

# latex 使用指南

摘要: 对 latex 中高频使用的语法进行记录

关键字:

## 目录

一、插入 pdf .....	2
二、浮动体环境 .....	2
2.1 插入图片 .....	2
2.2 插入表格 .....	2
2.2.1 宽表格 .....	3
2.2.2 长表格 .....	4
2.2.3 定宽表格 .....	5
三、插入题注 .....	5
四、导入参考文献 .....	5
附录 A 插入代码 .....	7
1.1 input 方法 .....	7

## 一、插入 pdf

`include`和`input`命令皆可，记得导入 `pdfpages` 宏包。

## 二、浮动体环境

一般来说，为了排版的美观，都会将表格图片等内容放入浮动体环境，这块主要是针对插入的图片或者表格过大的情况作说明。

### 2.1 插入图片

图片过大的问题相对容易解决，只需要直接在修改`width`的可选参数即可引用图片1

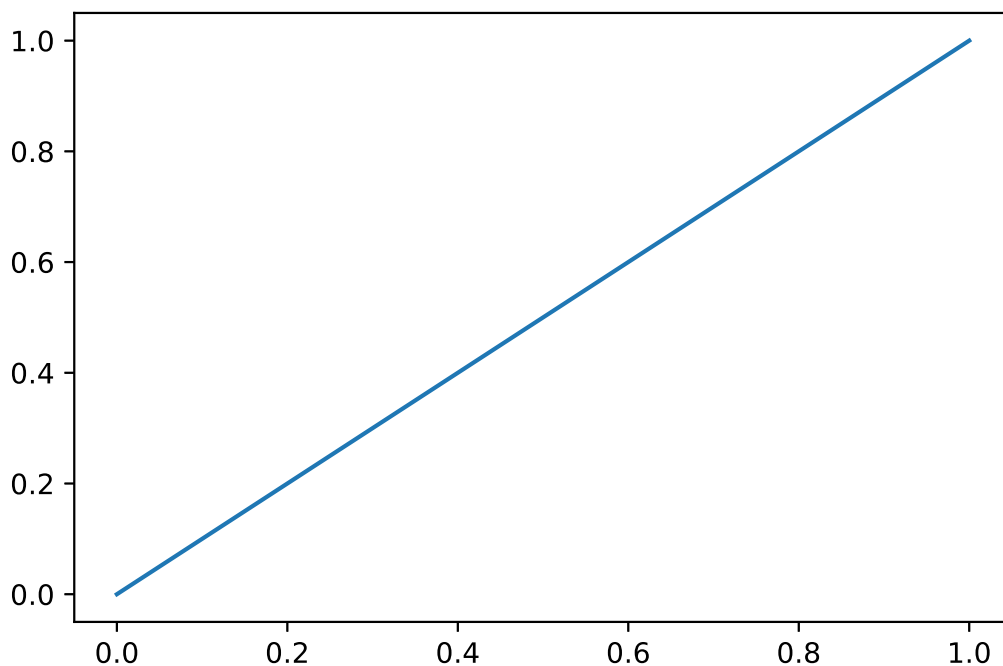


图 1 全国煤炭企业前 50 强产煤总量

### 2.2 插入表格

一般来插入的表格都可以通过 `python` 的 `DataFrame.to_latex()` 函数来导出到指定位置，然后通过 `input` 命令导入使用，经常会用到的导出模板为：

```
df.to_latex('./排版/tex代码/cr8.tex',caption='（$CR_8$）计算结果',label='table',header=False)
```

只需微调函数的参数即可，注意如下：

- 列名如果有数学公式的话导出会加转义符，这样在 latex 编译后不能正常显示。可以设置`escape=False`解决这个问题，不过这样设置数值中的转义符号也会被取消掉，这点在对于数值列中含有”%”的数据中需要注意
- 不想显示 index 列或者列名列可以通过调节`index`和`header`参数进行修改

插入表格的情况，相对复杂，可能出现的问题包括引入的表格过宽和过长的的问题。一个简单的例子如表1:

表 1 A simple Table

A	B
1	a
2	b
3	c
4	d

### 2.2.1 宽表格

针对插入表格过宽的情况

1. 一般选择使用`graphicx`宏包提供的`\resizebox{h-length%l}{v-length%l}{text}`命令来解决。这里给一个示例，语法如表2所示（代码在`tex`代码文件夹的`widetable`里边。

表 2 近十年行业集中率 ( $CR_8$ ) 计算结果

年份	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
行业集中率 ( $CR_8$ )	0.288718	0.301209	0.316948	0.331046	0.360429	0.359104	0.346434	0.40707	0.404583	0.477353	0.49315

2. 除了缩小表格内容以外，对于宽度溢出不太严重的表格，也可以选择修改水平排版开始位置，这也能使得本来溢出的表格可以被放下，而且也不会导致字体看起来太小（示例如表3所示）

表 3 近十年行业集中率 ( $CR_8$ ) 计算结果

年份	2011	2012	2013	2014	2015	2016	2017	2018
行业集中率 ( $CR_8$ )	0.288718	0.301209	0.316948	0.331046	0.360429	0.359104	0.346434	0.40707

## 2.2.2 长表格

长表格主要使用的是longtable宏包的longtable环境，与tabular类似，不过多了一些处理跨页操作的命令。因为to\_latex函数可以直接导出这种样式的表格，所以在实际中使用这种表格的频率也比较高，想要导出这种环境的表格时，只需在开始提到的模板中加入longtable=True的参数即可获得，使用比较方便<sup>1</sup>。

表 4 longtable 环境中的命令汇总

环境的水平对齐可选项	
留空	表格居中 <sup>2</sup>
[c]	表格居中
[l]	表格左对齐
[r]	表格右对齐
结束表格一行的命令	
\\	普通的结束一行表格
\\[< 距离 >]	结束一行，并增加额外间距
\\*	结束一行，禁止在此分页
\\kill	当前行不输出，只参与宽度计算
\\endhead	此命令以上部分是每页的表头
\\endfirsthead	此命令以上部分是表格第一页的表头
\\endfoot	此命令以上部分是每页的表尾
\\endlastfoot	此命令以上部分是表格最后一页的表尾
标题命令	
\\caption{< 标题 >}	生成带编号的表格标题
\\caption*{< 标题 >}	生成不带编号的表格标题
分页控制	
\\newpage	强制分页
\\pagebreak[< 程度 >]	允许分页的程度（0-4）
\\nopagebreak[< 程度 >]	禁止分页的程度（0-4）
脚注控制	
\\footnote	使用脚注 <sup>3</sup> ，注意不能用在表格头尾
\\footnotemark	单独产生脚注编号，不能用在表格头尾

接下一页表格……

<sup>1</sup>longtable 的表格宽度是根据表格内容调整的，当表格属于长而窄的类型时美观度比较有限，这时可以考虑利用 python 的 concat 函数对长表格进行拆分和列拼接

<sup>2</sup>实际上，留空的对齐方式是由一组命令控制的，参见宏包文档。

<sup>3</sup>普通表格中不能用。

(续表)

<code>\footnotetext</code>	单独产生脚注文字
长度参数	
<code>\LTleft</code>	对齐方式留空时，表格左边的间距，默认为 <code>\fill</code>
<code>\LTright</code>	对齐方式留空时，表格右边的间距，默认为 <code>\fill</code>
<code>\LTpre</code>	表格上方间距，默认为 <code>\bigskipamount</code>
<code>\LTpost</code>	表格下方间距，默认为 <code>\bigskipamount</code>
<code>\LTcapwidth</code>	表格标题的宽度，默认为 4in

### 2.2.3 定宽表格

刘海洋老师的书里有写到，可以去翻翻。用的是`tabularx`环境。考虑到这种环境不能使用 Python 的`to_latex`直接导出，更推荐的一种做法是在 python 中将表格进行调整之后再导出。

## 三、插入题注

这是一个题注<sup>4</sup>。

## 四、导入参考文献

参考文献的导入主要是使用 zotero 先导出 bib 文件。引文格式可能需要根据不同的使用场景进行调整。

对于中文文献，可以使用`gbt7714`宏包来生成满足我国标准的参考文献格式：

```
\documentclass{ctexart}
\usepackage{gbt7714}
\bibliographystyle{gbt7714-numerical}
\begin{document}
\cite{...}
...
\bibliography{bibfile}
\end{document}
```

按照国标的规定，参考文献的标注体系分为“顺序编码制”和“著者-出版年制”。用户应在导言区调用宏包 `gbt7714`，并且使用 `\bibliographystyle` 命令选择参考文献表的

---

<sup>4</sup>使用比较简单，这里不再赘述

样式，比如：

```
\bibliographystyle{gbt7714-numerical} %顺序编码制
```

或者

```
\bibliographystyle{gbt7714-author-year} %著者-出版年制
```

此外还可以使用 2005 版的格式 gbt7714-2005-numerical 和 gbt7714-2005-author-year。另外几点注意：

- 可以通过`\citestyle{citation style}`命令来改变引用位置的显示格式，可选参数为 `super`, `numbers`, `author-year`, 在顺序编码制下默认引用格式为上角标（`sub`）。
- 同一位置引用多篇文献时,如果想要在引用位置自动对文献进行排序则需要将`sort&compress`传递给宏包的可选参数

```
\usepackage[sort&compress]{gbt7714}
```

注意该宏包不要和 `cite` 宏包一同引入，会起冲突。

$\text{\LaTeX}$  中更为常用的一种导入参考文献的格式，其中`unsrt`样式表示参考文献条目按照文章中出现顺序排列。

```
\bibliographystyle{unsrt}
```

```
\bibliography{ref}
```

该方法下在引用位置处显示是类似“[4]”的形式。

## 附录 A 插入代码

```
data<-read.csv('./data.csv')
names(data)
# model<-lm(log(单价)~.-所属区-朝向-小区,data)
shapiro.test(log(data$单价))
shapiro.test(data$单价)
model <- lm(log(单价) ~ ., data)
summary(model)
sel_model <- step(model)
summary(sel_model)
# 多重共线性
library(car)
library(lmtest)
resettest(sel_model, power = 2)
library(lmtest)
bptest(sel_model)
bptest(model)
library("ivreg")
model <- ivreg(log(单价) ~ 容积率 + 总楼栋数 + 房龄 + 客厅数 + 卫生间数 + 总楼层+绿化率 +
  小区参考均价 | 容积率 + 总楼栋数 + 房龄 + 客厅数 + 卫生间数 + 总楼层+绿化率 + 交易价值 +
  居住品质, data = data)
summary(model)
```

### 1.1 input 方法

更为简洁的一种导入方法

```
print("hello Python")
```