

# Bulletin de la Dialyse à Domicile

## Réaliser des tableaux de bords avec le logiciel R

Claire Della Vedova

1087 chemin de Sainte Roustagne, 04100 MANOSQUE, France



### Résumé

NDLR : Le RDPLF a pour but principal d'être une aide pour permettre aux équipes de dialyse à domicile d'évaluer leurs pratiques cliniques et également conduire des études à partir d'exports anonymisés des données qu'elles saisissent. A cette fin, depuis juin 2019, un article de formation à l'utilisation du logiciel Libre R a été publié trimestriellement à chaque parution du Bulletin de la Dialyse à Domicile. Le but était de permettre à toutes les équipes de réaliser des statistiques de base et visualiser rapidement leurs données.

Le premier article de cette série d'initiation était consacré au téléchargement et à l'installation du logiciel R sur les ordinateurs Macintosh et PC : <https://doi.org/10.25796/bdd.v2i2.20513>.

Le second était consacré à la visualisation graphique des données statistiques avec le package Esquisse, simple d'utilisation et nécessitant peu d'apprentissage : <https://doi.org/10.25796/bdd.v2i3.21313>.

Le troisième article était consacré à la visualisation graphique avec le Package ggplot2 : <https://doi.org/10.25796/bdd.v2i4.52303>

La formation totale s'est déroulée sur 15 mois, au rythme d'un article par trimestre à chaque parution du BDD. Cela aura laissé largement le temps d'assimiler et tester les connaissances acquises entre chaque article. Pour ceux qui souhaiteront aller plus vite, ils pourront aller sur le blog (<https://statistique-et-logiciel-r.com/>).

Le quatrième article expliquait comment générer un rapport d'analyse automatiquement, et de façon dynamique, en utilisant "Rmarkdown".

Le cinquième article était consacré à la manipulation des données avec le package dplyr : <https://doi.org/10.25796/bdd.v3i1.54523>

Ce sixième et dernier article explique la réalisation des tableaux de bords avec le logiciel R : <https://doi.org/10.25796/bdd.v3i3.58203>

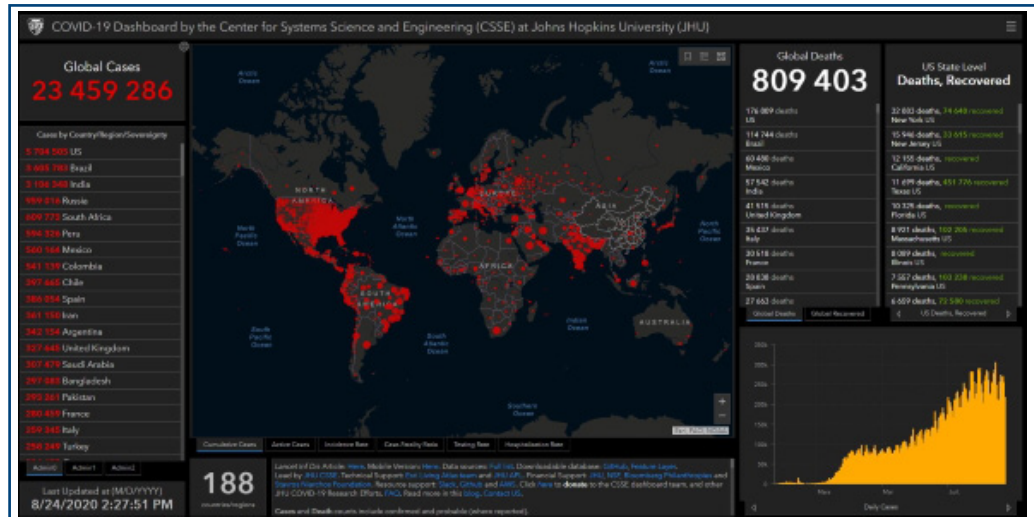
Comme dans les numéros précédent un fichier exemple, tiré de la base de données du RDPLF sera utilisé.

Claire Della Vedova est Ingénieure en biostatistique / data analyste, Elle utilise quotidiennement le logiciel R pour analyser des données. Elle a travaillé pendant plus de 15 ans dans les domaines de l'environnement et de la santé, et a formé de nombreux étudiants et chercheurs à l'utilisation de R. Elle anime depuis novembre 2017 le blog Statistique et Logiciel R dont le but est d'aider les débutants à mieux appréhender les méthodes statistiques classiques et à utiliser le logiciel R plus efficacement, notamment au travers de tutoriels : <https://statistique-et-logiciel-r.com/>.

Mots clés : Statistiques, épidémiologie, RDPLF, Logiciel libre R, cours de statistiques



Réaliser un tableau de bord (ou dashboard) pour communiquer et partager une analyse statistique peut être particulièrement efficace. J'en veux pour preuve les nombreux tableaux de bord qui ont été publiés pour décrire l'évolution de l'épidémie du Covid-19. Par exemple, celui-ci :



↑Fig. 1  
 Consultable à l'adresse : <https://gisanddata.maps.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6>

Ou encore celui-là :



↑Fig. 2  
 consultable à l'adresse : <https://mapthenews.maps.arcgis.com/apps/opsdashboard/index.html#/5e09dff7cb434fb194e22261689e2887>

Comme vous allez le voir, il est relativement facile de réaliser des tableaux de bord avec le logiciel R.

En effet, il existe un package dédié, le package *flexdashboard*. Ce dernier utilise le format R markdown dont nous avons parlé dans un précédent article (<https://bdd.rdpf.org/index.php/bdd/article/view/54523/51453>).

Dans cet article-ci, je vais vous montrer comment construire, en pas à pas, un tableau de bord simple, avec ce package *flexdashboard*.

## Préambule

### Installation du package flexdashboard

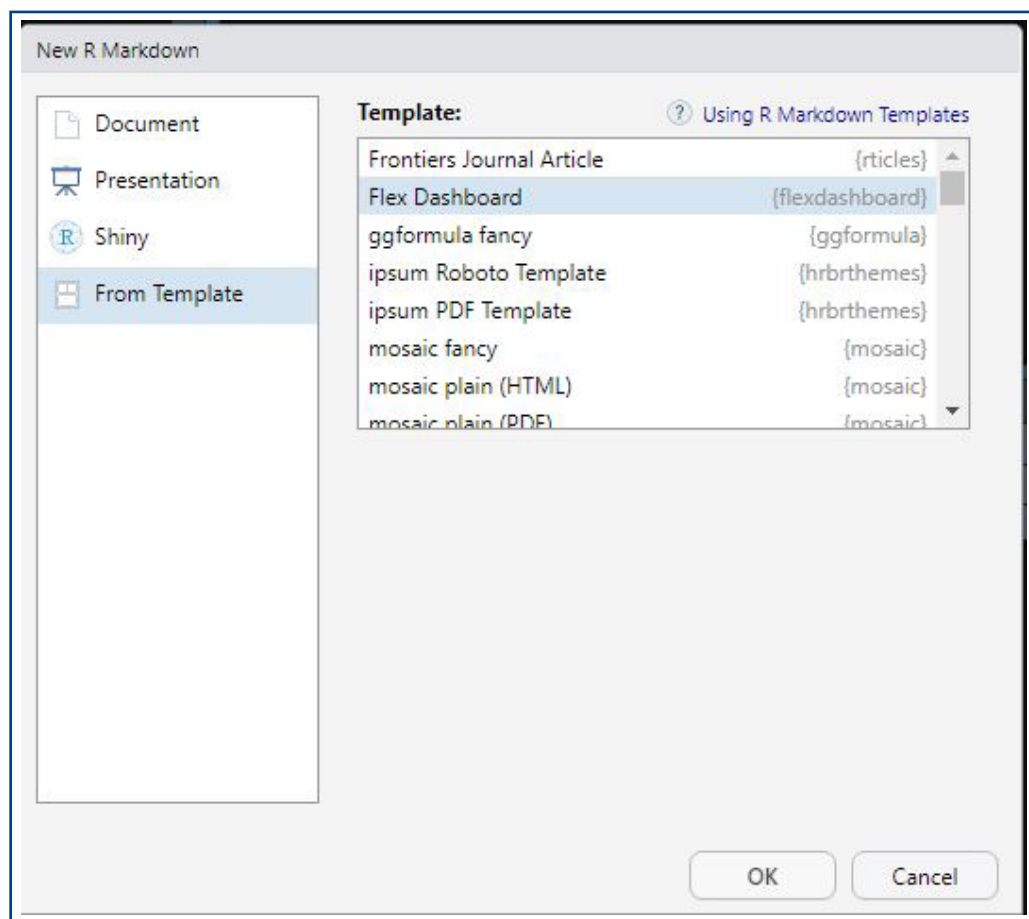
Pour commencer, il faut installer et charger le package *flexdashboard*. Pour cela, vous pouvez exécuter les deux commandes suivantes :

```
#install.packages(«flexdashboard»)  
library(flexdashboard)
```

### Ouverture d'un modèle

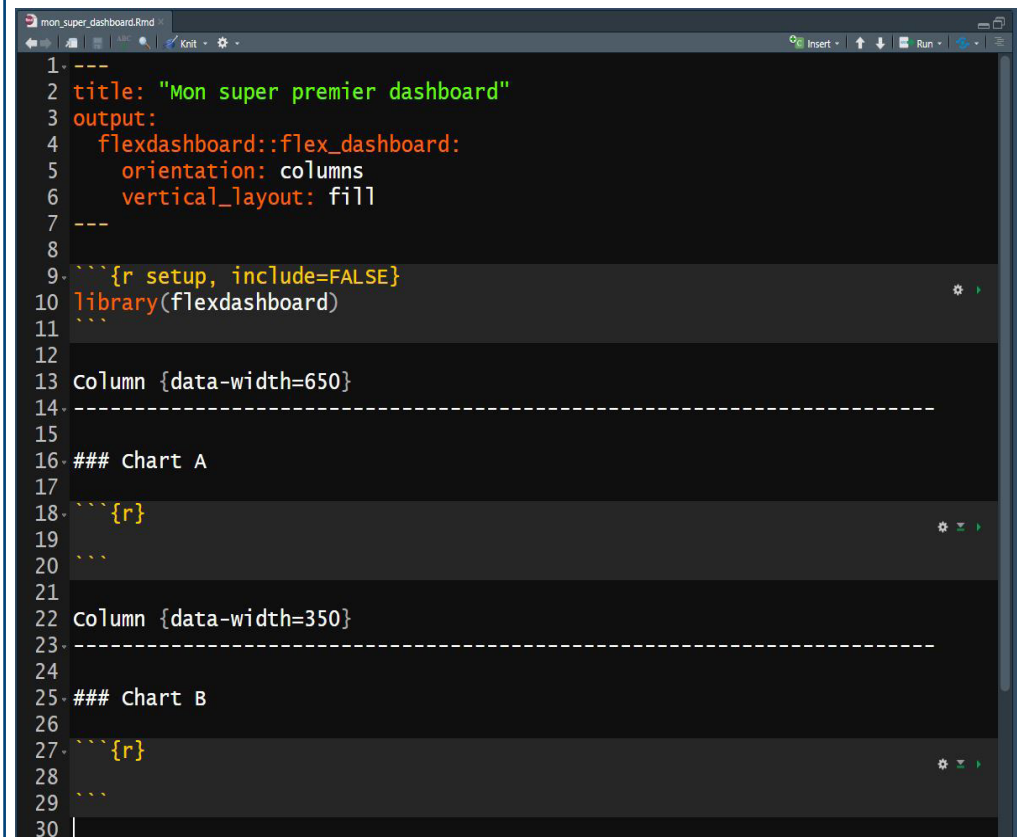
Il est ensuite nécessaire d'ouvrir un modèle (ou template) de fichier R markdown, spécifique aux tableaux de bord.

Pour cela : File -> New File -> Rmarkdown -> From Templates -> Flex Dashboard



↑ Fig. 3

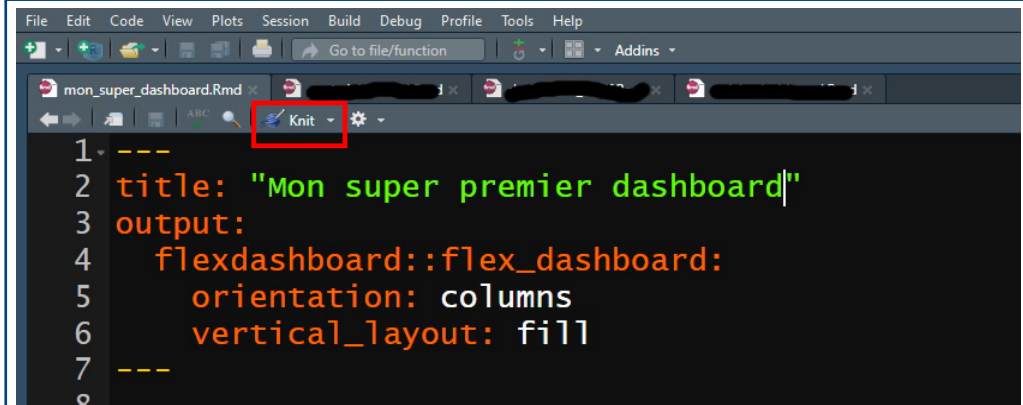
Cela va entraîner la création d'un fichier similaire à celui ci-dessous. J'ai seulement ajouté un titre de document : "Mon super premier dashboard".



```
1 ---
2 title: "Mon super premier dashboard"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7 ---
8
9 ```{r setup, include=FALSE}
10 library(flexdashboard)
11 ```
12
13 column {data-width=650}
14 -----
15
16 ### Chart A
17
18 ```{r}
19
20 ```
21
22 column {data-width=350}
23 -----
24
25 ### Chart B
26
27 ```{r}
28
29 ```
30
```

↑ Fig. 4

En knitant ce template, c'est-à-dire en cliquant sur le bouton Knit avec la pelote :

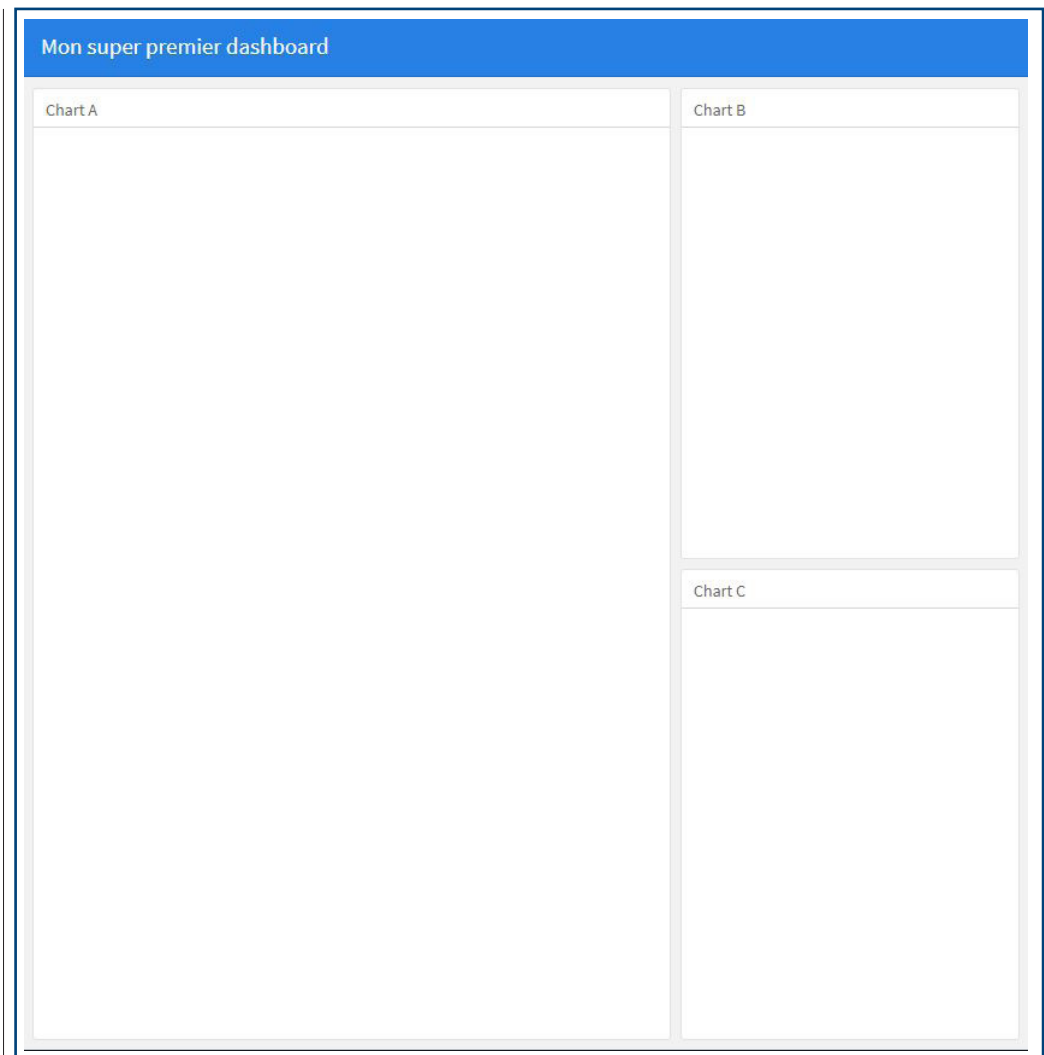


```
File Edit Code View Plots Session Build Debug Profile Tools Help
mon_super_dashboard.Rmd
Knit
```

```
1 ---
2 title: "Mon super premier dashboard"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: columns
6     vertical_layout: fill
7 ---
8
```

↑ Fig. 5

nous obtenons ce rendu :



↑ Fig. 6

Il s'agit d'un tableau de bord vide, que nous allons donc remplir !  
Mais avant cela, nous allons voir les possibilités offertes par *flexdashboard*, en termes d'éléments et de dispositions.

### Les possibilités de flexdashboard

#### Les principaux éléments

Le tableau de bord peut contenir de nombreux éléments. Ceux-ci sont créés en employant des fonctions disponibles au sein du package *flexdashboard*.

Ces fonctions sont utilisées à l'intérieur de chunks puisqu'il s'agit de code R. Si, cette notion ne vous est pas familière, je vous conseille de consulter l'article dédié à R markdown (<https://bdd.rdpf.org/index.php/bdd/article/view/54523/51453>.)

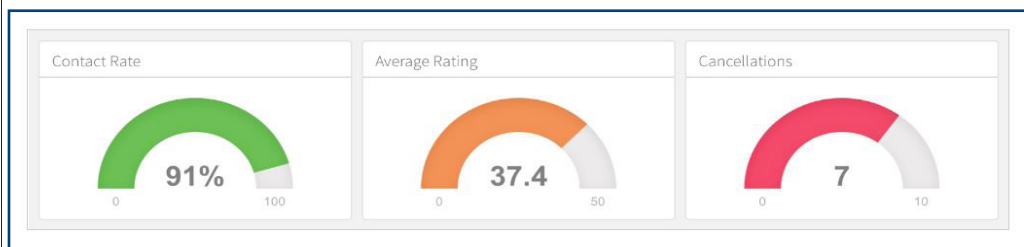
### Des boîtes avec des icônes modifiables



↑Fig. 7

Comme nous le verrons plus loin dans l'exemple, ces boîtes sont créées à l'aide de la fonction `valueBox()`. Les icônes peuvent être choisies sur le site "Font Awesome" (<https://fontawesome.com/icons?d=gallery>).

### Des gauges :



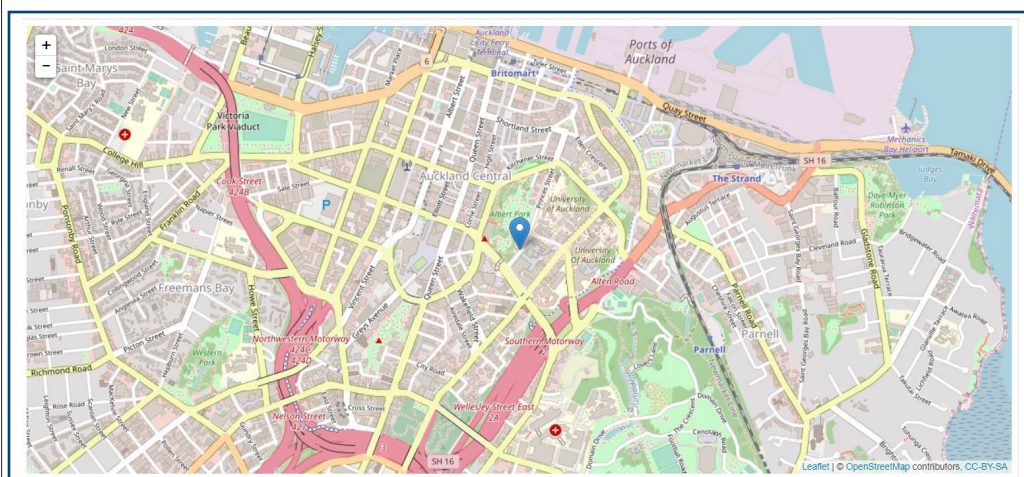
↑Fig. 8

Ces gauges sont créées en employant la fonction `gauge()`.

### Autres éléments

Les dashboards peuvent également contenir

- des graphes
- des tables
- des widget html comme une carte leaflet (<https://rstudio.github.io/leaflet/>),



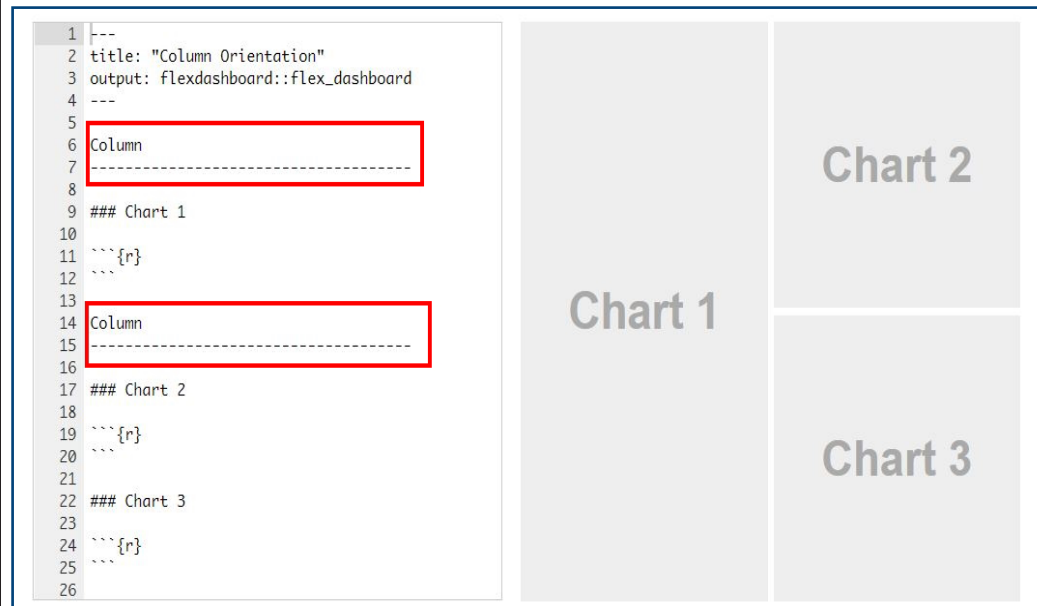
↑Fig. 9

Vous pourrez consulter un tutoriel ici : <https://statistique-et-logiciel-r.com/premiers-pas-en-cartographie-avec-r/>

- ou encore, plus simplement, du texte.

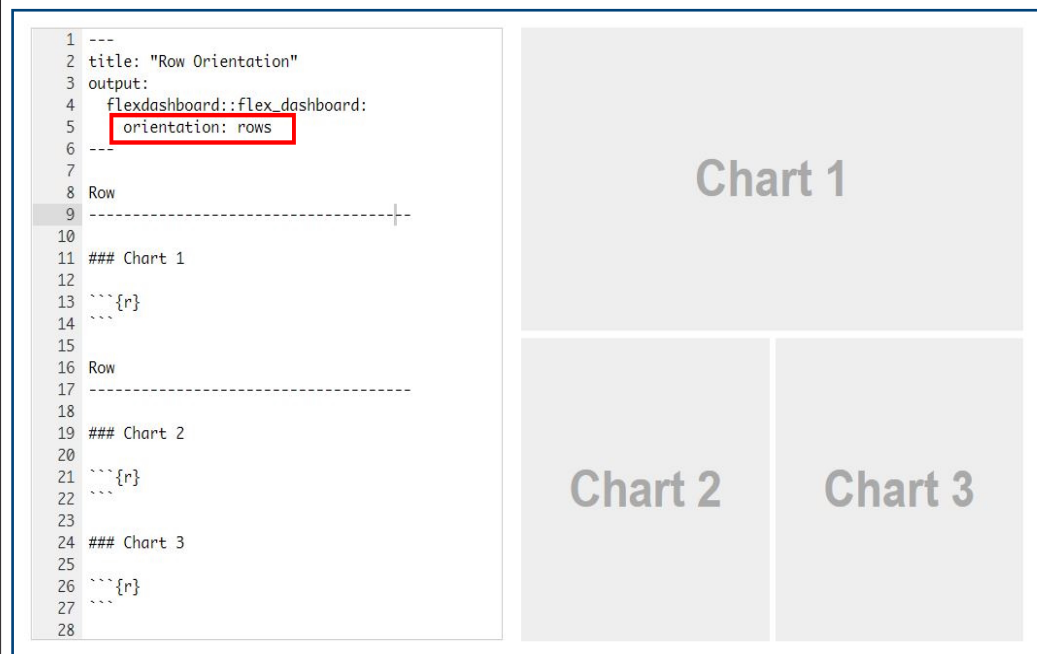
## Les dispositions

Par défaut, le dashboard est organisé en deux colonnes. Celles-ci sont séparées, dans le script, par un trait discontinu :



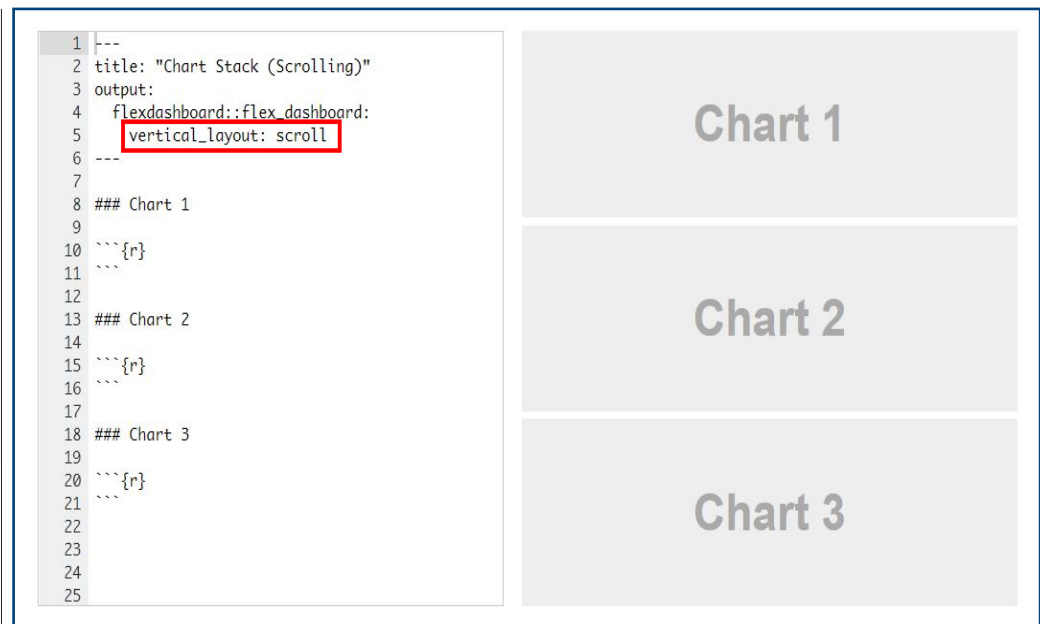
↑ Fig. 10

Une disposition en ligne peut être employée. Pour cela l'option *orientation : rows* est ajouté dans la section *output* de l'en-tête :



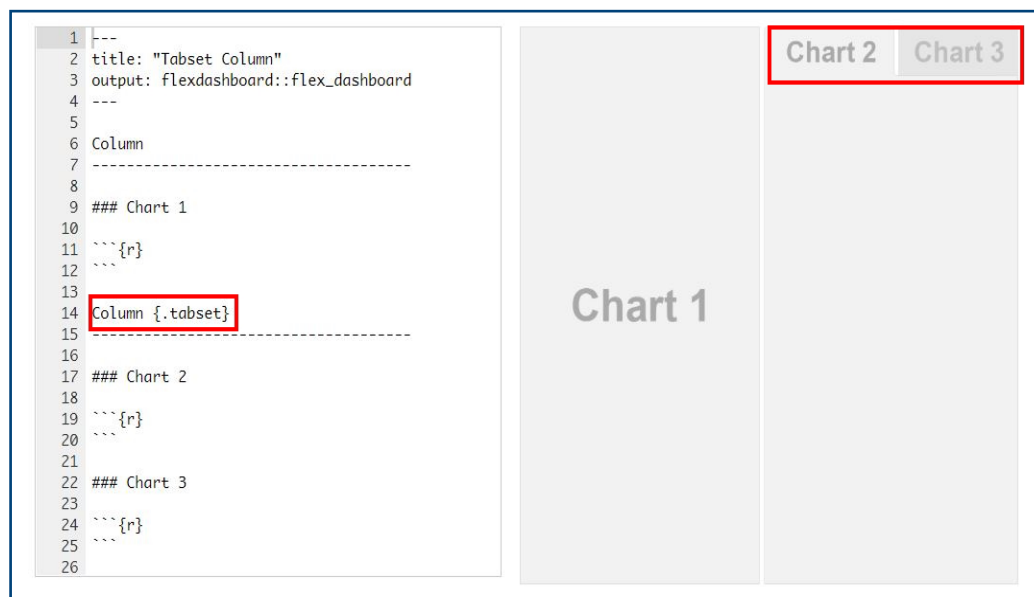
↑ Fig. 11

Une barre de scrolling peut également être ajoutée :



↑ Fig. 12

Ou encore des onglets (ou tabsets) :

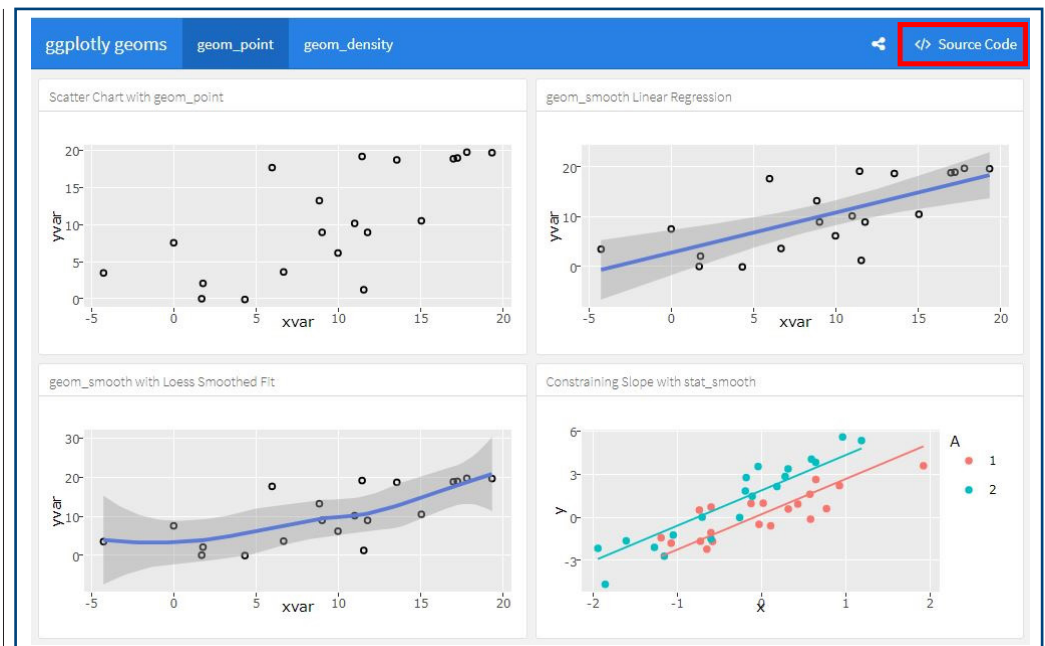


↑ Fig. 13

### Mises en page secondaires

Le tableau de bord peut également contenir plusieurs pages. Dans l'exemple ci dessous ([https://beta.rstudioconnect.com/jjallaire/htmlwidgets-ggplotly-geoms/htmlwidgets-ggplotly-geoms.html#geom\\_point](https://beta.rstudioconnect.com/jjallaire/htmlwidgets-ggplotly-geoms/htmlwidgets-ggplotly-geoms.html#geom_point)), il ya une page pour les *geom\_point*, et une page pour les *geom\_density*.





↑ Fig. 14

Nous le verrons plus tard dans l'exemple que nous allons élaborer, la création de ces différentes pages se fait simplement en ajoutant une double ligne dans le fichier de script :

```

23 Description
24 =====
25
26 Row
27 -----
28
29 ### Nombre de patients
30
31 {r}
32 nb_patients <- nrow(mydata)
33 valueBox(nb_patients, icon="fa-user")
34 {r}
35
36 ### Nombre de pays
37
38 {r}
39 nb_pays <- length(unique(mydata$PAYS))
40 valueBox(nb_pays, icon="fa-flag")
41 {r}
42
    
```

↑ Fig. 15

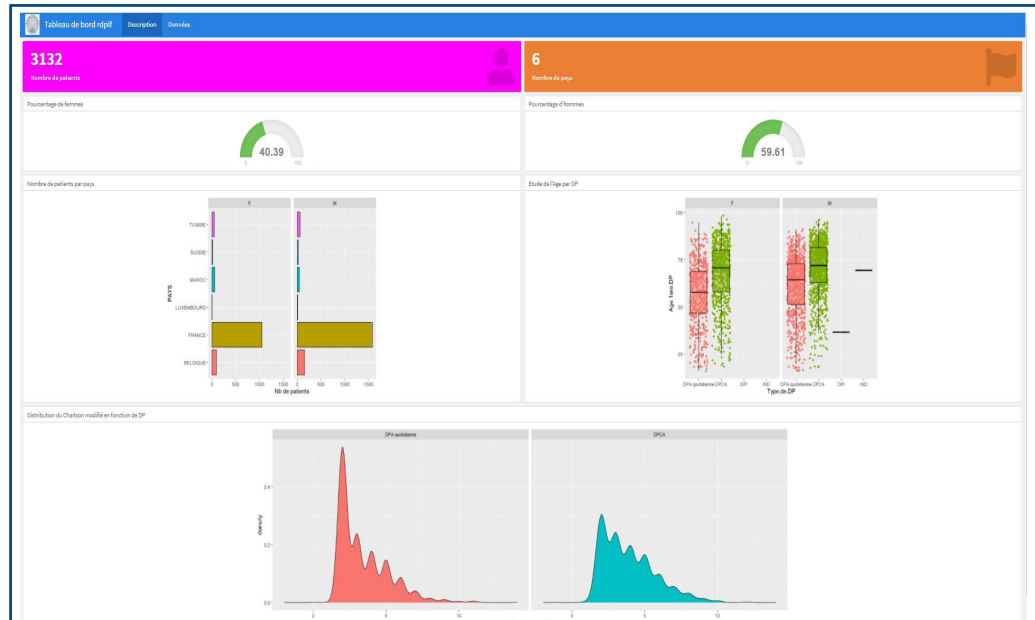
Des thèmes sont encore directement utilisables. Et l'apparence du tableau de bord peut être géré via un fichier css.

Enfin, vous pouvez encore ajouter un logo !

Pour plus d'informations sur les diverses possibilités de ces tableaux de bords construits avec *flexdashboard* vous pouvez consulter la documentation du package : <https://rmarkdown.rstudio.com/flexdashboard/using.html>

## Création d'un tableau de bord

Voici le tableau de bord que nous allons construire :



↑ Fig. 16

Vous pouvez le visualiser ici : [https://rpubs.com/blog\\_slr/tableau\\_bord\\_rdplf](https://rpubs.com/blog_slr/tableau_bord_rdplf)

Les données employées sont les même que celles des articles précédents, elles sont téléchargeables au format csv à cette adresse : <https://www.rdpf.org/exempleR/FichierExempleStat.csv>.

Voici des copies écran des différentes parties du dashboard :

### L'en-tête

```
1 ---
2 title: "Tableau de bord rdplf"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: rows
6     vertical_layout: scroll
7     logo: logo_journal2.jpg
8 editor_options:
9   chunk_output_type: console
10 ---
11
```

↑ Fig. 17

J'ai donc choisi une disposition en ligne, avec une barre de scrolling verticale, et j'ai ajouté un logo (attention l'image doit se trouver dans le même emplacement que le script qui sert à réaliser le tableau de bord).

## Le chunk set up

```
11  
12. ````{r setup, include=FALSE}  
13 library(flexdashboard)  
14 library(knitr)  
15 library(ggplot2)  
16 library(tidyverse)  
17 library(DT)  
18 mydata <- read.csv2(here::here("data", "FichierExemplestat.csv" ))  
19  
20  
21
```

↑ Fig. 18

C'est ici que nous pouvons charger les packages qui seront nécessaires, ainsi que le jeu de données.

## Page Description

### Les boîtes de la première ligne

```
22  
23 Description  
24 =====  
25  
26 Row  
27 -----  
28  
29 ### Nombre de patients  
30  
31 ````{r}  
32 nb_patients <- nrow(mydata)  
33 valueBox(nb_patients, color="magenta", icon="fa-user")  
34  
35  
36 ### Nombre de pays  
37  
38 ````{r}  
39 nb_pays <- length(unique(mydata$PAYS))  
40 valueBox(nb_pays, icon="fa-flag", color="#eb8034")  
41  
42
```

↑ Fig. 19

Comme expliqué précédemment, le double trait permet de créer une page qui sera intitulée “Description”.

Puis le simple trait, en dessous du mot clé “Row” permet de créer la première ligne du tableau de bord.

Les 3 symboles # permettent d’ajouter du texte dans la boîte.  
Enfin, le code de création des boîtes est utilisé dans un chunk.

### Les jauges de la seconde ligne

```
43 Row
44 -----
45 -----
46 ### Pourcentage de femmes
47 {r}
48
49 Freq_homme <- round(100*sum(mydata$sexe=="M")/nb_patients,2) *
50
51 Freq_femme=100-Freq_homme
52 gauge(Freq_femme, min=0, max=100)
53
54
55 ### Pourcentage d'hommes
56
57 {r}
58 gauge(Freq_homme, min=0, max=100)
59
60
61
```

↑ Fig. 20

### Les plots de la troisième ligne

```
62 Row
63 -----
64 -----
65 ### Nombre de patients par pays
66
67 {r}
68 ggplot(mydata, aes(y=PAYS, fill=PAYS))+
69   geom_bar(color="black", show.legend = FALSE)+
70   facet_wrap(~sexe)+
71   xlab("Nb de patients")
72
73
74
75 ### Etude de l'Age par DP
76
77 {r}
78 ggplot(mydata, aes(y=Age.1ere.DP, x=Type.de.DP, fill=Type.de.DP,
79   colour=Type.de.DP))+
80   geom_jitter(height=0, show.legend = FALSE)+
81   geom_boxplot(alpha=0.5, colour="black", show.legend = FALSE,
82   outlier.alpha = 0)+
83   facet_wrap(~sexe)
```

↑ Fig. 21

###Les plots de la troisième ligne

```
84  
85 Row  
86 -----  
87  
88  
89 ### Distribution du Charlson modifié en fonction de DP  
90  
91 {r, fig.width=15}  
92 subset <- mydata %>%  
93   filter(Type.de.DP %in% c("DPA quotidienne", "DPCA" ))  
94  
95 ggplot(subset, aes(x=Charlson_modif, fill=Type.de.DP))+  
96   geom_density(, show.legend = FALSE)+  
97   facet_wrap(~Type.de.DP)
```

↑ Fig. 22

### Page des données

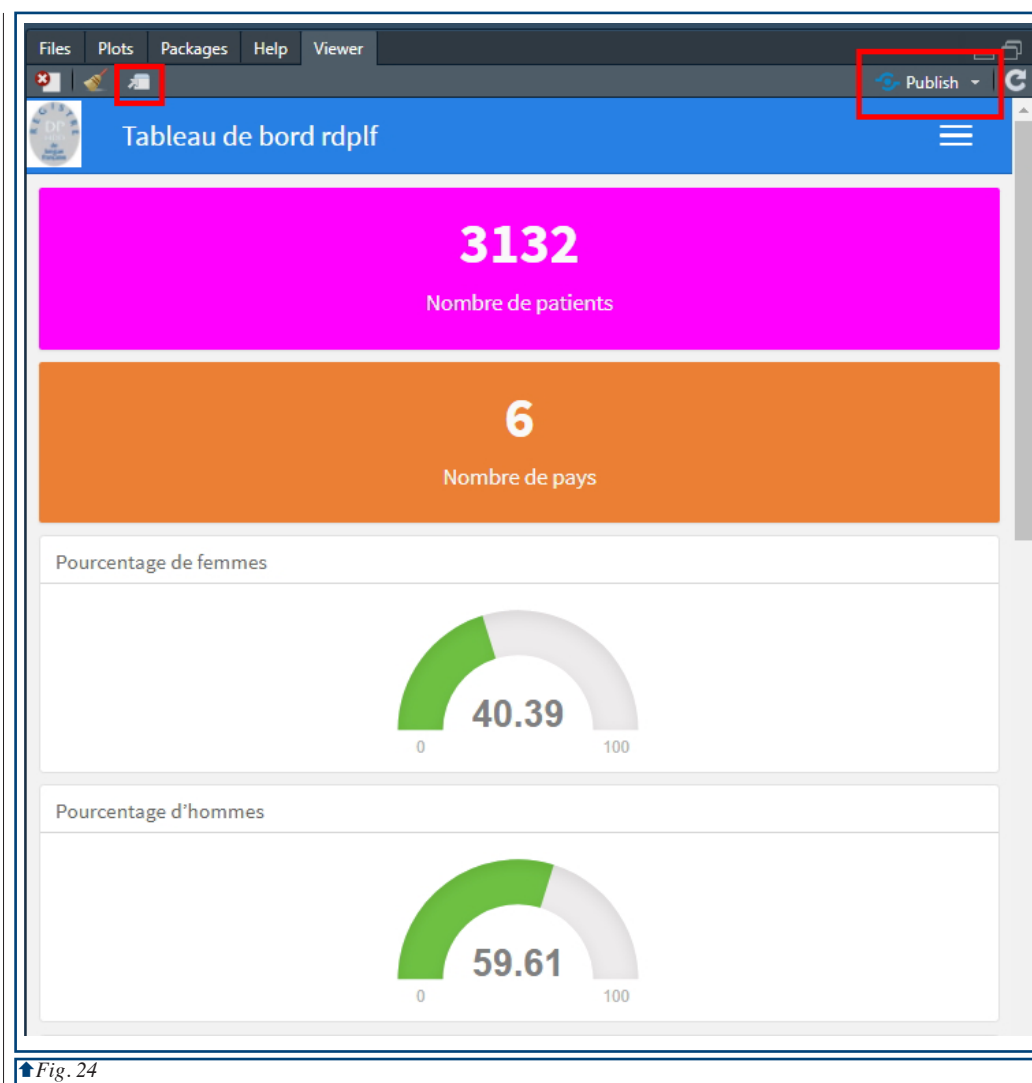
```
98  
99 Données  
100 =====  
101 {r}  
102 DT::datatable(mydata, rownames=TRUE, fillcontainer=T,  
103               options=list(pageLength=100))
```

↑ Fig. 23

Ici, un second double trait est employé afin de créer une page nommée “Données”. Vous trouverez plus d’information sur l’utilisation du package DT pour créer des table de données ici : <https://rstudio.github.io/DT/>

### Obtention du tableau de bord et déploiement

Pour visualiser votre tableau de bord dans le viewer de R Studio (un des onglets de la fenêtre en bas à droite), il suffit simplement de kniter le document. Vous pouvez également l’ouvrir dans une fenêtre de votre navigateur en cliquant sur l’icône de la fenêtre (en haut à gauche du Viewer).



↑ Fig. 24

Vous pouvez ensuite, très facilement déployer votre tableau de bord, sur le site Rpubs, par exemple, en cliquant sur le bouton publish en haut à droite du viewer.

### Conclusion

J'espère que cette initiation vous aura convaincu que la réalisation de tableaux de bord avec le logiciel est relativement simple à mettre en oeuvre, et qu'elle vous aura donné envie d'essayer! Car les tableaux de bord peuvent vraiment être une bonne solution pour présenter et partager vos analyses de données.

Et si ce sujet de la création de tableaux de bord sous R vous intéresse, vous trouverez un autre exemple de réalisation dans cet article : <https://statistique-et-logiciel-r.com/playlist-spotify-r/>

Open Access : cet article est sous licence Creative commons CC BY 4.0 : <https://creativecommons.org/licenses/by/4.0/deed.fr>

Vous êtes autorisé à :

Partager — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats

Adapter — remixer, transformer et créer à partir du matériel pour toute utilisation, y compris commerciale.

Cette licence est acceptable pour des œuvres culturelles libres.

L'Offrant ne peut retirer les autorisations concédées par la licence tant que vous appliquez les termes de cette licence. selon les conditions suivantes :

Attribution — Vous devez créditer l'Œuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'Œuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son Œuvre. <http://creativecommons.org/licenses/by/4.0/>.