Prof. David Draper
Department of Statistics
University of California, Santa Cruz

# STAT 206: Getting PDF files into LaTeX documents

In Take-Home Test 1, I've asked you to make some plots and include them in your solutions document. If you're using LaTeX to create your documents in this class, here's how to bring plots, created by an environment such as `R`, into LaTeX.

Create your plot in PDF–c format[1] and put it in the directory where your solutions document lives. As a simple example, below please find some `R` code to simulate $M = 1{,}000$ IID draws from the Exponential distribution (Take-Home Test 1 problem 2(B)) with parameter $\lambda = 2$, superimpose the Exponential(2) density function on a histogram of the random draws, and export a **PDF–c** file to the directory in which the `R` session occurred. Below I call the directory containing the LaTeX document that imports the **PDF–c** file `.../PDF-Importation` .

With the Gamma$(\alpha, \beta)$ distributions, of which the Exponential$(\lambda)$ is the special case Gamma$(1, \lambda)$, there are two popular parameterizations, and it's important to be clear on which one your software is using. In Take-Home Test 1 problem 2(B) I use the following Exponential$(\lambda)$ parameterization for $\lambda > 0$:

$$(y_i \,|\, \lambda \, \boldsymbol{E} \, \mathcal{B}) \overset{\text{IID}}{\sim} \text{Exponential}(\lambda): \quad \text{i.e.,} \;\; p(y_i \,|\, \lambda \, \boldsymbol{E} \, \mathcal{B}) = \left\{ \begin{array}{cc} \frac{1}{\lambda} \exp(-\frac{y_i}{\lambda}) & y_i > 0 \\ 0 & \text{otherwise} \end{array} \right\} . \tag{1}$$

My reason for making this choice on this occasion is that, with this parameterization, the expected value of $(y_i \,|\, \lambda \, \boldsymbol{E} \, \mathcal{B})$ is $\lambda$, which makes interpretation of results easier. Gelman et al. (2014) uses the other parameterization for $\beta > 0$:

$$p_{Gelman}(y_i \,|\, \beta \, \boldsymbol{E} \, \mathcal{B}) = \left\{ \begin{array}{cc} \beta \exp(-\,\beta \, y_i) & y_i > 0 \\ 0 & \text{otherwise} \end{array} \right\} ; \tag{2}$$

you can see that the $\beta$ in equation (2) is just the reciprocal of the $\lambda$ in equation (1). As you can discover from a `help( rexp )` function call, `R` uses the Gelman parameterization, referring to $\beta$ in equation (2) as `rate`, but I can get what I want below (in sampling from what equation (1) would call the Exponential$(\lambda)$ density) just by setting `rate` to $\frac{1}{\lambda}$.

In the code block in Table 1 below, calculating the mean of the simulated $y^*$ values provides a simple sanity check that I do indeed understand the implications of my parameterization: the sample mean of the object `y.star` should be close to $\lambda = 2$ (up to Monte Carlo noise), and it is (the noise level is rather high because I only used an $M$ value of 1,000). In the top panel of the plot in the figure I built, I deliberately set the option `breaks = 64` to the `hist` function call too high, to create too many histogram bars (for locally accurate density estimation); this clearly illustrates (a) the Monte Carlo noise with $M = 1{,}000$ and (b) what happens when you override the default algorithm in `hist` for choosing the number of bars (the default is given in the bottom panel of the plot). At the end of the code block I export the PDF version of the plot to the file `illustrative-plot.pdf`; the LaTeX code to include the graph in the LaTeX PDF–c document is at the end of the `.tex` file, and produces Figure 1.

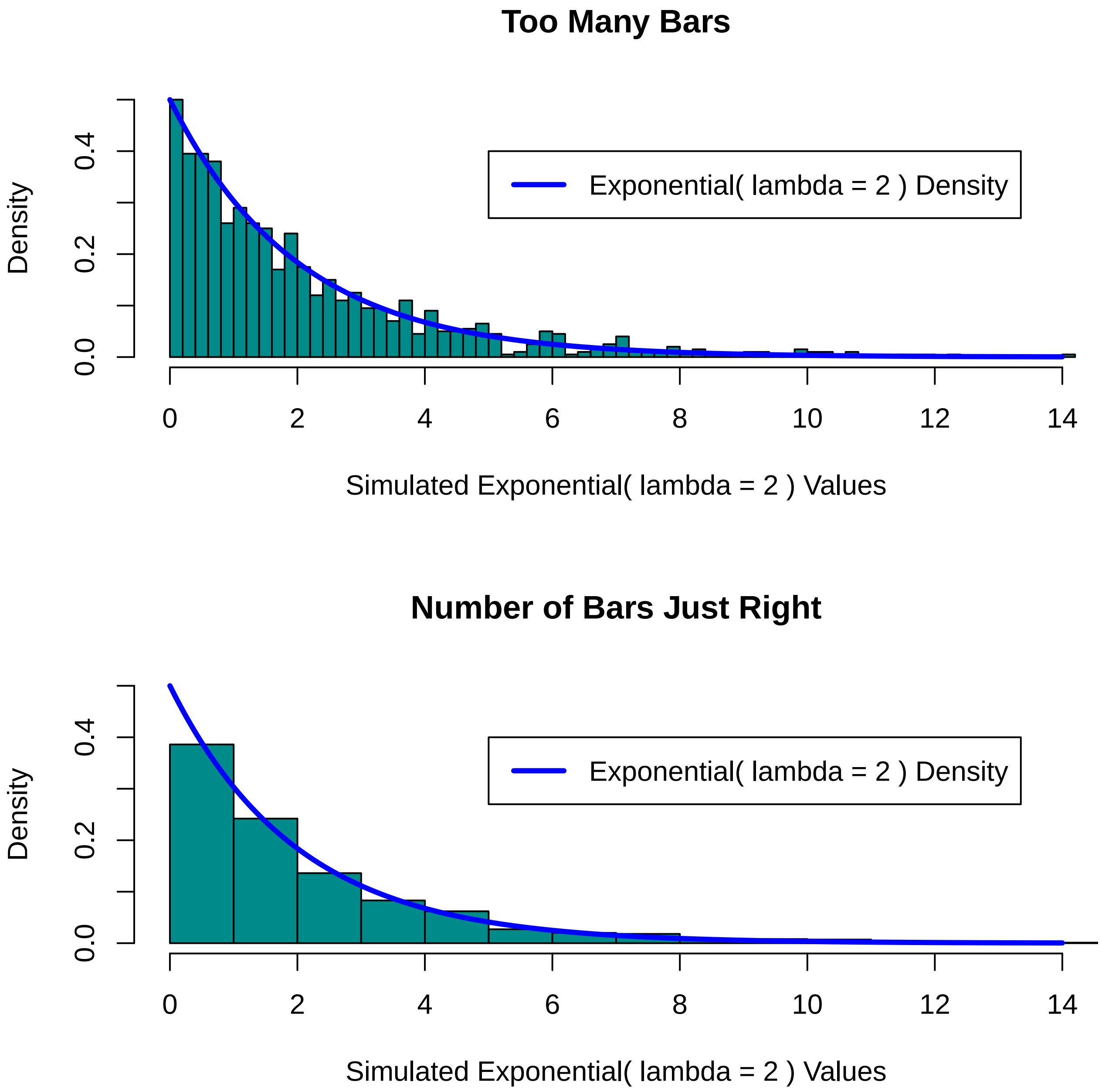
---

[1] Note the terminological collision between PDF (probability density function [STAT 131 and 206] and PDF (portable document format [computing]). When this issue may create confusion, I'll try to remember to use the terms **PDF–p** (for density) and **PDF–c** (for computing).

Table 1: *R code to create a mildly interesting plot and export it as a PDF–c file.*

```
setwd( .../PDF-Importation )
set.seed( 314159 )
help( rexp )
lambda <- 2
M <- 1000
print( mean( y.star <- rexp( M, rate = 1 / lambda ) ) )
# [1] 2.03832
par( mfrow = c( 2, 1 ) )
hist( y.star, prob = T, xlab = ’Simulated Exponential( lambda = 2 ) Values’,
  main = ’Too Many Bars’, breaks = 64, xlim = c( 0, 14 ), col = ’darkcyan’ )
y.grid <- seq( 0, 14, length = 500 )
lines( y.grid, dexp( y.grid, rate = 1 / lambda ), lwd = 3,
  col = ’blue’ )
legend( 5.0, 0.4, ’Exponential( lambda = 2 ) Density’, lwd = 3,
  col = ’darkcyan’ )
hist( y.star, prob = T, xlab = ’Simulated Exponential( lambda = 2 ) Values’,
  main = ’Number of Bars Just Right’, xlim = c( 0, 14 ),
  ylim = c( 0, 0.5 ), col = ’darkcyan’ )
y.grid <- seq( 0, 14, length = 500 )
lines( y.grid, dexp( y.grid, rate = 1 / lambda ), lwd = 3,
  col = ’blue’ )
legend( 5.0, 0.4, ’Exponential( lambda = 2 ) Density’, lwd = 3,
  col = ’blue’ )
par( mfrow = c( 1, 1 ) )
pdf( ’illustrative-plot.pdf’ )
par( mfrow = c( 2, 1 ) )
hist( y.star, prob = T, xlab = ’Simulated Exponential( lambda = 2 ) Values’,
  main = ’Too Many Bars’, breaks = 64, xlim = c( 0, 14 ), col = ’darkcyan’ )
y.grid <- seq( 0, 14, length = 500 )
lines( y.grid, dexp( y.grid, rate = 1 / lambda ), lwd = 3,
  col = ’blue’ )
legend( 5.0, 0.4, ’Exponential( lambda = 2 ) Density’, lwd = 3,
  col = ’blue’ )
hist( y.star, prob = T, xlab = ’Simulated Exponential( lambda = 2 ) Values’,
  main = ’Number of Bars Just Right’, xlim = c( 0, 14 ), col = ’darkcyan’,
  ylim = c( 0, 0.5 ) )
y.grid <- seq( 0, 14, length = 500 )
lines( y.grid, dexp( y.grid, rate = 1 / lambda ), lwd = 3,
  col = ’blue’ )
legend( 5.0, 0.4, ’Exponential( lambda = 2 ) Density’, lwd = 3,
  col = ’blue’ )
par( mfrow = c( 1, 1 ) )
dev.off( )
```

Figure 1: *The plot produced by the `R` code in Table 1.*



**Important note:** the `\includegraphics` command in the LaTeX code here, which imported the PDF–c file into the document to produce Figure 1, is not automatically understood by most versions of LaTeX; instead you make it available to LaTeX in compiling your document with the

```
\usepackage{ graphicx }
```

command in the preamble (see the top of the `.tex` file).