
UC-1: Load Data Sources

- **Primary Actor:** Data Analyst / Data Engineer
- **Goal:** Import external datasets into Noeta's workspace.
- **Pre-conditions:** Data file path is available.
- **Post-conditions:** Dataset is loaded under a specified alias.
- **Priority:** High
- **Frequency:** Frequent
- **Normal Flow:** `load "file.csv" as alias`
- **Alternatives:** Support JSON, Excel in future.
- **Exceptions:** File not found, unreadable format → handled via error message.
- **Special Requirements:** Use Pandas `read_csv`, graceful error handling.
- **Assumptions:** File exists and is readable.

UC-2: Filter Data

- **Primary Actor:** Data Analyst
- **Goal:** Filter rows based on logical conditions.
- **Pre-conditions:** Dataset is loaded.
- **Post-conditions:** A new filtered dataset alias is created.
- **Priority:** High
- **Frequency:** Frequent
- **Normal Flow:** `filter alias [condition] as new_alias`

- **Exceptions:** Invalid column or syntax → error message.
- **Special Requirements:** Simple expression parser for conditions.
- **Assumptions:** Column exists, condition is logically valid.

UC-3: Join Datasets

- **Primary Actor:** Data Analyst / Engineer
- **Goal:** Join two datasets on common column(s).
- **Pre-conditions:** Two datasets loaded.
- **Post-conditions:** Merged dataset available under new alias.
- **Priority:** High
- **Frequency:** Regular
- **Normal Flow:** `join alias1 with: alias2 on: col as new_alias`
- **Exceptions:** Missing join column, datatype mismatch.
- **Special Requirements:** Support inner/outer joins.
- **Assumptions:** Join columns exist in both datasets.

UC-4: Group & Aggregate Data

- **Primary Actor:** Data Analyst
- **Goal:** Group and summarize data using aggregate functions.
- **Pre-conditions:** Dataset is loaded and cleaned.
- **Post-conditions:** New alias with grouped/aggregated data.
- **Priority:** High
- **Frequency:** Frequent

- **Normal Flow:** `groupby alias by: {col1} agg: {avg: col2} as new_alias`
- **Exceptions:** Invalid aggregation function or non-numeric column.
- **Special Requirements:** Auto support `sum`, `avg`, `min`, `max`.
- **Assumptions:** Columns used are valid.

UC-5: Handle Missing Data

- **Primary Actor:** Data Analyst
- **Goal:** Drop or fill missing values in data.
- **Pre-conditions:** Dataset contains null/missing values.
- **Post-conditions:** Cleaned dataset under a new alias.
- **Priority:** Medium
- **Frequency:** Frequent
- **Normal Flow:** `dropna alias columns:{col1}` or `fillna alias value:0`
- **Exceptions:** Column not found.
- **Special Requirements:** Optional column specification.
- **Assumptions:** Nulls are present.

UC-6: Data Visualization

- **Primary Actor:** Data Analyst
- **Goal:** Visualize datasets using common plot types.
- **Pre-conditions:** Dataset is ready for plotting.
- **Post-conditions:** Plot shown in Jupyter or saved.

- **Priority:** High
- **Frequency:** Frequent
- **Normal Flow:** `plotchart alias` / `boxplot alias columns:{col}`
- **Exceptions:** Column not found or not numeric.
- **Special Requirements:** Matplotlib/Seaborn integration.
- **Assumptions:** Columns are plot-friendly.

UC-7: Statistical Analysis

- **Primary Actor:** Data Analyst
- **Goal:** Derive key stats or perform tests.
- **Pre-conditions:** Dataset is numerical or partially numerical.
- **Post-conditions:** Result shown or saved.
- **Priority:** Medium
- **Frequency:** Frequent
- **Normal Flow:** `describe alias` / `quantile alias column: col1 q: 0.75`
- **Exceptions:** Column not numeric.
- **Special Requirements:** Built-in statistical summary functions.
- **Assumptions:** Data supports requested operation.

UC-8: Export Data & Plots

- **Primary Actor:** Data Analyst
- **Goal:** Export datasets and plots.

- **Pre-conditions:** Output is ready.
- **Post-conditions:** File is saved to disk.
- **Priority:** Medium
- **Frequency:** Regular
- **Normal Flow:** `save alias to: "output.csv"` / `export_plot filename:"fig.png"`
- **Exceptions:** File write permissions, invalid path.
- **Special Requirements:** Support for CSV, PNG, JPEG.
- **Assumptions:** Output directory is writable.

UC-9: Interactive Querying (Jupyter)

- **Primary Actor:** Data Analyst
- **Goal:** Run Noeta code in cells, interactively.
- **Pre-conditions:** Jupyter installed, Noeta kernel selected.
- **Post-conditions:** Result shown in notebook output.
- **Priority:** High
- **Frequency:** Frequent
- **Normal Flow:** `load`, `select`, `plotchart` etc. in cells.
- **Exceptions:** Syntax/semantic errors shown in Noeta terms.
- **Special Requirements:** Kernel plumbing (stdin_channel, etc.).
- **Assumptions:** Notebook supports inputs.

UC-10: CLI Interaction

- **Primary Actor:** Developer / Analyst
- **Goal:** Run `.noeta` scripts or commands via terminal.
- **Pre-conditions:** Noeta is installed.
- **Post-conditions:** Output displayed or written to file.
- **Priority:** Medium
- **Frequency:** Regular
- **Normal Flow:** `noeta run script.noeta` / `--run` / `--output`
- **Exceptions:** Invalid directive, missing arguments.
- **Special Requirements:** Typer-based CLI with help output.
- **Assumptions:** Python + CLI environment ready.
