NeurIPS Journal Format

John Joe
Electrical and Computer Engineering
Texas A&M University
College Station, TX 77843, USA
computo@sfds.asso.fr

a friend
Another Affiliation

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.

1 About this document

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

2 Formatting

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

> **Block title 1**
>
> We will only give some formatting elements. Authors can refer to the Quarto web page for a complete view of the formatting possibilities.

2.1 Block title 2

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

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

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

2.2 Basic markdown formatting

Bold text or *italic*

- This is a list

Preprint. Under review.

- 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

2.3 Mathematics

2.3.1 Mathematical formulae

LaTeX code is natively supported[1], which makes it possible to use mathematical formulae:

will render

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

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

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

2.3.2 Theorems and other amsthem-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:*

$$\overline{X}_n \xrightarrow{a.s.} \mu \qquad \textit{when } n \to \infty.$$

See Theorem 2.1.

2.4 Code

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

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

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

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

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

[1] We use katex for this purpose.

2.4.1 R

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

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

2.4.2 Python

2.5 Figures

Plots can be generated as follows and referenced. See plot **?@fig-gg**:

Interactive plots may also be produced in the HTML output of the document[2]:

It is also possible to create figures from static images:

Figure 1: SFdS logo (c.a. 2021)

2.6 Tables

2.6.1 Markdown syntax

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

2.6.2 List-table filter

We also integrate the list tables filter from Pandoc, so that you may alternatively use this format , 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
:::
```

[2]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
row 2, column 1		row 2, column 3
row 3, column 1	row 3, column 2	

2.6.3 Table generated from code

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

```
knitr::kable(summary(cars), caption = "Table caption.")
```

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

2.7 Algorithms

A solution to typeset pseudocode just like you would do with LaTeX, yet with HTML output is to rely on the JavaScript pseudocode.js. Your pseudocode is written inside a Code Block with the `pseudocode` class. Do not forget the class tag, that will trigger the rendering process of your pseudo-code. The result is as follows[3]:

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

[3]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.
````

```
    \EndIf
    \State exchange $A[i]$ with $A[r]$
  \EndFor
\EndProcedure
\end{algorithmic}
\end{algorithm}
```

Algorithm 1 Quicksort

procedure QUICKSORT(A, p, r)
  if $p < r$ **then**
    $q =$ PARTITION(A, p, r)
    QUICKSORT($A, p, q - 1$)
    QUICKSORT($A, q + 1, r$)
  end if
end procedure
procedure PARTITION(A, p, r)
  $x = A[r]$
  $i = p - 1$
  for $j = p, \ldots, r - 1$ **do**
    if $A[j] < x$ **then**
      $i = i + 1$
      exchange $A[i]$ with $A[j]$
    end if
    exchange $A[i]$ with $A[r]$
  end for
end procedure

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

2.8 Diagrams

In addition of quarto supported diagrams, we also support tikz diagrams. The following example[4] is rendered as follows.

A simple example of a commutative diagram with `tikz`.

You may refer to it as Figure 2.

2.9 Handling references

2.9.1 Bibliographic references

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

2.9.2 Other cross-references

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

2.10 To go further

> **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.

[4] This is the new syntax for cross-references since quarto 1.4, see Crossreferenceable elements

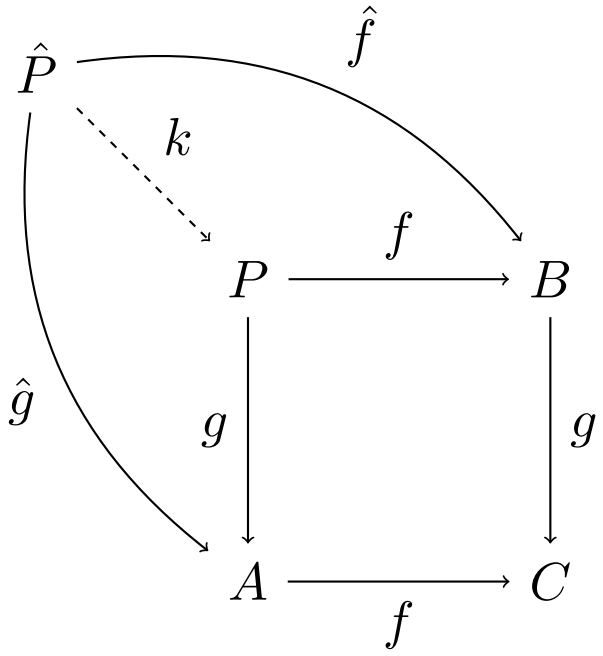

Figure 2

Bibliography

Computo Team. 2021. "Computo: Reproducible Computational/Algorithmic Contributions in Statistics and Machine Learning." *Computo*.

Perez, Fernando, Brian E Granger, and John D Hunter. 2011. "Python: An Ecosystem for Scientific Computing." *Computing in Science & Engineering* 13 (2): 13–21.

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

Ushey, Kevin, JJ Allaire, and Yuan Tang. 2020. *Reticulate: Interface to Python*. https://github.com/rstudio/reticulate.