

Réaliser un article scientifique avec Quarto

Marie Vaugoyeau

18 March 2025

Table of contents

1	C'est quoi Quarto ?	1
2	Création d'un projet Quarto	2
3	Lecture d'un fichier Quarto	2
3.1	En-tête YAML (facultative)	3
3.2	Syntaxe du texte	3
3.3	Portion de code	3
3.4	Ajout d'une équation	4
3.5	Ajout de référence bibliographiques	4
3.5.1	Grâce à Zotero	4
3.5.2	Grâce à une recherche	4
3.6	Intégrer un graphique	5
4	En savoir un peu plus sur moi	6

1 C'est quoi Quarto ?



Après une dizaine d'année de développement sur RMarkdown, Posit (anciennement RStudio) a annoncé en 2022 la naissance de Quarto.

Donc Quarto est l'héritier de RMarkdown qui combine le texte sous format Markdown et les

portions de code appelé **chunk**.

Le code peut-être en **R** comme en **Python**.

Comme **RMarkdown**, sa grande force est de rendre **reproductible**, **répétable** et **réutilisable** les lignes de codes.

Innovation

Il n'y aura pas ou peu d'innovations sur **RMarkdown** maintenant que **Quarto** existe.

2 Création d'un projet Quarto

La grosse nouveauté entre **RMarkdown** et **Quarto** est que **Quarto** est une infrastructure plus large qui remplace le classique projet **RStudio**.

Pour créer un projet **Quarto**, c'est très simple, il suffit de cliquer sur l'icône bleue projet ou **File > New Project...**

Ensuite, il faut cliquer sur **Quarto project**. S'il n'apparaît pas ce que **RStudio** n'est pas à jour. Il est nécessaire de commencer par mettre à jour **R** et **RStudio**.

Le projet créé contient trois fichiers :

- __ Le fichier **_quarto.yml** qui contient les paramètres du dossier
- __ Le fichier **nom_du_projet.qmd** qui est un document **Quarto** généré automatiquement
- __ Le fichier **nom_du_projet.Rproj**, le fichier du projet

Le document **Quarto** généré automatiquement est très simple. Il est là pour comprendre le fonctionnement de base. Il est très utile pour apprendre à manipuler **Quarto**.

3 Lecture d'une fichier Quarto

Les fichiers **Quarto** ont une composition classique : une en-tête **YAML** facultative avec un enchaînement de **texte** et de **blocs de codes**.

Deux **vues** sont disponibles :

- __ **Visual** : Qui ressemble à ce qui sera obtenu avec des clics boutons proches d'un logiciel de traitement texte tels que **Word**, **Mot**, **Writer...**
- __ **Source** : Qui contient le script en **Markdown**

3.1 En-tête YAML (facultative)

Elle permet de paramétrer les sorties du document mais elle n'est pas obligatoire

! Indentation

Si l'indentation (les espaces avant le texte) n'a que peu d'importance dans le code R, elle est très importante en YAML et doit-être regardé avec attention !

Le document généré automatiquement ne possède qu'un titre dans l'en-tête.

3.2 Syntaxe du texte

Le texte en Quarto, utilise la [syntaxe Markdown](#) :

__ Les titres sont caractérisés par des # en fonction du niveau : #Titre de niveau 1, ##Titre de niveau 2...

__ La mise en forme du texte se fait avec les étoiles ou les impostrophes (apostrophe à l'envers) :

- Une étoile * avant et après la partie à valoriser permet de mettre en *italique* (codé *italique*)
- Deux ** avant et après mettent en **gras** (codé **gras**)
- Trois *** avant et après mettent en ***gras et italique*** (codé ***gras et italique***)
- Un **double espace** permet de passer à la ligne. Il faut donc toujours mettre deux espaces à la fin de chaque ligne. Sans ces espaces, toutes les lignes sont collées les unes à la suite des autres
- Pour éditer un **format code** (sans qu'il se lance) pour présenter les packages, les fonctions ou les objets utilisés (comme fait dans ce document), il faut encadrer d'impostrophes

Des images peuvent aussi être intégrées grâce à ![*légende_de_l_image*](adresse_de_l_image.format_de_l_image)

3.3 Portion de code

Le code est enregistré dans des chunks délimités par trois impostrophes comme ci-dessous.

Les paramètres du chunk sont précédés de #|.

Cela permet de donner un nom au chunk avec #| label: nom_du_chunk ou de choisir les sorties #| echo: true : le code est affiché dans le format de sortie, #| warning: false : les

messages de type `warning` ne sont pas affichés...
Tous les paramétrages peuvent se retrouver dans [le guide Quatro](#).

```
```${r}  
#| label: nom_du_chunk
#| warning: false

1 + 1
```
```

[1] 2

Si un des impostrophes est supprimé par erreur, le code ne peut pas se lancer !
Il faut toujours vérifier que les chunks soient bien fermés (fond d'une couleur différente).

3.4 Ajout d'une équation

Dans la ligne avec le symbole dollar Soit on peut choisir de les intégrer dans le texte : $\$Y = aX + b\$$ ou dans un bloc à part grâce au double dollar $\$ \$ Z = Y + zb \$ \$$

Soit on peut choisir de les intégrer dans le texte : $Y = aX + b$

$$Z = Y + zb$$

3.5 Ajout de référence bibliographiques

3.5.1 Grâce à Zotero

Association via Zotero Web grâce à une clé API : Global Options > R Markdown > Citations

Plus d'info sur le [tuto de Claire Della Vedova](#)

3.5.2 Grâce à une recherche

Dans l'onglet Visual, cliquer sur Insert puis Citation.
Il est possible de chercher directement via le [DOI](#) ou dans [Crossref](#), [DataCite](#) ou [PubMed](#).

3.6 Intégrer un graphique

Il y a deux possibilités pour générer un graphique :

- Le coder dans le document Quarto
- Le coder dans un autre document Quarto puis l'intégrer dans celui de l'article

Dans les deux cas, il faut faire le graphique dans un chunk en le nommant avec `label: fig-nom_du_graph`

```
```{r}
#| label: fig-iris

plot(iris)
```
```

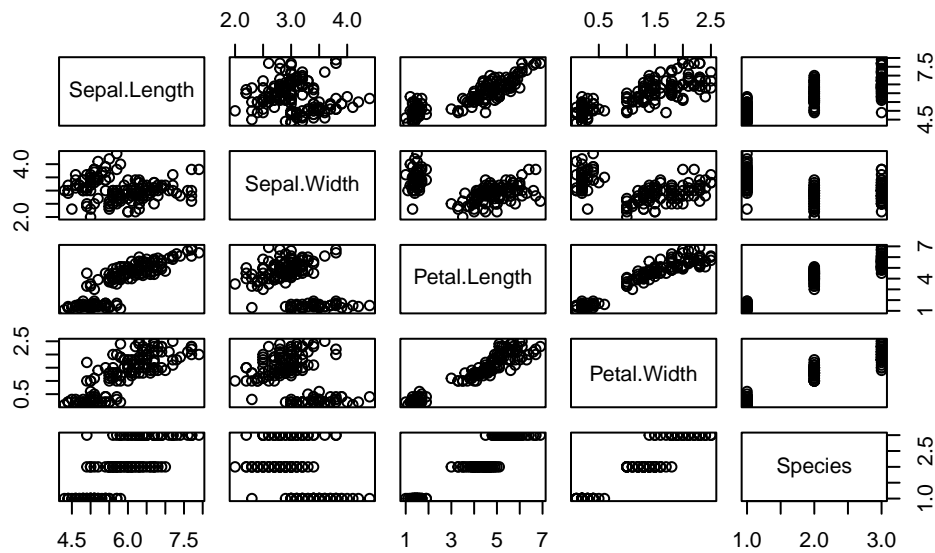


Figure 1

Si le graphique est généré dans un autre document, il peut être intégré grâce à `{{< embed nom_du_doc.qmd#fig-nom_du_graph >}}` codé `{{< embed 01_graph_explo.qmd#fig-graph_iris >}}` dans la [vidéo sur Twitch](#).

Pour ensuite faire un lien hypertexte vers un graphique, il suffit dans le texte de rajouter @fig-nom_du_graph codé @fig-iris dans la [vidéo](#).

4 En savoir un peu plus sur moi

Bonjour,

Je suis Marie Vaugoyeau et je suis disponible pour des **missions en freelance d'accompagnement à la formation** à R et à l'analyse de données et/ou en **programmation** (reprise de scripts, bonnes pratiques de codage, développement de package). Ayant un **bagage recherche en écologie**, j'ai accompagné plusieurs chercheuses en biologie dans leurs analyses de données mais je suis ouverte à d'autres domaines.

Vous pouvez retrouver mes offres [ici](#).

En plus de mes missions de consulting je diffuse mes savoirs en R et analyse de données sur plusieurs plateformes :

- J'ai écrit [un livre aux éditions ENI](#)
- Tous les mois je fais [un live sur Twitch](#) pour parler d'un package de R, d'une analyse
- Je rédige une **newsletter** de manière irrégulière pour parler de mes **inspirations** et transmettre **des trucs et astuces sur R**. Pour s'y inscrire, [c'est par là](#). J'ai aussi [un blog](#) sur lequel vous pourrez retrouver une version de cet article.

Pour en savoir encore un peu plus sur moi, il y a [LinkedIn](#) et pour retrouver [tous ces liens et plus encore, c'est ici](#)

N'hésitez pas à me contacter sur marie.vaugoyeau@gmail.com !

Bonne journée

Marie

