




# Computo Journal Format

To be used as template for contribution to Computo

ISSN 2824-7795

The Computo Team <sup>1</sup> Statistique, Société Française de Statistique  
a friend Another Affiliation

Date published: 2023-01-02 Last modified: 2023-11-24

## Abstract

This is the abstract - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur posuere vestibulum facilisis. Aenean pretium orci augue, quis lobortis libero accumsan eu. Nam mollis lorem sit amet pellentesque ullamcorper. Curabitur lobortis libero eget malesuada vestibulum. Nam nec nibh massa. Pellentesque porttitor cursus tellus. Mauris urna erat, rhoncus sed faucibus sit amet, venenatis eu ipsum.

*Keywords:* template, quarto, R, Python, reproducibility

## 1 Contents

2	<b>1 About this document</b>	<b>2</b>
3	<b>2 Formatting</b>	<b>2</b>
4	2.1 Basic markdown formatting . . . . .	2
5	2.2 Mathematics . . . . .	2
6	2.2.1 Mathematical formulae . . . . .	2
7	2.2.2 Theorems and other amsthem-like environments . . . . .	3
8	2.3 Code . . . . .	3
9	2.3.1 R . . . . .	3
10	2.3.2 Python . . . . .	5
11	2.4 Figures . . . . .	5
12	2.5 Tables . . . . .	5
13	2.5.1 Markdown syntax . . . . .	5
14	2.5.2 List-table filter . . . . .	5
15	2.5.3 Table generated from code . . . . .	6
16	2.6 Algorithms . . . . .	6
17	2.7 Diagrams . . . . .	8
18	2.8 Handling references . . . . .	9
19	2.8.1 Bibliographic references . . . . .	9
20	2.8.2 Other cross-references . . . . .	9
21	2.9 To go further . . . . .	10
22	Bibliography . . . . .	10

---

<sup>1</sup>Corresponding author: [computo@sfs.asso.fr](mailto:computo@sfs.asso.fr)

## 23 1 About this document

24 This document provides a template based on the [quarto system](#) for contributions to **Computo**  
25 (Computo Team 2021). We show how Python (Perez, Granger, and Hunter 2011) or R (R Core Team  
26 2020) code can be included.

## 27 2 Formatting

28 This section covers basic formatting guidelines. [Quarto](#) is a versatile formatting system for authoring  
29 HTML based on markdown, integrating  $\LaTeX$  and various code block interpreted either via Jupyter  
30 or Knitr (and thus deal with Python, R and many other languages). It relies on the [Pandoc Markdown](#)  
31 markup language.

### **i** Note

We will only give some formatting elements. Authors can refer to the [Quarto web page](#) for a complete view of the formatting possibilities.

32

### **i** Note

Quarto itself is a work-in-progress and a lot of bugs are constantly fixed or features added. As such, we recommend:

- searching any encountered issue with renders in [the upstream quarto dev repo](#)
- using a [version of quarto > 1.2](#)

33

34 To render/compile a document, run `quarto render`. A document will be generated that includes  
35 both content as well as the output of any embedded code chunks within the document:

```
quarto render content.qmd # will render to html
```

### 36 2.1 Basic markdown formatting

37 **Bold text** or *italic*

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

41 But we can also do a numbered list

- 42 1. This is my first item
- 43 2. This is my second item
- 44 3. This is my third item

### 45 2.2 Mathematics

#### 46 2.2.1 Mathematical formulae

47  $\LaTeX$  code is natively supported<sup>2</sup>, which makes it possible to use mathematical formulae:

48 will render

---

<sup>2</sup>We use [katex](#) for this purpose.

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

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

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

### 50 2.2.2 Theorems and other amsthem-like environments

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

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

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

55 See Theorem 2.1.

### 56 2.3 Code

57 Quarto uses either Jupyter or knitr to render code chunks. This can be triggered in the yaml header,  
58 e.g., for Jupyter (should be installed on your computer) use

```
---
title: "My Document"
author: "Jane Doe"
jupyter: python3
---
```

59 For knitr (R + knitr must be installed on your computer)

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

60 You can use Jupyter for Python code and more. And R + Knitr for if you want to mix R with Python  
61 (via the package reticulate Ushey, Allaire, and Tang (2020)).

#### 62 2.3.1 R

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

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

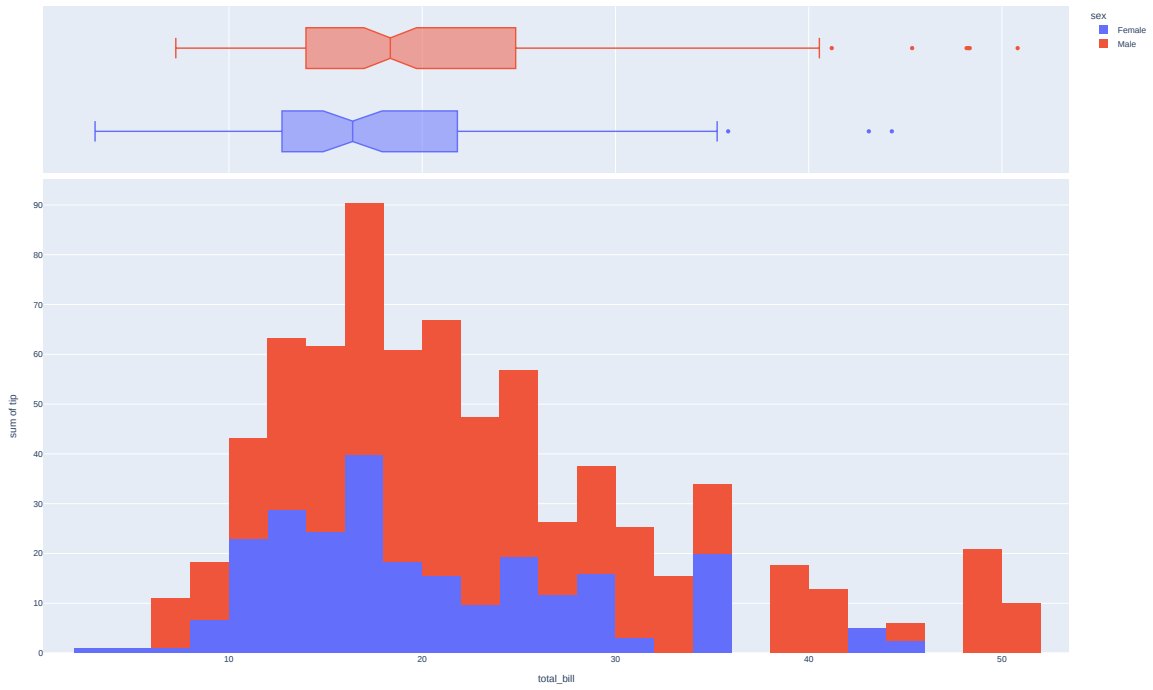


Figure 1: A simple python plotly example

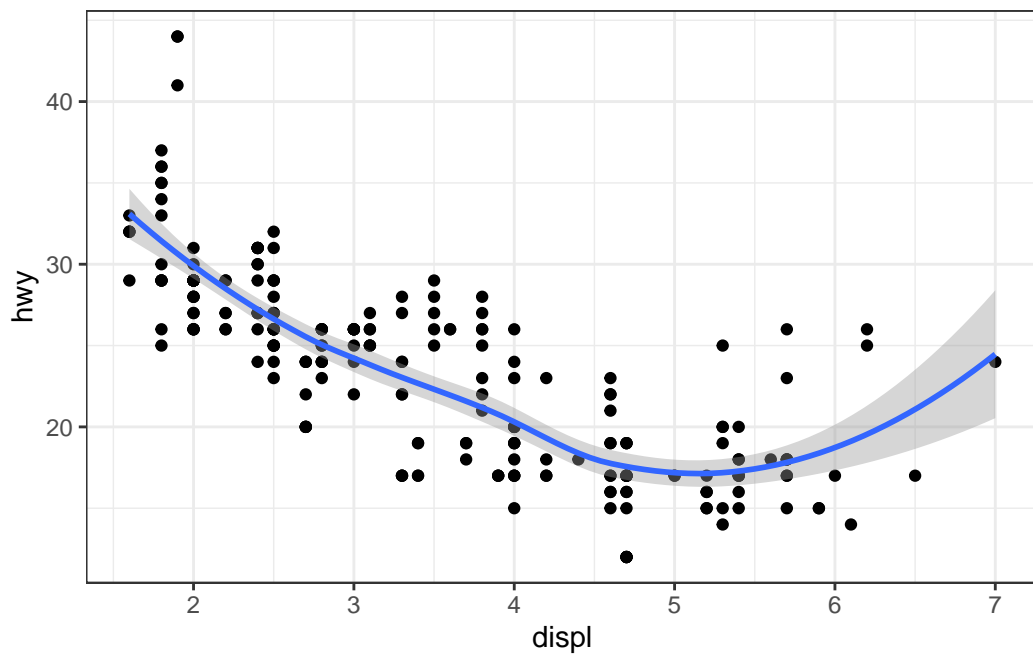


Figure 2: A simple ggplot example

64 **2.3.2 Python**

65 **2.4 Figures**

66 Plots can be generated as follows and referenced. See plot Figure 2:

67 Interactive plots may also be produced in the HTML output of the document<sup>3</sup>:

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

69 **2.5 Tables**

70 **2.5.1 Markdown syntax**

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

```
| Tables | Are | Cool |
|-----|:-----:|-----:|
| col 1 is | left-aligned | $1600 |
| col 2 is | centered | $12 |
| col 3 is | right-aligned | $1 |
: my table caption {#tbl-mylabel}
```

Table 1: my table caption

Tables	Are	Cool
col 1 is	left-aligned	\$1600
col 2 is	centered	\$12
col 3 is	right-aligned	\$1

72 **2.5.2 List-table filter**

73 We also integrate the [list tables](#) filter from Pandoc, so that you may alternatively use this format ,  
74 easier to write and maintain:

```
:::list-table
* - row 1, column 1
- row 1, column 2
- row 1, column 3

* - row 2, column 1
-
- row 2, column 3

* - row 3, column 1
- row 3, column 2
:::
```

row 1, column 1	row 1, column 2	row 1, column 3
row 2, column 1		row 2, column 3
row 3, column 1	row 3, column 2	

<sup>3</sup>The pdf output is just a screenshot of the interactive plot from the html output

---

---

row 1, column 1	row 1, column 2	row 1, column 3
-----------------	-----------------	-----------------

---

---

### 75 2.5.3 Table generated from code

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

Table 3: 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

## 77 2.6 Algorithms

78 A solution to typeset pseudocode just like you would do with  $\LaTeX$ , yet with HTML output is to rely  
79 on the JavaScript [pseudocode.js](#). Your pseudocode is written inside a [Code Block](#) with the pseudocode  
80 class. Do not forget the class tag, that will trigger the rendering process of your pseudo-code. The  
81 result is as follows<sup>4</sup>:

```
```\pseudocode
#| label: alg-quicksort
#| html-indent-size: "1.2em"
#| html-comment-delimiter: "/"
#| html-line-number: true
#| html-line-number-punc: ":"
#| html-no-end: false
#| pdf-placement: "htb!"
#| pdf-line-number: true

\begin{algorithm}
\caption{Quicksort}
\begin{algorithmic}
\Procedure{Quicksort}{ $A$ ,  $p$ ,  $r$ }
  \If{ $p < r$ }
    \State  $q = \text{\Call{Partition}{ $A$ ,  $p$ ,  $r$ }}$ 
    \State  $\text{\Call{Quicksort}{ $A$ ,  $p$ ,  $q - 1$ }}$ 
    \State  $\text{\Call{Quicksort}{ $A$ ,  $q + 1$ ,  $r$ }}$ 
  \EndIf
\EndProcedure
\Procedure{Partition}{ $A$ ,  $p$ ,  $r$ }
  \State  $x = A[r]$ 
  \State  $i = p - 1$ 
  \For{ $j = p, \dots, r - 1$ }
```

---

<sup>4</sup>For proper pdf rendering, use [Camel cased](#) names for all `algorithmic` keywords, not upper case ones, like the examples in `pseudocode.js`'s documentation, which are not compatible with  $\LaTeX$ .

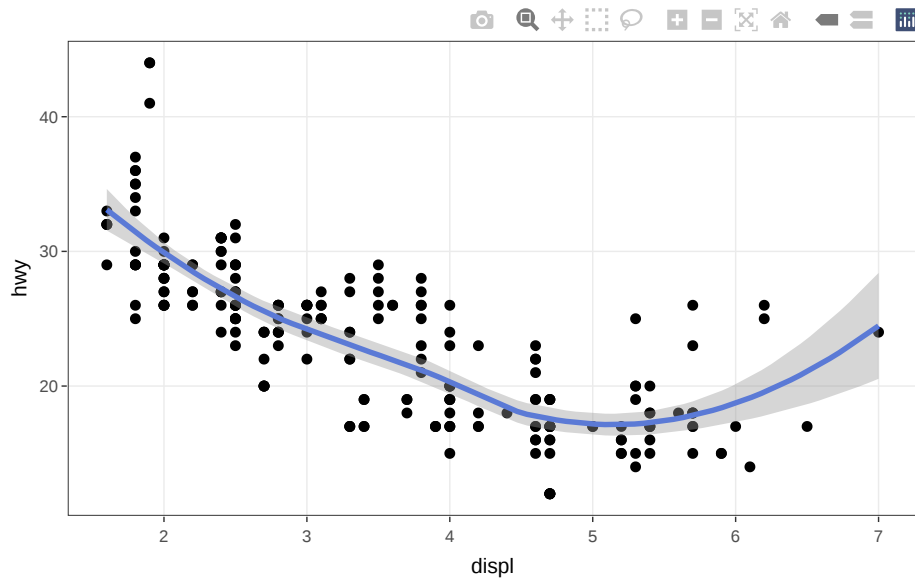


Figure 3: A simple ggplotly interactive example



Figure 4: SFdS logo (c.a. 2021)

```

\If{ $A[j] < x$ }
  \State  $i = i + 1$ 
  \State exchange
   $A[i]$  with  $A[j]$ 
\EndIf
\State exchange  $A[i]$  with  $A[r]$ 
\EndFor
\EndProcedure
\end{algorithmic}
\end{algorithm}
...

```

---

### Algorithm 1 Quicksort

---

```

1: procedure QUICKSORT( $A, p, r$ )
2:   if  $p < r$  then
3:      $q = \text{PARTITION}(A, p, r)$ 
4:     QUICKSORT( $A, p, q - 1$ )
5:     QUICKSORT( $A, q + 1, r$ )
6:   end if
7: end procedure
8: procedure PARTITION( $A, p, r$ )
9:    $x = A[r]$ 
10:   $i = p - 1$ 
11:  for  $j = p, \dots, r - 1$  do
12:    if  $A[j] < x$  then
13:       $i = i + 1$ 
14:      exchange  $A[i]$  with  $A[j]$ 
15:    end if
16:    exchange  $A[i]$  with  $A[r]$ 
17:  end for
18: end procedure

```

---

82 Algorithm 1 is extracted from Chapter 7, Introduction to Algorithms (3rd edition).

## 83 2.7 Diagrams

84 In addition of [quarto supported diagrams](#), we also support [tikz](#) diagrams. The following example<sup>5</sup> is  
85 rendered as follows.

```

:::{#fig-tikz}

``` {.tikz opt-additional-packages="\usetikzlibrary{arrows}}
\begin{tikzpicture}[node distance=2cm, auto, thick, scale=2, every node/.style={transform shape}]
\node (P) {$P$};
\node (B) [right of=P] {$B$};
\node (A) [below of=P] {$A$};
\node (C) [below of=B] {$C$};
\node (P1) [node distance=1.4cm, left of=P, above of=P] {$\hat{P}$};
\draw[->] (P) to node {$f$} (B);

```

---

<sup>5</sup>This is the new syntax for cross-references since quarto 1.4, see [Crossreferenceable elements](#)



```

\draw[->] (P) to node [swap] {$g$} (A);
\draw[->] (A) to node [swap] {$f$} (C);
\draw[->] (B) to node {$g$} (C);
\draw[->, bend right] (P1) to node [swap] {$\hat{g}$} (A);
\draw[->, bend left] (P1) to node {$\hat{f}$} (B);
\draw[->, dashed] (P1) to node {$k$} (P);
\end{tikzpicture}
...

```

A simple example of a commutative diagram with `\texttt{tikz}`.

...

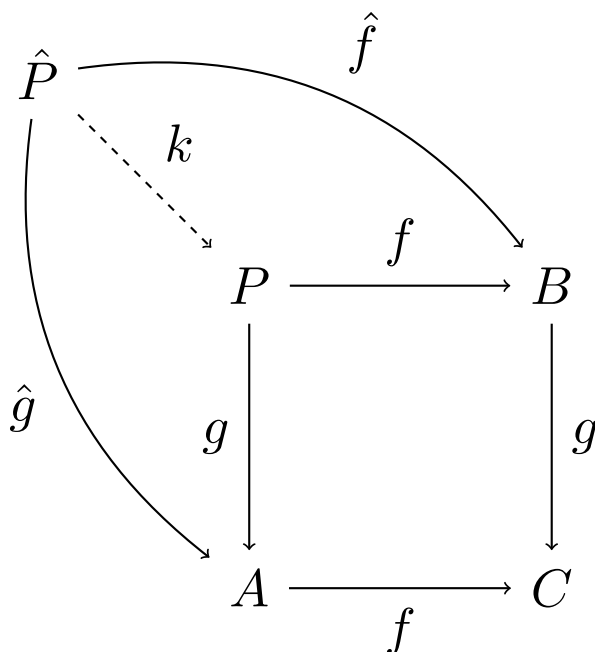


Figure 5: A simple example of a commutative diagram with `tikz`.

86 You may refer to it as Figure 5.

## 87 2.8 Handling references

### 88 2.8.1 Bibliographic references

89 References are displayed as footnotes using `BibTeX`, e.g. `[@computo]` will be displayed as (Com-  
90 puto Team 2021), where `computo` is the `bibtex` key for this specific entry. The bibliographic in-  
91 formation is automatically retrieved from the `.bib` file specified in the header of this document  
92 (`here:references.bib`).

### 93 2.8.2 Other cross-references

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

## 96 2.9 To go further

### 97 **i** One last note

To go into more involved details, you can also simply check the source code of this document (button at the top), or have a look at the source of our [t-sne remake example](#).

## 99 **Bibliography**

- 100 Computo Team. 2021. “Computo: Reproducible Computational/Algorithmic Contributions in Statis-  
101 tics and Machine Learning.” *Computo*.
- 102 Perez, Fernando, Brian E Granger, and John D Hunter. 2011. “Python: An Ecosystem for Scientific  
103 Computing.” *Computing in Science  
104 & Engineering* 13 (2): 13–21.
- 105 R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R  
106 Foundation for Statistical Computing. <https://www.R-project.org/>.
- 107 Ushey, Kevin, JJ Allaire, and Yuan Tang. 2020. *Reticulate: Interface to Python*. [https://github.com/  
108 rstudio/reticulate](https://github.com/rstudio/reticulate).