Preparing Advanced Manuscripts with LaTeX

Garrett Darl Lewis*
Department of Politics
Princeton University

March 25, 2015

Abstract

This is an interactive document designed to introduce seminar participants to the features and capabilities of LaTeX. After working through this document, users should be able to write a basic text document with equations, graphics, tables, code, and a professional-looking bibliography. All of the files related to this document are available at my website, along with links to other helpful documents and resources. To get the most out of this document, users are encouraged to read raw code alongside the finished document using an appropriate LaTeX development environment, such as *Overleaf* or *TeXworks*.

*Department of Politics, Princeton University 130 Corwin Hall, Princeton, NJ 08544-1012 E-mail: glewis@princeton.edu,

Contents

1 Introduction

LaTeX is a particularly versatile typesetting environment. I am assuming that you are already familiar with the basics of the software and terminology, so we will primarily be focusing on expanding your skill set and introducing more advanced features. Of course, to gain a full appreciation of the capabilities of the software takes practice and effort beyond this seminar, but I hope to provide tools that will be of use throughout the learning process and allow you to proceed independently in your own work.

This seminar will proceed in six parts as follows:

1. **Text**

TEXample 1: Theorem

Theorem 1. *All governments are unjust.*

Proof. Consider an arbitrary government. Since it is arbitrary, it is obviously unjust. The assertion being correct for an arbitrary government, it is thus true for all governments. □

2. **Code**

TEXample 2: Code Snippet

```
% Don't do this!
i=1
while i>0
     i++
end;
```

3. **Bibliographies**

TEXample 3: Book Citation

Lewis, Darl. 2014a. *A LaTeX Book*. NJ: My Computer.

4. **Math**

TEXample 4: Multiline Equation

$$
\begin{aligned}
a + b &= c \\
2(a + b) - (a + b) &= 2c - c \\
2a + 2b - 2c &= a + b - c \qquad \text{(Bad Math)} \\
2(a + b - c) &= a + b - c \\
2 &= 1
\end{aligned}
$$

5. Tables

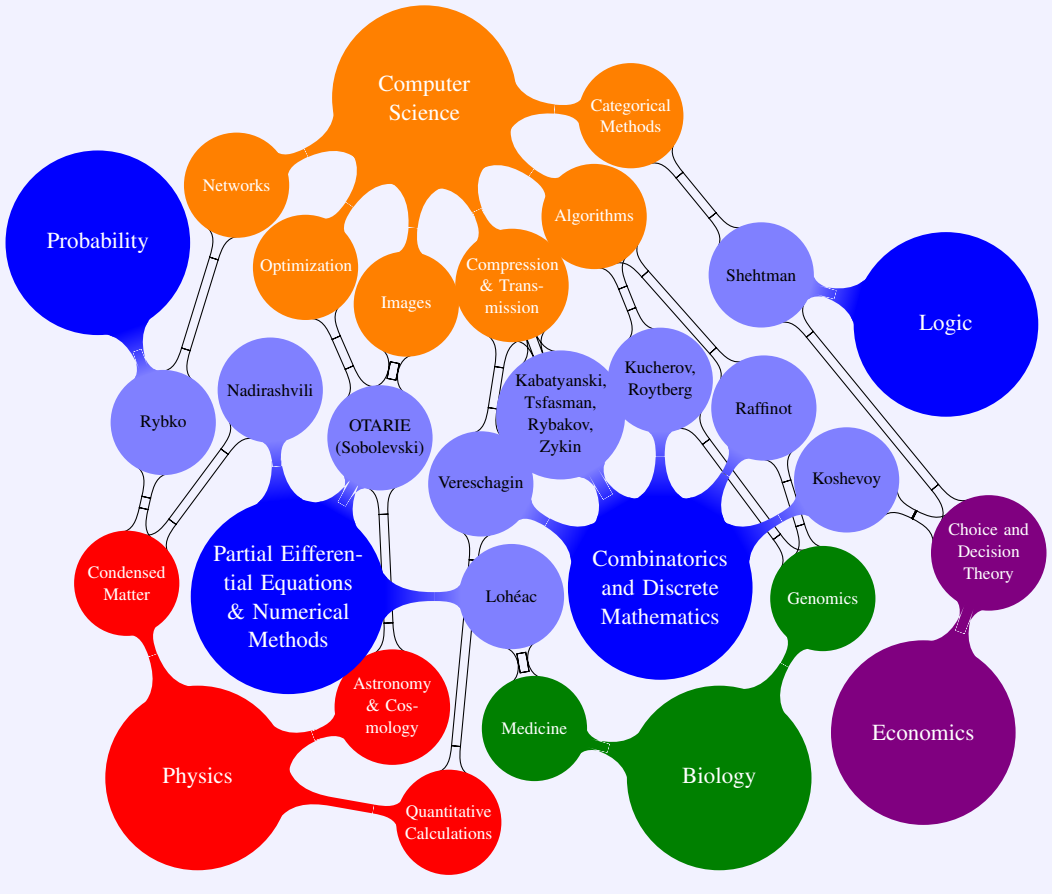
TEXample 5: Regression Output

	Model 1		Model 2	
	(1)	(2)	(1)	(2)
Weight (lbs.)	1.747*	.008	4.614*	.000
(p-value)	(–.050)	(.567)	(.263)	(.020)
Foreign*Mileage			–.307	.006*
			(.240)	(.000)
Controls	N	Y	N	Y
Adj. R^2	.273	.435	.526	.773
No. of cases	74	74	74	74

Source: auto.dta

6. Graphics

TEXample 6: Mindmap



2 The Main Body

The following code is sufficient to produce a basic document. In the preamble, it defines the type of document being created (*documentclass*), any special packages that might be required to compile the document (*usepackage*), as well as a statement of the title and author of the work. As I introduce more sophisticated capabilities of the software, I will also discuss more packages and settings that can be called in the preamble when they are needed.

At the end of the preamble, the command, `\begin{document}`, begins the main body. After this command, I tell the software to include a title, table of contents, and an abstract. I then insert the main text, and finally call a bibliography using references written to the file, *CiteFile.bib*. The command, `\end{document}` ends the document. After saving, I typically run three commands in my compiler to prepare the document and ensure that all of the auxiliary files have been properly updated. These three commands are *pdfLaTeX*, *MakeIndex*, and *BibTeX*. It is generally not necessary to run all three, but as it is better than spending hours on end trying to figure out why your document is not compiling properly, only to find out that your **.bbl* file was out of date.

Code Block 1: Basic Structure

```
\documentclass[letter,12pt]{article}
%There are several packages which I use throughout the file.
Not all will be helpful in every document, but the first several
included here are almost always useful.  I try to note the
relevant heading material in the body below as I use it.

% Important packages
\usepackage{latexsym, amsmath, amssymb, amsthm, natbib, setspace}

% Define the title, \&c
\title{My Title}
\author{Me\thanks{Department of Politics\\ Princeton University}
\date{\today}

\doublespacing  % Use double spacing

\begin{document}    % Begin the body
```

```
\maketitle      % Generate the title
\tableofcontents    % Generate a ToC

\begin{abstract}
Put your abstract here!
\end{abstract}

\section{The Body}
Hello World!

%\newpage   % Skip to the next page
\bibliography{BiblioTemp} % Insert a bibliography
\end{document}
% Everything down here is ignored
```

3 Text

Basic text is pretty easy. Just define your document class, open your document, and start typing, maybe adding a chapter or section along the way. However that's pretty boring, especially if you have to use things like, *you know*, **MATH**. Let's look at some of the options...[1]

In the previous paragraph, we saw features like italics and bold face type, as well as an ellipsis and a footnote. These are typeset as,

Code Block 2: Text

```
...have to use things like, \emph{you know}, \textbf{MATH}.
Let's look at some of the options\dots\footnote{I want to
quickly point out a few features that can drive someone crazy if
they are not paying attention.  Many symbols, such as \&,
\textasciitilde, and \textbackslash\ require special notation in
\LaTeX , since the symbols themselves have certain meaning in
the software.  Make sure you proofread your finished
document--esp. where you are likely to use symbols--to ensure
```

[1] I want to quickly point out a few features that can drive someone crazy if they are not paying attention. Many symbols, such as &, ~, and \ require special notation in LaTeX since the symbols themselves have certain meaning in the software. Make sure you proofread your finished document–esp. where you are likely to use symbols–to ensure that there are no odd missing items or strange additions.

```
that there are no odd missing items or strange additions.}
```

3.1 Lists

Writing lists in LaTeX is relatively easy using the *itemize* and *enumerate* commands. Here I am using the two commands to format my description of their capabilities.

1. *Itemize* produces a list of bullet points.
    - This is what you do if you do not care about numbering.
    - The bullet shape changes at every level
        - Like this.
2. This is what you do if the order matters.
3. But you can also choose your own shape using the *enumitem* package.

    i) Like this.

Code Block 3: Lists

```
\begin{enumerate}
\item \emph{Itemize} produces a list of bullet points.
\begin{itemize}
\item This is what you do if you do not care about numbering.
\item The bullet shape changes at every level
\begin{itemize}
\item Like this.
\end{itemize}
\end{itemize}
\item This is what you do if the order matters.
\item But you can also choose your own shape using the
\emph{enumitem} package.
\begin{enumerate}[label=\roman*)]
\item Like this.
\end{enumerate}
\end{enumerate}
```

TEXercise 1: List

Problem 1. *List your five favorite states in descending order. List two cities in your favorite state.*

Hint.

Code Block 4: Nested List

```
\begin{enumerate}
\item Missouri
\begin{itemize}
\item Kansas City
\item St. Louis
\end{itemize}
\item Florida
\item Texas
\item California
\item Maryland
\end{enumerate}
```

□

TEXercise Solution 1: List

Solution. YOUR SOLUTION ■

3.2 Theorems

There are a few environments readily available for writing theorems and proofs and the like, but sometimes you have to define your own. Luckily, that's easy to do. In the preamble of my document, I have defined environment, *Theorem*, using this command:

Code Block 5: Theorem

```
\newtheorem{theorem}{Theorem}
```

LaTeX counts each environment separately unless specified otherwise using any of the *counter* tools such as *addtocounter* and *newcounter*.

Theorem 2. *All governments are unjust.*

Proof. Consider an arbitrary government. Since it is arbitrary, it is obviously unjust. The assertion being correct for an arbitrary government, it is thus true for all governments. □

Code Block 6: Theorem and Proof

```
\begin{theorem}
All governments are unjust.
\end{theorem}

\begin{proof}
Consider an arbitrary government. Since it is arbitrary, it is
obviously unjust.  The assertion being correct for an arbitrary
government, it is thus true for all governments.
\end{proof}
```

I did not have to separately define the *proof* environment used here, as it is included with the *amsthm* package. However, below I have written an exercise which uses, among others, a *solution* environment that is slightly different from the *proof* environment, using a black box instead of an empty one. Creating this environment is somewhat more involved that creating *theorem*:

Code Block 7: Solution

```
\newenvironment{solution}{\begin{trivlist} \item
\textbf{Solution}. }{ \hspace*{\fill}
$\blacksquare$\end{trivlist}}
```

3.3 Counters

When you define a new basic environment LaTeX automatically produces a counter with the same name and increments that counter every time you use the environment. Sometimes, though, you may want to use a different numbering system. The code below defines two new environments,

one of which uses the same counter as the theorem environment, and one of which is numbered by section:

Code Block 8: Theorem

```
\newtheorem{corollary}[theorem]{Corollary}
\newtheorem{definition}{Definition}[section]
```

Note, in some more advanced contexts, you may need to define a new counter yourself using the *newcounter* command.

TEXercise 2: Question

Problem 2. *Generate a* Question *environment identical to the* Theorem *environment. Then create an* Answer *environment that looks like the* Proof *environment, but uses a star instead of an open box. Set the counter on* question *to be the same as that on* problem*. Finally, ask yourself a question and then answer it.*

Hint.

Code Block 9: Question

```
\newtheorem{question}[problem]{Question}
\newenvironment{answer}{\begin{trivlist} \item
\textit{Answer}. }{ \hspace*{\fill} $\star$\end{trivlist}}

\begin{question}
Is this correct?
\end{question}

\begin{answer}
Yes, it is.
\end{answer}
```

□

TEXercise Solution 2: Question

Solution. YOUR SOLUTION ■

3.4 Importing Other Files

Sometimes with large projects, you may wish to separate the content into multiple files. This is particularly true of book projects and dissertations. Likewise, large Ti*k*Z graphics (covered later) may warrant a separate document. To compile these separate files as part of a single unified document, use the *input* or *include* commands. One disadvantage of these commands, however, is the need to return to the parent document to compile your work. To get around this, you may wish to use the *subfiles* package, which allows you to compile each chapter individually, which can give you a significant boost if you are trying to quickly compile one section of a larger project.

TEXercise 3: Import

Problem 3. *Import the file,* ToImport.tex.

Hint.

Code Block 10: Import

```
1 \input{ToImport.tex}
```

□

TEXercise Solution 3: Import

Solution. YOUR SOLUTION ■

3.5 Code

For simple snippets of code, the *verbatim* package may work. This package is also useful for the *comment* command which allows you to quickly block out large blocks of code and avoid the compiler. When you are interested in presenting longer blocks of code, however, as I have done throughout this document, *listings* provides much more power. In the file, *NewCommands.tex*, which I loaded into the preamble using *input*, I specified the following settings for *listings*:

Code Block 11: Listings

```
\lstset{language=TeX, %
basicstyle=\ttfamily\normalsize, frame=lines, %
showspaces=false, showstringspaces=false,%
tabsize=4, aboveskip=10pt, belowskip=10pt, %
lineskip=2pt, numbers=left, numberstyle=\tiny, stepnumber=1,
numbersep=5pt,%
numberblanklines=false, %
breaklines, breakatwhitespace, prebreak=, breakindent=0pt %
, backgroundcolor=\color{green!10} %
}
```

These settings, including defining the language as LaTeX, have defined the appearance of the coding blocks I have presented thus far. To enter the listing environment, use the command, *lstlisting*.

TEXercise 4: Code

Problem 4. *Generate a block of code. In this exercise, to prevent the software from getting confused by the different environments we are passing between, you will have to generate a separate* *.tex *file which contains the code and import that file as you did above.*

Hint. In a separate file, *'PROSOL/CodeSnip.tex'*, I wrote:

Code Block 12: Code

```
\begin{lstlisting}
\section{The Main Body}\label{mainbody}
\end{lstlisting}
```

Within your own documents, you may typically place the code directly into the document you are writing. □

TEXercise Solution 4: Code

Solution. YOUR SOLUTION ■

3.6 References

Often times while writing, you will want to refer to various pieces of your document. There are several ways to do this, depending on exactly what you are trying to reference, but they all follow one general format, with the exception of references to works cited.

Cross References

As you are writing, anything you wish to label, such as Section 3.6 here, should be tagged using the *label* command. Then when you want to refer back to that item, you need only call the appropriate label using the *ref* command. This works for the majority of items you might reference, as I demonstrate here in my references to Table 1 and Figure 3 (In general, it is best to label the captions on figures, rather than the figure itself). Equations are a slightly special case and should be referenced with the *eqref* command, and with Equation (2), although it is not generally necessary.

Code Block 13: Cross References

```
...Section \ref{references} here, should be tagged using the
\emph{label} command.  Then when you want to refer back to that
item, you need only call the appropriate label using the
\emph{ref} command...Equations are a slightly special case and
should be referenced with the \emph{eqref} command, as with
Equation \eqref{euler}.
```

Footnotes

Footnotes may be generated in LaTeX using the *footnote* command.[2]

Code Block 14: Footnotes

```
Footnotes may be generated in \LaTeX\ using the \emph{footnote}
command.\footnote
{
```

[2] Text placed within the braces will go into a footnote. Once in the environment, you can reference of items, display math, and generally use LaTeX as you would in any other environment.

```
Text placed within the braces will go into a footnote.  Once in
the environment, you can reference of items, display math, and
generally use \LaTeX\ as you would in any other environment.
}
```

Bibliography

To generate a bibliography of works cited, requires you to use a separate **.bib* file and the software, BibTeX, which is included with most LaTeX distributions. It also requires the *natbib* package. When referencing you bibliography, the program will skip over citations if it they cannot be found in your **.bib* file (that has also been appropriately compiled using BibTeX). If this happens, you will receive a question mark in lieu of the citation. Luckily, the **.bib* file is easy to assemble, since it is little more than a list of all your sources. The block of code below gives examples of how to record several types of sources material:

Code Block 15: *.bib File

```
@ARTICLE{Lewis14,
   AUTHOR  = {Lewis, Darl},
   TITLE   = {A \LaTeX Template},
   YEAR    = {2014},
   JOURNAL = {A Journal},
   VOLUME  = {1},
   NUMBER  = {1},
   PAGES   = {1-12}
}
%
@BOOK{Lewis14a,
   AUTHOR  = {Lewis, Darl},
   TITLE   = {A \LaTeX Book},
   YEAR    = {2014},
   PUBLISHER = {My Computer},
   ADDRESS  = {NJ},
}
%
@INCOLLECTION{Lewis14b,
   AUTHOR  = {Lewis, Darl},
```

```
21    TITLE   = {Another \LaTeX Template},
22    YEAR    = {2014},
23    EDITOR = {Lewis, Darl},
24    BOOKTITLE  = {Another \LaTeX Book},
25    PUBLISHER = {My Computer},
26    ADDRESS  = {NJ},
27 }
```

To insert the list of references at the end of your document, simply define the bibliography style and call the bibliography before you end the document. Of course, if you want to insert the references elsewhere, you may do that as well. As a default, I use the bibliography style associated with the *Journal of Political Economy*, but this option is quite flexible. The *nocite* option allow the software to include references in the works cited section whether or not they have been cited within the document.

Code Block 16: Bibliography

```
1 \bibliographystyle{jpe}
2 \nocite{*}
3 \bibliography{CiteFile}
```

When you want to refer back to the works you have cited, you may use one of several *cite* commands, the most useful of which are *cite* (which produces this:Lewis (2014c)) and *citep* (which produces (Lewis, 2014c)).

Code Block 17: Citations

```
1 ...the most useful of which are \emph{cite} (which produces
this:\cite{Lewis14}) and \emph{citep} (which produces
\citep{Lewis14}).
```

TEXercise 5: Refer

Problem 5. *I have labelled the following equation, "TestEq." Write a comment which refers to this equation, and insert a footnote citing Lewis (2014c) at the end of that comment.*

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

Hint.

Code Block 18: References

```
This comment refers to Section
\eqref{TestEq}.\footnote{\cite{Lewis14}}
```

□

TEXercise Solution 5: Refer

Solution. YOUR SOLUTION ■

4 Math

The value of LATEX becomes most apparent when writing equations. After a while, it becomes very easy to simply type equations as you go in a way that does not accrue with many other software platforms. Most mathematical symbols are simply a \ followed by the name of the symbol, and most of the common structures you are likely to encounter are designed to be implemented with relative ease and flexibility. If you want to write an inline equation, just use write the appropriate commands between two dollar signs, $, to produce something like this $a + b = c$. If you want to to write something longer or make it more visible by pulling it out of the text, though, use the *equation* environment:

$$\mathrm{e}^{i\pi} + 1 = 0 \tag{2}$$

Code Block 19: Numbered

```
\begin{equation}\label{euler}
\mathrm{e}^{i\pi}+1=0
```

```
\end{equation}
```

If you don't want to attach a number, there are several options. I usually just go with this one...

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right)=-\nabla p+\nabla\cdot\mathbf{T}+f$$

Code Block 20: No Number

```
\[
\rho\left(\frac{\partial \mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right)=-\nabla p+\nabla\cdot\mathbf{T}+f
\]
```

4.1 Complicated Equations and Numbering

Suppose you are trying to write a longer equation of a system of related equations. In such a scenario, you may prefer to use more than the basic features provided by the equation environment. First, I present a series of related equations, which I have given a special non-numeric label:

$$\begin{aligned} a+b&=c \\ 2(a+b)-(a+b)&=2c-c \\ 2a+2b-2c&=a+b-c \\ 2(a+b-c)&=a+b-c \\ 2&=1 \end{aligned} \tag{Bad Math}$$

Code Block 21: Split

```
\begin{equation}\tag{Bad Math}
\begin{split}
a+b&=c \\
```

```
2(a+b)-(a+b)&=2c-c \\
2a+2b-2c&=a+b-c \\
2(a+b-c)&=a+b-c \\
2&=1
\end{split}
\end{equation}
```

Note the use of the *split* environment and the use of the symbol, &, to align equations in this series of equations. Suppose I want each to have a unique number though. In that case, I can use the *align* command in lieu of *equation* and *split*:

$$a+b=c \tag{3}$$

$$2(a+b)-(a+b)=2c-c \tag{4}$$

$$2a+2b-2c=a+b-c \tag{5}$$

$$2(a+b-c)=a+b-c \tag{6}$$

$$2=1 \tag{7}$$

Code Block 22: Align

```
\begin{align}
a+b&=c \\
2(a+b)-(a+b)&=2c-c \\
2a+2b-2c&=a+b-c \\
2(a+b-c)&=a+b-c \\
2&=1
\end{align}
```

Alternatively, you can nest a few related equations under one principal number. To do so, use *subequations*:

$$a + b = c \tag{8a}$$

$$2(a + b) - (a + b) = 2c - c \tag{8b}$$

$$2a + 2b - 2c = a + b - c \tag{8c}$$

$$2(a + b - c) = a + b - c \tag{8d}$$

$$2 = 1 \tag{8e}$$

Code Block 23: Subequations

```
\begin{subequations}\label{mygrp}
\begin{align}
a+b&=c \label{firstgrp}\\
2(a+b)-(a+b)&=2c-c \\
2a+2b-2c&=a+b-c \\
2(a+b-c)&=a+b-c \\
2&=1
\end{align}
\end{subequations}
```

If you label them properly, you can easily refer back to previous equations like (8). That makes it easy to talk about how similar (8a) is to the first equation I presented here in Section 4.

As one last example, consider a long equation that does not fit on a single line. Here we use *multline*, as well as introduce the use of *array* for the vectors and *cases* for the piecewise component.

TEXercise 6: Solve

Problem 6. *Solve the equation,* $4x + 5y = 7x + 3y$ *for* y*. Show your work.*

Hint.

Code Block 24: Solve

```
\begin{subequations}
\begin{align}
4x+5y &=7x+3y \\
5y-3y &=7x-4x \\
2y &=3x \\
y &=\frac{3}{2}x
\end{align}
\end{subequations}
```

□

TEXercise Solution 6: Solve

Solution. YOUR SOLUTION ■

$$\begin{bmatrix} a \\ c \\ e \end{bmatrix} \cdot \begin{bmatrix} d & e & f \end{bmatrix} + \delta(x) \sum_{i=0}^{N} y_i + \mathcal{H}(x) \prod_{j \in \Omega} z_j + \int_{-\infty}^{\infty} \frac{1}{\pi(1+x^2)} dx$$

$$= \begin{cases} ad + be + cf + 1 & \text{if } x < 0 \\ ad + be + cf + \sum_{i=0}^{N} y_i + \frac{1}{2} \prod_{j \in \Omega} z_j + 1 & \text{if } x = 0 \\ ad + be + cf + \prod_{j \in \Omega} z_j + 1 & \text{if } x > 0 \end{cases} \tag{9}$$

Code Block 25: Multline

```
\begin{multline}
\left[\begin{array}{c}
```

```
a \\
c \\
e \\
\end{array}\right]
\cdot\left[\begin{array}{ccc}
d&e&f \\
\end{array}\right]
+\delta(x)\sum_{i=0}^{N}y_{i}
+\mathcal{H}(x)\prod_{j\in\Omega}z_{j}
+\int_{-\infty}^{\infty}\frac{1}{\pi(1+x^{2})}dx\\
=\begin{cases}
ad+be+cf+1 &\mbox{if } x<0\\
ad+be+cf+\sum_{i=0}^{N}y_{i}
+\frac{1}{2}\prod_{j\in\Omega}z_{j}+1 &\mbox{if } x=0\\
ad+be+cf+\prod_{j\in\Omega}z_{j}+1 &\mbox{if } x>0\\
\end{cases}
\end{multline}
```

There are still more ways of separating an equation into multiple lines with slightly varying effects. For more on mathematical features and symbols, I will refer you to a well-written document in Downes (2002), which offers a quick run-down of the capabilities of the software. For a comprehensive list of symbols available, both in the equation environment and the text environment, see Pakin (2009).

TEXercise 7: Multi

Problem 7. *Reconfigure this equation into three lines using* multline*:*

$$\begin{aligned} a+b+c+d+e+f+g+h+i+j &= \alpha+\beta+\gamma+\delta+\epsilon+\zeta+\eta+\theta+\iota+\kappa \\ &= 1+2+3+4+5+6+7+8+9+10 \end{aligned} \tag{10}$$

For convenience, here is the code I used to produce this format:

Code Block 26: Split

```
\begin{equation}
\begin{split}
a+b+c+d+e+f+g+h+i+j &=\alpha +\beta +\gamma +\delta
+\epsilon +\zeta +\eta +\theta +\iota+ \kappa \\
&=1+2+3+4+5+6+7+8+9+10
\end{split}
\end{equation}
```

Hint.

Code Block 27: Multiline

```
\begin{multline}
a+b+c+d+e+f+g+h+i+j \\
=\alpha +\beta +\gamma +\delta +\epsilon +\zeta +\eta
+\theta +\iota +\kappa \\
=1+2+3+4+5+6+7+8+9+10
\end{multline}
```

□

TEXercise Solution 7: Multi

Solution. YOUR SOLUTION ■

Table 1: The auto data

	Model 1		Model 2	
	(1)	(2)	(1)	(2)
Weight (lbs.)	1.747*	.008	4.614*	.000
(p-value)	(–.050)	(.567)	(.263)	(.020)
Foreign*Mileage			–.307	.006*
			(.240)	(.000)
Controls	N	Y	N	Y
Adj. R^2	.273	.435	.526	.773
No. of cases	74	74	74	74

Source: auto.dta

5 Tables

5.1 Regression Output

We often want to display data or reults in table form. Often times, the software you are using will provide resources to export data formatted for a table in a LaTeX document. *Matlab*, *Stata*, and *R* all have such packages. Sometimes you may need to prepare a table yourself, which is the goal of the *Table TEXercise*

5.2 Other Tabular Objects

The *equation* and *tabular* environments may be used for more than simply writing equations and summarizing data, however. Below, I use these two environments to visualize the prisoner's dilemma in Figure 1.

Code Block 29: Prisoner's Dilemma

```
\begin{figure}[h!]
\caption{Prisoner's Dilemma}\label{PD1}
\setlength{\arraycolsep}{0pt}
\[
\begin{array}{ccccc}
 & & \quad L \quad & & \quad R \quad & \\[.2ex]
```

TEXercise 8: Table

Problem 8. *Recreate Table (1), but include a third model. You may ignore the* table *environment (not the* tabular *environment!) and the caption. Choose whatever numbers you wish for the results.*

Hint.

Code Block 28: Table

```
\begin{center}
\begin{tabular}{l*{3}{rr}} \hline
&\multicolumn{2}{c}{Model 1} &\multicolumn{2}{c}{Model 2}
 &\multicolumn{2}{c}{Model 3}\\
& (1)&(2)&(1)&(2)&(1)&(2)\\ \hline\\[-1ex]
Weight (lbs.)& 1.74* & .08 & 4.14* & .00 & 4.64* & .00\\
($p$-value)& (.50) & (.56) & (.23) & (.00) & (.23) & (.00)\\
Foreign*Mileage & & & --.37 & .06* & --.37 & .06*\\
& & & (.20) & (.00) & (.24) & (.00)\\
Controls & N & Y & N & Y & N & Y\\ \hline
Adj. $R^2$  & .23 & .45 & .56 & .73 & .56 & .73\\
No. of cases & 74 & 74 & 74 & 74 & 74 & 74\\ \hline
\small\textit{Source:} auto.dta
\end{tabular}
\end{center}
```

□

TEXercise Solution 8: Table

Solution. YOUR SOLUTION ■

```
\cline{3-5} T \quad & \vline{} & \quad 1,1  \quad & \vline{} &
\quad -1,2 \quad & \vline{} \\
\cline{3-5} B \quad & \vline{} & \quad 2,-1 \quad & \vline{} &
\quad 0,0  \quad & \vline{} \\
\cline{3-5}
\end{array}
\]
%\end{figure}
```

Figure 1: Prisoner's Dilemma

	L	R
T	$1, 1$	$-1, 2$
B	$2, -1$	$0, 0$

And the same game in an alternate, fancier format:

Payoff Matrix

		Firm B: Defect	Firm B: Cooperate
Firm A	Defect	0 0	2 −1
Firm A	Cooperate	−1 2	~~1~~ ~~1~~

```
And the same game in an alternate, fancier format:

%\begin{table*}
\begin{center}
\begin{tabular}{ccccc}
&&\multicolumn{2}{c}{Payoff Matrix}\\
&&\multicolumn{2}{c}{Firm B}\\
&&Defect&Cooperate\\
\cline{3-4}
\raisebox{-0.25cm}{\rotatebox{90}{A}}
&\multicolumn{1}{p{1.6cm}}{Defect}&
\multicolumn{1}{|p{1.5cm}}{\hfill \hilight{$0$}\newline
\hilight{$0$}\hfill}&
\multicolumn{1}{|p{1.5cm}|}{\hfill $2$\newline $-1$\hfill}\\
\cline{3-4}
\raisebox{-0.25cm}{\rotatebox{90}{Firm}}
&\multicolumn{1}{p{1.6cm}}{Cooperate}&
\multicolumn{1}{|p{1.5cm}}{\hfill $-1$\newline $2$\hfill}&
\multicolumn{1}{|p{1.5cm}|}{\hfill \cancel{$1$}\newline
\cancel{$1$}\hfill}\\\cline{3-4}
\end{tabular}
\end{center}
%\end{table*}
\end{figure}
```

For the second version, I have also used two special commands, *cancel* and *hilight*. The first of these two can be implemented after loading the *cancel* package. The second is more involved, though, and requires us to write a new command. I placed this command in the preamble, so that I don't have to worry about where I try to implement it or where to find it if I wish to edit it, but you can place the command anywhere. Below, you will define a second, more advanced highlight command, *highlight*, which allows you to choose the color you use.

TEXercise 9: Highlighter

Problem 9. *Generate a command called* "highlighter" *and use it to highlight each color in the following sentence:*

I like red, green, and blue.

For reference, the following snippet generates the hilight *command I used above:*

Code Block 30: *hilight*

```
\newcommand{\hilight}[1]{\colorbox{yellow}{#1}}
```

Hint.

Code Block 31: *highlight*

```
\newcommand{\highlight}[2]{\colorbox{#1}{#2}}

I like \highlight{red!50}{red,}
\highlight{green!50}{green,} and \highlight{blue!50}{blue.}
```

□

TEXercise Solution 9: Highlighter

Solution. YOUR SOLUTION ■

In the final exercise for this section, you can write your own Tic-Tac-Toe game.

TEXercise 10: Game

Problem 10. *Generate a Tic-Tac-Toe game with X as the winner.*

Hint.

Code Block 32: Game

```
\[
\begin{array}{ccccc}
\quad X \quad & \vline{} & \quad O \quad & \vline{} & \quad
 \quad \\
\cline{1-5}
\quad O \quad & \vline{} & \quad X \quad & \vline{} & \quad
 \quad \\
\cline{1-5}
\quad  \quad & \vline{} & \quad  \quad & \vline{} & \quad X
\quad \\
\end{array}
\]
```

□

TEXercise Solution 10: Game

Solution. YOUR SOLUTION ■

6 Graphics

6.1 Importing Graphics

Here is an image with a caption included using the *graphicx* package. I have placed to copies of it inside a figure to make manipulation easier (i.e. placement within the paper, centering, &c), with the first image being imported as a raster image (**.png* or **.jpg*) and the second being an identical image in vector format (**.eps* or **.pdf*). Note that the latter file type produces a much smoother image, although this problem can be ameliorated using high-definition raster images.

Code Block 33: Outside Image

Figure 2: If you like nerdy comics, try *xkcd*. Also, go Caltech!

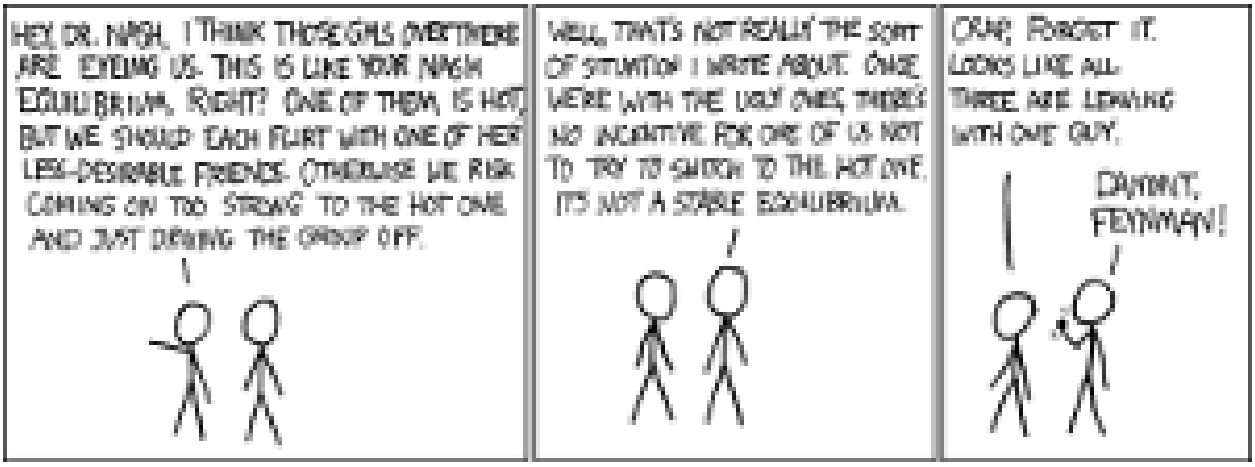
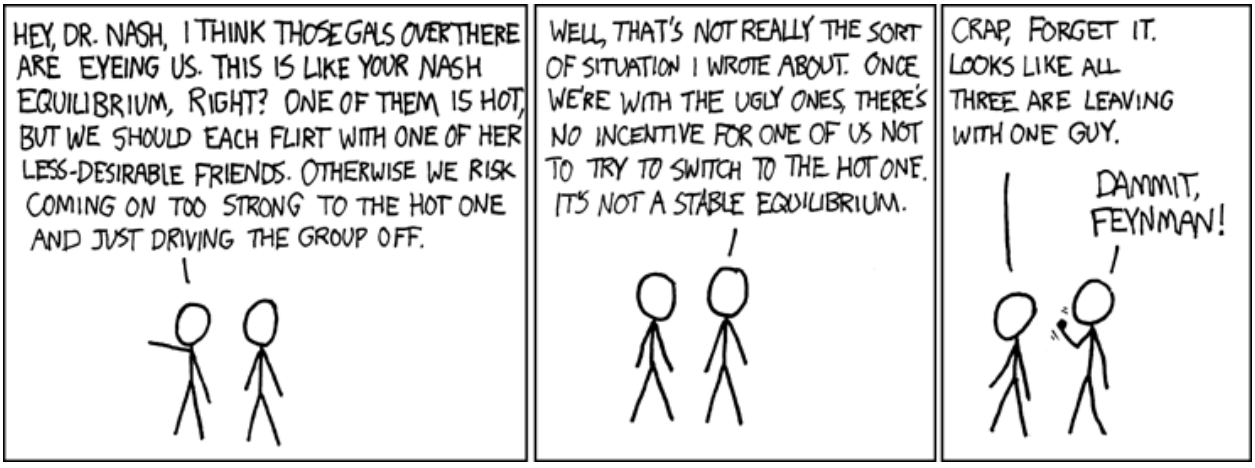


Figure 3: If you like nerdy comics, try *xkcd*. Also, go Caltech!




```
\begin{figure}[h!]
  \caption{If you like nerdy comics, try \emph{xkcd}.  Also, go
  Caltech!}
  \centering
  \includegraphics[width=\textwidth]{nash.png}

  \caption{If you like nerdy comics, try \emph{xkcd}.  Also, go
  Caltech!}\label{feynman}
  \centering
  \includegraphics[width=\textwidth]{nash.pdf}
\end{figure}
```

TEXercise 11: Image

Problem 11. *Import the image of your choice. I have provided* medusoid.pdf *if you are not inclined to find one of your own. Include a caption with your graphic.*

Hint.

Code Block 34: Image

```
\begin{center}
\includegraphics[width=.8\textwidth]{medusoid.pdf}
\captionof{figure}{Artificial Jellyfish}
\end{center}
```

**Note, typically the following code will work, but only when it is possible to place the image in a figure.

Code Block 35: Image II

```
\begin{figure}[h!]
\caption{figure}{Artificial Jellyfish}
\centering
\includegraphics[width=.8\textwidth]{medusoid.pdf}
\end{figure}
```

□

TEXercise Solution 11: Image

Solution. YOUR SOLUTION ■

6.2 Drawing in LaTeX

Of course, sometimes you don't have a preexisting picture and want to draw one yourself. Here are a few ways you can do that using the Ti*k*Z package. I start with some that will probably be more useful and move up to a couple of much more complicated diagrams. Of course there are many more options out there if you search for Ti*k*Z examples. Look at Crémer (2011) to start.

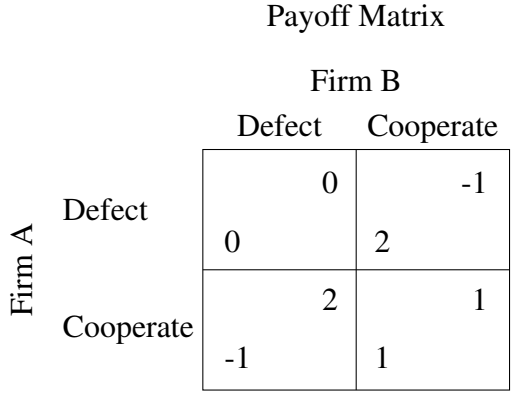
The Prisoner's Dilemma

As a basis for measuring the capabilities of Ti*k*Z, Figure 4 is a replication of the prisoner's dilemma as it was presented in the last example of the previous section. To complete this graphic, I have loaded three libraries in the preamble:

Code Block 36: Libraries

```
\usetikzlibrary{calc, matrix, positioning}
```

Figure 4: Prisoner's Dilemma



Code Block 37: Ti*k*Z Prisoner's Dilemma

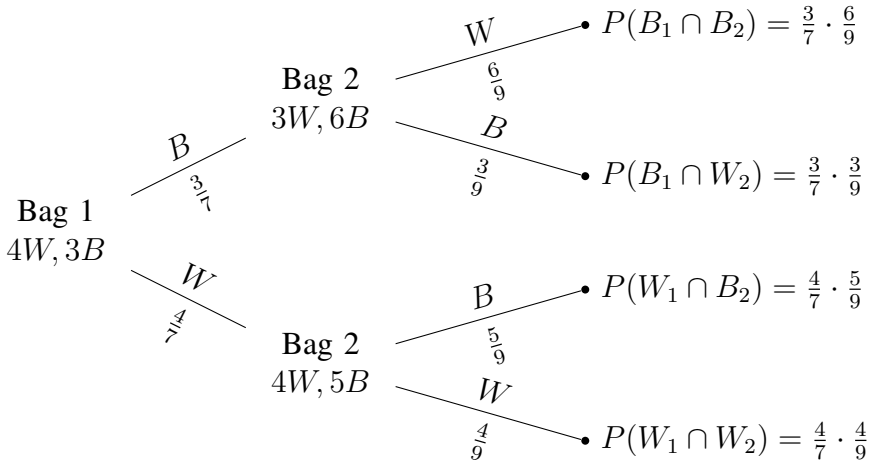
```
\begin{figure}[h!]
\caption{Prisoner's Dilemma}\label{PD3}
\centering
\begin{tikzpicture}
\matrix[matrix of math nodes,every odd
row/.style={align=right},every even
row/.style={align=left},every node/.style={text width=1.5cm},row
sep=0.2cm,column sep=0.2cm] (m) {
0&-1\\
0&2\\
2&1\\
-1&1\\
};
\draw (m.north east) rectangle (m.south west);
```

```
\draw (m.north) -- (m.south);
\draw (m.east) -- (m.west);

\coordinate (a) at ($(m.north west)!0.25!(m.north east)$);
\coordinate (b) at ($(m.north west)!0.75!(m.north east)$);
\node[above=5pt of a,anchor=base] {Defect};
\node[above=5pt of b,anchor=base] {Cooperate};

\coordinate (c) at ($(m.north west)!0.25!(m.south west)$);
\coordinate (d) at ($(m.north west)!0.75!(m.south west)$);
\node[left=20pt of c,align=center,text width=1cm]  {Defect};
\node[left=20pt of d,align=center,text width=1cm]  {Cooperate};

\node[above=18pt of m.north,align=center] (firm b) {Firm B};
\node[left=2.4cm of m.west,rotate=90,align=center,anchor=center]
{Firm A};

\node[above=5pt of firm b]  {Payoff Matrix};
\end{tikzpicture}
\end{figure}
```

Probability Tree

This Ti*k*Z graphic uses the *inputenc* package, as well as the *trees* library within Ti*k*Z.

Figure 5: Probability Tree



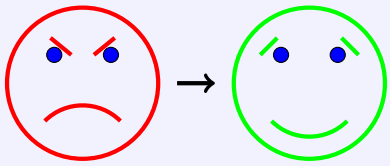
Code Block 38: Tree

```
% Set the overall layout of the tree
\tikzstyle{level 1}=[level distance=3.5cm, sibling distance=3.5cm]
\tikzstyle{level 2}=[level distance=3.5cm, sibling distance=2cm]

% Define styles for bags and leafs
\tikzstyle{bag} = [text width=4em, text centered]
\tikzstyle{end} = [circle, minimum width=3pt,fill, inner sep=0pt]

\begin{figure}[h!]
\caption{Probability Tree}
\centering
% The sloped option gives rotated edge labels. Personally
% I find sloped labels a bit difficult to read. Remove the sloped options
% to get horizontal labels.
\begin{tikzpicture}[grow=right, sloped]
\node[bag] {Bag 1 $4W, 3B$}
    child {
        node[bag] {Bag 2 $4W, 5B$}
            child {
                node[end, label=right:
                    {$P(W_1\cap W_2)=\frac{4}{7}\cdot\frac{4}{9}$}] {}
                edge from parent
                node[above] {$W$}
                node[below]  {$\frac{4}{9}$}
            }
            child {
                node[end, label=right:
                    {$P(W_1\cap B_2)=\frac{4}{7}\cdot\frac{5}{9}$}] {}
                edge from parent
                node[above] {$B$}
                node[below]  {$\frac{5}{9}$}
            }
            edge from parent
            node[above] {$W$}
            node[below]  {$\frac{4}{7}$}
    }
```

```
    child {
        node[bag] {Bag 2 $3W, 6B$}
        child {
                node[end, label=right:
                    {$P(B_1\cap
                    W_2)=\frac{3}{7}\cdot\frac{3}{9}$}] {}
                edge from parent
                node[above] {$B$}
                node[below]  {$\frac{3}{9}$}
            }
            child {
                node[end, label=right:
                    {$P(B_1\cap
                    B_2)=\frac{3}{7}\cdot\frac{6}{9}$}] {}
                edge from parent
                node[above] {$W$}
                node[below]  {$\frac{6}{9}$}
            }
        edge from parent
            node[above] {$B$}
            node[below]  {$\frac{3}{7}$}
    };
\end{tikzpicture}
\end{figure}
```

TEXercise 12: Smiley

Problem 12. *Turn the frown upside down! Transform the image on the left into the one on the right. I have included the code used to produce the frown.*



Code Block 39: Frown

```
\begin{center}
\begin{tikzpicture}[scale=0.5]
\draw [red, ultra thick] (0,0) circle [radius=2];
\draw [fill=blue] (.75,.75) circle [radius=.2];
\draw [fill=blue] (-.75,.75) circle [radius=.2];
\draw [red, ultra thick] [-] (.85,1.2) to  (.3,.75);
\draw [red, ultra thick] [-] (-.85,1.2) to  (-.3,.75);
\draw [red, ultra thick] [-] (-1,-1) to[out=45,in=135]
(1,-1);
\end{tikzpicture}
\end{center}
```

Hint. The following lines were changed to produce the smiley:

Code Block 40: Smiley

```
\draw [green, ultra thick] (0,0) circle [radius=2];
\draw [green, ultra thick] [-] (.85,1.2) to  (1.3,.75);
\draw [green, ultra thick] [-] (-.85,1.2) to  (-1.3,.75);
\draw [green, ultra thick] [-] (-1,-1) to[out=-45,in=-135]
(1,-1);
```

□

TEXercise Solution 12: Smiley

Solution. YOUR SOLUTION ■

7 Conclusion

Now that you have made it this far, it is time to just jump in and start playing around with the software. Practice makes perfect, right? If you have more requests for information to include, let me know. I will try to update this document over time to add functionality as well as make the learning experience easier. Hopefully there is enough here to at least give you an idea of what to look for as you attempt to write documents in LaTeX.

March 25, 2015

References

Crémer, Jacques. 2011. "A Very Minimal Introduction to Ti*k*Z." *Manuscript* 0 (0):1–24.

Downes, Michael. 2002. "A Short Math Guide for LaTeX." *AMS Manuscript* :1–17.

Lewis, Darl. 2014a. "Another LaTeX Template." In *Another LaTeX Book*, edited by Darl Lewis. NJ: My Computer.

———. 2014b. *A LaTeX Book*. NJ: My Computer.

———. 2014c. "A LaTeX Template." *A Journal* 1 (1):1–12.

Pakin, Scott. 2009. "The Comprehensive LaTeX Symbol List." *Manuscript* :1–164.