

Project Dynamica: oefenopgaven met R

De onderstaande opgaven dienen in R gemaakt te worden; uitwerkingen hoeven niet ingeleverd te worden. Zie de website <http://www.r-project.org/> voor R manuals.

Start R door op het R-icoon te klikken. In het window met de naam *R Console* kunnen de R commando's worden ingevoerd. Het teken '`>`' aan het begin van de regel is de R-prompt. De `>` prompt wordt gevolgd door de uit te voeren commando's. De `>` prompt wordt vervangen door een `+` prompt als het commando nog niet is afgelopen. Met het commando `q()` kan R worden afgesloten.

Functies in R kennen twee soorten argumenten: 1) argumenten die de gebruiker verplicht dient te specificeren, en 2) default argumenten die de gebruiker kan specificeren, maar een standaard instelling kennen. De syntax `> ?functienaam` toont de help-file van de functie `functienaam`.

Nog enkele tips:

- i) Met de `↑` en de `↓` cursor-toetsen is het mogelijk door reeds ingevoerde commando's te lopen.
- ii) Via **File** → **New script** opent R een script. Hier kunnen vele regels aan commando's ingevoerd worden. Een selectie hiervan kan direct in R gerund worden: **Edit** → **Run line or selection**.
- iii) Sla altijd het script op! Dat scheelt veel dubbel werk.

Opgave 0: De volgende oefensessie toont enkele mogelijke commando's in R. De R prompt wordt gevolgd door het commando dat ingeven dient te worden, en een korte beschrijving van wat het commando doet. Voer dit zelf uit: voer het commando in gevolgd door `<ENTER>`.

<code>> x <- c(-1, 8, 1)</code>	maak een vector <code>x</code> met waarden -1,8, en 1.
<code>> x <- c(x, 99)</code>	voeg de waarde 99 toe het eind van de vector <code>x</code> .
<code>> x <- c(sqrt(2), x, 99)</code>	voeg de waarde $\sqrt{2}$ toe het begin van de vector <code>x</code> .
<code>> x <- x[-2]</code>	verwijder het tweede element van de vector <code>x</code> .
<code>> x <- c(x,x)</code>	dupliceer de vector <code>x</code> .
<code>> x <- 1:20</code>	maak een vector <code>x</code> met waarden 1, 2, 3, ..., 20.
<code>> x</code>	print waarden van <code>x</code> op het scherm.
<code>> m <- matrix(x,4,5,byrow=T)</code>	maak een matrix <code>m</code> met 4 rijen en 5 kolommen met de waarden van <code>x</code> rij-gewijs geordend.
<code>> m</code>	print matrix <code>m</code> op het scherm.
<code>> y <- sample(1:100,20)</code>	genereer een random sample van omvang 20 uit de getallen 1, 2, 3, ..., 100.
<code>> z <- x+y</code>	bereken de som van <code>x</code> en <code>y</code> elements-gewijs.
<code>> y <- x+ 2*y</code>	transformeer <code>y</code> elements-gewijs.
<code>> cbind(x,y)</code>	maak een matrix met kolommen <code>x</code> en <code>y</code> en print het resultaat.

<pre> > plot(x,y) > lsfit(x,y)\$coef > abline(lsfit(x,y)) > lines(spline(x,y)) > x <- rnorm(50,0,sqrt(2)) > y <- rnorm(50) > mean(x) > sd(x) > var(x) > cor(x,y) > x[x<0] > sum(x<0) > hist(x, prob=T) > help(hist) > ?hist > u <- seq(-5,5,0.1) > v <- dnorm(u,0,sqrt(2)) > lines(u,v) > {hist(x,xlim=c(-6,6),prob=T) + lines(u,v)} > plot(c(-5,sort(x),5), + c(0,1:50/50,1), + type="s",ylim=c(0,1), + xlab="x",ylab="the empdf of x") > lines(u,pnorm(u)) > w <- seq(-pi,pi,length=100) > plot(cos(w),sin(w),type="l") </pre>	zet y tegen x uit in een grafiek. bereken de coëfficiënten van de kleinste kwadraten lijn. voeg de kleinste kwadraten lijn toe aan de grafiek. voeg ook een exacte spline fit toe. trek een steekproef van omvang 50 uit de normale verdeling met verwachting 0 en variantie 2. idem voor de standaard normale verdeling. bereken het gemiddelde van de waarden in x. bereken de standaard deviatie van de waarden in x. bereken de variantie van de waarden in x. bereken de correlatie tussen x en y. selecteer de negatieve elementen uit x. tel het aantal negatieve elementen in x. maak een histogram. vraag de documentatie over de function <code>hist</code> op. idem. maak een rij van punten tussen -5 and 5 met stapgrootte 0.1. bereken de dichtheid van de normale verdeling met verwachting 0 en variantie 2. voeg curve van de berekende normale verdeling toe. teken het histogram opnieuw op groter interval, maar voer het commando nog niet uit. plus additioneel commando, en voer beide uit. teken de empirische verdelingsfunctie van x op $(-5,5)$.¹ voeg de werkelijke distributie functie toe. maak een regelmatig grid van 100 punten. teken een cirkel.
--	---

¹ Merk op: de kleinste en grootste waarde van de op de x -as weer te geven vector dienen respectievelijk kleiner dan `min(x)` en groter dan `max(x)` te zien. Wellicht dien je de waarden -5 en/of 5 aan te passen om dit te ondervangen.

Opgave 1: Vektoren

- 1a) Maak een vector x bestaande uit de getallen 23, 0.149, -5.15, and 36.
- 1b) Voeg aan deze vector het getal -28 toe.
- 1c) Orden de elementen van x van klein naar groot en noem deze geordende vector y .
- 1d) Vermenigvuldig elk element van x met 4.
- 1e) Rond elk element van x af tot een decimaal.
- 1f) Selecteer alle positieve elementen van x .
- 1g) Verander het derde element van x in 7.

Opgave 2: Matrices

- 2a) Maak een vector x_2 bestaande uit de rij van getallen tussen 2 en 4 met stapgrootte 0.1.
Maak een matrix m met zeven rijen en drie kolomen in welke de 21 element van x_2
kolomsgewijs geordend zijn.
- 2b) Print het element in de tweede rij en in de eerste kolom van m .

- 2c) Print de derde kolom van m .
- 2d) Bereken het gemiddelde van de getallen in m .
- 2e) Bereken de gemiddelde waarde van elke kolom van m middels de R-functie `apply`.

Voor de opgaven hieronder dient een eigen functie in R geschreven te worden. Functies in R kennen de volgende syntax:

```
functienaam <- function(argument1, argument2, ...){
  ...
  some command lines
  ...
  return(object-to-be-returned)
}
```

Meer hierover in de handleiding of op internet.

Opgave 3: Functies

- 3a) Schrijf een functie dat elk element van een vector x met 5 vermenigvuldigt en er vervolgens 3 bij optelt, en het resultaat als output heeft.
Een vaas bevat 40 rode en 60 witte ballen.
- 3b) Schrijf een functie die, met teruglegging, 50 ballen uit de vaas trekt, en tevens de fractie rode en witte ballen in deze trekking berekent. De functie geeft de fracties als output terug.
- 3c) Herschrijf de functie van onderdeel 3b) voor de situatie dat n ballen met teruglegging worden getrokken uit een vaas met mr rode en mw witte ballen. Neem als argument voor de functie n , mr en mw met default waarden $n = 100$, $mr = 40$ en $mw = 60$.

Opgave 4: Benader $\pi\sqrt{2}$ in vijf decimalen. *Hint:* bekijk de manual van de functie `round`.

Opgave 5: Vind $x \in [0, 1]$ z.d.d. $x^2 - 3x + 1 = 0$. *Hint:* functies `uniroot` en `function`. Een alternatief is de functie: `polyroot`.

Opgave 6: Definieer de functies $f(x) = 5x - 1$ en $g(x) = \cos(x)$. Probeer vervolgens op te lossen: $f(x) = g(x)$. Als dit niet met R lukt, benader dan de oplossing zo goed mogelijk. Teken vervolgens in één figuur de grafieken van beide functies en zet bij het snijpunt een 'S'. *Hint:* vorige opgave, en de functies `plot`, `lines` en `text`.

Opgave 7: Bereken de integraal van $f(x) = x \arctan(x)$ over het interval $[0, 100]$. *Hint:* Functies `atan` and `integrate`.

Opgave 8: *Merk op: deze opgave vereist schrijfrechten op de computer.*

Met $H_n(x)$ noteren we het n -de graads Hermite polynoom. Zoek op internet in welk package deze polynomen gedefinieerd zijn. Installeer het package (hiervoor heb je schrijfrechten nodig, en kan mogelijk niet op een VU-computer!). Laad het package. Zoek in R op wat Hermite polynomen zijn en geef vervolgens het 4-de graads Hermite polynoom. *Hint:* functies `library`.

Opgave 9: De matrix $\mathbf{A}$ en de vector $\mathbf{b}$ zijn gegeven door:

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 1 \\ 2 & 3 & 5 \\ -1 & 6 & 5 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$$

- 9a) Bepaal de lengte van $\mathbf{b}$. *Hint:* commando voor matrix vermenigvuldiging `%*%`,

bijv: `A %*% A`.

9b) Geef een basis voor de nulruimte van **A**. *Hint*: functie `Null` in de `MASS`-library.

9c) Los op: $\mathbf{Ax} = \mathbf{b}$. *Hint*: functie `solve`.

Opgave 10) Bepaal de eigenwaarden en bijbehorende eigenvectoren van

$$\mathbf{C} = \begin{pmatrix} 3 & -2 & 1 \\ 8 & -5 & 2 \\ 4 & -2 & 0 \end{pmatrix}.$$

Opgave 11) (Deze opgave fungeert als oefening voor de eindopdracht.) Op de waddeneilanden wordt het weer elke dag geclassificeerd als zonnig, bewolkt of regenachtig. Het weer voor de volgende dag hangt alleen af van het weer van vandaag, maar niet van de dagen daarvoor. Als het vandaag zonnig is, dan is het morgen zonnig, bewolkt of regenachtig met de respectievelijke kansen 0.70, 0.10 en 0.20. Voor een bewolkte dag zijn de respectievelijke kansen voor de volgende dag gelijk aan 0.50, 0.25 en 0.25, terwijl voor een regenachtige dag de respectievelijke kansen voor de volgende dag 0.40, 0.30 en 0.30 zijn. We definiëren een Markov keten $\{X_n\}$ met drie toestanden 1, 2 en 3. Het proces is op dag n in toestand 1 bij een zonnige dag, in toestand 2 bij een bewolkte dag en in toestand 3 bij een regenachtige dag.

- Stel de matrix **P** op van 1-staps overgangskansen [dit is dus de matrix in het stelsel vergelijkingen $y_{k+1} = \mathbf{P} y_k$, waarbij y_k de vector is uit $\mathbb{R}^3$ die de kansen aangeeft op zonnig, bewolkt resp. regenachtig].
- Wat is de kans dat het overmorgen zonnig is, als het vandaag regenachtig is? En hoe zit dat over drie dagen?
- Bepaal de kansverdeling van het weer over een groot aantal dagen [kortom, bepaal de limiet van de vector y_k voor $k \rightarrow \infty$].

Hieronder nog twee opgaven ten einde het programmeren in **R** te oefenen. I.h.b. het `if`-statement en de `for`- en `while`-loop. Deze structuren kennen de volgende syntax:

```
if ( cond ){ expr1 } else { expr2 }
for ( var in list ){ expr }
while ( cond ){ expr }
```

Meer hierover in de handleiding of op internet.

Opgave 12) Priemgetallen: Laat alle priemgetallen tussen de 100 en 200 afdrukken en laat **R** uitrekenen hoeveel dat er zijn (druk dat ook af!). *Hint*: de functie `isprime` in de `gmp`-library bepaald of een integer priem is of niet; gebruik tevens een `for`-loop en het `if`-statement. Mocht dit package niet beschikbaar zijn, programmeer dan zelf een functie die bepaald of een getal priem is. Dit kan eveneens met een `for`-loop. De modulus-functie is in **R** geïmplementeerd als `%`.

Opgave 13) Newton-Raphson: Met het algoritme van Newton-Raphson is het (meestal) mogelijk een nulpunt van een differentieerbare functie f te bepalen. Het is een iteratieve methode, die als volgt is gedefinieerd:

$$\begin{cases} x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \\ x_0 \text{gegeven.} \end{cases}$$

Als we het nulpunt van $f(x) = x^2 - 2$ hiermee berekenen, vinden we in feite een benadering van $\sqrt{2}$. Voer dit uit en gebruik als stopcriterium dat twee opeenvolgende iteraties minder dan 10^{-9} van elkaar verschillen. *Hint:* gebruik een `while`-loop.