Working with Data

Arni Magnusson

Statistical Modeling in R

Universidad de Concepción

19–23 January 2026

Outline

Data Objects

numbers, strings, vectors, tables

Special Objects

lists, date/time, formula

Information

class, dimensions, names

Manipulation

subset, round, aggregate

Simple objects

integer	1	2	3
numeric	0.1	0.2	0.3
character	"one"	"two"	"three"
logical	TRUE	FALSE	TRUE

Is `x` integer?

```
is.integer(x)
```

Convert `x` to integer

```
as.integer(x)
```

Unusual values

`0`	zero	`x == 0`
`""`	empty string	`x == ""`
`NA`	not available	`is.na(x)`
`NULL`	not defined	`is.null(x)`
`Inf`	large positive number	`is.infinite(x)`
`-Inf`	large negative number	`is.infinite(x)`
`FALSE`	not TRUE	`!x`

Vector and factor

```
numbers <- c(10, 20, 30)
strings <- c("ten", "twenty", "thirty")
```

```
vec <- c("West", "Center", "East", "West", "Center")
fac <- factor(vec)
ord <- ordered(vec, levels=c("West","Center","East"))

table(fac)
table(ord)

plot(fac)
plot(ord)
```

Data frame, matrix, table

```
# numbers <- c(10, 20, 30)
# strings <- c("ten", "twenty", "thirty")
data.frame(one=numbers, two=strings)


matrix(c(10, 20, 30, 40), ncol=2)

mtcars
class(mtcars)
as.matrix(mtcars)

table(mtcars$cyl)
table(mtcars$am, mtcars$cyl)
```

Data frame, matrix, table

`data.frame`	data in columns (analysis, plots) **default choice** for statistical analysis supports `y∼x` formula notation supports `x$name` column selection
`matrix`	all values of same mode (linear algebra)
`table`	frequency table (view)
`tibble`	when using the tidyverse package
`data.table`	when using data.table package

List

```
list(one=rivers, two=TRUE, three=sleep, four=pi)
```

Date/time

```
as.Date("2026-01-19")
```

```
as.POSIXct("2026-01-19 23:59:59")
```

Formula

```
plot(mpg ~ cyl, data=mtcars)
```

```
lm(mpg ~ cyl, data=mtcars)
```

```
aggregate(mpg ~ cyl, data=mtcars, mean)
```

Object information

```r
length(rivers)
dim(mtcars)
nrow(mtcars)
ncol(mtcars)

names(mtcars)
colnames(mtcars)
rownames(mtcars)
dimnames(mtcars)
```

```r
mode(WorldPhones)
class(WorldPhones)
unclass(mtcars)
attributes(mtcars)

head(mtcars)
tail(mtcars)
unique(mtcars$cyl)
object.size(mtcars)
```

Names

```
v <- c(10, 20, 30)

names(v) <- c("one", "two", "three")
```

```
head(cars)

names(cars) <- c("s", "d")   # or colnames

dimnames(cars)
```

Logical expressions

```
pi == 3

pi != 3
```

```
pi < 3

pi <= 3
```

```
pi > 3

pi >= 3
```

Logical expressions

AND

```
logical && logical   # one value
vector  &  vector    # many values
```

OR

```
logical || logical   # one value
vector  |  vector    # many values
```

NOT

```
!logical
!vector
```

Logical expressions

```
pi > 3

is.character(pi)

!is.character(pi)



pi > 3  &&  is.character(pi)

pi > 3  ||  is.character(pi)

!(pi > 3  ||  is.character(pi))
```

Logical expressions

```
# numbers <- c(10, 20, 30)
# strings <- c("ten", "twenty", "thirty")
```

```
numbers >= 20
strings == "thirty"
numbers >= 20  |  strings == "thirty"
numbers >= 20  &  strings == "thirty"
```

```
any(numbers >= 20)
all(numbers >= 20)
```

Logical expressions

```
chickwts$feed=="soybean" | chickwts$feed=="casein"
```

```
chickwts$feed %in% c("soybean","casein")
```

Ways to subset

Vector (logical, integer, names)

```
islands[islands < 20]
islands[1:3]
islands[c("Greenland", "Iceland", "Britain")]
```

Data frame (dollar, logical, integer, names)

```
cars$dist
cars[1, 2]
cars[1:10, 1]
cars[,1]
```

List (dollar, logical, integer, names)

```
z <- list(one=rivers, two=TRUE, three=sleep, four=pi)
z$two
z["two"]      # returns a list of length one
z[["two"]]    # returns the value, same as z$two
```

Extract

Vector

```
v <- c(1, 3, 5, 7, 9)
v[1:3]
```

Data frame

```
x <- data.frame(num=v, char=letters[v])
x[1:3, "char"]
```

List

```
z <- list(one=rivers, two=TRUE, three=sleep, four=pi)
z$two
```

Replace

Vector

```
v <- c(1, 3, 5, 7, 9)
v[1:3] <- 0
v <- v[-(1:3)]
```

Data frame

```
x <- data.frame(num=v, char=letters[v])
x[1:3, "char"] <- ""
x <- x[-(1:3),]
```

List

```
z <- list(one=rivers, two=TRUE, three=sleep, four=pi)
z$two <- FALSE
z$two <- NULL
```

Subset summary

```
x[i]        x["name"]              x[c(TRUE,FALSE)]
```

select elements from vector

```
x[i,]       x["name",]             x[c(TRUE,FALSE),]  # row
x[,j]       x[,"name"]             x[,c(TRUE,FALSE)]  # column
x[i, j]     x["name", "name"]      x[c(TRUE,FALSE), c(TRUE,FALSE)]
```

select rows/columns/elements from data frame or matrix

```
x$name
```

select column in data frame, or element in list

Repeat, sample, order

```
rep(10, 3)
rep(1:10, 3)
rep(1:10, each=3)
rep(1:10, length=22)

sample(month.abb, 10, replace=TRUE)

sort(islands)
sort(islands, decreasing=TRUE)

rev(rivers)
order(rivers)  # rivers[order(rivers)]
```

Numbers

```
1:10
seq(1, 10, 0.5)
seq(1, 10, length=5)

rnorm(10, m=0, s=1)
runif(10, min=0, max=1)
rpois(10, lambda=1)

round(pi)

trunc(pi)

pi %% 1
```

String manipulation

```
nchar(month.name)

paste(month.abb[1], month.abb[3], sep="-")
paste(month.abb, collapse=".")

substring(month.abb, first=2, last=3)

grep("r", month.name)
month.abb[grep("r", month.name)]
month.abb[grep("r", month.name, invert=TRUE)]

gsub("J", "Y", month.abb)
```

Bind, apply, transpose

```
v <- 1:10
cbind(v)
cbind(v^2, log(v))
rbind(v)
rbind(v^2, log(v))

apply(WorldPhones, 1, sum)  # within row
apply(mtcars, 2, max)       # within column

a <- list(rivers=rivers, islands=islands, precip=precip)
sapply(mtcars, max)         # within element
sapply(a, median)           # within element

t(WorldPhones)
```

Aggregate and crosstab

```
aggregate(hp ~ cyl, data=mtcars, mean)
```

```
z <- aggregate(qsec ~ cyl+am, data=mtcars, mean)
xtabs(qsec ~ cyl+am, data=z)
```

Outline

Data Objects

numbers, strings, vectors, tables

Special Objects

lists, date/time, formula

Information

class, dimensions, names

Manipulation

subset, round, aggregate