Alexander Strobel
Faculty of Psychology

Graphics

A short guide on how to create figures

Graphics
Motivation

If you …

— can‘t tell the difference between vector graphics and raster images
— create your figures in Powerpoint or Word or simply use SPSS output
— don‘t know about the standard figure width of the journal you want to submit your paper to
— have never heard of encapsulated postscript or tagged image file format
— think that your software‘s 72 dpi default resolution is fairly high
— feel insecure what information should be given in a figure and what information can be omitted

Then this is a workshop for you because you will learn …

— that all of the above is doomed to failure
— how to create high quality figures from scratch that conform to most journals‘ standards
— to provide what is needed to be provided about your data and results in a figure
— to omit what simply seems fancy but is not needed or may even be confusing

Graphics
Overview

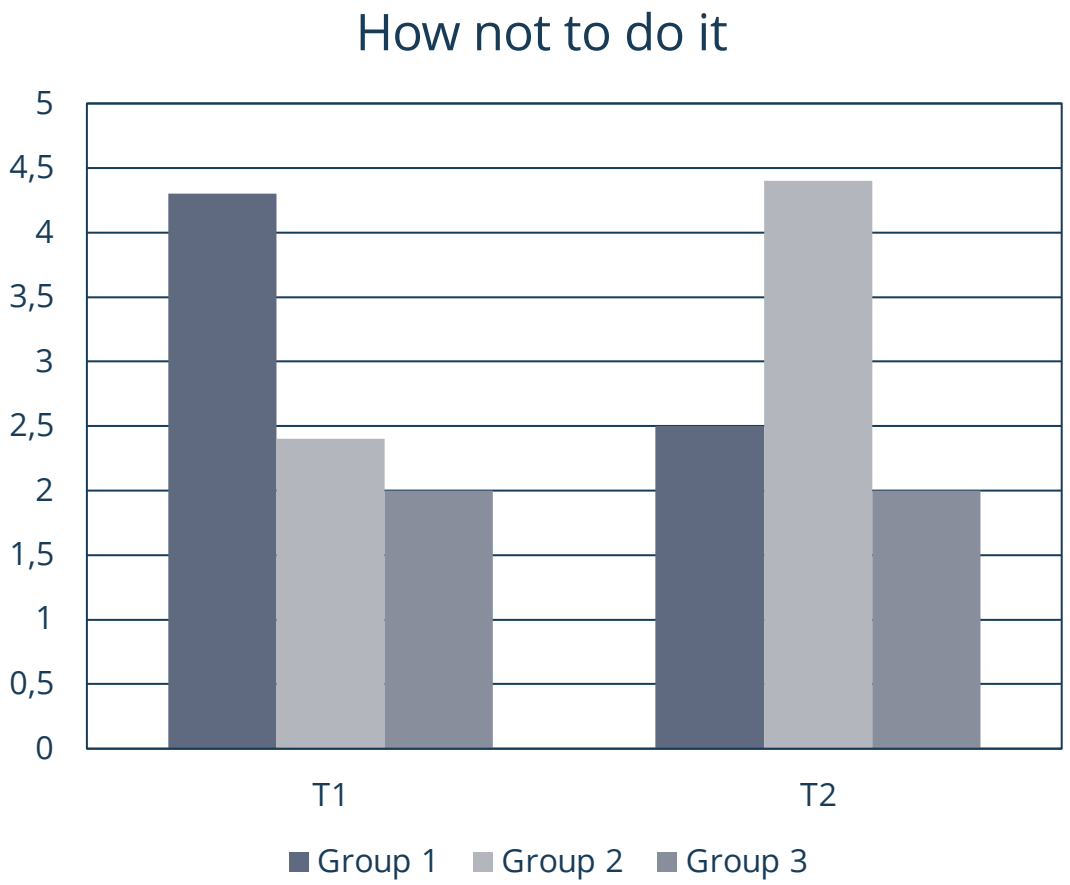
Types of graphics
- Vector graphics
- Raster images

Standard image formats and sizes
- EPS
- TIF
- Image sizing and resolution
- Other issues to keep in mind

Information to be provided in figures
- and information that can be omitted

Fine-tuning

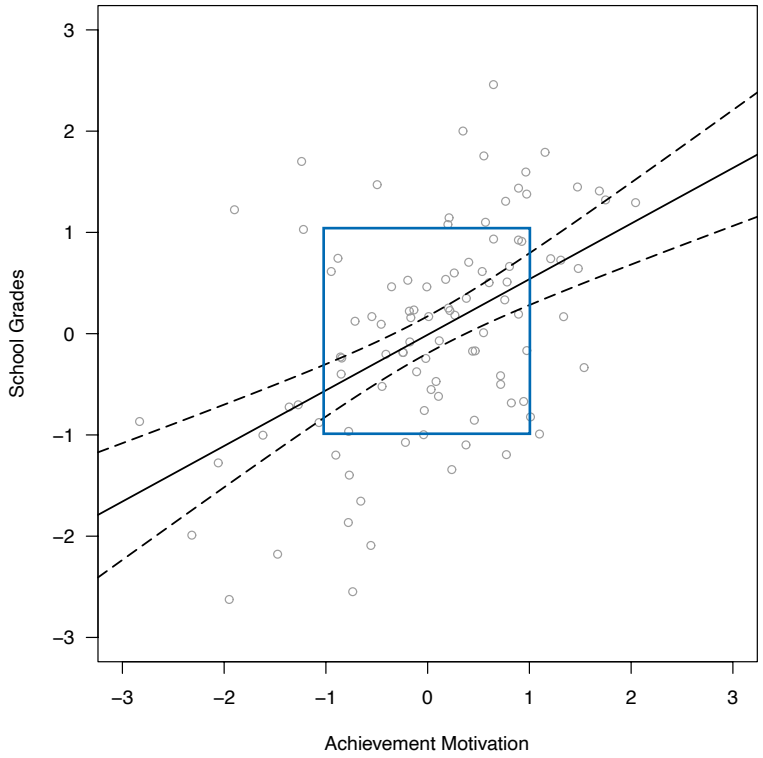
Summary and outlook



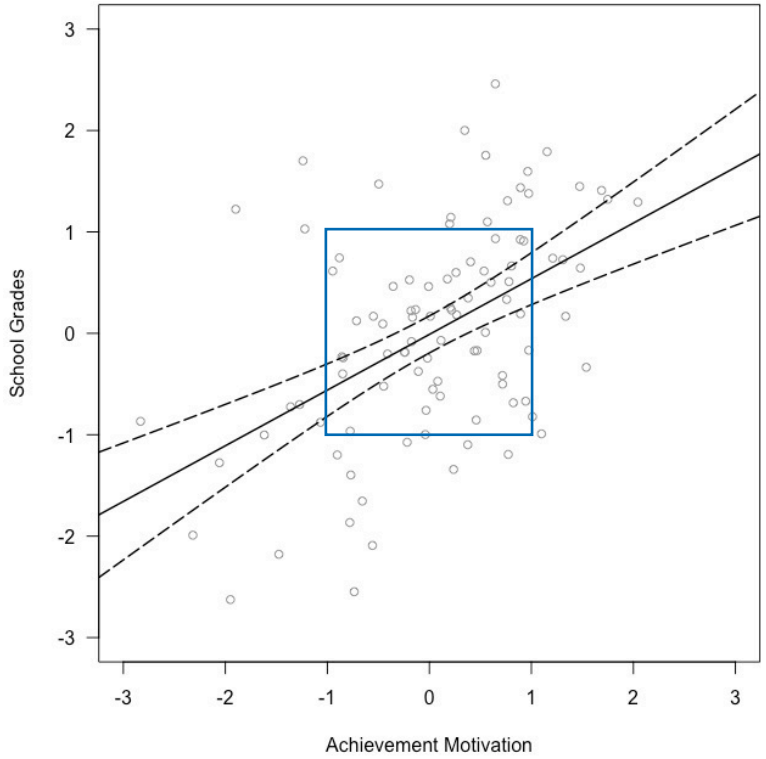
Types of graphics

Types of graphics
Vector vs. raster

Vector image (*.eps, 14 KB)

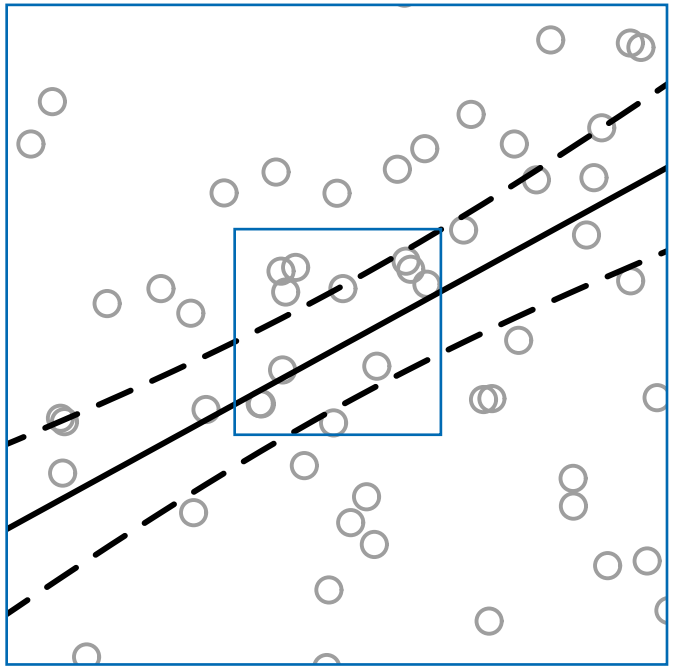


Raster image (*.jpg, 72 dpi, 45 KB)



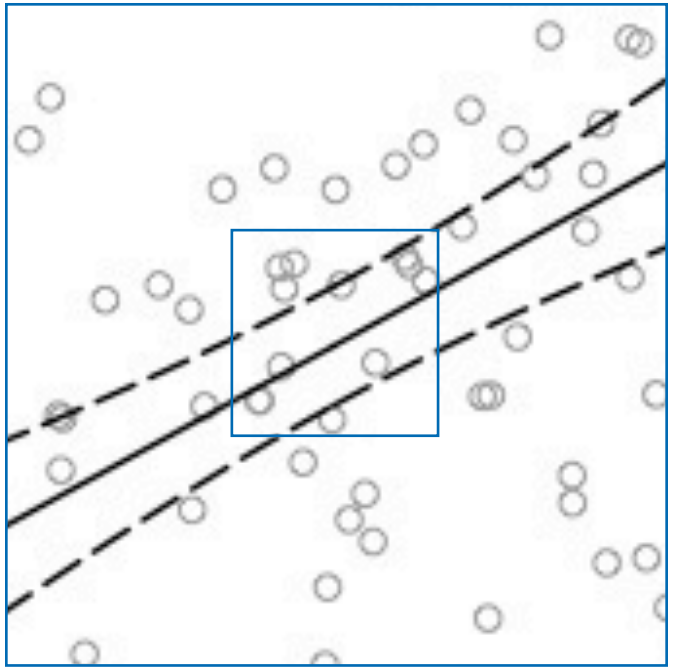
Types of graphics
Vector vs. raster

Vector image
(*.eps, 14 KB)



300%

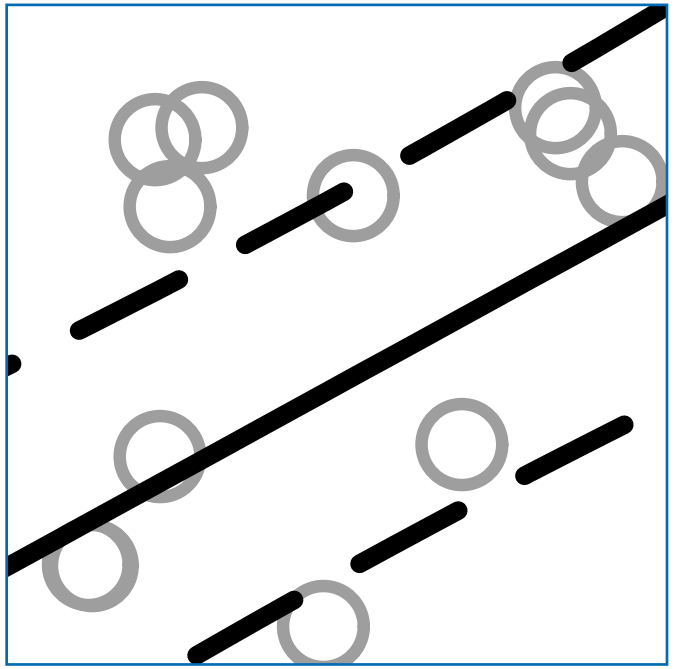
Raster image
(*.jpg, 72 dpi, 45 KB)



300%

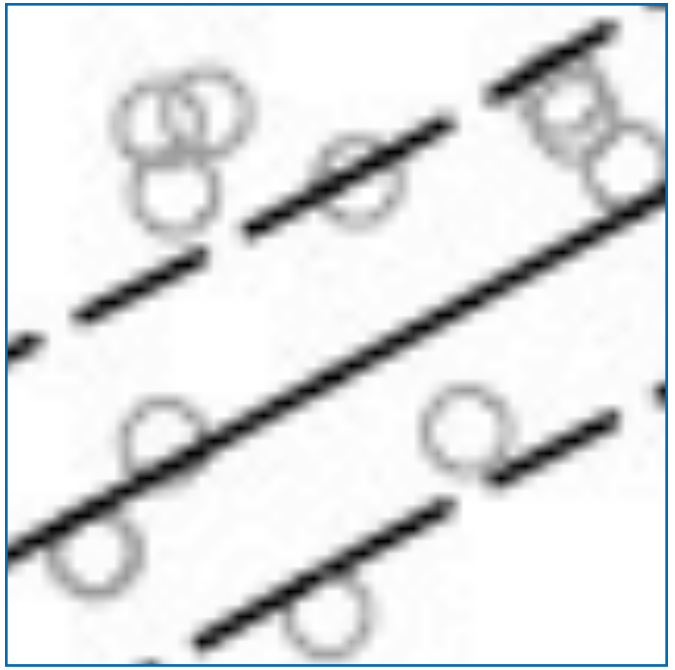
Types of graphics
Vector vs. raster

Vector image
(*.eps, 14 KB)



900%

Raster image
(*.jpg, 72 dpi, 45 KB)

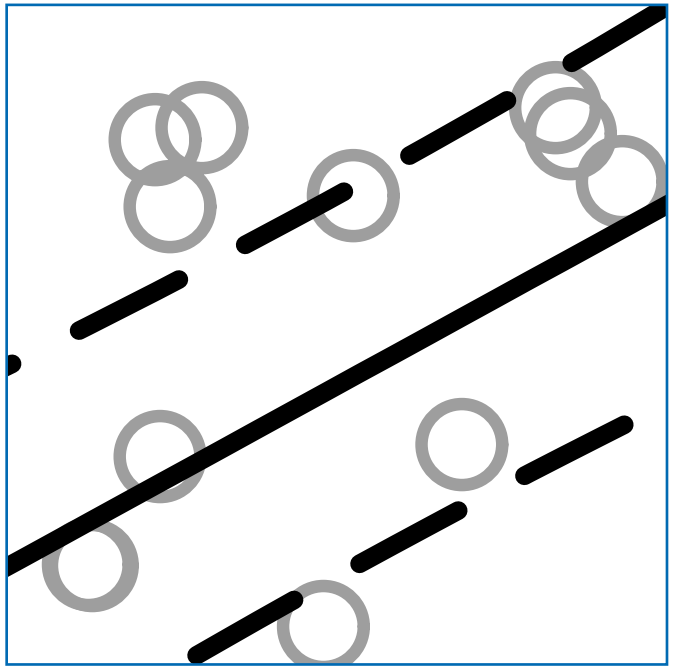


900%

Types of graphics
Vector vs. raster

How is this lossless magnification possible?

Because this …



… is actually this:

```
%!PS-Adobe-3.0 EPSF-3.0
%%DocumentNeededResources: font Helvetica
%%+ font Helvetica-Bold
%%+ font Helvetica-Oblique
%%+ font Helvetica-BoldOblique
%%+ font Symbol
%%Title: R Graphics Output
...
/bg { 1 1 1 srgb } def
0.00 0.00 496.50 537.75 r p2
59.04 73.44 466.26 478.71 cl
0.6196 0.6196 0.6196 srgb
0.75 setlinewidth
[] 0 setdash
1 setlinecap
1 setlinejoin
10.00 setmiterlimit
372.53 358.71 2.43 c p1
338.66 322.40 2.43 c p1
...
```

Types of graphics
Vector graphics

Principle

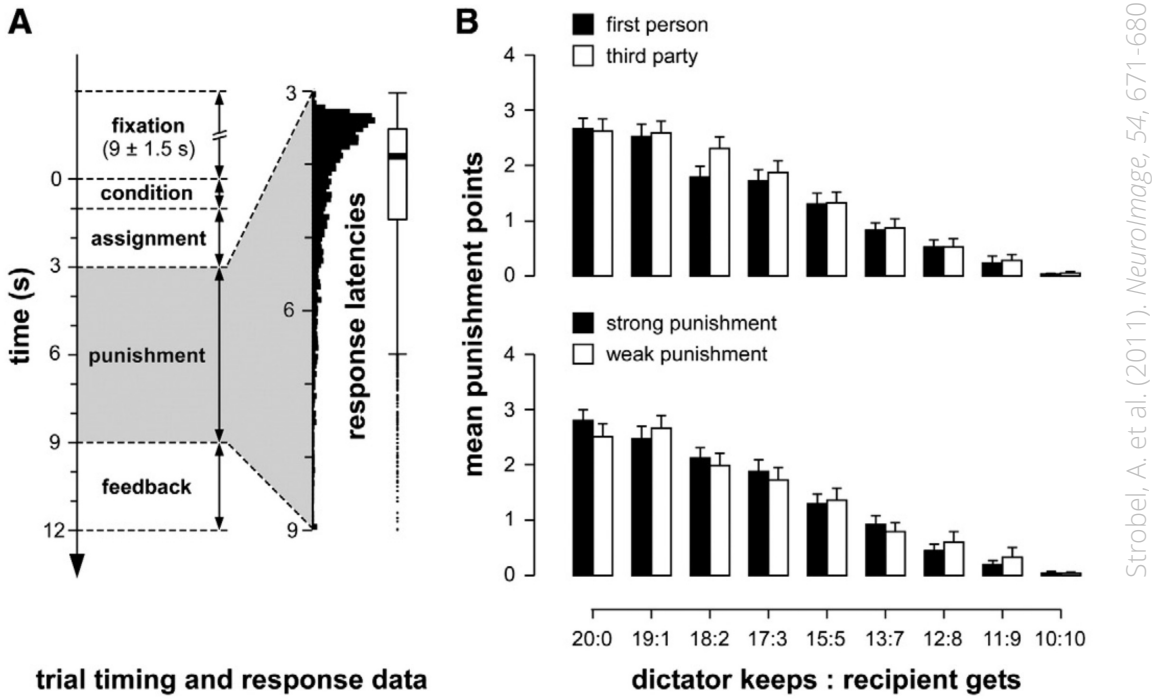
An image (usually line art) is decomposed into paths and only the information describing the paths is stored

Example: a line can be defined by the following characteristics

— start and end point
— width, style (e.g., dashed) and color

Standard formats

— Portable Document Format (*.pdf)
— Scalable Vector Graphics (*.svg)
— Encapsulated Post Script (*.eps)



Strobel, A. et al. (2011). *NeuroImage, 54*, 671-680.

Types of graphics
Raster images

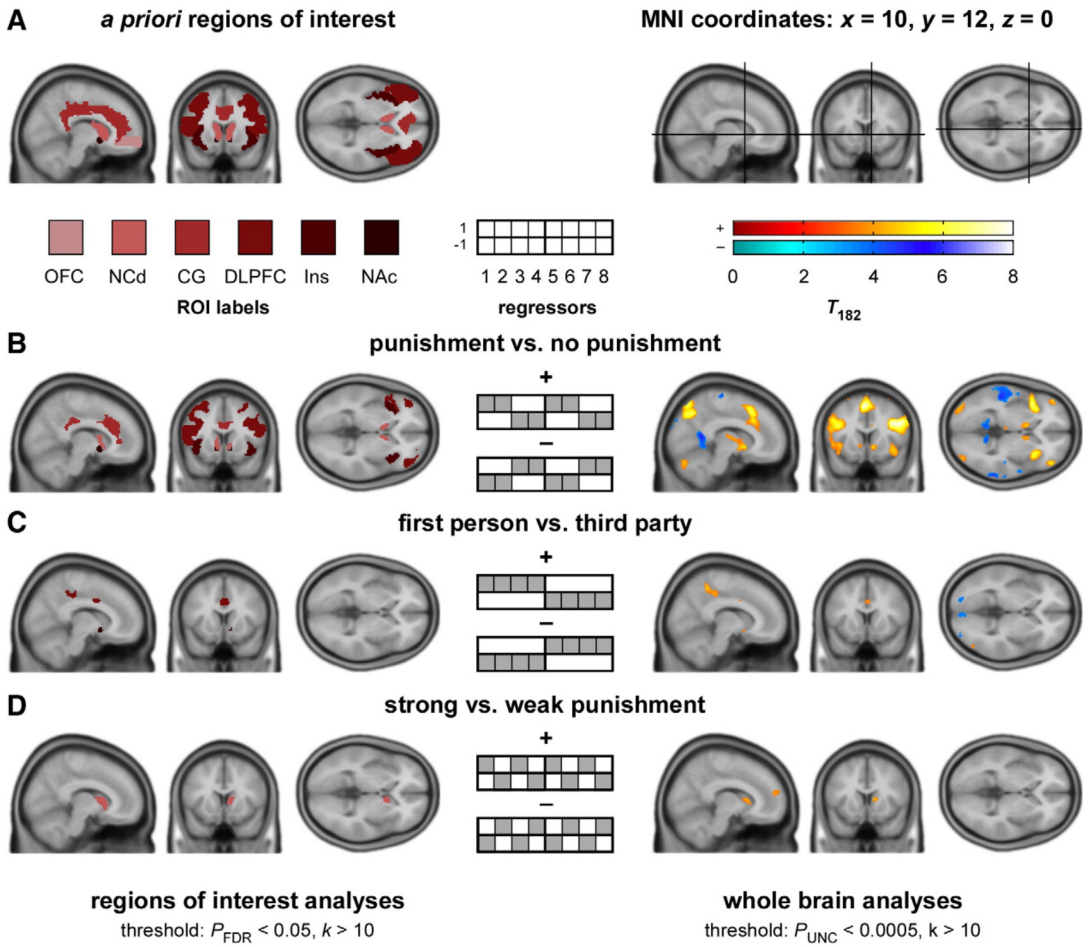
Principle

Here, an image consists not of paths, but of pixels, and each pixel needs to be described in terms of its color (standard case is a photo)

For high-resolution images (300 dpi minimum), this results in large image files that still cannot be magnified ad libitum without loss of quality

Standard formats

- Bitmap (*.bmp)
- JPEG (*.jpg)
- Portable Network Graphics (*.png)
- Tagged Image File Format (*.tif)



Strobel, A. et al. (2011). *NeuroImage, 54*, 671-680.

Standard image formats and sizes

Standard image formats and sizes
Image formats

Journal requirements

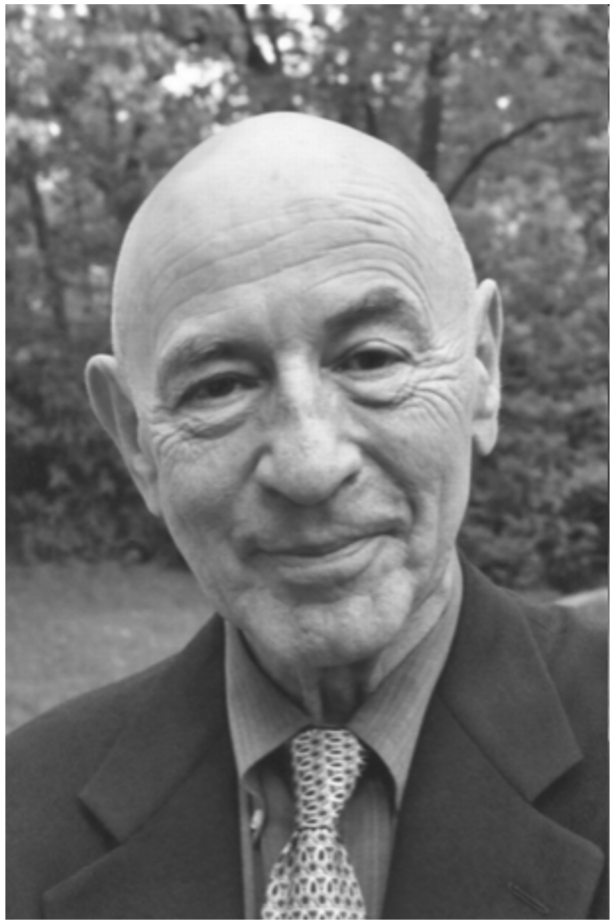
Until recently, journals had very different (and changing) requirements for image formats/sizes, making it very hard for authors to anticipate what format and size their images should have for submission

If anything good can be seen in publishing monopoles such as Elsevier or SpringerNature, it is their now unified Submission Guidelines for all journals of a given publisher that also include requirements for artwork

Frontiers or PLoS also have unified guidelines for artwork

All publishers have in common that they usually accept
— **Encapsulated Postscript (*.eps)** for line art (i.e., vector graphics)
— **Tagged Image File Format (*.tif)** for raster or combined images

The picture of Walter Mischel on the right has no purpose except that it is another example of a raster image and introduces the term *grayscale*



Standard image formats and sizes
Image formats: Vector

Encapsulated Postscript (*.eps)

Figures can be saved in *.eps format by R or MATLAB, PowerPoint only allows to save images in a number of raster formats (or PDF, which is sometimes accepted by some journals/publishers, but don't count on that)

To edit *.eps files or other vector graphics, you need software that is non-standard on most machines, be it PC or Mac (but you might be well equipped when using Linux)

Software tools to edit vector graphics are

- **Adobe Illustrator** (but you have to rent a license and need to renew it annually, so that is not an option, we don't like this type of licensing)
- **Inkscape** (a free and open source tool, that – like many other free and open source software tools – has some disadvantages in using it)
- **Affinity Designer** (also licensed, but a rather cheap purchase via ZIH and a fully versatile tool that may even outperform Adobe Illustrator)

Yet …

If prepared properly in R or MATLAB, there should be no need to edit your figures anyway. A different issue are combined vector/raster images (like the one on slide 10). Such figures can still be composed solely using R or MATLAB, but it is painstaking.

Standard image formats and sizes
Image formats: Raster

Tagged Image File Format (*.tif)

R (and most likely also MATLAB) allows to save images as *.tif

The Preview App on Mac also allows you to export raster images to *.tif, on Windows machines there are certainly also tools to do so (such, as I guess, IrfanView)

Software tools to edit raster images are

- **Adobe Photoshop** (but again you have to rent an annual license and that's crap)
- **Gimp** (a free and open source tool, that – like many other free and open source software tools including Inkscape – has some disadvantages in using it)
- **Affinity Photo** (also licensed, but a rather cheap purchase via ZIH and a fully versatile tool that yet has to grow up to challenge Adobe Photoshop)

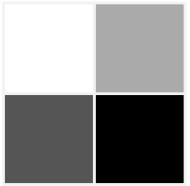
Still …

Simply converting a raster image to *.tif will not impress the people at Elsevier or Frontiers. It has to be in the correct resolution. The required resolution is indicated in the guidelines, but to be on the safe side, save your *.tif file with 1000 dpi and LZW compression!

Standard image formats and sizes
Color spaces

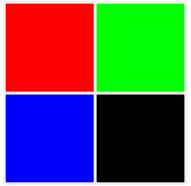
Grayscale

I already mentioned that term a few slides earlier, it refers to black and white images with gray-shading



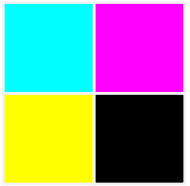
RGB

stands for **R**ed-**G**reen-**B**lue and is the most common color space for images delivered on screen (such as *.jpg). Every pixel is described via a hexadecimal value ranging from #000000 = black to #FFFFFF = white; #FF0000 = pure red (among R users also known as col=2), #00FF00 = pure green (col=3) and #0000FF = pure blue (col=4)



CMYK

This is a color space different from RGB. It describes images in terms of the proportion of **C**yan-**M**agenta-**Y**ellow-Blac**k** hues of your image. It is used mainly for printing purposes, and the less actual printing is done nowadays by publishers, the less important CMYK has become for authors



Standard image formats and sizes
Image sizing and resolution

Sizing/resolution requirements differ not much across publishers

Consider whether your figure is best presented in one-column or full-page width

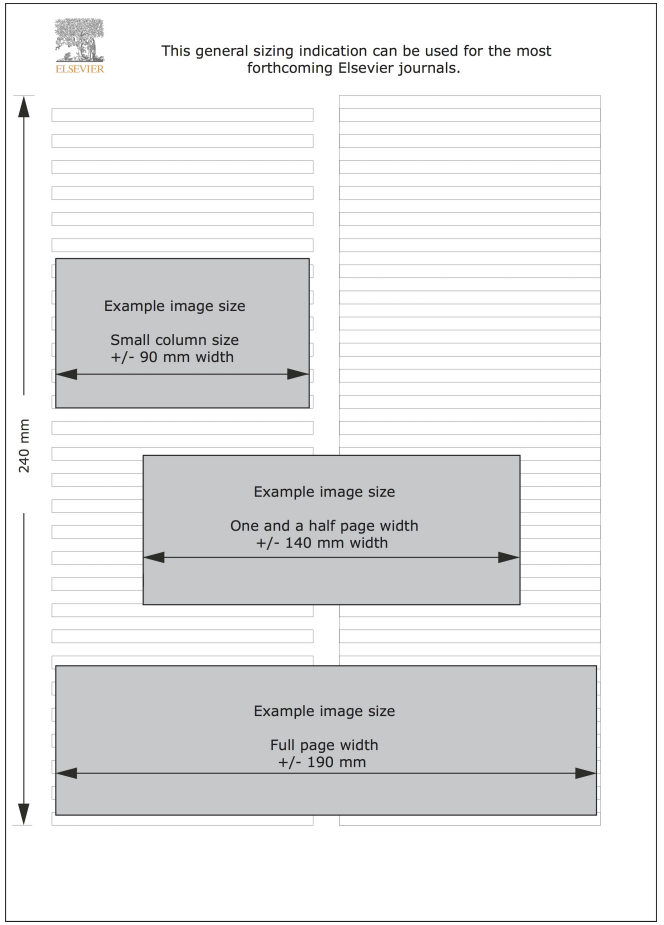
— one-columns width is usually around 90 mm
— full-page width is usually between 180 and 190 mm

In any case, your raster images (mostly *.tif with LZW compression, *.jpg is sometimes accepted) should have a resolution of ≥ 300 dpi

Make sure to strictly adhere to journal standards, here a few examples

— Frontiers
— PLoS
— Elsevier
— Nature

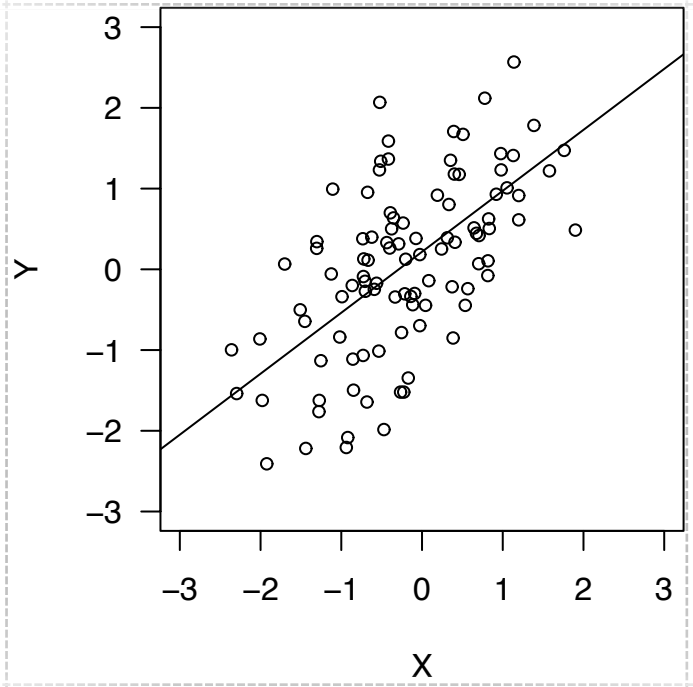
The latter link is especially recommended



Standard image formats and sizes
Image sizing and resolution

Example for R

```
# function to convert millimeters to inches
mm2in <- function(mm)  { mm * .0393701 }

# generate data to plot
set.seed(242)
X = rnorm(100)
Y = .5 * X + rnorm(100)

# set plot margins to exactly fit plot content
par(mar=c(4, 4, 0.1, 0.1))

# scatterplot of X and Y with regression line
plot(X, Y, xlim = c(-3, 3), ylim = c(-3, 3), cex = .8, las = 1)
abline(lm(Y ~X))

# copy current plot to *.eps file with width and height of 90 mm each
dev.copy2eps(file = "Fig1.eps", width = mm2in(90), height = mm2in(90))

# reset plot margins
par(mar=c(5, 4, 4, 2))
```



Standard image formats and sizes
Other issues to keep in mind

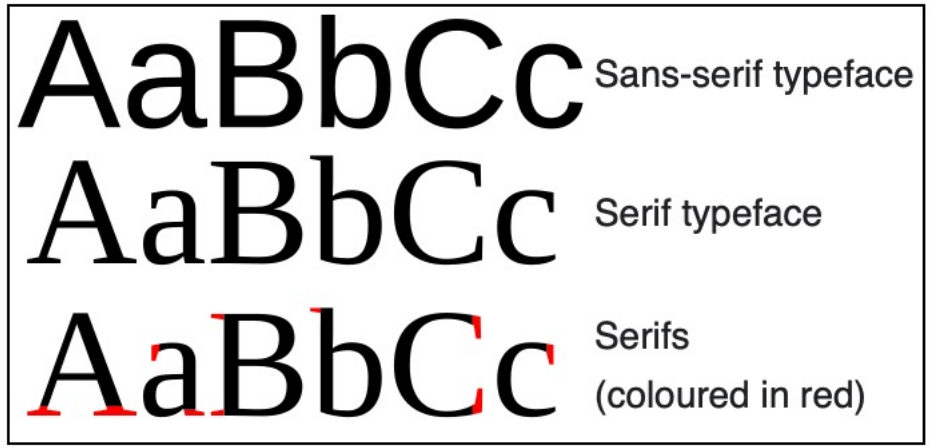
Font type and size

Always use Helvetica (as R does per default) or Arial, i.e., so-called sans serif fonts

Some people believe that APA-standard requires to use serif fonts like Times New Roman for figures. They are wrong (see right lower panel).

Font sizing should be chosen to ensure maximum readability (min. font size often is 6 or 7 pt) and eye-friendliness (max. font size should not exceed 10-12 pt)

Do not use unusual (e.g., Tahoma) or copyrighted (e.g., Frutiger) fonts even if they are sans serif, and *never ever* use Comic Sans for whatever reason (you won't use it in figures anyway, but for greeting cards? Don't!!!)



https://en.wikipedia.org/wiki/Sans-serif

Be certain in figures of all types that

- lines are smooth and sharp,
- typeface is simple (sans serif) and legible,
- units of measure are provided,
- axes are clearly labeled, and
- elements within the figure are labeled or explained.

APA 6th guidelines, p. 153

Standard image formats and sizes
Other issues to keep in mind

Point and line style and color

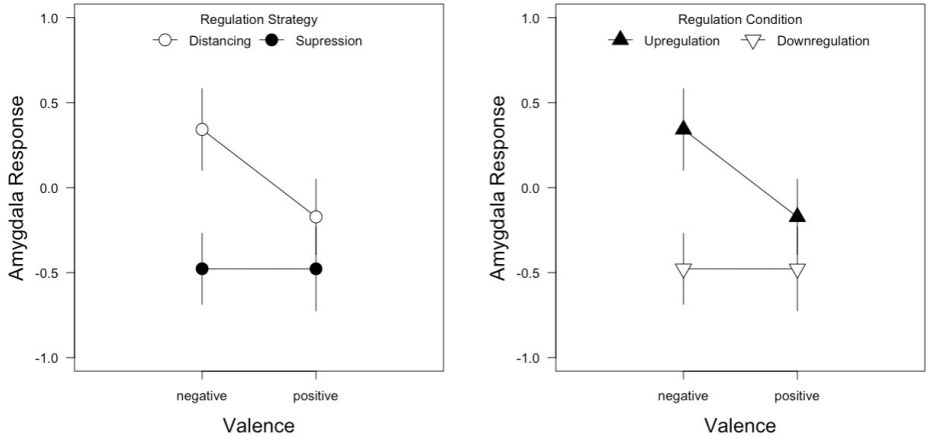
Use point style meaningfully! Mostly, circles will be the best choice (top left), yet, if you want to compare emotion down- vs. upregulation, triangles have meaning (top right)

Here, lines need no differentiable style (solid vs. dashed)

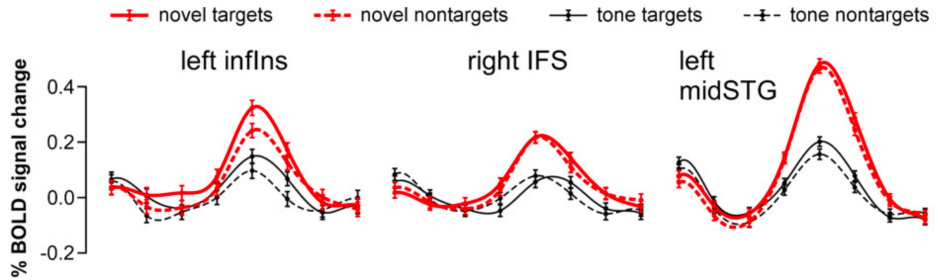
Back then when your article was disseminated mostly via print, you had to pay quite hefty color page charges. Things have changed, but many people still have b/w printers and might want to print out your paper

In the example on the right, color makes it easy to differentiate the conditions, but differentiation would still be possible if the same image was printed in b/w because of different line width, style, and color

Example point style



Example line style and color



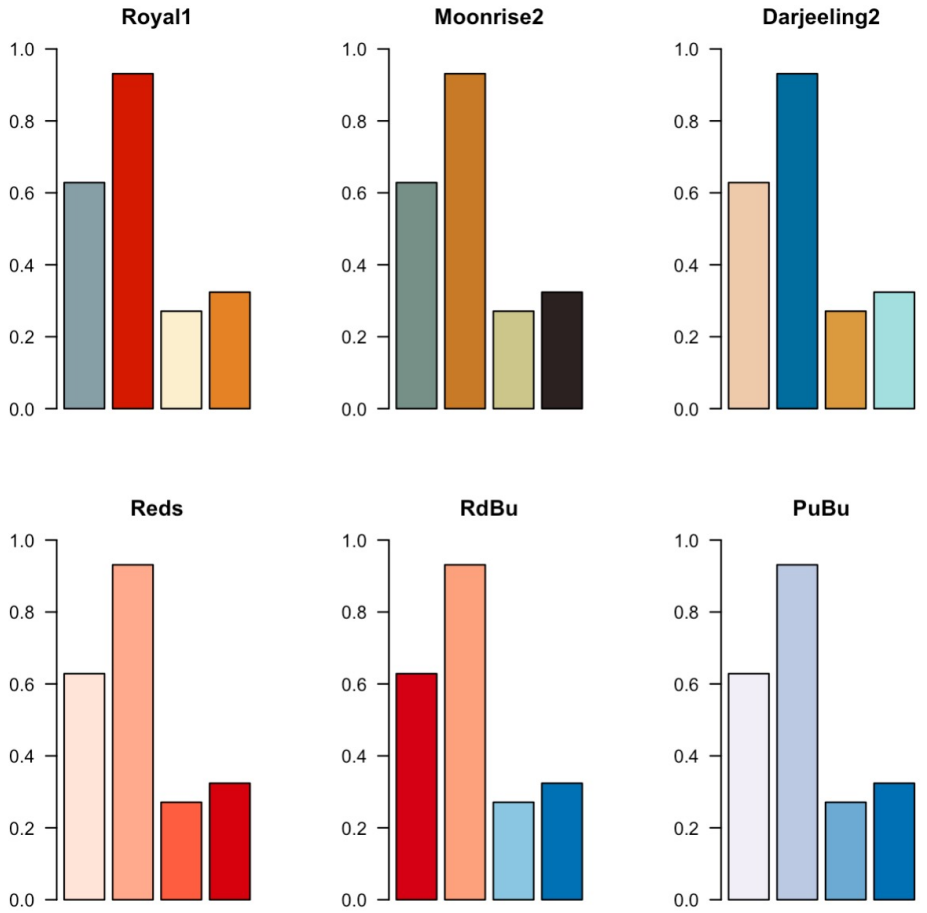
Standard image formats and sizes
Other issues to keep in mind

Color palettes

Base R comes with very basic colors that are quite distinguishable but do not look nice, here are several alternatives:

- The *wesanderson* package for R (see top right) comes with a number of great color palettes!
- Less fancy and more flexible is the *RColorBrewer* package (see bottom right)
- Further packages are *viridis* or *ggsci (*especially suited for *ggplot2)*

Still, you might want to play with colors and create your own color palettes, but make sure they are differentiable in b/w and for color-blind individuals (see http://www.cookbook-r.com/Graphs/Colors_(ggplot2)



Information to be provided in figures
and information that can be omitted

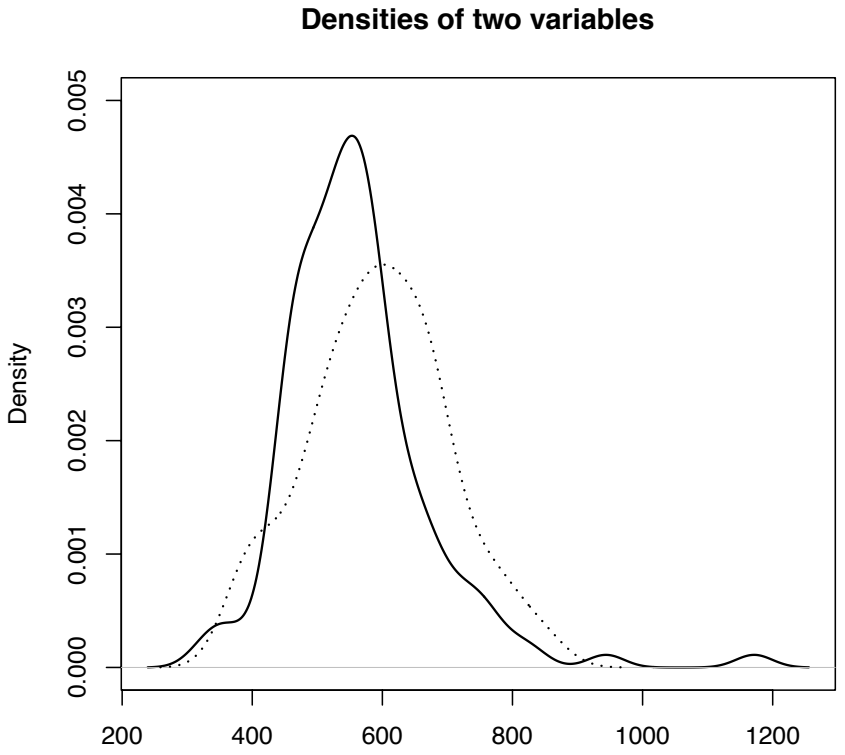
Information to be provided in figures
and information that can be omitted

Descriptive plots

Always visualize the distributional characteristics of your (main) data, as readers cannot infer them easily from means, SD, skew and kurtosis

For example, see figure on the right

Line	M	SD	Skew	Kurtosis
solid	560.90	111.74	1.70	6.85
dotted	560.91	108.65	0.04	-0.34

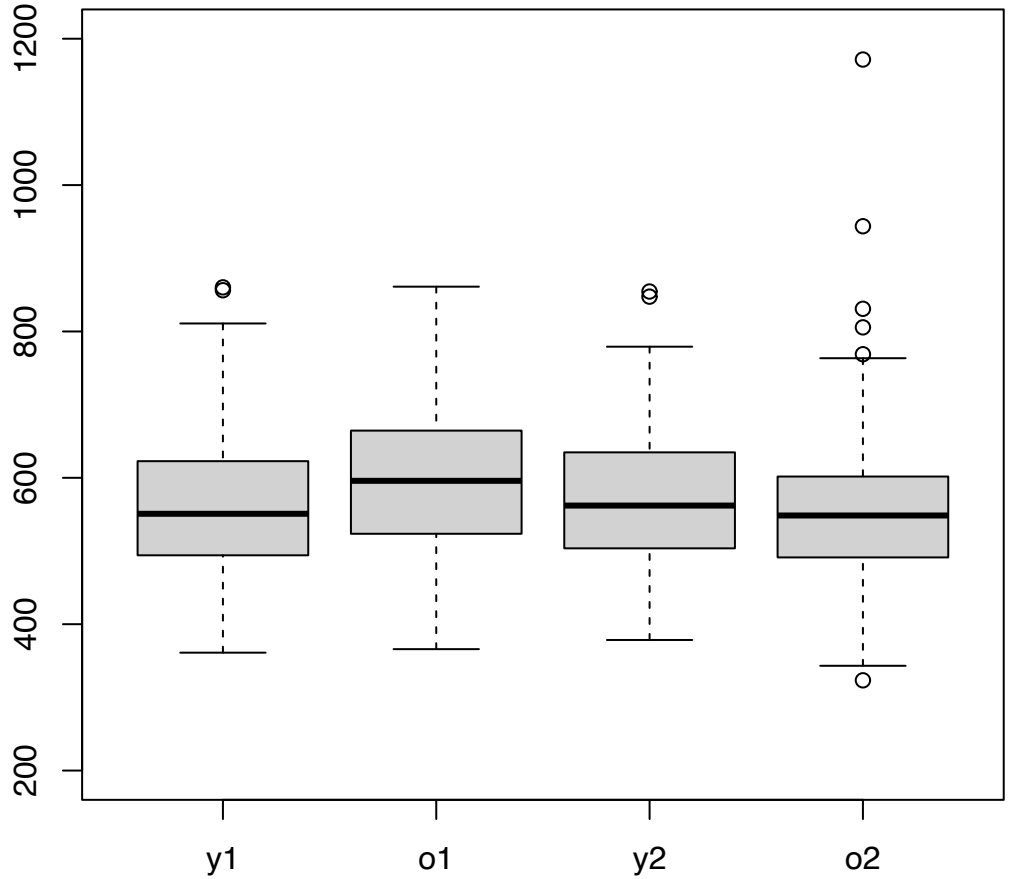


Information to be provided in figures
and information that can be omitted

Descriptive plots

Always visualize the distributional characteristics of your (main) data, as readers cannot infer them easily from means, SDs, skew and kurtosis

Boxplots will sometimes (but not always) do, but the basic R function will produce ugly and in part uninformative output



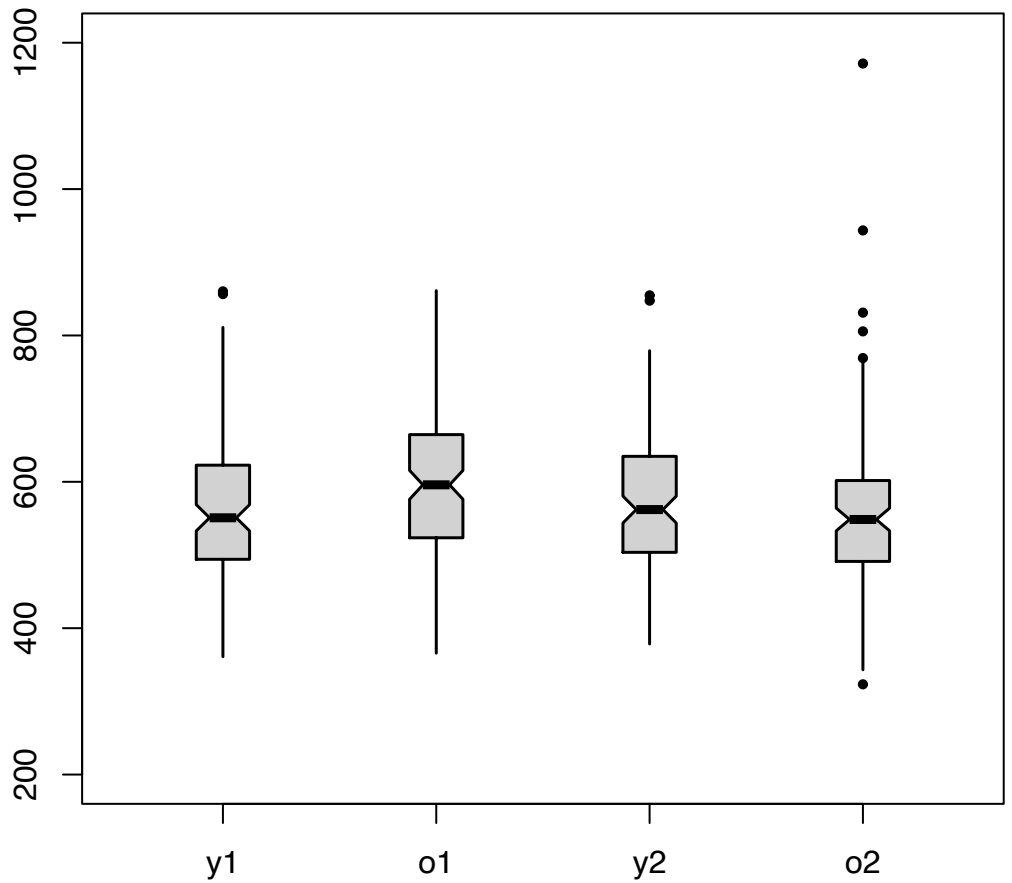
Information to be provided in figures
and information that can be omitted

Descriptive plots

Always visualize the distributional characteristics of your (main) data, as readers cannot infer them easily from means, SDs, skew and kurtosis

Boxplots will sometimes (but not always) do, but the basic R function will produce ugly and in part uninformative output

You can polish it and add notches (giving an impression of significant differences between variables), but still, condition differences are less obvious



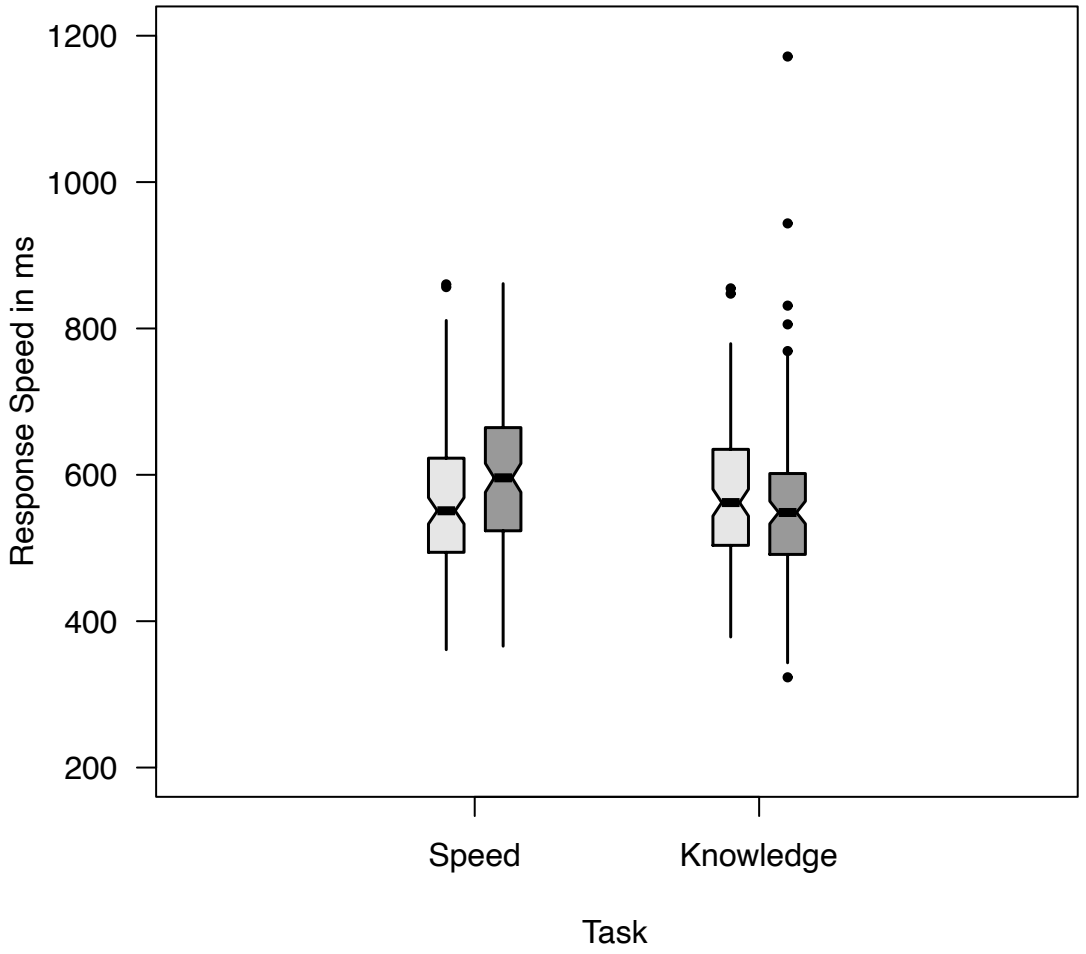
Information to be provided in figures
and information that can be omitted

Descriptive plots

Always visualize the distributional characteristics of your (main) data, as readers cannot infer them easily from means, SDs, skew and kurtosis

Boxplots will sometimes (but not always) do, but the basic R function will produce ugly and in part uninformative output

You can polish it and add notches (giving an impression of significant differences between variables), but still, condition differences are less obvious as compared to this plot



Information to be provided in figures
and information that can be omitted

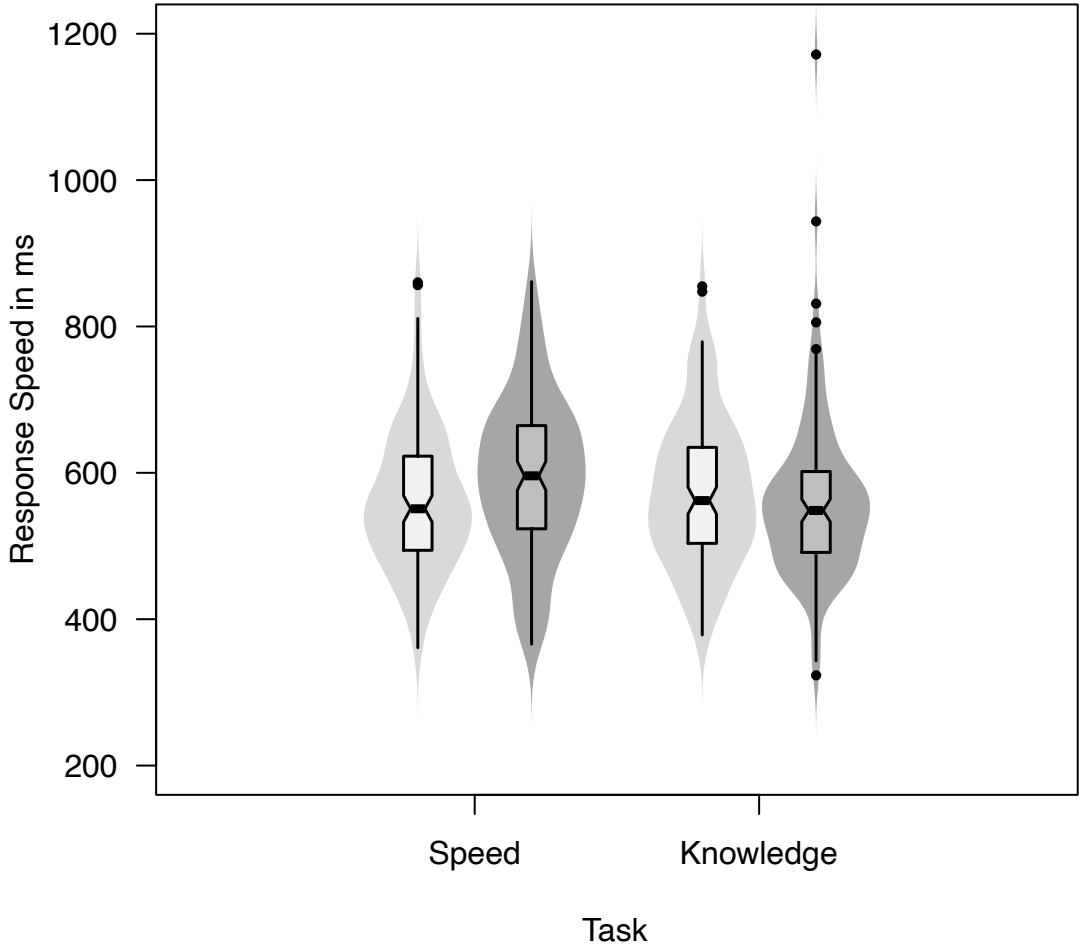
Descriptive plots

Always visualize the distributional characteristics of your (main) data, as readers cannot infer them easily from means, SDs, skew and kurtosis

Boxplots will sometimes (but not always) do, but the basic R function will produce ugly and in part uninformative output

You can polish it and add notches (giving an impression of significant differences between variables), but still, condition differences are less obvious as compared to this plot

Even better, you combine violin and boxplots!



Information to be provided in figures
and information that can be omitted

Descriptive plots

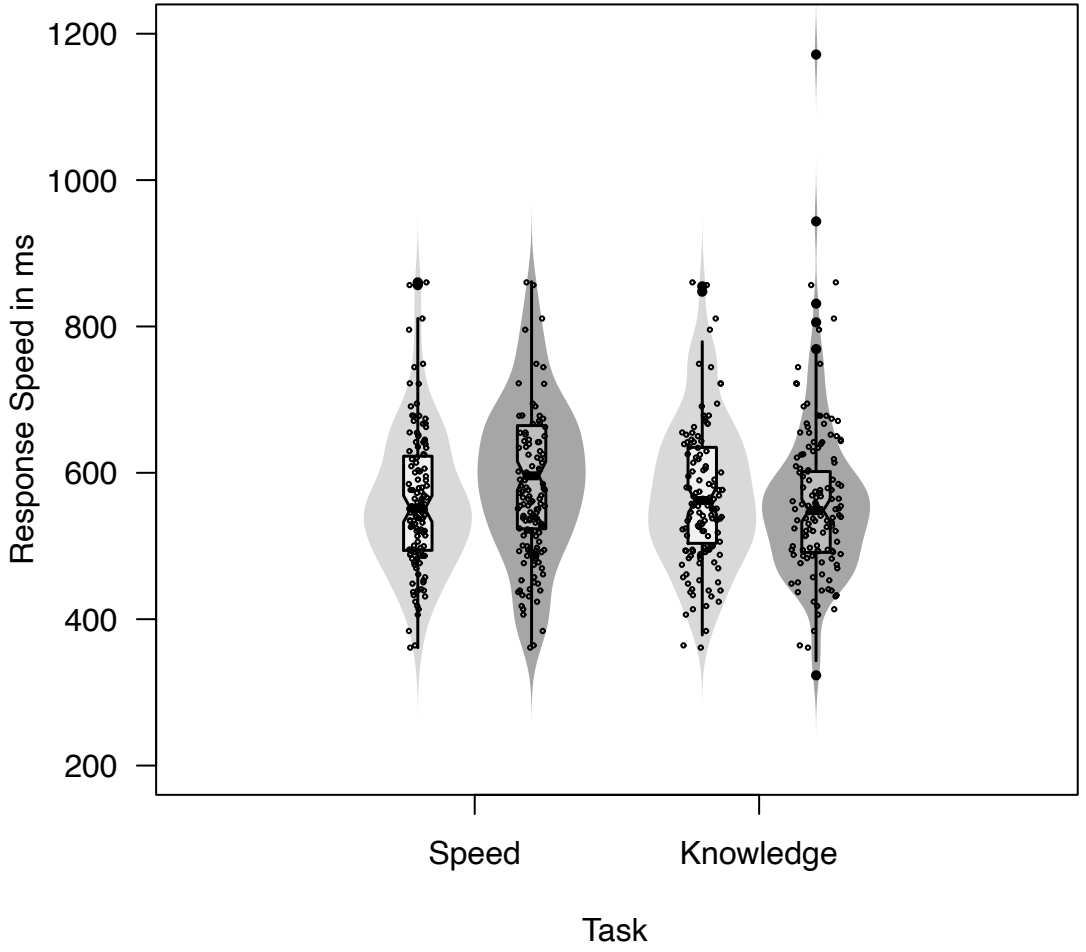
Always visualize the distributional characteristics of your (main) data, as readers cannot infer them easily from means, SDs, skew and kurtosis

Boxplots will sometimes (but not always) do, but the basic R function will produce ugly and in part uninformative output

You can polish it and add notches (giving an impression of significant differences between variables), but still, condition differences are less obvious as compared to this plot

Even better, you combine violin and boxplots!

Individual data points are not always a good idea

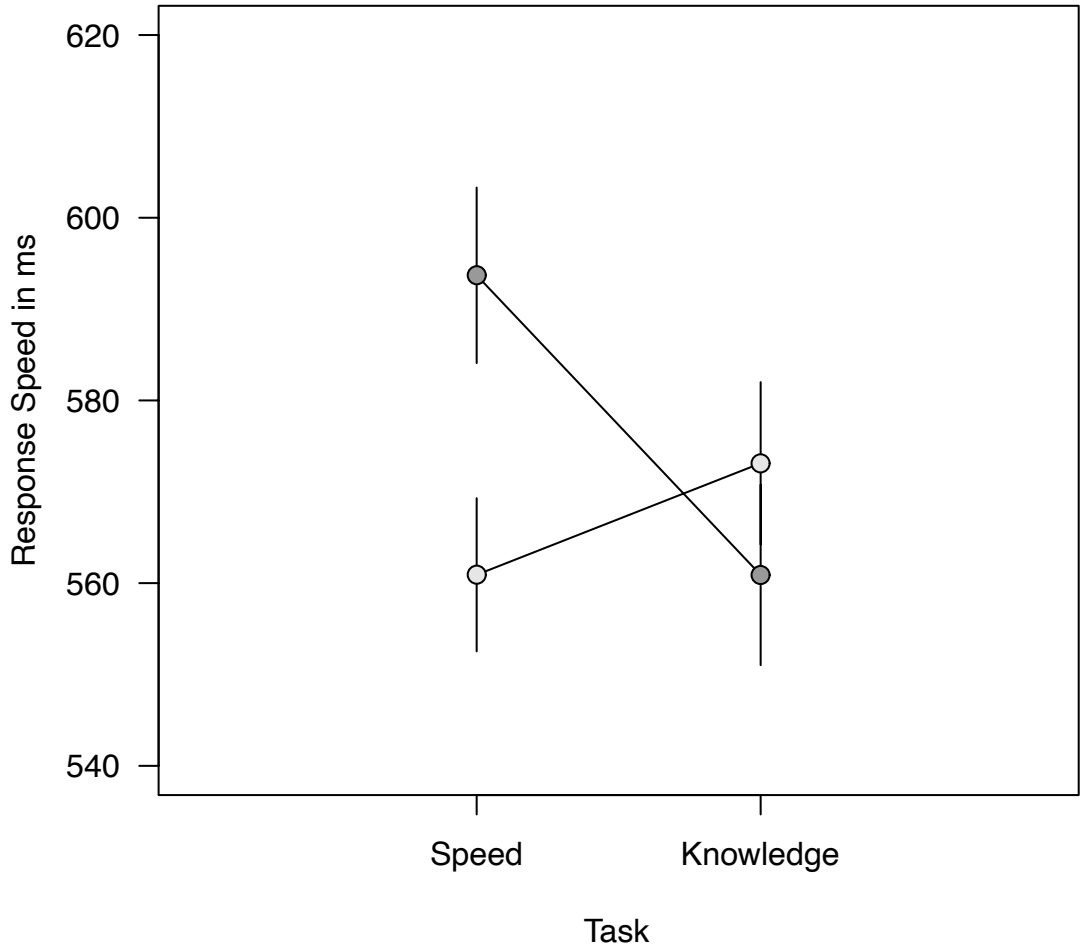


Information to be provided in figures
and information that can be omitted

Plots illustrating condition differences

Always give error bars!

Most common have been standard errors (SE). Yet, they overlap on the right handside, but how much exactly (adding horizontal lines at the end of the error bars may help but does not always look nice) ...



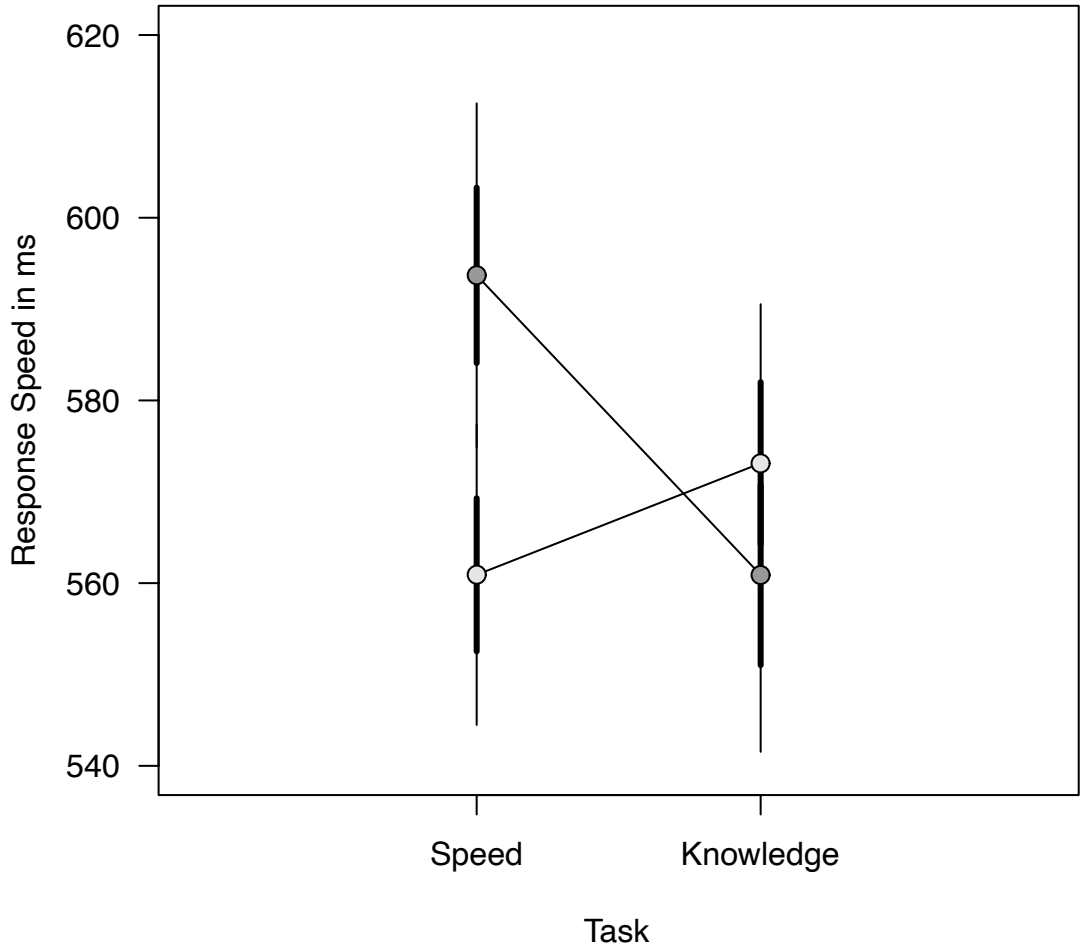
Information to be provided in figures
and information that can be omitted

Plots illustrating condition differences

Always give error bars!

Most common have been standard errors (SE). Yet, they overlap on the right handside, but how much exactly (adding horizontal lines at the end of the error bars may help but does not always look nice) …

Moreover, using (or adding) 95% confidence intervals (CI) of the mean is more informative (and more cautious)



Information to be provided in figures
and information that can be omitted

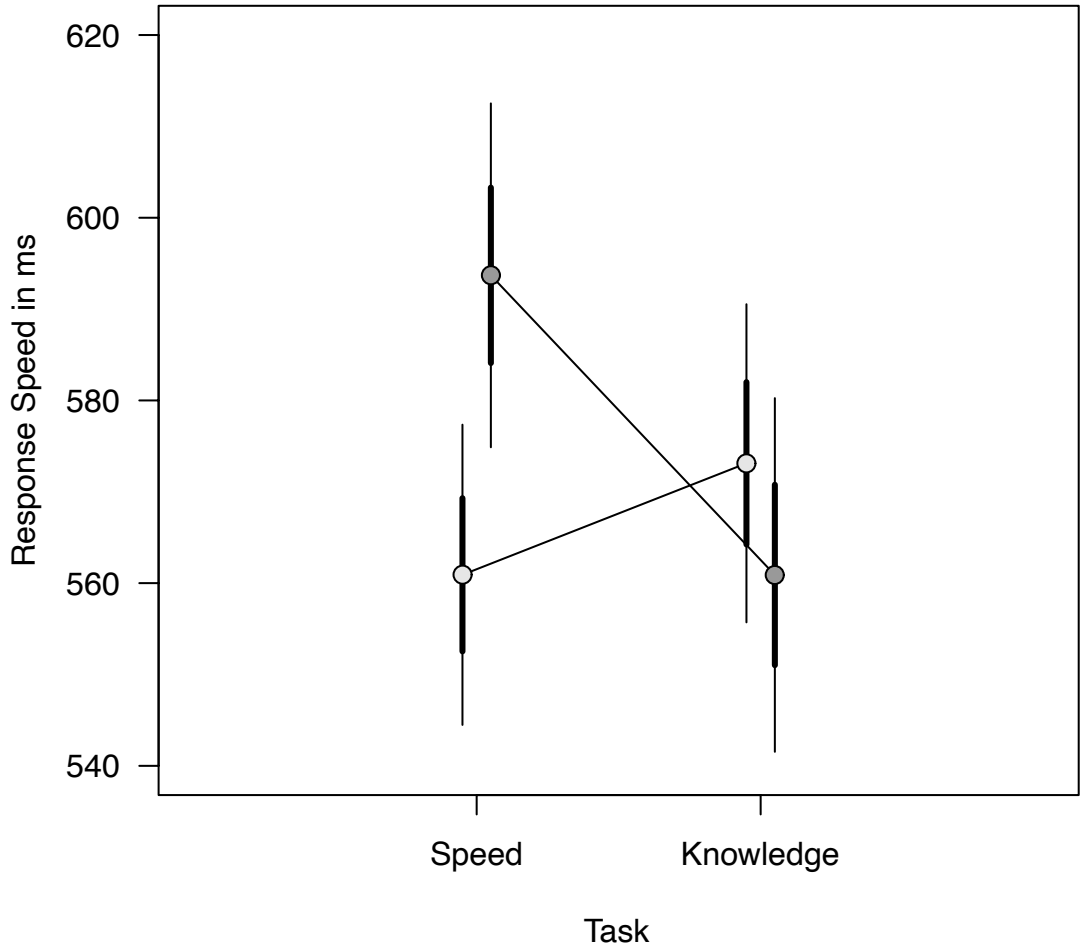
Plots illustrating condition differences

Always give error bars!

Most common have been standard errors (SE). Yet, they overlap on the right handside, but how much exactly (adding horizontal lines at the end of the error bars may help but does not always look nice) ...

Moreover, using (or adding) 95% confidence intervals (CI) of the mean is more informative (and more cautious)

To avoid overlap, you could use jittered plots



Information to be provided in figures
and information that can be omitted

Plots illustrating condition differences

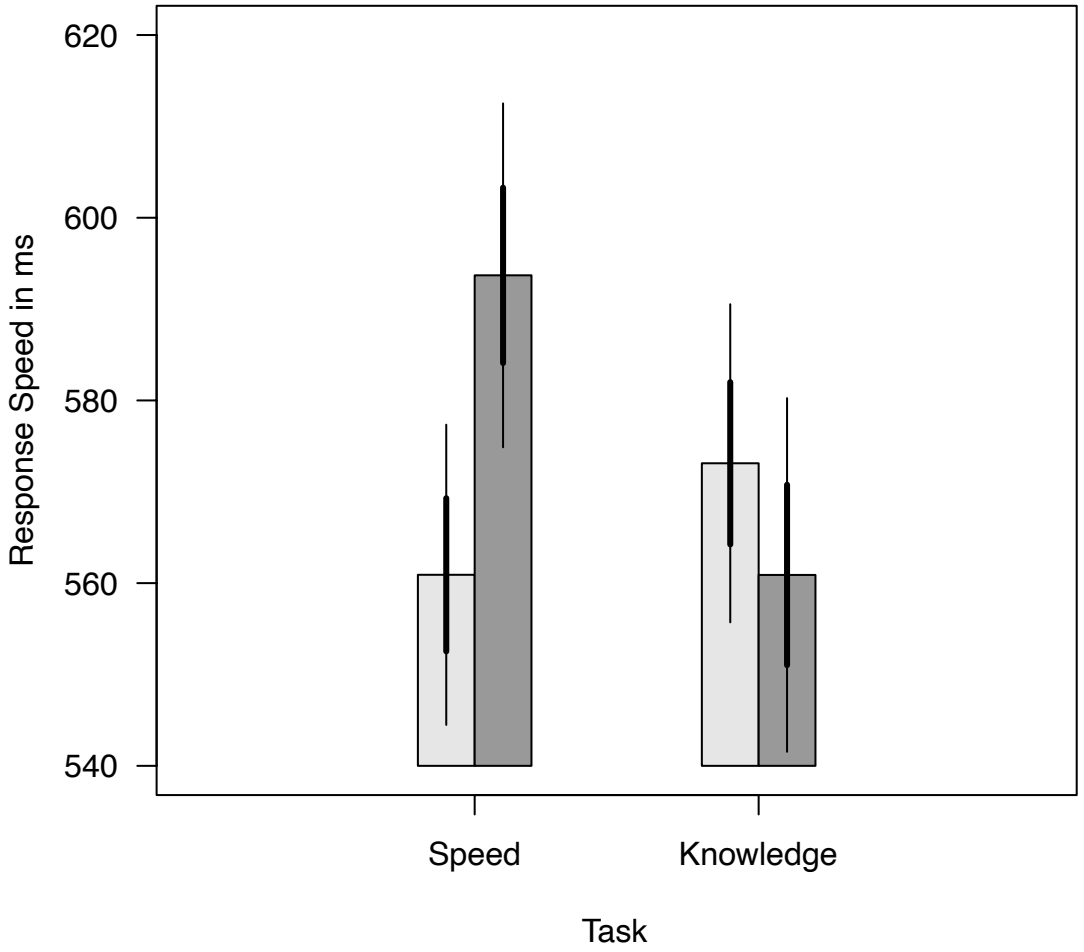
Always give error bars!

Most common have been standard errors (SE). Yet, they overlap on the right handside, but how much exactly (adding horizontal lines at the end of the error bars may help but does not always look nice) …

Moreover, using (or adding) 95% confidence intervals (CI) of the mean is more informative (and more cautious)

To avoid overlap, you could use jittered plots

Or you might want to consider bar plots



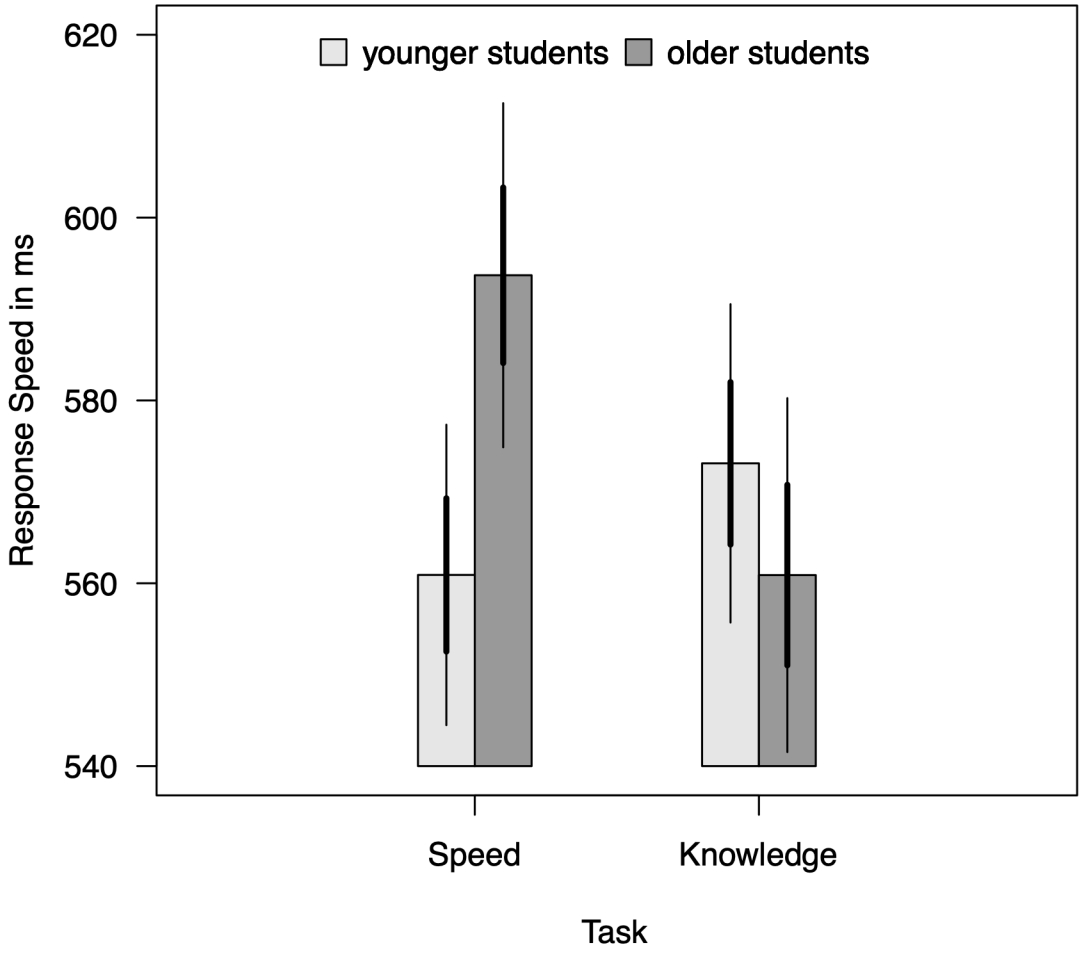
Information to be provided in figures
and information that can be omitted

Plots illustrating condition differences

Always provide a legend and give anything else to understand the plot in the caption!

Example caption:

Figure 2. Differences in response speed in milliseconds depending on task and age. Younger students (light grey bars) respond considerably faster in the speed task than older students (dark grey bars), while older students respond somewhat faster in the knowledge task. Error bars are standard errors of the means (thick lines) and 95% confidence intervals of the means (thin lines).



Information to be provided in figures
and information that can be omitted

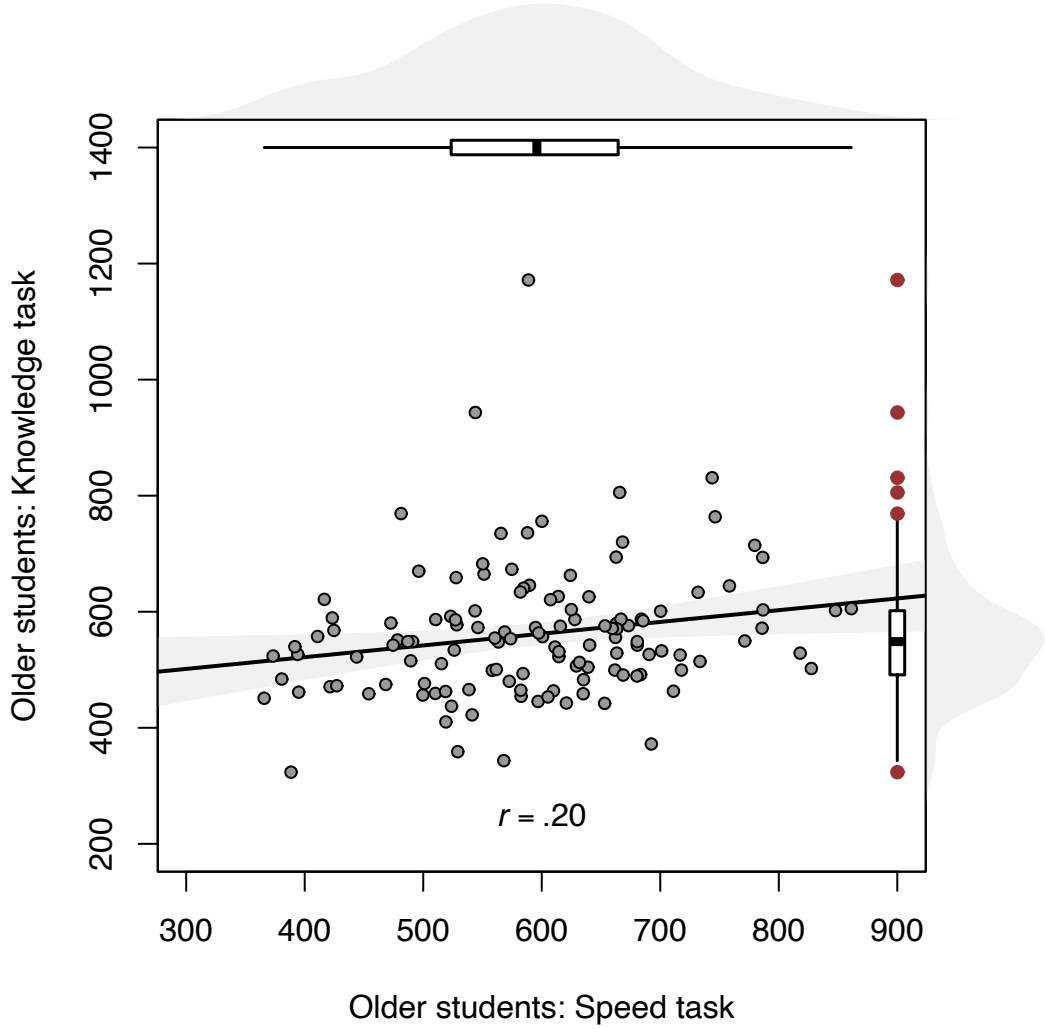
Further tips

Consider whether putting too much information into one plot is a good idea.

In the example on the right, it may work, but consider whether two separate plots or panels – one for the distribution and and one for the correlation – would be better

Consider using colors (if you do not have to pay color page charges or are well funded to pay them)

But, as said, make sure that differently colored plot elements remain distinguishable (in b/w print-outs, for the color-blind)



Fine-tuning
Design, layout and creation

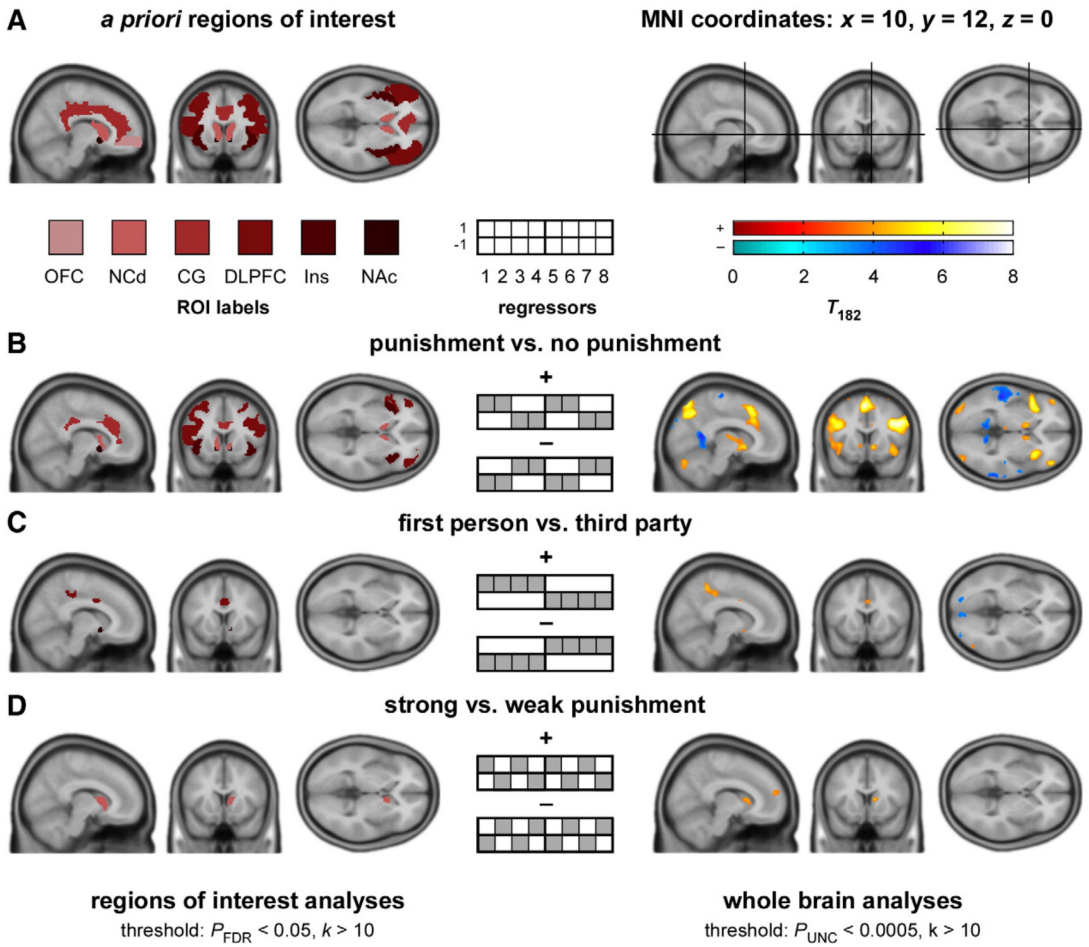
Fine-tuning
Design

Do not use standard figure output!

For the figure on the right, I used the standard SPM output, but then

- first tried to find the best coordinates to illustrate the effects of interest (top right)
- removed the black part surrounding the scans and somewhat blurred the borders of the skull via Photoshop
- loaded the raster images to some vector graphics software
- added vector graphics of the conditions of interest (middle) and text labels
- finally aligned all figure parts properly

Here, b/w printout will not work well, though



Strobel, A. et al. (2011). *NeuroImage, 54*, 671-680.

Fine-tuning
Layout

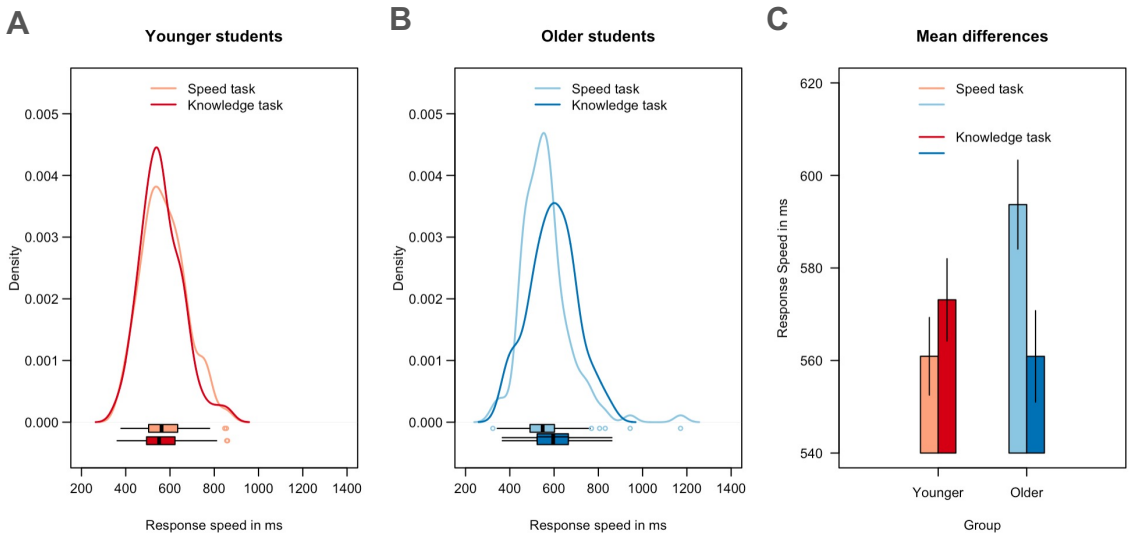
Use one multi-panel figure instead of several single panel figures when it best illustrates a certain result!

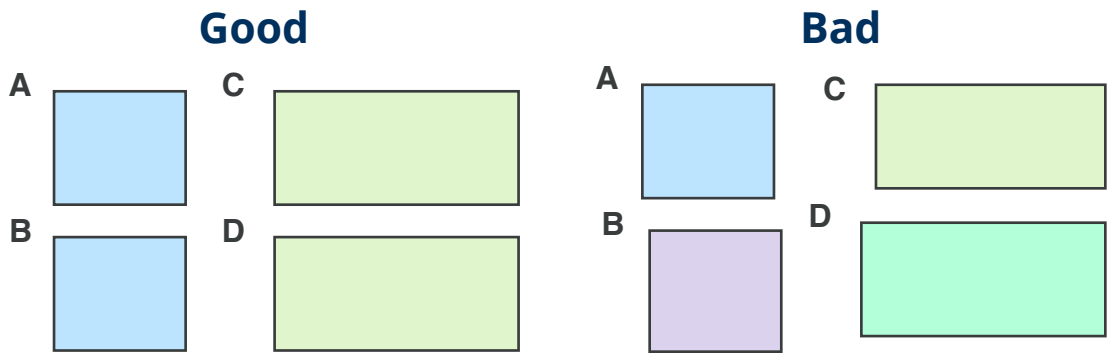
Otherwise, the journal might spread your figures as it fits best to their production regime and may place related figures too far apart

Yet, do not overload multipanel figures!

Aling and size the panels and other element of your figure properly!

Also, for variables, conditions etc. that remain the same thoughout your manuscript, always use the same colors

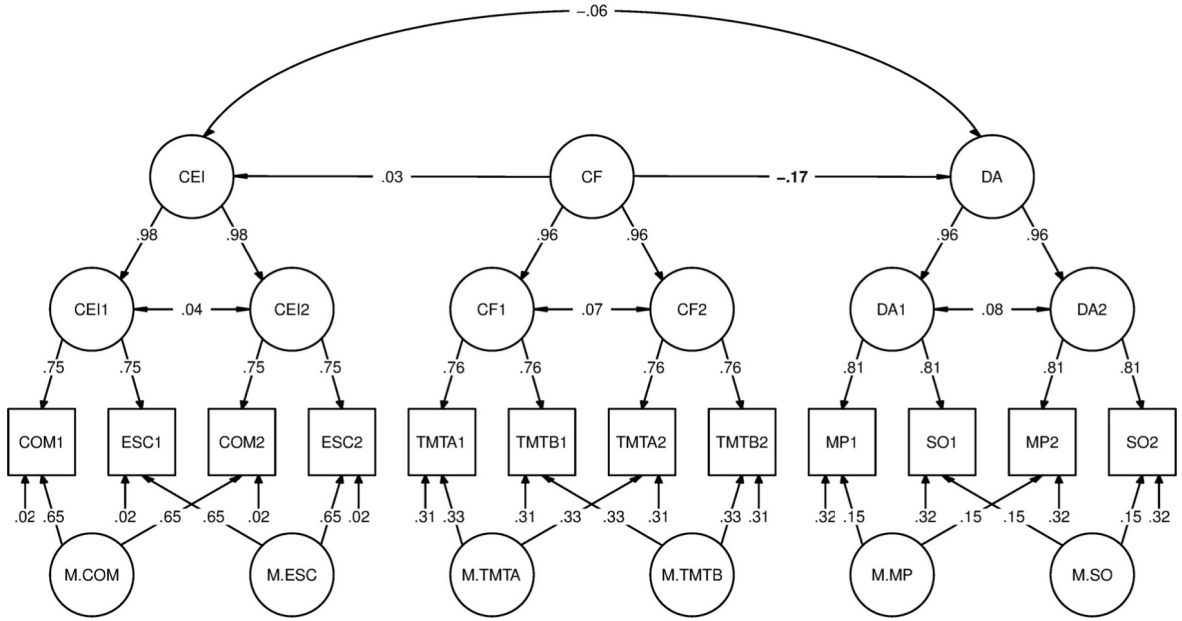




Fine-tuning
Creation

If your figure contains coefficients, or may change during review, prepare updateable figures, e.g., via R!

The figure on the right was generated with R and can be updated if something changes. It would be very inconvenient to edit it by hand when you have to change sample size (because R1 wants you to drop multivariate outliers) or model settings (because R2 requires to use MLR instead of ML estimation)



Strobel, A. et al. (2020). *PLoS ONE*, 15(10). e0239817

Summary and outlook

Summary
What you should always keep in mind …

Prefer vector over raster graphics!

If you need to combine vector and raster graphics, save them in high resolution (≥ 300 dpi, format depending on the journal). In doubt, use 1000 dpi *.tif images with LZW compression

Get yourself the right software tools!

One for handling vector and one for handling raster graphics. Ideally, they come from the same developer, will be maintained in the foreseeable future, are free and open source or at least do not come with a licence that has to be renewed annually

Consult the relevant artwork guidelines!

If in doubt, use RGB color space, 90 mm width for smaller figures that fit on in column (if the journal uses columns anyway) or 180 mm width for larger figures, and serif font (ideally Helvetica)

Think twice when it comes to point and line style and color!

Point style should meaningful, lines solid, and different colors should remain distinguishable for everyone under every circumstance

Provide all necessary information, but not all possible information!

Outlook
Things not covered here

R packages for plotting

Base R (i.e., the *graphics* package) comes with great capabilities for graphics, but sometimes, it may be a bit cumbersome to arrive at what you want (the left figure needs 4-6 lines of code)

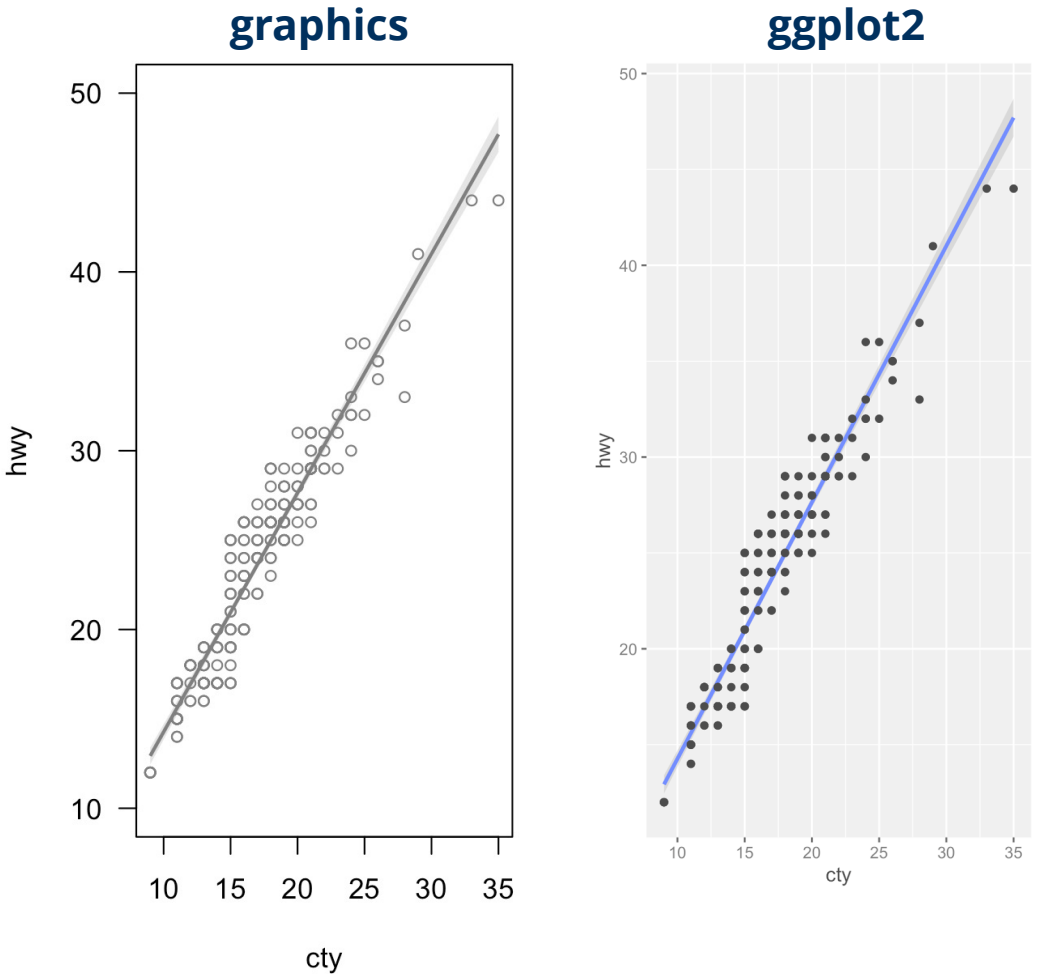
Dedicated packages such as *ggplot2* let you arrive at the same with perhaps 3 code lines (right figure)

Yet, learning to master *ggplot2* really takes time …

Handling software to edit plots

Learning to work with Adobe or Affinity software (and even more so with free alternatives) can be tiresome as well

So better start right now …



Find out more about *ggplot2* at https://ggplot2.tidyverse.org/index.html

Thank you!

And now you might want to discuss …

The present workshop

— Did you learn anything that was new to you?
— If so, what, and what was most helpful?
— What would be most helpful to learn in a possible follow-up workshop?

Your own experience

— What graphics tools have you used so far and with what success?
— Which ones can you recommend, which ones would you discourage to use?
— Is anyone among you an expert for a certain graphics tool and can support others in their handling?

Is putting so much effort into graphics really worth the time?