CausalLab CLI

User Manual | schema_version: 2.0 | generated: 2026-07-04 (UTC)

Overview

CausalLab CLI is a local prototype command-line tool for exploratory causal-style association analysis on small tabular CSV datasets. It loads a local CSV file, inspects its columns, computes simple deterministic associations between a numeric target variable and other variables, and can explain and simulate effects in human-readable text or JSON.

A typical workflow starts from a CSV file: you first inspect the inferred schema, then list variables with their types and completeness, compute associations for a chosen numeric target, explain the main relationships, optionally simulate the effect of changing a driver variable, and finally generate a stable JSON report summarizing the dataset and results. Running the same sequence of commands on the same CSV produces the same observable outputs.

The CLI entrypoint is python -m app.cli (shown as the causalab command in help text) with subcommands schema, variables, correlate, explain, simulate, and report. Each command expects a positional input CSV path and supports a global --format option to select text (default) or JSON output.

Getting started

Start with the discovery commands below to inspect the available CLI surface.

Use top-level help, command-group help, and action-level help to identify the documented commands, options, and actions.

End-to-end workflows are described only when the available documentary material supports them.

Starter commands

```
$ python -m app.cli --help
$ python -m app.cli
$ python -m app.cli schema data.csv
$ python -m app.cli variables data.csv
$ python -m app.cli report data.csv --target <target-variable>
     --var <driver-variable> --delta <delta-value> --output report.json --format json
```

How to use the prototype

Discover available commands and global options

Use the top-level help to discover the CausalLab CLI surface, including available commands and the global --format option. Running the CLI without arguments also prints the same usage information.

This is a safe way to confirm the command names, required input argument, and output format choices before running any analysis.

Commands

```
$ python -m app.cli --help
$ python -m app.cli
```

Notes

- Both commands print usage: causalab <command> input [options].

- Top-level help lists the schema, variables, correlate, explain, simulate, and report commands.

- Help output includes the global --format {text,json} option (default: text) and exits with status code 0.

Inspect the inferred schema of a CSV dataset

Use the schema command to inspect the inferred schema of a CSV file. This describes the expected columns and formats for example fixtures and documents the runtime contract consumed by other commands.

You can render the schema as human-readable text or as JSON, which is useful for programmatic inspection of the CLI surface, expected CSV structure, and report payload shape.

Commands

```
$ python -m app.cli schema data.csv --format text
$ python -m app.cli schema data.csv --format json
```

Notes

- schema takes a positional argument input: the path to the input CSV file.

- The command consumes a local CSV file only; no network or external services are used.

- When using --format json, the output is a deterministic machine-readable contract aligned with the active CLI surface, fixture CSV expectations, and JSON report shape.

List variables and completeness metrics

Use the variables command to list all detected variables (columns) in your CSV file along with inferred types and basic completeness metrics.

This helps you understand which variables are numeric, which have missing data, and how many values are available before computing associations or simulations.

Commands

```
$ python -m app.cli variables data.csv --format text
$ python -m app.cli variables data.csv --format json
```

Notes

- variables takes a positional argument input: path to the input CSV file.

- Output includes variable names, inferred logical types, value counts, and missing-value counts.

- The output ordering is deterministic so repeated runs on the same CSV produce the same listing.

Compute associations for a target variable

Use the correlate command to compute simple exploratory associations between one numeric target variable and other variables in the dataset.

The command returns a stable summary of associations that can be rendered as text or JSON, forming the basis for later explanations and simulations.

Commands

```
$ python -m app.cli correlate data.csv --target <target-variable>
    --format text
$ python -m app.cli correlate data.csv --target <target-variable>
    --format json
```

Notes

- correlate takes a positional argument input (CSV path) and a --target option specifying the target numeric variable name.

- Associations are computed in-memory using deterministic logic; the same CSV and target yield the same association summary.

- --format controls whether you see a human-readable text summary or a JSON structure suitable for downstream processing.

Explain top associations for a target variable

Use the explain command to obtain a readable explanation of the main associations for a given numeric target variable.

The explanation translates association values into qualitative descriptions such as positive, negative, or weak associations, using deterministic French text rendering.

Commands

```
$ python -m app.cli explain data.csv --target <target-variable> --format text
$ python -m app.cli explain data.csv --target <target-variable> --format json
```

Notes

- explain takes a positional argument input (CSV path) and a --target option for the target numeric variable.

- Text output summarizes which variables are most strongly associated with the target and the direction of those associations.

- JSON output provides a structured version of the explanation payload.

Simulate the effect of changing a driver variable

Use the simulate command to estimate how a change (delta) in one driver variable might affect a numeric target variable, based on previously computed associations.

This provides a deterministic exploratory simulation rather than a full causal model, helping you reason about potential effects of simple changes.

Commands

```
$ python -m app.cli simulate data.csv --target <target-variable>
     --var <driver-variable> --delta <delta-value> --format text
$ python -m app.cli simulate data.csv --target <target-variable>
     --var <driver-variable> --delta <delta-value> --format json
```

Notes

- simulate takes a positional argument input (CSV path) plus --target for the target variable, --var for the driver variable, and --delta for the numeric variation applied to the driver.

- The command validates that the target and driver variables are present and numeric in the dataset.

- Simulation results are deterministic mappings ready for text or JSON rendering; there is no random component.

Generate a deterministic JSON report for one workflow

Use the report command to generate a deterministic JSON report for a complete workflow on one CSV file. The report summarizes the dataset, variables, associations, explanation, and simulation for the specified target and driver variables.

The report is written as JSON to an explicit file path that you control, making it suitable for archiving or automated pipelines.

Commands

```
$ python -m app.cli report data.csv --target <target-variable>
     --var <driver-variable> --delta <delta-value> --output report.json --format json
```

Notes

- report takes a positional argument input (CSV path) plus --target, --var, and --delta options to define the target variable, driver variable, and numeric variation.

- --output specifies the exact destination JSON report file path; the reporting module creates parent directories, writes one file, and returns the written path.

- The report payload is serialized as stable JSON with deterministic ordering; running the same report command on the same inputs produces the same JSON file.

Command reference

Documented global options

Global options
--help
--format

Group: correlate

Field	Value
Actions	input

Commands

```
$ python -m app.cli correlate --help
$ python -m app.cli correlate
```

Group: explain

Field	Value
Actions	input

Commands

```
$ python -m app.cli explain --help
$ python -m app.cli explain
```

Group: report

Field	Value
Actions	input

Commands

```
$ python -m app.cli report --help
$ python -m app.cli report
```

Group: schema

Field	Value
Actions	input

Commands

```
$ python -m app.cli schema --help
$ python -m app.cli schema
```

Group: simulate

Field	Value
Actions	input

Commands

```
$ python -m app.cli simulate --help
$ python -m app.cli simulate
```

Group: variables

Field	Value
Actions	input

Commands

```
$ python -m app.cli variables --help
$ python -m app.cli variables
```

Inputs and outputs

All CausalLab CLI commands expect a positional input argument that is the path to a local CSV file. The tool reads this file into memory and never contacts external services or databases.

Analysis commands produce either human-readable text or JSON on standard output, depending on the --format option, while the report command additionally writes a deterministic JSON report file to a user-specified path.

Type	Value
Required input	Path to the input CSV file for all commands (positional argument input).
Required input	Target numeric variable name for correlate, explain, simulate, and report (--target).
Required input	Driver numeric variable name for simulate and report (--var).
Required input	Numeric variation value applied to the driver variable in simulate and report (--delta).
Required input	Destination file path for the JSON report in the report command (--output).
Produced output	Text or JSON description of the inferred dataset schema.
Produced output	Text or JSON listing of variables with inferred types and completeness metrics.
Produced output	Text or JSON summary of exploratory associations for a target variable.
Produced output	Readable text or JSON explanation of the main associations for a target variable.
Produced output	Text or JSON summary of the simulated effect of changing a driver variable on the target.
Produced output	Deterministic JSON report file written to the path provided via --output.

Configuration

CausalLab CLI has a minimal configuration surface focused on choosing the output format and, for the report workflow, the report destination path.

You control the representation of command results with the global --format option, and you control where the report workflow writes its JSON file with the --output option on the report command.

Current limitations

The prototype is designed for local exploratory analysis of small tabular CSV files and does not integrate with remote services or non-CSV data sources.

It focuses on simple deterministic associations and simulations rather than full causal inference or large-scale machine learning workflows.

Documented limitations
Operates only on local CSV files; no database or network inputs are supported.
Processes data in memory and is intended for small datasets rather than large-scale data processing.
Provides association-based exploration and simple simulations, not full causal modeling or predictive machine learning.
Workflows are deterministic: running the same command sequence on the same CSV produces the same observable outputs.

Project map

Application layout

```
app/
|_ cli.py
|_ orchestrator.py
|_ associations.py
|_ csv_loader.py
|_ json_rendering.py
|_ reporting.py
|_ runtime_schema.py
|_ schema_inference.py
|_ simulation.py
|_ text_rendering.py
```