STATS 779 Group Project

Due 2359hrs Friday 27 May 2016

How to submit this project

1. E-mail me (j.curran@auckland.ac.nz) your project as a R source package – i.e. use the *Build Source Package* menu option in the R Studio *Build* menu.
2. **Each** team member must send me an e-mail describing their contributions to the project. The e-mails from each member should be sent independently and should not be obvious duplicates.

This is a group project. I have assigned you to teams of three. I will email each group with the group members. This will enable to contact the members of your team. There is also a check-pointed timetable on the back of this sheet.

The brief for this project is very simple – I want you to write a small R package using Rcpp. Rcpp makes it easy to utilize the speed and power of C++ from R. The package will implement the probability density function (pdf), cumulative distribution function (cdf) and quantile function (cdf) for a **non-standard** univariate continuous distribution. Non-standard means that it is not built into R, so for example the normal distribution, Student's t, F and chi-squared distributions are all ruled out (note just because I haven't written all the possibilities here doesn't mean the rest are allowable).

Some points worth noting about this project are:

1. I strongly suggest that you choose some sort of lifetime distribution because the cdf and quantile functions are likely to be analytically defined.
2. Your chosen distribution must be different from the other teams, so there is an incentive to getting started early
3. Your programme should use your own C++ code. It should not be someone else's – e.g. not something you "found" on Stack Overflow. It can depend on other distributions if necessary – for example some distributions require the normal cdf and quantile function.
4. The bulk of the computation should be performed with C++.
5. I highly recommend that you work within R Studio. This makes the process of creating a package with Rcpp and compiling it infinitely easier.

6. Your package must be properly documented and each help file should include some examples that can be run using the `example` function.
7. Your three functions should take the standard *d*, *p*, and *q* format, e.g. `dnorm` for the pdf, `pnorm` for the cdf, and `qnorm` for the inverse cdf
8. There are allocated marks for making your functions fully vectorised.
9. Your package should be installable using the `install.packages` function, and it should install without error.

There will be no direct instruction on RCpp. The exception is a short demonstration of how to create a new R package with Rcpp in R Studio. I (James) will give you help as long as you show me that you have done some work. If you want me to help you, then you need to bring your code on a flash-drive, or bring your laptop to my office.

The programmes will be judged by both lecturers ranked. Marks will then be awarded according to the ranking.

Guaranteed ways to get **zero** marks are:

1. Handing in a programme that does not work
2. Handing in code that has no documentation in it
3. Copying someone else's project in its entirety from the web

This may sound like a very vague project and it is. We do not want to stifle your originality with too many constraints. Good luck!

Check points

Due Date	Task	Marks
Friday March 18	Email distribution choice to James. Do this earlier so I can tell you if you need to choose again	0.2
Friday April 25	Submit a half page progress report to Canvas dropbox	1.0
Friday May 27	Hand in project files (Canvas dropbox)	3.8
	Total	5