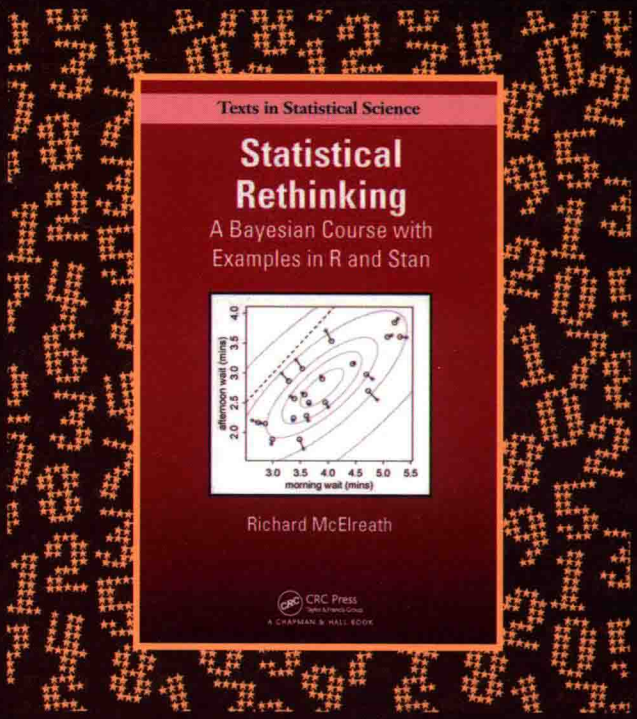
HZ BOOKS
华章IT
CRC Press
Taylor & Francis Group
数据科学与工程技术丛书

统计反思

用R和Stan例解贝叶斯方法

[美] 理查德·麦克尔里思(Richard McElreath) 著
林荟 译



STATISTICAL RETHINKING
A BAYESIAN COURSE WITH EXAMPLES IN R AND STAN

机械工业出版社
China Machine Press

数据科学与工程技术丛书

STATISTICAL RETHINKING
A BAYESIAN COURSE WITH EXAMPLES IN R AND STAN

统计反思
用R和Stan例解贝叶斯方法

[美] 理查德·麦克尔里思(Richard McElreath) 著
林荟 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

统计反思：用 R 和 Stan 例解贝叶斯方法 /（美）理查德·麦克尔里思（Richard McElreath）著；林荟译 . —北京：机械工业出版社，2019.4
（数据科学与工程技术丛书）
书名原文：Statistical Rethinking: A Bayesian Course with Examples in R and Stan

ISBN 978-7-111-62491-2

I. 统… II. ①理… ②林… III. 贝叶斯方法 – 应用 – 数理统计 IV. O212.8

中国版本图书馆 CIP 数据核字（2019）第 066643 号

本书版权登记号：图字 01-2016-8665

本书以 Stan 统计软件为基础，以 R 代码为例，提供了一个实际的统计推断的基础。从贝叶斯统计方法的角度出发，介绍了统计反思的相关知识，以及一些常用的进行类似权衡的工具，展示了两个完整的最常用的计数变量回归，介绍了应对常见的单一模型无法很好地拟合观测数据的排序分类模型与零膨胀和零增广模型，提出了基于贝叶斯概率和最大熵的广义线性分层模型以及处理空间和网络自相关的高斯过程模型。

本书适合统计、数学等相关专业的高年级本科生、研究生，以及数据挖掘的从业人士阅读。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）
责任编辑：赵 静 责任校对：李秋荣
印 刷：北京市兆成印刷有限责任公司 版 次：2019 年 4 月第 1 版第 1 次印刷
开 本：185mm × 260mm 1/16 印 张：26.25
书 号：ISBN 978-7-111-62491-2 定 价：139.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88379833 投稿热线：（010）88379604
购书热线：（010）68326294 读者信箱：hzjsj@hzbook.com

译 者 序

这是我希望在学习贝叶斯统计的时候首先阅读的一本书。与其说贝叶斯是一种不同的统计方法，不如说它是一种不一样的统计哲学，也是一种看待生活中很多问题的不同的视角。不是所有的应用数据科学领域都需要用到贝叶斯，但即使你所处的行业用得很少，了解贝叶斯的基本概念也是很有必要的。因为这种根据证据改变自己想法的思维方式能帮助我们约束直觉，这是一种高级的思维方式。

贝叶斯推断不外乎计算在某假设下事情可能发生的方式的数目。事情发生方式多的假设成立的可能性更高。一旦我们定义了假设，贝叶斯推断会强制施行一种通过已经观测到的信息进行纯逻辑的推理过程。频率法要求所有概率的定义都需要和可计数的事件以及它们在大样本中出现的频率联系起来。这使得频率学的不确定性依赖于想象的数据抽样的前提——如果多次重复测量，我们将会收集到一系列呈现某种模式的取值。这也意味着参数和模型不可能有概率分布，只有测量才有概率分布。这些测量的分布称为抽样分布。这些所谓的抽样只是假设，在很多情况下，这个假设很不合理。而贝叶斯方法将“随机性”视为信息的特质，这更符合我们感知的世界运转模式。所以，在很多应用场景中，贝叶斯也更加合适。

总体说来，本书有如下亮点：

1. 可重复。这点实在是太重要了。书中的数据很容易获取，书中的代码、建模过程都可以重复。读者可以在阅读的过程中实践代码，并且生成书中展示的结果。也可以自己修改代码，看看结果的变化，这对理解内容有极大的帮助。

2. 前 3 章中有我见过的对贝叶斯及哲学最清晰的讲解。对于那些只想知道贝叶斯模型是什么但不想花太多时间深入学习更加复杂的贝叶斯模型的读者，推荐仔细阅读前 3 章。第 1 章反思了流行的统计和科学哲学，指出我们不该仅使用各种自动化的工具，而应该学着在实际应用中建立、评估不同的模型。接下来的第 2 章和第 3 章介绍了贝叶斯推断和进行贝叶斯计算的基本工具。其中作者的讲解方式很绕、很慢，特别强调了概率理论的纯逻辑解释。但我希望读者能够耐心地认真阅读这 3 章，这对以深入理解贝叶斯为目标的人来说，一点儿也不啰嗦。

3. 本书提供了 R 包 rethinking 来实现模型，使用更加简单直接。更好的方法当然是直接学习使用 Stan。rethinking 中的一些函数（map 和 map2stan）对 stan 进行包装，隐藏了背后的 stan 代码，这使得一些错误信息让人难以理解。如果要在工作中应用书中介绍的模型，最好还是在之后花时间学习 Stan。好在读过本书之后，学习 Stan 应该不难。

4. rethinking 包中自带的数据以及一些绘图函数可极大地帮助读者对真实数据进行建模，并且通过可视化解释结果。在这些绘图的函数中，有些能直接对后验预测进行可视化，并通过这种方式比较模型和参数。对于简单的模型，可以通过参数估计总结表来理解模型。但只

要模型稍微复杂一点，尤其是含有交互效应（见第 7 章），解释后验分布就会变得很难。如果要在模型解释中考虑参数间的相关性，那可视化就不可或缺。

5. 书中的一些关于社会科学的例子不仅展示了如何建立模型，更重要的是展示了如何定义问题本身。社会学的问题往往是开放的，很复杂。所以通过数据建模解决这类问题的难点不仅仅是模型本身，还有将开放式问题转化成一个封闭式问题的过程。本书中有很多这样的例子，而且作者对数据所处的实际语境也进行了详细的解释。

为了使行文更加通顺，在翻译的过程中采用了较多的意译，有的地方加上了译者注以帮助读者理解。华章公司的编辑对本书的翻译工作给予了大力的支持和帮助。在此对所有为本书中文版问世做出努力的人表示感谢！由于译者水平有限，书中难免有错误和不妥之处，恳请读者批评指正。

林　荟

2018 年 12 月

前　言

石匠，开始动工之前（Masons，when they start upon a building），
总会小心测试鹰架（Are careful to test out the scaffolding）。

确保模板不会滑落在繁忙的街口（Make sure that planks won't slip at busy points），
牢牢钉好每把梯子，拴紧所有螺丝（Secure all ladders，tighten bolted joints）。

这一切付出在完工后都得被拆除（And yet all this comes down when the job's done），
展露结实的石墙（Showing off walls of sure and solid stone）。

所以，亲爱的，就算我们之间的桥梁（So if，my dear，there sometimes seem to be），
偶尔因为老旧看似即将倒塌（Old bridges breaking between you and me）。

别害怕。让那鹰架倒下吧（Never fear. We may let the scaffolds fall），
相信我们建造的墙坚不可摧（Confident that we have build out wall）。

（《鹰架》（Scaffolding），作者 Seamus Heaney，1939—2013）

本书意在帮助你增进统计模型的知识以及使用模型的信心。就像造墙时的鹰架，能够帮助你建造需要的石墙，虽然最终你要将鹰架拆除。因此，本书讲解的方式有些拐弯抹角，但那是为了促使你们亲自实践模型背后的每一个计算步骤，虽然真实建模的过程常常是自动的。这样小题大做是为了让你能够对方法背后的细节有足够的了解，以能够合理地选择和解释模型。虽然你最终会用一些工具自动建模，但刚开始放慢步伐、夯实基础是很重要的。耐心建立坚实的墙然后再拆去鹰架。

目标读者

本书主要面向自然和社会科学的研究人员，可以是新入学的博士生，也可以是有经验的专业人士。你需要有回归的基本知识，但不一定需要对统计模型驾轻就熟。如果你接受这样的事实：一些在 21 世纪早期广泛使用的典型统计学方法并非完全正确，其中大部分和 p 值以及令人迷惑的各种统计检验有关。如果你在一些杂志和书上读到过一些替代的方法，但不知道从何学习这些方法，那么本书就是为你而写的。

事实上，本书并不是要直接抨击 p 值和相关的方法。在我看来，问题并不在于人们习

惯用 p 值来解决科学界的各种问题，而在于人们忽略了许多其他有用的工具。因此，我假定本书的读者已经准备好不使用 p 值做统计推断。仅有这种心理准备还不够，最好能有一些文献资料帮助你探查与 p 值和传统统计检验有关的错误及误解。即使我们不用它们，也要对其有所了解。我因此查阅了一些相关的资料，但由于本书篇幅所限不能详细讨论，否则本书会太厚，也会打乱原本的教学节奏。

这里要提醒一点，反对 p 值不仅仅是贝叶斯学派的观点。事实上，显著性检验能够（其实也已经）构建为贝叶斯过程。其实真正促使人们避免使用显著性检验的是出于认识论的考虑，关于这一点我会在第 1 章简单讨论。

教学方法

本书使用更多的是程序代码而非数学公式。直到真正对算法付诸实践，即使最出色的数学家可能也无法理解该过程。因为用代码实践的过程去除了算法中所有模棱两可的地方。因此，如果一本书同时教你如何实践算法的话，学习起来会更轻松。

展示代码除了有利于教学也是必需的，因为许多统计模型现在都需要计算，纯数学的方法无论如何也不能解决问题。你在本书后面部分可以看到，同样的数理统计模型的实现方法可以有多种，而且我们有必要区分这些方法。当你在本书之外探索更高级或更有针对性的统计模型时，这里强调的编程计算知识将帮助你识别和应对各种实际困难。

本书的每一部分都只揭示了冰山一角。我丝毫没有涵盖所有相关内容的打算，而是试图将其中一些东西解释清楚。在此尝试中，我在数据分析的实例中穿插了许多模型概念和内容。例如，书中没有一个单元专门讲预测变量的中心化，但我在数据分析中使用并解释了这项技术。当然，不是所有读者都喜欢这样的讲解方式。但是我的很多学生喜欢这种讲解方式。我很怀疑这样的讲解能否对大部分要学习这些内容的读者起作用。从心底来说，这反映了我们在现实中是如何在自己的研究中学会这些方法的。

如何使用本书

这不是参考书，而是教科书。本书不是让你在遇到问题时用来查阅相关部分的，而是一个完整连贯的教学过程。这在教学上很有优势，但可能不符合很多科学家现实中的阅读习惯。

本书正文中有很多代码。这样做是因为在 21 世纪从事统计分析工作必须要会编程，或多或少会一些。编程不是候选技能，而是必备技能。在书中的很多地方，我宁可过多地展示代码，也不愿过少展示代码。根据我对编程新手的教学经验，当学生手上有可以运行的代码时，让他们在此基础上修改比让他们从 0 开始写程序效果更好。我们这代人可能是最后一代需要用命令的方式操作计算机的了，因此编程也越来越难教。我的学生非常熟悉计算机，但他们不知道计算机代码长什么样。

本书要求读者具备什么基础？ 本书的目的不是教读者关于编程的基本知识。我们假设读者已经知道 R 的基本安装和数据处理知识。在大多数情况下，入门级的 R 编程介绍便足够。据我所知，许多人觉得 Emmanuel Paradis 所著的《R for Beginners》很有帮助。你可以通过链接 http://cran.r-project.org/other-docs.html 找到该指南以及许多入门级教程。要顺利阅读本书，你得知道 $y<-7$ 指的是将 7 这个值赋给变量 y。你要知道后面紧跟括号的符号是函数。你要能够辨认出循环并且知道命令可以相互嵌套（递归）。除了使用循环，知道 R 可

以将很多代码矢量化也很重要。但是阅读本书不要求你精通 R 语言。

在书中你会不可避免地看到一些之前没有见过的代码。在书中我会尽量对一些特别重要或者不常见的编程技巧进行说明。事实上，本书花了大量篇幅解释代码。这么做的原因是学生确实需要这样的解释。除非能将每行命令同与之对应的方法目标联系起来，不然当代码无法运行时，学生无法判定错误原因。我在讲授数理进化论理论的时候也遇到过这样的问题——学生的代数知识比较薄弱，当他们答不上题时，通常不知道是因为数学上的小疏忽还是策略问题。书中对代码进行延伸介绍的目的在于帮助读者理解代码，之后在实践中能够自己发现并且修正错误。

代码使用。书中的代码都带有蓝底，相应的代码输出在其下以等宽字体展示。例如：

R code 0.1

```
print( "All models are wrong, but some are useful." )
```

```
[1] "All models are wrong, but some are useful."
```

在代码旁边有一个数字标识，你可以在本书的网站上寻找相应的代码文本文件。希望读者能够跟上教学进度，运行书中的代码，然后将输出结果和书中展示的结果进行比较。我非常希望你能够自己运行代码，因为和你不能光靠看李小龙的电影就学会功夫一样，你不能仅靠阅读一本书而不实践就学会编写统计模型程序。你得真正到格斗场上出拳，当然也可能挨拳。

如果你觉得困惑，记得你可以独立运行每行代码，检查过程中的每一步计算。这是你学习和解决问题的方式。例如，下面的代码用一种让人抓狂的方法计算 10 乘以 20：

R code 0.2

```
x <- 1:2
x <- x*10
x <- log(x)
x <- sum(x)
x <- exp(x)
x
```

```
200
```

如果你不理解某个特定的步骤，你可以随时查看那一步之后变量 x 的内容。你要用这种方式学习书中的代码。对于你自己写的代码，可以用这种方法找到代码中的错误并修复它们。

选读部分。本书中的选读部分真正阅读起来是这样的。书中有两类选读部分：1）再思考（Rethinking）；2）深入思考（Overthinking）。再思考部分看起来是这样的：

再思考：再想想。再思考部分意在提供更广泛的资料，略微提及当前介绍的方法和与其他方法的联系，提供一些背景材料，或者指出一些常见误解。这些文本框是选读部分，但它使本书更完整并能激发读者进一步思考。

深入思考部分看起来是这样的：

深入思考：亲自实践。深入思考部分提供了更详细的代码解释或数学背景。这部分材料对理解主要文本并不那么关键，但它也很有价值，尤其是在第二遍阅读的时候。例

如，有时你的计算方式对结果是有影响的。从数学的角度看下面这两个表达式是等价的：

$$p_1 = \log(0.01^{200})$$
$$p_2 = 200 \times \log(0.01)$$

但是当你用 R 进行计算时，这两种方法得到的结果不同：

R code 0.3

```
( log( 0.01^200 ) )
( 200 * log(0.01) )
```

```
[1] -Inf
[1] -921.034
```

第二种方法得到的结果是正确的。之所以会有这样的问题，是因为 R 对小数的近似，计算机将很小的值近似为 0。精度的缺失可能导致推断结果大幅度偏差。这就是为什么我们在统计计算时，总使用概率的对数值，而非概率本身。

初次阅读本书时，你可以跳过所有的深入思考部分。

命令行是最好的工具。在 21 世纪，要达到能够实践统计推断的编程水平并不是那么复杂，但一开始你对此不太熟悉。为什么不教读者使用现成的用户交互软件呢？学习如何用命令行进行统计分析的好处远高于点击菜单。

谁都知道命令行功能很强大，但同时它的速度也很快，并且符合道德义务。因为用代码分析无形中对分析过程进行了存档。几年以后你可以通过这些代码重复之前的分析。你也可以重复使用之前写的代码，将它们分享给同事。鼠标点击的方式使得过程无法追踪。一个嵌入 R 代码的文件可以保存分析过程。一旦你习惯用这种方式计划、运行并且保存分析过程，在之后的职业生涯中会获益不断。如果你始终用鼠标点击的方法，那在之后要不停重复。不像使用代码，相同的分析只需要编写一次代码，之后可以用它进行同样的分析。分析过程的保存和可重复也是起码的科学道德，日后的检查以及项目的迭代也建立在此基础上。用代码进行统计分析可自然而然地达到这个结果，而鼠标点击却不行。

因此，我们不是因为要证明自己的技术实力或者自己是人才才使用命令行的。我们使用命令行是因为它确实更好。使用命令行一开始可能比鼠标点击难，因为你需要学习一些基本的语句才能开始使用。但是为了提高工作效率，我们应该使用命令行。

你该如何工作。如果我只是告诉你们使用命令行而不告诉你们怎么使用，那也太不厚道了。对于那些之前使用其他语言的人，你们可能要再学一些新语言的规则，但变化并不大。对于那些之前只用鼠标点击菜单的统计软件的读者，开始可能会觉得很不习惯，但过几天就会适应了。对于那些之前使用过其他通过命令行进行分析的软件（如 Stata 和 SAS）的读者，还需要调整适应。我会先解释总的方法，然后解释为什么 Stata 和 SAS 的使用者也需要适应。

首先，使用脚本统计分析是在一个**纯文本编辑器**和 R 语言之间来回切换。纯文本编辑器是用来创建和修改简单无格式文本文件的程序。常见的有 Notepad（Windows 操作系统）、TextEdit（Mac OS X 操作系统）、Emacs（大部分 * NIX 系统，包括 Mac OS X）。还有很多专门针对程序员的高大上的文本编辑器。我们推荐读者使用 RStudio 或者

Atom 文本编辑器，它们都是免费的。注意 MSWord 不是简易文本，不要用它来写代码。

你要通过简易文本编辑器记录在 R 中执行过的代码。你一定不想直接在 R 控制器中键入代码。你该在简易文本编辑器中写代码，然后复制粘贴到 R 控制器中运行。或者一次性将代码文本读入 R。如果你只是用 R 进行数据探索、查错，或者仅仅只是尝试一些代码，那可以直接将代码键入控制器。但任何严谨的工作都应该用文本编辑器记录代码，原因之前已经讲过了。

你可以在 R 代码中添加评论来帮助你编写代码，也方便之后回顾代码。在评论前键入 # 符号。为了确保大家理解，下面是一小段完整的进行线性回归的程序，使用 R 中的一个内置数据集。即使你现在不知道该代码是干什么的，但是希望你能将其看作一个基本的拥有注释的格式清晰的例子。

R code 0.4

```
# 载入数据:
# 车的刹车距离（英尺）[⊖]和速度（千米 / 小时）之间的关系
# 更多详情键入?cars
data(cars)

# 拟合距离和速度的线性回归模型
m <- lm( dist ~ speed , data=cars )

# 模型估计参数
coef(m)

# 绘制模型残差对速度图
plot( resid(m) ~ speed , data=cars )
```

最后，即使是那些熟悉 Stata 和 SAS 脚本语言的人，也需要重新学习 R。像 Stata 和 SAS 这类程序语言对应的处理信息范式和 R 是不同的。在使用中，过程命令如 PROC GLM 是在模仿菜单命令。这些过程会产生大量使用者不需要的默认输出。R 不是这样的，它强制使用者自己决定需要输出什么信息。你可以先用 R 拟合统计模型，然后接着用命令获取相应拟合结果信息。通过书中的例子，读者会更加熟悉这样的调查范例。但要注意，你需要主动决定需要模型的哪方面信息。

安装 R 包 rethinking

书中的代码例子要求你安装 R 包 rethinking。该包中含有例子中用到的数据以及本书使用的许多建模工具。rethinking 包本身依赖于另外一个包 rstan 来拟合本书后半部分讲到的更加高级的模型。

你必须先安装 rstan 包。根据你的系统，依照 mc-stan.org 网站上的相应安装指示。你需要安装 C++ 编译器（也叫作“工具链”）和 rstan 包。网站上有关于如何安装这两者的说明。

接下来你就能够在 R 中安装 rethinking 和其依赖的包，安装代码如下：

⊖ 1 英尺≈30.48 厘米——编辑注

R code 0.5

```
install.packages(c("coda","mvtnorm","devtools"))
library(devtools)
devtools::install_github("rmcelreath/rethinking")
```

注意 rethinking 不在 CRAN 的包列表中，至少现在还没有。把包上传到 CRAN 并没有实质的好处。你总能通过搜索引擎找到关于安装最新版本 rethinking 包的说明。如果你在使用包时发现任何 bug，可以到 github.com/rmcelreath/rethinking 上查看是否有现成的解决方法。如果没有，你可以提交一个 bug 报告，这样在有解决方法时你会收到通知。此外，如果你想对包中的一些函数进行修改，包的所有源代码可以在那里找到。大家可以自行从 Github 上 fork 该包，任意修改。

致谢

在本书的写作过程中许多人提供了宝贵的意见及想法。他们大多数是选修我教授的统计学课程的研究生，还有一些征求我意见的同事。这些人教我如何教授这些知识，有时我学习新的知识就是因为他们需要。许多人付出时间对本书的一些章节或者其中的代码进行了评论。这些人有：Rasmus Bååth、Ryan Baldini、Bret Beheim、Maciek Chudek、John Durand、Andrew Gelman、Ben Goodrich、Mark Grote、Dave Harris、Chris Howerton、James Holland Jones、Jeremy Koster、Andrew Marshall、Sarah Mathew、Karthik Panchanathan、Pete Richerson、Alan Rogers、Cody Ross、Noam Ross、Aviva Rossi、Kari Schroeder、Paul Smaldino、Rob Trangucci、Shravan Vasishth、Annika Wallin 以及很多匿名的审稿人。Bret Beheim 和 Dave Harris 很给力，他们对本书早期版本给予了大量的建议。Caitlin DeRango 和 Kotrina Kajokaite 花时间改进了几章和章后的习题。Mary Brooke McEachern 对本书的内容和讲解方式提供了重要意见，并且对本书的写作给予了支持，对（书中的不足之处）表现出宽容。许多匿名审稿人对各章提供了详细的反馈。他们中没有一个人是完全赞同本书的，书中的所有错误和不足都由本人负责。但是正是因为我们各执己见，才使本书更加与众不同。

本书献给 Parry M. R. Clarke 博士（1977—2012），是他促使我写作本书。Parry 对统计、数学和计算机科学的探索帮助了他身边的每一个人。他让我们变得更好。

目　录

01 CHAPTER

第1章
布拉格的泥人

在16世纪，哈布斯堡皇室(House of Habsburg)控制了中欧、荷兰和西班牙的大部分地区，还有西班牙在美洲的殖民地。哈布斯堡皇室可能是第一个真正在世界范围内具有影响力的家族，仿佛上天总是在眷顾他们。它的首领同时也是神圣罗马帝国的皇帝，权力中心位于布拉格。16世纪末，神圣罗马帝国的皇帝鲁道夫二世(Rudolph Ⅱ)是一位知识热爱者。他大力推动艺术、科学(包括占星术和炼金术)和数学的发展，使得布拉格成为当时的世界学术中心。在这样浓厚的学习氛围中，最早的机器人——布拉格的泥人——的出现显得合情合理。

这里所说的泥人(golem)是犹太传说中由黏土加水，然后经火烧制而成的机器人。传说在这些机器人的手肘处镌刻emet这个词(希伯来语中的解释是“真理”)，就能够让机器人复活。这些机器人由真理激活，但却没有自由意志，它们总是严格地按照指示行动。这很幸运，因为它们难以置信得强大，能够经受它们的制造者难以承受的考验，完成难以完成的任务。但它们的服从也会带来危险，因为不经思考的命令或者一些出乎意料的情况就能够使这些机器人反抗主人。它们的力量越大，智慧越少。

在一些关于机器人的传说的版本中，拉比Judah Loew ben Bezalel找到了一种保护布拉格犹太人的方法。16世纪在中欧的许多地方，布拉格的犹太人都遭到了迫害。通过使用喀巴拉(Kabbalah，犹太神秘主义中的一种)中的秘密技术，拉比Judah能够造出一种机器人，用“真理”将这些机器人激活，然后命令它们保护布拉格的犹太人。并非每个人都赞同Judah的行为，他们害怕使用这些力量强大的生命体会导致无法预期的后果。最终Judah被迫毁掉了这些机器人，因为这些缺乏智慧却有强大力量的生命最终伤及了无辜。通过将emet这个词中的第一个字母抹去剩下met(意为“死”)，拉比Judah毁掉了这些机器人。

1.1 统计机器人

科学家们也在制造机器人[⊖]。我们的机器人鲜有实体形式，但它们也是由实物制

⊖ 该比喻来自于Collins和Pinch(1998)，《The Golem：What You Should Know about Science. E. T. Jaynes》在2003年也类似将统计模型比喻成机器人，但是更加夸张。

造的，位于美国硅谷以计算机代码的面目示人。这些机器人就是科学模型。它们通过预测、挑战直觉或者激发灵感，真实地影响世界。对真理的关注使模型变得有生气，但和机器人一样，科学模型无所谓真假，既不是预言家，也不是江湖医生。它们是为了某种目的而设计出的构造，这些构造难以置信得强大，一丝不苟地执行编好的程序。

有时候它们缜密的逻辑可以揭示设计者之前没有觉察到的暗示，这些暗示可能是极有价值的发现。或者它们可以实施愚蠢并且危险的行为。科学模型不是理想中的理性天使，而是有蛮力而没有自由意志的泥制机器人，按照短视的指令踉跄前行。就像拉比Judah的机器人，科学机器人让人既敬畏又恐惧。我们当然要使用这些方法，但是这么做总是有风险的。

统计模型种类繁多。即使某人只是使用一个非常简单的统计方法，比如 t 检验，也算部署了一个小机器人，它会严格按照指示进行计算，且几乎[⊖]每次都进行相同的计算，从来不抱怨。科学的每一个分支都依赖于统计机器人是否能理智行动。在很多情况下，如果不建模的话，对感兴趣的现象进行测量几乎是不可能的事情。为了衡量自然选择的力量、中微子的速度，或者亚马逊丛林中物种的数目，我们必须使用模型。这里的机器人就好像假体，帮助我们测量、计算，寻找表面上并不明显的模式。

但是机器人没有智慧，不能辨别答案是否符合当前语境。它只知道按照程序实施过程，仅此而已。它只按指示行动。统计科学包含许多不同的模型，它们分别适用于不同的语境，这依旧是统计科学的胜利。从这个角度看，统计既不是数学也不是科学，而是一系列工程设计。和工程学类似，共同的设计原则和限制产生了各式各样有针对性的应用。

这些应用的多样性解释了为什么入门统计课程常常让初学者感到那么迷惑。学生在课上学的是很多现存的“假设检验”，而不是学习如何建立、优化和验证统计模型。每个假设检验都有各自的目的。如图 1-1 中所示的决策树就很常见。通过回答一系列问题，用户能够为当前研究的问题选出一个“正确的”方法。

不幸的是，虽然很有经验的统计学家能够掌握所有这些分析方法，但是学生和其他学科研究人员却很少能够这样。高阶统计课确实非常强调工程学原理，但是大部分科学家都没有学习这样高阶的课程。当前这样的统计学教育方式好比逆向讲授工程学，一开始先讲如何建设桥梁，结尾再讲基础物理。因此许多学生和研究人员使用图 1-1 这样的决策树，而不问原因，也不太理解决策树中牵扯到的模型，没有建立在实际研究中对各种方法进行利弊权衡的思维模式。这不是他们的错。

对一些人来说，一工具箱事先造好的机器人正是他们需要的。它们处在严格控制的环境中，仅仅需要根据任务相应使用合适的分析过程就可以很好地完成科学工作。这就好比水管工不需要了解流体动力学也能够熟练完成工作。但是一旦学者想要进行创新性的研究，扩展他们专业的边界，这时问题就产生了。好比我们想要将水管工升级成为水力工程师。

⊖ 没有任何算法和机器是从不损坏、弯曲或者失灵的。对于这点人们常常引用 Wittgenstain(维特根斯坦，1953)所著的《哲学研究》的 193 小节。我们之后考虑更复杂的模型和拟合数据的过程时，会对模型失灵的情况更感兴趣。