■vector データの生成

値指定	c(x, y, z)
1刻み	m:n
s 刻み	seq(m, n, s)
t 等分	seq(m, n, length=t)
k 回繰り返し	rep(x, k)
m:n を k 回繰り返し	rep(m:n, k)
p に達するまで繰り返し	rep(m:n, length.out=p)
n 回ずつ繰り返し	rep(c(1, 5), each=2): 1155 (例)
m,n,… 回ずつ繰り返し	rep(c(1, 5), c(2, 3)): 11555 (例)
0ベクトルの生成	vector("numeric", length=n), numeric(n), rep(0, n): どれでも

■乱数データの生成

一様乱数	runif(n, a, b): 最小値a, 最大値b, n個生成(省略時0～1, 両端含まず)
正規乱数	rnorm(n, a, b): 平均a, 分散b, n個生成(省略時 平均0, 分散1)
シードの設定	set.seed(n)
ランダムサンプリング	sample(x, n, replace=T): xの中から n個取り出す(replace=Tで重複を許可)

■matrix データの生成 (単一データ型からなる行列)

値指定	x <- matrix(1:6, 2, 3): 2行3列の例 1 3 5 2 4 6 : 縦に並ぶことに注意 x <- 1:6; dim(x) <- c(2, 3) でも可
NaNで初期化	matrix(nrow=m, ncol=n)

■list データの生成 (異なるデータ長、データ型を格納可)

要素指定	list(aa=obj1, bb=obj2, …)
NULLで初期化	x <- list()
	length(x) <- n

■data.frame データの生成 (列毎に異なるデータ型を格納可)

値指定	data.frame(aa=1:2, bb=c(T,F)): 例

■vector, list のデータ操作

データの追加	x <- c(x, y) x <- c(x, list(y)): listでyを1要素に
データの削除	x <- x[-n]: n番目の要素を削除
入れ替え	x <- x[c(2, 1, 3)]: 1,2番目入れ替え
インデックスの取得	which(x==a): 値a の index を列挙
ラベル	names(x)
逆順	rev(x): rev(x)[1]で最後の要素を抽出
ソート	sort(x)

■matrix, data.frame のデータ操作

列方向に結合	cbind(x, y)
行方向に結合	rbind(x, y)
再帰的に結合する場合	x <- NULL: NULLを入れておく x <- c(x, y): エラーにならない
data.frameの結合	merge(x, y, all=T): 共通の列で結合 all.x=T とした場合は左結合 all.y=T とした場合は右結合
factorの不要なlevel削除	x <- x[drop=T]
転置	t(x)
列の入れ替え	y <- x[,c(2, 1, 3)]: 1,2列目入替
列の削除	y <- x[,c(-2, -3)]: 2, 3列目を削除
列名から列番号を取得	match("aaa", colnames(x))
行の操作	列と同様
行ラベル	rownames(x)
列ラベル	colnames(x)
ソート	x[order(y),]: yで整列

■オブジェクトを調べる

データ型の確認	mode(x)
クラスの確認	class(x)
構造の確認	str(x)
サマリの表示	summary(x)
属性の確認	attributes(x), attr(x, "name")
factorのlevelを表示	levels(x)

■要素の数

長さ	length(x)
次元	dim(x)
行の数	nrow(x)
列の数	ncol(x)
重複せずにカウント	sum(unique(x))
levelごとの要素数	table(x)
条件に該当する要素数	sum(iris$Species=="setosa")

■データの抽出

先頭の k 行を表示	head(x, n=k)
最後の k 行を表示	tail(x, n=k)
重複せずに抽出	unique(x)
vector, list	x[n]: n番目 x[m:n]: m～n番目 x[c(m, n, k:l, …)]: 飛び飛びでも x["aaa"]: ラベルで指定 x[c("aaa", "bbb", …)]: 複数も可 x[x>=a]: 条件式で指定
list	x[[n]], x[["aaa"]], x$aaa:要素を抽出
matrix, data.frame	x[m, n]: m行n列の要素 x[m,]: m行のデータ x[,n]: n列のデータ list等と同様、ラベルや条件式も指定可
data.frame	x[[n]], x[["aaa"]], x$aaa: 列を抽出
[]演算子の連続指定	x[2:4][2]: 2～4の2番目(3番目)を返す

■型変換、型チェック

型変換	as.integer(x), as.data.frame(x) など
型チェック	is.character(x), is.logical(x) など

■NAの除去

na.rm引数を持つ関数	sum(x, na.rm=T): NAを無視
vector	bad <- is.na(x) x[!bad]
matrix, data.frame	good <- complete.cases(x) x[good,]

■テキストファイルの読み込み、保存

読み込み	x <- read.table("aa.txt",sep="¥t"): タブ区切りの例 (下記等も指定可) header=F: ヘッダー行の有無 skip=0: 先頭から読み飛ばす行数 nrows=-1: 行数を指定 ディレクトリは「/」で区切る 「#」以降は省略されるので注意
保存	write.table(x, "aa.txt", quote=F, sep="¥t", row.names=F, na=""): 引用符なし, タブ区切り, 行ラベルなし , NAを空白とする例 (下記等も指定可) append=F: 追記 col.names=T: 列ラベル
ラッパー関数	read.csv(), read.delim(),
(.delim: タブ区切り)	write.csv(), write.delim() 等
一行ずつ	readLines(), writeLines()

■算術演算子、統計関数の例

整数商	%/%
剰余	%%
累乗	^
指数	exp(x), expm1(x): x小で高精度
平方根	sqrt(x)
絶対値	abs(x)
対数	log(x), log10(x), log2(x)
比較演算子	==, !=, <, >, <=, >=
比較用関数	is.null(), is.na(), is.nan(),
	is.finite(), is.infinite(): 無限?,
	complete.cases(): 欠損なし?
論理演算子	!, &, |, &&, ||, xor()
平均値	mean(x)
最頻値	mode(x)
頻度の集計	table(x): factorで良く使用する
中央値	median(x)
四分位偏差	quantile(x)[2]: 第1四分位点
	quantile(x)[4]: 第3四分位点
	(quantileは、最小値, 第1四分位,
	中央値, 第3四分位, 最大値 を返す)
パーセンタイル	quantile(x, c(0.1, 0.9)):
	10%, 90%パーセンタイルを返す
合計	sum(1:3): 6 を返す
TRUEの数をカウント	sum(c(T, F, T)): 2 を返す

■オブジェクトの操作

オブジェクトを列挙	ls()
検索順の確認	search()
存在の確認	exists("x")
オブジェクトの消去	rm(x), rm(list=ls()):全て消去
オブジェクトサイズの確認	print(object.size(get("x")), units="auto"): 個別 str(eapply(.GlobalEnv,object.size)): 全て

■サンプルデータ（あらかじめデータセットが用意されている）

一覧表示	data()$results[,c("Item","Title")]
解析サンプルの表示	example(cars): carsの例

■パッケージ

一覧表示	library()
読み込み	library(aaa), require(aaa)
インストール	install.packages("aaa", dependencies=T): 依存関係含む

■作業スペースの読み込み、保存

読み込み	load("aaa.Rdata")
保存	save(x,y,…,file="aaa.Rdata"): 個別 save(list=ls(),file=""): 全て

■環境操作

画面クリア	Ctrl + l キー
ワークディレクトリ確認	getwd()
ワークディレクトリ設定	setwd("aaa")

■処理時間の計測

スクリプトの処理時間	t <- proc.time() # 処理 proc.time() - t
関数の処理時間	system.time(my.func(x))

■その他

ヘルプ	help(func.name), ?func.name
処理の中断	ESC キー

■基本文法

コメント	# コメント
代入	y <- x: 普通の代入
TRUE, FALSE の略語	T, F で、TRUE, FALSEを表す
値の表示	x: xの値を表示 print(x): 関数やループ内で明示的に

■条件分岐

if文	if(条件文) { 処理1 } else if(条件文) { 処理2 } else { 処理3 } 条件文でx < -1 を x<-1とくっつける と、代入文と見なされるので注意！
?:演算子風	y <- if(条件文) { x1 } else { x2 }

■繰り返し

for文	for(i in 1:n) { 処理 }
	for(I in 1:length(x)) { 処理 }
	for(i in seq_len(n)) { 処理 }
	for(i in seq_along(x)) { 処理 }

foreach風	for(x in y) { 処理 }

2次元ループの例	for(i in seq_len(nrow(x))) {
	  for(j in seq_len(ncol(x))) {
	  処理
	  }
	}

whileループ	while(条件文) { 処理 }
無限ループ	repeat { if(条件文) { break } }
ループ制御	break: ループを抜ける
	next: 次の回へ(continueに相当)

■apply系（for文より圧倒的に高速）

data.frame用	apply(x, 1, func): 行毎にfuncを適用 apply(x, 2, func): 列毎にfuncを適用 apply(x, 2, function(x) { 処理 })
vector, list用	lapply(x, func, arg=val, …) sapply(): lapplyのシンプル表示版 mapply(func, arg1, arg2, …): 複数の引数の組に対してfuncを適用 tapply(x, f, func): factor型変数 f のクラス毎にfuncを適用 tapply(x, f, function(x) { 処理 })
splitとの併用	s <- split(x[drop=T], f): fのクラス毎に分割 sapply(s, func): リストs毎にfunc適用
with関数 (apply系以外でも便利)	tapply(mtcars$hp, mtcars$cyl, mean) は、with関数を使うと、毎回「mtcars$」と書かずに済む with(mtcars, tapply(hp, cyl, mean))

■関数の定義

書式	my.func <- function(a, b=初期値) {
	  処理
	}
	戻り値ある場合は return (y) 等とする
	戻り値が複数の場合 return(list(x, y))

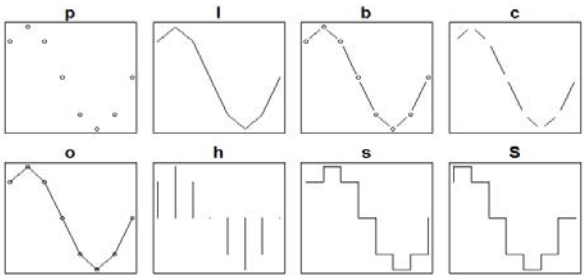
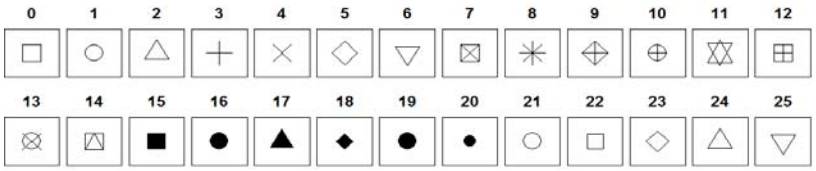
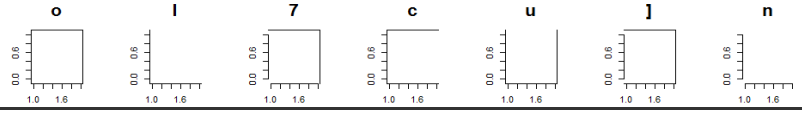
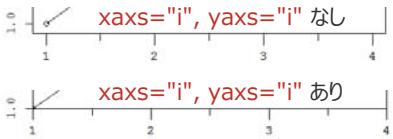
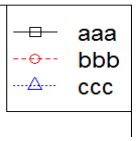
■おまけ

文字列をコマンドとして実行	eval(parse(text="command")) で、"command"を実行
代入演算子	「=」でも代入できるが、スコープの範囲が限られるため「<-」推奨
オブジェクト名	obj.nameのようにピリオドで単語を区切る命名法が一般的

■グラフウィンドウ、ファイル

新しいウィンドウを開く	dev.new()
サイズ指定で開く	win.graph(m, n): サイズをインチで指定（windows用のコマンド）
pngファイルの生成	png("aaa.png", width=w, height=h): サイズをピクセル値で指定
ファイルのクローズ	dev.off()

■時系列プロット

時系列プロット、散布図	plot(x): 時系列プロット、plot(x, y): 散布図
タイプ	type="p" 点プロット(省略時設定) type="l" 線プロット type="b" 点と線の両方（隙間あり） type="c" "b"の点なしプロット type="o" 点と線の両方（隙間なし） type="h" 垂線プロット type="s" 左側の点を延ばして階段状に結ぶ type="S" 右側の点を延ばして階段状に結ぶ type="n" 軸だけ描画してプロットしない(lines()等で追記する場合などに使用) 
点の種類	pch=1: ○でプロット 0-25: 右図参照 32-127: ASCII文字 pch="x": xでプロット （文字でも指定可） 
点の大きさ	cex=2: 点の大きさ2倍
線の種類	lty=0: 実線で描画 lty="solid": でも同様 （lty: line type の略） 
線の太さ	lwd=2: 線の太さ2倍
波形色	col=1: 黒で描画（1: 黒, 2: 赤, 3: 緑, 4: 青, 5: シアン（水色）, 6: マゼンタ（紫）, 7: 黄 8: 灰） （番号以外にも、"blue"などの文字色、rgb(0, 0, 0)、"#FFFFFF", rainbow(6) 等の指定も可）
グラフタイトル	main="title": メインタイトル、sub="subtitle": サブタイトル
タイトル文字サイズ	cex.main=2: メインタイトルの文字のサイズ2倍、cex.sub=2: サブタイトルの文字サイズ2倍
軸の範囲	xlim=c(x.min, x.max): x軸の範囲、ylim=c(y.min, y.max): y軸の範囲
対数軸	xlog=T: x軸を対数軸に、ylog=T: y軸を対数軸に（log="x", log="y", log="xy"でも可）
y／xアスペクト比	asp=1: x軸とy軸の目盛の範囲を1:1に調整（グラフ描画サイズは変わらず目盛の範囲が変化）
軸の非表示	xaxt="n": x軸を非表示、yaxt="n": y軸を非表示、axes=F: x軸y軸を非表示（枠線まで消える）
軸の種類	bty="o": 周囲が囲まれた枠 （bty: box type の略） 
軸ラベル	xlab="x.label": x軸ラベル、ylab="y.label": y軸ラベル（描画しない場合は xlab=""のようにする）
軸ラベル文字サイズ	cex.lab=2: 軸ラベルの文字のサイズ2倍、cex.axis=2: 目盛の文字サイズ2倍
軸ラベル等の非表示	ann=F: 軸ラベル、タイトルを非表示（軸ラベルだけ非表示にするときは、xlab="", ylab=""とする）
ラベル(目盛)の向き	las=0: 各軸に平行、las=1: 全て水平、las=2: 各軸に垂直、las=3: 全て垂直
目盛線の長さ	tck=1: グラフ内に目盛線を描画、tcl=0.5: 目盛線を内側に文字列の0.5行分描画（負の値は外側）
目盛のカスタマイズ	plot(1:4, type="o", xaxt="n", xlab="", xaxs="i", yaxs="i"): グラフが枠の端まで来る（右図下段） axis(1, at=c(1:4)-0.5, labels=F): 第一引数 1:下,2:左,3:上,4:右 axis(1, at=1:4, tcl=-0.8): tclは目盛線の長さ（初期値 -0.5） 
表示色	col.main: メインタイトル, col.sub: サブタイトル, col.axis: 目盛, col.lab: 軸ラベルの色
文字スタイル	font.axis=2: ラベル（目盛）をBoldに指定（1:plain, 2:bold, 3:Italic, 4:Bold-italic, 5:Symbol） 他に、font.lab: 軸ラベル, font.main: メインタイトル, font.sub: サブタイトルも設定可
凡例を追加	labs <- c("aaa", "bbb", "ccc") ltys <- c(1, 2, 3); pchs <- c(0, 1, 2); cols <- c("black", "red", "blue") legend("topright", legend=labs, lty=ltys, pch=pchs, col=cols, cex=1.5, pt.cex=1.5): cex はラベル文字の大きさ、pt.cex は記号の大きさ 
プロットの追加	points(x, y): 点プロットを追加, lines(x, y): 線プロットを追加
線の追加	abline(a, b): 直線を追加(a:傾き, b:切片)、abline(h=y): 水平線を追加、abline(v=x):垂直線を追加 grid(a, b): 水平、垂直線を追加（垂直線a本、水平線b本）
文字の追加	text(x, y, labels): x,y座標に追加, mtext(text, side=3, line=0): 周囲の余白に追加
その他の項目の追加	box(): 枠線、title("main", "sub"): タイトル、axis(): 軸（上記参照）、legend(): 凡例（上記参照）

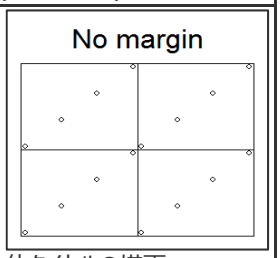
R quick reference (graphic 2)

■グラフ領域の操作

全体の設定	plot()を呼び出す前に、par(param=value)を実行する（par(font.lab=2): 軸ラベルをBold など）
マージン（余白）	par(mar=c(1, 1, 4, 1)): 余白の大きさを文字列1行分の高さ(mex)で指定（下、左、上、右の順） par(mai=c(0.5, 0.5, 2, 0.5): インチで指定
背景の色	rect(par("usr")[1], par("usr")[3], par("usr")[2], par("usr")[4], col ="grey"): plot()後に指定
フォントファミリー	family="mono": 値は、"serif", "sans", "mono", "symbol"を指定

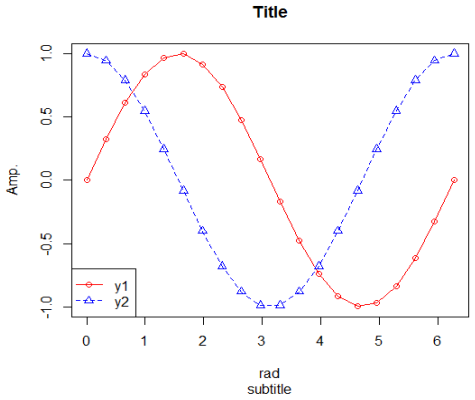
■複数のグラフ描画

row 行 col 列 に分割	par(mfrow=c(row, col)): row行、col列に分割、plot()で行方向に順に描画（列方向はmfcol）
全体のマージンを指定	par(oma=c(bottom,left,top,right)):mexで指定された文字サイズの倍率で余白を指定(全て4くらい)
各グラフのマージン	par(mar=c(bottom,left,top,right)): 単位はmex(枠のみの場合全て0) 以下、2 x 2 に分割、マージン無しの例 par(mfrow=c(2,2)); par(oma=c(1,1,4,1)) par(mar=c(0,0,0,0)); plot(1:4, ann=F, xaxt="n", yaxt="n") par(mar=c(0,0,0,0)); plot(1:4, ann=F, xaxt="n", yaxt="n") par(mar=c(0,0,0,0)); plot(1:4, ann=F, xaxt="n", yaxt="n") par(mar=c(0,0,0,0)); plot(1:4, ann=F, xaxt="n", yaxt="n") par(oma=c(0,0,3,0)); mtext(text="No margin", side=3, outer=T, cex=2): 全体タイトルの描画



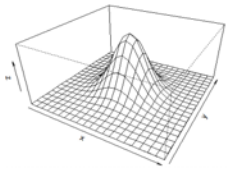
■plot()の雛形

各グラフのマージン	N <- 2*pi; x <- seq(0,N,length=20)
	y1 <- sin(x); y2 <- sin(x+0.5*pi)
	plot(x, y1, # x, y1 の散布図
	  type="o", # 点と線のプロット
	  lty=1, # ラインタイプ：1=実線
	  pch=1, # プロット文字：1=○
	  col="red", # 線の色：赤
	  xlim=c(0, N), ylim=c(-1,1), # x軸、y軸範囲
	  main="Title", sub="subtitle", # タイトル
	  xlab="rad", ylab="Amp.") # 軸ラベル
	lines(x, y2, type="o",
	  lty=2, pch=2, col="blue") # 線プロット追加
	legend("bottomleft", legend=c("y1","y2"), lty=c(1,2), pch=c(1,2), col=c("red","blue") # 凡例



■その他のグラフ

複数時系列プロット	matplot(x): x は行列で、列毎にグラフを描画
3次元プロット	persp(x, y, z, theta=a, phi=b, expand=c, col=d) x <- seq(-4,4,0.4); y <- x z <- outer(x, y, function(x,y){dnorm(sqrt(x^2+y^2))}); persp(x,y,z, theta = 30, phi = 30, expand = 0.5) # persp以外にも rgl パッケージの plot3d() がおススメ
棒グラフ	plot(f): f は因子型(factor)オブジェクト
棒グラフ2	barplot(x): vectorなら普通の棒グラフ、行列なら積み上げ棒グラフ (space=1: 棒の間隔)
箱ひげ図	plot(f, y): f は因子型(factor)オブジェクト，y は数値ベクトル，あるいは、boxplot(x)
ヒストグラム	hist(x, breaks=n): n分割してヒストグラムを描画（freq=Fを指定して頻度分布ではなく確率密度の描画も可）
円グラフ	pie(x, radius=0.8, border=NULL, col=rainbow(10), density=30, main="title"): radius負で左端から
レーダーチャート	stars(x, key.loc = c(loc.x, loc.y)): key.locは凡例の表示位置を示す座標



R quick reference (string)

■文字列操作

部分文字列	substring(x, 1,10): x の1～10文字目を抽出
文字列の結合	paste(x, y, …, sep=" "): x, y, … をスペースで区切って連結、sep=""でスペースなしで連結
文字列の分割	strsplit(x, split=" "): スペースで分割（スペースは無くなる）、split=""を指定すると各文字に分解
文字列の置き換え	gsub("aaa", "bbb", strs): 文字列ベクトル strs の各文字列に含まれる "aaa" を "bbb" に置き換え 他に、最初の 1件だけを書き換える sub() や、1文字ずつ一括で置き換える chartr() がある
文字数のカウント	nchar(x): x の文字数を返す
文字列を含むかどうか	grep("aaa", strs): 文字列ベクトル strs に、"aaa"を含む文字列のインデックスを返す x <- c("b", "bc", "bcd"); grep("bc", x) を実行すると、2 3 が返る
文字列が含まれる位置	regexpr("aaa", strs): 文字列ベクトル strs に、"aaa"を含む文字列のマッチした位置を返す x <- c("b", "bc", "bcd"); regexpr("bc", x) を実行すると、-1 1 1 が返る
文字列の一致	match("aaa", strs): 文字列ベクトル strs に、完全に一致する文字列のインデックスを返す x <- c("b", "bc", "bcd"); match("bc", x) を実行すると、2 が返る