R语言简介与语法

中南大学生命科学学院

刘可夫

2021年

你在将来的课题中有多大概率/频率会用到R/bioinformatic？

你最有可能/最希望用R开展什么分析?

作业：

课堂练习 (个人作业)　　课后协作 (团队作业)

作业提交：

注册账号，作业提交至

GitHub

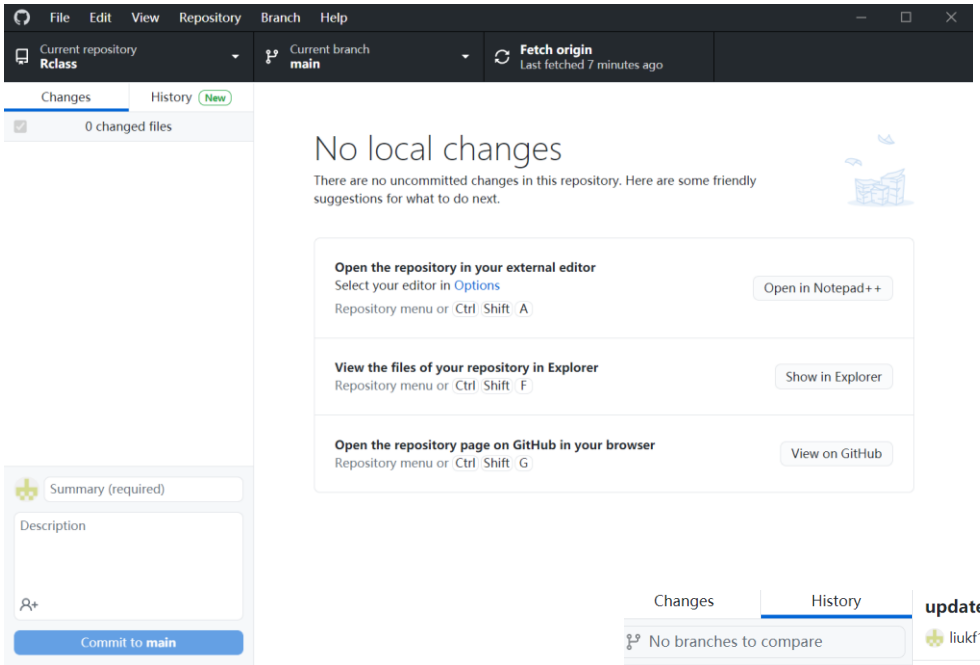
Gitee: 码云

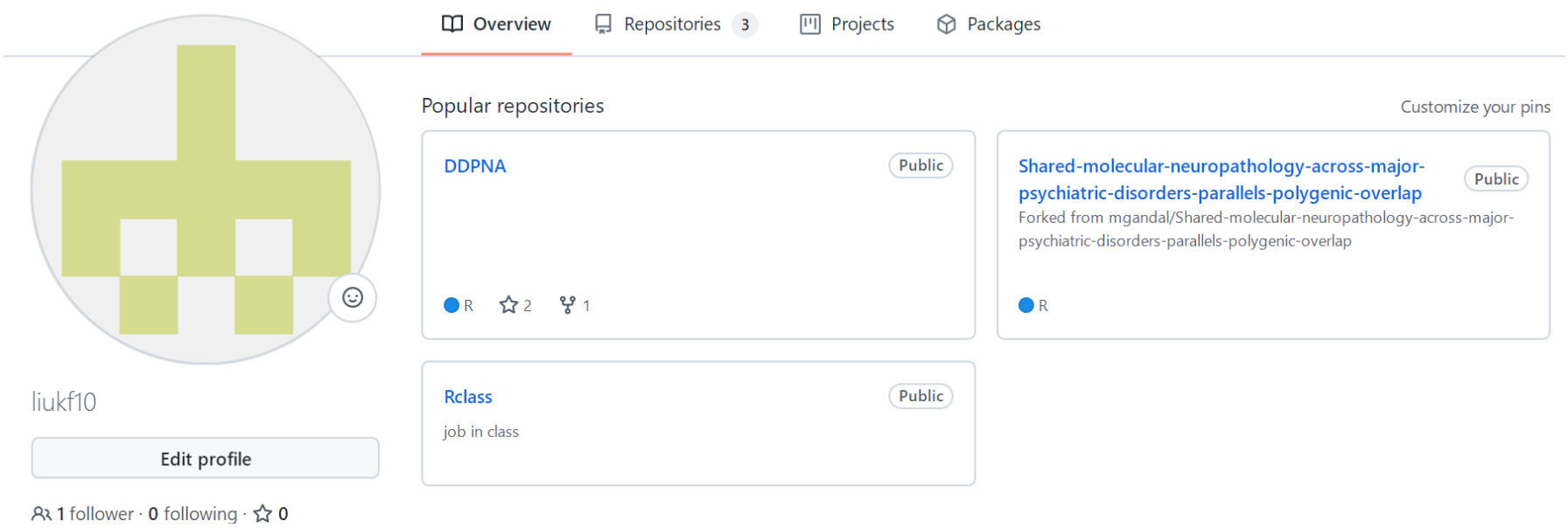
个人作业上传到个人仓库，团队作业由组长归档上传到团队作业仓库

GitHub Desktop

https://desktop.github.com/

GitHub Desktop





作业发布：

https://github.com/liukf10/Rclass

https://gitee.com/liukf10/Rclass/

liukf10 / Rclass Public

Unwatch 1 | Star 0 | Fork 0

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

main | 1 branch | 0 tags | Go to file | Add file | Code

liukf10 update for class1 ... f1ef016 6 minutes ago 2 commits

practice1	update for class1	6 minutes ago
.gitattributes	Initial commit	3 days ago
.gitignore	Initial commit	3 days ago
LICENSE	Initial commit	3 days ago
README.md	update for class1	6 minutes ago

About

job in class

Readme

AGPL-3.0 License

Releases

No releases published

Create a new release

GSE67835	update for class1	7 minut
diabets.RData	update for class1	7 minut
readme.md	update for class1	7 minut

readme.md

#练习1

题目1

读取diabets.RData文件，利用for, while, if, tapply, cut或者其他函数组合，计算diabets数据中GLU值在3.9至6.1中间的女性数量。（尝试不少于2种函数组合实现）

题目2

将GSE67835文件夹中的csv文件读入并合并成一个数据框

课后协作 (团队作业)

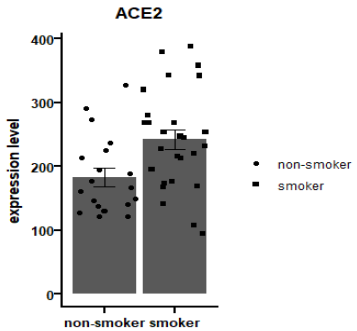
For your research purpose

1. barplot / dotplot one-way
2. barplot / dotplot two-way
3. boxplot / violin plot one-way
4. boxplot / violin plot two-way
5. One-way ANOVA or T test analysis
6. Two-way ANOVA analysis
7. 带颜色散点图
8. heatmap

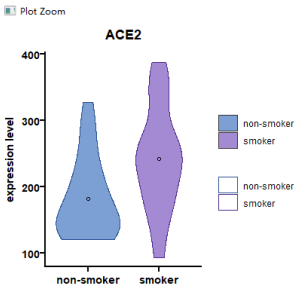
Welcome to submit your request

…

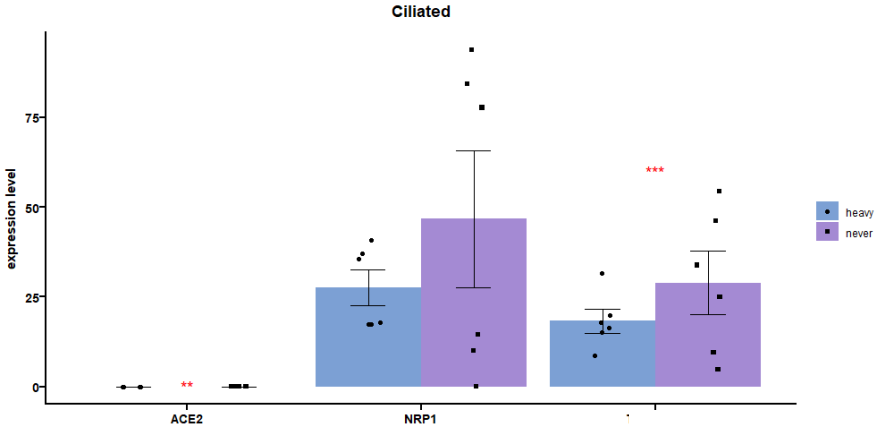
barplot / dotplot one-way



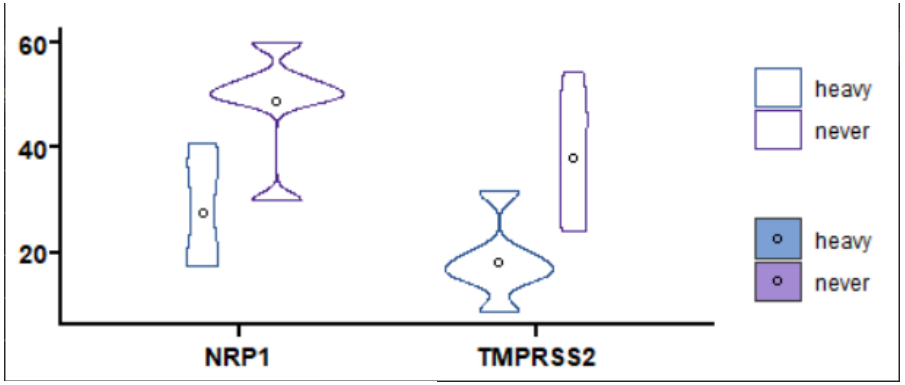
boxplot / violin plot one-way

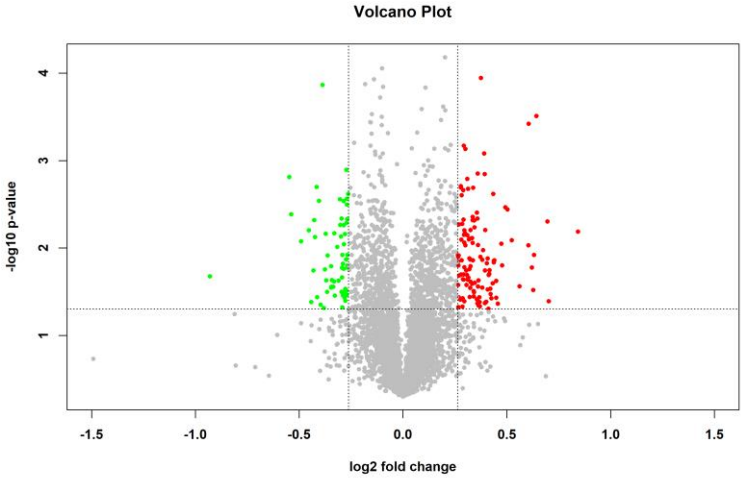


barplot / dotplot two-way



boxplot / violin two-way





团队作业内容:

1. **Maintainer is team leader.**
2. **Readme file (说明文档)**
3. **Code file (详细注释信息, 各个contributor 贡献)**
4. **Example file (测试使用)**
5. **PPT**
6. **PPT maker /PPT speaker and Maintainer should not the same people**

What is R

R 语言是一门统计计算与作图的语言。

- R是基于S语言的一个GNU项目，所以也可以当做S语言的一种实现
- 词法是Scheme，词源是S语言
- 由新西兰奥克兰大学Ross Ihaka和Robert Gentleman开发
- 是为了统计计算而开发


Why use R

- **免费开源！！**
- 统计研究功能全面详尽，面向统计分析研究人员而设计, 涵盖了基础统计学、社会学、经济学、生态学、地理学、医学统计学、生物信息学等多方面内容。
- 是程序设计语言，允许分支和循环以及使用函数的模块化编程，可以使用用户定义的函数扩展，提供了大量的用户开发的Package
- 制图功能强大
- 允许C，C++，.Net，Python或FORTRAN语言编写的过程集成
- 多个平台均能使用（Windows, UNIX,Mac 等）

SPSS

STATA

SAS

VS

R

- 商业软件，需要收费
- 功能固定，更新慢
- 非开源，内部算法未知
- 在商界目前还是主流（主要针对统计师）

- 免费
- 功能非常多，更新快
- 开源，内部算法可追溯
- 学术界主流，商界也开始兴起，尤其对于生物信息分析师，R不可或缺

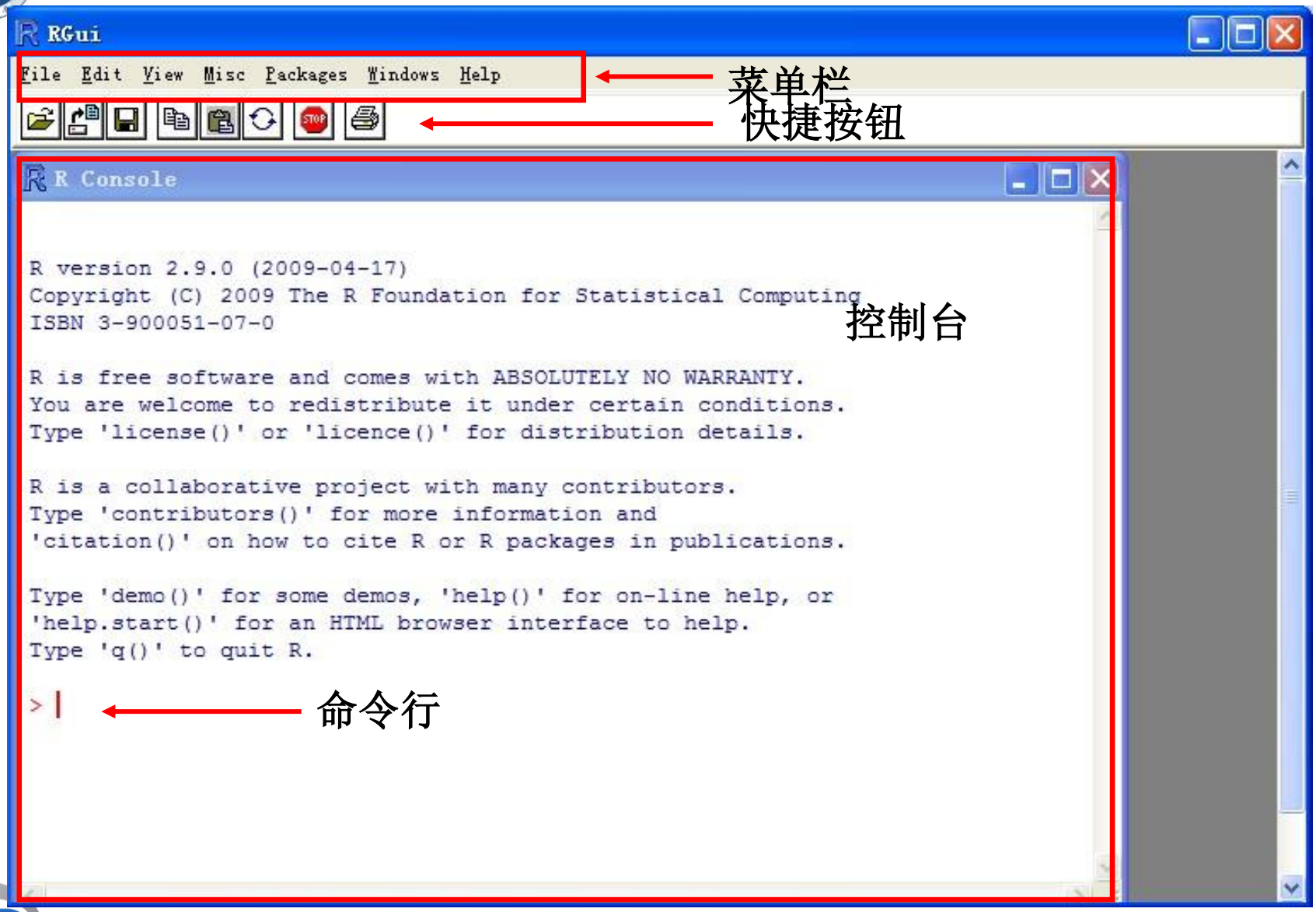
R

R website

http://www.r-project.org
R主页 R基本介绍，最新版本的R及其更新主要特性

http://cran.r-project.org
Comprehensive R Archive Network
R包的一个发布平台，R下载 以及主要的R包的下载

https://www.bioconductor.org/
Bioconductor: 基于R语言地用于解决生物学高通量数据处理的软件包，数据包，及注释包的集合
有大量生物信息学相关包

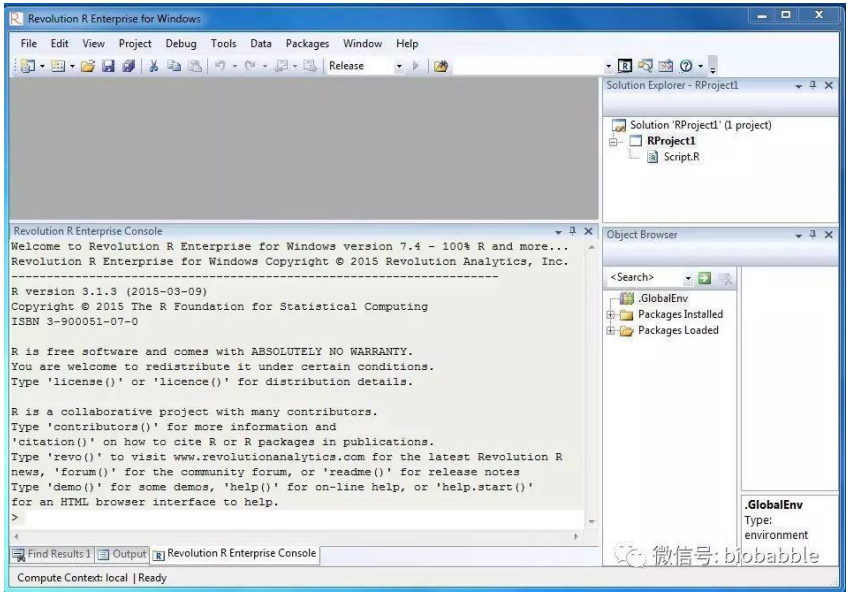


R登陆界面(Windows版)　　不推荐

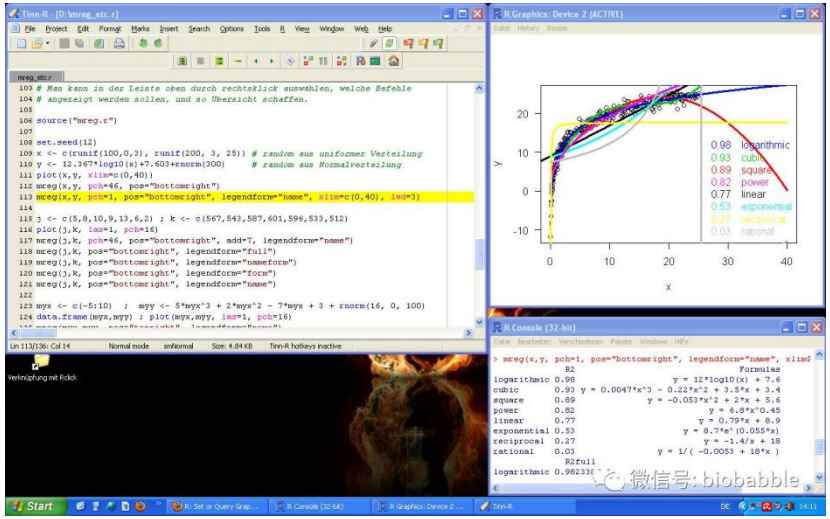
Other platform

Windows系统

Revolution R Enterprise

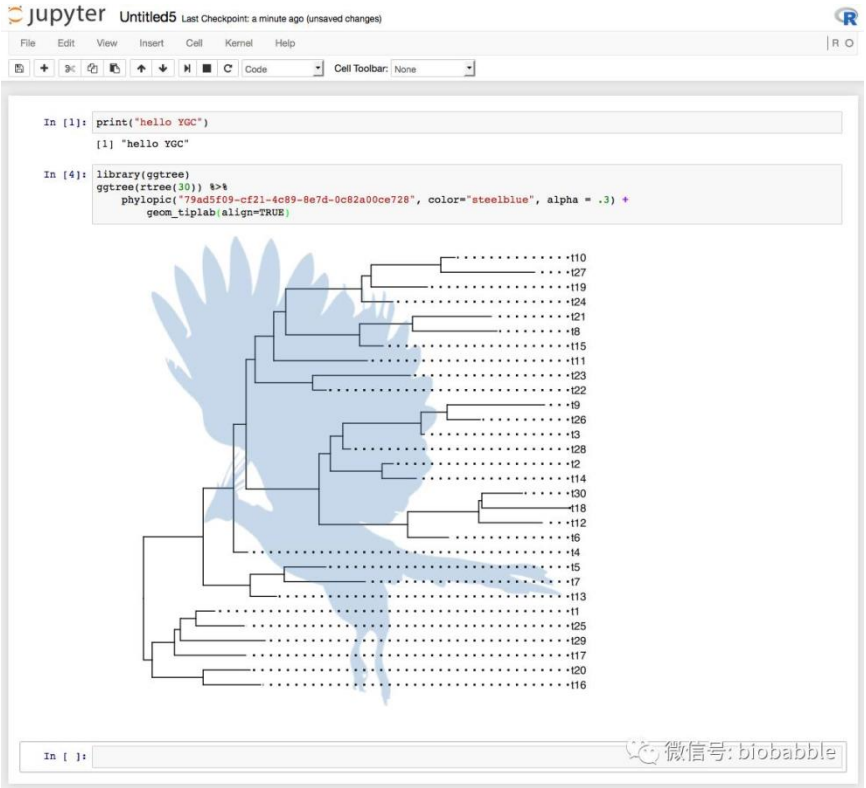


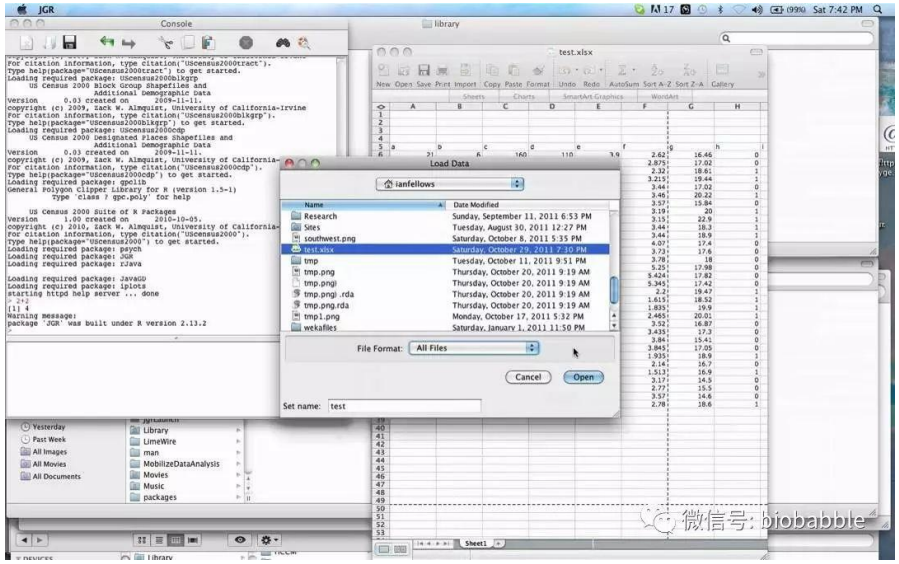
tinn-R



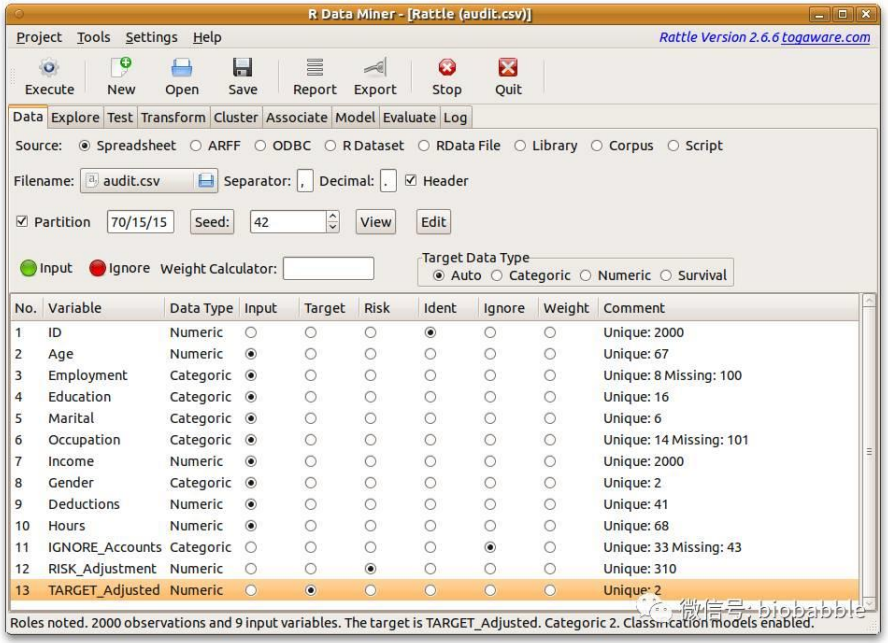
jupyter

跨平台 基于java: JGR

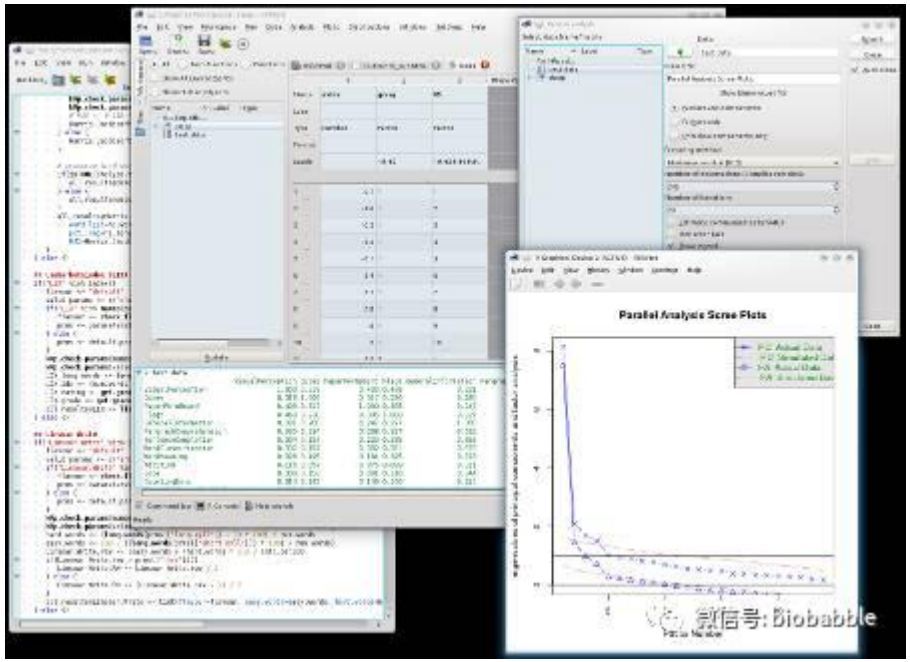




Linux GTK界面： rattle



基于QT的RKWard



各个平台都有其自己的优缺点，每个人可以选择自己最适合的R平台。

但是目前最主流，对于大部分而言最便捷直观的是R Studio，我们也使用该平台进行所有的R相关功能探索。

R in Bioinformatics

大量的生物信息学常用Package:

ggplot2以及一系列衍生包：绘图神器

Seurat: 单细胞分析

DESeq2,limma, EdgeR: 差异分析

clusterProfiler: 富集分析

Biostring: 生物序列处理

R in Bioinformatics

解决主要是以海量DNA信息数据为主的数据处理，提取，比对，分析，以及统计等问题

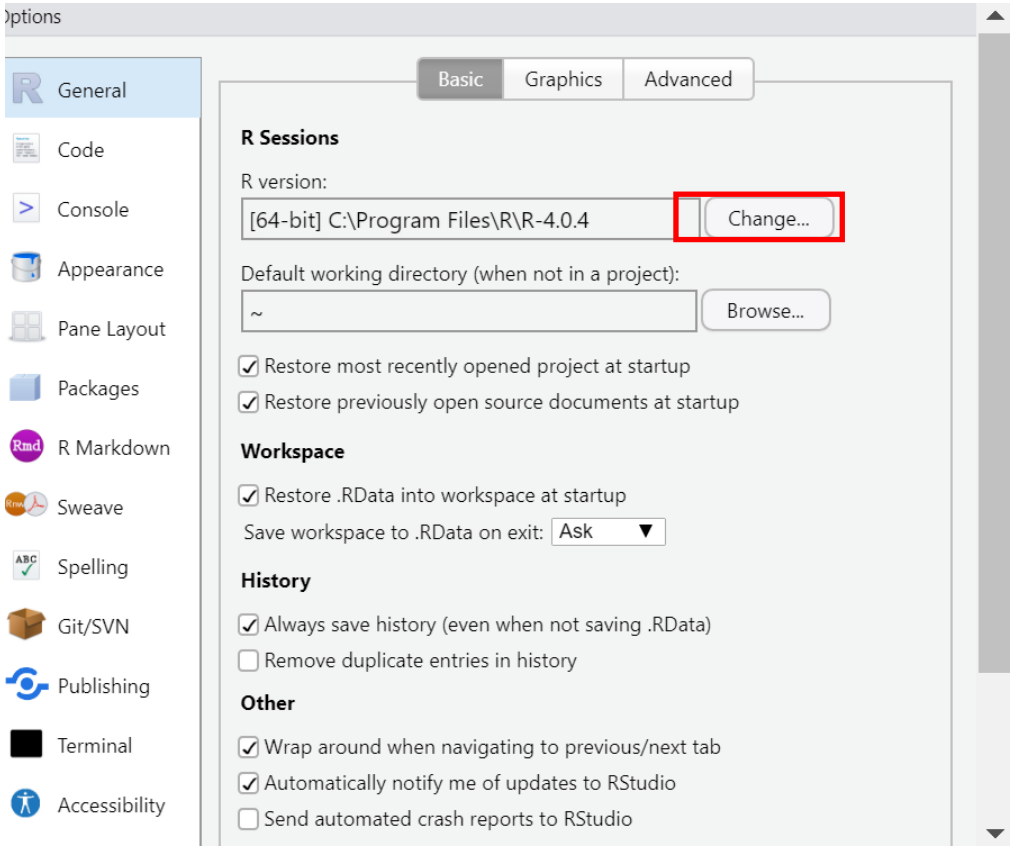
数据的图形展示等多种非人工可以实现的功能

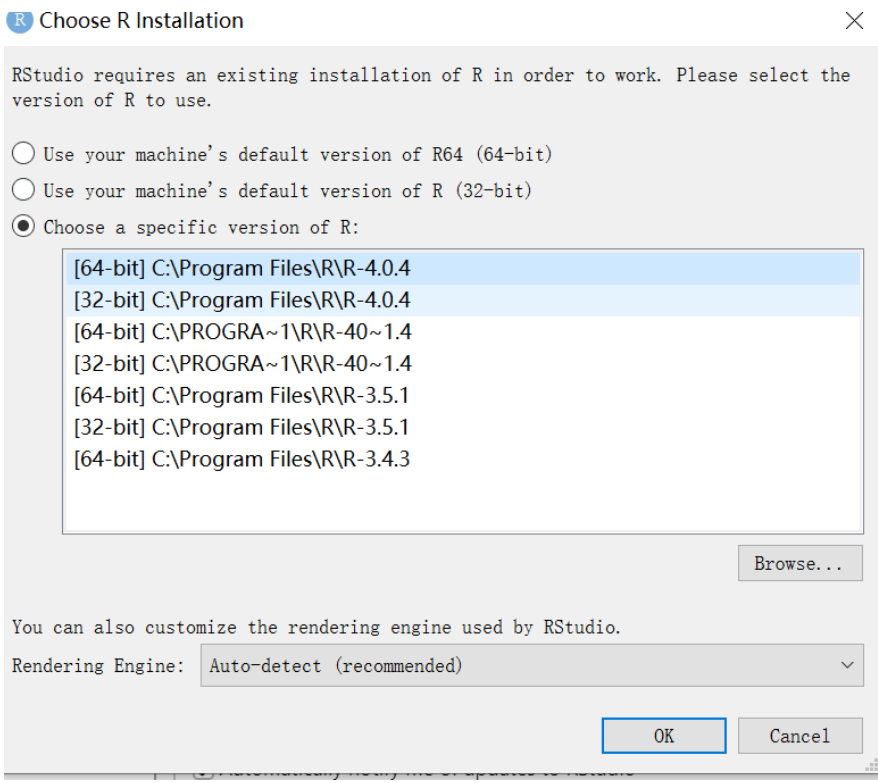
R源程序自带的基础包

System Library

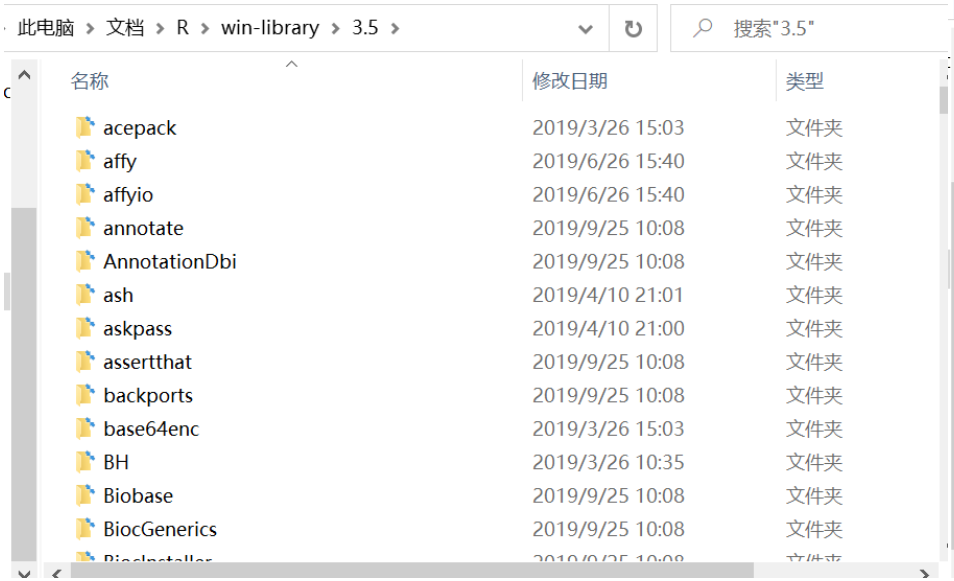
☑	base	The R Base Package	4.0.4
☐	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-26
☐	class	Functions for Classification	7.3-18
☐	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.0
☐	codetools	Code Analysis Tools for R	0.2-18
☐	compiler	The R Compiler Package	4.0.4
☑	datasets	The R Datasets Package	4.0.4
☐	MASS	Support Functions and Datasets for Venables and Ripley's MASS	7.3-53
☐	Matrix	Sparse and Dense Matrix Classes and Methods	1.3-2
☑	methods	Formal Methods and Classes	4.0.4
☐	mgcv	Mixed GAM Computation Vehicle with Automatic Smoothness Estimation	1.8-33
☐	nlme	Linear and Nonlinear Mixed Effects Models	3.1-152
☐	nnet	Feed-Forward Neural Networks and Multinomial Log-Linear Models	7.3-15
☐	foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-81
☑	graphics	The R Graphics Package	4.0.4
☑	grDevices	The R Graphics Devices and Support for Colours and Fonts	4.0.4
☐	grid	The Grid Graphics Package	4.0.4
☐	KernSmooth	Functions for Kernel Smoothing Supporting Wand & Jones (1995)	2.23-18
☐	lattice	Trellis Graphics for R	0.20-41
☐	parallel	Support for Parallel computation in R	4.0.4
☐	rpart	Recursive Partitioning and Regression Trees	4.1-15
☐	spatial	Functions for Kriging and Point Pattern Analysis	7.3-13
☐	splines	Regression Spline Functions and Classes	4.0.4
☑	stats	The R Stats Package	4.0.4
☐	stats4	Statistical Functions using S4 Classes	4.0.4
☐	survival	Survival Analysis	3.2-7
☐	tcltk	Tcl/Tk Interface	4.0.4
☐	tools	Tools for Package Development	4.0.4
☐	translations	The R Translations Package	4.0.4
☑	utils	The R Utils Package	4.0.4

Rstudio 支持多版本R源切换





Rstudio 支持多版本R源切换



不同版本之间安装好的包在各自的文件夹下，并不共用

R 包安装

CRAN包安装 install.packages(“ggplot2”)

Bioconductor包安装

```
if (!requireNamespace("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install(version = "3.12")
BiocManager::install("Biostrings")
```

Github上发布的包安装

```
install.packages("devtools")
library(devtools)
install_github("RevolutionAnalytics/RHad
oop")
```

R help文件

Files | Plots | Packages | Help | Viewer

t.test | Refresh Help Topic

R: Student's t-Test | Find in Topic

t.test {stats} R Documentation

Student's t-Test

Description

Performs one and two sample t-tests on vectors of data.

Usage

```
t.test(x, ...)

## Default S3 method:
t.test(x, y = NULL
```

Details

`alternative = "greater"` is the alternative that `x` has a larger mean than `y`. For the one-sample case: that the mean is positive.

If `paired` is `TRUE` then both `x` and `y` must be specified and they must be the same length. Missing values are silently removed (in pairs if `paired` is `TRUE`). If `var.equal` is `TRUE` then the pooled estimate of the variance is used. By default, if `var.equal` is `FALSE` then the variance is estimated separately for both groups and the Welch modification to the degrees of freedom is used.

If the input data are effectively constant (compared to the larger of the two means) an error is generated.

Arguments

`x`	a (non-empty) numeric vector of data values.
`y`	an optional (non-empty) numeric vector of data values.
`alternative`	a character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`. You can specify just the initial letter.
`mu`	a number indicating the true value of the mean (or difference in means if you are performing a two sample test).
`paired`	a logical indicating whether you want a paired t-test.

Value

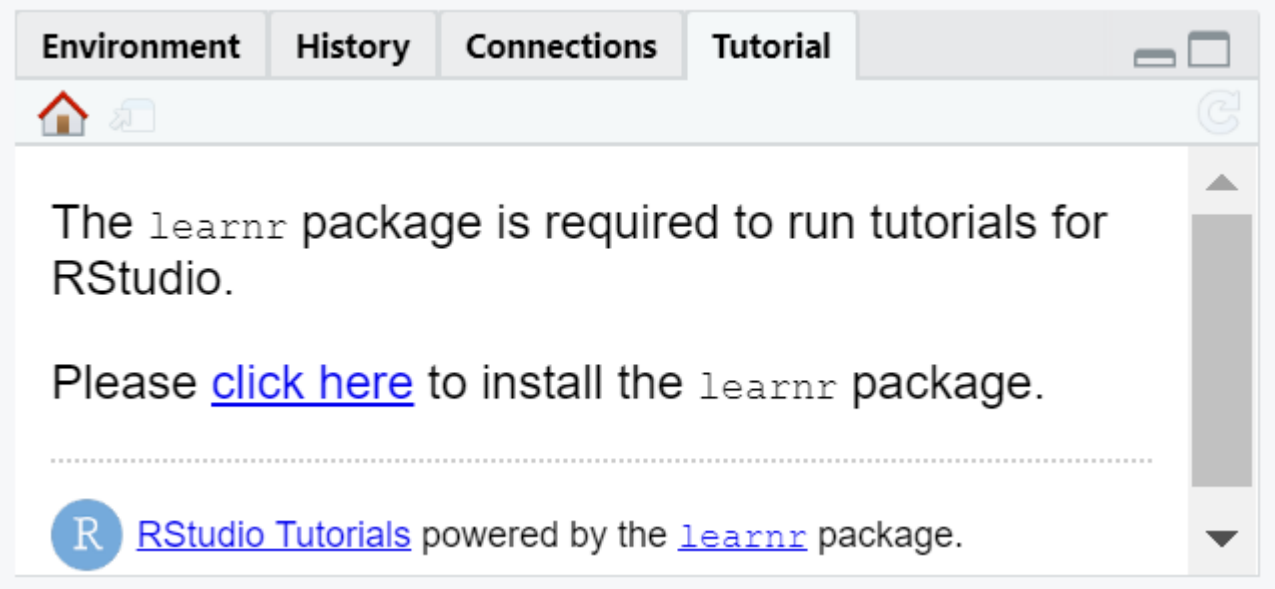
A list with class `"htest"` containing the following components:

`statistic`	the value of the t-statistic.
`parameter`	the degrees of freedom for the t-statistic.
`p.value`	the p-value for the test.
`conf.int`	a confidence interval for the mean appropriate to the specified alternative hypothesis.
`estimate`	the estimated mean or difference in means depending on whether it was a one-sample test or a two-sample test.
`null.value`	the specified hypothesized value of the mean or mean difference depending on whether it was a one-sample test or a two-sample test.

Examples

```
require(graphics)

t.test(1:10, y = c(7:20))      # P = .00001855
t.test(1:10, y = c(7:20, 200)) # P = .1245    -- NOT significant

## Classical example: Student's sleep data
plot(extra ~ group, data = sleep)
## Traditional interface
with(sleep, t.test(extra[group == 1], extra[group == 2]))
```

Rstudio 新版本新功能Tutorial
Rstudio 2017年出了一个learnr的包 互动式教程

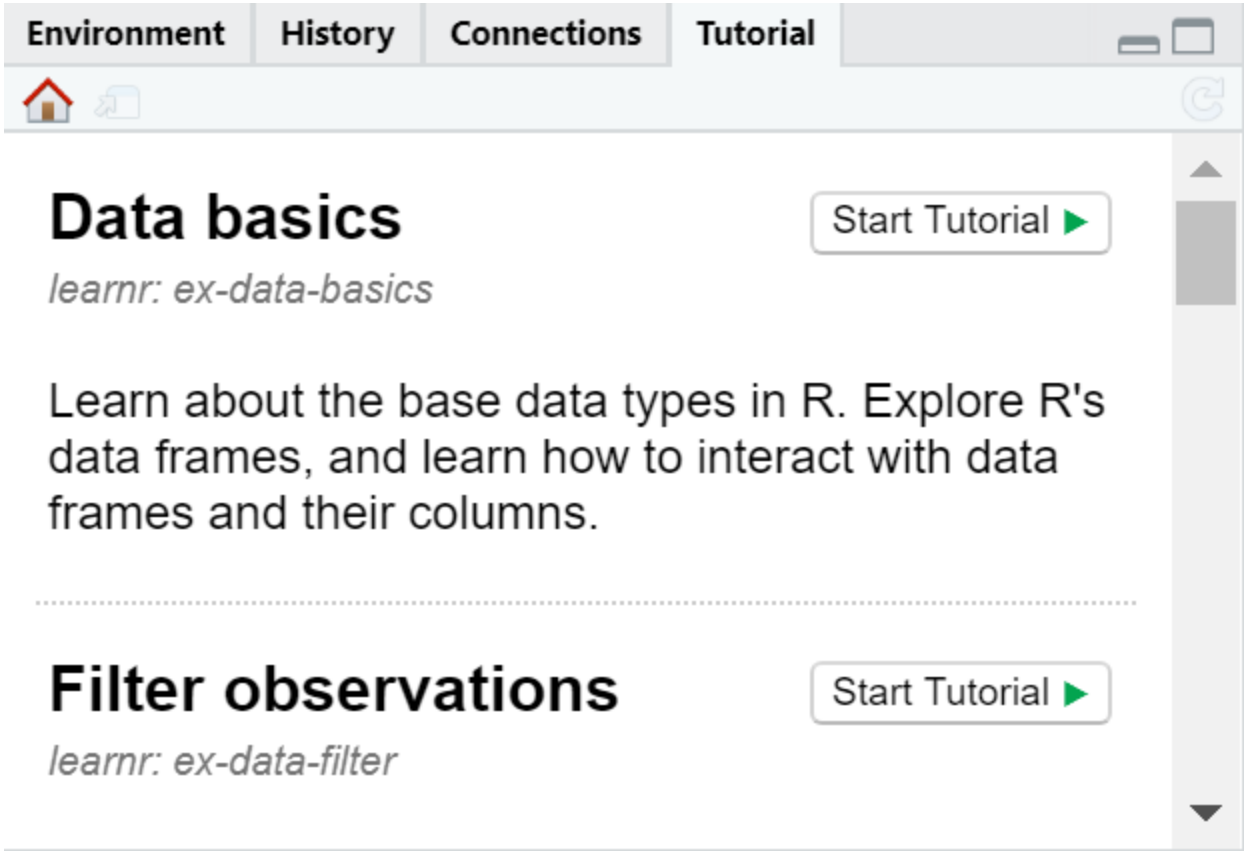


R 学习

```
> install.packages("learnr")
Installing package into ‘C:/Users/lkf/Documents/R/win-library/4.0’
(as ‘lib’ is unspecified)
also installing the dependencies ‘fs’, ‘magrittr’, ‘stringi’, ‘Rcpp’,
 ‘BH’, ‘sass’, ‘jquerylib’, ‘digest’, ‘base64enc’, ‘rlang’, ‘yaml’, ‘h
ighr’, ‘stringr’, ‘xfun’, ‘mime’, ‘httpuv’, ‘xtable’, ‘R6’, ‘sourcetoo
ls’, ‘later’, ‘promises’, ‘crayon’, ‘fastmap’, ‘commonmark’, ‘glue’,
 ‘bslib’, ‘cachem’, ‘lifecycle’, ‘tinytex’, ‘backports’, ‘withr’, ‘rap
pdirs’, ‘rprojroot’, ‘jsonlite’, ‘htmltools’, ‘htmlwidgets’, ‘evaluat
e’, ‘knitr’, ‘markdown’, ‘shiny’, ‘rmarkdown’, ‘ellipsis’, ‘checkmat
e’, ‘renv’
```

```
package ‘knitr’ successfully unpacked and MD5 sums checked
package ‘markdown’ successfully unpacked and MD5 sums checked
package ‘shiny’ successfully unpacked and MD5 sums checked
package ‘ellipsis’ successfully unpacked and MD5 sums checked
package ‘checkmate’ successfully unpacked and MD5 sums checked
package ‘renv’ successfully unpacked and MD5 sums checked
package ‘learnr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\lkf\AppData\Local\Temp\Rtmpyk2Kbm\downloaded_packages
installing the source packages ‘cachem’, ‘rmarkdown’

试开URL’https://mirrors.tuna.tsinghua.edu.cn/CRAN/src/contrib/cachem_1.
0.4.tar.gz'
Content type 'application/x-gzip' length 24493 bytes (23 KB)
downloaded 23 KB

试开URL’https://mirrors.tuna.tsinghua.edu.cn/CRAN/src/contrib/rmarkdown
_2.7.tar.gz'
Content type 'application/x-gzip' length 3215356 bytes (3.1 MB)
downloaded 3.1 MB

* installing *source* package 'cachem' ...
** 成功将'cachem'程序包解包并MD5和检查
** using staged installation
** libs
```

R 学习


Environment
History
Connections
Tutorial
Data basics
Start Tutorial
learnr: ex-data-basics
Learn about the base data types in R. Explore R's data frames, and learn how to interact with data frames and their columns.
Filter observations
Start Tutorial
learnr: ex-data-filter

R 学习

Environment | History | Connections | Tutorial

Data basics

Data frames

✓ What is a data frame?

A **data frame** is a rectangular collection of values, usually organized so that variables appear in the columns and observations appear in rows.

Here is an example: the `mpg` data frame contains observations collected by the US Environmental Protection Agency on 38 models of cars. To see the `mpg` data frame, type `mpg` in the code chunk below and then click "Submit

the US Environmental Protection Agency on 38 models of cars. To see the `mpg` data frame, type `mpg` in the code chunk below and then click "Submit
Answer."

Code | Start Over | Hint | Run Code | Submit Answer

```
1
2
3
```

Continue

语法

- R 是一种语法非常简单的表达式语言，大小写敏感
- 命名字符集依赖于R 所运行的系统和国家(就是系统的local 设置)。通常，数字，字母，.和_都是允许的(在一些国家还包括重音字母)
- 一个命名必须以. 或者字母开头，并且以. 开头时第二个字符不允许是数字
- 基本命令要么是表达式（expressions）要么就是赋值（assignments）。
- 如果一条命令是表达式，那么它将会被解析，并将结果显示在屏幕上，同时清空该命令所占内存。赋值同样会解析表达式并且把值传给变量但结果不会自动显示在屏幕上。
- 命令可以被(;)隔开，或者另起一行。
- 基本命令可以通过大括弧{和} 放在一起,构成一个复合表达式（compound expression）。
- 如果一条命令在一行结束的时候在语法上还不完整， R 会给出一个不同的提示符，默认是+

表达式（expressions）

```
> 1+1
[1] 2
> t.test(c(1:5),c(2,5,4,3,6))

        Welch Two Sample t-test

data:  c(1:5) and c(2, 5, 4, 3, 6)
t = -1, df = 8, p-value = 0.3466
alternative hypothesis: true difference in means is n
ot equal to 0
95 percent confidence interval:
 -3.306004  1.306004
sample estimates:
mean of x mean of y
        3         4
```

赋值（assignments）

赋值符

= 或 <-

```
> a = 1
> A = 2
> a1 = 3
> a.1 = 4
> a_1 = 5
> 中文 = "chinese"
> 流程 = "其他"
> x <- 6
>
```

Environment | History | Connections | Tutorial

Import Dataset | List

R | Global Environment

Values

流程	"\u5176\u4ed6"
中文	"chinese"
a	1
A	2
a_1	5
a.1	4
a1	3
x	6

```
> 1a<-5
错误: unexpected symbol in "1a"
```

反向赋值符

->

Console Terminal Jobs

```
> 1->x
>
```

Environment History Connections Tutorial

Import Dataset

R Global Environment

Values

x	1

```
> z=1
> x<-z->y->w
>
>
```

Environment History Connections Tutorial

Import Dataset

R Global Environment

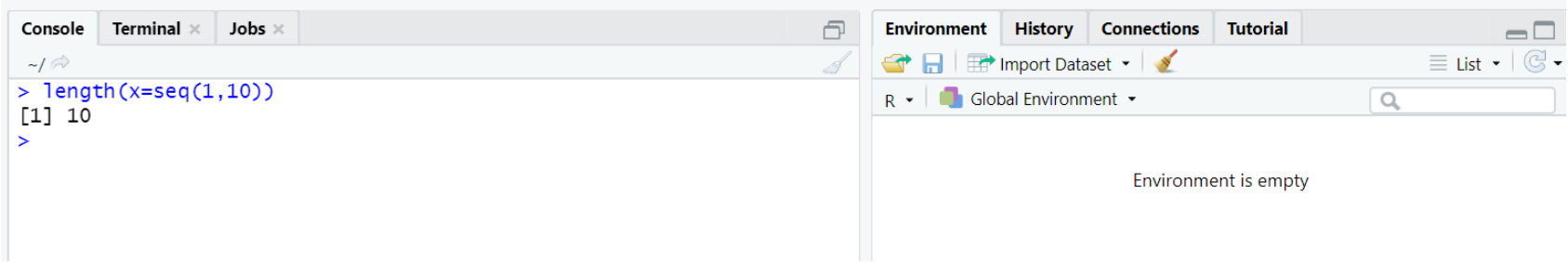
Values

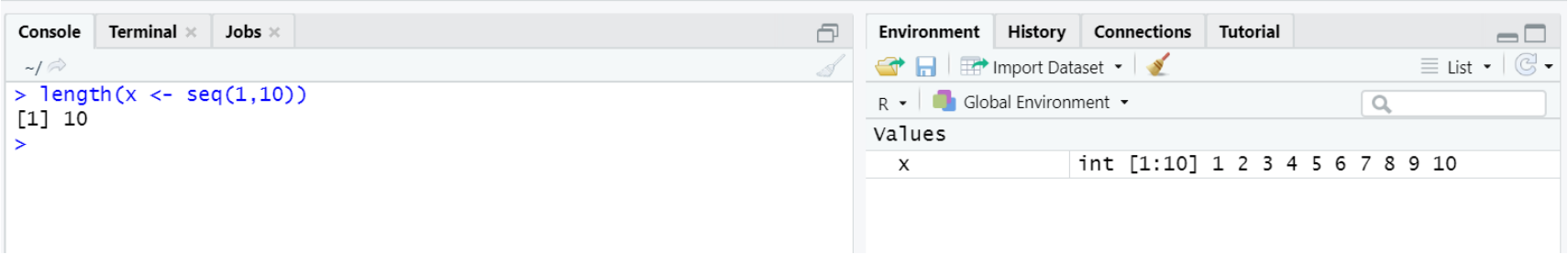
w	1
x	1
y	1
z	1

语法

= 可以是赋值，也可以是传参

<- 只能是赋值






```
> y <- x = 10
Error in y <- x = 10 : 没有"<-<-"这个函数
> y = x <- 10
>
```

```
> x <<- 1
> 1 ->> x
>
```

语法

可以分行识别

```
> x=1
> y=2
>
```

```
> x=1;y=2
```

Untitled1*

Source on Save Run Source

```
x=c(1
    ,2)
```

```
> x=c(1
+     ,2)
```

语法

- 一行中，从井号(#)开始到句子收尾之间的语句就是注释。

Untitled1*

Source on Save

```
# x是 阿伏伽德罗常数
x = 6.02214076*10^23
```

Console Terminal Jobs

~/

```
> # x是 阿伏伽德罗常数
> x = 6.02214076*10^23
>
```

语法

- 符号
  - > 命令或运算提示符
  - + 续行符

```
> x=c(1
+      ,2)
```

不需要我们手动输入

- 基本算术运算
  - + 加号
  - - 减号
  - * 乘号
  - / 除号
  - ^ 乘方

```
> 1+1
[1] 2
> 10-1
[1] 9
> 2*5
[1] 10
> 10/2
[1] 5
> 2^5
[1] 32
```

数据类型

- 向量 vector　　因子 factor
- 矩阵 matrix　　数据框 data.frame
- 列表 list　　数组 array
- 特殊面向对象数据类型S3, S4

对象	类型	是否允许同一对象内有多种类型
向量	数值型、字符型、复数型、逻辑型	否
因子	数值型、字符型	否
数组	数值型、字符型、复数型、逻辑型	否
矩阵	数值型、字符型、复数型、逻辑型	否
数据框	数值型、字符型、复数型、逻辑型	是
时间序列	数值型、字符型、复数型、逻辑型	否
列表	数值型、字符型、复数型、逻辑型、函数、表达式等	是

向量

- 一个数据序列；
- 比如：

序号：(1,2,3,4,5)

成绩：(89，90，78，88)

“ad12314”

“R语言在生物信息学中的应用”

产生方式

seq() : 向量(序列)具有较为简单的规律
rep() 向量(序列)具有较为复杂的规律
c() 向量(序列)没有什么规律

A <- c(1:3)

B <- seq(from =1 , to = 5, by =1)

D <- rep(0,5)

Values	
A	int [1:3] 1 2 3
B	num [1:5] 1 2 3 4 5
D	num [1:5] 0 0 0 0 0

修改向量函数

- append()　　插入数据　　append(A,1)
- replace()　　替换　　replace(A,1,2)
- 向量的下标(index)与向量子集(元素)的提取
  - 正的下标　　提取向量中对应的元素
  - 负的下标　　去掉向量中对应的元素
  - 逻辑运算　　提出向量中元素的值满足条件的元素

注：R中向量的下标从1开始，这与通常的统计或数学软件一致而与比如C语言等计算机高级语言不一致，它们的向量下标则从0开始！

```
A <- c(1:10);
#删除第3个元素
A[-3]
#删除位置7、8的元素
A[-c(7,8)]
#第3个元素
A[3]
#A向量里面小于5的所有数
A[A < 5]
```

```
> A <- c(1:10);
> #删除第3个元素
> A[-3]
[1]  1  2  4  5  6  7  8  9 10
> #删除位置7、8的元素
> A[-c(7,8)]
[1]  1  2  3  4  5  6  9 10
> #第3个元素
> A[3]
[1] 3
> #A向量里面小于5的所有数
> A[A < 5]
[1] 1 2 3 4
```

向量运算中的循环法则(recycling rule)

>1:2+1:4

$$\begin{bmatrix}1\\2\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\end{bmatrix}=\begin{bmatrix}1\\2\\1\\2\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\end{bmatrix}=\begin{bmatrix}2\\4\\4\\6\end{bmatrix}$$

>1:4+1:7

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\end{bmatrix}=\begin{bmatrix}1\\2\\3\\4\\1\\2\\3\end{bmatrix}+\begin{bmatrix}1\\2\\3\\4\\5\\6\\7\end{bmatrix}=\begin{bmatrix}2\\4\\6\\8\\6\\8\\10\end{bmatrix}$$

复数型和特殊值

- 复数

  a = 2+5i

  b = 5i

- 无穷数

  Inf

  -Inf

```
> 1/0
[1] Inf
> -1/0
[1] -Inf
```

- 缺失值

  NA #Not available

  NaN #Not a Number

- 空值

  NULL

字符型

```
x = "This is character string!"
y = c("1", "abc", "", " ", "Show“, 2)
x
y
```

```
> x = "This is character string!"
> y = c("1", "abc", "", " ", "Show",2)
> x
[1] "This is character string!"
> y
[1] "1"    "abc"  ""     " "    "Show" "2"
>
```

```
x = "R语言在生物信息学中的应用"
print(x)
cat(x)
```

```
> x = "R语言在生物信息学中的应用"
> print(x)
[1] "R\u8bed\u8a00\u5728\u751f\u7269\u4fe1\u606f\u5b66\u4e2d
\u7684\u5e94\u7528"
> cat(x)
R语言在生物信息学中的应用>
>
```

- 字符串分割函数：strsplit()
  - strsplit(s,分隔符)
- 字符串连接函数：paste()及paste0()
  - paste(s,s2,sep=)及paste0(s,s2)
- 计算字符串长度：nchar()及length()
  - nchar(s)及length(s)
- 字符串截取函数：substr()及substring()
  - substr(s,start,stop)
  - substring(s,start,stop=lenth(s))
-
- 正则表达式

  正则表达式通常被用来检索、替换那些符合某个模式(规则)的文本。

逻辑型

逻辑型向量的值可以是TRUE、FALSE、NA

A = 1:10
X = A < 5
x

```
> A = 1:10
> x = A < 5
> x
 [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
 [9] FALSE FALSE
>
```

```
> x=c(TRUE,FALSE,NA)
> A[x]
[1]  1 NA  4 NA  7 NA 10
```

排序函数

sort(xx) #从小到大排序

rev(xx) #反排列， 所以从大到小排序

order(xx) #返回从小到大排列的位置

rank(xx) # 返回的是对向量中每个数值对应的秩

```
> data=c(5,6,8,2,4,9)
> sort(data)
[1] 2 4 5 6 8 9
> rank(data)
[1] 3 4 5 1 2 6
> order(data)
[1] 4 5 1 2 3 6
> data[order(data)] #等同于sort(data)
[1] 2 4 5 6 8 9
>
```

因子

- 人的性别：男、女
- 年龄段：小孩、年轻、中年、老年

- factor(x = character(), levels, labels = levels, exclude = NA, ordered = is.ordered(x), nmax = NA)

- levels： 用来指定因子可能的水平（缺省值是向量x中互异的值）
- labels：用来指定水平的名字
- exclude：表示从向量x中剔除的水平值

因子

- >colour <- c('G', 'G', 'R', 'Y', 'G', 'Y', 'Y', 'R', 'Y')
- >col <- factor(colour)
- >col1 <- factor(colour, levels = c('G', 'R', 'Y'), labels = c('Green', 'Red', 'Yellow'))

#labels的内容替换colour相应位置对应levels的内容

- col2 <- factor(colour, levels = c('G', 'R', 'Y'), labels = c('1', '2', '3'))

因子

- ordered()
  - >score <- c('A', 'B', 'A', 'C', 'B')
  - >score1 <- ordered(score, levels = c('C', 'B', 'A'))

- cut()函数
  - >exam <- c(98, 97, 52, 88, 85, 75, 97, 92, 77, 74, 70, 63, 97, 71, 98, 65, 79, 74, 58, 59, 60, 63, 87, 82, 95, 75, 79, 96, 50, 88)
    >exam1 <- cut(exam, breaks = 3) #切分成3组区间，区间步长这样计算(max(exam)-min(exam))/3

因子

- tapply()
  - > gender <- c('f','m','m','m','f')
  - > age <- c(12,35,25,12,25)
  - > tapply(age,gender,mean)

转换

判断数据类型

is.numeric()　是否数值型数据
is.integer () 是否整数型数据
is.double()　是否双精度数值
is.character()　是否字符型数据
is.vector()　是否向量数据
is.factor()　是否因子数据
is.logical()　是否逻辑型数据
is.na()　　是否是缺失值
。　。　。　。

转换数据类型

as.numeric()
as.integer ()
as.double()
as.character()
as.vector()
as.factor()
as.logical()
as.na()
。　。　。　。

矩阵

- 二维数据表
- 除了数学上的矩阵外，也可以由逻辑值，字符组成

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

- mat <- matrix(c(1:12),nrow=3,ncol=4)
- 维数获取：dim()

矩阵

- 矩阵与标量相乘：*
- 矩阵与矩阵相乘：%*%
- 满足矩阵相乘的条件
- t() 转置
- diag() 对角矩阵
- upper.tri() 上三角矩阵
- lower.tri() 下三角矩阵
- apply(mat,1,sum)　按行计算

  apply(mat,2,sum)　按列计算

数据框

- Data Frame一般被译为数据框，感觉就像是R中的表，由行和列组成，与Matrix不同的是，每个列可以是不同的数据类型，而Matrix是必须相同的。Data Frame每一列有列名，每一行也可以指定行名。如果不指定行名，那么就是从1开始自增的Sequence来标识每一行。所以说数据框在R语言中可是个好东西，R中它是用的非常频繁也是非常有用的数据集合。

- **应用最多的数据类型**

  - dat <- data.frame()

基本操作

- 访问：索引、$
- 添加新列
- 查询：查询满足条件的记录
- 矩阵=>数据框：as.data.frame()

高级操作

- 连接：merge()

- **合并(频繁使用)**：
- rbind()：列数相同
- cbind()：行数相同

- lapply()
- sapply()

列表

- 列表是一种复杂的数据结构，可以包含不同类型的数据。

  - C语言的结构体
  - Python的字典

- L <- list()
  - 列表索引：顺序、$
  - 绑定列表：attach()
  - 转化为向量：unlist()

IF语句

if (A) {B} 如果A成立，则B

if (A) { B } else {C} 如果A成立 则B 否则 C

if (A) { B } else if (C) { D }

ifelse(test, yes, no)

break

next

For循环语句

```
for(i in a)
    {..}
```

While循环语句

```
while (i < a)
    {..}
```

repeat循环语句

```
repeat {
 s <- s+i
 i <- i+1
 if(i > 100)
 break()
}
```

函数

- 函数是一系列语句的组合，在R中可以写出自己的函数
- 形式: 变量名 = function(变量列表) 函数体
- 函数引用: 变量名(变量的值)

```
factorial = function(n) {
    if (n>=0) gamma(n+1)
    }
```

- 用于处理错误的函数 – 用于处理用户输入不正确的类型而可能出现的错误
  - warning()　若错误不严重以至影响整个计算
  - stop()　若错误可能导致计算中止
  - print()　显示必要的信息
  - formatC()　数值作为字符串输出
  - cat()　字符串联，可以插入\n(换行)及\t(tab键)
  - paste()　字符粘贴(非字符型自动转换)

例子：

```
>cat("R", "is","a good", "software.\n")
>formatC(1/3, format = "f", digits = 4)
> formatC(1/3, format = "e", digits = 4)
>paste(1:12)  # 与as.character(1:12)等价
>paste("A", 1:6, sep = "")
>paste("today is", date())
```

查看函数

```
> solve
function (a, b, ...)
UseMethod("solve")
<bytecode: 0x0000017fbfc93410>
<environment: namespace:base>
>
```

```
> factorial
function (x)
gamma(x + 1)
<bytecode: 0x0000017fbaaa6750>
<environment: namespace:base>
```

输入输出

cat/print：显示对象

sink：输出转向到指定文件

dump，dput，write：输出对象

scan，read.table，dget：读入

输入输出

获取当前工作路径　Working Directory

```
> getwd()
[1] "C:/Users/lkf/Documents"
>
```

设置工作路径

```
setwd("E:/")
```

新建一个文件夹

```
> dir.create("E://Rclass")
Warning message:
In dir.create("E://Rclass") : 'E:\\Rclass'已存在
> dir.exists("Rclass")
[1] TRUE
```

#查询文件夹或者文件是否存在

load("……")　#读入R数据文件

#查询路径下的文件

```
> dir(path = "Rclass")
character(0)
> dir(path = "C://Program Files",pattern = "^W")
 [1] "Windows Defender"            "Windows Mail"
 [3] "Windows Media Player"        "Windows Multimedia Platform"
 [5] "Windows NT"                  "Windows Photo Viewer"
 [7] "Windows Portable Devices"    "Windows Security"
 [9] "Windows Sidebar"             "WindowsApps"
[11] "WindowsPowerShell"           "WinRAR"
```

#查询路径下的所有文件夹

list.dirs(path = ".", full.names = TRUE, recursive = TRUE)

#查询路径下的所有文件

```
list.files()
```

输入输出

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
           dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

sep=""，即表示分隔符可为一个或多个空格、制表符、换行符或回车符。

merge {base}

Merge Two Data Frames

Description

Merge two data frames by common columns or row names, or do other versions of database *join* operations.

Usage

```
merge(x, y, ...)

## Default S3 method:
merge(x, y, ...)

## S3 method for class 'data.frame'
merge(x, y, by = intersect(names(x), names(y)),
      by.x = by, by.y = by, all = FALSE, all.x = all, all.y = all,
      sort = TRUE, suffixes = c(".x",".y"), no.dups = TRUE,
      incomparables = NULL, ...)
```

```
x <- merge(x1,x2,by="Protein.IDs", sort = FALSE)

x <- merge(x1,x2,by="Protein.IDs" )
```

数据框与矩阵相关

cbind {base}

Combine R Objects by Rows or Columns

Description

Take a sequence of vector, matrix or data-frame arguments and combine by *c*olumns or *r*ows, respectively.

Usage

```
cbind(..., deparse.level = 1)
rbind(..., deparse.level = 1)
## S3 method for class 'data.frame'
rbind(..., deparse.level = 1, make.row.names = TRUE,
      stringsAsFactors = default.stringsAsFactors(), factor.exclude = TRUE)
```

```
pos <- match(x3$Protein.IDs, x4$Protein.IDs);
x <- cbind(x3, x4[pos,-1])
```

```r
x_nosort <- merge(x1, x2, by = "Protein.IDs")
x_sort <- merge(x1, x2, by = "Protein.IDs", sort = FALSE)

write.xlsx(
  list(x_nosort, x_sort),
  file = "result.xlsx",
  sheetName = c("x_nosort", "x_sort")
)
```

```
> wb <- createWorkbook()
> addWorksheet(wb, sheetName = "Sheet 1")
> addWorksheet(wb, sheetName = "Sheet 2")
> writeData(wb, sheet = 1, x_nosort)
> writeData(wb, sheet = 1, x_sort)
> saveWorkbook(wb, "xx.xlsx", overwrite = TRUE)
```

```
> write.xlsx(x_nosort,file="x.xlsx",sheetName="sheet1",append=TRUE)
> write.xlsx(x_sort,file="x.xlsx",sheetName="sheet2",append=TRUE)
```

apply

Description

Returns a vector or array or list of values obtained by applying a function to margins of an array or matrix.

Usage

```
apply(X, MARGIN, FUN, ...)
```

Apply a Function Over a Ragged Array

Description

Apply a function to each cell of a ragged array, that is to each (non-empty) group of values given by a unique combination of the levels of certain factors

Usage

```
tapply(X, INDEX, FUN = NULL, ..., default = NA, simplify = TRUE)
```

主要参考资料:

1.R语言的前世今生，陈凯，
https://www.cnblogs.com/chenkai/archive/2013/05/16/3082889.html
2. YuLabSMU公众号 余光创
https://mp.weixin.qq.com/s/BsEm76Eq9_tbw3myobD_SA
https://mp.weixin.qq.com/s/3MTTrAREsUVb56zHGufL2A
3. 【R语言入门】小白速成与实践,尚学堂,
https://www.bilibili.com/video/BV157411F7ZV
4.R, wiki, https://en.wikipedia.org/wiki/R_(programming_language)
5. 哈佛R语言课程--1.R简介,生信星球,
https://mp.weixin.qq.com/s/64WrT93bKv20rA-DhXLhPw
6. YuLabSMU 余光创
7. Think SAS, 胡江堂
https://cosx.org/2010/04/think-sas-1/
8. 统计学专业应该使用什么样的统计软件, 谢益辉,
https://cosx.org/2008/11/which-statistical-software-should-we-use/
9. 扫盲！SPSS、SAS、Stata、R有何区别，你该学哪个？,
https://cloud.tencent.com/developer/article/1043243