

# Rbasics

PhD toolbox - 40th PhD cycle



Part V - Base graphics in R (some tips)

# Part V - Base graphics in R (some tips)

the base function to create graphics is `plot()` it simply creates a Cartesian plane where you can plot your data.

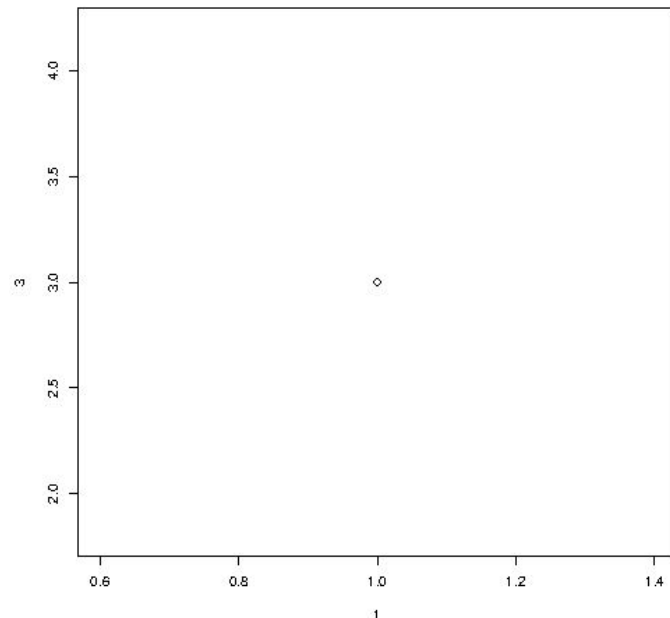
```
>plot(1,3)
```

or

```
>plot(x,y)
```

## EXERCISE:

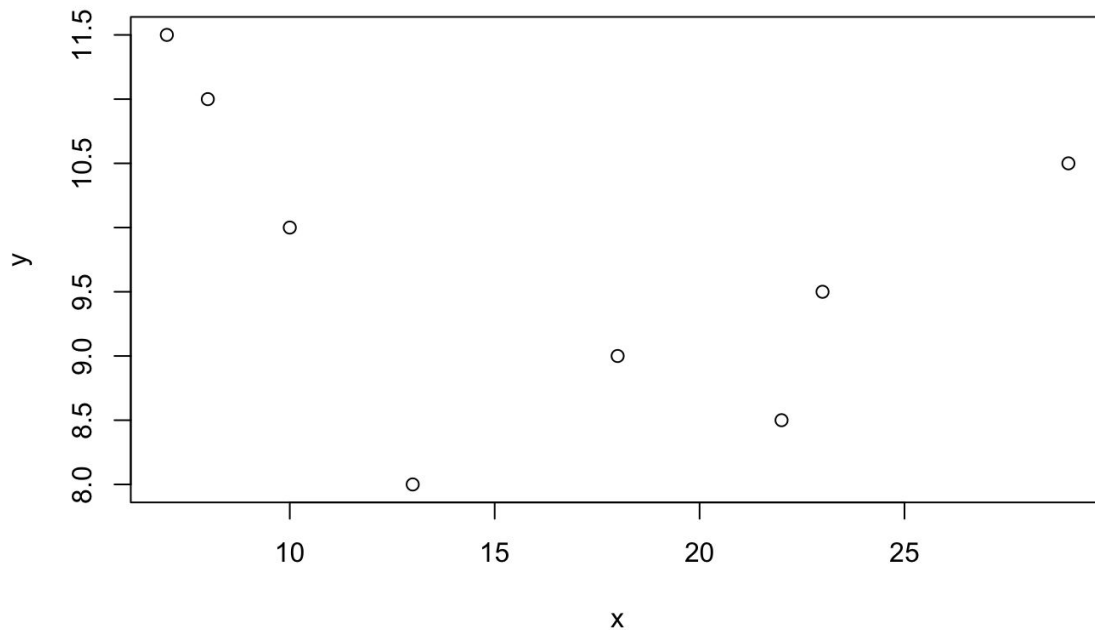
Substitute `x` and `y` with `colname_2` and `colname_3` vectors from the *df* object.



# Base graphics in R

## Exercise Solution:

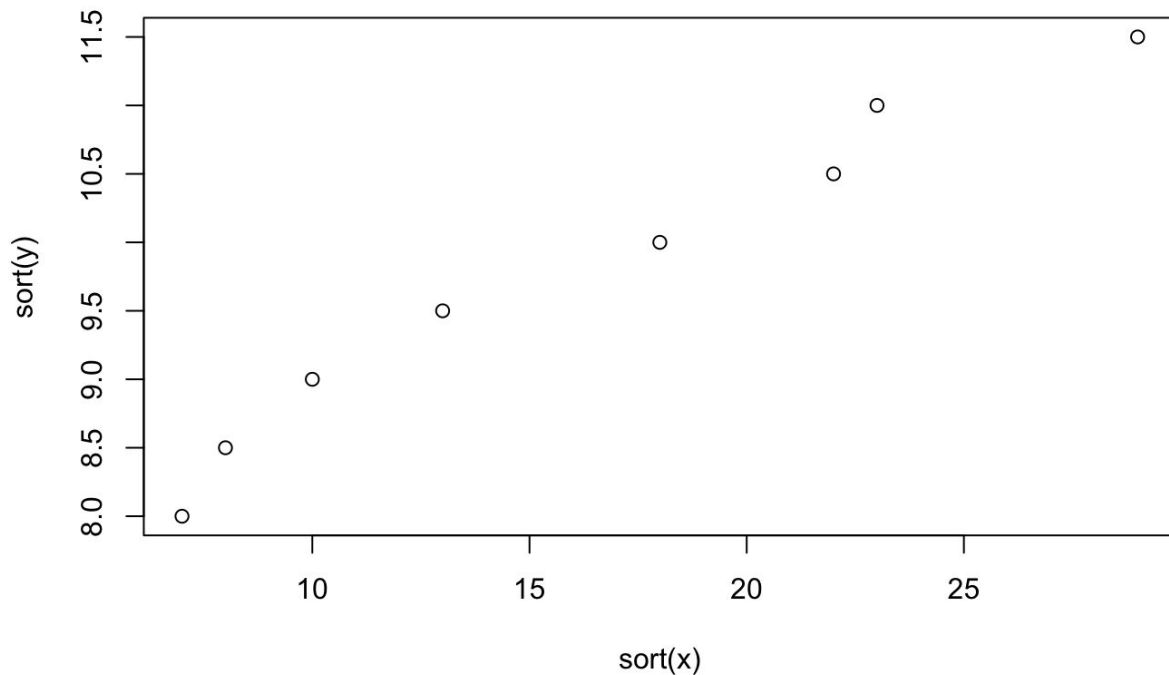
```
> x = df$colname_2  
> y = df$colname_3  
> plot(x,y)
```



# Base graphics in R

you can modify vectors directly before plotting to ameliorate the graphical output

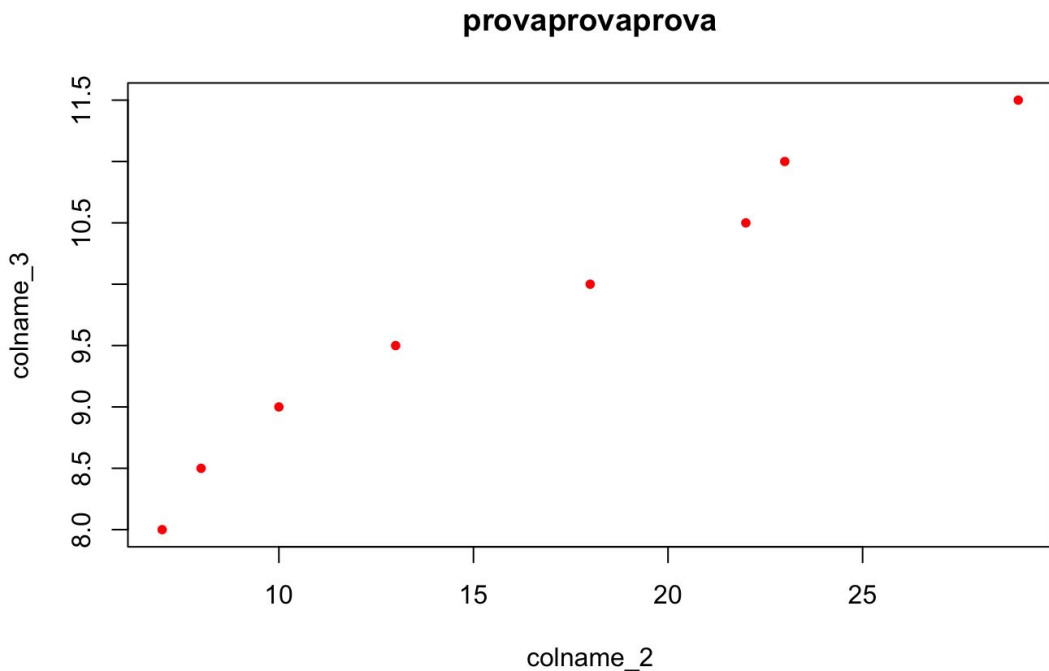
```
> plot(sort(y)~sort(x))
```



# Base graphics in R

Plots can be modified in many different ways (most of Radvance program)

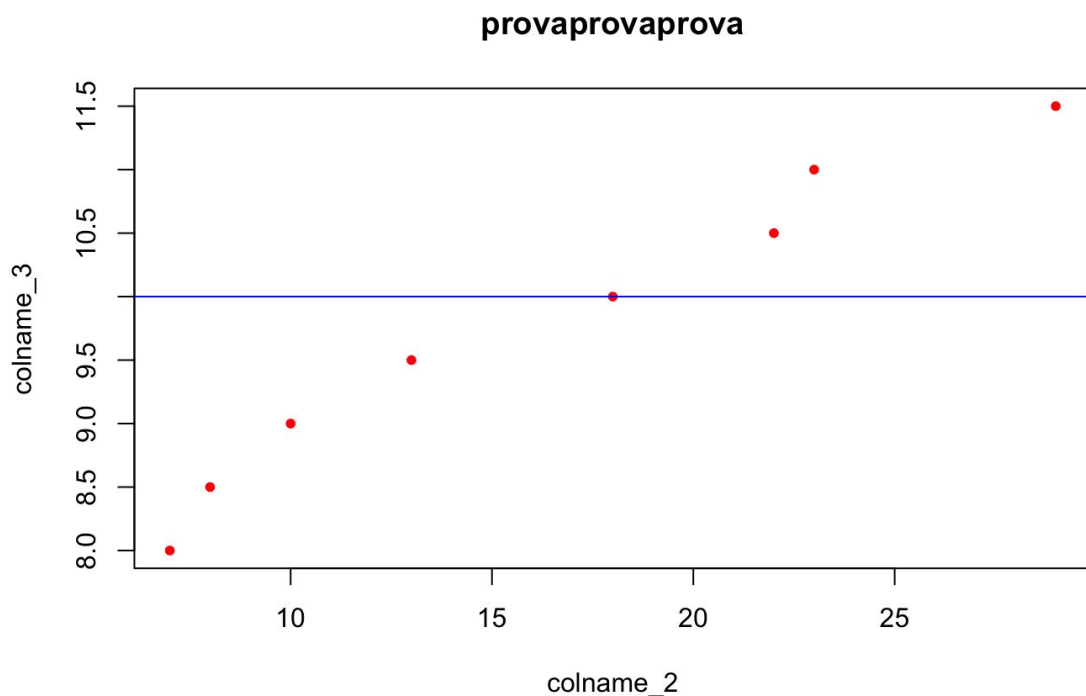
```
> plot(sort(y)~sort(x),  
       col = "red",  
       pch = 20,  
       main = "provaprova",  
       ylab = "colname_3",  
       xlab = "colname_2")
```



# Base graphics in R

`plot()` is a canvas on which you can draw secondary elements, such as lines and legends

```
> plot(sort(y)~sort(x),  
       col = "red",  
       pch = 20,  
       main = "provaprova",  
       ylab = "colname_3",  
       xlab = "colname_2")  
> abline(10,0, color = "blue")
```



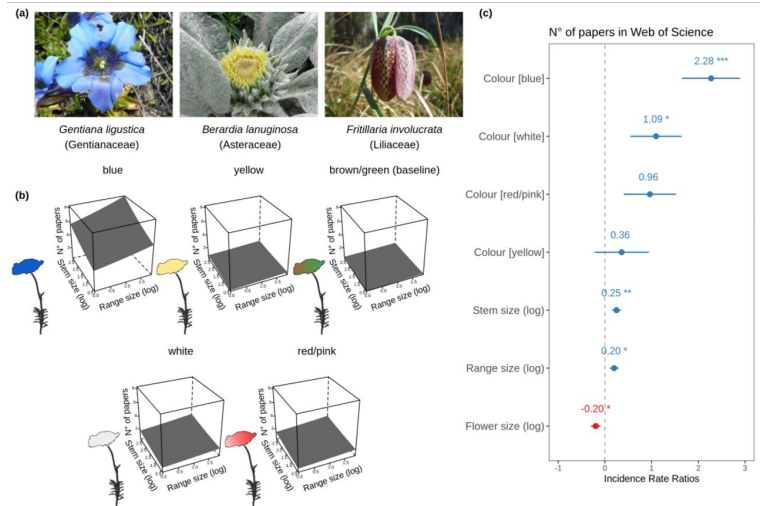
# benefits of scientific graphics in R

## PROS

1. **Understanding:** steep learning curve
2. **Efficiency:** display different information in small space.
3. **Location:** it integrates mapping directly in graphs
4. **Cost:** R is free country to many other graphic tools

## CONS

1. **Time:** especially first times could be time-consuming
2. **Distraction:** you can build complex and fancy graphics-rich reports and charts, focusing more on the form than the function.



# graphical notes for scientific data plotting

Easy/natural  
color  
associations

Use as few  
colors as you  
can

Use the same  
color for the  
same object  
through the  
whole report



Each part  
should be easy  
readable

No gradients  
for categories

Legends are  
gold as well as  
measure units



# COLOR PALETTES

## QUANTITATIVE DATA - SEQUENTIAL OR DIVERGING COLORS

Color is used to show variations in the data. The palette contains a sequence of colors that clearly indicate which values are larger or smaller than which other ones (sequential scale). It can also visualize the deviation of data values in one of two directions relative to a neutral midpoint (diverging scale). Diverging scale can be viewed as two merged sequential scales.



## CATEGORICAL DATA - QUALITATIVE COLORS

Color is used to separate areas into distinct categories. The palette should consist of colors as distinct from one another as possible. The maximum number of categories that can be displayed is about 12 (practically speaking, probably fewer).



All examples are available in Seaborn library. Check also: [medialab.github.io/iwanthue/](https://medialab.github.io/iwanthue/)

## USAGE GUIDELINES



Colors are useful make your graphs readable, but they must be used in a proper way

Journals are increasingly asking for **color-blind** friendly figures:

~8% of the global population is affected by colorblindness (mainly males)

there are many packages to create palettes in R:

the most famous = [RColorBrewer](https://rcolorbrewer.github.io/)

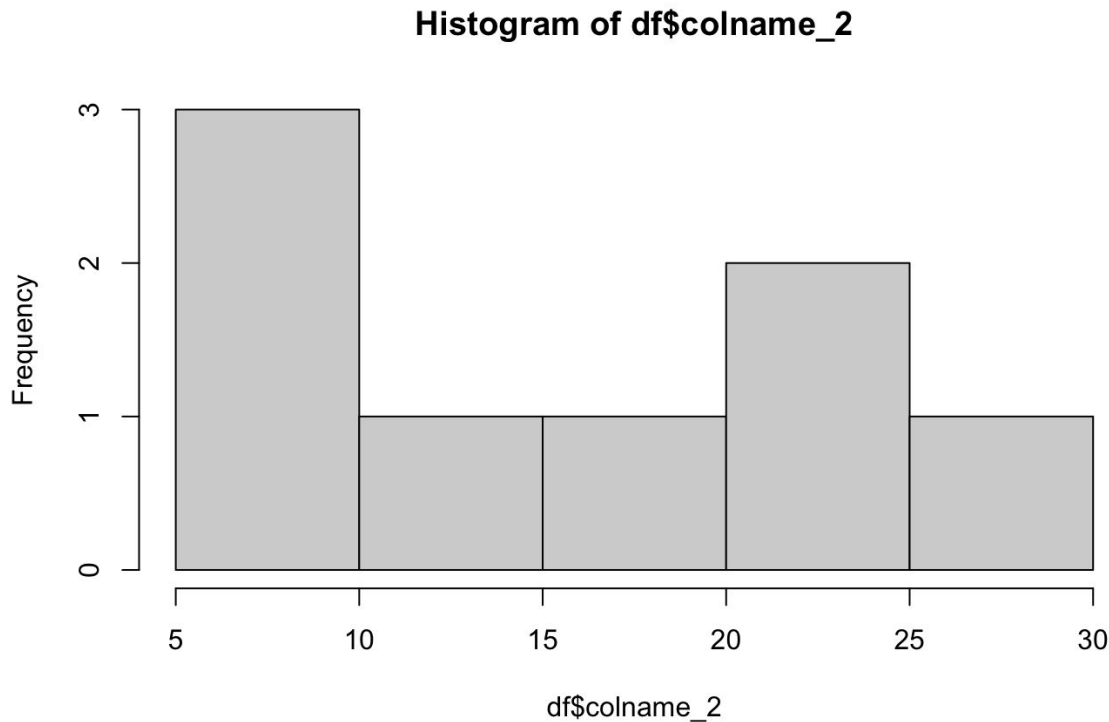
the most complete = [paletteer](https://github.com/mbauman/paletteer)

Additionally to colors there are shadings, line styles, point styles

# main base graph functions - histograms

hist() is useful for visualize frequencies

```
> hist(df$colname_2)
```



# main base graph functions - boxplots

```
> str(df)

'data.frame':  8 obs. of  3 variables:
 $ colname_1: chr  "A" "A" "B" "B" ...
 $ colname_2: int   13 22 18 23 10 29 8 7
 $ colname_3: num   8 8.5 9 9.5 10 10.5 11 11.5
```

Boxplots are useful to see a variable response to a specific factor ... than you need to verify that you actually have a factor!

# main base graph functions - boxplots

```
> boxplot(df$colname_2 ~ as.factor(df$colname_1), main = "box1")
```

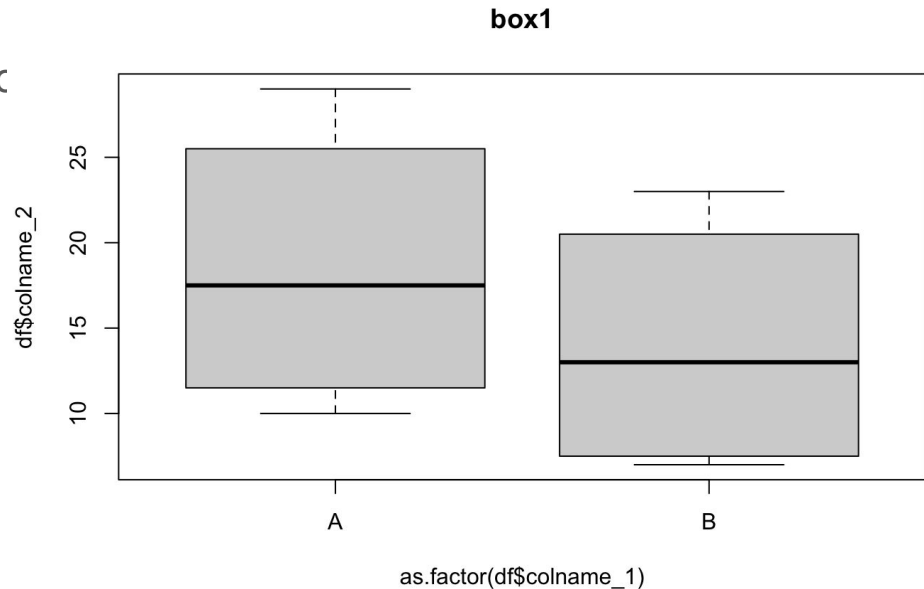
Factors must be in the second argument

you can see multiple plots using the function

```
> par(mfrow=c(plots x row, plots x col))
```

## Exercise:

Visualize the two possible boxplots from df in a single image.



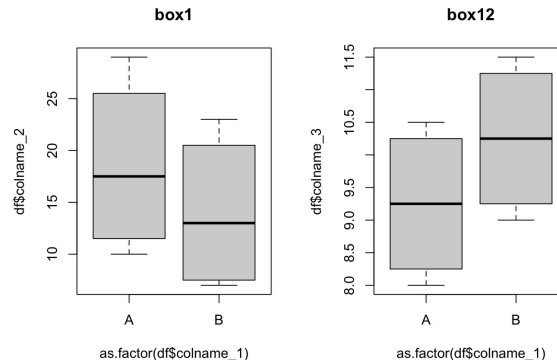
# main base graph functions - boxplots

## Solution:

```
> par(mfrow=c(1,2))
```

```
> boxplot(df$colname_2 ~ as.factor(df$colname_1), main =  
"box1")
```

```
> boxplot(df$colname_3 ~ as.factor(df$colname_1), main =  
"box2")
```



# Saving figures

Find your-own way, but remember that:

- export tool from Rstudio depends on the resolution of your screen.
- figure sizes will depend from the plot window size (by default in Rstudio)
- you can avoid this steps saving images by using the command line (specific functions)
- journals want high resolutions figures (usually 300 dpi or higher), exporting \*.pdf figures you save vectorial figures corresponding to infinite dpi!
- post-edit figure as few as you can
- post-edit figures with appropriate softwares (NO POWERPOINT!)

More hints in Stream 2 lessons ... don't miss it!

# PhD Toolbox - Get ready for Stream 2!

## Working with **lists**

- More advanced stuff on graphics (**ggplot2**)
- composite graphs panels (**gridExtra**, ...)
- Exporting figures
- **Plotting Maps using R**

## **Aula 1**

<b>Thursday January 23</b>	<b>h 10-13</b>	<b>(sede di Viale Mattioli 25 - Botanical Garden)</b>
<b>Friday January 24</b>	<b>h 10-13</b>	<b>(sede di Viale Mattioli 25 - Botanical Garden)</b>
<b>Tuesday January 28</b>	<b>h 10-13</b>	<b>(sede di Viale Mattioli 25 - Botanical Garden)</b>
<b>Thursday January 30</b>	<b>h 10-13</b>	<b>(sede di Viale Mattioli 25 - Botanical Garden)</b>
<b>February February 4</b>	<b>h 9-13</b>	<b>(sede di Viale Mattioli 25 - Botanical Garden)</b>