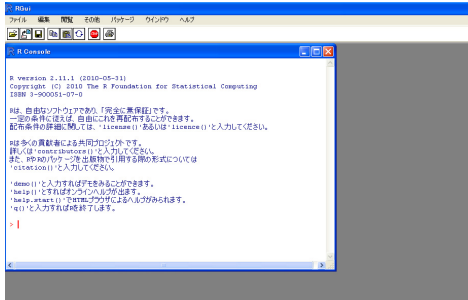


R 入門

櫻井彰人

R 環境

- プロンプト文字: >
- 現在のディレクトリ(working directory): `getwd()`
- ディレクトリの変更: `setwd("c:/R")`
- ヘルプ: `?log` または `help(log)`
 - "fuzzy search" には, `??log`
 - 言語に属するキーワードは、ダブルクォートで括る:
 - `?"for"`, `help("for")`, `??"for"`
- An Introduction to R
 - <http://cran.r-project.org/doc/manuals/R-intro.html>



まずは計算

• 算術計算

```

> -27*12/21
[1] -15.42857

> sqrt(10)
[1] 3.162278

> log(10)
[1] 2.302585

> log10(2+3*pi)
[1] 1.057848

> exp(2.7689)
[1] 15.94109

> (25 - 5)^3
[1] 8000

> cos(pi)
[1] -1

> prod(3:1)
[1] 6

# 10.9.8.7.6.5.4
> prod(10:4)
[1] 604800

> prod(10:4)/prod(40:36)
[1] 0.007659481

> choose(5, 2)
[1] 10

> 1/choose(5, 2)
[1] 0.1

```

Rの値

```

> 2
[1] 2
> 3.1
[1] 3.1
> c(1,2,3,4)
[1] 1 2 3 4
> class(3)
[1] "numeric"
> class(c(1,2,3,4))
[1] "numeric"
> T
[1] TRUE
> TRUE
[1] TRUE
> F
[1] FALSE

> "string"
[1] "string"
> class("string")
[1] "character"
> c(1,"string",3.1)
[1] "1" "string" "3.1"
> class(c(1,"string",3.1))
[1] "character"

```

Rの変数

- 大文字・小文字は区別される

```
a <- 5
A <- 7
B <- a+A
```
- 識別子中に空白はダメ

```
var a <- 5
```
- ピリオドはOK. しかし、使わない方が無難

```
var.a <- 5
var.b <- 10
var.c <- var.a + var.b
```

Rの式

- 変数 <- 式
- 変数 <- 関数名(引数(達))
`r <- lm(y ~ x)` # linear model fitting
- 算術演算子と比較演算子と論理演算子の例

<code>x + y</code>	<code>x == 5</code>
<code>x - y</code>	<code>x != 5</code>
<code>x * y</code>	<code>y < x</code>
<code>x / y</code>	<code>x > y</code>
<code>x ^ y</code>	<code>z <= 7</code>
	<code>p >= 1</code>

`A & B`
`A | B`
`!`

この場合だけが、
この場合には必須

if文

```
if ( 論理式 ) {  
  文s  
} else {  
  文s  
}
```

else はなくてもよい

繰り返し

```
for(i in 1:5) {  
  print(i*i)  
  i <- i+sqrt(i)  
}  
出力  
[1] 1  
[1] 4  
[1] 9  
[1] 16  
[1] 25
```

```
i <- 1  
while(i<=10) {  
  print(i*i)  
  i <- i+sqrt(i)  
}  
出力  
[1] 1  
[1] 4  
[1] 11.65685  
[1] 27.68836  
[1] 57.0912
```

数値列を作る

- 列 `seq(from, to, by)`
- 1 から12まで(1刻み)の配列を値とする変数を作るには:
`> x <- 1:12`
`> x`
`[1] 1 2 3 4 5 6 7 8 9 10 11 12`

seqの使用例

```
> seq(12)  
[1] 1 2 3 4 5 6 7 8 9 10 11 12  
  
> seq(4, 6, 0.25)  
[1] 4.00 4.25 4.50 4.75 5.00 5.25 5.50 5.75 6.00
```

`seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)), length.out = NUL)`

同じ数字パターンを繰り返すには

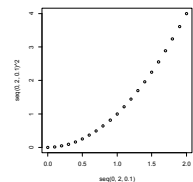
- Repetition - `rep(x, times, ...)`
`> rep(10, 3)`
`[1] 10 10 10`

`> rep(c(1:4), 3)`
`[1] 1 2 3 4 1 2 3 4 1 2 3 4`

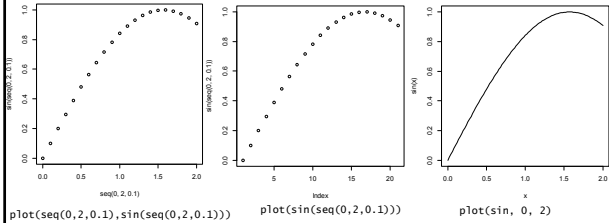
`> rep(c(1.2, 2.7, 4.8), 5)`
`[1] 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8`

グラフ

```
plot(seq(0,2,0.1),seq(0,2,0.1)^2)
```

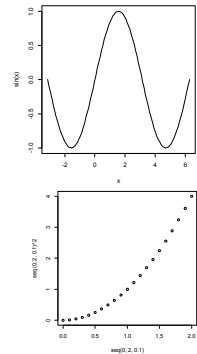


グラフ



グラフ

plot(sin, -pi, 2*pi)
plot(seq(0,2,0.1),seq(0,2,0.1)^2)



いろいろな例

```
> x <- c(1:10)
> x
[1] 1 2 3 4 5 6 7 8 9 10
> x[(x>8) | (x<5)]
[1] 1 2 3 4 9 10
> x > 8
[1] FALSE FALSE FALSE FALSE FALSE FALSE
[8] FALSE TRUE TRUE
> x < 5
[1] TRUE TRUE TRUE TRUE FALSE FALSE
[8] FALSE FALSE
> x > 8 | x < 5
[1] TRUE TRUE TRUE TRUE FALSE FALSE
[8] FALSE TRUE TRUE
> x[c(T,T,T,T,F,F,F,T,T)]
[1] 1 2 3 4 9 10
> subset(x, (x>8) | (x<5))
[1] 1 2 3 4 9 10
```

x <- c(1:10)
x[(x>8) | (x<5)]
x <- c(1:10)
x
x > 8
x < 5
x > 8 | x < 5
x[c(T,T,T,T,F,F,F,T,T)]
subset(x, (x>8) | (x<5))

いろいろな例

```
> m <- matrix(c(1,2,4,1), ncol=2)
> m
     [,1] [,2]
[1,]    1    4
[2,]    2    1
> solve(m)
     [,1] [,2]
[1,] -0.1428571  0.5714286
[2,]  0.2857143 -0.1428571
```

いろいろな例

```
> x <- 1
> y <- 2
> ls()
[1] "x" "y"
> x2 <- 9
> y2 <- 10
> ls()
[1] "x" "x2" "y" "y2"
> ls(pattern="x")
[1] "x" "x2"
> rm(x2,y)
> ls()
[1] "x" "y2"
```

ワークスペース内にあるオブジェクトをリストする

リストするときに、ばらばらに指定することが可能

x2とyをワークスペースから削除

Rのワークスペース

- Rを使用している時に生成されたオブジェクトは記憶されている。こうしたオブジェクトの集合を、Rでは、ワークスペースと呼んでいる。
- Rのワークスペースは、自分で保存しない限り、失われる。
- Rを終了するときに、ワークスペースを保存するか否かを聞かれる。保存を選ぶとその時のワーキングディレクトリに .RData というファイル名で保存される。
- または、コマンドで随時保存できる。

R のワークスペース

- 保存するには、

```
## 現在のワーキングディレクトリに保存。ファイル名は .RData になる。
save.image()
## 現在のワーキングディレクトリを知るgetwd()
## フルパスとファイル名を指定する
save.image("C:\\Exercises\\R\\R-2.14.0\\Test.RData")
## または
save.image("C:/Exercises/R/R-2.14.0/Test.RData")
```

- 回復するには

```
## 現在のワーキングディレクトリからであれば、
load(".RData")
## 任意のディレクトリからは
load("C:\\Exercises\\R\\R-2.14.0\\Test.RData")
## または
load("C:/Exercises/R/R-2.14.0/Test.RData")
```

ワーキングディレクトリ

- 現在のワーキングディレクトリを知るには

```
getwd()
```

- 設定するには

```
mydirectory <- "c:/docs/mydir"
setwd(mydirectory)
## または
setwd("c:/docs/mydir")
```

確率の計算

二項分布

$$P(k | n, p) = C_k^n p^k (1-p)^{n-k}$$

```
dbinom(k, n, p)
```

```
> dbinom(2, 3, 0.60)
[1] 0.432
```

例:

```
plot(dbinom(0:30, 30, 0.5))
plot(dbinom(0:30, 30, 0.5, log))
```

ポアソン分布

$$P(X = k | \lambda) = \frac{e^{-\lambda} \lambda^k}{k!}$$

```
dpois(k, \lambda)
```

```
P(X = 2 | \lambda = 1) = \frac{e^{-1} 1^2}{2!} = 0.1839
> dpois(2, 1)
[1] 0.1839397
```

確率の計算

正規分布

$$P(a \leq X \leq b) = \int_a^b f(x) dx$$

$$\text{pnorm}(a, \text{mean}, \text{sd}) = \int_{-\infty}^a f(x) dx = P(X \leq a | \text{mean}, \text{sd})$$

高さが、平均 156 標準偏差 4.6 の正規分布に従うとき、高さが150以下である確率は

```
> pnorm(150, 156, 4.6)
[1] 0.0960575
```

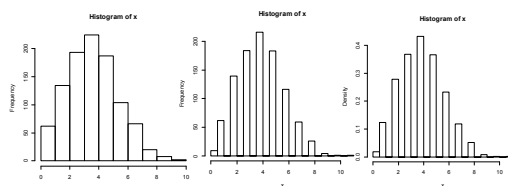
他の例:

```
plot(function(x) pnorm(x, 0, 1), xlim=c(-2.5, 2.5))
plot(function(x) pnorm(x, 0, 1), -2.5, 2.5)
```

簡単なシミュレーション: 二項分布

- ある母集団で 20% が病気であったとしよう。調査時にはこの母集団から20人ランダムに選ぶとする。調査ごとに、何人かは病気であろう。この人数を x とする。1000回調査したときの、この x の分布はどうなるか？

```
x <- rbinom(1000, 20, 0.20)
hist(x)
hist(x, breaks=30)
hist(x, breaks=30, freq=F)
```

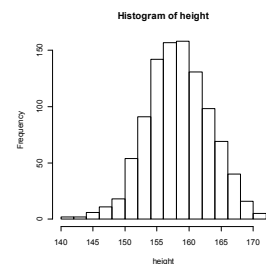


正規分布の場合

- 日本人の20~24歳女性では、平均身長は158.45cm、標準偏差 5.02cm である。この分布から1000回サンプルしたときのヒストグラムをつくろう

```
height <- rnorm(1000, mean=158.45, sd=5.02)
hist(height, breaks=18)
```

breaks には、数字や配列の他、"Sturges" (nclass.Sturges), "Scott" (nclass.scott) や "FD"/"Freedman-Diaconis" (nclass.FD) が標準で指定できる



年齢別体格測定の結果 http://www.mext.go.jp/b_menu/houdou/20/10/08092414/002.xls

サンプリング

- 40人の被験者がいる(1,2,3,...,40). ランダムに5人を選びたい。どうしたらよいか？

```
> sample(1:40, 5)
[1] 30 29 32 17 6
> sample(1:40, 5)
[1] 39 11 32 9 33
> sample(1:40, 5)
[1] 37 10 21 35 11
> sample(1:40, 5)
[1] 4 32 28 39 15
>
```

重複を許したサンプリング

- 重複を許したサンプリング: 10人の被験者を50人のなかから選びたい。しかし、重複してよい。どうしたらよいか。

```
> sample(1:50, 10, replace=T)
[1] 49 29 7 33 19 19 18 44 7 45
```

Dataframe

Excelのシート	data.frame
列	変数 (variable)
行	観測 (observations)

あ	か
0.117	0.155
0.196	0.127
0.106	0.115
0.16	0.068
0.143	0.105
0.222	0.083
0.259	0.102
0.136	0.173
0.107	0.144
0.139	0.083

Data frame: songs
変数: a, ka
観測数: 10

c()を用いた直接書込み

```
> a <- c(0.117,0.196, 0.106, 0.16, 0.143, 0.222, 0.259, 0.136, 0.107, 0.139)
> ka <- c(0.155, 0.127, 0.115, 0.068, 0.105, 0.083, 0.102, 0.173, 0.144, 0.083)
> songs <- data.frame(a, ka)
> attach(songs)
```

The following object(s) are masked by_ '.GlobalEnv':

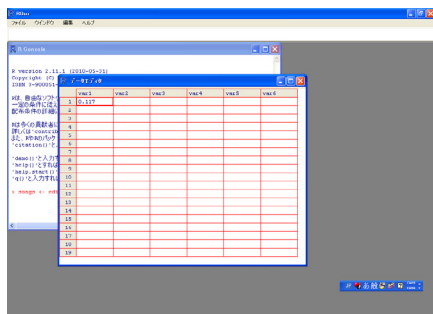
```
a, ka
> songs
  a ka
1 0.117 0.155
2 0.196 0.127
3 0.106 0.115
4 0.160 0.068
5 0.143 0.105
6 0.222 0.083
7 0.259 0.102
8 0.136 0.173
9 0.107 0.144
10 0.139 0.083
>
```

これ以降、songs\$aで、aの列ベクトルが参照できる。なお、aとkaの代わりにあとかを用いることもできる。試してみてください

あ	か
0.117	0.155
0.196	0.127
0.106	0.115
0.16	0.068
0.143	0.105
0.222	0.083
0.259	0.102
0.136	0.173
0.107	0.144
0.139	0.083

edit(data.frame())を用いたデータ入力

```
songs <- edit(data.frame())
```



ファイルからの読み込み: read.table()

```
あ か
0.117 0.155
0.196 0.127
0.106 0.115
0.16 0.068
0.143 0.105
0.222 0.083
0.259 0.102
0.136 0.173
0.107 0.144
0.139 0.083
```

```
setwd("d:/R/Sample")
songs <- read.table("Sample01.txt", header=TRUE)
```

読み込んだ結果

```
> setwd("d:/R/Sample")
> songs <- read.table("Sample01.txt", header=TRUE)
> songs
   あ   か   さ   た   な   は   ま   や   ら   わ   を   ん
1  0.117 0.155 0.039 0.136 0.107 0.058 0.068 0.117 0.039 0.019 0.010 0.029
2  0.196 0.127 0.049 0.157 0.088 0.029 0.069 0.069 0.118 0.000 0.000 0.010
3  0.106 0.115 0.159 0.062 0.115 0.053 0.035 0.097 0.106 0.018 0.000 0.009
4  0.160 0.068 0.123 0.091 0.082 0.014 0.110 0.041 0.091 0.005 0.014 0.050
5  0.143 0.105 0.095 0.062 0.095 0.029 0.086 0.019 0.057 0.000 0.000 0.081
6  0.222 0.083 0.056 0.083 0.139 0.069 0.111 0.042 0.056 0.000 0.014 0.000
7  0.259 0.102 0.102 0.157 0.046 0.037 0.065 0.019 0.028 0.009 0.009 0.000
8  0.136 0.173 0.062 0.062 0.136 0.049 0.049 0.062 0.099 0.062 0.012 0.025
9  0.107 0.144 0.048 0.102 0.112 0.086 0.091 0.037 0.102 0.021 0.011 0.021
10 0.139 0.083 0.046 0.093 0.167 0.037 0.148 0.083 0.111 0.000 0.009 0.000
   あ   が   さ   だ   ば   ば   あ   い   u   e   o
1  0.019 0.019 0.010 0.029 0.029 0.000 0.340 0.117 0.117 0.146 0.282
2  0.020 0.020 0.000 0.010 0.039 0.000 0.275 0.167 0.157 0.088 0.314
3  0.009 0.062 0.000 0.018 0.035 0.000 0.416 0.177 0.124 0.097 0.186
4  0.041 0.009 0.027 0.032 0.041 0.000 0.301 0.247 0.137 0.110 0.205
5  0.062 0.057 0.010 0.057 0.010 0.033 0.305 0.176 0.176 0.090 0.252
6  0.014 0.028 0.014 0.042 0.028 0.000 0.222 0.278 0.208 0.083 0.208
7  0.074 0.065 0.000 0.000 0.028 0.000 0.213 0.204 0.241 0.083 0.259
8  0.012 0.025 0.012 0.025 0.000 0.000 0.481 0.148 0.160 0.049 0.160
9  0.000 0.043 0.005 0.032 0.037 0.000 0.321 0.187 0.134 0.123 0.235
10 0.009 0.009 0.046 0.019 0.000 0.000 0.259 0.241 0.185 0.093 0.222
>
```

Excel ファイル(csv)からの読み込み: read.csv()

- 次のようにすればよい(拡張子は無関係):
setwd("d:/R/Sample")
songs <- read.csv("Sample01.csv", header=TRUE)

```
あかさたなはまやらわをんあがさだばばあいueo
0.117 0.155 0.039 0.136 0.107 0.058 0.068 0.117 0.039 0.019 0.010 0.029 0.010 0.029 0.029 0.340 0.117 0.117 0.146 0.282
0.196 0.127 0.049 0.157 0.088 0.029 0.069 0.069 0.118 0.000 0.000 0.010 0.029 0.029 0.029 0.275 0.167 0.157 0.088 0.314
0.106 0.115 0.159 0.062 0.115 0.053 0.035 0.097 0.106 0.018 0.000 0.009 0.062 0.018 0.035 0.041 0.016 0.177 0.124 0.097 0.186
0.160 0.068 0.123 0.091 0.082 0.014 0.110 0.041 0.091 0.005 0.014 0.050 0.027 0.035 0.041 0.010 0.301 0.247 0.137 0.110 0.205
0.143 0.105 0.095 0.062 0.095 0.029 0.086 0.019 0.057 0.000 0.000 0.081 0.062 0.057 0.010 0.057 0.010 0.033 0.305 0.176 0.176 0.090 0.252
0.222 0.083 0.056 0.083 0.139 0.069 0.111 0.042 0.056 0.000 0.014 0.028 0.014 0.042 0.028 0.028 0.278 0.208 0.083 0.208
0.259 0.102 0.102 0.157 0.046 0.037 0.065 0.019 0.028 0.009 0.009 0.074 0.065 0.010 0.028 0.010 0.213 0.204 0.241 0.083 0.259
0.136 0.173 0.062 0.062 0.136 0.049 0.049 0.062 0.099 0.062 0.012 0.025 0.012 0.025 0.012 0.025 0.012 0.025 0.012 0.025 0.012 0.025
0.107 0.144 0.048 0.102 0.112 0.086 0.091 0.037 0.102 0.021 0.011 0.021 0.012 0.021 0.012 0.021 0.012 0.021 0.012 0.021 0.012 0.021
0.139 0.083 0.046 0.093 0.167 0.037 0.148 0.083 0.111 0.000 0.009 0.000 0.009 0.009 0.009 0.009 0.009 0.009 0.009 0.009 0.009 0.009
```

結果

```
> setwd("d:/R/Sample")
> songs <- read.csv("Sample01.csv", header=TRUE)
> songs
   あ   か   さ   た   な   は   ま   や   ら   わ   を   ん
1  0.117 0.155 0.039 0.136 0.107 0.058 0.068 0.117 0.039 0.019 0.010 0.029
2  0.196 0.127 0.049 0.157 0.088 0.029 0.069 0.069 0.118 0.000 0.000 0.010
3  0.106 0.115 0.159 0.062 0.115 0.053 0.035 0.097 0.106 0.018 0.000 0.009
4  0.160 0.068 0.123 0.091 0.082 0.014 0.110 0.041 0.091 0.005 0.014 0.050
5  0.143 0.105 0.095 0.062 0.095 0.029 0.086 0.019 0.057 0.000 0.000 0.081
6  0.222 0.083 0.056 0.083 0.139 0.069 0.111 0.042 0.056 0.000 0.014 0.000
7  0.259 0.102 0.102 0.157 0.046 0.037 0.065 0.019 0.028 0.009 0.009 0.000
8  0.136 0.173 0.062 0.062 0.136 0.049 0.049 0.062 0.099 0.062 0.012 0.025
9  0.107 0.144 0.048 0.102 0.112 0.086 0.091 0.037 0.102 0.021 0.011 0.021
10 0.139 0.083 0.046 0.093 0.167 0.037 0.148 0.083 0.111 0.000 0.009 0.000
   あ   が   さ   だ   ば   ば   あ   い   u   e   o
1  0.019 0.019 0.010 0.029 0.029 0.000 0.340 0.117 0.117 0.146 0.282
2  0.020 0.020 0.000 0.010 0.039 0.000 0.275 0.167 0.157 0.088 0.314
3  0.009 0.062 0.000 0.018 0.035 0.000 0.416 0.177 0.124 0.097 0.186
4  0.041 0.009 0.027 0.032 0.041 0.000 0.301 0.247 0.137 0.110 0.205
5  0.062 0.057 0.010 0.057 0.010 0.033 0.305 0.176 0.176 0.090 0.252
6  0.014 0.028 0.014 0.042 0.028 0.000 0.222 0.278 0.208 0.083 0.208
7  0.074 0.065 0.000 0.000 0.028 0.000 0.213 0.204 0.241 0.083 0.259
8  0.012 0.025 0.012 0.025 0.000 0.000 0.481 0.148 0.160 0.049 0.160
9  0.000 0.043 0.005 0.032 0.037 0.000 0.321 0.187 0.134 0.123 0.235
10 0.009 0.009 0.046 0.019 0.000 0.000 0.259 0.241 0.185 0.093 0.222
>
```

データの一部を取り出す

```
setwd("d:/R/Sample")
BP <- read.table("BloodPressure.txt", header=TRUE)
attach(BP) ← これ以降、BP$Alcohol を単に Alcohol でアクセスできることになる。R のオブジェクト探索パスの2番目にBPを置くことを意味する。
head(BP, 5)

AlcoholNone <- subset(BP, Alcohol=="None")
AlcoholLittle <- subset(BP, Alcohol=="Little")
AlcoholMedium <- subset(BP, Alcohol=="Medium")
PHigh <- subset(BP, Pressure>=91)
PHighAN <- subset(BP, Pressure>=91 & Alcohol=="None")
```

BloodPressure というデータ

```
Date Day Temp Alcohol Pressure
0 Tue 18 None 107
1 Wed 20 Little 78
2 Thu 20 Little 92
3 Fri 20 Little 87
5 Sun 20 Little 86
6 Mon 20 Medium 90
...
170 Thu 28 Little 93
171 Fri 28 Little 96
174 Mon 29 Little 88
175 Tue 29 Little 90
176 Wed 29 Little 90
177 Thu 29 Little 96
178 Fri 28 Medium 92
```

データのmerge

- 2個のデータをマージ(merge)することができる。マージは、databaseのjoinであり、下記のようにデータを作ることを指示する。

```
> d1 <- BP[c(1,2)] > head(d,5)
> d2 <- BP[c(1,3)] Date Day Temp
> d <- merge(d1,d2,by="Date") 1 0 Tue 18
> head(d1,5) 2 1 Wed 20
Date Day 3 2 Thu 20
1 0 Tue 4 3 Fri 20
2 1 Wed 5 5 Sun 20
3 2 Thu >
4 3 Fri
5 5 Sun
> head(d2,5)
Date Temp
1 0 18
2 1 20
3 2 20
4 3 20
5 5 20
```

データからデータを作る

```

bfp <- c(16.6, 16.4, 17.7, 17.3, 17.5, 15.2, 17.5, 18.1, 17.5, 17.7, 17.1, 16.4, 17.3, 17.3, 16.2,
17.5, 18.2, 17.3, 15.9, 16.3, 17.2, 16.1, 16.1, 18.3, 17.9, 17.6, 16.8, 16.3, 16.9, 15.2, 17
16.9, 16.8, 15.2, 17.9, 18.3, 18.3, 16.8, 16.6, 16.6, 17.8, 18.1, 17.5, 15.2, 17.7, 18.1, 1
6.8, 16.3, 17.1, 16.1, 17.1, 15.9, 16.4, 16.9, 16.9, 16.4, 17.1, 16.9, 16.4, 17.4, 17.4, 17
7, 17.1, 16.9, 17.1, 18.4, 17.7, 17.4, 17.8, 17.1, 15.7, 15.7, 17.2, 17.9, 17.2, 14.6, 16.8,
17.3, 16.5, 16.7, 17.2, 19.3, 16.2, 15.9, 17.7, 16.9, 17.6, 17.2, 16.8, 17.1, 18.1, 17.1, 16
2, 17.1, 17.9, 16.3, 17.1, 17.7, 17.2, 16.8, 16.9, 17.3, 15.5, 16.6, 18.1, 18.9, 17.7, 4.4,
16.4, 15.3, 16.4, 17.4, 16.7, 15.8, 16.9, 14.8, 15.1, 15.2, 16.4, 16.4, 16.6, 16.2, 13
9, 13.9, 14.5, 15.7, 16.7, 17.7, 16.8, 16.4, 14.5)

bfpRange <- rep(0, length(bfp))
bfpRange[bfp <= 15.5] <- -1
bfpRange[bfp > 15.5 & bfp <= 18.5] <- 2
bfpRange[bfp > 18.5] <- 3
data <- data.frame(bfp, bfpRange)

head(data, 10)
  bfp bfpRange
1 16.6      0
2 16.4      0
3 17.7      0
4 17.3      0
5 17.5      0
6 16.2     -1
7 17.5      0
8 18.0      0
9 17.5      0
10 17.0      0
  
```

```

bfpRange <- rep(-10, length(bfp))
bfpRange <- replace(bfpRange, bfp <= 15.5, 1)
bfpRange <- replace(bfpRange, bfp > 15.5 & bfp <= 18.5, 2)
bfpRange <- replace(bfpRange, bfp > 18.5, 3)
  
```

データの度数分布

- 階級を作り、階級ごとにデータの出現度数を数えればよい。

```

> library(Hmisc)
> grouped <- cut2(bfp, g=4)
> table(grouped)
grouped
[13.9,16.5) [16.5,17.0) [17.0,17.5) [17.5,19.3]
      42          26          33          32
  
```

次に cut2 がここにある
4区間に分割する
頻度を数える

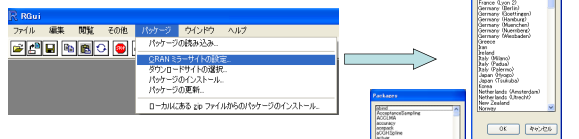
因みに grouped というデータは
[17.5, 19.3) の区間に32個

```

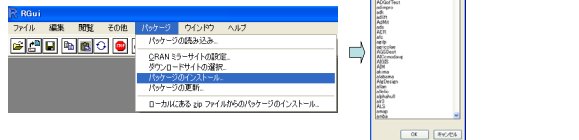
> head(grouped, 10)
[1] [16.5,17.0) [13.9,16.5) [17.5,19.3] [17.0,17.5) [17.5,19.3] [13.9,16.5)
[7] [17.5,19.3] [17.5,19.3] [17.5,19.3] [17.0,17.5)
Levels: [13.9,16.5) [16.5,17.0) [17.0,17.5) [17.5,19.3]
  
```

パッケージのインストール

- まず CRAN ミラーサイトの設定



- 次に パッケージのインストール



まとめ

- R は、プログラミング言語である。
- インタラクティブにも使える
- 統計計算用に、いろいろな道具が用意されている
 - 具体的な内容は、順次
- 機械学習の道具もいろいろあり
 - これも、講義の進展に従い、順次
- Rを使って、機械学習のアルゴリズムを試してみよう！