


# Inleiding tot

Ludo Poelaert\*  
UGent



Departement Industriële Systemen en Design  
Phone: +32 477 99 25 97

December 26, 2017

Deze inleiding is bedoeld voor hen die statistische berekeningen wensen te maken en die geen 'data-analyse' programma's wensen aan te schaffen. In de handel vindt men uitstekende programma's zoals "SPSS", "Statgraphics Centurion", "Stata" en nog zoveel meer. Wie een uitgebreide lijst wil van software met statistische mogelijkheden, verwijs ik naar de wikipedia website : "Stat programs by Wiki".



 is een data-analyse programma uit de open-source gemeenschap. Je kan het dus kosteloos en vrij gebruiken onder de GNU licentie.

Je kan de GNU website vinden op volgend webadres : "GNU web pagina"

Bij het gebruik van "open source software" (zoals ) is het correct een korte referentie naar  in je publicatie op te nemen. Hiermee erken je de waarde van het pakket en breng je "hulde" aan de makers ervan.



Hoe je  citeert zal ik later aangeven.

## 1 Waar vind je R?



Het programma  kan gedownload worden via de R project Home pagina "<http://www.r-project.org/>" Op deze pagina vind je onder de rubriek "Getting started" een download knop. Je kan  downloaden voor Windows, Linux en MacOS. Als je de "download R" knop hebt aangeklikt, dan kom je op het "Comprehensive R Archive Network", afgekort als "CRAN". "CRAN" is een netwerk van "ftp" en "web" servers. Deze servers stockeren identische up-to-date versies van de code en van de documentatie van . Het is namelijk van één van deze "CRAN" mirror sites dat je jouw copie van  kan afhalen. Ga bijvoorbeeld naar de Belgische mirrorsite van  op de K.U.Leuven. Je komt dan op het zogenaamde "CRAN" uit. Hier kies je het beheerssysteem van je computer.

---

\*Ik wil graag mijn twee zonen, Jeroen en Ruben Poelaert, bedanken voor de intense samenwerking en verbetering van de vele versies van dit document




Je kiest uiteraard die knop die past bij jouw beheerssysteem : Windows, Unix of MacOS X. Je komt dan op de pagina die je toelaat de recentste versie van  te downloaden naar je computer. Op datum van 25 Augustus 2011 is de laatste versie van  versie 2.13.1.

## 1.1 -installatie voor MacOSX

Na de download krijg je op de Mac een package “R-2.13.1.pkg”. Dat pakket installeer je zoals elk ander MacOSX programma. Eenmaal geïnstalleerd, sleep je het  icoon naar je “dock” en vanaf nu ben je klaar om met  te werken.

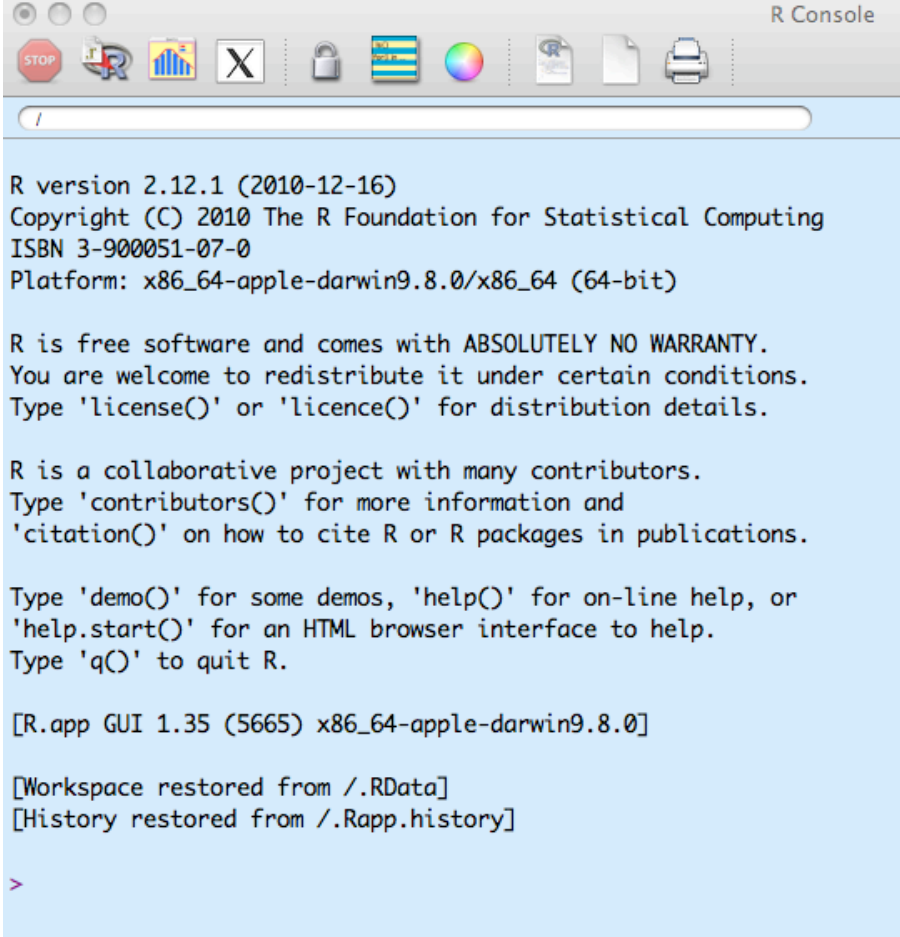
## 1.2 -installatie voor Windows

Eenmaal je “Download R for Windows” hebt gekozen, kom je op een pagina waar je een knop vindt “Install R for the first time”. Klik op deze knop. Je komt op een pagina waar je een nieuwe knop vindt met als titel : “Download R 2.13.1 for Windows”. Als je hier op klikt, bekom je een “exe” bestand, nl. “R-2-1.13.1-win.exe”

Dat kan je uitvoeren en via de setup wizard bekom je een geïnstalleerd  programma. Je kan het  logo naar de taskbar slepen om gemakkelijk  op te starten.

## 2 Enkele algemeenheden over

Ik stel voor dat je  lanceert. Je bekommt een scherm dat er als volgt uitziet.



```
R version 2.12.1 (2010-12-16)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.



Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.35 (5665) x86_64-apple-darwin9.8.0]

[Workspace restored from /.RData]
[History restored from /.Rapp.history]

>
```

Figure 1: Het beginscherm bij 



Je zit nu in de zogenaamde -console. Zoals je ziet is  een commando-gestuurd programma. Onderaan het scherm staat een groter dan teken “>”. Dat is de -“prompt” genoemd.  verwacht dat je een commando intikt na deze prompt en daarna op de returntoets drukt. Om de leesbaarheid van deze tekst te verhogen, zullen de zaken die door jou moeten worden ingetikt in het **blauw** worden getoond. Voor alle duidelijkheid : de -prompt staat reeds op de console en hoeft niet te worden ingetikt.


Bijvoorbeeld :


```
>date()
```


 komt onmiddellijk terug met een antwoord :

```
[1] “Thu Aug 25 09:17:58 2011”
```

Het  programma is uitstekend gedocumenteerd. Via het wereld wijde web kan je een uitgebreide documentatie raadplegen. Op de reeds vermelde hoofdpagina van , nl. <http://www.r-project.org/>, bevindt zich links een menu “documentation”. Hier vind je manuals en veel gestelde vragen (FAQ).

Er bestaat een uitgebreide literatuur over . Persoonlijk gebruik ik de volgende boeken :

- A Handbook of Statistical Analyses Using , van Brian Everitt en Torsten Hothorn
- The R-Book van Michael Crawley
- The R Full Reference Manual, gemaakt door het R Development Core Team

Het -programma bevat zelf ook een hulp-motor. Je tikt hiervoor gewoon een vraagteken in gevolgd door het commando waarover je hulp wenst. Meteen vlieg je op het wereld wijde web naar de hulp-server en krijg je voor het betreffende commando een uitvoerige uitleg. Er worden zelfs voorbeelden gegeven hoe je het commando moet gebruiken.

Voorbeeld :

```
>?mean
```

Je bekomt dan volgend scherm :

```
mean (base) R Documentation
```

**Arithmetic Mean**

**Description**

Generic function for the (trimmed) arithmetic mean.

**Usage**

```
mean(x, ...)
```

## Default S3 method:  
mean(x, trim = 0, na.rm = FALSE, ...)

**Arguments**

**x** An R object. Currently there are methods for numeric logical vectors and [date](#), [date-time](#) and [time interval](#) objects, and for data frames all of whose columns have a method. Complex vectors are allowed for `trim = 0`, only.

**trim** the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.

**na.rm** a logical value indicating whether NA values should be stripped before the computation proceeds.

... further arguments passed to or from other methods.

Figure 2: Het hulpscherm voor de functie "mean"

### 3 als wetenschappelijke rekenmachine

Stel dat we de vierkantswortel van 679 willen berekenen.

Je schrijft na de -prompt het commando : `sqrt(679)` en drukt op de returntoets.

```
>sqrt(679)
```

 komt terug met het antwoord :

```
>[1] 26.05763
```

Een ander voorbeeld :



We berekenen 2 maal  $\pi$

We schrijven na de R-prompt `2 * pi`

```
>2*pi
```

 komt terug met het volgende antwoord :

```
>[1] 6.283185
```


Zoals je ziet geeft  voor beide berekeningen het antwoord met 7 cijfers. Intern houdt  veel meer cijfers bij, maar laat er blijkbaar 7 zien.

Stel dat je het antwoord wil met 14 cijfers, dan gebruikt men volgend commando:

```
>print(2*pi,digits=14)
```

 komt terug met het antwoord

```
>[1] 6.2831853071796
```



De klassieke wiskundige operatoren in  zijn +,-,\*,/. De operator voor de machtsverheffing is  $\wedge$ .

Stel dat je 5 tot de 4<sup>de</sup> macht wil berekenen. Je schrijft dan

```
>5^4
```

 komt terug met het antwoord

```
>[1] 625
```

 is een krachtige rekenmachine. Zo kan  gemakkelijk complexe berekeningen aan. Een voorbeeld :

We berekenen het product van 2 matrices A maal B.

Matrix A

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

Matrix B

$$\begin{pmatrix} 10 & 13 & 16 \\ 11 & 14 & 17 \\ 12 & 15 & 18 \end{pmatrix}$$

We voeren de matrices in op de volgende manier :

```
>A = matrix(c(1,2,3,4,5,6,7,8,9),nrow=3)
```

```
>B = matrix(c(10,13,16,11,14,17,12,15,18),nrow=3)
```

Het product van beide matrices wordt bekomen door :

```
>A%%B
```


De operator om twee matrices met elkaar te vermenigvuldigen is niet \* maar %% !

 komt terug met het antwoord

```
> [,1] [,2] [,3]
[1,] 138 174 210
[2,] 171 216 261
[3,] 204 258 312
```

Het product van de matrices is

$$\begin{pmatrix} 138 & 174 & 210 \\ 171 & 216 & 261 \\ 204 & 258 & 312 \end{pmatrix}$$

Om de kracht van  te ervaren, zullen we een 10 x 10 matrix verheffen tot de 10de macht. We beperken de getallen tot 2 cijfers na de komma.

Matrix TenByTen =

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\ 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 & 49 & 50 \\ 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 59 & 60 \\ 61 & 62 & 63 & 64 & 65 & 66 & 67 & 68 & 69 & 70 \\ 71 & 72 & 73 & 74 & 75 & 76 & 77 & 78 & 79 & 80 \\ 81 & 82 & 83 & 84 & 85 & 86 & 87 & 88 & 89 & 90 \\ 91 & 92 & 93 & 94 & 95 & 96 & 97 & 98 & 99 & 100 \end{pmatrix}$$

Om deze matrix in te voeren in de  $\mathbb{R}$ -console, tik je volgend commando in:

```
>TenByTen = matrix(c(1,11,21,31,41,51,61,71,81,91,
2,12,22,32,42,52,62,72,82,92,
3,13,23,33,43,53,63,73,83,93,
4,14,24,34,44,54,64,74,84,94,
5,15,25,35,45,55,65,75,85,95,
6,16,26,36,46,56,66,76,86,96,
7,17,27,37,47,57,67,77,87,97,
8,18,28,38,48,58,68,78,88,98,
9,19,29,39, 49,59,69,79,89,99,
10,20,30,40,50,60,70,80,90,100),nrow=10)
```

Om de machtsverheffing van de bovenstaande matrix te berekenen, tik je in de  $\mathbb{R}$  console het volgende commando in:

```
>print(TenByTen %*% TenByTen %*% TenByTen %*% TenByTen
%*%TenByTen %*% TenByTen %*% TenByTen %*% TenByTen %*%
TenByTen %*% TenByTen/1025,digits=3) in.
```

We delen door  $10^{25}$  omdat elk getal in de uitkomstmatrix hiervan een veelvoud is.


$\mathbb{R}$  komt terug met


	[1,]	[2,]	[3,]	[4,]	[5,]	[6,]	[7,]	[8,]	[9,]	[10,]
[1,]	1.77	1.81	1.85	1.88	1.92	1.96	1.99	2.03	2.07	2.11
[2,]	4.27	4.36	4.45	4.54	4.63	4.72	4.81	4.90	4.99	5.08
[3,]	6.77	6.92	7.06	7.20	7.34	7.48	7.63	7.77	7.91	8.05
[4,]	9.27	9.47	9.66	9.86	10.05	10.25	10.44	10.64	10.83	11.03
[5,]	11.77	12.02	12.27	12.52	12.76	13.01	13.26	13.51	13.75	14.00
[6,]	14.28	14.58	14.88	15.18	15.48	15.78	16.08	16.38	16.68	16.98
[7,]	16.78	17.13	17.48	17.83	18.19	18.54	18.89	19.24	19.60	19.95
[8,]	19.28	19.68	20.09	20.49	20.90	21.30	21.71	22.11	22.52	22.92
[9,]	21.78	22.24	22.69	23.15	23.61	24.07	24.52	24.98	25.44	25.90
[10,]	24.28	24.79	25.30	25.81	26.32	26.83	27.34	27.85	28.36	28.87

Dit is niets anders dan de matrix

$$\begin{pmatrix} 1.77 & 1.81 & 1.85 & 1.88 & 1.92 & 1.96 & 1.99 & 2.03 & 2.07 & 2.11 \\ 4.27 & 4.36 & 4.45 & 4.54 & 4.63 & 4.72 & 4.81 & 4.90 & 4.99 & 5.08 \\ 6.77 & 6.92 & 7.06 & 7.20 & 7.34 & 7.48 & 7.63 & 7.77 & 7.91 & 8.05 \\ 9.27 & 9.47 & 9.66 & 9.86 & 10.05 & 10.25 & 10.44 & 10.64 & 10.83 & 11.03 \\ 11.77 & 12.02 & 12.27 & 12.52 & 12.76 & 13.01 & 13.26 & 13.51 & 13.75 & 14.00 \\ 14.28 & 14.58 & 14.88 & 15.18 & 15.48 & 15.78 & 16.08 & 16.38 & 16.68 & 16.98 \\ 16.78 & 17.13 & 17.48 & 17.83 & 18.19 & 18.54 & 18.89 & 19.24 & 19.60 & 19.95 \\ 19.28 & 19.68 & 20.09 & 20.49 & 20.90 & 21.30 & 21.71 & 22.11 & 22.52 & 22.92 \\ 21.78 & 22.24 & 22.69 & 23.15 & 23.61 & 24.07 & 24.52 & 24.98 & 25.44 & 25.90 \\ 24.28 & 24.79 & 25.30 & 25.81 & 26.32 & 26.83 & 27.34 & 27.85 & 28.36 & 28.87 \end{pmatrix}$$

## 4 Hoe citeren ?

Het commando `>citation()` geeft aan hoe je  citeert.

 komt terug met volgende boodschap :

```
To cite R in publications use:
```

```
R Development Core Team (2010). R: A language and environment for
statistical computing. R Foundation for Statistical Computing,
Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.
```

```
A BibTeX entry for LaTeX users is
```

```
@Manual{,
  title = {R: A Language and Environment for Statistical Computing},
  author = {{R Development Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2010},
  note = {{ISBN} 3-900051-07-0},
  url = {http://www.R-project.org/},
}
```

```
We have invested a lot of time and effort in creating R, please cite it
when using it for data analysis. See also 'citation("pkgname")' for
citing R packages.
```

Figure 3:  citation




## 5 Datastructuren in

### 5.1 Variabelen

Numerieke gegevens bevatten de punt als decimaal teken, niet als aanduiding van duizendtallen. Zo is **54603.6** gelijk aan vierenvijftigduizend zeshonderd en drie komma zes.

Lettergegevens worden omgeven door aanhalingstekens“”.

“**Algemene diensten**” is een alfanumerieke waarde, in mensentaal een tekst-gegeven.

 is een object-georiënteerd programma. Een belangrijk “object” is de **variabele**. Aan een variabele **grootte** kan je een waarde toewijzen. De variabele **grootte** is een object.

Het toewijzen van een numerieke waarde doe je door het commando :


```
>grootte < - 1.80
```

Zoals je ziet werd de toewijzingsoperator < - gebruikt om de waarde 1.80 toe te wijzen aan de variabele **grootte**. In het vervolg zal ik evenwel gewoon het gelijk aan teken = gebruiken.

Vanaf nu kan je het object **grootte** gebruiken in nieuwe berekeningen.

Bijvoorbeeld :

```
>grootte*6
```

 komt terug met

```
>[1] 10.8
```

De naam van variabelen mag geen blanco's bevatten noch komma's.

Je mag de naam van een variabele ook niet omgeven door aanhalingstekens “”.

Het toewijzen van een tekst-waarde doe je door het commando :

```
>weekdag = “Vrijdag”
```

Vanaf nu is het object **weekdag** een variabele en heeft de waarde “Vrijdag”.

Een doordenker : Is “**grootte**” een variabele of een waarde ?

## 5.2 Lijsten

$\mathbb{R}$  kent vele andere gegevenstypes. Een veel gebruikt gegevenstype is de “lijst”. Stel dat je de maanden van het jaar wil toewijzen aan de lijst Jaar. Dat doe je als volgt :

```
>Jaar=c(“Jan”,“Feb”,“Mar”,“Apr”,“Mei”,“Jun”,“Jul”,“Aug”,“Sep”,  
“Oct”,“Nov”,“Dec”)
```

Stel dat je één element, het vijfde bv., uit de lijst wil halen, dan kan dat door het commando:

```
>Jaar[5]
```

$\mathbb{R}$  komt terug met

```
>[1] “Mei”
```

Numerieke lijsten bestaan ook.

```
>Oogst=c(123,456,789,741,852,963,951,627,843,258,369,147)
```

$\mathbb{R}$  beschouwt nu Oogst als een wiskundig object (een lijst), waarmee berekeningen kunnen worden gemaakt.

We wijzen een nieuwe waarde toe aan de variabele **Oogst**:

```
>Oogst = Oogst*545.6+Oogst^2
```

$\mathbb{R}$  komt terug met het resultaat

```
>[1] 82237.8 456729.6 1052999.4 953370.6 1190755.2 1452781.8 1423266.6  
735220.2 1170589.8 207328.8 337487.4 101812.2
```

De bewerking  $>Oogst[6]$  levert als resultaat :

```
>[1] 1452781.8
```

Opgepast !

$\mathbb{R}$  is hoofdlettergevoelig. De variabele Oogst is voor  $\mathbb{R}$  iets anders dan de variabele oogst. Probeer zelf enkele berekeningen in  $\mathbb{R}$  te maken.

### 5.3 Dataframes

Een belangrijk concept in  $\mathbb{R}$  is dat van dataframes. Het is een object met rijen en kolommen, eigenlijk een beetje vergelijkbaar met een matrix. Bij het uitvoeren van onderzoek zal je wellicht vele “observaties” of “metingen” doen van een bepaalde variabele en van elke variabele zal je wellicht verschillende eigenschappen willen bijhouden. Dan is de dataframe de geschikte datastructuur. De observaties of metingen uit je studie/onderzoek zal je terugvinden in de rijen van de dataframe. De “variabelen” of eigenschappen van elke observatie zal je terugvinden in de kolommen van de dataframe.

De vergelijking met een matrix gaat slechts gedeeltelijk op. Een matrix kan immers alleen numerieke gegevens bevatten. Een dataframe kan elk type gegevens bevatten.

Een voorbeeld van een dataframe is :

Veld	Oppervlakte	Vegetatie	PH	Worm dichtheid
Waarbekeveld	30	Grasland	5.2	4
Schoonveld	20.6	Braak	9.5	2
Halleveld	370	Grasland	6.3	5
Geraardsbergenveld	90	Leem	5.2	5
Nieuwenhovenveld	60	Klei	5.2	1

Zo'n dataframe leest als volgt : “het Waarbekeveld is zo'n 30 (ha) groot. De vegetatie is grasland en de PH-waarde van de grond is er blijkbaar 5,2. De dichtheid aan wormen is 4 wormen/vierkante meter.

Bovenstaand dataframe zou men bijvoorbeeld als volgt invoeren in  $\mathbb{R}$  :

```
>Veld = c(“Waarbekeveld”, “Schoonveld”, “Halleveld”, “Geraardsbergenveld”, “Nieuwenhovenveld”)
>Oppervlakte = c(30,20.6, 370,90,60)
>Vegetatie = c(“Grasland”, “Braak”, “Grasland”, “Leem”, “Klei”)
>PH = c(5.2,9.5,6.3,5.2,5.2)
>Worm_dichtheid = c(4,2,5,5,1)
```

Het volgende commando creëert het dataframe :

```
>data_frame = data.frame(Veld,Oppervlakte,Vegetatie,PH,Worm_dichtheid)
```

Tik `data_frame` in om dit te verifiëren.

Wellicht zal je zelden de gegevens van een uitgebreid dataframe rechtstreeks in  $\mathbb{R}$  invoeren. De gegevensinvoer zal vaak gebeuren op het terrein in een rekenblad. Dat rekenblad zal je importeren in  $\mathbb{R}$ . (Zie verder)

## 6 Data invoeren of importeren in $\mathbb{R}$

### 6.1 Gegevens intikken op de $\mathbb{R}$ console.

We hebben reeds één manier gezien om data toe te wijzen aan het object, variabele. Opnieuw een voorbeeld :

```
>A = 1:10
```

Door dit commando wordt aan de variabele met naam A een lijst van getallen toegewezen. Wiskundigen noemen A een vector met lengte 10 en waarde

$$A = \{1,2,3,4,5,6,7,8,9,10\}$$

Op deze vector kunnen nu functies losgelaten worden. Stel dat je vector A wil vermenigvuldigen met 5 en je wil het resultaat hiervan toewijzen aan vector B.

Je schrijft dan eenvoudig :

```
>B=A*5
```

Als je  $\mathbb{R}$  vraagt om B te tonen ( `>B` ) dan komt  $\mathbb{R}$  terug met :

```
>[1] 5 10 15 20 25 30 35 40 45 50
```

Je kan ook waarden toewijzen aan een vector via het toetsenbord. Je gebruikt dan de scan functie.

```
>V = scan()
```

Tik bovenstaand commando in en nadat je op de returntoets hebt gedrukt, wacht  $\mathbb{R}$  op de invoer van gegevens. Je voert een gegeven in en daarna klik je op de returntoets. Je kan zo doorgaan tot wanneer alle gegevens zijn ingevoerd. Om de invoer van gegevens te beëindigen, klik je tweemaal op de returntoets. Hiermee weet  $\mathbb{R}$  dat je wil stoppen met data in te voeren.  $\mathbb{R}$  zal je melden hoeveel gegevens je hebt ingevoerd.

```
>V = scan()
```

```
1: 10
```

```
2: 11
```

```
3: 12
```

```
4: 13
```

```
5: Read 4 items
```

Als je nu de waarde van de variabele V (in de wiskunde noemt men V een vector met lengte 4) opvraagt (dat doe je door `>V` te schrijven), komt  $\mathbb{R}$  terug met : `[1] 10 11 12 13`

Probeer het zelf even uit. Het is nu mogelijk om element 3 van de vector V op te vragen via volgend commando :

```
>V[3]
```

$\mathbb{R}$  komt terug met het resultaat

```
[1] 12
```

## 6.2 Gegevens kopiëren en plakken uit Mirosoft-Excel

Het kan ook anders. Stel dat je gegevens in tabelvorm hebt ingevoerd in een Microsoft-excel rekenblad. Het rekenblad ziet er als volgt uit :

	kolom A	kolom B
rij 1	Verkoop	Aankoop
rij 2	102	98
rij 3	106	85
rij 4	125	142
rij 5	100	88
rij 6	159	144
rij 7	102	78
rij 8	123	120

Table 1: Gegevens uit een rekenblad

Je kan nu de gegevens uit het rekenblad kopiëren. Dat doe je via het “Edit”-menu en het commando “Copy”.

Je kopiëert alleen de gegevens uit kolom A: vanaf 102 tot 123.

In de  $\mathbb{R}$  console tik je na de prompt :

```
>Verkoop = scan()
```

en je drukt op de return-toets.

$\mathbb{R}$  komt terug met

```
> 1:
```

Nu kan je het “plak”-commando uit hetzelfde “Edit”-menu gebruiken en de data uit het rekenblad “Paste” in de  $\mathbb{R}$  console. Je eindigt met tweemaal de return-toets in te tikken.

Om het resultaat te zien, tik je gewoon het woord `>Verkoop` in en  $\mathbb{R}$  komt terug met

```
[1] 102 106 125 100 159 102 123
```

Op deze manier zijn de bovenstaande waarden 102 tot 123 toegewezen aan de variabele `Verkoop`.

Je doet hetzelfde met de tweede kolom “Aankoop”. Deze gegevens (van 98 tot 120) wijs je toe aan de variabele `Aankoop`. Na de invoer zal de variabele `Aankoop` de waarden 98 85 142 88 144 78 120 bevatten.

Via het “data.frame” commando kan je beide variabelen verbinden tot een data.frame.

```
>Data = data.frame(Verkoop,Aankoop)
```

Het commando `>Data` levert volgend resultaat :

	Verkoop	Aankoop
1	102	98
2	106	85
3	125	142
4	100	88
5	159	144
6	102	78
7	123	120

## 6.3 Een bestand inladen in $\mathbb{R}$

### 6.3.1 Een gegevensbestand vanop het www inladen

De bovenstaande “scan()” functie is nogal omslachtig, zeker als het over grote aantallen gegevens gaat. Als je gegevens wil analyseren, dan heb je die wellicht in één of ander bestand zelf verzameld of ontdekt op het wereld wijde web. We nemen aan dat het om een Microsoft Excel-bestand gaat.

Zo'n gegevensbestand kan worden ingeladen in  $\mathbb{R}$ .

#### Het databestand “human\_dev”

Ik stel voor dat we een bestand downloaden via het internet vanuit mijn drop-box folder. Daar bevindt zich een voor iedereen toegankelijke folder die “Public” noemt. Binnen de folder bevindt zich een bestand “human\_development.csv”.

Je kan dit bestand binnen  $\mathbb{R}$  laden door onderstaand commando in te tikken:

```
>human_dev = read.table  
 (“http://dl.dropbox.com/u/2195906/human_development.csv”,  
 header=TRUE,sep=“;”)
```

Dit lange commando verdient enige uitleg.

Het commando “read.table()” heeft binnen de haken enkele “argumenten” nodig. Zoals je kan zien zijn de argumenten gescheiden door “komma's”. Een eerste argument is de naam van het bestand. Deze naam moet omgeven worden door aanhalingstekens“. Dan volgt een komma. Na de komma volgt de aanduiding dat ook de kolomhoofden - dit zijn de namen van de gegevens uit het bestand - moeten worden ingeladen. Dan volgt opnieuw een komma en de aanduiding dat de gegevens in het “csv” bestand gescheiden zijn door punt-komma's. De “separator” is een punt-komma.

Tik het commando zelf in binnen de  $\mathbb{R}$ -console.

De “ ” binnen deze tekst verschillen van de aanhalingstekens binnen  $\mathbb{R}$  en

eenvoudig kopiëren en plakken leidt niet tot het correcte resultaat. Een commandolijn binnen `R` moet in één stuk worden geschreven en mag niet verdeeld worden over verschillende lijnen.

Het bestand bevat gegevens over 39 landen. Over elk land worden 7 gegevens bijgehouden. De gegevens werden in “tabel-vorm” bijgehouden. Het is een tabel van 39 rijen en 7 kolommen.

Bovenaan de rijen staan de zogenaamde “headers” (“kolomhoofden”) : dat zijn de namen van elke kolom en ze worden in het `R` jargon “variabelen” genoemd. In de eerste kolom staat de naam van het land. Het databestand gaf de naam “C1.T” aan deze variabele. In de tweede kolom staan het percentage van de mensen die Internet gebruiken. De naam van deze tweede variabele is “INTERNET”. Kolom drie geeft het Nationaal Inkomen per hoofd van de bevolking weer. De naam van de variabele is “GDP”. Dit Nationaal Inkomen per hoofd is uitgedrukt in duizend US\$. Kolom vier geeft de  $CO_2$ -uitstoot per hoofd weer van elk land. De eenheid hiervoor is kubieke meter. De naam van de variabele is “CO2”. Kolom vijf heeft als naam “CELLULAR” en geeft het percentage aan volwassenen weer die een mobiele telefoon bezitten in het land. Kolom zes geeft aan hoeveel kinderen per volwassen vrouw er zijn in het land en de variabele noemt “FERTILITY”. De laatste kolom geeft de scholingsgraad aan in % voor elk betrokken land en de variabele noemt “LITTERACY”.

Het commando “`read.table()`” van hierboven laadt het databestand “human\_dev” in `R`. Eenmaal ingeladen is kent `R` alleen de totaliteit van de dataframe. De variabelen “C1.T” tot “LITTEARCY” zijn voor `R` nog onbekend.

Het commando `>human_dev` toont inderdaad het ganse databestand. Als we de gegevens van één variabele willen zien, moeten we volgend commando gebruiken:

```
>human_dev$INTERNET
```

Dit is vrij omslachtig. De naam van de variabele moet immers telkens worden voorafgegaan door de naam van het databestand en het dollarteken. Om dit te vermijden bestaat het commando `attach`.

```
>attach(human_dev)
```

Vanaf nu kan je elke variabele opvragen door eenvoudig de naam van de variabele in te tikken.

```
>INTERNET
```

Hieronder vind je het “human\_dev” bestand.

C1-T	INTERNET	GDP	CO2	CELLULAR	FERTILITY	LITERACY
Algeria	0.65	6.09	3	0.3	2.8	58.3
Argentina	10.08	11.32	3.8	19.3	2.4	96.9
Australia	37.14	25.37	18.2	57.4	1.7	100
Austria	38.7	26.73	7.6	81.7	1.3	100
Belgium	31.04	25.52	10.2	74.7	1.7	100
Brazil	4.66	7.36	1.8	16.7	2.2	87.2
Canada	46.66	27.13	14.4	36.2	1.5	100
Chile	20.14	9.19	4.2	34.2	2.4	95.7
China	2.57	4.02	2.3	11	1.8	78.7
Denmark	42.95	29	9.3	74	1.8	100
Egypt	0.93	3.52	2	4.3	3.3	44.8
Finland	43.03	24.43	11.3	80.4	1.7	100
France	26.38	23.99	6.1	60.5	1.9	100
Germany	37.36	25.35	9.7	68.2	1.4	100
Greece	13.21	17.44	8.2	75.1	1.3	96.1
India	0.68	2.84	1.1	0.6	3	46.4
Iran	1.56	6	4.8	3.2	2.3	70.2
Ireland	23.31	32.41	10.8	77.4	1.9	100
Israel	27.66	19.79	10	90.7	2.7	93.1
Japan	38.42	25.13	9.1	58.8	1.3	100
Malaysia	27.31	8.75	5.4	31.4	2.9	84
Mexico	3.62	8.43	3.9	21.7	2.5	89.5
Netherlands	49.05	27.19	8.5	76.7	1.7	100
New Zealand	46.12	19.16	8.1	59.9	2	2
Nigeria	0.1	0.85	0.3	0.3	5.4	57.7
Norway	46.38	29.62	8.7	81.5	1.8	100
Pakistan	0.34	1.89	0.7	0.6	5.1	28.8
Philippines	2.56	3.84	1	15	3.2	95
Russia	2.93	7.1	9.8	5.3	1.1	99.4
Saudi Arabia	1.34	13.33	11.7	11.3	4.5	4.5
South Africa	6.49	11.29	7.9	24.2	2.6	2.6
Spain	18.27	20.15	6.8	73.4	1.2	96.9
Sweden	51.63	24.18	5.3	79	1.6	100
Switzerland	30.7	28.1	5.7	72.8	1.4	100
Turkey	6.04	5.89	3.1	29.5	2.4	77.2
United Kingdom	32.96	24.16	9.2	77	1.6	1.6
United States	50.15	34.32	19.7	45.1	2.1	2.1
Vietnam	1.24	2.07	0.6	1.5	2.3	90.9
Yemen	0.09	0.79	1.1	0.8	7	26.9



## Het databestand “weil”

Dit bestand is een voorbeeld van een goed uitgebouwde dataset met vele variabelen en vele observaties. We downloaden een bestand met gegevens over 211 landen van de wereld. Per land worden 114 gegevens bijgehouden. Dat bestand staat op een server op het wereld wijde web. Het bestand heeft het formaat “csv” of “comma separated file”.

Zoals reeds gezien, worden deze bestanden in  $\mathbb{R}$  ingeladen via het commando “read.table()”.

De in te lezen gegevens zullen we toewijzen aan een object dat we **weil** zullen noemen.

```
>weil = read.table(“http://wps.aw.com/wps/media/objects/5321/5448761/weil_labdata_2e_final.csv”, header=TRUE, sep=”,”)

```

Als je het bovenstaande commando uitvoert, zie je op het eerste zicht niets. Toch heeft  $\mathbb{R}$  de gegevens ingeladen.

Nadat het bestand is ingeladen, wordt het ”gelinkt” aan  $\mathbb{R}$  met het commando **>attach(human\_dev)**

Om de gegevens zichtbaar te maken, gebruiken we het commando **>weil**. Er vliegt heel wat data over je scherm en je wordt wellicht niet veel wijzer. Wat je ziet, is het laatste stuk van die enorme hoeveelheid gegevens. Als je naar boven “scrollt”, kan je de rest van de gegevens zien.

Hoe stellen we ons deze gegevens nu voor ?

Stel je voor dat de gegevens in een enorme matrix van 211 rijen en 114 kolommen zitten. Elke rij is een ander land en voor elk land werden 114 gegevens opgenomen : die vind je in de 114 kolommen. Elke kolom heeft een naam. Het commando “names()” laat toe die namen zichtbaar te maken.

```
>names(weil)

```

Je krijgt een scherm, waarvan ik hier slechts de eerste drie en de laatste drie lijnen weergeef:

```
[1,]      "country"      "ccode"      "prodgrowth"
[4,]      "agedep05"    "pop70"     "tradetax05"
[7,]      "colledlabforce2000" "percemployagr05" "lifeex70"
.....
.....
.....
[106,]    "topincsh03"   "botincsh03"   "thrift"
[109,]    "obedience"  "perseverance" "faith"
[112,]    "technologyvstradition" "avgyrsfemaleed" "Number.of.Missing.Values"
```

Deze namen zijn de kolomhoofden uit het bestand. In het  $\mathbb{R}$ -jargon noemen deze kolomhoofden “variabelen”. In het bestand, dat we vanaf nu “weil” zullen noemen, zijn dus 211 maal 114 gegevens opgenomen.

De eerste variabele heeft als naam “country” en bevat inderdaad 211 landennamen. De tweede variabele noemt “ccode”, countrycode en geeft voor elk land de drieletter afkorting van de naam. En zo gaat dit door tot de laatste variabele die “Number.of.Missing.Values” noemt.

Om de grootte van het weil bestand te ontdekken gebruiken we volgende commando’s.

```
>nrow(weil)
>ncol(weil)
```



$\mathbb{R}$  geeft inderdaad de getallen 211 en 114.

Het  $\mathbb{R}$  programma laat je toe om bewerkingen uit te voeren op elk van deze 211 maal 114 gegevens. Om de grote hoeveelheid gegevens van het weil bestand meer overzichtelijk te maken, bestaat het commando `>fix(weil)`

Je bekomt een nieuw scherm met daarop een “rekenblad”-achtig beeld van de weil gegevens. Je kan dat scherm verlaten door het venster linksbovenaan te sluiten. Je komt op die manier terug in de  $\mathbb{R}$ -console.

We weten ondertussen dat het object “weil” een “dataframe” genoemd wordt in het  $\mathbb{R}$ -jargon.

### 6.3.2 Bestanden en pakketten die in R zelf ingebouwd zitten

 heeft een aantal pakketten ingebouwd. Het zijn deze pakketten die de mogelijkheden van  bepalen. Om te achterhalen welke pakketten op een bepaald moment actief zijn bestaat het commando “library()” :

```
>library()
```


Na het intikken van de returntoets komt  terug met een nieuw scherm :

```
Packages in library '/Users/Papa/Library/R/2.12/library':
```

akima	Interpolation of irregularly spaced data
bitops	Functions for Bitwise operations
caTools	Tools moving window statistics, GIF, Base64, ROC AUC, etc.
chron	Chronological objects which can handle dates and times
gdata	Various R programming tools for data manipulation
gplots	Various R programming tools for plotting data
gtools	Various R programming tools
HSAUR2	A Handbook of Statistical Analyses Using R (2nd Edition)
ineq	Measuring Inequality, Concentration, and Poverty
lme4	Linear mixed-effects models using S4 classes
MASS	Support Functions and Datasets for Venables and Ripley's MASS
nlme	Linear and Nonlinear Mixed Effects Models
odesolve	Solvers for Ordinary Differential Equations
pwt	Penn World Table
scatterplot3d	3D Scatter Plot
tree	Classification and regression trees

Packages in library '/Library/Frameworks/R.framework/Resources/library':

akima	Interpolation of irregularly spaced data
base	The R Base Package
boot	Bootstrap R (S-Plus) Functions (Canty)
class	Functions for Classification
cluster	Cluster Analysis Extended Rousseeuw et al
codetools	Code Analysis Tools for R
colorspace	Color Space Manipulation
datasets	The R Datasets Package
digest	Create cryptographic hash digests of R objects
foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
ggplot2	An implementation of the Grammar of Graphics
graphics	The R Graphics Package
grDevices	The R Graphics Devices and Support for Colours and Fonts
grid	The Grid Graphics Package
grImport	Importing Vector Graphics
Hmisc	Harrell Miscellaneous
ISwR	Introductory Statistics with R
iterators	Iterator construct for R
itertools	Iterator Tools
KernSmooth	Functions for kernel smoothing for Wand & Jones (1995)
lattice	Lattice Graphics
MASS	Support Functions and Datasets for Venables and Ripley's MASS
Matrix	Sparse and Dense Matrix Classes and Methods
methods	Formal Methods and Classes
mgcv	GAMs with GCV/AIC/REML smoothness estimation and GAMMs by PQL
nlme	Linear and Nonlinear Mixed Effects Models
nnet	Feed-forward Neural Networks and Multinomial Log-Linear Models
plyr	Tools for splitting, applying and combining datas
proto	Prototype object-based programming
pwt	Penn World Table
random	True random numbers using random.org
reshape	Flexibly reshape data.
rpart	Recursive Partitioning
sm	Smoothing methods for nonparametric regression and density estimation
spatial	Functions for Kriging and Point Pattern Analysis
splines	Regression Spline Functions and Classes
stats	The R Stats Package
stats4	Statistical Functions using S4 Classes
survival	Survival analysis, including penalised likelihood
tcltk	Tcl/Tk Interface
tools	Tools for Package Development
UsingR	Data sets for the text "Using R for Introductory Statistics"
utils	The R Utils Package
vioplot	Violin plot
XML	Tools for parsing and generating XML within R and S-Plus


Elk van deze pakketten voegt nieuwe functies en/of "datasets" toe aan .

De hierboven aangehaalde pakketten bevatten verschillende databestanden.

Het commando `>data(package = .packages(all.available = TRUE))` brengt je naar een venster dat alle bestandsnamen weergeeft van de databestanden die je op een bepaald ogenblik kan gebruiken.

Hieronder vind je een extract van de lijst van datasets die je kan gebruiken. Voer het commando zelf uit om de volledige lijst te ontdekken.

Data sets in package 'Boot'	
acme	Monthly Excess Returns
aids	Delay in AIDS Reporting in England and Wales
aircondit	Failures of Air-conditioning Equipment
aircondit7	Failures of Air-conditioning Equipment
amis	Car Speeding and Warning Signs
aml	Remission Times for Acute Myelogenous Leukaemia
beaver	Beaver Body Temperature Data
bigcity	Population of U.S. Cities
brambles	Spatial Location of Bramble Canes
breslow	Smoking Deaths Among Doctors
calcium	Calcium Uptake Data
cane	Sugar-cane Disease Data
capability	Simulated Manufacturing Process Data
catsM	Weight Data for Domestic Cats
cav	Position of Muscle Caveolae
cd4	CD4 Counts for HIV-Positive Patients
cd4.nested	Nested Bootstrap of cd4 data
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
rats	Rat data from Gail et al.
stanford2	More Stanford Heart Transplant data
survexp.mn (survexp)	Census Data Sets for the Expected Survival and Person Years Functions
survexp.us (survexp)	Census Data Sets for the Expected Survival and Person Years Functions
survexp.usr (survexp)	Census Data Sets for the Expected Survival and Person Years Functions
tobin	Tobin's Tobit data
veteran	Veterans' Administration Lung Cancer study
Data sets in package 'TTR'	
ttrc	Technical Trading Rule Composite data
Data sets in package 'xts'	
sample.matrix	Sample Data Matrix For xts Example and Unit Testing

Vooraleer toegang te krijgen tot een welbepaald databestand, moet de bibliotheek, waartoe het bestand behoort, in  worden opgeladen. Om bijvoor-

beeld het databestand **BCG** te kunnen gebruiken, moet je eerst de bibliotheek “HSAUR” inladen. Dat doe je door het commando :

```
>library(HSAUR)
```

Daarna kan je gewoon **>BCG** intikken en het bestand verschijnt op de **R**-console.

Laten we willekeurig één een ander bestand uit de lijst pikken : **mtcars**.

We sluiten dit venster.

We zijn nu terug in **R** - console. Als we het commando **>mtcars** intikken en op de returntoets drukken, komt volgend databestand op het scherm.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.9	2620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.9	2875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5250	17.98	0	0	3	4
Lincoln Continenta	10.4	8	460.0	215	3.00	5424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3520	16.87	0	0	3	2
AMC Javelin	15.02	8	304.0	150	3.15	3435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.00	2780	18.60	1	1	4	2

“Mtcars” is een bestand met gegevens over 32 wagens.

Het databestand (in het Engels “dataset” genoemd) is opgebouwd als een tabel met rijen en kolommen. (Een dataframe). Elke rij is een nieuw “record” en de kolommen zijn eigenschappen van dat record. Zo heeft “mtcars” 11 kolommen en geeft dus 11 eigenschappen per wagen. Om de structuur te achterhalen van een databestand heeft **R** de “help” functie voorzien. Het commando **help(mtcars)** geeft hierover onmiddellijk meer info. Zo zien we dat de vijfde kolom van het bestand, nl. “hp”, het brutovermogen van elke wagen geeft. Het bestand is immers een lijst van 32 rijen en 11 kolommen.

Opmerking: bovenstaand reken-achtig beeld van het bestand “mtcars” kan de indruk wekken dat het bestand “mtcars” 12 variabelen (= kolommen) heeft. De naam van elke wagen is evenwel GEEN variabele. Deze dataframe heeft gewoon rijnamen in de plaats van rijnummers.

### 6.3.3 Het databestand “pwt7.0” uit het pakket “pwt”


We zien in de bibliotheek “pwt” een bestand zitten dat pwt7.0 heet. Dit bestand wordt in het Engels “the Penn World Tables” genoemd. Zie “[www.ludopoelaert.be](http://www.ludopoelaert.be)” voor meer uitleg over deze Penn Tables.

Dit pakket wordt als volgt geïnstalleerd :

```
>install.packages("pwt", repo="http://R.research.att.com")
```

Het “Center for International Comparisons of Production, Income and Prices at the University of Pennsylvania” is de auteur van dit interessante pakket. Het onderliggende databestand bevat 467400 gegevens over de landen van de wereld. De informatie betreffende de landen is daarenboven beschikbaar van 1950 tot 2009. Dit laat toe om evoluties in de tijd te bestuderen voor de verschillende variabelen die in het databestand “pwt7.0” zijn opgenomen.

Om het pakket te gebruiken, voeren we het commando `>library(pwt)` in.


Zoals hierboven gezien, klikken we het “pwt”-pakket vast in  door het commando `>attach(pwt7.0)`

Om een overzicht te krijgen van het pakket kunnen we weer het commando `>fix(pwt7.0)` gebruiken. Je verlaat dit scherm door het venster te sluiten.

Je kan nu zelf “spelen” met het pakket en bijvoorbeeld een grafiek maken van de evolutie van het Nationaal Inkomen van België sinds 1950. Je gebruikt hiervoor het commando “plot” en de variabelen “country” en “rgdpch” uit het databestand.

### 6.3.4 Het databestand “Forbes2000” uit het pakket “HSAUR”

Laten we eens bekijken welke de top 2000 grootste bedrijven in de wereld zijn. Op de website van Forbes vinden we deze lijst. Ga hiervoor naar de volgende link : “Forbes2000 list ”. De lijst is vrij recent van april 2011!


Bedoeling is om deze lijst nu in onze  console te krijgen.

Hiervoor moeten we uiteraard onze hand kunnen leggen op het onderliggende bestand.

Het databestand “Forbes2000” zit binnen het pakket HSAUR. Dit pakket gaan we nu installeren. We doen dit door het commando :


```
>install.packages("HSAUR", repo="http://R.research.att.com")
```

Nu hebben we toegang tot het databestand “Forbes2000”, door het commando:  
>**Forbes2000**

Om het effectief te kunnen gebruiken, linken we het databestand aan  door het commando:  
>**attach(Forbes2000)**

Om een idee te krijgen van het databestand, tikken we volgend commando in:  
>**fix(Forbes2000)**

We voeren het volgende commando uit om de names van de “headers” te kennen:  
>**names(Forbes2000)**

 komt terug met


```
[1] "rank" "name" "country" "category" "sales" "profits" "assets" "marketvalue"
```

Het aantal kolommen en rijen leren we met:

```
>ncol(Forbes2000)  
>nrow(Forbes2000)
```

Het verwondert ons niet dat er 2000 rijen zijn : dit zijn de top 2000 bedrijven in de wereld volgens Forbes. Er zijn 8 kolommen en dit betekent dat er 8 eigenschappen worden bijgehouden van elk bedrijf. Een interessant commando is het “structure” commando :

```
>str(Forbes2000)
```

 komt terug met

```
'data.frame': 2000 obs. of 8 variables:
```

```
$ rank : int 1 2 3 4 5 6 7 8 9 10 ...  
$ name : chr "Citigroup" "General Electric" "American Intl Group" "ExxonMobil" ...  
$ country : Factor w/ 61 levels "Africa","Australia", ... : 60 60 60 60 56 60 56 28 60  
60 ...  
$ category : Factor w/ 27 levels "Aerospace & defense",... : 2 6 16 19 19 2 2 8 9 20 ...  
$ sales : num 94.7 134.2 76.7 222.9 232.6 ...  
$ profits : num 17.85 15.59 6.46 20.96 10.27 ...  
$ assets : num 1264 627 648 167 178 ...  
$ marketvalue: num 255 329 195 277 174 ...
```



We leren dat het “object” Forbes2000 “klasse” dataframe heeft. Van elke observatie (er zijn 2000 observaties, lees 2000 bedrijven!) worden 8 variabelen bijgehouden.

De “rank” is de rang die het bedrijf krijgt. De “name” is uiteraard de naam van het bedrijf. De variabele “country” is het land waarin het bedrijf ligt. De variabele “categorie” beschrijft de sector/activiteit waartoe het bedrijf behoort. De variabele “sales” geeft de globale omzet van het bedrijf in miljard US\$. De variabele “profits” geeft de winst die het bedrijf maakt eveneens uitgedrukt in miljard US\$. De variabele “assets” geeft de totale activa weer die het bedrijf bezit. De laatste kolom (lees variabele) geeft de marktwaarde weer van het bedrijf in miljard US\$.

Zoals je kan zien uit bovenstaande beschrijving, heeft elke variabele een bepaalde “klasse”. Zo heeft de variabele “rank” klasse “int” : dit wil zeggen dat “rank” wordt weergegeven door gehele getallen. “sales”, “profits”, “assets” en “market-value” zijn dan weer van het type “numeric”, en dat wil zeggen reële getallen. Dit zijn getallen met mogelijk cijfers na de komma.

De variabelen “country” en “category” hebben klasse “Factor”. Het zijn variabelen met wel vooraf bepaalde waarden. Zo heeft de variabele “category” 27 mogelijke waarden.

```
>levels(Forbes2000[,"category"])
```

🔍 komt terug met

```
[1] "Aerospace & defense" "Banking" "Business services & supplies" "Capital goods"  
[5] "Chemicals" "Conglomerates" "Construction" "Consumer durables"  
[9] "Diversified financials" "Drugs & biotechnology" "Food drink & tobacco" "Food markets"  
[13] "Health care equipment & services" "Hotels restaurants & leisure" "Household & personal  
products" "Insurance"  
[17] "Materials" "Media" "Oil & gas operations" "Retailing"  
[21] "Semiconductors" "Software & services" "Technology hardware & equipment" "Telecom-  
munications services"  
[25] "Trading companies" "Transportation" "Utilities"
```



Er zijn nog andere “beschrijvende” commando’s die alle een overzicht geven van het databestand “Forbes2000”.

Probeer ze zelf even uit.


```
>class(Forbes2000)
```

```
>dim(Forbes2000)
```

### 6.3.5 Bestanden vanop je computer inladen in R

Men kan ook bestanden vanaf de harde schijf van de computer rechtstreeks in  inladen. Het is waakzaam om het “pad” aan te geven waar het bestand zich bevindt. Het commando “getwd()” geeft aan in welke folder (“directory”)  het bestand zal zoeken.

```
>getwd()
```

 komt terug met

```
[1] ”/”
```

Dit zal je moeten aanpassen zodat  wel degelijk weet waar jouw bestand zich bevindt.

We nemen even aan dat de folder, waar het bestand “human\_dev.csv” zich bevindt, als naam “R files” heeft. Deze folder bevindt zich in een folder die “Dropbox” noemt. De folder “Dropbox” bevindt zich in een folder “Papa”. Folder “Papa” zit dan weer in de folder “Users”. Daarboven bevindt zich de hoofdfolder die wordt aangeduid door “./”. Het pad naar het bestand “human\_dev” wordt dan als volgt aangeduid:

```
>setwd("./Users/Papa/Dropbox/R files")
```

Merk het puntje op vóór de “/” !

Met het volgende commando kan je alle bestanden zien die zich in de aangeduide folder bevinden :




```
>dir()
```


Als alles goed is, moet jouw bestand “human\_dev.csv” voorkomen in de getoonde lijst.

Met het commando “read.table()” kan je dan het bestand inlezen, en moet je het pad niet meer herhalen. Het volstaat om te schrijven :

```
>human_dev = read.table(“human_dev.csv”, header=TRUE, sep=”,”)
```


## 7 De tekst-editor van


Binnen het programma  kan je ook scripts maken. Een script is een stukje R-code. R-code is een opeenvolging van R-commando's. Stel dat je een hele reeks commando's na elkaar wil geven, dan is het gemakkelijk deze op te nemen in een script. De menu-bar van het programma  geeft toegang tot de tekst-editor. Je gaat naar het "file" menu. Bij Windows kies je "New script", bij MacOSX kies je "New document". Meteen kom je in een editor die jouw  commando's kan opnemen. Stel dat we een berekening willen maken van de BMI, body mass index, van vier personen.


We gaan naar de tekst-editor binnen .

Tik in de editor het volgende in :

```
options(digits=3)  
grootte=c(1.7,1.75,1.8,1.85)  
gewicht=c(70,75,80,85)  
bmi=gewicht/grootte^2  
bmi
```

Let op! In de tekst-editor gebruik je geen -prompt !  
Je kan dit script nu uitvoeren. In Windows ga je naar het menu "Edit" en kies je "Run All". Bij de Mac kies je in het menu "Edit" de optie "Execute". (Ik neem eventjes aan dat je een Engelstalig beheerssysteem hebt.)

 zal nu het ganse script uitvoeren.



 komt terug met de bmi van de vier personen.  
>[1] 24.2 24.5 24.7 24.8

Het kan nuttig zijn om commentaarlijnen toe te voegen aan een script. Dan weet je later nog wat je bedoelde.  
Commentaar wordt voorafgegaan door het hash-teken "#". Het script ziet er al veel leesbaarder uit als volgt :

```
#We beperken ons tot 3 significante cijfers  
options(digits=3)  
#We voeren de grootte in van 4 personen. Het formaat is in meter  
en centimeters  
grootte=c(1.70,1.75,1.80,1.85)  
#Voer het gewicht in van de vier personen (in kg)  
gewicht=c(70,75,80,85)  
bmi=gewicht/grootte^2  
bmi
```

De tekst-editor laat toe om tekstbestanden (.txt) te openen, te wijzigen en te saven. Zo kan je dit script saven onder een betekenisvolle naam en kan je dit script later terug oproepen. Je hoeft dan de code niet opnieuw in te tikken.

## 8 gebruiken vanop het internet



Je kan  ook gebruiken zonder dat je het programma hoeft te installeren. Je moet wel toegang hebben via een computer, tablet of smartphone tot het wereldwijde web. Ga hiervoor naar de website “  from the cloud ”

Op deze pagina ga je naar “JavaScript Version of Rweb”.

Opgepast : het gebeurt dat deze webserver niet online is.

## 9 Wiskundige en statistische bewerkingen uitvoeren op databestanden

### 9.1 Berekeningen op het bestand “human\_dev”



Eenmaal ingeladen in  kunnen we bewerkingen uitvoeren op de data. We verzekeren er ons van dat de data echt gelinkt zit in  door het commando `>attach(human_dev)` uit te voeren. We kunnen nu elk gegeven uit het bestand apart manipuleren. We weten inmiddels dat een structuur in tabelvorm, zoals deze van human\_dev, een “data.frame” wordt genoemd.

Stel dat we alleen de gegevens van het Nationaal Inkomen per hoofd willen zien. Het commando `>GDP` verwezenlijkt dit.

Stel dat we het gemiddelde percentage van scholingsgraad willen berekenen van de 39 landen. Tik hiervoor het volgende commando in :

```
>mean(LITERACY)
```


De gemiddelde scholingsgraad voor de 39 landen bedraagt 86,3%

Opgepast : zoals je ziet zijn sommige letters in hoofdletters geschreven. De kolomhoofden staan in het bestand “human\_dev” inderdaad in hoofdletters. Dat was een keuze van de maker van het bestand. In  moet je deze keuze respecteren. De commando's in  zelf worden in kleine letters geschreven.

Het Nationaal Inkomen per hoofd in Russia is dan weer :

```
>human_dev$GDP[C1.T == "Russia"]
```

en is gelijk aan 7.100 US\$ per persoon.

Probeer dit zelf in . Het laatste commando lijkt op het eerste zicht ingewikkeld. Tussen de vierkante haken staat een logische uitdrukking “C1.T==”Russia””. We zoeken het GDP (per capita) voor het land met naam “Russia”. Het kolomhoofd “C1.T” is de naam van de kolom met alle landennamen. We zochten een welbepaald Nationaal Inkomen per hoofd, nl. dat van “Russia”.

Stel dat je een grafiek wil die het verband weergeeft tussen Nationaal Inkomen per hoofd van de bevolking en het Internetgebruik in het land. Dat kan door het “plot” - commando.

```
>plot(human_dev$INTERNET~human_dev$GDP)
```

Het symbool ~ is de tilde

Dit geeft volgende grafiek:

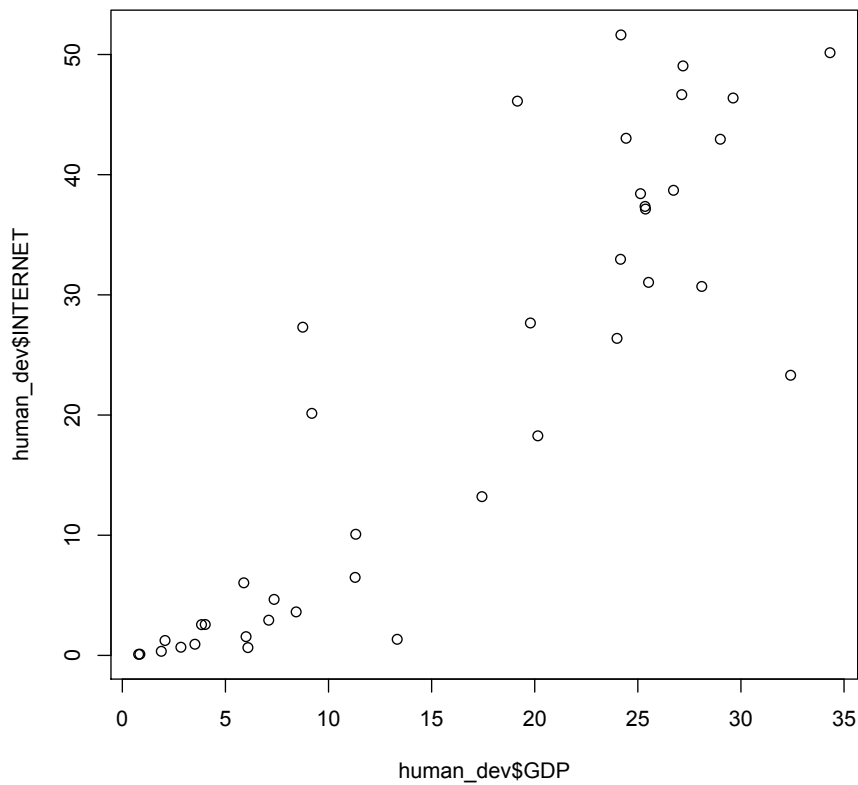



Figure 4: Verband tussen Nationaal Inkomen per hoofd en Gebruik van Internet

Zoals je ziet zijn beide assen nogal “raar” benoemd.  laat je toe elke detail van een grafiek in handen te nemen. Als je de “syntax” van het plot-commando niet kent, volstaat het om `> help(plot)` in te tikken.

Het volgende commando zal de namen van de assen wijzigen in verstaanbare

namen en zal de holle bolletjes veranderen in volle bolletjes. Tevens wordt een titel aan de grafiek gegeven.

```
>plot(human_dev$INTERNET~human_dev$GDP, xlab="Nationaal Inkomen per hoofd",ylab="Internet Gebruik",main="Verband tussen NI/hoofd en Internet gebruik",pch=16)
```

Dit geeft volgende grafiek:

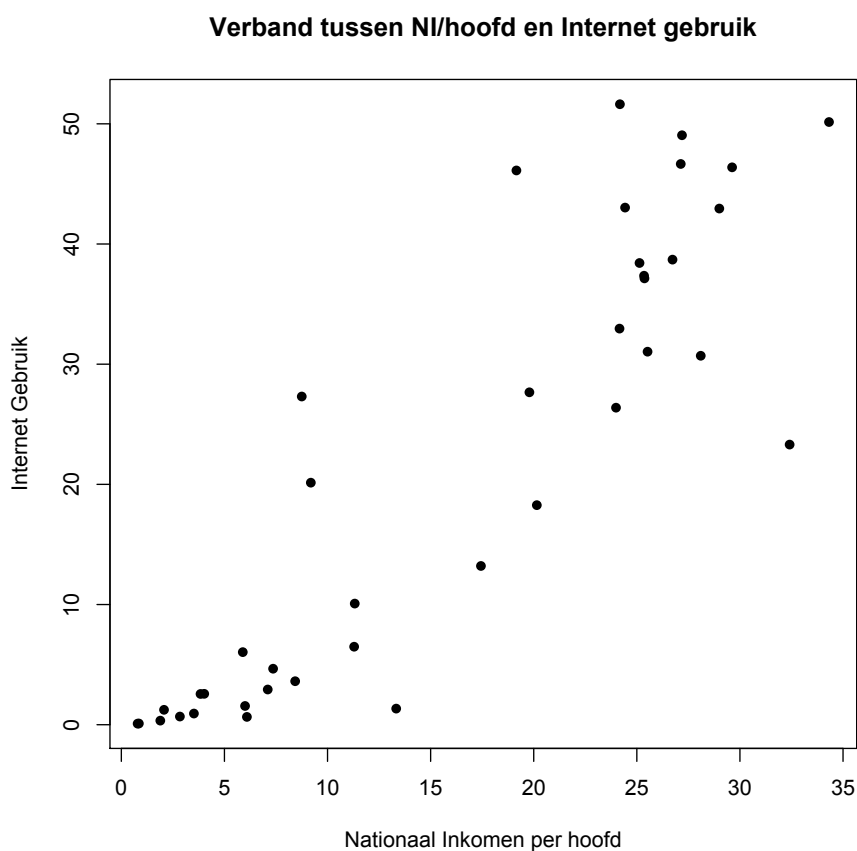



Figure 5: Verband tussen Nationaal Inkomen per hoofd en Gebruik van Internet

We gebruiken data-analyse programma's met statistische verwerkings-mogelijkheden om verbanden te zoeken tussen variabelen uit de data. Laten we een lineaire regressie toepassen op onze "human\_dev" data.frame.  bevat zeer krachtige statistische modellen en wij gaan nu de eenvoudigste gebruiken. We zoeken een verband tussen enerzijds het Nationaal Inkomen per hoofd en het gebruik van Internet. Uit vorige grafieken kon worden afgeleid dat er een sterke "correlatie" bestaat tussen beide "variabelen".

Hoe groter het Nationaal Inkomen per hoofd, hoe groter het gebruik van internet. Dit is by the way ergens logisch. Lees wel dat er een verband (“correlatie”) bestaat, dit is niet noodzakelijk een causaal verband !!

We zullen een lineaire regressie toepassen op onze gegevens uit “human\_dev”. Dat wil zeggend dat we een verband van volgende aard zoeken :

```
>lm(human_dev$INTERNET~human_dev$GDP)
```

🔍 komt terug met de regressie-coëfficiënten :

```
Coefficients:
(Intercept)  human_dev$INTERNET
5.2193      0.5097
```

Uit deze resultaten kunnen we besluiten dat :

Internet\_gebruik = 0,5 maal Nationaal\_Inkomen\_per\_hoofd + 5,2

Als we gaan kijken naar onze data en we nemen de gegevens voor Argentinië, dan is het Nationaal Inkomen per hoofd gelijk aan 11.320 US\$ en het percentage internetgebruik gelijk aan 10,08%. We vergelijken dit even met de lineaire regressierechte die we hierboven hebben bekomen :

Internet\_gebruik = 0.5 maal 11,32 + 5,2 = 5,66 + 5,2 = 10,86 en ligt aardig in de buurt van de 10,08%.

We bekijken ook de sterkte van de “correlatie” via de correlatie-coëfficiënt :

We gebruiken hiervoor het “cor” commando :

```
>cor(human_dev$INTERNET,human_dev$GDP)
```

🔍 komt terug met de correlatie-coëfficiënt R van 0,888. Dat wijst op een sterke correlatie tussen beide variabelen in dit gegevensbestand. We hernemen de vorige grafiek en tekenen er nu de regressie-rechte bij. We realiseren dit met behulp van het commando : “abline”.

```
>plot(human_dev$INTERNET~human_dev$GDP, xlab="Nationaal
Inkomen per hoofd",ylab="Internet Gebruik",main="Verband tussen
NI/hoofd en Internet gebruik",pch=16)
```

```
>abline(lm(human_dev$GDP~human_dev$INTERNET))
```

We zien dat de regressierechte inderdaad de “trend” van het verband tussen de variabelen Nationaal Inkomen per hoofd en Internetgebruik aangeeft. Dit geeft volgende grafiek:

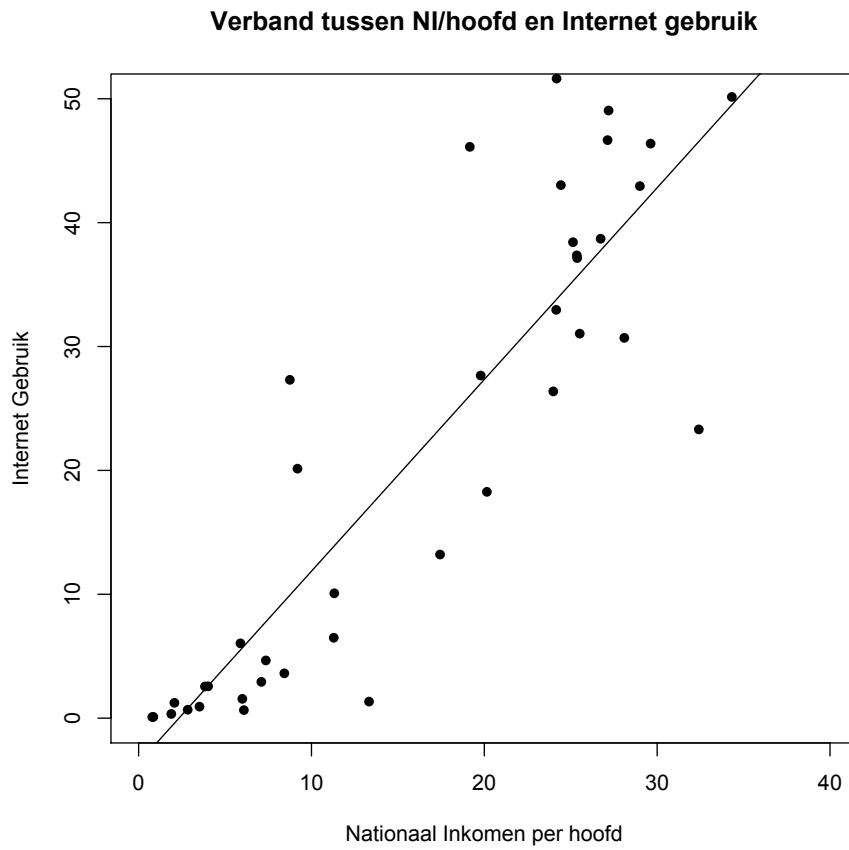


Figure 6: Verband tussen Nationaal Inkomen per hoofd en Gebruik van Internet en de regressierechte




## 9.2 Bewerkingen en statistische berekeningen op het databestand “weil”

Het databestand “Weil” bevat gegevens over 211 landen. Over elk land zijn zo’n 114 gegevens bijgehouden. In en vroegere paragraaf werd aangegeven welke deze zijn.

Het commando `>names(weil)` gaf aan welke die “variabelen” zijn die in het bestand over elk land werden bijgehouden.

We gaan ervan uit dat het command `>attach(weil)` is uitgevoerd.

 heeft een commando dat voor elke variabele een reeks eigenschappen weergeeft. Probeer dit even uit : `>summary(weil)`

Stel dat je 1 kolom uit het “weil” bestand wil gaan selecteren. Bijvoorbeeld de kolom die het Nationaal Inkomen per hoofd weergeeft voor het jaar 1960. Dit doe je als volgt :

```
>weil[,25]
```

Stel dat je het Nationaal Inkomen per hoofd voor het jaar 1960 wil geven voor elk land uit het “weil” bestand.

```
>weil[c(“country”,“rgdpch1960”)]
```

Stel dat je de gemiddelde jaarlijkse groei van een bepaald land wil kennen. Het is de variabele “gy7005” die dit gegeven bijhoudt. We nemen als land België

```
>gy7005[c(country==“Belgium”)]
```

We kunnen ons afvragen welke landen in het jaar 2000 een Nationaal Inkomen per hoofd hebben dat groter is dan 8000 US\$  
De list wordt gegeven door het volgende commando

```
>country[rgdpch2000 > 8000]
```

Zoals je ziet staan er in de lijst enkele <NA> symbolen. Dat zijn blijkbaar landen waarvoor het gegeven “rgdpch2000” niet is ingevuld in de dataset. Het volgende commando is daarom wellicht iets eleganter:

```
>sort(country[rgdpch2000 > 8000])
```

Soms kan het nuttig zijn om die landen te vinden waarvan het Nationaal Inkomen per hoofd in bijvoorbeeld het jaar 1970 groter is dan het Nationaal Inkomen per hoofd van een ander land, bijvoorbeeld China. Volgend commando geeft het resultaat:

```
>sort(country[rgdpch1970 > rgdpch1970[country ==“China”]])
```

Dat zijn blijkbaar 148 landen!

Stel dat we die landen willen vinden die een Nationaal Inkomen per hoofd hadden in 1960 dat groter was dan dat van België in 1960

```
>sort(country[rgdpch1960 > rgdpch1960[country ==“Belgium”]])
```

Het antwoord is een lijst met de volgende 14 landen :

[1] Australia Austria Canada Denmark France Iceland Luxembourg Netherlands  
New Zealand Norway Sweden [12] Switzerland United Kingdom United States

We stellen dezelfde vraag voor het jaar 2000

```
>sort(country[rgdpch2000 > rgdpch2000[country ==“Belgium”]])
```

We bekomen dan het volgend antwoord :

[1] Australia Austria Bermuda Brunei Canada Denmark France Germany [9]  
Hong Kong, China Iceland Ireland Kuwait Luxembourg Netherlands Norway  
Qatar [17] Singapore Sweden Switzerland United Arab Emirates United King-  
dom United States

We stellen vast dat het aantal landen, dat een Nationaal Inkomen per hoofd heeft dat groter is dan dat van België, in 2000 gegroeid is tot 22 landen.

Om twee variabelen uit een databestand naast elkaar op te lijsten, volstaat volgend commando :

```
>weil[c(“country”,“rgdpch1960”)]
```

Je kan deze lijst ook nog sorteren volgens bijvoorbeeld het Nationaal Inkomen per hoofd in 1960 :

```
>weil[order(rgdpch2000),c(“country”,“rgdpch1960”)]
```

We stellen vast dat het Nationaal Inkomen per hoofd varieerde van 400 US\$ voor Ethiopië tot 15253 US\$ voor Zwitserland.

Stel dat we een idee willen over de statistische verdeling van het Nationaal Inkomen per hoofd in het jaar 1970. Dit kan door het commando “hist” :

```
>hist(weil$rgdpch1970,50)
```

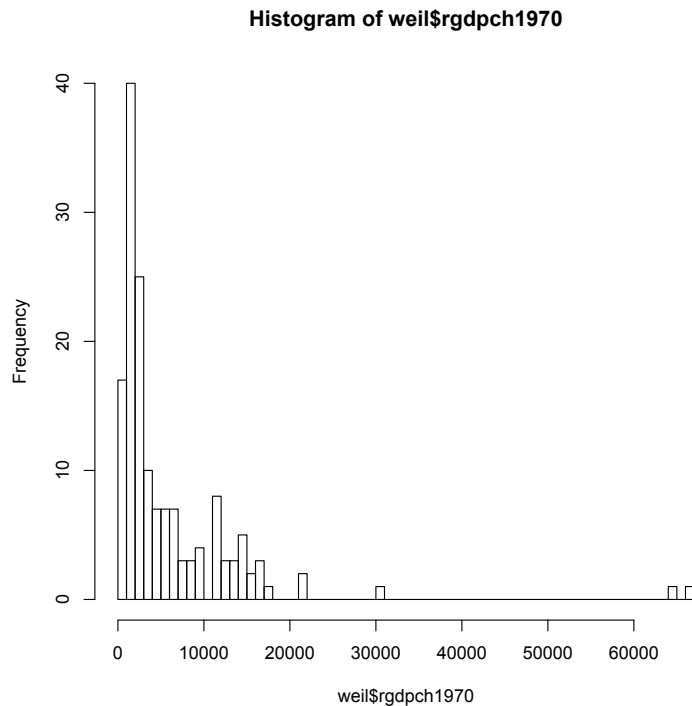


Figure 7: Verdeling van het Nationaal Inkomen per hoofd in 1970 van 153 landen

Het kan nuttig zijn om de “weil” dataset te sorteren volgens een bepaald criterium. Stel dat je wil gan sorteren van laag naar hoog voor de eigenschap “gemiddelde jaarlijkse groei”. De variabele hebben we al gezien : “gy7005”. We definiëren een “sorteringscriterium” als volgt :

```
>sortcrit = c(“gy7005”)
```

```
>weil.gesorteerd = weil[do.call(order, weil[sortcrit]),]
```

Als we dan het land willen zien met de jaarlijkse gemiddelde economische groei ernaast, volstaat volgend commando:

```
>weil.gesorteerd[c(“country”,“gy7005”)]
```

### 9.3 Bewerkingen op het bestand “Maddison”

Een ander interessant gegeven is het Maddison databestand. Dit gegevensbestand is van de hand van dr. Angus Maddison. Deze man wordt terecht de historische econoom van de 21 ste eeuw genoemd. Hij was professor aan het GDC, het “Groningen Growth and Development Center”. Hij schreef enkele opzienbarende publicaties zoals “Contours of the world Economy, 1-2030 AD; Essays in Macroeconomic History, Oxford University Press, September 2007”.

Het hiernavolgend bestand geeft in enkele cijfers een interessant beeld over de voorbije 2000 jaar in economische cijfers.

Je kan het bestand vanop mijn Dropboxfolder downloaden. Hiertoe volstaat volgend commando:

```
>Maddison = read.table
“http://dl.dropbox.com/u/
2195906/Maddison.csv”,header=TRUE,sep=“;”)
>attach(Maddison)
>Maddison
```

Je bekomt volgend databestand:

year	West.Eur	East.Eur	USSR	West.Offsh	Lat.Am	Japan	Asia	Africa	World
0	450	400	400	400	400	400	450	425	444
1000	400	400	400	400	400	425	450	416	435
1500	777	462	500	400	416	500	572	400	565
1600	894	516	553	400	437	520	575	400	593
1700	1024	566	611	473	529	570	571	400	615
1820	1232	636	689	1201	665	669	575	418	667
1870	1974	871	953	2431	698	737	543	444	867
1913	3473	1527	1488	5257	1511	1387	640	585	1510
1950	4594	2120	2834	9288	2554	1926	635	852	2114
1973	11534	4985	6058	16172	4531	11439	1231	1365	4104
1998	17921	5461	3893	26146	5795	20413	2936	1368	5709

Table 2: Nationaal Inkomen per hoofd (in 1990 international \$)

We zijn geïnteresseerd in een vergelijking van de groei van de Western Offshoots ten opzichte van de groei van Africa voor de periode 1973-1998. Het antwoord op deze vraag wordt gegeven door volgende wiskundige formule:

De gemiddelde jaarlijkse groeisnelheid  $g$  wordt gegeven door:

$$g = \sqrt[n]{\frac{X_n}{X_0}} - 1 \quad (1)$$

Hierbij is  $X_n$  het Nationaal Inkomen per hoofd in jaar n en  $X_0$  het Nationaal Inkomen per hoofd in jaar 0.

Als we deze formule omzetten naar  $\mathbb{R}$ -taal met de Maddison gegevens, wordt dit:

$$\begin{aligned} >\text{groei\_van\_Western.Offshoots=} \\ &(\text{Western.Offshoots}[\text{year}==1998] / \\ &\text{Western.Offshoots}[\text{year}==1973])^{(\frac{1}{1998-1973})} - 1 \end{aligned}$$

We berekenen eveneens de gemiddelde jaarlijkse groei voor Africa:

$$\begin{aligned} >\text{groei\_van\_Africa=} \\ &(\text{Africa}[\text{year}==1998] / \\ &\text{Africa}[\text{year}==1973])^{(\frac{1}{1998-1973})} - 1 \end{aligned}$$

Om de verhouding te kennen, voldaait het beide resultaten door elkaar te delen:

$$>\text{groei\_van\_Western.Offshoots} / \text{groei\_van\_Africa}$$

Het resultaat is verbazingwekkend : de Western Offshoots groeiden over een periode van nauwelijks 25 jaar een factor 220 sneller dan Africa.

Kom jij ook tot dit resultaat ?

## 9.4 Berekeningen op het bestand “Forbes”

Zoals reeds aangehaald geeft de volgende website “[www.forbes.com/global2000/](http://www.forbes.com/global2000/)” informatie over de top 2000 bedrijven

Na het bestand te hebben ingeladen en de data te hebben “gelinkt” aan `R`, kunnen we enkele interessante berekeningen en grafieken maken om inzicht te verwerven in de “Forbes2000” data.

Opmerking : het ingeladen bestand bevat de “Forbes2000” data van het jaar 2004. De website geeft de actuele informatie voor april 2011. Er zijn dus wel degelijk verschillen tussen de gegevens van de website en deze die ingeladen worden in `R`.

Stel dat we de bedrijven uit de Forbes lijst willen ordenen volgens grootte van omzet. Dat kan gemakkelijk met volgend commando:

```
>gesorteerd_volgens_omzet= order(Forbes2000$sales)
```

```
>gesorteerd_volgens_omzet
```

Dit commando geeft de indexen van de geordende elementen uit de “Forbes2000” van de variabele “sales”.

Als we de namen willen kennen van de 3 grootste bedrijven volstaat het volgende commando:

```
>Forbes2000$name[gesorteerd_volgens_omzet[1998:2000]]
```

`R` komt terug met :  
[1] “ExxonMobil” “BP” “Wal-Mart Stores”

Als we de data in een meer inzichtelijk schema wensen, volstaat volgend commando:

```
>Forbes2000[gesorteerd_volgens_omzet
[c(2000,1999,1998)],c("name","sales","profits","assets")]
```

Ⓜ komt terug met :

```
10 Wal-Mart Stores 256.33 9.05 104.91
5 BP 232.57 10.27 177.57
4 ExxonMobil 222.88 20.96 166.99
```

Een eerste “statistiek” omtrent de “Forbes2000” dataset bekomen we met volgend commando:

```
>summary(Forbes2000)
```

Ⓜ komt terug met :

rank	name	country	
Min. : 1.0	Length:2000	United States :751	
1st Qu.: 500.8	Class :character	Japan :316	
Median :1000.5	Mode :character	United Kingdom:137	
Mean :1000.5		Germany : 65	
3rd Qu.:1500.2		France : 63	
Max. :2000.0		Canada : 56	
		(Other) :612	

category	sales	profits
Banking : 313	Min. : 0.010	Min. :-25.8300
Diversified financials: 158	1st Qu.: 2.018	1st Qu.: 0.0800
Insurance : 112	Median : 4.365	Median : 0.2000
Utilities : 110	Mean : 9.697	Mean : 0.3811
Materials : 97	3rd Qu.: 9.547	3rd Qu.: 0.4400
Oil & gas operations : 90	Max. :256.330	Max. : 20.9600
(Other) :1120		NA's : 5.0000

assets	marketvalue
Min. : 0.270	Min. : 0.02
1st Qu.: 4.025	1st Qu.: 2.72
Median : 9.345	Median : 5.15
Mean : 34.042	Mean : 11.88
3rd Qu.: 22.793	3rd Qu.: 10.60
Max. :1264.030	Max. :328.54

Het volgende commando geeft eveneens algemene informatie over het databestand “Forbes2000”:

**>lapply(Forbes2000, summary)**

Dit laatste commando geeft iets meer informatie dan het vorig commando summary(Forbes2000).

Voor de variabelen “country”, “category” krijg je nu ook de informatie hoeveel bedrijven in dat land en binnen welke categorie deze bedrijven vallen.

Voer het commando uit en kijk zelf naar het resultaat.

Stel dat we geïnteresseerd zijn in een vergelijking van de bedrijfswinsten binnen elke categorie. Voor elke categorie wordt met het volgende commando de bedrijfswinst opgehaald en wordt het gemiddelde (“mean”) berekend per category.

Merk het argument “na.rm” op in het onderstaand commando. Het geeft aan dat de NON AVAILABLE winsten niet mogen meetellen bij de berekening.

**>gemiddelde\_winsten = tapply(Forbes2000\$profits, Forbes2000\$category, mean, na.rm=TRUE)**

📄 komt terug met :

Aerospace & defense	Banking	
0.2884211	0.4220767	
Business services & supplies	Capital goods	
0.1707143	0.0954717	
Chemicals	Conglomerates	
0.2606000	1.0145161	
Construction	Consumer durables	
0.1981013	0.5663514	
Diversified financials	Drugs & biotechnology	
0.4995570	1.4477778	
Food drink & tobacco	Food markets	
0.5938554	0.2490909	
Health care equipment & services Ho	tels restaurants & leisure	
0.3609231	0.2586486	
Household & personal products	Insurance	
0.5497727	0.3430000	
Materials	Media	
0.1959794	0.2106557	
Oil & gas operations	Retailing	
1.3055556	0.4759091	
Semiconductors	Software & services	
0.4365385	0.5677419	
Technology hardware & equipment T	elecommunications services	
0.2055932	-0.9080303	
Trading companies	Transportation	
0.0280000	0.1388462	
Utilities		
0.2114545		



Als we de verdeling van de variabele “marktwaarde van de bedrijven” willen in kaart brengen, gebruiken we volgende commando’s:

```
>layout(matrix(1:2,nrow=2)) en >hist(Forbes2000$marketvalue)
```

We krijgen dan volgend histogram :

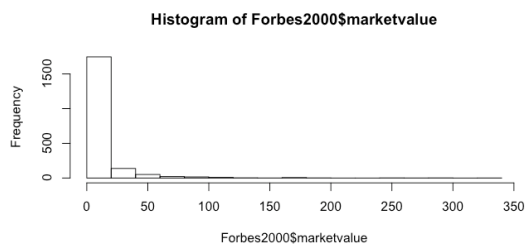


Figure 8: Forbes\$2000 marktwaardes

Als we de log-waarde van de marktwaarden in grafiek brengen krijgen we een meer symmetrische figuur. Door het layout commando komen beide grafieken mooi onder mekaar te staan: zie volgende grafiek.

```
>hist(log(Forbes2000$marketvalue))
```

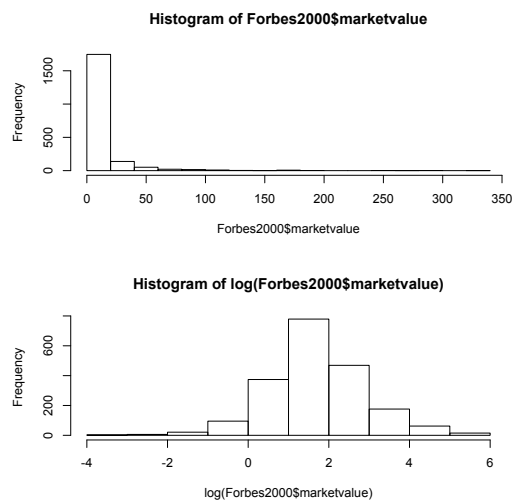


Figure 9: Forbes\$2000 marktwaardes

We bekijken nu het verband tussen de marktwaarden van de bedrijven en hun omzet. Hiervoor heeft R een ingebouwd model “lm” genoemd. We “plotten” de  $\log(\text{marktwaardes})$  versus de  $\log(\text{omzet})$ . We gebruiken volgend commando:

```
>plot(log(marketvalue) ~ log(sales),  
data= Forbes2000, col = rgb(0,0,0,0.1),pch = 16)
```

Dat geeft volgende grafiek:

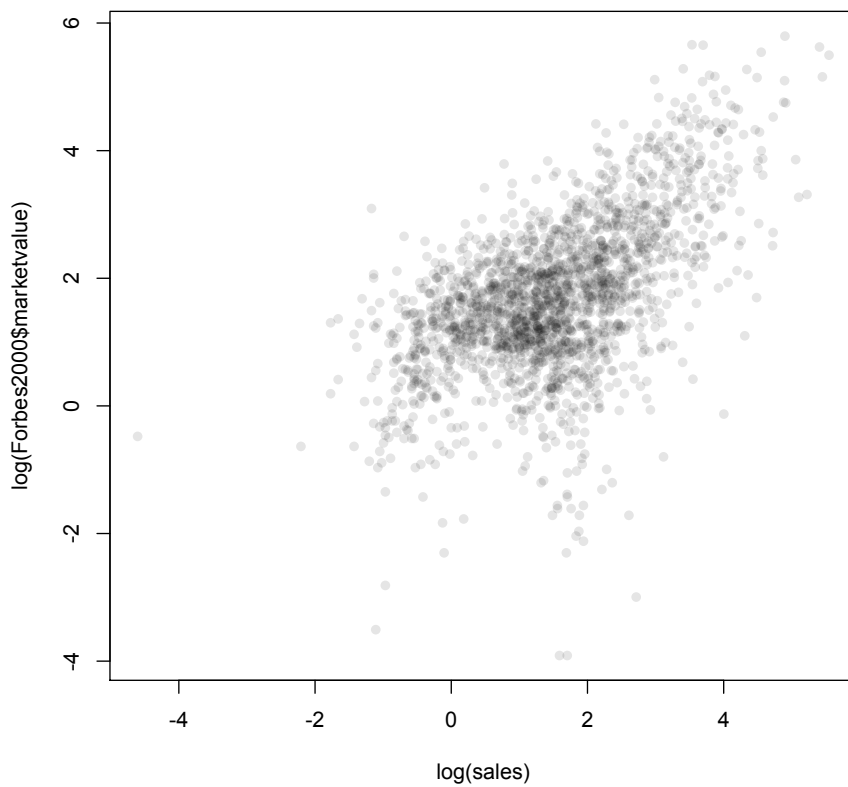


Figure 10: Forbes\$2000 marktwaardes

## 9.5 Het maken van een 'tijd-serie'

Stel dat we een grafische voorstelling willen maken van de groei van het Nationaal Inkomen per hoofd voor 2 landen over een periode van 30 jaar. Het ene land "land\_1" groeit aan een groeisnelheid van 2% per jaar. Het tweede land groeit aan 5% per jaar. Een manier om dit "elegant" op te lossen, is gebruik te maken van de functie "ts" ("time-serie") in  $\mathbb{R}$ .

We beginnen met een "vector" te creëren :

```
>land_1 = rep(100,30)
```

Als we gaan kijken wat de variabele (vector) land\_1 is, dan zien we dat  $\mathbb{R}$  terugkomt met :

```
[1] 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 [20] 100 100 100 100 100 100 100 100 100 100 100 100
```

Land\_2 zullen we eveneens "initialiseren" :

```
>Land_2 = rep(100,30)
```

We creëren via de functie "ts" twee tijdseries.

```
>Land_1_tijds_serie.ts = ts(Land_1,start = 1981,frequency = 1)
```

```
>Land_2_tijds_serie/ts = ts(Land_2,start = 1981,frequency = 1)
```

We gebruiken de techniek van de "iteratieve lus" om voor elk land de eindwaarde na groei te berekenen.

```
>for (1 in 2:30) {
Land_1_tijds_serie.ts[i]= Land_1_tijds_serie.ts[i-1] * (1 + 1.5%)
Land_2_tijds_serie.ts[i]= Land_2_tijds_serie.ts[i-1] * (1 + 3.5%)
}
```

Opgelet : bovenstaand commando moet in 1 lijn worden ingevoerd in de  $\mathbb{R}$ -console !

De volgende commando's maken de grafiek:

```
>plot( land_1_tijds_serie.ts,xlab="years",ylab="growth")
```

```
>lines(land_2_tijds_serie.ts,lty="dashed")
```

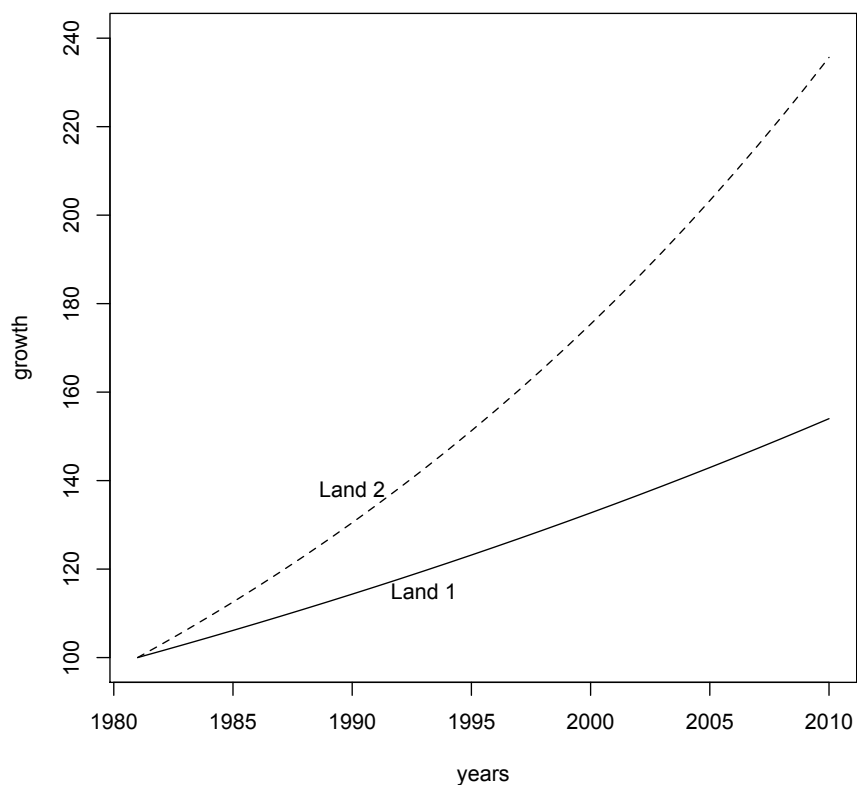


Figure 11: Een Time-serie voorbeeld

Land\_1 groeide jaarlijks met 1,5% en Land\_2 groeide met 3,5%. Na dertig jaar is het verschil tussen beide landen qua Nationaal Inkomen per hoofd enorm. Beide landen vertrokken bij Jaar\_0 op 100. Land\_1 met groeisnelheid van 1,5% eindigt na 30 jaar op 154. Land\_2 dat groeide aan 3,5% eindigt na 30 jaar op 236. Dat is 53% hoger !

# Contents

<b>1</b>	<b>Waar vind je R?</b>	<b>1</b>
1.1	R-installatie voor MacOSX . . . . .	2
1.2	R-installatie voor Windows . . . . .	2
<b>2</b>	<b>Enkele algemeenheden over R</b>	<b>3</b>
<b>3</b>	<b>R als wetenschappelijke rekenmachine</b>	<b>5</b>
<b>4</b>	<b>Hoe R citeren ?</b>	<b>8</b>
<b>5</b>	<b>Datastructuren in R</b>	<b>9</b>
5.1	Variabelen . . . . .	9
5.2	Lijsten . . . . .	10
5.3	Dataframes . . . . .	11
<b>6</b>	<b>Data invoeren of importeren in R</b>	<b>12</b>
6.1	Gegevens intikken op de R console. . . . .	12
6.2	Gegevens kopiëren en plakken uit Mirosoft-Excel . . . . .	13
6.3	Een bestand inladen in R . . . . .	14
6.3.1	Een gegevensbestand vanop het www inladen . . . . .	14
6.3.2	Bestanden en pakketten die in R zelf ingebouwd zitten . . . . .	19
6.3.3	Het databestand “pwt7.0” uit het pakket “pwt” . . . . .	23
6.3.4	Het databestand “Forbes2000” uit het pakket “HSAUR” . . . . .	23
6.3.5	Bestanden vanop je computer inladen in R . . . . .	26
<b>7</b>	<b>De tekst-editor van R</b>	<b>27</b>
<b>8</b>	<b>R gebruiken vanop het internet</b>	<b>28</b>
<b>9</b>	<b>Wiskundige en statistische bewerkingen uitvoeren op databestanden</b>	<b>28</b>
9.1	Berekeningen op het bestand “human_dev” . . . . .	28
9.2	Bewerkingen en statistische berekeningen op het databestand “weil” . . . . .	33
9.3	Bewerkingen op het bestand “Maddison” . . . . .	36
9.4	Berekeningen op het bestand “Forbes” . . . . .	38
9.5	Het maken van een ‘tijd-serie’ . . . . .	43