Verasight Data Scientist Case Study

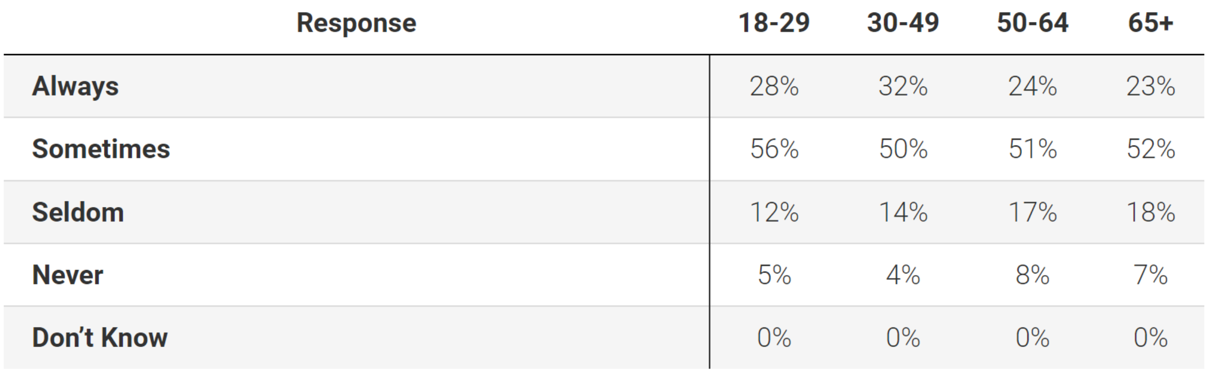
Attached, you will find four files:

1. A sample of registered Verasight panelists (users.rds), who take repeated surveys after creating their accounts
2. A dataset with responses from one specific recent survey (2024-054_responses.rds)
3. A reference file with information about the survey questions in the response dataset (2024-054_reference.rds)
4. An aggregate response dataset with survey responses from many 2024 Verasight projects (full-response-db.rds)

Using these files, complete the following tasks in R and comment the code appropriately. Please feel free to consult external sources. If you use external sources, we ask that you also send a list of the sources used. When finished, please send all code and output to careers@verasight.io. **Please do not spend more than 3 hours on this case study.** If you don't know how to do a certain step of the case study, please outline the questions that you would ask our team to be able to complete it. If you run out of time, please briefly describe the steps you would take to complete the remaining tasks.

Task 1

Use iteration tools (e.g., `map`, `apply`, or `for` loop) to produce a proportion table like the following for each survey item and demographic variable in the recent response dataset (2024-054_responses.rds). If possible, use the `weight` variable to make these weighted proportion tables. For this task, do not write individual code for each table—rather, use one or more tools to create the tables iteratively.



Response	18-29	30-49	50-64	65+
Always	28%	32%	24%	23%
Sometimes	56%	50%	51%	52%
Seldom	12%	14%	17%	18%
Never	5%	4%	8%	7%
Don't Know	0%	0%	0%	0%

1. The demographic variables that should be included are `age_group4`, `education`, `gender`, and `raceeth`.
2. The survey outcomes are all items starting with "q" (e.g., `q28`).
3. If possible, as part of the report, include the label/question wording for each survey question from the 2024-054_reference file.

Task 2

When analyzing data from our survey respondent community, we are sometimes interested in a concept we refer to as “density.” This concept helps us measure whether our survey responses are concentrated among a small number of panelists, or spread out more evenly over a larger number of users/panelists.

1. Using the aggregate response database (response-db.rds), please answer the following question: Out of all Verasight users in the database, what is the smallest percent of users who can account for 50% of all recorded survey responses?
2. Calculate the same metric, but restricted to Verasight users who registered within the last 90 days.
3. Create a visualization showing the density in more detail – in other words, visualize the percent of surveys accounted for at various percentiles of response contribution. (For example, what is the minimum percent of users who account for 10%/20%/30% of survey responses?)

Task 3
Verasight runs over 100 survey projects each year. Each lives in its own subfolder, typically with a nested `data/` folder containing raw and/or cleaned respondent-level data. While project structures are broadly similar, there are minor inconsistencies across folders (e.g. file naming, schema variations).

Each survey includes quality control fields such as:

- Attention check flags
- Speeder indicators (e.g. fast completes)
- Straight-lining

Currently, these fields are reviewed on a project-by-project basis. However, we want to track these quality signals across all projects to better understand patterns at:

1. The respondent level
2. The project level
3. The overall/panel level

Sketch out your approach to building a pipeline or repeatable process for this use case. Please address the following:

- How would you prototype a working version using a sample of 5–10 projects in one week?
- If given 2–3 months, how would you scale this into a more robust and maintainable solution?
- How would you ingest, clean, and combine data across projects with slightly inconsistent structures?
- What tools would you use for version control and workflow automation?
- How would you structure outputs (e.g., summary tables, dashboards, monitoring files)?
- How would you ensure that your approach is maintainable, auditable, and extensible as new projects and quality indicators are added?

You're encouraged to include any, but not necessarily all, of the following:

- Brief code snippets or pseudocode
- Diagrams or logic outlines
- Assumptions you're making
- Notes on tradeoffs, edge cases, or failure points

We're particularly interested in your approach, technical thinking, and how you balance short-term delivery with long-term sustainability.