

Automatiser la production de fiches. Exemple: la création d'un atlas botanique

Fabrice DESSAINT

Inra – Agroécologie, Dijon

24 mai 2013

Les données : la zone atelier de Féney (21)

- \approx 200 parcelles agricoles
- Cinq années de relevés : 2008 – 2012 (2013 en cours)
- Un (2) relevé(s) botanique(s) par an (note d'abondance)
- Deux zones observées : le plein champ (PC) et le bord intérieur du champ (IF)
- \approx 200 taxons

- Base de données sous Access
- Parcellaires dans une base SIG, sous ArcGis
- Référentiel taxonomique (ITIS : Integrated Taxonomic Information System)

La demande et la proposition

La demande :

- Production d'un « carnet de terrain » / « atlas » botanique
- Une fiche par taxon observé
 - ▶ informations sur le taxon
 - ▶ cartes annuelles de répartition
- Mise à jour facile et automatique
- Accès à des sources d'informations hétérogènes
- Possibilité d'évolution (ajout de photos)

La proposition :

- L^AT_EX
- R + R.rsp (version 0.8.2).

La sortie : un format de sortie possible

Alopecurus myosuroides

Nom valide : *Alopecurus myosuroides* Hudt. (source ITIS)

Classe : Magnoliopsida
 Ordre : Poales
 Famille : Poaceae
 Genre : Alopecurus
 Code EPPO : ALOMY

Une description des caractéristiques du taxon est accessible sur le site de Tela Botanica.

Distribution des notes selon les années

Compartment PC

	Notes d'abondance				Total	
	0/1	2	3	4		
2008	0	11	1	0	0	12
2009	3	8	1	0	0	12
2010	3	13	2	0	0	18
2011	2	10	1	0	0	13
2012	2	8	1	1	0	12

Compartment IF

	Notes d'abondance				Total	
	0/1	2	3	4		
	3	18	5	2	0	28
	1	13	1	3	0	18
	2	23	5	5	1	38
	0	28	11	4	0	43
	0	19	8	5	0	32

Distribution des notes selon les cultures

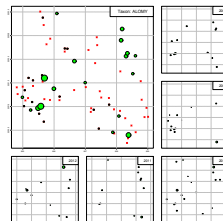
Compartment PC

	Notes d'abondance					Total
	0/1	1	2	3	4	
Ble hiver	2	15	2	0	0	19
Colza hiver	3	11	2	0	0	16
Luzerne	0	1	0	0	0	1
Mais	0	0	0	0	0	0
Monarde	2	8	0	0	0	11
Orge hiver	1	6	0	1	0	8
Orge printemps	0	6	1	0	0	7
Pois printemps	0	0	0	0	0	0
Seigle	0	0	0	0	0	0
Tournefort	1	0	0	0	0	1
Triticale	0	3	1	0	0	4

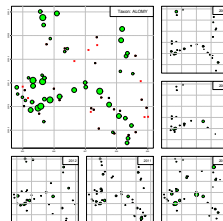
Compartment IF

	Notes d'abondance					Total
	0/1	1	2	3	4	
	1	39	10	6	0	57
	0	20	4	5	1	30
	0	0	0	0	0	0
	0	1	0	0	0	1
	1	6	5	1	0	14
	0	25	6	5	0	36
	1	6	3	0	0	10
	0	0	0	1	0	1
	1	1	1	0	0	3
	1	2	0	0	0	3
	1	1	1	1	0	4

Carte de répartition de ALOMY dans le compartiment PC



Carte de répartition de ALOMY dans le compartiment IF



R.rsp : flux de traitement

- 1 un fichier « texte » avec balises RSP et une double extension :

Atlas.tex.rsp

- 2 utilisation de la fonction `rsp()` sous R

```
rsp("Atlas.tex.rsp")
```

- 3 production d'un script R

Atlas.tex.rsp.R

- 4 exécution du script et production du fichier « texte »

Atlas.tex

Pour certains types de fichiers, le flux de traitement peut continuer. C'est le cas des fichiers \LaTeX .

- 5 compilation du fichier avec PDF \LaTeX et production d'un fichier .pdf

Atlas.pdf

RSP : un langage de balises

Le package R.rsp utilise un langage de balises.

Balise

- 4 balises de traitement (exécution du code R)
 - ▶ `<% code %>` : pas d'insertion.
 - ▶ `<%= code %>` : insère le résultat seul dans le document.
 - ▶ `<%: code %>` : insère le code seul dans le document.
 - ▶ `<%=evalCapture({code})%>` : insertion du code et/ou du résultat dans le document.
- une balise de commentaire.
 - ▶ `<!-- commentaire -->`

Une fonction `rsp()`

RSP : la balise <% %>

Pas d'insertion du code ou du résultat

Code

```
Le vecteur x contient les 10 premiers entiers<%x <- 1:10; x%>.
```

donne après traitement

Sortie

Le vecteur x contient les 10 premiers entiers.

Le vecteur x est créé et disponible pour des traitements ultérieurs.

RSP : la balise `<%= %>`

Insertion du résultat seul

Code

```
La racine carré de 25 est égale à <%=sqrt(25)%>.
```

donne après traitement par la fonction `rsp()`

Sortie

La racine carré de 25 est égale à 5.

Le résultat est inséré sous la forme d'une chaîne de caractère.

RSP : la balise `<%= %>`

Si il y a plusieurs instructions R

Code

```
Plusieurs instructions: <%=x; sqrt(25)%>.
```

seul le résultat de la dernière est retourné.

Sortie

```
Plusieurs instructions : 5.
```

RSP : la balise `<%= %>`

Si le résultat contient plusieurs valeurs

Code

```
Le vecteur x contient les 10 premiers entiers: <%=x%>
```

la chaîne insérée, les contient tous.

Sortie

```
Le vecteur x contient les 10 premiers entiers : 12345678910
```

RSP : la balise `<%= %>`

On peut utiliser la fonction `paste()` (ou la fonction `hpaste()`).

Code

Le vecteur `x` contient les 10 premiers entiers:

```
<%=paste(x,collapse=",")%>
```

donne

Sortie

Le vecteur `x` contient les 10 premiers entiers : 1,2,3,4,5,6,7,8,9,10

RSP : <%: %>

Insertion du code seul

Code

```
<%:  
x <- 1:10; # 10 premiers entiers  
x  
sqrt(5)  
%>
```

donne

Sortie

```
x <- 1:10; # 10 premiers entiers  
x  
sqrt(5)
```

On a une copie *verbatim* des instructions R : absence de l'invite (prompt) de R.

RSP : `<%=evalCapture({})%>`

Code

```
<%=evalcapture({  
x <- 1:10; # 10 premiers entiers  
sqrt(5)  
x  
},code=TRUE,output=TRUE)%>
```

Deux arguments, `code` et `output`, permettent l'insertion ou non du code et/ou du résultat.

Sortie

```
> x <- 1:10  
> sqrt(25)  
[1] 5  
> x  
[1] 1 2 3 4 5 6 7 8 9 10
```

Comportement analogue à celui de Sweave, par exemple.

RSP : balises

Attention!!

- Le code et les sorties ne sont pas *protégées*
- Il n'y a pas d'environnements particuliers (Soutput ou Sinput de Sweave)
- Penser à l'ajouter (verbatim sous \LaTeX).

RSP : les gabarits

- Le langage offre la possibilité de créer des « gabarits » (*templates*).
- La démarche est analogue à la création d'une fonction dans R.

Code

```
<% somme <- function(n, ...){ %>
La somme de $x=<%=hpaste(1:n, abbreviate="\ \ldots") %>$
est égale à $<%=sum(1:n)%>$.
<% } # Fin de la fonction %>
```

L'insertion du code `<%=somme(n=3)%>` dans un texte, produit :

Sortie

La somme de $x = 1, 2, 3$ est égale à 6.

RSP : les constructions

- Le langage offre aussi la possibilité de faire des itérations

Code

```
\begin{itemize}  
<% for (k in c(5,10,100) ) { %>  
\item <%somme(n=k) %>  
<% } %>  
\end{itemize}
```

produit

Sortie

- La somme de $x = 1, 2, 3, 4, 5$ est égale à 15.
- La somme de $x = 1, 2, 3, \dots, 10$ est égale à 55.
- La somme de $x = 1, 2, 3, \dots, 100$ est égale à 5050.

Application : Atlas.tex.rsp

```
\documentclass[11pt,french]{scrreprt}
\begin{document}

<% source("Lecture.R")%>

<!-- Définition de gabarits --%>
<% tableau <- fonction(zone, var, ligne, ...) {%>
...
<% } # fin de la fonction tableau %>
<% carte <- fonction(zone,...) { %>}
...
<% } # fin de la fonction carte %>

<!-- Début de la boucle sur les taxons --%>
<% for (k in 1:nbTaxons) { %>

<% tableau(zone="PC",var=c(1,2),ligne=TRUE) %>
...
<% carte(zone="PC") %>
<% } # Fin de boucle %>
\end{document}
```

Application

```
<!-- Fonction tableau --%>
<% tableau <- fonction(zone, var, ligne, ...) {%>
\begin{minipage}[t]{.50\linewidth}
\small
\textbf{Compartiment <%=zone%>}
\medskip

<%=evalCapture({
... instructions R
},code=FALSE,output=TRUE)
%>
\end{minipage}
<% } # fin de la fonction tableau %>
<%----%>
```

Application

```
<!-- Fonction carte -->
<% carte <- fonction(zone,...) { %>
\subsection*{Carte de répartition de <%=Code%> dans le compartiment <%=zone%>}
<%
... instructions R
%>
\centerline{
\includegraphics[height=0.5\textheight,
                 page=<%=ifelse(zone=="PC", (k-1)*2+1, (k-1)*2+2)%>]{Cartes}
}
<% } # fin de la fonction carte %>
```

Application

```
<!-- Fichier des sorties graphiques -->
<%
CairoPDF(file="Cartes.pdf",width=15,height=15,onefile=TRUE)
m <- matrix(c(1,1,2,
              1,1,3,
              6,5,4),nrow=3,byrow=TRUE)
layout(m,widths=c(1,1,1),heights=c(1,1,1))
par(oma=rep(2,4),mar=rep(2,4),bg="transparent",pty="s")
%>
...

<% dev.off() %>
```

Alopecurus myosuroides

Nom valide : *Alopecurus myosuroides* Hud. (source ITIS)

Classe : Magnoliopsida
 Ordre : Poales
 Famille : Poaceae
 Genre : Alopecurus
 Code EPO : ALOMY

Une description des caractéristiques du taxon est accessible sur le site de Tela Botanica.

Distribution des notes selon les années

Compartiment PC

	Notes d'abondance					Total
	0/1	2	3	4	5	
2008	0	11	1	0	0	12
2009	3	8	1	0	0	12
2010	3	13	2	0	0	18
2011	2	10	1	0	0	13
2012	2	8	1	1	0	12

Compartiment IF

	Notes d'abondance					Total
	0/1	2	3	4	5	
	3	18	5	2	0	28
	1	13	1	3	0	18
	2	23	5	5	1	38
	0	28	11	4	0	43
	0	19	8	5	0	32

Distribution des notes selon les cultures

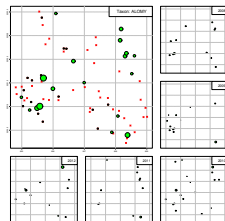
Compartiment PC

	Notes d'abondance					Total
	0/1	2	3	4	5	
Ble hiver	2	15	2	0	0	19
Colza hiver	3	11	2	0	0	16
Luzerne	0	1	0	0	0	1
Mais	0	0	0	0	0	0
Monarde	2	8	0	0	0	11
Orge hiver	1	6	0	1	0	8
Orge printemps	0	6	1	0	0	7
Pois printemps	0	0	0	0	0	0
Seigle	0	0	0	0	0	0
Tournefort	1	0	0	0	0	1
Triticale	0	3	1	0	0	4

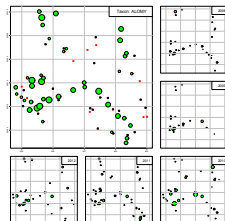
Compartiment IF

	Notes d'abondance					Total
	0/1	2	3	4	5	
	1	39	10	6	0	57
	0	20	4	5	1	30
	0	0	0	0	0	0
	0	1	0	0	0	1
	1	6	5	1	0	14
	0	25	6	5	0	36
	1	6	3	0	0	10
	0	0	0	1	0	1
	1	1	1	0	0	3
	1	2	0	0	0	3
	1	1	1	1	0	4

Carte de répartition de ALOMY dans le compartiment PC



Carte de répartition de ALOMY dans le compartiment IF



Conclusion

Le package R.rsp

- Le balisage RSP peut être utilisé pour créer différents types de documents/fichiers
 - ▶ des documents de présentation : .html, .xml, .pdf ...
 - ▶ des fichiers de type Sweave, knitr, \LaTeX , ...
 - ▶ des scripts de type R, javascript, php, SVG, ...
- Contrainte : documents au format texte