

# R 语言基础春令营 (1)：导论

陈堰平

统计之都

<http://cos.name>

April 14, 2013

# 大纲

- 1 课程介绍
- 2 R 基本介绍和安装
- 3 R 语言基本语法

# Section 1

## 课程介绍

# 课程目标

在完成本课程学习后，学员将可以：

- 自行在各种平台上安装和使用 R 语言；
- 用 R 语言实现对各种常见数据的基本分析和绘图；
- 用 R 语言绘制符合论文发表要求的统计图形；
- 了解 R 语言的整体情况，可以根据自己的需求选择进一步学习的方向。

# 课程计划

- R 语言简介和系统安装
- R 语言基本语法
- R 语言基本统计分析与绘图
- 简介 R 语言高级统计和绘图方法

# 讲师介绍

## 陈堰平

- 2007.9~ 2010.7, 中国人民大学统计学院, 数理统计专业, 硕士
- 北京交通大学理学院, 信息与计算科学, 本科
- 统计之都理事会理事
- 中国 R 语言会议理事会理事
- COS 沙龙理事会理事
- 《R 语言编程艺术》的主要译者

## Section 2

### R 基本介绍和安装

# 什么是 R?

*R* 是一个用于统计计算和图形的自由软件环境。

*R is a free software environment for statistical computing and graphics.*



# R 与数据分析

- 基本统计分析
- 多元统计分析
- 数据挖掘：分类、回归、聚类、推荐、关联规则
- 专业领域数据分析：社交网络、生命科学、行为分析、商业智能.....
- 大数据分析：Hadoop、HANA、Bigmem
- 生物数据分析：Bioconductor

# R 语言与数据可视化

- 基本图形系统：R 软件中最基本的绘图系统，简单、易用，功能较弱，系统性不强，例如 `plot` 函数、`hist` 函数等
- Lattice、grid、`rgl` 等图形系统：由于 R 基本图形系统的不足，诞生了很多第三方的图形系统
- `ggplot2` 基于图形语法 ( grammar of graphics )，理论上可以画出所有的统计图形，而且语法简洁。但跟基本统计图形的使用有较大差异，有一定的学习难度。
- 自动文档：`knitr`

# 统计之都与 R 语言

- 主站有大量介绍 R 的文章
- 论坛上关于 R 的问题讨论的最多
- 创始人和主要管理员的贡献
- 翻译编写了大量关于 R 语言的书 <http://cos.name/books>
- R 语言会议 <http://cos.name/chinar/>

# 如何安装

R 的官方网站 <http://www.r-project.org/>

The R Project for Statistical Computing

PCA 5 vars  
(principal components)

Clustering 4 groups

Factor 1 [41%]

Factor 3 [19%]

Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of platforms, Windows and MacOS. To **download R**, please choose your preferred **CRAN mirror**.
- If you have questions about R like how to download and install the software, or what the license terms are, please **answers to frequently asked questions** before you send an email.

# 开发环境

RStudio 官方网站 <http://www.rstudio.com>

# 工作空间

- `getwd()` 显示当前的工作目录
- `setwd("D:/work")` 或 `setwd("D:\\work")` 修改当前的工作目录为 `mydirectory`  
列出当前工作空间中的对象
- `rm(objectlist)` 移除 (删除) 一个或多个对象
- `help(options)` 显示可用选项的说明
- `options()` 显示或设置当前选项
- `history(N)` 显示最近使用过的 `N` 个命令 (默认值为 25)
- `savehistory("myfile")` 保存命令历史到文件 `myfile` 中 (默认值为 `Rhistory`)
- `loadhistory("myfile")` 载入一个命令历史文件 (默认值为 `Rhistory`)

## 工作空间 (续)

- `save.image("myfile")` 保存工作空间到文件 `myfile` 中 (默认值为 `.RData`)
- `save(objectlist, file="myfile")` 保存指定对象到一个文件中
- `load("myfile")` 读取一个工作空间到当前会话中 (默认值为 `.RData`)
- `q()` 退出 R。将会询问你是否保存工作空间

# 第一个 R 会话

见代码



## 获取帮助

- `help.start()` 打开帮助文档首页
- `help("foo")`或`?foo` 查看函数 `foo` 的帮助 (引号可以省略)
- `help.search("foo")`或`??foo` 以 `foo` 为关键词搜索本地帮助文档
- `example("foo")` 函数 `foo` 的使用示例 (引号可以省略)
- `RSiteSearch("foo")` 以 `foo` 为关键词搜索在线文档和邮件列表存档
- `apropos("foo", mode="function")` 列出名称中含有 `foo` 的所有可用函数
- `data()` 列出当前已加载包中所含的所有可用示例数据集
- `vignette()` 列出当前已安装包中所有可用的 `vignette` 文档
- `vignette("foo")`为主题 `foo` 显示指定的 `vignette` 文档
- `??, ?`

# 包

## R 使用包来存储若干相关联的程序文件

---

安装包	<code>install.packages("ggplot2")</code>
加载包	<code>library(ggplot2)</code>
当前加载包的情况	<code>search()</code>
本地安装包的列表	<code>.packages(all.available = TRUE)</code>

---

## 启动时自动加载的包

包	描述
stats	常用统计函数
graphics	基础绘图函数
grDevices	基础或 grid 图形设备
utils	R 工具函数
datasets	基础数据集
methods	用于 R 对象和编程工具的方法和类的定义
base	基础函数

## 参考资料

- R 菜鸟入门
- 《153 分钟学会 R》
- R 参考卡片
- R in a Nutshell
- ggplot2: Elegant Graphics for Data
- R Graphics Cookbook
- The Art of R Programming

## Section 3

### R 语言基本语法

# 基本数据类型

## 数据类型

- 向量 vector
- 矩阵 matrix
- 数组 array
- 数据框 data frame
- 因子 factor
- 列表 list

# 向量

- 单个数值（标量）没有单独的数据类型，它只不过是向量的一种特例
- 向量的元素必须属于某种模式（mode），可以整型（integer）、数值型（numeric）、字符型（character）、逻辑型（logical）、复数型（complex）
- 循环补齐（recycle）：在一定情况下自动延长向量
- 筛选：提取向量子集
- 向量化：对向量的每一个元素应用函数
- 使用 `seq()` 创建向量
- 使用 `rep()` 重复向量常数

# 矩阵

矩阵 (matrix) 是一种特殊的向量，包含两个附加的属性：行数和列数。所以矩阵也和向量一样，有模式的概念，例如数值型或字符型。(但反过来，向量却不能看作是只有一列或一行的矩阵。)

- 创建矩阵
- 矩阵运算
- 索引
- 增加或删除行 (列)



# 数组

数组 (array) 是 R 里一个更一般的对象，矩阵是数组的一个特殊情形。数组可以是多维的。例如一个三维的数组可以包含行、列和层 (layer)，而一个矩阵只有行和列两个维度。

```
array(data = NA, dim = length(data), dimnames = NULL)
as.array(x, ...)
```

```
is.array(x)
```

# 列表

向量的元素要求都是同类型的，而列表（list）与向量不同，可以组合多个不同类型的对象

# 数据框

数据框类似矩阵，有行和列这两个维度。然而，数据框与矩阵不同的是，数据框的每一列可以是不同的模式（mode）。例如，某列可能由数字组成，另一列可能由字符串组成。

# 因子

因子的设计思想来源于统计学中的名义变量 (nominal variables)，或称之为分类变量 (categorical variables)。这些变量的值本质上不是数字，而是对应为分类，例如民主党、共和党和无党派，尽管它们可以用数字编码。

# 算术运算

- $x + y$  加法
- $x - y$  减法
- $x * y$  乘法
- $x / y$  除法
- $x ^ y$  乘幂
- $x \% \% y$  模运算
- $x \% / \% y$  整数除法

# 逻辑运算

- $x == y$  判断是否相等
- $x <= y$  判断是否小于等于
- $x >= y$  判断是否大于等于
- $x \&\& y$  标量的逻辑“与”运算
- $x || y$  标量的逻辑“或”运算
- $x \& y$  向量的逻辑“与”运算 ( $x$ 、 $y$  以及运算结果都是向量)
- $x | y$  向量的逻辑“或”运算 ( $x$ 、 $y$  以及运算结果都是向量)
- $!x$  逻辑非
- 逻辑值 TRUE 和 FALSE 可以缩写为 T 和 F (两者都必须是大写)。而在算术表达式它们会转换为 1 和 0

# 函数

```
g <- function(x) {  
  return(x+1)  
}
```

函数也是对象

# 条件语句

```
if (r == 4) {  
  x <- 1  
} else {  
  x <- 3  
  y <- 4  
}
```



# 循环语句

- for
- while
- repeat

repeat 没有逻辑判断退出条件，必须利用 break ( 或者类似 return() ) 的语句。当然，break 也可以用在 for 循环中。

# 代码格式化工具

```
library(formatR)
```