

# 脚本添加注释的功能, R会自动忽略其后的内容  
只需注意Error  
运行结果出现问题一定要回答  
# 代码不会补齐  
?代码——查询代码功能作用

运算符  
比较运算符 > < <= >= == != #不等于 - #除了  
逻辑运算符 & #和 | #或 ! #非, 只针对逻辑值  
数据类型  
判断数据类型 is.numeric() is.character() is.logical() as.numeric() as.character() as.logical()  
生成向量  
逐一致一起 c()  
重复向量 rep("x", times=3) 重复  
rnorm(n=3) 生成3个随机数  
seq(from=1, to=100, by=2) 有规律间隔  
past  
past0(rep("x", times=3), 1:3) [1] "x1" "x2" "x3" #0间隔  
past() #默认有空格连接 past(x,y, sep=" ")  
对单个向量操作  
max() min() mean() median() var() #方差 sd() #标准差 sum() length()  
unique() #去重  
sort(x, decreasing=F) #默认升序排列 sort(x, decreasing=T) #降序排列  
sample() #抽样函数 sample(letters[1:5], 6, replace=T) #允许重复抽样, 默认不允许重复抽样  
head() #默认显示数据前6个 head(数据n) #显示前n个  
tail(数据n) #显示最后n个  
对两个向量操作  
x+y x-y #运算, 存在循环补齐  
paste() #连接, 存在循环补齐  
intersect(x,y) #交集 union(x,y) #并集 setdiff(x,y) #X的某种元素Y没有 setdiff(y,x) #Y的某种元素X没有  
x%in%y #y的每个元素都在x中存在吗 (一对多)  
x==y #一对一

数据框 data.frame # 每列只允许一种数据类型  
位置取 x[a] #取第a个 x[a,b] #取第a-b个 x[c(a,b)] #取第a个和第b个 x[-a] #除了a之外的  
逻辑值取 x[x\$in%y] #取TRUE那些  
修改向量元素 x[c(a,b)] <- c(x1,x2)  
scores=c(10,40,47)  
names(scores)=c("jimmy", "nicker", "Damn")  
结果:  
jimmy nicker Damn  
10 40 47  
元素命名  
新建 df <- matrix(1:9, nrow=3)  
矩阵 matrix # 只允许一种数据类型  
转置 # 行列转换位置 t(矩阵名)  
转换 矩阵名 <- as.data.frame(矩阵名) # 转换为数据框  
热图 heatmap: heatmap(矩阵名) heatmap: heatmap(矩阵名, cluster\_cols=F, cluster\_rows=F)  
查看属性 dim() # 看行列  
nrow() # 行数 ncol() # 列数  
rownames() # 提取行名 colnames() # 提取列名  
df\$列名  
取子集 df[a,b] # 提取第a行第b列 df[a,] df[,b] df[c(1,3), 1:2]  
df[df\$a>0]  
df[a,b] <- d df\$gene <- c(12, 11, 14, 2)  
数据框修改 df\$列名称 <- c(a,d,g)  
rownames(df) <- c("a", "f", "w", "g") # 改变行名 colnames(df) <- "r" # 改变列名称  
数据框连接 merge()  
inner\_join(test1, test2, by="name") # 交集  
left\_join(test1, test2, by="name") # 左为主  
right\_join(test1, test2, by="name") # 右为主  
full\_join(test1, test2, by="name") # 全连接  
semi\_join() # 半连接  
anti\_join() # 反连接  
数据框 list # 可装万物  
新建 a <- list(m1=matrix(1:9, nrow=3), m2=matrix(2:9, nrow=2))  
取子集 a[[2]] # 取第2个元素, 返回2, 取出的为list  
a[2] # 取第2个元素, 返回2, 取出的为矩阵  
a\$m1  
删除变量 rm(df1) # 删除数据框df1  
rm(df1, df2) # 删除数据框df1, df2  
rm(list=b0) # 删除整个数据集  
ctrl+i # 清空控制台, 只清空历史, 不清空已运行结果

函数  
形式参数、实际参数、默认参数  
写函数 a <- function(a,b,m=2){(a+b)^m}  
a(3,6,-2) #(3+6)^-2  
函数的集合、配以数据、说明、示例  
安装 CRAN: install.packages("包名称")  
Bioconductor: BiocManager::install("包名称")  
Github: devtools::install\_github("作者用户名/包名称")  
R包  
加载 library(包名称)  
require(包名称) #除了可以加载程序包外, 还会返回一个逻辑值  
if(require(stringr)) install.packages("stringr")  
查看包内有哪些函数 ls("package:包名称")  
包名 函数名  
pheatmap: pheatmap(volcano) 相当于 library(pheatmap); pheatmap(volcano)  
BiocManager: install("limma") 相当于 library(BiocManager); install("limma")  
devtools: install\_github("jzeng1314/dmap1") 相当于 library(devtools); install\_github("jzeng1314/dmap1")  
一次安装, 每次打开新的session都要重新加载

文件的读写  
读取 read.csv("文件全名.csv") # 通常读取csv文件  
read.table("文件全名.txt") # 通常读取txt文件  
load("文件全名.Rsave") # 直接以原格式载入  
read.table("import/xxx.txt") # import是文件名称, # 表示文件夹和地下一级文件夹的分割, # 表示当前目录的上一级目录  
read.table("a.txt", header=T) # 指定第一行为列名  
read.csv("ex2.csv", row.names=1, check.names=F) # row.names=1 把第一列作为行名; check.names=F 不核对名称是否符合R语言的规则  
read.table("a.txt", header=T, sep="\t") # sep="\t" 表示一个分隔符(Tab)  
数据框: write.csv(要导出的数据框的变量名, file="新命名.csv")  
write.table(要导出的数据框的变量名, file="新命名.txt")  
save(要导出的数据框的变量名, file="新命名.txt")  
导出 file.create() # 创建文件  
file.exists() # 文件是否存在于工作目录  
file.remove() # 删除文件  
读取/导出文件的包  
base read.csv() read.table() read.delim() write.csv() write.table()  
readr read\_csv() read\_table() read\_tsv() write\_csv() write\_table()  
data.table包 fread()  
r10包 import() import\_list() export() # 可导出list  
dir() # 列出工作目录下的所有文件 (向量的形式)  
dir(pattern="R") # 列出文件中含有R的文件  
dir(pattern="R3") # 列出以 "R" 为后缀的文件

作图  
base # 基础包绘图函数  
plot() # 多种图形  
hist() # 频率直方图  
boxplot() # 箱线图  
stripchart() # 点图  
barplot() # 柱状图  
dotplot() # 点图  
piechart() # 饼图  
matplot() # 数字图形  
lines() # 添加线  
curve() # 添加曲线  
abline() # 添加指定斜率的线  
points() # 添加点  
segments() # 折线  
arrows() # 箭头  
axis() # 坐标轴  
box() # 外框  
title() # 标题  
text() # 文字  
ggplot2  
ggplot(data=<DATA>)+  
# GEOM\_FUNCTION(mapping\_ase<MAPPINGS>)  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="...", color="...", size="...", shape="...", alpha="...", fill="...")) # alpha透明度, fill填充色, color图形颜色  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="...", color=Species))+  
scale\_color\_manual(values=c("blue", "grey", "red")) # 自定义颜色  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="..."))+  
facet\_wrap(~Species) # 将图按照Species的个数将图分为若干个小图, 并按从左到右从上到下排列  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="..."))+  
facet\_grid(Group~Species) # 将图按照Group分行, Species分列, 其中Group中要有重复值  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="..."))+  
geom\_smooth(mapping\_ase(x="...", y="...")) # 线图和点图叠加 局部设置——当前图层有效  
ggplot(data=<DATA>, mapping\_ase(x="...", y="..."))+  
geom\_point()+  
geom\_smooth() # 线图和点图叠加 全局设置——所有图层有效  
ggplot(data=<DATA>)+  
geom\_bar(mapping\_ase(x=cut)) # 柱状图, 没有y参数, 统计的是的  
ggplot(data=<DATA>)+  
geom\_count(mapping\_ase(x=cut)) # 画一个展示x数量的图  
ggplot(data=<DATA>)+  
geom\_bar(mapping\_ase(x="...", y="...", group=1)) # 将x的分类, 并计算其百分比; group=1不可省略  
# GEOM\_FUNCTION: 画图函数  
# mapping\_ase(<MAPPINGS>): 中的mapping可省略

作图  
ggplot2  
ggplot(data=<DATA>)+  
# GEOM\_FUNCTION(mapping\_ase<MAPPINGS>)  
ggscatter<DATA>: x="变量", y="变量", ... 相当于 geom\_point(mapping\_ase(x=变量, y=变量...))+ theme\_classic()  
ggpubr  
p <- ggboxplot<DATA>: x="变量", y="变量", color="参数", shape="参数", add="jitter")  
my\_comparisons <- list(c("变量1", "变量2"), c("变量1", "变量3"), c("变量2", "变量3"))  
p <- stat\_compare\_means(comparisons=my\_comparisons) # stat\_compare\_means(label.y=9)  
拼图  
par(mfrow=c(2,2), mar=c(5,5,5,5)) # 将数据框整体按照变量从小到大的顺序排列  
cowplot, customLayout # 将数据框整体按照变量从大到小的顺序排列  
博究地址: https://www.data-imagist.com/2019/patch-it-up-and-send-it-out/  
patchwork包官方教程: https://patchwork.data-imagist.com/index.html  
导出  
classic三段论  
pdf("图片名.后缀")  
作图代码, 如: boxplot<DATA>[,1:~>DATA[,5], text(6.5, labels="hello")  
devoff()  
coffice library(coffice)  
toprpt(代码中画图代码所赋值的对象, filename="图片名.pptx") # 将图片保存为PPT, 可再进行编辑调整  
... 画图代码 ...  
ggsave("图片名.后缀")  
或 ggsave(代码中画图代码所赋值的对象, filename="图片名.后缀") # ggplot2, ggpubr等以gg开头的R包都可以用

字符串  
string函数  
str\_length() # 字符串长度 length() # 针对向量的长度  
str\_split() # 字符串拆分  
str\_sub() # 按位置提取字符  
str\_detect() # 字符检测  
str\_replace() str\_replace\_all() # 字符替换  
str\_remove() str\_remove\_all() # 字符删除  
数据框  
arrange(数据框, 变量) # 将数据框整体按照变量从小到大的顺序排列  
arrange(数据框, desc(变量)) # 将数据框整体按照变量从大到小的顺序排列  
去重 distinct<DATA>: 变量, keep\_all=T) # 将变量中第2次及以上出现的数据删除  
新增列 <DATA>=mutate<DATA>, 新列名=计算公式)  
筛选 select()  
filter()  
管道符号 %>%  
if(一个逻辑值) {命令1} # 是, 就执行命令  
if(一个逻辑值) {命令1} else {命令2} # 是执行1, 否执行2  
ifelse(逻辑值, 命令1, 命令2) # 逻辑值可多个, 是执行1, 否执行2  
if() { } # 跳过1里脚本  
if() { } # 执行1里脚本  
循环  
for(i in X1:L) { } # 对X里的每个进行同一操作  
矩阵: 数据框 apply(矩阵, 数据框名, 或2.函数) # 表示行, 2表示列; 对每一行/列进行函数操作  
隐式循环  
sapply<DATA>, 函数) # 与apply相比, 结果为向量, 可操作性强  
lapply<list>, fivenum) # 将list简化为矩阵, 可操作性更强

ggplot2  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="...", color="...", size="...", shape="...", alpha="...", fill="...")) # alpha透明度, fill填充色, color图形颜色  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="...", color=Species))+  
scale\_color\_manual(values=c("blue", "grey", "red")) # 自定义颜色  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="..."))+  
facet\_wrap(~Species) # 将图按照Species的个数将图分为若干个小图, 并按从左到右从上到下排列  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="..."))+  
facet\_grid(Group~Species) # 将图按照Group分行, Species分列, 其中Group中要有重复值  
ggplot(data=<DATA>)+  
geom\_point(mapping\_ase(x="...", y="..."))+  
geom\_smooth(mapping\_ase(x="...", y="...")) # 线图和点图叠加 局部设置——当前图层有效  
ggplot(data=<DATA>, mapping\_ase(x="...", y="..."))+  
geom\_point()+  
geom\_smooth() # 线图和点图叠加 全局设置——所有图层有效  
ggplot(data=<DATA>)+  
geom\_bar(mapping\_ase(x=cut)) # 柱状图, 没有y参数, 统计的是的  
ggplot(data=<DATA>)+  
geom\_count(mapping\_ase(x=cut)) # 画一个展示x数量的图  
ggplot(data=<DATA>)+  
geom\_bar(mapping\_ase(x="...", y="...", group=1)) # 将x的分类, 并计算其百分比; group=1不可省略  
抖动点图  
ggplot(data=<DATA>, mapping\_ase(x="...", y="..."))+  
geom\_boxplot()+  
geom\_point() # 点图的点是重复的, 若想展示建议应用抖动点图, 将geom\_point() 换为geom\_jitter()  
堆叠直方图  
ggplot(data=<DATA>)+  
geom\_bar(mapping\_ase(x="...", fill="...")) # x根据fill分层  
并列直方图  
ggplot(data=<DATA>)+  
geom\_bar(mapping\_ase(x="...", fill="..."), position="dodge") # x根据fill分层  
翻转坐标系  
ggplot(data=<DATA>, mapping\_ase(x="...", y="..."))+  
geom\_函数(x)+  
coord\_flip() # x轴y轴转换位置  
极坐标系  
bar <- ggplot(data=diamonds)+  
geom\_bar(mapping\_ase(x=变量, fill=同一变量), width=1)+  
theme(aspect\_ratio=1)+  
labs(x=NULL, y=NULL)  
bar + coord\_flip()  
bar + coord\_polar() # 用扇形大小表示数值多少

# GEOM\_FUNCTION: 画图函数  
# mapping\_ase(<MAPPINGS>): 中的mapping可省略

R语言常用函数代码

字符串

数据框

条件语句

循环

隐式循环

# GEOM\_FUNCTION: 画图函数  
# mapping\_ase(<MAPPINGS>): 中的mapping可省略