High Resolution Figures in R

By **Daniel Hocking** / March 12, 2013

Converted to *pdf* from his blog.

I realized the default resolution of R and RStudio images are insufficient for publication. PLOS ONE requires 300 ppi images in TIFF or EPS (encapsulated postscript) format. In R plots are exported at 72 ppi by default. I love RStudio but was disappointed to find that there was no options for exporting figures at high resolution.

PLOS ONE has extensive instructions for scaling, compressing, and converting image files to meet their standards. Unfortunately, there is no good way to go from low resolution to high resolution (i.e. rescaling in Photoshop) as my friend Liam Revell, and phytools author, pointed out with entertaining illustration from PhD comics (upper right panel). The point is if the original image isn't created at a high enough resolution, then using Photoshop to artificially increase the resolution has problems of graininess because Photoshop can only guess how to interpolate between the points. This might not be a big problem with simple plots created in R because interpolation between points in a line shouldn't be difficult, particularly when starting with a PDF.

Even if scaling up from a low resolution PDF would work, it would be better to have a direct solution in R.

[UPDATE: In the original post, I wrote about my trial and error when first trying this out. It was helpful for me but less helpful for others. Since this post gets a fair amount of traffic, I now have the solution first and the rest of the original post below that]

If you need to use a font not included in R, such as the Arial family of fonts for a publisher like PLOS, the extrafont package is very useful but takes a long time to run (but should only have to run once – except maybe when you update R you'll have to do it again).

```
install.packages("extrafont")
library(extrafont)
font_import() # this gets fonts installed anywhere on your computer, most commonly
from MS Office install fonts. It takes a LONG while.
```

Now to make a nice looking plot with a higher resolution postscript is the best but you may need to download another program to view it on you computer. TIFF is good for Raster data including photos and color gradients. PDF should be good for line and point based plots. JPEG is not going to be good for high resolution figures due to compression and detail loss but is easy to view and use for presentations, and PNG is lower quality that is useful for websites. Here are some high resolution figure examples. They all start by telling R to open a connection (device) to make a PDF or TIFF or EPS rather than just print to the R/RStudio plot default device. Then making the plot, then closing the device, at which point the file is saved. It won't show up on the screen but will be saved to your working directory.

```
x = 1:20
y = x * 2

setwd('/Users/.../Folder/') # place to save the file - can be over-ridden by
```

```
putting a path in the
                              file = " " part of the functions below.

pdf(file = "FileName.pdf", width = 12, height = 17, family = "Helvetica") #
defaults to 7 x 7 inches
plot(x, y, type = "l")
dev.off()

postscript("FileName.eps", width = 12, height = 17, horizontal = FALSE,
           onefile = FALSE, paper = "special", colormodel = "cmyk",
           family = "Courier")
plot()
dev.off()

bitmap("FileName.tiff", height = 12, width = 17, units = 'cm',
       type = "tifflzw", res = 300)
plot()
dev.off()

tiff("FileName.tiff", height = 12, width = 17, units = 'cm',
     compression = "lzw", res = 300)
plot()
dev.off()
```

[ORIGINAL POST Follows]

It took some time to figure out but here are some trials and the ultimate solution I came up with:

```
x <- 1:100
y <- 1:100
plot(x, y) # Make plot
tiff("Plot1.tiff")
dev.off()
```

Nothing happens in this case. However, by setting up the tiff file first, then making the plot, the resulting TIFF file is saved to your working directory and is 924 KB, 72 ppi, 480 x 480 pixels.

To increase the resolution I tried the following:

```
tiff("Plot2.tif", res = 300)
plot(x, y) # Make plot
dev.off()
```

but in RStudio the plot could not be printed and hence not saved because it was too large for the print area. Therefore, I had to open up R directly and run the code. Interestingly, a blank TIFF file was created of the same size as Plot1.tiff. This is where I got hung up for a while. I eventually found that R can't figure out the other image parameters when resolution changes or because the default is too big to print, so they have to be specified directly as such:

```r
tiff("Plot3.tiff", width = 4, height = 4, units = 'in', res = 300)
plot(x, y) # Make plot
dev.off()
```

This creates a TIFF file that is 5,800 KB, 300 ppi, 4 inches by 4 inches. Surprisingly, on a Mac it still indicates that it's only 72 ppi when viewed in Preview. The larger size indicates that it is actually 300 ppi. I ran the same code but specifically specified res = 72 and the file was only 334 KB, suggesting that Preview is incorrect and the file is really 300 ppi. I played with various compressions but lzw and none were the same while rle resulted in a larger file (less compression). That seems odd again.

Finally, I tried using the bitmap function to create a TIFF:

```r
bitmap("Plot7.tiff", height = 4, width = 4, units = 'in', type="tifflzw", res=300)
plot(x, y)
dev.off()
par(mfrow = c(1,1))
```

Interestingly, this file is only 9 KB but is listed as 300 dpi, 1200 x 1200 pixels. I'm really not sure why these functions don't seem to be working as smoothly as expected but hopefully this can help get high resolution images directly from R for publication. I plan to use the bitmap function in the future to create high resolution TIFF files for publication. This is what is desired by outlets such as PLOS ONE. It's easier than dealing with postscript. I also don't know if EPS files from R or RStudio are created with LaTeX. I know that can be a problem for PLOS ONE.

[UPDATE: Here is the code for my final figure for PLOS ONE with both postscript and tiff options plus it uses the extrafonts package to allow Arial fonts in postscript figures as required by PLOS ONE]

```r
install.packages("extrafont")
library(extrafont)
font_import() # this gets fonts installed anywhere on your computer,
# most commonly from MS Office install fonts.
# It takes a long while.

bitmap("CummulativeCaptures.tiff", height = 12, width = 17,
units = 'cm', type="tifflzw", res=300)
postscript("CummulativeCaptures.eps", width = 8, height = 8,
horizontal = FALSE, onefile = FALSE, paper = "special",
colormodel = "cmyk", family = "Arial")
par(mar = c(3.5, 3.5, 1, 1), mgp = c(2, 0.7, 0), tck = -0.01)
plot(refmCumm$date, refmCumm$cumm, type = "n", xaxt = "n",
xlab = "Date",
ylab = "Cumulative number of salamanders per plot",
xlim = c(r[1], r[2]),
ylim = ylims)
axis.POSIXct(1, at = seq(r[1], r[2], by = "year"), format = "%b %Y")
lines(refmCumm$date, refmCumm$cumm, lty = 1, pch = 19)
lines(depmCumm$date, depmCumm$cumm, lty = 2, pch = 24)
```

```
arrows(refmCumm$date, refmCumm$cumm+refmCumm$se, refmCumm$date, refmCumm$cumm-
refmCumm$se, angle=90, code=3, length=0)
arrows(depmCumm$date, depmCumm$cumm+depmCumm$se, depmCumm$date, depmCumm$cumm-
depmCumm$se, angle=90, code=3, length=0)
dev.off()
```

EDIT: Just found a nice blog post with recommendations on device outputs on Revolutions here

Below is all the code that includes comparison of sizes with PDF, PNG, JPEG, and EPS files as well.

```
plot(x, y) # Make plot
tiff("Plot1.tiff")
dev.off()
```

```
tiff("Plot2.tiff", res = 300)
plot(x, y) # Make plot
dev.off()
```

```
tiff("Plot3.tiff", width = 4, height = 4, units = 'in', res = 300)
plot(x, y) # Make plot
dev.off()
```

```
tiff("Plot3.72.tiff", width = 4, height = 4, units = 'in', res = 72)
plot(x, y) # Make plot
dev.off()
```

```
tiff("Plot4.tiff", width = 4, height = 4, units = 'in', res = 300, compression =
'lzw')
plot(x, y) # Make plot
dev.off()
```

```
tiff("Plot5.tiff", width = 4, height = 4, units = 'in', res = 300, compression =
'none')
plot(x, y) # Make plot
dev.off()
```

```
tiff("Plot6.tiff", width = 4, height = 4, units = 'in', res = 300, compression =
'rle')
```

```
 # Make plot
dev.off()
```

```
bitmap("Plot7.tiff", height = 4, width = 4, units = 'in', type="tifflzw", res=300)
plot(x, y)
dev.off()
par(mfrow = c(1,1))
```

```
bitmap("Plot8.tiff", height = 480, width = 480, type="tifflzw", res=300)
plot(x, y)
dev.off()
par(mfrow = c(1,1))
```

```
jpeg("Plot3.jpeg", width = 4, height = 4, units = 'in', res = 300)
plot(x, y) # Make plot
dev.off()
```

```
tiff("Plot4b.tiff", width = 4, height = 4, pointsize = 1/300, units = 'in', res =
300)
plot(x, y) # Make plot
dev.off()
```

```
png("Plot3.png", width = 4, height = 4, units = 'in', res = 300)
plot(x, y) # Make plot
dev.off()
```

```
pdf("Plot3.pdf", width = 4, height = 4)
plot(x, y) # Make plot
dev.off()
```

```
postscript("Plot3.eps", width = 480, height = 480)
plot(x, y) # Make plot
dev.off()
```