화 기법과 Shapiro-wilk 검정을 사용하여 시행한다.

```
import pingouin as pg
import matplotlib.pyplot as plt

plt.subplot(1,2,1);
ax = pg.qqplot(control, dist='norm')
plt.ylim(-3, 3);
plt.xlim(-3, 3);

plt.subplot(1,2,2);
ax = pg.qqplot(treated, dist='norm')
plt.ylim(-3, 3);
plt.xlim(-3, 3);

plt.show()
```



Control 그룹의 경우 정규성을 띄는 것으로 판단되나 Treated 그룹의 두 표본이 정규성을 만족하지 않는 것으로 보인다. 정확한 검정을 위하여 유의수준 5% 하에서 Shapiro-Wilk 검정을 시행한다.

- 귀무가설: 데이터가 정규분포를 따른다.
- 대립가설: 데이터가 정규분포를 따르지 않는다.

```
from scipy.stats import shapiro
shapiro(prac2.iloc[:5,1])
```

```
## ShapiroResult(statistic=0.9245066046714783, pvalue=0.5594186186790466)
```

```
shapiro(prac2.iloc[5:16,1])
```

```
## ShapiroResult(statistic=0.9170808792114258, pvalue=0.2950724959373474)
```

두 그룹 데이터 모두 p-value 값, 0.55와 0.29가 이 유의수준인 5%보다 크므로 귀무가설을 기각할 수 없다. 따라서 두 그룹 데이터 모두 정규성을 만족한다고 판단한다.

등분산성 검정 두 그룹이 정규성을 만족한다는 것을 확인하였으므로, 유의수준 5%하에서 F 검정을 통하여 등분산성 가정을 확인한다.

- 귀무가설: 두 그룹의 모분산이 같다.
- 대립가설: 두 그룹의 모분산은 같지 않다.

```
import numpy as np
import scipy.stats as stats

group1 = prac2[prac2["group"] == "control"]["score"]
group2 = prac2[prac2["group"] == "treated"]["score"]
```

```
#perform F-test
f_value, p_value = f_test(group1, group2)

print("Test statistic: {:.3f}".format(f_value))
```

```
## Test statistic: 7.788
```

```
print("p-value: {:.3f}".format(p_value))
```

```
## p-value: 0.008
```

F 검정의 p-value값 0.0081이 유의수준인 5% 보다 작으므로 귀무가설을 기각한다. 따라서 두 그룹 데이터의 등분산 가정을 만족하지 않는다고 판단한다.

t 검정

주어진 표본은 정규성을 만족하나 등분산성을 만족하지 못하므로 Welch's two sample t-test를 실시한다.

```
from scipy.stats import ttest_ind
import math

t_value, p_value = ttest_ind(group1, group2,
                             equal_var = False,
                             alternative='greater')

print("control mean: ", np.mean(group1).round(3))
```

```
## control mean:  89.0
```

```
print("treated mean: ", np.mean(group2).round(3))
```

```
## treated mean:  53.727
```

```
print("Test statistic: ", t_value.round(3))
```

```
## Test statistic:  2.307
```

```
print("p-value: ", p_value.round(3))
```

```
## p-value:  0.038
```

컨트롤 그룹 식물들의 표본 평균 높이는 89.0이고, 고양이 소변을 뿌린 식물 그룹의 표본 평균 높이는 53.727을 나타냈다.

t 검정통계량 값 2.3069에 대응하는 p-value값 0.03767이 유의수준인 5%보다 작으므로, 귀무가설을 기각한다. 즉, 고양이 소변을 뿌린 그룹과 그렇지 않은 식물 그룹의 평균 높이 차는 주어진 유의수준 하에서 통계적으로 유의미하다고 말할 수 있다. 따라서 고양이 소변이 식물 성장을 저해하는 효과가 있다고 판단한다.

Chapter 4. 비모수 검정 친해지기

문제 1. 신제품 촉매제

슬통 회사에서는 이번에 출시한 새로운 촉매제의 효능을 검증하고 싶어한다. 신제품 촉매재는 기존 공정에서 사용되는 화학반응 속도를 혁신적으로 줄여주는 기능이 탑재되어 있다고 한다.

회사 제품 검증 부서에서는 기존 공정의 화학 반응속도와 촉매제를 넣은 후의 반응 속도를 측정하여 표 4.4 데이터를 만들었다.

1. 유의수준 5%하에서 신제품 촉매제가 기존의 화학 공정을 단축시킨다고 할 수 있는지에 대하여 검정하시오.
2. 촉매제로 인한 단축된 공정 시간에 대하여 90% 신뢰구간을 구하시오.

```
import pandas as pd
import numpy as np

# 파이썬 코드
time = [2.17, 0.86, 0.91, 3.11, 1.29, 1.25, 0.76, 2.98, 1.21,
2.23, 0.67, 1.22, 1.23, 1.21, 1.71, 1.80, 1.41, 1.01,
0.82, 1.03, 2.03, 0.65, 1.01, 0.45, 0.98, 1.04]

treat = ['before']*13 + ['add_catalyst']*13
id = list(range(1,27))
prac4_1 = pd.DataFrame({'ID':id,
                        'Time':time,
                        'treat':treat})
prac4_1.head()
```

```
##    ID  Time   treat
## 0   1  2.17  before
## 1   2  0.86  before
## 2   3  0.91  before
## 3   4  3.11  before
## 4   5  1.29  before
```

표 11.4: 촉매제 성능 비교 데이터

ID	Time	Treat
1	2.17	before
2	0.86	before
3	0.91	before
4	3.11	before
5	1.29	before
6	1.25	before
7	0.76	before
8	2.98	before
9	1.21	before
10	2.23	before
11	0.67	before
12	1.22	before
13	1.23	before
14	1.21	add_catalyst
15	1.71	add_catalyst
16	1.80	add_catalyst
17	1.41	add_catalyst
18	1.01	add_catalyst
19	0.82	add_catalyst
20	1.03	add_catalyst
21	2.03	add_catalyst
22	0.65	add_catalyst
23	1.01	add_catalyst
24	0.45	add_catalyst
25	0.98	add_catalyst
26	1.04	add_catalyst

가설 설정

- H_0: 신제품 촉매제는 화학 공정 시간을 단축시키지 않는다.
  - $\mu_{before} \leq \mu_{catalyst}$
- H_A: 신제품 촉매제는 화학 공정 시간을 단축시킨다.
  - $\mu_{before} > \mu_{catalyst}$

정규성 검정

각 그룹에 대한 정규성 검정을 시각화 기법과 Shapiro-wilk 검정을 사용하여 시행한다.

```
before = prac4_1[prac4_1['treat']=='before']
add_catalyst = prac4_1[prac4_1['treat']=='add_catalyst']

import pingouin as pg
import matplotlib.pyplot as plt
```

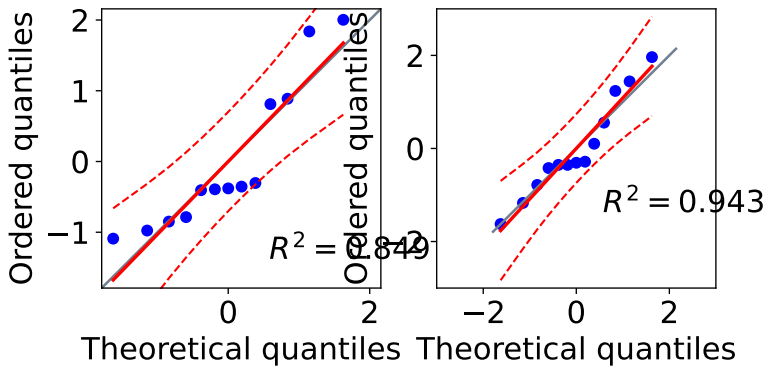
```
plt.subplot(1,2,1)
```

```
## <AxesSubplot:>
```

```
ax = pg.qqplot(before['Time'], dist='norm')

plt.subplot(1,2,2)
```

```
## <AxesSubplot:>
```

```
ax = pg.qqplot(add_catalyst['Time'], dist='norm')
plt.ylim(-3, 3);
plt.xlim(-3, 3);
plt.show()
```



두 그룹의 QQ plot을 판단해 보았을 때, 신뢰구간 안에 모든 데이터가 들어있으나 가운데 부분에서 두 그룹 모두 기준선을 벗어나는 경향성을 보인다. 좀 더 정확한 검정을 위하여 유의수준 5% 하에서 Shapiro-Wilk 검정을 시행한다.

- Shapiro-Wilk 검정의 귀무가설과 대립가설은 다음과 같다.
  - 귀무가설: 데이터가 정규분포를 따른다.
  - 대립가설: 데이터가 정규분포를 따르지 않는다.

```
from scipy.stats import shapiro
shapiro(before['Time'])
```

```
## ShapiroResult(statistic=0.8367362022399902, pvalue=0.019262485206127167)
```

```
shapiro(add_catalyst['Time'])
```

```
## ShapiroResult(statistic=0.941055178642273, pvalue=0.4707110524177551)
```

두 그룹 중 before 그룹에 대응하는 p-value 값 0.01926이 유의수준인 5%보다 작으므로 귀무가설을 기각한다. 따라서 데이터가 정규성을 따른다고 판단할 수 없으므로, 비모수 검정을 진행하도록 한다. 따라서, 앞에서 설정한 귀무가설, 대립가설의 모수가 모분포의 중앙값을 나타내는 것에 주의하자.

등분산성 검정 유의수준 5%하에서 두 그룹 데이터의 등분산성 가정을 Levene's test를 통하여 확인한다.

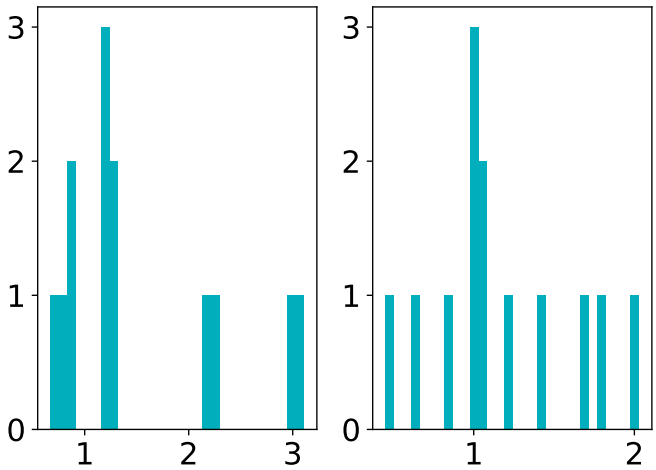
```
plt.subplot(1,2,1)
```

```
## <AxesSubplot:>
```

```
plt.hist(before['Time'], bins=30);

plt.subplot(1,2,2)
```

```
## <AxesSubplot:>
```

```
plt.hist(add_catalyst['Time'],bins=30);
plt.show();
```



before 그룹의 히스토그램으로 판단했을때 정규성이 보장되지 않는 데이터이며, 오른쪽으로 살짝 치우쳐있는 데이터 이므로, 좀 더 robust한 검정을 위하여 `center` 옵션은 `median`으로 설정한 후 진행하였다.

- 귀무가설: 두 그룹의 모분산이 같다.
- 대립가설: 두 그룹의 모분산은 같지 않다.

```
from scipy.stats import levene

a = before['Time']
b = add_catalyst['Time']
levene(b, a, center='median')
```

```
## LeveneResult(statistic=1.404486168289024, pvalue=0.24757651251351717)
```

검정의 p-value값 0.2476이 유의수준인 5% 보다 크므로 귀무가설을 기각하지 못한다. 따라서 두 그룹 데이터는 등분산 가정을 만족한다고 판단하였다.

위와 같은 이유로, 가장 적합한 검정은 Mann-Whitney-Wilcoxon U test로 판단하여 검정을 진행한다.

- H_0: 신제품 촉매제는 화학 공정 시간을 단축시키지 않는다.
  - $\eta_{before} = \eta_{catalyst}$
- H_A: 신제품 촉매제는 화학 공정 시간을 단축시킨다.
  - $\eta_{before} > \eta_{catalyst}$

```
from scipy.stats import mannwhitneyu

mannwhitneyu(before['Time'],add_catalyst['Time'], alternative='greater')
```

```
## MannwhitneyuResult(statistic=106.5, pvalue=0.1350263415415442)
```

검정에 대응하는 p-value 값 0.135가 유의수준 0.05보다 크므로, 귀무가설을 기각하지 못한다. 따라서 신제품 촉매제가 기존 화학 공정 시간을 단축 시키지 않는다고 판단한다.

- 참고: 데이터에 tie가 존재하는 경우 `mannwhitneyu()` 함수는 정규근사에 의한 유의 확률을 계산한다.

```
len(prac4_1['Time']) - len(prac4_1["Time"].unique())
```

```
## 2
```

신뢰구간

비모수 검정의 정확한 신뢰구간을 구하는 것은 상당히 까다롭다. 따라서 주어진 표본으로 가능한 감소 시간의 표본들을 만들고, 이를 통하여 유추하도록 한다.

```
# 'before' 그룹과 'add_catalyst' 그룹의 score 값을 추출
u = prac4_1[prac4_1['treat'] == 'before']['Time'].values
v = prac4_1[prac4_1['treat'] == 'add_catalyst']['Time'].values
```

```
# u와 v의 가능한 모든 조합 생성
df = pd.DataFrame({'Var1': np.repeat(u, len(v)),
                   'Var2': np.tile(v, len(u))})

 # 각 조합에 대해 Var1과 Var2의 차이 계산
m = df['Var1'] - df['Var2']

# m의 5% 및 95% 분위수 계산
quantiles = m.quantile([0.05, 0.95])
quantiles
```

```
## 0.05   -0.920
## 0.95    2.092
## dtype: float64
```

따라서 우리는 촉매제로 인한 공정 시간의 분포 실제 중앙값 감소 시간은 $(-0.920, 2.092)$ 구간에 존재할 것이라 90% 신뢰한다.

문제 2. 심장 질환 약 효능

슬통제약의 신약이 심장 질환 환자의 혈압을 낮출 수 있는지 검증하려고 한다. 표본으로 15명의 환자가 선택되었으며, 약을 복용하기 전과 복용한 후의 혈압을 측정하였다.

- 복용전: 130, 125, 120, 135, 140, 136, 129, 145, 150, 135, 128, 140, 139, 130, 145
- 복용후: 125, 120, 115, 130, 135, 134, 128, 140, 145, 134, 127, 140, 138, 129, 142

약이 혈압을 실제로 낮추는 것인지 검증하기 위하여 부호 검정을 실시하라. 유의 수준은 0.05로 하고, 검정을 수행하시오.

예시 답안

부호 검정은 각 쌍의 차이에 대해 부호를 검사하여, 이 경우에는 복용 전, 후의 혈압 차이가 0인지 아니면 혈압이 낮아졌는지(전 - 후 혈압 차이가 0보다 큰 지)를 검사하고자 합니다. 진행하고자 하는 검정의 귀무가설과 대립가설은 다음과 같습니다.

- 귀무 가설: 약 복용 전과 복용 후 혈압차 분포의 중앙값은 0이다.
  - $\Delta := \eta_{pair_1} - \eta_{pair_2} = 0$
- 대립 가설: 약 복용 전과 복용 후 혈압차 분포의 중앙값은 0보다 크다.
  - $\Delta > 0$

먼저, 각 환자의 혈압 차이(복용 전 - 복용 후)를 계산하고, 양수와 0의 개수를 세어보겠습니다.

```
from scipy.stats import binom_test
import numpy as np
```

```
# 복용 전 후의 혈압
before_np = np.array([130, 125, 120, 135, 140, 136, 129,
                      145, 150, 135, 128, 140, 139, 130, 145])
after_np = np.array([125, 120, 115, 130, 145, 134, 128,
                     140, 145, 134, 127, 140, 140, 129, 142])

# 차이 계산
diff = before_np - after_np

# 차이가 0보다 큰 경우를 세기 (혈압이 낮아진 경우)
successes = sum(diff > 0)
successes
```

```
## 12
```

Python에서 제공하는 sign_test() 함수의 경우 단측검정 옵션을 제공하고 있지 않습니다.

```
from statsmodels.stats.descriptivestats import sign_test

# statsmodels의 sign_test 함수를 사용하여 부호 검정 수행
test_statistic, p_value = sign_test(diff, mu0=0)
test_statistic, p_value
```

```
## (5.0, 0.012939453125)
```

binom_test() 함수를 사용하여 단측검정 p-vlaue를 계산하면 다음과 같습니다.

```
# 이항 검정 수행
# from scipy.stats import binom
# 1-binom.cdf(11, 14, 0.5)
p_value = binom_test(successes, 14, alternative='greater')

successes, len(diff[diff ≠ 0]), p_value
```

```
## (12, 14, 0.0064697265625)
```

이항분포 (14, 0.5)에서 검정 통계량 값 12보다 크거나 같은 값이 나올 확률은 0.6%로 유의 수준 5%보다 낮아 귀무가설을 기각한다. 따라서 신약이 심장 질환 환자의 혈압을 낮출 수 있다고 판단한다.

Chapter 5. 카이제곱 검정 친해지기

문제 1. 휴대전화 사용자들의 정치 성향은 다를까?

다음은 휴대전화와 유선전화를 모두 사용하는 사용자들과 유선전화만을 사용하는 사용자들의 정치 성향을 조사한 데이터이다. 유의수준 5%하에서 정당 지지와 핸드폰 사용 유무 사이에 상관성을 검정해보세요.

표 11.5: 정치 성향 설문조사 결과

정당지지	핸드폰	유선전화
진보	49	47
중도	15	27
보수	32	30

데이터 입력

```
import pandas as pd
data = [[49,47],[15,27],[32,30]]
columns = ["핸드폰", "유선전화"]
index = ["진보", "중도", "보수"]

phone_data = pd.DataFrame(data, columns=columns, index=index)
phone_data
```

```
##     핸드폰  유선전화
## 진보   49    47
## 중도   15    27
## 보수   32    30
```

귀무가설 vs. 대립가설

- H_0: 핸드폰 사용 여부와 정당 지지 성향은 독립이다.
- H_A: 핸드폰 사용 여부와 정당 지지 성향은 독립이 아니다.

기대빈도 체크하기 독립성 검정은 각 셀의 기대빈도가 모두 5 이상 되어야 결과를 신뢰할 수 있다. 아래의 결과를 살펴보면, 각 셀의 기대빈도가 모두 5 이상인 것을 확인할 수 있다.

```
from scipy.stats import chi2_contingency
result = chi2_contingency(phone_data)

result[3]
```

```
## array([[46.08, 49.92],
##        [20.16, 21.84],
##        [29.76, 32.24]])
```

검정통계량과 p-value

```
x_squared, p_value, df, expected = result

print('x-squared:',x_squared)
```

```
## x-squared: 3.2199060739887355
```

```
print('p-value:',p_value)
```

```
## p-value: 0.19989700161872206
```

카이제곱 통계량 3.219에 대응하는 p-value 0.199는 유의수준 5%보다 크므로, 귀무가설을 기각하지 못한다. 따라서, 휴대폰 사용여부는 정당지지와는 관련이 없다 (독립이다) 라고 판단한다.

문제 2. 여자아이 vs. 남자아이

다음은 4자녀를 둔 130가구를 조사하여 여자아이의 수를 조사한 자료이다. 여자 아이의 출생 비율이 50% 인지 유의수준 5%하에서 검정해보세요.

표 11.6: 4자녀 가정 여자아이 숫자 조사 결과

Girl	Frequency
0	10
1	31
2	44
3	34
4	11

이항분포

확률변수 X가 이항분포 n, p를 따를 때, X의 확률질량함수는 다음과 같다.

$$p\left(x; n, p\right) = \left(\begin{array}{c} n \\ x \end{array} \right) p^x \left(1-p\right)^{n-x}$$

귀무가설 vs. 대립가설

- H_0: 여자아이의 출생율은 0.5이다.
- H_A: 여자아이의 출생율은 0.5가 아니다.

기대빈도

귀무가설 하에서 4자녀 가정에서 여자아이의 숫자는 이항분포 $n = 4, p = 0.5$에서 관찰된 관찰값이라 생각할 수 있다. 따라서 4자녀 가구에서 여자아이가 0명에서 4명이 있을 확률을 다음과 같이 구할 수 있다.

```
from scipy.stats import binom

binom.pmf(range(5), 4, 0.5)
```

```
## array([0.0625, 0.25  , 0.375 , 0.25  , 0.0625])
```

위의 확률을 사용하여 기대빈도를 구하면 다음과 같다.

```
130 * binom.pmf(range(5), 4, 0.5)
```

```
## array([ 8.125, 32.5  , 48.75 , 32.5  ,  8.125])
```

기대 빈도가 모두 5이상이므로 카이제곱 검정을 실시할 수 있다.

적합도검정

chisq.test()에서는 다음과 같이 관찰값과 대응하는 확률값을 사용하여 검정을 진행할 수 있다.

```
from scipy.stats import chisquare
import numpy as np

observed_frequencies = [10, 31, 44, 34, 11]

expected_frequencies = binom.pmf(range(5), 4, 0.5) * sum(observed_frequencies)
chisquare(observed_frequencies, expected_frequencies)
```

```
## Power_divergenceResult(statistic=2.0512820512820507, pvalue=0.726327096627745)
```

검정통계량 2.0513에 대응하는 유의확률인 p-value 값 0.7263이 유의수준 5%보다 크므로, 귀무가설을 기각하지 못한다. 따라서 여자아이 출생비율이 50%라고 판단한다.

문제 3. 지역별 대선 후보의 지지율

어느 도시에 있는 3개의 선거구에서 특정후보 A를 지지하는 유권자의 비율을 비교하기 위해 각 선거구에서 300명을 무작위를 추출하여 조사한 데이터이다. 주어진 데이터를 대상으로 후보A를 지지하는 비율이 3개 선거구 간에 차이가 있는지를 5% 유의수준에서 검정하라.

표 11.7: 지역별 대선 후보의 지지율

구분	선거구 1	선거구 2	선거구 3
지지함	176	193	159
지지하지 않음	124	107	141

귀무가설 vs. 대립가설

- H_0: 각 선거구 별 A 후보의 지지율은 동일하다. $p_1 = p_2 = p_3$
- H_A: 지지율이 다른 선거구가 적어도 하나 존재한다.

p-value 계산 및 결론 도출

```
import numpy as np
from scipy.stats import chi2_contingency

# 데이터 설정: 교차표
data = np.array([[176, 124],  # 선거구 1
                 [193, 107],  # 선거구 2
                 [159, 141]]) # 선거구 3
chi2, p, df, expected = chi2_contingency(data)
expected
```

```
## array([[176., 124.],
##        [176., 124.],
##        [176., 124.]])
```

각 셀의 기대빈도가 5보다 크므로 카이제곱 동질성 검정의 결과를 신뢰 할 수 있다.

```
print(chi2.round(3), p.round(3))
```

```
## 7.945 0.019
```

검정통계량 7.9454에 대응하는 p-value 값 0.019가 유의수준 0.05보다 작으므로, 귀무가설을 기각한다. 따라서 모든 선거구의 지지율이 같지 않다고 판단할 통계적 근거가 충분하다.

문제 4. 데이터가 특정분포를 따를까?

다음의 데이터가 주어졌을때, 카이제곱 검정법을 사용하여 데이터가 모수가 2인 지수분포를 따르는지 유의수준 5% 하에서 검정해보세요.

0.211, 0.098, 0.736, 0.091, 0.756, 1.039, 0.391, 0.172, 2.113, 0.013, 0.073, 0.812, 0.132, 0.263, 0.124, 0.339, 0.092, 0.24, 0.438, 0.584, 0.722, 0.231, 0.033, 0.203, 0.177, 0.095, 0.352, 0.023

- 데이터는 3개 구간 (0, 0.2], (0.2, 0.4], (0.4, Inf)을 사용해서 검정하세요.

데이터 구간 나누기

```
x = [0.211, 0.098, 0.736, 0.091, 0.756, 1.039, 0.391, 0.172,
2.113, 0.013, 0.073, 0.812, 0.132, 0.263, 0.124, 0.339,
0.092, 0.24, 0.438, 0.584, 0.722, 0.231, 0.033, 0.203,
0.177, 0.095, 0.352, 0.023]

result = pd.cut(x,
                bins=[0, 0.2, 0.4, float("inf")],
                labels=["0,0.2", "0.2,0.4", "0.4,Inf"])

print(result.value_counts())
```

```
## 0,0.2      12
## 0.2,0.4     8
## 0.4,Inf     8
## dtype: int64
```

기대빈도 구하기

주어진 구간에 대응하는 확률을 구하면 다음과 같다.

```
from scipy.stats import expon
import numpy as np

x = [0.2, 0.4, float("inf")]
rate = 2

exp_cdf = expon.cdf(x, scale=1/rate)
exp_p = exp_cdf - np.insert(exp_cdf, 0, 0)[:-1]
print(exp_p)
```

```
## [0.32967995 0.22099108 0.44932896]
```

주어진 확률로 기대빈도를 구하면 다음과 같다.

```
exp_p * 28
```

```
## array([ 9.23103871,  6.18775029, 12.581211  ])
```

각 셀의 기대빈도 값이 5보다 크므로, 카이제곱 적합도 검정을 수행할 수 있다.

적합도 검정

```
from scipy.stats import chisquare
chi2, p_value = chisquare(f_obs=result.value_counts(), f_exp=exp_p * 28)

print("Chi-square statistic:", chi2)
```

```
## Chi-square statistic: 3.0295112382237264
```

```
print("p-value:", p_value)
```

```
## p-value: 0.2198619083314235
```

검정통계량 값 3.0295에 대응하는 유의확률 0.2199는 유의수준 5%보다 크므로, 귀무가설을 기각할 수 없다.

설문조사 환자 수

슬통 병원에서는 최근 도입한 진료 서비스에 대한 환자 만족도를 평가하고자 합니다. 병원은 환자들에게 만족도 설문지를 무작위로 분배하려고 합니다. 설문을 시작하기 전에, 병원은 적절한 표본 크기를 결정하려고 합니다.

설문지는 환자의 진료 서비스에 대한 만족도를 7점 척도로 묻습니다. 병원은 특히 환자 중 만족하거나 매우 만족하는 비율 p에 관심이 있습니다 (이는 7점 척도 중 상위 두 단계에 해당합니다).

병원은 만족도 추정의 신뢰 수준을 95%로 설정하려고 하며, 오차 범위는 3% 또는 0.03 이하로 원합니다. 보수적인 추정을 위하여 확률은 0.5로 가정하고자 합니다. 조건 만족을 위하여 설문에 필요한 최소 환자 수를 구해주세요.

공식을 사용한 표본수 구하기

신뢰구간 공식을 사용하여 표본수와 관련한 다음과 같은 공식을 유도할 수 있다.

$$\frac{\hat{p}(1-\hat{p})}{\left(\frac{0.03}{z_{\alpha/2}}\right)^2} \leq n$$

여기에 $\hat{p}$ 값을 0.5로 가정하고 n 구하면 다음과 같다.

```
from scipy.stats import norm

p_hat = 0.5
p_hat * (1-p_hat) / (0.03 / norm.ppf(0.975))**2
```

```
## 1067.0718946372572
```

따라서 필요한 표본 수는 1068개 이다.

설문조사 환자 수 2

다음은 슬통 병원에서 작년에 조사해놓은 환자들의 만족도 조사 결과 입니다.

7, 7, 6, 7, 6, 3, 6, 4, 5, 2, 7, 6, 6, 6, 6, 6, 6, 6, 7, 6

위의 데이터를 사용하여 설문에 필요한 최소 환자 수를 구해주세요. 병원은 만족도 추정의 신뢰수준을 98%로 설정하려고 하며, 오차 범위는 2% 또는 0.02 이하로 원합니다.

표본 수 구하기 (데이터 사용)

표본을 구하는 것은 위에서 사용한 공식을 그대로 이용할 수 있다. 주어진 데이터에서 6점과 7점의 비율을 구하자.

```
import numpy as np

# Convert data to numpy array
data_np = np.array([7, 7, 6, 7, 6, 3, 6, 4, 5, 2,
                    7, 6, 6, 6, 6, 6, 6, 7, 6])

# Calculate the proportion using numpy
p_hat_np = np.mean((data_np == 6) | (data_np == 7))
p_hat_np
```

```
## 0.8
```

만족도 추정값은 0.8로 설정한다.

```
from scipy.stats import norm

p_hat = 0.8
p_hat * (1-p_hat) / (0.02 / norm.ppf(0.99))**2
```

```
## 2164.757772421736
```

따라서 주어진 조건을 만족하기 위한 필요한 총 표본수는 2165명이다.

유권자의 마음

슬통 신문사에서는 다음과 같은 42명의 시민들을 대상으로 지난 1월 A 대통령 후보를 지지하는 조사하였습니다. 다음은 대통령이 된 A 후보의 임기 시작 후 6개월이 지난 오늘, 다시 한번 동일 인원들에게 전화를 걸어 대통령 후보를 지지하는지 물어본 결과입니다.

	당선 후 지지여부	
당선 전 지지여부	지지함	지지하지 않음
지지함	17	7

	당선 후 지지여부	
지지하지 않음	5	13

사람들의 A 후보에 대한 지지율이 당선 전과 당선 후 변하였는지 검정해보세요.

McNemar 검정

McNemar 검정은 2x2 교차표(contingency table)에 대한 카이제곱 검정의 특별한 경우입니다. 주로, 두 시점 또는 두 조건에서 **동일한 대상**들의 범주형 응답을 비교할 때 사용됩니다.

	전처리 결과 A	전처리 결과 B
처리 전 A	a	b
처리 전 B	c	d

- 검정 통계량: $\chi^2 = \frac{(b-c)^2}{b+c} \sim \chi^2_{(1)}$

먼저, McNemar 검정의 귀무 가설과 대립 가설을 설정합니다:

귀무 가설 : 당선 전과 당선 후의 지지율은 동일하다. 대립 가설 : 당선 전과 당선 후의 지지율은 다르다.

```
from scipy.stats import chi2

b = 7
c = 5

# 검정통계량 계산
stat = (b - c)**2 / (b + c)
stat

# 유의확률
```

```
## 0.3333333333333333
```

```
1 - chi2.cdf(stat, 1)
```

```
## 0.5637028616507731
```

검정통계량 값 0.3에 대응하는 p-value 값이 유의 수준 0.05보다 크므로 귀무가설을 기각하지 못한다. 따라서 당선 전과 후의 지지율은 동일하다고 판단한다.

statsmodels 버전 (참고)

```python
from statsmodels.stats.contingency_tables import mcnemar

# 주어진 데이터
observed = [[17, 7], [5, 13]]

# McNemar 검정 수행 (옵션 2개 꼭 꺼줘야 함)
result = mcnemar(observed, exact=False, correction=False)

print("검정 통계량:", result.statistic)
```

```
## 검정 통계량: 0.3333333333333333
```

```python
print("p-값:", result.pvalue)
```

```
## p-값: 0.5637028616507731
```

Chapter 6. 분산분석 친해지기

펭귄의 부리길이

palmerpenguins 패키지의 penguins 데이터에는 펭귄 종류별 부리길이 (bill_length_mm) 정보가 들어있다. 펭귄의 종류에 따라서 부리길이가 다르다고 할 수 있는지 유의수준 1% 하에서 검정해보세요.

```python
from palmerpenguins import load_penguins
penguins = load_penguins()
penguins.isnull().sum()
```

```
## species              0
## island               0
## bill_length_mm       2
## bill_depth_mm        2
## flipper_length_mm    2
## body_mass_g          2
## sex                 11
## year                 0
## dtype: int64
```

변수 별 결측치 정보를 조사해 보았을 때, 독립 변수인 부리길이가 결측인 데이터 2개가 존재한다. 이를 제외한 나머지 데이터를 사용하여 분석을 진행하도록 하자.

```
my_penguins = penguins[['species', 'bill_length_mm']].dropna()
print(my_penguins.shape)
```

```
## (342, 2)
```

총 342개의 관찰값이 존재한다. 펭귄 종 별 부리 길이의 모평균이 다른지 검정하기 위하여 ANOVA를 진행하도록 한다.

귀무 vs. 대립가설

- 귀무가설: 펭귄 종별 부리길이 평균은 동일하다.
  - $\mu_{adelie} = \mu_{chinstrap} = \mu_{gentoo}$
- 대립가설: 펭귄 종 간 부리 길이가 다른 그룹이 적어도 하나 존재한다.
  - Not all of the μ_i are equal

Python에서 ANOVA 테이블 구하기

```
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols('bill_length_mm ~ C(species)', data=my_penguins).fit()
aov_table = sm.stats.anova_lm(model, typ=2)
print(aov_table)
```

```
##                  sum_sq     df           F        PR(>F)
## C(species)  7194.317439    2.0  410.600255  2.694614e-91
## Residual    2969.888087  339.0         NaN           NaN
```

ANOVA 테이블 결과를 보면, 귀무가설 하에서 검정 통계량인 F 값이 410.6이 나왔고, 대응하는 p-value는 2e-91보다 작은 것을 확인하였다. 따라서 유의수준 1%보다 훨씬 작으므로 귀무가설을 기각한다. 펭귄 종 간 부리길이가 다른 종이 적어도 하나 존재한다.

위의 ANOVA 검정 결과를 신뢰할 수 있는지 ANOVA 모델의 가정을 체크하자.

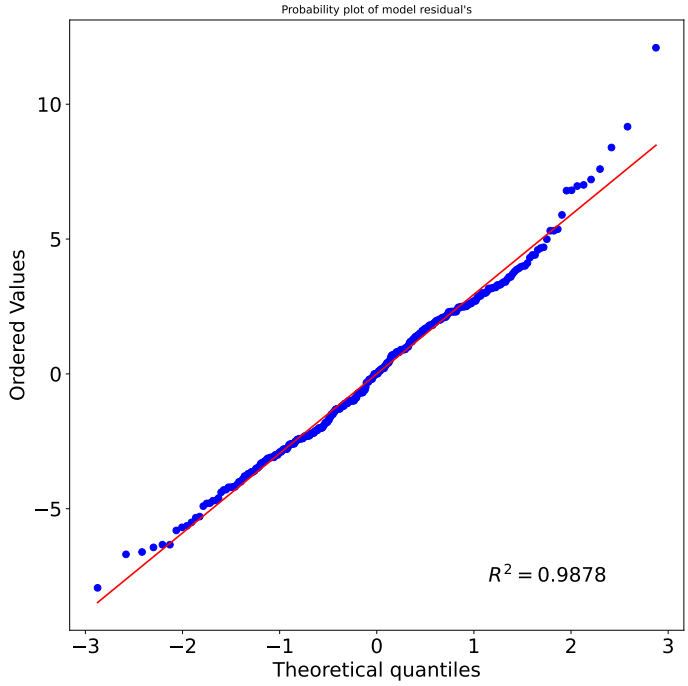
- 잔차의 정규성
- 잔차의 등분산성

```
import matplotlib.pyplot as plt
import scipy.stats as stats

fig = plt.figure(figsize= (10, 10))
ax = fig.add_subplot(111)
```

```
normality_plot, stat = stats.probplot(model.resid, plot= plt, rvalue= True)
ax.set_title("Probability plot of model residual's", fontsize= 10)
```

```
## Text(0.5, 1.0, "Probability plot of model residual's")
```

```
ax.set;

plt.show()
```



그래프로 보아 잔차의 QQ plot이 기준선을 따라 분포하는 것을 확인할 수 있다. 따라서, 정규성을 만족할 것으로 판단된다. 최종 판단을 위해서 샤피로-윌크 검정을 수행한다.

- 귀무가설: 잔차가 정규분포를 따른다.
- 대립가설: 잔차가 정규분포를 따르지 않는다.

```
stats.shapiro(model.resid)
```

```
## ShapiroResult(statistic=0.989031195640564, pvalue=0.011304551735520363)
```

샤피로 윌크 검정의 검정 통계량 값 $W = 0.98903$에 대응하는 p-value값이 0.01131로 주어진 유의수준 0.01보다 크므로 귀무가설을 기각할 수 없다. 따라서 잔차가 정규성을 만족한다고 판단한다.

다음은 잔차의 등분산 검정을 위하여 levene's 검정을 수행하도록 한다.

- 귀무가설: 펭귄 종별 잔차의 분산은 같다.

  - $\sigma^2_{adelie} = \sigma^2_{chinstrap} = \sigma^2_{gentoo}$
- 대립가설: 펭귄 종별 잔차의 분산 중 다른 쌍이 적어도 하나 존재한다.
  - Not all of the σ^2_is are equal

```
stats.levene(my_penguins[my_penguins['species'] == 'Adelie']['bill_length_mm'],
             my_penguins[my_penguins['species'] == 'Chinstrap']['bill_length_mm'],
             my_penguins[my_penguins['species'] == 'Gentoo']['bill_length_mm'],
             center='mean')
```

```
## LeveneResult(statistic=2.833610648953925, pvalue=0.060193797718201124)
```

검정통계량 값 2.833에 대응하는 p-value는 0.06019로 유의수준 1% 하에서 귀무가설을 기각하지 못한다. 따라서, 잔차의 집단간 등분산성 가정 역시도 만족한다고 판단한다.

사후검정

ANOVA의 귀무가설이 기각되었으므로, 사후 분석을 통하여 모평균이 다른 그룹을 판변하기 위한 사후분석을 수행한다. 사후 분석은 `aov()` 함수의 결과값을 입력값으로 받을 수 있는 `TukeyHSD()`을 사용하여 수행하도록 한다.

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

tukey_result = pairwise_tukeyhsd(my_penguins['bill_length_mm'], my_penguins['species'], alpha=0.01)

print(tukey_result)
```

```
##    Multiple Comparison of Means - Tukey HSD, FWER=0.01
## ============================================================
##   group1    group2  meandiff p-adj   lower   upper  reject
## ------------------------------------------------------------
##    Adelie Chinstrap  10.0424   -0.0  8.7745 11.3104   True
##    Adelie    Gentoo   8.7135   -0.0   7.659   9.768   True
## Chinstrap    Gentoo  -1.3289 0.0089 -2.6409  -0.017   True
## ------------------------------------------------------------
```

결과를 확인해보면, 모든 펭귄 종별 부리 길이는 통계적으로 유의미한 차이를 보인다고 판단할 수 있다. (각 사후 검정의 adjusted p-value 값이 0.01보다 작다.)

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_palette(["#00AFBB", "#E7B800", "#FC4E07"])

plt.figure(figsize=(8, 6))
```

```
## <Figure size 800×600 with 0 Axes>
```

```
sns.boxplot(x='species', y='bill_length_mm', data=my_penguins)
```

```
## <AxesSubplot:xlabel='species', ylabel='bill_length_mm'>
```

```
sns.swarmplot(x='species', y='bill_length_mm', data=my_penguins, color='black', size=2)
```

```
## <AxesSubplot:xlabel='species', ylabel='bill_length_mm'>
```

```
sns.pointplot(x='species', y='bill_length_mm', data=my_penguins, color='red', markers='o', errorbar
```

```
## <AxesSubplot:xlabel='species', ylabel='bill_length_mm'>
```

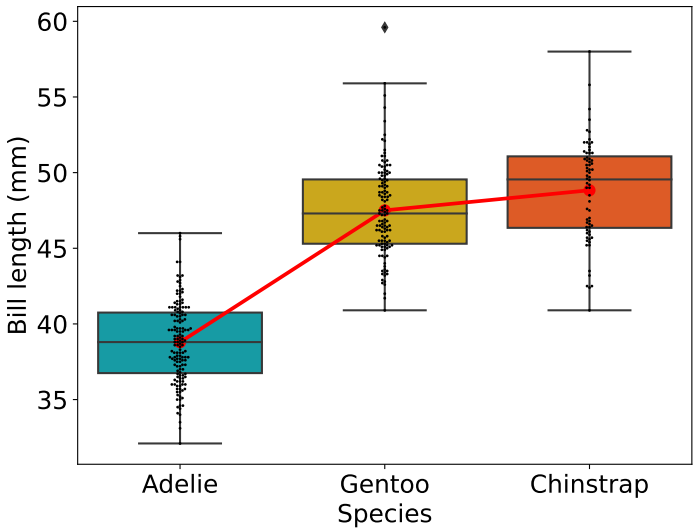
```
plt.xlabel('Species')
```

```
## Text(0.5, 26.722222222222207, 'Species')
```

```
plt.ylabel('Bill length (mm)')
```

```
## Text(52.847222222222214, 0.5, 'Bill length (mm)')
```

```
plt.show()
```



따라서, 유의수준 0.01을 기준으로 각 펭귄 종별 부리 길이는 통계적으로 유의미한 차이를 보인다고 판단하며, Adelie < Gentoo < Chinstrap의 순서대로 평균이 증가한다고 판단한다.

드릴 도구 검정 절차 (Drilling process)

drilling-tool-exam.csv 데이터를 사용하여 다음 물음에 답하세요.

슬통 철공 회사에서는 5개의 브랜드의 드릴 소재를 사용하여 철판에 구멍을 뚫은 작업을 하고 있다. 회사 제품은 2.5cm 직경을 가진 철판인데, 브랜드 별 사용하는 소재들이 미세하게 달라, 직경에 차이가 발생하는지 알아보고자 한다. 품질 팀은 제품 품질을 유지하기 위하여 온도의 영향도 조사하기로 하였다. 슬통이는 한 명의 품질 관리사에게 각기 다른 온도에서 브랜드별 드릴을 무작위로 20개씩 선택하여 뚫은 구멍의 직경을 측정하도록 하였다.

1) 데이터를 사용 각 브랜별, 온도별 평균과 표준편차 정리 표를 만드세요.

데이터 로드 및 전처리 (Wide 형태 데이터 전처리 연습)

```
import pandas as pd

# Load the dataset
data = pd.read_csv("./data/drilling-tool-exam.csv", skiprows=1)
data.head()
```

```
##   Brand  Temp  Mesurement 1  ...  Mesurement 18  Mesurement 19  Mesurement 20
## 0     A   100     25.031768  ...       25.029239      25.029938      25.032016
## 1     A   200     25.027100  ...       25.028416      25.028347      25.028968
## 2     A   300     25.024713  ...       25.023498      25.026318      25.025532
## 3     B   100     25.018161  ...       25.017452      25.018334      25.016180
## 4     B   200     25.020642  ...       25.020781      25.018409      25.021757
##
## [5 rows x 22 columns]
```

데이터를 불러오면 브랜드별 온도별로 총 20개의 직경 데이터가 존재한다. 각 그룹별 평균과 표준편차값을 계산하면 다음과 같다.

```
drill_long = pd.melt(
    data, id_vars=['Brand', 'Temp'],
    value_vars=[col for col in data.columns if 'Mesurement' in col],
    var_name='Mesurement',
    value_name='mm')

drill_long.head()
```

```
##   Brand  Temp    Mesurement         mm
## 0     A   100  Mesurement 1  25.031768
## 1     A   200  Mesurement 1  25.027100
## 2     A   300  Mesurement 1  25.024713
```

```
## 3     B   100  Mesurement 1  25.018161
## 4     B   200  Mesurement 1  25.020642
```

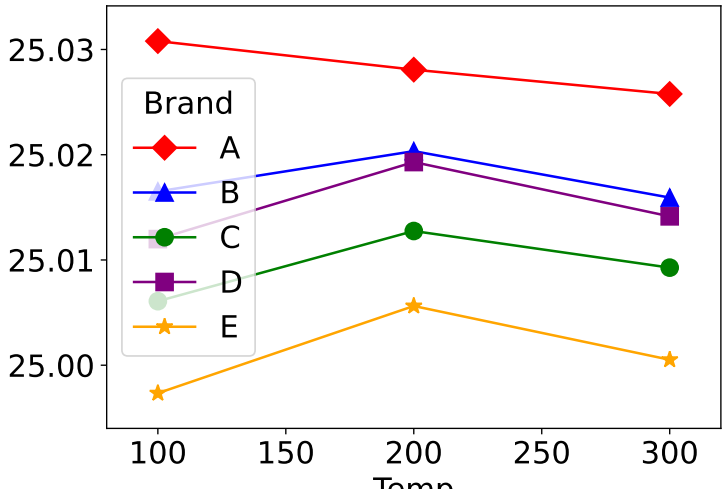
- 브랜드별 온도별 평균, 표준편차표

```
drill_long['mm'] = pd.to_numeric(drill_long['mm'])

# Calculate mean and standard deviation grouped by Tool and Temp
drilling_summary = drill_long.groupby(['Brand', 'Temp'])
drilling_summary.agg(mean_mm=('mm', 'mean'), sd_mm=('mm', 'std')).reset_index()
```

```
##    Brand  Temp    mean_mm     sd_mm
## 0      A   100  25.030796  0.001195
## 1      A   200  25.028082  0.000762
## 2      A   300  25.025773  0.001013
## 3      B   100  25.016513  0.001040
## 4      B   200  25.020340  0.000814
## 5      B   300  25.015927  0.000888
## 6      C   100  25.006082  0.001162
## 7      C   200  25.012751  0.001289
## 8      C   300  25.009271  0.000939
## 9      D   100  25.011982  0.000761
## 10     D   200  25.019303  0.000846
## 11     D   300  25.014145  0.001145
## 12     E   100  24.997342  0.001214
## 13     E   200  25.005625  0.001305
## 14     E   300  25.000520  0.001012
```

2) 브랜드별 온도별로 ANOVA main effect & interaction plot을 그려보세요.

```
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.graphics.factorplots import interaction_plot

fig = interaction_plot(
    drill_long['Temp'], drill_long['Brand'], drill_long['mm'],
    colors=['red', 'blue', 'green', 'purple', 'orange'],
    markers=['D', '^', 'o', 's', '*'], ms=10)
plt.show()
```



3) Two-way ANOVA 분석을 진행해 주세요.

Two-way ANOVA의 경우 3개의 귀무가설이 존재한다.

- 귀무가설1: Brand 변수의 main effect가 존재하지 않는다.
- 대립가설1: Brand 변수의 main effect가 존재한다.
- 귀무가설2: Temp 변수의 main effect가 존재하지 않는다.
- 대립가설2: Temp 변수의 main effect가 존재한다.
- 귀무가설3: Brand 변수와 Temp 변수의 interaction이 존재하지 않는다.
- 대립가설3: Brand 변수와 Temp 변수의 interaction이 존재한다.

```
# Conduct two-way ANOVA
formula = 'mm ~ C(Brand) + C(Temp) + C(Brand):C(Temp)'
model = ols(formula, data=drill_long).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
anova_table
```

```
##                     sum_sq     df            F         PR(>F)
## C(Brand)          0.024130    4.0  5561.567051  4.952237e-269
## C(Temp)           0.001299    2.0   598.805667  8.790226e-103
## C(Brand):C(Temp)  0.000893    8.0   102.903992   1.874030e-79
## Residual          0.000309  285.0          NaN            NaN
```

ANOVA 테이블을 살펴보면, 각 변수의 main effect에 대한 검정 통계량값인 F값이 5561.6과 598.8이 계산되었고, 대응하는 p-value값이 모두 2e-16보다 낮아서 유의수준 5%하에서 귀무가설 1과 2가 기각된다. 따라서, 두 변수의 main effect가 존재한다고 판단한다.

두 변수의 교호작용을 검정하는 F 값의 경우 역시 102.9가 나왔으며, 대응하는 유의수준 역시 2e-16보다 낮아서 유의수준 5%하에서 귀무가설이 기각된다.

따라서, 브랜드별, 온도별 구멍 평균 직경은 다르다고 판단하고, 두 변수의 교호작용에 영향을 받는다고 판단한다. 앞에서 작성한 그래프를 참고하면, 브랜드 별 구멍 직경의 차이가 온도가 낮은 경우에 온도가 중간인 경우에 비하여 좀 더 많이 차이가 나는 것을 확인할 수 있다.

4) 각 메인 effect가 통계적으로 유의한 경우 사후 분석을 진행해주세요.

브랜드와 온도 변수에 대하여 각 그룹의 평균 구멍 직경이 다르다고 나왔으므로, 각 변수에 대하여 사후 분석을 진행한다. 먼저 Brand 변수에 대하여 사후분석을 진행하기 위해 `TukeyHSD()` 함수를 사용한다.

```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

posthoc_brand = pairwise_tukeyhsd(drill_long['mm'], drill_long['Brand'])
print(posthoc_brand.summary())
```

```
## Multiple Comparison of Means - Tukey HSD, FWER=0.05
## ====================================================
## group1 group2 meandiff p-adj   lower   upper  reject
## ----------------------------------------------------
##      A      B  -0.0106   -0.0 -0.0121 -0.0092   True
##      A      C  -0.0188   -0.0 -0.0203 -0.0174   True
##      A      D  -0.0131   -0.0 -0.0145 -0.0116   True
##      A      E  -0.0271   -0.0 -0.0285 -0.0256   True
##      B      C  -0.0082   -0.0 -0.0097 -0.0068   True
##      B      D  -0.0024 0.0001 -0.0039  -0.001   True
##      B      E  -0.0164   -0.0 -0.0179  -0.015   True
##      C      D   0.0058   -0.0  0.0043  0.0072   True
##      C      E  -0.0082   -0.0 -0.0097 -0.0067   True
##      D      E   -0.014   -0.0 -0.0154 -0.0125   True
## ----------------------------------------------------
```

결과를 살펴보면 브랜드 변수의 모든 레벨에 대하여 사후 분석을 진행한 경과, p-value값이 모두 다 낮게 나오고, 따라서 통계적으로 유의한 차이를 보인다고 판단할 수 있다.

```
posthoc_temp = pairwise_tukeyhsd(drill_long['mm'], drill_long['Temp'])
print(posthoc_temp.summary())
```

```
## Multiple Comparison of Means - Tukey HSD, FWER=0.05
## ====================================================
## group1 group2 meandiff p-adj   lower  upper  reject
## ----------------------------------------------------
##    100    200   0.0047 0.0012  0.0016 0.0078   True
##    100    300   0.0006 0.8956 -0.0025 0.0037  False
##    200    300  -0.0041 0.0054 -0.0072 -0.001   True
```

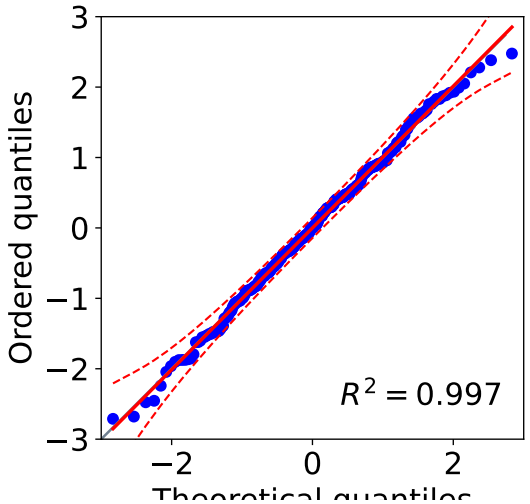
```
## ------------------------------------------------------
```

온도 변수 역시 Low, Medium, High 그룹 간 평균 차이가 통계적으로 유의한 차이를 보인다는 것을 확인할 수 있다. 또한 계산된 diff 값에 비추어보면 Low < High < Medium 순으로 평균 구멍 직경이 커져간다고 판단 할 수 있다.

가정체크

Two way ANONA의 결과를 신뢰하기 위해서는 ANOVA에서 가정하고 있는 잔차의 정규성과 등분산성에 대한 검정이 필요하다.

```
import scipy.stats as stats
import pingouin as pg

# QQ plot
ax = pg.qqplot(model.resid, dist='norm')
plt.ylim(-3, 3);
plt.xlim(-3, 3);
plt.show()
```



잔차의 정규성

잔차 그래프를 살펴보면 잔차들이 0을 중심으로 퍼져있고, QQ plot 역시 정규성을 만족하는 것으로 판단된다. 정확한 검정을 위하여 샤피로 윌크 검정을 사용하여 정규성 체크를 하도록 하자.

- 귀무가설: 잔차가 정규성을 따른다.
- 대립가설: 잔차가 정규성을 따르지 않는다.

```
stats.shapiro(model.resid)
```

```
## ShapiroResult(statistic=0.9961785078048706, pvalue=0.6853092312812805)
```

검정 통계량 값 0.996에 대응하는 p-value값이 0.6852이므로 유의수준 5%하에서 귀무가설을 기각하지 못한다. 따라서 정규성을 만족한다고 판단한다.

잔차의 등분산성

- 귀무가설: 잔차들의 그룹별 모분산이 동일하다.
- 대립가설: 잔차들의 그룹별 모분산 중 다른 것이 적어도 하나 존재한다.

```
# 인터렉션 그룹 만들기
drill_long['Brand_Temp'] = drill_long['Brand'] + "_" + drill_long['Temp'].astype(str)

group_vec = drill_long['Brand_Temp']
levene_test = stats.levene(
    *[model.resid[group_vec == i] for i in group_vec.unique()], center='mean')

levene_test
```

```
## LeveneResult(statistic=1.3624630069062795, pvalue=0.17083426897357204)
```

검정통계량 값 1.3625에 대응하는 p-value 값 0.1708이 유의수준 5%보다 크므로 귀무 가설을 기각할 수 없다. 따라서, 등분산 가정도 만족한다고 판단한다.

결과적으로 앞에서 수행한 two way ANOVA 의 결과를 신뢰할 수 있다.

Chapter 7. 회귀분석의 이해

펭귄 부리길이와 깊이의 관계

palmerpenguins 패키지에는 남극 Palmer station에서 관측한 펭귄 정보들이 포함된 데이터이다.

```
import pandas as pd
import numpy as np
from palmerpenguins import load_penguins

penguins = load_penguins()
print(penguins.head())
```

```
##   species     island  bill_length_mm  ...  body_mass_g     sex  year
## 0  Adelie  Torgersen            39.1  ...       3750.0    male  2007
## 1  Adelie  Torgersen            39.5  ...       3800.0  female  2007
## 2  Adelie  Torgersen            40.3  ...       3250.0  female  2007
## 3  Adelie  Torgersen             NaN  ...          NaN     NaN  2007
## 4  Adelie  Torgersen            36.7  ...       3450.0  female  2007
##
## [5 rows x 8 columns]
```

1) train_index 를 사용하여 펭귄 데이터에서 인덱스에 대응하는 표본들을 뽑아서 train_data를 만드세요. (단, 결측치가 있는 경우 제거)

```
np.random.seed(2022)
train_index=np.random.choice(penguins.shape[0],200)

train_data = penguins.iloc[train_index]
train_data = train_data.dropna()
train_data.head()
```

```
##     species  island  bill_length_mm  ...  body_mass_g     sex  year
## 220  Gentoo  Biscoe            43.5  ...       4700.0  female  2008
## 173  Gentoo  Biscoe            45.1  ...       5000.0  female  2007
## 112  Adelie  Biscoe            39.7  ...       3200.0  female  2009
## 177  Gentoo  Biscoe            46.1  ...       5100.0    male  2007
## 240  Gentoo  Biscoe            47.5  ...       4875.0  female  2009
##
## [5 rows x 8 columns]
```

2) train_data의 펭귄 부리길이 (bill_length_mm)를 부리 깊이 (bill_depth_mm)를 사용하여 산점도를 그려보세요.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Scatter plot using seaborn
plt.figure(figsize=(10,6))
```

```
## <Figure size 1000×600 with 0 Axes>
```

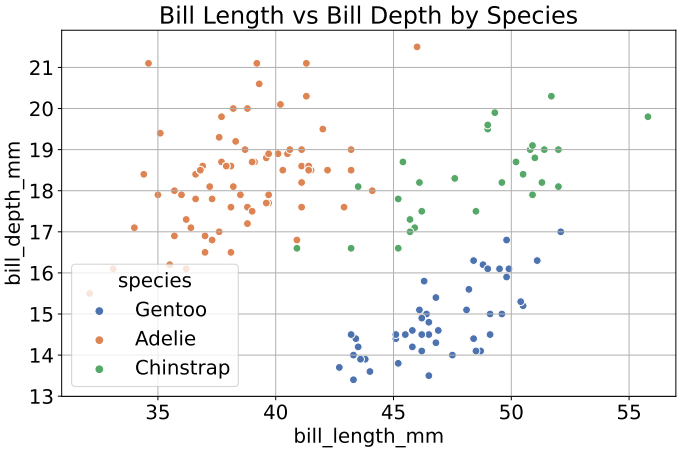
```
sns.scatterplot(data=train_data,
                x='bill_length_mm',
                y='bill_depth_mm',
                hue='species',
                palette='deep', edgecolor='w', s=50)
```

```
## <AxesSubplot:xlabel='bill_length_mm', ylabel='bill_depth_mm'>
```

```
plt.title('Bill Length vs Bill Depth by Species')
```

```
## Text(0.5, 1.0, 'Bill Length vs Bill Depth by Species')
```

```
plt.grid(True)
plt.show()
```



3) 펭귄 부리길이 (bill_length_mm)를 부리 깊이 (bill_depth_mm)의 상관계수를 구하고, 두 변수 사이에 유의미한 상관성이 존재하는지 검정해보세요.

- 귀무가설 H_0: 두 변수의 상관계수 $\rho = 0$
- 대립가설 H_A: 두 변수의 상관계수 $\rho \neq 0$

두 변수의 상관계수 검정은 `pearsonr()` 함수를 사용하여 수행할 수 있다.

```
from scipy.stats import pearsonr

# Calculate the Pearson correlation coefficient
# and the p-value for testing non-correlation
corr_coef, p_value = pearsonr(train_data['bill_length_mm'], train_data['bill_depth_mm'])
print(corr_coef)
```

```
## -0.24938519717051547
```

```
print(p_value)
```

```
## 0.00040929638362032476
```

유의수준 5%하에서 상관계수 값 -0.2493에 해당하는 p-value 값 0.000409296이 상당히 작으므로, 귀무가설을 기각한다. 따라서, 두 변수의 상관계수가 0이 아니라는 통계적 근거가 충분하다.

위의 검정 결과를 신뢰 할 수 있는지 확인하기 위하여, 상관관계 분석에서 가정하고 있는 정규성을 체크해보도록 하자.

```
import statsmodels.api as sm
import matplotlib.pyplot as plt
import pingouin as pg
```

```
# Set up the subplots: 1 row, 2 columns
fig, axs = plt.subplots(1, 2, figsize=(12, 6))

# Q-Q plot for bill_depth_mm on the left
pg.qqplot(train_data['bill_depth_mm'], dist='norm', confidence=0.95, ax=axs[0])
```

```
## <AxesSubplot:xlabel='Theoretical quantiles', ylabel='Ordered quantiles'>
```

```
axs[0].set_title("bill_depth_mm")
```

```
## Text(0.5, 1.0, 'bill_depth_mm')
```

```
axs[0].set_ylim(-3, 3);
axs[0].set_xlim(-3, 3);

# Q-Q plot for bill_length_mm on the right
```

```
## (-3.0, 3.0)
```

```
pg.qqplot(train_data['bill_length_mm'], dist='norm', confidence=0.95, ax=axs[1])
```

```
## <AxesSubplot:xlabel='Theoretical quantiles', ylabel='Ordered quantiles'>
```

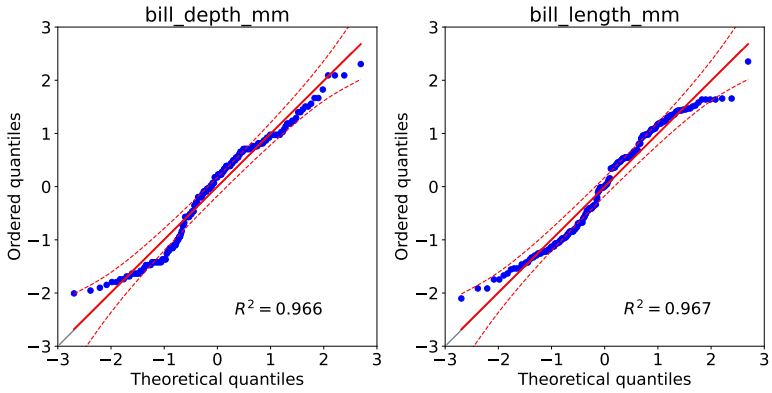
```
axs[1].set_title("bill_length_mm")
```

```
## Text(0.5, 1.0, 'bill_length_mm')
```

```
axs[1].set_ylim(-3, 3);
axs[1].set_xlim(-3, 3);

# Display the plots
```

```
## (-3.0, 3.0)
```

```
plt.tight_layout()
plt.show()
```



각 변수에 해당하는 QQ plot을 그려 보았을 때, 부리 길이 (bill length) 변수와 부리 깊이 (bill depth) 변수에 해당하는 그래프는 정규성을 띄지 않는 것을 알 수 있다. Shapiro-Wilk 검정 결과 역시 두 그룹 표본이 정규성 가정을 위반한 다는 것을 말해주고 있다.

Shapiro-Wilk 검정

- 귀무가설 H_0: 데이터의 분포가 정규성을 띈다.
- 대립가설 H_A: 데이터의 분포가 정규성을 띄지 못한다.

```
import scipy.stats as sp

sp.shapiro(train_data['bill_depth_mm'])
```

```
## ShapiroResult(statistic=0.9621433019638062, pvalue=3.906726124114357e-05)
```

```
sp.shapiro(train_data['bill_length_mm'])
```

```
## ShapiroResult(statistic=0.9634734988212585, pvalue=5.4907886806176975e-05)
```

즉, 유의수준 5%하에서 두 변수 모두 귀무가설을 기각하게 되므로, 위의 상관계수 검정 결과는 신뢰할 수 없다.

4) 펭귄 부리길이 (bill_length_mm)를 부리 깊이 (bill_depth_mm)를 사용하여 설명하는 회귀 모델을 적합시킨 후 2번의 산점도에 회귀 직선을 나타내 보세요. (모델 1)

```
from statsmodels.formula.api import ols

model1 = ols("bill_length_mm ~ bill_depth_mm", data=train_data).fit()
model1.params
```

```
## Intercept        55.410976
## bill_depth_mm    -0.706191
## dtype: float64
```

```
sns.scatterplot(data=train_data,
                x='bill_depth_mm', y='bill_length_mm',
                palette='deep', edgecolor='w', s=50)

# Use the slope and intercept to plot the regression line
```

```
## <AxesSubplot:xlabel='bill_depth_mm', ylabel='bill_length_mm'>
```

```
x_values = train_data['bill_depth_mm']
y_values = 55.4110 - 0.7062 * x_values
plt.plot(x_values, y_values, color='red', label='Regression Line')
```

```
## [<matplotlib.lines.Line2D object at 0×000001ABA622F708>]
```

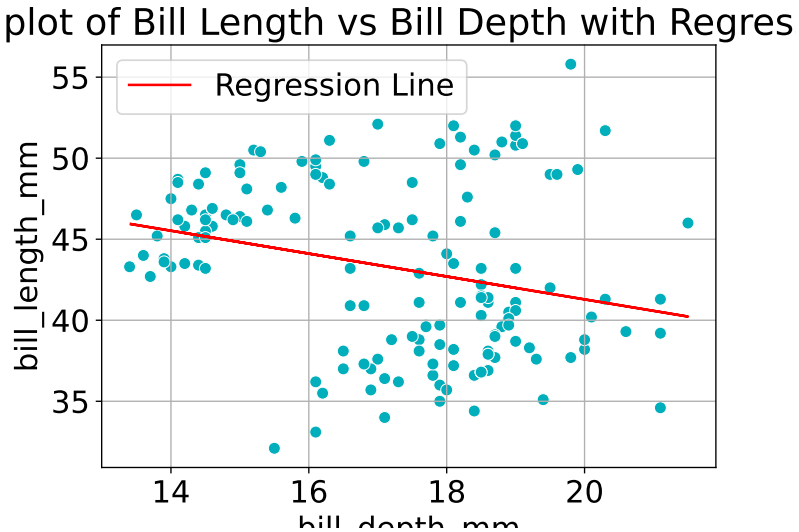
```
plt.title('Scatter plot of Bill Length vs Bill Depth with Regression Line')
```

```
## Text(0.5, 1.0, 'Scatter plot of Bill Length vs Bill Depth with Regression Line')
```

```
plt.grid(True)
plt.legend()
```

```
## <matplotlib.legend.Legend object at 0×000001ABA77E6F88>
```

```
plt.show()
```



5) 적합된 회귀 모델이 통계적으로 유의한지 판단해보세요.

부리 깊이 변수를 사용하여 부리 길이 변수를 설명하는 회귀 모델을 설정한다.

```
model1.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:         bill_length_mm   R-squared:                       0.062
## Model:                            OLS   Adj. R-squared:                  0.057
## Method:                 Least Squares   F-statistic:                     12.93
## Date:                  토, 28 10 2023   Prob (F-statistic):           0.000409
## Time:                        14:03:42   Log-Likelihood:                -602.49
## No. Observations:                 197   AIC:                             1209.
## Df Residuals:                     195   BIC:                             1216.
## Df Model:                           1
## Covariance Type:            nonrobust
## ================================================================================
##                    coef    std err          t      P>|t|      [0.025      0.975]
## --------------------------------------------------------------------------------
## Intercept        55.4110      3.392     16.336      0.000      48.721      62.101
## bill_depth_mm    -0.7062      0.196     -3.596      0.000      -1.093      -0.319
## ==============================================================================
## Omnibus:                        4.987   Durbin-Watson:                   1.786
## Prob(Omnibus):                  0.083   Jarque-Bera (JB):                3.609
## Skew:                           0.193   Prob(JB):                        0.165
## Kurtosis:                       2.461   Cond. No.                         159.
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
## """
```

유의수준 5%하에서 F 검정 통계량 값 12.93에 대응하는 p-value값 0.000409에 비추어 보았을 때, 회귀 모델은 통계적으로 유의한 것으로 판단한다.

6) R^2 값을 구한 후 의미를 해석해 보세요.

위의 결과값에서 R square 값은 0.062이며, 이는 회귀모델이 데이터 전체 변동성의 6.21%를 설명하고 있다는 의미이다. 상대적으로 낮은 R square 값을 미루어 보았을때, 추가 변수를 사용하여 모델의 설명력을 높이는 것을 고려할 수 있다.

7) 적합된 회귀 모델의 계수를 해석해 보세요.

- 절편과 기울기가 모두 유의하게 나오지만, 각 변수의 뜻을 고려하면, 절편의 해석은 무의미하다. (부리 깊이 0인 경우, 부리 길이 56 mm)

- 기울기 -0.7062 값의 의미는, 팔머 펭귄의 경우 부리 깊이가 1 mm 증가할 때, 부리 길이는 0.7062 mm 만큼 감소하는 경향을 보인다고 해석할 수 있다.

8) 1번에서 적합한 회귀 모델에 새로운 변수 (종 - species) 변수를 추가하려고 합니다. 성별 변수 정보를 사용하여 점 색깔을 다르게 시각화 한 후 적합된 모델의 회귀 직선을 시각화 해보세요. (모델 2)

```
model2 = ols("bill_length_mm ~ bill_depth_mm + species", data=train_data).fit()
model2.params

# Set up the plot
```

```
## Intercept               14.577564
## species[T.Chinstrap]     9.886486
## species[T.Gentoo]       12.912783
## bill_depth_mm            1.320354
## dtype: float64
```

```
sns.scatterplot(data=train_data,
                x='bill_depth_mm', y='bill_length_mm',
                hue='species', palette='deep', edgecolor='w', s=50)

# Generate regression lines for each species
```

```
## <AxesSubplot:xlabel='bill_depth_mm', ylabel='bill_length_mm'>
```

```
for species in train_data['species'].unique():
    # Filter data by species
    subset = train_data[train_data['species'] == species]

    # Predict values using the regression model
    x_vals = subset['bill_depth_mm'].sort_values()
    y_vals = model2.predict(pd.DataFrame({'bill_depth_mm': x_vals, 'species': species}))

    # Plot regression line for this species
    sns.lineplot(x=x_vals, y=y_vals, label=f'Regression Line ({species})')
```

```
## <AxesSubplot:xlabel='bill_depth_mm', ylabel='bill_length_mm'>
## <AxesSubplot:xlabel='bill_depth_mm', ylabel='bill_length_mm'>
## <AxesSubplot:xlabel='bill_depth_mm', ylabel='bill_length_mm'>
```

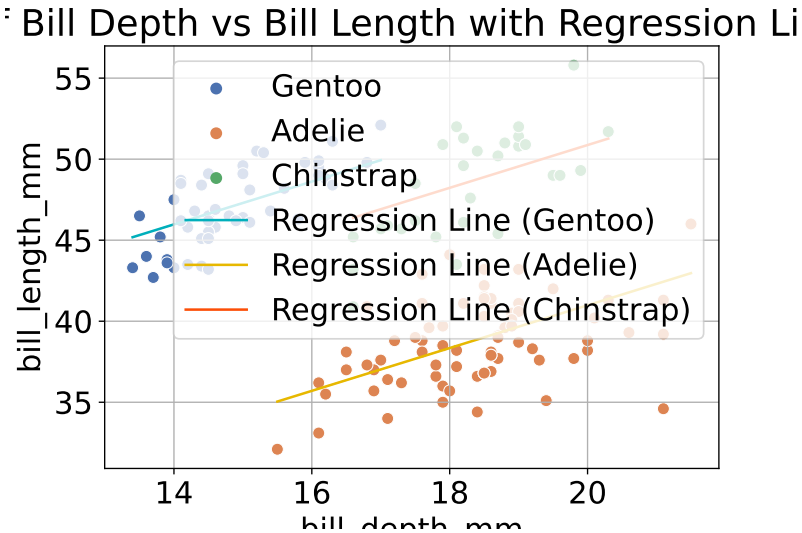
```
plt.title('Scatter plot of Bill Depth vs Bill Length with Regression Lines by Species')
```

```
## Text(0.5, 1.0, 'Scatter plot of Bill Depth vs Bill Length with Regression Lines ↩
by Species')
```

```
plt.grid(True)
plt.legend()
```

```
## <matplotlib.legend.Legend object at 0×000001AB97FCD488>
```

```
plt.show()
```



결과를 살펴보면, 종 (species) 변수를 추가하므로써 회귀직선의 방향이 바뀌었다. 또한, 전체적인 데이터의 추세를 회귀 직선이 더 잘 설명하고 있는 것을 볼 수 있다. 이러한 현상을 심슨's 패러독스라고 부른다.

9) 종 변수가 새로 추가된 모델 2가 모델 1 보다 더 좋은 모델이라는 근거를 제시하세요.

```
table = sm.stats.anova_lm(model1, model2) #anova
print(table)
```

```
##    df_resid          ssr  df_diff     ss_diff           F        Pr(>F)
## 0     195.0  5228.862451      0.0         NaN         NaN           NaN
## 1     193.0  1044.226281      2.0  4184.63617  386.714449  3.071739e-68
```

모델 2는 모델 1을 포함하는 Full 모델과 Reduced 모델로 볼 수 있다. 두 모델 사이의 통계적 유의성을 F 검정을 통하여 보일 수 있다.

- 귀무가설 H_0: Reduced 모델이 알맞음.
- 대립가설 H_A: Full 모델이 알맞음.

위의 ANOVA 코드의 결과에서 F 검정 통계량 386.71에 대응하는 p-value값 3.07e-68이 유의수준 5% 하에서 귀무가설을 기각할 수 있으므로, 모델 1 보다 모델 2가 더 알맞은 모델이라 판단한다.

10) 모델 2의 계수에 대한 검정과 그 의미를 해석해 보세요.

```
model2.summary()
```

```
## <class 'statsmodels.iolib.summary.Summary'>
## """
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:         bill_length_mm   R-squared:                       0.813
## Model:                            OLS   Adj. R-squared:                  0.810
## Method:                 Least Squares   F-statistic:                     279.2
## Date:                  토, 28 10 2023   Prob (F-statistic):           6.28e-70
## Time:                        14:03:46   Log-Likelihood:                -443.81
## No. Observations:                 197   AIC:                             895.6
## Df Residuals:                     193   BIC:                             908.8
## Df Model:                           3
## Covariance Type:            nonrobust
## ↩
===================================================================================================
##                           coef    std err          t      P>|t|      [0.025      0.975]
## ↩
---------------------------------------------------------------------------------------------------
## Intercept              14.5776      2.855      5.107      0.000       8.947      20.208
## species[T.Chinstrap]    9.8865      0.446     22.159      0.000       9.007      10.766
## species[T.Gentoo]      12.9128      0.638     20.243      0.000      11.655      14.171
## bill_depth_mm           1.3204      0.156      8.448      0.000       1.012       1.629
## ==============================================================================
## Omnibus:                        0.570   Durbin-Watson:                   2.084
## Prob(Omnibus):                  0.752   Jarque-Bera (JB):                0.322
## Skew:                          -0.074   Prob(JB):                        0.851
## Kurtosis:                       3.132   Cond. No.                         304.
## ==============================================================================
##
## Notes:
## [1] Standard Errors assume that the covariance matrix of the errors is correctly ↩
specified.
## """
```

회귀분석의 결과 모형이 통계적으로 유의미하고, 모든 계수 역시 통계적으로 유의한 것을 확인할 수 있다.

- 부리 깊이에 대응하는 계수는 1.3204로, 의미는 부리깊이 1mm가 증가하면, 부리길이가 1.3204mm가 증가하는 경향성을 보인다고 해석할 수 있다.
- 기준 레벨이 되는 아델리 펭귄 종의 부리깊이에 따른 부리길이 추세는 $14.5776 + 1.3204 \times bill_depth$를 따른다.
- `species[T.Chinstrap]` 변수의 계수 9.8865, `species[T.Gentoo]` 변수의 계수 12.9128에서 친스트랩 펭귄 종은 아델리 펭귄 종보다 평균적으로 약 9.88mm, 겐투 펭귄 종은 아델리 펭귄 종보다 평균적으로 약 12.91mm 가 긴 경향성을 보인다고 해석할 수 있다.

11) 모델 2 에 잔차 그래프를 그리고, 회귀모델 가정을 만족하는지 검증을 수행해주세요.

```
import scipy.stats as stats

# Set up the subplots: 1 row, 2 columns
residuals = model2.resid
fitted_values = model2.fittedvalues

plt.figure(figsize=(16,8))
```

```
## <Figure size 1600×800 with 0 Axes>
```

```
plt.subplot(1,2,1)
```

```
## <AxesSubplot:>
```

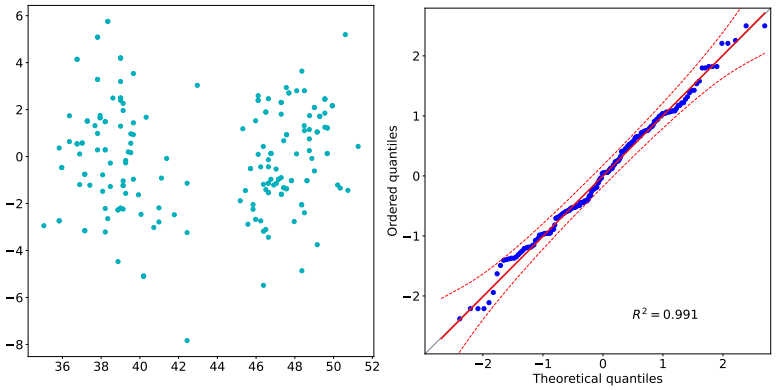
```
plt.scatter(fitted_values, residuals);

plt.subplot(1,2,2)
```

```
## <AxesSubplot:>
```

```
pg.qqplot(residuals, dist='norm', confidence=0.95);

# Display the plots
```

```
## <AxesSubplot:xlabel='Theoretical quantiles', ylabel='Ordered quantiles'>
```

```
plt.tight_layout()
plt.show()
```



잔차 그래프와 잔차의 QQ plot 그래프로 보아 잔차의 등분산성과 정규성을 만족하는 것으로 판단된다.

```
from scipy.stats import shapiro
from statsmodels.stats.diagnostic import het_breuschpagan
from statsmodels.stats.stattools import durbin_watson

shapiro(residuals)
```

```
## ShapiroResult(statistic=0.9919706583023071, pvalue=0.35017824172973633)
```

```
durbin_watson(residuals)
```

```
## 2.083836063916197
```

```
labels = ['Lagrange multiplier statistic', 'p-value', 'f-value', 'f p-value']
bp_test = het_breuschpagan(residuals, model2.model.exog)
bp_result = dict(zip(labels, bp_test))
bp_result
```

```
## {'Lagrange multiplier statistic': 12.575027525429496, 'p-value': ↩
0.005651859166962036, 'f-value': 4.3865720927834015, 'f p-value': 0.005177711883822766}
```

유의수준 5%하에서 잔차 정규성을 검정하는 Shapiro-Wilk 검정, 잔차의 등분산성을 검정하는 Breusch–Pagan 검정, 잔차의 독립성을 검정하는 Durbin-Watson 검정의 결과가 회귀모델의 가정을 만족하는 것을 확인할 수 있다.

Durbin-Watson 통계량의 값은 0에서 4 사이에 있으며, 2 근처의 값은 잔차 간에 상관관계가 없음을 나타냅니다. 값이 2보다 크면 음의 상관관계, 2보다 작으면 양의 상관관계가 있을 가능성이 있습니다.

12) 모델 2 의 잔차를 통하여 영향점, 혹은 이상치의 유무를 판단해보세요.

```
influence = model2.get_influence()
stud_res = influence.resid_studentized_external

# 2. Identify observations with studentized residuals
# greater than 3 in absolute value
outliers = np.where(np.abs(stud_res) > 3)[0]

# 3. Retrieve these rows from train_data
outlier_data = train_data.iloc[outliers]
outlier_data["bill_depth_mm"]
```

```
## 14    21.1
## Name: bill_depth_mm, dtype: float64
```

```
outlier_data["bill_length_mm"]
```

```
## 14    34.6
## Name: bill_length_mm, dtype: float64
```

```
print(outlier_data)
```

```
##    species     island  bill_length_mm  ...  body_mass_g   sex  year
## 14  Adelie  Torgersen            34.6  ...       4400.0  male  2007
##
## [1 rows x 8 columns]
```

스튜던트화 잔차를 기준으로 3 표준편차 밖으로 벗어나 있는 표본은 위와 같다. Adelie 표본의 경우 비슷한 부리 깊이의 표본들과는 너무 많은 차이를 보이므로 이상치로 판단한다.