AI 활용사례

AI에게 배우고, AI로 가르치다 — 동아리 EDA 강의 준비기

0. 어떤 상황에서 AI를 활용했나

데이터 분석 동아리 TOBIG's에서 신입 부원들에게 EDA(탐색적 데이터 분석)를 가르치는 강의를 맡게 됐다.

처음엔 "아는 걸 정리해서 슬라이드로 옮기면 되겠지"라고 생각했다. 하지만 막상 준비하다 보니 "설명할 수 있는 것"과 "정말 아는 것" 사이에 간극이 있다는 걸 깨달았다. Simpson's Paradox 같은 개념을 슬라이드에 넣으려는데, 처음 배우는 사람 입장에서 왜 이런 현상이 생기는지 직관적으로 설명할 자신이 없었던 것이다.

그래서 이번 강의 준비를 하나의 실험으로 삼기로 했다. AI를 "슬라이드 만들어주는 도구"로만 쓰지 않고, 두 단계 모두에 서로 다른 방식으로 활용해보기로 했다.

- 내가 먼저 확실히 이해하는 **자기주도학습 단계**
- 그 이해를 학생에게 전달하는 **수업준비 단계**

배우는 사람과 가르치는 사람의 입장이 다르면 AI에게 요구해야 하는 것도 달라진다는 걸 이번에 알게 됐다. 배울 때는 "이게 정말 맞는지 검증해줄 도구"가 필요했고, 가르칠 때는 "내가 이해한 걸 학생이 체감하게 만들 장치"가 필요했다.

이 경험이 비단 동아리 강의뿐 아니라, 스스로 공부하면서 동시에 누군가를 가르쳐야 하는 모든 사람(과외 선생님, 조교, 사내 스터디 리더 등)에게도 적용 가능하다고 생각해 이 사례를 공유한다.

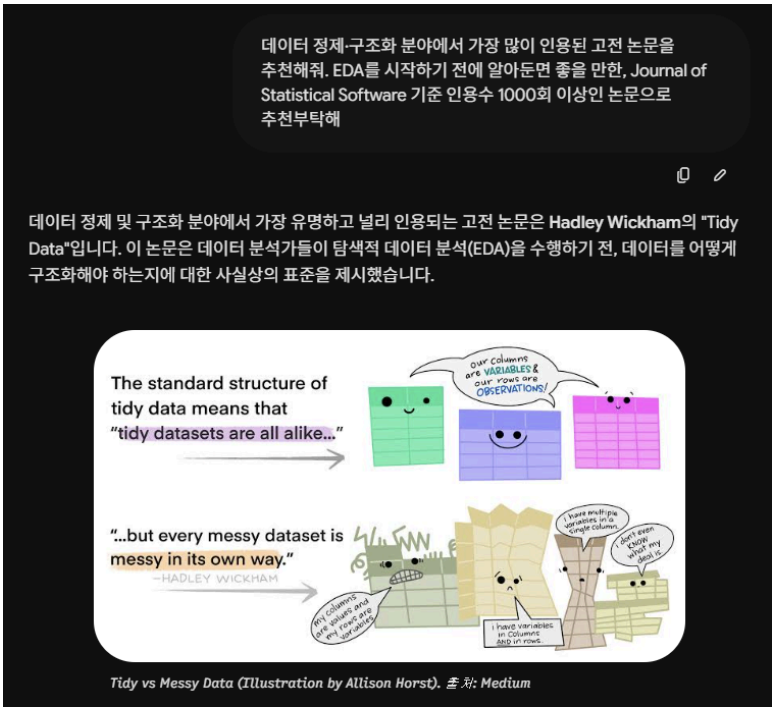
강의를 준비하며 실제로 거친 다섯 단계 — **탐색, 검증, 이미지 제작, 적용, 자가진단** — 를 순서대로 기록했다. 각 단계마다 어떤 AI를 왜 선택했는지, 그리고 AI가 준 결과를 그냥 쓰지 않고 어떻게 검증했는지를 중심으로 풀어보려 한다.

1. 탐색 — Moonlight AI로 개념의 뿌리 찾기

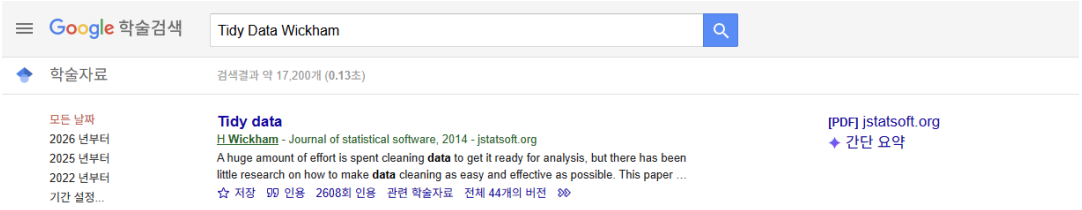
[자기주도학습]

강의를 준비하며 가장 먼저 든 생각은, EDA를 가르치기 전에 "데이터가 애초에 어떤 구조를 갖춰야 분석하기 좋은가"라는 더 근본적인 개념부터 짚고 싶다는 것이었다.

그래서 바로 자료를 뒤지는 대신, Gemini에게 인용수와 저널 조건을 명시적으로 걸고 "데이터 정제·구조화 분야에서 가장 많이 인용된 고전 논문"을 추천해달라고 물었다.



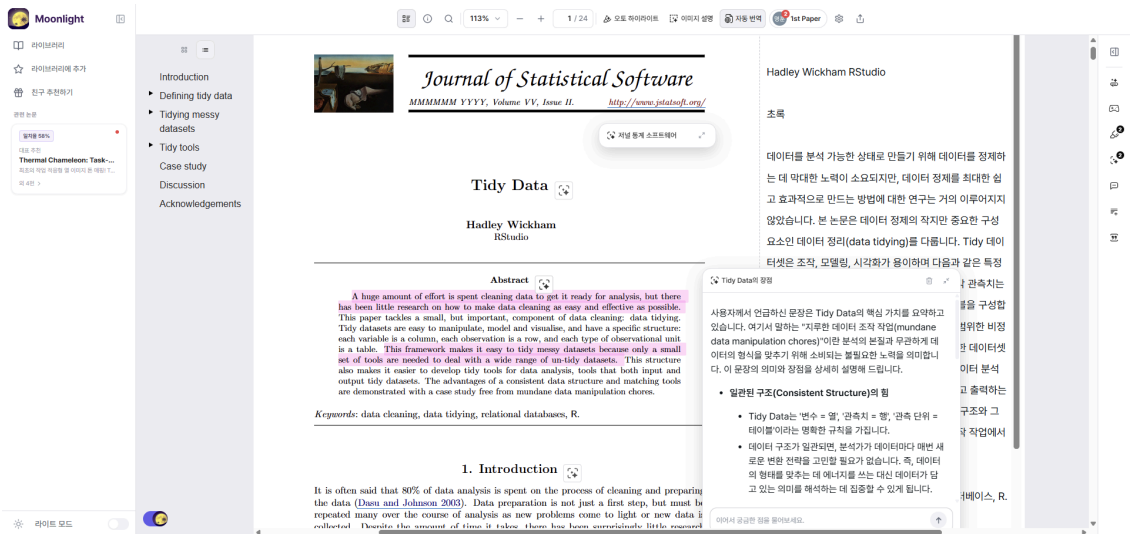
추천을 그대로 믿지 않고, 실제로 그만큼 인용됐는지부터 확인했다. 구글 스칼라에서 논문명을 직접 검색해 "Cited by" 수치를 눈으로 확인한 뒤에야 이 논문을 읽어볼 가치가 있다고 판단했다.



논문을 확인한 뒤에는 Moonlight AI로 원문을 열어 초록을 번역·확인했다.

여기서도 초록만 훑고 넘어가지 않았다. "tidy datasets are easy to manipulate, model and visualise... free from mundane data manipulation chores"라는 문장에서 "mundane data manipulation chores"가 정확히 어떤 의미인지 AI에게 다시 캐물었다.

단순히 "귀찮은 작업" 정도로 흘려 읽을 수도 있었지만, 실제로는 "데이터 형식을 맞추는 데 드는 불필요한 노력"이라는 구체적인 의미가 있었다. 이걸 이해하고 나니 Tidy Data의 핵심 주장(일관된 구조를 갖추면 분석가가 매번 새로운 변환 전략을 고민할 필요가 없어진다는 것)이 훨씬 선명하게 들어왔다.



이 과정에서 중요했던 건 두 가지였다.

- AI가 추천한 자료를 그대로 신뢰하지 않고 인용수로 재확인한 것
- 초록을 훑는 데서 그치지 않고 모호하게 넘어갈 수 있는 표현을 붙잡고 더 깊이 파고든 것

자기주도학습에서 AI의 역할은 "답을 알려주는 존재"가 아니라 "내가 어디까지 아는지 스스로 시험하게 만드는 상대"에 가까웠다.

---

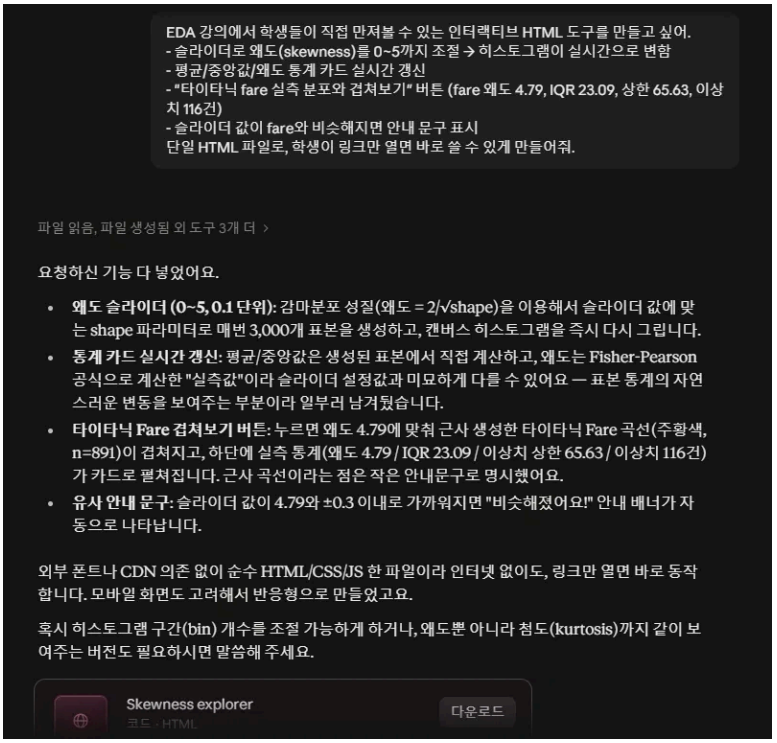
2. 검증 — 직접 움직여보며 확인하는 인터랙티브 도구

[자기주도학습 → 수업준비로 확장]

논문을 읽고 "이해했다"고 느끼는 것과 "정말 이해했다"는 건 다르다.

그래서 이번엔 나 혼자 확인하고 끝내는 게 아니라, 학생들이 강의 시간에 직접 슬라이더를 움직여보며 왜도(skewness)와 분포 모양의 관계를 스스로 체감할 수 있는 실습 도구를 만들기로 했다. 나를 위한 검증 도구로 시작한 게, 학생을 위한 학습 도구로 자연스럽게 확장된 지점이었다.

AI에게 결과물만 받아 쓰는 게 아니라, 내가 원하는 기능을 프롬프트로 구체적으로 명시하고, 나온 결과를 직접 확인하는 과정 자체를 기록했다.



요청한 것은 세 가지였다.

- 왜도 슬라이더를 움직이면 히스토그램이 실시간으로 바뀔 것
- 평균·중앙값·왜도를 통계 카드로 실시간 갱신할 것
- 타이타닉 fare 실측 분포(왜도 4.79, IQR 23.09, 상한 65.63, 이상치 116건)와 겹쳐볼 수 있는 버튼을 넣을 것

특히 눈에 띄었던 건 AI가 왜도 계산에 Fisher-Pearson 공식을 써서 "슬라이더로 설정한 값"이 아니라 "생성된 3,000개 표본에서 실제로 계산한 값"을 보여주도록 만들었다는 점이었다. 슬라이더 값과 계산된 왜도가 미묘하게 다른 게 오히려 표본 통계의 자연스러운 변동을 보여주는 장치였다.

외부 CDN 없이 단일 HTML 파일로 만들어달라고 한 것도, 학생들이 인터넷 환경과 무관하게 링크 하나로 바로 실습할 수 있게 하려는 의도였다.

[실습영상 mp4 따로 첨부]

AI가 만들어준 도구를 그냥 쓴 게 아니라, 학생 입장에서 직접 조작해보며 의도한 대로 작동하는지(안내 문구가 정확한 순간에 뜨는지, 통계 카드 숫자가 실제로 갱신되는지)를 확인한 뒤에야 강의 자료로 채택했다.

이 인터랙티브 도구는 "내가 이해했는지 스스로 시험하는 도구"이자 동시에 "학생들이 개념을 손으로 만지며 체득하는 실습 도구"였다. 하나의 프롬프팅 결과물이 자기주도학습의 검증 수단과 수업의 실습 자료, 두 역할을 동시에 해낸 셈이다.

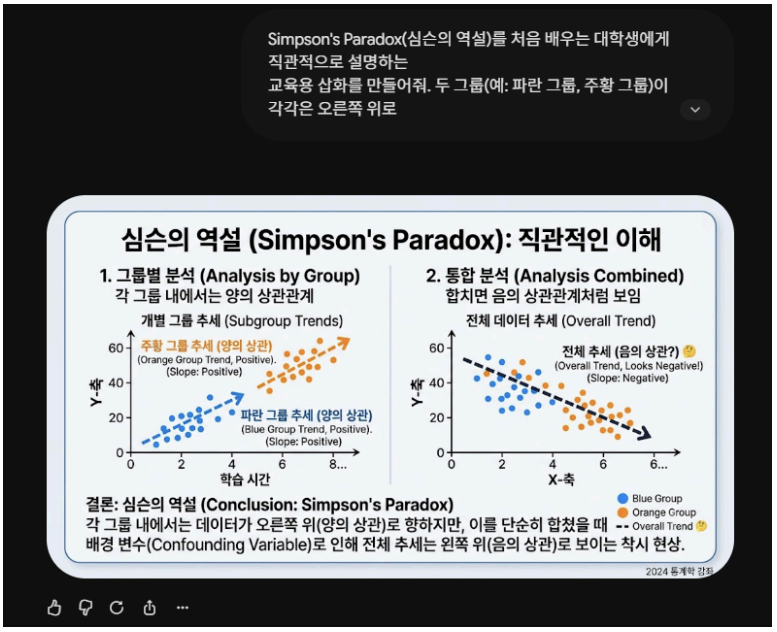
---

3. 이미지 제작 — 개념을 직관적으로 보여줄 삽화 만들기

[수업준비]

앞서 "탐색" 단계에서 Simpson's Paradox 같은 개념을 슬라이드에 넣으려는데, 왜 이런 현상이 생기는지 직관적으로 설명할 자신이 없었다고 했었다.

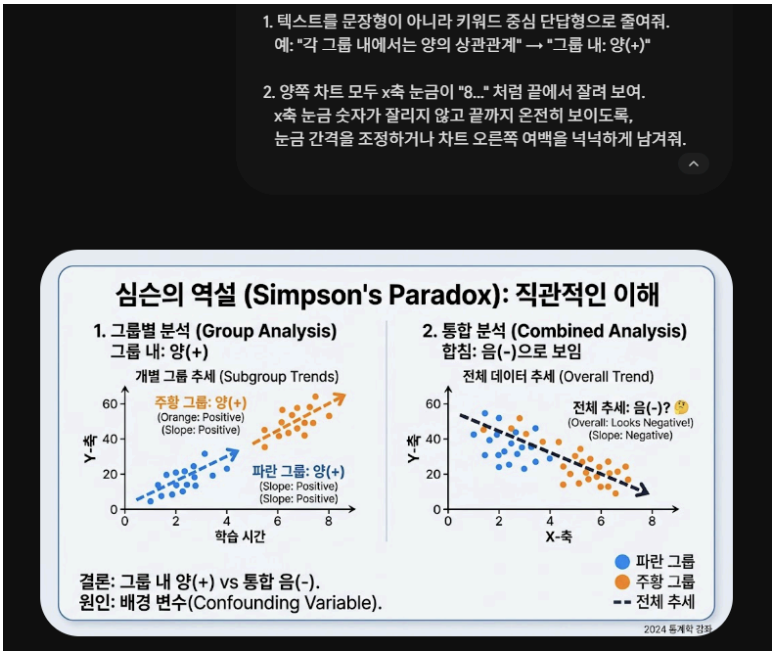
표나 산점도만으로는 "그룹 내에서는 양의 상관인데 합치면 왜 음의 상관처럼 보이는가"라는 착시를 한눈에 전달하기 어려웠다. 그래서 이 개념 하나를 위한 전용 삽화를 만들기로 했다.



처음 나온 이미지는 개념 구조 자체는 정확했다(그룹별 추세는 양의 상관, 합치면 음의 상관처럼 보이는 구도). 하지만 두 가지가 아쉬웠다.

- 설명 텍스트가 문장형이라 슬라이드에 그대로 쓰기엔 너무 빽빽했다
- 양쪽 차트 모두 x축 눈금이 "8..."처럼 끝에서 잘려 보였다

발표 슬라이드는 말로 풀 부분과 화면에 남길 부분이 구분돼야 하는데, 초안은 그 구분이 없었고 잘린 눈금은 완성도를 떨어뜨렸다.



수정본은 "그룹 내: 양(+) / 통합: 음(-)으로 보임 / 원인: 배경 변수"처럼 키워드로 정리되면서, 발표할 때 이 키워드를 짚어가며 설명하면 되는 구조로 바뀌었다. x축 눈금도 0부터 8까지 잘림 없이 온전히 보이게 고쳐졌다.

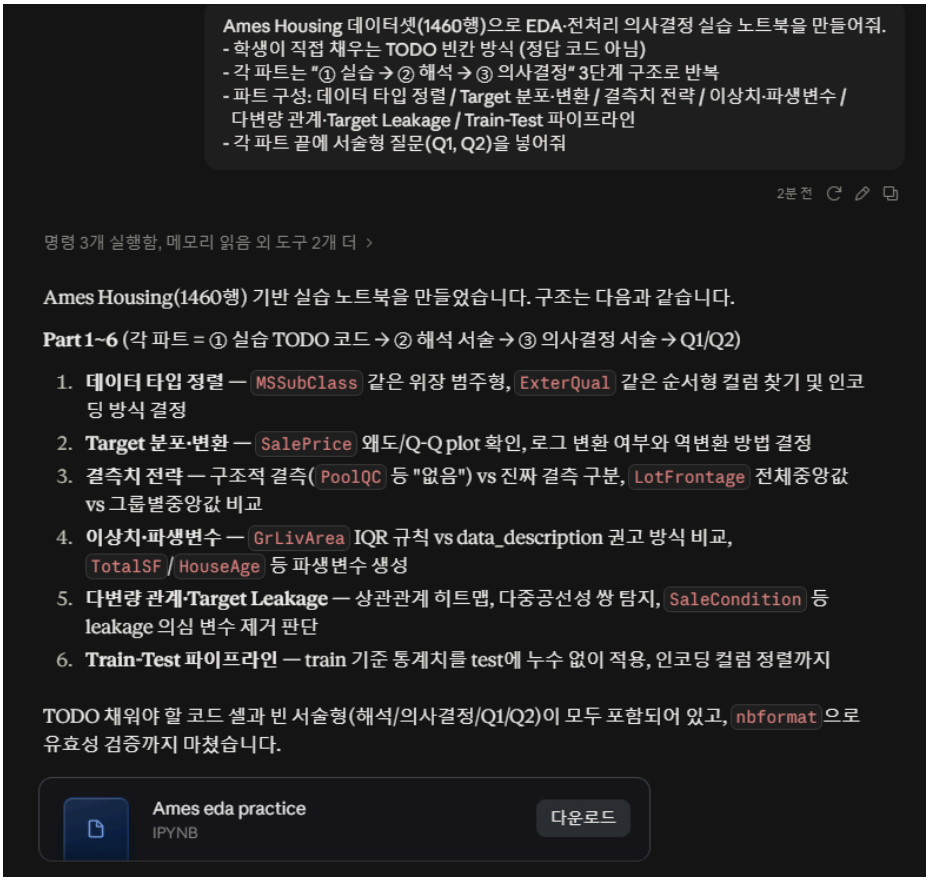
단순히 "예쁜 그림을 넣었다"가 아니라, 학생들이 그래프만으로는 놓치기 쉬운 "합쳐서 보면 방향이 뒤집힌다"는 반직관적인 지점을, 말로 설명하기 전에 시각적으로 먼저 각인시키는 것이 목적이었다. 교사 입장에서 AI 이미지 생성 도구는 "내가 말로 설명하기 힘든 순간을 대신 그려주는 조교" 역할을 했다.

---

4. 적용 — 생성 AI와 검증 AI를 분리해서 실습 코드 만들기

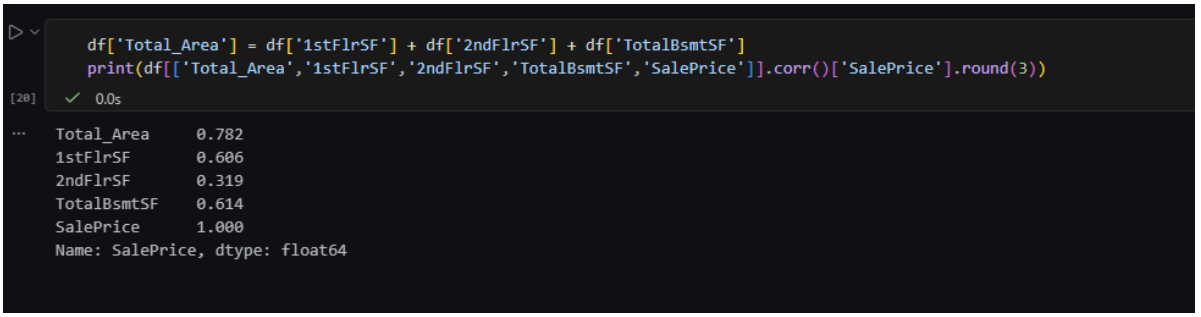
[수업준비]

이해한 개념을 학생이 직접 풀어볼 수 있는 실습 노트북으로 만들 차례였다. Claude Code에 데이터 구조부터 문제 형식까지 구체적으로 명시해 초안을 요청했다.



TODO 빈칸과 서술형 질문까지 포함된 노트북이 나왔다.

학생 입장에서 직접 정답을 채워보며 검증하는 과정에서 Part 4(이상치·파생변수)를 풀다가 걸리는 지점이 나왔다. `Total_Area`라는 파생변수를 `1stFlrSF + 2ndFlrSF + TotalBsmtSF`로 만들고 상관관계를 확인해봤는데, 원본 변수들과의 관계가 심상치 않았다.

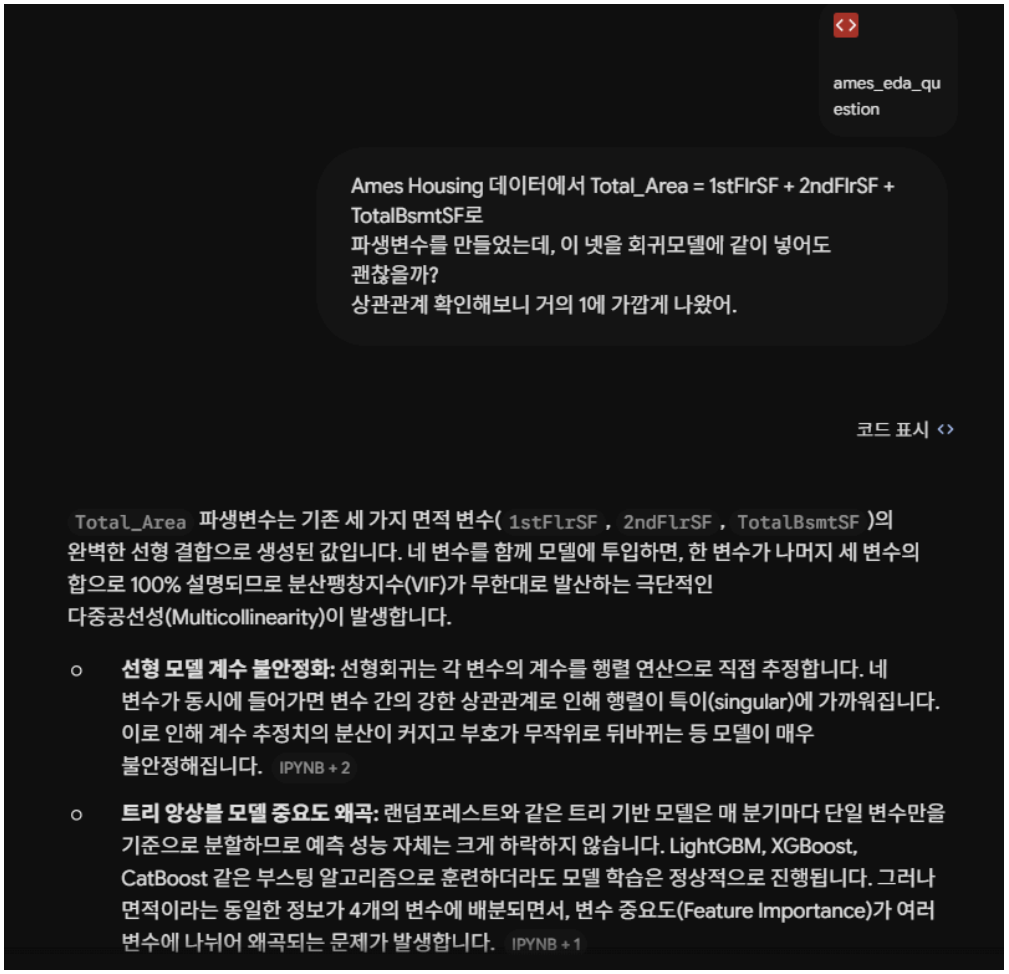


```
df['Total_Area'] = df['1stFlrSF'] + df['2ndFlrSF'] + df['TotalBsmtSF']
print(df[['Total_Area','1stFlrSF','2ndFlrSF','TotalBsmtSF','SalePrice']].corr()['SalePrice'].round(3))
```

[20] ✓ 0.0s

```
Total_Area     0.782
1stFlrSF       0.606
2ndFlrSF       0.319
TotalBsmtSF    0.614
SalePrice      1.000
Name: SalePrice, dtype: float64
```

이걸 그냥 넘기지 않고 다른 AI에게 검증을 요청했다. Claude Code로 만든 노트북이었으니, 검토는 의도적으로 다른 모델에 맡겼다.



AI의 설명을 말로만 받아들이지 않고 직접 VIF를 계산해 재현했다.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
import pandas as pd

X = df[['Total_Area','1stFlrSF','2ndFlrSF','TotalBsmtSF']].dropna()
vif = pd.DataFrame()
vif['feature'] = X.columns
vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
print(vif)
```

```
[21] ✓ 0.1s
...        feature  VIF
0   Total_Area  inf
1     1stFlrSF  inf
2     2ndFlrSF  inf
3  TotalBsmtSF  inf
c:\Users\82104\AppData\Local\Programs\Python\Python313\Lib\site-packages\statsmodels\stats\outliers_influence.py:197: RuntimeWarning: divide by zero encountered in scalar divide
  vif = 1. / (1. - r_squared_i)
```

VIF가 무한대로 나온 건 계산 오류가 아니었다. 파생변수가 원본 세 변수의 합 그 자체이기 때문에 발생하는 완벽한 다중공선성을, 숫자가 정확히 보여준 것이었다.

이 발견을 조용히 코드만 고쳐서 넘어가지 않고, 노트북 Part 4의 서술형 질문으로 그대로 옮겼다. "파생 변수와 원본 3개를 함께 넣으면 어떤 문제가 생기는지" 학생이 스스로 판단하게 만든 것이다.

한 AI가 만든 결과물을 다른 AI로 검증하고, 그 과정에서 발견한 진짜 문제를 다시 학생을 위한 질문으로 재구성하는 것. 교사가 실습 자료를 준비할 때 AI를 검증 없이 그대로 쓰면 오류를 그대로 학생에게 전달하게 되지만, 교차검증을 거치면 그 발견 자체가 더 좋은 문제로 바뀔 수 있다는 걸 확인한 단계였다.

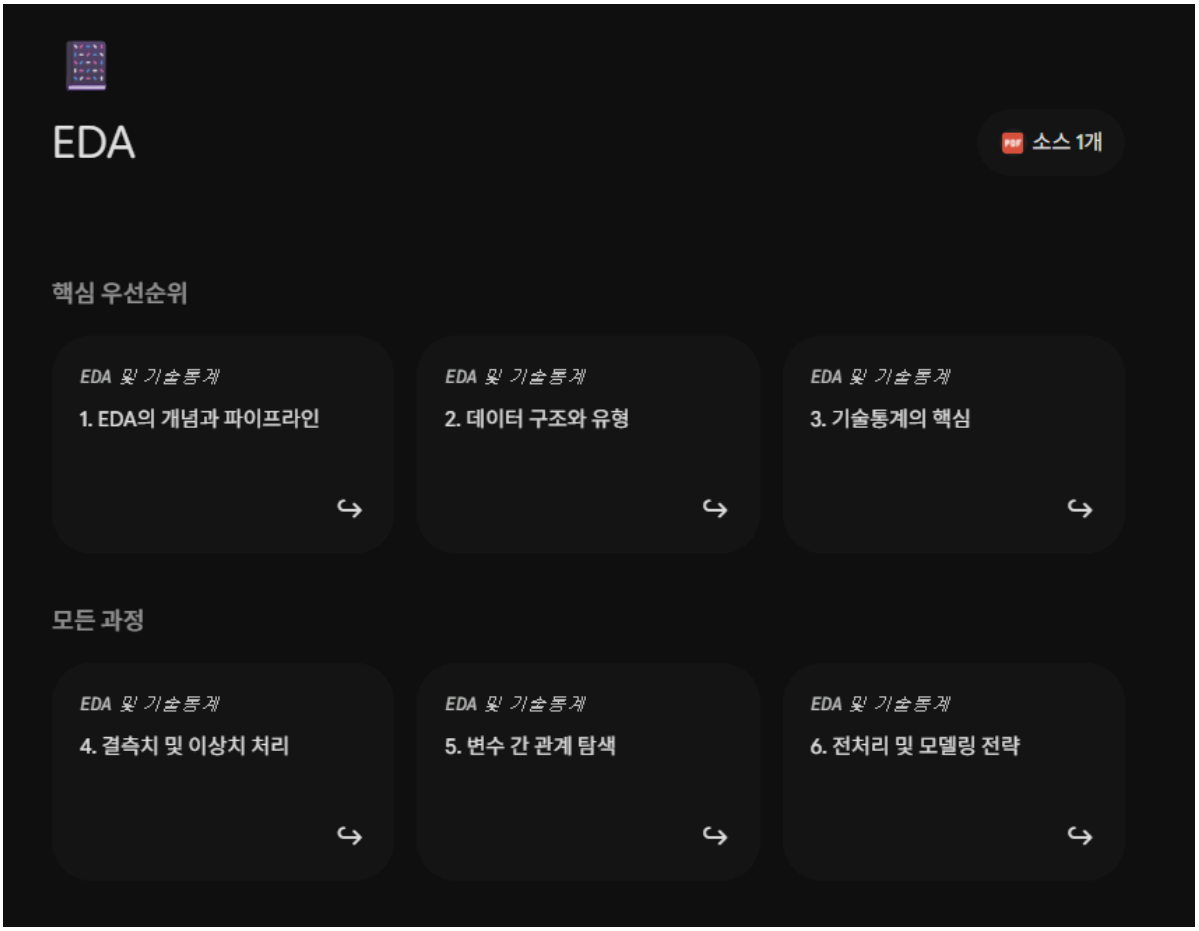
5. 자가진단 — Gemini NotebookLM으로 퀴즈 만들기

[수업준비]

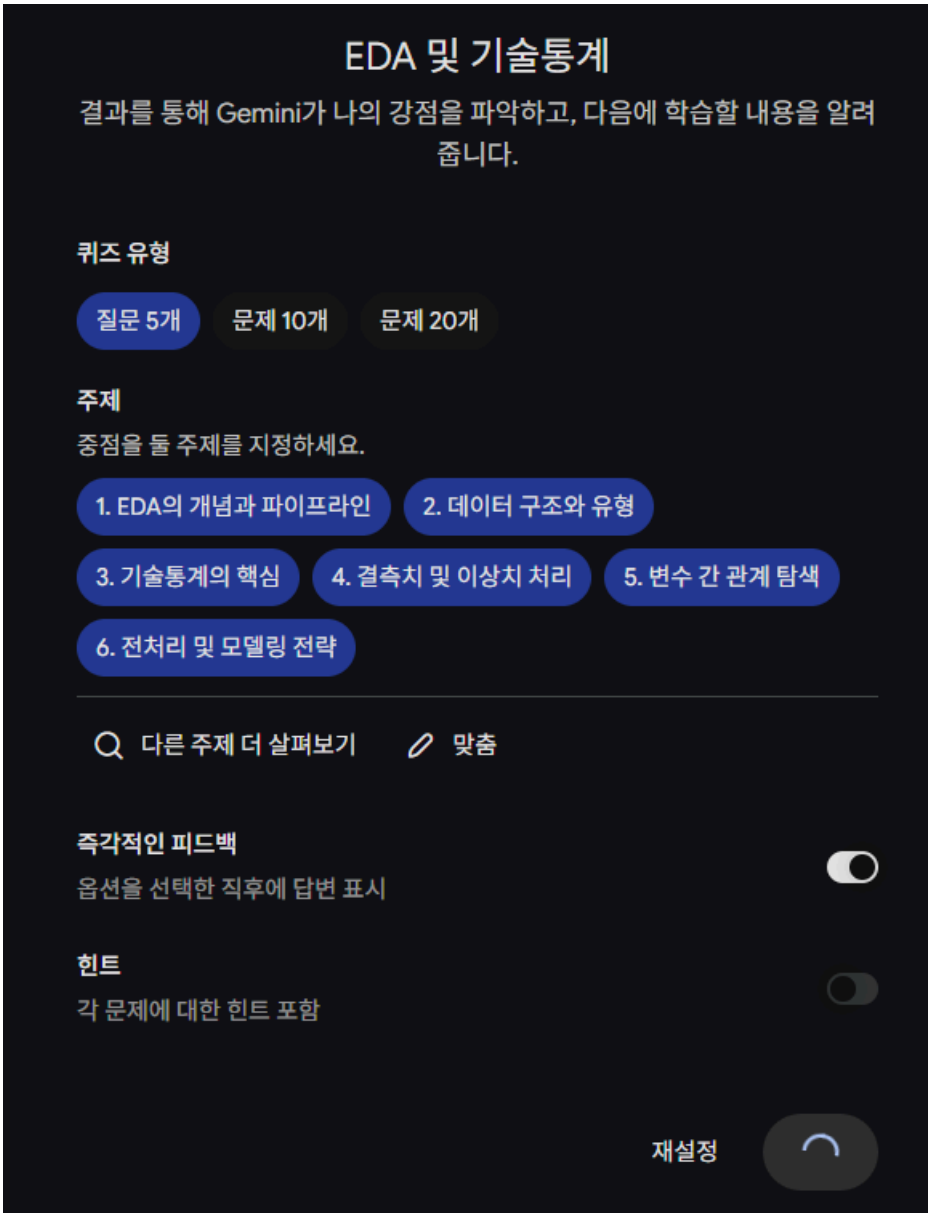
강의자료 초안이 다 나온 뒤, 이걸 다시 한번 AI에게 "학생의 시선"으로 읽혀보고 싶었다.

완성된 PDF를 Gemini NotebookLM에 소스로 올리자, 60장 가까운 슬라이드를 사람이 다시 훑지 않아도 AI가 알아서 6개 토픽으로 우선순위까지 매겨 정리해줬다.

- EDA의 개념과 파이프라인
- 데이터 구조와 유형
- 기술통계의 핵심
- 결측치 및 이상치 처리
- 변수 간 관계 탐색
- 전처리 및 모델링 전략



여기서 흥미로웠던 건, 이 요약이 단순히 "정리"에서 끝나지 않고 바로 퀴즈 생성으로 이어진다는 점이었다. 문항 수(5/10/20개)와 중점 토픽을 직접 고를 수 있어서, 강의 전체를 아우르는 종합 퀴즈 대신 "결측치 및 이상치 처리"처럼 헷갈리기 쉬운 파트만 골라 집중적으로 문제를 뽑아낼 수도 있었다.



이렇게 뽑은 초안 문항들을 그대로 쓰지 않고, 강의 의도에 맞게 다듬어서 최종 퀴즈로 다시 정리했다.

객관식 문항은 "알맞지 않은 것을 모두 고르세요" 형식으로 만들어, 그럴듯해 보이지만 틀린 문장(예: "결측치 비율이 77.2%로 높아도 무조건 평균으로 대체하는 게 옳다")을 일부러 섞어 넣었다. 정답을 아는지가 아니라, 강의에서 강조했던 원칙("결측 원인에 따라 처방이 다르다")을 실제로 체화했는지 가려내기 위한 장치였다.

객관식

Q. 다음은 이번 강의 전체 내용을 요약한 설명들입니다. 이 중 **알맞지 않은 것을 모두 고르세요.**

1. EDA는 모델링에 앞서, 가설을 세우기 이전에 데이터가 무엇을 말하는지 들어보는 과정이다.
2. 타이타닉 fare의 평균(32.2)이 중앙값(14.5)보다 훨씬 큰 이유는 소수의 매우 큰 요금이 평균을 끌어올렸기 때문이며, 이런 경우 중앙값이 이상치에 더 강건한 대표값이다.
3. Deck 변수처럼 결측치 비율이 77.2%로 매우 높아도 무조건 평균이나 중앙값으로 대체하여 정보 손실을 최소화하는 것이 올바른 처리 방법이다.
4. fare 변수는 왜도(skewness) 4.79로 강한 우왜(right-skewed) 분포를 보이며, 이런 경우 log1p 변환을 통해 왜도를 줄여 정규분포에 가깝게 만들 수 있다.
5. 상관계수 히트맵은 범주형 변수만 포함하므로, 성별처럼 중요한 수치형 변수의 영향력이 표에 아예 드러나지 않을 수 있다.

서술형 문항은 반대로, 정답 하나를 맞히는 게 아니라 개념을 연결해서 설명하는 능력을 보게 만들었다. "정확도 100%가 왜 완벽한 모델이 아니라 문제 신호인지, Target Leakage로 설명하라"는 질문은 강의 전체(결측치·왜도·상관관계·Target Leakage)를 하나로 꿰어야 답할 수 있는 문제였다.

서술형

Q. EDA 없이 14개 컬럼을 모두 넣고 모델을 돌렸더니 정확도가 30회 반복 모두 100.0%(±0.0)로 나왔다. 왜 이것이 "완벽한 모델"이 아니라 오히려 **문제가 있다는 신호**인지, Target Leakage 개념을 이용해 설명하시오.

이 과정에서 느낀 건, NotebookLM 같은 도구는 "퀴즈를 대신 만들어주는 도구"라기보다 "내가 만든 자료를 학생 입장에서 한 번 더 훑어주는 두 번째 눈"에 가깝다는 것이었다.

자료를 다 쓴 사람은 이미 전체 맥락을 알고 있어서 정작 뭐가 헷갈릴 만한 지점인지 스스로 판단하기 어렵다. 그런데 AI가 뽑아낸 초안 문항들을 보면 "아, 이 부분이 오해하기 쉽겠구나"가 역으로 보였다.

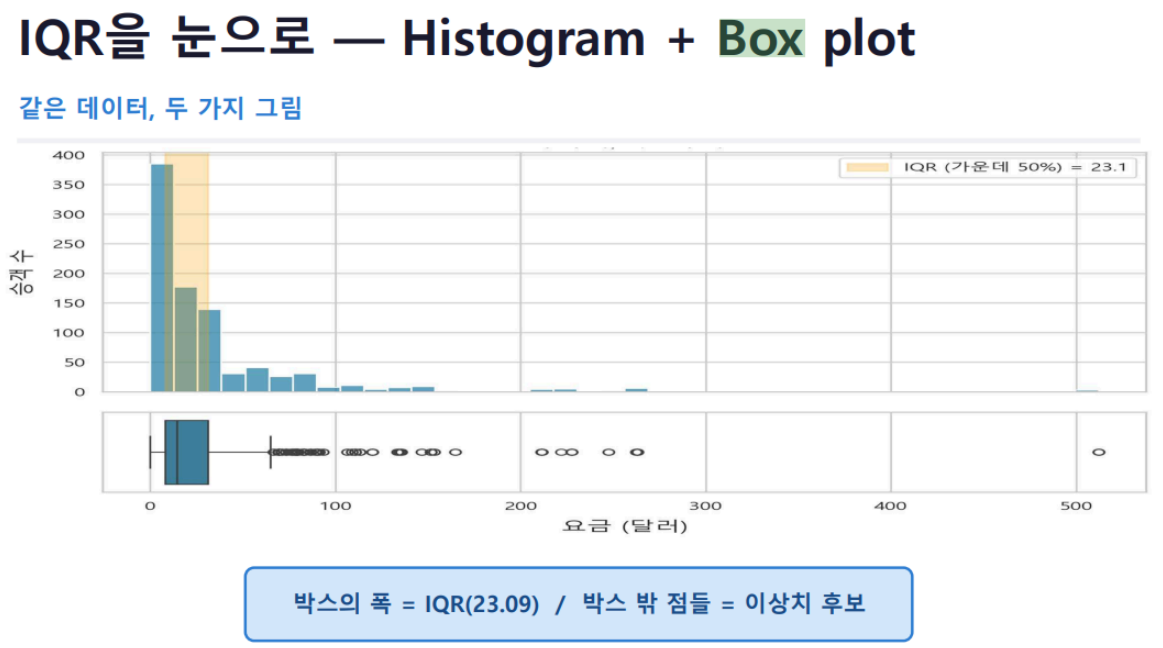
그래서 이번 한 번으로 끝낼 방법이 아니라, 앞으로 강의든 자기주도학습이든 자료를 완성한 뒤 습관처럼 NotebookLM에 넣어보는 루틴으로 가져가기로 했다. 자료를 만든 직후가 아니라, 한 김 식은 뒤 AI의 눈으로 다시 점검받는 단계를 하나 추가하는 셈이다.

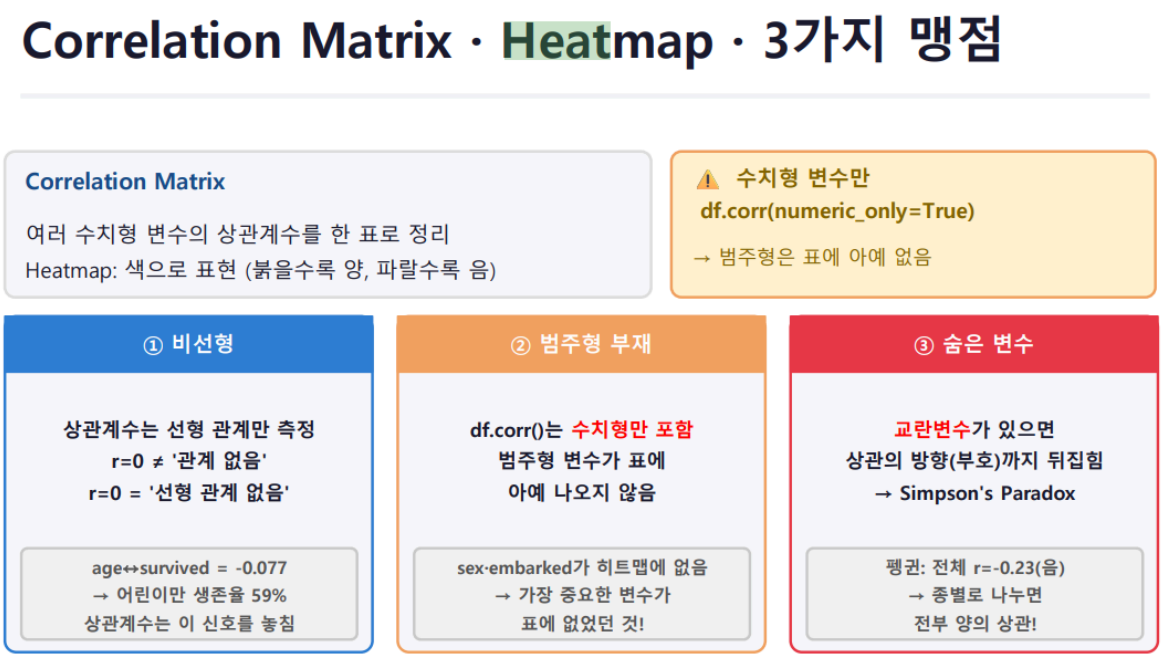
---

6. 결과 — 실제 강의와 학생 반응

이렇게 준비한 강의는 60장 안팎의 슬라이드, Colab 실습 노트북, Ames Housing 기반 과제로 완성됐다.

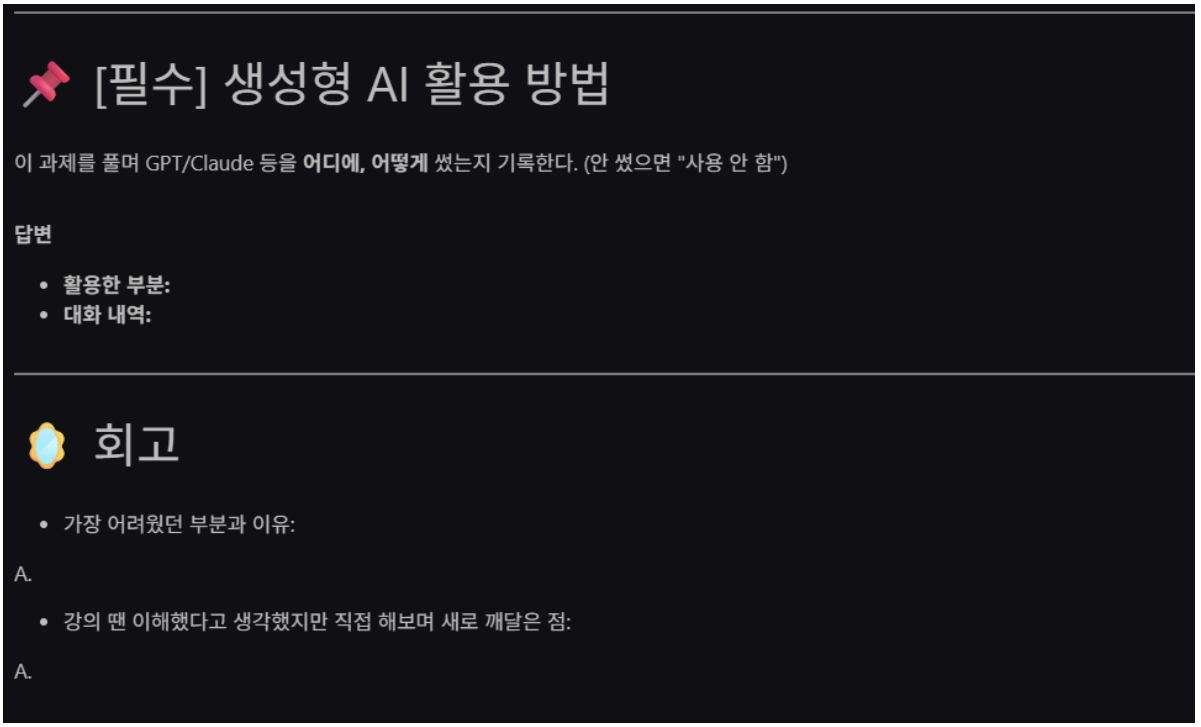
타이타닉 fare의 이상치를 IQR과 박스플롯으로 함께 보여주는 슬라이드, 상관계수 히트맵의 3가지 맹점(비선형·범주형 부재·숨은 변수)을 구조화한 슬라이드처럼, 앞선 탐색·검증·이미지 제작 단계에서 다듬은 개념들이 실제 강의 자료 한 장 한 장으로 들어갔다.





과제 노트북 끝에는 "생성형 AI 활용 방법" 섹션을 필수 항목으로 넣었다. 나 혼자 AI를 검증하며 강의를 준비한 것으로 끝내지 않고, 학생들에게도 AI를 어디에 어떻게 썼는지 스스로 기록하게 만든 것이다.

이 항목을 과제 맨 끝, 회고 섹션 바로 앞에 배치한 것도 의도가 있었다. 문제를 다 풀고 난 직후, "AI를 어디까지 쓰고 어디부터는 직접 생각했는지"를 스스로 되짚어보게 만드는 중간 점검 장치였다. AI를 아예 안 썼으면 "사용 안 함"이라고 적으면 그만이지만, 이 한 줄을 적기 위해서라도 학생은 자신의 풀이 과정을 한 번 더 복기하게 된다.



실제로 이 과제의 제출률은 98%로 나왔다. 단순히 정답 채점용 과제가 아니라 "AI를 어떻게 썼는지" 같은 과정을 함께 기록하게 만든 구조였는데도 제출률이 떨어지지 않았다는 점에서, 이 장치가 부담으로 작용하기보다는 학생 각자의 상황(AI를 많이 활용했든, 거의 안 썼든)에 맞춰 자연스럽게 채울 수 있는 형태였다고 판단했다.

나만의 방식 / 개선 포인트

이 다섯 단계(탐색, 검증, 이미지 제작, 적용, 자가진단)를 관통하는 한 가지 원칙이 있었다.

AI가 준 결과를 그대로 쓰지 않고, 매 단계마다 직접 검증하거나 재구성하는 과정을 거쳤다는 것.

- 논문 요약은 원문과 대조했고
- 통계 수치는 직접 코드를 돌려 재현했고
- 이미지는 강의 메시지에 맞는지 기준으로 다시 생성했고
- 실습 코드는 다른 AI에게 검토를 맡겨 교차검증했고
- 퀴즈는 내 강의의 빈틈을 찾는 데 거꾸로 활용했다

이 원칙은 배우는 사람일 때와 가르치는 사람일 때 서로 다른 방식으로 작동했다.

- **자기주도학습 단계**(탐색, 검증)에서는 AI가 준 답을 그대로 믿지 않고 "내가 정말 이해했는가"를 확인하는 데 썼다
- **수업준비 단계**(이미지 제작, 적용, 자가진단)에서는 그렇게 검증된 이해를 "학생이 직접 체감하고, 스스로 점검할 수 있는 형태"로 바꾸는 데 썼다

AI를 한 번 쓰고 끝내는 게 아니라, 매번 "이걸 어떻게 확인할 것인가"를 한 단계 더 붙인 셈이다. 그 결과 학생들에게도 같은 습관(AI 활용 내역을 스스로 기록하는 것)을 요구하는 장치를 과제에 넣을 수 있었고, 실제로 98%라는 제출률로 이어졌다.

다른 사람도 따라 할 수 있나

이 방식은 특정 도구나 특정 과목에 묶여있지 않다.

탐색(자료 조사) → 검증(직접 확인) → 시각화(이미지·자료 제작) → 적용(교차검증) → 자가진단(퀴즈화)

이 5단계 루틴은, 어떤 주제를 누군가에게 가르쳐야 하는 상황이라면 그대로 옮겨 쓸 수 있다.

- 앞의 두 단계(탐색·검증)는 혼자 공부하는 학생에게도 그대로 적용된다
- 뒤의 세 단계(이미지 제작·적용·자가진단)는 과외 선생님이 개념을 정리해 전달할 때도, 조교가 실습 자료를 만들 때도, 사내 스터디 리더가 발표 자료를 준비할 때도 똑같이 쓸 수 있다

핵심은 도구가 아니라 순서다. AI에게 물어보고 끝내는 게 아니라, 매 단계마다 "이걸 어떻게 확인할 것인가"를 한 번씩 더 거치는 습관.

앞으로 강의를 준비할 일이 또 생긴다면, 이 루틴을 하나의 습관으로 가져가볼 생각이다.