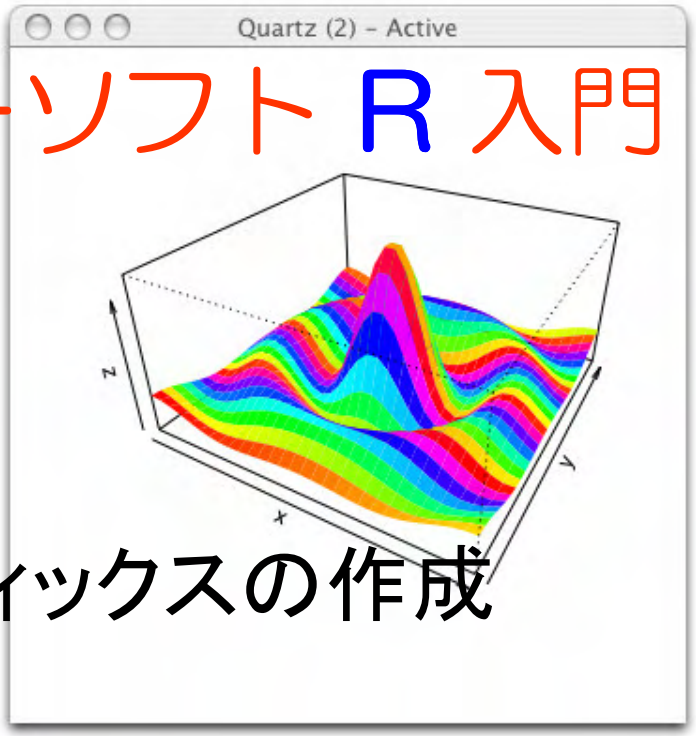
統計解析フリーソフト R 入門

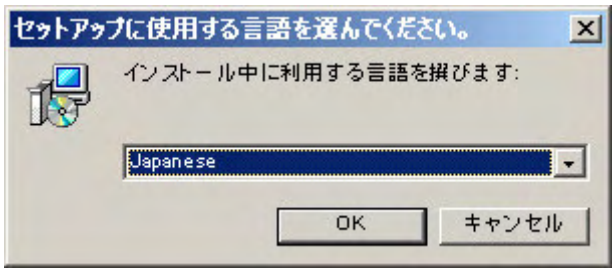
R によるグラフィックスの作成



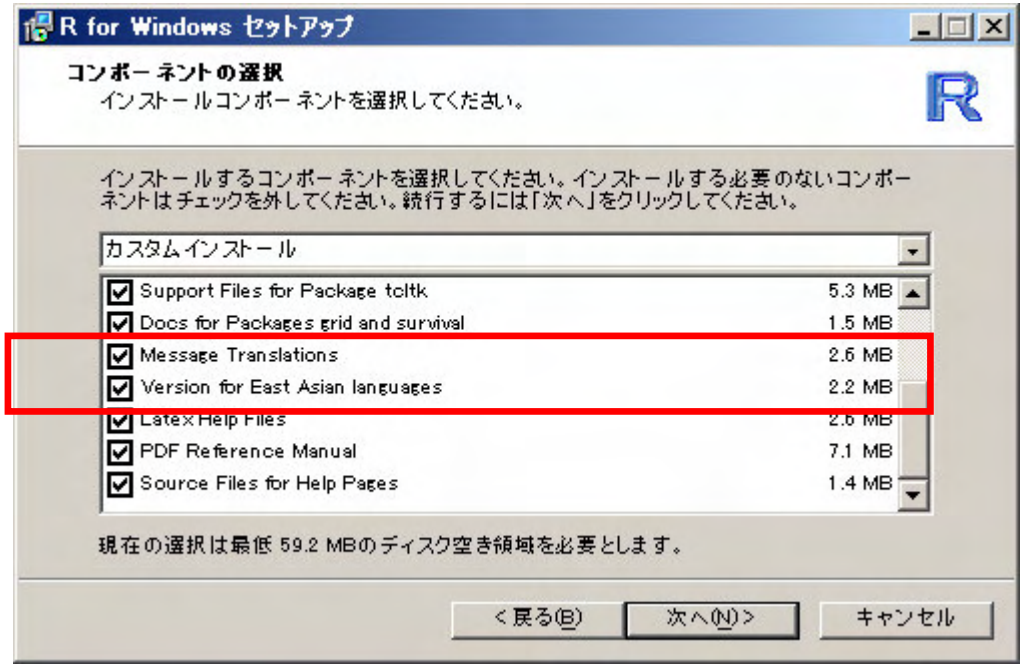

```
> plot(1:10)
> 1+2
[1] 3
> 3+4
[1] 7
> 1+2
[1] 3
> 3+4
[1] 7
> ?persp
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = rainbow(200))
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, border=NA, col = rainbow(50))
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, border=NA, col = rainbow(200))
>
```

R -2.2.0（＋なかまさんのパッチ）

- 完全国際化（面倒なセットアップ不要）



- インストール時にメニューからバージョンを選択



- 画像出力の際，日本語が入った EPS や PDF を出力しても文字化けしない

第４回のメニュー

- グラフィックスの作成手順
  - R でグラフィックスを作成する利点
  - グラフ作成の道具
  - グラフ作成の流れ
- 高水準作図関数
  - 高水準作図関数の一覧
  - 作図例
- 低水準作図関数
  - 低水準作図関数の一覧
  - 作図例

第４回のメニュー

- グラフィックスの作成手順
  - R でグラフィックスを作成する利点
  - グラフ作成の道具
  - グラフ作成の流れ
- 高水準作図関数
  - 高水準作図関数の一覧
  - データの種類による引数指定方法の違い
- 低水準作図関数
  - 低水準作図関数の一覧
  - 作図例

R でグラフィックスを作成する利点

- グラフィックスは R のセールス（？）ポイント
  - 簡単な命令で見栄えの良いグラフを作成することが出来る
    ⇒ 例えば，SAS のように多数のオプションを指定する必要がない
  - 図のカスタマイズが非常に簡単
    ⇒ 低水準作図関数で完成図に追記が可能
    ⇒ 専用のパッケージを使うことが出来る
  - 複雑な図や立体的な図も簡単に描くことが出来る

グラフ作成の道具

- Rでは，以下の2つを用いて作図を行う
  - 作図関数：グラフを出力する関数
  - 作図デバイス：図を出力する装置（ウィンドウ）
- 作図関数は主に以下の2つを用いる
  - 高水準作図関数：1枚の完成された図を描く

    （例）散布図を描く場合

    SAS → GPLOTプロシジャ，R → 関数 plot()

  - 低水準作図関数：図に図形や文字などを追記する

グラフ作成の道具

- 作図デバイスの種類
  - パソコンの画面に表示するデバイス
  - 画像ファイルに保存するためのデバイス
    - bmp()：ビットマップ 形式
    - jpeg()：JPEG 形式（3段階で品質が選択出来る）
    - pdf()：ADOBE PDF 形式
    - pictex()：LaTeX の画像形式
    - png()：PNG 形式
    - postscript：ADOBE PostScript 形式
      ⇒ 関数 dev.copy2eps()でEPSファイルへの保存も出来る

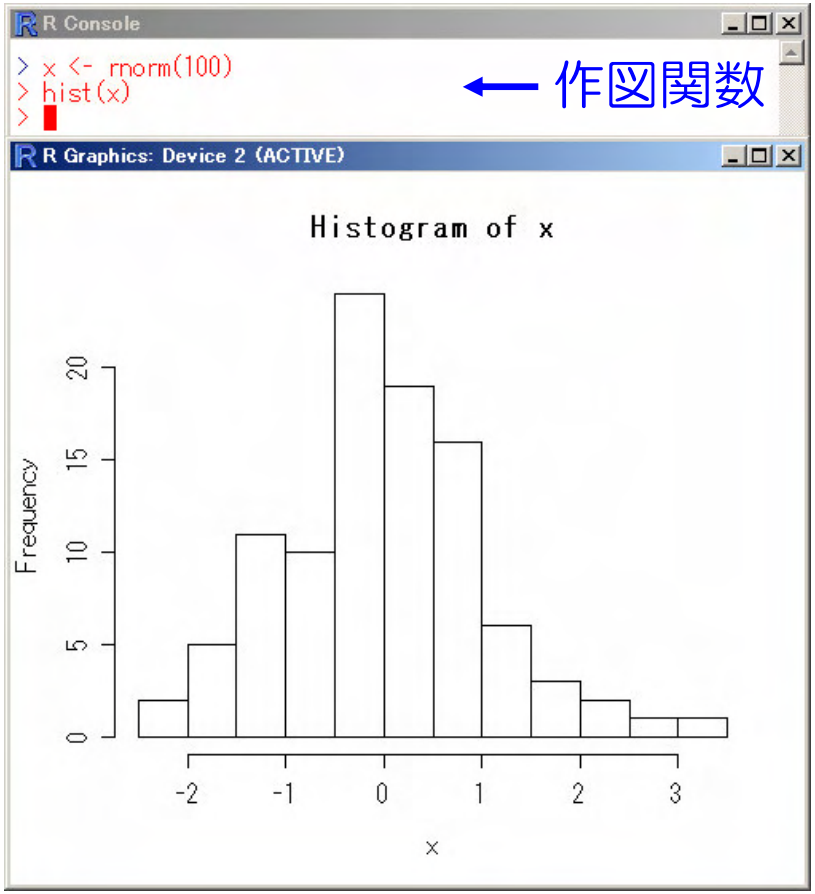
グラフ作成の流れ(ヒストグラム)

1. データを用意する(正規乱数を100個生成)

   x <- rnorm(100)

2. 関数 hist(x) を実行する

■ 関数にデータを指定するだけ！



作図デバイス →

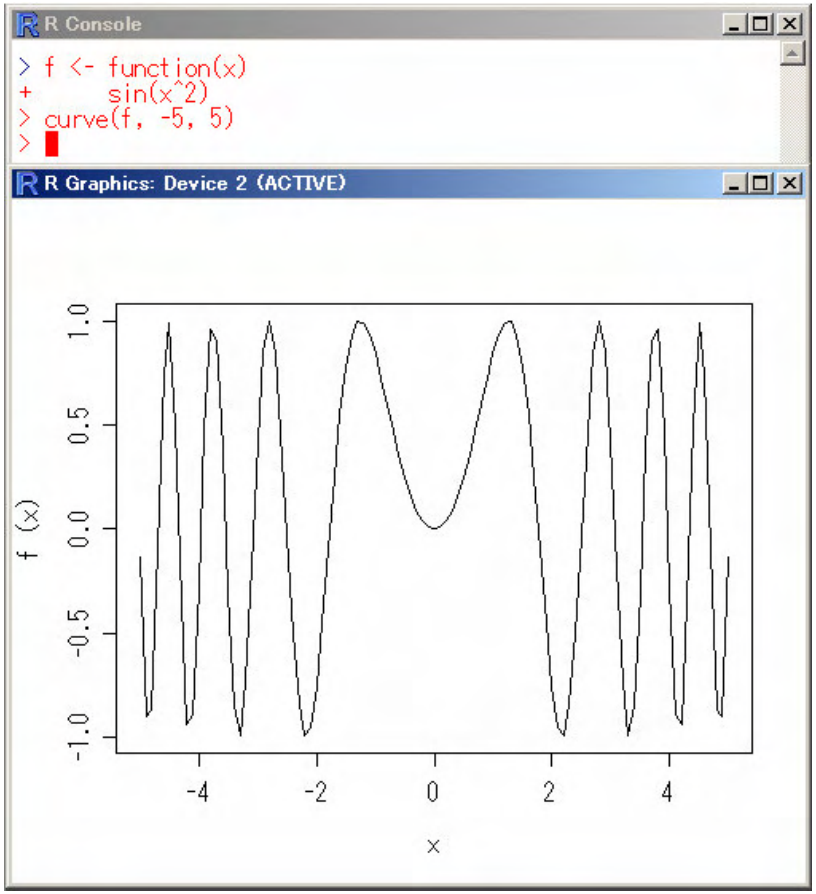
グラフ作成の流れ(数学関数のグラフ)

1. 数学関数を定義する

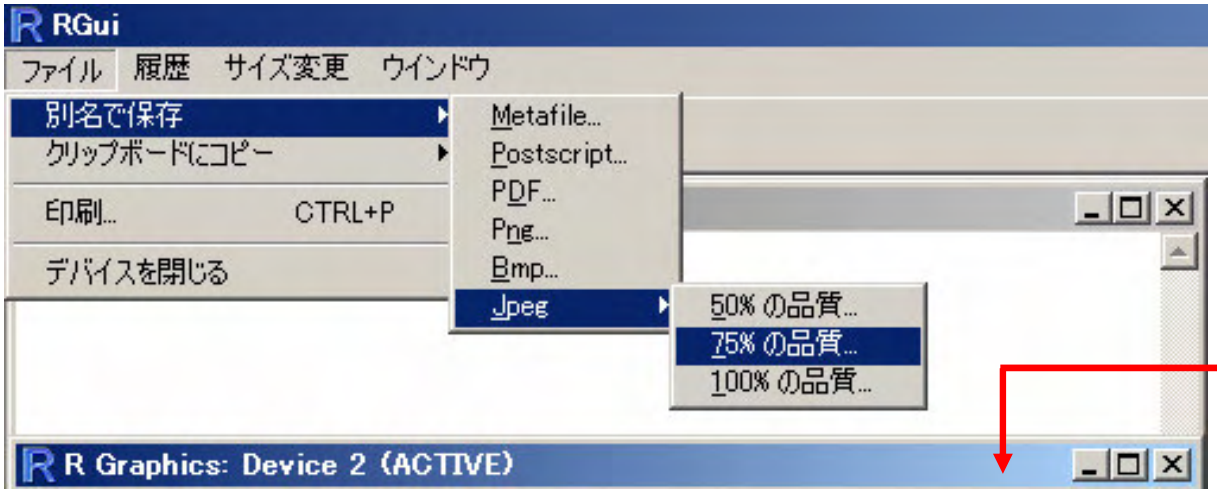
```
f <- function(x)
        sin(x^2)
```

2. 関数 curve(f)
   を実行する

■ 関数に数学関数を
指定するだけ！

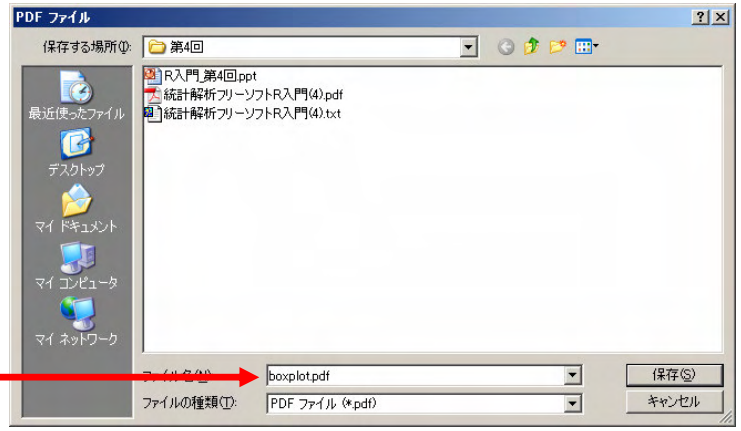


グラフ作成の流れ（図の保存）



★作図デバイスがアクティブになっていることを確認！

- 図を保存する場合は，メニューの［別名で保存］を選択する



第４回のメニュー

- グラフィックスの作成手順
  - R でグラフィックスを作成する利点
  - グラフ作成の道具
  - グラフ作成の流れ
- 高水準作図関数
  - 高水準作図関数の一覧
  - 作図例
- 低水準作図関数
  - 低水準作図関数の一覧
  - 作図例

高水準作図関数とは

- １枚の完成された図を作成するための関数
- 高水準作図関数で作図できる図：

  散布図，ヒストグラム，棒グラフ，円グラフ，
  箱ひげ図，数学関数のプロット・・・，他多数

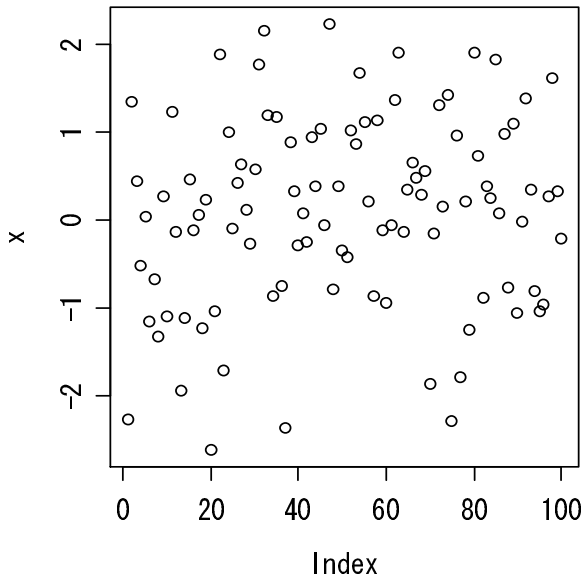
- 高水準作図関数を実行すると，自動的に作図デバイスが起動され，ウインドウに図が表示される．
- 図を描いた後，作図デバイスから画像ファイルへ図を出力することが出来る．

高水準作図関数の一覧

グラフの種類	関数
散布図	plot(データ)
ヒストグラム	hist(データ)
棒グラフ	barplot(データ)
円グラフ	pie(データ)
箱ひげ図	boxplot(データ)
分割表の図	mosaicplot(行列データ)
スターチャート	stars(行列データ)
対散布図	pairs(行列データ)
1次元関数のグラフ	curve(関数, 左端, 右端)
2次元関数のグラフ	persp(x軸, y軸, z軸)

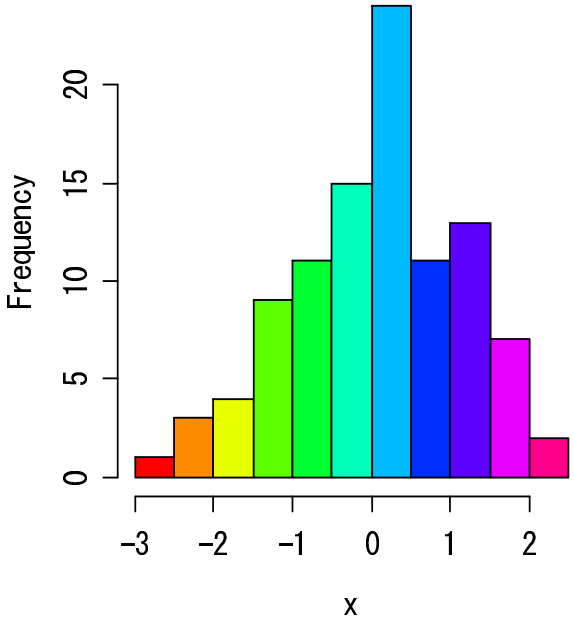
高水準作図関数の作図例(1)

- 100個の正規乱数データ x についてプロット



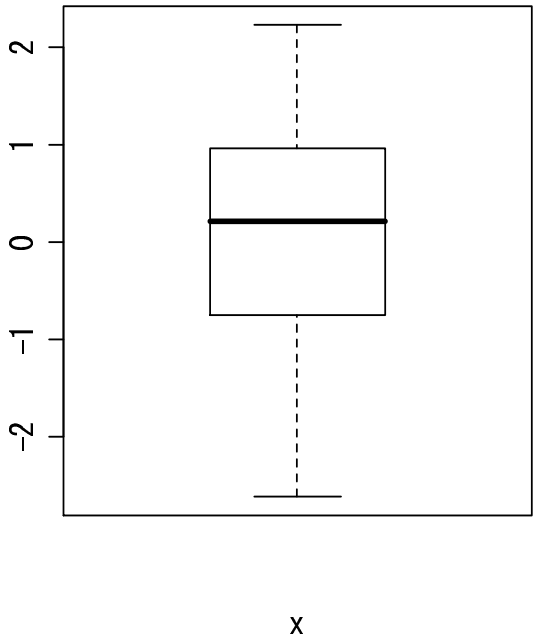
散布図

plot(x)



ヒストグラム

hist(x,col=rainbow(11))



箱ひげ図

boxplot(x, xlab="x")

高水準作図関数の作図例(2)

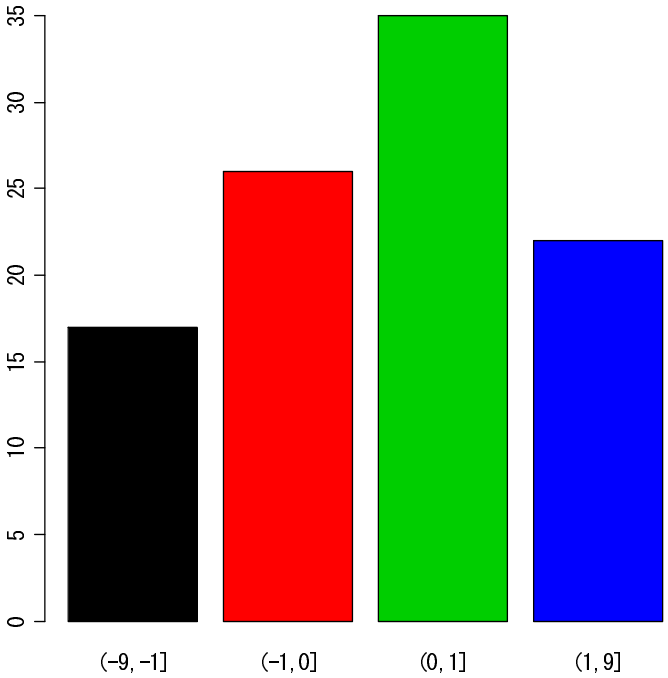
- 100個の正規乱数データ x について
  度数分布表 y を作成する

```
x <- rnorm(100)
y <- table( cut(x, c(-9, -1, 1, 9)) )
```

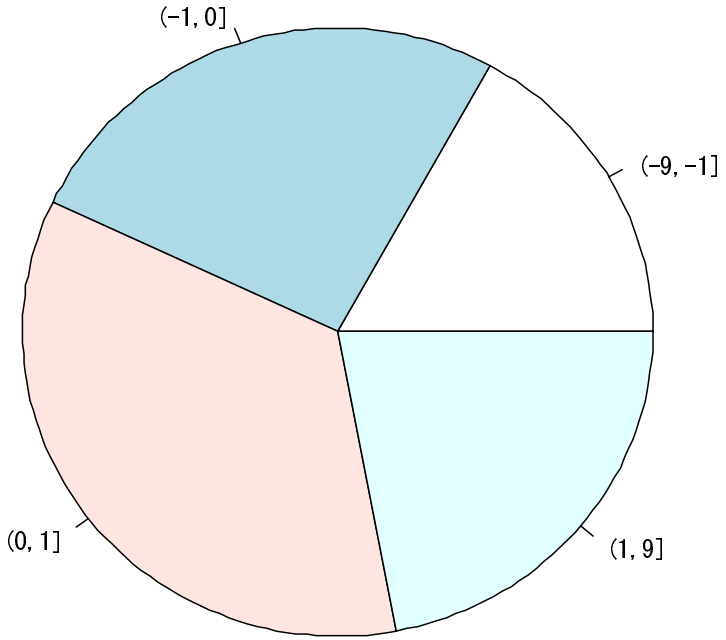
```
(-9,-1]  (-1,0]   (0,1]   (1,9]
     17      26      35      22
```

高水準作図関数の作図例(2)

- 度数分布表 y についてプロット



棒グラフ
barplot(y, col=1:4)



円グラフ
pie(y, r=1.0)

高水準作図関数の作図例(3)

- データフレーム x についてプロットを行う

```
x <- read.delim("clipboard")
```

- treatment : 治療群(因子)
- gender : 性別 (因子)
- weight : 体重 (数値)

⇒ 文字型は因子に強制変換

treatment	gender	weight
Placebo	male	64.4
Placebo	male	76.1
Placebo	male	68.9
Placebo	male	65.3
Placebo	female	52.2
Placebo	female	50.8
Active	female	53.7
Active	female	48.8
Active	female	51.9
Active	male	64.2
Active	male	65.7
Active	male	65.4

高水準作図関数の作図例(3)

■ データフレーム x についてプロットを行う

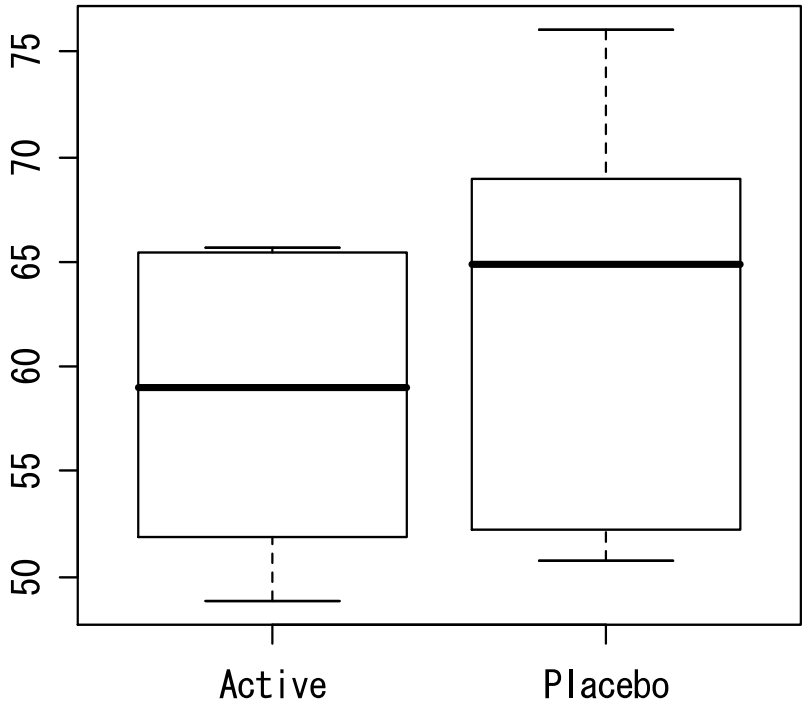
【書式】

関数名(変数 ~ 群,

data=データ名)

【箱ひげ図の場合】

```
boxplot(weight ~ treatment,
        data=x)
```



高水準作図関数の作図例(3)

■ データフレーム x についてプロットを行う

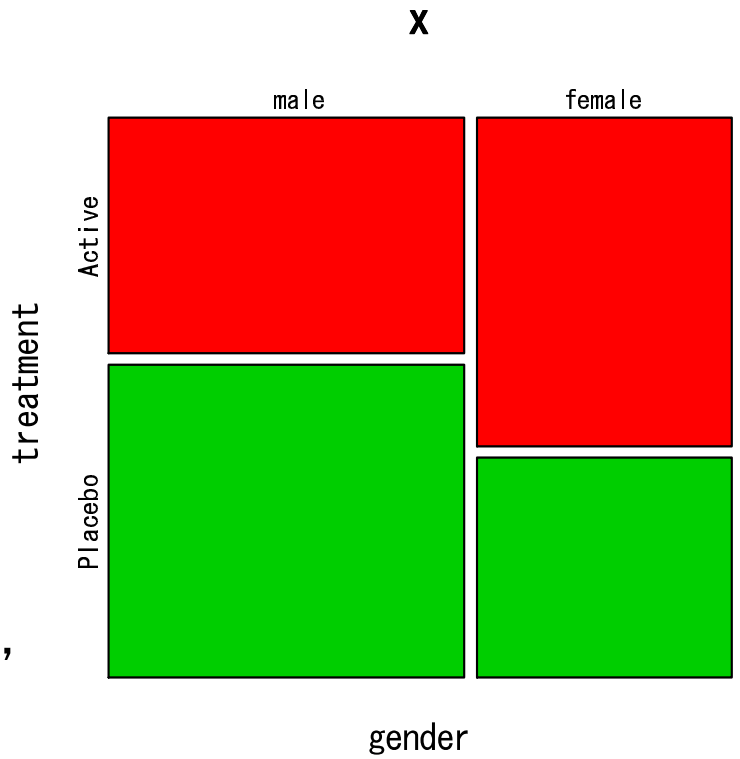
【書式】

関数名(変数 ~ 群,

data=データ名)

【モザイクプロットの場合】

```
mosaicplot(gender ~ treatment,
           col=2:3, data=x)
```



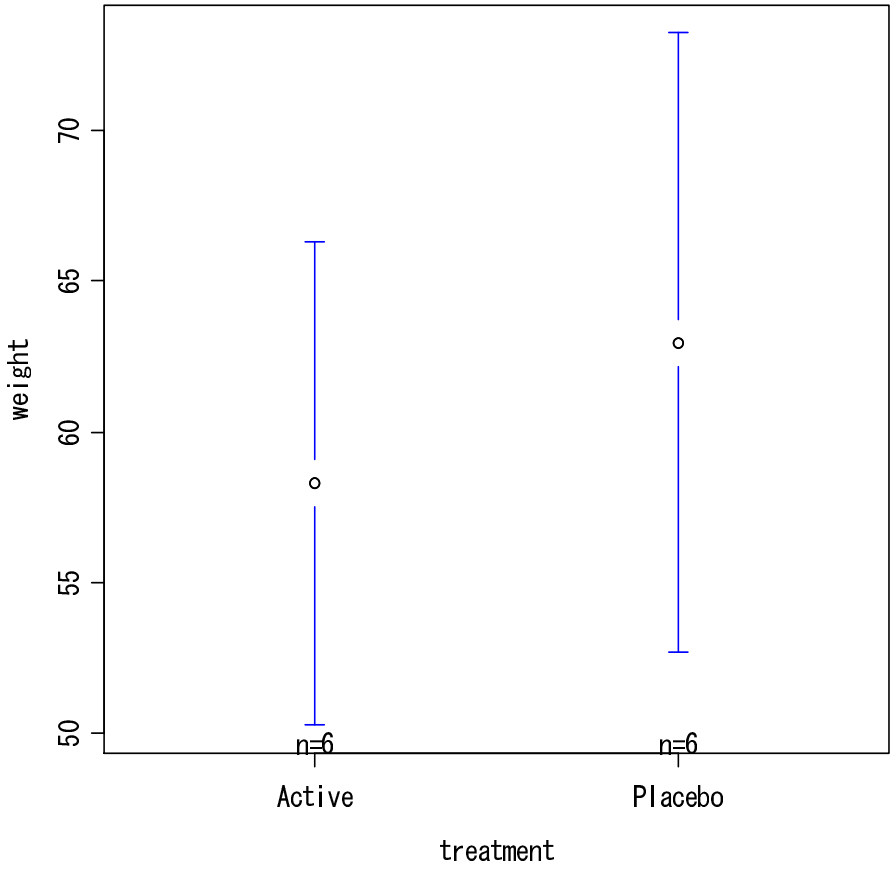
高水準作図関数の作図例(3)

■ データフレーム x についてプロットを行う

【平均値±標準偏差の表示】

```
library(gplots)
plotmeans(weight ~ treatment,
          connect=F, data=x)
```

⇒ パッケージ gplot を使用

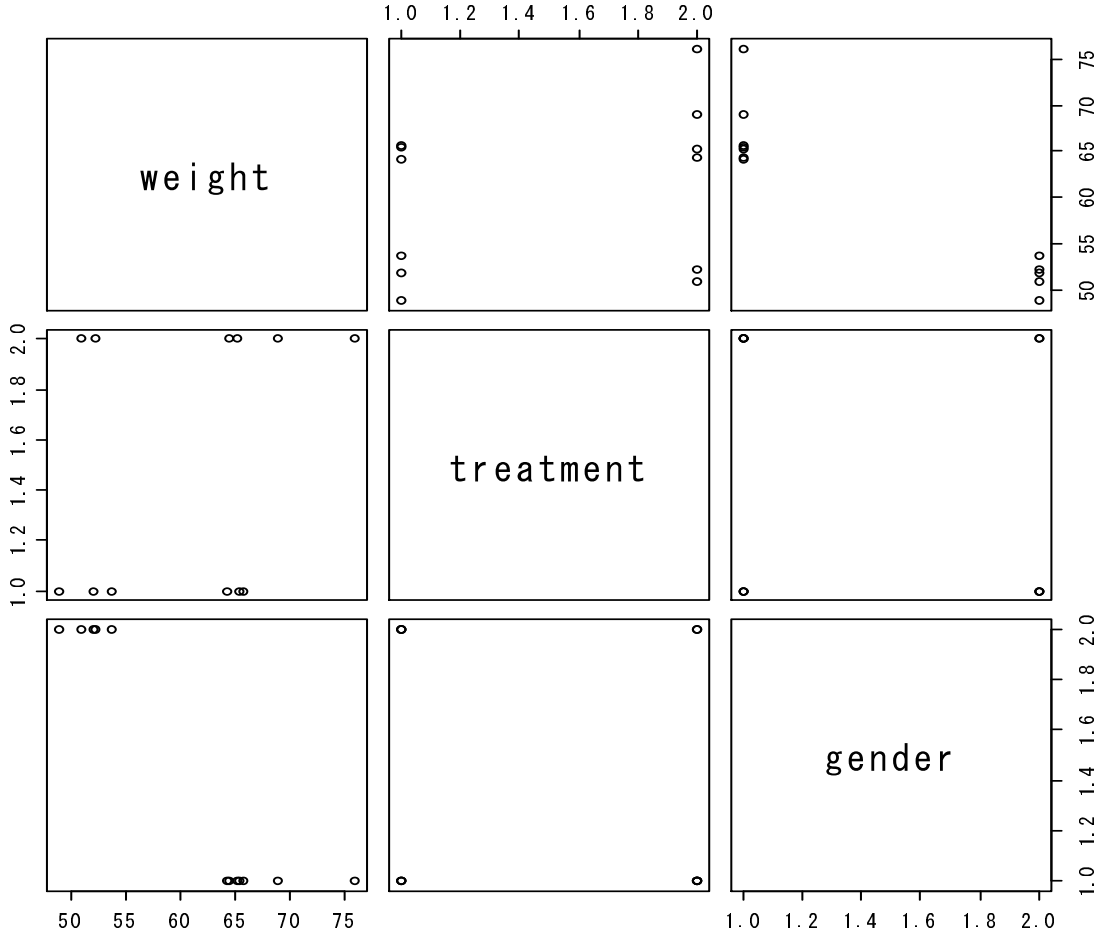


高水準作図関数の作図例(3)

- データフレーム x についてプロットを行う

【対散布図の場合】

```
pairs(weight ~
        treatment
        + gender,
        data=x)
```



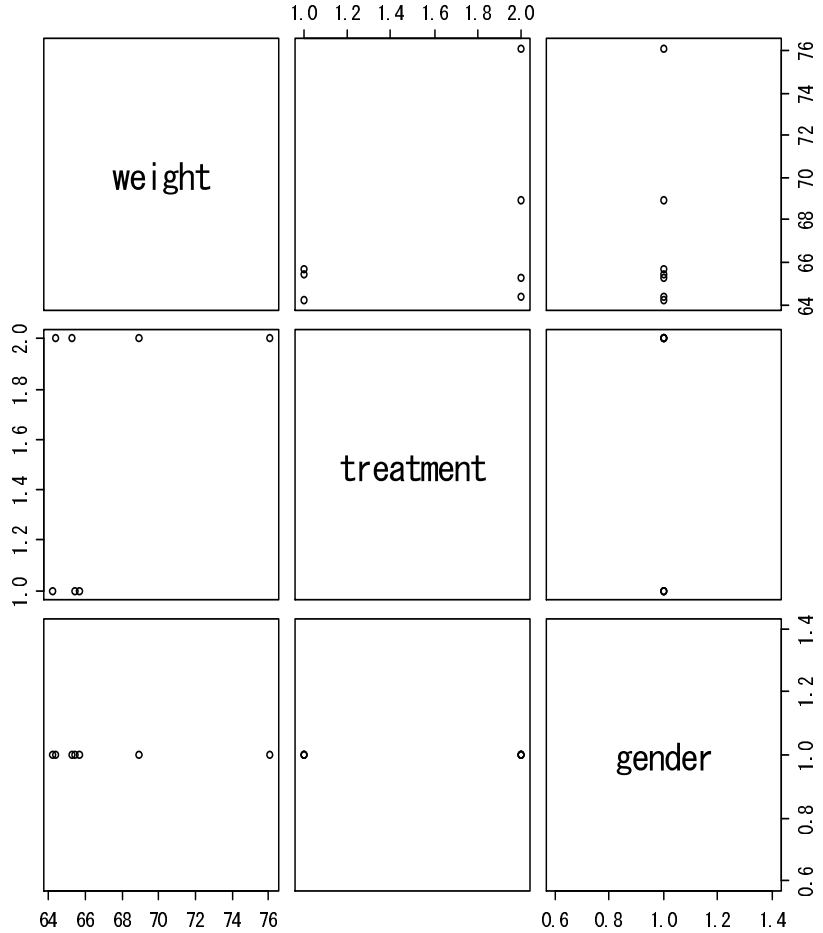
高水準作図関数の作図例(3)

■ データフレーム x についてプロットを行う

【対散布図の場合】

```
pairs(weight ~
        treatment + gender,
        subset=weight>60,
        data=x)
```

⇒ 部分集団の作図も可



高水準作図関数の作図例(4)

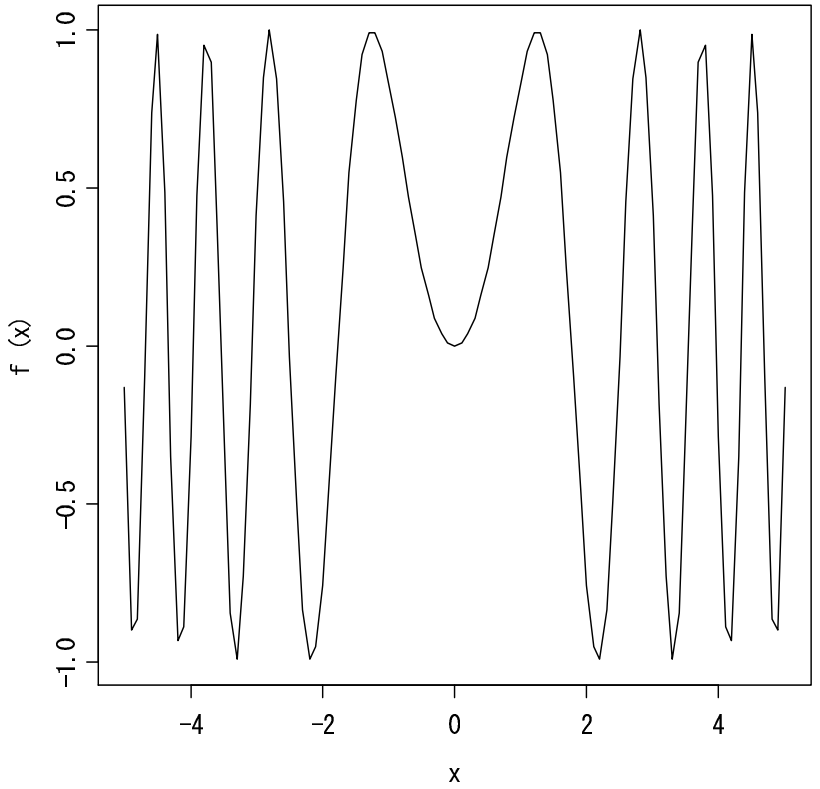
■ 一次元関数のプロット

【書式】

curve(数学関数名, 下限, 上限)

【作図例】

```
f <- function(x) {
 return( sin(x^2) )   ← 関数の定義
}
curve(f, -5, 5)    ← 関数 f の作図
```



高水準作図関数の作図例(4)

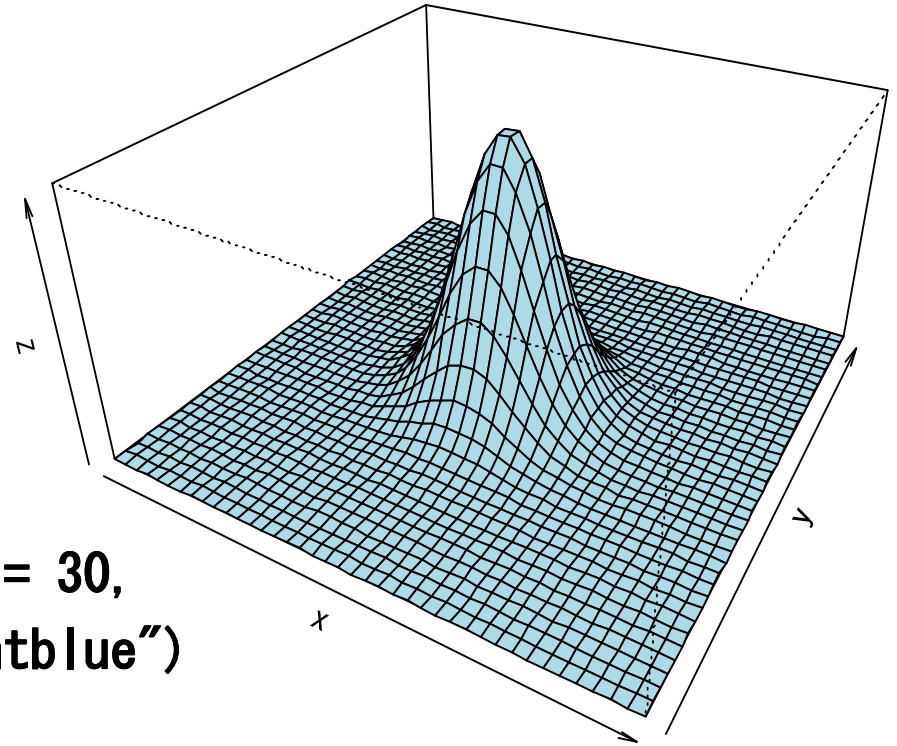
■ 二次元正規分布のプロット

```
x <- seq(-4, 4, length= 40)                    ← x の分点
y <- x                                         ← y の分点
f <- function(x, y) {
  return( exp(-(x^2 + y^2))/(2*pi) )           ← 密度関数
}
z <- outer(x, y, f)                            ← x と y の外積
persp(x, y, z, theta = 30, phi = 30,
      expand = 0.5, col = "lightblue")         ← 関数 f の作図
```

高水準作図関数の作図例(4)

■ 二次元正規分布のプロット

```
x <- seq(-4, 4, length= 40)
y <- x
f <- function(x, y) {
  exp(-(x^2 + y^2))/(2*pi)
}
z <- outer(x, y, f)
persp(x, y, z, theta = 30, phi = 30,
      expand = 0.5, col = "lightblue")
```



第４回のメニュー

- グラフィックスの作成手順
  - R でグラフィックスを作成する利点
  - グラフ作成の道具
  - グラフ作成の流れ
- 高水準作図関数
  - 高水準作図関数の一覧
  - 作図例
- 低水準作図関数
  - 低水準作図関数の一覧
  - 作図例

低水準作図関数とは

- 高水準作図関数で作成された図に，図形や文字を追記するための関数

  ⇒ いきなり低水準作図関数で図を描くことは出来ない

- 低水準作図関数で作図できる図：

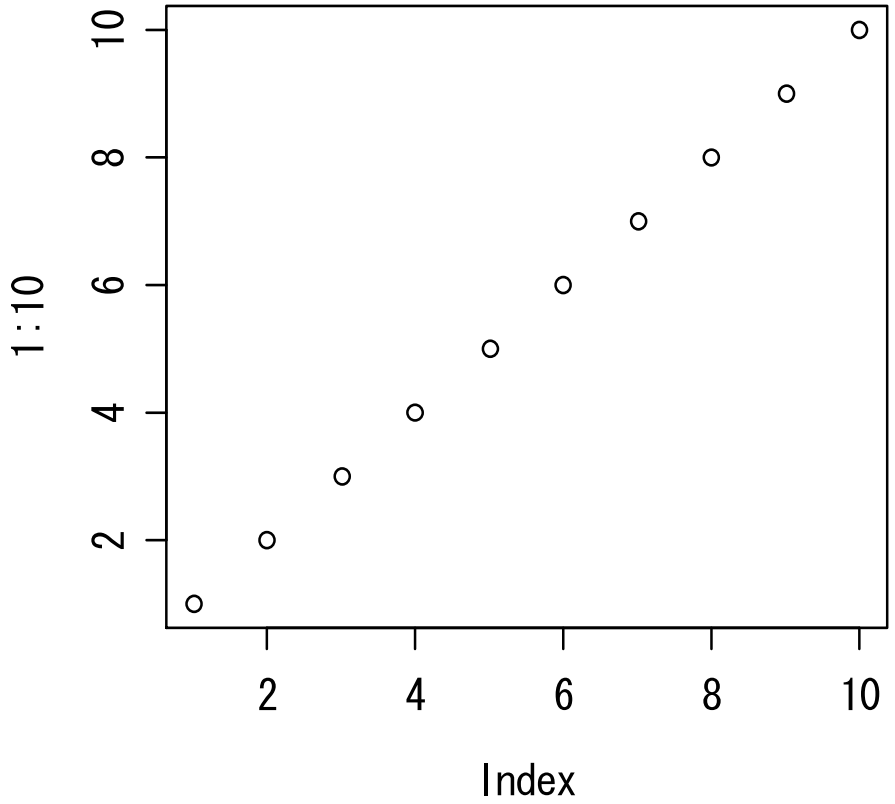
  点，直線，線分，格子，矢印，矩形，文字，枠，軸，凡例，多角形など

低水準作図関数の一覧

追記する図形	関数
点	points()
直線(1)	lines(), segments()
直線(2)	abline(), abline(回帰分析の結果)
格子	grid()
矢印	arrows()
矩形	rect()
文字	text(), mtext(), title()
枠と軸	box(), axis()
凡例	legend()
多角形	polygon()

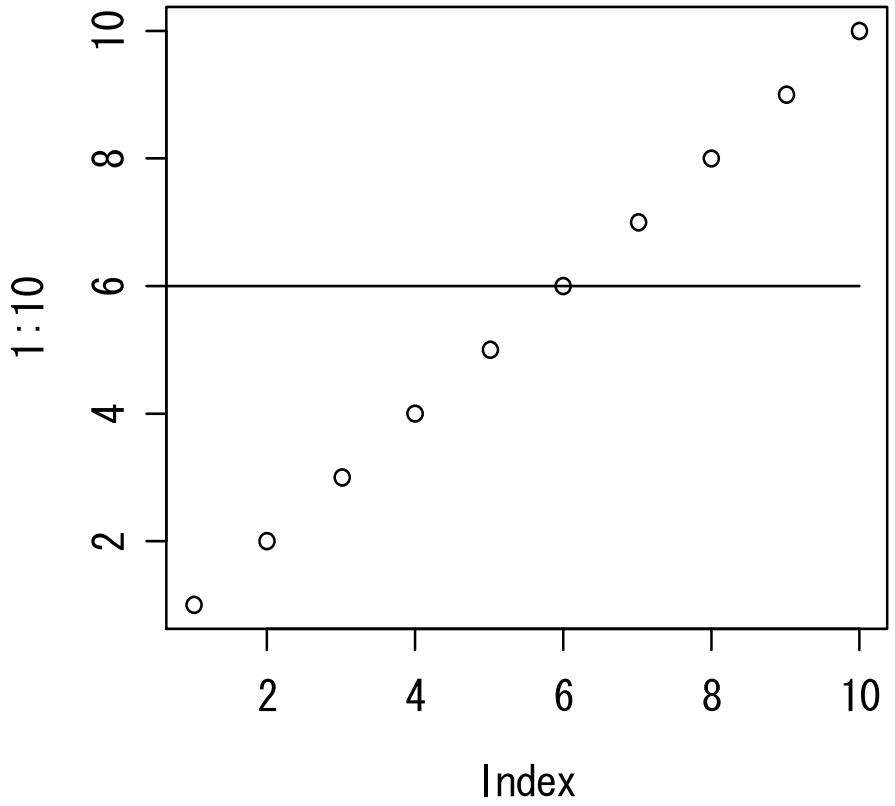
低水準作図関数の作図例(1)

```
plot(1:10)
```



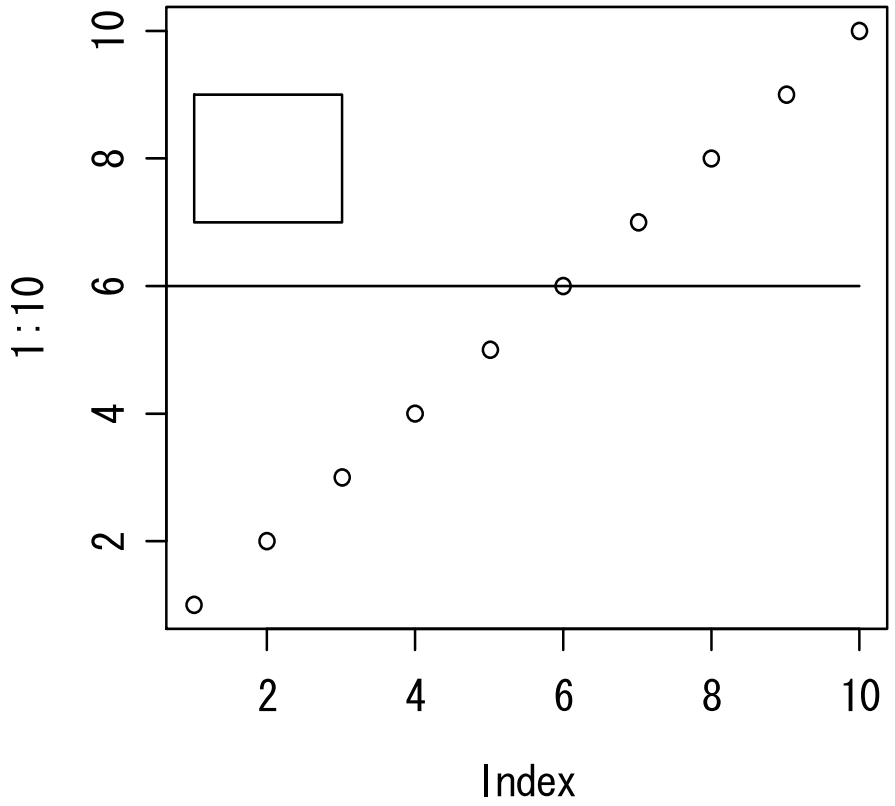
低水準作図関数の作図例(1)

```
plot(1:10)
lines(c(0,10), c(6,6))
```



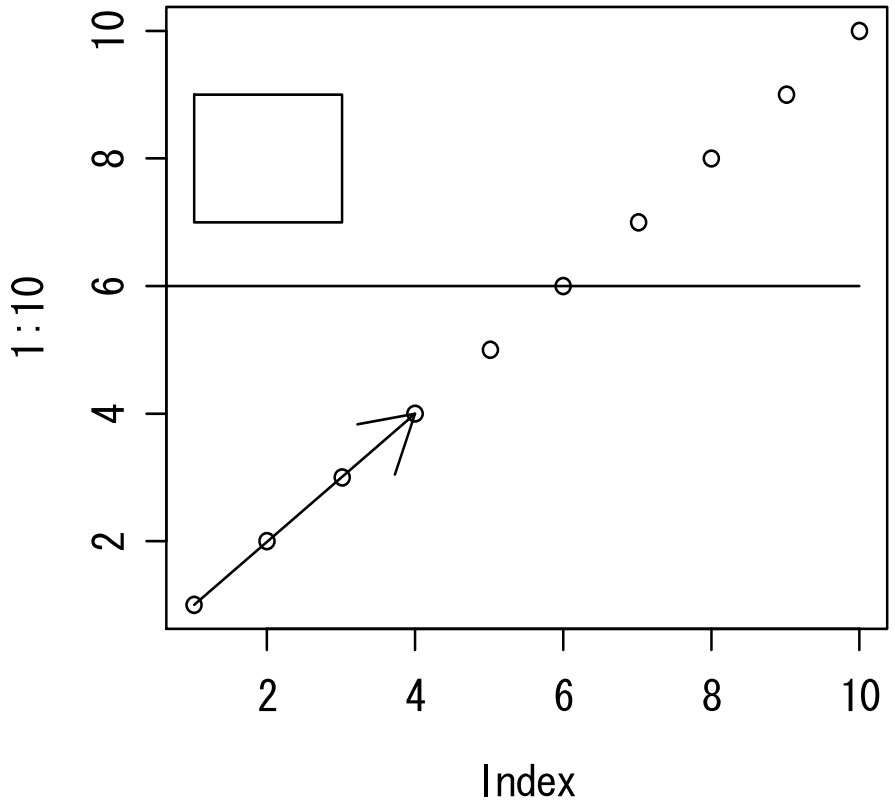
低水準作図関数の作図例(1)

```
plot(1:10)
lines(c(0,10), c(6,6))
rect(1,7, 3,9)
```



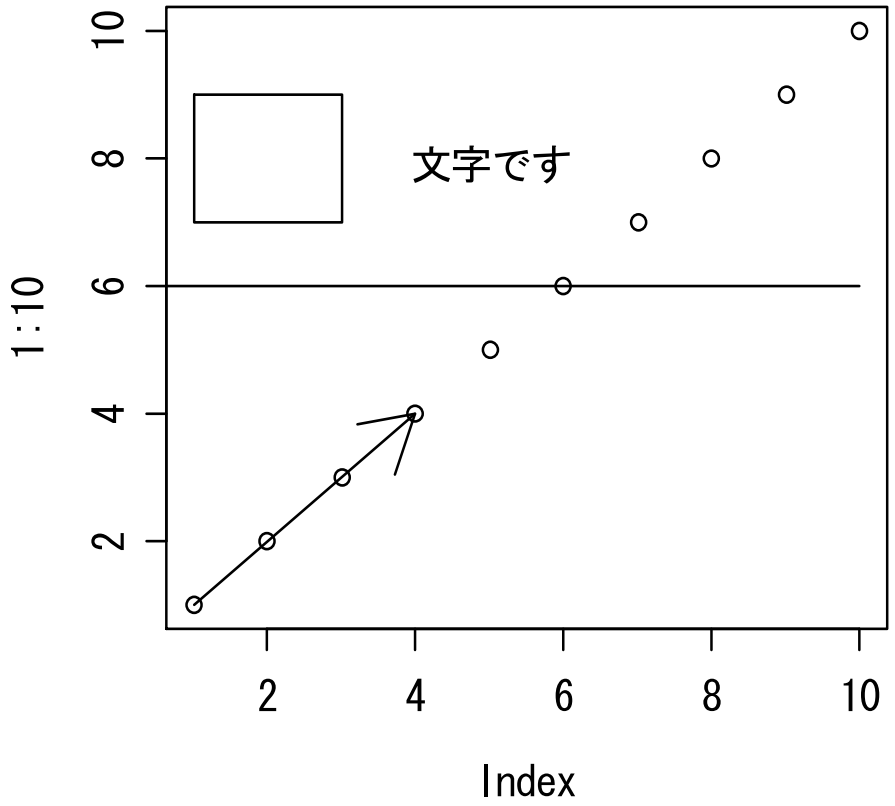
低水準作図関数の作図例(1)

```
plot(1:10)
lines(c(0,10), c(6,6))
rect(1,7, 3,9)
arrows(1,1, 4,4)
```



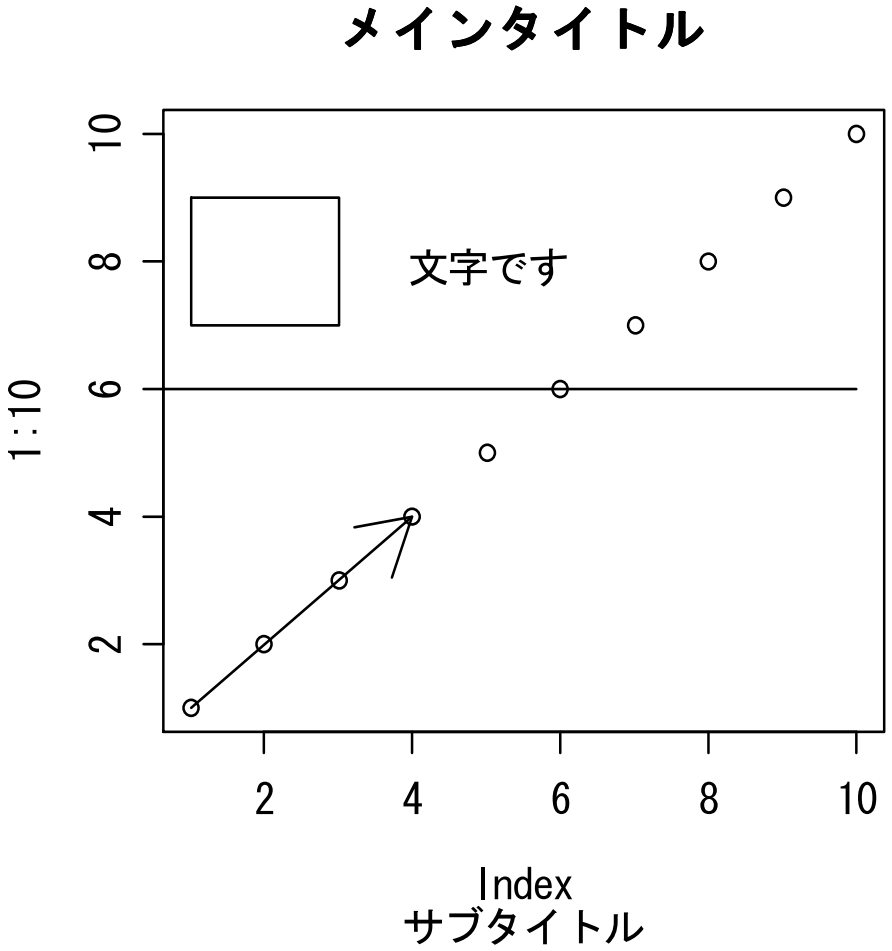
低水準作図関数の作図例(1)

```
plot(1:10)
lines(c(0,10), c(6,6))
rect(1,7, 3,9)
arrows(1,1, 4,4)
text(5,8,"文字です")
```



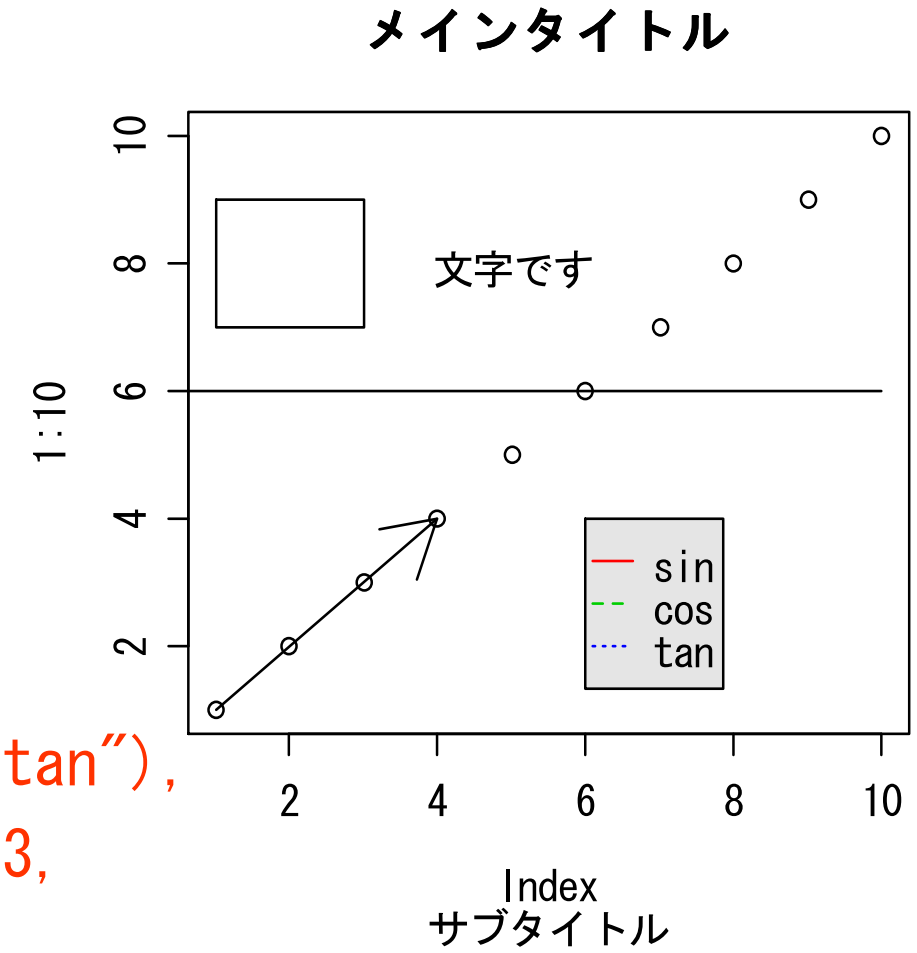
低水準作図関数の作図例(1)

```
plot(1:10)
lines(c(0,10), c(6,6))
rect(1,7, 3,9)
arrows(1,1, 4,4)
text(5,8,"文字です")
title("メインタイトル",
      "サブタイトル")
```



低水準作図関数の作図例(1)

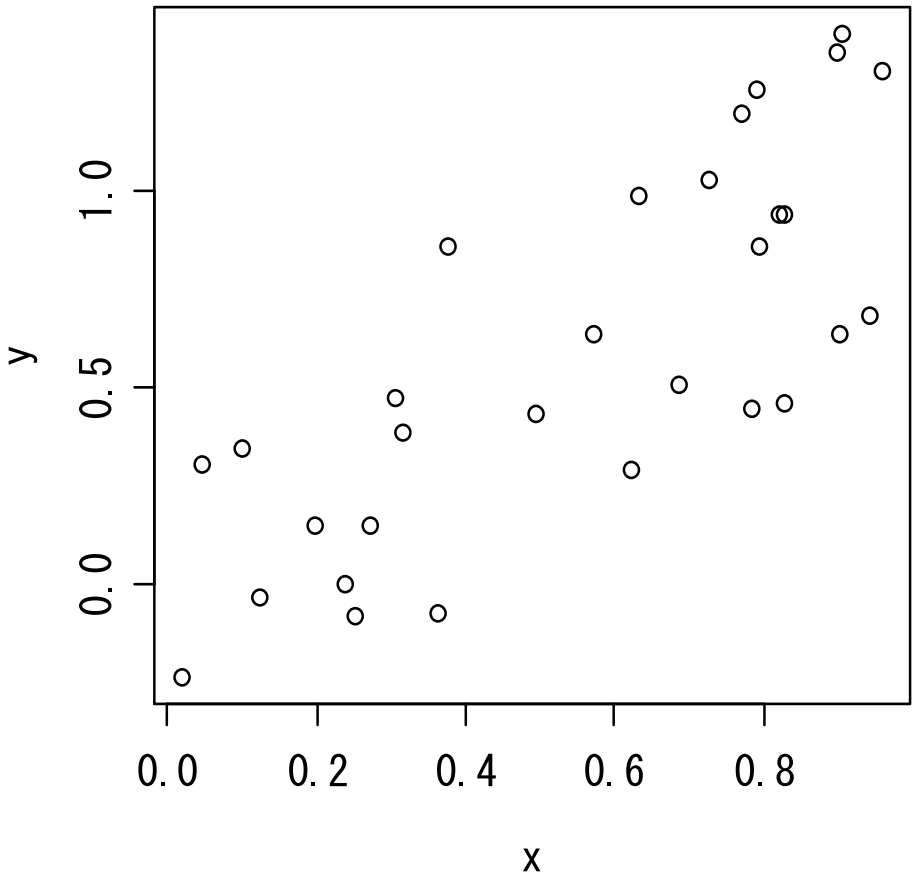
```
plot(1:10)
lines(c(0,10), c(6,6))
rect(1,7, 3,9)
arrows(1,1, 4,4)
text(5,8,"文字です")
title("メインタイトル",
      "サブタイトル")
legend(6, 4,
       c("sin", "cos", "tan"),
       col=2:4, lty = 1:3,
       bg='gray90')
```



低水準作図関数の作図例(2)

- 低水準作図関数を用いて，回帰直線を追記する

```
x <- runif(30)
y <- jitter(x,amount=0.5)
plot(x, y)
```

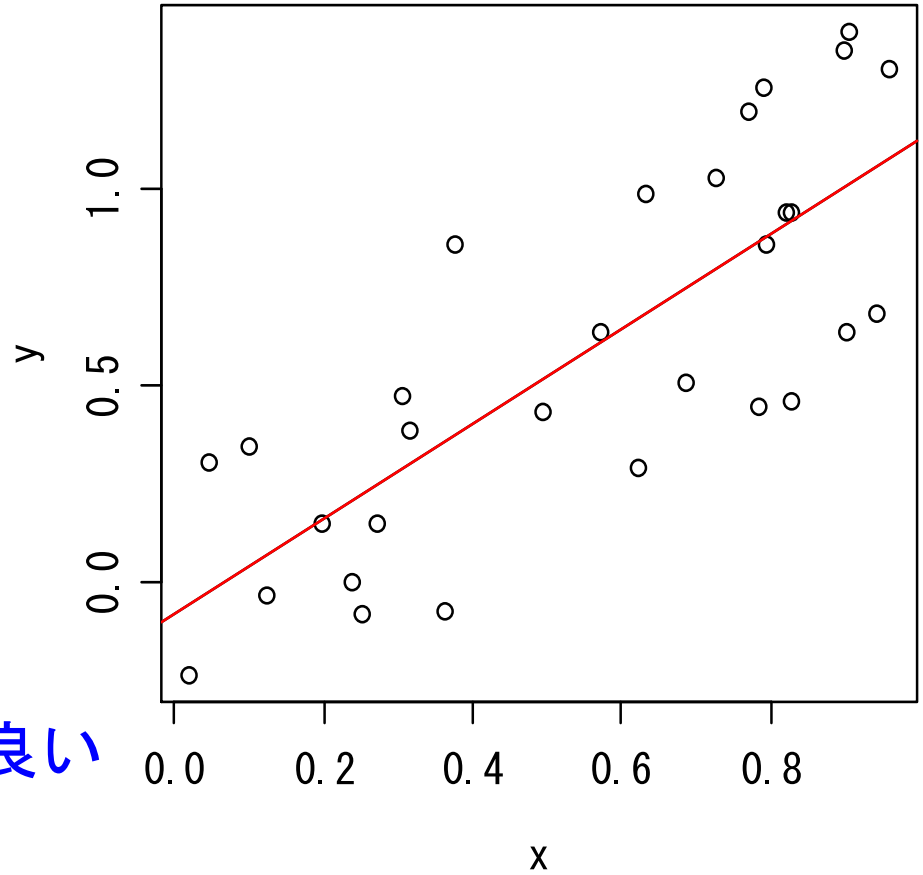


低水準作図関数の作図例(2)

- 低水準作図関数を用いて，回帰直線を追記する

```
x <- runif(30)
y <- jitter(x,amount=0.5)
plot(x, y)
result <- lm(y ~ x)
abline(result, col="red")
```

⇒この後，図を保存しても良い



第４回のまとめ

- グラフィックスの作成手順
  - R でグラフィックスを作成する利点
  - グラフ作成の道具
  - グラフ作成の流れ
- 高水準作図関数
  - 高水準作図関数の一覧
  - 作図例
- 低水準作図関数
  - 低水準作図関数の一覧
  - 作図例

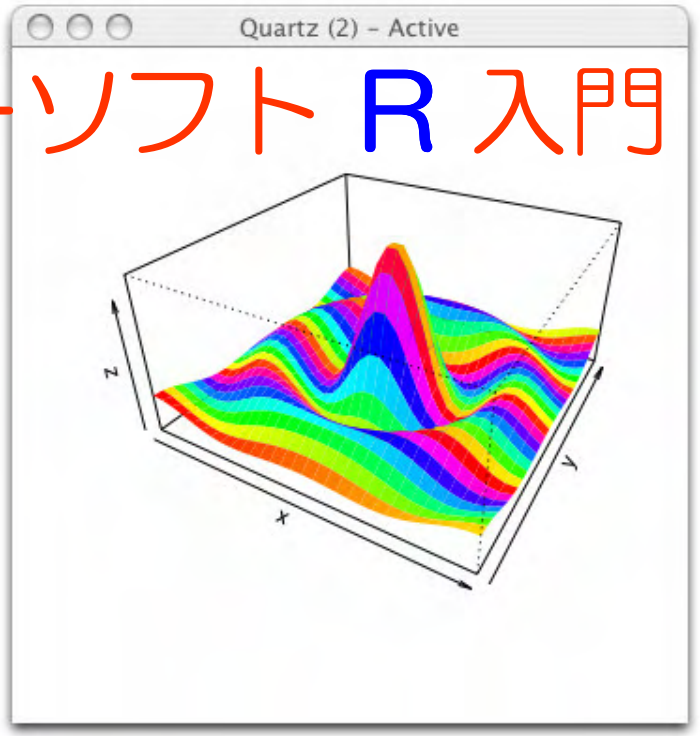
グラブ　ファイル　編集　取り込み　ウインドウ　ヘルプ　火 0:54

R Console

/Users/x

```
> plot(1:10)
> 1+2
[1] 3
> 3+4
[1] 7
> 1+2
[1] 3
> 3+4
[1] 7
> ?persp
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue")
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = rainbow(200))
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, border=NA, col = rainbow(50))
> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> op <- par(bg = "white")
> persp(x, y, z, theta = 30, phi = 30, expand = 0.5, border=NA, col = rainbow(200))
>
```

統計解析フリーソフト R 入門



終