R で学ぶデータ解析とシミュレーション

④

～ データハンドリング入門 ～

４時間目のメニュー

- パッケージについて
  - パッケージとは
  - パッケージの呼び出し
  - 追加パッケージのインストール
- データハンドリング入門
  - データフレームとは
  - 種々のテキストファイルを R に読み込ませる方法
  - データハンドリング手法一覧
  - 演習

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」という形にまとめている
- どのようなパッケージがあるのかは関数 library() を実行することで知ることが出来る

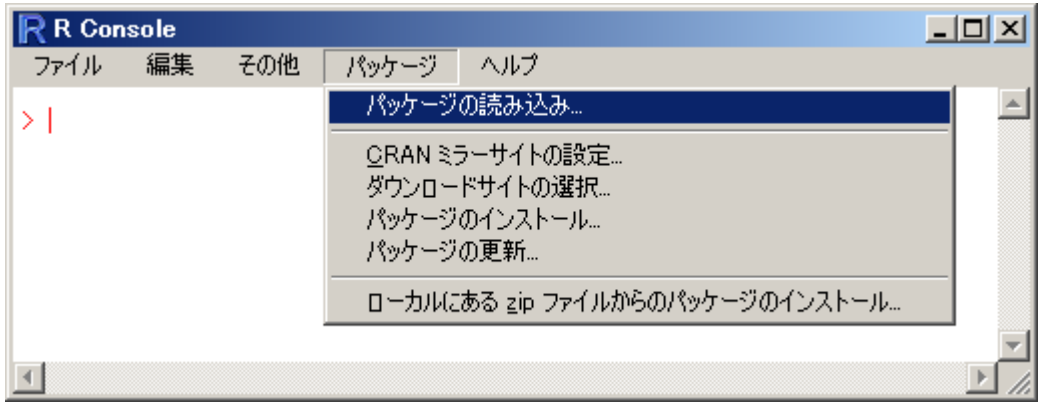
パッケージ名	解説
boot	ブートストラップに関するパッケージ
foreign	R 以外のデータファイルを読み込むためのパッケージ
lattice	ラティス・グラフィックス関数パッケージ
nlme	線形＆非線形混合効果モデル用のパッケージ
nnet	ニューラル・ネットワーク用のパッケージ
rpart	CART に関するパッケージ
splines	スプライン回帰用のパッケージ
survival	生存時間解析用のパッケージ

パッケージの呼び出し

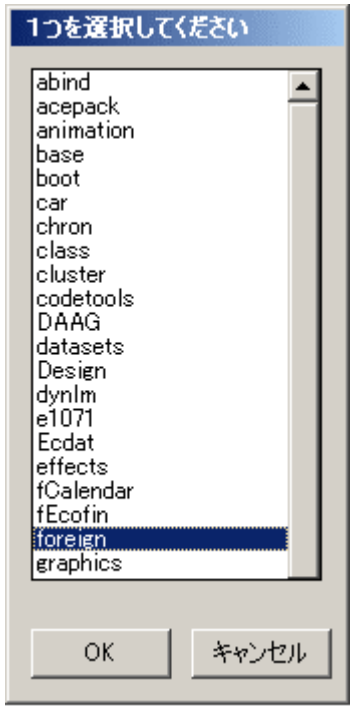
■ コマンドでパッケージ「foreign」を呼び出す場合：

```
> 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

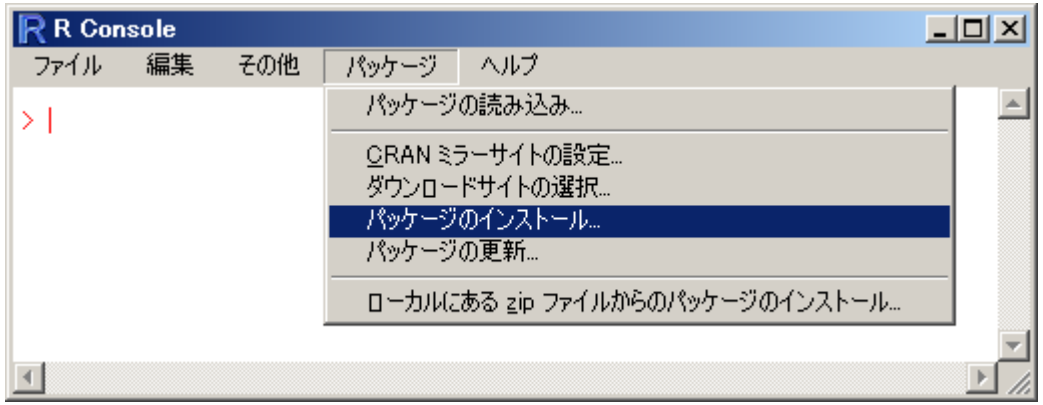


追加パッケージのインストール

- コマンドでパッケージ「xlsReadWrite」をインストールする：

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

- メニューからパッケージ「xlsReadWrite」をインストールする：



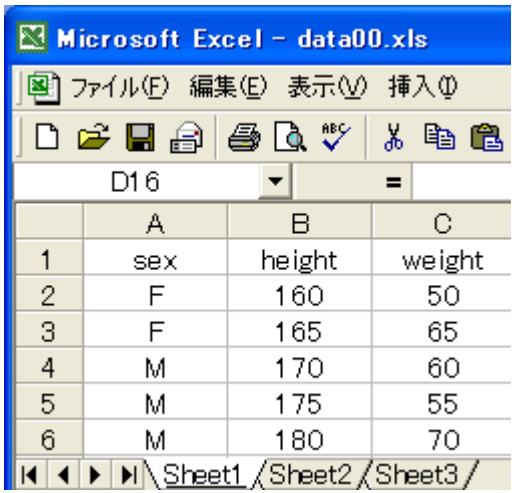
① メニュー「パッケージ」から
「パッケージのインストール」を選択
② 「Japan(Tsukuba)」⇒ [OK] をクリック
③ インストールするパッケージを選択して
[OK] をクリック

４時間目のメニュー

- パッケージについて
  - パッケージとは
  - パッケージの呼び出し
  - 追加パッケージのインストール
- データハンドリング入門 ←
  - データフレームとは
  - 種々のテキストファイルを R に読み込ませる方法
  - データハンドリング手法一覧
  - 演習

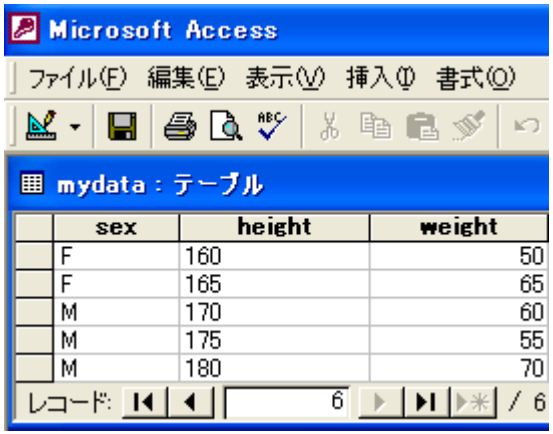
データフレームとは

- 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）



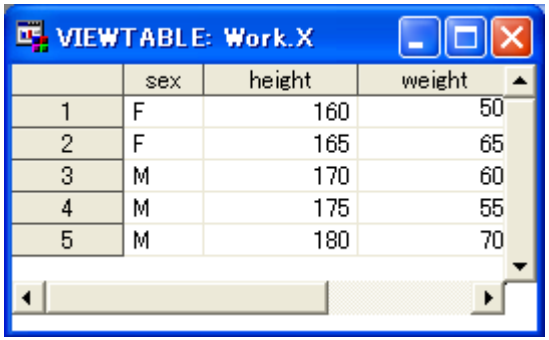
	A	B	C
1	sex	height	weight
2	F	160	50
3	F	165	65
4	M	170	60
5	M	175	55
6	M	180	70

EXCEL：シート



sex	height	weight
F	160	50
F	165	65
M	170	60
M	175	55
M	180	70

ACCESS：テーブル



	sex	height	weight
1	F	160	50
2	F	165	65
3	M	170	60
4	M	175	55
5	M	180	70

SAS：データセット

データフレームとは

- 数値ベクトルや文字ベクトル，因子ベクトル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
  - ⇒ 各列の要素の型はバラバラでも構わない
- 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

SEX	HEIGHT	WEIGHT
F	158	51
F	162	55
M	177	72
M	173	57
M	166	64

データフレームの作成

- 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（いわゆる手入力）
  - ⇒ 「性別」「身長」「体重」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する

- 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
  - ⇒ 関数 **read.table()** などでファイルからデータを読み込む
  - ⇒ パッケージ xlsReadWrite の関数 **read.xls()** で EXCEL ファイルを読み込む
  - ⇒ 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（手入力）

SEX	HEIGHT	WEIGHT
F	158	51
F	162	55
M	177	72
M	173	57
M	166	64

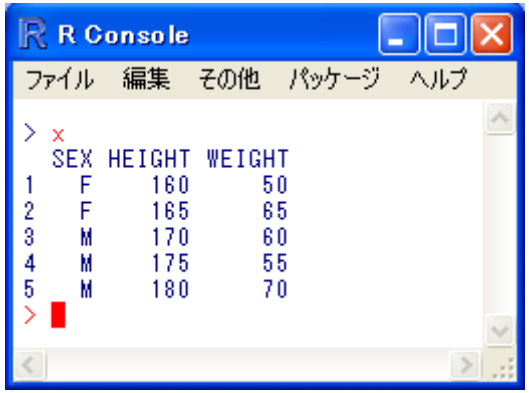
data.frame()

SEX	HEIGHT	WEIGHT
F	158	51
F	162	55
M	177	72
M	173	57
M	166	64

```
> sex    <- c("F","F","M","M","M")
> height <- c(158,162,177,173,166)
> weight <- c( 51, 55, 72, 57, 64)

> x <- data.frame(SEX=sex, HEIGHT=height,
                  WEIGHT=weight)
```

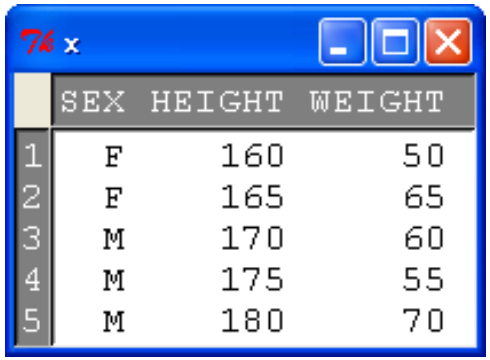
データフレームの閲覧

- 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

データエディタ
> edit(x)



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

データフレームを作成すると・・・

```
> summary(x)     # データフレームの列ごとの特徴を見る

 SEX        HEIGHT          WEIGHT
 F:2   Min.   :158.0   Min.   :51.0
 M:3   1st Qu.:162.0   1st Qu.:55.0
       Median :166.0   Median :57.0
       Mean   :167.2   Mean   :59.8
       3rd Qu.:173.0   3rd Qu.:64.0
       Max.   :177.0   Max.   :72.0

# 密度関数のプロット
> library(lattice)
> densityplot(~ height | voice.part,
              data=singer, layout=c(2,4))

# 対散布図
> splom(~ iris[1:4], groups = Species, data = iris,
+         panel = panel.superpose)
```

SEX	HEIGHT	WEIGHT
F	158	51
F	162	55
M	177	72
M	173	57
M	166	64

データフレームの作成（⇔ .txt ）

- 関数 **read.table()** などでテキストファイルからデータを読み込むことが出来る

```
> x <- read.table("data.txt",
                  header=T, sep="," )
  sex height weight
1   F    158     51
2   F    162     55
3   M    177     72
4   M    173     57
5   M    166     64

> x <- read.csv("data.txt")
```

C:¥

```
sex,height,weight
F,158,51
F,162,55
M,177,72
M,173,57
M,166,64
```

data.txt

データフレームの作成（⇔ .xls ）

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

```
> x <- read.xls("data.xls", sheet=1)
> x

  sex height weight
1   F    158     51
2   F    162     55
3   M    177     72
4   M    173     57
5   M    166     64
```

	A	B	C	D
1	SEX	HEIGHT	WEIGHT	
2	F	158	51	
3	F	162	55	
4	M	177	72	
5	M	173	57	
6	M	166	64	
7				

data.xls

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("c:/data00.xls")          # データに接続
> tmp <- odbcConnectAccess("c:/data00.mdb")         #  (Access の場合)
> sqlTables(tmp)                                    # テーブルを表示
> x <- sqlQuery(tmp,"select * from [Sheet1$]")      # 読み込み
> x <- sqlQuery(tmp, "select * from [mydata]")      #  (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

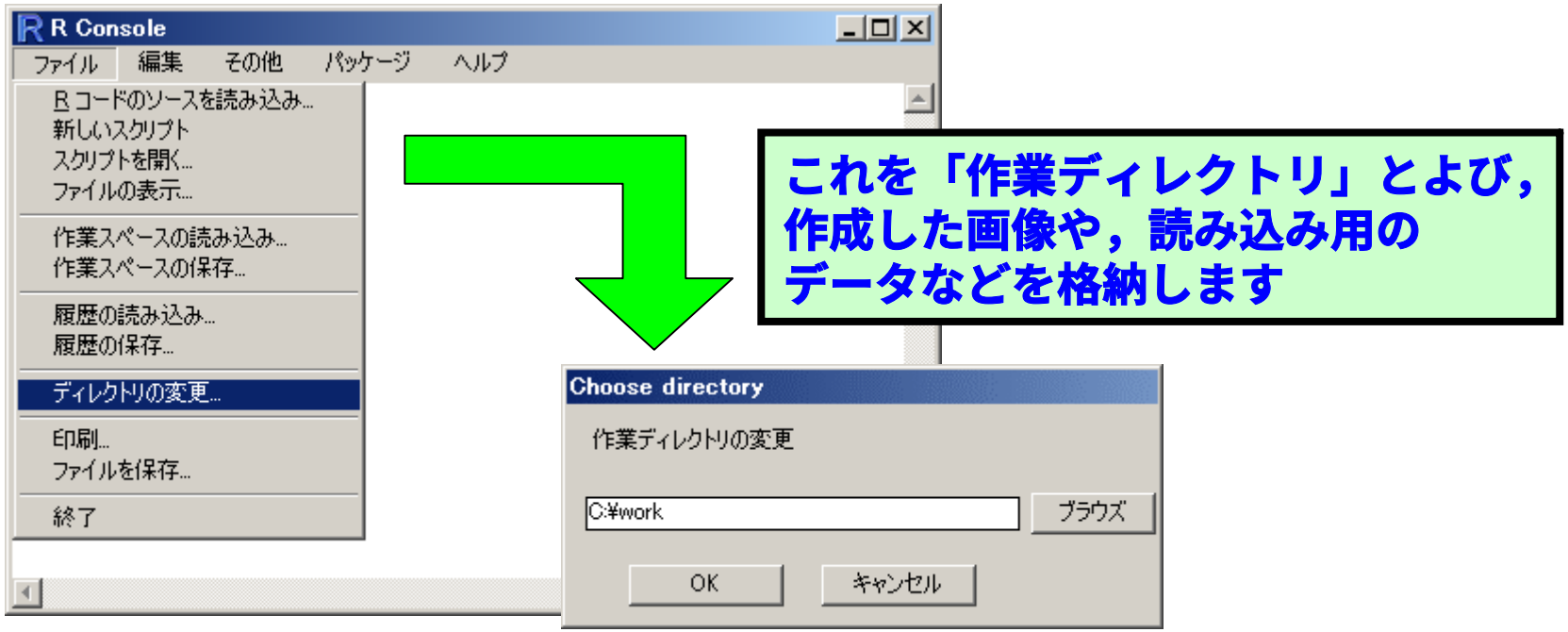
データフレームの作成（foreign）

- パッケージ **foreign** の中には，外部データを読み込むための関数が多数用意されている

関数	用途
data.restore()	Read an S3 Binary File
read.dbf()	Read a DBF File
read.dta()	Read Stata Binary Files
read.epiinfo()	Read Epi Info Data Files
read.mtp()	Read a Minitab Portable Worksheet
read.octave()	Read Octave Text Data Files
read.spss()	Read an SPSS Data File
read.ssd()	Obtain a Data Frame from a SAS Permanent
read.systat()	Obtain a Data Frame from a Systat File
read.xport()	Read a SAS XPORT Format Library

【演習】作業フォルダの作成＆変更【準備】

1. R の「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"
```

【演習】

2. データ「data.txt」をフォルダ「work」に格納してください

3. 2. で格納したデータを変数 x に読み込んでください

4. 関数 head() を用いて，変数 x の 1 行目から 3 行目を表示してください

```
> head(x)        # これだと 5 行目まで表示されてしまう・・・
```

5. パソコンに EXCEL がインストールされている方は，パッケージ xlsReadWrite の関数 read.xls でデータ「data.xls」を読み込んでください

```
> ( x <- read.xls("data.xls", sheet=1) )
```

【参考】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」「因子（カテゴリ）」などを区別する

```
> sex    <- c("F","F","M","M","M")     # 文字型
> 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
```

- 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<u>要変換</u>！

４時間目のメニュー

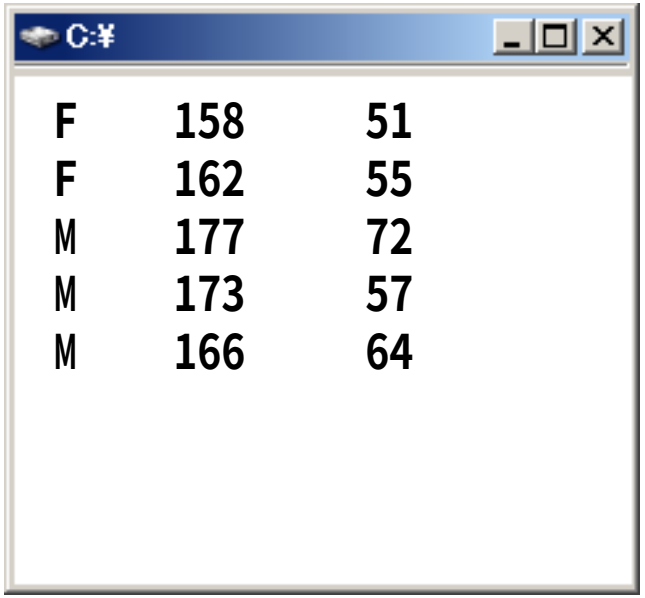
- パッケージについて
  - パッケージとは
  - パッケージの呼び出し
  - 追加パッケージのインストール
- データハンドリング入門
  - データフレームとは
  - 種々のテキストファイルを R に読み込ませる方法 ←
  - データハンドリング手法一覧
  - 演習

★ テキストファイル ⇒ データフレーム

(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ている

```
> x <- read.table("data01.txt")

  V1  V2 V3
1  F 158 51
2  F 162 55
3  M 177 72
4  M 173 57
5  M 166 64
```



data01.txt

★ テキストファイル ⇒ データフレーム

(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("data02.txt",
                  header=T )
  sex height weight
1   F    158     51
2   F    162     55
3   M    177     72
4   M    173     57
5   M    166     64
```

C:¥

```
sex height weight
  F    158     51
  F    162     55
  M    177     72
  M    173     57
  M    166     64
```

data02.txt

★ テキストファイル ⇒ データフレーム

(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

```
> x <- read.table("data03.txt",
                  header=T, skip=1 )
  sex height weight
1   F    158     51
2   F    162     55
3   M    177     72
4   M    173     57
5   M    166     64
```

```
### data03.txt
sex height weight
  F    158     51
  F    162     55
  M    177     72
  M    173     57
  M    166     64
```

data03.txt

★ テキストファイル ⇒ データフレーム

(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("data04.txt",
                  header=T, sep="," )
  sex height weight
1   F    158     51
2   F    162     55
3   M    177     72
4   M    173     57
5   M    166     64
```

C:¥

```
sex,height,weight
F,158,51
F,162,55
M,177,72
M,173,57
M,166,64
```

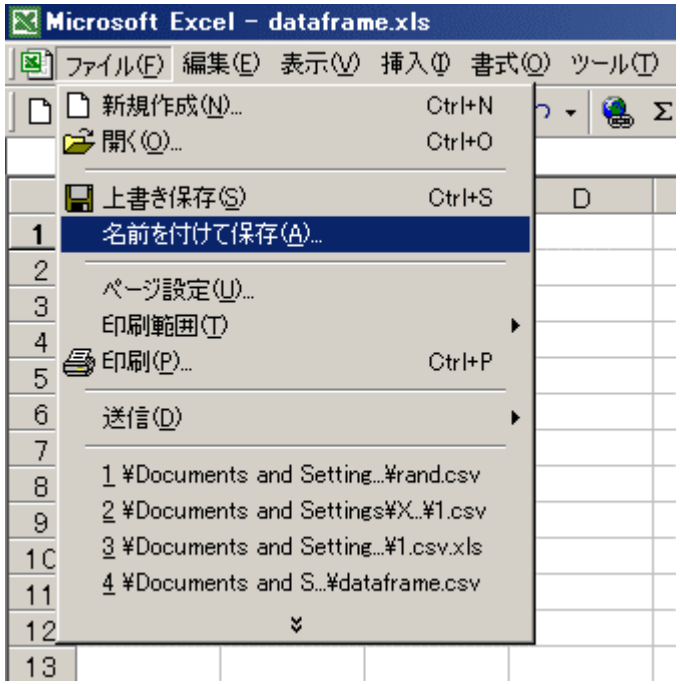
data04.txt

★ テキストファイル ⇔ EXCEL

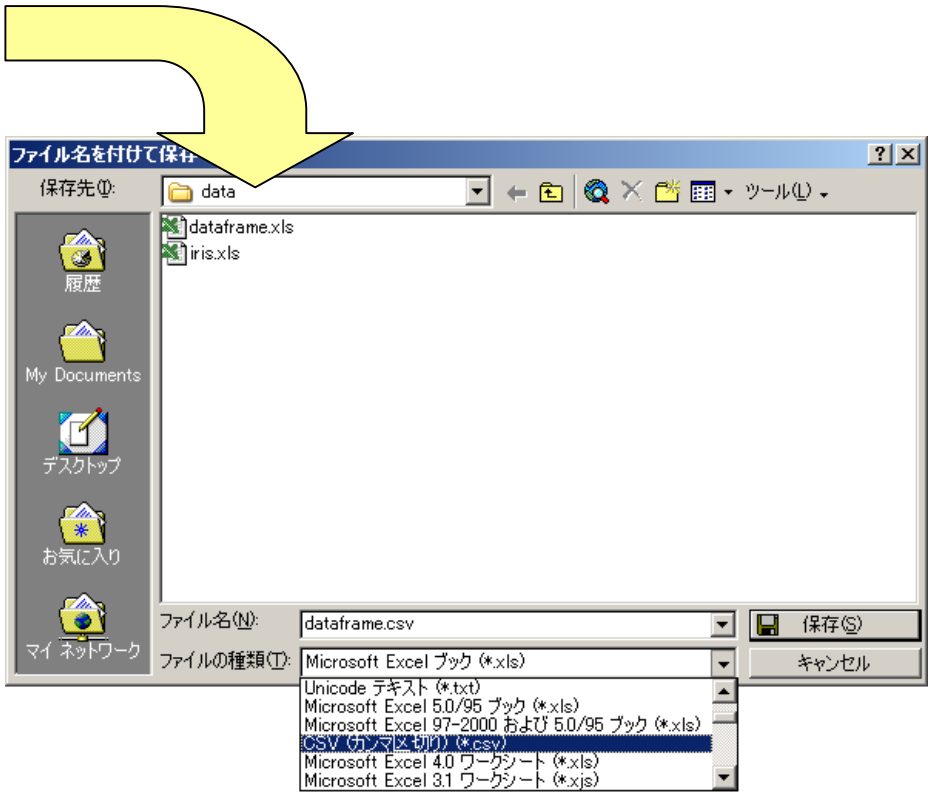
- EXCEL ファイル（.xls）を R に読み込ませる場合：
  - .xls ファイルをそのまま読み込ませる
  - .csv ファイルに変換して読み込ませる ← ここに焦点を当てる
- 目的は関数 read.csv() で読み込める形式にすること（前節の data04.txt の状態）
  - まず，EXCEL ファイルを開き，メニューの [ファイル] の [開く] から，[名前をつけて保存] を選択する
  - 保存する名前をつけた後，次に [ファイルの種類] から [CSV カンマ区切り] を選択して保存する

★ テキストファイル ⇔ EXCEL

■ 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

★ テキストファイル ⇒ データフレーム

(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("data04.csv")

   sex height weight
1    F    158     51
2    F    162     55
3    M    177     72
4    M    173     57
5    M    166     64
```

C:¥

```
sex,height,weight
F,158,51
F,162,55
M,177,72
M,173,57
M,166,64
```

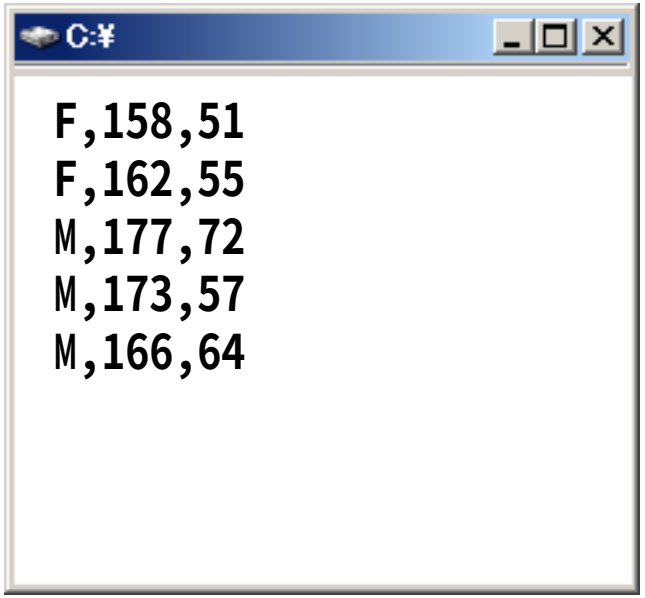
data04.csv

★ テキストファイル ⇒ データフレーム

(5) 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ている

```
> myname <- c("SEX","HEIGHT","WEIGHT")
> x <- read.csv("data05.csv",
          header=F, col.names=myname )
 sex height weight
1   F    158     51
2   F    162     55
3   M    177     72
4   M    173     57
5   M    166     64
```

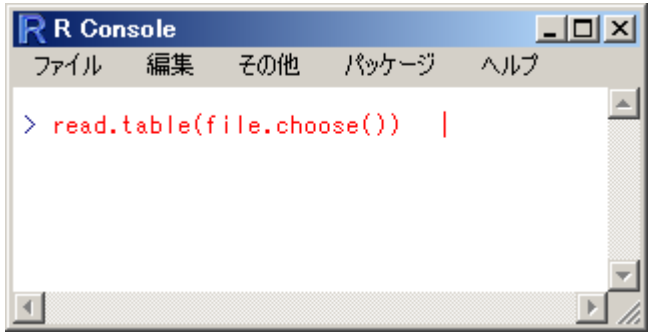


data05.csv

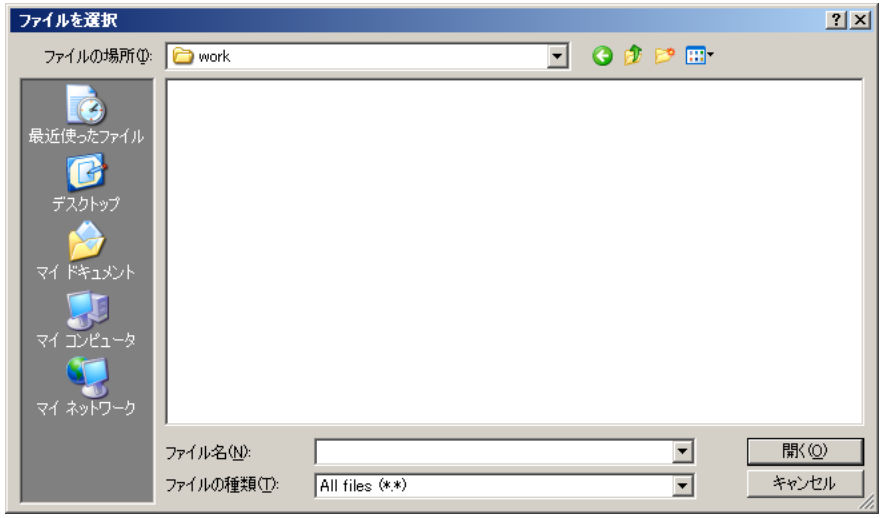
【参考】データファイルの読み込み

- 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

```
> read.table(file.choose())
```

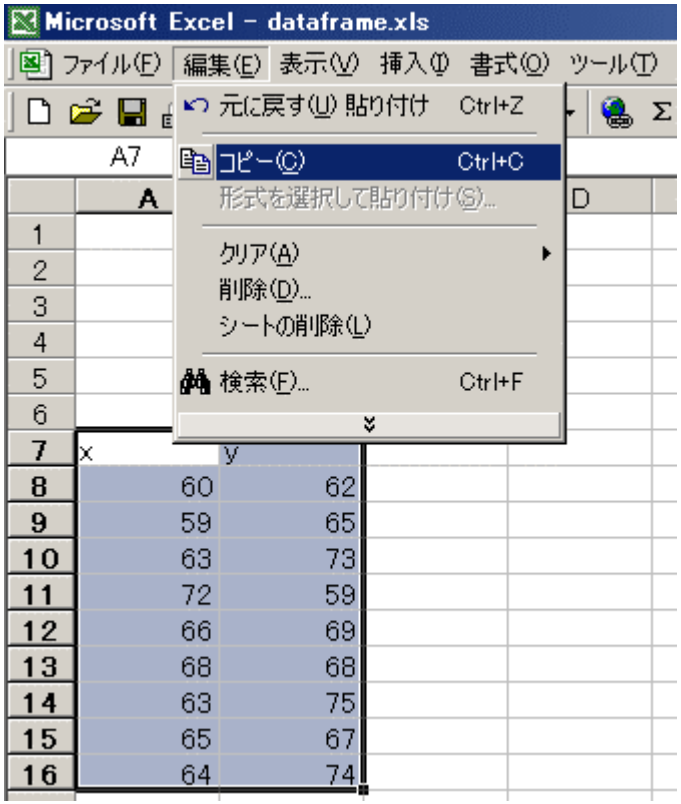


直接ファイル名を指定せずにマウスでファイルを指定することが出来るようになる！



【参考】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！




```
# 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 )
```

４時間目のメニュー

- パッケージについて
  - パッケージとは
  - パッケージの呼び出し
  - 追加パッケージのインストール
- データハンドリング入門
  - データフレームとは
  - 種々のテキストファイルを R に読み込ませる方法
  - データハンドリング手法一覧 ←
  - 演習

データへのアクセス方法

コマンド	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[SEX=="F",]	性別が F（女性）である行を表示
x[,SEX=="F" & WEIGHT>50]	性別が F（女性）かつ体重が 50kg より大きい行を表示

データへのアクセス例

■ データフレーム[行番号，列番号] で指定する

```
> x[c(1,3,5),]          # 1,3,5行目にアクセス
  sex height weight
1   F    158     51
3   M    177     72
5   M    166     64

> x[,c(1,3)]            # 1,3列目にアクセス
  sex weight
1   F     51
2   F     55
3   M     72
4   M     57
5   M     64
```

データフレーム x

SEX	HEIGHT	WEIGHT
F	158	51
F	162	55
M	177	72
M	173	57
M	166	64

データへのアクセス例

■ データフレーム$列名 で指定する

```
> x$height                    # 身長データ
[1] 158 162 177 173 166

> x$height <- NULL            # 身長を削除
> x
  sex weight
1   F     51
2   F     55
3   M     72
4   M     57
5   M     64
```

データフレーム x

SEX	HEIGHT	WEIGHT
F	158	51
F	162	55
M	177	72
M	173	57
M	166	64

データの加工・抽出

コマンド	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）

データの加工・抽出例

```
> DF$W <- ifelse(DF$SEX=="F", NA, DF$W)  # 女性の体重を隠す
> DF

  ID SEX   H  W
1  1   F 158 NA
2  2   F 162 NA
................

> cond <- (DF$H >= 170)   # H≧170 の人を抽出
> DF[cond,]

  ID SEX   H  W
3  3   M 177 72
4  4   M 173 57

> subset(DF, ID>3)        # ID>3の人を抽出

  ID SEX   H  W
4  4   M 173 57
5  5   M 166 64
```

DF

ID	SEX	H	W
1	F	158	51
2	F	162	55
3	M	177	72
4	M	173	57
5	M	166	64

データの加工・抽出例

```
> sum(DF$W)                # 体重の和を求める
> DF$W <- DF$W * 1000      # kg から g に変換する
> DF
  ID SEX   H     W
1  1   F 158 51000
2  2   F 162 55000
3  3   M 177 72000
4  4   M 173 57000
5  5   M 166 64000

> # 以下は DF$G <- DF$W * 1000 と同じ
> transform(DF, G=DF$W * 1000 )

  ID SEX   H     W
1  1   F 158 51000
2  2   F 162 55000
3  3   M 177 72000
.....................
```

DF

ID	SEX	H	W
1	F	158	51
2	F	162	55
3	M	177	72
4	M	173	57
5	M	166	64

データの加工・抽出例

```
> # W について横展開する
> y <- reshape(DF, v.names="W", idvar="SEX",
+              timevar="ID", direction="wide")
> y

  SEX W.1 W.2 W.3
1   F  51  55  NA
3   M  72  57  64

> # 性別（SEX）ごとに W の平均を求める
> apply(y[,2:4], 1, mean)

       1        3
      NA 64.33333
```

DF

SEX	ID	W
F	1	51
F	2	55
M	1	72
M	2	57
M	3	64

データの結合など

コマンド	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 を併合（マージ）する. ※通常は引数に all=T を指定し，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.

データの縦結合

```
> sex1    <- c("F","F")
> sex2    <- c("M","M","M")
> height1 <- c(158,162)
> height2 <- c(177,173,166)
> weight1 <- c( 51, 55)
> weight2 <- c(72, 57, 64)
> x <- data.frame(SEX=sex1, HEIGHT=height1, WEIGHT=weight1)
> y <- data.frame(SEX=sex2, HEIGHT=height2, WEIGHT=weight2)
> x
  SEX HEIGHT WEIGHT
1   F    158     51
2   F    162     55
> y
  SEX HEIGHT WEIGHT
1   M    177     72
2   M    173     57
3   M    166     64
> rbind(x, y)
  SEX HEIGHT WEIGHT
1   F    158     51
2   F    162     55
3   M    177     72
4   M    173     57
5   M    166     64
```

データのマージ例

```
### データフレーム x1 と x2 を併合（マージ）する
> x <- merge(x1, x2, by="NAME", all=T, sort=F)

### x1 と x3 で，マージするためのキー変数が異なる場合
> x <- merge(x1, x3, by.x="NAME", by.y="NAME2")
```

x1

NAME	DAY	KG_AM	KG_PM
Aさん	1	65.2	66.6
Cくん	1	71.3	71.5
D女史	1	62.7	61.7
B氏	1	58.5	58.0
B氏	2	58.1	59.5
Aさん	2	65.8	64.0
D女史	2	61.2	61.4
Cくん	2	71.2	70.2
Cくん	3	69.8	71.2
Aさん	3	63.5	61.5
B氏	3	59.0	58.6
D女史	3	59.8	59.9

＋

x2

NAME	GROUP
Aさん	Active
Cくん	Placebo
D女史	Active
B氏	Placebo

＋

x3

NAME2	GROUP
Aさん	Active
Cくん	Placebo
D女史	Active
B氏	Placebo

データのソート（1変数のソート）

```
### 行についてデータを NAME で整列（ソート；1変数）する
> sortlist <- order(x$NAME)        # 順番を取得
> X <- x[sortlist,]                # 整列
> rownames(X) <- c(1:nrow(X))      # 行番号の整形
```

データフレーム X

NAME	GROUP	DAY	KG_AM	KG_PM
Aさん	Active	1	65.2	66.6
Aさん	Active	2	65.8	64.0
Aさん	Active	3	63.5	61.5
B氏	Placebo	1	58.5	58.0
B氏	Placebo	2	58.1	59.5
B氏	Placebo	3	59.0	58.6
Cくん	Placebo	1	71.3	71.5
Cくん	Placebo	2	71.2	70.2
Cくん	Placebo	3	69.8	71.2
D女史	Active	1	62.7	61.7
D女史	Active	2	61.2	61.4
D女史	Active	3	59.8	59.9

データのソート（2変数のソート）

```
### 行についてデータを整列（ソート；2変数）する
### sortlist <- order(第1変数, pmax(第1変数, 第2変数))
> sortlist <- order(x$GROUP, pmax(x$GROUP, x$DAY))  # 順番を取得
> X <- x[sortlist,]                                 # 整列
> rownames(X) <- c(1:nrow(X))                       # 行番号の整形
> X
```

NAME	GROUP	DAY	KG_AM	KG_PM
Aさん	Active	1	65.2	66.6
D女史	Active	1	62.7	61.7
Aさん	Active	2	65.8	64.0
D女史	Active	2	61.2	61.4
Aさん	Active	3	63.5	61.5
D女史	Active	3	59.8	59.9
Cくん	Placebo	1	71.3	71.5
B氏	Placebo	1	58.5	58.0
B氏	Placebo	2	58.1	59.5
Cくん	Placebo	2	71.2	70.2
Cくん	Placebo	3	69.8	71.2
B氏	Placebo	3	59.0	58.6

欠損の扱い(1)

- 手入力でデータフレームを作成する場合で欠損が含まれているデータを読み込む場合は，ベクトル中の欠損部分をNA としておけば，該当部分に欠損値（NA）が入る

```
> sex <- c("F",NA,"M"); height <- c(158,162,NA);
> weight <- c(51,55,72)
> ( x <- data.frame(SEX=sex, HEIGHT=height, WEIGHT=weight) )

   SEX HEIGHT WEIGHT
1    F    158     51
2 <NA>    162     55
3    M     NA     72
```

欠損の扱い(2)

- ファイルからデータを読み込む場合で欠損が含まれているデータを読み込む場合は，データ間がコンマで区切られている方が処理しやすい
- この場合，単に欠損部分を空白にしておけば，該当部分に欠損値（NA）が入る

```
x <- read.table("data06.txt",
                header=T, sep=",")
  sex height weight
1   F    158     51
2   F    162     55
3   M     NA     72
4   M    173     57
5   M    166     64
```

C:¥

```
sex,height,weight
F,158,51
F,162,55
M,   ,72
M,173,57
M,166,64
```

data06.txt

４時間目のメニュー

- パッケージについて
  - パッケージとは
  - パッケージの呼び出し
  - 追加パッケージのインストール
- データハンドリング入門
  - データフレームとは
  - 種々のテキストファイルを R に読み込ませる方法
  - データハンドリング手法一覧
  - 演習 ←

データハンドリング例(導入)

- Aさん～D女史 (変数はNAME) の 4 人に，やせ薬 (GROUP==Active) か偽薬 (GROUP==Placebo) を投薬する
- 投薬してから1日目(DAY==1)～3日目(DAY==3)に体重を測定する
- 体重は，午前 (KG_AM) と午後 (KG_PM) の1日2回測定する

NAME	DAY	KG_AM	KG_PM
Aさん	1	65.2	66.6
Cくん	1	71.3	71.5
D女史	1	62.7	61.7
B氏	1	58.5	58.0
B氏	2	58.1	59.5
Aさん	2	65.8	64.0
D女史	2	61.2	61.4
Cくん	2	71.2	70.2
Cくん	3	69.8	71.2
Aさん	3	63.5	61.5
B氏	3	59.0	58.6
D女史	3	59.8	59.9

NAME	GROUP
Aさん	Active
Cくん	Placebo
D女史	Active
B氏	Placebo

データハンドリング例(導入)

- データの形式は EXCEL で保存されていると仮定する
- 体重に関するデータとグループに関するデータの 2 つ
- まずは 2 つのデータセットを R に読み込ませる

data1.xls ⇒ データフレーム x1 に

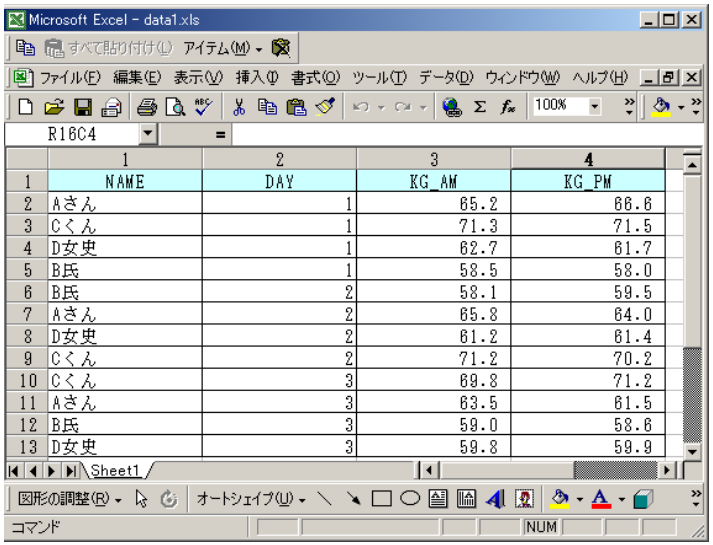
NAME	DAY	KG_AM	KG_PM
Aさん	1	65.2	66.6
Cくん	1	71.3	71.5
D女史	1	62.7	61.7
B氏	1	58.5	58.0
B氏	2	58.1	59.5
Aさん	2	65.8	64.0
D女史	2	61.2	61.4
Cくん	2	71.2	70.2
Cくん	3	69.8	71.2
Aさん	3	63.5	61.5
B氏	3	59.0	58.6
D女史	3	59.8	59.9

data2.xls ⇒ x2 に

NAME	GROUP
Aさん	Active
Cくん	Placebo
D女史	Active
B氏	Placebo

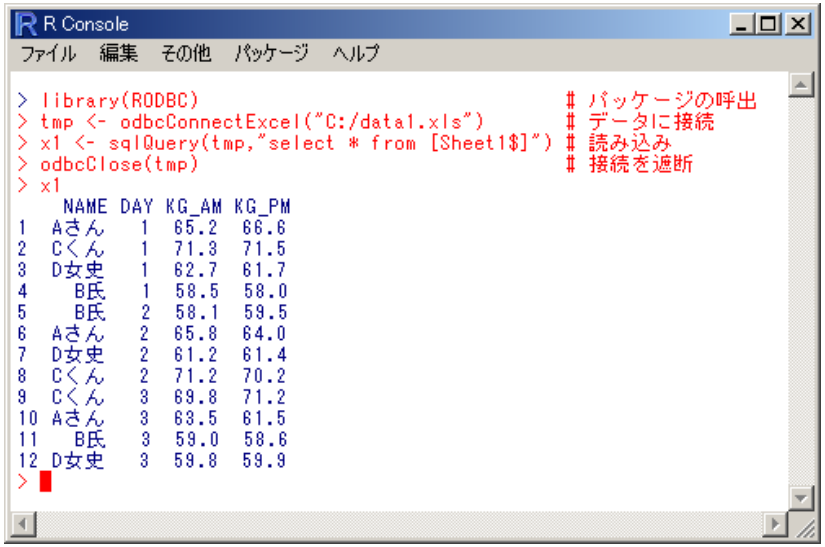
データハンドリング例(1)

```
### データの読み込み

> library(RODBC)                                    # パッケージの呼出
> tmp <- odbcConnectExcel("C:/data1.xls")           # データに接続
> x1 <- sqlQuery(tmp,"select * from [Sheet1$]")     # 読み込み
> odbcClose(tmp)                                    # 接続を遮断

> tmp <- odbcConnectExcel("C:/data2.xls")           # データに接続
> x2 <- sqlQuery(tmp,"select * from [Sheet1$]")     # 読み込み
> odbcClose(tmp)                                    # 接続を遮断
```



	1	2	3	4
1	NAME	DAY	KG_AM	KG_PM
2	Aさん	1	65.2	66.6
3	Cくん	1	71.3	71.5
4	D女史	1	62.7	61.7
5	B氏	1	58.5	58.0
6	B氏	2	58.1	59.5
7	Aさん	2	65.8	64.0
8	D女史	2	61.2	61.4
9	Cくん	2	71.2	70.2
10	Cくん	3	69.8	71.2
11	Aさん	3	63.5	61.5
12	B氏	3	59.0	58.6
13	D女史	3	59.8	59.9

読み込み



データハンドリング例(2)

```
### データフレーム x1 と x2 を併合（マージ）する
### 本当は列について自動で整列されるが，あえて整列させない

> x <- merge(x1, x2, by="NAME", all=T, sort=F)
```

データフレーム x

NAME	DAY	KG_AM	KG_PM	GROUP
Aさん	1	65.2	66.6	Active
Cくん	1	71.3	71.5	Placebo
D女史	1	62.7	61.7	Active
B氏	1	58.5	58.0	Placebo
B氏	2	58.1	59.5	Placebo
Aさん	2	65.8	64.0	Active
D女史	2	61.2	61.4	Active
Cくん	2	71.2	70.2	Placebo
Cくん	3	69.8	71.2	Placebo
Aさん	3	63.5	61.5	Active
B氏	3	59.0	58.6	Placebo
D女史	3	59.8	59.9	Active

データハンドリング例(3)

```
### 列の順番を入れ替える

> x <- x[,c(1,5,2,3,4)]
> x <- x[,c("NAME","GROUP","DAY","KG_AM","KG_PM")]
```

データフレーム x

NAME	GROUP	DAY	KG_AM	KG_PM
Aさん	Active	1	65.2	66.6
Cくん	Placebo	1	71.3	71.5
D女史	Active	1	62.7	61.7
B氏	Placebo	1	58.5	58.0
B氏	Placebo	2	58.1	59.5
Aさん	Active	2	65.8	64.0
D女史	Active	2	61.2	61.4
Cくん	Placebo	2	71.2	70.2
Cくん	Placebo	3	69.8	71.2
Aさん	Active	3	63.5	61.5
B氏	Placebo	3	59.0	58.6
D女史	Active	3	59.8	59.9

【演習】

NAME	GROUP	DAY	KG_AM	KG_PM
Aさん	Active	1	65.2	66.6
Aさん	Active	2	65.8	64.0
Aさん	Active	3	63.5	61.5
B氏	Placebo	1	58.5	58.0
B氏	Placebo	2	58.1	59.5
B氏	Placebo	3	59.0	58.6
Cくん	Placebo	1	71.3	71.5
Cくん	Placebo	2	71.2	70.2
Cくん	Placebo	3	69.8	71.2
D女史	Active	1	62.7	61.7
D女史	Active	2	61.2	61.4
D女史	Active	3	59.8	59.9

sample.csv

「3日目の体重の平均」と「1日目の体重の平均」の変化量が -2kg 未満の人を抽出してください

NAME
Aさん
D女史

- まず「sample.csv」を読み込んでください
- 順番の一例は「各日の平均を求める」→「『各日の平均』の列を横に展開する」→「3日目の平均－1日目の平均」→「抽出」です

データハンドリング例(4)

```
### 行についてデータを NAME で整列（ソート；1変数）する
> sortlist <- order(x$NAME)        # 順番を取得
> X <- x[sortlist,]                # 整列
> rownames(X) <- c(1:nrow(X))      # 行番号の整形
```

データフレーム X

NAME	GROUP	DAY	KG_AM	KG_PM
Aさん	Active	1	65.2	66.6
Aさん	Active	2	65.8	64.0
Aさん	Active	3	63.5	61.5
B氏	Placebo	1	58.5	58.0
B氏	Placebo	2	58.1	59.5
B氏	Placebo	3	59.0	58.6
Cくん	Placebo	1	71.3	71.5
Cくん	Placebo	2	71.2	70.2
Cくん	Placebo	3	69.8	71.2
D女史	Active	1	62.7	61.7
D女史	Active	2	61.2	61.4
D女史	Active	3	59.8	59.9

データハンドリング例(5)

```
### apply(列を指定，1(行毎という意味)，適用する関数)
> tmp <- apply(X[,4:5], 1, mean)
> tmp <- apply(X[,4:5], 1, function(x){sum(x)/length(x)} )

### 列を追加する関数 transform(データフレーム名，列名=ベクトル)
> X1  <- transform(X, MEAN=tmp)
```

NAME	GROUP	DAY	KG_AM	KG_PM	MEAN
Aさん	Active	1	65.2	66.6	65.90
Aさん	Active	2	65.8	64.0	64.90
Aさん	Active	3	63.5	61.5	62.50
B氏	Placebo	1	58.5	58.0	58.25
B氏	Placebo	2	58.1	59.5	58.80
B氏	Placebo	3	59.0	58.6	58.80
Cくん	Placebo	1	71.3	71.5	71.40
Cくん	Placebo	2	71.2	70.2	70.70
Cくん	Placebo	3	69.8	71.2	70.50
D女史	Active	1	62.7	61.7	62.20
D女史	Active	2	61.2	61.4	61.30
D女史	Active	3	59.8	59.9	59.85

データハンドリング例(6)

```
### 列の数を絞る transform(データフレーム[,残す列])
> X2 <- transform(X1[,c(1,2,3,6)])

### 列の数を絞る subset(データフレーム名,
###                          select=-c(削る列1,削る列2, ...))
> X2 <- subset(X1, select = -c(KG_AM, KG_PM))
```

NAME	GROUP	DAY	MEAN
Aさん	Active	1	65.90
Aさん	Active	2	64.90
Aさん	Active	3	62.50
B氏	Placebo	1	58.25
B氏	Placebo	2	58.80
B氏	Placebo	3	58.80
Cくん	Placebo	1	71.40
Cくん	Placebo	2	70.70
Cくん	Placebo	3	70.50
D女史	Active	1	62.20
D女史	Active	2	61.30
D女史	Active	3	59.85

データハンドリング例(7)

```
### 「DAY毎の平均」という列を追加
### 横展開する関数 reshape(データフレーム名, idvar=固定する列,
### timevar=横展開する列, direction="wide")
> X3 <- reshape(X2, idvar  =c("NAME","GROUP"),
+ timevar="DAY", direction="wide")
```

NAME	GROUP	MEAN.1	MEAN.2	MEAN.3
Aさん	Active	65.90	64.9	62.50
B氏	Placebo	58.25	58.8	58.80
Cくん	Placebo	71.40	70.7	70.50
D女史	Active	62.20	61.3	59.85

データハンドリング例(8)

```
###「3日目の体重の平均」－「1日目の体重の平均」
### という列「DIFF」を作成した後，列の数を絞る
> X4 <- transform(X3[,c("NAME","GROUP")],
                  DIFF = (X3$MEAN.3 - X3$MEAN.1) )
```

NAME	GROUP	DIFF
Aさん	Active	-3.40
B氏	Placebo	0.55
Cくん	Placebo	-0.90
D女史	Active	-2.35

データハンドリング例(9)

体重の変化量が -2kg 未満：subset(データフレーム，条件式)

```
> subset(X4, DIFF < -2)
```

NAME	GROUP	DIFF
Aさん	Active	-3.40
D女史	Active	-2.35

体重の変化量が -2kg 未満の人：subset(データ，条件式，列名)

```
> subset(X4, DIFF < -2, c(NAME))
```

NAME
Aさん
D女史

ちなみに，これはエラーとなる・・・

```
> subset(X4, DIFF<-2)
```

★演習（18枚目のスライド分）の回答例

```
> setwd("C:/work")                                  # 設問 2 の回答例
> x <- read.table("data.txt", head=T, sep=",")  # 設問 3 の回答例
> head(x, n=3)                                      # 設問 4 の回答例
> library(xlsReadWrite)                             # 設問 5 の回答
> ( x <- read.xls("data.xls", sheet=1)          # 設問 5 の回答例
```

４時間目にやったこと

- パッケージについて
  - パッケージとは
  - パッケージの呼び出し
  - 追加パッケージのインストール
- データハンドリング入門
  - データフレームとは
  - 種々のテキストファイルを R に読み込ませる方法
  - データハンドリング手法一覧
  - 演習

終