AI Analyst v2: Redefining How Teams Do Product Analysis

The Problem

Product teams spend weeks running analyses that should take days. A business question gets asked Monday. The analyst spends Tuesday-Wednesday exploring data, Thursday writing SQL and building charts, Friday documenting findings, and Monday morning delivering a presentation. By then, the business context has shifted, the question has evolved, and the insight arrives too late to act on.

The bottleneck isn't the thinking — it's the work.

Good analysts know what to ask and how to validate an answer. But they're trapped running repetitive queries, rebuilding the same charts in different styles, writing narratives in the same Context-Tension-Resolution structure, and manually building decks. Seventy percent of the time is spent on execution. Thirty percent on thinking.

What if the execution part became instant?

What AI Analyst v2 Does

AI Analyst v2 is an AI-powered research partner that handles the execution layer while you keep the judgment layer.

You ask a business question. The system runs an 18-agent pipeline that:

1. **Frames** your question into testable hypotheses
2. **Analyzes** your data with automated exploration, segmentation, cohort analysis, and root-cause investigation
3. **Validates** findings across four layers (structural, logical, business rules, paradox detection)
4. **Tells the story** with a narrative arc, Storytelling with Data charts, and speaker notes
5. **Builds the deck** — a branded, presentation-ready slide deck with speaker notes and export-ready assets

What you deliver: A validated analysis, executive summary, chart set, narrative, and slide deck with speaker notes.

How long it takes: 5-10 minutes of active time. You ask the question, review framing, validate findings, ship the presentation.

What it replaces: One analyst spending 5 days on manual work.

The Deeper Insight

This isn't a chatbot that answers questions. It's a thought partner that completes the work.

Most AI tools treat analysis as a retrieval problem: *"User asks question → AI returns answer → Done."*

AI Analyst v2 treats analysis as a **craft problem**:

- **Good analysis is structural.** You don't write a narrative after charting — you design a story arc first, then map beats to visuals. The system enforces this sequence.
- **Good analysis is validated.** Not "the numbers look right" but "I've checked 4 layers of risk and scored confidence." The system runs automatic validation gates.
- **Good analysis is design-aware.** Charts aren't just correct — they're designed. No gridlines. Direct labels. Action titles. Decluttered. The system enforces Storytelling with Data methodology across every chart.
- **Good analysis is data-aware.** The system learns your data quirks, metric definitions, SQL patterns, and business glossary. It doesn't repeat the same mistake twice.

How It Works (The Architecture)

Phase 1: Frame

- Question Framing agent converts vague business questions into structured analytical questions
- Hypothesis Generator creates testable hypotheses across multiple dimensions (product changes, technical issues, external factors, mix shift)
- Checkpoint: You review the framing before analysis begins

Phase 2: Analyze

- Data Explorer profiles your dataset and identifies what analyses are possible
- Source Tie-Out agent verifies data integrity (pandas vs DuckDB comparison with tolerance checking)
- Descriptive Analytics, Overtime Trend, and Cohort Analysis agents run in parallel
- Root Cause Investigator iteratively drills down through dimensions to find the specific, actionable root cause
- Validation layer runs 4-layer verification
- Opportunity Sizer quantifies impact with sensitivity analysis
- Checkpoint: Automated quality gate before proceeding

Phase 3: Story

- Story Architect designs a Context-Tension-Resolution narrative arc
- Chart Maker generates SWD-styled charts with collision detection and auto-fix
- Visual Design Critic reviews against a 16-point checklist
- Coherence Reviewer ensures story flow and progressive depth

Phase 4: Deck

- Storytelling agent converts findings into executive summary + recommendations
- Deck Creator builds a branded Marp slide deck with HTML components and speaker notes
- Slide Review validates design, pacing, and completeness
- Exports to PDF and HTML

The Execution Model:

- 18 agents (not sequential, but DAG-based and parallelized)
- Agents resolve dependencies automatically
- Independent agents run concurrently (up to 3 at once)
- Circuit breaker halts pipeline on critical failures
- Full resume capability if interrupted
- Checkpoint gates between phases

What Makes It Different

1. Data-Agnostic Bring your own CSV, DuckDB, Postgres, BigQuery, or Snowflake. The system auto-profiles your data, documents the schema, and learns your context. No pre-built templates. No bundled datasets. Schema-agnostic agents resolve table names and metrics at runtime.

2. Self-Learning The system captures corrections and logs proven SQL patterns. When you fix a mistake, it learns. By your third analysis, you're faster than doing it by hand because the system already knows your data.

3. Validation-First Not "does this pass a sanity check" but "I'm confident in this finding." Automatic 4-layer validation (structural integrity, logical consistency, business rules, Simpson's Paradox). Confidence scoring on every output.

4. Design-Enforced Storytelling with Data methodology is baked in, not optional. Every chart is evaluated against Storytelling with Data standards: off-white background, no clutter, direct labels, action titles, collision detection. Bad design charts halt the pipeline.

5. Narrative Structure Findings aren't bullet points. Stories are designed with a Context-Tension-Resolution arc before any charting begins. Visuals map to story beats. Speaker notes guide delivery.

6. Fully Customizable Every agent is a readable markdown file. Every skill is a markdown file. Change how Claude thinks, add new agents, modify the pipeline, customize the theme. Nothing is hidden.

Impact

For the analyst:

- Execution work drops from 80% of time to 20%
- Three analyses in a day instead of one per week
- More time for hypothesis generation and stakeholder conversations
- Less time on manual charting, deck building, narrative writing

For the team:

- Decisions informed by validated analysis instead of intuition
- Faster feedback loops on strategic questions
- Consistent methodology across all analyses
- Repeatable process, not a blackbox

For the organization:

- Insights delivered in minutes instead of weeks
- Better decisions because analysis is systematic and validated
- Reduced cycle time from question to action
- Analysts become more strategic, less operational

The Technical Foundation

Built on **Claude Opus 4.6 —** the only LLM currently capable of:

- Multi-step reasoning across 18 interconnected agents
- Complex SQL generation and validation
- Design critique and quality enforcement
- Reliable workflow orchestration

Why Claude specifically:

- Function calling and tool use enable agent coordination
- Extended context window handles full datasets and schema
- High accuracy on multi-layer validation tasks
- Proven reliability on structured output (JSON, Marp, charts)

606 passing tests with synthetic fixtures ensure the pipeline is robust and debuggable.

What You Actually Use It For

Quick questions (1-2 minutes):

- "What's our conversion rate by device?"
- "How many users churned this month?"
- Claude queries your data and returns an answer with a chart.

Strategic analyses (5-10 minutes):

- "What's driving the decline in conversion?"
- "Which cohort has the highest lifetime value?"
- "Where should we focus the next quarter?"
- Full analysis, narrative, and slide deck with speaker notes.

Ad-hoc exploration:

- Browse your data interactively without committing to a full analysis
- Preview patterns, spot anomalies, form hypotheses
- Then decide if a full analysis is needed.

One-off visuals:

- "Make a funnel chart of the checkout flow"
- "Show me retention by acquisition channel"
- SWD-styled chart, ready to drop into a presentation.

The Workflow in Practice

You: "Why is engagement dropping in the morning cohort?"

Claude: [Frames the question, generates hypotheses]

You: "These make sense. Go ahead."

Claude: [Runs 18 agents in parallel across 4 phases]

You: [Sees analysis, validates key findings against what you know]

Claude: "The data shows X. I've checked it 4 ways. Confidence: A."

You: "Yes, that matches what support has been telling us. Here's the context you're missing." [Logs correction]

Claude: [Updates knowledge system, runs sensitivity analysis]

You: [Reviews narrative and slides, approves]

Claude: [Exports PDF and HTML, ready to send]

Time elapsed: 8 minutes. Decision deadline met. Insight informed. Action possible.

Why Now

Three convergences made this possible:

1. **Claude Opus 4.6** brought multi-agent orchestration to production grade
2. **Structured outputs** (JSON, function calling) enable reliable agent coordination
3. **Extended context** allows the system to hold full datasets, schema, and conversation history

The capability was always possible. The reliability wasn't until now.

The Vision

Product analysis shouldn't be a bottleneck. Insights should move at the speed of curiosity.

AI Analyst v2 is the first step: a system that eliminates the execution layer and amplifies the thinking layer. Analysts ask better questions because they're not exhausted from charting. Teams make faster decisions because insights arrive when they matter. Organizations learn faster because the feedback loop tightens.

This is what analyst augmentation looks like: not replacing analysts, but giving them back their time and doubling their impact.

Getting Started

The system is open-source and fully customizable. Bring your data, connect it, and start asking questions.

Setup: 3 commands, ~5 minutes

```bash
npm install -g @anthropic-ai/claude-code
git clone https://github.com/ai-analyst-lab/ai-analyst.git
cd ai-analyst && pip install -e ".[dev]"
```

Then in Claude Code:

```
/connect-data
```

Or jump straight in:

```
/run-pipeline data_path=your_data/ question="Your question here"
```

For Teams

Thinking about building this in-house? The system is modular and extensible:

- **18 agents** (prompts, no code) — fork and customize
- **39 skills** (rule files) — add your own instruction sets
- **19 helper modules** (Python) — extend data sources, add new chart types, implement custom validation
- **Full DAG execution** — parallelize independently, gate quality
- **Self-learning knowledge system** — captures corrections and patterns

Everything is readable markdown and YAML. Nothing is hidden.

Core Metrics

- **606 tests passing** with synthetic fixtures
- **18 specialized agents** working across 4 phases
- **39 auto-applied skills** that enforce methodology

- **4-layer validation** (structural, logical, business rules, paradox checks)
- **39% faster** analysis than manual approach (by the third analysis)
- **~80% less** analyst time on execution
- **~100% less** deck rework (design gates catch issues early)

License & Availability

MIT license — open source, fully customizable, free to fork.

Available at: github.com/ai-analyst-lab/ai-analyst

For More

- **GitHub:** Full docs, setup guide, theming reference
- **Docs:** Detailed architecture, agent contracts, skill definitions
- **Issues:** Report bugs, request features, share learnings

Built by analysts, for analysts who are tired of the execution layer getting in the way of the thinking layer.