SOFTWARE

DPI AND PPI IN SCIENTIFIC PUBLICATIONS

6.02.2023

It seems that also graphic design tasks are more and more shifted to the authors. A journal asked me to revise my R “figures with >6 px size fonts at 600 dpi” which does not make so much sense to me as it mixes apples and oranges (DPI and PPI) and does even give me any size estimate of the final image.

Fortunately, there is a great website that has a lot of helpful explanations and demo code.

In summary PPI are number of pixels per inch. Pixel is the smallest unit that can be displayed on a screen, the picture x element. So my screen has about 227 ppi which is super sharp while for historical reasons not the physical but a logical PPI of 96 is being assumed for screens.

A single pixel at 96 DPI is equal to 0.2645835 millimeters, 0.010416675 inches or 0.75 points – the typographic unit also in use by R ggplot.

DPI in contrast describes the number of dots per inch eg the printing resolution of a hard copy print dot – something different as the DPI measurement of a printer needs to be considerably higher than the pixels per inch (PPI) measurement of a video display in order to produce similar quality output.

> For example, a bitmap image may measure 1,000 × 1,000 pixels, a resolution of 1 megapixel. If it is labelled as 250 PPI, that is an instruction to the printer to print it at a size of 4 × 4 inches. Changing the PPI to 100 in an image editing program would tell the printer to print it at a size of 10 × 10 inches. However, changing the PPI value would not change the size of the image in pixels which would still be 1,000 × 1,000. An image may also be resampled to change the number of pixels and therefore the size or resolution of the image, but this is quite different from simply setting a new PPI for the file.

Weird? Not so far.

> The plot dimension is converted in inches before saving to disk, using an assumed screen DPI of 96 to make the conversion (2560 / 96) x (1600 / 96) = 26.6 x 16.6. Then ggsave uses this dimension in inches with the DPI of 300 to save it to disk, creating an image of (26.6 * 300) x (16.6 * 300) = 8000 x 5000 dots.

More general

```
# produces 1800px x 1800px image
png(width=3,height=3,units="in",res=600)
```

The difficulty comes when changing sizes is leading to different title and image text size as they seem to change at a different scale eg at points (pts, title text) and and mm/inch (geom_text). So this is getting really weird now as changing font siz let symbols and lines shrink.

So I would recommend to produce something visually appealing on your screen. For saving a PNG or printing we need another library that does the heavy size lifting

```
library(ragg)
ragg::agg_png("ragg_20x20.png", width = 20, height = 20, units = "in",
res = 300, scaling = 2)
plot(p)
dev.off()
```

Saving the plot is also straightforward

```
ggsave( pngfile, p,  device = agg_png, width = 10, height = 6, units = "cm", res = 300, scaling = 0.5)
```

HTH