



Quite-Elegant-Typ: 简单的书籍模板

ElegantBook 的 Typst 复刻

作者: 编译型战狼

时间: 2025-03-25

版本: 0.1.0

自定义: 信息

目录

第 1 章 模板介绍	1
1.1 模板说明	1
1.2 模板下载	1
1.3 模板使用	1
1.4 注意事项	1
1.5 待办事项	1
1.6 其他	1
第 2 章 字体设置	3
2.1 需要的字体	3
2.2 具体细节	3
第 3 章 模板设置说明	4
3.1 颜色主题	4
3.1.1 颜色主题设置	4
3.1.2 具体配置	4
3.2 封面设置	4
3.2.1 默认封面设置	4
3.2.2 自定义封面	5
3.3 目录选项	5
3.4 数学环境简介	5
3.4.1 定理类环境	6
3.4.2 示例类环境	6
3.4.3 结论类环境	6
3.4.4 颜色设置	7
3.4.5 新增数学环境	7
3.5 列表环境	8
3.6 数学公式编号	8
3.7 添加序章	8
3.8 语言模式	8
第 4 章 写作示例	9
4.1 Lebesgue 积分	9
4.1.1 积分的定义	9

第 1 章 模板介绍

1.1 模板说明

本模板旨在用 Typst 对 **ElegantBook** 进行复刻.

1.2 模板下载

模板发布在 <https://github.com/a31474/quite-elegant-typ>

1.3 模板使用

下载后, 把模板放在本地文件夹下. 然后导入 `lib.typ` 即可使用

```
1 #import "lib.typ": *
2
3 #show: conf
4 #default-cover()
5 #default-outline()
```

注意把 `lib.typ` 改成其对应的本地路径

其中 `#default-cover()` 和 `#default-outline()` 分别添加了封面和目录. 若不需要, 可以不用加上.

封面和目录设置请参考 [3.2 封面设置](#) 和 [3.3 目录选项](#)

1.4 注意事项

本模板字体请一定按 [第 2 章 字体设置](#) 配置好, 本模板依据特定字体进行编写. 没有考虑到使用其他字体的情况.

1.5 待办事项

本模板仍未完善, 未能完成对 **ElegantBook** 的完全复刻. 还有以下事项需要解决

1. 章节摘要
2. 章后习题
3. 参考文献

1.6 其他

使用模板时, 若需要添加 `show`, `set` 规则时. 注意把 `show` 规则添加到本文档设置 `#show: conf` 前, 以免与模板设置发生冲突. 而可以把 `set` 规则添加到 `#show: conf` 后, 以覆盖模板的配置.

如本文档对代码块进行的 `show`, `set` 设置

```
1 #show raw.where(block: true): it => {
2   set par(justify: false)
3   block(
4     fill: luma(245),
5     inset: (top: 4pt, bottom: 4pt),
6     radius: 4pt,
7     width: 100%,
8     stack(
9       ..it.lines.map(raw_line => block(
```

```
10     inset: 3pt,  
11     width: 100%,  
12     grid(  
13         columns: (1em + 4pt, 1fr),  
14         align: (right + horizon, left),  
15         column-gutter: 0.7em,  
16         row-gutter: 0.6em,  
17         text(gray, [#raw_line.number]), raw_line,  
18     ),  
19 ),  
20 ),  
21 )  
22 }  
23  
24 #show: conf  
25 #default-cover(  
26     image: image("image/cover.jpg", width: 100%),  
27     title: [Quite-Elegant-Typ: 简单的书籍模板],  
28     subtitle: [ElegantBook 的 Typst 复刻],  
29 )  
30 #default-outline()  
31  
32 #set raw(lang: "typst")  
33
```

第 2 章 字体设置

2.1 需要的字体

本模板需要以下字体

- Times New Roman
- 方正书宋
- 方正黑体
- 方正楷体
- 方正仿宋

其中方正的方正书宋、方正黑体、方正楷体、方正仿宋 四款字体均可免费试用，且可用于商业用途。

2.2 具体细节

本模板的详细配置如下

```
1 #set text(font: ("Times New Roman", "FZShuSong-Z01S"))
2
3 #show emph: set text(font: ("Times New Roman", "FZKai-Z03S"))
4
5 #show strong: set text(font: ("Times New Roman", "FZHei-B01S"))
6 #show text.where(weight: "bold"): set text(font: ("Times New Roman", "FZHei-B01S"))
```

选用方正书宋作为正文字体

选用方正黑体作为加粗字体

选用方正楷体作为强调字体

第 3 章 模板设置说明

3.1 颜色主题

3.1.1 颜色主题设置

本模板内置 5 种颜色主题，分别为 **green**、**cyan**、**blue**（默认）、**gray**、**black**。

调用其他颜色主题的方法为

```
1 #show: it => conf(it, color-theme: "green")
```

或

```
1 #show: conf.with(color-theme: "green")
```





















也可以自定义颜色主题, 如

```
1 #show: it => conf(  
2   it,  
3   color-theme: (  
4     structure: rgb(255, 0, 0),  
5     main: rgb(0, 255, 0),  
6     second: rgb(0, 0, 255),  
7     third: rgb(255, 255, 0),  
8   ),  
9 )
```

需传入一字典, 字典的 keys 必须分别为 **structure**, **main**, **second**, **third**. 各自的 value 为对应颜色.

3.1.2 具体配置

下表为内置颜色主题的具体配置

	green	cyan	blue	gray	black	主要使用的环境
structure						chapter section subsection
main						definition exercise problem
second						theorem lemma corollary
third						proposition

3.2 封面设置

3.2.1 默认封面设置

封面 **default-cover** 提供了以下参数

cover 封面图片

rect-color 中间色块颜色

logo 徽标

title 标题

subtitle 副标题

author 作者,

institute 机构,

date 日期,

version 版本,

other 自定义元素,

extrainfo 自定义内容,

其中 **other** 需要传入一字典来, 显示自定义的元素.

而 **extrainfo** 可以传入 **content**

如下为本文档的封面

```
1 #default-cover(
2   cover: image("image/cover.jpg", width: 100%),
3   title: [Quite-Elegant-Typ: 简单的书籍模板],
4   subtitle: [ElegantBook 的 Typst 复刻],
5   author: [编译型战狼],
6   date: datetime.today().display(),
7   version: version(0, 1, 0),
8   other: (自定义: "信息"),
9 )
```

3.2.2 自定义封面

若不需要默认封面, 想自定义封面, 可以自己编写, 但要注意本模板添加了页眉, 页脚. 需要手动取消. 还要记得把页面计数重置.

例如

```
1 #show: conf
2 // 自定义封面
3 #{
4   page/footer: none, header: none [
5     #align(center+horizon)[#text(30pt)[封面]]
6   ]
7   // 重置页面计数
8   counter(page).update(1)
9 }
```

3.3 目录选项

目录的颜色主题不会随 **conf** 的设置变化, 需要手动设定

```
1 #default-outline(color-theme: "green")
```

3.4 数学环境简介

本模板定义了三类数学环境, 有以下这些

- 定理类环境
 - **definition**

- theorem、lemma、corollary、axiom、postulate
- proposition
- 示例类环境
 - example、problem、exercise
- 结论类环境
 - note
 - conclusion、assumption、property、remark、solution
 - proof

3.4.1 定理类环境

包括三种格式

`definition` 环境，颜色为 `main`；

`theorem`、`lemma`、`corollary`、`axiom`、`postulate` 环境，颜色为 `second`；

`proposition` 环境，颜色为 `third`。

并且提供参数 `number`，控制是否显示计数器。

```
1 #definition[ 测试][ $a \times b$ ]
2 #definition[ 测试][ $a \times b$ ]
3 #definition(number: false)[ (测试)][ $c \times d$ ]
```

定义 3-1 测试

$$a \times b \quad (3.1)$$

定义 (测试)

$$c \times d \quad (3.2)$$

定义 3-2 测试

$$a \times b \quad (3.3)$$

3.4.2 示例类环境

包括 `example`、`problem`、`exercise`。

并且提供参数 `number`，控制是否显示计数器。

```
1 #example() 测试
2
3 #example(number: false) 测试
4
5 #example() 测试
```

例 3-1 测试

例 测试

例 3-2 测试

3.4.3 结论类环境

包括 `note`、`conclusion`、`assumption`、`property`、`remark`、`solution`、`proof`


```
1 #note[测试]
2
3 #note[] 测试
```

笔记 测试

笔记 测试

被包含的内容字体会有变化

3.4.4 颜色设置

数学环境的颜色想要更改需要手动更改文件.

需要更改 math 文件夹下的 math.typ 中

```
1 // 颜色主题
2 #let color-themes = color-select("blue")
```

改为对应颜色主题, 如

```
1 #let color-themes = color-select("black")
```

或像 3.1.1 颜色主题设置, 传入一字典

```
1 #let color-themes = (
2   structure: rgb(255, 0, 0),
3   main: rgb(0, 255, 0),
4   second: rgb(0, 0, 255),
5   third: rgb(255, 255, 0),
6 )
```

3.4.5 新增数学环境

要新增数学环境可以使用 `#math-fun-def`, `#math-fun-exam`, `#math-fun-note`, 分别对应定理类环境, 示例类环境, 结论类环境.

可以参照以下进行编写

定理类

```
1 #let postulate(number: true, name, content) = math-fun-def(
2   main-color: color-themes.second,
3   type: "假设",
4   number: number,
5   name,
6   content,
7 )
```

示例类

```
1 #let example(number: true) = math-fun-exam(
2   main-color: color-themes.main,
3   number: number,
4   type: "例",
5 )
```

结论类

```
1 #let conclusion(body) = math-fun-note(
```

```

2  main-color: color-themes.third,
3  font: ("Times New Roman", "FZKai-Z03S"),
4  "结论",
5  body,
6 )

```

若新增的数学环境用到了计数器, 请向 `conf` 传入一 `array`, 内容为新增计数器的 `type`

```

1 #show: conf.with(new-math-fun: ("假设", "例"))

```

3.5 列表环境

如下为列表环境, 没有进行太多设置. 如有需要可以自行设定

- | | |
|---------|------------|
| • item1 | 1. item1 |
| • 项目 2 | 2. 项目 2 |
| ▸ item1 | (a). item1 |
| ▸ 项目 2 | (b). 项目 2 |
| – item1 | (c). item1 |
| – 项目 2 | (d). 项目 2 |

3.6 数学公式编号

数学公式默认编号到一级标题

$$a \times b \tag{3.4}$$

可以改为二级标题

```

1 #show: conf.with(eq-level: 2)

```

3.7 添加序章

可以像这样添加无编号标题

```

1 #heading(numbering: none)[序章]
2 #heading(numbering: none, level: 2)[小节]

```

3.8 语言模式

本文档只按照编写中文文档的目标进行配置, 比如设置了

```

1 #set text(lang: "zh", region: "cn", size: 10pt)

```

第 4 章 写作示例

4.1 Lebesgue 积分

在前面各章做了必要的准备后,本章开始介绍新的积分。在 Lebesgue 测度理论的基础上建立了 Lebesgue 积分,其被积函数和积分域更一般,可以对有界函数和无界函数统一处理。正是由于 Lebesgue 积分的这些特点,使得 Lebesgue 积分比 Riemann 积分具有在更一般条件下的极限定理和累次积分交换积分顺序的定理,这使得 Lebesgue 积分不仅在理论上更完善,而且在计算上更灵活有效。

Lebesgue 积分有几种不同的定义方式。我们将采用逐步定义非负简单函数,非负可测函数和一般可测函数积分的方式。

由于现代数学的许多分支如概率论、泛函分析、调和分析等常常用到一般空间上的测度与积分理论,在本章最后一节将介绍一般的测度空间上的积分。

4.1.1 积分的定义

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 4-1 (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 $\int_a^b f(x)$

$$\int_E f \, dx = \sum_{i=1}^k a_i m(A_i) \quad (4.1)$$

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? 在 4.4 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

$$\int_0^1 D(x) \, dx = \int_0^1 \chi_{Q_0}(x) \, dx = m(Q_0) = 0 \quad (4.2)$$

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。

有界变差函数是与单调函数有密切联系的一类函数。有界变差函数可以表示为两个单调递增函数之差。与单调函数一样,有界变差函数几乎处处可导。与单调函数不同,有界变差函数类对线性运算是封闭的,它们构成一线空间。

练习 4-1 设 $f \in L(\mathcal{R}^1)$, g 是 \mathcal{R}^1 上的有界可测函数。证明函数

$$I(t) = \int_{\mathcal{R}^1} f(x+t)g(x) \, dx \quad t \in \mathcal{R}^1 \quad (4.3)$$

是 \mathcal{R}^1 上的连续函数。

解 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。

证明 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。

定理 4-1 (Fubini 定理)

(1) 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

$$\int_{\mathcal{R}^p \times \mathcal{R}^q} f(x, y) dx dy = \int_{\mathcal{R}^p} \left(\int_{\mathcal{R}^q} f(x, y) dy \right) dx \quad (4.4)$$

(2) 若 $f(x, y)$ 是 $\mathcal{R}^p \times \mathcal{R}^q$ 上的可积函数, 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 上的可积函数, 并且 $g(x) = \int_{\mathcal{R}^q} f(x, y) dy$ 是 \mathcal{R}^p 上的可积函数。而且式 (4.4) 成立。

笔记 在本模板中, 引理 (lemma), 推论 (corollary) 的样式和定理 4-1 的样式一致, 包括颜色, 仅仅只有计数器的设置不一样。

命题 4-1 (最优性原理)

如果 u^* 在 $[s, T]$ 上为最优解, 则 u^* 在 $[s, T]$ 任意子区间都是最优解, 假设区间为 $[t_0, t_1]$ 的最优解为 u^* , 则 $u(t_0) = u^{\{*\}}(t_0)$, 即初始条件必须还是在 u^* 上。

我们知道最小二乘法可以用来处理一组数据, 可以从一组测定的数据中寻求变量之间的依赖关系, 这种函数关系称为经验公式。本课题将介绍最小二乘法的精确定义及如何寻求点与点之间近似成线性关系时的经验公式。假定实验测得变量之间的 n 个数据, 则在平面上, 可以得到 n 个点, 这种图形称为“散点图”, 从图中可以粗略看出这些点大致散落在某直线近旁, 我们认为其近似为一线性函数, 下面介绍求解步骤。

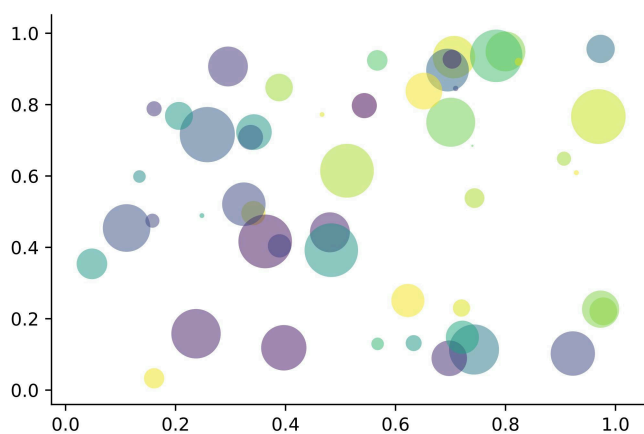


图 4.1 散点图示例 $\hat{y} = a + bx$

以最简单的一元线性模型来解释最小二乘法。什么是一元线性模型呢? 监督学习中, 如果预测的变量是离散的, 我们称其为分类 (如决策树, 支持向量机等), 如果预测的变量是连续的, 我们称其为回归。回归分析中, 如果只包括一个自变量和一个因变量, 且二者的关系可用一条直线近似表示, 这种回归分析称为一元线性回归分析。如果回归分析中包括两个或两个以上的自变量, 且因变量和自变量之间是线性关系, 则称为多元线性回归分析。对于二维空间线性是一条直线; 对于三维空间线性是一个平面, 对于多维空间线性是一个超平面。

性质 柯西列的性质

1. x_k 是柯西列, 则其子列 x_k^i 也是柯西列。
2. $x_k \in \mathcal{R}^n$, $\rho(x, y)$ 是欧几里得空间, 则柯西列收敛, (\mathcal{R}^n, ρ) 空间是完备的。

结论 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。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。