Data Analysis Using R: Chapter08

罗智超 (ROKIA.ORG)

1 通过本章你将学会

- 创建图形
- 图形参数

2 图形输出

```
# 1 inch =2.5cm
# 1 foot = 12 inch
# 1 yard = 3 foot
# default graphic 6*6 inch
attach(mtcars)
pdf("mygraph3.pdf",width=6,height=3)
plot(wt,mpg)
abline(lm(mpg~wt))
title("Regression of MPG on Weight")
detach(mtcars)
dev.off()

# use dev.new() can create a new graphic
# use win.metafile() png() jpeg() bmp() tiff() xfig() postscript()
# How to output pdf with chinese characters
pdf(family='GB1',file='TestChiese.pdf' )
```

```
plot(1:10,1:10,main='Test chinese output')
dev.off()
```

3

3.1 Graphic Parameters

Parameter	Description
pch	Specifies the symbol to use when plotting points
cex	Specifies the symbol size.
lty	Specifies the line type.
lwd	Specifies the line width.

```
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")

#par()
opar <- par(no.readonly = TRUE)
par(lty = 2, pch = 17,bg= "#FFFF40FF")
plot(dose, drugA, type = "b")



plot(dose, drugA, type = "b", lty = 2, pch = 17)
plot(dose, drugA, type = "b", lyt = 3, lwd = 3, pch = 15, cex = 2)
par(opar)
```

4

4.1 Colour Parameters

Parameter	Description
col	Default plotting color.
col.axis	Color for axis text.
col.lab	Color for axis labels.
col.main	Color for titles.
col.sub	Color for subtitles.
fg	The plot??s foreground color.
bg	The plot??s background color.

```
#col=1
#col="white"
#col="#FFFFFF"
#col=rgb(1,1,1)
#col=hsv(0,0,1)
#Followed functions can create continued colour
#rainbow(),heat.colors(),terrain.colors(),topo.colors(),cm.colors(),gray()

n <- 24
mycolors <- terrain.colors(n)
pie(rep(1, n), labels = mycolors, col = mycolors)
mygrays <- gray(0:n/n)
pie(rep(1, n), labels = mygrays, col = mygrays)
```

5

5.1 Text Size Parameters

Parameter	Description
cex	1=default, 1.5 is 50% larger,etc.
cex.axis	Magnification of axis text relative to cex.
cex.lab	Magnification of axis labels relative to cex.
cex.main	Magnification of titles relative to cex.
cex.sub	Magnification of subtitles relative to cex.

6

6.1 Font family, size, and style

Parameter	Description
font	Integer specifying font to use for plotted text.
font.axis	Font for axis text.
font.lab	Font for axis labels.
font.main	Font for titles.
font.sub	Font for subtitles.
ps	Font point size (roughly 1/72 inch).The text size = ps*cex. family | Font family for drawin

```
windowsFonts(
    A=windowsFont("Arial Black"),
    B=windowsFont("Bookman Old Style"),
    C=windowsFont("Comic Sans MS"))
# On aMac, use quartzFonts() instead

par(family="A")
pdf(file="myplot.pdf",family="fontname")
```

7

7.1 Graph and margin dimensions

Parameter	Description
pin	Plot dimensions (width, height) in inches.
mai	Numerical vector indicating margin size in inches.
mar	Numerical vector indicating margin size in lines.

```
par(pin=c(4,3), mai=c(1,.5, 1, .2))
#c(bottom, left, top, right)
#The default mar is c(5, 4, 4, 2) + 0.1.

opar <- par(no.readonly = TRUE)
par(pin = c(2, 3))
par(lwd = 1, cex = 1)
par(cex.axis = 1, font.axis = 1)
plot(dose, drugA, type = "b", pch = 19, lty = 2, col = "red")
plot(dose, drugB, type = "b", pch = 23, lty = 6, col = "blue",
    bg = "green")
par(opar)

# mar demo

pdf(file='plot.pdf', width=10, height=10)
par(mfrow=c(10,10), mar=c(1,1,1,1))
for(i in 1:100){plot(rnorm(i))}
dev.off()

# reset par
resetPar <- function() {
    dev.new()
```

```
    op <- par(no.readonly = TRUE)
    dev.off()
    op
}
resetPar()

par()
```

8

8.1 Adding text, customized axes, and legends

```
plot(dose, drugA, type="b", col="red",
     lty=2, pch=2, lwd=2,
     main="Clinical Trials for Drug A",
     sub="This is hypothetical data",
     xlab="Dosage", ylab="Drug Response",
     xlim=c(0, 60), ylim=c(0, 70))

title(main="My Title", col.main="red",
      sub="My Sub-title", col.sub="blue",
      xlab="My X label", ylab="My Y label",
      col.lab="green", cex.lab=0.75)
```

9

9.1 Axis options

Parameter	Description
side	An integer indicating the side of the graph to draw the axis (1=bottom, 2=left, 3=top, 4=

Parameter	Description
at	A numeric vector indicating where tick marks should be drawn.
labels	A character vector of labels to be placed at the tick marks (if NULL, the at values will be
pos	The coordinate at which the axis line is to be drawn (that is, the value on the other axis w
lty	Line type.
col	The line and tick mark color.
las	Labels are parallel (=0) or perpendicular (=2) to the axis.
tck	Length of tick mark as a fraction of the plotting region (a negative number isoutside the gr

```r
#Useful
x <- c(1:10)
y <- x
z <- 10/x
opar <- par(no.readonly = TRUE)

par(mar = c(5, 4, 4, 8) + 0.1)
plot(x, y, type = "b", pch = 21, col = "red", yaxt = "n",
    lty = 3, ann = FALSE)
lines(x, z, type = "b", pch = 22, col = "blue", lty = 2)
axis(2, at = x, labels = x, col.axis = "red", las = 2)
axis(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")
par(opar)

#Use library(Hmisc) minor.tick() function can create sub axis

library(Hmisc)
minor.tick(nx=2,ny=3,tick.ratio=0.5)
```

10

10.1 Reference lines

```
# h:horizon, v:vertical
abline(h=yvalues, v=xvalues)
abline(v=seq(1, 10, 2), lty=2, col="blue")
```

11

11.1 Legend

```
#location:bottom, bottomleft, left,topleft, top, topright, right, bottomright, or cente

opar <- par(no.readonly = TRUE)
par(lwd = 2, cex = 1.5, 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 = 1.5, lty = 2, col = "grey")
library(Hmisc)
minor.tick(nx = 3, ny = 3, tick.ratio = 0.5)
legend("topleft", inset = 0.05,
       title = "Drug Type",
       c("A", "B"),
       lty = c(1, 2),
       pch = c(15, 17),
```

```
       col = c("red", "blue"))
par(opar)
```

12

12.1 Text annotations

- Text can be added to graphs using the text() and mtext() functions.
- text() placestext within the graph whereas mtext() places text in one of the four margins.
- text(location, "text to place", pos, ...)
- mtext("text to place", side, line=n, ...)

```
attach(mtcars)
plot(wt, mpg,main="Mileage vs. Car Weight",
     xlab="Weight", ylab="Mileage",
     pch=18, col="blue")
text(wt, mpg,row.names(mtcars),
     cex=0.6, pos=4, col="red")
detach(mtcars)

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)
```

13

13.1 Combining graphs

- R makes it easy to combine several graphs into one overall graph, using either the par() or layout() function.

```
attach(mtcars)
opar <- par(no.readonly = TRUE)
par(pin=c(4,3), mai=c(1,.5, 1, .2))
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")
par(opar)
detach(mtcars)
```

- The layout() function has the form layout(mat) where mat is a matrix object speci-fying the location of the multiple plots to combine.

```
attach(mtcars)
layout(matrix(c(1, 1, 2, 3),
              2, 2, byrow = TRUE),
       widths=c(3, 1),
       heights=c(1, 2))
hist(wt)
hist(mpg)
hist(disp)
detach(mtcars)
#widths = a vector of values for the widths of columns
#heights = a vector of values for the heights of rows
```

14

14.1 Creating a figure arrangement with fine control

- There are times when you want to arrange or superimpose several figures to createa single meaningful plot. Doing so requires fine control over the placement of the figures. You can accomplish this with the fig= graphical parameter.

```
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$wt, horizontal = TRUE, axes = FALSE)
par(fig = c(0.65, 1, 0, 0.8), new = TRUE)
boxplot(mtcars$mpg, axes = FALSE)
mtext("Enhanced Scatterplot", side = 3, outer = TRUE,
    line = -3)
par(opar)
```

15 This Chapter include

- Bar
- Pie
- Histogram
- Kernel Density
- Boxplot

16 BarPlot

- barplot()

```
library(vcd)
counts<-table(Arthritis$Improved)
counts
par(mfrow=c(1,2))

barplot(counts,main="Simple Bar Plot",
        xlab="Improvement", ylab="Frequency")
barplot(counts,main="Simple Bar Plot",
        xlab="Improvement", ylab="Frequency",horiz=TRUE)
dev.off()
```

- If the categorical variable to be plotted is a factor or ordered factor,you can create a vertical bar plot quickly with the plot() function .

```
par(mfrow=c(1,2))
plot(Arthritis$Improved, main="Simple Bar Plot",
     xlab="Improved", ylab="Frequency")
plot(Arthritis$Improved, horiz=TRUE,
     main="Horizontal Bar Plot",
     xlab="Frequency", ylab="Improved")
dev.off()
```

- Stacked and grouped bar plot

```
counts <- table(Arthritis$Improved, Arthritis$Treatment)
counts
barplot(counts, main="Stacked Bar Plot",
        xlab="Treatment", ylab="Frequency",
        col=c("red", "yellow","green"),
        legend=rownames(counts))
```

```
barplot(counts,  main="Grouped Bar Plot",
        xlab="Treatment", ylab="Frequency",
        col=c("red", "yellow", "green"),
        legend=rownames(counts), beside=TRUE)

#beside=TRUE
```

17 Bie Charts

```
par(mfrow = c(2, 2))
dev.off()
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")

pie(slices, labels = lbls, main = "Simple Pie Chart")

pct <- round(slices/sum(slices) * 100)
lbls2 <- paste(lbls, " ", pct, "%", sep = "")
pie(slices, labels = lbls2,
    col = rainbow(length(lbls)),
    main = "Pie Chart with Percentages")

library(plotrix)
pie3D(slices, labels = lbls,
      explode = 0.1, main = "3D Pie Chart ")

mytable <- table(state.region)
lbls <- paste(names(mytable), "\n", mytable, sep = "")

pie(mytable, labels = lbls,
    main = "Pie Chart from a Table \n (with sample sizes)")
```

```
dev.off()

#Pie Charts are not recommended usually.
library(plotrix)
slices <- c(10, 12,4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
fan.plot(slices, labels = lbls, main="Fan Plot")
```

18 Histograms

- hist(x)

```
par(mfrow = c(2, 2))

hist(mtcars$mpg)

hist(mtcars$mpg, breaks = 12, col = "red",
    xlab = "Miles Per Gallon",
    main = "Colored histogram with 12 bins")

hist(mtcars$mpg, freq = FALSE,
     breaks = 12, col = "red",
    xlab = "Miles Per Gallon",
    main = "Histogram, rug plot, density curve")
rug(jitter(mtcars$mpg))
lines(density(mtcars$mpg), col = "blue", lwd = 2)

# Histogram with Superimposed Normal Curve
# (Thanks to Peter Dalgaard)
x <- mtcars$mpg
h <- hist(x, breaks = 12, col = "red",
    xlab = "Miles Per Gallon",
```

```
    main = "Histogram with normal curve and box")
xfit <- seq(min(x), max(x), length = 40)
yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
yfit <- yfit * diff(h$mids[1:2]) * length(x)
lines(xfit, yfit, col = "blue", lwd = 2)
box()
```

19 Kernel density plots

- plot(density(x))

```
par(mfrow = c(2, 1))
d <- density(mtcars$mpg)
d
plot(d)

d <- density(mtcars$mpg)
plot(d, main = "Kernel Density of Miles Per Gallon")
polygon(d, col = "red", border = "blue")
rug(mtcars$mpg, col = "brown")
```

20 Box plots

- boxplot()

```
dev.off()
boxplot(mpg ~ cyl, data=mtcars,
        main="Car Mileage Data",
        xlab="Number of Cylinders",
        ylab="Miles Per Gallon")
```

```
boxplot(mpg ~ cyl,
        data=mtcars,
        notch=TRUE,
        varwidth=TRUE,
        col="red",
        main="Car Mileage Data",
        xlab="Number of Cylinders",
        ylab="Miles Per Gallon")
```

- Box plots for two crossed factors

```
mtcars$cyl.f <- factor(mtcars$cyl, levels = c(4, 6,
    8), labels = c("4", "6", "8"))

mtcars$am.f <- factor(mtcars$am,
                      levels = c(0, 1),
                      labels = c("auto", "standard"))

#mpg ~ am.f * cyl.f
boxplot(mpg ~ am.f * cyl.f, data = mtcars,
    varwidth = TRUE, col = c("gold", "darkgreen"),
    main = "MPG Distribution by Auto Type",
    xlab = "Auto Type")
```

21 Dot plots

- Dot p lots provide a method of plotting a large number of labeled values on a simplehorizontal scale.
- dotchart(x, labels=)

```
dotchart(mtcars$mpg, labels = row.names(mtcars),
    cex = 0.7,
    main = "Gas Milage for Car Models",
    xlab = "Miles Per Gallon")

# Sorted colored grouped dot chart

x <- mtcars[order(mtcars$mpg), ]
x$cyl <- factor(x$cyl)
x$color[x$cyl == 4] <- "red"
x$color[x$cyl == 6] <- "blue"
x$color[x$cyl == 8] <- "darkgreen"
dotchart(x$mpg, labels = row.names(x), cex = 0.7,
    pch = 19, groups = x$cyl,
    gcolor = "black", color = x$color,
    main = "Gas Milage for Car Models\ngrouped by cylinder",
    xlab = "Miles Per Gallon")
```