

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[(-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) $A^T B$ tcrossprod(A, B) AB^T
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[, "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 轴标题 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(x 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 sfApply, sfAapply
并行结束防止内存泄漏解散集群: 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(),
textInputAreaInput().
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. 源代码包 (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

Rcpp	R	说明
IntegerVector	c(...) or vector(mode = "integer", ...)	整数向量
NumericVector	c(...) or vector(mode = "double", ...)	数值向量
LogicalVector	c(...) or vector(mode = "bool", ...)	逻辑型向量
CharacterVector	c(...) or vector(mode = "character", ...)	字符串向量
IntegerMatrix	matrix()	整数矩阵
NumericMatrix	matrix()	数值矩阵
CharacterMatrix	matrix()	字符串矩阵
DataFrame	data.frame()	数据框
List	list()	列表

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) 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。

