

度数分布表

質的変数

table(変数名)

prop.table(テーブル名)

table : 度数.

prop.table : 割合.

量的変数

library(fdth)

オブジェクト名 <- fdt(変数名, start=始まりの値, end=終わりの値, h=階級幅))

あらかじめ fdth パッケージをインストールしておく必要がある.

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",")
>
```

> # 質的データの度数分布表

> t1 <- table(d1\$学科)

> p1 <- prop.table(t1)*100

> rbind(t1, p1)

```
      医学      看護学      心理学
t1 76.00000 100.00000 94.00000
p1 28.14815  37.03704 34.81481
>
```

> # 量的変数の度数分布表

> library(fdth)

> (t1 <- fdt(d1\$自己効力感, start=25, end=80, h=5))

```
Class limits  f   rf rf(%)  cf  cf(%)
[25, 30)    4 0.01  1.48   4   1.48
[30, 35)    8 0.03  2.96  12   4.44
[35, 40)   29 0.11 10.74  41  15.19
[40, 45)   42 0.16 15.56  83  30.74
[45, 50)   54 0.20 20.00 137  50.74
[50, 55)   50 0.19 18.52 187  69.26
[55, 60)   40 0.15 14.81 227  84.07
[60, 65)   28 0.10 10.37 255  94.44
[65, 70)    9 0.03  3.33 264  97.78
[70, 75)    5 0.02  1.85 269  99.63
[75, 80)    1 0.00  0.37 270 100.00
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリウム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

クロス表

度数だけのクロス表

```
table(変数名1, 変数名2)
```

行周辺度数（行和）

```
margin.table(テーブル名, 1)
```

行和に対する割合

```
prop.table(テーブル名, 1)
```

列周辺度数（列和）

```
margin.table(テーブル名, 2)
```

列和に対する割合

```
prop.table(テーブル名, 2)
```

総周辺度数（総和）

```
margin.table(テーブル名)
```

総和に対する割合

```
prop.table(テーブル名)
```

次元名（変数名）の指定

```
table 関数でdnnオプションを使う
```

```
dnn=list("行名", "列名")
```

作成した表にあとから次元名を付ける

```
names(dimnames(テーブル名)) <- c("変数名1", "変数名2", ...)
```

次元名を指定すると、行名、列名を表示することができる（指定しないと表示されない）。

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("度数分布_データ.csv", header=TRUE, sep=",")
> head(d1)
  id sex score grade
1  1 f    85    A
2  2 f   100    A
3  3 m    80    A
4  4 f    80    A
5  5 f    65    C
6  6 m    80    A
>
```

> # 度数だけのクロス表

```
> t1 <- table(d1$sex, d1$grade, dnn=list("sex", "grade"))
```

```
> t1
```

```
      grade
sex      A      B      C      D
f         25     18      7      1
m         15     12      8      4
```

```
>
```

```
>
```

> # 行周辺度数（行和）

```
> margin.table(t1, 1)
```

```
sex
f      m
51     39
```

```
>
```

> # 行和に対する割合

```
> prop.table(t1, 1)
```

```
      grade
sex      A      B      C      D
f  0.49019608 0.35294118 0.13725490 0.01960784
m  0.38461538 0.30769231 0.20512821 0.10256410
```

	A	B	C	D
1	id	sex	score	grade
2	1	f	85	A
3	2	f	100	A
4	3	m	80	A
5	4	f	80	A
6	5	f	65	C
7	6	m	80	A
8	7	m	75	B
9	8	m	100	A
10	9	f	65	C
11	10	m	55	D
12	11	f	90	A
13	12	m	80	A
14	13	f	65	C
15	14	f	75	B
16	15	m	70	B
17	16	f	70	B
18	17	f	75	B
19	18	m	65	C
20	19	m	50	D
21	20	f	95	A

>

> # 列周辺度数(列和)
> margin.table(t1, 2)

```
grade
A      B      C      D
40     30     15      5
```

>

> # 列和に対する割合
> prop.table(t1, 2)

```
grade
sex    A      B      C      D
f      0.6250000 0.6000000 0.4666667 0.2000000
m      0.3750000 0.4000000 0.5333333 0.8000000
```

>

> # 総周辺度数(総和)
> margin.table(t1)

```
[1] 90
```

>

> # 総和に対する割合
> prop.table(t1)

```
grade
sex    A      B      C      D
f      0.27777778 0.20000000 0.07777778 0.01111111
m      0.16666667 0.13333333 0.08888889 0.04444444
```

>

>

>

割合なども入ったクロス表

gmodels パッケージの CrossTable 関数を使う方法

```
library(gmodels)
CrossTable(変数1, 変数2)
```

あらかじめ gmodels パッケージをインストールしておく必要がある。
 expected=TRUE とすると、期待度数と、カイ 2 乗値検定の結果を表示する。
 fisher=TRUE や mcnemar=TRUE とすると、フィッシャーの直接検定やマクネマーの検定をしてくれる。

descr パッケージの CrossTable 関数を使う方法

```
library(descr)
CrossTable(変数1, 変数2)
```

あらかじめ descr パッケージをインストールしておく必要がある。
 デフォルトで期待値の表示と、カイ 2 乗検定をしてくれる。
 fisher=TRUE や mcnemar=TRUE とすると、フィッシャーの直接検定やマクネマーの検定をしてくれる。

オリジナル関数を使う方法

```
cross.table(変数名1, 変数名2)
```

R 起動後に、一度関数部分を実行しておく必要がある。
 digit=桁数で、セルパーセントの有効桁数を指定できる。

```
> setwd("i:\\Rdocuments\\scripts\\")
> dl <- read.table("度数分布_データ.csv", header=TRUE, sep=",")
>
> table(dl$sex, dl$grade)
      A      B      C      D
f      25     18      7      1
m      15     12      8      4
>
```

```
> #gmodels パッケージの CrossTable 関数を使う方法
> library(gmodels)
> CrossTable(dl$sex, dl$grade, expected=TRUE)
```

Cell Contents

	N
	Expected N
Chi-square contribution	
N / Row Total	
N / Col Total	
N / Table Total	

Total Observations in Table: 90

	A	B	C	D
1	id	sex	score	grade
2	1	f	85	A
3	2	f	100	A
4	3	m	80	A
5	4	f	80	A
6	5	f	65	C
7	6	m	80	A
8	7	m	75	B
9	8	m	100	A
10	9	f	65	C
11	10	m	55	D
12	11	f	90	A
13	12	m	80	A
14	13	f	65	C
15	14	f	75	B
16	15	m	70	B
17	16	f	70	B
18	17	f	75	B
19	18	m	65	C
20	19	m	50	D
21	20	f	95	A

dl\$sex	dl\$grade	A	B	C	D	Row Total
f		25	18	7	1	51
		22.667	17.000	8.500	2.833	
		0.240	0.059	0.265	1.186	
		0.490	0.353	0.137	0.020	0.567
		0.625	0.600	0.467	0.200	
		0.278	0.200	0.078	0.011	
m		15	12	8	4	39
		17.333	13.000	6.500	2.167	
		0.314	0.077	0.346	1.551	
		0.385	0.308	0.205	0.103	0.433
		0.375	0.400	0.533	0.800	
		0.167	0.133	0.089	0.044	
Column Total		40	30	15	5	90
		0.444	0.333	0.167	0.056	

Statistics for All Table Factors

Pearson's Chi-squared test

Chi² = 4.038462 d.f. = 3 p = 0.2573408

警告メッセージ:

In chisq.test(t, correct = FALSE, ...) :
Chi-squared approximation may be incorrect

```
> detach("package:gmodels")
>
>
```

```
> #descr パッケージの CrossTable関数を使う方法
> library(descr)
> (ctdl <-CrossTable(d1$sex, d1$grade))
```

Cell Contents

N
Expected N
Chi-square contribution
N / Row Total
N / Col Total
N / Table Total

d1\$sex	d1\$grade				Total
	A	B	C	D	
f	25	18	7	1	51
	22.7	17.0	8.5	2.8	
	0.240	0.059	0.265	1.186	
	0.490	0.353	0.137	0.020	0.567
	0.625	0.600	0.467	0.200	
	0.278	0.200	0.078	0.011	
m	15	12	8	4	39
	17.3	13.0	6.5	2.2	
	0.314	0.077	0.346	1.551	
	0.385	0.308	0.205	0.103	0.433
	0.375	0.400	0.533	0.800	
	0.167	0.133	0.089	0.044	
Total	40	30	15	5	90
	0.444	0.333	0.167	0.056	

Statistics for All Table Factors

Pearson's Chi-squared test

Chi² = 4.038462 d.f. = 3 p = 0.2573408

警告メッセージ:

In chisq.test(t, correct = FALSE, ...) :
Chi-squared approximation may be incorrect

オリジナル関数を使う方法

```
#-----
#
cross.table <- function(var1, var2, digit=1){
  tt <- table(var1,var2); rn.tt <- rownames(tt); cn.tt <- colnames(tt)
#
  t1 <- margin.table(tt,1); ta <- margin.table(tt); t1 <- c(t1,ta)
  t2 <- margin.table(tt,2); tt <- rbind(tt,t2); tt <- cbind(tt,t1)
  rn.tt <- c(rn.tt, "Total"); cn.tt <- c(cn.tt, "Total")
#
  tt.1 <- round((prop.table(tt,1) * 200), digit)
  tt.2 <- round((prop.table(tt,2) * 200), digit)
  tt.a <- round((prop.table(tt) * 400), digit)
#
  nr.tt <- nrow(tt); nc.tt <- ncol(tt)
  m.tt <- matrix(c(0), (4*nr.tt), nc.tt)
  rn.m.tt <- c(1: (4*nr.tt))
#
  for (i in 1:nr.tt){
    j <- 4*i - 3
    m.tt[j,] <- tt[i,]; m.tt[j+1,] <- tt.1[i,];
    m.tt[j+2,] <- tt.2[i,]; m.tt[j+3,] <- tt.a[i,];
    rn.m.tt[j] <- rn.tt[i]; rn.m.tt[j+1] <- "row prpp";
    rn.m.tt[j+2] <- "col prop"; rn.m.tt[j+3] <- "prop"
  }
#
  rownames(m.tt) <- rn.m.tt; colnames(m.tt) <- cn.tt
  return(round(m.tt, 3))
}
#-----
```

```
> (class.seibetsu <- cross.table(d1$sex, d1$grade))
      A      B      C      D      Total
f      25.0   18.0    7.0    1.0   51.0
row prpp 49.0   35.3   13.7    2.0  100.0
col prop 62.5   60.0   46.7   20.0   56.7
prop    27.8   20.0    7.8    1.1   56.7
m      15.0   12.0    8.0    4.0   39.0
row prpp 38.5   30.8   20.5   10.3  100.0
col prop 37.5   40.0   53.3   80.0   43.3
prop    16.7   13.3    8.9    4.4   43.3
Total   40.0   30.0   15.0    5.0   90.0
row prpp 44.4   33.3   16.7    5.6  100.0
col prop 100.0  100.0  100.0  100.0  100.0
prop    44.4   33.3   16.7    5.6  100.0
>
```

```
> (class.seibetsu <- cross.table(d1$sex, d1$grade, digit=0))
      A      B      C      D      Total
f      25    18     7     1     51
row prpp 49    35    14     2    100
col prop 62    60    47    20     57
prop    28    20     8     1     57
m      15    12     8     4     39
row prpp 38    31    21    10    100
col prop 38    40    53    80     43
prop    17    13     9     4     43
Total   40    30    15     5     90
row prpp 44    33    17     6    100
col prop 100   100   100   100   100
prop    44    33    17     6    100
>
```

群別のクロス表

度数だけのクロス表

table(変数名1, 変数名2, 群分け変数名)

周辺度数

新データフレーム名 <- データフレーム名[データフレーム名\$群分け変数==値,]

margin.table(新データフレーム名)

割合

新データフレーム名 <- データフレーム名[データフレーム名\$群分け変数==値,]

prop.table(新データフレーム名)

```
> setwd("i:\\Rdocuments\\scripts\\")
> d1 <- read.table("連関係数_データ.csv", header=TRUE, sep=",")
>
```

> #度数だけのクロス表

```
> (t1 <- table(d1$method, d1$grade, d1$sex))
```

```
, , = F
      A  B  C  D
1  5 19  5  1
2 12 13  4  5
```

```
, , = M
      A  B  C  D
1  5 10  1  2
2  9  5  4  0
```

```
>
> d1F <- d1[d1$sex=="F",]
> d1M <- d1[d1$sex=="M",]
> t1F <- table(d1F$method, d1F$grade)
> t1M <- table(d1M$method, d1M$grade)
>
```

> #行周辺度数, 行和に対する割合

```
> margin.table(t1F, 1)
```

```
1 2
30 34
> prop.table(t1F, 1)
```

```
      A      B      C      D
1 0.1666667 0.6333333 0.1666667 0.0333333
2 0.3529412 0.3823529 0.1176471 0.1470588
```

> #列周辺度数, 列和に対する割合

```
> margin.table(t1F, 2)
```

```
 A  B  C  D
17 32  9  6
> prop.table(t1F, 2)
```

```
      A      B      C      D
1 0.2941176 0.5937500 0.5555556 0.1666667
2 0.7058824 0.4062500 0.4444444 0.8333333
```

> #総周辺度数, 総和に対する割合

```
> margin.table(t1F)
```

```
[1] 64
> prop.table(t1F)
```

```
      A      B      C      D
1 0.078125 0.296875 0.078125 0.015625
2 0.187500 0.203125 0.062500 0.078125
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	F	1	A
6	5	M	1	B
7	6	M	2	A
8	7	F	2	B
9	8	F	2	B
10	9	F	2	B
11	10	F	1	B
12	11	M	2	B
13	12	F	2	A
14	13	M	1	D
15	14	F	2	C
16	15	F	2	D
17	16	F	2	B
18	17	F	2	A
19	18	F	2	B
20	19	F	2	B
21	20	F	1	B

フラットなクロス表

元データから作成する場合

```
ftable(データフレーム名[, c("変数名1", "変数名2", ...)], row.vars=c("変数名1", "変数名2", ...))
```

row.vars で行にカテゴリを並べる変数を指定する. col.vars というオプションをある.

ftable 関数で作られる表では, カテゴリ名が初回のみ表示され, 2回目以降は省略される.

2回目以降もカテゴリ名を表示するには, ftable で作成した表をデータフレーム化する. その際, 後ろの変数から順にソートされてしまうので, 前の変数からソートし直すと比較しやすい.

既にあるクロス表から作成する場合

```
ftable(列変数名 ~ 行変数名1 + 行変数名2 + ..., data=データフレーム名)
```

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("連関係数_データ.csv", header=TRUE, sep=",")
> head(d1)
  id sex method grade
1  1  F      1     B
2  2  M      1     B
3  3  F      2     A
4  4  F      1     A
5  5  M      1     B
6  6  M      2     A
>
>
```

> # クロス表

```
> (t1 <- table(d1[, c("method", "grade", "sex")], dnn=list("method", "grade", "sex")))
, , sex = F
```

```
      grade
method A B C D
  1  5 19 5 1
  2 12 13 4 5
```

```
, , sex = M
```

```
      grade
method A B C D
  1  5 10 1 2
  2  9  5 4 0
```

```
>
> # 既にあるクロス表からフラットなクロス表を作成
> # 性別で表が分離していないことに注意
```

```
> ftable(grade ~ sex + method, data=t1)
      grade
sex method A B C D
```

```
F  1      5 19 5 1
   2     12 13 4 5
M  1      5 10 1 2
   2      9  5 4 0
```

```
>
>
```

> # フラットなクロス表

```
> # カテゴリ表示が省略されていることに注意
```

```
> (ft1 <- ftable(d1[, c("sex", "method", "grade")], row.vars=c("sex", "method", "grade")))
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	F	1	A
6	5	M	1	B
7	6	M	2	A
8	7	F	2	B
9	8	F	2	B
10	9	F	2	B
11	10	F	1	B
12	11	M	2	B
13	12	F	2	A
14	13	M	1	D
15	14	F	2	C
16	15	F	2	D
17	16	F	2	B
18	17	F	2	A
19	18	F	2	B
20	19	F	2	B
21	20	F	1	B


```
sex method grade
F 1 A 5
   B 19
   C 5
   D 1
  2 A 12
   B 13
   C 4
   D 5
M 1 A 5
   B 10
   C 1
   D 2
  2 A 9
   B 5
   C 4
   D 0
>
```

```
>
> # フラットなクロス表をデータ行列に変換
> # カテゴリがすべて表示されていることに注意
> (fd1 <- as.data.frame(ft1))
```

```
   sex method grade Freq
1    F      1     A     5
2    M      1     A     5
3    F      2     A    12
4    M      2     A     9
5    F      1     B    19
6    M      1     B    10
7    F      2     B    13
8    M      2     B     5
9    F      1     C     5
10   M      1     C     1
11   F      2     C     4
12   M      2     C     4
13   F      1     D     1
14   M      1     D     2
15   F      2     D     5
16   M      2     D     0
```

```
> # ソートし直す
```

```
> (fd1 <- fd1[order(fd1$sex, fd1$method, fd1$grade),])
```

```
   sex method grade Freq
1    F      1     A     5
5    F      1     B    19
9    F      1     C     5
13   F      1     D     1
3    F      2     A    12
7    F      2     B    13
11   F      2     C     4
15   F      2     D     5
2    M      1     A     5
6    M      1     B    10
10   M      1     C     1
14   M      1     D     2
4    M      2     A     9
8    M      2     B     5
12   M      2     C     4
16   M      2     D     0
>
>
```

フラットなクロス表から通常のクロス表を作成

`xtabs`(度数変数名 ~ 変数名1 + 変数名2 + ..., `data`=フラットなクロス表名)

フラットなクロス表は、`fTable` で作成した表でも、データフレームでもよい。

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("連関係数_データ.csv", header=TRUE, sep=",")
> head(d1)
  id sex method grade
1  1  F      1      B
2  2  M      1      B
3  3  F      2      A
4  4  F      1      A
5  5  M      1      B
6  6  M      2      A
>
>
>
> # フラットなクロス表
> (ft1 <- fTable(d1[,c("sex", "method", "grade")], row.vars=c("sex", "method", "grade")))
```

```
sex method grade
F    1      A      5
      B     19
      C      5
      D      1
      2      A     12
      B     13
      C      4
      D      5
M    1      A      5
      B     10
      C      1
      D      2
      2      A      9
      B      5
      C      4
      D      0
```

```
>
>
> # フラットなクロス表をデータ行列に変換
> (fd1 <- as.data.frame(ft1))
```

```
1  F      1  A      5
2  M      1  A      5
3  F      2  A     12
4  M      2  A      9
5  F      1  B     19
6  M      1  B     10
7  F      2  B     13
8  M      2  B      5
9  F      1  C      5
10 M      1  C      1
11 F      2  C      4
12 M      2  C      4
13 F      1  D      1
14 M      1  D      2
15 F      2  D      5
16 M      2  D      0
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	F	1	A
6	5	M	1	B
7	6	M	2	A
8	7	F	2	B
9	8	F	2	B
10	9	F	2	B
11	10	F	1	B
12	11	M	2	B
13	12	F	2	A
14	13	M	1	D
15	14	F	2	C
16	15	F	2	D
17	16	F	2	B
18	17	F	2	A
19	18	F	2	B
20	19	F	2	B
21	20	F	1	B

> # フラットなクロス表からクロス表を作成
 > # フラットなクロス表でも、それをデータフレーム化したものでも、クロス表が作成できる

> xtabs(Freq~method+grade+sex, data=ft1)

, , sex = F

	grade			
method	A	B	C	D
1	5	19	5	1
2	12	13	4	5

, , sex = M

	grade			
method	A	B	C	D
1	5	10	1	2
2	9	5	4	0

> xtabs(Freq~method+grade+sex, data=fd1)

, , sex = F

	grade			
method	A	B	C	D
1	5	19	5	1
2	12	13	4	5

, , sex = M

	grade			
method	A	B	C	D
1	5	10	1	2
2	9	5	4	0

>

円グラフ

円グラフ

pie(テーブル名, clockwise=TRUE, col=色名)

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学   女          高          49        23 就職
2    2      20Y2 心理学   男          低          57        29 就職
3    3      20Y1  医学   女          高          42        23 進学
4    4      20Y1 看護学   女          高          41        23 就職
5    5      20Y2  医学   男          低          41        22 就職
6    6      20Y1 心理学   女          低          47        24 就職
```

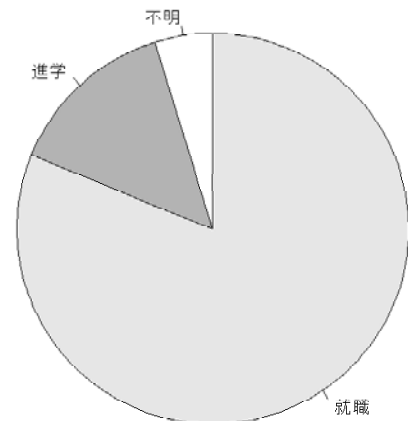
```
> #度数
> (t1 <- table(d1$進路))
```

```
就職 進学 不明
219   38   13
>
```

```
> # 円グラフ
> par(mar=c(1, 1, 1, 1))
> pie(t1, clockwise=TRUE, col=gray(c(0.9, 0.7, 1.0)), cex=1.4)
>
```

```
> #度数の割合
> (p1 <- prop.table(t1))
```

```
      就職      進学      不明
0.8111111 0.1407407 0.04814815
>
```



	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

帯グラフ

帯グラフ

```
barplot(as.matrix(割合テーブル名), horiz=TRUE, beside=FALSE,
        ylim=c(0, 1), width=0.3, col=色名, legend.text=TRUE)
```

ylim: 帯グラフ領域全体の高さ
width: 帯の幅
cex: 文字の大きさの調整
beside: 従属変数の水準別に棒を分けるか
horizontal: 横向きグラフ
las: 軸ラベルの向き. 1は水平
segments: 直線を引く

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",")
> head(d1)
  番号 入学年度 学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学 女          高          49        23 就職
2    2      20Y2 心理学 男          低          57        29 就職
3    3      20Y1 医学 女          高          42        23 進学
4    4      20Y1 看護学 女          高          41        23 就職
5    5      20Y2 医学 男          低          41        22 就職
6    6      20Y1 心理学 女          低          47        24 就職
>
```

```
> #度数
> (t1 <- table(d1$進路))
```

```
就職 進学 不明
219   38   13
>
```

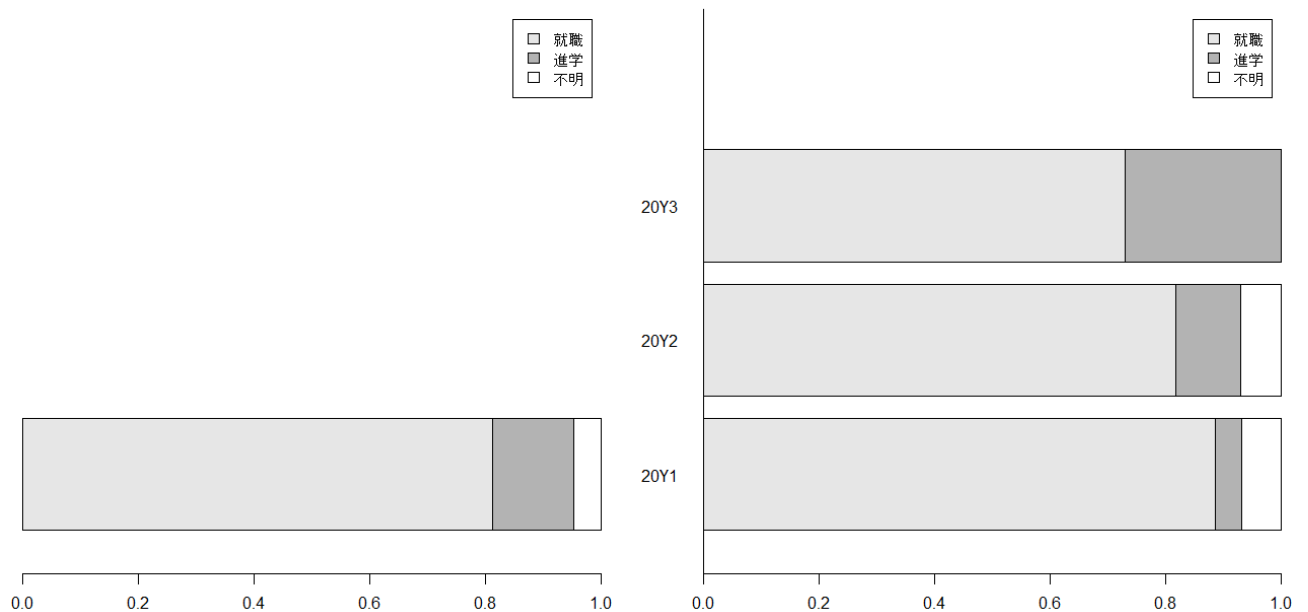
```
> #度数の割合
> (p1 <- prop.table(t1))
```

```
      就職      進学      不明
0.81111111 0.14074074 0.04814815
>
```

```
> # 帯グラフ
> # 度数の割合ベクトルを行列に変換するのがポイント
> par(mar=c(3, 4, 3, 1))
> barplot(as.matrix(p1), horiz=TRUE, beside=FALSE, las=1,
+         ylim=c(0, 1.4), width=0.3, col=gray(c(0.9, 0.7, 1.0)), legend.text=T)
>
```

```
> # 年度別の帯グラフ
> # 行ごとに割合を算出
> t2 <- table(d1$進路, d1$入学年度)
> p2 <- prop.table(t2, 2)
> par(mar=c(3, 4, 3, 1))
> barplot(as.matrix(p2), horiz=TRUE, beside=FALSE, las=1,
+         ylim=c(0, 1.4), width=0.3, col=gray(c(0.9, 0.7, 1.0)), legend.text=T)
> segments(0, -1, 0, 2)
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラリウム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職



棒グラフ

barplot(変数 (または行列) 名, ...)

群別 (変数別) のグラフの比較

barplot(変数 (または行列) 名, beside=TRUE, legend=TRUE, space=棒の間の大きさ)

オプション

width : 棒の幅を設定する (横軸に関する他のオプションと一緒に使って初めて有効となる)

space : 棒間の間隔を設定する

offset : 縦軸の開始点を設定する. 表示するデータからも, 当該量を引いておく必要がある. ほかにも, 文字の大きさを変えるオプションなど, いろいろ使える.

```
> setwd("i:\\Rdocuments\\scripts\\")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学   女           高          49        23 就職
2    2      20Y2 心理学   男           低          57        29 就職
3    3      20Y1  医学   女           高          42        23 進学
4    4      20Y1 看護学   女           高          41        23 就職
5    5      20Y2  医学   男           低          41        22 就職
6    6      20Y1 心理学   女           低          47        24 就職
>
```

> # 度数

```
> (t1 <- table(d1$進路))
```

```
就職 進学 不明
219   38   13
```

> # 棒グラフ

```
> par(mar=c(5, 5, 3, 1))
> barplot(as.matrix(t1), xlim=c(0, 5), ylim=c(0, 250), ylab="Frequency",
+ las=1, beside=TRUE, space=0.5, col=gray(c(0.9, 0.7, 1.0)), legend=TRUE)
> segments(-1, 0, 5, 0)
>
```

> # 学科 × 進路のクロス表

```
> (t2 <- table(d1$学科, d1$進路))
```

```
      就職 進学 不明
医学      64   10    2
看護学    80   13    7
心理学    75   15    4
>
```

> # 進路ごとに学科を比較した棒グラフ

```
> par(mar=c(5, 5, 3, 1))
> barplot(as.matrix(t2), xlim=c(0, 14), ylim=c(0, 80), ylab="Frequency",
+ las=1, beside=TRUE, col=gray(c(0.9, 0.7, 1.0)), legend=TRUE)
> segments(-1, 0, 14, 0)
>
```

> # 進路 × 学科のクロス表

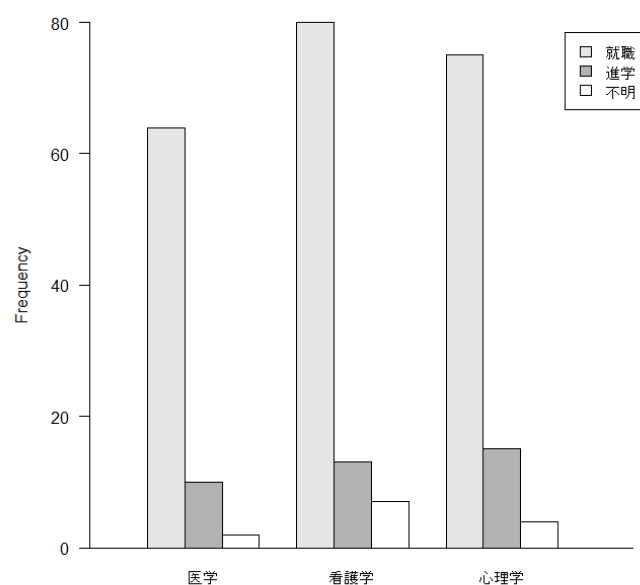
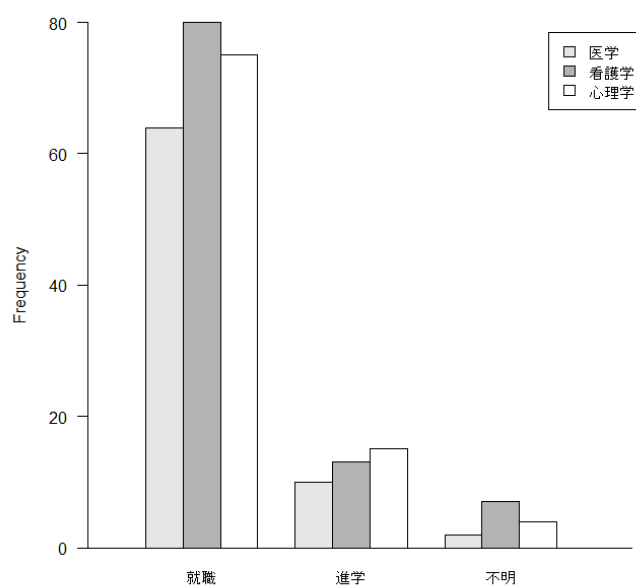
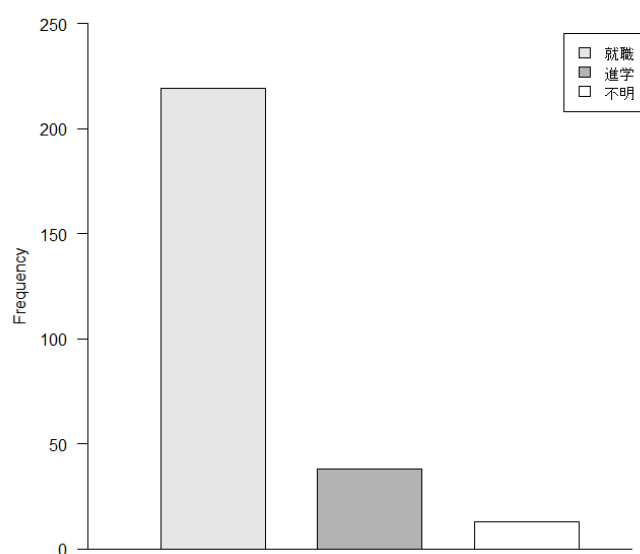
```
> (t3 <- table(d1$進路, d1$学科))
```

```
      医学 看護学 心理学
就職      64     80     75
進学      10     13     15
不明       2      7      4
>
```

> # 学科ごとに進路を比較した棒グラフ

```
> par(mar=c(5, 5, 3, 1))
> barplot(as.matrix(t3), xlim=c(0, 14), ylim=c(0, 80), ylab="Frequency",
+ las=1, beside=TRUE, col=gray(c(0.9, 0.7, 1.0)), legend=TRUE)
> segments(-1, 0, 14, 0)
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モトリウム自己効力感	学習意欲	進路	
2		1 20Y1	看護学	女	高	49	23	就職
3		2 20Y2	心理学	男	低	57	29	就職
4		3 20Y1	医学	女	高	42	23	進学
5		4 20Y1	看護学	女	高	41	23	就職
6		5 20Y2	医学	男	低	41	22	就職
7		6 20Y1	心理学	女	低	47	24	就職
8		7 20Y3	看護学	女	高	41	26	就職
9		8 20Y2	医学	男	低	43	29	就職
10		9 20Y2	看護学	女	低	54	21	就職
11		10 20Y3	医学	女	高	55	30	進学
12		11 20Y1	心理学	男	低	66	25	就職
13		12 20Y1	心理学	女	低	38	30	就職
14		13 20Y1	心理学	女	低	45	29	就職
15		14 20Y1	医学	男	高	53	23	就職
16		15 20Y1	心理学	男	高	47	25	就職




```
> # 棒の間隔を調整したグラフを描く方法
> # space オプションを利用する
```

```
> # あるテスト得点について、性別、自己効力感別（3段階）、および、学校別の平均値が以下だったとする.
```

```
>
```

```
> # 性別
```

```
> mF <- 63; mM <- 61
```

```
>
```

```
> # 自己効力感
```

```
> m1 <- 60; m2 <- 64; m3 <- 62
```

```
>
```

```
> # 学校別
```

```
> mA <- 61; mB <- 63; mC <- 62; mD <- 62
```

```
>
```

```
> # 各要因の平均値の棒グラフを、同時に表示する
```

```
> mgraph <- c(mF, mM, m1, m2, m3, mA, mB, mC, mD)
```

```
> names(mgraph) <- c("F", "M", "1", "2", "3", "A", "B", "C", "D")
```

```
>
```

```
> barplot(mgraph, xlim=c(1, 18), ylim=c(0, 70), ylab="Mean",
+ las=1, col=gray(c(0.4, 1.0, 0.4, 0.7, 1.0, 0.4, 0.6, 0.8, 1.0)), cex.names=0.8,
+ legend=F, width=1.2, space=c(2, 0, 2, 0, 0, 2, 0, 0, 0), main="各群の平均点")
> segments(0, 0, 20, 0)
>
```

```
> # 縦軸の原点をずらした図を描く方法
> # offsetオプションを利用する
```

```
> # テスト得点を間隔尺度と考えると、縦軸を0から始める必要はなく、
```

```
> # もっと見やすいグラフを書くことも考えられる.
```

```
> # 縦軸の範囲を、50 から 70 にすることを考える.
```

```
>
```

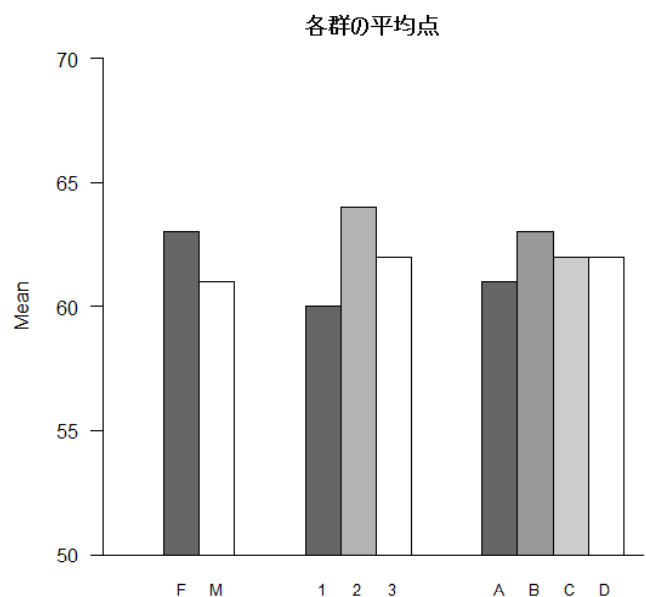
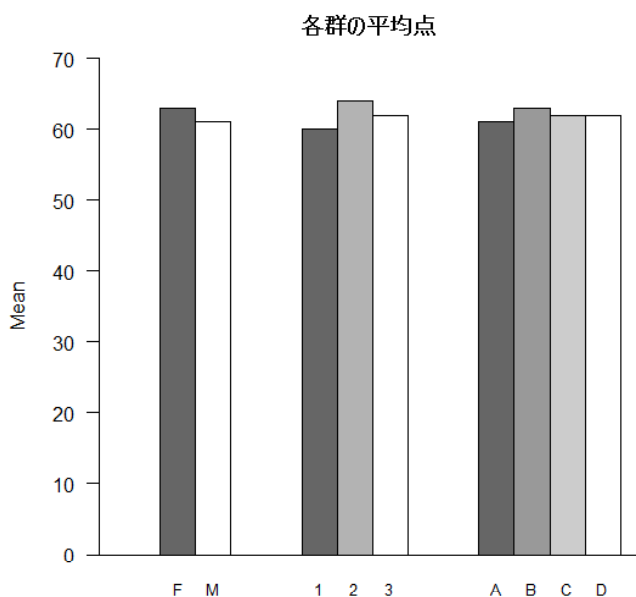
```
> voffset <- 50
```

```
> mgraph <- (c(mF, mM, m1, m2, m3, mA, mB, mC, mD) - voffset)
```

```
> names(mgraph) <- c("F", "M", "1", "2", "3", "A", "B", "C", "D")
```

```
>
```

```
> barplot(mgraph, xlim=c(1, 18), ylim=c(voffset, 70), ylab="Mean", offset=voffset,
+ las=1, col=gray(c(0.4, 1.0, 0.4, 0.7, 1.0, 0.4, 0.6, 0.8, 1.0)), cex.names=0.8,
+ legend=F, width=1.2, space=c(2, 0, 2, 0, 0, 2, 0, 0, 0), main="各群の平均点")
> segments(0, voffset, 20, voffset)
>
```



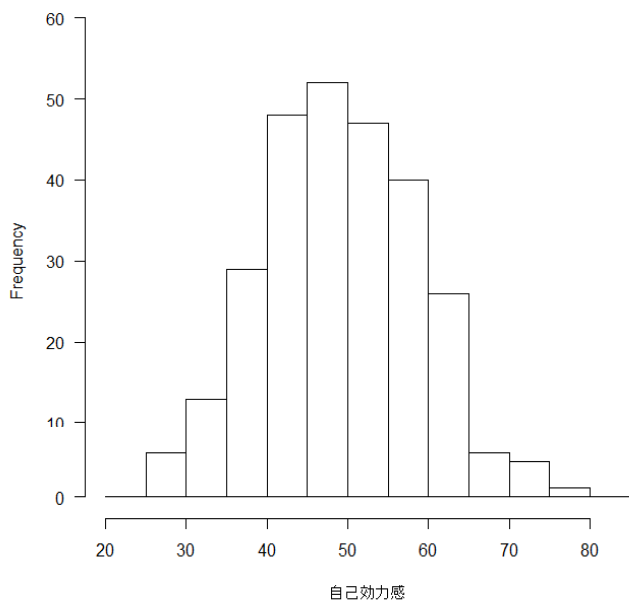
ヒストグラム

hist(変数名, オプション群)

breaks=seq(a, b, by=c) : x軸の値を, aからbまで増分cで区切っていく
 ylim=c(a, b) : y軸の範囲をaからbまでとする.
 las=1 : 目盛りの値を水平にする.
 xlab="aa" : x軸のラベルをaaとする.

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",")
> head(d1)
  番号 入学年度 学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1    20Y1 看護学 女      高           49      23 就職
2    2    20Y2 心理学 男      低           57      29 就職
3    3    20Y1 医学 女      高           42      23 進学
4    4    20Y1 看護学 女      高           41      23 就職
5    5    20Y2 医学 男      低           41      22 就職
6    6    20Y1 心理学 女      低           47      24 就職
>
>
> # ヒストグラム
> par(mar=c(5, 5, 3, 1))
> hist(d1$自己効力感, breaks=seq(20, 85, by=5), ylim=c(0, 60), las=1, xlab="自己効力感", main="")
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職



箱ひげ図

1変数の箱ひげ図

boxplot(変数名)

群別の箱ひげ図

boxplot(変数名~群変数名, data=データフレーム名)

クラスごとの群別箱ひげ図

boxplot(変数名~群変数名, data=データフレーム名, subset=クラス変数名==クラス値)

対応のある変数の箱ひげ図

boxplot(対応のある変数を入れたデータフレーム名)

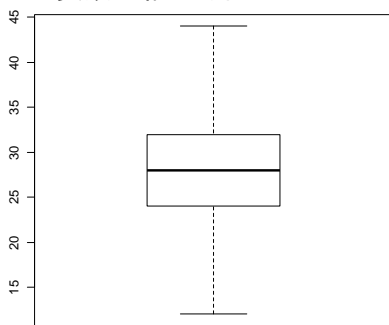
```

> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("箱ひげ図_データ.csv", header=TRUE, sep=",")
>
> #1変数の箱ひげ図
> boxplot(d1$pre)
>
> #群別の箱ひげ図
> boxplot(pre~seibetsu, data=d1)
>
> #クラスごとの群別箱ひげ図
> boxplot(pre~seibetsu, data=d1, subset=class=="A", main="class=A")
> boxplot(pre~seibetsu, data=d1, subset=class=="B", main="class=B")
>
> #対応のある変数の箱ひげ図
> d2 <- d1[,c("pre", "post1", "post2")]
> boxplot(d2)
>

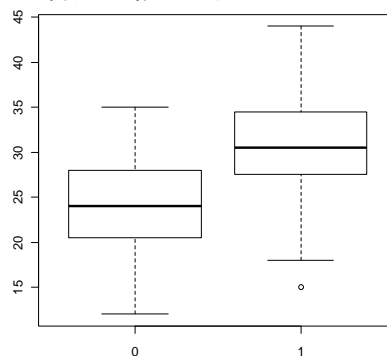
```

	A	B	C	D	E
1	class	seibetsu	pre	post1	post2
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	24	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

1変数の箱ひげ図

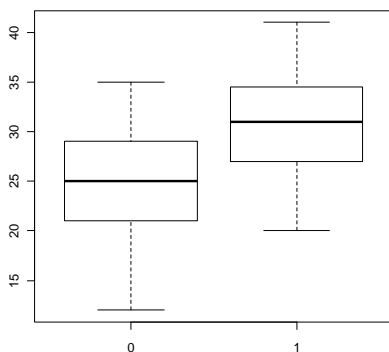


群別の箱ひげ図

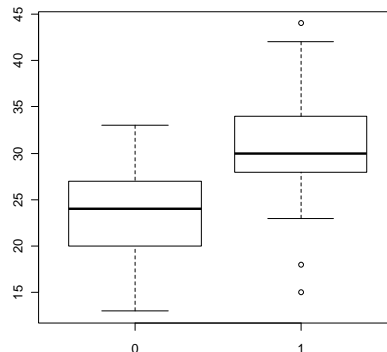


クラスごとの群別箱ひげ図

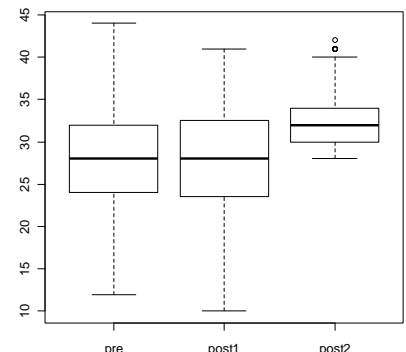
class=A



class=B



対応のある変数の箱ひげ図



折れ線グラフ

plot(変数ベクトル名, type="タイプ名", オプション名)

type : 1…折れ線, b…○と線分(分離), o…○と線分(結合), c…分離した折れ線
 lty : 線種. 1実線. 2破線. 3点線.
 lwd : 線の太さ.
 xaxt="n" : x軸とラベルを表示しない.
 yaxt="n" : y軸とラベルを表示しない.

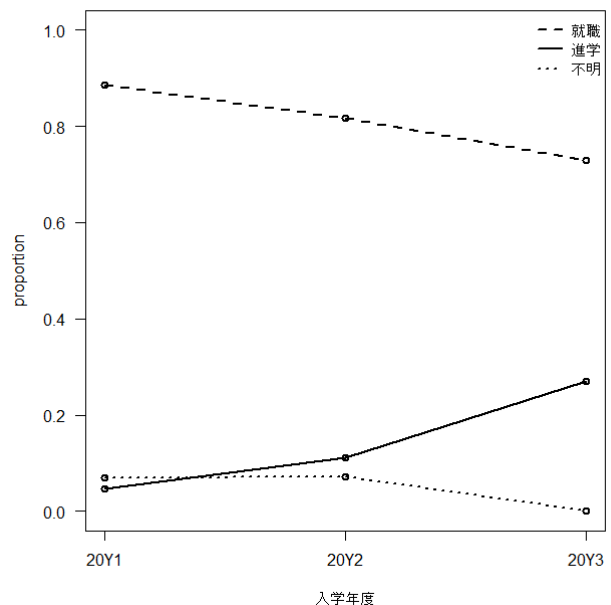
```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",")
> head(d1)
```

	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23	就職
2	2	20Y2	心理学	男	低	57	29	就職
3	3	20Y1	医学	女	高	42	23	進学
4	4	20Y1	看護学	女	高	41	23	就職
5	5	20Y2	医学	男	低	41	22	就職
6	6	20Y1	心理学	女	低	47	24	就職

```
> # 折れ線グラフ
> par(mar=c(5, 5, 3, 3))
> t1 <- table(d1$進路, d1$入学年度)
> p1 <- prop.table(t1, 2)
> plot(p1["就職", ], type="o", ylim=c(0, 1), las=1, xlab="入学年度", ylab="proportion", lty=2, lwd=2, xaxt="n")
> par(new=TRUE)
> plot(p1["進学", ], type="o", ylim=c(0, 1), xlab="", ylab="", lty=1, lwd=2, xaxt="n", yaxt="n")
> par(new=TRUE)
> plot(p1["不明", ], type="o", ylim=c(0, 1), xlab="", ylab="", lty=3, lwd=2, xaxt="n", yaxt="n")
> axis(1, c(1, 2, 3), colnames(p1))
>

> # 凡例
> text(3, 1.0, "就職")
> text(3, 0.96, "進学")
> text(3, 0.92, "不明")
> segments(2.8, 1.0, 2.9, 1.0, lty=2, lwd=2)
> segments(2.8, .96, 2.9, .96, lty=1, lwd=2)
> segments(2.8, .92, 2.9, .92, lty=3, lwd=2)
>
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2		1 20Y1	看護学	女	高	49	23	就職
3		2 20Y2	心理学	男	低	57	29	就職
4		3 20Y1	医学	女	高	42	23	進学
5		4 20Y1	看護学	女	高	41	23	就職
6		5 20Y2	医学	男	低	41	22	就職
7		6 20Y1	心理学	女	低	47	24	就職
8		7 20Y3	看護学	女	高	41	26	就職
9		8 20Y2	医学	男	低	43	29	就職
10		9 20Y2	看護学	女	低	54	21	就職
11		10 20Y3	医学	女	高	55	30	進学
12		11 20Y1	心理学	男	低	66	25	就職
13		12 20Y1	心理学	女	低	38	30	就職
14		13 20Y1	心理学	女	低	45	29	就職
15		14 20Y1	医学	男	高	53	23	就職
16		15 20Y1	心理学	男	高	47	25	就職



散布図

2変数の散布図

`plot(x軸変数名, y軸変数名, pch=マーク番号(または"記号"))`

多変量の散布図

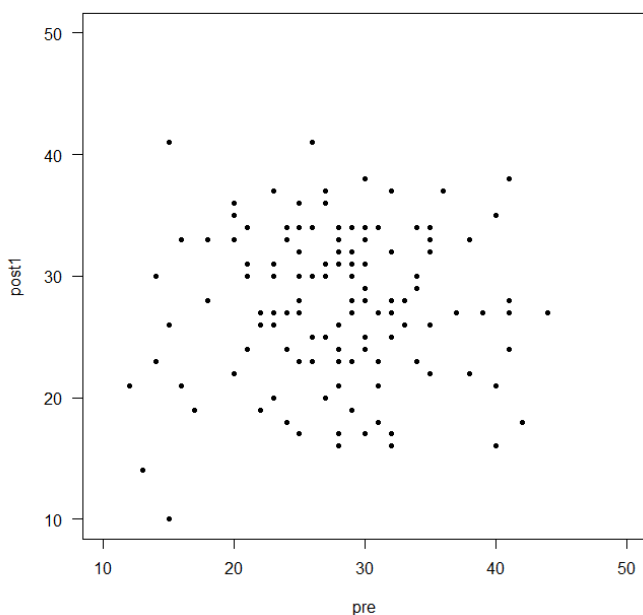
`pairs(データフレーム (または行列) 名, pch=マーク番号(または"記号"))`

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("グラフ_データ.csv", header=TRUE, sep=",")
> head(d1)
  class seibetsu pre post1 post2
1     A         0  29   34   30
2     A         0  28   24   32
3     A         1  30   24   33
4     A         1  30   28   31
5     A         1  33   26   34
6     A         1  29   28   34
>
```

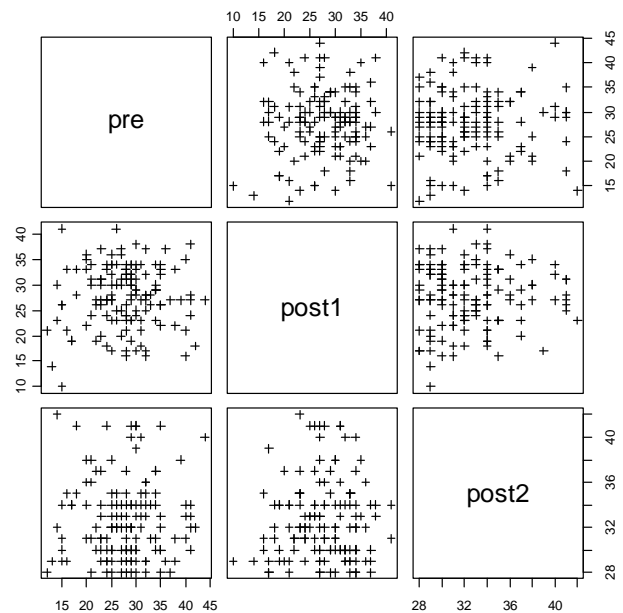
	A	B	C	D	E
1	class	seibetsu	pre	post1	post2
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	24	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

```
> #2変数の散布図
> plot(d1$pre, d1$post1, pch=20, las=1, xlim=c(10,50), ylim=c(10,50),
+       xlab="pre", ylab="post1")
>
>
> #多変量の散布図
> d2 <- d1[,c("pre", "post1", "post2")]
> pairs(d2, pch=3)
>
```

2変数の散布図



多変量の散布図



グラフィックオプション

par
「ヘルプ」→「Rの関数(テキスト)」で「**par**」を検索すると、グラフに関するさまざまなオプションが表示される。以下はその中の一部である。

図の配列 **par(mfrow=c(行数, 列数))** 図を何行何列に配列するかを指定する。

マーク **pch=マーク番号 または"記号"**
番号 (0~25, 33~126?) により, ○, ●, △, ▲, □, ■, ◆, + などの記号が選べる。

色 **col="色名"**
colors() とすると色名がたくさん表示される。black, blue, gray, green, pink, red, yellow など。

タイトル **main="タイトル"**

軸ラベル **xlab="x軸ラベル", ylab="y軸ラベル"**

軸ラベルの向き **las=0, 1, 2, or 3** (軸に平行, 水平, 軸に垂直, 垂直)

軸の表示 **axes=TRUE** (デフォルト)。FALSEにすると軸を自動的に書かない。

軸の範囲 **xlim=c(下限, 上限), ylim=c(下限, 上限)**

軸を書く **axis()** 自動設定しないで手動で書く。
目盛りや目盛りの値を指定できる。side=1:x軸下, 2:y軸左, 3:x軸上, 4:y軸右

棒の幅 **width=幅の大きさ** (xlim, ylimを指定時に有効)

横書き **horiz=TRUE** (デフォルトはFALSE)

並列配置 **beside=TRUE** (FALSEにすると積み上げグラフ)

凡例 **legend=TRUE** (FALSEにすると凡例非表示)

グラフの重ね合わせ **add=TRUE** (グラフオプション)
または **par(new=TRUE)** (コマンド)

他にもいろいろなオプションがある。

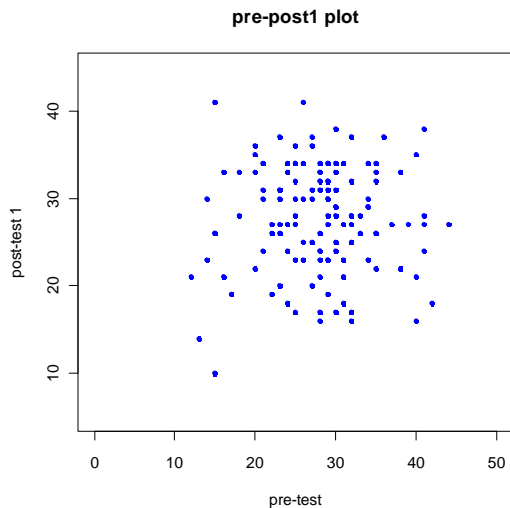
	A	B	C	D	E
1	class seibetsu	pre	post1	post2	
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	24	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

```
> setwd("i:\\Rdocuments\\scripts\\")
> d1 <- read.table("グラフ_データ.csv", header=TRUE, sep=",")
>
> #データの準備
> d1a <- d1[d1$class=="A",]
> d1b <- d1[d1$class=="B",]
>
> m3 <- sapply(d1[,c("pre", "post1", "post2")], tapply, list(d1$seibetsu, d1$class), mean)
> rownames(m3) <- c("A0", "A1", "B0", "B1")
>
> m4 <- t(m3)
> m4
      A0      A1      B0      B1
pre  24.56 30.71429 23.20588 31.09756
post1 28.64 27.77143 27.14706 27.46341
post2 32.56 32.94286 32.52941 32.51220
>
>
```

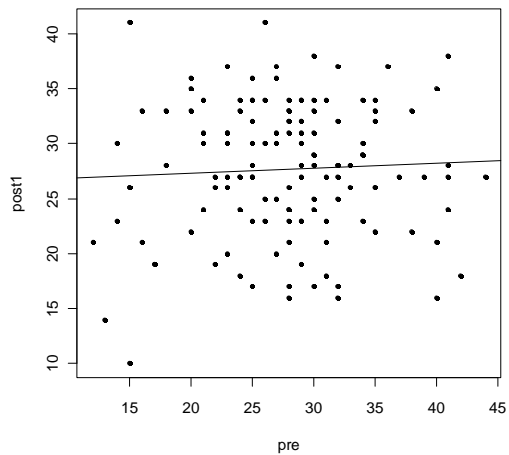
```
> #マーク, 色, タイトル, 軸ラベル, 軸の範囲の設定
> plot(d1$pre, d1$post1, pch=20, col="blue",
+      main="pre-post1 plot", xlab="pre-test", ylab="post-test 1",
+      xlim=c(0, 50), ylim=c(5, 45))
```

```
> #すでにあるグラフに重ねてグラフを書く
> #回帰直線
> plot(d1$pre, d1$post1, pch=20, xlab="pre", ylab="post1")
> abline(lm(post1~pre, data=d1))
>
```

マーク, 色, タイトル, 軸ラベル, 軸の範囲の設定



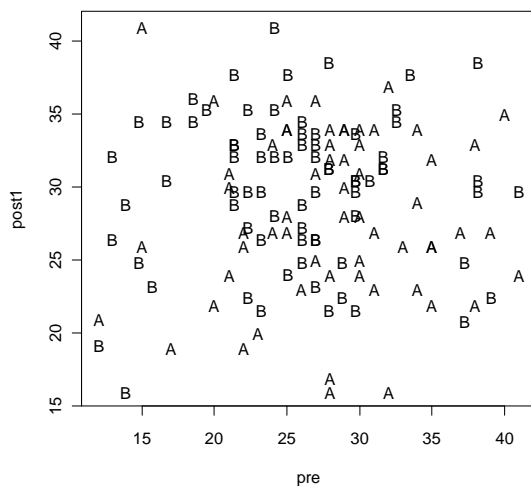
回帰直線の追加



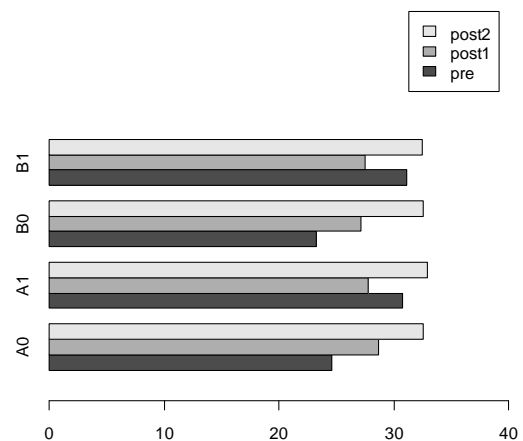
```
> #2群の散布図
> plot(d1a$pre, d1a$post1, pch="A", xlab="pre", ylab="post1")
> par(new=TRUE)
> plot(d1b$pre, d1b$post1, pch="B", axes=FALSE, xlab="", ylab="")
>
```

```
> #棒の幅, 横書き, 軸の範囲, 並列配置, 凡例の設定
> barplot(m4,width=0.5, horiz=TRUE, xlim=c(0, 40), ylim=c(0, 12),
+        beside=TRUE, legend=TRUE)
>
```

2群の散布図の重ね合わせ



棒の幅, 横書き, 軸の範囲, 並列配置, 凡例の設定



> #軸を手動で書く

> # グラフの目盛りラベルを明示的に書く

> boxplot(d1[, 3:5], axes=FALSE, xlab="Time", ylab="Score")

>

> # side = 1:x軸下. 1から3目盛りまでに, c()で指定したラベルを書く

> axis(side=1, 0:4, c("", "Pre Test", "Post Test 1", "Post Test 2", ""))

>

> # side = 2: y 軸左

> axis(side=2)

>

> # side = 3: x軸上

> axis(side=3, 1:3, c("Time 1", "Time 2", "Time 3"))

>

> # side = 4: y軸右

> axis(side=4, seq(10, 45, by=7))

>

