

SAE Studio: The Complete How-To Guide

The file, the column roles, the model setup, the results, the diagnostics, and the exports.

MAINEBLOCK · NUMBERS, DONE RIGHT.

October 2026 · maineblock.com

Contents

1. What SAE Studio Does
2. Your First Run: The Sample Data
3. Step One: Upload and Profile
4. Preparing Your Own File
5. Confirming the Column Roles
6. Step Two: Setting Up the Model
7. Choosing Covariates
8. Run Settings
9. Before You Run: Preview and Checks
10. Running the Model
11. Step Three: Reading Your Results
12. Step Four: Diagnostics and the Publication Checklist
13. The Methods Report
14. What the Validation Record Shows
15. Exports and the License
16. The Companion Workbook
17. Your Project, Saved in This Browser
18. Troubleshooting

What SAE Studio Does

SAE Studio is a tool for small area estimation. A small area is a geography or a group for which a survey collected too few responses to support a direct estimate with acceptable precision. Counties, towns, school districts, and similar units are common examples. The direct estimate for such an area is the figure computed from the responses in that area alone. It is unbiased in the usual survey sense when the design is respected, but its sampling variance can be large, its coefficient of variation can be high, and its interval can be so wide that publication would mislead a reader about what is known.

The model in SAE Studio addresses that problem by borrowing strength. It assumes that the direct estimates across areas share a pattern explained in part by covariates that are known for every area, and in part by an area effect that remains after the covariates are taken into account. Areas with precise direct estimates keep most of their own information. Areas with noisy direct estimates are pulled more toward the prediction from the covariates. The result is an estimate for each area that is a weighted combination of its direct estimate and its regression prediction. The weight is called shrinkage, and the guide returns to it in detail in the results chapter.

The model fitted is the Fay–Herriot area-level model, named for the paper by Fay and Herriot in 1979. Area-level means the model works with one direct estimate per area, together with its sampling variance, rather than with one record per respondent at the fitting stage. When you supply a respondent-level file, the tool first computes the direct estimates and their variances from the microdata under the survey design you declare through the column roles, and then it fits the same area-level model to those computed values. The two input paths therefore meet at the same model, and the later steps of the workflow are the same for both.

Three limits define what the tool handles, and they are stated at the start so that no later chapter needs to qualify them.

First, every outcome the tool handles is a proportion. Published values live in the interval from zero to one. An area-level input is a proportion on the scale from zero to one, or a percent on the scale from zero to one hundred, and you declare which scale your file uses. Means and totals are not accepted as area-level inputs. If your source file holds a mean or a total, it cannot be run on the area-level path in this tool. Convert the work to a proportion with a clear numerator and denominator before you begin, or use a different tool for that outcome.

Second, intervals always lie inside zero to one. Intervals are formed on a transformed scale and mapped back to the proportion scale, so an interval never extends below zero and never extends above one. That property holds in the chart on screen and in the numbers in the exports.

Third, the statewide figure in the results comes from population-weighted figures on the proportion scale. When population counts are available, the areas are weighted by their population counts. When population counts are not available, the figure is the plain mean of the areas. The statewide figure is never a mean taken on the transformed scale. A transformed-scale value is a diagnostic used inside the fit, and the guide does not present it as a statewide result.

The tool holds four steps. Navigation between the steps is free, and you can open any step at any time. A step that has no content yet shows an empty state that tells you what to do first. The four steps, with their exact titles and subtitles, are as follows.

Step	Title	Subtitle	What It Holds
1	"Upload & Profile"	"File, Roles, and Coverage Check"	File intake, sample data, coverage view, and column role confirmation
2	"Configure Model"	"Covariates, Transform, Estimator"	Model setup, run controls, "What Will Run", feasibility preview, and quality checks
3	"Results"	"EBLUP Table and Chart"	Statewide readout, estimates table, direct and model chart, and "Reading These Results"
4	"Diagnostics & Export"	"Checks, Interpretation, Files"	Diagnostics, publication checklist, export center, QC report, and project controls

The address is https://app.maineblock.com/. The browser tab title reads "SAE Studio | Small Area Estimation Without Writing Code". No account is needed to open the tool and run the trial workflow, and nothing needs installing. The calculations run in the page itself.

The tool is designed for desktop use. On a screen narrower than 760 pixels, a notice appears above the app with this exact text: "SAE Studio is designed for desktop use. The full workflow requires a larger screen. Open this address on a desktop or laptop computer to run the complete tool." The notice has a close button with the accessible label "Dismiss desktop notice". Dismissal is remembered for the browser session only. The full workflow requires the larger screen, and this guide assumes a desktop or laptop display throughout.

> Your data stays in your browser. The file is read in your browser and is never uploaded. The project autosaves in this browser. No data, results, or file names are sent to a server. The only network call the app makes is the license check, and that check sends the email address and the license key. Chapter Fifteen describes the license check in full.

On a first visit in a clean browser, the app loads a fresh project named "New project", starting on Step One, with the default run configuration already set. If the same browser holds an autosaved project, that project is restored instead. The header shows the SAE Studio logo, the project name after a dash, and a subtitle line built from the loaded file name when a file is loaded, the area count when a file is profiled, and the current run mode, which reads "Quick mode" or "Full mode". On first load the subtitle reads only "Quick mode".

The header also holds the appearance controls, a status badge, a "Save Project" button, and a license button. The status badge reads "TRIAL PREVIEW · EXPORTS LOCKED" when no license is active and "LICENSED" when a license is active. The license button reads "Unlock the Tool" when no license is active. When a license is active, the button shows the license email, or the word "License" if no email is on record. Both forms of the button open the license dialog.

Appearance is controlled in the header and never changes what is computed. A segmented switch with the group label "Color mode" offers three icon buttons with the accessible labels "Light mode", "Dark mode", and "System mode". The default is "Light mode". "System mode" follows the operating system light and dark setting as it changes. Beside the switch, a round color button with the tooltip "Theme color" opens a strip in the header labeled "Theme Color" with nine swatches, in this order: "Signal Orange", "Blue", "Sky", "Violet", "Rose", "Amber", "Green", "Slate", and "Mono". "Signal Orange" is the default. Picking a swatch applies it and leaves the strip open. The color button or the Escape key closes the strip. Color mode and theme color persist in this browser only and are applied before the first paint on later visits.

Two kinds of file are accepted, and the distinction matters for every later step.

An area-level file has one row per area, with the direct estimate and its sampling variance already computed. It must also carry the sample size for each area. A standard error can be supplied in place of the sampling variance, and the tool squares it. Covariates can sit in the same file or in a separate companion file keyed by the same area codes.

A microdata file has one row per respondent. It must carry the outcome variable, the survey weight, and the design variables, which are strata and primary sampling units, or replicate weights. The tool computes the direct estimates and variances for you from those columns.

Accepted formats are stated on screen as .csv, .xlsx, .sav (SPSS), .dta (Stata), .sas7bdat and .xpt (SAS). The file picker also accepts .tsv, and the reader treats a .txt file as a comma-separated file. Column roles are detected automatically. You confirm them before anything runs.

The remainder of this chapter sets out the shape of a complete run so that the walkthrough in the next chapter needs no new ideas. You load a file in Step One and confirm the column roles. You set the outcome, the covariates, and the run settings in Step Two, and you read the preview and the quality checks before you run. You press "Run Model →" and the app fits one model cell for the outcome as a whole, or one cell for each selected subgroup level when subgroup variables are in use. You read the estimates, the chart, and the interpretation in Step Three. You check the diagnostics and the publication checklist in Step Four, and you download the exports there if a license is active. If a setting changes after a run, the results are marked as stale until you run again. That sequence, load, confirm, configure, preview, run, read, check, and export, is the whole workflow.

Your First Run: The Sample Data

This chapter runs the built-in sample data from an empty page to a finished result. It uses the shipped defaults at every choice, so the figures at the end are the proven figures for that configuration. Follow it once before you load your own file. Every later chapter assumes you have seen each step at least once in this form.

The sample data is a file named maine_county_poverty_sample.csv. It holds sixteen Maine counties, one row per county. The columns are county, direct_pct, sampling_variance, n, snap_rate, unemployment_rate, median_household_income, and population. The column direct_pct is the county poverty rate in percent. Its values run from 7.8 for Cumberland to 19.6 for Piscataquis. The column sampling_variance is in percent squared. The column n is the respondent count for the county, and its values run from forty-two to five hundred twenty-four across the sixteen counties. The covariates are snap_rate, unemployment_rate, and median_household_income. The column population is the county total population from the 2020 Census Redistricting Data, Table P1, and the sixteen figures sum to 1,362,359.

1. Open your browser and go to https://app.maineblock.com/. You should see the header with the project name "New project", the subtitle "Quick mode", the badge "TRIAL PREVIEW · EXPORTS LOCKED", and Step One open below. The step bar shows the four steps in order, each with its number, title, and subtitle. Step One is titled "Upload & Profile" with the subtitle "File, Roles, and Coverage Check".

2. Read the left card on Step One. Its heading is "Start with a File, or Use the Sample Data". Inside the card is a drop zone headed "Drop Your File Here", with the text "Area-level file or respondent-level microdata. The file is read in your browser and is never uploaded." Two buttons stand in the drop zone: "Choose File" and "Use Sample Data".

3. Press "Use Sample Data". The tool loads the sample file itself through the same reading and profiling path that a file you supply would take. A file row appears with the file name maine_county_poverty_sample.csv in bold, the row and column counts, a detected kind, and the words " · Sample data" appended. At the right of the file row you should see "✓ Profiled". The project is renamed to "Sample project – Maine county poverty", and that new name appears in the header title line.

4. Read the four-figure summary strip that appears under the file row. Its labels are "Areas Detected", "Covariates Found", "Design Variables Found", and "Indicator Candidates", each with its count above the label. For the sample file, the strip records sixteen areas, three covariates, one design variable, and no indicator candidates, which matches the file described above.

5. Look at the right column. The coverage card is headed "Coverage: Direct Estimate by Area" for an area-level file, with the right-hand label "Rate". It shows one shaded tile per area, with the area code and the direct estimate as a percent, and a minimum to

maximum legend. You should see sixteen tiles, one for each Maine county. Before any file loads, this card reads: "Load a file and the coverage view shows every area it contains."

6. Read the "Column Roles" card below the coverage card. Its introduction reads: "Detection is automatic; confirmation is yours. Anything marked for review is a best estimate. Correct it here; nothing runs until the roles are confirmed." For the sample, the detection maps county to the area role, direct_pct to the direct estimate role, sampling_variance to the sampling variance role, n to the sample size role, population to the population count role, and snap_rate, unemployment_rate, and median_household_income to the covariate role. The estimate scale for the sample is declared as percent, because the values in direct_pct are percents. Chapter Five explains each role and the confirmation step in full. For this first run, leave the roles as detected.

7. Press "Continue to Model Setup →" at the foot of the "Column Roles" card. The app moves to Step Two and triggers the quality checks and the feasibility preview for this setup.

Step Two is titled "Configure Model" with the subtitle "Covariates, Transform, Estimator". The left card is headed "Model Setup", and its introduction reads: "Fay–Herriot area-level model. Choices update the preview on the right. Nothing is fitted until you run the model."

8. Read the "Outcome" field in the "Model Setup" card. It holds the indicator name, which begins as the detected estimate column name. Under it, a line follows the pattern "Estimate column: {column} · Variance: {column} · Sample size: {column}". For the sample, those slots name direct_pct, sampling_variance, and n.

9. Read the field labeled "Estimates Are Stored As". Its dropdown offers exactly two options: "Proportions (0 to 1)" and "Percents (0 to 100)". For the sample, the percent option is the correct declaration, because direct_pct holds percents. The app proposes a scale from the values, since any estimate above one proposes percent, but it never applies a proposal silently. You confirm the scale here. Leave the percent declaration in place for this run.

10. Read the field labeled "Covariate Candidates". Each candidate has three toggle buttons: "Automatic", "Always In", and "Exclude". The default for every covariate is "Automatic", which means the selection search described in Chapter Seven decides. The three sample covariates, snap_rate, unemployment_rate, and median_household_income, all start on "Automatic". Leave them as they stand.

11. Read the remaining run settings and leave each one on its shipped default for this run. "Run Mode" stands on "Quick: Analytic MSE". "Benchmarking" stands on "On: Align to the Statewide Direct Estimate", with the method row showing "Logit Shift" selected over "Ratio". "Small-Sample Bias Correction" stands on "Off", with the design effect basis row

showing "Statewide Design Effect" selected. "Disclosure Preset" stands on "Standard · Flag CV ≥ 30%". "Transformation (Expert)" stands on "Automatic (Arcsine for Proportions; Declared in the Output)". Chapter Eight explains each setting, what it changes, and when a different choice is warranted.

12. Read the card headed "What Will Run" in the right column. Its introduction reads: "Generated from the choices above, in the form a reviewer would use to specify the same model." With the shipped defaults on the sample data, its seven lines read as follows.

> Fay–Herriot · REML · arcsine (auto) transform Indicators: direct_pct Covariates: snap_rate, unemployment_rate, median_household_income MSE: analytic (Prasad–Rao); bootstrap for adjusted-REML cells, B = 200, seed "sae-studio" Benchmarking: logit shift to the statewide direct estimate Bias correction: off · Design effect: statewide Disclosure: standard preset

13. Read the card headed "Feasibility Preview". On arrival from Step One it is computed automatically. Its four figures carry the labels "Cells That Will Fit", "Cells with Too Few Areas", "Areas with No Direct Estimate", and "Projected Suppressions". For the sample, one cell is projected to fit and no area lacks a direct estimate. Chapter Nine explains the preview and the quality checks in full.

14. Read the card headed "Quality Checks". Its summary follows the pattern "{N} of 20 checks clear · {M} findings. The full report is in Step 4 after a run." A blocking problem stops the run. The sample setup clears the checks that gate a run, so no blocker stands under the run button.

15. Press "Run Model →". While the run is in flight the button reads "Running…" and is disabled. A progress bar and a stage checklist appear, built from the stages "Direct estimates", "Design effects", "Model fitting", and "Disclosure checks", with a step count beside the active stage and a check mark beside each finished stage. On success the app advances to Step Three automatically, showing the first fitted cell.

Step Three is titled "Results" with the subtitle "EBLUP Table and Chart".

16. Read the readout band at the top of Step Three. Its first block is labeled "Statewide Estimate". For the sample run with the shipped defaults, the big figure is 12.1%. The sub-line beneath it reads "Statewide direct 12.1%". The two figures agree because benchmarking was applied. Under benchmarking, the population-weighted model mean equals the population-weighted statewide direct estimate by construction, and that target is 12.1% when formatted to one decimal. The second block, "Areas Fitted", records sixteen areas fitted of sixteen areas in total. The third block, "Benchmarking", reads "Applied" with the sub-line naming the logit shift. The fourth block, "Flagged Areas", records the flagged count for the cell, with a suppressed count in its sub-line when any area is suppressed.

17.

Read the stats row below the readout band. Its four blocks carry the labels "MODEL ESTIMATE RANGE", "MEDIAN CV – MODEL", "BIGGEST CORRECTION", and "AREAS IMPROVED". Each block also carries a small line that gives the direct counterpart, so the model figures are never shown without the direct figures beside them in the same view.

18. Read the estimates table, which has the accessible label "Estimates by area". It has eight columns, in this order: "Area", "n", "Direct %", "Direct SE", "Model (EBLUP) %", "Model CV", "Shrinkage γ", and "Flag". For the sample run, all sixteen counties appear on screen. The sample data is exempt from the ten-area trial cap that applies to your own data, so no license is needed to see all sixteen rows. The covariate selected by the selection search in this proven run is snap_rate. No other result figures for the sample run are stated in this guide, because no other figures are proven for it. Use the table on your own screen for the remaining values.

19. Read the note beneath the table. It reads: "γ (gamma) is the weight the model places on the direct estimate. A low γ means the direct estimate was noisy, and the model estimate rests more on the regression prediction. Select any column heading to sort. Suppressed values remain visible in this workspace; suppression is enforced on exports." Note in particular that the table has no interval column. Intervals appear on this screen only as bars in the chart, and as numbers only in the exports.

20. Read the chart card headed "Direct vs. Model Estimates". Its caption reads, with the color word following the active theme accent: "Gray dots are the direct estimates; orange dots are the model estimates, with the model interval as a bar. Small, noisy areas move the most." The legend reads "Direct estimate", "Model (EBLUP)", and "Model interval". The chart is a forest plot with one row per visible area, ordered by direct estimate with the highest first. Each row shows the model interval as a bar with end caps, the direct estimate as a gray dot, the model estimate as an accent-colored dot, and an arrow from the direct estimate to the model estimate when they differ visibly. At the right, a coefficient of variation strip shows a banded dot beside the model coefficient of variation value.

21. Read the section headed "Reading These Results" below the chart. It holds up to three paragraphs generated from a fixed template, with every number inserted from the run. The first paragraph names the area the model moved most and counts how many areas saw their coefficient of variation fall. The second paragraph gives the model range and the direct range and explains shrinkage in one sentence. The third paragraph always closes with the instruction to report the coefficient of variation alongside every rate, and not to rank model estimates as if they were exact. When flagged or suppressed areas exist, the third paragraph opens with a caution that names both counts.

22. Open Step Four, titled "Diagnostics & Export" with the subtitle "Checks, Interpretation, Files". Read the card headed "Diagnostics", the card headed "Publication Checklist", and the card headed "QC Report". Then read the card headed "What You Can Download with a License". In the trial, the five export buttons show a lock symbol, and clicking

one opens the license dialog instead of producing a file. That is the expected trial behavior, and Chapter Fifteen explains it. The sample run itself is complete on screen without any export.

You have now seen every step once. The chapters that follow take each part in order and in full: the intake and coverage view, the preparation of your own file, the column roles, the model setup for both input paths, the covariate choice, the run settings, the preview and checks, the run itself, the results, the diagnostics, the methods report, the validation record, the exports and the license, the Companion Workbook, the saved project, and troubleshooting.

Step One: Upload and Profile

Step One is titled "Upload & Profile" with the subtitle "File, Roles, and Coverage Check". Its work is intake and confirmation. Nothing is fitted in this step, and nothing runs until the roles are confirmed and you move to Step Two. The layout is two columns. The left column holds the intake card and, below it, the covariate file block. The right column holds the coverage card and, below it, the "Column Roles" card once a file is loaded.

The intake card is headed "Start with a File, or Use the Sample Data". Its introduction paragraph reads as follows, in full.

> "Two kinds of file are accepted. An area-level file has one row per area, with the direct estimate and its sampling variance already computed. A microdata file has one row per respondent, and the tool computes the direct estimates and variances. Accepted formats: .csv, .xlsx, .sav (SPSS), .dta (Stata), .sas7bdat and .xpt (SAS). Column roles are detected automatically; you confirm them before anything runs."

That paragraph is the contract for the step. It names the two file kinds, it names the accepted formats as the screen states them, and it states the division of labor between detection and confirmation. Detection is automatic. Confirmation is yours.

The drop zone sits inside the intake card. Its heading is "Drop Your File Here". The text under the heading reads: "Area-level file or respondent-level microdata. The file is read in your browser and is never uploaded." You can drop a file onto the zone, or you can use one of its two buttons.

1. Press "Choose File" to open the file picker, and select the file from your machine. The picker accepts comma-separated and tab-separated text files, Excel workbooks, SPSS files, Stata files, and SAS files in both the native and transport forms. The reader also treats a text file with the .txt extension as a comma-separated file.
2. Or press "Use Sample Data" to load the built-in sample file. Chapter Two walks that path in full. The sample file is loaded through the same reading and profiling path as a file you supply, so the screens that follow are the same screens you will see with your own data.
3. Wait for the file row to appear. The file name appears in bold, followed by the row count and the column count in the pattern "{rows} rows · {columns} columns", followed by the detected kind in the pattern " · Detected: {kind}". The kind is exactly one of "microdata file", "area-level file", "area covariates file", or "file type not certain". For the sample, the words " · Sample data" are appended. At the right of the file row, the marker "✓ Profiled" confirms that profiling completed.
4.

Read any read issues listed under the file row. The reader reports data-level problems as issues in its own words rather than refusing the file outright. Structural failures refuse the file with a single message. Chapter Eighteen collects the refusal messages and the validation messages, grouped by cause, with the fix for each.

Below the drop zone, a box headed "What Your File Must Contain" states the file requirements in four bullets. The bullets read as follows.

- " **Area** – the small area each row belongs to (county, town, district). One row per area in an area-level file. Required."
- " **Respondent-level file** – the outcome variable, the survey weight, and the design variables (strata and PSU, or replicate weights). The tool computes direct estimates and variances for you."
- " **Area-level file** – one row per area: the direct estimate, its sampling variance (a standard error works too; the tool squares it), and the sample size."
- " **Covariates** – optional area-level predictors the model can borrow strength from, in the same file or a companion file keyed by the same area codes. Population control totals, when available, feed benchmarking, which is on by default."

The closing line of the box reads: "Compare your file with the **Sample File (.csv)** or the **Blank Template (.csv)** and match your columns to them." The two names in that line are buttons. Each button generates its download in the browser. Nothing is fetched from a server.

The first download is labeled "Sample File (.csv)" and produces a file named maine_county_poverty_sample.csv. It is the sample data file described in Chapter Two: the header row county, direct_pct, sampling_variance, n, snap_rate, unemployment_rate, median_household_income, population, followed by sixteen data rows, one per Maine county, in the order Piscataquis, Washington, Somerset, Aroostook, Androscoggin, Penobscot, Oxford, Franklin, Waldo, Kennebec, Hancock, Knox, Lincoln, Sagadahoc, York, and Cumberland. Download it when you want to inspect the exact shape of a valid area-level file in a spreadsheet, or when you want to run the sample outside the built-in "Use Sample Data" path by loading the downloaded file with "Choose File".

The second download is labeled "Blank Template (.csv)" and produces a file named sae_area_level_template.csv. Its entire contents are one header row and nothing else: area, direct_estimate, sampling_variance, n, covariate_1, covariate_2. Use it as the starting shape for a new area-level file. Rename the covariate columns to your own covariate names, add or remove covariate columns as your data require, and fill one row per area beneath the header. A file that matches the template in structure loads and profiles without role corrections in most cases, because the column names carry the meaning the profiler looks for.

After a file loads, a four-figure summary strip appears. Its labels, in order, are "Areas Detected", "Covariates Found", "Design Variables Found", and "Indicator Candidates". Each label sits below its count. Read the strip as a first check on the profiler. If the area count does not match the number of areas in your file, or if a covariate you expected is not counted, the cause is usually a column name or a column type that the profiler read differently from your intent, and the correction belongs in the "Column Roles" card, not in a reload. Reloading the same file reproduces the same detection.

The coverage card sits in the right column. Before any file loads, it reads: "Load a file and the coverage view shows every area it contains." After a file loads, its heading depends on the file kind. For an area-level file the heading is "Coverage: Direct Estimate by Area" and the right-hand label is "Rate". For a microdata file the heading is "Coverage: Respondents by Area" and the right-hand label is "Respondents". The card shows one shaded tile per area. Each tile carries the area code and either the direct estimate as a percent, for an area-level file, or the respondent count, for a microdata file. A minimum to maximum legend explains the shading. Read the coverage view before you confirm the roles. An area that is missing from the view is missing from the file as profiled. An area whose tile stands far outside the range of the others is worth checking in the source file, because the model will treat that value as data.

Below the intake card, a block headed "Covariate File (Optional)" handles the separate covariate table. When no covariate file is loaded, the block reads: "A separate table keyed by the same area codes, when the covariates are not in the main file." A link button labeled "Add a Covariate File" follows. It accepts the same extensions as the main file. When a covariate file is loaded, the block reads in the pattern "{filename} – {N} covariates detected." and offers a link button labeled "Remove". A companion file can also carry the population counts used for benchmarking. If the roles of the companion file include a population count column, those counts are taken into the setup.

Use the covariate file path when your covariates live in a separate table, for example a census extract keyed by county code that you maintain apart from the survey file. The key requirement is exact: the area codes in the companion file must match the area codes in the main file, because the join is made on those codes. An area whose covariate row is missing cannot receive a model estimate, and the validation messages in Chapter Eighteen name that case. If your covariates already sit as columns in the main file, you do not need a companion file at all, and the block can be left empty.

A final point about this step governs the whole guide. Navigation is free. You can open Step Two, Step Three, or Step Four before Step One is complete. Step Two without a profiled file shows a card headed "Load a File First" with the text "Model setup starts once a file has been profiled in Step 1." and a button labeled "Go to Step 1". Step Three without results shows a card headed "No Results Yet". Step Four without a run shows a card headed "No Diagnostics Yet". Visiting a step never marks progress by itself. A step tab receives a done marker only when its work actually exists. The empty states are therefore informative rather than punitive: each one names the step that holds the missing work and offers a button that takes you there.

Preparing Your Own File

This chapter describes what your own file must contain before it is loaded, for each of the two file kinds and for the optional covariate file. It also covers the scale declaration, which is the single preparation decision that cannot be corrected by the profiler, and it closes with a preparation checklist. The refusal and validation messages themselves are collected in Chapter Eighteen. This chapter explains the requirements that produce those messages, so that a file prepared here loads cleanly there.

The area-level file

An area-level file has exactly one row per area. Duplicate area codes are a validation failure: the message names the codes and states that an area-level file must have exactly one row per area, and that duplicates must be combined or removed. Prepare the file so that each area appears once.

Each row must carry four things, plus covariates if you have them.

First, the area code. It can be a name, such as a county name, or a code, such as a census geography code. It must be present on every row, and it must match the codes in a companion covariate file exactly if you use one.

Second, the direct estimate. It is a proportion, stored either as a proportion on the scale from zero to one or as a percent on the scale from zero to one hundred. No other kind of value is accepted on this path. A mean cannot be run. A total cannot be run. If the only figure available for an area is a mean or a total, this tool cannot model it, and no role assignment in Step One changes that. Estimates outside the declared range are flagged by validation, with a message that names the areas and the range that was violated. An area with no direct estimate is excluded from the model fit and named in the QC report. Its covariates still inform predictions if they are provided, but the area itself receives no fitted value from a fit it did not enter. The feasibility preview in Step Two names such areas in advance, so an exclusion should never come as a surprise at run time if the preview was read.

Third, the sampling variance, or a standard error in its place. Supply one of the two, not neither, and preferably not both. If both a variance column and a standard error column are mapped, the tool uses the variance column and reports that choice as a validation message. If neither is mapped, the run is blocked in Step Two with a message that asks you to confirm the sampling variance or standard error column in Step One. A variance or a standard error cannot be negative, and validation names any column that holds negative values. A value of zero claims perfect precision and will distort the model, and validation warns about zero values in the same style. Check zero variances in the source file before loading. In practice, a zero usually means a missing variance that was filled with zero during file preparation, and the correction belongs in the source.

Fourth, the sample size, carried in the column usually named n. Every area needs its respondent count. Validation reports the count of areas for which the sample size column is missing or not numeric. The sample size is not an optional refinement. It enters the transformed sampling variance used by the model, it enters the effective sample size used by the disclosure rules, and it enters the fit itself. A file without a trustworthy count per area is not ready for this tool.

Covariates are optional and are covered in full in Chapter Seven. In the file, each covariate must be numeric. Validation reports non-numeric values by column and reports missing covariate values by area. An area without a complete covariate row cannot receive a model estimate, so completeness of the covariate columns matters as much as completeness of the estimate column when covariates are in use.

A population count column is optional and valuable. When population counts are available, they weight the statewide figure, they provide the benchmark target basis described in Chapter Eight, and they enter the population-weighted calculations that the methods report records. Use counts from a stated source and vintage, and record that source outside the tool, because the file itself carries no provenance field. The sample file uses the 2020 Census Redistricting Data for its population column, which is a good pattern to follow: one source, one vintage, named in your own project notes.

The microdata file

A microdata file has one row per respondent. The tool computes the direct estimates and variances from the rows, under the design that the column roles declare. Prepare the file so that the following columns exist and are complete.

The area column gives the small area each respondent belongs to. Respondent rows with no area code are excluded, and validation names the count and the rows. Every respondent who should enter an estimate needs an area code.

The outcome variable is the variable from which indicators are built. In Step Two, each indicator is defined by a rule that marks every distinct value of its column as yes, no, or missing. The cleanest preparation is to recode binary outcomes to zero and one in the source file before loading. Validation reports an outcome column that contains values other than zero and one and asks for an explicit recode rule before running. That message is a preparation instruction, not a formatting complaint: decide the recode on substantive grounds, apply it in your source or through the rule builder in Step Two, and keep a record of it, because the methods report records the rule that was used.

The survey weight column gives each respondent a numeric, positive weight. Every respondent needs a numeric weight. Validation counts rows with a missing or non-numeric weight, and it counts rows with a weight of zero or less, stating that analysis weights must be positive. A weight file that arrives with structural zeros or with missing weights for a subgroup needs correction in the source, because the tool does not impute weights.

The design variables describe how the sample was drawn. Provide strata and primary sampling unit columns, or provide replicate weights. Frame primary sampling unit counts, when you have them, are carried by their own role and are subject to their own rules: a frame count must be a positive whole number, the same on every row of its stratum, and no smaller than the number of primary sampling units sampled in that stratum. Each violation has its own validation message. Strata with no frame count are treated as sampled with replacement, and that treatment is stated as an issue rather than applied silently, so you will see it in the QC report and can judge whether it matches your design.

Subgroup variables, when you want estimates split by a characteristic, should be columns with a manageable number of levels and no level so sparse that it cannot support an estimate. Step Two proposes a collapse for sparse levels, and Chapter Six describes that proposal and your choice to accept it or to use the original levels. Preparation cannot remove sparseness, but it can remove accidental levels: stray codes, trailing spaces in text values, and inconsistent capitalization all create levels that are artifacts of the file rather than groups in the population. Clean those in the source before loading.

The covariate file

A covariate file is a separate table keyed by the same area codes as the main file. It follows the same format list as the main file. Prepare it with one row per area, the area code column first in your own working order, and one numeric column per covariate. If the file also carries population counts, include the population count column there and assign its role when the file is profiled. After loading, the setup notes the source in the line "Covariates come from {filename}." under the covariate list in Step Two, so a later reader of a screenshot or a methods report can tell which path supplied the covariates.

The scale declaration

The scale declaration is the decision this chapter most wants you to make deliberately. In Step Two, the field labeled "Estimates Are Stored As" offers "Proportions (0 to 1)" and "Percents (0 to 100)". The app proposes a scale from the values: any estimate above one proposes percent. The app never applies its proposal silently. You confirm the declaration, and the declared scale governs how the estimates and their variances are read.

The risk is asymmetric. A file of percents declared as proportions holds values up to one hundred, and validation flags estimates outside the zero to one proportion range. That error announces itself. A file of proportions declared as percents holds values below one, all of which sit inside the zero to one hundred percent range, so no range check can catch it, and the model would read a proportion of 0.12 as a percent of 0.12, which is a proportion of 0.0012. That error is silent. The protection is procedural: state the scale in your file preparation notes before you load, check the coverage view against a figure you know from the source, and read the "Outcome" line in Step Two, which names the estimate column, before you run. The sample file declares percent, because direct_pct is in percent and its variance is in percent squared. If you supply a standard error instead of a variance, its scale follows the estimate scale: a standard error for a percent is in percentage points, and the tool squares it on that basis.

Preparation checklist

Before loading your own file, work through the following list.

- One row per area in an area-level file, or one row per respondent in a microdata file, with no duplicate area rows in the area-level form.
- An area code on every row, matching any companion file exactly.
- Direct estimates that are proportions, on one declared scale, with the scale written in your project notes.
- A sampling variance for every area, or a standard error for every area, with no negative values and no zeros that are really missing values in disguise.
- A respondent count for every area in an area-level file, and a positive numeric weight for every respondent in a microdata file.
- Design columns that match the design that was actually used: strata and primary sampling units, or replicate weights, plus frame counts where you have them.
- Covariates that are numeric and complete for every area that should receive a model estimate.
- Outcome values recoded to zero and one, or a written rule ready for the rule builder.
- Subgroup columns cleaned of artifact levels.
- Population counts from one named source and vintage, when counts are used.

A file that passes this list will still be checked by the profiler, the validation pass, the twenty quality checks, and the feasibility preview. Those checks are described where they appear on screen. Preparation does not replace them. It makes them quick.

Confirming the Column Roles

The "Column Roles" card appears in the right column of Step One once a file is loaded. It is the confirmation step on which the whole run depends, because every later screen reads the roles rather than the raw columns. A column assigned the wrong role produces a wrong run that can look entirely plausible. This chapter names all fifteen roles, explains the confidence display and the review tag, and walks the confirmation itself.

The card introduction reads: "Detection is automatic; confirmation is yours. Anything marked for review is a best estimate. Correct it here; nothing runs until the roles are confirmed."

Each column of the file gets one row in the card. The row shows the column name in bold, with its file-supplied variable label under it when the format carries one. Statistical formats such as SPSS and Stata files often carry variable labels, and those labels appear here and help confirm that a cryptically named column is the variable you intend. Beside the name sits a dropdown holding the assigned role. At the right of the row sit the detection confidence as a percent, for example "92%", and the tag "review" when the detection needs review or is contested. A row is also highlighted while the detected role still stands on a row that needs review. Treat every highlighted row as a question the profiler is asking you, and answer it by confirming the role or by changing it.

The dropdown first option is "Unassigned". An unassigned column plays no part in the run. The fifteen role options, in the exact order the dropdown presents them, are as follows.

1. "Area"
2. "Area Name"
3. "Survey Weight"
4. "Stratum"
5. "PSU / Cluster"
6. "Frame PSU Count"
7. "Respondent ID"
8. "Direct Estimate"
9. "Sampling Variance"
10. "Standard Error"
11. "Sample Size (n)"
12. "Population Count"
13. "Indicator (Outcome)"

14. "Subgroup Variable"

15. "Covariate"

The roles divide by file kind. An area-level setup needs "Area", "Direct Estimate", one of "Sampling Variance" or "Standard Error", and "Sample Size (n)". It can also use "Area Name" when a separate name column exists beside a code column, "Population Count" for weighting and benchmarking, and "Covariate" for each predictor column. A microdata setup needs "Area", "Survey Weight", the design roles "Stratum" and "PSU / Cluster" or replicate weight columns as the profiler presents them, "Indicator (Outcome)" for each outcome column, and "Subgroup Variable" for each split column. "Frame PSU Count" carries the population cluster counts used for a finite population correction. "Respondent ID" marks an identifier column so that it is treated as a label rather than as data.

Three role decisions deserve deliberate attention.

The first is the variance decision. Assign "Sampling Variance" to a column of variances and "Standard Error" to a column of standard errors, and do not assign both unless your file genuinely carries both and you intend the variance column to govern. When both roles are mapped, the tool uses the variance column and reports that choice. When neither is mapped on the area-level path, the run is blocked in Step Two. The most damaging version of this error is a standard error column assigned the variance role, because the values then enter the model unsquared and far too small, which makes the direct estimates look far more precise than they are and distorts the shrinkage for every area. If you are unsure which quantity a column holds, compare its size with the estimate scale. For a percent near twelve with a few hundred respondents, a standard error is a small number of percentage points, and a variance is that number squared. The sample file is a useful reference: its sampling_variance value of 30.25 is 5.5 squared, which is the pattern a variance column follows.

The second is the scale-bearing decision between "Direct Estimate" and "Indicator (Outcome)". An area-level file assigns "Direct Estimate" to its estimate column. A microdata file assigns "Indicator (Outcome)" to its outcome column. Assigning the indicator role to a column builds its indicator rule automatically, with a yes, no, and missing value mapping derived from the column values. The rule is edited in Step Two, as Chapter Six describes. Assigning the direct estimate role to a respondent-level column, or the indicator role to an area-level column, mixes the two paths and produces blockers or nonsense. If the classification line described below names a file kind that does not match your file, the cause is usually a role of this kind standing on the wrong column.

The third is the population decision. Assign "Population Count" to a genuine count of persons per area, not to a rate, not to a sample count, and not to a weight total that you have not checked. Population counts weight the statewide figure and set the benchmark target basis. A wrong population column does not stop the run. It moves the statewide figure and the benchmarking in ways that are hard to see from the results screen alone. The methods report names the covariate and population sources, but the correctness of the column is your confirmation, and this card is where you make it.

Under the role list, a line follows the exact pattern: "Classification: {kind label} ({confidence}% confidence). Preset: {preset name}." The kind label restates the detected file kind, the confidence restates the profiler confidence in the classification, and the preset name comes from the profiler. Read this line as the summary of the card. If the classification says "microdata file" for a file you prepared as area-level, stop and correct the roles before you continue. If the classification says "file type not certain", the run will ask for the missing confirmations through its blockers, and the fastest path is to assign the required roles by hand in this card until the classification resolves.

At the foot of the card stands the button "Continue to Model Setup →". Pressing it moves to Step Two and triggers the quality checks and the feasibility preview for the confirmed setup. Pressing it is the confirmation act. You can return to Step One at any time and change a role, because navigation is free, but a role change after a run makes the existing results stale, and the app marks them as such in Step Two and Step Three. The practical order is therefore: confirm the roles carefully once, read the preview and the checks in Step Two, correct roles in Step One only if the preview or the checks expose an error, and then run.

A confirmation routine that catches most errors takes less than a minute. Read the rows top to bottom and ask, for each row, whether the role names the quantity the column actually holds. Check that exactly one column holds "Area". Check that the estimate role and the variance or standard error role sit on different columns. Check that "Sample Size (n)" sits on a count column and "Population Count", if used, sits on a population column and not on the sample count. Check that every covariate you intend to offer the model holds the "Covariate" role, and that no column you intend to exclude from the model holds it by accident. Check that every "review" tag has been answered. Then read the classification line, and only then press "Continue to Model Setup →".

Step Two: Setting Up the Model

Step Two is titled "Configure Model" with the subtitle "Covariates, Transform, Estimator". With no file profiled, it shows a card headed "Load a File First", the text "Model setup starts once a file has been profiled in Step 1.", and a button labeled "Go to Step 1". With a file loaded, the layout is two columns. The left column holds one card headed "Model Setup". The right column holds, stacked in order, the cards headed "What Will Run", "Feasibility Preview", and "Quality Checks", and, after a run, a card headed "Fit Summary". This chapter covers the "Model Setup" card itself. Chapter Seven covers the covariate choice, Chapter Eight covers the run settings that sit below the setup fields in the same card, and Chapter Nine covers the three cards in the right column.

The "Model Setup" card introduction reads: "Fay–Herriot area-level model. Choices update the preview on the right. Nothing is fitted until you run the model." Two consequences follow. First, every choice in the card updates "What Will Run" and can change the feasibility preview, so the right column is part of the setup screen and should be read while you set up, not after. Second, no choice fits anything by itself. Fitting happens only when you press "Run Model →", which Chapter Ten covers.

The card presents one of two paths, depending on the file kind confirmed in Step One.

The area-level path

For an area-level file, the card opens with the field labeled "Outcome". The field is a text box holding the indicator name. It begins as the detected estimate column name and it is editable. The name you leave in the box is the name the results, the exports, and the methods report use for the outcome, so give it the name a reader should see, in plain words, without internal codes. Under the box, a line follows the exact pattern: "Estimate column: **{column}** · Variance: **{column}** · Sample size: **{column}** ". The variance slot shows the standard error column instead when the standard error role is the one mapped. Read this line before anything else in the card. It restates the three role assignments the fit depends on, and a wrong column named here is a wrong run, however sound the later settings are.

Below it stands the field labeled "Estimates Are Stored As", with a dropdown offering exactly two options: "Proportions (0 to 1)" and "Percents (0 to 100)". Chapter Four explained the declaration and its asymmetric risk. The app proposes a scale from the values, since any estimate above one proposes percent, but it never applies the proposal silently. The value standing in this dropdown when you run is the declaration. For the sample data, the correct standing value is "Percents (0 to 100)".

No indicator rule and no subgroup control appears on this path, because the area-level file already holds one estimate per area for one outcome. If your area-level work needs estimates for a second outcome, prepare a second file or a second project for that outcome and run it separately. The project holds one area-level outcome at a time.

The microdata path

For a respondent-level file, the card opens with the field labeled "Indicators to Estimate". One block stands for each indicator column, meaning each column that holds the "Indicator (Outcome)" role from Step One.

Each block shows a checkbox with the indicator label, or the column name, in bold. Under the name, a rule summary follows the exact pattern: "Yes: {values} · No: {values} · Missing: {values}". When the rule is incomplete, the words " · some values unmarked" are appended. When no rule exists for the column, the block shows the text "No rule yet. Assign the indicator role in Step 1." and its checkbox is disabled. When no indicator columns exist at all, the field shows the text: "No indicator columns yet. Assign the indicator role to a column in Step 1."

Select the indicators you want estimated by checking their boxes. At least one indicator must be selected, or the run is blocked with the message "Select at least one indicator in Step 2." Each selected indicator becomes its own model cell in the run, or several cells when subgroup variables are also selected.

Each block carries a link button labeled "Edit Rule", which changes to "Close the Rule Builder" while the builder is open. The rule builder introduction reads: "Mark every value as yes, no, or missing. Unmarked values are treated as missing and counted in the run report." The builder lists each distinct value of the column, the first thirty values, each with its row count. Each value gets three toggle buttons labeled "Yes", "No", and "Missing". Mark every value, then press "Apply Rule".

Two details of the builder matter. First, unmarked values are not an error that stops you. They are treated as missing and counted in the run report, so a partly marked rule runs with a smaller effective base than the file suggests, and the report tells you by how much. Complete the rule unless the unmarked values are genuinely values that should be excluded as missing. Second, when the column has more than thirty distinct values, a note appears in the exact pattern: "This column has {N} distinct values; only the first thirty are shown. A rule is complete only when every value is marked." A column with that many distinct values is usually a measured variable rather than an indicator source. If you meet this note, reconsider the variable. Recode it to a small set of substantive values in the source file, or choose a different column as the indicator source. A rule that cannot be completed in the builder cannot be verified in the builder either, and an unverified rule is a weak basis for a published estimate.

Below the indicators stands the field labeled "Subgroup Breakdown". One block stands for each subgroup variable, meaning each column that holds the "Subgroup Variable" role.

Each block shows a checkbox with the variable name and a collapse summary in one of these exact forms: "No sparse levels to collapse.", "{N} levels, none sparse.", or "{N} sparse level(s) combined into “{label}” (proposed).", with "(accepted)" standing in place of "(proposed)" after you accept. When a proposal combines at least one level, two link

buttons appear in turn: "Accept the Proposed Collapse" before acceptance, and "Use the Original Levels Instead" after acceptance. When no subgroup variables exist, the field shows the text: "No subgroup variables detected. Assign the subgroup role in Step 1."

A sparse level is a level with too few respondents to support a stable direct estimate for its cell. The proposal combines sparse levels into a combined level with a stated label. Accepting the proposal means the run estimates the combined group rather than the original thin levels. Using the original levels instead means each level becomes its own cell, with the risk that a thin cell falls below the fit minimum or produces an estimate whose precision is poor. Neither choice is hidden: the collapse state is recorded, and the reproducibility package carries the collapse information when it applies. Decide on substantive grounds. Combine levels that belong together in the subject matter, such as adjacent age bands or neighboring districts with the same program rules. Resist combining levels that merely look similar in the data, because the combined estimate will be quoted for the combined group and a reader will interpret it as that group.

Each selected subgroup variable crossed with each of its levels becomes its own separately fitted model cell in the run. A run with one indicator, one subgroup variable with four levels, and the overall cell therefore produces several cells, and Step Three lets you switch among them with its result set dropdown. Cell counts multiply quickly. Two subgroup variables with several levels each can produce a run whose cells are each thin. The feasibility preview counts the cells that will fit before you run, and Chapter Nine explains how to read it. Prefer one subgroup breakdown per run, and run a second breakdown as a second run in the same project by changing the selection, or as a separate project when both breakdowns must be preserved side by side.

Choosing Covariates

Covariates are the area-level predictors the model can borrow strength from. In the Fay–Herriot model, the regression prediction for an area is built from its covariate values. An area whose direct estimate is noisy rests more on that prediction, so the quality and the relevance of the covariates directly govern the quality of the estimates for the areas that need the model most. This chapter covers the control in Step Two, the selection method behind its automatic setting, the intercept-only case, and the practical choice of which covariates to offer.

In the "Model Setup" card, the field is labeled "Covariate Candidates". One row stands for each candidate covariate, meaning each column that holds the "Covariate" role in the main file, plus each covariate detected in a companion covariate file when one is loaded. Each row carries three toggle buttons with the exact labels "Automatic", "Always In", and "Exclude". The default for every covariate is "Automatic".

The three settings mean the following.

- "Automatic" means the selection search decides. The covariate may enter the final model or may be left out, according to the criterion described below.
- "Always In" means the covariate is locked in. It enters the final model regardless of the selection criterion. Use it for a covariate that the subject matter requires, for example a program participation rate that defines the population at risk, when its absence would make the model hard to defend even if the criterion would drop it.
- "Exclude" means the covariate never enters. Use it for a covariate you no longer trust, for a covariate whose vintage does not match the direct estimates, or for a covariate you want to hold out for a comparison run.

When the covariates come from the companion file, a line under the list reads: "Covariates come from {filename}." When there are no candidates at all, the field shows the exact text: "No covariate candidates. The model will fit an intercept-only model; adding area-level covariates usually improves it."

The selection search behind "Automatic" uses the Bayesian information criterion, known as BIC. The search considers combinations of the automatic covariates, with any "Always In" covariates present in every combination, and selects the combination with the best BIC balance of fit and parsimony for the cell being fitted. Selection is per cell. In a run with subgroup cells, different cells can select different covariates, and the methods report records the selected set by cell under its subheading "Covariates selected, by cell". The Step Two "Fit Summary" and the Step Four diagnostics address the selected cell you are viewing, so check the selection per cell rather than assuming one selection governs the run.

Two screens guard the search. A pair of covariates that are very highly correlated cannot both support stable coefficients, and the model fitting applies a correlation screen at an absolute correlation above 0.90 and a variance inflation screen above a factor of 10, as the methods report records. In practical terms, offering two covariates that measure nearly the same thing, such as two income measures from the same source and year, usually results in one of them carrying the signal and the other being dropped or screened. That outcome is correct behavior. Choose the better measured of the pair as "Always In" and set the other to "Exclude" if you want the choice to be yours rather than the screen deciding it.

In the proven sample run from Chapter Two, all three covariates stood on "Automatic", and the covariate selected was snap_rate. The unemployment rate and the median household income were offered and were not selected for that cell under the shipped defaults. That result illustrates the search doing its work: offering a covariate is not a promise that it enters, and a covariate left out of one cell is not a judgment that it is unrelated to the outcome in general. It is a judgment that, for that cell, on that criterion, the selected set fit better without it.

The intercept-only case deserves a plain statement. With no covariates selected and none locked in, the model reduces to a common mean with area effects. Shrinkage then pulls noisy areas toward the overall mean rather than toward an area-specific prediction. Such a fit can still improve on the direct estimates when the areas are broadly similar, but it discards all area-level information beyond the direct estimates themselves. The on-screen text says that adding area-level covariates usually improves the model, and that wording is accurate. If you have any credible area-level predictor measured for every area on the same vintage as the estimates, offer it.

Practical guidance for the choice itself is short. Offer covariates that are measured for every area without missing values, because an area without a complete covariate row cannot receive a model estimate. Offer covariates from a source and a year that match the direct estimates as closely as the available data allow. Offer a small number of strong candidates rather than a long list of weak ones, because each added candidate enlarges the search and raises the chance that a noise pattern in a small set of areas selects a covariate that will not generalize. Set "Always In" sparingly, for one or two covariates at most, since locking in a weak covariate forces every cell to carry it. Set "Exclude" decisively when a covariate is superseded, because leaving a superseded version on "Automatic" beside its replacement invites the correlation screen to choose between them on arithmetic grounds alone.

After a run, verify the selection before you interpret the results. The "Fit Summary" card in Step Two reports the fit for the first fitted cell. The "Diagnostics" card in Step Four reports, for the selected cell, the item "Model Fit: Covariates" with the count of covariates selected and, when available, the Fay–Herriot coefficient of determination. The coefficients block in the same card lists the intercept and each selected covariate with its coefficient and standard error on the fitting scale. Chapter Twelve walks those displays. If the selection

surprises you, the comparison to run is a second run with the surprising covariate set to "Exclude", or with a required covariate set to "Always In", followed by a comparison of the feasibility figures, the median coefficient of variation, and the biggest correction in Step Three. Change one setting at a time so that the comparison has a single cause.

Run Settings

Below the outcome, indicator, subgroup, and covariate fields, the "Model Setup" card holds five run settings: run mode, benchmarking, small-sample bias correction, disclosure preset, and transformation. This chapter takes each in the order it appears on screen, states its default, and explains what changes when you choose differently. Two run quantities have no user controls at all, and the chapter states that plainly where it belongs: the bootstrap replicate count and the run seed, and the confidence level.

Run Mode

The field is labeled "Run Mode", with two toggle buttons labeled "Quick: Analytic MSE" and "Full: Bootstrap MSE (Slower)". The default is Quick.

The run mode chooses how the mean squared error of the model estimates is computed, and therefore how the reported precision, the coefficients of variation, and the interval widths are derived. In Quick mode, the mean squared error is analytic, using the Prasad–Rao formula. In Full mode, the mean squared error is computed by parametric bootstrap, which refits the model on simulated data many times and measures the spread of the results. Full mode is slower, as its label states, and the difference grows with the number of areas and cells.

One exception qualifies the Quick label, and the guide states it without qualification elsewhere: adjusted-REML cells use the parametric bootstrap in every mode. When a cell is fitted by the adjusted restricted maximum likelihood method, which is the boundary fallback described in the methods report, its mean squared error comes from the parametric bootstrap with two hundred replicates, in Quick mode and in Full mode alike. The "What Will Run" block states this exception on its MSE line in Quick mode, in the form: "MSE: analytic (Prasad–Rao); bootstrap for adjusted-REML cells, B = 200, seed "sae-studio"". In Full mode the same line reads in the form: "MSE: bootstrap, B = {B}, seed "{seed}"".

The bootstrap replicate count, B, is two hundred. The run seed is the text "sae-studio". Neither has a user control anywhere in the app. Both appear in the "What Will Run" text, and the seed also appears in the run manifest and in the publication checklist, so a run can be reproduced from its records even though neither value can be changed on screen. The confidence level for intervals is ninety-five percent, and it also has no user control anywhere in the app. Intervals in the chart, in the exports, and in the methods report are ninety-five percent intervals, and no setting in Step Two changes that level.

A practical consequence of the two modes is recorded in the methods report and repeated here because it affects reading: the same data can land on opposite sides of the thirty percent flag threshold in Quick mode and in Full mode. The coefficient of variation of a given area can sit just below the threshold under one mean squared error method and just above it under the other. When an area sits near a threshold, state the mode beside the

figure when you quote it, and do not treat a threshold crossing that appears only in one mode as a change in the underlying data. Chapter Fourteen returns to the precision record behind this caution.

Choose Quick mode for exploration, for setup correction, and for most production runs. Choose Full mode when the run is the final run for publication and the added computation time is acceptable, or when a reviewer asks for bootstrap precision specifically. Record the mode in your project notes either way, because the mode is part of the method.

Benchmarking

The field is labeled "Benchmarking", with two toggle buttons labeled "Off" and "On: Align to the Statewide Direct Estimate". The default is On. When benchmarking is on, a second toggle row appears with the method options "Logit Shift" and "Ratio". The default method is "Logit Shift".

Benchmarking aligns the model estimates to the statewide direct estimate. When it is applied, the population-weighted model mean equals the benchmark target by construction. The target is the population-weighted statewide direct estimate when population counts are available. The logit shift method applies the alignment as a shift on the logit scale, which keeps every adjusted estimate inside zero to one and moves estimates near the middle of the scale differently from estimates near the ends. The ratio method applies the alignment as a proportional adjustment. For proportions, the logit shift is the safer default, because a ratio adjustment can press an estimate that is already high toward the upper bound without regard to the bound, while the shifted form respects the scale on which the proportions live. Keep "Logit Shift" unless a stated external requirement specifies ratio benchmarking for comparability with an earlier published series.

Benchmarking is a free, on-screen model setting. It is on by default, it runs in the trial exactly as it runs with a license, and it is not gated in any way. The license adds other things, which Chapter Fifteen lists exactly, and benchmarking is not among them.

Read the benchmarking state in the results. The readout band in Step Three has a block labeled "Benchmarking" that reads "Applied" with a sub-line naming the shift method, or "Off", or "Skipped", or a dash when no value applies. The methods report has a section titled "Benchmarking" that states the method and the target when benchmarking was applied, or states that it was not applied. Chapter Fourteen records a validation finding that belongs with this setting: when benchmarking was not applied in a flagship-scale validation study, the unbenchmarked model estimates averaged 1.48 percentage points below the true values. That finding is a reason the default stands On, and it is a reason to treat an unbenchmarked run as a deliberate choice that needs a stated reason in your notes, not as a neutral variant.

Small-Sample Bias Correction

The field is labeled "Small-Sample Bias Correction", with two toggle buttons labeled "Off" and "On: Correct the Transformed Inputs". The default is "Off". Directly beneath it, a second toggle row sets the design effect basis, with the options "Pooled Design Effect" and "Statewide Design Effect". The default is "Statewide Design Effect".

The correction addresses a known small-sample effect in the transformed inputs. The model fits on a transformed scale, and the transformation of a proportion estimated from a small number of respondents is biased in a way that depends on the sample size and the design effect. The correction adjusts the transformed inputs before fitting. The design effect basis chooses which design effect enters that adjustment and related calculations: a pooled design effect estimated across the design, or a statewide design effect. The basis row is present as part of this setting group, and its standing value is recorded in the "What Will Run" block on the line that reads in the form "Bias correction: off · Design effect: statewide", with the words changing to match the choices in force.

Leave the correction off for a first run and for comparability with the proven sample configuration. Turn it on when your areas have small respondent counts and a stated design effect basis, and when you intend to report that the correction was applied. The methods report records whether the transform bias correction was applied and which design effect basis the run used, so the choice is auditable either way. Chapter Fourteen records the related totals finding from the validation work: population-weighted model totals averaged 0.23 percentage points below the true total across the validation populations, because transformed values sit slightly low at small samples when prevalence is under fifty percent. That finding describes the behavior the correction addresses. It does not promise that the correction removes the difference in any given run.

Disclosure Preset

The field is labeled "Disclosure Preset", with four toggle buttons in this order: "Standard · Flag CV ≥ 30%", "Strict · Flag CV ≥ 20%", "SAE Standard", and "Off". The default is "Standard · Flag CV ≥ 30%".

A disclosure preset is the set of reliability rules that flag or suppress estimates. Flagging marks an estimate as unstable on screen and in exports. Suppression withholds the value in every export, while the value remains visible in the workspace on screen. That division is stated in the table note in Step Three and holds throughout: suppression is enforced on exports, and suppressed values stay visible in the app so that you can see what was withheld and why. The flag cell in the results table reads "Suppressed: Do Not Publish" for a suppressed area and "Flagged: Unstable" for a flagged area, with the suppression reason available as the flag tooltip when a reason is recorded.

The four presets are as follows.

"Standard" is built on the published data presentation standards for proportions by Parker and others in 2017, the standards used by the National Center for Health Statistics. Its rules, as the app applies them, are: suppress when the effective sample size, defined as the respondent count divided by the smoothed design effect, is below thirty; suppress when the interval is at least 0.30 wide, or more than 0.05 wide and more than one hundred thirty percent of the estimate; a coefficient of variation ladder that flags above thirty percent and suppresses above fifty percent; a minimum of ten respondents; and a minimum of one primary sampling unit per area.

"Strict" keeps the same coefficient of variation ladder shape with the flag rung lowered to twenty percent, while suppression stays at fifty percent. The structural proportion rules of the Standard preset do not apply under Strict, because Strict runs as a custom preset built on the ladder. Choose Strict when your publication standard flags earlier than thirty percent and your governing rules are coefficient of variation rules rather than the published proportion standards.

"SAE Standard" is the model-based preset that carries the same ladder numbers: a minimum of ten respondents, a flag at a coefficient of variation of thirty percent, suppression at fifty percent, and a minimum of one primary sampling unit per area, with its own structural constants governing the rest. Choose it when your review process specifies the tool model-based preset by that name.

"Off" sets the thresholds so that no finite coefficient of variation or count can trip them, with a minimum respondent count of zero. Choose it only for internal exploration. A run with disclosure checks off produces exports without suppression enforcement from a preset, and the publication checklist in Step Four records the state as disclosure checks off, so the choice is visible to any later reader of the run.

When "Standard" is selected, a paragraph appears verbatim under the row. The guide carries its substance here, because it changes how the preset behaves at small area sample sizes. The paragraph reads as follows.

> "The Standard preset is built on published data presentation standards for proportions, with two departures from the published standard: interval widths for direct estimates are the logit interval widths, not the standard's Clopper-Pearson widths, and the preset is also applied to model-based estimates, which the standard does not cover. At about 20 respondents per area, this preset suppresses nearly all exported estimates (0.93% published with frame counts supplied, 0.00% without)."

Read the last sentence with care. At about twenty respondents per area, which is the flagship scale in the validation work, the Standard preset published 0.93 percent of area estimates when frame counts were supplied and 0.00 percent when they were not. At smaller scale in the same validation work, the same preset published about ninety percent

of area estimates. The consequence therefore belongs to the sample size per area, not to the preset at every scale. If your areas average near twenty respondents and you select "Standard", expect nearly every export value to be suppressed, and plan the publication around that fact or choose a different preset on stated grounds. Chapter Fourteen gives the fuller record, and Chapter Twelve shows where the counts appear in the publication checklist.

The fuller account in the methods report names the Korn-Graubard interval as the width basis in the published standard for direct estimates, where the Step Two paragraph names Clopper-Pearson widths. The two shipped texts describe the same departure in different words. The substance is the same in both: the tool uses logit interval widths for direct estimates where the published standard uses its own width basis, and the tool applies the preset to model-based estimates, which the standard does not cover.

Transformation (Expert)

The field is labeled "Transformation (Expert)", with a dropdown offering exactly three options: "Automatic (Arcsine for Proportions; Declared in the Output)", "Arcsine", and "Linear (no transformation)". The default is the Automatic option.

The transformation is the scale on which the model is fitted. Proportions have sampling variances that depend on the proportion itself, and the arcsine square root transformation stabilizes that dependence, which is why it is the working scale for proportion outcomes in this model. Under Automatic, the resolved transformation is the arcsine transformation for every outcome the app handles, because all outcomes are proportions, and the resolved transformation is declared in the output. Selecting "Arcsine" directly gives the same working scale with the choice stated by you rather than by the automatic rule. Selecting "Linear (no transformation)" fits on the proportion scale itself. That choice removes the variance stabilization and can produce fitted values and intervals that behave poorly near zero and one, although the reported intervals still lie inside zero to one by construction of the reporting step. Keep the Automatic setting unless a replication requirement names a specific transformation. Whatever you choose, the choice is recorded: the "What Will Run" first line names the transformation, the publication checklist records it in its methods metadata line, and the methods report states the transformation and the arcsine construction in its model section.

One sentence collects the defaults for reference. Run mode Quick, transformation Automatic resolving to arcsine, benchmarking On with the logit shift method, bias correction Off with the statewide design effect basis, and disclosure preset Standard. Those are the shipped defaults, they are the configuration of the proven sample run, and they are the configuration the Companion Workbook mirrors. Change them one at a time, and read the "What Will Run" block after each change.

Before You Run: Preview and Checks

The right column of Step Two holds three cards that tell you what a run will do before you spend it, and a fourth card that summarizes a run after it completes. This chapter covers all four: "What Will Run", "Feasibility Preview", "Quality Checks", and "Fit Summary". Reading them in order is the cheapest quality step in the workflow. The preview and the checks cost no fitting, and they expose role errors, missing direct estimates, thin cells, and projected suppressions while correction is still a settings change rather than a rerun.

What Will Run

The card is headed "What Will Run". Its introduction reads: "Generated from the choices above, in the form a reviewer would use to specify the same model." Its body is a fixed seven-line specification block generated from the current choices. With the shipped defaults on the sample data, the block reads as quoted in Chapter Two. Its line rules are as follows.

The first line follows the form "Fay–Herriot · REML · {transform} transform". The transform slot reads "arcsine (auto)" when Automatic is chosen, and otherwise the chosen name. The second line begins "Indicators:" and lists the area-level outcome name, or the selected microdata indicator identifiers, or the words "none selected" when no indicator is selected. The third line begins "Covariates:" and lists every candidate covariate name, or the words "intercept only" when there are no candidates. Note that this line lists candidates offered, not covariates selected, because selection happens during fitting. The fourth line begins "MSE:" and gives the mean squared error route for the mode in force, including the adjusted-REML bootstrap exception in Quick mode, with the replicate count and the seed. The fifth line begins "Benchmarking:" and reads either a shift form naming the logit or ratio method and the statewide direct target, or the word "off". The sixth line begins "Bias correction:" and follows the form "Bias correction: off · Design effect: statewide", with its words following the choices in force. The seventh line begins "Disclosure:" and reads "standard preset", "strict (flag CV ≥ 20%)", "SAE Standard preset", or "off", matching the preset in force.

Read the block as a reviewer would. If any line names something you did not intend, an indicator you did not mean to select, a covariate list that omits a candidate you expected, benchmarking off when you expected it on, a disclosure state that does not match your publication rules, then correct the setting in the left card and read the block again. The block is also the shortest accurate description of the run for your project notes. Copy its substance into your notes before you run, and the methods report later confirms the same specification from the completed run.

Feasibility Preview

The card is headed "Feasibility Preview". Before it is computed, its introduction reads: "Before fitting, project every cell: how many areas will fit, which areas have no direct estimate, and which results are projected to be flagged or suppressed." A button labeled "Preview This Run" computes it. After computation, the button reads "Refresh Preview". The preview is also computed automatically when you arrive in Step Two from the "Continue to Model Setup →" button in Step One.

The computed view shows four figures with these exact labels: "Cells That Will Fit", shown in the form "{fitted} of {total}"; "Cells with Too Few Areas"; "Areas with No Direct Estimate"; and "Projected Suppressions".

When areas lack direct estimates, a line follows in the form: "Excluded and named in the report: {area list}." The line lists eight areas, then adds "and {N} more" when the count exceeds eight. A standing note follows the figures: "Projected statuses use the direct estimates and are conservative: model estimates are usually more precise." Read projected suppressions with that note in mind. A projection based on direct precision overstates suppression for areas the model will improve, so the projected count is a planning ceiling rather than a forecast of the export outcome. If the preview cannot be computed, the card shows the text in the form: "The preview could not be computed: {reason}". Correct the named cause and refresh.

Use the four figures as decisions, not as decoration. If cells that will fit is smaller than the total, find which cells are too thin and decide whether a subgroup collapse, a broader level grouping in the source, or a separate run is the right answer before you fit. If areas with no direct estimate is greater than zero, confirm that each named area is genuinely without an estimate in the source, because those areas are excluded from the fit and named in the report. If projected suppressions consume most of your areas, revisit the disclosure preset choice and the per-area respondent counts now, while the response can still be a design decision rather than a publication problem discovered in Step Four.

Quality Checks

The card is headed "Quality Checks". Before they run, its text reads: "Twenty checks run against the confirmed setup: weights, design variables, indicator coverage, subgroup levels, covariates, and disclosure minimums. A blocking problem stops the run." A button labeled "Run Quality Checks" runs them. After they run, the button reads "Re-Run Checks".

After they run, the card summary follows the exact pattern: "{N} of 20 checks clear · {M} findings. The full report is in Step 4 after a run." The twenty checks cover the setup as a whole: the weights, the design variables, the indicator coverage, the subgroup levels, the covariates, and the disclosure minimums. The full list, with each failing check identifier and its issue messages in the words the check itself uses, appears in Step Four in the "QC Report" card after a run. In Step Two, the operational question is narrower: does any check block the run?

A blocking check stops the run. When a blocking check stands, Step Two also shows its message on its own line in the form "Blocking check: {message}". Chapter Ten lists the run blockers, including the blocking check form. Correct a blocking finding at its source, which is usually a role in Step One, a missing selection in Step Two, or a value in the source file, and then re-run the checks. Non-blocking findings are warnings to read and record. They travel into the QC report, and a reviewer will see them there, so a finding you judge acceptable should be acceptable in writing.

The run itself re-runs the checks automatically if they have never run for the current setup, so skipping this card does not skip the checks. It only skips your chance to read them before fitting.

Fit Summary

After a run, a card headed "Fit Summary" appears in the right column of Step Two. If settings changed since the run, the card opens with the text: "Settings changed since this run. Run the model again to update these figures." In that state, the figures describe the previous run and must not be quoted as the current setup results.

The card shows four figures with these exact labels: "Variance component A", the fitted between-area variance component shown to three decimals; "Shrinkage γ range", shown in the form "{min} – {max}" across the fitted cell areas; "Median CV vs. Direct", the fit median coefficient of variation ratio minus one, shown as a percent; and "Areas Flagged or Suppressed", shown in the form "{N} of {total}".

A closing line follows the pattern: "Model CV for the median area: {CV}. Direct estimates are beside the model estimates in Step 3." The card is shown for the first fitted cell. In a multi-cell run, use Step Three and Step Four, which follow the selected cell, for cell-specific figures, and use this card as the quick confirmation that the first cell fitted, that its variance component is not at a degenerate value, that its shrinkage range spans a sensible interval, and that its median precision moved in the expected direction. A median coefficient of variation ratio that shows no improvement, or a shrinkage range pressed against zero or one for every area, is a reason to inspect the covariates and the variance column before you publish anything from the run.

Running the Model

The run controls stand at the foot of the "Model Setup" card in Step Two. The primary button is labeled "Run Model →". Beside it stands a button labeled "View Current Results", which is disabled until a result exists, and then jumps to Step Three. This chapter covers the states of the run button, the progress display, every blocker that can stand in the way of a fit, what happens on success and on failure, and the stale results state that follows a settings change.

One behavior of the run button corrects a natural assumption, so the guide states it directly. The "Run Model →" button is not disabled by blockers. It stays clickable at all times when no run is in flight. Blocker text appears beneath the button when a blocker stands, and clicking "Run Model →" with a blocker present puts that blocker into the run error line instead of fitting. Read the line beneath the button before you click. If it names a missing confirmation, the click will not fit the model, and the fastest path is to follow the blocker text to the step it names.

The blockers, in the exact words the app uses, are as follows.

- "Load a data file first."
- "The data file is not stored in this browser. Add the file again to run."
- For a microdata setup: "Confirm the area variable and the weight variable in Step 1."
- For a microdata setup: "Select at least one indicator in Step 2."
- For an area-level setup: "Confirm the area, direct estimate, and sample size columns in Step 1."
- For an area-level setup: "Confirm the sampling variance (or standard error) column in Step 1."
- For a blocking quality check, in the form: "A quality check blocks this run: {check message}"

Each blocker names its own fix. The first asks for a file. The second arises when a restored project was saved without its data file, because the "Keep the Data File with the Project" setting was off; Chapter Seventeen covers that state, and the fix is to add the file again in Step One. The next two pairs ask for role confirmations in Step One or a selection in Step Two, and they name the exact columns or the exact selection that is missing. The last asks you to resolve the named quality check. When more than one problem stands, the app surfaces them in order, so correcting the named problem and reading the line again is a reliable routine.

While a run is in flight, the button reads "Running…" and is disabled. A progress display appears with a progress bar and a stage checklist built only from the events the fitting process itself reports. The stage labels, in order, are "Direct estimates", "Design effects", "Model fitting", and "Disclosure checks". Beside the active stage, a count follows the form " – step {completed} of {total}". Beside each finished stage stands a check mark. The checklist is therefore a truthful progress account rather than an animation: a stage is marked only when the process reports it complete. Microdata runs spend visible time in the first two stages, because the direct estimates and the design effects are computed from the respondent rows. Area-level runs pass through those stages quickly, because the direct estimates arrive with the file. Both paths spend their main time in "Model fitting", and Full mode spends more of it there than Quick mode, with the bootstrap exception for adjusted-REML cells adding bootstrap time in either mode.

On success, the app advances to Step Three automatically, showing the first fitted cell, or the first cell if none fitted. That last clause matters. A run can complete as a process while a cell within it does not fit, most often because the cell holds fewer than five areas with direct estimates. The fit minimum is five areas with direct estimates per cell. Below that minimum, the cell is not fitted, direct estimates are reported for it, and the run report names the reason. A cell that fits on five to nine areas fits with a warning that the model was fitted on fewer than ten areas and that the estimates should be treated with care. Both messages appear in the run report in Step Four and are quoted in pattern form in Chapter Twelve. Success of the run as a whole therefore does not certify every cell. Read the cell status in Step Three, where a too-small cell carries its own note, before you read any figures.

A failed run shows its error in an alert line under the buttons. The error text comes from the fitting process itself, and a generic fallback reads "The run failed unexpectedly." when no specific text is available. A failure leaves your setup intact. Correct the cause named in the error, or return to the preview and the checks if the error is not specific, and run again. No partial result replaces a previous good result on a failure, and the previous result remains available through "View Current Results" until a new run succeeds.

After a successful run, any change to a setting that enters the fit marks the result as stale. Step Three opens in that state with the note: "Settings changed since this run. These are the previous results; run the model again in Step 2 to update them." The Step Two "Fit Summary" card carries the shorter form: "Settings changed since this run. Run the model again to update these figures." Treat stale results as a record of the previous configuration only. They remain useful for comparison while you adjust a setting, which is why the app keeps them visible rather than clearing the screen, but no figure from a stale display should be quoted, exported as current, or compared against a fresh figure from a different configuration without naming the difference. Exports are generated from a completed run, and the disciplined routine is simple: finish adjusting, run once more, confirm that no stale note stands on Step Three, and only then move to Step Four for diagnostics and exports.

A closing note on run discipline collects the habits the earlier chapters built. Confirm the roles in Step One. Read the "Outcome" line and the scale declaration at the top of the "Model Setup" card. Read "What Will Run" as a reviewer would. Read the feasibility preview figures and the named exclusions. Read the quality checks summary and clear every blocker. Then run. A run performed in that order rarely surprises, and when it does surprise, the surprise is in the data or in the model, which is where surprises carry information, rather than in the setup, where they only cost time.

Step Three: Reading Your Results

Step Three is titled "Results" with the subtitle "EBLUP Table and Chart". EBLUP is the standard abbreviation for the empirical best linear unbiased predictor, which is the name of the model estimate this model produces for each area. With no results yet, the step shows a card headed "No Results Yet". When data is loaded, its text reads: "Run the model in Step 2. The estimates table, chart, and interpretation appear here." When no data is loaded, its text reads: "Load a data file in Step 1, then run the model in Step 2. The estimates table, chart, and interpretation appear here." A button labeled "Go to Step 2", or "Go to Step 1" when that is the missing step, takes you to the work. With results present, the screen stands in a fixed order, and this chapter follows that order: the toolbar, the readout band, the trial lock note when it applies, the stats row, a note for a too-small cell when it applies, the estimates table, the chart, and "Reading These Results".

The toolbar

The toolbar holds a search box, a result set dropdown when the run produced more than one cell, a sort pill, and two quick export buttons.

The search box has the placeholder "Search areas..." and the accessible label "Search areas". It filters the table rows by area identifier text as you type. Clearing it restores every row. Searching never changes the fit, the selected cell, or any figure outside the table display.

The result set dropdown has the accessible label "Result set". It appears only when the run produced more than one cell. Each entry names a cell in the form "{indicator}", or in the form "{indicator}: {subgroup variable}: {level}" for a subgroup cell. The selected cell drives the table, the chart, the readout band, and everything shown in Step Four. When you switch cells, read the readout band again before the table, because the statewide figure, the fitted count, and the flagged count all belong to the selected cell and change with it.

The sort pill shows the current table sort, for example "Sorted by model estimate ↓". It reflects the sort state described with the table below.

The two quick export buttons are labeled "↓ Per-Area Report (.xlsx)" and "↓ Results Table (.csv)". Each carries a lock symbol in the trial. They are two of the five exports, offered here for convenience. The full export center, with all five exports and their detail lines, stands in Step Four. Chapter Fifteen covers every export, the trial lock behavior, and the license.

The readout band

The readout band holds four blocks with exact labels.

The first block is labeled "Statewide Estimate". Its big figure is the population-weighted model mean for the selected cell, formatted as a percent with one decimal. Its sub-line follows the form "Statewide direct {percent}" and gives the statewide direct estimate on

the same basis. These figures come from the stored proportion-scale figures for the cell. When population counts exist, the weights are the population counts of the areas. When population counts do not exist, the figure is the plain mean of the areas. When benchmarking was applied, the model mean equals the benchmark target, the statewide direct estimate, by construction, which is why the proven sample run shows 12.1% in both the big figure and the sub-line. These statewide figures are never the aggregation means from the fit diagnostics, which are computed on the transformed arcsine scale and are not display values. If you ever see a statewide figure that cannot be reconciled with the direct estimates and the population counts, the proportion-scale readout is the figure to trust and report.

The second block is labeled "Areas Fitted". Its figure counts the areas with a model estimate, and its sub-line follows the form "of {total} areas", where the total is the full area count of the cell even when the trial cap hides rows in the table. A gap between the two numbers means some areas have no model estimate, because they had no direct estimate, because their covariate row was incomplete, or because the cell did not fit. The run report in Step Four names the reason per area or per cell.

The third block is labeled "Benchmarking". It reads "Applied" with a sub-line naming the logit or ratio shift, or "Off", or "Skipped", or a dash. Skipped and Off are different states. Off means the setting was off. Skipped means the setting was on and the run could not apply it to this cell, and the run report is the place that records the circumstances.

The fourth block is labeled "Flagged Areas". Its figure is the flagged count for the cell, and its sub-line follows the form "{N} suppressed" when any area is suppressed. Flagged and suppressed are distinct outcomes of the disclosure preset, as Chapter Eight explained. A flagged estimate is published with a warning. A suppressed estimate is withheld from every export.

The trial lock note

When your own data exceeds ten areas and no license is active, a lock note appears below the readout band. It reads as follows.

> "Trial preview. Showing ten of {total} areas. A license displays every area and unlocks all exports. Unlock the Tool"

The last words are a button that opens the license dialog. The note states the trial cap plainly, and the guide states it in the same plain terms. On your own data in the trial, the full workflow runs and the model fits all areas, but the results display is capped at ten areas per cell. The cap binds the Step Three table, the chart, the stats row, and the interpretation, and it binds the per-area lists in Step Four. Aggregate counts in Step Four, such as how many areas were flagged, suppressed, or excluded, are run facts and are not capped. The sample data is exempt from the cap, so the sample run shows all sixteen areas without a license. Chapter Fifteen gives the complete trial and license account.

The stats row

Below the readout band, four blocks summarize the selected cell over the visible areas. Their labels are "MODEL ESTIMATE RANGE", "MEDIAN CV – MODEL", "BIGGEST CORRECTION", and "AREAS IMPROVED".

"MODEL ESTIMATE RANGE" gives the smallest and largest model estimates as percents, in the form "{min}% – {max}%", with a small line that gives the direct range in the form "Direct range was {min}% – {max}%". Compare the two ranges to see the shrinkage effect in one glance: the model range is usually the narrower one, because shrinkage pulls the extremes toward the pattern the covariates support.

"MEDIAN CV – MODEL" gives the median model coefficient of variation, with a small line in the form "Direct median {CV}". The coefficient of variation here is the standard error of the estimate expressed as a percent of the estimate itself. A lower median for the model than for the direct estimates is the central precision claim of the method, stated as a comparison on your own run rather than as a promise.

"BIGGEST CORRECTION" names the area the model moved most, with a small line in the form "{direct}% direct → {model}% model". That area deserves a manual look. A big correction is expected when the direct estimate was noisy and the covariates are informative. It deserves checking when the area direct estimate was based on a reasonable count, because then the correction says the covariates and the area disagree, and one of them may be wrong in a way the model cannot detect.

"AREAS IMPROVED" gives a count in the form "{N} / {areas with model}", with the small line "Model CV below the direct CV". It counts the visible areas whose model coefficient of variation sits below their direct coefficient of variation. It is a count, not a guarantee: an area can fail to improve, and Chapter Fourteen records validation cases in which the model was less accurate than the direct estimates for a given design. Read the count beside the biggest correction and the median comparison, as a panel, rather than alone.

All four blocks are computed over the visible, post-cap areas only. In a capped trial display, they describe the ten visible areas, not the full cell. Keep that scope in mind when you compare a trial display with a licensed display of the same run.

The too-small cell note

When the selected cell has the status of a cell that was too small to fit, a note appears in the exact form: "This set of areas was too small to fit a model, so direct estimates are shown. The run report in Step 4 names the reason." The fit minimum is five areas with direct estimates. When you meet this note, the table below it shows direct estimates without model values for that cell, and the correct responses are to accept the direct estimates for that cell, to broaden the cell by collapsing subgroup levels or by removing the subgroup split for that run, or to add areas if the source genuinely holds more. Lowering standards to force a fit is not among the options the tool offers, because the minimum is a property of the fit, not a setting.

The estimates table

The estimates table has the accessible label "Estimates by area". It has eight columns, and this guide lists them in order and explains each, because the table is the primary record of the run on screen.

Column	What It Holds
"Area"	The area identifier
"n"	The respondent count for the area
"Direct %"	The direct estimate as a percent with one decimal
"Direct SE"	The standard error of the direct estimate
"Model (EBLUP) %"	The model estimate as a percent with one decimal
"Model CV"	The coefficient of variation of the model estimate
"Shrinkage γ"	The weight the model places on the direct estimate, to two decimals
"Flag"	The disclosure outcome for the area

Percents show with one decimal, for example "12.1%". The count column shows whole numbers with thousands separators. A coefficient of variation shows as a percent with no decimal at or above 9.5 percent, and with one decimal below that point. A missing value shows a dash. At phone width, the columns "n", "Direct SE", and "Shrinkage γ" are hidden and the other five remain, although the tool is designed for desktop use and the desktop display is the complete one.

Every heading except "Flag" is a sort button. Clicking a heading sorts by that column, and clicking it again reverses the direction. The heading shows an up arrow, a down arrow, or a double arrow to mark its sort state. The default sort is "Model (EBLUP) %" descending, so the table opens with the highest model estimates first. Sorting changes display order only. It does not change any value, and it does not change the chart order, which follows the direct estimates.

The "Flag" cell reads "Suppressed: Do Not Publish" for a suppressed area and "Flagged: Unstable" for a flagged area, and it is empty otherwise. When a suppression reason is recorded for an area, the reason is the tooltip of the flag. Hover over the flag to read why that area was suppressed under the active preset, for example an effective sample size below the preset minimum or an interval wider than the preset allows.

Shrinkage, the "Shrinkage γ" column, deserves its own paragraph because it is the most informative single number in the table for judging each row. Gamma is the weight the model places on the direct estimate. A gamma near one means the model estimate rests almost entirely on the direct estimate, which happens when the direct estimate is precise relative to the between-area variation. A gamma near zero means the direct estimate was noisy and the model estimate rests more on the regression prediction from the covariates. Read gamma beside the count column and the direct standard error. A low gamma with a small count is the model working as designed. A low gamma with a large count is a signal to check the variance for that area, because a large count with a large sampling variance is an unusual pairing that can come from a design effect, a weight problem, or a variance column error.

The note beneath the table reads in full: "γ (gamma) is the weight the model places on the direct estimate. A low γ means the direct estimate was noisy, and the model estimate rests more on the regression prediction. Select any column heading to sort. Suppressed values remain visible in this workspace; suppression is enforced on exports."

The table has no interval column. The guide states this plainly because other descriptions of tools of this kind often imply one. On this screen, the model interval for an area exists only as the bar in the chart. In numeric form, intervals exist only in the exports. If you need an interval figure for a specific area while you work on screen, read its bar against the percent axis in the chart for an approximate reading, and take the exact figure from the "Results Table (.csv)", the master dataset, or the area workbook in the Per-Area Report after export, with a license active.

The chart

The chart card is headed "Direct vs. Model Estimates". Its caption reads, with the color word following the active theme accent and reading "orange" under the default theme: "Gray dots are the direct estimates; orange dots are the model estimates, with the model interval as a bar. Small, noisy areas move the most." Its legend reads "Direct estimate", "Model (EBLUP)", and "Model interval". Its accessible label is "Direct estimates beside model estimates, with model intervals".

The chart is a forest plot with one row per visible area, ordered by direct estimate with the highest first. Each row shows the model interval as a horizontal bar with end caps, the direct estimate as a gray dot, the model estimate as an accent-colored dot, and an arrow from the direct estimate to the model estimate when the two differ visibly. That arrow is the shrinkage pull for the area, drawn to scale. At the right of each row, a coefficient of

variation strip shows a banded dot beside the model coefficient of variation value. The banding runs teal to twenty percent, amber to thirty percent, and red above thirty percent, which matches the flag region of the Standard preset and lets you see, without reading numbers, which rows sit in the caution bands. Flagged and suppressed rows are dimmed and carry a badge at the right edge reading "FLAGGED" or "SUPPRESSED". The axis is labeled in percents.

Read the chart in three passes. First, read the arrows. Long arrows concentrated among the small areas are the expected pattern. A long arrow on a large area is the chart form of the biggest correction check in the stats row. Second, read the intervals. Rows whose bars span a large share of the axis remain uncertain after modeling, and no point estimate from such a row should be quoted without its interval. Third, read the dots against each other down the ranking. If the model order differs sharply from the direct order in the middle of the distribution, where counts are usually adequate, inspect those rows in the table before you describe any ranking. The interpretation section below the chart closes with the standing instruction on this point: do not rank model estimates as if they were exact.

Reading These Results

The section headed "Reading These Results" holds up to three paragraphs generated by a fixed template. Every number in the paragraphs is inserted from the run, and no sentence is free-written. The template is described here so that you can recognize each sentence when you meet it and know what computed it.

The first paragraph, when any area has both estimates, follows the form: "The model moved {area} the most: {direct} direct to {model} model", with a clause added when both coefficients of variation exist, in the form "and its CV fell from {CV} to {CV}". It continues with a count of the areas with model estimates and how many of them saw their coefficient of variation fall, and it closes with the sentence: "Areas with large samples move very little: where the direct estimate was already precise, the model estimate stays close to the direct estimate."

The second paragraph, when both ranges exist, follows the form: "Model estimates run from {min} to {max}; the direct estimates ran from {min} to {max}." It closes with the sentence: "Shrinkage pulls the extremes toward the pattern the covariates support, which is why the model range is usually the narrower one."

The third paragraph always appears. When flagged or suppressed areas exist, it opens with a caution in the form: "Use with care: {N} area(s) carry a reliability flag under the active disclosure preset, and {N} is/are suppressed and must not be published." It then always closes with the sentence: "Report the CV alongside every rate, and do not rank model estimates as if they were exact."

All interpretation figures are computed over the visible, post-cap areas, like the stats row. In a capped trial display, the paragraph describes the visible ten areas. A licensed display of the same run can therefore word the biggest correction and the counts differently, because its visible set is the full cell. That difference is a display scope difference, not a change in the fit.

Step Four: Diagnostics and the Publication Checklist

Step Four is titled "Diagnostics & Export" with the subtitle "Checks, Interpretation, Files". With no run completed, it shows a card headed "No Diagnostics Yet". When data is loaded, its text reads: "Diagnostics, the publication checklist, the QC report, and the export center appear once a run completes." When no data is loaded, the same sentence is preceded by the words: "Load a data file in Step 1 and run the model in Step 2." A button labeled "Go to Step 2", or "Go to Step 1" when that is the missing step, takes you to the work.

Everything on this step is shown for the result set selected in Step Three. When the run has more than one cell, the "Diagnostics" card says so with the line: "Shown for the result set selected in Step 3." Switch cells in Step Three to change what Step Four displays. The step holds five cards: "Diagnostics", "Publication Checklist", the export center headed "What You Can Download" or "What You Can Download with a License", "QC Report", and "Project". This chapter covers the diagnostics, the checklist, and the QC report. Chapter Fifteen covers the export center, and Chapter Seventeen covers the "Project" card.

Diagnostics

The card is headed "Diagnostics". Its introduction reads: "The checks a reviewer will ask about, on the same screen as the results." When the selected cell has no fitted model, the card shows the exact text: "This result set has no fitted model (too few areas), so no model diagnostics are available. The direct estimates are reported as computed." Otherwise, the card shows three diagnostic items, a coefficients block, and nothing else. Each item has a name, a right-hand value, a meter or summary, and a note.

The first item is named "Residual Normality". Its right-hand value follows the form "Shapiro–Wilk p = {p}", or reads "Needs at least ten areas" when no p-value stands for the cell. In the shipped tool, no Shapiro–Wilk value is computed, so that reading stands for every cell, including the sixteen-area sample run of Chapter Two; read it as the test not being reported, not as a judgment that the cell is too small. When a p-value is reported, a meter displays it. Its note reads "No area is an outlier on standardized residuals." when no area is flagged, or names the flagged residuals in the form "Flagged residuals: {area} ({value}), ..." The note then gives the skewness and kurtosis in the form "Skewness {value}, kurtosis {value}." Residual normality is the working assumption behind the area effects in the model. A very small p-value, heavy skewness, or a named outlier area does not invalidate a run by itself, but each is a reason to inspect the named area and to note the finding in your project record. An outlier area often turns out to be an area whose direct estimate or variance was mis-entered, or an area whose covariate values belong to a different vintage. Correct such causes in the source and rerun. An outlier with no data cause is a genuine model finding, and the honest response is to report the diagnostics as they stand.

The second item is named "Model Fit: Covariates". Its right-hand value follows the form "{N} selected", with the addition " · R² (Fay–Herriot) {value}" when that coefficient is available. A meter displays the coefficient of determination. Its note follows the form "Median shrinkage γ is {value}." When shrinkage is low, the note adds the sentence: "Shrinkage is low: the direct estimates carry most of the weight in this fit." Read the two halves together. A high coefficient of determination with a moderate median shrinkage describes a fit in which the covariates explain much of the between-area pattern and the model still respects precise direct estimates. A selected count of zero describes an intercept-only fit, whatever the meter shows, and returns you to the covariate discussion in Chapter Seven.

A note on the third item requires care, because it mixes two kinds of figure and the guide separates them. The third item is named "Bias Check: Model vs. Direct". Its right-hand value follows the form "{percent} apart". That value is the absolute relative difference of the diagnostics aggregate check, which is a transformed-scale diagnostic used only as a gap indicator. It is not a statewide figure and must not be quoted as one. Its note follows the form: "The {population-weighted model mean | model mean} is {percent} against a statewide direct estimate of {percent}." Those two figures in the note are the proportion-scale statewide figures, the same figures the readout band in Step Three displays. When the gap exceeds the five percent review threshold, the note adds the sentence: "The gap is larger than the 5% review threshold." Judge the item by its note figures and by the threshold sentence. Use the right-hand value only as a signal that a gap exists and deserves the reading the note provides.

Under the three items stands a code block headed "Coefficients (fitting scale)". It holds one line per term, the intercept first and then each selected covariate, in the pattern "{name} β {value} SE {value}". A final line follows the pattern: "A (REML) {value} · method: {method used}", where the value is the fitted between-area variance component to three decimals and the method names the fitting method used for the cell. The coefficients are on the fitting scale, which under the default configuration is the arcsine scale. Do not interpret an arcsine-scale coefficient as a percentage point effect. Its sign tells you the direction of association on the transformed scale, its standard error tells you the precision of that association, and together they support a statement that a covariate is associated with the outcome in a stated direction. A percentage point statement belongs to the estimates themselves, not to these coefficients.

Publication Checklist

The card is headed "Publication Checklist". Its sub-line follows the pattern: "{Standard preset | Strict preset | SAE Standard preset | Disclosure checks off} applied to this run, for the result set in Step 3." The card holds six lines. A failing line is styled as a failure. Read the card top to bottom before you export anything, because it is the last consolidated statement of whether this cell, under this preset, is ready to publish.

The six lines, in their template forms, are as follows.

1. "Minimum n met in all {N} areas (smallest n = {n})": This line passes when the smallest respondent count meets the minimum respondents of the active preset.
2. "{N} of {M} model CVs under the {30% | 20%} flag threshold": The threshold named is the flag threshold of the active preset.
3. "No suppressed areas under the preset": Or, when areas are suppressed, the line names the count in the form "{N} area(s) is/are suppressed under the preset and will not export".
4. "Methods metadata recorded: transform {transform}, MSE {bootstrap | analytical}": With benchmarking added in the form ", benchmarking by {method} shift" when benchmarking was applied.
5. "No area was excluded from the model": Or, when areas were excluded, the line reads: "Areas with no direct estimate are excluded and named in the run report below".
6. "Reproducibility: seed "{seed}" and engine version {version} recorded in the run manifest": The seed named is the run seed "sae-studio", which has no user control, as Chapter Eight stated.

A checklist with no failing line does not certify that the estimates are correct. It certifies that the minimum count rule is met, that the coefficient of variation position against the preset threshold is as stated, that the suppression outcome is as stated, that the methods metadata was recorded, that exclusions are as stated, and that the run can be reproduced from its manifest. Correctness of the inputs remains the preparation work of Chapter Four and the confirmation work of Step One and Chapter Five. Keep the two judgments separate in your own review notes, and a reviewer reading those notes will thank you for it.

QC Report

The card is headed "QC Report". Its introduction reads: "Every check that ran, every exclusion, and every warning. No finding is dropped silently." The card has two halves.

The first half is headed "File and Setup Checks". It opens with any file read issues, in the words the reader issued them. It then shows either the text "Quality checks have not run on the current setup." with a link button labeled "Run them now", or a summary in the form "{N} of 20 checks passed with no findings." followed by each failing check identifier, for example "QC-07", and its issue messages in the words the check uses. The check identifiers are on-screen text in this card and in the methods report. They name the twenty setup checks in order, and a failing identifier beside its messages is the shortest route to the cause: correct the setup element the message names, re-run the checks from Step Two or from the link button here, and confirm that the finding clears.

The second half is headed "Run Report". It shows either the text "The run reported no issues." or the run issue list. Each issue line follows the pattern " **{code}:** {message}", where the code names the kind of issue and the message gives its substance in the words of the fitting process. The codes you are most likely to meet name a missing direct estimate for one or more areas, a cell with too few areas to fit, a low area count warning for a fitted cell, and a transformation note. Areas with no direct estimate are excluded from the fit and named here. A low area count message states that the model was fitted on a stated number of areas, fewer than ten, and that the estimates should be treated with care. A too-few-areas message states how many areas have direct estimates and that the model needs at least five areas to fit, and confirms that direct estimates are reported for that cell.

Read the QC report as the run record of the project. Together with the methods report, it answers the questions a reviewer asks first: what was checked, what failed, what was excluded, and what was warned about. If a figure in Step Three is ever questioned, the answer begins in this card, continues in the methods report, and ends in the reproducibility package. That chain is deliberate, and Chapter Thirteen follows it into the report itself.

The Methods Report

The methods report is a Word document generated from a completed run. It is one of the five exports, labeled "Methods Report (.docx)" in the export center in Step Four, with the detail line "Model, transform, MSE method, and every rule that fired." Step Three quick actions do not offer it. A license is required, as it is for every export. The downloaded file is named in the form {project-slug}_methods_report.docx. This chapter describes the document itself, its fifteen sections in order, its watermark, and how to read the tables it explains. No PDF version of the report exists, and no export produces one. Chapter Fifteen states the full export list, and this chapter states the same fact where a reader of the report chapter would look for it: the formatted narrative report of a run is this Word document, and there is no PDF export in the app.

The document title property follows the form "SAE Studio methods report - run {runId}". On the page, a title band reads "Methods Report" in thirty-six point bold type, in warm white on Signal Orange. Below the band stand the survey name line, which is the project name, and a run line in the form "Run {runId} - {startedAt}". A watermark line follows. The watermark, which names the licensee and the run identification, also runs in the footer of every page and in the document properties. The presentation follows the house export standard: Calibri type, body text at twelve point, section headings at eighteen point bold on pale orange, Letter pages, and one-inch margins.

The report holds fifteen sections, in this exact order, with these exact headings.

1. "Summary"
2. "Data"
3. "Survey design and weights"
4. "Direct estimation"
5. "Small area model"
6. "Model fitting"
7. "Uncertainty"
8. "Benchmarking"
9. "Reliability and disclosure rules"
10. "Diagnostics summary"
11. "Quality checks and exclusions"
12. "Limitations"
13. "How to read the tables"

14. "Software and reproducibility"

15. "References"

The "Summary" section states the indicators, the areas, the respondents, the domains, the fitted cells, the transformation scale, the mean squared error method, and the published, flagged, and suppressed counts. Read it as the one-paragraph account of the run for a reader who will read nothing else, and check its counts against the Step Three readout and the Step Four checklist for the same cell before you circulate the document. A count that disagrees across those three places means the documents were generated from different runs or different cells, which is a filing error rather than a modeling one, and it is caught only by this comparison.

The "Data" section names the source file and its format, and the covariate source. It is the provenance record for the run inputs. If a covariate file was used, this section is where its contribution is stated, matching the "Covariates come from {filename}." line in Step Two.

The "Survey design and weights" section names the weight, stratum, and primary sampling unit columns, or states their absence and the fallback that was used. It states the Taylor linearization basis for the variance calculations, the lonely primary sampling unit rule, and any finite population correction. On the area-level path, the same section states that the published values were used as delivered, that the published sample size is treated as the effective sample size with a design effect of one, and that direct intervals are Wald intervals. That area-level statement matters for interpretation: an area-level run inherits the precision claims of the published direct estimates exactly as supplied, and any design effect already inside a published variance stays inside it.

The "Direct estimation" section gives the account of direct estimation for the microdata path, covering the Hájek estimator, and the logit and Korn-Graubard interval forms. It states which design effect basis the run used. It also carries the validation record for the direct estimation basis, including the coverage and precision findings that Chapter Fourteen summarizes in plain words. The full record stands in this section of every report, so a report circulated without this guide still carries the evidence about its own basis.

The "Small area model" section gives the arcsine construction in the form theta equals asin of the square root of p, and psi equals one divided by four times n divided by the design effect. It gives the model equation. It describes the BIC covariate selection with its screens, an absolute correlation screen above 0.90 and a variance inflation screen above ten. It states the design effect basis of the run, and it states whether the transform bias correction was applied. A subheading in this section, "Covariates selected, by cell", lists the selected covariates for each fitted cell. In a multi-cell run, read that subheading before you compare cells, because cells that selected different covariates are not estimates from the same model in the strict sense, and a comparison that ignores the difference misstates what was fitted.

The "Model fitting" section records the restricted maximum likelihood fitting, the Li–Lahiri adjusted-REML boundary fallback, and the Prasad–Rao fallback. It holds a per-cell table giving, for each cell, its status, its fitting method, its between-area variance component, and its areas fitted. When a cell method reads as the adjusted method, remember the precision consequence from Chapter Eight: that cell mean squared error came from the parametric bootstrap in every run mode.

The "Uncertainty" section states the mean squared error route for the run mode in force. It states that Quick mode root mean squared error is interval-based on the arcsine scale. It states that the same data can land on opposite sides of the thirty percent flag threshold in Quick mode and in Full mode. It states that intervals are formed on the transformed scale and always lie inside zero to one. Keep that last sentence beside any interval you quote from the run.

The "Benchmarking" section states the method and the target when benchmarking was applied, or states that benchmarking was not applied. When it was applied, this section is the citable record of the target figure for the cell.

The "Reliability and disclosure rules" section states the active preset and its rule account. For the Standard preset, it gives the Parker and others tests as applied, including the Korn-Graubard width basis for direct estimate intervals and the note that the coefficient of variation ladder is an addition of the tool rather than a part of the published standard. It gives the outcome counts under the preset, it describes complementary suppression where it applies, and it carries the published share account that Chapter Eight summarized: at about twenty respondents per area, the Standard preset published 0.93 percent of area estimates with frame counts supplied and 0.00 percent without them, at flagship scale, and about ninety percent at smaller scale.

The "Diagnostics summary" section tabulates, per cell, the median gamma, the median coefficient of variation ratio, the aggregate check, and the outlier counts. It is the tabular counterpart of the Step Four "Diagnostics" card, extended across all cells rather than the selected cell only.

The "Quality checks and exclusions" section reports the twenty quality checks and names every excluded area. It carries conditional subheadings, "Named exclusions", "Other run findings", and "Transformations applied", which appear when their content exists. An excluded area named here with its reason is the complete answer to the most common reviewer question about a missing area, and no separate correspondence should be needed to answer it.

The "Limitations" section carries the bias disclosures and, when a cell triggers the conditions, the out-of-validated-range account. Chapter Fourteen presents the same substance in guide form. A cell is flagged as out of validated range when any of six conditions holds for it: benchmarking was not applied to it; it fits fewer than ten areas; the median fitted area has fewer than six respondents; any model estimate falls below 0.02;

any model estimate reaches 0.40 or above; or the run supplied no frame primary sampling unit counts and the median fitted area has fewer than fifty-four respondents. When the account appears in a report, read the estimates for that cell with the caution the account describes, because the cell sits outside the conditions under which the validation record was established.

The "How to read the tables" section explains the direct and model estimates, gamma, and the two markers used in the exported tables. A flagged value is marked with a single asterisk character. A suppressed value is marked with two dash characters in place of the value. Those markers are how suppression appears in files: the value is withheld, and its row, its status, and its reason remain. When you receive an exported table from a colleague, read this section of the accompanying methods report before you read the table, and the markers need no guessing.

The "Software and reproducibility" section gives the engine version, the run identifier, the seed, and the SHA-256 digest of the run manifest. It describes the R script in the reproducibility package, which recomputes the run with the R survey and sae packages and compares the result against the packaged results. This section is what makes the run checkable by a third party without access to your browser.

The "References" section lists nine references: Fay and Herriot in 1979; Prasad and Rao in 1990; Datta and Lahiri in 2000; Li and Lahiri in 2010; Korn and Graubard in 1998; Lumley in 2010; Molina and Marhuenda in 2015; Rao and Molina in 2015; and Parker and others in 2017. Each method named in the report traces to one of those sources, and a reviewer who wants the primary statement of any procedure in the report will find it there.

Handle the report as the record of the run. File it with the project name and the run identifier in its file name unchanged, keep it beside the exported tables it describes, and circulate it with any table you publish from the run. A table without its methods report asks its reader to trust it. A table with its methods report lets its reader check it, and that difference is the purpose of the document.

What the Validation Record Shows

Every methods report carries a validation record for the basis on which the shipped tool computes its direct estimates, its model estimates, and its reported precision. This chapter presents that record in plain words, in the same register as the report itself. It makes no claim that the model is better than the direct estimates in general, and it names the cases in which the model performed worse as plainly as the cases in which it performed better. Read this chapter before you publish from the tool for the first time, and return to it when a run sits near any boundary it describes.

Coverage across the validation designs

The shipped basis was re-measured across seventeen validation designs, including two arms that used real data rather than simulated populations. Pooled interval coverage fell inside the band from ninety-two percent to ninety-seven percent at sixteen of the seventeen designs. A nominal ninety-five percent interval that covers at a rate inside that band is behaving close to its stated level across a range of conditions. One design fell outside the band, and the two real-data arms show the two directions in which performance can depart from the simulated pattern, so each arm is stated separately.

On the first real-data arm, coverage was 92.19 percent, which is inside the band. On that same arm, the model root mean squared error was 0.0727 against 0.0596 for the direct estimates. In that test, the model was less accurate than the direct estimates it refines. The intervals covered at an acceptable rate while the point estimates were, on average, farther from the true values than the direct estimates were. That combination is possible because coverage describes the intervals and root mean squared error describes the points, and a method can be honest about its uncertainty while still moving points in the wrong direction on a particular real dataset whose pattern the covariates do not capture.

On the second real-data arm, which was a sealed holdout, the model root mean squared error was 0.0571 against 0.0853 for the direct estimates, so the model was more accurate there. Its coverage was 98.20 percent, which is above the band. Coverage above the band means the intervals run wider than necessary for that arm: they contain the true values more often than the stated level requires, at the cost of width. A reader who treats a wide interval as a weak result should know that, on this arm, the width was more than the realized errors required.

The practical lesson is narrow and should be kept narrow. Across most tested designs, the intervals cover near their stated level. On real data, the model has been both less accurate and more accurate than the direct estimates, in different datasets. No validation record of that shape supports a blanket statement that the model improves every dataset. It supports the workflow this guide teaches: check the covariates, read the diagnostics, compare the model and direct coefficients of variation on your own run, inspect the biggest correction, and report the coefficient of variation alongside every rate.

Precision by band of the estimate

The validation work also scored reported precision by the band of the published estimate. The bands are the intervals from zero up to 0.10, from 0.10 to 0.20, from 0.20 to 0.40, and from 0.40 to one. Within each band, the ratio of the reported root mean squared error to the realized root mean squared error was compared against a criterion band from 0.90 to 1.10. A ratio inside that band means the reported precision matches the realized error closely. A ratio below the band means the reported precision runs narrow: the published coefficients of variation and interval widths understate the realized error. A ratio above the band means the reported precision runs wide: the published figures overstate the realized error.

The record shows two departures that a user of the tool should carry into every reading.

In the lowest band, below 0.10, the ratio falls outside the criterion band for subgroup cells and on both real-data arms. In plain terms, in the low band the reported precision runs narrow. The reported coefficients of variation and interval widths understate the realized error, and the reliability flags fire less often than they should. A small estimate with a reassuring coefficient of variation in the low band deserves more caution than its printed precision suggests. Where a low-band estimate will drive a decision, seek corroboration from the direct estimate, from the interval in the export, and from the covariate pattern, and consider reporting the estimate with an explicit caution about its band.

In the band from 0.10 to 0.20, at flagship scale, the ratio falls outside the criterion band in the other direction. In plain terms, reported precision in that band runs wide at that scale. The published figures overstate the realized error. An estimate in that band that is flagged at the margin may be more stable than its printed coefficient of variation suggests. That is not a license to ignore a flag. It is a reason to read the flag as the preset working as designed while knowing the direction of the known departure.

Mechanism scoring behind the low-band finding adds one refinement. At flagship scale in the low band, the published intervals themselves covered at close to the stated rate, at 92.5 percent to 93.4 percent, while the reported width ran narrow. The failure in that setting sits in the reported precision figures derived from the widths, rather than in the interval coverage itself. At two other designs in the record, the intervals themselves also ran narrow in the low band, with coverage from 88.0 percent to 91.4 percent. The distinction matters when you decide what to quote. An interval from a flagship-scale low-band cell has a better validation basis than the coefficient of variation printed beside the same estimate. When the two must be separated, quote the interval and treat the single-number precision figure with reserve.

A worked instance from the record fixes the scale of the low-band departure. Flagship subgroup cells in the low band showed a ratio of 0.839 with coverage of 93.5 percent. On one real-data arm, the low band showed a ratio of 0.580 with coverage of 81.0 percent, computed on four hundred fifty-seven area replicates, which is below the five hundred

replicates the validation treated as adequate for a firm reading. The guide gives those figures because they are in the shipped record, and because a general statement that precision runs narrow in the low band, without any scale, invites both overreaction and dismissal. The departure is material in the weakest case and moderate in the flagship subgroup case. Neither case supports quoting low-band precision at face value.

Totals and the unbenchmarked case

Two further findings from the validation work stand in the limitations discussion of every methods report, and both concern totals rather than individual areas.

Across forty validation populations of thirty areas each, with samples averaging about fifty-eight respondents per area and prevalence near twenty-eight percent, the population-weighted model totals averaged 0.23 percentage points below the true total, which is 0.78 percent in relative terms. The stated cause is that transformed values sit slightly low at small samples when prevalence is under fifty percent. A second effect stands beside the bias figure: model estimates pull each area toward its regression prediction, which compresses differences between areas and compresses rankings. A set of model estimates will therefore usually show less spread between the highest and lowest areas than the true values show, even when its total is close. Do not interpret a narrower spread across areas as evidence that the underlying differences narrowed. In this method, a narrower spread is partly the mechanics of the estimator.

The unbenchmarked finding is stronger, and it is the empirical reason benchmarking stands On by default. In a flagship-scale validation study with one hundred thirty-three areas, about twenty respondents per area, and prevalence near ten percent, unbenchmarked model estimates averaged 1.48 percentage points below the true values, in individual areas as well as in totals. An unbenchmarked run at similar scale and prevalence should be expected to sit low as a pattern, not merely to scatter around the truth. If you run with benchmarking off, state why in your project notes, and do not compare the levels from that run with levels from a benchmarked run as if the two were on the same footing. Their totals differ by construction and by this measured tendency.

The Standard preset published share

Chapter Eight summarized the disclosure finding, and this chapter places it in the validation record where it belongs. The Standard preset minimum effective sample size is the respondent count divided by the smoothed design effect. At about twenty respondents per area, nearly every estimate falls below the minimum effective sample size of thirty and is suppressed at export. Measured on the shipped basis at flagship scale, the published share was 0.93 percent of area estimates with frame counts supplied and 0.00 percent without them. At smaller scale, the same preset published 92.88 percent of area estimates under one truth design and 90.62 percent under the other. The consequence belongs to the flagship sample size, not to the preset at every scale. A project whose areas average near twenty respondents and whose governing standard is the Standard preset has a publication

problem to solve at the design stage: supply frame counts where the design has them, increase the per-area counts where fieldwork allows, or adopt a different preset on stated grounds before data collection closes. Discovering the published share in Step Four, after the run, is the latest and least useful point at which to learn it.

Outside the validated range

The methods report flags a cell as outside the validated range when any of six conditions holds for that cell. The six triggers are as follows.

- Benchmarking was not applied to the cell.
- The cell fits fewer than ten areas.
- The median fitted area has fewer than six respondents.
- Any model estimate in the cell falls below 0.02.
- Any model estimate in the cell reaches 0.40 or above.
- The run supplied no frame primary sampling unit counts, and the median fitted area has fewer than fifty-four respondents.

A flag of this kind does not say the estimates are wrong. It says the cell sits outside the conditions covered by the validation record summarized in this chapter, so the coverage, precision, and totals findings above cannot be assumed to carry over to it. The correct responses are to note the flag and its trigger in your project record, to give the flagged cell estimates a wider margin in any written interpretation, and, where a trigger is remediable, to remedy it. Supplying frame counts, applying benchmarking, and fitting more areas are remedies for three of the triggers. The estimate-level triggers, an estimate below 0.02 or at 0.40 or above, are properties of the outcome distribution, and the remedy there is caution in the low band and the high range rather than a settings change.

Quick and Full mode near a threshold

One sentence from the uncertainty record closes this chapter, because it prevents a common misreading of two correct runs. The same data can land on opposite sides of the thirty percent flag threshold in Quick mode and in Full mode. The two modes compute mean squared error by different routes, as Chapter Eight described, and an area near the threshold can be flagged in one mode and unflagged in the other with no change in data and no error in either run. When you change mode between runs of the same project, expect threshold-adjacent flags to move, and do not report that movement as a change in the areas. Report the mode with the run, keep the mode constant within a published series where you can, and when a flag decision matters for a single area near the threshold, examine its interval and its count before you rest a publication decision on the flag alone.

The validation record, read whole, supports a measured account of the tool. Its intervals cover near the stated level at most tested designs. Its reported precision runs narrow in the lowest estimate band and wide in the band from 0.10 to 0.20 at flagship scale. Its totals run slightly low on average in the validation populations, and its unbenchmarked estimates at flagship scale run lower still. Its Standard preset suppresses nearly everything at twenty respondents per area. Its model has beaten the direct estimates on one real dataset and lost to them on another. A guide that stated only the favorable halves of those sentences would mislead you about the tool you are using. The methods report carries the same record into every run, so the account travels with the numbers wherever the numbers go.

Exports and the License

This chapter covers the trial as it actually stands, the five exports one by one with their real file shapes, the suppression rule inside every export, and the license in full: what it adds, what it costs, where it is bought, how the dialog works field by field, every failure message and its fix, how a lost key is recovered, and how a license is removed from a machine. Read the trial account first, because it corrects the two most common assumptions about the tool: that the trial limits the modeling, and that the license unlocks model features. Neither is true.

The trial exactly

The trial has no time limit, no run limit, and no feature lock on any model setting. Benchmarking, bias correction, every disclosure preset, both run modes, and project saving all work in the trial exactly as they work with a license. The trial boundaries are two, and only two.

On the sample data, the full workflow runs with no cap. You can load the sample, configure it, run it, see all sixteen areas in the table and the chart, and read all diagnostics. The sample is the complete demonstration of the tool, and it stays complete without a license.

On your own data, the full workflow also runs: load, profile, configure, run, and diagnostics. The boundary is display and download. The results display is capped at ten areas per cell, as Chapter Eleven described, with the lock note naming the total and offering the "Unlock the Tool" button. Exports are all locked. Every export button, in Step Three and in Step Four, shows a lock symbol, and clicking one opens the license dialog. No file is produced and no error is shown. The store-level message for the locked state, which the dialog flow conveys, reads: "Exports require an active license. Enter the email address used at purchase and the license key from your receipt email to unlock every export for this project."

With a license active but no completed run, an export attempt gives the other gating message: "Run the model first; exports are generated from a completed run." Exports are generated from a completed run in all cases, so the disciplined order is run first, check second, export third.

What the license adds exactly

A license adds precisely two things.

First, every area on screen. The ten-area display cap lifts for your own data, so the table, the chart, the stats row, the interpretation, and the per-area lists in Step Four show the full cell.

Second, all exports. The five export kinds listed below become available.

Nothing else is gated. The license does not add benchmarking, because benchmarking is already free and on by default. It does not add Full mode, and it does not add any preset. It does not add project saving, which works in the trial. The Companion Workbook is a separate license deliverable outside the app, and Chapter Sixteen describes it as the standalone file it is. Do not look for a workbook button in the app, because none exists.

The price is $149 per year. Purchase happens on the product page, not inside the app. The app contains no in-app checkout. The license key arrives in the receipt email for the purchase. The header badge changes from "TRIAL PREVIEW · EXPORTS LOCKED" to "LICENSED" when a license is active, and the export center heading in Step Four changes from "What You Can Download with a License" to "What You Can Download", with the sub-line "Your license is active; every export is available." In the trial, that sub-line reads: "In the trial these downloads stay locked." A lock note under the export buttons in the trial reads as follows.

> "Why locked? The trial runs the full workflow on sample data and a capped preview on your own data. Exports require a license. Unlock the Tool"

The five exports

The export center in Step Four lists the five exports in the order below. The order follows the house export standard: formatted reports first, and the plain-data comma-separated file last. Two of the five, the first and the fifth, also appear as quick action buttons in Step Three. Suppression is enforced in every export: the values of a suppressed area are withheld in every file, while its row, its status, and its reason remain. In exported tables, a flagged value is marked with a single asterisk and a suppressed value is marked with two dash characters, as the methods report explains in its section "How to read the tables".

There is no PDF export. No export kind produces a PDF, and no button offers one. The formatted narrative report is the Word methods report, listed second below. State that fact in any project plan that currently promises a PDF from this tool, and plan the Word document as the report deliverable instead.

The first export is labeled "Per-Area Report (.xlsx)" with the detail line "One formatted workbook per area, in the regional-table layout." What actually downloads is a compressed folder, a .zip file named in the form {project}_area_reports.zip, holding one formatted Excel workbook per area. Each workbook presents that area's results in the regional-table layout, formatted to the house export standard for reading and printing. A suppressed area does not receive a workbook of tables. It receives a notice workbook instead, which states that the area values were suppressed under the active preset, so the folder accounts for every area in the run, including the withheld ones. Use this export when each area goes to a different reader, as in a county-by-county release, because each recipient receives a self-contained workbook for their area alone.

The second export is labeled "Methods Report (.docx)" with the detail line "Model, transform, MSE method, and every rule that fired." What downloads is one Word file named in the form {project}_methods_report.docx. Chapter Thirteen describes the document section by section. Use it as the methods record that accompanies any published table from the run, and file it with the run outputs.

The third export is labeled "Master Results Dataset" with the detail line "Every domain and reliability measure, in .xlsx, .csv, and .sav." What downloads is a .zip file named in the form {project}_master_dataset.zip, holding three files: master_dataset.csv, master_dataset.xlsx, and master_dataset.sav. The three files carry the same dataset: every domain in the run, meaning every fitted cell and area combination, with its estimates and reliability measures, in Excel form for reading, in comma-separated form for general use, and in SPSS form for work in SPSS. Intervals appear here as numbers, one of the three places interval figures exist outside the chart bars; the other two are the Results Table (.csv) and the per-area workbooks. Use this export as the analysis dataset of record for the run.

The fourth export is labeled "Reproducibility Package (.zip)" with the detail line "Inputs, configuration, seed, manifest, and the R reproduction script." What downloads is a .zip file named in the form {project}_reproducibility.zip. It holds a configuration file named config.csv and a cells file named cells.csv, indicator and collapse files in comma-separated form when the run used them, a data folder holding the source table as provided, the covariates, and the area rows as the fitting process consumed them, a results folder holding master_dataset.csv, and a script named reproduce.R. The R script recomputes the run with the R survey and sae packages and compares its results against the packaged results. Use this export when a reviewer, an auditor, or a future version of your own project must be able to rebuild the run outside the browser. It is the strongest answer the tool gives to the question of how a figure was produced.

The fifth export is labeled "Results Table (.csv)" with the detail line "The estimates table as shown, with flags and intervals – plain data; CSV carries no formatting." What downloads is one comma-separated file named in the form {project}_results.csv. It carries the estimates table for the selected cell as shown on screen, with flags and with intervals as numbers. Comma-separated form carries no formatting by definition: it is plain data for loading into another program, and its place last in the list reflects the export standard under which the formatted Excel and Word reports are the primary versions and the plain-data file is the companion. Use it for a quick load into a spreadsheet, a database, or a statistical program when you need the selected cell alone rather than the full master dataset.

The license dialog

The license dialog opens from the "Unlock the Tool" button in the header, from the "Unlock the Tool" button in the trial lock note in Step Three, from the corresponding button in the Step Four lock note, or by clicking any locked export button. Its accessible label is "Unlock SAE Studio". At its top stands the brand line "SAE Studio". Below it, an eyebrow reads "Returning Buyer", and a heading reads "Unlock with Your License". Its introduction reads as follows.

> "Enter the email address used at purchase and the license key from your receipt email. The pair unlocks SAE Studio on any machine. Verification sends your email address and the license key only; survey data never leaves the browser."

The introduction states the position exactly. Verification sends both the email address and the license key, as a pair, to the verification service. That pair is the only information the license check sends, and no survey data, results, or file names are part of it. The product page states the same position in its privacy panel: license verification sends the email address and the license key only.

The dialog holds two fields. The first is labeled "Email Used at Purchase" with the placeholder "you@example.org". The second is labeled "License Key" with the placeholder "Paste the key from your receipt email". Two buttons stand below: "Cancel" and "Unlock the Tool". The "Unlock the Tool" button is disabled until both fields hold text. While the check runs, it reads "Checking…". The Escape key closes the dialog, and the Tab key cycles within it.

Help text under the buttons reads as follows.

> "Lost your key? A lost key can be re-sent only to the email address used at purchase, through the lost-key form on the product page. A license is $149 per year, purchased on the product page, and the key arrives in the receipt email."

There is no recovery form inside the dialog and no buy button inside the dialog. Recovery is the lost-key form on the product page, and purchase is the product page. The dialog points to both and performs neither.

To unlock the tool, work through the following steps.

1. Buy the license on the SAE Studio product page, and keep the receipt email, which holds the license key.
2. In the app, press "Unlock the Tool" in the header, or click any locked export. The dialog opens with the heading "Unlock with Your License".
3. Type or paste the email address used at purchase into the field labeled "Email Used at Purchase". Use the purchase address exactly. The pair is matched together, so a correct key with a different address fails in the same way as a wrong key.

4. Type or paste the license key into the field labeled "License Key", exactly as it arrived in the receipt email.
5. Press "Unlock the Tool". The button reads "Checking…" while the verification runs. On success, the dialog shows the message: "Unlocked. Every export and every area is now available." The header badge reads "LICENSED", the export buttons lose their lock symbols, and a capped display expands to every area.
6. Close the dialog and continue in Step Four, where the export center sub-line confirms that the license is active and every export is available.

On later visits, the dialog in its licensed state shows a success panel in the form: "**Licensed.** Unlocked with {email} · license active through {date}." Two buttons stand in that panel: "Remove This License from This Machine" and "Done". Removing the license forgets the license in this browser only. The license itself is unaffected, and the same email and key pair unlocks the tool again on this machine or on any other machine. Use removal before lending a machine, before handing a browser profile to another person, or when you move your work to a new machine and want the old machine returned to the trial state. Chapter Seventeen covers the stronger action, "Delete All Local Data", which clears the license record together with the project.

A successful verification stores its signed pass, holding the email, the key, the pass itself, and the expiry, in the same browser storage as the project. The tool therefore reopens unlocked on later visits in the same browser without a new entry. An expired stored license restores as the trial state, with an expiry message in the dialog on the next visit.

License failures

Every failure message the dialog can show is listed here with its cause and its fix. The messages are quoted exactly, because matching the message on your screen to this list is the fastest diagnosis.

For an invalid pair, the message reads: "That email and license key do not match a license on file. Check both entries and try again." Check the email address first, because a key entered under a different address produces this same message. Then check the key character by character against the receipt email, including every separator. Paste rather than type where you can, and confirm that no leading or trailing space came with the paste.

For a revoked key, identified by the service response for a revoked license, the message reads: "This license key has been revoked. If you bought a license, contact support; a replacement key will be issued to the email address on file." Follow the message. A revoked key cannot be restored by re-entry, and the replacement goes to the email address on file, so make sure you can receive mail at that address before you contact support.

For an expired license, the message reads: "This license has expired. Renew it on the product page to keep using the full tool." Renewal is a product page purchase action. The same message appears in the dialog on a later visit when a stored license has expired since it was saved, so an expiry message at the start of a session means the stored pass aged out, not that anything was deleted.

For a network failure, the message reads: "The license check could not reach the verification service. Check your connection and try again." Your work is unaffected, because the check is the only network call and no work depends on it. Restore the connection and press "Unlock the Tool" again.

For an unavailable service, the message reads: "The verification service is unavailable. Your work is saved in this browser. Try the check again in a few minutes." Wait and retry. The message confirms that your work is saved in the browser, which is where the project lives regardless of the license state.

Two further messages belong to the restored license state rather than to a fresh entry. When the stored pass cannot be found, the dialog shows: "The license stored on this machine could not be found. Enter your email address and license key to unlock again." Enter the pair again in the same dialog. When the stored license has expired, the dialog shows the expired message quoted above. Both states leave the project itself intact and restored as usual.

Lost keys are recovered outside the app. Use the lost-key form on the product page. A lost key can be re-sent only to the email address used at purchase, so recovery requires access to that mailbox. Keep the receipt email in a folder you retain for the life of the license, and record the purchase address in your project notes, because the pair, not the key alone, is the credential.

The Companion Workbook

The Companion Workbook is a standalone Excel file that implements the same Fay–Herriot workflow as the Studio, as Excel formulas. It is included with the license. This chapter describes what the file is, what it holds, how its defaults match the Studio, and how it relates to a Studio run. One clarification stands at the head of the chapter because product descriptions have blurred it: the workbook is not an export from the app. No button in the app produces a copy of the workbook, filled or blank. The five exports are the five files listed in Chapter Fifteen, and the workbook is not among them. Never look for a workbook button in the app, and never plan a workflow that waits for one.

The file is named SAE_Studio_Companion_Workbook.xlsx, in version 1.0. It is a macro-free Microsoft 365 Excel workbook. Its calculations are Excel formulas, including named LAMBDA functions, so the whole computation can be inspected cell by cell in the formula bar and audited with the ordinary Excel tools for tracing precedents and dependents. It holds eighteen sheets. Together, the sheets carry the inputs, the direct estimation inputs, the transformation, the model fitting, the precision calculations, the benchmarking, the disclosure checks, and the presentation tables of the same workflow the Studio runs, in workbook form.

The workbook arrives with the sample data loaded. The same sixteen-county Maine poverty sample that the Studio loads with "Use Sample Data" is entered in the workbook, so every sheet shows a completed run when the file is first opened. Open it first in that state and read it as a worked example: follow one county from its input row through the transformed values, the fitted values, and the final table, and the structure of the workbook explains itself more quickly than any sheet-by-sheet description could. Clearing the demo rows makes the blank form. Delete the loaded sample values from the input rows, keep the formulas and the structure, and enter your own areas in their place. Save the cleared form under a project file name before you enter data, and keep an untouched copy of the delivered file as your master blank, so a damaged working copy can always be replaced by a fresh clearing of the master.

The workbook defaults mirror the Studio defaults. Its starting configuration is Quick mode in effect, the arcsine transformation, bias correction off, benchmarking on with the logit shift method, the disclosure preset Standard with a minimum of ten respondents, a flag at a coefficient of variation of thirty percent, suppression at fifty percent, and a minimum of one primary sampling unit per area. Covariate selection is on, using the BIC criterion. The confidence level is 0.95, matching the Studio ninety-five percent intervals. A workbook run and a Studio run that start from the same inputs and the same settings therefore start from the same method, and differences in their outputs point to a difference in inputs or in a setting changed on one side, which is exactly the comparison the workbook exists to support.

The relationship between the workbook and the Studio is that of a parallel implementation, not a parent and a copy. The workbook alignment record confirms the standing of that relationship: the workbook registered comparison battery was re-run against the shipped Studio engine and recorded 1,193 comparisons with no failures, and the round required no workbook change. The workbook as delivered is aligned with the shipped Studio. Use the two accordingly. Use the Studio for the workflow itself: profiling, role confirmation, selection, fitting, diagnostics, and exports. Use the workbook when a reviewer wants the calculation in spreadsheet form, when your organization requires an Excel artifact for its files, when you need to teach the method by walking its formulas, or when you need to check a Studio figure by an independent implementation that you can inspect cell by cell.

Two limits on the workbook must be stated as plainly as its strengths. First, the workbook shipped state is the standalone file with the demo data loaded. A delivery path that fills the workbook from your own Studio run inside the app does not exist in the shipped app, whatever a specification or a product description may project. To use the workbook with your own data, clear the demo rows and enter your data in the workbook directly, or enter the key figures from a Studio run by hand where a sheet asks for them. Second, an in-Excel recalculation check of the delivered workbook by the owner remains outstanding in the alignment record and is not claimed there. The recorded alignment rests on the comparison battery against the engine. A workbook you modify should be recalculated in Excel and spot-checked against its own completed sample state before its outputs are quoted, as with any spreadsheet model that has been edited.

File handling for the workbook follows ordinary spreadsheet discipline. Keep the delivered file unchanged as the master. Work in copies named for the project. Keep the workbook beside the methods report of the matching Studio run when both describe the same analysis, and note in your project record which of the two is the source of any quoted figure. A figure quoted from the workbook and a figure quoted from the Studio for the same cell and settings should agree within rounding. When they do not, stop and find the input or setting that differs before either figure is published. That disagreement check, used honestly, is worth more than the convenience of either tool alone.

Your Project, Saved in This Browser

SAE Studio keeps your project in the browser you are working in. There is no server copy of your project, your data, your results, or your file names. This chapter describes the autosave, the "Save Project" button, the setting that decides whether the data file itself is kept, the restored states you can meet on a later visit, and the two project controls that start again or clear everything. A short section on appearance closes the chapter, because appearance settings share the same browser-only persistence.

Autosave and Save Project

The project autosaves in this browser as you work. Every change of state is saved after a short pause of 250 milliseconds: the project name, the current step, the source file metadata, the confirmed roles and setup, the run configuration, the quality check and feasibility results, the run result, and the selected cell. A run that is in flight when a save occurs is stored as idle, so a restored project never claims to be mid-run. On the next visit in the same browser, the saved project is restored exactly as left. You reopen on your work, not on a fresh screen, and the header subtitle, the step position, and the results all return with it.

The button labeled "Save Project" in the header forces an immediate save of any pending change and reads "Saved" for two seconds after a click. Treat it as a reassurance control rather than as the only save path. Click it before you close the tab at the end of a session if you want visible confirmation that the latest state is stored. Its temporary "Saved" reading confirms the save completed. Autosave continues regardless, so a closed tab without a click still restores the project as of the last autosaved change.

The standing text in the "Project" card in Step Four states the storage position in the words the app itself uses: "The project autosaves in this browser as you work. No data, results, or file names are sent to a server; the only network call this app makes is the license check." That sentence is the complete data handling account for project storage. The license check sends the email address and the license key, as Chapter Fifteen stated, and nothing else in the app uses the network at all.

Keep the Data File with the Project

The "Project" card in Step Four holds a checkbox with the exact label: "Keep the Data File with the Project", followed by its explanation: "Stored only in this browser, so the project reopens exactly as left. If this is off, the settings and results are kept, but the file itself is not stored." The setting is on by default.

With the setting on, the data table itself is stored with the project, and the project reopens exactly as left, ready to rerun or to adjust and rerun. With the setting off, the settings and the results are kept, but the rows of the table are dropped from storage. A project restored in that state shows its file row in the state described as file not stored, and a run attempt is

blocked with the message: "The data file is not stored in this browser. Add the file again to run." Add the same file again in Step One and the restored settings apply to it, after which the project runs as before.

Choose the off state when the data file is sensitive and the browser is shared, and the settings and results alone are safe to leave in place. Choose the on state, the default, for ordinary work on your own machine, because it is the state in which a restored project is immediately workable. Note that the off state does not delete results: a restored project without its file still displays its previous results, marked under the usual stale rules if settings change, and Step Four still shows its diagnostics and QC report for the selected cell. The file is needed to run again, not to read what was run.

Project name, New Project, and Delete All Local Data

The "Project" card holds a field labeled "Project Name" with a text box. Renaming the project updates the header title at once, because the header title line is the logo followed by a dash and the project name. Name projects for the analysis and the data vintage, in plain words, because the project name becomes the survey name line in the methods report and the base of the export file names. A name chosen for filing, rather than a name accepted by default, pays off in every artifact the run produces.

The button labeled "New Project" starts a fresh project with the default configuration, at Step One, under the name "New project". The license on the machine is kept. A new project is therefore still licensed if the machine was licensed, and its exports are available as soon as it has a completed run. Starting a new project replaces the current project in the autosave, so finish or file the exports of the current project before you press it if those outputs must be kept. The results of the replaced project are not archived anywhere by the app.

The button labeled "Delete All Local Data" is the clearing control. Pressing it arms a confirmation button labeled "Confirm: Delete All Local Data". Pressing the confirmation wipes the project and the license record from this browser and resets the app to its first-visit state. The license itself is not cancelled or revoked by this action. The same email and key pair unlocks the tool again afterward, through the dialog in Chapter Fifteen. Use the full delete when you leave a machine, when a browser profile changes hands, or when a corrupted saved state is suspected and a clean start is the fastest diagnosis. Use the narrower "Remove This License from This Machine" control in the license dialog when only the license should leave the machine and the project should stay.

Restoring has two failure-adjacent states worth recognizing, both covered in Chapter Fifteen for their messages. A stored license that has expired restores the project as a trial project with the full workflow intact and the exports locked. A stored license pass that cannot be found restores the project and asks for the pair again in the dialog. In neither state is project work lost. The project store and the license store sit together in the browser, but they fail separately, and the app reports the license half without disturbing the project half.

Appearance

Appearance settings persist in this browser only, in local storage, separately from the project store. The color mode and the theme color you choose are applied before the first paint on later visits, so the app opens in your chosen appearance without a flash of the default. "System mode" follows the operating system light and dark setting as it changes during a session. Changing appearance never changes a setting, a result, or a saved project, and clearing local project data through "Delete All Local Data" is a project and license action described above. Set appearance once, to the combination that makes long tables easiest for you to read, and leave it. For table-heavy review work, many readers prefer the light mode with the default "Signal Orange" accent, because the chart caption color word, the flag badges, and the coefficient of variation banding were all described in this guide against that default. In a different theme color, the chart accent dots and bars follow your theme, and the caption names that color in place of orange. The numbers do not change.

A final practical note on browsers closes the chapter. Because the project, the license pass, and the appearance all live in one browser, they live and die with that browser storage. Clearing site data for the app address in the browser settings has the same effect as "Delete All Local Data", without the confirmation step. A different browser on the same machine holds a different, empty store, and the project does not follow you between browsers. Work in one browser for a given project, export the files you must keep, and treat the browser store as working storage rather than as the archive. The archive of a run is its exported files: the methods report, the master dataset, and the reproducibility package.

Troubleshooting

This chapter collects the messages the app can give you, grouped by cause, with the fix for each. It covers file refusals and validation messages first, then run blockers, then export messages, then license failures, then the desktop notice, and it closes with the route for asking a question the guide does not answer. Messages are quoted exactly as the app gives them, with the parts that come from your file shown in braces. When a message on your screen matches a message here, apply the fix beside it before you try general remedies. Most problems in this tool are named precisely by the message that reports them.

File refusals

A structural failure refuses the file with a single message and no partial load. A failed load inside the app keeps your license, names the file, and shows the reader message. The generic fallback, when no specific message is available, reads: "The file could not be read."

An empty file gives the message: "This file is empty. Choose a file with data in it." Export or save the source again so that the file holds its rows, and confirm the file size is not zero before you load it again.

A file that cannot be parsed as the format its name suggests gives the message in the form: "This file could not be read as a {format} file. Check that the file is complete and in the format its name suggests." The usual causes are a truncated download, a file renamed to a different extension without conversion, and a spreadsheet saved in a legacy format under a current extension. Open the file in its native program, save it in the named format, and load the saved copy.

An unsupported type gives the message: "This file type is not supported. Use an SPSS .sav, Stata .dta, SAS .sas7bdat or .xpt, Excel .xlsx, or CSV file." Convert the file to one of the listed formats. Text files should be saved as comma-separated or tab-separated files with a .csv or .tsv extension.

A compressed SPSS file gives one of two forms of the same refusal: "This is a compressed SPSS file (.zsav), which is not in the supported format list. Save it as a standard .sav file and try again." or "Compressed SPSS files (.zsav) are not in the supported format list. Save the file as a standard .sav and try again." Save the file from SPSS as a standard, uncompressed .sav file and load that copy.

Validation messages

Validation messages load with the file and are listed under the file row in Step One and again in the "QC Report" card in Step Four. They describe data-level problems in a file that was read successfully. Grouped by cause, they are as follows.

A missing mapped column gives the message in the form: "The file has no column named "{column}", which is needed as the {role}. Check the spelling, or map a different column to this role." The column named in a role assignment does not exist in the file as loaded. Correct the spelling in the file or assign a different column to the role in the "Column Roles" card.

An empty structure gives one of two messages: "The file has columns but no respondent rows." or "The file has columns but no area rows." The header read correctly and no data rows followed it. The usual cause is a file saved from a filtered view that held no rows, or a template saved before data entry. Load a file that holds the rows.

Rows without an area give the message in the form: "{N} respondent rows have no area code in "{column}" and will be excluded (rows {list})." Those rows are excluded from every estimate. Assign their area codes in the source if they belong in the analysis, or accept the exclusion and note the count, because the excluded rows change the respondent base.

Weight problems give two messages. Missing or non-numeric weights give the message in the form: "The weight column "{column}" has {N} rows with a missing or non-numeric weight. Every respondent needs a numeric weight." Weights of zero or less give the message in the form: "The weight column "{column}" has {N} rows with a weight of zero or less. Analysis weights must be positive." Correct the weight column in the source. The tool does not impute weights and does not drop the rows silently: the messages stand until the column is sound or the setup changes.

An outcome coding problem gives the message in the form: "The outcome column "{column}" contains values other than 0 and 1 ({values}). Recode it to 0/1 with an explicit recode rule before running." Recode the values in the source, or mark every value in the rule builder in Step Two so that the mapping is explicit and recorded.

Variance role problems give three messages. When both a variance and a standard error are mapped, the message follows the form: "Both a variance column ("{a}") and a standard-error column ("{b}") were mapped; the variance column is used." No fix is required if the variance column is the intended one. Unassign the standard error role if it was assigned by mistake, to keep the record clean. When neither is mapped, the file loads and the run is blocked in Step Two, as the blocker list below records. Negative values give the message in the form: "The {variance|standard error} column "{column}" has {N} negative values. A {variance|standard error} cannot be negative; check the file." Zero values give the message in the form: "{N} areas have a {variance|standard error} of zero in "{column}". A zero {variance|standard error} claims perfect precision and will distort the model; verify before running." Treat zeros as missing values in disguise until proven otherwise, and replace them with the computed variance or standard error in the source.

Duplicate areas give the message in the form: "The area code(s) {list} appear on more than one row. An area-level file must have exactly one row per area; combine or remove the duplicates." Combine the duplicate rows in the source on substantive grounds, because an average or a sum of two rows for one area is a data decision, not a formatting one.

Missing direct estimates give the message in the form: "No direct estimate for {areas}. Those areas are excluded from the model fit and named in the QC report; their covariates still inform predictions if provided." Accept the exclusion if the areas genuinely have no estimate. Restore the estimates in the source if the absence is a file error.

Out-of-range estimates give one of two messages: "Direct estimates outside the 0-100 percent range for {areas}." or "Direct estimates outside the 0-1 proportion range for {areas}.", with a scale hint appended. The usual cause is a scale declaration that does not match the file. Correct the declaration in the field labeled "Estimates Are Stored As" in Step Two if the file is sound, or correct the values if the file itself holds errors. Chapter Four explained why a proportions file declared as percents produces no range message and must be caught by reading the coverage view against a known figure.

A sample size problem gives the message in the form: "The sample-size column "{column}" is missing or non-numeric for {N} areas. Every area needs its respondent count." Supply the counts in the source. An area-level file without respondent counts cannot support the model transformed variance or the disclosure effective sample size, and no role reassignment substitutes for the missing numbers.

Covariate problems give two messages. Non-numeric values give the message in the form: "The covariate column "{column}" has {N} non-numeric values. Covariates must be numbers." Missing values give the message in the form: "The covariate "{column}" is missing for {N} areas. Areas without a complete covariate row cannot receive model estimates." Clean the column in the source, or set a covariate you cannot complete to "Exclude" in Step Two and accept the selection without it. An area with an incomplete covariate row receives no model estimate while that covariate is in play, so the choice is between completing the data and narrowing the model, and it should be made deliberately.

Frame count problems each carry their own message in the same style. A frame primary sampling unit count must be a positive whole number, the same on every row of its stratum, and no smaller than the number of primary sampling units sampled in that stratum. Correct whichever property the message names, in the source file. Strata with no frame count are treated as sampled with replacement, and that treatment is stated as an issue in the QC report rather than applied silently. If your design was not sampled with replacement, supply the frame counts, because the finite population correction depends on them.

Run blockers and run failures

The seven blocker forms are listed in full in Chapter Ten. In troubleshooting form: if the line under "Run Model →" names a file, load the file or add it again. If it names columns, confirm those roles in Step One. If it names an indicator, select at least one in Step Two. If it names a quality check in the form "A quality check blocks this run: {check message}", resolve the named check and re-run the checks. The button stays clickable under a blocker by design, so a click that produces the blocker text in the error line is the app reporting the blocker, not a failed fit.

A failed run shows its error in an alert line under the buttons, with the fallback text "The run failed unexpectedly." when no specific error is available. Correct the cause named in the error, or, for the fallback, return to the feasibility preview and the quality checks, confirm the roles and the scale declaration, and run again. A small cell that does not fit is not a run failure: it is a completed run in which that cell reports direct estimates, with the reason in the run report, as Chapter Ten and Chapter Eleven described.

A stale results display is not an error. If Step Three opens with the note "Settings changed since this run. These are the previous results; run the model again in Step 2 to update them.", a setting changed after the run. Run again in Step Two before you quote or export any figure.

Export messages

A locked export in the trial opens the license dialog and produces no file and no error message. That behavior is the design, and the remedy is the license in Chapter Fifteen. The store-level statement of the locked state reads: "Exports require an active license. Enter the email address used at purchase and the license key from your receipt email to unlock every export for this project."

An export attempt with a license and no completed run gives the message: "Run the model first; exports are generated from a completed run." Run the model, confirm in Step Three that the results are not stale, and export from Step Four.

A suppressed value missing from an export is not an export error. Suppression is enforced in every export, the value is replaced by the two-dash marker in tabular files or by a notice workbook in the per-area export, and its row, status, and reason remain. To change the outcome, change the preset or the data on stated grounds and rerun. Editing an exported file to restore a suppressed value defeats the disclosure rule the run applied, and the methods report records the suppression count, so the restored value would disagree with the run record.

A missing interval column in the on-screen table is not an export problem either. The screen table has no interval column by design. Intervals are in the chart bars on screen and in the numbers in the "Results Table (.csv)", the master dataset, and the per-area workbooks after export.

License failures

Chapter Fifteen lists all five license failure messages in full with their fixes, plus the two restored-license messages. In short form: an invalid pair message means the email and the key do not match a license together, so check the purchase email address before the key. A revoked message means a replacement must be issued to the address on file. An expired message means renewal on the product page. A network message means check the connection and retry. A service unavailable message means wait a few minutes and retry, with your work safe in the browser throughout. A lost key is recovered only through the lost-key form on the product page, and only to the purchase email address. There is no recovery form and no buy button inside the app, by design.

The desktop notice and display problems

On a screen narrower than 760 pixels, the notice appears above the app with the text: "SAE Studio is designed for desktop use. The full workflow requires a larger screen. Open this address on a desktop or laptop computer to run the complete tool." Its close button carries the accessible label "Dismiss desktop notice". Dismissal lasts for the browser session only, so the notice returns in a new session on the same narrow screen. The remedy is the one the notice states: open the address, https://app.maineblock.com/, on a desktop or laptop computer. Tables on a narrow screen also hide the columns "n", "Direct SE", and "Shrinkage γ", so a table that looks short of columns on a phone is the reduced display, not a failed run. Judge column problems on the desktop display.

If the app opens on a fresh project when you expected your saved work, check three things in order. First, confirm that you are in the same browser, because the store does not follow you between browsers. Second, confirm that site data for the address was not cleared, in the browser settings or by a cleanup tool, because clearing it has the same effect as "Delete All Local Data". Third, confirm that you did not start a "New Project" or confirm a delete at the end of the previous session. A restored project without its file is a different state, covered above: its results remain readable, and adding the file again restores the ability to run.

If the theme or the color mode looks wrong on a new visit, reset it in the header. The "Color mode" switch and the "Theme color" button are described in Chapter One and Chapter Seventeen. Appearance lives in the same browser storage family as the project, so a storage clearing resets appearance together with the project, and both need setting again in that case.

JavaScript is required. A browser with JavaScript disabled shows the fallback line: "SAE Studio needs JavaScript to run." Enable JavaScript for the address and reload. Before JavaScript loads on any visit, a static shell with the heading "Small area estimation without writing code" and a short how-it-works list is visible briefly. The running app replaces it whole. A page that stays on the static shell has failed to load its script, and reloading the address is the first remedy.

Where to ask a question

If a message on your screen matches nothing in this chapter, or a fix here does not clear the problem, ask through the contact route on the SAE Studio product page at maineblock.com. Include, in your message, the step you were on, the exact message as it appeared, the file kind you loaded, and whether a license is active on the machine. Do not include your license key in the message, and do not attach a data file that holds respondent-level records. A question that names the step and quotes the message can usually be answered from this guide chapter that covers that step, and the fastest route to that answer is the information above. Questions about purchase, renewal, revocation, and key reissue belong to the same product page route, because purchase and recovery live on the product page rather than in the app.

The closing advice of this guide returns to its opening pattern. Load, confirm, configure, preview, run, read, check, and export, in that order, and read every message the app gives you in the place it appears. The app names its own problems with unusual precision. A user who follows its messages, checks the statewide figure against the population-weighted basis, respects the suppression markers, and files the methods report with every published table will produce small area estimates that a reviewer can follow from the file to the final number, which is the standard this tool was built to meet.