

Applied L^AT_EX for Researchers

Daniel Sánchez Pazmiño, MA

Laboratorio de Investigación para el Desarrollo del Ecuador

February 2025

Math Mode Revisited

Math Mode Revisited

- Math mode is arguably L^AT_EX's most powerful feature
 - However, its syntax often takes some time to master
- Fortunately, tools have been developed to help users write L^AT_EX code more easily
- In this section, we will explore some of these tools

Using AI

- Never ignore AI in your workflows
 - AI will not replace you, someone who knows how to use it will
- ChatGPT can help you with any sort of text generation or syntax question
 - Ask it to generate an equation in L^AT_EX for you
 - Alternatively, you may feed it a picture of an equation and ask it to generate the L^AT_EX code for you
- Writefull AI is Overleaf's own AI assistant
 - Cheaper than ChatGPT, but still not free
- Check out There's an AI for That for more AI tools
 - A quick search yields several other tools, such as LaTeX AI.

Codecogs

- Codecogs is a free online L^AT_EX equation editor
- Following a more “click and drag” approach, it is a good tool for beginners
 - Much like Overleaf’s equation editor, you can click on symbols to add them to your equation
 - It also has a “Copy to Clipboard” button, which makes it easy to paste the equation into your document

Mathpix

- Mathpix is one of the best tools for converting images of equations into L^AT_EX code
- Available as a desktop app, you can take a screenshot of an equation and Mathpix will convert it into L^AT_EX code
 - It also has a mobile app, which is useful for taking pictures of equations in textbooks
- Without spending any money, it will allow you to convert a limited number of equations per month
 - However, the paid version is not expensive

Word Equations

- If for any reason you need to write a Word document with complex equations, you can actually write the equations in L^AT_EX and then convert them to Word format
- This will require you to turn on the “LaTeX” option in the Equation tab, rather than the “Unicode” option.

Creating Bibliographies

Creating Bibliographies

- The bibliography or reference list is a crucial part of any academic paper.
- In L^AT_EX, you can create a bibliography using several different packages
 - natbib, biblatex, and apacite are some of the most popular
 - Each package has its own syntax and features
- In this section, we will focus on biblatex, which is the most modern and flexible of the three
 - The workflow is similar to referencing in Quarto/RMarkdown

Where to store your bibliography

- You can create a bibliography in L^AT_EX on your own, but it is much easier to use a reference manager
 - Zotero, Mendeley, and EndNote are some of the most popular reference managers
- Zotero is free and open-source, and it has a plugin for Word and LibreOffice
 - It also has a plugin for Google Docs, which is useful for collaborative writing
 - Here, we will focus on Zotero
- You can add references to Zotero using the Chrome add-on, or by manually entering them

Loading the package into L^AT_EX

- To use bibl_atex, you need to load the package in the preamble of your document
 - You also need to specify the style of the bibliography
 - The most common styles are `apa`, `mla`, and `chicago`
- A very good guide on citation is on the Overleaf website
 - Bibliography management in L^AT_EX
 - Bibliography management with bibl_atex

Adding the bibliography package biblatex

```
\usepackage[style=apa]{biblatex}
```

- The style option specifies the style of the bibliography
 - The apa style is based on the American Psychological Association's citation style
 - You can find a list of all the available styles in the biblatex documentation
- You also need to specify the .bib file that contains your references

The .bib file

- The .bib file is a plain text file that contains all the references you want to include in your bibliography
 - Each reference is stored in a separate entry
 - The format of the entries is specified by the bibl_atex package
-
- It follows a L^AT_EX-like syntax, with each entry starting with an @ symbol
 - The type of the entry is specified after the @ symbol
 - The fields of the entry are enclosed in curly braces {}
- Don't try to write the .bib file by hand!
 - Zotero can export your references to a .bib file
 - You only need to put it in the same folder as your .tex file or upload it to Overleaf

The .bib file

```
@article{einstein,
  author = {Albert Einstein},
  title = {Zur Elektrodynamik bewegter K{"o"}rper},
  journal = {Annalen der Physik},
  volume = {322},
  number = {10},
  pages = {891--921},
  year = {1905},
  publisher = {Wiley-VCH},
  doi = {10.1002/andp.200510497}
}
```

Adding the bibliography file

```
\addbibresource{references.bib}
```

- The `\addbibresource` command specifies the `.bib` file that contains your references
 - You need to put the `.bib` file in the same folder as your `.tex` file or upload it to Overleaf
 - The file extension `.bib` is optional
- You can add multiple `.bib` files to your document
 - Just use the `\addbibresource` command multiple times
- The `\printbibliography` command prints the bibliography in your document

Adding the bibliography

```
\printbibliography
```

- The `\printbibliography` command prints the bibliography in your document
 - You can specify the title of the bibliography by passing an argument to the command
 - You can also filter the bibliography by passing options to the command
 - The options are specified in square brackets []

Citation commands

- To include a citation in your document, you need to use the `\cite` command
 - The `\cite` command takes the key of the reference you want to cite as an argument
 - You can cite multiple references by separating the keys with commas
 - The `\cite` command prints the citation in the text and adds the reference to the bibliography

Citation commands

```
\cite{einstein}
```

- The `\cite` command takes the key of the reference you want to cite as an argument
 - You can cite multiple references by separating the keys with commas
 - The `\cite` command prints the citation in the text and adds the reference to the bibliography
- The `\textcite` command is similar to the `\cite` command, but it prints the author's name in the text
- The `\parencite` command is similar to the `\cite` command, but it adds parentheses around the citation

Citation commands

```
\textcite{einstein} showed that \ldots
```

```
\parencite{einstein}
```

```
\cite{einstein}
```

```
\cite{einstein,dirac}
```

```
\cite{einstein,dirac,bohr}
```

Exporting from statistical software to L^AT_EX

Exporting from statistical software to L^AT_EX

- If you are using statistical software to analyze your data, you can export the results to L^AT_EX
 - This is useful if you want to include tables, figures, or other output in your document
 - Most statistical software packages have built-in support for exporting to L^AT_EX
- Here, we will focus on how to export tables from R and Stata to L^AT_EX
 - Both have packages that make it relatively easy to export tables to L^AT_EX

The export environment in R

Exporting tables from R

- Many options exist for exporting tables from R to L^AT_EX
 - `stargazer` is likely the most known package for exporting tables to L^AT_EX
 - However, `modelsummary` is a more modern and flexible package
- Other packages, such as `kableExtra`, `xtable`, and `huxtable`, can also be used to export tables to L^AT_EX
 - `kableExtra` is particularly useful for creating complex tables with formatting

An example with stargazer

- We export the mtcars dataset to a L^AT_EX table using stargazer

```
library(stargazer)

data(mtcars)

stargazer(mtcars,
          type = "latex",
          summary = T,
          title = "Descriptive Statistics")
```


modelsummary

- `modelsummary` is a more modern and flexible package for exporting tables to L^AT_EX
 - It is particularly useful for exporting regression tables
 - Exports to a larger amount of formats, including HTML, Word, and Markdown
 - It is also more flexible than `stargazer`
 - Can do summary statistics with `datasummary`
- Check the author's GitHub page for more information
 - Includes awesome vignettes and documentation!








modelsummary::datasummary

- `datasummary` has a wide range of possibilities, you can see the vignettes [here](#).
 - `datasummary_skim` produces quick summary statistics
 - `datasummary_balance` specifically works for “balance tables”
 - `datasummary` is the most general function, and it can be used for a wide range of tables

An example with `modelsummary::datasummary_skim`

```
library(modelsummary)

datasummary_skim(mtcars)
```

	Unique	Missing Pct.	Mean	SD	Min	Median	Max	Hist
mpg	25	0	20.1	6.0	10.4	19.2	33.9	
cyl	3	0	6.2	1.8	4.0	6.0	8.0	
disp	27	0	230.7	123.9	71.1	196.3	472.0	
hp	22	0	146.7	68.6	52.0	123.0	335.0	
drat	22	0	3.6	0.5	2.8	3.7	4.9	
wt	29	0	3.2	1.0	1.5	3.3	5.4	
qsec	30	0	17.8	1.8	14.5	17.7	22.9	

The formula syntax in `datasummary`

- `datasummary` requires a formula (like that in `lm`) to work
 - This allows for amazing flexibility
- For instance, if we want to report the mean of `mpg`, we can use the following formula: `mpg ~ Mean`
 - `Mean` is the special `mean` function in `datasummary`, which is just `mean(x, na.rm = TRUE)`
 - Others available are `Median`, `SD`, `N`, among others
- To do more statistics, sum them to the LHS of the formula
 - to add groups, use the nesting operator `*` in the RHS of the formula

An example with `modelsummary::datasummary`

```
data(mtcars)
```

```
datasummary(mpg ~ Mean, data = mtcars)
```

	Mean
mpg	20.09

An example with `modelsummary::datasummary`

```
datasummary(mpg ~ as.factor(cyl)*(Mean + SD + N), data = mtcars)
```

	4			6			8		
	Mean	SD	N	Mean	SD	N	Mean	SD	N
mpg	26.66	4.51	11	19.74	1.45	7	15.10	2.56	14

Regression tables with `modelsummary`

- `modelsummary` is particularly useful for exporting regression tables to L^AT_EX
 - It can export tables from `lm`, `glm`, `feols` and many other packages for modelling
- It is very customizable, and you can add custom statistics, change the formatting, and add notes to the table
- Check the function's GitHub page for more information

An example with `modelsummary`

```
data(mtcars)

model <- lm(mpg ~ wt + hp, data = mtcars)

modelsummary(model)
```


An example with `modelsummary`

	(1)
(Intercept)	37.227
	(1.599)
wt	-3.878
	(0.633)
hp	-0.032
	(0.009)
Num.Obs.	32
R ²	0.827
R ² Adj.	0.815
AIC	156.7
BIC	162.5
Log Lik	-74.326

modelsummary functionality

Among other things, `modelsummary` can:

- Add a custom covariance matrix (adjust the standard errors)
- Add custom statistics (goodness of fit, for instance)
- Change the shape of the table
- Add notes to the table
- Change the formatting of the table (e.g. rounding)
- Titles, subtitles, exponentiation of coefficients, etc.
- Name titles
- Exporting to HTML, Word, Markdown, and other formats

The Export Environment in Stata

Exporting tables from Stata

- Stata can export to multiple formats, including Excel, LaTeX and Word
- After careful study and comparison, we suggest the best method may be to export to Excel, then use the Excel2LaTeX add-in to convert the table to LaTeX
 - Packages include estout and outreg2, but other methods available.

Using estout