

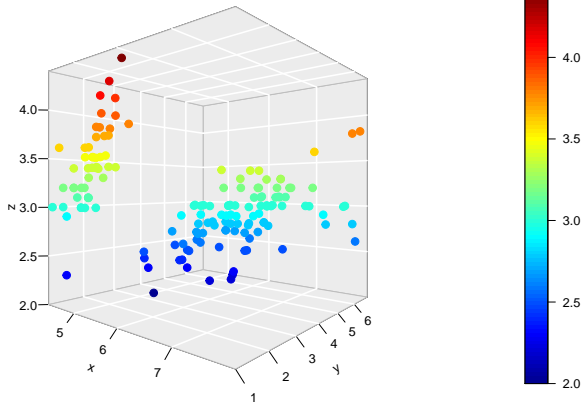
R language and data analysis:plot basics

Qiang Shen

Dec.12,2017

Visualization

- A picture is worth a thousand words



Visualization

- ▶ “The adage ‘A picture is worth a thousand words’ refers to the notion that a complex idea can be conveyed with just a single still image, or that an image of a subject conveys its meaning/essence more effectively than a description does. It also aptly characterizes one of the main goals of visualization namely making it possible to absorb large amounts of data quickly.” –Wikipedia
- ▶ picture superiority effect

Visualization

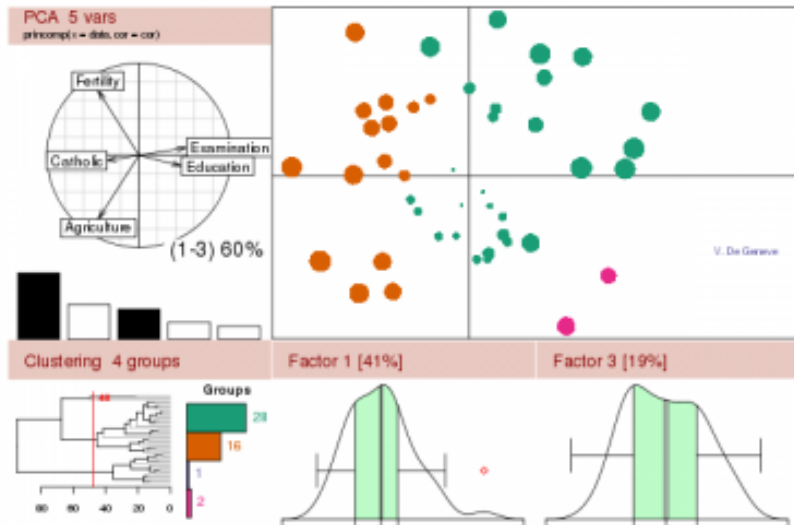


图 1:

Visualization

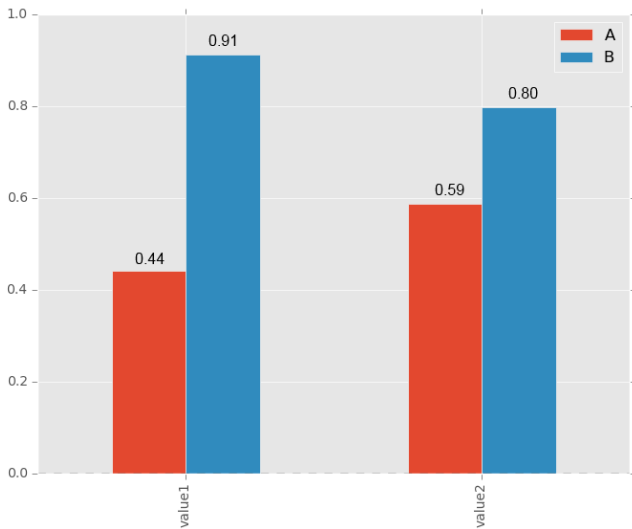


图 2:

R packages

- ▶ graphics: built-in packages (Chapter 3)
- ▶ lattice: implements trellis graphics (Bonus Chapter)
- ▶ ggplot2: R's famous package for making beautiful graphics (Chapter 19)
- ▶ ggvis: Interactive, web based graphics built with the grammar of graphics
- ▶ rgl: Interactive 3D visualizations with R
- ▶ rCharts/recharts/plot_ly
- ▶ ...

content

- ▶ create and save graphs

content

- ▶ create and save graphs
- ▶ customize plot: symbols,lines,colour

content

- ▶ create and save graphs
- ▶ customize plot: symbols, lines, colour
- ▶ annotate with text and titles

content

- ▶ create and save graphs
- ▶ customize plot: symbols,lines,colour
- ▶ annotate with text and titles
- ▶ combined graphs

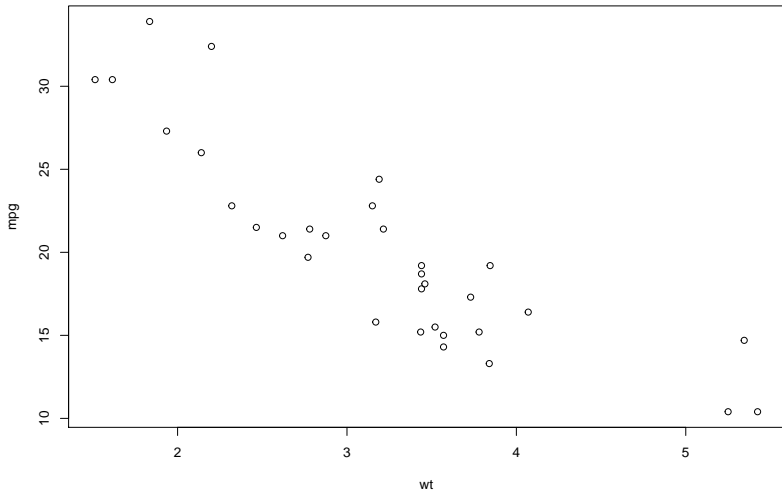
demo

```
demo(graphics)
```

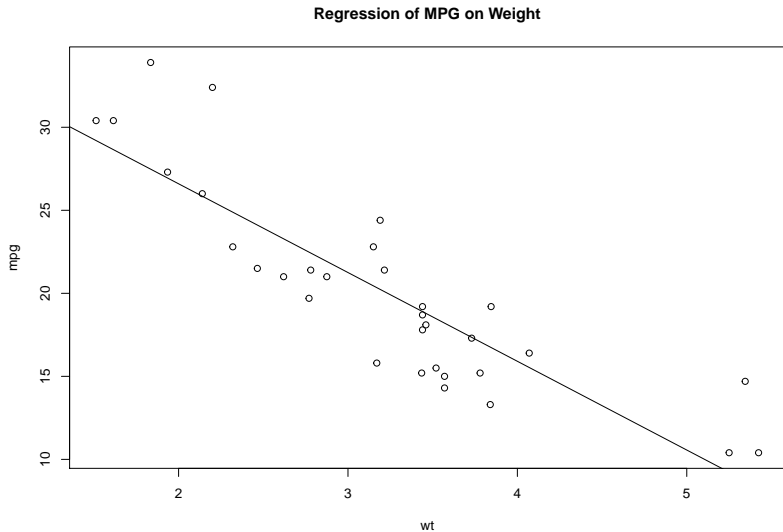
content

- ▶ *create and save graphs*

basic plot



basic plot continued



Export the figure

```
pdf("mtcars_demo.pdf")  
with(mtcars,plot(wt,mpg))  
abline(lm(mpg~wt,data=mtcars))  
title("Regression of MPG on Weight")  
dev.off()
```

Export the figure

- ▶ vector (矢量图): pdf
- ▶ bitmap (位图): png,jpg,bmp,tiff

Function	Output
<code>bmp("filename.bmp")</code>	BMP file
<code>jpeg("filename.jpg")</code>	JPEG file
<code>pdf("filename.pdf")</code>	PDF file
<code>png("filename.png")</code>	PNG file
<code>postscript("filename.ps")</code>	PostScript file

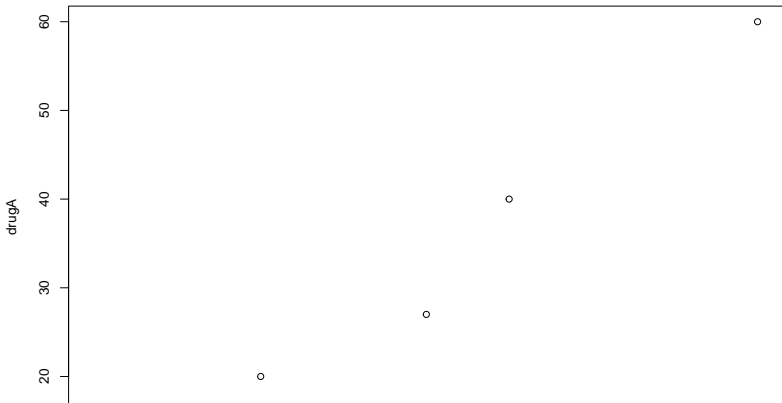
图 3:

content

- ▶ create and save graphs
- ▶ *customize plot: symbols, lines, colour*

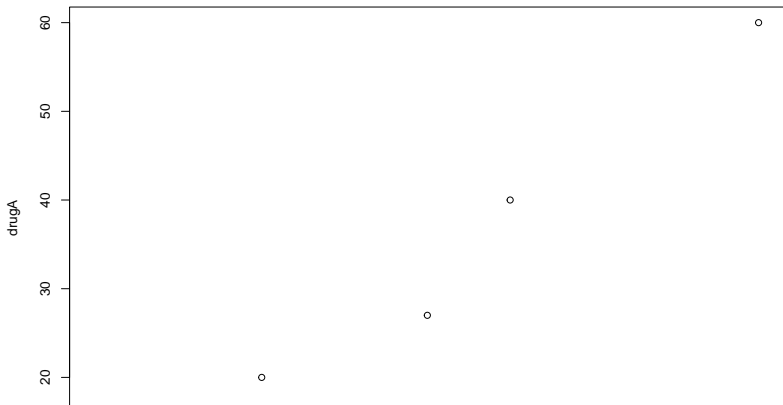
plot: default type: **point**

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
plot(dose, drugA) #default type:point (type='p')
```



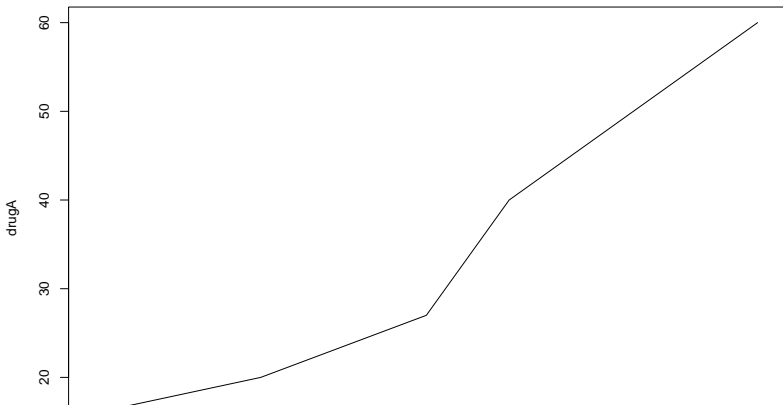
plot:point

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
plot(dose, drugA, type='p')
```



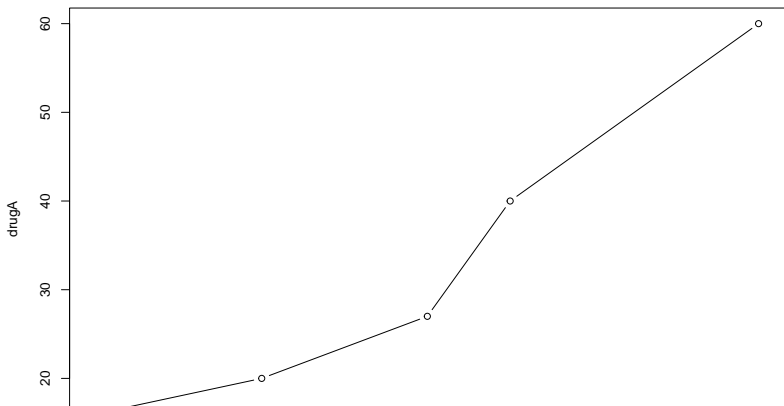
plot:line

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
plot(dose, drugA, type = "l")
```



plot: **point** and **line**

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
plot(dose, drugA, type = "b")
```



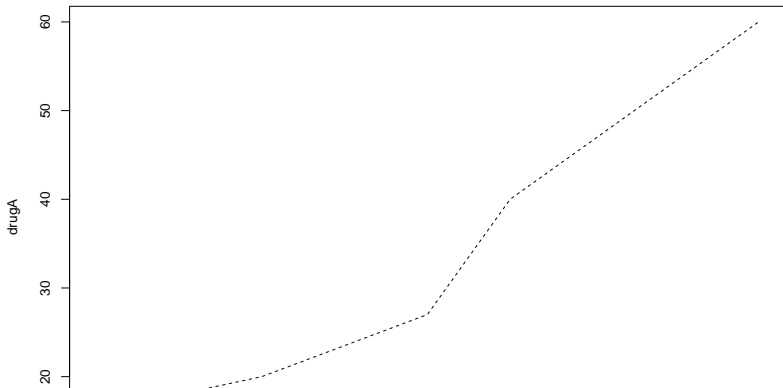
type parameter

- ▶ type parameter

plot:line type and point type

- ▶ line type (type='l'): lty
- ▶ point type (type='p'): pch

```
opar <- par(no.readonly = TRUE)## get the default parameter  
plot(dose, drugA, type = "l", lty = 2, pch = 17)
```



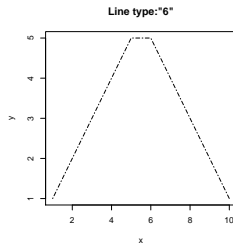
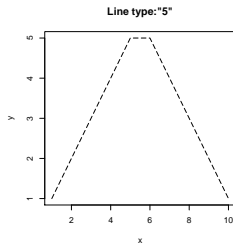
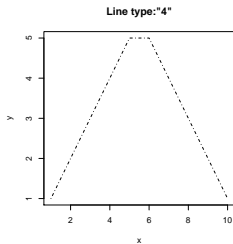
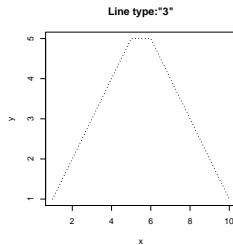
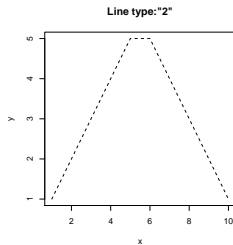
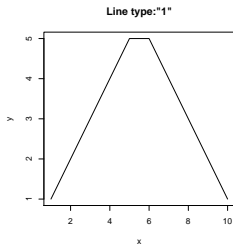
Parameters for symbols and lines

Parameter	Description
<code>pch</code>	Specifies the symbol to use when plotting points (see figure 3.4).
<code>cex</code>	Specifies the symbol size. <code>cex</code> is a number indicating the amount by which plotting symbols should be scaled relative to the default. 1 = default, 1.5 is 50% larger, 0.5 is 50% smaller, and so forth.
<code>lty</code>	Specifies the line type (see figure 3.5).
<code>lwd</code>	Specifies the line width. <code>lwd</code> is expressed relative to the default (1 = default). For example, <code>lwd=2</code> generates a line twice as wide as the default.

图 4:

line parameter

- ▶ line parameter `lty` given `type='l'` in plot function



line parameter:line type

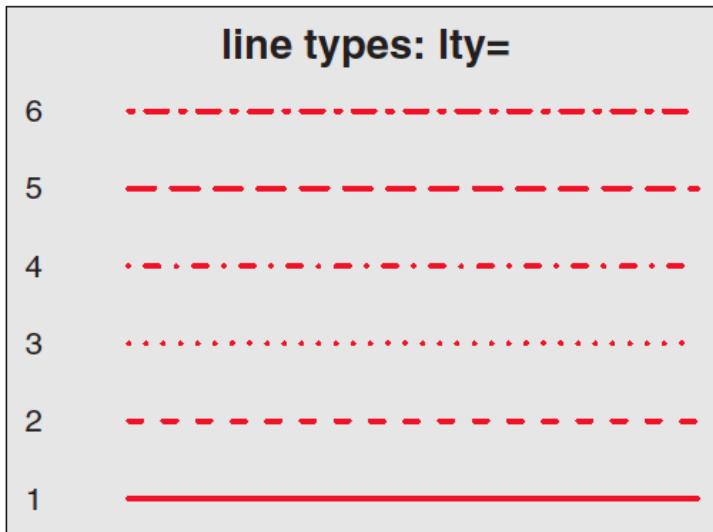
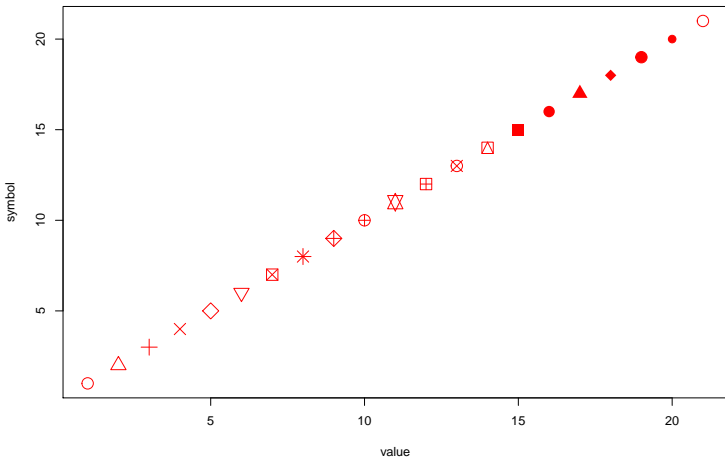


图 5:

symbol parameter: point symbol

- ▶ point symbol parameter given `type='p'` in `plot` function



symbol parameter:point symbol

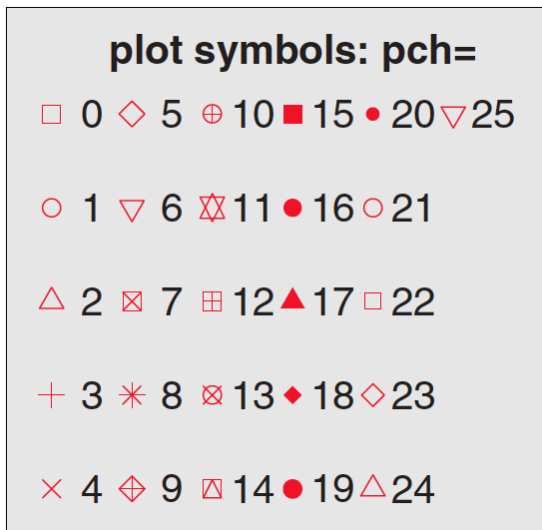
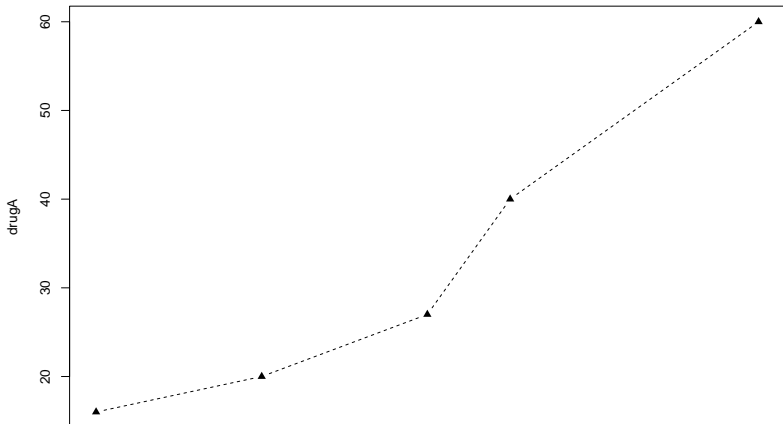


图 6:

plot:line type and point type

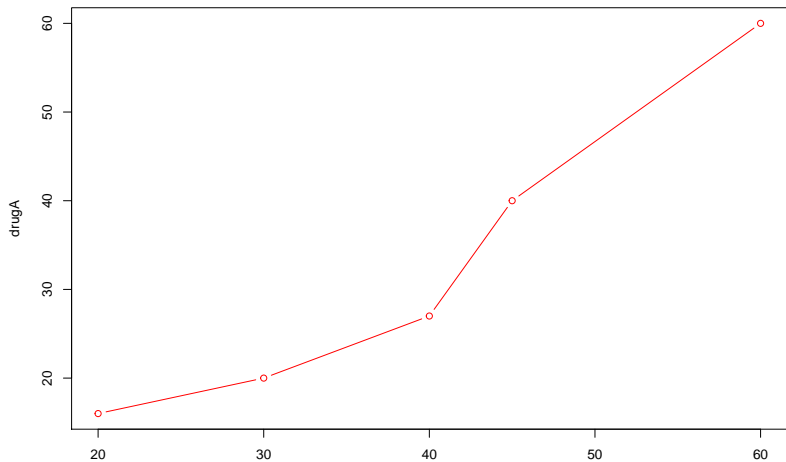
- ▶ line type (type='l'): lty
- ▶ point type (type='p'): pch

```
plot(dose, drugA, type = "b", lty = 2, pch = 17)
```



plot:colour

```
opar <- par(no.readonly = TRUE)  
plot(dose, drugA, type = "b", lty = 1,col='red')
```



plot: colour

► `colors()`

```
length(colors())
```

```
## [1] 657
```

```
colors()[1:5]
```

```
## [1] "white"           "aliceblue"       "antiquewhite"    "ant
```

```
## [5] "antiquewhite2"
```

Palette:Aesthetics.

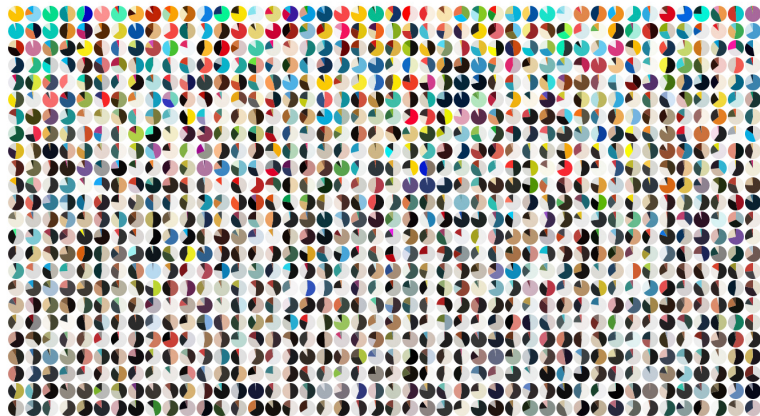
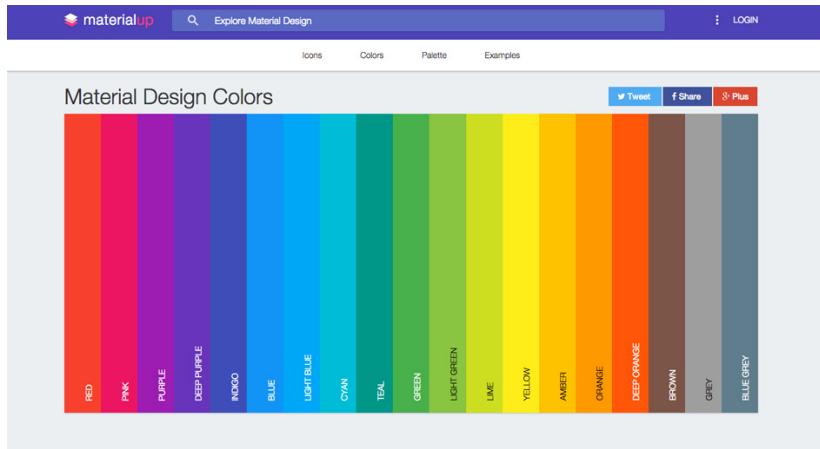


图 7:

Palette:Aesthetics.



8:

Palette:Aesthetics

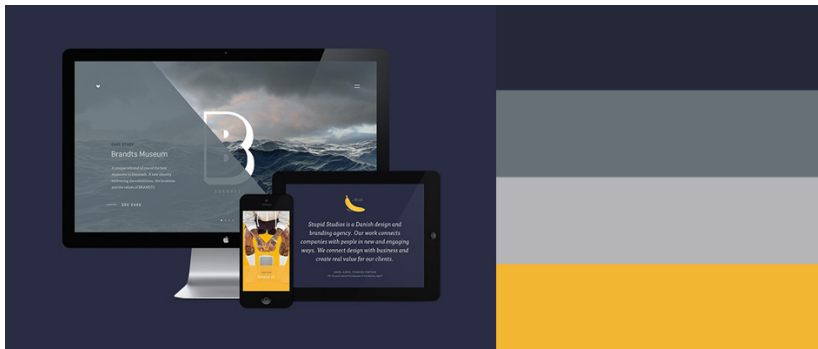
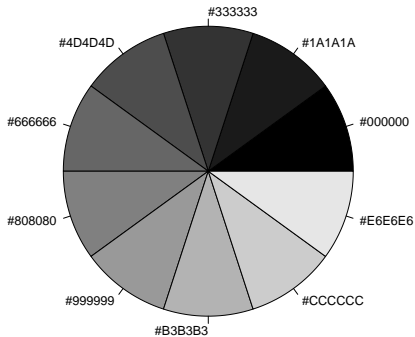


图 9:

► what can we do in R?

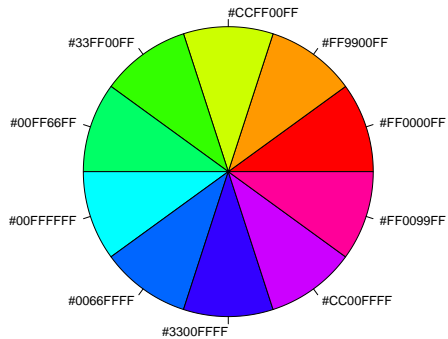
grDevices:gray

► `gray(0:10/10)`



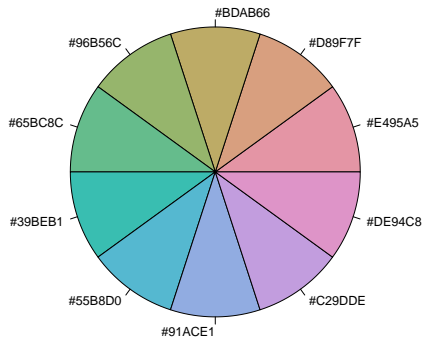
grDevices: rainbow

► `rainbow(10)`

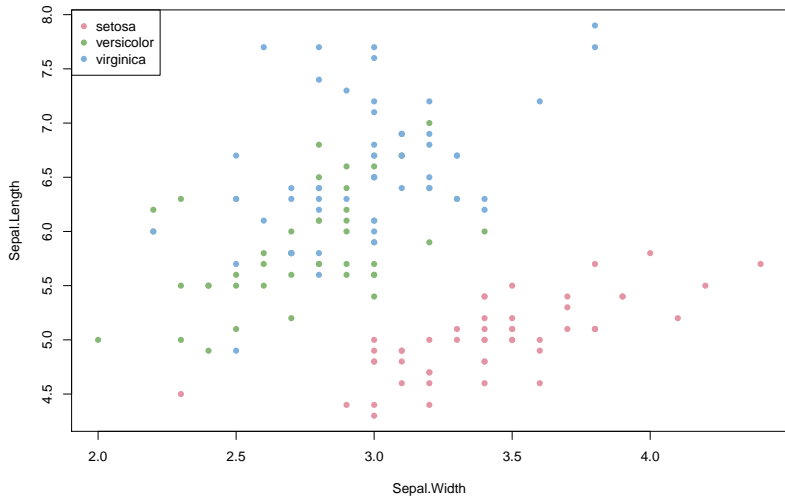


colorspace:rainbow_hcl

► rainbow_hcl(10)



colour example



colour example

```
library(colorspace)
plot(Sepal.Length ~ Sepal.Width, col=
  rainbow_hcl(3)[c(Species)], data=iris, pch=16)
legend("topleft", pch=16, col=
  rainbow_hcl(3), legend=unique(iris$Species))
```

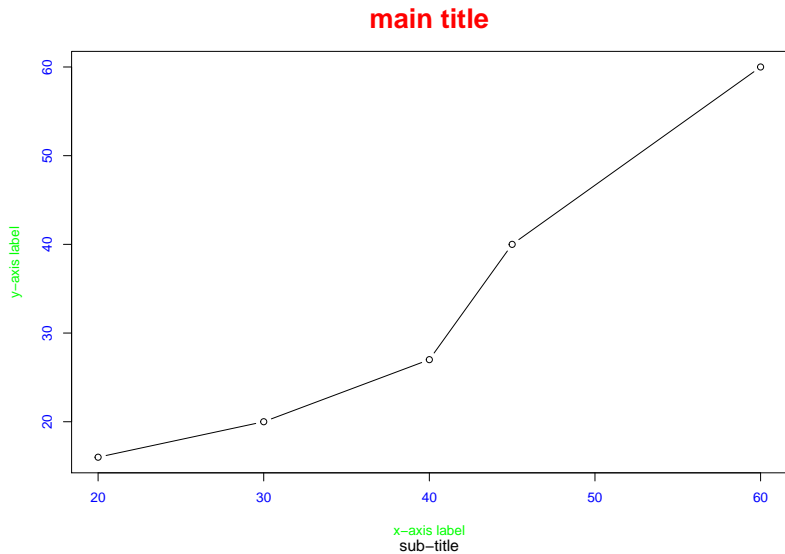
content

- ▶ create and save graphs
- ▶ customize plot: symbols, lines, colour
- ▶ *annotate with text and titles*

text and titles

- ▶ axis texts:axis
- ▶ axis labels:lab
- ▶ titles:main
- ▶ subtitles: sub

title and axis



Colour Parameter

► axis + title

Parameter	Description
<code>col</code>	Default plotting color. Some functions (such as <code>lines</code> and <code>pie</code>) accept a vector of values that are recycled. For example, if <code>col=c("red", "blue")</code> and three lines are plotted, the first line will be red, the second blue, and the third red.
<code>col.axis</code>	Color for axis text.
<code>col.lab</code>	Color for axis labels.
<code>col.main</code>	Color for titles.
<code>col.sub</code>	Color for subtitles.
<code>fg</code>	Color for the plot's foreground.
<code>bg</code>	Color for the plot's background.

Size parameter

► axis + title

Parameter	Description
<code>cex</code>	Number indicating the amount by which plotted text should be scaled relative to the default. 1 = default, 1.5 is 50% larger, 0.5 is 50% smaller, and so on.
<code>cex.axis</code>	Magnification of axis text relative to <code>cex</code> .
<code>cex.lab</code>	Magnification of axis labels relative to <code>cex</code> .
<code>cex.main</code>	Magnification of titles relative to <code>cex</code> .
<code>cex.sub</code>	Magnification of subtitles relative to <code>cex</code> .

Font parameter

► axis + title

Parameter	Description
<code>font</code>	Integer specifying the font to use for plotted text. 1 = plain, 2 = bold, 3 = italic, 4 = bold italic, and 5=symbol (in Adobe symbol encoding).
<code>font.axis</code>	Font for axis text.
<code>font.lab</code>	Font for axis labels.
<code>font.main</code>	Font for titles.
<code>font.sub</code>	Font for subtitles.
<code>ps</code>	Font point size (roughly 1/72 inch). The text size = <code>ps*cex</code> .
<code>family</code>	Font family for drawing text. Standard values are <code>serif</code> , <code>sans</code> , and <code>mono</code> .

parameters for colour,size and font

- ▶ common parameter

	main	sub	lab	axis
col	col.main	col.sub	col.lab	axis.lab
cex	cex.main	cex.sub	cex.lab	cex.axis
font	font.main	font.sub	font.lab	font.axis

figure demo

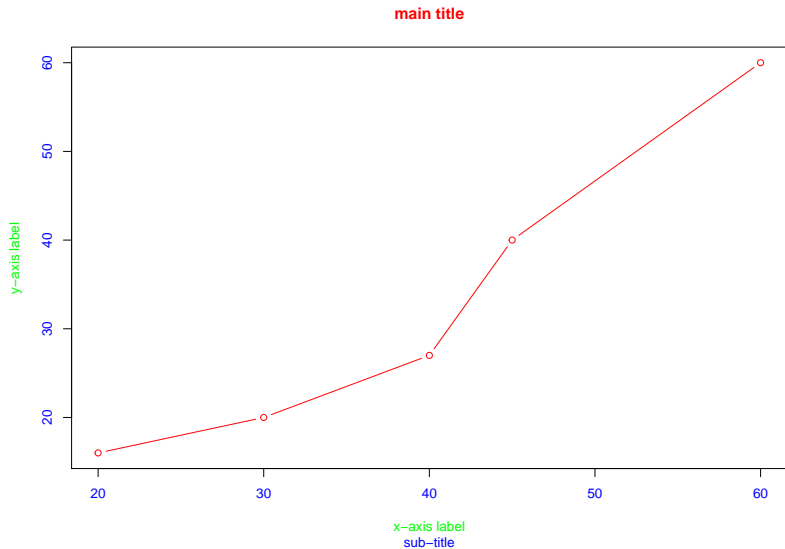
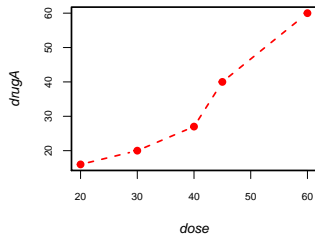


figure demo

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
par(ann=F) ##inhibit x y axis labels but not text.
plot(dose, drugA, type = "b",
      col = "red",col.axis='blue')
# plot(dose, drugA, type = "b",
# col = "red",col.axis='blue',xaxt='n',yaxt='n')
##inhibit axis tick and text
title(main='main title',col.main='red',cex.main=2,
sub='sub-title',col.sub='blue',
xlab='x-axis label',
ylab='y-axis label',col.lab='green',cex.lab=1)
```


Example

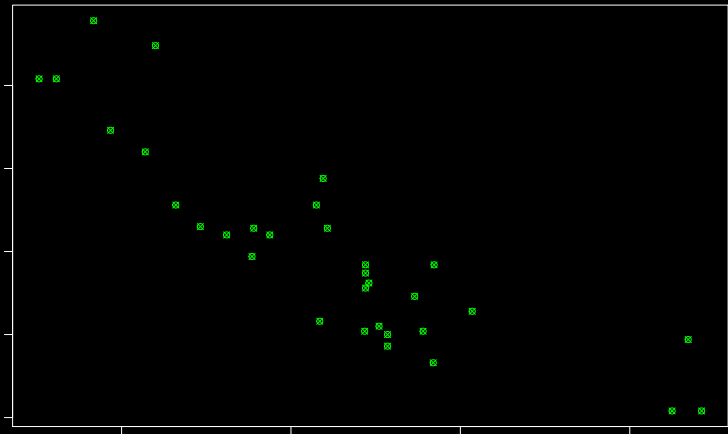


Example

```
opar <- par(no.readonly = TRUE)
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
par(font.lab=3, cex.lab=1,
     font.main=4, cex.main=2)
par(pin = c(3, 2))
##The current plot dimensions,
par(lwd = 2, cex = 1)
##magnified index of plotted text and symbols
par(cex.axis = 0.75, font.axis = 1)
plot(dose, drugA, type = "b", pch = 19,
     lty = 2, col = "red") ##pch plot symbols
```

Foreground and background

```
par(bg='black',fg='white')  
with(mtcars,plot(wt,mpg,col='green',pch=13))
```

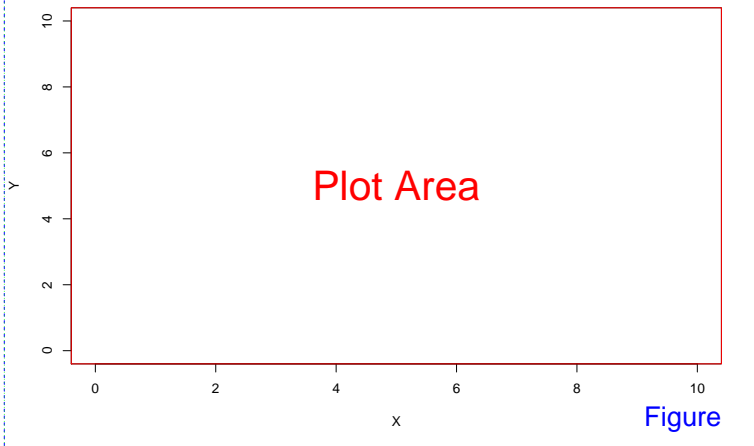


PRACTICE!

Margin and outer margin

```
oma = c(2,2,2,2)
```

```
mar = c(5.1,4.1,4.1,2.1)
```



Outer Margin Area


```
par(opar)  
par('pin')
```

```
## [1] 8.76 5.16
```

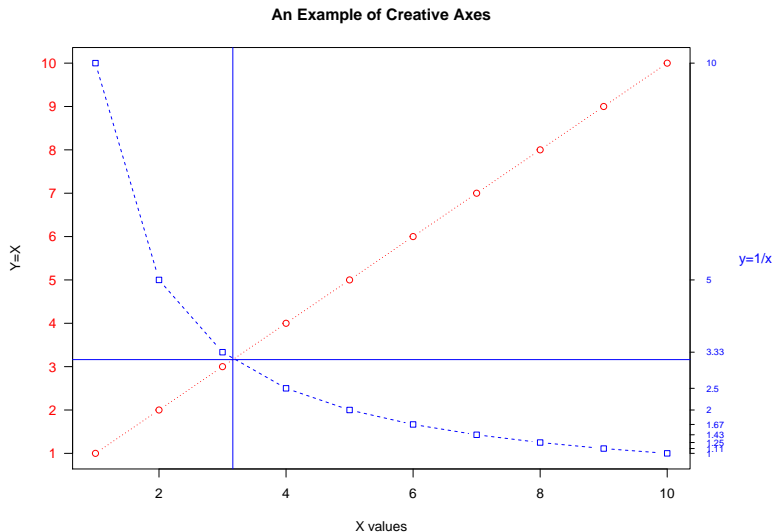
```
par('mar')
```

```
## [1] 5.1 4.1 4.1 2.1
```

```
par('oma')
```

```
## [1] 0 0 0 0
```

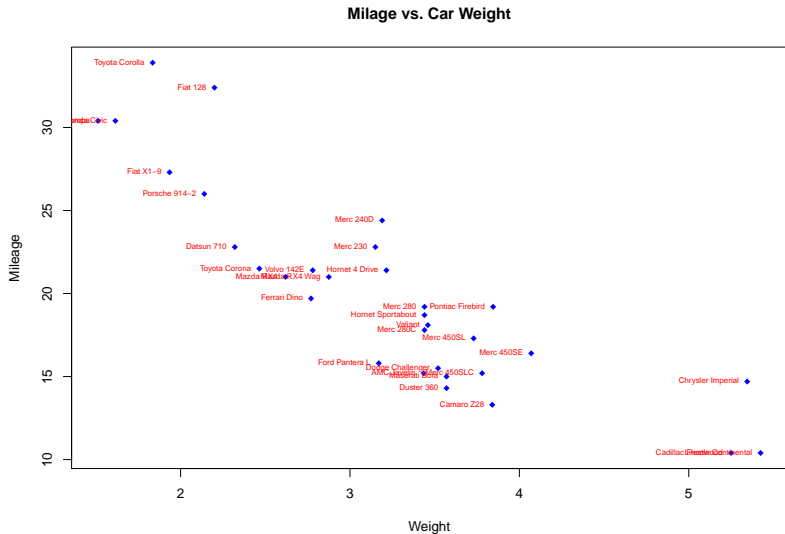
An Example of two y Axes



An Example of two y Axes

```
x <- c(1:10); y <- x; z <- 10/x
par(mar = c(5, 4, 4, 8) + 0.1)
plot(x, y, type = "b", lty = 3, pch = 21, col = "red",
     yaxt = "n", ann = FALSE)
lines(x, z, type = "b", pch = 22, col = "blue", lty = 2)
axis(side=2, at = x, labels = x, col.axis = "red",
     las = 1)
axis(side=4, at = z, labels = round(z, digits = 2),
     col.axis = "blue", las = 2, cex.axis = 0.7, tck = -0.01)
mtext("y=1/x", side = 4, line = 3, cex.lab = 1, las = 2,
     col = "blue")
title("An Example of Creative Axes", xlab = "X values",
     ylab = "Y=X")
abline(h=sqrt(10), lty=1, col='blue')
abline(v=sqrt(10), lty=1, col='blue')
```

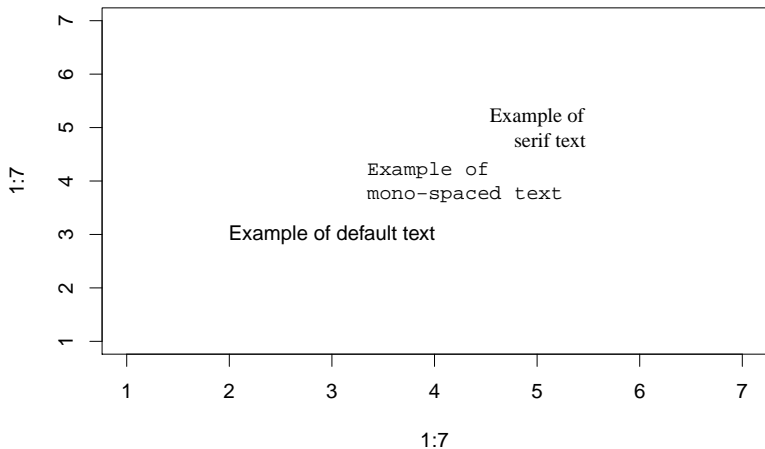
Example of labeling points



Example of labeling points

```
with(mtcars,plot(wt, mpg,main = "Milage vs. Car Weight",  
                xlab = "Weight",  
                ylab = "Mileage", pch = 18, col = "blue"))  
text(mtcars$wt, mtcars$mpg, row.names(mtcars),  
     cex = 0.6, pos = 2,  
     col = "red")
```

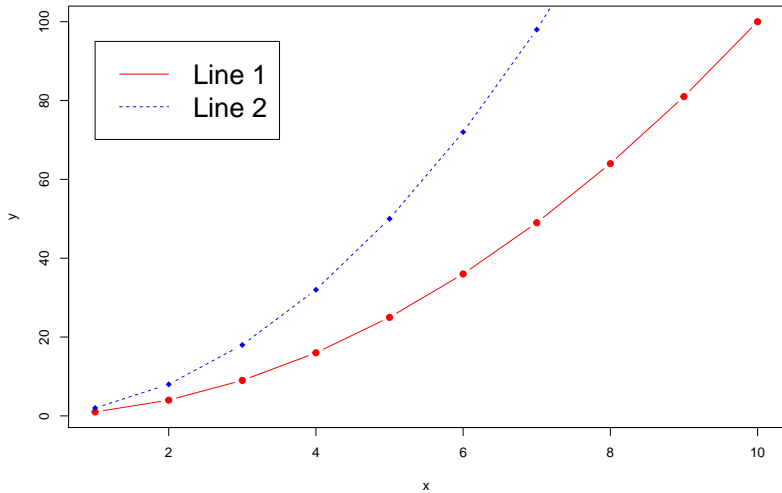
View font families



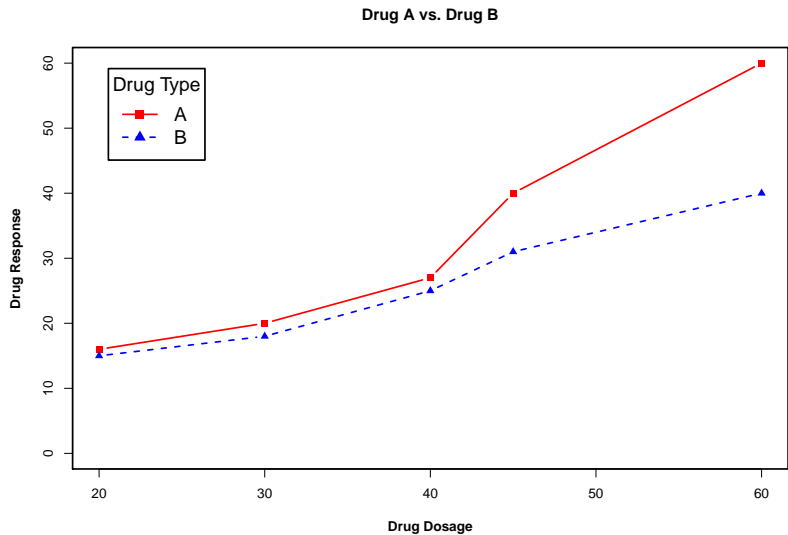
View font families

```
opar <- par(no.readonly = TRUE)
par(cex = 1.5)
plot(1:7, 1:7, type = "n")
text(3, 3, "Example of default text")
text(4, 4, family = "mono", "Example of
      mono-spaced text")
text(5, 5, family = "serif", "Example of
      serif text")
par(opar)
```

Legend



Legend



Legend

```
dose <- c(20, 30, 40, 45, 60)
drugA <- c(16, 20, 27, 40, 60)
drugB <- c(15, 18, 25, 31, 40)
opar <- par(no.readonly = TRUE)
par(lwd = 2, cex = 1, font.lab = 2)
plot(dose, drugA, type = "b", pch = 15, lty = 1,
      col = "red",
      ylim = c(0, 60), main = "Drug A vs. Drug B",
      xlab = "Drug Dosage",
      ylab = "Drug Response")
lines(dose, drugB, type = "b", pch = 17, lty = 2,
      col = "blue")
# abline(h = c(30), lwd = 3, lty = 2, col = "green")
legend("topleft", inset = 0.05, title = "Drug Type",
      c("A", "B"), lty = c(1, 2), pch = c(15, 17),
      col = c("red", "blue"), cex=1)
par(opar)
```


Legend position

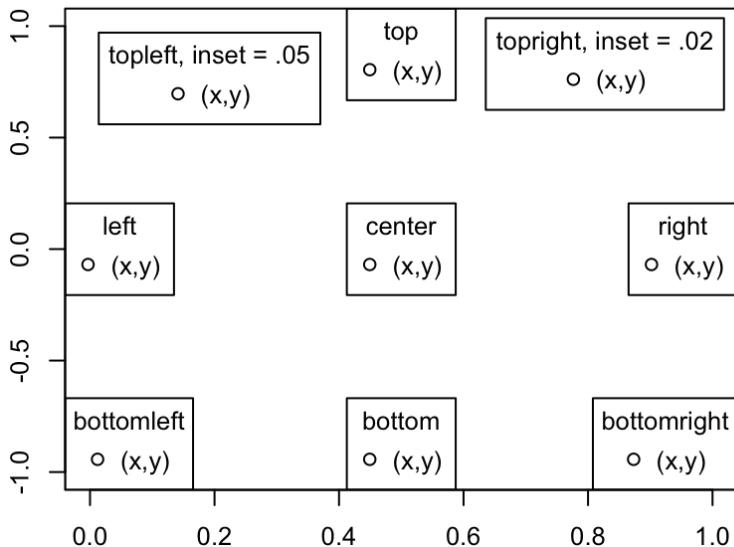


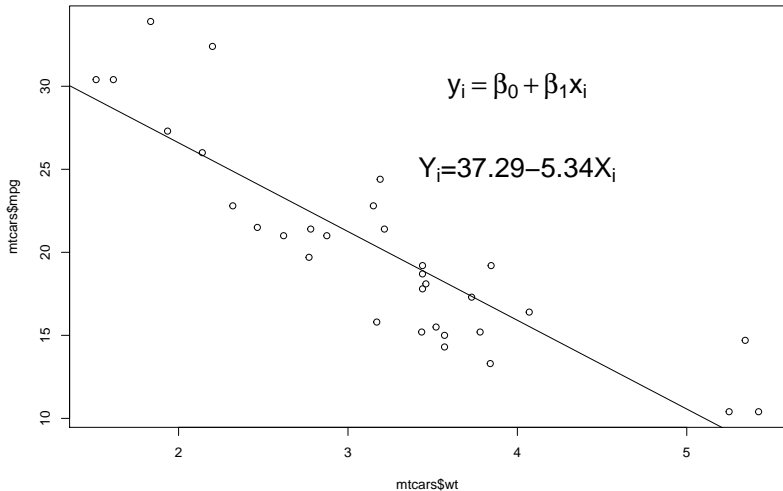
图 11:

Equation notation

► demo(plotmath)

Arithmetic Operators		Lists	
$x + y$	$x + y$	<code>list(x, y, z)</code>	x, y, z
$x - y$	$x - y$	Relations	
$x * y$	xy	$x == y$	$x = y$
x/y	x/y	$x != y$	$x \neq y$
$x \% + - \% y$	$x \pm y$	$x < y$	$x < y$
$x \% / \% y$	$x \div y$	$x <= y$	$x \leq y$
$x \% * \% y$	$x \times y$	$x > y$	$x > y$
$x \% . \% y$	$x \cdot y$	$x >= y$	$x \geq y$
$-x$	$-x$	$x \% \sim \% y$	$x \approx y$
$+x$	$+x$	$x \% == \% y$	$x \equiv y$
Sub/Superscripts		$x \% == \% y$	$x \equiv y$
$x[i]$	x_i	$x \% \text{prop} \% y$	$x \propto y$
$x^{\wedge}2$	x^2	$x \% \sim \% y$	$x \sim y$
Juxtaposition		Typeface	
$x * y$	xy	<code>plain(x)</code>	x
<code>paste(x, y, z)</code>	xyz	<code>italic(x)</code>	x
Radicals		<code>bold(x)</code>	\mathbf{x}
<code>sqrt(x)</code>	\sqrt{x}	<code>bolditalic(x)</code>	\mathbf{x}
<code>sqrt(x, y)</code>	$\sqrt[y]{x}$	<code>underline(x)</code>	\underline{x}

Equation notation



Equation notation

```
with(mtcars,plot(wt, mpg))
abline(lm(mpg~wt,data=mtcars))
result<-summary(lm(mpg~wt,data=mtcars))
text(4.2, 30, expression(Y[i] == beta[0] + beta[1]*X[i]))

text(4.2,25,
      substitute(
        paste(y[i], "=", beta0, beta1,x[i]),
        list(beta0 = round(result$coe[1,1],2),
              beta1 = round(result$coe[2,1],2))))
```

content

- ▶ create and save graphs
- ▶ customize plot: symbols, lines, colour
- ▶ annotate with text and titles
- ▶ *combined graphs*

Margin and outer margin

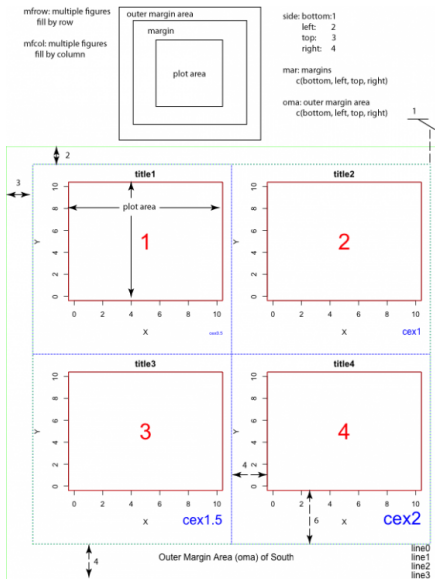
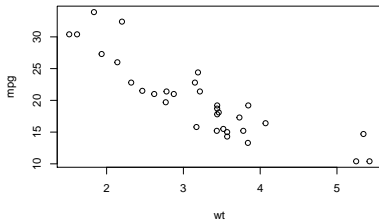


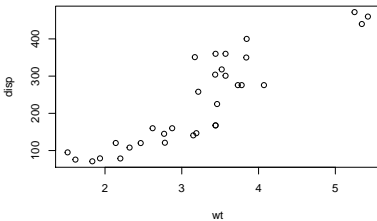
图 13:

Combine graphs

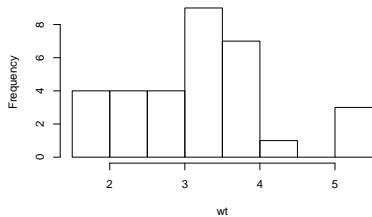
Scatterplot of wt vs. mpg



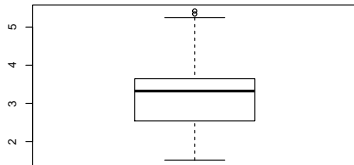
Scatterplot of wt vs disp



Histogram of wt



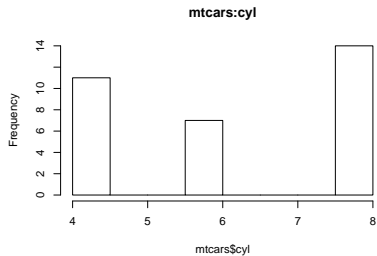
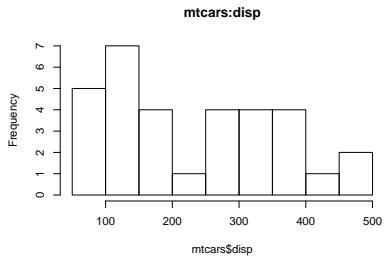
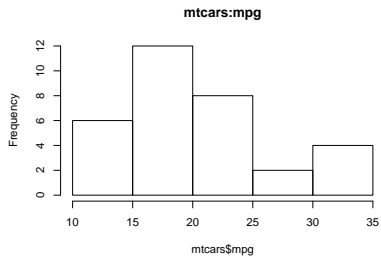
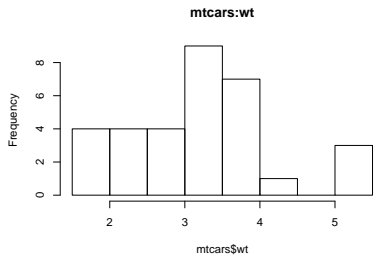
Boxplot of wt



Combine graphs

```
opar <- par(no.readonly = TRUE)
attach(mtcars)
par(mfrow = c(2,2))
plot(wt, mpg, main = "Scatterplot of wt vs. mpg")
plot(wt, disp, main = "Scatterplot of wt vs disp")
hist(wt, main = "Histogram of wt")
boxplot(wt, main = "Boxplot of wt")
detach(mtcars)
par(opar)
```

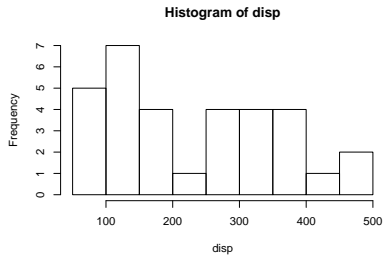
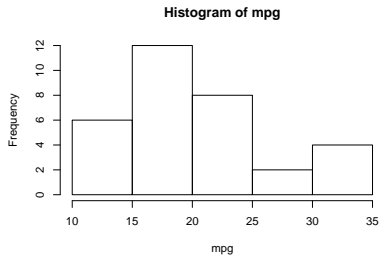
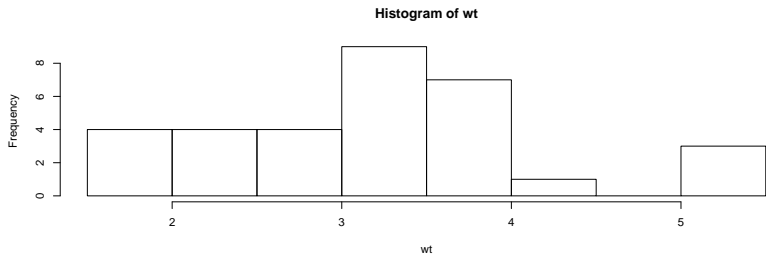

Combine



Combine

```
opar <- par(no.readonly = TRUE)
par(mfrow = c(2, 2))
hist(mtcars$wt,main='mtcars:wt');
hist(mtcars$mpg,main='mtcars:mpg')
hist(mtcars$disp,main='mtcars:disp')
hist(mtcars$cyl,main='mtcars:cyl')
par(opar)
```

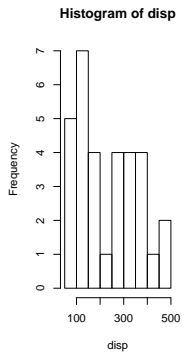
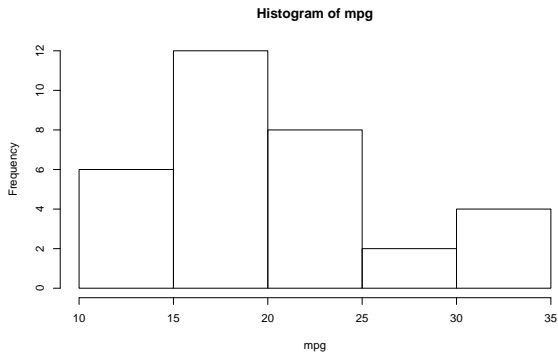
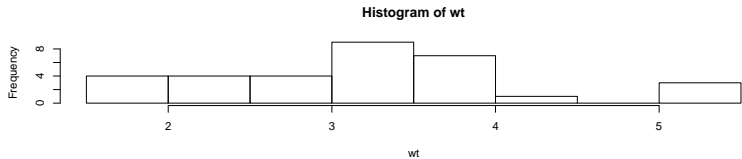
Layout



Layout

```
attach(mtcars)
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
hist(wt)
hist(mpg)
hist(displ)
detach(mtcars)
```

Proportion of figures

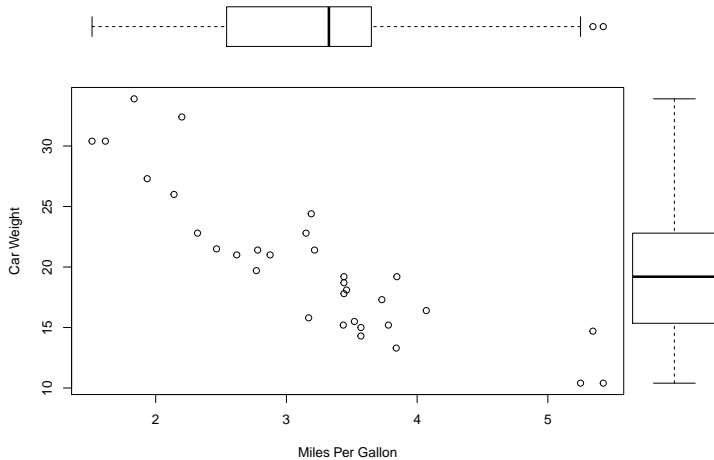


Proportion of figures

```
par(mar=c(5.1,4.1,4.1,2))
par(oma=c(0,0,0,2))
attach(mtcars)
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE),
widths = c(3, 1), heights = c(1, 2))
#3:1 for heights 1:2 for widths
hist(wt)
hist(mpg)
hist(displ)
detach(mtcars)
```

Fine placement of figures

Enhanced Scatterplot



Fine placement of figures

```
opar <- par(no.readonly = TRUE)
par(fig = c(0, 0.8, 0, 0.8))
plot(mtcars$wt, mtcars$mpg, xlab = "Miles Per Gallon",
      ylab = "Car Weight")
par(fig = c(0, 0.8, 0.55, 1), new = TRUE)
boxplot(mtcars$wt, horizontal = TRUE, axes = FALSE)
par(fig = c(0.6, 1, 0, 0.8), new = TRUE)
boxplot(mtcars$mpg, axes = FALSE)
mtext("Enhanced Scatterplot", side = 3, outer = TRUE,
      line = -2) ##side=3 top
par(opar)
```


content

- ▶ create and save graphs
- ▶ customize plot: symbols,lines,colour
- ▶ annotate with text and titles
- ▶ combined graphs

	main	sub	lab	axis
col	col.main	col.sub	col.lab	axis.lab
cex	cex.main	cex.sub	cex.lab	cex.axis
font	font.main	font.sub	font.lab	font.axis

图 14: