

数据框架(Data Frames):

创建新数据集: 使用 `c()`, `matrix()`, 和 `data.frame()` 函数。

数据框架的结构: 利用 `mode()` 和 `str()` 函数查看数据框架的结构。

访问数据框架: 使用 `$`, `[]`, 和 `[[]]` 进行数据访问。

NA 值处理:

使用 `subset()` 函数选择数据子集。

使用 `complete.cases()` 函数移除含有 NA 的行。

数据框架合并:

使用 `rbind()` 和 `cbind()` 函数添加行或列。

使用 `merge()` 函数合并具有至少一个共同列名的两个数据框架。

dplyr 包的使用:

`filter()`: 根据条件选择行。

`select()`: 从数据框中选择变量。

`mutate()`: 基于现有列计算新列。

`summarise()`: 计算变量的单数摘要。

`arrange()`: 更改行的顺序。

数据预处理:

数据导入: 使用 `readxl` 包和 `read_excel()` 函数导入数据。

数据筛选与选择: 使用管道操作符 `|>` 和 `filter()`, `select()` 函数。

数据变换: 使用 `mutate()` 和 `ifelse()` 函数处理和转换数据。

分组操作: 使用 `group_by()` 和 `summarise()` 函数进行分组摘要统计。

dplyr 与基础 R 的对比:

使用 `tapply()` 函数在基础 R 中进行类似 `dplyr` 的操作。

性能比较:

使用 `rbenchmark` 包比较 `dplyr` 和基础 R 的性能。

教训和结论:

数据框架和 `tibbles` 是 R 中处理矩形数据集的有用数据结构。

`dplyr` 可以轻松地对数据框架进行数据操作。

图表基础:

创建基本图表，设置坐标轴限制，添加图例，使用数学符号和表达式。

调整图表样式:

设置持久的默认图表参数，如边距和背景色。

保存图表:

将图表保存为多种格式，如 PDF 或 PNG。

颜色和调色板:

使用 R 的颜色系统，添加透明度，更改颜色调色板。

ggplot2 包:

利用 ggplot2 包，根据图形的语法来创建图表，包括使用数据、美学和几何对象。

ggplot2 构建块:

使用 ggplot2 包中的数据、美学、几何体、分面、统计、坐标和主题。

ggplot2 绘图示例:

使用 ggplot2 创建点图、盒图、直方图、密度图和条形图。

地图绘制:

使用 maps 包和 ggplot2 在地图上绘制数据。

其他 ggplot2 功能:

使用主题、分面和颜色来增强图表的表现力。

交互式图表:

使用 plotly 包将 ggplot2 图表转换为交互式图表。

动画图表:

使用 gganimate 包制作动态图表。

其他资源:

提供学习 ggplot2 和 R 数据可视化的其他资源链接。

总结:

介绍了在 R 中使用 plot 函数和 ggplot2 包创建和自定义高质量图表的方法。

R 与其他语言的接口: 介绍了如何将 R 与 C、Python 等语言结合使用。

版本控制与 GitHub: 讨论了使用 GitHub 进行版本控制的方法。

编写自己的 R 包: 提供了如何使用 devtools 和 roxygen2 来创建和文档化 R 包的指南。

数据导入：使用 readr 和 rio 包从不同格式导入数据。

列表和 Tibbles：介绍了列表的使用和 tibbles 中列表作为列的应用。

整洁数据：遵循整洁数据的原则（每个变量形成一列，每个观察形成一行，每个单元格一个值）。

使用 tidyr 重塑数据：使用 pivot_longer()和 pivot_wider()函数转换数据框的形状。

分类数据处理：包括使用 factors 和 levels 处理分类数据，使用 forcats 包和 table()函数处理和整理分类数据。

字符数据处理：使用 stringr 包处理字符串，包括使用正则表达式、字符串转换和文件名处理。

日期和时间数据处理：介绍了如何使用 lubridate 包处理日期和时间数据，例如创建和解析日期。

数据整理技巧：包括如何有效地使用 factors 和 stringr 函数进行数据整理和分析。

循环 (For, While, Repeat)：解释了如何使用不同类型的循环结构。

条件判断 (If-Else 语句)：展示了如何根据条件执行不同的代码块。

算术和布尔运算符：介绍了在 R 中进行基本数学和逻辑操作的方法。

函数编写：解释了如何定义函数，包括默认参数值、return 语句和匿名函数。

环境和作用域：讨论了 R 中的环境和作用域，如何影响变量的访问。

递归和排序函数：介绍了递归函数的使用，包括快速排序算法的示例。

其他编程技巧：包括如何编写自定义操作符和处理替换函数。

对象大小：使用 lobstr 包探索对象大小，包括 obj_size()和 obj_sizes()函数。

内存使用：讨论了 R 中内存分配的重要性，包括在 RStudio IDE 中的内存使用。

向量赋值和修改时复制：解释了 R 中向量赋值的内部工作原理。

列表和数据框：讨论列表和数据框的使用及其对内存的影响。

代码性能分析：介绍了使用 Rprof 进行代码性能分析的方法，以及如何使用它来优化代码。

功能编程的基础：定义了功能编程的概念，包括 R 中的函数如何作为一等公民操作。

Base R 的*apply 函数族：讨论了 lapply、sapply、vapply 等函数的用法。

purrr 包：介绍了 purrr 包，它提供了一组用于功能编程的一致且易于使用的工具。

各种 map 函数：详细说明了如何使用 purrr 包中的各种 map 函数（如 map()、map2()、pmap()）来处理列表和向量。

S3 类和泛型函数：介绍了 S3 类的创建和管理，以及如何使用和编写 S3 泛型函数。

S4 类：讨论了 S4 类的创建和特性，包括如何定义和使用 S4 对象。

继承和多态性：探讨了在 R 中使用继承和多态性的方法。

S3 和 S4 的比较：对比了 S3 和 S4 系统的特点和应用场景。

其他面向对象系统：简要介绍了 R 中其他的面向对象编程系统，如 R6。