S&DS 238/538 pset xxx

Xxx

Xxx xx, 20xx

This is supposed to be an example that can help you produce nice documents using R's "spinning" facility in preparing your homework assignments. *You **don't** need to use this method.* As discussed in class, you can do whatever works for you to produce a readable electronic homework submission. But if you want to try it, which probably means you are willing either to use some latex for equations or put in a bit of code to import any images (which could be scanned images of a solution containing equations written out on a piece of paper), this can work very well, since it will take your R code and "knit" it together with the output from R and make a document (word, pdf, or html).

If you want to learn to use this kind of method, you can look at the Word or pdf output in pset-spinning-help.docx or pset-spinning-help.pdf and then look at the r script pset-spinning-help.r to see the code that produced that output.

Here I'm writing free text in the R script on lines that begin with `#'`. It is up to your own preference or whim whether you divide up the text in your R script into lots of separate lines (you can look in the R script file to see clearly what I am talking about), or just keep writing and writing without hitting "Enter" on your keyboard.

Problem 1

This could be a problem that requires R. Here is some text first, and then we'll write some R code. Note the difference between this text (on lines that start with `#'` in the R script) and comments (starting with just `#` in the R script): this text appears as normal text in the document and can be formatted elaborately using Markdown commands, while R comments are more restricted and look more like code.

```r
k <- 40 # number of people; this is an R comment
set.seed(123) # you can sets the random number seed if you want to get
              # the same random numbers every time you run your code
bdays <- sample(1:365, k, replace=TRUE)
bdays
```

```
##  [1] 105 288 150 323 344  17 193 326 202 167 350 166 248 210  38 329  90
## [18]  16 120 349 325 253 234 363 240 259 199 217 106  54 352 330 253 291
## [35]   9 175 277  79 117  85
```

```r
table(bdays)
```

```
## bdays
##   9  16  17  38  54  79  85  90 105 106 117 120 150 166 167 175 193 199
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 202 210 217 234 240 248 253 259 277 288 291 323 325 326 329 330 344 349
##   1   1   1   1   1   1   2   1   1   1   1   1   1   1   1   1   1   1
## 350 352 363
##   1   1   1
```

```r
# Maximum number of birthdays that are on the same day
max(table(bdays))
```

```
## [1] 2
```

So we got a match – two people with birthdays on day 253. (I was able to write that without worrying that the next run of the code would come out differently because I set the random number seed and saw how the numbers came out).

Having performed a birthday experiment once, we can create a vector `maxs` and then use a loop to repeat the above commands and store the results in `maxs`:

```
nrep <- 10000
maxs <- rep(0, nrep)
for(i in 1:nrep){
  bdays <- sample(1:365, k, replace=TRUE)
  maxs[i] <- max(table(bdays))
}
table(maxs)
```

```
## maxs
##    1    2    3    4    5
## 1090 8250  635   24    1
```

```
mean(maxs > 1)
```

```
## [1] 0.891
```

Here the fraction of times we got a birthday match was 0.891.

Note the previous sentence gave an example of how to include an inline R calculation in your text. You can use backticks to render text in `verbatim style` or to call R commands or objects.

Problem 2

This is a paper and pencil problem, with no R and a mixture of text and math. We can put equations in it using Latex expressions.[1] For example, you can write an inline equation: $e^3 \approx 20$, and also $\int_0^1 x^2\,dx = \frac{1}{3}$. You just need to put inline math between single dollar signs, `$`, when you write your R script.

If you want math to be displayed on its own line, centered, you can simply put it between double dollar signs `$$` in your R script, like this:

$$\int_0^1 x^2\,dx = \frac{1}{3}.$$

See how the previous line was displayed? We could explain that last calculation in more detail with a multiline equation:

$$\begin{aligned}
\int_0^1 x^2\,dx &= \frac{1}{3}x^3\Big|_0^1 \\
&= \frac{1}{3}1^3 - \frac{1}{3}0^3 \\
&= \frac{1}{3}
\end{aligned}$$

Basically it's just a bunch of latex math, with `\\` ending the lines in the multiline equation, and `&` signs marking where the equations should be aligned. A convenient and helpful "cheatsheet" for Latex math can be found here. (And you've just seen an example of how to insert a hyperlink. Or you can just write the URL and that will be turned into a hyperlink.)

[1] A footnote! And, to repeat, Latex isn't necessary–any legible homework is accepted!

If compiling to MS Word, depending on the version of Word you're using, it's possible you might need to change the font for them to display properly. But it's likely with a modern version of Word that it will just look fine out of the box.

If you want to compile directly to pdf, you will need to have Latex installed; more information about installation can be found at http://www.latex-project.org. But you can also get pdf by compiling to Word and then saving the Word as pdf, so there is no particular need to install latex unless you'd like to.

A tool on the web that might be helpful for writing mathematical expressions in a relatively intuitive way is here. In this tool you click on symbols and things, and the latex code comes out in the yellow box.

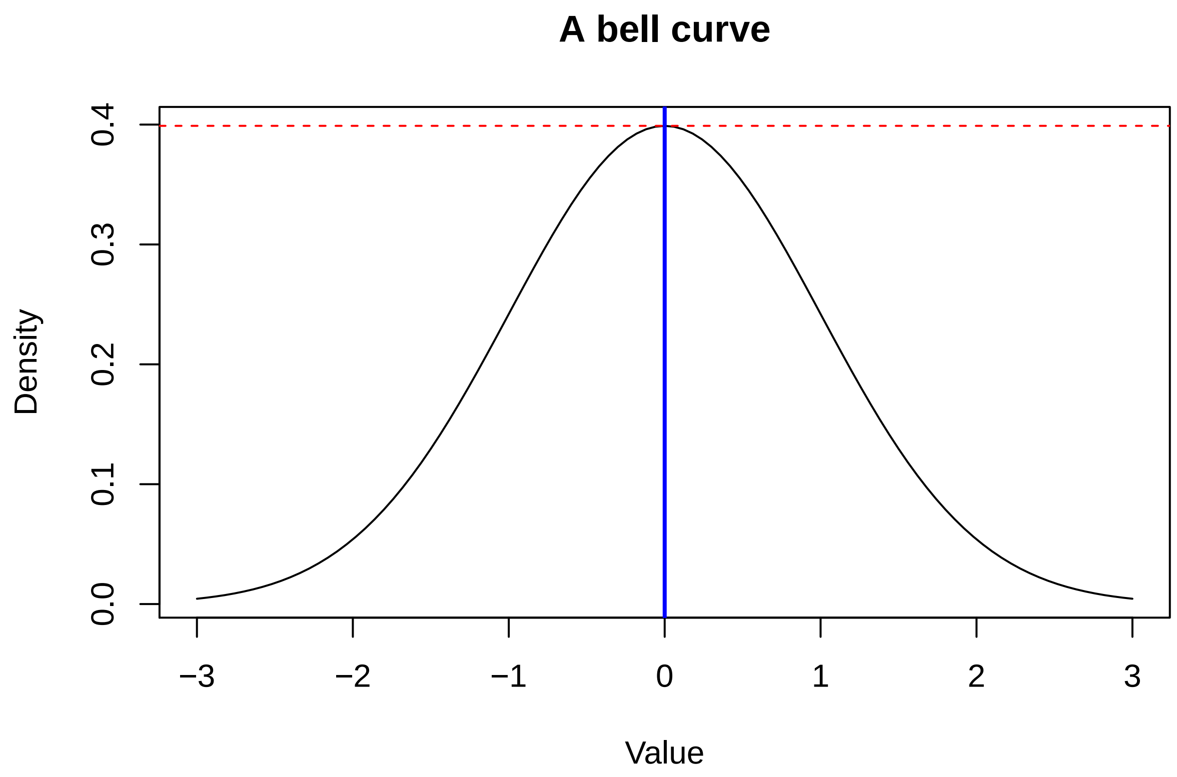
Problem 3

Part 1. Here is a way to put in a picture, using the `knitr::include_graphics` function.

For R to be able to find the picture file "puppies.jpg" just given the file name, the file should be in the same folder (or directory) as the R script. (If not, you can specify the whole path to the file instead of just the file name, but it's probably simplest just to put picture files in the same folder as the R script that uses them.) The line beginning with `#+` is supposed to scale the picture to determine the width (it works in html and pdf but apparently not Word), and the `echo=FALSE` causes the following "R chunk" *not* to be "echoed" or included in the resulting document.

Part 2. And here is a way to put in an R plot.

```r
plot(dnorm, -3, 3, main = "A bell curve", ylab = "Density", xlab = "Value", type="l")
abline(v = 0, lwd = 2, col = "blue") # a vertical line
abline(h = dnorm(0), lty=2, col = "red") # a horizontal dotted line
```



A note on compiling: Reports are compiled using a knitr function called "spin." You can look up `knitr::spin` if you want to learn more options for your text and code presentation. Knitr uses Markdown formatting, which you can look up for things like styling text and creating lists.