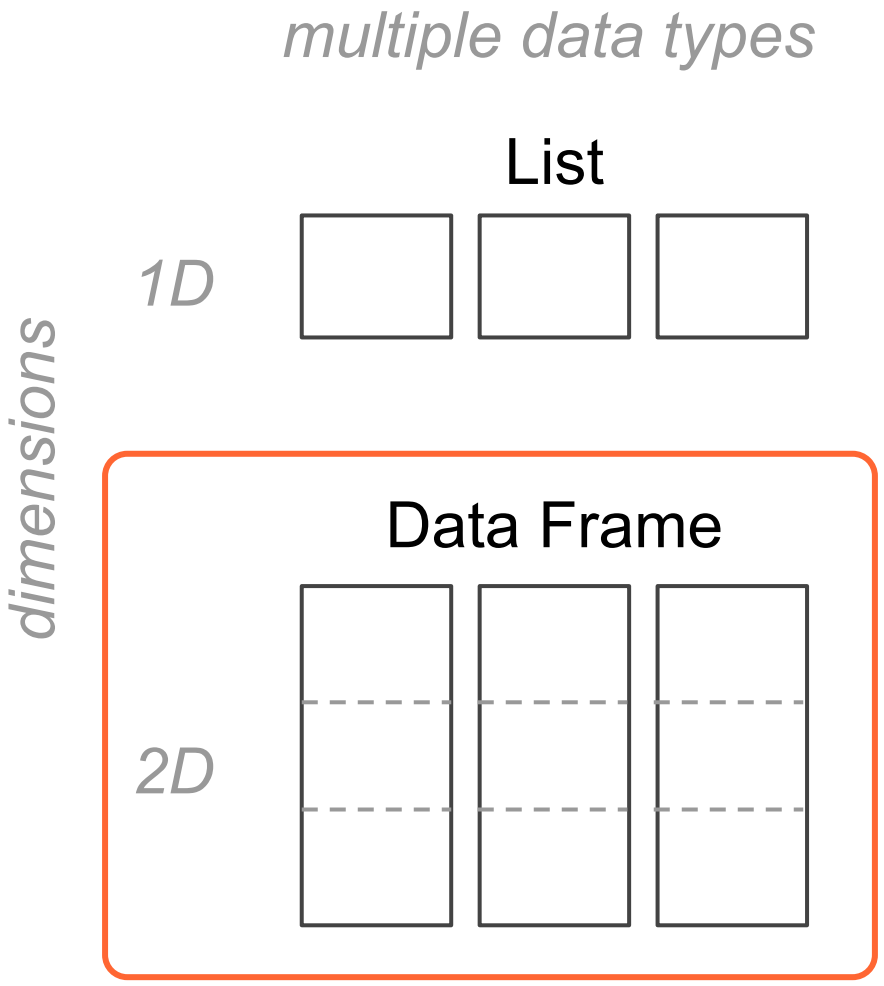
Importing Tables in R (1)

Stat 133 with Gaston Sanchez

Importing Tables in R

When using reading-table functions in R, a data table gets imported as a data frame


multiple data types
List
1D
dimensions
Data Frame
2D

R data frames reminder

R data frames are special kinds of lists

Stored in R as a list of vectors (or factors)

Columns are typically atomic structures

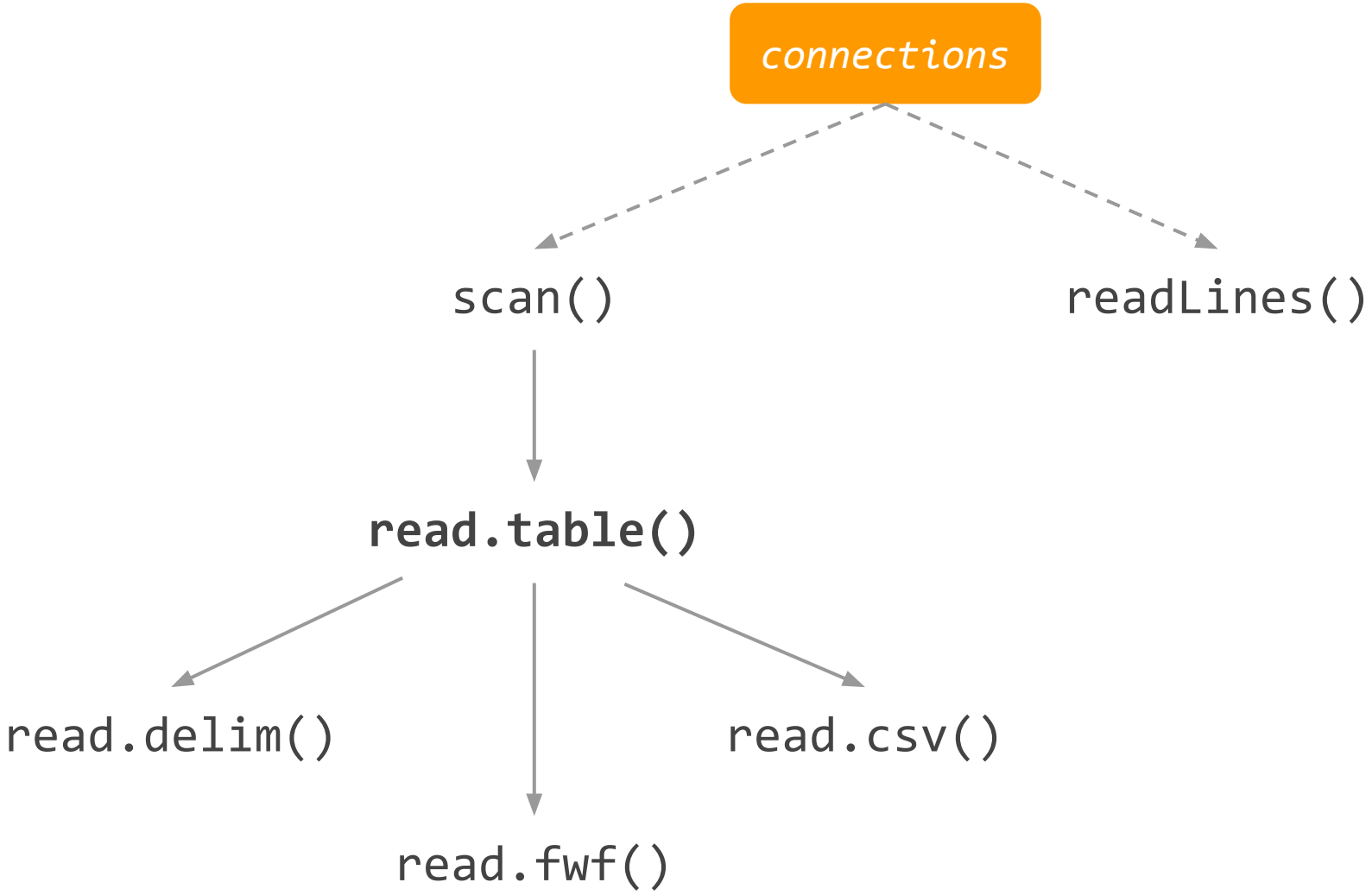
But since a data frame is a list, you can mix columns of different data types

Functions to import tables

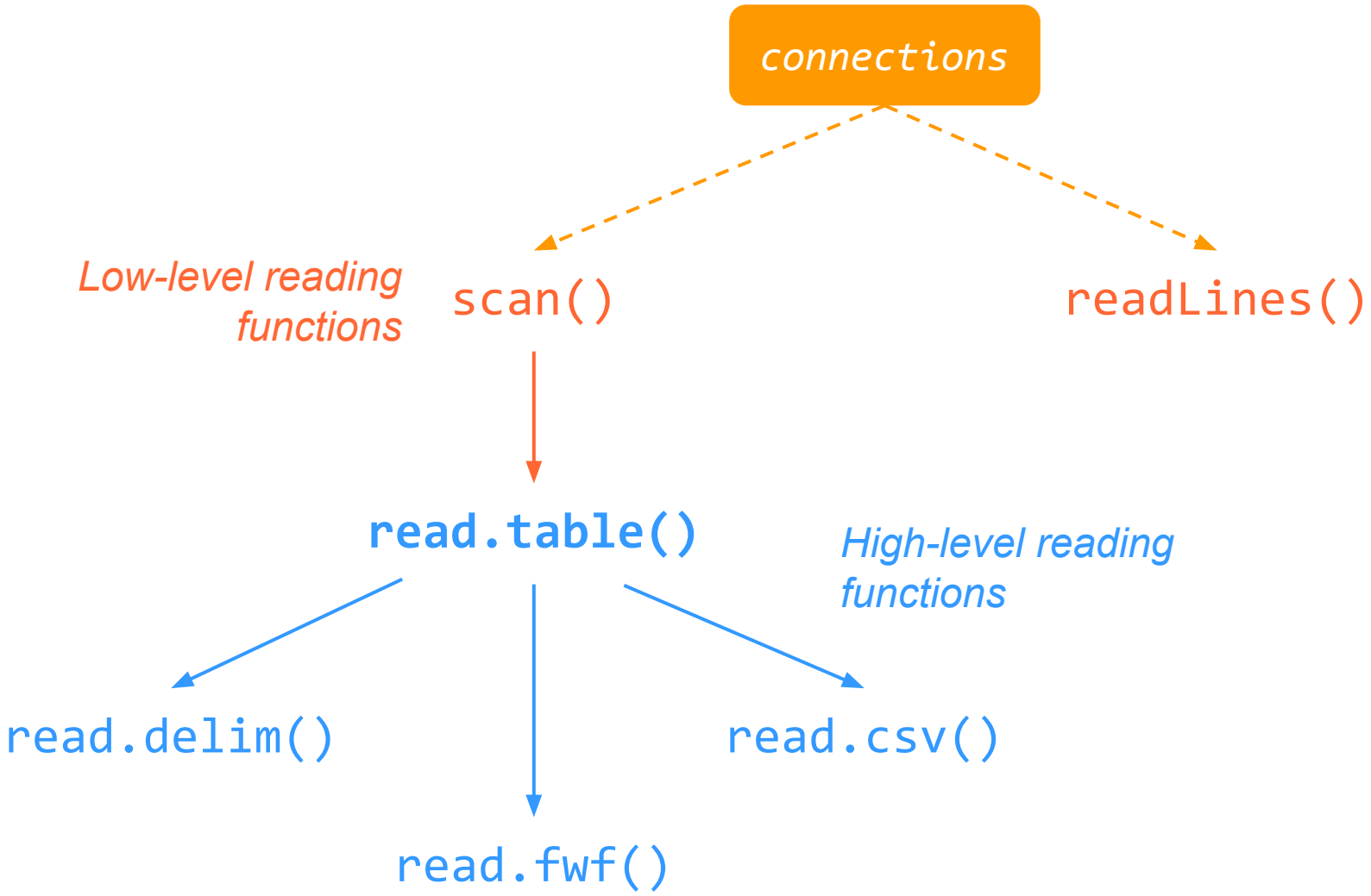
R comes with a family of functions that allows you to import most common data table formats

- `read.table()`
- `read.delim()`, `read.delim2()`
- `read.csv()`, `read.csv2()`
- `read.fwf()`

Base R functions to read data



Base R functions to read data



R Data Import Manual

There's a wide range of ways and options to import data tables in R.

The authoritative document to know almost all about importing (and exporting) data is the manual **R Data Import/Export**

https://cran.r-project.org/doc/manuals/r-release/R-data.html

Before importing a data table in R

What is the character(s) used as field delimiter?

Does the file contain names of columns?

Does the file contain a column for row names?

Are there any missing values?

How are missing values codified?

Do you want to read in all rows?

Before importing a data table in R

Do you need to convert delimiter characters? (e.g. from space to comma)

Can you determine the data-type of each column?

Are there any uninformative numbers?

Can you convert those uninformative numbers to informative labels?

Function `read.table()`

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", row.names, col.names,
           as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE,
           fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text,
           skipNul = FALSE)
```

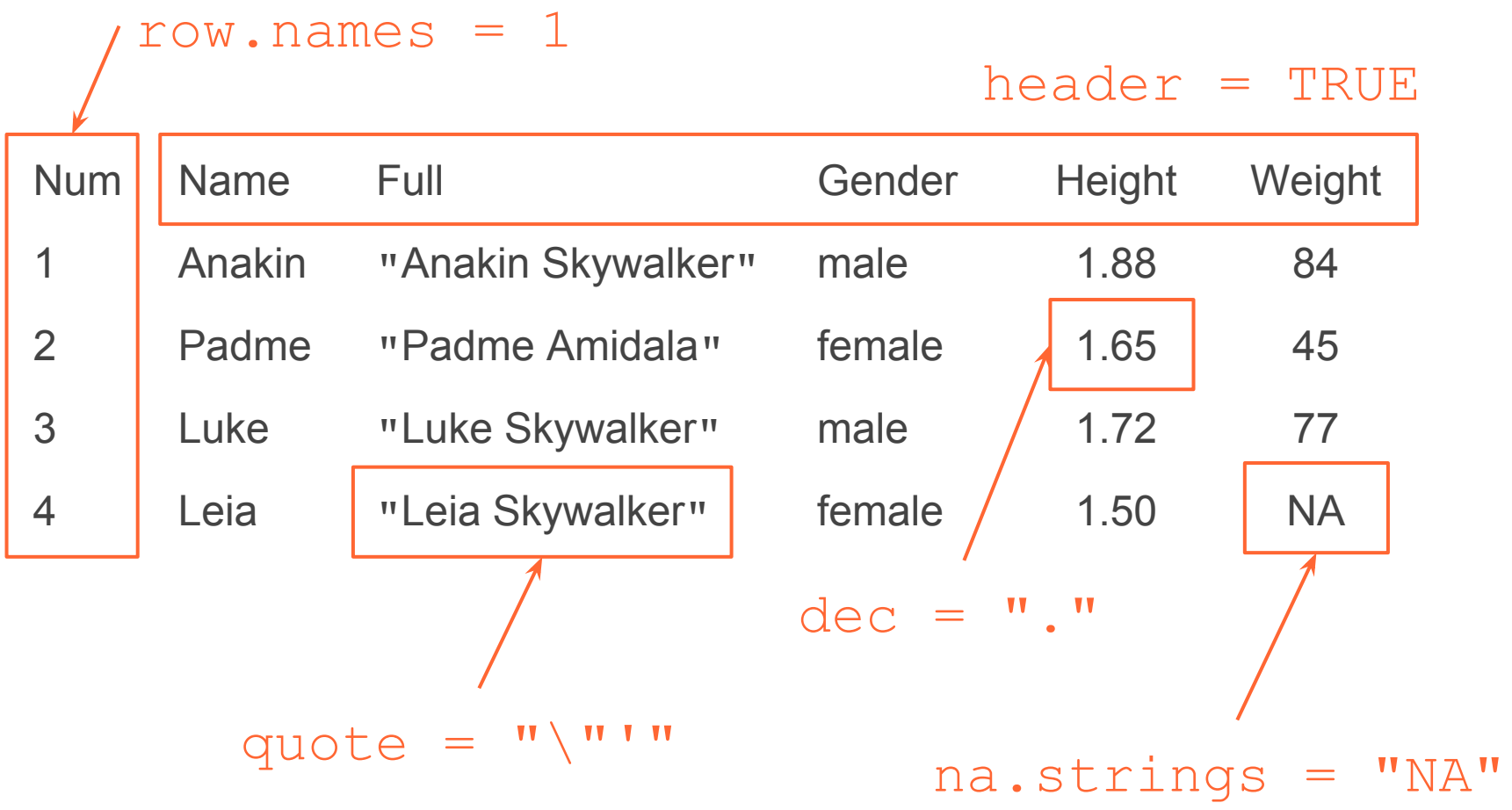
Some `read.table()` arguments

Argument	Description
`file`	Name of file
`header`	Whether column names are in 1st line
`sep`	Field separator
`quote`	Quoting characters
`dec`	Character for decimal point
`row.names`	Optional vector of row names
`col.names`	Optional vector of column names
`na.strings`	Characters treated as missing values
`colClasses`	Optional vector of data types for columns
`nrows`	Maximum number of rows to read in
`skip`	Number of lines to skip before reading data
`check.names`	Check valid column names
`stringsAsFactors`	Chould characters be converted to factors

Consider some data set (in a table)

Num	Name	Full	Gender	Height	Weight
1	Anakin	"Anakin Skywalker"	male	1.88	84
2	Padme	"Padme Amidala"	female	1.65	45
3	Luke	"Luke Skywalker"	male	1.72	77
4	Leia	"Leia Skywalker"	female	1.50	NA

Arguments for `read.table()`



Strings and Factors

By default, strings are converted to factors when loading data frames.

This is the wrong default

Use `stringsAsFactors = FALSE`

You should always explicitly convert strings into factors later

File: starwarstoy.csv

```
Name,Gender,Homeworld,Born,Jedi
Anakin,male,Tatooine,41.9BBY,yes
Amidala,female,Naboo,46BBY,no
Luke,male,Tatooine,19BBY,yes
Leia,female,Alderaan,19BBY,no
Obi-Wan,male,Stewjon,57BBY,yes
Han,male,Corellia,29BBY,no
Palpatine,male,Naboo,82BBY,no
R2-D2,unknown,Naboo,33BBY,no
```

Example

The main function to read in tables is **`read.table()`**; this example shows one way to import the CSV file of the previous slide

```
sw1 <- read.table(
  file = "starwarstoy.csv",
  header = TRUE,
  sep = ","
)
```

Example

By default, **`read.csv()`** assumes a header, and commas as field-delimiter

```
sw1 <- read.csv(
  file = "starwarstoy.csv"
)
```

Data types with `colClasses`

Whenever possible, specify the data type of each column

```
sw1 <- read.csv(
  file = "starwarstoy.csv",
  colClasses = c(
    "character",  # Name
    "factor",     # Gender
    "character",  # Homeworld
    "Character",  # Born
    "Factor")     # Jedi
)
```

Example: row names

```
# first column as row names
sw1 <- read.csv(
  file = "starwarstoy.csv",
  row.names = 1
)
```

Example: number of rows

```
# limit number of rows to read in
sw1 <- read.csv(
  file = "starwarstoy.csv",
  header = TRUE,
  row.names = 1,
  nrows = 4,
)
```

Example: skip rows

```
# skip the first row (no header)
sw1 <- read.table(
  file = "starwarstoy.csv",
  header = FALSE,
  skip = 1,
  row.names = 1,
  nrows = 4,
)
```

Other packages

Files from other programs

Type	Package	Function
Excel	gdata	`read.xls()`
Excel	xlsx	`read.xlsx()`
Excel	readxl	`read_excel()`
Excel	XLConnect	`readWorksheet()`
SPSS	foreign	`read.spss()`
SAS	foreign	`read.ssd()`
SAS	foreign	`read.xport()`
Matlab	R.matlab	`readMat()`
Stata	foreign	`read.dta()`
Octave	foreign	`read.octave()`
Minitab	foreign	`read.mtp()`
Systat	foreign	`read.systat()`

Data from google sheets?

Data from google sheets

R package “googlesheets” by Jennifer Bryan and Joanna Zhao

https://github.com/jennybc/googlesheets

After importing tables in R ....

There's a bunch of functions to inspect a data.frame object

Function	Description
`str()`	Structure
`head()`	First rows
`tail()`	Last rows
`summary()`	Descriptive statistics
`dim()`	Dimensions (# rows, # columns)
`nrow()`	Number of rows
`ncol()`	Number of columns
`names()`	Column names
`colnames()`	Column names
`rownames()`	Row names
`dimnames()`	List with row and column names

Functions to inspect a data frame

```
# display structure
str(airquality)

# display structure but showing
# few elements
str(airquality, vec.len = 1)
```

Functions to inspect a data frame

```
# first n rows
head(airquality, n = 5)

# last n rows
tail(airquality, n = 5)
```

Functions to inspect a data frame

```
# column summaries
summary(airquality)

# memory size
object.size(airquality)

# attributes
attributes(airquality)
```

Functions to inspect a data frame

```
# data frame dimensions
dim(airquality)

# number of rows
nrow(airquality)

# number of columns
ncol(airquality)
```

Functions to inspect a data frame

```
# row names
rownames(airquality)

# column names
colnames(airquality)

# column names
names(airquality)
```

Functions to inspect a data frame

```
# object class ('data.frame')
class(airquality)

# check if object is data.frame
is.data.frame(airquality)

# data.frame is also a list
is.list(airquality)
```