Creating a Quick Report with knitr, xtable, R Markdown, Pandoc (and some OpenBLAS Benchmark Results)

Introduction

I was looking for ways to create professional-looking documents with R codes. After some experiments with Sweave, I repeatedly reinforced my uncomfortable feeling with LaTeX. Luckily, the availability of knitr, xtable and Pandoc has made things easier for me. I would say this is a solution that gives satisfactory results with minimal effort.

While I was researching on this subject, I also noticed that a new version of OpenBLAS had been made available (as always, thanks to Tal's R-bloggers!). I thought a simple performance comparison between my current version of OpenBLAS (v0.2.6-1) and the new one (v0.2.8-1) would be interesting. A blog post about code-generated report **with** the OpenBLAS benchmark results would be even better! So here we go.

As this post is more about generating report with codes, I am not going into the details of the OpenBLAS installation and the benchmarking process. For more information, see this, and this.

Brief Overview of Workflow

I would say the documentation process involves three major steps:

1. Write R scripts to carry out the analysis and to summarise the results.
2. Create a R Markdown file (.Rmd) to document the results.
3. Write another R script to convert R Markdown (.Rmd) into Markdown (.md) and eventually a LaTeX-PDF using Pandoc.

The codes I used to produce this report are available on **Github.**

Comparing R-25 Benchmark Results

The R-25 benchmark consists of 15 tests. Originally, they are split into 3 groups (matrix calculation, matrix functions and programmation) with trimmed means for each group. For this comparison, I decided to take the overall mean. The benchmark results (in seconds) from each OpenBLAS test are summarised in the tables below.

Some basic specs of my machine: Intel i7-2630QM (8 cores), 8GB RAM, Linux Mint 15 Cinnamon and R 3.0.1 x64.

Group	Test	v0.2.6	v0.2.8	Difference
Matirx calculation	Creation, trans ...	1.233	1.072	1.2x faster
Matirx calculation	2400x2400 norma ...	0.819	0.822	<= 2.5%
Matirx calculation	Sorting of 7,00 ...	0.875	0.873	<= 2.5%
Matirx calculation	2800x2800 cross ...	0.650	0.550	1.2x faster
Matirx calculation	Linear regr. ov ...	0.377	0.295	1.3x faster
Matirx functions	FFT over 2,400, ...	0.474	0.453	1x faster
Matirx functions	Eigenvalues of ...	0.826	1.354	1.6x slower
Matirx functions	Determinant of ...	0.485	0.389	1.2x faster
Matirx functions	Cholesky decomp ...	0.440	0.353	1.2x faster
Matirx functions	Inverse of a 16 ...	0.379	0.392	1x slower
Programmation	3,500,000 Fibon ...	0.862	0.750	1.1x faster
Programmation	Creation of a 3 ...	0.440	0.280	1.6x faster
Programmation	Grand common di ...	0.855	0.640	1.3x faster
Programmation	Grand common di ...	0.688	0.710	1x slower
Programmation	Escoufier's met ...	0.425	0.842	2x slower

Table 1: R-25 Benchmark Results

Group	Test	v0.2.6	v0.2.8	Difference
Summary	Total time for ...	9.827	9.775	<= 2.5%
Summary	Overall mean (n ...	0.655	0.652	<= 2.5%

Table 2: R-25 Benchmark Comparison Summary

According to these preliminary results, the latest version of OpenBLAS is slightly faster than the previous version installed on my machine for most of the tasks. Yet, for the tests "Eigenvalues of a 640x640 random matrix" and "Escoufier's method on a 45x45 matrix (mixed)", there is a disappointing drop in performance. I would guess it is due to the overhead involved (from 2 cores in version 0.2.6-1 to the maximum 8 cores in version 0.2.8-1) but more robust tests are needed to verify this. For now, I am happy to stay with the 0.2.8-1 version.

ggplot2

How could I finish my report without some ggplot2 magic? Here is a simple plot to visualise the benchmark results comparison:

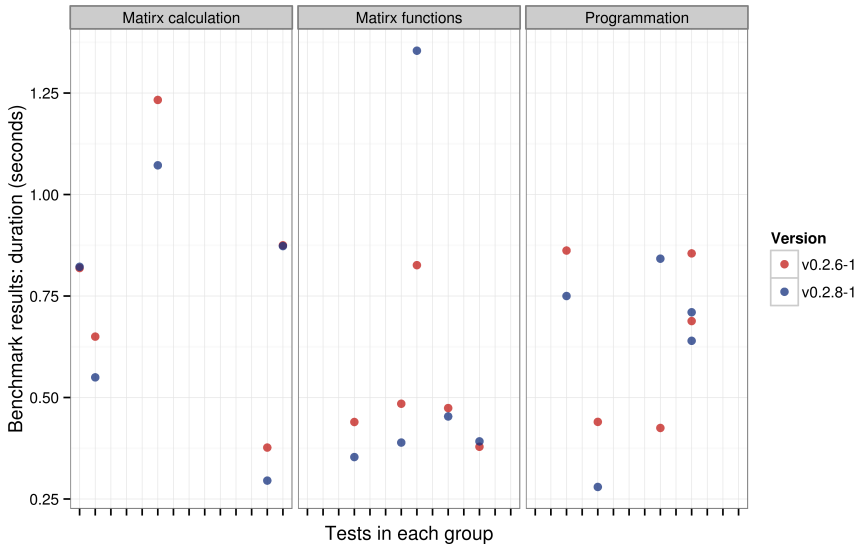

Figure 1: Benchmark Results Comparison

Acknowledgement

A **big thank you** to the people who developed the tools and made them available to the community: The RStudio Team (RStudio IDE and R Markdown), Yihui Xie (knitr), David Dahl & Charles Roosen (xtable) and John MacFarlane (Pandoc).

Conclusion

I hope this very basic example will get you interested in code-generated documents (with very limited experience of LaTeX like myself). I only wish I had learned about all these open-source tools a lot earlier (well, like 3 years earlier when I first started my EngD project ...)