
LNG-1100 : Méthodes expérimentales et analyse de données

Communication scientifique et académique : Quarto et Markdown

Guilherme D. Garcia

fr.gdgarcia.ca

8



UNIVERSITÉ
LAVAL

Plan de la séance

Dans RStudio aujourd'hui

1. Quarto et Markdown
2. Gestion bibliographique
3. Code dans le texte

remplacer MS Word, MS PowerPoint

remplacer EndNote, Mendeley

remplacer Excel, SPSS

👉 Pratique dans votre équipe

Quarto

- Un système de production de documents dans RStudio

👉 **Les avantages** : facile, automatisé, professionnel




Quarto

La composition de documents

- Pour les documents scientifiques/académiques, il faut combiner l'analyse, la gestion de références, et le texte de façon **optimale**. On a besoin d'être précis.
- Il existe différentes options, dont voici deux :

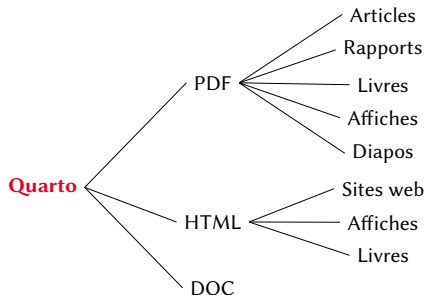
	Analyse	Gestion	Texte
1	SPSS	EndNote	MS Word
2	RStudio	BIB \TeX	L \TeX

1. Cette option est chère et ses résultats ne sont pas optimaux
2. Cette option est 100% gratuite et elle offre les meilleurs résultats

👉 Quarto nous permet d'utiliser l'option 2 à travers du [Markdown](#) 

Quarto

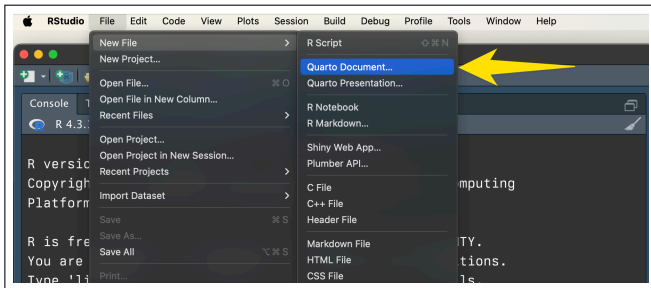
La composition de documents



1. Quarto est capable de générer plusieurs types de documents à partir d'un seul langage très facile à apprendre
2. Il est également capable de gérer nos références bibliographiques
3. Finalement, le système interprète notre code R nativement

👉 **Tout cela se fait dans RStudio**

1. File > New File > Quarto Document...
2. Sélectionnez « PDF », et désélectionnez « Use visual markdown editor »
3. Cliquez sur « Create ». RStudio signalera qu'il faut installer une extension. Installez-la
4. Cliquez sur l'icône d'engrenage et sélectionnez « Preview in Viewer Pane »



PARTIE I : MARKDOWN

Markdown

Un langage très simple

- Consultez [cette page](#) pour en savoir plus. Voici un résumé :

Élément	Syntaxe Markdown
Titres	<code># H1 titre</code> <code>## H2 sous-titre</code> <code>### H3 sous-sous-titre</code>
Texte	<code>gras</code> <code><i>italique</i></code>
Liens	<code>[ULaval](https://ulaval.ca)</code>
Code	<code>`fonction()`</code> <code>`r print("Oui!")`</code>
Figure	<code>![Une figure](image.jpg)</code>

Markdown

L'automatisation

- En général, votre document sera mis en forme automatiquement
- Vous pouvez donc vous concentrer sur le **contenu** du document

☞ Dans un éditeur de texte, on a besoin de **deux** éléments :

1. Formatage de texte (par exemple, MS Word, Google Docs)
 2. Gestion de références bibliographiques (par exemple, EndNote, Zotero, Mendeley)
- Voyons comment Quarto simplifie ces éléments

Markdown

Philosophie minimaliste

- Un texte académique est esthétiquement **simple**
 - Donc, les mots gras et italiques sont généralement suffisants
 - La couleur ne se trouve pas dans le texte (typiquement), mais dans les **figures**
- 👉 **Résultat** : la plupart des options de MS Word sont **inutiles** dans un doc académique

PARTIE II : GESTION BIBLIOGRAPHIQUE

Gestion bibliographique

- Idée : un fichier `.bib` pour notre bibliothèque de références
- Chaque référence suit une syntaxe particulière :

```
1 @article{keyword,  
2   author={nom, prenom and nom, prenom},  
3   title={titre de l'article},  
4   journal={publication},  
5   year={2025}
```

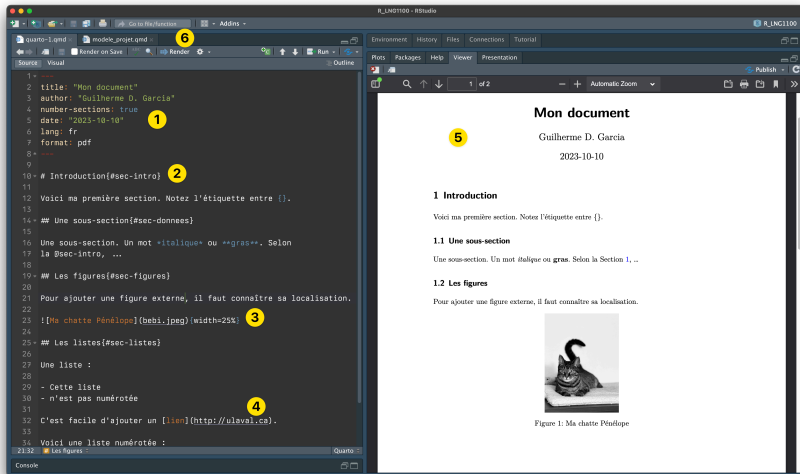
Gestion bibliographique

- Exemple réel : un livre

```
1 @book{wickham2009ggplot2,  
2   title={ggplot2: elegant graphics for data analysis},  
3   author={Wickham, Hadley and Sievert, Carson},  
4   year={2009},  
5   publisher={Springer},  
6   address={New York}  
7 }
```

- Mais comment trouver ces informations...? L'option la plus simple : [Google Scholar](#)

PARTIE III : CODE + TEXTE



Dans la dernière capture d'écran :

1. Le préambule (en-tête) : les spécifications du document (auteur, date, format, titre, etc.)
2. Un exemple d'étiquetage pour créer des références croisées (p. ex., ligne 14)
3. Une figure externe
4. Un lien
5. Le PDF généré à partir de notre document
6. Pour générer le PDF, on clique sur « Render »

- Un bloc de code indique en environnement qui doit être interprété par Quarto
- Cet environnement est démarqué par trois accents graves :

```
1 ```{r}
2 #| ...
3 #| ...
4 #| ...
5
6 ```
```

☞ Les lignes 2–4 indiquent des arguments du bloc :

<code>message: false</code>	les messages ne sont pas imprimés
<code>echo: false</code>	le code sera caché
<code>eval: false</code>	le code ne sera pas interprété
<code>fig- ...</code>	des paramètres pour les figures

- On peut combiner notre texte et notre code dans une même ligne :

```
1 ```{r}
2 #| message: false
3 library(tidyverse)
4 d = read_csv("donnees/villes2.csv")
5 ```
6 # Introduction
7
8 La moyenne générale est `r mean(d$note) |> round(2)`.
```

- Ici, on calcule la moyenne des notes et on arrondit le chiffre (à deux décimales)

De retour au chapitre 7 : 1h

📄 Créez une analyse en format PDF pour les exercices du chapitre 7

• Votre analyse doit contenir :

1. Une introduction* où vous décrivez les données et posez la question de recherche (10 min)
2. Une section d'exploration où vous créez des tableaux/figures (10 min)
3. Une section d'analyse où vous exécutez un modèle (10 min)
4. Une section de discussion et conclusion où vous interprétez les résultats (10 min)

• Votre texte sera très court dans cet exercice. *Vous devez citer l'extension [tidyverse](#) ainsi que les œuvres suivantes lors de la description des données :

- Balota, D., Cortese, M., Sargent-Marshall, S., Spieler, D. and Yap, M. (2004) Visual word recognition for single-syllable words, *Journal of Experimental Psychology: General*, 133, 283–316.
- Baayen, R.H., Feldman, L. and Schreuder, R. (2006) Morphological influences on the recognition of monosyllabic monomorphemic words, *Journal of Memory and Language*, 53, 496–512.

Pourquoi utiliser Markdown?

- La possibilité de combiner texte + code + références dans un même document
- Différents formats possibles : pdf, html, doc
- Vitesse, automatisation, esthétique professionnelle et peu de travail