

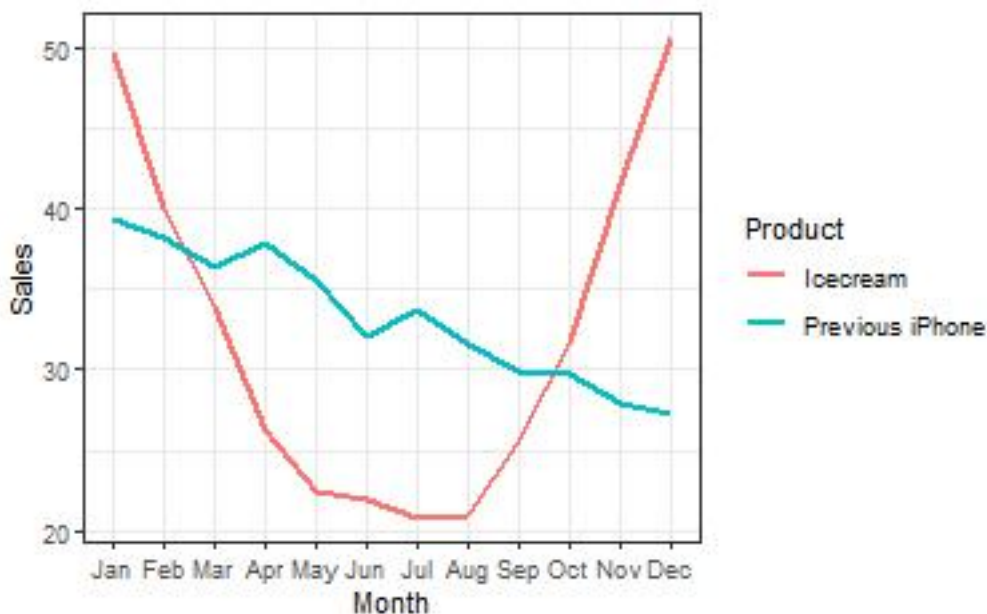
PDF example for illustrating the importance of figure format

Sean van der Merwe

2025-08-18

jpegs are for photos, not plots

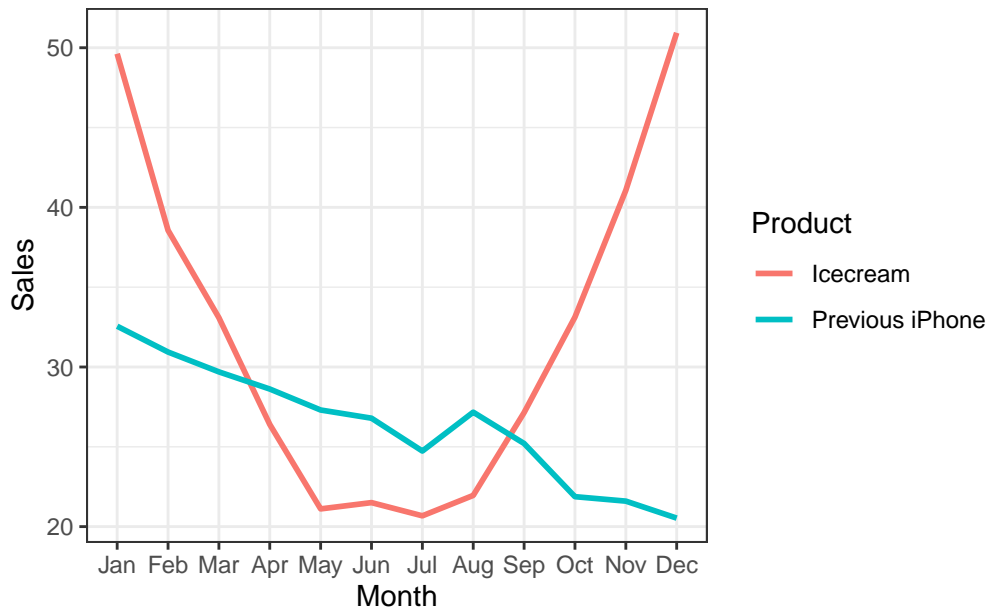
```
data.frame(Month = factor(rep(month.abb, 2), levels = month.abb),  
           Sales = c((((1:12)-6.5)^2 + 20, 39:28) + rnorm(24)),  
           Product = rep(c("Icecream", "Previous iPhone"), each = 12)) |>  
ggplot(aes(x = Month, y = Sales, colour = Product, group = Product)) +  
geom_line(linewidth = 1) + theme_bw()
```



Zoom in to maximum and squint at the blur. This is because we are storing vector information in a raster format.

Vector graphics are the answer

```
data.frame(Month = factor(rep(month.abb, 2), levels = month.abb),
           Sales = c(((1:12)-6.5)^2 + 20, 32:21) + rnorm(24),
           Product = rep(c("Icecream", "Previous iPhone"), each = 12)) |>
ggplot(aes(x = Month, y = Sales, colour = Product, group = Product)) +
geom_line(linewidth = 1) + theme_bw()
```



Now zoom in to see crisp and clear text. This is because the vectors are stored in a vector format.

So what are the best formats?

For PDF use PDF, EPS, or SVG.

For Word use EMF, WMF, or SVG.

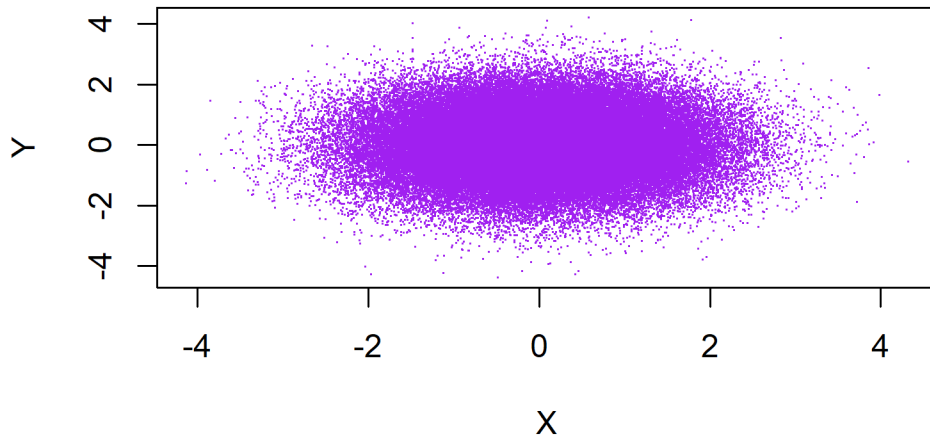
For HTML (web) use SVG or JS (*e.g. plotly*).

Exceptions?

Yes, there are always exceptions.

If your plot has 100,000+ dots then a vector format will start to cause you problems. Here it is better to rasterize. Be sure to use a high enough resolution for clear text **and use a lossless compression format like PNG** (not JPG). You can increase the DPI for more detail.

```
n <- 100000
plot(rnorm(n), rnorm(n), main="", xlab = "X", ylab = "Y", col = "purple", pch = ".")
```



With 10,000 points vector graphics is probably better. Your plot and situation must get really crazy before it makes sense to rasterize (like if you are planning on airbrushing it or overlaying it on a photo).

SVG is the way to go in most cases.