

Oefening 4

Opgave:

Stel $z = \sqrt{2} e^{\frac{3\pi i}{5}}$. Bepaal het reëel en imaginair deel. Bepaal (in graden) het argument (= poolhoek) van het complex toegevoegde van $(z+1)$.

Oplossing:

We beginnen met het definiëren van het complex getal z in de exponentiële vorm. Hier gebruiken we de notatie $1i$ voor de complexe eenheid. Als we i of j zouden gebruiken dan bestaat het gevaar dat we deze variabelen zouden overschrijven en ze niet meer de complexe eenheid als waarde hebben. $1i$ kan niet overschreven worden en dus dekken we ons op die manier in.

```
z = sqrt(2)*exp(3*pi*1i/5)
```

```
z = -0.4370 + 1.3450i
```

Het reëel en imaginair deel van het complex getal kunnen we bepalen met `real` en `imag`.

```
R = real(z)
```

```
R = -0.4370
```

```
I = imag(z)
```

```
I = 1.3450
```

Het complex toegevoegde van $z + 1$ noemen we `zconj` en berekenen we met behulp van het commando `conj`. Dit wordt door Matlab al automatisch in cartesische vorm voorgesteld.

```
zconj = conj(z+1)
```

```
zconj = 0.5630 - 1.3450i
```

Complexe getallen kunnen naast de cartesische vorm $(a + bj)$, ook in de exponentiële vorm geschreven worden: $(r \cdot e^{\theta j})$. Voor de exponentiële vorm hebben wij de hoek θ en de straal r nodig. Deze berekenen we met `angle` en `abs`. De hoek wordt automatisch berekend in radialen. Wil je het in graden hebben dan gebruik je het commando `rad2deg`.

```
hoek = angle(zconj)
```

```
hoek = -1.1744
```

```
rad2deg(hoek)
```

```
ans = -67.2870
```

```
straal = abs(zconj)
```

```
straal = 1.4581
```

Oefening 5

Opgave:

Voer op de meest handige manier de matrix A in die 18 rijen bevat:

$$A = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 1 & 3 & 5 & 7 & 9 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 3 & 5 & 7 & 9 \end{pmatrix}$$

Oplossing:

We definiëren eerst een rijvector met startwaarde 1, een stap van 2 en eindwaarde 9. Dit komt overeen met de rijen uit de matrix die we willen genereren.

v=1:2:9

$$v = \begin{matrix} 1 & 3 & 5 & 7 & 9 \end{matrix}$$

Daarna definiëren we een 18×1 -matrix met 1 als herhaalde waarde.

```
een=ones(18,1)
```

Door deze kolomvector en rijvector met elkaar te vermenigvuldigen verkrijgen we de gevraagde matrix.

A=een*v

A = 18x5				
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
1	3	5	7	9
.

Oefening 9

Opgave:

Bereken het gemiddelde van de elementen uit de verzameling $\{f(0), f(0.1), \dots, f(1)\}$ met $f(t) = \frac{t}{1 + \sqrt{t}}$. (Werk zonder lussen!)

Oplossing:

We definiëren eerst een rij met startwaarde 0, een stap van 0.1 en eindwaarde 1.

```
t_lijst = 0:0.1:1;
```

We definiëren een nieuwe rij waarvan de elementen de overeenkomstige afbeeldingen $f(t)$ zijn van de oorspronkelijke rij. Hiervoor moeten we een **elementsgewijze** deling uitvoeren ($./$). De bewerking `sqrt` gebeurt sowieso elementsgewijs.

```
f_lijst = t_lijst./(1+sqrt(t_lijst))
```

```
f_lijst = 1x11
0    0.0760    0.1382    0.1938    0.2450    0.2929    0.3381    0.3811 ...
```

We bereken het gemiddelde van deze afbeeldingen met $\frac{1}{N} \sum_{i=1}^N f_i$, waarbij N het aantal elementen is in de lijst dat we kunnen berekenen met `length`.

```
gem = sum(f_lijst)/length(f_lijst)
```

```
gem = 0.2772
```

Oefening 10

Opgave:

Een rechthoekige driehoek heeft een omtrek van 5 cm en de lengte van de schuine zijde is $2/3$ van de som van de lengtes van de rechthoekslijnen. Wat is de waarde van de zijden van deze driehoek?

Opllossing:

Er zijn 3 onbekenden: a , b en c (namelijk de 3 zijden van een rechthoekige driehoek). Aangezien het een rechthoekige driehoek is, geldt de stelling van Pythagoras ($a^2 + b^2 = c^2$). Er is ook gegeven dat de omtrek van de driehoek gelijk is aan 5 cm (m.a.w. $a + b + c = 5$) en dat $c = \frac{2}{3}(a + b)$. Met behulp van deze 3 vergelijkingen kunnen de onbekenden a , b en c berekend worden. We werken dit eens numeriek uit en ook eens symbolisch en dan vergelijken we de resultaten.

Numeriek:

We beginnen met een driedimensionale functie te definiëren in drie veranderlijken met behulp van `@`. Elke component van deze functie komt overeen met een voorwaarde van de driehoek. We gebruiken het commando `fsolve` om naar oplossingen te zoeken. `fsolve` kan enkel nulpunten vinden van functies en dus moeten we de gegeven vergelijkingen omvormen zodat alle termen in hetzelfde lid staan en zodat de componenten nul zijn bij het vinden van de juiste driehoek. Het algoritme van `fsolve` vergt dat we een startwaarde meegeven. Dit is het punt waar het algoritme zal vertrekken bij het zoeken naar een oplossing. Als je al weet waar deze oplossing ongeveer ligt, dat kan je een startwaarde meegeven dat hier in de buurt ligt. In deze oefening weten we niet zo veel over waar de oplossingen zullen liggen, wel dat de waarden positief moeten zijn en de som niet groter is dan 5. Hier kiezen we als startwaarde $a = b = c = 1$.

```
F=@(x) [x(1)+x(2)+x(3)-5,x(1)^2+x(2)^2-x(3)^2,x(3)-2/3*(x(1)+x(2))];  
fsolve(F, [1,1,1])
```

No solution found.

`fsolve` stopped because the problem appears regular as measured by the gradient,
but the vector of function values is not near zero as measured by the
value of the function tolerance.

```
<stopping criteria details>  
ans = 1×3  
1.4655 1.4655 2.0662
```

Matlab waarschuwt dat er geen oplossing gevonden is. Het vermelde antwoord is slechts de beste benadering die gevonden kon worden. Als we de waarden invullen in de functie krijgen bij gevolg ook niet nul:

```
F([1.4655 1.4655 2.0662])
```

```
ans = 1×3  
-0.0028 0.0262 0.1122
```

Symbolisch:

Hier maken we drie symbolen aan voor de onbekende driehoekslijnen met behulp van `syms`.

```
syms a b c
```

Met behulp van het commando `solve` kunnen we een stelsel van symbolische vergelijkingen oplossen. Let op dat we hier twee gelijkheidstekens gebruiken in de plaats van één voor het aanduiden van een vergelijking.

```
[a,b,c]=solve(a+b+c==5,c^2==a^2+b^2,c==2/3*(a+b))
```

a =

$$\begin{pmatrix} \frac{3}{2} + \frac{1}{2}i \\ \frac{3}{2} - \frac{1}{2}i \end{pmatrix}$$

b =

$$\begin{pmatrix} \frac{3}{2} - \frac{1}{2}i \\ \frac{3}{2} + \frac{1}{2}i \end{pmatrix}$$

c =

$$\begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

Met de symbolische werkwijze bekomen we enkel complexe oplossingen. We vinden dus geen relevante oplossingen voor deze oefening, een driehoekszijde kan namelijk geen complexe lengte hebben.

Conclusie: Er bestaat geen driehoek die aan deze gevraagde voorwaarden voldoet.

Oefening 11

Opgave:

Hoeveel van de volgende getallen zijn groter dan 0.5?

$$\sin(1), \sin(2), \dots, \sin(1000)$$

Oplossing:

Hier moeten we een bepaalde operatie een eindig aantal keer herhalen, hiervoor gebruiken we een for-lus. Bij een for-lus is er steeds een variabele die een vector afloopt. De actie die moet worden ondernomen, hangt af van de waarde van die variabele. De vector die we hier aflopen is $[1, 2, \dots, 1000]$; voor elk van deze waarden moeten we nagaan of de sinus van dit getal groter is dan 0.5. Dit kunnen we doen aan de hand van het if-commando. Met het commando if wordt eerst nagegaan of de voorwaarde voldaan is. Zo ja, dan kan een opdracht uitgevoerd worden.

We kunnen de tel bijhouden door een variabele teller te introduceren en die in het begin van de code gelijk te stellen aan nul. Telkens als het if-commando de voorwaarde goedkeurt, kan er bij teller één worden opgeteld.

Voor deze oefening gaan we een for-lus combineren met een if-commando zoals volgt:

```
teller = 0;
for i = 1:1000
    if (sin(i) > 0.5)
        teller = teller+1;
    end
end
```

We kunnen de waarde van teller op het einde van de lus opvragen met behulp van disp.

```
disp(teller)
```

Oefening 13

Opgave:

Bereken $\sum_{n=0}^{10} n!$.

Oplossing:

We definiëren eerst een rij met startwaarde 0, een stap van 1 en eindwaarde 10:

```
v = 0:10;
```

We gebruiken het commando factorial om de elementgewijze faculteiten van de waarden in v te berekenen:

```
v_fac = factorial(v);
```

We kunnen de som van de waarden in v_fac van nul tot tien met behulp van het commando sum uitrekenen.

```
sum(v_fac)
```

```
ans = 4037914
```

Oefening 14

Opgave:

Bereken $\tan\left(2\arccos\left(-\frac{1}{5}\right)\right)$.

Oplossing:

Het commando voor het berekenen van de `bfcos` is `acos`, voor de tangens is dit `tan`.

```
tan(2*acos(-1/5))
```

```
ans = 0.4260
```

Oefening 15

Opgave:

Een rechthoekige driehoek heeft een omtrek van 5 cm. Wat is de waarde van de zijden wanneer de driehoek maximale oppervlakte heeft.

Oplossing:

Deze driehoek voldoet aan twee voorwaarden. Namelijk de stelling van Pythagoras: $a^2 + b^2 = c^2$ en de omtrek is 5: $a + b + c = 5$. Uit de tweede vergelijking vinden we $c = 5 - a - b$. Als we dit substitueren in de eerste vergelijking vinden we: $a^2 + b^2 = (5 - a - b)^2$. Deze vergelijking kunnen we symbolisch oplossen zodanig dat we a in functie van b kunnen schrijven.

```
syms a b;
solve(a^2 + b^2 == (5-a-b)^2, a)
```

```
ans =
```

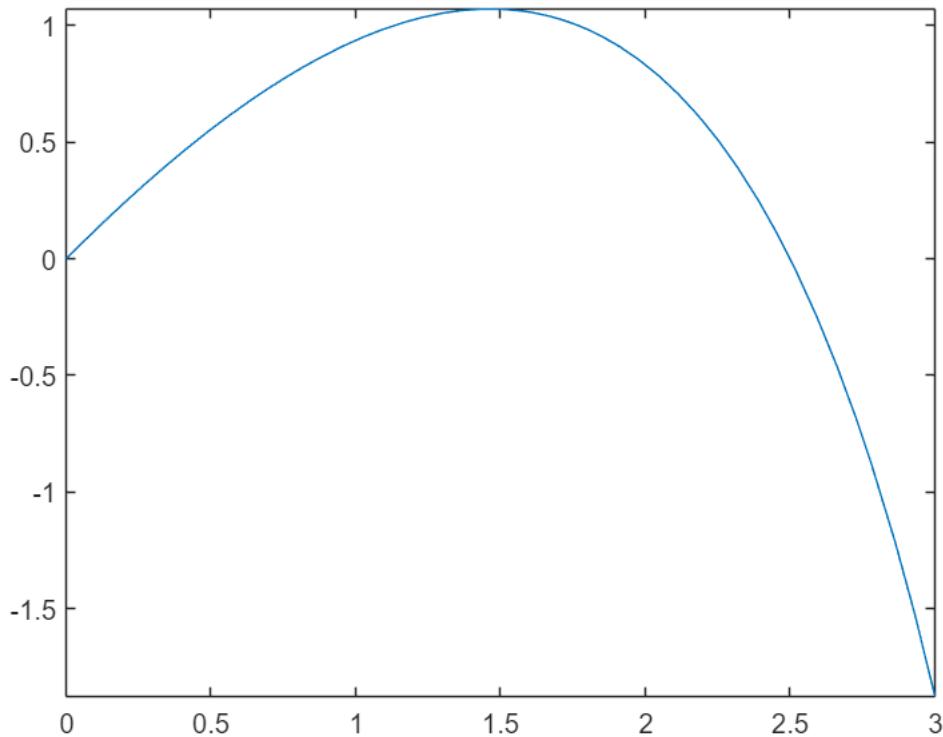
$$\frac{10b - 25}{2b - 10}$$

De oppervlakte van de driehoek is gelijk aan $\frac{1}{2}ab$. Als we de bovenstaande vergelijking hier in substitueren krijgen we $\text{opp} = \frac{1}{2} \cdot \frac{10b^2 - 25b}{2b - 10}$; hiervan kunnen we een functie maken.

```
f = @(b) 0.5*(10*b^2 - 25*b)/(2*b-10);
```

We tekenen de functie om te zien of er effectief een maximum is en waar die ongeveer ligt.

```
fplot(f(b), [0, 3])
```



Om rekenwerk te besparen kunnen we het commando `fminbnd` gebruiken om het maximum te vinden. Let op: met dit commando kan je enkel het minimum van een functie vinden. We moeten onze functie dus spiegelen ten opzichte van de X-as. Op die manier wordt een maximum een minimum.

```
f_min = @(b) (-f(b));
[b0,val] = fminbnd(f_min,1.4,1.6)
```

```
b0 = 1.4645
val = -1.0723
```

Voor de waarde van b vinden we 1.4645. Voor a vinden we:

```
a0 = (10*b0-25)/(2*b0-10)
```

```
a0 = 1.4645
```

En voor c vinden we:

```
c0 = 5 - a0 - b0
```

```
c0 = 2.0711
```

De waarde van de oppervlakte is 1.0723.

Oefening 17

Opgave:

Los op over \mathbb{R} : $2x^3 - x^2 - 15x + 18 > 0$

Oplossing:

Er zijn twee manieren om deze oefening op te lossen.

Eerste methode:

We beginnen met de functie te definiëren die overeenkomt met het linkerlid van de vergelijking:

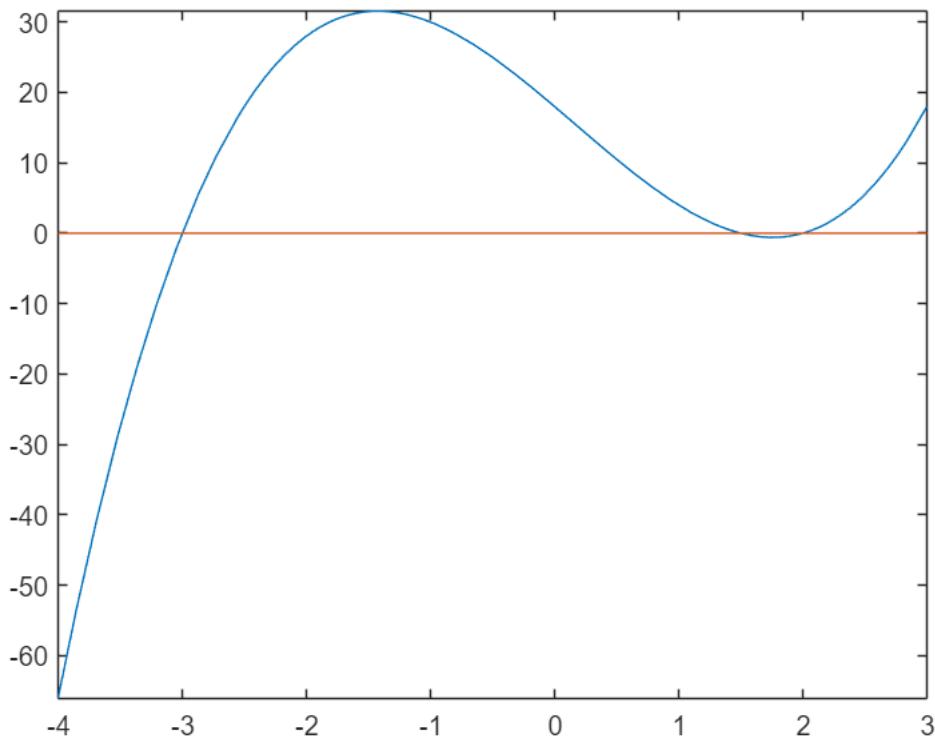
```
f= @(x) (2*x^3-x^2-15*x+18);
```

Nadat we deze functie hebben gedefinieerd kunnen we op zoek gaan naar de nulpunten. Het is best om eerst een grafische voorstelling van de functie te maken, zodat we weten hoeveel nulpunten er zijn, waar ze ongeveer liggen en waar de functie negatief of positief is.

```
fplot(f, [-4,3])
hold on
```

Tezamen met de functie plotten we ook de x-as ($y = 0$):

```
fplot(0, [-4,3])
```



Op de grafiek is te zien dat er drie nulpunten zijn. Deze nulpunten kunnen we numeriek gaan berekenen met fzero:

```
fzero(f,1.5)
```

```
ans = 1.5000
```

```
fzero(f,2)
```

```
ans = 2
```

```
fzero(f,-2.5)
```

```
ans = -3
```

Nu dat we de nulpunten exact weten liggen, kunnen we afleiden uit de grafiek waar de functie positief is.

Namelijk als $x \in]-3, 1.5] \cup [2, +\infty[$.

```
clear;
```

Tweede methode:

Bij de tweede methode zoeken we op een symbolische manier naar de oplossing van de ongelijkheid.

Ongelijkheden kunnen net zoals gelijkheden opgelost worden met de methode solve; er zijn echter een paar extra stappen die moeten worden gezet.

```
syms x
f = @(x) 2*x^3-x^2-15*x+18;
```

Bij de methode solve worden twee extra opties meegegeven: ReturnConditions en Real. De optie Real zorgt ervoor dat er enkel naar reële oplossingen wordt gezocht. Dit heeft een zuiver wiskundige reden, complexe getallen hebben namelijk geen ordening, met andere woorden, er kan niet gezegd worden van een bepaald complex getal dat het groter is dan een ander. Daarom is de vergelijking $f(x) > 0$ slecht gedefinieerd in het complexe domein en moet er een expliciete restrictie naar de reële getallen worden gemaakt, waar deze vergelijking wel goed is gedefinieerd.

De optie RealConditions gaat ervoor zorgen dat er naast oplossingen voor de variabele x ook parameters en conditions worden teruggegeven door de methode solve. Deze drie dingen worden allemaal opgeslaan in de variabele opl.

```
opl = solve(f(x)>0, x, 'ReturnConditions', true, 'Real', true)
```

```
opl = struct with fields:
    x: [2x1 sym]
    parameters: x
    conditions: [2x1 sym]
```

In het geval van een ongelijkheid zoals in deze oefening is conditions hetgeen waar we opzoek naar zijn. Dit zijn namelijk de voorwaarden voor de variabele x opdat er aan de vergelijking $f(x) > 0$ wordt voldaan. Deze voorwaarden kunnen we opvragen door gebruik te maken van opl.conditions.

```
opl.conditions
```

ans =

$$\left(\begin{array}{l} 2 < x \\ -3 < x \wedge x < \frac{3}{2} \end{array} \right)$$

Oefening 18

Opgave:

Los op over \mathbb{R} : $x^5 - 5x = -2$

Oplossing:

Eerste methode:

We kunnen proberen van een symbolische oplossing te vinden voor dit probleem:

```
syms x;
f = solve(x^5-5*x == -2, x)
```

```
f =

$$\begin{pmatrix} \text{root}(z^5 - 5z + 2, z, 1) \\ \text{root}(z^5 - 5z + 2, z, 2) \\ \text{root}(z^5 - 5z + 2, z, 3) \\ \text{root}(z^5 - 5z + 2, z, 4) \\ \text{root}(z^5 - 5z + 2, z, 5) \end{pmatrix}$$

```

Hier komen we echter een antwoord uit waar we niet veel mee zijn. Dit komt omdat deze vijfdegraadsvergelijking eigenlijk geen analytische oplossing heeft en Matlab kan dit dus niet op een symbolische manier vinden. Door middel van het commando vpa kunnen we wel aan Matlab vragen om hier numerieke waarden voor te berekenen. Het argument 5 staat hier voor het aantal beduidende cijfers dat gebruikt wordt in de berekening.

Let op: dit kan de nauwkeurigheid van het antwoord beïnvloeden en dus moet je ervoor zorgen dat je een waarde kiest die groot genoeg is.

```
vpa(f, 5)
```

```
ans =

$$\begin{pmatrix} 0.4021 \\ 1.3719 \\ -0.095975 - 1.5108i \\ -0.095975 + 1.5108i \\ -1.582 \end{pmatrix}$$

```

Hier krijgen we vijf oplossingen, we moeten zelf de reële er uit halen.

Tweede methode:

We kunnen deze vergelijking omvormen naar: $x^5 - 5x + 2 = 0$. Om deze vergelijking op te lossen moeten we dus op zoek gaan naar de nulpunten van een veelterm. Dit kunnen we doen met het commando roots.

Eerst en vooral moeten we daarvoor een vector maken met de gerangschikte coëfficiënten van de veelterm (beginnen met de coëfficiënt horende bij de grootste orde):

```
coeff = [1 0 0 0 -5 2];
roots(coeff)
```

```
ans = 5x1 complex
-1.5820 + 0.0000i
-0.0960 + 1.5108i
-0.0960 - 1.5108i
1.3719 + 0.0000i
0.4021 + 0.0000i
```

Zoals verwacht vinden we vijf nulpunten. Er wordt enkel naar de reële nulpunten gevraagd en deze zijn -1.582, 1.3719 en 0.4021.

Oefening 20

Opgave:

Schets de kromme $y = \operatorname{Bgtg}\left(\frac{x-1}{2} - \frac{1}{2x-2}\right)$. Bepaal de limiet in 1 en in ∞ . Bepaal y' . Zoek punten waarin $y' = 2$. Verklaar de resultaten op de grafiek.

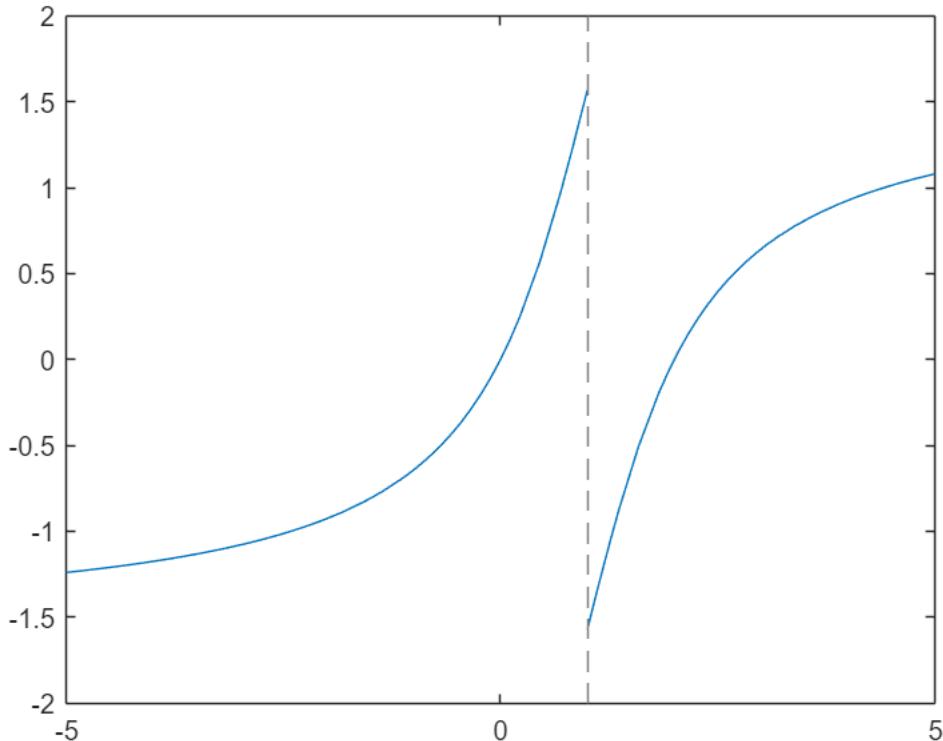
Oplossing:

We beginnen met het symbool x en de functie in x te definiëren:

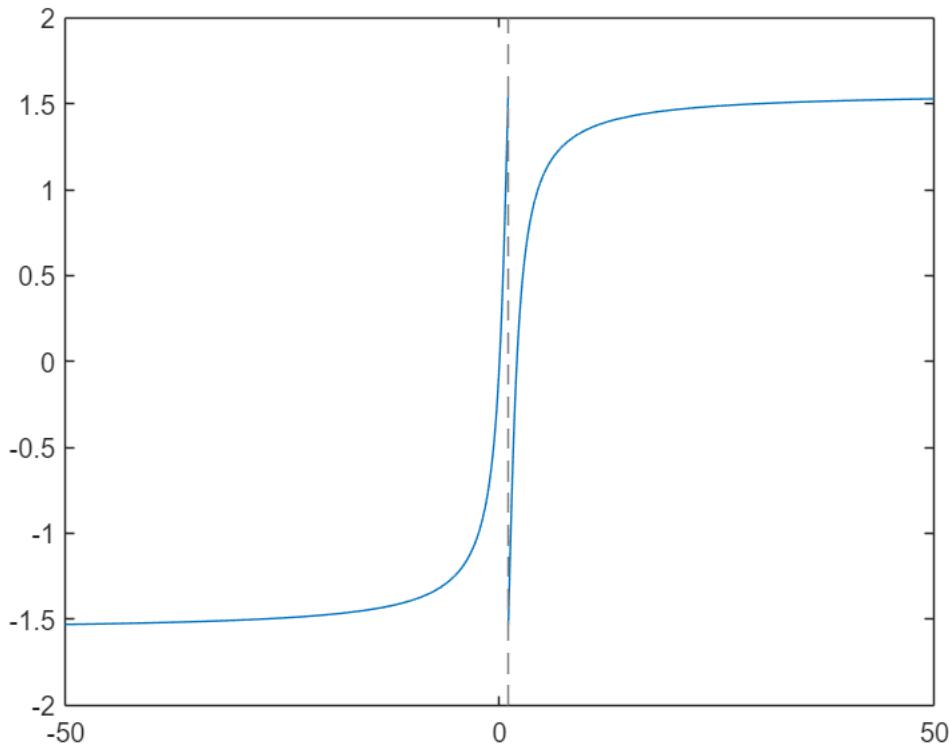
```
syms x
y = @(x) atan((x-1)/2-1/(2*x-2));
```

Zoals gevraagd maken we een tekening van de gegeven functie.

```
fplot(y(x))
axis([-5 5 -2 2])
```



```
fplot(y(x))
axis([-50 50 -2 2])
```



Vervolgens proberen we de limiet in $x = 1$ te berekenen:

```
limit(y(x), x, 1)
```

```
ans = NaN
```

Als output krijgen we NaN. Dit betekent *not a number* en duidt aan dat er ergens iets is misgelopen. In dit geval is het probleem dat $x = 1$ een punt van discontinuïteit van de functie is. Om dit op te lossen zullen we de linkerlimiet en de rechterlimiet apart moeten berekenen:

```
limit(y(x), x, 1, 'left')
```

```
ans =
```

$$\frac{\pi}{2}$$

```
limit(y(x), x, 1, 'right')
```

```
ans =
```

$$-\frac{\pi}{2}$$

De linker- en rechterlimiet in $x = 1$ streven respectievelijk niet naar $+\infty$ en $-\infty$, maar naar $\frac{\pi}{2}$ en $-\frac{\pi}{2}$ (zie figuur)

1). Daarna bereken we de limieten in $+\infty$ en $-\infty$:

```
limit(y(x), x, inf)
```

```
ans =
```

$$\frac{\pi}{2}$$

```
limit(y(x), x, -inf)
```

```
ans =
```

$$-\frac{\pi}{2}$$

Dit is ook duidelijk te zien in figuur 2. Als x naar $+\infty$ nadert, nadert $y(x)$ naar $\frac{\pi}{2}$. Als x naar $-\infty$ nadert, nadert $y(x)$ naar $-\frac{\pi}{2}$. De functie $y(x)$ heeft dus twee horizontale asymptoten.

Het volgende puntje is het berekenen van de afgeleide van $y(x)$:

```
dy = diff(y(x), x)
```

```
dy =
```

$$\frac{\frac{2}{(2x-2)^2} + \frac{1}{2}}{\left(\frac{1}{2x-2} - \frac{x}{2} + \frac{1}{2}\right)^2 + 1}$$

```
solve(dy==2, x)
```

```
ans =
```

```
Empty sym: 0-by-1
```

We vinden geen punten waar de afgeleide van y gelijk is aan 2. Om de wiskundige reden te achterhalen proberen we toch de oplossingen te vinden op een numerieke manier.

```
vpasolve(dy == 2, x)
```

```
ans =
```

$$\begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}$$

We krijgen twee waarden, één voor de linkerlimiet in $x = 1$ en één voor de rechterlimiet in $x = 1$. Deze oplossingen kunnen we echter niet aanvaarden omdat ze niet in het domein van de functie liggen.

Oefening 23

Opgave:

Schets de krommen $y = \frac{1}{1+x^2}$ en $y = \frac{x^2}{2}$.

Bepaal de snijpunten en bereken de oppervlakte van het gebied ingesloten door beide krommen. Gebruik verschillende kleuren voor de krommen.

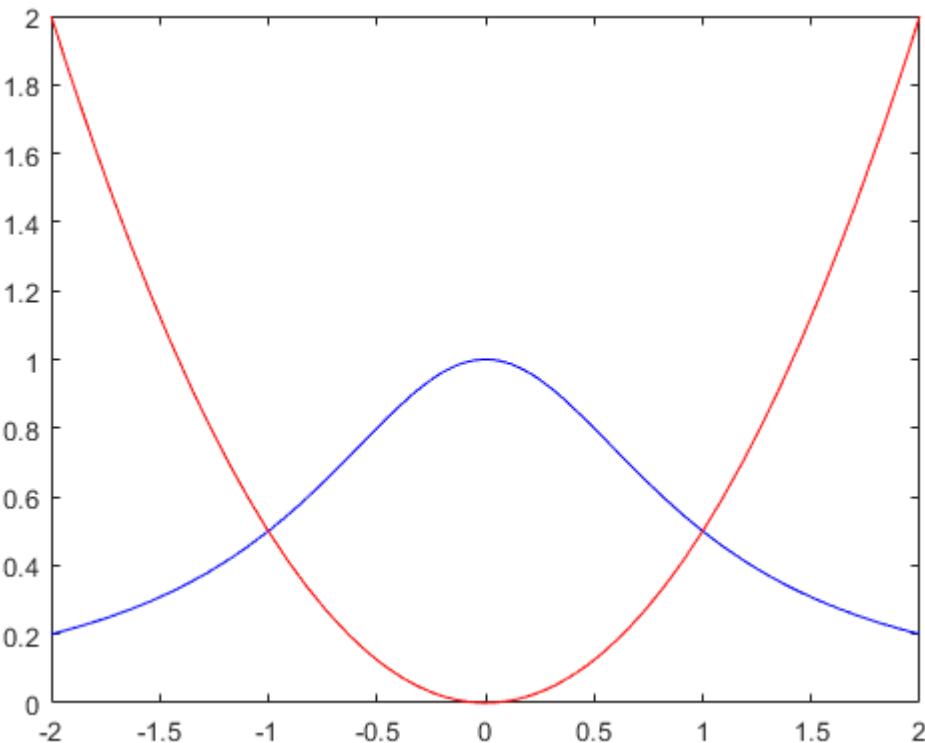
Oplossing:

We kiezen ervoor om met symbolische variabelen te werken in deze oefening omdat we moeten integreren. We defiëren daarvoor het symbool x en de twee functies.

```
syms x;
f1 = @(x) 1/(1+x^2);
f2 = @(x) x^2/2;
```

Dan plotten we de functies. Aangezien we verschillende kleuren moeten gebruiken moeten we een extra argument toevoegen bij het commando fplot. Hier gebruiken we 'b' voor blauw en 'r' voor rood. Met hold on kunnen we meerdere krommen op één grafiek plaatsen.

```
fplot(f1(x), [-2 2], 'b')
hold on
fplot(f2(x), [-2 2], 'r')
```



Op de grafiek zien we dat de krommen twee snijpunten hebben. Hiervan moeten we de x-waarden bepalen omdat dit de grenzen van de integraal zijn. We gebruiken het commando solve:

```
opl = solve(f1(x)==f2(x))
```

```
opl =

$$\begin{pmatrix} -1 \\ 1 \\ -\sqrt{2} i \\ \sqrt{2} i \end{pmatrix}$$

```

We vinden twee reële nulpunten, -1 en 1. Om het oppervlakte van het gebied ingesloten door deze krommen te berekenen maken wij gebruik van een bepaalde integraal met als integrandum het verschil tussen de bovenste en de onderste kromme. De grenzen van deze integraal komen overeen met de twee snijpunten.

```
I = int(f1(x)-f2(x), [-1 1])
```

```
I =

$$\frac{\pi}{2} - \frac{1}{3}$$

```

We kunnen de numerieke waarde berekenen met eval:

```
eval(I)
```

```
ans = 1.2375
```

Oefening 26

Opgave:

Maak een 3D plot en een contourplot van $z = \frac{\sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}$, $-10 \leq x \leq 10$, $-10 \leq y \leq 10$.

Oplossing:

We definiëren de vectoren x en y . Deze zijn respectievelijk verzamelingen van punten op de X- en Y-as.

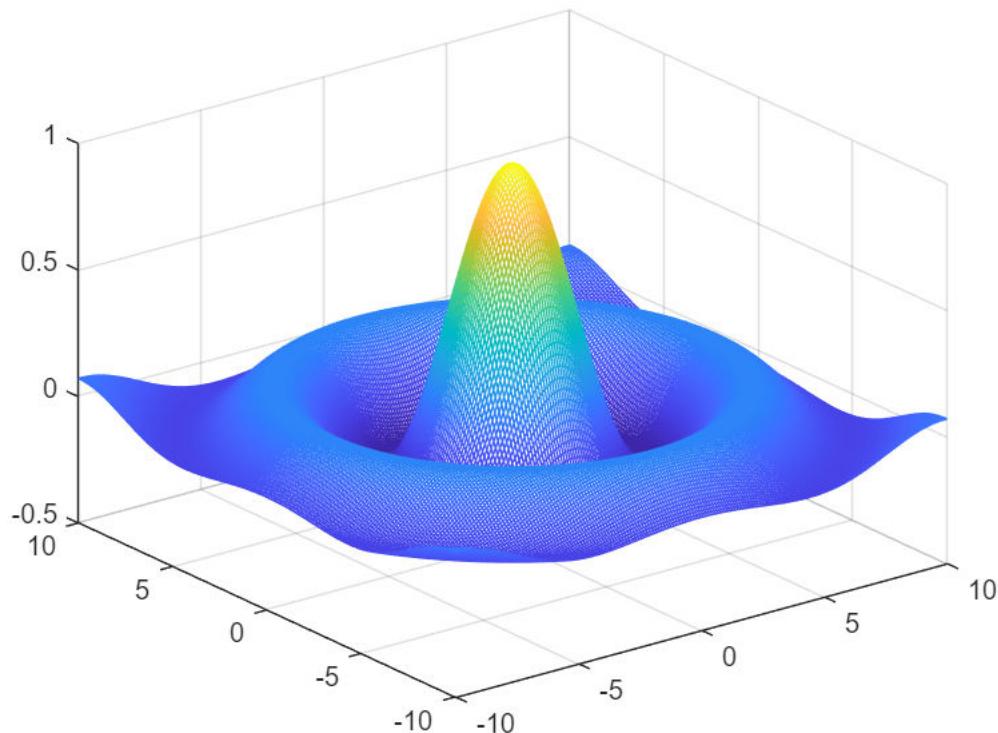
```
x = -10:0.1:10;
y = -10:0.1:10;
[X,Y] = meshgrid(x,y);
```

De veranderlijken X en Y zijn matrices die beide coördinaten bevatten van de roosterpunten (x_i, y_i) , waarbij x_i de elementen zijn van de vector x en y_i de elementen zijn van de vector y . We definiëren nu de matrix Z die de functiewaarden in de roosterpunten aangeeft. We gebruiken hier $.^{\wedge}$ voor elementsgewijze machtsverheffing.

```
Z = sin(sqrt(X.^2+Y.^2))./sqrt(X.^2+Y.^2);
```

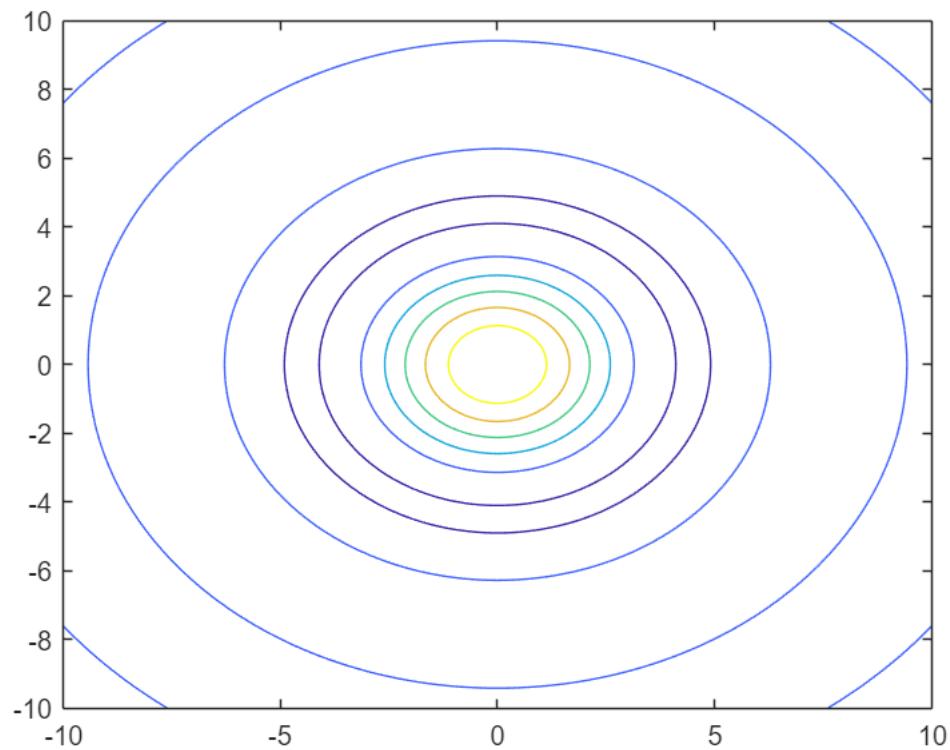
Het commando `mesh` geeft het oppervlak weer in drie dimensies.

```
mesh(X,Y,Z)
```



Contourlijnen zijn een vlakke afdruk van een functie in twee veranderlijken. De contourlijnen of hoogtelijnen verbinden (x,y) -coördinaten met dezelfde z -waarde. Hiervoor gebruiken we het commando contour.

```
contour(X,Y,Z)
```



Oefening 29

Opgave:

Teken een gladde voorstelling van de kromme bepaald door.

$$\begin{cases} x = \cos(5t + \pi/4) \\ y = \sin(4t) \end{cases}$$

Oplossing:

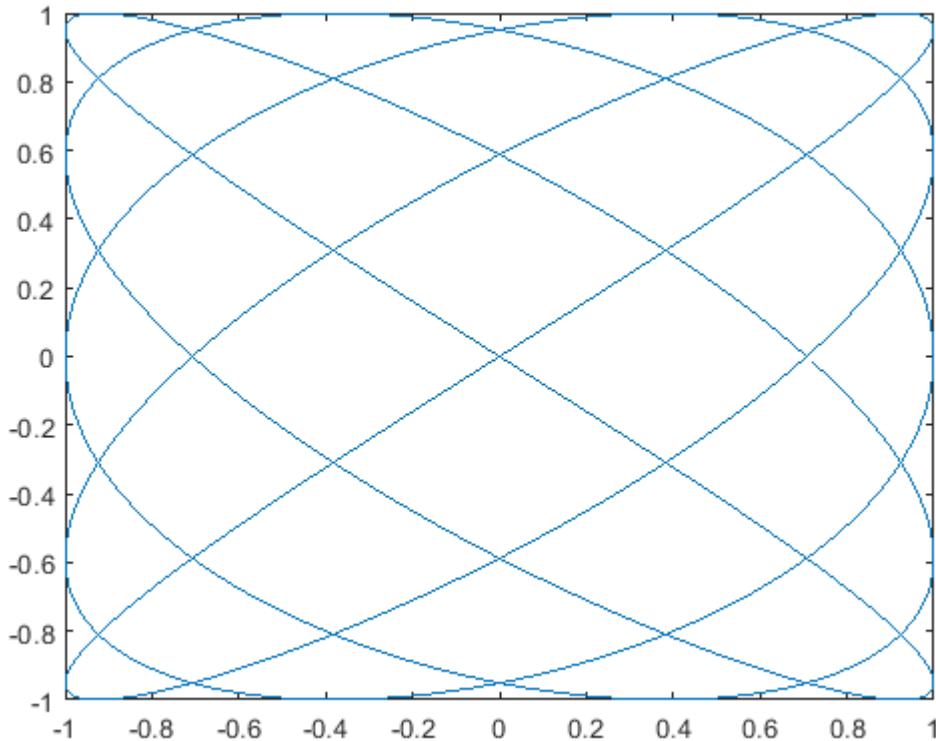
Eerste methode:

We maken een lijst aan van t-waarden:

```
t=0:0.01:2*pi;
```

We plotten de y-waarden tegenover de x-waarden met plot:

```
plot(cos(5*t+pi/4), sin(4*t))
```



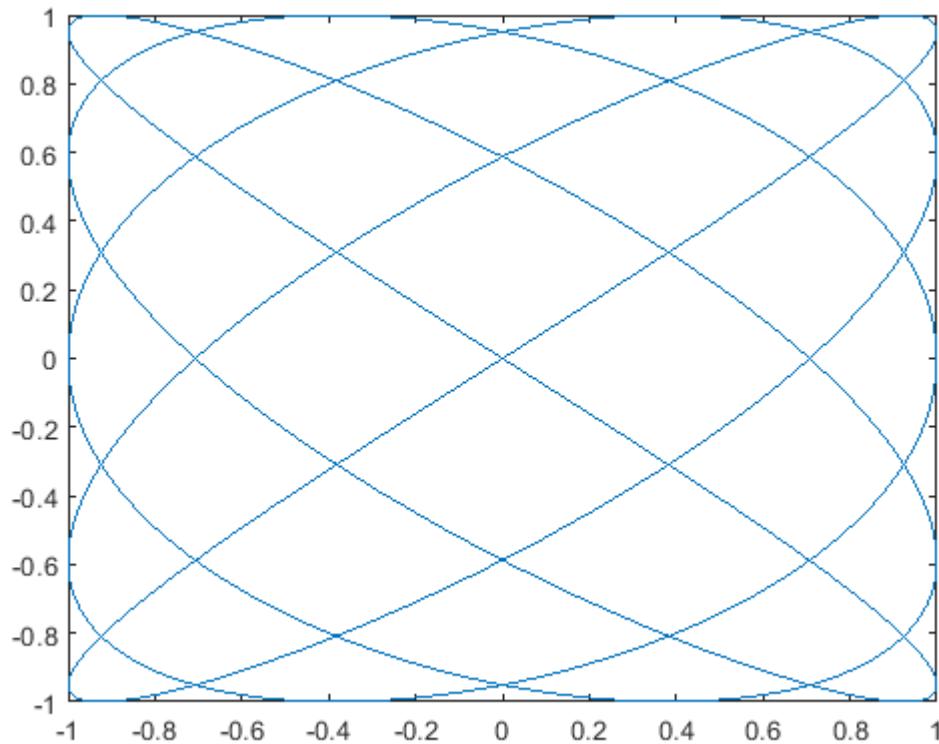
Tweede methode:

In de tweede methode werken we met symbolische variabelen. We beginnen met de symbolische variabele t en de twee functie x en y te definiëren.

```
clear;
syms t;
x = @(t) cos(5*t+pi/4);
y = @(t) sin(4*t);
```

Aangezien we met symbolische variabelen werken gaan we gebruik maken van fplot om deze functies te plotten. In het fplot-commando geven we de grenzen van t.

```
fplot(x(t),y(t), [0, 2*pi])
```



Oefening 30

Opgave:

Teken een kromme in het complexe vlak waarbij $z = t \cdot e^{it}, 0 \leq t \leq 4\pi$.

Oplossing:

We maken een lijst aan van t-waarden:

```
t=0:0.01:4*pi;
```

We definiëren de lijst van overeenkomstige beelden van de gegeven functie. Let op: we gebruiken elementsgewijze vermenigvuldiging (.*)�

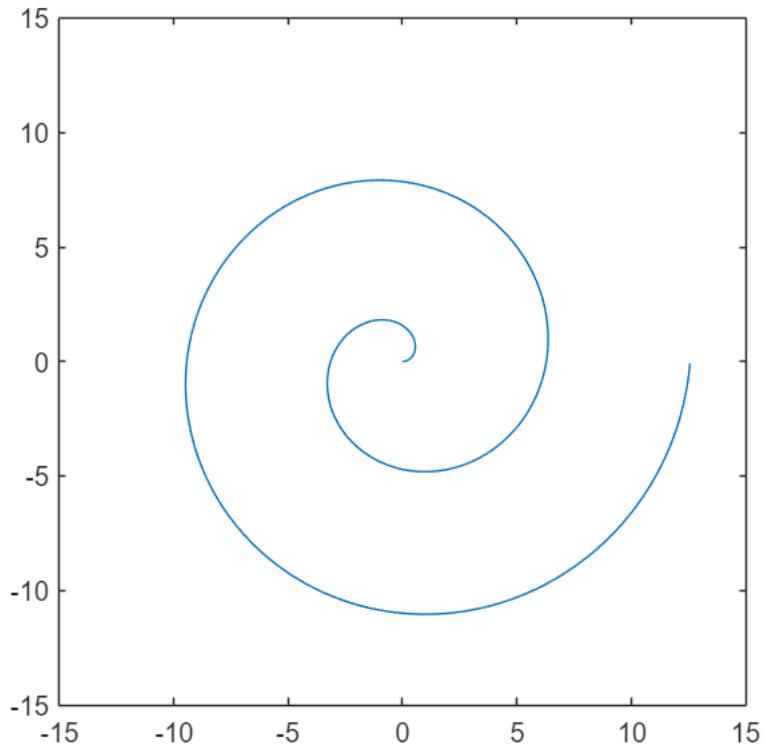
```
z=t.*exp(t*1i);
```

Het vlak waarin we de complexe getallen als koppel voorstellen heet het vlak van Gauss. De horizontale as stelt de reële as voor, terwijl de verticale as doorgaat als imaginaire as. Het reële deel van de z-waarden kunnen we berekenen met `real` en het imaginaire deel met `imag`.

```
plot(real(z), imag(z))
```

We gebruiken `axis equal` om ervoor te zorgen dat de schaal op beide assen dezelfde is. Daarnaast passen we het bereik van beide assen aan zodat de kromme mooi gecentreerd is.

```
axis equal  
axis([-15 15 -15 15])
```



Oefening 33

Opgave:

Bepaal de reële oplossingen van $\begin{cases} x^2 - 2x + y^2 = 0 \\ 9x^2 - 18x + 4y^2 = 27 \end{cases}$. Verklaar grafisch het resultaat.

Oplossing:

We definiëren x en y als symbolische variabelen en analoog f1 en f2 als uitdrukkingen in x en y.

```
syms x y
f1 = x^2-2*x+y^2;
f2 = 9*x^2-18*x+4*y^2;
```

We zoeken symbolisch naar oplossingen van het stelsel:

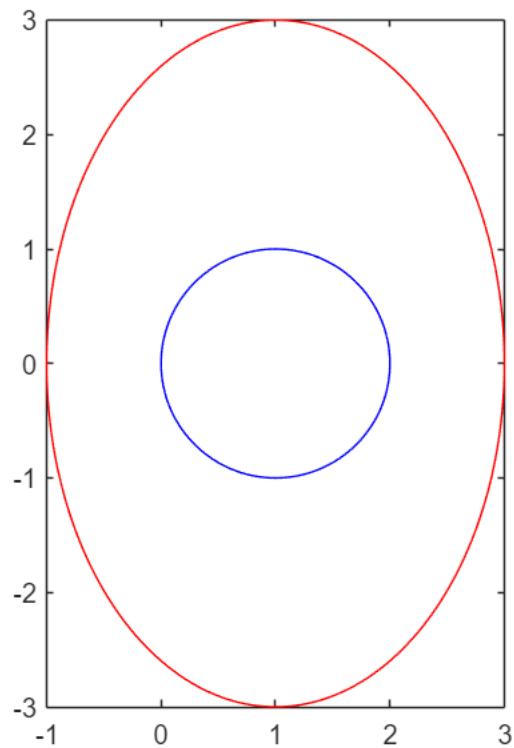
```
[x, y] = solve([f1==0, f2==27])
```

$$x = \begin{pmatrix} 1 - \frac{4\sqrt{10}}{5} \\ \frac{4\sqrt{10}}{5} + 1 \\ 1 - \frac{4\sqrt{10}}{5} \\ \frac{4\sqrt{10}}{5} + 1 \end{pmatrix}$$

$$y = \begin{pmatrix} -\frac{3\sqrt{15}i}{5} \\ -\frac{3\sqrt{15}i}{5} \\ \frac{3\sqrt{15}i}{5} \\ \frac{3\sqrt{15}i}{5} \end{pmatrix}$$

We stellen vast dat er enkel complexe oplossingen zijn. Om dit grafisch te gaan verklaren kunnen we de twee krommen gaan voorstellen op dezelfde grafiek door gebruik te maken van hold on en twee verschillende kleuren. In beide gevallen hebben we te maken met impliciete krommen. Hiervoor hebben we een specifiek commando nodig, namelijk fimplicit. Om te zorgen dat beide assen dezelfde indeling hebben gebruiken we het commando axis equal.

```
fimplicit(f1==0, 'b')
hold on
fimplicit(f2==27, 'r')
axis equal
```



In de grafiek is te zien dat de twee krommen (een cirkel en een ellips) elkaar niet snijden. Dit komt overeen met het eerdere resultaat dat er geen reële oplossingen zijn.

Oefening 35

Opgave:

Maak met behulp van een tekening een schatting voor welke waarden van t de voerstraal van de punten van de kromme met parametervoorstelling $\begin{cases} x(t) = 2 \sin(t) + \cos(2t) \\ y(t) = -2 \cos(t) - \sin(2t) \end{cases}$ maximaal zal zijn en controleer via berekeningen.

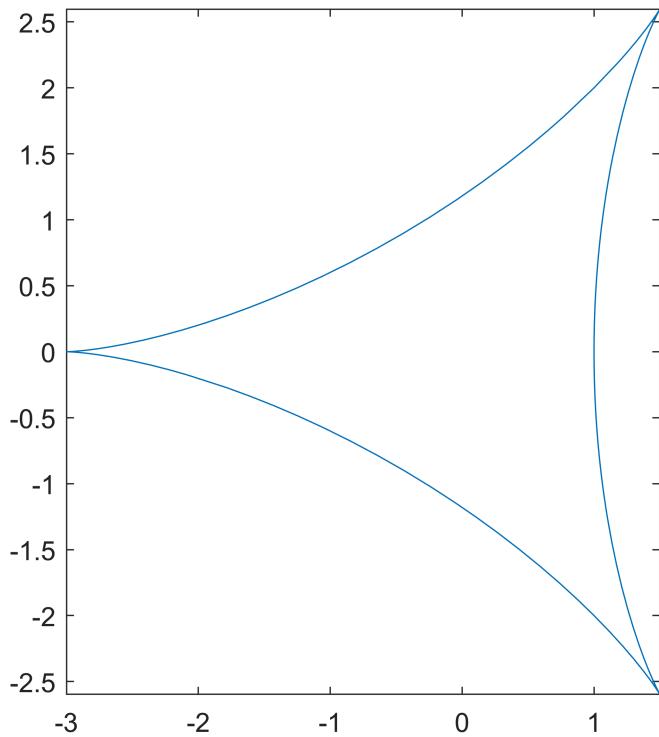
Oplossing:

We definiëren de variabele t als een symbool en de functies $x(t)$ en $y(t)$ door middel van $@(t)$.

```
syms t
x = @(t) 2*sin(t)+cos(2*t);
y = @(t) -2*cos(t)-sin(2*t);
```

We tekenen de gegeven parameterkromme. We beperken ons tot waarden van t tussen 0 en 2π want de periode van deze parameterkromme is 2π . We gebruiken `axis equal` om ervoor te zorgen dat de verdeling van de twee assen hetzelfde is.

```
fplot(x,y, [-pi, pi])
axis equal
```



De voerstraal is gelijk aan $\sqrt{x(t)^2 + y(t)^2}$. We kunnen de voerstraal eveneens definiëren als een functie en zoeken naar de waardes van t zodat de voerstraal maximaal is. De kandidaat extrema vinden we door de

eerste orde afgeleide van de voerstraal gelijk te stellen aan nul. Daarna moeten we de functiewaarden van deze extrema nog eens controleren om enkel de maxima er uit te halen.

We definiëren de functie $r(t)$ door middel van een $@(t)$. Opmerking: om ervoor te zorgen dat we straks alle kanidaat extrema tegelijk kunnen controleren is het het best om met een elementsgewijze machtsverheffing te werken (.^).

```
r = @(t) sqrt(x(t).^2+y(t).^2);
```

We zoeken naar de punten waar de afgeleide gelijk is aan nul. Let op: omdat de functie periodiek is zijn er eigenlijk oneindig veel oplossingen voor dit probleem. We gaan ons dus beperken tot één periode van de functie. Deze beperking moeten we expliciet aan Matlab meegeven, anders zal Matlab maar een beperkt aantal oplossingen geven.

Zonder beperking:

```
opl=solve(diff(r(t),t)==0, t)
```

opl =

$$\begin{pmatrix} \frac{\pi}{2} \\ \frac{\pi}{6} \\ \frac{5\pi}{6} \end{pmatrix}$$

Met beperking:

```
opl=solve([diff(r(t),t)==0, (-pi<=t) , (t<=pi)], t) %Explicit vragen naar oplossingen binnen e
```

opl =

$$\begin{pmatrix} -\frac{\pi}{2} \\ \frac{\pi}{2} \\ -\frac{\pi}{6} \\ \frac{\pi}{6} \\ -\frac{5\pi}{6} \\ \frac{5\pi}{6} \end{pmatrix}$$

Matlab geeft zowel minima als maxima. Het is aan ons om nog eens te controleren welke de maxima zijn.

```
r_opl = r(opl)
```

```
r_opl =
```

$$\begin{pmatrix} 3 \\ 1 \\ 1 \\ 3 \\ 1 \\ 3 \end{pmatrix}$$

Hieruit kan afgeleid worden dat er drie t-waarden overeenkomen met een minmale voerstraal gelijk aan 1. En drie t-waarden overeenkomen met een maximale voerstraal gelijk aan 3.

De drie t-waarden waarvoor de voerstraal maximaal is, zijn $\frac{\pi}{6}$, $\frac{5\pi}{6}$ en $-\frac{\pi}{2}$.

We kunnen de corresponderende x- en y-waarden bepalen:

$$x_{opl} = x(opl)$$

$$x_{opl} =$$

$$\begin{pmatrix} -3 \\ 1 \\ -\frac{1}{2} \\ \frac{3}{2} \\ -\frac{1}{2} \\ \frac{3}{2} \end{pmatrix}$$

$$y_{opl} = y(opl)$$

$$y_{opl} =$$

$$\begin{pmatrix} 0 \\ 0 \\ -\frac{\sqrt{3}}{2} \\ -\frac{3\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} \\ \frac{3\sqrt{3}}{2} \end{pmatrix}$$

Oefening 39

Opgave:

1. Teken op één grafiek de krommen met vergelijking $r = 1$ en $r = 2 \cos(3\theta)$
 2. Bereken de oppervlakte van het gemeenschappelijk deel binnen $r = 1$ en $r = 2 \cos(3\theta)$

Oplossing:

1. We beginnen met een lijst aan te maken van θ -waarden:

```
tt=0:0.01:2*pi;
```

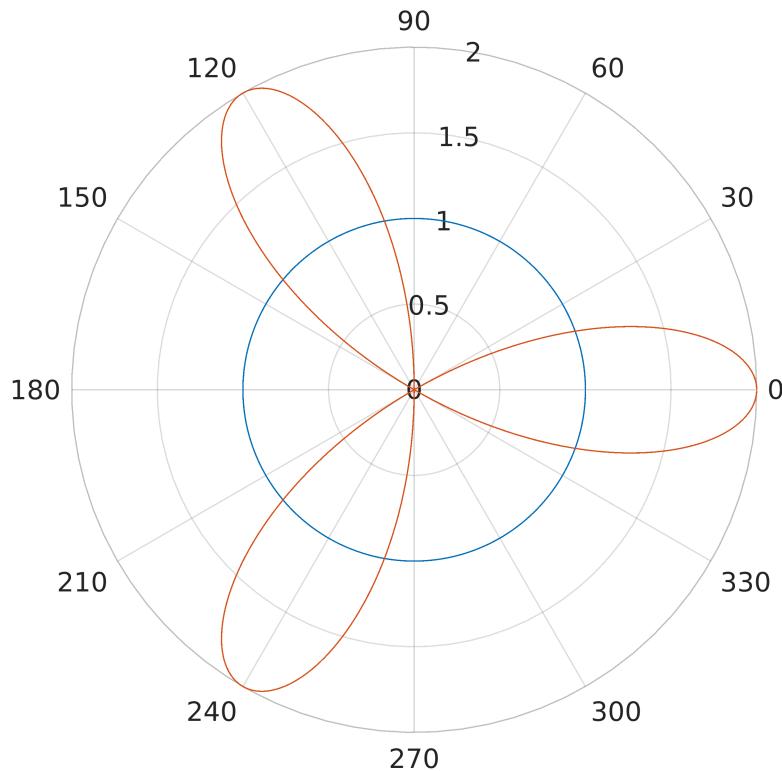
De overeenkomstige lijsten met r -waarden van de twee krommen berekenen we als volgt. Voor de eerste kromme, namelijk $r = 1$, gebruiken we `ones` om een lijst met 1-waarden te maken. Hierbij moeten we opletten met de dimensies van `ones`. We willen namelijk een vector die evenveel componenten heeft als vector `tt`. Omdat we een rijvector (en geen kolomvector) willen moeten we als eerste argument `1` en als tweede argument `length(tt)` gebruiken (één rij, meerdere kolommen). Voor de tweede kromme moeten we rekening houden met het domein (het domein van een poolkromme vind je door na te gaan wanneer $r(\theta) \geq 0$). Om de waarden met negatieve straal weg te werken kunnen we het commando `max` gebruiken. Op die manier stellen we de functiewaarde gelijk aan nul als deze negatief zou zijn.

```
r1 = ones(1, length(tt))
```

```
r2 = max( 0, 2*cos( 3*tt ) );
```

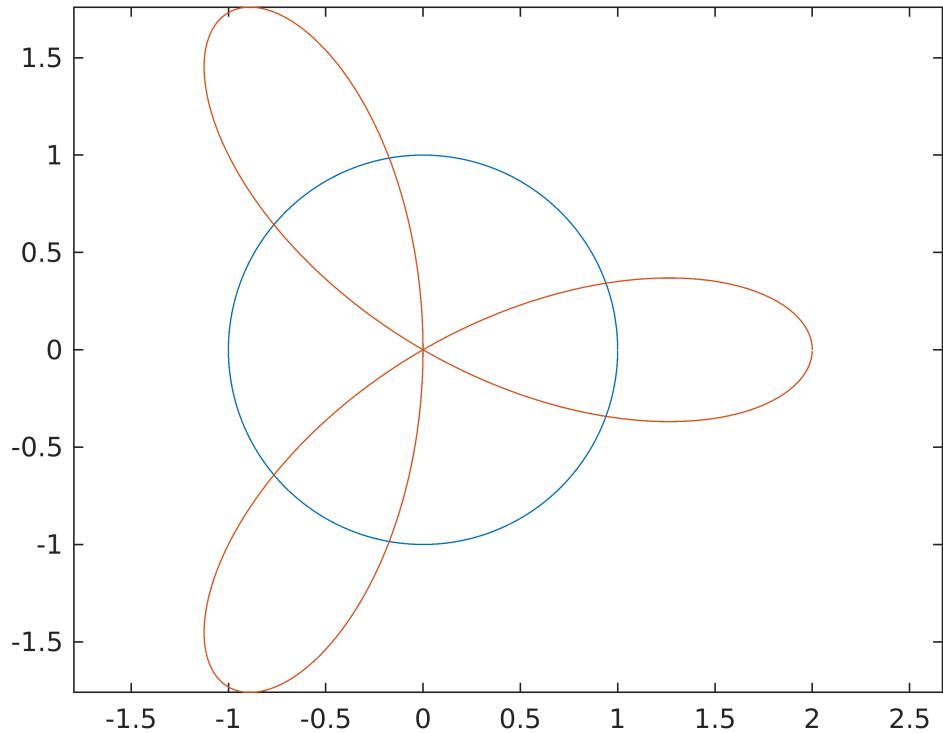
Plot in poolcoördinaten, met het commando `polarplot`. Het eerste argument is de lijst met θ -waarden, het tweede argument is de lijst met r-waarden:

```
polarplot(tt, r1)
hold on
polarplot(tt, r2)
hold off
```



We kunnen ook het commando `plot` gebruiken maar dan moeten we eerst de poolkromme omzetten naar cartesische coördinaten. Hiervoor gebruiken we de transformatieregels $\begin{cases} x = r \cos(\theta) \\ y = r \sin(\theta) \end{cases}$:

```
plot(r1.*cos(tt), r1.*sin(tt))
hold on
plot(r2.*cos(tt), r2.*sin(tt))
axis equal
```



2. Het is een rozet met drie, even grote lussen. Elke lus heeft een symmetrie-as. De eerste lus is symmetrisch ten opzichte van de poolas, we kunnen ons dus beperken in onze berekening tot enkel het bovenste stuk van de eerste lus. Dit stuk bestaat uit twee delen, s_1 en s_2 . Het eerste gaat van de poolas tot het snijden met de cirkel. Het tweede stuk gaan van het snijpunt met de cirkel tot het punt waar de rozet door de pool gaat ($r(\theta) = 0$).

Om deze grenzen te bepalen moeten we dus de punten vinden waar de voerstraal van de rozet gelijk is 1 of 0. Hiervoor maken we gebruik van de symbolische methode `solve`. We beperken ons tot het eerste kwadrant.

```
syms t
t2 = solve([1==2*cos(3*t), 0<=t , t<=pi/2])
```

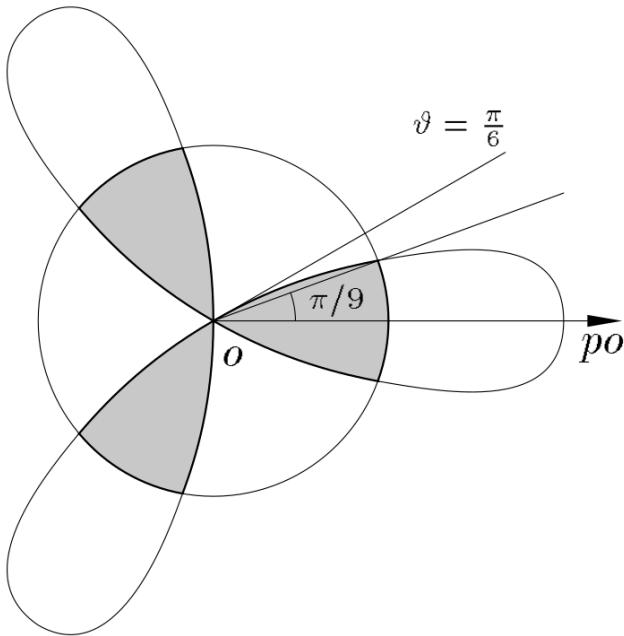
```
t2 =
```

$$\frac{\pi}{9}$$

```
t3 = solve([0==2*cos(3*t), 0<=t , t<=pi/2])
```

```
t3 =
```

$$\frac{\pi}{6}$$



Nu kunnen de oppervlaktes s_1 en s_2 berekend worden. Hieruit kan de totale oppervlakte berekend worden ($6*(s_1+s_2)$). De oppervlakte onder een poolkromme is: $S = \frac{1}{2} \int_{\theta_1}^{\theta_2} r^2 d\theta$.

Let op: het integrandum moet een functie zijn afhankelijk van het symbool t . Hiervoor schrijven we bij het eerste oppervlak $1+0*t$ als integrandum in de plaats van gewoon 1.

```
s1 = (0.5*int(1+0*t, 0, pi/9));
s2 = (0.5*int((2*cos(3*t))^2, pi/9, pi/6));
oppervlakte = 6*(s1+s2)
```

```
oppervlakte =
```

$$\frac{2\pi}{3} - \frac{\sqrt{3}}{2}$$

Oefening 43

Opgave:

Teken de poolkromme $r = \sin(4\theta)$.

Oplossing:

We beginnen met een lijst van θ -waarden gaande van 0 tot 2π met een spaciëring van 0.01.

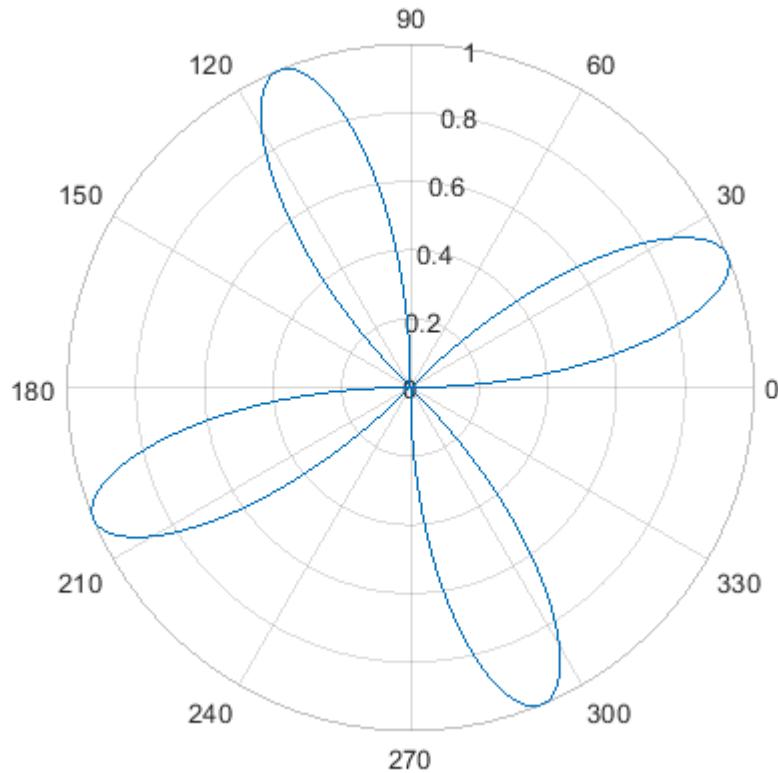
```
theta=0:0.01:2*pi;
```

We bepalen de lijst met functiewaarden r . Hier moeten we rekening houden met het domein van een poolkromme, namelijk de waarden van θ waarvoor $r(\theta) \geq 0$. Om de waarden met negatieve straal weg te werken kunnen we het commando `max` gebruiken. Op die manier stellen we de functiewaarde gelijk aan nul als deze negatief zou zijn.

```
r = max(0, sin(4*theta));
```

Het plotten van de poolkromme kunnen we doen met het commando `polarplot`:

```
polarplot(theta, r)
```



Oefening 44

Opgave:

Bereken de booglengte van de kromme $r = 1 - \cos(\theta)$ gelegen buiten de kromme $r + \sqrt{3} \sin(\theta) = 0$.

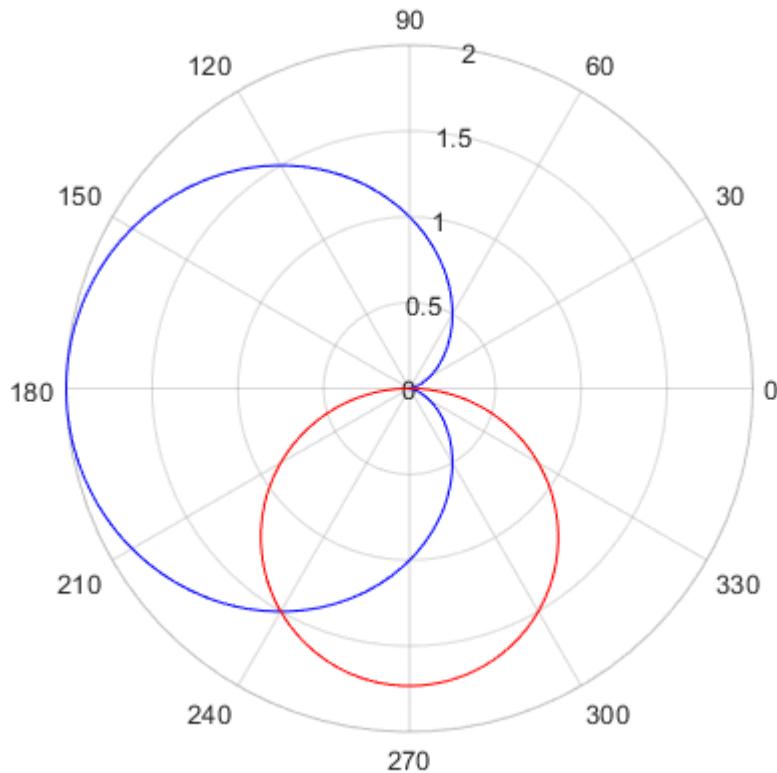
Oplossing:

Bij dit soort oefeningen is het best van eerst een grafische voorstelling van het probleem te maken. Om de twee poolkrommen te plotten maken we eerst een lijst van θ -waarden gaande van 0 tot 2π . Daarna maken we een lijst van functiewaarden voor de twee krommen; bij deze stap maken we gebruik van `max` zodat we negatieve waarden kunnen gelijk stellen aan nul.

```
theta=0:0.01:2*pi;
r1=max(0,1-cos(theta));
r2=max(0,-sqrt(3)*sin(theta));
```

Het plotten in poolcoördinaten doen we aan de hand van `polarplot`. De eerste kromme plotten we in het blauw en de tweede in het rood.

```
polarplot(theta, r1, 'b')
hold on
polarplot(theta, r2, 'r')
hold off
```



De booglengte van een poolkromme is gelijk aan $\int_{\theta_1}^{\theta_2} \sqrt{r'^2 + r^2} d\theta$. In ons geval is de kromme in kwestie de cardioïde $r = 1 - \cos(\theta)$. Nu zoeken we het snijpunt van de twee krommen, zodanig dat we de bovengrens van de intergraal kennen. De ondergrens is gelijk aan $\theta = 0$. Aangezien we weten dat het snijpunt in het derde kwadrant ligt beperken we ons tot het interval $[\pi, \frac{3\pi}{2}]$.

```
syms t
opl = solve([1-cos(t)==-sqrt(3)*sin(t), (pi<=t) & (t<=(3/2)*pi)], t)
```

```
opl =
```

$$2\pi - \log\left(-\frac{1}{2} - \frac{\sqrt{3}}{2}i\right)i$$

De oplossing die we vinden lijkt op een complex getal. Als we echter de numerieke waarde van dit getal opvragen met eval dan vinden we dat dit een zuiver reëel getal is.

```
eval(opl)
```

```
ans = 4.1888 + 0.0000i
```

Nu we de bovengrens en ondergrens kennen, kunnen we de booglengte van het deel van de cardioïde berekenen:

```
I = int(sqrt((1-cos(t))^2+diff(1-cos(t), t)^2), 0, opl)
```

```
I = 6
```

Oefening 45

Opgave:

1. Teken $f(x) = e^{2x+1}$ samen met zijn 2 afgekapte Taylorontwikkelingen rond $x = 1$, resp. na de term in x en na de term in x^2 . Gebruik daarvoor het commando taylor.
2. Herhaal deze tekening voor $f(x) = x^2 - x$. Leg het resultaat uit.

Oplossing:

We definiëren het symbool x en de functie f :

```
syms x
f=exp(2*x+1);
```

Een Taylorreeks kunnen we berekenen met het commando taylor. Hiervoor gebruiken we volgende argumenten: de functie die we willen benaderen, de variabele van de ontwikkeling en het punt waarrond we willen ontwikkelen. Als extra argument gebruiken we Order; we moeten $n+1$ opgeven om de n -de orde Taylorontwikkeling te krijgen.

```
tay2 = taylor(f, x, 1, 'Order', 2)
```

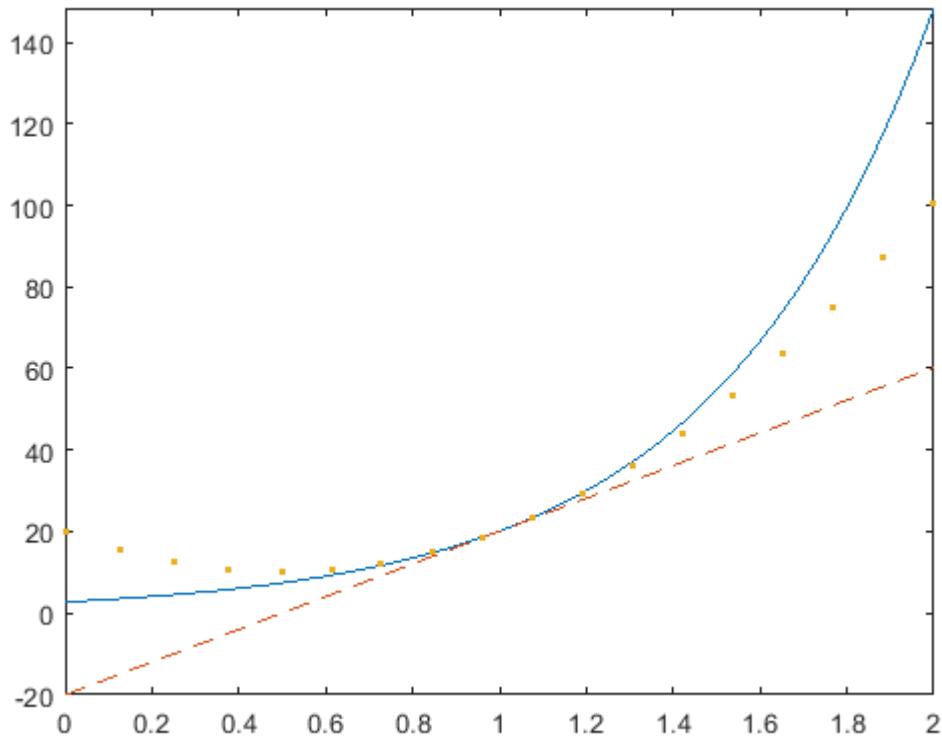
```
tay2 = e^3 + 2 e^3 (x - 1)
```

```
tay3 = taylor(f, x, 1, 'Order', 3)
```

```
tay3 = e^3 + 2 e^3 (x - 1) + 2 e^3 (x - 1)^2
```

Aangezien we op een symbolische manier werken gebruiken we het commando fplot. Om meerdere grafieken te plotten gebruiken we hold on, we geven elke curve ook een ander type lijn.

```
fplot(f, [0,2])
hold on
fplot(tay2, [0,2], '--')
fplot(tay3, [0,2], '.')
hold off
```



Zoals we kunnen zien in de tekening, hoe groter de orde van de benadering, hoe nauwkeuriger de benadering.

We passen dezelfde methode toe op de functie $f(x) = x^2 - x$:

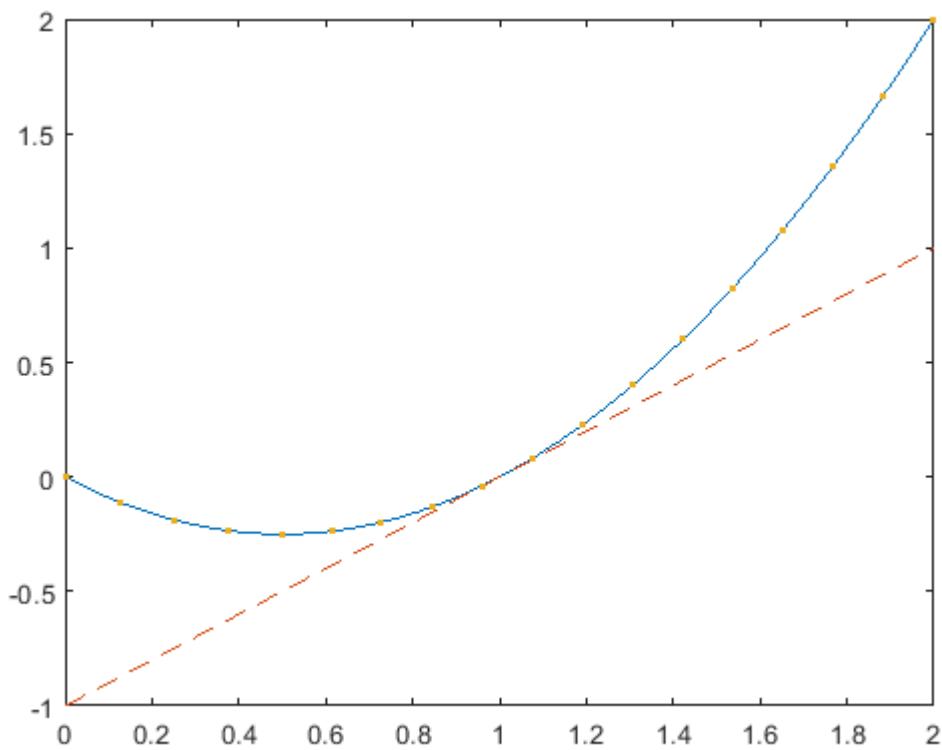
```
syms x
f=x^2-x;
tay2 = taylor(f, x, 1, 'Order', 2)
```

```
tay2 = x - 1
```

```
tay3 = taylor(f, x, 1, 'Order', 3)
```

```
tay3 = x + (x - 1)^2 - 1
```

```
fplot(f, [0,2])
hold on
fplot(tay2, [0,2], '--')
fplot(tay3, [0,2], '.')
```



Hier kunnen we zien dat de tweede orde Taylorontwikkeling van de veelterm van de tweede orde gelijk is aan die veelterm. Op de grafiek vallen die twee krommen dan ook samen. Als we de gevonden ontwikkeling bewerken krijgen we de oorspronkelijk veelterm terug:

```
expand(simplify(tay3))
```

```
ans = x^2 - x
```

Oefening 46

Opgave:

Bepaal het inverse Laplacebeeld van $F(s) = \frac{2s^3 + 8s^2 + 12s + 32}{s^4 + 4s^3 + 8s^2 + 16s + 16}$.

Oplossing:

We beginnen met het aanmaken van de twee symbolen s en t . Daarna definiëren we de functie uit de opgave.

```
syms s t
F= @(s) (2*s^3+8*s^2+12*s+32)/(s^4+4*s^3+8*s^2+16*s+16);
f = @(t) ilaplace(F(s),s,t);
f(t)
```

```
ans =
2 e^-2 t + sin(2 t)/2 + 3 t e^-2 t
```

Let op: de methode `ilaplace` van Matlab houdt geen rekening met negatieve t -waarden en gaat ervan uit dat de originele functie $f(t)$ enkel gedefinieerd is voor $t \geq 0$. Om hiervoor te corrigeren moeten we de verkregen functie $f(t)$ nog eens vermenigvuldigen met een Heavisidefunctie `heaviside(t)`. Met andere woorden gaan we verkregen functie causaal maken.

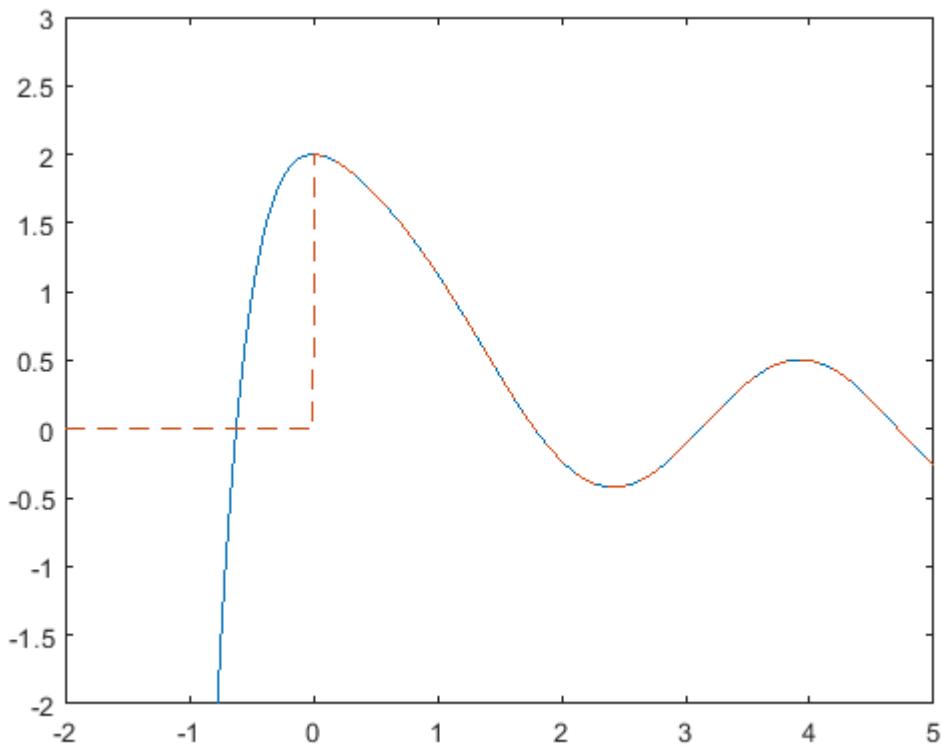
```
f_causaal = @(t) f(t)*heaviside(t)
```

```
f_causaal = function_handle with value:
@(t)f(t)*heaviside(t)
```

```
f_causaal(t)
```

```
ans =
heaviside(t) (2 e^-2 t + sin(2 t)/2 + 3 t e^-2 t)
```

```
fplot(f(t))
hold on
fplot(f_causaal(t), "--")
axis([-2 5 -2 3])
```



Opmerking: je hoeft niet per se met een functie aan de hand van $f(t)$ te werken. Dit kan ook gewoon een uitdrukking zijn.

```
clear;
syms s t
F = (2*s^3+8*s^2+12*s+32)/(s^4+4*s^3+8*s^2+16*s+16);
f = ilaplace(F,s,t)
```

```
f =  
2 e-2 t + sin(2 t)  
----- + 3 t e-2 t  
2
```

```
f_causaal = f*heaviside(t)
```

```
f_causaal =  
heaviside(t)  $\left(2 e^{-2 t} + \frac{\sin(2 t)}{2} + 3 t e^{-2 t}\right)$ 
```

Oefening 48

Opgave:

Bereken het Laplacebeeld van de functie die niet nul is tussen 1 en 2 en daar als functiewaarde $2 - x$ aanneemt. Bereken van het resultaat het inverse Laplacebeeld en maak een tekening van de functie.

Oplossing:

We beginnen met de symbolen x en s aan te maken.

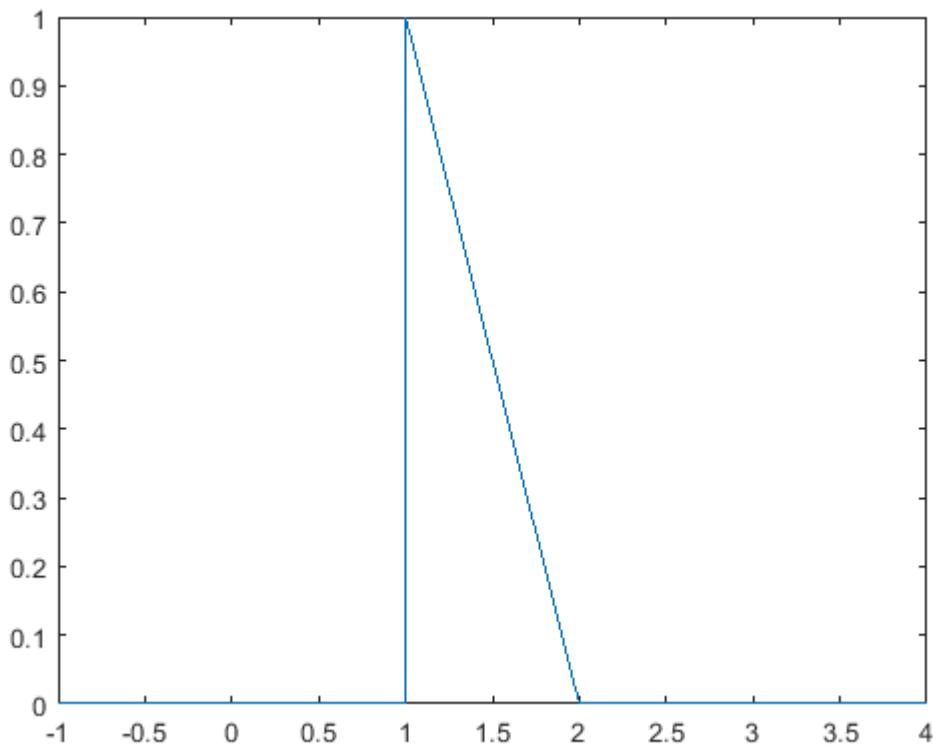
```
syms x s
```

De functie uit de opgave kunnen we als volgt voorstellen: $f(x) = \begin{cases} 0 & \text{als } x < 1 \\ 2 - x & \text{als } 1 < x < 2 \\ 0 & \text{als } x > 2 \end{cases}$

We kunnen deze functie definiëren met behulp van de Heaviside functie. Anders is deze functie niet causaal.

Er zijn twee punten van discontinuïteit. Wat betekent dat we twee keer de Heavisidefunctie zullen moeten gebruiken. Op elk punt van discontinuïteit gaan we de bijhorende Heavisidefunctie vermenigvuldigen met het verschil van de functie aan de rechterkant van de discontinuïteit en de functie aan de linkerkant van de discontinuïteit. Wat we krijgen is als volgt:

```
f = @(x) heaviside(x-1)*(-0 + (2-x)) + heaviside(x-2)*(-1*(2-x) + 0);  
fplot(f(x),[-1,4])
```



Let op: op de grafiek worden de punten van discontinuïteit ook geplot.

Het Laplacebeeld berekenen we met de methode `laplace`:

```
g = laplace(f(x),x,s)
```

$$g = \frac{e^{-2s}}{s^2} + \frac{e^{-s}(s-1)}{s^2}$$

Het inverse Laplacebeeld berekenen we met de methode `ilaplace`:

```
f_ = ilaplace(g,s,x)
```

$$f_ = \text{heaviside}(x-1) - \text{heaviside}(x-1)(x-1) + \text{heaviside}(x-2)(x-2)$$

```
simplify(f_)
```

$$\text{ans} = -(\text{heaviside}(x-1) - \text{heaviside}(x-2)) (x-2)$$

Oefening 51

Opgave:

Bepaal alle waarden van a zodat $|A| = -1$ met $A = \begin{pmatrix} 1 & a & 0 & 6 \\ 4 & 0 & 1 & 2 \\ 1 & 3 & -1 & 6 \\ 1 & 10 & 3 & 11 \end{pmatrix}$.

Oplossing:

We beginnen met het symbool a aan te maken en een functie te definiëren die determinant van de matrix voorstelt.

```
syms a;
f = @(a) det([1 a 0 6; 4 0 1 2; 1 3 -1 6; 1 10 3 11]);
```

```
ans = 113 a - 433
```

Eerste mogelijkheid: Met `solve` zoeken we de waarde van a waarvoor de functie -1 is.

```
opl = solve(f(a)==-1)
```

```
opl =
432
113
```

```
eval(opl)
```

```
ans = 3.8230
```

Tweede mogelijkheid: We brengen -1 naar het ander lid en krijgen zo $113a - 432 = 0$. Op die manier kunnen we het commando `roots` gebruiken om naar oplossingen te zoeken.

```
roots([113, -432])
```

```
ans = 3.8230
```

Oefening 52

Opgave:

Een oefening uit de cursus elektriciteit geeft aanleiding tot volgend stelsel:

$$\begin{cases} i_2 - i_1 - i_3 = 0 \\ i_3 - i_4 - i_5 = 0 \\ 30i_1 + 30i_2 = -60 \\ 30i_2 + 15i_3 + 30i_5 = 60 \\ -30i_5 + 10i_4 = 200 \end{cases}$$

Bepaal de stroomsterkten.

Oplossing:

Eerste methode: We maken de matrix A met de coëfficiënten van het stelsel en de kolomvector B met de waarden in de rechterleden. Dan kunnen we het stelsel voorstellen als $A \cdot C = B$ waarbij C de kolomvector van de juiste stroomsterktes voorstelt. Deze matrixvergelijking kunnen oplossen door vector B te "delen" door de matrix A aan de hand van $A \setminus B$.

```
A = [-1 1 -1 0 0; 0 0 1 -1 -1; 30 30 0 0 0; 0 30 15 0 30; 0 0 0 10 -30]
```

```
A = 5x5
-1    1    -1    0    0
 0    0     1   -1   -1
 30   30     0    0    0
 0    30    15    0   30
 0    0     0   10  -30
```

```
B = [0; 0; -60; 60; 200]
```

```
B = 5x1
 0
 0
-60
 60
 200
```

```
A\B
```

```
ans = 5x1
-4.2000
 2.2000
 6.4000
 9.8000
-3.4000
```

Tweede methode: Deze methode maakt gebruik van het Gauss-Jordan algoritme. We maken de uitgebreide coëfficiëntenmatrix D aan die het stelsel voorstelt (namelijk de matrix die ontstaat door aan de matrix A de kolomvector B toe te voegen). Met het commando rref kunnen we deze matrix in de gereduceerde echelonvorm krijgen, dan kunnen we de oplossingen gemakkelijk aflezen.

```
D = [-1 1 -1 0 0 0; 0 0 1 -1 -1 0; 30 30 0 0 0 -60; 0 30 15 0 30 60; 0 0 0 10 -30 200]
```

```
D = 5x6
-1    1    -1    0    0    0
```

```

0      0      1      -1     -1      0
30     30      0      0      0     -60
0      30     15      0     30      60
0      0      0     10    -30     200

```

```
rref(D)
```

```

ans = 5x6
1.0000      0      0      0      0     -4.2000
0    1.0000      0      0      0      2.2000
0      0    1.0000      0      0      6.4000
0      0      0    1.0000      0      9.8000
0      0      0      0    1.0000     -3.4000

```

Derde methode: We kunnen gebruik maken van de solve functie om het lineaire stelsel direct op te lossen (zonder matrices). Hiervoor moeten we eerst de vijf variabelen als symbolen definiëren:

```
syms i_1 i_2 i_3 i_4 i_5
```

We kunnen de oplossing symbolisch vinden met solve of numeriek met vpasolve:

```
opl = solve([i_2-i_1-i_3 == 0, i_3 - i_4 - i_5 ==0, 30*i_1 + 30*i_2 == -60, 30*i_2 + 15*i_3 + 30*i_4 == 200])
```

```
opl = struct with fields:
i_1: -21/5
i_2: 11/5
i_3: 32/5
i_4: 49/5
i_5: -17/5
```

```
opl = vpasolve([i_2-i_1-i_3 == 0, i_3 - i_4 - i_5 ==0, 30*i_1 + 30*i_2 == -60, 30*i_2 + 15*i_3 + 30*i_4 == 200])
```

```
opl = struct with fields:
i_1: -4.2
i_2: 2.2
i_3: 6.4
i_4: 9.8
i_5: -3.4
```

Oefening 55

Opgave:

Bepaal een orthonormaal stel eigenvectoren van de lineaire transformatie $g : (x, y, z) \rightarrow (-x, -y, -z)$. Wat is de fysische interpretatie hiervan? Verklaar het resultaat.

Oplossing:

De lineaire transformatie kunnen we voorstellen als een matrixvermenigvuldiging: $\begin{pmatrix} -x \\ -y \\ -z \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$

$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$. In deze matrix is elke kolom het beeld van de basisvectoren onder g . We voeren de transformatiematrix in Matlab in onder de naam T.

```
T = [-1 0 0; 0 -1 0; 0 0 -1]
```

```
T = 3x3
-1      0      0
 0      -1      0
 0       0     -1
```

De eigenvectoren en eigenwaarden kunnen we vinden aan de hand van het commando eig.

```
[V, D] = eig(T)
```

```
V = 3x3
 1      0      0
 0      1      0
 0      0      1
D = 3x3
-1      0      0
 0      -1      0
 0       0     -1
```

De kolommen van de matrix V komen overeen met de eigenvectoren. De elementen op de diagonaal van de matrix D zijn de eigenwaarden. Matlab geeft als resultaat dat er één eigenwaarde is, namelijk $\lambda = -1$ met

multipliciteit 3. Er worden dus drie eigenvectoren gegeven. $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ en $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$.

Alle veelvouden van de bekomen eigenvectoren door Matlab zijn ook eigenvectoren want: indien er geldt dat $g(\vec{v}) = T \cdot \vec{v} = \lambda \vec{v}$ dan geldt ook $g(k \vec{v}) = T \cdot (k \vec{v}) = \lambda k \vec{v}$

Dit betekent dat elke vector van \mathbb{R}^3 een eigenvector is. Elke vector van \mathbb{R}^3 kan namelijk als een lineaire combinatie van de drie eigenvectoren geschreven worden:

$\forall \vec{v} \in \mathbb{R}^3 : \vec{v} = k_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + k_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + k_3 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ met $k_1, k_2, k_3 \in \mathbb{R}$

Deze drie eigenvectoren zijn namelijk lineair onafhankelijk: ze zijn m.a.w. een basis van \mathbb{R}^3 .

Fysische interpretatie: De lineaire transformatie komt overeen met een spiegeling van de vector om de oorsprong. Elke vector van \mathbb{R}^3 is dus een eigenvector, want de lineaire transformatie van elke vector levert een vector op die dezelfde richting heeft. Opmerking: de bekomen vector heeft nog dezelfde grootte, maar heeft een andere zin.

Oefening 58

Opgave:

Maak een simulatie met 100 steekproefwaarden van een variabele met een $F(7, 5\text{df})$ -verdeling. Herhaal dit 400 keer. Maak passende tekeningen om hierin de centrale limietstelling te herkennen.

Oplossing:

Aan de hand van het commando `frnd(7, 5, 100, 400)` kunnen we randomwaarden genereren uit een F-verdeling (Fisher verdeling). 7 is de eerste parameter en staat voor het aantal vrijheidsgraden in de teller, 5 is de tweede parameter en staat voor het aantal vrijheidsgraden in de noemer. De derde en vierde parameter, 100 en 400, zorgen ervoor dat `frnd` een 100×400 matrix genereert met daarin steekproefwaarden uit een F-verdeling. De 400 kolommen in deze matrix staan voor de 400 uitgevoerde experimenten. Elk van deze kolommen bezit dan de 100 steekproefwaarden per experiment.

```
simulatie=frnd(7, 5, 100, 400);
```

`mean` berekent het gemiddelde over de verschillende kolommen en dus per experiment. Dus krijgen we een rijvector met 400 componenten met daarin de gemiddelde waarden per experiment.

```
xbar=mean(simulatie)
```

```
xbar = 1x400  
1.2522    1.4367    1.1856    1.4853    1.7961    1.7265    2.8462    1.5514 ...
```

Met `histogram` maken we een histogram uit de waarden van de `xbar`-vector. Met 'Normalization', 'pdf' zorgen we ervoor dat de totale oppervlakte van de rechthoekjes gelijk is aan 1 (zie pagina 25 cursus statistiek).

```
histogram(xbar, 'Normalization', 'pdf')  
hold on;
```

Aangezien elk experiment 100 willekeurige steekproefwaarden heeft kunnen we gebruik maken van de centrale limietstelling om de bekomen histogram op een theoretische manier te benaderen (zie pagina 55 cursus statistiek). Dit moet als volgt:

Bij deze $F(7, 5\text{df})$ -verdeling is $\mu_0 = \frac{\nu_2}{\nu_2 - 2} = \frac{5}{5 - 2} = \frac{5}{3}$ en

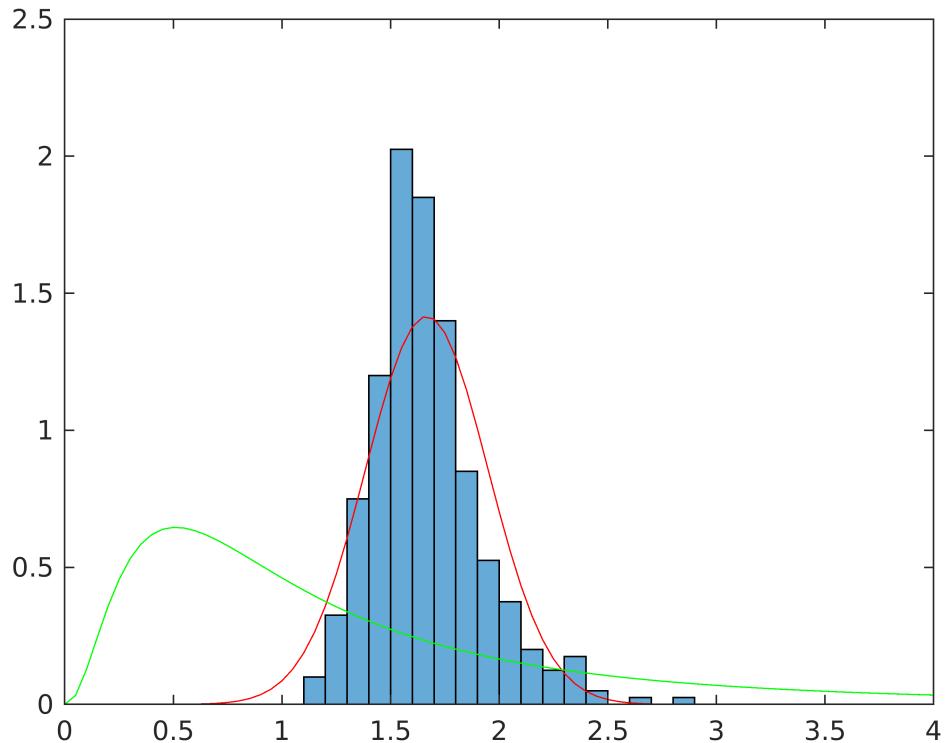
$$\sigma_0^2 = \frac{2\nu_2^2(\nu_1 + \nu_2 - 2)}{\nu_1(\nu_2 - 4)(\nu_2 - 2)^2} = \frac{2 \cdot 25(7 + 5 - 2)}{7(5 - 4)(5 - 2)^2} = \frac{500}{63} \Rightarrow \sigma_0 = \sqrt{\frac{500}{63}} = \frac{10\sqrt{5}}{3\sqrt{7}}.$$

De verdeling van de steekproefgemiddelden kan benaderd worden door een normaalverdeling met gemiddelde $\mu_0 = \frac{5}{3}$ en standaard afwijking $\frac{\sigma_0}{\sqrt{n}} = \frac{\sqrt{5}}{3\sqrt{7}}$. Dit wordt met volgend commando geplot (in het rood):

```
x=0:0.05:4;  
plot(x,normpdf(x,5/3,sqrt(5/7)/3), 'r')  
hold on;
```

De oorspronkelijke F-verdeling waaruit de steekproeven werden gesampled wordt tevens geplot (in het groen):

```
plot(x,fpdf(x,7,5), 'g')
```



Oefening 59

Opgave:

Creeer een nieuwe data-file met volgende gegevens:

24, 19, 14, 10, 7, 5, 6, 8, 12, 16, 21, 27

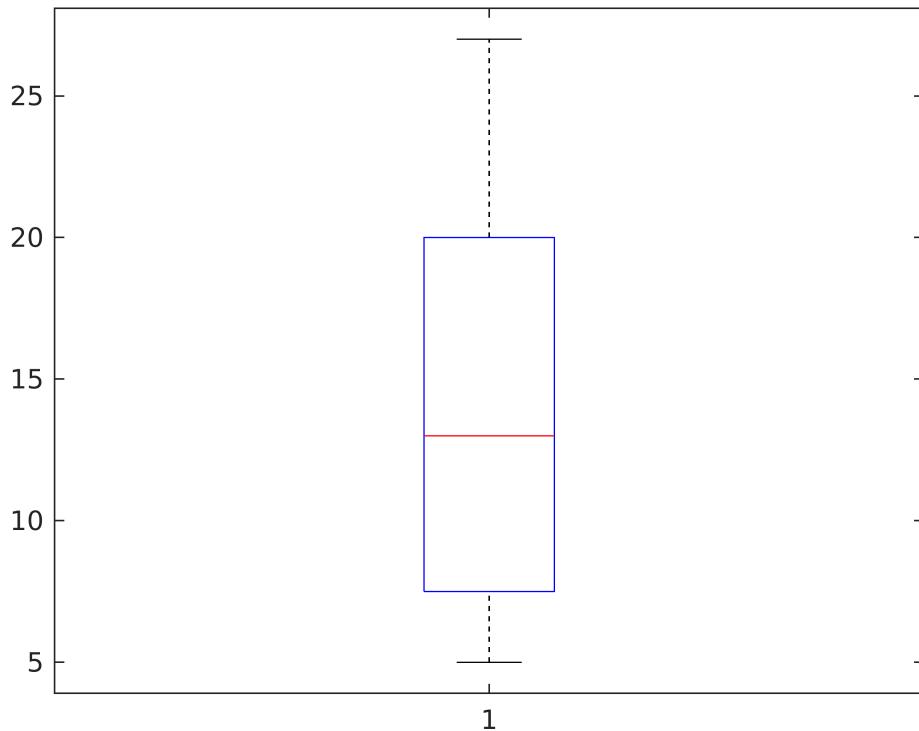
1. Maak een boxplot.
2. Ga na of de gegevens uit een normaal verdeelde populatie komen.
3. Ga na of de gegevens uit een normaal verdeelde populatie met gemiddeld 7 komen.

Oplossing:

Eerst en vooral maken we een rijvector A met daarin de gegevens. Met boxplot kunnen we een boxplot maken van deze gegevens.

```
A=[24, 19, 14, 10, 7, 5, 6, 8, 12, 16, 21, 27];  
boxplot(A);
```

Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For more information, click here.



mean kunnen we gebruiken voor het gemiddelde en std voor de standaardafwijking te berekenen.

```
gem=mean(A)
```

```
gem = 14.0833
```

```
sig=std(A)
```

```
sig = 7.3665
```

Om na te gaan of de gegevens uit een normaalverdeelde populatie komen gebruiken we de Kolmogorov-Smirnov (KS) test.

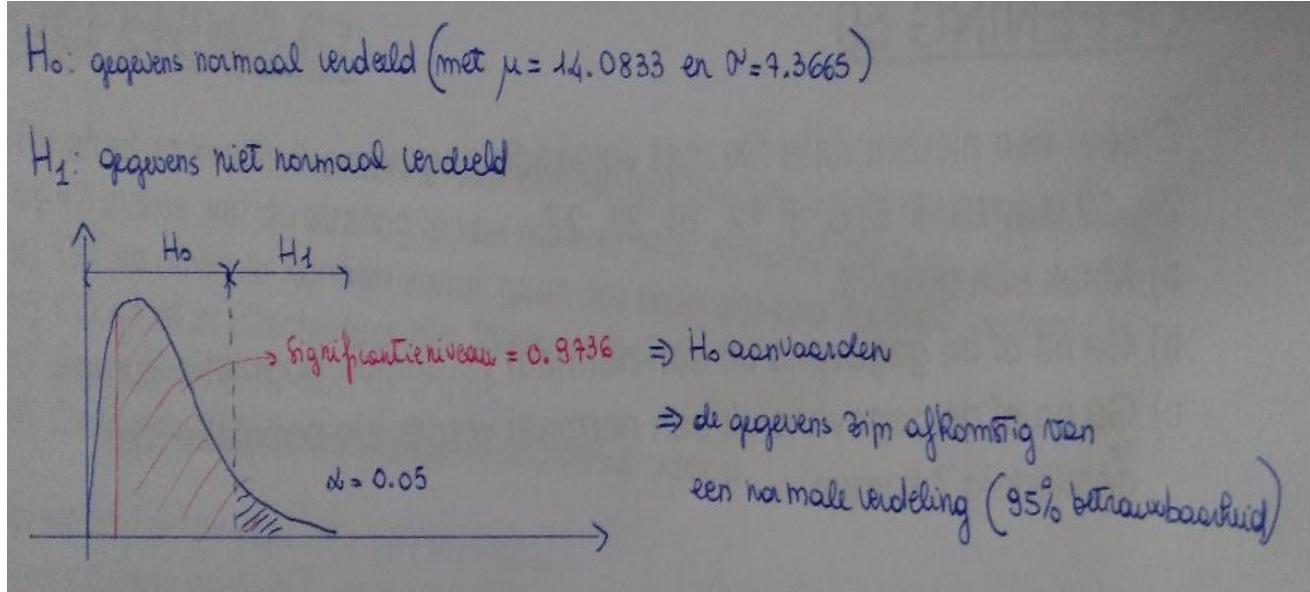
Eerst genereren we een distributie `test_cdf` met `makedist`. De argumenten kunnen we als volgt uitleggen: 'Normal' betekent dat we een normaalverdeling maken met parameters '`mu`' en '`sigma`'. In deze oefening stellen we '`mu`' en '`sigma`' gelijk aan de respectievelijke, eerder berekende waarden `gem` en `sig`. Het object `test_cdf` is de theoretische distributie waarmee de empirische distributie (die volgt uit de data) vergeleken wordt tijdens het gebruik van de KS-test.

De argumenten gebruikt bij het commando `kstest` zijn als volgt: `A` zijn de **empirische** gegevens waarvan we de distributie willen controleren. Met '`CDF`', `test_cdf` geven we de **theoretische** distributie mee, in dit geval een normaalverdeling met gemiddelde `mu` en standaardafwijking `sig`. Het commando `kstest` geeft drie waarden terug: `h`, `p` en `D`.

`h` is de uitkomst van de test met $\alpha = 0.05$, `h = 0` betekent dat we H_0 aanvaarden en `h = 1` betekent dat we H_0 verwerpen (en dus H_1 accepteren). Als het niet anders gespecificeerd wordt is de α -waarde waarmee we werken steeds 0.05. De waarde `p` is het significantieniveau van de test en `D` is de KS-test veranderlijke.

```
test_cdf=makedist('Normal','mu',gem,'sigma',sig);
[h,p,D]=kstest(A,'CDF',test_cdf)
```

```
h = logical
0
p = 0.9736
D = 0.1289
```



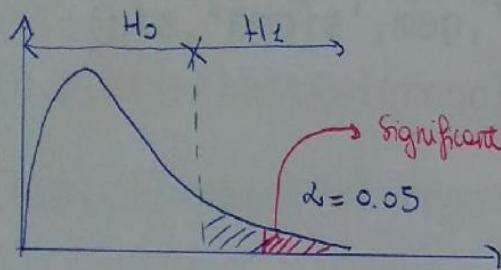
Voor puntje 3 herhalen we deze test maar nu stellen we μ gelijk aan 7.

```
test_cdf=makedist('Normal','mu',7,'sigma',sig);
[h,p,D]=kstest(A,'CDF',test_cdf)
```

```
h = logical  
1  
p = 0.0351  
D = 0.3930
```

H_0 : gegevens normaal verdeeld (met $\mu = 7$ en $N = 7.3665$)

H_1 : gegevens niet normaal verdeeld



Significantieniveau $p = 0.0351 \Rightarrow H_0$ verwijzen
 \Rightarrow de gegevens zijn niet afkomstig
van een normale verdeling met $\mu = 7$
(95% betrouwbaarheid)

Oefening 60

Opgave:

1. Geef de boxplot voor volgende data: 25, 50, 22, 10, 28, 95, 20, 14, 25, 30. Bepaal de uitschieter(s).
2. Test of de gegevens normaalverdeeld zijn op 80%-niveau.
3. Verwijder de uitschieter(s) en test opnieuw.

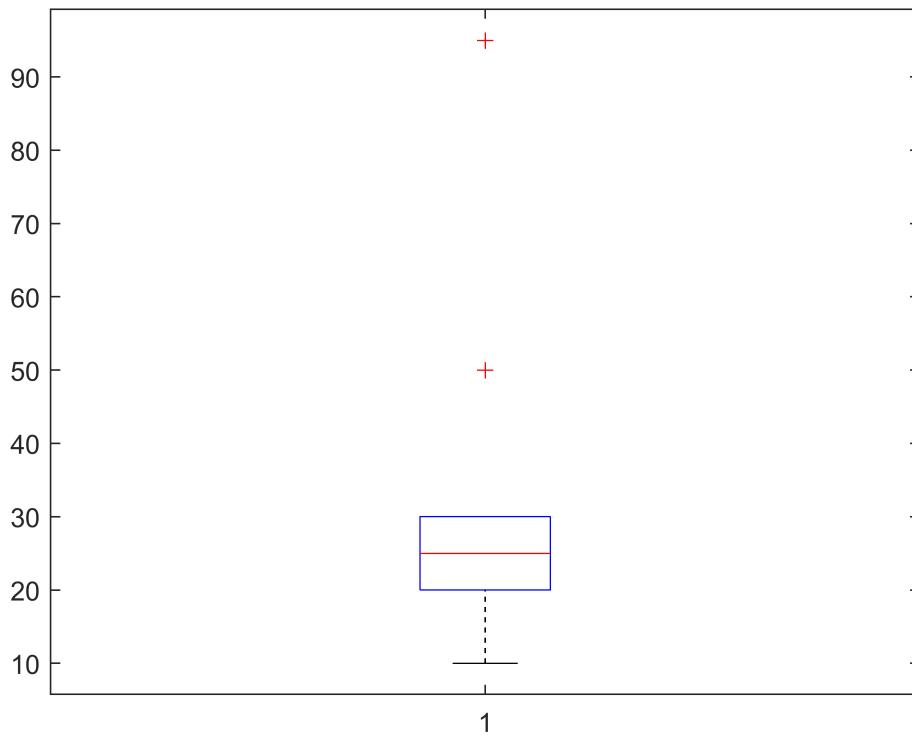
Oplossing:

We maken een vector met de gegevens:

```
A=[25, 50, 22, 10, 28, 95, 20, 14, 25, 30];
```

1. We tekenen de boxplot van de gegevens:

```
boxplot(A)
```



In de tekening van de boxplot worden de uitschieters automatisch aangeduid maar we kunnen de criteria voor de uitschieters nog eens expliciet berekenen (zie cursus statistiek paragraaf 2.5):

```
percentiel_25 = prctile(A, 25);
percentiel_75 = prctile(A, 75);
interkwartiel=iqr(A);
ondergrens = percentiel_25-1.5*interkwartiel
```

```
ondergrens = 5
```

```
bovengrens = percentiel_75+1.5*interkwartiel
```

```
bovengrens = 45
```

Volgens deze berekening zijn er twee uitschieter in de dataset, namelijk 50 en 95.

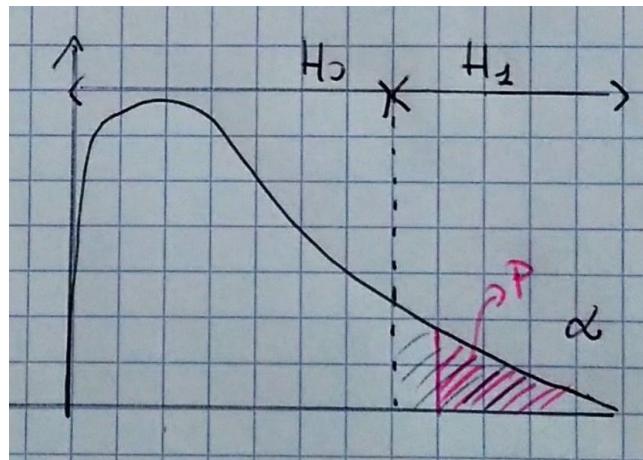
2. We kunnen testen of de gegevens normaal verdeeld zijn met een KS-test:

H_0 :de gegevens zijn normaal verdeeld

H_1 :de gegevens zijn niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(A),'sigma',std(A));  
[h,p,D]=kstest(A,'CDF',test_cdf,'Alpha',0.2)
```

```
h = Logical  
1  
p = 0.1782  
D = 0.3307
```



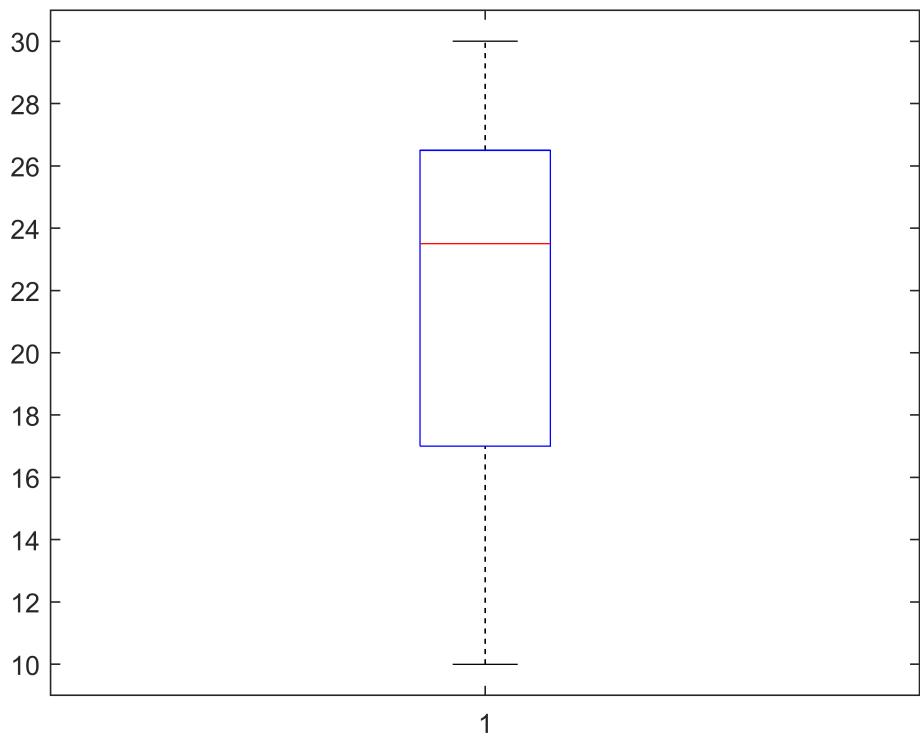
$p = 0.1782 < 0.20 \Rightarrow H_0$ verwerpen: de gegevens zijn niet normaal verdeeld met 80% betrouwbaarheid.

3. Als we de uitschieters 50 en 95 verwijderen verandert de dataset naar:

```
A=[25, 22, 10, 28, 20, 14, 25, 30];
```

Een nieuwe boxplot en KS-test geeft:

```
boxplot(A)
```



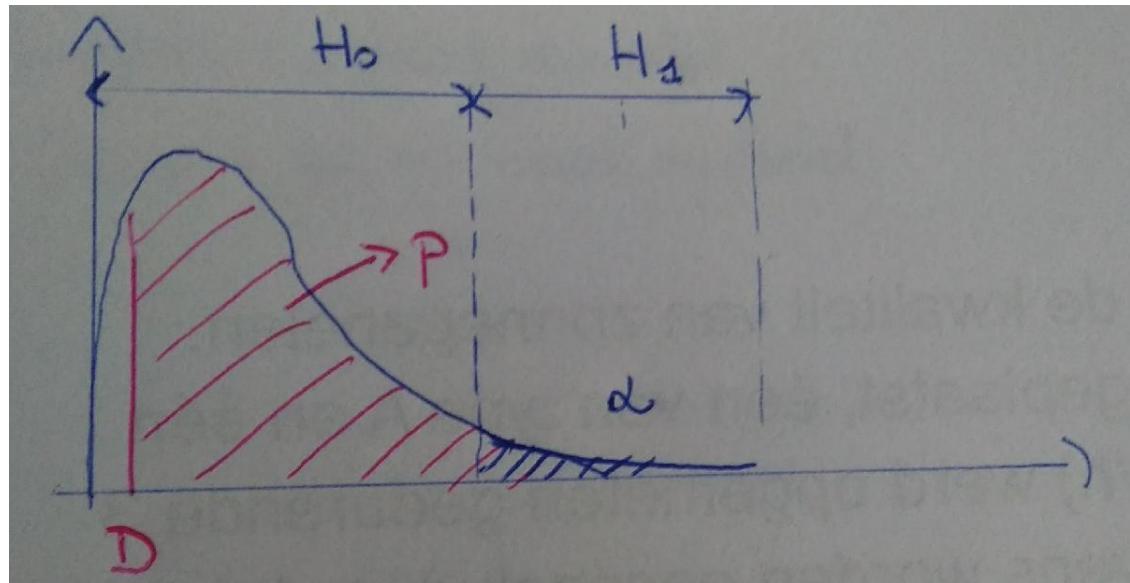
Nu de dataset veranderd is zijn er geen uitschieters meer.

H_0 :de gegevens zijn normaal verdeeld

H_1 :de gegevens zijn niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(A),'sigma',std(A));
[h,p,D]=kstest(A,'CDF',test_cdf,'Alpha',0.2)
```

```
h = Logical
0
p = 0.9129
D = 0.1821
```



$p = 0.9129 > 0.20 \Rightarrow H_0$ aanvaarden: de gegevens zijn normaal verdeeld met 80% betrouwbaarheid als de uitschieters verwijderd worden uit de dataset.

Oefening 61

Opgave:

Bij een fabriek die klinkers maakt, wil men de maatvastheid onderzoeken van de geproduceerde klinkers. Daarvoor werd een steekproef gedaan en de lengte gemeten:

Lengte 12 13 14 16 15 18 19 10 11 12 13 14 15 13 14 12

1. Presenteer een histogram van deze gegevens.
2. Kunnen we aannemen dat deze gegevens normaal verdeeld zijn?
3. Vind de mediaan, de interkwartielafstand, de modus, het gemiddelde, variantie, het 10% percentiel.

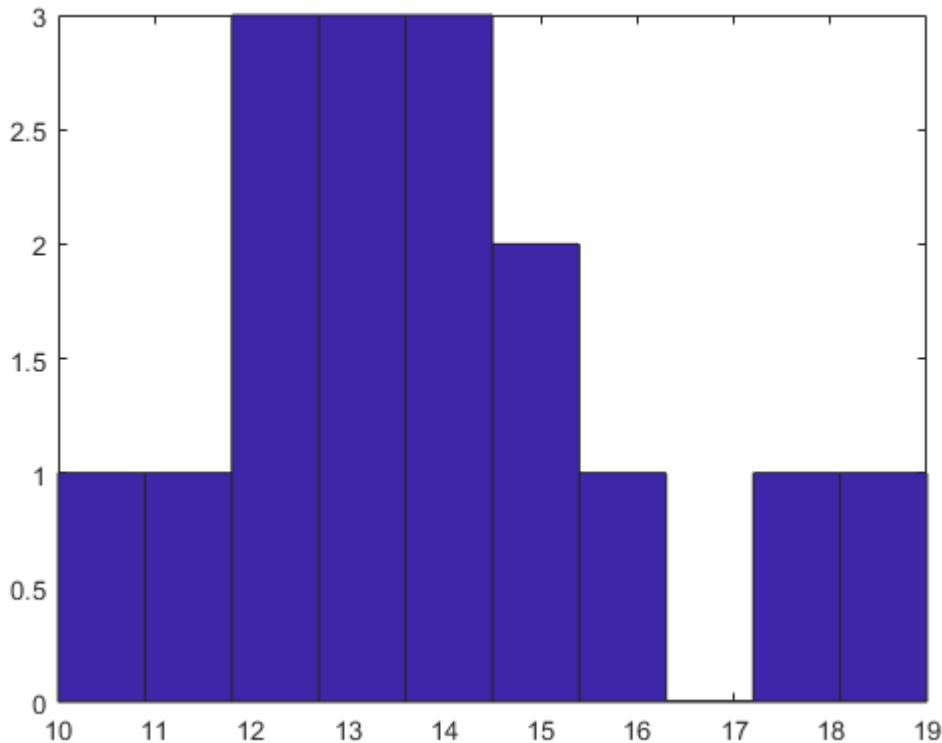
Oplossing:

We slaan de gegevens op in een vector:

```
A=[12 13 14 16 15 18 19 10 11 12 13 14 15 13 14 12];
```

1. Met het commando `hist` kunnen we een histogram maken van de gegevens:

```
hist(A)
```



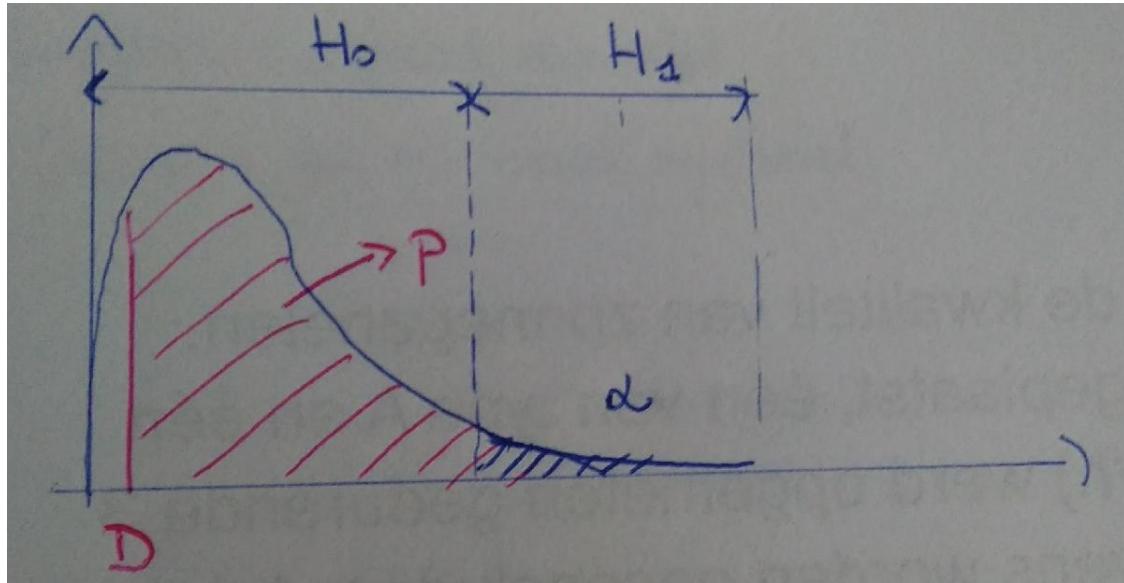
2. We gebruiken een KS-test om na te gaan of de gegevens normaal verdeeld zijn:

H_0 : de gegevens zijn normaal verdeeld

H_1 : de gegevens zijn niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(A),'sigma',std(A));
[h,p,D]=kstest(A,'CDF',test_cdf)
```

```
h = Logical  
0  
p = 0.7739  
D = 0.1564
```



$p = 0.7739 > 0.05 \Rightarrow H_0$ aanvaarden: de gegevens zijn normaal verdeeld met 95% betrouwbaarheid

3. We berekenen de gevraagde waarden met al gebruikte commando's:

```
mediaan=median(A), interkwartiel=iqr(A), modus=mode(A)
```

```
mediaan = 13.5000
interkwartiel = 3
modus = 12
```

```
gemiddelde=mean(A), variantie=var(A), per_10=prctile(A,10)
```

```
gemiddelde = 13.8125
variantie = 5.7625
per_10 = 11.1000
```

Oefening 62

Opgave:

In onderstaande tabel staan de maandelijkse nettolonen van een KMO in de bouwsector.

Maandloon	[1050, 1200[[1200, 1350[[1350, 1500[[1500, 1650[[1650, 1800[[1800, 1950[[1950, 2100[[2100, 2250[
aantal Werknemers	4	7	10	12	7	3	0	2

1. Schat het gemiddelde loon binnen deze KMO. Wat is de standaardafwijking?
2. Bereken zelf het 95% betrouwbaarheidsinterval voor het gemiddelde loon voor een KMO in de bouwsector op basis van deze gegevens

Oplossing:

In deze oefening kunnen we werken met de gemiddelde waarden van elke klasse bij de berekeningen (zie sectie 2.4.3). Deze waarden kunnen we als volgt berekenen: we beginnen met het midden van de eerste klasse (1125), maken dan stappen die even groot zijn als de breedte van de klassen en stoppen bij het midden van de laatste klasse. Deze waarden slaan we op in de vector A.

```
A=1125:150:2175;
```

Het aantal werknemers per klasse slaan we op in de vector freq.

```
freq=[4 7 10 12 7 3 0 2];
```

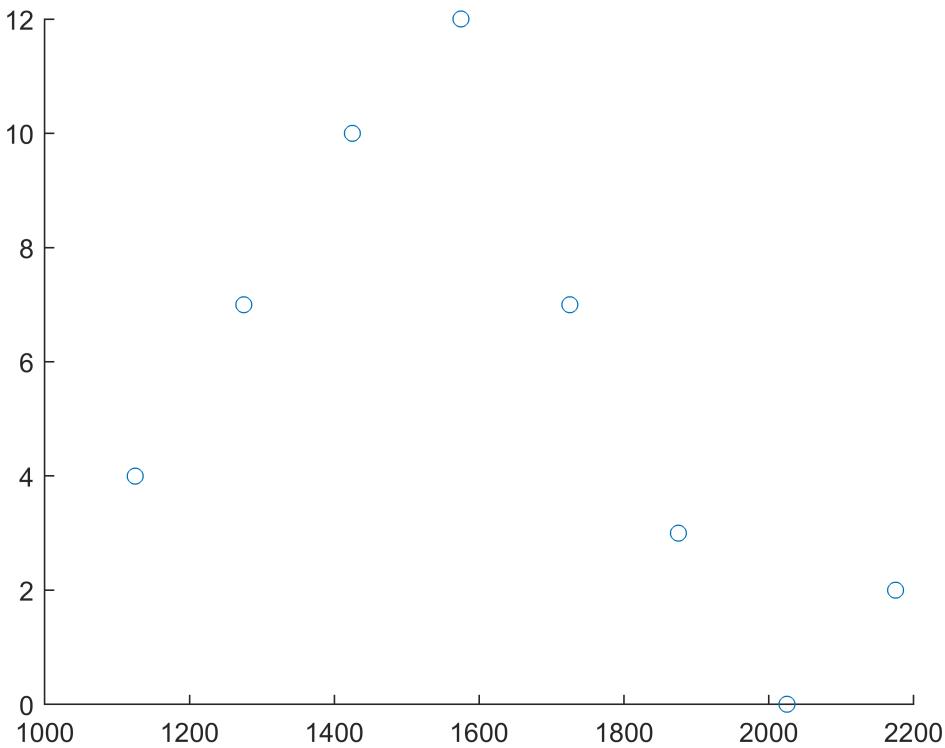
Het totaal aantal werknemers is de som van alle werknemers per klasse:

```
n=sum(freq)
```

```
n = 45
```

We kunnen de data visualiseren aan de hand van een scatter plot:

```
scatter(A, freq)
```



Het steekproefgemiddelde \bar{x} berekenen we als volgt (zie cursus statistiek 2.3.3):

```
gemiddelde=sum(freq.*A)/n
```

```
gemiddelde = 1525
```

En de steekproef standaardafwijking s als volgt (zie cursus statistiek 2.3.3):

```
s=sqrt(sum(freq.*(A-gemiddelde).^2)/(n-1))
```

```
s = 247.7168
```

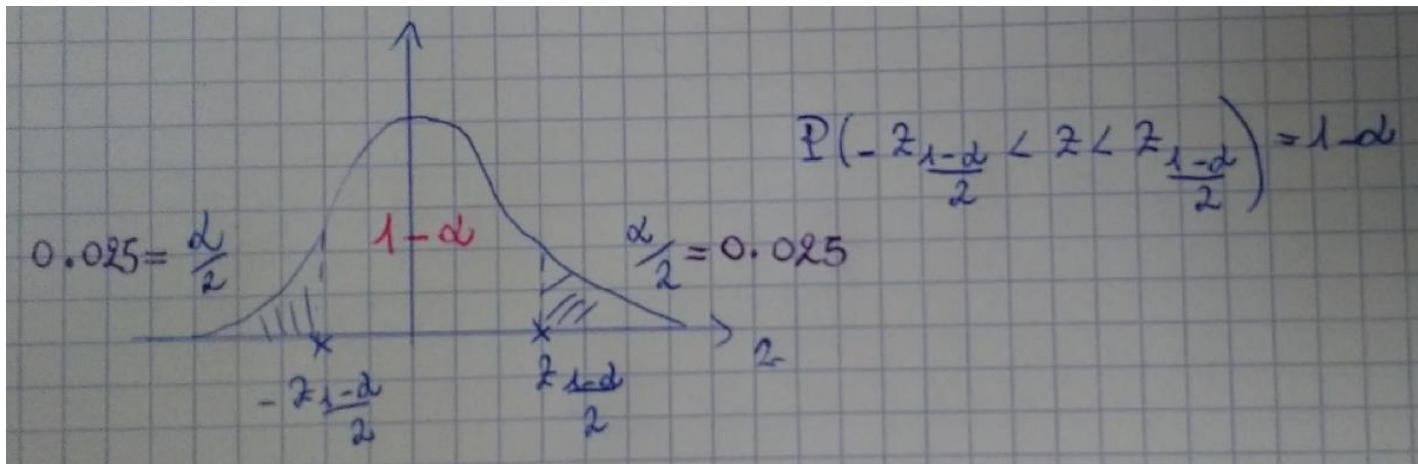
Het aantal steekproefwaarden is 45. Dit is groot genoeg (> 30) om te werken met de normale verdeling

$N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$ als benadering voor de verdeling van het steekproefgemiddelde \bar{x} wegens de centrale limietstelling

(zie sectie 6.3.2). Het berekenen van de grenzen van het betrouwbaarheidsinterval maken we gebruik van de inverse cumulatieve distributiefunctie (zie handleiding pagina 32) van $N(0, 1)$.

We maken eerst volgende transformatie: $z = \frac{\bar{x} - \mu}{s/\sqrt{n}}$. De variabele z heeft een $t(44 \text{ df})$ -verdeling ($44 = n - 1$),

maar gaan we benaderen door een $N(0, 1)$ -verdeling gezien er genoeg steekproefwaarden zijn. Daardoor kan het 95% betrouwbaarheidsinterval voor μ berekend worden met $\left[\bar{x} - z_{0.475} \frac{s}{\sqrt{n}}, \bar{x} + z_{0.475} \frac{s}{\sqrt{n}}\right]$.



$z_{0.475}$ kunnen we berekenen met de inverse cumulatieve distributiefunctie. Let op de notatie voor $z_{0.475}$, 0.475 staat voor de waarde van de integraal tussen $z = 0$ en $z = z_{0.475}$, niet tussen $z = -\infty$ en $z = z_{0.475}$. We moeten hier dus 0.5 bij optellen als we norminv gebruiken.

$$LL = \text{gemiddelde}-\text{norminv}(0.975, 0, 1)*(s/\sqrt{n})$$

$$LL = 1.4526e+03$$

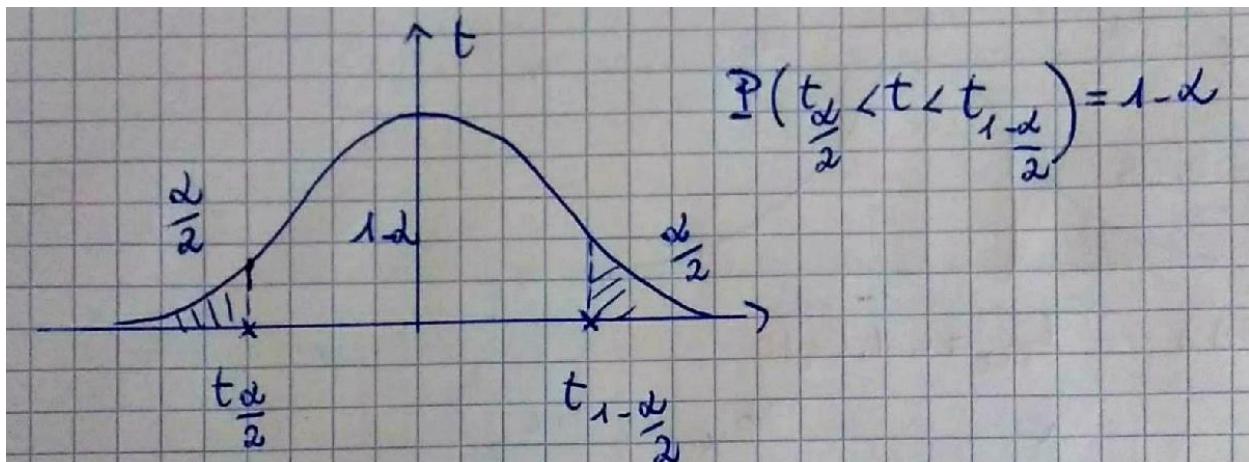
$$RL = \text{gemiddelde}+\text{norminv}(0.975, 0, 1)*(s/\sqrt{n})$$

$$RL = 1.5974e+03$$

Om de benadering van de verdeling van \bar{x} te evalueren, vergelijken we het bekomen betrouwbaarheidsinterval gebaseerd op de t -verdeling met 44 vrijheidsgraden. De exacte waarde van het 95% betrouwbaarheidsinterval kan dan berekend worden als

$$[\text{gemiddelde}-tinv(0.975, n-1)*s/\sqrt{n}, \text{gemiddelde}+tinv(0.975, n-1)*s/\sqrt{n}]$$

$$\begin{aligned} ans &= 1 \times 2 \\ 10^3 \times & \\ 1.4506 &\quad 1.5994 \end{aligned}$$



Conclusie: de overeenkomst tussen de twee betrouwbaarheidsintervallen is groot. Dit komt omdat het aantal steekproefwaarden groot genoeg is.

Oefening 63

Opgave:

De resultaten (/100) behaald door een selectie van 10 studenten op het examen wiskunde en economie worden gegeven in onderstaande tabel:

Student	1	2	3	4	5	6	7	8	9	10
Score Wiskunde	30	73	51	57	81	21	75	46	85	21
Score Economie	45	70	40	60	71	30	80	50	70	10

- Wat is de gemiddelde score voor het examen wiskunde bij deze 10 studenten?
- Schat de verwachte score voor het examen economie. Wat is daarbij de standaardfout?
- Bij welk examen is de variantie het grootst voor deze groep studenten?
- Maak een scatterplot waarbij je de score voor wiskunde uitzet ten opzichte van de score voor economie.

Oplossing:

Eerst definiëren wij een 2×10 matrix met daarin de data uit de tabel.

```
A=[30 73 51 57 81 21 75 46 85 21;
 45 70 40 60 71 30 80 50 70 10];
```

Met `mean(A, 2)` berekenen we het gemiddelde van de elementen van de rijen in de matrix A en we krijgen een kolomvector met twee componenten en die noemen we `gem`.

```
gem = mean(A, 2)
```

```
gem = 2×1
54.0000
52.6000
```

Er bestaat een tweede manier om dit te berekenen door eerst rijvectoren van de rijen in A te maken en dan apart het gemiddelde te berekenen.

```
gem=[mean(A(1,:));mean(A(2,:))]
```

```
gem = 2×1
54.0000
52.6000
```

Hieruit volgt dat de gemiddelde score voor het examen wiskunde 54 is.

Met `var(A, 0, 2)` berekenen we de variantie van de elementen van de rijen in de matrix A en wij krijgen een kolomvector met twee componenten en die noemen we `variantie`. De extra parameter 0 betekent dat we geen extra gewichten toekennen aan de verschillende elementen van de rijen.

```
variantie=var(A,0,2)
```

```
variantie = 2×1
592.0000
477.6000
```

Er bestaat een tweede manier om dit te berekenen door eerst rijvectoren van de rijen in A te maken en dan apart de variantie te berekenen.

```
variantie=[var(A(1,:));var(A(2,:))]
```

```
variantie = 2x1  
 592.0000  
 477.6000
```

De variantie is het grootst bij het examen wiskunde.

De steekproefstandaardafwijking is gelijk aan de vierkantswortel van de steekproefvariantie

```
sigma_economie = sqrt(variantie(2))
```

```
sigma_economie = 21.8541
```

De standaardfout kunnen we berekenen door de steekproefstandaardafwijking te delen door de vierkantswortel van het aantal waarnemingen in de steekproef.

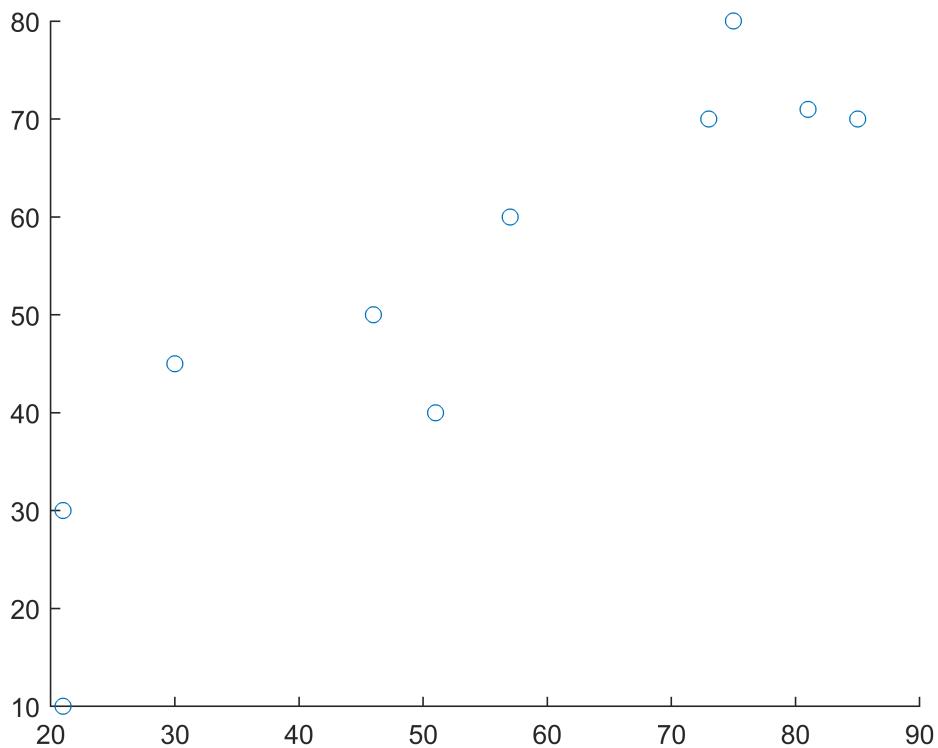
```
standaardfout=sigma_economie/sqrt(10)
```

```
standaardfout = 6.9109
```

De geschatte verwachte score bij het examen economie stellen we gelijk aan het gemiddelde, namelijk 52, met standaardfout 6.9.

We maken een scatterplot met het commando `scatter` met de wiskundescores op de x-as en de economiescores op de y-as.

```
scatter(A(1,:),A(2,:))
```



Oefening 64

Opgave:

Genereer een steekproef met 200 waarnemingen van een variabele met als verdeling $N(2, 1)$.

1. Teken de histogram van deze steekproef.
2. Vind een interval waarin het steekproefgemiddelde van een dergelijke steekproef zal liggen met 80% betrouwbaarheid.

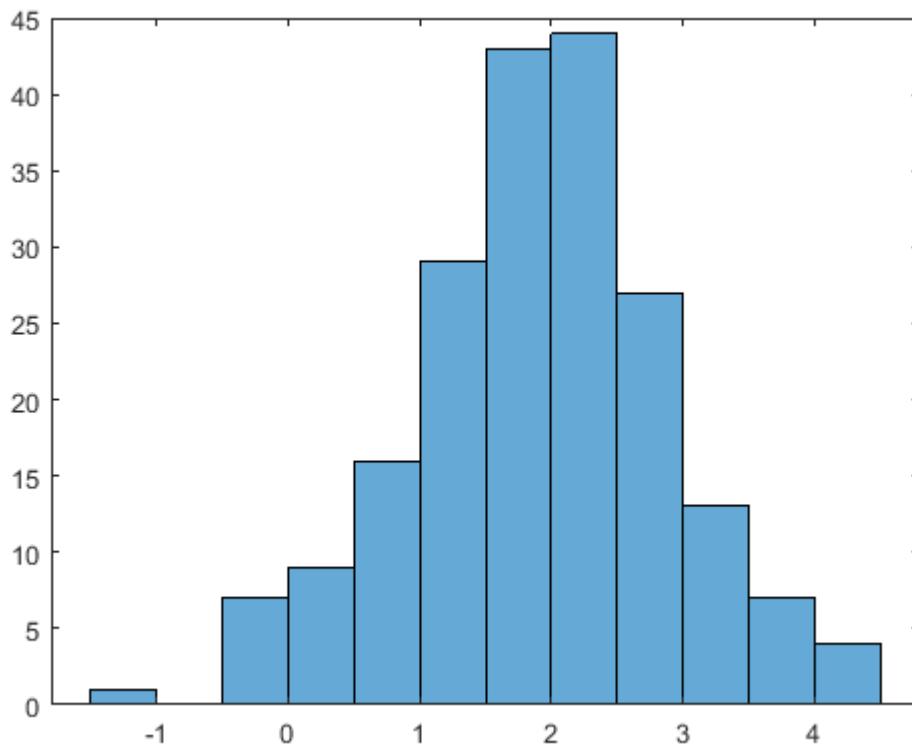
Oplossing:

Aan de hand van het commando `normrnd(2, 1, 1, 200)` kunnen we randomwaarden genereren uit een normaalverdeling. 2 is de eerste parameter en staat voor het gemiddelde, 1 is de tweede parameter en staat voor de standaardafwijking. De derde en vierde parameter, 1 en 200, zorgen ervoor dat `normrnd` een 1×200 matrix genereert met daarin steekproefwaarden van de normaalverdeling.

```
steekproef=normrnd(2,1,1,200);
```

Met `histogram` maken we een histogram uit de waarden van de steekproef-matrix.

```
histogram(steekproef)
```



Het berekenen van het betrouwbaarheidsinterval gebeurt als volgt:

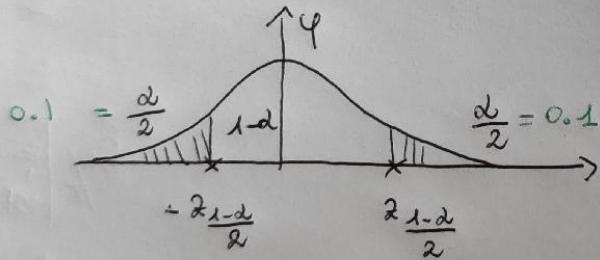
Het $(1-\alpha)\%$ BI voor μ indien σ gekend is. (zie cursus blz 7)

$$\bar{x} \text{ is } N\left(\mu, \frac{1}{\sqrt{200}}\right)$$



$$Z = \frac{\bar{x} - \mu}{\frac{1}{\sqrt{200}}} \text{ is } N(0,1) \text{ verdeeld}$$

$$80\% \text{ BI} \Rightarrow \alpha = 1 - 0.8 = 0.2$$



$$P\left(-\frac{z_{1-\alpha/2}}{\sqrt{200}} < Z < \frac{z_{1-\alpha/2}}{\sqrt{200}}\right) = 1 - \alpha = \gamma$$

Met behulp van norminv kunnen we de grenzen van het gezochte interval berekenen (zie oefening 65). Dit doen we voor $\frac{\alpha}{2}$ en $1 - \frac{\alpha}{2}$ (0.1 en 0.9).

```
norminv([0.1 0.9], 2, 1/sqrt(200))
```

```
ans = 1x2  
1.9094 2.0906
```

Oefening 65

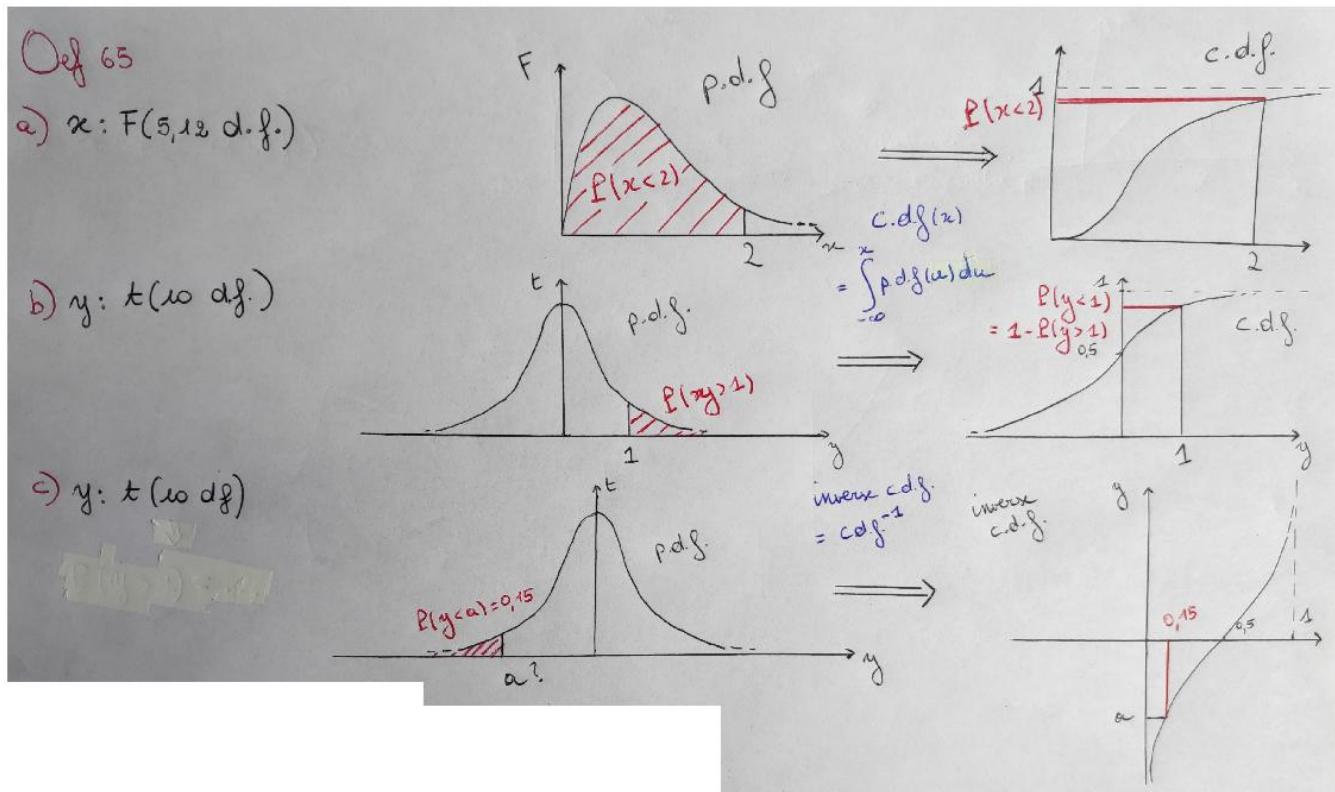
Opgave:

1. Als $x : F(5, 12\text{d.f.})$, dan is $P(x < 2) = \dots$
2. Als $y : t(10\text{d.f.})$, dan is $P(y > 1) = \dots$
3. Wat is a als $P(y < a) = 0.15$?
4. Als $z : Bin(500, 0.02)$, dan is $P(z < 12) = \dots$

(controleer dit laatste ook met de normale verdeling en geef commentaar)

Oplossing:

Voor de eerste twee oefeningen moeten we beroep doen op de CDF van de verschillende delingen, dit doen we met `fcdf` en `tcdf`. Bij de derde oefening moeten we met het inverse van de CDF werken, dit doen we met `tinv`. Onderstaande tekeningen illustreren de wiskundige redeneringen hiervoor.



```
P_x_kl_2 = fcdf(2, 5, 12)
```

```
P_x_kl_2 = 0.8491
```

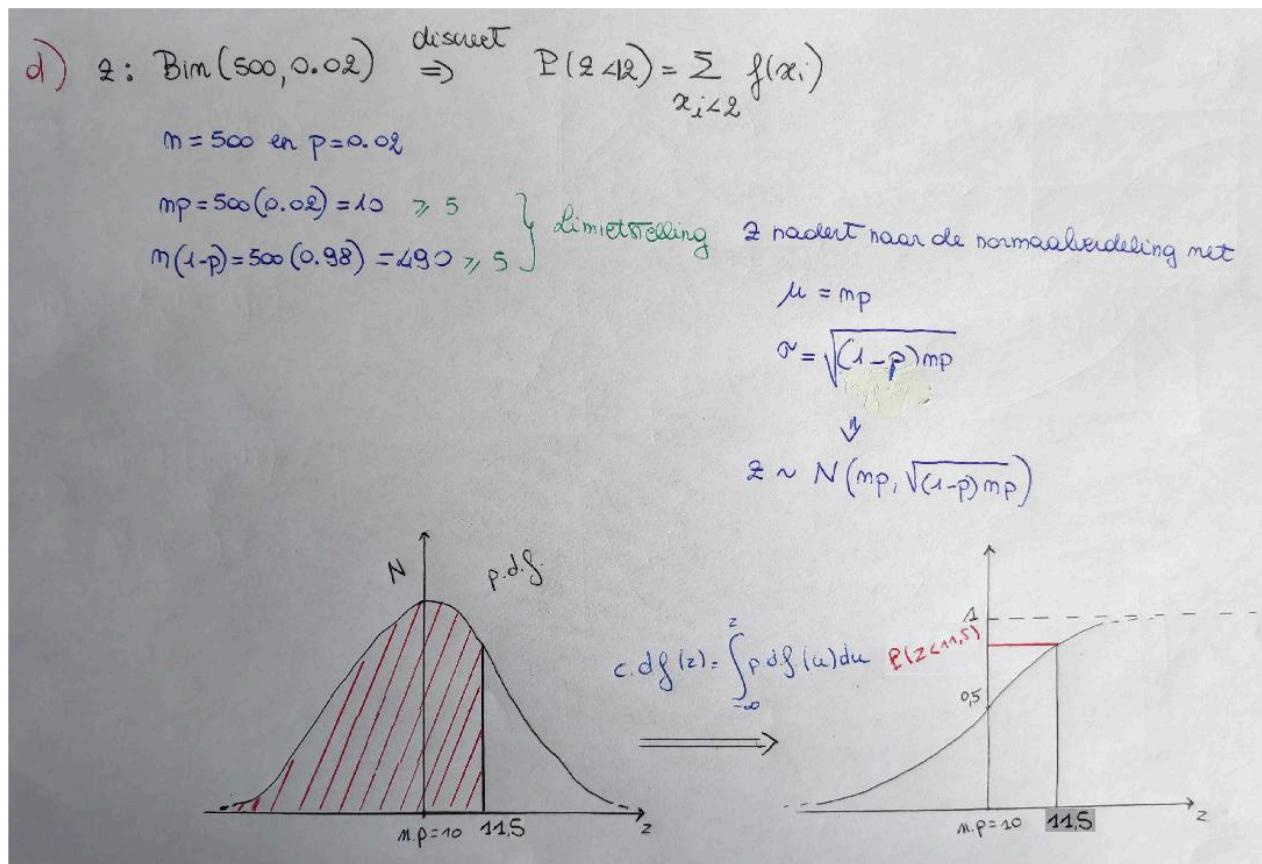
```
P_y_gr_1 = 1-tcdf(1,10)
```

```
P_y_gr_1 = 0.1704
```

```
a = tinv(0.15,10)
```

```
a = -1.0931
```

De vierde oefening is analoog met de eerste. Met het verschil dat we nu met een discrete (binomiale) verdeling werken. Als extra vraag moeten we dit resultaat vergelijken met het resultaat als we de binomiale verdeling gaan benaderen door een normaalverdeling. Dit wordt uiteengezet in volgende redenering:



```
binocdf(11,500,0.02)
```

```
ans = 0.6979
```

Bij het overgaan van de discrete binomiaalverdeling naar de continue normaalverdeling moeten we een continuïteitscorrectie doorvoeren. Dit komt er op neer dat de discrete grens 11 vertaald wordt naar een continue grens van 11.5.

```
n=500;  
p=0.02;
```

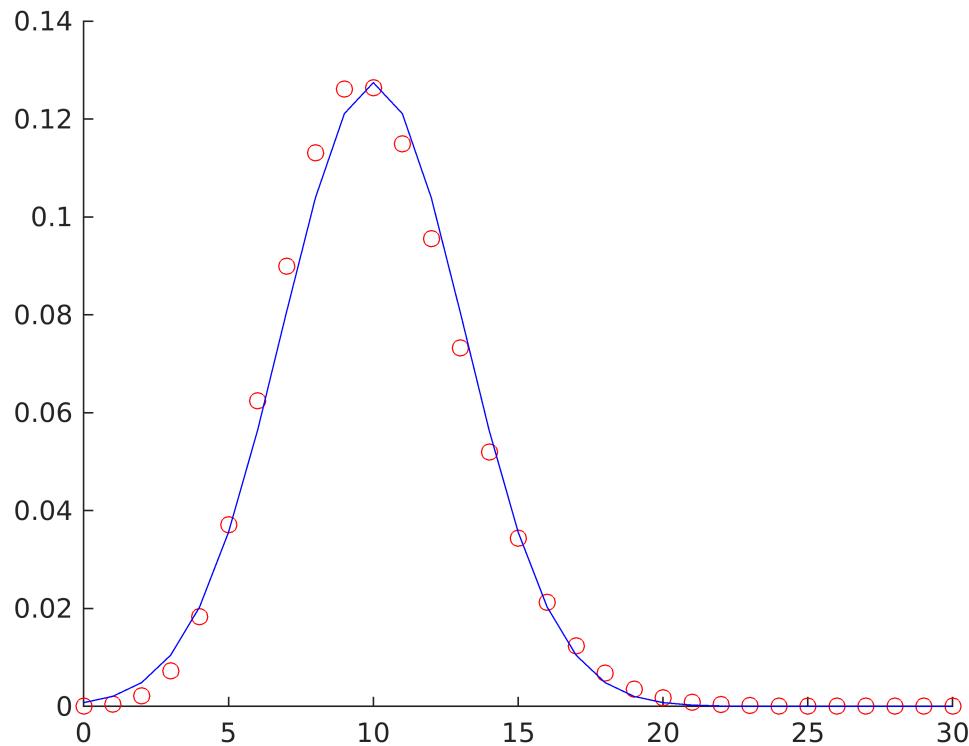
```
normcdf(11.5,n*p,sqrt(n*p*(1-p)))
```

```
ans = 0.6841
```

Er kan gezien worden dat er een redelijke overeenkomst is, dit is te verwachten want $n \cdot p$ en $n \cdot (1 - p)$ zijn beide groter dan 5. De overeenkomst is echter niet perfect, dit komt omdat $n \cdot p = 10$ relatief dicht bij de vuistregel van 5 aanleunt.

Door de PDF van beide distributies boven elkaar te zetten kunnen ze grafisch vergeleken worden. Omdat de binomiale verdeling discreet is, is het best om hierbij een scatterplot te gebruiken in de plaats van een continue curve zoals bij de normaalverdeling.

```
i = 0:30;
scatter(i, binopdf(i,500,0.02), 'r')
hold on;
plot(i, normpdf(i,n*p,sqrt(n*p*(1-p))), 'b')
```



Oefening 66

Opgave:

Genereer zelf een steekproef met 100 waarnemingen van lengtes van planken. De leverancier stelt dat de lengtes normaal verdeeld zijn met gemiddelde 2 m en $\sigma=9$ cm. Onze zaagmachine zaagt van elke plank een deel af met een lengte die normaal verdeeld verondersteld wordt met gemiddelde 1.5 m en $\sigma=1$ cm. Simuleer dit. Welke verdeling verwacht je voor de lengtes van de overblijvende stukken van de planken als je weet dat de werking van de machine onafhankelijk is van de lengte van de plank. Controleer dit aan de hand van de grafieken van de cumulatieve distributiefuncties (theoretische functie in het rood).

Oplossing

We definiëren een 2×100 matrix A waarvan elke rij door het commando `normrnd` wordt gegenereerd. Aan de hand van het commando `normrnd(2, 0.09, 1, 100)` kunnen we randomwaarden genereren uit een normaalverdeling voor de oorspronkelijke lengtes van de plank. 2 is de eerste parameter en staat voor het gemiddelde, 0.09 is de tweede parameter en staat voor de standaardafwijking. De derde en vierde parameter, 1 en 100, zorgen ervoor dat `normrnd` een 1×100 matrix (voor de eerste rij van de matrix A) genereert met daarin steekproefwaarden van de normaalverdeling. Analoog gebruiken we het commando `normrnd(1.5, 0.01, 1, 100)` voor de tweede rij van de matrix A die de lengtes van de afgezaagde delen voorstelt.

```
A=[normrnd(2, 0.09, 1, 100);normrnd(1.5, 0.01, 1, 100)];
```

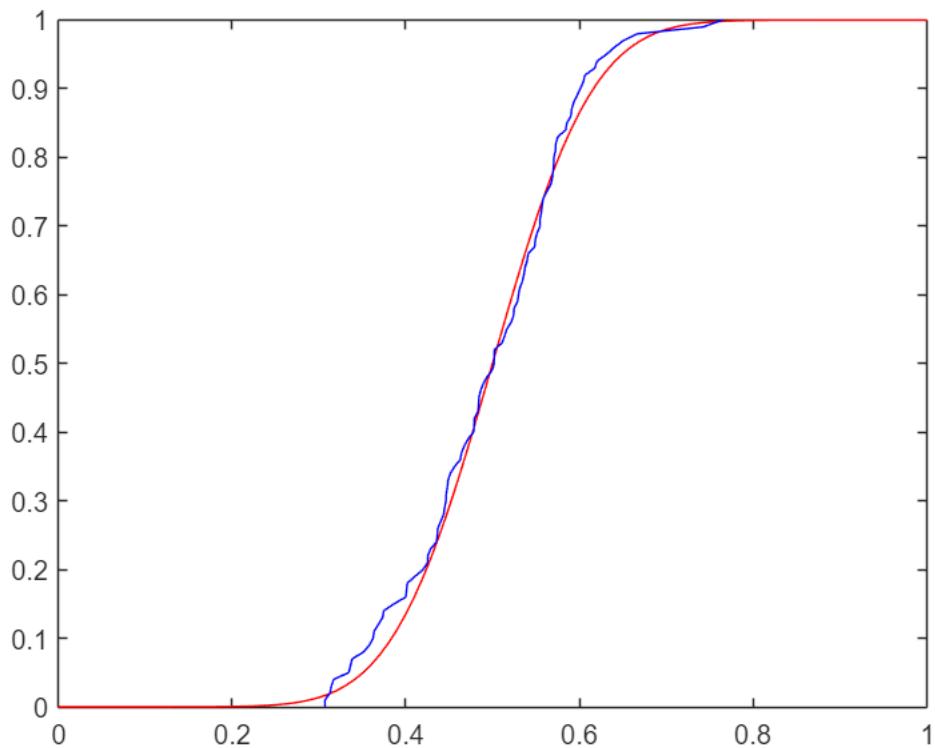
Het commando `normcdf` definieert de cumulatieve verdelingsfunctie van een normaalverdeling.

Aangezien de werking van de machine onafhankelijk van de lengte van de plank is, is de lengte van de overblijvende stukken van de planken normaal verdeeld met parameters $\mu = 2 - 1.5 = 0.5$ en $\sigma = \sqrt{(0.09)^2 + (0.01)^2}$ (zie eigenschap 5.3.3 blz 52 van de cursus statistiek). De cumulatieve distributiefunctie van deze theoretische normale verdeling wordt met volgend commando geplot (in het rood):

```
x=0:1/100:1;
y_th=normcdf(x, 0.5, sqrt(0.09^2+0.01^2));
plot(x,y_th, 'r-')
hold on
```

We kunnen ook de lengtes van de overblijvende stukken van de 100 planken berekenen (verschil tussen de oorspronkelijke lengte van de planken en het gezaagde deel) en dan de overeenkomstige (empirische) cumulatieve verdelingsfunctie met het commando `ecdf`. `ecdf` geeft zowel de waarden op de x-as en y-as terug en we kunnen die gelijk stellen aan x en f door `[f, x]` te gebruiken. Plotten kunnen we dan eenvoudig doen met `plot`.

```
y_sim=A(1,:)-A(2,:);
[f,x]=ecdf(y_sim);
plot(x,f, 'b')
```



Oefening 67

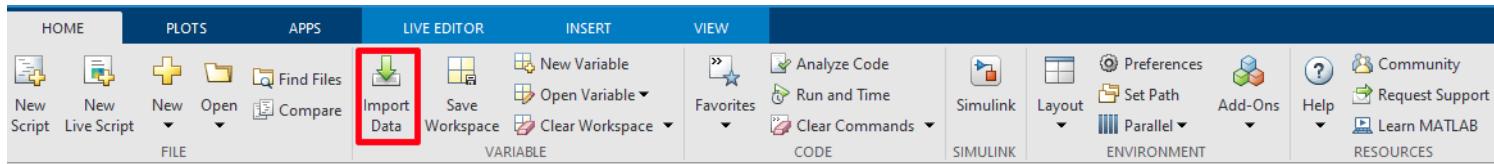
Opgave:

De afdeling kwaliteitscontrole van een fabriek die microgolfovens maakt, meet bij 42 ovens wat de straling is van de oven met gesloten deuren. De gegevens vind je in het bestand `microgolf.dat`.

1. Ga na of we er kunnen vanuit gaan dat deze emissie normaal verdeeld is.
2. Ga na of er uitschieters zijn. Welke zijn die?
3. Test of $\mu = 0.10$ met $\alpha = 0.05$.
4. Geef een 99% betrouwbaarheidsinterval voor μ .

Oplossing:

Bij deze oefening moeten we data uitlezen uit een bestand. De naam van dit bestand is `microgolf.dat`. Om de data uit dit bestand te importeren moet je eerst en vooral dit bestand opslaan op je computer (of je UGent drive als je met Athena werkt), daarna kan je het bestand importeren met de 'Import Data' knop op het 'Home' scherm.



Nadat je het juiste bestand hebt gekozen moet je nog een selectie maken over welke data je precies wilt importeren. Dit moet je doen door de relevante data te selecteren en te importeren met 'Import Selection'. De eerste rij van de tabel zijn de kolomnamen en je moet opletten dat je die niet selecteert! Normaal gezien selecteert Matlab op een automatische manier de data en moet je juist maar eens controleren.

The screenshot shows the MATLAB Import Data dialog box. The 'Import Selection' button at the bottom right is highlighted with a red box. The dialog includes settings for Delimited or Fixed Width files, a range of A2:B43, and an output type of Table. The 'Import Selection' button is located in the bottom right corner of the dialog.

De ingelezen kolommen kan je dan oproepen door naam_van_bestand.naam_van_kolom. In dit geval:
microgolf.emissie:

```
microgolf.emissie
```

```
ans = 42x1
0.1500
0.0900
0.1800
0.1000
0.0500
0.1200
0.0800
0.0500
0.0800
0.1000
:
:
```

1. Het testen of de gegevens normaal verdeeld zijn doen we aan de hand van een KS-test:

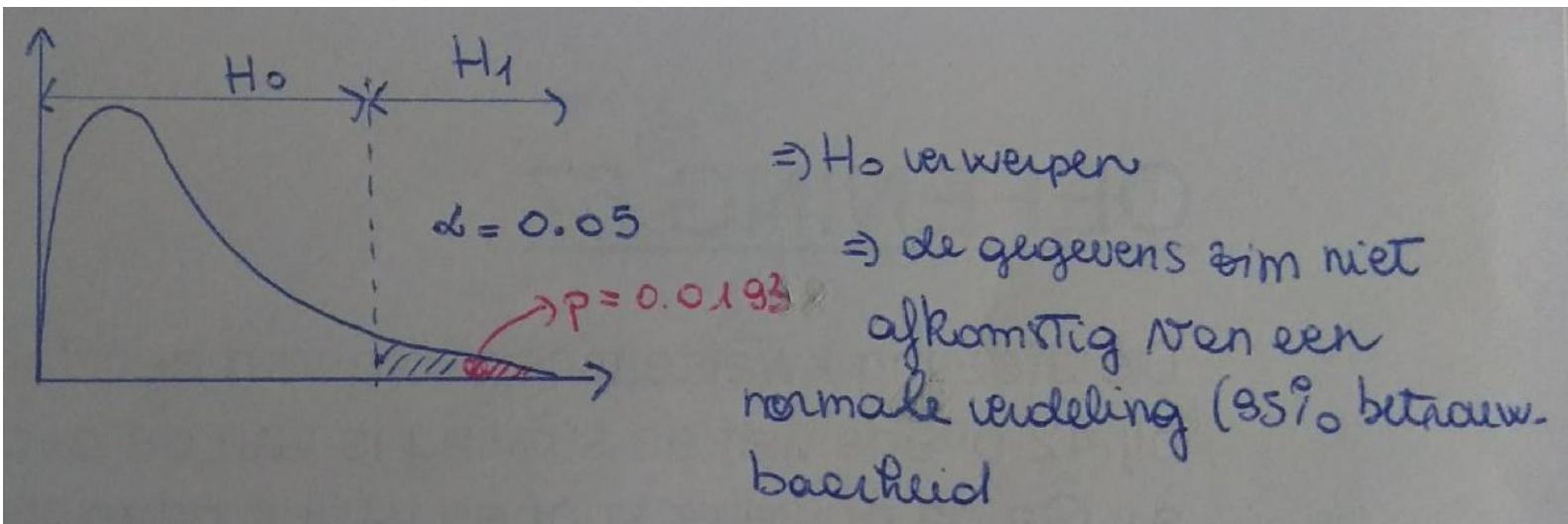
H_0 :normaal verdeeld

H_1 :niet normaal verdeeld

```
gem=mean(microgolf.emissie);
sig=std(microgolf.emissie);
test_cdf=makedist('Normal','mu',gem,'sigma',sig);
[h,p,D]=kstest(microgolf.emissie,'CDF',test_cdf)

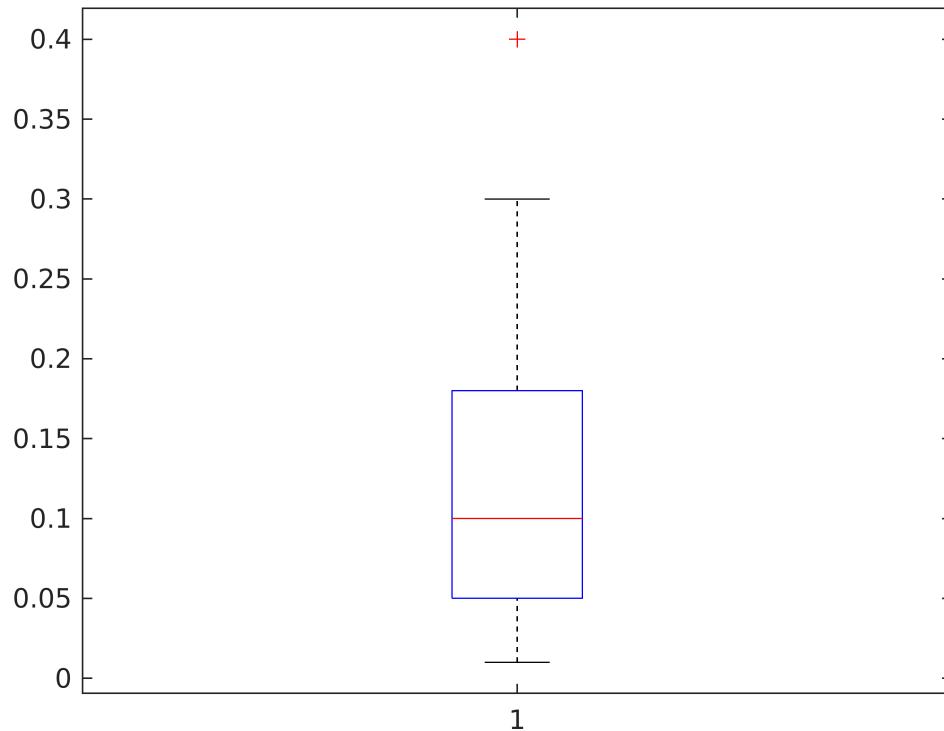
h = logical
```

$$\begin{aligned} p &= 0.0193 \\ D &= 0.2303 \end{aligned}$$



2. Om het probleem grafisch voor te stellen kunnen we een boxplot van de data maken (zie cursus statistiek paragraaf 2.5).

```
boxplot(microgolf.emissie)
```



Uit de boxplot kunnen we aflezen dat er een uitschieter is met waarde 0.4. We kunnen de criteria voor uitschieters berekenen aan de hand van het eerste kwartiel (25 percentiel), het derde kwartiel (75 percentiel) en de lengte van de box (interkwartielafstand). Percentielen kunnen we uitrekenen met het commando `prctile`,

met als argumenten de vector met data en het percentiel dat we willen uitrekenen. De interkwartielafstand kunnen we berekenen met `iqr`.

```
percentiel_25 = prctile(microgolf.emissie, 25);
percentiel_75 = prctile(microgolf.emissie, 75);
interkwartiel = iqr(microgolf.emissie);
ondergrens = percentiel_25 - 1.5 * interkwartiel
```

```
ondergrens = -0.1450
```

```
bovengrens = percentiel_75 + 1.5 * interkwartiel
```

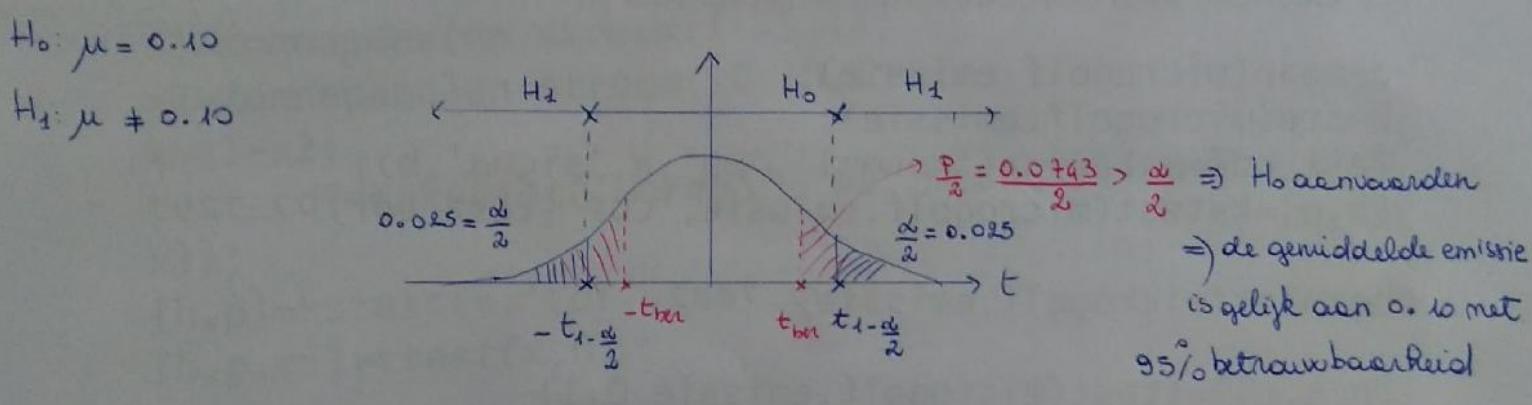
```
bovengrens = 0.3750
```

0.4 is groter dan 0.3750 (de bovengrens) en dus kunnen we inderdaad spreken van een uitschieter.

3. Om vraag 3 te beantwoorden moeten we werken met een 1-sample, tweezijdige t-test. Het is beter om een niet-parametrische test uit te voeren aangezien er geen normaalverdeling aanwezig is. We kunnen echter ook een (parametrische) t-test gebruiken zolang we in het achterhoofd houden dat het resultaat niet even betrouwbaar is. Deze t-test voeren we uit met het commando `ttest`. De argumenten van dit commando zijn: de naam van de vector met de gegevens en de theoretische waarde van μ . De return waarden van dit commando zijn als volgt: `h` is de uitkomst van de test (met $\alpha = 0.05$ tenzij anders gespecificeerd), `p` is het significantieniveau en `ci` is het (95%-)betrouwbaarheidsinterval van μ . Om extra belangrijke statistische waarden te krijgen, zoals de t-statistiek, het aantal vrijheidsgraden en de standaardafwijking, vragen we een extra return value aan (deze noemen we `stats`).

```
[h,p,ci, stats]=ttest(microgolf.emissie,0.1)
```

```
h = 0
p = 0.0743
ci = 2x1
    0.0971
    0.1596
stats = struct with fields:
    tstat: 1.8313
    df: 41
    sd: 0.1003
```



Merk op: $t_{ber} = t_{stat}$.

4.Om het 99% procent betrouwbaarheidsinterval te vinden moeten we 'Alpha' , 0.01 als optie toevoegen.

Wat we hier gebruiken als tweede argument (de theoretische waarde van μ) maakt in se niet uit voor het berekenen van het betrouwbaarheidsinterval (zie cursus Statistiek paragraaf 6.3.1.3), hier kiezen we 0.

```
[h,p,ci]=ttest(microgolf.emissie, 0, 'Alpha', 0.01)
```

```
h = 1
p = 2.6174e-10
ci = 2x1
    0.0865
    0.1701
```

Dit levert het 99% betrouwbaarheidsinterval [0.0865, 0.1701] voor μ .

Oefening 68

Opgave:

Een consumentenorganisatie evalueert de kwaliteit van zonnepanelen. Daarvoor werden op 15 daken panelen geplaatst, één van type A en één van type B. De geleverde stroom (in kWh) werd opgemeten gedurende 3 maanden. De data kan je vinden in `zonnepanelen.txt`. Kan op basis van deze gegevens worden geconcludeerd dat er een significant verschil is in opgewekte stroom tussen de twee soorten panelen?

Oplossing:

Eerst een vooral importeren we de data uit de file `zonnepanelen.txt` (zie oefening 67). We maken twee vectoren aan met de geleverde stroom, één voor type A en één voor type B. Deze noemen we `x1` en `x2`.

```
x1=zonnepanelen.stroom(1:15,1)
```

```
x1 = 15x1  
2330  
3993  
1447  
4687  
2104  
2340  
1780  
3544  
1696  
4082
```

```
:
```

```
x2=zonnepanelen.stroom(16:30,1)
```

```
x2 = 15x1  
2500  
3501  
1647  
5875  
2080  
2695  
2239  
3979  
1609  
4530
```

```
:
```

Het is evident dat de waarnemingen (geleverde stroom type A en type B) gekoppeld zijn, omdat de gemeten stroom inherent afhankelijk is van het dak waar het zonnepaneel op ligt, door onder andere de helling van het dak en de oriëntatie (O, W, N, Z). Het effect van de verschillende daken op de geleverde stroom kan echter weg gefilterd worden door te werken met een gepaarde t-test en we beschouwen dus de metingen als één steekproef met de vijftien verschillen als nummerieke gegevens. De t-test is tweezijdig omdat we de eventuele gelijkheid (of verschil) tussen de twee types zonnepanelen willen aantonen.

Er moet een extra variabele aangemaakt worden die het verschil is van de twee steekproeven:

```
x=x1-x2;
```

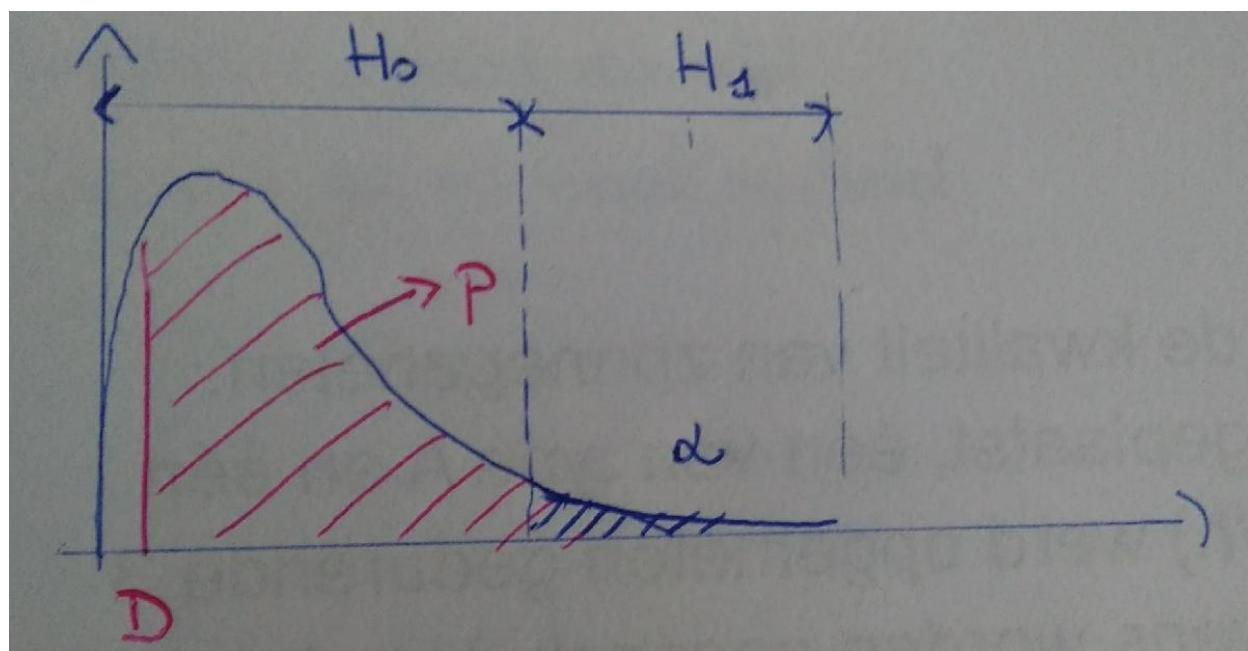
Om deze t-test uit te voeren moeten we eerst controleren of het verschil tussen de twee steekproeven normaal verdeeld is. Dit doen we aan de hand van een KS-test (zie oefening 59):

H_0 : de gegevens van vector x zijn normaal verdeeld

H_1 : de gegevens van vector x zijn niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p, D]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.9333
D = 0.1300
```



$p = 0.9333 > 0.05 \Rightarrow H_0$ aanvaarden: de gegevens van vector x zijn normaal verdeeld met 95% betrouwbaarheid.

Nu de normaliteit is aangetoond kunnen we verder gaan met de t-test. We voeren de one-sample tweezijdige t-test uit met als hypotheses:

$H_0 : \mu_d = 0$ met $\mu_d = \mu_A - \mu_B$

$H_1 : \mu_d \neq 0$ (tweezijdig)

We gebruiken het commando `ttest(x, 0)`, de parameter 0 gebruiken we omdat het theoretische verschil bij de nulhypothese gelijk is aan nul ($\mu_d = 0$).

```
[h,p,ci,stats]=ttest(x,0)
```

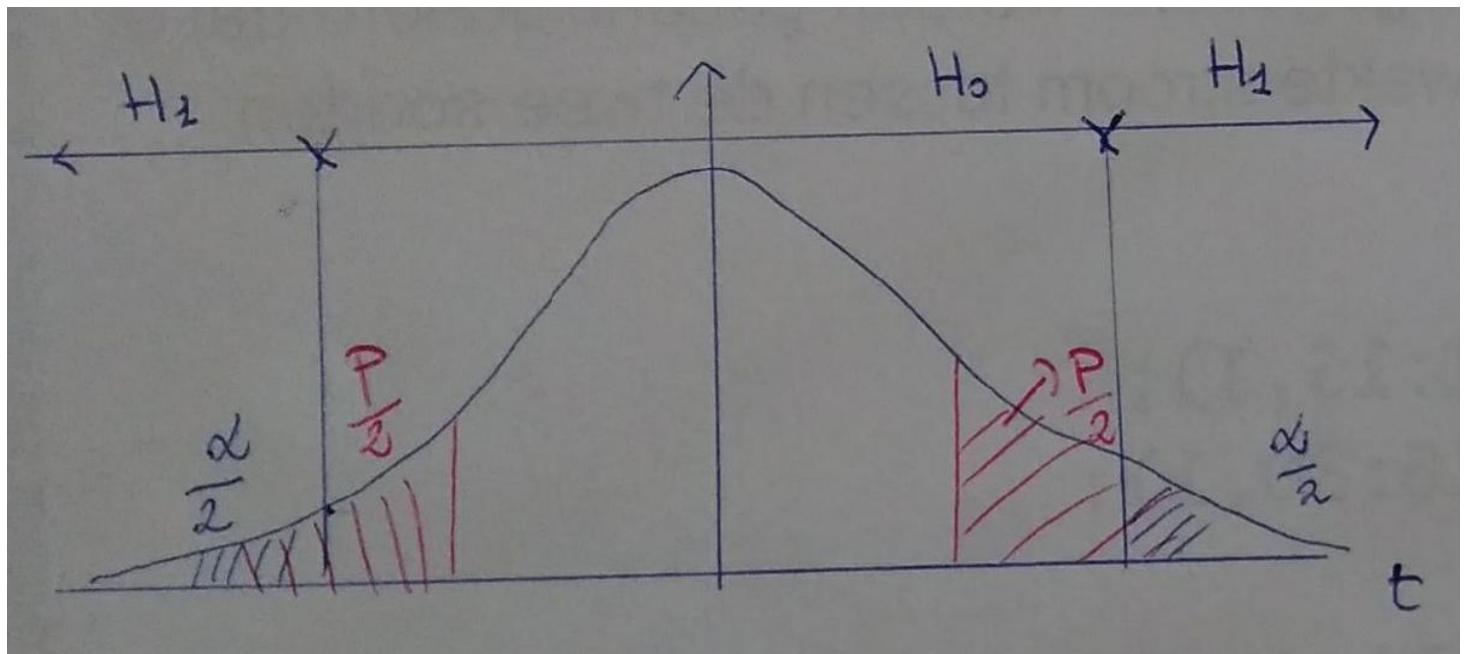
```
h = 0
```

```

p = 0.3461
ci = 2x1
-406.5805
152.4472
stats = struct with fields:
    tstat: -0.9750
    df: 14
    sd: 504.7366

```

$p = 0.3461 > 0.05 \Rightarrow H_0$ aanvaarden: de gemiddelde geleverde stroom door zonnepaneel type A is gelijk aan de gemiddelde geleverde stroom door zonnepaneel type B met 95% betrouwbaarheid (het gemiddelde verschil is niet significant verschillend van 0). Er kan dus met 95% betrouwbaarheid besloten worden dat er geen significant verschil is tussen de zonnepaneeltypes i.v.m. de geleverde stroom.



Oefening 69

Opgave:

Een consumentenorganisatie evalueert de kwaliteit van zonnepanelen. Daarvoor wordt op 30 daken één type zonnepaneel gelegd: ofwel type A ofwel type B. De geleverde stroom (in kWh) werd opgemeten gedurende 3 maanden. De data kan je vinden in zonnepanelen.txt. Kan op basis van deze gegevens worden geconcludeerd dat er een significant verschil is in opgewekte stroom tussen de twee soorten panelen? Maak voor beide groepen een box-and-Whisker plot en becommentarieer.

Oplossing:

Eerst en vooral importeren we de data uit de file zonnepanelen.txt (zie oefening 67). We maken twee vectoren aan met de geleverde stroom, één voor type A en één voor type B. Die noemen we $x1$ en $x2$.

```
x1=zonnepanelen.stroom(1:15,1);
x2=zonnepanelen.stroom(16:30,1);
```

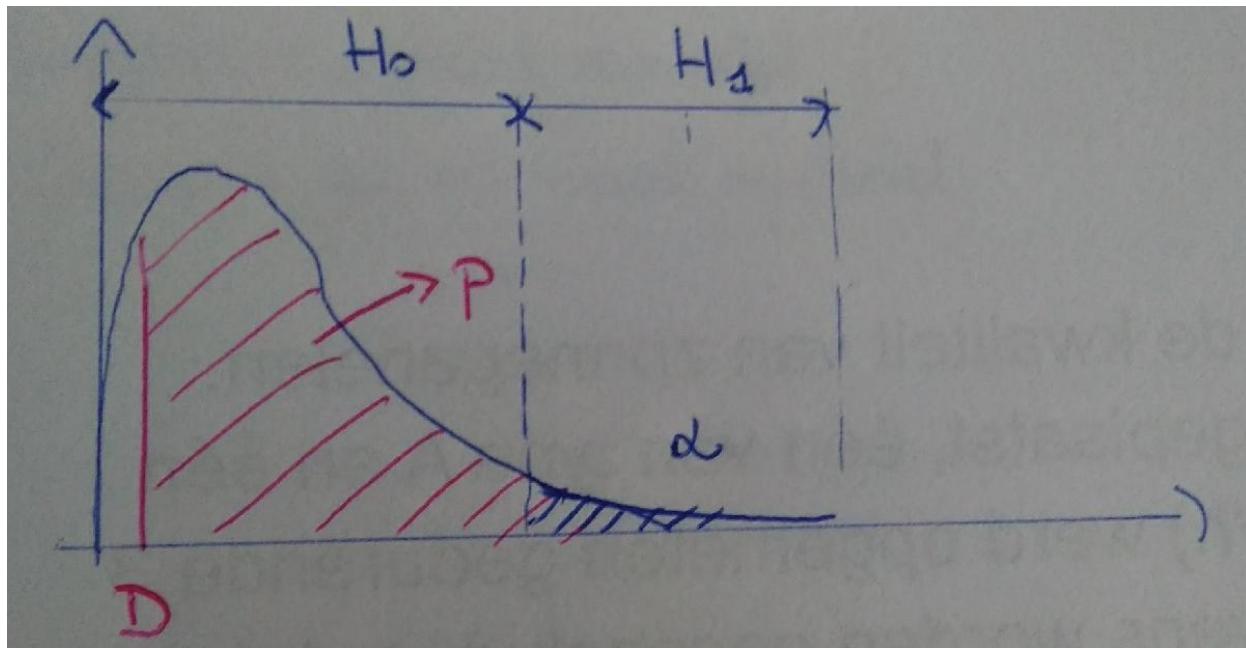
We moeten nagaan of de gegevens per groep (A en B) normaal verdeeld zijn. Dit doen we aan de hand van twee KS-testen (zie oefening 59), één voor type A en één voor type B.

H_0 : de gegevens zijn normaal verdeeld

H_1 : de gegevens zijn niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(x1),'sigma',std(x1));
[h,p, D]=kstest(x1,'CDF',test_cdf)
```

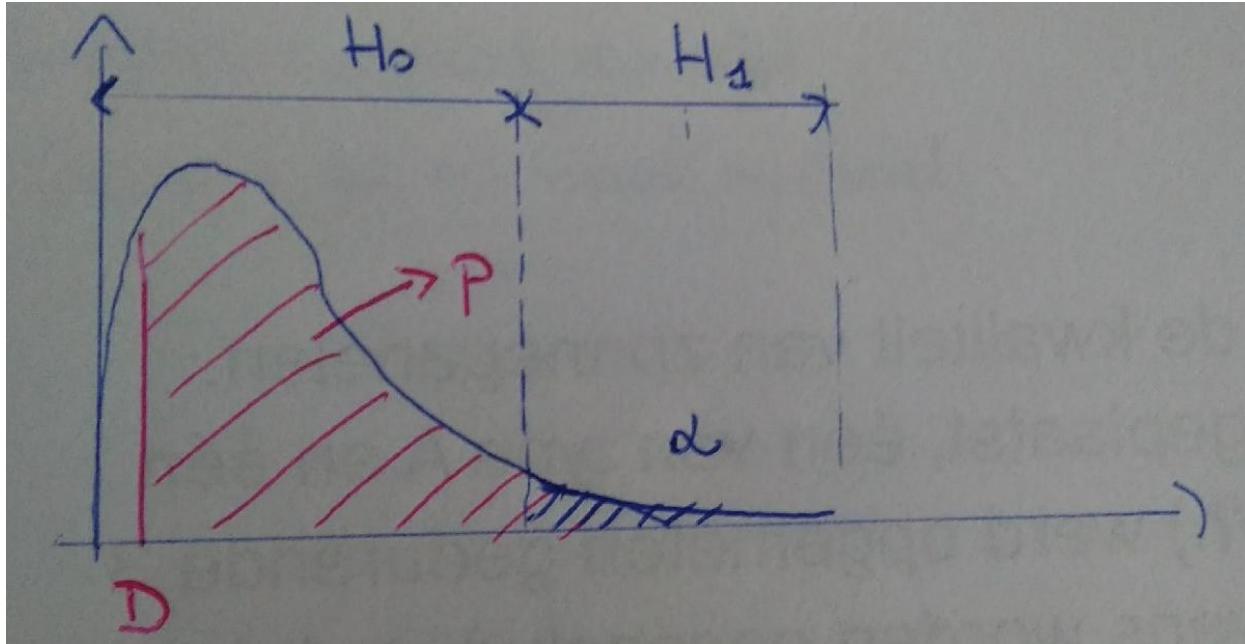
```
h = logical
0
p = 0.1342
D = 0.2884
```



$p = 0.1342 > 0.05 \Rightarrow H_0$ aanvaarden: de gegevens uit $x1$ zijn normaal verdeeld met 95% betrouwbaarheid.

```
test_cdf=makedist('Normal','mu',mean(x2),'sigma',std(x2));
[h,p, D]=kstest(x2,'CDF',test_cdf)
```

```
h = logical
0
p = 0.7755
D = 0.1610
```



$p = 0.7755 > 0.05 \Rightarrow H_0$ aanvaarden: de gegevens uit $x2$ zijn normaal verdeeld met 95% betrouwbaarheid.

Voor de t-test van twee onafhankelijke steekproeven moeten we eerst controleren of de twee steekproeven dezelfde variantie hebben. Dit doen we aan de hand van een Levene test met als hypothesen:

$$H_0: \sigma_A^2 = \sigma_B^2 \text{ (gelijke varianties)}$$

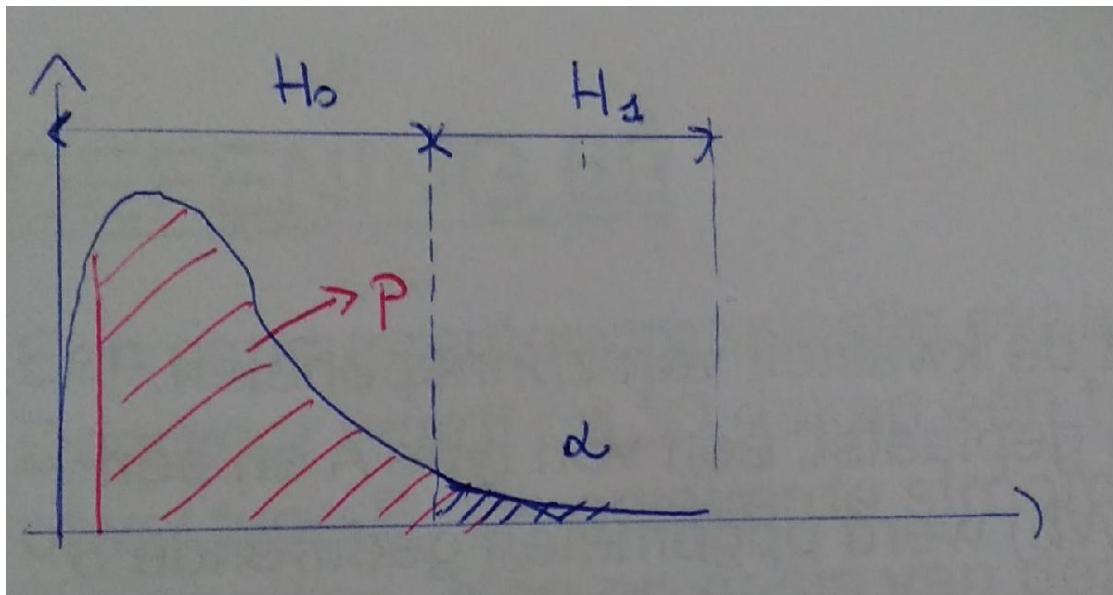
$$H_1: \sigma_A^2 \neq \sigma_B^2 \text{ (GEEN gelijke varianties)}$$

De Levene test roepen we op met het commando `vartestn`. In de argumenten specifiëren we dat het testtype Levene is (`LeveneAbsolute`) en dat we geen extra informatie willen naast de p-waarde (`Display off`).

```
p = vartestn([x1, x2], 'TestType', 'LeveneAbsolute', 'Display', 'off')
```

```
p = 0.9421
```

```
%Let op: x1 en x2 moeten kolomvectoren zijn!
```



$p = 0.9421 > 0.05 \Rightarrow H_0$ aanvaarden: vectoren x_1 en x_2 hebben gelijke variantie met 95% betrouwbaarheid.

De voorwaarden zijn voldaan en we kunnen nu de two-sample t-test uitvoeren.

$$H_0 : \mu_d = 0 \text{ met } \mu_d = \mu_A - \mu_B$$

$$H_1 : \mu_d \neq 0 \text{ (tweezijdig)}$$

Dit doen we aan de hand van `ttest2`, dit commando werkt op dezelfde manier als `ttest`, het enige verschil is dat er bij dit commando twee vectoren als argumenten worden meegegeven.

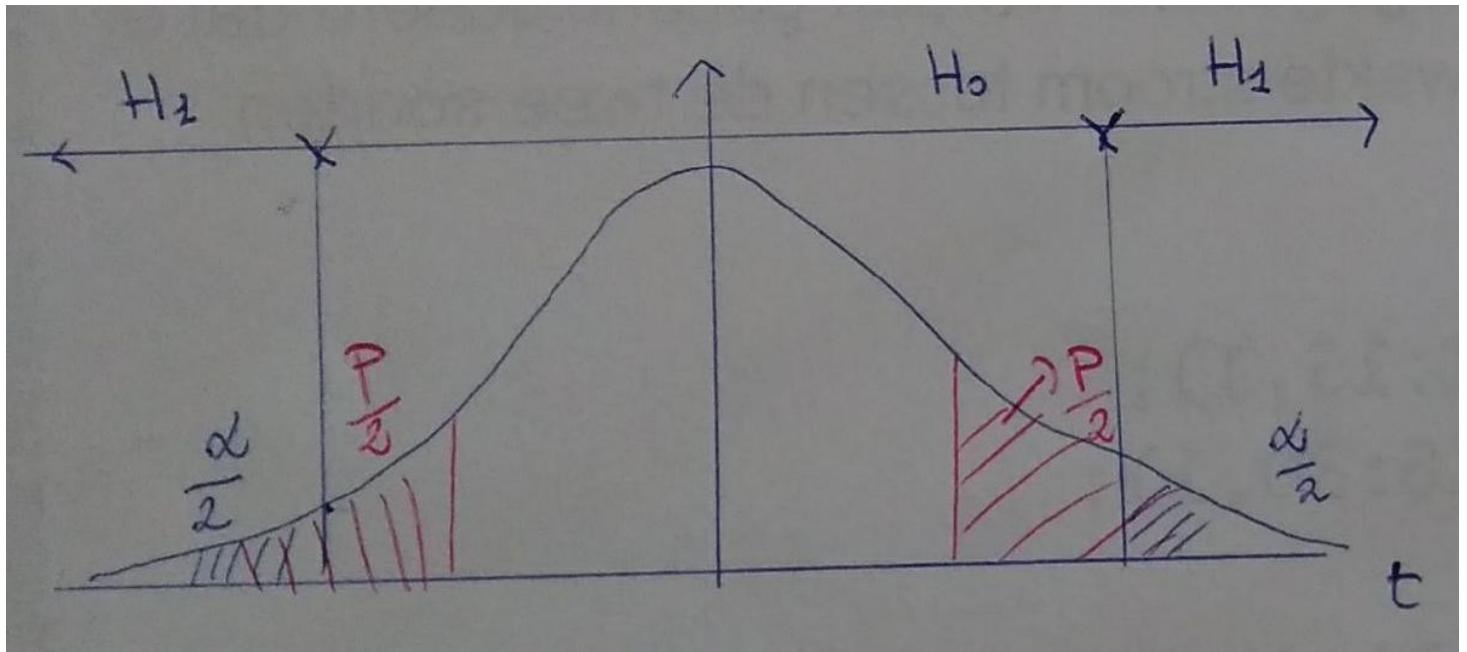
```
[h,p,ci, stats]=ttest2(x1,x2)
```

```

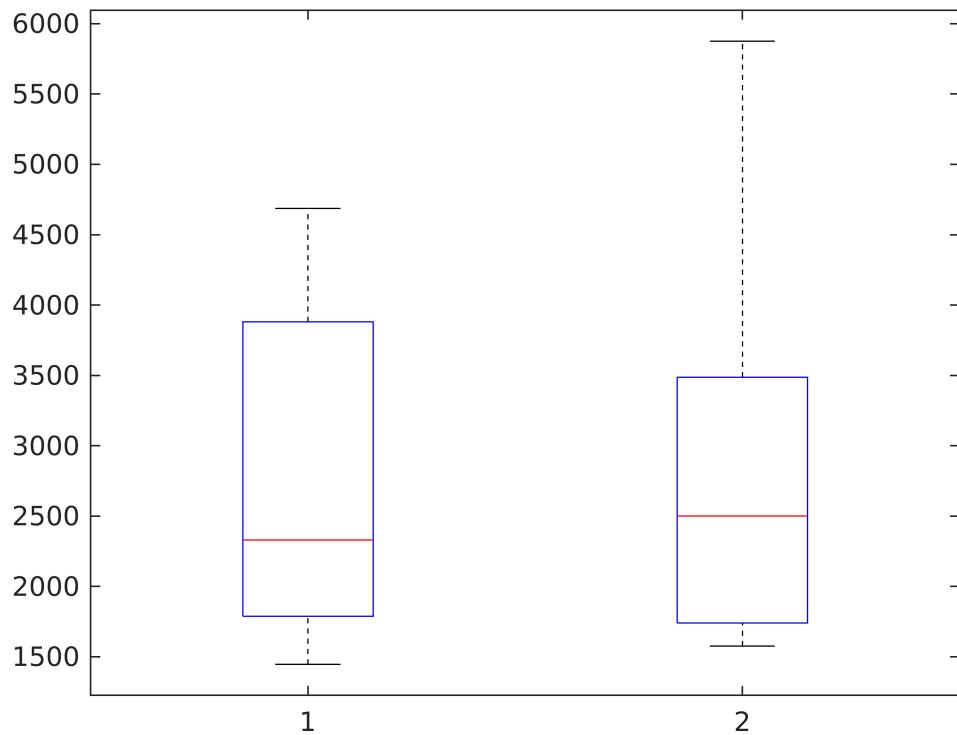
h = 0
p = 0.7714
ci = 2x1
10^3 *
-1.0142
 0.7601
stats = struct with fields:
    tstat: -0.2934
      df: 28
       sd: 1.1861e+03

```

$p = 0.7714 > 0.05 \Rightarrow H_0$ aanvaarden met 95% betrouwbaarheid: de gemiddelde geleverde stroom door zonnepaneel type A is gelijk aan de gemiddelde geleverde stroom door zonnepaneel type B met 95% betrouwbaarheid (het gemiddelde verschil is niet significant verschillend van 0). Er kan dus met 95% betrouwbaarheid besloten worden dat er geen significant verschil is tussen de zonnepaneeltypes i.v.m. de geleverde stroom.



```
boxplot([x1,x2])
```



Voor beide paneeltypen is de geleverde stroom scheef verdeeld met de staart naar boven. De mediaan van beide boxplotten (het middelste rode streepje in de doos) ligt sterk naar beneden. Er kan gezien worden dat de data voor beide types sterk overlapt, dit is een visuele bevestiging voor de uitkomst van de t-test.

Oefening 71

Opgave:

Voor het vervaardigen van synthetisch diamant wordt winst geboekt als de karaat-waarde > 0.5.

Een steekproef van 6 diamanten geeft:

Karaat-waarde 0.46 0.61 0.52 0.48 0.57 0.54

Is algemeen gesproken de productie de moeite waard (test op 85% niveau)?

Oplossing:

We slaan de gegevens in een vector op:

```
A=[0.46 0.61 0.52 0.48 0.57 0.54];
```

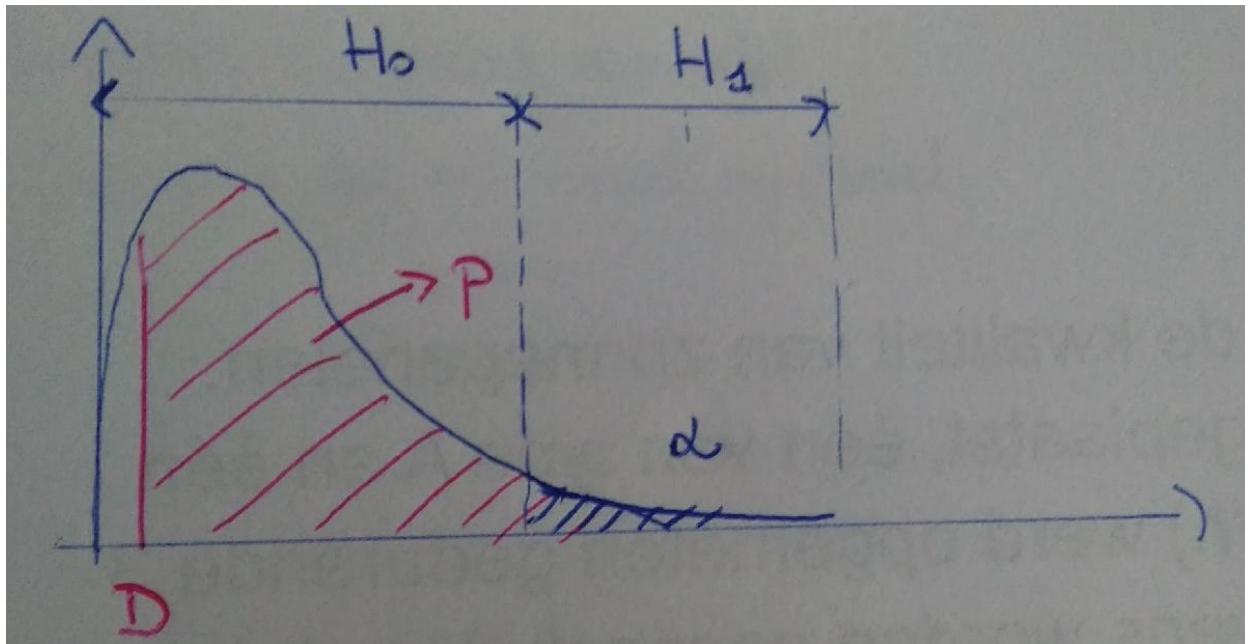
We veronderstellen dat het productieproces de moeite waard is indien de gemiddelde karaat-waarde boven 0.5 ligt. Dit moeten we controleren met een eenzijdige one-sample t-test. Voordat we deze t-test uitvoeren moeten we eerst controleren of de waarden van de steekproef normaal verdeeld zijn. Dit doen we aan de hand van een KS-test:

H_0 : gegevens zijn normaal verdeeld

H_1 : gegevens zijn niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(A),'sigma',std(A));
[h,p, stats]=kstest(A,'CDF',test_cdf, 'alpha', 0.15)
```

```
h = Logical
0
p = 0.9880
stats = 0.1480
```



$p = 0.9880 > 0.15 \Rightarrow H_0$ aanvaarden: de karaat-waarden zijn normaal verdeeld met 85% betrouwbaarheid.

De gegevens zijn normaal verdeeld en dus is de nodige voorwaarde voor de t-test voldaan. De hypotheses voor de eenzijdige one-sample t-test zijn als volgt:

$$H_0 : \mu = 0.5$$

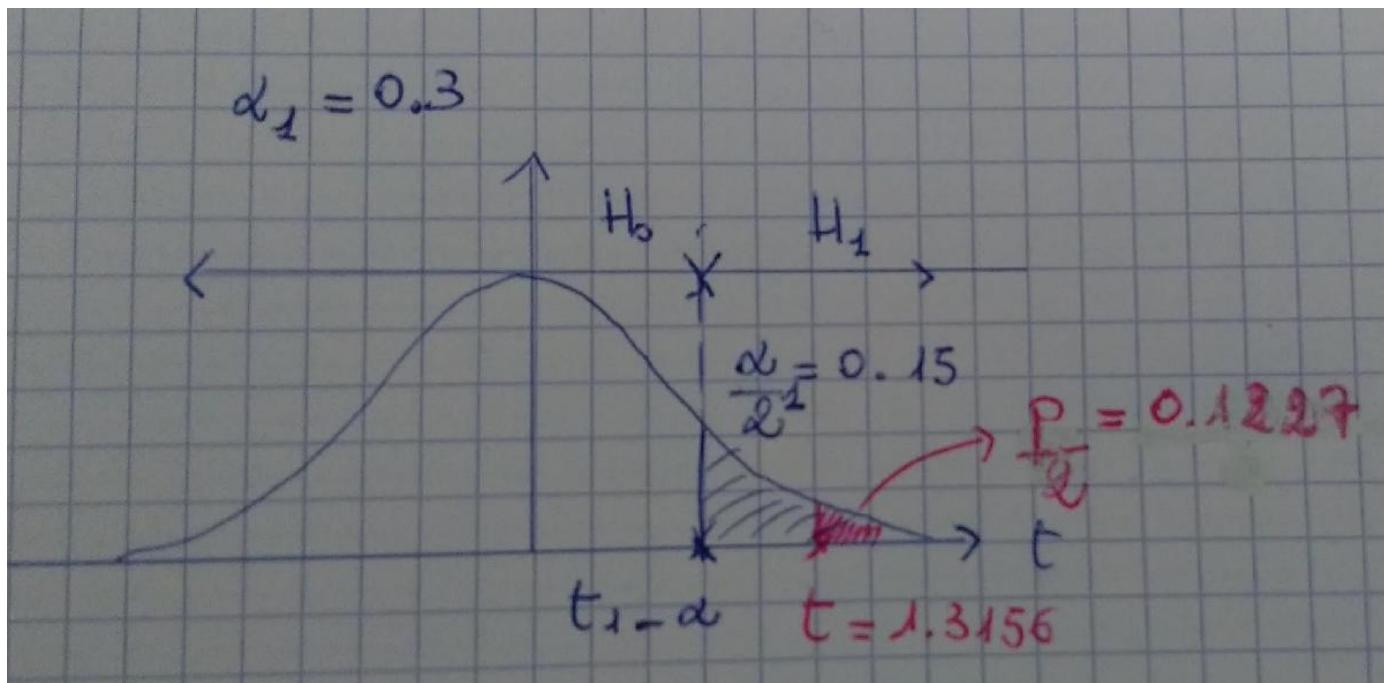
$$H_1 : \mu > 0.5 \text{ (éénzijdig)}$$

Om de t-test éénzijdig toe te passen kunnen we twee technieken gebruiken.

Eerste techniek: we kunnen een tweezijdige t-test gebruiken met $\alpha_1 = 2 \times \alpha = 0.3$ (let wel op dat de p-waarde standaard gegeven wordt voor een tweezijdige versie van de test en moet daarom gehalveerd worden om te kunnen vergelijken met α).

```
[h,p,ci, stats]=ttest(A,0.5,'Alpha',0.3)
```

```
h = 1
p = 0.2454
ci = 1x2
    0.5036    0.5564
stats = struct with fields:
    tstat: 1.3156
    df: 5
    sd: 0.0559
```



$\frac{p}{2} = 0.1227 < 0.15 \Rightarrow H_0$ verwerpen en H_1 aanvaarden met 85% betrouwbaarheid. Het betrouwbaarheidsinterval

dat we met deze techniek krijgen is in feite het 70%-betrouwbaarheidsinterval $[0.5036, 0.5564]$. We zijn in dit geval niet geïnteresseerd in de bovengrens van μ en dus kunnen we de bovengrens van dit interval gelijkstellen

aan $+\infty$, wat van het 70%-betrouwbaarheidsinterval $[0.5036, 0.5564]$ het 85%-betrouwbaarheidsinterval $[0.5036, +\infty]$ maakt. Merk op dat 0.5 niet in dit 85%-betrouwbaarheidsinterval ligt, zoals is te verwachten aangezien we H_0 hebben verworpen.

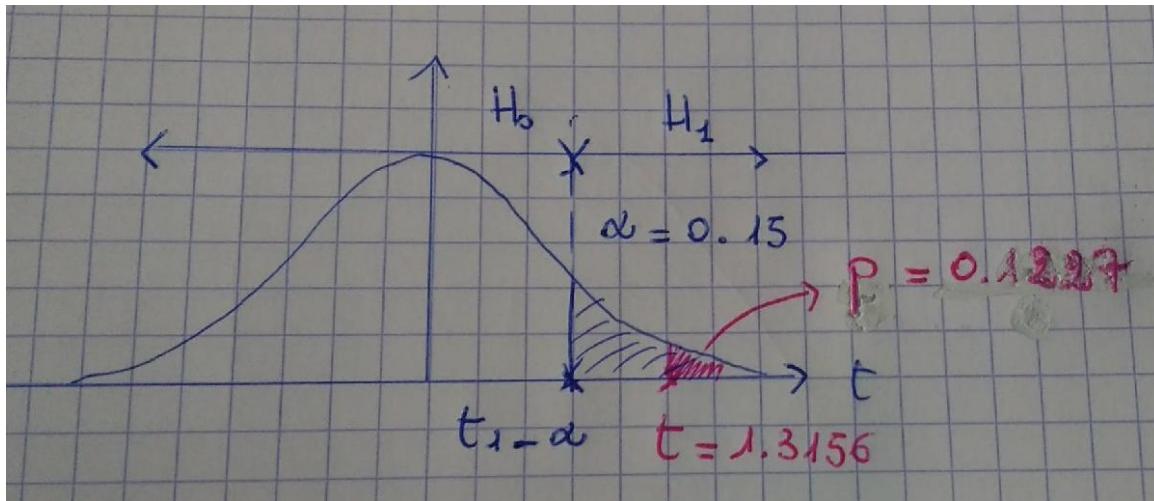
Tweede techniek: we kunnen het extra argument 'Tail', 'right' gebruiken om expliciet een rechtszijdige t-test uit te voeren; we gebruiken 'right' omdat de alternatieve hypothese is dat $\mu > 0.5$ (als $H_1 : \mu < 0.5$ zou zijn, dan moeten we een linkszijdige test doen met 'Tail', 'left').

```
[h,p,ci, stats]=ttest(A,0.5,'Alpha',0.15, 'Tail', 'right')
```

```

h = 1
p = 0.1227
ci = 1x2
    0.5036      Inf
stats = struct with fields:
    tstat: 1.3156
    df: 5
    sd: 0.0559

```



$p = 0.1227 < 0.15 \Rightarrow H_0$ verwijzen en H_1 aanvaarden met 85% betrouwbaarheid. Merk op dat 0.5 niet in het 85%-betrouwbaarheidsinterval $[0.5036, +\infty]$ ligt, zoals is te verwachten aangezien we H_0 hebben verworpen. Hieruit volgt dat het productieprocess de moeite waard is met 85% betrouwbaarheid.

Oefening 7

Opgave:

Bereken op een efficiënte manier de som van de eerste honderd kwadraten: $1^2 + 2^2 + \dots + 100^2$.

Oplossing:

We definiëren eerst een rij met startwaarde 1, een stap van 1 en eindwaarde 100. Als er in de code geen stapwaarde gegeven wordt dan is dit automatisch 1.

```
v=1:100;
```

We definiëren een nieuwe rij waarvan de elementen de overeenkomstige kwadraten zijn van de elementen uit de oorspronkelijke rij. Hiervoor moeten we een **elementsgewijze** machtverheffing uitvoeren (\cdot^2). Als we dit puntje niet zouden noteren dan berekenen we gewoon het kwadraat van de vector zelf.

```
v2 = v.^2;
```

Als laatste stap berekenen we de som van de elementen van de rij v2 met behulp van het commando sum:

```
sum(v2)
```

```
ans = 338350
```

Oefening 73

Opgave:

Onderzoekers willen nagaan of mensen die niet sporten een hogere hartslag hebben. Hiervoor doen ze metingen bij veertien verschillende personen, zeven die twee uur per week sporten en zeven die geen sport doen. De gegevens vind je terug in onderstaande tabel:

Hartslag	1	2	3	4	5	6	7
Sport	55	59	60	58	61	62	65
Geen Sport	57	62	62	62	60	64	64

Kan op basis van deze steekproef gesteld worden dat de hartslag op populatieniveau hoger is voor de groep die geen sport doet ($\alpha = 0.05$)?

Oplossing:

```
sport=[55 59 60 58 61 62 65];
geen_sport=[57 62 62 62 60 64 64];
```

Er moet hier een éénzijdige t-test voor twee onafhankelijke steekproeven uitgevoerd worden. Vooraleer dit uitgevoerd mag worden, moet de voorwaarde van het normaal verdeeld zijn van de hartslag in beide groepen gecontroleerd worden. Er moet ook nagegaan worden of de varianties gelijk zijn.

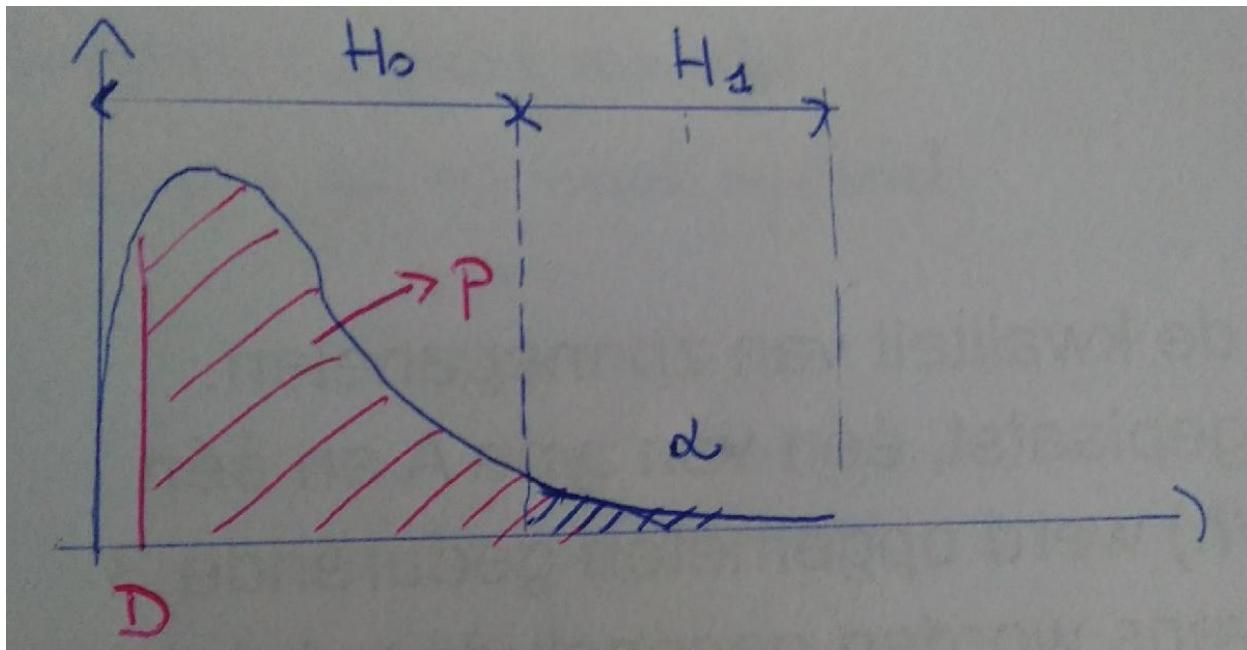
Voor het controleren van normaliteit gebruiken we de KS-test:

H_0 : hartslag is normaal verdeeld

H_1 : hartslag is niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(sport),'sigma',std(sport));
[h,p]=kstest(sport,'CDF',test_cdf)
```

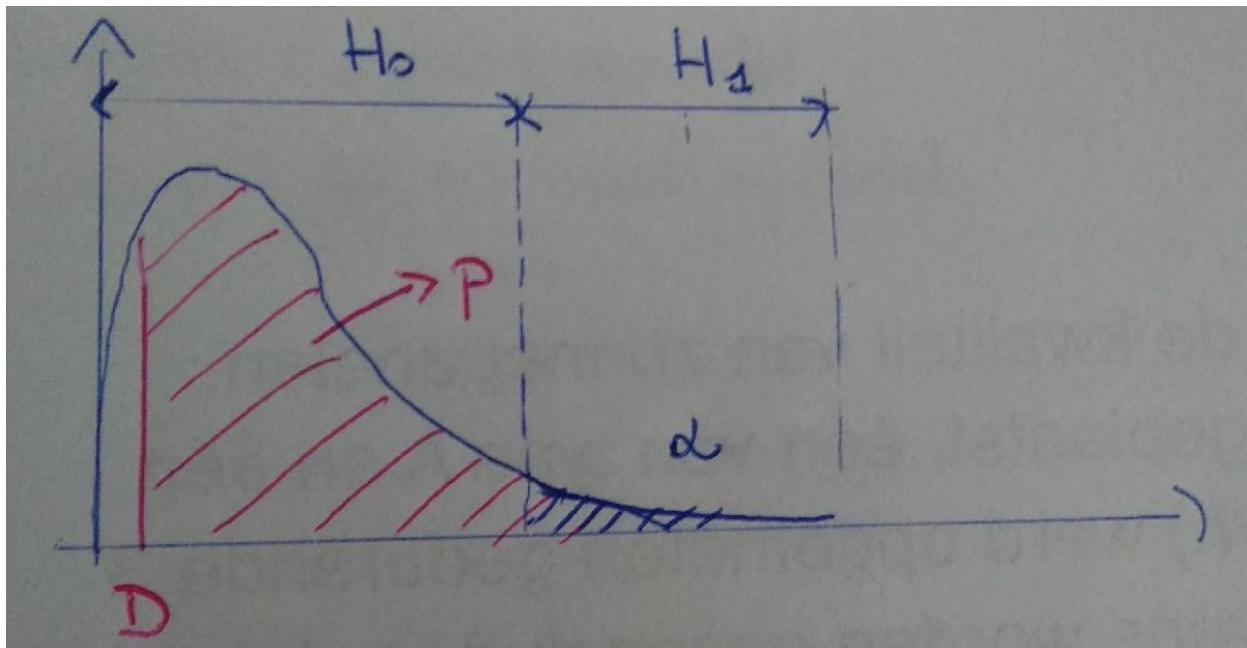
```
h = logical
0
p = 0.9958
```



Groep 1 (2 uren sport per week): $p = 0.9958 > 0.05 \Rightarrow H_0$ aanvaarden: hartslag is normaal verdeeld met 95% betrouwbaarheid

```
test_cdf=makedist('Normal','mu',mean(geen_sport),'sigma',std(geen_sport));
[h,p]=kstest(geen_sport,'CDF',test_cdf)
```

```
h = logical
0
p = 0.5328
```



Groep 2 (0 uren sport per week): $p = 0.5328 > 0.05 \Rightarrow H_0$ aanvaarden: hartslag is normaal verdeeld met 95% betrouwbaarheid

Voor het testen van gelijke variantie gebruiken we Levene's test:

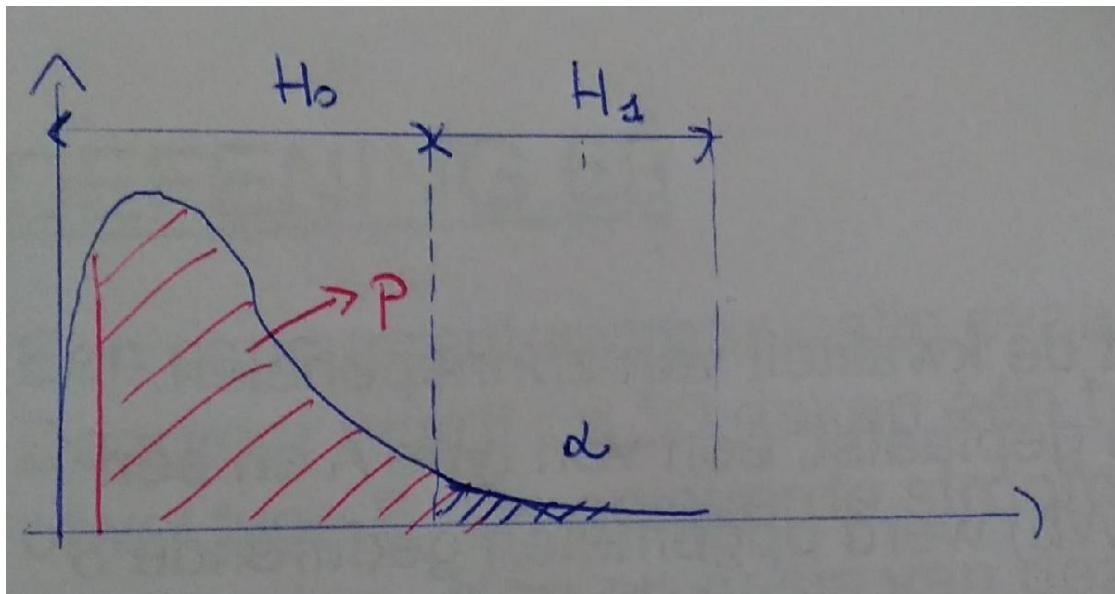
$$H_0 : \sigma_A^2 = \sigma_B^2 \quad (\text{gelijke varianties})$$

$$H_1 : \sigma_A^2 \neq \sigma_B^2 \quad (\text{GEEN gelijke varianties})$$

```
p = vartestn(['sport', 'geen_sport'], 'TestType', 'LeveneAbsolute', 'Display', 'off')
```

p = 0.5852

%Let op: hier worden sport en geen_sport getransponeerd zodat het kolomvectoren worden
%dit is noodzakelijk voor deze methode!



$p = 0.5852 > 0.05 \Rightarrow H_0$ aanvaarden: gelijke varianties met 95% betrouwbaarheid

De voorwaarden zijn voldaan en we kunnen nu de enkelzijdige two-sample t-test uitvoeren. In het geval van een enkelzijdige two-sample t-test zal Matlab het gemiddelde van de tweede variabele aftrekken van het gemiddelde van de eerste variabele. Met andere woorden: als we willen aantonen dat de hartslag bij sport kleiner is dan die bij geen sport en we kiezen ze als respectievelijk eerste en tweede argument dan moeten we een linkszijdige two-sample t-test doen. Net zoals bij oefening 71 kunnen we de eenzijdige t-test op twee verschillende manieren doen.

$$H_0 : \mu_{\text{sport}} = \mu_{\text{geen sport}}$$

$$H_1 : \mu_{\text{sport}} < \mu_{\text{geen sport}} \text{ (éénzijdig)}$$

Eerste techniek:

Bij de eerste techniek moeten we werken met $\alpha_1 = 2 \times \alpha = 0.10$.

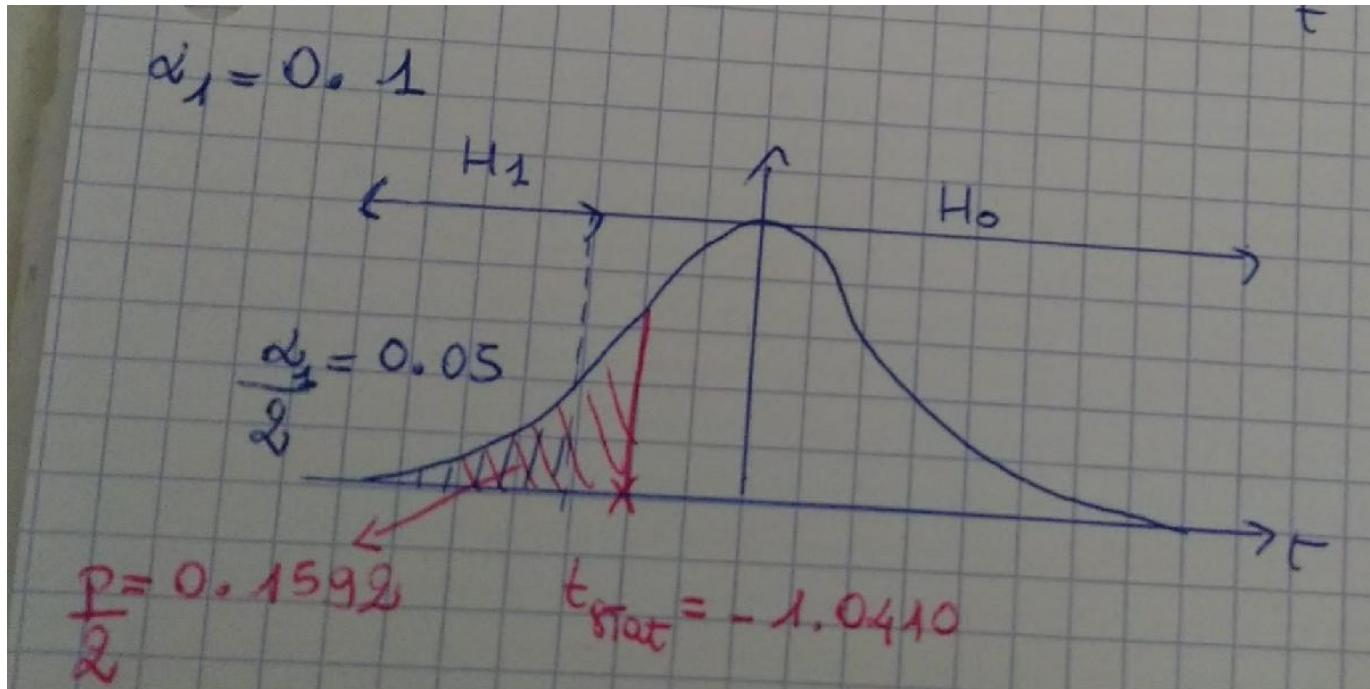
```
[h,p,ci, stats]=ttest2(sport,geen_sport, 'Alpha', 0.10)
```

h = 0
p = 0.3184

```

ci = 1x2
-4.2620    1.1191
stats = struct with fields:
tstat: -1.0410
df: 12
sd: 2.8242

```



$\frac{P}{2} = 0.1592 > 0.05 \Rightarrow H_0$ aanvaarden met 95% betrouwbaarheid, er is geen significant verschil van de hartslag bij sporten en niet sporten.

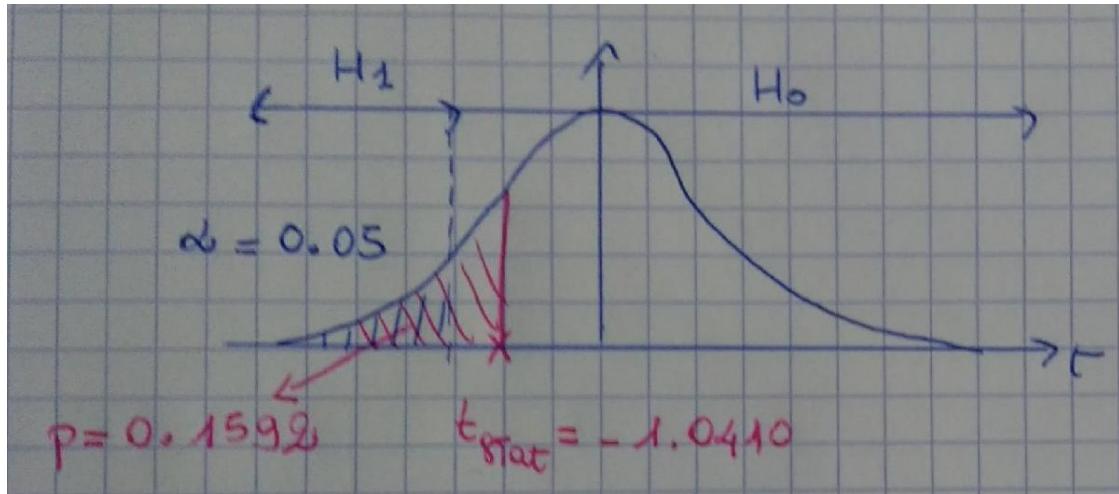
Tweede techniek:

```
[h,p,ci, stats]=ttest2(sport,geen_sport, 'Tail', 'left')
```

```

h = 0
p = 0.1592
ci = 1x2
-Inf    1.1191
stats = struct with fields:
tstat: -1.0410
df: 12
sd: 2.8242

```



$p = 0.1592 > 0.05 \Rightarrow H_0$ aanvaarden met 95% betrouwbaarheid, er is geen significant verschil van de hartslag bij sporten en niet sporten.

Oefening 75

Opgave:

Een onderzoek werd opgericht om na te gaan of de leeftijd van de autobestuurder invloed heeft op zijn rijgedrag, met name het aantal auto-ongevallen waarin hij betrokken geraakt gedurende een jaar. Test met een betrouwbaarheid van 95% of de leeftijd van de bestuurder invloed heeft op het aantal ongevallen. De data vind je in rijgedrag.dat.

Oplossing:

Eerst en vooral moeten we de data inlezen via 'Import Data'. De vector met de leeftijdscategoriën (genummerd van 2 tot en met 5) noemen we x , de vector met het respectievelijk aantal ongevallen noemen we y .

```
x=rijgedrag.leeftijd;
y=rijgedrag.aantal_ongevallen;
```

Voor het testen van de afhankelijkheid moeten we een contingentietabel en een χ^2 -test gebruiken. Deze test heeft volgende hypotheses:

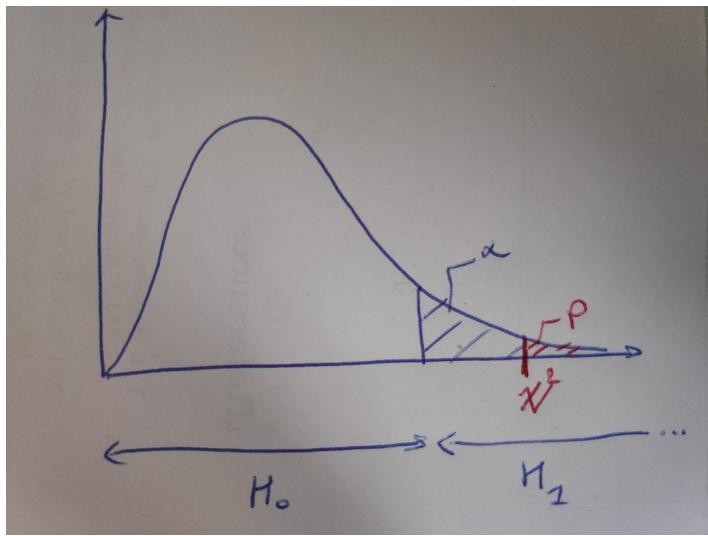
H_0 : Er bestaat geen afhankelijkheid tussen de leeftijd en het aantal ongevallen.

H_1 : Er bestaat een afhankelijkheid tussen de leeftijd en het aantal ongevallen.

In Matlab bestaat er één commando voor de contingentietabel en de χ^2 -test, deze is `crosstab`. Deze geeft ons de contingentietabel die we `table` noemen en de resultaten van een χ^2 -test: de χ^2 -statistiek die we `chi2` noemen en de p-waarde van de test die we `p` noemen. De rijen van `table` komen overeen met de leeftijdscategoriën en de kolommen met het aantal accidenten.

```
[table,chi2,p]=crosstab(x,y)
```

```
table = 4x4
    44    16    10     3
    25    16     6     6
    10    12    15    12
    11    14    17     8
chi2 = 38.0654
p = 1.6966e-05
```



$p = 1.6966 \times 10^{-5} < 0.05 \Rightarrow H_0$ verwerpen en H_1 aanvaarden: er bestaat een afhankelijkheid tussen de leeftijd en het aantal ongevallen met 95% betrouwbaarheid.

Oefening 77

Opgave:

Men wenst te onderzoeken wie kiest voor het bouwen van passieve woningen. Daarvoor heeft men gepeild naar het opleidingsniveau van de mensen die een woning bouwen.

	1 groep 1 = hoger opgeleiden	groep 2 = middengroep	groep 3 = laaggeschoolden
Passieve woning	14	30	11
Geen passieve woning	44	142	257

1. Ga je op basis van deze steekproef akkoord met de stelling dat er geen verschil is qua opleidingsniveau bij de keuze voor een passieve woning? (onbetrouwbaarheidsdrempel = 5%)?
2. Hoeveel procent van de passieve woningen worden gebouwd door laaggeschoolden?
3. Hoeveel procent van de hoger opgeleiden kiest voor een passieve woning?

Oplossing:

Indien de gegevens in de vorm van een contingentietabel zijn gegeven moeten we de χ^2 -test met een omweggetje laten uitvoeren. We moeten eerst de data voorstellen als twee lange kolomvectoren door gebruik te maken van repmat. Het commando repmat genereert matrices waarbij alle elementen dezelfde waarde hebben en heeft drie argumenten nodig: het eerste argument is het element dat we willen herhalen, het tweede argument is het aantal rijen van de matrix en het derde element is het aantal kolommen (hiervoor gebruiken we telkens 1).

We definiëren eerst drie variabelen die overeen komen met het totaal aantal mensen per soort opleiding:

```
N1 = 58; N2 = 172; N3 = 268;
```

We definiëren ook drie andere variabelen die overeenkomen met het aantal mensen in de eerste rij van de tabel:

```
n1 = 14; n2 = 30; n3 = 11;
```

De tweede rij van de tabel kan dan berekend worden door het verschil van de eerste drie variabelen met de tweede drie.

De eerste kolomvector die we gaan maken is x en deze staat voor de parameter 'soort opleiding'. Er zijn 58 hoger opgeleiden (groep 1), 172 mensen zitten in de middengroep (groep 2) en 268 mensen zijn laaggeschoold (groep 3). In de vector x moeten dus de eerste 58 elementen een 1 zijn, symbool voor groep 1. Analoog voor groep 2 (172 2's) en voor groep 3 (268 3's).

De tweede kolomvector is de vector y , hierin wordt de parameter van het soort woning opgeslaan (1 voor 'passief' en 2 voor 'niet passief'). De elementen in de y vector komen respectievelijk overeen met de elementen uit de x vector. De eerste 58 elementen van y komen overeen met de groep van hoger opgeleiden (groep

1), hiervan hebben er 14 een passieve woning, wat weergegeven wordt door een 1 en 44 een niet passieve woning, wat weergegeven wordt door een 2. Voor groep 2 en 3 moet hetzelfde gebeuren: 30 1'en en 142 2's voor groep 2 en 11 1'en en 257 2's voor groep 3.

```
x = [repmat(1,N1,1); repmat(2,N2,1); repmat(3,N3,1)];
y = [repmat(1,n1,1); repmat(2,N1-n1,1); repmat(1,n2,1);
      repmat(2,N2-n2,1); repmat(1,n3,1); repmat(2,N3-n3,1)];
```

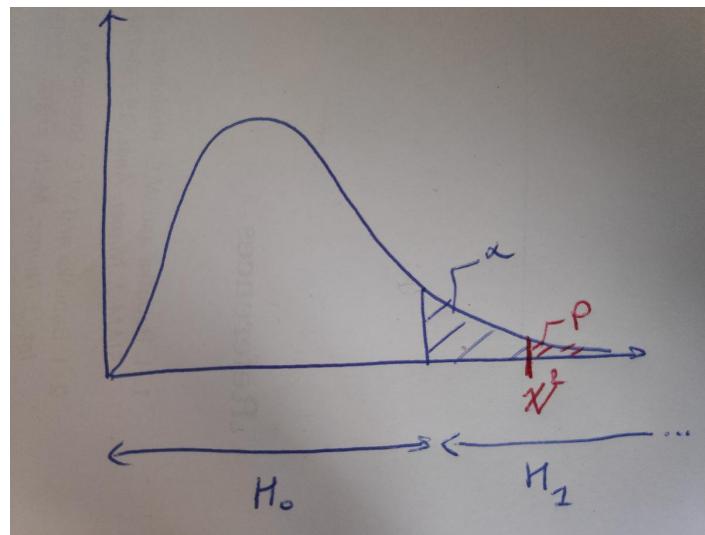
1. Eens we x en y gedefinieerd hebben kunnen we overgaan tot de χ^2 -test met crosstab:

H_0 : Er bestaat geen afhankelijkheid tussen de woning (passieve of geen passieve woning) en het opleidingsniveau.

H_1 : Er bestaat een afhankelijkheid tussen de woning (passieve of geen passieve woning) en het opleidingsniveau.

```
[table,chi2,p]=crosstab(y,x)
```

```
table = 2x3
    14      30      11
    44     142     257
chi2 = 30.4248
p = 2.4737e-07
```



$p = 2.4737 \times 10^{-7} < 0.05 \Rightarrow H_0$ verwerpen en H_1 aanvaarden: met 95% betrouwbaarheid kunnen we besluiten dat er een afhankelijkheid is tussen de soort woning en het opleidingsniveau.

2. Voor het procent van de passieve woningen gebouwd door laaggeschoolden te berekenen maken we gebruik van volgende formule:

$$\frac{\text{aantal pas. woningen dr. laaggesch.}}{\text{totaal aantal pas. woningen}} \times 100\%$$

$$(n3/(n1+n2+n3))*100$$

ans = 20

3. Het procent van de hoger opgeleiden die kiezen voor een passieve woning berekenen we als volgt:

$$\frac{\text{aantal pas. woningen dr. hooggesch.}}{\text{totaal aantal woningen dr. hooggesch.}} \times 100\%$$

$$(n1/N1)*100$$

ans = 24.1379

Oefening 79

Opgave:

Een onderzoeker is geïnteresseerd in de breeksterkte van verschillende gelamineerde balken gemaakt uit drie houtvariëteiten en drie verschillende soorten lijm. Om deze te vergelijken werden vijf balken van elk van de negen combinaties aangemaakt en daarna onderworpen aan een spanningstest. De data bestaat uit waarden voor de druk waarbij de balken braken. Maak een vergelijkend onderzoek (met two-way ANOVA) en controleer eventuele assumpties. Zie cursus paragraaf 8.4, blz. 105. De data voor deze oefening vind je in `balken.dat`.

Oplossing:

We beginnen met de data te importeren en de gegevens voor druk op te slaan in een matrix A, dit doen we op een manier die ANOVA achteraf gemakkelijker maakt. In de drie kolommen staan de gegevens horende bij de verschillende lijmsoorten: de eerste kolom komt overeen met de eerste lijmsoort, enzovoort. In de rijen de gegevens van de verschillende houtvariëteiten. De eerste vijf rijen van de matrix komen overeen met de eerste houtsoort, de tweede vijf rijen met de tweede houtsoort en de derde vijf met de derde houtsoort.

```
A(:,1)=balken.pressure(1:15);  
A(:,2)=balken.pressure(16:30);  
A(:,3)=balken.pressure(31:45);  
A
```

```
A = 15x3  
196 214 258  
208 216 250  
207 235 234  
216 240 248  
221 252 202  
216 215 246  
228 217 247  
240 235 264  
224 219 250  
236 181 255  
:  
:
```

We moeten nagaan of de kwantitatieve variabele pressure (druk) normaal verdeeld is over de verschillende niveaus van factor 1 (lijm = glue). Daarnaast moeten we ook nagaan of de variabele pressure normaal verdeeld is over de verschillende niveaus van factor 2 (houtsoort = wood). Voor het testen van normaliteit gebruiken we een KS-test:

H_0 : normaal verdeeld

H_1 : niet normaal verdeeld

Eerste lijmsoort (komt overeen met de eerste kolom):

```
x=A(:,1);  
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));  
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical  
0
```

```
p = 0.9273
```

Tweede lijmsoort (komt overeen met de tweede kolom):

```
x=A(:,2);
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.4125
```

Derde lijmsoort (komt overeen met de derde kolom):

```
x=A(:,3);
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.1036
```

Eerste houtsoort (komt overeen met de eerste vijf rijen):

```
x=[A(1,:),A(2,:),A(3,:),A(4,:),A(5,:)]
```

```
x = 1x15
196 214 258 208 216 250 207 235 234 216 240 248 221 ...
```

```
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.7448
```

Tweede houtsoort (komt overeen met rij zes tot tien):

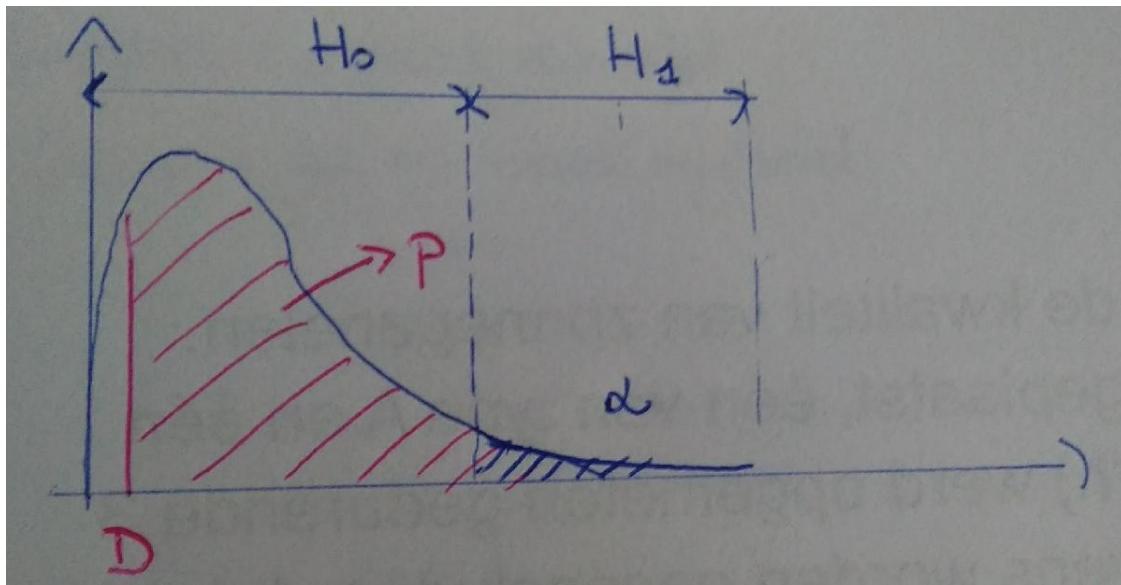
```
x=[A(6,:),A(7,:),A(8,:),A(9,:),A(10,:)];
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.8639
```

Derde houtsoort (komt overeen met rij elf tot vijftien):

```
x=[A(11,:),A(12,:),A(13,:),A(14,:),A(15,:)];
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.9300
```



Uit de verschillende KS-testen volgt dat de normaliteit in alle gevallen geaccepteerd wordt met 95% betrouwbaarheid, omdat telkens $p > \alpha$.

Het testen van gelijke variantie kunnen we doen met Levene's test. Dit moeten we doen voor zowel de verschillende houtsoorten als de verschillende soorten lijm.

$$H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 \quad (\text{gelijke varianties})$$

$$H_1 : \text{Niet alle varianties zijn gelijk}$$

De methode die we hiervoor gebruiken is vartestn. Deze vergelijkt de variantie van de kolommen van de gegeven matrix. We passen dit eerst toe voor de verschillende soorten lijm (daarvoor moeten we de kolommen van A vergelijken, dit doet vartestn standaard)

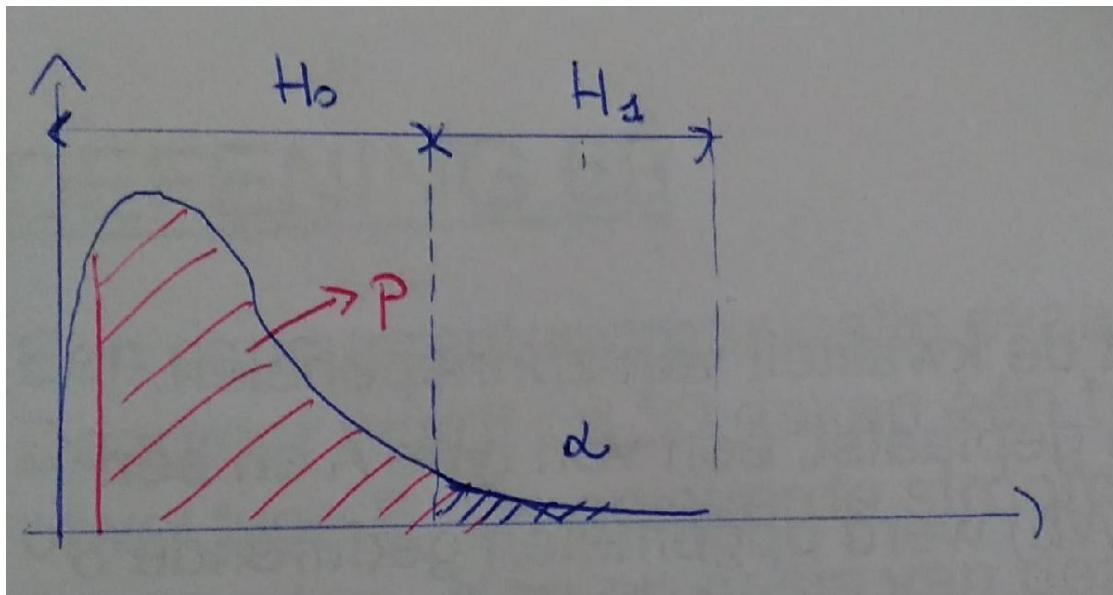
```
p = vartestn(A, 'TestType', 'LeveneAbsolute', 'Display', 'off')
```

```
p = 0.8024
```

Nu doen we het ook voor de verschillende soorten hout, hiervoor moeten we echter de matrix A hervormen zodat de verschillende houtsoorten in de kolommen staan.

```
p = vartestn([[A(1,:),A(2,:),A(3,:),A(4,:),A(5,:)]', ...
[A(6,:),A(7,:),A(8,:),A(9,:),A(10,:)]', ...
[A(11,:),A(12,:),A(13,:),A(14,:),A(15,:)]']', ...
'TestType', 'LeveneAbsolute', 'Display', 'off')
```

```
p = 0.6147
```



Voor beide testen is $p > \alpha$ en de variantie bij de verschillende hout- en lijmsoorten wordt met 95% betrouwbaarheid geaccepteerd als gelijk. De groepen hebben gelijke varianties en zijn normaal verdeeld, de voorwaarden voor two-way ANOVA zijn dus voldaan.

Voor de two-way ANOVA hebben we 3 nulhypothesen:

Test 1:

$H_0 : \mu_1 = \mu_2 = \mu_3$ (geen verschil in gemiddelde druk voor de 3 verschillende lijmsoorten)

$H_1 : \text{niet alle populatiegemiddelden zijn gelijk}$

Test 2:

$H_0 : \mu_1 = \mu_2 = \mu_3$ (geen verschil in gemiddelde druk voor de 3 verschillende houtsoorten)

$H_1 : \text{niet alle populatiegemiddelden zijn gelijk}$

Test 3:

$H_0 : \text{er is GEEN interactie}$

$H_1 : \text{er is interactie}$

De two-way ANOVA wordt gedaan met de methode `anova2`. De eerste parameter van deze methode is `A`, die staat voor de datamatrix, de tweede parameter is 5 omdat er vijf drukmetingen werden gedaan voor elk van de negen combinaties van lijmsoort en houtsoort.

```
[p,table,stats]=anova2(A,5)
```

ANOVA Table

Source	SS	df	MS	F	Prob>F
<hr/>					
Columns	6643.73	2	3321.87	20.12	0
Rows	682.53	2	341.27	2.07	0.1413
Interaction	2715.73	4	678.93	4.11	0.0076
Error	5944	36	165.11		
Total	15986	44			

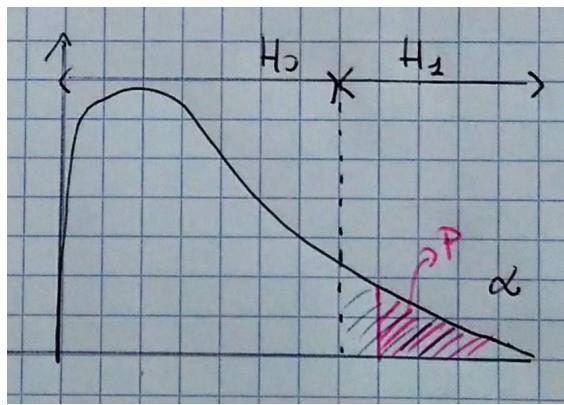
```
p = 1x3
    0.0000    0.1413    0.0076
table = 6x6 cell
```

	1	2	3	4	5	6
1	'Source'	'SS'	'df'	'MS'	'F'	'Prob>F'
2	'Columns'	6.6437e+03		3.3219e+03	20.1190	1.3626e-06
3	'Rows'	682.5333		341.2667	2.0669	0.1413
4	'Interaction'	2.7157e+03		678.9333	4.1120	0.0076
5	'Error'	5944		165.1111	[]	[]
6	'Total'	1.5986e+04		44	[]	[]

```
stats = struct with fields:
    source: 'anova2'
    sigmasq: 165.1111
    colmeans: [224.5333 221.0667 248.4000]
    coln: 15
    rowmeans: [226.4667 231.5333 236]
    rown: 15
    inter: 1
    pval: 0.0076
    df: 36
```

Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For more information, click here.

De anova2 methode geeft ons drie outputs terug. Het eerste zijn de p-waarden van testen 1, 2 en 3. Het tweede is de ANOVA-tabel met de gekwadrateerde sommen (SS), het aantal vrijheidsgraden (df), de variantie (MS), de F-waarden van de testen (F) en de p-waarden (Prob>F). De derde output geeft een aantal relevante statistische waarden. We moeten eerst kijken naar de uitkomst van de derde test om na te gaan of er interactie is tussen de factoren lijm en hout. Voor de derde test is $p = 0.0076 < \alpha$ en dus verwerpen we de nulhypothese en accepteren we met 95% betrouwbaarheid dat er interactie is tussen houtsoort en lijmssoort. Aangezien er interactie is, is het niet zinvol om verder te kijken naar de uitkomst van testen 1 en 2, omdat interactie net inhoudt dat je geen algemene besluiten kunt trekken over bijvoorbeeld de invloed van lijmssoort.



Oefening 80

Opgave:

Men wenst onderzoek te doen rond verschillende samenstellingen van wegdek om een snelweg mee te bouwen. Bij een proefproject worden de vier verschillende types wegdek (die verschillen qua samenstelling van asphalt) gebruikt op de E17, de E40, de R4 en de E413, telkens met een strook van 1 km. Na vijf jaar wordt geëvalueerd hoeveel herstellingen dienden te gebeuren (zie tabel). Maak een totale variantie-analyse voor wat betreft de types wegdek op de verschillende snelwegen. Zie cursus paragraaf 8.3.

	Type 1	Type 2	Type 3	Type 4
E17	4	3	6	10
E40	10	8	5	16
R4	2	0	3	4
E413	1	2	3	4

Oplossing:

We beginnen met het maken van een datamatrix A:

```
A=[ 4 3 6 10; 10 8 5 16; 2 0 3 4; 1 2 3 4 ]
```

```
A = 4x4
    4     3     6     10
    10    8     5     16
    2     0     3     4
    1     2     3     4
```

Deze oefening is een voorbeeld van Block design ANOVA. De waarnemingen verbonden met eenzelfde autosnelweg (bijvoorbeeld: veel of weinig vrachtverkeer) vormen een blok omdat mogelijks de karakteristieken van de autosnelweg ook een invloed kunnen hebben. Hier wordt er geen interactie tussen de twee factoren getest omdat daar niet genoeg metingen voor gedaan werden, en de twee hypothesen zijn:

Test 1:

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 \text{ (geen verschil in evaluatie voor de verschillende soorten wegdek)}$$
$$H_1 : \text{niet alle populatiegemiddelen zijn gelijk}$$

Test 2:

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 \text{ (geen verschil in evaluatie voor de verschillende autosnelwegen)}$$
$$H_1 : \text{niet alle populatiegemiddelen zijn gelijk}$$

De ANOVA-test voeren we uit met anova2. Als argumenten geven we de matrix A en 1, het aantal herhalingen bij de metingen onder dezelfde omstandigheden (zelfde snelweg en zelfde soort wegdek).

```
[p,table,stats]=anova2(A,1)
```

```
p = 1x2
    0.0300    0.0026
```

table = 5x6 cell

	1	2	3	4	5	6
1	'Source'	'SS'	'df'	'MS'	'F'	'Prob>F'
2	'Columns'	65.6875	3	21.8958	4.7414	0.03
3	'Rows'	147.6875	3	49.2292	10.6602	0.0026
4	'Error'	41.5625	9	4.6181	[]	[]
5	'Total'	254.9375	15	[]	[]	[]

stats = struct with fields:

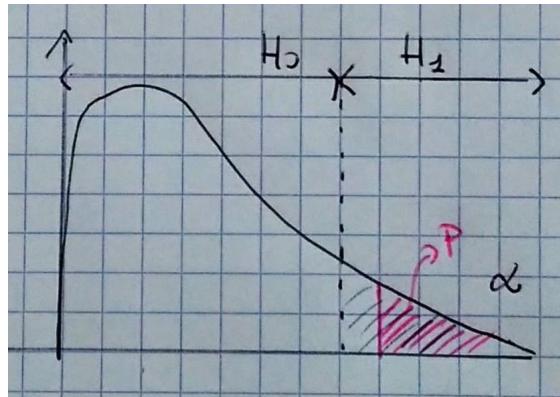
```

    source: 'anova2'
    sigmasq: 4.6181
    colmeans: [4.2500 3.2500 4.2500 8.5000]
    coln: 4
    rowmeans: [5.7500 9.7500 2.2500 2.5000]
   rown: 4
    inter: 0
    pval: NaN
    df: 9
  
```

Uit de p-waarden volgt:

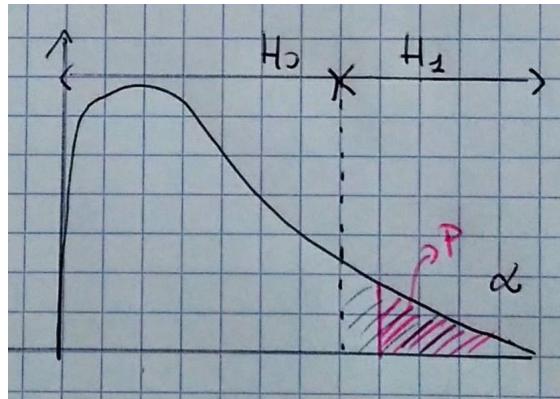
Test 1:

$p = 0.0300 < 0.05 \Rightarrow H_0$ verwerpen en H_1 accepteren met 95% betrouwbaarheid: niet alle populatiegemiddelen zijn gelijk voor de verschillende soorten wegdek.



Test 2:

$p = 0.0026 < 0.05 \Rightarrow H_0$ verwerpen en H_1 accepteren met 95% betrouwbaarheid: niet alle populatiegemiddelen zijn gelijk voor de verschillende snelwegen.



Met `multcompare` kunnen de verschillende subhypothesen getest worden, dit commando kunnen we gebruiken voor de kolommen (types wegdek) en de rijen (verschillende autosnelwegen). `multcompare` geeft zowel een figuur als een tabel terug, uit beide kunnen we eigenlijk dezelfde informatie ontleden.

Op de figuur kunnen we zien of er overlap is tussen de 95%-betrouwbaarheidsintervallen van de populatiegemiddelen μ_i of niet, indien deze intervallen voor twee verschillende groepen i en j overlappen dan zullen we de subhypothese $\mu_i = \mu_j$ aanvaarden, zoniet dan zullen we deze verwerpen.

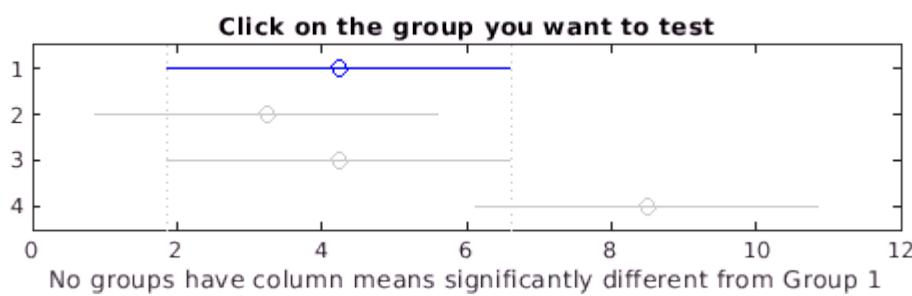
In de tabel staan een aantal belangrijke waardes waarop we ons kunnen baseren om de subhypothesen na te gaan. In de eerste kolommen wordt aangegeven welke groepen vergeleken worden, i en j . In de vierde kolom staat het verschil van de steekproefgemiddelen van de twee groepen $\bar{x}_i - \bar{x}_j$. In de derde en vijfde kolom staan de grenzen van het 95%-betrouwbaarheidsinterval van het verschil van de populatiegemiddelen $\mu_i - \mu_j$, als 0 in dit interval ligt, dan accepteren we de subhypothese $\mu_i = \mu_j$. In de laatste kolom staan ook nog eens de p-waarden voor de verschillende subhypothesen:

$$H_0 : \mu_i = \mu_j$$

$$H_1 : \mu_i \neq \mu_j$$

We vergelijken eerst de kolommen en dus de verschillende types wegdek, 1 komt hier overeen met het wegdek Type 1, 2 met Type 2 enzovoort.

```
multcompare(stats)
```



ans = 6x6

1.0000	2.0000	-3.7437	1.0000	5.7437	0.9101
1.0000	3.0000	-4.7437	0	4.7437	0
1.0000	4.0000	-8.9937	-4.2500	0.4937	0.0815
2.0000	3.0000	-5.7437	-1.0000	3.7437	0.9101
2.0000	4.0000	-9.9937	-5.2500	-0.5063	0.0303
3.0000	4.0000	-8.9937	-4.2500	0.4937	0.0815

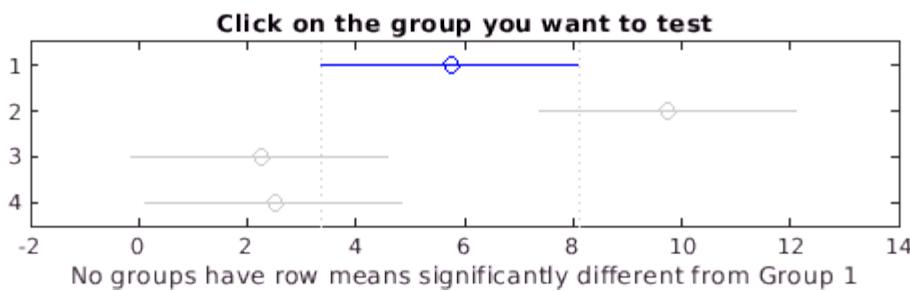
Conclusie met 95% betrouwbaarheid:

- $\mu_1 = \mu_2$
- $\mu_1 = \mu_3$
- $\mu_1 = \mu_4$
- $\mu_2 = \mu_3$
- $\mu_2 \neq \mu_4$
- $\mu_3 = \mu_4$

LET OP: De p-waarde in de tweede rij van de tabel, die waar Type 1 en 3 worden vergeleken, wordt hier weergeven als 0. Dit zou echter 1 moeten zijn aangezien het verschil van de steekproefgemiddelden van de twee groepen gelijk is aan nul. Dit kunnen we extra controleren met het 95%-betrouwbaarheidsinterval [-4.7437, 4.7437] dat nul omvat en het overlappen van de betrouwbaarheidsintervallen in de figuur. Het gaat hier dus om een bug in deze versie van Matlab.

Na de verschillende types wegdek te vergelijken kunnen we de verschillende autosnelwegen vergelijken (deze komen overeen met de verschillende rijen). 1 komt in dit geval overeen met de E17, 2 met de E40 enzovoort.

```
multcompare(stats, 'estimate', 'row')
```



```
ans = 6x6
1.0000    2.0000   -8.7437   -4.0000    0.7437    0.1041
1.0000    3.0000   -1.2437    3.5000    8.2437    0.1683
1.0000    4.0000   -1.4937    3.2500    7.9937    0.2123
2.0000    3.0000    2.7563    7.5000   12.2437    0.0037
2.0000    4.0000    2.5063    7.2500   11.9937    0.0046
3.0000    4.0000   -4.9937   -0.2500    4.4937    0.9983
```

Hieruit kunnen met 95% betrouwbaarheid de volgende conclusies trekken:

- $\mu_1 = \mu_2$
- $\mu_1 = \mu_3$
- $\mu_1 = \mu_4$

- $\mu_2 \neq \mu_3$
- $\mu_2 \neq \mu_4$
- $\mu_3 = \mu_4$

Oefening 81

Opgave:

Op verschillende plaatsen worden watermonsters genomen en het zuurstofgehalte bepaald:

- plaats 1: stroomopwaarts
- plaats 2: nabij een fabriek
- plaats 3: stroomafwaarts

Ga na of er een significant verschil is tussen de plaatsen.

	Plaats 1	6.3	6.6	6.4	6.4	6.5
Plaats 2	4.8	4.3	5.0	4.7	5.1	
Plaats 3	6.0	6.2	6.3	5.8	6.2	

Oplossing:

Voor deze oefening maken we gebruik van One-Way ANOVA. We beginnen met het opslaan van de data in een matrix:

```
A=[6.3 6.6 6.4 6.4 6.5;4.8 4.3 5.0 4.7 5.1;6.0 6.2 6.3 5.8 6.2]
```

```
A = 3x5
6.3000    6.6000    6.4000    6.4000    6.5000
4.8000    4.3000    5.0000    4.7000    5.1000
6.0000    6.2000    6.3000    5.8000    6.2000
```

We beginnen met het testen of de drie steekproeven normaal verdeeld zijn met een KS-test:

H_0 :normaal verdeeld

H_1 :niet normaal verdeeld

```
x=A(1,:);
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.8821
```

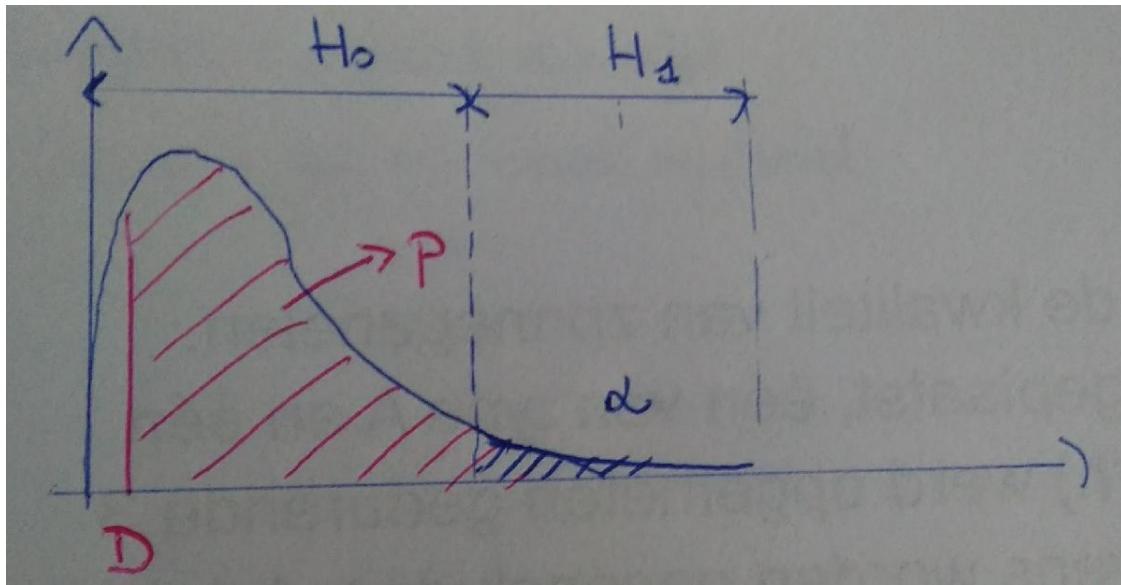
```
x=A(2,:);
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
0
p = 0.9621
```

```
x=A(3,:);
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = logical
```

$$p = 0.6974$$



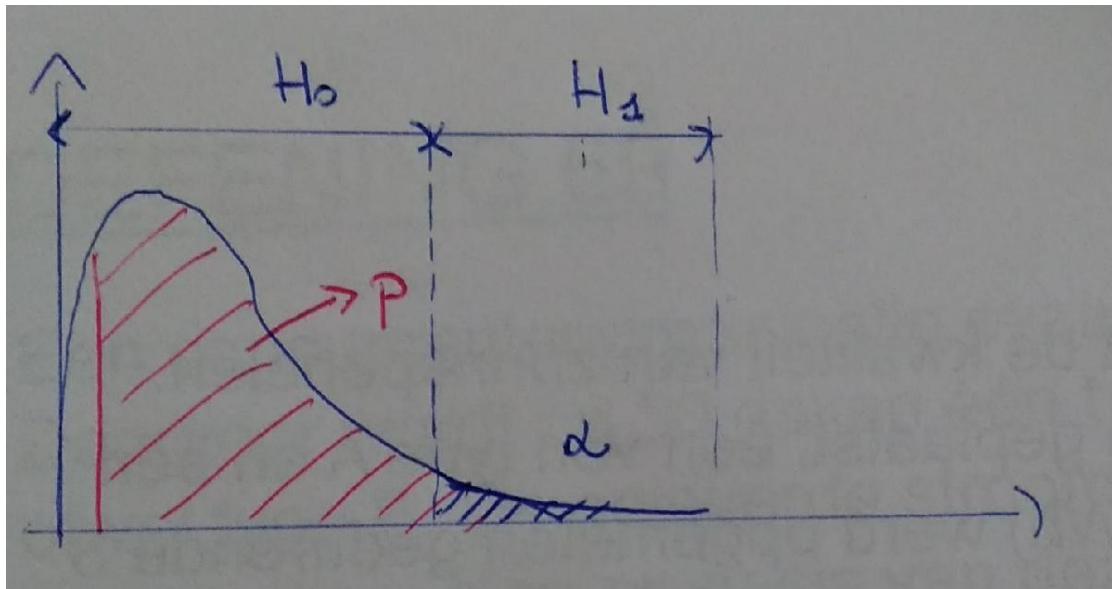
Voor de drie steekproeven geldt $p > 0.05$. We aanvaarden de 3 nulhypothesen met 95% betrouwbaarheid. De voorwaarde van normaliteit voor One-Way ANOVA is dus voldaan. De tweede voorwaarde die we moeten testen is die van gelijke variantie. Dit kunnen we doen met Levene's test. Let op: hier transponeren we A , vartestn vergelijkt namelijk de variantie van de verschillende kolommen.

$$H_0: \sigma_1^2 = \sigma_2^2 = \sigma_3^2 \quad (\text{gelijke varianties})$$

H_1 : de varianties zijn niet alledrie dezelfde

```
p = vartestn(A, 'TestType', 'LeveneAbsolute', 'Display', 'off')
```

$$p = 0.2574$$



$p = 0.2574 > 0.05 \Rightarrow H_0$ accepteren met 95% betrouwbaarheid: de varianties van de steekproeven op verschillende plaatsen zijn gelijk. Nu de twee voorwaarden zijn voldaan kunnen we overgaan tot het uitvoeren van de One-Way ANOVA:

$H_0 : \mu_1 = \mu_2 = \mu_3$ (geen verschil in zuurtegraad op de verschillende plaatsen)

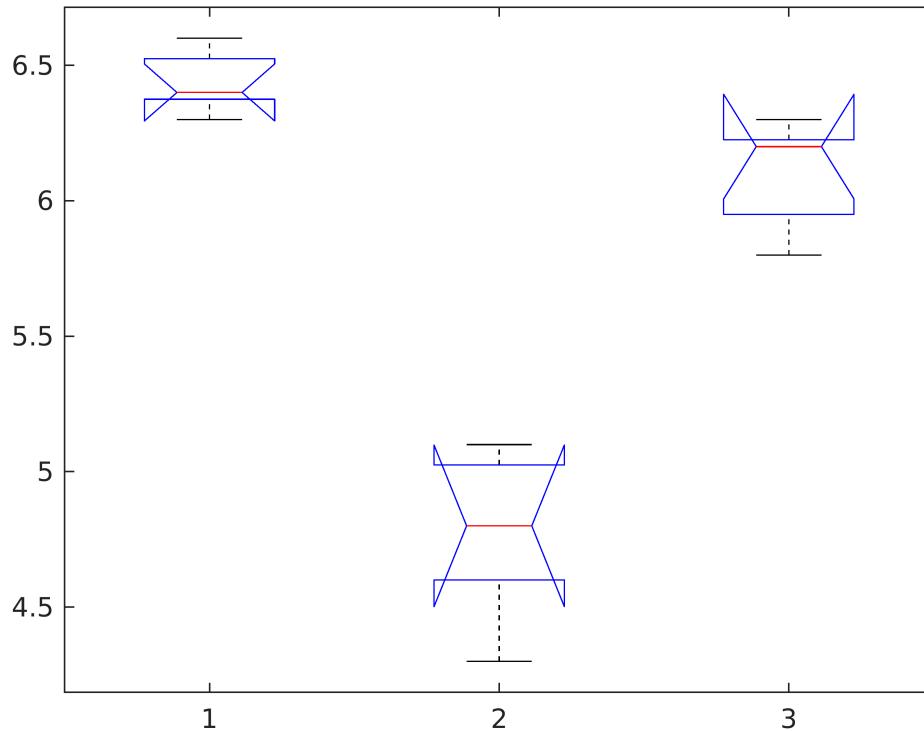
H_1 : niet alle populatiegemiddelen zijn gelijk

Dit doen we aan de hand van `anova1`. Let op: hier transponeren we A, `anova1` vergelijkt namelijk de verschillende kolommen.

```
[p,table,stats]=anova1(A')
```

ANOVA Table

Source	SS	df	MS	F	Prob>F
<hr/>					
Columns	7.68933	2	3.84467	76.89	1.4381e-07
Error	0.6	12	0.05		
Total	8.28933	14			



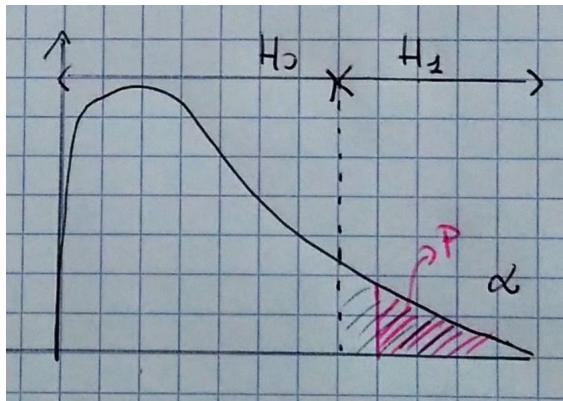
```
p = 1.4381e-07
table = 4x6 cell
```

	1	2	3	4	5	6
1	'Source'	'SS'	'df'	'MS'	'F'	'Prob>F'
2	'Columns'	7.6893	2	3.8447	76.8933	1.4381e-07
3	'Error'	0.6000	12	0.0500	[]	[]

	1	2	3	4	5	6
4	'Total'	8.2893	14	[]	[]	[]

```
stats = struct with fields:
gnames: [3x1 char]
n: [5 5 5]
source: 'anova1'
means: [6.4400 4.7800 6.1000]
df: 12
s: 0.2236
```

Uit de berekening volgt:



$p = 1.4381 \times 10^{-7} < 0.05 \Rightarrow H_0$ verwerpen en H_1 accepteren met 95% betrouwbaarheid: niet alle populatiegemiddelen zijn gelijk. Nu moeten we zoeken wat de verschillen zijn in de populatiegemiddelen. Hiervoor gebruiken we drie verschillende subhypothesen:

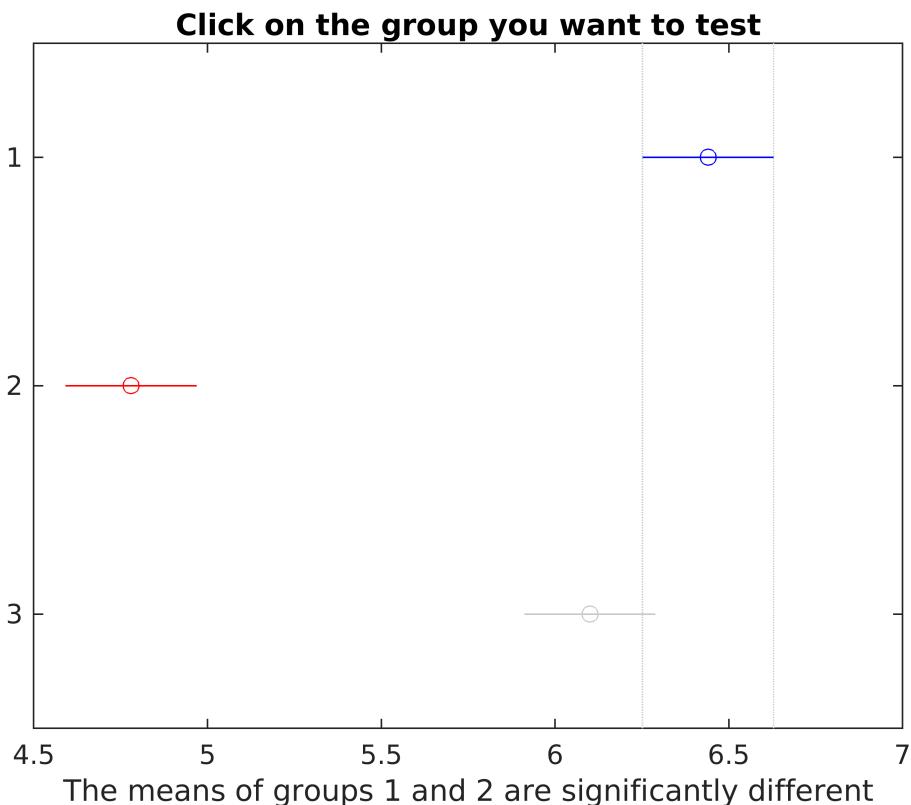
- $H_0 : \mu_1 = \mu_2$
 $H_1 : \mu_1 \neq \mu_2$

- $H_0 : \mu_1 = \mu_3$
 $H_1 : \mu_1 \neq \mu_3$

- $H_0 : \mu_2 = \mu_3$
 $H_1 : \mu_2 \neq \mu_3$

Deze subhypothesen kunnen we testen met `multcompare`:

```
table = multcompare(stats)
```



```
table = 3x6
 1.0000    2.0000    1.2827    1.6600    2.0373    0.0000
 1.0000    3.0000   -0.0373    0.3400    0.7173    0.0790
 2.0000    3.0000   -1.6973   -1.3200   -0.9427    0.0000
```

De waarden in `table` kunnen we als volgt interpreteren: de eerste twee kolommen staan voor de plaatsen die vergeleken worden (1.0000 voor plaats 1, 2.0000 voor plaats 2 etc.). De vierde kolom is het verschil van de populatiegemiddelden van de twee groepen. In de derde en vijfde kolom staan de grenzen van het (95%) betrouwbaarheidsinterval van het verschil van de populatiegemiddelden. In de laatste kolom staan de p-waarden voor de verschillende subhypothesen.

Uit deze tabel volgt (met 95% betrouwbaarheid):

- $\mu_1 \neq \mu_2$
- $\mu_1 = \mu_3$
- $\mu_2 \neq \mu_3$
- $\mu_2 < \mu_1, \mu_3$

Deze resultaten kunnen we ook zien in de grafiek.

Oefening 82

Opgave:

Er wordt een studie gedaan waarin de Fe-concentratie bepaald wordt in grondstalen. Op 3 verschillende plaatsen worden telkens zes stalen genomen en de metingen op elk staal worden uitgevoerd in 2 verschillende labo's, die elk een andere meettechniek gebruiken. In de tabel (zie cursus) worden de resultaten weergegeven. Ga na in hoeverre het labo (factor A) en de plaats (factor B) effect hebben op de meting ($\alpha=0.05$).

Oplossing:

Eerst maken we een matrix met de gegevens. De drie plaatsen worden gegeven door de kolommen. De eerste zes rijen komen overeen met het eerste labo en de tweede zes rijen met het tweede labo.

```
A=[5.59 5.67 5.75;5.59 5.67 5.47;5.37 5.55 5.43;  
5.54 5.57 5.45;5.37 5.43 5.24;5.42 5.57 5.47;  
5.90 5.90 5.81;5.75 6.01 5.90;6.07 5.85 5.81;  
5.90 5.54 5.81;6.01 5.81 5.65;6.06 5.70 5.75]
```

```
A = 12x3  
5.5900 5.6700 5.7500  
5.5900 5.6700 5.4700  
5.3700 5.5500 5.4300  
5.5400 5.5700 5.4500  
5.3700 5.4300 5.2400  
5.4200 5.5700 5.4700  
5.9000 5.9000 5.8100  
5.7500 6.0100 5.9000  
6.0700 5.8500 5.8100  
5.9000 5.5400 5.8100  
:  
:
```

We moeten nagaan of de kwantitatieve variabele Fe-concentratie normaal verdeeld is over de verschillende niveaus van factor 1 (Plaats). Daarnaast moeten we ook nagaan of de Fe-concentratie normaal verdeeld is over de verschillende niveaus van factor 2 (Labo). Voor het testen van normaliteit gebruiken we een KS-test:

H_0 : normaal verdeeld

H_1 : niet normaal verdeeld

Voor de verschillende plaatsen:

```
x=A(:,1);  
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));  
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = Logical  
0  
p = 0.7776
```

```
x=A(:,2);  
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));  
[h,p]=kstest(x,'CDF',test_cdf)
```

```
h = Logical
```

```

0
p = 0.8113

```

```

x=A(:,3);
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)

```

```

h = Logical
0
p = 0.5305

```

Voor de verschillende labo's:

```

x=[A(1,:),A(2,:),A(3,:),A(4,:),A(5,:),
A(6,:)];
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)

```

```

h = Logical
0
p = 0.9335

```

```

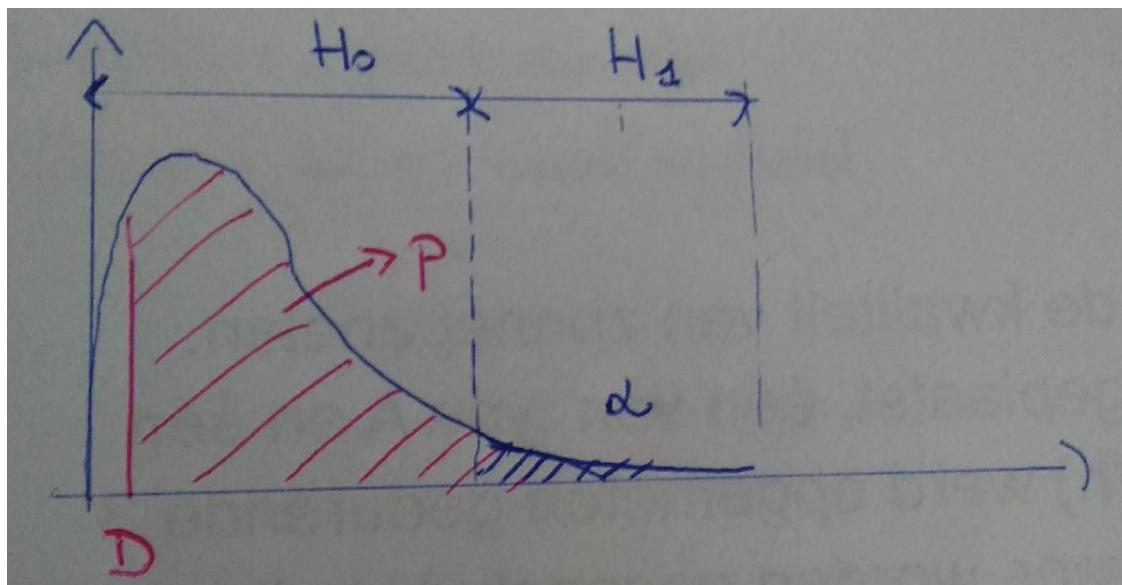
x=[A(7,:),A(8,:),A(9,:),A(10,:),
A(11,:),A(12,:)];
test_cdf=makedist('Normal','mu',mean(x),'sigma',std(x));
[h,p]=kstest(x,'CDF',test_cdf)

```

```

h = Logical
0
p = 0.8887

```



Uit de verschillende KS-testen volgt dat de normaliteit in alle gevallen geaccepteerd wordt met 95% betrouwbaarheid, omdat telkens $p > \alpha$. Naast normaliteit moet we ook de gelijkheid van de variantie testen, dit doen we met Levene's test. Voor drie plaatsen wordt dit:

$$H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 \quad (\text{gelijke varianties})$$

$$H_1 : \text{Niet alle drie varianties zijn gelijk}$$

De verschillende plaatsen komen overeen met de kolommen van A en dus kunnen we test als volgt doen:

```
p = vartestn(A, 'TestType', 'LeveneAbsolute', 'Display', 'off')
```

```
p = 0.0659
```

Voor de twee labo's zijn de hypothesen voor Levene's test als volgt:

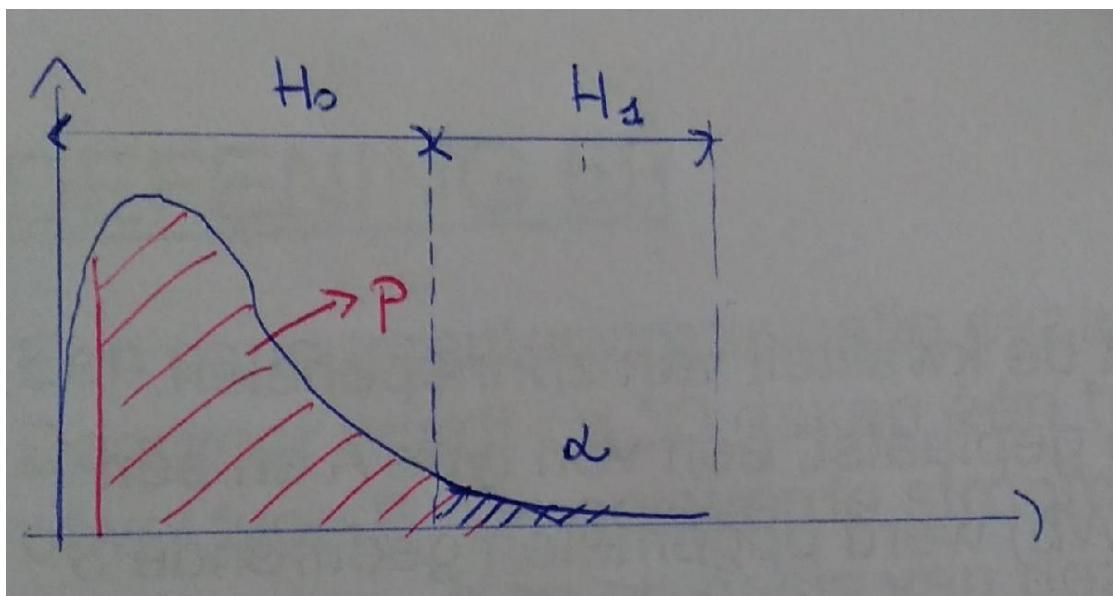
$$H_0: \sigma_1^2 = \sigma_2^2 \quad (\text{gelijke varianties})$$

$$H_1: \sigma_1^2 \neq \sigma_2^2 \quad (\text{geen gelijke variantie})$$

Aangezien vartestn de matrixkolommen vergelijkt moeten we eerst de matrix A omvormen zodat de kolommen overeen komen met de twee verschillende labo's.

```
p = vartestn([[A(1:6,1);A(1:6,2);A(1:6,3)],[A(7:12,1);A(7:12,2);A(7:12,3)]] ...  
, 'TestType', 'LeveneAbsolute', 'Display', 'off')
```

```
p = 0.7977
```



Voor beide testen is $p > \alpha$ en de variantie bij de verschillende plaatsen en labo's wordt met 95% betrouwbaarheid geaccepteerd als gelijk. De twee voorwaarden zijn voldaan en we kunnen verder gaan met Two-way ANOVA. De 3 nulhypothesen zijn:

Test 1:

$$H_0: \mu_1 = \mu_2 = \mu_3 \quad (\text{geen verschil in gemiddelde Fe-concentratie voor de 3 verschillende plaatsen})$$

$$H_1: \text{niet alle populatiegemiddelen zijn gelijk}$$

Test 2:

$$H_0: \mu_1 = \mu_2 \quad (\text{geen verschil in gemiddelde Fe-concentratie voor de 2 verschillende labo's})$$

H_1 : de populatiegemiddelen zijn niet gelijk

Test 3:

H_0 : er is GEEN interactie

H_1 : er is interactie

```
[p,table,stats]=anova2(A,6)
```

```
p = 1x3  
0.2427 0.0000 0.0727
```

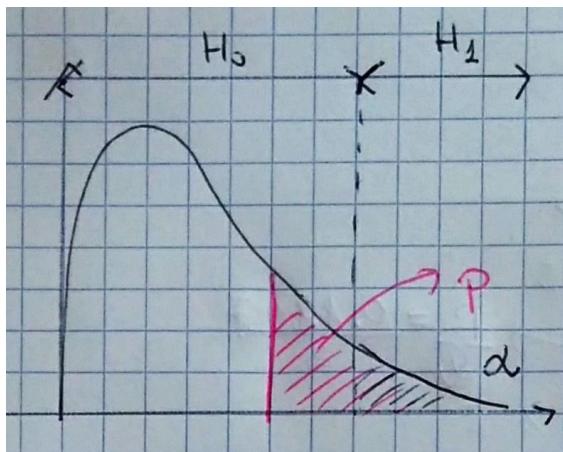
```
table = 6x6 cell
```

	1	2	3	4	5	6
1	'Source'	'SS'	'df'	'MS'	'F'	'Prob>F'
2	'Columns'	0.0468	2	0.0234	1.4849	0.2427
3	'Rows'	1.0268	1	1.0268	65.2011	5.1653e-09
4	'Interaction'	0.0902	2	0.0451	2.8649	0.0727
5	'Error'	0.4725	30	0.0157	[]	[]
6	'Total'	1.6363	35	[]	[]	[]

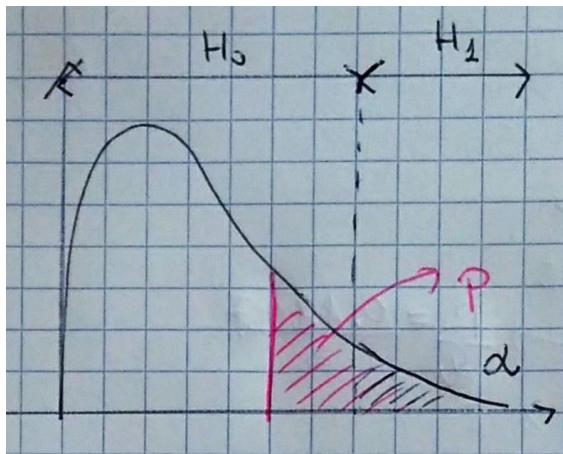
```
stats = struct with fields:
```

```
source: 'anova2'  
sigmasq: 0.0157  
colmeans: [5.7142 5.6892 5.6283]  
coln: 12  
rowmeans: [5.5083 5.8461]  
rown: 18  
inter: 1  
pval: 0.0727  
df: 30
```

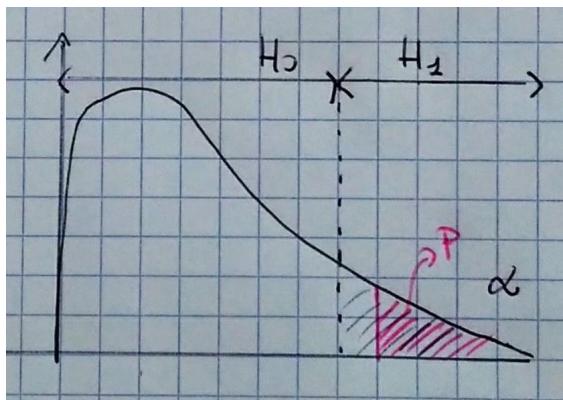
Eerst en vooral kijken we naar de uitkomst van de derde test. Hier is $p = 0.0727 > 0.05$ en dus kunnen we de nulhypothese dat er geen interactie is met 95% betrouwbaarheid accepteren (merk wel op dat dit een grensgeval is).



Nu moeten we verder kijken naar testen 1 en 2 om te kijken of één van de twee factoren een rol speelt in de metingen. Uit test 1 volgt met 95% betrouwbaarheid dat de verschillende plaatsen geen rol spelen in de metingen ($p = 0.2427 > 0.05$)



Uit test 2 volgt met 95% betrouwbaarheid dat het labo wel een verschil maakt ($p = 5.1653 \times 10^{-9} < 0.05$).



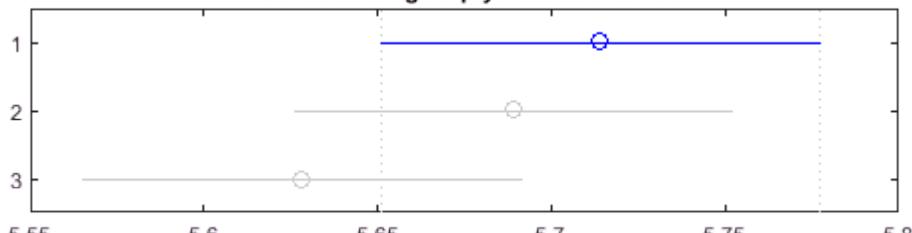
Met multcompare kunnen deze resultaten grafisch voorgesteld worden.

In de grafiek voor de drie plaatsen kunnen we zien dat de 95%-betrouwbaarheidsintervallen overlappen, er is dus geen significant verschil.

```
multcompare(stats)
```

Note: Your model includes an interaction term. A test of main effects can be difficult to interpret when the model includes interactions.

Click on the group you want to test



No groups have column means significantly different from Group 1

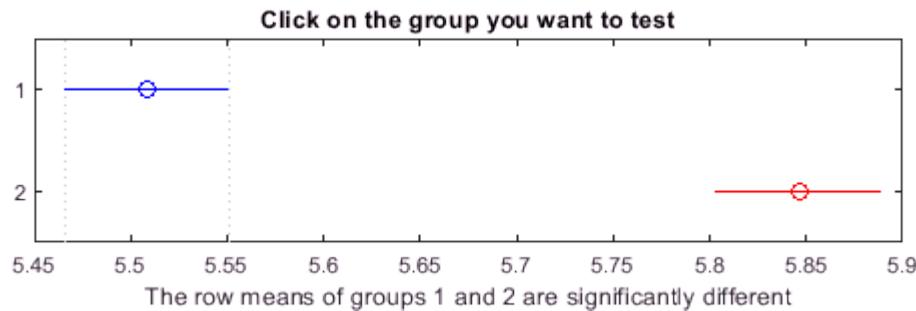
```
ans = 3x6
1.0000    2.0000   -0.1013    0.0250    0.1513    0.8775
1.0000    3.0000   -0.0405    0.0858    0.2121    0.2311
```

```
2.0000    3.0000   -0.0655    0.0608    0.1871    0.4697
```

In de grafiek voor de twee labo's zien we dat er geen overlap is tussen de 95%-betrouwbaarheidsintervallen en dus is er wel een significant verschil.

```
multcompare(stats, 'estimate', 'row')
```

Note: Your model includes an interaction term. A test of main effects can be difficult to interpret when the model includes interactions.



```
ans = 1x6
1.0000    2.0000   -0.4232   -0.3378   -0.2523    0.0000
```

Oefening 8

Opgave:

Bereken de oppervlakte van de driehoek met hoekpunten $a(1, 2, 0)$, $b(3, 0, -3)$ en $c(5, 2, 6)$.

Oplossing:

We beginnen met het definiëren van de drie vectoren \vec{a} , \vec{b} en \vec{c} (met andere woorden, vectoren met als beginpunt de oorsprong en als eindpunt de hoekpunten van de driehoek).

```
a = [1 2 0]
```

```
a = 1x3  
1 2 0
```

```
b = [3 0 -3]
```

```
b = 1x3  
3 0 -3
```

```
c = [5 2 6]
```

```
c = 1x3  
5 2 6
```

De oppervlakte van een driehoek die opgespannen wordt door twee vectoren met hetzelfde beginpunt \vec{v} en \vec{w} kan berekend worden als $\frac{1}{2} ||\vec{v} \times \vec{w}||$ (zie cursus *Wiskunde 1*).

Hier kunnen we de vectoren \vec{v} en \vec{w} definiëren als volgt: $\vec{v} = \vec{b} - \vec{a}$ en $\vec{w} = \vec{c} - \vec{a}$.

```
v = b - a
```

```
v = 1x3  
2 -2 -3
```

```
w = c - a
```

```
w = 1x3  
4 0 6
```

Het vectoriëel product kan berekend worden met het commando *cross*.

```
product = cross(v, w)
```

```
product = 1x3  
-12 -24 8
```

Nu moeten we enkel nog de norm van dit product berekenen met het commando *norm* en delen door twee.

```
norm(product)/2
```

```
ans = 14
```

Oefening 88

Opgave:

Gasmaatschappijen moeten voldoende gastoeroer voorzien om hun klanten het ganse jaar door te bevoorradden. Bij de planning wil men in staat zijn de benodigde hoeveelheid te voorspellen op basis van gegevens rond temperatuur van de dag zelf, van de voorbije dag, de windsnelheid, het feit of het weekend is of niet. De data vind je in gasbedrijf.dat.

1. Stel op basis van de data die vorhanden is, een lineair regressiemodel op dat ons in staat moet stellen om de nodige gashoeveelheid te modelleren.
2. Hoeveel procent van de variantie kan verklaard worden door het regressiemodel? Interpreteer.
3. Wat is volgens het model de hoeveelheid gas die de maatschappij moet voorzien als de temperatuur van de dag (woensdag) gelijk is aan 18 graden Celcius, de temperatuur van de dag ervoor 15 graden Celcius, de windsnelheid 10 bedraagt?
4. Wat is volgens het model het 95% betrouwbaarheidsinterval voor de coëfficiënt bij de temperatuur van de dag?

Oplossing:

1. We moeten eerst de data importeren via Import Data, daarna kunnen we de data van de verklarende variabelen (temperatuur, temperatuur de dag ervoor, windsnelheid en of het al dan niet weekend is) in de matrix x opslaan.

```
x(:,1)=gasbedrijf.temperatuur;
x(:,2)=gasbedrijf.temp_vorige_dag;
x(:,3)=gasbedrijf.windsnelheid;
x(:,4)=gasbedrijf.weekend;
```

We voegen één extra kolom met 1'en toe aan de linkerkant van de matrix x en noemen dit x_ext. Dit doen we voor het toevoegen van een constante term (β_0) aan het regressiemodel.

```
x_ext = [ones(size(x,1),1) x(:,1:4)]
```

```
x_ext = 10x5
1   23   25   12   1
1   24   23   8    1
1   25   24   8    0
1   21   25   8    0
1   27   21   12   1
1   9    14   8    0
1   22   9    8    0
1   17   22   18   0
1   3    17   22   0
1   -2   3    18   1
```

De data van de te verklaren variabele (gastoeroer) slaan we op in y.

```
y=gasbedrijf.gastoeroer
```

```
y = 10x1
227
236
```

228
252
238
333
266
280
386
415

Het lineaire regressiemodel dat we gebruiken is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

met

x_1 = temperatuur

x_2 = temp. vorige dag

x_3 = windsnelheid

x_4 = weekend

De lineaire regressie voeren we uit met `regress(y, x_ext)` (zie handleiding).

```
[b,bint,r,rint,stats] =regress(y,x_ext)
```

```
b = 5x1
392.3905
-5.4551
-1.5445
1.2673
-3.2382
bint = 5x2
348.3992 436.3819
-7.1877 -3.7225
-3.3247 0.2357
-1.1764 3.7110
-21.0295 14.5531
r = 10x1
-13.2792
3.1561
-1.0825
2.6416
13.3633
1.1905
-2.6155
-8.4853
8.3513
-3.2402
rint = 10x2
-31.3461 4.7878
-21.8366 28.1487
-28.5016 26.3367
-22.9464 28.2295
-2.0162 28.7428
-19.8839 22.2648
-15.2954 10.0644
-29.0018 12.0312
-8.4031 25.1057
-17.8032 11.3228
stats = 1x4
0.9873 97.3651 0.0001 106.7050
```

b bevat de schatters voor de regressiecoëfficiënten, b_int bevat de 95%-betrouwbaarheidsintervallen voor de schatters uit b , r bevat de residues van het regressiemodel en r_int bevat de 95%-betrouwbaarheidsintervallen van de residues. $stats$ bevat een aantal statistische waarden, namelijk de R^2 -waarde, de F-statistiek, de p-waarde en een schatting van de variantiefout (in die volgorde).

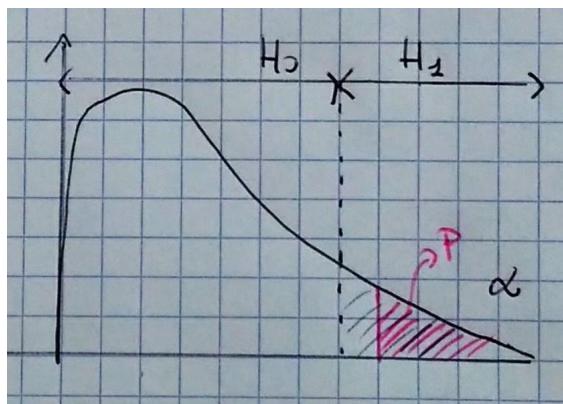
Het regressiemodel dat we vinden is:

$$y = 392.3905 - 5.4551 \cdot x_1 - 1.5445 \cdot x_2 + 1.2673 \cdot x_3 - 3.2382 \cdot x_4$$

Het regressiemodel moet voldoen aan een aantal voorwaarden. De meest belangrijke voorwaarde is dat het regressiemodel significant is. Dit wordt automatisch door Matlab getest met een F-test. Deze heeft als hypothesen:

H_0 : Het regressiemodel is niet significant

H_1 : Het regressiemodel is significant



De F-statistiek van deze test is het tweede element van $stats$ en de overeenkomstige p-waarde is het derde element van $stats$. Hieruit volgt:

$p = 0.0001 < 0.05 \Rightarrow H_0$ verwerpen en H_1 aanvaarden: het regressiemodel is significant met 95% betrouwbaarheid.

De volgende voorwaarde is dat de coëfficiënten $\beta_i (i \in \{0, 1, 2, 3, 4\})$ allemaal significant verschillend zijn van nul. Dit doen we door te kijken naar de 95%-betrouwbaarheidsintervallen van de coëfficiënten. Deze vind je in de matrix $bint$, de rijen komen overeen met de verschillende coëfficiënten.

$$\beta_0 : 0 \notin [348.3992, 436.3819]$$

$$\beta_1 : 0 \notin [-7.1877, -3.7225]$$

$$\beta_2 : 0 \in [-3.3247, 0.2357]$$

$$\beta_3 : 0 \in [-1.1764, 3.7110]$$

$$\beta_4 : 0 \in [-21.0295, 14.5531]$$

Er kan gezien worden dat enkel voor β_0 en β_1 de waarde 0 niet in het betrouwbaarheidsinterval ligt. Enkel β_0 en β_1 zijn dus significant verschillend van nul met 95% betrouwbaarheid.

Aangezien de andere coëfficiënten niet significant verschillen van nul zou er een beter (enkelvoudig) regressiemodel kunnen opgesteld worden waar enkel de temperatuur en een constante term in voorkomen:

$$y = \beta_0 + \beta_1 x_1$$

We passen opnieuw de methode van daarnet toe:

```
x_ext = [ones(size(x,1),1) gasbedrijf.temperatuur]
```

```
x_ext = 10x2
 1    23
 1    24
 1    25
 1    21
 1    27
 1    9
 1    22
 1    17
 1    3
 1   -2
```

```
[b,bint,r,rint,stats] =regress(y,x_ext)
```

```
b = 2x1
 399.4760
 -6.7086
bint = 2x2
 382.1260  416.8260
 -7.6026  -5.8147
r = 10x1
 -18.1773
 -2.4687
 -3.7600
 -6.5946
 19.6573
 -6.0983
 14.1141
 -5.4291
 6.6499
 2.1067
rint = 10x2
 -40.2587  3.9041
 -29.5606  24.6233
 -30.4612  22.9411
 -33.7307  20.5415
  0.1474  39.1672
 -32.5106  20.3141
 -10.4450  38.6732
 -33.0642  22.2059
 -17.1490  30.4488
 -18.8857  23.0991
stats = 1x4
 0.9740  299.4929  0.0000  136.8841
```

Het regressiemodel dat we vinden is:

$$y = 399.4760 - 6.7086 \cdot x_1$$

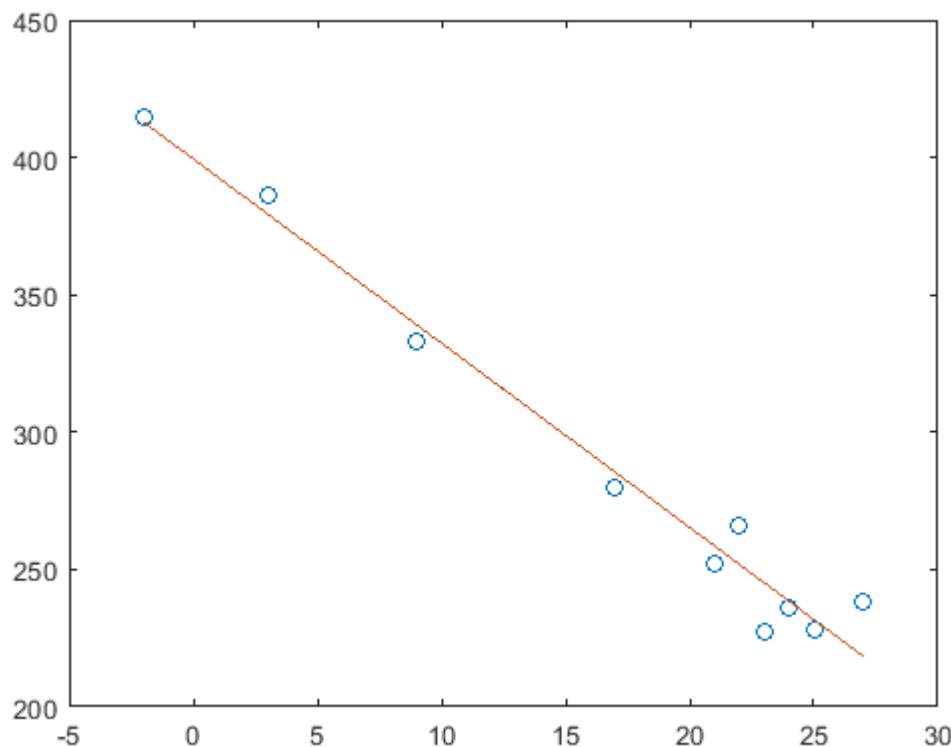
Nu we maar één verklarende variabele hebben is het gemakkelijk om dit regressiemodel te plotten:

```
f = b(1)+b(2)*x(:,1);
```

```

plot(x(:,1),y, 'o') %we gebruiken bolletjes om de datapunten te plotten
hold on
plot(x(:,1),f, '-.') %we gebruiken een lijn voor het regressiemodel
hold off

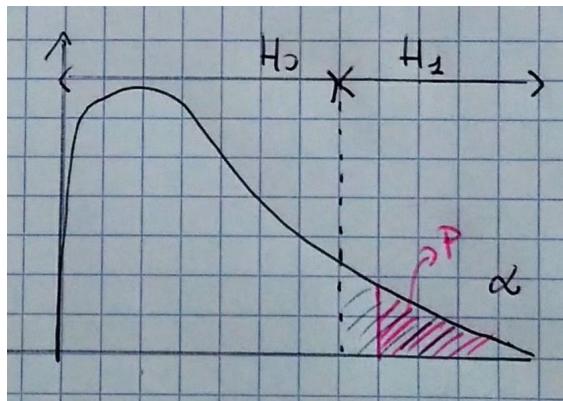
```



Dit nieuw regressiemodel moet opnieuw getest worden op de verschillende voorwaarden. We beginnen weer met de significantie van het model aan de hand van een F-test:

H_0 : Het regressiemodel is niet significant

H_1 : Het regressiemodel is significant



$p = 0.0000 < 0.05 \Rightarrow H_0$ verwerpen en H_1 aanvaarden met 95% betrouwbaarheid: het regressiemodel is significant.

De tweede voorwaarde is ook voldaan aangezien de twee coëfficiënten significant verschillend van nul zijn, dit is natuurlijk geen verassing.

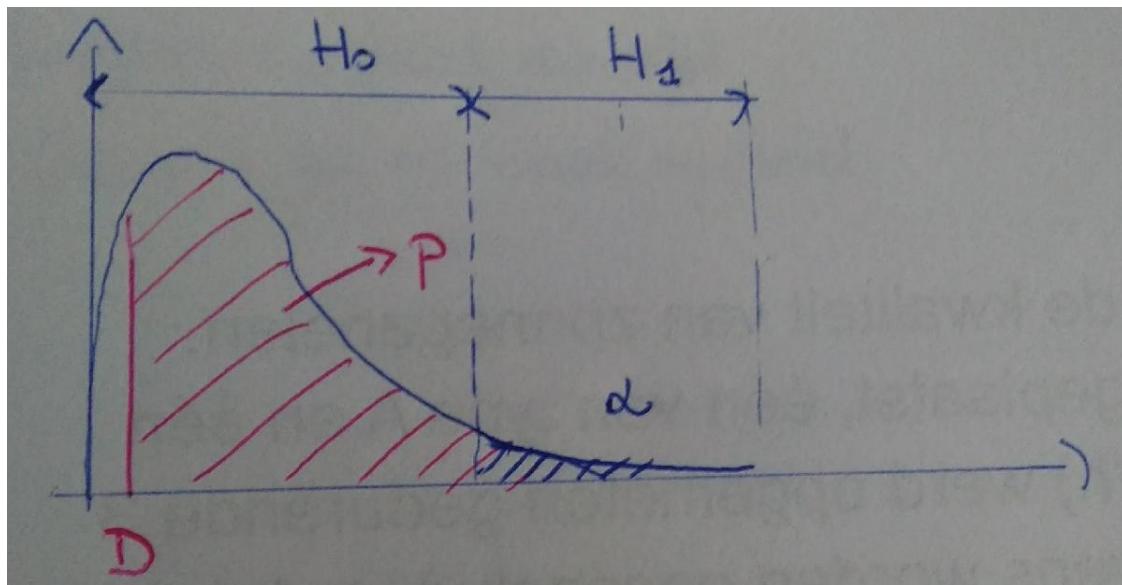
De volgende twee voorwaarden gaan over de residues van de regressie, die kan je vinden in de vector r. De residues moeten normaal verdeeld zijn en mogen niet trendgevoelig zijn. Trendgevoeligheid betekent dat er geen patroon te zien is. Vooral dat de positieve en negatieve residues niet bij elkaar zitten en mooi verspreid zijn. Het testen van normaliteit kunnen we doen met de KS-test:

H_0 : normaal verdeeld

H_1 : niet normaal verdeeld

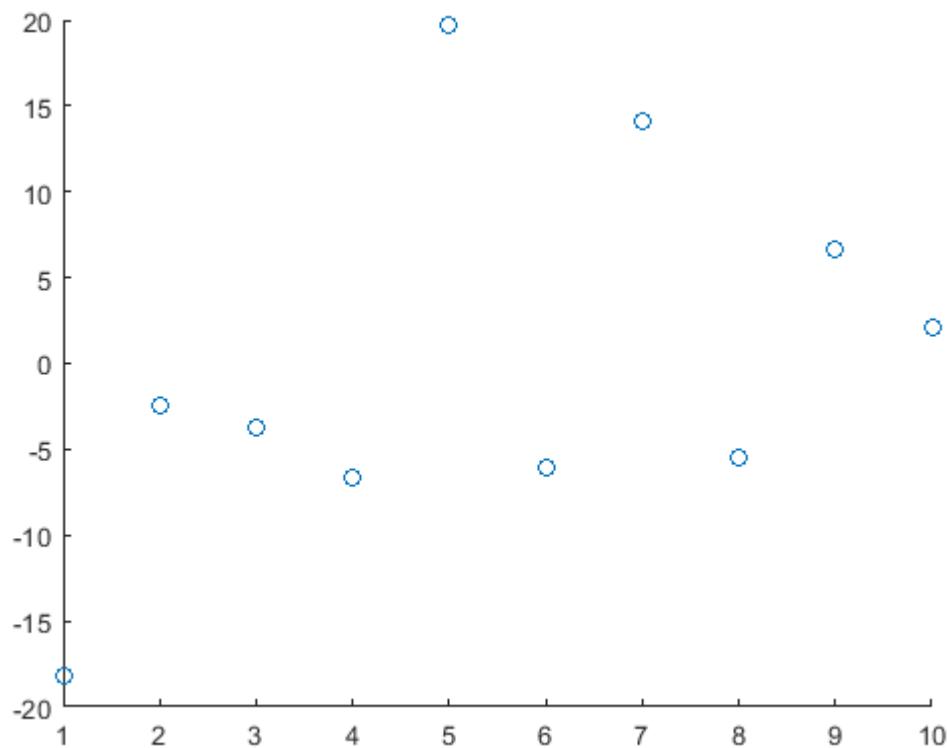
```
test_cdf=makedist('Normal','mu',mean(r),'sigma',std(r));
[h,p]=kstest(r,'CDF',test_cdf)
```

```
h = Logical  
0  
p = 0.8067
```



$p = 0.8067 > 0.05 \Rightarrow H_0$ accepteren met 95% betrouwbaarheid: residues zijn normaal verdeeld. Voor de trendgevoeligheid kunnen we een plot maken van de residues:

```
scatter(1:10, r)
```



Op de plot kan gezien worden dat er geen trend in de residues zit.

Alle vier voorwaarden zijn voldaan en we kunnen dus dit regressiemodel accepteren.

2. De R^2 -waarde van het regressiemodel is 0.9740 (dit is de eerste waarde van stats). Dit betekent dat 97.4% van de variantie in het gastoevoer te verklaren is door de lineaire regressie op de temperatuur.

3. Het gevonden regressiemodel is:

$$y = 399.4760 - 6.7086 \cdot x_1$$

Hier moeten we gewoon x_1 vervangen door de temperatuur van de dag, namelijk 18. We vinden:

$$y = 399.4760 - 6.7086 * 18$$

$$y = 278.7212$$

4. Het 95%-betrouwbaarheidsinterval voor β_1 kunnen we vinden in b_int en is gelijk aan $[-7.6026, -5.8147]$.

Oefening 89

Opgave:

Twee wijnproevers werden gevraagd een rangschikking te maken van 9 gegeven wijnen. De data vind je in `wijndegustatie.dat`.

1. Bereken de correlatiecoëfficiënt van de scores van beide wijnproevers en trek er je conclusies uit over de mening van de twee kenners.
2. Maak een gepaste grafiek die het verband tussen de 2 beoordelingen weergeeft.

Oplossing:

Eerst moet de data geïmporteerd worden met Import Data. We slaan de data op in de vectoren `x` en `y`.

```
x=wijndegustatie.wijnkenner_1;
y=wijndegustatie.wijnkenner_2;
```

1. De correlatiematrix en Pearson test kunnen we berekenen met `corrcoef(x,y)`.

```
[rho,p]=corrcoef(x,y)
```

```
rho = 2x2
 1.0000    0.1333
 0.1333    1.0000
p = 2x2
 1.0000    0.7324
 0.7324    1.0000
```

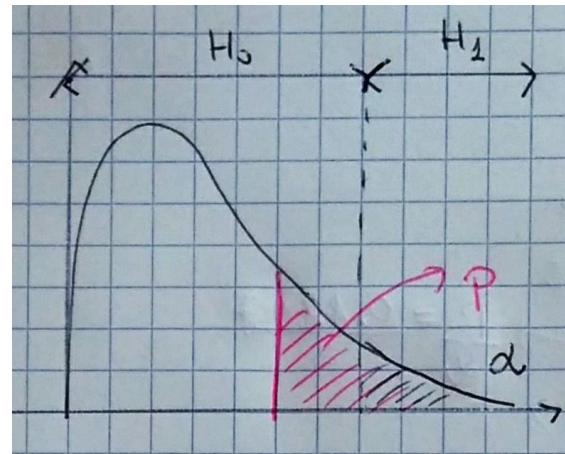
`rho` is de correlatiematrix, de correlatiecoëfficiënt ρ tussen kenner 1 en 2 is de niet-diagonaalwaarde van de matrix, zijnde 0.1333. Er moet echter wel eerst gecontroleerd worden of de gevonden correlatiecoëfficiënt significant verschillend is van nul. Dit doen we aan de hand van de Pearson test. De hypothesen van de Pearson test zijn:

$$H_0 : \rho = 0 \text{ (Er is geen correlatie)}$$

$$H_1 : \rho \neq 0 \text{ (Er is wel correlatie)}$$

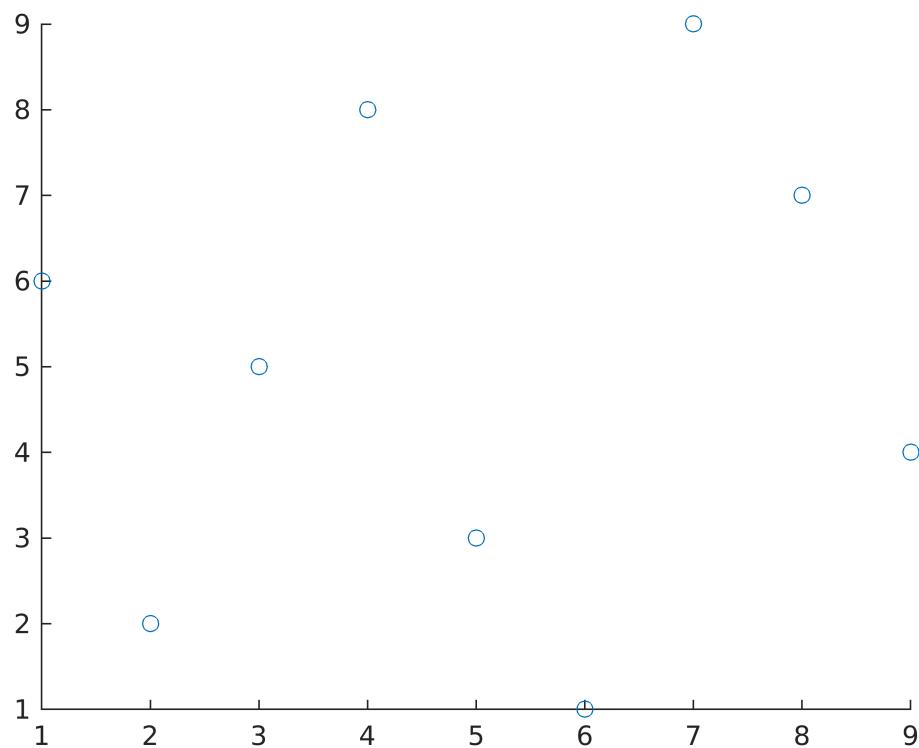
De relevante p-waarde voor deze test is de niet-diagonaalwaarde van de `p` matrix:

$p = 0.7324 > 0.05 \Rightarrow H_0$ aanvaarden. We kunnen dus met 95 % betrouwbaarheid besluiten dat de correlatie NIET significant verschillend is van 0.



2. Door het maken van een scatterplot kan ingezien worden dat er inderdaad geen correlatie is aangezien de punten overal verspreid zijn.

```
scatter(x,y)
```



Oefening 90

Opgave:

Het gewicht van personen wordt vergeleken met de systolische bloeddruk. Dit geeft:

Gewicht	81	92	83	75	81	77	92	101	87
Bloeddruk	122	144	138	145	152	110	118	160	108

Bepaal het lineair verband dat algemeen de bloeddruk uitdrukt in termen van het gewicht. Bespreek voor $\alpha = 0.05$.

Oplossing:

We definiëren kolomvector x met de waarden van de verklarende variabele (gewicht) en kolomvector y met de waarden van de te verklaren variabele (bloeddruk).

```
x = [81 92 83 75 81 77 92 101 87]'
```

```
x = 9x1
81
92
83
75
81
77
92
101
87
```

```
y = [122 144 138 145 152 110 118 160 108]'
```

```
y = 9x1
122
144
138
145
152
110
118
160
108
```

We voegen een extra kolom 1'en toe aan x om een constante term in het regressiemodel te brengen.

```
x_ext = [ones(9,1) x]
```

```
x_ext = 9x2
1 81
1 92
1 83
1 75
1 81
1 77
1 92
1 101
1 87
```

```
[b,bint,r,rint,stats]=regress(y, x_ext)
```

```

b = 2x1
75.2405
0.6760
bint = 2x2
-91.3830 241.8641
-1.2659 2.6179
r = 9x1
-7.9956
6.5685
6.6524
19.0603
22.0044
-17.2917
-19.4315
16.4846
-26.0515
rint = 9x2
-54.6036 38.6124
-39.0785 52.2156
-40.8539 54.1587
-19.0544 57.1751
-19.8636 63.8724
-58.4603 23.8770
-61.2635 22.4006
-13.8150 46.7843
-66.5547 14.4516
stats = 1x4
0.0883 0.6776 0.4376 375.1183

```

Het model dat we krijgen is:

$$\text{bloeddruk} = 75.2405 + 0.6760 \cdot \text{gewicht}$$

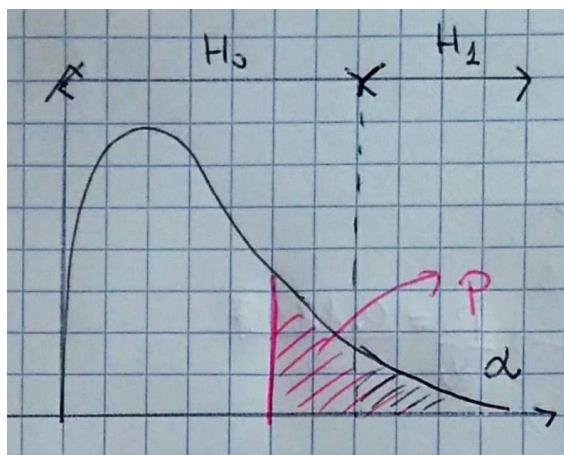
Eerst controleren we of het model significant is met een F-test:

H_0 : Het regressiemodel is niet significant

H_1 : Het regressiemodel is significant

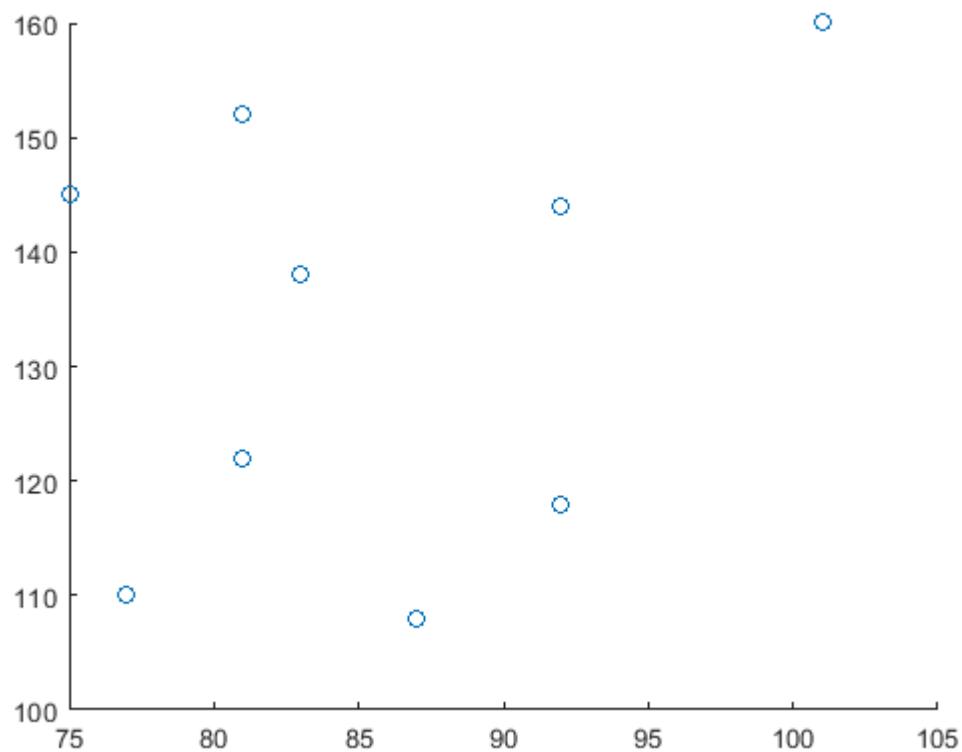
Hieruit volgt:

$p = 0.4376 > 0.05 \Rightarrow H_0$ aanvaarden: het regressiemodel is niet significant met 95% betrouwbaarheid.



Als we een scatterplot maken van de data kan er direct ingezien worden waarom het regressiemodel faalt, de datapunten zijn namelijk ver verspreid en er is geen trend te zien.

```
scatter(x, y)
```



Oefening 91

Opgave:

Gegeven zijn volgende meetresultaten:

x :	1.5	2	3.1	4.2	5	6.1	6.9	8.3	9.5	10.1
y :	7	20	65	64	187	430	600	1300	2000	3500

Bepaal op basis van deze gegevens het best passende lineair verband $y = \beta_0 + \beta_1 x$. Kan je het model verbeteren?

Oplossing:

```
x = [1.5 2 3.1 4.2 5 6.1 6.9 8.3 9.5 10.1]';
y = [7 20 65 64 187 430 600 1300 2000 3500]';
x_ext = [ones(10,1) x]
```

```
x_ext = 10×2
1.0000    1.5000
1.0000    2.0000
1.0000    3.1000
1.0000    4.2000
1.0000    5.0000
1.0000    6.1000
1.0000    6.9000
1.0000    8.3000
1.0000    9.5000
1.0000    10.1000
```

```
[b,bint,r,rint,stats]=regress(y, x_ext)
```

```
b = 2×1
103 ×
-1.0288
0.3256
bint = 2×2
103 ×
-2.0232 -0.0344
0.1692 0.4819
r = 10×1
103 ×
0.5474
0.3976
0.0845
-0.2747
-0.4122
-0.5273
-0.6178
-0.3736
-0.0643
1.2403
rint = 10×2
103 ×
-0.6542 1.7490
-0.8955 1.6907
-1.3278 1.4967
-1.7136 1.1642
-1.8423 1.0180
```

```

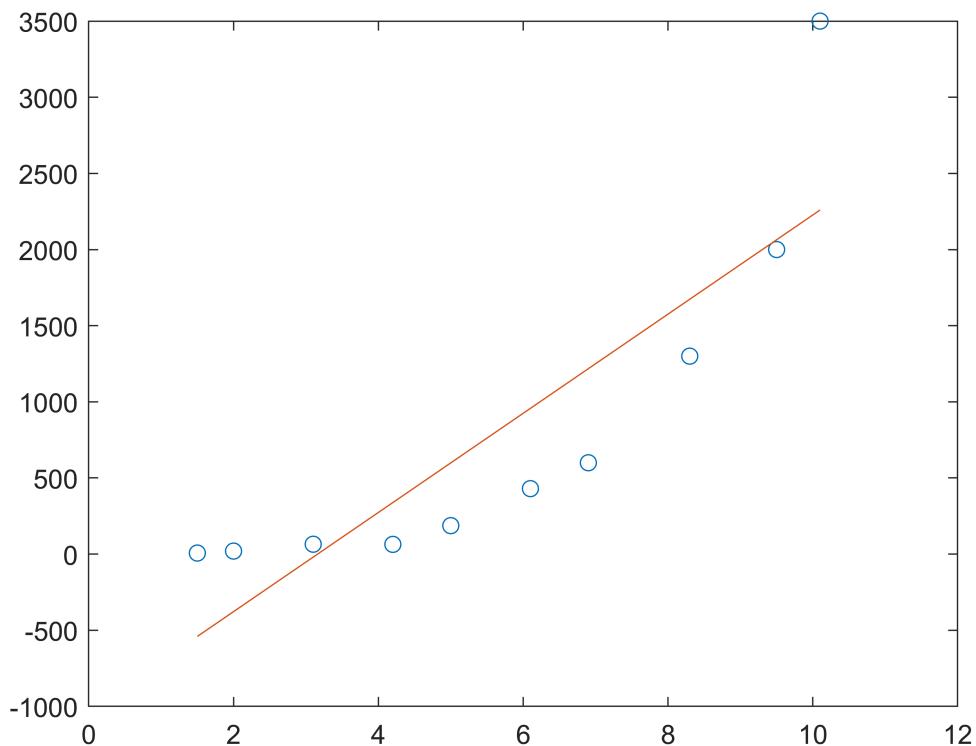
-1.9297  0.8750
-1.9760  0.7405
-1.7445  0.9973
-1.3909  1.2623
  0.6169  1.8638
stats = 1x4
105 ×
  0.0000    0.0002    0.0000    3.8151

```

```

f = b(1)+b(2)*x(:,1);
plot(x(:,1),y, 'o')
hold on
plot(x(:,1),f, '-')
hold off

```



Aangezien stats in wetenschappelijke notatie is geschreven met maar vier beduidende cijfers moeten we de waarde van p nog eens explicet opvragen om de exacte waarde te krijgen.

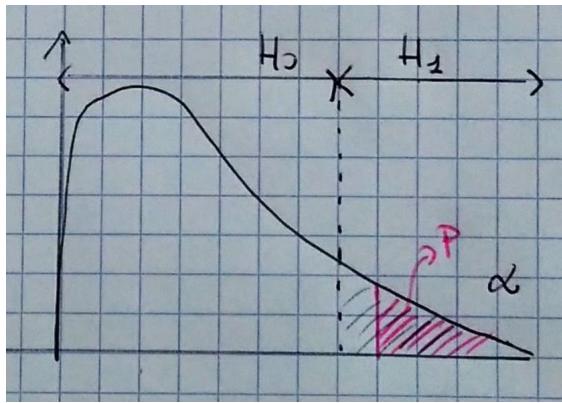
```
p = stats(1,3)
```

```
p = 0.0014
```

Uit de F-test met onderstaande hypothesen volgt:

H_0 : Het regressiemodel is niet significant

H_1 : Het regressiemodel is significant



$p = 0.0014 < 0.05 \Rightarrow H_0$ verwerpen en H_1 aanvaarden: het regressiemodel is significant met 95% betrouwbaarheid.

De tweede voorwaarde die we moeten nagaan is of alle coëfficiënten significant verschillend zijn van nul. Dit kunnen we nagaan door te kijken naar de 95%-betrouwbaarheidsintervallen in bint. Hieruit volgt dat β_1 en β_0 significant verschillend zijn van nul met 95% betrouwbaarheid.

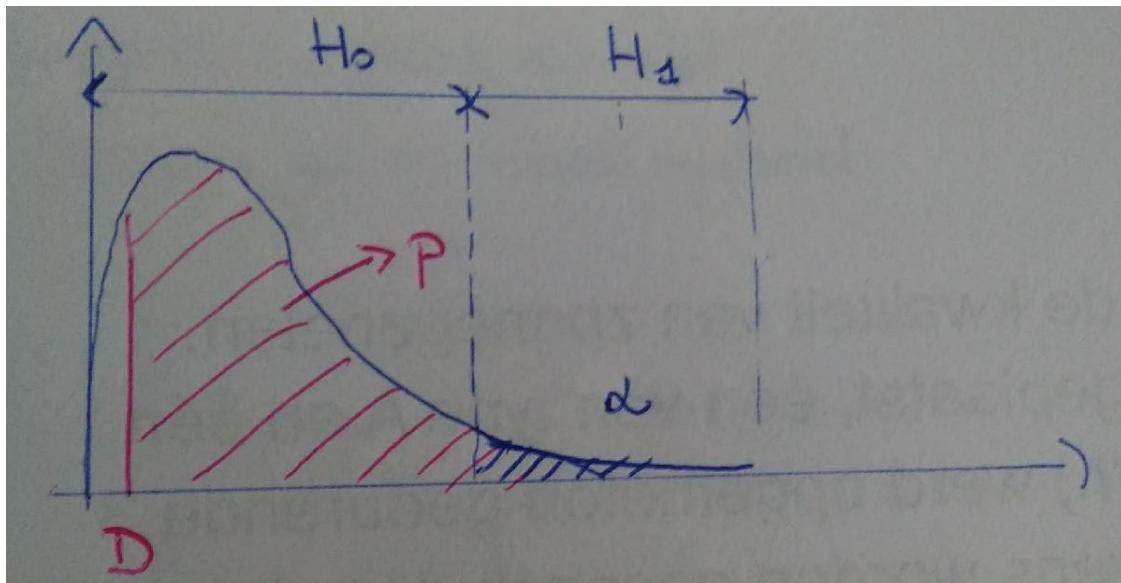
De volgende voorwaarden zijn die voor de residues, die moeten normaal verdeeld zijn en mogen niet trendgevoelig zijn. De normaliteit testen we met de KS-test:

H_0 : normaal verdeeld

H_1 : niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(r),'sigma',std(r));
[h,p]=kstest(r,'CDF',test_cdf)
```

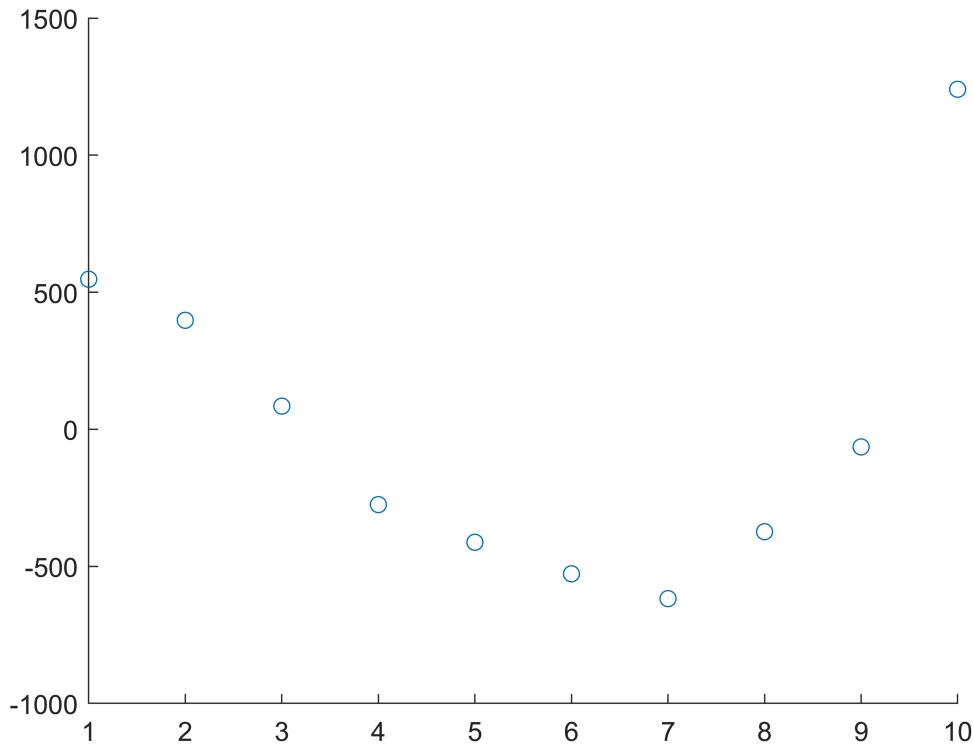
```
h = Logical
0
p = 0.8407
```



$p = 0.8407 > 0.05 \Rightarrow H_0$ accepteren met 95% betrouwbaarheid: de residuen zijn normaal verdeeld. Deze voorwaarde is dus voldaan.

De voorwaarde van de trendgevoeligheid van de residuen faalt echter want er is een duidelijke trend te zien.

```
scatter(1:10, r)
```



De eerste drie residuen zijn positief, de volgende zes negatief en de laatste weer positief. Dit verwijst ernaar dat het gebruikte regressiemodel beperkt is en dat we eventueel over moeten gaan naar een ander model.

Als een verbetering op het lineair regressiemodel proberen we eerst regressie met een tweedegraadsveelterm:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2$$

Het fitten van niet-lineaire curves kan je doen met `fitnlm`. Om deze methode toe te passen moeten we eerst een modelfunctie definiëren die we willen fitten, deze noemen we `modelfun`. Daarnaast moeten we een schatting geven van de coëfficiënten (`initiele_beta`), deze schatting wordt als beginwaarde gebruikt door het fitalgoritme. Het kiezen van deze waarden speelt in dit geval geen grote rol, de methode zal steeds naar dezelfde waarden convergeren. Bij het fitten van niet-lineaire modellen zijn de belangrijkste eigenschappen de significantie van het model en de R^2 -waarde.

```
modelfun = @(b,x)(b(1) + b(2)*x+b(3)*x.^2);
initiele_beta = [1,1,1];
fitnlm(x,y,modelfun,initiele_beta)
```

```

ans =
Nonlinear regression model:
y ~ (b1 + b2*x + b3*x^2)

```

Estimated Coefficients:

	Estimate	SE	tStat	pValue
b1	751.38	400.8	1.8747	0.10297
b2	-490.78	160.26	-3.0623	0.018263
b3	70.43	13.533	5.2044	0.001247

Number of observations: 10, Error degrees of freedom: 7

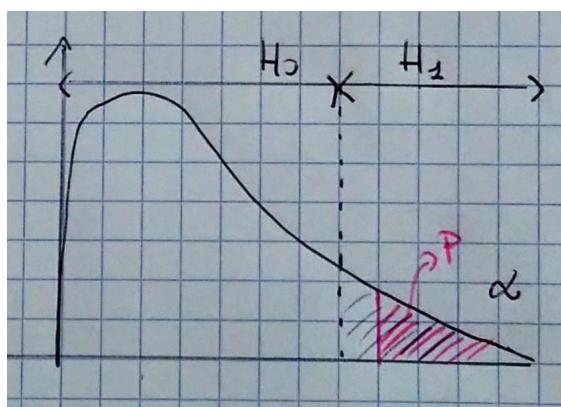
Root Mean Squared Error: 299

R-Squared: 0.947, Adjusted R-Squared 0.932

F-statistic vs. constant model: 62.7, p-value = 3.4e-05

`fitnlm` geeft ons redelijk wat informatie terug. De belangrijkste waarden zijn de geschatte coëfficiënten (linker kolom), hun p-waardes (rechterkolom) en de p-waarde voor de significantie van het model (rechtsbeneden). Let op: $\beta_0 = b_1; \beta_1 = b_2; \beta_2 = b_3$.

Het gevonden kwadratisch model is significant met 95% betrouwbaarheid ($p = 0.000034 < 0.05$).



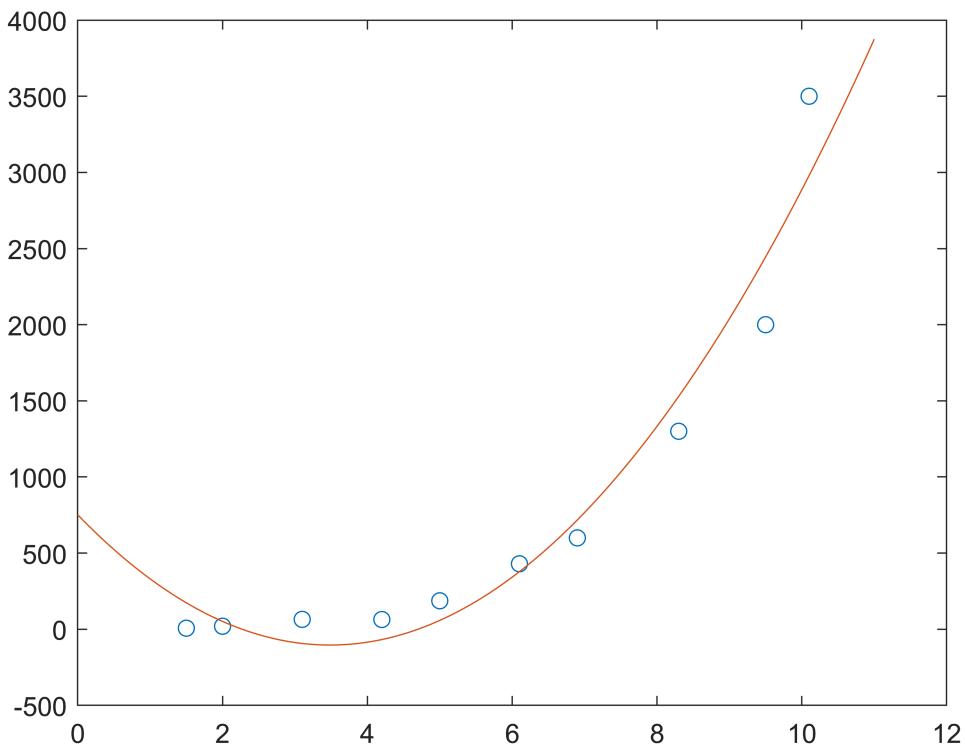
Het plotten van dit model geeft:

```

x1 = 0:0.05:11;
y1 = 751.38-490.78*x1+70.43*x1.^2;

plot(x,y, 'o')
hold on
plot(x1,y1)
hold off

```



Een ander vaak gebruikt model is dat van exponentiële groei:

$$y = \beta_0 \cdot \exp(\beta_1 x)$$

Het fitten van dit model doen we op net dezelfde manier:

```
modelfun = @(b,x)(b(1)*exp(b(2)*x));
beta0 = [10, 1];
mdl = fitnlm(x,y,modelfun,beta0)
```

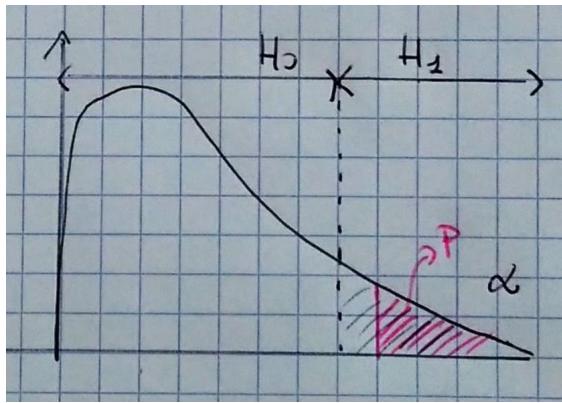
```
mdl =
Nonlinear regression model:
y ~ (b1*exp(b2*x))
```

Estimated Coefficients:

	Estimate	SE	tStat	pValue
b1	9.9019	4.7634	2.0787	0.071255
b2	0.57566	0.049472	11.636	2.7098e-06

```
Number of observations: 10, Error degrees of freedom: 8
Root Mean Squared Error: 153
R-Squared: 0.984, Adjusted R-Squared 0.982
F-statistic vs. zero model: 390, p-value = 1.06e-08
```

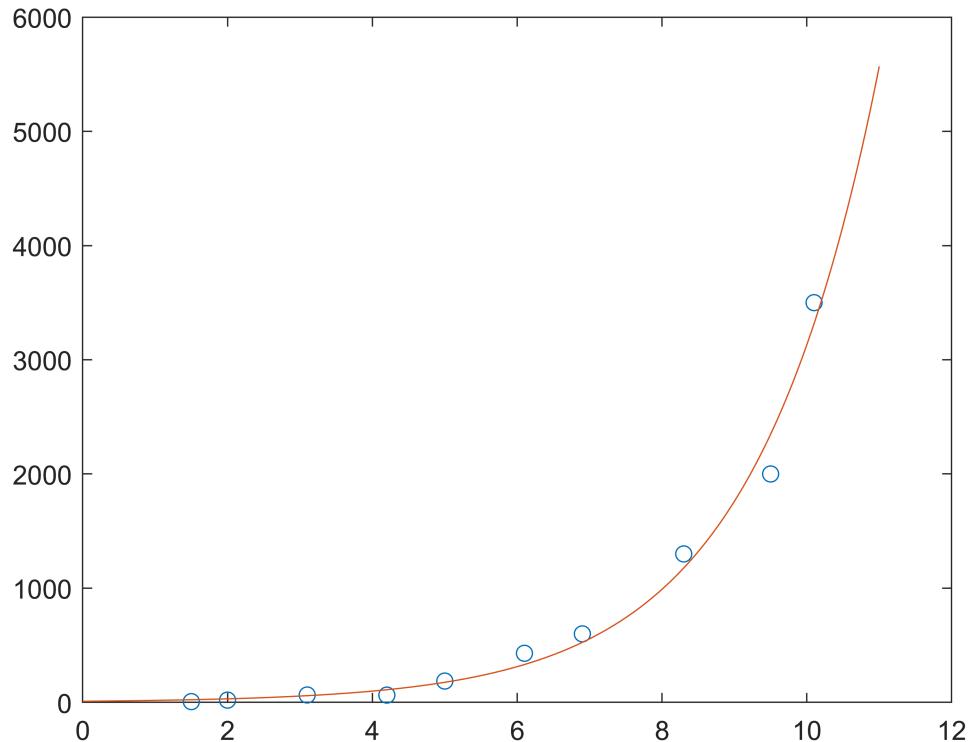
Ook dit model is met 95% betrouwbaarheid significant ($p = 1.06 \times 10^{-8} < 0.05$).



Ook dit model kunnen we plotten:

```
x1 = 0:0.05:11;
y1 = 9.9019*exp(0.57566*x1);

plot(x,y, 'o')
hold on
plot(x1,y1)
```



Zowel het parabolische als exponentiële model zijn significant. De R^2 van het exponentieel model is echter het grootst. Dit wijst erop dat het exponentiële model het best passende model is.

Oefening 93

Opgave:

Men wil bestuderen hoe de weerstand van rubber tegen afschuren beïnvloed wordt door de hardheid en treksterkte van rubber. Twintig monsters rubber zijn getest op de hardheid (hoe hoger het getal des te sterker is het rubber) en treksterkte (in kg/cm²). Daarna is elk monster een uur lang afgeschuurd (elk monster op dezelfde manier) en is het afgeschuurde gewicht (in gram) aan rubber per uur (y) bepaald. Bepaal het meervoudige lineaire model $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ en illustreer met een 3D tekening.

Oplossing:

De data van de te verklaren variabele wordt opgeslaan in de kolomvector y . De data van de verklarende variabelen (hardheid en treksterkte) slaan we op in kolomvectoren x_1 en x_2 .

```
y = [372 206 175 154 136 112 55 45 221 166 ...
      164 113 82 32 228 196 128 97 64 249]';
x1 = [45 55 61 66 71 71 81 86 53 60 64 68 ...
      79 81 56 68 75 83 88 59]';
x2 = [162 233 232 231 131 237 224 219 203 ...
      189 210 210 196 180 200 173 188 161 119 161]';
```

We maken een matrix van de verklarende variabelen en voegen een kolom van 1'en toe voor de constante term in het model:

```
x_ext=[ones(20,1),x1, x2]
```

```
x_ext = 20x3
    1    45    162
    1    55    233
    1    61    232
    1    66    231
    1    71    131
    1    71    237
    1    81    224
    1    86    219
    1    53    203
    1    60    189
    :
    :
```

We voeren de regressie uit met regress:

```
[b,bint,r,rint,stats] =regress(y,x_ext)
```

```
b = 3x1
721.7233
-6.6207
-0.6139
bint = 3x2
606.9419  836.5048
-7.7015   -5.5400
-0.9977   -0.2301
r = 20x1
47.6626
-8.5425
```

```

-0.4321
11.0575
-35.2300
5.8446
7.0709
27.1049
-25.2013
-42.4510
:
:
rint = 20x2
    8.2051    87.1201
   -60.9582    43.8731
   -54.2550    53.3907
   -42.8355    64.9506
   -82.3734    11.9133
   -47.2322    58.9214
   -45.2040    59.3459
   -21.7442    75.9541
   -77.0681    26.6655
   -92.9961     8.0940
:
:
stats = 1x4
    0.9081    83.9939    0.0000    696.5310

```

Het gevonden regressiemodel is: $y = 721.7233 - 6.6207 \cdot x_1 - 0.6139 \cdot x_2$

De belangrijkste voorwaarde voor het accepteren van het regressiemodel is of het al dan niet significant is. Dit controleren we door te kijken naar de uitkomst van de F-test met als hypothesen:

H_0 : Het regressiemodel is niet significant

H_1 : Het regressiemodel is significant

De p-waarde van deze test is het derde element in de stats-vector. Deze is gelijk aan 0.000. We kunnen aan Matlab meer beduidende cijfers vragen door het commando `format long` (het omgekeerde hiervan is `format short`):

```

format long
stats(3)

```

```

ans =
1.542052164876697e-09

```

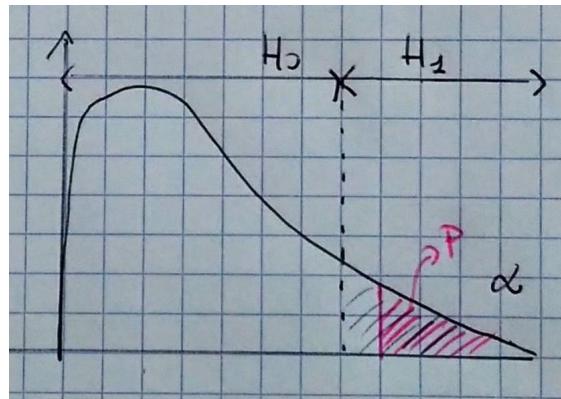
```

format short

```

We krijgen dus:

$p = 1.542 \times 10^{-9} < 0.05 \Rightarrow H_0$ verwerpen: het regressiemodel is significant met 95% betrouwbaarheid. De eerste voorwaarde is dus in orde.



De tweede voorwaarde is dat de coëfficiënten β_n allemaal significant verschillend zijn van nul. Dit doen we door te kijken naar de 95%-betrouwbaarheidsintervallen van de coëfficiënten. Deze vind je in de matrix bint, de rijen komen overeen met de grenzen van de betrouwbaarheidsintervallen van de verschillende coëfficiënten. De coëfficiënten zijn significant verschillend van nul met 95% betrouwbaarheid als nul niet in het 95%-betrouwbaarheidsinterval ligt. Dit is voor alle drie de coëfficiënten het geval, de tweede voorwaarde is dus voldaan.

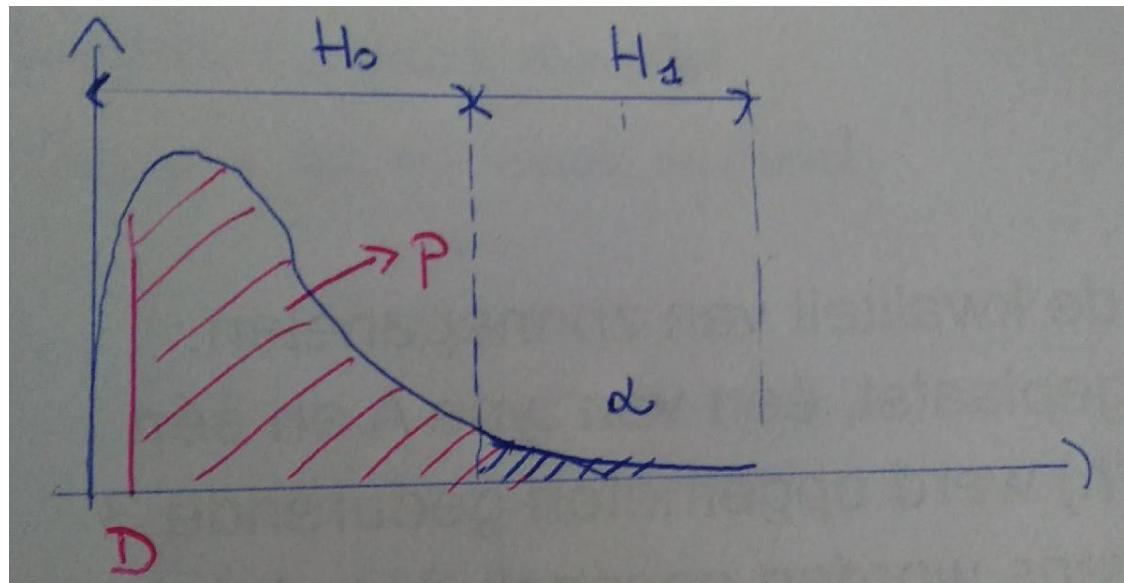
De derde voorwaarde is dat de residues normaal verdeeld moeten zijn. Dit controleren we met een KS-test:

H_0 : normaal verdeeld

H_1 : niet normaal verdeeld

```
test_cdf=makedist('Normal','mu',mean(r),'sigma',std(r));
[h,p]=kstest(r,'CDF',test_cdf)
```

```
h = logical
0
p = 0.9060
```



$p = 0.9060 > 0.05 \Rightarrow H_0$ accepteren met 95% betrouwbaarheid: residues zijn normaal verdeeld met 95% betrouwbaarheid. Deze voorwaarde is dus ook voldaan.

De vierde en laatste voorwaarde is dat er geen trend te zien is in de residues. Deze voorwaarde is ook voldaan aangezien de residues mooi verspreid zijn en er geen trend te zien is (dit is te zien in de finale 3D plot). Alle voorwaarden zijn voldaan en dus kunnen we de uitgevoerde regressie accepteren.

Met `scatter3` kunnen we een 3D scatterplot maken van de datapunten:

```
scatter3(x1,x2,y)
hold on
```

De meervoudige regressiefunctie kunnen we plotten met de methode `mesh`. Hiervoor moeten we eerst vectoren aanmaken met x_1 - en x_2 -waarden (`x1_list` en `x2_list`) en deze omzetten naar een grid met `meshgrid`. Daarna definiëren we de regressiefunctie aan de hand van de gevonden coëfficiënten in `b` en kunnen we de functie plotten met `mesh`. Als je een interactieve plot wilt moet je dit doen in je command window.

```
x1_list = min(x1):5:max(x1);
x2_list = min(x2):10:max(x2);
[x1grid,x2grid] = meshgrid(x1_list,x2_list);
yfit = b(1) + b(2)*x1grid + b(3)*x2grid;
mesh(x1grid,x2grid,yfit)
```

We voegen ook de juiste labels aan de drie assen toe:

```
xlabel('X1')
ylabel('X2')
zlabel('Y')
```

