사내 실무 강의 · 180분 집중

WORLD BANK WDI · 실데이터

데이터 고급 통계 분석 실무 활용

AI로 통계 분석의 **난이도를 낮춘다** — 외부망에서 배우고, 폐쇄망 STATA로 현업에 적용

데이터	대상	핵심 도구	최종 산출물
WDI 패널 · 217국×23년	Python·STATA 입문 동료	생성형 AI · Python · STATA	폐쇄망 STATA 코드 꾸러미

오늘 우리는 —

회귀 · 교차표 · 패널 분석을 직접 돌립니다.
단, 문법은 한 줄도 외우지 않습니다.

오늘의 약속

강의가 끝나면 폐쇄망 STATA에서 바로 돌아가는 코드 꾸러미를 손에 쥐고 돌아갑니다.

우리 회사엔 두 개의 환경이 있다

이 강의의 전제 — '배우는 곳'과 '일하는 곳'이 분리되어 있다.

외부망 · 학습·탐색

여기서 배우고 만든다

- ✓ 생성형 AI 사용 가능 (코드 생성)
- ✓ Python 사용 가능
- ✓ 인터넷 연결 · 실습 제약 없음

→

코드로
변환·반입

폐쇄망 · 현업·실행 (air-gapped)

여기서 돌아간다

- ✗ 인터넷 X · AI X · Python X
- ● STATA O ← 실제 업무는 여기서
- ● 결과물은 여기서 실행된다

전략 · AI·Python으로 **외부망에서 빠르게 배워** → 폐쇄망 STATA에서 그대로 돌아가는 **코드 꾸러미**로 변환해 들고 돌아간다

이 강의의 두 축

난이도는 **AI**가 낮추고, 신뢰는 **인간**이 높인다.

축 ①

AI가 난이도를 낮춘다

1. **설계** — 무엇을·왜 분석할지 · 사람
2. **생성** — 코드를 AI에게 시킨다 · AI·외부망
3. **검증·해석** — 3대 무기로 · 사람

축 ②

최종 판단은 인간 — 3대 무기

- 🖼️ **시각화** — 그림으로 보면 오류가 보인다
- 🧭 **도메인 지식** — "그럴 리 없다"를 안다
- 👥 **동료 회람** — 여러 눈으로 함께 잡는다

"AI가 분석을 **쉽게** 만들어도, **맞게** 만드는 건 여전히 인간이다."

0 1

Python과 STATA, 무엇이 다른가

전용 소프트웨어 vs 범용 언어 — 그리고 **AI가 바꾼 게임**

STEP 1
한 문장으로 끝내는 차이

STEP 2
7가지 실제 차이

STEP 3
AI가 게임을 바꿨다

한 문장으로 끝내는 차이

오늘 다룰 모든 차이는 이 한 문장에서 파생된다. 외울 건 이것 하나뿐.

STATA

통계를 위해 태어났다

통계 분석 전용 소프트웨어

계량경제학자가 만든 **완성된 도구**. 통계 작업엔 특화돼 간결하다.
단, 그 바깥(웹·자동화·ML)은 약하다.

Python

통계를 (나중에) 배웠다

범용 프로그래밍 언어

통계는 **라이브러리(부품)**를 얹어서 한다. 수집·자동화·AI 무엇이든.
단, 부품을 조립할 줄 알아야 한다.

🔧 **비유** — STATA는 잘 만든 **전동 공구** 한 자루, Python은 그 공구를 만들 수도 있는 **공방** 전체. 못 박을 땐 공구가, 가구를 짤 땐 공방이 필요하다.

같은 회귀, 두 세계

"소득이 높을수록 기대수명이 길까?"(Preston 곡선) — 똑같은 회귀분석을 양쪽으로.

STATA 동사 하나로 결과까지 직행

```
regress life_exp log_gdp
```

통계 **전용 도구**라서 **단 한 줄.** 계수·표준오차·p값·R^2가 출판 수준 표로 바로 나온다.

Python 불러오고·만들고·맞추고·출력

```
import statsmodels.formula.api as smf

model = smf.ols("life_exp ~ log_gdp", data=df).fit()
print(model.summary())
```

범용 언어라 라이브러리를 조립한다. 그만큼 통계 **바깥**도 무엇이든 자유롭다.

→ 그런데 나오는 **계수와 p값은 완전히 똑같다.** 길이가 다를 뿐, 정답은 하나다.

🔗 라이브 데모 **setosa.io/ev/ordinary-least-squares-regression** — 점을 끌면 회귀선과 잔차가 실시간으로 바뀝니다

그 한 문장에서 갈라지는 7가지 차이

근본 차이(전용 SW vs 범용 언어)가 현장에서 어떻게 나타나는가.

차이	STATA · 전용 SW	Python · 범용 언어
데이터 방식	한 데이터셋이 작업공간에 상주	이름 붙은 객체 여러 개 동시 조작
명령 vs 함수	동사형 명령어 — 문장처럼 읽힘	함수·메서드 조합 — 유연하나 조립
통계 출력	출판 수준 표준 표 기본 제공	직접 가공 · 무한 커스터마이즈
가능 범위	통계·계량 안에서 깊다	수집·API·ML·텍스트·배포까지
비용	상업용 유료 라이선스	완전 무료 오픈소스
재현성	.do + .log — 계량 저널 표준	노트북 · 스크립트 + Git
학습곡선 ★	**낮다 — 한 줄로 고급 분석, 입문친화**	**높다 → AI가 무력화**

★ 7번 **학습곡선**을 기억해 두세요 — 잠시 뒤 이 차이를 **무력화**시킵니다.

누가, 어디서 쓰나 — 분야 지도

도구는 진공에 있지 않다. 각자의 '사용자 부족(部族)'이 있다.

STATA의 세계

- 경제학 · 계량경제
- 보건경제 · 역학
- 정책평가 · 임팩트평가(RCT)
- World Bank · 학계 · 국제개발 연구

Python의 세계

- 데이터과학 · 머신러닝
- 테크 · 엔지니어링
- 자동화 · 데이터 파이프라인
- 점점 → 개발협력 · 평가로 확산

능력 차이가 아니라 '생태계 관습'이다 — RCT·임팩트평가도 Python으로 다 된다. 다만 관행상 STATA 보고서가 많고, 수집·대시보드는 Python이 흔할 뿐. **둘 다 읽고 쓰면 현장 전체가 보인다.**

반전 — AI가 게임을 바꿨다

입문자의 가장 큰 장벽은 늘 **문법**이었다. 그런데 문법은 AI가 대신 짠다.

예전 **도구 선택 = "내가 이 문법을 짤 수 있는가?"**

→ 입문자는 쉬운 STATA로 도망쳤다.

↓

지금 **도구 선택 = "이 작업에 어느 생태계가 맞는가?"**

→ AI가 양쪽 문법을 다 짜주니, **일에 맞는 도구를 고른다.**

문법이라는 입장료를 AI가 대신 낸다 → 첫 시간부터 회귀·교차표·패널로 직행. "둘 다 입문이지만 오늘 중급으로 도약한다."

도입 한 장 요약 — 사진 찍어두세요

한 줄로 정리하면 이렇다.

	STATA	Python
한마디	통계 전용 소프트웨어	범용 프로그래밍 언어
잘하는 것	통계 분석에 특화 — 명령이 간결	수집·자동화·ML·배포까지 무엇이든
비용	유료	무료
학습곡선 ★	**낮음**	**높음 → AI가 무력화**
현업 환경	폐쇄망에 설치됨 (실행)	외부망 학습·비교용 (폐쇄망엔 없음)

오늘의 전략 · 외부망에서 AI로 배우고 → **폐쇄망 STATA 코드 꾸러미**로 변환 → 인간이 검증한다

02

데이터 소개 + AI로 첫 분석

분석 전에 **재료**부터 — 데이터를 알아야 결과가 말이 되는지 판단한다.

📂 실습 파일 — notebooks/01_load_clean.ipynb · stata/01_load_clean.do

STEP 1
WDI 데이터 감 잡기

STEP 2
AI 분석 의뢰 프롬프트

STEP 3
불러오기·정제 시연

WDI란 무엇인가

World Bank WDI(World Development Indicators) — 세계은행 대표 개발지표 DB. 보고서의 **'1인당 GDP'·'기대수명'** 출처가 십중팔구 여기다. 오늘은 그 진짜 데이터를 직접 만진다.

200+

개국 커버리지

전 세계 거의 모든 나라

1,486

개 지표

소득·보건·교육·인구·환경

1960~

현재까지

60년+ 시계열 · 공개·무료

📥 직접 받기 `databank.worldbank.org/source/world-development-indicators` · data.worldbank.org — 강의 중 **실제 페이지를 열어** 데이터셋을 보여주세요.

오늘 우리가 쓰는 데이터

WDI에서 핵심 지표만 뽑은 국가×연도 패널 — 한 행 = 한 국가의 한 해.

패널 데이터

217개국

23년 · 2000~2022

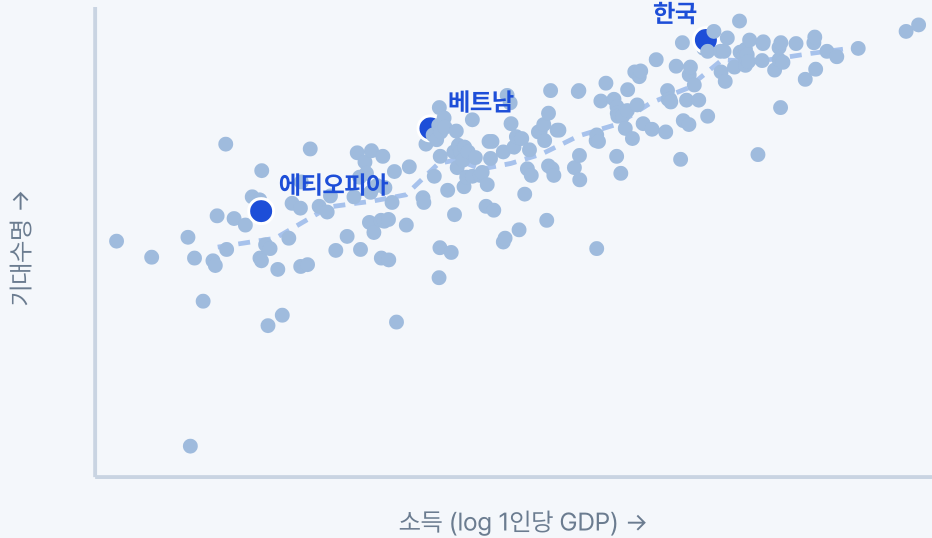
4,991행

같은 나라를 여러 해 추적 → 모듈 4 **고정효과**의 토대.

변수	의미	단위·예시
`economy·name`	국가코드 · 국가명	KOR · Korea
`year`	연도	2000~2022
`region_name`	지역	7개 권역
`income_name`	소득그룹	고/상위중/하위중/저
`gdp_pc` ★	1인당 GDP	현재 US$
`life_exp` ★	기대수명	세(출생 시)
`under5_mort` ★	5세 미만 사망률	1,000명당
`pop` ★	총인구	명
`prim_enroll` ★	초등 취학률	% (100%+ 가능)

개발의 스펙트럼 — Preston 곡선

2021년 210개국. 한 점 = 한 나라. **점에 마우스를 올려보세요.**



📈 **Preston 곡선**

소득과 기대수명의 관계(S. Preston, 1975). 저소득 구간에선 소득이 조금만 늘어도 수명이 **급증**, 고소득에선 **완만**해진다.

🇪🇹 에티오피아	$893 · 65세
🇻🇳 베트남	$3,704 · 74세
🇰🇷 한국	$37,518 · 84세

'진짜 데이터'라서 생기는 것들

합성 데이터엔 없는 현실. 그리고 이게 모듈 2 정제와 모듈 5 검증의 진짜 소재.

결측이 있다

일부 국가·연도에 값이 빠져 있다. **under5_mort 약 10%**, **prim_enroll 약 23%**.

분류에 빈칸도

일부 국가는 소득그룹이 **'Not classified'**. "진짜 데이터는 늘 손이 간다."

어떻게 받았나

World Bank API를 Python **wbgapi**로 수집. → 모듈 4-B에서 **직접** 해본다.

결측·빈칸이 보여도 당황 마세요. **진짜 데이터라서** 그런 거고, 그걸 다루는 게 실무다. **오히려 깨끗하기만 한 데이터가 더 수상하다.**

AI와 대화로 분석을 설계한다

"pandas로 짜줘" 같은 정형 프롬프트는 이제 구식 — AI가 알아듣고, 모호하면 **먼저 되묻는다.**

WDI 데이터로 소득이 기대수명에 주는 영향을 보고 싶어.

1인당 GDP는 왜도가 크니 **로그변환**할까요? 결측 국가는 **제외**할까요?

응, 로그변환하고 결측 제외. STATA랑 Python 둘 다 줘.

네 — 양쪽 코드와 결과 읽는 법, 제가 가정한 것까지 정리했습니다. ✔

예전

정형 프롬프트를 외웠다(역할·데이터·도구…)

지금

목표만 말하면 AI가 되묻고 함께 완성

변하지 않는 핵심 — **무엇을·왜 분석할지(설계)와 결과 검증은 사람. AI는 구현과 되물음을 맡는다.**

★ 가장 중요

폐쇄망 실행용 프롬프트

이 한 가지가 외부망 학습을 현업 실행으로 잇는다.

💬 AI에게 이렇게 시킨다

"이 STATA 코드를 추가 패키지(SSC) 없이 **base STATA만으로, 인터넷 없이** 오프라인에서 돌아가게 짜줘. **경로·변수명은 파일 맨 위에** 모아 한 곳만 바꾸면 되게."

- ✔ SSC 패키지 불필요
- ✔ 인터넷 없이 오프라인
- ✔ STATA 19 MP (우리 환경)

WDI 불러오기·정제를 양쪽으로

로드 → 결측 확인 → 소득 음수 점검 → 로그변수 → 요약. 같은 작업, 두 도구.

Python · pandas

```python
df = pd.read_csv("wdi_panel.csv")
df.info()
df.isna().sum()        # 결측 현황
work = df.dropna(subset=["gdp_pc","life_exp"])
work["log_gdp"] = np.log(work["gdp_pc"])
```

STATA · .do

```
import delimited "wdi_panel.csv", clear
describe
misstable summarize          // 결측 현황
drop if missing(gdp_pc, life_exp)
gen log_gdp = ln(gdp_pc)
```

첫 검증 — 행 수가 맞나? 결측은 어떻게 처리했나? 단위는 맞나? AI가 임의로 한 가정을 들춰낸다.

→ 01_load_clean.do

0 3

같은 분석을 Python과 STATA로

대표 분석 3종을 양쪽으로 — **숫자가 같으면 신뢰, 다르면 하나가 틀린 것.**

📂 실습 파일 — notebooks/02_core_analysis.ipynb · stata/02_crosstab · 03_group_compare · 04_regression.do

분석 ①
기술통계 — 지역×소득 교차표

분석 ②
t검정 · ANOVA

분석 ③
회귀 — Preston 곡선

① 기술통계 — 교차표

데이터를 **요약·묘사**하는 첫걸음(평균·분포·빈도…). 범주형 두 변수의 분포는 **교차표**로 본다.

🔎 **살펴보기 — 각 지역에 어느 소득그룹 나라가 몇 개 있는가? (국가 단위)**

Python

```python
countries = df.drop_duplicates("economy")
# 국가당 1행
pd.crosstab(countries["region_name"],
            countries["income_name"])
```

STATA

```
bysort economy (year): keep if _n==1
// 국가당 1행
tabulate region_name income_name
```

→ **해석** — 저소득국 다수가 **사하라이남**에, 고소득국은 유럽·동아태 등에 분포.

⚠️ 여기까지는 **탐색적 데이터 분석(EDA)** — '분포를 **봤다**'이지 '분석을 **끝냈다**'가 아니다. 검정·인과는 **② t검정 · ③ 회귀**부터.

② 집단 비교 — t검정 · ANOVA

🔎 연구질문 — 사하라이남 vs 그 외(2집단)? · 소득그룹 4개 사이(3집단+)?

Python · scipy

```python
from scipy import stats
# 2집단 — Welch t검정
stats.ttest_ind(ssa, rest, equal_var=False)
# 3집단+ — 일원 ANOVA
stats.f_oneway(*groups)
```

STATA

```
* 2집단 — Welch t검정
ttest life_exp, by(ssa) unequal

* 3집단+ — 일원 ANOVA
oneway life_exp income_n, tabulate
```

두 집단이면 t검정, **세 집단 이상**이면 ANOVA. 분산이 다를 수 있어 t검정은 **Welch(unequal)**를 쓴다.

결과 읽기 — 무엇이 유의한가

p값은 '우연일 가능성'. 0.05보다 작으면 통계적으로 유의하다.

t검정 · 2집단

사하라이남 vs 그 외

두 집단의 기대수명 **평균 차이가 유의**하다. **p < 0.05**

ANOVA · 4집단

F = 1877

F = 집단간 분산 ÷ 집단내 분산

집단 간 차이가 집단 내 노이즈보다 **클수록 F↑** → 차이가 **매우 유의**.

소득그룹별 평균 기대수명 격차

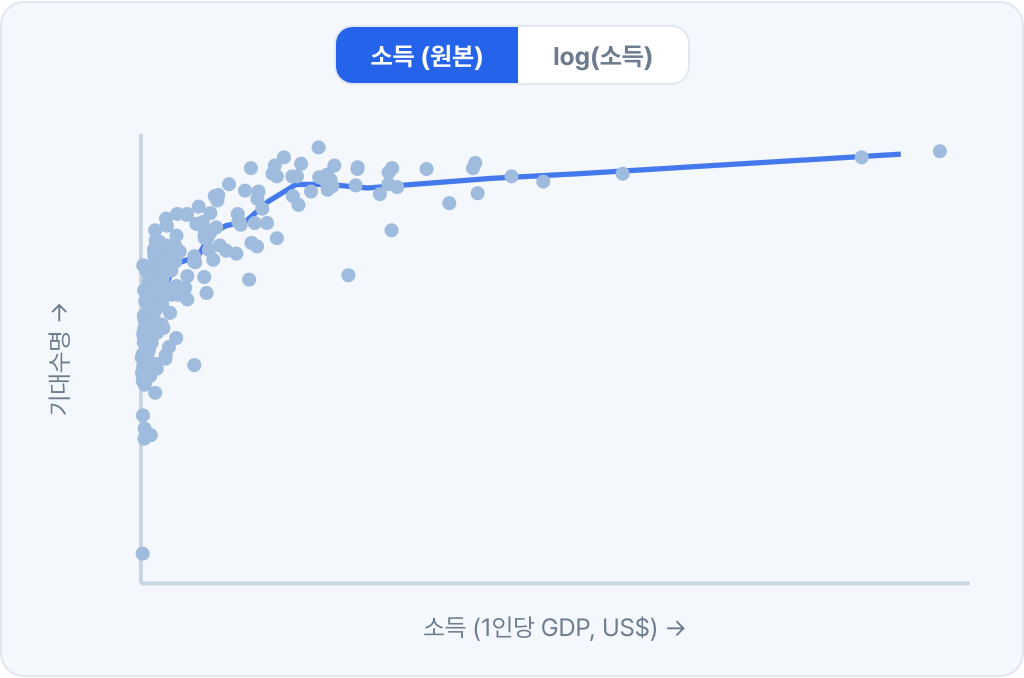
저소득 ~58세 → 고소득 ~78세

🔗 데모 `rpsychologist.com/d3/nhst` — p값·유의성

🔗 데모 `rpsychologist.com/d3/one-way-anova` — F의 출처

③ 회귀 설계 — 왜 로그변환?

선형회귀는 **'직선' 관계**를 가정한다. 휘어 있는 관계를 펴려고 로그를 씌운다.



📈 휜 관계

소득↔기대수명은 저소득서 **급증**, 고소득서 **평평**(Preston 곡선) — 직선이 아니다.

📐 펴기 = 선형화

log(소득)으로 바꾸면 거의 **직선** → 선형회귀에 딱 맞는다.

\+ 분포 쏠림·이상치 영향 완화, % 해석은 **덤**.

AI는 이걸 **자동으로 안 할 수 있다.** "관계가 휘었으니 로그를 씌우자"는 데이터를 아는 사람의 판단 — **설계는 사람 몫.**

③ 회귀 코드와 결과 — Preston 곡선

🔎 연구질문 — 소득(log)이 기대수명을 얼마나 설명하는가?

Python

```
smf.ols("life_exp ~ log_gdp", data=df).fit()
```

STATA

```
regress life_exp log_gdp
```

소득 계수

4.59

양수 · 유의

설명력 R^2

0.68

설명력 높음

→ **해석** — 소득↑ 기대수명↑. **Python·STATA 계수·R^2가 정확히 일치.**

🔗 라이브 데모 **mlu-explain.github.io/linear-regression** — 계수·R^2·잔차를 스크롤하며 단계별로 해부

숫자는 같다. 출력 모양만 다르다.

두 도구의 결과가 같으면 신뢰, 다르면 하나가 틀린 것 — 이것이 **교차검증**이다.

지표	Python	STATA
회귀 계수 (log_gdp)	4.59	4.59
설명력 R^2	0.68	0.68
ANOVA F	1877	1877

✅ 세 지표 모두 일치 → **신뢰할 수 있다.** 도구는 달라도 정답은 하나.

0 4

AI로 고급 분석까지 인과추론 + 머신러닝

비교로 끝내지 않는다 — **진짜 고급 기법 두 가지를 실제로 돌린다.**

📂 실습 파일 — notebooks/03_panel_fe.ipynb · 04_python_strength.ipynb · stata/05_panel_fe.do

4-A · STATA
이원 고정효과 (인과추론)

4-B · Python
수집 + 랜덤포레스트 (ML)

4-C
도구 선택은 환경·작업으로

식별의 문제 — 진짜 효과는?

소득·기대수명은 시간이 지나며 함께 오른다. 단순 회귀는 둘을 **혼동(confounding)** — 소득 계수가 부풀려진다.

🌍 **국가 고유차** — 제도·기후·문화 (시간 불변)

📈 **전세계 시대추세** — 기술·의료 발달 (모든 나라 공통)

❌ **p값으로는 못 거른다** — 계수가 유의해도(p<0.05) 교란이 사라진 게 아니다. p값은 **'우연 여부(정밀도)'**만 볼 뿐 **편향(교란이 섞였나)**은 못 본다. 실제로 **pooled 4.59**도 p≈0(매우 유의)였지만 교란으로 부풀려진 값이었다.

🔍 **고정효과가 남아 있나? — 이렇게 판단한다**

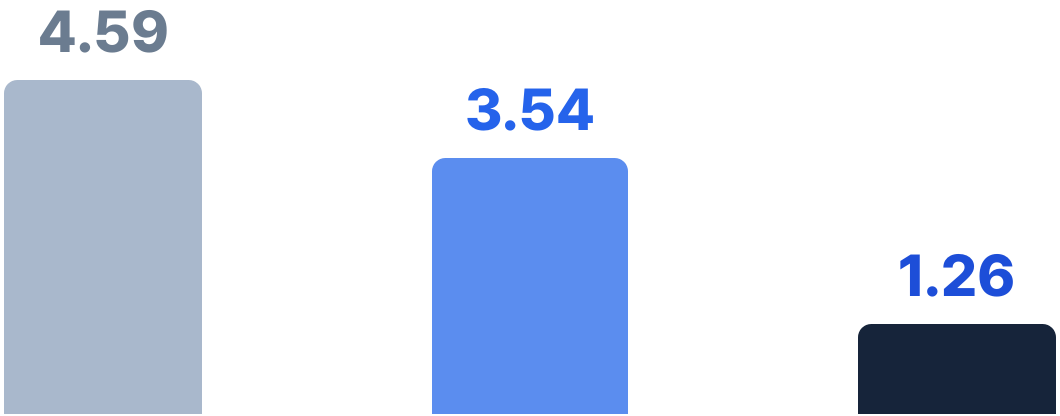
① **모형 비교** — 통제(FE)를 추가했을 때 계수가 **크게 바뀌면** 빠졌던 교란이 있었다는 신호 (다음 장 4.59→3.54→1.26).

② **검정** — **'국가효과 = 0' F검정**(Stata `xtreg, fe`가 자동 출력)·**Hausman 검정**(FE vs 확률효과)으로 고정효과 필요 여부를 통계적으로 확인.

통제할수록 답이 바뀐다

무엇을 통제하느냐에 따라 '소득 계수'가 달라진다 — 교란을 걷을수록 순효과에 근접.

▶ 단계별로 보기



산식 — 각 단계 = "평균을 빼고(within) 회귀". 빼는 평균이 늘수록 교란이 빠져 계수↓

① Pooled
수명 ~ log소득
원자료 그대로 → 4.59

② 국가 FE
(수명−국가평균) ~ (log소득−국가평균)
국가 고유차 제거 → 3.54

③ 이원 FE
국가평균 + 연도평균 모두 차감 후 회귀
시대추세까지 제거 → 1.26

이원 고정효과를 양쪽으로

🔎 국가 + 연도를 동시에 통제하면, 소득의 순효과는?

STATA · base STATA

```
encode economy, gen(country_id)
xtset country_id year
xtreg life_exp log_gdp i.year, ///
      fe vce(cluster country_id)
```

Python · linearmodels

```
PanelOLS.from_formula(
   "life_exp ~ log_gdp"
   "+ EntityEffects + TimeEffects",
   panel).fit()
```

두 도구의 소득 계수가 **정확히 일치** — 도구가 달라도 식별 결과는 하나.

계수 1.262

이원 FE = DiD의 대표 수단

차분-차분(Difference-in-Differences) — 임팩트평가의 표준. 방금 돌린 이원 고정효과가 곧 그 **회귀 구현**이다.

정책효과 = 변화의 차이

처치군의 변화에서 대조군의 변화를 뺀다.

	처치 전	처치 후	변화
처치군	A	B	**B−A**
대조군	C	D	**D−C**

DiD = (B−A) − (D−C)

개체 FE 나라별 **고유차** 통제 — 대조 비교의 역할

시간 FE 공통 **시대추세** 통제 — 전·후 보정

→ 둘을 동시에 통제한 **이원 FE = 일반화된 DiD.**

🎯 KOICA **M&E**(모니터링·평가) 분석과 바로 연결 — 폐쇄망 base STATA로 실행.

AI가 코드를 짜줘도, 어떤 교란을 통제할지(설계)와 결과 검증은 — 사람 몫.

Python의 고급 영역 — 수집 + ML

STATA로는 매우 빈약한 두 가지. 외부망 Python에서.

① 라이브 수집 · wbgapi

World Bank API에서 **실시간 수집·자동화**. 이 강의 데이터 자체가 이렇게 만들어졌다.

```
import wbgapi as wb
wb.data.DataFrame(
   ["NY.GDP.PCAP.CD", "SP.DYN.LE00.IN"],
   ["KOR","VNM","ETH"], time=2021)
```

② 랜덤포레스트 · ML

의사결정나무 **수백 그루의 앙상블** — 데이터·변수를 무작위로 뽑아 학습해 **과적합 ↓**, 비선형·상호작용을 자동 포착.

train/test 분할·교차검증으로 정직하게 평가

RF test R^2 0.96 > OLS 0.82

수집·자동화·ML은 Python의 영역. 단, **"예측이 잘된다 ≠ 인과"** — 도메인 검증 필수.

🔗 라이브 데모 **mlu-explain.github.io/random-forest** — 여러 결정트리가 '숲'으로 합쳐지는 과정을 시각적으로

오늘 못 다룬, 알아두면 좋은 기법

위 둘은 계량 분석, 아래 둘은 머신러닝 — 이름만 말하면 AI가 불러온다. '무엇을 풀지'가 핵심.

인과추론

계량 · 임팩트평가

상관을 넘어 **진짜 효과**를 식별. **오늘 본 이원 FE(DiD)를 넘어** — 도구변수(IV)·회귀불연속(RDD)·성향점수매칭(PSM). 무작위배정 없이 정책효과 추정, KOICA M&E 직결.

주성분분석 (PCA)

계량 · 차원축소

여러 상관된 지표를 **핵심 축 몇 개**로 압축. 개발 복합지수(자산지수 등) 설계에.

↗ 라이브 데모 · setosa

로지스틱 회귀

ML · 분류

예/아니오 이진 결과의 확률을 예측 — **회귀의 분류 버전**. 빈곤 여부·사업 성패·채무불이행 등.

↗ 라이브 데모 · MLU

K-최근접이웃 (K-NN)

ML · 분류·예측

가장 **가까운 이웃 K개**를 보고 판정. 직관적·비모수 — 비슷한 나라로 결측 추정에도.

↗ 라이브 데모 · ML Playground

인과추론·차원축소·분류… 도구는 많다. 호출은 AI가, '무엇을 왜 풀지'와 결과 검증은 사람이.

도구 선택은 '환경·작업'으로

'무료냐 유료냐'가 아니라, '어디서 돌려야 하고 무슨 일이냐'로 고른다.

이럴 땐	도구	이유
폐쇄망에서 바로 실행해야 하는 현업 분석	STATA	현재 우리 폐쇄망 실행 환경
수집·API·자동화·ML·텍스트·시각화·배포	Python	압도적 생태계, 외부망에서
모르겠다	둘 다	AI에게 둘 다 시켜 **교차검증**

05

인간의 검증력 — 3대 무기

AI가 난이도를 낮춰도, **검증은 인간이 더 잘한다.** 여러분에겐 무기가 있다.

📂 실습 파일 — `notebooks/05_human_verification.ipynb`

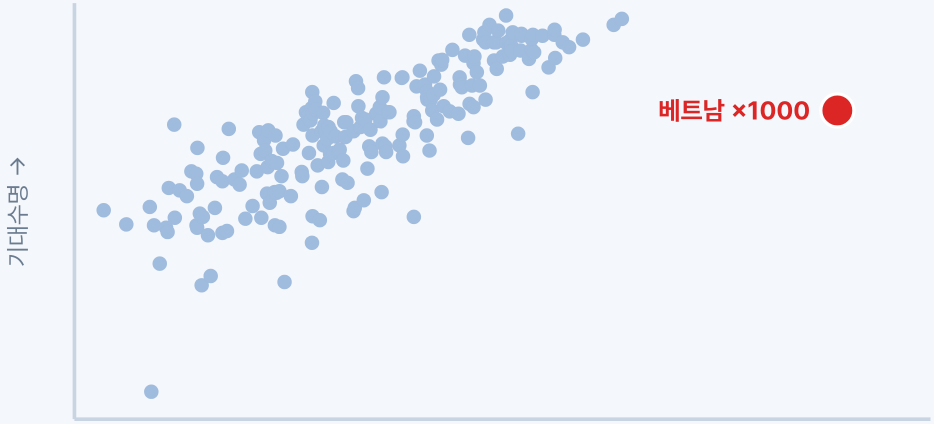
🖼️ 시각화 — 그려서 본다

🧭 도메인 지식 — 현장을 안다

👥 동료 회람 — 함께 본다

한 데이터, 여러 그림 — 각도를 바꾸면 보인다

같은 오류 데이터(베트남 GDP ×1,000)를 세 방식으로. **어느 그림에서든 튄다.**



📍 산점도 | 📊 히스토그램 | 📦 박스플롯

📍 **산점도** — 추세에서 벗어나 오른쪽으로 튄 점

📊 **히스토그램** — 분포에서 동떨어진 외딴 막대

📦 **박스플롯** — 수염 밖 유일한 이상치(outlier)

한 각도로 안 보여도, 여러 그림으로 그리면 같은 이상치가 거듭 드러난다.

🔗 라이브 데모 `research.autodesk.com · Datasaurus` — 평균·상관이 같아도 그림은 천차만별

나머지 두 무기 — 도메인 · 회람

AI가 못 가진, 사람만의 강점.

🧭 도메인 지식 — 현장을 안다

"베트남이 그 해 세계 최고 소득? 말이 안 된다."

AI에겐 현장의 사실(ground truth)이 없다. **여러분에겐 있다.**

👥 동료 회람 — 함께 본다

1장 요약을 동료에게 회람 → "이 수치 단위가 이상한데요?"

교차검증 — Python·STATA 숫자가 일치하나? (다르면 하나가 틀린 것)

한 사람(혹은 AI)이 놓친 걸 **여러 눈이 잡는다.** 검증을 통과한 결과만 의사결정 언어로 보고한다.

인간의 3대 무기 — 종합

AI는 빠르지만 보지 못하고, 맥락이 없고, 혼자 일한다.

무기	AI의 약점	인간의 강점
🖼️ **시각화**	숫자만 보고 이상치를 놓침	**그림으로 보면 오류가 보인다**
🧭 **도메인 지식**	현장의 사실(ground truth)을 모름	**"그럴 리 없다"를 안다**
👥 **동료 회람**	혼자, 진공 상태에서 답함	**여러 눈으로 함께 잡는다**

AI로 난이도를 낮추고, 이 세 무기로 신뢰도를 높인다.

AI로 난이도를 낮추고, 인간이 검증한다.

① 설계
무엇을·왜 — 사람

② 생성
코드 — AI·외부망

③ 검증·해석
3대 무기 — 인간

📦 코드 꾸러미 (완성)

github.com/amnotyoung/oda-ai-stats

Colab 노트북 5 · STATA .do 6 · WDI 데이터 · 핸드아웃 — 폐쇄망엔 stata/ + data/ 반입

다음 학습 경로

내 업무 질문 1개 → 외부망 AI로 설계 → STATA 코드로 변환 → **폐쇄망 실행**

감사합니다 · Q&A

분석, 그다음 — 곱씹어 볼 세 가지

기법은 도구일 뿐이다. 강의를 마치며, 현업에서 생각해 볼 질문.

1 엄정한 분석은 모두를 기쁘게 할까?

고정효과를 걷어내면 **우리 사업 효과가 작게** 보일 수 있다. '우리 덕분'이 **인과가 아니라 상관**일 뿐임이 드러나면, 상급자·기관·시민은 **반길까?**

2 좋은 분석은 어떻게 전달되는가?

상대가 **이해하지 못하면** 의사결정은 안 바뀐다. **커뮤니케이션**이 막히면 — 애써 얻은 분석의 가치는 **뚝 떨어진다.**

3 복잡한 분석·코드, 어떻게 관리할까?

챙길 건 많고 코드는 복잡하다. 버전·재현·협업을 어떻게 챙기지?

→ 버전관리, GitHub를 쓰자!

오늘 받은 코드 꾸러미가 바로 그 예.

분석은 끝이 아니라 시작이다 — 도구(AI·GitHub)는 거들고, 판단·소통·책임은 사람 몫.

모니터링 vs 영향평가 — 분석의 깊이가 다르다

같은 데이터라도, 답하려는 **질문**에 따라 요구되는 분석의 깊이가 다르다.

모니터링 · 예: UN MDG 2015

질문: "무엇이 **변했나?**"

분석: 표·그래프·% 변화 **(기술통계)**

예) 극빈율 47%→14%, 아동사망 90→43

인과는 다루지 않음 — 변화를 **묘사**

영향평가 · 예: KOICA 메콩

질문: "우리 사업 **때문에** 변했나?"

분석: 회귀→고정효과→ESR, **처치효과(ATT)**

교란 통제 + 반사실(대조군) 비교

인과를 추정 — 효과를 **귀속**

분석 깊이 →

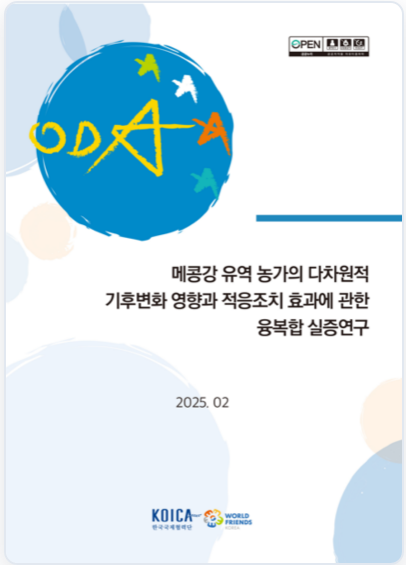
기술통계 | t검정·ANOVA | 회귀 | 고정효과 | 인과추론(ESR·ATT)

└ 모니터링은 여기까지

영향평가는 끝까지 간다 ┘

오늘 배운 걸로 — 이 보고서가 읽힌다

어려워 보여도 — 이 보고서에 쓰인 방법은 **전부 오늘 배운 것**이다.



KOICA 메콩강 농가 기후변화
영향평가 (2025, 서울대)

보고서가 쓴 방법	= 오늘 배운 것
다중회귀분석 (OLS)	계수·R^2로 읽은 회귀
패널 고정효과 (Panel FE)	4.59 → 3.54 → 1.26 계수 사다리
주성분분석 (PCA)	여러 지표를 한 '지수'로
내생성 통제 (ESR)	'상관 ≠ 인과' · 편향 잡기
패널 자료 · 집단 비교	패널 데이터 · t검정/ANOVA

용어만 길지, 전부 아는 개념이다. 이제 실제 표를 읽어보자. →

그 표를 오늘 배운 걸로 해석하면

〈표 4-1c〉 베트남 적응조치의 효과 (내생적교체회귀 ESR) — 실제 보고서의 표 전체.

〈표 4-1c〉 (베트남) 분류화된 기후변화 적응조치의 효과분석 (내생적교체회귀 분석)

베트남 (내생적교체회귀 분석)	1인당 소득 (로그)		1인당 농업소득 (로그)		1인당 소비지출 (로그)		1인당 식품지출 (로그)		현재 삶에 대한 만족도		미래 삶에 대한 낙관도	
	ATT	ATU	ATT	ATU	ATT	ATU	ATT	ATU	ATT	ATU	ATT	ATU
품종/작물 다각화	1.213*** (0.054)	-0.442*** (0.046)	1.504*** (0.060)	0.342*** (0.051)	0.470*** (0.030)	0.868*** (0.025)	0.187*** (0.023)	-0.377*** (0.024)	-0.099*** (0.008)	-0.233*** (0.006)	-0.002 (0.008)	0.049*** (0.006)
농법 조정	0.364*** (0.038)	-0.266*** (0.040)	0.576*** (0.044)	0.628*** (0.047)	0.124*** (0.021)	0.651*** (0.021)	-0.454*** (0.018)	-0.251*** (0.020)	-0.062*** (0.005)	-0.365*** (0.007)	0.087*** (0.005)	0.030*** (0.006)
물 사용관리/시설개선	0.300*** (0.047)	0.191*** (0.033)	0.372*** (0.057)	0.687*** (0.039)	0.175*** (0.021)	0.717*** (0.021)	-0.650*** (0.020)	-0.037* (0.020)	0.015*** (0.005)	-0.270*** (0.006)	0.073*** (0.006)	-0.184*** (0.005)
소득 다각화	0.529*** (0.051)	-0.292*** (0.042)	1.656*** (0.069)	0.491*** (0.046)	0.242*** (0.025)	0.755*** (0.024)	0.043** (0.021)	-0.054** (0.024)	0.132*** (0.007)	-0.260*** (0.007)	0.015* (0.008)	0.020*** (0.006)
기후정보 활용	0.360*** (0.041)	0.139*** (0.038)	0.496*** (0.045)	0.718*** (0.045)	0.148*** (0.120)	0.423*** (0.025)	0.033** (0.017)	-0.306*** (0.022)	-0.089*** (0.005)	-0.265*** (0.006)	-0.007 (0.006)	0.013** (0.005)
보험 구입	-2.34*** (0.072)	0.903*** (0.046)	-1.619*** (0.074)	0.654*** (0.047)	0.304*** (0.058)	0.057** (0.025)	-1.455*** (0.031)	0.712*** (0.025)	0.122*** (0.009)	-0.216*** (0.007)	-0.056*** (0.009)	-0.211*** (0.006)

Heteroscedasticity robust standard errors in parentheses. Control variables are included.

* p < 0.10, ** p < 0.05, *** p < 0.01

소득 다각화 → 소득 +0.529* **: 한 농가가 안 한 농가보다 소득이 **유의하게 ↑**(★★★=p<0.01). **ATT=처치효과**라, 배운 대로 **'상관'이 아니라 '적응조치 덕분'**이라는 **인과**로 읽는다.

보험 구입 → 소득 −2.34* **: **음수·유의** → 오히려 소득을 **낮춘다**. '효과 없음'이 아니라 **역효과** — **부호와 별표를 함께** 읽어야 안 속는다.