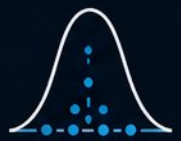
How to Use Statistics in Data Science Projects: A Practical Workflow

- Statistics is the guiding framework from question to deployment, not an afterthought
- Real 2025–2026 failures: $220K fraud loss from silent schema corruption
- A fall-detection model scored 94% accuracy — honest evaluation revealed 69%
- Pearson dispersion bias made a Tweedie model predict 99%+ zeros against 93% observed
- This training delivers a repeatable, project-friendly statistical workflow

Why Statistical Rigor Determines Project Success or Failure

- Silent schema corruption in a 12TB fraud pipeline: 0.82 accuracy masked $220K in annual losses

- Fall-detection model reported 94% accuracy; actual performance was 69% due to frame-level data leakage



- Tweedie model predicted >99% zeros against 93% observed—catastrophic Pearson dispersion bias

- Predictive metrics alone are dangerously misleading without sound statistical foundations

Framing the Problem:
From Business Question to Testable Hypothesis

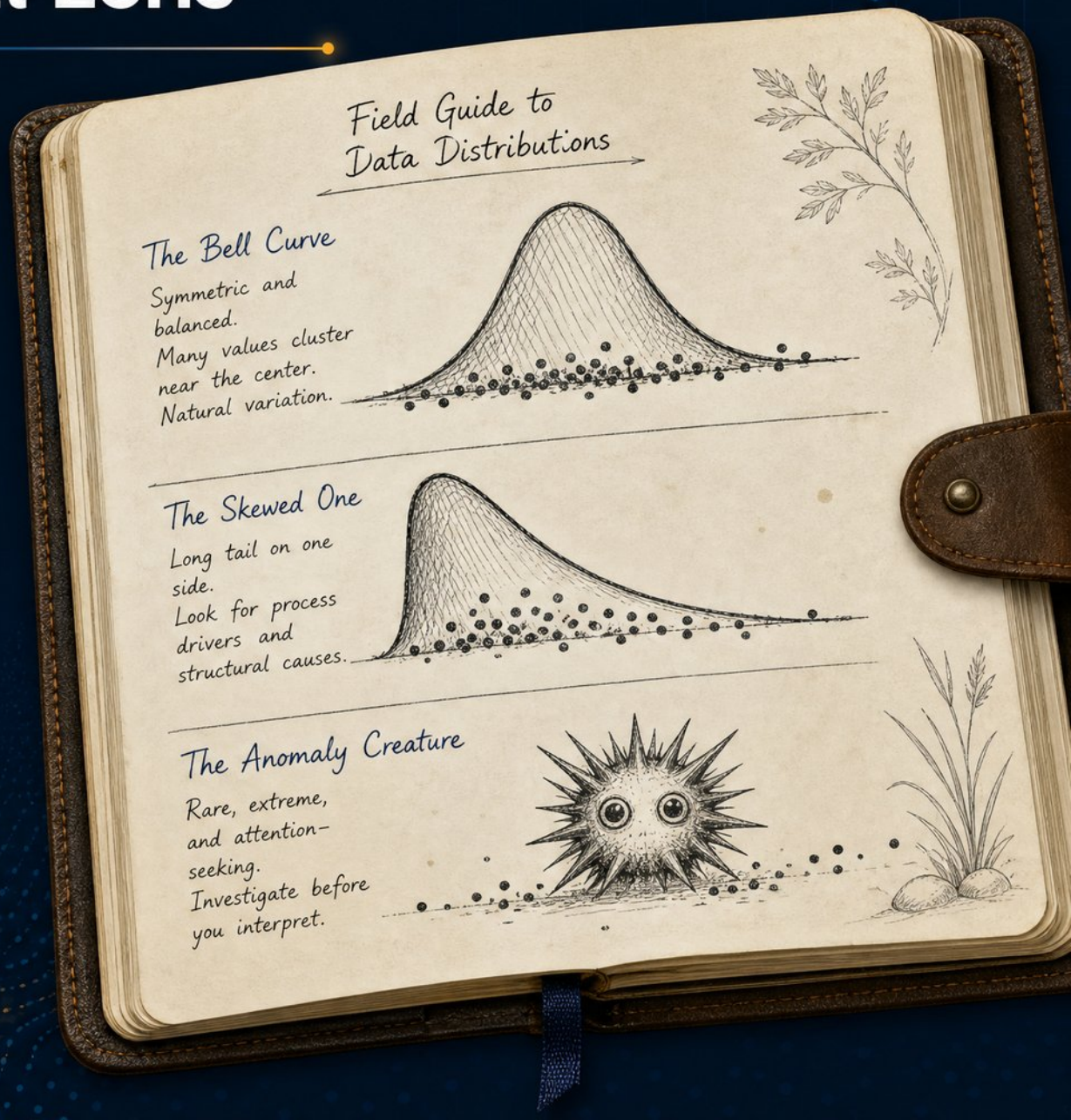
- Turn vague goals into measurable, testable statistical hypotheses
- Choose estimation, hypothesis testing, or prediction as your strategy
- Map business KPIs to statistical parameters with decision rules
- Spot poorly formed questions before they waste analysis time

Sampling and Data Collection Design for Reliable Inference

- Probability vs. non-probability sampling defines inference scope — know the difference before data collection.
- Power analysis balances sample size, effect size, alpha, and power to avoid underpowered studies.
- Identify selection bias, measurement error, and confounding at the design stage.
- Modern planning tools: browser calculators, Python simulation, and R libraries like Spower.

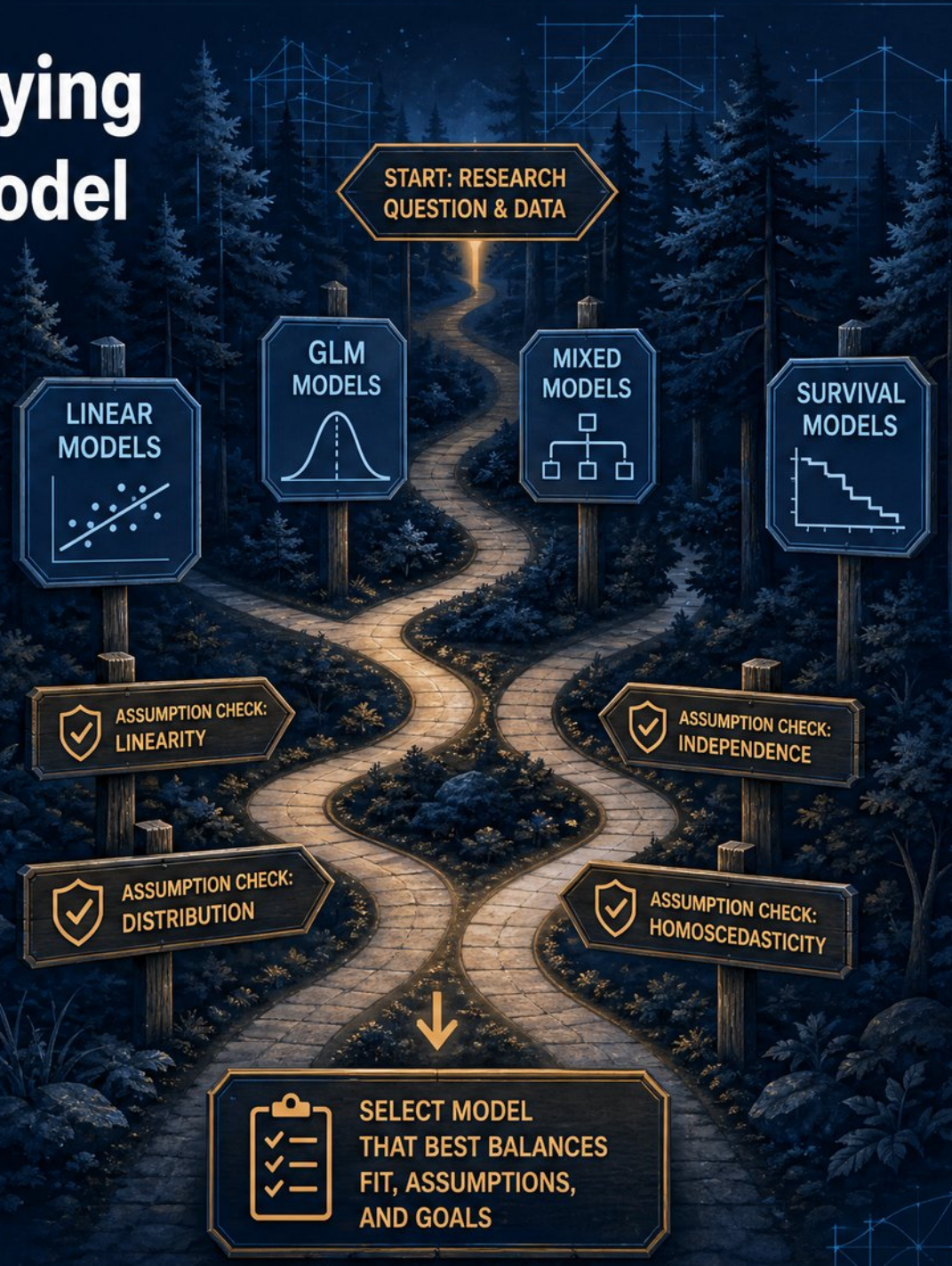
Exploratory Data Analysis with a Statistical Lens

- Choose summaries and plots that match variable type and scale
- Spot anomalies, missing-data patterns, and distributional surprises early
- Use EDA to sharpen hypotheses, never to confirm them
- Guard against over-interpretation with principled data-snooping safeguards



Choosing and Specifying the Right Statistical Model

- Map research question and data structure to model families (linear, GLM, mixed, survival)
- Write model equations with explicit assumptions, link functions, and distribution families
- Balance interpretability, complexity, and computational cost with real trade-off examples
- Follow a decision flowchart for model selection and assumption checks



Validating Models and Quantifying Uncertainty

- Diagnose residuals and influence with visual, assumption-focused checks
- Assess generalizability using cross-validation and bootstrap strategies
- Build confidence and prediction intervals to quantify model uncertainty
- Communicate uncertainty clearly to non-technical stakeholders
- Reproduce validation workflows with modern Python and R code

From Inference to Action: Interpreting and Reporting Results

- Statistical vs. practical significance: always report effect sizes
- Translate odds ratios and coefficients into business impact
- One-pager template: question, method, result, decision
- P-values are not probability of null; CIs are not probability intervals

Common Statistical Traps and How to Avoid Them

- P-hacking, HARKing, and overfitting through real-world examples
- Confounding, collider bias, and Simpson's paradox explained visually
- Why prediction-driven variable selection yields wrong treatment effects
- Use a practical self-audit checklist for your own statistical work

Case Study: Debugging a Broken Causal Pipeline

SIGNAL vs. NOISE RESONANCE

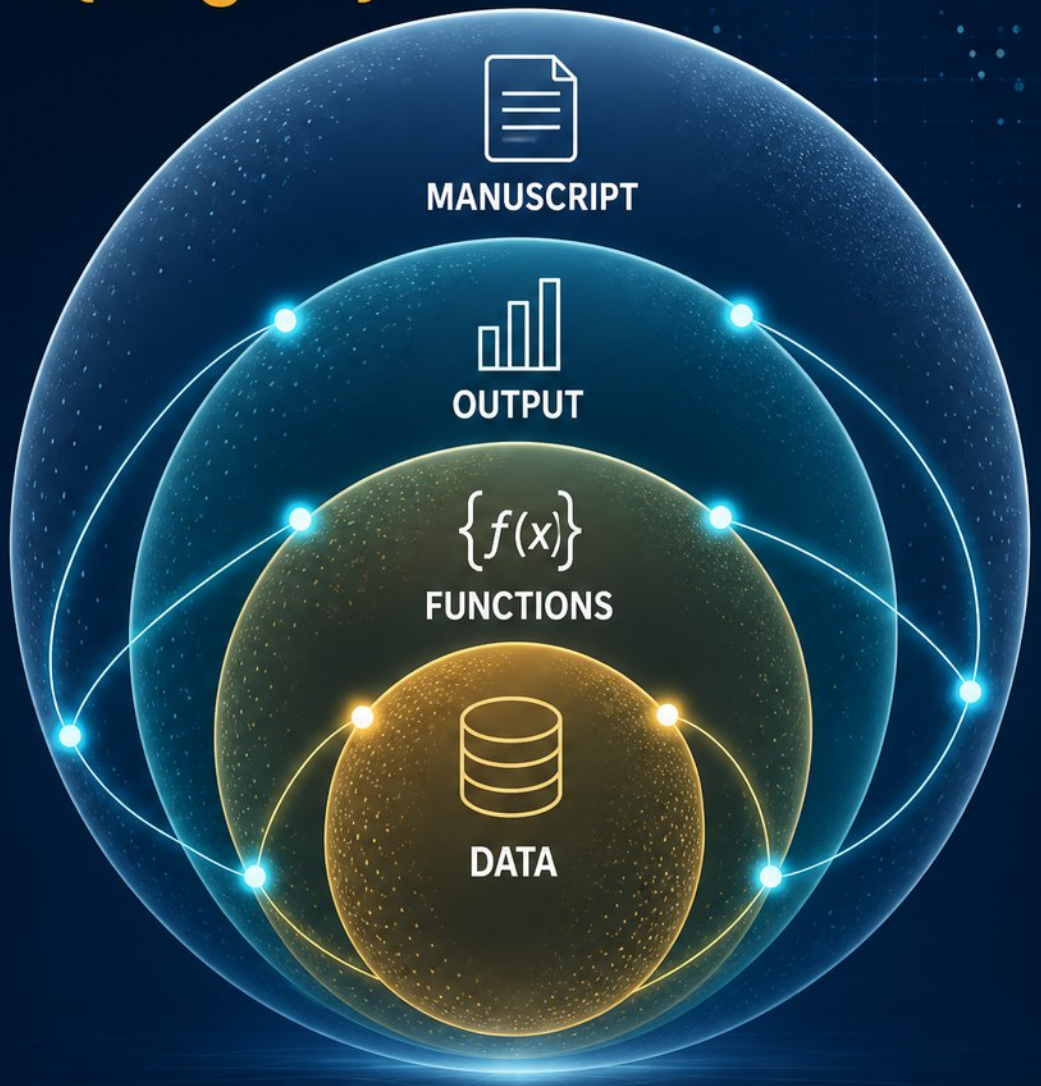
- DML price elasticity model: R^2 of 0.003 reveals deep pipeline failure
- Frisch-Waugh-Lovell theorem detects double-removal of critical variation
- Weakened instruments amplify bias, destroying causal signal
- Diagnose nuisance model performance; separate R^2 interpretation from prediction tasks

Automating and Reproducing the Statistical Workflow

- Version-controlled pipelines: Git, DVC, Docker from raw data to report
- Integrate code with docs: Quarto, Jupyter, R Markdown, {targets}
- Lock dependencies with renv for long-term reproducibility
- Programmatically test analytical code and validate outputs

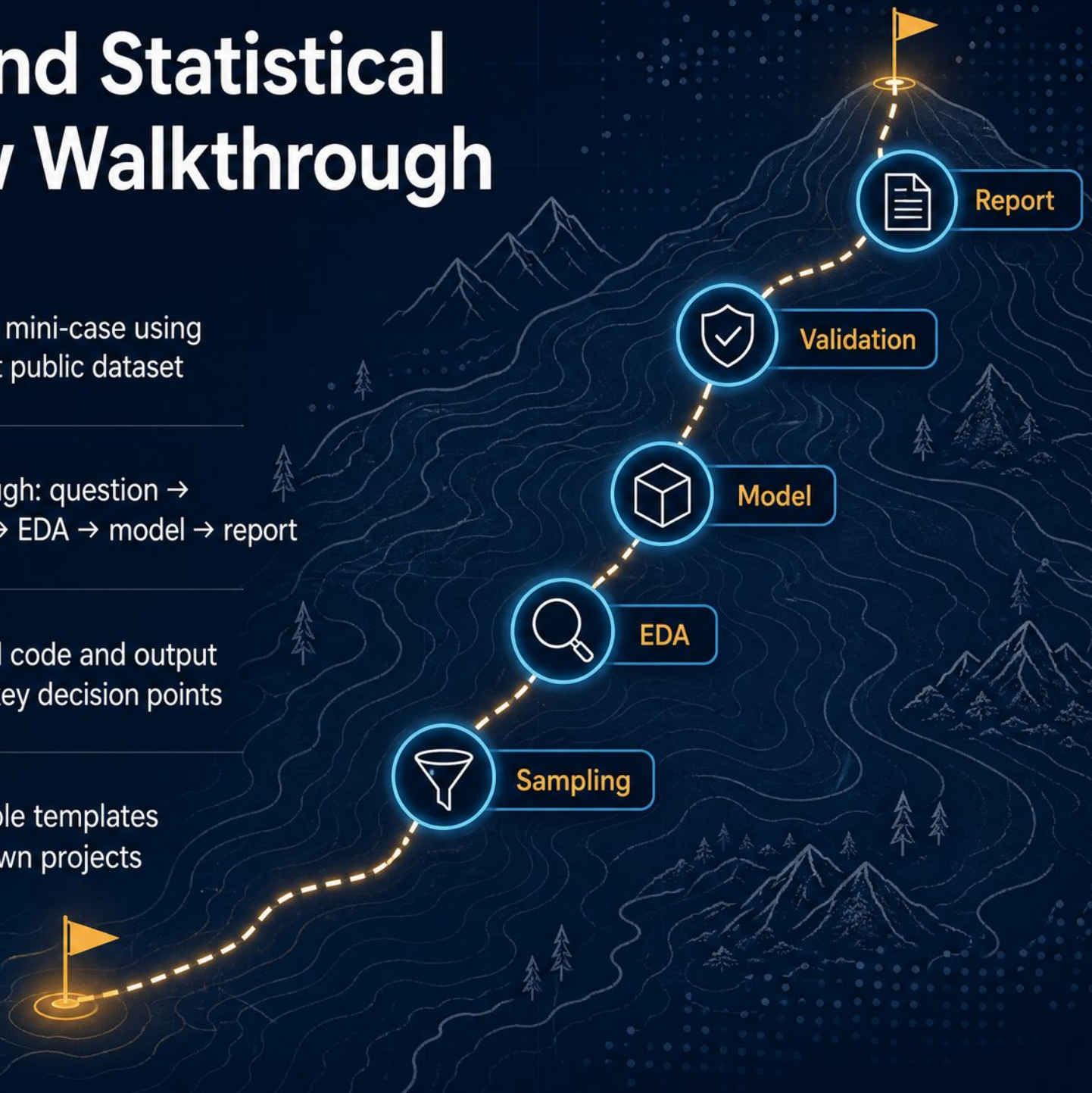
Building Reproducible Research Projects with Quarto and {targets}

- **Quarto:** one source for manuscripts, slides, and websites (Python, R, Julia).
- **{targets}:** pipeline with automatic dependency tracking for objects and files.
- **renv** locks R packages for a fully reproducible environment.
- **Template:** data cleaning → modeling → manuscript → website.
- **Keep experiments out:** playground vs. production pipeline separation.

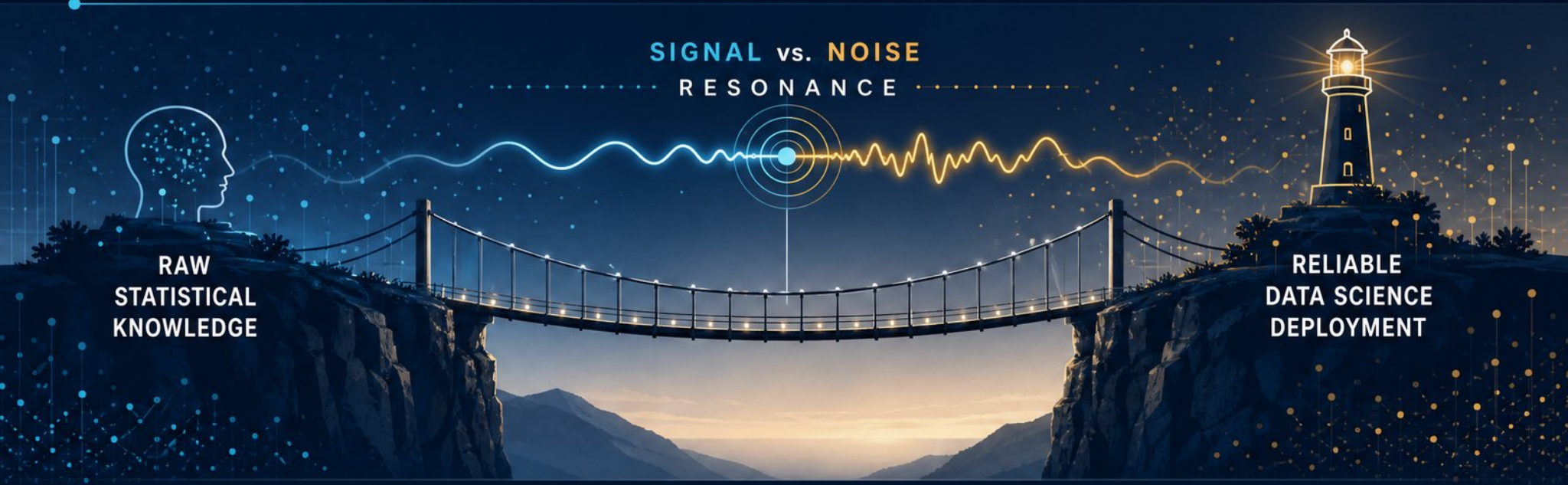


End-to-End Statistical Workflow Walkthrough

- Hands-on mini-case using a compact public dataset
- Walkthrough: question → sampling → EDA → model → report
- Annotated code and output highlight key decision points
- Transferable templates for your own projects



Key Takeaways and Your Next Steps



- Integrate statistics as a workflow from design through deployment
- Choose metrics aligned with actual decisions, not generic defaults
- Audit for data leakage, silent corruption, and evaluation mismatches
- Adopt Quarto, {targets}, renv, and version control by default
- Explore curated resources to deepen your statistical practice

PersonWise.

PersonWise • AI digital-human course platform

Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/3b993117/public