Introduction à R

Graphiques

Pascal Bessonneau

06/2015

Les méthodes

Tout d'abord R fournit des fonctions permettant de produire des graphiques simples grâce à quelques fonctions de base.
Ces graphiques dits "de base" sont assez simples à manipuler et à produire.
Après des méthodes plus avancés mais demandant beaucoup plus de dextérité sont disponibles dans les packages grid notamment.
On parle de graphique "grid" ou "treillis".

Les méthodes

Les graphiques grid ne seront pas évoqués. Par contre il existe deux packages réalisant de beaux graphiques simplement utilise en fait non les graphiques de base mais le type "grid" : *lattice* et *ggplot2*. *lattice* est quelque peu passé de mode maintenant aussi il est préférable d'utiliser le paquet *ggplot2*. Sa belle esthétique est caractéristique pour un connaisseur.

Les fonctions de base

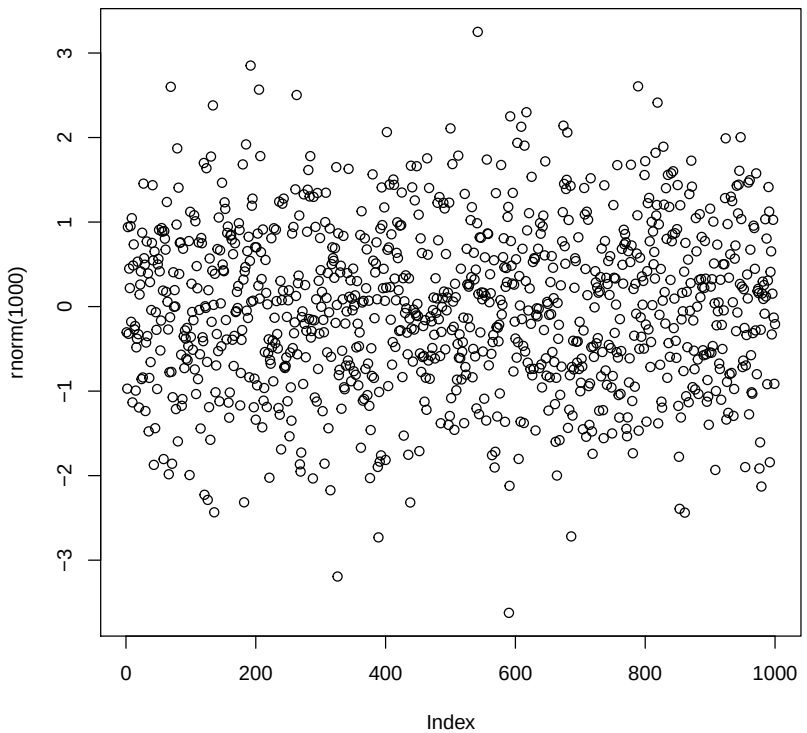
hist	pour faire un histogramme
barplot	pour faire un diagramme en barre
boxplot	pour faire des boites à moustaches
plot	pour faire des "scatterplots"
sunflowerplot	pour faire des "scatterplots" quand les points se superposent
...	

Les fonctions de base

La fonction *plot* est polymorphe. C'est une fonction générique qui accepte de nombreux types d'objets en entrée et dont le résultat varie selon les arguments qui lui sont passés.
C'est une fonction générique de R.

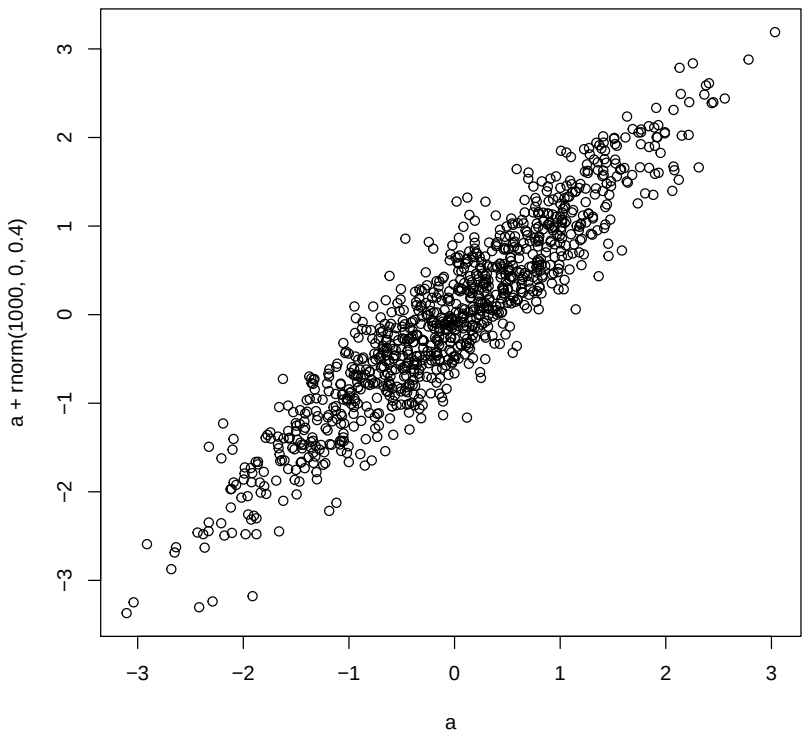
Les fonctions de base

```
> plot(rnorm(1000))
```



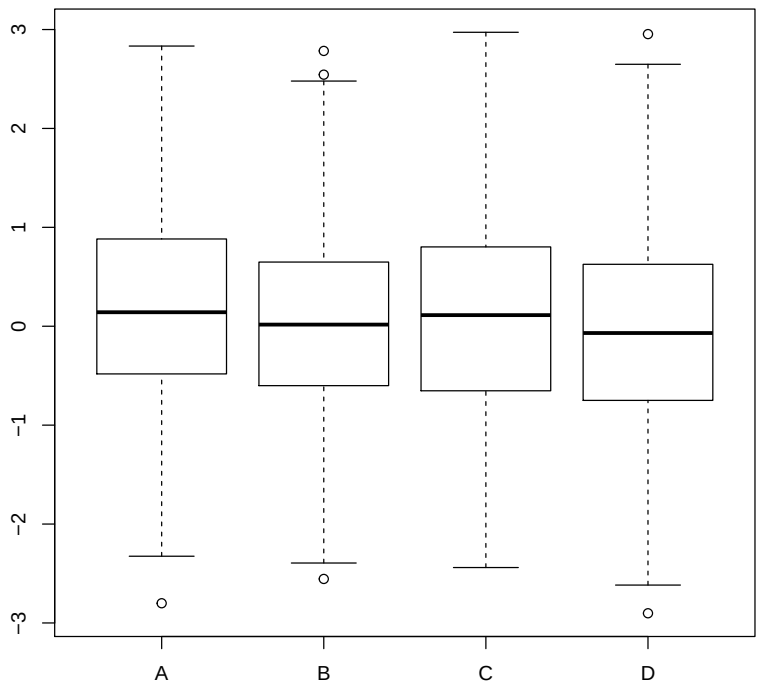
Les fonctions de base

```
> a=rnorm(1000)
> plot(a,a+rnorm(1000,0,0.4))
```



Les fonctions de base

```
> plot(factor(sample(LETTERS[1:4],1000,replace=T)),
+              rnorm(1000)
+              )
```



Les fonctions de base

De nombreux packages utilisent la fonction *plot* pour produire des graphiques en passant un argument propre au paquet.
C'est le cas par exemple pour une régression linéaire dans le paquet de base.

```
> a <- rnorm(1000)
> dt <- data.frame(
+   a=a,
+   b=a+rnorm(1000,0,0.4)
+ )
> rl <- lm(b ~ a)
> plot(rl,ask=F)
```

Les fonctions de base

```
## Error in eval(expr, envir, enclos): objet 'b' introuvable
```

```
> class(rl)

## Error in eval(expr, envir, enclos): objet 'rl' introuvable
```

La fonction *plot* est une fonction générique. En fait la fonction spécialisé pour les objets de type *lm* est appelé en lieu et place de la fonction usuelle.

Les fonctions de base

```
> methods(class=class(rl))

## Error in methods(class = class(rl)): objet 'rl' introuvable
```

Cette commande est très pratique pour connaître les fonctions implémentées pour ce type d'objet.

Les fonctions de base

C'est en partie la raison pour laquelle on dit que R est un langage objet car c'est un langage dont certaines fonctions sont polymorphiques.
En fait on parle dans ce cas de modèle S3. Le modèle S4 qui est plus proche d'un vrai langage objet est peu utilisé pour l'instant.

Les fonctions de base

Dans le cas des fonctions graphiques de base, les coordonnées sont calculés lors du première appel à la fonction.
Ensuite on peut rajouter des points ou des surfaces et les coordonnées sont exprimées dans les mêmes unités que celles des données.
Ce n'est pas le cas pour le paquet grid ce qui rend les manipulations plus complexes.

Les arguments les plus fréquents

Les fonctions de base accepte des arguments par défaut très souvent les mêmes :

- **xlim** un vecteur contenant le minimum et le maximum pour l'axe des abscisses
- **ylim** un vecteur contenant le minimum et le maximum pour l'axe des ordonnées
- **main** le titre du graphique
- **xlab** le nom de l'axe des abscisses
- **ylab** le nom de l'axe des ordonnées
- **col** la couleur des poiunts ou des surfaces

Les arguments les plus fréquents

Le nombre de paramètres graphiques est impressionnant et il varie malheureusement un peu selon les fonctions.
on peut les visualiser avec la commande *par()*.

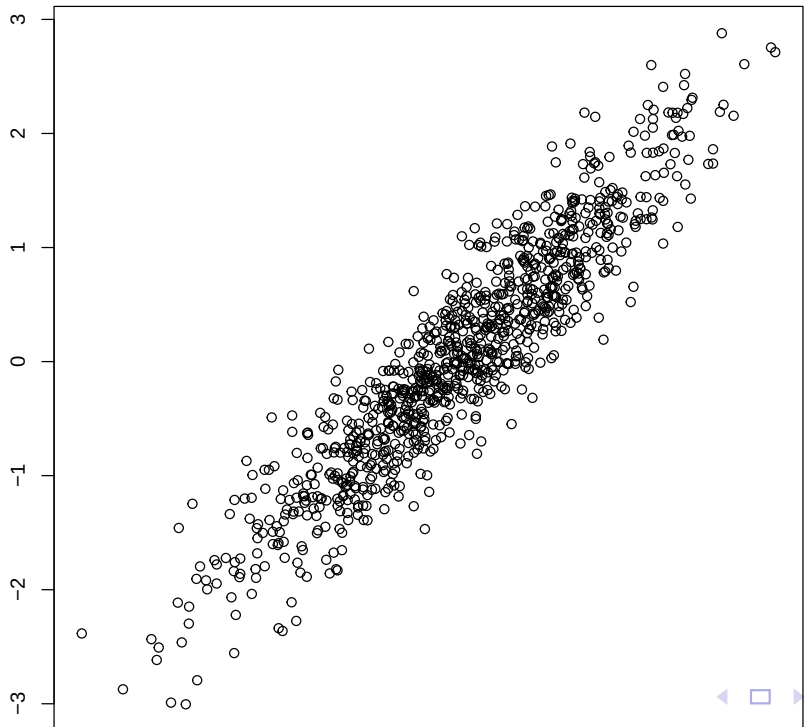
Les arguments les plus fréquents

Par exemple on peut réduire les marges :

```
> par()$mar

## [1] 5.1 4.1 4.1 2.1

> par(mar=c(3.1,2.1,2.1,2.1))
> plot(dt$b,dt$a)
```



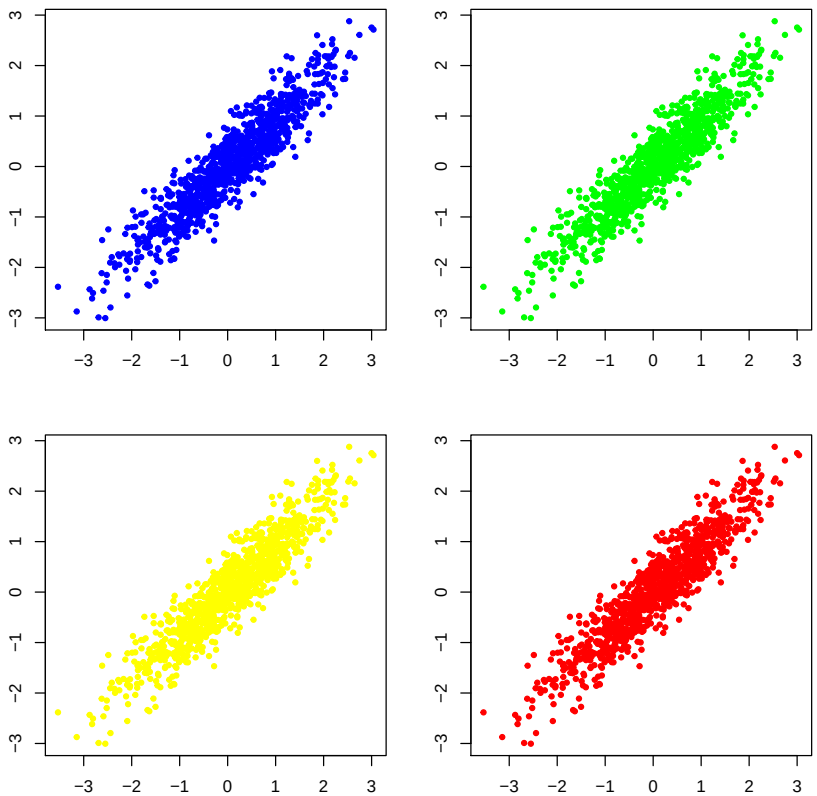
Les arguments les plus fréquents

On peut également faire plusieurs graphiques sur la même page :

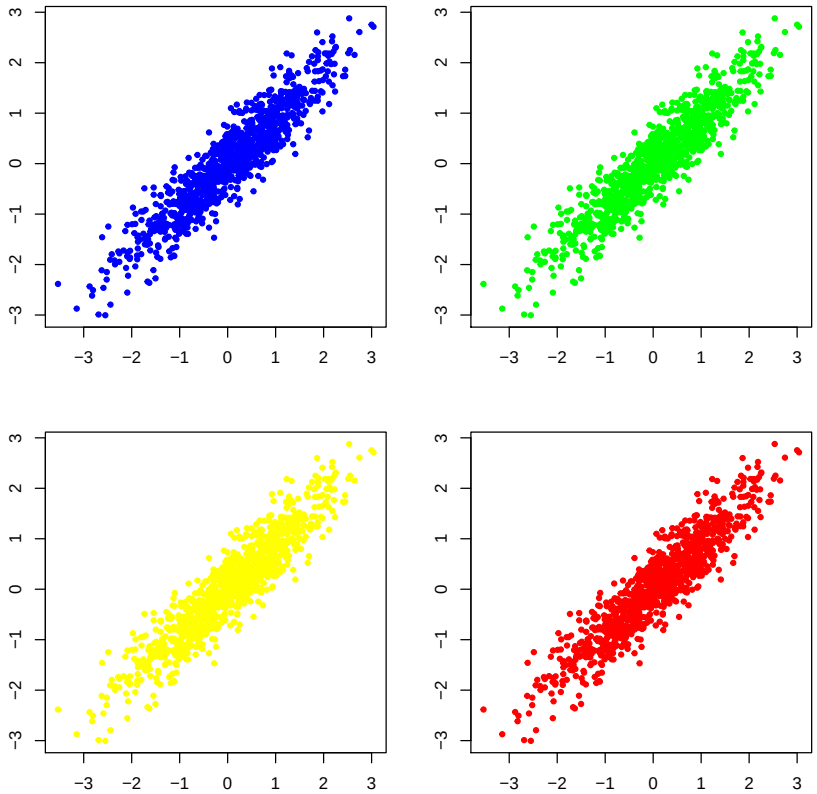
```
> par(mfrow=c(2,2))
> par(mar=c(3.1,2.1,2.1,2.1))
> plot(dt$b,dt$a,col="blue",pch=20)
> plot(dt$b,dt$a,col="green",pch=20)
> plot(dt$b,dt$a,col="yellow",pch=20)
> plot(dt$b,dt$a,col="red",pch=20)
```

Les arguments les plus fréquents

On peut également faire plusieurs graphiques sur la même page :



Les arguments les plus fréquents



Les fonctions de superposition

Il faut distinguer deux types de fonctions :

- les fonctions d'initialisation et de tracé
- les fonctions de superpositions sur un tracé

Un graphique R de base n'existe pas comme dit plus sans une échelle des X et une échelle des Y. Les fonctions comme *plot*, *barplot*, *hist*, ... initialise le graphique et font tout ce qu'il faut pour tracer un graphique.

Les fonctions de superposition

Les fonctions de superposition permettent de réaliser des tracés mais en utilisant sur des graphiques existants.
Soit il s'agit de fonctions distinctes :

- *lines*
- *points*
- *text*
- ...

Les fonctions de superposition

Les fonctions de superposition permettent de réaliser des tracés mais en utilisant sur des graphiques existants.
Soit il s'agit de fonctions d'initialisation mais avec un argument spécifique : généralement il s'agit de rajouter l'argument *add* :

```
> plot(x,y,add=T)
```

La notion de *device*

La sortie par défaut sur R est une fenêtre graphique. Par exemple dans RStudio l'onglet plot en bas à droite.
Mais on peut créer des graphiques dans des devices différents tels que des fichiers. Par exemple pour créer un fichier jpeg, il faut ouvrir un device *jpeg* qui va se substituer à la fenêtre graphique. Puis on va fermer le device pour finaliser l'export.

La notion de *device*

```
> jpeg("graphiques/MonGraphique.jpeg")
> par(mfrow=c(2,2))
> par(mar=c(3.1,2.1,2.1,2.1))
> plot(dt$b,dt$a,col="blue",pch=20)
> plot(dt$b,dt$a,col="green",pch=20)
> plot(dt$b,dt$a,col="yellow",pch=20)
> plot(dt$b,dt$a,col="red",pch=20)
> dev.off()

## pdf
##   2
```

La notion de *device*

Les formats sont nombreux :

- png
- pdf
- svg
- ...

Chaque device porte le nom du type de fichier qu'il va créé.

La procédure avec les *device*

La procédure se fait avec les étapes suivantes :

1. vous créez le device avec par exemple *png("file")*
2. vous dessinez
3. vous refermez le device avec la commande *dev.off()*

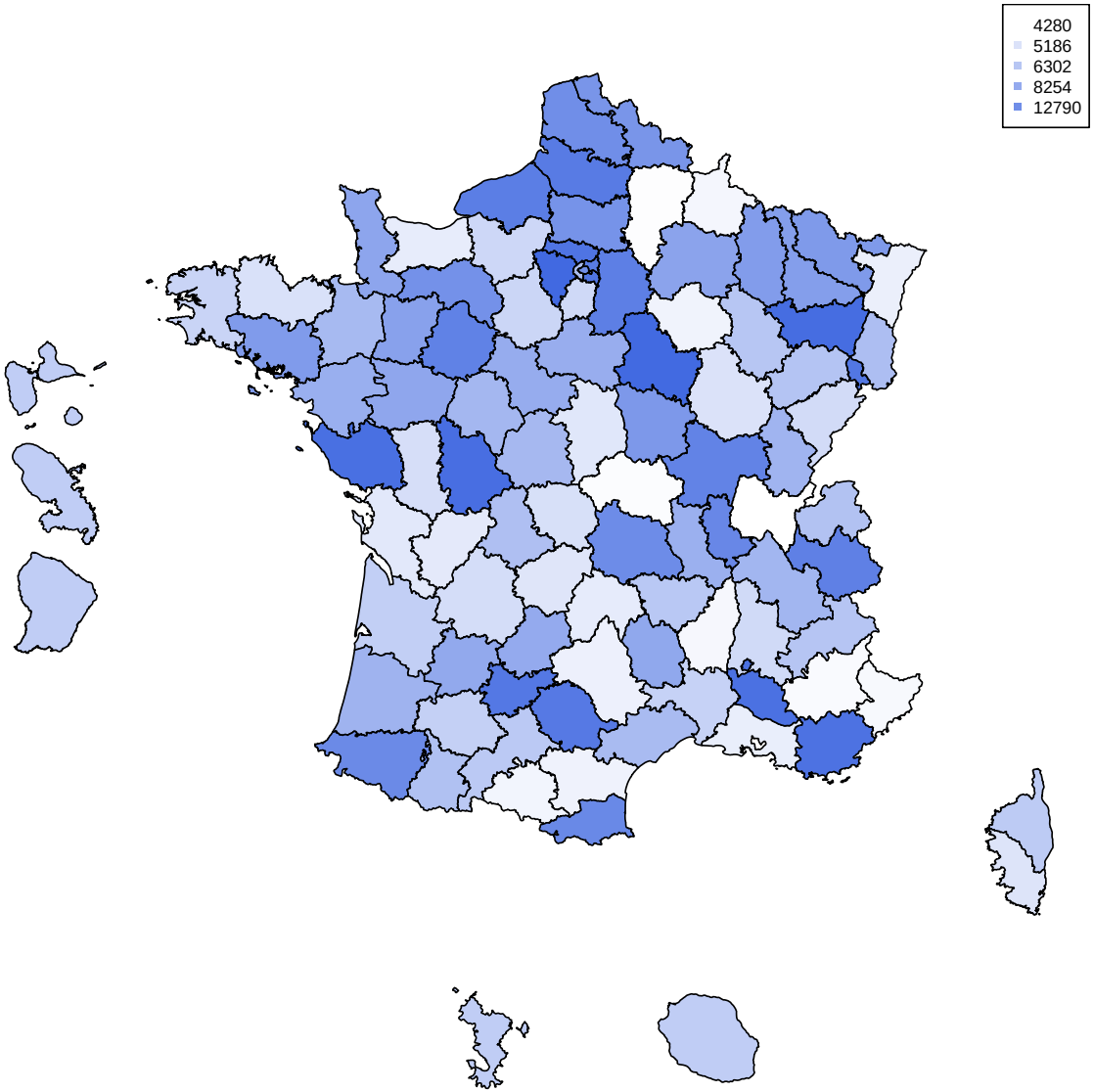
Quelque devices comme le pdf permet de produire plusieurs graphiques sur plusieurs pages comme par exemple le format PDF.

Les arguments des *device*

Les arguments varient, il suffit de regarder l'aide. Souvent on trouve l'argument dpi qui permet de donner la résolution du graphique.
Les arguments width et height donnent la taille dont l'unité dépend du device choisi.

Les options d'agencement avancée

L'option *layout* permet de définir des agencements avancés comme sur le graphique ci-dessous.



Les options d'agencement avancée

Il suffit de donner en entrée une matrice avec un numéro dans l'ordre de ce qui va être dessiner. Les fusions des cellules de la matrice donnent la taille de chaque zone.
Dans l'exemple, on représente un grand graphique central pour la France métropolitaine puis des petits carrés pour les DOMs.

Les options d'agencement avancée

Il suffit de donner en entrée une matrice avec un numéro dans l'ordre de ce qui va être dessiner. Les fusions des cellules de la matrice donnent la taille de chaque zone.
Dans l'exemple, on représente un grand graphique central pour la France métropolitaine puis des petits carrés pour les DOMs.

Les options d'agencement avancée

```
> xmetro <- 9
> xdom <- 1
> mm <- matrix(
+     c(
+       rep( 7,xdom), rep( 1,xmetro),
+       rep( 7,xdom), rep( 1,xmetro),
+       rep( 7,xdom), rep( 1,xmetro),
+       rep( 2,xdom), rep( 1,xmetro),
+       rep( 3,xdom), rep( 1,xmetro),
+       rep( 4,xdom), rep( 1,xmetro),
+       rep( 0,xdom), rep( 1,xmetro),
+       rep( 0,xdom), rep( 1,xmetro),
+       rep( 0,xdom), rep( 1,xmetro),
+       rep( 0,xdom), rep(0,3), rep(5,1), rep(0,1), rep(6,1), rep(0,xme
+     ),
+     ncol=10,
+     nrow=10,
+     byrow=T
+ )
> mm
```

Les options d'agencement avancée

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7]
##  [1,]    7    1    1    1    1    1    1
##  [2,]    7    1    1    1    1    1    1
##  [3,]    7    1    1    1    1    1    1
##  [4,]    2    1    1    1    1    1    1
##  [5,]    3    1    1    1    1    1    1
##  [6,]    4    1    1    1    1    1    1
##  [7,]    0    1    1    1    1    1    1
##  [8,]    0    1    1    1    1    1    1
##  [9,]    0    1    1    1    1    1    1
## [10,]    0    0    0    0    5    0    6
##       [,8] [,9] [,10]
##  [1,]    1    1     1
##  [2,]    1    1     1
##  [3,]    1    1     1
##  [4,]    1    1     1
##  [5,]    1    1     1
##  [6,]    1    1     1
##  [7,]    1    1     1
##  [8,]    1    1     1
##  [9,]    1    1     1
```