So You Want To Learn LaTeX?
- OR -
How to Escape the Pain that is Word.[1]

Leaning LaTeX(pronounced "La - tech" – not like the plastic) is a very good use of your time in graduate school for several reasons:

- It is free (i.e., within a grad student budget).
- It is available on any system you are likely to ever use (Windows, Linux, OS X).
- It is highly customizeable and adaptable.
- It handles mathematics really well.
- It produces cool looking documents.

There are many excellent books available that you should buy. I personally own Leslie Lamport *LaTeX: A Document Preparation System* and Helmut Kopka and Patrick W. Daly *A Guide to LaTeX: Document Preparation for Beginners and Advanced Users*. There are also scores of online documents that you should avail yourself of and Google can usually point you to the answer of any question you might have. It will take some time and frustration to learn, but it will pay off the first time you need to insert an equation into one of your papers.

Today we are going to look briefly at:

- Installation.
- Basic document formatting.
- Basic text commands.
- Basic math commands.
- Basic image manipulation.
- Creating and using bibliographies.
- Compiling and viewing documents.

1 Installation

I am assuming that you are using a Windows machine (on my iMac running OS X I have installed and used TEXshop).[2] I use WinEdt (http://www.winedt.com) as a text editor and the MikTeX (http://www.miktex.org/) LaTeXdistribution. These two programs integrate pretty seemlessly as our experiences today will (hopefully) demonstrate. WinEdt has the added benefit of having an "R-mode" that integrates with R. Other programs have the same functionality and I make no claims about relative superiority; I learned LaTeXusing WinEdt so that is what I use (hopefully my time is better spent researching than learning a new LaTeXenvironment).[3]

Following the installation instructions for WinEdt and MikTeX should get you up and running in no time. One aspect that deserves some mention is the loading of new packages. The functionality of LaTeXcan be expanded by adding packages. For example, if you wish to be able to change from single to double spacing easily you might want to add the package *setspace*.[4]

Use the MikTeX package manager to load new packages. Go to Start $->$ Programs $->$ MikTeX $->$ MikTeX Package Manager. Select "Database" and "Site Wizard..." In the new dialog box select Internet and choose your favorite site. To install a new package, simply scroll through the list, find your package and left click. Select "install" and watch the program install to make sure everything works – for a few of

[1] Prepared by Joshua D. Clinton, Department of Politics, Princeton University

[2] TEX shop is available at http://www.uoregon.edu/ koch/texshop/texshop.html.

[3] Perhaps the most popular, especially among the Linux crowd is a text-editor called EMACS that also has a built-in interface for LaTeX.

[4] How do you find about the name of packages? Just Google "double spacing" "latex" "packages."

the mirrors I have sometimes received errors about being unable to find the file. Just try another mirror when this happens. When the dialog box closes the package is installed and you can make use of it in your documents.

If it happens that a package you need is not available from the Package Manger, you can always install the package manually. First, go to the Comprehensive TeX Archive Network (www.ctan.org) and search for the file or just Google the file directly. Once you find it. Second, create a new folder named the same as the package you are installing. In this case we are going to create a folder named setspace in the following directory "/ texmf / tex / latex" and copy all files into this directory. In this case we copy setspace.sty into the setspace folder. Now we need to tell MikTeX to update the list of packages it recognizes. Go to Start $->$ Programs $->$ MikTeX $->$ MikTeX Options. Click on the "Refresh Now" button in the dialog box that appears for "File Name database". Click on this button a couple of times. Now try to compile your document (see section 7). If it crashes and says that the package cannot be found, navigate through the Start menu again and "Refresh now" again. In my experience sometimes it takes a couple of tries before it recognizes the addition.

Now you should be good to go.

2 Basic Document Formatting

All documents begin with a header that tells LaTeXhow to format the content you provide. If we are interested in writing a document, the following will create a new document for editing. Save the document as mydoc.tex (or anything you wish).

```
\documentclass{article}
\begin{document}
\doublespacing
This is my first document.
\end{document}
```

If we are interested in writing books, letters, memoirs (or maybe even a thesis[5], we can change **documentclass** to reflect this. There are also options that we can use to apply to the formatted document. The default font size is 10 point. We can also choose 11 point [**11pt**] or 12 point [**12pt**]. We can choose to have the documented formatted in two columns [**twocolumn**] or formatted for A4 paper [**a4paper**]. For example, to create a new document in 12 point font, two columns and on A4 paper you would type:

```
\documentclass[12pt,twocolumn,a4paper]{article}
\begin{document}
This is my first document.
\end{document}
```

The header is also where you can tell the program to load other elements that you might use. For example, if you want to use double spacing in the paper, you need to tell the program this in the header by telling it to use the *setspace* package. So long as you have the package installed LaTeXwill recognize it when you compile the document. If you do not have it installed, the program will choke and let you know. You can install several packages just as you can install several **documentclass** options by separating them by a comma.

```
\documentclass{article}
\usepackage{setspace}
\begin{document}
\doublespacing
 This is my first document.
\end{document}
```

Although not technically a header, prior to getting started you might want to include such things as a title page, an abstract, acknowledgements and contact information. The following code does this for you.

[5]Search for and save puthesis.sty for future use.

```
\documentclass{article}
\begin{document}
\begin{titlepage}
\thispagestyle{empty}
\title{Our Magnum Opus\thanks{We thank you.}}
\author
{Student 1 \\ Department of Politics \\ Princeton University \\
Princeton, NJ 08544-1012 \\student1@princeton.edu

\and % Delete if solo-authored

Student 2 \\ Department of Politics \\ Princeton University \\
Princeton, NJ 08544-1012 \\student2@princeton.edu}

\maketitle \thispagestyle{empty}
\end{titlepage}

\pagebreak  %   Delete if you want abstract on title page.

\thispagestyle{empty}
\begin{abstract}
This is the greatest work of all time.  I prove everything true
everything that is true.\end{abstract}

\pagebreak % Delete if you want abstract on text page.

A=A.
\end{document}
```

3 Basic Text Commands

You can **bold**, *italicize* "quote correctly," make the text tiny, very small, smaller, small, normal size large larger even larger, still larger and largest. Once the text is changed it will remain so until it is readjusted.

And the code is:

```
You can \textbf{bold}, \textit{italicize} ``quote correctly," make
the text \tiny tiny, \scriptsize very small, \footnotesize
smaller, \small small, \normalsize normal size \large large \Large
larger \LARGE even larger, \huge still larger and \Huge largest.
\normalsize Once the text is changed \tiny it will remain so until
it is \Large readjusted.
```

You can also

or make the text left justified.

The code is:

```
You can also
\begin{center}
center the text on the page \\
```

```
\end{center}
\begin{flushright} Make the text right justified \\ \end{flushright}
\begin{flushleft} or make the text left justified. \end{flushleft}
```

Double right slashes

```
\\
```

inserts a line break, while inserting the command

```
\pagebreak
```

forces a page break. The environment

```
verbatim creates typewriter text.
```

Horizontal and vertical spaces can be creating using **vpsace** and **hspace** commands. For example

```
\vspace{2 in}
```

creates a two inch vertical space between text.

> There are also two commands that allow you to indent from both sides. These are useful when enclosing large quotations in the text.
>
> They differ by whether or not the first line is indented. The quote environment separates the quote-ed text with an empty line but does not indent as in the paragraph above. This paragraph, done using the quotation environment indents the first line.

If you have a list of items you can use one of three environments:

There is the *itemize* environment ...

- One fish.
- Two fish.
- Red fish.
- Blue fish.

There is the *enumerate* environment ...

1. One fish.
2. Two fish.
3. Red fish.
4. Blue fish.

There is the *description* environment ...

One fish.

Two fish.

Red fish.

Blue fish.

And the code is:

```
\begin{center}
There is the \textit{itemize} environment \dots
\end{center}
\begin{itemize}
\item One fish. \item Two fish. \item Red fish. \item Blue fish.
\end{itemize}

\begin{center}
There is the \textit{enumerate} environment \dots
\end{center}
\begin{enumerate}
\item One fish. \item Two fish. \item Red fish. \item Blue fish.
\end{enumerate}

\begin{center}
There is the \textit{description} environment \dots
\end{center}
\begin{description}
\item[One] fish. \item[Two] fish. \item[Red] fish. \item[Blue]
fish.
\end{description}
```

Tables are easy to create using the tabular environment. You can control whether columns are right, center or left justified, whether (and how many) horizontal lines separate the columns, whether (and how many) vertical lines separate the rows. Note that just using the tabular environment uses the default justification for the document unless you define otherwise.

Variable	**Coefficient**	**Stnd. Err.**
Constant	4.56	.23
Party	2.36	.54
R^2	.59	

The code for the above is:

```
\begin{tabular}{r||c|c}
  \hline \hline
  \textbf{Variable} & \textbf{Coefficient} & \textbf{Stnd. Err.} \\
  \hline
  Constant & 4.56 & .23 \\
  Party & 2.36 & .54 \\
  \hline
R$^2$ & .59 &  \\
\hline \hline
\end{tabular}
```

Note that you can (and should) imbed a tabular environment within a table environment. This gives you a couple of additional parameters to adjust. You can now easily center justify the entire table, add a caption and add a label to refer to the Table without hardwiring table numbers. This is useful when editing and you are not sure what the Table number will be but you still want to refer to it in the text. For example:

Table 1 reports the results:

Variable	Coefficient	Stnd. Err.
Constant	4.56	.23
Party	2.36	.54
R^2	.59	

Table 1: The Main Result: This table reports the regression coefficients for equation 5 in the text.

```
Table \ref{MainResult} reports the results:
\begin{table}[h*]
\centering
\begin{tabular}{r||c|c}
  \hline \hline
  \textbf{Variable} & \textbf{Coefficient} & \textbf{Stnd. Err.} \\
  \hline
  Constant & 4.56 & .23 \\
  Party & 2.36 & .54 \\
  \hline
R$^2$ & .59 &  \\
\hline \hline
\end{tabular}
\caption{The Main Result: This table reports the regression
coefficients for equation 5 in the text.}\label{MainResult}
\end{table}
```

Footnoting is easy, just include the command **footnote** wherever you want a footnote.[6]

```
Footnoting is easy, just include the command \textbf{footnote}
wherever you want a footnote.\footnote{They can be places anywhere
-- not just at the end of sentences.  You can use them in tables
as well.}
```

A final wrinkle is that journals now appear to like endnotes rather than footnotes. To do this requires several steps. First, load the package *endnotes*. Second, in the header of the document include the following:

```
 \let\footnote=\endnote
```

. This will replace the **footnote** command (but not the word footnote) wherever it appears in the text when you compile the document. In other words, when writing, continue to use the **footnote** command. Finally, wherever you want the page of endnotes to begin (typically after the bibliography) include the following:

```
\newpage
\begingroup
\parindent 0pt
\parskip 2ex
\def\enotesize{\normalsize}
\theendnotes
\endgroup
```

4 Basic Math Commands

The beauty of LaTeXis its ability to integrate text and math. To insert math in a sentence, simply use a dollar sign to denote the beginning and ending of the math.

This bit of math-text

[6]They can be placed anywhere – not just at the end of sentences. You can use them in tables as well.

Given this structure, the Bernoulli probability of y_{ti} is: $Pr(y_{ti} = 1|z_t, \alpha_i, \beta_i) = F(\beta_i z_t - \alpha_i)^{y_{ti}}[1 - F(\beta_i z_t - \alpha_i)]^{1-y_{ti}}$. Assuming that the ratings are independent conditional on the true latent quality of legislation z_t and the rater parameters α_i and β_i implies that $Pr(y_{t1}, ..., y_{tN}|z_t, \alpha_i, \beta_i) = Pr(y_{i1}|z_t, \boldsymbol{\alpha}, \boldsymbol{\beta}) \times ... \times Pr(y_{iN}|z_t, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \Pi_{j=1}^{N} F(\beta_i z_t - \alpha_i)^{y_{ti}}[1 - F(\beta_i z_t - \alpha_i)]^{1-y_{ti}}$.

is produced using

```
Given this structure, the Bernoulli probability of $y_{ti}$ is:
$Pr(y_{ti}=1 | z_t , \alpha_i , \beta_i ) = F(\beta_i z_t -
\alpha_i)^{y_{ti}} [1- F(\beta_i z_t - \alpha_i)]^{1 - y_{ti}}$.
Assuming that the ratings are independent conditional on the true
latent quality of legislation $z_t$ and the rater parameters
$\alpha_i$ and $\beta_i$ implies that $Pr(y_{t1},...,y_{tN} | z_t
, \alpha_i , \beta_i) =$ $Pr(y_{i1}  | z_t , \bm{\alpha}  ,
\bm{\beta} ) \times ... \times Pr(y_{iN}  | z_t ,  \bm{\alpha}  ,
\bm{\beta})$ $= \Pi_{j=1}^{N} F(\beta_i z_t - \alpha_i)^{y_{ti}}
[1- F(\beta_i z_t - \alpha_i)]^{1 - y_{ti}}$.
```

From this it is trivial to see how to subscript, superscript, use lowercase and uppercase Greek, and use mathematical notation. There are too many commands to name – you'll have to look them up (although the WinEdt GUI has the ability to insert the correct command if you click on the icon).

If you have an entire line of math, you can set it apart from the text like this:

$$L(\boldsymbol{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \Pi_{t=1}^{T}\Pi_{j=1}^{N} F(\beta_i z_t - \alpha_i)^{y_{ti}}[1 - F(\beta_i z_t - \alpha_i)]^{1-y_{ti}}$$

using this code:

```
\[
L(\bm{z},\bm{\alpha},\bm{\beta}) = \Pi_{t=1}^{T} \Pi_{j=1}^{N}
F(\beta_i z_t - \alpha_i)^{y_{ti}} [1- F(\beta_i z_t -
\alpha_i)]^{1 - y_{ti}}
\]
```

Alternatively, you can use the equation environment to produce labelled equations. Equation A1 denotes something:

$$\sum_{t=1}^{T} D_{x_i}\left(\frac{D_{x_i}(\rho(x_i^*, y_t, q_t))}{\rho(x_i^*, y_t, q_t)}\right) = \sum_{t=1}^{T} D_{x_i}\left(\frac{D_{x_i}(\rho(x_i^*, y_t, y_{m(t)}))}{\rho(x_i^*, y_t, y_{m(t)})}\right) \tag{A1}$$

The code to produce this is:

```
Equation \ref{EqtnA1} denotes something:
\begin{equation}
\sum_{t=1}^{T}D_{x_{i}}\left(
\frac{D_{x_{i}}(\rho(x_{i}^{*},y_{t},q_{t}))}{\rho(x_{i}^{*},y_{t},q_{t})}\right)
=\sum_{t=1}^{T}D_{x_{i} }\left(
\frac{D_{x_{i}}(\rho(x_{i}^{*},y_{t},y_{m(t)}))}{\rho(x_{i}^{*},y_{t},y_{m(t)})}\right)
\tag{A1} \label{EqtnA1}
\end{equation}
```

Note how subscripts and superscripts are used to index the summation, how one can use the **frac** command to specify the numerator and denominator of a fraction respectively (although journal editors hate this because it eats up a lot of space), and how the left and right commands are used in conjuction with parenthesis to create sufficiently large parentheses to indicate clearly the elements it contains. Loading the packages *amsmath*, *amsfonts*, and *amssymb* gives you the ability to relabel the equation using the **tag** command. The default is a simple numeric ordering, but there is no reason why you need to restrict yourself to that. The other benefit is that you can use the **label** command to reference the equation in the text.

If you have a multi-line equation you can use the **array** command.

$$\begin{array}{rl} Pr(y_{ti}=1) = & Pr(x_{ti} > \gamma_i) \\ = & Pr(z_t + \epsilon_{ti} > \gamma_i) \\ = & Pr(\epsilon_{ti} > \gamma_i - z_t) \\ = & 1 - F((\gamma_i - z_t)/\sigma_i) \\ = & F((z_t - \gamma_i)/\sigma_i) \end{array}$$

The code to do this is:

```
\[
\begin{array}{rl}
Pr(y_{ti}=1) = &  Pr(x_{ti} > \gamma_{i}) \\
             = & Pr(z_{t} + \epsilon_{ti} > \gamma_{i}) \\
             = & Pr (\epsilon_{ti} > \gamma_i - z_t) \\
             = & 1 - F( (\gamma_i - z_t)/ \sigma_i) \\
             = & F( (z_t - \gamma_i)/ \sigma_i) \\
\end{array}
\]
```

There is also an **eqnarray** command that you can use to accomplish almost the same thing. Using **eqnarray*** instead of **eqnarray** drops the line-by-line equation numbers. Note that is is not possible to justify using **eqnarray** – it is default center justified.

$$\begin{eqnarray} Pr(y_{ti}=1) = & Pr(x_{ti} > \gamma_i) & (1) \\ = & Pr(z_t + \epsilon_{ti} > \gamma_i) & (2) \\ = & Pr(\epsilon_{ti} > \gamma_i - z_t) & (3) \\ = & 1 - F((\gamma_i - z_t)/\sigma_i) & (4) \\ = & F((z_t - \gamma_i)/\sigma_i) & (5) \\ & & (6) \end{eqnarray}$$

The code to do this is:

```
\begin{eqnarray}
Pr(y_{ti}=1) = &  Pr(x_{ti} > \gamma_{i}) \\
             = & Pr(z_{t} + \epsilon_{ti} > \gamma_{i}) \\
             = & Pr (\epsilon_{ti} > \gamma_i - z_t) \\
             = & 1 - F( (\gamma_i - z_t)/ \sigma_i) \\
             = & F( (z_t - \gamma_i)/ \sigma_i) \\
\end{eqnarray}
```

This is only the tip of a very large iceberg. You can have equation arrays with any number of columns, you can include math in the tabular environment and you can create equations sufficiently complicated so that no one can figure out what it means. You now have at your disposal more symbols that you (hopefully) know what to do with (although Adam claims to have once had to use Hebrew because he "used up" all of the Greek letters). The one caveat is that because it is not WYSIWYG you will undoubtedly have errors in your math the first time you compile. Therefore compile after every equation so you can identify where the error is likely to be. Also, read the math carefully as it is easy to get lost in the LaTeXnotation as the complexity of the code for some of the above examples demonstrate.

5 Basic Image Manipulation

You can also include graphics in your LaTeXdocument. Be forewarned that this is some of the most frustrating parts of LaTeXand I have spent many a night wrestling with formatting on the verge of introducing my computer to the beauty of gravitational attraction.

The first requirement is that you need to load the package *graphicx* in the header. This gives the LaTeXcompiler the needed capabilities to handle images.

Note that the **includegraphics** command can be modified in several ways. First, one can adjust the **height** and **width** of the image in LaTeX(e.g., includegraphics[height=7in,width=6in]). This is not recommended – it is better to create the original image in the desired size. Second, one can rotate the image using the **angle** option (e.g., includegraphics[angle=90]). Note that the picture being used must be in the same directory as the *.tex file – or else directory information needs to be supplied. Also, always use encapsulated postscript (or regular postscript). PDF can be used, but then a different command needs to be used.

The final bit of hassel is the location of where to place the figure. The figure environment has several options:

h here – insert exactly where the code is.

b At the bottom of the page if space is available, if not it gets pushed further back.

t At the top of the page if space is available, if not it gets pushed further back.

p At the end of the document on a separate page.

Any of the above in combination with ! (e.g., h!) forces LaTeXto suspend formatting restrictions and put the figure there. Note that these same location parameters control the location of the table environment as well.

When I include figures, I usually include something like the following code:

```
\begin{center}
[Insert Figure \ref{PP} About Here]
\end{center}

\begin{figure}[p]
\includegraphics{PP.ps}
\caption{\textbf{Joint Distributions of Representative Roll Call
and Measures of (Sub) Constituency Preferences:} The figures plot
plots the joint distribution of the representative ideal point for
the 1st session of the 106th Congress and, proceeding clockwise
from the upper-left, the average ideology of the district,
in-party partition, out-party partition and independent partition
using all constituents (including unlikely voters). Solid (open)
points represent districts with Republican (Democratic)
representatives in the 106th Congress.\label{PP}}
\end{figure}
```

which produces the following...

[Insert Figure 1 About Here]

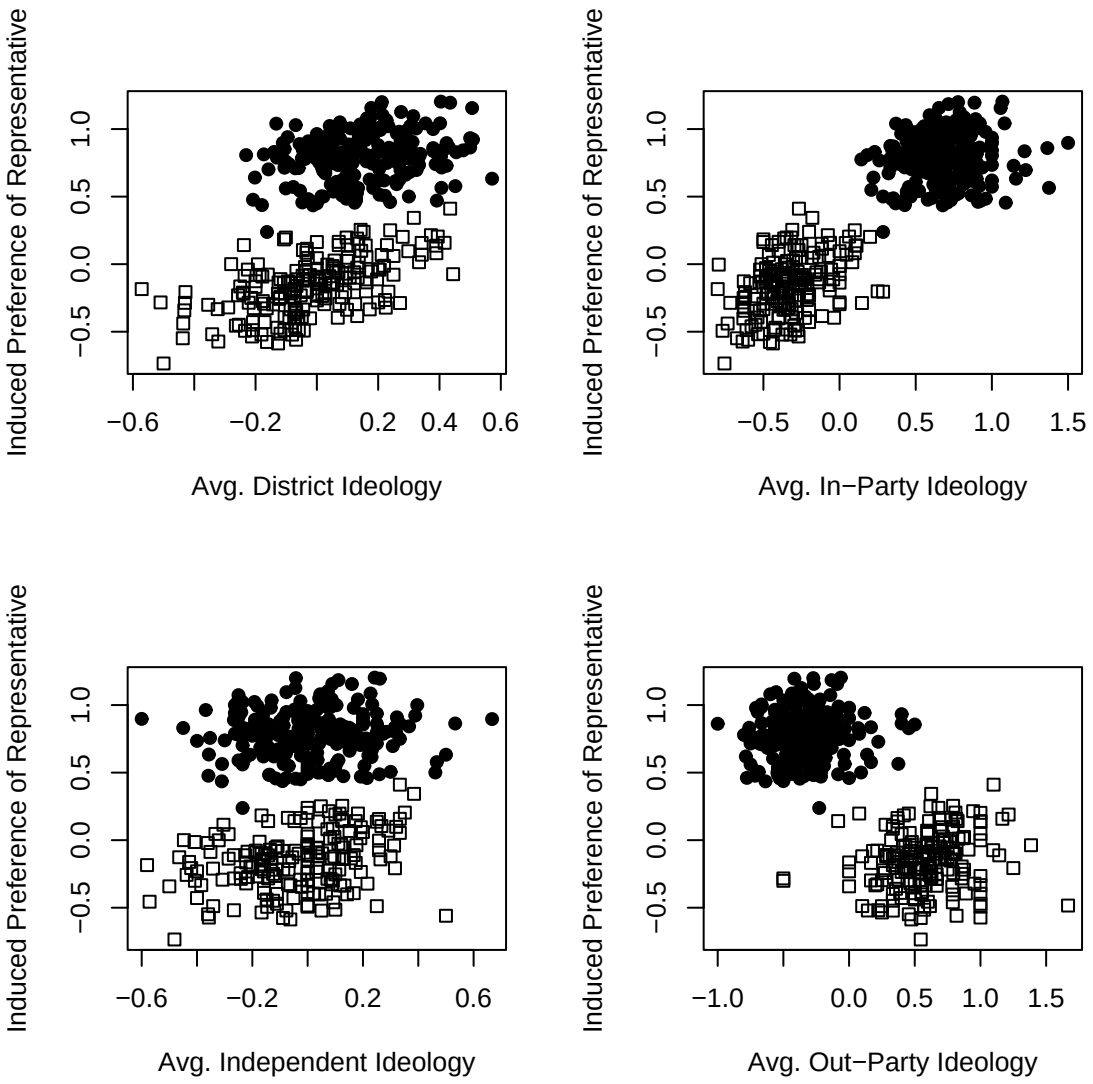


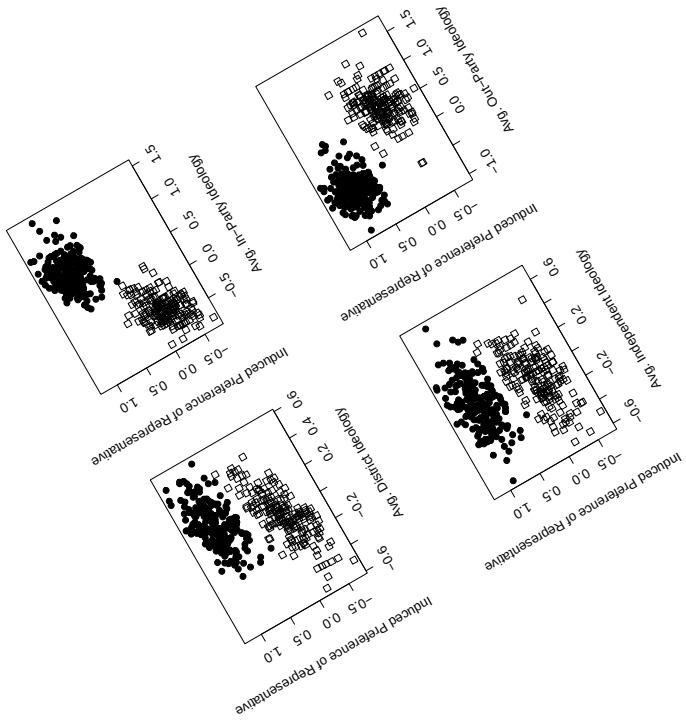
Figure 1: **Joint Distributions of Representative Roll Call and Measures of (Sub) Constituency Preferences:** The figures plot plots the joint distribution of the representative ideal point for the 1st session of the 106th Congress and, proceeding clockwise from the upper-left, the average ideology of the district, in-party partition, out-party partition and independent partition using all constituents (including unlikely voters). Solid (open) points represent districts with Republican (Democratic) representatives in the 106th Congress.

Too see the effects of the optional parameters for the figure, consider the following. As a personal note, I never seem to be able to get the figures to locate where I want them to without using the ! toggle (and sometimes a **pagebreak** or two).[7]

```
\begin{figure}[b!]
\includegraphics[height=3in,width=3in,angle=120]{PP.ps}
\end{figure}
```

Although the code to produce the figure appears before the verbatim environment above, not that the location parameter forces the image to appear at the bottom of the screen. Again, LaTeXis not WYSIWYG – the order in which tables, figures and text appears in the *.tex document is not necessarily the order it will appear when the document is compiled. Make sure you check to make sure that the outcome is desirable.

[7]Learning LaTeXis a ongoing process.



6 Creating and Using Bibliographies

Another desirable feature of LaTeXis the ability to use bibliography files and insert cites "automatically." This means that instead of having to type in the references at the end of each document you can have one document with all of the cites you would ever use and just reference that it the text. To use the bibliography feature requires using BibTeX, which is thankfully fully integrated into MikTeX and WinEdt for your convenience.

To use BibTeX, **usepackage** *natbib* and include the following in the header (I put it right after **usepackages**).

```
\bibliographystyle{apsr}
```

This code tells the compiler to format references according to the *American Political Science Review*. If you do a Google search you can find others, but this is probably the most useful for you. Each handles references and citations differently, so you can play around with the options if you like. **pa** is another useful style file – being the bibliography of *Political Analysis*.

To use BibTeX you need to create a new document my.bib (or anything *.bib). This file will contain all of the references that you will use in the paper (or ever). The file consists only of BibTeX entries – no headers or other text/code is required. There are several BibTeX "autoformats" that you can use: article, book, unpublished, chapter in book, Ph.D. thesis to name a few. It is easiest to let WinEdt create an entry for you. Go to Insert − > BibTeX Items and then select your reference type.

Selecting the Article entry produces:

```
@ARTICLE{*,
  AUTHOR =       "*",
  TITLE =        "*",
  JOURNAL =      "*",
  YEAR =         "*",
  volume =       "*",
  number =       "*",
  pages =        "*",
  month =        "*",
  note =         "*",
  abstract =     "*",
  keywords =     "*",
  source =       "*",
  file = F
}
```

which you can then fill out as much (or as little as needed. Note that the default character WinEdt uses as a space-holder will cause LaTeXto choke if you compile it. You do not need to fill out every entry – anything that you delete will not appear in the compiled document. Even if you fill out every line, every line will not appear in your bibliography page. What items are displayed depends upon the bibliography style that you use. Consequently, the **apsr** style uses publication month, the **pa** style does not.

Suppose we create a new file in the same location as the *.tex file called my.bib with the following two entries

```
@ARTICLE{McCarty:2001,
  AUTHOR =       "McCarty, Nolan, Keith Poole and Howard Rosenthal",
  TITLE =        "The Hunt for Party Discipline in Congress",
  JOURNAL =      "American Political Science Review",
  YEAR =         "2001",
  volume =       "95",
  number =       "3",
  pages =        "673-687",
  month =        "September",
 }
@BOOK{Arnold:1990,
```

```
  AUTHOR =       "Arnold, R. Douglas",
  editor =       "1990",
  TITLE =        "The Logic of Congressional Action",
  PUBLISHER =    "Yale University Press",
  YEAR =         "1990",
  address =      "New Haven",
}
```

We can now reference the items in a couple of ways. We can cite paranthetically using (Arnold, 1990) or else we can use a citation if we are talking about the work of Arnold (1990) directly.

```
We can now reference the items in a couple of ways.  We can cite
paranthetically using \citep{Arnold:1990} or else we can use a
citation if we are talking about the work of \citet{Arnold:1990}
directly.
```

Note that I have found citing multiple authors to be a bit buggy. For example, citing the McPR piece in a similar manner yields – McCarty and Rosenthal (2001). For these cases I usually just physically type the reference in McCarty, Poole and Rosenthal (2001) and then take advantage of the **nocite** command. **nocite** will list a reference in the bibliography even if it is not explicitly cited in the text.

The final bit of information is a location as to where the bibliography is to be located. This is done by simply including the following. Note that the bibliography will appear wherever in the document that this command is invoked – it does not automatically appear at the end. You may want to use a **pagebreak** to separate it from the text. If you need to doublespace, use the **doublespacing** command from the *setspace* package.

```
\bibliography{my}
```

References

Arnold, R. Douglas. 1990. *The Logic of Congressional Action*. New Haven: Yale University Press.

McCarty, Nolan, Keith Poole and Howard Rosenthal. 2001. "The Hunt for Party Discipline in Congress." *American Political Science Review* 95(3):673–687.

7 Compiling and Viewing Documents

Compiling and viewing using WinEdt and MikTeX is easy. Simply click on the LaTeXicon in WinEdt. This will bring up a command prompt window that will compile the code automatically. If everything is fine you will see text fly by and the window close. If it sees a mistake, it will stop and wait for you to do something. At this point you should try to fix the mistake if you know where it is, close the window and try to compile again (you can only have one compiling window open at a time). When in doubt, just press the enter key – which "ignores" the mistake. You will be able to see the problem soon enough. If you are also using BibTeX, after compiling LaTeXyou need to compile BibTeX by clicking on the Bib icon. Note that things are a bit buggy so you will need to click on each a couple times – I usually click on LaTeXonce and then on the Bib icon 5 times or so, see if the references are recognized and repeat as needed.

After having compiled using both LaTeXand BibTeX you are now ready to view your wondrous creation. The default format created by LaTeXis a *.dvi (device independent) file. This file is created and saved automatically by compiling LaTeX.

To view the document simply click on the DVI icon. You can also print if you wish. You can also save the document as a PDF or a Postscript file by clicking on the dvi $->$ PDF and dvi $->$ ps icons respectively. If you have Ghostview (and GSview installed), running dvi $->$ ps will highlight the owl icon. Clicking on the owl opens Ghostscript and views the *.ps file. If Acrobat or Acrobat Reader is installed, running dvi $->$ PDF will highlight the Adobe icon. Clicking on the icon will open the *.pdf file in Acrobat or Acrobat Reader.

The other icons: TeX, PDFLaTeX, PDFTeX should be ignored. As you get proficient, you make want to learn PDFLaTeX (which is basically the same as LaTeX except it uses different graphic commands to use PDF images instead of PS (and EPS) images and produces PDF files directly) – but none of this is necessary.

There are a huge number of addition items you can make, including overhead slides, powerpoint-ish graphical presentations, c.v.'s and many others. Buy some books, search the internet and play around with code. The best way to learn is to experiment. Find existing code (such as the code that produced this document or code from a paper someone has written) and "hack" that. Frustration is inevitable, but once you start "getting it," you won't want to go back. Welcome to the club.