AD12

System design considerations for automating HTA statistical analysis

Christian Haargaard Olsen, Novo Nordisk A/S, Copenhagen, Denmark
Nicolai Skov Johnsen, Novo Nordisk A/S, Copenhagen, Denmark

1 Abstract

Health Technology Assessment (HTA) submissions often require thousands of analyses per submission, which can be repetitive and highly standardized. This process is time-consuming for programmers and computationally intensive, requiring significant code re-writes for each submission despite functional overlap. At Novo Nordisk, we are developing a system to automate statistical analyses for HTA submissions, using the German AMNOG process as a guide.

The list of desired system features is extensive, but accommodating all of them would make the system inflexible and development impossible. To address this, we have adopted a modularized approach that focuses on the characteristics of different components. We start with foundational pieces and leave flexible specifics for later. This approach allows us to reduce the resource burden and streamline the development process. The Minimum Viable Product (MVP) we've developed currently comprises two modules: one for orchestration and another for statistical methods. Despite the tool's preliminary and unfinished state, it has already proven its efficiency by halving the programming resources needed for two AMNOG submissions.

2 Introduction

HTA submissions often require thousands of analyses per submission, which can be repetitive and highly standardized, leading to significant code re-writes for each submission despite functional overlap.

Given the highly structured requirements of HTA submissions, they are ideal candidates for automation. At Novo Nordisk, the HTA department in Biostatistics collaborated with the Data Science Automation Group to develop a tool in R that facilitates such analyses. The initial work was based on the German AMNOG process (Arzneimittelmarkt-Neuordnungsgesetz, or in English, Pharmaceutical Market Restructuring Act), which has been used as a guide for the development of the tool. While the tool provides the greatest benefit for repetitive analyses, the framework supports any type of statistical analysis.

3 Process requirements

A central requirement of the German AMNOG is that analysis should match the German population intended for the drug, and preferrably be compared to current available treatment options in Germany. Furthermore requirements for efficacy and safety data dictates how data should be filtered, grouped and analysed to provide the desired overview for the authorities. At Novo Nordisk we use an external collaborator to help put together the dossier, but do the statistical analysis internally based on requests from the collaborator.

4 Requirements

Our initial brainstorm on what wishes we had for such a tool lead to an immense list of requirements - some more tangible than others. See Figure 1. Looking at the long list we decided early that this was something we needed to build gradually, and which should be flexible if (when) we discovered the need for new functionality.

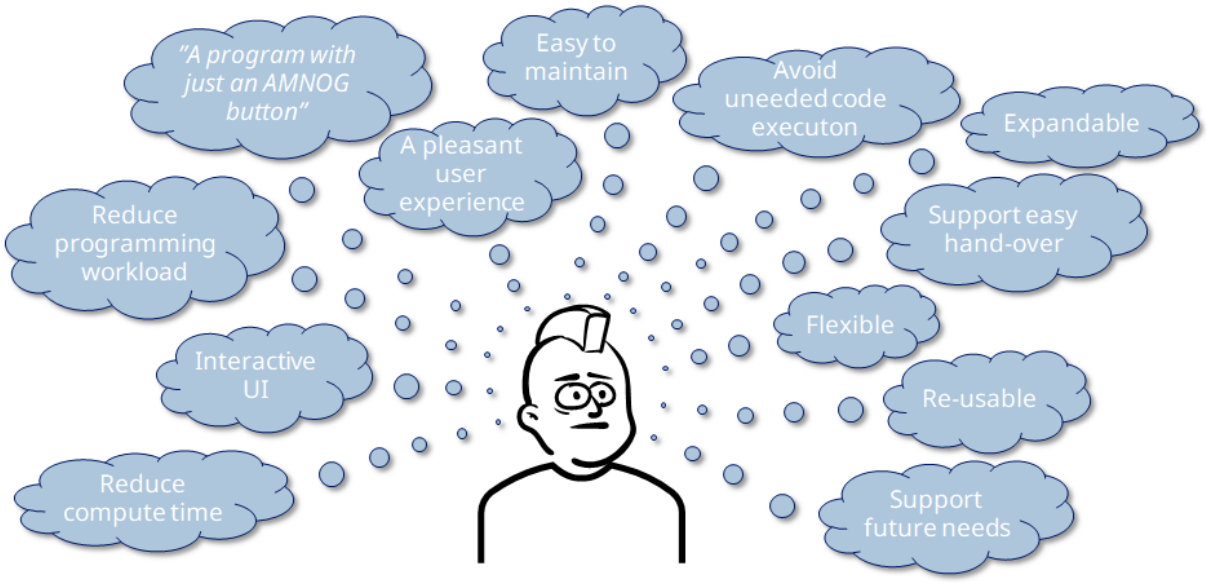

Figure 1: Results from our brainstorming on features for our tool

Upon reviewing the wish list, we identified that some wishes dictated the choice of technologies for building the solution, while others influenced the design of the tool's infrastructure. This was not only to directly support certain wishes (e.g., expandability), but also to enable us to divide the project into smaller, more manageable tasks. This approach ensures that our tool can evolve and adapt as needed, while keeping the development process efficient and feasible.

5 Dissection of requirements and tool

We decided to work towards an framework/ecosystem of packages that could work together, but which could also function separately for smaller tasks. We decided that any component of the system should serve one purpose so each component would be focused while also allowing users to provide custom modules if they were unhappy with what we made available. For now we have split the initial requirements into six planned modules:

- **Input:** A module for specifying the required analysis
- **Orchestration:** A module for slicing and filtering data and applying the specified statistical methods to the resulting data cuts.
- **Statistics:** A module containing the statistical methods.
- **Output:** A data model for storing the raw results (numbers).
- **TFL:** A module (or several) to generate the actual outputs required the submissions.
- **Visualisation:** A module/App to explore the results interactively.

Because the statistical analaysis is the primary objective of the tool, and because earlier programmers on earlier AMNOG submissions spent a significant amount of time orchestrating and keeping tabs on populations, filters and subgroups, we focused our efforts on the Orcestration and Statistics modules.

5.1 Orchestration module

We built our tool using R because it is trusted in the pharmaceutical industry and because it is functional programming language, which makes it a good choice for modular development. We rely on the `targets` package by Will Landau to do a lot of the bookkeeping, cache results of calculations and make parallelization easy, And on `data.tables` to make data wrangling fast.

The input for the module is a data table with specific variables, each following a pre-specified format. Because the module has these clear requirements on the input, users can build the input specification as they like. In figure 2 an endpoint definition in raw R code is shown.

```
data.table::data.table(
  project = "nn1234",
  trial = "nn1234-0001",
  instance = "current",
  pop_var = "SAFFL",
  pop_val = "Y",
  treatment_var = "TRTA",
  treatment_refval = "Xanomeline High Dose",
  period_var = "SAFFL",
  period_val = "Y",
  endpoint_filter = "!is.na(AEDECOD) == TRUE",
  endpoint_group = list(list(AEDECOD=c())),
  subgroups = list(c('SEX', 'AGEGR1')),
  endpoint_label = "Example 1: Adverse events safety analysis set ",
  adam_fun_nm = "mk_adae",
  base_stats_set = list(c("N_sub","N_subev","P_subev")),
  analysis_stats_set = list(c("OR","RR","RD","Pval")),
  criterion_endpoint = "",
  criterion_subgroup_description = "",
  criterion_subgroup_analysis = ""
)
```

Figure 2: Example specification for summary of AEs by code, split on SEX and AGEGR1, calculating number of events, number of subjects with events, fraction of subjects with events, odds ratio, relative risk, risk difference and Barnards test. This specification alone results in 18,624 statistical results when using example data from the R-package `safetyData`. The criterian arguments can be used to criteria when to include or exclude results.

The tool is built as a pipeline where each step enrich the internal data models based on inputs and earlier enrichments. See Figure 3 for an illustration of the steps in the pipeline.

5.2 Statistics module

The statistics module consists of methods to calculate different statistics: From simple counts and percentage to statistical tests. Each method are then referenced by the orcestration module as defined in the analysis specification. The methods are very simple in that they receive a dataset and names of the treatment variable, the variable to do grouping by, the subject id variable, and the name of a variable that flags records for inclusion in the analysis. The statistical method then returns a dataset with treatment

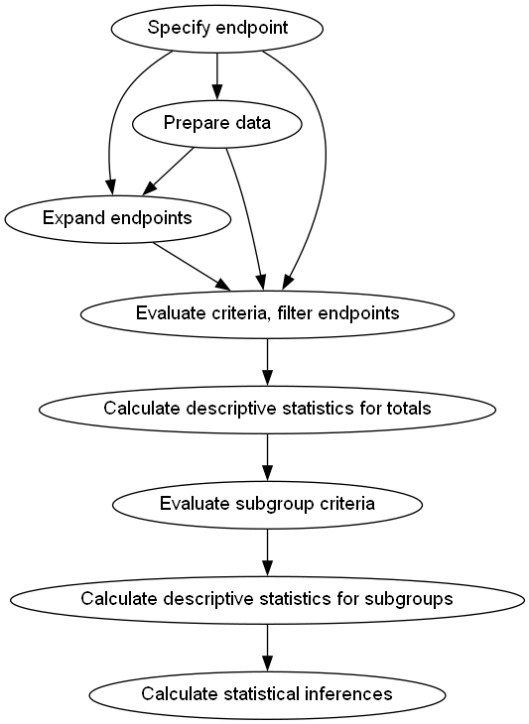

Figure 3: Illustration of the internal pipeline in the orchestration module.

variable, grouping variable and a numerical value. See Figure 4 for an example of the code that calculates number of subjects, table 1 for an example of input data `ax` and table 2 for the corresponding results, when calling the function as shown in Figure 5.

```
N_sub <- function(dat,
                  treatment_var,
                  subgroup_var,
                  subjectid_var = "USUBJID",
                  keepflg_var = "event_flg") {
  dat |>
    unique(by = c(subjectid_var, treatment_var, subgroup_var)) |>
    .[, .(value = .N), by = c(treatment_var, subgroup_var)]
}
```

Figure 4: Underlying code for the statistics module calculating number of subjects. Note the use of the `data.table` package for the last step.

```
result <- N_sub(dat = ax,
                treatment_var = "TRTA",
                subgroup_var = "AGEGR2")
result <- result[order(TRTA, AGEGR2)].
```

Figure 5: Call to the `N_sub` method. Note that during runtime the framework would make the call, so calling the function directly is only done here for the purpose of illustrating the structure of the interface

Because the interface between the package is specified, it is easy to add additional methods as needed while still leveraging the overall framework. This design ensures the module's flexibility and adaptability, allowing it to accommodate a wide range of statistical needs.

Table 1: Example input data `ax` for the `N_sub` method.

TRTA	USUBJID	AGEGR2
A	1	18<= to <65 years
A	1	18<= to <65 years
A	2	65<= years
A	3	18<= to <65 years
A	4	18<= to <65 years
A	5	65<= years
A	6	18<= to <65 years
A	6	18<= to <65 years
B	7	18<= to <65 years
B	8	18<= to <65 years
B	9	18<= to <65 years
B	10	18<= to <65 years
B	10	18<= to <65 years
B	10	18<= to <65 years
B	11	18<= to <65 years
B	11	18<= to <65 years

Table 2: Results from `N_sub` on the `ax` dataset from table 1.

TRTA	AGEGR2	value
A	18<= to <65 years	4
A	65<= years	2
B	18<= to <65 years	5

6 The tool in practice

After developing the MVP based on completed AMNOG submissions from the past, we have deployed the tool on two separate projects: for Safety and Efficacy endpoints. The tool takes some time to get used to, as some of the users are both transitioning from SAS and from are more traditional direct programming approach with a REPL - as opposed to the indirect approach of the `targets` package where code is invoked through the start of a pipeline. In addition we have had to built task-specific and very rough tools for the specification and the reporting steps as these modules have not been developed yet. Despite these challenges, the tool has helped dramatically decrease the time spent on completing the analysis for these two AMNOG dossiers: A reduction in the order of 50%.

7 Conclusion

We have built a tool that can produce a lot of outputs through a simple specification. By specifying one desired endpoint we can obtain all the relevant statitics for that endpoint, for all relevant populations, for all relevant groups. The design is modular, so each module can be used individually, combined as desired or expanded with custom components. We have focused on the computational infrastructure and the statistical methods, but even with very rough interfaces we have been able to reduce the workload of the statistical analyses for the AMNOG dossier significantly.

8 Contact

Contact the authors at:

- Christian Haargaard Olsen, Novo Nordisk A/S, Copenhagen, Denmark, cino@novonordisk.com
- Nicolai Skov Johnsen, Novo Nordisk A/S, Copenhagen, Denmark, nosj@novonordisk.com