使用 R 语言进行渔业建模和定量方法研究

Using R for Modelling and Quantitative Methods in Fisheries

Malcolm Haddon

目录

# Preface

注：本书由 Chapman & Hall/CRC 在其 R 系列中出版。本书可从 Routledge 以及亚马逊等销售商处购买。

这本新书《使用 R 进行渔业建模和定量方法》由我早期的《渔业建模和定量方法》(Haddon 2011)演变而来。新书旨在介绍一系列在进行渔业资源评估时有用的分析方法，但与此同时，其中许多方法也更普遍地适用于生态学。通过集中介绍一般分析方法，同时保留早先对渔业建模的一些关注，其目的是使新书适用于更广泛的生态学读者，希望对他们有用，而不仅仅是渔业科学家。与原书一样，新书仍将文字与实例相结合。然而，时过境迁，用于此类分析的软件工具（如 R（R 核心团队，2019 ））也在不断发展，许多自然科学专业学生的背景知识也在不断发展。考虑到这一点，并且在收到许多关于提供早期示例框的 R 版本的电子邮件请求后，我想到编写一本新书，重点介绍许多分析技术，并使用 R 进行这些分析，这可能是一个有益的贡献。

早先的书应该是一本入门读物，然而，正如第二版序言(Haddon 2011)中所说：“……在涵盖一些更重要但本质上更复杂的方法时，要使这本书成为一本大多数人都能读懂的入门读物是一项挑战”。当时的挑战来自于尝试使用 Microsoft Excel 来实现这些高级方法，这需要工作表映射、宏和相关策略（希望大多数人会同意，在 Excel 中实现马尔可夫链蒙特卡洛分析并不是一项真正明智的工作）。本书仍然是一本入门书籍，目的是说明这些方法是如何工作的，并提供一个可以进行实验和学习的框架。希望这本书能为读者提供一个基础，在此基础上转向更多基于生产的方法。

我试图在书中加入更多的实际细节和开展工作所需的示例代码。为了补充和简化本书中的代码部分，我还开发了一个 R 软件包 **MQMF**，可从 <https://www.github.com/haddonm/MQMF> 获取，并发布到 CRAN，<https://cran.r-project.org/> 。书中的示例需要用到这个软件包。示例的 R 代码块全部在六个函数的帮助部分提供，分别称为 chapter2、chapter3、……、chapter7。每个示例都按照每章的顺序排列，并附有章节标题，以方便查找感兴趣的特定示例。在 RStudio 中，我们只需选择所需的代码部分，然后像往常一样按 {ctrl}{enter}。否则，选择代码并将其粘贴到 R 控制台或编辑器中，即可根据自己的意愿调整每个脚本。

我并不自诩为计算机程序员，而是一名科学程序员。这就意味着我写的代码可能没有那么高效，也没有那么好的结构。不过，我希望它的意图是明确的，而且相对简单易懂。R 代码有一个非常好的方面，那就是它通常是开源的，所以你可以随意修改或改进这里的代码。

与所有入门书籍一样，需要考虑的问题是包括哪些内容以及在何处停止。本书所包含的方法调查范围相对较广，对每种方法的处理深度也不尽相同。我的主要希望是，读者能在本书中找到他们认为对自己的工作和对分析的理解有用的东西。

Malcolm Haddon

Hobart, January 2020

## 致谢

我要特别感谢位于霍巴特的澳大利亚联邦科学与工业研究组织（CSIRO）海洋与大气部，感谢他们给了我成为该组织荣誉研究员的机会。能够继续使用图书馆和办公室为本书的写作提供了极大的便利。除了实用性之外，能在这样一个活跃的工作环境中与这么多优秀的人共事也很有价值，我在从事渔业科学工作期间曾与他们共事和合作。此外，我还要感谢塔斯马尼亚大学任命我为海洋与南极研究所的兼职教授。即使在 2008 年底离开塔斯马尼亚大学后，我仍继续与那里的许多优秀员工合作，特别是克雷格-蒙迪（Craig Mundy）博士，我还要感谢他允许我在 **MQMF** R 软件包中的许多数据集中使用鲍鱼研究项目的各种数据。

我还要感谢许多通过电子邮件与我联系的人，他们提出了许多有趣的问题。他们帮助我找出了我没有说清楚的地方。

最后，我要感谢我在Chapman & Hall/CRC的编辑罗布-卡尔弗（Rob Calver），他安排我在github页面或 <http://www.bookdown.org> 上免费发布本书的在线git-book版本。在我看来，bookdown 的最大优势之一就是它能为多种出版选择提供便利。开放源码社区（R、CRAN 和 RStudio 的部分内容是其中的最佳范例）是一项了不起的成就，我们都应该为它喝彩，并为相关人员鼓掌。

## 作品细节

本书是在RStudio（RStudio，2019）中使用Bookdown(Xie 2016)编写的，我使用了Chapman & Hall/CRC LaTeX class (kranz.cls)，该类经谢一辉修改，并由我自己对页面分数进行了一些非常小的试错编辑。代码块使用 Consolas 字体书写，将显示在略微灰色的方框中。软件包名称将以粗体显示（**MQMF**），函数名称（function()）将以圆括号结束，如果它们只是在文本中被引用。括号的作用是将它们与其他code 文本区分开来，后者通常是指变量、参数和函数参数。与数字有关的代码块一般指颜色，但本书是黑白印刷，因此也使用不同的图案或特定的顺序来区分不同的行。在其他地方，一般会使用颜色或图案，但不会同时使用两种。我在代码块和图例中保留了对颜色的引用，这样代码就保持了一般性，但希望不会造成混淆。

代码块的控制台输出一般会预置 # 号，而代码块中的整行注释则总是缩进一个空格。无论代码块试图使用什么字体（font=7 意味着衬线字体），用于定义 pdfengine 如何渲染数字的 LaTeX 类都设置为使用无衬线字体。因此，如果单独使用代码块，它们将以不同的方式渲染图表。

# 1. 关于建模

## 1.1 数据模型的特点

### 1.1.1 简介

常规的渔业种群评估通常基于种群生产过程和捕捞群体种群动态的数学模型。正面的生产过程包括补充动态（增加数量）和个体生长（增加生物量），而负面的生产过程包括自然死亡率和捕捞死亡率（包括选择性），这既减少了数量，也减少了生物量。种群动态包括一些细节，例如建模动态中使用的时间步长、是否对生物量或数量建模（年龄或大小，或两者兼有）、空间结构的细节以及手头案例的其他细节。如此多的潜在细节意味着生物种群和过程的数学模型种类繁多。然而，仍有可能就这些模式发表一些常规性声明。

由建模人员对现在已知过程或现象进行建模得到的所有模型都是抽象和模拟的。数学模型只是所有模型类别的一个子集，模型可能有多种形式，涵盖了构建的物理表示（想想由Watson 和 Crick，1953年制作的DNA球和棒模型），图表模型（如地理地图），以及这里讨论的更抽象的数学表达。我们可以通过关注模型的不同属性和建模人员做出的决定所施加的一些限制，对这种多样性施加某种概念顺序。

### 1.1.2 模型设计或选择

作为抽象，模型从来不是模型主题的完美副本，因此必须有一定程度的选择，建模者认为是系统的基本属性或组件。这种”基本属性或组件”的概念假定系统的所有部分并非都同样重要。例如，在人体血液循环系统的模型中，皮肤某处的浅静脉不会像肾动脉那么重要。如果接受这一假设，那么建模背后的一个基本想法是选择要包含的属性，以便模型的行为可能预期显示与建模系统的可观察行为接近。这种选择被认为是系统的重要属性允许，甚至迫使建模者强调系统建模的特定方面。路线图显示道路在真实地理尺度上大大放大，因为这是地图的要点。拓扑图强调不同的东西，因此在确定使用什么结构时，使用模型的目的也很重要。

选择一个系统在模型中包括哪些方面决定模型是否将普遍适用于一类系统，或者如此专业，以至于它试图模拟特定系统的详细行为（对于系统，人们可能会读取鱼群或种群）。然而，通过选择自然系统的特定部分，模型也受到约束，它可以描述。假设尽管没有完成，但它将充分描述利益进程，而且不包括这些方面不会意外地扭曲整个的代表性（Haddon，1980）。

当然，为了进行抽象，首先需要了解整体，但不幸的是，在现实世界中，还有许多仍然未知或被误解的东西。因此，模型很有可能成为所谓的”误用”。这就是模型的动态或行为，未能捕获正在研究的系统的全部动态。在本书稍后说明的一些示例中，我们将看到资源的平均预测生物质轨迹无法解释资源大小的振荡，显示大约 10 年周期（例如，参见第 [7](#sec-surplusproduction) 章”[剩余产量模型](07-spm.qmd)“的”Bootstrap 置信区间*“*节中拟合 dataspm 数据集的剩余产量模型）。在这种情况下，存在以未知机制作用于资源量的影响（或多种影响），这种影响以一种看似有规律或可重复的方式发生作用。假设这种规律是有意义的，由于其背后的机制不包含在模型结构中，那么，很明显，模型不能解释它的影响。这是典型的错误表达，虽然不是所有的错误表达都如此清晰或有如此清晰的规律。

模型设计或模型选择是复杂的，因为将模型放在一起时做出的决定将取决于已知的内容和模型的使用。

### 1.1.3 模型类型的局限

模型可以是物理、口头、图形或数学，但是，为模型选择的特定形式对其所能描述的内容施加了限制。例如，对动态种群过程的口头描述对任何人都是挑战，因为人们使用文字捕捉或表达种群的动态特性总是有限度的。单词似乎更适合静态对象的描述。这种限制不一定是由于演讲者缺乏语言技能。相反，这是因为口语（至少我知道的语言）似乎不适合描述动态过程，尤其是在系统中的多个变量或方面正在随着时间的推移或相对于其他变量而变化的情况下。令人高兴的是，我们可以认为数学是一种替代语言，它提供了描述动态系统的极好方式。但是，即使以数学作为我们描述的基础，也有许多决定需要做出。

### 1.1.4 数学模型

有许多类型的数学模型。它们可被描述为描述性、解释性、现实性、理想主义、一般性或特殊性：它们也可以是决定性的、随机的、连续的和离散的。有时它们可以是其中一部分或所有类型的组合。有了所有这些可能性，对于数学模型究竟在科学研究中能发挥什么作用，就有可能产生混淆。为了更好地了解特定模型的潜在局限性，我们将尝试解释其中一些术语的含义。

可将数学种群模型称为动态的，因为可以通种群/渔业的过去状态表示现在状态，并有可能描述未来的状态。例如，种群生物量动力学的Schaefer模型 (M. B. Schaefer 1957)可以部分地表示为：

其中变量 是时间 段捕获的渔获量， 是时间 开始时的群体生物量（ 也是模型的输出变量）。模型参数是 ，表示生物量（或数量，具体取决于对的解释， 或许 ）的种群增长率， 为系统所能达到的最大生物量（或数量）（这些参数来自早期数学生态学的 logistic 模型；参见第 [3](#sec-simplemodel) 章”简单种群模型*“*）。通过检查这个相对简单的模型，人们可以看到，在时间 的期望生物量水平 与渔获量和前期生物量直接相关 （时间 ; 该值是连续相关的） 。前期生物量对种群增长的影响由 和 这两个参数的共同调控。通过计算不同时间段的变量之间的序列相关性，这种动态状态模型与传统的统计分析有显著差异。序列相关性意味着，如果我们每年对一个种群进行抽样，那么严格地说，样本将不独立，这是更经典的统计分析的要求。例如，在封闭的种群中，一年内两龄鱼的数量不能超过上一年的一龄鱼的数量：他们不独立。

### 1.1.5 参数和变量

在最原始的层面上，数学模型由变量和参数组成的。模型的变量必须表示可定义或可测量的性质（至少在原则上是这样）。参数会修改变量对模型输出的影响或贡献，或与模型内变量之间的关系有关。参数是定量决定变量如何相互作用的因素。它们与模型的变量不同，因为参数是模型安装到观测数据时估计的参数。在 [公式 1.1](#eq-1_1) 中， 和 是变量， 和 是参数。可以有重叠，例如，人们可能会估计 系列中非常初始的值，也许 因此，该系列将由一个参数组成，其余的为 的直接函数， 和 为参数和 为变量。

在任何模型中，如 [公式 1.1](#eq-1_1) 中，我们必须估计或提供参数的恒定值。对于这些变量，任一提供观察到的值（例如，时间系列渔获量， ）或它们是模型的输出（ 如上所述 ）。因此，在 [公式 1.1](#eq-1_1) 中，给出时间系列的观测渔获量以及 、 和 的参数估计值，然后是一系列生物量值，，由模型作为输出。只要人们意识到在观测值、估计、变量、参数和模型输出等术语中可能出现混淆的可能性，就可以更清楚地了解在建模特定现象时究竟在做什么。理论与模型结构的关系不一定简单。背景知识和理论可能是模型结构选择背后的驱动力。一组变量之间提议的关系可能构成关于自然组织的一个新假设或理论，或者仅仅是对目前已知内容的总结，准备随着学习的更多而修改。

模型误用的另一个方面源于这样一个事实，即控制种群动态的参数往往被认为是随着时间而保持不变的，这通常应被承认为近似值。如果种群增长率 或承载能力 随时间而随机变化，但假设是恒定的，这将是所谓的过程错误的一个例子。这种过程错误将增加从人群中采集的样本的可观察到的变化，即使可以毫无差错地收集（没有测量错误）。如果参数因人口外部的某些因素（环境因素或生物因素（如捕食者或竞争对手）而变化，那么这种非随机反应就有可能增进对自然世界的了解。因此，构建模型时所作决策的一个重要方面是明确对所选结构的假设。

## 1.2 数学模型属性

### 1.2.1 决定论与随机性

我们可以将模型参数定义为*定量属性（系统建模），假定该属性要么在可用数据的期间保持不变，要么由环境变化调节*。大致而言，参数在模型应用的时间尺度上保持不变的模型称为确定性模型。对于给定的一组输入，由于其恒定的参数，确定性模型始终会为相同的输入提供相同的输出。由于模型变量之间的关系是固定的（恒定参数），因此给定输入的输出由模型的结构”决定”。人们不应被以下情况所混淆：确定模型中的参数通过采取一组预先确定值（例如，招聘指数或可捕获性指数可能每年更改和更改）而发生顺序变化。在这种情况下，虽然估计参数预计会随着时间而变化，但它们以可重复的、确定性的方式（在较长的时间尺度上保持不变）进行，给定输入始终提供相同输出的主要属性仍然有效。

确定性模型与随机模型形成对比，在模型所涵盖的时间段内，至少有一个参数以随机或不可预知的方式变化。因此，如果给出一组输入值，相关的输出值将不确定。不同的参数将从预定的概率分布（无论是从经典概率密度函数 （PDF） 之一）或自定义分布中随机值。因此，例如，在模拟鱼群时，每年的补充量水平可能达到平均值正负随机量，由随机变种 [公式 1.2](#eq-1_2) 的性质决定。

其中 是年 的补充量， 是年间的平均补充量（这本身可能是资源规模的函数）， 是用于随机变量的符号，其值在本示例中是均值为0 和方差为 的正态分布（即具有正值和负值）。通过将正态分布以指数项表示，这将指定为对数正态变异， 为补充量时间系列中 Log-Normal 变异的偏差更正项（Haltuch等，2008）。

模拟模型与具有估计参数的模型不同。这两种模型的目标也不同，前者可能被用来探讨不同管理方案的影响，而后者可能被用来估计资源的当前衰竭状态。

鉴于一组输入数据（假定是完整和准确的;注意这些假设），一个确定性模型表示其所有可能的反应。然而，随机模型构成了所谓的蒙特卡洛模拟的基础，其中模型是反复运行相同的输入数据，但每次运行新的随机值产生为随机参数，如 [公式 1.2](#eq-1_2) 。 对于每个运行，都会产生不同的输出，并绘制这些输出表或图表，以查看从此类系统中可以预期到哪些结果范围。即使模型固有的变化通常是分布的，但这并不意味着可以预期特定输出通常分布在某些平均值上。如果模型中有非线性方面，可能会出现偏斜和其他更改。

未来的种群预测、风险评估和确定数据中不确定性的影响都需要使用蒙特卡洛建模。模型结构的模拟测试是一个非常强大的工具。在 第 [6](#sec-uncertainty) 章 “[不确定性](06-uncertainty.qmd)（*On Uncertainty*）和第 [7](#sec-surplusproduction) 章“[剩余产量模型](07-spm.qmd)”（*Surplus Production Models*）中，详细介绍了这些预测的运行情况。

### 1.2.2 连续与离散模型

早期的渔业建模专家使用连续的微分方程来设计他们的模型，所以模型中的时间步长都是无限小的 （Beverton 和 Holt，1957）。当时，计算机还处于起步阶段，分析解决方案是当今的文化。因此，早期渔业模型是利用微积分形成的，其结构的某些部分更多地取决于可以分析解决的问题，而不是因为它们以特定准确的方式反映了自然。同时，这些模型的应用反映了或假定了平衡条件。幸运的是，我们现在可以使用易于访问的计算机和软件模拟资源状况，我们可以使用更现实或更详细的公式。虽然可能无法通过分析来求解此类模型（即，如果模型配方具有该结构，那么它的解就必须是这样的），但通常可以通过数值方式求解（了解情况并改进试验和错误）。尽管这两种方法仍在使用，但渔业科学的一大变化是从连续的微分方程向差异方程的转变，差分方程试图通过离散间隔（从无穷小到每年的时间步长）变化时对系统进行建模。

模型构建的其他方面可以限制模型可以捕获或描述的行为。模型的实际结构或形式施加了限制。例如，如果数学建模人员使用差分方程来描述系统，那么事件的分辨率不能比模型构建的时间间隔更精确。这种明显的效果发生在许多地方。例如，在包含季节性成分的模型中，分辨率明显有限，具体取决于可用数据是持几周、几月还是其他间隔。例如，在第 [5](#sec-staticmodel)章 “[静态模型](05-staticModel.qmd)”（*Static Models*）中，我们使用每周收集的数据来拟合季节性增长曲线，很明显，如果数据是每年收集的，那么描述季节性增长将是不可能的。

### 1.2.3 描述性与解释性

一个模型是离散的还是连续的，是确定性的还是随机的，这是模型结构的问题，它会明显影响着可以建模的内容。使用模型的目的也很重要。为了使模型具有描述性，它只需要模拟观察数据的经验行为。例如，对个体生长数据的精确拟合通常可以通过使用多项式方程获得:

其中未试图解释使用的多个参数(通常人们不会使用大于6阶的多项式，2阶或3阶更为常见)。这样的描述模型可以被视为黑盒，它为给定的输入提供确定性的输出。不需要知道这些模型的工作原理；甚至可以使用简单的查询表，通过从值的交叉列表中逐字查找输出，从给定的输入值生成特定的输出值。这样的黑盒模型只能是描述性的，除此之外别无其他。即使经验描述性模型可以做出假设，但如果特定的情况不能满足这些假设，这并不意味着需要完全拒绝模型，而只是必须限制该模型应用于哪些系统。除了所描述的变量外，这种纯描述性模型不需要具有现实主义元素，尽管它们的参数通常可以给出解释(如可实现的最大规模)。但同样重要的是，这些模型描述现有数据的程度，而不是它们的参数值是否具有生物学意义。在第 [4](#sec-paraestimat) 章“[模型参数估计](04-application.qdm)”（*Model Parameter Estimation*）中，我们将考察三条增长曲线，包括著名的 von Bertalanffy 曲线。该部分将对这种描述性模型的使用进行更深入的讨论。

解释性模型还提供了对感兴趣的实证观察的描述，但除此之外，它们试图提供一些理由或解释，一种机制，说明为什么所注意到的特定观察发生而不是不同的数据。对于解释性模型，有必要考虑假设和参数，以及构成模型的变量。通过尝试使参数和变量，以及变量如何相互作用，反映自然，解释模型试图模拟自然界中的真实事件。如果模型包含理论构造（假设、变量或参数），则模型具有解释性，这些结构声称与自然过程有关，而不仅仅是与自然的行为方式相关。

### 1.2.4 测试解释模型

解释性模型至少部分地是关于自然的机制和结构以及自然如何运作的假设或理论。因此，它们应该根据自然界的观测结果进行测试。但是，我们如何测试解释模型呢？数据拟合模型可以提供模型测试吗？如果模型预测的观测数据的预期值占观测数据内变异性的很大比例，那么我们对模型充分描述观测结果的信心可能很大。但初始模型拟合并不构成对模型结构的直接测试。拟合模型并不测试模型是否解释观测到的数据；它只测试模型描述的程度和与数据一致性（Haddon，1980）。解释和描述之间的区别非常重要。一个纯粹的描述性或经验模型可以提供同样适合的数据，这有望表明，我们需要进一步的，独立的观察，以真正测试模型的结构。需要测试的不仅是模型是否适合一组观察到的数据（即不仅适合的质量），而且模型假设是否有效，以及模型变量之间的相互作用（如模型中的编码）是否密切反映自然。

将目前拟合的模型与新的观测结果进行比较确实构成了某种测试。理想情况下，考虑到特定的输入，该模型将提供预测的观测以及围绕预期结果的置信间隔。如果模型预测，鉴于输入，其价值极不可能，则观察结果将与模型不一致。但是，有了这个测试，如果有反驳，没有迹象表明模型的哪个方面有问题。这是因为这不是对模型结构的测试，而只是对特定参数值（ 给定模型结构）是否足以预测未来结果的测试！我们不知道拟合过程是否有限，因为现有数据没有充分说明所研究资源固有的变化潜力。是假设还是建模者使变量相互作用的特殊方式？模型是否过于简单，这意味着重要的相互作用或变量被排除在结构外？如果没有对假设或特定变量重要性的独立测试，我们就无法判断。

如果新的观测结果与模型一致，那么人们就没有什么收获了。实际上，新数据很可能会被包括在原始数据和重新估计的参数中。但这同样适用于纯粹的经验模型。所需要的是独立测试，确保所选择的结构不遗漏重要的变异来源；要验证这一点，需要的不仅仅是将预期输出与实际观测结果进行简单比较。

虽然我们可以满足于观测数据和模型预测数据之间的拟合质量，但我们永远无法确定我们确定的模型是最好的。当然，有些模型可能看起来不太可接受，因为其它模型可能更有效地拟合数据。

然而，任何关于哪条曲线或模型最能代表一组数据的讨论，不仅取决于拟合的质量，而且还取决于有关变量之间关系形式的其他信息。带有每个数据点参数的经验模型可以精确地拟合数据集，但不能提供任何有用的信息。显然，在这种情况下，除了数值拟合的质量之外，还必须使用其他标准来决定应该选择哪个模型。在第 [5](#sec-staticmodel) 章”[静态模型](05-staticModel.qmd)“（*Static Models*）中，我们考虑了第 [5.4](#sec-label) 节 *目标模型选择* 的方法，试图评估增加模型中的参数数量在统计上是否合理。任何解释模型都必须具有生物学上的合理性。甚至可以给一个完全任意的模型结构的参数赋予意义。然而，这种解释将是临时的，而且仅在表面上可信。模型除了描述一组特定的数据外，不可能做更多的事情。解释性模型应该适用于新的数据集，尽管可能需要一组新的特定参数来适应新的情况。

即使在现实的模型中，精度也不可能实现，因为我们对拟合变量的估计(观测误差)或系统响应(可能是环境变化(模型参数的过程误差))的内在不确定性。换句话说，在我们预测的系统结果的精度方面，可能无法超越某些限制(拟合质量可能有内在限制)。

### 1.2.5 现实主义与普遍性

与我们是否应该使用解释性模型相关的问题是模型中的现实主义问题。纯粹的描述性模型不需要任何现实的东西。但这只是一个假设，即如果某个科研人同正在开发一个解释模型，那么至少解释模型的一部分必须是现实的。例如，在年龄或体长可以区分的种群中，年龄或体长结构模型会被认为比将所有年龄或体长组集中在一起的模型更现实。但是一个模型可以是真实的和经验的结合。

一般模型将具有非常广泛的适用性领域，即在许多情况下可以有效应用。在渔业科学的发展中，将许多描述特定过程的模型（例如，个体生长）纳入一个更普遍的数学模型，这些模型是特殊情况（见第[5](#sec-staticmodel) 章“[静态模型](05-statiicModel.qmd)”（*Static Models*））。通常这涉及增加所涉及的参数数量，但尽管如此，这些新模型显然在数学上更为通用。很难就这种更笼统的方程/模型是否不太现实得出结论。这将是一个问题，是否额外的参数可以现实地解释，或者它们是否只是临时解决方案，将不同的方程组合成一个更数学通用的方程。随着更复杂的现象，如年龄结构模型，一般模型通常不会给出准确的预测，因为更专业的模型调整到特定的情况。正因如此，建模人员在处理特定情况时，往往认为数学上一般模型不太现实（Maynard-Smith, 1974）。

### 1.2.6 模型是理论

所有模型都可能被认为具有理论成分，甚至被认为是经验模型。它成为一个感知问题，而不是模型结构。例如，通过简单的模型，基本假设可以开始承担假设断言的重担。因此，如果使用 logistic 方程来描述种群的增长，它就导入了种群增长率的密度依赖补偿与种群密度线性相关的假设。换言之，种群规模增长对种群增长的负面影响与种群规模呈线性关系（见第 [3](#sec-simplemodel) 章“简单种群模型*”*（*Simple Population Models*））。这可以被视为一个领域假设（即模型只能有效地应用于密度依赖效应与种群密度线性相关的情况）或理论（非线性密度依赖效应在建模的系统中不重要）。这显然是一个感知或建模目标的问题，即这两种可能性中哪一种是获得的。这是一个很好的理由，人们应该明确解释一个人的模型的假设。

如果将自己局限于纯粹的经验关系，那么他的模型唯一可以改进的方法就是增加模型所解释的观测结果的方差。没有有效的期望，经验模型将提供洞察一个系统的未来行为。解释/理论模型的一个优点是，它应该能够检验假设、变量之间的关系和误差结构，独立于与观测结果的拟合质量。

因此，应该有可能提出证据来支持一个模型，这超出了拟合的质量。那些建议的结构没有以这种方式得到支持的模型，也可能是经验性的。

## 1.3 结束语

编写和讨论模型，它们的使用和构建有时是有价值的，因为它提醒我们在其中工作的框架。如果你想成为一名建模人员，对数学模型的优缺点的理论理解总是有价值的。然而，通常理解模型及其属性的最佳方法是实际使用它们，并通过操作它们的参数和检查它们在实践中如何操作来探索它们的属性。希望您会发现，使用 R 作为编程语言使这些探索相对容易实现。

下面的材料包括非常一般的方法和其他更具体的方法。这本书的目的是鼓励你，也许是为你提供一个开始，发展你自己的分析功能，也许是通过修改本书中的一些内容。您可以这样做，以便您自己的分析变得更快、更容易，并且在某种程度上是自动化的，从而让您有更多的时间来思考和解释您的发现。如果您使用的是其他更少的程序性分析的话，您需要机械地进行分析的时间越少，就有越多的时间来思考科学问题并进行更深入的探索。使用R来实现您的建模的一个主要优势是，您所做的任何工作都应该变得更容易重复，因此，大概也更容易防御。当然，这里所涉及的主题范围只是触及了可用内容的表面，但试图探索一些基本方法，如最大似然估计。请记住，有大量的 R 包可用，这些可以帮助您实现自己的模型，无论是统计或动态。

# 2. R 语言的不完全介绍

## 2.1 简介

这本书是《*渔业建模和定量方法*》（*Modelling and Quantitative Methods in Fisheries*）中描述的一些材料的分支和改编版本 (Haddon 2011)。主要变化涉及使用 R语言（R 核心团队，2019 ）以扩展和实施选定的渔业和生态分析。这本书以不同的文字，更多是强调如何进行分析，在许多地方，使用的例子将在细节上与先前书中使用的不同。

使用 R语言 进行分析与使用 Microsoft Excel (Haddon 2011)（如Haddon, 2001, 2011 ）是非常不同的命题。这意味着分析的机制和结果不会直接摆在分析人员面前。但是，即使在使用 Excel时，如果分析涉及到在Visual Basic for Application 中使用宏，那也涉及到一定程度的抽象。使用 R语言 作为编程语言只是在所有的例子中完成抽象。一般来说，我尽量坚持”没坏就不修”的原则，所以你可以假设有很好的理由搬到 R。

## 2.2 R语言中的编程

开源软件R语言对于进行统计分析显然很有用，但它也提供了非常灵活和强大的编程环境。对于科学程序员来讲，R有很多优势，因为它是一种非常高级的语言，可以通过编写自己的脚本或函数来扩展这些命令，以进行专门的分析或其他操作，如数据操作或定制制图。如此灵活有很多优点，但也意味着学习 R 的语法可能是面临着挑战。很多可用的，大多是开源的R包（请参阅<https://cran.r-project.org/>），增强其灵活性，但也有可能增加其语法的复杂性。

幸运的是，已经出版了许多优秀的书籍，可以向人们展示他们可能需要使用 R 并成为 R 程序员的技能 (如， Crawley 2007; Matloff 2011; Murrell 2018; Venables 和 Ripley 2002; Wickham 2019；等)。Wickham (2019) 的书为《*高级R*》（*Advanced R*），但不要让”高级”让你失望，它充满了优秀的细节，如果您想先体验一下，可以在以下网址查看网络版本 <https://adv-r.hadley.nz/>。本章附录中列出了其他学习资源。

它还试图描述一些复杂性，比如如何处理不确定性，以及如何处理它们。本书的重点是开发和拟合数据模型，使它们运行，并展示它们的输出。

因此，本书不是用来教任何人如何使用 R 编程的。事实上，此处包含的 R 代码很少优雅，也不太可能是万无一失的，但其目的是尽可能简单地说明使用 R 用数据拟合生物模型的不同方法。它还试图描述一些复杂性，如处理不确定性，这个问题你一直会碰到，以及如何处理他们。本书的重点是开发和拟合数据的模型，使其运行，并展示结果的输出。然而，R 中有几个构造将在接下来的内容中多次使用，本章旨在引入它们，以便用户在相对较新的 R 时不会放慢速度，但需要强调的是，假设用户熟悉 R。我们将简要介绍如何开始，如何检查函数内的代码、打印、绘图、因子的使用，以及函数的使用和编写。重要的是，我们还将考虑使用 R 的非线性解算或优化。会考虑许多示例，并包括示例数据集以加快这些示例速度。但是，如果读者有自己的数据集，那么这些数据集应尽可能使用，因为很少有东西能鼓励学习和理解，以及分析自己的数据。

### 2.2.1 从 R 语言开始

请记住，这本书的目的不是教任何人如何使用R，而是试图展示R可以用于生态分析的方法，特别是种群动态和渔业。在这本书中开始使用R需要一些东西:

* 从<https://cran.r-project.org/>下载最新版本的 R 语言并将其安装在计算机上。
* 下载 RStudio，可以从 RStudio 网站<https://rstudio.com/>上的下载选项免费获得。RStudio 现在是一个成熟且结构良好的研发环境，建议使用RStudio。或者，使用您自己选择的其他软件来编辑文本文件和存储代码文件。找到操作方式，自己的工作流程，这对你感觉最好。理想情况下，你应该能够专注于你试图做什么，而不是你是如何做到这一点。
* 从 CRAN 或 github 下载本书的程序包 **MQMF**。要从 CRAN上下载，可以非常简单地使用RStudio中的 “package”选项 。也可在 <https://github.com/haddonm/MQMF> 的github上 ，克隆包的R项目文件或直接使用devtools用下列命令直接安装devtools::install\_github("https://github.com/haddonm/MQMF")
* 至少拥有一些关于R的实用知识以及如何使用它。如果你是R的初学者，明智的做法是使用一个或多个介绍性文本，以及在网上搜索 “introductory R” 也将产生一个实质性的清单。RStudio 的文档也会让你开始。请参阅本章的附录。

为了节省空间，本书的文本中将只描述分析中使用的功能的一些细节。从 **MQMF** 包中检查代码的详细信息将是了解每个分析以及如何实现这些分析的一个重要部分。阅读每个功能的帮助文件（键入控制台），并尝试提供示例。在 RStudio 中，人们只需在帮助页面中选择示例代码，并且像往常一样，使用 {ctrl}{enter} 来运行选定的行。或者，选择并复制它们到控制台中以运行示例，改变输入以探索它们的工作原理。包装功能很可能包含额外的细节和错误捕获，这是改进，但这需要空间和时间在书中的阐述。

### 2.2.2 R 语言程序包

用于各种各样分析的大量 R 包（几千个）（见<https://cran.r-project.org/>）是R语言令人难以置信的强项，但公平地说，这种多样性也导致了学习所有可用函数所需的 R 语法的一些复杂性。基础 R（安装 R 时状态）自动包含 7个 R 包（请在控制台中使用 sessionInfo() 尝试查看列表）。因此， 在 R 中使用R包通常情况下是很自然的，不使用其他人的包将是一个非常愚蠢的想法，因为这意味着将有大量的完全可用的包的重复使用，但因为这里的目的是说明分析的细节，而不是专注于 R 语言，主要是使用默认或”基础”安装，虽然在少数地方，建议自己将作出考虑和探索R包。例如，当我们需要使用多元正态分布时，我们将使用 **mvtnorm** 包。在 RStudio 中使用 [package] 选项卡，很容易安装这些 R 包。或者，控制台中使用 install.packages("mvtnorm") 安装，这也应该做的工作。其他包将根据需要进行安装。无论我在哪里写 “install library(xxxxx)”，都应理解为安装 指定库及其所有依赖库。许多程序包都需要并调用其他程序包，因此除了程序包自身外，还需要有其它依赖包。例如，**MQMF** 包需要安装包 **mvtnorm** 和 **MASS**。如果你在控制台中输入 packageDescription("MQMF)，就会列出程序包的描述文件内容（所有程序包都有显示）。在显示内容中你会看到，**MASS** 和 **mvtnorm** 都需载入，因此它们需要随着 **MQMF** 一起安装。

这本书侧重于使用”基础”R包，这是一个有意识的决定。目前，R 社区内部正在发生一些重要发展，特别是所谓的”tidyverse”的出现，其为的库集，即旨在改进或至少改变 R 使用方式。您可以在<https://www.tidyverse.org/>上探索使用 R 的新方法，其主页目前指出：“tidyverse 是专为数据科学设计的 R 包的有意见的集合。所有软件包都具有基本的设计理念、语法和数据结构。其中许多创新确实是有效的，非常值得学习。但是，”基础”R 的一个优点是，它非常稳定，只有经过对与以前版本兼容性的广泛测试后才引入更改。此外，开发使用”基础”R 的基准，可以更知情地选择如果您愿意，整洁的哪些部分可以采用。无论情况如何，在这里，我们将使用尽可能少的额外库，除了’基地’安装和书自己的包。

### 2.2.3 开始使用 MQMF

**MQMF** 程序包已经过测试，在 PC、Linux（Fedora） 和Apple计算机上的行为都与本书中的预期一致。在 Apple 计算机上，有必要在 PC 上安装包的 .tar.gz 版本，在 PC 上，无论是 .zip 还是 .tag.gz 都可以安装，RStudio 也可以为您安装此 R 包。tar.gz 版本的一个优点是，源文件和库的 tar.gz 存档包含所有原始代码，以便在您愿意时进行更深入的检查。此代码也可以在 MQMF github 仓库中检查。

安装 **MQMF** 包后，通过命令 library(MQMF) 可以将该包含在 R 会话中。通过使用 ?MQMF 或 ?MQMF-Package 立即生成某些文档。该列表的底部是包名、版本号和”索引”的超链接。如果您单击该”索引”链接，则会生成与 **MQMF**内与导出函数和数据集相关的所有文档列表；这也可适用于检查任何 R 包。通过打开所需的链接或使用 ? 操作符可列出包中包括的每个输出函数和数据集，也可以获得每个包的更多详细信息；例如，尝试在控制台键入或输入 ?search 或 ?mean。RStudio 的 “Help” 选项卡中生成的材料描述了该函数的功能，并且通常至少提供了如何使用函数的示例。在本书中，可以在 RStudio 中的控制台输入R代码或作为脚本保存并发送给 R 以 consolas 字体（所有字符都使用相同空间的单间间字体）在略带灰色的框形中呈现。包名用粗体表示，函数名以 function() 开始，以圆形括号结束。括号是将其与其他 code 文本区分开来，后者通常将指变量、参数和函数参数。

当然，开始的另一种选择是完成所有必需的安装，然后开始下一章的工作，尽管首先完成本章的其余部分可能是明智的，因为它将介绍绘图和检查不同函数的工作方式等内容。

### 2.2.4 在函数内检查代码

提高对任何编程语言理解的一个有效方法是研究其他人如何解决不同的问题。R 的一个极好的方面是，只需在控制台中键入一个函数名，即可研究其他人的代码，而省略任何括号。如果使用 library(MQMF) 加载 **MQMF**，比如，则可尝试键入 plot1。 如果RStudio帮助添加了一组括号，那么在按 return 之前删除它们。这将 plot1() 的代码显示在控制台上。尝试完全自己键入 lm，则会从 stats包中列出线性模型函数的内容。在检查函数（如 mean ）时，有许多函数仅提到类似 useMethod("mean") 的内容。这表明该函数是 S3 通用函数，指向可用于该函数的每个对象类别的不同方法（假设读者知道 R 中的不同类，如向量、矩阵、列表等;如果不搜索”R 类”，则应启发您）。当然，R 中有许多类别的对象，但除此之外，还可以定义自己的类。当我们讨论我们自己编写的函数时，我们会多谈一些这类事情。同时，当找到 UseMethod("mean") 时，然后可用于这些函数的具体方法可以通过使用 methods("mean") 列出：例如，尝试 method("print")。每个此类通用函数都有默认行为，因为它指向对象类，而对象类没有特定定义。如果你键入 getS3method("print","default")，那么你可以看到与函数相关的代码，并可将它与 getS3mehtod("print", "table")的代码进行比较。例如，在某些情况下，print.table将直接运行，但 getS3method 应始终有效。如果想要开发可开发的专用 R 对象（定义类）的定制打印、绘图或进一步处理，则可以自己使用 S3 类 (Venables 和 Ripley 2002; Wickham 2019)。

一旦加载了一个 R 包，比如 library(MQMF)，其中的输出对象列表可通过使用 ls 函数得到 ，也可用 ls("package:MQMF") 或 ls.str("package:MQMF")； 通过使用 ?ls.str 查看两个 ls函数。或者，无需加载包，也可以使用，例如，mvtnorm::dmvnorm、 MQMF::plot1 或 MQMF::'%ni%'，其中:: 允许我们查看内部包，假设它们至少已安装完毕。如果 :: 报告不存在此类函数，请尝试三个冒号 :::，因为仅在包内内部使用的函数通常不会导出，这意味着没有 :::选项就不容易访问（即使未导出其名称也可能在导出的函数内可见）。最后，无需先加载库，就可以通过使用 getNamespaceExports("MQMF") 看到导出的函数名，其输出可以通过使用 sort(getNamespaceExports("MQMF")) 命令进行简化。

### 2.2.5 使用函数

R是一种使用函数的语言，它既可以进行交互使用，也可以使用脚本 (Chambers 2008, 2017; Wickham 2019)。这意味着最好对函数、它们的结构以及如何使用它们有一个很好的理解。您会发现，一旦开始编写代码块，自然就会将它们封装到函数中，这样重复调用代码就会变得更简单。MQMF的示例函数 countones 可用来说明其结构。

#make a function called countones2, don't overwrite original   
countones2 <- function(x) return(length(which(x == 1))) # or   
countones3 <- function(x) return(length(x[x == 1]))   
vect <- c(1,2,3,1,2,3,1,2,3) # there are three ones   
countones2(vect) # should both give the answer: 3

[1] 3

countones3(vect)

[1] 3

set.seed(7100809) # if repeatability is desirable.   
matdat <- matrix(trunc(runif(40)\*10),nrow=5,ncol=8)   
matdat #a five by eight matrix of random numbers between 0 - 9

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
[1,] 5 9 6 0 5 3 1 7  
[2,] 8 8 5 2 1 5 8 2  
[3,] 7 9 5 1 7 1 0 8  
[4,] 5 1 7 9 2 6 2 9  
[5,] 5 5 4 1 6 0 2 4

apply(matdat,2,countones3) # apply countones3 to 8 columns

[1] 0 1 0 2 1 1 1 0

apply(matdat,1,countones3) # apply countones3 to 5 rows

[1] 1 1 2 1 1

countones2()函数的作用是读取一个向量 ，计算其中 1 的个数，并返回该个数;它可以单独使用，也可以在另一个函数(如 apply()中使用，以将 countones()函数应用于矩阵或 data.frame 的行或列。apply()函数以及相关的 lapply()和 sapply()函数，一般用作列表（参见 lapply()的帮助文件），对于将给定函数应用到矩阵的列或行和data.frames 都非常有用。data.frame 是特殊的列表，允许混合类的矩阵。可以生成在 apply() 族中使用的非常短的函数，MQMF包括了countgtone、countgtzero、countones、countNAs和countzeros，作为示例实用函数;每个帮助文件中都给出了示例。

函数可有输入参数，也可有默认值，它应该返回一个对象(可能是一个值，一个向量，一个矩阵，一个列表，等等)或执行一个操作。理想情况下，不应改变工作或全局环境 (Chambers 2017; Wickham 2019) ，尽管对象打印或绘图的函数是个例外:

#A more complex function prepares to plot a single base graphic   
 #It has the syntax for opening a window outside of Rstudio and   
 #defining a base graphic   
plotprep2 <- function(plots=c(1,1),width=6, height=3.75,usefont=7,   
 newdev=TRUE) {   
 if ((names(dev.cur()) %in% c("null device","RStudioGD")) &   
 (newdev)) {   
 dev.new(width=width,height=height,noRStudioGD = TRUE)   
 }   
 par(mfrow=plots,mai=c(0.45,0.45,0.1,0.05),oma=c(0,0,0,0))   
 par(cex=0.75,mgp=c(1.35,0.35,0),font.axis=usefont,font=usefont,   
 font.lab=usefont)   
} # see ?plotprep; see also parsyn() and parset()

但愿这个 plotprep2() 示例能够强调这样一个事实：如果您看到某人编写的函数，您可以对其进行定制以更符合自己的需求，那么没有理由你不应该这么做。 当然，在文档中引用已知的原始函数是很好的做法，但是编写自己的函数的主要优势正是这种自定义工作环境的能力(请参阅后面的*编写函数*部分)。

### 2.2.6 随机数生成

R提供了大量的概率密度函数(例如，正态分布、Beta 分布等等)。对建模人员有很大好处的是，R还提供了从每个分布中生成随机样本的标准方法。当然，这些样本实际上是伪随机数，因为真正的随机过程是非常难模拟的。尽管如此，R中还是有一些非常有效的伪随机数生成器。试着在控制台中输入 RNGkind()，看看当前在您自己的计算机上使用的是什么。 如果在代码中键入 RNGkind （不带括号），在代码中，将看到从*kind* 向量中可用的生成器列表。

伪随机数生成器在长循环中生成一长串数字，每个数字都依赖于前面的数字。这类生成机的早期版本臭名昭著地具有短循环，但R中默认的生成器，即众所周知的 Mersenne-Twister，据报道其周期为 ，这是一个相当大的数字。然而，即使是一个大的循环，它也必须从某个地方开始，而那个地方被称为种子。如果每次都使用相同的种子，那么每次尝试使用它们时就会获得相同的伪随机值系列。如果使用随机数的特定模拟具有特定的行为方式，那么这种重复性就非常有用，因为它有助于了解所使用的种子，以便能够进行重复。在 ?RNGkind 的帮助文件中，它指出:“最初，没有种子;当需要时，从当前时间和进程ID创建一个新的。因此，在默认情况下，不同的会话将给出不同的模拟结果。”但是该帮助也声明:“提供的大多数统一生成器返回被转换为双精度的32位整数值，因此它们最多接受 个不同的值，非常长的运行将返回重复的值”。 10亿(千百万) 所以要警告。它还警告说，如果恢复以前保存的工作区，那么以前的种子将会返回。这些隐藏的细节是值得记住的，这样人们就可以充分意识到分析实际上在做什么。

为了使我们自己的模拟可重复，我们希望保存使用的种子和特定试验的结果。可以使用 set.seed() 函数设置种子，如 set.seed(12345)。我特意用 12345 来说明，在你的头脑中创建一个种子不一定是一个好主意。我们可能都有很好的想象力(也可能没有)，但当我们要编造一系列随机的种子时，最好使用不同的方法。尽管人们的最佳意图是编造种子，但这往往会导致重复使用特定数字，从而导致模拟结果存在偏差。生成此类种子的一种方法是使用 **MQMF** 函数 getseed()。它使用系统时间（ Sys.time()）获得一个整数，该整数的数字随后被伪随机地重新排序。看看它的代码，看看它是如何工作的。 似乎只在真正需要能够重复使用随机数的特定分析时才使用 set.seed() 是合理的。

#Examine the use of random seeds.  
library(MQMF)  
seed <- getseed() # you will very likely get different naswers  
set.seed(seed)   
round(rnorm(5),5)

[1] -1.19869 -1.28760 -0.03116 -0.39546 1.48911

set.seed(123456)   
round(rnorm(5),5)

[1] 0.83373 -0.27605 -0.35500 0.08749 2.25226

library(MQMF)  
set.seed(seed)   
round(rnorm(5),5)

[1] -1.19869 -1.28760 -0.03116 -0.39546 1.48911

### 2.2.7 R语言中的输出

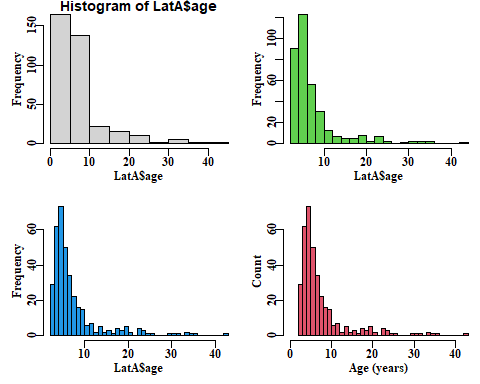
大多数显示意味着分析的输出被写入控制台，尽管将对象的内容写入文件的选项也存在（参见?save 和 ?load，也参见 ?write.table 和 ?read.table）。如果您想要保存发送到控制台的所有输出的记录，那么也可以使用 sink() 函数来做到这一点，它将输出从R转换到一个指定的文本文件。通过使用 sink(file="filename"， split=TRUE)， split=TRUE 参数意味着可以将输出显示在屏幕上，也可以发送到文件中。可以通过重复 sink() 调用来关闭文件的输出，但括号中不包含任何内容。如果没有显式包含”filename”的路径，那么该文件将被发送到工作目录(参见 ?getwd)。sink() 函数对于记录会话很有用，但是如果会话在关闭接收之前崩溃，那么它仍然可以运行。与其试图记住这个问题，还不如在脚本的早期包含类似于 on.exit(suppressWarnings(sink())) 的内容。on.exit() 函数对于在运行可能无法完成的分析之后进行清理非常有用。

### 2.2.8 R语言中的绘图

使用R简化了高质量图形的制作，在 www 和书籍中都有许多有用的资源描述R的图形可能性 (例如 Murrell 2018)。在这里，我们只关注于使用R标准安装中的基本图形。与R中实现的其他主要图形包相比，基本图形包的复杂程度或能力较低是可能的(例如，您应该探索使用 lattice、ggplot2、graph、vcd、scatterplot3d 和其他可用图形包的选项;您可能会发现其中一个或多个可能比基本图形更适合您的编程风格)。然而，通过基础或传统图形，可以满足我们所需要的大多数绘图需求。如何在R中生成图形提供了一个很好的说明，在任何给定的编程语言中通常有许多方法来做同样的事情。对于特定的任务，有些方法显然比其他方法更有效，但无论如何，找到你觉得舒服的方法、方法和包是一个好主意，因为这里的目的是快速、正确地完成工作，不要学习 R 语言中所有的选项(现在所有可用的包都非常广泛）。也就是说，你对R语言和语法的了解越多，你解决特定问题或任务的选择就越多;到处都有权衡!

其中，**MQMF** 包含一个执行绘图函数的函数数组。关于R的一个优点是，通过调整各种图形参数，可以很容易地生成并实时优化图表和绘图，直到得到所需的形式。一旦最后定稿，还可以生成出版物质量的图形文件，以便编入报告或随手稿寄出。首先检查 MQMF::plotprep 代码，以了解它如何定义绘图区域、边距以及如何格式化绘图的其他细节。plotprep() 可以用作快捷设置格式为基础的图形块(外部RStudio情节窗口),如果想要变化，那么 parsyn() 为 par() 函数写的核心语法到控制台,可以复制到您自己的代码和用于起动器基本图形绘制。如果你满足于RStudio中的绘图窗口，那么你只需要 parsyn() 。一旦熟悉了 par() 函数的语法，您就可以使用 MQMF::parset() 函数作为快捷方式或包装器函数。

library(MQMF) # The development of a simple graph see Fig. 2.1  
data("LatA") #LatA = length at age data; try properties(LatA)   
 #The statements below open the RStudio graphics window, but opening   
 #a separate graphics window using plotprep is sometimes clearer.   
 #plotprep(width=6.0,height=5.0,newdev=FALSE)   
setpalette("R4") #a more balanced, default palette see its help   
par(mfrow=c(2,2),mai=c(0.45,0.45,0.1,0.05)) # see ?parsyn   
par(cex=0.75, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
hist(LatA$age) #examine effect of different input parameters   
hist(LatA$age,breaks=20,col=3,main="") # 3=green #try ?hist   
hist(LatA$age,breaks=30,main="",col=4) # 4=blue   
hist(LatA$age, breaks=30,col=2, main="", xlim=c(0,43), #2=red   
 xlab="Age (years)",ylab="Count")



An example series of histograms using data from the MQMF data set LatA, illustrating how one can iterate towards a final plot. See also the function uphist().

### 2.2.9 因素处理

R的一个可能导致挫折和困难的方面是使用因素来表示分类变量。这将分类变量转换为因子的级别(不同值)，甚至是已经有数值的因子。例如，在标准化渔业 CPUE 时，通常至少使用一些分类变量，因此，如果一个水深类别为 50 - 600 米，每个步骤为 50，那么在标准化中需要将这些类别作为固定的分类因子而不是数值处理。但是，一旦转换为一组因子，它们就不再表现为原始的数值变量，而这通常是绘制时所需要的。但是，如果它们最初是*数值*的，则可以使用函数 facttonum() 进行恢复。在探索与因素有关的一些问题的示例代码中，请注意 try() 函数的使用（另请参阅 tryCatch()）。这用于防止 r-code 由于试图将一个因子乘以标量的错误而停止。这在开发代码和使用 bookdown 和 Rmarkdown 编写书籍时非常有用。

#Dealing with factors/categories can be tricky   
DepCat <- as.factor(rep(seq(300,600,50),2)); DepCat

[1] 300 350 400 450 500 550 600 300 350 400 450 500 550 600  
Levels: 300 350 400 450 500 550 600

try(5 \* DepCat[3], silent=FALSE) #only returns NA and a warning!

[1] NA

as.numeric(DepCat) # returns the levels not the original values

[1] 1 2 3 4 5 6 7 1 2 3 4 5 6 7

as.numeric(levels(DepCat)) #converts 7 levels not the replicates

[1] 300 350 400 450 500 550 600

DepCat <- as.numeric(levels(DepCat))[DepCat] # try ?facttonum   
 #converts replicates in DepCat to numbers, not just the levels   
5 \* DepCat[3] # now treat DepCat as numeric

[1] 2000

DepCat <- as.factor(rep(seq(300,600,50),2)); DepCat

[1] 300 350 400 450 500 550 600 300 350 400 450 500 550 600  
Levels: 300 350 400 450 500 550 600

facttonum(DepCat)

[1] 300 350 400 450 500 550 600 300 350 400 450 500 550 600

通常，获取 data.frame 的一个副本并在该副本中分解变量会更简单，因此，如果您想使用原始变量进行进一步的操作或绘图，您不需要将它们从数字转换为因子。

当数据被读入 R 中的 data.frame 时，默认的行为是将字符串或字符视为因子。您可以通过在控制台中输入 options() 来发现很多 R 选项的默认设置。如果你搜索 $stringsAsFactors (注意大写字母)，你会发现它默认为 TRUE。如果您希望关闭此功能，可以使用 option(stringsAsFactors=FALSE) 轻松完成。从关于因素的代码块中可以明显看出，在处理分类变量时需要格外小心。

### 2.2.10 数据输入

与所有编程语言(和统计包)一样，一个共同的需求是输入数据以便进行分析。R 非常擅长数据操作，这类事情在 Crawley (2007) 等书中有很好的描述。当然，也有许多处理数据操作的 R 包。一个特别有用的是 dplr，它是 tidyverse 组的一部分。但是在这里，我们只会提到几种将数据输入 r 的方法。一种非常常见和低级的方法是简单地使用文本文件作为输入。我仍然经常使用逗号分隔值 (.csv) 文件，因为它易于生成，而且可以从许多不同的编辑器打开和编辑。可以轻松使用 read.csv()或 read.table()。一旦数据被读入 R，就可以通过 save(object,file="filename.RData") 将其保存为更有效的存储格式，之后可以使用 load(file="filename.RData") 将其加载到R。查看 ?save 和 ?load 的细节。

在本书的示例中，我们将使用属于 **MQMF** 包一部分的数据集，但是通过复制这些数据集中使用的格式，一旦将数据加载到R中，您就可以使用自己的数据了。

## 2.3 编写函数

如果你打算使用 R 来进行任何分析，你通常会使用一个文本编辑器来输入你的 R 代码，并且至少保存这些脚本以备以后使用。在许多情况下，您可能需要重复一系列命令，这些命令可能使用的变量和或参数的输入值有所不同。尽管使用 source() 作为中间步骤并非不可能，但这种情况和许多其他情况最好的解决办法是将代码转换为函数。您将在下面的文本和 **MQMF** 包中看到许多函数示例，为了充分利用文本中表达的思想，您会发现学习如何编写自己的函数很有帮助。

函数的结构总是相同的:

#Outline of a function's structure   
functionname <- function(argument1, fun,...) {   
 # body of the function   
 #   
 # the input arguments and body of a function can include other   
 # functions, which may have their own arguments, which is what   
 # the ... is for. One can include other inputs that are used but   
 # not defined early on and may depend on what function is brought   
 # into the main function. See for example negLL(), and others   
 answer <- fun(argument1) + 2   
 return(answer)   
} # end of functionname   
functionname(c(1,2,3,4,5),mean) # = mean(1,2,3,4,5)= 3 + 2 = 5

[1] 5

我们将开发一个示例来说明如何将这些函数组合在一起。

### 2.3.1 简单函数

通常我们会有一个简单的公式，比如生长函数或者描述鱼类成熟度的函数。例如，von Bertalanffy 曲线在渔业研究中使用得相当多。

在R中 [公式 2.1](#eq-2_1) 可以以多种方式呈现,第一个简单的代码在一个代码块中，用年龄步长进行循环运算，或者矢量化,或者更合理,使用方程转化为一个函数,可以多次调用而不是复制显式行代码。

# Implement the von Bertalanffy curve in multiple ways   
ages <- 1:20   
nages <- length(ages)   
Linf <- 50; K <- 0.2; t0 <- -0.75   
 # first try a for loop to calculate length for each age   
loopLt <- numeric(nages)   
for (ag in ages) loopLt[ag] <- Linf \* (1 - exp(-K \* (ag - t0)))   
 # the equations are automatically vectorized so more efficient   
vecLt <- Linf \* (1 - exp(-K \* (ages - t0))) # or we can convert   
 # the equation into a function and use it again and again   
vB <- function(pars,inages) { # requires pars=c(Linf,K,t0)   
 return(pars[1] \* (1 - exp(-pars[2] \* (inages - pars[3]))))   
}   
funLt <- vB(c(Linf,K,t0),ages)   
ans <- cbind(ages,funLt,vecLt,loopLt)

Three different ways of generating the same growth curve

| ages | funLt | vecLt | loopLt |
| --- | --- | --- | --- |
| 1 | 14.77 | 14.77 | 14.77 |
| 2 | 21.15 | 21.15 | 21.15 |
| 3 | 26.38 | 26.38 | 26.38 |
| 4 | 30.66 | 30.66 | 30.66 |
| 5 | 34.17 | 34.17 | 34.17 |
| 6 | 37.04 | 37.04 | 37.04 |
| 7 | 39.39 | 39.39 | 39.39 |
| 8 | 41.31 | 41.31 | 41.31 |
| 9 | 42.89 | 42.89 | 42.89 |
| 10 | 44.18 | 44.18 | 44.18 |
| 11 | 45.23 | 45.23 | 45.23 |
| 12 | 46.10 | 46.10 | 46.10 |
| 13 | 46.80 | 46.80 | 46.80 |
| 14 | 47.38 | 47.38 | 47.38 |
| 15 | 47.86 | 47.86 | 47.86 |
| 16 | 48.25 | 48.25 | 48.25 |
| 17 | 48.56 | 48.56 | 48.56 |
| 18 | 48.82 | 48.82 | 48.82 |
| 19 | 49.04 | 49.04 | 49.04 |
| 20 | 49.21 | 49.21 | 49.21 |

使用向量化比使用循环更有效地编写代码，而且一旦人们习惯了这种想法，代码也会变得更易读。当然，可以直接复制进行计算的 R 的矢量化行，即 vecLt <- Linf \* (1 - exp(- k \* (ages - t0)))，而不是使用函数调用。但也可以在函数中包含错误检查和其他细节，使用函数调用应该有助于避免意外引入错误。此外，许多函数包含大量的代码行，因此使用函数调用也使整个程序更易于阅读和维护。

#A vB function with some error checking   
vB <- function(pars,inages) { # requires pars=c(Linf,K,t0)   
 if (is.numeric(pars) & is.numeric(inages)) {   
 Lt <- pars[1] \* (1 - exp(-pars[2] \* (inages - pars[3])))   
 } else { stop(cat("Not all input values are numeric! \n")) }   
 return(Lt)   
}   
param <- c(50, 0.2,"-0.75")   
funLt <- vB(as.numeric(param),ages) #try without the as.numeric   
halftable(cbind(ages,funLt))

ages funLt ages funLt  
1 1 14.76560 11 45.23154  
2 2 21.15251 12 46.09592  
3 3 26.38167 13 46.80361  
4 4 30.66295 14 47.38301  
5 5 34.16816 15 47.85739  
6 6 37.03799 16 48.24578  
7 7 39.38760 17 48.56377  
8 8 41.31130 18 48.82411  
9 9 42.88630 19 49.03726  
10 10 44.17579 20 49.21178

### 2.3.2 函数输入值

使用函数有很多优点。每个函数都有自己的分析环境，其主要优点是每个函数的环境都与函数外部的分析隔离开来。此外，全局环境(在其中完成许多工作)与不同函数中的工作是隔离的。因此，在函数中可以声明一个变量 *popnum*，并以多种方式更改它的值，但这对函数外同名的变量没有影响。这意味着一个函数的操作范围仅限于该函数。有一些方法可以让函数内部发生的事情直接影响外部变量，但我不会教人们坏习惯，我只会给出搜索符号的提示 ->>。忘记特殊的情况，一个函数与它发现自己所在的环境(一个人可以有函数，在函数中，在函数中，…) 交互的唯一方式是通过它的参数，以及通过它完成时的返回。在上面定义的 vB() 函数中，参数是 *pars* 和 *inages*，它显式地返回*Lt*。函数将返回上一次活动计算的结果，但在本书中，我们总是调用 return() 以确保返回的内容是显式和清晰的。有时，人们可以牺牲增加的简洁来增加清晰度。

在使用函数时，可以显式地使用参数名 vB(pars=param, inages=1:20) 或隐式地使用 vB(param,1:20)。如果隐式使用，则输入参数的顺序很重要。如果以错误的顺序输入,函数将愉快地使用 1:20 的前三个值作为参数，并将 param=c(50,0.02, 0.75) 作为所需的年龄(从 *VB(ages, param)* 获得以下预测长度: 1.0000 -269.4264 -1807.0424)。如果使用了参数名，那么顺序并不重要。输入所有这些名称可能会让人觉得需要付出更多的努力，但它最终确实使编程变得更容易，并且充当了另一个自我文档的来源。当然，应该开发一种适合自己的编程风格，自然地，可以从错误中学习。

从本质上讲，软件是顺从的。如果我们将 vB() 函数的参数前后混乱，如我们所见，它仍然会尽力返回一个答案或尝试失败。使用相对简单的软件和函数，不难确保任何这样的参数和数据输入以正确的顺序和可识别的形式呈现给函数。一旦我们开始生成更多的相互联系和复杂的软件，其中一个过程的输出形成了对其他过程的输入，那么所涉及的计划需要扩展到参数和数据使用的格式。错误检查就变得比只使用单个函数重要得多。

### 2.3.3 R对象

我们将R语言视为一种编程语言，在编写任何软件时都应该认真对待它的设计。只关注R，有两个原则可以帮助这样的设计 (Chambers 2017, p4) 。 1. 存在于R中的一切都是一个对象。

1. 在R中发生的一切都是一个函数调用。

这意味着任何函数输入的每个参数或形参和数据集都是R对象，就像函数(如果有的话)返回的都是R对象一样。

### 2.3.4 对象范围

您可能想知道为什么我在上面的 vB() 函数参数中使用 *inages* 而不是 *ages*。这纯粹是为了避免混淆每个变量的范围。如上所述，在R中，所有操作都发生在环境中，全局环境是在使用控制台或简单脚本时输入的环境。重要的是要认识到，您编写的任何函数都有自己的环境，其中包含内部变量的作用域。然而，即使 R 对象没有作为参数传递给函数，该对象仍然可以在函数中看到。但是，函数内部的工作对全局或调用环境是隐藏的。如果在函数内部定义了一个变量，但在函数外部使用了相同的变量名，那么当使用变量名时，它首先会在自己的环境中搜索，并使用内部定义的版本，而不是外部定义的版本。不过，最好的做法是在函数内部使用与函数外部不同的对象名称。

# demonstration that the globel environment is 'visible' inside a   
 # a function it calls, but the function's environment remains   
 # invisible to the global or calling environment   
vBscope <- function(pars) { # requires pars=c(Linf,K,t0)   
 rhside <- (1 - exp(-pars[2] \* (ages - pars[3])))   
 Lt <- pars[1] \* rhside   
 return(Lt)   
}   
ages <- 1:10; param <- c(50,0.2,-0.75)   
vBscope(param)

[1] 14.76560 21.15251 26.38167 30.66295 34.16816 37.03799 39.38760 41.31130  
 [9] 42.88630 44.17579

try(rhside) # note the use of try() which can trap errors ?try

Error in eval(expr, envir) : 找不到对象'rhside'

### 2.3.5 函数输入与输出

像大多数软件一样，R 函数总是有输入和输出(尽管两者可能都是 *NULL*；函数的括号可以为空，例如 parsyn())。当输入参数向量或数据矩阵时，采用标准格式是明智的，我的意思是设置标准格式。这样，任何处理此类数据的函数都可以对其格式进行假设。理想情况下，在使用任何输入数据之前进行数据检查仍然是一个好主意，但这可以通过遵循任何确定的标准格式来简化。

重要的是要知道和理解输入到每个函数中的数据的结构，因为向量繁殖需要以不同的方式引用到矩阵中，而矩阵的行为也与 data.frame 不同。通常最好通过列名显式地引用矩阵，因为不能保证所有这样的数据集都按照 *schaef* 中找到的顺序。我们可以比较 schaef[，2] 和 schaef[，"catch"]，和 schaef$catch。

#Bring the data-set schaef into the working of global environment   
data(schaef)

#examine the properties of the data-set schaef   
class(schaef)

[1] "data.frame"

a <- schaef[1:5,2]   
b <- schaef[1:5,"catch"]   
c <- schaef$catch[1:5]   
cbind(a,b,c)

a b c  
[1,] 60913 60913 60913  
[2,] 72294 72294 72294  
[3,] 78353 78353 78353  
[4,] 91522 91522 91522  
[5,] 78288 78288 78288

mschaef <- as.matrix(schaef)   
mschaef[1:5,"catch"] # ok

1934 1935 1936 1937 1938   
60913 72294 78353 91522 78288

d <- try(mschaef$catch[1:5]) #invalid for matrices

Error in mschaef$catch : $ operator is invalid for atomic vectors

d # had we not used try()eveerything would have stopped.

[1] "Error in mschaef$catch : $ operator is invalid for atomic vectors\n"  
attr(,"class")  
[1] "try-error"  
attr(,"condition")  
<simpleError in mschaef$catch: $ operator is invalid for atomic vectors>

通过测试 schaef 对象的*类*，我们可以看到它是 data.frame 而不是矩阵。因此，处理这些输入可能不像乍一看那么简单。是的，列的顺序可能会改变，但名称也可能不同。因此，我们至少要做一些检查，使我们的软件至少在一定程度上是安全的。通常我们是唯一使用我们编写的软件的人，但我发现这并不意味着我不需要尝试使它万无一失(似乎有时我是愚蠢的)。

注意，在 schaef 中，我们使用了所有小写的列标题，这并不总是常见的。正如我们所看到的，列名很重要，因为引用 data.frame 通常会更好，就像引用矩阵一样。通常，在编程时，我们可以使用不止一个选项。我们可以强制输入数据矩阵变成 data.frame，但也可以强制列名为小写。我知道列名已经是小写的了，但是这里我们讨论的是一个接收任意给定数据集的函数。

#Convert column names of a data.frame or matrix to lowercase   
dolittle <- function(indat) {   
 indat1 <- as.data.frame(indat)   
 colnames(indat) <- tolower(colnames(indat))   
 return(list(dfdata=indat1,indat=as.matrix(indat)))   
} # return the original and the new version   
colnames(schaef) <- toupper(colnames(schaef))   
out <- dolittle(schaef)   
str(out, width=63, strict.width="cut")

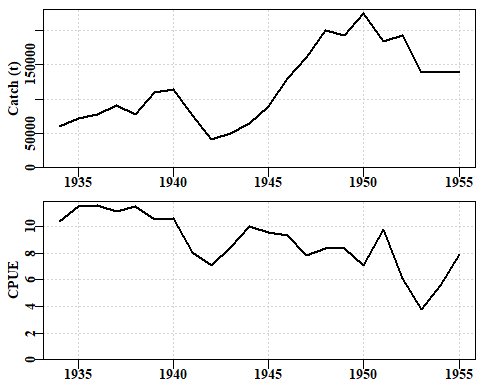
List of 2  
 $ dfdata:'data.frame': 22 obs. of 4 variables:  
 ..$ YEAR : int [1:22] 1934 1935 1936 1937 1938 1939 1940 1..  
 ..$ CATCH : int [1:22] 60913 72294 78353 91522 78288 110417..  
 ..$ EFFORT: int [1:22] 5879 6295 6771 8233 6830 10488 10801..  
 ..$ CPUE : num [1:22] 10.4 11.5 11.6 11.1 11.5 ...  
 $ indat : num [1:22, 1:4] 1934 1935 1936 1937 1938 ...  
 ..- attr(\*, "dimnames")=List of 2  
 .. ..$ : chr [1:22] "1934" "1935" "1936" "1937" ...  
 .. ..$ : chr [1:4] "year" "catch" "effort" "cpue"

因此，在接下来的内容中，我们将尝试仅对与渔业相关的数据使用小写列名，我们将使用标准名称 “year”、“catch”、“cpue” 等，但我们也将在期望输入数据矩阵的函数中包含一条转换为 *tolower* 的语句。当输入包含与渔业相关的数据矩阵、年龄组成数据、与生长、成熟、补充和选择性有关的生物常数向量以及其他数据的更复杂列表对象时，这些细节就变得更加重要。在任何稍微复杂一点的软件设计中，都应该确定标准格式并检查此类输入。

在 schaef 的情况下，这是在评估中与剩余产量模型一起使用的数据。这是合理的想法开发函数提前预支了一个先于它是否包含所有所需的数据,在数据是否有差距,和其他细节,之前用于评估一个甚至可以开发*S3*类为特定分析,这将允许将函数添加到通用函数，如 *print*、*plot*、*summary*，以及创建自己的新泛型方法函数。

另外，可以简单地编写函数来为特定的分析生成可定制的图，以执行可能重复多次的特定任务。可以通过函数 plotspmdat() 找到一个示例，该函数将获取典型的渔业数据，并绘制渔获量和 cpue 的时间序列。

#Could have used an S3 plot method had we defined a class Fig.2.2   
plotspmdat(schaef) # examine the code as an eg of a custom plot



The catch and cpue data from the MQMF data-set schaef plotted automatically.

## 2.4 附录：少有人使用的函数

下面列出的函数只是为了帮助读者记住R中一些容易被遗忘的函数，因为它们很少被使用，尽管它们在处理实际问题时非常有用。

getNamespaceExports("MQMF")

getS3method("print","default") 通常 print.default 也可以

ls(“package:MQMF”) 列出包中的输出函数

ls.str(“package:MQMF”)列出函数及其语法

methods(“print”)列出适用的S3方法

on.exit(suppressWarnings(sink())) 退出运行的清洗方法（即使失败）

packageDescription("MQMF")

RNGkind()正在使用的随机数生成器

sessionInfo()

sink("filename",split=TRUE) then after lots of screen output sink(),

suppressWarnings(sink()) removes a sink quietly if a run crashes and leaves one open

## 2.5 附录：其它学习资源

Other Books: these two have a tidyverse bent.

Wickham and Grolemund (2017) *R for Data Science* <https://r4ds.had.co.nz/>

Grolemund (2014) *Hands-On Programming with R* <https://rstudio-education.github.io/hopr/>

There are blogs devoted to R. I rather like <https://www.r-bloggers.com/> but I get that delivered to my email. This blog lists many others.

There is even an open-source journal concerning R

<https://journal.r-project.org/>

# 3. 简单种群模型

## 3.1 简介

简单的种群模型构成了众多渔业模型的基础，事实上，渔业模型与生态种群模型的区别只是人为的划分。理解任何种群模型的运行方式都将有助于更好地理解渔业模型，因此在本章中，我们将使用R来检验种群生物学中的一些基本思想的动力学。

### 3.1.1 离散Logistic模型

我们将首先考察已成为生态学中经典种群模型之一的离散逻辑斯蒂模型所展示的动力学。这种模型出现在许多地方，以许多不同的形式出现，因此值得进行一些详细的探索。这里我们将使用 (M. Schaefer 1991) Schaefer(1954, 1957)版本，它有四个组成部分或项:

项为时间 时的数量（如年 的开始时），最后一项为 ，它是在时间 内（如年份 ）捕获的渔获量。第一行反映了初始状态，定义为时间 时的数量（或生物量）。要注意下标，因为无论是数量还是生物量，它们都是指特定时间的数量和生物量，通常是确定的时间段 的开始时的数量或生物量，而渔获量指的是整个时间段 内的渔获量。这个方程是对时间步长的动力学的总结； 它是一个差分方程而不是连续微分方程。第二个方程的中间项是更复杂的组成部分，该项定义了所谓的渔业生产曲线。它由 和 组成， 为种群的内在增长率， 为承载力或最大种群规模。如果动力学表示为在开发开始之前处于平衡状态，那么 会取与 相同的值，但它是分开的，以考虑任何时间序列开始时偏离均衡的情况。在 **MQMF** 的 discretelogistic() 中，你会看到另一个参数 ，这个参数将在“剩余产量模型”中详细探讨。在此，只需说明的是，当使用默认值 时， [公式 3.1](#eq-3_1) 中的方程等同于 Schaefer 模型。这里，Schaefer 模型输出实时数量，尽管在渔业中，它通常直接用实时生物量（）表示。这些术语可以以这种方式改变，这一事实强调了该模型忽略了生物学实际和属性，如种群体长结构和年龄结构，以及性别和其他属性之间的差异。令人惊讶的是，这样一个简单的模型在理论和实践中仍然是有用的。

我们可以通过将式 [公式 3.1](#eq-3_1) 的中间项与 值的向量绘制成曲线来说明产量曲线。子项 表示无限制的种群指数增长，在这个离散形式的方程中，并将此项与 相加。因此，只要 及渔获量为0，我们就可以期待从方程的 部分得到种群的无尽增长。然而,子项 为抑制指数增长的项，而且随着种群规模的增长，这种作用越来越大。这就是所谓的密度依赖效应。

surprod <- function(Nt,r,K) return((r\*Nt)\*(1-(Nt/K)))   
densdep <- function(Nt,K) return((1-(Nt/K)))   
r <- 1.2; K <- 1000.0; Nt <- seq(10,1000,10)   
par(mfrow=c(2,1),mai=c(0.4,0.4,0.05,0.05),oma=c(0.0,0,0.0,0.0))   
par(cex=0.75, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
plot1(Nt,surprod(Nt,r,K),xlab="Population Nt",defpar=FALSE,   
 ylab="Production")   
plot1(Nt,densdep(Nt,K),xlab="Population Nt",defpar=FALSE,   
 ylab="Density-Dependence")

|  |
| --- |
| 图 3.1: 离散 Schaefer（1954，1957）剩余生产模型的完整生产曲线以及与种群大小相当的等效线性密度依赖关系。 |

密度依赖是种群规模调节的重要组成部分 (May 和 McLean 2007)。从 [图 3.1](#fig-3-1) 可以明显看出，Schaefer 模型中的密度依赖项与种群规模呈线性递减关系。这就是”线性密度依赖”一词的字面意思，你可以在种群动态文献中找到它。这个密度依赖项，试图补偿由 驱动的指数增长 ，限制种群规模不大于未捕捞种群规模的长期平均(关于补偿不足时会发生什么，请参阅下一节)。

### 3.1.2 动态行为

众所周知，微分方程形式的 Schaefer 模型一直非常稳定，最大种群数量真正受限于 值。令人惊讶的是，人们发现离散形式的动力学更为复杂和有趣 (May, 1973，1976)。我们已经在函数 discretelogistic() 中实现了上述方程的一种形式，你可以通过 *?discrete telogistic* 读取它的帮助文件，并且可以通过在控制台输入 discretelogistic 来查看它的代码(注意，没有括号)。如果您检查这个函数中的代码，您将看到它使用 *for* 循环来逐年间计算每年的资源量。我希望，作为一名 R 程序员，您会想知道为什么我不使用向量化方法来避免使用 *for* 循环。这突出了种群动力学的一个基本方面，即 时的数量总是 时数量的函数(初始条件可能于 参数)。种群动态的顺序性质意味着我们不能以任何有效的方式使用矢量化。种群动态的顺序性是如此明显的事实，以至于它施加在种群模型上的约束和结构常常被忽视。

discretelogistic() 函数（隐形）输出时间矩阵， （用 *year* 表示）、 时的数量 （表示为 ）以及 时的数量 （表示为 ）的矩阵。该矩阵也给出 **MQMF** 中包含的了 *dynpop* 类和S3方法（*plot.dynpop*；见*A non-introduction to R* 一章S3类和方法的使用。如果要查看代码，在控制台中输入*MQMF:::plot.dynpop*，记住当中是3个”:“，而非2个）。当然，可以写任何想写的东西。试着将rv的值 设定为0.5、1.95、2.2、2.475、2.56和2.8，用这些值按顺序运行下面的代码。

如果你运行生成 [图 3.2](#fig-3-2) 的代码，使用上面提到的6个rv值，你应该会看到一个单调阻尼平衡(一个实心点)，然后是一个阻尼振荡平衡(仍然是一个实心点)，然后是介于阻尼振荡平衡和两环稳定极限环之间的东西(有点涂抹了两个实心点)。当 时，有一个 4 圈稳定的极限环( 4 个实心点)，当 时，有一个 8 圈稳定的极限环。最后，当 时，动态以混乱告终，尽管所有的点都落在如图所示的抛物线上，但每一步的结果都不是随机的，也不是直接可预测的。在混沌理论中，这个抛物线被称为奇怪的吸引子，尽管这是一个非常简单的吸引子。这种混沌系统也依赖于起始条件。如果把初始数字改为 101 而不是 100，数字的时间线就会发生根本性的变化，尽管抛物线保持不变。当然，我们可以输入任意范围的数字，也可以使用稍微不同的模型来改变动态。虽然混乱很有趣，玩起来也很有趣 (Gleick，1988；Lauwerier, 1991)，但在模拟自然种群的实际应用中似乎很少有用。事实上，一些人声称，他们观察到的明显的随机性是由种群对随机环境效应的反应引起的，而不是由密度依赖模型中的补偿不足和补偿过度引入的非线性（Higgins等，1997）。

#Code for Figure 3.2. Try varying the value of rv from 0.5-2.8   
yrs <- 100; rv=2.8; Kv <- 1000.0; Nz=100; catch=0.0; p=1.0   
ans <- discretelogistic(r=rv,K=Kv,N0=Nz,Ct=catch,Yrs=yrs,p=p)   
avcatch <- mean(ans[(yrs-50):yrs,"nt"],na.rm=TRUE) #used in text   
label <- paste0("r=",rv," K=",Kv," Ct=",catch, " N0=",Nz," p=",p=p)   
plot(ans, main=label, cex=0.9, font=7) #Schaefer dynamics

|  |
| --- |
| 图 3.2: Schaefer 模型动态。左图是按年份的数值，展示了从 r 值为 2.8 开始的混沌动态。右图是时间 t+1 的数值与时间 t 的关系，称为相图。最后 20%的点用红色标出以展示任何平衡行为。灰色对角线是 1:1 线。 |

在实际应用中，如果发现模型表现出明显的混沌或周期性行为，甚至是负的种群数量，尝试确定所观察到的现象是真实的生物群体行为(显然不是负数)，还是更简单地说，只是一个特殊参数的数学表达式。人们总是可以通过在计算中包含诸如 之类以避免模型向下跌落到0以下（参见 discretelogistic() 代码。类似地，用 以避免计算值高于），尽管对代码的此类添加仍然是临时的（应该，比如，任何超过的值都应允许，是 ，还是 ？有些种群自然变动的）。在第 [7](#sec-surplusproduction) 章 “[剩余产量模型](07-spm.qmd)” 一章中我们将研究惩罚函数的使用，惩罚函数可用于防止在估计这些参数时产生异常值，同时避免使用，如 这类粗糙的工具。

我们可以通过将 discretelogistic() 函数的不可见输出转换为可以使用的对象来检查每年的实际资源量（[表 3.1](#tbl-3-1)）。

#run discrete logistic dynamics for 600 years   
yrs=600   
ans <- discretelogistic(r=2.55,K=1000.0,N0=100,Ct=0.0,Yrs=yrs)

# layout-ncol: 1  
  
kable(halftable(ans[(yrs-29):yrs,],yearcol="year",subdiv=3),  
 digits=c(0,1,1,0,1,1,0,1,1))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.1: ans 对象中最后 30 个按年排列的数字，展示了在 r = 2.2 时呈现 8 周期稳定极限。如果你在 discretelogistic() 中尝试 r = 2.2，它将导致 2 周期稳定极限在 746.24 和 1162.84 之间波动。尝试使用 plot(ans) 并观察每次变化。   | year | nt | nt1 | year | nt | nt1 | year | nt | nt1 | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 571 | 493.9 | 1131.3 | 581 | 752.4 | 1227.4 | 591 | 515.6 | 1152.4 | | 572 | 1131.3 | 752.4 | 582 | 1227.4 | 515.6 | 592 | 1152.4 | 704.5 | | 573 | 752.4 | 1227.4 | 583 | 515.6 | 1152.4 | 593 | 704.5 | 1235.4 | | 574 | 1227.4 | 515.6 | 584 | 1152.4 | 704.5 | 594 | 1235.4 | 493.9 | | 575 | 515.6 | 1152.4 | 585 | 704.5 | 1235.4 | 595 | 493.9 | 1131.3 | | 576 | 1152.4 | 704.5 | 586 | 1235.4 | 493.9 | 596 | 1131.3 | 752.4 | | 577 | 704.5 | 1235.4 | 587 | 493.9 | 1131.3 | 597 | 752.4 | 1227.4 | | 578 | 1235.4 | 493.9 | 588 | 1131.3 | 752.4 | 598 | 1227.4 | 515.6 | | 579 | 493.9 | 1131.3 | 589 | 752.4 | 1227.4 | 599 | 515.6 | 1152.4 | | 580 | 1131.3 | 752.4 | 590 | 1227.4 | 515.6 | 600 | 1152.4 | 704.5 | |

当我们设置 值来产生2循环的稳定限制(例如 )，可以从其他年份的数值重复中清楚地看到。检查实际的数值显然比相信图表的视觉印象更准确。这也意味着我们可以尽可能详细地查看结果。在上述情况下，最近50年的平均渔获量为859.675，明显小于1000的值。由于 与 的相位关系，上一年的最后一行通常不会有 的值，因此，在代码中，输入的年份数增加1，以便最终生成 (请参阅代码)。

### 3.1.3 确定行为间的边界

我们可以通过求 ans 中 和 列的行平均值来找到一个稳定的限制循环，查看最后 100 个值(四舍五入到小数点后三位)。table() 函数的名称标识循环点的值(如果有的话)。如果只有一个或两个平均值来确定一个渐近平衡或一个两周期稳定的限制循环。通过检查每个 ans 对象中值的时间序列，可以搜索所标识值的首次出现，从而确定循环行为（精确到小数点后三位）何时首次出现。我们将这些值四舍五入，并使用 600 年或更长的时间，因为如果我们使用所有 15 位小数，任何超过 8 的循环都可能无法被清楚地识别出来（尝试将 r = 2.63 更改为 5；并试着用 plot(ans) 绘图）。

#run discretelogistic and search for repeated values of Nt   
yrs <- 600   
ans <- discretelogistic(r=2.55,K=1000.0,N0=100,Ct=0.0,Yrs=yrs)   
avt <- round(apply(ans[(yrs-100):(yrs-1),2:3],1,mean),2)   
count <- table(avt)   
count[count > 1] # with r=2.55 you should find an 8-cycle limit

avt  
812.64 833.99 864.65 871.5 928.45 941.88 969.92 989.93   
 12 13 12 13 12 13 12 13

我们可以建立一个程序来寻找 的值生成不同周期的稳定限制环，尽管下面的代码只是部分成功。通过设置一个 *for* 循环，将不同的值替换为 值输入到 discretelogistic()，我们可以搜索一个长时间序列的最后几年在数值的时间序列中唯一的值。然而，四舍五入误差可能会导致意想不到的结果，特别是在不同类型的动态行为之间的边界。我们可以通过将检查的数值四舍五入到小数点后三位来避免这些问题，但请尝试在下面的代码中散列该行，以查看问题变得更糟。

#searches for unique solutions given an r value see Table 3.2  
testseq <- seq(1.9,2.59,0.01)   
nseq <- length(testseq)   
result <- matrix(0,nrow=nseq,ncol=2,   
 dimnames=list(testseq,c("r","Unique")))   
yrs <- 600   
for (i in 1:nseq) { # i = 31   
 rval <- testseq[i]   
 ans <- discretelogistic(r=rval,K=1000.0,N0=100,Ct=0.0,Yrs=yrs)   
 ans <- ans[-yrs,] # remove last year, see str(ans) for why   
 ans[,"nt1"] <- round(ans[,"nt1"],3) #try hashing this out   
 result[i,] <- c(rval,length(unique(tail(ans[,"nt1"],100))))   
}

在上面的例子中，在 值的范围内，最近的100次观测中，最多可以产生60个唯一值。这些代表了计算机上舍入误差的问题。如果持续运行的年数显著增加，那么预期最终会出现一种平衡。无论哪种情况，都存在一个明显的分裂或过渡，从单一平衡点，到2循环，到4循环，最后到8循环稳定极限。在行为的每个转变中，由于舍入误差，结果有一些不稳定性。

kable(halftable(result,yearcol = "r", subdiv=3),digits=c(0,1,1,0,1,1,0,1,1))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.2: Schaefer 模型动态。结果内容：在 600 年的预测动态中，从不同 r 值获得的 60 年内独特种群数量（N）。100 表示非平衡甚至混沌。   | r | Unique | r | Unique | r | Unique | | --- | --- | --- | --- | --- | --- | | 2 | 1 | 2.1 | 2 | 2.4 | 2 | | 2 | 1 | 2.1 | 2 | 2.4 | 2 | | 2 | 1 | 2.2 | 2 | 2.4 | 2 | | 2 | 1 | 2.2 | 2 | 2.4 | 2 | | 2 | 1 | 2.2 | 2 | 2.4 | 2 | | 2 | 1 | 2.2 | 2 | 2.4 | 2 | | 2 | 1 | 2.2 | 2 | 2.4 | 4 | | 2 | 1 | 2.2 | 2 | 2.5 | 100 | | 2 | 7 | 2.2 | 2 | 2.5 | 4 | | 2 | 100 | 2.2 | 2 | 2.5 | 4 | | 2 | 100 | 2.2 | 2 | 2.5 | 4 | | 2 | 4 | 2.2 | 2 | 2.5 | 4 | | 2 | 2 | 2.3 | 2 | 2.5 | 4 | | 2 | 2 | 2.3 | 2 | 2.5 | 4 | | 2 | 2 | 2.3 | 2 | 2.5 | 4 | | 2 | 2 | 2.3 | 2 | 2.5 | 4 | | 2 | 2 | 2.3 | 2 | 2.5 | 4 | | 2 | 2 | 2.3 | 2 | 2.5 | 8 | | 2 | 2 | 2.3 | 2 | 2.6 | 8 | | 2 | 2 | 2.3 | 2 | 2.6 | 100 | | 2 | 2 | 2.3 | 2 | 2.6 | 100 | | 2 | 2 | 2.4 | 2 | 2.6 | 100 | | 2 | 2 | 2.4 | 2 | NA | NA | | 2 | 2 | 2.4 | 2 | NA | NA | |

### 3.1.4 经典的混沌分岔图

May(1976) 对基于差分方程的单物种种群模型的动力学进行了综述。文中，他从非常简单的模型描述了混沌动力学的潜力。在那篇文章中，他绘制了一个图，他称之为分岔图（bifurcation diagram），该图显示了模型的平衡属性如何随着单个参数的改变而改变。我们可以使用discretelogistic()函数和下面的代码，以几乎任何精度复制 [图 3.3](#fig-3-3) 。

在分岔图的混沌中，偶尔会有一些简化，由图中的白色间隙表示出来。可以通过替换 *testseq* 向量中的数值来检查其中的下限位置，如下限值=2.82，上限= 2.87和inc = 0.0001 。这些起始点提供了更多的细节，在这些细节中有以前被掩盖的细节。这些也可以通过改变数值*testseq < - c(2.845, 2.855, 0.00001)* 检查，于是分岔,2 - 4 -和更高的周期可以看到(甚至可以看到更多的细节通过改变*limy= 0* 参数绑定y-轴在 内感兴趣的区域绑定y轴所需的范围。这个图是通过在每个r的序列值处使用最终尾<- 100点绘制的。可以增加模拟的年数和绘制的最终点的数量，以获得更好的分辨率在混沌区域。要查看更多细节，请在需要细节的区域扩展规模。

#the R code for the bifurcation function   
bifurcation <- function(testseq,taill=100,yrs=1000,limy=0,incx=0.001){   
 nseq <- length(testseq)   
 result <- matrix(0,nrow=nseq,ncol=2,   
 dimnames=list(testseq,c("r","Unique Values")))   
 result2 <- matrix(NA,nrow=nseq,ncol=taill)   
 for (i in 1:nseq) {   
 rval <- testseq[i]   
 ans <- discretelogistic(r=rval,K=1000.0,N0=100,Ct=0.0,Yrs=yrs)   
 ans[,"nt1"] <- round(ans[,"nt1"],4)   
 result[i,] <- c(rval,length(unique(tail(ans[,"nt1"],taill))))   
 result2[i,] <- tail(ans[,"nt1"],taill)   
 }   
 if (limy[1] == 0) limy <- c(0,getmax(result2,mult=1.02))   
 parset() # plot taill values against taill of each r value   
 plot(rep(testseq[1],taill),result2[1,],type="p",pch=16,cex=0.1,   
 ylim=limy,xlim=c(min(testseq)\*(1-incx),max(testseq)\*(1+incx)),   
 xlab="r value",yaxs="i",xaxs="i",ylab="Equilibrium Numbers",   
 panel.first=grid())   
 for (i in 2:nseq)   
 points(rep(testseq[i],taill),result2[i,],pch=16,cex=0.1)   
 return(invisible(list(result=result,result2=result2)))   
} # end of bifurcation

#Alternative r value arrangements for you to try; Fig 3.3   
 #testseq <- seq(2.847,2.855,0.00001) #hash/unhash as needed   
 #bifurcation(testseq,limy=c(600,740),incx=0.0001) # t   
 #testseq <- seq(2.6225,2.6375,0.00001) # then explore   
 #bifurcation(testseq,limy=c(660,730),incx=0.0001)   
testseq <- seq(1.9,2.975,0.0005) # modify to explore   
bifurcation(testseq,limy=0)

|  |
| --- |
| 图 3.3: Schaefer 模型动态。一个经典的分岔图（May，1976），绘制了平衡动态与 r 值的函数关系，展示了从 2-、4-、8-循环以及混沌行为的转变。 |

### 3.1.5 捕捞对种群动态的影响

如果我们设置一个默认的 discretelogisitic() 运行，以在渔获量为零时拥有8个周期的稳定极限，我们可以应用不同水平的恒定渔获量，看看这可能如何影响动态。在这里，我们应用了 0,50,200 和 325 个个体(尽管 ， 和 也可能是吨)。当然是初始值 必须大于捕获值，否则可以自由更改值。在下面四种情况中，我们假设 值为 1.0 (= Schaefer模型)时，可以尝试 探讨使用与 Fox 模型等价的方法所产生的差异。

#Effect of catches on stability properties of discretelogistic   
yrs=50; Kval=1000.0   
nocatch <- discretelogistic(r=2.56,K=Kval,N0=500,Ct=0,Yrs=yrs)   
catch50 <- discretelogistic(r=2.56,K=Kval,N0=500,Ct=50,Yrs=yrs)   
catch200 <- discretelogistic(r=2.56,K=Kval,N0=500,Ct=200,Yrs=yrs)   
catch300 <- discretelogistic(r=2.56,K=Kval,N0=500,Ct=300,Yrs=yrs)

我们组合了一个小函数 plottime()，以简化最终对象的重复绘图。我们还使用了效用函数 parsyn() 来辅助 par() 函数的语法，以定义基本图形图的边界。请注意，生成这样的函数可以简化后续的代码，我建议构建这样一个有用的函数集合，可以与函数 source() 一起使用，将它们引入到自己的会话中(就像预包;不要忘记在编写函数时记录文档，RStudio 甚至有一个插入 Roxygen 骨架命令)。在这里，随着捕获量的增加，可以看到动态开始稳定 4 到 2 周期，然后渐近平衡。注意，随着渔获量的增加，随着时间的推移(图表底部的数字)，平均渔获量下降。

#Effect of different catches on n-cyclic behaviour Fig3.4   
plottime <- function(x,ylab) {   
 yrs <- nrow(x)   
 plot1(x[,"year"],x[,"nt"],ylab=ylab,defpar=FALSE)   
 avB <- round(mean(x[(yrs-40):yrs,"nt"],na.rm=TRUE),3)   
 mtext(avB,side=1,outer=F,line=-1.1,font=7,cex=1.0)   
} # end of plottime   
 #the oma argument is used to adjust the space around the graph   
par(mfrow=c(2,2),mai=c(0.25,0.4,0.05,0.05),oma=c(1.0,0,0.25,0))   
par(cex=0.75, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
plottime(nocatch,"Catch = 0")   
plottime(catch50,"Catch = 50")   
plottime(catch200,"Catch = 200")   
plottime(catch300,"Catch = 300")   
mtext("years",side=1,outer=TRUE,line=-0.2,font=7,cex=1.0)

|  |
| --- |
| 图 3.4: Schaefer 模型动态。左上角的图描绘了未捕捞的预期动态（8 周期），而接下来的三个图，随着平均捕捞量的减少，说明了捕捞量增加而其他因素不变的影响。 |

我们可以对四个相图做一些类似的事情。这里，我们从 MQMF:::plot.dynpop() 函数中借用了部分代码，以定义一个小型定制函数，以方便一起绘制四个阶段的图。

#Phase plot for Schaefer model Fig 3.5   
plotphase <- function(x,label,ymax=0) { #x from discretelogistic   
 yrs <- nrow(x)   
 colnames(x) <- tolower(colnames(x))   
 if (ymax[1] == 0) ymax <- getmax(x[,c(2:3)])   
 plot(x[,"nt"],x[,"nt1"],type="p",pch=16,cex=1.0,ylim=c(0,ymax),   
 yaxs="i",xlim=c(0,ymax),xaxs="i",ylab="nt1",xlab="",   
 panel.first=grid(),col="darkgrey")   
 begin <- trunc(yrs \* 0.6) #last 40% of yrs = 20, when yrs=50   
 points(x[begin:yrs,"nt"],x[begin:yrs,"nt1"],pch=18,col=1,cex=1.2)   
 mtext(label,side=1,outer=F,line=-1.1,font=7,cex=1.2)   
} # end of plotphase   
par(mfrow=c(2,2),mai=c(0.25,0.25,0.05,0.05),oma=c(1.0,1.0,0,0))   
par(cex=0.75, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
plotphase(nocatch,"Catch = 0",ymax=1300)   
plotphase(catch50,"Catch = 50",ymax=1300)   
plotphase(catch200,"Catch = 200",ymax=1300)   
plotphase(catch300,"Catch = 300",ymax=1300)   
mtext("nt",side=1,outer=T,line=0.0,font=7,cex=1.0)   
mtext("nt+1",side=2,outer=T,line=0.0,font=7,cex=1.0)

|  |
| --- |
| 图 3.5: Schaefer 模型动态。四个持续捕捞场景的相图。 |

随着固定捕获量的增加，隐含抛物线曲线(奇异吸引子)的最大值下降。渔获量( 捕食者)似乎对种群的动态有稳定的影响。在自然界中是否如此，将取决于特定种群的生物学是否反映了用于描述动态的模型的基本假设(例如，Schaefer 模型假设种群中存在线性密度依赖)。

现在，我们应该很清楚，即使是简单的模型也可以表现出复杂的行为。我们只是稍微触及了这个主题，但同样清楚的是，使用 R，可以探索这些想法到任何深度。

### 3.1.6 决定论

本章所有的模型都表现出确定性动力学，即使是混沌动力学，在初始条件重复的情况下也会遵循相同的轨迹。当然，在自然界中，这种确定性行为是罕见的。模型总是对可能发生的事情进行抽象。与生产力相关的自然过程(自然死亡率、生长和繁殖)都将随着时间的推移而变化，这是对食物供应、生态相互作用和环境影响(如温度)的响应。通过包括所谓的过程误差，可以将过程中的这种可变性近似地包括到模型中，过程误差可以用方程形式表示为:

其中 和 表示每个模型参数随时间 变化的正态随机误差。有时，在模拟中，只有一个参数会添加变化。作为种群现象，这将导致种群数量遵循不一定是平滑的轨迹，而且在模拟中，单个复制将不足以捕捉可能的动态的潜在范围。在模拟中是否包括这种过程误差将取决于模拟的目的。在这里，目的是探索确定性种群动态的期望，但应该始终记住，这些模型只是近似，旨在代表平均行为。

## 3.2 年龄结构的建模概念

当然，在我们刚刚考察的剩余产量模型中，最大的简化是，它们忽略了与年龄和个体大小有关的生物学和行为差异的所有方面。这种简化的模型仍然有用(见第 [7](#sec-surplusproduction) 章“[剩余产量模型](07-spm.qmd)”（*Surplus Production Models*） )，但面对公认的网具选择性(类似于捕食者有偏好的猎物)和不同年龄的生长和繁殖力、渔业和其他自然资源科学家经常在他们的模型中包括年龄和或大小。在这里，我们将只对这些想法进行初步处理，因为在详细讨论这些复杂问题之前，还需要建模动态过程的许多其他细节。

### 3.2.1 世代中的残存

很明显，个体的残存，，两个时间段之间的残存可用 时的数量与 时的数量的比值表示(记住，如果知道体重与年龄的关系，年龄-数量可以转换为年龄-体质量或年龄-生物量):

这种随时间的比例减少暗示着负指数增长。给定瞬时总死亡率为 (自然死亡率和捕捞死亡率之和)，则每年的残存率(即存活的比例)等于 。这可以通过设置一个具有恒定初始条件的模拟种群，并应用不同瞬时总死亡率向量来说明。

这将相当于考虑一个单一世代补充的为 0+ 年龄的个体。假设没有移入个体，这样的单一世代种群只能在数量上下降，尽管这看起来很明显，但在考虑年龄结构的种群时，这是重要的直觉。总死亡率(通常用 表示)仅仅是自然死亡率(通常用 表示)和捕捞死亡率(通常用 表示)的总和。在下面的例子中，请注意，应用 会导致即使在 50 年后种群数量也大于零。图示中的每个世代轨迹都描述了种群规模的指数或恒定比例下降。

#Exponential population declines under different Z. Fig 3.6   
yrs <- 50; yrs1 <- yrs + 1 # to leave room for B[0]   
years <- seq(0,yrs,1)   
B0 <- 1000 # now alternative total mortality rates   
Z <- c(0.05,0.1,0.2,0.4,0.55)   
nZ <- length(Z)   
Bt <- matrix(0,nrow=yrs1,ncol=nZ,dimnames=list(years,Z))   
Bt[1,] <- B0   
for (j in 1:nZ) for (i in 2:yrs1) Bt[i,j] <- Bt[(i-1),j]\*exp(-Z[j])   
plot1(years,Bt[,1],xlab="Years",ylab="Population Size",lwd=2)   
if (nZ > 1) for (j in 2:nZ) lines(years,Bt[,j],lwd=2,col=j,lty=j)   
legend("topright",legend=paste0("Z = ",Z),col=1:nZ,lwd=3,   
 bty="n",cex=1,lty=1:5)

|  |
| --- |
| 图 3.6: 不同总死亡率水平下的指数型种群衰退。最上方的曲线是 Z = 0.05，最陡峭的曲线是 Z = 0.55。 |

### 3.2.2 瞬时 Vs 年死亡率

在前一节中，我们讨论了所谓的瞬时率。它们可以用来生成残存率，即在给定瞬时死亡率的一年后预期残存的比例。瞬时死亡率的概念是指在无限小的时间段内的死亡率。

我们已经知道每年残存的比例:

因此1年时间死亡率（ ）表示为：

如果只考虑瞬时捕捞死亡率 ，而忽略自然死亡率，则死亡的比例称为捕获率，。

如果知道了每年的捕获率，就可以将其反向转换来估计瞬时捕捞死亡率。

其中 log 是自然对数或以 为底的对数。

有时似乎很难直观地理解瞬时速率的含义。如果举例说明瞬时率是如何推导出来的，这个困难就会减轻。从 1000 个个体的种群开始，假设每年的移除率为 ，这意味着只有 500 个个体能存活一年。将年总死亡率代入最后一个方程，可估计瞬时总死亡率为 。如果我们使用两个 6 个月周期来近似这个水平所需的时间步长，以获得 50% 的存活率，我们需要将 0.693147 除以 2.0，并将死亡率用 2 次。类似地，如果时间步长是每月的，我们将除以 12，并将结果应用 12 次。虽然很难将每个月称为瞬时，但这将是适用近似瞬时死亡率 (0.693147/12) 的期间。这种重新缩放可以在越来越小的时间间隔中发生，直到时间间隔足够小，结果与预期的 500 个年终剩余个体没有显著差异（ [表 3.3](#tbl-3-3) ）。

显而易见，应用近似瞬时速率的时间间隔越短，最终值就越接近预期的1000的0.5(即500)。希望这些代码能够帮助您对瞬时死亡率概念有直观的了解。

可以根据瞬时死亡率绘制年死亡率图，以说明两者之间的差异。请注意，瞬时死亡率大约 为 0.18 时，年收捕获 率也大致相同。应该清楚的是，不可能获得大于 1.0 的捕获率(相当于捕获>100%的可开发生物量)。高的捕获率开始显得难以置信，如果它们出现在适合渔业模式的情况下，那么在被接受之前，它们确实需要得到保护。另一方面，瞬时率显然可以超过 1.0，但理想情况下，这样的事件不应再被误解。

#Prepare matrix of harvest rate vs time to appoximate F   
Z <- -log(0.5)   
timediv <- c(2,4,12,52,365,730,2920,8760,525600)   
yrfrac <- 1/timediv   
names(yrfrac) <- c("6mth","3mth","1mth","1wk","1d","12h",   
 "3h","1h","1m")   
nfrac <- length(yrfrac)   
columns <- c("yrfrac","divisor","yrfracH","Remain")   
result <- matrix(0,nrow=nfrac,ncol=length(columns),   
 dimnames=list(names(yrfrac),columns))   
for (i in 1:nfrac) {   
 timestepmort <- Z/timediv[i]   
 N <- 1000   
 for (j in 1:timediv[i]) N <- N \* (1-timestepmort)   
 result[i,] <- c(yrfrac[i],timediv[i],timestepmort,N)   
}

kable(result,digits=c(10,0,8,4))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.3: 将恒定的总死亡率分配到越来越短的时间段内，且这些时间段总和为一年时的结果。yrfrac 是一年的分数，Approx 是瞬时总死亡率除以除数，Remain 是一年后剩余的动物数量（从 1000 开始计算）。   |  | yrfrac | divisor | yrfracH | Remain | | --- | --- | --- | --- | --- | | 6mth | 0.5000000000 | 2 | 0.34657359 | 426.9661 | | 3mth | 0.2500000000 | 4 | 0.17328680 | 467.1104 | | 1mth | 0.0833333333 | 12 | 0.05776227 | 489.6953 | | 1wk | 0.0192307692 | 52 | 0.01332975 | 497.6748 | | 1d | 0.0027397260 | 365 | 0.00189903 | 499.6706 | | 12h | 0.0013698630 | 730 | 0.00094952 | 499.8354 | | 3h | 0.0003424658 | 2920 | 0.00023738 | 499.9589 | | 1h | 0.0001141553 | 8760 | 0.00007913 | 499.9863 | | 1m | 0.0000019026 | 525600 | 0.00000132 | 499.9998 | |

#Annual harvest rate against instantaneous F, Fig 3.7   
Fi <- seq(0.001,2,0.001)   
H <- 1 - exp(-Fi)   
parset() # a wrapper for simplifying defining the par values   
plot(Fi,H,type="l",lwd=2,panel.first=grid(),   
 xlab="Instantaneous Fishing Mortality F",   
 ylab="Annual Proportion Mortality H")   
lines(c(0,1),c(0,1),lwd=2,lty=2,col=2)

|  |
| --- |
| 图 3.7: 瞬时捕捞死亡率（实线）与年度捕捞死亡率或年度收获率（虚线）之间的关系。在 F 值大于约 0.2 时，数值开始分离。 |

## 3.3 简单单位补充量渔获量

我们知道，一旦某一世代补充到群体中，其数量只会减少(假设没有迁入)。注意”补充量（recruitment）“的概念，它可以指后期仔鱼进入资源后，然而，它也可以指个体变得可用或易受渔业损害。这里我们关注的是仔鱼后加入资源后的情况。当我们在下面和第 [5](#sec-staticmodel) 章“静态模型”（*Static Models*）的章节中考虑选择性和可用性概念时，我们将处理补充量加入到渔业。有趣的是,尽管数字预计将减少通过时间,鉴于个体体长和重量上生长，我们还不知道如果世代的质量总是随时间减少或者增加然后减少最终所有的个体死亡。

一个世代的质量是否会在几年内增加，即使它的数量首次发现下降，因为对于捕获和努力之间的关系有一种简单但普遍的直觉。很明显，一种鱼类的渔获量总是会随着捕捞努力量的增加而增加，但这只是短期的想法。这个看似显而易见的概念在1930年之前就被驳斥了，并被 Russell (1942) 的一些出版著作很好地阐述了，他证明了一个渔业的最大持续渔获量不一定来自于最大努力量(或最大捕捞死亡)。这不仅是一个学术问题，而且直接关系到渔业的管理。一旦人们意识到过度捕捞甚至是鱼类资源崩溃的可能性，这一问题直到 20 世纪初才被正式承认 (Garstang, 1900)，那么如何最好地管理渔业就成为一个重要的问题。在管理渔业时，曾试图管理所作的全部努力量。支持这一观点的一个论点来自于单位补充量渔获量 (yield-per-recruit，YPR) 的概念，该概念以数量和质量的形式从数学上跟踪了在不同水平的强制捕捞死亡率下的世代的命运；捕捞死亡率被认为与所采取的努力量直接相关。“单位补充量”的概念意味着它遵循个体世代，并以最大化每个补充量的收益为目标。

我们将重复 Russell (1942) 的例子来说明YPR背后的思想。Russell 的例子只涉及捕捞死亡率，这与假设没有自然死亡率（忽略了 M）是一样的。对于这个简单的 YPR 我们也会忽略自然死亡率和应用一系列常数收获率每年计算年龄-数量（numbers-at-age）(相当于式 [公式 3.3](#eq-3_3) ),我们将使用一个年龄-体重乘以年龄-数量计算年龄-渔获量的重量,从中我们可以获得总的预期渔获量。

# Simple Yield-per-Recruit see Russell (1942)   
age <- 1:11; nage <- length(age); N0 <- 1000 # some definitions   
 # weight-at-age values   
WaA <- c(NA,0.082,0.175,0.283,0.4,0.523,0.7,0.85,0.925,0.99,1.0)   
 # now the harvest rates   
H <- c(0.01,0.06,0.11,0.16,0.21,0.26,0.31,0.36,0.55,0.8)   
nH <- length(H)   
NaA <- matrix(0,nrow=nage,ncol=nH,dimnames=list(age,H)) # storage   
CatchN <- NaA; CatchW <- NaA # define some storage matrices   
for (i in 1:nH) { # loop through the harvest rates   
 NaA[1,i] <- N0 # start each harvest rate with initial numbers   
 for (age in 2:nage) { # loop through over-simplified dynamics   
 NaA[age,i] <- NaA[(age-1),i] \* (1 - H[i])   
 CatchN[age,i] <- NaA[(age-1),i] - NaA[age,i]   
 }   
 CatchW[,i] <- CatchN[,i] \* WaA   
} # transpose the vector of total catches to   
totC <- t(colSums(CatchW,na.rm=TRUE)) # simplify later printing

不同的渔获率对种群年龄结构的影响在渔获的分年龄数量和分年龄重量上都很明显。随着渔获率的增加，高年龄组的捕获数量越来越少，分年龄捕获数也越来越少。对于较高的捕获率，种群变得依赖于最近的补充量，而不是在较老的年龄组中数量和生物量的积累。

kable(NaA,digits=c(0,0,0,0,0,0,0,0,1,1),row.names=TRUE)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.4: 应用简单年龄结构模型进行单位补充量产量分析得到的10个年渔获率情形下，种群的分年龄渔获数量。   |  | 0.01 | 0.06 | 0.11 | 0.16 | 0.21 | 0.26 | 0.31 | 0.36 | 0.55 | 0.8 | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 1 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000.0 | 1000.0 | | 2 | 990 | 940 | 890 | 840 | 790 | 740 | 690 | 640 | 450.0 | 200.0 | | 3 | 980 | 884 | 792 | 706 | 624 | 548 | 476 | 410 | 202.5 | 40.0 | | 4 | 970 | 831 | 705 | 593 | 493 | 405 | 329 | 262 | 91.1 | 8.0 | | 5 | 961 | 781 | 627 | 498 | 390 | 300 | 227 | 168 | 41.0 | 1.6 | | 6 | 951 | 734 | 558 | 418 | 308 | 222 | 156 | 107 | 18.5 | 0.3 | | 7 | 941 | 690 | 497 | 351 | 243 | 164 | 108 | 69 | 8.3 | 0.1 | | 8 | 932 | 648 | 442 | 295 | 192 | 122 | 74 | 44 | 3.7 | 0.0 | | 9 | 923 | 610 | 394 | 248 | 152 | 90 | 51 | 28 | 1.7 | 0.0 | | 10 | 914 | 573 | 350 | 208 | 120 | 67 | 35 | 18 | 0.8 | 0.0 | | 11 | 904 | 539 | 312 | 175 | 95 | 49 | 24 | 12 | 0.3 | 0.0 | |

kable(CatchW[2:11,],digits=c(2,2,2,2,2,2,2,2,2,2),row.names=TRUE)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.5: 应用简单年龄结构模型进行单位补充量产量分析得到的10个年渔获率情形下，种群的分年龄渔获重量。   |  | 0.01 | 0.06 | 0.11 | 0.16 | 0.21 | 0.26 | 0.31 | 0.36 | 0.55 | 0.8 | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 2 | 0.82 | 4.92 | 9.02 | 13.12 | 17.22 | 21.32 | 25.42 | 29.52 | 45.10 | 65.60 | | 3 | 1.73 | 9.87 | 17.13 | 23.52 | 29.03 | 33.67 | 37.43 | 40.32 | 43.31 | 28.00 | | 4 | 2.77 | 15.00 | 24.66 | 31.95 | 37.09 | 40.29 | 41.77 | 41.73 | 31.52 | 9.06 | | 5 | 3.88 | 19.93 | 31.02 | 37.93 | 41.42 | 42.14 | 40.74 | 37.75 | 20.05 | 2.56 | | 6 | 5.02 | 24.50 | 36.10 | 41.66 | 42.78 | 40.78 | 36.75 | 31.59 | 11.80 | 0.67 | | 7 | 6.66 | 30.82 | 43.00 | 46.84 | 45.23 | 40.39 | 33.94 | 27.06 | 7.10 | 0.18 | | 8 | 8.00 | 35.18 | 46.47 | 47.78 | 43.39 | 36.29 | 28.44 | 21.03 | 3.88 | 0.04 | | 9 | 8.62 | 35.99 | 45.01 | 43.67 | 37.30 | 29.22 | 21.35 | 14.65 | 1.90 | 0.01 | | 10 | 9.14 | 36.21 | 42.87 | 39.26 | 31.54 | 23.15 | 15.77 | 10.03 | 0.92 | 0.00 | | 11 | 9.14 | 34.38 | 38.54 | 33.31 | 25.17 | 17.30 | 10.99 | 6.49 | 0.42 | 0.00 | |

kable(totC,digits=c(1,1,1,1,1,1,1,1,1,1))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.6: 应用简单年龄结构模型进行单位补充量产量分析得到的10个年渔获率情形下种群的渔获量。   | 0.01 | 0.06 | 0.11 | 0.16 | 0.21 | 0.26 | 0.31 | 0.36 | 0.55 | 0.8 | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 55.8 | 246.8 | 333.8 | 359.1 | 350.2 | 324.5 | 292.6 | 260.2 | 166 | 106.1 | |

如果使用 **MQMF** 中的 plot1()函数绘制渔获总重量对捕获率的图，我们可以立即看到，在产量方面确实存在一个最佳渔获率，而不是最大利用率。如果让努力量增加，渔获量并不总是增加。如果捕获率很低，那么捕获的鱼类太少，不足以弥补显著的捕获量。然而，产量随着捕获率的增加而迅速增加，而当超过产生最大产量的捕获率时，产量下降得更慢。

#Use MQMF::plot1 for a quick plot of the total catches. Figure 3.8   
plot1(H,totC,xlab="Harvest Rate",ylab="Total Yield",lwd=2)

|  |
| --- |
| 图 3.8: 忽略自然死亡率、年龄选择性或任何其他影响的简化单位补充量-渔获量分析。该分析使用了 Russell（1942）的部分年龄-体重数据，尽管 Russell 仅考察了两种捕捞率，但本分析考察了更广泛的捕捞率。 |

为了快速说明不同捕获率的影响，这个例子在 1942 年获得了成功。对单位补充量渔获量的更全面的处理需要考虑动态中的自然死亡率和网具选择性带来的年龄死亡率差异。

### 3.3.1 单位补充量渔获量中的选择性

通常，在完全年龄结构的渔业模型中，人们会用一个选择性函数来描述每个年龄组(或体长组)对特定捕捞网具的不同脆弱性。随着鱼的年龄增加(或变大)，鱼会越来越多地补充到渔业中，直到它们完全易于受到捕捞的影响。通常，这种选择性曲线的参数是在将年龄结构模型与渔业数据拟合的过程中获得的。然而，在这里，我们将仅仅是给定了属性值(参见第 [5](#sec-staticmodel)章“静态模型”（*static Models*）中选择性的更多细节)。早期的 YPR 分析经常使用所谓的刀刃选择性，这意味着鱼类在特定年龄 100% 容易受到渔具的攻击。然而，在这里，我们将实现一个更现实、但更简单的脆弱性随年龄的选择性曲线。有许多不同的方程用来描述不同渔具的选择性特性，但拖网渔具使用的一个非常常见的方程是标准 logistic 或 s 形曲线，它也可以有许多公式化的表达式。一个常用来描述随年龄 或体长变化，选择性 logistic 形状 ，以及年龄或体长的成熟度定义为:

式中 和 为 logistic 参数， 为 0.5（50%）选择性处的年龄。四分位数之间的距离(字面上分位数的25%到75%；logistic曲线梯度的度量)定义为 （参见 **MQMF** 函数 mature() 用于该函数的执行）。一般来说，在年龄结构模型中，需要长度或年龄组成数据，以便直接估计网具选择性和渔业可用性。在计算单位补充量渔获量时，通常的理由是要包括一种形式的选择性，以确定开始应用捕捞死亡率的最佳年龄。在管理方面，这可以用来确定拖网或刺网的网眼尺寸。这就是为什么刀刃选择性经常被使用的原因，它可以识别年龄在什么年龄以下没有选择，在什么年龄以上有 100% 的选择。这在 **MQMF** 函数 logistic() 中实现（使用不同的公式），但在 mature() 中没有实现。刀刃选择性并不倾向于在完整的年龄结构种群评估模型中使用。

#Logistic S shaped cureve for maturity   
ages <- seq(0,50,1)   
sel1 <- mature(-3.650425,0.146017,sizeage=ages) #-3.65/0.146=25   
sel2 <- mature(-6,0.2,ages)   
sel3 <- mature(-6,0.24,ages)   
plot1(ages,sel1,xlab="Age Yrs",ylab="Selectivity",cex=0.75,lwd=2)   
lines(ages,sel2,col=2,lwd=2,lty=2)   
lines(ages,sel3,col=3,lwd=2,lty=3)   
abline(v=25,col="grey",lty=2)   
abline(h=c(0.25,0.5,0.75),col="grey",lty=2)   
legend("topleft",c("25\_15.04","30\_10.986","25\_9.155"),col=c(1,2,3),   
 lwd=3,cex=1.1,bty="n",lty=1:3)

|  |
| --- |
| 图 3.9: 使用 mature() 函数的逻辑斯蒂 S 形曲线示例。图例包含每条曲线的 L50 和 IQ，其参数在代码中定义。 |

### 3.3.2 Baranov 渔获量方程

[公式 3.4](#eq-3_4) 描述了结合捕捞和自然死亡后的残存率，但简单的总死亡率 意味着所有鱼的捕捞死亡率相同。如果我们要考虑选择性，那么我们需要考虑年龄的捕捞死亡率。

其中 为 年 龄鱼的残存率， 为自然死亡系数（假设随时间不变）， 为年龄 的选择性， 已知， 年完全选择的捕捞死亡系数。

一旦残存数量 已知，那么死亡的数量显然不同时期存在差异。从一个时间段到下一个时间段一个世代的死亡总数量可以表示为:

如果用残存等式代替 ，就可得到残存的补偿方程，其描述了时间段间总死亡数量：

这里 为年龄 死亡的数量。当然，不是所有死亡的鱼都作为渔获，一些是自然死亡的，但也出现捕捞一些自然死亡的个体的现象。将死亡率分为 （假设不同年龄的自然死亡系数不变）和 (捕捞努力量引起的完全选择死亡)，它可以随时间 而变化，可以计算渔获量和残存量。可以将 简化为 。有可能进行这种分开的假设是年龄结构和体长结构的种群评估模型的一个重要组成部分。使用瞬时率，由于捕捞死亡而非自然死亡的数量用分数 来描述，如果将其包含到 [公式 3.11](#eq-3_11) 中，我们发现我们已经推导出了所谓的 Baranov 渔获量方程 (QUINN 2003; Quinn 和 Deriso 1999) ，现在通常用于渔业建模，以估计渔获中的数量。它被用来追踪单个世代的命运，如果将所有年龄段相加，就能得出总的渔获量:

其中为年年龄的期望渔获，为年年龄的瞬时捕捞死亡系率（），为自然死亡率（假设年龄间不变），为年年龄的数量。该方程可用**MQMF**的函数bce()运行。

# Baranov catch equation   
age <- 0:12; nage <- length(age)   
sa <-mature(-4,2,age) #selectivity-at-age   
H <- 0.2; M <- 0.35   
FF <- -log(1 - H)#Fully selected instantaneous fishing mortality   
Ft <- sa \* FF # instantaneous Fishing mortality-at-age   
N0 <- 1000   
out <- cbind(bce(M,Ft,N0,age),"Select"=sa) # out becomes Table 3.7

kable(out, digits = 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.7: 将巴拉诺夫捕捞方程应用于年捕捞率为 0.2 且瞬时自然死亡率为 0.3 的种群。   |  | Nt | N-Dying | Catch | Select | | --- | --- | --- | --- | --- | | 0 | 1000.000 | NA | NA | 0.018 | | 1 | 686.191 | 291.645 | 22.164 | 0.119 | | 2 | 432.501 | 192.368 | 61.322 | 0.500 | | 3 | 250.395 | 116.618 | 65.488 | 0.881 | | 4 | 141.728 | 66.827 | 41.840 | 0.982 | | 5 | 79.943 | 37.766 | 24.018 | 0.998 | | 6 | 45.071 | 21.298 | 13.574 | 1.000 | | 7 | 25.409 | 12.007 | 7.655 | 1.000 | | 8 | 14.325 | 6.769 | 4.316 | 1.000 | | 9 | 8.075 | 3.816 | 2.433 | 1.000 | | 10 | 4.553 | 2.151 | 1.372 | 1.000 | | 11 | 2.567 | 1.213 | 0.773 | 1.000 | | 12 | 1.447 | 0.684 | 0.436 | 1.000 | |

### 3.3.3 生长和各年龄体重

为了获得渔获量的质量(即渔获量的重量)，需要将渔获量中的各年龄数量乘以各年龄重量。各年龄体重可以是特定年份特定渔业的平均体重观测值的向量 (Beverton 和 Holt 1993) ，或者，更常见的是，从 von Bertalanffy 生长曲线推导出的标准年龄体重方程，它从各年龄长度开始。

其中，为年龄的长度，为平均最大年龄长度，为系数，它决定了如何快速达到 ， 为长度为 0 时的假设年龄。用 **MQMF** 的函数 vB()运行。假设各年龄长度和各年龄体重之间存在幂次关系，因此增加了两个参数:

其中， 为恒定值， 为 ， 为指数，近似为 3.0（大小是二维的，而重量是三维的）。对数变换形式当然是线性的。在第 [5](#sec-staticmodel)章“静态模型”（*Static Models*）中，我们将看到如何估计这些模型的参数。

## 3.4 完整的单位补充量产量

把所有这些整合在一起，意味着我们可以产生一个更完整的单位补充量产量（YPR）分析。标准的 YPR 分析假设恒定的补充量，这意味着我们可以跟踪单个世代的命运，并仍然捕捉所需的细节。当然，补充量不是一个常数，所以这种方法被认为是基于长期或平衡条件。以这种方式忽略变化和随机性意味着，任何关于潜在生产力和由此产生的管理决策的结论都需要谨慎对待（事实上，非常谨慎）。但这些方法是在渔业分析都是确定性的，并假定处于平衡状态时发展起来的，不确定性的含义尚未被探索 (Beverton 和 Holt 1993) 。

当目标是最大化产量时，这样的分析是有意义的。但是，恒定补充量的假设和对不确定性的忽视意味着这些分析通常不是保守的。人们尝试改进 YPR 分析提出的建议，其中比较有用的是 的出现(发音为 F zero point one)。定义为产量曲线起点处产量增长率 1/10 的渔获率 (Hilborn 和 Walters 1992)。 的优势这是捕捞努力量的相对 大幅下降，只会导致产量的很小损失。这可以提高渔业的经济效益和可持续性。然而，本质上使用 仍然是一个经验规则，它在实践中比 （最高产量点）更可持续，通常比 (平衡时可产生 MSY 的捕捞死亡率)更好。

# A more complete YPR analysis   
age <- 0:20; nage <- length(age) #storage vectors and matrices   
laa <- vB(c(50.0,0.25,-1.5),age) # length-at-age   
WaA <- (0.015 \* laa ^ 3.0)/1000 # weight-at-age as kg   
H <- seq(0.01,0.65,0.05); nH <- length(H)   
FF <- round(-log(1 - H),5) # Fully selected fishing mortality   
N0 <- 1000   
M <- 0.1   
numt <- matrix(0,nrow=nage,ncol=nH,dimnames=list(age,FF))   
catchN <- matrix(0,nrow=nage,ncol=nH,dimnames=list(age,FF))   
as50 <- c(1,2,3)   
yield <- matrix(0,nrow=nH,ncol=length(as50),dimnames=list(H,as50))   
for (sel in 1:length(as50)) {   
 sa <- logist(as50[sel],1.0,age) # selectivity-at-age   
 for (harv in 1:nH) {   
 Ft <- sa \* FF[harv] # Fishing mortality-at-age   
 out <- bce(M,Ft,N0,age)   
 numt[,harv] <- out[,"Nt"]   
 catchN[,harv] <- out[,"Catch"]   
 yield[harv,sel] <- sum(out[,"Catch"] \* WaA,na.rm=TRUE)   
 } # end of harv loop   
} # end of sel loop

#A full YPR analysis Figure 3.10   
plot1(H,yield[,3],xlab="Harvest Rate",ylab="Yield",cex=0.75,lwd=2)   
lines(H,yield[,2],lwd=2,col=2,lty=2)   
lines(H,yield[,1],lwd=2,col=3,lty=3)   
legend("bottomright",legend=as50,col=c(3,2,1),lwd=3,bty="n",   
 cex=1.0,lty=c(3,2,1))

|  |
| --- |
| 图 3.10: 不同捕捞强度和不同初次开发年龄对总平衡产量的影响。图例标明了不同的初次开发年龄。 |

这些详细的分析考虑了选择性、年龄体重和自然死亡率，提供了一个渔场在不同条件下的潜在产量的更准确的表示，尽管仍然是确定性的。一旦考虑到不确定性和自然变化，即使只是近似地使用 ，那么YPR分析仍然可以为了解特定渔业的生产能力提供一些有用的见解。当然，如今onbe更有可能进行单位补充量的利润分析，但原则是不变的。

kable(yield, digits = 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 3.8: 不同捕捞率（行）和不同首次开发年龄（列）对总平衡产量的影响。   |  | 1 | 2 | 3 | | --- | --- | --- | --- | | 0.01 | 85.825 | 84.152 | 81.136 | | 0.06 | 347.125 | 352.465 | 350.567 | | 0.11 | 447.067 | 469.250 | 480.134 | | 0.16 | 475.149 | 514.595 | 540.249 | | 0.21 | 471.987 | 526.522 | 565.880 | | 0.26 | 455.843 | 522.962 | 574.288 | | 0.31 | 434.880 | 512.360 | 573.986 | | 0.36 | 412.718 | 498.720 | 569.223 | | 0.41 | 390.958 | 483.969 | 562.163 | | 0.46 | 370.252 | 469.031 | 553.935 | | 0.51 | 350.834 | 454.348 | 545.146 | | 0.56 | 332.715 | 440.100 | 536.113 | | 0.61 | 315.808 | 426.329 | 526.995 | |

## 3.5 结束语

我们使用了相对简单的种群模型来说明渔业和生态学中的许多观点。重要的是，重点是模拟模型，而不是将模型与数据拟合(见下一章“模型参数估计”)。模拟模型通常包括模型参数值的不确定性，但是，由于许多题目在这里只作了介绍性的处理，更彻底的处理本身需要一本书。然而，通过探索在考虑的各种模型上施加一系列参数值的含义，这些模型的属性和隐含的动力学可以被揭示出来。因此，模拟研究是对任何自然过程建模的重要工具。然而，同样重要的是，使用这些参数的合理或现实的组合进行模拟。理解所有模型的数学性质是很有用的，但是，自然地，最重要的兴趣是当模型具有现实值并可能在自然界中找到时。

在使用任何种群模型或其组成部分时获得的直觉对理解任何未来的工作都有价值，但是，当然，所说明的每个模型(特别是相对简单的模型)中的上下文和假设总是要记住的。令人惊讶的是，即使是简单的模型有时也能提供对各种自然过程的洞察。

在本章中，我们只考虑了简单的种群模型，但我们也可以很好地使用 R 来探索物种之间在竞争、捕食、寄生、共生等过程中的相互作用。所有的早期模型(Lotka，1925；Wolterra，1927；Gause，1934)考虑了没有空间结构的种群。使用 R 会相对直接地增加这种模型的复杂性，并包括空间细节，如 Huffaker(1958) 的实验探索。Huffaker 后来的一些实验性捕食者-补捕食者安排持续了 490 天。在这种情况下，模拟研究对于扩展探索的可能性非常有用。这样有趣的工作将需要成为另一本书的一部分，因为在渔业领域，我们还有许多其他领域需要探索和讨论。

使用 R 作为进行此类模拟的工具，使得分析比在电子表格上布置参数操作更抽象。然而，代码块和已开发函数的可重用性，以及更复杂模型和函数的增量开发的潜力所带来的优势，超过了编程环境的更抽象的本质。使用 R 作为一种编程语言来开发这些不同的分析非常适合渔业和生态建模工作。

# 4. 模型参数估算

## 4.1 引言

生态学和渔业学学建模的一个更重要的方面涉及到模型与数据的拟合。这种模型拟合需要:

* 从对自然感兴趣的过程中获得的数据（样本、观测），
* 明确地选择一个适合手头任务的模型结构（模型设计，然后选择——大的主题本身），
* 明确地选择概率密度函数来表示当比较时，模拟过程的预测将如何不同于自然观测的预期分布(选择残差结构) ，最后,
* 寻找模型参数，优化之间的预测模型和任何观测数据(模型拟合的准则)匹配。

在上面的最后一个需求中，将模型与数据相匹配时所涉及的许多技巧/诡计/魔术都集中在那个看起来无害的单词或概念上进行**优化**。这是一个几乎是恶作剧的想法，有时会导致一个人陷入麻烦，虽然它也是一个挑战，往往是有趣的。生成所谓*最佳拟合*模型的不同方法是本章的重点。它围绕着在描述模型的质量拟合可用数据时使用什么标准的想法，以及如何实现明确选择的标准。

我一直使用”显式”这个词，并且有很好的理由，但是需要一些解释。很多人都有过对数据进行线性回归拟合的经验，但是，根据我的经验，很少有人意识到，当他们拟合这样一个模型时，他们假设使用了加性正态随机残差(正态误差) ，并且他们正在最小化这些残差的平方和。根据上述四个要求，当将一个线性回归应用到一个数据集时，线性关系的假设回答了第二个要求，正态误差的使用(附加常数方差的假设)回答了第三个要求，平方和的最小化是满足第四个要求的选择。通常情况下，最好是明确地了解自己在做什么，而不是仅仅出于习惯或模仿他人。为了对这些模型拟合需求做出最合适的选择(即做出可以辩护的选择) ，分析师还需要了解被建模的自然过程。一个人可以假设和断言几乎任何事情，但只有这样的选择可以得到有效的辩护。作为一个更一般的陈述，如果一个人不能为一组选择辩护，那么他就不应该做出这些选择。

### 4.1.1 最优化

在 Microsoft Excel 中，当对数据进行模型拟合时，使用内置的Excel解算器找到最佳模型参数。这包括设置电子表格，使最佳拟合标准（平方和、最大似然等，见下文）由一个单元格的内容表示，而模型参数和使用的数据则包含在其他相互关联的单元格中。改变一个模型的参数会改变产生的预测值，这反过来又会改变最佳拟合的标准值。一个 “最佳”参数集可以通过寻找能优化观察值和预测值之间的匹配的参数来找到。这听起来很简单，但实际上是一门艺术，其中要做许多假设和决定。在 Excel 求解器中，人们确定了包含模型参数的单元格，然后求解器的内部代码将修改这些值，同时监测 “最佳拟合标准”单元格，直到找到一个最小（或最大）值（或遇到一个例外）。实际上，这样的电子表格设置构成了在Excel中使用求解器的语法。我们将在 R 中使用求解器或优化函数，它们也有一个必要的语法，但它并不比设置电子表格更复杂，只是更抽象而已。

当对单一数据集（如年龄-长度）用2至6个参数进行非动态过程建模时，模型拟合通常是相对简单的。然而，当处理一个种群的动态过程时，可能会变得更加复杂，涉及到补充、个体生长、自然死亡和多个捕鱼船队的捕捞死亡。可能有许多类型的数据，可能有多于几十个甚至几百个参数。在这种情况下，为了调整预测值与观测值的拟合质量，必须使用某种形式的自动优化或非线性求解器。

优化是一个非常大的研究课题，在CRAN任务视图中详细讨论了许多可用的选项。在 CRAN 任务视图：优化和数学编程中可以找到详细的讨论，网址是<https://cran.r-project.org/>。在本章的学习中，我们将主要使用内置的函数 nlm()(尝试 ?nlm)，但也有许多替代方法（包括 nlminb()、optim()等）。如果你要参与模型拟合，那么真的值得阅读 R-CRAN 上关于优化的任务视图，并且作为第一步，探索 nlm() 和 optim() 函数的帮助和例子。

有时有可能猜测出一组参数，产生看似合理的可视拟合，至少对简单的静态模型来说是如此。然而，虽然这种通过*目测拟合*（fitting-by-eye）可以为估计模型的参数提供可用的起点，但它并不构成将模型与数据拟合的可辩护的标准。这是因为我的 “目测拟合”（或称 “黑暗中的狂刺”）很可能与你的 “目测拟合”（或称 “有根据的猜测”）不同。与其使用这样的观点，不如使用一些更正式定义的模型与数据拟合质量的标准。

这里的重点是如何设置 R 代码，以便使用最小二乘法或最大似然法进行模型参数估计，特别是后者。我们以后对贝叶斯方法的考虑将主要集中在对不确定性的描述上。我们将通过重复的例子和相关的解释来说明模型的拟合过程。目的是阅读本节应该使读者能够建立自己的模型来解决参数值。我们将尝试以一种一般的方式来做这件事，它应该适合于许多问题的调整。

## 4.2 最佳拟合标准

确定什么是数据的最佳模型拟合通常有三种方法。

一般来讲，模型拟合包括观测变量 与为描述建模过程而提出候选模型所预测值 之间残差平方的最小化（ssq()）：

其中 是 个观测中的第 个观测值， 是给定观测值 的模型预测值（例如，如果 为鱼 的年龄-体长，则 为一些候选生长模型中得到的鱼 的年龄-体长预测值）。 中的 表示 的预测值。

或者，模型拟合可以涉及最小化负对数似然（在本书中为-veLL 或 negLL），这需要确定1）定义的模型结构，2）一组模型参数和 3）残差的期望概率分布的情况下，每个观测数据点的似然有多大。最小化负对数似然相当于最大化所有似然的乘积或所有数据点的对数似然总和。给定一个观测值 的集合，一个可以预测 的模型结构，以及一组模型参数 ，那么这些观测值的总似然定义为：

其中 为总似然，等于 或给定参数值 每个观测值 的概率密度（似然）积（在每种情况下，离期望值 越远，似然越低）。*-veLL* 是给定候选模型参数 时观测值 的总负对数似然，每个观测点 的 对数似然的负数和。 我们使用对数似然，因为大多数似然是非常小的数值，当与许多其他非常小的数字相乘时，会变得非常小，以至于有可能导致浮点溢出的计算机错误。对数变换将乘法变为加法，避免了这种风险( 转变为 ）。

第三种方法是使用贝叶斯方法，该方法使用先验概率，即在模型拟合中给予每个备选参数向量的初始相对权重。贝叶斯方法结合并更新任何关于最可能的模型参数的先验知识（先验概率），以及在不同的候选参数向量 下任何新数据的似然。就我们的目的而言，贝叶斯方法和最大似然法之间的两个关键区别是包含先验似然和重新缩放数值，以便使后验概率的总和达到1.0。重要的一点是，将给定一组参数的数据似然转换为给定数据参数的真实概率。给定数据 的特定参数集 的后验概率定义为：

其中 为参数集 的先验概率，通过给定参数 的数据 似然、 进行更新，除数 对结果重新缩放或归一化，因此在给定数据的情况下，所有参数向量 的后验概率之和为 1.0。最终 [公式 4.3](#eq-4_3) 是一个近似值，因为除数中的总和实际上应该是一个连续分布的积分，但在实践中，近似值就足够了，而且是处理复杂渔业模型参数时的唯一实际选择，其后验分布没有简单的分析解。

这里我们将主要关注负对数似然的最小化（相当于最大似然）。尽管其他方法也会得到一些关注。当我们探讨不确定性的特征时，贝叶斯方法将得到更多的关注。

微软的Excel在很多方面都是很好的软件，但是实现最大似然法，特别是贝叶斯法往往是缓慢和笨拙的，它们更适合于在R中实现。

识别平方残差之和、最大似然法和贝叶斯法并不是一个详尽的清单，它是在对数据进行模型拟合时可能使用的标准。例如，可以使用 “绝对残差之和”（sum of absolute residuals, SAR），它通过使用残差的绝对值而不是将其平方化来避免合并正负残差的问题 (Birkes 和 Dodge 1993)。尽管存在这种最佳模型拟合的替代标准，我们将只关注上述三种。其他更常用的替代方法包括所谓的稳健方法，这些方法致力于减少现有数据中的离群值，或极端的、被认为是不典型的值的影响。如前所述，优化是一个庞大而详细的研究领域，我向你推荐它的研究，并祝你好运。

## 4.3 R语言中的模型拟合

虽然覆盖参数空间的网格搜索可能是寻找最佳参数集的一种可能的方法，但随着参数数量增加到两个以上，它将变得越来越难操作，直到最后变得不可行。我们将不再考虑这种可能性。相反，为了便于寻找最佳参数集，我们需要一个用软件实现的非线性优化器。

R系统有一系列不同的优化函数，每个函数都使用不同的算法（请参见CRAN任务中关于优化的内容）。解算函数（nlm()）和其他函数一样，需要给出一个参数值的初始猜测，然后这些初始参数值由优化函数改变，在每次改变时，预测值会像ssq() 或negLL() 一样重新计算。优化函数，如nlm()，继续改变参数值（它们如何做到这一点是算法不同的地方），直到找到一个组合，根据所选择的任何标准被定义为 “最适合”（或无法找到进一步的改进）。渔业种群评估模型通常有许多参数，数量在10或100个左右（一些有更多的参数，需要更多的专业软件，例如 Daid A. Fournier, Hampton, 和 Sibert (1998); David A. Fournier 等 (2012); Kristensen 等 (2016) ）。在本书中，我们不会估计大量的参数，但无论数量多少，其原理都是相似的。

### 4.3.1 模型需求

讨论模型拟合的理论是有帮助的，但并没有阐明如何在 R 中实现在实践中拟合模型。优化软件用于改变参数向量内的值，但我们需要为其提供重复计算预测值的方法，该预测值将用于与观测值进行多次比较以找到最佳解决方案（如果成功的）。我们需要开发可以重复调用的代码块，这正是设计 R 语言函数的目的。为了在 R语言 中实现对现实世界问题的模型拟合，我们需要考虑四个形式要求：

* 来自所研究系统的观测（数据）。这可能是渔业中一个具有观测到的渔获量、cpue、渔获量的年龄和长度组成等，或者它可能是更简单的东西，例如鱼样本的观测到的长度和相关的年龄（但是如何将其放入 R ?),
* 第一个 R语言函数，表示系统的候选模型，当提供参数向量时，该函数用于计算预测值，以便与任何可用的观测值进行比较，
* 第二个 R语言 函数，计算选定的最佳拟合标准、最小化最小二乘或最小化负对数似然，以便将观测值与预测值进行比较。这需要能够返回单个值，反映输入参数和数据，然后可以通过最终所需的函数最小化，即
* 第三个R 语言函数（我们将倾向于使用 nlm()）来自动优化所选最佳拟合标准的值。

因此，需要输入数据和三个函数（ [图 4.1](#fig-4-1) ）但是，因为我们可以使用内置函数来进行优化，模型拟合通常需要编写最多两个函数，一个用于从使用的任何模型计算预测值另一个用于计算拟合标准（有时，在更简单的练习中，这两个可以组合成一个函数）。

我们在本书中假设读者至少熟悉模型拟合背后的概念，如拟合线性回归，因此我们将直接转向非线性模型拟合。这些相对简单的示例的主要目的是介绍 R 中可用求解器的使用和语法。

|  |
| --- |
| 图 4.1: 将模型拟合到数据时的输入、功能需求和输出。优化函数（此处为 nlm() ）最小化负对数似然（或平方和）并需要一个初始参数向量开始。此外，优化器需要一个函数（可能是 negLL() ）来计算它在搜索最小值时产生的每个参数向量的对应负对数似然。 要计算负对数似然需要一个函数（可能是 vB() ）来生成预测值，以便与输入的观测值进行比较。 |

### 4.3.2 年龄-体长示例

将模型拟合到数据仅仅意味着估计模型的参数，以便其预测与观测结果相匹配，以及根据选择的最佳拟合标准。作为在 R 中将模型拟合到数据的第一个说明，我们将使用一个简单的示例用著名的 von Bertalanffy 生长曲线 (von Bertalanffy, 1938) 拟合一组年龄-长度数据。这样的数据集包含在 R 包 **MQMF**（尝试 ?LatA）中。要使用您自己的数据，一种选择是生成一个逗号分隔的变量 (csv) 文件，其中包含最少的年龄和长度列，每个列都有一个列名（LatA 仅有年龄和长度列；请参阅其帮助页面）。可以使用 laa <- read.csv(file="filename.csv", header=TRUE) 将此 csv 文件读入 R中。

von Bertalanffy 长度-年龄生长曲线表示为：

其中 为年龄 的期望或预测长度， 为渐近平均最大长度， 为是决定达到最大值的增长率系数， 为物种长度0时的假设年龄（von Bertalanffy, 1938），一旦我们有了， ， 的估算值（或假设值），该非线性方程就提供了一种预测不同年龄的年龄-长度的方法。在拟合数据模型时，使用 [公式 4.4](#eq-4_4) 中的下面两个方程，其中 为观测值，等于预测值加上正态随机离差 ，其中的每个值可能是正的或负的（某一年龄的观测值可能大于或小于期望长度）。最下面的方程实际上是关于决定使用什么剩余误差结构。在本章中，我们将描述生态学和渔业中使用的一系列可供选择的误差结构。它们并非都是可加的，有些是用函数关系而不是常数定义的（如）。

关于 [公式 4.4](#eq-4_4) 是非线性的表述是明确的，因为早期估计 von Bertalanffy （ vB() ）生长曲线参数值的方法涉及各种旨在近似线性化曲线的变换（例如 Beverton和Holt, 1957）。在20世纪50年代末和60年代，拟合 von Bertalanffy 曲线不是一件小事。令人高兴的是，不再需要这样的转换，这样的曲线拟合也变得很简单。

### 4.3.3 其它的生长模型

个体生长的研究文献很多，描述生物体生长的模型多种多样（Schnute and Richards, 1990）。自Beverton和Holt (1957)引入以来，von Bertalanffy （vB() ）曲线一直是主要的渔业模型，已被渔业科学家广泛应用。然而，只是因为该模型非常普遍命令使用，对于所有物种来说，并不一定意味着该模型总是提供生长的最佳描述。模型选择是渔业建模中一个至关重要但经常被忽视的方面 (Burnham 和 Anderson 2004; Helidoniotis 和 Haddon 2013)。这里两种替代 vB()的两种模型可能是 Gompertz 生长曲线 （Gompertz, 1925）:

以及广义Michaelis-Menten 方程（Maynard Smith and Slatkin, 1973; Legendre and Legendre, 1998):

每个模型也有 a、b 和 c 三个参数，每个模型都能对生长过程的经验数据作出令人信服的描述。可以对某些参数进行生物学解释（如最大平均长度），但这些模型最终只提供了对生长过程的经验描述。如果把模型解释为反映了现实，就会导致完全不可信的预测，如不存在一米长的鱼（Knight，1968 ）。在文献中，参数可以有不同的符号（例如，Maynard Smith 和 Slatkin（1973）用 代替 表示 Michaelis-Menton），但基本结构形式是相同的。在对 **MQM**F 的年龄-长度数据集 中的鱼类进行 von Bertalanffy 生长曲线拟合后，我们可以利用不同模型来说明尝试这些替代模型的价值，并对应该使用哪种模型保持开放的心态。这个问题在我们讨论不确定性时会再次出现，因为我们可以从不同的模型中得到不同的结果。在对任何自然过程建模时，模型选择都是需要做出的重大决定之一。重要的是，通过这种方式尝试不同的模型，还能强化模型与数据拟合的过程。

## 4.4 残差平方和

数据拟合模型的经典方法称为 “最小残差平方和”（见 [公式 4.1](#eq-4_1) 和 [公式 4.7](#eq-4_7) ），或更常称为 “最小二乘法”。这种方法被认为是高斯提出的（Nievergelt, 2000, 引用了高斯 1823 年用拉丁文撰写的一本书的译文：*Theoria combinationis observationum erroribus minimis obnoxiae*）。无论如何，最小二乘法符合两个多世纪以来用于确定一组预测值与观测值最佳拟合的策略。这种策略就是确定一个所谓的目标函数（最佳拟合标准），根据函数结构，可以将其最小化或最大化。就残差平方和而言，我们需要从相关的观测值中减去每个预测值，将不同的结果平方（以避免出现负值），然后将所有值相加，并使用数学（解析解）或其他方法将该相加值最小化：

其中， 为 *n* 个观测值的残差平方和， 为第 个观测值，为第第 个观测值的期望值或预测值。**MQMF** 包中的ssq 函数仅仅是一个封装器，它调用了用于生成预测值的任何函数，然后计算并返回平方差和。根据不同问题的复杂程度和数据输入，我们通常需要创建新的函数作为封装。ssq() 很好地说明了这样一个事实，即在一个传递给函数的参数中，也可以传递其他函数（在本例中，在 ssq()内我们调用了传递给 funk 的函数，当然在使用ssq()时，我们输入的是与当前问题相关的实际函数，也许是vB，注意在用作函数参数时没有括号）。

### 4.4.1 最小二乘法的假设

最小二乘方法的一个主要假设是，残差项呈正态分布，所有观测变量值具有相等的方差；即在 中， 是常数。如果以任何方式对数据进行变换，则变换对残差的影响可能违反这一假设。相反，如果残差以系统的方式变化，则转换可以标准化残差方差。因此，如果数据是对数正态分布，那么对数变换将使数据标准化，然后可以有效地使用最小二乘。与往常一样，考虑或可视化数据和残差的形式(由拟合模型产生)是一种很好的做法。

### 4.4.2 数值求解

渔业科学中大多数有趣的问题都没有解析解（如线性回归），有必要使用数值方法通过定义的”最佳拟合”标准（如最小残差平方和(最小二乘)）来寻找最佳拟合模型。这显然会涉及到一点R编程，但是R的一个很大的优势是，一旦你开发了一套分析方法，就可以直接地将其应用到新的数据集。

在下面的示例中，我们用 **MQMF** 中的一些实用函数来辅助描述。我们需要 5 个函数拟合和比较上文定义的3种不同的生长模型，其中4个需要编写。前3个函数用于估计与观测数据进行比较的各年龄长度预测值。本例有3个候选模型函数分别对应3种不同的生长曲线：vB()、 Gz()和 mm()。第4个函数用作计算预测值及其相关观测值的平方和残差。这里将使用 **MQMF** 中的函数 ssq()（你应该检查并理解其代码）。该函数返回单一数值，该值将由最后一个函数 nlm()最小化，该函数需要自动最小化求解。R 函数 nlm()使用用户定义的广义函数，用 表示 (尝试 args(nlm)，或formals(nlm)以查看完整的参数列表），用于计算最小值（在本例中为 ssq()），而ssq()又使用预测生长曲线中各年龄长度的函数（例如vB()）。如果我们使用不同的生长曲线函数（例如Gz()），只需将nlm()调用代码中指向vB() 的地方改为 Gz() ，并修改参数值以适应 Gz()函数，以便代码产生可用的结果。从根本上说，nlm()通过改变输入参数(称作 *p* )来最小化ssq()，无论选择哪个参数，都会改变生长函数vB()、Gz()或mm()的结果，。

nlm()只是 **R** 中用于非线性优化的函数之一，候选函数包括 optim()和 nlminb()（请阅读 nlm中的文档，CRAN 上关于优化的任务视图列出了旨在解决优化问题的包）。

library(MQMF)  
library(ggplot2)  
 #setup optimization using growth and ssq   
data(LatA) # try ?LatA assumes library(MQMF) already run   
 #convert equations 4.4 to 4.6 into vectorized R functions   
 #These will over-write the same functions in the MQMF package   
  
vB <- function(p, ages) return(p[1]\*(1-exp(-p[2]\*(ages-p[3]))))   
Gz <- function(p, ages) return(p[1]\*exp(-p[2]\*exp(p[3]\*ages)))   
mm <- function(p, ages) return((p[1]\*ages)/(p[2] + ages^p[3]))   
  
 #specific function to calc ssq. The ssq within MQMF is more   
ssq <- function(p,funk,agedata,observed) { #general and is   
 predval <- funk(p,agedata) #not limited to p and agedata   
 return(sum((observed - predval)^2,na.rm=TRUE))   
} #end of ssq   
 # guess starting values for Linf, K, and t0, names not needed   
pars <- c("Linf"=27.0,"K"=0.15,"t0"=-2.0) #ssq should=1478.449   
ssq(p=pars, funk=vB, agedata=LatA$age, observed=LatA$length)

[1] 1478.449

ssq()函数取代了全局环境中的 MQMF::ssq()函数，但也返回一个数值，例如上面的 1478.449，它是 nlm()函数的第一个输入，也是需要最小化的值。

### 4.4.3 将函数作为参数传递给其它函数

在上例中，我们定义了一些用于模型拟合数据所需的函数，确定了要比较的生长模型，也定义了计算平方和的函数。刚刚做的一个非常重要的方面是，为了计算平方和，我们将 vB()函数作为参数传递给 ssq()函数。这意味着我们传递了一个函数，它有参数，作为另一个函数的参数之一。你可以在这里看到潜在的混乱，所以有必要集中精力，保持清晰。目前，我们定义 ssq()的方式似乎并没有那么引人注目，因为我们已经在对ssq()的调用中显式地定义了两个函数的参数。但是 R 有一些妙招，我们可以用它来泛化包含其他函数作为参数的函数，主要的一个使用了神奇的省略号…，对于任何R函数，除非实参在其定义中设置了默认值，否则每个实参都必须给定一个值。在上面的ssq()函数中，我们包含了仅由 ssq() 使用的参数（ *funk* 和 *observed* ），以及仅由函数 funk使用的参数（ *p* 和 *agedata*）。这在本例子中很有效，因为我们故意将生长函数定义为具有相同的输入参数，但如果我们想使用的 funk 有不同的输入，可能是因为我们拟合的是选择性曲线而不是生长曲线，该怎么办？显然，我们需要编写一个不同的 ssq() 函数。为了允许在更多情况下重用更通用的函数，R 的作者（R Core Team, 2019）包含了这个概念 …，它将匹配其他方式不匹配的任何参数，因此可用于输入funk 函数的参数。因此，我们可以这样重新定义 ssq():

# Illustrates use of names within function arguments   
vB <- function(p,ages) return(p[1]\*(1-exp(-p[2] \*(ages-p[3]))))   
ssq <- function(funk,observed,...) { # only define ssq arguments   
 predval <- funk(...) # funks arguments are implicit   
 return(sum((observed - predval)^2,na.rm=TRUE))   
} # end of ssq   
pars <- c("Linf"=27.0,"K"=0.15,"t0"=-2.0) # ssq should = 1478.449   
ssq(p=pars, funk=vB, ages=LatA$age, observed=LatA$length) #if no

[1] 1478.449

ssq(vB,LatA$length,pars,LatA$age) # name order is now vital!

[1] 1478.449

这意味着 ssq() 函数现在更加通用，可以与任何输入函数一起使用，这些输入函数可用于生成一组预测值，以便与一组观测值进行比较。**MQMF** 中的 ssq() 函数就是这样实现的；查阅帮助 ?ssq。一般的想法是，您必须定义主函数中使用的所有参数，但任何仅在被调用函数（此处称为 *funk* ）中使用的参数都可以在 …中传递。最好是显式地命名参数，这样它们的顺序就无关紧要了，并且您需要非常小心地键入，如果您拼错了通过…传递的参数的名称，并不一定会出错！例如，使用大写的 *LatA$Age* 而不是 *LatA$age* 不会出错，但会导致结果为 0 而不是 1478.440。这是因为 *LatA$Age = NULL*，即使输入不正确也是有效的。很明显，… 可能非常有用，但如果你像我一样输入错误，其本身也是有风险的。

# Illustrate a problem with calling a function in a function   
 # LatA$age is typed as LatA$Age but no error, and result = 0   
ssq(funk=vB, observed=LatA$length, p=pars, ages=LatA$Age) # !!!

[1] 0

如果你匆忙行事，没有给参数命名，那么如果你弄错了参数顺序，也会失败。例如，如果你要输入的是ssq(LatA$length, vB, pars, LatA$age)而不是ssq(vB, LatA$length, pars, LatA$age)，就会出现错误：*Error in funk(…): could not find function ” funk “*。为了以防万一，你可以自己试试。对你的代码进行试验几乎没什么坏处，不会弄坏你的电脑，而你可能会学到一些东西。

### 4.4.4 模型拟合

如果我们绘制 *LatA* 数据集图（ [图 4.2](#fig-4-2) ），可以看到一些典型的年龄-长度数据。图中有 358 个点（试试 dim(LatA)），许多点彼此重叠，但是当我们使用 plot()函数中 rgb()选项来改变图中颜色的透明度时，较高年龄组鱼类的颜色相对稀疏性就会显现出来。另外，我们可以使用 jitter()为每个绘制点的位置添加噪声，以查看数据点的相对密度。每当你处理经验数据时，总是值得你花时间至少将其绘制成画并探索其属性。

#plot the LatA data set Figure 4.2   
parset() # parset and getmax are two MQMF functions   
ymax <- getmax(LatA$length) # simplifies use of base graphics. For   
 # full colour, with the rgb as set-up below, there must be >= 5 obs   
plot(LatA$age,LatA$length,type="p",pch=16,cex=1.2,xlab="Age Years",   
 ylab="Length cm",col=rgb(1,0,0,1/5),ylim=c(0,ymax),yaxs="i",   
 xlim=c(0,44),panel.first=grid())   
  
# ggplot(data = LatA, aes(x = age, y = length)) +  
# geom\_point(size = 3, color = "red", alpha = 0.2) +  
# labs(x = "Age Years",   
# y = "Length cm") +  
# theme\_bw()

|  |
| --- |
| 图 4.2: 基于澳大利亚东部样本模拟的 358 条红鱼（*Centroberyx affinis*）的雌性年龄长度数据。全强度颜色表示 >= 5 个点。 |

与其继续推测参数值并手动修改它们，我们可以使用 nlm() 或 optim()或 nlminb()，它们的语法不同)来拟合所选的 *LatA* 数据的生长模型或曲线。这不仅说明了nlm()的语法，还说明了另外两个 **MQMF** 实用R函数 magnitude()和 outfit()的使用(请查阅?nlm、?magnitude和 ?outfit)。还可以查看每个函数中的代码（在控制台中输入每个函数的名称，不带参数或括号）。从现在起，我将减少提示你查看所使用函数细节的次数，但是如果看到一个新的函数，希望现查看其帮助、语法，尤其是代码是有意义的，就像查看每个所用变量的内容一样。

3 个生长模型中的每一个都需要估计 3 个参数，我们需要对每个参数进行初始猜测，以启动 nlm()求解。我们所做的就是为 nlm()函数的每个形参/实参提供值。此外，我们还使用了 2 个额外的参数，*typsize* 和 *iterlim*。*typesize* 在 nlm()帮助中定义为”每个参数最小值的估计”。加入这 2 个参数一般有助于稳定搜索算法，因为它可以确保对每个参数值迭代变化尺度大致相同。一种非常常见的替代方法（我们将在更复杂的模型中使用）是在输入 nlm()时对每个参数进行对数变换，然后在调用的函数中对它们进行反变换，以计算预测值。但是，这只能在保证参数始终为正值情况下才可行。例如，对于 von Bertalanffy 曲线， 参数通常是负值，因此应使用 magnitude()而不是对数变换方法。默认的 iterlim=100意味着最多迭代 100 次，这有时是不够的，所以如果迭代达到 100 次，您应该将该数值增加到 1000 。你很快就会发现，在每次模型拟合过程中，唯一改变的是ssq()中的 *funk* 所指向的函数和初始参数值。这可以通过有意识地构建生长函数，使其使用完全相同的参数（通过…传递）。您也可以尝试在不设置 *typsize* 和 *interlim* 选项的情况下运行其中的一个或两个函数。还请注意，我们运行 Michaelis-Menton 曲线时使用了两个略有不同的初始点。

# use nlm to fit 3 growth curves to LatA, only p and funk change   
ages <- 1:max(LatA$age) # used in comparisons   
pars <- c(27.0,0.15,-2.0) # von Bertalanffy   
bestvB <- nlm(f=ssq,funk=vB,observed=LatA$length,p=pars,   
 ages=LatA$age,typsize=magnitude(pars))   
outfit(bestvB,backtran=FALSE,title="vB"); cat("\n")

nlm solution: vB   
minimum : 1361.421   
iterations : 24   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient  
1 26.8353971 -1.134306e-04  
2 0.1301587 -6.198614e-03  
3 -3.5866989 8.330772e-05

pars <- c(26.0,0.7,-0.5) # Gompertz   
bestGz <- nlm(f=ssq,funk=Gz,observed=LatA$length,p=pars,   
 ages=LatA$age,typsize=magnitude(pars))   
outfit(bestGz,backtran=FALSE,title="Gz"); cat("\n")

nlm solution: Gz   
minimum : 1374.36   
iterations : 28   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 26.4444554 2.725617e-05  
2 0.8682518 -6.452607e-04  
3 -0.1635476 -2.042292e-03

pars <- c(26.2,1.0,1.0) # Michaelis-Menton - first start point   
bestMM1 <- nlm(f=ssq,funk=mm,observed=LatA$length,p=pars,   
 ages=LatA$age,typsize=magnitude(pars))   
outfit(bestMM1,backtran=FALSE,title="MM"); cat("\n")

nlm solution: MM   
minimum : 1335.961   
iterations : 12   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient  
1 20.6633224 -0.02622725  
2 1.4035207 -0.37744267  
3 0.9018319 -0.05039237

pars <- c(23.0,1.0,1.0) # Michaelis-Menton - second start point   
bestMM2 <- nlm(f=ssq,funk=mm,observed=LatA$length,p=pars,   
 ages=LatA$age,typsize=magnitude(pars))   
outfit(bestMM2,backtran=FALSE,title="MM2"); cat("\n")

nlm solution: MM2   
minimum : 1335.957   
iterations : 25   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 20.7464274 8.464689e-06  
2 1.4183164 -3.856394e-05  
3 0.9029899 -1.297065e-04

这些都是数值解，它们不能保证是正确的解。请注意，第一个 Michaelis-Menton 解（从 26.2,1,1 开始）的梯度相对较大，但它的 SSQ(1335.96) 非常接近第二个 Michaelis-Menton 模型拟合，而且比 vB 或 Gz 曲线更小（更好）。然而，梯度值表明该模型的拟合可以而且应该得到改进。如果您将参数 *a*（首个 MM 参数）的初始参数估计值从上次模型拟合中的 26.2 降到了 23，我们就会得到稍有不同的参数值、稍小的 SSQ 以及更小的梯度，从而更有把握地认为结果是个真正的最小值。实际上，如果要运行cbind(mm(bestMM1$estimate,ages)， mm(bestMM2$estimate,ages))，可以计算出预测值的差异从 -0.018 到 0.21%，而如果将 vB 预测值包括在内，MM2 与 vB 预测的差异从 -6.15 到 9.88% (忽略 40.6% 的最大偏差)。你也可以尝试从 vB 模型的估计中省略*typsize* 参数，这样仍然会得到最佳结果，但可查看梯度以了解为什么使用 *typsize* 有助于优化。在设置这些示例时，偶尔运行 Gz()模型会出现 *steptol* 可能太小的注释，将其从默认的 1e-06 改为 1e-05 可以很快解决这个问题。如果您遇到了这种情况，请在nlm()命令中添加一条语句 steptol=1e-05，看看诊断是否有所改善。

显而易见的结论是，我们应该经常查阅 nlm()的诊断注释，考虑得到的解的方案的梯度，并且使用多组初始参数猜测，以确保得到稳定的解。数值求解以软件实现为基础，用于决定何时停止迭代的规则有时会被次优解所欺骗。我们的目标是找到全局最小值，而不是局部最小值。任何非线性模型都可能产生此类次优解，因此自动拟合此类模型并非易事。在这种情况下，永远不要假设在这种情况下你得到的第一个答案一定是你正在寻找的最佳答案，即使描绘的模型拟合看起来是可以接受的。

在函数调用中，如果你为每个参数命名，那么严格来讲，顺序并不重要，但我发现一致的用法可以简化代码的阅读，因此即使使用显式名称，也建议使用标准顺序。如果我们不使用显式名称，则 nlm()的语法要求首先定义函数最小化（*f*）。此外，它还要求 *f* 函数，无论它是什么，在 *p* 参数中使用初始参数猜测，如果未命名，则必须排在第二位。如果你在控制台中输入 formals(nlm)或 args(nlm)，就可得到可以输入到函数的可能参数以及它们的默认值（如果存在）:

#The use of args() and formals()   
args(nlm) # formals(nlm) uses more screen space. Try yourself.

function (f, p, ..., hessian = FALSE, typsize = rep(1, length(p)),   
 fscale = 1, print.level = 0, ndigit = 12, gradtol = 1e-06,   
 stepmax = max(1000 \* sqrt(sum((p/typsize)^2)), 1000), steptol = 1e-06,   
 iterlim = 100, check.analyticals = TRUE)   
NULL

正如所见，首先要最小化函数 *f* （在本例中是 ssq() ），然后是初始参数 *p*，它必须是*f*所指向的任何函数所需的第一个参数。接着是省略号(三个点)，它概括了任何函数*f*的 nlm()代码，其后是可能的参数集，所有这些参数都具有默认值。我们修改了 *typsize* 和 *iterm* (有时也修改了Gz() 中的 *steptol*）；请参阅 nlm()帮助以获得对每种方法的解释。

在R中，”…“ 实际上指的是所需的任何其他输入，例如函数*f*指向的任何参数（本例中为ssq）。如果 Et ?ssq查看参数或代码或帮助，将会看到所需函数 *funk*，该函数将用于计算相对于ssq()的下一个所需输入的期望值，为观测值的向量（如）。请注意，这里没有明确提到 *funk* 使用的参数，这些参数假定是通过 …. 传递的。在对 ssq()的每次调用中，我们都显式地填充了这些参数，例如nlm(f=ssq,funk=Gz, observed=LatA$length, p=pars, ages=LatA$age，)。这样，所有要求都已满足，ssq()就可以开始工作。如果不小心忽略了ages=LatA$age参数，那么在这种情况下，R 会出现如下揭示：*Error in funk(par, independent) : argument “ages” is missing, with no default*（我相信你会相信我的，但你自己试试也无妨!)。

就生长曲线模型拟合而言，绘制结果提供了一个直观的比较，说明了三条生长曲线之间的差异（Murrell, 2011）。

#Female length-at-age + 3 growth fitted curves Figure 4.3   
predvB <- vB(bestvB$estimate,ages) #get optimumpredicted lengths   
predGz <- Gz(bestGz$estimate,ages) # using the outputs   
predmm <- mm(bestMM2$estimate,ages) #from the nlm analysis above   
ymax <- getmax(LatA$length) #try ?getmax or getmax [no brackets]   
xmax <- getmax(LatA$age) #there is also a getmin, not used here   
parset(font=7) # or use parsyn() to prompt for the par syntax   
plot(LatA$age,LatA$length,type="p",pch=16, col=rgb(1,0,0,1/5),   
 cex=1.2,xlim=c(0,xmax),ylim=c(0,ymax),yaxs="i",xlab="Age",   
 ylab="Length (cm)",panel.first=grid())   
lines(ages,predvB,lwd=2,col=4) # vB col=4=blue   
lines(ages,predGz,lwd=2,col=1,lty=2) # Gompertz 1=black   
lines(ages,predmm,lwd=2,col=3,lty=3) # MM 3=green   
 #notice the legend function and its syntax.   
legend("bottomright",cex=1.2,c("von Bertalanffy","Gompertz",   
 "Michaelis-Menton"),col=c(4,1,3),lty=c(1,2,3),lwd=3,bty="n")   
  
# ggplot() +  
# geom\_point(data = LatA, aes(x = age, y = length),   
# color = "red", alpha = 0.2) +  
# geom\_line(aes(x = ages, y = predvB), color = "blue") +  
# geom\_line(aes(x = ages, y = predGz), color = "black") +  
# geom\_line(aes(x = ages, y = predmm), color = "green") +  
# theme\_bw()

|  |
| --- |
| 图 4.3: 来自 358 个模拟的雌性红鱼年龄-长度数据，顶部绘制了三条最佳拟合生长曲线。Female Length-at-Age data from 358 simulated female redfish with three optimally fitted growth curves drawn on top. |

在图中使用的 rgb() 函数意味着颜色强度代表观测数量，最强烈的颜色至少表示有五个观测值。很明显，在这份数据中， vB() 和 Gz() 曲线在大部分观测范围内几乎重合，而 mm() 曲线与其他两条曲线偏离，但主要是在极端值处。Michaelis-Menton 曲线被强制通过原点，而另外两条曲线则不受这种限制（尽管这个想法可能更接近现实）。可以包括幼鱼长度来将高姆珀茨曲线和 von Bertalanffy 曲线向下拉。但生物生长过程很复杂。许多鲨鱼和鳐鱼是胎生，确实在发育后期开始生活时体型显著大于零。始终要记住，这些曲线只是数据的经验性描述，反映现实的能力有限。

大部分可用数据集中在 3 至 12 龄之间（尝试 table(LatA$age) ），然后只有 24 龄以上单次出现。在 3 至 24 龄之间，Gompertz 曲线和 von Bertalanffy 曲线基本上遵循相同轨迹，而 Michaelis-Menton 曲线差异很小（你可以尝试以下代码查看实际差异 cbind(ages, predvB, predGz, predmm) ）。超出这个年龄范围，差异更大，尽管缺乏年轻动物表明捕捞样本的渔具选择性可能无法充分代表 3 龄以下的鱼类。在拟合相对质量（残差平方和）方面，最终的 Michaelis-Menton 曲线最小 ssq() ，其次是 von Bertalanffy 曲线，然后是 Gompertz 曲线。但每种模型都为 3 至 24 龄（数据最密集范围）之间的生长提供了合理的平均描述。当老年年龄组的数据如此稀疏时，也存在该样本是否代表这些年龄组种群的问题。质疑数据、所用模型及由此产生的解释，是构建自然过程有用模型的重要方面。

### 4.4.5 目标模型选择

在上述三种生长模型中，最优模型拟合定义为使预测值与实测值之间的平方和残差最小。根据这一标准，第二个生长模型 Michaelis-Menton 曲线比 von Bertalanffy 曲线和 Gompertz 曲线更适合。但我们真的能说第二条 Michaelis-Menton 曲线比第一条曲线”更好”拟合吗?就最终溶液的梯度而言，第二条曲线显然更好，但严格的拟合标准只是最小 SSQ，差异小于 0.01 个单位。模型选择通常是使用参数的数量和根据所选标准的拟合质量之间的权衡。如果我们设计一个具有更多参数的模型，这通常会导致更大的灵活性和更接近观测数据的改进能力。在极端情况下，如果我们有和观测值一样多的参数，我们可以有一个完美的模型拟合，但是，当然，我们对我们正在建模的系统一无所知。LatA 数据集有 358 个参数，这显然是过度参数化的情况，但如果我们只将参数数量增加到10个呢？毫无疑问，曲线的形状会很奇怪，但 SSQ 可能会更低。Burnham 和 Anderson(2002) 详细讨论了参数数量和模型与数据的拟合质量之间的权衡关系。在 20 世纪 70 年代，人们开始使用信息论来开发一种量化模型参数和模型拟合质量之间权衡的方法。Akaike(1974) 描述了他的Akaike信息标准（AIC），该标准基于最大似然和信息理论原理（稍后会详细介绍），但幸运的是，Burnham 和 Anderson(2002) 在使用最小平方和残差时提供了另一种选择，这是Atkinson（1980）中包含的一种变体:

其中 为观测数， 为”模型内独立调整参数数量”（Akaike，1974，p716）， 为方差的最大似然估计，仅表示残差平方和除以 而非 ：

即使有了*AIC*，也很难确定，当使用最小二乘时，差异是否可以被认为是统计上显著的差异。有与方差分析相关的方法，但当使用最大似然时，这些问题能够得到更可靠的回答，所以我们将在后面的部分中解决这个问题。

如果想在拟合模型时获得生物学上合理的或可站得住脚的解释，那么模型选择不能仅仅依赖于统计拟合的质量。相反，它应该反映理论预期（例如，种群中的平均生长是否包括随着时间的推移个体大小的平稳增长，等等）。除了数据的统计拟合之外，这些考虑因素似乎没有得到足够的重视，但只有在出现生物学上不可信的模型结果或提出不可信的模型结构时才变得重要。它显然有助于理解正在建模的过程的生物学期望。

### 4.4.6 残差选择对模型拟合的影响

在生长模型例子中，我们使用了正态随机偏差，但我们可以问，如果我们使用，例如对数正态偏差，我们是否会得到相同的答案？在这种情况下，我们所需要做的就是在计算残差平方和之前对观测值和预测值进行对数变换（参见下面关于对数正态残差的内容）。

这里我们继续在 outfit()中使用 *backtran=FALSE* 选项，因为我们是对数据进行对数转换，而不是对参数进行对数转换，因此不需要进行反向转换。

# von Bertalanffy   
pars <- c(27.25,0.15,-3.0)   
bestvBN <- nlm(f=ssq,funk=vB,observed=LatA$length,p=pars,   
 ages=LatA$age,typsize=magnitude(pars),iterlim=1000)   
outfit(bestvBN,backtran=FALSE,title="Normal errors"); cat("\n")

nlm solution: Normal errors   
minimum : 1361.421   
iterations : 22   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 26.8353990 -3.645466e-07  
2 0.1301587 -1.578320e-05  
3 -3.5867005 3.198205e-07

# modify ssq to account for log-normal errors in ssqL   
ssqL <- function(funk,observed,...) {   
 predval <- funk(...)   
 return(sum((log(observed) - log(predval))^2,na.rm=TRUE))   
} # end of ssqL   
bestvBLN <- nlm(f=ssqL,funk=vB,observed=LatA$length,p=pars,   
 ages=LatA$age,typsize=magnitude(pars),iterlim=1000)   
outfit(bestvBLN,backtran=FALSE,title="Log-Normal errors")

nlm solution: Log-Normal errors   
minimum : 3.153052   
iterations : 25   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 26.4409587 8.898258e-08  
2 0.1375784 7.546830e-06  
3 -3.2946086 -1.122823e-07

在这种情况下，使用正态和对数正态残差产生的曲线几乎没有区别（ [图 4.4](#fig-4-4) ）。尽管它们的参数不同（使用 ylim=c(10,ymax) 使差异更明显）。除了视觉上的不同，不同的模型甚至没有可比性。如果我们比较它们各自的平方和残差，一个是 1361.0，另一个只有 3.153。当我们考虑到平方和计算中对数变换的影响时，这并不奇怪。但这意味着我们不能只看表格输出，然后决定哪个版本比另一个更适合数据。它们是严格不相称的，尽管它们使用的是完全相同的模型。不同残差结构的使用需要在考虑相对模型拟合以外的方式进行辩护。这个例子强调，虽然模型的选择显然很重要，但残差结构的选择也是模型结构的一部分，同样重要。

# Now plot the resultibng two curves and the data Fig 4.4   
predvBN <- vB(bestvBN$estimate,ages)   
predvBLN <- vB(bestvBLN$estimate,ages)   
ymax <- getmax(LatA$length)   
xmax <- getmax(LatA$age)   
parset()   
plot(LatA$age,LatA$length,type="p",pch=16, col=rgb(1,0,0,1/5),   
 cex=1.2,xlim=c(0,xmax),ylim=c(0,ymax),yaxs="i",xlab="Age",   
 ylab="Length (cm)",panel.first=grid())   
lines(ages,predvBN,lwd=2,col=4,lty=2) # add Normal dashed   
lines(ages,predvBLN,lwd=2,col=1) # add Log-Normal solid   
legend("bottomright",c("Normal Errors","Log-Normal Errors"),   
 col=c(4,1),lty=c(2,1),lwd=3,bty="n",cex=1.2)

|  |
| --- |
| 图 4.4: Female Length-at-Age data from 358 female redfish, Centroberyx affinis, with two von Bertalanffy growth curves fitted using Normal and Log-Normal residuals. |

### 4.4.7 关于初始模型拟合的说明

上面例子中的曲线比较本身就很有趣，不过，我们还说明了 nlm() 的语法以及如何将模型拟合到数据中。将函数作为参数传递给另一个函数的能力（就像这里我们将 ssq 作为 *f* 传递给 nlm()，将 vB、Gz 和 mm 作为 *funk* 传递给 ssq()）是 R 的优势之一，但也是其复杂性所在。它简化了ssq()等函数的重用，在这些函数中，我们只需改变输入函数，就能从本质上相同的代码中得到完全不同的答案。熟悉这些方法的最佳途径是使用自己的数据集。绘制您的数据和任何模型拟合图，因为如果模型拟合图看起来不寻常，那么很可能就是不寻常的，需要再三审视。

使用平方和法可以取得很好的效果，但在处理现实世界的多样性时，要求在期望值附近的残差呈正态分布以及方差恒定的假设是有限制的。为了使用正态分布以外的概率密度分布和非恒定方差，我们应该转而使用最大似然法。

## 4.5 最大似然

在 R 中使用似然比较简单，因为有许多概率密度函数（PDF）的内置函数以及一系列定义其他 PDF 的软件包。再重复一遍，最大似然法的目的是使用软件搜索模型参数集，使观测数据的总似然最大化。要使用这一最优模型拟合标准，需要对模型进行定义，以便将每个观测值（可用数据）的概率或似然值指定为模型中参数值和其他变量的函数（ [公式 4.2](#eq-4_2) 和 [公式 4.11](#eq-4_11) ）。重要的是，这种规范包括对所选 PDF 的变异性或扩散性的估计（正态分布中的 只是最小二乘法的副产品）。使用最大似然法的一个主要优点是，残差结构或关于数据预期中心倾向的观测值的预期分布不一定是正态分布。如果可以定义概率密度函数 (PDF)，则可以在最大似然法框架中使用；有关许多有用概率密度函数的定义，请参见 Forbes et al, (2011)。

### 4.5.1 简要示例

我们将使用众所周知的正态分布来说明这些方法，然后扩展该方法以包含一系列可选的 PDF。对于数据模型拟合而言，每个PDF的主要意义在于定义单个观测值的概率密度或似然值。对于平均期望值为 或 的正态分布，给定单个值 的概率密度或似然定义为:

其中 为与 相关的标准差。这确定了最小二乘法和最大似然方法之间的直接区别，在后者中，需要一个 PDF 的完整定义，在正态分布的情况下，它包括对均值估计 周围残差的标准偏差的显式估计。这样的估计不需要最小二乘，虽然很容易从 SSQ 值中得到。

例如，我们可以从正态分布中生成一个观测样本(参见?rnorm)，然后计算该样本的均值和标准差，并比较给定的样本值与 *rnorm* 函数中使用的原始均值和标准差的参数估计值的可能性有多大（ [表 4.1](#tbl-4-1) ）：

# Illustrate Normal random likelihoods. see Table 4.1   
set.seed(12345) # make the use of random numbers repeatable   
x <- rnorm(10,mean=5.0,sd=1.0) # pseudo-randomly generate 10   
avx <- mean(x) # normally distributed values   
sdx <- sd(x) # estimate the mean and stdev of the sample   
L1 <- dnorm(x,mean=5.0,sd=1.0) # obtain likelihoods, L1, L2 for   
L2 <- dnorm(x,mean=avx,sd=sdx) # each data point for both sets   
result <- cbind(x,L1,L2,"L2gtL1"=(L2>L1)) # which is larger?   
result <- rbind(result,c(NA,prod(L1),prod(L2),1)) # result+totals   
rownames(result) <- c(1:10,"product")   
colnames(result) <- c("x","original","estimated","est > orig")   
knitr::kable(result)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 4.1: 使用正态随机值及相关正态似然函数的示例。估计列具有更大的总似然值。1=真，0=假。   |  | x | original | estimated | est > orig | | --- | --- | --- | --- | --- | | 1 | 5.585529 | 0.3360953 | 0.3320130 | 0 | | 2 | 5.709466 | 0.3101778 | 0.2868818 | 0 | | 3 | 4.890697 | 0.3965663 | 0.4901017 | 1 | | 4 | 4.546503 | 0.3599578 | 0.4536973 | 1 | | 5 | 5.605887 | 0.3320438 | 0.3246562 | 0 | | 6 | 3.182044 | 0.0764269 | 0.0574370 | 0 | | 7 | 5.630099 | 0.3271127 | 0.3158617 | 0 | | 8 | 4.723816 | 0.3840136 | 0.4827694 | 1 | | 9 | 4.715840 | 0.3831564 | 0.4819139 | 1 | | 10 | 4.080678 | 0.2614493 | 0.3073533 | 1 | | product | NA | 0.0000048 | 0.0000089 | 1 | |

[表 4.1](#tbl-4-1) 的最下面一行包含每列似然值的乘积(使用R函数 prod() 求得)。毫不奇怪，当我们使用样本的均值和标准差估计(估计，L2)而不是原始值的 *mean=5* 和 *sd=1.0* (原始，L1)时，求得最大似然，即 。在本例中，我可以确定这些值，因为在代码开始时使用了R函数 set.seed() ，以便在特定位置启动伪随机数生成器。如果你通常使用 set.seed()，不会重复使用相同的旧序列，例如 ，因为您可能会破坏伪随机数是随机数序列的良好近似值的想法，也许可以使用 getseed()来提供合适的种子数。

因此，rnorm() 函数提供了由均值和标准差确定分布中的伪随机数，dnorm() 函数提供了观测值在均值和标准差条件下的概率密度或似然（相当于 [公式 4.11](#eq-4_11) ）。累积概率密度函数（cdf）由函数 pnorm() 提供，量化值由 qnorm() 确定。平均值自然具有最大的似然。还要注意的是，正态曲线是围绕均值对称的。

# some examples of pnorm, dnorm, and qnorm, all mean = 0   
cat("x = 0.0 Likelihood =",dnorm(0.0,mean=0,sd=1),"\n")

x = 0.0 Likelihood = 0.3989423

cat("x = 1.95996395 Likelihood =",dnorm(1.95996395,mean=0,sd=1),"\n")

x = 1.95996395 Likelihood = 0.05844507

cat("x =-1.95996395 Likelihood =",dnorm(-1.95996395,mean=0,sd=1),"\n")

x =-1.95996395 Likelihood = 0.05844507

# 0.5 = half cumulative distribution   
cat("x = 0.0 cdf = ",pnorm(0,mean=0,sd=1),"\n")

x = 0.0 cdf = 0.5

cat("x = 0.6744899 cdf = ",pnorm(0.6744899,mean=0,sd=1),"\n")

x = 0.6744899 cdf = 0.75

cat("x = 0.75 Quantile =",qnorm(0.75),"\n") # reverse pnorm

x = 0.75 Quantile = 0.6744898

cat("x = 1.95996395 cdf = ",pnorm(1.95996395,mean=0,sd=1),"\n")

x = 1.95996395 cdf = 0.975

cat("x =-1.95996395 cdf = ",pnorm(-1.95996395,mean=0,sd=1),"\n")

x =-1.95996395 cdf = 0.025

cat("x = 0.975 Quantile =",qnorm(0.975),"\n") # expect ~1.96

x = 0.975 Quantile = 1.959964

# try x <- seq(-5,5,0.2); round(dnorm(x,mean=0.0,sd=1.0),5)

我们可以看到，单个似然值可能是相对较大的数字，但当它们相乘时，很快就会变成相对较小的数字。当观测数据的数量增加时，就会出现误差。即使只有十个数字，当我们将所有单个似然值相乘（使用 prod()）时，结果也会很快变得非常小。如果使用与 [表 4.1](#tbl-4-1) 中的十个数字类似的另外十个数字，总似然很容易降到 或 。随着观察次数的增加，出现四舍五入错误的几率（即使在 64 位计算机上）也开始增加。与其将许多小数相乘得到一个极小的数，不如将这些小数相乘的标准解决方案是对似然值进行自然对数变换，然后相加。最大化对数变换似然值之和与最大化单个似然值之积一样，都能获得最佳参数。此外，许多软件中的优化器似乎都是为了最有效地最小化某个函数而设计的。简单的解决方案是，我们不最大化单个似然的乘积，而是最小化负对数似然的总和（*-veLL* 或 negLL()）。

## 4.6 正态分布的似然

概率似乎是一种相当奇怪的生物。当它们来自连续的概率密度函数（PDFs）时，尽管它们具有许多相同的属性，但严格来说它们并不是概率（Edwards，1972 ）。严格来讲，它们与概率密度函数下某一点的概率密度有关。根据概率的定义，整条曲线下的面积总和必须为 1.0，但连续概率密度函数任何一点下的面积都会变得无穷小。正态似然的定义使用 [公式 4.11](#eq-4_11) ），而累积密度函数为:

我们可以使用 dnorm() 和 pnorm() 计算似然和累积密度函数（cdf）（ [图 4.5](#fig-4-5) ）。

# Density plot and cumulative distribution for Normal Fig 4.5   
x <- seq(-5,5,0.1) # a sequence of values around a mean of 0.0   
NL <- dnorm(x,mean=0,sd=1.0) # normal likelihoods for each X   
CD <- pnorm(x,mean=0,sd=1.0) # cumulative density vs X   
plot1(x,CD,xlab="x = StDev from Mean",ylab="Likelihood and CDF")   
lines(x,NL,lwd=3,col=2,lty=3) # dashed line as these are points   
abline(h=0.5,col=4,lwd=1)

|  |
| --- |
| 图 4.5: 一条虚线红色曲线描绘了均值为 0 和标准差为 1.0 的预期正态似然，以及作为黑色线的相同正态似然的累积密度。蓝线标识了累积概率为 0.5。 |

这听起来不错，但在这样的曲线下， 变量的特定值来说，确定这样一条曲线下的特定值意味着什么呢？在 [图 4.5](#fig-4-5) 中，我们用虚线表示图中的似然是局部估计，不构成连续的线。每个都表示在给定 值处的似然。如前所述，对于 和 的分布，在 值处的概率密度为 。让我们简单地查看一下似然和概率之间可能存在的混淆。如果我们考虑概率密度函数的一小部分，在 到 的值之间 ， 的概率密度函数，我们可能会看到类似 [图 4.6](#fig-4-6) 的情况:

#function facilitates exploring different polygons Fig 4.6   
plotpoly <- function(mid,delta,av=5.0,stdev=1.0) {   
 neg <- mid-delta; pos <- mid+delta   
 pdval <- dnorm(c(mid,neg,pos),mean=av,sd=stdev)   
 polygon(c(neg,neg,mid,neg),c(pdval[2],pdval[1],pdval[1],   
 pdval[2]),col=rgb(0.25,0.25,0.25,0.5))   
 polygon(c(pos,pos,mid,pos),c(pdval[1],pdval[3],pdval[1],   
 pdval[1]),col=rgb(0,1,0,0.5))   
 polygon(c(mid,neg,neg,mid,mid),   
 c(0,0,pdval[1],pdval[1],0),lwd=2,lty=1,border=2)   
 polygon(c(mid,pos,pos,mid,mid),   
 c(0,0,pdval[1],pdval[1],0),lwd=2,lty=1,border=2)   
 text(3.395,0.025,paste0("~",round((2\*(delta\*pdval[1])),7)),  
 cex=1.1,pos=4)   
 return(2\*(delta\*pdval[1])) # approx probability, see below   
} # end of plotpoly, a temporary function to enable flexibility   
 #This code can be re-run with different values for delta   
x <- seq(3.4,3.6,0.05) # where under the normal curve to examine   
pd <- dnorm(x,mean=5.0,sd=1.0) #prob density for each X value   
mid <- mean(x)   
delta <- 0.05 # how wide either side of the sample mean to go?   
parset() # a pre-defined MQMF base graphics set-up for par   
ymax <- getmax(pd) # find maximum y value for the plot   
plot(x,pd,type="l",xlab="Variable x",ylab="Probability Density",   
 ylim=c(0,ymax),yaxs="i",lwd=2,panel.first=grid())   
approxprob <- plotpoly(mid,delta) #use function defined above

|  |
| --- |
| 图 4.6: 正态分布变量 X 的概率密度，均值为 5.0，标准差为 1.0。在 x = 3.5 处的 PDF 值为 0.129518，因此红色框线所围成的矩形的面积是 0.0129518，这近似于 3.45 到 3.55 之间的总概率，实际上应该是曲线下的面积。 |

完整概率密度函数（ PDF） 下的面积总和为 1.0，因此得到 [图 4.6](#fig-4-6) 中 和 之间值的概率是长方形面积减去左三角形面积再加上右三角形面积之和。三角形几乎是对称的，因此可以近似地相互抵消，所以近似解法就是将其中一个长方形的面积乘以 。当 delta（长方形在 轴上的宽度）为 时，概率 。如果将 delta 值改为 ，那么近似概率 ，随着 delta 值的减小，总概率也随之减小，尽管 时的概率密度仍为 。显然，似然值与连续 PDF 中的概率并不相同（见 Edwards, 1972）。曲线下面积概率的最佳估计值为 pnorm(3.55,5,1) - pnorm(3.45,5,1)，即 = 0.0129585。

### 4.6.1 与平方和等价性

当使用正态似然来拟合数据模型时，我们实际做的是设置，使得每个可用观测值的负对数似然之和最小化。幸运的是，我们可以使用dnorm()来估计似然。事实上，如果我们使用正态分布残差或对数变换的对数正态分布数据使用极大似然方法拟合模型，则得到的参数估计与使用最小二乘法得到的参数估计相同(参见下文 [公式 4.19](#eq-4_19) 的推导和形式)。拟合模型需要生成一组预测值 (x-hat)，为其他自变量 的函数，其中 是函数关系中使用的参数列。 个观测值的对数似然定义为:

读作给定 个参数（ 和 ）时观测值 的对数似然；符号”|“读作”给定”。这个看似复杂的方程式其实可以大大简化。首先，我们可以将指数项之前的常数移到求和项之外，乘以 ，然后将剩余指数项的自然对数反变换为指数：

是数据方差的最大似然估计（记住是除以 而非 ）：

如果用 [公式 4.15](#eq-4_15) 中 代入 [公式 4.14](#eq-4_14) ，则 用 代替：

化简平方根意味着将 移至对数项外面， 变成 ，我们可以将平方根变成指数 ，然后将 项加到 项上：

将幂指数 移到第1个 *log* 项外：

然后将 简化，并将整个方程两边乘以 ，以转换为负对数似然，从而得到正态分布值的负对数似然的最终简化:

其中唯一的非常数部分是 的值，它是残差平方和除以 的平方根，所以现在应该很清楚了，为什么使用最大似然时获得的参数，如果使用正态随机误差，与从最小二乘方法得到的参数相同。

### 4.6.2 应用正态似然拟合数据模型

我们可以使用数据集 *LatA* 中的模拟雌性红鱼数据重复这个例子，图，

我们可以使用数据集 LatA中的模拟雌性红鱼数据来重复这个例子（ [图 4.7](#fig-4-7) ），我们用它来说明平方残差之和的使用。理想情况下，我们应该得到相同的答案，但估计值为 的估计值。**MQMF** 函数 plot1() 只是绘制单个图形（*type=“l”*或 *type=“p”*；请参阅 ?plot1）的一种快速方法，没有太多空白。如果你比我更喜欢空白，请编辑 plot1()！

#plot of length-at-age data Fig 4.7   
data(LatA) # load the redfish data set into memory and plot it   
ages <- LatA$age; lengths <- LatA$length   
plot1(ages,lengths,xlab="Age",ylab="Length",type="p",cex=0.8,   
 pch=16,col=rgb(1,0,0,1/5))

|  |
| --- |
| 图 4.7: LatA 数据集中包含的雌性红鱼 Centroberyx affinis 的年龄-长度数据。全彩表示 >= 5 个点。 |

现在，我们可以使用 **MQMF** 函数 negNLL()（负正态对数似然值）以确定使用正态随机误差的负对数似然值之和（negLL()对对数正态分布数据也有同样的作用）。如果你查看一下 negNLL() 的代码，就会发现它与 ssq() 一样，都是将一个函数作为参数传递给它，然后用它来计算每个输入年龄的预测平均值（在本例中是使用 **MQMF** 函数 vB()计算的年龄长度），然后使用 dnorm() 以预测值作为平均值，并使用数据中的年龄-长度观测值来计算 -veLL 之和。年龄数据是通过省略号（…）传递的，并没有在 negNLL() 中明确声明为参数。该函数在结构上与 ssq() 非常相似，输入要求完全相同，但 *pars* 是显式传递的，而不是在…中传递，因为 *pars* 的最后一个值必须是残差的 stdev，它将在 negNLL() 中使用，而不只是在 *funk* 中使用。因此，negNLL()的运行方式与ssq()非常相似，它是调用函数生成预测值的封装程序，然后使用 dnorm() 在每次调用时返回一个数字。因此，nlm() 会最小化 negNLL()，而 negNLL() 又会调用 vB()。

# Fit the vB growth curve using maximum likelihood   
pars <- c(Linf=27.0,K=0.15,t0=-3.0,sigma=2.5) # starting values   
 # note, estimate for sigma is required for maximum likelihood   
ansvB <- nlm(f=negNLL,p=pars,funk=vB,observed=lengths,ages=ages,   
 typsize=magnitude(pars))   
outfit(ansvB,backtran=FALSE,title="vB by minimum -veLL")

nlm solution: vB by minimum -veLL   
minimum : 747.0795   
iterations : 26   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 26.8354474 4.490629e-07  
2 0.1301554 1.659593e-05  
3 -3.5868868 2.852562e-07  
4 1.9500896 8.278354e-06

如果回顾一下 von Bertalanffy 曲线的 ssq() 例子，你会看到我们从358尾鱼的样本中得到SSQ值为 （试试 nrow(LatA)）。因此 ssq() 方法对 的估计值为 ，与预期的最大似然估计基本相同。

和以前一样，我们所需要做的就是把一个不同的生长曲线函数代入 negNLL() 就可得到结果。我们只需要记住在 向量中包含第四个参数( )。同样，使用正态随机误差得到的数值解与使用 ssq() 方法得到的数值解本质上是相同的。

#Now fit the Michaelis-Menton curve   
pars <- c(a=23.0,b=1.0,c=1.0,sigma=3.0) # Michaelis-Menton  
ansMM <- nlm(f=negNLL,p=pars,funk=mm,observed=lengths,ages=ages,   
 typsize=magnitude(pars))   
outfit(ansMM,backtran=FALSE,title="MM by minimum -veLL")

nlm solution: MM by minimum -veLL   
minimum : 743.6998   
iterations : 34   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 20.7464280 -6.195246e-06  
2 1.4183165 1.601881e-05  
3 0.9029899 2.461309e-04  
4 1.9317669 -3.359816e-06

同样，解决方案的梯度很小，这增加了解决方案不仅仅是局部最小值的信心，所以我们应该画出解决方案，看看它与数据的相对拟合。

通过在数据点上绘制拟合曲线，数据不会遮挡线条。现在可以从该分析中产生的实际预测，也可以与残值一起制成表格。通过包含单个残差的平方，可以更清楚地看出哪些点(见记录3)可能具有最大的影响。

#plot optimum solutions for vB and mm. Fig 4.8   
Age <- 1:max(ages) # used in comparisons   
predvB <- vB(ansvB$estimate,Age) #optimum solution   
predMM <- mm(ansMM$estimate,Age) #optimum solution   
parset() # plot the deata points first  
plot(ages,lengths,xlab="Age",ylab="Length",type="p",pch=16,   
 ylim=c(10,33),panel.first=grid(),col=rgb(1,0,0,1/3))   
lines(Age,predvB,lwd=2,col=4) # then add the growth curves  
lines(Age,predMM,lwd=2,col=1,lty=2)   
legend("bottomright",c("von Bertalanffy","Michaelis-Menton"),   
 col=c(4,1),lwd=3,bty="n",cex=1.2,lty=c(1,2))

|  |
| --- |
| 图 4.8: 雌性 LatA 红鱼数据拟合的最优 von Bertalanffy 和 Michaelis-Menton 生长曲线。注意两条曲线在观测数据最集中的区域几乎重合。注意 y 轴从 10 开始。 |

通常，我们会生成残差图来检查残差特征（ [图 4.9](#fig-4-9) ）。

# residual plot for vB curve Fig 4.9   
predvB <- vB(ansvB$estimate,ages) # predicted values for age data   
resids <- lengths - predvB # calculate vB residuals   
plot1(ages,resids,type="p",col=rgb(1,0,0,1/3),xlim=c(0,43),   
 pch=16,xlab="Ages Years",ylab="Residuals")   
abline(h=0.0,col=1,lty=2) # emphasize the zero line

|  |
| --- |
| 图 4.9: The residual values for von Bertalanffy curve fitted to the female LatA data. There is a clear pattern between the ages of 3 - 10, which reflects the nature of residuals when the mean expected length for a given age is constant and compared to these rounded length measurements |

生长数据的图（[图 4.8](#fig-4-8) 和 [图 4.9](#fig-4-9) ）中，数据的网格性质清楚地表明测量的长度精确到厘米，年龄四舍五入到最接近的整年。在 x 轴和 y 轴上的这种舍入与我们用经典的 y-on-x 方法拟合这些模型的问题相结合（Ricker, 1973），并且假设在 x 轴上的测量没有变化，但不幸的是，与年龄有关，这个假设是完全错误的。从本质上讲，我们将长度和年龄的变量视为离散的，而不是连续的，年龄数据是精确的，没有误差。这些特征值得进一步探索，但也有助于强调，当处理来自生活世界的数据时，很难收集，而且通常我们处理的信息并不完美。在渔业和现实世界的生态学中建立模型的真正诀窍是从那些不太完美的数据中获得有用和有趣的信息，并以一种站得住的方式这样做。

## 4.7 对数正态似然

正态分布具有高于或低于预期平均值的加性残差误差，这是众所周知的，其特性构成了许多有关自然的直觉的基础。然而，在开发种群中发现的许多变量（CPUE、渔获量、努力量……）都呈现出高度倾斜的分布，中心倾向并不位于分布的中心（正态分布就是这样）。对数正态分布（Log-Normal distribution）是用于描述此类数据的一种非常常见的 PDF 分布：

其中， 为数据点 的似然， 为 点变量实例的中位数， （ ）。用 的均值估算 ， 是 的标准差。

对数正态似然方程（[公式 4.20](#eq-4_20) ）在视觉上与正态分布相似，不同之处在于对数正态分布的残差是乘性的，而不是加性的。详细地说，每个似然乘以观测值的倒数，观测值和期望值进行对数变换。具体上讲，每个似然乘以观测值的倒数，观测值和期望值进行对数变换。对观测数据和期望值的对数变换意味着，我们不是使用 的变量，而是使用等价的 计算残差。这意味着观测值除以预期值，而不是从观测值中减去预期值。所有这些残差都将是正值，并将围绕 1.0 的值变化。残差为 1.0 意味着数据点完全符合该数据点的预期中值。

### 4.7.1 对数正态似然的简化

与正态似然一样，对数正态似然也可以简化，以方便后续计算：

的最大似然估计是：

再次注意方差的最大似然估计用 而非 。 [公式 4.21](#eq-4_21) 末尾的 项是常量，在拟合模型时经常忽略掉。如上所述，假设我们忽略 项，那么 [公式 4.22](#eq-4_22) 似乎与正态分布相同（ [公式 4.13](#eq-4_13) ）。然而，现在 需要将观测值和预测值进行对数转换（ [公式 4.22](#eq-4_22) ）。因此，只要在分析前对数据和预测值进行对数变换，我们就可以使用与正态似然相关的函数（如 negNLL()）来拟合使用对数正态残差的模型。不过，一般情况下，我们会使用 negLL()，它需要对数变换的观测值和一个生成预测值对数的函数（见 ?negLL）。

### 4.7.2 对数正态分布的性质

对于正态分布，我们知道分布的期望均值、中位数和模都是相同的，但对于对数正态分布，情况并非如此。给定一组连续变量 的值，其中位数的估计值为

其中 是 的均值，对数正态分布的模定义为：

其中 是 的标准差。最终，对数正态分布的均值或期望值定义为：

这些方程（ [公式 4.23](#eq-4_23) 到 [公式 4.25](#eq-4_25) ） 表明对数正态分布总是右斜（在模的右侧有一长尾）。此外，与正态分布相比，对数正态分布仅定义了 为正值的情形，见 [图 4.10](#fig-4-10) 。

# meanlog and sdlog affects on mode and spread of lognormal Fig 4.10   
x <- seq(0.05,5.0,0.01) # values must be greater than 0.0   
y <- dlnorm(x,meanlog=0,sdlog=1.2,log=FALSE) #dlnorm=likelihoods   
y2 <- dlnorm(x,meanlog=0,sdlog=1.0,log=FALSE)#from log-normal   
y3 <- dlnorm(x,meanlog=0,sdlog=0.6,log=FALSE)#distribution   
y4 <- dlnorm(x,0.75,0.6) #log=TRUE = log-likelihoods   
parset(plots=c(1,2)) #MQMF shortcut plot formatting function   
plot(x,y3,type="l",lwd=2,panel.first=grid(),   
 ylab="Log-Normal Likelihood")   
lines(x,y,lwd=2,col=2,lty=2)   
lines(x,y2,lwd=2,col=3,lty=3)   
lines(x,y4,lwd=2,col=4,lty=4)   
legend("topright",c("meanlog sdlog"," 0.0 0.6",   
" 0.0 1.0"," 0.0 1.2"," 0.75 0.6"),   
 col=c(0,1,3,2,4),lwd=3,bty="n",cex=1.0,lty=c(0,1,3,2,4))   
plot(log(x),y3,type="l",lwd=2,panel.first=grid(),ylab="")   
lines(log(x),y,lwd=2,col=2,lty=2)   
lines(log(x),y2,lwd=2,col=3,lty=3)   
lines(log(x),y4,lwd=2,col=4,lty=4)

|  |
| --- |
| 图 4.10: 两幅展示对数正态概率密度函数的图。左图是一组不同参数集的似然分布，右图是前四个对数正态分布的对数变换版本。 Two plots illustrating the Log-Normal probability density function. Left is a group of likelihood distributions for different parameter sets, while right is the log-transformed versions of these first four Log-Normal distributions. |

同样，可以从对数正态分布生成随机数，与之前一样，对数变换应生成正态分布, 如 [图 4.11](#fig-4-11) 所示。

set.seed(12354) # plot random log-normal numbers as Fig 4.11   
meanL <- 0.7; sdL <- 0.5 # generate 5000 random log-normal   
x <- rlnorm(5000,meanlog = meanL,sdlog = sdL) # values   
parset(plots=c(1,2)) # simplifies the plots par() definition   
hist(x[x < 8.0],breaks=seq(0,8,0.25),col=0,main="")   
meanx <- mean(log(x)); sdx <- sd(log(x))   
outstat <- c(exp(meanx-(sdx^2)),exp(meanx),exp(meanx+(sdx^2)/2))   
abline(v=outstat,col=c(4,1,2),lwd=3,lty=c(1,2,3))   
legend("topright",c("mode","median","bias-correct"),   
 col=c(4,1,2),lwd=3,bty="n",cex=1.2,lty=c(1,2,3))   
outh <- hist(log(x),breaks=30,col=0,main="") # approxnormal   
hans <- addnorm(outh,log(x)) #MQMF function; try ?addnorm   
lines(hans$x,hans$y,lwd=3,col=1) # type addnorm into the console

|  |
| --- |
| 图 4.11: 5000 个随机点的对数正态分布，meanlog=0.7，sdlog=0.5，显示偏差校正后的平均值、模式和中位数。右侧是拟合正态分布的对数变换版本。A Log-Normal distribution of 5000 random points with meanlog=0.7 and sdlog=0.5 showing the bias-corrected mean, the mode, and the median. On the right is the log-transformed version with a fitted normal distribution. |

我们可以检查输入参数的预期统计量，并将其与变量 outstat 中的参数估计进行比较。

#examine log-normal propoerties. It is a bad idea to reuse   
set.seed(12345) #'random' seeds, use getseed() for suggestions   
meanL <- 0.7; sdL <- 0.5 #5000 random log-normal values then   
x <- rlnorm(5000,meanlog = meanL,sdlog = sdL) #try with only 500   
meanx <- mean(log(x)); sdx <- sd(log(x))   
cat(" Original Sample \n")

Original Sample

cat("Mode(x) = ",exp(meanL - sdL^2),outstat[1],"\n")

Mode(x) = 1.568312 1.603512

cat("Median(x) = ",exp(meanL),outstat[2],"\n")

Median(x) = 2.013753 2.052606

cat("Mean(x) = ",exp(meanL + (sdL^2)/2),outstat[3],"\n")

Mean(x) = 2.281881 2.322321

cat("Mean(log(x) = 0.7 ",meanx,"\n")

Mean(log(x) = 0.7 0.7001096

cat("sd(log(x) = 0.5 ",sdx,"\n")

sd(log(x) = 0.5 0.4944283

中位数与均值之间的差异，即 项，被称为偏差修正项，并试图通过将中心倾向的测量值进一步向右偏离模式来解释分布的向右偏斜。模似乎位于最高组的左侧，但这只是反映了 R 绘制直方图的方式（您可以增加一半的二进制宽度来解决这个视觉问题）。

### 4.7.3 应用对数似然拟合曲线

我们可以使用 Penn 和 Caputi（1986）关于澳大利亚埃克斯茅斯湾老虎虾（*Penaeus semisulcatus*；MQMF 数据集 *tigers*）的产卵种群生物量及其补充量数据。种群补充关系通常假定为对数正态残差（有时为伽马分布，见后），通常在种群评估模型中得出。Penn 和 Caputi（1986 年）使用的是 Ricker 曲线，但作为替代方案，我们将尝试根据这些观测数据拟合 Beverton-Holt 种群补充量曲线（我们将在静态模型一章中更详细地研究种群补充量关系）。Beverton-Holt 种群补充量关系可以有多种形式，但在本例中我们将使用 [公式 4.26](#eq-4_26) ：

其中 是年 的补充量， 是繁殖亲体量，它能生产出 的补充量， 是渐近最大补充量水平， 是生产最大补充量的 50% 时的繁殖亲体量。残差误差是 、方差为 的对数分布，这些参数可以通过对数据进行模型拟合得到。我们将继续使用 negNLL()，但如果您检查一下 negNLL() 的代码，就会发现我们需要输入对数变换后的观测补充量水平，并编写一个简短函数来计算预测补充量水平的对数。我们将再次使用 nlm() 来最小化 negNLL() 的输出结果。当我们考虑 [表 4.2](#tbl-4-2) 中的对数正态残差（观测补充量/预测补充量）时，注意到有两个特殊的残差值，一个接近 ，另一个接近 。低点的潜在影响可能值得进一步研究，因为它是相对特殊的事件（事实上，它似乎受到气旋发生的影响，Penn 和 Caputi，1986 ）。

# fit a Beverton-Holt recruitment curve to tigers data Table 4.2   
data(tigers) # use the tiger prawn data set   
lbh <- function(p,biom) return(log((p[1]\*biom)/(p[2] + biom)))   
 #note we are returning the log of Beverton-Holt recruitment   
pars <- c("a"=25,"b"=4.5,"sigma"=0.4) # includes a sigma   
best <- nlm(negNLL,pars,funk=lbh,observed=log(tigers$Recruit),   
 biom=tigers$Spawn,typsize=magnitude(pars))   
outfit(best,backtran=FALSE,title="Beverton-Holt Recruitment")

nlm solution: Beverton-Holt Recruitment   
minimum : 5.244983   
iterations : 16   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 27.344523 -5.109267e-08  
2 4.000166 1.265602e-07  
3 0.351939 1.789283e-06

predR <- exp(lbh(best$estimate,tigers$Spawn))   
 #note exp(lbh(...)) is the median because no bias adjustment   
result <- cbind(tigers,predR,tigers$Recruit/predR)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 4.2: The Exmouth Gulf tiger prawn data set of spawning biomass with consequent recruitment levels, with the predicted recruitment level from the optimum model fit, along with the Log-Normal   | SpawnB | Recruit | PredR | Residual | | --- | --- | --- | --- | | 2.4 | 11.6 | 10.25393 | 1.1312736 | | 3.2 | 7.1 | 12.15284 | 0.5842256 | | 3.9 | 14.3 | 13.49891 | 1.0593447 | | 5.7 | 19.1 | 16.06816 | 1.1886865 | | 6.0 | 12.4 | 16.40644 | 0.7558007 | | 7.4 | 19.7 | 17.74969 | 1.1098783 | | 8.2 | 37.5 | 18.37886 | 2.0403883 | | 10.0 | 18.4 | 19.53157 | 0.9420646 | | 10.1 | 22.1 | 19.58698 | 1.1283005 | | 10.4 | 26.9 | 19.74859 | 1.3621223 | | 11.3 | 19.2 | 20.19541 | 0.9507111 | | 12.8 | 21.0 | 20.83372 | 1.0079815 | | 18.0 | 9.9 | 22.37262 | 0.4425051 | | 24.0 | 26.8 | 23.43802 | 1.1434411 | |

我们可以绘制出解，直观地比较数据拟合结果：

# Fig 4.12 visual examination of the fit to the tigers data   
plot1(tigers$Spawn,predR,xlab="Spawning Biomass","Recruitment",   
 maxy=getmax(c(predR,tigers$Recruit)),lwd=2)   
points(tigers$Spawn,tigers$Recruit,pch=16,cex=1.1,col=2)

|  |
| --- |
| 图 4.12: The optimum fit to the Exmouth Gulf tiger prawns Beverton-Holt stock recruitment relationship using log-normal likelihoods. |

### 4.7.4 应用对数正态误差拟合动态模型

本例中，我们将回顾第 [7](#sec-surplusproduction) 章“剩余生产模型 ”中的一个例子（并在第 [3](#sec-simplemodel) 章“简单种群模型 ”中查看）。具体而言，我们将使用所谓的Schaefer（1954，1957）剩余生产模型（Hilborn 和 Walters，1992；Polacheck 等，1993；Prager，1994；Haddon，2011）。用于描述 Schaefer 模型的最简单方程包含两个项：

其中 是内禀自然增长率（种群增长率项）， 是环境容纳量或未被捕捞时的生物量，一般用 表示（注意的是，这里的 简单地表示 时的初始生物量）， 可能已衰退低于 。 表示年 可用生物量，那么 是第一年中可用生物量。如果资源未被捕捞，那么 ，但在其他情况下，它将构成一个单独的模型参数。最后， 是年 中捕捞的总获量。当然，时间步长不必以年为单位，可能需要更短的时间，这取决于相关物种的生物学特性，不过，使用年是很常见的。注意 [公式 4.27](#eq-4_27) 中没有误差项。这意味着种群动态是确定的，渔获量是已知的，没有误差。估算该模型的参数就是使用观察误差估算的一个例子。

[公式 4.27](#eq-4_27) 中的简单动态模型，在所需参数（、 及 ）已知时，将初始生物量向前推算，生成种群生物量水平的时间序列。如果有相对丰度指数的时间序列（可能是调查得出的生物量估计值，也可能是渔业相关数据得出的标准化单位努力量渔获量（CPUE）），就可以将剩余产量模型与自然界的观测结果进行比较。在下面的例子中，我们将使用 **MQMF** 数据集 *abdat* 中潜水捕捞无脊椎动物的渔获量和 CPUE。假设相对丰度指数与种群生物量之间存在简单的线性关系：

其中 是年 的观测 CPUE， 是年 的努力量， 为可捕系数（Arreguin-Sanchez, 1996）， 表示CPUE 和 种群生物量关系的对数正态残差误差。 该 可以作为一个参数直接估算，尽管也有所谓的闭合形式估算值（Polacheck 等，1993）：

基本上就是观测到的 CPUE 向量的几何平均数除以预测的每年生物量水平。使用这种封闭形式的一个好处是，在拟合数据时，模型需要估计的参数较少。它还强调 参数只是一个比例因子，反映了可开发生物量与相对丰度指数之间的假定线性关系。如果假定存在非线性关系，则需要更复杂的可捕量表示方法。

上例中拟合的种群增殖曲线使用了一个相对简单的方程和相关函数作为拟合模型，但模型中不涉及动态变化。当试图将 [公式 4.27](#eq-4_27) 至 [公式 4.29](#eq-4_29) 中描述的剩余产量模型与观测到的 CPUE 和渔获量数据进行拟合时，得出预测 CPUE 的函数需要更加复杂，因为它需要包含种群动态。对于简单的非动态模型，自变量和因变量的使用相对简单。在此，我们将说明（并强化）在将动态模型拟合到数据时需要做哪些工作。

我们将使用名为 *abdat* 的 **MQMF** 数据集，该数据集因包含鲍鱼数据而得名。

data(abdat) # plot abdat fishery data using a MQMF helper Fig 4.13   
plotspmdat(abdat) # function to quickly plot catch and cpue

|  |
| --- |
| 图 4.13: The abdat data set plotting the catch and the cpue through time to illustrate their relationship. |

我们需要在 nlm() 中使用两个函数来找到最佳参数。您应该检查每个函数的代码，并了解它们与所用方程的关系。第一个函数用于计算预测 cpue 的对数（我们使用 simpspm()），第二个函数用于计算 *-veLL*，其中我们使用对数正态残差误差来表示 cpue 数据的残差（因此我们使用 negLL()；将其代码与 negNLL() 进行比较）。请注意模型参数将进行对数变换的预期，我们这样做是因为它通常比使用 *typsize* 选项更稳定。你应该通过尝试不同的起始点来实验这段代码。你应该仔细检查 simpspm 和 negLL() 的代码，直到理解它们之间的相互作用，并相信您可以用不同的数据集重复这一分析（参见第 [7](#sec-surplusproduction) 章“剩余产量模型 ” ）。

# Use log-transformed parameters for increased stability when   
 # fitting the surplus production model to the abdat data-set   
param <- log(c(r= 0.42,K=9400,Binit=3400,sigma=0.05))   
obslog <- log(abdat$cpue) #input log-transformed observed data   
bestmod <- nlm(f=negLL,p=param,funk=simpspm,indat=as.matrix(abdat),   
 logobs=obslog) # no typsize, or iterlim needed   
 #backtransform estimates, outfit's default, as log-transformed   
outfit(bestmod,backtran = TRUE,title="abdat") # in param

nlm solution: abdat   
minimum : -41.37511   
iterations : 20   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient transpar  
1 -0.9429555 6.743051e-06 0.38948  
2 9.1191569 -9.223729e-05 9128.50173  
3 8.1271026 1.059182e-04 3384.97779  
4 -3.1429030 -8.116218e-07 0.04316

将观察到的数据与最优模型的预测值对比绘制成图总是一个好主意。在这里，我们将它们绘制成对数比例，以准确说明拟合结果，并帮助确定哪些点与预测值相差最大。由于很少有人能直观地掌握对数标度，因此在标称标度上绘制数据和拟合结果也是一个好主意，此外，我们通常还会绘制残差图来寻找规律。

# Fig 4.14 Examine fit of predicted to data   
predce <- simpspm(bestmod$estimate,abdat) #compare obs vs pred   
ymax <- getmax(c(predce,obslog))   
plot1(abdat$year,obslog,type="p",maxy=ymax,ylab="Log(CPUE)",   
 xlab="Year",cex=0.9)   
lines(abdat$year,predce,lwd=2,col=2)

|  |
| --- |
| 图 4.14: The optimum fit of the Schaefer surplus production model to the abdat data set plotted in log-space on the y-axis. |

## 4.8 二项式分布的似然

到目前为止，通过正态分布和对数正态分布，我们一直在处理连续变量（如产卵生物量和 cpue）特定值的似然。当然，有些观测结果和事件在本质上是离散的。常见的例子包括动物是否成熟、是否有标签或类似的 “是”/“否”情况。与连续变量不同，当使用离散型分布（如二项分布）计算特定值（如在 个观测样本中看到 个标记的似然）的似然时，它们是真正的概率，而不仅仅是概率密度。这样，就避免了理解并非真实概率的似然的复杂性；这可能就是为什么许多介绍极大似然方法的文章倾向于从使用二项分布的例子开始的原因。

在观察结果有真有假的情况下（所谓伯努利试验；例如，捕获的鱼要么有标记，要么没有标记）， 次观察（试验）的成功概率为参数 ，那么通常最好使用二项分布来描述观察结果。二项分布概率密度函数产生真实概率，有两个参数：，即试验次数（样本大小）；，即试验成功的概率（事件/观察结果证明为真）：

读为给定 次试验中 个事件为真的概率（如 个观测样本中出现 个标记），其中 是事件为真的概率。项 常写作 ，即 。 “！”符号表示阶乘。方括号中的项是组合数，是从 个元素中一次取 个元素形成的组合数，有时写成:

总有 的情况，因为一个人不可能有比试验更多的成功。

### 4.8.1 二项式似然的示例

作为第一个示例子，我们可以从一个种群中只捕捞雄性（澳大利亚昆士兰州捕捞泥蟹（*Scylla serrata*）就是一个例子）。人们可能会问，这种管理策略是否会对特定种群中合法规格动物的性别比例产生负面影响。在一个假设的 只动物样本中，如果我们获得了 只雌性动物，而只有 只雄性动物，那么我们是否可以得出结论，渔业对性别比例产生了影响？回答这个问题的一种方法是研究出现这种结果的相对似然（虽然二项式似然是真正的概率，但我们仍将其称为似然）。典型的性别比为 ，这意味着我们可能会期望从 个样本中找到 只雄性动物，因此，如果我们在本例中宣布找到一只雄性动物是成功的，我们就应该研究不同 值的似然（ 个样本中雄性个体数），并确定在样本中找到 的相对似然。

#Use Binomial distribution to test biased sex-ratio Fig 4.15   
n <- 60 # a sample of 60 animals   
p <- 0.5 # assume a sex-ration of 1:1   
m <- 1:60 # how likely is each of the 60 possibilites?   
binom <- dbinom(m,n,p) # get individual likelihoods   
cumbin <- pbinom(m,n,p) # get cumulative distribution   
plot1(m,binom,type="h",xlab="Number of Males",ylab="Probability")   
abline(v=which.closest(0.025,cumbin),col=2,lwd=2) # lower 95% CI

|  |
| --- |
| 图 4.15: 如果性别比为 1:1，那么在 60 只动物样本中只观察到 20 只雄性动物的可能性有多大。垂直红线是 95% 置信区间的下限，表明只观察到 20 只的可能性不大。A graphical answer to the question of how likely is it to obtain only 20 males in a sample of 60 animals if the sex ratio is 1:1. The vertical red line is the lower bound of the 95% confidence intervals, which suggests that observing only 20 would be unlikely. |

我们可以通过检查 *binom*（单个似然）和 *cumbin*（累积二项式似然）的内容来研究具体值。但很明显，如果性别比为 ，那么获得少于 或多于 的可能性都非常小。事实上，只有 个雄性的可能性不到 95%。通过研究 *binom[20]*，我们可以看到，得到整整 个雄性（假设性别比为 1:1）的似然刚刚超过 的三分之一（），而 *cumbin[20]* 告诉我们，得到 个或更少的概率只有 ）。我们当然有理由说，从这一群中抽取样本的性别比率出现了下降。我们还在累积概率约为 的地方划了一条垂直线。由于我们对上限并不感兴趣，我们可以使用 ，这样会更保守一些。但是，这种上限具有任意性，真正重要的是，捕鱼对性别比率没有重要影响的证据权重是多少？

请注意，当样本相对较大时，二项分布会变得对称。不过，尾部比正态分布窄。对于较小的样本，尤其是低 值，二项分布可能会高度倾斜，零成功率值具有特定的概率。

与其确定一个给定的性别比例是否合理，我们可以寻找使 个样本中有 个雄性的似然最大化的性别比例（ 值）。我们期望它是 ，但仍然有兴趣知道合理值的范围。

# plot relative likelihood of different p values Fig 4.16   
n <- 60 # sample size; should really plot points as each independent   
m <- 20 # number of successes = finding a male   
p <- seq(0.1,0.6,0.001) #range of probability we find a male   
lik <- dbinom(m,n,p) # R function for binomial likelihoods   
plot1(p,lik,type="l",xlab="Prob. of 20 Males",ylab="Prob.")   
abline(v=p[which.max(lik)],col=2,lwd=2) # try "p" instead of "l"

|  |
| --- |
| 图 4.16: Representing the relative likelihood of the proportional sex-ratio when a sample exhibits only 20 males out of 60. Note that the likelihood of there having been a sex-ratio of 0.5 is confirmed as very low. |

当我们只搜索一个参数时，使用 optimize()（或 optimise()，查看 ?optimize）比使用 nlm() 更有效。它需要一个函数，该函数的第一个项是要在定义的区间内最小化或最大化的值，在本例中为 ，定义的区间内最小化或最大化的值，因此 dbinom() 使用了简单的封装函数。

# find best estimate using optimize to finely search an interval   
n <- 60; m <- 20 # trials and successes   
p <- c(0.1,0.6) #range of probability we find a male   
optimize(function(p) {dbinom(m,n,p)},interval=p,maximum=TRUE)

$maximum  
[1] 0.3333168  
  
$objective  
[1] 0.1087251

### 4.8.2 开放海湾幼年海狗种群数量

与其使用第二个假设的例子，不如使用一个真实案例的数据更有意思。对新西兰南岛西海岸开阔湾群岛上的新西兰海狗（*Arctocephalus forsteri*）幼崽种群进行了适当的研究（Greaves，1992 ；York 和 Kozlof，1987 ）。新西兰海狗在 19 世纪被大量捕杀后一直在恢复，目前在南岛和北岛都发现了新的出没地点。1894 年，新西兰正式停止了对毛皮海豹的捕猎，1978 年开始在新西兰专属经济区内对其进行全面保护（Greaves，1992 ）。格里夫斯女士与新西兰保护部合作，前往其中一个近海岛屿，并在岛上度过了一周的时间。她剪掉了 151 只海狗幼崽头上的一小块护毛，给它们做了标记，然后对海狗群进行了多次巡视，以重新观察被标记的海狗（Greaves，1992 ）。每次巡视都是进一步的抽样调查，每次抽样调查都发现了不同数量的被标记动物。问题是：海狗幼崽种群的数量（）是多少？

# Juvenile furseal data-set Greaves, 1992. Table 4.3   
furseal <- c(32,222,1020,704,1337,161.53,31,181,859,593,1125,   
 135.72,29,185,936,634,1238,153.99)   
columns <- c("tagged(m)","Sample(n)","Population(X)",   
 "95%Lower","95%Upper","StErr")   
furs <- matrix(furseal,nrow=3,ncol=6,dimnames=list(NULL,columns),   
 byrow=TRUE)

knitr::kable(furs)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 4.3: 格雷夫斯（1992）在新西兰南岛西海岸开阔湾岛对新西兰海狗幼崽进行的三次计数。种群估计值、标准误差和置信区间均采用确定性方程计算得出。前两行为独立计数，后一行为六次独立计数的平均值。Three of the counts of New Zealand fur seal pups made by Greaves (1992) on Open Bay Island, West Coast, South Island, New Zealand. The Population estimates, Standard error and confidence intervals were calculated using deterministic equations. The top two rows were independent counts while the bottom row averages six separate counts.   | tagged(m) | Sample(n) | Population(X) | 95%Lower | 95%Upper | StErr | | --- | --- | --- | --- | --- | --- | | 32 | 222 | 1020 | 704 | 1337 | 161.53 | | 31 | 181 | 859 | 593 | 1125 | 135.72 | | 29 | 185 | 936 | 634 | 1238 | 153.99 | |

所有标记实验的常规假设均适用；即我们处理的是一个封闭种群–没有迁入或迁出，在实验期间没有自然死亡或标记死亡，没有标记丢失，标记不会影响动物的重捕概率。最后，在不同的日子里，幼崽可以四处走动，因此看到的动物都是相互独立的。格里夫斯（1992）估计了所有这些影响，并在她的分析中考虑了这些影响。在对标记进行标记和重新观察后，利用彼得森估算器找到了确定性答案（Caughley, 1977; Seber, 1982）：

其中 是种群中标记数量， 是后续样本量， 是回捕的标记数， 是估算的资源量。另一种估计方法是对第二个样本的计数进行调整，以考虑到在这种情况下我们处理的是离散事件这一事实。这就是贝利调整（Bailey’s adjustment）（Caughley，1977）：

相关的标准误差估计值用于使用正态近似法计算近似的 95% 置信区间，从而在使用确定性方程时得出对称的置信区间。

这些方程可以用来证实 [表 4.3](#tbl-4-3) 中的估计。然而，与其使用确定性方程，一个好的替代方法是使用二项概率密度函数应用最大似然来估计总体大小 。

我们只估计一个参数，，即总体大小，这需要搜索使数据的似然最大化的总体大小。对于二项分布，，[公式 4.30](#eq-4_30) 给出了在标记比例为 的总体中，从 个样本中观察到 个标记个体的概率(Snedecor和Cochran, 1967, 1989;Forbes et al, 2011)。标记的幼崽比例与总体大小 和最初标记的幼崽数量( 151 只)有关。因此 。我们将重新分析 [表 4.3](#tbl-4-3) 中的前两个样本。

# analyse two pup counts 32 from 222, and 31 from 181, rows 1-2 in   
 # Table 4.3. Now set-up storage for solutions   
optsol <- matrix(0,nrow=2,ncol=2,   
 dimnames=list(furs[1:2,2],c("p","Likelihood")))   
X <- seq(525,1850,1) # range of potential population sizes   
p <- 151/X #range of proportion tagged; 151 originally tagged   
m <- furs[1,1] + 1 #tags observed, with Bailey's adjustment   
n <- furs[1,2] + 1 # sample size with Bailey's adjustment   
lik1 <- dbinom(m,n,p) # individaul likelihoods   
 #find best estimate with optimize to finely search an interval   
 #use unlist to convert the output list into a vector   
 #Note use of Bailey's adjustment (m+1), (n+1) Caughley, (1977)   
optsol[1,] <- unlist(optimize(function(p) {dbinom(m,n,p)},p,   
 maximum=TRUE))   
m <- furs[2,1]+1; n <- furs[2,2]+1 #repeat for sample2   
lik2 <- dbinom(m,n,p)   
totlik <- lik1 \* lik2 #Joint likelihood of 2 vectors   
optsol[2,] <- unlist(optimize(function(p) {dbinom(m,n,p)},p,   
 maximum=TRUE))

我们当然可以将结果制成表格，但更清楚的是将它们绘制为每个假设总体大小的似然(在这里是概率)。然后我们可以使用变量 *optsol* 中的 列来计算每种情况下的最佳总体大小。该图的优点是，人们可以立即看到每个样本的似然曲线的重叠，并获得任何百分位数置信区间都不是对称的印象。

# Compare outcome for 2 independent seal estimates Fig 4.17   
 # Should plot points not a line as each are independent   
plot1(X,lik1,type="l",xlab="Total Pup Numbers",   
 ylab="Probability",maxy=0.085,lwd=2)   
abline(v=X[which.max(lik1)],col=1,lwd=1)   
lines(X,lik2,lwd=2,col=2,lty=3) # add line to plot   
abline(v=X[which.max(lik2)],col=2,lwd=1) # add optimum   
 #given p = 151/X, then X = 151/p and p = optimum proportion   
legend("topright",legend=round((151/optsol[,"p"])),col=c(1,2),lwd=3,   
 bty="n",cex=1.1,lty=c(1,3))

|  |
| --- |
| 图 4.17: 通过标记实验，对151只海狗进行了标记，得出了海狗种群大小的两种估计，种群大小与似然分布(Greaves, 1992)。右边的黑线来自222只观察到的幼崽的32个标记，而左边的虚线来自181只观察到的31个标记。模型(最佳种群估计值)由竖线和图例表示。 |

### 4.8.3 使用多个独立样本

如果有多个调查、观测或样本，或不同类型的数据，而这些数据又是相互独立的，那么就有可能将这些估计值合并起来，以改进总体估计值。就像概率一样，一组独立观测数据的似然是特定观测数据似然的乘积。因此，我们可以将刚才研究的两个样本的每个种群大小的似然值相乘，得到一个联合似然值，在前面的例子中，这个似然值被放入变量 *totlik* 中，[图 4.18](#fig-4-18)。

#Combined likelihood from 2 independent samples Fig 4.18   
totlik <- totlik/sum(totlik) # rescale so the total sums to one   
cumlik <- cumsum(totlik) #approx cumulative likelihood for CI   
plot1(X,totlik,type="l",lwd=2,xlab="Total Pup Numbers",   
 ylab="Posterior Joint Probability")   
percs <- c(X[which.closest(0.025,cumlik)],X[which.max(totlik)],   
 X[which.closest(0.975,cumlik)])   
abline(v=percs,lwd=c(1,2,1),col=c(2,1,2))   
legend("topright",legend=percs,lwd=c(2,4,2),bty="n",col=c(2,1,2),   
 cex=1.2) # now compare with averaged count   
m <- furs[3,1]; n <- furs[3,2] # likelihoods for the   
lik3 <- dbinom(m,n,p) # average of six samples   
lik4 <- lik3/sum(lik3) # rescale for comparison with totlik   
lines(X,lik4,lwd=2,col=3,lty=2) #add 6 sample average to plot

|  |
| --- |
| 图 4.18: 前两个海狗幼崽样本的综合似然（黑线）。与单个样本相比，这些样本的百分位数置信区间（垂直细红线）更小。六个样本的平均计数和样本量（绿色虚线）在平均位置上与单个样本相似，但百分比置信区间更宽。图例显示了组合似然曲线的最佳值和 95% 百分位数 CI。The combined likelihood from the first two fur seal pup samples (black line). These have tighter percentile confidence intervals (vertical thin red lines) than the individual samples. The average count and sample size of six samples (green dashed line) remains similar in mean location to the single samples but would have much wider CI. The legend shows the optimum and the 95% percentile CI foir the combined likelihoods curve. |

请注意，虽然中心倾向仍然非常相似，但从组合似然值（红色细线）得出的 95% 置信区间比从六个组合样本得出的平均值要小，而且在平均值周围不对称。与所有分析一样，只要使用的程序是站得住脚的，那么分析就可以继续进行（例如，在这种情况下，可以说样本是独立的，这意味着它们在时间上相隔足够远，因此无法得知被标记幼崽的位置，幼崽也可以移动等）。这与贝叶斯方法利用新数据的似然更新先验概率有相似之处。具体的贝叶斯方法将在后面详细介绍。对六个样本取平均值以获得平均计数和样本大小的方法提供了非常相似的种群估计值，但其不确定性范围更大。取这些样本的平均值并不是最佳的分析策略。

### 4.8.4 分析方法

有些生物过程，如动物是否成熟、是否被渔具选择捕捞等，非常适合用二项分布作为观测的基础进行分析。然而，这类过程是以累积的方式运行的，例如，随着时间的推移，种群中成熟的群体百分比最终应达到 100%。这种过程通常可以用众所周知的 logistic 曲线来很好地描述。我们可以使用数值方法来估计 logistic 模型参数，也可以使用二项分布的广义线性模型。我们将在本章之后的静态模型一章中介绍这些方法。

## 4.9 其它分布

在基础 stats R 软件包中（使用 sessionInfo() 确定加载的 7 个基础软件包），有许多分布的概率密度函数。事实上，只要在控制台中输入 ?Distributions（不带括号），就能立即获得 R 中可用的分布列表。此外，CRAN 系统中还有一个关于分布的任务视图，网址是：https://CRAN.R-project.org/view=Distributions。该任务视图提供了详细的讨论，同时也指向了大量独立软件包，这些软件包提供了更广泛的统计分布。

在渔业科学中，有几种其他分布可以直接使用。它们都有用于计算似然的概率密度函数（以 *d* 开头，如 dmultinom()），而且通常都有随机数生成器（以 r 开头，如 rgamma()）。如果你查看一些函数的帮助，就会明白其中的含义和概要。下面我们将详细介绍几种对渔业和生态工作更有用的随机数生成器（另见 Forbes 等，2011 ）。

## 4.10 多项式分布的似然

如上文所述，当观测结果有两种可能结果（真/假、标记/未标记、成熟/未成熟）时，我们会使用二项分布。不过，在很多情况下，观测结果可能会有两种以上的离散结果，在这种情况下，我们可以使用多项分布。在处理长度或年龄样本内的频率分布时，就会出现这种情况；例如，随机给定一尾鱼，它的年龄可能只是众多年龄类别中的一个。在这种多变量意义上，多项式分布是二项分布的扩展。多项式分布是另一种离散分布，它提供的是不同的概率，而不仅仅是似然。

对于二项分布，我们使用 来表示似然。对于多项式，这需要扩展，这样就不是只有两个结果(一个概率 )，而是在 个观测值的样本中，我们有 个可能结果中的每一个的概率( )。多项式分布的概率密度函数为(Forbes et al .， 2011):

其中 是 个样本中 类事件发生的次数(在统计文献中通常称为试验)， 是 类可能事件中每种事件的单独概率。每种事件类型的期望为 ，其中 为样本量， 为事件类型 的概率。由于 PDF 中存在阶乘项，这可能导致数值溢出问题，因此进行对数转换:

实际上，对数变换后的阶乘项 还有 ，涉及 的和，其中 从 到 , 到 的步长总是常数，通常在计算中被省略，此外负对数似然在最小化中使用:

我们有理由费心研究概率密度函数的简化方法，因为 R 中已经开发出了非常适用于计算概率密度函数的函数。了解我们所使用的任何函数的作用始终是个好主意，这也是了解我们希望在分析中使用的任何统计函数的属性的明智之举。在某些情况下，例如 R 的 dmultinom() 用于多项式分布，它的帮助页面告诉我们，它目前还没有矢量化，因此在渔业中的使用有点笨拙。相反，我们将在 **MQMF** 函数 mnnegLL() 中对 [公式 4.37](#eq-4_37) 进行 R 运行。此外，在只处理几百个观测值时，计算速度的差异并不重要，但如果要重复计算一组数据的总对数似然几百万次（在渔业模型中使用马尔可夫链-蒙特-卡洛或 MCMC（有时为 McMC）时很有可能出现这种情况），那么任何节省时间的方法都是有价值的。我们将在讨论贝叶斯方法和其他方法来描述模型和数据中固有的不确定性时讨论这些问题。

### 4.10.1 使用多项式分布

在对年龄或大小组成数据进行模型拟合时，通常使用多项式分布来表示观测结果在可能类别中的预期分布。我们将以塔斯马尼亚德文特河口希望岛的黑唇鲍（*Haliotis rubra*）幼体的生长模态分析研究为例（Helidoniotis 和 Haddon，2012）。在 1992 年 11 月采集的样本中，以 2 毫米组距绘图时两个模式非常明显。

#plot counts x shell-length of 2 cohorts Figure 4.19   
cw <- 2 # 2 mm size classes, of which mids are the centers   
mids <- seq(8,54,cw) #each size class = 2 mm as in 7-9, 9-11, ...   
obs <- c(0,0,6,12,35,40,29,23,13,7,10,14,11,16,11,11,9,8,5,2,0,0,0,0)   
 # data from (Helidoniotis and Haddon, 2012)   
dat <- as.matrix(cbind(mids,obs)) #xy matrix needed by inthist   
parset() #set up par declaration then use an MQMF function   
inthist(dat,col=2,border=3,width=1.8, #histogram of integers   
 xlabel="Shell Length mm",ylabel="Frequency",xmin=7,xmax=55)

|  |
| --- |
| 图 4.19: The length-frequency counts of a sample of juvenile abalone from the south-east of Tasmania illustrating two modes taken in 1992. |

假设数据中的可观测模式与不同的世代或栖息地有关，并且我们希望估计每个世代的属性。使用正态概率密度函数来描述每种模式/队列中每个大小组的预期相对频率，并将其组合起来生成样本中 个 2mm 长度组中每一个的期望相对频率(Helidoniotis和Haddon, 2012)。需要五个参数，每个世代的均值和标准差，以及第一个世代中包含的观测总数的比例(第二个世代的比例是通过减去1.0得到的)。因此，使用 ，其中，在这种情况下，有 个世代，我们可以得到每个大小分组内观测值的期望比例。一种方法是计算每个 2mm 分组中心的相对似然，乘以样本总数，再根据分配给每个世代的比例进行调节[公式 4.38](#eq-4_38)。

其中 是每个分组 中观测的期望数， 表示分组数（这晨在7-55mm 之间为 2mm 的步长，中心值为 ）。 是世代 1 中发现的观测总数 的比例， 为每个世代 的平均大小， 为它们的标准差。 [公式 4.38](#eq-4_38) 最后一行的 是分组 中观测值的期望比例，其中 是每个分组的中点。

或者，更准确地说，我们可以从每个大小类的顶部的累积概率中减去每个大小类i的底部的累积概率。然而，一般来说，这在分析中几乎没有什么不同，所以这里我们将只关注第一种方法(您可以尝试自己实现替代方法来进行比较，因为有时相信打印出来的一切并不是一个好主意!)。

无论采用哪种方法，我们都需要定义两个正态分布，并求和它们的相对贡献。我们可以使用对数正态分布或伽马分布等其他累积统计分布来代替正态分布。如果没有令人信服的理由来证明使用其中一种分布是正确的，那么理想的做法是比较使用其他分布与现有数据的相对拟合程度。

为了获得预期频率与观测频率进行比较，有必要限制预期总数与观测总数大致相同。多项式的负对数似然值为

其中， 和 是 世代假设生成观察分布的均值和标准差。有 个分组和 2 个世代 ， 是分组 的观测频率， 是组合分布分组 内期望比例。 目标是最小化负对数似然，以找到 个正态分布参数和 的最佳组合。

#cohort data with 2 guess-timated normal curves Fig 4.20   
parset() # set up the required par declaration   
inthist(dat,col=0,border=8,width=1.8,xlabel="Shell Length mm",   
 ylabel="Frequency",xmin=7,xmax=55,lwd=2) # MQMF function   
 #Guess normal parameters and plot those curves on histogram   
av <- c(18.0,34.5) # the initial trial and error means and   
stdev <- c(2.75,5.75) # their standard deviations   
prop1 <- 0.55 # proportion of observations in cohort 1   
n <- sum(obs) #262 observations, now calculate expected counts   
cohort1 <- (n\*prop1\*cw)\*dnorm(mids,av[1],stdev[1]) # for each   
cohort2 <- (n\*(1-prop1)\*cw)\*dnorm(mids,av[2],stdev[2])# cohort   
 #(n\*prop1\*cw) scales likelihoods to suit the 2mm class width   
lines(mids,cohort1,lwd=2,col=1)   
lines(mids,cohort2,lwd=2,col=4)

|  |
| --- |
| 图 4.20: 根据对 1992 年塔斯马尼亚南部采样的幼鲍长度-频率计数的初始参数猜测得出的两个正态分布。Two Normal distributions from initial parameter guesses imposed upon the length-frequency counts of juvenile abalone from southern Tasmania sampled in 1992. |

两种模式的初始试验和误差猜测的中心估计值似乎是合理的，但左侧世代中的散布似乎太小，世代间的比例分配似乎偏向于第一个世代。由于比例会以非线性方式改变数值，因此寻找最佳似然值对起始值很敏感。我们可以将 nlm() 应用于一个封装函数来生成多项式的负对数似然估计值，从而寻找一个更理想的参数集，就象 [公式 4.39](#eq-4_39) 中定义的那样。如前所述，我们需要一个函数来生成每个大小分组中观测值的预测数，在 **MQMF** 中我们称之为 predfreq()。我们还需要一个封装函数来计算使用 predfreq() 函数的负对数似然，这里我们将其称为 wrapper()（见下面的代码块）。在开发 predfreq() 函数时，我们要求对参数进行排序，首先是拟合的各世代的均值，然后是标准差，最后是分配给除最后一世代之外的所有世代的比例。这样，算法就能在规定的位置找到所需的参数。因此，对于三个世代的问题来说， 在 R 语言中，我们将有 。我们还加入了使用累积正态概率密度的选项。

#wrapper function for calculating the multinomial log-likelihoods   
 #using predfreq and mnnegLL, Use ? and examine their code   
wrapper <- function(pars,obs,sizecl,midval=TRUE) {   
 freqf <- predfreq(pars,sum(obs),sizecl=sizecl,midval=midval)   
 return(mnnegLL(obs,freqf))   
} # end of wrapper which uses MQMF::predfreq and MQMF::mnnegLL   
mids <- seq(8,54,2) # each size class = 2 mm as in 7-9, 9-11, ...   
av <- c(18.0,34.5) # the trial and error means and   
stdev <- c(2.95,5.75) # standard deviations   
phi1 <- 0.55 # proportion of observations in cohort 1   
pars <-c(av,stdev,phi1) # combine parameters into a vector   
wrapper(pars,obs=obs,sizecl=mids) # calculate total -veLL

[1] 708.3876

# First use the midpoints   
bestmod <- nlm(f=wrapper,p=pars,obs=obs,sizecl=mids,midval=TRUE,   
 typsize=magnitude(pars))   
outfit(bestmod,backtran=FALSE,title="Using Midpts"); cat("\n")

nlm solution: Using Midpts   
minimum : 706.1841   
iterations : 27   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 18.3300619 6.369667e-06  
2 33.7907454 4.467972e-06  
3 3.0390094 -2.839356e-05  
4 6.0306017 6.993965e-06  
5 0.5763628 7.495453e-05

#Now use the size class bounds and cumulative distribution   
 #more sensitive to starting values, so use best pars from midpoints   
X <- seq((mids[1]-cw/2),(tail(mids,1)+cw/2),cw)   
bestmodb <- nlm(f=wrapper,p=bestmod$estimate,obs=obs,sizecl=X,   
 midval=FALSE,typsize=magnitude(pars))   
outfit(bestmodb,backtran=FALSE,title="Using size-class bounds")

nlm solution: Using size-class bounds   
minimum : 706.1815   
iterations : 24   
code : 1 gradient close to 0, probably solution   
 par gradient  
1 18.3299573 2.356852e-06  
2 33.7903327 -4.690083e-06  
3 2.9831560 2.221789e-05  
4 6.0030194 -2.882406e-05  
5 0.5763426 -5.168099e-05

现在对照原始数据绘制这些最优解。绘制数据图非常简单，但随后我们需要找出每种情况下的最优参数，并计算出每个参数的隐含正态分布。

#prepare the predicted Normal distribution curves   
pars <- bestmod$estimate # best estimate using mid-points   
cohort1 <- (n\*pars[5]\*cw)\*dnorm(mids,pars[1],pars[3])   
cohort2 <- (n\*(1-pars[5])\*cw)\*dnorm(mids,pars[2],pars[4])   
parsb <- bestmodb$estimate # best estimate with bounds   
nedge <- length(mids) + 1 # one extra estimate   
cump1 <- (n\*pars[5])\*pnorm(X,pars[1],pars[3])#no need to rescale   
cohort1b <- (cump1[2:nedge] - cump1[1:(nedge-1)])   
cump2 <- (n\*(1-pars[5]))\*pnorm(X,pars[2],pars[4]) # cohort 2   
cohort2b <- (cump2[2:nedge] - cump2[1:(nedge-1)])

#plot the alternate model fits to cohorts Fig 4.21   
parset() # set up required par declaration; then plot curves   
pick <- which(mids < 28)   
inthist(dat[pick,],col=0,border=8,width=1.8,xmin=5,xmax=28,   
 xlabel="Shell Length mm",ylabel="Frequency",lwd=3)   
lines(mids,cohort1,lwd=3,col=1,lty=2) # have used setpalette("R4")   
lines(mids,cohort1b,lwd=2,col=4) # add the bounded results   
label <- c("midpoints","bounds") # very minor differences   
legend("topleft",legend=label,lwd=3,col=c(1,4),bty="n",   
 cex=1.2,lty=c(2,1))

|  |
| --- |
| 图 4.21: 长度-频率信息左侧模式的最佳拟合。使用长度分组中值或边界值的参数值差异出现在小数点后第四位。右侧模式的差异很小，无法辨别。The optimum fit to the left-hand mode of the length-frequency information. The differences in parameter values between using size-class midpts or bounds occurs at the fourth decimal place. The difference in the right-hand mode was so small as not to be discernible. |

在这种情况下，使用不同长度组的中点所得到的参数与使用每个长度组的上限和下限所得到的参数几乎没有差别，所得到的正态曲线几乎完全重合。我们可以通过列出每种情况下的观测计数和预测计数，以及每个长度组中预测数和观测数之间的差异来了解这些拟合的细节。如 [图 4.21](#fig-4-21) 所示， [表 4.4](#tbl-4-4) 中的数字本身更清楚地说明了两种方法的接近性。从严格意义上讲，使用上下限更为正确，但在实践中往往差别不大。

Venebles 和 Ripley（2002，p 436）采用不同的策略将模态分布拟合为混合分布。他们的方法更为复杂和优雅，因为他们使用了分析梯度来辅助模型拟合过程。希望大家都能清楚地认识到，几乎所有分析问题都有不止一种解决方法。使用数值方法往往需要探索其他方法来寻找解决方案。

# setup table of results for comparison of fitting strategies   
predmid <- rowSums(cbind(cohort1,cohort2))   
predbnd <- rowSums(cbind(cohort1b,cohort2b))   
result <- as.matrix(cbind(mids,obs,predmid,predbnd,predbnd-predmid))   
colnames(result) <- c("mids","Obs","Predmid","Predbnd","Difference")   
result <- rbind(result,c(NA,colSums(result,na.rm=TRUE)[2:5]))

knitr::kable(result)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 4.4: A tabulation of the predicted counts for the two normal distributions from the optimum model fit. The original sample had 262 observations.   | mids | Obs | Predmid | Predbnd | Difference | | --- | --- | --- | --- | --- | | 8 | 0 | 0.1243773 | 0.1487018 | 0.0243245 | | 10 | 0 | 0.9323332 | 1.0428918 | 0.1105586 | | 12 | 6 | 4.5513306 | 4.8235655 | 0.2722349 | | 14 | 12 | 14.4343959 | 14.6974970 | 0.2631010 | | 16 | 35 | 29.7390161 | 29.5254720 | -0.2135441 | | 18 | 40 | 39.8898972 | 39.2109794 | -0.6789178 | | 20 | 29 | 35.1655925 | 34.7612096 | -0.4043829 | | 22 | 23 | 21.2937759 | 21.4732877 | 0.1795118 | | 24 | 13 | 10.8867799 | 11.2235972 | 0.3368173 | | 26 | 7 | 8.0156286 | 8.1948201 | 0.1791915 | | 28 | 10 | 9.5119745 | 9.5509907 | 0.0390161 | | 30 | 14 | 12.0773614 | 12.0505122 | -0.0268492 | | 32 | 11 | 14.0532683 | 13.9953724 | -0.0578959 | | 34 | 16 | 14.6762590 | 14.6093929 | -0.0668661 | | 36 | 11 | 13.7319605 | 13.6776679 | -0.0542926 | | 38 | 11 | 11.5102599 | 11.4832323 | -0.0270277 | | 40 | 9 | 8.6431354 | 8.6453958 | 0.0022604 | | 42 | 8 | 5.8142145 | 5.8367730 | 0.0225585 | | 44 | 5 | 3.5038405 | 3.5336705 | 0.0298300 | | 46 | 2 | 1.8916081 | 1.9184137 | 0.0268057 | | 48 | 0 | 0.9148534 | 0.9339392 | 0.0190857 | | 50 | 0 | 0.3963743 | 0.4077103 | 0.0113361 | | 52 | 0 | 0.1538484 | 0.1596018 | 0.0057535 | | 54 | 0 | 0.0534951 | 0.0560238 | 0.0025287 | | NA | 262 | 261.9655804 | 261.9607187 | -0.0048617 | |

## 4.11 Gamma 分布的似然

与我们在前几节中所考虑的统计分布相比，Gamma 分布通常不太为人所知。然而，Gamma 分布在渔业建模和模拟中越来越常用；在基于长度的种群模型中可以找到实际的例子(Sullivan等，1990；Sullivan,1992)。Gamma分布的概率密度函数有两个参数，一个尺度参数 （ ；有时使用的替代方式是 ，其中 ），以及一个形状参数 （）。分布范围是 （即无负值）。分布的期望或均值 与两个参数（尺度参数 和形状参数 ）有关。因此：

计算 Gamma 分布的单个似然的概率密度函数为（Forbes et al, 2011）：

其中 是变量值 ， 是尺度参数， 是形状参数， 是 参数的 gamma 函数。 取整数值的情况下，分布还被称为 Erlang 分布，此时 gamma 函数（）替代为阶乘 (Forbes 等，2011)：

与通常的似然计算一样，为了避免过流和欠流的计算问题，标准的做法是计算对数似然，更具体地说是负对数似然。使用对数似然总是更能规避风险：

对于多个观测值，这可以将它们相加而不是相乘，但仍需要计算对数 gamma 函数 。幸运的是在 R 语言中可以使用 gamma() 和 lgamma() 两个函数。

Gamma 分布（不要与 gamma 函数 混淆）非常灵活，形状的范围从实际上的反向曲线，到右偏曲线，再到近似正态曲线。它的灵活性使其成为一个非常有用的模拟函数，[图 4.22](#fig-4-22)。

#Illustrate different Gamma function curves Figure 4.22   
X <- seq(0.0,10,0.1) #now try different shapes and scale values   
dg <- dgamma(X,shape=1,scale=1)   
plot1(X,dg,xlab = "Quantile","Probability Density")   
lines(X,dgamma(X,shape=1.5,scale=1),lwd=2,col=2,lty=2)   
lines(X,dgamma(X,shape=2,scale=1),lwd=2,col=3,lty=3)   
lines(X,dgamma(X,shape=4,scale=1),lwd=2,col=4,lty=4)   
legend("topright",legend=c("Shape 1","Shape 1.5","Shape 2",   
 "Shape 4"),lwd=3,col=c(1,2,3,4),bty="n",cex=1.25,lty=1:4)   
mtext("Scale c = 1",side=3,outer=FALSE,line=-1.1,cex=1.0,font=7)

|  |
| --- |
| 图 4.22: Different Gamma distributions all with a scale c = 1.0. |

在 “多项式分布”一节中，我们使用正态分布来描述一个世代中长度的预期分布。也有可能使用 Gamma 分布来更好地描述物种的生长模式。

## 4.12 Beta 分布的似然

Beta 分布只适用于 0.0 到 1.0 之间的变量值。这使它成为模拟中另一个非常有用的分布，因为从 0 到 1 之间的分布中采样是比较常见的，而不可能获得超出这些限制的值，如 [图 4.23](#fig-4-23) 。在 “不确定性”一章中，我们将使用多变量正态分布，这需要使用一个额外的 R 软件包。可用于分析和模拟的分布阵列非常庞大。探索是发现其特性的最佳途径。

#Illustrate different Beta function curves. Figure 4.23   
x <- seq(0, 1, length = 1000)   
parset()   
plot(x,dbeta(x,shape1=3,shape2=1),type="l",lwd=2,ylim=c(0,4),   
 yaxs="i",panel.first=grid(), xlab="Variable 0 - 1",   
 ylab="Beta Probability Density - Scale1 = 3")   
bval <- c(1.25,2,4,10)   
for (i in 1:length(bval))   
 lines(x,dbeta(x,shape1=3,shape2=bval[i]),lwd=2,col=(i+1),lty=c(i+1))   
legend(0.5,3.95,c(1.0,bval),col=c(1:7),lwd=2,bty="n",lty=1:5)

|  |
| --- |
| 图 4.23: Different Beta probability density distributions all with a shape1 value = 3.0, and with values of shape2 ranging from 1 to 10. |

## 4.13 贝叶斯定理

### 4.13.1 简介

贝叶斯统计在渔业科学中的应用不断扩大（McAllister 等，1994；McAllister 和 Ianelli，1997；Punt 和 Hilborn，1997；反对意见见 Dennis，1996；最近见 Winker 等，2018）。Gelman 等（2014）出版了一本与这些方法的使用相关的优秀书籍。在此，我们不打算对渔业中使用的方法进行回顾；Punt 和 Hilborn（1997）中有很好的介绍，而且还有许多更近期的例子。相反，我们将集中讨论渔业中使用的贝叶斯方法的基础，并与最大似然法进行一些比较。关于如何使用这些方法的详情，请参阅 “不确定性”一章。

条件概率用于描述人们对特定事件发生的概率感兴趣的情况，比如在事件 已经发生的情况下，事件 发生的概率。这个表达式的符号是 ，其中垂直线“|” 表示“给定（given）”的意思。

贝叶斯定理就是基于对这种条件概率的操纵。因此，如果一组 个事件，标记为 ，在事件 发生的情况下发生，那么我们可以正式地发展贝叶斯定理。首先，我们考虑在 发生的情况下观察到特定 的概率:

也就是说，在 已经发生的情况下， 发生的概率等于 和 同时发生的概率除以 发生的概率。以同样的方式，我们可以考虑给定事件 发生的情况下事件 的条件概率。

这两种条件概率可以重新排列，得到:

我们使用这个代替 [公式 4.44](#eq-4_44) 中的 项可得到 传统的贝叶斯定理：

如果我们把这个看似晦涩的定理翻译成这样，即 代表从自然界获得的数据，而各种 作为可以用来解释数据的单独假设（假设是模型加上参数向量 ），那么我们就可以推导出渔业中使用的贝叶斯定理的形式。因此， 就是给定模型加上参数 （假设）的数据 的似然;我们已经从极大似然理论和实践中熟悉了这一点。新内容是 ，它是在对数据 进行任何分析或考虑之前假设的概率。这就是假设 的**先验**概率。 [公式 4.47](#eq-4_47) 中的 是所有可能的数据和假设组合的组合概率。“所有”这个词的使用需要强调，因为它实际上意味着考虑所有可能的结果。这种完备性就是为什么贝叶斯统计在纸牌游戏和其他受限制的机会游戏等封闭系统中如此有效的原因。然而，在开放世界中，所有的可能性都被考虑过是一个强有力的假设。最好的解释是，所考虑的一系列备选假设（模型加特定参数）构成了正在进行的分析的适用范围。b也就是对于所有 ， ，因此得到了 [公式 4.48](#eq-4_48)。

### 4.13.2 贝叶斯方法

如前所述，贝叶斯定理与条件概率有关(Gelman et al, 2014)，因此，当我们试图确定一系列 个离散假设（ ）中哪个是最可能的时，我们使用:

其中 指的是被考虑的 个假设中的 （假设是具有特定参数值集的特定模型），而数据只是模型拟合的数据。重要的是， 是假设 的**后验**概率(即 0 到 1 之间的严格概率)。这定义了除数 ，重新调整后的概率总和为 1.0。在考虑观察数据之前， 是假设的先验概率（模型加上特定的参数值）。同样，这是一个严格的概率，所有假设的先验值之和必须为 1。最后， 是给定假设 时数据的似然，正如之前在最大似然法部分所讨论的那样。

如果参数是连续变量（如von Bertalanffy 曲线中 和 ），替代假设必须使用连续参数的向量来描述，而不是使用离散参数集的列表，这样贝叶斯条件概率就变成了连续的:

在渔业和生态学中，要使用贝叶斯定理生成所需的后验分布，我们需要三样东西：

1. 要考虑的模型假设列表（即我们要尝试的参数和模型的组合（或范围））；
2. 计算每个假设 下观测数据的概率密度所需的似然函数, ；
3. 每个假设 的先验概率，标准化后所有先验概率之和等于 1.0。

除了对一组先验概率的要求之外，这与确定最大似然的要求完全相同。然而，先验概率的引入是一个很大的区别，也是我们将重点讨论的内容。这种方法的本质是利用数据信息更新先验概率。

如果模型中有许多参数需要估计，那么确定特定问题中的后验概率所涉及的整合工作就会耗费大量的计算机时间。用于确定贝叶斯后验分布的技术有很多，Gelman 等（2014）介绍了比较常用的方法。我们将介绍和讨论一种估算贝叶斯后验概率的灵活方法（MCMC），我们将在讨论不确定性特征的章节中使用这种方法。这实际上是一种新的模型拟合方法，但为了方便起见，我们将把它包含在有关不确定性的章节中。贝叶斯分析法的明确目标不仅仅是发现后验分布的模式，从最大似然的角度看，后验分布可能被认为代表了最优模型。相反，其目的是明确描述分析得出的不同可能结果的相对概率，即描述每个参数和模型输出的不确定性。可能会有一个最可能的结果，但它是在所有其他可能性的概率分布的背景下提出的。

### 4.13.3 先验概率

对于如何确定先验概率没有任何限制。人们可能已经从以前对同一物种的同一种群或不同种群的研究中对模型的参数有了很好的估计，或者至少对参数有了有用的限制（如不可能出现负增长或存活率 >1）。如果没有足够的信息来产生有参考价值的先验概率，通常会采用一组均匀或无信息的先验概率，即所有被考虑的假设都被赋予相等的先验概率。这样做的效果是在分析前给每个假设赋予相同的权重。当然，如果在分析中不考虑某一假设，这就等于给该假设（模型加上特定参数）分配了一个零权重或先验概率。

使用先验概率的想法之所以如此吸引人，原因之一是认为所有可能的参数值都具有同等可能性的想法有违直觉。渔业和生物学方面的任何经验都会让人对生物体的自然制约因素有先验知识。因此，举例来说，即使在彻底取样之前，就应该预料到深水（大于 800 米水深）鱼类物种，如橙连鳍鲑（*Hoplostethus atlanticus*），很可能寿命长、生长慢。这一特征反映了生活在低温和低生产力环境中的影响。贝叶斯方法的一大优势是，它允许人们摆脱所有可能性都具有同等可能性的反直觉假设。我们可以尝试在先验分布中捕捉模型中不同参数值的相对可能性。这样，先验知识就可以直接纳入分析。

使用先前信息可能引起争议的地方是在纳入意见时。例如，可以召集渔业利益相关者，了解他们对当前生物量等参数状况的看法（也许是相对于五年前的看法）。这种以委员会为基础的参数先验概率分布可以很容易地纳入贝叶斯分析，就像单独评估的结果一样（不是以前的评估，人们会认为以前的评估使用了大部分相同的数据，而是使用了独立的数据）。在正式分析中，是否应同样接受来自这些不同来源的先验数据，经常引起争论。Punt 和 Hilborn（1997，第 43 页）在一篇可读性很强的文章中讨论了先验来源的合理性问题：

*因此，我们强烈建议，无论何时进行贝叶斯评估，都应非常谨慎地充分记录各种先验分布的依据….。在选择先验函数形式时应小心谨慎，因为选择不当会导致不正确的推论。我们还注意到一种低估不确定性的倾向，即指定不切实际的先验信息–这种倾向应得到明确承认并加以避免。*

辩论的有效性的使用丰富的案底已经这样，walters 和 Ludwig （1994）建议非翔实的先验被用来作为一个默认在贝股的评估。 然而，除了不同意沃尔特斯和路德维希，平底船和Hilborn(1997年)突出强调的一个问题与我们的能力产生非翔实的前科(Box 和 Tiao,1973). 一个问题产生非翔实的先验是，他们感到特别测量系统( [图 4.24](#fig-4-24)). 因此，现有概率密度，是统一的线性规模将不代表一个统一的密度在一个日志的规模。

# can prior probabilities ever be uniniformative? Figure 4.24   
x <- 1:1000   
y <- rep(1/1000,1000)   
cumy <- cumsum(y)   
group <- sort(rep(c(1:50),20))   
xlab <- seq(10,990,20)   
par(mfrow=c(2,1),mai=c(0.45,0.3,0.05,0.05),oma=c(0.0,1.0,0.0,0.0))   
par(cex=0.75, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
yval <- tapply(y,group,sum)   
plot(x,cumy,type="p",pch=16,cex=0.5,panel.first=grid(),   
 xlim=c(0,1000),ylim=c(0,1),ylab="",xlab="Linear Scale")   
plot(log(x),cumy,type="p",pch=16,cex=0.5,panel.first=grid(),   
 xlim=c(0,7),xlab="Logarithmic Scale",ylab="")   
mtext("Cumulative Probability",side=2,outer=TRUE,cex=0.9,font=7)

|  |
| --- |
| 图 4.24: A constant set of prior probabilities accumulated on a linear scale and on a log scale. |

由于渔业模型往往充满非线性关系，使用非信息先验是有争议的，因为对某些参 数非信息先验很可能对其他参数有信息影响。虽然这种影响可能是无意的，但却不容忽视。这意味着信息可能以完全无意的方式被纳入模型，这也是讨论先验概率时争议的一个来源。如果要使用先验概率，那么 Punt 和 Hilborn（1997 年）关于充分记录先验概率的起源和属性的建议是非常明智的。我们将在 “不确定性”一章中更详细地探讨贝叶斯方法的使用。

## 4.14 结语

在任何分析情况下，使用哪种方法最合适在很大程度上取决于分析的目标。如果只想找到模型的最佳拟合方法，那么使用最小二乘法、最大似然法还是贝叶斯法其实并不重要。有时，先用最小二乘法拟合模型，然后再用似然法或贝叶斯法建立置信区间和进行风险评估，可能会更容易一些。

模型参数和输出的置信区间可以用传统的渐近方法（保证对称，对于强非线性模型，只能大致近似）、似然法或贝叶斯后验积分法（两者显然密切相关）来生成，也可以用引导法或蒙特卡洛技术。

并不是说只有使用贝叶斯方法才能更详细地评估备选管理方案的相对风险。Bootstrapping 和 Monte Carlo 方法为开展此类工作提供了必要的工具。首要问题是确定分析目标。然而，如果在拟合模型时，不对每个参数的不确定性以及模型对各种输入参数动态的敏感性有一定的了解，那将是一种糟糕的做法。由于各种方法之间没有明显的优胜者，如果有时间，使用一种以上的方法（特别是比较似然概 念、贝叶斯后验和引导法）是一个合理的想法。至于是否有足够的资源让资源建模者有时间进行这一系列分析，则是一个不同但同样现实的问题。如果发现存在重大差异，那么最好对其背后的原因进行调查。如果不同的程序得出的答案大相径庭，则可能是对现有数据的要求过高，需要进行不同的分析。

# 5. 静态模型

## 5.1 简介

在本书中，我们将重点讨论相对简单的种群模型，但也将开始为理解更复杂模型的某些要求做准备。我们所说的 “更复杂”是指通常用于模拟捕捞种群动态的年龄结构模型。然而，我们可以利用这些更复杂模型的组成部分来说明我们可以称之为 “静态模型”的数据拟合。静态模型用于描述个体生长过程的方程形式、成熟度如何随年龄或体型变化、补充与成熟或产卵生物量之间的 “种群-补充”关系，以及渔具对被捕捞种群的选择性。这种静态模型显示了相关变量（年龄长度、成熟年龄或长度等）之间的函数关系，而且这些关系被赋予了一个相当大的假设，即随着时间的推移保持不变。

这种静态模型与我们所说的动态模型形成对比，动态模型试图描述诸如资源量变化之类的过程，其中估算的值是早期资源量的函数。这样的动态模型总是包含一个自我参考的元素，可能还有其他驱动因素，所有这些都有助于我们试图描述或建模的动态。

在某些情况下，如果假定静态或稳定的过程发生了明显的变化，这些本应是静态的关系可能会被 “时间阻断”，这意味着假定这些关系在两个不同的时间段之间以阶梯状的方式发生变化。实际上，如果这些过程发生了变化，很可能是在一段时期内比较平稳地发生了变化，但遗憾的是，拥有足够的数据来很好地描述这种渐进的变化通常是不可能的。

我们首先用这类模型来实现更多的例子，因为正如我们在第 [4](#sec-paraestimat) 章 “[模型参数估计](04-application.qmd)（*Model Parameter Estimation*）”中所看到的，这些静态模型往往比具有足够灵活性来描述动态过程的模型更容易实现。

## 5.2 生产力参数

顾名思义，所有的生物种群都是由个体集合而成的，因此会表现出一些新出现的特性，这些特性部分地概括了这些种群的特性。鉴于目前正在发生的许多物种灭绝事件，我同意，一个物种中的少数甚至单个个体可能会构成 “种群（population）”的某种限制，但我们将只关注更丰富的种群，而撇开这些可悲的极端情况。在这里，我们将重点关注个体的生长、成熟和补充，所有这些都与种群的生产力有关。与生长快、成熟早、繁殖力高的物种相比，生长慢、成熟晚、繁殖力低的物种的生产力往往较低，尽管其潜在稳定性更高。生命史特征的进化是一个复杂而又引人入胜的课题，我建议大家对其进行研究 (Beverton 和 Holt 1993; S. C. Stearns 1977; S. Stearns 1992)。不过，在这里我们将重点讨论简单得多的模型。即便如此，在对捕捞种群进行建模时，了解相关物种的生命史特征可能产生的影响，对于解释观察到的动态变化还是很有帮助的。不要忘记，生物过程建模极大地受益于生物知识和理解。

## 5.3 生长

忽略任何潜在的迁入和迁出，种群生产力是种群中新个体的补充和种群中已有个体的生长的综合结果。这些都是积极因素，而自然死亡率以及捕捞种群中的捕捞死亡率等消极因素则抵消了这些积极因素。个体生长的重要性是渔业生态学中有大量关于个体生长的文献的原因之一（Summerfelt 和 Hall，1987）。尽管我刚才写到，如果要建立模型，了解生物过程非常重要，但其中大部分文献都与生长生物学有关，我们将忽略这些文献。相反，我们将专注于个体生长的数学描述。重点在于描述，这一点很重要，因为有些人似乎过于偏爱特定的生长模型，而这些模型仅仅描述了生长过程，并没有真正解释生长过程。

在第 [4](#sec-paraestimat) 章 “[模型参数估算](04-application.qmd)”中，我们已经介绍了可用于描述年龄-长度的三种候选模型 （von Bertalanffy、Gompertz 和 Michaelis-Menton），因此我们不再重新讨论这些模型。相反，我们将考虑描述生长的另外两个方面。在年龄-长度方面，我们将研究与季节性生长模型有关的观点，尽管这些观点在淡水系统中很重要，但往往用处有限。更常用的是，我们还将研究如何从标记数据中估计个体生长情况。

在本章中，我们将主要讨论 von Bertalanffy 生长曲线以及使用函数 vB() 对其进行的拓展：

### 5.3.1 季节性生长曲线

生长速度是一个复杂的过程，不仅受每种动物的体长或年龄的影响，还受环境条件的影响。因此，在温带和极地地区，全年温度变化很大，生长率也会随季节变化。这可以表现为鱼类耳骨（耳石）和其他坚硬部位的确切生长环，它们源于一年中新陈代谢的变化。当然，还有其他因素和事件带来的复杂性和不确定性（例如，硬壳中的假环），但在此我们还是要把重点放在对数据的描述上，假定在分析之前已经解决了这些复杂性（实际情况往往与此不同，不要低估收集有效渔业数据的难度！）。

不同的季节性生长模型已很多，但在此我们将使用 Pitcher 和 Macdonald (1973) 提出的一个模型，对 von Bertalanffy 生长曲线进行了修改，在生长率项中加入了正弦波，试图将周期性水温作为生长率的驱动因素，使生长率在冬季减慢，在夏季加快。

其中 、 和 是 von Bertalanffy 参数， 是长度 的周龄， 决定了围绕非季节性增长曲线的振荡幅度， 是正弦波在年份中的起点，使用常数 52 反映了使用周作为年内时间步长的单位。如果我们使用月或日作为取样事件之间的时间单位，那么我们将分别使用 12 或 365。我们可以使用 Pitcher 和 Macdonald (1973) 对英国淡水鲦鱼（*Phoxinus phoxinus*；虽然鲦鱼的数据是从他们的图中读取的，因此只是近似正确，但足以说明问题）研究的数据。

首先，我们可以绘制现有数据并拟合一条标准的非季节性 von Bertalanffy 生长曲线。拟合曲线使用的是正态随机误差而非对数正态误差，因此我们使用了 negNLL() 函数。使用 ssq() 函数也是可行的，但由于我们对预测曲线的变化特别感兴趣，因此下文将继续使用 negNLL()。如果我们忽略季节性趋势，那么预计曲线周围的变化（由 *sigma* 参数描述）将相对较大。

#vB growth curve fit to Pitcher and Macdonald derived seasonal data   
data(minnow); week <- minnow$week; length <- minnow$length   
pars <- c(75,0.1,-10.0,3.5); label=c("Linf","K","t0","sigma")   
bestvB <- nlm(f=negNLL,p=pars,funk=vB,ages=week,observed=length,   
 typsize=magnitude(pars))   
predL <- vB(bestvB$estimate,0:160)   
outfit(bestvB,backtran = FALSE,title="Non-Seasonal vB",parnames=label)

nlm solution: Non-Seasonal vB   
minimum : 150.6117   
iterations : 41   
code : 3 Either ~local min or steptol too small   
 par gradient  
Linf 89.447640823 5.878821e-05  
K 0.009909338 2.705709e-01  
t0 -16.337065207 -4.717809e-05  
sigma 3.741419172 2.711355e-04

根据季节性数据拟合的标准 von Bertalanffy 曲线 [公式 5.1](#eq-5_1) ，其 *sigma* 参数约为 ，这只是反映了一个事实，即数据围绕平均增长曲线摆动，因此残差会相对较大。如果我们使用季节性调整曲线，这种变化会大大减少，因此我们可以预测 *sigma* 应该小得多。为了拟合季节性增长曲线，我们需要定义一个反映新模型结构的修改后 vB() 函数 [公式 5.2](#eq-5_2)。当然，在拟合任何新模型时，第一步必然是需要一个函数来根据一组参数生成预测值。我们可以将负对数似然的计算包含在这个新函数中，但将预测长度的生成保留在一个单独的函数中可以增加我们代码的灵活性。因此，我们将坚持分别使用预测函数和负对数似然计算函数的策略。在参数向量 *pars* 中，我们还需要包含 [公式 5.2](#eq-5_2) 中 和 的初始估计值。

#plot the non-seasonal fit and its residuals. Figure 5.1   
parset(plots=c(2,1),margin=c(0.35,0.45,0.02,0.05))   
plot1(week,length,type="p",cex=1.0,col=2,xlab="Weeks",pch=16,   
 ylab="Length (mm)",defpar=FALSE)   
lines(0:160,predL,lwd=2,col=1)   
 # calculate and plot the residuals   
resids <- length - vB(bestvB$estimate,week)   
plot1(week,resids,type="l",col="darkgrey",cex=0.9,lwd=2,   
 xlab="Weeks",lty=3,ylab="Normal Residuals",defpar=FALSE)   
points(week,resids,pch=16,cex=1.1,col="red")   
abline(h=0,col=1,lwd=1)

|  |
| --- |
| 图 5.1: 年龄长度数据来自 Pitcher 和 Macdonald（1973 年），与最佳拟合的 von Bertalanffy 曲线相比， 显示出生长率的强烈季节性波动。下图中的正态随机残差的季节性模式非常明显，并随着年龄的增长而减小。 |

# Fit seasonal vB curve, parameters = Linf, K, t0, C, s, sigma   
svb <- function(p,ages,inc=52) {   
 return(p[1]\*(1 - exp(-(p[4] \* sin(2\*pi\*(ages - p[5])/inc) +   
 p[2] \* (ages - p[3])))))   
} # end of svB   
spars <- c(bestvB$estimate[1:3],0.1,5,2.0) # keep sigma at end   
bestsvb <- nlm(f=negNLL,p=spars,funk=svb,ages=week,observed=length,   
 typsize=magnitude(spars))   
predLs <- svb(bestsvb$estimate,0:160)   
outfit(bestsvb,backtran = FALSE,title="Seasonal Growth",   
 parnames=c("Linf","K","t0","C","s","sigma"))

nlm solution: Seasonal Growth   
minimum : 105.2252   
iterations : 21   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient  
Linf 89.06448059 7.259920e-05  
K 0.01040808 3.973383e-01  
t0 -13.46176841 -5.575919e-05  
C 0.10816263 -5.358897e-03  
s 6.96964772 2.208197e-05  
sigma 1.63926525 7.775261e-05

增长率的季节性调整对 和 值的影响较小，但对 值的影响较大，因为季节性比标准 vB() 函数更能允许最初的快速上升。正如预期的那样，对 *sigma* 参数的影响很大(从 3.74 下降到 1.64 )。模型拟合的改进也很大(增加两个参数的代价是 -veLL 从 150 下降到 105)。如 [图 5.2](#fig-5-2) 所示，这也反映在整个残差模式的减少及其最大值和最小值的减半上。

#Plot seasonal growth curve and residuals Figure 5.2   
parset(plots=c(2,1)) # MQMF utility wrapper function   
plot1(week,length,type="p",cex=0.9,col=2,xlab="Weeks",pch=16,   
 ylab="Length (mm)",defpar=FALSE)   
lines(0:160,predLs,lwd=2,col=1)   
 # calculate and plot the residuals   
resids <- length - svb(bestsvb$estimate,week)   
plot1(week,resids,type="l",col="darkgrey",cex=0.9,xlab="Weeks",   
 lty=3,ylab="Normal Residuals",defpar=FALSE)   
points(week,resids,pch=16,cex=1.1,col="red")   
abline(h=0,col=1,lwd=1)

|  |
| --- |
| 图 5.2: Pitcher 和 Macdonald（1973）的近似龄长数据与 von Bertalanffy 生长曲线的拟合季节版本。下图是正态随机残差图，残差仍有一系列高于和低于零的运行，但不如非季节性曲线那么有规律。 |

这当然不是唯一可以描述生长率季节性变化的生长曲线，但本节只是对其原理的介绍。事实上，文献中还有很多可供选择的生长曲线，例如甲壳动物的生长曲线，它们是通过蜕皮来增大体型的。因此，严格来说，描述它们的生长需要估算从初始尺寸开始的生长增量，以及它们蜕皮时生长增量的持续时间和频率。不过，在种群层面，如果数量足够多，在种群动力学模型中，连续的生长曲线往往可以充分近似地描述这种生长。

Pitcher 和 Macdonald (1973) 对他们的第一条季节性生长曲线所预测的负增长的出现表示反对，但由于他们处理的是从鱼群中选取的样本，并对这些样本的平均值进行拟合，因此没有生物学上的理由可以解释为什么这些样本的平均值在一年中的某些时候不能略有下降。他们还指出，使用直接拟合过程“……在带有交互式图形的 PDP 8-E 计算机上联机需要几个小时：然而，有效的优化方法可以大大缩短时间” (Pitcher 和 Macdonald 1973，第603页)。希望这样的陈述能让读者更加了解现在非线性优化器的易得性、卓越的计算机速度及其分析能力（免费提供的软件，如 R，确实是一种能力）。

### 5.3.2 标记数据的 Fabens 方法

到目前为止，在第[4](#sec-paraestimat) 章 “[模型参数估计](04-application.qmd)”一章和本节的 “静态模型”中，我们主要讨论的是年龄-长度数据，这显然需要对样本进行年龄测定和长度测量。遗憾的是，并非所有动物的年龄都能精确到足以使用这种生长描述。在渔业相关科学的早期，一种常用的做法是给鱼类标记、放流（Petersen，1896 ）。回捕鱼类可用于描述其运动特征，后来，通过测量标记时和回捕时的鱼体长度，可用于描述动物的生长情况。在使用标记数据拟合曲线时，需要对用于描述更标准的年龄长度曲线的方程进行转换。因为我们不知道被标记动物的年龄，我们需要一个方程，根据 von Bertalanffy 参数生成预期长度增量，标记时的长度 ，以及经过时间 后的长度将是 。Fabens(1965) 对 von Bertalanffy 曲线进行了改造，使其可以用于从标记程序中获得的信息(参见本章附录中的完整推导)。通过对一般的 von-Bertalanffy 曲线（ [公式 5.1](#eq-5_1)） 的处理，Fabens 得到下面的方程：

其中，某一动物的初始长度是 ， 是 时间段的长度的期望变化。通过使用最小二乘法或负对数似然法，可以估算出 和 的值。为了估算 值，需要可知年龄的平均长度，因此，通常无法进行 估计，并且无法确定生长曲线沿年龄轴的确切位置。在这种情况下， 通常被设为 。

如果能够获得至少记录了标记时的初始长度、标记和回捕之间的时间间隔以及在该时间间隔内发生的生长增量的数据，那么我们在给定初始长度和两个参数 和 的情况下，建立一个函数，生成所需的预测增长增量，然后利用最大似然，我们就可以获得我们所拥有的任何数据的最佳拟合。

这最好通过绘制一些标记的图来说明 [图 5.3](#fig-5-3) （数据在 **MQMF** 数据集 *blackisland*（鲍鱼种群）中找到）。包括一些较大鲍鱼的零增长。数据云是分散的（噪声），但某种形式的趋势是明显的，生长模型将试图拟合这种趋势。

# tagging growth increment data from Black Island, Tasmania   
data(blackisland); bi <- blackisland # just to keep things brief   
parset()   
plot(bi$l1,bi$dl,type="p",pch=16,cex=1.0,col=2,ylim=c(-1,33),   
 ylab="Growth Increment mm",xlab="Initial Length mm",   
 panel.first = grid())   
abline(h=0,col=1)

|  |
| --- |
| 图 5.3: 从塔斯马尼亚州西南角黑岛采集的黑唇鲍鱼标记数据。标记与重捕之间的时间间隔平均为 1.02 年。 |

与第 [4](#sec-paraestimat) 章“[模型参数估计](04-application.qmd)”中对生长的描述一样，这里我们将拟合两条生长曲线并与示例标记数据 进行比较。这两条曲线分别是 von Bertalanffy 曲线（Fabens，1965；见 [公式 5.3](#eq-5_3) ）（鱼类和其它种类的）以及逆 logistic 曲线（Haddon 等，2008 ），更适合难以测定年龄的无脊椎动物所表现出的不确定连续生长。与 von Bertalanffy 曲线相比，逆 logistic 曲线的局限性更大，因为它设计用于以一年为单位的时间增量，或至少是一个恒定的时间增量。

等于 ，其中 和 是增长率分别为 的 50% 和 5% 时的长度， 表示一定长度的平均预期增量的正态随机残差误差。 事后来看， 可能定义为 更好。 请注意，由于进行了 Fabens 变换，Fabens 模型的残差误差与年龄长度模型的残差误差有所不同，尽管二者都使用了正态概率密度函数（稍后详述）。

**MQMF** 中定义了两个函数来生成 von Bertalanffy 曲线的预测生长增量：fabens() 和逆 logistic invl()。这两个函数的实现非常简单，反映了 [公式 5.3](#eq-5_3) 和 [公式 5.4](#eq-5_4) 。读者应检查这些函数的代码（即不带括号的 fabens() 或 invl()），并阅读每个函数的帮助。读者很快就会明白，这些函数的定义方式可以方便地使用列名不是 和 的 data.frames。在下文中，我们明确使用了这些名称，尽管我们本可以直接使用默认值。

### 5.3.3 拟合标记数据模型

在模型拟合方面中，我们已经有了生成预测生长增量（ ）所需的函数（fabens() 和 invl() ）以及优化器 nlm()，仍然需要的是在搜索最小值期间计算负对数似然的函数。在这一点上，没有理由使用正态随机误差以外的其他误差。我们将使用 **MQMF** 函数 negNLL() 来拟合两条曲线，然后通过可视化和似然比检验对它们进行比较。如果需要更改用于所需数据的列名，可以使用 ….。

# Fit the vB and Inverse Logistic to the tagging data   
linm <- lm(bi$dl ~ bi$l1) # simple linear regression   
param <- c(170.0,0.3,4.0); label <- c("Linf","K","sigma")   
modelvb <- nlm(f=negNLL,p=param,funk=fabens,observed=bi$dl,indat=bi,   
 initL="l1",delT="dt") # could have used the defaults   
outfit(modelvb,backtran = FALSE,title="vB",parnames=label)

nlm solution: vB   
minimum : 291.1691   
iterations : 24   
code : 1 gradient close to 0, probably solution   
 par gradient  
Linf 173.9677972 9.563666e-07  
K 0.2653003 2.656839e-04  
sigma 3.5861240 1.391768e-05

predvB <- fabens(modelvb$estimate,bi)   
cat("\n")

param2 <- c(25.0,130.0,35.0,3.0)   
label2=c("MaxDL","L50","delta","sigma")   
modelil <- nlm(f=negNLL,p=param2,funk=invl,observed=bi$dl,indat=bi,   
 initL="l1",delT="dt")   
outfit(modelil,backtran = FALSE,title="IL",parnames=label2)

nlm solution: IL   
minimum : 277.0122   
iterations : 26   
code : 1 gradient close to 0, probably solution   
 par gradient  
MaxDL 21.05654 -2.021972e-06  
L50 130.92643 5.900276e-07  
delta 40.98771 2.218945e-07  
sigma 3.14555 4.535835e-06

predil <- invl(modelil$estimate,bi)

逆 logistic 模型的负对数似然小于 von Bertalanffy 模型的负对数似然，而且逆 logistic 呈现出较小的 值。如果我们将这两条增长曲线与原始数据进行对比，就可以更清楚地看到它们之间的差异[图 5.4](#fig-5-4)。此外，对各自的残差进行检验也会发现曲线之间的差异，预测的 von Bertalanffy 曲线的残差呈圆弧状，这与其预测的初始长度与生长增量之间的线性关系是一致的。在 [图 5.4](#fig-5-4) 中，线性回归绘制在 von Bertalanffy 曲线的上方，以说明它实际上是重合的。

#growth curves and regression fitted to tagging data Fig 5.4   
parset(margin=c(0.4,0.4,0.05,0.05))   
plot(bi$l1,bi$dl,type="p",pch=16,cex=1.0,col=3,ylim=c(-2,31),   
 ylab="Growth Increment mm",xlab="Length mm",panel.first=grid())   
abline(h=0,col=1)   
lines(bi$l1,predvB,pch=16,col=1,lwd=3,lty=1) # vB   
lines(bi$l1,predil,pch=16,col=2,lwd=3,lty=2) # IL   
abline(linm,lwd=3,col=7,lty=2) # add dashed linear regression   
legend("topright",c("vB","LinReg","IL"),lwd=3,bty="n",cex=1.2,   
 col=c(1,7,2),lty=c(1,2,2))

|  |
| --- |
| 图 5.4: 黑岛黑唇鲍标记数据的von Betalanffy（黑色）、逆 logistic （红色曲线虚线）和线性回归（黄色虚线）拟合。 标记与回捕之间的时间间隔为 1.02 年。显然，vB 和线性回归是相同的。 |

两条生长曲线的残差也显示了各自模型拟合的差异。

#residuals for vB and inverse logistic for tagging data Fig 5.5   
parset(plots=c(1,2),outmargin=c(1,1,0,0),margin=c(.25,.25,.05,.05))   
plot(bi$l1,(bi$dl - predvB),type="p",pch=16,col=1,ylab="",   
 xlab="",panel.first=grid(),ylim=c(-8,11))   
abline(h=0,col=1)   
mtext("vB",side=1,outer=FALSE,line=-1.1,cex=1.2,font=7)   
plot(bi$l1,(bi$dl - predil),type="p",pch=16,col=1,ylab="",   
 xlab="",panel.first=grid(),ylim=c(-8,11))   
abline(h=0,col=1)   
mtext("IL",side=3,outer=FALSE,line=-1.2,cex=1.2,font=7)   
mtext("Length mm",side=1,line=-0.1,cex=1.0,font=7,outer=TRUE)   
mtext("Residual",side=2,line=-0.1,cex=1.0,font=7,outer=TRUE)

|  |
| --- |
| 图 5.5: 黑岛黑唇鲍标记数据的 von Betalanffy 曲线残差图（左侧）和 逆 Logistic 曲线残差图（右侧）。标记与回捕之间的时间间隔为 1.02 年。 |

### 5.3.4 对 Fabens 方法的深入探讨

von Bertalanffy 方程的 Fabens 变换所描述的生长曲线使用了相同的参数，但却暗藏了一些不易察觉的差异。您一定还记得我们使用两种不同的残差误差结构拟合 vB 曲线的例子，这使得结果无法比较，或者更严格地说，无法比较。Fabens 变换改变了残差结构以及参数之间的相互作用方式。取而代之的是 变成了绝对最大值，这就是它能预测负增长的原因。严格来说，Fabens 模型的参数与 von Bertalanffy 模型的参数含义不同。

我们对 von Bertalanffy 曲线和逆 logistic 曲线进行了比较，但由于我们使用了鲍鱼种群的数据作为例子，我们对哪条曲线可能更有用的看法有失偏颇。在大多数有鳞鱼类渔业中，人们发现 von Bertalanffy 曲线作为对生长的一般描述是最有用的，尽管它还存在一些问题。但所有生长模型（不只是标记生长模型）的一个普遍问题是我们迄今为止所使用的假设，即围绕预测生长趋势的观测变化将是恒定的。如果我们看一下 [图 5.3](#fig-5-3) 或 [图 5.4](#fig-5-4) ，还记得我们在第 [4](#sec-paraestimat) 章“[模型参数估计](04-application.qmd)”中绘制的年龄长度数据，那么在每种情况下，初始长度或年龄的变化趋势都比较明显。与其坚持在残差中使用恒定的方差（如 SSQ 所要求的），不如使用恒定的变异系数（ ）。如果我们使用最大似然法，就有可能对残差的方差进行单独建模，从而得出一系列不同的结果；Francis（1988 ）正是这样做的。他使用最大似然法对数据进行了标记模型拟合，并假定每种情况下的残差都是正态分布的，但他对残差方差与期望 之间的关系提出了一些不同的函数形式（与初始长度的函数关系也是可能的）。迄今为止，我们使用的方法假定方差为常数，因此我们估算了一个常数 参数。Francis（1988）则建议，方差可以有一个介于 和 之间的恒定乘数，标记数据之间可以存在反比关系。这些考虑因素也适用于年龄-长度模型，在这种模型中，常数乘数将使年龄与 之间呈线性关系。

其中 是期望 的常数乘数，需要分别估算。此时正态似然将变为：

其中用 替代一般正态似然的 。除这一简单的替代方案外，Francis（1988 ）还建议对标记数据采用指数递减的残差标准差，再加上一个可估算的常数 ：

最后，Francis（1988）建议采用幂律描述的残差标准差：

Francis（1988）还通过考虑偏差、生长率的季节性变化以及离群污染的影响，将 Fabens 方法扩展到分析标记数据。文献中往往只强调了报告的一个方面，这也是经常查阅原始文献而不是依赖其他论文或书籍（如这本书）中的摘要的一个很好的理由。如果需要对标记数据拟合生长曲线，Francis（1988）是一个很好的起点。

由于方差与预期 之间关系的表达不同，常数 和 的解释也发生变化。 如前所述，如果假定的误差结构不同，从同一模型中得到的参数估计值也会不同，从而变得不可比较。遗憾的是，如何选择最合适的误差结构并不是一个简单的问题。充其量，无论选择哪种曲线，使用非恒定方差都能提供与之相关的不确定性的另一种视角。

### 5.3.5 非恒定方差的实现

通过包含期望 和 残差方差之间关系，我们需要改变包装函数，围绕如何使用预测值 计算负对数似然。之前，对于常数 我们用 negNULL() ，所以我们可从它开始修改。我们所做的只是包括一个额外函数 funksig() ，作为 negnormL() 的参数（参见其帮助和代码），从而包括 sigma 值的计算。通过这种方法，我们可以在 funksig() 中保留常数 的长版本。但这也表明函数的行为符合预期。然后，我们可以加入一个替代的 funksig()，实现上述三个选项之一（或我们自己设计的选项）。

# fit the Fabens tag growth curve with and without the option to   
 # modify variation with predicted length. See the MQMF function   
 # negnormL. So first no variation and then linear variation.   
sigfunk <- function(pars,predobs) return(tail(pars,1)) #no effect   
data(blackisland)   
bi <- blackisland # just to keep things brief   
param <- c(170.0,0.3,4.0); label=c("Linf","K","sigma")   
modelvb <- nlm(f=negnormL,p=param,funk=fabens,funksig=sigfunk,   
 indat=bi,initL="l1",delT="dt")   
outfit(modelvb,backtran = FALSE,title="vB constant sigma",   
 parnames = label)

nlm solution: vB constant sigma   
minimum : 291.1691   
iterations : 24   
code : 1 gradient close to 0, probably solution   
 par gradient  
Linf 173.9677972 9.563666e-07  
K 0.2653003 2.656839e-04  
sigma 3.5861240 1.391768e-05

sigfunk2 <- function(pars,predo) { # linear with predicted length   
 sig <- tail(pars,1) \* predo # sigma x predDL, see negnormL   
 pick <- which(sig <= 0) # ensure no negative sigmas from   
 sig[pick] <- 0.01 # possible negative predicted lengths   
 return(sig)   
} # end of sigfunk2   
param <- c(170.0,0.3,1.0); label=c("Linf","K","sigma")   
modelvb2 <- nlm(f=negnormL,p=param,funk=fabens,funksig=sigfunk2,   
 indat=bi,initL="l1",delT="dt",   
 typsize=magnitude(param),iterlim=200)   
outfit(modelvb2,backtran = FALSE,parnames = label,   
 title="vB inverse DeltaL, sigma < 1")

nlm solution: vB inverse DeltaL, sigma < 1   
minimum : 288.1933   
iterations : 64   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient  
Linf 171.0000001 45579.23697  
K 0.2724177 77.18211  
sigma 0.4139252 19.64508

请记住，通过改变残差结构，似然值变得不相称，因此我们无法确定哪种拟合效果更好。恒定方差使得 von Bertalanffy 曲线在长度大于 148 左右时有效地保持在数据之上，而变化方差生长曲线大部分时间仍在数据之上，但由于方差随预测长度的减少而向后推近数据。我们可以通过重写 negnormL() 来进行实验，使用初始长度而不是预测长度。如前所述，将平均预测长度与相关方差分离可以获得极大的灵活性。

#plot to two Faben's lines with constant and varying sigma Fig 5.6   
predvB <- fabens(modelvb$estimate,bi)   
predvB2 <- fabens(modelvb2$estimate,bi)   
parset(margin=c(0.4,0.4,0.05,0.05))   
plot(bi$l1,bi$dl,type="p",pch=1,cex=1.0,col=1,ylim=c(-2,31),   
 ylab="Growth Increment mm",xlab="Length mm",panel.first=grid())   
abline(h=0,col=1)   
lines(bi$l1,predvB,col=1,lwd=2) # vB   
lines(bi$l1,predvB2,col=2,lwd=2,lty=2) # IL   
legend("topright",c("Constant sigma","Changing sigma"),lwd=3,   
 col=c(1,2),bty="n",cex=1.1,lty=c(1,2))

|  |
| --- |
| 图 5.6: 与黑岛黑唇鲍标记数据拟合的具有恒定 sigma von Bertalanffy（皇家蓝）和 具有与不断变化的 DeltaL 相关的 sgima 的 von Bertalanffy（红色虚线）。 标记与再捕之间的平均时间间隔为 1.02 年。 |

我们的眼睛习惯于欣赏对称性，因此 [图 5.6](#fig-5-6) 中不断变化的变化线看起来拟合得相对较差。但是，假设误差只存在于 y 轴上（y-on-x 问题），并且残差的方差也在不断变化，作为拟合模型的反映，下部的红色虚线使数据的值内有一条内在的曲线。使用更复杂的残差结构使模型拟合对初始条件更敏感。您会注意到，在输出结果中，假设残差方差恒定只需要 20-30 次迭代，而使用方差函数关系则需要两倍的迭代次数（这是在使用 Typsize 可选参数来增强稳定性的情况下，而在更简单的模型中没有使用）。如果修改这些起始值，很容易得到难以置信的答案。

尽管在使用年龄长度数据时，特别是在非常小的年龄段，变化的差异往往很明显，但在建模中引入这种复杂性还是需要有充分的理由。

## 5.4 目标模型选择

在第 [4](#sec-paraestimat) 章“[模型参数估计](04-application.qmd)（*Model Parameter Estimation*） ”中，我们看到了一个类似的章节，其中介绍了基于平方和的 Akaike 信息准则（AIC；Akaike，1974）的信息理论近似值（Burnham 和 Anderson，2002）。现在我们使用最大似然法，我们将介绍 AIC 的原始版本，并讨论似然比。

### 5.4.1 Akaike 信息准则

在试图描述数据集中的模式时，选择使用哪种模型时，使用的参数数量与模型拟合数据集的质量之间的权衡始终是一个问题。经典的方法是选择最简单的模型，该模型能很好地描述该模式（奥卡姆剃刀只是一个关于简约性的劝告，因此如果两个模型产生的结果相同，就选择最简单的一个）。但是，“等效结果”到底有多接近，“最简单”又是什么意思。Akaike 的解决方案（1974）是将似然比和 “模型内独立调整参数的数量”结合起来使用。AIC 是一种基于似然值的惩罚性模型选择标准，最小的 AIC 表示最佳模型（在似然值与参数数量之间取得平衡）。

其中， 是总对数似然值， 是拟合模型时显式调整的参数数量，而 是一个乘数（惩罚项，在惩罚似然中），其值由 Akaike（1974）设定为 2.0，这是基于信息论的一个论点。强调“独立调整的参数”是“显式调整的”是为了避免混淆那些保持不变或从其他参数和模型变量推导出来的模型参数（我们在过剩产量模型中的捕获率参数中会看到一个例子）。例如，在渔业模型中一个常见的假设是自然死亡率，通常表示为 ，是一个常数。因此，在 AIC 的  值中不会将其计算在内。将  作为惩罚项的显式标识，而不是直接使用常数值 2.0，是为了强调关于应使用的确切值存在争议（Bhansali 和 Downham, 1977；Schwartz, 1978；Akaike, 1979；Atkinson, 1980），以确保模型复杂度（参数数量）与拟合质量（最大似然或最小平方和）之间的平衡能够反映模拟中的实际情况。 尽管统计论据相对较为密集，最终归结为  的值  应该多大？这一结果被称为施瓦茨的贝叶斯信息准则，简称 BIC。

其中 是样本大小或模型拟合的数据点数， 是参数数量。只要数据点数大于或等于 8（ ），BIC 将比 AIC 更严厉地惩罚复杂模型。另一方面，从直观上讲，考虑样本大小是有道理的。如果我们使用 MQMF 函数 aicbic() 对 von Bertalanffy 模型和逆逻辑斯蒂模型的拟合结果进行比较，其中使用了常数 估计 [图 5.4](#fig-5-4)，逆逻辑斯蒂模型的 AIC 和 BIC 均较小，因此在这种情况下，逆逻辑斯蒂模型将是更优的选择。

#compare the relative model fits of Vb and IL   
cat("von Bertalanffy \n")

von Bertalanffy

aicbic(modelvb,bi)

aic bic negLL p   
588.3382 596.3846 291.1691 3.0000

cat("inverse-logistic \n")

inverse-logistic

aicbic(modelil,bi)

aic bic negLL p   
562.0244 572.7529 277.0122 4.0000

### 5.4.2 似然比检验

可以使用广义似然比检验（Neter et al, 1996）来比较不同模型拟合效果与复杂度之间的差异。该方法依赖于似然比检验在样本量增大时渐近趋近于 分布的事实。这意味着在通常的渔业数据量下，这种方法只是近似的方法。似然比检验正如其名所示，是两个似然性的比值的对数，或者直接处理对数似然性时，是两个对数似然性的相减，两者是等价的（当然，这两个似然性必须来自同一个概率密度函数）。我们希望确定两个使用相同数据和残差结构但模型结构不同的模型（即不同的参数），哪一个能显著更好地拟合可用数据。由于似然比可以由 分布描述，因此可以正式回答这个问题。 因此，对于两个模型的似然性，如果它们要么具有相同数量的参数，要么仅相差一个参数，在考虑显著性差异时，它们的比值需要大于所需的自由度（不同参数的数量）对应的 分布。

其中 是模型 的 参数的似然， 是等效的对数似然。 是 分布的 百分位数（例如，对于 95% 的置信区间， 和 ，即 。

简而言之，如果我们要比较两个仅相差一个或没有参数的模型，那么如果它们的负对数似然值相差超过 1.92（3.84/2），则可以认为这两个模型在统计上显著不同，其中一个模型将比另一个模型提供显著更好的拟合。如果两个模型之间相差两个参数，那么这两个模型在统计上显著不同的负对数似然值的最小差异需要达到 自由度为 2），依此类推，对于参数差异更多的情况（Venzon 和 Moolgavkar, 1988）。

关于 von Bertalanffy 曲线和逆逻辑斯蒂曲线的比较，逆逻辑斯蒂负对数似然值为 ，明显小于 von Bertalanffy 曲线的负对数似然值。我们可以使用 MQMF 函数 likeratio()来说明这两种曲线在拟合数据方面的差异具有高度显著性。

# Likelihood ratio comparison of two growth models see Fig 5.4   
vb <- modelvb$minimum # their respective -ve log-likelihoods   
il <- modelil$minimum   
dof <- 1   
round(likeratio(vb,il,dof),8)

LR P mindif df   
28.31380340 0.00000005 3.84145882 1.00000000

### 5.4.3 似然比检验的注意事项

由于在渔业模型中广泛使用加权对数似然，当使用似然比检验来比较替代模型时，需要小心确保这些模型实际上是可比的。当我们使用相同的数据和相同的概率密度函数来描述残差分布时，可以使用似然比检验。但请记住，当我们仅对相同的生长模型应用不同的残差误差结构假设（对数正态而非正态）时，这些曲线是完全不可比的（不具可比性）。同样，在使用惩罚似然时，不同版本的模型可能会对不同的数据流给予不同的权重，这种变化使得模型不可比。很明显，应该只比较可比的模型，但在一些更复杂的模型中，确定什么是公平比较有时并不总是简单的。

## 5.5 关于生长的说明

对个体生长过程的描述对渔业模型很重要，因为个体的大小和重量的生长是特定种群生产力的重要组成部分。这种生长显然是增加种群生物量过程与减少种群生物量过程之间平衡的一个积极因素。但重要的是要理解，生长的描述总是描述性的，并且主要适用于我们有数据的大小范围。有许多实例表明，von Bertalanffy 曲线外推的弱点会导致生物学上荒谬的预测（Knight，1968）。但这类实例是将模型参数的特定解释误认为现实，而曲线仅仅是生长数据的描述，严格来说只在有数据的地方有效。这就是为什么已经产生了生长模型转换，这些转换估计的参数可以位于可用数据的范围内（Schnute，1981；Francis，1988，1995）。

## 5.6 成熟

### 5.6.1 引言

目前，渔业管理倾向于使用所谓的生物参考点来构建其希望构成合理且可辩护的管理决策的基础，这些决策针对的是可再生的自然资源（FAO，1995，1996，1997；Restrepo 和 POwers，1999；Haddon，2007）。简单来说，这个想法是设定一个目标参考点，通常定义为成熟生物量水平或其某种替代指标，该指标被认为是种群的一种理想状态。它是理想状态，因为通常它应该能够为后续的补充提供良好的水平，并为种群提供足够的恢复力以抵御偶尔的环境冲击。常见的默认值可以是 ，即 ，或 40% 的消耗水平，这被用作 的替代指标，或者是能够可持续产生最大可持续产量的产卵生物量。在澳大利亚，联邦收获策略政策将目标定义为 ，这被用作 的替代指标，即应该能够产生最大经济效益的产卵生物量（注意，目标是生物量水平 ，而不是 ；DAFF，2007；Rayns，2007）。 此外，还需定义一个极限参考点，同样以成熟生物量（或替代指标）的水平来表示，低于该水平时，所关注的种群将被认为其补充量存在受损害的风险。生物参考点通常以未受干扰的产卵生物量的估计值（ ）为基准来表述。 的概念可以是均衡概念，也可以是更动态的版本，试图解释补充量的变化。无论哪种情况，这个数值在评估中往往具有不确定性，并且在不同评估中经常发生变化（Punt 等，2018 ）。为了提供管理建议，需要三个要素：

* 对当前种群状态进行评估，包括繁殖生物量的消耗情况，从而
* 当前未捕捞的产卵生物量估计值，最后
* 一个决定下一个季节（些）允许的捕捞死亡率、努力量或渔获量的捕捞控制规则。

也许你注意到了，我在成熟和产卵生物量这两个术语之间互换使用。这样做并不是想让你感到困惑，而是想让你习惯于文献中会遇到的术语变化。

强调成熟生物量或产卵生物量的重要性，是获得合理成熟大小或年龄估计的原因。这是为了确保对成熟生物量有多少的估计能够反映鱼种的生物学特性以及当前渔业状态。在本书中，我们试图专注于建模，但始终试图模拟潜在的生物学现实。事实上，生物学现实，就像生长的细节一样，往往相当复杂。在渔业理论中，许多基础来自北半球，特别是对北海和大西洋的高生产力鱼种进行研究（Smith，1994）。那里的许多商业物种具有相对简单的繁殖历史。有雄性和雌性，作为一个种群，它们按一般节律成熟，它们在死亡之前要么产卵一次，要么每年产卵。当然，这是一种过度简化，生物学世界比这要多样化得多，即使在北半球也是如此。 有些物种从一开始就是雌性，其中一些较大的雌性会转变为雄性（原雌雄同体），反之亦然（原雄雌同体）。存在许多不同的繁殖策略，其中许多策略会影响诸如成熟大小和/或年龄等因素。然而，在这里我们将只关注经典方法。尽管如此，使用你自己的例子时，不要对生物学做出假设，始终情况下，来自种群的代表性证据比假设更好。

成熟时的个体大小或年龄是一个种群特性。个体可能经历成熟过程，并需要时间才能具备繁殖能力，但我们描述的过程是针对整个种群的平均值。给定一个样本，希望是一个足够大的样本，能够覆盖成熟时个体大小或年龄的范围，我们的目标是描述每个个体大小或年龄的鱼类中成熟的比例。成熟时个体大小（size-at-maturity ）的概念通常被总结为 种群中50% 个体性成熟（具备繁殖能力）时的个体大小。实际上， ，即 50% 成熟的个体长度，只是重要因素的一部分，此外，正如我们将看到的，关于物种随长度或年龄成熟的速度的某种度量也同样重要。一些更简单的渔业模型仍然使用一种被称为“刀刃式成熟”（knife-edge matruity）的假设，这意味着一个强烈的假设，即 100% 的动物在某个特定年龄成熟。我们将在后面讨论种群成熟所需的时间如何影响种群动态。 如何确定成熟度在此不予考虑，尽管理想情况下应使用组织学来确认成熟的配子，而不是仅仅通过观察鱼类的性腺。再次强调，有大量文献致力于详细说明确定各种物种成熟阶段的细节，但我们不会尝试去探讨这些内容。我们所要完成的任务是找到预期种群成熟曲线的数学描述。预期的是，被认为成熟的动物比例会随着体型和年龄的增长而增加，直到 100%成熟（尽管即使对此也有例外，某些物种的雌性会花几年时间从产生卵子所需的大量能量投资中恢复，从而暂时停止繁殖）。生命史特征体现在物种是适应一次性繁殖（半繁殖性）还是多次繁殖（多次繁殖性），或者它产生的是大量小卵还是远少得多的大卵，甚至生产少量活体幼崽。始终要记住，生物学可以非常复杂，而生物学的数学描述是相对简单的抽象。

### 5.6.2 替代的成熟度曲线

一个种群内成熟的一般模式是，成熟度会分布在一系列长度和年龄中，其中一部分早熟，一部分晚熟，大多数则在某个平均时间成熟。如果我们想象一个随时间变化的成熟比例的抛物线曲线，可能接近正态分布但尾部更短，那么我们想要的是该分布的累积密度函数，在某个长度或年龄范围内从 0% 到 100% 运行。结果是常见的 S 形曲线，通常称为逻辑斯蒂曲线，在文献中，有大量不同的数学表达式来描述这类曲线。我们关注的统计量是 （50%成熟的大小/年龄），以及 ，即四分位距，它衡量成熟度曲线（在 25%和 75%成熟时的年龄）之间的大小/年龄范围。使用四分位距是一个任意选择，但反映了常见做法（如箱线图所示），但并不排除使用更宽或更窄的范围，如果这些范围更方便的话。

在众多不同版本中，有一个经典的逻辑斯蒂曲线 [公式 5.12](#eq-5_12)，常用于描述许多种群的成熟过程（参见 **MQMF** 函数）。这种形式非常适合使用具有二项式误差的广义线性模型进行拟合。我们一直使用 R 作为编程环境，但这里是一个提醒我们，它的最初目的是作为统计分析工具的机会。

是长度 的成熟比例，而 和 是指数参数。请注意，我在方程中未包含误差项。这里我们试图对不同个体大小（为简洁起见，我将停止每次写“尺寸或年龄”）的成熟或不成熟观测进行预测。这正是建议我们应使用二元误差的原因，如模型参数估计章节所述。这种公式的另一个优点是 ，即 50% 成熟时的个体大小，可以直接从参数中推导出来。类似地，四分位距也可以从参数中推导出来，表示为 。

成熟数据具有二元特性，在特定采样时间对种群进行采样时，观测值是每个大小为 的采样鱼是否成熟。本书的 R 包 **MQMF** 包含一个示例数据集 tasab，其中包含来自塔斯马尼亚海岸线 16 公里段上两个地点的黑唇鲍鱼（*Haliotis rubra*）数据。

# The Maturity data from tasab data-set   
data(tasab) # see ?tasab for a list of the codes used   
properties(tasab) # summarize properties of columns in tasab

Index isNA Unique Class Min Max Example  
site 1 0 2 integer 1 2 1  
sex 2 0 3 character 0 0 I  
length 3 0 85 integer 62 160 102  
mature 4 0 2 integer 0 1 0

table(tasab$site,tasab$sex) # sites 1 & 2 vs F, I, and M

F I M  
 1 116 11 123  
 2 207 85 173

鲍鱼因其生物学特征在相对较小的空间尺度上具有高度变异性而臭名昭著，因此采样地点的细节非常重要（Haddon 和 Helidoniotis，2013）。这些数据都是由同一群人在同一个月同年收集的，因此除了壳长之外，我们唯一可能认为会影响成熟度的因素就是具体地点（性别似乎在同一时间、同一个体大小成熟； [图 5.7](#fig-5-7) ）。

#plot the proportion mature vs shell length Fig 5.7   
propm <- tapply(tasab$mature,tasab$length,mean) #mean maturity at L   
lens <- as.numeric(names(propm)) # lengths in the data   
plot1(lens,propm,type="p",cex=0.9,xlab="Length mm",   
 ylab="Proportion Mature")

|  |
| --- |
| 图 5.7: tasab 数据集中黑唇鲍鱼成熟数据的长度成熟比例。 |

数据看起来相对嘈杂，虽然这很难判断，因为一些观察到的长度，其成熟比例不是 0 也不是 1，可能只有少数几次观测。例如，在长度为 100mm 处有一个成熟比例为 0.5 的点，如果你深入数据会发现该点仅由两个观测组成，一个成熟另一个不成熟。可以通过 pick <- which(tasab$length == 100) 查找这些长度，或者通过 which(propm == 0.5) 查看数据，再使用 tasab[pick,]进一步分析。

使用 properties(tasab) 或 head(tasab,10) 查看数据集，可以告诉我们有哪些变量。虽然我们显然希望检查成熟和长度之间的关系，但请记住，最初我们还想检查站点对成熟的影响。重要的是，我们需要将站点变量转换为分类因子，否则它将被视为包含 1 和 2 的整数向量，而不是潜在的不同处理。我们不能将性别作为因子包括进来，因为所有动物最初都是未成熟状态，尽管可以始终分别分析雄性和雌性数据。然而，在更多的站点上，到目前为止，在黑唇牡蛎中，尚未发现性别在成熟速率或平均大小上的可重复差异。一旦拟合，我们会发现站点对分析无信息性（它不显著），因此我们重新进行不包括站点的分析。

#Use glm to estimate mature logistic   
binglm <- function(x,digits=6) { #function to simplify printing   
 out <- summary(x)   
 print(out$call)   
 print(round(out$coefficients,digits))   
 cat("\nNull Deviance ",out$null.deviance,"df",out$df.null,"\n")   
 cat("Resid.Deviance ",out$deviance,"df",out$df.residual,"\n")   
 cat("AIC = ",out$aic,"\n\n")   
 return(invisible(out)) # retain the full summary   
} #end of binglm   
tasab$site <- as.factor(tasab$site) # site as a factor   
smodel <- glm(mature ~ site + length,family=binomial,data=tasab)   
outs <- binglm(smodel) #outs contains the whole summary object

glm(formula = mature ~ site + length, family = binomial, data = tasab)  
 Estimate Std. Error z value Pr(>|z|)  
(Intercept) -19.797096 2.361561 -8.383056 0.000000  
site2 -0.369502 0.449678 -0.821703 0.411246  
length 0.182551 0.019872 9.186463 0.000000  
  
Null Deviance 564.0149 df 714   
Resid.Deviance 170.7051 df 712   
AIC = 176.7051

model <- glm(mature ~ length, family=binomial, data=tasab)   
outm <- binglm(model)

glm(formula = mature ~ length, family = binomial, data = tasab)  
 Estimate Std. Error z value Pr(>|z|)  
(Intercept) -20.464131 2.265539 -9.032787 0  
length 0.186137 0.019736 9.431291 0  
  
Null Deviance 564.0149 df 714   
Resid.Deviance 171.3903 df 713   
AIC = 175.3903

cof <- outm$coefficients   
cat("Lm50 = ",-cof[1,1]/cof[2,1],"\n")

Lm50 = 109.9414

cat("IQ = ",2\*log(3)/cof[2,1],"\n")

IQ = 11.80436

在第一次分析中，包括了站点因素，因为我在汇总数据时确保所包含的站点相似，因此站点 2（P=0.411）与站点 1 没有显著差异，这意味着在这种情况下，站点因素对分析来说并不具有信息性。因此，我们再次进行了分析，将站点从方程中剔除。如果站点之间表现出显著差异，那么我们需要使用多个曲线来描述模型结果。无论怎样，我们还是会将它们画成不同的曲线来说明过程。额外的站点参数是对初始指数截距值的修饰。因此，ab$length 参数在两种情况下都是 参数，但站点 1 的曲线可以通过将截距视为 参数来获得，而站点 2 的曲线则需要将截距和 ab$site2 的参数值相加，作为 参数。因此，站点 1 模型为 和 ，而站点 2 模型为 和 。将最终无站点模型添加到图中表明，综合模型更接近站点 2，这反映了站点 2 的观测次数为 465 次，而站点 1 仅为 250 次。 注意，参数少一个后，残差偏差略有增加，但尽管如此，包含站点因子的模型的 AIC 值仍然大于简化模型的 AIC 值，这再次表明简化模型在复杂性和模型拟合度之间提供了更好的平衡。这与广义线性模型及其比较和操作有明显的类比。如果确实发现截距之间存在显著差异，那么测试差异是否也扩展到 参数，以确定是否需要完全分开处理这些曲线是有意义的。共享公共参数通常是有帮助的，因为它可以增加样本量。

#Add maturity logistics to the maturity data plot Fig 5.8   
propm <- tapply(tasab$mature,tasab$length,mean) #prop mature   
lens <- as.numeric(names(propm)) # lengths in the data   
pick <- which((lens > 79) & (lens < 146))   
parset()  
plot(lens[pick],propm[pick],type="p",cex=0.9, #the data points   
 xlab="Length mm",ylab="Proportion Mature",pch=1)   
L <- seq(80,145,1) # for increased curve separation   
pars <- coef(smodel)   
lines(L,mature(pars[1],pars[3],L),lwd=3,col=3,lty=2)   
lines(L,mature(pars[1]+pars[2],pars[3],L),lwd=3,col=2,lty=4)   
lines(L,mature(coef(model)[1],coef(model)[2],L),lwd=2,col=1,lty=1)   
abline(h=c(0.25,0.5,0.75),lty=3,col="grey")  
legend("topleft",c("site1","both","site2"),col=c(3,1,2),lty=c(2,1,4),   
 lwd=3,bty="n")

|  |
| --- |
| 图 5.8: 黑唇牡蛎成熟度数据（tasab 数据集）的长度成熟比例。不考虑站点的综合分析（两者）更接近站点 2（点划线）而非站点 1（划线），这反映了站点 2 样本量更大。 |

大样本量通常能提高模型拟合的质量。如果有足够的数据，当前数据所表现出的变异性可能会被充分降低，从而可以在这些站点之间找到统计学上的显著差异。然而，如果我们考虑站点 1 和站点 2 的 差异约为 2 毫米（总共 110 毫米），从生物学角度来看，这种差异可能并不重要，因为两个站点之间的生长速率也可能不同。正如任何模型或统计分析一样，在关注统计显著性的同时，还应考虑生物学意义。

### 5.6.3 对称性假设

在很多情况下，标准逻辑曲线很好地描述了从未成熟到成熟（或者可能是其他生物过渡，如隐性到显性、 molt 阶段 A 到阶段 B 等）的过程。然而，经典逻辑曲线的主要限制是它在 点对称，这可能不是对现实世界事件的最佳描述。

已经提出了多种替代的非对称曲线，但幸运的是，Schnute 和 Richards（1990）提出了一种通用或统一模型，用于描述“鱼类生长、成熟和存活数据”，这种模型不仅推广了用于描述成熟度的经典逻辑斯谛模型，还推广了 Gompertz（1825）、von Bertalanffy（1938）、Richards（1959）、Chapman（1961）和 Schnute（1981）提出的生长模型，其中一些模型也可以用于描述成熟过程。

Schnute 和 Richards (1990) 模型有四个参数：

可以重新排列， [公式 5.14](#eq-5_14) ，以更好地展示其与经典的逻辑斯蒂曲线 [公式 5.12](#eq-5_12) 的关系，如果我们将两者设置为 ，则相当于经典的逻辑斯蒂曲线（例如，设置 和 ）。

如果使用其中一种特殊的经典情况，可能会有解析解来确定 和 ，但通常情况下需要通过数值方法来找到它们（参见 **MQMF** 函数bracket()和 linter() ）。这条曲线可以用二项式误差来拟合成熟数据，就像之前那样，尽管可以使用 nlm() 或其他非线性求解器（Schnute 和 Richards, 1990 年提供了所需的似然函数）。但特殊情况可能为完整的四参数模型提供更稳定的解。从（5.14）式所示的可能曲线的不对称性，可以很容易地使用 **MQMF** srug() 函数（Schnute 和 Richards 统一增长模型）来证明。在没有找到 和 的解析解时，我们可以使用两个函数 bracket() 和 linter()，它们界定了目标值（在这种情况下为 0.25、0.5 和 0.75），然后线性插值以生成所需的统计量的近似估计值（参见它们的帮助文件和示例）。

#Asymmetrical maturity curve from Schnute and Richard's curve Fig5.9   
L = seq(50,160,1)   
p=c(a=0.07,b=0.2,c=1.0,alpha=100)   
asym <- srug(p=p,sizeage=L)   
L25 <- linter(bracket(0.25,asym,L))   
L50 <- linter(bracket(0.5,asym,L))   
L75 <- linter(bracket(0.75,asym,L))   
parset()  
plot(L,asym,type="l",lwd=2,xlab="Length mm",ylab="Proportion Mature")   
abline(h=c(0.25,0.5,0.75),lty=3,col="grey")   
abline(v=c(L25,L50,L75),lwd=c(1,2,1),col=c(1,2,1))

|  |
| --- |
| 图 5.9: 使用 Schnute 和 Richards 统一生长曲线的假设示例中达到成熟长度的比例。通过绿色虚线所示的四分位距左右两侧的差异，展示了该逻辑斯蒂曲线的不对称性。 |

使用 [图 5.9](#fig-5-9) 中使用的参数作为基准，我们可以单独改变各个参数来确定其对结果曲线的影响。 和 参数相对影响四分位距，而所有四个参数都影响 。

Schnute（1981）的模型在文献中似乎比 Schnute 和 Richards（1990）的更通用的模型使用得更多，这在 Schnute 和 Richards 模型旨在展示所有这些不同曲线可以统一，并因此具有一定程度的共性时是自然的。这强调了这些曲线主要提供的是生物学过程的经验描述，而这些过程本身是所研究种群的涌现属性。严格地对这些参数进行生物学解释，并期望自然界总是提供合理的或有意义的参数值，这是过度推断。例如，Knight（1968）描述的情况，他讨论的是生长，但实际上是在讨论是否应该对经验模型参数进行明确解释。他指出：“更为重要的是，将 视为自然事实，而不是数据分析的产物，会引发扭曲的观点。”（Knight, 1968, p 1306）。 描述性而非解释性的模型（关于过程的假设性描述）仍然是经验性描述，当尝试解释其参数时需要格外小心。

## 5.7 补充量

### 5.7.1 引言

种群生物量生产的主要贡献者包括新个体对种群的补充和个体的生长（其中，种群的定义意味着我们无需考虑迁入）。再次强调，关于补充过程的文献非常丰富（Cushing，1988；Myers 和 Barrowman，1996；Myers，2001），但在这里我们将主要关注在尝试将其动态纳入模型时如何描述它。与其他生长和成熟静态模型一样，这些模型被认为随时间保持不变，尽管在种群评估模型中存在时间分段选项，可以在特定界定的时间段内使用不同的参数集（Wayte，2013）。本节中我们将重点关注简单的描述。

鱼类种群的补充量自然倾向于高度变异，不同物种的变异程度有所不同。如果你觉得我在一句话中频繁使用”变异”及相关术语，或许这会强化补充量确实高度变异的观点。渔业科学家面临的主要问题是补充量是由繁殖群体资源量、环境变异或两者的某种组合决定的。这让人回想起 20 世纪 50 年代种群动态学中的争论，即种群是否受密度依赖或密度独立效应的控制（Andrewartha and Birch, 1954）。这个变异问题导致了一种普遍但错误的信念，即在一个观察到的群体规模范围内，渔业的补充量实际上与成鱼群体规模无关。这可能是一个危险的假设（Gulland, 1983）。补充量与繁殖群体资源量之间没有关系的观点认为，科学家和管理者可以忽视群体补充量关系，除非有明确证据表明补充量与群体资源量无关。 种群补充量无关系的观点源于通常情况下关于这种关系的数据显得非常分散，没有明显的特征。

尽管存在争议，但在渔业模型中，对 补充过程的建模仍然具有重要意义。因为用这些模型提供从当前时间点出发的管理建议时，通常会在评估不同捕捞或努力量水平的影响时，将模型动态向前预测。通过在这些预测中包含不确定性，可以估计不同管理方案未能实现预期管理目标的相对风险（Francis，1992）。但要进行此类预测，需要了解鱼群的生产力。假设已包含了个体生长动态，那么对补充量水平的时间序列估计或某种定义的鱼群补充量关系可以用来为这些预测生成未来补充量水平的估计。

在本章中，我们将仅考虑鱼类种群补充量关系的数学描述，并且大多忽略这些关系背后的生物学因素，尽管总会有例外。我们将回顾两种最常用的鱼类种群补充量数学模型，并讨论它们在复杂程度不同的种群评估模型中的应用。

### 5.7.2 “良好”种群补充量关系的性质

Cushing（1988）为种群补充关系的生物学过程提供了很好的介绍，他概述了卵和幼鱼死亡的原因，并提供了丰富的实例和该主题的参考文献。与其他讨论的基本生产过程一样，关于种群补充量关系的生物学及其调节因素的文献非常丰富。

已有研究已证实多种生物学和物理因素对补充量结果有影响。我们不会考虑任何真实物种的生物学细节，除非指出种群资源量与补充量之间的关系并非决定性，且可能存在多种反馈形式影响结果。已提出多种描述种群补充量关系的数学模型，但我们仅考虑 Beverton 和 Holt 以及 Ricker 的模型，而 Deriso-Schnute（Deriso, 1980; Schnute, 1985）等模型也同样重要。

Ricker (1975) 列出了他认为理想的平均种群补充关系的四个特性：

* 种群补充曲线应通过原点；也就是说，当种群数量为零时，补充量应为零。这假设所考虑的观测值与总种群相关，并且不存在由移民组成的“补充”。
* 补充量不应在高种群密度时降至零。这不是一个必要条件，尽管观察到补充量随种群密度增加而下降，但并未观察到补充量降至零的情况。即使种群在最大生物量时达到平衡，补充量仍应与自然死亡率相匹配。
* 补充量率（单位繁殖群体补充量）应随着亲本群体的增加而持续减少。只有当正密度依赖机制（补偿性机制）在起作用时（例如，群体增加导致幼体死亡率增加），这种情况才是合理的。但如果负密度依赖机制（消耗性机制）在起作用（例如，捕食者饱和和 Allee 效应；Begon and Mortimer, 1986），那么这并不总是成立。
* 补充量必须超过亲代种群在亲代种群可能范围的部分。严格来说，这仅适用于在死亡前只繁殖一次的物种（例如鲑鱼）。对于寿命更长、多次繁殖的物种，应解释为补充量必须足够高，以超过因年度自然死亡率造成的损失。

Hilborn 和 Walters（1992）提出了另外两种他们认为与良好种群补充关系相关的普遍特性：

* 平均繁殖群体补充量曲线应该是连续的，在群体规模的小幅变化中不应有剧烈变化。他们指的是连续性，即平均补充量应该随着群体规模平滑变化，与上述条件 3 相关。
* 平均种群补充关系的稳定性是随时间不变的。这就是平稳性，即这种关系不会随时间发生显著变化。这一假设在生态系统发生显著变化（其中被开发种群是其中一部分）的系统中可能不成立。但在模型中，可以通过使用参数时间块来处理这种情况（例如，参见 Wayte，2013 年）。

### 5.7.3 补充型过度捕捞

“过度捕捞（overfishing）”这一术语通常在两种情境下进行讨论，即生长型过度捕捞（growth overfishing）和补充型过度捕捞（recruitment overfishing）。生长型过度捕捞是指捕捞强度过大，导致最终大多数个体在相对较小时就被捕捞。这在“单位补充量渔获量”（YPR）方面是一个问题。YPR 的分析着重于平衡因总死亡率导致的种群损失与因个体生长带来的种群增长，旨在确定开始捕捞该物种的最佳个体大小和年龄（这里的“最佳”可以有多种含义）。生长型过度捕捞是指鱼群在没有时间达到这种最佳个体大小之前就被捕获。

补充型过度捕捞发生时，意味着对鱼群捕捞过度，使得繁殖群体规模降低到其作为一个种群无法产生足够的新补充量来替代那些死亡（无论是自然死亡还是其他原因）的个体水平。显然，这种情况不可能持续太久，不幸的是，过度捕捞通常会导致渔业崩溃。虽然需要记住，渔业崩溃通常意味着渔业不再具有经济可行性，但这并不意味着字面意义上的灭绝。

虽然生长型过度捕捞相对容易检测（如果种群处于 YPR 最佳状态？尽管当然，可能会有复杂情况）。不幸的是，对于补充型过度捕捞的检测来说，情况并非如此，这可能需要确定成熟或产卵种群资源量与补充水平之间的关系。这已被证明是许多渔业面临的艰巨任务。相反，随着正式捕捞策略的出现，人们普遍的做法是确定一个被认为对后续补充构成不可接受风险的繁殖生物量水平。一个非常常见的极限参考点是未受干扰繁殖生物量的 20%（ ）。关于极限参考点耗竭水平 的最早参考似乎来自 Beddington & Cooke (1983)。他们以如下方式解释了他们对不同种群潜在产量分析所施加的限制：

“…使用预期未开发产卵生物量 20%的逃逸水平。这个数字并非保守估计，但它代表了一个较低的限制，即在此水平下可能观察到补充量的下降。”（Beddington & Cooke, 1983, 第 9-10 页）。

导致产生 （注意 的不同表示方式）作为潜在过度捕捞指标的一个合理捕捞强度的观念的最具影响力的文件，是一份为美国国家海洋渔业服务局准备的文件（Restrepo 等，1998）。事实上，他们推荐 ，但认为 是该数值的一个可接受的替代值。然而，需要注意的是，这只是一个“经验法则”，没有实证基础将替代值 或 联系起来。实际上，对于某些物种选择 可能导致 远低于 。

### 5.7.4 Beverton-Holt 补充量

历史上，Beverton 和 Holt（1957）的种群补充曲线因其简单的解释而有用，也就意味着可以从基本原理推导出这种关系。对他们来说，数学上易于处理也很重要，因为当时只有解析方法可行。事实上，它持续使用似乎很大程度上源于惯性和传统。请注意，如果我们仅仅将 Beverton-Holt 曲线视为一种数学描述，那么实际上任何具有前面列出的良好性质的曲线都可以使用。然而，人们谈论“Beverton-Holt 种群补充模型”，但它有多种形式。有两种常见的使用形式：

其中 代表第 年的补充量， 代表第 年的繁殖生物量， 和 是两个 Beverton-Holt 参数， 表示预测模型值与自然观测值之间的关系呈对数正态分布（常表示为 ）。 值决定渐近极限（ ），而 的不同值与每条曲线达到渐近线的速度成反比，从而决定原点附近的相对陡峭程度（一个关键词，我们将会更多地讨论它）（ 的值越小， 补充量达到最大值的速度越快）。当然，这是一种平均关系，曲线周围的散布与曲线本身同样重要。一个常见的替代重新参数化公式是：

其中 代表最大最大补充 量（即 ），而 代表为产生平均最大补充量一半（即 ）所需的产卵群体（即 ）。使用 年的生物量代表产生第 年补充量，这会影响产卵时间。如果产卵发生在十二月，而仔稚鱼沉底发生在次年，那么这样是正确的；如果两者发生在同一年，那么显然需要更改下标。需要注意的是，生物学现实，在这种情况下与时间相关，甚至可能渗入非常简单的模型中。数学模型可以提供对自然的绝佳表征，但需要了解所建模动物的生物学特性，以避免简单的错误！

从 [图 5.10](#fig-5-11) 中可以看出，Beverton–Holt 曲线的初始陡峭程度以及渐近值捕捉了该方程行为的重要方面。渐近线由参数 的值给出，而初始陡峭程度则由( )的值近似表示，这发生在 相对较小时。

#plot the MQMF bh function for Beverton-Holt recruitment Fig 5.11   
B <- 1:3000   
bhb <- c(1000,500,250,150,50)   
parset()  
plot(B,bh(c(1000,bhb[1]),B),type="l",ylim=c(0,1050),  
 xlab="Spawning Biomass",ylab="Recruitment")   
for (i in 2:5) lines(B,bh(c(1000,bhb[i]),B),lwd=2,col=i,lty=i)   
legend("bottomright",legend=bhb,col=c(1:5),lwd=3,bty="n",lty=c(1:5))   
abline(h=c(500,1000),col=1,lty=2)

|  |
| --- |
| 图 5.10: 有恒定 a = 1000 和五个递减的 b 值的Beverton-Holt 种群补充曲线，导致越来越高效的曲线。 |

### 5.7.5 Ricker 补充量

由 Ricker（1954，1958）提出了 Beverton-Holt 曲线的替代方案，但这同样有多种参数化方式：

其中 代表 年产卵群体 产生的补充量， 代表低种群水平时的单位繁殖资源量-补充量， 与单位繁殖补充量随 增加而减少的速率相关。 表示关系与观测数据之间的对数正态残差误差。请注意，参数 和 与 Beverton–Holt 方程中的参数非常不同。该种群-补充量曲线不会达到渐近线，而是在高种群水平时表现出补充量水平的下降 [图 5.11](#fig-5-12) 。

#plot the MQMF ricker function for Ricker recruitment Fig 5.12   
B <- 1:20000   
rickb <- c(0.0002,0.0003,0.0004)   
parset()  
plot(B,ricker(c(10,rickb[1]),B),type="l",xlab="Spawning Biomass",   
 ylab="Recruitment")   
for (i in 2:3)   
 lines(B,ricker(c(10,rickb[i]),B),lwd=2,col=i,lty=i)   
legend("topright",legend=rickb,col=1:3,lty=1:3,bty="n",lwd=2)

|  |
| --- |
| 图 5.11: 两条具有相同常数 但具有不同 值的 Ricker 种群补充量曲线。请注意， 值主要影响补充量随生物量增加而下降的水平，对初始陡峭程度影响较小。 |

补充量随着繁殖生物量增加而下降背后的理念是，这与某些群体对其他群体的竞争或捕食效应（同类相食）有关。已经提出了各种机制，包括成鱼捕食幼鱼、疾病在密度依赖性传播、繁殖成鱼对彼此产卵场的损害（主要发生在像鲑鱼这样的鱼类生活的河流中），以及最终密度依赖性生长与大小依赖性捕食相结合。这些机制中的每一种都可以导致对 Ricker 曲线参数的不同解释。

再次，方程是否应该被解释为对可观察世界的说明，而不仅仅是平均补充量的便利经验描述，这一点变得重要。此外，尽管参数当然可以被赋予现实世界的解释，但方程仍然倾向于过于简单，最好被视为对事件的说明，而不是经验描述（Punt and Cope, 2019）。

### 5.7.6 Deriso 的通用模型

Deriso (1980) 提出了一个通用方程，其中 Beverton–Holt 和 Ricker 种群补充曲线是它的特例。Schnute (1985) 重新构建了 Deriso 的方程，产生了一个更加灵活的版本，具有更高的灵活性：

在这个 Schnute (1985) 通用情况下有三个参数 、 和 ，前两个参数应始终为正，但 可以为正或负。通过修改 值，会出现不同的特例 [图 5.12](#fig-5-13) ：

第三个情况中的箭头表示“趋近”，例如 趋近于零。第一个情况是密度无关的补充率常数，也可以通过设置 得到。接下来的三种情况分别对应于 Beverton–Holt (1957)、Ricker (1954, 1958) 和 Schaefer (1954) 的标准种群-补充量关系，这是一种逻辑斯蒂曲线的形式。当然，在每种情况下都需要小心选择 和 的适当值。

Deriso–Schnute 模型存在一些数学上不稳定的特性，如果我们设 ，这一点应该很清楚。这会导致数学上的奇点（除以零）。参数限制应该始终为 ，无论是从负方向还是正方向。使用这个方程，有许多参数组合会产生不合理的种群补充关系。这个方程的主要价值在于展示不同曲线之间的关系，虽然人们不会在拟合模型中使用 Deriso-Schnute 模型，但它可能在模拟模型中很有用。

# plot of three special cases from Deriso-Schnute curve Fig. 5.13   
deriso <- function(p,B) return(p[1] \* B \*(1 - p[2]\*p[3]\*B)^(1/p[3]))   
B <- 1:10000   
plot1(B,deriso(c(10,0.001,-1),B),lwd=2,   
 xlab="Spawning Biomass",ylab="Recruitment") # BH   
lines(B,deriso(c(10,0.0004,0.25),B),lwd=2,col=2,lty=2) # DS   
lines(B,deriso(c(10,0.0004,1e-06),B),lwd=2,col=3,lty=3) # Ricker   
lines(B,deriso(c(10,0.0004,0.5),B),lwd=2,col=1,lty=3) # odd line   
legend(x=7000,y=8500,legend=c("BH","DS","Ricker","odd line"),   
 col=c(1,2,3,1),lty=c(1,2,3,3),bty="n",lwd=3)

|  |
| --- |
| 图 5.12: Beverton-Holt (BH)、Ricker 和 Deriso-Schnute (DS) 种群-补充量曲线的比较， 这些曲线在 Deriso-Schnute 通用方程 Equ (5.19) 中实现。 对于 DS 曲线， 导致不切实际的结果（奇线）。 |

### 5.7.7 重新参数化的 Beverton-Holt 方程

在《模型参数估计》( [章节 4](#sec-paraestimat)) 章中，对 tigers 数据集中的虎虾 Beverton-Holt 种群-补充量曲线的参数进行了估计。相对丰度的估计可以等同于短命对虾种类的补充量水平，但对于使用年龄结构模型的寿命更长的物种，估计此类参数将相对抽象。Francis（1992）对更复杂的种群评估模型做出了重要发展，他重新参数化了 Beverton-Holt 种群-补充量关系，以曲线的初始陡峭程度 和初始补充量 来表示，其中 是从未捕捞或原始繁殖生物量 推导出来的。他首先将这些想法应用于胸棘鲷（*Haplostethus atlanticus*）的年龄结构剩余生产模型，该模型具有捕捞数据、来自调查的相对丰度指数以及与年龄相关生长和重量的生物学数据。使用这些数据，他能够以更合理的方式拟合模型，并利用他对补充量的更有意义的参数化方法。

Francis（1992）将陡峭程度，记作 ，定义为当成熟生物量减少到未捕捞水平的 20%时，种群产生的确定性补充量。他开始描述时使用了：

因此，通过定义陡峭度， ，我们得到：

其中 和 是 Beverton-Holt 参数， 是未受捕捞种群中稳定年龄分布的单位补充量总繁殖生物量。”单位补充“ 部分对年龄结构模型很重要，因为它允许我们独立于 Beverton-Holt 方程 [公式 5.15](#eq-5_15) 来确定 和 之间的关系。稳定的年龄分布是由恒定的补充量水平 在恒定的自然死亡率（ ）产生的 ，导致年龄组数量呈现标准指数下降。如果自然死亡率较低，则可能需要增加一个年龄组。这个稳定年龄分布可以定义为：

其中 表示年龄 的单位补充量未被捕捞的数量，而 表示模型中模拟的最大年龄。 作为一组附加年龄组，因此需要除以 以提供指数序列的和。生物量 将是具有恒定补充水平为 1 时产生的种群生物量。因此，对于生物量 ，在稳定的年龄分布下，产生的补充水平将是 （反之亦然）。

其中 表示成熟年龄， 表示年龄 的单位补充量未被捕捞的数量， 表示年龄， 表示年龄 的单位鱼体重量。人们还可以将部分自然死亡率纳入此方程，使其在一年中的某个时间点与补充量相等。 作为缩放因子，因为在任何恒定的补充量水平下，未受干扰的种群都会出现稳定的年龄分布。 的幅度将根据估计的初始生物量 进行缩放，但其值相对于维持稳定年龄分布所需的恒定补充量 将保持不变。在实践中，人们会根据生物学信息推导出 ，并在种群评估模型中估计 。

Francis（1992）随后也使用了他对补充量陡峭程度的定义来重新参数化Beverton-Holt 参数 、 、 和 （见本章附录）：

将这些公式代入 [公式 5.16](#eq-5_16) 中，得到：

将上层的 移到下层：

然后，在可能的情况下消去 ，可以将 Beverton-Holt 重新定义为陡峭度 、 和 ，这些都有更明确的解释，并且使用 [公式 5.25](#eq-5_25) 或 [公式 5.28](#eq-5_28) 将更容易在评估模型中进行估计：

### 5.7.8 重新参数化的 Ricker 方程

类似地，Ricker 种群补充方程可以重新参数化为陡峭度 、 和 ：

[公式 5.25](#eq-5_25) 和 [公式 5.28](#eq-5_28) 现在都是用于种群评估和试图包含补充量的模拟模型中更常用的参数化方法。

## 5.8 选择性

### 5.8.1 引言

在种群评估模型（及模拟）中使用的另一类静态模型与渔具对特定物种的选择性有关。其核心思想是，如果一个被开发物种的种群在某个区域存在，那么如果使用特定的渔具（如拖网、丹麦围网、刺网、龙虾笼等），渔具的构造及其使用方式将影响在遇到渔具时，哪些可用的种群成员会对其变得脆弱。在年龄结构评估模型中，选择性的概念因可用性的概念而变得复杂。例如，如果某个物种的主要捕捞场所在水深超过 250 米的区域，而较浅水域中主要只有较小的幼体存在，那么如果仅使用浅水域的数据来估计渔具的选择性，其结果很可能与使用深水域数据得到的估计结果不同。实际上，在评估模型中估计的选择性曲线应该被视为选择性/可用性曲线。

选择选择性曲线的形状是一个重要的决策。某些捕捞工具通常用特定的方程来描述其选择性。因此，拖网工具的选择性通常用逻辑斯蒂方程来描述，而长线工具（使用钩子）通常用拱形选择性函数来描述。

只有在有捕捞年龄或大小组成数据的情况下，才能在资源评估模型中拟合选择性模型。正如我们在模型参数估计章节中所看到的，通常会用多项式似然来拟合这种组成数据。在资源评估模型中，通常会根据预期年龄数量来模拟种群动态，这会隐含某些大小数量。因此，在尝试生成预测的捕捞组成数据（无论是年龄还是大小）时，需要将可用的年龄或大小数量乘以预测的选择性。因此，在拟合过程中，组成数据会帮助估计选择性参数。

本节中我们仅将说明其他选择性的方程式，并展示它们的不同特性。

### 5.8.2 逻辑斯蒂选择

描述不同渔具选择性特征的不同方程式有很多，但其中一种极其常见的是标准逻辑斯蒂曲线或 S 形曲线，这通常是拖网渔具选择性的典型特征。它意味着随着年龄或个体大小的增加，渔具的易受害性会逐渐增加，直到 100% 的个体在遇到渔具时都变得易受害（这种逐渐增加到 100% 的过程与成熟过程相同）。通常使用两个方程式，第一个由 **MQMF** 函数 logist() 描述：

其中 是年龄 的选择性（比例）， 是选择性达到 50% 时的年龄， 是曲线的斜率，定义为从选择性 50%到选择性 95%之间的年数。当我们谈论年龄时，同样也可以谈论大小。 的上限为 95%（实际上 是 ）。另一种对数曲线，我们在《简单种群模型》[章节 3](#sec-simplemodel) 一章的 “YPR 中的选择性” [章节 3.3.1](#sec-ypr_selectivity) 节中已经见过，也被用来描述年龄成熟度，它在函数 mature() 中定义：

其中 和 是逻辑斯蒂参数， 是0.5 选择性（即 50%）的年龄。四分位距（字面意思是 25%分位数到 75%分位数）定义为 （有关此函数的实现，请参阅 **MQMF** 函数 mature()）。通常，在年龄结构模型中，需要长度或年龄组成数据，以便可以直接估计渔具选择性和渔业可利用性。在进行单位补充量渔获量计算时，通常包含选择性的原因是确定开始应用捕捞死亡率的最佳年龄。出于这个原因，通常使用所谓的刀刃选择性，它本质上识别出以下没有选择性的特定年龄，以及以上 100% 选择性的年龄。这在 **MQMF** 函数 logist() 中实现；尽管刀刃选择性不再倾向于用于完整的年龄结构种群评估模型，但它仍然用于延迟差分模型（Schnute，1985；Hilborn 和 Walters，1992）。

#Selectivity curves from logist and mature functions See Fig 5.14  
ages <- seq(0,50,1); in50 <- 25.0   
sel1 <- logist(in50,12,ages) #-3.65/0.146=L50=25.0   
sel2 <- mature(-3.650425,0.146017,sizeage=ages)   
sel3 <- mature(-6,0.2,ages)   
sel4 <- logist(22.0,14,ages,knifeedge = TRUE)   
plot1(ages,sel1,xlab="Age Years",ylab="Selectivity",cex=0.75,lwd=2)   
lines(ages,sel2,col=2,lwd=2,lty=2)   
lines(ages,sel3,col=3,lwd=2,lty=3)   
lines(ages,sel4,col=4,lwd=2,lty=4)   
abline(v=in50,col=1,lty=2); abline(h=0.5,col=1,lty=2)   
legend("topleft",c("25\_eq5.30","25\_eq5.31","30\_eq5.31","22\_eq5.30N"),   
 col=c(1,2,3,4),lwd=3,cex=1.1,bty="n",lty=c(1:4))

|  |
| --- |
| 图 5.13: logist() 和 mature() 函数的逻辑 S 形曲线示例。虚线红色和实线黑色曲线具有相同的 L50，但梯度不同。点线绿色展示了改变 mature() 函数的 b 参数的效果，而交叉线蓝色曲线展示了刀刃式选择。图例显示了 L50 和使用的方程。 |

### 5.8.3 球面选择性

一种上升至峰值的选择性模式、可能存在平台段，然后下降的模式被称为穹顶形，这是网具（如围网）和钩具（如延绳钓）等渔具的典型特征。由于包含如此多的组成部分，选择性曲线往往更为复杂，因为上升段、平台段和下降段都需要连接在一起。现代拟合此类模型的方法倾向于使用自动微分软件，如 AD-Model Builder 或相关软件（Bull 等，2012；Fournier 等，2012；Kristensen 等，2016）。这意味着评估模型中的组件模型需要可微分，以便穹顶形选择性曲线的三个组成部分之间的连接需要连续（Methot 和 Wetzell，2013；Hurtado-Ferro 等，2014）。此类方程至少包含五个组成部分：上升段（*asc*）、选择性等于 1.0 的平台段、下降段（*des*）、以及连接这三个主要部分之间的两个连接函数、 和 ， [公式 5.32](#eq-5_32) ：

其中 表示长度 的选择性。各种分量函数定义如下：

其中 是长度组 的平均长度， 是最小长度组的平均长度， 是选择性达到 1.0 的个体大小， 是选择性从 1.0 开始下降的个体大小（如果 和 相等则没有平台期）， 影响上升段（*asc*）的斜率， 影响下降段（*dsc*）的斜率， 是 处选择性的对数， 是 选择性的对数， [图 5.14](#fig-5-15) 。

因此，需要 到 六个参数以及所使用的长度组的平均长度来定义这种穹顶选择性曲线。

#Examples of domed-shaped selectivity curves from domed. Fig.5.15   
L <- seq(1,30,1)   
p <- c(10,11,16,33,-5,-2)   
plot1(L,domed(p,L),type="l",lwd=2,ylab="Selectivity",xlab="Age Years")   
p1 <- c(8,12,16,33,-5,-1)   
lines(L,domed(p1,L),lwd=2,col=2,lty=2)   
p2 <- c(9,10,16,33,-5,-4)   
lines(L,domed(p2,L),lwd=2,col=4,lty=4)

|  |
| --- |
| 图 5.14: 由函数 domed() 产生的三个拱形选择性曲线示例，改变了达到选择性 1.0 的初始年龄（10、8、9），以及停止时的年龄（11、12、10），以及最终年龄组的选择性。 |

## 5.9 静态模型的结论性评述

比我们已考虑的剩余产量模型更复杂的种群评估模型，大多是混合了群体进展的动态模型和本章所说明的静态模型集合。因此，了解生长、成熟、选择性和补充模型对于理解更高级模型的结构至关重要。有时人们会在评估模型之外估计它们的参数，但通常估计会是拟合整个种群评估模型的一部分。一次性拟合所有独立模型组件的优势在于，组件之间的任何交互作用都可以自动考虑。

本章为一系列静态模型提供了入门基础。目标是当你遇到其他模型时，能够像我们这里使用的方法一样使用和拟合它们。每种这样的模型都有其自身的假设。只要你知道这些假设，你应该能够为你在建模中选择使用哪种静态模型做出决策并提供辩护。

## 5.10 附录：Fabens 变换的推导

Fabens（1965）将 von Bertalanffy 生长曲线进行转换，使其与标记计划提供的数据相匹配（标记时的长度和日期，以及重捕时的长度和日期）。von Bertalanffy 曲线的年龄-长度版本为：

在标记的背景下，它在给定时间增量（ ）后的长度定义为：

在指数项中，可以提取 的贡献：

现在，在 期间预期发生的生长增量 可以定义为 减去 ：

$$
\Delta L = (L\_{t + \Delta t}-L\_t) = \\
\left(L\_{\infty}-L\_{\infty}e^{-K(t-t\_0)}e^{-K \Delta t} \right) - \left(L\_{\infty} - L\_{\infty}e^{(-K((t-t\_0))} \right)
\qquad(5.37)$$

我们可以去掉括号，这将使第二项中的否定变为加号，然后重新排列分离的项。

这两个孤立的 项可以消去以提高证明的清晰度，但它们稍后会重新出现。此外，我们还可以重新排列剩下的两个指数项：

重新排列使得可以提出出 项来简化整个方程：

现在我们可以将 项移至左边，并返回 项

如果我们添加一些括号，将加号变为减号，我们最终就能认识到哪些项的组合等于 ：

通过将 项移至右侧，得到经典的 Fabens 生长增量方程：

需要注意的是，这种转换也会转换长度-年龄方程中的正态随机残差，而这些残差当然是在参数估计时使用的。这意味着，尽管这些参数具有相同的名称，但它们并不严格可比。

## 5.11 附录：Beverton-Holt 模型的重新参数化

Francis（1992）以更具有生物学意义的陡峭程度（ ）、 初始成熟生物量（ ）和 初始补充量（ ）等术语来定义 Beverton-Holt 参数。与其使用

不如使用另外一种公式，尽管最后还是要回到使用关系 和 。

因此，我们将使用：

其中 是年 的补充量， 是 年繁殖资源量， 和 是 Beverton-Holt 参数。在未受捕捞的平衡状态下，我们可以将方程表示为：

陡度（ ）定义为初始生物量的20% 时获得的补充量：

如果 的平衡方程代入这个方程，我们将得到：

因此：

通过乘法得到 ：

通过交换项以将 和 分开，然后乘以5 以消除 0.2：

简化后得到：

因此 可重新参数为：

该版本的 可以用于原始方程，并重新排列以对 参数做类似处理：

转换为：

除以 ，然后将等号左侧的第二项 乘以 ，简化为：

记住 和 ，所以就可以完成了：

以及：

如 Francis (1992) 所定义。这重新定义了 Beverton-Holt 模型参数，以 、 和 表示。需要确定 和 之间的关系，但我们不能使用平衡方程来计算，因此我们假设未受捕捞的种群处于具有稳定年龄分布的平衡状态。从稳定年龄分布中每单位亲体产生的成熟生物量 ( ) 因此定义了 和 之间所需的关系。

# 6. 不确定性

## 6.1 引言

根据一组数据拟合模型，需要寻找参数估计值，以优化观测数据与模型预测之间的关系。模型参数估计值代表了我们感兴趣的群体的属性。虽然在每种情况下都有可能找到最佳参数值，但无论使用什么数据，都只是从总体中抽取的一组样本。假设可以从同总体中抽取不同的、独立的同类数据样本，如果对它们进行独立分析，很可能会得出至少略有不同的参数估计值。因此，在根据数据拟合模型时，估计参数的确切值其实并不是最重要的问题，相反，我们想知道的是，如果我们能够获得多个独立样本，这些估计值的可重复性有多大。也就是说，参数只是估计值，我们想知道对这些估计值有多大把握。例如，通常情况下，高度多变的数据通常会导致每个模型参数都可能具有广泛的可信估计值分布，通常情况下，这些估计值的置信区间也会很宽。

在本章中，我们将探讨其他可用的方法，以描述任何建模情况中至少一部分固有的不确定性。虽然可能有许多潜在的不确定性来源会阻碍我们管理自然资源的能力，但这里只能对其中的一部分进行有用的研究。一些不确定性来源会影响所收集数据的可变性，其他来源则会影响可用数据的类型。

### 6.1.1 不确定性类型

有多种方式来描述不同类型的不确定性，这些不确定性影响渔业模型参数估计以及如何使用这些估计算。这些一般被称为误差来源，通常是指残差误差而不是指已经犯下的错误。不幸的是，“误差”这一术语，如残差误差，有导致混淆的潜力，因此最好使用“不确定性”这一术语。虽然只要你知道这个问题，它对你来说不应该是个问题。

Francis & Shotton (1997) 列出了与自然资源评估和管理相关的六种不同类型的不确定性，而 Kell 等 (2005) 在遵循 Rosenberg 和 Restrepo (1994) 的基础上，将这些缩减为五种。或者，它们都可以归纳为四个标题下（Haddon, 2011），其中一些有子标题，如下所示：

* 过程不确定性：种群动态率（如生长、年际平均补充量、年际自然死亡率）和其他生物学特性及过程中的潜在自然随机变化。
* 观测不确定性：抽样误差和测量误差反映了样本旨在代表总体但仅是样本这一事实。数据收集不充分或非代表性会加剧观测不确定性，任何错误、数据的故意误报或未报告都会导致观测不确定性，这在渔业统计领域并不罕见。
* 模型不确定性：与所选模型结构描述研究系统动力学的能力相关：
  + 不同的结构模型可能会提供不同的答案，并且对于哪个模型是自然界更好的表示存在不确定性。
  + 为给定过程选择残差误差结构是模型不确定性的一种特殊情况，这对参数估计具有重要影响。
  + 估计不确定性是模型结构中参数之间存在相互作用或相关性的一种结果，以至于略微不同的参数集可能导致对数似然值相同（在数值模型拟合中使用的精度范围内）。
* 实施不确定性：管理行动的效果或范围可能与预期不同。定义不明确的管理选项可能导致实施不确定性。
  + 制度不确定性：管理目标定义不明确导致无法实施的管理。
  + 决策制定与实施之间的时间滞后可能导致更大差异。评估通常在渔业数据收集一年或更长时间后进行，而管理决策的执行往往又需要一年或更长时间。

我们这里特别关注过程、观测和模型的不确定性。每种不确定性都可能影响模型参数估计、模型结果和预测。实施不确定性更多是关于模型或模型的结果在管理自然资源时如何使用。这会对基于库存评估模型结果的不同管理策略的效率产生重要影响，但它并不影响这些即时结果（Dichmont 等，2006）。

模型不确定性可以是定量的，也可以是定性的。因此，使用完全相同的数据和残差结构的模型可以相互比较，并选择最佳拟合的模型。这些模型可以被认为既相关又不同。然而，当模型不相容时，例如，使用相同结构模型但采用不同的残差误差结构时，它们各自可以产生最佳拟合，模型选择必须基于除拟合质量之外的其他因素。这类模型之间不能平滑过渡，而是构成对所研究系统的定性和定量上不同的描述。模型不确定性是模型选择背后的驱动力之一（Burnham and Anderson, 2002）。即使只有一个模型被开发出来，它也往往是从许多可能的模型中隐含选择出来的。在特定情况下使用多种类型的模型（例如，同时使用过剩生产模型和完全年龄结构模型）往往能带来仅使用一种模型会错过的见解。 遗憾的是，随着用于种群评估建模的资源普遍减少，现在使用多个模型已成为一种日益罕见的选择。

模型和实施的不确定性在自然资源管理中都十分重要。我们已经比较了不同模型的输出结果，然而，在这里我们将重点介绍能够帮助我们表征在管理建议方面具有意义的不同模型参数估计值和其他模型输出结果的可接受置信度的方法。

针对任何参数估计或其他模型输出，存在多种策略来表征不确定性。一些方法侧重于数据问题，而另一些方法则侧重于在现有数据条件下，合理参数值的潜在分布。我们将考虑四种不同的方法：

1. 自助法（bootstraping），关注数据样本中固有的不确定性，并通过检验如果采取略有不同的样本对参数估计的影响来运作。
2. 渐近误差（asymptotic errors）使用参数估计之间的协方差矩阵来描述这些参数值周围的不确定性，
3. 似然曲线（likelihood profiles）基于主要参数构建，以获得每个参数更具体的分布，最后，
4. 贝叶斯边缘后验分布（Bayesian marginal characterize）表征模型参数和输出估计中固有的不确定性。

这些方法都允许对参数和模型输出的不确定性进行表征，并提供确定每个参数均值或期望值周围选定百分位数范围的方法（例如， 在某些情况下可能不对称）。

我们将使用一个简单的剩余生产模型来说明所有这些方法，尽管它们应该具有更广泛的适用性。

### 6.1.2 示例模型

这是在 “模型参数估计”（ [章节 4](#sec-paraestimat) ）一章中关于使用对数正态似然拟合动态模型部分所描述的相同模型。我们将在本章的许多示例中使用该模型，其种群动态相对简单，如 [公式 6.1](#eq-6_1) 所示。

其中 表示第 年的可利用生物量， 为可利用数据第一年的生物量（ ），考虑了记录开始时的任何初始消耗。 是内在自然增长率（种群增长率项）， 是承载力或未捕捞的可利用生物量，在别处通常称为 （不要与 混淆）。最后， 是第 年的总捕捞量。为了将这些动态与除捕捞量以外的渔业观测结果联系起来，我们使用一个相对丰度指数（ ，通常是单位努力捕捞量或 cpue，但也可以是调查指数）。

其中 是第 年的捕捞率（CPUE 或 cpue）， 是第 年的捕捞努力量，而 被称为可捕系数（它也可能随时间变化，但我们假设它是恒定的）。由于 仅将种群生物量与捕捞量进行比例缩放，我们将使用可捕系数的封闭形式估计方法来减少需要估计的参数数量（Polacheck 等，1993）。

我们将使用 **MQMF** *abdat* 数据集拟合这个模型，然后在接下来的章节中检查该模型拟合的不确定性。我们将通过最小化基于对数正态分布的残差来拟合模型。简化后如下：

其中 是参数向量 ( ， 和 )， 是在年份 观测到的 cpue 值， 的最大似然估计定义为：

注意除以 而不是 。由于唯一的非恒定值是 ，使用最大似然法得到的结果与使用最小二乘残差法得到的结果相同（只要观察到的和预测的 cpue 都经过对数转换）。然而，在分析不确定性时，使用最大似然法比使用平方和法具有优势。严格来说， 值也是一个模型参数，但在这里我们特别处理它，只是为了说明与最小二乘方法的等价性。

#Fit a surplus production model to abdat fisheries data   
library(MQMF)  
library(ggplot2)  
library(knitr)  
  
data(abdat); logce <- log(abdat$cpue)   
param <- log(c(0.42,9400,3400,0.05))   
label=c("r","K","Binit","sigma") # simpspm returns   
bestmod <- nlm(f=negLL,p=param,funk=simpspm,indat=abdat,logobs=logce)   
outfit(bestmod,title="SP-Model",parnames=label) #backtransforms

nlm solution: SP-Model   
minimum : -41.37511   
iterations : 20   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient transpar  
r -0.9429555 6.743051e-06 0.38948  
K 9.1191569 -9.223729e-05 9128.50173  
Binit 8.1271026 1.059182e-04 3384.97779  
sigma -3.1429030 -8.116218e-07 0.04316

模型拟合情况如 [图 6.1](#fig-6-1) 所示。Schaefer 模型的简单动力学似乎为这些观察到的鲍鱼捕捞率数据提供了一个合理的描述。实际上，在时间序列中，法律最小尺寸有所变化，渔业内部引入了区域划分，以及其他重要变化，因此这一结果最多只能被视为一种近似，并且只能被视为提供了一种方法论的示例。观察到/预测到的对数正态残差，如果乘以预测值，显然会得到观察值（灰线）。这里展示这一点，因为我们即将在自助法（bootstrap）数据时使用这种简单关系。

最大持续产量（MSY）可以从 Schaefer 模型中简单地估计为 ，这意味着这种最佳拟合表明存在 。在接下来的各节中，我们将尝试回答的两个问题是：预测 cpue 在 [公式 6.1](#eq-6_1) 中观察数据周围的合理分布范围是多少？以及，平均 MSY 估计值周围的 90%置信区间是多少？

#plot the abdat data and the optimum sp-model fit Fig 6.1   
predce <- exp(simpspm(bestmod$estimate,abdat))   
optresid <- abdat[,"cpue"]/predce #multiply by predce for obsce   
ymax <- getmax(c(predce,abdat$cpue))   
plot1(abdat$year,(predce\*optresid),type="l",maxy=ymax,cex=0.9,   
 ylab="CPUE",xlab="Year",lwd=3,col="grey",lty=1)   
points(abdat$year,abdat$cpue,pch=1,col=1,cex=1.1)   
lines(abdat$year,predce,lwd=2,col=1) # best fit line

|  |
| --- |
| 图 6.1: *abdat* 数据集的 Schaefer 剩余产量模型 的最佳拟合用在线性空间表示（红色实线）。灰线穿过数据点，以说明与预测线的差异。The optimum fit of the Schaefer surplus production model to the abdat data set plotted in linear-space (solid red-line). The grey line passes through the data points to clarify the difference with the predicted line. |

## 6.2 自助法（Bootstrapping）

从总体中采集的数据被视为（假定）能够代表该总体以及预期样本值的潜在概率密度分布。这是一个非常重要的假设，最早由 Efron（1979）提出。他提出了这样一个问题：当样本包含或就是关于总体所有可用信息时，为什么不能假设样本真的就是总体，以估计检验统计量的抽样分布？因此，给定一个包含 个观测值的原始样本，自助样本将是从原始样本中**有放回地**抽取的 个观测值的随机样本。自助样本（即对样本数据值进行有放回的随机重抽样）被假定为近似那些通过反复抽样原始抽样总体所可能产生的值的分布。这些自助样本中的每一个都被视为来自原始总体的独立随机样本。这种有放回的重抽样对某些人来说似乎有违直觉，但可以用来拟合标准误差、百分位数置信区间以及进行假设检验。 据报道，“bootstrap”这个名字来源于故事《莫丘森男爵的冒险记》，在这个故事中，男爵通过自己拉自己的靴带而逃脱溺水，从而从一口井中脱身（Efron and Tibshirani, 1993）。

### 6.2.1 经验概率密度分布

假设给定一个来自总体的样本，非参数的最大似然估计值就是该样本本身。也就是说，如果样本包含 个观测值 ，那么对于 个观测值中的每个观测值 ，其概率密度分布的最大似然非参数估计值就是将概率质量的 。必须强调的是，这并不意味着所有值都有相等的似然，而是意味着每个观测值出现的似然性相等，尽管可能有多个观测值具有相同的值（在进行下一步之前，请确保你清楚这个区别！）。如果被抽样的总体变量有一个众数值，那么我们期望，有时会得到与该众数附近相同或相似值的出现频率高于样本分布两端的值。

自助法（Bootstrapping）包括应用蒙特卡洛方法，即从原始样本本身中进行有放回抽样，仿佛它是一个理论统计分布（类似于正态分布、伽马分布和贝塔分布）。有放回抽样与本质上无限大的总体一致。因此，我们将样本视为代表整个总体。

总之，自助法用于估计参数值或模型输出的不确定性。这是通过汇总来自重复样本的自助法参数估计值来实现的，这些重复样本是通过用从原始样本中估计的样本替换真实总体样本而得到的。

## 6.3 简单自助法示例

为了了解如何在 R 中实现自助法，从简单的例子开始是明智的。澳大利亚北部对虾渔业，在卡奔塔里亚湾和约瑟夫·波拿巴湾之间，沿着大陆右上侧，是一个混合渔业，捕获多种对虾（Dichmont 等，2006；Robins 和 Somers，1994）。我们将使用 1970 年至 1992 年间捕获的虎虾（*Penaeus semisulcatus* 和 *P. esculentus*）和基围虾（*Metapenaeus endeavouri* 和 *M. ensis*）的例子。这些捕获量之间似乎存在相关性， [图 6.2](#fig-6-2) ，但数据大量分散。刀额新对虾在价值更高的虎虾渔业中总是作为副产品捕获，这反映在其相对捕获量上。

虾捕捞数据相对较为嘈杂，这在虾捕捞中并不意外。捕捞基围虾与虎虾捕捞的相关性也不应令人惊讶。基围虾通常被视为更有价值的虎虾渔业中的兼捕物，因此可以预期虎虾总捕捞量与捕捞努力虾总捕捞量之间存在某种关系。另一方面，如果你将香蕉虾捕捞量与虎虾捕捞量作图，则不应期望存在这种关系，因为这两个渔业几乎相互独立（大多在同一区域），一个在白天捕捞，另一个在夜间捕捞。在混合渔业中探索这种关系，往往能提出关于物种间或不同渔业间相互作用可能性的假设。

#regression between catches of NPF prawn species Fig 6.2   
data(npf)   
model <- lm(endeavour ~ tiger,data=npf)   
plot1(npf$tiger,npf$endeavour,type="p",xlab="Tiger Prawn (t)",   
 ylab="Endeavour Prawn (t)",cex=0.9)   
abline(model,col=1,lwd=2)   
correl <- sqrt(summary(model)$r.squared)   
pval <- summary(model)$coefficients[2,4]   
label <- paste0("Correlation ",round(correl,5)," P = ",round(pval,8))   
text(2700,180,label,cex=1.0,font=7,pos=4)   
  
# ggplot(data = npf, aes(x = tiger, y = endeavour)) +  
# geom\_point() +  
# geom\_smooth(method = "lm", se = FALSE) +  
# theme\_bw() +  
# labs(x = "Tiger Prawn (t)", y = "Endeavour Prawn (t)")

|  |
| --- |
| 图 6.2: 1970 - 1992 年间澳大利亚北部对虾渔业中基围虾和虎虾产量之间的正相关关系（数据来自 Robins 和 Somers，1994）。The positive correlation between the catches of endeavour and tiger prawns in the Australian Northern Prawn Fishery between 1970 - 1992 (data from Robins and Somers, 1994). |

尽管回归高度显著（ ），但虾捕捞量的变异性意味着我们难以确定可以有多大信心相信相关性。通常情况下，在相关系数周围估计置信区间并不直接，幸运的是，使用自助法（bootstrapping），这可以轻松完成。我们可以从原始数据集中取出 23 对数据，进行 5000 次自助法抽样，每次计算相关系数。完成后，我们可以计算平均值和不同的分位数。在这种情况下，我们不是对单个值进行自助法，而是对值对进行自助法；我们必须取值对以保持任何内在的相关性。在 R 中，可以通过首先对每个数据向量的位置进行有放回抽样，以确定每个自助法样本要取哪些对进行重新抽样。

# 5000 bootstrap estimates of correlation coefficient Fig 6.3   
set.seed(12321) # better to use a less obvious seed, if at all   
N <- 5000 # number of bootstrap samples   
result <- numeric(N) #a vector to store 5000 correlations   
for (i in 1:N) { #sample index from 1:23 with replacement   
 pick <- sample(1:23,23,replace=TRUE) #sample is an R function   
 result[i] <- cor(npf$tiger[pick],npf$endeavour[pick])   
}   
rge <- range(result) # store the range of results   
CI <- quants(result) # calculate quantiles; 90%CI = 5% and 95%   
restrim <- result[result > 0] #remove possible -ve values for plot   
parset(cex=1.0) # set up a plot window and draw a histogram   
bins <- seq(trunc(range(restrim)[1]\*10)/10,1.0,0.01)   
outh <- hist(restrim,breaks=bins,main="",col=0,xlab="Correlation")   
abline(v=c(correl,mean(result)),col=c(4,3),lwd=c(3,2),lty=c(1,2))   
abline(v=CI[c(2,4)],col=4,lwd=2) # and 90% confidence intervals   
text(0.48,400,makelabel("Range ",rge,sep=" ",sigdig=4),font=7,pos=4)   
label <- makelabel("90%CI ",CI[c(2,4)],sep=" ",sigdig=4)   
text(0.48,300,label,cex=1.0,font=7,pos=4)

|  |
| --- |
| 图 6.3: 5000次自助法估算的基围虾和老虎虾渔获量之间相关性，原始均值用绿色虚线表示，自助法均值和 90% CI 用蓝色实线表示。出于图示目的，已删除了可能的负相关（尽管没有出现）。 |

虽然相关性值的分布显然向左偏斜，但我们有信心原始数据中的高相关性是现有关系数据的合理表示。

## 6.4 自助法时间序列数据

尽管前一个例子中的对虾捕捞量跨越了数年，但数据到达的具体年份与其之间的相关性无关。因此，我们可以对数据对进行自助法处理，并继续分析。然而，在自助法处理物种种群动态信息时，必须保持数据的时间序列特性，其中某年的数值（数量或生物量）以某种方式依赖于前一年的数值。例如，在剩余产量模型中，观测数据进入分析的顺序是种群动态的一个关键方面，因此盲目地自助法处理数据对既不合适也不明智。我们采用的解决方案是：我们首先获得最佳模型拟合及其预测的 cpue 时间序列，然后对每个点的个别残差进行自助法处理。在每个循环中，自助法抽样的残差应用于最佳预测值，以生成一个新的自助法“观测”数据序列，然后重新拟合。 你会记得观测值可以从最优预测 cpue 值乘以对数正态残差得到， [图 6.1](#fig-6-1) 。基于原始数据的最优解，特定年份 的对数正态残差为：

其中 指的是第 年观察到的单位捕捞努力量渔获量，而 是第 年预测的最优单位捕捞努力量渔获量。一个最优解将意味着最优预测单位捕捞努力量渔获量的时间序列和相关的对数正态残差的时间序列。鉴于我们使用的是乘法对数正态残差，一旦我们对残差进行自助采样（有放回的随机样本），我们需要将最优预测单位捕捞努力量渔获量的时间序列乘以自助残差序列。对数正态残差预计将围绕 1.0 中心，较低值受零约束，较高值无约束；因此可能出现对数正态分布的偏斜。

其中上标 表示自助值，如 ，或自助样本，如 。如果我们使用简单的加性正态随机残差，那么我们会使用右边的方程，但不会进行对数转换和指数化。对于剩余产量模型，对数正态版本可以这样实现：

# fitting Schaefer model with log-normal residuals with 24 years   
data(abdat); logce <- log(abdat$cpue) # of abalone fisheries data   
param <- log(c(r= 0.42,K=9400,Binit=3400,sigma=0.05)) #log values   
bestmod <- nlm(f=negLL,p=param,funk=simpspm,indat=abdat,logobs=logce)   
optpar <- bestmod$estimate # these are still log-transformed   
predce <- exp(simpspm(optpar,abdat)) #linear-scale pred cpue   
optres <- abdat[,"cpue"]/predce # optimum log-normal residual   
optmsy <- exp(optpar[1])\*exp(optpar[2])/4   
sampn <- length(optres) # number of residuals and of years

result <- cbind(abdat, predce, optres)  
  
knitr::kable(result, digits = 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.1: abdat 数据集及相关的最佳预测 cpue（predce）和最佳残差（optres）。The abdat data-set with the associated optimum predicted cpue (predce), and the optimum residuals (optres).   | year | catch | cpue | predce | optres | | --- | --- | --- | --- | --- | | 1985 | 1020 | 1.000 | 1.135 | 0.881 | | 1986 | 743 | 1.096 | 1.071 | 1.023 | | 1987 | 867 | 1.130 | 1.093 | 1.034 | | 1988 | 724 | 1.147 | 1.076 | 1.066 | | 1989 | 586 | 1.187 | 1.105 | 1.075 | | 1990 | 532 | 1.202 | 1.183 | 1.016 | | 1991 | 567 | 1.265 | 1.288 | 0.983 | | 1992 | 609 | 1.320 | 1.388 | 0.951 | | 1993 | 548 | 1.428 | 1.479 | 0.966 | | 1994 | 498 | 1.477 | 1.593 | 0.927 | | 1995 | 480 | 1.685 | 1.724 | 0.978 | | 1996 | 424 | 1.920 | 1.856 | 1.034 | | 1997 | 655 | 2.051 | 1.998 | 1.027 | | 1998 | 494 | 2.124 | 2.049 | 1.037 | | 1999 | 644 | 2.215 | 2.147 | 1.032 | | 2000 | 960 | 2.253 | 2.180 | 1.033 | | 2001 | 938 | 2.105 | 2.103 | 1.001 | | 2002 | 911 | 2.082 | 2.044 | 1.018 | | 2003 | 955 | 2.009 | 2.003 | 1.003 | | 2004 | 936 | 1.923 | 1.952 | 0.985 | | 2005 | 941 | 1.870 | 1.914 | 0.977 | | 2006 | 954 | 1.878 | 1.878 | 1.000 | | 2007 | 1027 | 1.850 | 1.840 | 1.005 | | 2008 | 980 | 1.727 | 1.782 | 0.969 | |

通常情况下，至少需要进行 1000 次自助采样。请注意，我们将 *bootfish* 设置为矩阵而不是数据框。如果你移除 as.matrix，使 *bootfish* 成为数据框，比较执行该操作所需的时间，并看到在计算密集型工作中使用矩阵的优势。

# 1000 bootstrap Schaefer model fits; takes a few seconds   
start <- Sys.time() # use of as.matrix faster than using data.frame   
bootfish <- as.matrix(abdat) # and avoid altering original data   
N <- 1000; years <- abdat[,"year"] # need N x years matrices   
columns <- c("r","K","Binit","sigma")   
results <- matrix(0,nrow=N,ncol=sampn,dimnames=list(1:N,years))   
bootcpue <- matrix(0,nrow=N,ncol=sampn,dimnames=list(1:N,years))   
parboot <- matrix(0,nrow=N,ncol=4,dimnames=list(1:N,columns))   
for (i in 1:N) { # fit the models and save solutions   
 bootcpue[i,] <- predce \* sample(optres, sampn, replace=TRUE)   
 bootfish[,"cpue"] <- bootcpue[i,] #calc and save bootcpue   
 bootmod <- nlm(f=negLL,p=optpar,funk=simpspm,indat=bootfish,   
 logobs=log(bootfish[,"cpue"]))   
 parboot[i,] <- exp(bootmod$estimate) #now save parameters   
 results[i,] <- exp(simpspm(bootmod$estimate,abdat)) #and predce   
}   
cat("total time = ",Sys.time()-start, "seconds \n")

total time = 5.044982 seconds

在我写这本书时使用的电脑上，bootstrap 运行大约需要 4 秒。这包括从样本中抽取 24 年的 bootstrap 样本，并将过剩生产模型拟合到样本上，每次迭代大约需要 0.004 秒。对于任何需要多次拟合模型或计算似然值的计算密集型方法，这都是值得了解的。了解分析运行所需的时间有助于设计这些分析的规模。过剩生产模型的拟合时间非常短，但如果一个复杂的年龄结构模型拟合需要大约 1 分钟，那么 1000 个复制（按现代标准来说，这是最低要求，而且更多的复制无疑会提供更精确的结果）将需要超过 16 个小时！在可能的情况下，优化代码速度仍然非常重要；当我们在本章后面讨论使用 Rcpp 包近似贝叶斯后验分布时，我们将讨论优化速度的方法。

引导样本通常可以绘制出来，以提供模型拟合质量的直观印象， [图 6.4](#fig-6-4) 。可以将quants() 函数应用于包含最优预测 cpue 引导估计的结果矩阵，从而在图的灰色边界内绘制百分位数置信区间，但在这里结果非常紧密，这样做大多只会使图形显得杂乱。

1988 年和 1989 年的数值具有最大的残差，因此永远不会被超过。

# bootstrap replicates in grey behind main plot Fig 6.4   
plot1(abdat[,"year"],abdat[,"cpue"],type="n",xlab="Year",   
 ylab="CPUE") # type="n" just lays out an empty plot   
for (i in 1:N) # ready to add the separate components   
 lines(abdat[,"year"],results[i,],lwd=1,col="grey")   
points(abdat[,"year"],abdat[,"cpue"],pch=16,cex=1.0,col=1)   
lines(abdat[,"year"],predce,lwd=2,col=1)

|  |
| --- |
| 图 6.4: 鲍鱼渔业 abdat 数据集最佳预测 cpue 的 1000 个自助估计值。黑点为原始数据，黑线为原始模型拟合的最佳预测 cpue，灰色轨迹为预测 cpue 的 1000 个自举估计值。1000 bootstrap estimates of the optimum predicted cpue from the abdat data set for an abalone fishery. Black points are the original data, the black line is the optimum predicted cpue from the original model fit, and the grey trajectories are the 1000 bootstrap estimates of the predicted cpue. |

在 1000 次重复中，每个图中仍存在一些不够清晰的部分，尤其是在后期年份，因此必须小心，不能仅仅绘制灰色轨迹的轮廓（可能使用 chull() 定义），否则预测轨迹空间中相对较窄的区域可能会被忽略。通常，进行 2000-5000 次自助法抽样可能看起来有些过度，但要避免轨迹空间中的实际空白，这样的数量可以是有利的。这类图表有帮助，但主要发现与模型参数和输出相关，例如 MSY。我们可以使用 和 的自助法估计来估计 MSY 的自助法估计，所有这些都可以绘制为直方图，并使用 quants() ，我们可以识别出我们想要的任何百分位数置信区间。 quants()默认提取 0.025、0.05、0.5、0.95 和 0.975 分位数（可以输入其他范围），允许识别出 95%和 90%的中心置信区间以及中位数。

#histograms of bootstrap parameters and model outputs Fig 6.5   
dohist <- function(invect,nmvar,bins=30,bootres,avpar) { #adhoc   
 hist(invect[,nmvar],breaks=bins,main="",xlab=nmvar,col=0)   
 abline(v=c(exp(avpar),bootres[pick,nmvar]),lwd=c(3,2,3,2),   
 col=c(3,4,4,4))   
}   
msy <- parboot[,"r"]\*parboot[,"K"]/4 #calculate bootstrap MSY   
msyB <- quants(msy) #from optimum bootstrap parameters   
parset(plots=c(2,2),cex=0.9)   
bootres <- apply(parboot,2,quants); pick <- c(2,3,4) #quantiles   
dohist(parboot,nmvar="r",bootres=bootres,avpar=optpar[1])   
dohist(parboot,nmvar="K",bootres=bootres,avpar=optpar[2])   
dohist(parboot,nmvar="Binit",bootres=bootres,avpar=optpar[3])   
hist(msy,breaks=30,main="",xlab="MSY",col=0)   
abline(v=c(optmsy,msyB[pick]),lwd=c(3,2,3,2),col=c(3,4,4,4))

|  |
| --- |
| 图 6.5: 前三个模型参数的 1000 个自举估计值和 MSY 的柱状图。在每幅图中，两条细外线定义了中位数周围的 90%置信区间，中间的垂直线表示最佳估计值，但这些估计值通常紧靠中位数下方，Binit 模型除外。The 1000 bootstrap estimates of each of the first three model parameters and MSY as histograms. In each plot the two fine outer lines define the inner 90% confidence bounds around the median, the central vertical line denotes the optimum estimates, but these are generally immediately below the medians, except for the Binit. |

再次使用仅 1000 个自助法估计，直方图并不能很好地表示所讨论的所有参数和输出值的经验分布。更多的重复将使输出平滑，并稳定分位数或百分位数的置信界限。即便如此，也可以对这类参数和输出值的生成精度有一个概念。

### 6.4.1 参数相关性

我们也可以通过使用 R 函数 pairs() 将每个参数和模型输出绘制在一起，来检查它们之间的任何相关性。 、 和 之间的强相关性立即显现出来。与 sigma 值缺乏相关性是残差内部变化应如何表现的一种典型情况。更有趣的是， （它是 和 的函数）与其他参数的相关性有所降低。这种降低反映了 和 之间的负相关性，这种负相关性会抵消彼此变化的影响。因此，对于相当不同的 和 值，我们可以有相似的 值。在 [图 6.6](#fig-6-6) 中使用 rgb() 函数来根据点的密度变化颜色强度，也可以在绘制 1000 条轨迹时使用，以识别图(6.4)中最常见的轨迹。

#relationships between parameters and MSY Fig 6.6   
parboot1 <- cbind(parboot,msy)   
 # note rgb use, alpha allows for shading, try 1/15 or 1/10   
pairs(parboot1,pch=16,col=rgb(red=1,green=0,blue=0,alpha = 1/20))

|  |
| --- |
| 图 6.6: 用 Schaefer 模型拟合 abdat 数据集时，1000 个自助法估算的最佳参数估算值与得出的 MSY 值之间的关系。全色强度由至少 20 个点得出。更多的自举重复将使这些强度图更加完整。The relationships between the 1000 bootstrap estimates of the optimum parameter estimates and the derived MSY values for the Schaefer model fitted to the abdat data-set. Full colour intensity derived from a minimum of 20 points. More bootstrap replicates would fill out these intensity plots. |

## 6.5 渐近误差

置信区间的概念（通常为 90% 或 95% CI）在 Snedecor 和 Cochran (1967, 1989) 以及许多其他文献中经典定义为：

其中 是 个观测样本（实际上，任何参数估计）的平均值， σσ 是样本标准差，是样本变异性的度量， tνtν 是具有 ν=(n−1)ν=(n−1) 个自由度的 t 分布（另见 rt() 、 dt() 、 pt() 和 qt() ）。如果我们有多个独立样本，则可以估计样本均值组的标准差。 σ/√nσ/n 被称为变量 xx 的标准误差，并且是当只有一个样本时估计样本均值集预期标准差的一种分析方法（生成多个样本的引导法是另一种方法，尽管在这种情况下，直接使用百分位数置信区间会更好）。由于我们处理的是正态分布数据，这种经典置信区间在均值或期望值周围对称分布。这在处理单个样本时是可行的，但我们要解决的问题是如何确定在将多参数模型拟合到数据时，我们可以以多大信心信任参数估计和模型输出。

在这种情况下，可以通过在最优参数集附近估计模型参数的协方差矩阵来产生渐近标准误差，如 [公式 6.8](#eq-6_8) 所示。在实践中，假设多参数模型在最优点的最大似然或平方和表面的梯度近似于多元正态分布，并可用于表征不同参数之间的关系。这些关系是生成所需协方差矩阵的基础。该矩阵用于生成每个参数的标准误差，如 [公式 6.8](#eq-6_8) 所示，然后这些标准误差可用于估计参数的近似置信区间。它们是近似的，因为这种方法假设在最优点附近的拟合表面是规则的和平滑的，并且表面在最优点附近近似于多元正态（=对称）分布。这意味着所得标准误差将在最优解周围对称，这可能适当也可能不适当。 然而，作为初步近似，渐近标准误差可以提供关于参数估计值周围固有变异的指示。

Hessian 矩阵描述了最大似然表面在最优值附近的局部曲率或梯度。更正式地说，它由描述不同参数集似然函数的二阶偏导数组成。也就是说，每个参数相对于自身和其他所有参数的变化率的变化率。所有 Hessian 矩阵都是方阵。如果我们只考虑 Schaefer 模型的四个参数中的 、 和 ，即 [公式 6.1](#eq-6_1) 中的三个参数，那么描述这三个参数的 Hessian 矩阵将是：

如果我们正在计算二阶偏导数的函数 使用对数似然来将模型拟合到数据上，那么协方差矩阵是 Hessian 矩阵的逆。然而，请注意这一点，如果函数 使用最小二乘法进行数据分析，那么形式上协方差矩阵是最佳拟合处的残差方差与 Hessian 矩阵的逆的乘积。残差方差反映了估计参数的数量：

这是观测值与预测值之间的平方偏差之和，除以观测次数（ ）减去参数个数（ ）。

因此，要么通过求 Hessian 矩阵的逆来估计方差-协方差矩阵（ ）， （在使用最大似然时），要么在使用最小二乘法时，我们将 Hessian 矩阵的逆的元素乘以残差方差：

在这里我们将重点关注最大似然方法，但了解在使用最大似然方法和最小二乘法方法时的程序差异是值得的（建议在使用渐近误差时始终使用最大似然方法）。

向量中每个参数的标准误差估计是通过取方差-协方差矩阵的对角元素（方差）的平方根获得的：

在 Excel 中，我们使用有限差分法来估计 Hessian 矩阵（Haddon，2011），但这种方法在参数强相关时并不总是表现良好。幸运的是，在 R 中，许多可用的非线性求解器在拟合模型时提供了自动生成 Hessian 矩阵估计值的选项。

#Fit Schaefer model and generate the Hessian   
data(abdat)   
param <- log(c(r= 0.42,K=9400,Binit=3400,sigma=0.05))   
 # Note inclusion of the option hessian=TRUE in nlm function   
bestmod <- nlm(f=negLL,p=param,funk=simpspm,indat=abdat,   
 logobs=log(abdat[,"cpue"]),hessian=TRUE)   
outfit(bestmod,backtran = TRUE) #try typing bestmod in console

nlm solution:   
minimum : -41.37511   
iterations : 20   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient transpar  
1 -0.9429555 6.743051e-06 0.38948  
2 9.1191569 -9.223729e-05 9128.50173  
3 8.1271026 1.059182e-04 3384.97779  
4 -3.1429030 -8.116218e-07 0.04316  
hessian :   
 [,1] [,2] [,3] [,4]  
[1,] 3542.8630966 2300.305473 447.63247 -0.3509662  
[2,] 2300.3054728 4654.008776 -2786.59928 -4.2155105  
[3,] 447.6324673 -2786.599276 3183.93947 -2.5662897  
[4,] -0.3509662 -4.215511 -2.56629 47.9905538

# Now generate the confidence intervals   
vcov <- solve(bestmod$hessian) # solve inverts matrices   
sterr <- sqrt(diag(vcov)) #diag extracts diagonal from a matrix   
optpar <- bestmod$estimate #use qt for t-distrib quantiles   
U95 <- optpar + qt(0.975,20)\*sterr # 4 parameters hence   
L95 <- optpar - qt(0.975,20)\*sterr # (24 - 4) df   
cat("\n r K Binit sigma \n")

r K Binit sigma

cat("Upper 95% ",round(exp(U95),5),"\n") # backtransform

Upper 95% 0.45025 10948.12 4063.59 0.05838

cat("Optimum ",round(exp(optpar),5),"\n")#\n =linefeed in cat

Optimum 0.38948 9128.502 3384.978 0.04316

cat("Lower 95% ",round(exp(L95),5),"\n")

Lower 95% 0.33691 7611.311 2819.693 0.0319

### 6.5.1 模型输出的不确定性

渐近标准误差也可以提供模型输出（如 MSY 估计）的近似置信区间，尽管这需要稍微不同的方法。这种方法基于一个假设，即关于最优解的对数似然曲面可以用多元正态分布来近似。这通常定义为：

其中，在本节讨论的情况下， 是最优参数估计的向量（均值的向量），而 是最优参数向量的协方差矩阵。

一旦这些输入被估计，我们可以通过从具有最优参数估计值的均值和由逆 Hessian 估计的协方差矩阵的多变量正态分布中抽样来生成随机参数向量。这些随机参数向量可以像 bootstrap 参数向量一样使用，用来生成参数和模型输出的百分位数置信区间。使用多变量正态分布（有些人写作 multivariate normal），参数之间的相关性会自动被考虑在内。

### 6.5.2 从多元正态分布中取样

基础 R 没有用于处理多元正态分布的函数，但可以使用包含合适函数的几个 R 包。**MASS** 包（Venables 和 Ripley, 2002）包含一个合适的随机数生成器，而 **mvtnorm** 包则拥有更广泛的多元概率密度函数。这里我们将使用 **mvtnorm**。

# Use multi-variate normal to generate percentile CI Fig 6.7   
library(mvtnorm) # use RStudio, or install.packages("mvtnorm")   
N <- 1000 # number of multi-variate normal parameter vectors   
years <- abdat[,"year"]; sampn <- length(years) # 24 years   
mvncpue <- matrix(0,nrow=N,ncol=sampn,dimnames=list(1:N,years))   
columns <- c("r","K","Binit","sigma")   
 # Fill parameter vectors with N vectors from rmvnorm   
mvnpar <- matrix(exp(rmvnorm(N,mean=optpar,sigma=vcov)),   
 nrow=N,ncol=4,dimnames=list(1:N,columns))   
 # Calculate N cpue trajectories using simpspm   
for (i in 1:N) mvncpue[i,] <- exp(simpspm(log(mvnpar[i,]),abdat))   
msy <- mvnpar[,"r"]\*mvnpar[,"K"]/4 #N MSY estimates   
 # plot data and trajectories from the N parameter vectors   
plot1(abdat[,"year"],abdat[,"cpue"],type="p",xlab="Year",   
 ylab="CPUE",cex=0.9)   
for (i in 1:N) lines(abdat[,"year"],mvncpue[i,],col="grey",lwd=1)   
points(abdat[,"year"],abdat[,"cpue"],pch=16,cex=1.0)#orig data   
lines(abdat[,"year"],exp(simpspm(optpar,abdat)),lwd=2,col=1)

|  |
| --- |
| 图 6.7: 从最优参数及其相关方差-协方差矩阵定义的多变量正态分布中采样的随机参数向量得出的 1000 条 cpue 预测轨迹。The 1000 predicted cpue trajectories derived from random parameter vectors sampled from the multi-variate Normal distribution defined by the optimum parameters and their related variance-covariance matrix. |

和引导法示例一样，即使是 1000 个轨迹的样本也不足以完全填充轨迹空间，导致某些区域比其他区域稀疏。更多的重复实验可能会填补这些空白， [图 6.7](#fig-6-7) 。在引导法 cpue 线中使用 rgb() 也有助于识别稀疏区域。

我们也可以使用 pairs() 绘制隐含参数相关性（如果有），就像我们对自助样本所做的那样， [图 6.8](#fig-6-8) 。然而，在这里，相对于自助过程的结果， [图 6.6](#fig-6-6) ，结果似乎相对均匀且清晰分布。更平滑的分布反映了这些值都是从一个明确定义的概率密度函数中抽取的，而不是经验分布。有理由认为，自助过程将预期提供对数据特性的更准确表示。然而，哪个结果集更好地代表原始样本来源的总体却不太容易有信心地回答。真正重要的是它们的汇总统计数据是否不同，考虑到 [表 6.2](#tbl-6-2) ，我们可以看到，虽然每个参数和模型输出的分布细节略有不同，但似乎没有一个单一的规律表明其中一个比另一个更宽或更窄，且这种模式是一致的。

#correlations between parameters when using mvtnorm Fig 6.8   
  
pairs(cbind(mvnpar,msy),pch=16,col=rgb(red=1,0,0,alpha = 1/10))

|  |
| --- |
| 图 6.8: 从估计的多变量正态分布中取样的 1000 个参数估计值（假定围绕最佳参数估计值）与 Schaefer 模型拟合的丰年鱼量值之间的关系。The relationships between the 1000 parameter estimates sampled from the estimated multi-variate Normal distribution assumed to surround the optimum parameter estimates and the derived MSY values for the Schaefer model fitted to the abdat data set. |

最后，我们可以通过绘制参数和 MSY 的直方图数组及其预测的置信界限来展示渐近置信区间，如 [图 6.9](#fig-6-9) 所示。在这种情况下，我们使用内部 90%界限。均值和中位数比自助法示例更加接近，这进一步反映了使用多元正态分布的对称性。

#N parameter vectors from the multivariate normal Fig 6.9   
mvnres <- apply(mvnpar,2,quants) # table of quantiles   
pick <- c(2,3,4) # select rows for 5%, 50%, and 95%   
meanmsy <- mean(msy) # optimum bootstrap parameters   
msymvn <- quants(msy) # msy from mult-variate normal estimates   
   
plothist <- function(x,optp,label,resmvn) {   
 hist(x,breaks=30,main="",xlab=label,col=0)   
 abline(v=c(exp(optp),resmvn),lwd=c(3,2,3,2),col=c(3,4,4,4))   
} # repeated 4 times, so worthwhile writing a short function   
par(mfrow=c(2,2),mai=c(0.45,0.45,0.05,0.05),oma=c(0.0,0,0.0,0.0))   
par(cex=0.85, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
plothist(mvnpar[,"r"],optpar[1],"r",mvnres[pick,"r"])   
plothist(mvnpar[,"K"],optpar[2],"K",mvnres[pick,"K"])   
plothist(mvnpar[,"Binit"],optpar[3],"Binit",mvnres[pick,"Binit"])   
plothist(msy,meanmsy,"MSY",msymvn[pick])

|  |
| --- |
| 图 6.9: 从最优解估算的多变量正态中得出的 r、K、Binit 和推导的 MSY 的 1000 个参数估计直方图。在每幅图中，绿线表示算术平均值，蓝色粗线表示中位数，两条蓝色细线表示中位数周围 90%的置信区间。Histograms of the 1000 parameter estimates for r, K, Binit, and the derived MSY, from the multi-variate normal estimated at the optimum solution. In each plot, the green line denotes the arithmetic mean, the thick blue line the median, and the two fine blue lines the inner 90% confidence bounds around the median. |

knitr::kable(bootres, digits = 3)  
knitr::kable(mvnres, digits = 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.2: 自助百分位数参数置信区间与方差-协方差矩阵渐近估计值参数置信区间的比较。上表是自举法结果，下表是多元正态分布结果。A comparison of the bootstrap percentile confidence bounds on parameters with those derived from the Asymptotic estimate of the variance-covariance matrix. The top table relates to the bootstrapping and the bottom to the values from the multi-variate normal.   |  | r | K | Binit | sigma | | --- | --- | --- | --- | --- | | 2.5% | 0.335 | 7740.636 | 2893.714 | 0.025 | | 5% | 0.345 | 8010.341 | 2970.524 | 0.026 | | 50% | 0.390 | 9116.193 | 3387.077 | 0.039 | | 95% | 0.435 | 10507.708 | 3889.824 | 0.050 | | 97.5% | 0.445 | 11003.649 | 4055.071 | 0.052 |  |  | r | K | Binit | sigma | | --- | --- | --- | --- | --- | | 2.5% | 0.339 | 7717.875 | 2882.706 | 0.033 | | 5% | 0.347 | 7943.439 | 2945.264 | 0.034 | | 50% | 0.389 | 9145.146 | 3389.279 | 0.043 | | 95% | 0.437 | 10521.651 | 3900.034 | 0.055 | | 97.5% | 0.444 | 10879.571 | 4031.540 | 0.058 | |

## 6.6 似然剖面

“似然剖面”（Likelihood profile）这个名字暗示了生成这些分析所使用的过程。在本章中，我们已经最优地拟合了一个四参数模型，并探讨了如何表征这些参数估计值周围的不确定性。可以想象从这四个参数中选择一个，并将其值固定在远离其最优值的位置。如果随后将所选参数固定，而允许其他参数变化，重新拟合模型，我们可以想象剩余参数会找到一个新的最优解，但负对数似然值会大于当所有四个参数都自由变化时获得的最优值。这个过程是生成似然曲线的基本思想。

目的是使用最大似然方法（最小化负对数似然）来拟合模型，但仅拟合特定参数，同时将其他参数保持在最优值周围的常数值。通过这种方式，可以在给定参数具有一系列固定值的情况下获得新的“最优”拟合。因此，我们可以确定固定参数对模型拟合总似然的影响。一个例子可以帮助使这个过程更加清晰。

再次使用 Schaefer 剩余生产模型对 abdat 渔业数据进行分析，我们得到最优参数 r = 0.3895，K = 9128.5，Binit = 3384.978，以及 sigma = 0.04316，这意味着 MSY = 888.842 吨。

#Fit the Schaefer surplus production model to abdat   
data(abdat); logce <- log(abdat$cpue) # using negLL   
param <- log(c(r= 0.42,K=9400,Binit=3400,sigma=0.05))   
optmod <- nlm(f=negLL,p=param,funk=simpspm,indat=abdat,logobs=logce)   
outfit(optmod,parnames=c("r","K","Binit","sigma"))

nlm solution:   
minimum : -41.37511   
iterations : 20   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient transpar  
r -0.9429555 6.743051e-06 0.38948  
K 9.1191569 -9.223729e-05 9128.50173  
Binit 8.1271026 1.059182e-04 3384.97779  
sigma -3.1429030 -8.116218e-07 0.04316

如何调整有限数量的模型参数，同时保持其余参数不变的问题，通过修改用于最小化负对数似然的功能得以解决。我们不是使用 negLL() 函数来计算对数正态分布的 cpue 值的负对数似然，而是使用了 **MQMF** 函数 negLLP() （负对数似然曲线）。这增加了固定某些参数的能力，同时通过仅改变非固定参数来求解最优解。如果我们查看 negLLP() 函数的 R 代码，并将其与 negLL() 函数进行比较，我们可以看到除了参数之外，重要区别在于 *logpred* 语句之前的三个代码行。请参阅帮助页面（?）以获取更多详细信息，尽管我希望你能立刻看出，如果你忽略 *initpar* 和 *notfixed* 参数， negLLP() 应该会给出与 negLL() 相同的结果。

#the code for MQMF's negLLP function   
negLLP <- function(pars, funk, indat, logobs, initpar=pars,   
 notfixed=c(1:length(pars)),...) {   
 usepar <- initpar #copy the original parameters into usepar   
 usepar[notfixed] <- pars[notfixed] #change 'notfixed' values   
 npar <- length(usepar)   
 logpred <- funk(usepar,indat,...) #funk uses the usepar values   
 pick <- which(is.na(logobs)) # proceed as in negLL   
 if (length(pick) > 0) {   
 LL <- -sum(dnorm(logobs[-pick],logpred[-pick],exp(pars[npar]),   
 log=T))   
 } else {   
 LL <- -sum(dnorm(logobs,logpred,exp(pars[npar]),log=T))   
 }   
 return(LL)   
} # end of negLLP

例如，为了确定 参数的估计精度，我们可以强制它取 0.3 到 0.45 之间的常数值，同时将其他参数拟合以获得最佳拟合效果，然后绘制总似然与给定 值的函数关系图。与 R 中所有模型拟合一样，我们需要两个函数：一个用于生成所需的预测值，另一个作为包装器将观测值和预测值结合起来供最小化器使用，在此情况下为 nlm()。为了继续进行，我们使用 negLLP() 来允许某些参数保持固定值。 nlm() 通过迭代修改输入参数，朝着改善模型拟合的方向进行调整，然后将这些参数反馈到生成预测值的输入模型函数中，与观测值进行比较。因此，我们需要将模型函数安排为不断返回我们希望固定为初始设定值的特定参数值。 negLLP() 中的代码是完成这一操作的一种方法。 所需的更改包括在 *initpar* 中有一个独立的原始 pars 集合，该集合必须包含指定的固定参数，以及一个 *notfixed* 参数，用于识别从 *initpar* 中哪些值将被 *pars* 中的值覆盖，这些值将随后被 nlm 修改。

最佳实践是检查 negLLP() 产生的结果与使用 negLL() 相同，尽管代码检查让我们确信它会如此（我不再惊讶于自己会犯错），通过允许所有参数变化（默认设置）。

#does negLLP give same answers as negLL when no parameters fixed?   
param <- log(c(r= 0.42,K=9400,Binit=3400,sigma=0.05))   
bestmod <- nlm(f=negLLP,p=param,funk=simpspm,indat=abdat,logobs=logce)   
outfit(bestmod,parnames=c("r","K","Binit","sigma"))

nlm solution:   
minimum : -41.37511   
iterations : 20   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient transpar  
r -0.9429555 6.743051e-06 0.38948  
K 9.1191569 -9.223729e-05 9128.50173  
Binit 8.1271026 1.059182e-04 3384.97779  
sigma -3.1429030 -8.116218e-07 0.04316

令人高兴的是，正如预期的那样，我们得到了相同的解，因此现在我们可以继续研究固定 的值并重新拟合模型的影响。通过后见之明（这比现实情况要好得多），我们选择了介于 0.325 和 0.45 之间的 值。我们可以设置一个循环来依次应用这些值并拟合相应的模型，将解保存在循环进行的过程中。下面我们列出了前几个结果， [表 6.3](#tbl-6-3) ，以说明随着固定值 越来越远离其最优值，负对数似然如何增加。

#Likelihood profile for r values 0.325 to 0.45   
rval <- seq(0.325,0.45,0.001) # set up the test sequence   
ntrial <- length(rval) # create storage for the results   
columns <- c("r","K","Binit","sigma","-veLL")   
result <- matrix(0,nrow=ntrial,ncol=length(columns),   
 dimnames=list(rval,columns))# close to optimum   
bestest <- c(r= 0.32,K=11000,Binit=4000,sigma=0.05)   
for (i in 1:ntrial) { #i <- 1   
 param <- log(c(rval[i],bestest[2:4])) #recycle bestest values   
 parinit <- param #to improve the stability of nlm as r changes   
 bestmodP <- nlm(f=negLLP,p=param,funk=simpspm,initpar=parinit,   
 indat=abdat,logobs=log(abdat$cpue),notfixed=c(2:4),   
 typsize=magnitude(param),iterlim=1000)  
 bestest <- exp(bestmodP$estimate)   
 result[i,] <- c(bestest,bestmodP$minimum) # store each result   
}   
minLL <- min(result[,"-veLL"]) #minimum across r values used.

knitr::kable(result[1:12,], digits = 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.3: nlm 解决方案 126 行中的前 12 条记录用于绘制 r 的似然曲线。The first 12 records from the 126 rows of the nlm solutions that are used to make the likelihood profile on r. The strong correlation between r, K, and Binit is, once again, apparent.   |  | r | K | Binit | sigma | -veLL | | --- | --- | --- | --- | --- | --- | | 0.325 | 0.325 | 11449.17 | 4240.797 | 0.048 | -38.618 | | 0.326 | 0.326 | 11403.51 | 4223.866 | 0.048 | -38.696 | | 0.327 | 0.327 | 11358.24 | 4207.082 | 0.048 | -38.772 | | 0.328 | 0.328 | 11313.35 | 4190.442 | 0.048 | -38.848 | | 0.329 | 0.329 | 11268.83 | 4173.945 | 0.048 | -38.922 | | 0.33 | 0.330 | 11224.69 | 4157.589 | 0.048 | -38.996 | | 0.331 | 0.331 | 11180.91 | 4141.373 | 0.048 | -39.070 | | 0.332 | 0.332 | 11137.49 | 4125.293 | 0.047 | -39.142 | | 0.333 | 0.333 | 11094.43 | 4109.350 | 0.047 | -39.213 | | 0.334 | 0.334 | 11051.72 | 4093.540 | 0.047 | -39.284 | | 0.335 | 0.335 | 11009.11 | 4077.752 | 0.047 | -39.354 | | 0.336 | 0.336 | 10967.34 | 4062.316 | 0.047 | -39.422 | |

### 6.6.1 基于似然比的置信区间

Venzon 和 Moolgavkar（1988）描述了一种获取他们称之为“近似似然比置信区间”的方法，该方法基于对通常用于似然比检验方法的重新排列。该方法依赖于这样一个事实：随着样本量的增大，似然比检验渐近地趋近于 分布，因此这种方法只是近似的。毫不奇怪，似然比检验基于两个似然值的比率，或者如果处理的是对数似然值，则是从一个中减去另一个：

其中 是 参数的似然值， 下标表示最大似然（假设所有其他参数也进行了最佳拟合）， 是等效的对数似然值。对于单个参数的实际置信区间，假设其他参数保持最优（如在似然轮廓中），预期的对数似然值由以下公式给出（Venzon 和 Moolgavkar，1988）：

其中 是 分布的第 分位数，自由度为1（例如，对于 95% 置信区间， 和 ，以及 。

对于单个参数 ，其95%的近似置信区间的求解可表述为：找到所有满足以下条件的 值，该参数对应的对数似然与全局最优对数似然之差的 2 倍小于或等于 3.84（ ），或者（ [公式 6.15](#eq-6_15) 的最后一行），也可以通过搜索满足对数似然值等于最大对数似然减去所需 值一半的 （即，当自由度为 1 时，计算 ）。若构建双参数似然曲面时，则需在 值设定为 5.99（自由度为2）的条件下搜索 ，此时应从最大似然值中减去 2.995（对于更高的自由度以此类推）。

我们可以绘制 参数的似然剖面，并包含这些近似 95% 置信区间。检查函数 plotprofile() 中的代码，以了解从分析结果计算置信区间的步骤。

#likelihood profile on r from the Schaefer model Fig 6.10   
plotprofile(result,var="r",lwd=2) # review the code

|  |
| --- |
| 图 6.10: Schaefer 剩余产量模型对 abdat 数据集拟合得到的 r 参数的似然曲线。水平实线是最小值和最小值减去 1.92（95% 水平，自由度为 1，见正文）。外部的垂直线是围绕中心均值 0.389 的近似 95% 置信区间。 |

我们可以对 Schaefer 模型中的其他参数重复此分析。例如，我们可以使用几乎相同的代码以类似的方式对 参数进行类似分析。请注意， 参数的剖面图中呈现出更明显的不对称性。虽然 参数的似然剖面也存在轻微不对称（从均值估计中减去 95%置信区间），但对于 参数，这种不对称性肉眼可见。 的最优参数估计为 9128.5，但剖面数据中的最大似然值指向 9130。这仅仅是似然剖面步长的反映。当前它设置为 10，如果设置为 1，我们可以得到更接近的近似值，当然，分析运行时间会延长 10 倍。

#Likelihood profile for K values 7200 to 12000   
Kval <- seq(7200,12000,10)   
ntrial <- length(Kval)   
columns <- c("r","K","Binit","sigma","-veLL")   
resultK <- matrix(0,nrow=ntrial,ncol=length(columns),   
 dimnames=list(Kval,columns))   
bestest <- c(r= 0.45,K=7500,Binit=2800,sigma=0.05)   
for (i in 1:ntrial) {   
 param <- log(c(bestest[1],Kval[i],bestest[c(3,4)]))   
 parinit <- param   
 bestmodP <- nlm(f=negLLP,p=param,funk=simpspm,initpar=parinit,   
 indat=abdat,logobs=log(abdat$cpue),   
 notfixed=c(1,3,4),iterlim=1000)   
 bestest <- exp(bestmodP$estimate)   
 resultK[i,] <- c(bestest,bestmodP$minimum)   
}   
minLLK <- min(resultK[,"-veLL"])   
 #kable(head(result,12),digits=c(4,3,3,4,5)) # if wanted.

#likelihood profile on K from the Schaefer model Fig 6.11   
plotprofile(resultK,var="K",lwd=2)

|  |
| --- |
| 图 6.11: Schaefer 剩余产量模型对 abdat 数据集的 K 参数的似然曲线，与 r 参数的处理方式相同。红线是最小值和最小值加 1.92（卡方分布 1 自由度的 95% 水平，见正文）。垂直粗线是围绕 9128.5 均值的近似 95% 置信区间。 |

### 6.6.2 负对数似然或似然

我们已经绘制了负对数似然图，例如 [图 6.11](#fig-6-11) ，在以对数空间操作时这些图是合适的，但很少有人对对数空间有清晰的理解。另一种方法，可能更容易理解百分位数置信区间背后的原理，是将负对数似然转换回简单的似然值。当然，在对数值 -41 转换回原始值时表示一个非常小的数，但我们可以在确保所有似然值之和为 1.0 的过程中重新调整这些值。为了将负对数似然转换回原始值，我们需要先取其相反数，然后进行指数运算。

因此，对于应用于 参数的似然曲线，我们可以看到当 的值接近最优值时，线性空间的似然值开始增加，如果我们绘制这些似然值，分布的形状可以更直观地理解。尾部保持在零以上，但远离 95%置信区间：

#translate -velog-likelihoods into likelihoods   
likes <- exp(-resultK[,"-veLL"])/sum(exp(-resultK[,"-veLL"]),   
 na.rm=TRUE)   
resK <- cbind(resultK,likes,cumlike=cumsum(likes))

knitr::kable(resK[1:8,], digits = 5)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.4: 用于制作 K 的似然轮廓的 nlm 解的前 8 条记录，共481行。包括反转换后的负对数似然值（缩放到总和为 1.0）及其运行累积和。   |  | r | K | Binit | sigma | -veLL | likes | cumlike | | --- | --- | --- | --- | --- | --- | --- | --- | | 7200 | 0.47314 | 7200 | 2689.875 | 0.05158 | -37.09799 | 7e-05 | 0.00007 | | 7210 | 0.47257 | 7210 | 2693.444 | 0.05147 | -37.14518 | 7e-05 | 0.00014 | | 7220 | 0.47201 | 7220 | 2697.023 | 0.05137 | -37.19213 | 8e-05 | 0.00022 | | 7230 | 0.47145 | 7230 | 2700.602 | 0.05127 | -37.23881 | 8e-05 | 0.00030 | | 7240 | 0.47089 | 7240 | 2704.182 | 0.05118 | -37.28524 | 8e-05 | 0.00038 | | 7250 | 0.47033 | 7250 | 2707.762 | 0.05108 | -37.33141 | 9e-05 | 0.00047 | | 7260 | 0.46977 | 7260 | 2711.341 | 0.05098 | -37.37732 | 9e-05 | 0.00056 | | 7270 | 0.46922 | 7270 | 2714.933 | 0.05088 | -37.42298 | 1e-04 | 0.00065 | |

#K parameter likelihood profile Fig 6.12   
   
oldp <- plot1(resK[,"K"],resK[,"likes"],xlab="K value",   
 ylab="Likelihood",lwd=2)   
lower <- which.closest(0.025,resK[,"cumlike"])   
mid <- which(resK[,"likes"] == max(resK[,"likes"]))   
upper <- which.closest(0.975,resK[,"cumlike"])   
abline(v=c(resK[c(lower,mid,upper),"K"]),col=1,lwd=c(1,2,1))   
label <- makelabel("",resK[c(lower,mid,upper),"K"],sep=" ")   
text(9500,0.005,label,cex=1.2,pos=4)   
par(oldp) # return par to old settings; this line not in book

|  |
| --- |
| 图 6.12: Schaefer 剩余产量模型对 abdat 数据集拟合的 K 参数似然曲线。在这种情况下，负对数似然值已被反转为似然值并缩放到总和为 1.0。垂直线是围绕均值的近似 95% 置信界限。最上面的三个数字是界限和估计的最优值。 |

### 6.6.3 模型输出中的分位数似然剖面

通常，对种群评估模型兴趣的关联在于那些并非直接作为参数估计的模型输出。如我们所见，在参数估计周围生成置信区间相对直接，但如何为模型输出（如 MSY（对于 Schaefer 模型 ））提供类似的关于不确定性的估计呢？例如，一个评估模型可能估计种群生物量，或最大持续产量，或某种其他可视为模型间接输出的性能指标。通过在负对数似然中添加一个惩罚项来产生此类模型输出的似然曲线，该惩罚项试图将似然约束到最优目标（参见 [公式 6.17](#eq-6_17) ）。通过这种方式，偏离最优值的对数似然的影响可以被表征。

其中 是负对数似然，output 是目标变量（在接下来的示例中，这是 MSY），target 是该变量的最优值（来自整体最优解的 MSY），而 ww 是一个权重因子。权重因子应尽可能大，以生成最窄的似然曲线，同时仍能收敛到解。下面，我们描述一个专门用于处理模型输出周围似然曲线的函数 negLLO() （这不在 **MQMF** 中）。它只是一个修改版的 negLL() 函数，允许引入 [公式 6.17](#eq-6_17) 中描述的权重因子。我们可以通过检查 887.729 吨最优 MSY 值周围的似然曲线来举例

说明这一点，检查范围可以是 740 至 1050 吨，权重为 900。

#examine effect on -veLL of MSY values from 740 - 1050t   
 #need a different negLLP() function, negLLO(): O for output.   
 #now optvar=888.831 (rK/4), the optimum MSY, varval ranges 740-1050   
 #and wght is the weighting to give to the penalty   
negLLO <- function(pars,funk,indat,logobs,wght,optvar,varval) {   
 logpred <- funk(pars,indat)   
 LL <- -sum(dnorm(logobs,logpred,exp(tail(pars,1)),log=T)) +   
 wght\*((varval - optvar)/optvar)^2 #compare with negLL   
 return(LL)   
} # end of negLLO   
msyP <- seq(740,1020,2.5);   
optmsy <- exp(optmod$estimate[1])\*exp(optmod$estimate[2])/4   
ntrial <- length(msyP)   
wait <- 400   
columns <- c("r","K","Binit","sigma","-veLL","MSY","pen",   
 "TrialMSY")   
resultO <- matrix(0,nrow=ntrial,ncol=length(columns),   
 dimnames=list(msyP,columns))   
bestest <- c(r= 0.47,K=7300,Binit=2700,sigma=0.05)   
for (i in 1:ntrial) { # i <- 1   
 param <- log(bestest)   
 bestmodO <- nlm(f=negLLO,p=param,funk=simpspm,indat=abdat,   
 logobs=log(abdat$cpue),wght=wait,   
 optvar=optmsy,varval=msyP[i],iterlim=1000)   
 bestest <- exp(bestmodO$estimate)   
 ans <- c(bestest,bestmodO$minimum,bestest[1]\*bestest[2]/4,   
 wait \*((msyP[i] - optmsy)/optmsy)^2,msyP[i])   
 resultO[i,] <- ans   
}   
minLLO <- min(resultO[,"-veLL"])

#tabulate first and last few records of profile on MSY   
   
kable(head(resultO[,1:7],4),digits=c(3,3,3,4,2,3,2))   
kable(tail(resultO[,1:7],4),digits=c(3,3,3,4,2,3,2))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.5: 用于制作 MSY 似然曲线的 nlm 解中前 113 行的前 7 条和最后 7 条记录（可以更多）。行名是试验的 MSY 值，pen 是惩罚值。   |  | r | K | Binit | sigma | -veLL | MSY | pen | | --- | --- | --- | --- | --- | --- | --- | --- | | 740 | 0.389 | 9130.914 | 3385.871 | 0.0432 | -30.16 | 888.883 | 11.22 | | 742.5 | 0.389 | 9130.911 | 3385.872 | 0.0432 | -30.53 | 888.883 | 10.84 | | 745 | 0.389 | 9130.911 | 3385.872 | 0.0432 | -30.90 | 888.883 | 10.47 | | 747.5 | 0.389 | 9130.911 | 3385.872 | 0.0432 | -31.26 | 888.883 | 10.11 |  |  | r | K | Binit | sigma | -veLL | MSY | pen | | --- | --- | --- | --- | --- | --- | --- | --- | | 1012.5 | 0.389 | 9130.911 | 3385.872 | 0.0432 | -33.63 | 888.883 | 7.74 | | 1015 | 0.389 | 9130.911 | 3385.872 | 0.0432 | -33.32 | 888.883 | 8.06 | | 1017.5 | 0.389 | 9130.911 | 3385.872 | 0.0432 | -32.99 | 888.883 | 8.38 | | 1020 | 0.389 | 9130.911 | 3385.872 | 0.0432 | -32.66 | 888.883 | 8.71 | |

#likelihood profile on MSY from the Schaefer model Fig 6.13   
   
plotprofile(resultO,var="TrialMSY",lwd=2)

|  |
| --- |
| 图 6.13: 用 Schaefer 剩余生产模型拟合 abdat 数据集得到的 MSY 的似然曲线。水平红色线是最小值和最小值加 1.92（卡方分布 1 自由度的 95% 水平，见正文）。垂直线是围绕 887.729 吨均值的近似 95% 置信区间。 |

不幸的是，确定我们称之为 negLLO() 的惩罚项中的最佳权重，只能通过经验（试错法）来完成。我建议使用 900 的权重，因为我已经发现在这个水平上 95%置信区间（CI）趋于稳定。但那需要我从 100 尝试到 700，以 100 为步长，然后再以 50 为步长来发现这一点。你应该尝试使用 500、700、800、900 和 950 的权重值，以观察它们收敛到稳定值的过程。关于允许稳定解的最大权重的建议仍然是一点模糊的指导。这使得这种方法在可重复应用方面可能最为棘手。如果你尝试使用 wght = 400 来分析上述内容，那么 95% CI 将变为 825 - 950，而不是 847.5 - 927.5。在这种情况下，没有确定性解，因此它变成了尝试不同的权重并寻找一个可重复且一致的解的问题。

使用此似然轮廓方法获得的 95%置信区间（CI）与使用自助法（856.7 - 927.5）和使用渐近误差（849.7 - 927.4）获得的置信区间相当。每种方法根据所选样本量可能会产生略有不同的结果。然而，这些方法之间的一致性表明它们各自都能合理地描述该模型与这些数据组合中固有的变异。

## 6.7 贝叶斯后验分布

用模型参数拟合观测数据可以类比为在模型和给定数据集所隐含的多维似然表面上寻找最大似然的位置。如果似然表面在最优附近非常陡峭，那么与这些参数相关的不确定性就会相对较小。相反，如果似然表面在最优附近相对平坦，这意味着可以从相当不同的参数值中获得相似的似然，因此这些参数及其相关模型输出的不确定性预计会相对较高。如果存在强参数相关性，也可以提出类似的论点。我们忽略了在更复杂的模型中处理高度多维参数空间的复杂性，因为陡峭性和表面的几何概念仍然适用于多维似然。

如我们所见，如果我们假设在最优值附近似然面是多元正态分布的，那么我们可以使用渐近标准误差来定义参数估计值的置信区间。然而，对于渔业中的许多变量和模型输出，这可能是一个非常强的假设。例如，Schaefer 参数的估计似然面相对偏斜，如 [图 6.12](#fig-6-12) 所示。理想情况下，我们应该使用能够独立于任何预先设定的概率密度函数来表征最优解附近似然面的方法。如果我们能够做到这一点，我们就可以使用百分位数方法的等效方法来提供参数和模型输出置信区间的估计。

通过形式化的似然剖面分析，我们可以在参数空间中进行搜索，以构建二维似然曲面。然而，对于超过两个参数的形式似然剖面分析，或许使用这样的网格搜索会非常笨拙，并且随着参数数量的进一步增加，会越来越不切实际。我们需要的是一种能够同时整合多个维度以生成类似多维似然剖面的方法。事实上，有几种方法可以实现这一点，包括抽样重要性采样（SIR；参见 McAllister 和 Ianelli，1997；以及 McAllister 等，1994），以及马尔可夫链蒙特卡罗（MCMC；参见 Gelman 等，2013）。在接下来的示例中，我们将实现 MCMC 方法。有许多替代算法可以用于执行 MCMC，但我们将专注于一种相对灵活的方法，称为 Gibbs-within-Metropolis 采样器（有时也称为 Metropolis-Hastings-within-Gibbs）。Metropolis 算法（Metropolis 等，1953 ）最初是从二维积分开始的，后来由 Hastings（1970 年）推广，因此称为 Metropolis-Hastings。 在文献中，你会找到关于马尔可夫链蒙特卡洛和蒙特卡洛马尔可夫链的提及。前者是渔业标准参考（Gelman 等，2013）中使用的，尽管我个人认为用蒙特卡洛方法生成马尔可夫链的想法更直观明显。尽管存在这种潜在的混淆，我建议在写作时坚持使用 MCMC，如果必要的话，可以忽略我的直觉，使用马尔可夫链蒙特卡洛。这些事情并不重要到需要花时间担心，除了有时这些差异确实有意义（这种混淆仍然是一个问题，但只要你知道这些陷阱，就可以避免它们！）。

显然，MCMC（马尔可夫链蒙特卡洛方法）使用马尔可夫链来遍历多维似然曲面。马尔可夫链描述了一个过程，其中每个状态都是根据前一个状态以概率方式确定的。随机游走可以构成马尔可夫链的一种形式，其最终状态将是对随机分布的描述。然而，这里的目的是生成一个马尔可夫链，其最终平衡状态，即所谓的平稳分布，能够提供贝叶斯统计中目标或后验分布的描述。

马尔可夫链以某些参数值组合、可用的观测数据和所使用的模型为起点，这些共同定义了似然空间中的一个位置。根据参数值的不同，似然显然可以是小也可以是大（以模型的最大似然为上限）。MCMC 过程涉及根据每个新候选参数集相对于前一集的似然，遵循一套规则在参数空间中迭代地逐步前进，以确定哪些步骤将成为马尔可夫链的一部分。每次迭代中做出的决策是哪些参数向量将被接受，哪些被拒绝？该过程每一步都涉及生成一个新的候选参数值集，这是通过随机方式完成的（因此称为马尔可夫链蒙特卡罗），在 Gibbs-within-Metropolis 中，一次一个参数（Casella 和 George，1992；Millar 和 Meyer，2000）。这些新的候选参数集，结合可用的数据和模型，定义了新的似然。 这个新的参数组合是否被接受为马尔可夫链的下一步，取决于似然值的变化程度。在所有似然值增加的情况下，这一步会被接受，这似乎是合理的。现在来到关键部分，当似然值减少时，如果新似然值与旧似然值的比率大于某个均匀随机数（介于 0 和 1 之间；见下方方程式），这一步仍然可以被接受。

存在另一个与参数相关性和蒙特卡洛过程可能导致连续参数集自相关相关约束的问题。如果参数集之间存在显著的序列相关性，这可能会对参数空间中变化的全面程度产生有偏见的结论。所采用的解决方案是对结果链进行稀疏化处理，以便最终马尔可夫链中只包含每个 点。关键在于选择这种稀疏化率，使得 参数向量之间的序列相关性足够小，以至于不再影响总体方差。在实践示例中，我们将探讨这种稀疏化率。

### 6.7.1 生成马尔可夫链

如果给定一组数据 ，可以定义特定模型参数集 的似然性，并且已知参数集的贝叶斯先验概率 ，那么就可以生成一个马尔可夫链：

我们可以通过随机增加 中的至少一个参数来生成一个新的试验或候选参数集 ，这将改变隐含的似然性：

如果 和 的比值大于 1，那么从 到 的跃迁就会被接受到马尔可夫链中（ ）。

或者，如果比率 ( ) 小于 1，那么只有当比率 大于一个新选择的均匀随机数时，跳跃才会被接受。如果未被接受，则 恢复为 ，并开始一个新的候选周期：

事实上，因为 0 和 1 之间的最大均匀随机数是 1，所以 [公式 6.20](#eq-6_20) 并非严格必要。我们可以直接估计比率并使用 [公式 6.21](#eq-6_21) （这就是函数 do\_MCMC() 的实现方式）。如果候选参数向量被拒绝，它会恢复为原始值，并使用新的候选参数集重新开始周期。随着马尔可夫链的发展，它应该在参数空间中描绘出一个多维体积。经过足够多的迭代后，它应该收敛到平稳分布。在对这些理论细节进行一些扩展后，我们将通过一些实例来阐述所有这些概念。

### 6.7.2 起始点

开始马尔可夫链只需选择一个参数向量。一个解决方案是选择一个接近但并不完全等同于最大似然最优解的向量。Gelman 和 Rubin（1992）建议从一个类似于预期分布的超分散分布中抽取样本，但这假设你已经有了一个关于应该从哪个超分散分布开始的概念，而选择这一点仍然有点像是一种艺术形式（Racine-Poon，1992）。正如 Racine-Poon（1992）进一步指出的：“然而，目标分布通常是多峰的，特别是在高维问题中，我对自动化这个过程不太自信。”但是，所使用的起始点可能会对结果产生影响，因此 Gelman 和 Rubin（1992）建议从非常不同的起始点开始多个链（而不是仅仅一个很长的链；Geyer，1992），然后确保它们都收敛到相同的最终分布。 所有这些建议都源于 20 世纪 90 年代初，当时计算机的速度远慢于现在，因此，在多条链或多条非常长的链上进行的讨论（例如，参见《统计科学》第 7 卷第 4 期的大部分内容）已不再是问题，也没有真正理由不运行多条非常长的链（尽管使用最高效的软件仍然是有道理的，因为 MCMCs 通常需要大量时间投入）。

### 6.7.3 预烧期

与从后验分布的多维模式可能相距较远的点开始马尔可夫链的一个重要点是，生成的第一个点序列预计会以随机方式向更高似然区域移动。然而，这些早期点可能会给本应是最终可接受参数向量云添加笨拙和不适当的尾部。因此，标准做法是简单地删除前 个点，其中 被称为燃烧期。Gelman 等（2013 ）建议将每个链的前半部分（50%）作为燃烧期删除，但在他们讨论的是只有几百步长的链，所以在渔业中处理更高维的问题时，我们不必如此激进。几百个早期点甚至更少通常就足够了，特别是如果起始点离最大似然估计不远的话。但初步探索可能生成的链类型应该能够为每个问题找到一个合理的燃烧期。

### 6.7.4 收敛至稳定分布

20 世纪 90 年代关于应该有多少条链以及它们应该有多长的讨论，是由需要证明生成的马尔可夫链已经收敛到一个稳定解（在处理贝叶斯统计时，是指平稳分布或后验分布）的需求驱动的。Gelman 和 Rubin（1992）以及 Geyer（1992）都提出了用于确定收敛证据的经验方法。这些方法通常围绕比较链内的方差与链间的方差，或者马尔可夫链的一个部分的方差与同一链的另一个部分的方差。一种直观的方法是将多个链或链的部分的重要参数的边缘分布叠在一起绘制。如果它们匹配得足够好，就可以假设已经达到了收敛，尽管似乎没有人能确切定义什么程度可以被视为足够。由于这些方法都是经验性的，因此它们不能保证收敛是完整的，但迄今为止，还没有发现更多的分析方法。 在这里，我们将重点关注多个链的简单比较统计量（均值、中位数、分位数和方差）以及边缘分布图，但存在多种此类诊断方法（Geweke, 1989; Gelman and Rubin, 1992; Geyer, 1992）。R 包 **coda** 也实现了许多诊断工具，并可以使用简单的语句（如 post = coda::mcmc(posterior) ）导入我们即将生成的简单矩阵输出。然而，在这里，我们将继续使用显式的 R 进行制表和绘图，以便读者能够轻松地了解其机制。是否使用像 **coda** 这样已经非常出色的包，还是自己编写定制的绘图和制表函数，是一个需要你自己决定的问题，但前提是你必须知道如何编写自己的函数。

### 6.7.5 跳跃分布

每个新的候选参数向量是如何生成的取决于所谓的跳跃分布。有许多选项可用作跳跃分布，但通常情况下，对于集合 中的每个参数依次生成一个标准正态随机偏差 ，并将其缩放 以适应参数 的尺度，然后进行增量，即 [公式 6.22](#eq-6_22) 。如果有关参数相关性的信息，则有可能在每一步使用多元正态分布来生成一个完整的新候选参数向量；使用标准多元正态分布仍然是一种可能，这样就会忽略参数相关性。然而，在应用 Metropolis-Hastings 方法（Gibbs-内部-Metropolis）之前依次增量每个参数，在直观上似乎更容易理解（尽管并不总是最高效的）。

将 作为过程循环遍历 中的每个参数进行缩放非常重要，因为如果参数空间中的跳跃太大，那么跳跃的成功率可能会非常低；但如果跳跃太小，成功率可能会过高，并且可能需要巨大的迭代次数和时间来充分探索多维似然曲面并收敛到平稳分布。这个过程中存在试错元素，没有固定的或简单的规则来确定使用什么缩放因子。新候选值接受率是性能效率的指标。一个简单的经验法则可能是将正态随机偏差缩放到大约原始参数值的 0.5% 到 1.0%。在调整参数集的增量时，0.2 到 0.4（20-40%）的接受率可能是一个合理的目标。缩放值 通常会在不同参数之间有所不同，在开发新的分析时可能需要一些详细的搜索来找到可接受的值。可以将这种自适应搜索构建到运行 MCMC 的代码中。 通常，进行 MCMC 分析时，人们会使用专门用于执行此类分析的软件。例如，Gelman 等（2013）在附录中有一个题为《R 和 Bugs 中的计算示例》的部分，但现在建议使用名为 Stan 的软件而不是 Bugs（参见 https://mc-stan.org/，那里有丰富的文档）。然而，在这里，为了确保说明保持透明，没有黑箱，我们将完全在 R 环境中进行，同时强调此类分析需要根据每个案例进行一定程度的定制。

如果使用多元正态分布一次性增加所有参数，那么缩放因子将比单独增加参数时更小，这种减少与同时变化的参数数量有关。

### 6.7.6 MCMC的应用示例

我们再次使用 *abdat* 数据，以 Schaefer 剩余生产模型拟合为例，来说明这些方法。对于更复杂的模型（更多参数），进行分析所需的时间可能会大大增加，但基本原理仍然适用。

#activate and plot the fisheries data in abdat Fig 6.14   
data(abdat) # type abdat in the console to see contents   
plotspmdat(abdat) #use helper function to plot fishery stats vs year

|  |
| --- |
| 图 6.14: abdat 数据集中 cpue 和捕捞量时间序列。 |

### 6.7.7 马尔可夫链蒙特卡罗（MCMC）

描述 Gibbs-within-Metropolis 的方程式看起来相对直接，甚至简单。然而，它们的实现涉及许多更多细节。正如我们在模型参数估计章节中所见，在计算贝叶斯统计时，我们需要一种计算参数集似然的方法，但我们还需要一种计算其先验概率的方法（即使我们将其归因于分析的非信息性先验）。要实现 MCMC，还需要其他一些先决条件：

* 计算负对数似然和每个候选参数集的先验概率所需的函数
* 我们打算以哪个参数集开始 MCMC 过程，以及我们应该运行多长时间的 MCMC 过程，才开始存储接受的参数向量（即多长的 burn-in）？
* 在过程中，我们应该多久考虑接受一个结果（thinning rate）？
* 我们打算生成多少个独立的马尔可夫链，以及我们打算生成的链应该有多长，我们才停止 MCMC 过程？
* 在特定情况下，我们如何选择一个合适的权重集（即 ）？

依次回答这些要求和问题：

用于计算负对数似然值的函数仍然是 negLL() ，尽管我们在探索参数空间时，为了避免 低于零，或许应该使用 negLL1() ，它会在向量中的第一个参数接近零时施加惩罚（查看 negLL1() 的帮助和代码）。在这里，我们将始终假设每个参数的非信息性先验，因此我们在 MQMF 中包含了一个函数 calcprior() ，它仅将每个链长度的倒数对数相加。也就是说，它对所有可能的参数值赋予相等的似然。因此，单个似然值为 ，其中 是每个链的预定长度，包括燃烧长度。我们使用它们的对数值，因为我们处理的是对数似然值，以避免因处理极小数而产生的舍入误差。如果您有一个分析问题，其中希望对所有或部分参数使用信息性先验，您只需重写 calcprior() ，这将覆盖 **MQMF** 中的版本，所需结果将随之而来。

通常的做法是设置一个所谓的”燃烧期”，即从 MCMC 过程的起始点运行若干次迭代而不存储结果。这种舍弃早期结果的做法是为了确保马尔可夫链开始探索模型的似然曲面，而不是在极低似然空间中游荡。当然，这取决于你用来开始马尔可夫链的起始值。你可以用最大似然解来启动 MCMC，但这通常被认为可能使结果产生偏差。然而，如果你包含几百步的燃烧期，那么被接受的起始点就会偏离最优解。尽管如此，开始马尔可夫链候选参数向量时，最好使其与最优解有一定距离，并且对于多个链，每个链从不同的点开始。

观察 Gibbs-within-Metropolis 的方程式，可能会让人产生这样的印象：每一步通过选择候选参数向量并对其进行检验的过程，都可以导致马尔可夫链的增量。如果不存在参数相关性或连续抽样的自相关性，这种情况可能成立。然而，为了避免马尔可夫链中连续步骤之间的自相关性，通常在经过一定数量的迭代后，才考虑将某一步纳入链中。这种链抽稀设计旨在消除任何自相关性水平。我们已经知道 Schaefer 参数高度相关，并将以此知识为基础，探索需要多少程度的链抽稀才能消除马尔可夫链中的自相关性。一个潜在的混淆是，在使用 Gibbs-within-Metropolis 时，我们需要为每个参数进行抽稀步骤。因此，如果希望抽稀步长为 4，那么 do\_MCMC() 需要将 作为 *thinstep* 参数。

生成马尔可夫链的目标是使其收敛于平稳分布（后验分布），这将全面表征模型及其数据的不确定性。但如何确定这种收敛是否已经实现。一种方法是生成多个马尔可夫链，从不同的起点开始。如果它们都收敛，使得每个参数的边缘分布在重复链中非常相似，这将证明已经收敛。这可以通过图形可视化。然而，除了使用图形指标外，还应使用诊断统计量来指示是否已收敛到平稳分布。有许多此类统计量可用，但在这里我们仅提及一些简单的策略。与任何非线性求解器一样，从广泛的初始值开始 MCMC 过程是个好主意。任何用于识别 MCMC 是否已达到目标平稳分布的诊断测试，实际上都会考虑不同马尔可夫序列的收敛性。

当然，在进行任何比较之前，有必要丢弃所谓的燃烧阶段。燃烧阶段是指马尔可夫链在开始表征后验分布之前可能仅遍历稀疏似然空间。Gelman 等（2013 ）建议丢弃每个序列的前半部分，但实际选择的分数应由检查确定。最简单的诊断统计包括比较不同序列或同一序列不同部分的中位数和方差。如果来自不同序列（或序列的子集）的中值没有显著差异，那么可以认为序列已经收敛。同样，如果序列内（或序列子集）的方差与序列间没有显著差异，那么可以识别出收敛。使用单个序列可能看起来很方便，但如果收敛速度相对较慢且未知，那么依赖单个序列可能会提供错误的结果。 Gelman 和 Rubin（1992a）的标题清晰地指出了问题：“来自 Gibbs 采样的单个序列会带来虚假的安全感。”只需说，使用多个起始点来生成多个序列，并配合一系列诊断统计和图形，可以确保从任何 MCMC 模拟中得出的结论不是虚假的（Gelman 等，2013）。这些方法之所以被称为计算密集型，是有充分理由的。确定每个链生成序列的长度也是一个只有经验答案的问题。需要运行链直到收敛发生。这可能很快发生，也可能需要很长时间。模型中的参数越多，通常需要的时间就越长。高度不确定或不平衡的模型甚至可能无法收敛，这会是一种低效识别此类问题的方法。

为每个参数生成的增量所赋予的权重是另一个只能真正通过经验确定的事情。通过制定标准，这个过程可以自动化，并且通常在漫长的预热阶段开发，但在这里我们将采用试错法，并力求接受率在 20-40%之间。

### 6.7.8 MCMC的第一个示例

我们将通过第一个示例来说明上述的一些思想。为此，我们将从接近最大似然解的位置开始生成一个包含 10000 步的马尔可夫链，但有一个 50 次迭代的预热阶段（以进入信息量大的似然空间），以及一个 4 步的链稀疏率（这将不会避免连续点之间的自相关性，因为每 4 步进行一次稀疏，而参数为 4 意味着没有稀疏，但稍后再详细说明）。凭借后见之明（因为我多次运行了这个程序），我设置了 的值，以使试验的接受率在 20%到 40%之间。最后，我们还将运行三个包含 100 次迭代的短链，没有预热阶段，并且从不同的起始点开始，以说明预热阶段的影响以及使用不同起始点的作用。通过使用 set.seed() 函数，我们也将确保获得可重复的结果（这通常不是一个明智的选择）。

大部分工作由 MQMF 函数 do\_MCMC() 完成。你应该查看这个函数的帮助文档和代码，追踪描述 Gibbs-within-Metropolis 的方程在何处运行，以及先验概率是如何被包含的。你也应该能够看到接受率的计算方法。

# Conduct MCMC analysis to illustrate burn-in. Fig 6.15   
data(abdat); logce <- log(abdat$cpue)   
fish <- as.matrix(abdat) # faster to use a matrix than a data.frame!   
begin <- Sys.time() # enable time taken to be calculated   
chains <- 1 # 1 chain per run; normally do more   
burnin <- 0 # no burn-in for first three chains   
N <- 100 # Number of MCMC steps to keep   
step <- 4 # equals one step per parameter so no thinning   
priorcalc <- calcprior # define the prior probability function   
scales <- c(0.065,0.055,0.065,0.425) #found by trial and error   
set.seed(128900) #gives repeatable results in book; usually omitted   
inpar <- log(c(r= 0.4,K=11000,Binit=3600,sigma=0.05))   
result1 <- do\_MCMC(chains,burnin,N,step,inpar,negLL,calcpred=simpspm,   
 calcdat=fish,obsdat=logce,priorcalc,scales)   
inpar <- log(c(r= 0.35,K=8500,Binit=3400,sigma=0.05))   
result2 <- do\_MCMC(chains,burnin,N,step,inpar,negLL,calcpred=simpspm,   
 calcdat=fish,obsdat=logce,priorcalc,scales)   
inpar <- log(c(r= 0.45,K=9500,Binit=3200,sigma=0.05))   
result3 <- do\_MCMC(chains,burnin,N,step,inpar,negLL,calcpred=simpspm,   
 calcdat=fish,obsdat=logce,priorcalc,scales)   
burnin <- 50 # strictly a low thinning rate of 4; not enough  
step <- 16 # 16 thinstep rate = 4 parameters x 4 = 16   
N <- 10000 # 16 x 10000 = 160,000 steps + 50 burnin  
inpar <- log(c(r= 0.4,K=9400,Binit=3400,sigma=0.05))   
result4 <- do\_MCMC(chains,burnin,N,step,inpar,negLL,calcpred=simpspm,   
 calcdat=fish,obsdat=logce,priorcalc,scales)   
post1 <- result1[[1]][[1]]   
post2 <- result2[[1]][[1]]   
post3 <- result3[[1]][[1]]   
postY <- result4[[1]][[1]]   
cat("time = ",Sys.time() - begin,"\n")

time = 13.45083

cat("Accept = ",result4[[2]],"\n")

Accept = 0.3471241 0.3437158 0.354289 0.3826251

现在我们可以将这 10000 长的链绘制为一组点，并在这些点之上叠加三个较短的链，这些链没有使用预热阶段。这突出了预热阶段的重要性，同时也说明了不同的链如何独立地开始探索似然空间（在这里仅以二维表示）。如果这三个链更长，我们预计它们会在由灰色点占据的空间中穿越更广阔的区域。

#first example and start of 3 initial chains for MCMC Fig6.15   
parset(cex=0.85)   
P <- 75 # the first 75 steps only start to explore parameter space  
plot(postY[,"K"],postY[,"r"],type="p",cex=0.2,xlim=c(7000,13000),   
 ylim=c(0.28,0.47),col=8,xlab="K",ylab="r",panel.first=grid())   
lines(post2[1:P,"K"],post2[1:P,"r"],lwd=1,col=1)   
points(post2[1:P,"K"],post2[1:P,"r"],pch=15,cex=1.0)   
lines(post1[1:P,"K"],post1[1:P,"r"],lwd=1,col=1)   
points(post1[1:P,"K"],post1[1:P,"r"],pch=1,cex=1.2,col=1)   
lines(post3[1:P,"K"],post3[1:P,"r"],lwd=1,col=1)   
points(post3[1:P,"K"],post3[1:P,"r"],pch=2,cex=1.2,col=1)

|  |
| --- |
| 图 6.15: 从不同的起点（三角形、正方形、圆形）出发的三个独立 MCMC 链中的前 75 个点。这些短链没有设置预烧，因此记录从起点开始。灰色圆点是第四条链中的 10000 个点，其中有 50 个点的 “预烧”和 4 个点的 “稀疏率”，这给出了所有链都应趋近的静态分布的大致概念。 |

我们也可以通过使用 pairs() 函数和 rgb() 函数将每个参数与其他参数绘制出来，并使用颜色填充来检查参数的相关性细节，这使我们能够可视化点向 较大值和 较小值方向逐渐变软的趋势。要看到效果变化，将 *alpha* 参数（即 1/50）改为 1/1，使用 50 作为除数似乎在这个情况下（有 10000 个点）是一个合理的折中方案，用于展示密度的梯度。

#pairs plot of parameters from the first MCMC Fig 6.16   
posterior <- result4[[1]][[1]]   
msy <-posterior[,1]\*posterior[,2]/4   
pairs(cbind(posterior[,1:4],msy),pch=16,col=rgb(1,0,0,1/50),font=7)

|  |
| --- |
| 图 6.16: Schaefer 模型参数后验分布的 10000 个样本与 MSY 之间的关系。通常情况下，我们会使用比 4 更长的稀疏化步长来描述后验结果。图中的全部颜色至少来自 50 个点。 |

构成后验分布的接受参数向量可以单独绘制，以参数编号为横轴，提供每个参数的轨迹。理想情况下，应获得通常所说的“毛毛虫”形状，这在 [图 6.17](#fig-6-17) 中 参数的轨迹中尤为明显。其他轨迹上下波动，暗示每个轨迹中存在一定程度的自相关。 和 参数轨迹中明显的互补变化模式也支持这一观点。边缘分布提供了对每个参数经验分布形状的初步检验。根据数据情况，可能存在多个峰值或较平的顶部。然而，不规则的形状则表明缺乏收敛。

#plot the traces from the first MCMC example Fig 6.17   
posterior <- result4[[1]][[1]]   
par(mfrow=c(4,2),mai=c(0.4,0.4,0.05,0.05),oma=c(0.0,0,0.0,0.0))   
par(cex=0.8, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
label <- colnames(posterior)   
N <- dim(posterior)[1]   
for (i in 1:4) {   
 ymax <- getmax(posterior[,i]); ymin <- getmin(posterior[,i])   
 plot(1:N,posterior[,i],type="l",lwd=1,ylim=c(ymin,ymax),   
 panel.first=grid(),ylab=label[i],xlab="Step")   
 plot(density(posterior[,i]),lwd=2,col=2,panel.first=grid(),main="")   
}

|  |
| --- |
| 图 6.17: 四个 Schaefer 模型参数的轨迹以及每个参数的隐含边际分布。如果稀疏化步长增加到 128、256 或更长，迹线内明显的自相关性就会得到改善。 |

我们可以使用 R 函数 acf() 来检查马尔可夫链中连续步骤之间的自相关程度。该函数将向量中的值与其自身、滞后 1、滞后 2 等依次进行相关性分析。我们期望滞后 0 时的相关性为 1，但理想情况下，如果我们要避免低估总变异，马尔可夫链中连续项之间的相关性应该迅速降至无意义水平。将每个参数的自相关图与其轨迹和边缘分布进行比较。 与其它参数相比，其序列相关性相对较低，这一视觉差异应该很明显。

#Use acf to examine auto-correlation with thinstep = 16 Fig 6.18   
posterior <- result4[[1]][[1]]   
label <- colnames(posterior)[1:4]   
parset(plots=c(2,2),cex=0.85)   
for (i in 1:4) auto <- acf(posterior[,i],type="correlation",lwd=2,   
 plot=TRUE,ylab=label[i],lag.max=20)

|  |
| --- |
| 图 6.18: 四个 Schaefer 模型参数的轨迹显示的自相关性。这是在对四个参数中的每个参数进行 16 = 4 的稀疏化处理后得到的结果。显然，要消除出现的强相关性，需要大幅提高步长。 |

如果我们使用一个远大于初始值的薄化率运行 MCMC，希望能够观察到序列相关性的减少。这里我们将薄化率增加了 128 倍（从 4x4=16 增加到 4x128=512），并绘制了结果。尽管我们将链的长度减少到 1000，但总步数是（512 x 1000）+（512 x 100）= 563200，因此我们可以预期这次运行会比初始的 MCMC 运行稍长一些。了解运行时间总是个好主意。这些例子最多只需几分钟就能运行完成，而大多数用于严肃模型的 MCMC 运行需要数小时甚至数天。

#setup MCMC with thinstep of 128 per parameter Fig 6.19   
begin=gettime()   
scales <- c(0.06,0.05,0.06,0.4)   
inpar <- log(c(r= 0.4,K=9400,Binit=3400,sigma=0.05))   
result <- do\_MCMC(chains=1,burnin=100,N=1000,thinstep=512,inpar,   
 negLL,calcpred=simpspm,calcdat=fish,   
 obsdat=logce,calcprior,scales,schaefer=TRUE)   
posterior <- result[[1]][[1]]   
label <- colnames(posterior)[1:4]   
parset(plots=c(2,2),cex=0.85)   
for (i in 1:4) auto <- acf(posterior[,i],type="correlation",lwd=2,   
 plot=TRUE,ylab=label[i],lag.max=20)

|  |
| --- |
| 图 6.19: 当抽稀步长为 512 = 4 x 128 时，四个 Schaefer 模型参数的轨迹中显示的自相关情况，这是第一个自相关图中所用抽稀步长的 128 倍。 |

cat(gettime() - begin)

46.8787

当抽稀步长从 4 增加到 128（4 \* 128 = 512）时，四个参数轨迹中显示的自相关明显减少（ [图 6.19](#fig-6-19) ）。但即便 32 倍的增幅也不足以将滞后 2、3 和 4 内的相关性降至无意义。尽管如此，使用该抽稀步长生成的轨迹明显显示出差异（改进），相对于图 6.17 中的轨迹。然而，我们注意到，在当前使用的台式计算机上，563200 步（1100 长度链）大约耗时 40 秒，这台计算机开始对交互式工作有些慢。如果我们使用 512 的抽稀率进行 10000 次迭代，那将耗时近 7 分钟，这开始变得有些繁琐漫长。更复杂的模型可能需要更长的时间，有时甚至需要数天！因此，在我们探索更大抽稀步长的有效性之前，我们将首先找到显著加快每个循环的方法。 通过持续试验发现，将 i 增加到 1024（4 x 256）仍然存在显著的滞后 1 和有时滞后 2 的自相关，而将 *thinstep* 增加四倍到 2048（4 \* 3 \* 128）才能消除所有变量的自相关。

在寻找更快捷的方法之前，我们将完成对标准说明性图的分析，这些图可能在执行模型框架内不确定性或变异的 MCMC 检查时使用。

### 6.7.9 边际分布

一种可视化后验分布的方法是检查每个参数接受值的频率分布。在直方图上绘制 density() 函数的轮廓也可以改善我们对 MCMC 找到的分布的观察。在下面的情况下，即使每个参数的抽稀率为 128，1000 个复制样本似乎也不足以平滑每个分布。然而，确定复制样本数量是否足够实际上是在问后验分布是否收敛于平稳分布。与其依赖直觉和各种图形的外观，不如使用各种标准的诊断方法。大多数使用抽稀率为 16 绘制的与 MCMC 输出相关的图形似乎表明可能平滑的解。然而，自相关太大，输出可能存在偏差。最好使用可量化的诊断方法。

# plot marginal distributions from the MCMC Fig 6.20   
dohist <- function(x,xlab) { # to save a little space   
 return(hist(x,main="",breaks=50,col=0,xlab=xlab,ylab="",   
 panel.first=grid()))   
}   
 # ensure we have the optimum solution available   
param <- log(c(r= 0.42,K=9400,Binit=3400,sigma=0.05))   
bestmod <- nlm(f=negLL,p=param,funk=simpspm,indat=abdat,   
 logobs=log(abdat$cpue))   
optval <- exp(bestmod$estimate)   
posterior <- result[[1]][[1]] #example above N=1000, thin=512   
par(mfrow=c(5,1),mai=c(0.4,0.3,0.025,0.05),oma=c(0,1,0,0))   
par(cex=0.85, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
np <- length(param)   
for (i in 1:np) { #store invisible output from hist for later use   
 outH <- dohist(posterior[,i],xlab=colnames(posterior)[i])   
 abline(v=optval[i],lwd=3,col=4)   
 tmp <- density(posterior[,i])   
 scaler <- sum(outH$counts)\*(outH$mids[2]-outH$mids[1])   
 tmp$y <- tmp$y \* scaler   
 lines(tmp,lwd=2,col=2)   
}   
msy <- posterior[,"r"]\*posterior[,"K"]/4   
mout <- dohist(msy,xlab="MSY")   
tmp <- density(msy)   
tmp$y <- tmp$y \* (sum(mout$counts)\*(mout$mids[2]-mout$mids[1]))   
lines(tmp,lwd=2,col=2)   
abline(v=(optval[1]\*optval[2]/4),lwd=3,col=4)   
mtext("Frequency",side=2,outer=T,line=0.0,font=7,cex=1.0)

|  |
| --- |
| 图 6.20: 稀疏率为 128 时 1000 个点的边际后验分布。在每种情况下，垂直蓝线都是最大似然最优估计值。可能需要更多的重复来平滑分布。后验模式不一定与最大似然估计值相同。 |

## 6.8 Rcpp的应用

对于本章我们已使用的简单模型，运行 20 万个复制（薄化步长为 8 意味着这将涉及 160 万次似然计算）所需的时间并不算过于繁重。然而，对于更复杂的模型，或试图消除高水平的序列或自相关时，运行 MCMC 所需的时间可能会变得令人厌烦。如果你有一个运行起来感觉花费很长时间的流程，那么很可能值得对代码的这部分进行性能分析。在 R 中，这很容易通过使用 Rprof() 函数来完成。一旦启动，该函数会不时中断执行（默认为 0.02 秒），并确定中断时正在运行哪个函数。这些情况都会被记录下来，一旦软件运行完成，就可以应用 summaryRprof() 函数来发现哪些函数运行时间最长（self.time）。然后可以尝试加快这些慢速部分。total.time 包括函数本身花费的时间以及它调用的任何其他函数的时间。

#profile the running of do\_MCMC using the now well known abdat   
data(abdat); logce <- log(abdat$cpue); fish <- as.matrix(abdat)   
param <- log(c(r=0.39,K=9200,Binit=3400,sigma=0.05))   
Rprof(append=TRUE) # note the use of negLL1()   
result <- do\_MCMC(chains=1,burnin=100,N=20000,thinstep=16,inpar=param,   
 infunk=negLL1,calcpred=simpspm,calcdat=fish,   
 obsdat=logce,priorcalc=calcprior,   
 scales=c(0.07,0.06,0.07,0.45))   
Rprof(NULL)   
outprof <- summaryRprof()

kable(head(outprof$by.self,12))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.6: 将 Rprof() 函数应用于 do\_MCMC() 函数调用后的输出结果，仅查看输出列表的 by.self 部分（按每个函数的运行时间排序），检查 outprof 的结构。总采样时间在 outprof$sampling.time 中。self.pct 的总和为 99.99，因此这些值是需要关注的。   |  | self.time | self.pct | total.time | total.pct | | --- | --- | --- | --- | --- | | “funk” | 92.32 | 60.11 | 124.64 | 81.16 | | “mean” | 11.28 | 7.34 | 14.14 | 9.21 | | “max” | 10.08 | 6.56 | 10.08 | 6.56 | | “do\_MCMC” | 8.18 | 5.33 | 153.40 | 99.88 | | “which” | 6.88 | 4.48 | 9.54 | 6.21 | | “infunk” | 6.24 | 4.06 | 140.92 | 91.76 | | “dnorm” | 4.64 | 3.02 | 4.64 | 3.02 | | “priorcalc” | 2.34 | 1.52 | 2.76 | 1.80 | | “isTRUE” | 2.16 | 1.41 | 2.96 | 1.93 | | “mean.default” | 2.00 | 1.30 | 2.86 | 1.86 | | “penalty0” | 1.88 | 1.22 | 1.88 | 1.22 | | “numeric” | 1.22 | 0.79 | 1.22 | 0.79 | |

显然，几乎所有时间（total.time）都花在了 do\_MCMC() 函数中，但在该函数内部，大约 80%的时间用于 funk()= calcpred() = simpspm()。在函数 mean() 中也花费了合理的时间，等等。[*.data.frame*] 和都与将结果索引放入 do\_MCMC() 函数中的矩阵有关。如果它们出现在你的 Rprof 列表中，你可以通过将输入数据的 data.frame 转换为矩阵来部分提高速度。你应该比较使用每种数据形式的计算速度。R 软件确实非常出色，但即使近年来版本速度有所提升，以及使用现代计算机，也没有人会声称它在 MCMC 等计算密集型方法上迅速。显然，如果我们能够加快 simpspm() 函数（即 *funk*）的速度，那么在运行 MCMC 时可能会获得一些显著的速度提升。也许最好的方法是用 Stan（参见<https://mc-stan.org/>），但在尽可能保持在基础 R 范围内的前提下，我们将考察一种替代方案。

提高执行速度的一个非常有效的方法是将 R 代码与另一种可以编译成可执行代码而非 R 解释代码的计算机语言相结合。将 C++代码包含到 R 代码中最简单的方法可能是使用 **Rcpp** 包（Eddelbuettel & Francois, 2011; Eddelbuettel, 2013; Eddelbuettel & Balmuta, 2017）。显然，要使用这种方法，需要同时拥有 C++编译器和 **Rcpp** 包，这两者都可以从 CRAN 仓库下载（在 RStudio 中完成最为容易）。如果读者正在使用 Linux 或 Mac 计算机，那么他们已经拥有 GNU C++编译器（**Rcpp** 所使用的编译器）。在 Windows 系统上，安装 GNU C++编译器最简单的方法是访问 CRAN 主页，点击”Download R for Windows”链接，然后点击 Rtools 链接。务必将安装目录添加到路径中。这还提供了许多用于编写 R 包的工具。**Rcpp** 提供了一些将 C++代码包含进来的方法，其中最简单的方法可能是使用 cppFunction() 在每个会话开始时编译代码。 然而，更好的方法是使用函数 Rcpp::sourceCpp() 从磁盘加载 C++文件，就像你可能使用 source() 加载 R 代码文件一样。所以有多种选择，如果你打算采用这种加速代码的策略，这些选择都值得探索。

### 6.8.1 处理向量和矩阵

在下面的代码块中，可以看到 cppFunction() 需要将 C++代码输入为一段长文本字符串。使用 C++的一个复杂之处在于，在 R 中，向量、矩阵和数组的索引从 1:N,…，而在 C++中，相同数量的单元格索引将从 0:(N-1),….习惯的力量在我们来回切换 R 和 C++时可能会让我们犯傻（或者也许只是我）。例如，在开发下面的 C++代码时，我最初设置了 *biom[0] = ep[3]*。因此，我记住了在 *biom[0]*部分使用 0 而不是 1，但在设置生物量时间序列的初始生物量水平时，我又迅速忘记了索引问题，将参数向量中的 sigma 值设置为初始生物量 Binit 而不是。在 R 中，pars 变量在索引 1 中包含 ，在索引 2 中包含 ，在 3 中包含 ，在 4 中包含 ，但在 C++中，索引是 0、1、2 和 3。如果你觉得最后几句让你感到困惑，那就把它当作一件好事，因为希望如果你选择这条加速代码的路线，你会记得非常小心地在向量和矩阵中索引变量。 正如你很可能发现的那样，如果你在 C++中使用指向数组外部的索引（例如，在包含 0、1、2 和 3 的向量中，索引 4 不存在，但它会指向内存中的某个位置！），这通常会导致 R 崩溃并需要重启。在 R 中开发 C++代码时，你会很快学会在运行任何东西之前保存所有内容，我建议你也这样做。

当然，用 C++编程和使用 Rcpp 都有其复杂性，但本章或本书的目的并不是回顾这些主题。尽管如此，希望这个简单的例子能够说明在使用计算机密集型方法时，使用这些方法具有相当显著的优点，并成功鼓励你在适当的场合学习和使用这些方法。Eddelbuettel（2013）和 Wickham（2019）对 Rcpp 的优点提供了优秀的介绍。

### 6.8.2 simpspm() 的替代方法

如果要使用 simpspmC() 函数代替 simpspm()，则需要在运行任何代码之前运行以下代码块中的代码。

library(Rcpp)   
 #Send a text string containing the C++ code to cppFunction this will   
 #take a few seconds to compile, then the function simpspmC will   
 #continue to be available during the rest of your R session. The   
 #code in this chunk could be included into its own R file, and then   
 #the R source() function can be used to include the C++ into a   
 #session. indat must have catch in col2 (col1 in C++), and cpue in   
 #col3 (col2 in C++). Note the use of ; at the end of each line.   
 #Like simpspm(), this returns only the log(predicted cpue).   
cppFunction('NumericVector simpspmC(NumericVector pars,   
 NumericMatrix indat, LogicalVector schaefer) {   
 int nyrs = indat.nrow();   
 NumericVector predce(nyrs);   
 NumericVector biom(nyrs+1);   
 double Bt, qval;   
 double sumq = 0.0;   
 double p = 0.00000001;   
 if (schaefer(0) == TRUE) {   
 p = 1.0;   
 }   
 NumericVector ep = exp(pars);   
 biom[0] = ep[2];   
 for (int i = 0; i < nyrs; i++) {   
 Bt = biom[i];   
 biom[(i+1)]=Bt+(ep[0]/p)\*Bt\*(1-pow((Bt/ep[1]),p))-   
 indat(i,1);   
 if (biom[(i+1)] < 40.0) biom[(i+1)] = 40.0;   
 sumq += log(indat(i,2)/biom[i]);   
 }   
 qval = exp(sumq/nyrs);   
 for (int i = 0; i < nyrs; i++) {   
 predce[i] = log(biom[i] \* qval);   
 }   
 return predce;   
}')

一旦运行了 cppFunction() 代码，我们就可以在任何使用过 simpspm() 函数的地方使用 simpspmC() 函数。一个小麻烦是，simpspmC() 希望输入数据 *abdat* 是矩阵，而 *abdat* 以 data.frame 开始（实际上是列表，试试 class(abdat)）。输入 data.frame 而不是矩阵会导致 C++ 函数失效，因此，为了解决这个问题，在下面的代码中，你会看到我们使用了 as.matrix() 函数，以确保向 simpspmC() 发送正确的对象类别，幸运的是，使用矩阵比使用 data.frame 更快，因此我们也将其发送给了 simpspm()。我们还加入了 **microbenchmark** 软件包，以便准确比较两个不同函数的运行速度。显然，要使用该软件包，必须安装该软件包（如果不想安装，也可以省略）。在我的 Windows 10 2018 XPS 13 上进行的比较中，根据时间中位数，simpspmC() 所花的时间通常只有 simpspm() 的 20%，如表（6.8）所示。第一次使用 simpspmC() 函数时，有时启动速度非常慢，这会影响平均值，但中位数受到的干扰较小。

#Ensure results obtained from simpspm and simpspmC are same   
library(microbenchmark)   
data(abdat)   
fishC <- as.matrix(abdat) # Use a matrix rather than a data.frame   
inpar <- log(c(r= 0.389,K=9200,Binit=3300,sigma=0.05))   
spmR <- exp(simpspm(inpar,fishC)) # demonstrate equivalence   
 #need to declare all arguments in simpspmC, no default values   
spmC <- exp(simpspmC(inpar,fishC,schaefer=TRUE))   
out <- microbenchmark( # verything identical calling function   
 simpspm(inpar,fishC,schaefer=TRUE),   
 simpspmC(inpar,fishC,schaefer=TRUE),   
 times=1000   
)   
out2 <- summary(out)[,2:8]   
out2 <- rbind(out2,out2[2,]/out2[1,])   
rownames(out2) <- c("simpspm","simpspmC","TimeRatio")

kable(halftable(cbind(spmR,spmC)),row.names=TRUE,digits=c(4,4,4,4,4,4))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.7: simpspm() 和 simpspmC() 的预测结果并排展示，以证明代码从参数 c(r=0.389, K=9200, Binit=3300, sigma=0.05) 生成了相同的答案。   |  | spmR | spmC | spmR | spmC | | --- | --- | --- | --- | --- | | 1 | 1.1251 | 1.1251 | 1.9956 | 1.9956 | | 2 | 1.0580 | 1.0580 | 2.0547 | 2.0547 | | 3 | 1.0774 | 1.0774 | 2.1619 | 2.1619 | | 4 | 1.0570 | 1.0570 | 2.2037 | 2.2037 | | 5 | 1.0827 | 1.0827 | 2.1314 | 2.1314 | | 6 | 1.1587 | 1.1587 | 2.0773 | 2.0773 | | 7 | 1.2616 | 1.2616 | 2.0396 | 2.0396 | | 8 | 1.3616 | 1.3616 | 1.9915 | 1.9915 | | 9 | 1.4538 | 1.4538 | 1.9552 | 1.9552 | | 10 | 1.5703 | 1.5703 | 1.9208 | 1.9208 | | 11 | 1.7056 | 1.7056 | 1.8852 | 1.8852 | | 12 | 1.8446 | 1.8446 | 1.8276 | 1.8276 | |

现在我们可以汇总微基准测试的输出结果

kable(out2,row.names=TRUE,digits=c(3,3,3,3,3,3,3,0))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 6.8: simpspm 和 simpspmC 函数的微基准比较输出结果。微秒值分别是最小值、25 分位数、均值和中位数、75 分位数、最大值以及比较中的评估次数。TimeRatio 是第二行除以第一行，因此就均值而言，simpspmC 所需时间约为 simpspm 的 7%。实际值会在不同运行中有所变化，但变化不会太大，尽管不同计算机和不同版本的 R 之间可能会有差异（最好使用最新版本，通常最快）。   |  | min | lq | mean | median | uq | max | neval | | --- | --- | --- | --- | --- | --- | --- | --- | | simpspm | 48.700 | 52.650 | 70.386 | 58.400 | 72.700 | 3861.500 | 1000 | | simpspmC | 2.700 | 3.000 | 4.859 | 3.400 | 3.950 | 1039.000 | 1000 | | TimeRatio | 0.055 | 0.057 | 0.069 | 0.058 | 0.054 | 0.269 | 1 | |

simpspmC 的最大值有时会大于 simpspm，这是因为第一次调用时有时会花费更长的时间。如果出现这种情况，请尝试再次运行比较并注意最大值的变化。然而，真正感兴趣的比较是使用 simpspmC() 而不是 simpspm() 运行 MCMC 平均快多少。我们可以不用 microbenchmark 来做这个比较，因为每次运行所花费的时间现在是可以察觉的。相反，我们使用 MQMF 函数 gettime() ，它提供从每天开始以来的时间（以秒为单位）。

#How much does using simpspmC in do\_MCMC speed the run time?   
 #Assumes Rcpp code has run, eg source("Rcpp\_functions.R")   
set.seed(167423) #Can use getseed() to generate a suitable seed   
beginR <- gettime() #to enable estimate of time taken   
setscale <- c(0.07,0.06,0.07,0.45)   
reps <- 2000 #Not enough but sufficient for demonstration   
param <- log(c(r=0.39,K=9200,Binit=3400,sigma=0.05))   
resultR <- do\_MCMC(chains=1,burnin=100,N=reps,thinstep=128,   
 inpar=param,infunk=negLL1,calcpred=simpspm,   
 calcdat=fishC,obsdat=log(abdat$cpue),schaefer=TRUE,   
 priorcalc=calcprior,scales=setscale)   
timeR <- gettime() - beginR   
cat("time = ",timeR,"\n")

time = 21.51238

cat("acceptance rate = ",resultR$arate," \n")

acceptance rate = 0.319021 0.3187083 0.3282664 0.368747

postR <- resultR[[1]][[1]]   
set.seed(167423) # Use the same pseudo-random numbers and the   
beginC <- gettime() # same starting point to make the comparsion   
param <- log(c(r=0.39,K=9200,Binit=3400,sigma=0.05))   
resultC <- do\_MCMC(chains=1,burnin=100,N=reps,thinstep=128,   
 inpar=param,infunk=negLL1,calcpred=simpspmC,   
 calcdat=fishC,obsdat=log(abdat$cpue),schaefer=TRUE,   
 priorcalc=calcprior,scales=setscale)   
timeC <- gettime() - beginC   
cat("time = ",timeC,"\n") # note the same acceptance rates

time = 4.067156

cat("acceptance rate = ",resultC$arate," \n")

acceptance rate = 0.319021 0.3187083 0.3282664 0.368747

postC <- resultC[[1]][[1]]   
cat("Time Ratio = ",timeC/timeR)

Time Ratio = 0.1890611

尽管每次运行所需的确切时间会有所不同，因为你的电脑会同时运行其他进程，但通常使用 simpspmC() 所需的时间仅是使用 simpspm() 所需时间的 1/5 或 20%。在处理分钟时，这可能看起来并不重要，但一旦 MCMC 运行需要 20-40 小时，那么节省 16-32 小时可能会被认为更有价值。当然，还有其他潜在的方法可以加速这个过程，对于真正计算密集型的分析，这通常是值得的。

使用这两个函数中的任意一个所得到的结果与运行不同的链是等效的，尽管只有当 set.seed() 函数使用不同的值，或者，更好的是，根本不使用它时才成立。如果你使用了相同的种子，那么得到的结果边缘分布将是相同的。使用不同的种子，但只有 2000 次迭代，可能会出现一些偏差。现在我们有了更快速的方法，我们可以在保持较大的抽样间隔的同时探索更多的迭代次数。

#compare marginal distributions of the 2 chains Fig 6.21   
par(mfrow=c(1,1),mai=c(0.45,0.45,0.05,0.05),oma=c(0.0,0,0.0,0.0))   
par(cex=0.85, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
maxy <- getmax(c(density(postR[,"K"])$y,density(postC[,"K"])$y))   
plot(density(postR[,"K"]),lwd=2,col=1,xlab="K",ylab="Density",   
 main="",ylim=c(0,maxy),panel.first=grid())   
lines(density(postC[,"K"]),lwd=3,col=5,lty=2)

|  |
| --- |
| 图 6.21: 比较由 simpspm 函数（实黑线）和 simpspmC 函数（虚蓝线）生成的链的 K 参数密度分布，每条链具有相同的起始位置和相同的随机种子，它们相互重叠。使用不同的种子或不同的起始位置重复这些示例，以观察效果。 |

### 6.8.3 多个独立链

在进行 MCMC 分析时，最佳做法是运行多个链，但在实际操作中，生成链的总数往往需要在可用时间与至少三个或更多链之间进行权衡。重要的是提供足够的证据来支持分析师关于分析已达到收敛的说法。这里我们将只使用三个链，尽管实际上，对于这样一个简单的模型，运行更多的链会更有说服力。为了提高速度，我们现在只使用 simpspmC()，因为每条细化前的链长为 万次迭代）。

#run multiple = 3 chains   
setscale <- c(0.07,0.06,0.07,0.45) # I only use a seed for   
set.seed(9393074) # reproducibility within this book   
reps <- 10000 # reset the timer   
beginC <- gettime() # remember a thinstep=256 is insufficient   
resultC <- do\_MCMC(chains=3,burnin=100,N=reps,thinstep=256,   
 inpar=param,infunk=negLL1,calcpred=simpspmC,   
 calcdat=fishC,obsdat=log(fishC[,"cpue"]),   
 priorcalc=calcprior,scales=setscale,schaefer=TRUE)   
cat("time = ",gettime() - beginC," secs \n")

time = 119.5804 secs

#3 chain run using simpspmC, 10000 reps, thinstep=256 Fig 6.22   
par(mfrow=c(2,2),mai=c(0.4,0.45,0.05,0.05),oma=c(0.0,0,0.0,0.0))   
par(cex=0.85, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
label <- c("r","K","Binit","sigma")   
for (i in 1:4) {   
 plot(density(resultC$result[[2]][,i]),lwd=2,col=1,   
 xlab=label[i],ylab="Density",main="",panel.first=grid())   
 lines(density(resultC$result[[1]][,i]),lwd=2,col=2)   
 lines(density(resultC$result[[3]][,i]),lwd=2,col=3)   
}

|  |
| --- |
| 图 6.22: 在 64（4x64=256）的稀疏率下使用 10000 次重复和 simpspmC 函数计算的四种 Schaefer 参数的边际密度分布在三个链之间的差异。在线宽大于平均值的地方，会出现明显的细微差别。 |

我们还可以生成不同链的汇总统计数据。事实上，有许多不同的诊断统计和图表可以使用。

#generate summary stats from the 3 MCMC chains   
av <- matrix(0,nrow=3,ncol=4,dimnames=list(1:3,label))   
sig2 <- av # do the variance   
relsig <- av # relative to mean of all chains   
for (i in 1:3) {   
 tmp <- resultC$result[[i]]   
 av[i,] <- apply(tmp[,1:4],2,mean)   
 sig2[i,] <- apply(tmp[,1:4],2,var)   
}   
cat("Average \n")

Average

av

r K Binit sigma  
1 0.3821707 9495.580 3522.163 0.04805695  
2 0.3809524 9530.307 3537.186 0.04811021  
3 0.3822318 9487.911 3522.021 0.04810015

cat("\nVariance per chain \n")

Variance per chain

sig2

r K Binit sigma  
1 0.0009018616 1060498.2 151208.8 6.264484e-05  
2 0.0008855405 998083.0 142153.1 6.177037e-05  
3 0.0009080043 978855.6 138585.3 6.288734e-05

cat("\n")

for (i in 1:4) relsig[,i] <- sig2[,i]/mean(sig2[,i])   
cat("Variance Relative to Mean Variance of Chains \n")

Variance Relative to Mean Variance of Chains

relsig

r K Binit sigma  
1 1.0037762 1.0474275 1.0501896 1.0033741  
2 0.9856108 0.9857815 0.9872949 0.9893677  
3 1.0106130 0.9667911 0.9625155 1.0072582

如果我们对不同分布的量值进行比较，就能更清楚地了解差异的程度。百分比差异是指我们直接比较 2.5%和 97.5% 分位数（分布的中心 95%）的值以及第二和第三边际分布的中值，只有一个比较点（Binit 的 97.5% 上限点）大于 1%。

#compare quantile from the 2 most widely separate MCMC chains   
tmp <- resultC$result[[2]] # the 10000 values of each parameter   
cat("Chain 2 \n")

Chain 2

msy1 <- tmp[,"r"]\*tmp[,"K"]/4   
ch1 <- apply(cbind(tmp[,1:4],msy1),2,quants)   
round(ch1,4)

r K Binit sigma msy1  
2.5% 0.3206 7926.328 2942.254 0.0356 853.1769  
5% 0.3317 8140.361 3016.340 0.0371 859.6908  
50% 0.3812 9401.467 3489.550 0.0472 896.5765  
95% 0.4287 11338.736 4214.664 0.0624 955.1773  
97.5% 0.4386 11864.430 4425.248 0.0662 970.7137

tmp <- resultC$result[[3]]   
cat("Chain 3 \n")

Chain 3

msy2 <- tmp[,"r"]\*tmp[,"K"]/4   
ch2 <- apply(cbind(tmp[,1:4],msy2),2,quants)   
round(ch2,4)

r K Binit sigma msy2  
2.5% 0.3225 7855.611 2920.531 0.0355 853.0855  
5% 0.3324 8090.493 3001.489 0.0371 859.3665  
50% 0.3826 9370.715 3475.401 0.0471 895.8488  
95% 0.4316 11248.955 4188.052 0.0626 952.1486  
97.5% 0.4416 11750.426 4376.639 0.0665 966.2832

cat("Percent difference ")

Percent difference

cat("\n2.5% ",round(100\*(ch1[1,] - ch2[1,])/ch1[1,],4),"\n")

2.5% -0.6006 0.8922 0.7383 0.4636 0.0107

cat("50% ",round(100\*(ch1[3,] - ch2[3,])/ch1[3,],4),"\n")

50% -0.3871 0.3271 0.4055 0.2278 0.0812

cat("97.5% ",round(100\*(ch1[5,] - ch2[5,])/ch1[5,],4),"\n")

97.5% -0.6817 0.9609 1.0985 -0.5278 0.4564

### 6.8.4 所需重复实验以避免序列相关

我们在前面已经看到，如果稀疏率过低，每个参数的迹线或序列内的自相关性就会很高。显然，增加稀疏化步数会降低自相关性。但不太清楚的是，需要多大的稀疏率才能使这种相关性变得不明显。

现在我们有了一种更快的方法来探讨这个问题，我们可以寻找所需的稀疏率规模。我们从之前的试验中得知，每个参数的稀疏率为 128 时，滞后 2 到 4 步之间仍然存在显著的相关性，因此，我们应该研究稀疏率为 1024（）和 2048（）时的效果。为了更严格地进行比较，我们平衡了稀疏率和迭代次数，因此我们使用 和 ，两者都 。问题在于消除自相关是否能更好地掌握不同参数的全部变化。但是，为了使试验具有可比性，未稀疏链的长度必须相同，这样才能对似然曲面进行相同程度的探索。因此，在较小的稀疏率下，我们需要更多的迭代次数，同时还需要考虑烧入期（一个烧入 100 次，另一个烧入 50 次）。[图 6.23](#fig-6-23) 中，稀疏率为 1024 时，滞后期为 1 时仍有显著相关性，而稀疏率为 2028 时，相关性消失。即使使用 simpspmC()，该例程也需要 60 秒左右。

消除序列内相关性的重要性在于，如果序列内相关性很高，就会干扰向静态分布的收敛（因为序列点是相关的，而不是跟踪似然曲面的全部范围），结果可能无法捕捉到模型和所研究的可用数据固有的全部变化范围。

#compare two higher thinning rates per parameter in MCMC   
param <- log(c(r=0.39,K=9200,Binit=3400,sigma=0.05))   
setscale <- c(0.07,0.06,0.07,0.45)   
result1 <- do\_MCMC(chains=1,burnin=100,N=2000,thinstep=1024,   
 inpar=param,infunk=negLL1,calcpred=simpspmC,   
 calcdat=fishC,obsdat=log(abdat$cpue),   
 priorcalc=calcprior,scales=setscale,schaefer=TRUE)   
result2 <- do\_MCMC(chains=1,burnin=50,N=1000,thinstep=2048,   
 inpar=param,infunk=negLL1,calcpred=simpspmC,   
 calcdat=fishC,obsdat=log(abdat$cpue),   
 priorcalc=calcprior,scales=setscale,schaefer=TRUE)

#autocorrelation of 2 different thinning rate chains Fig6.23   
posterior1 <- result1$result[[1]]   
posterior2 <- result2$result[[1]]   
label <- colnames(posterior1)[1:4]   
par(mfrow=c(4,2),mai=c(0.25,0.45,0.05,0.05),oma=c(1.0,0,1.0,0.0))   
par(cex=0.85, mgp=c(1.35,0.35,0), font.axis=7,font=7,font.lab=7)   
for (i in 1:4) {   
 auto <- acf(posterior1[,i],type="correlation",plot=TRUE,   
 ylab=label[i],lag.max=20,xlab="",ylim=c(0,0.3),lwd=2)   
 if (i == 1) mtext(1024,side=3,line=-0.1,outer=FALSE,cex=1.2)   
 auto <- acf(posterior2[,i],type="correlation",plot=TRUE,   
 ylab=label[i],lag.max=20,xlab="",ylim=c(0,0.3),lwd=2)   
 if (i == 1) mtext(2048,side=3,line=-0.1,outer=FALSE,cex=1.2)   
}   
mtext("Lag",side=1,line=-0.1,outer=TRUE,cex=1.2)

|  |
| --- |
| 图 6.23: 两条链在 Schaefer 模型四个参数上的自相关性，综合稀疏率分别为 1024 和 2048。请注意，Y 轴上的最大值缩小了，使两者之间的差异更加明显。 |

我们可以用比较上述三个复制链的相同方法来比较具有不同稀疏率的两个链。也就是说，我们可以绘制它们的边际分布图，并比较它们的量值分布。由于稀疏化后的最终复制数量有限，它们的边际分布惊人地相似（[图 6.23](#fig-6-23)）。然而，在比较它们的量值分布时，观察到的分布中心 95% 之间的差异往往比上文 “多重独立链”一节中比较的三条链要大。这些差异似乎并没有遵循任何特定的方向，不过，随着下限和上限向同一方向移动，似乎存在一些偏差，但需要更多的重复才能澄清这一点。

#visual comparison of 2 chains marginal densities Fig 6.24   
parset(plots=c(2,2),cex=0.85)   
label <- c("r","K","Binit","sigma")   
for (i in 1:4) {   
 plot(density(result1$result[[1]][,i]),lwd=4,col=1,xlab=label[i],   
 ylab="Density",main="",panel.first=grid())   
 lines(density(result2$result[[1]][,i]),lwd=2,col=5,lty=2)   
}

|  |
| --- |
| 图 6.24: 在 2048（虚线）和 1024（黑色实线）的稀疏率下，使用 1000 和 2000 个重复序列计算的 K 参数边际密度分布在两个链之间的变化。The variation between two chains in the marginal density distributions for the K parameter using 1000 and 2000 replicates at thinning rates of 2048 (dashed line) and 1024 (solid black line). |

独立的 MCMC 链总会存在一定程度的差异，这就是相似性标准概念变得重要的原因。在这两条链中，虽然存在明显的差异，但中位数的实际差异都小于 1%，而在 10 个外部量级中，有 8 个的实际差异都小于 1%。我们理应相信薄化率更高的链。

#tablulate a summary of the two different thinning rates.   
cat("1024 thinning rate \n")

1024 thinning rate

posterior <- result1$result[[1]]   
msy <-posterior[,1]\*posterior[,2]/4   
tmp1 <- apply(cbind(posterior[,1:4],msy),2,quants)   
rge <- apply(cbind(posterior[,1:4],msy),2,range)   
tmp1 <- rbind(tmp1,rge[2,] - rge[1,])   
rownames(tmp1)[6] <- "Range"   
print(round(tmp1,4))

r K Binit sigma msy  
2.5% 0.3221 7918.242 2943.076 0.0352 853.5243  
5% 0.3329 8139.645 3016.189 0.0367 858.8872  
50% 0.3801 9429.118 3499.826 0.0470 895.7376  
95% 0.4289 11235.643 4172.932 0.0627 953.9948  
97.5% 0.4392 11807.732 4380.758 0.0663 973.2185  
Range 0.2213 7621.901 2859.858 0.0612 238.5436

posterior2 <- result2$result[[1]]   
msy2 <-posterior2[,1]\*posterior2[,2]/4   
cat("2048 thinning rate \n")

2048 thinning rate

tmp2 <- apply(cbind(posterior2[,1:4],msy2),2,quants)   
rge2 <- apply(cbind(posterior2[,1:4],msy2),2,range)   
tmp2 <- rbind(tmp2,rge2[2,] - rge2[1,])   
rownames(tmp2)[6] <- "Range"   
print(round(tmp2,4))

r K Binit sigma msy2  
2.5% 0.3216 7852.002 2920.198 0.0351 855.8295  
5% 0.3329 8063.878 3000.767 0.0368 859.8039  
50% 0.3820 9400.708 3482.155 0.0468 896.6774  
95% 0.4313 11235.368 4184.577 0.0628 959.2919  
97.5% 0.4434 11638.489 4456.164 0.0676 975.4358  
Range 0.2189 8156.444 3161.232 0.0546 257.1803

cat("Inner 95% ranges and Differences between total ranges \n")

Inner 95% ranges and Differences between total ranges

cat("95% 1 ",round((tmp1[5,] - tmp1[1,]),4),"\n")

95% 1 0.1172 3889.49 1437.682 0.0311 119.6942

cat("95% 2 ",round((tmp2[5,] - tmp2[1,]),4),"\n")

95% 2 0.1218 3786.487 1535.966 0.0325 119.6064

cat("Diff ",round((tmp2[6,] - tmp1[6,]),4),"\n")

Diff -0.0024 534.5429 301.3746 -0.0066 18.6367

## 6.9 结束语

本章的目的不是鼓励人们编写自己的自举、渐近误差、似然分布或 MCMC 函数，而是让他们探索这些方法，获得直觉，以便在使用这些方法时能够清楚地认识到它们的优势，同样重要的是，认识到它们的局限性。在实施 MCMC 分析时尤其如此，人们最好使用 Stan 或 Template Model Builder（Kristensen 等，2016）或 AD Model Builder（Fournier 等，1998；Fournier 等，2012）等工具。尽管如此，我们还是通过使用 MCMC 详细介绍了贝叶斯统计的应用，因为这确实是捕捉任何特定建模分析中固有的所有不确定性的最佳方法。尽管如此，在许多渔业模型中，许多参数，如自然死亡率、繁殖陡度和一些选择性参数，都被设定为常数，在贝叶斯背景下，这意味着信息量极大的先验。这种说法有点矫揉造作，因为如果不对这些参数进行估计，我们就不需要考虑任何先验概率，但原则上这就是它的含义。解决此类问题的通常方法是研究此类参数的似然曲线，或进行敏感性分析，研究使用不同常数的影响。甚至可以使用标准化的似然曲线作为先验概率。

诚然，使用 MCMC 可以更全面地描述建模情景中的变化，但这种分析无法捕捉模型的不确定性，在这种情况下，结构不同的模型可能会对所评估的种群动态提供不同的看法。这就是通常讨论的模型平均概念。不过，这也提出了一个问题，即哪个模型被认为是最现实的，以及在它们可能完全不相称的情况下，每个模型的权重是多少。不过，在研究任何建模结果时，都需要考虑模型的不确定性。正如 Punt 和 Hilborn（1997）所说：“最常见的方法是选择一个单一的结构模型，并只考虑其参数的不确定性。另一种更站得住脚的方法是考虑一系列真正不同的结构模型。然而，除了计算量更大之外，还很难’约束’所考虑的模型范围。与此相关的一个问题是，如何确定有多少模型参数应被视为不确定参数”。

不确定性的特征描述非常重要，因为它提供了在提供管理建议时可以有多大信心的一些概念。人们可能会迷失在计算细节中，而忘记了主要目标是为某种自然资源提供站得住脚的管理建议。要做到这一点，并没有单一的方法，因此，如果情况导致 MCMC 始终无法收敛，仍有可能采用其他方法，并对评估结果和种群的状况进行长期描述。如果了解这些方法，并知道如何使用和解释这些方法所能发现的模型及其数据，显然会有所帮助。但最终，了解渔业历史以及除渔获量外的其他影响因素也会有所帮助。

# 7. 剩余生产模型

## 7.1 引言

在前面的章节中，我们已经使用并拟合了所谓的静态模型，这些模型在一段时间内是稳定的（*例如*，使用vB() 、Gz()或mm() 的生长模型 ）。此外，在第 [4](#sec-paraestimat) 章 [模型参数估计](04-application.qmd) 和第 [6](#sec-uncertainty) 章[不确定性](06-uncertainty.qmd)两章中，我们已经介绍了剩余生产模型（surplus production model, spm），这些模型可用于进行资源评估（例如 Schaefer 模型），并提供了一个时间序列数据的动态模型的示例。然而，当我们专注于特定的建模方法时，此类模型的细节开发受到限制。我们将在本章深入研究剩余生产模型。

剩余生产模型（或者生物量动态模型）(Hilborn 和 Walters 1992) 将补充、生长和死亡率（生产的所有方面）的总体效应汇集到一个单一的生产方程中，处理无差异的生物量（或数量）。“无差异的（undifferentiated）”一词意味着忽略了年龄和长度组成以及性别和其他差异的所有方面，实际上都被忽略了。

为了进行正式的种群评估，就必须以某种方式对已开发资源的动态行为和生产力进行建模。这些动态的主要组分是种群对不同捕捞压力随时间推移的反应方式，即其资源量增加或减少的程度。通过研究不同捕捞强度水平的影响，一般可以评估种群的生产力。剩余生产模型提供了最简单的种群评估，尝试在模型与渔业数据拟合的基础上对种群动态进行描述。

在20世纪50年代，M. B. Schaefer (1957); M. Schaefer (1991) 描述了如何使用剩余生产模型来生成渔业种群评估。此后，这些模型得到很多发展 (Hilborn 和 Walters 1992; Prager 1994; Haddon 2011; Winker, Carvalho, 和 Kapur 2018)，我们将在本章简要介绍这些较近期的动态模型。

### 7.1.1 数据需求

估算这种模型的现代离散模型参数所需的最基本数据至少是一个相对丰度指数的时间序列和一个相关的渔获量数据时间序列。渔获量数据可以涵盖比相对丰度指数数据的年份更长。简单模型中使用的相对种群丰度指数一般是单位努力渔获量（cpue），但也可以是一些与渔业无关的丰度指数（例如，来自拖网调查、声学调查），或者两者皆具。

### 7.1.2 对比的需求

尽管最近偶尔使用 (Elder 1979; Saila 等 1979; Punt 和 Hilborn 1997)，但剩余产量模型在 20 世纪 80 年代似乎不再流行。可能是因为在开发这些模型的早期，须假设被评估的种群处于平衡状态 (Elder 1979; Saila 等 1979)，而这往往导致过于乐观的结论，从长远来看是站不住脚的。Hilborn (1979) 分析了许多此类情况，并证明使用的数据往往过于单一；他们的努力量水平上缺乏对比，因此对相关种群的动态缺乏信息。数据缺乏对比度意味着渔获量和努力量信息只能用于有限范围的种群丰度水平和有限的捕捞强度水平。有限的努力强度范围意味着对不同捕捞强度水平的反应范围也将有限。当种群动态更多地受环境因素而非渔获量的驱动时，也会出现这种对比的缺乏，因此种群似乎以意想不到的方式对渔业做出反应（例如，尽管渔获量或努力量没有变化，但种群发生了巨大变化）。

剩余产量模型的一个重要假设是，所使用的相对丰度度量能够提供了种群相对丰度随时间的信息指标。一般来说，假设种群丰度与 CPUE 或其他指数之间存在线性关系（尽管这不一定是 1：1 的关系）。显而易见的风险是，这个假设要么是错误的，要么可以根据情况发生变化。例如，cpue 可能会变得非常稳定，这意味着即使种群数量减少或增加，它也不会发生变化。或者，由于外部因素影响，指数的变化可能会非常大，以至于无法检测到丰度趋势。例如，可能会观察到不同年份间 CPUE 的巨大变化，但考虑到种群的生产力，这种变化在生物学上是不可能的 （Haddon，2018）。

一个不同但相关的假设是，努力量的质量和随后的渔获率在一段时间内保持不变。不幸的是，由于捕捞网具的技术变化、捕捞行为或方法的改变，或捕捞效率的其他变化，从而形成“努力量爬升”的概念，对于依赖 CPUE 作为相对丰度指数的评估来说，总是一个挑战或问题。努力量递增变的概念意味着努力量的有效性增加，因此任何基于名义努力量观测到的名义 cpue都会高估相对种群丰度（偏高）。cpue 的统计标准化 （Kimura，1981；Haddon，2018）可以解决其中一些问题，但显然只能考虑可获得数据的因素。例如，如果在渔业中引入 GPS 绘图仪或彩色回声测深仪，这往往会提高捕捞效率，但却没有记录哪些船只以及何时引入这些设备，那么这些设备对渔获率的正面影响将无法通过标准化来解释。

### 7.1.3 渔获率何时具有参考价值

检验丰度与任何相对丰度指数之间的假定关系是否真实和有信息的一个可能方法是，在发达渔业中，如果允许渔获量增加，预计 cpue 会在一段时间后开始下降。同样，如果渔获量减少到小于剩余产量（可能是通过管理或营销变化），那么随着种群规模的增加，预计cpue会在一段时间后很快增加。其原理是，如果渔获量低于种群当前的产量，那么最终种群规模和 cpue 都会增加，反之亦然。如果渔获量因供应不足而下降，但仍保持或高于当前生产力，则 cpue 当然不会增加，甚至可能进一步下降，尽管渔获量可能略有减少。重点放在发达渔业上，因为当渔业开始时，生物量的任何初始枯竭都会导致 “意外”渔 获量 (MacCall 2009)，因为种群被捕捞减少，这反过来又会导致 cpue 水平，一旦种群从未捕获水平减少，cpue 水平将无法维持。

因此，预计在发达渔业中，cpue 在许多情况下与渔获量呈负相关，可能在 cpue 随渔获量变化而变化之间存在时滞。如果我们能发现这种关系，通常意味着数据中存在一定程度的反差；如果我们找不到这种负相关关系，通常意味着数据中有关种群如何对渔业做出反应的信息含量太低，无法仅根据渔获量和相对丰度指数进行评估。也就是说，在渔获量的基础上，相对丰度指数几乎没有增加更多的信息。

我们将使用 **MQMF** 数据集 *schaef* 来说明这些观点。*Schaefer* 包含 M. B. Schaefer (1957) 原始黄鳍金枪鱼数据的渔获量和 CPUE，这是使用剩余产量模型进行种群评估的早期范例。

# Yellowfin-tuna data from Schaefer 12957  
  
library(MQMF)  
library(tidyverse)  
library(gt)  
library(patchwork)  
library(knitr)  
  
data(schaef)  
  
kable(schaef[1:11,], digits = 3, row.names = FALSE)  
kable(schaef[12:22,], digits = 3, row.names = FALSE)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | year | catch | effort | cpue | | --- | --- | --- | --- | | 1934 | 60913 | 5879 | 10.361 | | 1935 | 72294 | 6295 | 11.484 | | 1936 | 78353 | 6771 | 11.572 | | 1937 | 91522 | 8233 | 11.116 | | 1938 | 78288 | 6830 | 11.462 | | 1939 | 110417 | 10488 | 10.528 | | 1940 | 114590 | 10801 | 10.609 | | 1941 | 76841 | 9584 | 8.018 | | 1942 | 41965 | 5961 | 7.040 | | 1943 | 50058 | 5930 | 8.441 | | 1944 | 64094 | 6397 | 10.019 | | | year | catch | effort | cpue | | --- | --- | --- | --- | | 1945 | 89194 | 9377 | 9.512 | | 1946 | 129701 | 13958 | 9.292 | | 1947 | 160134 | 20381 | 7.857 | | 1948 | 200340 | 23984 | 8.353 | | 1949 | 192458 | 23013 | 8.363 | | 1950 | 224810 | 31856 | 7.057 | | 1951 | 183685 | 18726 | 9.809 | | 1952 | 192234 | 31529 | 6.097 | | 1953 | 138918 | 36423 | 3.814 | | 1954 | 138623 | 24995 | 5.546 | | 1955 | 140581 | 17806 | 7.895 | |

表 7.1: 1934 - 1955年的黄鳍金枪鱼渔业数据（Schaefer，1957）。 渔获量以千磅为单位，努力量以千个标准4级剪网日为单位，cpue以千磅/日为单位。

渔获量、cpue 及其关系图（[图 7.1](#fig-spm1)）仅显示 cpue 和渔获量之间微弱的负相关关系。如果我们用 summary(model) 检验 lm() 回归结果，我们发现回归仅有 的显著性。但是，这反映了没有时滞的相关性，即 。我们不知道需要经过多少年才能发现渔获量变化对 cpue 的潜在影响，因此，需要对 cpue 和渔获量之间进行时滞相关分析；为此，我们可以使用基本 R 语言的交互相关函数 ccf() 。

# schaef fishery data and regress cpue and catch Fig 7.1  
# parset(plots=c(3,1),margin=c(0.35,0.4,0.05,0.05))  
# plot1(schaef[,"year"],schaef[,"catch"],ylab="Catch",xlab="Year",  
# defpar=FALSE,lwd=2)  
# plot1(schaef[,"year"],schaef[,"cpue"],ylab="CPUE",xlab="Year",  
# defpar=FALSE,lwd=2)  
# plot1(schaef[,"catch"],schaef[,"cpue"],type="p",ylab="CPUE",  
# xlab="Catch",defpar=FALSE,pch=16,cex=1.0)  
# model <- lm(schaef[,"cpue"] ~ schaef[,"catch"])  
# abline(model,lwd=2,col=2) # summary(model)  
  
  
p1<- ggplot(data = schaef, aes(x = year, y = catch)) +  
 geom\_line() +  
 labs(x = "Year", y = "Catch") +  
 theme\_bw()  
  
p2<- ggplot(data = schaef, aes(x = year, y = cpue)) +  
 geom\_line() +  
 labs(x = "Year", y = "CPUE") +  
 theme\_bw()  
  
p3<- ggplot(data = schaef, aes(x = catch, y = cpue)) +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 labs(x = "Catch", y = "CPUE") +  
 theme\_bw()  
  
 p1/p2/p3

|  |
| --- |
| 图 7.1: Schaefer（1957）黄鳍金枪鱼渔业数据的各年渔获量和捕获量， 以及它们之间的回归关系。 |

如前所述，有迹象表明 只是刚刚显著。然而，在时滞2年时，CPUE与渔获量呈显著负相关（[图 7.2](#fig-spm2)），表明黄鳍金枪鱼数据中有足够的对比度来为剩余生产模型提供信息（在 1 年、3 年和 4 年也有显著影响）。如果我们将 CPUE 数据物理滞后两年，这种相关性应该会变得更加明显（ [图 7.3](#fig-spm3)）。

# cross correlation between cpue and catch in schaef Fig 7.2  
parset(cex = 0.85) # sets par parameters for a tidy base graphic  
ccf(  
 x = schaef[, "catch"], y = schaef[, "cpue"], type = "correlation",  
 ylab = "Correlation", plot = TRUE  
)

|  |
| --- |
| 图 7.2: 使用 R 中的 ccf() 函数获得的 Schaefer（1957）黄鳍金枪鱼渔业数据 (schaef) 的渔获量与 cpue 之间的交叉相关性。 |

# now plot schaef data with timelag of 2 years on cpue Fig 7.3  
model2 <- lm(schaef[3:22, "cpue"] ~ schaef[1:20, "catch"])  
# parset(plots=c(3,1),margin=c(0.35,0.4,0.05,0.05))  
# plot1(schaef[1:20,"year"],schaef[1:20,"catch"],ylab="Catch",  
# xlab="Year",defpar=FALSE,lwd=2)  
# plot1(schaef[3:22,"year"],schaef[3:22,"cpue"],ylab="CPUE",  
# xlab="Year",defpar=FALSE,lwd=2)  
# plot(schaef[1:20,"catch"],schaef[3:22,"cpue"],type="p",  
# ylab="CPUE",xlab="Catch",defpar=FALSE,cex=1.0,pch=16)  
#  
# abline(model2,lwd=2,col=2)  
  
p1 <- schaef |>  
 filter(year <= 1953) |>  
 ggplot(aes(x = year, y = catch)) +  
 geom\_line() +  
 theme\_bw()  
  
p2 <- schaef |>  
 filter(year > 1935) |>  
 ggplot(aes(x = year, y = cpue)) +  
 geom\_line() +  
 theme\_bw()  
  
l <- nrow(schaef)  
schaef1 <- data.frame(  
 catch = schaef[1:(l - 2), "catch"],  
 cpue = schaef[3:l, "cpue"]  
)  
  
p3 <- ggplot(data = schaef1, aes(x = catch, y = cpue)) +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 theme\_bw()  
  
p1/p2/p3

|  |
| --- |
| 图 7.3: Schaefer （1957）黄鳍金枪鱼渔业数据中的渔获量和 cpue 及其关系。当 cpue 时间序列的负滞后期为 2 年时，负相关或反相关关系变得更加明显。 |

M. B. Schaefer (1957) 的黄鳍金枪鱼渔业数据中 cpue 与渔获量之间的关系，对 cpue 时间序列施加了2年的负时滞后（第3：22行与第1：20行）。极小的梯度反映了以千磅为单位报告的渔获量。

# write out a summary of he regression model2  
summary(model2)

Call:  
lm(formula = schaef[3:22, "cpue"] ~ schaef[1:20, "catch"])  
  
Residuals:  
 Min 1Q Median 3Q Max   
-3.10208 -0.92239 -0.06399 1.04280 3.11900   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 1.165e+01 7.863e-01 14.814 1.59e-11 \*\*\*  
schaef[1:20, "catch"] -2.576e-05 6.055e-06 -4.255 0.000477 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 1.495 on 18 degrees of freedom  
Multiple R-squared: 0.5014, Adjusted R-squared: 0.4737   
F-statistic: 18.1 on 1 and 18 DF, p-value: 0.0004765

## 7.2 一些方程

使用相对丰度指数来描述评估种群的动态。该指数无论如何得到，都假定其反映了用于估算该指数的方法（渔业相关的 cpue 或独立调查）所能获得的生物量，并且假定该生物量受到捕捞渔获量的影响。这意味着，如果我们使用商业单位努力量渔获量（cpue），严格来说，我们处理的是可开发生物量，而不是繁殖生物量（这是种群评估更常见的目标）。不过，一般假设捕捞的选择性接近成熟度曲线，因此所使用的指数仍是繁殖生物量指数，至少是近似指数。即便如此，在得出这样的结论之前，仍应明确考虑究竟指的是什么。

一般来讲，动态变化是指年 起始时的生物量方程，尽管根据定义，它可以指一年中的不同日期。请记住，一年的结束日期与下一年的起始日期实际上是相同的，但具体使用哪个日期会影响分析的开始和结束（例如，从哪个生物量年去除特定年份的渔获量）：

其中 为数据开始时的初始生物量。如果数据从捕捞开始就有，那么 ， 是承载能力或未捕捞时的生物量。 表示 年初的种群生物量， 表示种群的内禀生长率，以及 表示种群生物量的生产函数，该函数考虑了新个体的补充、现存个体生物量的任何增长、自然死亡率，并假设密度对种群增长率的线性影响。最后的项， 是年 的渔获量，表示了捕捞死亡率。每一项所指的年份非常重要，因为它决定了方程中的动态建模方式以及随后的 R 代码。

为了将这种评估模型的动态与现实世界进行比较和拟合，还利用模型动态生成每年相对丰度指数的预测值：

式中 是年 相对丰度指数的预测或估计平均值，与观测到的指数进行比较，使模型与数据相匹配。 是年 的捕捞努力量， 是可捕系数（定义为生物量/单位努力量的渔获量）。这种关系还提出了一个强有力的假设，即种群生物量就是所谓的动态库。这意味着，无论地理距离如何，渔业或环境对种群动态的任何影响都会在所用的每个时间段内（通常为一年，但可能更短）对整个种群产生影响。这是一个强有力的假设，特别是如果一个种群中出现任何一致的空间结构，或者渔业的地理规模使得一个区域的鱼需要大量时间才能到达另一个区域。同样，需要了解这些假设，才能理解其局限性并适当地解释任何此类分析。

### 7.2.1 产量方程

已经提出了大量的方程形式来描述种群的生产力以及如何响应资源量。我们将考虑两个形式，即 Schaefer 模型和 Fox （1970） 模型的修改形式，以及包含这两种模型的概括：

Schaefer（1954,1957）模型的产量方程为：

而 Fox(1970) 模型的修改版本使用：

该修改版本将 作为第一项，其作用仅仅是将最大生产率保持在 Schaefer 模型中类似参数大致相当的水平。

Pella和Tomlinson（1969）提出了一个广义产量方程，其中包括了 Schaefer 和 Fox 模型作为特例的情况。在此，我们将使用 Polacheck, Hilborn, 和 Punt (1993) 提出的替代公式，它提供了种群动态的一般方程，可用于 Schaefer 和 Fox 模型，以及两者之间的渐变，具体取决于单个参数 的值。

其中第一项，，是时间 的种群生物量，最后一项 ，是时间 的渔获量。中间项是比较复杂的部分，定义了产量曲线。它由种群的瞬时增长率 、时间 的现存生物量 、环境容纳量或最大种群数量 ，以及控制产量曲线任何不对称性的项 组成。如果将 设置为 1.0（**MQMF** 中 discretelogistic() 函数的默认值），该公式简化为经典的 Schaefer 模型 （Schaefer， 1954， 1957）。Polacheck, Hilborn, 和 Punt (1993) 引入了上述公式，但往往称之为 Pella-Tomlinson（1969）剩余产量模型（尽管他们的公式不同，但具有非常相似的性质）。

子项 表示不受约束的指数种群增长（因为在该差分方程中，将其加到 中），只要 ，将导致在没有渔获物的情况下种群持续正增长（尽管 14 世纪的瘟疫曾在一段短暂但特别不愉快的时期内扭转了这一趋势，但世界人口的正指数增长仍然是这一趋势的例证）。子项 为指数增长项提供了约束条件，因为随着种群数量的增加，指数增长项的值趋于零。这被称为密度依赖（density-dependent）效应。

当将 设置为 时，该公式将与 Schaefer 模型相同（线性密度依赖性）。但是当 设置为一个很小的数值时，比如 ，那么公式就等价于 Fox 模型的动力学公式。 的值会导致生产曲线向左倾斜，而模位于中心偏右。当 或 时，密度依赖性将不再是线性的。一般我们会固定 值，而不会尝试使用数据来拟合。仅凭渔获量和相对丰度指数通常不足以估算相对种群数量对产量的详细影响。

# plot productivity and density-dependence functions Fig7.4  
prodfun <- function(r, Bt, K, p) {  
 return((r \* Bt / p) \* (1 - (Bt / K)^p))  
}  
densdep <- function(Bt, K, p) {  
 return((1 / p) \* (1 - (Bt / K)^p))  
}  
r <- 0.75  
K <- 1000.0  
Bt <- 1:1000  
sp <- prodfun(r, Bt, K, 1.0) # Schaefer equivalent  
sp0 <- prodfun(r, Bt, K, p = 1e-08) # Fox equivalent  
sp3 <- prodfun(r, Bt, K, 3) # left skewed production, marine mammal?  
parset(plots = c(2, 1), margin = c(0.35, 0.4, 0.1, 0.05))  
plot1(Bt, sp,  
 type = "l", lwd = 2, xlab = "Stock Size",  
 ylab = "Surplus Production", maxy = 200, defpar = FALSE  
)  
lines(Bt, sp0 \* (max(sp) / max(sp0)), lwd = 2, col = 2, lty = 2) # rescale  
lines(Bt, sp3 \* (max(sp) / max(sp3)), lwd = 3, col = 3, lty = 3) # production  
legend(275, 100, cex = 1.1, lty = 1:3, c(  
 "p = 1.0 Schaefer", "p = 1e-08 Fox",  
 "p = 3 LeftSkewed"  
), col = c(1, 2, 3), lwd = 3, bty = "n")  
plot1(Bt, densdep(Bt, K, p = 1),  
 xlab = "Stock Size", defpar = FALSE,  
 ylab = "Density-Dependence", maxy = 2.5, lwd = 2  
)  
lines(Bt, densdep(Bt, K, 1e-08), lwd = 2, col = 2, lty = 2)  
lines(Bt, densdep(Bt, K, 3), lwd = 3, col = 3, lty = 3)

|  |
| --- |
| 图 7.4: p 参数对 Polacheck 等（1993） 生产函数的影响（上图）和对密度依赖项的影响（下图）。注意生产力的重新缩放，以符合 Schaefer 曲线的结果。种群规模可以是生物量或数量。 |

Schaefer 模型假设生产曲线对称，在 处具有最大剩余生产或最大持续产量（MSY），密度依赖项的线性变化趋势是，当种群数量非常小时密度依赖项为1.0，当 趋于 时密度依赖项为0 。当 的值很小时，比如说 ，该模型近似为Fox 模型，这会产生一条不对称的生产曲线，在某个较低的消耗水平下（本例中使用 bt[which.max(sp0)] 发现为 ,），会形成最大产量。密度相关项是非线性的，最大持续产量（MSY）出现在密度相关项 = 1.0 的地方。如果不对[图 7.4](#fig-spm4) 中进行重新缩放，Fox模型通常比 Schaefer 模型更有效率，因为当种群数量低于最大持续产量生物量 时，密度相关项在大于1.0。 时，最大产量出现在较高的种群数量处，当种群数量较低时，种群增长率几乎呈线性增长，而只有在种群数量相当大时，才会出现密度依赖下降的情况。只发生在相当高的种群水平上。与鱼类相比，这种动态在海洋哺乳动物中更为典型。

可以认为 Schaefer 模型比 Fox 模型更保守，因为它需要更高的种群数量才能达到最大产量，并且通常会导致较低水平的渔获量，由于 Fox 类型模型的产量通常较高，可能会出现例外情况。

### 7.2.2 Schaefer 模型

对于 Schaefer 模型，我们通过设置 得到：

给定渔业数据的时间序列，总会有一个初始生物量，可能是 或 是 的某个分数，取决于在首次获得渔业数据时是否认为该种群已经枯竭。 不可能高于 ，因为实际种群往往不会表现出稳定的平衡。

根据数据拟合模型至少需要3个参数，即 、 、可捕系数 （可能还需要 ）。但是，可以使用所谓的“封闭形式”方法来估计可捕获性系数 :

即观测到的渔获量除以预测的可开发生物量的反演几何平均数 (Polacheck, Hilborn, 和 Punt 1993)。这样就得到了时间序列的平均可捕量。如果渔业发生了重大变化，CPUE 的质量也发生了变化，则有可能对时间序列的不同部分产生不同的可捕量估计值。然而，需要注意为这种建议的模型规格进行有力的辩护，特别是用于估算的时间序列越 短，就越需要注意。 的不确定性就越大。

### 7.2.3 残差平方和

该模型可以使用最小二乘法进行拟合，或者更准确地说，可以使用残差误差的平方和进行拟合：

由于 CPUE 一般呈对数正态分布，而最小二乘法意味着正态随机误差，因此需要进行对数变换。最小二乘法在首先寻找一组参数，使模型与现有数据相匹配时，往往相对稳健。然而，一旦接近解决方案，如果使用最大似然法，就会有更多建模选择。全对数正态对数似然为：

除了对数变换之外，这与插在 项之前相关变量（此处为 ）的正态 PDF 似然不同。幸运的是，如第 [4](#sec-paraestimat) 章”模型参数估算“ 所示，可以对负对数似然简化 (Haddon 2011)，变为：

其中，标准差（）的最大似然估计，由下式得到：

注意除以 而不是*除以* 。严格来讲，对于对数正态（[公式 7.10](#eq-spm10) 中）， 后面应该跟着一个附加项：

对数转换后的观测捕获率之和。但是，由于该项是恒量，因此通常会省略。当然，当使用 R 时，我们总是可以使用内置的概率密度函数实现（参见 negLL() 和 negLL1() ），因此这种简化并不是绝对必要的，但是当人们希望使用 **Rcpp** 加快分析速度时，它们仍然有用，尽管 *Rcpp-syntactic-sugar*，导致 C++ 代码看起来非常像 R 代码，现在包括 dnorm() 版本和相关的分布函数。

### 7.2.4 估算管理统计

只需使用以下方法即可计算 Schaefer 模型的最大可持续产量：

然而，对于使用 (Polacheck, Hilborn, 和 Punt 1993) 的 *p* 参数的更一般方程，需要使用：

当 时，上式可以简化为[公式 7.13](#eq-spm13)。我们可以使用 **MQMF** 函数getMSY() 来计算 [公式 7.14](#eq-spm14) ，这也说明了Fox 模型比 Schaefer 模型具有更高的生产力。

# compare Schaefer and Fox MSY estimates for same parameters  
param <- c(r = 1.1, K = 1000.0, Binit = 800.0, sigma = 0.075)  
cat("MSY Schaefer = ", getMSY(param, p = 1.0), "\n") # p=1 is default

MSY Schaefer = 275

cat("MSY Fox = ", getMSY(param, p = 1e-08), "\n")

MSY Fox = 404.6674

当然，如果将两个模型与实际数据进行拟合，通常会为每个模型生成不同的参数，因此得到的 MSY 值可能更接近。

还可以生成基于努力量的管理统计数据。如果努力量水平一直持续下去，使种群达到平衡时*MSY* 的努力水平称为 :

其中Schaefer模型中 时种群将衰竭，但对于参数 *p* 的其他值仍具有普遍性。也可以估计平衡渔获率（每年捕获的种群比例），从而得到 ，即MSY*剩余*产量时的生物量：

可以经常看到将[公式 7.16](#eq-spm16) 写为 ，但这可能会产生误导，因为通常将 解释为瞬时捕捞死亡率，而在这种情况下，实际上是成比例的捕捞率。出于这个原因，我明确使用了 。

### 7.2.5 均衡的麻烦

现实世界中对管理目标的解释并不总是直截了当的。现在，人们认为大多数捕捞种群不可能达到平衡，因此，如果种群以最佳方式捕捞，MSY 的解释更像是平均的、长期的预期潜在产量；动态平衡可能是更好的描述。如果一直应用， 是到得MSY的努力量，但前提是种群生物量达到 ，即产生最大剩余产量的生物量。每种管理统计都源自均衡思想。显然，应当通过将努力量限制在 处来管理渔业，但如果种群生物量开始严重枯竭，那么将不会产生平均长期产量。事实上， 努力量程度可能太高，无法在这个非平衡的世界中重建种群。同样地，但仅当种群生物量为 的情况下， 将按预期运行。可以估计导致种群恢复到 的渔获量或努力量水平，可以称之为 ，但这需要进行种群预测并寻找最终达到预期结果的渔获量水平。我们将在后面的章节中研究如何预测种群。

需要强调的是，MSY观点及其相关统计是以平衡思想为基础的，在现实世界中是罕见的。充其量，动态平衡是可以实现的，但无论如何，使用这种平衡统计都存在风险。当首次提出时，大家者认为 𝑀𝑆𝑌 概念是管理渔业的合适目标。现在，尽管这一概念已作为渔业管理的总体目标纳入了一些国家渔业法案和法律，但更安全的做法是将 𝑀𝑆𝑌 作为捕捞死亡率（渔获量）的上限，是一个极限参考点，而不是目标参考点。

理想情况下，评估结果需要通过捕捞控制规则（HCR）传递，该规则根据估计的种群状况（捕捞死亡率和种群枯竭水平）就未来渔获量或努力量提供正式的管理建议。然而，如果不对其数值的不确定性有所了解，这些潜在的管理产出中几乎没有价值。正如我们已经指出的那样，能够将这些模型预测到未来，以便对替代管理战略进行风险评估，也将非常有用。但首先，我们需要利用模型拟合数据。

## 7.3 模型拟合

模型参数和与模型相关的其他详细信息也可以在每个函数的帮助文件中找到（试着使用 ?spm 或 simpspm ）。简而言之，模型参数 为种群净增长率（综合了重量、补充和自然死亡率等方面的个体生长）， 为种群环境容纳量或未捕捞时的生物量， 是第一年的生物量。只有当相对丰度数据指数（通常是 cpue）在渔业实施了几年后才可获得，这意味着种群已经在某种程度上枯竭时，才需要此参数。如果假设没有初始损耗，则参数列表中不需要 ，并且在函数中设置为等于 。最后一个参数是 ，即用于描述残差的对数正态分布的标准差。为了用最大似然法而编写了 simpspm() 和 spm() ，因此即使使用 ssq() 作为最佳拟合标准，参数向量中也需要 值。

在澳大利亚，相对种群丰度指数通常是单位努力量渔获量（cpue），但也可以是一些与渔业无关的丰度指数（例如，来自拖网调查、声学调查），或者在某一分析中都使用两者（见 simspm()）。通过分析，可以提出持续管理的产量建议以及确定种群状况。

在本节中，我们将详细介绍如何进行剩余产量分析、如何从分析中提取结果以及如何绘制这些结果的图解。

### 7.3.1 种群评估的可能工作流程

根据剩余产量模型进行种群评估时，可能的工作流程包括:

1. 读取渔获量和相对丰度数据的时间序列。拥有检查数据完整性、缺失值和其他潜在问题的功能会有所帮助，但最好还是了解自己的数据及其局限性。
2. 使用ccf() 分析以确定 cpue 数据相对于渔获量数据是否具有参考价值。 如果发现明显的负相关关系，这将增强分析的防御力。
3. 定义/估计初始参数集，包括了 和 ，以及可选 初始生物量，如果怀疑渔业数据是在种群某种程度上消耗后才开始的，则使用该值。
4. 使用函数 plotspmmod() 绘制假定的初始参数集对动力学的影响。这在寻找可信的初始参数集时非常有用。
5. 使用nlm() 或 fitSPM() 在输入可能可行的初始参数集后，搜索最佳参数。参见讨论。
6. 使用plotspmmod() 用最佳参数来说明最佳模型及其相对拟合的影响（尤其是使用残差图）。
7. 理想情况下，应通过使用多个不同的初始参数集作为模型拟合过程的起点来检查模型拟合的鲁棒性，见 robustSPM()
8. 一旦对模型拟合的鲁棒性感到满意，就可以使用spmphaseplot() 绘制出生物量与渔获率的相图，以便直观地确定和说明种群状况。
9. 使用 spmboot()，用渐近标准误差或 *On Uncertainty* （第 [6](#sec-uncertainty) 章）中的贝叶斯方法来描述模型拟合和输出中的不确定性。将此类输出制成表格并绘制成图表。请参阅稍后内容。
10. 记录并捍卫得出的任何结论。

目前 **MQMF** 有两种常见的动力模型：经典的 Schaefer 模型（Schaefer，1954）和近似的 Fox 模型（Fox，1970；Polacheck等，1993），Haddon（2011）对两种模型都有描述。Prager（1994）提供了许多其他形式的分析，这些分析可以使用剩余产量模型进行，Haddon（2011）也提供了实际应用。

# Initial model 'fit' to the initial parameter guess Fig 7.5  
data(schaef)  
schaef <- as.matrix(schaef)  
param <- log(c(r = 0.1, K = 2250000, Binit = 2250000, sigma = 0.5))  
negatL <- negLL(param, simpspm, schaef, logobs = log(schaef[, "cpue"]))  
ans <- plotspmmod(  
 inp = param, indat = schaef, schaefer = TRUE,  
 addrmse = TRUE, plotprod = FALSE  
)

|  |
| --- |
| 图 7.5: 使用初始参数值将剩余生产模型与 schaef 数据集的暂定拟合。CPUE 图中的绿色虚线是简单的loess 拟合，而实线是猜测的输入参数所隐含的拟合。渔获量图中的水平红线是预测的 MSY。残差图中的数字是对数正态残差的均方根误差。 |

时得到 ，并且在 1950 年之前所有残差都低于1.0，之后有 4 个较大的正残差。将 值设置为最大渔获量的10倍左右，这一数量级（10倍到20倍）的生物量通常会得到足够的生物量，使种群生物量和CPUE轨迹偏离x轴，以便进入最小化/优化程序。我们使用了plotprod = FALSE 选项（默认值），因为在用模型拟合数据之前，查看预测的产量曲线几乎没有意义。

用数值方法拟合数据模型时，通常需要采取措施确保获得稳健的且生物学上合理的拟合模型。稳健性的一个方法是对模型拟合两次，第二次拟合的输入参数来自第一次拟合。我们将使用optim() 和 nlm()以及negLL1() 的组合来估计每次迭代期间的负对数似然（这是fitSPM()实现的方式）。在**MQMF**中，我们有一个函数 spm() ，根据生物量、CPUE、消耗和渔获率的预测变化来计算所有的动态变化。虽然这样做相对较快，但为了加快迭代模型拟合过程，我们未使用spm()，而是使用simspm() ，仅输出预测的 CPUE 的对数，以便最小化，而不是每次都计算完整的动态。当我们只有相对丰度指数的单一时间序列时，我们使用simspm() 。如果我们有多个指数序列，我们将使用 simpsmpM() 、spmCE() 以及negLLM() ；参阅帮助文件（?simpsmpM、?spmCE、?negLLM ）及其代码，查阅每个示例的运行情况。除了使用 simpsmpM() 、spmCE() 和 negLLM() 外，对于多个时间序列的指数，它还用于说明模型拟合有时会产生生物学上难以置信的解决方案，便在数学上却是最优的解。以及第一个参数 施加惩罚，以防止其小于0.0，在多指数函数示例中使用了极端渔获量历史记录，根据起始参数，我们还需要对年渔获率进行惩罚，以确保其保持小于 1.0（见 penalty1()）。从生物学角度看，渔获量显然不可能超过生物量，但如果我们不对模型进行数学限制，那么渔获率非常大在数学上也没有什么问题。

随着我们所使用模型的复杂性增加，或者我们开始使用计算机密集型方法，对速度的考虑就变得更加重要。我们的参数都不应该变为负值，而且它们的大小差别很大，因此我们在这里使用的是自然对数转换的参数。

# Fit the model first using optim then nlm in sequence  
param <- log(c(0.1, 2250000, 2250000, 0.5))  
pnams <- c("r", "K", "Binit", "sigma")  
best <- optim(  
 par = param, fn = negLL, funk = simpspm, indat = schaef,  
 logobs = log(schaef[, "cpue"]), method = "BFGS"  
)  
outfit(best, digits = 4, title = "Optim", parnames = pnams)

optim solution: Optim   
minimum : -7.934055   
iterations : 41 19 iterations, gradient  
code : 0   
 par transpar  
r -1.448503 0.2349  
K 14.560701 2106842.7734  
Binit 14.629939 2257885.3255  
sigma -1.779578 0.1687  
message :

cat("\n")

best2 <- nlm(negLL, best$par,  
 funk = simpspm, indat = schaef,  
 logobs = log(schaef[, "cpue"])  
)  
outfit(best2, digits = 4, title = "nlm", parnames = pnams)

nlm solution: nlm   
minimum : -7.934055   
iterations : 2   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient transpar  
r -1.448508 6.030001e-04 0.2349  
K 14.560692 -2.007053e-04 2106824.2701  
Binit 14.629939 2.545064e-04 2257884.5480  
sigma -1.779578 -3.688185e-05 0.1687

数值优化程序两次应用的输出结果表明，我们不需要进行两次处理，但为了谨慎起见，还是不要太相信数值方法。无论如何都要进行单一模型拟合，但要自担风险（或许我不得不比许多人处理更多质量较差或一般的数据！）。

现在，我们可以从 *best2* 拟合中获取最佳参数，并将其输入plotspmmod()函数中，可直观显示模型拟合结果。这样我们获得了最佳参数，因此可以通过将 *plotprod* 参数设置为 *TRUE* 来绘制包含生产力曲线。 plotspmmod() 的作用不仅仅是图示结果，还隐形返回一个大的列表对象，因此，如果我们想要它，就需要将其赋值给变量或对象（在本例中*为 ans*）。

# optimum fit. Defaults used in plotprod and schaefer Fig 7.6  
ans <- plotspmmod(  
 inp = best2$estimate, indat = schaef, addrmse = TRUE,  
 plotprod = TRUE  
)

|  |
| --- |
| 图 7.6: 根据 nlm() 最终拟合的最优参数，将剩余生产模型与 schaef 数据集进行拟合的摘要。在 CPUE 图中，绿色虚线为简单的黄土曲线拟合，红色实线为最优模型拟合。 |

plotspmmod()返回的对象是包含结果集合的对象列表，包括最优参数、包含预测最优动态预测的矩阵 （*ans$Dynamics$outmat*）、产量曲线和大量汇总结果。一旦分配给工作环境中的特定对象，就可以快速提取这些对象以用于其他函数。在不使用 *max.level=1* 参数或将其设置为2的情况下尝试运行 str() ，以查看更多详细信息。很多函数会生成大量信息丰富的对象，您应该熟悉探索这些对象，以确保了解不同分析中产生的结果。

# the high-level structure of ans; try str(ans$Dynamics)  
str(ans, width = 65, strict.width = "cut", max.level = 1)

List of 12  
 $ Dynamics :List of 5  
 $ BiomProd : num [1:200, 1:2] 100 10687 21273 31860 42446 ...  
 ..- attr(\*, "dimnames")=List of 2  
 $ rmseresid: num 1.03  
 $ MSY : num 123731  
 $ Bmsy : num 1048169  
 $ Dmsy : num 0.498  
 $ Blim : num 423562  
 $ Btarg : num 1016409  
 $ Ctarg : num 123581  
 $ Dcurr : Named num 0.528  
 ..- attr(\*, "names")= chr "1956"  
 $ rmse :List of 1  
 $ sigma : num 0.169

还有一些 **MQMF** 函数可以帮助提取此类结果或使用 plotspmmod() 的结果（参见 summspm() 和 spmphaseplot() ），这就是为什么该函数包含参数 *plotout = TRUE，*因此不需要生成绘图。但是，在许多情况下，只需在高级对象（在本例中*为 ans*）中指向所需的对象即可。请注意，从生成的生产率曲线获得的 MSY 与从最优参数计算得出的 MSY 相差很小。这是因为生产力曲线是通过计算不同生物量水平向量的生产率数值得出的。因此，其分辨率受到用于生成生物量向量的步骤限制。其估计值将始终略小于参数派生值。

# compare the parameteric MSY with the numerical MSY  
round(ans$Dynamics$sumout, 3)

msy p FinalDepl InitDepl FinalB   
 123734.068 1.000 0.528 1.072 1113328.480

cat("\n Productivity Statistics \n")

Productivity Statistics

summspm(ans) # the q parameter needs more significantr digits

Index Statistic  
q 1 0.0000  
MSY 2 123731.0026  
Bmsy 3 1048168.8580  
Dmsy 4 0.4975  
Blim 5 423562.1648  
Btarg 6 1016409.1956  
Ctarg 7 123581.3940  
Dcurr 8 0.5284

最后，为了简化此双模型拟合过程的未来使用，有一个 **MQMF** 函数 fitSPM() 来实现该过程。您可以使用该函数（查看其代码等），也可以重复原始代码的内容，以方便使用。

### 7.3.2 分析是否稳健？

尽管我多次警告，但您可能想知道为什么我们要费力地对模型进行两次拟合，而第二次拟合的起点就是第一次拟合的最优估计值。我们应该始终记住，在拟合这些模型时，我们使用的是数值方法。这种方法并非万无一失，可能会发现错误的最小值。如果模型参数之间存在相互作用或相关性，那么稍有不同的组合就会导致非常相似的负对数似然值。最佳模型拟合在 cpue 时间序列的末尾仍显示出三个相对较大的残差，如 [图 7.6](#fig-spm6) 。这些残差没有表现出任何特定的模式，因此我们认为它们只代表不确定性，这应该让人怀疑模型拟合的好坏以及分析输出统计量的可靠性。我们可以通过检查初始模型参数对模型拟合的影响来检查模型拟合的稳健性。

稳健性测试的一种实现使用 **MQMF** 函数 robustSPM()。 该函数将生成 𝑁 个随机初始值，是通过将最佳对数刻度参数值作为某些正态随机变量的相应平均值，其各自的标准差值通过将这些均值除以*缩放器*参数值获得（有关完整详细信息，请参阅 robustSPM()代码和帮助）。robustSPM() 输出的对象包括 *N* 随机变化的初始参数值的向量，这允许说明和表征它们的变化。当然，作为除数，*标度值*越小，初始参数向量的可变性就越大，也就越容易导致模型拟合无法找到最小值。

# conduct a robustness test on the Schaefer model fit  
data(schaef)  
schaef <- as.matrix(schaef)  
reps <- 12  
param <- log(c(r = 0.15, K = 2250000, Binit = 2250000, sigma = 0.5))  
ansS <- fitSPM(  
 pars = param, fish = schaef, schaefer = TRUE, # use  
 maxiter = 1000, funk = simpspm, funkone = FALSE  
) # fitSPM  
# getseed() #generates random seed for repeatable results  
set.seed(777852) # sets random number generator with a known seed  
robout <- robustSPM(  
 inpar = ansS$estimate, fish = schaef, N = reps,  
 scaler = 40, verbose = FALSE, schaefer = TRUE,  
 funk = simpspm, funkone = FALSE  
)  
# use str(robout) to see the components included in the output

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | ir | iK | iBinit | isigma | iLike | r | | --- | --- | --- | --- | --- | --- | --- | | 6 | 0.232 | 2521208 | 2394188 | 0.173 | -5.765 | 0.235 | | 10 | 0.242 | 2564306 | 1386181 | 0.166 | 14.306 | 0.235 | | 11 | 0.237 | 2189281 | 2032237 | 0.181 | -7.025 | 0.235 | | 1 | 0.239 | 2351319 | 3401753 | 0.169 | -6.351 | 0.235 | | 8 | 0.244 | 2201215 | 2934055 | 0.180 | -7.078 | 0.235 | | 3 | 0.233 | 3164529 | 1632687 | 0.170 | 22.093 | 0.235 | | 12 | 0.237 | 3492106 | 1895315 | 0.165 | 23.789 | 0.235 | | 2 | 0.247 | 2359029 | 2137751 | 0.179 | -5.575 | 0.235 | | 5 | 0.234 | 3057512 | 1502916 | 0.171 | 23.720 | 0.235 | | 7 | 0.242 | 1671149 | 2512111 | 0.169 | 4.228 | 0.235 | | 4 | 0.233 | 3482370 | 1584633 | 0.168 | 34.534 | 0.235 | | 9 | 0.230 | 1391893 | 1753155 | 0.175 | 138.808 | 0.235 | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | K | Binit | sigma | -veLL | MSY | Iters | | --- | --- | --- | --- | --- | --- | --- | | 6 | 2107069 | 2258144 | 0.169 | -7.934 | 123724.7 | 5 | | 10 | 2107034 | 2258103 | 0.169 | -7.934 | 123726.0 | 8 | | 11 | 2107243 | 2258322 | 0.169 | -7.934 | 123717.4 | 7 | | 1 | 2107178 | 2258293 | 0.169 | -7.934 | 123721.9 | 17 | | 8 | 2107119 | 2258218 | 0.169 | -7.934 | 123720.1 | 4 | | 3 | 2107386 | 2258484 | 0.169 | -7.934 | 123713.3 | 4 | | 12 | 2107417 | 2258533 | 0.169 | -7.934 | 123712.5 | 27 | | 2 | 2106866 | 2257912 | 0.169 | -7.934 | 123728.2 | 6 | | 5 | 2107294 | 2258319 | 0.169 | -7.934 | 123712.7 | 10 | | 7 | 2107319 | 2258401 | 0.169 | -7.934 | 123712.2 | 9 | | 4 | 2106799 | 2257706 | 0.169 | -7.934 | 123732.1 | 28 | | 9 | 2106435 | 2257279 | 0.169 | -7.934 | 123739.1 | 30 | |

表 7.2: A robustness test of the fit to the schaef data-set. By examining the results object we can see the individual variation. The top columns relate to the initial parameters and the bottom columns, perhaps of more interest, to the model fits.

通过使用 *set.seed* 函数，用于生成分散初始参数向量的伪随机数的结果是可重复的。在 [表 7.2](#tbl-spm2) 中，我们可以看到，在 12 次试验中，我们得到了 12 个相同的最终负对数似然（精确到小数点后 5 位），尽管与实际的 、 和 略有不同，这导致估计的 MSY 值的微小变化。如果我们增加试验的次数，最终会发现一些试验结果与最佳结果略有不同。

通常情况下，我们会尝试 12 次以上的试验，并检查标度参数的效果。因此，我们现在将使用相同的最佳拟合和随机种子重复该分析100次。robustSPM()输出*结果*表按最终的 -ve 对数似然排序，但即使是相同的小数点后五位，也会发现参数估计值略有不同。这只是使用数值方法的反映。

# Repeat robustness test on fit to schaef data 100 times  
set.seed(777854)  
robout2 <- robustSPM(  
 inpar = ansS$estimate, fish = schaef, N = 100,  
 scaler = 25, verbose = FALSE, schaefer = TRUE,  
 funk = simpspm, funkone = TRUE, steptol = 1e-06  
)  
lastbits <- tail(robout2$results[, 6:11], 10)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7.3: The last 10 trials from the 100 illustrating that the last three trials deviated a little from the optimum negative log-likelihood of -7.93406.   | r | K | Binit | sigma | -veLL | MSY | | --- | --- | --- | --- | --- | --- | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105553 | 2256358 | 0.169 | -7.934 | 123770.0 | | 0.235 | 2105527 | 2256328 | 0.169 | -7.934 | 123770.8 | | 0.235 | 2105510 | 2256327 | 0.169 | -7.934 | 123772.4 | |

[表 7.3](#tbl-spm3) 只列出了排序后 100 个重复样本中的后 10 条记录，这表明所有重复样本的负对数似然值相同（精确到小数点后 5 位）。同样，如果仔细观察 、 、 和 MSY 的值，就会发现它们之间的差异。如果我们将最终拟合的参数值用图显示，变化的标度就会很明显，如[图 7.7](#fig-spm7)。

# replicates from the robustness test Fig 7.7  
result <- robout2$results  
parset(plots = c(2, 2), margin = c(0.35, 0.45, 0.05, 0.05))  
hist(result[, "r"], breaks = 15, col = 2, main = "", xlab = "r")  
hist(result[, "K"], breaks = 15, col = 2, main = "", xlab = "K")  
hist(result[, "Binit"], breaks = 15, col = 2, main = "", xlab = "Binit")  
hist(result[, "MSY"], breaks = 15, col = 2, main = "", xlab = "MSY")

|  |
| --- |
| 图 7.7: 对 schaef 数据集模型拟合的稳健性测试中 100 次试验的主要参数和 MSY 的直方图。参数估计值都很接近，但仍存在差异，这反映了估计的不确定性。为了改善这种情况，可以尝试使用较小的 steptol，默认值为 1e-06，但并不总是能得到稳定的解决方案。如果使用 steptol = 1e-07，整个变量的取值范围会变得更小，但仍会有一些微小的变化，这也是使用数值方法时的预期结果。这也是为什么参数估计的特定值在我们也有变化或不确定性估计值时最有意义的另一个原因。 |

即使负对数似然值非常接近（精确到小数点后五位）（[图 7.7](#fig-spm7)），也可能与最常出现的最佳值略有偏差。这强调了仔细检查分析中的不确定性的必要性。鉴于大多数试验得出相同的最佳值，所有试验的中值可以确定最佳值。

另一种可视化稳健性检验参数估计值最终变化的方法是使用 R 函数 pairs() 绘制各参数与模型输出值的对比图，如 [图 7.8](#fig-spm8) 所示，该图说明了参数之间的强相关性。

# robustSPM parameters against each other Fig 7.8  
pairs(result[, c("r", "K", "Binit", "MSY")], upper.panel = NULL, pch = 1)

|  |
| --- |
| 图 7.8: 100 个最优解决方案中参数之间的关系图，这些解源于将剩余生产模型拟合到 *schaef* 数据集。参数之间的相关性是显而易见的，尽管需要强调的是，估计值之间的比例差异非常小，约为0.2-0.3%。 |

### 7.3.3 使用不同的数据？

*schaef* 数据集得出的结果相对稳健。在继续分析之前，使用 *dataspm* 数据集进行重复分析会更有启发，因为该数据集的结果更多变。希望这些发现能鼓励今后的建模者阅读本文时，不要相信数值优化程序给出的第一个解决方案。

# Now use the dataspm data-set, which is noisier  
set.seed(777854) # other random seeds give different results  
data(dataspm)  
fish <- dataspm # to generalize the code  
param <- log(c(r = 0.24, K = 5174, Binit = 2846, sigma = 0.164))  
ans <- fitSPM(  
 pars = param, fish = fish, schaefer = TRUE, maxiter = 1000,  
 funkone = TRUE  
)  
out <- robustSPM(ans$estimate, fish,  
 N = 100, scaler = 15, # making  
 verbose = FALSE, funkone = TRUE  
) # scaler=10 gives  
result <- tail(out$results[, 6:11], 10) # 16 sub-optimal results

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7.4: The last 10 trials from 100 used with dataspm. The last six trials deviate markedly from the optimum negative log-likelihood of -12.1288, and five gave consistent sub-optimal optima. Variation across parameter estimates with the optimum log-likelihood remained minor, but was large for the false optima.   | r | K | Binit | sigma | -veLL | MSY | | --- | --- | --- | --- | --- | --- | | 0.243 | 5,171.268 | 2,844.290 | 0.164 | -12.129 | 313.537 | | 0.243 | 5,171.509 | 2,843.699 | 0.164 | -12.129 | 313.528 | | 0.243 | 5,171.805 | 2,846.729 | 0.164 | -12.129 | 313.545 | | 0.243 | 5,169.358 | 2,842.830 | 0.164 | -12.129 | 313.555 | | 3.561 | 149.623 | 50.741 | 0.223 | -2.524 | 133.201 | | 0.032 | 36,059.563 | 49.720 | 0.233 | -1.178 | 289.163 | | 40.310 | 0.257 | 49.720 | 0.233 | -1.178 | 2.592 | | 22.194 | 0.003 | 49.720 | 0.233 | -1.178 | 0.016 | | 1.186 | 6,062.637 | 49.720 | 0.233 | -1.178 | 1,797.041 | | 0.595 | 4,058.965 | 49.720 | 0.233 | -1.178 | 604.180 | |

在与 *dataspm* 进行拟合的底部六个模型中，我们可以看到 值非常大而 值非常小的情况，以及 值非常大而 值非常小的情况，此外，在最后两行中， 和 的值几乎是合理的，但 *Binit* 值却非常小。

## 7.4 不确定性

当我们测试一些模型拟合对初始条件的稳健性时，我们发现当拟合多个参数时，可以从略微不同的参数值中获得基本相同的数值拟合（达到给定的精度）。虽然这些值往往相差不大，但这一观察结果仍然证实，当使用数值方法估计一组参数时，特定参数值并不是唯一重要的结果。我们还需要知道这些估计的精确程度，我们需要知道与它们的估计相关的任何不确定性。有许多方法可以用来探索模型拟合中的不确定性。在这里，我们将使用 R 检查四个的实现：1）似然剖面，2）自举重采样，3） 渐近误差，以及 4）贝叶斯后验分布。

### 7.4.1 似然剖面图

似然剖面顾名思义，就是让人了解如果使用的参数稍有不同，模型拟合的质量会发生怎样的变化。使用最大似然法（最小化对数似然）对模型进行最佳拟合，然后在将一个或多个参数固定为预定值（保持不变）的同时，只对其余未固定的参数进行拟合。这样，在给定一个或多个参数固定值的情况下，就可以获得最佳拟合结果。因此，我们可以确定当所选参数在一系列不同值上保持固定时，模型拟合的总可能性将如何变化。通过一个实例，我们可以更清楚地了解这一过程。我们可以使用 abdat 数据集，该数据集对观测数据进行了合理拟合，但最优拟合的残差模式适中，最优解的最终梯度相对较大（尝试 outfit(ans) 查看结果）。

# Fig 7.9 Fit of optimum to the abdat data-set  
data(abdat)  
fish <- as.matrix(abdat)  
colnames(fish) <- tolower(colnames(fish)) # just in case  
pars <- log(c(r = 0.4, K = 9400, Binit = 3400, sigma = 0.05))  
ans <- fitSPM(pars, fish, schaefer = TRUE) # Schaefer  
answer <- plotspmmod(ans$estimate, abdat, schaefer = TRUE, addrmse = TRUE)

|  |
| --- |
| 图 7.9: 描述最佳参数与 *abdat* 数据集拟合的汇总图。拟合和 cpue 数据之间的对数正态残差中的剩余模式如右下角所示。 |

在 *“不确定性”* 一章中，我们研究了围绕单个参数的似然剖面，在这里我们将更深入地探讨与似然剖面的使用相关的一些问题。我们已经有了对 *abdat* 数据集的最佳拟合，可以以此为起点。反过来，如果我们考虑 𝑟 和 𝐾 参数时，编写一个简单的函数拟合每种情形下的剖面会更有效，以避免代码重复。和以前一样，我们使用 negLLP() 函数固定某些参数，同时改变其他参数。如 “*不确定性”* 一章所述，对于一个参数，95% 的置信边界的近似值为一个自由度的最小对数似然加上一个自由度 （=1.92） 的卡方值一半的参数范围。

在绘制每个剖面时，我们可以包括这个阈值，以查看它与似然剖面相交的位置，[图 7.10](#fig-spm10)。

# likelihood profiles for r and K for fit to abdat Fig 7.10  
# doprofile input terms are vector of values, fixed parameter  
# location, starting parameters, and free parameter locations.  
# all other input are assumed to be in the calling environment  
doprofile <- function(val, loc, startest, indat, notfix = c(2:4)) {  
 pname <- c("r", "K", "Binit", "sigma", "-veLL")  
 numv <- length(val)  
 outpar <- matrix(NA, nrow = numv, ncol = 5, dimnames = list(val, pname))  
 for (i in 1:numv) { #  
 param <- log(startest) # reset the parameters  
 param[loc] <- log(val[i]) # insert new fixed value  
 parinit <- param # copy revised parameter vector  
 bestmod <- nlm(  
 f = negLLP, p = param, funk = simpspm, initpar = parinit,  
 indat = indat, logobs = log(indat[, "cpue"]),  
 notfixed = notfix  
 )  
 outpar[i, ] <- c(exp(bestmod$estimate), bestmod$minimum)  
 }  
 return(outpar)  
}  
rval <- seq(0.32, 0.46, 0.001)  
outr <- doprofile(rval,  
 loc = 1, startest = c(rval[1], 11500, 5000, 0.25),  
 indat = fish, notfix = c(2:4)  
)  
Kval <- seq(7200, 11500, 200)  
outk <- doprofile(Kval,  
 loc = 2, c(0.4, 7200, 6500, 0.3), indat = fish,  
 notfix = c(1, 3, 4)  
)  
parset(plots = c(2, 1), cex = 0.85, outmargin = c(0.5, 0.5, 0, 0))  
plotprofile(outr, var = "r", defpar = FALSE, lwd = 2) # MQMF function  
plotprofile(outk, var = "K", defpar = FALSE, lwd = 2)

|  |
| --- |
| 图 7.10: Schaefer 模型的 r 和 K 参数的似然分布与 abdat 数据集拟合。水平红线将最小 -veLL 与界定 95% 置信区间的似然值分开。垂直绿线与最小值和 95% CI 相交。这些数字是围绕平均最佳值的 95% 置信区间。 |

估计这种置信度边界的一个问题是，如果只考虑单个参数，就会忽略参数之间的相互关系和相关性，而 Schaefer 模型对此是众所周知的。但是， 和 参数之间的强相关性意味着，通过组合沿 和 的两个单独的单独搜索而获得的方形网格搜索将导致许多组合甚至超出模型的近似拟合范围。创建二维似然剖面（实际上是曲面）或跨更多参数的剖面并非不可能，但即使是两个参数，通常也需要一次仔细搜索曲面的一小部分，或者以其他方式处理一些极差的模型拟合，这些拟合将通过简单的网格搜索获得。

在资源评估具有一个或多个固定值参数的情况下，跨单个参数的似然分布仍然有用。在 Schaefer 剩余生产模型等简单模型中不会发生这种情况，但在处理更复杂的种群评估模型时，这种情况并不少见，因为生物参数，如自然死亡率、种群招募曲线的陡峭程度，甚至生长参数可能未知或假定其值与相关物种相同。在评估中获得最佳模型拟合后，其中某些参数采用固定值，就可以重新运行模型拟合，同时更改其中一个固定参数的假设值，以生成该参数的似然剖面。这样，就可以看出模型拟合与固定参数的假设值的一致性。以这种方式生成似然剖面比仅仅进行敏感性分析更可取，在敏感性分析中，我们可以将这些固定参数更改为高于假设值的水平和低于假设值的水平，以查看效果。似然剖面提供了对建模对各个参数的敏感性的更详细的探索。

对于更简单的模型，例如我们在这里处理的模型，还有其他方法可以检查建模中固有的不确定性，这些方法可以尝试考虑参数之间的相关性。

### 7.4.2 Bootstrap 置信区间

表征模型拟合不确定性的一种方法是通过对与 cpue 相关的对数正态残差进行自举取样，生成新的 bootstrap cpue 样本来替换原始 cpue 时间序列，从而围绕参数和模型输出（*MSY* 等）生成百分位数置信区间（Haddon，2011）。每次制作这样的自举样本时，都会重新拟合模型并存储解决方案以供进一步分析。要对剩余生产模型进行这样的分析，可以使用 **MQMF** 函数 spmboot()。一旦我们找到了合适的起始参数，我们就可以使用函数 fitSPM() 来获得最佳拟合，并且引导的是与该最佳拟合相关的对数正态残差。在这里，我们将使用噪声相对较大的 *dataspm* 数据集来说明这些观点

#find optimum Schaefer model fit to dataspm data-set Fig 7.11   
data(dataspm)   
fish <- as.matrix(dataspm)   
colnames(fish) <- tolower(colnames(fish))   
pars <- log(c(r=0.25,K=5500,Binit=3000,sigma=0.25))   
ans <- fitSPM(pars,fish,schaefer=TRUE,maxiter=1000) #Schaefer   
answer <- plotspmmod(ans$estimate,fish,schaefer=TRUE,addrmse=TRUE)

|  |
| --- |
| 图 7.11: 描述最佳参数与 *dataspm* 数据集拟合的汇总图。拟合和 cpue 数据之间的对数正态残差如右下角所示。这些是自举的，每个自举样本乘以最佳预测的 cpue 时间序列，以获得每个自举 cpue 时间序列。 |

一旦我们获得了最佳拟合，我们就可以继续进行 bootstrap 分析。通常会运行至少 1000 次重复，甚至更多，即使这可能需要几分钟才能完成。在这种情况下，即使在最佳拟合状态下，对数正态残差中也存在模式，这表明模型结构缺少一些影响渔业的近似周期性事件。

#bootstrap the log-normal residuals from optimum model fit   
set.seed(210368)   
reps <- 1000 # can take 10 sec on a large Desktop. Be patient   
 #startime <- Sys.time() # schaefer=TRUE is the default   
boots <- spmboot(ans$estimate,fishery=fish,iter=reps)   
 #print(Sys.time() - startime) # how long did it take?   
str(boots,max.level=1)

List of 2  
 $ dynam : num [1:1000, 1:31, 1:5] 2846 3555 2459 3020 1865 ...  
 ..- attr(\*, "dimnames")=List of 3  
 $ bootpar: num [1:1000, 1:8] 0.242 0.236 0.192 0.23 0.361 ...  
 ..- attr(\*, "dimnames")=List of 2

输出结果包含每个运行的动态预测模型生物量、每个自举样本的 cpue、每个自举样本的预测 cpue、耗竭时间序列和年收获率时间序列（存储 5 个变量的 31 年运行 次）。每项分析都可用于说明和总结分析结果和不确定性。鉴于 [图 7.11](#fig-spm11) 中的残差相对较大，可以预计不确定性相对较高，见 [表 7.5](#tbl-spm5) 。

#Summarize bootstrapped parameter estimates as quantiles Table 7.6   
bootpar <- boots$bootpar   
rows <- colnames(bootpar)   
columns <- c(c(0.025,0.05,0.5,0.95,0.975),"Mean")   
bootCI <- matrix(NA,nrow=length(rows),ncol=length(columns),   
 dimnames=list(rows,columns))   
for (i in 1:length(rows)) {   
 tmp <- bootpar[,i]   
 qtil <- quantile(tmp,probs=c(0.025,0.05,0.5,0.95,0.975),na.rm=TRUE)   
 bootCI[i,] <- c(qtil,mean(tmp,na.rm=TRUE))   
}

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7.5: The quantiles for the Schaefer model parameters and some model outputs, plus the arithmetic mean. The 0.5 values are the median values.   |  | 0.025 | 0.05 | 0.5 | 0.95 | 0.975 | Mean | | --- | --- | --- | --- | --- | --- | --- | | r | 0.132 | 0.149 | 0.246 | 0.354 | 0.373 | 0.248 | | K | 3676.357 | 3840.696 | 5184.224 | 7965.332 | 8997.495 | 5481.514 | | Binit | 1727.198 | 1845.846 | 2829.008 | 4935.752 | 5603.287 | 3041.688 | | sigma | 0.139 | 0.142 | 0.157 | 0.163 | 0.163 | 0.155 | | -veLL | -17.232 | -16.465 | -13.479 | -12.316 | -12.238 | -13.815 | | MSY | 280.370 | 289.467 | 318.420 | 352.719 | 366.242 | 319.546 | | Depl | 0.338 | 0.367 | 0.529 | 0.669 | 0.699 | 0.524 | | Harv | 0.051 | 0.058 | 0.088 | 0.116 | 0.124 | 0.087 | |

可以使用直方图可视化此类百分位置信区间，并包括相应的选定百分位置信区间。

人们期望 1000 次重复将提供平滑的响应和具有代表性的置信范围，但有时，尤其是在嘈杂的数据中，需要更多的重复才能获得不确定性的平滑表示。2000 次重复需要 20 秒可能看起来很长，但考虑到这样的事情过去需要数小时甚至数天，大约 20 秒是了不起的。请注意，置信边界在均值或中位数估计值附近不一定是对称的。另请注意，在最后一年的消耗估计中，第 5 个百分位的置信区间远高于 ，这意味着即使这种分析是不确定的，目前的消耗水平也高于大多数地方使用的生物量消耗的默认极限参考点，可能性超过 95%。我们需要中央第 80 个百分位数才能找到下限 10%，但它必然高于第 5 个百分位数。 和 值所显示的中位数和均值比其他参数和模型输出的差异更大，这表明存在一些偏差证据（[图 7.12](#fig-spm12)） 。由于某些图仍然存在粗糙度，因此可以通过增加重复次数来改善粗糙度。

#boostrap CI. Note use of uphist to expand scale Fig 7.12   
{colf <- c(1,1,1,4); lwdf <- c(1,3,1,3); ltyf <- c(1,1,1,2)   
colsf <- c(2,3,4,6)  
parset(plots=c(3,2))   
hist(bootpar[,"r"],breaks=25,main="",xlab="r")   
abline(v=c(bootCI["r",colsf]),col=colf,lwd=lwdf,lty=ltyf)   
uphist(bootpar[,"K"],maxval=14000,breaks=25,main="",xlab="K")   
abline(v=c(bootCI["K",colsf]),col=colf,lwd=lwdf,lty=ltyf)   
hist(bootpar[,"Binit"],breaks=25,main="",xlab="Binit")   
abline(v=c(bootCI["Binit",colsf]),col=colf,lwd=lwdf,lty=ltyf)   
uphist(bootpar[,"MSY"],breaks=25,main="",xlab="MSY",maxval=450)   
abline(v=c(bootCI["MSY",colsf]),col=colf,lwd=lwdf,lty=ltyf)   
hist(bootpar[,"Depl"],breaks=25,main="",xlab="Final Depletion")   
abline(v=c(bootCI["Depl",colsf]),col=colf,lwd=lwdf,lty=ltyf)   
hist(bootpar[,"Harv"],breaks=25,main="",xlab="End Harvest Rate")   
abline(v=c(bootCI["Harv",colsf]),col=colf,lwd=lwdf,lty=ltyf) }

|  |
| --- |
| 图 7.12: The 1000 bootstrap replicates from the optimum spm fit to the dataspm data-set. The vertical lines, in each case, are the median and 90th percentile confidence intervals and the dashed vertical blue lines are the mean values. The function uphist() is used to expand the x-axis in K, Binit, and MSY. |

存储在 *boots$dynam* 中的拟合轨迹也可以直观地指示分析的不确定性。

#Fig7.13 1000 bootstrap trajectories for dataspm model fit   
dynam <- boots$dynam   
years <- fish[,"year"]   
nyrs <- length(years)   
parset()   
ymax <- getmax(c(dynam[,,"predCE"],fish[,"cpue"]))   
plot(fish[,"year"],fish[,"cpue"],type="n",ylim=c(0,ymax),   
 xlab="Year",ylab="CPUE",yaxs="i",panel.first = grid())   
for (i in 1:reps) lines(years,dynam[i,,"predCE"],lwd=1,col=8)   
lines(years,answer$Dynamics$outmat[1:nyrs,"predCE"],lwd=2,col=0)   
points(years,fish[,"cpue"],cex=1.2,pch=16,col=1)   
percs <- apply(dynam[,,"predCE"],2,quants)   
arrows(x0=years,y0=percs["5%",],y1=percs["95%",],length=0.03,   
 angle=90,code=3,col=0)

|  |
| --- |
| 图 7.13: A plot of the original observed CPUE (black dots), the optimum predicted CPUE (solid line), the 1000 bootstrap predicted CPUE (the grey lines), and the 90th percentile confidence intervals around those predicted values (the vertical bars). |

预测 CPUE 值与观测 CPUE 值之间存在明显偏差（[图 7.13](#fig-spm13)）,但估计值的中位数及其周围的置信区间仍然十分明确。

请记住，无论何时在时间序列数据上使用自举法，其中时刻 的值与时刻 的值相关，都有必要自举任何拟合模型的残差值，并将它们与最优拟合值联系起来。对于CPUE 数据，我们通常使用对数正态残差误差，因此一旦找到最优解，这些残差定义为:

其中 是年中的观测 CPUE， 是年 中观测 CPUE 除以预测 CPUE（对数正态残差 。这种残差会有一个时间序列，自举法的生成包括从时间序列中随机抽取数值，并进行替换， 从而得到一个对数正态残差的自举样本。然后将这些值乘以原始的最优预测 cpue 值，生成不同时间序列的自举 cpue。

其中上标 ∗ 表示自举样本， 表示年 的 Bootstrap CPUE， 表示来自对数正态残差的单个随机样本，然后将其乘以当年的预测 CPUE。这些方程反映了 **MQMF** 函数 spmboot() 中的特定代码行。

值得一做的是重复上述分析，但将 *schaefer = TRUE* 处改为 *FALSE* ，以便用 Fox 剩余产量模型来拟合模型。这样就可以比较两个模型的不确定性。

#Fit the Fox model to dataspm; note different parameters   
pars <- log(c(r=0.15,K=6500,Binit=3000,sigma=0.20))   
ansF <- fitSPM(pars,fish,schaefer=FALSE,maxiter=1000) #Fox version   
bootsF <- spmboot(ansF$estimate,fishery=fish,iter=reps,schaefer=FALSE)   
dynamF <- bootsF$dynam

# bootstrap trajectories from both model fits Fig 7.14   
parset()   
ymax <- getmax(c(dynam[,,"predCE"],fish[,"cpue"]))   
plot(fish[,"year"],fish[,"cpue"],type="n",ylim=c(0,ymax),   
 xlab="Year",ylab="CPUE",yaxs="i",panel.first = grid())   
for (i in 1:reps) lines(years,dynamF[i,,"predCE"],lwd=1,col=1,lty=1)   
for (i in 1:reps) lines(years,dynam[i,,"predCE"],lwd=1,col=8)   
lines(years,answer$Dynamics$outmat[1:nyrs,"predCE"],lwd=2,col=0)   
points(years,fish[,"cpue"],cex=1.1,pch=16,col=1)   
percs <- apply(dynam[,,"predCE"],2,quants)   
arrows(x0=years,y0=percs["5%",],y1=percs["95%",],length=0.03,   
 angle=90,code=3,col=0)   
legend(1985,0.35,c("Schaefer","Fox"),col=c(8,1),bty="n",lwd=3)

|  |
| --- |
| 图 7.14: A plot of the original observed CPUE (dots), the optimum predicted CPUE (solid white line) with the 90th percentile confidence intervals (the white bars). The black lines are the Fox model bootstrap replicates while the grey lines over the black are those from the Schaefer model. |

可以说，Fox 模型在捕捉这些数据的变异性方面更成功，因为黑线的扩散范围略大于灰色线（[图 7.14](#fig-spm14)）。或者，可以说 Fox 模型不太确定。总体而言，Schaefer 和 Fox 模型的输出之间没有太大差异，甚至像他们预测的那样 值非常相似（313.512 吨与 311.661 吨）。然而，最终，Fox 模型中密度依赖性的非线性似乎赋予了它更大的灵活性，因此它能够比更严格的 Schaefer 模型更好地捕获原始数据的变异性（因此它的 -ve 对数似然性更小，参见 outfit(ansF)）。但这两个模型都无法捕获残差中表现出的循环特性，意味着建模动力学中未包含某些过程，即模型错误规范。这两种模型都不完全充分，尽管它们都可以提供足够的近似动态，可以用来产生管理建议（关于周期过程随时间保持不变的警告，等等）。

### 7.4.3 参数相关性

组合的 bootstrap 样本和相关估计值提供了反映数据和拟合模型的参数之间变异性的表征。如果我们将各种参数相互绘制，任何参数相关性都会变得明显。之间强烈的负曲线-线性关系 和 非常明显，而与其他参数之间的关系也既不是随机的，也不是平滑正态的。在极端值下有一些点，但它们仍然很少见，但是，这些图确实说明了该分析中的变化形式。

# plot variables against each other, use MQMF panel.cor Fig 7.15   
pairs(boots$bootpar[,c(1:4,6,7)],lower.panel=panel.smooth,   
 upper.panel=panel.cor,gap=0,lwd=2,cex=0.5)

|  |
| --- |
| 图 7.15: 模型参数与 Schaefer 模型（ Fox 模型使用 bootsF$bootpar）的一些输出之间的关系。下方面板在数据中具有一条红色的平滑线，用于说明任何趋势，而上方面板具有线性相关系数。少数极值会扭曲绘图。The relationships between the model parameters and some outputs for the Schaefer model (use bootsF$bootpar for the Fox model ). The lower panels have a red smoother line through the data illustrating any trends, while the upper panels have the linear correlation coefficient. The few extreme values distort the plots. |

### 7.4.4 渐近误差

如*“不确定性”*一章所述，在模型拟合过程中，描述与参数估计相关的不确定性的经典方法是使用所谓的渐近误差。渐近误差源自方差-协方差矩阵，可用于描述模型参数之间的变异性和交互作用。在自举法的章节中，可以 pairs() 函数直观显示参数之间的关系， 而这些关系显然不是很好的多变量正态关系。尽管如此，仍然可以使用从方差-协方差矩阵 （*vcov*） 得出的多变量正态来描述模型的不确定性。在使用optim() 或 nlm() 拟合模型时，我们可以将 *vcov* 估计为一个选项来估计。

#Start the SPM analysis using asymptotic errors.   
data(dataspm) # Note the use of hess=TRUE in call to fitSPM   
fish <- as.matrix(dataspm) # using as.matrix for more speed   
colnames(fish) <- tolower(colnames(fish)) # just in case  
pars <- log(c(r=0.25,K=5200,Binit=2900,sigma=0.20))   
ans <- fitSPM(pars,fish,schaefer=TRUE,maxiter=1000,hess=TRUE)

通过使用 outfit() 函数，我们可以看到在 *hess* 参数设置为 “TRUE”的情况下，Schaefer 剩余产量模型与 *dataspm* 数据集的拟合结果。

#The hessian matrix from the Schaefer fit to the dataspm data   
 outfit(ans)

nlm solution:   
minimum : -12.12879   
iterations : 2   
code : 2 >1 iterates in tolerance, probably solution   
 par gradient transpar  
1 -1.417080 0.0031126661 0.24242  
2 8.551232 -0.0017992364 5173.12308  
3 7.953564 -0.0009892147 2845.69834  
4 -1.810225 -0.0021756288 0.16362  
hessian :   
 [,1] [,2] [,3] [,4]  
[1,] 1338.3568627 1648.147068 -74.39814471 -0.14039276  
[2,] 1648.1470677 2076.777078 -115.32342460 -1.80063349  
[3,] -74.3981447 -115.323425 25.48912486 -0.01822396  
[4,] -0.1403928 -1.800633 -0.01822396 61.99195077

fitSPM() 中的最终最小化使用的是最大似然法（实际上是最小负对数似然），因此我们需要反演赫斯方差以获得方差-协方差矩阵。对角线的平方根也给出了每个参数的标准误差估计值（参见 “不确定性”一章）。

#calculate the var-covar matrix and the st errors   
vcov <- solve(ans$hessian) # calculate variance-covariance matrix   
label <- c("r","K", "Binit","sigma")   
colnames(vcov) <- label; rownames(vcov) <- label   
outvcov <- rbind(vcov,sqrt(diag(vcov)))   
rownames(outvcov) <- c(label,"StErr")

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7.6: The variance-covariance (vcov) matrix is the inverse of the Hessian and the parameter standard errors are the square-root of the diagonal of the vcov matrix.   |  | r | K | Binit | sigma | | --- | --- | --- | --- | --- | | r | 0.0668 | -0.0563 | -0.0599 | -0.0015 | | K | -0.0563 | 0.0481 | 0.0534 | 0.0013 | | Binit | -0.0599 | 0.0534 | 0.1062 | 0.0014 | | sigma | -0.0015 | 0.0013 | 0.0014 | 0.0162 | | StErr | 0.2584 | 0.2194 | 0.3258 | 0.1271 | |

现在我们有了最优解和方差-协方差矩阵，可以使用多变量正态分布来获得多个参数的合理组合，这些参数组合可以用来计算输出，如 ，并描述预期动态。基本 R 不包括从多变量正态分布中采样的方法，但有一些免费提供的软件包可以做到。我们将使用可从 CRAN 下载的 mvtnorm 软件包。在使用这种软件包时，可以通过 packageDescription() 函数确定编写者和其他重要信息。另外，在查看软件包中某个函数的帮助文件时，如果滚动到页面底部并点击索引超链接，就可以直接阅读描述文件。

#generate 1000 parameter vectors from multi-variate normal   
library(mvtnorm) # use RStudio, or install.packages("mvtnorm")   
N <- 1000 # number of parameter vectors, use vcov from above   
mvn <- length(fish[,"year"]) #matrix to store cpue trajectories   
mvncpue <- matrix(0,nrow=N,ncol=mvn,dimnames=list(1:N,fish[,"year"]))   
columns <- c("r","K","Binit","sigma")   
optpar <- ans$estimate # Fill matrix with mvn parameter vectors   
mvnpar <- matrix(exp(rmvnorm(N,mean=optpar,sigma=vcov)),nrow=N,   
 ncol=4,dimnames=list(1:N,columns))   
msy <- mvnpar[,"r"]\*mvnpar[,"K"]/4   
nyr <- length(fish[,"year"])   
depletion <- numeric(N) #now calculate N cpue series in linear space   
for (i in 1:N) { # calculate dynamics for each parameter set   
 dynamA <- spm(log(mvnpar[i,1:4]),fish)   
 mvncpue[i,] <- dynamA$outmat[1:nyr,"predCE"]   
 depletion[i] <- dynamA$outmat["2016","Depletion"]   
}   
mvnpar <- cbind(mvnpar,msy,depletion) # try head(mvnpar,10)

[图 7.13](#fig-spm13) 和 [图 7.14](#fig-spm14) 通过自举法绘制出隐含的 CPUE 轨迹，结果似乎是可信的。另一方面，利用渐近误差，当我们绘制隐含的动态图时，如 [图 7.16](#fig-spm16) ，有一定比例的鱼类在第 90 个百分位数置信区间之外，而置信区间本身是极不对称的，会产生剧烈波动的动态，甚至可能意味着鱼类灭绝。

#data and trajectories from 1000 MVN parameter vectors Fig 7.16   
plot1(fish[,"year"],fish[,"cpue"],type="p",xlab="Year",ylab="CPUE",   
 maxy=2.0)   
for (i in 1:N) lines(fish[,"year"],mvncpue[i,],col="grey",lwd=1)   
points(fish[,"year"],fish[,"cpue"],pch=1,cex=1.3,col=1,lwd=2) # data   
lines(fish[,"year"],exp(simpspm(optpar,fish)),lwd=2,col=1)# pred   
percs <- apply(mvncpue,2,quants) # obtain the quantiles   
arrows(x0=fish[,"year"],y0=percs["5%",],y1=percs["95%",],length=0.03,   
 angle=90,code=3,col=1) #add 90% quantiles   
msy <- mvnpar[,"r"]\*mvnpar[,"K"]/4 # 1000 MSY estimates   
text(2010,1.75,paste0("MSY ",round(mean(msy),3)),cex=1.25,font=7)

|  |
| --- |
| 图 7.16: 从最优参数及其相关方差-协方差矩阵定义的多变量正态分布中采样的随机参数向量得出的 1000 条 cpue 预测轨迹。The 1000 predicted cpue trajectories derived from random parameter vectors sampled from the multi-variate normal distribution defined by the optimum parameters and their related variance-covariance matrix. |

使用渐近误差估计的平均 与自举估计值非常相似（319.546，[表 7.5](#tbl-spm5)），第90 分位置信区间看起来也很有意义，尽管比自举分析更加偏斜。然而，使用多变量正态分布显然会导致一些难以置信的参数组合，进而导致难以置信的 cpue 轨迹，与观测 cpue 相差甚远。这并不意味着不应该使用渐近误差，而是说如果确实使用了渐近误差，就应该对其影响的合理性进行研究。

在这种情况下，我们可以通过查找最终 cpue 值小于 0.4 的记录来搜索导致极端结果的参数组合。

#Isolate errant cpue trajectories Fig 7.17   
pickd <- which(mvncpue[,"2016"] < 0.40)   
plot1(fish[,"year"],fish[,"cpue"],type="n",xlab="Year",ylab="CPUE",   
 maxy=6.25)   
for (i in 1:length(pickd))   
 lines(fish[,"year"],mvncpue[pickd[i],],col=1,lwd=1)   
points(fish[,"year"],fish[,"cpue"],pch=16,cex=1.25,col=4)   
lines(fish[,"year"],exp(simpspm(optpar,fish)),lwd=3,col=2,lty=2)

|  |
| --- |
| 图 7.17: 预测 2016 年 cpue < 0.4 的 34 个渐近误差 cpue 轨迹。圆点为原始数据，虚线为最佳拟合模型。The 34 asymptotic error cpue trajectories that were predicted to have a cpue < 0.4 in 2016. The dots are the original data and the dashed line the optimum model fit. |

现在，我们已经确定了大多数错误轨迹及其各自的参数矢量，我们可以通过绘图来比较我们认为的非错误轨迹，这样我们就可以确定谁是谁了（[图 7.18](#fig-spm18)）。

#Use adhoc function to plot errant parameters Fig 7.18   
parset(plots=c(2,2),cex=0.85)   
outplot <- function(var1,var2,pickdev) {   
 plot1(mvnpar[,var1],mvnpar[,var2],type="p",pch=16,cex=1.0,   
 defpar=FALSE,xlab=var1,ylab=var2,col=8)   
 points(mvnpar[pickdev,var1],mvnpar[pickdev,var2],pch=16,cex=1.0)   
}   
outplot("r","K",pickd) # assumes mvnpar in working environment   
outplot("sigma","Binit",pickd)   
outplot("r","Binit",pickd)   
outplot("K","Binit",pickd)

|  |
| --- |
| 图 7.18: 渐近误差样本中参数值的分布，黑色部分为预测最终 cpue < 0.4 的参数值。看来，Binit 的低值是造成难以置信轨迹的主要原因。The spread of parameter values from the asymptotic error samples with the values that predicted final cpue < 0.4 highlighted in black. It appears that low values of Binit are mostly behind the implausible trajectories. |

当我们绘制模型变量之间的相互关系时（[图 7.19](#fig-spm19)），对于正态分布或多变量正态分布变量，预期 *simga* 和其他参数之间缺乏关系。然而，这与从自举法样本中获得的关系明显不同（[图 7.15](#fig-spm15)）。此外，三个主要参数 、 和 之间的关系远比在自举抽样中看到的平滑得多。在我们看来，这种对称性和界限的清晰度似乎比自举样本中的关系更容易接受（[图 7.19](#fig-spm19)）。尽管如此，损耗图显示一些轨迹似乎已经消失。

#asymptotically sampled parameter vectors Fig 7.19   
pairs(mvnpar,lower.panel=panel.smooth, upper.panel=panel.cor,   
 gap=0,cex=0.25,lwd=2)

|  |
| --- |
| 图 7.19: 使用多变量正态分布生成参数组合时 Schaefer 模型参数之间的关系。r - K 之间的关系比自举样本紧密得多，而 sigma 与其他参数之间几乎没有关系。损耗图显示一些轨迹已经消失。The relationships between the model parameters for the Schaefer model when using the multi-variate normal distribution to generate the parameter combinations. The relationship between r - K is much tighter than in the bootstrap samples and there is almost no relationship between sigma and the other parameters. The depletion plots indicate some trajectories go extinct. |

我们可以比较自举抽样和渐近误差抽样的参数值范围。来自渐近误差分布的参数样本比来自自举法的样本偏度小，但自举法对于 和 的值没有那么低。需要记住的是，使用多元正态分布来描述围绕最优参数集的似然曲面的形状仍然只是一个近似值。

# Get the ranges of parameters from bootstrap and asymptotic   
bt <- apply(bootpar,2,range)[,c(1:4,6,7)]   
ay <- apply(mvnpar,2,range)   
out <- rbind(bt,ay)   
rownames(out) <- c("MinBoot","MaxBoot","MinAsym","MaxAsym")

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7.7: 自举取样与渐近误差取样的参数值范围对比。The range of parameter values from the bootstrap sampling compared with those from the Asymptotic Error sampling.   |  | r | K | Binit | sigma | MSY | Depl | | --- | --- | --- | --- | --- | --- | --- | | MinBoot | 0.0653 | 3139.827 | 1357.264 | 0.1125 | 217.1636 | 0.0953 | | MaxBoot | 0.4958 | 25666.568 | 8000.087 | 0.1636 | 530.6523 | 0.7699 | | MinAsym | 0.1185 | 2055.714 | 1003.558 | 0.1069 | 271.9012 | 0.0054 | | MaxAsym | 0.7287 | 9581.273 | 9917.274 | 0.2344 | 374.5219 | 0.6820 | |

### 7.4.5 有时渐近误差起作用

在某些情况下，渐近误差法得到的结果与自举法的结果非常相似。如果我们使用的是 *abdat* 数据而不是 *dataspm* 数据，我们得到的结果与使用自举法得到的结果似乎没有什么区别（比较见 “不确定 性”一章中的自举法部分）。生成的轨迹看起来非常相似（[图 7.20](#fig-spm20)），而且成对图几乎没有区别。与 “关于不确定性”的自举示例一样，我们使用了 rgb() 着色以方便比较（[图 7.21](#fig-spm21)）。

#repeat asymptotice errors using abdat data-set Figure 7.20   
data(abdat)   
fish <- as.matrix(abdat)   
pars <- log(c(r=0.4,K=9400,Binit=3400,sigma=0.05))   
ansA <- fitSPM(pars,fish,schaefer=TRUE,maxiter=1000,hess=TRUE)   
vcovA <- solve(ansA$hessian) # calculate var-covar matrix   
mvn <- length(fish[,"year"])   
N <- 1000 # replicates   
mvncpueA <- matrix(0,nrow=N,ncol=mvn,dimnames=list(1:N,fish[,"year"]))   
columns <- c("r","K","Binit","sigma")   
optparA <- ansA$estimate # Fill matrix of parameter vectors   
mvnparA <- matrix(exp(rmvnorm(N,mean=optparA,sigma=vcovA)),   
 nrow=N,ncol=4,dimnames=list(1:N,columns))   
msy <- mvnparA[,"r"]\*mvnparA[,"K"]/4   
for (i in 1:N) mvncpueA[i,]<-exp(simpspm(log(mvnparA[i,]),fish))   
mvnparA <- cbind(mvnparA,msy)   
plot1(fish[,"year"],fish[,"cpue"],type="p",xlab="Year",ylab="CPUE",   
 maxy=2.5)   
for (i in 1:N) lines(fish[,"year"],mvncpueA[i,],col=8,lwd=1)   
points(fish[,"year"],fish[,"cpue"],pch=16,cex=1.0) #orig data   
lines(fish[,"year"],exp(simpspm(optparA,fish)),lwd=2,col=0)

|  |
| --- |
| 图 7.20: 利用渐近误差为 abdat 数据集生成可信的参数集及其隐含的 cpue 轨迹。最佳拟合模型以白线表示。The use of asymptotic errors to generate plausible parameter sets and their implied cpue trajectories for the abdat data-set. The optimum model fit is shown as a white line. |

#plot asymptotically sampled parameter vectors Figure 7.21   
pairs(mvnparA,lower.panel=panel.smooth, upper.panel=panel.cor,   
 gap=0,pch=16,col=rgb(red=0,green=0,blue=0,alpha = 1/10))

|  |
| --- |
| 图 7.21: 将 Schaefer 模型拟合到 abdat 数据并使用多变量正态分布生成后续参数组合时的模型参数关系。这与 “不确定性”一章中的自举法非常相似。Model parameter relationships when fitting the Schaefer model to the abdat data and using the multi-variate normal distribution to generate subsequent parameter combinations. These are very similar to the bootstrap equivalent in the On Uncertainty chapter. |

### 7.4.6 贝叶斯后验

在 *“不确定性”* 一章中，我们已经看到可以使用马尔可夫链蒙特卡罗（MCMC）分析来描述给定分析中固有的不确定性。在这里，我们将再次使用 *abdat* 数据集，因为它提供了一个表现良好的数据实例，该数据集导致了一个相对紧密拟合的模型和一个表现良好的 MCMC 分析。关于*“不确定性”*一章中给出了吉布斯-内大都会-哈斯丁（Gibbs-within-Metropolis-Hastings）（或单分量大都会-哈斯丁Metropolis-Hastings）策略背后的方程式。这些都在 do\_MCMC() 函数中实现。要使用该函数，首先要有一个基于最大似然法的最优拟合模型。这次我们将使用 Fox 模型选项。

#Fit the Fox Model to the abdat data Figure 7.22   
data(abdat); fish <- as.matrix(abdat)   
param <- log(c(r=0.3,K=11500,Binit=3300,sigma=0.05))   
foxmod <- nlm(f=negLL1,p=param,funk=simpspm,indat=fish,   
 logobs=log(fish[,"cpue"]),iterlim=1000,schaefer=FALSE)   
optpar <- exp(foxmod$estimate)   
ans <- plotspmmod(inp=foxmod$estimate,indat=fish,schaefer=FALSE,   
 addrmse=TRUE, plotprod=TRUE)

|  |
| --- |
| 图 7.22: 使用 Fox 模型和对数正态误差拟合的 abdat 数据集最佳模型。绿色虚线是较平滑的曲线，红线是最佳预测模型拟合。请注意对数正态残差的模式，这表明该模型在该数据方面存在微小不足。The optimum model fit for the abdat data-set using the Fox model and log-normal errors. The green dashed line is a smoother curve while the red line is the optimum predicted model fit. Note the pattern in the log-normal residuals indicating that the model has small inadequacies with regard to this data. |

由于最优解将接近后验模式，我们不再需要说明预演期的概念，但理想情况下，我们并不希望完全从最大似然解出发。因此，我们可以舍弃最优解，对马尔可夫链进行预烧，使参数集序列进入可信组合的范围。我们知道 和 参数之间有很强的相关性，因此我们可以使用 128（）的初始步长来减少任何连续接受值之间的自相关性，但这也会受到参数迭代之间跳跃的相对比例的影响。在这里，我们从 1% 到 2%之间的值开始，并尝试使用这些值，直到接受率介于 0.2 和 0.4 之间。最好使用较小的 N 值（使用 512 的稀疏度，即使 1000 也是 50 万次迭代）。只有当刻度设置得当时，才能将重复次数 N 扩大到更大的数目，以获得更清晰的结果。我们将继续使用 **MQMF** 函数 calcprior()，对每组可信参数设置同等权重，为了获得可重复的结果，需要在每条链上调用 set.seed()，但一般情况下我们不会这样做。在 R 中，所有操作系统都使用相同的随机数生成器，因此这应该可以在不同的计算机上运行，但我还没有在所有版本上都试过。为了提高计算速度，最好能有类似于 *“不确定性”*一章中描述的使用 **Rcpp** 的 simpspmC() 函数。在运行下面的 MCMC 之前，你需要编译本章的附录，或者在使用 do\_MCMC() 时调用 simpspm()，注意为了使用 Fox 模型，需要加入 *schaefer=FALSE* 参数。

#|echo: false  
  
library(Rcpp)   
   
cppFunction('NumericVector simpspmC(NumericVector pars,   
 NumericMatrix indat, LogicalVector schaefer) {   
 int nyrs = indat.nrow();   
 NumericVector predce(nyrs);   
 NumericVector biom(nyrs+1);   
 double Bt, qval;   
 double sumq = 0.0;   
 double p = 0.00000001;   
 if (schaefer(0) == TRUE) {   
 p = 1.0;   
 }   
 NumericVector ep = exp(pars);   
 biom[0] = ep[2];   
 for (int i = 0; i < nyrs; i++) {   
 Bt = biom[i];   
 biom[(i+1)] = Bt + (ep[0]/p)\*Bt\*(1 - pow((Bt/ep[1]),p)) -   
 indat(i,1);   
 if (biom[(i+1)] < 40.0) biom[(i+1)] = 40.0;   
 sumq += log(indat(i,2)/biom[i]);   
 }   
 qval = exp(sumq/nyrs);   
 for (int i = 0; i < nyrs; i++) {   
 predce[i] = log(biom[i] \* qval);   
 }   
 return predce;   
}')

# eval: false  
 # Conduct an MCMC using simpspmC on the abdat Fox SPM   
 # This means you will need to compile simpspmC from appendix   
set.seed(698381) #for repeatability, possibly only on Windows10   
begin <- gettime() # to enable the time taken to be calculated   
inscale <- c(0.07,0.05,0.09,0.45) #note large value for sigma   
pars <- log(c(r=0.205,K=11300,Binit=3200,sigma=0.044))   
result <- do\_MCMC(chains=1,burnin=50,N=2000,thinstep=512,   
 inpar=pars,infunk=negLL,calcpred=simpspm,   
 obsdat=log(fish[,"cpue"]),calcdat=fish,   
 priorcalc=calcprior,scales=inscale,schaefer=FALSE)   
 # alternatively, use simpspm, but that will take longer.   
cat("acceptance rate = ",result$arate," \n")

acceptance rate = 0.3136629 0.3337832 0.3789844 0.3660627

cat("time = ",gettime() - begin,"\n")

time = 82.84388

post1 <- result[[1]][[1]]   
p <- 1e-08   
msy <- post1[,"r"]\*post1[,"K"]/((p + 1)^((p+1)/p))

Fox 模型 MCMC 目前设置为 512 的稀释率、2000 次重复和 50 次老化，这意味着将有 次迭代用于生成所需的参数跟踪。在用于编写此内容的计算机上，即使使用 ，这大约需要 15 秒;使用 simpspm() 可能预计大约需要 75 秒。一旦知道了自己的系统的情况，显然可以计划分析，并对稀释速率和重复做出明确的选择（不要忘记使用最新版本的 R 以获得最快的时间）。

分析完成后，我们可以使用 pairs() 函数绘制每个变量与其他变量的对比图（[图 7.23](#fig-spm23)）。此外，我们还可以绘制每个主要参数的边际后验分布图和推导出的模型输出（MSY）。由于我们使用了 2000 个重复样本，并采用了 512 的样本链稀疏率（[图 7.24](#fig-spm24)），因此后验分布相对平滑。

#pairwise comparison for MCMC of Fox model on abdat Fig 7.23   
pairs(cbind(post1[,1:4],msy),upper.panel = panel.cor,lwd=2,cex=0.2,   
 lower.panel=panel.smooth,col=1,gap=0.1)

|  |
| --- |
| 图 7.23: MCMC 输出的成对散点图。实线是表示趋势的塬平滑线，上半部分的数字是成对散点图之间的相关系数。r、K 和 Binit 之间，以及 K、Binit 和 MSY 之间都有很强的相关性，而 sigma 与其他参数或 msy 与 r 之间的关系较小或没有关系。MCMC output as paired scattergrams. The solid lines are loess smoothers indicating trends and the numbers in the upper half are the correlation coefficients between the pairs. Strong correlations are indicated between r, K, and Binit, and between K, Binit, and MSY, with only minor or no relationships between sigma the other parameters or between msy and r. |

# marginal distributions of 3 parameters and msy Figure 7.24   
parset(plots=c(2,2), cex=0.85)   
plot(density(post1[,"r"]),lwd=2,main="",xlab="r") #plot has a method   
plot(density(post1[,"K"]),lwd=2,main="",xlab="K") #for output from   
plot(density(post1[,"Binit"]),lwd=2,main="",xlab="Binit") # density   
plot(density(msy),lwd=2,main="",xlab="MSY") #try str(density(msy))

|  |
| --- |
| 图 7.24: 将 Fox 模型应用于鳕鱼数据的 2000 次 MCMC 重复计算得出的三个参数的边际分布和隐含的 MSY。曲线的块状表明需要进行 2000 次以上的迭代。The marginal distributions for three parameters and the implied MSY from 2000 MCMC replicates for the Fox model applied to the abdat data. The lumpiness of the curves suggests more than 2000 iterations are needed. |

需要注意的是，要将 *sigma* 的接受率降到 0.4 以下，需要施加一个相对较大的比例因子。其他参数要求的值在 5% 到 9% 之间。如果使用 500 次重复来寻找合适的比例因子，然后将重复次数重设为 2000 次，那么整个过程所需的时间将是原来的四倍。如果再将步长增加到 1024 倍，那么所需的时间又会增加一倍。需要寻找适当的比例因子，以确保马尔可夫链在合理的时间内充分探索后验空间。如果比例因子太小，接受率就会增加，因为每次试验实际上都会非常接近原始试验，因此只能采取小步试验。静态分布最终仍会被发现，但可能需要大量的重复。在稀疏率为 512 的情况下，如果使用 acf() 函数绘制任何迹线的自相关图，如 acf(post1[, “r”])，就会发现在步长为 1 和 2 时仍然存在显著的相关性。要减少这种相关性，至少需要将步长增加到 1024 步。

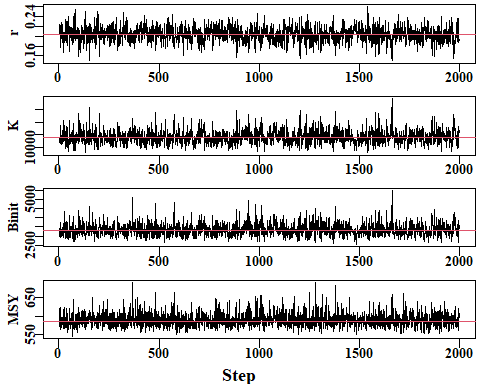
随着重复次数的增加，观测到的潜在参数组合的分布范围也在扩大。但是，如果我们检查第 90 分位数等值线的边界，这些边界会保持相对稳定。我们可以使用 **MQMF** 函数 addcontours() 在二维范围内进行检查，该函数可以为任意 x-y 数据点云生成等值线（任意但理想的平滑分布）。 2000 个观测点的第 50 和第 90 分位数等值线并不特别平滑，但即使是这样， 的边界也大约在 9500 - 14000 之间， 的边界大约在 0.17 - 0.24 之间（[图 7.25](#fig-spm25)）。随着数值的增加，等值线变得更加平滑，但其边界大致保持不变，即使在两种情况下 x 轴和 y 轴都有所延长。

#MCMC r and K parameters, approx 50 + 90% contours. Fig7.25   
puttxt <- function(xs,xvar,ys,yvar,lvar,lab="",sigd=0) {   
 text(xs\*xvar[2],ys\*yvar[2],makelabel(lab,lvar,sep=" ",   
 sigdig=sigd),cex=1.2,font=7,pos=4)   
} # end of puttxt - a quick utility function   
kran <- range(post1[,"K"]); rran <- range(post1[,"r"])   
mran <- range(msy) #ranges used in the plots   
parset(plots=c(1,2),margin=c(0.35,0.35,0.05,0.1)) #plot r vs K   
plot(post1[,"K"],post1[,"r"],type="p",cex=0.5,xlim=kran,   
 ylim=rran,col="grey",xlab="K",ylab="r",panel.first=grid())   
points(optpar[2],optpar[1],pch=16,col=1,cex=1.75) # center   
addcontours(post1[,"K"],post1[,"r"],kran,rran, #if fails make   
 contval=c(0.5,0.9),lwd=2,col=1) #contval smaller   
puttxt(0.7,kran,0.97,rran,kran,"K= ",sigd=0)   
puttxt(0.7,kran,0.94,rran,rran,"r= ",sigd=4)   
plot(post1[,"K"],msy,type="p",cex=0.5,xlim=kran, # K vs msy   
 ylim=mran,col="grey",xlab="K",ylab="MSY",panel.first=grid())   
points(optpar[2],getMSY(optpar,p),pch=16,col=1,cex=1.75)#center   
addcontours(post1[,"K"],msy,kran,mran,contval=c(0.5,0.9),lwd=2,col=1)   
puttxt(0.6,kran,0.99,mran,kran,"K= ",sigd=0)   
puttxt(0.6,kran,0.97,mran,mran,"MSY= ",sigd=3)

|  |
| --- |
| 图 7.25: MCMC 边际分布输出为 r 和 K 参数以及 K 和 MSY 值的散点图。灰点是成功的候选参数向量，等值线是近似的第 50 和第 90 分位数。文中给出了全部可接受的参数迹线范围。MCMC marginal distributions output as a scattergram of the r and K parameters, and the K and MSY values. The grey dots are from successful candidate parameter vectors, while the contours are approximate 50th and 90th percentiles. The text give the full range of the accepted parameter traces. |

最后，我们可以绘制 2000 个重复中每个重复的单个迹线。这表明，即使具有平滑的边际分布，偶尔也会出现参数值的峰值，以说明主要参数之间的强负相关。

#Traces for the Fox model parameters from the MCMC Fig7.26   
parset(plots=c(4,1),margin=c(0.3,0.45,0.05,0.05),   
 outmargin = c(1,0,0,0),cex=0.85)   
label <- colnames(post1)   
N <- dim(post1)[1]   
for (i in 1:3) {   
 plot(1:N,post1[,i],type="l",lwd=1,ylab=label[i],xlab="")   
 abline(h=median(post1[,i]),col=2)   
}   
msy <- post1[,1]\*post1[,2]/4   
plot(1:N,msy,type="l",lwd=1,ylab="MSY",xlab="")   
abline(h=median(msy),col=2)   
mtext("Step",side=1,outer=T,line=0.0,font=7,cex=1.1)



三个主要 Schaefer 模型参数和 MSY 估计值的迹线。如果细化步长增加到 1024 步或更长，迹线内剩余的自相关性应得到改善。The traces for the three main Schaefer model parameters and the MSY estimates. The remaining auto-correlation within traces should be improved if the thinning step were increased to 1024 or longer.

当然，理想情况下，我们会用多条链进行这样的分析，以确保每条链都收敛于相同的后验分布。此外，随着 MCMC 的进展，还有许多诊断性统计数据可以用来检查收敛程度的速率。同样理想的情况是，每条链都从不同的位置开始，但即使从同一位置开始，随机数序列最终也会将链引向截然不同的方向。我们可以使用与 robustSPM() 函数相同的方法来选择不同的随机起点。

#Do five chains of the same length for the Fox model   
set.seed(6396679) # Note all chains start from same place, which is   
inscale <- c(0.07,0.05,0.09,0.45) # suboptimal, but still the chains   
pars <- log(c(r=0.205,K=11300,Binit=3220,sigma=0.044)) # differ   
result <- do\_MCMC(chains=5,burnin=50,N=2000,thinstep=512,   
 inpar=pars,infunk=negLL1,calcpred=simpspmC,   
 obsdat=log(fish[,"cpue"]),calcdat=fish,   
 priorcalc=calcprior,scales=inscale,   
 schaefer=FALSE)   
cat("acceptance rate = ",result$arate," \n") # always check this

acceptance rate = 0.3140023 0.3327271 0.3801893 0.36673

#Now plot marginal posteriors from 5 Fox model chains Fig7.27   
parset(plots=c(2,1),cex=0.85,margin=c(0.4,0.4,0.05,0.05))   
post <- result[[1]][[1]]   
plot(density(post[,"K"]),lwd=2,col=1,main="",xlab="K",   
 ylim=c(0,4.4e-04),panel.first=grid())   
for (i in 2:5) lines(density(result$result[[i]][,"K"]),lwd=2,col=i)   
p <- 1e-08   
post <- result$result[[1]]   
msy <- post[,"r"]\*post[,"K"]/((p + 1)^((p+1)/p))   
plot(density(msy),lwd=2,col=1,main="",xlab="MSY",type="l",   
 ylim=c(0,0.0175),panel.first=grid())   
for (i in 2:5) {   
 post <- result$result[[i]]   
 msy <- post[,"r"]\*post[,"K"]/((p + 1)^((p+1)/p))   
 lines(density(msy),lwd=2,col=i)   
}

|  |
| --- |
| 图 7.26: K 参数的边际后验值和 5 链 2000 次重复（512 \* 2000 = 1049600 次迭代）得出的隐含 MSY。分布之间仍存在一些差异，尤其是在模式处，这表明更多的重复和更高的稀疏率可能会改善结果。the marginal posterior for the K parameter and the implied MSY from five chains of 2000 replicates (512 \* 2000 = 1049600 iterations). Some variation remains between the distributions, especially at the mode, suggesting that more replicates and potentially a higher thinning rate would improve the outcome. |

然而，尽管五条链在视觉上存在差异（[图 7.26](#fig-spm27)），如果我们检查 在不同的分位数上，我们发现差异很小（[表 7.8](#tbl-spm8)）。事实上，中值 对于每条链，彼此之间的距离在1.1%左右是令人鼓舞的，最大百分比变化为 2.7%。作为实验，使用相同的 random.seed，但每条链运行 4000 步（总共 万次迭代，但仍然不到 5 分钟），最大变异下降到 1.48%，同样是 0.975 分位数，其他分位数都低于 1%。在这种情况下，由于模型非常简单，而且每个链只需要很短的时间，因此增加步数是值得的。对于参数更多，更复杂的似然计算，在评估小组的最后期限内，这些分析的时间安排可能变得至关重要。

# get qunatiles of each chain   
probs <- c(0.025,0.05,0.5,0.95,0.975)   
storeQ <- matrix(0,nrow=6,ncol=5,dimnames=list(1:6,probs))   
for (i in 1:5) storeQ[i,] <- quants(result$result[[i]][,"K"])   
x <- apply(storeQ[1:5,],2,range)   
storeQ[6,] <- 100\*(x[2,] - x[1,])/x[2,]

knitr::kable(storeQ, digits = 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7.8: 针对 abdat 数据的 Fox 模型运行的五条 MCMC 链中 K 参数的五个量化值。最后一行是各链数值范围的百分比差异，显示它们的中位数相差略高于 1%。Five quantiles on the K parameter from the five MCMC chains run on the Fox model applied to the abdat data. The last row is the percent difference in the range of the values across the chains, which shows their medians differ by slightly more than 1%.   | 0.025 | 0.05 | 0.5 | 0.95 | 0.975 | | --- | --- | --- | --- | --- | | 9859.157 | 10160.471 | 11633.376 | 13740.430 | 14124.828 | | 9893.256 | 10162.570 | 11541.118 | 13689.079 | 14302.523 | | 9922.313 | 10157.503 | 11564.236 | 13620.369 | 14150.819 | | 9875.521 | 10107.843 | 11541.843 | 13533.356 | 13908.780 | | 9835.652 | 10088.899 | 11504.845 | 13640.376 | 14087.693 | | 0.873 | 0.725 | 1.105 | 1.507 | 2.753 | |

|

## 7.5 管理建议

### 7.5.1 两种风险观

正式的资源评估，即使是使用剩余产量模型的简单评估，也可以说明所评估的渔业的种群状况，但问题仍然是如何利用这种评估来提出渔业管理建议。当然，这种建议将取决于有关渔业的管理目标。但是，即使没有正式的渔业政策，也应该能够就未来采用不同渔获量的影响提供咨询意见。我们可以使用最优模型拟合来预测模型动态到未来，这种预测是管理建议的基础，这些建议来自大多数非纯粹经验的资源评估。一旦知道（或假设）了渔业目标，那么，使用模型预测，就可以对未来的努力或捕捞水平进行估计，从而有望引导种群实现选定的目标。

一个共同的目标是努力将渔业维持在平均可以产生最大可持续产量的生物量水平，即 。这种目标将被称为目标参考点，因为它源自讨论生物学参考点的文献（Garcia，1994；FAO，1995，1997）。可将 视为目标，但相关渔获量 （MSY） 实际上应该作为渔获量的上限。除了目标参考点之外，通常还有一个极限参考点，它定义了要避免的资源状态。这通常从被认为对后续补充量构成风险的资源水平的角度进行讨论，尽管这通常只是一个准则。通常，极限参考点 ，或者将通用代理设置为 。 这种限度和目标参考点通常是在正式收获战略的背景下确定的。

### 7.5.2 捕捞策略

在一个管辖区内，捕捞战略确定了决策框架，用于实现不同鱼类种群的既定生物目标，有时还包括经济和社会目标。一般来说，捕捞战略由三部分组成（FAO，1995、1997；Haddon，2007；Smith 等，2008）：

1.监测和收集有关每个相关渔业数据的手段。

2.评估每种渔业的明确方式，通常相对于预先选择的生物（或其他）参考点，例如捕捞死亡率、生物量水平或其替代物。

3.预先确定的捕捞控制规则或决策规则，用于将种群评估或种群状况转化为与未来努力量或捕捞水平相关的管理建议。

理想情况下，这种捕捞策略将经过模拟测试，以确定它们有效的条件，并摒弃无法实现预期目标的方案（Smith，1993；Punt 等，2016）

有许多众所周知的例子明确说明了辖区内渔业的管理目标（DAFF，2007；Deroba 和 Bence，2008；Magnuson-Stevens，2007）。例如，在澳大利亚联邦海洋管辖区，选定的目标是管理主要经济鱼类种群，使其生物量达到最大经济产量（）（DAFF, 2007; DAWR, 2018）；事实上，由于可用的信息不足，无法可靠地估算 ，大多数物种使用 的代用值。同样，将 定义为大多数物种的极限参考点，“其中没有支持选择特定种群的极限参考点[]的信息……”(DAWR, 2018，第10页)。如果估计种群数量低于极限参考点，则停止有针对性的捕捞，尽管在混合渔业中仍可能出现副渔获物。如果鱼量高于极限参考点，则进行预测，以确定未来的渔获量应能促使鱼量顺利增加到目标生物量水平。

## 7.6 风险评估预测

当然，对资源评估模型进行前瞻性预测的想法背后有许多重要的假设。首先，模型能成功捕捉到控制种群生物量的重要动态部分。在剩余产量模型中，这相当于假设对种群生产力的估计在未来将保持不变。请记住，在使用数据集 *dataspm* 时，残差保留了相对较大的振荡模式，这表明该模型在动态变化中遗漏了一些重要内容。尽管有这样的遗漏，模型仍可能保留对近似平均动态的充分估计，以进行有用的预测，但这就假定影响模型拟合的其他因素将继续以过去的方式运行。如果评估具有高度不确定性，那么未来的预测也将具有高度不确定性，这就降低了其在提供建议时的价值。

最简单的预测是使用最优参数估计，并采用恒定渔获量或努力量。我们需要当前种群的生物量和可捕量，以使用渔获量方程将指定的努力量水平转换为渔获量水平：

然后，就可以使用带有最佳参数的标准动力学方程来预测预测渔获量下的生物量水平。如果使用指定的渔获量，则只需使用动力学方程（此处使用 Polacheck 等（1993）的版本）：

### 7.6.1 确定性预测

如果我们使用最优的模型参数，那么对于一系列不同的前向预测渔获量，我们会得到不同的生物量和 cpue 轨迹。为了说明这一点，我们可以再次使用 *abdat* 数据集（注意，我们已将 *hessian* 选项设置为 “true”，因为我们将在后面使用）。

#Prepare Fox model on abdat data for future projections Fig7.28   
data(abdat); fish <- as.matrix(abdat)   
param <- log(c(r=0.3,K=11500,Binit=3300,sigma=0.05))   
bestmod <- nlm(f=negLL1,p=param,funk=simpspm,schaefer=FALSE,  
 logobs=log(fish[,"cpue"]),indat=fish,hessian=TRUE)   
optpar <- exp(bestmod$estimate)   
ans <- plotspmmod(inp=bestmod$estimate,indat=fish,schaefer=FALSE,   
 target=0.4,addrmse=TRUE, plotprod=FALSE)

|  |
| --- |
| 图 7.27: 使用 Fox 模型和对数正态误差拟合的 abdat 数据集最佳模型。绿色虚线为黄土曲线，红色实线为最佳预测拟合模型。请注意对数正态残差的模式，这表明该模型在该数据方面存在一些不足。The optimum model fit for the abdat data-set using the Fox model and log-normal errors. The green dashed line is a loess curve while the solid red line is the optimum predicted model fit. Note the pattern in the log-normal residuals indicating that the model has some inadequacies with regard to this data. |

MSY估计约为 854 吨，资源似乎略高于 目标水平，通常用作表示 。从图中，并检查初始 *abdat* 数据框，我们可以看到 2000 年至 2008 年的渔获量每年 都在910 - 1030 吨之间，这导致模型预测 cpue 和生物量将下降。因此，我们可以探索700-1000吨的十年渔获量预测，或许以50吨为步长。鉴于分析中的不确定性以及这些预测是确定性的，因此对未来太多年的预测研究意义不大。长期预测对于说明不同渔获量的影响可能很有价值，但出于实际目的，十年往往绰绰有余，这取决于被评估物种的寿命（预计寿命较长的物种比寿命短的物种表现出较慢的动态变化）。函数 plotspmmod() 绘制的动态细节是通过使用具有最佳参数的函数 spm() 生成的（查看 plotspmmod() 代码，以了解这些详细信息）。当使用最佳参数运行 spm() 时，其输出包括 *动态（Dynamics）* 对象中 *outmat* 表中的预测动态。这里我们将使用 Fox 模型而不是 Schaefer 运行模型。

函数 spm() 输出的对象是一个由五个部分组成的列表。模型参数，包括 q 值；*outmat* 是一个矩阵，包含随时间变化的动态信息；*msy*；*sumout* 包含五个关键统计量的汇总；*schaefer* 用于识别是 Schaefer 模型还是 Fox 模型。

#   
out <- spm(bestmod$estimate,indat=fish,schaefer=FALSE)   
str(out, width=65, strict.width="cut")

List of 5  
 $ parameters: Named num [1:5] 2.06e-01 1.13e+04 3.23e+03 4.38e..  
 ..- attr(\*, "names")= chr [1:5] "r" "K" "Binit" "Sigma" ...  
 $ outmat : num [1:25, 1:7] 1985 1986 1987 1988 1989 ...  
 ..- attr(\*, "dimnames")=List of 2  
 .. ..$ : chr [1:25] "1985" "1986" "1987" "1988" ...  
 .. ..$ : chr [1:7] "Year" "ModelB" "Catch" "Depletion" ...  
 $ msy : num 856  
 $ sumout : Named num [1:5] 8.56e+02 1.00e-08 4.34e-01 2.86e..  
 ..- attr(\*, "names")= chr [1:5] "msy" "p" "FinalDepl" "InitD"..  
 $ schaefer : logi FALSE

*outmat* 中的动态包括年份、生物量、cpue、预测的 cpue 和其他变量的详细信息（[表 7.9](#tbl-spm9)）。

knitr::kable(out$outmat[1:10,], digits = 3)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表 7.9: 前十行为 abdat 数据集所代表的种群动态预测值和最佳 Fox 模型拟合值。The first ten rows of the predicted dynamics of the stock represented by the abdat data-set and the optimal Fox model fit.   |  | Year | ModelB | Catch | Depletion | Harvest | CPUE | predCE | | --- | --- | --- | --- | --- | --- | --- | --- | | 1985 | 1985 | 3231.056 | 1020 | 0.286 | 0.316 | 1.000 | 1.126 | | 1986 | 1986 | 3043.801 | 743 | 0.269 | 0.244 | 1.096 | 1.061 | | 1987 | 1987 | 3122.663 | 867 | 0.276 | 0.278 | 1.130 | 1.088 | | 1988 | 1988 | 3082.389 | 724 | 0.272 | 0.235 | 1.147 | 1.074 | | 1989 | 1989 | 3182.683 | 586 | 0.281 | 0.184 | 1.187 | 1.109 | | 1990 | 1990 | 3426.836 | 532 | 0.303 | 0.155 | 1.202 | 1.194 | | 1991 | 1991 | 3736.575 | 567 | 0.330 | 0.152 | 1.265 | 1.302 | | 1992 | 1992 | 4020.891 | 609 | 0.355 | 0.151 | 1.320 | 1.401 | | 1993 | 1993 | 4267.332 | 548 | 0.377 | 0.128 | 1.428 | 1.487 | | 1994 | 1994 | 4574.992 | 498 | 0.404 | 0.109 | 1.477 | 1.594 | |

预测是将建模结果的时间序列（[表 7.9](#tbl-spm9) ）按顺序加入任何新的固定渔获量，并进行计算以填入所需列，从而继续进行动态计算。我们可以使用 **MQMF** 函数 spmprojDet()，它接收来自 spm() 函数的列表输出以及与确定性预测相关的一些细节，并为我们生成预测动态。您应该查看 spmproj() 代码，了解年份是如何设置的，代码之简短令人惊讶。

# Fig 7.29   
catches <- seq(700,1000,50) # projyr=10 is the default   
projans <- spmprojDet(spmobj=out,projcatch=catches,plotout=TRUE)

|  |
| --- |
| 图 7.28: 根据 abdat 数据集拟合的最佳 Fox 模型的确定性恒定渔获量预测。垂直绿线是可用数据的极限，其右侧的红线是主要预测。数字是施加的恒定渔获量。Deterministic constant catch projections of the optimum Fox model fit to the abdat data-set. the vertical green line is the limit of the data available and the red lines to the right of that are the main projections. The numbers are the constant catches imposed. |

利用确定性恒定渔获量预测（[图 7.28](#fig-spm29)），可以看出 850 吨的恒定渔获量（接近 估计值）是预测未来种群状况相对稳定的最接近的渔获量。

### 7.6.2 考虑不确定性

确定性预测的一个明显问题就在于它们是确定性的。它们没有考虑到即使在最佳模型拟合中仍然存在的不确定性。理想情况下，我们在进行模型预测时应考虑到估计的不确定性。我们可以采用渐近误差法、自举法模型拟合或贝叶斯法等三种方法来预测不确定性。在每种情况下，不同分析的输出结果都是可信参数组合列表。这些参数可用于描述模型拟合中包含的可信动态范围，并以与确定性预测相同的方式对每个单独的生物量轨迹进行预测。

在拟合最优模型时，我们已经计算了 *Hessian* 矩阵，因此我们将从一个例子开始，利用渐近误差生成一个可信参数集矩阵，然后再向前推算。

### 7.6.3 使用渐近误差

一旦我们有了一个最佳拟合模型，如果我们还计算了 Hessian 矩阵（如前所述），我们就可以利用估计的渐近误差生成一个充满可信参数向量的矩阵。然后就可以利用这些参数来生成复制的生物量轨迹，并绘制和总结这些轨迹。

在避免极限参考点方面，渔业管理成功的概率标准并不少见。根据不同的可信参数向量，通过大量重复的生物量轨迹，可以估算出有多大比例的预测会达到预期结果。通过将这些预测结果列表，管理者可以选择他们认为合适的风险水平。例如，澳大利亚联邦渔业政策中对可接受风险的明确定义是：“捕捞策略至少在 90% 的时间内将所有商业种群的生物量维持在极限参考点之上”。对这一点的解释是：“种群应在至少 90% 的时间内保持在极限生物量水平之上（即种群在 10 年中有 1 年的时间会低于极限生物量水平 的风险）”（DAWR, 2018a, p10）。

在上一节中，我们已经利用 Fox 剩余产量模型（Polacheck 等，1993）对 *abdat* 数据集进行了最佳模型拟合。正如我们在利用渐近误差描述不确定性时能够生成生物量轨迹一样，我们可以在给定恒定渔获量的条件下，将这些轨迹逐一向前推算，并寻找能产生理想结果的渔获量水平。首先要做的是从最优模型拟合生成多个可信参数向量。我们可以使用函数 parasympt() 来做到这一点。一旦生成了可信参数向量矩阵，我们就可以使用 spmproj() 函数对给定年份数和给定渔获量常数进行动态预测（通过检查代码再次确认其工作原理）。parasympt() 函数只是一个方便的包装，用于调用 rmvnorm() 函数（**mvtnorm** 软件包的一部分）并将结果返回为一个带标记的矩阵，而 spmproj() 函数则稍微复杂一些。为了简化预测，该函数首先扩展了输入鱼类矩阵，以包括预测年份及其恒定渔获量（用 NA 填充未来 cpue）。spmproj() 使用 spm() 函数计算动态变化，而仅以矩阵形式返回模拟生物量。使用 spm() 可能看起来效率不高，但这意味着可以很容易地修改 spmproj() 函数，以返回动力学估算的任何变量。这些变量包括模型生物量、耗竭水平、捕获率和预测的 cpue（当然也可以只从生物量、预测 cpue 和原始数据中得出其他变量）。运行以下代码并检查两个输出对象：*matpar* 包含参数向量，*projs* 包含生物量轨迹行。

# generate parameter vectors from a multivariate normal   
# project dynamics under a constant catch of 900t   
library(mvtnorm)   
matpar <- parasympt(bestmod,N=1000) #generate parameter vectors   
projs <- spmproj(matpar,fish,projyr=10,constC=900)#do dynamics

计算完成后，我们可以总结预测的结果。首先，我们可以使用函数 plotproj() 绘制 1000 个预测。

# Fig 7.30 1000 replicate projections asymptotic errors   
outp <- plotproj(projs,out,qprob=c(0.1,0.5),refpts=c(0.2,0.4))

|  |
| --- |
| 图 7.29: 1000 个预测值，通过使用反向哈希值和平均参数估计值生成 1000 个可信参数向量，并将每 个向量与 10 年不变的 900 吨渔获量后的渔获量向前推算得出。虚线为极限和目标参考点。蓝色垂直线是渔业数据的极限，黑色粗线是最优拟合，与最优线平行的红色细线是各年的第 10 和第 50 个量值。1000 projections derived from the using the inverse hessian and mean parameter estimates to generate 1000 plausible parameter vectors and projecting each vector forward with the fisheries catches followed by 10 years of a constant catch of 900t. The dashed lines are the limit and target reference points. The blue vertical line is the limit of fisheries data, the thick black line is the optimum fit and the thin red lines parallel to the optimum line are the 10th and 50th quantiles across years. |

很明显，10年后，假设动态保持不变，平均900吨的渔获量会导致种群从目前的状态有所下降，但使中值结果接近目标（上虚线），并且10年后不超过10%的轨迹越过极限参考点（LRP）（下细线高于 限制）。通过探索不同的恒定渔获量，将能够发现，如果渔获量增加到 1000 吨，那么 10 年后，第 10 分位数几乎违反了 LRP。将跨越 LRP 的轨迹比例制成表格以生成风险表，将澄清不同拟议的常数渔获量的影响，并有助于选择更具防御性的管理决策。

### 7.6.4 使用 Bootstrap 参数向量

预测的本质是通过最佳模型拟合，结合分析中固有的不确定性估计，生成一个可信的参数向量矩阵。我们也可以不使用假定为多元正态分布的渐近误差，而使用自举法过程来生成所需的参数向量矩阵。就像在分析中描述不确定性一样，我们可以使用 spmboot() 函数来创建所需的参数向量。如果该函数耗时过长，我们可以使用基于 **Rcpp** 的 simpspmC() 函数来加快 1000（或更多）次模型拟合的速度。

#bootstrap generation of plausible parameter vectors for Fox   
reps <- 1000   
boots <- spmboot(bestmod$estimate,fishery=fish,iter=reps,   
 schaefer=FALSE)   
matparb <- boots$bootpar[,1:4] #examine using head(matparb,20)

就象以前一样，我们可以使用这些参数向量来预测渔业的未来，并确定不同恒定捕捞水平对可持续性的任何风险（[图 7.30](#fig-spm31)）。

#bootstrap projections. Lower case b for boostrap Fig7.31   
projb <- spmproj(matparb,fish,projyr=10,constC=900)   
outb <- plotproj(projb,out,qprob=c(0.1,0.5),refpts=c(0.2,0.4))

|  |
| --- |
| 图 7.30: 1000 个预测值（灰色）来自使用自举过程生成的 1000 个可信参数向量，并将每个向量与 10 年 900 吨恒定渔获量之后的渔获量进行向前预测。虚线为极限和目标参考点。蓝色垂直线为渔业数据的极限，黑色粗线为最佳拟合，红线为各年的第 10 和第 50 个量值。 |

投影的灰线与使用渐近误差生成的灰线不同（在中位数附近看起来更紧密），但第 10 和第 50 分位数看起来非常相似。当然，汇总结果基本相同，不过在这种情况下，没有一个预测低于极限参考点（试试 *outb*ltLRP 进行比较）。

### 7.6.5 使用贝叶斯后验样本

正如我们利用渐近误差和自举法获取样本一样，我们也可以从贝叶斯后验中获取样本，生成可信的参数向量。在这种情况下，我们可以使用 do\_MCMC() 函数来进行 MCMC。我们只需要 1000 个可信参数向量，因此我们将从接近最大似然最大值的点开始进行合理的预烧，并使用较大的稀疏率来避免后验分布的连续抽样之间的序列相关性。如前所述，最好使用 **Rcpp** 派生函数 simpspmC() 进行 MCMC，因为我们仍在运行 214.5 万次迭代。由于我们使用的是 Fox 运行的剩余生产模型，其比例因子与 Schaefer 版本使用的比例因子有很大不同。如果您尚未编译 simpspmC() 函数（见附录），请修改以下代码以使用 simpspm()，为提高速度，您可以保留 as.matrix(fish)。

#Generate 1000 parameter vectors from Bayesian posterior   
param <- log(c(r=0.3,K=11500,Binit=3300,sigma=0.05))   
set.seed(444608)   
N <- 1000   
result <- do\_MCMC(chains=1,burnin=100,N=N,thinstep=2048,   
 inpar=param,infunk=negLL,calcpred=simpspmC,   
 calcdat=fish,obsdat=log(fish[,"cpue"]),   
 priorcalc=calcprior,schaefer=FALSE,   
 scales=c(0.065,0.055,0.1,0.475))   
parB <- result[[1]][[1]] #capital B for Bayesian   
cat("Acceptance Rate = ",result[[2]],"\n")

Acceptance Rate = 0.3341834 0.3087928 0.3504304 0.3506508

为了证明生成的 1000 个重复已经失去了它们的序列相关性，并代表了对稳态分布的合理近似，我们可以绘制自相关图和 参数 1000 个重复估计值的轨迹（[图 7.31](#fig-spm32)）。

# auto-correlation, or lack of, and the K trace Fig 7.32   
parset(plots=c(2,1),cex=0.85)   
acf(parB[,2],lwd=2)   
plot(1:N,parB[,2],type="l",ylab="K",ylim=c(8000,19000),xlab="")

|  |
| --- |
| 图 7.31: 从上图可以明显看出，过剩生产模型福克斯版本的 K 参数后验分布的 1000 次抽样中缺乏自相关性。下图中的迹线显示了典型的分散值，但保留了一些更极端的峰值。 |

可以从 MCMC 输出中提取这 1000 个可信参数向量，并通过与之前相同的 spmproj() 和 plotproj() 函数进行处理（[图 7.32](#fig-spm33)）。

# Fig 7.33   
matparB <- as.matrix(parB[,1:4]) # B for Bayesian   
projs <- spmproj(matparB,fish,constC=900,projyr=10) # project them   
plotproj(projs,out,qprob=c(0.1,0.5),refpts=c(0.2,0.4)) #projections

|  |
| --- |
| 图 7.32: 利用贝叶斯后验的 1000 个样本得出的 900 吨恒定渔获量的 1000 个预测值（灰色）。虚线为极限和目标参考点。蓝色垂直线为渔业数据的极限，黑色粗线为最佳拟合，红线为各年的第 10 和第 50 分位数。 |

请注意，[图 7.32](#fig-spm33) 中的中值细红线（第 50 分位数）略微偏离最大似然最佳模型拟合线（黑色）。但是，第 10 分位数相对于 LRP 保持在大致相同的位置，就像在使用渐近误差和自举的分析中观察到的那样。在这里，生物量轨迹的分布范围更广，但管理结果与前两种方法非常相似。

## 7.7 结束语

我们已经比较详细研究了如何使用剩余产量模型为渔业提供管理建议。输出结果可能是未来三至五年不同渔获量或努力量制度下的预期种群结果列表。假设相关管辖区存在某种管理目标，管理者可以做出决定，渔业评估科学家也可以为其结果辩护。当然，鉴于任何渔业数据固有的不确定性和补充量动态的变幻莫测，不可能有确切的保证，但假设种群动态至少与以前的经验相似，那么对结果进行辩护是可能的。

随着气候变化引起的生物生长和成熟过程的改变，我们也可以预料到补充量也会发生变化，因此显然需要更加谨慎。但是，如果大规模的变化是由单一的风暴或其他事件引起的，那将构成一种新的不确定性，而这种不确定性在评估中是没有考虑到的。这就强调了评估科学家必须了解种群所在区域的情况。任何评估，哪怕是简单的评估，都不应仅仅是分析性的，或主要是自动化的。

现实情况是，评估的复杂性和先进性往往与其相对价值相关。只有在渔业可用数据大量增加的情况下，才有可能使用更复杂的模型。因此，鱼类种群有一个自然排序，最有价值的种群通常最受关注。然而，目前在世界各地，人们对为数据贫乏的鱼种提供管理建议的兴趣大增，这通常是受法律要求的驱动。因此，尽管我们只回顾了相对简单的评估方法，但这些方法不应被唾弃或忽视。

## 7.8 附录：使用 Rcpp 代替 simpspm

在贝叶斯分析中，我们希望使用 Fox 剩余生产模型。这当然可以使用函数 simpspm()，通过修改 *schaefer* 参数来实现。但是

library(Rcpp)   
   
cppFunction('NumericVector simpspmC(NumericVector pars,   
 NumericMatrix indat, LogicalVector schaefer) {   
 int nyrs = indat.nrow();   
 NumericVector predce(nyrs);   
 NumericVector biom(nyrs+1);   
 double Bt, qval;   
 double sumq = 0.0;   
 double p = 0.00000001;   
 if (schaefer(0) == TRUE) {   
 p = 1.0;   
 }   
 NumericVector ep = exp(pars);   
 biom[0] = ep[2];   
 for (int i = 0; i < nyrs; i++) {   
 Bt = biom[i];   
 biom[(i+1)] = Bt + (ep[0]/p)\*Bt\*(1 - pow((Bt/ep[1]),p)) -   
 indat(i,1);   
 if (biom[(i+1)] < 40.0) biom[(i+1)] = 40.0;   
 sumq += log(indat(i,2)/biom[i]);   
 }   
 qval = exp(sumq/nyrs);   
 for (int i = 0; i < nyrs; i++) {   
 predce[i] = log(biom[i] \* qval);   
 }   
 return predce;   
}')

# 参考文献

Beverton, Raymond J. H., 和 Sidney J. Holt. 1993. *On the Dynamics of Exploited Fish Populations*. Springer Netherlands. <https://doi.org/10.1007/978-94-011-2106-4>.

Birkes, David, 和 Yadolah Dodge. 1993. 《Alternative Methods of Regression》. *Wiley Series in Probability and Statistics*, 六月. <https://doi.org/10.1002/9781118150238>.

Burnham, Kenneth P., 和 David R. Anderson, 编. 2004. *Model Selection and Multimodel Inference*. Springer New York. <https://doi.org/10.1007/b97636>.

Chambers, John M. 2008. *Software for Data Analysis*. Springer New York. <https://doi.org/10.1007/978-0-387-75936-4>.

———. 2017. *Extending R*. Chapman; Hall/CRC. <https://doi.org/10.1201/9781315381305>.

Crawley, Michael J. 2007. 《The R Book》, 四月. <https://doi.org/10.1002/9780470515075>.

Elder, R. D. 1979. 《Equilibrium Yield for the Hauraki Gulf Snapper Fishery Estimated from Catch and Effort Figures, 196074》. *New Zealand Journal of Marine and Freshwater Research* 13 (1): 31–38. <https://doi.org/10.1080/00288330.1979.9515778>.

Fournier, Daid A, John Hampton, 和 John R Sibert. 1998. 《MULTIFAN-CL: A Length-Based, Age-Structured Model for Fisheries Stock Assessment, with Application to South Pacific Albacore, *Thunnus Alalunga*》. *Canadian Journal of Fisheries and Aquatic Sciences* 55 (9): 2105–16. <https://doi.org/10.1139/f98-100>.

Fournier, David A., Hans J. Skaug, Johnoel Ancheta, James Ianelli, Arni Magnusson, Mark N. Maunder, Anders Nielsen, 和 John Sibert. 2012. 《AD Model Builder: Using Automatic Differentiation for Statistical Inference of Highly Parameterized Complex Nonlinear Models》. *Optimization Methods and Software* 27 (2): 233–49. <https://doi.org/10.1080/10556788.2011.597854>.

Haddon, Malcolm. 2011. *Modelling and Quantitative Methods in Fisheries*. Chapman; Hall/CRC. <https://doi.org/10.1201/9781439894170>.

Helidoniotis, Fay, 和 Malcolm Haddon. 2013. 《Growth Models for Fisheries: The Effect of Unbalanced Sampling Error On Model Selection, Parameter Estimation, and Biological Predictions》. *Journal of Shellfish Research* 32 (1): 223–35. <https://doi.org/10.2983/035.032.0129>.

Hilborn, Ray. 1979. 《Comparison of Fisheries Control Systems That Utilize Catch and Effort Data》. *Journal of the Fisheries Research Board of Canada* 36 (12): 1477–89. <https://doi.org/10.1139/f79-215>.

Hilborn, Ray, 和 Carl J. Walters. 1992. *Quantitative Fisheries Stock Assessment*. Springer US. <https://doi.org/10.1007/978-1-4615-3598-0>.

Kristensen, Kasper, Anders Nielsen, Casper W. Berg, Hans Skaug, 和 Bradley M. Bell. 2016. 《**TMB**: Automatic Differentiation and Laplace Approximation》. *Journal of Statistical Software* 70 (5). <https://doi.org/10.18637/jss.v070.i05>.

MacCall, Alec D. 2009. 《Depletion-Corrected Average Catch: A Simple Formula for Estimating Sustainable Yields in Data-Poor Situations》. *ICES Journal of Marine Science* 66 (10): 2267–71. <https://doi.org/10.1093/icesjms/fsp209>.

Matloff, Norman. 2011. *The Art of R Programming: A Tour of Statistical Software Design*. 1st edition. San Francisco: No Starch Press.

May, Robert, 和 Angela R. McLean, 编. 2007. *Theoretical Ecology*. Oxford University Press. <https://doi.org/10.1093/oso/9780199209989.001.0001>.

Murrell, Paul. 2018. *R Graphics, Third Edition*. Chapman; Hall/CRC. <https://doi.org/10.1201/9780429422768>.

Pitcher, T. J., 和 P. D. M. Macdonald. 1973. 《Two Models for Seasonal Growth in Fishes》. *The Journal of Applied Ecology* 10 (2): 599. <https://doi.org/10.2307/2402304>.

Polacheck, Tom, Ray Hilborn, 和 Andre E. Punt. 1993. 《Fitting Surplus Production Models: Comparing Methods and Measuring Uncertainty》. *Canadian Journal of Fisheries and Aquatic Sciences* 50 (12): 2597–607. <https://doi.org/10.1139/f93-284>.

Prager, Michael. 1994. 《A suite of extensions to a nonequilibrium surplus-production model》. *Fishery Bulletin* 92 (一月): 374–89.

Punt, Andre E., 和 Ray Hilborn. 1997. 《Fisheries Stock Assessment and Decision Analysis: The Bayesian Approach》. *Reviews in Fish Biology and Fisheries* 7 (1): 35–63. <https://doi.org/10.1023/A:1018419207494>.

QUINN, TERRANCE J. 2003. 《RUMINATIONS ON THE DEVELOPMENT AND FUTURE OF POPULATION DYNAMICS MODELS IN FISHERIES》. *Natural Resource Modeling* 16 (4): 341–92. <https://doi.org/10.1111/j.1939-7445.2003.tb00119.x>.

Quinn, Terrance J., 和 Richard B. Deriso. 1999. *Quantitative Fish Dynamics*. Biological Resource Management. Oxford, New York: Oxford University Press.

Russell, E. S. 1942. *The Overfishing Problem*. CUP Archive.

Saila, S. B., J. H. Annala, J. L. McKoy, 和 J. D. Booth. 1979. 《Application of Yield Models to the New Zealand Rock Lobster Fishery》. *New Zealand Journal of Marine and Freshwater Research* 13 (1): 1–11. <https://doi.org/10.1080/00288330.1979.9515775>.

Schaefer, M. 1991. 《Some Aspects of the Dynamics of Populations Important to the Management of the Commercial Marine Fisheries》. *Bulletin of Mathematical Biology* 53 (1-2): 253–79. <https://doi.org/10.1016/s0092-8240(05)80049-7>.

Schaefer, M. B. 1957. 《A study of the dynamics of the fishery for yellowfin tuna in the eastern tropical pacific ocean》. 收入. <https://www.semanticscholar.org/paper/A-study-of-the-dynamics-of-the-fishery-for-tuna-in-Schaefer/29677d4a85d251d68d645584b2505c51c1ef1728>.

Stearns, SC. 1992. *The evolution of life histories*. Oxford University Press.

Stearns, Stephen C. 1977. 《The Evolution of Life History Traits: A Critique of the Theory and a Review of the Data》. *Annual Review of Ecology and Systematics* 8 (1): 145–71. <https://doi.org/10.1146/annurev.es.08.110177.001045>.

Venables, W. N., 和 B. D. Ripley. 2002. *Modern Applied Statistics with S*. Springer New York. <https://doi.org/10.1007/978-0-387-21706-2>.

Wickham, Hadley. 2019. *Advanced R*. Chapman; Hall/CRC. <https://doi.org/10.1201/9781351201315>.

Winker, Henning, Felipe Carvalho, 和 Maia Kapur. 2018. 《JABBA: Just Another Bayesian Biomass Assessment》. *Fisheries Research* 204 (八月): 275–88. <https://doi.org/10.1016/j.fishres.2018.03.010>.

Xie, Yihui. 2016. 《bookdown: Authoring Books and Technical Documents with R Markdown》. The R Foundation. <https://doi.org/10.32614/cran.package.bookdown>.