

Rmd: document HTML

Table of Contents

- [Rmd: document HTML](#)
 - [Table des matières](#)
 - [COT flottant](#)
 - [Numérotation des sections](#)
 - [Sections à onglets](#)
 - [Apparence et style](#)
 - [CSS personnalisé](#)
 - [Options de figure](#)
 - [Impression de trames de données](#)
 - [Impression paginée](#)
 - [Code de pliage](#)
 - [équations de MathJax](#)
 - [Dépendances du document](#)
 - [Personnalisation avancée](#)
 - [Garder Markdown](#)
 - [Inclusion](#)
 - [Modèles personnalisés](#)
 - [Extensions de Markdown](#)
 - [Arguments Pandoc](#)
 - [Options partagées](#)
 - [fragments HTML](#)

Markdown a été conçu à l'origine pour la sortie HTML. Pour créer un document HTML à partir de R Markdown, spécifier le format de sortie `html_document` dans les métadonnées YAML de votre document:

```
---
title: Habits
author: John Doe
date: March 22, 2005
output: html_document
---
```

Table des matières

Ajouter une table des matières à l'aide de l'option `toc` et spécifier la profondeur des en-têtes auxquels elle s'applique à l'aide de l'option `toc_depth`. Par exemple:

```
---
```

```
title:  "Habits"
output: html_document:
  toc:  true
  toc_depth:  2
---
```

Si la profondeur de la table des matières n'est pas explicitement spécifiée, la valeur par défaut est 3 (ce qui signifie que tous les en-têtes de niveau 1, 2 et 3 seront inclus dans la table des matières).

COT flottant

Spécifier l'option `toc_float` pour faire flotter la table des matières à gauche du contenu du document principal. La table des matières flottante sera toujours visible même lorsque le document est déroulé. Par exemple:

```
---
title:  "Habits"
output: html_document:
  toc:  true
  toc_float:  true
---
```

Eventuellement spécifier une liste d'options pour le paramètre **toc_float** qui contrôlent son comportement. Ces options comprennent:

- **collapsed (par défaut, TRUE)** contrôle si la table des matières apparaît uniquement avec les en-têtes de niveau supérieur (par exemple, H2). Si elle est initialement réduite, la table des matières est automatiquement développée en ligne si nécessaire.
- **smooth_scroll (la valeur par défaut est TRUE)** détermine si les défilements de page sont animés lorsque les éléments de la table des `smooth_scroll` via des clics de souris.

Par exemple:

```
---
title:  "Habits"
output: html_document:
  toc:  true
  toc_float:
    collapsed:  false
    smooth_scroll:  false
---
```

Numérotation des sections

Ajouter une numérotation de section aux en-têtes à l'aide de l'option `number_sections` :

```
---
title:  "Habits"
```

```
output: html_document:
  toc: true
  number_sections: true
---
```



utiliser des en-têtes # (H1) dans le document, car les en-têtes ## (H2) incluent un point décimal, ainsi sans les en-têtes H1, les en-têtes H2 seront: numéroté avec 0.1 , 0.2 , et ainsi de suite.

Sections à onglets

On peut organiser le contenu à l'aide d'onglets en appliquant **.tabset** aux en-têtes d'un document. Ainsi, tous les sous-en-têtes de l'en-tête avec l'attribut **.tabset** apparaîtront dans des onglets plutôt que sous forme de sections autonomes. Par exemple:

```
## Quarterly Results {.tabset}
### By Product (tab content)
### By Region (tab content)
```

On peut également spécifier deux attributs supplémentaires pour contrôler l'apparence et le comportement des onglets:

- **.tabset-fade** provoque l'effacement des onglets lors du basculement entre les onglets.
- **.tabset-pills** fait en sorte que l'aspect visuel des onglets soit d'aspect "pillule" plutôt que des onglets traditionnels. Par exemple:

```
## Quarterly Results {.tabset .tabset-fade .tabset-pills}
```

Apparence et style

Plusieurs options permettent de contrôler l'apparence des documents HTML:

- **theme** spécifie le thème Bootstrap à utiliser pour la page (les thèmes sont tirés de la bibliothèque de thèmes de Bootstrap). Les thèmes valides sont les suivants: défaut, céruleen, journal, plat, sombre, lisible, spacelab, uni, cosmo, lumière, papier, grès, simplex et yeti. Passez null pour aucun thème (dans ce cas, vous pouvez utiliser le paramètre css pour ajouter vos propres styles).
- **highlight** spécifie le style de surbrillance de la syntaxe. Les styles pris en charge comprennent les styles par défaut , tango , pygments , kate , monochrome , espresso , zenburn , haddock et textmate . Passez null pour empêcher la coloration syntaxique.
- **smart (activé par défaut)** indique s'il faut produire une sortie typographiquement correcte, en convertissant les guillemets droits en guillemets bouclés, — d-tirets, - en tirets, et ... en ellipses.

Par exemple:

```

--- title:  "Habits"
output: html_document:
  theme:  united
  highlight:  tango
---
```

CSS personnalisé

On peut ajouter des CSS à un document HTML à l'aide de l'option `css` :

```

--- title:  "Habits"
output: html_document:
css:  styles.css
---
```

Pour fournir un styles à tout du document à partir d'un CSS, définir le theme (et éventuellement le highlight) sur `null` :

```

---
title:  "Habits"
output: html_document:
  theme:  null
  highlight:  null
css:  styles.css
---
```

On peut également cibler des sections spécifiques de documents avec des CSS personnalisées en ajoutant des identifiants ou des classes aux en-têtes de section du document. Par exemple, l'en-tête de section suivant:

```
## Next Steps {#nextsteps .emphasized}
```

Permettrait d'appliquer CSS à tout son contenu à l'aide de l'un des sélecteurs CSS suivants:

```
#nextsteps { color : blue ; } .emphasized { font-size : 1.2em ; }
```

Options de figure

Un certain nombre d'options affectent la sortie des figures dans les documents HTML:

- **fig_width** et **fig_height** peuvent être utilisés pour contrôler la largeur et la hauteur de la figure par défaut (7×5 est utilisé par défaut).
- **fig_retina** spécifie la mise à l'échelle à effectuer pour les affichages de rétine (la valeur par défaut est 2, ce qui fonctionne actuellement pour tous les affichages de rétine largement utilisés). Définir sur `null` pour empêcher la mise à l'échelle de la rétine.
- **fig_caption** contrôle si les figures sont rendues avec des légendes.
- **dev** contrôle le périphérique graphique utilisé pour rendre les figures (par défaut, png).

Par exemple:

```
---
title: "Habits"
output: html_document:
  fig_width: 7
  fig_height: 6
  fig_caption: true
---
```

Impression de trames de données

On peut améliorer l'affichage par défaut des trames de données via l'option **df_print**. Les possibles de l'option *dfprint pour le format html*document sont:

Valeurs	Description
default	Appelle la méthode générique print.data.frame
kable	Utilise la fonction knitr::kable
tibble	Utilise la fonction tibble::print.tbl_df
paged	Utilise rmarkdown::paged_table pour créer une table paginable

Impression paginée

Lorsque l'option **df_print** est définie sur **paged**, les tables sont imprimées sous forme de tables HTML prenant en charge la pagination sur des lignes et des colonnes. Par exemple :

```
---
title: "Motor Trend Car Road Tests"
output: html_document:
  df_print: paged
---
```{r}
mtcars
```
```

Options pour les tables HTML paginées.

| Option | Description |
|----------------|---|
| max.print | Le nombre de lignes à imprimer. |
| rows.print | Le nombre de lignes à afficher. |
| cols.print | Le nombre de colonnes à afficher. |
| cols.min.print | Le nombre minimum de colonnes à afficher. |
| pages.print | Le nombre de pages à afficher dans la navigation de page. |
| paged. | Lorsqu'il est défini sur FALSE désactive les tables paginées. |
| rownames.print | Lorsqu'il est défini sur FALSE désactive les noms de ligne. |

Ces options sont spécifiées dans chaque bloc comme ci-dessous:

```
```\n{r cols.print=3, rows.print=3}\nmtcars\n```\n
```

## Code de pliage

Lorsque l'option **knitr chunk echo = TRUE** est spécifiée (comportement par défaut), le code source R contenu dans les fragments est inclus dans le document rendu. Dans certains cas, il peut être approprié d'exclure entièrement le code ( `echo = FALSE` ), mais dans d'autres cas, peut-être que le code devrait être disponible mais ne soit pas visible par défaut.

L'option **code\_folding: hide** permet d'inclure du code R mais de le masquer par défaut. Les utilisateurs peuvent ensuite choisir d'afficher les fragments de code R masqués individuellement ou dans l'ensemble du document. Par exemple:

```
---\ntitle: "Habits"\noutput: html_document:\n code_folding: hide\n---
```

On peut spécifier **code\_folding: show** pour toujours afficher tout le code R par défaut, puis autoriser les utilisateurs à masquer le code s'ils le souhaitent.

## équations de MathJax

Par défaut, les scripts MathJax sont inclus dans les documents HTML pour le rendu des équations LaTeX et MathML. On peut utiliser l'option **mathjax** pour contrôler la manière dont MathJax est inclus:

- Spécifier **default** pour utiliser une URL HTTPS à partir d'un hôte CDN (actuellement fourni par RStudio).
- Spécifier **local** pour utiliser une version locale de MathJax (qui est copiée dans le répertoire de sortie). Lorsqu'on utilise "local" on doit également définir l'option **self\_contained** sur **false**.
- Spécifier une autre URL pour charger MathJax à partir d'un autre emplacement.
- Spécifier **null** pour exclure entièrement MathJax.

Par exemple, pour utiliser une copie locale de MathJax:

```
---\ntitle: "Habits" output:\nhtml_document:\n mathjax: local\n self_contained: false\n---
```

Pour utiliser une copie auto-hébergée de MathJax:

```

title: "Habits"
output: html_document:
 mathjax: "http://example.com/MathJax.js"

```

Pour exclure entièrement MathJax:

```

title: "Habits"
output: html_document:
 mathjax: null

```

## Dépendances du document

Par défaut, R Markdown génère des fichiers HTML autonomes sans dépendance externe, à l'aide de **data: URI** pour incorporer le contenu de scripts, de feuilles de style, d'images et de vidéos liés. Cela signifie qu'on peut partager ou publier le fichier exactement comme on partage des documents Office ou des PDF. Si on préfère conserver les dépendances dans des fichiers externes, spécifier **self\_contained: false**. Par exemple:

```

title: "Habits"
output: html_document:
self_contained: false

```



même pour les documents autonomes, MathJax est toujours chargé de l'extérieur (cela est nécessaire en raison de sa taille importante). Pour servir MathJax localement, il faut spécifier **mathjax: local** et **self\_contained: false**.

L'une des raisons courantes de conserver les dépendances externes est de servir les documents R Markdown à partir d'un site Web (les dépendances externes peuvent être mises en cache séparément par les navigateurs, ce qui accélère le chargement des pages). Pour également consolider les fichiers de bibliothèque dépendants (par exemple, Bootstrap et MathJax, etc.) dans un seul répertoire partagé par plusieurs documents, utiliser l'option **lib\_dir**. Par exemple:

```

title: "Habits"
output: html_document:
self_contained: false
lib_dir: libs

```

# Personnalisation avancée

## Garder Markdown

Lorsque knitr traite un fichier d'entrée R Markdown, il crée un fichier Markdown (\*.md) qui est ensuite transformé en HTML par Pandoc. Pour conserver une copie du fichier Markdown après le rendu, utiliser l'option **keep\_md**:

```

title: "Habits"
output: html_document:
 keep_md: true

```

## Inclusion

On peut personnaliser davantage la sortie en ajoutant du contenu HTML supplémentaire ou en remplaçant entièrement le modèle Pandoc principal. Pour inclure le contenu dans l'en-tête du document ou avant / après le corps du document, utiliser l'option **includes** comme suit:

```

title: "Habits"
output: html_document:
 includes:
 in_header: header.html
 before_body: doc_prefix.html
 after_body: doc_suffix.html

```

## Modèles personnalisés

On peut également remplacer le modèle Pandoc sous-jacent à l'aide de l'option `template` :

```

title: "Habits"
output: html_document:
 template: quarterly_report.html

```

## Extensions de Markdown

Par défaut, R Markdown est défini comme toutes les extensions Pandoc Markdown avec les modifications suivantes pour la compatibilité en amont avec l'ancien package de démarques (J. Allaire, Horner, et al. 2018) :

```
+autolink_bare_uris
+ascii_identifier
+tex_math_single_backslash
```

On peut activer ou désactiver les extensions de Markdown à l'aide de l'option **md\_extensions**. Par exemple:

```

title: "Habits"
output: html_document:
md_extensions:
 -autolink_bare_uris
 +hard_line_breaks

```

Ce qui précède désactiverait l'extension **autolink\_bare\_uris** et activerait l'extension **hard\_line\_breaks**.

## Arguments Pandoc

Pour utiliser certaines fonctionnalités Pandoc n'ayant pas leur équivalent dans les options YAML, utiliser en transmettant des arguments **pandoc\_args** personnalisés. Par exemple:

```
--- title: "Habits"
output: html_document:
 pandoc_args:
 ["--title-prefix" , "Foo" , "--id-prefix" , "Bar"]

```

## Options partagées

Pour spécifier un ensemble d'options par défaut à partager par plusieurs documents dans un répertoire, inclure un fichier nommé **\_output.yml** dans le répertoire. Aucun délimiteur YAML ( — ) ni champ de output englobant ne sont utilisés dans ce fichier. Par exemple:

```
html_document:
 self_contained: false
 theme: united
 highlight: textmate
```

Il ne devrait pas être écrit comme ceci :

```

output: html_document:
 self_contained: false
 theme: united
 highlight: textmate
```

---



Tous les documents situés dans le même répertoire que **\_output.yml** hériteront de ses options. Les options définies explicitement dans les documents remplacent celles spécifiées dans le fichier d'options partagées.

## fragments HTML

Pour créer un fragment HTML plutôt qu'un document HTML complet, on peut utiliser le format **html\_fragment**. Par exemple:

```

output: html_fragment

```



les fragments HTML ne sont pas des documents HTML complets. Ils ne contiennent pas le contenu d'en-tête standard que les documents HTML (ils ne contiennent que le contenu dans les balises **<body>** des documents HTML normaux). Ils sont destinés à être inclus dans d'autres pages Web ou systèmes de gestion de contenu (tels que des blogs). En tant que tels, ils ne supportent pas les fonctionnalités telles que les thèmes ou la mise en surbrillance de code (on s'attend à ce que l'environnement dans lequel ils sont finalement publiés gère ces choses).

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=outils:r-html>Last update: **2025/02/19 10:59**