Introduction to LaTeX

Jamie Monogan[*]

- What is LaTeX?
- Advantages: sharper presentation, increased publication rate, style files, equations, tables, bibliography
- Disadvantages: learning curve, less point and click, changing defaults can be hard
- Not discussed: posters and slideshows in Beamer

1 Course Objectives

By the end of this course, participants will be able to:

- Use web resources and documentation for future LaTeX help.
- Add basic programming code to compile text and include special characters in text.
- Create an article-style document that includes tables, figures, and a bibliography.
- Write mathematic equations in TeX.

2 Web Resources

`http://j.mp/learnlatex` (Monogan's LaTeX page)
`http://www.tug.org` (TeX user group)
`http://www.miktex.org` (LaTeX for Windows)
`http://www.tug.org/mactex/` (LaTeX for Mac)

[*]These notes are a revision and extension of a short course first taught by Evan Parker-Stephen at UNC on September 21, 2006. My thanks to Evan for sharing. This file was last revised on August 20, 2015.

3 Getting Started

By way of background, TeX is a typesetting program created by Donald Knuth. TeX is raw and hard to use, because it was never intended to structure or format a document. It does only low-level type-setting. LaTeX is Leslie Lamport's set of meta-commands to drive TeX. LaTeX was designed to be used by authors.

LaTeX gives you the means to typeset beautiful documents. Using it in tandem with your text editor of choice, you can control everything from the spacing of your document to the style of your tables to the (automatic) generation of your works cited page. At a minimum, you will need to use the following typesetting commands to begin your document:

```
\documentclass[12pt]{article}
\begin{document}
... Document body here.
\end{document}
```

Lengthier introductory commands, collectively called the "preamble," are more often the norm. These commands are calls to packages that affect the document's layout, styles, and so forth. This document, for example, was created using the preamble listed below. At this point these particulars are not nearly as important as the general lesson: you have control over the basic appearance of your documents and much of it is established by the packages you invoke in the preamble. Of course, TeX's typesetting algorithms do the heavy lifting.

```
\documentclass[12pt]{article}
\underline{}\usepackage{endnotes}
\usepackage{fullpage}
\usepackage{setspace}
\usepackage{amsfonts, amsmath, amssymb}
\usepackage{dcolumn, pstricks, multirow}
\usepackage{epsfig, subfigure, subfloat, graphicx, float}
\usepackage{anysize, setspace}
\usepackage{verbatim, rotating}
\usepackage{bib}
\usepackage{natbib}
\usepackage{hyperref}
\begin{document}

\title{Introduction to \LaTeX}
\author{Jamie Monogan}
\date{\today}
\maketitle
```

3.1 Including a Package in a Header File

By adding-in packages such as those listed in the previous preamble, we can gain added functions and features for our document. For example, if we normally want to make a comment, we precede it with the character `%`. But if we want a multi-line comment, this can be tedious. A nice feature of the "verbatim" package is it allows us to have a multi-line comment environment. Try adding the header, `\usepackage{verbatim}` to your document. Now I should be able to add a lot of text that will not be included in my final document. Try adding this code:

```
\begin{comment}
LaTeX is rude.
\end{comment}
```

Another package I like to add is `anysize`. By including this header, you can easily set the margin sizes in the preamble with the command `\marginsize{1in}{1in}{1in}{1in}`. The command sets the left, right, top, and bottom margins, in that order.

3.2 Section Headings

Every article uses headings. Creating headings in TeX is straightforward. To create the "Getting Started" heading above I used the command listed below. Note that adding an asterisk after the word "section" gives a non–numbered heading.

```
\section{Getting Started}
or
\section*{Getting Started}
```

Creating sub–headings is also possible, for example:

```
\subsection{Headings}
or
\subsubsection{Headings}
```

4 Special Characters

In LaTeX, several characters have special meaning for programming purposes. Some examples are: `&` separates columns in tables, `$` starts a math environment, `%` starts a comment, and `\` precedes nearly any active command. If you want to include one of these characters in your text, LaTeX needs a signal that this is how you want to use the character. The code to include these characters as well as other tricky characters, is listed below:

#	\#
$	\$
%	\%
&	\&
_	_
{	\{
}	\}
^	\^{}
~	\~{}
\	$\backslash$
—	---
“	``
”	''

Please note: The open quote uses the character on the top-left corner of the keyboard. There are four dash commands, each of which can be called by a different number of hyphens.

5 LaTeX Environments

LaTeX offers several paragraph environments, each of which gives TeX specific formatting instructions. First, every environment begins and ends in the same manner:

```
\begin{environment-name}
...
\end{environment-name}
```

5.1 Bulleted Lists

Creating a bulleted list requires the following environment:

```
\begin{itemize}
\item{McCants}
\item{Felton}
\item{Williams}
\end{itemize}
```

Which gives:

- McCants
- Felton
- Williams

By including a character in square brackets after `\item`, then we can use any kind of bullet we like. For example, including the header `\usepackage{amssymb}`, allows us to write `\item[$\square$] Lawson`.

5.2 Numbered Lists

Creating a numbered list requires the following command sequence:

```
\begin{enumerate}
\item{McCants}
\item{Felton}
\item{Williams}
\end{enumerate}
```

Which gives:

1. McCants
2. Felton
3. Williams

Note: lists may be embedded within lists to a maximum of four nested levels.

5.3 Description Lists

```
\begin{description}
\item[Forward] McCants
\item[Guard] Felton
\item[Forward] Williams
\end{description}
```

Which gives:

Forward McCants

Guard Felton

Forward Williams

6 Creating Tables

Tables are one of the more complicated aspects of LaTeX, but the tables you can create are far superior to those constructed using proprietary word processing software. What is perhaps most confusing about LaTeX tables is that you need two commands to create one. The "tabular" environment is the body of the table itself. It usually sits inside a "table" environment, which controls captions, labels, and the placement of the table on the page. Some LaTeX front ends have automated code builders that facilitate table construction. Of course, we also can simply write the code ourselves. Let's do this to produce Table 1:

Table 1: Manipulations of Coverage Source and Coverage Content

	Message Content	
Message Source	Negative	Positive
FOX News	x	x
CNN	x	x
Source: fabrication		

```
\begin{table}[h]
  \centering
  \caption{Manipulations of Coverage Source and
        Coverage Content}\label{cross}
\begin{tabular}{l|cc}
\hline \hline
&\multicolumn{2}{c}{\textbf{Message Content}}\\
  \textbf{Message Source} & Negative & Positive \\
\hline
  FOX News & x & x \\
  CNN & x & x \\
\hline \multicolumn{3}{l}{\small{Source: fabrication}}\\
\hline \hline
\end{tabular}
\end{table}
```

Notice the code on the third line, `\label{cross}`. This means we can always refer to this table number in text by using the command `\ref{cross}`. In addition to tables, `\label` and `\ref` can be used to call the numbers of sections, equations, or figures.

6.1 Easy Tables from R

If you use R, it is easy to convert any matrix or model output into a ready-to-use, error-free tables by installing the `xtable` package in R. Within R, if I ran a regression that I named `hmnrghts.model`, I could get my table by entering the following syntax:

```
   > install.packages("xtable")
   > library(xtable)
   > xtable(hmnrghts.model)
```

Which gives this output:

```
% latex table generated in R 2.6.2 by xtable 1.5-2 package
% Tue Jun 10 00:06:33 2008
\begin{table}[ht]
\begin{center}
\begin{tabular}{rrrrr}
  \hline
 & Estimate & Std. Error & t value & Pr($>$$|$t$|$) \\
  \hline
```

```
(Intercept) & 3.9095 & 0.1740 & 22.46 & 0.0000 \\
  democ & $-$0.0958 & 0.0237 & $-$4.04 & 0.0001 \\
  gnpcats & $-$0.3256 & 0.0588 & $-$5.54 & 0.0000 \\
   \hline
\end{tabular}
\end{center}
\end{table}
```

After pasting this output into LaTeX, I add the following line of code after `\begin{center}` to get a caption:

```
\caption{Regression Model of Human Rights Violations\label{human}}
```

This produces Table 2:

Table 2: Regression Model of Human Rights Violations

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.9095	0.1740	22.46	0.0000
democ	−0.0958	0.0237	−4.04	0.0001
gnpcats	−0.3256	0.0588	−5.54	0.0000

6.2 Easy Tables from Stata

Stata also can create easy tables with the user-written package `outtex`. The example code below as well as more information are posted at: `http://www.ats.ucla.edu/stat/stata/latex/estimates.htm`. To install, type `findit outtex` into your Stata console, and follow the links to install. Stata will then produce a ready-to-use table simply by following our estimation command with `outtex` on the next line:

```
use http://www.ats.ucla.edu/stat/stata/notes/hsb2, clear
regress write math female
outtex
```

This produces the following code:

```
{
\begin{table}[htbp]\centering
 \caption{Estimation results : regress
\label{tabresult regress}}
\begin{tabular}{l c c }\hline\hline
\multicolumn{1}{c} {\textbf{Variable}}
 & {\textbf{Coefficient}}  & \textbf{(Std. Err.)} \\ \hline
math  &  0.633  & (0.053)\\
female  &  5.218  & (0.998)\\
Intercept  &  16.614  & (2.909)\\
\hline
\end{tabular}
\end{table}
}
```

Pasted into LaTeX exactly, this produces table 3. Again, we can choose to manipulate the code by changing the caption or label name if we wish.

Table 3: Estimation results : regress

Variable	**Coefficient**	**(Std. Err.)**
math	0.633	(0.053)
female	5.218	(0.998)
Intercept	16.614	(2.909)

7 Inserting Figures

There are two means of including figures in a document, and you have to choose one for the entire document. First, we consider postscript figures, which is probably the preferred method when possible. Second, we consider pdf-ready figures, which may be more user-ready. For convenience, our code today will focus on the pdf-ready method.

7.1 Postscript Figures

The big advantage to using postscript figures, especially encapsulated postscript (EPS), is that this is publishers' preferred format for graphic files.[1] The reason is that EPS is a set of commands for drawing a graph, to be interpreted by the postscript driver chip. As such, it has no precision loss or rounding error. If, in contrast, you supply printed graphs, then their resolution can never be better than the printer on which you created them. Thus they come out in low resolution on the high resolution printers that publishers employ.

Stata, SAS, & `R` (but not SPSS) all can save figures in at least one of the two formats, `.ps` or `.eps`. If you plan to use postscript figures regularly, you may want to download a free program called "Ghostscript" from the web. Ghostscript is freeware that translates postscript (and encapsulated postscript) command language for a printer into visible figures on screen. Without it, `.ps` and `.eps` formatted figures sometimes do not appear on screen (in the dvi file) and cannot be printed on a non-postscript printer. Ghostscript, which TeX recognizes automatically, resolves these issues.

You may also want to download and configure a program called "wmf2eps", which allows you to cut and paste graphics from MS Excel and other programs that are based on the `.wmf` graphic format (i.e., "Windows Media File"). Wmf2eps converts `.wmf` files into high quality `.eps` files. The standard routine is to paste into wmf2eps, edit the graphic to change the size (width=5 inches works well), and then convert to the desired format.

[1]Much of this section is drawn from Jim Stimson's notes on LaTeX. My thanks to Jim for sharing.

To compile a TEXdocument with postscript figures, you need to include in the header: `\usepackage{epsfig,graphicx}`. Some example syntax for including a figure is shown here:

```
\begin{figure}[hbt]
\centerline{\includegraphics[width=5.25in,height=3in]{gdpepi.eps}}
\caption{Annual Percent Change in U.S. Gross Domestic Product and
the Economic Performance Index, 1980 to 2002} \label{gdpepi}
\end{figure}
```

IMPORTANT: You should place the graphical file in the same directory as your TEX document. When compiling, you must first TEX-ify to create at `.dvi` file. Then use `dvi2pdf`. NOTE: Whenever using `dvi2pdf`, `dvi2ps`, or any other file conversion button in WinEdt, the button will not light-up until an input file of the required format is available.

7.2 PDF-ready Figures

Alternatively, if creating postscript figures proves tricky, or if you need to use the figures in another setting where postscript is inappropriate, you may incorporate figures from several pdf-compatible formats. These are files ending in `.pdf`, `.jpg`, `.jpeg`, and `.png`. When incorporating files of these formats, you still need to include `graphicx` as a header file, and the code is basically identical to the postscript figure code:

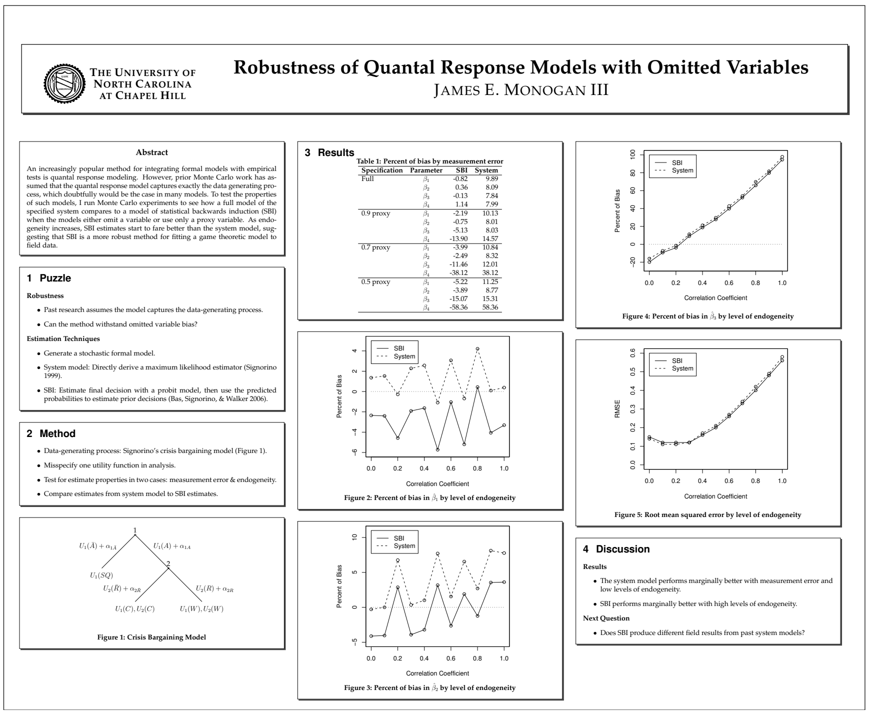
```
\begin{figure}[hbt]
\centerline{\includegraphics[width=4.5in]{polmeth07_poster2.pdf}}
\caption{An Example Poster} \label{poster}
\end{figure}
```

This produces figure 1. Again, be sure to include the figure in the same folder as your `.tex` file. When compiling, use `pdflatex` to create your document in a single step.

8 Citations

LATEX will create a works cited page for you, similar to the Endnote package in Microsoft Word. There are a few ways to set-up a bibliography in LATEX. One option is to use the `harvard` header file, and another is the `natbib` header file.

To use this feature, you will first need to install a BibTeX style folder in the proper directory. You can obtain the APSA style format (`apsr.bst`) and the bib style file (`bib.sty`) from my website. You will need to place these files in a directory such as: `C:\Program Files\MikTeX\tex\latex\my_styles\` on a PC or `/usr/local/texlive/texmf-local/tex/latex/local` on a Mac. Feel free to use the master.bib file on my website (an update of a file Jim Stimson gave me). (You can save this webpage with the extension ".bib"). Once you have the Master.bib file, you should place it in a directory resembling the following: `C:\Program Files\MikTeX\tex\bibtex\my_bibs\` on a PC or `/usr/local/texlive/texmf-local/bibtex/bib/local` on a Mac. Please note: exact



THE UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

Robustness of Quantal Response Models with Omitted Variables

JAMES E. MONOGAN III

Abstract

An increasingly popular method for integrating formal models with empirical tests is quantal response modeling. However, prior Monte Carlo work has assumed that the quantal response model captures exactly the data generating process, which doubtfully would be the case in many models. To test the properties of such models, I run Monte Carlo experiments to see how a full model of the specified system compares to a model of statistical backwards induction (SBI) when the models either omit a variable or use only a proxy variable. As endogeneity increases, SBI estimates start to fare better than the system model, suggesting that SBI is a more robust method for fitting a game theoretic model to field data.

1 Puzzle

Robustness

- Past research assumes the model captures the data-generating process.
- Can the method withstand omitted variable bias?

Estimation Techniques

- Generate a stochastic formal model.
- System model: Directly derive a maximum likelihood estimator (Signorino 1999).
- SBI: Estimate final decision with a probit model, then use the predicted probabilities to estimate prior decisions (Bas, Signorino, & Walker 2006).

2 Method

- Data-generating process: Signorino's crisis bargaining model (Figure 1).
- Misspecify one utility function in analysis.
- Test for estimate properties in two cases: measurement error & endogeneity.
- Compare estimates from system model to SBI estimates.

Figure 1: Crisis Bargaining Model

3 Results

Table 1: Percent of bias by measurement error

Specification	Parameter	SBI	System
Full	β_1	-0.82	9.89
	β_2	0.36	8.09
	β_3	-0.13	7.84
	β_4	1.14	7.99
0.9 proxy	β_1	-2.19	10.13
	β_2	-0.75	8.01
	β_3	-5.13	8.03
	β_4	-13.90	14.57
0.7 proxy	β_1	-3.99	10.84
	β_2	-2.49	8.32
	β_3	-11.46	12.01
	β_4	-38.12	38.12
0.5 proxy	β_1	-5.22	11.25
	β_2	-3.89	8.77
	β_3	-15.07	15.31
	β_4	-58.36	58.36

Figure 2: Percent of bias in $\hat{\beta}_1$ by level of endogeneity

Figure 3: Percent of bias in $\hat{\beta}_2$ by level of endogeneity

Figure 4: Percent of bias in $\hat{\beta}_3$ by level of endogeneity

Figure 5: Root mean squared error by level of endogeneity

4 Discussion

Results

- The system model performs marginally better with measurement error and low levels of endogeneity.
- SBI performs marginally better with high levels of endogeneity.

Next Question

- Does SBI produce different field results from past system models?

Figure 1: An Example Poster

path directories will vary among operating systems, releases of LaTeX, and versions of releases. Other names for this folder might include: MiKTeX or TeXLive. Be sure to find the correct folder for your release of LaTeX.

- To invoke the APSA bibtex package, you will need to include the following in your preamble. The last line (`bibpunct`) manipulates punctuation used in parenthetical citations, so you may adjust these accordingly:

  ```
  \usepackage{bib}
  \usepackage{natbib}
  \bibpunct[, ]{(}{)}{;}{a}{}{,}
  ```

- Alternatively, you could replace the reference to `natbib` with `\usepackage[abbr]{harvard}`.

- You'll also need to include the following at the end of your document:

  ```
  \bibliographystyle{apsr}
  \bibliography{...bibtex filename here...}
  ```

- Some common natbib citation commands include:

  ```
  1. \citep[213]{Zaller:1992}

  2. \citeyearpar[100]{Erikson:2002}
  ```

3. `\nocite{Converse:1964}`

4. `\citet{Monogan:2013}`

- Which produce, respectively:
  1. (Zaller 1992, 213)
  2. (2002, 100)
  3. no citation, but noted in references
  4. Monogan (2013)
- Note that alternative bibliography packages use different commands, so in a given document be careful to recognize which set of rules you are applying.
- A useful online resource for `natbib` is: `http://merkel.zoneo.net/Latex/natbib.php`.
- To add to your master bibliography, notice the field syntax. My website also explains how Google Scholar (`http://scholar.google.com`) can produce new entries automatically.
- A final, but less efficient, option is to use the `thebibliography` environment to create your own bibliography. You actually will find that BibTeXing your document will create a `bbl` file that has `thebibliography` code if, at the final stage of submitting something, you want to submit a self-contained document.

9 Additional Useful Commands

- Footnotes

  ```
  See below\footnote{It works!}
  ```

  See below[2]
- `\newpage` skips to the next page in the document, as does `\clearpage`, which has a tendency to insert floating graphs
- Italics (Note also: `\emph`)

  ```
  \textit{italicized text here}
  ```

  italicized text here
- Boldface

[2] It works!

`\textbf{text in bold here}`

text in bold here

- Sans serif font

  `\textsf{sans serif font}`

  sans serif font

- Underlined text

  `\underline{underlined text here}`

  underlined text here

- courier font

  `\texttt{courier font}`

  `courier font`

- Normal roman font

  `\textnormal{text in normal roman font}`

  text in normal roman font

- Reproduce ascii text verbatim (for in-line text)

  `\verb|t(x)%*%x|`

  `t(x)%*%x`

- Reproduce ascii text verbatim (for a long section, requires use of the `verbatim` header)

```
\begin{verbatim}
> #Logit Model in R:
> mil.model <- glm(military ~ lpop + gnpcats, family=binomial(link=logit))
> summary(mil.model)
\end{verbatim}
```

```
> #Logit Model in R:
> mil.model <- glm(military ~ lpop + gnpcats, family=binomial(link=logit))
> summary(mil.model)
```

- Print a live link: `\url{http://www.unc.edu}` (requires use of the `hyperref` header)
  `http://www.unc.edu`

10 Generating New Commands

Sometimes you will find that typing LaTeX commands can be burdensome. For example, if you regularly referred to matrix $\mathbf{X}_{ij}$ in the text, you would need to type `$\mathbf{X}_{ij}$` every time. The `newcommand` feature allows you to create short commands in the place of longer ones. By way of example, Evan Parker-Stephen always begins his documents with these new commands:

```
\newcommand{\be}{\begin{enumerate}}
\newcommand{\ee}{\end{enumerate}}
\newcommand{\bq}{\begin{quote}}
\newcommand{\eq}{\end{quote}}
\newcommand{\bd}{\begin{description}}
\newcommand{\ed}{\end{description}}
\newcommand{\bi}{\begin{itemize}}
\newcommand{\ei}{\end{itemize}}
\newcommand{\Mc}{$M_{i}^{(c)}$}
\newcommand{\Ma}{$M_{i}^{(a)}$}
\newcommand{\Wa}{$\mathbf{W}^{(a)}_t$}
\newcommand{\Wp}{$\mathbf{W}^{(p)}_{it}$}
\newcolumntype{d}[1]{D{.}{.}{#1}}
```

11 Math in LaTeX

The math environment in LaTeX expects different syntax than the regular text-processing environment. We can access the math environment in four ways: in-line text, the `equation` environment, the `eqnarray` environment, and the `array` environment (for matrices).

The easiest call to the math environment, in-line text, starts and ends with a `$`. Hence, `$x^2$` returns x^2, or `$y_i$` returns y_i. Normally, this environment will try to cram your equation into the normal space of a text line. For example, try the the summation command: `$\sum_{i=1}^{n}{x^2}$`. If you prefer, though, you can use the `\displaystyle` command: this makes the equation look sharper, but adds whitespace before and after the line to make room. See how the summation looks with this option: `$\displaystyle\sum_{i=1}^{n}{x^2}$`.

A second option that we have is to use the `equation` environment:

```
\begin{equation}
\frac{\partial}{\partial x} \{z*x^3\} = 3z*x^2\label{derivative}
\end{equation}
```

This produces equation 1, which I can refer to in text by typing `\ref{derivative}`.

$$\frac{\partial}{\partial x}\{z * x^3\} = 3z * x^2 \tag{1}$$

A third option is to use the `eqnarray` environment. This syntax also illustrates how to suppress equation numbering (which also applies in `equation`) and write Greek letters:

```
\begin{eqnarray}
\phi(x) &=& \frac{1}{\sqrt{2\pi\sigma^2}}
    \exp\{\frac{-(x-\mu)^2}{2\sigma^2}\}\label{normal}\\
\Phi(x) &=& \int_{-\infty}^{x} \phi(t)dt\nonumber
\end{eqnarray}
```

Which produces:

$$\begin{aligned}
\phi(x) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{\frac{-(x-\mu)^2}{2\sigma^2}\} \qquad (2)\\
\Phi(x) &= \int_{-\infty}^{x}\phi(t)dt
\end{aligned}$$

Notice how I added-in the `&` symbol: `eqnarray` allows two tab symbols, with the formatting that the first column is right-justified, the second is centered, and the third is left-justified. By using tabs, it is easier to line-up the equations, which looks nice. Although a number of complex symbols have been illustrated here, WinEdt also has a nice drop-down menu of symbol commands.

As a final example, let's create a matrix using this code:

```
\[ \left[
\begin{array}{ccc}
a   &b  &c\\
d   &e  &f\\
g   &h  &i
\end{array}
\right] \]
```

Which produces the matrix shown below. The only optional parts of this code are `\left[` and `\right]`, which control what symbol will surround the matrix.

$$\left[\begin{array}{ccc} a & b & c\\ d & e & f\\ g & h & i \end{array}\right]$$

References

Converse, Philip E. 1964. The Nature of Belief Systems in Mass Publics. In *Ideology and Discontent*, ed. David E. Apter. Ann Arbor: University of Michigan Press.

Erikson, Robert S., Michael B. MacKuen and James A. Stimson. 2002. *The Macro Polity.* New York: Cambridge University Press.

Monogan, III, James E. 2013. "Strategic Party Placement with a Dynamic Electorate." *Journal of Theoretical Politics* 25(2):284–298.

Zaller, John R. 1992. *The Nature and Origins of Mass Opinion.* New York: Cambridge University Press.