

# UglyRep: 一个“丑陋”的 $\text{\LaTeX}$ 报告模板

GitHubonline1396529

March 23, 2025

## 摘 要

自从Elegant $\LaTeX$ 项目停更之后,我就时常感到十分的无措,尽管大家都说Elegant $\LaTeX$ 的历史使命已经完成、已经过时了,不在适合继续维护,但是我原本很喜欢这个项目。系列模板使用起来也特别方便,尤其是可以通过在 Markdown 文件的 YAML Header 中使用 `documentclass` 指定文档类,再通过 Pandoc 一次性转换为 PDF via  $\LaTeX$  快速排版。

最初,为了满足我个人的使用需求,我自己搓了几个模板。后来觉得比较好用,我就觉得应该发出来跟大家分享,大家一起用。但是因为我的技术比较菜,而且没有什么艺术细胞,做不到 Elegant,所以我把项目命名为了 Ugly $\LaTeX$ , 很合理吧。

本文是 UglyRep 模板的排版效果示例及模板文档,在展示排版效果的同时简要阐述了模板的部分功能及其使用方法。UglyRep 模板对标的是Elegant $\LaTeX$ 项目中的ElegantBook,实际上却是基于 `report` 文档类构建的。这样的设计是出于如下的两个原因考虑:

1. **更好的 Pandoc 兼容性:** Pandoc 不支持在 Markdown 文件的 YAML Header 中设置 Book 文档类的 Preface, 但是可以设置 Report 文档类的 Abstract。
2. **从更强的实用性的角度出发:** 在日常生活中实际上编写书籍的机会更少,而编写带封面的 Report 的机会更多。

**关键词:**  $\LaTeX$ ; 排版; 文档类;

## 目 录

<b>1</b>	<b>模板须知</b>	<b>2</b>
1.1	模板介绍	·2
1.2	守正创新	·2
<b>2</b>	<b>文档类功能及使用方法</b>	<b>3</b>
2.1	文档类选项	·3
2.2	文档类命令	·3
2.3	定理类环境	·4
2.4	语言模式	·4
2.5	颜色模式	·4
2.6	文献引用	·5
2.7	代码块	·5
2.7.1	listings 宏包的使用	·6
2.7.2	minted 宏包的使用	·7
2.8	插入图片	·7
2.9	Pandoc Markdown 结合使用	·7
<b>3</b>	<b>写作示例</b>	<b>8</b>
3.1	来自 Elegant $\text{\LaTeX}$ 的写作示例	·8
3.2	更多写作示例	·9
	<b>附录</b>	<b>12</b>
<b>A</b>	<b>项目使用的宏包</b>	<b>12</b>
<b>B</b>	<b>插入代码块效果示例</b>	<b>12</b>

## 一、模板须知

### 1.1 模板介绍

自从Elegant $\LaTeX$ 项目停更之后,我就时常感到十分的无措,尽管大家都说Elegant $\LaTeX$ 的历史使命已经完成、已经过时了,不再适合继续维护,但是我原本很喜欢这个项目。系列模板使用起来也特别方便,尤其是可以通过在 Markdown 文件的 YAML Header 中使用 `documentclass` 指定文档类,再通过 Pandoc 一次性转换为 PDF via  $\LaTeX$  快速排版。

最初,为了满足我个人的使用需求,我自己搓了这几个模板。后来觉得比较好用,我就觉得应该发出来跟大家分享,大家一起用。但是因为我的技术比较菜,而且没有什么艺术细胞,做不到 Elegant,所以我把项目命名为了 Ugly $\LaTeX$ , 很合理吧。

### 1.2 守正创新

本模板延用了Elegant $\LaTeX$ 的部分功能的实现。尽管Elegant $\LaTeX$ 的部分功能(比如复合颜色的主题、可选的 BIB 引用模式)还没有实现出来,但是后续会逐渐增加。目前最基本最关键的已经有了。包括

- 语言模式切换: 支持通过文档类选项 `lang=cn` 和 `lang=en` 切换中英文语言模式。
- 定理与公式环境: 支持数学公式编辑,并提供了 11 种不同的定理类环境的选项,支持交叉引用。
- 适配不同设备,包括适配手机或平板电脑尺寸的 Pad,适用于演示文稿的 Screen (幻灯片),适用于电子阅读器的 Kindle,适用于电脑屏幕尺寸的 PC,以及默认的通用 (A4 纸张);
- 全局字体大小支持: 从 8pt 到 20pt 的自由变换;
- 原有的 6 套颜色主题: `blue` (默认)、`green`、`cyan`、`sakura` 和 `black`、`brown`;

除此之外,本项目还在Elegant $\LaTeX$ 的基础之上增加了一系列新的优势性功能,包括但不限于

- 新增三种排版: 小开本 (32 开 A5), 课本 (B5 纸张), 以及紧凑模式的布局 (在 A4 纸张上使用 2cm 页边距);
- 更现代化的目录结构: 模块化功能便于维护,支持使用 Makefile 安装到目录;
- Pandoc 兼容性: 从 Markdown 文件快速构建您的文档 PDF;
- 新的配色选项: 增加了新的配色方案如更深的蓝色`blue`<sup>1</sup>。
- 紧凑布局: 节约纸张,环境保护从我做起的思想觉悟。
- 风格更正式的排版: 我去除了  $\LaTeX$  本来的古板的学术画风,但我可以保留了一部分。因为只有保留一部分,你才能知道你用的是  $\LaTeX$  排版。

<sup>1</sup>Elegant $\LaTeX$ 项目原有的几种配色方案中存在颜色太浅、阅读体验不够理想的问题。引进新的配色有助于解决这种问题。详情参见章节2.1和章节2.5。

**评论** 原有的ElegantL<sup>A</sup>T<sub>E</sub>X经常被认为是将各种功能模块写得太死了，想要在排版的时候进行进一步的个性化格式修改就会很困难。这个问题在本项目中非但有之，而且更甚。本项目的理念就是要确保在排版的过程中不需要引入任何额外的样式修改，直接提供足够理想的终产物样式，并适配尽可能广的应用场景。

## 二、文档类功能及使用方法

### 2.1 文档类选项

此模板基于 L<sup>A</sup>T<sub>E</sub>X 的标准文档类 report 设计，所以 report 文类的选项也能传递给本模板，比如 a4paper, 10pt 等等。

除此之外，本文档类提供了如下的几个额外的文档类选项：

- color, 用于指定文档的配色，目前可选项有默认的 black 以及：
  1. ElegantL<sup>A</sup>T<sub>E</sub>X项目原有的配色系列，为了与新的配色加以区分，均冠以了 elegant 的前缀,可选选项包括 elegantblue、elegantbrown、elegantcyan、elegantgreen、elegantsakura;
  2. 新的增强型配色 blue、green、olive (代替了 brown)、cyan 和 crimson (代替了 sakura)。
- device, 用于控制纸张大小, 可选的选项包括默认的 normal, 以及 pad、pd、kindle、compact、screen、booklet、textbook。
- fontsize, 用于指定字体大小, 默认为 11pt, 选项包括 8pt, 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 和 20pt。

### 2.2 文档类命令

本模板定义了如下的额外的功能性命令：

1. \fig{< 图题 >}{< 图片文件名 >}{<label 内容 >}：用于快速方便地插入图片。
2. \subtitle{}：用于为文档添加一个副标题是。
3. \maketitlewithonecolabstract{< 摘要内容 >}：在使用双列模式的情况下，同时实现插入摘要文本和生成文档标题。
4. \shserifbold：使用思源宋体粗体，通常用在文档的大标题上——如果你不喜欢黑体作为文档总标题的话。

**评论** 使用 twocolumn 参数的情况下，文档的大标题会被默认安置在文档的一侧。如果您希望大标题横跨两栏，就必须使用\twocolumn 命令的参数栏将\maketitle 命令和onecolabstract 环境一起包裹起来：

```

\twocolumn[
  \maketitle
  \begin{onecolabstract}
    摘要内容……摘要内容……摘要内容……
  \end{onecolabstract}
]

```

这将会十分不方便。因此，为了解决这个问题，我们定义了专门的 `\maketitlewithonecolabstract` 命令，来实现 two column abstract 一步处理。

## 2.3 定理类环境

此模板采用了 `amsthm` 中的定理样式，使用了 4 类定理样式，所包含的环境分别为

- 定理类: `theorem`, `lemma`, `proposition`, `corollary`;
- 定义类: `definition`, `conjecture`, `example`;
- 备注类: `remark`, `note`, `case`;
- 证明类: `proof`。

**注** 在选用 `lang=cn` 时，定理类环境的引导词全部会改为中文。

## 2.4 语言模式

本模板内含两套语言环境，改变语言环境会改变图表标题的引导词 (图, 表)，文章结构词 (比如目录, 参考文献等)，以及定理环境中的引导词 (比如定理, 引理等)。不同语言模式的启用如下<sup>1</sup>:

```

\documentclass[cn]{uglynote}
\documentclass[lang=cn]{uglynote}
\documentclass[en]{uglynote}
\documentclass[lang=en]{uglynote}

```

**注** 无论是中文模式还是英文模式，都可以正常输入中文文本，而且全都使用  $\text{\LaTeX}$  编译。

## 2.5 颜色模式

本模板在颜色模式方面的改进，主要是针对  $\text{\LaTeX}$  中的各种颜色的问题进行了增强。例如原有的 `sakura` 和 `blue` 颜色方案由于颜色深度不足，在白色纸张上辨读效果不好。再比如，原有的 `brown` 因为颜色太深，导致无法和黑色字体区分开来。

表 2.1 展示了经过增强前后的颜色的展示效果对比。可以看出，本项目为了增强文本阅读特性，对每个颜色在 RGB 0 到 255 的色号之间都进行了均衡，相较以往的  $\text{\LaTeX}$  的配色方案，辨读效果更理想。

<sup>1</sup>这里以 `uglynote` 示例，这些设定也适用于其他两个文档类 `UglyPaper` 和 `UglyRep`

表 2.1 Elegant $\text{\LaTeX}$ 与本模板颜色对照

旧颜色	旧色号	对标颜色	新色号
black	0,0,0	black	0,0,0
blue	1,126,218	blue	0,91,150
green	0,120,2	green	11,102,35
cyan	0,175,152	cyan	0,128,128
sakura	255,183,197	crimson	184,15,10
brown	109,62,18	olive	128,128,0

## 2.6 文献引用

参考文献部分,本模板调用了 biblatex 宏包,并使用了 biber,采用国标 GB7714-2015。

关于文献条目 (bib item), 你可以在谷歌学术, Mendeley, Endnote 中取, 然后把它们添加到 reference.bib 中。可以使用 \addbibresource 在文档类开头引入 BIB 文件。在文中引用的时候, 引用它们的键值 (bib key) 即可。参考文献示例:<sup>[1-3]</sup>使用了中国一个大型的 P2P 平台 (人人贷) 的数据来检验男性投资者和女性投资者在投资表现上是否有显著差异。

在文件末尾使用如下的代码可以打印参考文献列表。

```
\printbibliography[title=\ebibName]
```

**注** 打印参考文献列表的呈现效果中包含“参考文献”字样, 无需使用 \section{参考文献}, 或者, 使用 UglyRep 文档类, 也不需要使用 \chapter{参考文献}。

## 2.7 代码块

在本文档类中插入代码块有两种方式: 通过 listing 宏包插入代码块, 或者通过 minted 宏包插入代码块。其中, minted 宏包需要在文档类开头使用 \usepackage{}, 而 listing 宏包已经预先声明。

**评论** 之所以默认导入和预设 listing 而不是 minted, 是出于以下的三个原因作出的考虑:

1. minted 宏包在使用的时候需要预先配置过 Python 2.6 或者以上版本的运行环境, 部分用户可能不具备相应的条件。
2. 使用 minted 宏包的在编译的时候, 要额外加上 -shell-escape 参数。如果用户在使用任何预先定义过编译规则的 IDE 环境, 就有可能出现不兼容的情况, 而且这种需要查文档才会了解的额外的参数指定对新手用户也不是很友好。
3. 如果用户使用的是 Pandoc Markdown to PDF via  $\text{\LaTeX}$ , 那么 Pandoc 会在编译含代码块的 Markdown 文档时正确地自动引用 minted 宏包, 无需用户做任何额外的设置。

接下来将会详细介绍两个宏包各自的使用方法。

### 2.7.1 listings 宏包的使用

对于 listing 宏包的代码有两种展示方式：直接在文档中插入代码块，或者从代码文件输入。前者适用于较短的代码文件的展示，后者则更适合长代码文件的展示 (例如放在附录中长代码)。listing 宏包的高亮模式已经在文档类中设置好。一个简单的 listing 代码块如下所示，使用 language 参数可以指定语言，使用 caption 参数可以设置标题。

```
\begin{lstlisting}[
  numbers=left, % 启用行号
  language=C,
  caption={
    一个简单的 Hello World 程序
    \label{lst:insert}
  }
]
#include <stdio.h>

int main( void ) {
    /* Print 'hello, world' message. */
    print("hello, world\n");
    return 0;
}
\end{lstlisting}
```

**注** 如果想要在文章的任意位置引用这段代码，可以像上面这样为标题增加一个 label。但是要注意，\label{} 要紧跟在 caption 文本的后面。

上述代码的实际演示效果如代码块2.1所示。

**代码 2.1 一个简单的 Hello World 程序**

---

```
1 #include <stdio.h>
2
3 int main( void ) {
4     /* Print 'hello, world' message. */
5     print("hello, world\n");
6     return 0;
7 }
8 }
```

---



**注** 如果你想要在双列模式下插入代码块，你可以尝试将代码块置于一个 `figure*` 环境中，这将会创建一个横跨两列的代码块。

### 2.7.2 minted 宏包的使用

与 `listing` 相比，`minted` 的使用要简单一些。由于文档类没有预先导入 `minted` 宏包，在使用之前你需要在文档的导言区使用：

```
\usepackage{minted}
```

不同于 `listing` 宏包，`minted` 宏包本身并不具备为文本增添 `caption` 的功能，但仍有相应的方法能够手动添加 `caption` 并进行交叉引用。这些内容与本文档类不相关，请自行查证有关的资料。此不赘述。其他有关的用法，请参考 `minted` 宏包的手册。

## 2.8 插入图片

考虑到以往的 `subfigure` 宏包已不再维护，模板将默认使用更加现代的 `subcaption` 宏包。图2.1展示了一个基本的分图示例，其中具体的分图如分图2.1(a)和分图2.1(b)所示。

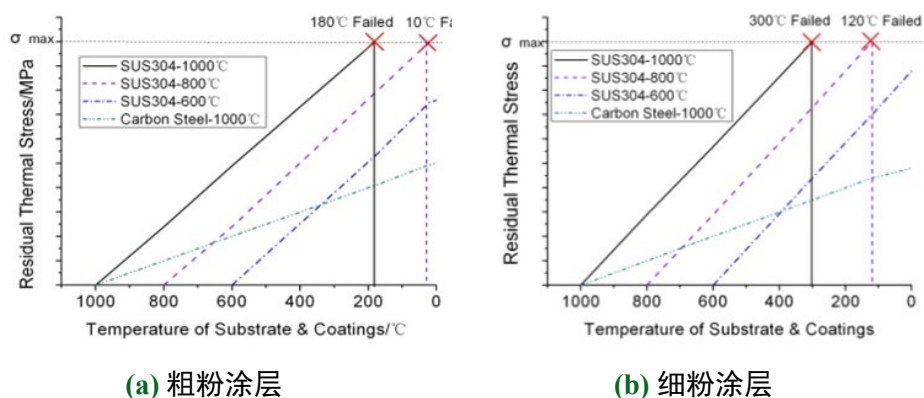


图 2.1 涂层在冷却过程中残余热应力的变化情况

## 2.9 Pandoc Markdown 结合使用

本模板类的一大优势是可用极其方便地与 Pandoc Markdown 相结合，从而将 Markdown 文件方便地转化为排版好的 PDF。你可以在 Markdown 文件开头处的 YAML Header 作如下指定，从而使用 `uglynote` 文档类，传递参数 `12pt`, `blue`：

```
documentclass: uglynote
classoptions: 12pt, blue
```

通过 `title`、`subtitle`、`author`、`date` 可以设置文档的大标题、副标题、作者名字和日期。这些参数被称为 Markdown 文档的“元数据”。

**title:** 文档标题  
**subtitle:** 文档副标题  
**author:** 作者的名字  
**date:** 2025 年 3 月 11 日

**注** 上述的这些参数在指定的时候, 如果要转化为 PDF via  $\text{\LaTeX}$  而不转化为其他文件格式, 则可以混合使用  $\text{\LaTeX}$  语法, 例如:

```

author: |
  作者的名字\thanks{
    XX 研究机构,
    Email: xxx@xxxmail.com
  }
date: \zhtoday
  
```

关于 Pandoc 的更多用法, 另请参阅 [Pandoc User's Guide](#)。

## 三、写作示例

### 3.1 来自Elegant $\text{\LaTeX}$ 的写作示例

我们将通过三个步骤定义可测函数的积分。首先定义非负简单函数的积分。以下设  $E$  是  $\mathcal{R}^n$  中的可测集。

**定义 3.1.1 (可积性)** 设  $f(x) = \sum_{i=1}^k a_i \chi_{A_i}(x)$  是  $E$  上的非负简单函数, 其中  $\{A_1, A_2, \dots, A_k\}$  是  $E$  上的一个可测分割,  $a_1, a_2, \dots, a_k$  是非负实数。定义  $f$  在  $E$  上的积分为 1.3

$$\int_E f dx = \sum_{i=1}^k a_i m(A_i). \quad (3.1)$$

一般情况下  $0 \leq \int_E f dx \leq \infty$ 。若  $\int_E f dx < \infty$ , 则称  $f$  在  $E$  上可积。

一个自然的问题是, Lebesgue 积分与我们所熟悉的 Riemann 积分有什么联系和区别? 之后我们将详细讨论 Riemann 积分与 Lebesgue 积分的关系。这里只看一个简单的例子。设  $D(x)$  是区间  $[0, 1]$  上的 Dirichlet 函数。即  $D(x) = \chi_{Q_0}(x)$ , 其中  $Q_0$  表示  $[0, 1]$  中的有理数的全体。根据非负简单函数积分的定义,  $D(x)$  在  $[0, 1]$  上的 Lebesgue 积分为

$$\int_0^1 D(x) dx = \int_0^1 \chi_{Q_0}(x) dx = m(Q_0) = 0 \quad (3.2)$$

即  $D(x)$  在  $[0, 1]$  上是 Lebesgue 可积的并且积分值为零。但  $D(x)$  在  $[0, 1]$  上不是 Riemann 可积的。

表 3.1 燃油效率与汽车价格

	(1)	(2)
燃油效率	-238.90*** (53.08)	-49.51 (86.16)
汽车重量		1.75*** (0.641)
常数项	11253.00*** (1171.00)	1946.00 (3597.00)
观测数	74	74
$R^2$	0.220	0.293

**定理 3.1.1 (Fubini 定理)** 若  $f(x, y)$  是  $\mathcal{R}^p \times \mathcal{R}^q$  上的非负可测函数, 则对几乎处处的  $x \in \mathcal{R}^p$ ,  $f(x, y)$  作为  $y$  的函数是  $\mathcal{R}^q$  上的非负可测函数,  $g(x) = \int_{\mathcal{R}^q} f(x, y) dy$  是  $\mathcal{R}^p$  上的非负可测函数。并且

$$\int_{\mathcal{R}^p \times \mathcal{R}^q} f(x, y) dx dy = \int_{\mathcal{R}^p} \left( \int_{\mathcal{R}^q} f(x, y) dy \right) dx. \quad (3.3)$$

**证明.** Let  $z$  be some element of  $xH \cap yH$ . Then  $z = xa$  for some  $a \in H$ , and  $z = yb$  for some  $b \in H$ . If  $h$  is any element of  $H$  then  $ah \in H$  and  $a^{-1}h \in H$ , since  $H$  is a subgroup of  $G$ . But  $zh = x(ah)$  and  $xh = z(a^{-1}h)$  for all  $h \in H$ . Therefore  $zH \subset xH$  and  $xH \subset zH$ , and thus  $xH = zH$ . Similarly  $yH = zH$ , and thus  $xH = yH$ , as required.

回归分析 (regression analysis) 是确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法。根据定理 3.1.1, 其运用十分广泛, 回归分析按照涉及的变量的多少, 分为一元回归和多元回归分析; 按照因变量的多少, 可分为简单回归分析和多重回归分析; 按照自变量和因变量之间的关系类型, 可分为线性回归分析和非线性回归分析。

## 3.2 更多写作示例

**例 3.2.1** 在使用 Miller-Tucker-Zemlin 约束作为 SEC 的情况下, 一个典型的 TSP 模型的最终表述形式为:

$$\begin{aligned} \min f &= \sum_{i=1}^n \sum_{j=1}^n d_{i,j} \times x_{i,j}, & \forall (i, j) \in E, i \neq j \\ \sum_{i=1}^n x_{ij} &= 1, & \forall i \in V \\ \sum_{j=1}^n x_{ji} &= 1, & \forall j \in V \\ u_i - u_j + n \cdot x_{ij} &\leq n - 1, & \forall (i, j) \in E, i \neq j \neq 1 \end{aligned} \quad (3.4)$$

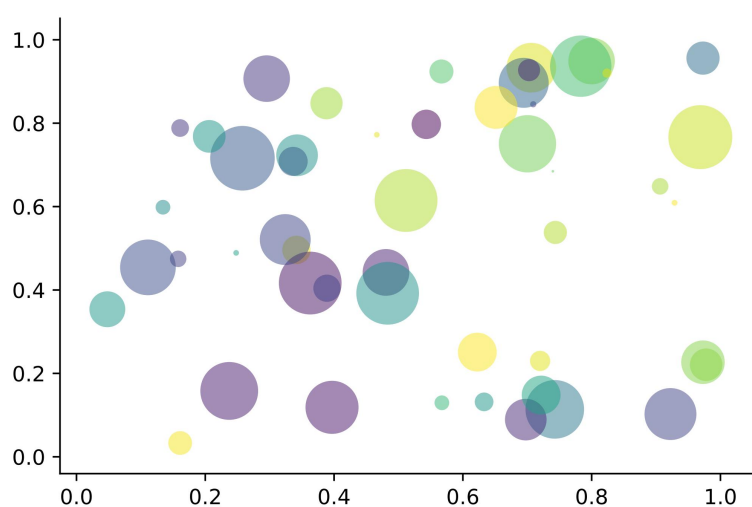


图 3.1 散点图示例  $\hat{y} = a + bx$

## 参考文献

- [1] 方军雄. 所有制、制度环境与信贷资金配置[J]. 经济研究, 2007(12): 82-92.
- [2] CARLSTROM C T, FUERST T S. Agency Costs, Net Worth, and Business Fluctuations: A Computable General Equilibrium Analysis[J]. The American Economic Review, 1997: 893-910.
- [3] LI Q, CHEN L, ZENG Y. The Mechanism and Effectiveness of Credit Scoring of P2P Lending Platform: Evidence from Renrendai.com[J]. China Finance Review International, 2018, 8(3): 256-274.

## 附录

## A. 项目使用的宏包

本文档类用到的所有宏包如下表1.1所示。在使用本文档类之前，请您先确保这些宏包已经被正确配置。

表 1.1 本文档类所使用到的宏包

1	2	3	4
ifxetex	kvoptions	etoolbox	calc
ctex	titling	titlesec	array
hologo	geometry	fontspec	float
authblk	amsmath	amssymb	amsthm
mathtools	mathrsfs	cancel	tocloft
titletoc	hyperref	fancyhdr	enumerate
enumitem	metalogo	setspace	caption
subcaption	appendix	graphicx	booktabs
multirow	longtable	pdfpages	braket
qcircuit	mhchem	chemfig	tikz
tikz-network	circuitikz	pdfpages	braket
multirow	longtable	pgfplots	color
xcolor	colortbl	xpatch	verbatim
matlab-prettifier	bookmark		

## B. 插入代码块效果示例

代码 B.1 系统聚类方法的实现 (Python):

```

1 """
2 demo.py - 示例代码
3 =====
4
5 一份示例代码，基于 :module:`scipy.cluster.hierarchy`
   模块实现了系统聚类，
6 用于测试代码块高亮渲染和排版效果。聚类的数据来源于随即数据。
7
8 使用方法：
9
10 1. 直接运行代码即可查看代码运行结果。
11 2. 使用高亮渲染工具查看代码文件。
12 """
13

```

```
14 # 导入必要的库
15 import numpy as np
16 import matplotlib.pyplot as plt
17
18 from sklearn.preprocessing import StandardScaler
19 from scipy.cluster.hierarchy import linkage, dendrogram
20
21 # 生成随机数据
22 np.random.seed(42)
23 data = np.random.rand(20, 2) * 100
24
25 # 标准化数据
26 scaler = StandardScaler()
27 data_normalized = scaler.fit_transform(data)
28
29 # 进行系统聚类
30 Z = linkage(data_normalized, method='ward')
31
32 # 绘制树状图
33 plt.figure(figsize=(10, 5))
34 dendrogram(
35     Z,
36     leaf_rotation=90,
37     leaf_font_size=10
38 )
39 plt.title('Hierarchical Clustering Dendrogram')
40 plt.xlabel('Sample index')
41 plt.ylabel('Distance')
42 plt.grid(True)
43 plt.tight_layout()
44 plt.show()
```

---