




Pagedown - utilisations possibles et création de thèmes

Rencontres R 2021


Thomas Vroylandt
13 juillet 2021

Qui suis-je ?

- Statisticien (à [l'Unédic](#) actuellement)
- Avec une [activité de conseil en plus](#)
- Développeur du package {pagedreport}: [documentation](#)
- Cette présentation est issue des travaux que j'ai pu mener



SCOUTS
GUIDES
DE FRANCE



OMNI

Document Title
Document Subtitle


Submitted to:
Client Name
Client Organization
Report Month Report Year

Author(s): Authors
Project Team: Project Team

For More Information:
Project Code: Project Code
projects@omni.org

Acknowledgements:
The OMNI Institute wants to thank Names of People to Thank for their contributions to the creation of this report.


Suggested Citation:
The OMNI Institute (Report Year). Document Title. Submitted to Client Organization, Client City, Client State



Date

Title of the report
Subtitle of the report

Author Name



PAGEDOWN POUR LES SCOUTS
ET GUIDES DE FRANCE

{pagedown} ???

Un problème ...

➡ On souhaite souvent générer des PDF directement depuis R car :

- assure la reproductibilité
- permet de ne pas changer d'outil
- autorise des approches paramétrisées
- c'est un format portable
- très utilisé pour les rapports ou les documents d'entreprise

Un problème ...

☛ Sauf que :

- les sorties LaTeX sont peu attrayantes

Untitled

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

This is a quote

This is a link: <http://rmarkdown.rstudio.com>

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0   Min.   : 2.00
## 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
```

- elles demandent des compétences pointues de personnalisation
- le multi-support (web + PDF est difficile à gérer)

... une solution

Il est possible d'imprimer des pages Web spécialement mises en forme grâce au projet [paged.js](#).

💡 Transcrit pour être utilisé en R dans [pagedown](#) par Romain Lesur et Yihui Xie

➡ Les avantages :

- Utilisation des outils du design Web (HTML / CSS / JS)
- Qualité des rendus
- Collaboration possible avec des spécialistes de ces technologies

! Nécessite de comprendre au moins les bases du CSS et d'HTML

Fondations

✓ Il existe un projet de spécifications officielles pour créer des documents paginés en CSS. Ce sont celles du W3C.

! Sauf qu'elle ne sont pas encore implementées correctement dans les navigateurs.

🔑 Le projet paged.js est une programme qui :

- utilise les règles présentes dans un fichier CSS
- pour construire un document paginé
- c'est une sorte de "ciseaux" + "collage" à l'échelle d'une page Web

📖 La documentation est extrêmement riche

Comment ça fonctionne ?

{pagedown} implémente principalement :

- Des formats RMarkdown, dont `html_paged` qui va appeler `paged.js` et des fonctions pour "découper" (*Fragmentation CSS*) et reconstruire le contenu en pages.
- Des fichiers CSS standards liés aux formats ➡ **nos modifications passent principalement par là**
- Et une fonction pour permettre l'impression en PDF : `chrome_print` (utilise l'impression de Chrome ou de Edge).

■ [Présentation à la RStudio Conf 2019.](#)

{pagedreport}

-  [Github](#)
-  [Documentation](#)
-  [Article de blog](#)

? Projet en collaboration avec David Keyes de [R for the Rest of Us](#).

💡 Le principe :

- construire des templates déjà designés ;
- qui sont personnalisables.

🎯 L'objectif :

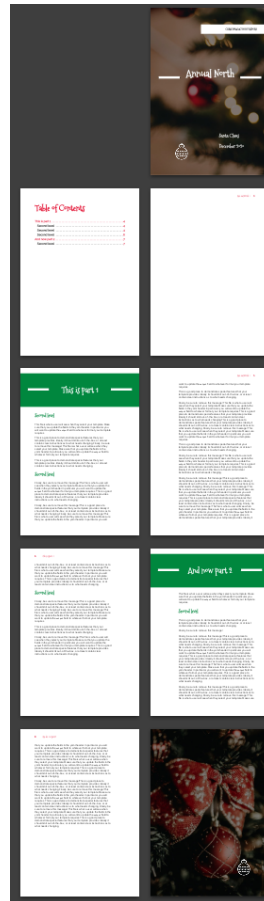
- construire des produits viables et utilisables rapidement ;
- itérer ensuite pour construire du sur-mesure.

{pagedreport}

💡 Tout se personnalise directement dans le YAML :

```
---
title: "Annual North"
subtitle: "Christmas activity report"
author: "Santa Claus"
date: "December 2020"
output:
  pagedreport::paged_hazelnuts:
    front_img: "https://images.pexels.com/photos/717988/pexels-photo
    back_img: "https://images.pexels.com/photos/3303615/pexels-photo
    logo: "https://www.flaticon.com/svg/static/icons/svg/268/268225..
    img_to_dark: TRUE
    logo_to_white: TRUE
knit: pagedown::chrome_print
toc-title: "Table of Contents"
main-color: "#E40A2D"
secondary-color: "#00873E"
google-font: TRUE
main-font: "Raleway"
header-font: "Mountains of Christmas"
---
```

{pagedreport}



Faire votre CV

Proposition la plus aboutie : pagedown - cv : le CV est une base de données (fichier Excel) et est généré ensuite.

Ulrik Lyngs <small>ulrik.lyngs@cs.ox.ac.uk https://ulriklyngs.com</small>			
Research summary		Grants	
Aims —to develop and evaluate design patterns that help people exercise self-control over their use of smartphones and laptops. Methods —controlled studies, user surveys, interviews, co-design workshops, web scraping, behaviour logging. I am passionate about open and transparent research. Since 2018, I have shared materials, analysis scripts, and data for all my work, and written my papers as reproducible documents in R Markdown ¹ .		2021 — 2023 Carlsberg Foundation Visiting Fellowship at University of Oxford ¹ (DKK 700,000 [~€85k]), Understanding Personal Digital Self-Control Struggles and Appropriate Interventions	
		2020 van Houten Fund ¹ (~€12k), The Reducing Digital Distraction (ReDD) Workshop for Oxford University students. Supervisor: Maureen Freed	
		2019 — 2020 EPSRC Doctoral Prize ¹ (~£28k), Supporting End-User Autonomy Over Digital Device Use. Supervisor: Nigel Shadbolt. Mentor: Max Van Kleef.	
		2018 Student Research Competition Grant (£500), CHI'18: ACM Conference on Human Factors in Computing Systems	
		2017 Workshop Travel Grant (£400), Designing for Curiosity, CHI'17: ACM Conference on Human Factors in Computing Systems	
		2017 Student Research Competition Grant (£500), CHI'17: ACM Conference on Human Factors in Computing Systems	
		2013 Lienhard & Bagby Research Grant (£1,000), Institute of Social & Cultural Anthropology, University of Oxford	
		2008 Summer School Scholarship (~€1000), University of Helsinki	
		2008 Undergraduate Travel Grant (DKK 1200), Aarhus University	
Education		Publications	
2016 — 2020	DPhil in Computer Science , University of Oxford Supervisor: Nigel Shadbolt Thesis: Examining the effectiveness of design patterns for digital self-control	Conference publications <small>In computer science, top-tier conferences (~30% acceptance rate) are as or more impactful than journals, see doi.org/10.1145/3285588 (fully reviewed, archival)</small>	
2012 — 2013	MSc in Cognitive and Evolutionary Anthropology , University of Oxford Supervisor: Oliver Curry Thesis: Fundamental motivations and risk-taking	2021 C6	How the Design of YouTube Influences User Sense of Agency K. Lukoff, U. Lyngs, H. Zade, V. Liao, J. Choi, K. Fan, S. Munson, A. Hinkler CHI'21: ACM Conference on Human Factors in Computing Systems
2011 — 2016	MA in the Study of Religion & Psychology , Aarhus University Supervisor: Lise Schjeldt Thesis: Religion as a cultural tool for cognitive control	2020 C5	From Ancient Contemplative Practice to the App Store: Designing a Digital Container for Mindfulness K. Lukoff, U. Lyngs, S. Gueorguieva, E. S. Dillman, A. Hinkler, S. A. Munson DIS'20: ACM conference on Designing Interactive Systems
2007 — 2011	BA in the Study of Religion & Psychology , Aarhus University Supervisors: Jesper Sørensen and Kristoffer Laigaard Niebo Thesis: Priming effects of religious concepts on moral judgment	C4	"I Just Want to Hack Myself to Not Get Distracted": Evaluating Design Interventions for Self-Control on Facebook U. Lyngs, K. Lukoff, P. Slovak, W. Seymour, H. Webb, M. Jiroka, J. Zhao, M. Van Kleef, N. Shadbolt CHI'20: ACM Conference on Human Factors in Computing Systems
Awards & honours		2019 C3	Self-Control in Cyberspace: Applying Dual Systems Theory to a Review of Digital Self-Control Tools U. Lyngs, K. Lukoff, P. Slovak, R. Birns, A. Slack, M. Indlich, M. Van Kleef, N. Shadbolt CHI'19: ACM Conference on Human Factors in Computing Systems 🏆 Best paper honourable mention (top 5%)
2020	Winner of the 2020 Oxford Three Minute Thesis competition² , Mathematical, Physical and Life Sciences Division, University of Oxford		
	Commendation Certificate for the Reducing Digital Distraction Workshop , MPS Impact Awards ¹ , University of Oxford		
2019	Best Talk Award , 2019 Oxford Computer Science Conference, University of Oxford		
	Best Paper Honourable Mention Award for "Self-Control in Cyberspace..." , CHI'19: ACM Conference on Human Factors in Computing Systems		
2018	Best Paper Award for "Third party tracking..." , WebSci'18: ACM Conference on Web Science, Amsterdam		
2017	DOMUS Prize for Outstanding Research Communication , Linacre College, University of Oxford		

¹ Last updated: January 2021

² Last updated: January 2021

Documentation

Un exemple : mon CV

**Et si je veux construire mon
propre thème ?**

Appliquer le reste du CSS

✓ On commence par construire les éléments classiques en CSS sur les différentes classes :

- Les polices

```
p {  
  font-family: "Roboto";  
}
```

```
h1, h2, h3 {  
  font-family: "Lato";  
}
```

- Les couleurs

```
ul, li {  
  color: #741336;  
}
```


Appliquer le reste du CSS

- Les tailles et styles

```
blockquote {  
  font-size: 12pt;  
  text-transform : uppercase;  
}
```

- Les bordures

```
h3 {  
  border-bottom: solid black 1px;  
}
```

- Aux espacements (line-height) et marges (margin et padding) -> 
attention aux interactions avec la suite !
- Et ainsi de suite...

Appliquer le reste du CSS

Il faut principalement penser :

- Aux titres, de `h1` à `h3`, voire plus si besoin
- Y compris les éléments de la première page : `h1.title`, `h1.subtitle`, `h2.date`, `h2.author`, sauf en ce qui concerne leur positionnement
- Aux paragraphes et listes : `p`, `li`, `ul`
- Aux liens : `a`
- Aux blocs : `blockquote`
- Aux tableaux éventuellement `tr`, `td`

 [Documentation Mozilla](#)

💡 Ce CSS peut être commun avec d'autres formats RMarkdown.

La page

✓ La page est l'élément central du package :

- Au travers de l'argument @page
- On fixe ses dimensions au début par

```
@page {  
  size: 210mm 297mm;  
}
```

- Certaines pages ont des caractéristiques particulières :
 - first et last
 - blank
 - left et right
 - les pages nommées

```
.chapter {  
  page: chapter;  
}
```

Tirer profit des contraintes de dimensions

📶 Dans le monde du Web, tout est relatif :

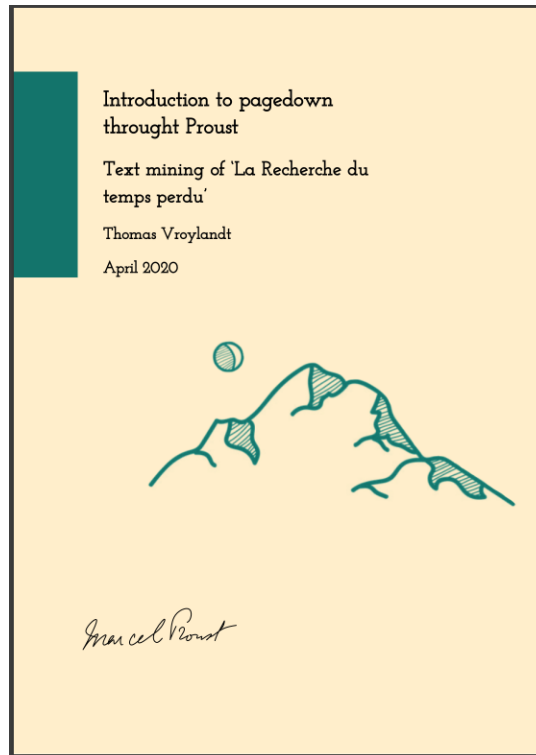
- les tailles des images
- les positions des éléments
- tout dépend de la taille de l'écran de l'utilisateur

💡 Ici, la dimension aide :

- les tailles des images sont fixes (ajouter un `img {max-width : 100%;}`)
- on peut utiliser le `position: absolute;` de façon récurrente pour placer les objets (sans que ce soit obligatoire)
- mais on peut avoir quelques problèmes sur les tableaux, avec des [solu-tions](#)

Page de garde et dernière page

Un exemple de première page :



Page de garde et dernière page

```
@page:first{  
  margin: 0 0 0 0;  
  background-image: url(../style/Marcel_Proust_signature.svg),  
                  var(--front-cover);  
  background-repeat: no-repeat, no-repeat;  
  background-size: 30%, 70%;  
  background-position: 100px 900px, 200px 500px;  
  background-color : #ffeeeb;  
}
```

Et

```
h1, h2, h3 {  
  color: var(--main-color);  
  font-weight: 900;  
}  
  
.title, .subtitle, .author, .date{  
  color : black;  
  margin-left: 1cm;  
  margin-right: 3cm;  
}
```

Page de garde et dernière page

 On peut paramétrer les images depuis le YAML :

Dans le YAML :

```
output:  
  pagedown::html_paged:  
    front_cover: style/mountain.png  
    back_cover: style/proust.png
```

Dans le CSS :

```
var(--front-cover) var(--back-cover)
```

Et on peut en mettre plusieurs - page 42.

Contenus dans les marges

On peut placer du contenu dans les marges

@top-left-corner	@top-left	@top-center	@top-right	@top-right-corner
@left-top	page area			@right-top
@left-middle				@right-middle
@left-bottom				@right-bottom
@bottom-left-corner	@bottom-left	@bottom-center	@bottom-right	@bottom-right-corner

Contenus dans les marges

Exemple de ma page de garde

```
@page: first {  
  @top-left {  
    content: none;  
  }  
  @top-right {  
    content: none;  
  }  
  @left-top {  
    content: "";  
    background-color: var(--main-color);  
  }  
  @left-middle {  
    content: "";  
    background-color: var(--color-paper);  
  }  
  @left-bottom {  
    content: "";  
    background-color: var(--color-paper);  
  }  
}
```

Contenus dans les marges

Ou pour créer une pagination alternée par exemple

```
@page:left {  
  @bottom-left {  
    content: counter(page);  
  }  
  
@page:right {  
  @bottom-right {  
    content: counter(page);  
  }  
}
```

Ou pour mettre une image depuis une variable CSS

```
@page chapter: {  
  @top-right {  
    content: string(title);  
    background-image: var(--state-shape);  
  }  
}
```


Sauts de pages

■ Pour sauter une page :

- la fin d'une page nommée termine la page ;
- on peut utiliser les fonctions CSS :
 - `page-break-after: always;` ou `avoid`
 - `page-break-inside`
 - avec des comportements parfois étranges

Autres fonctionnalités

D'autres fonctionnalités non évoquées existent ou sont en cours d'implémentation :

- table des matières ;
- [note de bas de page](#)

⚡ Et le package est en plein développement.

Développer en pratique

✓ Diviser ses fichiers CSS en :

- interface : tout ce qui sert à calibrer les fichiers : avec le paramètre `screen`
- modules : tout ce qui va effectivement faire le style
- [un exemple](#)

✓ Diviser les blocs de styles en plusieurs fichiers par thématique : layout, variables, colors, fonts, etc.

✓ Utiliser des variables CSS depuis : `root` pour construire le style : couleur, taille de marge, police, lien vers une image, etc.

```
:root {  
  /* fonts */  
  --font-body: "Calibri";  
  --font-title: "Calibri";  
  /* colors */  
  --main-color: #e5801c; /* light orange */  
  --secondary-color: #b35136; /* darker shade of orange */  
  /* img */  
  --state-shape: url(../img/state_shape/000_black_box.png);  
}
```

Développer en pratique

- ✓ Construire des templates Rmd pour une utilisation répétée
- ✓ Gérer les interactions depuis l'externe - option 1 : avec JavaScript
- 💡 On génère une variable CSS qu'on va réutiliser

```
{  
  let StateShapeCont = document.querySelector('h1.title')  
                        .textContent  
                        .replace(/\s+/g, "_");  
  let StateShapeContUrl =  
    `url(style/img/state_shape/${StateShapeCont}.png)`;  
  document.documentElement  
    .style  
    .setProperty('--state-shape', StateShapeContUrl);  
}
```

Développer en pratique

✓ Gérer les interactions depuis l'externe - option 2 : avec R

💡 On génère une variable CSS ou une image qu'on va réutiliser (nécessite un template)

```
if (img_to_dark == TRUE) {  
  front_img_init <-  
    magick::image_read(front_img)  
  front_img_ok <-  
    magick::image_colorize(front_img_init, opacity = 50, color = "black")  
  
  front_img <- paste0(tempfile("front_img"), ".jpg")  
  magick::image_write(front_img_ok, front_img, format = "jpg")  
}  
  
pagedown::html_paged(  
  front_cover = c(front_img),  
  ...  
)
```

Trouver de la documentation

📄 Regarder des utilisations existantes :

- Par [J-D Barillas](#)
- [Template SGDF](#)
- PROPRE (présentée plus bas)
- [Omni](#)

🖼️ [Présentation à uRos](#) par Romain Lesur

🔗 [Une PR sur le Github](#)

🔗 [Les issues](#)

📖 La documentation de [paged.js](#)

🔗 [Le Gitlab de paged.js](#)

Et en pratique, ça donne quoi ?

La démarche PROPRE - rapport

Pour plus de détail sur le projet, voir les présentations d'hier sur PROPRE et [gouvdown](#).

🔗 La [PR](#) et le [repo de démonstration de la brochure](#)

La démarche PROPRE - rapport

👥 Une organisation du travail originale avec un graphiste et un dev R : un grand merci à Julien Taquet.

✓ A la source des bonnes pratiques mentionnées plus haut

➤ Des interactions avec R pour définir une largeur de colonne variable

```
width_css <- tempfile(fileext = ".css")

writeLines(paste(
  paste0(":root {--main-column-width:", width_main_column , "%;}"),
  paste0(":root {--secondary-column-width:", 100 - width_main_column - 5,
    "%;}"), sep = "\n"
),
con = width_css)

# render
pagedown::html_paged(
  [...]
  css = c(css, book_css, width_css, made_with_css),
  ...
)
```

CT Data - fiches automatisées

 Un projet en collaboration avec [R for the Rest of Us](#) et [CTData](#) pour le [Partnership for Strong Communities](#)

 Générer des rapports automatisés sur chaque ville du Connecticut : Bloomfield



CT Data - fiches automatisées

🚫 La principale contrainte est la génération automatique :

- les graphiques ne comportent pas forcément le même nombre d'éléments
- des textes générés peuvent se retrouver à vide
- des trous dans les données peuvent créer des erreurs

✓ Il faut donc :

- bien spécifier la taille de chaque sortie
- remplir les valeurs manquantes dans les jeux de données et prévoir ces cas

📄 Pour une structure aussi stricte, la forme est très présente dans le Rmd :

- pages nommées
- utilisation de CSS pour les mises en colonnes

Merci de votre attention !

Merci aussi à Romain Lesur, Julien Taquet, Christophe Dervieux, Diane Beldame, David Keyes pour leurs échanges et leur travail sur le sujet !

 Pour discuter :

thomas@tillac-data.com