

# Notes about exporting plots from R.

**Use a special device** Do not save your plots from the graphics window: use a special device such as `pdf()` or `png()`. This will give you better control over the text size and the shape of the plot. Do not just use square plots because that is the default: think about what is appropriate for your particular plot.

**Reviewing plots** If using  $\LaTeX$ , always check how your plots look in Adobe Reader *not* Mac Preview or TexWorks. The latter may not even be able to display the plot.

**pdf is usually best** Use pdf plots unless you are confident that jpeg or png give as crisp text and as smooth lines in a particular case.

**semi-transparency** You may have printing problems if you use semi-transparent colours with large amounts of data. (Adobe tries to split the image up into non-overlapping areas of constant colour, and fails if there are too many points.) In this case you should be able to print the page by, in the print dialog, clicking on Advanced and then Print As Image. This will degrade quality slightly, so only do it if it is necessary.

**Word users** Those of you using Word can insert pdfs by using Insert/Object...

**Text size** The text on a plot should be approximately the same as the text around it.

Below are three plots, based on an example on the help page for `boxplot`. The first plot used the example code unchanged and was saved to pdf from the graphics screen. The second used a pdf device 1.5 times the size required in  $\LaTeX$ , with the default pointsize. *This is the recommended method.*

Lattice plots do not respond to changes in the pointsize in the pdf device, and the default size is different from that for base plots. You can alter the fontsize using lattice parameters, either by creating the device using `trellis.device`, or more easily, adding options to the plot. The equivalent to the `boxplot` is shown: an exercise just for comparison, not recommended. (I failed to get the legend to work with a pdf device.) The R code used for the lattice plot is given below. The plot used a pdf device height 4 and width 6: the default pointsize in lattice is 10, so it needs less shrinking. The actual plot code is not completely shown, as I had to amend `panel.bwplot` to widen the lines in the boxes and make the ends of the bars solid. The other changes were made via parameters, and are shown. Setting the aspect ratio to 0.5 was done to make a more pleasing result: without this I suspect the ratio of sizes required would be different. There are also a few added comments about changing font size in lattice plots.

Experiment with the dimensions and pointsize to get the best result.

And do not forget the `dev.off!`

All plots were sized to a height of 3 inches in  $\LaTeX$  using:

```
\begin{figure}[h]
  \begin{center}
    \includegraphics[height=3in]{Rplots....pdf}
  \end{center}
\end{figure}
```

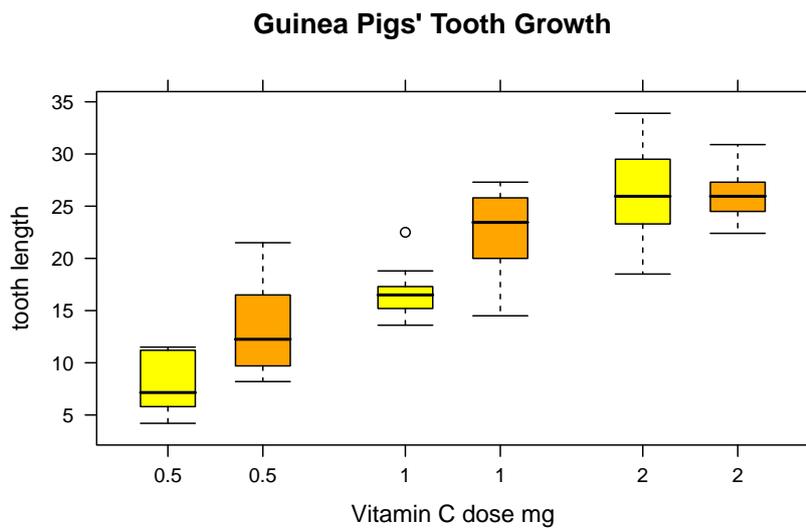
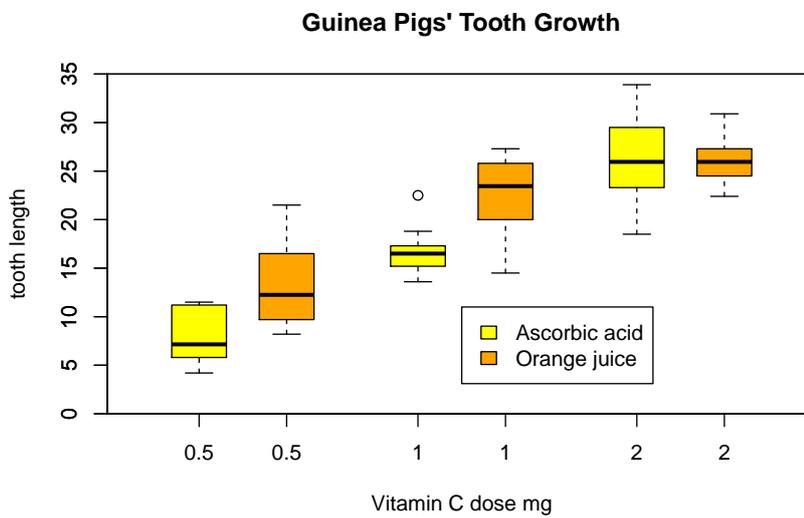
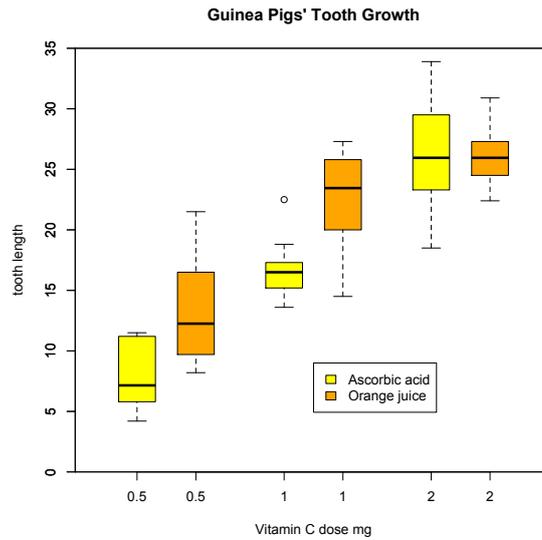


Figure 1: Upper: Screen plot saved to file. Centre: height=4.5, width=6.75, pointsize 12 (default). Lower: height=4, width=6, aspect ratio 0.5 (lattice)

```

mypan <- panel.bwplot ## need to edit this - for experts only

## create a variable and recode it
mydose <- ifelse(ToothGrowth$supp == "VC", ToothGrowth$dose - 0.2,
  ToothGrowth$dose + 0.2)
library(car)
mydose1 <- recode(mydose,
  "0.3=0.8; 0.7=1.2; 0.8=1.8; 1.2=2.2; 1.8=2.8; 2.2=3.2")

## open the device
pdf(file="lat45675a.pdf", height=4, width=6)

ops <- trellis.par.get()
ops1 <- ops
ops1$box.rectangle$col <- 'black'
ops1$box.umbrella$col <- 'black'
ops1$plot.symbol$col <- 'black'
trellis.par.set(ops1)

xyplot(len ~ mydose1 , groups=supp, data=ToothGrowth,
  horizontal=FALSE, fill=c("yellow","orange"), aspect=0.5,
  scales=list(x=list(labels=rep(c(0.5, 1, 2), each=2),
    at=sort(unique(mydose1)))), pch="|",
  panel= mypan,box.ratio=0.3, xlim=c(0.5,3.5),
  main = "Guinea Pigs' Tooth Growth",
  xlab = "Vitamin C dose mg",
  ylab = "tooth length")
dev.off()

## if you have to change the text size on the plot,
## use the argument par.settings eg
par.settings=list(fontsize=list(text=44, points=1))
## in the call to xyplot. (Extreme values so you can see
## the result!)
## Alternatively, use trellis.device instead of pdf:
trellis.device(pdf, file = "something.pdf",
  theme = list(fontsize =
  list(text = 18, points = 6))

```