데이터 분석을 위한 통계 기초

강사 소개

유재명

- 서울대학교 산업공학 학사
- 서울대학교 인지과학 박사
- 국민대학교 겸임교수
- (주)퀀트랩 대표

강의 내용

- 학습 목표: 통계의 기초 개념을 이해하고, 통계 프로그램을 활용하여 실제로 분석할 수 있다
- 통계의 기초 개념: 기술 통계, 상관 분석, 회귀 분석

통계학을 배우는 이유

- 데이터 = 패턴 + 노이즈
  - 측정의 불완전성
  - 데이터에 포함된 잡음
  - 제3의 변수
- 통계학의 역할: 위의 다양한 요소를 고려하여 합리적 결론을 유도
- 통계학과 머신러닝의 차이
  - 근본적인 차이는 없음
  - 머신러닝 = 인공지능 + 통계학
  - 통계학은 모형의 타당성에, 머신러닝은 예측 성능에 좀 더 관심을 기울이는 경향이 있음

사례와 변수

- 사례 *case*
  - 데이터 수집의 단위(예: 제품, 실험, 고객 등)
  - 데이터를 표로 나타낼는 한 행 *row*에 표시
- 변수 *variable*
  - 사례에 따라 달라지는 특성(예: 만족도, 성능, 색상)
  - 데이터를 표로 나타낼는 한 열 *column*에 표시

모집단과 표본

- 모집단 *population*: 연구의 관심이 되는 집단 전체
- 표본 *sample*: 특정 연구에서 선택된 모집단의 부분 집합
  - 주의: 표본은 특정 사례 하나가 아닌 사례들의 집합을 의미
- 기술 통계 *descriptive statistics*: 표본을 요약하고 묘사
- 추론 통계 *inferential statistics*: 표본을 통해 모집단에 대해 추측

범주형 변수 *categorical variable*

- 2개 이상의 범주 *category*를 값으로 가지는 변수
- 순서가 없는 범주: 국적, 차종
- 순서가 있는 범주: 학교(초등학교 〈 중학교 〈 고등학교 〈 대학교)

연속형 변수 *continuous variable*

- 실수 *real number*로 표현할수 있는 변수
- 값들의 간격이 일정
- 덧셈, 뺄셈 등의 계산이 의미가 있음
- 예: 무게, 나이, 시간, 거리

기술 통계

기술 통계

- **중심 경향치**: 데이터가 어디에 몰려있는가?
  - 평균, 중간값, 최빈값
- **분위수**: 데이터에서 각각의 순위가 어느 정도인가?
- **변산성 측정치**: 데이터가 어떻게 퍼져있는가?
  - 범위, IQR, 분산, 표준편차

평균 *mean*

- N개의 값이 있을 때, 그 합계를 N으로 나눈 것

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

평균과 극단값

- 평균은 극단값에 따라 영향을 크게 받음
  - 10, 20, 30, 40, 50이 있을 경우 → 평균 30
  - 10, 20, 30, 40, 500이 있을 경우 → 평균 120
- 1986년 미국 노스캐롤라이나 대학 (UNC) 대졸 초봉의 사례:
  - 졸업생 평균 초봉이 가장 높은 학과는 지리학과 (25만 달러)
  - 당시 미국 대졸 평균 초봉은 2만 2천달러 수준
  - 당시 마이클 조던이 UNC 지리학과를 졸업

중간값 *median*

- 값들을 크기 순으로 정렬했을 때 중간에 위치한 값
  - 10, 20, 30, 40, 50의 중간값 → 30
  - 10, 20, 30, 40, 500의 중간값 → 30
- "중위수"라는 표현도 많이 사용 (중위소득, 중위가격 등)
- 값이 짝수개 있을 경우는 가운데 두 값의 평균
  - 10, 20, 30, 40의 중간값 → 20과 30의 평균 = 25

최빈값 *mode*

- 가장 많은 사례에서 관찰된 값
  - 영어 mode에는 상태, 유행, 가장 많은 것 등의 뜻이 있음
- 연속 변수보다는 범주형 변수에서 유용
  - 예: 직원 중에 김씨가 30%가 가장 많음
- 연속 변수의 경우 구간을 나누어 최빈값을 구하는 경우가 많음
  - 예: 고객 중에 30대가 25%로 가장 많음
  - 구간을 나누는 방법에 따라 최빈값이 달라질 수 있음

분위수 *quantile*

- 데이터에서 값의 순위 0 ~ 1를 표현
  - 100을 곱하여 퍼센타일 *percentile*로 표현하기도 함
- 최소값 = 0.0 = 0퍼센타일
- 중간값 = 0.5 = 50퍼센타일
- 최대값 = 1.0 = 100퍼센타일

사분위수 *quartile*

- 데이터를 4등분하는 위치
  - 제1사분위수 = 1/4 지점 = 25퍼센타일
  - 제2사분위수 = 2/4 지점 = 50퍼센타일 = 중간값
  - 제3사분위수 = 3/4 지점 = 75퍼센타일
- 영어 철자에 주의. 분위수(quantile), 사분위수(quartile)

범위 *range*

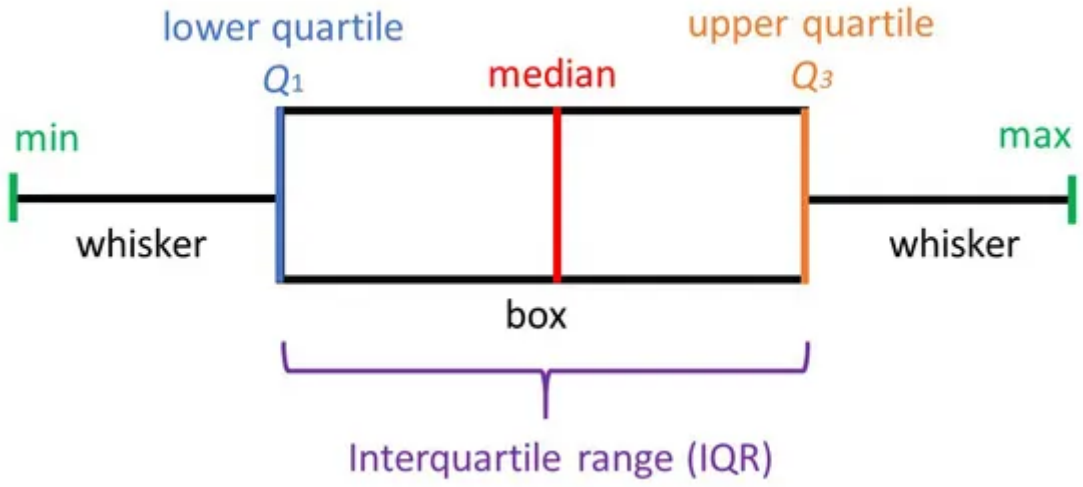
- 최대값 - 최소값
  - 예: 10, 20, 30, 40, 50의 경우 최대값(50) - 최소값(10) = 40
- 극단값이 있으면 커짐
  - 예: 10, 20, 30, 40, 500의 경우 490

사분위간 범위 *InterQuartile Range*

- 줄여서 IQR
- 제3사분위수 - 제1사분위수 또는 75퍼센타일 - 25퍼센타일
- 극단값은 최소값 또는 최대값 근처에 있으므로 극단값의 영향이 적음

상자 그림 *box plot*

- 제1사분위수 ~ 제3사분위수를 상자로 표현
- 중간값은 상자의 가운데 굵은 선으로 표시
- 최소값과 최대값은 수염 *whisker*로 표시
- 수염의 최대 길이는 IQR의 1.5배까지, 넘어가는 경우는 점으로 표시



편차 *deviation*

$$X_i - \bar{X}$$

- 값 - 평균
  - 원 데이터가 30, 40, 50인 경우(평균 40)
  - 편차는 -10, 0, +10

분산 *variance*

$$\mathrm{Var}(X) = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2$$

- 편차 제곱의 평균
- 직관적으로 이해하기는 어려우나 수학적으로 중요한 여러 성질이 있음
- 편차를 제곱하여 크기가 커지므로 표준편차 *standard deviation*를 많이 사용

$$\mathrm{SD}(X) = \sqrt{\mathrm{Var}(X)}$$

중심경향치

- 평균

```
df.price.mean()
```

- 중간값

```
df.price.median()
```

- 빈도표

```
df.model.value_counts()
```

최소, 최대, 분위수

- 최소

```
df.price.min()
```

- 최대

```
df.price.max()
```

- 분위수

```
df.price.quantile(.25)
```

분산과 표준편차

- 분산

```
df.price.var()
```

- 표준편차

```
df.price.std()
```

상자 그림

시각화 라이브러리 `seaborn` 불러오기

```
import seaborn as sns
```

가격의 상자 그림

```
sns.boxplot(y='price', data=df)
```

모델별로 나누어 그리기

```
sns.boxplot(x='model', y='price', data=df)
```

히스토그램 *histogram*

```
sns.histplot(x='price', data=df)
```

막대의 개수 바꾸기

```
sns.histplot(x='price', data=df, bins=30)
```

추론 통계

추론 통계

- 모집단 *population*: 연구의 관심 대상이 되는 집단 전체
- 표본 *sample*: 특정한 연구에서 선택된 모집단의 부분 집합
- 표집 *sampling*: 모집단에서 표본을 추출하는 절차
- 추론 통계: 표본 통계량을 일반화하여 모집단에 대해 추론 하는 것

모수와 통계량 *Parameter and Statistic*

- 모수 *Population Parameter*: 모집단의 특성을 기술하는 양
- 통계량 *Sample Statistic*: 표본에서 얻어진 수로 계산한 값 (=통계치)
- 통계량으로부터 모수를 추정
- 주의: population statistic이나 sample parameter는 없음

추정 *estimation*

- 통계량으로부터 모수를 추측하는 절차
- 점추정 *point estimation*: 가장 가능성이 높은 모수 하나를 추정
  - 예시: 모평균의 가장 가능성 높은 추정치 = 표본평균
- 구간 추정 *interval estimation*: 모수가 있을 가능성이 높은 범위를 추정

표집 오차 *sampling error*

- 모집단과 표본의 차이
- 표본의 크기가 클 수록 표집 오차는 작아짐
- 동일한 모집단에서 동일한 절차를 거쳐 추출한 표본끼리도 차이가 존재

신뢰구간 *confidence interval*

- 신뢰구간 = 표본 통계량 ± 오차 범위
- 대표적인 구간 추정 방법
- 표본 통계량은 모수보다 높을 수도 있고 낮을 수도 있음
- 모수의 근처 ± 오차 범위 내에 있을 확률이 높음
- 반대로 말하면 표본 통계량에서 ± 오차 범위 내에 모수가 있을 확률이 높음

신뢰수준 *confidence level*

- 신뢰구간을 무한히 넓게 예측하면 → 100% 모수를 포함 (신뢰수준 100%)
- 예측을 하는 이유는 행동을 하거나 결정을 하기 위해서 → 무한히 넓은 구간의 예측은 쓸모가 없음
- 신뢰구간을 좁게 예측하면 → 모수를 포함하지 못하는 경우가 생김 (신뢰수준 ↓)
- 신뢰구간을 극단적으로 좁히면 → 신뢰수준이 지나치게 낮아 쓸모가 없음
- 95%, 99% 등 일정한 신뢰수준으로 타협 (교과서적으로는 95%)

유의수준 *significance level*

- 유의수준 = 100% - 신뢰수준
- 추정한 구간이 모수를 포함하지 못할 확률

혼동 주의

- 일상적 표현에서는 신뢰도가 높으면 (측정) 오차가 적음 → 측정의 관점
- 통계에서 신뢰수준이 높으면 오차범위가 넓음 → 추론의 관점
- 예: 주사위 던지기의 경우
  - 측정 오차는 없음
  - 모수 추정에서 오차 범위가 존재 (각 눈이 나올 확률에 대한 확신의 부족)

신뢰구간에 영향을 주는 요소

- 신뢰구간이 좁을 수록 예측된 모수의 범위가 좁으므로 유용
- 신뢰수준 ↓
  - 큰 의미는 없음
- 표본의 변산성 ↓
  - 실험과 측정을 정확히 해서 변산성을 낮춤
  - 데이터에 내재한 변산성은 없앴 수 없음
- 표본의 크기 ↑
  - 가장 쉬운 방법이나 시간과 비용이 증가

신뢰구간을 구하는 방법

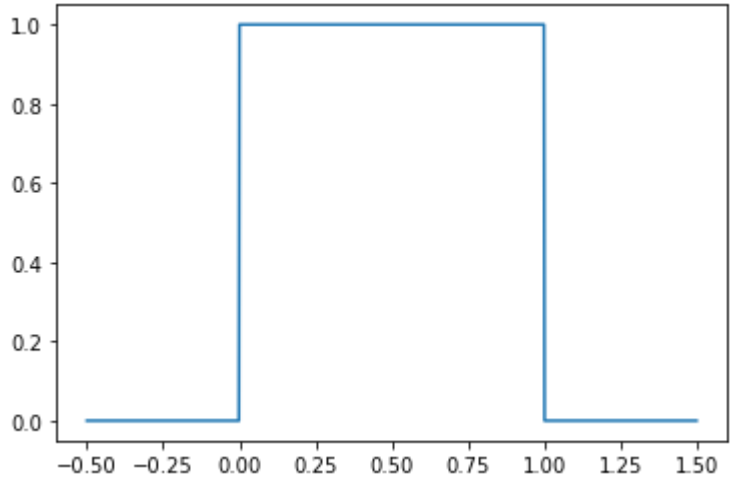
- 일정한 수학적 가정으로부터 표집 분포를 유도
  - 예: 평균의 경우, 중심극한정리를 이용
- 부트스트래핑(bootstrapping): 현재 가진 표본에서 재표집을 반복하여, 표집 오차를 시뮬레이션

중심극한정리 *Central Limit Theorem*

- 표본이 클 수록, 표본 평균의 분포가 정규 분포에 수렴
  - 평균이 μ이고 표준편차가 σ인 분포에서
  - 서로 독립 *independent* 이며 동일한 분포를 따르는 *identically distributed* (i.i.d.) 무작위 표본을 뽑았을 때
  - 이 표본의 크기가 n일 경우
  - 표본 평균의 표집 분포는 평균이 μ이고 표준편차가 $\sigma / \sqrt{n}$인 정규분포를 따름
- 중요한 점
  - 모집단의 분포에 무관
  - 표본의 분포가 정규 분포에 수렴하는 것이 아님

중심극한정리 모의실험

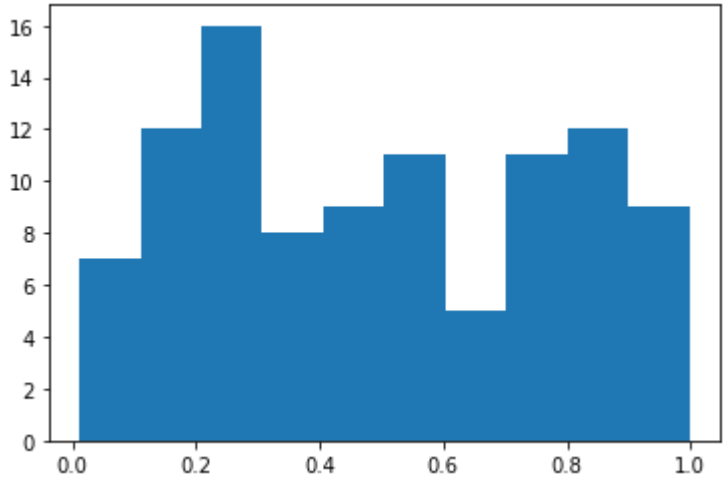
```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import uniform, norm
x = np.linspace(-.5, 1.5, 1000)
p = uniform.pdf(x)  # 균등분포
plt.plot(x, p)
```



표본 분포

- 균등분포에서 데이터 100개를 무작위 추출
- 1개의 표본
- 대체로 모집단의 분포(=균등 분포)와 비슷

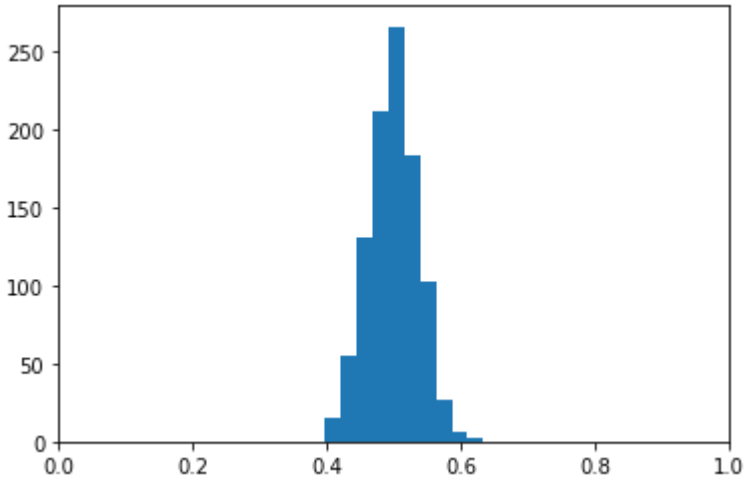
```
x = uniform.rvs(size=100)
plt.hist(x)
```



표집 분포 *sampling distribution*

- 64개의 데이터를 뽑아 평균 내기를 1000회 반복 → 1000개의 표본, 1000개의 표본 평균
- 표본 평균의 분포는 균등 분포와 달리 0.5 근처에 몰려있음

```
m = [uniform.rvs(size=64).mean() for _ in range(1000)]
plt.hist(m)
plt.xlim(0, 1)
```



Student의 t 분포

- 통계학자 윌리엄 고셋 *William Gosset* 이 발견한 확률 분포
  - Student는 고셋의 필명
- 모분산을 알면, 평균의 신뢰구간을 구할 때 중심극한정리에 따라 정규분포를 사용
- 모분산을 모르면, t 분포를 사용
- 표본의 크기가 충분히 크면, t 분포는 정규분포에 근사

충분히 크면 *sufficiently large*

- 어떤 변수 n에 따라 참 거짓이 변하는 명제 $P(n)$가 있을 때, $n > N$에서 $P(n)$이 항상 참인 어떤 N이 존재하는 경우
- n이 충분히 크면 참인 명제:

$$\frac{1}{n^2} < 0.0001$$

- n이 충분히 크면 참인 명제가 아님:

$$\sin n < 0.5$$

- n의 크기보다 점점 특정한 결론으로 수렴하는 형태에 관한 표현

pingouin

Python 통계 분석 라이브러리

설치

```
pip install pingouin
```

임포트

```
import pingouin as pg
```

Python 단일표본 t 검정

t 검정

```
pg.ttest(df.price, 0)
```

99% 신뢰구간

```
pg.ttest(df.price, 0, confidence=.99)
```

Python 부트스트래핑

임포트

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

부트스트랩으로 95% 신뢰구간 계산

```
bootstrap([df.price], np.mean, confidence_level=.95)
```

통계적 가설 검정 *statistical hypothesis testing*

- 유의 수준을 결정 (예: 5%)
- 귀무가설을 설정
  - 귀무(歸無): 무로 돌아간다. 즉, 기각할 *nullify* 가설을 의미
  - 예: 모수 = 0
- p 값: 귀무가설의 모수를 포함할 때까지 신뢰구간을 넓혔을 때의 유의 수준
- p값 < 유의수준: 귀무가설을 기각 → 통계적으로 유의하다

통계적 유의함의 의미

- 유의수준 내에서 귀무가설을 기각할 수 있을만큼의 증거가 있음
- 표본의 크기가 충분하다는 의미로 이해
- 현실적으로 유의미하다는 것이 아님

통계적 가설 검정에서 오류의 종류

- 1종 오류(False Alarm): 귀무가설이 참일 때 기각
- 2종 오류(Miss): 귀무가설이 거짓이나 기각 못함
- 통계적 가설 검정의 절차를 따르면 1종 오류는 유의수준만큼 발생
- 동일 조건에서 유의 수준을 낮추면 1종 오류 ↓, 2종 오류 ↑

상관 분석

상관 계수 *correlation coefficient*

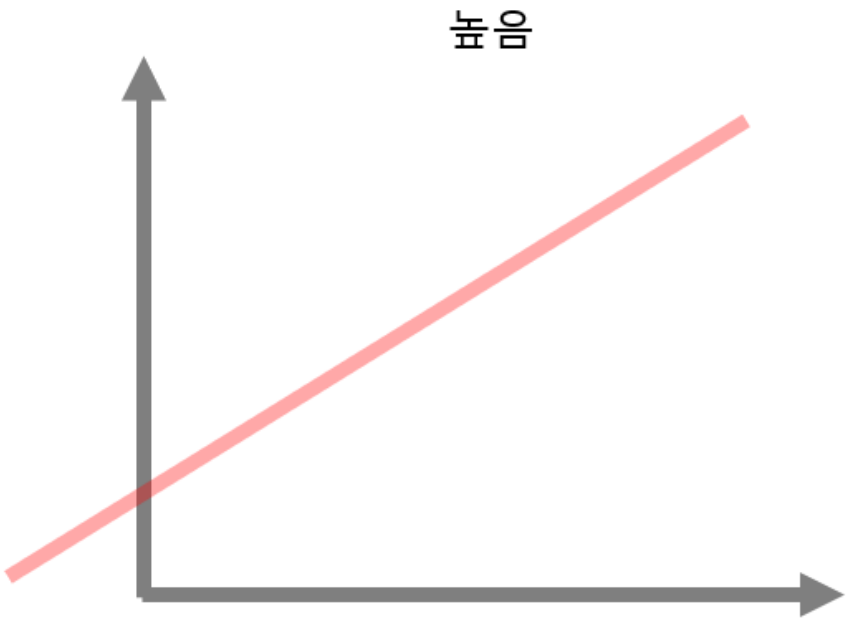
- 두 변수의 연관성을 -1 ~ +1 범위의 수치로 나타낸 것
- 두 변수의 연관성을 파악하기 위해 사용
  - 어휘력과 독해력의 관계
  - 주가와 금 가격의 관계
  - 엔진 성능과 고객만족도의 관계
- 본격적인 분석전에 탐색적 분석을 위해 많이 사용
- 데이터의 진단에도 사용: 설문 등의 경우 관련 문항 간에는 높은 상관관계가 있어야 함

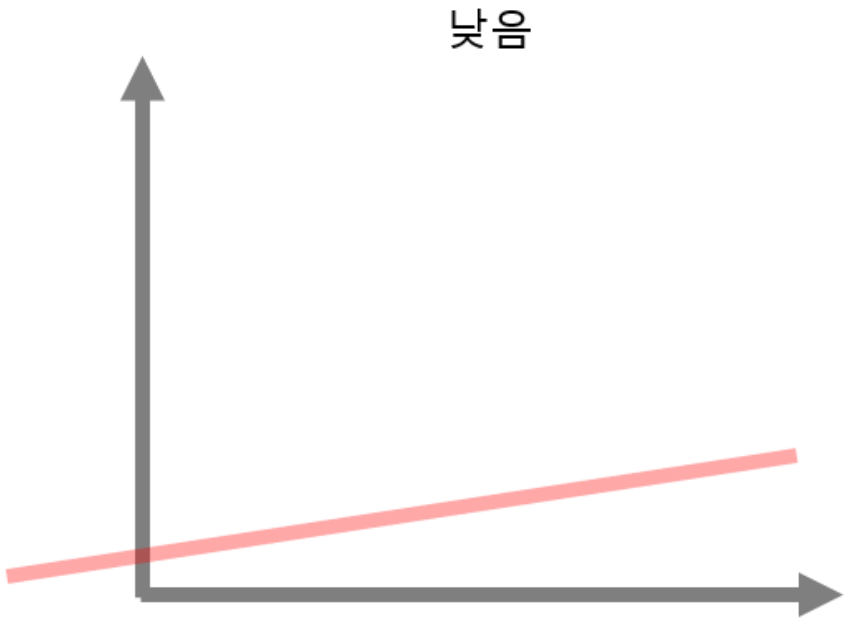
상관계수의 해석

- 부호:
  - +: 두 변수가 같은 방향으로 변화(하나가 증가하면 다른 하나도 증가)
  - −: 두 변수가 반대 방향으로 변화(하나가 증가하면 다른 하나는 감소)
- 크기:
  - 0: 두 변수가 독립, 한 변수의 변화로 다른 변수의 변화를 예측하지 못함
  - 1: 한 변수의 변화와 다른 변수의 변화가 정확히 일치

기울기

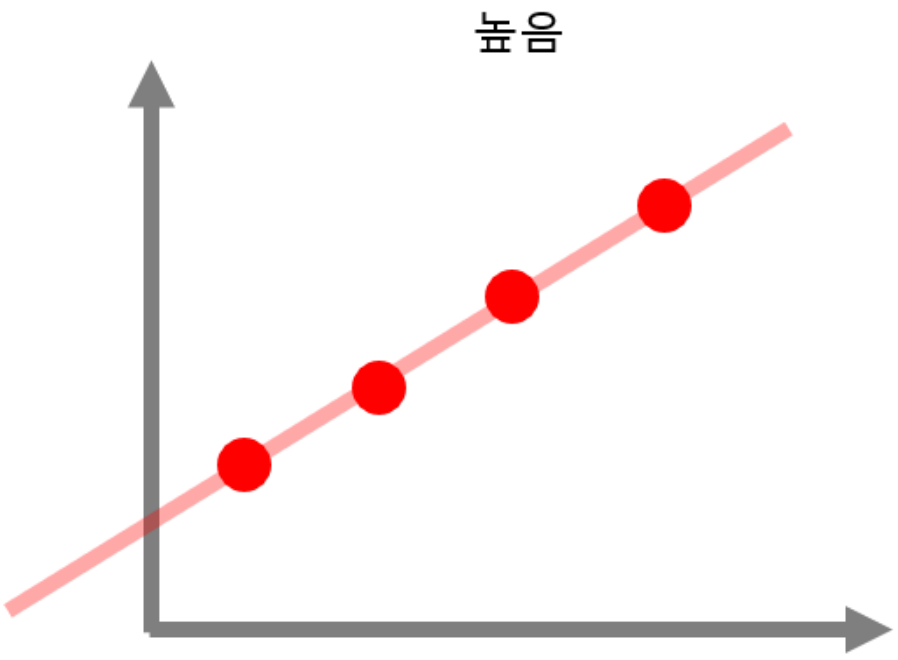
- $y = wx + b$에서 w
- x가 1만큼 변활 때, y의 변화량
- 기울기가 클 수록 y가 많이 변함

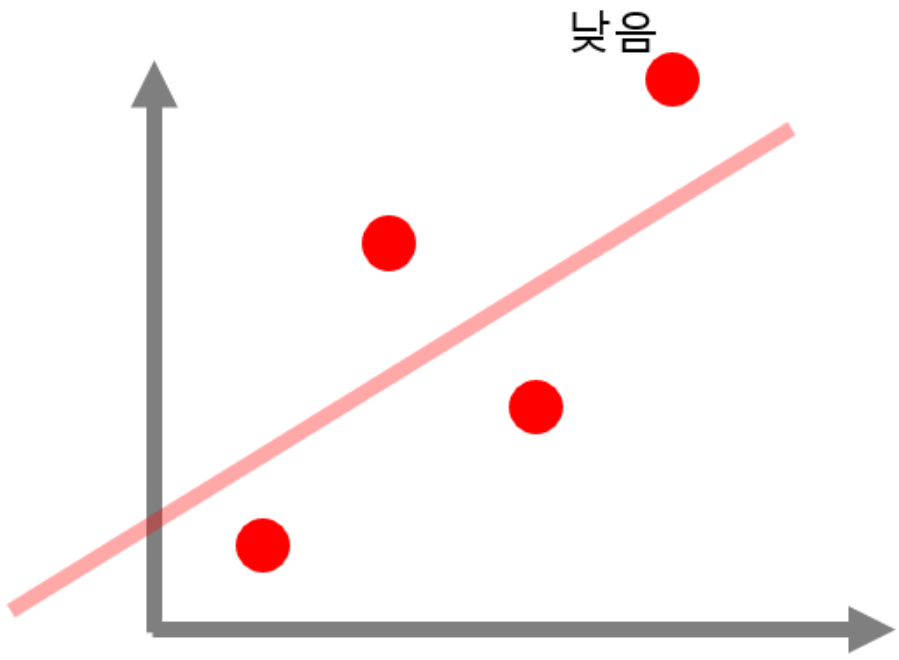




상관계수

- 두 변수의 관계의 강함
- -1~+1 범위
- 절대값이 클 수록 관계가 강함





피어슨 적률 상관계수

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma Y}$$

- 가장 대표적인 상관계수
- 선형적인 상관계수를 측정
- 공분산을 두 변수의 표준편차로 나눔 → -1 ~ +1 범위

상관계수와 비단조적 관계

- 상관계수는 우상향 또는 우하향하는 단조적 관계를 표현
- 복잡한 비단조적 관계는 잘 나타내지 못함
- 상관계수가 낮다고 해서 관계가 없는 것은 아님

Python 상관 분석

```
import pingouin as pg
pg.corr(df.price, df.mileage)
```

- 피어슨 상관계수
- `r`: 상관계수
- `CI95%`: 95% 신뢰구간
- `p-value`: 가설검정을 위한 p 값

상관계수의 신뢰구간

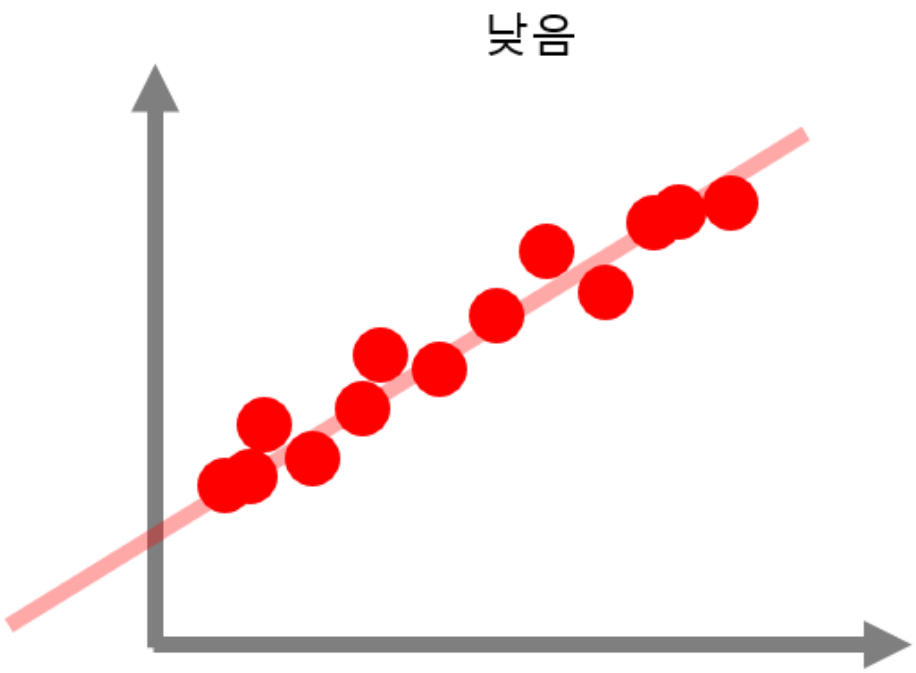
- `+ ~ +`: 모집단에서 두 변수의 관계가 +
- `- ~ +` : 모집단에서 두 변수의 관계는 −, 0, + 모두 가능
- `- ~ -` : 모집단에서 두 변수의 관계가 −

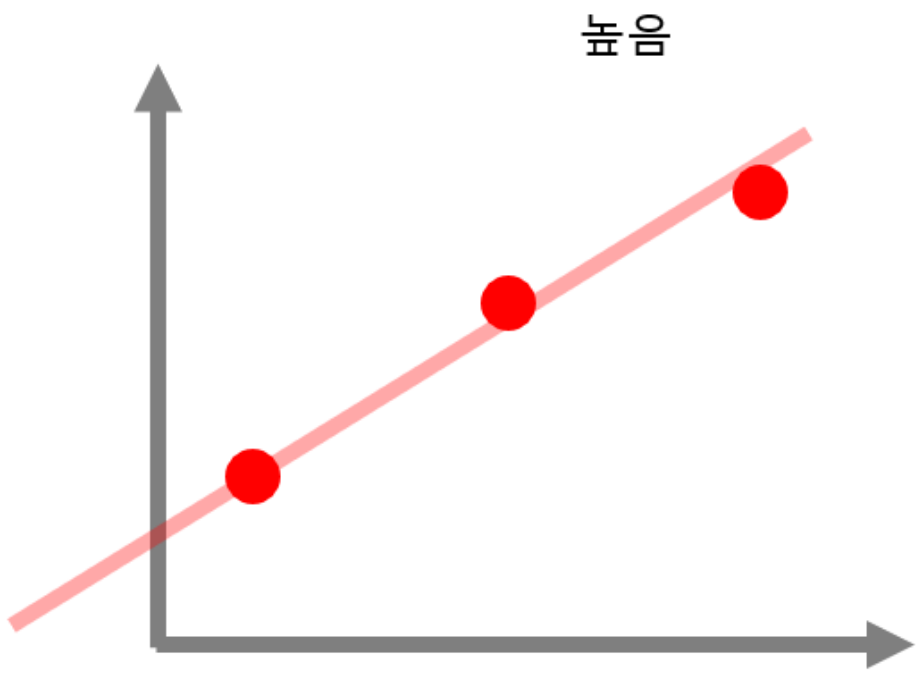
상관계수의 가설 검정

- 모집단에서 두 변수가 상관관계가 없다(상관계수 = 0)이라는 귀무가설 *null hypothesis*을 수립
- 귀무가설이 참일 때 관찰된 통계량 이상의 결과가 관찰될 가능성(p 값)을 계산
- p값과 유의수준(통상 5%)을 비교
  - p 〈 유의수준: 귀무가설을 기각하고 모집단에서 두 변수가 상관관계가 있다고 결론
  - p ≥ 유의수준: 결론을 유보(관련이 있을 수도 없을 수도 있음)
    - 결론을 원하면 더 많은 데이터를 수집

p-value

- 증거의 부족함
- 0~1 범위, 보통 0.05를 기준으로 함(유의수준 5%)
- $p < .05$이면 기울기가 + 또는 -라고 결론 내릴 수 있음





상관계수의 크기 해석

- 상관계수의 크기에 대해서는 몇 가지 권장 기준이 있음(예: Cohen, 1988)
  - 낮음 ~0.1
  - 중간 0.1 ~ 0.5
  - 높음 0.5 ~
- 엄밀한 근거에 바탕을 둔 것은 아님
- 실제 의사결정에서는 상대적으로 비교하는 것이 바람직
  - 예를 들어 상관계수 0.2인 요소 A와 0.3인 요소 B가 있고, 예산상 상관이 높은 한 가지 요소만 고려할 수 있다면 요소 B를 고려

회귀 분석

지도학습 *supervised learning*

- 독립변수 x를 이용하여 종속변수 y를 예측하는 것
  - 독립변수 *independent variable*: 예측의 바탕이 되는 정보, 인과관계에서 원인, 입력값
  - 종속변수 *dependent variable*: 예측의 대상, 인과관계에서 결과, 출력값
- 종속변수의 종류에 따른 구분
  - 회귀분석 *regression*: 종속변수가 연속(예측 - 실제가 작은 것이 중요)
  - 분류분석 *classification*: 종속변수가 범주형(예측과 실제가 맞는 것이 중요)

선형 모형 *linear model*

$$\hat{y} = wx + b$$

- $\hat{y}$: y의 예측치
- x: 독립변수
- w: 가중치 또는 기울기
- b: 절편($x = 0$ 일 때, y의 예측치)

잔차 *residual*

- 잔차: 실제값 y과 예측값 $\hat{y}$의 차이
- 잔차분산: 잔차를 제곱하여 평균낸 것
  - cf. 분산: 편차(실제값 y과 평균 $\bar{y}$의 차이) 제곱의 평균

$$\frac{1}{N}\sum(y-\hat{y})^2$$

- 잔차분산이 크다 → 예측이 잘 맞지 않음
- 잔차분산이 작다 → 예측이 잘 맞음

최소제곱법 *Ordinary Least Squares*

- 최소제곱법: 잔차 분산이 최소가 되게 하는 w, b 등 계수를 추정
- 최소'제곱'법인 이유: 분산의 계산에 제곱이 들어가므로
- 가장 널리 사용되는 추정방법

Python 회귀분석

```
import pandas as pd
from statsmodels.formula.api import ols
```

- 임포트

```
df = pd.read_excel('car.xlsx')
```

- 데이터 불러오기

Python 회귀분석

```
m = ols('price ~ mileage', data = df).fit()
```

- 분석
- `ols`는 최소제곱법(OLS)를 뜻함

```
m.summary()
```

- 결과

회귀계수 추정 결과

- `Intercept`는 절편 b, 나머지는 각 변수의 계수
- 계수 추정 결과는 추정치, 표준오차, t값, p-value, 신뢰구간 순
  - 표준오차, t값은 p-value를 구하기 위한 중간 결과로 직접 해석 필요 X
- 회귀계수의 가설검정: 모집단에서 기울기 = 0을 귀무가설로 p값 계산
  - p < 유의수준(통상 5%) → 기울기 ≠ 0으로 결론
  - p ≥ 유의수준 → 결론을 유보(필요하면 데이터를 더 모음)

Python 예측

회귀분석

```
df = pd.read_excel('car.xlsx')
m = ols('price ~ mileage', df).fit()
```

새로운 데이터 만들기(엑셀에서 아래와 같이 입력하고 `car2.xlsx`로 저장)

	A
1	mileage
2	10000
3	20000

모형에 대입하여 예측

```
new_df = pd.read_excel('car2.xlsx')
m.predict(new_df)
```

결정 계수 또는 R제곱

- R제곱 *R-Squared*: 회귀분석에서 예측의 정확성을 알기 쉽게 표현한 지표(0~1)
  - R제곱 = 0: 분석 결과가 y의 예측에 도움이 안됨
  - R제곱 = 1: y를 완벽하게 예측할 수 있음
- 단순회귀분석(독립변수가 1개인 회귀분석)의 경우, 회귀분석의 R제곱 = 독립변수와 종속변수의 피어슨 상관계수의 제곱

독립변수가 범주형인 경우

- 범주형 변수는 기울기를 곱할 수 없음
- 연속 변수로 변환하여 모형에 투입
- 여러 가지 방법이 있으나 가장 많이 사용하는 것은 더미 코딩
- Jamovi, R, Python은 자동으로 더미 코딩

더미 코딩 *dummy coding*

- 범주형 변수에 범주가 k개 있을 경우 k-1개의 더미 변수를 대신 투입
- 범주 중에 하나를 기준 *reference*로 지정
- 기본적으로 ABC 순으로 먼저 나오는 것이 기준(변경할 수도 있음)
- 기준를 제외한 범주들은 범주별로 더미 변수를 하나씩 가짐
- 더미변수는 해당 범주일 경우에만 고려
- 더미변수의 기울기는 기준과의 차이를 의미

Python 범주가 2개인 경우

```
ols('price ~ model', df).fit().summary()
```

- Avante가 기준, K3의 더미변수 `model[T.K3]`가 추가
  - Avante 예상 가격: 833만원
  - K3 예상가격: 833+80=913만원

Python 범주 목록 보기

- `unique` 함수를 사용하면 변수에서 범주의 목록을 확인 할 수 있음
- 기준 범주는 더미 변수가 없으므로, 범주 목록에서 확인

```
df.model.unique()
```

범주가 3개인 경우

- 데이터 파일`depression.xlsx`
  - `y`: 치료 효과
  - `TRT`: 치료 방법(A, B, C), A가 기준

관계식

```
y ~ TRT
```

- A(기준)의 치료효과: 62.3333
- B의 치료효과: 62.3333 - 10.4167 = 51.9166
- C의 치료효과: 62.3333 - 11.0833 = 51.2500

Python 기준범주 바꾸기

- `C` 함수로 변수를 범주형으로 지정
- `Treatment`로 기준 범주를 지정
- 따옴표에 주의

```
m = ols('price ~ C(model, Treatment("K3"))', df).fit()
```

다중회귀분석

- 독립변수가 2개 이상인 회귀분석
- R과 Python에서는 관계식에서 +로 변수를 구분

```
price ~ mileage + model
```

- 더하라는 뜻이 아님에 주의

통계적 통제

- 독립변수 x와 상관관계가 높은 요소 z가 존재할 경우
- z가 종속변수 y에 미치는 영향이 x의 기울기에 간접 반영될 수 있음
- 실험적 통제: 데이터에서 z를 일정하게 유지하여, z의 영향을 제거
- 통계적 통제: z를 모형에 독립변수로 함께 포함하여, x의 기울기에 z의 영향이 간접 반영되지 않도록 함

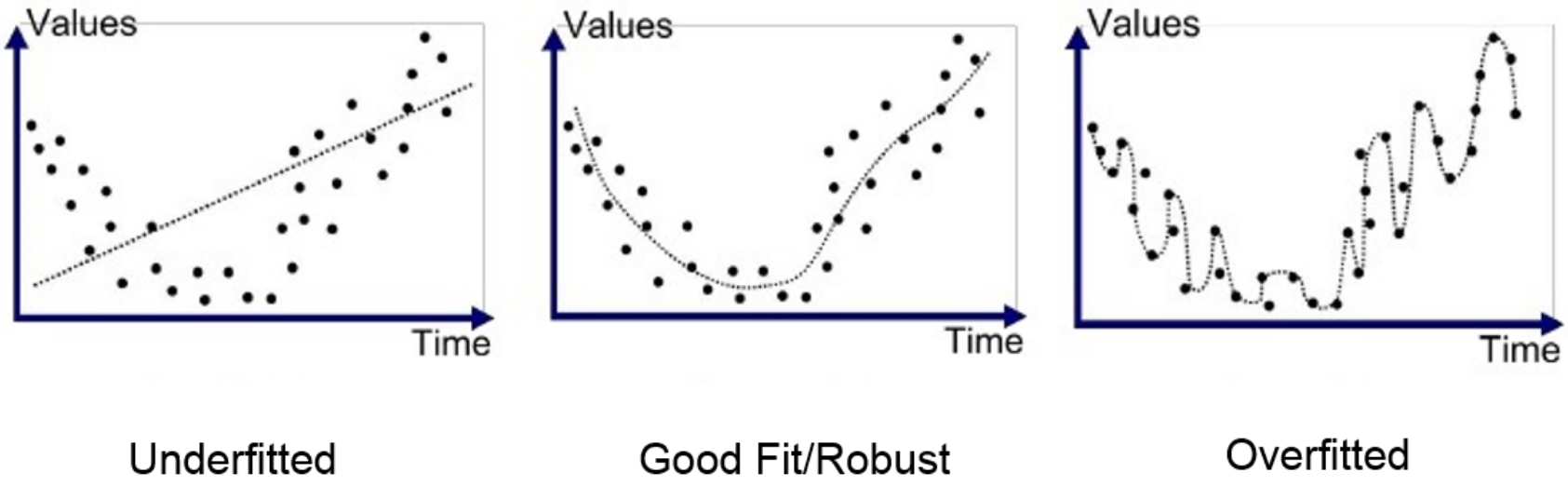
표준화 *standardization*

$$\frac{X - \mu}{\sigma}$$

- μ: 평균
- σ: 표준편차
- 변수별로 퍼진 정도(=분산)을 비슷하게 맞춰주는 절차
  - 표준화를 하면 평균 = 0, 표준편차 = 1이 됨
  - 단위가 다른 변수의 기울기를 비교할 때 사용 → 단위를 없애는 효과
- 관계식에는 `y ~ scale(x1) + scale(x2)` 형식으로 사용
  - 범주형 변수는 표준화 하지 않음

과대적합 *overfitting*

- 최소제곱법은 잔차분산이 가장 작은 계수를 추정
- 주어진 표본에 가장 맞는 계수를 찾게 됨
- 표집 오차가 존재하기 때문에, 주어진 표본에 지나치게 맞는데 계수를 추정하면 모집단의 계수와 다를 수 있음



수정 R제곱과 AIC, BIC

- 독립변수의 개수가 다른 모형을 비교할 경우, R제곱으로는 비교 어려움
- R제곱은 독립변수가 많을 수록 높아지는 경향이 있음
- 독립변수의 개수를 이론적으로 보정한 수정 R제곱, AIC, BIC 등의 지수 사용
- 수정 R제곱 *Adjusted R-Squared*: R제곱을 보정 → 클 수록 좋음
- AIC와 BIC: 잔차분산을 보정 → 작을 수록 좋음

수정 R제곱 *Adjusted R-squared*

$$\text{Adjusted } R^2 = 1 - \frac{RSS/(n-k-1)}{TSS/(n-1)}$$

- n: 표본의 크기
- k: 독립변수의 개수

독립변수를 추가하면 R^2이 작아지도록 보정

수정 R제곱이 클 수록 좋은 모형

주의: 수정 R제곱은 모형 간 비교의 용도. 한 모형이 종속변수의 분산을 설명하는 비율을 볼 때는 R제곱을 볼 것.

AIC *Akaike Information Criterion*

$$n\log(\frac{\mathrm{RSS}}{n})+2k$$

AIC는 작을 수록 좋은 모형

BIC *Bayesian Information Criterion*

$$n \log(\frac{\mathrm{RSS}}{n}) + k \log n$$

BIC는 작을 수록 좋은 모형

교차 검증

- 수정R제곱, AIC, BIC 등은 이론적 보정이므로 과적합을 정확히 반영 못함
- 데이터가 충분히 많다면, 데이터를 여러 개의 셋으로 나누어 교차 검증
- 한 데이터셋의 분석 결과를 다른 데이터셋에 적용하여 예측 오차를 확인 (예측 오차가 적은 모형이 좋은 모형)
- 이론적 가정에 의존하지 않으므로 데이터가 충분히 많을 때는 교차 검증을 권장

교차 검증의 종류

- LpO CV (Leave-p-out): p개를 제외한 모든 사례로 추정에 사용. p개는 가능한 모든 방법으로 조합. 조합이 지나치게 많아 비현실적
- LOOCV (Leave-one-out): p = 1인 경우. 데이터가 N개이면 N번 검증
- k-fold: 데이터를 크게 k개의 셋으로 나눔. 한 셋 씩 테스트셋으로 사용. k번 교차검증
- holdout: 데이터를 훈련 셋과 테스트 셋으로 한 번만 나누어 1회 교차 검증

교차 검증의 결과

- 훈련 오차와 테스트 오차가 모두 높은 경우
  - 과소적합
  - 모형을 더 복잡하게 수정
- 훈련 오차와 테스트 오차가 모두 낮은 경우: 바람직
- 훈련 오차는 낮고, 테스트 오차는 높은 경우
  - 과대적합
  - 모형을 더 단순하게 수정

Python 데이터 분할

임포트

```
from sklearn.model_selection import train_test_split
```

분할

```
train_df, test_df = train_test_split(
    df,                 # 원자료
    test_size=0.2,      # 테스트 데이터의 비율(0.2 = 20%)
    random_state=42)  # 난수 생성의 seed를을 고정(동일한 분할을 위해)
```

변수의 변형

- 선형 모형은 독립변수와 종속변수의 선형적 관계를 가정한다는 한계
- 독립변수를 비선형 변환하면 이 한계를 일부 극복할 수 있음
- R과 Python은 관계식에 수학 함수를 사용하면 자동으로 변수 변환

로그 함수

- 오른쪽 위로 갈 수록 완만해지는 형태
- 가로축에서 1, 10, 100이 세로축에서 같은 간격(예: 0, 1, 2)
- 데이터에 적용하면 오른쪽을 왼쪽으로 끌어당기는 효과
- 독립변수에 오른쪽으로 크게 떨어져 있는 값이 있는 경우(예: 소득), 로그 함수를 적용해주면 간격을 일정하게 만들어 줄 수 있음

```
import numpy as np
m = ols('price ~ np.log(mileage)', df).fit()
```

I 함수

- 관계식에 덧셈, 곱셈, 거듭제곱 등을 할 경우 적용이 불가
- R과 Python은 I 함수를 사용하여 이러한 계산을 적용 가능

```
y ~ I(x + z)
```

- 두 독립변수 x와 z를 더하여 하나의 변수로 변환

Python 2차항의 추가

- $y = ax^2 + bx + c$와 같은 모형을 관계식으로 만들 경우
- 거듭제곱에 `**`를 사용

```
y ~ I(x**2) + x
```

절편이 없는 모형

- 절편이 없는 모형 $y = wx + 0$을 표시하기 위해서는 관계식에 0 +를 추가

```
y ~ 0 + x
```

절편의 이동

- 절편은 $x = 0$일 때의 예측치
- 절편을 $x = 100$일 때의 예측치로 바꾸려면 x에 일괄적으로 100을 빼면 됨

```
y ~ I(x - 100)
```

- 분석 자체에는 영향이 없으나 절편의 해석이 더 쉬워질 수 있음

상호작용 *interaction*

- 상호작용 항: 두 독립변수의 곱으로 이뤄진 항

$$y = x + m + xm$$

- 관계식으로 쓸 때는 `:`을 사용

```
y ~ x + m + x:m
```

- 위의 식은 다음과 같이 축약할 수 있음(`*`의 의미가 일반적 용법 다르므로 주의)

```
y ~ x*m
```

상호작용의 해석

- 간단히 `x`는 연속형, `m`은 이분 범주형(0과 1만 있는 경우)이라고 할 때
- `y ~ x + m`: m에 따라 x의 절편이 바뀌는 것으로 해석할 수 있음
- `y ~ x + x:m`: m에 따라 x의 기울기가 바뀌는 것으로 해석할 수 있음
- `y ~ x + m + x:m`: m에 따라 x의 기울기와 절편이 바뀌는 것으로 해석할 수 있음

증거의 사다리

- 인과관계의 증거 수준

1. 실험적 통제
2. 무작위 대조군
3. 준실험
4. 반사실

실험적 통제 *experimental control*

- 처치를 제외한 다른 모든 조건을 동일하게 유지
- 인과관계를 확인할 수 있는 최선의 조건
- 매우 한정된 조건에서만 가능
- 예: 진공상태에서 물체의 낙하 실험

무작위 대조군 연구 *randomized controlled trials*

- 모든 조건을 완벽하게 통제할 수 없을 경우
- 실험군과 대조군에 무작위 할당
- 표집 오차가 있을 수 있음
- 예: 신약 임상 시험

준실험 *quasi-experiment*

- 대조군이 없거나 무작위 할당을 하지 않았지만 실험과 비슷한 상황
- 자연적으로 무작위 할당과 비슷한 결과가 생긴 경우
- 예: 인접한 두 주 *state*에서 최저임금을 다르게 인상한 경우

반사실 *counterfactual*

- 순수한 관찰 결과만을 가지고 인과관계를 추측
- 어떤 일이 벌어지지 않았을 때 일어날 일을 예측하는 모형이 필요
- 모형의 예측과 실제의 결과를 비교하여 인과관계를 추론
- 예: 예상 매출액과 비교하여 광고 효과를 추론

횡단 비교와 종단 비교

- 횡단 *cross-sectional* 비교: 동일 시점에 다른 대상이나 집단을 비교
- 종단 *longitudinal* 비교: 동일 대상을 다른 시점 간 비교

	집단 A	집단 B	횡단 비교
시점1	A1	B1	B1 - A1
시점2	A2	B2	B2 - A2
종단비교	A2-A1	B2 - B1	

이중차분법 *Difference-in-Differences*

- 실험이 불가능한 상황에서 사용하는 준실험적 방법
- 실험군 B에 어떤 처치를 했으나 대조군이 없을 때
- 실험군과 비슷한 집단 A를 이용하여 비교

$$d = (B_2 - B_1) - (A_2 - A_1)$$

- 결과 해석
  - $d = 0$: 실험군 B에서 변화는 대조군 A에서 변화와 비슷(처치 효과 없음)
  - $d \neq 0$: 실험군 B에서 대조군 A와 다른 변화를 관찰(처치 효과 있음)

평행 추세의 가정 *parallel trend assumption*

- 처치 효과가 없다면 실험군과 대조군의 비슷하게 변할 것이라고 가정
- 이러한 가정이 성립하지 않는다면 이중차분법의 결과는 무의미
- 가능한 비슷한 A와 B를 비교하는 것이 중요

회귀분석을 통한 이중차분법

상호작용을 이용해 분석

$$y = a \cdot \mathrm{GROUP} + b \cdot \mathrm{POINT} + d \cdot (\mathrm{GROUP} \times \mathrm{POINT}) + e$$

- GROUP: 집단 A(0), 집단 B(1)
- POINT: 처치 전(0), 처치 후(1)

	대조군 A	실험군 B	횡단 비교
시점1	e	a + e	a
시점2	b + e	a + b + d + e	a + d
종단비교	b	b + d	d

Card & Krueger (1994)

- 1992년 미국 뉴저지 주는 최저 임금을 시간당 4.25달러에서 5.05달러로 인상
- 이웃 펜실베니아 주는 4.25달러 최저 임금을 유지
- 두 주 경계에 위치한 패스트푸드 음식점을 대상으로 고용 변화(FTE)를 조사
- 최저임금의 상승이 고용에 미치는 효과를 이중차분법으로 측정
- 데이비드 카드는 2021년 노벨 경제학상 수상
- `njmin3.xlsx`
  - `fte`: 전일제 환산 고용률
  - `nj`: 뉴저지(1)/펜실베니아(0)
  - `d`: 최저임금 인상 전(0)/후(1)

Python 이중차분법

파일 열기

```
df = pd.read_excel('njmin3.xlsx')
```

이중차분법으로 분석

```
m = ols('fte ~ nj + d + nj:d', data=df).fit()
m.summary()
```

분석 결과 해석

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    23.331      1.072  21.767   <2e-16 ***
nj             -2.892      1.194  -2.423   0.0156 *
d              -2.166      1.516  -1.429   0.1535
nj:d            2.754      1.688   1.631   0.1033
```

	펜실베니아(nj=0)	뉴저지(nj=1)	횡단 비교
시점1(d=0)	28.331	25.439	-2.892
시점2(d=1)	26.165	28.919	
종단비교	-2.166		2.754