LaTeX Miscellany

Last Time ...

- `maths`, maths, and MaThs

This Time ...

- Tables
- Pictures
- Floats
- More math environments
- Words of Wisdom

Tables

- We can create tabular content in a LaTeX document either within a float environment or not. We will describe what a float is in a moment.
- Either way, tabular content like the following is created with the `tabular` environment.

Student	Alma Mater	Favorite Color
Tyson Chatagnier	Texas A&M	Denim
Jonathan Klingler	Notre Dame	Green
Gary Hollibaugh	UCSD	Orange and Aquamarine
Jonathan Olmsted	UDel	French Blue
Peter Haschke	UNC-Asheville	Titian Red
Lukas Pfaff	Iowa State University	Cornflower blue[1]

- The `tabular` environment takes a mandatory argument that is enclosed in curly braces. In the case of the above table where each column is center justified, we'd use `\begin{tabular}{c c c} \end{tabular}`. If we wanted the columns to be right-justified or left-justified we could use `r` or `l`.
- If we wanted to add a vertical line on either the left or right or in-between two columns we'd use the argument `{| c | c | c |}`.
- Horizontal lines are given by `\hline`.
- Tab breaks are caused by `&` and line breaks by `\\`.
- The full code for the above table is:

```
\begin{center}
  \begin{tabular}{c c c}
    \hline
    \hline
    Student & Alma Mater & Favorite Color \\
    \hline
    Tyson Chatagnier & Texas A\&M & Denim \\
    Jonathan Klingler & Notre Dame & Green \\
    Gary Hollibaugh & UCSD & Orange and Aquamarine \\
    Jonathan Olmsted & UDel & French Blue \\
    Peter Haschke & UNC-Asheville & Titian Red \\
    Lukas Pfaff & Iowa State University & Cornflower blue
    \footnote{This is not necessarily true, it is,
      however, just a corn joke!'}\\
    \hline
    \hline
  \end{tabular}
\end{center}
```

- The line breaks and whitespace in the above code are poorly selected, but this results from the need to fit the code in such a small block. Usually, the code for LaTeX tables can be several times wider than anything else in your LaTeX document.
- There are much more complex environments than `tabular` that you will have to use once your tabular content grows. They are similar, but have their own documentation to help you through.

Pictures

- LaTeX has the ability to draw arbitrary types of objects and schematics within a LaTeX document in a native language. However, this is seldom used and probably less advantageous than including external files.
- Download `Image1.pdf` and `Image2.png` from `www.rochester.edu/college/gradstudents/jolmsted/`
- In your preamble, add the `graphicx` package (i.e. `\usepackage{graphicx}`)
- Where you'd like the picture to show up, type `\includegraphics{./Image1}` without a file extension and be sure to use the appropriate path to the image file, relative to the LaTeX document. The file extension is not necessary.
- The `scale` optional argument is useful to change dimensions on the fly without distorting the aspect ratio.

- Try the following code in a LaTeX document environment:

```
\begin{center}
  \fbox{
    \fbox{
      \includegraphics[scale=.25]{./Image1}
    }
    \fbox{
      \includegraphics[width=3in,
      height=1in,
      angle=90]{./Image2}
    }
  }
\end{center}
```

- The result is something like

- Notice that we didn't have to specify the file extension. Notice the order in which the `angle` and then `height`/`width` arguments are applied. What does `\fbox{}` do?

Floats

- Although we can create tables with the `tabular` environment and we use `\includegraphics{}` to insert external image files, these commands

are seldom placed inside a document without entering them in a special environment.

- Typically, these kinds of content are placed in *floats* which act like containers for tables and figures. LaTeX, according to a set of rules, figures where these containers should be placed which depends on the content before the float, the content after the float, and the LaTeX code within the float.
- If nothing else, the big adjustment required by users placing content in floats is realizing that the placement of a table or figure is up to LaTeX and "jury-rigging"/"jimmy-rigging"/"jerry-rigging" the placement is ill-advised.
- The float for tables is `\begin{table}[htpb] \end{table}`. The optional `[thpb]` argument provides LaTeXwith some instructions on where to place the table.
- The float environment for figures is `\begin{figure}[htpb] \end{figure}`. Notice, again, the optional argument.
- In actuality, there need not be anything inside these float environments, or it could easily be regular LaTeX markup.
- `\caption{}`, placed somewhere in the float, allows a title of the content to be placed and automatically numbered.
- `\label{}`, placed immediately after the `\caption{}` command gives an identified to the object by which it can be referred for directing readers to Figure 1 or Table 4 without hard-coding the float order.
- Try:

```
\begin{figure}
  \begin{center}
    \fbox{
      \includegraphics[width=3in,
      height=1in,
      angle=120]{./Image2}
    }
    \caption{A Hero of a Man} \label{f:Riker}
  \end{center}
\end{figure}
```

- Now, see Figure 1 on page 5 to view the output. We were able to reference the figure number automatically using `\ref{f:Riker}` which matches our `\label`. We reference the page number automatically with `\pageref{}`.

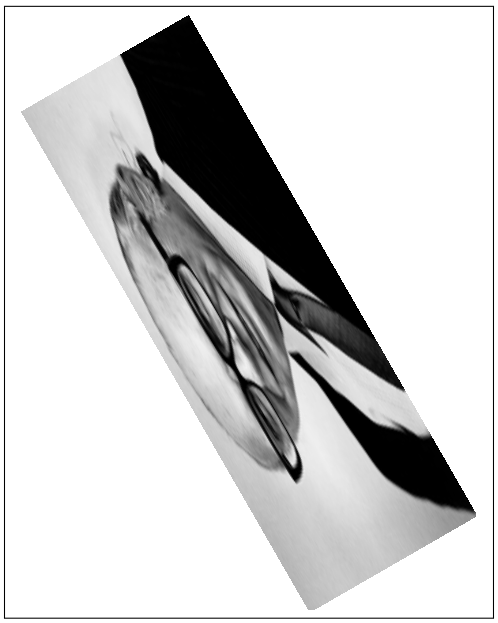

Figure 1: A Hero of a Man

More Math Environments

- There are a number of math environments that become useful when one is typesetting mathematical notation beyond very basic experessions.
- As in a previous tutorial, add `\usepackage{amsmath, amsthm, amssymb, amsfonts}` to the preamble if not already there.

Fraction

There are instances where 3/4 would look better as $\frac{3}{4}$ and this works equally well for longer expressions

$$\frac{\tan\left(\cos\left(\sin(X)\right)\right)}{\int_{\mathbb{R}} f(x)\ dx}.$$

It is the command `\frac{}{}` which provides this. The numerator is the first argument and the denominator is the second.

Equation

The equation environment is a very common way to typeset equations when reference numbers are being used. So, for example, `\begin{equation} 4=x^2 \end{equation}`

gives

$$4 = x^2. \tag{1}$$

Now, it is not necessary to place the equation environment in a math mode. In this sense, the math mode is implied. There is also a variant such that `\begin{equation*} 4=x^2 \end{equation*}` suppresses the equation numbering,

$$4 = x^2.$$

Equation numbers can be labeled and referenced as was done in the figure environment. This is a single equation environment and if you try to enter line breaks such that you could force another line, it will fail. In that case, I find the `align` approach being the easiest because it is flexible.

Align

The `align` environment allows you to enter multiple lines and include alignment stops. There is a un-numbered version, `align*`, too. The code

```
\begin{align}
   \sum_{x=1}^{4} \frac{1}{x} &
   &= \frac{1}{1} + \frac{1}{2}  + \frac{1}{3} + \frac{1}{4} & \\
   &= \frac{25}{12} &\\
   & &>2
\end{align}
```

produces a three line `align` environment.

$$\sum_{x=1}^{4} \frac{1}{x} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \tag{2}$$

$$= \frac{25}{12} \tag{3}$$

$$> 2 \tag{4}$$

Notice that the numbering is cumulative. The use of `\label{}` and `ref{}` works identically here.

Array

The `array` environment provides a unified way of representing vectors and matrices. The environment begins in the standard `\begin{array}{cc} \end{array}` way, but the mandatory `{cc}` argument specifies there are two center-justified columns. Changes to this argument proceed identically to the argument in the creation of tables. On important point is that the array environment does not create its own math mode environment, so we must put it inside one when we use it. We get

$$\left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right],$$

the two dimensional identity matrix from

```
\[
 \left[
\begin{array}{cc}
  1 & 0 \\
  0 & 1
\end{array}
\right],
\]
```

Notice the comma in the source code. It must be in the display math environment so that it is placed adjacent to the matrix and not in the text. By changing the number of rows, columns, and the delimiters, most matrix-like objects can be represented with this environment.

Cases

The `cases` environment is both extremely useful, but quite narrow in application. Although it is designed to be used only to represent piece-wise functions, it is a marked improvement over the alternative which would be to "hack" the `array` environment. Like the `array` environment, though, `cases` must be used within math mode. So,

$$\mathbf{1}_{\mathcal{X}}(x) = \begin{cases} 1, & x \in \mathcal{X} \\ 0, & \text{otherwise} \end{cases}$$

is the result of the code

```
\[
\mathbf{1}_{\mathcal{X}}(x) =
\begin{cases}
  1, & x \in \mathcal{X} \\
  0, & \textrm{otherwise}
\end{cases}.
\]
```

Notice how I use whitespace and line breaks to organize the code although LaTeX won't interpret it.

Words of Wisdom

- It is considered good practice to keep text file contents within the first 80 characters. This may seem weird or hard, but this, along with use of the `%` and comments will make the input file more human-readable.
- As you learn LaTeX don't worry about trying to make LaTeX look a certain way. Tell LaTeX about your content and its structure. Let LaTeX worry about the details of appearance.

- Google is your friend.
- Comment out error-laden parts of code. Add things back in one at a time until you've identified the source of your syntactical mistakes.
- Let WinEdt help you. It highlights the source file according to rules. If the rules are broken, the highlighting will appear other it should and this is a visual cue that something is wrong.
- Because LaTeX seldom interprets whitespace in too generous a way, use it as an organizational tool.