R for bioinformatics, data visualisation part 2

HUST Bioinformatics course series

Wei-Hua Chen (CC BY-NC 4.0)

22 October, 2024

section 1: TOC

前情提要

iterations 与并行计算

- for loop
- apply functions
- dplyr 的本质是遍历
- map functions in purrr package
- 遍历与并行计算

相关包

- purrr
- parallel
- foreach
- iterators

ggplot2 basics

- 图层（geom_xxx）

本次提要

part 2：ggplot2 进阶

1. 如何在一张图中画多个 panel？
2. 如何写公式
3. ggplot2 的核心是先计算，再做图
4. themes and legends
5. stacked bars 及其它

ggplot2 进阶 1 ： 如何在一张图中画多个 panel ?

key requirements for multi-panel plots

- order / position
- labeling
- layout

combine multiple plots

Useful packages:

- gridExtra
- cowplot
- grid
- lattice

install or load packages

```
if (!require("gridExtra")){
  install.packages("gridExtra");
}

if (!require("cowplot")){
  install.packages("cowplot");
}

library( cowplot );
library( gridExtra );
```

arranging multiple graphs using cowplot

cowplot:

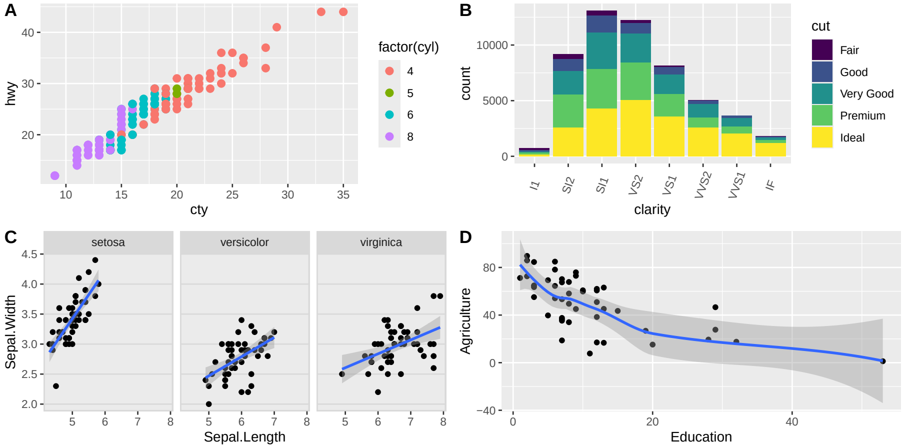
1. plot_grid
2. ggdraw + draw_plot

prepare four panels

```r
library(tidyverse);
plot.hwy <- ggplot(mpg, aes(x = cty, y = hwy, colour = factor(cyl)))+
  geom_point(size=2.5)
# Bar plot
plot.dose <- ggplot(diamonds, aes(clarity, fill = cut)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle=70, vjust=0.5))
plot.iris <- ggplot(iris, aes(Sepal.Length, Sepal.Width)) +
  geom_point() + facet_grid(. ~ Species) + stat_smooth(method = "lm") +
  background_grid(major = 'y', minor = "none") + # add thin horizontal lines
  panel_border();
plot.swiss <- swiss %>% ggplot(aes(Education, Agriculture)) +
  geom_point() + geom_smooth();
```

Combine four plots using plot_grid

```
cowplot::plot_grid(plot.hwy, plot.dose, plot.iris, plot.swiss,
                   labels=c("A", "B", "C", "D"),
                   ncol = 2, nrow = 2)
```

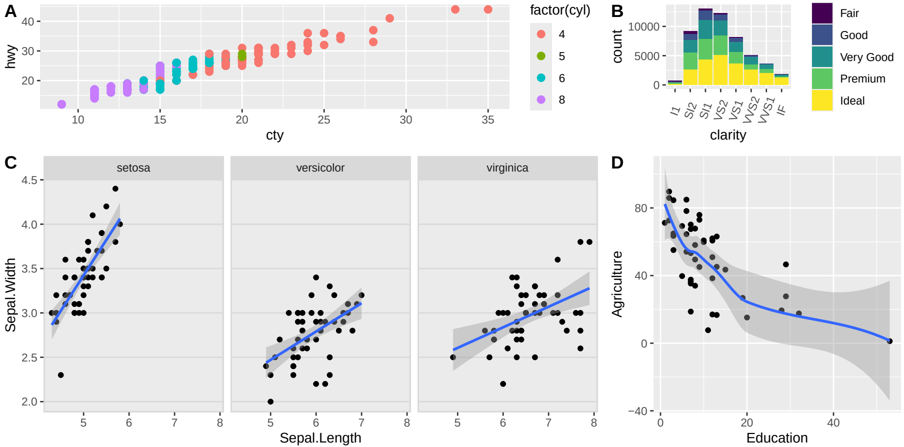


plot_grid parameters

```
plot_grid(
  ...,
  plotlist = NULL,
  align = c("none", "h", "v", "hv"),
  axis = c("none", "l", "r", "t", "b", "lr", "tb", "tblr"),
  nrow = NULL,
  ncol = NULL,
  rel_widths = 1, ## 相对宽度
  rel_heights = 1, ## 相对高度
  labels = NULL,
  label_size = 14,
  label_fontfamily = NULL,
  label_fontface = "bold",
  label_colour = NULL,
  label_x = 0,
  label_y = 1,
  hjust = -0.5,
  vjust = 1.5,
  scale = 1,
  greedy = TRUE,
  byrow = TRUE,
  cols = NULL,
  rows = NULL
)
```

调整相对大小

```
cowplot::plot_grid(plot.hwy, plot.dose, plot.iris, plot.swiss,
                   labels=c("A", "B", "C", "D"), ncol = 2, nrow = 2,
                   rel_widths = c(2,1), rel_heights = c(1, 2));
```



more complex layout using draw_plot

```
plot <-
  ggdraw() +
  draw_plot(plot.iris, x=0, y=.5, width=1, height=.5) +
  draw_plot(plot.hwy, 0, 0, .5, .5) +
  draw_plot(plot.dose, .5, 0, .5, .5) +
  draw_plot_label(c("A", "B", "C"), c(0, 0, 0.5), c(1, 0.5, 0.5), size = 15);
```

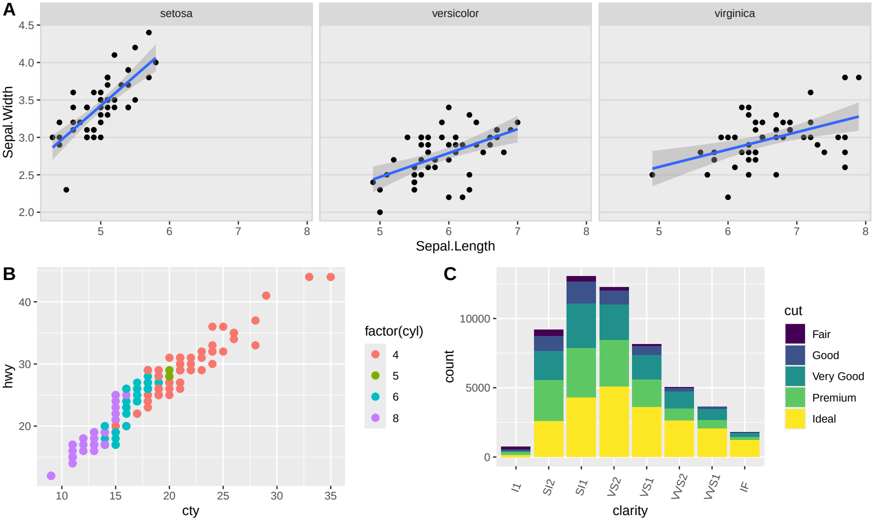
```
## `geom_smooth()` using formula = 'y ~ x'
```

`draw_plot(plot, x = 0, y = 0, width = 1, height = 1)` 详解：

- plot: the plot to place (ggplot2 or a gtable)
- x: The x location of the lower left corner of the plot.
- y: The y location of the lower left corner of the plot.
- width, height: the width and the height of the plot

draw_plot results

```
plot
```



draw_plot_label parameters

Use draw_plot_label to add the labels

```
draw_plot_label(c("A", "B", "C"), c(0, 0, 0.5), c(1, 0.5, 0.5), size = 15);

draw_plot_label(
  label,
  x = 0,
  y = 1,
  hjust = -0.5,
  vjust = 1.5,
  size = 16,
  fontface = "bold",
  family = NULL,
  color = NULL,
  colour,
  ...
)
```

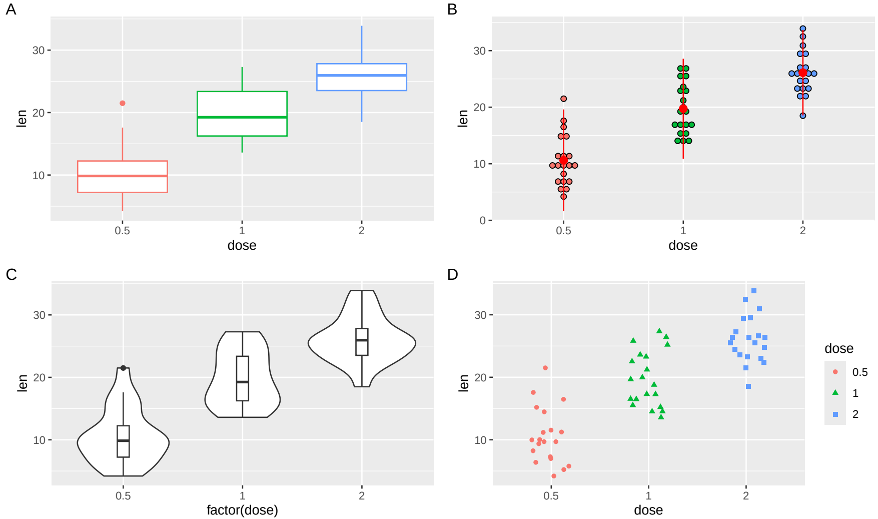
use gridExtra::grid.arrange

Create four plots, again

```
library(ggplot2); library("gridExtra");
df <- ToothGrowth
df$dose <- as.factor(df$dose)

bp <- ggplot(df, aes(x=dose, y=len, color=dose)) +
  geom_boxplot() +
  theme(legend.position = "none") + labs( tag = "A");
dp <- ggplot(df, aes(x=dose, y=len, fill=dose)) +
  geom_dotplot(binaxis='y', stackdir='center')+
  stat_summary(fun.data=mean_sdl, mult=1,
                 geom="pointrange", color="red")+
   theme(legend.position = "none") + labs( tag = "B")
vp <- ggplot(df, aes(x=factor(dose), y=len)) +
  geom_violin()+
  geom_boxplot(width=0.1) + labs( tag = "C")
sc <- ggplot(df, aes(x=dose, y=len, color=dose, shape=dose)) +
  geom_jitter(position=position_jitter(0.2))+
  theme(legend.position = "none") +
  theme_gray() + labs( tag = "D")
```

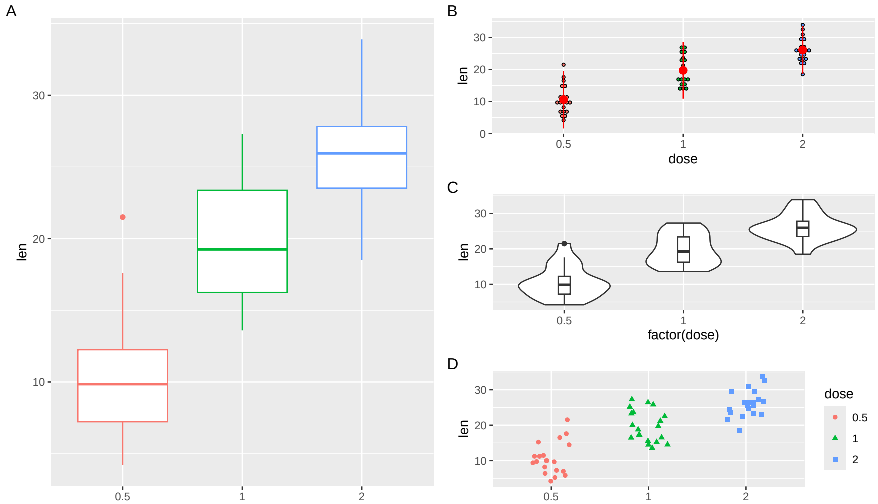
use gridExtra::grid.arrange

```
grid.arrange(bp, dp, vp, sc, ncol=2, nrow =2);
```



use layout_matrix parameter in grid.arrange

```
grid.arrange(bp, dp, vp, sc, ncol = 2,
             layout_matrix = cbind(c(1,1,1), c(2,3,4)));
```



explain `layout_matrix`

```
grid.arrange(bp, dp, vp, sc,
             ncol = 2,  ## two columns
             layout_matrix = cbind(c(1,1,1), c(2,3,4)) ## specify the layout
             );
```

How the layout look like??

```
cbind(c(1,1,1), c(2,3,4));
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    3
## [3,]    1    4
```

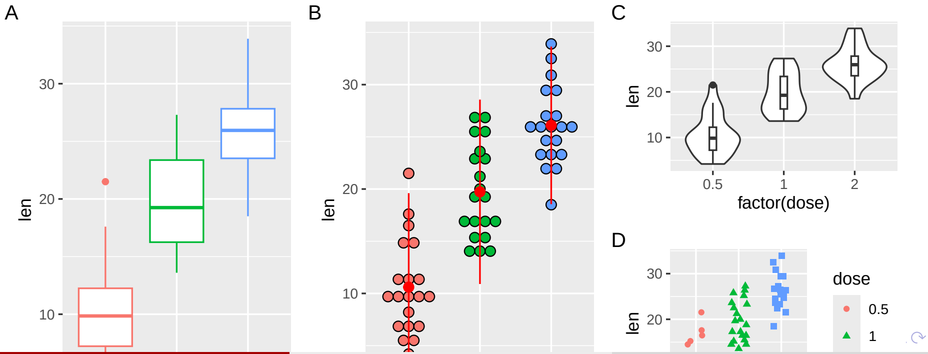
make a different layout???

three columns, A and B take the first two, C and D take the third one.

```
( laymat = cbind(c(1,1), c(2,2), c(3,4)) );
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    1    2    4
```

```
grid.arrange(bp, dp, vp, sc, ncol = 3, layout_matrix = laymat);
```



Add a common legend for multiple ggplot2 graphs

Prepare a function to extract legend from a plot. Note **the legend should exist**.

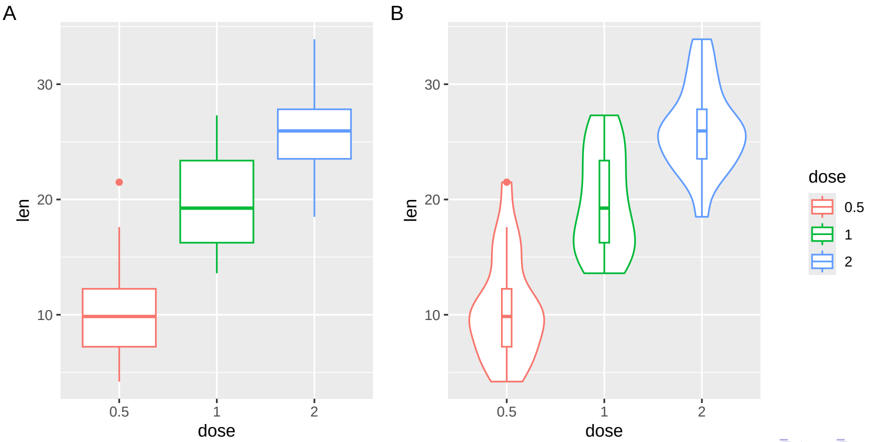
```
library(gridExtra)
get_legend<-function(myggplot){
  tmp <- ggplot_gtable(ggplot_build(myggplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)
}
```

Prepare the graphs and a legend

```r
## 1. Create a box plot WITH legend
bp <- ggplot(df, aes(x=dose, y=len, color=dose)) +
  geom_boxplot() + labs(tag = "A");
## 2. Create a violin plot WITHOUT legend
vp <- ggplot(df, aes(x=dose, y=len, color=dose)) +
  geom_violin()+ geom_boxplot(width=0.1) + labs( tag = "B") +
  theme(legend.position="none") ## no legend
## 3. extract the legend from the first plot
legend <- get_legend(bp);
## 4. remove the legend from the first plot
bp2 <- bp + theme(legend.position="none");
```

plot the common legned to the right

```
grid.arrange(bp2, vp, legend, ## three objects to plot
             ncol=3, ## plot by column
             widths=c(2.3, 2.3, 0.8)); ## set the width of each graph
```

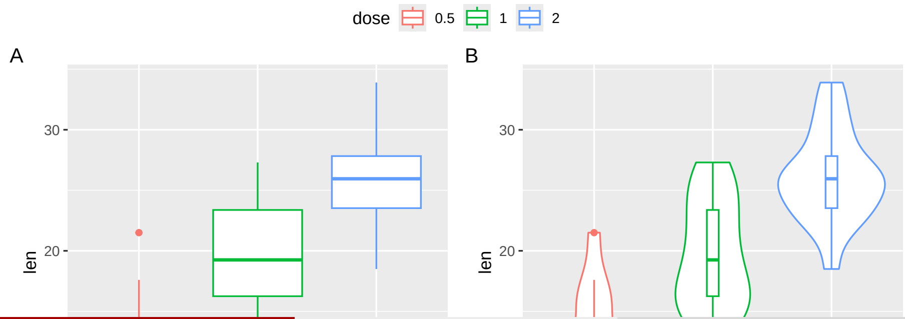


issue

缺点：legend 看起来属于 B，而不是共同的。

fix: place the legend at top and align to the center

```
## 1. re-make bp WITH legend, place the legend at top
bp <- ggplot(df, aes(x=dose, y=len, color=dose)) +
  geom_boxplot() + labs(tag = "A") + theme(legend.position = "top");
## 2. extract the legend
legend <- get_legend(bp);
## 3. remove teh legend from the plot
bp2 <- bp + theme(legend.position = "none")
## 4. plot
grid.arrange(legend, bp2, vp,  ncol=2, nrow = 2,
             layout_matrix = rbind(c(1,1), c(2,3)),
             widths = c(2.7, 2.7), heights = c(0.2, 2.5));
```



Explain

```
grid.arrange(legend, bp2, vp,
             ncol=2, nrow = 2,
             layout_matrix = rbind(c(1,1), c(2,3)),
             widths = c(2.7, 2.7), heights = c(0.2, 2.5));
```

```
rbind(c(1,1), c(2,3));
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    2    3
```

- legend takes the first row, and has a height of 0.2
- the other two graphs take the second row and has a height of 2.5

Practise on your own

To place the legend at:

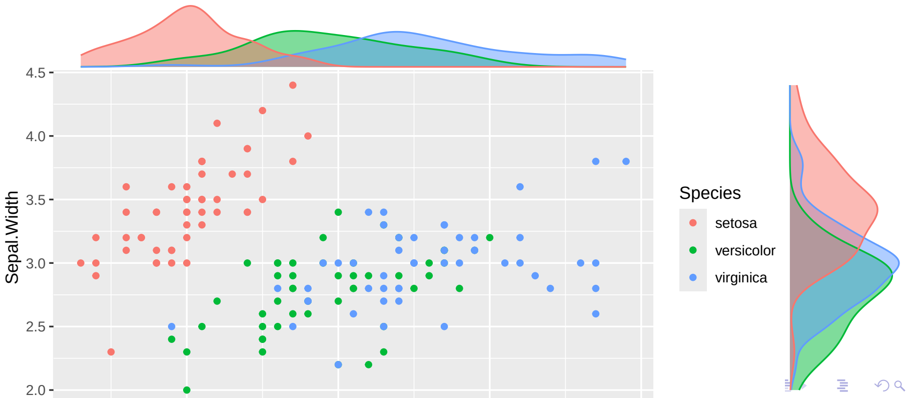
- the bottom, centered at the middle
- top-left
- top-right
- bottom-left
- bottom-right

ggExtra - Add marginal histograms to ggplot2

please install the package if not exists …

```
install.packages("ggExtra")
```

```
library(ggExtra);
piris <- ggplot(iris, aes(Sepal.Length, Sepal.Width, colour = Species)) +
  geom_point()
ggMarginal(piris, groupColour = TRUE, groupFill = TRUE)
```

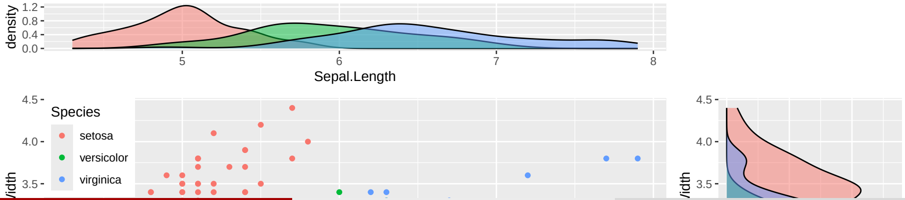


也可自己写代码实现

```
piris <- ggplot(iris, aes(Sepal.Length, Sepal.Width, colour = Species)) +
  geom_point() + theme(legend.position=c(0,1), legend.justification=c(0,1))
```

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
xdensity <- ggplot(iris, aes(Sepal.Length, fill = Species)) +
  geom_density(alpha=.5) + theme(legend.position = "none")
ydensity <- ggplot(iris, aes(Sepal.Width, fill=Species)) +
  geom_density(alpha=.5) + theme(legend.position = "none") + coord_flip()
grid.arrange(xdensity, NULL, piris, ydensity,
        ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4, 4));
```



Extended reading

Other ggplot2 extensions

See the gallery at https://exts.ggplot2.tidyverse.org/gallery/.
Or Google ggplot2 extensions gallery.

Explore the grid package

- create graphical objects (grobs)
- arrange multiple grobs using arrangeGrob function

Explore the gridExtra package

- plot table
- ...

complex layout 小结

Essentials for combining multiple graphs in one:

- ordering
- layout

cowplot

- plot_grid
- draw_plot

gridExtra

- grid.arrange

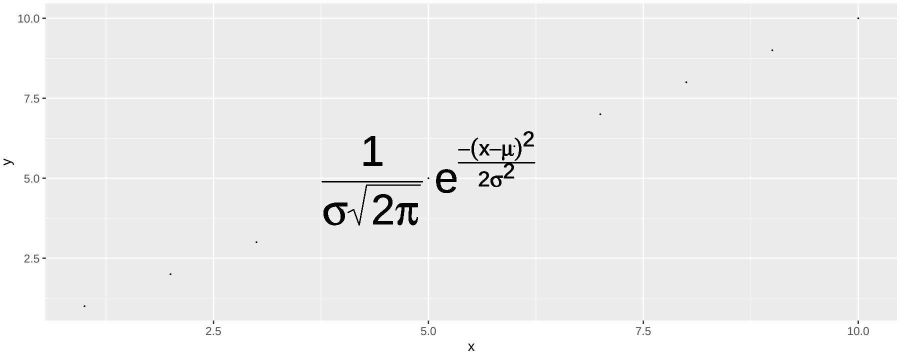
ggplot2 进阶 2：如何写公式？

two kinds of equations

1. expression w/o variable substitution
2. expression w/ variable substitution

the 1st kind equations (expression)

```
eq <- expression(paste(frac(1, sigma*sqrt(2*pi)), " ",
                       plain(e)^{frac(-(x-mu)^2, 2*sigma^2)}));

ggplot( data.frame(x=1:10, y=1:10), aes( x,y ) ) +
    geom_point( size = 0 ) +
    geom_text(data = NULL, x = 5, y = 5, size = 12,
              label = as.character(eq), parse = TRUE );
```



公式中的写法之代数符号

分类	R 的表达式	显示结果
代数符号	`expression(x + y)`	$x + y$
	`expression(x - y)`	$x - y$
	`expression(x * y)`	xy
	`expression(x / y)`	x/y
	`expression(x %+-% y)`	$x \pm y$
	`expression(x %/% y)`	$x \div y$
	`expression(x %*% y)`	$x \times y$
	`expression(x %.% y)`	$x \cdot y$
	`expression(x[i])`	x_i
	`expression(x^2)`	x^2
	`expression(sqrt(x))`	$\sqrt{x}$
	`expression(sqrt(x,y))`	$\sqrt[y]{x}$
	`expression(list(x,yz))`	x, y, z

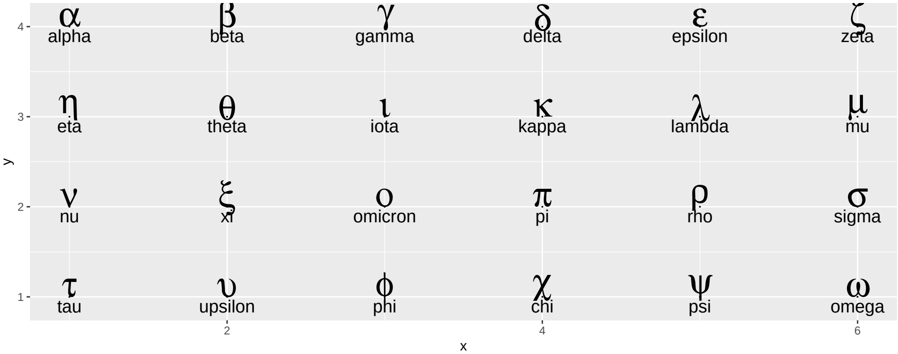
… 更多，不在这里介绍了。

希腊字符

代码

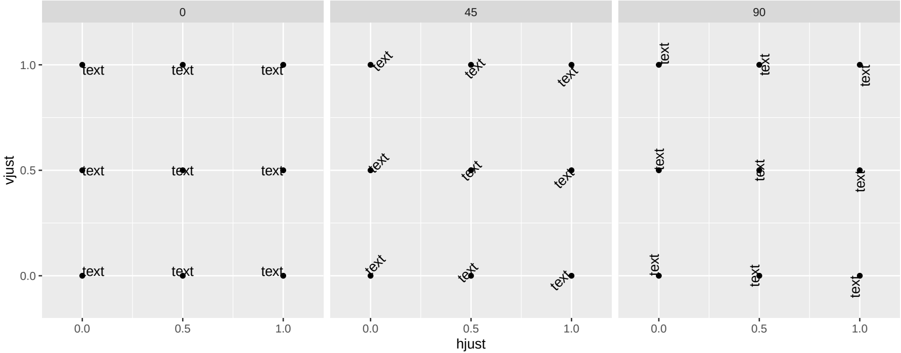
```r
library(ggplot2);
greeks <- c("Alpha", "Beta", "Gamma", "Delta", "Epsilon", "Zeta",
            "Eta", "Theta", "Iota", "Kappa", "Lambda", "Mu",
            "Nu", "Xi", "Omicron", "Pi", "Rho", "Sigma",
            "Tau", "Upsilon", "Phi", "Chi", "Psi", "Omega");

dat <- data.frame( x = rep( 1:6, 4 ), y = rep( 4:1, each = 6), greek = greeks );

plot2 <-
  ggplot( dat, aes(x=x,y=y) ) + geom_point(size = 0) +
    # 画希腊字符，注意下面两行代码的区别
    geom_text( aes( x, y + 0.1, label = tolower( greek ) ), size = 10, parse = T ) +
    geom_text( aes( x, y - 0.1, label = tolower( greek ) ), size = 5  );
```

希腊字符，cont.


α
alpha
β
beta
γ
gamma
δ
delta
ε
epsilon
ζ
zeta
η
eta
θ
theta
ι
iota
κ
kappa
λ
lambda
μ
mu
ν
nu
ξ
xi
ο
omicron
π
pi
ρ
rho
σ
sigma
τ
tau
υ
upsilon
ϕ
phi
χ
chi
ψ
psi
ω
omega
y
4
3
2
1
x
2
4
6

hjust 和 vjust

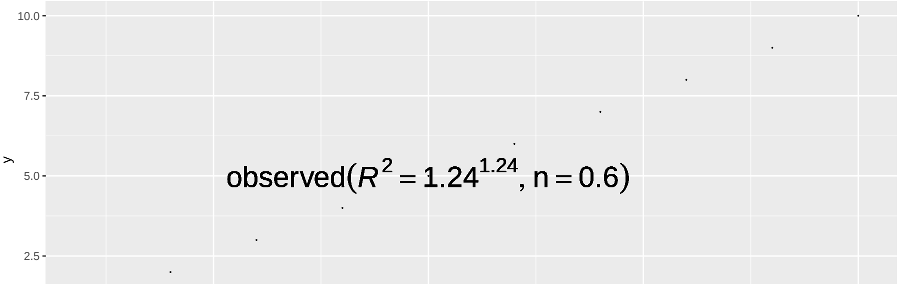
geom_text(aes(angle, hjust, vjust)) 三参数详解



2. 有变量代入值的公式

举例 1：

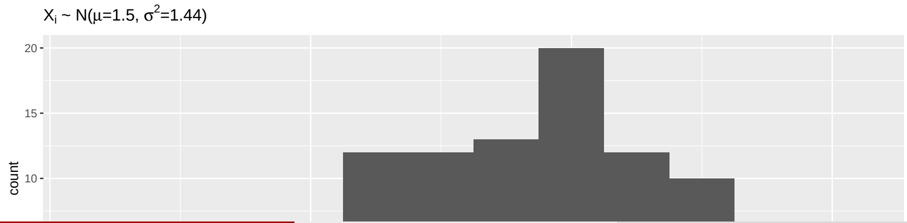
```
x <- 1.24;
y <- 0.6;

ex <- bquote(.(parse(text=paste( "observed (", "italic(R)^2==",
                               x,  "^bold(", x, "), n == ", y, ")",
                               sep = "  " ))) );

ggplot( data.frame(x=1:10, y=1:10), aes( x,y ) ) + geom_point( size = 0 ) +
    geom_text(data = NULL, x = 5, y = 5, size = 8,
              label = as.character(ex), parse = TRUE );
```



有代入变量值的公式举例 2：

使用 paste 和 substitute：

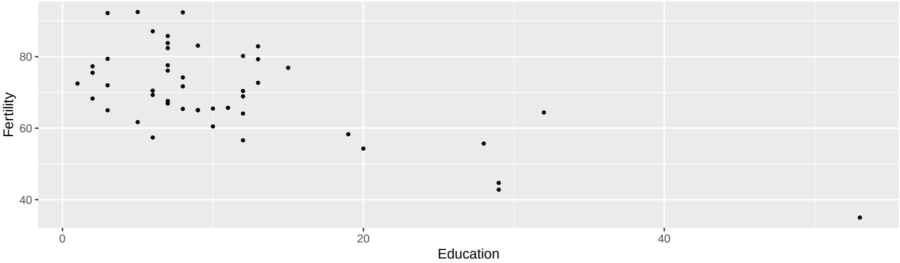
```
x_mean <- 1.5;
x_sd <- 1.2;

# 表达式
ex <- substitute(
    paste(X[i], " ~ N(", mu, "=", m, ", ", sigma^2, "=", s2, ")"),
    list(m = x_mean, s2 = x_sd^2)
);

# histogram
ggplot( data.frame( x = rnorm(100, x_mean, x_sd) ), aes( x ) ) +
    geom_histogram( binwidth=0.5 ) +
    ggtitle(ex); ## 为什么不需要 parse = TURE ????
```



详解

示例：显示两组数据间的相关性

```
## 作图
ggplot( swiss, aes( x = Education, y = Fertility ) ) +
  geom_point( shape = 20 );
```

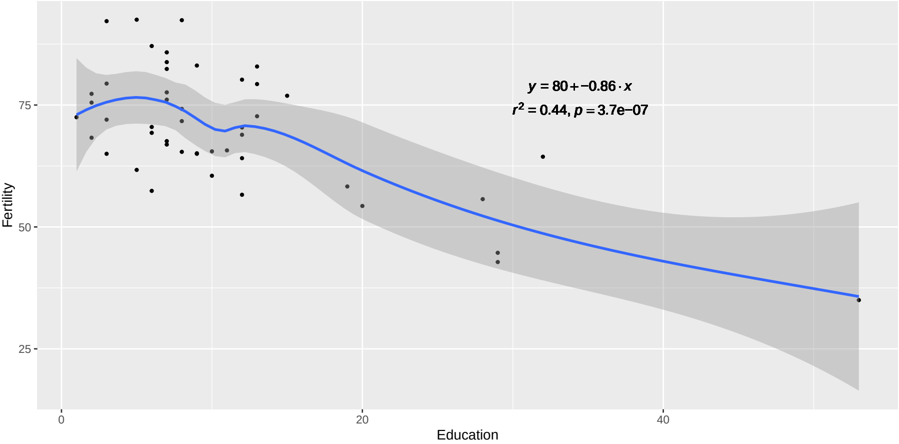


```
## 分析
with( swiss, cor.test( Education, Fertility  )$estimate );
```

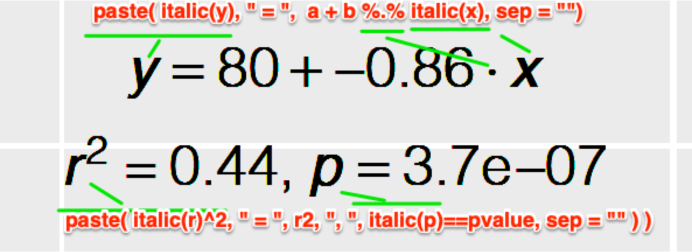
```
##        cor
## -0.6637889
```

在图中加入公式和统计信息

先展示一下结果



公式详解



```
paste( italic(y), " = ",  a + b %.% italic(x), sep = "")
```

$$y = 80 + -0.86 \cdot x$$

$$r^2 = 0.44,\ p = 3.7\mathrm{e}{-07}$$

```
paste( italic(r)^2, " = ", r2, ", ", italic(p)==pvalue, sep = "" ) )
```

Figure 1: equation explained!

公式详解，cont.

以下代码实现两个任务：

1. 将两个公式上下放置 atop (<equation_1> , <equation_2>);
2. 将公式中的某些值替换为数值 substitute(<equation>, list(...))

```
## 计算 ...
m = lm(Fertility ~ Education, swiss);
c = cor.test( swiss$Fertility, swiss$Education );

## 生成公式
eq <- substitute( atop( paste( italic(y), " = ",  a + b %.% italic(x), sep = ""),
                        paste( italic(r)^2, " = ", r2, ", ", italic(p)==pvalue, sep = "" ) ),
                      list(a = as.vector( format(coef(m)[1], digits = 2) ),
                           b =  as.vector(  format(coef(m)[2], digits = 2) ),
                           r2 =  as.vector( format(summary(m)$r.squared, digits = 2) ),
                           pvalue =  as.vector( format( c$p.value , digits = 2) ) )
    );

## 用 as.expression 对公式进行转化
eq <- as.character(as.expression(eq));
```

完整代码

```r
## 计算 ...
m = lm(Fertility ~ Education, swiss);
c = cor.test( swiss$Fertility, swiss$Education );

## 生成公式
eq <- substitute( atop( paste( italic(y), " = ",  a + b %.% italic(x), sep = ""),
                        paste( italic(r)^2, " = ", r2, ", ", italic(p)==pvalue, sep = "" ) ),
                      list(a = as.vector( format(coef(m)[1], digits = 2) ),
                           b =  as.vector(  format(coef(m)[2], digits = 2) ),
                           r2 =  as.vector( format(summary(m)$r.squared, digits = 2) ),
                           pvalue =  as.vector( format( c$p.value , digits = 2) ) )
    );

## 用 as.expression 对公式进行转化  !!!!
eq <- as.character(as.expression(eq));

## 作图，三个图层；特别是 geom_text 使用自己的 data 和 aes ...
ggplot(swiss, aes( x = Education,  y = Fertility ) ) +
        geom_point( shape = 20 ) +
        geom_smooth( se = T ) + ## smooth line ...
        geom_text( data = NULL,
                   aes( x = 30, y = 80, label= eq, hjust = 0, vjust = 1), ## hjust, vjust ???
                   size = 4, parse = TRUE, inherit.aes=FALSE); ## 注意： parse = TRUE !!!
```

equation 的其它写法（更复杂难懂）

```r
## 计算 ...
m = lm(Fertility ~ Education, swiss);
c = cor.test( swiss$Fertility, swiss$Education );

## 生成公式
eq <- substitute( atop( italic(y) == a + b %.% italic(x),
                            italic(r)^2~"="~r2*","~italic(p)==pvalue ),
                      list(a = as.vector( format(coef(m)[1], digits = 2) ),
                           b =  as.vector(  format(coef(m)[2], digits = 2) ),
                           r2 =  as.vector( format(summary(m)$r.squared, digits = 2) ),
                           pvalue =  as.vector( format( c$p.value , digits = 2) ) )
    );
## 用 as.expression 对公式进行转化  !!!!
eq <- as.character(as.expression(eq));

## 作图，三个图层；特别是 geom_text 使用自己的 data 和 aes ...
ggplot(swiss, aes( x = Education,  y = Fertility ) ) +
        geom_point( shape = 20 ) +
        geom_smooth( se = T ) + ## smooth line ...
        geom_text( data = NULL,
                   aes( x = 30, y = 80, label= eq, hjust = 0, vjust = 1), ## hjust, vjust ???
                   size = 4, parse = TRUE, inherit.aes=FALSE); ## 注意：  parse = TRUE !!!
```

公式详解

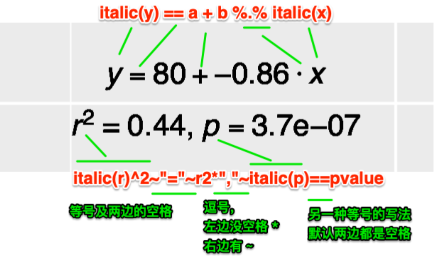


Figure 2: equation explained!

注

- 引号两边必须有 * 或 ~ 字符，~ 表示空格，* 表示什么都没有。~~ 表示两个空格。如果公式中需要 ~ 字符怎么办 ?? 见下面 “公式示例 3”。

公式小结

- 表达式
- 罗马字符
- 代入变量

ggplot2 进阶 3：核心在于先计算再做图

举例说明

先看数据（来自 talk05）：

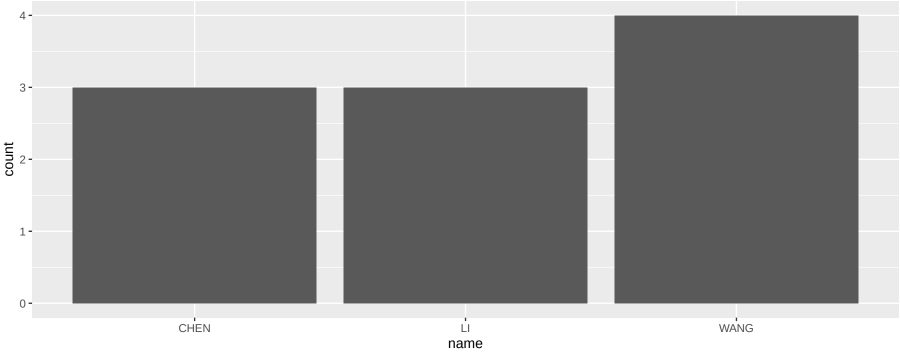
```
grades2 <- read_delim( file = "data/talk05/grades2.txt", delim = "\t",
                       quote = "", col_names = T);
knitr::kable( grades2 );
```

name	class	course	grade
CHEN	1	bioinformatics	90
CHEN	1	chemistry	92
CHEN	2	chinese	35
CHEN	3	german	62
LI	1	bioinformatics	44
LI	2	chinese	68
LI	3	microbiology	95
LI	3	japanese	90
WANG	1	bioinformatics	35
WANG	1	chemistry	76
WANG	1	mathmatics	82
WANG	3	german	100
WANG	3	spanish	78

geom_bar

任务：画出每位学生及格的课程数

```
ggplot( grades2 %>% filter( grade >= 60 ), aes( name ) ) +
  geom_bar();
```

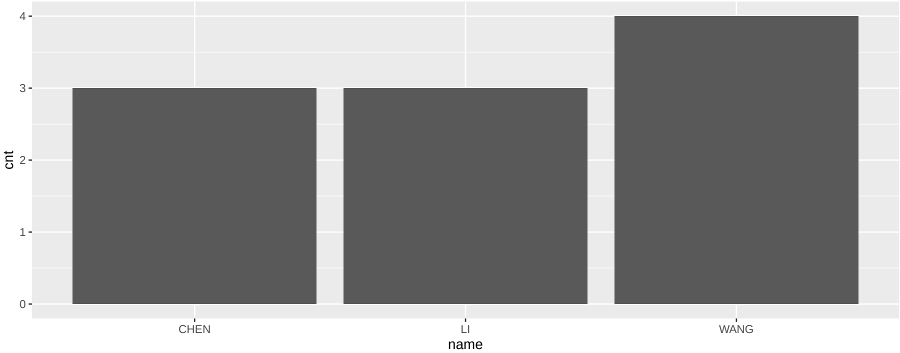


为什么会这样呢？因为 `geom_bar( stat = "count" )` 的默认参数是 `count`，即数一下每个 factor 的出现次数。

geom_bar，cont.

以上命令，实际上等于：

```
## 先做统计
cnt <- grades2 %>% group_by( name ) %>% summarise( cnt = sum( grade >= 60 )  );
ggplot( cnt, aes( x = name, y = cnt ) ) +
  geom_bar( stat = "identity" );
```



default stat behaviors (默认计算方法)

- geom_bar : count
- geom_boxplot : boxplot
- geom_count : sum
- geom_density : density
- geom_histogram : bin
- geom_quantile : quantile ...

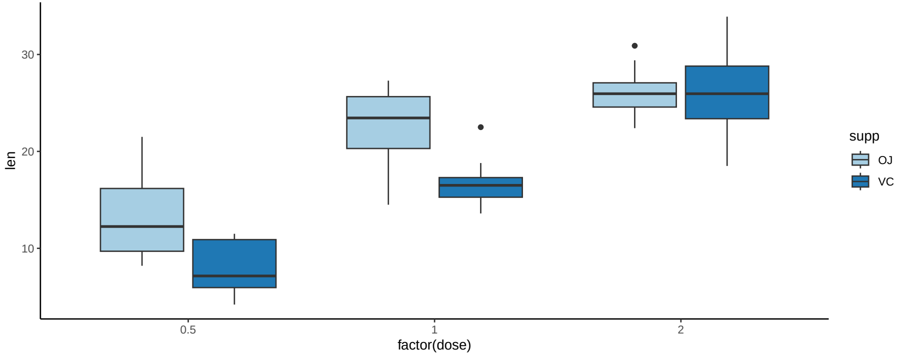
ggplot2 进阶 4: themes and legends

改变 theme()

theme 调整包括 theme() 函数，用于调整各个 elements

和 theme_xxx() 函数，直接使用已经定制好的内容；

```
ggplot(ToothGrowth, aes(x=factor( dose ), y=len, fill=supp)) +
  geom_boxplot() + scale_fill_brewer( palette = "Paired" ) + theme_classic();
```



ggplot2 中的主题

- theme_gray：系统默认主题
- theme_bw，theme_linedraw, theme_light, theme_dark, theme_minimal，theme_classic, theme_void()

see here for a complete list: https://ggplot2.tidyverse.org/reference/ggtheme.html

theme() 函数

除了 theme_ 用于调整整体视觉效果外，ggplot2 还提供了 theme() 函数用于细调。

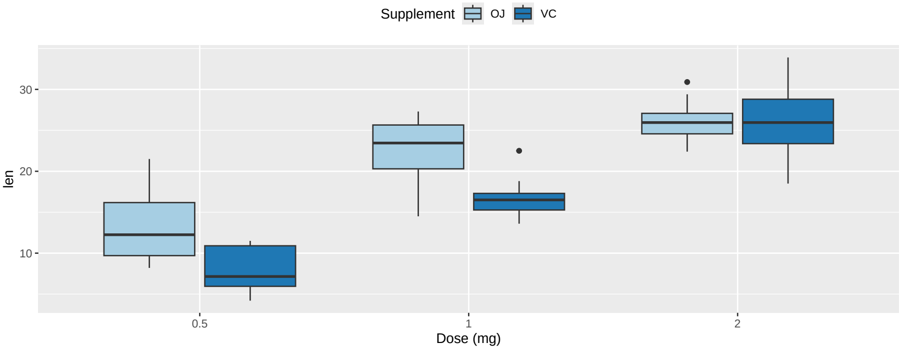
- line, rect, text, title ： 整体框架
- axis.<component> ： 调整坐标轴
- legend.<parameter> ： 调整图例
- plot.<> ： 控制 title, subtitle 等细节
- panel.<...> ： 调整 facet 情况下的 panel （facet 下面会介绍）
- strip.<...> ： 调整 facet 的标题细节 ...

更多详见：

官方： https://ggplot2.tidyverse.org/reference/theme.html

legend 细调

```r
ggplot(ToothGrowth, aes(x=factor( dose ), y=len, fill=supp)) +
  geom_boxplot() + scale_fill_brewer( palette = "Paired" ) +
  labs( fill = "Supplement", x = "Dose (mg)" ) +
  theme( legend.position = "top" )
```

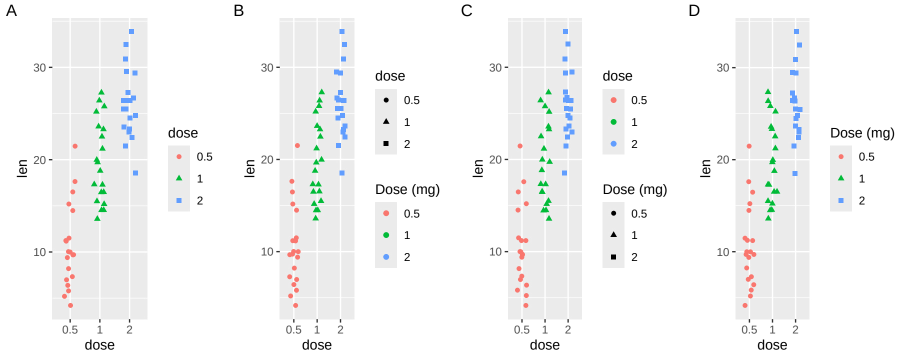


labs() function: Modify axis, legend, and plot labels

```
labs(
  ...,
  x = "<x label>",
  y = "<y label>",
  colour = "<legend title>", # 与 aes 里的 colour 配合使用
  fill = "<legend title>", # 与 aes 里的 fill 配合使用
  shape = "<legend title>", # 与 aes 里的 shape 配合使用
  title = waiver(),
  subtitle = waiver(),
  caption = waiver(),
  tag = waiver(),
  alt = waiver(),
  alt_insight = waiver()
)
```

labs() with examples

```
grid.arrange(sc + labs(tag = "A"),
             sc + labs( colour = "Dose (mg)" , tag = "B"),
             sc + labs( shape = "Dose (mg)" , tag = "C"),
             sc + labs( colour = "Dose (mg)", shape = "Dose (mg)", tag = "D"  ),
             ncol=4, nrow =1);
```



ggplot2 进阶 5: stacked bars 及其它

stacked bars

应用场景：宏基因组多样本物种丰度图

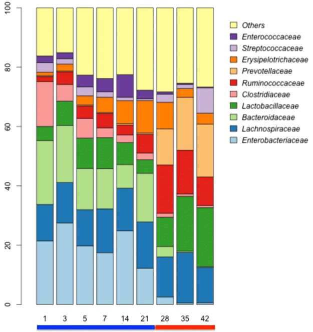


Figure 3: Microbiome 3, 28 2015

stacked bars , cont.

load data

```
speabu <-read_tsv( file = "data/talk09/mock_species_abundance.txt"  );
```

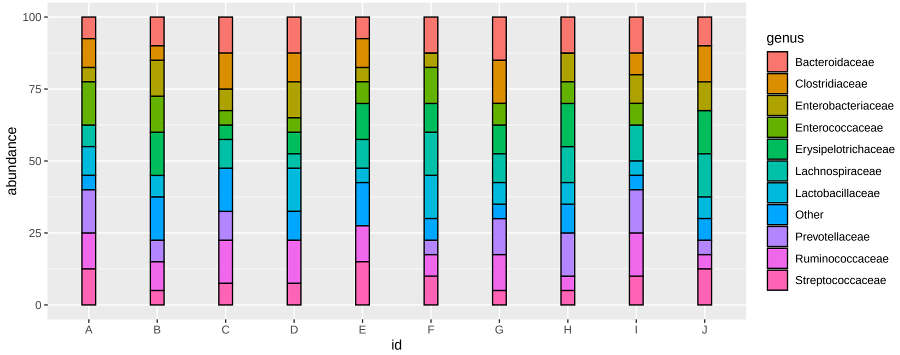
```
## Rows: 110 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): id, genus
## dbl (1): abundance
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head( speabu );
```

```
## # A tibble: 6 x 3
##   id    genus              abundance
##   <chr> <chr>                  <dbl>
## 1 A     Enterobacteriaceae       5
## 2 A     Lachnospiraceae          7.5
## 3 A     Bacteroidaceae           7.5
## 4 A     Lactobacillaceae        10
## 5 A     Clostridiaceae          10
## 6 A     Ruminococcaceae         12.5
```

stacked bars , cont.

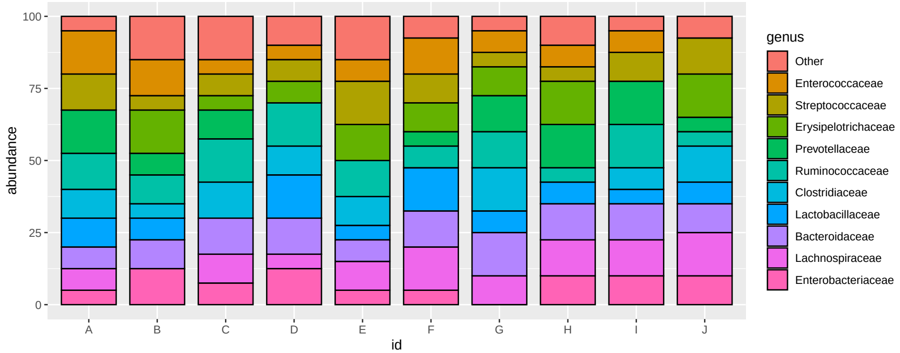
```
ggplot( speabu, aes( x = id, y = abundance, fill = genus ) ) +
  geom_bar( stat = "identity", position = "stack", color = "black", width = 0.2 );
```



指定 Genus 展示顺序

factor 的操纵详见第 4 章。

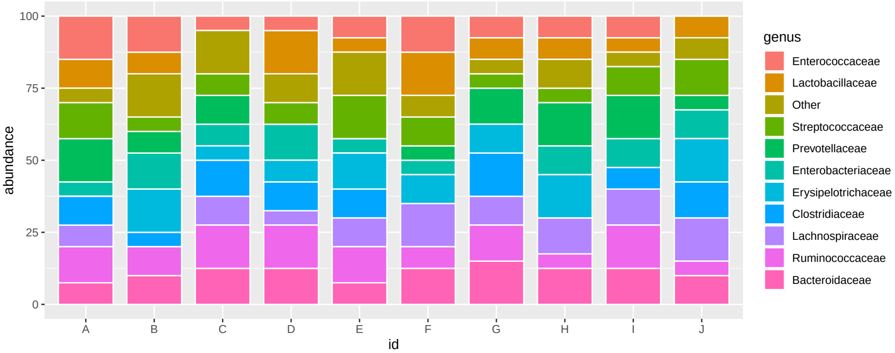
```
speabu$genus <- factor( speabu$genus, levels = rev( c( "Enterobacteriaceae", "Lachnospiraceae",
       "Clostridiaceae", "Ruminococcaceae", "Prevotellaceae", "Erysipelotrichaceae", "Streptoco
ggplot( speabu, aes( x = id, y = abundance, fill = genus ) ) +
  geom_bar( stat = "identity", position = "stack", color = "black", width = 0.8 );
```



按丰度排序

按丰度中值大小排序

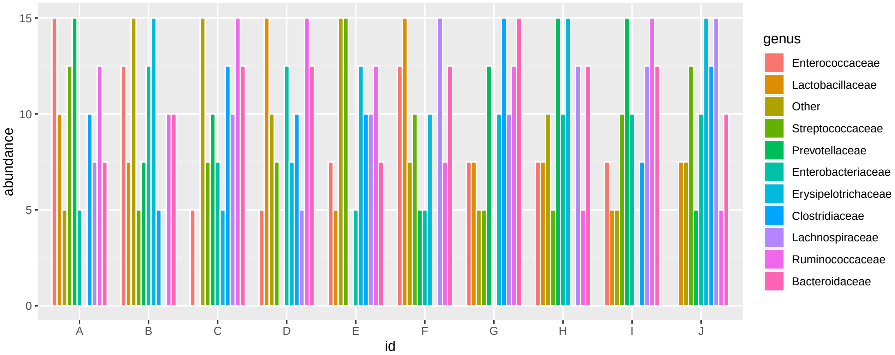
```
speabu$genus <- reorder( speabu$genus, speabu$abundance, median );
ggplot( speabu, aes( x = id, y = abundance, fill = genus ) ) +
  geom_bar( stat = "identity", position = "stack", color = "white", width = 0.8 );
```



position = "stack" 又是什么 ??

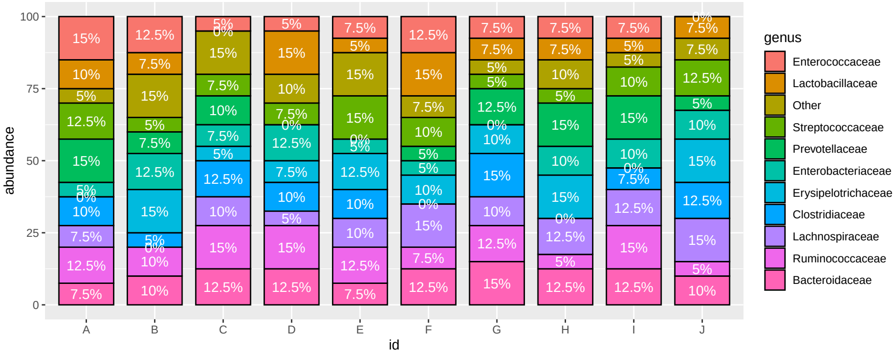
`position = "dodge"` : plot bars next to each other ...

```
ggplot( speabu, aes( x = id, y = abundance, fill = genus ) ) +
  geom_bar( stat = "identity", position = "dodge", color = "white", width = 0.8 );
```



显示数值 ...

```
## 先计算显示位置
speabu <- speabu %>% arrange( id, desc( factor( genus ) ) ) %>%
  group_by( id ) %>% mutate( ypos = cumsum( abundance ) - abundance / 2 );
## 画图
ggplot( speabu, aes( x = id, y = abundance, fill = genus ) ) +
  geom_bar( stat = "identity", position = "stack", color = "black", width = 0.8 ) +
  geom_text( aes( y = ypos, label = paste( abundance, "%", sep = "" ) ), color = "white" );
```



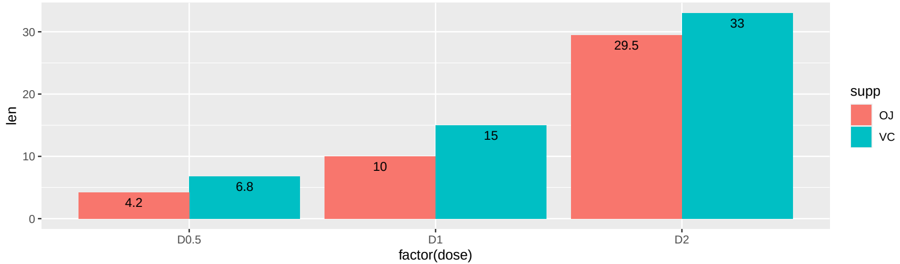
显示数值 ... , cont.

要点

- 使用 ddplyr 的 cumsum() 函数 ...
- 计算位置：当前累加值 - 自身值/2，使数字显示在当前值的中间
- 累加前，要对数据按 factors 进行排序；通过 arrange 函数实现；

在 position = "dodge" 的情况下添加 label

```
df2 <- data.frame(supp=rep(c("VC", "OJ"), each=3),
                dose=rep(c("D0.5", "D1", "D2"),2),
               len=c(6.8, 15, 33, 4.2, 10, 29.5))
ggplot( df2, aes(x=factor(dose), y=len, fill=supp)) +
  geom_bar(stat="identity", position=position_dodge())+
  geom_text(aes(label=len), vjust=1.6, color="black",
            position = position_dodge(0.9), size=3.5)
```



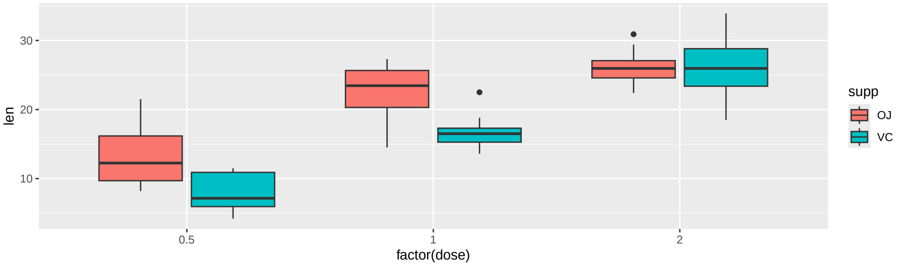
position 的其它取值

除了 “dodge”, “stack” 之外，position 还可以：

- position = position_stack(reverse = TRUE)
- position = position_dodge(reverse = TRUE)
- position = position_identity()
- position = position_jitter() ：jitter points to avoid overplotting ...
- position = position_nudge() ：is generally useful for adjusting the position of items on discrete scales by a small amount

不同的图层有不同默认值

```
ggplot(ToothGrowth, aes(x=factor( dose ), y=len, fill=supp)) +
  geom_boxplot()
```



geom_boxplot()：默认为 dodge

more to read

ggplot2 的在线书

`https://ggplot2-book.org/themes`

Exercise and home work

总结，本节内容

ggplot2 基础

- 优缺点
- 用法
- 基本组成

ggplot2 进阶

- 颜色和色板
- 复杂 layout 的实现
- 公式
- ggplot2 的数据统计逻辑

更多阅读

- Ggplot2: Elegant Graphics for Data Analysis, Book by Hadley Wickham
- ggplot2 gallery provided by RStudio on Github

写在后面

1. ggplot2 博大精深，需要一门课去讲
2. 上手容易，精通难
3. 太多记忆点
4. 本节内容只涉及了基础中的基础，更多内容，包括进阶技巧和生信相关的扩展包，更多的需要同学们自行探索
5. 遇到不会的图，先百度/Google，找包和代码

下次预告

data summary and modeling

作业

- `Exercises and homework` 目录下 `talk09-homework.Rmd` 文件；
- 完成时间：见钉群的要求