# Chapter 3 Data Input | Chapter 4 Data Frames

*Qianqian Shan*

*May 20, 2017*

## Data Input from the Keyboard

```
x <- scan()
```

Data input from files have variables called `fields` and rows called `cases`.

If the file name is forgotten, *file.choose()* could be used.

```
data <- read.table(file.choose(), header = TRUE)
```

- Data Input using `read.table`

```
data <- read.table("yields.txt", header = TRUE)
head(data)
```

```
##   sand clay loam
## 1    6   17   13
## 2   10   15   16
## 3    8    3    9
## 4    6   11   12
## 5   14   14   15
## 6   17   12   16
```

- `read.delim` can omit `header = T`

```
data <- read.delim("yields.txt")
```

- Data input using a defined function `rt`

```
rt <- function(x) read.table(paste(x, ".txt", sep = ''),header = TRUE)
data <- rt("yields")
head(data,2)
```

```
##   sand clay loam
## 1    6   17   13
## 2   10   15   16
```

As the default behavior of `read.table` is to convert character to factors, we need to use `as.is` to specify the columns that should =not be converted to factors.

```
murder <- read.table("murders.txt", header = TRUE)
str(murder)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ state     : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ population: int  3615 365 2212 2110 21198 2541 3100 579 8277 4931 ...
##  $ murder    : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ...
##  $ region    : Factor w/ 4 levels "North.Central",..: 3 4 4 3 4 4 2 3 3 3 ...
```

```
murder <- read.table("murders.txt", header = TRUE, as.is = "region")
str(murder)
```

```
## 'data.frame':    50 obs. of  4 variables:
```

```
##  $ state     : Factor w/ 50 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ population: int  3615 365 2212 2110 21198 2541 3100 579 8277 4931 ...
##  $ murder    : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ...
##  $ region    : chr  "South" "West" "West" "South" ...
```

Data input directly from the web

```
data2 <- read.table("http://www.bio.ic.ac.uk/research/mjcraw/therbook/data/cancer.txt", header = TRUE)
# URL stands for universal resource locator
head(data2)
```

```
##   death treatment status
## 1     4     DrugA      1
## 2    26     DrugA      1
## 3     2     DrugA      1
## 4    25     DrugA      1
## 5     7     DrugA      1
## 6     6     DrugA      0
```

## Read data using scan()

```
# scan will create a list of vectors and we'd like a data frame
data <- as.data.frame(scan("worms.txt", skip = 1, what= as.list(rep("", 7)))) # skip=1 will skip the he
# the last argument specify seven fields of character variables  ""

# the variable names are long and meanlingness, next obtain the names and apply them to the data
header <- unlist(scan("worms.txt", nlines = 1, what = as.list(rep("",7))))
header
```

```
## [1] "Field.Name"   "Area"         "Slope"        "Vegetation"
## [5] "Soil.pH"      "Damp"         "Worm.density"
```

```
names(data) <- header
head(data,2)
```

```
##       Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
## 1    Nashs.Field  3.6    11  Grassland     4.1    F            4
## 2 Silwood.Bottom  5.1     2     Arable     5.2    F            7
```

Input from complex file structures using scan

```
sapply(1:5, function(i) as.numeric(na.omit(scan("rt.txt", sep = "\t", quiet = TRUE)[(4*i - 3): (4*i)])))
```

```
## [[1]]
## [1] 138
##
## [[2]]
## [1] 27 44
##
## [[3]]
## [1]  19  20 345  48
##
## [[4]]
## [1]  115 2366
##
## [[5]]
## [1] 59
```

```
# quiet = T prevents the printing "Read 20 itmes"...
```

### Reading data from a file using `readLines`

This is an alternative of `scan`.

```
line <- readLines("worms.txt")
line
```

```
##  [1] "Field.Name\tArea\tSlope\tVegetation\tSoil.pH\tDamp\tWorm.density"
##  [2] "Nashs.Field\t3.6\t11\tGrassland\t4.1\tF\t4"
##  [3] "Silwood.Bottom\t5.1\t2\tArable\t5.2\tF\t7"
##  [4] "Nursery.Field\t2.8\t3\tGrassland\t4.3\tF\t2"
##  [5] "Rush.Meadow\t2.4\t5\tMeadow\t4.9\tT\t5"
##  [6] "Gunness.Thicket\t3.8\t0\tScrub\t4.2\tF\t6"
##  [7] "Oak.Mead\t3.1\t2\tGrassland\t3.9\tF\t2"
##  [8] "Church.Field\t3.5\t3\tGrassland\t4.2\tF\t3"
##  [9] "Ashurst\t2.1\t0\tArable\t4.8\tF\t4"
## [10] "The.Orchard\t1.9\t0\tOrchard\t5.7\tF\t9"
## [11] "Rookery.Slope\t1.5\t4\tGrassland\t5\tT\t7"
## [12] "Garden.Wood\t2.9\t10\tScrub\t5.2\tF\t8"
## [13] "North.Gravel\t3.3\t1\tGrassland\t4.1\tF\t1"
## [14] "South.Gravel\t3.7\t2\tGrassland\t4\tF\t2"
## [15] "Observatory.Ridge\t1.8\t6\tGrassland\t3.8\tF\t0"
## [16] "Pond.Field\t4.1\t0\tMeadow\t5\tT\t6"
## [17] "Water.Meadow\t3.9\t0\tMeadow\t4.9\tT\t8"
## [18] "Cheapside\t2.2\t8\tScrub\t4.7\tT\t4"
## [19] "Pound.Hill\t4.4\t2\tArable\t4.5\tF\t5"
## [20] "Gravel.Pit\t2.9\t1\tGrassland\t3.5\tF\t1"
## [21] "Farm.Wood\t0.8\t10\tScrub\t5.1\tT\t3"
```

Strip out the tab "\t"

```
db <- strsplit(line, "\t") # returns a set of lists
db <- (unlist(db))
dim(db) <- c(7, 21) # variable names dimention comes first
t(db)[-1, ] # the first row is the names
```

```
##       [,1]                [,2]  [,3] [,4]        [,5]  [,6] [,7]
##  [1,] "Nashs.Field"       "3.6" "11" "Grassland" "4.1" "F"  "4"
##  [2,] "Silwood.Bottom"    "5.1" "2"  "Arable"    "5.2" "F"  "7"
##  [3,] "Nursery.Field"     "2.8" "3"  "Grassland" "4.3" "F"  "2"
##  [4,] "Rush.Meadow"       "2.4" "5"  "Meadow"    "4.9" "T"  "5"
##  [5,] "Gunness.Thicket"   "3.8" "0"  "Scrub"     "4.2" "F"  "6"
##  [6,] "Oak.Mead"          "3.1" "2"  "Grassland" "3.9" "F"  "2"
##  [7,] "Church.Field"      "3.5" "3"  "Grassland" "4.2" "F"  "3"
##  [8,] "Ashurst"           "2.1" "0"  "Arable"    "4.8" "F"  "4"
##  [9,] "The.Orchard"       "1.9" "0"  "Orchard"   "5.7" "F"  "9"
## [10,] "Rookery.Slope"     "1.5" "4"  "Grassland" "5"   "T"  "7"
## [11,] "Garden.Wood"       "2.9" "10" "Scrub"     "5.2" "F"  "8"
## [12,] "North.Gravel"      "3.3" "1"  "Grassland" "4.1" "F"  "1"
## [13,] "South.Gravel"      "3.7" "2"  "Grassland" "4"   "F"  "2"
## [14,] "Observatory.Ridge" "1.8" "6"  "Grassland" "3.8" "F"  "0"
## [15,] "Pond.Field"        "4.1" "0"  "Meadow"    "5"   "T"  "6"
## [16,] "Water.Meadow"      "3.9" "0"  "Meadow"    "4.9" "T"  "8"
## [17,] "Cheapside"         "2.2" "8"  "Scrub"     "4.7" "T"  "4"
## [18,] "Pound.Hill"        "4.4" "2"  "Arable"    "4.5" "F"  "5"
## [19,] "Gravel.Pit"        "2.9" "1"  "Grassland" "3.5" "F"  "1"
## [20,] "Farm.Wood"         "0.8" "10" "Scrub"     "5.1" "T"  "3"
```

```r
# change it to data frame
frame <- as.data.frame(t(db)[-1,])
head(frame)
```

```
##                 V1  V2 V3        V4  V5 V6 V7
## 1     Nashs.Field 3.6 11 Grassland 4.1  F  4
## 2  Silwood.Bottom 5.1  2    Arable 5.2  F  7
## 3   Nursery.Field 2.8  3 Grassland 4.3  F  2
## 4     Rush.Meadow 2.4  5    Meadow 4.9  T  5
## 5 Gunness.Thicket 3.8  0     Scrub 4.2  F  6
## 6        Oak.Mead 3.1  2 Grassland 3.9  F  2
```

```r
# add names
names(frame) <- t(db)[1, ]
head(frame)
```

```
##          Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
## 1     Nashs.Field  3.6    11  Grassland     4.1    F            4
## 2  Silwood.Bottom  5.1     2     Arable     5.2    F            7
## 3   Nursery.Field  2.8     3  Grassland     4.3    F            2
## 4     Rush.Meadow  2.4     5     Meadow     4.9    T            5
## 5 Gunness.Thicket  3.8     0      Scrub     4.2    F            6
## 6        Oak.Mead  3.1     2  Grassland     3.9    F            2
```

Read non-standard files using `readLines`

```r
readLines("rt.txt")
```

```
## [1] "138\t\t\t"       "27\t44\t\t"      "19\t20\t345\t48" "115\t2366\t\t"
## [5] "59\t\t\t"
```

Split first on tabs and then on lines

```r
rows <- lapply(strsplit(readLines("rt.txt"), split = "\t"), as.numeric)
rows
```

```
## [[1]]
## [1] 138  NA  NA
##
## [[2]]
## [1] 27 44 NA
##
## [[3]]
## [1]  19  20 345  48
##
## [[4]]
## [1]  115 2366   NA
##
## [[5]]
## [1] 59 NA NA
```

```r
strsplit(readLines("rt.txt"), split = "\n") # this is ONE string
```

```
## [[1]]
## [1] "138\t\t\t"
##
## [[2]]
## [1] "27\t44\t\t"
##
## [[3]]
## [1] "19\t20\t345\t48"
##
## [[4]]
## [1] "115\t2366\t\t"
##
## [[5]]
## [1] "59\t\t\t"
```

```
# remove NAs from each of the vectors
sapply(1:5, function(i) as.numeric(na.omit(rows[[i]])))
```

```
## [[1]]
## [1] 138
##
## [[2]]
## [1] 27 44
##
## [[3]]
## [1]  19  20 345  48
##
## [[4]]
## [1]  115 2366
##
## [[5]]
## [1] 59
```

## Warnings when you `attach` the dataframe

The best approach is NOT to use `attach`

```
murder <- read.table("murders.txt", header = TRUE, as.is = "region")
attach(murder) # warning message shows up as the dataframe name is identical with one of the variable n
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     murder
```

```
               # better to rename the dataframe name
detach(murder)
data <- read.table("murders.txt", header = TRUE, as.is = "region")
attach(data)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     murder
```

```
detach(data)
```

Check files exists from the command line

```
file.exists("Decay.txt")
```

```
## [1] TRUE
```

Read dates and times from file, **refer Chapter 2 for instance**

**file paths** * set working directory by `setwd` * `basename` returns the last path of a complete path * `dirname` returns the full path except for the last

```
basename("c:/temp/thesis/data")
```

```
## [1] "data"
```

```
dirname("c:/temp/thesis/data")
```

```
## [1] "c:/temp/thesis"
```

```
file.path("","p1","p2","p3", c("file1", "file2"))
```

```
## [1] "/p1/p2/p3/file1" "/p1/p2/p3/file2"
```

```
basename(file.path("","p1","p2","p3", c("file1", "file2")))
```

```
## [1] "file1" "file2"
```

```
dirname(file.path("","p1","p2","p3","filename"))
```

```
## [1] "/p1/p2/p3"
```

# Chapter 4 : Dataframes

A **dataframe** is an object with rows and columns. Ways to create a dataframe:

- Use `read.table` to read fils .
- Use `data.frame` function to bind together a numner of vectors.

```
worms <- read.table("worms.txt", header = TRUE)
```

Summary of a dataframe

- `summary`
- `by` to summarize the dataframe on the basis of factor levels
- `aggregate`

```
# by(worms[,c(2,3)], worms$Vegetation, sum)
by(worms, worms$Vegetation, function(x) lm(Worm.density ~ Soil.pH, data= x))
```

```
## worms$Vegetation: Arable
##
## Call:
## lm(formula = Worm.density ~ Soil.pH, data = x)
##
## Coefficients:
## (Intercept)      Soil.pH
##      -9.689        3.108
##
## ---------------------------------------------------------
## worms$Vegetation: Grassland
##
```

```
## Call:
## lm(formula = Worm.density ~ Soil.pH, data = x)
##
## Coefficients:
## (Intercept)      Soil.pH
##     -15.041        4.265
##
## ---------------------------------------------------------
## worms$Vegetation: Meadow
##
## Call:
## lm(formula = Worm.density ~ Soil.pH, data = x)
##
## Coefficients:
## (Intercept)      Soil.pH
##          31           -5
##
## ---------------------------------------------------------
## worms$Vegetation: Orchard
##
## Call:
## lm(formula = Worm.density ~ Soil.pH, data = x)
##
## Coefficients:
## (Intercept)      Soil.pH
##           9           NA
##
## ---------------------------------------------------------
## worms$Vegetation: Scrub
##
## Call:
## lm(formula = Worm.density ~ Soil.pH, data = x)
##
## Coefficients:
## (Intercept)      Soil.pH
##      4.4758       0.1613
```

```r
aggregate(worms[, c(2, 3, 5, 7)], by = list(veg = worms$Vegetation), mean)
```

```
##         veg     Area    Slope  Soil.pH Worm.density
## 1    Arable 3.866667 1.333333 4.833333     5.333333
## 2 Grassland 2.911111 3.666667 4.100000     2.444444
## 3    Meadow 3.466667 1.666667 4.933333     6.333333
## 4   Orchard 1.900000 0.000000 5.700000     9.000000
## 5     Scrub 2.425000 7.000000 4.800000     5.250000
```

```r
# or with more than one classifying factors
aggregate(worms[, c(2, 3, 5, 7)], by = list(veg = worms$Vegetation, d = worms$Damp), mean)
```

```
##         veg     d     Area    Slope  Soil.pH Worm.density
## 1    Arable FALSE 3.866667 1.333333 4.833333     5.333333
## 2 Grassland FALSE 3.087500 3.625000 3.987500     1.875000
## 3   Orchard FALSE 1.900000 0.000000 5.700000     9.000000
## 4     Scrub FALSE 3.350000 5.000000 4.700000     7.000000
## 5 Grassland  TRUE 1.500000 4.000000 5.000000     7.000000
## 6    Meadow  TRUE 3.466667 1.666667 4.933333     6.333333
## 7     Scrub  TRUE 1.500000 9.000000 4.900000     3.500000
```

Note the *different* classes of these two:

```
class(worms[3, ])
```

```
## [1] "data.frame"
```

```
class(worms[, 3])
```

```
## [1] "integer"
```

Select rows from a dataframe randomly

```
worms[sample(1:20, 8, replace = FALSE), ]
```

```
##       Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 20      Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
## 16  Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 17      Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 15     Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 10 Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 1     Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 7    Church.Field  3.5     3  Grassland     4.2 FALSE            3
## 3  Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
```

```
# sample(x, size, replace = FALSE, prob = NULL)
```

### Sorting dataframes

```
worms[order(worms$Slope), ][1:3,]
```

```
##        Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 5 Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 8         Ashurst  2.1     0     Arable     4.8 FALSE            4
## 9     The.Orchard  1.9     0    Orchard     5.7 FALSE            9
```

```
# order reversely
worms[rev(order(worms$Slope)), ][1:5, ]
```

```
##           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 20          Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
## 11        Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 17          Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
```

```
# order by the first and ties broken by the second, third ...

worms[order(worms$Vegetation, worms$Worm.density), ][1:5, ]
```

```
##           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 8            Ashurst  2.1     0     Arable     4.8 FALSE            4
## 18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
## 2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
## 14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
## 12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
```

```
# select columns by variable names
worms[order(worms$Vegetation, worms$Worm.density), c("Vegetation", "Worm.density", "Soil.pH", "Slope")]
```

```
##    Vegetation Worm.density Soil.pH Slope
## 8      Arable            4     4.8     0
## 18     Arable            5     4.5     2
## 2      Arable            7     5.2     2
## 14  Grassland            0     3.8     6
## 12  Grassland            1     4.1     1
## 19  Grassland            1     3.5     1
```

### Using logical conditions to select rows from dataframe

```
worms[worms$Damp == TRUE, ]
```

```
##       Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
## 4    Rush.Meadow  2.4     5     Meadow     4.9 TRUE            5
## 10 Rookery.Slope  1.5     4  Grassland     5.0 TRUE            7
## 15    Pond.Field  4.1     0     Meadow     5.0 TRUE            6
## 16  Water.Meadow  3.9     0     Meadow     4.9 TRUE            8
## 17     Cheapside  2.2     8      Scrub     4.7 TRUE            4
## 20     Farm.Wood  0.8    10      Scrub     5.1 TRUE            3
```

```
# Use logical operator
worms[worms$Worm.density > median(worms$Worm.density) & worms$Soil.pH < 5.2, ]
```

```
##         Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 4      Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 5  Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 10   Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 15      Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 16    Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 18      Pound.Hill  4.4     2     Arable     4.5 FALSE            5
```

```
# extract all numeric columns
sapply(worms, is.numeric)
```

```
##   Field.Name         Area        Slope   Vegetation      Soil.pH
##        FALSE         TRUE         TRUE        FALSE         TRUE
##         Damp Worm.density
##        FALSE         TRUE
```

```
worms[, sapply(worms, is.numeric)]
```

```
##    Area Slope Soil.pH Worm.density
## 1   3.6    11     4.1            4
## 2   5.1     2     5.2            7
## 3   2.8     3     4.3            2
## 4   2.4     5     4.9            5
## 5   3.8     0     4.2            6
## 6   3.1     2     3.9            2
## 7   3.5     3     4.2            3
## 8   2.1     0     4.8            4
## 9   1.9     0     5.7            9
## 10  1.5     4     5.0            7
## 11  2.9    10     5.2            8
## 12  3.3     1     4.1            1
## 13  3.7     2     4.0            2
## 14  1.8     6     3.8            0
## 15  4.1     0     5.0            6
## 16  3.9     0     4.9            8
## 17  2.2     8     4.7            4
## 18  4.4     2     4.5            5
## 19  2.9     1     3.5            1
## 20  0.8    10     5.1            3
```

```
# similarly, extract all factor columns
worms[, sapply(worms, is.factor)]
```

```
##           Field.Name Vegetation
## 1        Nashs.Field  Grassland
## 2     Silwood.Bottom     Arable
## 3      Nursery.Field  Grassland
## 4        Rush.Meadow     Meadow
## 5    Gunness.Thicket      Scrub
## 6           Oak.Mead  Grassland
## 7       Church.Field  Grassland
## 8            Ashurst     Arable
## 9        The.Orchard    Orchard
## 10     Rookery.Slope  Grassland
## 11       Garden.Wood      Scrub
## 12      North.Gravel  Grassland
## 13      South.Gravel  Grassland
## 14 Observatory.Ridge  Grassland
## 15        Pond.Field     Meadow
## 16      Water.Meadow     Meadow
## 17         Cheapside      Scrub
## 18        Pound.Hill     Arable
## 19        Gravel.Pit  Grassland
## 20         Farm.Wood      Scrub
```

```
# exclude certain rows
worms[!(worms$Vegetation == "Grassland"), ]
```

```
##         Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 2   Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
## 4      Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 5  Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 8          Ashurst  2.1     0     Arable     4.8 FALSE            4
## 9      The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 11     Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 15      Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 16    Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 17       Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 18      Pound.Hill  4.4     2     Arable     4.5 FALSE            5
## 20       Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

```
# or use which function
worms[ - which(worms$Damp == FALSE), ]
```

```
##       Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
## 4    Rush.Meadow  2.4     5     Meadow     4.9 TRUE            5
## 10 Rookery.Slope  1.5     4  Grassland     5.0 TRUE            7
## 15     Pond.Field  4.1     0     Meadow     5.0 TRUE            6
## 16   Water.Meadow  3.9     0     Meadow     4.9 TRUE            8
## 17      Cheapside  2.2     8      Scrub     4.7 TRUE            4
## 20      Farm.Wood  0.8    10      Scrub     5.1 TRUE            3
```

```
# or
worms[!(worms$Damp == FALSE), ]
```

```
##       Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
## 4    Rush.Meadow  2.4     5     Meadow     4.9 TRUE            5
## 10 Rookery.Slope  1.5     4  Grassland     5.0 TRUE            7
## 15     Pond.Field  4.1     0     Meadow     5.0 TRUE            6
## 16   Water.Meadow  3.9     0     Meadow     4.9 TRUE            8
## 17      Cheapside  2.2     8      Scrub     4.7 TRUE            4
## 20      Farm.Wood  0.8    10      Scrub     5.1 TRUE            3
```

```
# or
worms[worms$Damp == TRUE, ]
```

```
##       Field.Name Area Slope Vegetation Soil.pH Damp Worm.density
## 4    Rush.Meadow  2.4     5     Meadow     4.9 TRUE            5
## 10 Rookery.Slope  1.5     4  Grassland     5.0 TRUE            7
## 15     Pond.Field  4.1     0     Meadow     5.0 TRUE            6
## 16   Water.Meadow  3.9     0     Meadow     4.9 TRUE            8
## 17      Cheapside  2.2     8      Scrub     4.7 TRUE            4
## 20      Farm.Wood  0.8    10      Scrub     5.1 TRUE            3
```

### Omitting rows containint missing values NA

- na.omit
- na.exclude , similar with na.omit, but different in the class of na.action attribute of the result, and thus na.exclude padded the original length by inserting NA for using naresid and napredict.
- complete.cases returns logical vector indicating which cases are complete

```
data <- read.table("worms.missing.txt", header = TRUE)
dim(data)
```

```
## [1] 20  7
```

```
nona <- na.omit(data)
dim(nona) # 3 NA values deleted
```

```
## [1] 17  7
```

```
nona1 <- na.exclude(data)
dim(nona1)
```

```
## [1] 17  7
```

```
#
complete.cases(data)
```

```
##  [1]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
```

```
# Analogue of na.omit
data[complete.cases(data), ]
```

```
##           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
## 4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
## 8            Ashurst  2.1     0     Arable     4.8 FALSE            4
## 9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
## 13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
## 14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
## 15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
## 20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

```
# check the number of NA values of each column
apply(data, 2, is.na)
```

```
##       Field.Name  Area Slope Vegetation Soil.pH  Damp Worm.density
##  [1,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
##  [2,]      FALSE FALSE  TRUE      FALSE   FALSE FALSE        FALSE
##  [3,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
##  [4,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
##  [5,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
##  [6,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
##  [7,]      FALSE FALSE FALSE      FALSE    TRUE  TRUE         TRUE
##  [8,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
##  [9,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [10,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [11,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [12,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [13,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [14,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [15,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [16,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [17,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [18,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
## [19,]      FALSE  TRUE FALSE      FALSE   FALSE FALSE        FALSE
## [20,]      FALSE FALSE FALSE      FALSE   FALSE FALSE        FALSE
```

```
# count the NA values
apply((apply(data, 2, is.na)), 2, sum)
```

```
##   Field.Name         Area        Slope   Vegetation      Soil.pH 
##            0            1            1            0            1 
##         Damp Worm.density 
##            1            1
```

### Using `order` and `!duplicated` to eliminate pseudoreplication

Extract each vegetation type and each has the highest density within that vegetation type.

```
# order data by density
new <- worms[rev(order(worms$Worm.density)), ]
new[!duplicated(new$Vegetation),]
```

```
##        Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 9     The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 16   Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 11    Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 10  Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 2  Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
```

### Complex ordering with mixed directions

There may be multiple sorting variables with different sorting directions.

```
# sort Vegetation in alphabetical order and density in decreasing order
worms[order(worms$Vegetation, -worms$Worm.density), ]
```

```
##           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
## 18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
## 8            Ashurst  2.1     0     Arable     4.8 FALSE            4
## 10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
## 3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
## 6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
## 13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
## 12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
## 19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
## 14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
## 16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

```
# As using minus sign only works for numerical variables, so for factors, we need to
# first use "rank" to convert levels to numeric
worms[order(-rank(worms$Vegetation), -worms$Worm.density), ]
```

```
##           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
## 5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
## 20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
## 9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
## 16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
## 15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
## 4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
## 1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
## 3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
## 6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
## 13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
## 12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
## 19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
## 14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
## 2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
## 18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
## 8            Ashurst  2.1     0     Arable     4.8 FALSE            4
```

```
# select columns that contains character "S"
grep("S", names(worms)) # returns the corresponding column number
```

```
## [1] 3 5
```

```
worms[, grep("S", names(worms))]
```

```
##    Slope Soil.pH
## 1     11     4.1
## 2      2     5.2
## 3      3     4.3
## 4      5     4.9
## 5      0     4.2
## 6      2     3.9
## 7      3     4.2
## 8      0     4.8
## 9      0     5.7
## 10     4     5.0
## 11    10     5.2
## 12     1     4.1
## 13     2     4.0
## 14     6     3.8
## 15     0     5.0
## 16     0     4.9
## 17     8     4.7
## 18     2     4.5
## 19     1     3.5
## 20    10     5.1
```

## A dataframe with row names instead of row numbers

Can suppress the creation of row numbers and allocate unique names to each row by altering the syntax of the read.table function. For example, add row.names= command.

```
worms2 <- read.table("worms.txt", header = TRUE, row.names = 1)
head(worms2) # row numbers are suppressed
```

```
##                   Area Slope Vegetation Soil.pH  Damp Worm.density
## Nashs.Field        3.6    11  Grassland     4.1 FALSE            4
## Silwood.Bottom     5.1     2     Arable     5.2 FALSE            7
## Nursery.Field    2.8     3  Grassland     4.3 FALSE            2
## Rush.Meadow      2.4     5     Meadow     4.9  TRUE            5
## Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## Oak.Mead         3.1     2  Grassland     3.9 FALSE            2
```

### Eliminating duplicated rows from a dataframe

```
dups <- read.table("dups.txt", header = TRUE)
dups # row 3 and 5 are the same
```

```
##   cow dog cat bat
## 1   1   2   3   1
## 2   1   2   2   1
## 3   3   2   1   1
## 4   4   4   2   1
## 5   3   2   1   1
## 6   6   1   2   5
## 7   1   2   3   2
```

```
# strip out duplicated rows
unique(dups)  # row numbers are still the original ones
```

```
##   cow dog cat bat
## 1   1   2   3   1
## 2   1   2   2   1
## 3   3   2   1   1
## 4   4   4   2   1
## 6   6   1   2   5
## 7   1   2   3   2
```

```
# the row that are duplicates
dups[duplicated(dups), ]
```

```
##   cow dog cat bat
## 5   3   2   1   1
```

### Dates in dataframes

```
nums <- read.table("sortdata.txt", header = TRUE)
head(nums, 3) # data is in format dmy
```

```
##     name       date   response treatment
## 1 albert 25/08/2003 0.05963704         A
## 2    ann 21/05/2003 1.46555993         A
## 3   john 12/10/2003 1.59406539         B
```

```
# In order to order the data by date, first need to convert date into date time format
# to avoid sorting based on day - month
dates <- strptime(nums$date, format = "%d/%m/%Y")
nums <- cbind(nums, dates)
head(nums[order(dates), ])
```

```
##       name       date response treatment      dates
## 49   albert 21/04/2003 30.66633         A 2003-04-21
## 63   james 24/04/2003 37.04140         A 2003-04-24
## 24    john 27/04/2003 12.70257         A 2003-04-27
## 33 william 30/04/2003 18.05707         B 2003-04-30
## 29 michael 03/05/2003 15.59891         B 2003-05-03
## 71     ian 06/05/2003 39.97238         A 2003-05-06
```

## Using `match` function in dataframes

```
herb <- read.table("herbicides.txt", header = TRUE)
# add corresponding recommended
recs <- data.frame(
  worms, hb = herb$Herbicide[match(worms$Vegetation, herb$Type)]
)
# match returns a vector of the positions of (first) matches of its first argument in its second
head(recs)
```

```
##        Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
## 1     Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
## 2  Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
## 3   Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
## 4     Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
## 5 Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
## 6        Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
##         hb
## 1 Allclear
## 2 Twinspan
## 3 Allclear
## 4 Propinol
## 5 Weedwipe
## 6 Allclear
```

## Merging two dataframes

```
lifeforms <- read.table("lifeforms.txt", header = TRUE)
flowering <- read.table("fltimes.txt", header = TRUE)
lifeforms
```

```
##    Genus     species lifeform
## 1   Acer platanoides     tree
## 2   Acer    palmatum     tree
## 3  Ajuga     reptans     herb
## 4 Conyza sumatrensis   annual
## 5 Lamium       album     herb
```

```
flowering
```

```
##       Genus       species flowering
## 1      Acer   platanoides       May
## 2     Ajuga       reptans      June
## 3  Brassica         napus     April
## 4 Chamerion angustifolium      July
## 5    Conyza     bilbaoana    August
## 6    Lamium         album   January
```

```
# two data have species in common

# merge with only rows had complete cases in both
merge(flowering, lifeforms)
```

```
##    Genus     species flowering lifeform
## 1   Acer platanoides       May     tree
## 2  Ajuga     reptans      June     herb
## 3 Lamium       album   January     herb
```

```
# include all
both <- merge(flowering, lifeforms, all = TRUE) # NA values produced
both
```

```
##       Genus       species flowering lifeform
## 1      Acer   platanoides       May     tree
## 2      Acer      palmatum      <NA>     tree
## 3     Ajuga       reptans      June     herb
## 4  Brassica         napus     April     <NA>
## 5 Chamerion angustifolium      July     <NA>
## 6    Conyza     bilbaoana    August     <NA>
## 7    Conyza   sumatrensis      <NA>   annual
## 8    Lamium         album   January     herb
```

```
# now add a new column from another data frame to the above data frame
seeds <- read.table("seedwts.txt", header = TRUE)
head(seeds) # columns have different names
```

```
##       name1         name2 seed
## 1      Acer   platanoides 32.0
## 2    Lamium         album 12.0
## 3     Ajuga       reptans  4.0
## 4 Chamerion angustifolium  1.5
## 5    Conyza     bilbaoana  0.5
## 6  Brassica         napus  7.0
```

```
merge(both, seeds, by.x = c("Genus", "species"), by.y = c("name1", "name2"))
```

```
##       Genus       species flowering lifeform seed
## 1      Acer      palmatum      <NA>     tree 21.0
## 2      Acer   platanoides       May     tree 32.0
## 3     Ajuga       reptans      June     herb  4.0
## 4  Brassica         napus     April     <NA>  7.0
## 5 Chamerion angustifolium      July     <NA>  1.5
## 6    Conyza     bilbaoana    August     <NA>  0.5
## 7    Conyza   sumatrensis      <NA>   annual  0.6
## 8    Lamium         album   January     herb 12.0
```

## Adding margins to a dataframe

```
frame <- read.table("sales.txt", header = TRUE)
frame
```

```
##              name spring summer autumn winter
## 1      Jane.Smith     14     18     11     12
## 2    Robert.Jones     17     18     10     13
## 3     Dick.Rogers     12     16      9     14
## 4 William.Edwards     15     14     11     10
## 5     Janet.Jones     11     17     11     16
```

```
# add row means
people <- rowMeans(frame[, -1])
people <- people - mean(people)
new.frame <- cbind(frame, people)
new.frame
```

```
##              name spring summer autumn winter people
## 1      Jane.Smith     14     18     11     12   0.30
## 2    Robert.Jones     17     18     10     13   1.05
## 3     Dick.Rogers     12     16      9     14  -0.70
## 4 William.Edwards     15     14     11     10  -0.95
## 5     Janet.Jones     11     17     11     16   0.30
```

```
# add col mean
season <- colMeans(frame[, -1])
season <- season - mean(season)  # cannot use rbind directly as columns are different

# copy one row
new.row <- new.frame[1, ]
new.row[1] <- "seasonal effects"
new.row[2:5] <- season
new.row[6] <- 0
new.frame <- rbind(new.frame, new.row)
new.frame
```

```
##               name spring summer autumn winter people
## 1       Jane.Smith  14.00  18.00  11.00  12.00   0.30
## 2     Robert.Jones  17.00  18.00  10.00  13.00   1.05
## 3      Dick.Rogers  12.00  16.00   9.00  14.00  -0.70
## 4  William.Edwards  15.00  14.00  11.00  10.00  -0.95
## 5      Janet.Jones  11.00  17.00  11.00  16.00   0.30
## 6 seasonal effects   0.35   3.15  -3.05  -0.45   0.00
```

```
# use sweep to subtract the grand mean from each value
gm <- mean(unlist(new.frame[1:5, 2:5])) # overall mean
gm <- rep(gm, 4)
new.frame[1:5, 2:5] <- sweep(new.frame[1:5, 2:5], 2, gm) # sweep out summary statistic


# put the grand/ overall mean in the bottom right corner
new.frame[6, 6] <- gm[1]
new.frame
```

```
##               name spring summer autumn winter people
## 1       Jane.Smith   0.55   4.55  -2.45  -1.45   0.30
## 2     Robert.Jones   3.55   4.55  -3.45  -0.45   1.05
## 3      Dick.Rogers  -1.45   2.55  -4.45   0.55  -0.70
## 4  William.Edwards   1.55   0.55  -2.45  -3.45  -0.95
## 5      Janet.Jones  -2.45   3.55  -2.45   2.55   0.30
## 6 seasonal effects   0.35   3.15  -3.05  -0.45  13.45
```