

基本运算

`a <- 4` 或 `a = 4` | 赋值，将4赋予变量a.
`1 + 1` | 加法运算，输出2.
`2 - 1` | 减法运算，输出1
`3 * 2` | 乘法运算，输出6.
`3 / 2` | 除法运算，输出1.5.
`5 %% 2` | 整除运算，输出2.
`5 %% 2` | 余除运算，输出1.
`2 ** 3` 或 `2^3` | 幂运算，2³，输出8.
`9 ** 0.5` | 开方运算，√9，输出3.

逻辑判断

`a < b` | 判断b是否小于等于7，TRUE.
`a > 3` | 判断a是否大于3，TRUE.
`a >= b` | 判断a是否大于等于b，FALSE.
`a == 1` | 判断a是否等于1， FALSE.
`a != b` | 判断a是否不等于b，TRUE.
`a %in% 1:5` | a是否在向量1:5中，TRUE.
`!a %in% 1:5` | a是否不在向量1:5中，FALSE.
`is.na(a)` | 判断a是否为缺失值，FALSE.
`is.null(a)` | 判断a是否为空值，FALSE.
`isTRUE(a)` | 判断a是否为TRUE，FALSE.
`!a` | NOT a.
`a | b` | a OR b.
`a & b` | a AND b.

文件读写

`read.table('file.txt')` | 读取txt文件.
`read.csv('file.csv')` | 读取csv文件.
`load('file.RData')` | 读取R数据文件.
`write.table(df, 'file.txt')` | 输出txt文件.
`write.csv(df, 'file.csv')` | 输出csv文件.
`fromJSON()` | 读取JSON文件（需要安装RJSONIO、jsonlite或rjson包）.
`toJSON()` | 将R对象输出为JSON文件（需要安装RJSONIO、jsonlite或rjson包）.
`xmlParse('file.xml')` | 读取XML文件（需要安装XML包）.
`read.xlsx("file.xlsx", 1)` | 读取Excel文件（需要安装xlsx包）.
`save(df, file='file.Rdata')` | 输出R数据文件.

包的使用

`install.packages('ggplot2')` | 安装ggplot2包.
`library(ggplot2)` | 加载ggplot2包.
`boot::cv.glm` | 使用boot包中的cv.glm()函数.
`data(iris)` | 加载R内置数据集iris.

工作环境

`getwd()` | 查看当前工作环境.
`ls()` | 列出当前环境下储存的所有变量.
`rm(x)` | 移除当前环境下的变量x.
`rm(list = ls())` | 移除当前环境下所有变量.
`setwd('/Users/Penglu/Desktop/')` | MacOS中变更工作环境.
`setwd('C:/Documents/Data/')` | Windows中变更工作环境.

辅助工具

`?rank` | 查看rank()函数的帮助文档.
`??cv.glm` | 查看函数名包含cv.glm的函数，返回boot::cv.glm.
`help.search('mean')` | 搜索文档中包含该字符'mean'的函数.
`help(package = 'ggplot2')` | 查看ggplot2包的帮助文档.

控制与函数

`for(x in 1:4){` | 执行相关操作
`}`

`while(条件){` | 执行相关操作
`}`

`if(条件){` | 执行相关操作
`}else{` | 执行其它操作
`}`

`my_func <- function(变量){` | 执行相关操作
`return(新变量)`
`}`

数学运算

`pi` | 圆周率π，3.141593
`abs(-3.2)` | 绝对值运算，3.2.
`sqrt(9)` | 平方根运算，√9.
`exp(5)` | 幂运算e⁵，148.4132.
`log(9, 3)` | 对数运算，log₃(9).
`log(9)` | 自然对数运算，logₑ(9).
`log10(100)` | 常用对数运算，log₁₀(100).
`factorial(5)` | 计算阶乘，120.
`min(1:4)` | 求最小值，1.
`mean(1:4)` | 求均值，2.5.
`median(1:4)` | 求中位数，2.5.
`max(1:4)` | 求最大值，4.
`sum(1:4)` | 求和，10.
`quantile(0:100)` | 计算分位数.
`round(3.141, 2)` | 保留两位小数，3.14.
`signif(3.19, 2)` | 保留2位有效数字，3.2.
`var(1:4)` | 计算方差，1.666667.
`cor(1:4, 4:1)` | 计算协方差，-1.
`sd(1:4)` | 计算标准差，1.290994.
`rank(c(5,1,4,7))` | 排名返回名次，3 1 2 4.
`sort(c(5,1,4,7))` | 排序返回数值，1 4 5 7.
`order(c(5,1,4,7))` | 排名返回索引，2 3 1 4.
`ceiling(-1.8)` | 大于该数最小的整数，-1.
`floor(-1.8)` | 小于该数最大的整数，-2.
`trunc(c(3.2, -1.8))` | 保留整数部分，3 -1.
`sin(x), cos(x), tan(x)` | 输入带有π的弧度，cos(pi/2).
`sinpi(x), cospi(x), tanpi(x)` | 输入不带有π的弧度，cospi(0.5).

变量特征

`str(x)` | 查看变量x的结构
`class(x)` | 查看变量x的类型.
`is.logical(4)` | 数字4是否为逻辑型，FALSE.
`is.numeric('4')` | 字符'4'是否为数字型，FALSE.
`is.character('4')` | 字符'4'是否为字符型，TRUE.
`is.factor(4)` | 数字4是否为因子型，FALSE.
`as.logical(4)` | 将数字4转换为逻辑值，TRUE.
`as.numeric('4')` | 将字符'4'转换为数字型，4.
`as.character(4)` | 将数字4转换为字符型，'4'.
`as.factor(4)` | 将数字4转换为因子，4 Levels: 4.

快速绘图

`plot(x)` | 绘制x与x索引的散点图.
`plot(x, y)` | 绘制x与y的散点图.
`hist(x)` | 绘制x的直方图.
`barplot(x)` | 绘制x的条形图.
`boxplot(x)` | 绘制x的箱线图.
`pie(x)` | 绘制x的饼图.
`pairs(x)` | 绘制数据框x的散点图矩阵.
`title(main=, sub=, xlab=, ylab=)` | 为图形添加标题、副标题、x轴标签和y轴标签.

apply函数

`apply(df, axis, func)` | 将函数运用到矩阵型数据的所有行或列上.
`lapply(list(a=1:4,b=5:9), sqrt)` | 将函数运用到列表中的每个元素并返回列表.
`sapply(list(a=1:4,b=5:9), sqrt)` | 将函数运用到列表或矩阵型数据中并返回矩阵.
`tapply(dfmpg, dfcyl, mean)` | 将数据按因子等级分组，并对所有分组使用该函数.

建模与统计检验

`lm(y~x, data=df)` | 计算y关于x的线性模型.
`glm(y~x, data=df)` | 计算y关于x的广义线性模型.
`summary()` | 查看模型内容.
`t.test(x, y)` | t检验.
`prop.test()` | 比例检验.
`pairwise.t.test()` | 结对t检验.
`aov()` | ANOVA方差分析.

向量

`a <- c(2, 4, 6)` | 创建向量2 4 6.
`a <- 2:6` | 创建从2到6的向量.
`seq(2, 8, by=02)` | 从2到8步幅为2的向量.
`rep(1:2, times=3)` | 从1到2整体重复3次的向量.
`rep(1:2, each=3)` | 从1到2每个重复3次的向量.
`rev(c(1,3,2))` | 反序向量，2 3 1.
`table(c(2,2,3))` | 对元素进行个数统计.
`unique(c(2,2,3))` | 对向量元素去重，2 3.
`length(c(1,3,2))` | 求向量长度，3.



`x[4]` | 向量x中索引为4的元素.
`names(x)` | 查看或修改向量x中的元素标签.
`x['t']` | 向量x中标签为't'的第一个元素.
`x[-4]` | 向量x中除第4个外所有其它元素.
`x[2:4]` | 向量x中第2到4个元素.
`x[-(2:4)]` | 除第2到4个外所有其它的元素.
`x[c(1, 5)]` | 向量x中第1、5个元素.
`x[x == 2]` | 向量x中等于5的元素.
`x[x > 5]` | 向量x中所有大于5的元素.
`x[x %in% c(1, 5)]` | 向量x与向量1 5的交集.

字符串

`toupper(x)` | 将字符串x转换成大写.
`tolower(x)` | 将字符串x转换成小写.
`nchar(x)` | 统计字符串x中的字母个数.
`paste(x, collapse='-')` | 将x中的元素用-链接.
`paste(x, y, sep='-')` | 将向量x与y中的元素分别用-链接.
`grep('a', x)` | 判断x中是否包含字符'a'.
`sub('l', 'L', x)` | 将字符串x中的第一个'1'替换成'L',并返回新字符串.
`gsub('l', 'L', x)` | 将字符串x中的所有'1'替换成'L',并返回新字符串.
`substr(x, start=2, stop=5)` | 提取字符串x中从第2个字母到第5个字母的部分.

列表

`L <- list(x=1:3, y=c('a', 'b'))` | 创建包含元素x和y的列表L.
`L[1]` | 提取列表L中的第一个元素,即x 1 2 3.
`L['x']` | 提取列表L中名为'x'的元素,即x 1 2 3.
`L[[1]]` | 提取列表L中第一个元素的值,即1 2 3.
`L$x` | 提取列表L中名为'x'元素的值,即1 2 3.

概率分布

分布	随机取样	单点概率	区间概率	分位数
正态	<code>rnorm()</code>	<code>dnorm()</code>	<code>pnorm()</code>	<code>qnorm()</code>
泊松	<code>rpois()</code>	<code>dpois()</code>	<code>ppois()</code>	<code>qpois()</code>
二项	<code>rbinom()</code>	<code>dbinom()</code>	<code>pbinom()</code>	<code>qbinom()</code>
均匀	<code>runif()</code>	<code>dunif()</code>	<code>punif()</code>	<code>qunif()</code>

因子

`a <- factor(1:3)` | 将向量1:3转换成无序因子a.
`b <- ordered(1:3)` | 将向量1:3转换成有序因子b.
`levels(b)` | 查看和修改因子a的因子水平.
`relevel(a, '3')`|更改无序因子a的第一个因子水平.

矩阵

创建矩阵
`m <- matrix(1:4, nrow=2, ncol=2, byrow=T)`
`n <- matrix(5:8, nrow=2)`

m=

1	2
3	4

n=

5	7
6	8

数据提取

`m[1,]` | 提取矩阵m中第1行的数,即1 2.
`n[, 2]` | 提取矩阵n中第2列的数,即7 8.
`m[2, 2]` | 提取矩阵m中第2行第2列的数,即4.

矩阵运算

`t(m)` | 矩阵转置
`m * n` | 矩阵m逐元素乘矩阵n.
`m %% n` | 矩阵m乘矩阵n.
`solve(m)` | 矩阵求逆.
`solve(m, n)` | 求解mx=n
`diag(m)` | 返回矩阵m的主对角线值.
`eigen(m)` | 求矩阵m的特征值和特征向量.

`cbind(m,n)` | 按行合并矩阵m和矩阵n.
`rbind(m,n)` | 按列合并矩阵m和矩阵n.

1	2	5	7
3	4	6	8

`cbind(m, n)`

1	2
3	4
5	7
6	8

`rbind(m, n)`

`rowMeans(m)` | 返回矩阵m各行的均数,等于1.5 3.5.
`rowSums(m)` | 返回矩阵m各行的合,等于3 7.
`colMeans(m)` | 返回矩阵m各列的均数,等于2 3.
`colSums(m)` | 返回矩阵m各列的合,等于4 6.

数据框

`df <- data.frame(x=c('a','b'), y=1:2)` | 创建数据框df.

df=

	x	y
1	a	1
2	b	2

`View(df)` | 查看完整数据框.
`head(df)` | 查看数据框前6行.
`tail(df)` | 查看数据框后6行.
`nrow(df)` | 查看数据框行数.
`ncol(df)` | 查看数据框列数.
`dim(df)` | 查看数据框行数和列数即2 2.
`summary(df)` | 查看数据框每一列的统计分析.
`names(df)` | 返回数据框列名,即'x' 'y'.
`rownames(df)` | 返回数据索引,即'1' '2'.

数据提取

`df$y` | 提取y列的数据-A.
`df[[2]]` | 提取第2列的数据-A.
`df[, 2]` | 提取第2列的数据-A.

	x	y
1	a	1
2	b	2

`A`

	x	y
1	a	1
2	b	2

`B`

`df[2,]` | 提取数据框df的第二行-B.
`df[df$x=='b',]` | 提取x列等于b的所有行-B.
`subset(df, x=='b')` | 提取x列等于b的所有行-B.
`df[df$y %in% 2:4,]` | 提取y列值在向量2:4中的所有行-B.
`df[df$x=='b' & df$y %in% c(2,3,4),]` | 提取x列值为b且y列值在向量2:4中的所有行-B.
`df[df$x=='b' & !df$y %in% c(0,1),]` | 提取x列值为b且y列值不在向量0:1中的所有行-B.
`subset(df, x=='b' & y==2)` | 提取x列为b且y列为2的所有行-B.
`df[2, 2]` | 提取第二行、第二列的值-C.
`df[2,]$y` | 提取第二行、y列的值-C.
`df[df$x == 'b',]$y` | 提取x列为b, y列的值-C.
`df[2]` | 提取第二列-D.
`which(df$x == 'b')` | 返回数据框df中x列的值为b的所有行索引-E.

`df[order(df$y, decreasing = TRUE),]` | 按y列降序排列数据框df-F.

	x	y
1	a	1
2	b	2

`C`

	x	y
1	a	1
2	b	2

`D`

	x	y
1	a	1
2	b	2

`E`

	x	y
2	b	2
1	a	1

`F`

数据框合并

创建新数据框 `df2 <- data.frame(x=c('a','c'), y=c(1,3))`

df2=

	x	y
1	a	1
2	c	3

`cbind(df, df2)` | 按行合并数据框df和数据框df2.
`rbind(df, df2)` | 按列合并数据框df和数据框df2.

`merge(x=df, y=df2, by='x', all.x=TRUE)`
左连接-G.
`merge(x=df, y=df2, by='x', all.y=TRUE)`
右连接-H.
`merge(x=df, y=df2)`
内连接-I.
`merge(x=df, y=df2, by='x', all=TRUE)`
外连接-J.
`merge(x=df, y=df2, by=NULL)`
全连接-K.

	x	yx	yy
1	a	1	1
2	b	2	NA

`G`

	x	yx	yx
1	a	1	1
2	c	NA	3

`H`

	x	y
1	a	1

`I`

	x	yx	yy
1	a	1	1
2	b	2	NA
3	c	NA	3

`J`

	xx	xy	xy	yy
1	a	1	a	1
2	b	2	a	1
3	a	1	c	3
4	b	2	c	3

`K`

