

易生信——最懂你的生信培训，学习生信更容易



R简介和统计绘图



易汉博

领先的大数据与健康解决方案
Leading solutions for big data and health



- 提前预习
- 仔细听讲
- 先运行再理解
- 紧跟步伐，跟不上的及时在课堂提出或寻找助教老师解决
- 课后复习，基础知识学习靠背和反复练
- 书读百变，其义自见
- 码敲十遍，不会也难



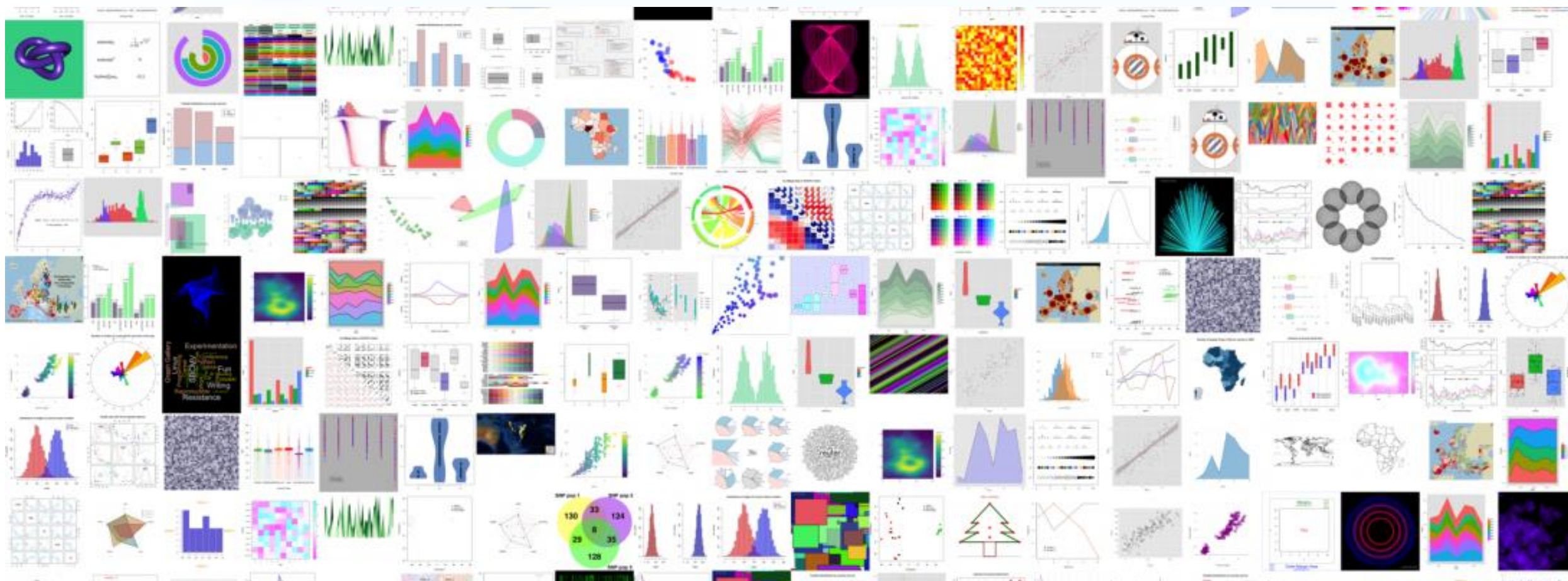
- 为什么要学习R
- R语言发展和学习资源
- R语言统计绘图实战
- R语言其它应用





- 为什么要学习R
- R语言发展和学习资源
- R语言统计绘图实战
- R语言其它应用

R画展 - 绘图神器



为什么选择R?



○ 多领域的资源

CRAN收录超过1.3万个包 (18年9月), 涵盖了统计学、经济学、生态学、进化生物学、生物信息学、物理、化学等方面。

○ 跨平台

R可在主流操作系统下运行, 包括Windows、MacOS、多种Linux。

○ 命令行驱动

R即时解释, 输入命令, 即可获得相应的结果, 方便绘图不断修改的需要。





- 统计分析能力突出，部分统计功能整合在R语言的底层，但大多数则以包(packages)的形式提供，资源极其丰富
- R具有强大的数据可视化能力，高质量的图像输出，多种现代绘图系统：如grid, lattice, ggplot2等
- 扩展和开发能力，可以编制自己的函数，或制作独立的统计分析包，快速实现新算法
- 灵活，便于与其他工具整合，实现流程化
- 自由、免费、源代码开放





- 学习曲线略陡

与代码打交道，需要记住常用命令。

- 占用内存

所有的数据需要全部读入内存才处理，不适于处理超大规模的数据。

- 运行速度稍慢

即时编译，约相当于C语言的1/20。



R语言谁创造的



- R 是一种有着强大统计分析和作图功能的软件系统（编程语言），受S语言和Scheme语言影响发展而来，最初是由新西兰奥克兰大学统计系的Ross Ihaka和Robert Gentleman两人合作编写，现由R开发核心小组负责维护，更新速度极快。截止目前R已更新至3.5.1版本。





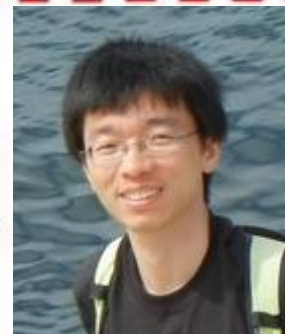
- 为什么要学习R
- **R语言发展和学习资源**
- R语言统计绘图实战
- R语言其它应用

R语言发展现状 – 2018 IEEE Spectrum排名





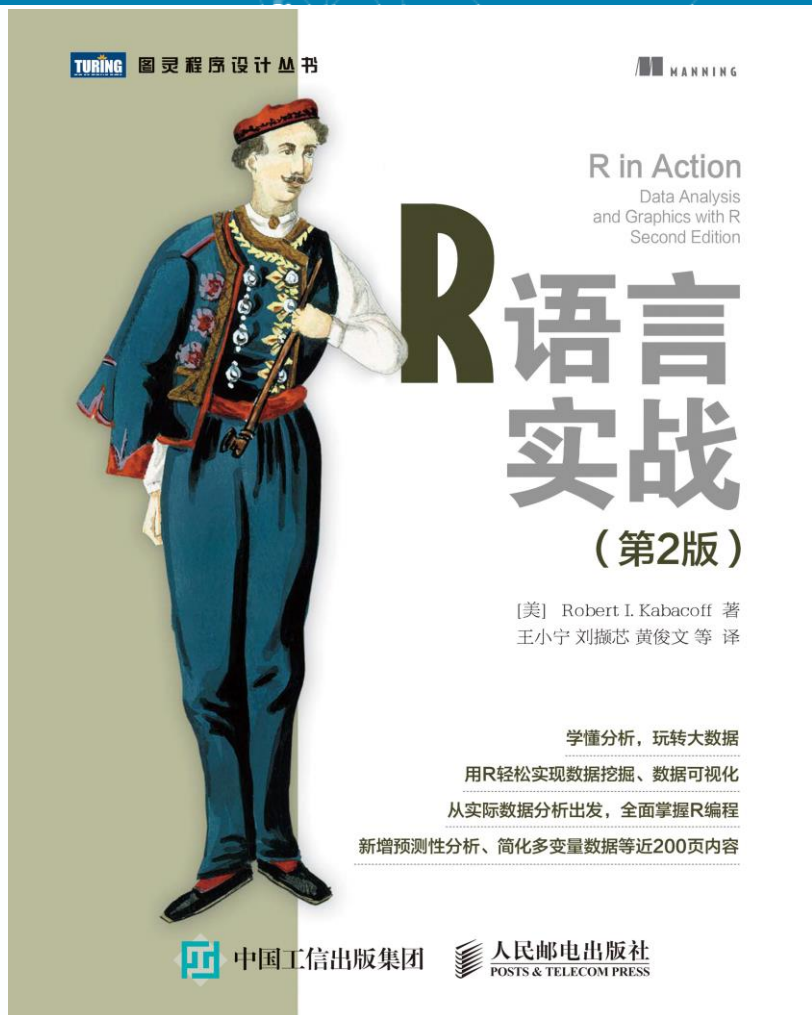
R语言中文社区（简称：R社区） 国内R语言细分领域最大的学习交流平台，提供R语言最新资讯、直播课程、实践分享及免费资料库等内容。2017全年发布推文600余篇，近200多位资深R爱好者参与分享，其中原创文章175篇（30%），同时文章图文页阅读人数、次数分别突破150万、260万。



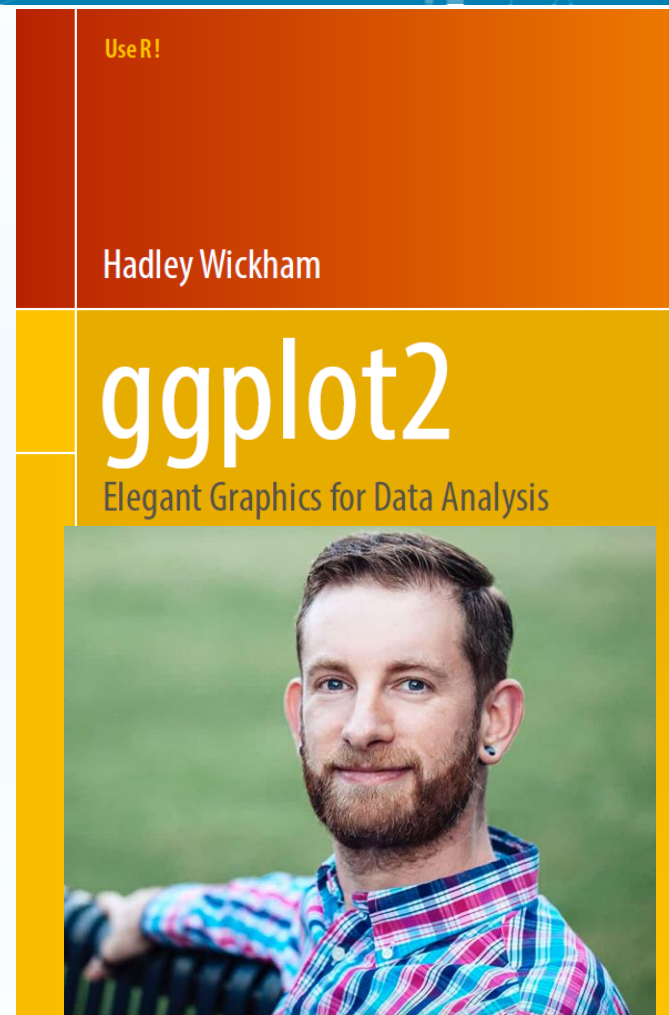
谢益辉
bookdown, animation,
knitr, formatR等众多R包
作者

统计之都 国内较早致力于推广与应用统计学知识的网站与社区（非盈利机构），是综合性的统计学服务平台，也是历届中国R会议的主要举办方，有力推动了R语言的应用实践。

R语言学习书籍



基础知识：1-7章，其它选学相关



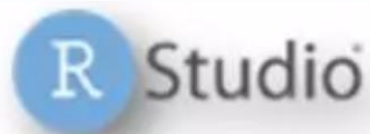
系统学习绘图艺术



易汉博

领先的大数据与健康解决方案
Leading solutions for big data and health

Wickham, Hadley. *ggplot2: elegant graphics for data analysis*. Springer, 2016.¹³



RStudio IDE是目前使用频次最高的R开发工具之一，其在便捷性、协同等工作场景下尤为优秀！以下为常用工具辅助推荐：

1.Shiny：支持开发交互式的web可视化应用

2.RStudio Server Pro：RStudio的企业化版本，支持web访问操作、团队协同等

3.RStudio Connect：团队协同工具，分享shiny应用、Markdown报告等

4.R Packages：RStudio Team开发了众多高效的R包，诸如rmarkdown、ggplot2、sparklyr等





- 为什么要学习R
- R语言发展和学习资源
- **R语言统计绘图实战**
- R语言其它应用

R软件下载安装



○ R语言 <https://www.r-project.org>

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

○ Rstudio编程环境 <https://www.rstudio.com>

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.423 - Windows Vista/7/8/10	85.8 MB	2018-02-07	a2411be84794b61fd8e79e70e7c0f0b0
RStudio 1.1.423 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-02-07	3e3e3db076b44f3c5276eb008614b4cf
RStudio 1.1.423 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-02-07	8515d8f5c78ac15b331bd9be0c1ea412
RStudio 1.1.423 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-02-07	f6e385c13ff7a1218891937f016e9383
RStudio 1.1.423 - Ubuntu 16.04+/Debian 9+ (64-bit)	65 MB	2018-02-07	1b5599d9f19c0971e87a5bcbf77aa8bc
RStudio 1.1.423 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-02-07	27664d49e08deee206879d259fd10512
RStudio 1.1.423 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-02-07	8d3d8c49260539a590d8eeea555eab08







- R包主要分布于CRAN, Bioconductor和Github三个网站上
- CRAN是R官方包发布网站, 安装方法如下: 以绘图包ggplot2为例

```
install.packages("ggplot2")
```

- 在Rstudio中默认使用rstudio数据源有时下载慢, 使用国内镜像

```
site="https://mirrors.tuna.tsinghua.edu.cn/CRAN"
```

```
install.packages("ggplot2", repos=site)
```

[R包安装视频讲解](#)

<https://mp.weixin.qq.com/s/XOZD4ftYLPZjM2IVLYgUpA>

```
> install.packages("ggplot2", repos=site)
Installing package into 'C:/Users/woodc/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
trying URL 'https://mirrors.tuna.tsinghua.edu.cn/CRAN/bin/windows/contrib/3.4/ggplot2_2.2.1.zip'
Content type 'application/zip' length 2784092 bytes (2.7 MB)
downloaded 2.7 MB

package 'ggplot2' successfully unpacked and MD5 sums checked
Warning in install.packages :
  unable to move temporary installation 'C:\Users\woodc\Documents\R\win-library\3.4\file104784f6f1059\ggplot2' to 'C:\Users\woodc\Documents\R\win-library\3.4\ggplot2'

The downloaded binary packages are in
  C:\Users\woodc\AppData\Local\Temp\Rtmpk12YB1\downloaded_packages
```





- 专门发布生信相关R包网站
- 此网站包安装方法



根据本地系统与R版本来选择软件库，安装加载基础包

```
source("https://bioconductor.org/biocLite.R")
```

包安装，以组间差异分析edgeR包为例

```
biocLite("edgeR")
```

```
> source("https://bioconductor.org/biocLite.R")
Bioconductor version 3.6 (BiocInstaller 1.28.0), ?biocLite for help
> # 包安装
> biocLite("edgeR")
BioC_mirror: https://bioconductor.org
Using Bioconductor 3.6 (BiocInstaller 1.28.0), R 3.4.3 (2017-11-30).
Installing package(s) 'edgeR'
also installing the dependency 'limma'

trying URL 'https://bioconductor.org/packages/3.6/bioc/bin/windows/contrib/3.4/limma_3.34.9
.zip'
Content type 'application/zip' length 4981399 bytes (4.8 MB)
```

开发者网站——Github



- 包的最新版本、未公开发布包
- 依赖Rstudio公司的devtools包实现安装

安装devtools包，仅需安装一次

```
install.packages("devtools")
```

加载devtools包

```
library(devtools)
```

安装ggpubr包

```
install_github("kassambara/ggpubr")
```

```
> install.packages("devtools")
Installing package into 'C:/Users/woodc/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
also installing the dependency 'git2r'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/git2r_0.21.0.zip'
Content type 'application/zip' length 3028640 bytes (2.9 MB)
library(devtools)
```

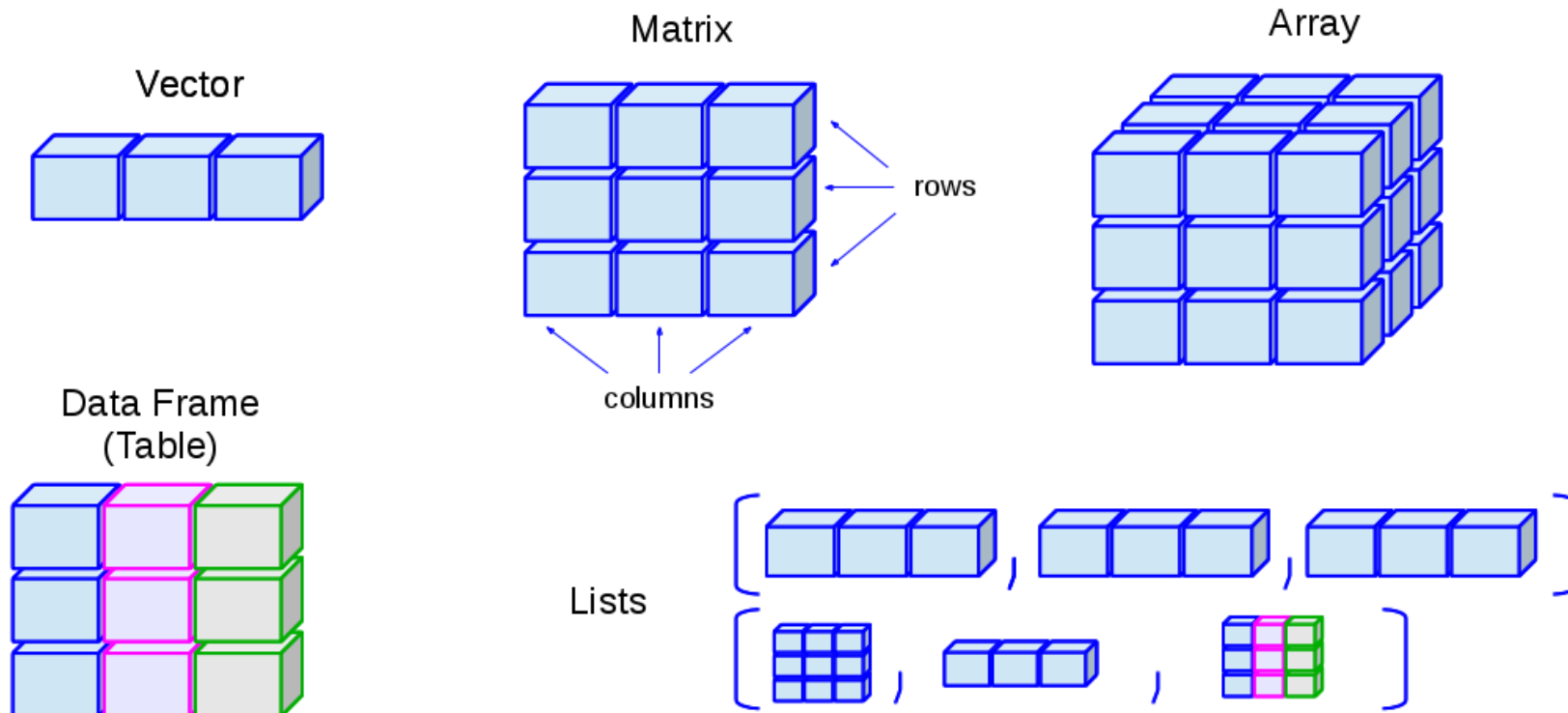
安装过程根据网速可能需要很久
安装失败按提示缺啥装啥，
或重启程序重试





- 数值型 (numeric): 1,2,3,4,5.1,6.6,1e5等 (绘图时一般作为连续变量对待)
- 字符型 (character): "ehbio", "e", 'h', 'yi sheng xin' (单个或多个字母或其他符号的组合, 需要用引号括起)
- 逻辑型 (logical): TRUE, FALSE (布尔值)
- 缺失值: NA, NaN, Inf
- 空值: NULL

R语言数据结构及其索引方式



Indexing vectors

<code>x[n]</code>	nth element
<code>x[-n]</code>	all but the nth element
<code>x[1:n]</code>	first n elements
<code>x[-(1:n)]</code>	elements from n+1 to end
<code>x[c(1,4,2)]</code>	specific elements
<code>x["name"]</code>	element named "name"
<code>x[x > 3]</code>	all elements greater than 3
<code>x[x > 3 & x < 5]</code>	all elements between 3 and 5
<code>x[x %in% c("a","if")]</code>	elements in the given set

Indexing lists

<code>x[n]</code>	list with elements n
<code>x[[n]]</code>	nth element of the list
<code>x[["name"]]</code>	element named "name"
<code>x\$name</code>	as above (w. partial matching)

Indexing matrices

<code>x[i,j]</code>	element at row i, column j
<code>x[i,]</code>	row i
<code>x[,j]</code>	column j
<code>x[,c(1,3)]</code>	columns 1 and 3
<code>x["name",]</code>	row named "name"

Indexing matrices data frames (same as matrices plus the following)

<code>X[["name"]]</code>	column named "name"
<code>x\$name</code>	as above (w. partial matching)





- 向量, vector, 存储相同的数据类型。如果有字符型元素, 所有元素都转换为字符型。
- is.atomic() 判断是否为向量。
- is.vector() 只能判断没有属性的向量或仅有names属性的向量。

```
> a <- c(2,5,9)
> a
[1] 2 5 9
> b <- c('e','h',"Bio")
> b
[1] "e"    "h"    "Bio"
> d <- c(TRUE, FALSE)
> d
[1] TRUE FALSE
> e <- c(1,2,'yishengxin')
> e
[1] "1"          "2"          "yishengxin"
```





- 因子，factor，节省存储空间的特殊类型，适用于重复值特别多时。每个原值只存储一份，在原数据中用数字表示。
- 去重后的原值被称为水平(level)，可以修改顺序。绘图时调整顺序就是使用的这个。

```
> a <- c('y','i','s','h','e','n','g','x','i','n')
> as.factor(a)
[1] y i s h e n g x i n
Levels: e g h i n s x y
>
> b <- factor(a,
+             levels=c("y","x","s","e","g","i","h","n"),
+             ordered=T)
> b
[1] y i s h e n g x i n
Levels: y < x < s < e < g < i < h < n
```

R数据类型 – 矩阵



- 矩阵, matrix, 二维的array, 所有元素都是相同类型。可以按照[row, col]索引。

- 矩阵相关函数

dim: 获取矩阵的维数

rowSums, colSums 获取行列加和

max, min 获取最大, 最小值

```
> a <- 1:20
> matrix(a, nrow=4)
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
>
> b = matrix(a, nrow=4, byrow=T)
> b
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
[3,]   11   12   13   14   15
[4,]   16   17   18   19   20
> b[1,2]
[1] 2
> b[1,]
[1] 1 2 3 4 5
```



R数据类型 – 列表



- 列表, list, 不同类型的变量组合在一起, 常用于存储各种复杂类型函数或绘图的输出。

```
> ysx_list <- list(web="www.ehbio.com/Training",
+                 staff=c("LYX","CL","ZX","CT"),
+                 contentL=list(R=c("basic","plot"),
+                               Linux=c("basic","windows")))
> ysx_list
$web
[1] "www.ehbio.com/Training"

$staff
[1] "LYX" "CL"  "ZX"  "CT"

$contentL
$contentL$R
[1] "basic" "plot"

$contentL$Linux
[1] "basic"  "windows"
```



R数据类型 – 列表查看



- `str(list_name)` 可以查看列表的结构，方便索引获得对应的元素。
- `list_name$list_subname` 提取对应子元素。
- 注意单方括号和双方括号的区别，一个取出子列表，一个取出子列表的元素。

```
> str(ysx_list)
List of 3
 $ web      : chr "www.ehbio.com/Training"
 $ staff    : chr [1:4] "LYX" "CL" "ZX" "CT"
 $ contentL:List of 2
  ..$ R      : chr [1:2] "basic" "plot"
  ..$ Linux: chr [1:2] "basic" "windows"
> ysx_list$web
[1] "www.ehbio.com/Training"
> ysx_list$contentL$R
[1] "basic" "plot"
> ysx_list[1]
$web
[1] "www.ehbio.com/Training"

> ysx_list[[1]]
[1] "www.ehbio.com/Training"
```





- 数据框，dataframe，子列表长度一致的特殊list，具有list和matrix的双重特性，通常与matrix可互换。
- 若在程序运行时碰到“x must be matrix”，需要做下as.matrix转换；若是“x must be data.frame”，需要做下as.data.frame转换。这里的x是你提供的数据变量。

```
> a <- data.frame(staff=c("LYX","CL","ZX","CT"),
+                 work=c("IGDB","IM","IM","EHBIO"))
> a
  staff work
1  LYX IGDB
2   CL  IM
3  ZX  IM
4  CT EHBIO
> a$work
[1] IGDB  IM    IM    EHBIO
Levels: EHBIO IGDB IM
```





- storage.mode 查看变量在内存中存储的方式
- class 查看对象所属的类
- mode 查看对象的类型
- typeof 查看对象的类型
- str 查看数据结构
- summary 查看数据内容总结

```
> mode(a)
[1] "list"
> class(a)
[1] "data.frame"
> typeof(a)
[1] "list"
> storage.mode(a)
[1] "list"
> mode(ysx_list)
[1] "list"
> class(ysx_list)
[1] "list"
> class(ysx_list$web)
[1] "character"
> storage.mode(class(ysx_list$web))
[1] "character"
> class(ysx_list$staff)
[1] "character"
> storage.mode(ysx_list$staff)
[1] "character"
> typeof(2)
[1] "double"
> typeof(as.integer(2))
[1] "integer"
```





- 读入数据，存储为data frame格式

```
read.table(filename, header=T, row.names=1, sep="\t")
```

- 写数据框或矩阵到磁盘

```
write.table(a, file="ysx.test", quote=F, sep="\t")
```

简单的统计：T-test两组比较



- `rnorm(10)`产生10个均值为0，标准差为1的正态分布随机数。
- `t.test` 比较两组数直接是否存在显著差异。

```
> t.test(rnorm(10),rnorm(10))
```

Welch Two Sample t-test

```
data:  rnorm(10) and rnorm(10)
t = 1.1048, df = 17.928, p-value = 0.2839
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.5260499  1.6922100
sample estimates:
 mean of x   mean of y 
0.52334419 -0.05973586
```

```
> t.test(rnorm(10),rnorm(10, mean=5))
```

Welch Two Sample t-test

```
data:  rnorm(10) and rnorm(10, mean = 5)
t = -13.742, df = 17.32, p-value = 9.504e-11
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6.630154 -4.867445
sample estimates:
 mean of x   mean of y 
-0.6582895  5.0905099
```



简单的统计：anova多组比较



`data(iris) # 加载R内容测试数据——鸢尾花数据`

`head(iris) # 显示数据格式`

`# anova统计不同物种花萼长宽数据`

`model = aov(Sepal.Length ~ Species, data=iris)`

`Tukey_HSD <- TukeyHSD(model, ordered = TRUE, conf.level = 0.95) # TukeyHSD检验`

`# 显示统计结果`

`Tukey_HSD$Species`

```
> data(iris) # 加载R内容测试数据——鸢尾花数据
> head(iris) # 显示数据格式
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4          0.2   setosa
2          4.9         3.0          1.4          0.2   setosa
3          4.7         3.2          1.3          0.2   setosa
4          4.6         3.1          1.5          0.2   setosa
5          5.0         3.6          1.4          0.2   setosa
6          5.4         3.9          1.7          0.4   setosa
> # anova统计不同物种花瓣宽度数据
> model = aov(Sepal.Length ~ Species, data=iris)
> Tukey_HSD <- TukeyHSD(model, ordered = TRUE, conf.level = 0.95)
> # 显示统计结果
> Tukey_HSD$Species
```

	diff	lwr	upr	p adj
versicolor-setosa	0.930	0.6862273	1.1737727	3.386180e-14
virginica-setosa	1.582	1.3382273	1.8257727	2.997602e-15
virginica-versicolor	0.652	0.4082273	0.8957727	8.287558e-09



简单的绘图：箱线图展示组间差异

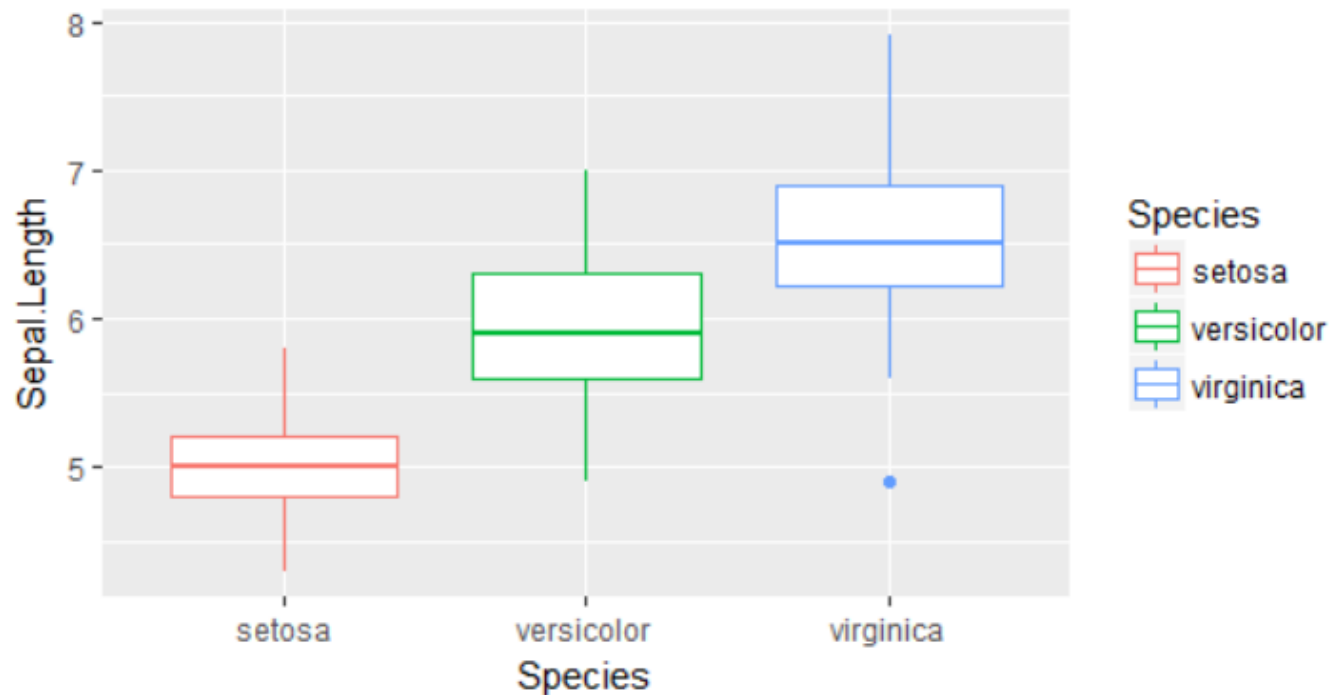


```
library(ggplot2) # 加载包
```

```
# ggplot命令绘图，指定数据、x、y、颜色、图表类型
```

```
ggplot(iris, aes(x=Species, y=Sepal.Length, color=Species)) + geom_boxplot()
```

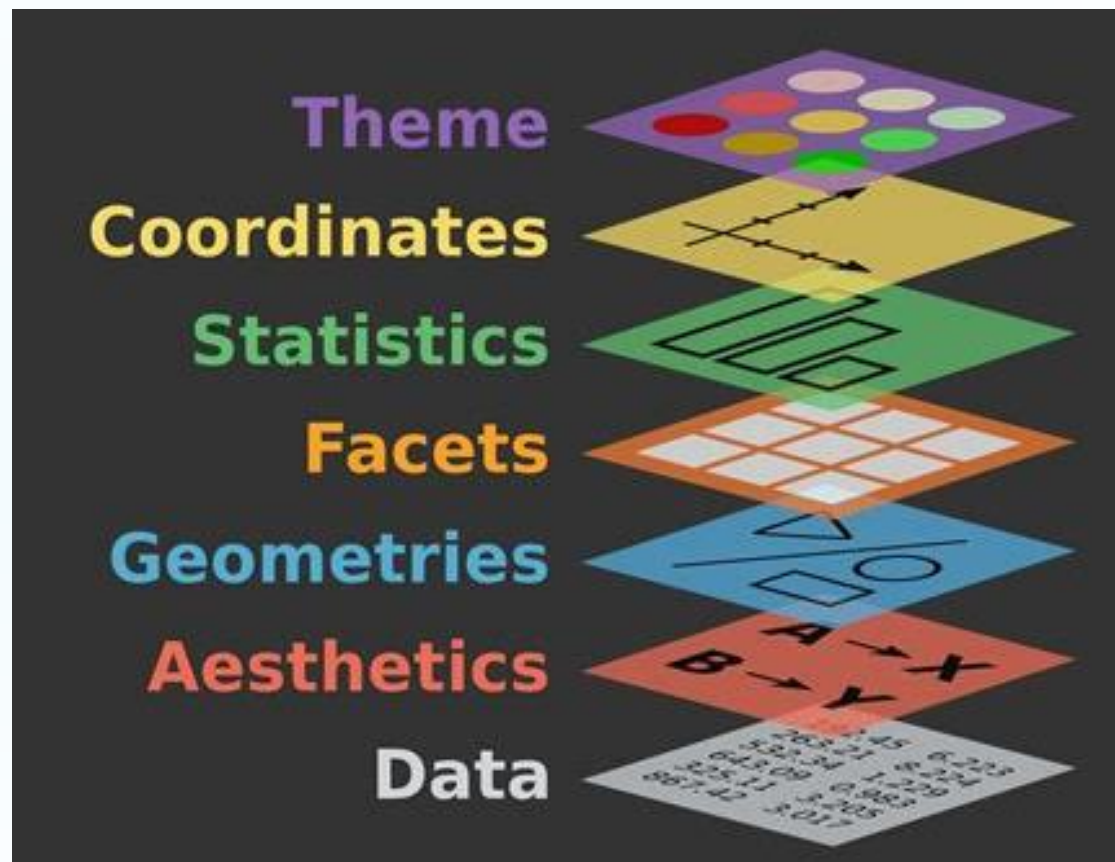
```
> library(ggplot2) # 加载包  
> # ggplot命令绘图，指定数据、x、y、颜色、图表类型  
> ggplot(iris, aes(x=Species, y=Sepal.Length,  
  color=Species)) + geom_boxplot()  
> |
```



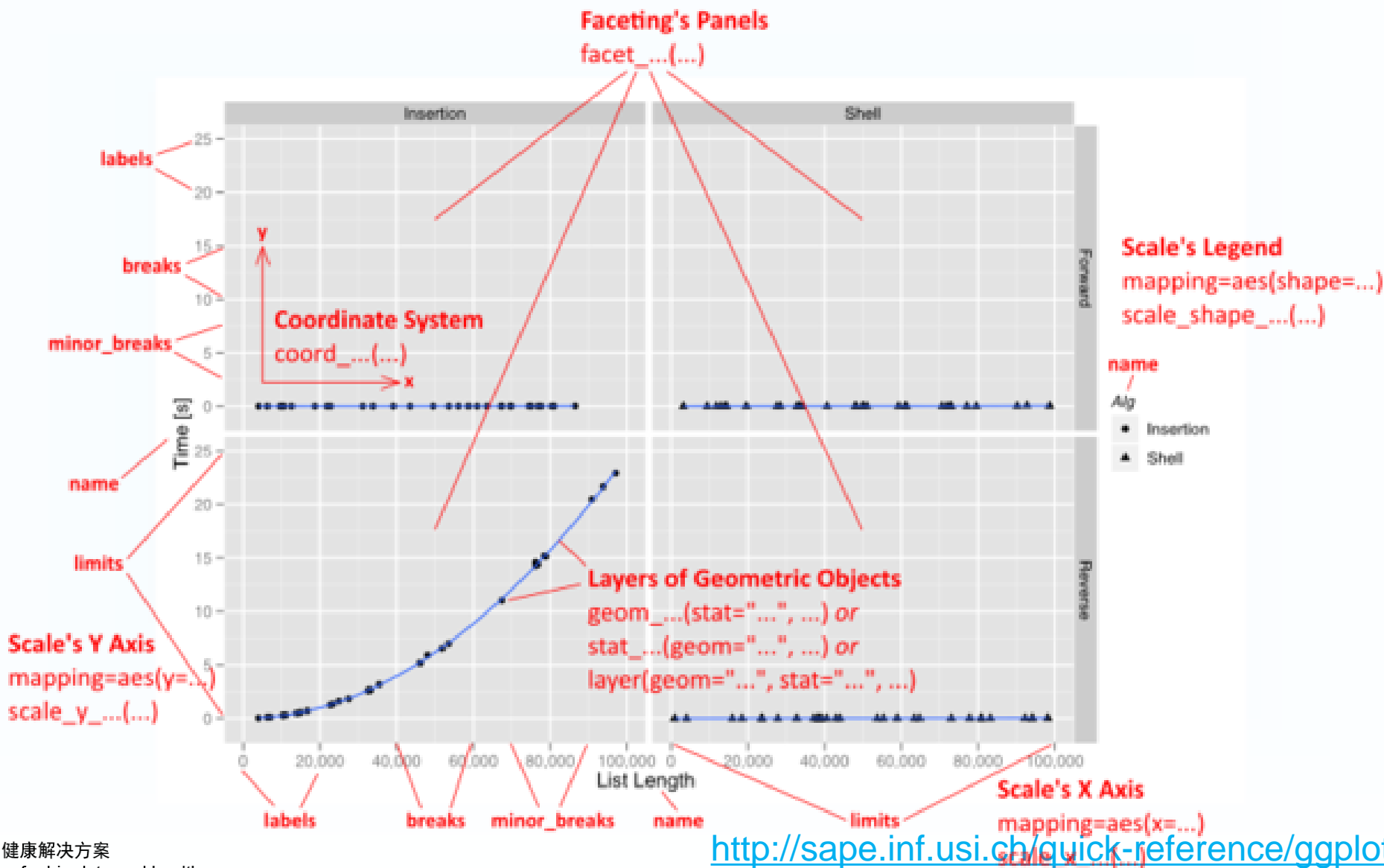
ggplot2的基本概念



- 数据 (Data)
- 映射 (Aesthetics)
- 几何对象 (Geometries)
- 分面 (Facets)
- 统计 (Statistics)
- 坐标系 (Coordinates)
- 主题 (Themes)



图形各部分元素解释



ggplot2 用图形及其属性展示数据的特征



- 统计图就是用把数据映射到几何形状如点、线、柱的美学属性如颜色、大小、形状上
- A statistical graphic is a mapping from **data** to **aesthetic attributes** (colour, shape, size) of **geometric objects** (points, lines, bars)
- 从数据信息到像素和颜色的转换过程，在ggplot2中，被称为“标度变换” (scaling)



ggplot2图绘图前准备 – 读入数据



- # 读入数据
- `data <- read.table(file, header=T, row.names=NULL, sep=";")`
- # 随机取出6条数据, 看下格式
- `data[sample(1:nrow(data),6),]`

OTU	Sample	Group	Abundance
OTU3	WT2	WT	30
OTU1	WT1	WT	2
OTU4	WT2	WT	25
OTU5	WT3	WT	10
OTU3	K03	K0	16
OTU3	K01	K0	16

文件内容

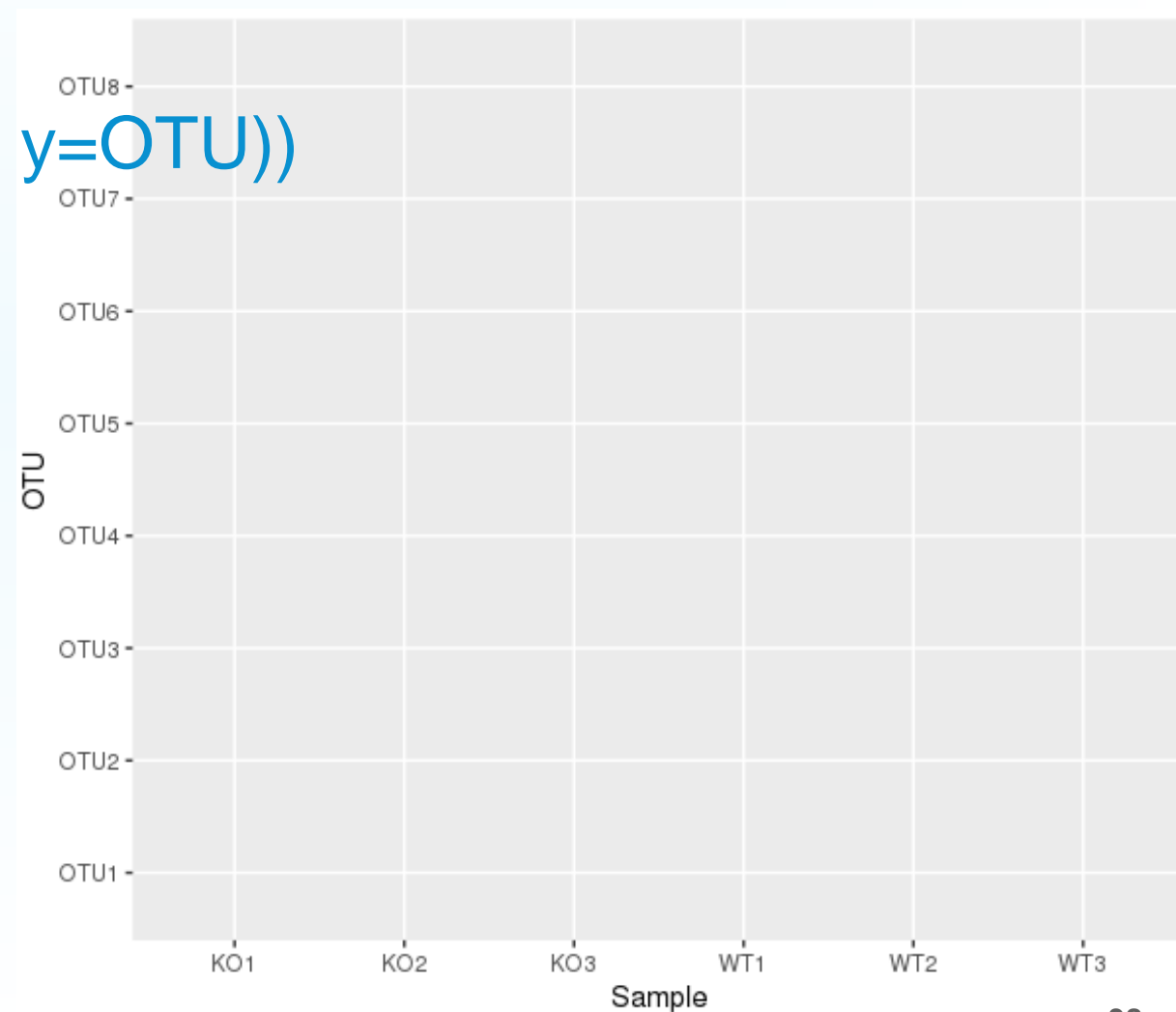
```
OTU;Sample;Group;Abundance
OTU1;K01;K0;1.2
OTU2;K01;K0;1.4
OTU3;K01;K0;16
OTU4;K01;K0;10
OTU5;K01;K0;25
OTU6;K01;K0;30
OTU7;K01;K0;2
OTU8;K01;K0;3
OTU1;K02;K0;1.2
OTU2;K02;K0;1.4
```



ggplot2绘图 – 初始化



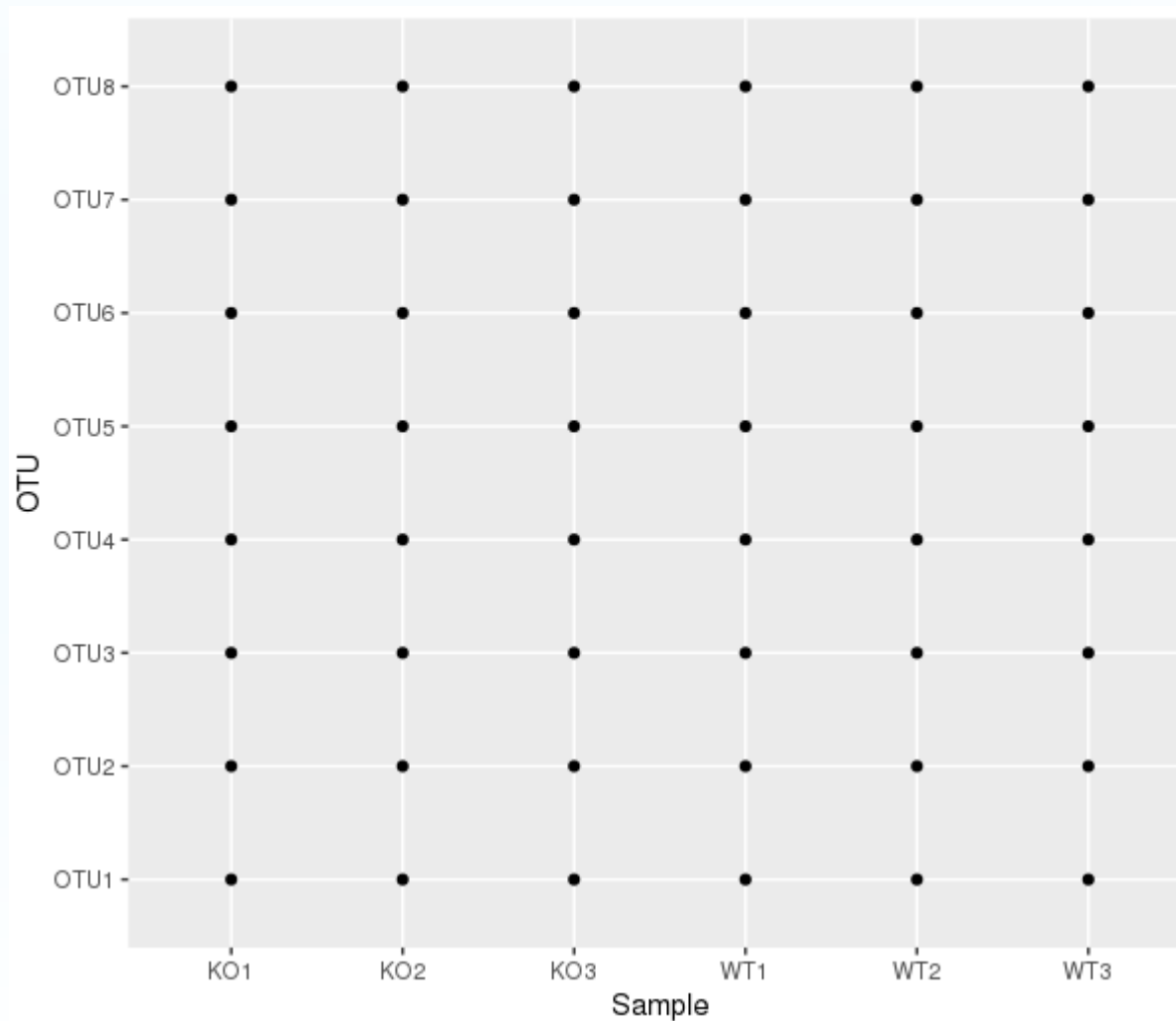
- `library(ggplot2)`
- `p <- ggplot(data, aes(x=Sample, y=OTU))`
- `p`



ggplot2绘图 – 指定几何对象



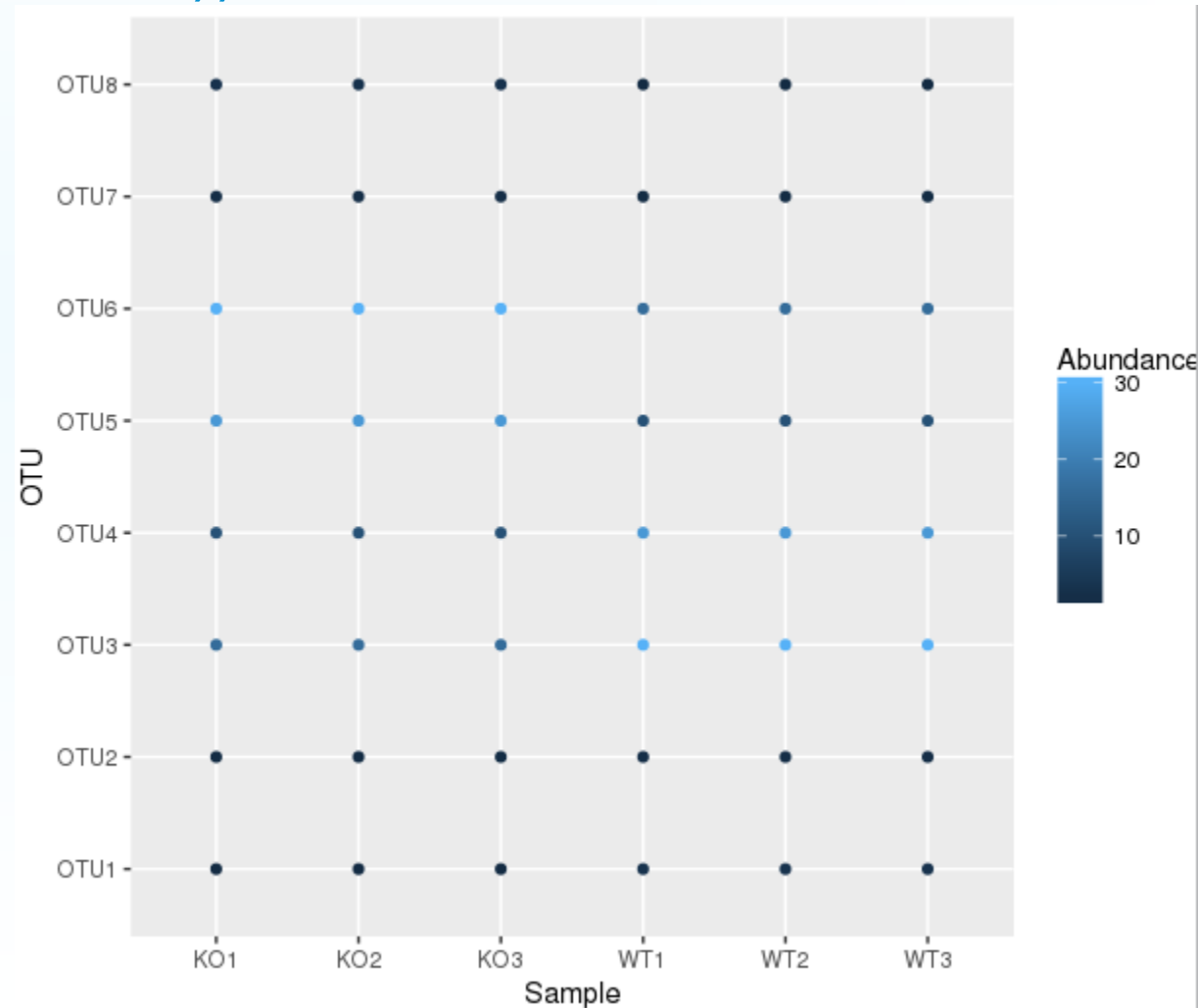
- `p + geom_point()`



ggplot2绘图 – 映射数据到几何对象的属性



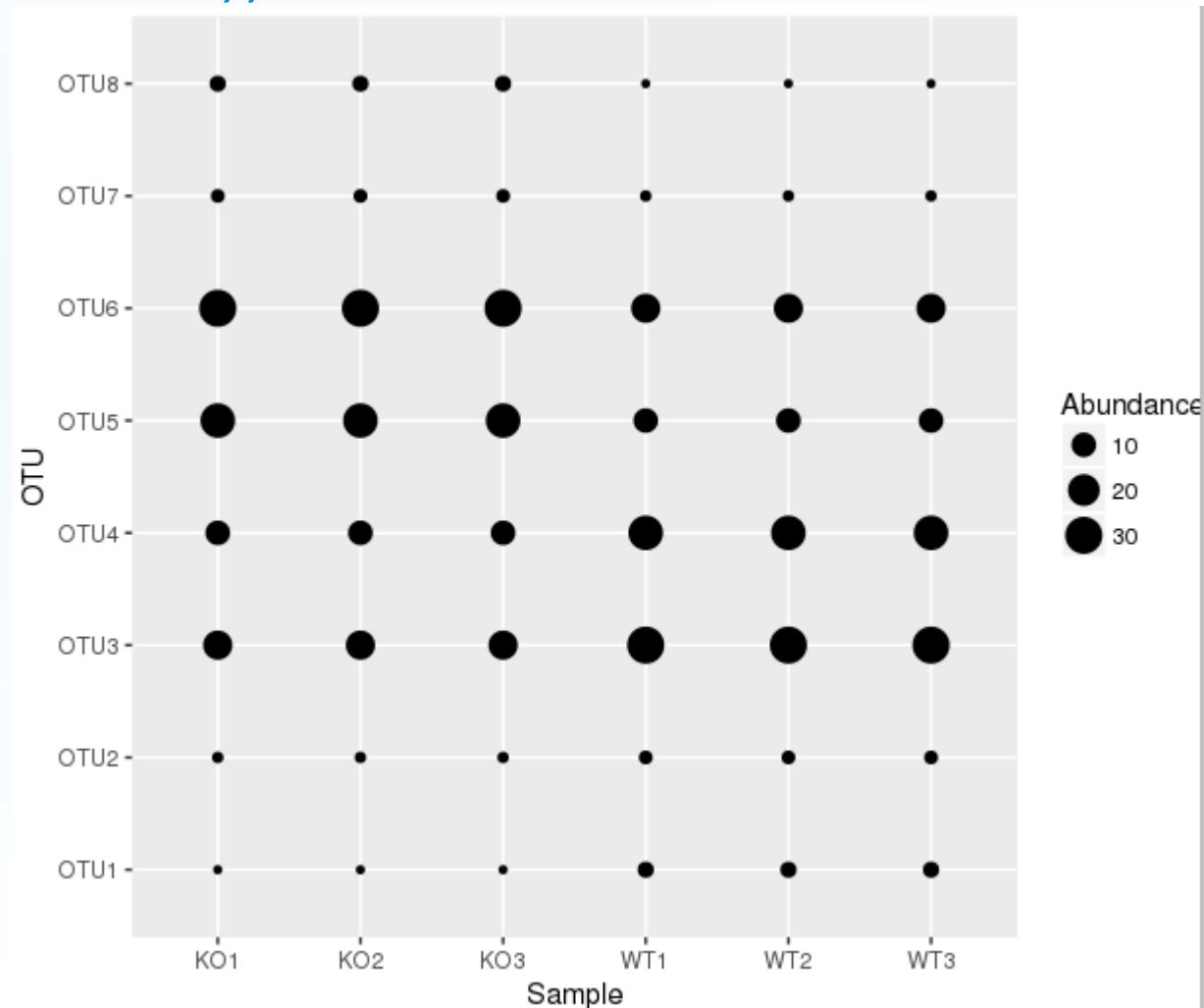
- o `p + geom_point(aes(color=Abundance))`



ggplot2绘图 – 映射数据到几何对象的属性



- o `p + geom_point(aes(size=Abundance))`



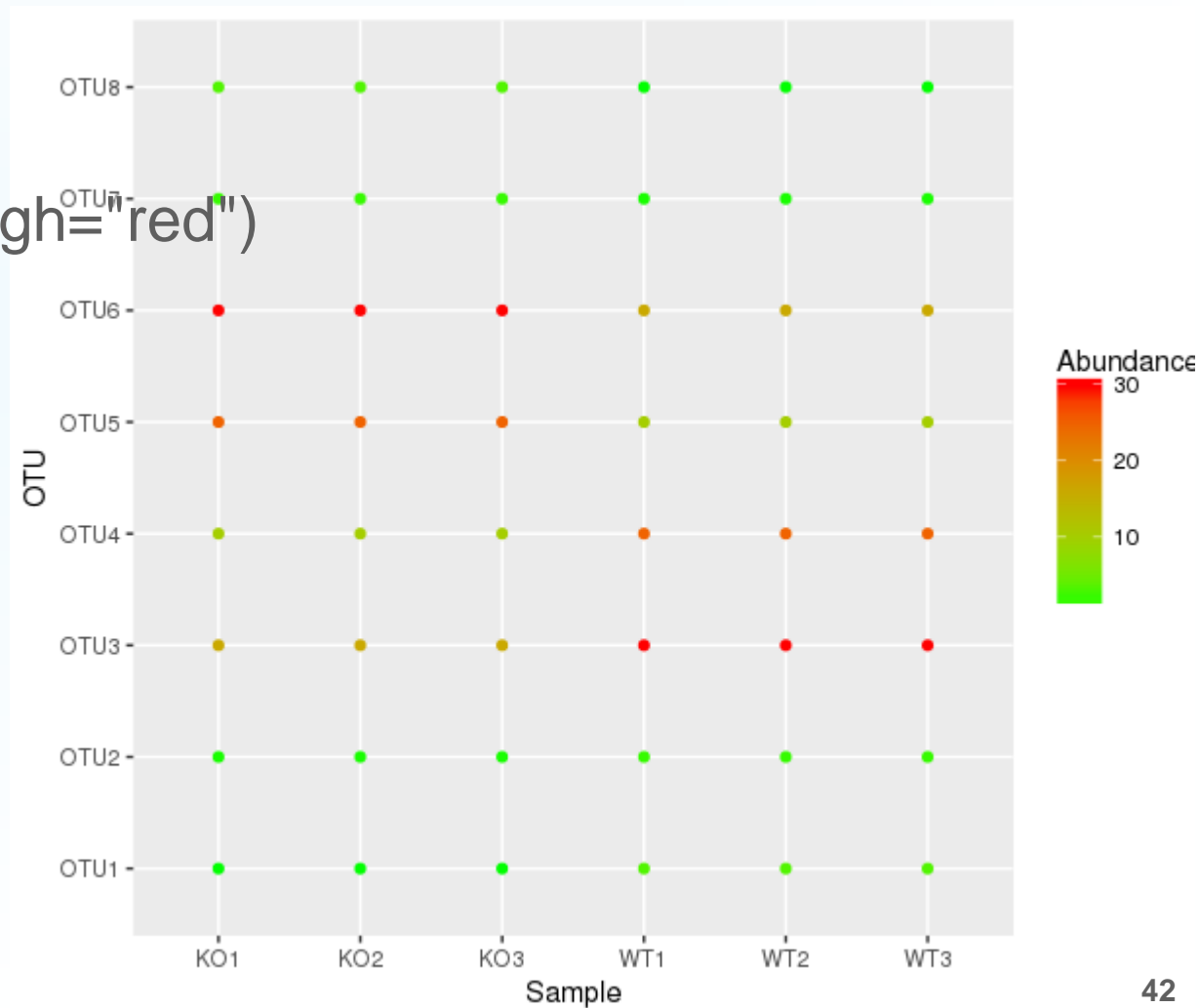
ggplot2绘图 – 调整颜色属性



p +

```
geom_point(aes(color=Abundance)) +
```

```
scale_color_continuous(low="green", high="red")
```



ggplot2绘图 – 更换主题

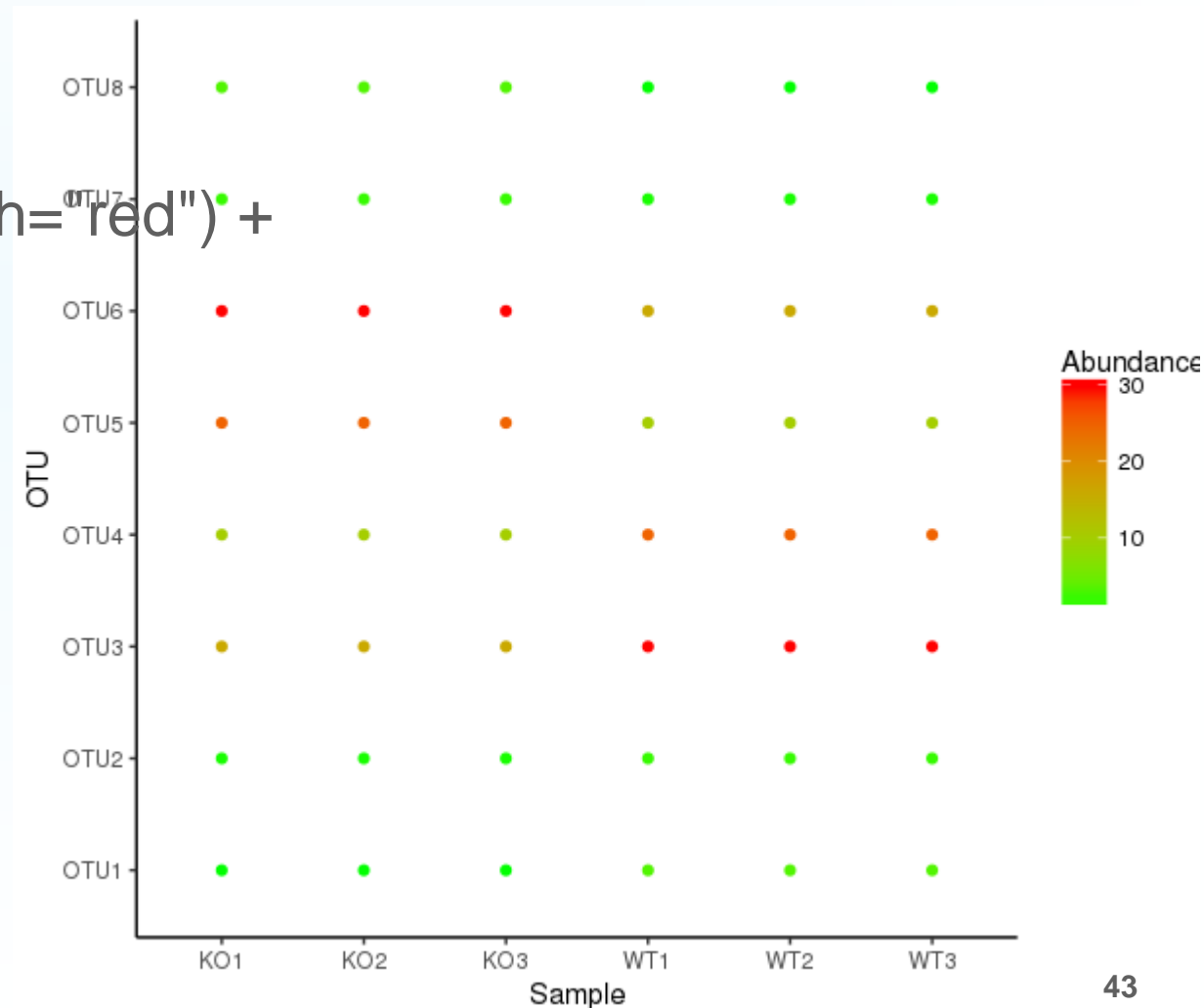


p +

```
geom_point(aes(color=Abundance)) +
```

```
scale_color_continuous(low="green", high="red") +
```

```
theme_classic()
```

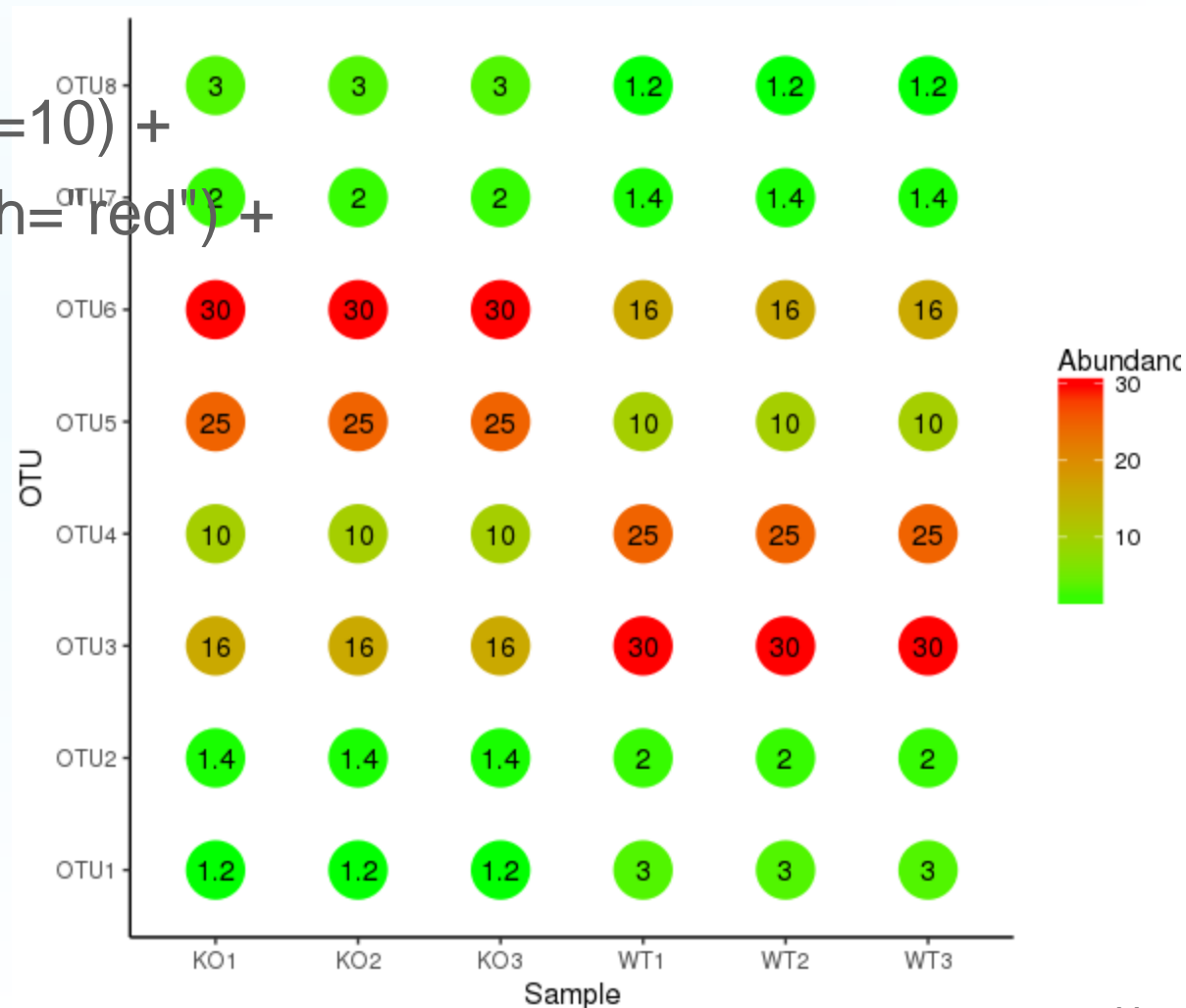


ggplot2绘图 – 放大点和填充文字



p +

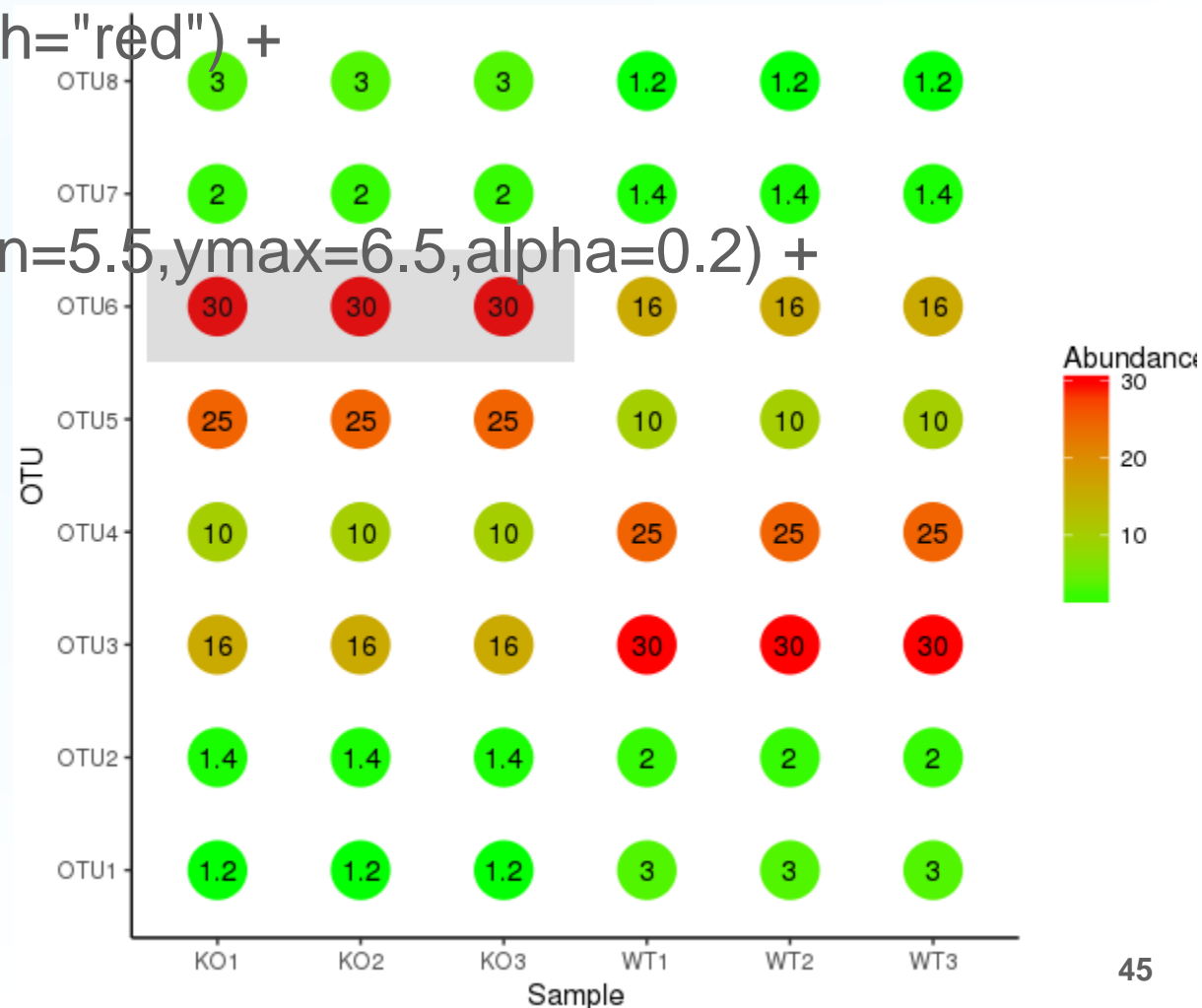
```
geom_point(aes(color=Abundance), size=10) +  
scale_color_continuous(low="green", high="red") +  
geom_text(aes(label=Abundance)) +  
theme_classic()
```



ggplot2绘图 – 增加注释



```
p + geom_point(aes(color=Abundance), size=10) +  
scale_color_continuous(low="green", high="red") +  
geom_text(aes(label=Abundance)) +  
annotate("rect",xmin=0.5, xmax=3.5, ymin=5.5,ymax=6.5,alpha=0.2) +  
theme_classic()
```



ggplot2绘图 – 调整横轴的顺序

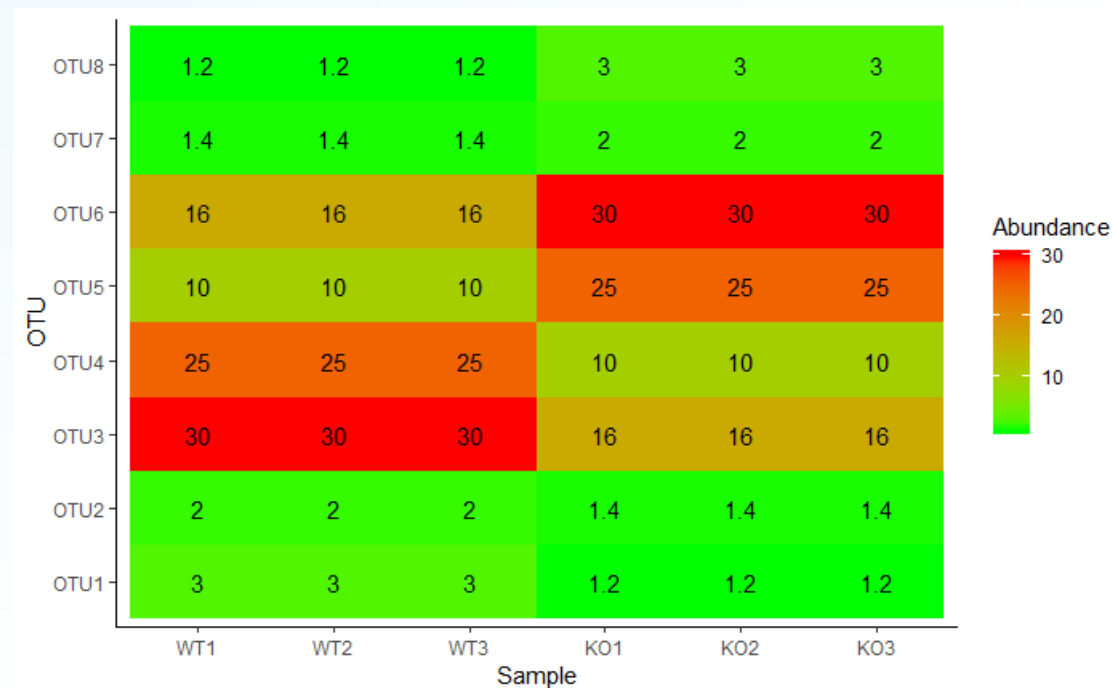


```
# paste0(rep(c("WT","KO"),each=3), rep(1:3))
```

```
data$Sample <- factor(data$Sample,levels=c("WT1","WT2","WT3","KO1","KO2","KO3"),  
ordered=T)
```

```
p <- ggplot(data, aes(x=Sample, y=OTU)) +  
  geom_tile(aes(fill=Abundance)) +  
  scale_fill_continuous(low="green", high="red") +  
  geom_text(aes(label=Abundance)) +  
  theme_classic()
```

p

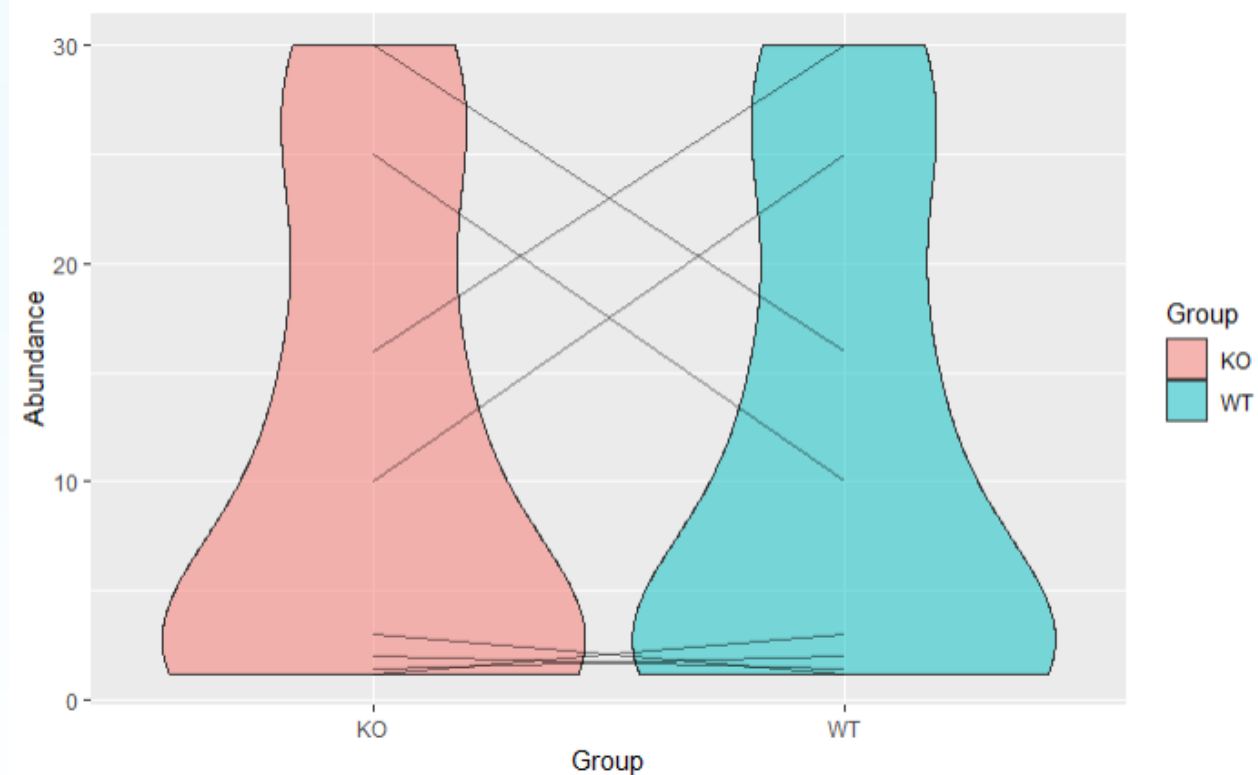


ggplot2绘制小提琴图 – 增加连线



```
p <- ggplot(data, aes(x=Group, y=Abundance)) +  
  geom_violin(aes(fill=Group), alpha=0.5) +  
  geom_line(aes(group=OTU), alpha=0.5)
```

p



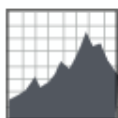
ggplot2 cheatsheet – 图形类型和变量对应



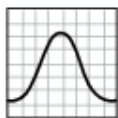
One Variable

Continuous

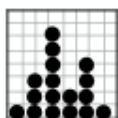
```
a <- ggplot(mpg, aes(hwy))
```



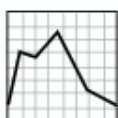
a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")



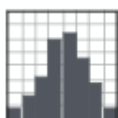
a + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))



a + geom_dotplot()
x, y, alpha, color, fill



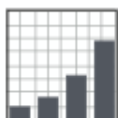
a + geom_freqpoly()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))



a + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

```
b <- ggplot(mpg, aes(fl))
```



b + geom_bar()
x, alpha, color, fill, linetype, size, weight

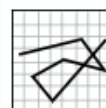
Graphical Primitives

```
map <- map_data("state")  
c <- ggplot(map, aes(long, lat))
```

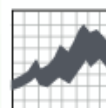


c + geom_polygon(aes(group = group))
x, y, alpha, color, fill, linetype, size

```
d <- ggplot(economics, aes(date, unemploy))
```

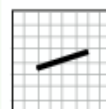


**d + geom_path(lineend = "butt",
linejoin = "round", linemitre = 1)**
x, y, alpha, color, linetype, size

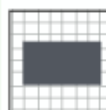


**d + geom_ribbon(aes(ymin = unemploy - 900,
ymax = unemploy + 900))**
x, ymax, ymin, alpha, color, fill, linetype, size

```
e <- ggplot(seals, aes(x = long, y = lat))
```



**e + geom_segment(aes(xend = long + delta_long,
yend = lat + delta_lat))**
x, xend, y, yend, alpha, color, linetype, size



**e + geom_rect(aes(xmin = long, ymin = lat,
xmax = long + delta_long,
ymax = lat + delta_lat))**
xmax, xmin, ymax, ymin, alpha, color, fill,
linetype, size



ggplot2 cheatsheet – 图形类型和变量类型



Two Variables

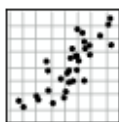
Continuous X, Continuous Y

```
f <- ggplot(mpg, aes(cty, hwy))
```



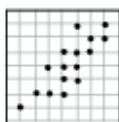
f + geom_blank()

(Useful for expanding limits)



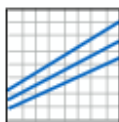
f + geom_jitter()

x, y, alpha, color, fill, shape, size



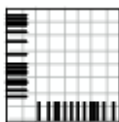
f + geom_point()

x, y, alpha, color, fill, shape, size



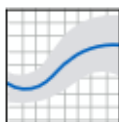
f + geom_quantile()

x, y, alpha, color, linetype, size, weight



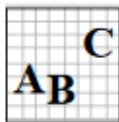
f + geom_rug(sides = "bl")

alpha, color, linetype, size



f + geom_smooth(method = lm)

x, y, alpha, color, fill, linetype, size, weight



f + geom_text(aes(label = cty))

x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

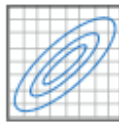
Continuous Bivariate Distribution

```
i <- ggplot(movies, aes(year, rating))
```



i + geom_bin2d(binwidth = c(5, 0.5))

xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight



i + geom_density2d()

x, y, alpha, colour, linetype, size

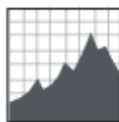


i + geom_hex()

x, y, alpha, colour, fill size

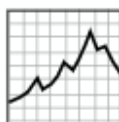
Continuous Function

```
j <- ggplot(economics, aes(date, unemploy))
```



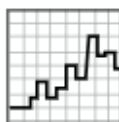
j + geom_area()

x, y, alpha, color, fill, linetype, size



j + geom_line()

x, y, alpha, color, linetype, size



j + geom_step(direction = "hv")

x, y, alpha, color, linetype, size

Visualizing error

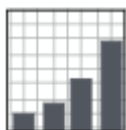


ggplot2 cheatsheet – 图形类型和变量类型



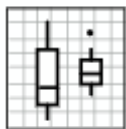
Discrete X, Continuous Y

```
g <- ggplot(mpg, aes(class, hwy))
```



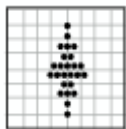
g + geom_bar(stat = "identity")

x, y, alpha, color, fill, linetype, size, weight



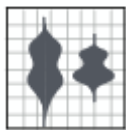
g + geom_boxplot()

lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight



g + geom_dotplot(binaxis = "y", stackdir = "center")

x, y, alpha, color, fill

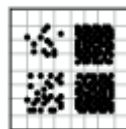


g + geom_violin(scale = "area")

x, y, alpha, color, fill, linetype, size, weight

Discrete X, Discrete Y

```
h <- ggplot(diamonds, aes(cut, color))
```



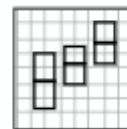
h + geom_jitter()

x, y, alpha, color, fill, shape, size

Visualizing error

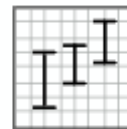
```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
```

```
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
```



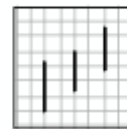
k + geom_crossbar(fatten = 2)

x, y, ymax, ymin, alpha, color, fill, linetype, size



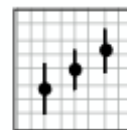
k + geom_errorbar()

x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)



k + geom_linerange()

x, ymin, ymax, alpha, color, linetype, size



k + geom_pointrange()

x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

Maps

```
data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
```

```
map <- map_data("state")
```

```
l <- ggplot(data, aes(fill = murder))
```



l + geom_map(aes(map_id = state), map = map) +

expand_limits(x = map\$long, y = map\$lat)

map_id, alpha, color, fill, linetype, size

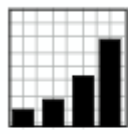


ggplot2 cheatsheet – 坐标系统和属性



Coordinate Systems

`r <- b + geom_bar()`



`r + coord_cartesian(xlim = c(0, 5))`

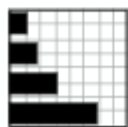
xlim, ylim

The default cartesian coordinate system

`r + coord_fixed(ratio = 1/2)`

ratio, xlim, ylim

Cartesian coordinates with fixed aspect ratio between x and y units



`r + coord_flip()`

xlim, ylim

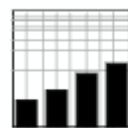
Flipped Cartesian coordinates



`r + coord_polar(theta = "x", direction=1)`

theta, start, direction

Polar coordinates



`r + coord_trans(ytrans = "sqrt")`

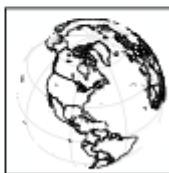
xtrans, ytrans, limx, limy

Transformed cartesian coordinates. Set xtrans and ytrans to the name of a window function.

`z + coord_map(projection = "ortho", orientation=c(41, -74, 0))`

projection, orientation, xlim, ylim

Map projections from the mapproj package (mercator (default), azequalarea, lagrange, etc.)



Color and fill scales

Discrete



`n <- b + geom_bar(aes(fill = fl))`

`n + scale_fill_brewer(palette = "Blues")`

For palette choices:
library(RColorBrewer)
display.brewer.all()



`n + scale_fill_grey(start = 0.2, end = 0.8, na.value = "red")`

Continuous



`o <- a + geom_dotplot(aes(fill = ..x..))`

`o + scale_fill_gradient(low = "red", high = "yellow")`

`o + scale_fill_gradient2(low = "red", high = "blue", mid = "white", midpoint = 25)`



`o + scale_fill_gradientn(colours = terrain.colors(6))`
Also: rainbow(), heat.colors(), topo.colors(), cm.colors(), RColorBrewer::brewer.pal()

Shape scales

Manual shape values



`p <- f + geom_point(aes(shape = fl))`



`p + scale_shape(solid = FALSE)`



`p + scale_shape_manual(values = c(3:7))`
Shape values shown in chart on right

0	□	6	▽	12	⊞	18	◆	24	▲
1	○	7	⊠	13	⊗	19	●	25	▼
2	△	8	✱	14	⊡	20	●	26	★
3	+	9	◇	15	■	21	●	27	●
4	×	10	⊕	16	●	22	■	28	○
5	◇	11	⊗	17	▲	23	◆	29	○

Size scales



`q <- f + geom_point(aes(size = cyl))`



`q + scale_size_area(max = 6)`
Value mapped to area of circle (not radius)

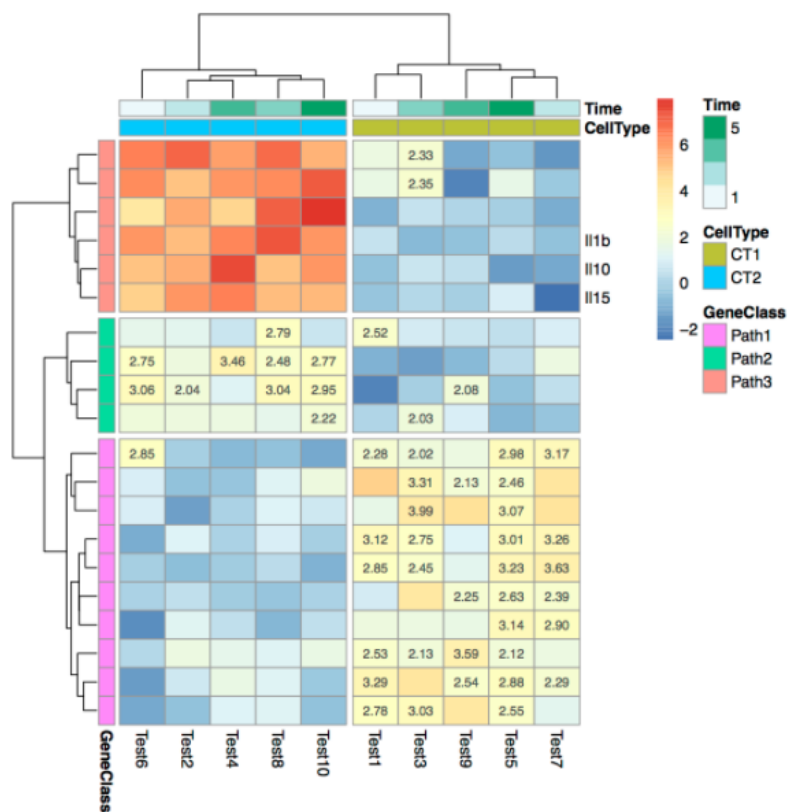


- 为什么要学习R
- R语言发展和学习资源
- R语言统计绘图实战
- R语言其它应用

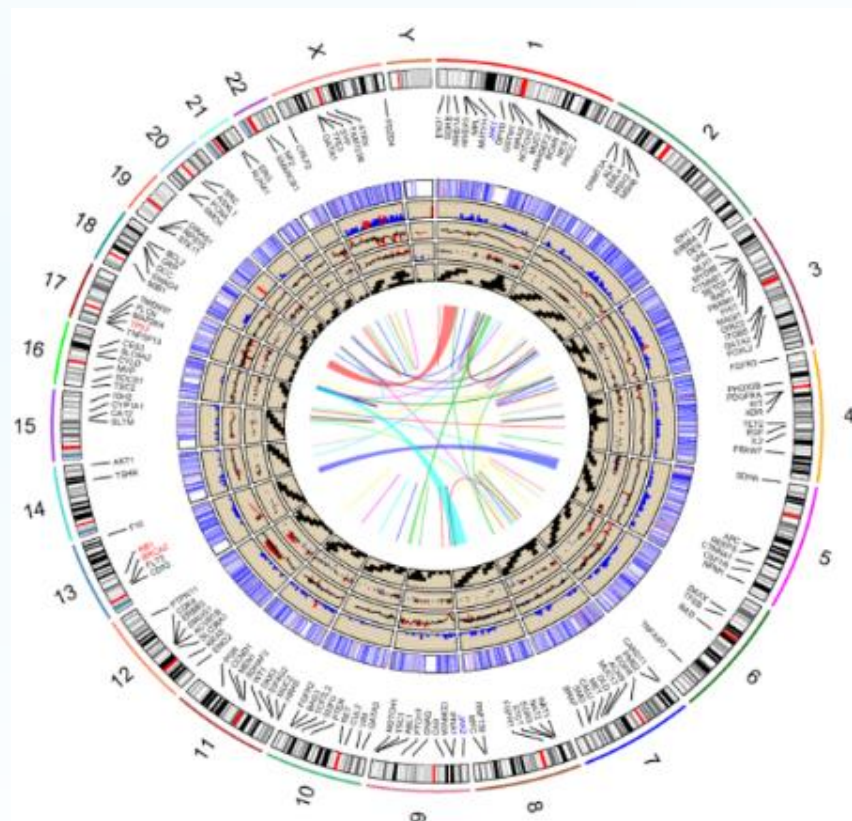
其它专业绘图包



pheatmap

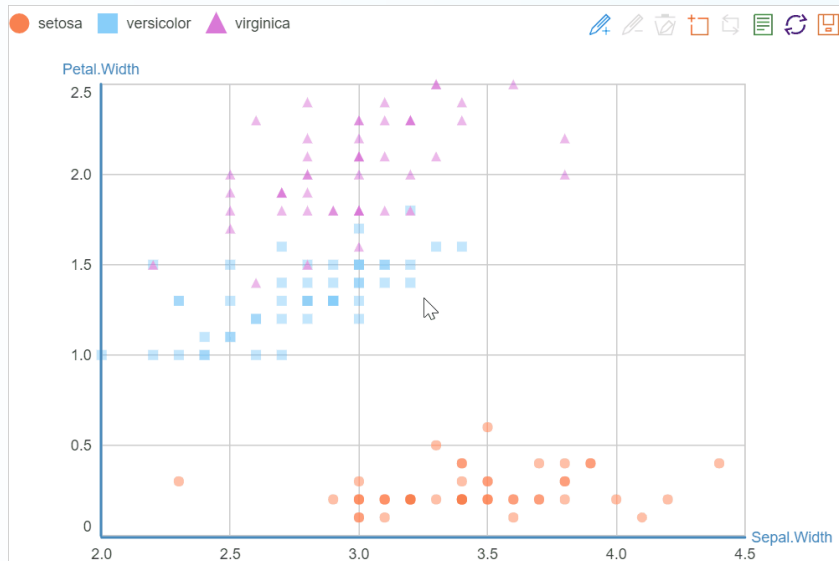
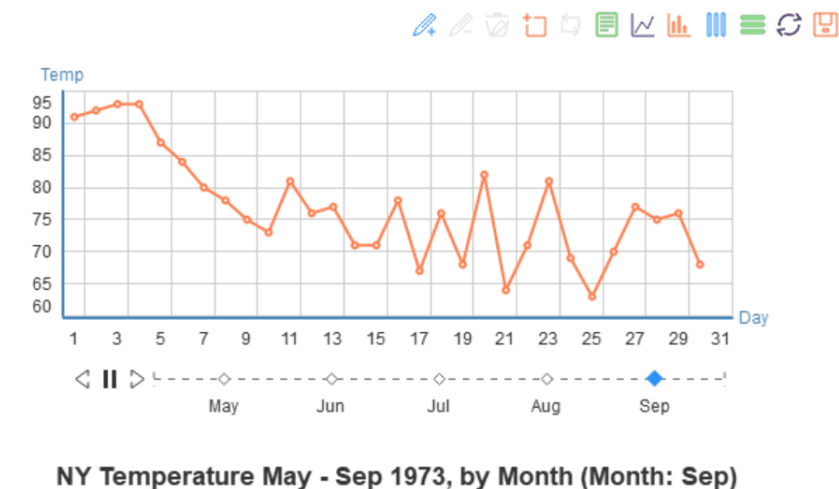


circlize

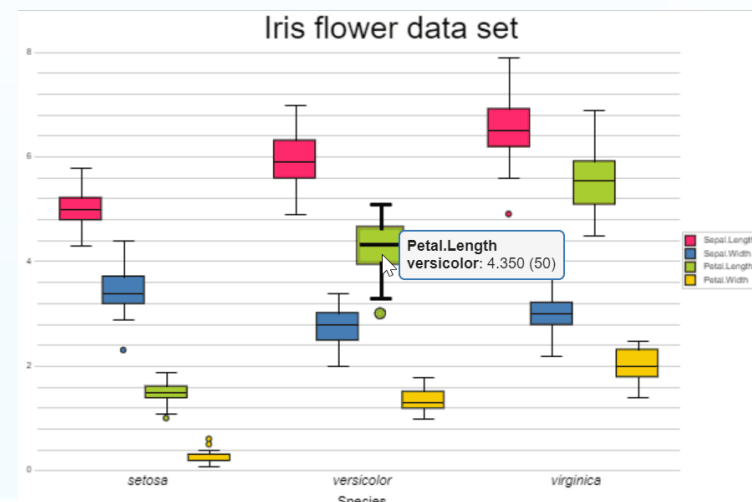




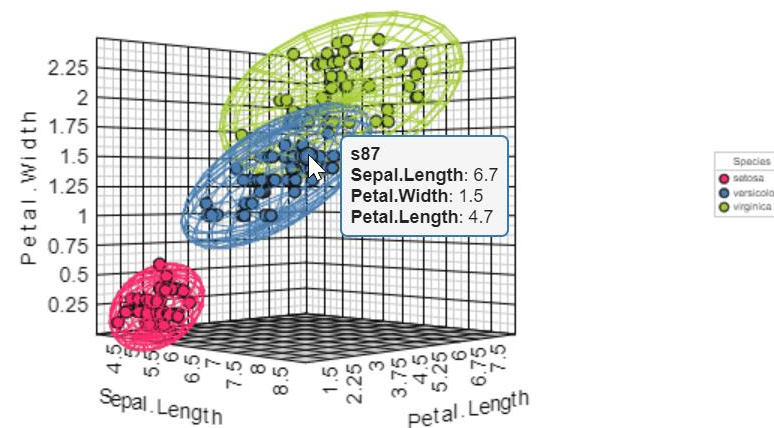
recharts

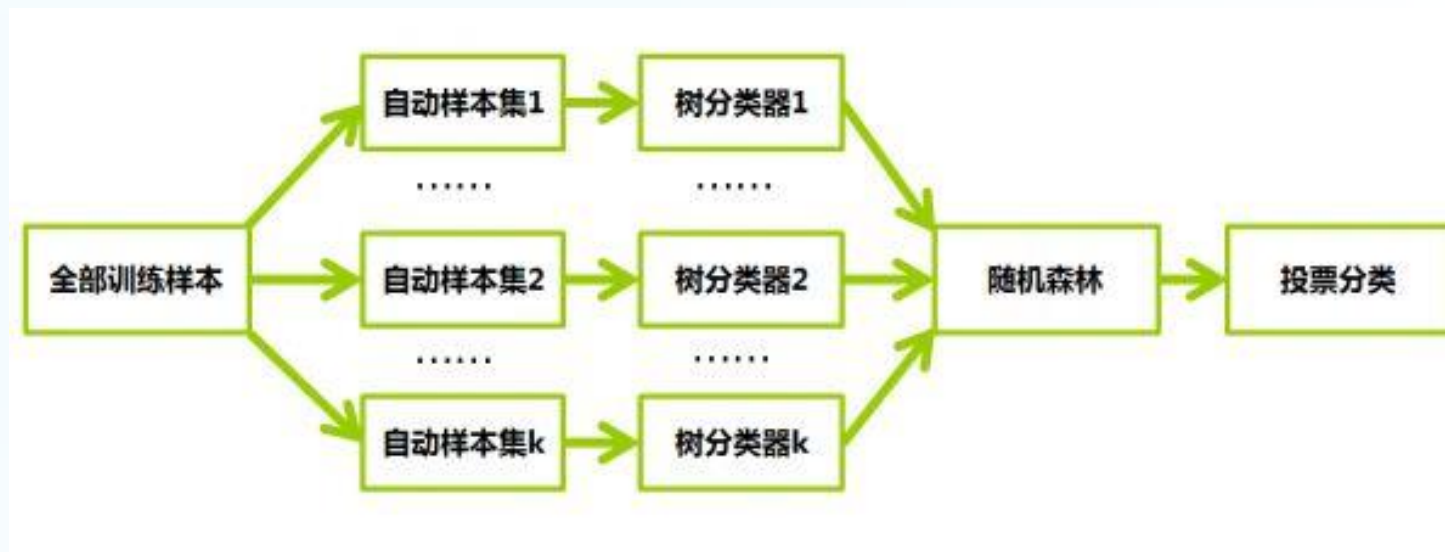


canvasXpress



Iris Data Set







生信宝典

微信号: Bio_data



- 在R中赞扬下努力工作的你, 奖励一份CheatSheet
- 别人的电子书, 你的电子书, 都在bookdown
- R语言 - 入门环境Rstudio
- R语言 - 热图绘制 (heatmap)
- R语言 - 基础概念和矩阵操作
- R语言 - 热图简化
- R语言 - 热图美化
- R语言 - 线图绘制
- R语言 - 线图一步法
- R语言 - 箱线图 (小提琴图、抖动图、区域散点图)
- R语言 - 箱线图一步法
- R语言 - 火山图
- R语言 - 富集分析泡泡图
- R语言 - 散点图绘制
- R语言 - 韦恩图
- R语言 - 柱状图
- R语言 - 图形设置中英文字体
- R语言 - 非参数法生存分析
- R语言 - 绘制seq logo图
- WGCNA分析, 简单全面的最新教程
- 一文看懂PCA主成分分析
- 富集分析DotPlot, 可以服
- 基因共表达聚类分析和可视化
- R中1010个热图绘制方法
- 还在用PCA降维? 快学学大牛最爱的t-SNE算法吧, 附Python/R代码
- 一个函数抓取代谢组学权威数据库HMDB的所有表格数据
- 文章用图的修改和排版
- network3D: 交互式桑基图
- network3D 交互式网络生成
- Seq logo 在线绘制工具——Weblogo
- 生物AI插图素材获取和拼装指导
- ggplot2高效实用指南 (可视化脚本、工具、套路、配色)
- 图像处理R包magick学习笔记
- SOM基因表达聚类分析初探

领先的生物大数据与健康解决方案
Leading solutions for big data and health

宏基因组

微信号: meta-genome



R语言基础系列:

- 你知道R中的赋值符号箭头 <- 和等号 = 的区别吗
- 使用dplyr进行数据操作30例
- 交集intersect、并集union、找不同setdiff
- 1数据类型 (向量、数组、矩阵、列表和数据框)
- 2读写数据所需的主要函数、与外部环境交互
- 3数据筛选——提取对象的子集
- 4向量、矩阵的数学运算
- 5控制结构
- 6函数及作用域
- 7认识循环函数lapply和sapply
- 8分解数据框split和查看对象str
- 9模拟—随机数、抽样、线性模型

ggplot2绘图基础系列:

- 1初识ggplot2绘制几何对象
- 2图层的使用—基础、加标签、注释
- 3工具箱—误差线、加权数、展示数据分布
- 4语法基础
- 5通过图层构建图像
- 6标度、轴和图例
- 7定位-分面 and 坐标系
- 8主题设置、存储导出
- 9绘图需要的数据整理技术

高级统计绘图:

- 热图绘制
- R做线性回归
- 绘图相关系数矩阵corrplot
- 相关矩阵可视化ggcorrplot
- 绘制交互式图形recharts
- 交互式可视化CanvasXpress
- 聚类分析factoextra
- LDA分析、作图及添加置信-ggord
- 解决散点图样品标签重叠ggrepel
- 添加P值或显著性标记ggpubr
- Alpha多样性稀释曲线rarefaction curve
- 堆叠柱状图各成分连线画法: 突出组间变化
- 冲击图展示组间时间序列变化ggalluvial
- 桑基图riverplot
- 微生物环境因子分析ggvegan
- 五彩进化树与热图更配ggtree
- 多元回归树分析mvpart
- 随机森林randomForest 分类Classification 回归Regression
- 加权基因共表达网络分析WGCNA
- circlize包绘制circos-plot

鼠标点点绘图系列

- SCI期刊数据可视化—折线图

<http://mp.weixin.qq.com/s/i71OMaUu6QtcY0pt1njHQA>

http://mp.weixin.qq.com/s/5jQspEvH5_4Xmart22gjMA



扫码关注生信宝典，学习更多生信知识



扫码关注宏基因组，获取专业学习资料

易生信，没有难学的生信知识



易汉博

领先的大数据与健康解决方案
Leading solutions for big data and health

ggplot2图层分步添加

