

A photograph of a modern building with a glass facade and a courtyard with greenery. The building has a white brick wall on the left and a glass facade on the right. The courtyard is filled with various green plants and a paved path. The sky is blue with a few clouds.

# Pretty Heatmaps

Vincent Guillemot

Nov. 25

swiss post

ONICS

# First things first

We will need the packages `RColorBrewer` and `pheatmap`:

- Check that `RColorBrewer` and `pheatmap` are installed
- If not, install them, then load them

```
library(RColorBrewer)
library(pheatmap)
```

We also need the “fruits” data:

```
data("fruits", package = "debuter")
```

# Program

- Colors, with or without `RColorBrewer`
- Heatmaps with `pheatmap`

Three ways to color objects

# Three ways to see colors

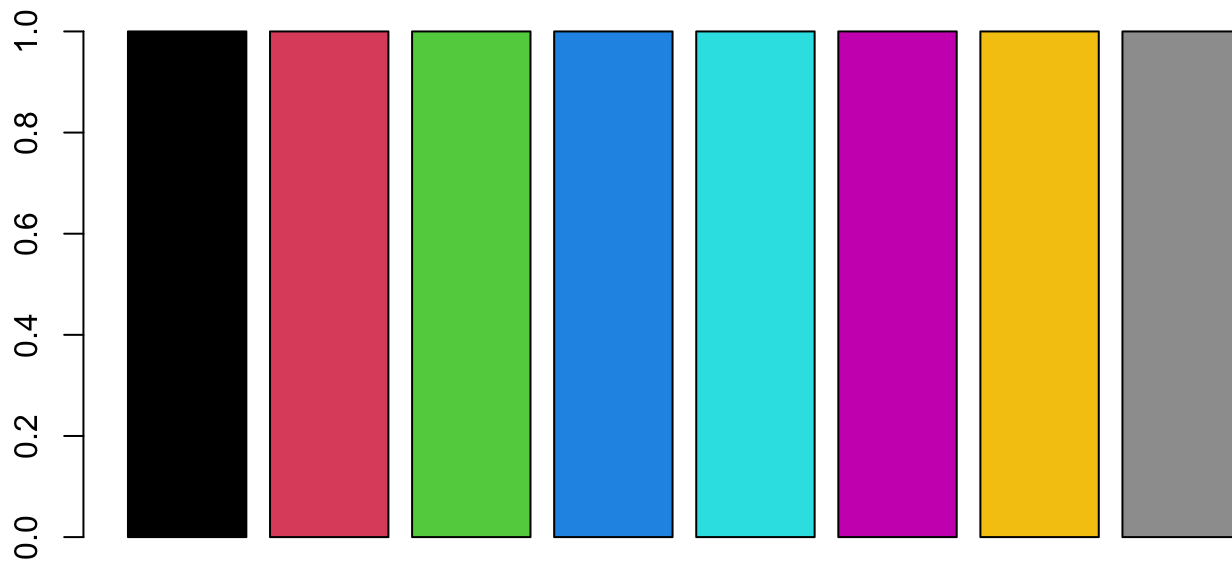
With integers, with color names or with an hexadecimal code (HEX)

Integer Name HEX Integer Name 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

# Integers (rapide !)

The 8-color default palette in R.

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



# Color names (more colors)

You can use color names (e.g. "black", "tomato", "steelblue", "darkorchid" etc.)

One can access these names with the command `colors()` :

```
sample(colors(), 7)
#> [1] "deepskyblue1"      "bisque2"           "orange1"
#> [4] "lightgoldenrod4"   "orchid"             "orangered4"
#> [7] "lightblue1"
```

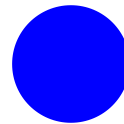
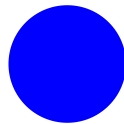
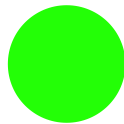
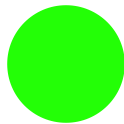
“Names” colors are used the same way as “integer” colors.

# HEX (even more colors !)

... with an hexadecimal code in the “Red - Green - Blue” color space:

- deux hexadecimals per base color ( $16 \times 16 = 256$ ) possible values)
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- Three base colors: red, green et blue
- 00 = none of this color
- FF = max of this color

#





# Example

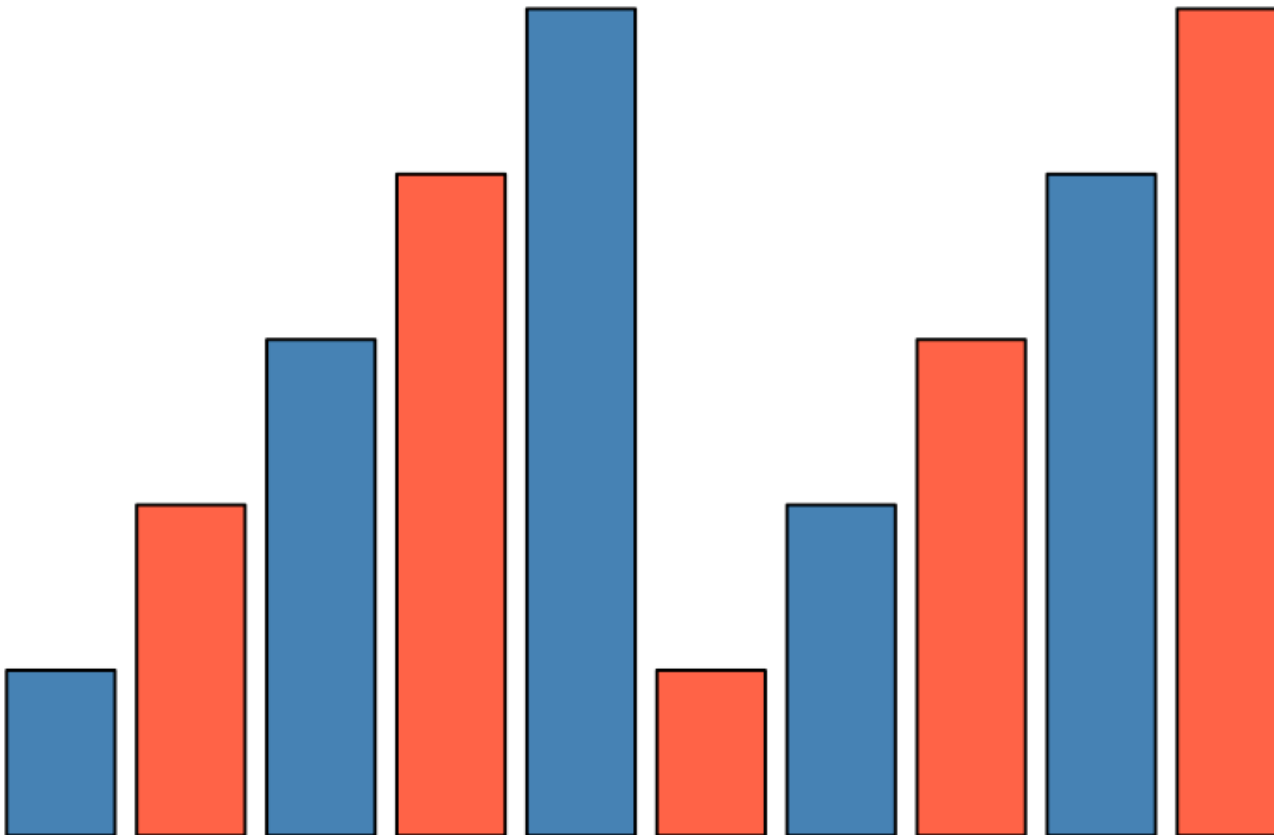
Three (almost) equivalent ways to obtain the following graph:

```
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"))
```



# Your turn!

Reproduce the graph below:



**ATTENTION !**

# Recycling is dangerous!

If there are more objects than colors, the colors will be recycled!

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



Palettes

# Use palettes

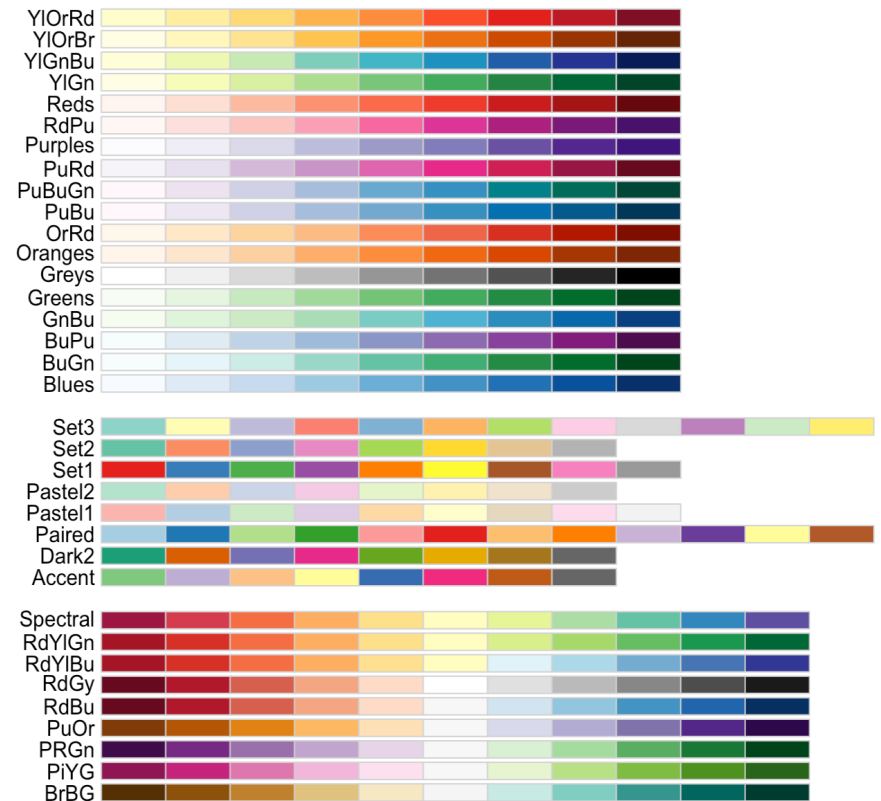
We will see **one** package in R that contains palettes: `RColorBrewer`.

The following command will display all the available palettes in `RColorBrewer`:

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

To get the colors from a specific palette:

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



# In a few words

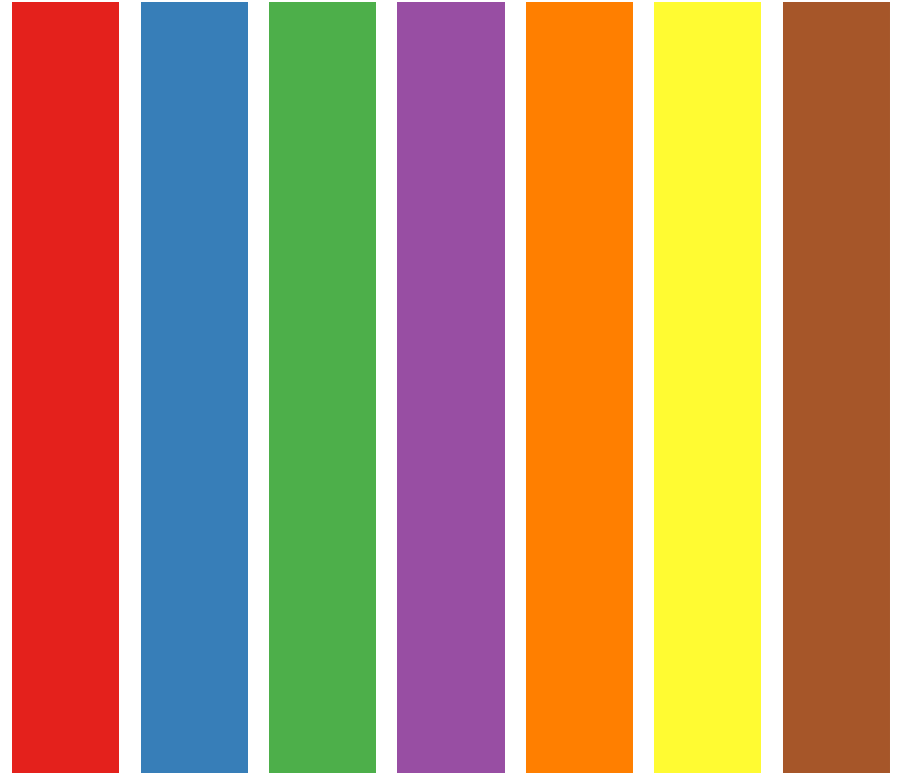
There are three types of palette : sequential, diverging and qualitative.

1. **sequential** : to distinguish low values (light colors) to high values (dark colors)
2. **diverging** : low and high values are both dark colors, middle values are light colors
3. **qualitative**: adapted to coloring qualitative data

# Your turn

Complete the code to obtain the graph on the right:

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





Heatmaps with `pheatmap`

# First try

```
pheatmap(fruits)
```

```
Error in hclust(d, method = method) :
```

```
NA/NaN/Inf dans un appel à une fonction externe (argument 10)
```

```
De plus : Warning messages:
```

```
1: In dist(mat, method = distance) :
```

```
  NAs introduits lors de la conversion automatique
```

```
2: In dist(mat, method = distance) :
```

```
  NAs introduits lors de la conversion automatique
```

Why does it fail?

# Second try: better?

```
pheatmap(fruits[, -(1:2)])
```

# The arguments

- `cluster_rows = FALSE` : remove row dendrogram
- `scale = "column"` : standardise variables
- `show_rownames = FALSE` : hide row names
- `cellwidth = 10` : smaller cells

To get a complete list of all arguments: `?pheatmap`

# Thrd try

```
pheatmap(  
  fruits[, -(1:2)],  
  cluster_rows = FALSE,  
  scale = "column",  
  show_rownames = FALSE,  
  cellwidth = 10  
)
```

# Fourth try: change the colors

```
colfun <- colorRampPalette(  
  c("darkorchid",  
    "white",  
    "limegreen"))
```

```
pheatmap(  
  fruits[, -(1:2)],
```

```
  cluster_rows = FALSE,  
  scale = "column",  
  show_rownames = FALSE,  
  cellwidth = 10,  
  color = colfun(20)  
)
```

# Fifth try: add “qualitative” information

```
colfun <- colorRampPalette(  
  c("darkorchid",  
    "white",  
    "limegreen"))  
fruitsDF <- data.frame(  
  fruits[, -1],  
  row.names = make.unique(fruits$nom))  
annotLignes <- fruitsDF[, "groupe",  
                        drop = FALSE]
```

```
pheatmap(  
  fruitsDF[, -1],  
  cluster_rows = FALSE,  
  scale = "column",  
  show_rownames = FALSE,  
  cellwidth = 10,  
  color = colfun(20),  
  annotation_row = annotLignes  
)
```



# Your turn!

Change the following command to get a pretty heatmap.

```
pheatmap(  
  t(fruits),  
  scale = "row",  
  color = c("black", "black"),  
  legend_breaks = c(-6, 0, +6),  
  border_color = "pink",  
  cellheight = 100,  
  cellwidth = 0.1,  
  show_colnames = "FALSE"  
)
```