

Formation Quarto

Parcours de formation R perfectionnement SSP

Michaël Delorme

2025-02-07

Table des matières

Pré-requis techniques	2
Introduction	2
Objectifs	3
Quarto	3
Avantages	4
Inconvénients	4
Sur CERISE	4
Syntaxe et structure	5
markdown	5
chunks	5
YAML	6
Rendu	6
Éléments	6
Texte	6
Titres	7
Listes	7
Liens	8
Images	9
Renvois	9
Notes de pied de page	10
Équations	10
Citations/bibliographie	11
Libellés	11
Code	12
Tableaux	14

Graphiques	16
Cartes	17
Fonctionnalités avancées	17
Rapports paramétrés	17
Inclusions	18
Cache	19
Crosstalk	19
Dashboards	19
Marque de l'État	19
Typographie Marianne	19
Couleurs	20
Références	20


Pré-requis techniques

- Quarto (≥ 1.5)
- RStudio (≥ 2022.07) et R (≥ 4)

```
library(dplyr) # manip. données
library(ggplot2) # graphiques
library(sf) # carto
library(leaflet) # carto
library(gt)

# un theme ggplot utilisant la police Marianne
theme_masa <- function(...) {
  theme_grey(...) %+replace%
  theme(
    text = element_text(family = "Marianne")
  )
}
```

Introduction

 Mise en garde

DIAPORAMA EN COURS DE CONSTRUCTION

Objectifs

- Intérêt des publications reproductibles
- Découvrir la versatilité de Quarto
- Réaliser un document, de simple à complexe

versatile : pour le langage en entrée et les formats en sortie (diapo suivante)

Quarto

Un système de publication

- multi-format
 - HTML (rapports, livres, présentations)
 - PDF
 - Appli web (statique/Shiny) : blog, site, tableau de bord
 - Powerpoint
 - MS Word
- multi-langage
 - R
 - Python
 - [Observable](#)
 - [Julia](#)

Le format PDF nécessite {tinytex}



Figure 1: Principe de fonctionnement (source RStudio)

Mélange code et texte

Successeur de Rmarkdown

Commandes complémentaires LaTeX ou Typst pour le PDF

Pandoc est un convertisseur “universel” de formats de texte

Avantages

- intégration avec R
- reproductibilité
 - limiter les copier/coller entre outils
 - prise en compte des changements de données ou de méthodes
- interactivité possible dans le document (HTML)
- versionnement avec git
- afficher du code dans le doc
- éviter les copier/coller : synchronisation analyse/résultats
- faciliter la reproductibilité avec {renv}

Inconvénients

- syntaxe (markdown)
- génération

Mais disponibilité d’un éditeur visuel [What You See Is What You Mean](#) dans RStudio.

Sur CERISE

Un package {quarto} et un binaire quarto.

```
```{r}
install.packages("quarto")
```
```

```
quarto check install
quarto check knitr
```

Le format PDF nécessite LaTeX.

```
quarto install tinytex
```

Syntaxe et structure

```
---
title: "Litany against fear"
author: "Bene Gesserit"
---

I must not fear. *Fear is the mind-killer*. Fear is the little-death that
brings total obliteration.

```{python}
import numpy as np
np.exp(np.pi * complex(0, 1)) + 1
```

**I will face my fear.** I will permit it to pass over me and through me.

```{r}
somme <- sum(1:10)
```

And when it has gone past I will turn the `{r} somme` inner eye to see its
path. Where the fear has gone there will be nothing.

> Only I will remain.
```

markdown

Un langage balisé simplifié

[doc](#)

chunks

Le code exécutable est intégré dans un *chunk* (bloc) :

```
```{r}
#| label: mon-chunk
#| warning: false

1+1
```
```

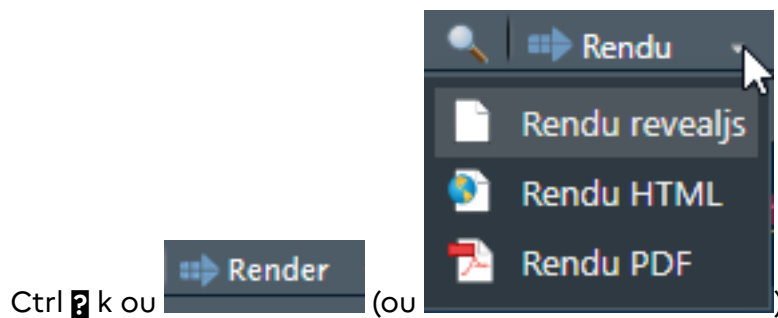
doc

YAML

- en-tête de configuration
- bloc entre ---
- *a minima* : title:
- variable selon les formats

Rendu

```
quarto render mon_document.qmd
```



i À vous de jouer

[Exercice 1.](#) : découvrir l'interface

Éléments

Texte

- 2 retours à la ligne pour faire un paragraphe

- 2 espaces et 1 retour à la ligne pour un saut de ligne
- * ou _ pour mettre en *italique* ou en **gras**
- [un lien] (https://www.example.org/) → [un lien](https://www.example.org/)
- une portion de texte peut être stylée en vert avec [vert]{style="color:darkgreen"}
- mais du HTML est aussi possible rouge pour mettre en rouge

doc

Titres

Préfixer le titre avec #, ## ou ### pour avoir les titres de niveau 1, 2 ou 3, etc..

Titre de niveau 4

Titre de niveau 4

Listes

```
- Atréides
  - Leto
  - Jessica
  - Paul
- Harkonnen
  - Vladimir
  - Rabban
  - Feyd-Rautha
```

ou

```
- Atréides
  * Leto
  * Jessica
  * Paul
- Harkonnen
  * Vladimir
  * Rabban
  * Feyd-Rautha
```

- Atréides
 - Leto
 - Jessica
 - Paul
 - Harkonnen
 - Vladimir
 - Rabban
 - Feyd-Rautha
-

```

- Atréides
- Leto
- Jessica
  2 espace + nouvelle ligne + aligner pour
  ajouter du texte dans les items
- Paul
- Harkonnen
- Vladimir
- Rabban
- Feyd-Rautha

```

- Atréides
 - Leto
 - Jessica
 - 2 espace + nouvelle ligne + aligner pour ajouter du texte dans les items**
 - Paul
- Harkonnen
 - Vladimir
 - Rabban
 - Feyd-Rautha

Liens

```
[Texte du lien] (https://fr.wikipedia.org/wiki/Arrakis)
```

Texte du lien

Images

! [Fremen] (images/images/dune_book.jpg)



Figure 2: Fremen

```
! [Fremen] (images/images/dune_book.jpg "Shai-hulud"){#fig-dune fig-alt="A sand worm on Arrakis"}
```

Utiliser le préfixe `fig-` pour l'identifiant afin de pouvoir bénéficier des renvois automatiques et de la numérotation automatique

Renvois

(*cross-references*)

Un lien automatique est créé si on indique @fig-dune ; il nous écrit le lien (type et numéro) renvoie à Figure 2.

Les noms des types affichés peuvent être modifiés dans l'en-tête :

```
-----  
crossref:  
  fig-prefix: figure  
  fig-title: Figure  
  tbl-prefix: tableau  
  tbl-title: Tableau  
  title-delim: " &ndash; "  
-----
```

Notes de pied de page

Les notes se construisent en ajoutant un ^[1] dans le texte, avec le 1 pouvant être une numérotation (pas obligatoirement dans l'ordre) ou tout autre mot clé. La note est ensuite introduite avec le même symbole dans un nouveau paragraphe :

[¹]: texte de la note

C'est normalement facile¹ à faire si on se rappelle de la syntaxe.

Équations

Syntaxe LaTeX entre des \$ (inline) ou \$\$ (bloc).

```
 $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}, \text{ où } \mu = \frac{1}{N} \sum_{i=1}^N x_i$ 
```

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}, \text{ où } \mu = \frac{1}{N} \sum_{i=1}^N x_i$$

[doc](#)

¹[page d'aide](#) sur les notes de bas de page.

Citations/bibliographie

Avec [Zotero](#) et l'extension [Better BibTeX](#) pour auto-exporter en BibTeX la collection quand elle est modifiée.

Préciser le fichier utilisé en en-tête, puis utilisation des identifiants du fichier BibTeX.

```
---
title: "Mon document"
bibliography: formation_quarto.bib
---

Comme le dit @rdevelopmentcoreteamLanguageEnvironmentStatistical2010 :

> Lorem ipsum dolor sit amet[@rstudioGuide2024], consectetur adipiscing elit
```

Comme le dit R Development Core Team (2010) :

 Lorem ipsum dolor sit amet (RStudio, 2024a), consectetur adipiscing elit

La bibliographie est directement générée :

- en fin de document ou
- après un <div> identifié avec #refs ex. :

```
## Références
::: {#refs}
:::
```

[doc](#)

Libellés

(Callout)

Les "libellés" permettent de mettre en exergue des informations complémentaires.

```
::: {.callout-note}
Il existe cinq types de libellés :
`note`, `tip`, `warning`, `caution` et `important`.
:::
```

Note

Il existe cinq types de libellés : `note`, `tip`, `warning`, `caution` et `important`.

```
::: callout-warning
Suite aux nombreux manquements à l'obligation de contrôle hebdomadaire (...)
:::
```

Avertissement

Suite aux nombreux manquements à l'obligation de contrôle hebdomadaire de l'étanchéité des colonnettes à crémaillère (...)

```
::: callout-warning
#### Attention

Le caisson de calibrage de la bride du commutateur à fuseaux (...)
:::
```

Attention

Le caisson de calibrage de la bride du commutateur à fuseaux rotatifs est HORS-SERVICE

[doc](#)

À vous de jouer

[Exercice 2](#). produire un document simple

Code

```
iris |>
  glm(Sepal.Length ~ Sepal.Width, data = _)
```

Call: `glm(formula = Sepal.Length ~ Sepal.Width, data = iris)`

Coefficients:

| | |
|-------------|-------------|
| (Intercept) | Sepal.Width |
| 6.5262 | -0.2234 |

Degrees of Freedom: 149 Total (i.e. Null); 148 Residual

Null Deviance: 102.2

Residual Deviance: 100.8 AIC: 372

Options de bloc de code

- exécution :
- `#| eval`: exécuter le code
- `#| echo`: afficher le code
- nommage :
- `#| label`: identifiant du chunk
- `#| fig-cap`: légende du bloc

Outils

Si on a `#| echo: true` l'utilisateur voit le code

On peut aussi ajouter des outils pour l'utilisateur pour qu'il affiche ou cache les chunks avec, dans l'en-tête YAML :

```
format:
  html:
    code-fold: true
    code-tools: true
```

Sorties de code

- texte, avec :
 - `#| results: markup` OU
 - `cat()` et `#| results: asis`
- tableaux
- images
- éléments interactifs

i À vous de jouer

Exercice 3. Code

Tableaux

Les tableaux sont balisés ainsi en Markdown :

```
année	titre
1965	Dune
1969	Le Messie de Dune
1976	Les Enfants de Dune
1981	L'Empereur-Dieu de Dune
1984	Les Hérétiques de Dune
1985	La Maison des mères

: Cycle de Dune {#tbl-cycle}
```

| année | titre |
|-------|-------------------------|
| 1965 | Dune |
| 1969 | Le Messie de Dune |
| 1976 | Les Enfants de Dune |
| 1981 | L'Empereur-Dieu de Dune |
| 1984 | Les Hérétiques de Dune |
| 1985 | La Maison des mères |

Table 1: Cycle de Dune

```
iris |>
  summarise(.by = Species,
            moy_sepales = sum(Sepal.Length))
```

| | Species | moy_sepales |
|---|------------|-------------|
| 1 | setosa | 250.3 |
| 2 | versicolor | 296.8 |
| 3 | virginica | 329.4 |

Table 2: Iris (tibble)

[doc](#)

Mais la plupart du temps on affichera des tableaux programmatiquement.

Avec {knitr} ou le paramètre en en-tête :

```
format:
  html:
    df-print: kable
```

```
iris |>
  summarise(.by = Species,
            moy_sepales = sum(Sepal.Length)) |>
  knitr::kable()
```

| Species | moy_sepales |
|------------|-------------|
| setosa | 250.3 |
| versicolor | 296.8 |
| virginica | 329.4 |

Table 3: Iris (kable)

[doc](#)

Contrôler l'impression de data.frame avec df-print: kable ou df-print: paged, df-print: tibble Interactivité javascript df-print: paged à voir en exercice

| Species | moy_sepales |
|------------|-------------|
| setosa | 250.3 |
| versicolor | 296.8 |
| virginica | 329.4 |

Table 4: Iris (gt)

ou avec {gt}

```
iris |>
  summarise(.by = Species,
            moy_sepales = sum(Sepal.Length)) |>
  gt()
```

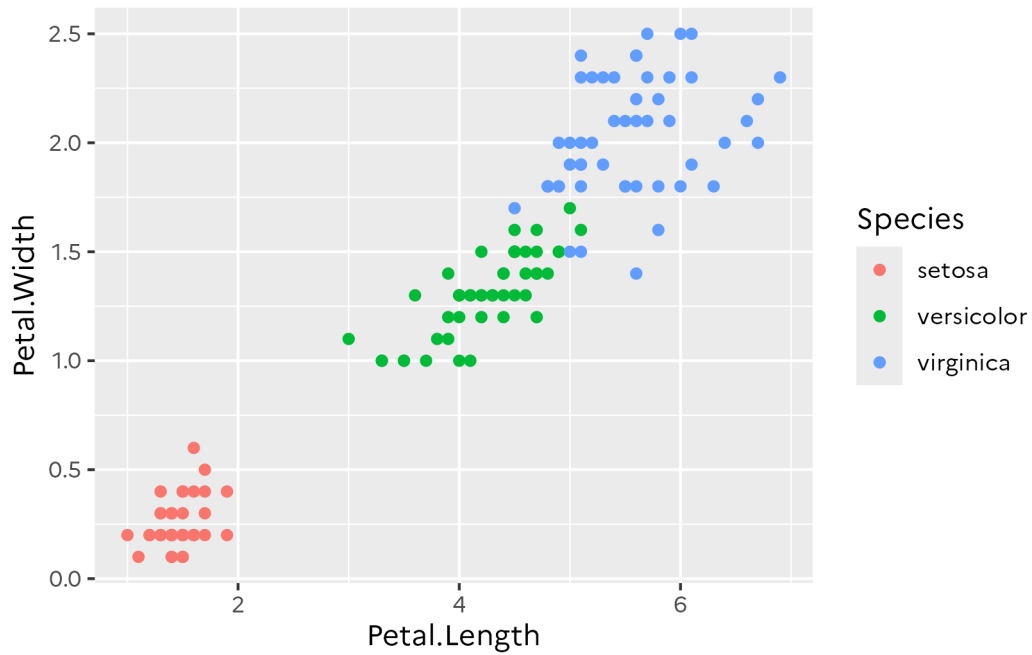
Utiliser le préfixe `tbl-` pour l'identifiant afin de pouvoir bénéficier des renvois automatiques et la numérotation automatique

i À vous de jouer

[Exercice 3. Tableaux](#)

Graphiques

```
ggplot(iris, aes(Petal.Length,
                 Petal.Width,
                 color = Species)) +
  geom_point() +
  theme_masa()
```

Listing 1 mon_document.qmd

```
---
title: Rapport
format:
  html: default
param:
  region: 84
  sortie: knitr
---

## Région `{r}` params$region`

```{r}
sau <- ra |>
 filter(siege_reg == params$region) |>
 summarise(.by = siege_dep,
 sau_tot = sum(sau_tot, na.rm = TRUE))

if (params$sortie == "knitr") kable::knitr(sau) else sau
```
```

```
```{r}
for (r in c(11, 94, 84, 04)) {
 quarto::quarto_render("mon_document.qmd",
 execute_params = list(region = r),
 output_file = paste0("rapport_", r, ".html"))
}
```
```

résultera en 4 rapports personnalisés : *rapport_11.html*, *rapport_94.html*,...

[doc](#)

Inclusions

```
{{< include entete.qmd >}}
```

Cache

Pour les calculs longs et rarement variables on peut gagner du temps avec `cache: true` (au niveau document ou chunk).

```
```{r}
#| label: long-traitement
#| cache: true
#| cache-extra : !expr file.mtime("donnees/departements.gpkg")

region <- read_sf("donnees/departements.gpkg") |>
 st_union()
```
```

Ne sera exécuté qu’une fois sauf si le code du chunk change ou si la date du fichier change ou si on passe à `cache: false`, ou via :

```
quarto render mon_document.qmd --cache-refresh
```

[doc](#)

Crosstalk

[doc](#)

Dashboards

[doc](#)

Marque de l’État

Typographie Marianne

Pour textes et graphiques (Service d’information du gouvernement, 2020)

- Installer `{ragg}`, `{systemfonts}` et `{textshaping}`
- Dans l’en-tête yaml du document Quarto :

```
---  
mainfont: Marianne  
knitr:  
  opts_chunk:  
    dev: "ragg_png"  
---
```

Permet d'avoir la police Marianne pour le texte et de la prendre en compte dans les graphiques ggplot.

Autre option (à mettre dans un chunk de configuration en début de document) :
`knitr::opts_chunk$set(dev = "ragg_png")`

Couleurs

Références

Arel-Bundock V. *Consistent Figures in Quarto*. Vincent Arel-Bundock. juillet 2024. Disponible sur : < https://arelbundock.com/posts/quarto_figures/index.html >

Canouil M. *Awesome-Quarto*. janvier 2025. Disponible sur : < <https://github.com/mcanouil/awesome-quarto> >

Choe J. *Setting up and Debugging Custom Fonts*. June Choe. juin 2021. Disponible sur : < <https://yjunechoe.github.io/posts/2021-06-24-setting-up-and-debugging-custom-fonts/> >

Heiss A. *Working with R, Cairo Graphics, Custom Fonts, and Ggplot*. Andrew Heiss. avril 2023. Disponible sur : < <https://www.andrewheiss.com/blog/2017/09/27/working-with-r-cairo-graphics-custom-fonts-and-ggplot/> > (consulté le 21 janvier 2025)

R Development Core Team. *R: A Language and Environment for Statistical Computing*. 2010. Disponible sur : < <http://www.R-project.org/> >

RStudio. *Guide*. Quarto. 2024a. Disponible sur : < <https://quarto.org/docs/guide/> >

RStudio. *Publish and Share with Quarto :: Cheatsheet*. 2024b. Disponible sur : < <https://rstudio.github.io/cheatsheets/html/quarto.html> >

Service d'information du gouvernement. *Charte graphique de l'État*. [s.l.] : République française, 2020. Disponible sur : < <https://www.gouvernement.fr/marque-Etat> >