




A fast algorithm to compute a curve of confidence upper bounds for the False Discovery Proportion using a reference family with a forest structure

Guillermo Durand ¹ Laboratoire de Mathématiques d'Orsay, Université Paris-Saclay

Date published: 2024-06-27 Last modified: 2024-06-27

Abstract

This paper presents a new algorithm (and an additional trick) that allow to compute fastly an entire curve of post hoc bounds for the False Discovery Proportion when the underlying bound $V_{\mathfrak{R}}^*$ construction is based on a reference family \mathfrak{R} with a forest structure à la [Durand et al. \(2020\)](#). By an entire curve, we mean the values $V_{\mathfrak{R}}^*(S_1), \dots, V_{\mathfrak{R}}^*(S_m)$ computed on a path of increasing selection sets $S_1 \subsetneq \dots \subsetneq S_m, |S_t| = t$. The new algorithm leverages the fact that going from S_t to S_{t+1} is done by only adding one hypothesis.

Keywords: multiple testing, algorithmic, post hoc inference, false discovery proportion, confidence bound

Contents

1	Contents	
2	1 Introduction	2
3	1.1 About this document	2
4	1.2 Setup a github repository for preparing your submission	2
5	1.3 Quarto	2
6	1.4 Requirements	2
7	1.5 Link with your usual tools	2
8	2 Formatting	3
9	2.1 Basic markdown formatting	3
10	2.2 Mathematics	3
11	2.2.1 Mathematical formulae	3
12	2.2.2 Theorems and other amsthm-like environments	3
13	2.3 R Code	4
14	2.4 Figures	4
15	2.5 Tables	4
16	2.6 Handling references	5
17	2.6.1 Bibliographic references	5
18	2.6.2 Other cross-references	5
19	2.7 Advanced formatting	5
20	3 Finalize your submission	6
21	3.1 Handle R dependencies with renv	6
22	3.2 Continuous integration	6
23	3.3 Data and large files	7

¹Corresponding author: guillermo.durand@universite-paris-saclay.fr

24	References	7
25	Session information	7

26 1 Introduction

27 1.1 About this document

28 This document, accompanied with the [hopefully finely tuned git repos](#), provides a template for
 29 writing contributions to **Computo** ([Computo Team, 2020](#)). We show how R code ([R Core Team, 2020](#))
 30 can be included and how the repository can be set up for triggering github actions for rendering the
 31 document, with dependencies handled by `renv`.

32 1.2 Setup a github repository for preparing your submission

33 You can start by clicking the “**use this template**” button, on the top of the page of the [github](#)
 34 [repository associated to this document](#). Of course, you can set your repository private during the
 35 preparation of your manuscript.

36 1.3 Quarto

37 [Quarto](#) is a versatile formatting system for authoring documents integrating markdown, LaTeX and
 38 code blocks interpreted either via Jupyter or Knitr (thus supporting Python, R and Julia). It relies on
 39 the [Pandoc](#) document converter.

40 1.4 Requirements

41 You need [quarto](#) installed on your system and the [Computo extension](#) to prepare your document. For
 42 the latter, once quarto is installed, run the following to install the extension in the current directory
 43 (it creates a `_extension` directory which is ignored by git thanks to `.gitignore` by default):

```
quarto add computorg/computo-quarto-extension
```

44 R and the following R packages must be installed on your computer: [knitr](#), [markdown](#).

45 1.5 Link with your usual tools

46 Quarto is expecting a `.qmd` markdown file, but will also works with a standard [Rmarkdown](#) (`.Rmd`)
 47 file. In addition, especially if you are not comfortable with the command line interface, quarto is
 48 fully integrated inside the [Rstudio IDE](#) so that you can write and build your quarto document inside
 49 Rstudio.

50 Quarto can also process a [Jupyter notebook](#) file if you are used to it (it will just require to add the
 51 proper YAML metadata²).

52 **Note:** *More advanced Jupyter-related functionality like `Myst/Jupyter book` are not supported in this*
 53 *Quarto setup. The markdown syntax inside the Jupyter notebook should follow the Quarto syntax (c.f.*
 54 *[below](#)). If you are more comfortable with using `Myst/Jupyter book`, we provide a [specific template](#) but it*
 55 *will requires more formatting work for Computo editorial team, thus highly encourage authors to use*
 56 *the Quarto templates.*

²the same metadata as in the [template-computo-R.qmd](#) file in the first cell, type “Raw”, of the notebook

2 Formatting

This section covers basic formatting guidelines for quarto documents.

To render a document, run `quarto render`. By default, both PDF and HTML documents are generated:

```
quarto render template-computo-R.qmd # will render both to html and PDF
```



Note

To check the syntax of the formatting below, you can use the `</>` source button at the top left of this document.

2.1 Basic markdown formatting

Bold text or *italic*

- This is a list
- With more elements
- It isn't numbered.

But we can also do a numbered list

1. This is my first item
2. This is my second item
3. This is my third item

2.2 Mathematics

2.2.1 Mathematical formulae

[LaTeX](#) code is natively supported³, which makes it possible to use mathematical formulae:

$$f(x_1, \dots, x_n; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2\right)$$

It is also possible to cross-reference an equation, see Equation 1:

$$\begin{aligned} D_{x_N} &= \frac{1}{2} \begin{bmatrix} x_L^\top & x_N^\top \end{bmatrix} \begin{bmatrix} L_L & B \\ B^\top & L_N \end{bmatrix} \begin{bmatrix} x_L \\ x_N \end{bmatrix} \\ &= \frac{1}{2} (x_L^\top L_L x_L + 2x_N^\top B^\top x_L + x_N^\top L_N x_N), \end{aligned} \tag{1}$$

2.2.2 Theorems and other amsthm-like environments

Quarto includes a nice support for theorems, with predefined prefix labels for theorems, lemmas, proposition, etc. see [this page](#). Here is a simple example:

Theorem 2.1 (Strong law of large numbers). *The sample average converges almost surely to the expected value:*

$$\bar{X}_n \xrightarrow{a.s.} \mu \quad \text{when } n \rightarrow \infty.$$

³We use [lualatex](#) for this purpose.

79 See Theorem 2.1.

80 2.3 R Code

81 Quarto uses either Jupyter or knitr to render code chunks. This can be triggered in the yaml header.
82 In this tutorial, we use knitr (R and packages knitr, markdown must be installed on your computer).

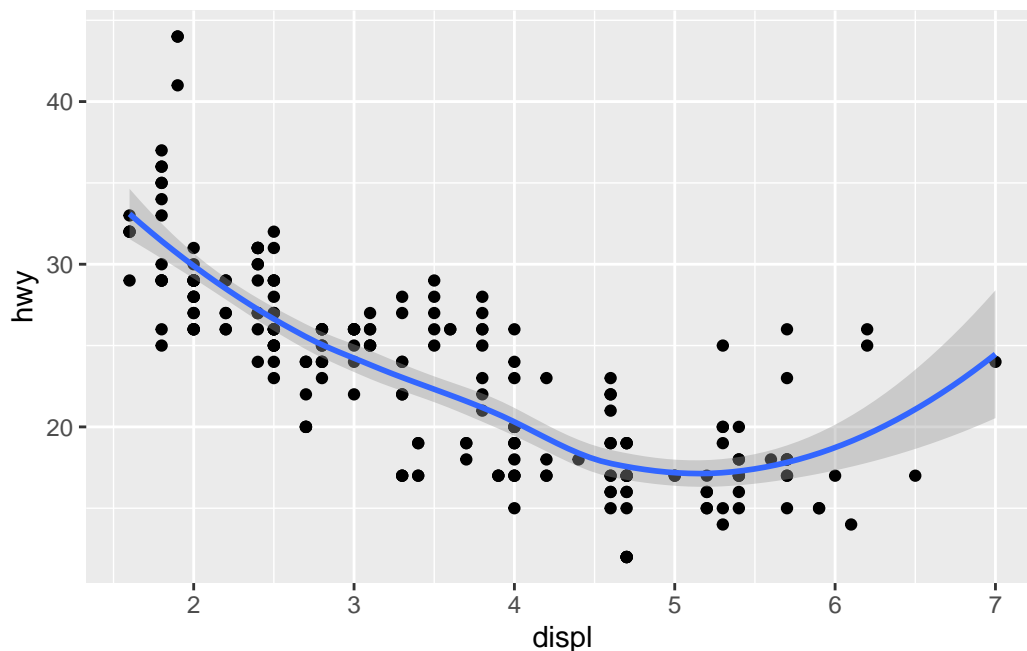
```
---  
title: "My Document"  
author "Jane Doe"  
---
```

83 R code (R Core Team, 2020) chunks may be embedded as follows:

```
x <- rnorm(10)
```

84 2.4 Figures

85 Plots can be generated as follows:



86 It is also possible to create figures from static images:

87 **Note:** Until Quarto version 1.3+ is released, including a remote image (from a web URL) in a document
88 (like the image above) will work in the rendered HTML document but will generate an error when
89 building the PDF document (c.f. [related bug report](#)).
90

91 2.5 Tables

92 Tables (with label: @tbl-mylabel renders Table 1) can be generated with markdown as follows

Table 1: my table caption

Tables	Are	Cool
col 1 is	left-aligned	\$1600

Tables	Are	Cool
col 2 is	centered	\$12
col 3 is	right-aligned	\$1

93 Table can also be generated by some code, for instance with knitr here:

Table 2: Table caption.

speed	dist
Min. : 4.0	Min. : 2.00
1st Qu.:12.0	1st Qu.: 26.00
Median :15.0	Median : 36.00
Mean :15.4	Mean : 42.98
3rd Qu.:19.0	3rd Qu.: 56.00
Max. :25.0	Max. :120.00

94 2.6 Handling references

95 2.6.1 Bibliographic references

96 References are displayed as footnotes using [BibTeX](#), e.g. `[@computo]` will be displayed as ([Computo](#)
97 [Team, 2020](#)), where `computo` is the bibtex key for this specific entry. The bibliographic informa-
98 tion is automatically retrieved from the `.bib` file specified in the header of this document (here:
99 `references.bib`).

100 2.6.2 Other cross-references

101 As already (partially) seen, Quarto includes a mechanism similar to the bibliographic references for
102 sections, equations, theorems, figures, lists, etc. Have a look at [this page](#).

103 2.7 Advanced formatting

104 Advanced formatting features are possible and documented (including interactive plots, pseudo-code,
105 (Tikz) diagrams, Lua filters, mixing R + Python in the same document), but are beyond the scope of
106 this simple introduction. We point several entries in this direction.

More information

- [The Quarto web site](#) for comprehensive documentation, including:
 - [Tutorial](#)
 - [User guide](#)
 - [Options reference](#)
- [The template distributed with the Computo Quarto extension](#), which uses such advanced features.
- [Our mock version of the t-SNE paper](#), a full and advanced example using Python and the Jupyter kernel.
- [The previously published papers in Computo](#) can be used as references.

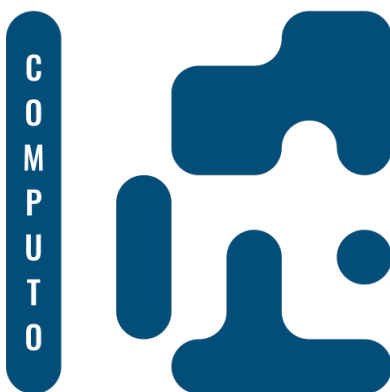


Figure 1: Computo logo (label)

3 Finalize your submission

3.1 Handle R dependencies with `renv`

To make your work reproducible, you need to fix the packages and environment used to run your analysis. For the R system, the `renv` package is one of the possible reliable method, supported by the community. You basically need a couple of commands to setup your environment on your local machine. First,

will initialize your repository. Then you just need to install the dependencies required to run your contribution, for instance,

Non-CRAN packages (e.g. Github packages) can be used. Once you are done, you can fix everything with the command

! Important

The only file that needs to be versioned by git is `renv.lock`. By default, the rest is ignored thanks to `.gitignore`.

More details for using `renv` can be found either

- on the [renv package webpage](#), or
- on the [quarto page dedicated to environments](#)

3.2 Continuous integration

The repository associated with this template is pre-configure to trigger an action on push that performs the following:

1. Check out repository on the ubuntu-latest machine
2. Install quarto and dependencies, including the Computo extension
3. Install R and dependencies with `renv`, using your `renv.lock` file
4. Render your `.qmd` file and Publish the results on a gh-page (both HTML and PDF)

The file `.github/workflows/build.yml` is largely inspired from [this file](#).

Once this is successful, you are ready to submit your manuscript to the [Computo submission platform](#).

Warning

The first time, you possibly need to create the branch for the action to work. This can be done by running the following command from your computer, in your git repository:

```
quarto publish gh-pages
```

Then, set the branch `gh-page` as the source of your github page, and trigger the action to check that everything works fine.

3.3 Data and large files

If your submission materials contain files larger than 50MB, **especially data files**, they won't fit on a git repository as is. For this reason, we encourage you to put your data or any materials you deem necessary on an external "open data" centered repository hub such a [Zenodo](#) or [OSF](#).

References

Computo Team. Computo: reproducible computational/algorithmic contributions in statistics and machine learning. 2020.

Guillermo Durand, Gilles Blanchard, Pierre Neuvial, and Etienne Roquain. Post hoc false positive control for structured hypotheses. *Scand. J. Stat.*, 47(4):1114–1148, 2020. ISSN 0303-6898. doi: 10.1111/sjos.12453. URL <https://doi.org/10.1111/sjos.12453>.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL <https://www.R-project.org/>.

Session information

R version 4.4.0 (2024-04-24)

Platform: x86_64-pc-linux-gnu

Running under: Ubuntu 22.04.4 LTS

Matrix products: default

BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3

LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-p-r0.3.20.so; LAPACK version 3.10.0

locale:

[1] LC_CTYPE=C.UTF-8	LC_NUMERIC=C	LC_TIME=C.UTF-8
[4] LC_COLLATE=C.UTF-8	LC_MONETARY=C.UTF-8	LC_MESSAGES=C.UTF-8
[7] LC_PAPER=C.UTF-8	LC_NAME=C	LC_ADDRESS=C
[10] LC_TELEPHONE=C	LC_MEASUREMENT=C.UTF-8	LC_IDENTIFICATION=C

time zone: UTC

tzcode source: system (glibc)

attached base packages:

[1] stats graphics grDevices datasets utils methods base

other attached packages:

[1] ggplot2_3.5.1

167

168 loaded via a namespace (and not attached):

```
169 [1] vctrs_0.6.5      nlme_3.1-163      cli_3.6.2         knitr_1.46
170 [5] rlang_1.1.3      xfun_0.43         renv_1.0.7        jsonlite_1.8.8
171 [9] labeling_0.4.3   glue_1.7.0        colorspace_2.1-0  htmltools_0.5.8.1
172 [13] scales_1.3.0     fansi_1.0.6       rmarkdown_2.26    grid_4.4.0
173 [17] evaluate_0.23    munsell_0.5.1     tibble_3.2.1      fastmap_1.1.1
174 [21] yaml_2.3.8       lifecycle_1.0.4   compiler_4.4.0    pkgconfig_2.0.3
175 [25] mgcv_1.9-1       farver_2.1.1      lattice_0.22-5    digest_0.6.35
176 [29] R6_2.5.1         utf8_1.2.4        splines_4.4.0     pillar_1.9.0
177 [33] magrittr_2.0.3   Matrix_1.6-5      withr_3.0.0       tools_4.4.0
178 [37] gtable_0.3.5
```