Rで統計解析入門

(2) データの読み込み

本日のメニュー

1. パッケージについて
2. 作業ディレクトリについて
3. 種々のデータの読み込み
4. データの抽出

パッケージとは

- R は関数とデータを機能別に分類して「パッケージ」という形で用意
- どのようなパッケージがあるのかは関数 library() を実行すると表示

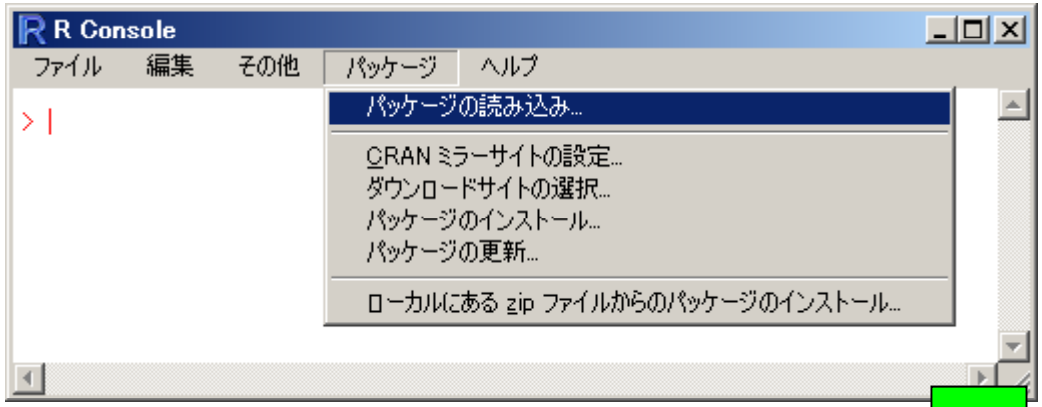
パッケージ名	解説
boot	ブートストラップに関するパッケージ
foreign	R 以外のデータファイルを読み込むためのパッケージ
lattice	ラティス・グラフィックス関数パッケージ
nlme	線形＆非線形混合効果モデル用のパッケージ
nnet	ニューラル・ネットワーク用のパッケージ
rpart	CART に関するパッケージ
splines	スプライン回帰用のパッケージ
survival	生存時間解析用のパッケージ

パッケージの呼び出し

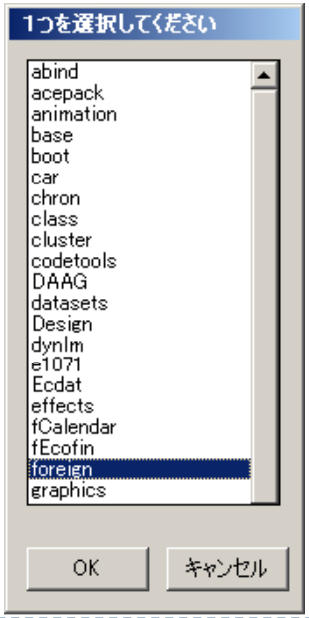
- コマンドでパッケージ「foreign」を呼び出す場合：

```
> library(foreign)          # パッケージ foreign を呼び出す
> library(help="foreign")   # パッケージ foreign のヘルプ
```

- メニューからパッケージ「foreign」を呼び出す場合：



① メニュー「パッケージ」から「パッケージの読み込み」を選択
② 読み込むパッケージ名を選択して [OK] を選択

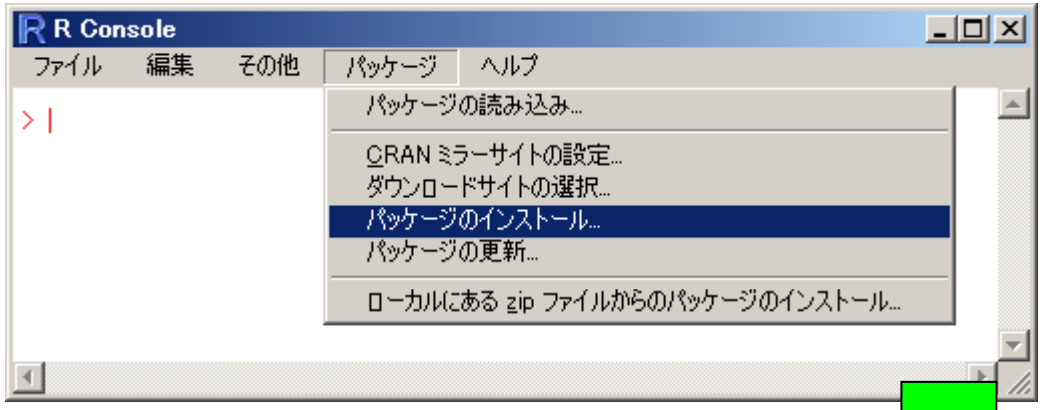


追加パッケージのインストール

- コマンドでパッケージ「xlsx」をインストールする：

```
> install.packages("xlsx")  # パッケージのインストール
```

- メニューからパッケージ「xlsx」をインストールする：



① メニュー「パッケージ」から
「パッケージのインストール」を選択
② 「Japan(Tsukuba)」⇒ [OK] をクリック
③ インストールするパッケージを選択して
[OK] をクリック

本日のメニュー

1. パッケージについて
2. 作業ディレクトリについて
3. 種々のデータの読み込み
4. データの抽出

作業ディレクトリの変更と確認

- Windows 版 R の場合は，「ファイル」→「ディレクトリの変更...」を選択した後，フォルダ「work」を選択してください


```
> setwd("c:/work")      # 作業ディレクトリを変更
> getwd()               # 現在のディレクトリを確認
[1] "c:/work"
```

本日のメニュー

1. パッケージについて
2. 作業ディレクトリについて
3. 種々のデータの読み込み
4. データの抽出

データの型

- R には「データの型」という概念があり，「数値」「文字」「日付」「因子（カテゴリ）」などを区別する ⇒ 日付の処理例は次頁

```
> height <- c(158,162,177,173,166)    # 数値型
> group  <- c("A","A","B","C","C")    # 文字型
> group  <- as.factor(group)          # 関数 as.factor で
>                                     # 因子型(カテゴリ)に変換
> groupc <- as.character(group)       # 文字型に変換
> date   <- as.Date("111111", format="%y%m%d") # 日付型に変換
```

- 外部ファイルを R に読み込むと「数値」は「数値型」，「文字」は「因子型（カテゴリ）」に自動変換される

  ⇒ 「文字」を「文字型」としたい場合は要変換！

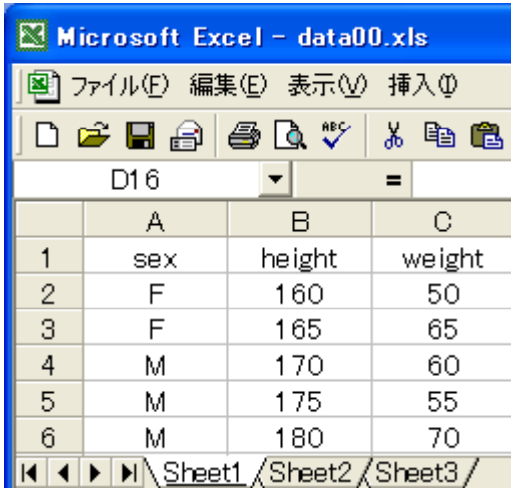
日付データのハンドリング例

```
> as.Date("2012/01/26", format="%Y/%m/%d") # 文字列を日付に変換
[1] "2012-01-26"
> x <- as.Date("2012/01/26", format="%Y/%m/%d") -
+       as.Date("111111", format="%y%m%d") # 日数の差を計算
> as.numeric(x)                           # 結果を数値に変換
[1] 76
```

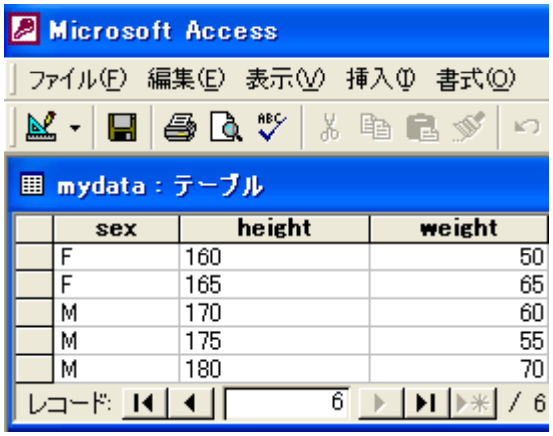
命令	機能
%A，%a	曜日の英語名（小文字は略記）
%B，%b	月の英語名（小文字は略記）
%d	日（01-31）
%m	月（01-12）
%Y，%y	西暦（大文字：4 桁表示，小文字：2 桁表示）

データフレームとは

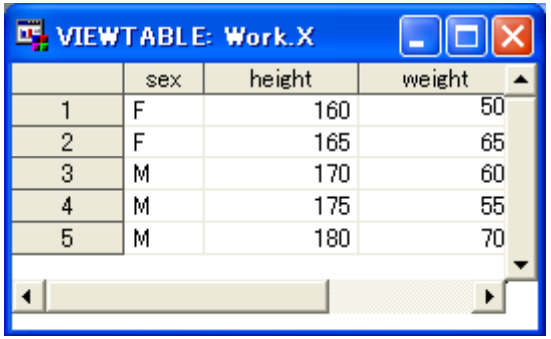
- 統計解析を行うデータの形式は様々
  - （R 上で）データを手で入力して・・・
  - テキストファイル，EXCEL，ACCESS，SAS などの形式
- R でデータ解析を行う際は，データフレームという形式にデータを変換することが多い（見た目は行列）



EXCEL：シート



ACCESS：テーブル



SAS：データセット

データフレームとは

- 数値や文字，因子（カテゴリ）や
  日付などの異なる型のデータを
  まとめてもつ変数
  ⇒ 外見は行列と同じ
  ⇒ 各列の値の型はバラバラでも良い
- データフレームの各行・各列はラベル
  を必ず持ち，ラベルによる操作が可能
- データは C:/ にあることを仮定する
  ⇒ 前もって setwd("C:/") を実行

データフレーム「demo」

ID	AGE	GENDER	DATE
2	50	1	2001/01/01
4	55	2	2002/02/02
6	60	2	2003/03/03
3	65	1	2004/04/04
1	70	2	2005/05/05
5	75	1	2006/06/06

※GENDER：1 が男性，2 が女性

データフレームの作成方法

1. R でベクトルデータを作成した後，データフレームを作成（手入力）
   - 「ID」「年齢」「性別」「検査日」などのデータをベクトルで用意した後，関数 **data.frame()** で１つのデータフレームに変換

2. ファイルからデータを読み込んで，データフレームを作成
   - テキストファイルから読み込み
     - 関数 **read.table()** や関数 **read.csv()**
   - EXCEL ファイルから読み込み
     - CSV ファイルに変換した後，関数 **read.csv()**
     - パッケージ xlsx の関数 **read.xlsx()**
     - EXCEL のセルを直接コピー

   cf. パッケージ RODBC の関数 **odbcConnectXXXXX()** でファイルにアクセスした後，関数 **sql.Query()** でデータを読み込む

データの型

- 手入力でデータフレームを作成する場合は，以下の書式に従う

```
> data.frame(変数名1 = ベクトル1，変数名2 = ベクトル2，…)
```

- 以下に例を挙げる

```
> x <- data.frame(ID    =c( 2, 4, 6, 3, 1, 5),
+                 AGE   =c(50,55,60,65,70,75),
+                 GENDER=c( 1, 2, 2, 1, 2, 1),
+                 DATE  =c("2001/01/01","2002/02/02",
+                          "2003/03/03","2004/04/04",
+                          "2005/05/05","2006/06/06") )
```

データフレームの作成方法

1. R でベクトルデータを作成した後，データフレームを作成（手入力）
   - 「ID」「年齢」「性別」「検査日」などのデータをベクトルで用意した後，関数 **data.frame()** で１つのデータフレームに変換

2. ファイルからデータを読み込んで，データフレームを作成
   - テキストファイルから読み込み
     - 関数 **read.table()** や関数 **read.csv()**
   - EXCEL ファイルから読み込み
     - CSV ファイルに変換した後，関数 **read.csv()**
     - パッケージ xlsx の関数 **read.xlsx()**
     - EXCEL のセルを直接コピー

   cf. パッケージ RODBC の関数 **odbcConnectXXXXX()** でファイルにアクセスした後，関数 **sql.Query()** でデータを読み込む

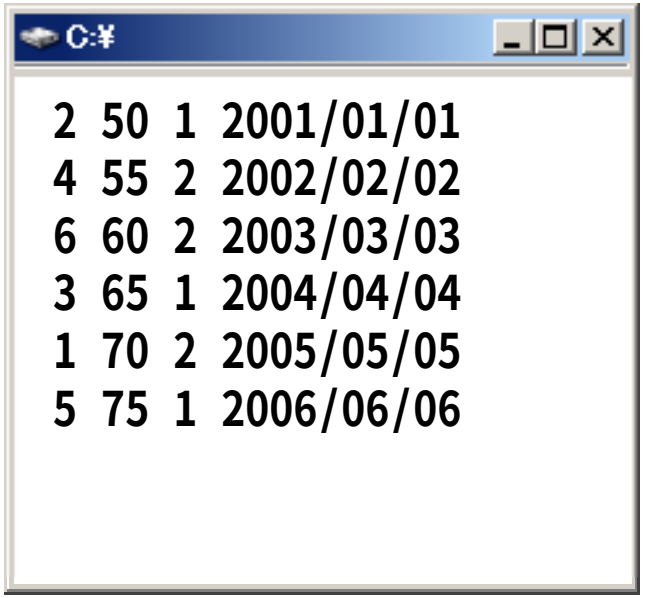
データフレームの作成（⇔ .txt ）

1. 列名がなく，データ間がスペースで区切られている場合

   ⇒ R が勝手に列名を決める

```
> x <- read.table("demo1.txt")
> x

  V1 V2 V3         V4
1  2 50  1 2001/01/01
2  4 55  2 2002/02/02
3  6 60  2 2003/03/03
4  3 65  1 2004/04/04
5  1 70  2 2005/05/05
6  5 75  1 2006/06/06
```



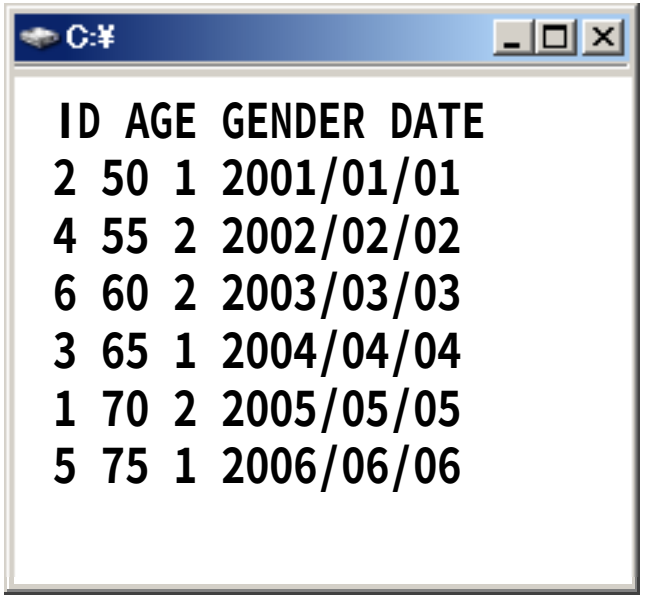
demo1.txt

データフレームの作成（⇔ .txt ）

2. 列名があり，データ間がスペースで区切られている場合

```
> x <- read.table("demo2.txt",
+                 header=T)
> x

  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
2  4  55      2 2002/02/02
3  6  60      2 2003/03/03
4  3  65      1 2004/04/04
5  1  70      2 2005/05/05
6  5  75      1 2006/06/06
```



demo2.txt

データフレームの作成（⇔ .txt ）

3. 1 行目にコメント，2 行目に列名があり，データ間がスペースで
   区切られている場合 ⇒ 1 行目を読み飛ばす

```
> x <- read.table("demo3.txt",
+                 header=T, skip=1)
> x

  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
2  4  55      2 2002/02/02
3  6  60      2 2003/03/03
4  3  65      1 2004/04/04
5  1  70      2 2005/05/05
6  5  75      1 2006/06/06
```

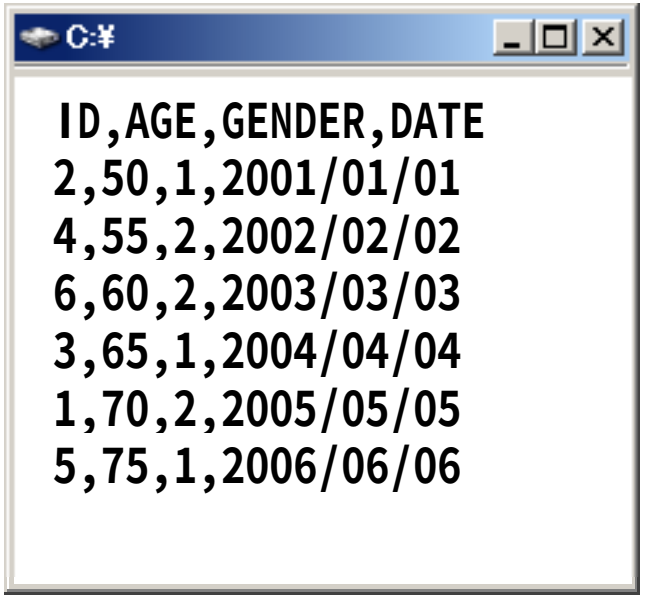
demo3.txt

データフレームの作成（⇔ .txt ）

4. 列名があり，データ間がコンマで区切られている場合

```
> x <- read.table("demo4.txt",
+                 header=T, sep=",")
> x

  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
2  4  55      2 2002/02/02
3  6  60      2 2003/03/03
4  3  65      1 2004/04/04
5  1  70      2 2005/05/05
6  5  75      1 2006/06/06
```



demo4.txt

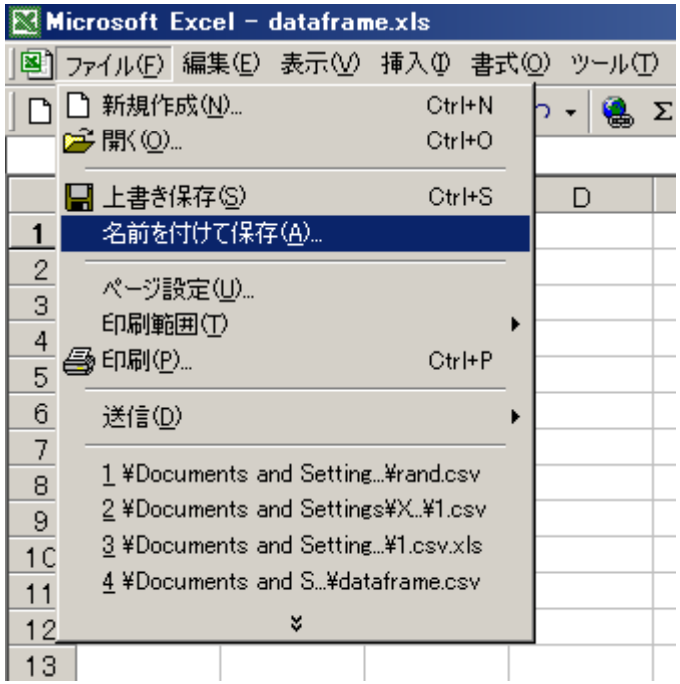
データフレームの作成（⇔ .xls/xlsx ）

- EXCEL ファイル（.xls/.xlsx）を R に読み込ませる場合：
  - .csv ファイルに変換して読み込ませる
  - EXCEL ファイルをそのまま読み込ませる ← パッケージ「**xlsx**」

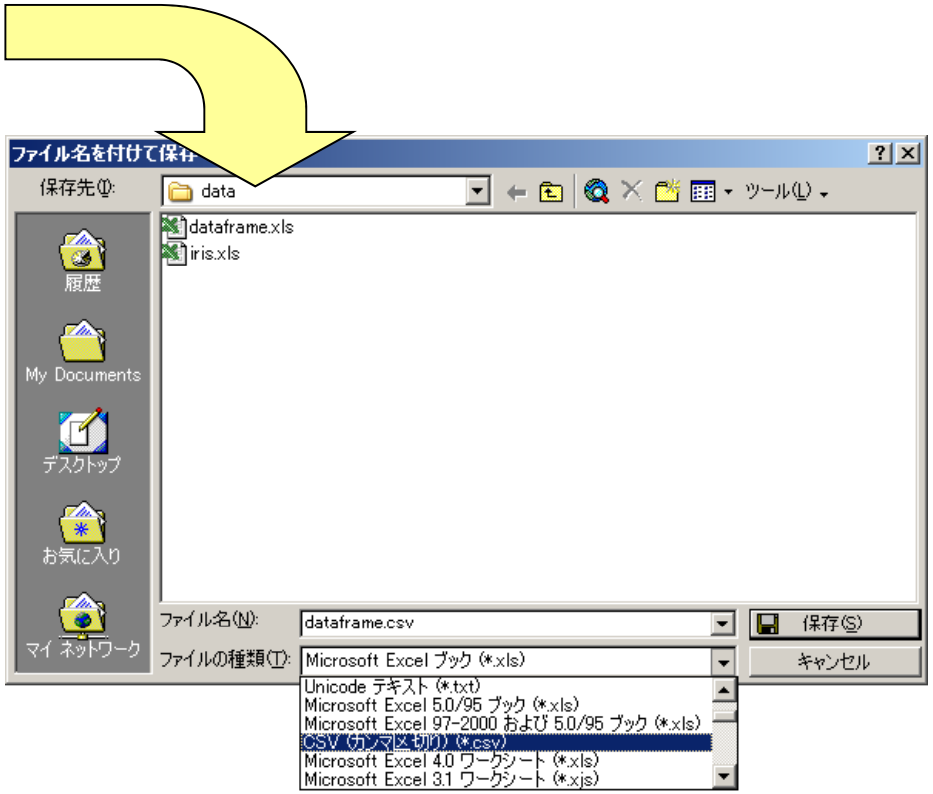
- .csv ファイルに変換して読み込ませる場合，前もって関数 read.csv() で読み込める形式にすること（前節の demo4.txt の状態）
  1. まず，EXCEL ファイルを開き，メニューの [ファイル] の [開く] から，[名前をつけて保存] を選択する
  2. 保存する名前をつけた後，次に [ファイルの種類] から [CSV カンマ区切り] を選択して保存する

データフレームの作成（⇔ .xls/xlsx ）

- EXCEL を別名で保存（.csv ファイルとして保存）



別名で保存



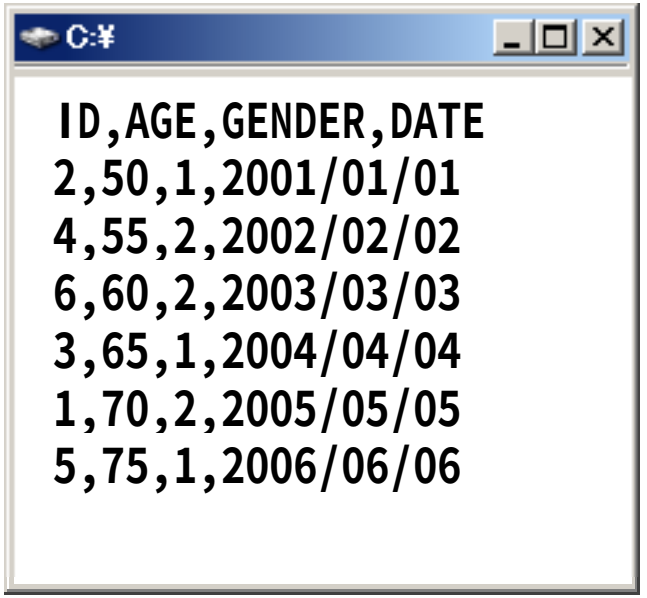
CSV（カンマ区切り）で保存

データフレームの作成（⇔ .txt ）

4'. 列名があり，データ間がコンマで区切られている場合

```
> x <- read.csv("demo4.csv")
> x

  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
2  4  55      2 2002/02/02
3  6  60      2 2003/03/03
4  3  65      1 2004/04/04
5  1  70      2 2005/05/05
6  5  75      1 2006/06/06
```



demo4.csv

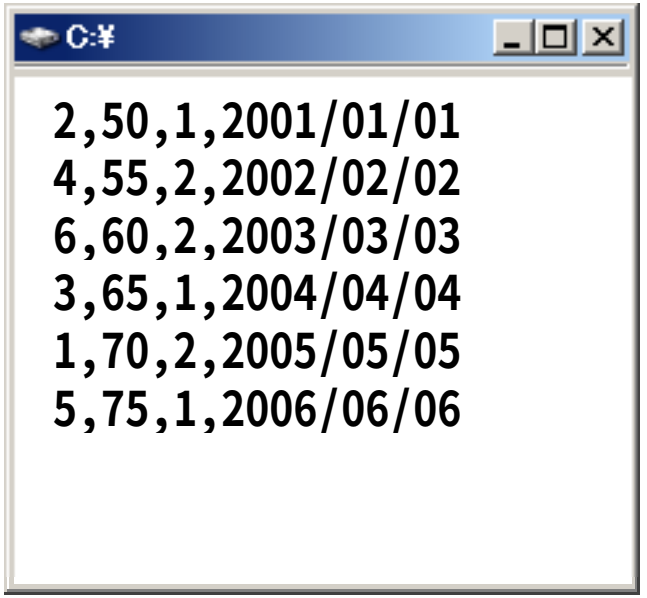
データフレームの作成（⇔ .txt ）

5. 列名がなく，データ間がコンマで区切られている場合

⇒ 先に列名を表す変数を作成した後，関数 read.csv() で読み込み

```
> myname <- c("ID","AGE","GENDER","DATE")
> x <- read.csv("demo5.csv",
+           header=F, col.names=myname)
> x

  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
2  4  55      2 2002/02/02
3  6  60      2 2003/03/03
4  3  65      1 2004/04/04
5  1  70      2 2005/05/05
6  5  75      1 2006/06/06
```

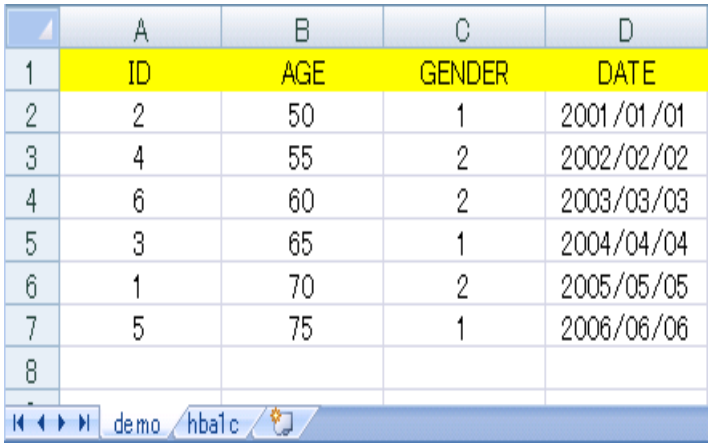


demo5.csv

データフレームの作成（⇔ .xlsx ）

- パッケージ xlsx の関数 read.xlsx() で EXCEL ファイルを読み込む

```
> install.packages("xlsx", dep=T)
> library(xlsx)
> ( x <- read.xlsx("c:/mydata.xlsx",
+                  sheetName="demo") )
ID AGE GENDER        DATE
1  2   50      1 2001/01/01
2  4   55      2 2002/02/02
3  6   60      2 2003/03/03
4  3   65      1 2004/04/04
5  1   70      2 2005/05/05
6  5   75      1 2006/06/06
```



	A	B	C	D
1	ID	AGE	GENDER	DATE
2	2	50	1	2001/01/01
3	4	55	2	2002/02/02
4	6	60	2	2003/03/03
5	3	65	1	2004/04/04
6	1	70	2	2005/05/05
7	5	75	1	2006/06/06
8				

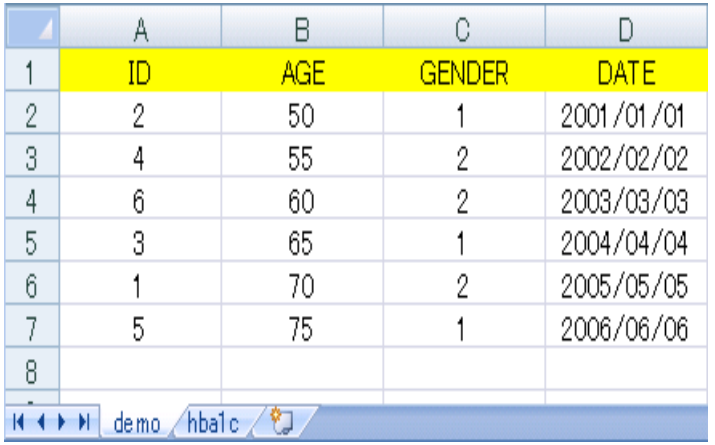
demo hbalc

c:/mydata.xlsx

データフレームの作成（⇔ .xlsx ）

- パッケージ XLConnect の関数 loadWorkbook() 等で EXCEL ファイルを読み込む ⇒ PC に EXCEL がインストールされてなくても使用可！

```
> install.packages("XLConnect", dep=T)
> library(XLConnect)
> tmp <- loadWorkbook("c:/mydata.xlsx")
> getSheets(tmp)                    # シート一覧
[1] "demo"  "hba1c"
> # シート名を指定して読み込み
> x <- readWorksheet(tmp, sheet="demo")
> x
  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
:  :   :      :          :
```



	A	B	C	D
1	ID	AGE	GENDER	DATE
2	2	50	1	2001/01/01
3	4	55	2	2002/02/02
4	6	60	2	2003/03/03
5	3	65	1	2004/04/04
6	1	70	2	2005/05/05
7	5	75	1	2006/06/06
8				

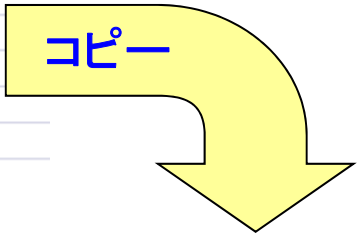
demo hba1c

c:/mydata.xlsx

EXCEL のセルをコピーして作成

- EXCEL のセルをコピーして，そのまま R に貼り付けることも出来る

	A	B	C	D
1	ID	AGE	GENDER	DATE
2	2	50	1	2001/01/01
3	4	55	2	2002/02/02
4	6	60	2	2003/03/03
5	3	65	1	2004/04/04
6	1	70	2	2005/05/05
7	5	75	1	2006/06/06
8				




```
# 列名をコピーした場合
x <- read.delim(pipe("pbpaste"), header=T)

# 列名をコピーしなかった場合
x <- read.delim(pipe("pbpaste"), header=F)
```

Mac OS X の場合

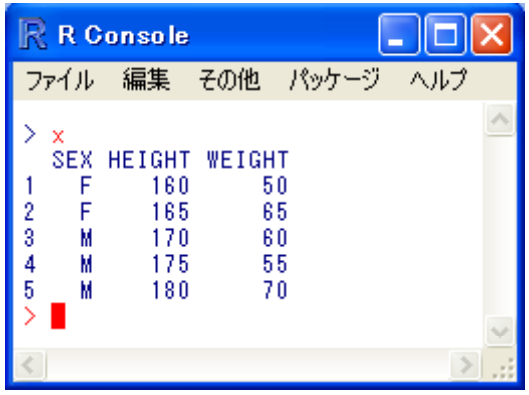
```
# 列名をコピーした場合
x <- read.delim("clipboard", header=T)

# 列名をコピーしなかった場合
x <- read.delim("clipboard", header=F)
```

Windowsの場合

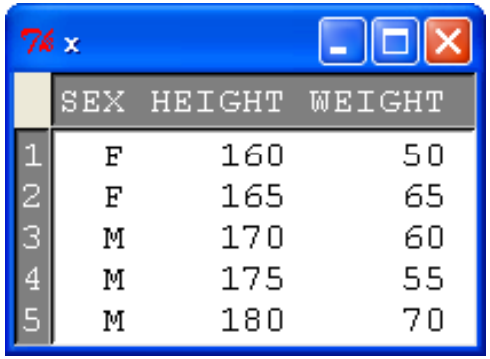
データフレームの閲覧

- データフレームの中身を確認したいときは・・・
  - R のコンソール画面で
  - R 標準のデータエディタで（←データを見ながらの作業不可）
  - relimp パッケージのテキストウインドウで



コンソール上
```
> x
> head(x)
```

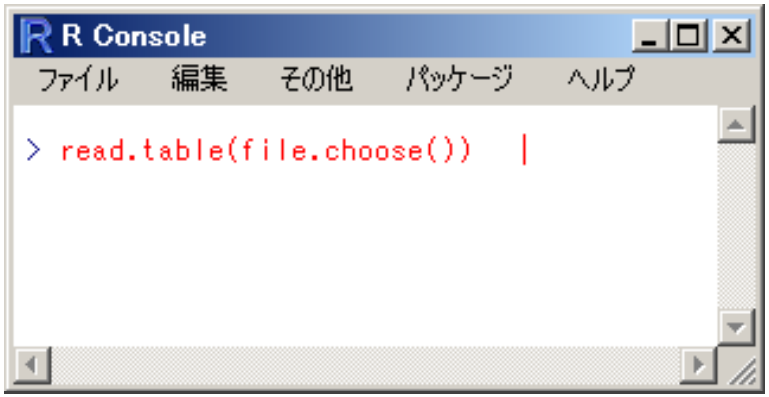
データエディタ
```
> edit(x)
```



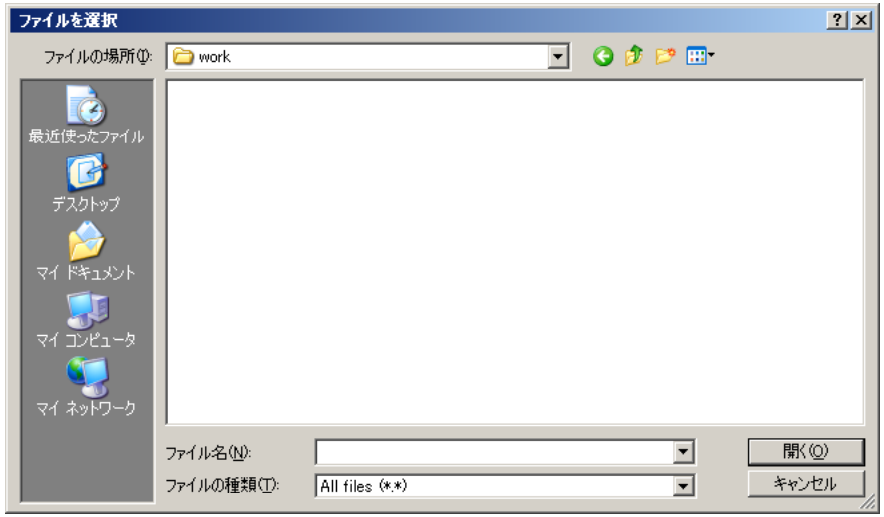
テキストウインドウ
```
> library(relimp)
> showData(x)
```

【参考】データファイルの読み込み

- Windows 版 R では，関数 file.choose() を使用すると，ファイル名を指定するダイアログが表示される



直接ファイル名を指定せずにマウスでファイルを指定することが出来るようになる！



【参考】データフレームの作成（RODBC）

- パッケージ RODBC の中の関数 odbcConnectXXXXX() でファイルにアクセスした後，関数 sql.Query() でデータを読み込むことが出来る

```
> library(RODBC)                                  # パッケージの呼出
> tmp <- odbcConnectExcel("mydata.xls")           # データに接続
> tmp <- odbcConnectAccess("mydata.mdb")          #  （Access の場合）
> sqlTables(tmp)                                  # テーブルを表示
> x <- sqlQuery(tmp,"select * from [demo$]")      # 読み込み
> x <- sqlQuery(tmp, "select * from [demo]")      #  （Access の場合）
> odbcClose(tmp)                                  # 接続を遮断
```

- 他にも ORACLE のデータベースや，その他のデータ形式ファイル（DBASE，MySQL，PostgreSQL）からデータを読み込むことも可

【参考】SAS データの読み込み

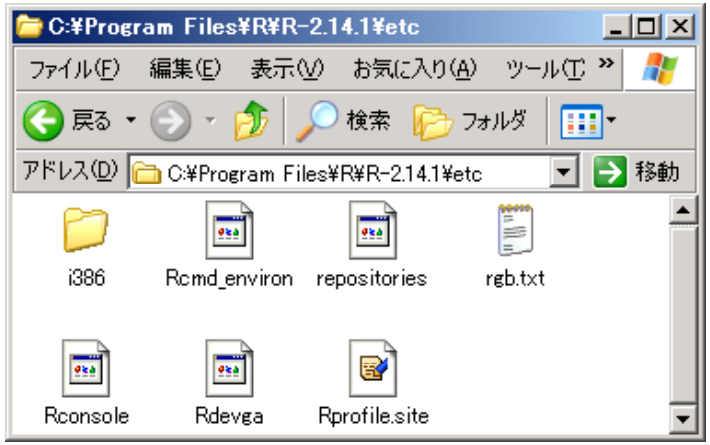
- パッケージ sas7bdat の中の関数 read.sas7bdat () で SAS データを読み込むことが出来る

```
> install.packages("sas7bdat")
> library(sas7bdat)
> mydata <- read.sas7bdat("c:/demo.sas7bdat")
> mydata

  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
2  4  55      2 2002/02/02
3  6  60      2 2003/03/03
4  3  65      1 2004/04/04
5  1  70      2 2005/05/05
6  5  75      1 2006/06/06
```

【おまけ】Windows 版 R の場合

- 「Rprofile.site」（拡張子は「.site」です）
  http://www.cwk.zaq.ne.jp/fkhud708/files/R-intro/R-stat-intro_data.zip
  をダウロードして解凍し，[C:¥Program Files¥R¥R-2.14.1¥etc] にある
  同名ファイルに上書き ⇒ メニュー画面からデータが読み込める！



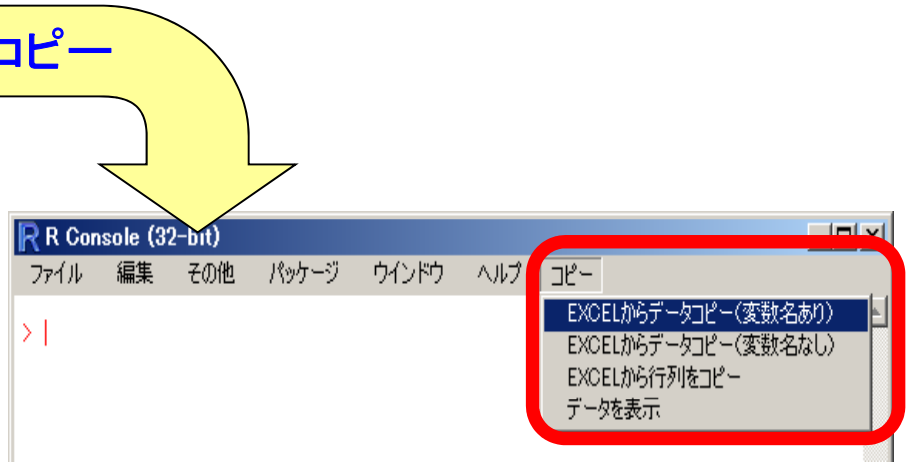
- その後，以下を実行して使用するパッケージをインストールする

```
> install.packages("relimp",    dep=T)
> install.packages("sas7bdat", dep=T)
> install.packages("xlsx",      dep=T)
```

【おまけ】EXCEL のセルをコピーして作成

- EXCEL のセルをコピーして，そのまま R に貼り付けることも出来る

	A	B	C	D
1	ID	AGE	GENDER	DATE
2	2	50	1	2001/01/01
3	4	55	2	2002/02/02
4	6	60	2	2003/03/03
5	3	65	1	2004/04/04
6	1	70	2	2005/05/05
7	5	75	1	2006/06/06
8				



- EXCEL のセルをコピーした後，メニュー「コピー」から
  「EXCEL からデータコピー（...）」を選択
  ⇒ 変数名をコピーした場合は「（変数名あり）」の方を選択
  ⇒ 変数 tmp にデータが読み込まれる

本日のメニュー

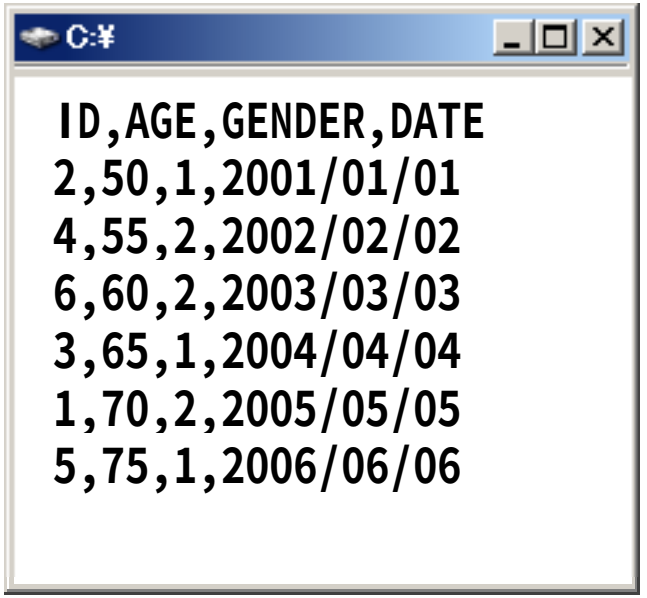
1. パッケージについて
2. 作業ディレクトリについて
3. 種々のデータの読み込み
4. データの抽出

使用するデータ

- C:/demo4.txt を変数 x に格納する（データフレーム x が出来上がる）
  ⇒ CSV ファイルの読み込み方法を参照

```
> x <- read.csv("c:/demo4.txt")
> x

  ID AGE GENDER       DATE
1  2  50      1 2001/01/01
2  4  55      2 2002/02/02
3  6  60      2 2003/03/03
4  3  65      1 2004/04/04
5  1  70      2 2005/05/05
6  5  75      1 2006/06/06
```



demo4.txt

データフレームの中身を見る場合

- データフレーム x の中身を見る場合

  ⇒ 変数名を入力，又は関数 head(データフレーム名, n=XX) を使用する

```
> x                          # データフレーム x の中身を抽出
ID AGE GENDER       DATE
1  2   50      1 2001/01/01
2  4   55      2 2002/02/02
3  6   60      2 2003/03/03
4  3   65      1 2004/04/04
5  1   70      2 2005/05/05
6  5   75      1 2006/06/06
> head(x, n=2)               # データフレーム x の先頭 2 行を抽出
ID AGE GENDER       DATE
1  2   50      1 2001/01/01
2  4   55      2 2002/02/02
```

データフレームの中身を見る場合

- パッケージ relimp の中の関数 **showData(データフレーム名)** でも可

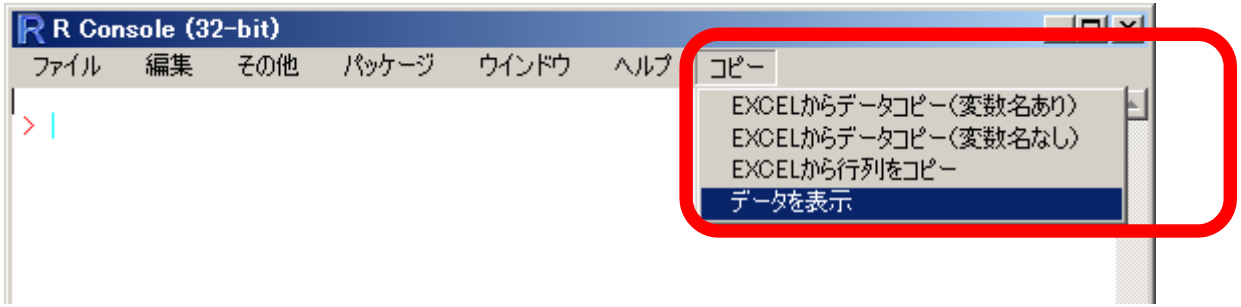
```
> install.packages("relimp", dep=T)     # パッケージのインストール
> library(relimp)                       # パッケージの呼出
> showData(x)                           # データフレーム x の表示
```

x

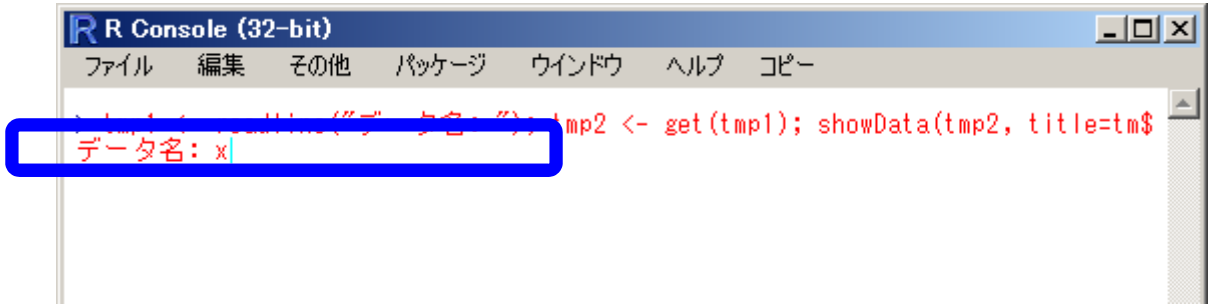
	ID	AGE	GENDER	DATE
1	2	50	1	2001/01/01
2	4	55	2	2002/02/02
3	6	60	2	2003/03/03
4	3	65	1	2004/04/04
5	1	70	2	2005/05/05
6	5	75	1	2006/06/06

データフレームの中身を見る場合

1. メニュー「コピー」→「データを表示」を選択する



2. 「データ名」というメッセージが出るので，表示したいデータ名を入力する ⇒ データが表示される



1 つの変数を取り出す場合

- データフレーム x の変数 AGE を取り出す場合

  ⇒ $ 又は 関数 subset(データフレーム名, select=変数名) を使用する

```
> x$AGE                    # ベクトルとして抽出
[1] 50 55 60 65 70 75
> subset(x, select=AGE)    # データフレームとして抽出
  AGE
1  50
2  55
3  60
4  65
5  70
6  75
```

2 つの変数を取り出す場合

- データフレーム x の変数 ID と変数 AGE を取り出す場合

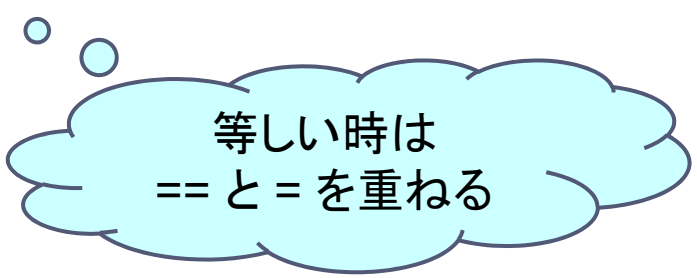
  ⇒ 関数 subset(データフレーム名, select=変数のリスト) を使用する

```
> ( y <- subset(x, select=c(ID,AGE)) )
  ID AGE
1  2  50
2  4  55
3  6  60
4  3  65
5  1  70
6  5  75
> y$AGE                          # さらに年齢のみを抽出
[1] 50 55 60 65 70 75
```

ある条件に合致したレコードを取り出す場合

- データフレーム x について，GENDER が 1 であるレコードの
  変数 ID と変数 AGE を取り出す場合
  ⇒ 関数 subset(データフレーム名, 条件式, 変数のリスト) を使用する

```
> subset(x, GENDER==1, select=c(ID,AGE))
  ID AGE
1  2  50
4  3  65
6  5  75
```



比較のための演算子

記号	==	!=	>=	>	<=	<
意味	等しい	≠	≧	＞	≦	＜

ある条件に合致したレコードを取り出す場合

- データフレーム x について，AGE が 65 歳以上であるレコードの変数 ID と変数 GENDER を取り出す場合
  ⇒ 関数 subset(データフレーム名, 条件式, 変数のリスト) を使用する
  （条件式の表は前頁参照）

```
> ( y <- subset(x, AGE>=65, select=c(ID,GENDER)) )
  ID GENDER
4  3      1
5  1      2
6  5      1
> y$GENDER      #さらに性別のみを抽出
[1] 1 2 1
> subset(x, AGE>=65, select=c(ID,GENDER))$GENDER    # 上と同じ命令
```

ある条件に合致したレコードを取り出す場合

```
> subset(x, AGE>=65, select=GENDER)        # 年齢が65歳以上の性別
  GENDER
4      1
5      2
6      1
> subset(x, AGE==65, select=GENDER)        # 年齢が65歳丁度の性別
  GENDER
4      1
```

比較演算子

記号	==	!=	>=	>	<=	<
意味	等しい	≠	≧	＞	≦	＜

複数の条件に合致したレコードを取り出す場合

```
> subset(x, GENDER==1 & AGE>60)    # GENDERが1かつ年齢が60歳より上
  ID AGE GENDER       DATE
4  3  65      1 2004/04/04
6  5  75      1 2006/06/06

> subset(x, GENDER==1 & AGE>60, select=ID)  # 変数をIDのみ抽出
  ID
4  3
6  5
```

複数の条件を重ねるときの演算子

記号	!	&	|
意味	否定	かつ	または

【参考】データへのアクセス方法

データフレーム x に対する命令	機能
x$列名，x["列名"]，x[["列名"]]	指定した列データを表示
x[2], x[[2]]	2 番目の列データを表示
x[3, 2], x[[3, 2]]	3 行 2 列目のデータを表示
x[[3,"列名"]], x[[3,"列名"]]	指定した列の 3 行目のデータを表示
x[c(1, 2)]	1 列目と 2 列目のデータを表示
x[c(3, 4),]	3 行目と 4 行目のデータを表示
x[,c(T,F,T)]	論理ベクトル c(T,F,T) が TRUE となっている列を表示
x[GENDER==2,]	性別が 2（女性）である行を表示
x[,GENDER==2 & AGE>60]	性別が 2（女性）かつ年齢が 60歳より大きい行を表示
subset(x, gender==2 & age>60)	x[,GENDER==2 & AGE>60] と同様の機能

【参考】データへのアクセス方法

データフレーム x に対する命令	機能
head(x，n=a)	先頭から a 行だけ抽出する
tail(x，n=b)	末尾から b 行だけ抽出する
na.omit(x)	NA を含む行を削除する
transform(x，y=ベクトル)	データフレーム x に新たな列 y を追加する
subset(x，条件式)	条件式に合う行のみを抽出する
subset(x，条件式，ベクトル)	ベクトルで指定した列に対し，条件式に合う行のみを抽出する
reshape(x, ...)	データフレーム x を横展開／縦展開する
apply(x[,範囲], 1, 関数)	データフレーム x の指定した範囲について，各行ごとに関数を適用する （各列ごと： apply(x[,範囲], 2, 関数) とする）

【参考】データへのアクセス方法

データフレーム x に対する命令	機能
ncol(x)	x の列数（変数の数）を求める
nrow(x)	x の行数（データ数）を求める
names(x)	x の列名を表示する
rbind(x,y)	x と y を縦に並べて結合する
cbind(x,y)	x と y を横に並べて結合する
data.frame(x,y)	x と y を横に並べて結合する
merge(x,y)	x と y をくっつける（マージする） ※ all=T を指定するとデータを全て残す all=T を指定しなければデータの共通部分が結果として返される

※ たまに関数 attach() や detach() を使用している資料が見受けられるが、
使わない方が良いです（実際にデータ解析を行う場合ではまず使わないです）

本日のメニュー

1. パッケージについて
2. 作業ディレクトリについて
3. 種々のデータの読み込み
4. データの抽出

Rで統計解析入門

終