

# R Studio

- R Studio: omgeving voor "ruwe" R (IDE)
- <http://www.rstudio.com/>

# Inleiding

- Waarom is R interessant?
- R vs SAS
  - In principe kunnen ze hetzelfde, maar...
  - In R kan je functies combineren, in SAS niet (= enorme flexibiliteit)
  - Uitstekende grafische mogelijkheden (wordt zeker een troef)
- R wordt een belangrijke tool, vooral binnen de wetenschap
- Leuke extratjes (rapporten, presentaties... )

# Inleiding

- Waarom is R interessant?
- R vs SAS
  - In principe kunnen ze hetzelfde, maar...
  - In R kan je functies combineren, in SAS niet (= enorme flexibiliteit)
  - Uitstekende grafische mogelijkheden (wordt zeker een topic)
- R wordt een belangrijke tool, vooral binnen de wetenschap
- Leuke extratjes (rapporten, presentaties... )

# Inleiding

- Waarom is R interessant?
- R vs SAS
  - In principe kunnen ze hetzelfde, maar...
  - In R kan je functies combineren, in SAS niet (= enorme flexibiliteit)
  - Uitstekende grafische mogelijkheden (wordt zeker een topic)
- R wordt een belangrijke tool, vooral binnen de wetenschap
- Leuke extratjes (rapporten, presentaties... )

# Inleiding

- Waarom is R interessant?
- R vs SAS
  - In principe kunnen ze hetzelfde, maar...
  - In R kan je functies combineren, in SAS niet (= enorme flexibiliteit)
  - Uitstekende grafische mogelijkheden (wordt zeker een topic)
- R wordt een belangrijke tool, vooral binnen de wetenschap
- Leuke extratjes (rapporten, presentaties... )

# Inleiding

- Waarom is R interessant?
- R vs SAS
  - In principe kunnen ze hetzelfde, maar...
  - In R kan je functies combineren, in SAS niet (= enorme flexibiliteit)
  - Uitstekende grafische mogelijkheden (wordt zeker een topic)
- R wordt een belangrijke tool, vooral binnen de wetenschap
- Leuke extratjes (rapporten, presentaties... )

# Inleiding

- Waarom is R interessant?
- R vs SAS
  - In principe kunnen ze hetzelfde, maar...
  - In R kan je functies combineren, in SAS niet (= enorme flexibiliteit)
  - Uitstekende grafische mogelijkheden (wordt zeker een topic)
- R wordt een belangrijke tool, vooral binnen de wetenschap
- Leuke extratjes (rapporten, presentaties... )

# Inleiding

- Waarom is R interessant?
- R vs SAS
  - In principe kunnen ze hetzelfde, maar...
  - In R kan je functies combineren, in SAS niet (= enorme flexibiliteit)
  - Uitstekende grafische mogelijkheden (wordt zeker een topic)
- R wordt een belangrijke tool, vooral binnen de wetenschap
- Leuke extratjes (rapporten, presentaties... )



# Overzicht

R heeft meerdere data structuren:

- Vector
- Matrix & Array
- List
- Data frame (cfr SAS & SPSS)
- Factor & table

# Definitie en types

- Vector: set van elementen van dezelfde soort (*mode*, type variabele)
- Belangrijkste datasoorten:
  - Integer
  - Numeric
  - Logical (Boolean)
  - Character
- Vergelijkbaar met een SAS-variabele, maar veel ruimer gebruik!

# Voorbeelden vectoren

- Voorbeeld *numeric vector*

```
[1] 0.2 6.1 9.3 -4.4 6.7
```

- Voorbeeld *character vector*

```
[1] "een"          "twee"          "spaties ook"
```

- Voorbeeld *boolean vector*

```
[1] TRUE FALSE FALSE TRUE
```

# Niet zomaar een dataset variabele

Onze numerieke vector...

```
> print(numeric.v)
```

```
[1] 0.2 6.1 9.3 -4.4 6.7
```

... kan als basis dienen voor een logische vector:

```
> x <- (numeric.v > 0)
```

```
> print(x)
```

```
[1] TRUE TRUE TRUE FALSE TRUE
```

# Vectorelementen toevoegen en verwijderen

```
> x <- c(5,3,7,9)
```

```
> length(x)
```

```
[1] 4
```

```
> print(x)
```

```
[1] 5 3 7 9
```

```
> x <- c(x,11)
```

```
> print(x)
```

```
[1] 5 3 7 9 11
```

```
> y <- x[1:2]
```

```
> print(y)
```

```
[1] 5 3
```

# Vectorelementen toevoegen en verwijderen

```
> print(x)
```

```
[1]  5  3  7  9 11
```

```
> y <- c(x[1:2], 20, x[3:5])
```

```
> print(y)
```

```
[1]  5  3 20  7  9 11
```

# Vectorelementen vervangen

```
> print(x)
```

```
[1]  5  3  7  9 11
```

```
> x[2] <- 13
```

```
> print(x)
```

```
[1]  5 13  7  9 11
```

# Operaties

```
> x <- c(1,2,7)
```

```
> x + 1
```

```
[1] 2 3 8
```

```
> x*2
```

```
[1] 2 4 14
```



# Operaties

```
> x <- c(1,2,7)
> y <- c(3,5,11)
> x+y

[1]  4  7 18
```

# Recycling

```
> x <- c(1,2,7)
> y <- c(3,5)
> x+y

[1]  4  7 10
```