



An agent can build your report in minutes. **Can you trust it?**

AI agents now turn a prompt and a data file into charts, slides and dashboards. The first result often looks good. The trouble starts with the **second run**, and with the question nobody can answer quickly: **is this report right?** Most ways to let an agent build a report give you a picture, not something you can check.

01 Where agentic reporting breaks

DIFFERENT EVERY RUN

Same prompt, same data, a new report. The chart type changes, the order changes, the scale changes. This month cannot be compared with last month, because the agent made new design decisions.

LAYOUT BY CHANCE

The agent writes chart code or fills slides piece by piece. Labels overlap, fonts shrink, two charts of the same measure use two scales. The agent cannot see the result, and nothing measures it.

NOTHING TO VALIDATE

The output is a file or an image. There is no schema that says what a correct report is. A wrong number looks exactly like a right one, and the tool cannot tell the agent it made a mistake.

NUMBERS WITHOUT ORIGIN

Which query, which filter, which rows? Values end up in a slide or a cell with no path back to the source. To check one figure you have to rebuild it by hand.

GAPS NOBODY SEES

A region is missing, a chart is empty, a value is in EUR instead of kEUR. The report still renders, so nobody notices. The agent does not know which part of the data reached the page.

NO REVIEW, NO AUDIT TRAIL

Each run produces a finished document, so each run must be checked from scratch. A chat log is not an audit trail. Nobody can say what changed since the last version, and why.

02 What an agent needs before you can trust its report

None of these is about a better model. They are properties of the format and the tools the agent works with.

I	II	III	IV	V
Deterministic	Checkable	Data-aware	Traceable	Reviewable
Same input, same output . Layout and notation are fixed by an engine, not decided again on every run.	A format with a schema , so a tool can say "wrong" before a person ever sees the result.	The agent can look at the data , and the tools tell it when data does not reach the page.	Every number goes back to a named query and a source. No values pasted by hand.	A change is a diff a person reads in minutes, in Git, with the same review as code.

THE REAL QUESTION

Not "can an agent make a report?" It can. The question is whether you can **repeat** it next month, **check** it before it goes out, and **see what changed** since the last version. That is a question about the report format, not about the model.

03 The answer: the agent writes **text**, bino does the rest

With bino a report is **plain YAML and SQL**. Language models read and write that well, and a tool can parse and check it. The agent decides **what** the report shows. The engine decides **how**: layout, scaling and IBCS notation come out the same on every run.

For the agent, bino ships an **MCP server** (`bino mcp`) with 29 tools. Any MCP client (Claude Code, Claude Desktop, Cursor) can read the schema, look at the data, check a draft, build the PDF and measure what was rendered. The steps below are the loop it follows.

The agent never places a pixel. It cannot make the chart look different next month, because it does not draw the chart.

WHAT THE AGENT WRITES

```
apiVersion: bino.bi/v1alpha1
kind: DataSet
metadata: { name: revenue_by_region }
spec:
  dependencies: [sales_csv]
  query: |
    SELECT region      AS category,
           region_id AS categoryIndex,
           actual      AS acl, -- actual
           plan        AS pll  -- plan
    FROM sales_csv
---
apiVersion: bino.bi/v1alpha1
kind: Table
metadata: { name: revenue_table }
spec:
  dataset: revenue_by_region
  scenarios: [pll, acl]
  variances: [dac1_pll_pos]
```

1 · learn Read the rules schema + IBCS skills	2 · look Look at the data ② <code>introspect_source</code>	3 · draft Write YAML + SQL in memory	4 · gate Check the draft ② <code>validate_draft</code>	5 · build Render the PDF ② <code>build</code>	6 · gate Measure the result ② <code>get_layout_state</code>	7 · gate A person signs off ② reads the PDF or the diff
--	---	---	---	--	--	--

every error, finding or review note ↩ back to step 3

04 Why this is better: each problem has an answer

THE PROBLEM (FRONT PAGE)	WHAT BINO DOES ABOUT IT
Different every run	The engine renders layout and IBCS notation. Same manifests + same data = identical PDF .
Layout by chance	<code>get_layout_state</code> reports empty charts, overflow, shrunk fonts and mixed scales.
Nothing to validate	A schema for every kind, <code>validate_draft</code> in memory, 33 lint rules. Invalid files are not written.
Numbers without origin	Every figure comes from a named DataSet query. <code>graph_deps</code> shows which column a chart reads.
Gaps nobody sees	Query results are validated. <code>introspect_source</code> and <code>get_rows</code> let the agent check the data.
No review, no audit trail	The report is text in Git. Every agent change is a diff, reviewed like code and rebuilt by CI.

05 What you can use today

BINO MCP · FOR ANY AGENT

29 tools to inspect, validate, write, build and measure, plus skills for IBCS and data modeling. Works with Claude Code, Claude Desktop, Cursor and any other MCP client.

CLAUDE CODE PLUGIN · CO-AUTHOR OR AUTOPILOT

`/bino:report` goes from data and a goal to a PDF. `/bino:autopilot` runs the pipeline and stops at four checkpoints, the last one always a person.

IN CI · THE SAME CHECKS

`bino lint --fail-on-warnings` and `bino build --layout-state` run the same checks on every push, so an agent's change cannot skip them.

TRY IT IN FIVE MINUTES

```
# install the CLI and its renderer
brew install bino-bi/tap/bino-cli
bino setup # browser runtime for PDFs
```

```
# give Claude Code the bino tools
claude plugin marketplace add bino-bi/bino-cli
/plugin install bino
/bino:report "monthly revenue vs plan
              by region, from sales.csv"
```

USEFUL LINKS

[cli.bino.bi/cli/mcp](#) · the MCP server
[cli.bino.bi/cli/claude-plugin](#) · the plugin
[cli.bino.bi/cli/layout-state](#) · layout checks
[www.bino.bi/agent-run](#) · a real agent run
[sandbox.bino.bi](#) · try bino in the browser