

---

# STATS 531 project report format

---

**Blinded**

## Abstract

This is the abstract. It should briefly describe the context of the report and highlight the findings of highest interest to the target readers (other students in the class, the GSI, and the instructor).

## 1 About this document

This document provides a template based on the [quarto system](#) based on <https://github.com/stevengogogo/neurips-quarto-extension/> and adapted for STATS 531. 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 L<sup>A</sup>T<sub>E</sub>X 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.

### **i** 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

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 pdf
```

#### 2.2 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

## 2.3 Mathematics

### 2.3.1 Mathematical formulae

`LATEX` code is natively supported<sup>1</sup>, which makes it possible to use mathematical formulae: will render

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

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

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

### 2.3.2 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:*

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

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)).

### 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 Figure 1:

BROKEN. NEED A SAMPLE PLOT HERE

It is also possible to create figures from static images:

<sup>1</sup>We use `katex` for this purpose.



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-table` 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
:::
```

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

speed	dist
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 L<sup>A</sup>T<sub>E</sub>X, 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<sup>2</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 = $ \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 $A[j]$
        \EndIf
        \State exchange $A[i]$ with $A[r]$
    \EndFor
\EndProcedure
\end{algorithmic}
\end{algorithm}
```

```

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

<sup>2</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 L<sup>A</sup>T<sub>E</sub>X.

```

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

```

Algorithm ?? 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<sup>3</sup> is rendered as follows.

For learning TiKZ, I recommend this website: [Tikz examples](#).

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 mechanism similar to the bibliographic references for sections, equations, theorems, figures, lists, etc. Have a look at [this page](#).

## 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.

---

<sup>3</sup>This is the new syntax for cross-references since quarto 1.4, see [Crossreferenceable elements](#)

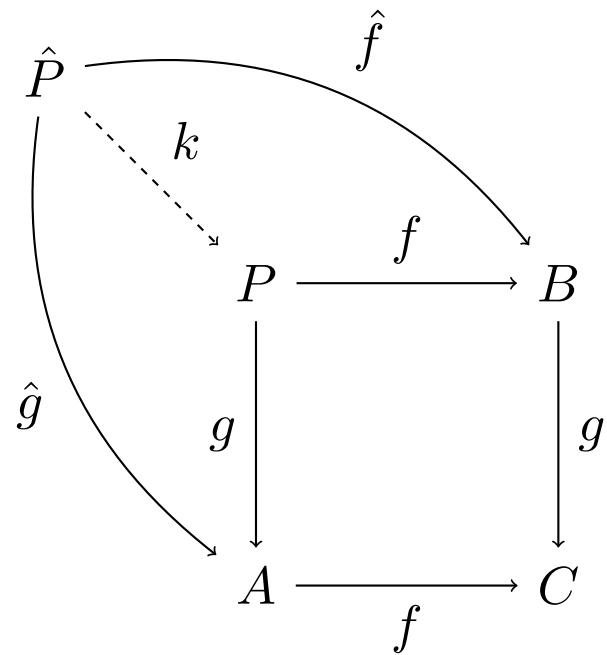


Figure 2

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/studio/reticulate>.