

L^AT_EX 学习

Liripo

2021-03-10

目录

1	⚙️基本配置	2
1.1	下载 Tex 发行版	2
1.2	编译	2
2	📖使用	3
2.1	源文件的基本框架	3
2.2	功能的扩展	4
2.3	LaTeX 用到的文件一览	4
2.4	排版基础	5
	2.4.1 基本的排版命令	6
2.5	表格	6
2.6	插图	7
2.7	数学公式	8
2.8	宏编程	8

前言

L^AT_EX 特点:

- 高质量排版

- 自动编号，比如目录等
- 可以使用各种宏包进行扩展
- ...

1 基本配置

1.1 下载 Tex 发行版

Tex 的发行版我选择使用 TinyTex—一个瘦身版的 [Texlive](#), 它可以使用 R 安装:

```
1 install.packages("tinytex")
2 library(tinytex)
3 install_tinytex()
```

当然, 可以在 [dockerhub](#) 找一些包含全部宏包的官方镜像。 下载之后里面就包含了 `pdflatex`, `xelatex`, `latexmk` 等应用程序, 但 Tinytex 是轻量化的, 缺少很多宏包, 都需要使用 `tlmgr` 管理下载。由于 CTAN 下载很慢, 需要配置镜像。可以使用 `tlmgr` 在命令行配置, 也可以使用封装的 R 函数配置:

```
1 tlmgr(c("option", "repository",
2        "https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/tlnet"))
```

1.2 编译

直接使用 R 函数 `tinytex::latexmk` 编译的话会自动安装缺失的包, 当然最大的好处我觉得是 `error` 报错的时候会在停止编译错误后面的内容。可以选择命令行使用 `latexmk` 编译 `tex` 文件 (遇到错误也会将后面的内容编译出来), `latexmk` 相较于使用 `xelatex` 优点是:

- 一条命令自动搞定交叉引用、参考文献，避免多次编译
- 支持增量编译，也就是说 latexmk 只编译新增的文件内容，从而显著提升编译速度

在命令行使用：

```

1 latexmk -pdf -c -xelatex test.tex
2 #-pdf: 最终输出的产物是 pdf，一般都要加上
3 #-pv: 编译完成后打开阅读器进行预览，可以酌情加或者不加
4 #-xelatex: 指定编译的引擎
5 #-c 可以清理当前工作目录中编译生成的中间文件，
   ↳ 这个命令不会清理 .pdf（最终产物）和 .synctex.gz
   ↳ （用于正反搜索）
6 #-C 彻底清理

```

2 ■ 使用

LaTeX 中命令有两种：一种以反斜线\ 开头，例如\LaTeX会打出 LaTeX。另一种是：

```

1 \command[option]{arguments}

```

其中，方括号中的是可选的（称为选项），花括号中的参数是必需的。

2.1 源文件的基本框架

```

1 \documentclass{ctexart} %
   ↳ 指定文档类型, article, ctexart, book等
2 % 导言区：全局设置，宏包调用等
3 \begin{document}

```

```
4 % 正文部分
5 Hello world!
6 \end{document} % 结束
```

2.2 功能的扩展

宏包调用方法 (只能出现在导言区)

```
1 \usepackage[选项]{宏包名}
```

当然,\usepackage 可以一次性调用多个宏包, 在 package-name 中用逗号隔开。这种用法一般不要指定选项。每个宏包的帮助文档可在<https://texdoc.org/index.html>查看, 或者使用 texdoc 程序查看。

2.3 LaTeX 用到的文件一览

```
1 .sty 宏包文件。宏包的名称与文件名一致。
2 .cls 文档类文件。文档类名称与文件名一致。
3 .bib BIBTEX 参考文献数据库文件。
4 .bst BIBTEX 用到的参考文献格式模板。LATEX
  ↳ 在编译过程中除了生成 .dvi 或 .pdf 格式的文档外,
  ↳ 还可能会生成相当多的辅助文件和日志。一些功能如交叉引用、
  ↳ 参考文献、目录、索引等, 需要先通过编译生成辅助文件,
  ↳ 然后再次编译时读入辅助文件得到正确的结果, 所以复杂的
  ↳ LATEX 源代码可能要编译多次(可以使用latexmk进行一次编译)
5 .log 排版引擎生成的日志文件, 供排查错误使用。
6 .aux LATEX 生成的主辅助文件, 记录交叉引用、目录、
  ↳ 参考文献的引用等。
7 .toc LATEX 生成的目录记录文件。
8 .lof LATEX 生成的图片目录记录文件。
```

```

9 .lot LATEX 生成的表格目录记录文件。
10 .bbl BIBTEX 生成的参考文献记录文件。
11 .blg BIBTEX 生成的日志文件。
12 .idx LATEX 生成的供 makeindex 处理的索引记录文件。
13 .ind makeindex 处理 .idx 生成的用于排版的格式化索引文件。
14 .ilg makeindex 生成的日志文件。
15 .out hyperref 宏包生成的 PDF 书签记录文件。

```

2.4 排版基础

```

1 \LaTeX{}源代码中，空格键和 Tab
   键输入的空白字符视为“空
   格”。连续的若干个空白字符视
2 为一个空格。一行开头的空格忽略
   不计。
3
4
5 行末的换行符视为一个空格；但连
   续两个换行符，也就是空行，会
   将文字分段。多个空行被
6 视为一个空行。\\
7 而使用\verb|\\|可以强制换行，然
   后段落缩进则使用宏包
   identfirst处理。更好地应该
   是使用\verb|\par|生成新段
   落。

```

LaTeX 源代码中，空格键和 Tab 键输入的空白字符视为“空格”。连续的若干个空白字符视为一个空格。一行开头的空格忽略不计。

行末的换行符视为一个空格；但连续两个换行符，也就是空行，会将文字分段。多个空行被视为一个空行。

而使用\\可以强制换行，然后段落缩进则使用宏包 identfirst 处理。更好地应该是使用\par生成新段落。

断行：

在西文排版中，断行的位置优先选取在两个单词之间，也就是在源代码中输入的“空格”。

但是这样一些很长的字符串没有空格，未必能自动断词，这时就必须在单词里手动使用\命令指定断词的位置。/home/liripo/test/aaa/NGS_fq_to_bam, 这种情况就会右边文本溢出了。

具体的处理就是在 to 前面加个\了。但是这样当你在这个段落增加了一个文字，就只得改了。一个可行的方法就是在这个长西文中加大量的\。

2.4.1 基本的排版命令

```

1 \chapter{ title }
2 \section{ title }
3 \subsection{ title }
4 \subsubsection{ title }
5 \paragraph{ title }
6 \subparagraph{ title }
7 \part{<title>}
8 \tableofcontents %生成目录,可使用tocbibind, titletoc、
   ↪  tocloft等宏包定制目录
9 \maketitle %生成标题
10 %对齐环境
11 \begin{center} ... \end{center}
12 \begin{flushleft} ... \end{flushleft}
13 \begin{flushright} ... \end{flushright}
14 %列表环境
15 \begin{enumerate}
16   \item ...
17 \end{enumerate}

```

2.5 表格

表格环境: tabular

```

1 \begin{tabular}[竖向位置]{列格式}
2   first line \\
3   .

```

```

4      .
5      .
6      last line \\
7 \end{tabular}

```

- 竖向位置: 表格在竖直方向与外部文本行的相对位置, 取值有 t 或 b, 分别表示上对齐和下对齐, 缺省为居中对齐
- 列格式: 用于指定各列的格式, 常用的参数有: l, c, r, |, ||, ...
- 行与行之间用 \\ 分隔, 每一行的列与列之间用 & 分隔
- 行与行之间的分界线: \hline: 与表格同宽的水平线
\cline{m-n}: 从第 m 列开始到第 n 列结束的水平线
- 高级表格: longtable, diagbox, colortbl, booktabs 等宏包

当然, 更便捷的可以使用 R 包 [kableExtra](#) 生成 \LaTeX 表格。

2.6 插图

```

1 \usepackage{graphicx}
2 \includegraphics[选项]{图形文件名}
3 %例如
4 \includegraphics[width=0.3\textwidth]{tiger.png}

```

浮动图表

```

1 \begin{figure}[位置]
2   \centering %居中,
3   \includegraphics[选项]{图形文件名}
4   \caption{注释}
5   \label{标签} %用于交叉引用

```

```
6 \end{figure}
```

- 浮动图表: 自动调整图表位置, 避免出现大片的空白
- 位置选项的取值: $h \rightarrow \text{here}$, $t \rightarrow \text{top}$, $b \rightarrow \text{bottom}$, $p \rightarrow \text{page}$
- 优先顺序: $h \rightarrow t \rightarrow b \rightarrow p$
- 缺省值为 tbp
- 固定在当前位置 (即是固定在源码位置): $H \rightarrow$ 需加载 `float` 宏包。
- 高级功能: `float`, `caption`, `subfigure` 宏包。

2.7 数学公式

这应该才是 \LaTeX 最常使用的部分, 但目前暂时没怎么接触。简单的例子

```
1 %因为markdown也使用$$添加公式,  
   所以比较习惯用这个。  
2 $$  
3 E=mc^2  
4 $$
```

$$E = mc^2$$

2.8 宏编程

\LaTeX 难点, 目前还不是很了解。