



# Gestion des couleurs sous R

---

**Michel BAYLAC**

*MNHN*

*Département Systématique et Evolution*

*OSEB*

*baylac@mnhn.fr*

## Gestion des couleurs sous R

**Michel Baylac**

**MNHN DSE, UMR 5202**

**Plate-forme Morphométrie, MNHN-CNRS IFR 101**

**[baylac@mnhn.fr](mailto:baylac@mnhn.fr)**

# Gestion des couleurs sous R

Un exemple : `plot(rep(1,20),pch=20, col=1:20, cex=8)`

quelques remarques préliminaires :

- `rep(1,20)` : spécifie un vecteur unitaire de longueur 20 :  

```
> rep(1,20)
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```
- `plot` avec un seul paramètre (`rep(1,20)`) :
  - une des (nombreuses) formes possibles d'appel (voir `?plot`) :
    - `rep(1,20)` définit dans ce cas les ordonnées
    - abcisses automatiquement égales à l'index 1:20
- `pch=20` spécifie le type de symboles pour les points, ici ronds pleins
- `cex= 8` spécifie la taille des symboles (= facteur multiplicateur), et
- `col` spécifie la couleurs des points

# Gestion des couleurs sous R

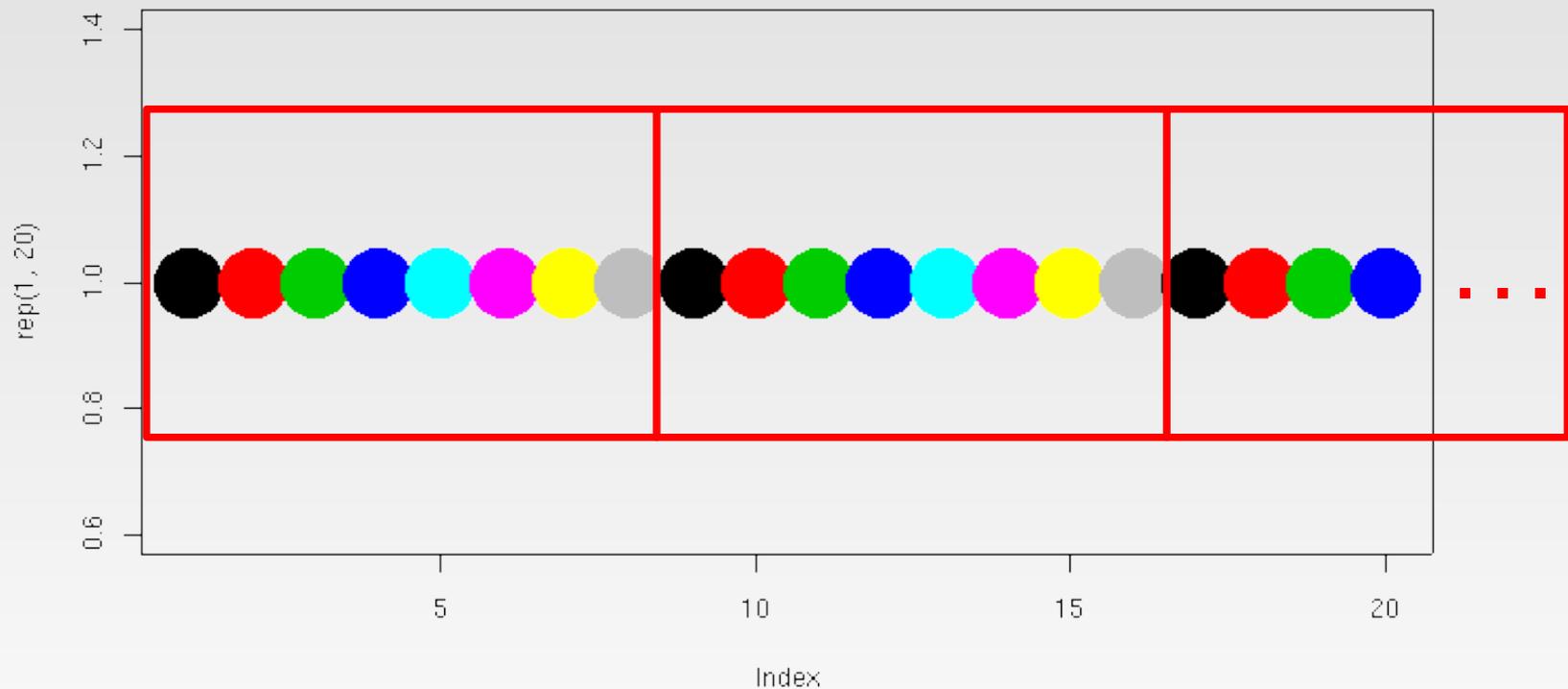
**pch**, **cex** et **col** sont des paramètres graphiques **modifiables** :

- **temporairement** comme ici c'est-à-dire applicables au graphe qui résultera de l'appel à **plot()**

- **pour plusieurs graphes** en altérant les valeurs des paramètres graphiques par défaut au moyen de l'instruction **par()**

# Gestion des couleurs sous R

```
plot(rep(1,20),pch=20, col=1:20, cex=8)
```



**Recyclage des couleurs :**  
**seulement 8 couleurs disponibles dans R ?**

# Gestion des couleurs sous R

**R comprend un ensemble de 657 couleurs prédéfinies :**

**> colors()**

```
[1] "white"           "aliceblue"       "antiquewhite"  
[4] "antiquewhite1"  "antiquewhite2"  "antiquewhite3"  
[7] "antiquewhite4"  "aquamarine"     "aquamarine1"  
[10] "aquamarine2"   "aquamarine3"   "aquamarine4"  
[13] "azure"          "azure1"         "azure2"  
[16] "azure3"         "azure4"         "beige"  
[19] "bisque"        "bisque1"        "bisque2"  
...  
[652] "yellow"         "yellow1"        "yellow2"  
[655] "yellow3"       « yellow4"       "yellowgreen"
```

**Les couleurs utilisées par défaut correspondent à un sous-ensemble (limité) des couleurs utilisables**

# Gestion des couleurs sous R

Le manuel R rappelle :

**col**=Colors to be used for points, lines, text, filled regions and images.

**col** s'applique donc à un ensemble d'éléments graphiques, mais pas à tous les éléments graphiques :

**col.axis, col.lab, col.main, col.sub** = The color to be used for axis annotation, x and y labels, main and sub-titles, respectively.

# Gestion des couleurs sous R

Toujours dans le manuel :

**col = a number from the current palette (see ?palette) or a named colour,**

**introduit le point essentiel qui nous occupera ici : la notion d'ensembles de couleurs nommés ou palettes**

**l'instruction `plot(rep(1,20),pch=20, col=1:20, cex=8)`**

**ne spécifie pas de palette particulière et utilise celle par défaut  
le contenu de la palette active est affiché par un appel à `palette()` :**

**`palette()`**

**`[1] "black" "red" "green3" "blue" "cyan" "magenta" "yellow"`**

**`[8] "gray"`**

# Gestion des couleurs sous R

Et dans l'aide de palette :

**help(palette) :**

**... View or manipulate the color palette which is used when a col= has a numeric index ...**

# Gestion des couleurs sous R

**colors()** n'est pas une palette. C'est un ensemble de couleurs prédéfinies  
**colors()** est une fonction qui renvoie un vecteur de type 'character'

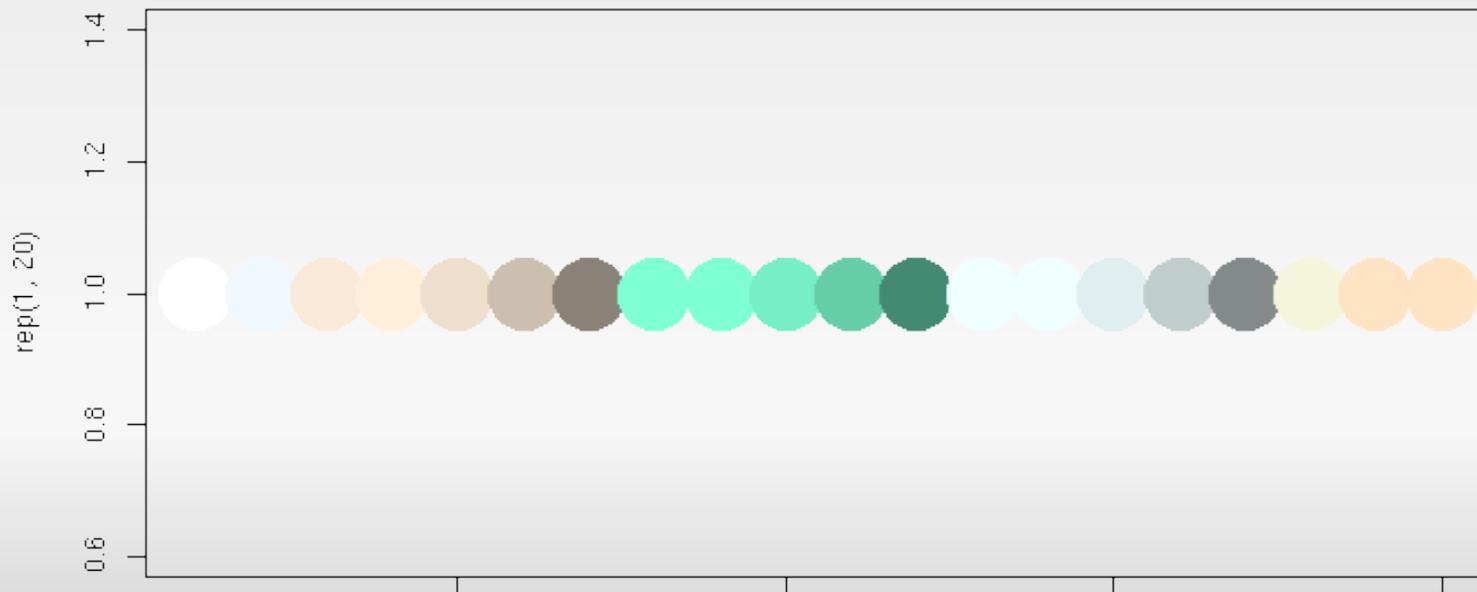
1. utilisable dans un graphe :

```
cc <- colors()
```

```
cc <- cc[1:20] # ou plus simplement
```

```
dd <- colors()[1:20]
```

```
plot(rep(1,20),pch=20, col=dd, cex=8) # dd utilisé comme vecteur de couleurs
```



# Gestion des couleurs sous R

## 2. pour créer une nouvelle palette :

**palette(dd)** # la palette active par défaut devient dd

**palette()**

```
[1] "white"      "aliceblue"  "antiquewhite" "antiquewhite1"  
[5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"  
[9] "aquamarine"  "aquamarine2" "aquamarine3"  "aquamarine4"  
[13] "azure"      "azure"      "azure2"       "azure3"  
[17] "azure4"     "beige"      "bisque"       "bisque"
```

**plot(rep(1,20),pch=20, col=1:20, cex=8)** # dd utilisé comme palette

**palette("default")** # rétablit la palette par défaut sous R

# Représentations des couleurs sous R

Il existe différentes palettes prédéfinies, en couleurs ou en gris et noir et blanc

**hsv()**

**gray() # ou grey()**

**rainbow()**

**heat.colors()**

**terrain.colors()**

**topo.colors()**

**cm.colors()**

Voir les aides pour leur utilisation

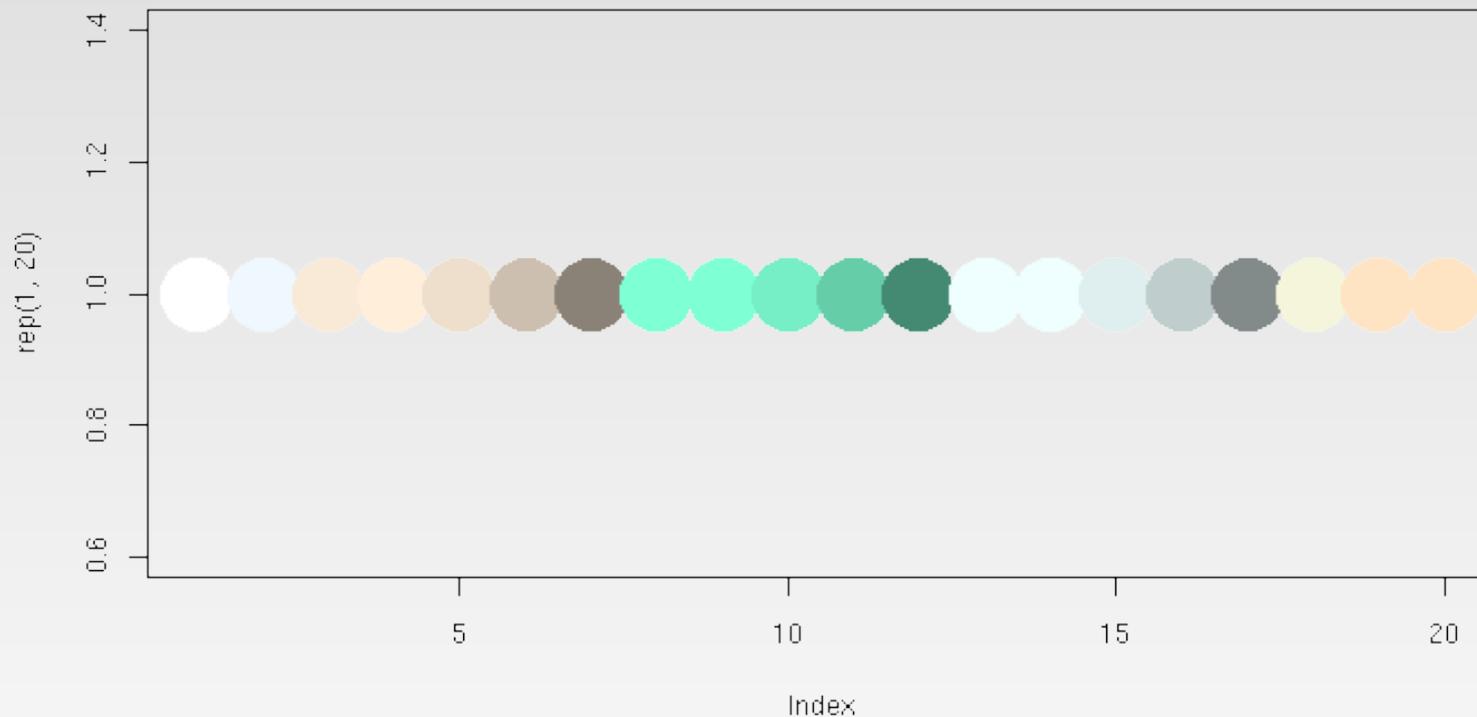
exemple en noir et blanc :

**palette(grey(0:10/10)) # définit une palette à 10 niveaux de gris, en fait de noir (0) à blanc (1) : vecteur de 0 à 1 par pas de 0.1**

**x11() ; plot(rep(1,20),pch=20, col=1:20, cex=8) :# lisibilité ?**

# Gestion des couleurs sous R

**Les exemples précédents montrent à l'évidence que créer une palette utilisable n'est pas simple !**



**il existe une bibliothèque permettant d'utiliser des palettes adaptées à différents types de graphes :**

**RColorBrewer**

# Gestion des couleurs sous R

## **RcolorBrewer**

**Creates nice looking color palettes especially for thematic maps**

**Author(s) : Erich Neuwirth, University of Vienna et John Maindonald, Australian National University**

**ColorBrewer is Copyright (c) 2002 Cynthia Brewer, Mark Harrower, and The Pennsylvania State University. All rights reserved.**

**The ColorBrewer palettes have been included in this R package with permission of the copyright holder.**

# Gestion des couleurs sous R

**RcolorBrewer définit 3 types de palette :**

1. **Sequential palettes** are suited to ordered data that progress from low to high.
2. **Diverging palettes** put equal emphasis on mid-range critical values and extremes at both ends of the data range.
3. **Qualitative palettes** do not imply magnitude differences between legend classes, and hues are used to create the primary visual differences between classes. Qualitative schemes are best suited to representing nominal or categorical data.

# Gestion des couleurs sous R

## Palettes séquentielles

9 palettes (avec chacune de 3 à 9 valeurs) : **Blues BuGn BuPu GnBu Greens Greys Oranges OrRd PuBu PuBuGn PuRd Purples RdPu Reds YIGn YIGnBu YIOrBr YIOrRd**

## Palettes divergentes

9 palettes (avec chacune de 3 à 11 valeurs) : **BrBG PiYG PRGn PuOr RdBu RdGy RdYIBu RdYIGn Spectral**

## Palettes qualitatives

8 palettes avec au minimum 3 valeurs :

<b>Accent</b>	<b>8</b>
<b>Dark2</b>	<b>8</b>
<b>Paired</b>	<b>12</b>
<b>Pastel1</b>	<b>9</b>
<b>Pastel2</b>	<b>8</b>
<b>Set1</b>	<b>9</b>
<b>Set2</b>	<b>8</b>
<b>Set3</b>	<b>12</b>

# Gestion des couleurs sous R

## Exemples

```
library(RColorBrewer)
```

```
display.brewer.all(type="div") # type divergent
```

```
display.brewer.all(type="seq") # type séquentiel
```

```
display.brewer.all(type="qual") # type qualitatif
```

# Représentations des couleurs sous R

Un (presque) dernier exemple :

```
data(iris)
```

```
palette(brewer.pal(3,"Accent"))
```

```
palette()
```

**[1] "#7FC97F" "#BEAED4" "#FDC086" : il reste à voir comment sont définies et codées les couleurs dans R : ce sera pour une prochaine séance !**

```
pairs(iris[,1:4], col=as.numeric(iris$Species), pch=20,cex=2)
```

# Merci de votre attention



`display.brewer.all()`