

# Writing Finance Papers Using L<sup>A</sup>T<sub>E</sub>X\*

Richard Stanton<sup>†</sup>

April 6, 2019, 22:10

## Abstract

This document provides some recommendations for efficiently writing finance papers using L<sup>A</sup>T<sub>E</sub>X and related tools, based on techniques that have made me more productive over the years. It also contains the necessary L<sup>A</sup>T<sub>E</sub>X style files and BibT<sub>E</sub>X format files (along with sample paper outlines) for making your paper and bibliography match the formats required by the *Journal of Finance*, *Journal of Financial Economics*, and *Review of Financial Studies*. The source file, `texintro.tex`, and various additional included files are important parts of this document.

**RS:** Notes like this appear throughout the document and are created using the `todonotes` package (see Section 4.6).

**RS:** Including the current time helps to locate the latest printout in that huge pile on your desk when you create many versions of a paper during the day (see `datetime` package, Section 4.7).

---

\*I am grateful to Jennifer Carpenter and Johan Walden for valuable input (even if some of it was inadvertent). I am also grateful to Tobias Oetiker for writing the `example` environment, which I use liberally in this document. **INCLUDES NOTES.**

<sup>†</sup>Haas School of Business, U.C. Berkeley, 545 Student Services Building #1900, Berkeley, CA 94720-1900, phone: 510-642-7382, fax: 510-643-1412, e-mail: stanton@haas.berkeley.edu.

**RS:** I like to have this on the front page to reduce the likelihood that the paper will go back to the journal complete with all my insulting comments about the referee.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installing L<sup>A</sup>T<sub>E</sub>X</b>	<b>3</b>
<b>3</b>	<b>Introduction to L<sup>A</sup>T<sub>E</sub>X</b>	<b>4</b>
3.1	Entering text . . . . .	4
3.2	Mathematical expressions . . . . .	7
<b>4</b>	<b>Useful L<sup>A</sup>T<sub>E</sub>X Packages</b>	<b>8</b>
4.1	Installing packages . . . . .	9
4.2	Changing line spacing: <code>setspace</code> . . . . .	11
4.3	Including figures: <code>graphicx</code> . . . . .	11
4.4	Mathematical typesetting: <code>amsmath</code> . . . . .	12
4.5	Setting page margins: <code>geometry</code> . . . . .	13
4.6	Inserting notes: <code>todonotes</code> . . . . .	14
4.7	Printing date and time: <code>datetime</code> . . . . .	15
4.8	Multiple versions from one file: <code>versionPO</code> . . . . .	15
4.9	List formatting: <code>enumitem</code> . . . . .	17
4.10	URLs and hyperlinks: <code>hyperref</code> . . . . .	17
4.11	Landscape and subfigures: <code>rotating</code> and <code>subfigure</code> . . . . .	18
4.12	Other useful packages . . . . .	18
<b>5</b>	<b>Citations and Bibliographies</b>	<b>20</b>
5.1	BibT <sub>E</sub> X . . . . .	21
5.2	The BibT <sub>E</sub> X database . . . . .	21
5.3	Telling BibT <sub>E</sub> X where to find the database . . . . .	22
5.4	Formatting the bibliography: <code>natbib</code> and <code>custom-bib</code> . . . . .	23
<b>6</b>	<b>Journal-Specific Document Formatting</b>	<b>24</b>
6.1	Formatting the bibliography . . . . .	24
6.2	Formatting the text . . . . .	25
<b>7</b>	<b>Working with Others: Version Control Systems</b>	<b>25</b>
7.1	Public hosting of DVCS repositories . . . . .	28
7.2	This project's Bitbucket repository . . . . .	28

# 1 Introduction

L<sup>A</sup>T<sub>E</sub>X is a document typesetting system, based on Knuth's program T<sub>E</sub>X (see ?), which makes it relatively simple to produce papers with high-quality equations and error-free cross-references to equations, tables, and figures, and to handle creation and formatting of bibliographies and citations.<sup>1</sup> This document is not a comprehensive overview of L<sup>A</sup>T<sub>E</sub>X. It is, rather, a collection of suggestions for tools and techniques that make me more productive when I write papers, along with some instructions and sample files for formatting your papers to match the requirements of the *Journal of Finance*, *Journal of Financial Economics*, and *Review of Financial Studies*.

**RS:** Section 5 deals with creating citations and bibliographies. Note that this link is live and clickable (see `hyperref` package, Section 4.10).

If you're starting from scratch, one very popular reference, free to download, is the "Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>" (?). There are also some excellent books on L<sup>A</sup>T<sub>E</sub>X. Standard references include ?, written by the original author of L<sup>A</sup>T<sub>E</sub>X; ?, a more recent, popular introductory text; and ?, an excellent reference on both L<sup>A</sup>T<sub>E</sub>X itself and many of the additional packages that have been created since L<sup>A</sup>T<sub>E</sub>X was created (the authors of this book are some of the main developers of L<sup>A</sup>T<sub>E</sub>X).<sup>2</sup>

Creation of a document using L<sup>A</sup>T<sub>E</sub>X is a two-step process:

1. Create a L<sup>A</sup>T<sub>E</sub>X source file, usually with the extension `.tex`, containing your text plus commands that tell L<sup>A</sup>T<sub>E</sub>X how to format your document.<sup>3</sup>
2. Process the `.tex` file to create the final PDF file.<sup>4</sup>

**RS:** By default, L<sup>A</sup>T<sub>E</sub>X inserts more space between list items than this (see `enumitem` package, Section 4.9).

For example, here is an extremely simple L<sup>A</sup>T<sub>E</sub>X file (the text after "`%`" on each line is just a comment; it is ignored by L<sup>A</sup>T<sub>E</sub>X):

---

<sup>1</sup>Plenty of people use word processors for writing papers instead. This makes no sense to me, but rather than engage in a debate, I'll just note that if you use Word this document is probably not going to be very useful to you.

<sup>2</sup>I own and refer to all of these books, especially ?, and I originally learned L<sup>A</sup>T<sub>E</sub>X from ?. You can also find other references at <http://www.ctan.org/starter.html>. However, I find that nowadays I use Google more than any other reference.

<sup>3</sup>For example, the source for this document is called `texintro.tex`. Unlike word-processor files, L<sup>A</sup>T<sub>E</sub>X files are plain text, with no special control codes. This makes them easily portable from one machine (or architecture or OS) to another, easy to view using any plain-text file viewer, and also means that you stand a decent chance of being able to rescue a lot of your material if the file accidentally gets corrupted, say by a misbehaving word processor or a disk failure. Not that files get corrupted all that often, of course, but just in case, it's nice to know. This also makes it easy to keep track of your work using a Version Control System (see Section 7).

<sup>4</sup>Before PDF became standard, it was usual for the `.tex` file to be processed to a `.dvi` (**D**evice **I**ndependent) file, which was then further processed to a printable format using a printer-specific executable (e.g., `dvips` for Postscript or `dvilj` for an HP Laserjet). To create a PDF file, you would 1) run `latex` to create a `.dvi` file, 2) run `dvips` to convert the `.dvi` file into a Postscript file, and 3) run Adobe Distiller (or Ghostscript) to convert the PS file to PDF. Nowadays this is usually done in one step (using the `pdflatex` program behind the scenes). Yes, I know you still insist on using `dvips`, Johan, though I have no idea why....

```

\documentclass{article}           % Tell LaTeX to use the "article" style
\begin{document}                 % Document starts here
This is my first paper.          % Text
\end{document}                   % Document ends here

```

If you process this file using L<sup>A</sup>T<sub>E</sub>X, it produces the following PDF file:<sup>5</sup>

**RS:** Describe what the paper contains.

## 2 Installing L<sup>A</sup>T<sub>E</sub>X

Writing papers using L<sup>A</sup>T<sub>E</sub>X requires, in principle, three separate pieces of software:

1. A text editor to edit the .tex source file.
2. T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X software to process your source file and create the final PDF file.
3. A PDF viewer so you can see the output.

While you can install three separate pieces of software, it is much simpler to use a L<sup>A</sup>T<sub>E</sub>X installation that comes with a “front-end” program, which seamlessly integrates all of these pieces into one so you never have to think about the different steps involved.<sup>6</sup> The standard L<sup>A</sup>T<sub>E</sub>X installations for Windows and OS X both come with a good front-end program:

OS	Installation Name	Front end	URL
Windows	MikTeX	TeXworks	<a href="http://miktex.org">http://miktex.org</a>
OS X	MacTeX	TeXShop	<a href="http://www.tug.org/mactex">http://www.tug.org/mactex</a> .

To install one of these on your system, go to the URL listed above (you can click on the link) and follow the installation instructions.<sup>7</sup>

<sup>5</sup>There’s also a page number at the bottom of the page, which I’ve omitted to save space.

<sup>6</sup>There is nothing to stop you switching to separate programs later, if you want. For example, I use Emacs (see <http://www.gnu.org/software/emacs/>) for editing .tex files (and for almost everything else in my life), and either SumatraPDF (Windows—see <http://blog.kowalczyk.info/software/sumatrapdf>) or Skim (OS X—see <http://skim-app.sourceforge.net/>) to view the resulting PDF files.

<sup>7</sup>There seems to be a bug in the MikTeX installation, which causes MikTeX to be installed with a default A4 paper size regardless of what you ask for during the installation. This in turn causes pages to be offset vertically by about an inch. You can get around this by setting fake margins in your document to compensate (yes, I mean you, Johan...), but this will annoy coauthors who don’t suffer from this bug. Instead, change the default paper size manually after installing MikTeX:

- From the Windows program menu, select MikTeX 2.09 → Maintenance (Admin) → Settings (Admin).
- Under “Paper” select “Letter (letterSize),” then press “Update formats” and press OK (see Figure 3).

Creating a document with either TeXworks or TeXShop involves loading the .tex file, editing it until you're happy with it, then pressing a button to create the PDF file, which will also be displayed on the screen. Figure 1 shows what the TeXworks environment looks like when working on the test file from above. Figure 2 shows the same thing for TeXShop. In both cases:

- Different syntactical elements are highlighted in different colors.
- Processing the .tex file to a PDF file entails just pressing a single button.<sup>8</sup>
- When there are errors, you can automatically be taken to the right place in the text.
- You can easily see which bit of the PDF file corresponds to which bit of the .tex file, making editing easier.

## 3 Introduction to L<sup>A</sup>T<sub>E</sub>X

### 3.1 Entering text

Entering text is pretty straightforward—just type it. A couple of things to note if you're used to a word processor (i.e., if you are between 3 and 90 years old): i. L<sup>A</sup>T<sub>E</sub>X doesn't care if you use one or several spaces or how often you insert a line break, and ii. a completely blank line tells L<sup>A</sup>T<sub>E</sub>X to start a new paragraph.

Lots of spaces and  
some line breaks, but \LaTeX\  
doesn't care.

A blank line tells \LaTeX\ to  
start a new paragraph.

Lots of spaces and some line breaks, but L<sup>A</sup>T<sub>E</sub>X  
doesn't care.

A blank line tells L<sup>A</sup>T<sub>E</sub>X to start a new para-  
graph.

Start a new section or subsection using the commands `\section` and `\subsection`. E.g.,

---

<sup>8</sup>To do this manually from a command prompt, you typically have to run a series of commands, especially if you're using BibT<sub>E</sub>X to handle your citations (see Section 5):

```
pdflatex test
bibtex test
pdflatex test
pdflatex test
```

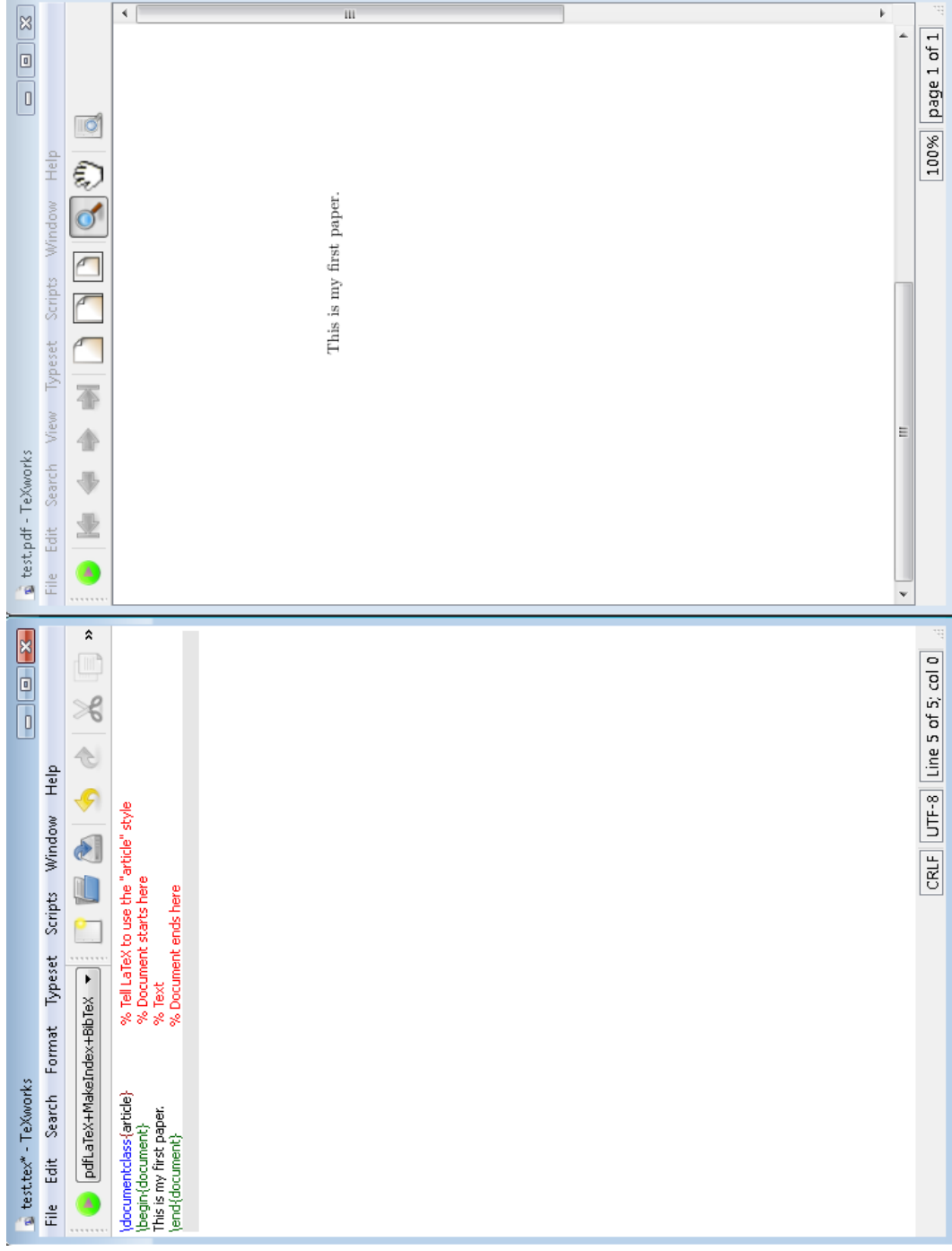


Figure 1: Simple document with TeXworks. Shows syntax highlighting and simultaneous side-by-side view of text and PDF files

**RS:** To insert graphics, use the `\includegraphics` command (see `graphics` package, Section 4.3). To create a landscape figure, use the `sidewaysfigure` environment (see `rotating` package, Section 4.11).

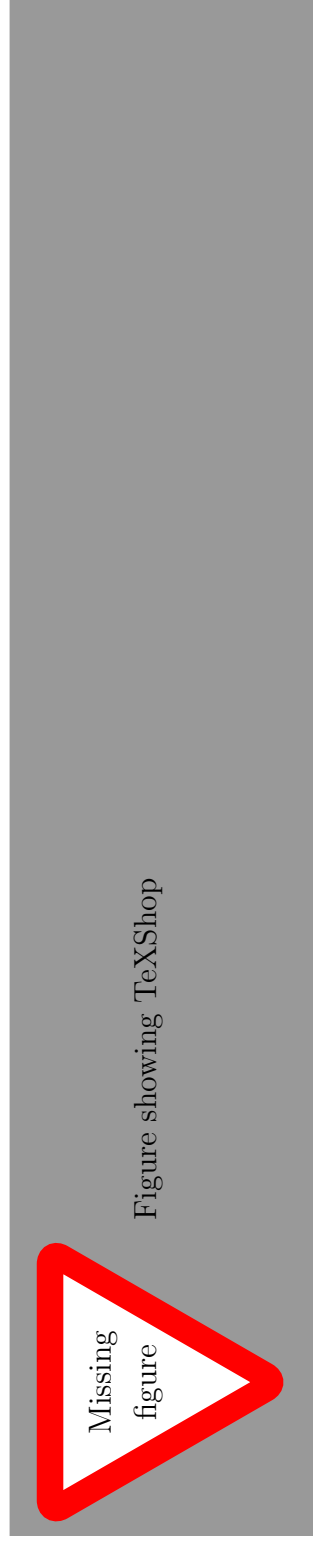


Figure 2: **Simple document with TeXShop.** Shows syntax highlighting and simultaneous side-by-side view of text and PDF files

```
\section{Section Title}
\subsection{Subsection title}
Some very uninteresting text\ldots
```

# 1 Section Title

## 1.1 Subsection title

Some very uninteresting text...

## 3.2 Mathematical expressions

L<sup>A</sup>T<sub>E</sub>X excels at typesetting mathematical expressions. There are two ways to display them:

1. **Inline:** The expression appears as part of the paragraph if surrounded by (single) dollar signs,  $\$ \dots \$$ , or (equivalently)  $\backslash( \dots \backslash)$ . For example,

```
Of course,  $e^{i\pi} + 1 =$ 
 $\backslash( \lim_{x \rightarrow 0}$ 
 $\frac{\sin(x)}{x} - 1 = 0 \backslash)$ .
```

Of course,  $e^{i\pi} + 1 = \lim_{x \rightarrow 0} \frac{\sin(x)}{x} - 1 = 0$ .

2. **Display:** The expression appears on a separate line, with more generous vertical spacing, if surrounded by  $\$ \$ \dots \$ \$$  or (equivalently)  $\backslash[ \dots \backslash]$ . For example,

```
Here is a displayed equation:

$$E = \frac{mv^2}{2}.$$

```

Here is a displayed equation:

$$E = \frac{mv^2}{2}.$$

```
Here is another:

$$\left[ \int_0^{\frac{\pi}{2}} \frac{\cos(x)}{2} dx \right]^3 =$$


$$\frac{1}{8}.$$

```

Here is another:

$$\left[ \int_0^{\frac{\pi}{2}} \frac{\cos(x)}{2} dx \right]^3 = \frac{1}{8}.$$

There are also several environments that create *numbered* equations. For example,

```
A numbered equation:
\begin{equation}
E = mc^2.
\label{eq:Einstein}
\end{equation}
```

A numbered equation:

$$E = mc^2. \tag{1}$$



Aligned, numbered equations:

```
\begin{align}
\sum_{i=1}^n i &= \\
\frac{n(n+1)}{2}; &\\
\sum_{i=1}^n i^2 &= \\
\frac{n(n+1)(2n+1)}{6}. &
\end{align}
```

Aligned, numbered equations:

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}; \quad (2)$$

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}. \quad (3)$$

To refer to equations in the text, use either `\ref` or `\eqref`:<sup>9</sup>

```
Eq.~\eqref{eq:Einstein} \ldots \\
Eq.~(\ref{eq:Einstein}) \ldots
```

Eq. (1) ...  
Eq. (1) ...

## 4 Useful L<sup>A</sup>T<sub>E</sub>X Packages

The standard L<sup>A</sup>T<sub>E</sub>X distribution comes with a large number of extra “packages.” These are files with the extension `.sty`, which allow you to customize the behavior of L<sup>A</sup>T<sub>E</sub>X, either by adding additional functionality or by changing the behavior of functions already defined. Many of these packages are extremely useful. For example, if you want to include graphics in your paper, you load the `graphicx` package (defined in the file `graphicx.sty`).

To use a package installed on your system, you use the command `\usepackage`. For example, to load the `graphicx` package, you’d include the command

```
\usepackage{graphicx}
```

in the preamble to your document. Near the top of this file you’ll find the lines

```
\usepackage{setspace,graphicx,epstopdf,amsmath,versionP0}
\usepackage{marginnote,datetime,url,enumitem,subfigure,rotating}
```

These lines load some packages that I find particularly useful, and include in almost every document I write. You can find full documentation on the Web, but here’s a brief description of why each of these packages is so useful, with some brief examples illustrating their use. Before that, however, a discussion of how to install them so L<sup>A</sup>T<sub>E</sub>X knows where to find them.

---

<sup>9</sup>The `\eqref` command and `align` environment require the `amsmath` package (see Section 4.4).

## 4.1 Installing packages

It's easy to use a package once it's installed on your system, but what do you have to do to install a package you haven't used before? The usual answer is “nothing.” Nearly all of the packages you'll ever want (including nearly all the packages described in this document) are installed as part of the L<sup>A</sup>T<sub>E</sub>X distribution.<sup>10</sup>

### 4.1.1 Installing Non-Standard Packages

From time to time, you may want to use a package that is not part of the standard L<sup>A</sup>T<sub>E</sub>X distribution. One of the packages I use on a regular basis, `versionPO`, falls into this category, and so do my customized journal-specific files, `jf.sty`, `jfe.sty`, and `rfs.sty`. One way to use packages like this is just to copy them into the directory containing your project's `.tex` file, and they will automatically be found when you process it. However, this is not a great solution, as you have to do it for *every* new project, and if you ever need to edit one of these files, it's very easy to lose track of the most recent version when you have dozens of copies at various places on your disk.

A much better method is to install these files (once) somewhere L<sup>A</sup>T<sub>E</sub>X will always know where to find them. This does require a bit of work, but it only has to be done once. Let's illustrate this with the `versionPO` package included with this document.

1. The first thing you need to decide is where to put your extra `.sty` file(s). You could put them in the same directory structure as the standard packages, but don't do this.<sup>11</sup> Instead, set up your own local directory tree. For example, on my Windows system, I store extra packages in subdirectories under `c:\RHS\texmf-local\tex\latex`, and on my Mac I use `/Users/stanton/Library/texmf/tex/latex`.<sup>12</sup>
2. After creating the relevant directory, tell your L<sup>A</sup>T<sub>E</sub>X installation where to find it.
  - MikTeX 2.09:
    - From the Windows program menu, select MikTeX 2.09 → Maintenance (Admin) → Settings (Admin).

---

<sup>10</sup>In some cases (e.g., MikTeX) you can choose not to install all packages, but to have the package manager install new packages for you whenever you ask to use them for the first time.

<sup>11</sup>When you next upgrade your installation, you'll have to remember to copy them elsewhere (which also means remembering which packages they were) to avoid losing them.

<sup>12</sup>This may seem complicated, but it follows the recommended T<sub>E</sub>X directory structure. You can pick anywhere you like for the root of this directory structure (`c:\RHS\texmf-local`, for example), but everything underneath this directory should follow the standard structure. For additional information on the standard T<sub>E</sub>X directory structure (TDS), see <http://www.tex.ac.uk/tex-archive/tds/tds.html>.

- Click on the “Roots” tab, click “Add,” and select the appropriate directory structure (in my case, this is `c:\RHS\texmf-local`).<sup>13</sup>
- TeX Live (MacTeX):
  - If you chose the same directory as I did (`~/Library/texmf/tex/latex`, where `~` denotes your own home directory, e.g., `/Users/stanton`), L<sup>A</sup>T<sub>E</sub>X will automatically look for files there and in all subdirectories, so you don’t need to do anything else here.

**Note:** Both of these steps only need to be done once for a given L<sup>A</sup>T<sub>E</sub>X installation. The next steps need to be performed for every non-standard package you install.

3. Copy any new `.sty` files into the directory structure you just created. For example, `c:\RHS\texmf-local\tex\latex\misc\versionP0.sty`.

Now run L<sup>A</sup>T<sub>E</sub>X on a file that uses the package `versionP0`. Wait! It fails with the error message

**RS:** Should work OK with the Mac; verify.

**! LaTeX Error: File ‘versionP0.sty’ not found.**

What the...?! L<sup>A</sup>T<sub>E</sub>X knows where you want to keep your extra files and you’ve put the file right where L<sup>A</sup>T<sub>E</sub>X expects to find it. This is a very frustrating error, which can cause people to get quite annoyed with the idiot coauthor who told them to install a non-standard package on their system (or so I’ve heard...). The reason for the error (which is easy to forget even for experienced L<sup>A</sup>T<sub>E</sub>X users, leading to potential frustration *every* time you install a new non-standard package) is that when you run L<sup>A</sup>T<sub>E</sub>X, it doesn’t actually search in every relevant directory to find all the files it needs to load; this would make it unbearably slow. Instead, it consults a database that tells it where to find each file. Each time you install a new file, there’s therefore one additional step:

4. Update the L<sup>A</sup>T<sub>E</sub>X file name database.

- MikTeX 2.09:
  - From the Windows program menu, select MikTeX 2.09 → Maintenance (Admin) → Settings (Admin).
  - Click on “Refresh FNDB” (**F**ile **n**ame **d**atabase).
- TeXLive (MacTeX):
  - Run the command `texhash` or `mktextlsr`.

**RS:** Shouldn’t need to do this if we chose the default location.

<sup>13</sup>This will allow L<sup>A</sup>T<sub>E</sub>X to find files anywhere under this directory.

## 4.2 Changing line spacing: `setspace`

Journals often want submissions to be double-spaced, while for most purposes I use 1.5 spacing. `setspace` is a convenient way to set or change the line spacing of the document, or parts of the document. For example, near the top of this file is the command

```
\onehalfspacing
```

This sets the overall document spacing to one and a half. You can also single-space part of the text (the abstract, say) by enclosing it in a `singlespace` environment:

```
\begin{singlespace}
```

This is single spaced. Once this environment is finished, the document will revert to its previous spacing.

```
\end{singlespace}
```

This is single spaced. Once this environment is finished, the document will revert to its previous spacing.

## 4.3 Including figures: `graphicx`

Use this package to include figures in your document. For example, here’s a picture of the MikTeX settings page described in footnote 7 on page 3.

```
\begin{figure}[H]
\centering
\includegraphics%
[width=1in]{figures/MikTeX_Options}
\caption{MikTeX main options page}
\label{fig:MikOptions}
\end{figure}
```

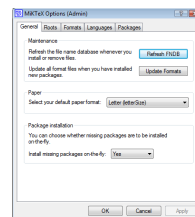


Figure 3: MikTeX main options page

### Notes

- Don’t include the extension in the file name. The package knows about several standard graphics types (in this example, I provided a PNG file). In particular, if you tell it to look for “file” then if you process the file straight to PDF using `pdfLATEX`, it will automatically look for “file.pdf” (`pdflatex` can’t handle EPS files) and if you’re processing it to a DVI file (and then to a PS file) using the “`latex`” command, it will automatically look for “file.eps” (`latex` can’t handle PDF files).

- If the `pdflatex` command cannot handle EPS files, what do you do if you don't want to convert your source file to `.dvi`, but only have EPS graphic files?<sup>14</sup> The answer is to include the `epstopdf` package, which automatically converts all EPS graphics files to PDF files on the fly when you process the source using `pdflatex`.
- There are various options for telling  $\text{\LaTeX}$  where to put a figure on the page. In this case I wanted to force  $\text{\LaTeX}$  to put the figure exactly where I defined it, so I used the `[H]` option, defined in the `float` package.

## 4.4 Mathematical typesetting: `amsmath`

Contains lots of useful extra mathematical features, fonts, etc. Some of the most useful:

### 4.4.1 `\eqref`

This puts parentheses around equation numbers for you instead of your doing it manually with `\ref`. For example,<sup>15</sup>

`Equation~(\ref{eq:Emc2}) versus \\`  
`Equation~\eqref{eq:Emc2}.`

Equation (9) versus  
Equation (9).

### 4.4.2 `\begin{align} ... \end{align}`

Use this instead of `eqnarray` for aligning multiple equations. Compare the spacing of the following equations:

`\begin{equation}`  
`E = mc^2.`  
`\end{equation}`

$$E = mc^2. \quad (4)$$

`\begin{align}`  
`E &= mc^2. \\`  
`A &= \pi r^2.`  
`\end{align}`

$$E = mc^2. \quad (5)$$

$$A = \pi r^2. \quad (6)$$

<sup>14</sup>For example, Matlab is much better at generating EPS files than PDF files.

<sup>15</sup>Yes, I realize both versions have the same number of characters, but for my two-fingered typing style it's quicker to type an extra "eq" than the two parentheses.

```
\begin{eqnarray}
E &=& mc^2. \\
A &=& \pi r^2.
\end{eqnarray}
```

$$E = mc^2. \quad (7)$$

$$A = \pi r^2. \quad (8)$$

**Note:** I recommend using `IEEEeqnarray` (part of the `IEEEtrantools` package) instead of `align` (see §4.4.3, for details on this environment and lots of useful discussion of how to typeset mathematical equations). It does seem to overcome some drawbacks of the `align` package, and maybe I'll try it out one day.... For now, here's the same example as above:<sup>16</sup>

```
\begin{IEEEeqnarray}{rCl}
E &=& mc^2. \label{eq:Emc2} \\
A &=& \pi r^2.
\end{IEEEeqnarray}
```

$$E = mc^2. \quad (9)$$

$$A = \pi r^2. \quad (10)$$

#### 4.4.3 `\begin{cases} ... \end{cases}`

Useful when an equation's right-hand side can take several forms. For example,

```
\begin{equation}
C(S) = \begin{cases}
0 & \text{if } S \leq K, \\
S - K & \text{otherwise.}
\end{cases}
\end{equation}
```

$$C(S) = \begin{cases} 0 & \text{if } S \leq K, \\ S - K & \text{otherwise.} \end{cases} \quad (11)$$

## 4.5 Setting page margins: geometry

A convenient way to set the page layout, margins, etc.<sup>17</sup> For example, to set one-inch margins on all sides, you simply type

```
\usepackage[margin=1in]{geometry}
```

This can all be done manually instead, but it takes a lot more work.

<sup>16</sup>One negative: `IEEEtrantools` is not part of the MiKTeX distribution, so you'll have to install it manually. All of my coauthors already hate me for making them learn how to install non-standard packages... (see Section 4.1.1). In addition, `IEEEtrantools` interacts badly with `enumitem` unless you load it with the command `\usepackage[retainorgcmds]{IEEEtrantools}`.

<sup>17</sup>For some reason, the default L<sup>A</sup>T<sub>E</sub>X margins are huge.

## 4.6 Inserting notes: todonotes




An extremely useful package for adding notes.<sup>18</sup> These can either be in the text or in the margin, can be colored as you desire, can appear in a global To Do list or not, can easily be switched off whenever you need to print a neat version of the paper, etc. I usually use marginal notes, and set the margins and paper size so that the layout doesn't change (much) when you turn notes on or off (see also Section 4.8).<sup>19</sup> This package has lots of different options. I define my default note command, `\rhs`, as follows:

```
\newcommand{\smalltodo}[2] [] {\todo[caption={#2}, size=\scriptsize,%
  fancyline, #1]{\begin{spacing}{.5}#2\end{spacing}}}%
\newcommand{\rhs}[2] [] {\smalltodo[color=green!30,#1]{\bf RS:} #2}}
```

The command `\rhs` produces (by default) a green margin note with a small font and tight interline spacing (to allow me to get more text into a cramped space), and inserts a nice arrow pointing to the relevant point in the text.<sup>20</sup> The note is also listed by default in the global To Do list, if there is one. This list, which summarizes the notes in the entire paper, can be inserted anywhere in your text (usually at the top) with the command `\listoftodos`:

```
\listoftodos[Richard's To Do List]
```

### Richard's To Do List

 <b>RS:</b> Describe what the paper contains.	3
Figure: Figure showing TeXShop . . . . .	6
 <b>RS:</b> Dear author, this is gibberish. . .	14
 <b>RS:</b> A red note . . . . .	14

This is useful for keeping track of large numbers of notes, but I often don't include it as it changes the layout of the paper (and the notes are pretty easy to see in the text anyway). You can easily modify the definition of or add additional options to the `\rhs` command to create different results. For example, it's easy to change the color or to make an inline note.

```
\rhs[inline,nolist]{Here's an inline note. I don't use these much as
the paper's spacing changes when you turn them off, but they're useful
if you have a really long note. This note does not appear in the To Do
list because I used the \texttt{nolist} option.}
```

<sup>18</sup>For full details, see the package documentation at <http://www.ctan.org/tex-archive/macros/latex/contrib/todonotes/todonotes.pdf>.

<sup>19</sup>You also need to include the `marginnote` package if you want to generate notes in footnotes.

<sup>20</sup>You need to run L<sup>A</sup>T<sub>E</sub>X several times to get the note and arrow correctly placed.

**RS:** (Sample note) Dear author, this is gibberish. Note that this note appears both here and in the global To Do list on page 14. Note also that I used the `caption` to create a shorter entry for the To Do list.

**RS:** Like this one, inserted with the command `\rhs`{Like this one, inserted with the command `\rhs`{Like this one, inserted...}

**RS:** A red note, inserted with the command `\rhs[color=red!30]{A red note, inserted with the command ...}`

**RS:** Here's an inline note. I don't use these much as the paper's spacing changes when you turn them off, but they're better if you have a really long note. This note does not appear in the To Do list because I used the `nolist` option.

**Note:** By default, todo notes do not work inside footnotes or floats (figures and tables). If you try, L<sup>A</sup>T<sub>E</sub>X produces an incomprehensible error message and no note. A work-around for this problem is to redefine the command `\marginpar` using the command<sup>21</sup>

```
\renewcommand{\marginpar}{\marginnote}
```

This allows notes to be created inside footnotes, but you will now find that notes on your page overlap if you create two notes close together in the text. To prevent this, my current work-around is *not* to include the command above, but instead to define a new note command `\rhsnfn`, to be used only in footnotes and floats, which performs this redefinition temporarily. This is done by the following lines at the top of this document:

```
\let\oldmarginpar\marginpar    % Save original definition of \marginpar
\newcommand{\rhsnfn}[2][{}]{%  To be used in footnotes and floats
\renewcommand{\marginpar}{\marginnote}%
\smalltodo[color=green!30,#1]{\bf RS:} #2}%
\renewcommand{\marginpar}{\oldmarginpar}}
```

## 4.7 Printing date and time: `datetime`

Allows various choices of formatting dates, as well as referring to the current time (via `\currenttime`). This is useful for mammoth editing sessions where you've created printouts of several different versions of a document during the same day.

## 4.8 Multiple versions from one file: `versionPO`

This package allows you to include text in your paper conditionally, so you can generate multiple different versions of a document from a single source file by changing just one or two statements.<sup>22</sup> here are many, *many* uses for this package, including

<sup>21</sup>This requires you to load the `marginnote` package.

<sup>22</sup>One drawback of `versionPO.sty` is that it's not included as part of the standard L<sup>A</sup>T<sub>E</sub>X installation. I've therefore included it with this document, and it needs to be installed as described in Section 4.1.1. `versionPO.sty` was written by Piet van Oostrum in about 1991 and originally named `version.sty`. I renamed it `versionPO.sty` to avoid any possible conflicts with a different file, also called `version.sty`, that *is* in the standard distribution. There are several standard packages that do similar things to this one, including `version`, `versions`, `optional`, and `comment`. You should feel free to use one of them instead, but I am used to using `versionPO` and don't like change.



- Creating working papers and journal submissions from a single source file.
- Optionally including responses to referees.
- Adding notes to yourself that you can switch on or off at will.
- Writing an exam and having the option to add extra space for students to write their answers, or having the option to include solutions, again all in a single file.

Here's a simple example. See what happens to the output below when you change the command `\includeversion{notes}` to `\excludeversion{notes}` at the top of the file.

```
Notes \ifnotes{\emph{are}}}%
{are \emph{not}} shown in this file.
```

Notes *are* shown in this file.

A slightly more interesting example is the following, from the top of this file:

```
\ifnotes{%
\usepackage[margin=1in,paperwidth=10in,right=2.5in]{geometry}%
\usepackage[textwidth=1.4in,shadow,colorinlistoftodos]{todonotes}%
}%
\usepackage[margin=1in]{geometry}%
\usepackage[disable]{todonotes}%
}
```

If the command `\includeversion{notes}` appears at the top of this file, then this command loads `todonotes` with various options, and then loads `geometry` with some slightly strange paper size and margin definitions that allow both the text and the margin notes to be printed on a standard piece of paper without changing the text layout. If the command `\excludeversion{notes}` appears instead, notes are disabled and standard one-inch margins are used, producing a version of the document ready for distribution. One environment defined by default (and excluded) is `comment`, so you can comment out text by either surrounding it with `\ifcomment{...}` or (useful for larger blocks) putting it inside `\begin{comment} ... \end{comment}`. This is useful for temporarily removing text that you might want to put back later. For example,

```
This is the first sentence.
\begin{comment}
The second sentence is commented out.
\end{comment}
This is the third sentence.
```

This is the first sentence. This is the third sentence.

Here are some more examples from the top of this document. Try changing their settings to see what happens when you reprocess the document.

**RS:** To maximize how much I can say, I use a small font and the widest margin notes I can get my printer to generate.

```

\includeversion{notes}           % Include notes?
\includeversion{links}           % Turn hyperlinks on?
\excludeversion{submit}          % Format for conference submission?
\includeversion{toc}             % Include table of contents?

```

## 4.9 List formatting: enumitem

`enumitem` enables easy control over many aspects of numbering and spacing in `enumerate`, `itemize`, and `description` lists. For example, at the top of this file you'll see the command

```
\setlist{noitemsep}
```

This removes all extra vertical space between list items, which I prefer to the default spacing (by default,  $\text{\LaTeX}$  puts extra vertical space between list items). You can also use this package to change the indentation of a list. For example,

Here is an unindented itemized list:

```

\begin{itemize}[leftmargin=0]
\item Item 1.
\item Item 2.
\end{itemize}

```

Here is an unindented itemized list:

- Item 1.
- Item 2.

## 4.10 URLs and hyperlinks: hyperref

This package allows you to include clickable hyperlinks in your paper for URLs, citations, equations, references, etc. For example,

```

\cite{Stanton:95} \\\
\url{http://www.ctan.org/} \\\
\href{http://www.ctan.org/}%
{CTAN Web site} \\\
Equation~\eqref{eq:Emc2} \\\
Section~\ref{sec:todo}

```

?  
<http://www.ctan.org/>  
CTAN Web site  
Equation (9)  
Section 4.6

You can customize how (and if) these are displayed in your text (colored text, boxes, etc.). There are also unlinked versions of some commands, e.g.,

```
\nolinkurl{http://www.ctan.org/}
```

<http://www.ctan.org/>

If you don't want any hyperlinks in your document, you can turn them all off with the command `\hypersetup{draft=true}`.<sup>23</sup> You can try this out by changing the command `\includeversion{links}` to `\excludeversion{links}` at the top of this document.

## 4.11 Landscape and subfigures: rotating and subfigure

The `rotating` package allows the creation of tables and figures in landscape mode using `\begin{sidewaystable} ... \end{sidewaystable}` for tables and `\begin{sidewaysfigure} ... \end{sidewaysfigure}` for figures.<sup>24</sup> The `subfigure` package allows creation of a single figure with multiple subfigures. Figure 4 illustrates the use of both `rotating` and `subfigure`, created using the command

```
\begin{sidewaysfigure}
\begin{center}
\subfigure[A somewhat familiar figure]{
  \includegraphics[width=3.5in]{figures/MikTeX_Options}
  \label{fig:sub1}
}
\subfigure[That was fun. Let's plot it again!]{
  \includegraphics[width=3.5in]{figures/MikTeX_Options}
  \label{fig:sub2}
}
\end{center}
\caption{This completely gratuitous figure plots the same graphic
we saw earlier, but does so twice and in landscape mode. How exciting!
}
\label{fig:fig_subfig}
\end{sidewaysfigure}
```

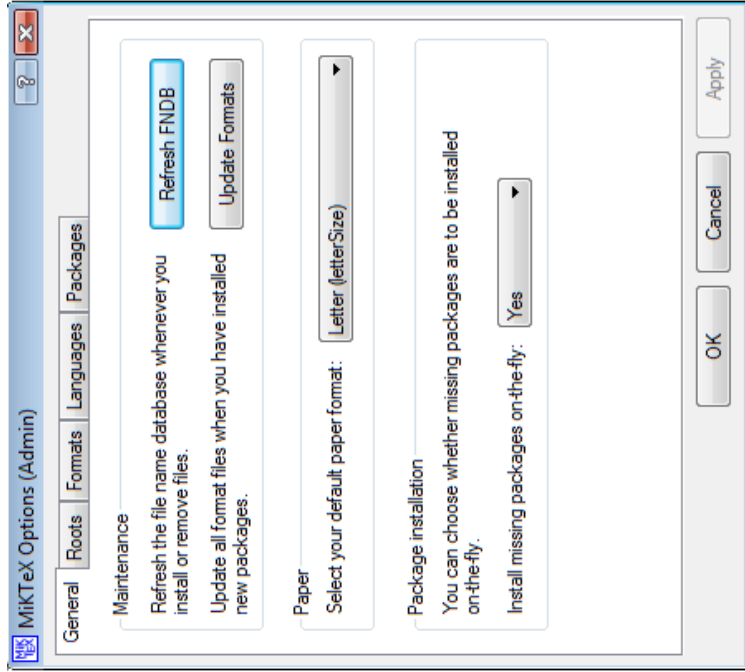
Note that you can also refer to the whole figure or to individual subfigures in the text, e.g.,

`Figure~\ref{fig:fig_subfig}. \\\`  
`Figure~\ref{fig:sub2}.`

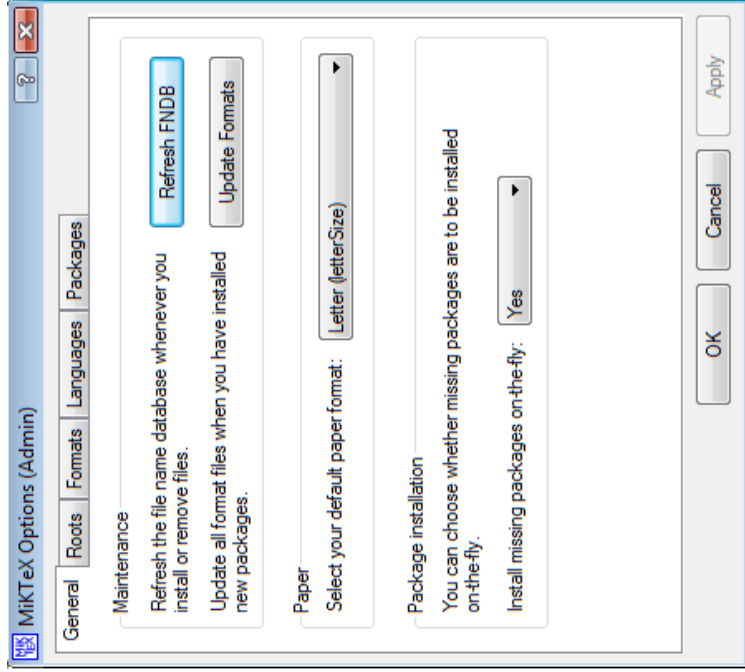
Figure 4.  
 Figure 4(b).

<sup>23</sup>An alternative is to use the `url` package, but it is less flexible than `hyperref`.

<sup>24</sup>Note that these are placed on separate pages. This is usually what you want, as you're rotating the figure or table because it's too large to be set in portrait mode. However, there is also the `sideways` environment, which allows you to have rotated text/figures on the same page as non-rotated material.



(a) A somewhat familiar figure



(b) That was fun. Let's plot it again!

Figure 4: This completely gratuitous figure plots the same graphic we saw earlier, but does so twice and in landscape mode. How exciting!

## 4.12 Other useful packages

Though I don't use these in every document, here are a few other packages I sometimes find useful:

- **bibunits**: Allows you to create multiple bibliographies in a single document. For example, you might be teaching a class and want a single document with a section on each of a number of topics and a separate bibliography per section. You can optionally also have a single global bibliography. Each bibliography can have a different format.
- **tocvsec2**: Allows you to control which level of section appears in the table of contents, and you can change this section by section. For example, you might want to list subsections from the main body of the paper, but only sections from the Appendix.
- **lastpage**: Allows you to refer to the number of the last page of the document.
- **tikz**: Very powerful environment for creating diagrams.
- **indentfirst**: Indents the first line of a new section (JF asks for this).
- **caption**: Gives you easy control over the format of figure and table captions. Useful for satisfying journal requirements.
- **xr**: Allows you to refer in one document to labels in another document. Useful, for example, for creating an Internet Appendix for the Journal of Finance where you want to be able to refer to equation numbers in the main paper.<sup>25</sup>
- **amsthm**: Allows good control over theorem-like environments.
- **booktabs**: Allows good control over table layout and spacing.
- **longtable**: Allows a single table to span more than one page, keeping the same column headings, etc.

## 5 Citations and Bibliographies

L<sup>A</sup>T<sub>E</sub>X makes it very easy to create and format citations and bibliographies. The most basic method is to insert the bibliography manually into your paper using the `thebibliography` environment and one `\bibitem` command for each entry. E.g.,

```
\begin{thebibliography}{99}
\bibitem{Stanton:95} Stanton, Richard, 1995, Rational prepayment and the
value of mortgage-backed securities, {\em Review of Financial Studies}\}
8, 677--708.
\end{thebibliography}
```

---

<sup>25</sup>This could alternatively be done by putting everything in a single file and using `bibunits` to generate the two (separate) bibliographies, but then you'd need to split the resulting PDF file in half. It's your choice.

This entry would be cited in the text using the command `\cite{Stanton:95}`.

This is how bibliography creation is described in ? and ? (though the latter does mention using BibTeX), but **don't do it this way!** Major drawbacks of this manual method include:

1. If you remove a citation from your text, you need to remove that paper from the bibliography manually (after, of course, checking to make sure you're not still citing it somewhere else).
2. To match a journal's bibliography style, you have to edit every entry by hand.
3. Since there's no central database of references, you typically end up having to retype (after re-Googleing) the same citations over and over again in different papers. This is both time-consuming and error-prone.
4. By default, L<sup>A</sup>T<sub>E</sub>X produces numerical citations, whereas most finance journals want author-year citations, e.g., ?.

## 5.1 BibTeX

The solution to the problems above is to create bibliographies using BibTeX, a separate program installed as part of any standard L<sup>A</sup>T<sub>E</sub>X installation.<sup>26</sup> Compared with the manual method,

1. BibTeX automatically inserts only references you actually cite (or that you tell it to insert even though you don't actually cite them, using the `\nocite` command). If you delete all citations to a particular paper, it will automatically be removed from your bibliography.
2. Reformatting a bibliography usually requires editing just a single command in your paper (see Section 5.4.2).
3. You keep all references in a central database. As a result,
  - (a) You only need to type or look each reference up once.
  - (b) There's no doubt about where to find the latest version of any reference.
  - (c) Once you've corrected any errors in a reference, they remain corrected forever.
4. It's easy to generate citations in any format you desire, especially author-year format, using just the standard `\cite` command (see Section 5.4.1).

---

<sup>26</sup>One day, it will probably be even better to use the `bibtex` package instead. This has some more features than BibTeX (for example, you can have citations appear in footnotes, and can refer to the title of a reference within your text), and it is in principle easier to customize, but it is *currently* easier to generate journal-specific bibliography formats using BibTeX with `custom-bib`.

## 5.2 The BibTeX database

When using BibTeX, you store all your references in one or more .bib files, taking the following format:

```
@ARTICLE{Stanton:95,  
  author = {Richard Stanton},  
  title = {Rational Prepayment and the Value of Mortgage-Backed Securities},  
  journal = {Review of Financial Studies},  
  year = {1995},  
  volume = {8},  
  pages = {677--708},  
  number = {3}  
}
```

Note that this entry contains all of the logical information about the reference (author, title, etc.), but says nothing about formatting, which is handled separately. Once you've entered a reference in your .bib file and told L<sup>A</sup>T<sub>E</sub>X and BibTeX where to find it (see below), when you use the `\cite` command, BibTeX will automatically find the right reference, add it to your bibliography, and insert the appropriate citation in the text, e.g.,

`\cite{Stanton:95}`

?
---

Just like a .tex file, a .bib file is a plain-text file and can be edited with the same editor you use to edit your .tex files. However, there are also some BibTeX-specific editors, including JabRef (available on multiple platforms from <http://jabref.sourceforge.net>) and BibDesk (OS X only, installed as part of the MacTeX distribution).

## 5.3 Telling BibTeX where to find the database

You tell BibTeX to look for references in the file `master.bib` by putting the following line in your TeX file:

```
\bibliography{master}
```

**Note:** BibTeX will always find the file `master.bib` if it is in the current directory, but you don't want to have lots of different versions of this file sitting in every different project directory. A better approach is to set the environment variable `BIBINPUTS` to point to the directory in which you've stored your Bib file, and then BibTeX will find it no matter where you run it from.

- On my Windows machine, BIBINPUTS is set to `c:/RHS/texmf-local/bibtex/bib//`, which will look for Bib files in the current directory and then, failing that, in directory `c:\RHS\texmf-local\bibtex\bib` and in all subdirectories.<sup>27</sup>
  - To set an environment variable in Windows, go to Control Panel and select System → Advanced → Environment Variables. Then under “System Variables,” click “New,” enter the variable name (BIBINPUTS) and the variable value (`c:/RHS/texmf-local/bibtex/bib//`), and click OK a few times.
- On my Mac, BIBINPUTS is set to `./:/Users/stanton/tex/texmf-local/bibtex/bib//`, which will look for Bib files in the current directory and, failing that, in the directory `c:\RHS\texmf-local\bibtex\bib` and in all its subdirectories.
  - To set an environment variable in OS X, edit file `~/.profile` and add the line<sup>28</sup>

```
export BIBINPUTS=./:/tex/texmf-local/bibtex/bib//
```

## 5.4 Formatting the bibliography: natbib and custom-bib

### 5.4.1 natbib

This package provides finer control than basic L<sup>A</sup>T<sub>E</sub>X over citation formatting. For example,

```
\cite{Stanton:95} \\  
\citep[see, for example,]%  
    [p. 3]{Stanton:95} \\  
\citet[Equation~1]{Stanton:95} \\  
\citealt[Equation~1]{Stanton:95} \\  
\citealp[Equation~1]{Stanton:95} \\  
\citeauthor{Stanton:95} \\  
\citeyear{Stanton:95}
```

```
?  
(see, for example, ?, p. 3)  
?, Equation 1  
?, Equation 1  
?, Equation 1  
?  
?
```

By loading the extra package `bibentry`, you can also print the full reference. For example,

```
\bibentry{Stanton:95}
```

<sup>27</sup>Searches in Windows almost always look in the current directory first, so this doesn’t need specifying.

<sup>28</sup>Strictly speaking, this only has an effect if you’re running L<sup>A</sup>T<sub>E</sub>X from a command shell. If run from within a GUI application, this may not work. If this is a problem for you, set the environment variable instead using the command `launchctl setenv BIBINPUTS ./:/tex/texmf-local/bibtex/bib//`.



### 5.4.2 Customizing the bibliography format: custom-bib

This package allows you to define a new bibliography style ( a .bst file), matching the format of your bibliography to that of the journal you are submitting to.<sup>29</sup> To create a new bibliography style called, say, `jf.bst` for the *Journal of Finance*, run the command

```
latex makebst
```

and then answer various questions about the name of the output file and how you want your bibliography formatted. It then creates a customized .bst file, `jf.bst`. To use the format defined in this file, you insert the command

```
\bibliographystyle{jf}
```

in your document. This tells L<sup>A</sup>T<sub>E</sub>X/BibT<sub>E</sub>X to format the bibliography using information in the file `jf.bst`.

## 6 Journal-Specific Document Formatting

Journals typically have their own requirements for section numbers, figure captions, etc., as well as for the format of the bibliography. Meeting these requirements requires you to modify some of the default definitions.

### 6.1 Formatting the bibliography

I've used `custom-bib` to create bibliography styles for

- *Journal of Finance* (**jf.bst**),
- *Journal of Financial Economics* (**jfe.bst**),
- *Review of Financial Studies* (**rfs.bst**).

These files are included with this document. Change the command `\bibliographystyle{jf}` below to `\bibliographystyle{jfe}` or `\bibliographystyle{rfs}`, reprocess the document (by running LaTeX, BibTeX, and LaTeX again), and see how the bibliography format changes.<sup>30</sup>

---

<sup>29</sup>You can alternatively create or edit the appropriate .bst file by hand. However, before you even think of doing this, take a look at one of the .bst files in your L<sup>A</sup>T<sub>E</sub>X installation. These use a mysterious syntax that is most definitely *not* fun to edit...

<sup>30</sup>The bibliography contains references to published papers (?), books (?), and working papers (?), so you can see how various entry types are formatted.

**Note:** You’ll typically find one or two minor journal bibliography conventions that custom-bib can’t quite handle.<sup>31</sup> You now have two choices. One is to spend hours manually editing the .bst file to do what you want. The other (which I recommend) is to wait until the bibliography is really final, then just copy the contents of the .bbl file into the body of the text (removing the `\bibliographystyle` and `\bibliography` commands), and edit it manually to make these last few changes.<sup>32</sup>

## 6.2 Formatting the text

In addition to the bibliography format files mentioned above, I have also have written style files to help make your paper match the submission requirements of either the *Journal of Finance* (`jf.sty`), *Journal of Financial Economics* (`jfe.sty`), or *Review of Financial Studies* (`rfs.sty`). I have also included a sample “paper” for each journal, which you can use as the basis for preparing your own documents. They include similar packages to this document, but each also includes the journal-specific style file and bibliography format file, and also has some journal-specific formatting commands in the text. The specific files included for each journal are:

### 1. Journal of Finance:

- `jf.sty`: Formatting commands for papers.
- `jflA.sty`: Formatting commands for Internet Appendices.
- `jf.bst`: Bibliography format.
- `jfsample.tex`: Sample paper (T<sub>E</sub>X file).
  - PDF file.
- `jflAsample.tex`: Sample Internet Appendix (T<sub>E</sub>X file).
  - PDF file.

### 2. Journal of Financial Economics:

- `jfe.sty`: Formatting commands for papers.
- `jfe.bst`: Bibliography format.
- `jfesample.tex`: Sample paper (T<sub>E</sub>X file).
  - PDF file.

### 3. Review of Financial Studies:

---

<sup>31</sup>For example, you may have page numbers listed in your BibT<sub>E</sub>X database as “1023–1045”, while the journal insists on “1023–45”.

<sup>32</sup>Save even more time by doing this *after* the copy-editor has taken a look and pointed out where you need to make changes.

- rfs.sty: Formatting commands for papers.
- rfs.bst: Bibliography format.
- rfssample.tex: Sample paper (T<sub>E</sub>X file).
  - PDF file.

## 7 Working with Others: Version Control Systems

When working on a paper with other people, you have to be careful to keep track of who is working on what version of each file associated with the paper (both .tex and program files, e.g., Matlab code). One common way to do this is to edit the files in turn, making sure that each person only starts editing a file once the prior person in line has emailed them a version containing all of their changes. This works most of the time, so it isn't an absolutely terrible idea, but it has some significant drawbacks. For example,

- It involves a lot of emailing.
- With multiple files, it's easy to forget who's supposed to be editing which file.
  - What happens when your coauthor accidentally edits the version of the file you sent her 4 days ago instead of the version containing all your latest edits?
- What happens when you remember there was a section (or a Matlab function) in the version you presented at the WFA meetings 6 months ago that's now gone from the file, but which you'd like to retrieve?
- What happens when you've spent a week editing your program, and realize the direction you were going in just doesn't work, so you want to go back to the version right before you started these edits?
- What happens when you realize that at some (unknown) time during the last month, one of your coauthors accidentally deleted a section of the paper you really liked?
- What happens when you realize that your code now produces different results than it did a year ago, but shouldn't? How do you track down what changes to the code during the last year caused the change in behavior?

A good back-up system (e.g., Time Machine) will help by allowing you to retrieve old versions of your files.<sup>33</sup> Using Dropbox or some similar file-sharing software is also very helpful in

---

<sup>33</sup>But how do you remember which version from about 6 months ago you actually need? This problem is commonly "solved" by saving lots of versions of your paper and code with clever descriptive names like

- program.WFA2011.m
- program.remember\_this\_as\_I\_might\_want\_to\_go\_back\_to\_this\_version.m
- program.XXX.m

solving the constant-emailing problem, and it also means that everyone knows that the latest version of the file is always stored in the shared Dropbox directory (at least as long as whoever edited it last has remembered to copy it from their working directory to the Dropbox directory...). For these and many other reasons I strongly recommend that everyone back up their important files on a regular basis. I also love Dropbox. However,

- Suppose you forget that it's not your turn to edit a file (or you come up with an algorithm or way of expressing a thought that's so brilliant you can't possibly wait to commit it to paper—or a .tex file). So now both you and your coauthor are editing the file at the same time.
- Now suppose you save your edits on Dropbox first and then your coauthor saves her edits. Your edits have now disappeared from the latest version of the file (which is quite a shame, given how brilliant they were).

More generally, wouldn't it be nice not to have to worry about editing the file sequentially, so any of the coauthors could edit it whenever they had time or a good idea (or both), without worrying about losing track of the “current” version of the file or of their or anyone else's edits? And wouldn't it be nice if you could at any time easily go back to any prior version of any file? A **Version Control System** (VCS) solves all of the problems we discussed above. Among many advantages,

- It makes it easy to merge simultaneous edits by different coauthors, so you don't have to worry any more about editing files sequentially. Edit whatever file you want whenever you want.
- It allows you to keep track of every revision of every file that's ever existed, along with a log of what changed from one revision to the next. If you want to go back to a prior version, or see what's changed since a version 2 years ago, it's almost trivial.

The second advantage makes using a VCS a good idea even when there's only one author, but it's particularly valuable when there is more than one.<sup>34</sup>

An in-depth discussion of VCS is beyond the scope of this document. Some excellent introductions, which explain way better than I could what version control is, why it's important, and how to use it, are ?, ?, and ?.<sup>35</sup> There are numerous VCS to choose

---

However, after a while there are so many files that you can't find the one you want in the crowd, and you'll eventually forget what those names, which seemed so clear when you first came up with them, mean.

<sup>34</sup>I've been using a VCS to keep track of every .tex file I care about for over 20 years, starting with RCS (a “first generation” VCS), then switching to CVS (a “second generation,” centralized VCS, or CVCS), and more recently Mercurial (a “third generation,” distributed VCS, or DVCS).

<sup>35</sup>? covers several different VCS, including Subversion, Mercurial, and Git. ? and ? focus exclusively on Mercurial.

from, but my personal recommendation if you're starting from scratch is to use one of the three most popular DVCS, Mercurial (see <http://mercurial.selenic.com>), Git (see <http://git-scm.com>), or Bazaar (see <http://bazaar.canonical.com>).<sup>36</sup> With these systems, each author edits and keeps track of his or her changes locally, and periodically “pushes” those changes to a shared central repository.

## 7.1 Public hosting of DVCS repositories

At this point, many of you are probably starting to sweat, worried that your coauthors will ask you to be the person who manages and hosts the shared repository.<sup>37</sup> Fortunately, there are public hosting services that make it extremely simple to host your central repository, as well as giving you the ability to keep track of bugs, access your files from anywhere there's a Web connection, etc. Examples include GitHub (<https://github.com/>: Git), BitBucket (<https://bitbucket.org/>: Mercurial and Git), and Launchpad (<https://launchpad.net/>: Bazaar). You can host either public (open to everyone) or private (accessible only to those you explicitly designate) repositories on these sites.

## 7.2 This project's Bitbucket repository

The central (Mercurial) repository for this project is publicly available on Bitbucket at <https://bitbucket.org/rhstanton/texintro>. Feel free to clone this repository and experiment with it (and Mercurial).<sup>38</sup>

**Note:** I know you probably won't all start using a VCS tomorrow. I've been extolling their virtues for years, and so far I've managed to get one coauthor to admit—grudgingly—that they might think about using one in about 150 years and the rest of them no longer return my calls or emails. Using a VCS is still a good idea. Really...!

---

<sup>36</sup>While there are loud arguments on the Web about which of these is the best, the similarities are way more striking than the differences. I chose Mercurial because Git and Mercurial seem to have more users than Bazaar and because Mercurial and Bazaar play slightly nicer with the standard Emacs version-control packages than Git (a factor that probably won't play much of a role in your decision).

<sup>37</sup>Can you remind me again how to set up a password-protected Apache Web server on my Mac...?

<sup>38</sup>For details of how to set up an account at Bitbucket and how to interact with Bitbucket repositories, see ? and ?.