

# R-dyndoc une alternative à Sweave

R. Drouilhet

Laboratoire Jean Kuntzmann, Grenoble  
remy.drouilhet@upmf-grenoble.fr

**Mots clefs** : Statistique, Rapport automatique.

## 1 Motivation

Le projet **R-dyndoc** a commencé il y a une quinzaine d'années avec pour premiers objectifs :

- **Enseignement** : gain de temps lors de la rédaction des sujets d'examens et préparation de documents de cours.
- **Consulting statistique** : idée de proposer des outils de génération automatique de rapports en intégrant les résultats extraits de traitements **R**.
- **Programmation** : création d'un outil permettant la combinaison de plusieurs langages aux caractéristiques complémentaires (par exemple, la facilité du **R** à la manipulation et le traitement de données et la manipulation aisée en **ruby** des chaînes de caractères).

Grâce à la facilité à "embarquer" le système **R** dans des langages de programmation basés sur des **API C** (mon préféré étant **ruby**), il est aujourd'hui relativement facile de développer dans l'un de ces langages une alternative à **Sweave** (intégralement développé en **R**). L'un des avantages est certainement la possibilité offerte d'utiliser un plus grand nombre de bibliothèques.

Maintenant, **R-dyndoc** est un outil développé en **ruby** avec pour principales caractéristiques :

- **système de modèles ("templating system")** : basé sur un langage de balises (de type "domino") suffisamment exotique pour s'intégrer facilement dans tout document au format "human readable" (tel que **latex**, **html**, ...) afin d'en dynamiser le contenu.
- **langage (de script)** : permettant d'intégrer pleinement les langages **ruby** et **R** (avec possibilité d'intégration de **python** et tout autre langage interfaçable en **C**).

Les principaux avantages de **R-dyndoc** sont ses possibilités :

- de créer (dans un *mode développeur*) des bibliothèques regroupant des collections de fonctions et objets **R-dyndoc** (spécifiquement dédiés à la génération de parties textuelles relatives à certaines expertises) facilement utilisables (dans un *mode utilisateur*) à la rédaction finale de rapports.
- de rassembler dans un même document **R-dyndoc** (plus communément appelé "template" **R-dyndoc**) à la fois des traitements (**R** et **ruby**) effectués dans une analyse statistique (par exemple) et le contenu du rapport à diffuser au format **pdf** ou **html**.
- de créer, à partir d'un unique "template" **R-dyndoc**, plusieurs documents (éventuellement dans différents formats **latex** et **html**).

Notez que le **R-dyndoc** n'est pas encore disponible en version stable. Il est toutefois possible de le tester en installant deux bibliothèques **ruby** (appelés **gems**) ainsi que deux packages **R** (voir page web <http://dyndoc.upmf-grenoble.fr/DyndocInstall.html>).

## 2 Quelques exemples

Pour conclure, illustrons à travers quelques exemples basiques (malheureusement non commentés ici par manque de place) quelques fonctionnalités offertes par R-dyndoc.

```
1  [#r<] mister <- "misteR" #R code
2  [#rb<] mister = "mister" #ruby code
3  [#<] This is not output!
4      {#def}hello[#,]name[Miss][#>]
5          [Hello #{name}]
6      [#}
7  [#>]
8      [from Dyn, {#hello#}
9          |from Dyn, {#hello}Mister[#}
10         |from ruby, hello :{mister}
11         |from R, hello :r{mister}]
```

Résultat :

```
from Dyn, Hello Miss
from Dyn, Hello Mister
from ruby, hello mister
from R, hello mister
```

```
1  [#=] docs [part2,part3,part1]
2  [#>]{#case}#{docs}
3      [#when]part1[#>]Partie~ I
4      [#when]part2[#>][Partie~ II<\n>]
5      [#when]part3[#>]Partie~ III
6  [#}|
```

Résultat :

```
Partie~II
Partie~III
Partie~I
```

```
1  [#>]for loop in R:
2  [#R>] for(cpt in 1:4) {
3      {#>}item:r{cpt} [#>}
4  }
5  [#>]<\n>sapply loop in R:
6  [#R>] sapply(5:8,function(cpt)
7      {#>}item#r{cpt} [#>}
8  )
```

Résultat :

```
for loop in R:
item1 item2 item3 item4
sapply loop in R:
item5 item6 item7 item8
```

```
1  [#=]toto[TOTO]
2  [#=]toto$c(1,3,2)
3  [#>]Before: #{toto} AND #{toto$}
4  [#R<]
5  <toto:> = tolower(<toto:>)
6  <toto$>[1] = 4
7  [#>]After: #{toto} AND #{toto$}
```

Résultat :

```
Before: TOTO AND 1.0,3.0,2.0
After: toto AND 4.0,3.0,2.0
```

Les sorties suivantes montrent le même exemple en Sweave puis en R-dyndoc :

```
1  \documentclass{article}
2  \begin{document}
3  <<==
4  a<-c(1,3,2)
5  a
6  @
7  La moyenne est \Sexpr{mean(a)}.
8  \end{document}
```

```
1  {#rverb}
2  a<-c(1,3,2)
3  a
4  [#}
5  La moyenne est :r{mean(a)}.
```