

1 – Aan de slag

Rekenen

```
1 + 2
## [1] 3

3 * 4 + 5
## [1] 17

3 * (4 - 5)
## [1] -3

2^4
## [1] 16

(1 + 7) / (3 + 8)
## [1] 0.7272727
```

Variabelen

De toewijzingsoperator voor variabelen in R is <-.

```
x <- 2
y <- 3
x + y
## [1] 5
```

De operator = wordt ook gebruikt en is typisch bij het opgeven van functieparameters met een naam. In het voorbeeld hieronder worden 50 willekeurige normaal verdeelde getallen gegenereerd rond een gegeven gemiddelde en standaardafwijking.

```
# Geef parameters op in de juiste volgorde
rnorm(50, 0, 1)

## [1] -0.65974754  1.26352852 -2.11777398  1.38968401  0.67374609
## [6]  0.76240006  1.61377981 -0.73717639  0.62562519  1.58517901
## [11]  0.59289173 -0.48606756  0.71151914  1.28718861 -0.72893618
## [16] -0.40397811 -0.19917593 -0.10652196  0.12228084  0.63464467
## [21] -0.15373223  0.36174936  1.16995509 -0.09450837  1.94495639
## [26] -1.09190286 -0.79357209  1.42103711 -0.06224247 -0.27928047
## [31] -1.01987158 -1.60609431  0.01957670 -0.02331633  0.15247670
## [36] -0.23182457 -1.17568392 -0.19972437 -0.60800011 -1.19359977
```

```
## [41] 2.34877624 -1.73163975 -1.40959674 -0.73838159 -0.17111264
## [46] 0.94040016 -0.29469093 -0.50474338 -0.24741976 -1.13801533
# Benoem de parameters en geef ze een waarde
rnorm(50, mean = 0, sd = 1)
## [1] -0.98431633 -0.69133253 -1.72340047 0.90668062 -0.36828016
## [6] -1.43903894 -0.20928140 -0.44135044 -1.86222213 -1.05461634
## [11] 0.05153841 -0.06389314 -0.23658008 1.31746675 0.36694946
## [16] 0.09221109 0.23881310 0.35340102 -0.60346602 0.40443957
## [21] -0.34937959 -1.18639855 -0.40118019 -0.43454086 -0.30394871
## [26] 0.21923888 0.55686961 -0.57539337 -0.29409087 1.01251108
## [31] 2.20301177 0.30871346 -0.47533175 1.34929563 1.99641127
## [36] 0.64438271 -1.24171421 0.58737353 0.90801298 0.44994769
## [41] -0.38734774 -1.42563773 0.42284980 -0.88801335 0.22000594
## [46] 0.99998156 1.71415075 -0.34519414 -0.07260347 -0.42219617
```

De tweede notatie is leesbaarder, je moet je dan niet herinneren in welke volgorde de parameters moeten opgegeven worden. Rstudio kan de namen van parameters vervullen.

Lijsten/vectors

```
# Creëer lijsten
a <- c(1, 2, 3, 4)
b <- c(5, 6, 7, 8)
# Een lijst met de gehele getallen van 0 tot en met 10
c <- 0:10

# Bewerkingen op een lijst en een getal:
# Pas de bewerking toe op elk getal in de lijst
a + 5
## [1] 6 7 8 9
a - 9
## [1] -8 -7 -6 -5
a * 4
## [1] 4 8 12 16
a / 3
## [1] 0.3333333 0.6666667 1.0000000 1.3333333
a ^ 2
## [1] 1 4 9 16
```

```
# Bewerkingen op twee lijsten: Pas de bewerking toe op
# de overeenkomstige elementen uit beide lijsten
a + b
## [1] 6 8 10 12
a - b
## [1] -4 -4 -4 -4
a * b
## [1] 5 12 21 32
a / b
## [1] 0.2000000 0.3333333 0.4285714 0.5000000
a ^ b
## [1] 1 64 2187 65536
```

Bij het selecteren van elementen uit een lijst, moet je er op letten dat de index begint bij 1!

```
b[1] # Eerste element
## [1] 5
b[2:3] # Tweede t/m derde element
## [1] 6 7
```

Functies voor lijsten

```
length(a) # Lengte
## [1] 4
min(a) # Kleinste waarde
## [1] 1
max(a) # Grootste waarde
## [1] 4
sum(a) # Som van alle waarden
## [1] 10
```

Matrices

Het volgende voorbeeld creëert een 2d array of *matrix*. Je geeft een vlakke lijst op van alle waarden, en het aantal rijen en kolommen. Merk op dat de matrix kolom per kolom ingevuld wordt!

```
mat <- matrix(
  c(1, 2, 3,
    4, 5, 6),
```

```
nrow = 3,  
ncol = 2)  
mat
```

```
##      [,1] [,2]  
## [1,]    1    4  
## [2,]    2    5  
## [3,]    3    6
```

Selecteren uit een matrix:

```
mat[3,2] # Het element op de 3e rij, 2e kolom  
## [1] 6  
mat[,2] # De tweede kolom  
## [1] 4 5 6  
mat[1,] # De eerste rij  
## [1] 1 4  
mat[2:3,] # De tweede t/m derde rij  
##      [,1] [,2]  
## [1,]    2    5  
## [2,]    3    6
```

Data frames

De meest courante datastructuur voor het analyseren van data uit een experiment is een data frame. Die bestaat uit een aantal kolommen met een naam, en een aantal rijen.

Elke kolom kan je beschouwen als een statistische variabele en kan van een ander meetniveau zijn, hetzij een kwalitatieve (in R een *factor* genoemd, of ook *category* of *enumerated type*), hetzij een kwantitatieve.

We gaan eerst aan de slag met een data frame die standaard in R beschikbaar is onder de naam `mtcars`.

```
?mtcars # Bekijk de "man-page" voor deze dataset  
View(mtcars) # Bekijk de inhoud van deze dataset in tabelvorm  
nrow(mtcars) # Aantal rijen  
## [1] 32  
ncol(mtcars) # Aantal kolommen  
## [1] 11
```

Elementen uit een data frame ophalen:

```
mtcars[1,2] # eerste rij, tweede kolom  
## [1] 6
```

```
mtcars[3,] # derde rij
```

```
##           mpg cyl disp hp drat   wt  qsec vs am gear carb
## Datsun 710 22.8   4  108 93 3.85 2.32 18.61  1  1    4    1
```

```
mtcars[3:5,] # derde t/m vijfde rij
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Datsun 710      22.8   4  108   93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive  21.4   6  258  110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360  175 3.15 3.440 17.02  0  0    3    2
```

```
mtcars[,2] # tweede kolom
```

```
## [1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 8 6 8 4
```

```
mtcars[,2:4] # tweede t/m vierde kolom
```

```
##           cyl  disp  hp
## Mazda RX4      6 160.0 110
## Mazda RX4 Wag   6 160.0 110
## Datsun 710      4 108.0  93
## Hornet 4 Drive   6 258.0 110
## Hornet Sportabout 8 360.0 175
## Valiant         6 225.0 105
## Duster 360       8 360.0 245
## Merc 240D        4 146.7  62
## Merc 230         4 140.8  95
## Merc 280         6 167.6 123
## Merc 280C        6 167.6 123
## Merc 450SE       8 275.8 180
## Merc 450SL       8 275.8 180
## Merc 450SLC      8 275.8 180
## Cadillac Fleetwood 8 472.0 205
## Lincoln Continental 8 460.0 215
## Chrysler Imperial 8 440.0 230
## Fiat 128         4  78.7  66
## Honda Civic      4  75.7  52
## Toyota Corolla   4  71.1  65
## Toyota Corona    4 120.1  97
## Dodge Challenger  8 318.0 150
## AMC Javelin      8 304.0 150
## Camaro Z28       8 350.0 245
## Pontiac Firebird  8 400.0 175
```

```
## Fiat X1-9          4  79.0  66
## Porsche 914-2      4 120.3  91
## Lotus Europa       4  95.1 113
## Ford Pantera L     8 351.0 264
## Ferrari Dino       6 145.0 175
## Maserati Bora      8 301.0 335
## Volvo 142E         4 121.0 109
```

```
mtcars$cyl # kolom met naam "cyl"
```

```
## [1] 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 8 6 8 4
```

```
mtcars[c("mpg", "hp")] # tabel met enkel kolommen mpg en hp
```

```
##          mpg  hp
## Mazda RX4      21.0 110
## Mazda RX4 Wag  21.0 110
## Datsun 710     22.8  93
## Hornet 4 Drive  21.4 110
## Hornet Sportabout 18.7 175
## Valiant        18.1 105
## Duster 360     14.3 245
## Merc 240D      24.4  62
## Merc 230       22.8  95
## Merc 280       19.2 123
## Merc 280C      17.8 123
## Merc 450SE     16.4 180
## Merc 450SL     17.3 180
## Merc 450SLC    15.2 180
## Cadillac Fleetwood 10.4 205
## Lincoln Continental 10.4 215
## Chrysler Imperial 14.7 230
## Fiat 128       32.4  66
## Honda Civic    30.4  52
## Toyota Corolla 33.9  65
## Toyota Corona  21.5  97
## Dodge Challenger 15.5 150
## AMC Javelin    15.2 150
## Camaro Z28     13.3 245
## Pontiac Firebird 19.2 175
## Fiat X1-9      27.3  66
```

```
## Porsche 914-2      26.0  91
## Lotus Europa      30.4 113
## Ford Pantera L    15.8 264
## Ferrari Dino      19.7 175
## Maserati Bora     15.0 335
## Volvo 142E       21.4 109
```

Je kan ook een subset selecteren op basis van een soort query. Bijvoorbeeld, “Geef het aantal cylinders (cyl) van de auto’s die geen automatische versnellingsbak (am) hebben.”

```
mtcars$cyl[mtcars$am == 0]
```

```
## [1] 6 8 6 8 4 4 6 6 8 8 8 8 8 8 4 8 8 8 8
```

Of “Geef alle info over de auto’s met 6 cylinders.” Merk de komma op binnen de rechte haken!

```
mtcars[mtcars$cyl == 6,]
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1   4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1   4    4
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0   3    1
## Valiant        18.1   6 225.0 105 2.76 3.460 20.22  1  0   3    1
## Merc 280       19.2   6 167.6 123 3.92 3.440 18.30  1  0   4    4
## Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90  1  0   4    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.50  0  1   5    6
```

```
mtcars[mtcars$cyl <= 6,]
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1   4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1   4    4
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1   4    1
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0   3    1
## Valiant        18.1   6 225.0 105 2.76 3.460 20.22  1  0   3    1
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0   4    2
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0   4    2
## Merc 280       19.2   6 167.6 123 3.92 3.440 18.30  1  0   4    4
## Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90  1  0   4    4
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1   4    1
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1   4    2
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1   4    1
## Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01  1  0   3    1
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1   4    1
```

```
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2
mtcars[mtcars$cyl > 6,]
## mpg cyl disp hp drat wt qsec vs am gear carb
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8
```

Je kan ook zulke “queries” combineren met logische operatoren *en* (&), *of* (|) en *niet* (!).
Bijvoorbeeld, “Geef alle informatie over auto’s met een verbruik tussen 20 en 25 mpg.”

```
mtcars[mtcars$mpg >= 20 & mtcars$mpg <= 25,]
## mpg cyl disp hp drat wt qsec vs am gear carb
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2
```