

基本无害的经济学 L^AT_EX 技巧

邓东升

ElegantL^AT_EX 项目组

2019 年 11 月 05 日

此文档为经济学专业的 L^AT_EX 技巧总结，包括环境搭建、基础知识、参考文献以及幻灯片制作等内容，仅作为经济学专业的师生作为入门 L^AT_EX 使用。使用 dlove ♥ 模板和 Xe_L^AT_EX 编译完成。

1 环境搭建

目前 L^AT_EX 的主要发行版本如下：

- MiK_T_EX：Windows 上的发行版，国内的 C_T_EX¹ 使用的就是 MiK_T_EX；
- T_EX Live：编辑器 T_EXworks，跨系统，每年更新一个大版本，最新版为 TeX Live 2019；
- Mac_T_EX：Mac 上的 T_EX Live，为了适应 Mac 系统做了一些细微的调整；

推荐使用 T_EX Live 2019，可以使用默认的 T_EXworks 或者配合其他编辑器（比如 Sublime Text、Visual Studio Code）的插件进行编写，后面我们会细说这部分。

1.1 安装 T_EX Live

首先，我们进入 T_EX Live 的[官网](#)地址，点击页面中的 [download](#)，然后可以选择在线安装或者下载安装文件之后离线安装，推荐使用离线安装。

- **在线安装**：点击 [install-tl-windows.exe](#)（Windows）或者 [install-tl-unix.tar.gz](#)（Linux/Unix），视自己系统选择，然后按照提示进行安装。
- **离线安装**：首先下载镜像文件，点击 [generic mirror.ctan.org url](#)，这个时候我们会跳转到 T_EX Live 的[镜像站](#)，下拉找到国内的镜像站。在国内的镜像列表中，选择离自己比较近的地地区的镜像进行下载²。下载镜像文件之后，使用资源管理器或者镜像挂载工具³进行挂载，**请不要使用压缩软件对镜像文件解压缩**。

更多的关于 T_EX Live 的安装问题，可以参考[嘯行的一份简短的关于 L^AT_EX 安装的介绍](#)。

1.2 配置编译环境

在安装好 T_EX Live 之后，我们需要选择一个编辑器，目前主流的编辑器有：

- WinEdt，T_EX 专用，过去 Windows 上非常流行的编辑器，收费软件，编辑方便，有输入辅助面板，编码支持不好，适合新手，但不推荐；

¹已过时，请尽量不要使用。

²比如上海的用户可以选择[上海交大的镜像地址](#)，然后往上拉找到 T_EX Live，点击进入上海交大 T_EX Live 的[下载地址](#)，选择 `./images/`，然后将 `texlive.iso` 下载即可。

³推荐使用 WinCDEmu 进行挂载，WinCDEmu 下载后直接安装即可，这里不再赘述。

- **T_EXstudio**, T_EX 专用, 开源软件, 兼顾了易用性以及可定制性, 代码提示优秀, 有输入辅助面板, 适合新手, 推荐全阶段使用。
- **T_EXworks**, T_EX 专用, T_EX Live 自带的编辑器, 界面非常简洁, 对于新手不友好, 自动补全功能还可以, 需要的可以参考我之前的一个总结: [T_EXworks 自动补全功能](#), 推荐不想安装额外软件的用户。
- **Sublime Text**, 颜值很高、高可定制化的文本编辑器, 非 T_EX 专用, 付费软件, 界面非常简洁, 自动补全功能完善, 代码高亮非常优秀, 支持自定义代码片段, 插件开发成熟, 极度不适合新手, 另外插件安装可能受限, 极度不适合无法科学上网的用户, 只适合高玩以及颜值主义者。
- **Visual Studio Code**, 微软推出的高可定制化文本编辑器, 非 T_EX 专用, 免费软件。可配置快捷编译按钮, 自动补全、代码高亮很优秀, 插件体验良好, 但由于处于不断更新迭代过程, 中间可能会有重大改动, 需要关注开发者。推荐熟悉 L^AT_EX 的用户使用。

我在我的小圈子里做了一个 L^AT_EX 编辑器体验调查, 表 1 列出了各个编辑器的用户平均评分, 此表为主观打分, 仅供参考。此表的绘制参考了 [Jake \[2019\]](#) 的 TikZ 代码。如果你对这些编辑器有自己的评价, 欢迎 [下载评分表](#), 对自己熟悉的编辑器打分, 然后发给我 ddswhu@outlook.com, 我将加入到这个评分表中。

表 1: L^AT_EX 编辑器对比

	WinEdt	T _E Xstudio	T _E Xworks	Sublime Text	VS Code
插件依赖				L ^A T _E XTools	L ^A T _E X Workshop
主流系统	Win	全平台	Linux/Win	全平台	全平台
软件类型	商业软件	开源软件	开源软件	商业软件	商业软件
软件价格	219 元	0	0	80 美元	0
授权方式	终身/教育			终身/个人	
代码高亮	★★★★☆	★★★★☆	★★★☆☆	★★★★☆	★★★★☆
颜色主题	★★★★☆	★★★★☆	★★★☆☆	★★★★☆	★★★★☆
自动补全	★★★★☆	★★★★☆	★★★☆☆	★★★★☆	★★★★☆
代码片段	★★★★☆	★★★★☆	★★★☆☆	★★★★☆	★★★★☆
辅助输入	★★★★☆	★★★★☆	★★★☆☆	★★★★☆	★★★★☆
开发完成	★★★★☆	★★★★☆	★★★★☆	★★★★☆	★★★★☆
推荐指数	★★★★☆	★★★★☆	★★★☆☆	★★★★☆	★★★★☆

2 基础知识

2.1 命令与环境

命令和环境是 L^AT_EX 最重要的组成部分, 因此我们先了解下这两个概念。

命令由反斜杠 \ 引导, 一般的形式为 `\cmd[可选参数]{必选参数}`, 命令名全部由**英文字母**构成, 并且大小写敏感, 因此 `\LaTeX{}` 是对的, 而 `\latex{}` 是错的。必选参数可能有 0 个或者多个, 无参数的命令为 `\cmd`, 多个参数的形式为 `\cmd[可选参数]{参数 1}{参数 2}...{参数 N}`。下面是一个示例:

无参数命令: `\LaTeX{}` `\vline`

单参数命令: `\textbf{加粗}` `\textit{Italic Style}`

多参数命令: `\rule[2cm]{0.4pt}`

无参数命令: L^AT_EX |

单参数命令: 加粗 *Italic Style*

多参数命令: _____

这里对部分命令解释一下, `\vline` 不带参数时, 会输出一条高度为行高的竖线 (vertical line), 而

`\textbf` 和 `\textit` 分别为字体加粗 (**bold face**) 和斜体 (*italic*) 的命令。`\hrule` 类似于 `\vline`, 用于画一个长为 2 cm, 宽度为 0.4 pt 的水平线段 (horizontal rule)。

环境相对于命令而言, 是更加高层的命令组合, 用于实现一系列功能、格式的定制化。一个环境 (environment) 的基本结构为

<pre>\begin{环境名} % 内容 \end{环境名}</pre>	<pre>\begin{环境名}[可选参数]{必需参数} % 内容 \end{环境名}</pre>
---------------------------------------	---

当没有可选参数或者必需参数, 可以直接将其对应的括号 `[]` 以及 `{}` 去掉。下面以 \TeX 中默认的环境来举例说明环境的用法。如果我们需要对一段文字进行居中, 则可以使用 `center` 环境⁴。

<pre>\begin{center} 落霞与孤鹜齐飞，秋水共长天一色。渔舟唱晚，响穷 彭蠡之滨；雁阵惊寒，声断衡阳之浦。 \end{center}</pre>	落霞与孤鹜齐飞，秋水共长天一色。渔舟唱晚，响穷 彭蠡之滨；雁阵惊寒，声断衡阳之浦。
--	--

关于带参数的环境的使用, 后文有很多例子, 这里不赘述。另外, 你可能会问 \TeX 内置哪些环境可以使用, 以及他们分别用来干嘛的, 关于这部分内容, 你可以参考 [CEU Community \[2019\]](#) 的 [\$\text{\TeX}\$ Standard Environments](#)。

载入 `enumerate` 宏包之后, `列表` 环境可以传入参数改变标签:

<pre>\begin{itemize} \item First item; \item Second item; \item Third item。 \end{itemize}</pre>	<ul style="list-style-type: none">• First item;• Second item;• Third item。
---	--

2.2 最简示例

<pre>\documentclass{article} % 导言区 \begin{document} Hello World. \end{document}</pre>	Hello World.
---	--------------

其中, `\documentclass` 是文档类型命令, 声明本文档是文章 (article), 而正文部分需要放在 `\begin{document}` 和 `\end{document}` 之间, 比如上面例子中的 `Hello World` 就是正文内容。

而在 `\documentclass` 和 `\begin{document}` 之间的部分, 我们称为导言区。在这部分, 我们可以引入功能宏包, 进行选项设置或者定义一些命令等等。 \TeX 本身能做很多事情, 但是很多定制化的功能或者需求需要宏包支持, 因此几乎全部的文档都是基于 \TeX 基本文档类型的命令和环境配合其他宏包命令、环境共同使用的。

2.3 文档类型

\TeX 默认文档类型 (简称文类) 主要有

- 文章文类: `article`;
- 书籍文类: `book`;
- 演示文稿: `beamer`;

⁴居左为 `flushleft` 环境, 居右为 `flushright` 环境。

- 报告文类: report;

声明文类使用 `\documentclass[文类选项]{文类}`, 注意, 文类的声明可以带可选项。以文类 article 为例, 由于 article 的默认纸张为 legal 大小, 将其更改为 A4 纸张, 需要使用 `\documentclass[a4paper]{article}`。我们也可以通过文类选项设置字体大小, 比如正文字体大小为 11pt 的设置为 `\documentclass[a4paper, 11pt]{article}`

2.4 宏包

表 2: 一些常见的功能宏包

数学公式	插图插表	页面元素	字体设置	参考文献	表格
amsmath	graphicx	geometry	fontspec	natbib	booktabs
amsthm	caption	hyperref	ctex	gbt7714	longtable
amssymb	subfig		xeCJK		multirow

2.5 中文支持

目前流行的中文支持有两个方式:

- ctex 宏包, 或者与其相适应的 ctexart 等文类。
- xeCJK 宏包, 需要使用 Xe_{La}TeX 编译。

2.6 数学字母

常用的一些希腊字母见表 3, 需要注意的是这些希腊字母需要在数学模式 (比如 `α`) 或者数学环境中使用。

表 3: 希腊字母表

符号	命令	符号	命令	符号	命令
α	<code>\alpha</code>	ι	<code>\iota</code>	σ Σ	<code>\sigma \Sigma</code>
β	<code>\beta</code>	κ	<code>\kappa</code>	τ	<code>\tau</code>
γ Γ	<code>\gamma \Gamma</code>	λ Λ	<code>\lambda \Lambda</code>	υ Υ	<code>\upsilon \Upsilon</code>
δ Δ	<code>\delta \Delta</code>	μ	<code>\mu</code>	ϕ Φ	<code>\phi \Phi</code>
ϵ	<code>\epsilon</code>	ν	<code>\nu</code>	χ	<code>\chi</code>
ζ	<code>\zeta</code>	π Π	<code>\pi \Pi</code>	ψ Ψ	<code>\psi \Psi</code>
η	<code>\eta</code>	ρ	<code>\rho</code>	ω Ω	<code>\omega \Omega</code>
θ	<code>\theta</code>	ε	<code>\varepsilon</code>		

2.7 文本模式与数学模式

在 \LaTeX 中, 文本和数学是作为两个独立的模式存在的, 如果需要在文本模式中输入数学式, 需要使用英文状态下的美元符号 `$` 将数学命令包围, 比如 `$\alpha$` 输出为 α 。

假设 `$y_{\{i\}}` 是被解释变量的第 `$i$` 次观测, `$x_{\{i\}}` 是解释变量的第 `i` 次观测, 设定回归方程为 `$y_{\{i\}} = \alpha + \beta x_{\{i\}} + \varepsilon_{\{i\}}$`。

假设 y_i 是被解释变量的第 i 次观测, x_i 是解释变量的第 i 次观测, 设定回归方程为 $y_i = \alpha + \beta x_i + \varepsilon_i$ 。

2.8 数学环境

数学环境中，最简单的就是 `equation` 环境，这个环境会对数学公式进行自动编号，如果不需要编号，可以使用 `equation*` 环境。

```
\begin{equation}
y_{\{i\}} = \alpha + \beta x_{\{i\}} + \varepsilon_{\{i\}}
\end{equation}
```

$$y_i = \alpha + \beta x_i + \varepsilon_i \quad (1)$$

3 表格输入

TeX 中表格的输入并不太方便，最简单的一个表格示例如下：

```
\begin{tabular}{ccc}
English & Context & 996 \\
Right & Here & 1024 \\
Chinese & $\alpha$ & $\beta$
\end{tabular}
```

English	Context	996
Right	Here	1024
Chinese	α	β

在上述命令中，创建表格的环境名为 `tabular`，而 `tabular` 后的选项为列的对齐方式，分别有居中对齐 (c)，左对齐 (l)，右对齐 (r)，而同一行的不同列之间用 `&` 隔开，而换行使用 `\\`。很显然，这种表格并不是我们想要的，我们需要加入一些表格框线：

```
\begin{tabular}{|l|c|r|}
\hline
English & Context & 996 \\
Right & Here & 1024 \\
Chinese & $\alpha$ & $\beta$
\hline
\end{tabular}
```

English	Context	996
Right	Here	1024
Chinese	α	β

可以发现，`|` 为表格的列添加竖线，而 `\hline` 为表格的行添加了横线。

3.1 三线表

在实际写作中，我非常推荐大家使用三线表，而不要添加过多的横线或者竖线，利用 `booktabs` 宏包中的 `\toprule`、`\midrule` 以及 `\bottomrule` 能够非常方便的制作出三线表。示例如下：

```
\begin{tabular}{lcr}
\toprule
Language & Infor & Number \\
\midrule
English & Context & 996 \\
Right & Here & 1024 \\
Chinese & $\alpha$ & $\beta$
\bottomrule
\end{tabular}
```

Language	Infor	Number
English	Context	996
Right	Here	1024
Chinese	α	β

3.2 长表格

如果表格非常长，可以使用 `longtable` 代替 `table`。

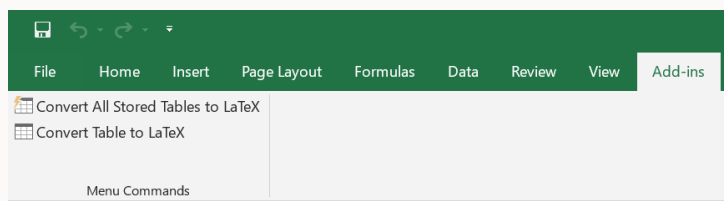


图 1: Excel2 \LaTeX 插件

```
\begin{table}
  \begin{tabular}
    % 表格内容
  \end{tabular}
\end{table}
```

```
\begin{longtable}
  \begin{tabular}
    % 表格内容
  \end{tabular}
\end{longtable}
```

3.3 辅助工具

手动输入表格是一个非常枯燥的过程，而且容易出错，因此我们推荐借助其他工具辅助制作表格，其中个人体验最好的一个工具是 [Excel2 \$\text{\LaTeX}\$](#) 。你可以通过 CTAN 的[下载地址](#)或者[此处](#)下载此插件，将插件下载解压缩之后，双击打开即可使用，不过建议把 `Excel2LaTeX.xla` 置于 Excel 的启动文件夹内，这样以后就不用每次查找这个 Excel 宏才能使用。我本人的 Office 是 2019，对应的 Excel 的启动目录为 `C:\Program Files\Microsoft Office\root\Office16\XLSTART`，如此，在你的 Excel 上方会出现一个插件选项卡，有两个表格转换的选项，见图 1。选中所需要转换的表格，然后选择 `Convert Table to LaTeX` 即可。

另外在线转换的工具 [Table Convert](#) 也可以尝试一下。

3.4 回归表格

```
outreg2
R
Python
```

4 颜色

在 \LaTeX 中，有 7 种内置的颜色，分别是 `white`, `black`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow`。

4.1 定义颜色

4.2 使用颜色

5 文献

5.1 thebibliography 环境

5.2 BibTeX 的使用

5.3 natbib 包

6 幻灯片

Beamer 是 \LaTeX 用于制作幻灯片的一个文类，由于它的格式简洁、易于使用、方便展示数学公式和逻辑演绎，在学术界特别是国外非常受欢迎。下面分别是英文 Beamer 和中文 Beamer 的一个简单示例：

```
\documentclass{beamer}

% title information
\title{An Example of Beamer Class}
\author{Dongsheng DENG}
\institute{Fudan University}
\date{\today}

\begin{document}
\maketitle

\begin{frame}{frame title}
Be honest rather clever.
\end{frame}

\end{document}
```

```
\documentclass{beamer}
\usepackage[UTF8,scheme=plain]{ctex}
% 标题信息
\title{Beamer 文类示例}
\author{邓东升}
\institute{复旦大学}
\date{2019 年 10 月 23 日}

\begin{document}
\maketitle

\begin{frame}{帧标题}
有志者事竟成，百二秦关终属楚。
\end{frame}

\end{document}
```

7 文档说明

本文档使用了 fontspec 和 xeCJK 设置英文字体和中文字体，用户需要的字体列表如下：

表 4: 本文档字体设置

	衬线字体	非衬线字体	等宽字体
英文/fontspec	Amiri	Roboto	Ubuntu Mono
中文/xeCJK	方正书宋简体	方正楷体简体	方正仿宋简体

需要注意的是，在 Win 10 中，安装字体时需要为所有用户安装，否则即便安装了字体， \LaTeX 也无法找到。

另外，本文高亮使用了 minted 宏包，所以，需要调用 -shell-escape 选项并用 \XeLaTeX 进行编译，如果使用命令行编译，命令如下：

```
xelatex --shell-escape main.tex
```

参考文献

Jake. [Drawing Stars/Similar with TikZ](#), 2019. Accessed: 2019-10-29.

Overleaf. [Bibliography management with BibTeX](#), 2019a. Accessed: 2019-10-29.

Overleaf. [Tables](#), 2019b. Accessed: 2019-10-27.

CEU Community. [L^AT_EX standard environments](#), 2019. Accessed: 2019-11-05.