

A photograph of a modern building facade. The left side shows a white brick wall, and the right side features a large glass facade reflecting the sky. A semi-transparent white box is overlaid on the center of the image, containing the title and author's name. In the bottom right corner, there is a logo for 'World Pastors' and the letters 'SICS' in gold.

Les couleurs dans R

Vincent Guillemot

Avant toutes choses

Nous aurons besoin du package `RColorBrewer` :

- Vérifier que le package `RColorBrewer` est bien installé
- Si non, l'installer, puis le charger

```
library(RColorBrewer)
```

Nous allons également avoir besoin des données `fruits` :

```
fruits <- readxl::read_excel("fruits.xlsx")
```

Trois façons de colorier des objets

Trois façons de voir les couleurs

Voici trois manières de définir une couleur, qui permettent de colorier un graphe soit très rapidement (entiers), soit avec plus de possibilités (couleurs nommées) ou encore de manière très raffinée (HEX).

```
#>
#> Attaching package: 'kableExtra'
#> The following object is masked from 'package:dplyr':
#>
#>   group_rows
```

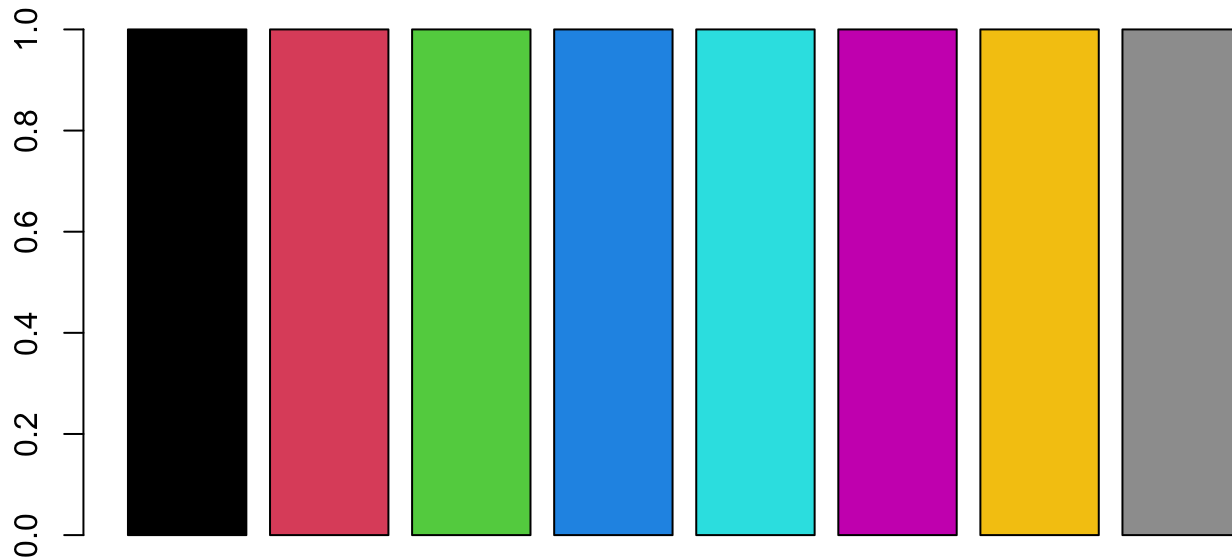
Entier
Nom
HEX
Entier
Nom
HEX

1	2	3	4
black	indianred2	palegreen3	dodgerblue2
#000000	#DF536B	#61D04F	#2297E6
5	6	7	8
turquoise	magenta3	darkgoldenrod1	gray62
#28E2E5	#CD0BBC	#F5C710	#9E9E9E

Les chiffres (rapide !)

La palette de 8 couleurs par défaut de R est codée par les entiers de 1 à 8.

```
barplot(rep(1,8), col = 1:8)
```



Les noms de couleurs (plus de couleurs)

On peut également colorier avec des “noms” de couleur (e.g. "black", "tomato", "steelblue", "darkorchid" etc.)

On peut accéder à tous ces noms de couleur avec la commande `colors()` :

```
sample(colors(), 7)
#> [1] "tomato"          "slategray1"      "lavenderblush4" "darkgoldenrod4"
#> [5] "gray6"           "darkorchid1"    "snow4"
```

Ces couleurs “nommées” s'utilisent de la même façon que les couleurs “numériques”.

HEX (encore plus de couleurs !)

On peut aussi utiliser des codes hexadécimaux pour coder une couleur dans le système de référence "Rouge - Vert - Bleu" :

- deux symboles hexadécimaux par couleur ($16 \times 16 = 256$ valeurs possible)
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Trois couleurs de base : rouge, vert et bleu
- 00 = pas de cette couleur
- FF = max de cette couleur



Exemple

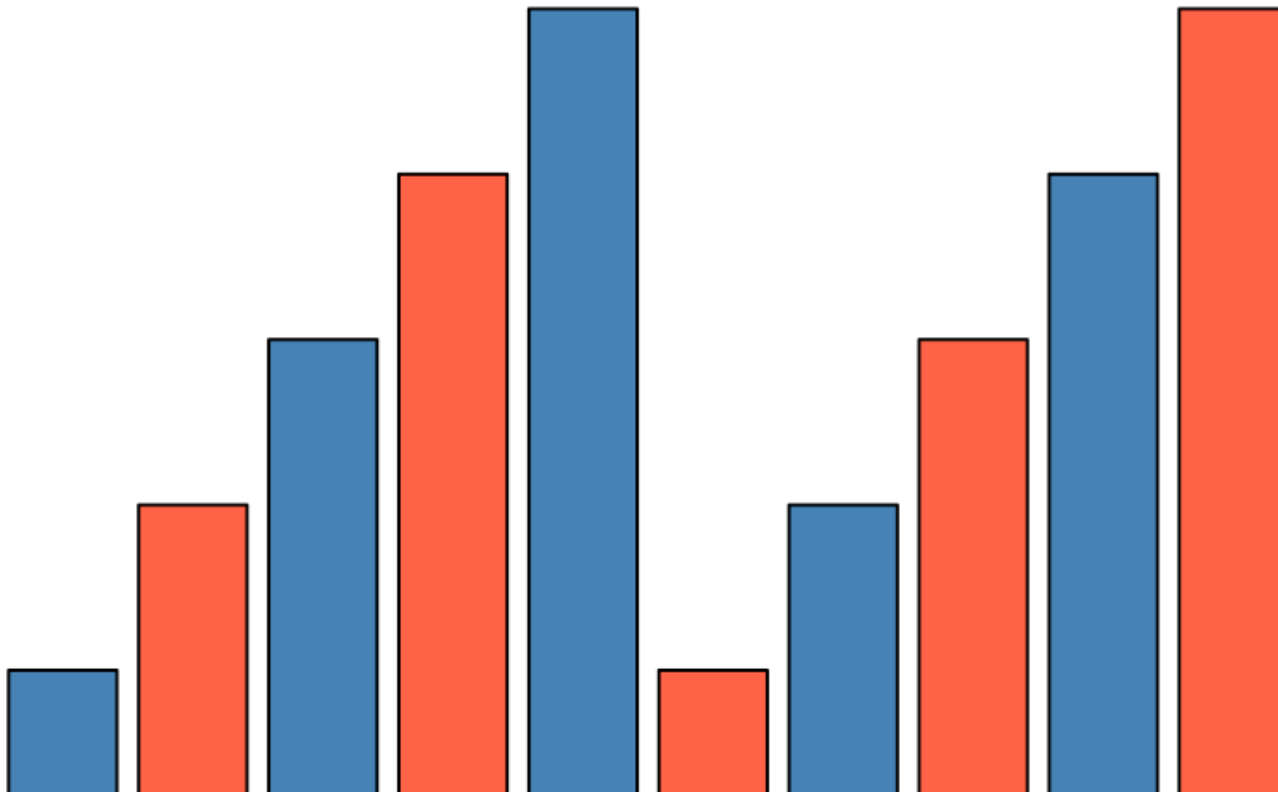
Voici trois commandes équivalentes pour réaliser le même graphe ci-dessous :

```
barplot(rep(1, 3), col = 2:4)
barplot(rep(1, 3), col = c("indianred2", "palegreen3", "dodgerblue2"))
barplot(rep(1, 3), col = c("#DF536B", "#61D04F", "#2297E6"))
```



A vous !

Reproduisez le graphe ci-dessous avec le système de couleurs de votre choix :



ATTENTION !

Attention au recyclage !

S'il y a plus d'objets à colorier que de couleurs, les couleurs sont recyclées !

```
par(mar = c(0, 0, 0, 0))  
barplot(rep(1,80), col = 1:8, border = NA, space = 0, axes = FALSE)
```



Les palettes

Utiliser des palettes

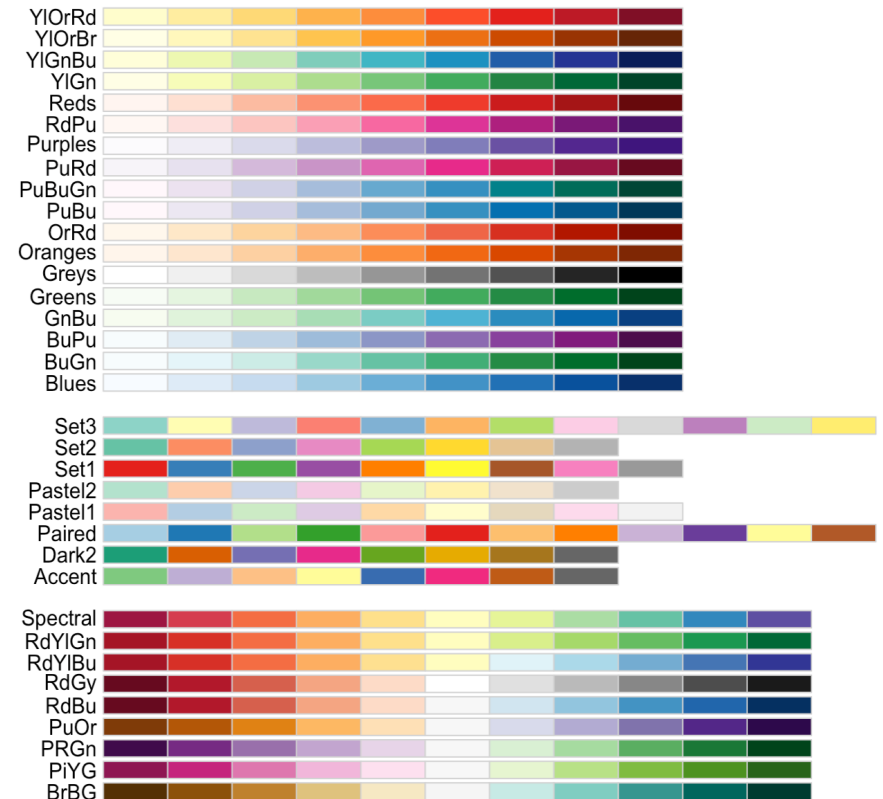
Il existe de nombreux packages en R permettant de générer des palettes de couleur. Nous allons en utiliser un seul pendant ce cours : RColorBrewer.

La commande suivante permet de visualiser toutes les palettes que ce package permet d'utiliser :

```
display.brewer.all()
```

Pour extraire des couleurs de ces palettes :

```
brewer.pal(n = 3, name = "Set3")  
#> [1] "#8DD3C7" "#FFFFB3" "#BEBADA"
```



En quelques mots

Il y a trois types de palettes : séquentielles, divergentes et qualitatives.

1. Les palettes séquentielles permettent de distinguer des valeurs petites (en clair) de valeurs grande (en foncé)
2. Les palettes divergentes permettent de distinguer les valeurs petites des valeurs grandes, mais ces deux extrêmes sont toutes les deux de ton foncé dans des couleurs très différentes. Les valeurs intermédiaires sont représentées en ton clair.
3. Les palettes qualitatives sont faites pour distinguer toutes les couleurs les unes des autres. Elle sont adaptées à la représentation de données qualitatives.

A vous

Complétez le code suivant pour obtenir le graphe ci-contre :

```
pal <- brewer.pal(***, ***)  
barplot(rep(1, 7),  
        col = pal,  
        axes = ***,  
        border = ***)
```

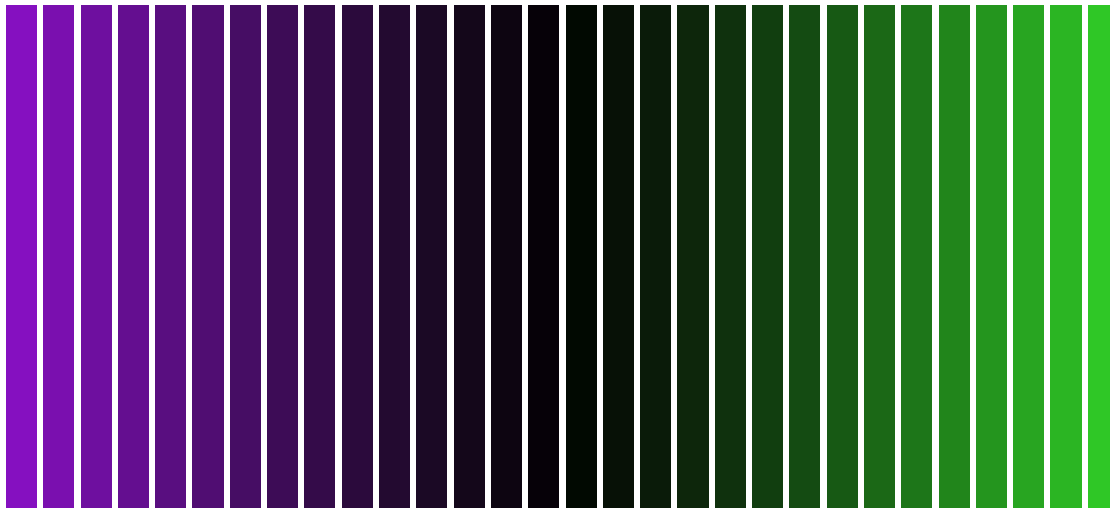


Extrapoler des couleurs

Pour extrapoler des couleurs...

On utilise la fonction de base `grDevices::colorRampPalette` :

```
colfun <- colorRampPalette(c("darkorchid", "black", "limegreen"))  
barplot(rep(1, 30), col = colfun(30), axes = F, border = NA)
```



Exercice

Reproduisez le graphe ci-contre en modifiant la commande ci-dessous.

```
colfun <- colorRampPalette(  
  c(***, ***, ***)  
)  
barplot(rep(1, 100),  
  col = colfun(100),  
  axes = F, border = NA)
```

