**语言数据科学研究教程**

**目录大纲**

|  |  |
| --- | --- |
| 入门篇 | |
| 第一章 | 数据科学简介 |
| index | 1.1 数据科学发展简史  1.2 数据科学家  1.3 数据科学工作流程 |
| 第二章 | R语言简介 |
| 1 | 2.1 R语言发展简史  2.2 R语言/R-studio及软件包的安装  2.3 R-studio界面及R脚本和R-markdown文件创建  2.4 R语言的数据结构 |
| 基础篇 | |
| 第三章 | 数据整理 |
| 2 | 3.1 数据与变量  3.2 数据清理  3.3 变量筛选筛选 select filter  3.3 数据重塑 spread gather  3.4 数据集成 inner\_join |
| 第四章 | 探索性数据分析 |
| 3 | 4.1 什么是探索性数据分析  4.2数据转化与描述性统计mutate summarise  4.3 数据可视化  4.4 统计建模 |
| 进阶篇 | |
| 第五章 | 语言实验数据分析 |
| 4 | 感知同化  5.1数据导入  5.2 数据整理  5.3 数据转化  5.4 数据可视化  5.4 数据建模 |
| 第六章 | 文本数据分析 |
| 5 | 6.1 数据导入  6.2 数据整理  6.3 词汇分析：词频分析、共现分析、搭配分析（侦探文学）  6.4 文本风格分析 |
| 第七章 | 语音数据分析 |
| 6 | 7.1 构建语音库  7.2 时长数据提取与分析  7.3 共振峰数据提取与分析  7.4 基频数据提取与分析 |
| 高阶篇 | |
| 第八章 | 统计建模基础 |
| 7 | 8.1 卡方检验  8.2 t检验与方差检验  8.3 相关分析  8.4 逻辑回归 |
| 第九章 | 机器学习建模基础 |
| 8 | 9.1 监督学习：例子二语英语信息焦点  9.2 非监督学习：翻译文体风格 |
| 实战篇 | |
| 第十章  9 | 实验行为数据分析案例  10.1研究背景+研究问题  10.2数据整理+描述性统计  10.3数据可视化  10.4统计建模-混合模型  10.5结论 |
| 第十一章  10 | 文本语料数据分析案例  案例一、狄仁杰小说翻译文体风格  11.1研究背景  11.2 数据加工  11.3 可视化  11.4 统计建模  11.5 结论  案例二 基于语料库的汉语失语症核心词表研制 |
| 第十二章 | 语音声学数据分析案例  案例一、越南语南北部方言声调实验  12.1研究背景+研究问题  12.2 数据整理+描述性统计  12.3 可视化  12.4 统计建模-广义增性模型  12.5 结论 |
| 第十三章 | 机器学习建模案例  预测声调学习难点  13.1研究背景+研究问题  13.2 数据整理+描述性统计  13.3 可视化  13.4 统计建模-广义增性模型  13.5 结论 |
| 第十四章 | 文献计量分析案例  语音感知研究  14.1研究背景+研究问题  14.2 数据整理+描述性统计  14.3 可视化  14.4 统计建模-广义增性模型  14.5 结论 |
| 第十五章  14 | 数据可视化  15 |
| 第十六章 | R语言学习资源 |
|  |  |

﻿# 导入文本数据

style.corpus <- load.corpus(files = "all", corpus.dir = "~/Nutstore Files/310\_Tutorial/Language data science/data/ch6/detectives",

encoding = "UTF-8")

# 对文本进行分词

tokenized.corpus <- txt.to.words.ext(style.corpus,

preserve.case = TRUE)

# 根据研究目的我们可以设置一个词表去除一些干扰词。比如我们想排除文本中高频专有名词的干扰。

#proper.noun = c("DEE","Judge","Dee","")

# clean.corpus = delete.stop.words(tokenized.corpus,

# stop.words = proper.noun)

# 一些研究认为代词对于计算文体风格有干扰，比如如果是一个大女主的小说，那么女性代词就会很多。但是这些不能代表作者风格，只能代表作品风格。

clean.corpus = delete.stop.words(tokenized.corpus,

stop.words = stylo.pronouns(corpus.lang = "English"))

corpus.char.1 <- txt.to.features(clean.corpus,

ngram.size = 1,

features = "w")

# 将数据划分成1000个词一份

sliced.corpus.char.1 <- make.samples(corpus.char.1,

sampling = "normal.sampling",

sample.size = 1000)