|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| typeof()返回类型  cat(…, file, sep, append)  sink() 把命令行窗口显示的运行结果转向保存到指定的文本文件中，如 果希望保存到文件的同时也在命令行窗口显示， 使用 split = TRUE 选项。结束时再调用一次sink()  source(file, local, eval, print.eval, verbose, prompt.echo, max.deparse.length, encoding)  ls()返回当前空间中的对象  rm()删除对象  install.packages()  加载: library(), require()返回bool  is.foo()判断 as.foo()转换  ^幂 %%取余 %/%整除 exp() log10()  round(a, b)四舍五入取b位小数，默认取整  floor(), ceiling()向上下取整  判断两个浮点型对象是否完全相同，不能直接采用==和identical()，而应该用all.equal()  and & or | not !  nchar(str, type) 字符串的长度，默认字符个数，可以按字节(type=”bytes”)  toupper()转为大写, tolower()转为小写  substr(x, start, stop), substring(x, start, stop)  从字符串x中取出从第start个到第stop个的子串，substring可省略stop，取到末尾  gsub(pattern, replacement, x, ignore.case)  paste(…, sep=” ”, collapse) paste0()无缝连接  re:  [0-9]表示数字，等价于\d  [a-zA-Z0-9]表示数字和所有的英文字母，等价于\w和[:alnum:]  [:cntrl:]表示ASCII控制字符  [:punct:]表示既不属于[:alnum:]也不属于[:cntrl:]的  [:space:]表示任意空格  + 表示一个或多个，? 表示零个或一个，^x 表示除x外的任意字符  Date类型一般用整数保存，数值为从1970-1-1经过的天数  POSIXct把日期时间保存为从1970年1月1日零时到该日期时间的时间间隔秒数  POSIXlt把日期时间保存为一个包含年、月、日、星期、时、分、秒等成分的列表  Sys.date(), Sys.time()  as.Date(x, format, origin)  format: %d 日 %m 月（数字格式） %b 月（英文简称） %B 月（英文全称） %y年（两位） %Y 年（四位）  origin: x为数字时设定起始日期  提取：weekdays, months, days, quarters  difftime(time1, time2, units=”auto”)  factor(x = character(), levels, labels = levels, exclude = NA, ordered = is.ordered(x), nmax = NA)  向量  向量构建：c(), 1:10,  seq(from, to, by, length.out, along.with)  by为步长（两元素之差）可负（即降序）  a[1:4] # 取出第 1 到 4 项，包含第 1 和第 4 项 a[c(1, 3, 5)] # 取出第 1, 3, 5 项 a[c(-1, -5)] # 去掉第 1 和第 5 项 | 向量与标量的运算为每个元素与标量的运算  等长向量的运算为对应元素两两运算  两个不等长向量的四则运算，规则是每次从头重复利用短的一个  unique()找出唯一的元素  setequal()判断集合是否相等  intersect()交集 union()并集 setdiff()差集  向量比较：  x %in% table: x的每个元素是否都在table  返回逻辑向量  向量函数：  数学函数：sqrt(), sign(), abs(),log(), exp()等  数据统计函数：mean(), median(), sd(), var(), max(), min(), length(), sum(), cumsum(), prod()  summary(), range()  排序函数：  sort(x, decreasing=FALSE), rev()  order(…, decreasing=FALSE, na.last)  逻辑运算函数：  identical()精细比较  all.equal(target, current, tolerance)  all(), any(), which(),  match(x, table, nomatch=NA, incomparables),  which.max(),  which.min()  字符型向量函数： paste(), strsplit()等  向量长度和属性：length(), attributes(), names()  命名：names(x) <- c(…)  用字符串作为下标时，如果该字符串不在向量的元素名中，读取时返回缺失值结果，赋值时该向量会增加一个元素并以该字符串为元素名  矩阵  matrix(data, nrow, ncol, byrow=FALSE, dimnames)  rbind(), cbind()  diag() 数值/向量：创建矩阵；矩阵：返回对角元素组成的向量；  dim() 返回矩阵的行数和列数  nrow(), ncol()  dimnames() 返回矩阵的行名和列名  矩阵与标量的运算为每个元素与标量的运算  矩阵+-\*/均为对应元素计算，%\*%矩阵乘法  向量左乘矩阵时，看成行向量； 向量右乘矩阵时，看成列向量  t()转置 det() solve()求逆 solve(A, b)返回解  crossprod(A, B) tcrossprod(A, B)  参数只有一个矩阵同理  apply(A, i, FUN) 把矩阵 A 的每一列分别输入到函数 FUN 中，得到对应于每一维度的结果，其中 i = 1 表示对行进行运算， i = 2 表示对列进行运算。  summary()函数按列输出汇总信息  rowMeans(),colMeans(),rowSums(), colSums()  A[1, ]取第一列 A[1:2, c(1, 3)]指定行和列  drop=FALSE可以保留原有维度  which(mat == min(mat, na.rm=TRUE), arr.ind = TRUE) # 返回行号和列号  数组(array)  矩阵运算一般都可以运用到数组  数组名 <- array(数组元素, dim=c(第一下标个数, 第二下标个数, ..., 第s下标个数)) | 列表(list)  保存不同类型数据, list()  可通过names()命名，也可在定义时命名  []访问会返回列表，$和[[]]访问返回元素  赋值不存在的名称可添加元素，NULL删除元素  若想赋值为NULL可以[]访问，赋值为list(NULL)  unlist()转换为基本向量  数据框(dataframe)  data.frame(name=c(), age=c(), married=c())  names()列名 rownames()行名  rbind(d, list()) cbind(d, c())添加行/列  访问类似矩阵，一行返回仍为df  with(data, expr)  获取数据  load(“name”)读取二进制文档.RData/.Rda  read.table(file, header=TRUE, sep, stringAsFactors=TRUE, fileEncoding=“UTF-8”)  head()查看前几行（默认6）  read.csv()中na.strings指定缺失值，row.names=x指定列名为x的列为行名，skip=x跳过x行，nrows=x只读取x行（header=TRUE时不包括列名）  readLines(file, n)按行读取，每行一个字符串, n行数  read\_excel(path, sheet, range, col\_names=TRUE)  range例如B3:D87，sheet工作簿名称或序号  导出数据  导出二进制文档save(thing, file=“name”)  save.image(“name”)保存工作空间中所有变量  用 write.csv(), write.table(), write.delim(), write.fwf() , writeLines() 等输出 data.frame至外面文件  openxlsx::write.xlsx(d, file="mydata.xlsx", asTable=TRUE) 保存到excel  预处理数据  head(x, n) tail(x, n)选择数据框正/倒数n行  subset(d, d$年龄<18 & d[,"性别"]=="男", select = c("性别","年龄","疗程"))  order(…, decreasing=FALSE, na.last) 返回位置  如果行号都一样，那么使用data.frame(dat1,dat2)或者cbind(dat1,dat2)直接合并即可  merge() by, by.x, by.y, all, all.x, all.y  scale(x, center, scale)减去该列平均值，除以标准差  table()查看因子类型变量在每一类的频数分布  table( d[,"分型"], d[,"性别"], useNA= “ifany”)  常用的汇总函数有：  总体信息： summary()，table()  位置度量： mean() , median() 。  分散程度（变异性）度量： sd() , IQR() , mad() 。  分位数： min() , max() , quantile(x, 0.9)  可添加na.rm=TRUE  aggregate() 函数对输入的数据框用指定的分组变量（或交叉分组） 分组进行概括统计  aggregate(d[,c(3:5,7)], by = d[c("分型","性别")], mean)  tapply(X, INDEX, FUN, useNA)分组概括  useNA= “always”/ “ifany”可把NA计算在内 | dplyr  filter()：按行筛选数据  filter(d, d$年龄<18, 性别=="男")  select()：按名称选取变量/列  filter(select(d, 性别,年龄, 疗程), d$年龄<18)  arrange()：对行排序数据，缺失值总是排在最后  arrange(d, 性别, desc(年龄))  mutate()：创建新变量（列）返回新数据框  mutate(d, 住院时间 = 出院时间 - 入院时间 + 1)  summarise()：汇总数据  summarise(group\_by(d, 分型,性别), mean(年龄), mean(入院时间), mean(出院时间))  group\_by()：分组汇总数据  xxx\_joins()：合并数据  %>%：管道 x %>% f(y)即为f(x, y)  随机数  set.seed(seed, kind = NULL, normal.kind = NULL, sample.kind = NULL)  sample(x, size, replace = FALSE, prob = NULL)  sample(1:3, size = 100, replace = TRUE, prob = c(.2, .3, .5))  runif(n) 产生 n 个标准均匀分布随机数  rnorm(n) 产生 n 个标准正态分布随机数  分布名称前面可以添加字母  p:cdf; q:quantile; d:pdf; r:random  binom, chisq, exp, f, gamma, geom, lnorm, nbiom, norm, pois, t, unif  画图  在画图前用 postscript(), pdf(), tiff(), jpeg() ，png()等函数制定输出的文件名，文件大小等。  在画图后用 dev.off() 函数关闭当前画图设备并生成输出文件  条形图batplot() main=标题, col=颜色horiz=FALSE(横排) beside=FALSE(并排)  饼图pie() 箱线图boxplot()  直方图hist(x) freq=FALSE 改变y轴的计数为比例  main=标题 xlab=, ylab= x/y轴标签 xlim=, ylim=范围  density(x)核密度估计  lines(density(d[, "年龄"]), lwd = 3, col = "black")  散点图plot(x, y) pch点的形状, col颜色, cex大小  plot(疗程 ~ 年龄, main = "疗程与年龄的关系", data = d, col=ifelse(性别=='男', 'blue', 'red'), pch=16,  cex = 1 + (入院时间- min(入院时间))/(max(入院时间)- min(入院时间))) 用大小表示第三维  pairs 函数对矩阵或数据框的各列两两做散点图，构成一个散点图矩阵  plot()中使用type=‘l’画折线图  lwd指定线宽， lty指定虚线  qqnorm(data) qqline(data, col= “red”)等plot()参数  qqplot(data1, data2) abline(0, 1) y=a+bx  curve(expr, from, to) 可以对以 x 为自变量的表达式做函数曲线，或者对某个函数作函数曲线  三维图  persp 函数作三维曲面图, contour 作等值线图，image 作色块图  image(x, y, normal.mat, col=terrain.colors(20))  contour(x, y, normal.mat, add = TRUE)  低级图形函数  abline()添加直线 a截距b斜率v垂直线h水平线  lines()折线points()散点 legend()图例 | axis()坐标轴 第一个参数取1，2，3，4， 分别表示横轴、纵轴、上方和右方; 参数at为刻度线位置; labels 为标签; las: 设置坐标轴标签的方向（0 = 并行于轴，1 = 总是水平，2 = 垂直于轴，3 = 总是垂直）  text(x, y, “text”)坐标区域内文本, mtext()区域外  expression()显示数学公式到坐标轴/文本  par() 可以用于定制任意的图形参数，包括字体、颜色、坐标轴、标题等。均可用到前面的绘  图函数中。  col, col.axis, col.lab, ...:指定图形颜色、坐标轴颜色、标签颜色  lty: 线的类型，1是实线，2是虚线，3以后是点虚线的各种组合  lwd: l线的宽度，默认为1  pch: 点的类型  cex, cex.axis, cex.lab, cex.main: 符号大小倍数，基本值为1。  font: 字体，一般 font=1 是正体, 2 是粗体, 3 是斜体, 4 是粗斜体。  par()通过设置mfrow（按行）或mfcol（按列）将页面分成几个区域，每个区域对应一个图形。  layout(mat, widths, heights, respect)  mat: 一个矩阵。矩阵中的数字表示图形的位置，数字相同的单元格属于同一个图形区域。  widths: 一个向量，指定每列的相对宽度。  heights: 一个向量，指定每行的相对高度。  respect: 一个逻辑值，如果设置为TRUE，则layout() 会尝试保持宽高比。  ggplot2  aes() 函数将数据映射到图形属性。如：  x: 映射到x轴的变量  y: 映射到y轴的变量  color: 图形属性的轮廓颜色  fill: 图形属性的内部填充颜色  alpha: 图形的透明程度（0代表全透明）  linetype: 线的类型，如实线solid, 虚线dashed, 点线dotted  shape: 点的形状  size: 图形属性的大小  geom\_bar(): 展示离散变量的分布  geom\_boxplot(): 箱线图  geom\_histogram(): 直方图  binwidth窗宽  geom\_line(): 折线图  geom\_point(): 散点图  geom\_ribbon(): 条带，带有垂直厚度的路径  geom\_smooth(): 最佳拟合平滑曲线  geom\_density(): 密度估计图  geom\_vline(): vertical line  geom\_vline(aes(xintercept=mean(hwy))  geom\_violin(): 小提琴图  guides(color=FALSE)不显示颜色图例  stat\_ecdf()：计算经验累计分布函数图  stat\_summary()：在不同的 x值上概述y  stat\_qq()：Q-Q图的计算  stat\_unique()：去掉重复的行  stat\_bin() 等价于 geom\_histogram()  改变集合对象为折线：geom=“line“ |
| stat\_summary()默认mean\_se均值+-标准误  median: 中位数  mean\_cl\_boot: 均值和bootstrap得到的置信区间（默认为95%的置信区间）  mean\_cl\_normal: 均值和基于正态分布的置信区间（默认为95%的置信区 间）  mean\_dsl: 均值 ± 若干倍的标准差（默认为两倍）  median\_hilow: 中位数和分位数（默认为0.025和 0.975分位数）  stat\_qq() stat\_qq\_line()  geom\_point(stat= “summary”)可修改默认统计变换  统计变换内部采用一种数据框的形式作为输入并返回一个数据框，因此可通过统计变换增加新的变量到原始数据集中  如stat\_bin()直方图产生count(每个组里观测值数目), density(每个组里观测值密度)  stat\_bin(aes(y = ..density..))密度  标度 library(scales)  组成：scale\_图形属性名\_标度名  图形属性名:colour, shape, x, y, fill  标度名: 连续时间(continuous)，颜色(brewer, grey)，人工(discrete, manual)  scale\_x\_continuous()和scale\_y\_continuous()默认是线性地将数据映射到x轴和y轴上，我们可以通过trans =对其进行变换  scale\_y\_continuous(trans = “log10”)/scale\_y\_log10()  lims, xlim, ylim: 设定x轴和y轴的画图范围  lims(x=c(1,7), y=c(10,50))  xlab, ylab, labs: 设定x轴和y轴的标签  ggtitle: 设定图的标题  坐标系  线性坐标系：  coord\_cartesian()：默认的笛卡尔坐标系  coord\_fixed()：宽高比固定的直角坐标系  coord\_flip()：x轴和y轴反转了的笛卡尔坐标系  非线性坐标系：  coord\_map()/coord\_quickmap()：地图投影  coord\_polar：极坐标系  coord\_trans()：对数据进行统计变换之后，对x和y位置进行任意变换  coord\_trans(y="log10"), coord\_polar(theta="y")  分面  通过切割数据生成一系列小连号图，有三种类型  facet\_null(): 单个图像，默认情况。  facet\_wrap(): 把1维面板条状封装在2维中  nrow, ncol控制有多少行/列（设其中一个即可）  as.table控制分面的布局， TRUE最高值显示在右下角，反之则显示在右上角  dir控制封装的方向， h 表示横向， v 表示纵向  facet\_wrap(~ class, nrow = 3)  facet\_grid(): 生成一个2维的面板网格，其中行和列由变量组成  . ~ a 把a的值按列展开  b ~ . 把b的值按行展开  b ~ a 把a的值按列展开， 把b的值按行展开。可以是多个的，如 a + b ~ c + d | 通过调整参数 scales 来控制面板的位置标度是相同的还是允许变化的。  scales = "fixed": x和y的标度在所有面板中都固定  scales = "free\_x": x的标度可变，y的标度固定  scales = "free\_y": y的标度可变，x的标度固定  scales = "free": x和y的标度在每个面板都可以变化  多个子图  grid.arrange(p1, p2, p3, ncol=2, widths=c(1,2))  主题  ggplot2自带了八种内置主题，分别为theme\_grey(), theme\_bw(), theme\_linedraw(), theme\_light(), theme\_dark(), theme\_minimal(), theme\_classic(), theme\_void()  修改单个主题组件，则需要使用形如 plot + theme(element.name = element\_function())  主题元素（element.name）制定了能控制的非数据元素，如 plot.title控制了图像标题的外观， axis.ticks.x控制了x轴上的标签。  元素函数（element\_function）描述的是元素的视觉属性， 如element\_text()设定了字体的大小、颜色。内置的元素函数有四种基本类型：  文字（element\_text）: 绘制标签和标题，如控制字体的family(字体族)、face(字型)、colour(颜色)、size(大小)、hjust(横向对齐，0.5居中)等  线条（element\_line）: 绘制线条，参数有colour(颜色)、size(大小)和linetype(线条类型)  矩形（element\_rect）: 绘制（背景的）矩形，参数有fill(填充)的颜色、边缘的colour(颜色)、size(大小)和linetype(线条类型)  空白（element\_blank）: 不绘制任何东西。  保存图片：  ggsave("myplot.png", plot=p, width=7, height=5, units = "in")  控制流  if条件包含NA则不会执行任何代码块  条件中有向量需要用any()或all()  如果对向量每个元素遍历并保存结果，应在循环之前先将结果变量产生等长的存储，在循环内为已经分配好存储空间的输出向量的元素赋值。  为了产生长度为 n 的数值型向量，用 numeric(n) ；为了产生长度为 n 的列表，用 vector("list", n)  对向量元素遍历采用下标访问，避免丢失属性，且使用seq\_along(x)而不是1:length(x)避免长度为0  for(i in seq\_along(x)) instead of for(xi in x)  replicate(重复次数, 要重复的表达式)  函数  函数名 <- function(形参表){函数体}  stopifnot() 检查括号内的表达式是否满足，如果有不满足的则停止执行往后的命令并返回错误信息  并行  sfCpus()得到4  sfInit( parallel=TRUE, cpus=4)  sfExport()把计算所依赖的对象预先传送到每个节点  sfLibrary把计算所依赖的包导入到每个节点  sfSource把计算所依赖的源代码文件导入每个节点  sfExportAll导入所有的全局变量 | sfSapply()是sapply的并行版本，对 k 并行地循环  sfSapply(2:(nk+1), function(k) f10(k, n))  类似的有sfLapply, sfAapply  并行结束防止内存泄漏需解散集群：sfStop()  RShiny library(shiny)  fluidPage( # 创建网页  titlePanel(), # 标题栏  sidebarLayout( # 常用的边栏布局，将输入布局到左侧，输出布局到右侧  # 设置position="right"，则布局相反。  sidebarPanel(), # 边栏  mainPanel() # 主体部分 接下来的右括号省略  网页函数：  fluidPage()：流动布局的页面。  fixedPage()：具有固定宽度的页面，防止网页过宽  fillPage()：创建一个页面，可把一些空隙去掉，让网页内容充满整个页面  输入控制函数都有第一形参inputId，如果inputId为"name"， 后台会通过input$name来访问  常见的输入控制函数有：  自由文本：textInput(), passwordInput(), textAreaInput().  数值输入： numericInput()， sliderInput().  日期输入： dateInput(), dateRangeInput() .  单选：selectInput()， radioButtons().  多选：checkboxGroupInput().  上传文件： fileInput()  numericInput("num", "Number one", value = 0, min = 0, max = 100),  sliderInput("num2", "Number two", value = 50, min = 0, max = 100),  sliderInput("rng", "Range", value = c(10, 20), min = 0, max = 100)  selectInput("state", "来自于哪个学校", province)  radioButtons("animal", "最喜欢的动物？", animals)  输出控制函数都有第一形参outputId，如果outputId为"name"， 后台通过output$name访问  plotOutput()图形  tableOutput() 静态表格  textOutput() 带有格式的文本  verbatimTextOutput() 代码和控制台输出  dataTableOutput() 动态表格  imageOutput() 图片  后端  server()函数将前端的input输入到第一步的代码中，并将代码运行后的结果作为输出内容output和前端的输出匹配。  这里输出内容是图形，使用renderPlot()进行输出，定义输出的名称为distPlot，和第二步中的plotOutput("distPlot")相对应。  调用前端输入参数input$bins实现指定窗格个数  每个前端的输出函数都在后端匹配render\*\*\*()函数，将R中某种 形式的输出传递到网页上  具体的匹配如下：  文本： textOutput()和 renderText()  代码和控制台输出：verbatimTextOutput()和 renderPrint()  静态表格：tableOutput()和 renderTable()  动态表格： dataTableOutput()和 renderDataTable()  绘图： plotOutput()和 renderPlot() | R包  五个生命周期  1. 源码包（source）是处在开发版本的包，只是包含R/目录，DESCRIPTION 等组件的一个目录。  2. 压缩包 （Bundled）是把源码包打包压缩成的，扩展名为.tar.gz，调用 devtools::build()来创建。  3. 二进制包（Binary）是可以不通过编译就可以安装的文件，依赖于平台，如 Mac下的扩展名为.tgz，Windows下的扩展名为.zip。调用 devtools::build(binary = TRUE)来创建。  4. 已安装的包（Installed）是安装并解压到一个包库的二进制包  5. 内存中的包（In-memory）  可通过.libPaths() 查看当前正在使用的库  Rcpp install.packages(“Rcpp”)  sourceCpp("sumCpp.cpp")  sumC(1, 2)  #include <Rcpp.h>  函数前加上//[[Rcpp::export]]这种特殊的注释。  用wrap()把C++变量返回到R中。当C++中赋值运算的右侧表达式是一个R对象或R对象的部分内容时， 可以隐含地调用as()将其转换成左侧的C++类型  用as()函数把R变量转换为C++类型。当C++中赋值运算的左侧表达式是一个R 对象或其部分内容时， 可以隐含地调用wrap()将右侧的C++类型转换成R类型    NumericVector类在C++中保存双精度一维数组， 可以与R的实数型向量 (class为numeric)相互转换  LogicalVector 类可存储C++值true, false, 还可以保存缺失值NA\_REAL, R\_NaN, R\_PosInf, 但是这些不同的缺失值转换到R中都变成NA  NumericMatrix x(n,m)产生一个未初始化的矩阵， 元素为double类型  x.nrow()行数， x.ncol()列数， x.size()元素个数  x(i,j)i行j列 ; x.row(i)或x(i,\_)访问第i行  x.column(j)或x(\_,j)访问x的第j列(0开始)  NumericMatrix::zeros(n)n阶元素为0的方阵NumericMatrix::eye(n)返回n阶单位阵  python  set.add(), set.remove()  可变参数的形式有 \*parameter 和 \*\* parameter 两种。  \*parameter可以接收任意多参数，并将这些参数放在一个元组中。参数传递时，按照位置传递。  \*\* parameter 接收任意多个参数并将参数放在一个字典中。在参数传递时，按照参数名传递  map(function, iterable, …)向量化运算，返回迭代器，可使用list()转换为列表  lambda param1, param2,...: expression  filter(func, list)输出使判断函数为True的子列表 | # Define UI for application that draws a histogram  ui <- fluidPage(  # Application title  titlePanel("Old Faithful Geyser Data"),  # Sidebar with a slider input for number of bins  sidebarLayout(  sidebarPanel(  sliderInput("bins",  "Number of bins:",  min = 1,  max = 50,  value = 30)  ),  # Show a plot of the generated distribution  mainPanel(  plotOutput("distPlot")  )  )  )  # Define server logic required to draw a histogram  server <- function(input, output) {  output$distPlot <- renderPlot({  # generate bins based on input$bins from ui.R  x <- faithful[, 2]  bins <- seq(min(x), max(x), length.out = input$bins + 1)  # draw the histogram with the specified number of bins  hist(x, breaks = bins, col = 'darkgray', border = 'white',  xlab = 'Waiting time to next eruption (in mins)',  main = 'Histogram of waiting times')  })  }  # Run the application  shinyApp(ui = ui, server = server)  lapply(X, FUN, …)用于对列表或向量的每个元素应用一个函数，并返回一个列表  sapply()多一个参数simplify=TRUE返回一个向量或矩阵而不是列表，尝试简化输出结果  vapply(X, FUN, FUN.VALUE, ..., USE.NAMES = TRUE)  FUN.VALUE: 预期的返回值类型的模板。 |