

# 데이터저널리즘의 AI 도구 활용 방안: 프롬프트를 넘어 에이전트 설계까지

이종혁(경희대 미디어학과)

# 데이터저널리즘 + AI

- 웹·유튜브·SNS까지 동시 대응해야 하는 멀티플랫폼 시대
  - 소수 인력으로 더 깊이있고 많은 기사·영상·데이터 생산 요구
- 고품질 뉴스 프로젝트가 팔리는 데이터저널리즘 시대
  - 심층기획+데이터기반+시각화
  - DB+개인맞춤형+웹+챗봇
  - 오래가는 유용한 프로젝트 필요
- AI 활용과 코딩 학습
  - AI는 기자를 대체하는 로봇이 아니라 ‘두 번째 뇌’

# 취재 워크플로우 9단계

- 1) 아이디어 발굴 · 뉴스가치 평가
- 2) 정보수집 · 리서치 · 기사/이슈 모니터링
- 3) 데이터 수집·분석·시각화
- 4) 인터뷰 · 현장취재 (녹취·번역·음성·영상 처리)
- 5) 기사 작성 · 번역 · 요약 · 스타일/오탈자 검토
- 6) 사실검증 · 이미지/영상 진위 확인
- 7) 로그·댓글·여론 반응 분석
- 8) 유튜브·쇼츠·SNS 버전 제작
- 9) 취재원·지식·자료 관리 (Knowledge DB)

# 데이터저널리즘 1.0 / 2.0 / 3.0

- 1.0: 전통적 데이터저널리즘
  - 엑셀·Tableau·Datawrapper·R 중심 (부분적 도구 보조)
- 2.0: AI 활용 저널리즘
  - 범용 LLM·전용 AI 도구
  - 프롬프트 엔지니어링 (적극적 AI 활용)
- 3.0: 에이전트 설계 저널리즘
  - 에이전트·RAG·MCP·A2A + DB + 챗봇
  - 코드 리터러시 + 콘텍스트 엔지니어링 (자동화 시스템)

# 데이터저널리즘 1.0 – 전통적 방법

- 공공·민간 데이터를 수집·분석해 그래프·지도·표로 스토리화
- 대표 도구: Excel/구글시트, R, Tableau, Datawrapper, Flourish
- 출력물: 지면 기사 + 정적 그래픽 중심
- 강점: 구조화된 데이터에 강함
- 한계: 속도·규모·인터랙션 제약

# 데이터저널리즘 1.0 – 워크플로우

- 아이디어: 과거 기사·회의·독자 제보·공공데이터 포털
- 정보수집: RSS, 구글, 포털·빅데이터(빅카인즈 등) 검색
- 데이터: 통계 다운로드 → CSV/엑셀 → R/SPSS 분석
- 인터뷰: 녹음기 + 수동 전사, 수기 메모, 이메일·전화
- 작성·편집·검증·반응 분석·자료 관리까지 대부분 수작업

# 데이터저널리즘 1.0 – 기법과 한계

- 대표 기법
  - 집계·비교·지도·네트워크(인물 관계), 시계열 추세 분석
  - 예: 집값 지도, 지역별 재정·환경·범죄 비교, 의회 활동비·발언 분석
- 한계
  - 수작업 비중이 커서 속도·규모에 한계
  - 코드 의존성 증가로 학습 장벽 존재
  - 실시간·개인화·인터랙티브 서비스 구현 어려움

# 데이터저널리즘 2.0 – AI 활용 저널리즘

- 기존 워크플로우 위에 LLM·전용 AI 도구를 얹어 생산성 극대화
- 프롬프트만 잘 써도 녹취·요약·초안·번역·코드·수식 자동 생성
- 핵심: 반복 업무는 AI에게, 해석·판단·윤리는 기자에게
- 위험: 환각·오역·편향 → 검증 리터러시 필수

# 데이터저널리즘 2.0 – 대표 도구

- 브레인스토밍·구조화: ChatGPT, Claude, Gemini
- 검색+요약 리서치: Perplexity, Genspark, Liner, Consensus
- 자료 기반 리서치: NotebookLM, AURA, Satchel(공공회의  
요약)
- 국문 녹취·요약: 클로바노트, 다글로, Otter 등 STT 도구
- 시각화 보조: Datawrapper, Flourish, Napkin AI 등

취재보도용 도구 가이드: [https://jonghhh.github.io/test/journalism\\_ai.html](https://jonghhh.github.io/test/journalism_ai.html)

# 데이터저널리즘 2.0 - 한계

- 도구·프롬프트 편차로 인한 재현성 부족
- 조직 아카이브·지식이 LLM과 연결되지 않음
- 일회성 파일·챗 기록 누적, 시스템 부재
- 반복 리서치·요약은 잘하지만 탐사형 문제의 구조적 진단에 취약
- 장기 프로젝트에서 사건 흐름·타임라인 자동 추적 어려움
- 도구 중심으로 정확성·깊이·검증 일관성 확보 어려움
- 속보·다중 기사 생산 압박으로 심층 분석에 시간 부족

# 데이터저널리즘 3.0 – 에이전트 설계 저널리즘

- 코드 리터러시와 데이터 구조 이해를 바탕으로 취재·분석·검증의 원칙을 에이전트 시스템에 구현
- 프롬프트 엔지니어링 → 컨텍스트 엔지니어링
  - 중요한 것은 ‘마법 문장’이 아니라 모델이 이해할 컨텍스트 설계
  - 키워드: 멀티에이전트, RAG, MCP, A2A
- 데이터·문서·도구까지 포함한 전체 맥락 조합 능력이 핵심
- AI 사용하는 기자 → AI 기반 취재 시스템 설계하는 기자

# 데이터저널리즘 3.0 – 시스템 구성

- 데이터 레이어: 기사 아카이브, 팩트체크 DB, 공공데이터, 여론조사 등
- 검색·임베딩: 벡터DB(Chroma, FAISS), 키워드 검색, 필터
- LLM 레이어: ChatGPT·Claude·Gemini + 자체 미세조정 모델
- 오케스트레이션: LangGraph, CrewAI, DS-STAR 등 멀티에이전트
- 인터페이스: 기자용 내부 툴, 독자용 챗봇, 대시보드, API

# 데이터저널리즘 3.0 – 필요 역량

- 기초 코드 리터러시: Python, JavaScript, SQL
- 데이터·모델 이해: RAG, 임베딩, API
- 워크플로우 설계: 에이전트가 분업할 수 있도록 단계 구성과 연결

# 1.0 / 2.0 / 3.0 비교

- 1.0: 전통적 데이터저널리즘
  - 데이터+그래프 중심, 수작업 비중 큼, 정적 기사·그래픽
- 2.0: AI 활용 저널리즘
  - 프롬프트+도구 활용, 생산성↑, 재현성·조직 지식 연결은 부족
- 3.0: 에이전트 설계 저널리즘
  - 데이터+DB+에이전트를 엮어 취재보도 시스템 설계

# 요약과 논의

- "AI를 사용했느냐?"가 아니라, "AI 덕분에 더 깊이 취재 보도했는가?"
- 데이터저널리즘 1.0: 그래프를 잘 만드는 기자
- 데이터저널리즘 2.0: AI 도구를 잘 다루는 기자
- 데이터저널리즘 3.0: 취재·보도 시스템을 설계하는 기자

"**AI**는 기자를 대체하지 않습니다.  
하지만 **AI**를 활용하는 기자가  
활용하지 않는 기자를 대체할 것입니다."