

国内首部系统介绍Clojure数据分析技术的著作，内容全面而深入，为高效利用Clojure进行数据分析提供最佳指导

通过大量典型Clojure数据分析案例全面解析Clojure数据分析的各种技术细节、方法和最佳实践，实战性强



技术丛书



Clojure Data Analysis Cookbook

Clojure数据分析秘笈

(美) Eric Rochester 著

刘德海 张玫◎译



机械工业出版社
China Machine Press



技术丛书

Clojure Data Analysis Cookbook

Clojure数据分析秘笈

(美) Eric Rochester 著

刘德海 张玫◎译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Clojure 数据分析秘笈 / (美) 罗切斯特 (Rochester, E.) 著; 刘德海, 张玫译. —北京: 机械工业出版社, 2014.8

(大数据技术丛书)

书名原文: Clojure Data Analysis Cookbook

ISBN 978-7-111-47326-8

I. C… II. ①罗… ②刘… ③张… III. 程序语言—语言设计 IV. TP312

中国版本较长书馆 CIP 数据核字 (2014) 第 153705 号

本书版权登记号: 图字: 01-2013-6904

Eric Rochester: Clojure Data Analysis Cookbook (ISBN: 978-1-78216-264-3)

Copyright © 2013 Packt Publishing. First published in the English language under the title “Clojure Data Analysis Cookbook”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2014 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

Clojure 数据分析秘笈

[美] Eric Rochester 著

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 秦 健

责任校对: 董纪丽

印 刷: 三河市宏图印务有限公司

版 次: 2014 年 8 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 16.5

标准书号: ISBN 978-7-111-47326-8

定 价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京市大成律师事务所 韩光 / 邹晓东

数据无处不在！随着数据变得越来越普遍，人们利用数据的愿望也愈发强烈。数据中隐藏着很多有价值的信息，如潜在的购买者、用户的浏览模式、人口统计信息等。只要能够发现数据中的信息，就会获得对事物深刻的认知并且做出更明智的决策。

本书会帮助你做到这一点。

Clojure 编程语言会帮上忙。Clojure 由 Rich Hickey 于 2007 年首次发布，是 Lisp 语言大家庭的一员，因此它继承了 Lisp 语言所特有的功能性和灵活性。Clojure 也是一门函数式编程语言，因此 Clojure 程序简单易懂。另外，Clojure 在并发和并行编程方面表现突出，在分析数据时，它的所有这些特性会促使这一过程变得更加简单迅速。

许多强大的库进一步提升了 Clojure 在数据分析方面的实用性。Incanter 提供了处理数据并执行统计分析的实用环境；Cascalog 是覆盖 Hadoop 和 Cascading 技术的一个简单易用的包装器。最后，在准备发布结果时，ClojureScript（一个可以生成 JavaScript 的 Clojure 实现）可以帮助以有效且有说服力的方式对数据进行可视化。

此外，Clojure 运行在 Java 虚拟机（JVM）上，因此为 Java 编写的库也可以用于 Clojure。这使得 Clojure 可以非常高效地解决大多数领域的问题。

希望本书可以为你提供分析数据过程中所需要的工具和技术。

本书主要内容

第 1 章涉及如何从不同种类数据源中读入数据，这些数据源有 CSV 文件、网页和链接语义网数据。

第 2 章提供标准化数据、修正拼写错误以及使用大数据集的策略和实现。把数据塑造成一种可供使用的形式是数据分析的一个重要阶段，但这一过程经常被忽视。

第 3 章包含 Clojure 的并发特性和如何通过这些特性简化程序。

第 4 章包含通过 Clojure 的并行处理能力提高处理数据的速度。

第 5 章包含在 Hadoop 和 Cascading 库上使用 Cascalog 处理分布在多台计算机上的大量数据。该章介绍的最后一个方法使用 Pallet 在亚马逊 EC2 服务上执行一个简单的分析。

第 6 章介绍使用 Incanter 数据集的基础知识。数据集是 Incanter 使用的核心数据结构，理解它们是高效使用 Incanter 的基础。

第 7 章包含在数据分析中用到的一系列统计处理过程和测试。其中一些很简单，例如生成汇总统计。其他的可能更复杂些，例如执行线性回归和使用本福德定律审计数据。

第 8 章讲解如何配置 Clojure 以便与 Mathematica 和 R 交互。它们都是强大的数据分析系统，有时可能需要使用它们。该章将展示如何使这些分析系统一起工作，以及这些工具交互时我们可以完成的任务。

第 9 章包含更多高级的机器学习技术。在该章中，我们将主要使用 Weka 的机器学习库，以及讨论如何使用它和其中内置的数据结构，另外还将展示机器学习算法的一些方法。

第 10 章将展示如何在 Incanter 中生成图和进行可视化。对于探索和研究数据来说这些很重要，同时对于发布和展示结果来也非常重要。

第 11 章将展示如何安装一个简单的网页应用来展现数据分析的发现。该章将介绍利用强大的 D3 可视化库的一些方法。

阅读本书需要做的准备

学习本书需要使用 Java Development Kit (JDK) 软件，可以从 <http://www.oracle.com/technetwork/java/javase/downloads/index.html> 下载获得。JDK 是在 Java 平台上运行和开发程序的必需品。

Leiningen 2 是另外一个需要用到的软件，可以从 <https://github.com/technomancy/leiningen> 下载安装。Leiningen 2 是用来管理 Clojure 项目和其他依赖的一个工具。很快它将成为 Clojure 社区事实上的标准项目管理工具。

全书会使用许多其他 Clojure 和 Java 库，包括 Clojure 本身。当需要这些库的时候，可以通过 Leiningen 下载。

同时还需要一款文本编辑器，或者集成开发环境 (Integrated Development Environment, IDE)。如果你已经有一款自己喜欢使用的文本编辑器，那么可以直接使用。在使用自己喜欢的开发环境时，推荐浏览 <http://dev.clojure.org/display/doc/Getting+Started> 中的提示和插件。如果你没有偏好，建议考虑使用带有 Counterclockwise 的 Eclipse，安装指导可以在 <http://dev.clojure.org/display/doc/Getting+Started+with+Eclipse+and+Counterclockwise> 中找到。

这就是本书需要的全部软件。然而，在全书中的有些地方，有一些方法会使用其他的软件。例如在第 8 章中，与 Mathematica 相关的内容很显然会使用 Mathematica，同样与 R 相关的内容会用到 R。但是这些软件不会在书中的其他部分用到，并且对这些方法是否感兴趣

可能取决于你是否安装了这些软件。

本书读者对象

本书是为熟悉 Clojure 语言并且想将其应用在数据分析过程中的程序员和数据科学家准备的。本书并不是 Clojure 教程（现在已经有很多介绍 Clojure 的优秀著作），因此读者需要对 Clojure 这门语言熟悉，但不必是 Clojure 语言方面的专家。

排版约定

在本书中，你会发现用以区分内容的不同文本样式。下面是这些样式的例子和含义说明。

当需要读者注意代码块中的特定部分时，相关的行或者项目会设置为加粗显示。

```
[ring.middleware.file-info :only (wrap-file-info)]
[ring.middleware.stacktrace :only (wrap-stacktrace)]
[ring.util.response :only (redirect)]


### hiccup core element page


[hiccup.middleware :only (wrap-base-url)]))
```

所有命令行的输入或者输出都按以下方式展示：

```
$ lein cljsbuild auto
Compiling ClojureScript.
Compiling "resources/js/scripts.js" from "src-cljs"...
Successfully compiled "resources/js/script.js" in 4.707129 seconds.
```

新术语和重要词汇会以加粗的形式展示。

示例代码下载

你可以从 <http://www.packtpub.com> 上你的账户中下载购买的所有 Packt 书籍中的示例代码文件。如果你通过其他渠道购书，可以访问 <http://www.packtpub.com/support> 并注册，我们会通过邮件直接将这些文件发送给你。

勘误表

尽管我们尽最大努力来保证内容的准确性，但错误不可避免还会出现。如果在书中发现错误（可能是文字或者代码错误），请告知我们。这样一来，你可以帮助其他读者免受错误干扰，并帮助改进本书后续的版本。如果发现错误，请通过访问 <http://www.packtpub.com/submit-errata>，点击 **errata submission form** 链接，输入错误详情向我们报告错误。一旦你的勘误验证通过，我们将采纳你的提交的意义，然后将勘误上传到我们的网站上，或者加入

对应主题的勘误栏目下现存的勘误列表中。任何现存的勘误表都可以从 <http://www.packtpub.com/support> 上点击浏览。

问题

当你有关于本书任何方面的问题时，请通过 questions@packtpub.com 联系，我们会尽最大努力解决它。

About the Reviewers 关于技术审校者

Jan Borgelin 是一位拥有超过 10 年软件开发经验的技术怪才。曾在企业软件领域不同职位工作，现在担任 BA 集团有限公司（芬兰一家 IT 咨询公司）CEO 和高级咨询师。在过去的两年里，他参与了函数式编程语言的开发工作并对 Clojure 很感兴趣。

Thomas A. Faulhaber, Jr. 是 Infolace 的负责人（www.infolace.com，旧金山一家咨询公司）。Infolace 帮助从初创业型公司到国际品牌公司的客户将原始数据转换为信息并将信息转换为行动。在 Thomas 的职业生涯中，他曾参与开发高性能 TCP/IP 系统、大规模科学可视化系统、能源交易系统等。

在 Clojure 和 Incanter 的最早期阶段，他就成为其贡献者和使用者。Clojure 与其生态系统（包括代码和人）的能力是 Thomas 的一个重要贡献。

Charles Norton 有超过 25 年的编程经验，涉及领域从工厂自动化应用和固件到网络中间件，现在是波士顿地区政府的一名程序员和应用专家。他维护并开发了一系列支持金融、健康保险和供水设施管理的软件应用。这些系统由多种语言实现，其中也包括 Clojure。

Miki Tebeka 有超过 10 年的软件开发经验。他开发了多种产品，从新闻交易系统的汇编程序和链接器到云架构。他现在就职于 Adconion，该公司每月处理超过 60 亿个事件。在他闲暇的时间里，他是多个开源社区的活跃者。

目 录 *Contents*

前言

关于技术审校者

第 1 章 导入分析数据	1
1.1 引言	1
1.2 新建项目	1
1.3 将 CSV 数据读入 Incanter 数据集	2
1.4 将 JSON 数据读入 Incanter 数据集	4
1.5 使用 Incanter 读入 Excel 数据	5
1.6 从 JDBC 数据库读取数据	6
1.7 将 XML 数据读入 Incanter 数据集	9
1.8 从网页表中抓取数据	11
1.9 从网页中抓取文本数据	15
1.10 读取 RDF 数据	17
1.11 使用 SPARQL 读取 RDF 数据	20
1.12 整合不同格式的数据	24
第 2 章 清洗和校验数据	29
2.1 引言	29
2.2 使用正则表达式清洗数据	30
2.3 使用同义词映射保持一致性	31
2.4 识别并去除重复数据	33
2.5 标准化数字格式	35

2.6	调整词频值的度量	36
2.7	标准化日期和时间	38
2.8	大数据集的延迟处理	40
2.9	大数据集抽样	42
2.10	修正拼写错误	43
2.11	解析自定义数据格式	46
2.12	使用 Valip 校验数据	49
第 3 章	使用并发编程管理复杂度	51
3.1	引言	51
3.2	使用 STM 管理程序复杂度	52
3.3	使用 agent 管理程序复杂度	55
3.4	使用 commute 获得更好的性能	57
3.5	将 agent 和 STM 结合使用	58
3.6	使用 ensure 维护一致性	60
3.7	将安全的副作用引入 STM 中	63
3.8	使用 validator 维护数据一致性	65
3.9	使用 watcher 追踪处理过程	68
3.10	使用 watcher 调试并发程序	70
3.11	从 agent 中错误恢复	71
3.12	使用 sized queue 管理输入	73
第 4 章	使用并行编程提高性能	74
4.1	引言	74
4.2	使用 pmap 并行处理	75
4.3	使用 Incanter 并行处理	77
4.4	将蒙特卡罗模拟进行划分使 pmap 性能提升	79
4.5	使用模拟退火算法最优化分块大小	83
4.6	使用 reducers 并行处理	86
4.7	使用 reducers 生成在线统计	89
4.8	使用 OpenCL 和 Calx 驾驭你的 GPU	91
4.9	使用类型提示	94
4.10	使用 Criterion 制定基准	96

第 5 章 使用 Cascalog 进行分布式数据处理	100
5.1 引言	100
5.2 使用 Cascalog 和 Hadoop 分布式处理	101
5.3 使用 Cascalog 查询数据	104
5.4 使用 Apache HDFS 分布数据	105
5.5 使用 Cascalog 解析 CSV 文件	108
5.6 使用 Cascalog 执行复杂查询	110
5.7 使用 Cascalog 聚合数据	112
5.8 定义新 Cascalog 操作符	113
5.9 组成 Cascalog 查询	116
5.10 处理 Cascalog 工作流中的错误	118
5.11 使用 Cascalog 转换数据	119
5.12 使用 Pallet 在云上执行 Cascalog 查询	120
第 6 章 使用 Incanter 数据集	126
6.1 引言	126
6.2 加载 Incanter 样例数据集	126
6.3 将 Clojure 数据结构加载到数据集中	127
6.4 使用 view 交互式查看数据集	129
6.5 将数据集转换为矩阵	130
6.6 在 Incanter 中使用 infix 公式	132
6.7 使用 \$ 选择列	133
6.8 使用 \$ 选择行	135
6.9 使用 \$where 过滤数据集	136
6.10 使用 \$group-by 对数据分组	138
6.11 将数据集另存为 CSV 和 JSON	139
6.12 使用 \$join 进行多数据集投影	141
第 7 章 使用 Incanter 准备并执行统计数据分析	144
7.1 引言	144
7.2 使用 \$rollup 生成汇总统计	144
7.3 通过变量差别展示变化	146

7.4	调整变量以简化变量关系	148
7.5	使用 Incanter Zoo 处理时间序列数据	150
7.6	平滑数据以降低噪声	152
7.7	使用 bootstrapping 验证抽样统计	154
7.8	线性关系建模	156
7.9	非线性关系建模	158
7.10	多峰贝叶斯分布建模	162
7.11	使用本福德定律找出数据错误	165
第 8 章	使用 Mathematica 和 R	167
8.1	引言	167
8.2	在 Mac OS X 和 Linux 系统中配置 Mathematica 与 Clojuratica 的交互环境	168
8.3	在 Windows 系统中配置 Mathematica 与 Clojuratica 的交互环境	170
8.4	在 Clojuratica 中调用 Mathematica 函数	172
8.5	在 Clojuratica 中向 Mathematica 发送矩阵	173
8.6	在 Clojuratica 中运行 Mathematica 脚本	174
8.7	从 Mathematica 中创建函数	175
8.8	在 Mathematica 中并行处理函数	176
8.9	配置 R 与 Clojure 交互	177
8.10	在 Clojure 中调用 R 的函数	179
8.11	将向量传入 R	180
8.12	在 Clojure 中执行 R 文件	181
8.13	在 Clojure 中使用 R 绘图	183
第 9 章	聚类、分类和使用 Weka	185
9.1	引言	185
9.2	将 CSV 和 ARFF 文件加载到 Weka 中	185
9.3	在 Weka 数据集中对列进行过滤和重命名	187
9.4	使用 K-means 聚类发现成组数据	190
9.5	在 Weka 中寻找层次聚类结构	195
9.6	在 Incanter 中使用 SOM 聚类	197
9.7	使用决策树分类数据	199
9.8	使用朴素贝叶斯分类器分类数据	201

9.9 使用支持向量机分类数据..... 203

9.10 使用 Apriori 算法发现数据中的关联 205

第 10 章 使用 Incanter 绘图..... 208

10.1 引言..... 208

10.2 使用 Incanter 创建散点图 209

10.3 使用 Incanter 创建柱形图 211

10.4 在柱形图中绘制非数值型数据..... 212

10.5 使用 Incanter 创建直方图 213

10.6 使用 Incanter 绘制函数 215

10.7 为 Incanter 图表加入方程式..... 216

10.8 为散点图加入直线 217

10.9 使用 JFreeChart 定制图表..... 218

10.10 将 Incanter 图保存为 PNG 格式..... 220

10.11 使用 PCA 绘制多维数据..... 221

10.12 使用 Incanter 创建动态图表..... 223

第 11 章 创建网页图表..... 225

11.1 引言..... 225

11.2 使用 Ring 和 Compojure 提供数据 226

11.3 使用 Hiccup 创建 HTML 网页 229

11.4 配置和使用 ClojureScript..... 231

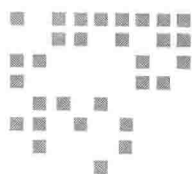
11.5 用 NVD3 创建散点图 234

11.6 用 NVD3 创建条形图..... 239

11.7 用 NVD3 创建直方图..... 242

11.8 使用力向布局进行图像可视化 244

11.9 用 D3 创建交互式可视化..... 248



导入分析数据

1.1 引言

如果没有大量数据是无法进行数据分析的，因此任何项目的第一步都是评估有什么样的数据和需要什么样的数据。一旦知道需要什么样的数据，就要想办法得到它。

本章和本书中的许多方法使用 Incanter (<http://incanter.org/>) 导入数据并规定使用 Incanter 数据集。Incanter 是 Clojure 中完成统计分析和图形化显示的库，其作用类似于 R。Incanter 可能并不适用于所有任务（后面将使用 Weka 库进行聚类 and 机器学习），但它仍是在 Clojure 中进行数据分析的重要工具。本章介绍如何收集数据和使数据适用于 Clojure。首先来看一下如何创建一个新的项目。我们将从以逗号分隔值（Comma-Separated Value, CSV）的简单数据格式开始，然后学习使用 JDBC 从关系数据库中读取数据。最后考虑更复杂的数据源，例如从网络上抓取的数据和链接数据（RDF）。

1.2 新建项目

本书将用到许多第三方库和外部依赖包，因此需要一款下载并记录它们的工具，也需要一款工具用于搭建环境并启动交互式终端解释器（REPL），以便查看代码或者执行程序。

可以使用 Leiningen (<http://leiningen.org/>) 完成这项工作，它已经成为标准的自动打包和管理系统。

1.2.1 准备工作

访问 Leiningen 官方网站 (<http://leiningen.org/>) 并下载 lein 脚本, 执行该脚本会下载 Leiningen 所需的 JAR 文件。安装说明很清晰, 整个安装过程比较简单。

1.2.2 具体实现

用 lein new 命令生成一个新项目, 将项目名称作为参数传入:

```
$ lein new getting-data
Generating a project called getting-data based on the 'default' template.
To see other templates (app, lein plugin, etc), try 'lein help new'.
```

现在你会发现一个名为 getting-data 的子目录, 其中将包含 getting-data.core 命名空间和测试用的桩程序。

下载样例代码

你可以从 <http://www.packtpub.com> 你的账户中下载购买的所有 Packt 书籍中的样例代码文件。如果你通过其他渠道购书, 可以访问 <http://www.packtpub.com/support> 并注册, 这些文件会通过邮件直接发送给你。

1.2.3 实现原理

新建项目目录下有一个名为 project.clj 的文件, 这个文件中包含关于该项目的元数据信息: 项目名、版本和许可。文件中也包含代码需要使用的依赖列表。该文件中所使用的规范允许在 Maven 仓库目录和 Clojure 库目录 (Clojars, <https://clojars.org/>) 搜索下载项目的依赖包。

```
(defproject getting-data "0.1.0-SNAPSHOT"
  :description "FIXME: write description"
  :url "http://example.com/FIXME"
  :license {:name "Eclipse Public License"
            :url "http://www.eclipse.org/legal/epl-v10.html"}
  :dependencies [[org.clojure/clojure "1.4.0"]])
```

在每种方法的准备部分, 都可以从这个文件的 :dependencies 部分看到项目需要列出的库。

1.3 将 CSV 数据读入 Incanter 数据集

以逗号分隔值 (CSV) 是最简单的数据格式之一, 并且这种数据格式应用非常普遍。

Excel 可以直接读写 CSV 文件，而且多数数据库也可以。由于 CSV 文件就是无格式的文本，因此使用任何编程语言都很容易生成或者访问它。

1.3.1 准备工作

首先，确保加载了正确的库。Leiningen (<https://github.com/technomancy/leiningen>) 的项目文件 `project.clj` 应该包含以下依赖（也可以使用更新的版本）：

```
:dependencies [[org.clojure/clojure "1.4.0"]
               [incanter/incanter-core "1.4.1"]
               [incanter/incanter-io "1.4.1"]]
```

另外，在 REPL 或者代码文件中，包含以下代码：

```
(use 'incanter.core
     'incanter.io)
```

最后，我将包含以下数据的文件命名为 `data/small-sample.csv`。

```
Gomez,Addams,father
Morticia,Addams,mother
Pugsley,Addams,brother
Wednesday,Addams,sister
...
```

这个文件可以从 <http://www.ericrochester.com/clj-data-analysis/data/small-sample.csv> 下载，还有一个带有标题行的版本，可以从 <http://www.ericrochester.com/clj-data-analysis/data/small-sample-header.csv> 下载。

1.3.2 具体实现

1. 使用 `incanter.io/read-dataset` 函数。

```
user=> (read-dataset "data/small-sample.csv")
[:col0 :col1 :col2]
["Gomez" "Addams" "father"]
["Morticia" "Addams" "mother"]
["Pugsley" "Addams" "brother"]
["Wednesday" "Addams" "sister"]
...
```

2. 如果 CSV 文件中有标题行，在调用 `read-dataset` 函数时应包含 `:header true`。

```
user=> (read-dataset "data/small-sample-header.csv" :header true)
[:given-name :surname :relation]
["Gomez" "Addams" "father"]
["Morticia" "Addams" "mother"]
["Pugsley" "Addams" "brother"]
```


1.3.3 实现原理

通过 Clojure 和 Incanter 可以使许多普通工作变得简单，下面是一个很好的例证。

读入外部数据（在本例中来自 CSV 文件），并将其加载到 Incanter 数据集中。在 Incanter 中，一个数据集就是一张表，类似于电子表格中的一个表单或者数据库中的表。每一列包含一个数据域，每一行包含一条数据。有些列包含字符串数据（本例中所有列都是字符串数据），有些列包含日期或者数值数据。如果一列包含数值数据，Incanter 会自动将其检测出来并将其转换为 Java 中的 int 或者 double 数据格式。Incanter 解决了导入数据过程中遇到的许多令人头疼的问题。

1.3.4 更多信息

如果你不想涉及 Incanter，比如当你不想添加依赖时，使用 data.csv 也是比较简单的 (<https://github.com/clojure/data.csv>)。后续的章节中会使用这个库，例如 2.8 节。

1.3.5 参阅

□ 第 6 章

1.4 将 JSON 数据读入 Incanter 数据集

另一个越来越流行的数据格式是 JavaScript Object Notation (JSON，详见 <http://json.org/>)。与 CSV 相似，JSON 也是无格式的文本，因此程序容易处理。它提供了相对于 CSV 文件关于数据的更多信息，但是代价是更冗长。JSON 格式允许用更复杂的方式使数据结构化，例如层次结构或者序列层次结构。

由于 JSON 是比 CSV 更完善的数据模型，因此使用时可能需要转换数据。在这种情况下，可以将感兴趣的信息取出，在传入 Incanter 之前精简嵌入的映射。然而这种方法仅用于处理相当简单的数据结构。

1.4.1 准备工作

首先，在 Leiningen 的 project.clj 文件中包含以下依赖：

```
:dependencies [[org.clojure/clojure "1.4.0"]
                [incanter/incanter-core "1.4.1"]
                [org.clojure/data.json "0.2.1"]]
```