

Smath Documentation

T. Stumpf-Fétizon

April 19, 2015

`smath` is a collection of \LaTeX macros that automates and centralizes most of the formatting in math mode. The package defines appropriate formatting for different mathematical objects which can be easily applied to any string. The overall goal is to let you focus on the meaning of your writing, just as outside of math mode.

1 Use Cases

While providing users with the power to typeset any mathematical expression, \LaTeX still requires you to pay a lot of attention to the finer points of mathematical typography: spacing has to be increased around functions, operators have to be printed in normal font, delimiters require scaling... all these things increase cognitive load and divert attention from the actual content. Furthermore, direct formatting clutters the source code and is inconsistent with the principle of separation of layout and content. Thankfully, as 99 percent of the objects requiring further formatting may be grouped into a dozen cases, a few macros are sufficient to improve upon that situation. We reap the following benefits:

- Almost all of the formatting details are automated through the use of a dozen macros. This includes most cases in which delimiters have to be scaled to reach around their content.
- Commands become more indicative of the object they're formatting. This improves readability.
- More gains to readability are made by freeing the source code of direct formatting commands.
- Finally, you can consistently change the layout of every equation in the document by slightly modifying the package.

2 Importing smath

Place `smath.sty` in the same folder as your `.tex` file and add `\usepackage{smath}` to your preamble. `Smath` requires the `mathtools` and `xparse` packages. You can now use any macro defined in `smath.sty`. Consult the file for macro definitions. You can easily extend the package by adding your own macros to the file. Consult the `xparse` documentation on how to define macros.

The goal is to have a set of options to choose from to extend functionality. Currently, there is a `stats` option that mostly provides macros for density functions.

3 Coding Examples

This section introduces some concrete applications of `Smath` macros to math formatting.

3.1 Operators and Functions

Formatting operators is the most complex use case of this package. Accordingly, the `op` command provides a lot of flexibility that requires a little bit of studying. Consider, for example, the inequality giving a chernoff bound:

Listing 1: Chernoff bound

```
1 \op{Pr}{\op*{\sum}[i] X_i \geq a} \leq \op*{inf}[t \geq 0] \set{e^{-ta}}
   \op{E}{\op*{\prod}[i = 1][n] e^{tX_i}}
```

$$\Pr \left[\sum_i X_i \geq a \right] \leq \inf_{t \geq 0} \left\{ e^{-ta} \mathbf{E} \left[\prod_{i=1}^n e^{tX_i} \right] \right\} \quad (1)$$

Observe how compact the markup is in spite of the complexity of the formatting. The following 4 items provide you with all you need to achieve similar efficiency whenever you are formatting operators:

- We want operators to stand out, typically by using normal or bold font. This requires a formatting command on every operator. The solution is to pass the operator name as the first argument to the `op` command.

Listing 2: Names

```
1 \op{inf}
```

$$\mathbf{inf} \quad (2)$$

- In many cases, we want operators to have additional spacing around them, lest we end up with something like $a\mathbf{E}X$. In some cases, we only want additional spacing on one side because the operator is followed by subscripts or brackets. You can obtain this behavior by passing sub (sup) scripts as optional arguments.

Listing 3: Spacing

```
1 \op{E}[X] X
```

$$\mathbf{E}_X X \quad (3)$$

- For some operators, we want sub/supscripts directly below and on top of the operator. All you have to do is add a star (*) to the `op` command. Note that this only applies to display mode.

Listing 4: Limits

```
1 \op*{inf}[x \in \mathbf{R}] \func{f}{x}
```

$$\mathbf{inf}_{x \in \mathbf{R}} f(x) \quad (4)$$

- If we apply an operator to larger expressions, we want that expression to be surrounded by brackets. Passing that expression to the `op` command ensures that the delimiters are scaled correctly and makes the association clearer. It is always passed last, after any optional arguments:

Listing 5: Brackets

```
1 \op{E}{\op*{\prod}[i = 1][n] X_i}
```

$$\mathbf{E} \left[\prod_{i=1}^n X_i \right] \quad (5)$$

3.2 Linear Algebra

Linear algebra often requires a lot of extra formatting to emphasize vectors and matrices. The least squares estimator is a good example:

Listing 6: OLS

```
1 \est{\vect{\beta}} = (\mat{X}\T \mat{X})^{-1} \mat{X}\T \vect{y}
```

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (6)$$

We use `vect` for vectors, `mat` for matrix and `T` for transpose. This saves us from having to deal with three different font formatting commands across the equation. To grasp the difference, consider the underlying code that the macros apply:

Listing 7: OLS with direct formatting

```
1 \widehat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathsf{T}} \mathbf{X})^{-1} \mathbf{X}^{\mathsf{T}} \mathbf{y}
```

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (7)$$

All of the usual advantages are present: The code is more compact, more meaningful, and you can change formatting styles for the entire document by modifying macros.

3.3 Canned Formulae

Some complicated expressions occur frequently and it is worth learning some extra macros to print them quickly. Consider for example the density function of the multivariate normal distribution:

Listing 8: Multivariate normal distribution

```
1 \func{f}{\vect{x}, \vect{\mu}, \mat{Q}} = (2\pi)^{-\frac{D}{2}} \op{det}
    {\mat{Q}}^{-\frac{1}{2}} \op{exp}{-\frac{1}{2} (\vect{x} - \vect{\mu}
    )\T \mat{Q} (\vect{x} - \vect{\mu})}
```

$$f(\mathbf{x}, \boldsymbol{\mu}, \mathbf{Q}) = \frac{\det[\mathbf{Q}]^{1/2}}{(2\pi)^{D/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{Q} (\mathbf{x} - \boldsymbol{\mu}) \right] \quad (8)$$

You can avoid hand-coding that formula by using canned macros, such as `dmnorm`:

Listing 9: Multivariate normal distribution (canned form)

```
1 \func{f}{\vect{x}, \vect{\mu}, \mat{Q}} = \dmnorm{\vect{x}}{\vect{\mu}
    }{\mat{Q}}
```

$$f(\mathbf{x}, \boldsymbol{\mu}, \mathbf{Q}) = \frac{\det[\mathbf{Q}]^{1/2}}{(2\pi)^{D/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{Q} (\mathbf{x} - \boldsymbol{\mu}) \right] \quad (9)$$

Note that not all arguments will produce a nicely formatted formula, so you might have to go back to coding manually in some cases.

4 Further Reading

- American Mathematical Society, *User's Guide for the amsmath Package* (1999).
<ftp://ftp.ams.org/pub/tex/doc/amsmath/amsldoc.pdf>
- The L^AT_EX3 project, *The xparse package Document command parser* (2014).
<http://mirror.math.ku.edu/tex-archive/macros/latex/contrib/l3packages/xparse.pdf>