|  |  |
| --- | --- |
| Biostatica  Door Alistair Vardy, Bart van Trigt en Mark Schrauwen | Matlab Wk2.1 |

Inhoudsopgave

[Versiebeheer 2](#_Toc496524143)

[1 Inleiding 3](#_Toc496524144)

[2 Herhaling functies vorige week 4](#_Toc496524145)

[3 Functies met meerdere outputs 4](#_Toc496524146)

[3.1 Vragen en opdrachten 9](#_Toc496524147)

[3.2 Antwoorden en uitwerkingen 10](#_Toc496524148)

[4 Functie met meerdere inputs 10](#_Toc496524149)

[4.1.1 Vragen en opdrachten 14](#_Toc496524150)

[4.2 Antwoorden en uitwerkingen 14](#_Toc496524151)

[5 Ingewikkeldere functies maken 16](#_Toc496524152)

[5.1 Vragen en opdrachten 19](#_Toc496524153)

[5.2 Antwoorden en uitwerkingen 20](#_Toc496524154)

[6 Specifieke terminologie en eigenschappen van Matlab 21](#_Toc496524155)

[6.1.1 Dimensie 21](#_Toc496524156)

[6.1.2 end 21](#_Toc496524157)

[6.1.3 Semicolon (;) puntkomma 21](#_Toc496524158)

[6.1.4 Colon (:) (de dubbele punt) 22](#_Toc496524159)

[6.1.5 Index 23](#_Toc496524160)

[6.2 Vragen en opdrachten 23](#_Toc496524161)

[6.3 Antwoorden en uitwerkingen 24](#_Toc496524162)

[7 Handige Matlab functies 24](#_Toc496524163)

[Vragen en opdrachten 26](#_Toc496524164)

[7.1 Antwoorden en uitwerkingen 27](#_Toc496524165)

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Beschrijving | Door |
| 0.0 | 11-07-2017 | Eerste versie | Alistair Vardy |
| 0.1 | 13-09-2017 | Under construction | Bart van Trigt |
| 0.2 | 14-09-2017 | Meerder outputs en meerdere inputs klaar  Nu verder met ingewikkeldere functies | Bart van Trigt |
| 0.3 | 18-09-2017 | Ingewikkeldere functies | Bart van Trigt |
| 0.4 | 19-09-2017 | Afronding versie 1 | Bart van Trigt |
| 0.5 | 02-10-2017 | Paar opmerkingen van Chadier Wilson en Denice Vis verwerkt. | Mark Schrauwen |
| 0.6 | 23-10-2017 | Opmerkingen Herre, Mark, Chadier en Denice verwerkt | Bart van Trigt |

# Inleiding

Vorige week heb je kennis gemaakt met MATLAB. Als het goed is weet je nu waarom jij als bewegingstechnoloog MATLAB moet gebruiken. Vorige week heb je geleerd wat operatoren zijn, hoe je kan debuggen, hoe je vectoren maakt in MATLAB, standaard functies in MATLAB gebruikt en je hebt al een keer een functie zelf gemaakt! We gaan in deze reader verder met het maken van een functie. Tot nu toe heb je makkelijke functies gemaakt; je gaf een input en er kwam een output uit. Maar wat nu als we bijvoorbeeld een formule hebben die meerdere inputs heeft.

De volgende leerdoelen zijn hiervoor gemaakt, na het doorwerken van deze reader kan de student:

1. In eigenwoorden uitleggen wat een functie in Matlab is.
2. Gebruik maken van standaard functies in Matlab.
3. Een functie met meerdere outputs aanroepen.
4. Een functie met meerdere inputs aanroepen.
5. Zelf een functie aanmaken in Matlab met meerdere inputs en outputs.
6. Uitleggen dat een matrix rijen en kolommen bevat.
7. Matlab terminologie interpreteren en in eigenwoorden uitleggen, zoals: colon, operator, indices, etc.
8. De help functie raadplegen en uitleggen waarom deze belangrijk is bij het gebruik van Matlab.

**Zie je een fout? Of heb je een aanbeveling dan horen we dat graag! Stuur dan een e-mail naar** [**mjschrau@hhs.nl**](mailto:mjschrau@hhs.nl) **en wij passen het dan z.s.m. aan.**

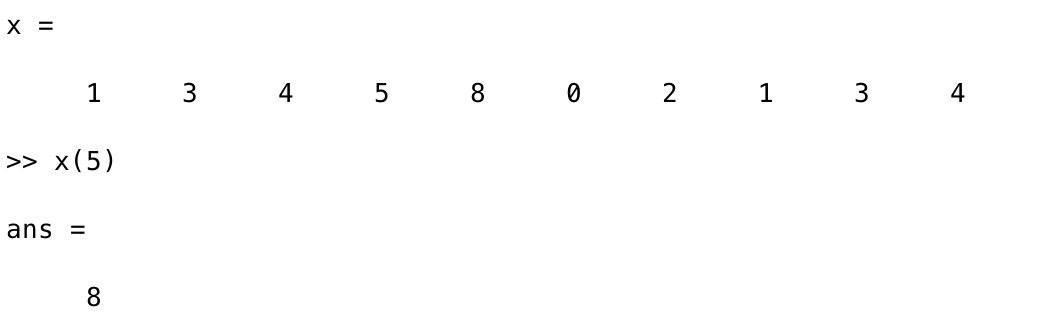
# Herhaling vorige week

Vorige week heb je kennis gemaakt met functies. Deze reader gaat hier verder op in. Om deze reader goed te begrijpen is het van belang dat je de reader van vorige week hebt afgerond. Mocht je dit niet gedaan hebben zorg ervoor dat je dit eerst afmaakt (reader 1\_2, hoofdstuk 4). We gaan in deze reader meer leren over functies met meerdere inputs en outputs. Succes!

# Functies met meerdere outputs

We beginnen met een functie die meerdere outputs geeft. Stel je wilt bij een 100 meter sprint van Usain Bolt weten op welk moment zijn snelheid maximaal is. Nu gaat er vast een lampje branden, vorige week heb je geleerd dat er een functie max(x) is. Waarmee je een maximale waarde kan berekenen. Als output krijg je dan de maximale waarde, maar we willen meer weten, namelijk bij welke afstand is dit? Dit kan je doen door te kijken naar een index nummer.

**Wat is een index?**  
Voordat we het gaan berekenen waar de snelheid van Usain Bolt maximaal is moeten we weten wat een index is. Index betekend letterlijk dat je waardes selecteert die horen bij bepaalde elementen. In figuur 2 zie je een vector x. Je kunt het vijfde element van een vector x opvragen met het commando x(5). Het getal 5 tussen de haakjes is hier de index. Dus wanneer je van het onderstaande plaatje 5 stapjes telt zie je dat je bij 8 uitkomt.



Figuur Hierboven is een vector x te zien, doormiddel van de index op te vragen tussen haakjes krijg je de waarde 8 hieruit.

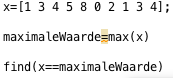
Je kan het ook op de ander kant op doen, stel je wilt weten welke index bij 8 hoort dan kan je het volgende doen (zie figuur 3). Je zoekt naar de waarde 8 en daar komt de index 5 uit.



Figuur

Het kan veel tijdkosten, wanneer je een hele lange vector hebt, om de maximale waarde te vinden. Door gebruikt te maken van de functie max() krijg je daar meteen de hoogstewaarde uit. Wanneer je die hebt kan je met find(x==*’de gevonden waarde’*) invullen en weet je het index cijfer van die maximale waarde.

**Voer het onderstaande commando uit, wat krijg je als antwoord?**



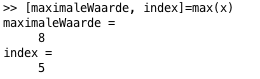
Figuur

Zoals je verwacht had zal je nu als indexwaarde 5 vinden. Maar dit kost veel programmeer regels en daarom is er bedacht om aan een functie meerdere outputs mee te geven. Dit kan je doen met blokhaken voor de functie aanvraag te zetten en ziet er als volgt uit:

**[output1, output2] =functienaam(input)**

Dit gaan we bekijken aan de hand van de max() functie. Bij de functie max() kan je naast dat je de maximale waarde eruit krijgt, ook meteen het indexnummer eruit krijgen van die maximale waarde.

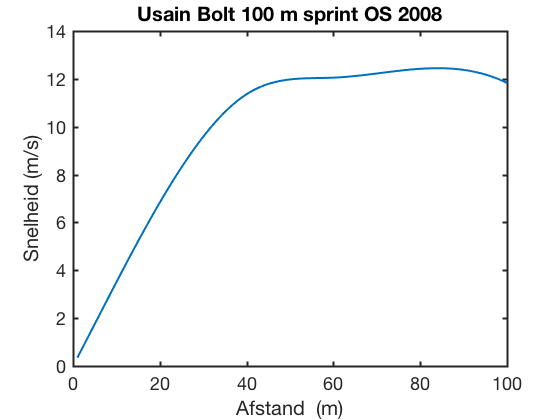
**Vul de onderstaande code in:**



Wat is de waarde die je eruit krijgt en wat is de index waarde?

Nu heb je een voorbeeld gezien met een functie waaruit meerder outputs komen, hoe kan je dit nu in de praktijk gebruiken?

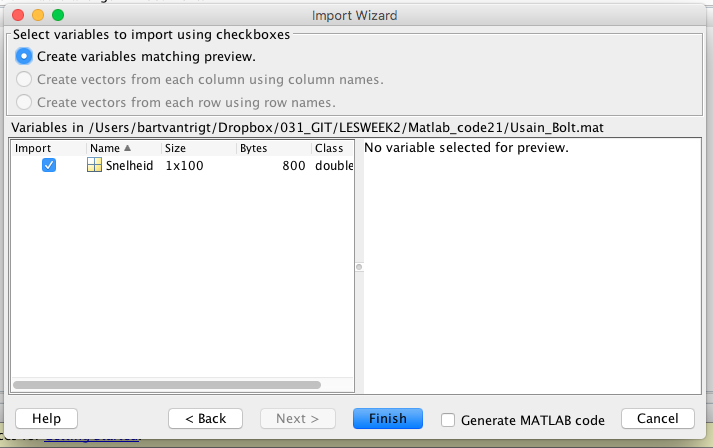
**Terug naar Usain Bolt.**  
Nu gaan we weer terug naar het voorbeeld van Usain Bolt. We hebben Usain Bolt zijn snelheid gemeten met versnellingssensoren, waar mee we door te differentiëren de snelheid krijgen (Dit ga je leren bij kinematica in het volgende blok). In de figuur hieronder zie je de snelheid tegen de afstand van Usain Bolt op de 100 m tijdens de Olympische Spelen van 2012 in Beijing.



Figuur de snelheid tegen de afstand op een 100 meter sprint van Usain Bolt tijdens de OS van 2008 in Beijing.

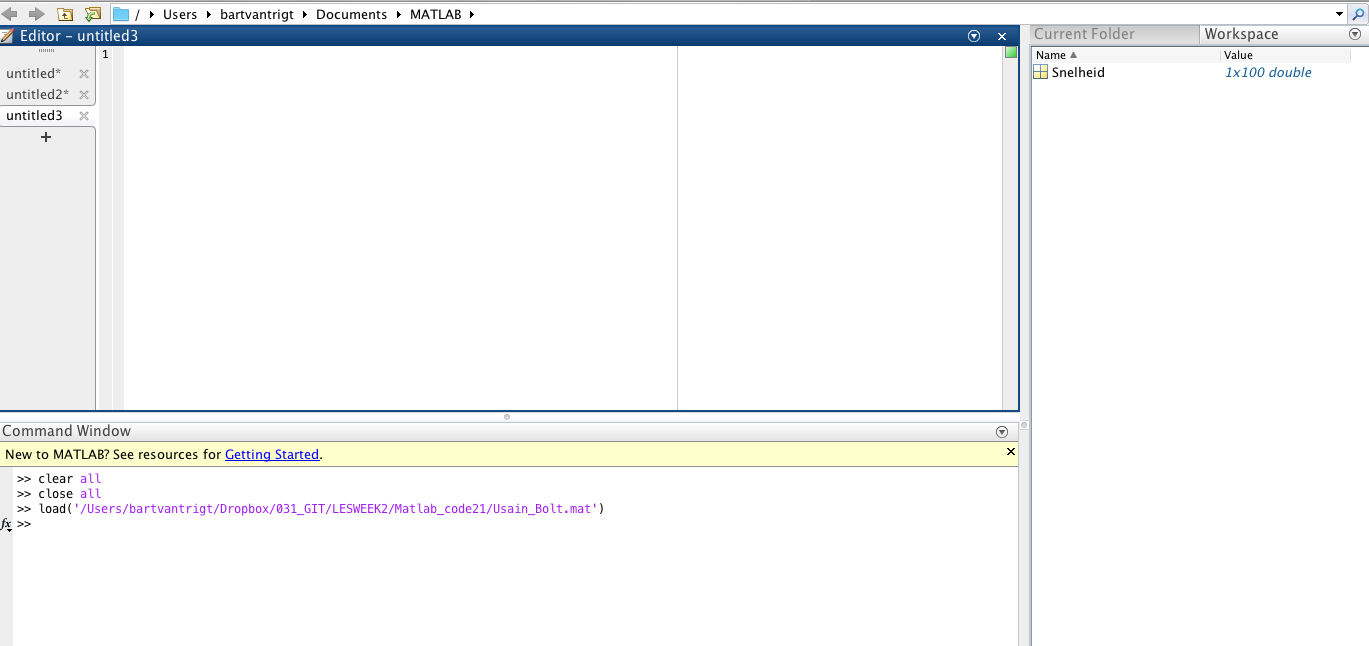
**We willen nu weten wat zijn maximale snelheid is en bij hoeveel meter dat is. Dat gaan we als volgt doen:**

* Maak eerste je workspace leeg met clear all.
* De snelheid staat onder de naam Usain\_Bolt.mat op blackboard. Open het bestand Usain\_Bolt.mat door er dubbel op te klikken, vervolgens krijg je import wizard, zoals je hieronder ziet, klik dan op finish.



Figuur import wizard dat laat zien hoe je een variabele load

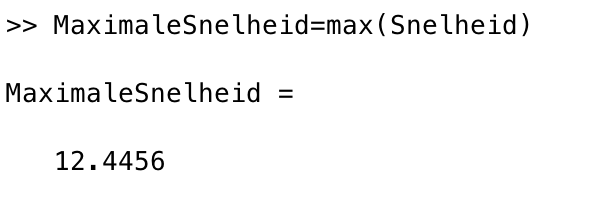
Nu verschijnt er een variabele Snelheid in je Workspace, zoals je kan zien in figuur 7 is er rechtsboven in de Workspace de variabele Snelheid toegevoegd.



Figuur De variabele snelheid is hier toegevoegd aan je Workspace.

De variabele Snelheid die nu in je workspace staat is een vector. Deze Snelheidsvector is zo gemaakt dat elke sample een meter is.

1. Gebruik nu de functie max. Vul in maximaleSnelheid=max(Snelheid)



Figuur output van de functie met een output variabele

We weten dus dat zijn maximale snelheid 12.44 m/s is, maar we weten nog steeds niet op welk moment. Dit kunnen we doen door nog een output te krijgen. Dit doe je door voor je functie aanvraag blokhaken te zetten, tussen de blokhaken kan je meerdere output variabele zetten, dat ziet er dan als volgt uit:

**[output1, output2] =functienaam(input) 🡪 Algemeen**

**[maximaleSnelheid, index]=max(Snelheid) 🡪 voor deze functie**

**Voer dit in in Matlab.**  
Op hoeveel meter is Usain Bolt op zijn maximale snelheid? Klopt dit met de waardes in figuur 5. Laat je antwoord zien aan de docent.

## Vragen en opdrachten

Beantwoord de volgende vragen met juist of onjuist:

1. We hebben y =[3 6 7 12 8 15 20 2 4]; Om het maximum te bepalen maken we gebruik van het commando max[y].
2. Een functie ziet er als volgt uit: [output]=functienaam(input)
3. We hebben een commando max en willen het indexnummer weten dit doen we met het volgende statement: [hoogstewaarde,index]=max(x)
4. Je wilt meerdere outputs berekenen hiervoor gebruik je { output1, output2 }=min(x)

Voer het volgende commando in:

x = [20 11 11 5 10 23 14 8 8 20 11 4 5 ];

1. Bepaal de maximale en index waarde van vector x.
2. Bepaal de minimale en index waarde van vector x. (Tip: bedenk wat je meegeeft als output naam)

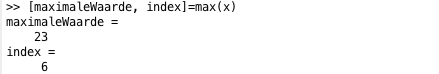
a = [20 11 11 5 10 2 14 8 8 20 19 4 5 ];

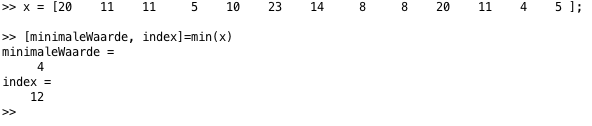
1. Bekijk de bovenstaande vector, wat is zijn de maximale waardes en wat zijn de indexen hierbij? Doe dit zonder het in Matlab uit te voeren.

Voer de bovenstaande vector a in Matlab in.

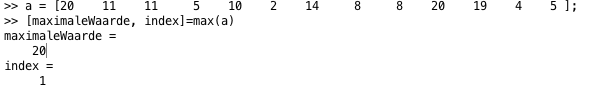
1. Wat geeft Matlab je als antwoord terug?
2. Wat krijg je terug wanneer je find(a==20)

## Antwoorden en uitwerkingen

1. Het antwoord is onjuist, het commando bevat ronde haken, geen blokhaken.
   1. Functies worden aangeroepen als volgt: max() whos () size(). Echter in deze vraag zie je dat de functie max wordt aangevraagd met behulp van blokhaken [], en niet van ronde haken. Deze stelling is dus fout omdat de functie een syntaxfout bevat.
2. Het antwoord is juist.
3. Het antwoord is juist, met dit statement bereken je de hoogste waarde en de index.
4. Het antwoord is onjuist, je gebruikt blokhaken [ ] en geen accolade { }.
5. Antwoord:
6. Antwoord:



1. De maximalewaardes zijn 20 bij de index 1 en 10.
2. Wanneer er twee even groten maximale waardes zijn geeft matlab maar één index nummer terug. Let dus goed op!



1. Wanneer je dit command uitvoert krijg je nu wel twee indexen terug. Dit kan altijd nog uitvoeren na de max functie om te controleren of er niet meer maximale waardes zijn.



# Functie met meerdere inputs

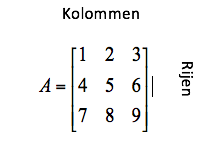
Net hebben we gezien dat een functie meerdere outputs kan hebben. Je raadt het al: een functie kan ook meerder inputs hebben.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Kolom 1 | Kolom 2 | Kolom 3 |
| Rij 1 | 1 | 2 | 3 |
| Rij 2 | 4 | 5 | 6 |
| Rij 3 | 7 | 8 | 9 |

Bekijk de volgende matrix. Een matrix kan gezien worden als soort tabel. LET OP! Dit is een versimpeling in de toekomst ga je meer leren over matrices maar nu vatten we het even op als een tabel. Een matrix bestaat, net als een tabel, uit rijen en kolommen.

Leuk weetje:

De eerste drie letters van MATLAB staat voor matrix en de laatste drie voor laboratory

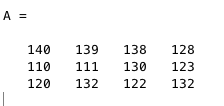


Het berekenen van het gemiddelde van een vector hebben we vorige week al gedaan. Dat doe je met het commando mean(x), de x is hier je vector. Van een matrix kunnen we de rijgemiddelden en de kolomgemiddelden uitrekenen. Echter, we zullen dit wel moeten aangeven. Een matrix/tabel heeft twee dimensies; de eerste zijn de rijen, de tweede de kolommen. Bij het op vragen van een element van een matrix heb je twee indices nodig in plaats van één index. Namelijk een rij en een kolom index. Om element 8 te krijgen kunnen we deze aanroepen met het volgende command: A(3,2). Je gaat 3 rijen naar beneden en twee kolommen naar rechts. A(rijen,kolommen)

Stel, je hebt een bloeddrukmeting gedaan bij drie proefpersonen. Zoals je weet moet je een meting meerdere keren uitvoeren wil deze betrouwbaar zijn, dus je hebt vier keer de systolische (boven)bloeddruk gemeten. De namen van proefpersonen staan vermeld in de rijen en de waardes van de metingen in de kolommen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Meting 1 | Meting 2 | Meting 3 | Meting 4 |
| Mark | 140 | 139 | 138 | 128 |
| Bart | 110 | 111 | 130 | 123 |
| Hubert | 120 | 132 | 122 | 132 |

**Van de meetgegevens is de volgende matrix gemaakt:**



**Kopieer de onderstaande code naar je Command Window en je hebt dezelfde matrix.**

A= [140 139 138 128; 110 111 130 123; 120 132 122 132]

Nu willen we het gemiddelde per proefpersonen berekenen over de vier metingen. Dit kan je doen door het volgende commando in te voeren mean(A,2). Je geeft hierin de matrix A mee en met de tweede input geef je aan dat je de tweede dimensie wilt hebben. Je krijg hier dus nu een kolomvector uit.

**Vraag: Wie heeft de laagste gemiddelde bloeddruk en hoeveel is die?**  
antwoord: Bart met 118.5

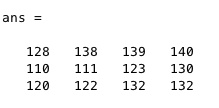
Je kan ook middelen over de kolommen. Dit kan je doen om te controleren of misschien de eerste meting gemiddeld hoger is dan de laatste meting. Dit zou je kunnen doen omdat proefpersonen tijdens de eerste meting misschien meer zenuwen hebben vergeleken met de laatste meting, omdat tijdens de eerste meting alles nieuw is. Wanneer je dit doet zal er een rijvector overblijven. Matlab telt dan alle rijen bij elkaar op en middelt ze dan over de rijen. Dit doe je dan niet met mean(A,2) maar met mean(A,1). Dus over de eerste dimensie en dit zijn de rijen.

**Voer dit commando uit. Is er gemiddeld een hogere of lagere bloeddruk tijdens de eerste meting?**

Stel, we willen de bloedrukmetingen op volgorde van grootte zetten per proefpersoon. Hiervoor heeft Matlab ook een ingebouwde functie namelijk sort. Bij het aanvragen van de functie sort geef je zelfs drie inputs mee aan de functie. Namelijk de matrix, de dimensie en een string waarmee je aangeeft dat het oplopend (ascend in het Engels) moet zijn. Je kan zien dat het een string is omdat je het tussenhaakjes zet.

**Voer het volgende commando in**:





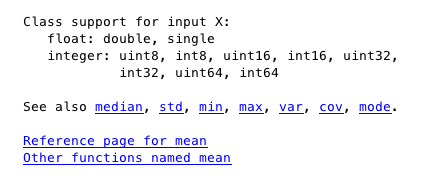
Je ziet nu dat de metingen oplopen per proefpersoon. Hiermee is de matrix wel helemaal door elkaar gehusseld en kan je niet meer achterhalen of een bepaalde meting de eerste of de derde was. Daarom is het misschien handig, afhankelijk van wat je er verder nog mee moet doen, een kopie van de originele matrix te bewaren.

**De help functie!**

Vorige week heb je al gelezen over de help functie, maar omdat die zo belangrijk is herhalen we het hier nog een keer! Dit doen we omdat er zoveel functies zijn die we nooit kunnen allemaal kunnen onthouden. Ook het aantal inputs en outputs is niet te onthouden. Daarom maken we gebruik van de help functie.

Voer maar eens in je Command Window: **help mean**

Er verschijnt nu een hele uitleg over wat de input en outputs zijn van de functie. Om het wat overzichtelijker te maken kan je onderaan op de **reference page for mean** klikken.

****

Dit kan je voor elke functie doen. De help functie is je grootste vriend, ook al is het soms lastig te lezen in het Engels. Weet je dat je dat je Engelse woorden gewoon kunt intikken in Google met als extra zoektermen ‘Engels’ en ‘vertaling’. Dat wil weleens helpen

Weet je nog hoe je deze reference page ook kan openen uit je Command Window? **Doe dit voor de functie mean.**

**Gebruik de help functie om erachter te komen hoe je de waarden aflopend kunt sorteren en voer dit ook uit.**

**Gebruik de help ook eens om de functie sortrows te bekijken. Wat is het verschil met de gewone sort functie?**

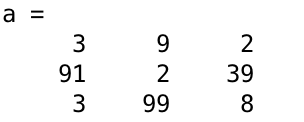
## Vragen en opdrachten

Beantwoorde de volgende vragen met juist of onjuist:

1. Een functie kan meerdere inputs hebben.
2. Wanneer je een element wilt oproepen van een Matrix gebruik je: A(kolommen,rijen)
3. Een functie kan je aanroepen vanuit je Command Window en je script.
4. Met sort(a,2,’descend’) kan je de matrix a laten **aflopen** over de kolommen.
5. Met sort(a,1,’ascend’) kan je de matrix a laten **oplopen** over de kolommen.

Open vragen:

1. Bereken de maximale waardes per rij van de onderstaande matrix. Je krijg hier dus een kolomvector uit. (Tip, raadpleeg de help functie)



1. Wat gebeurt er wanneer je maar 1 input invoert, terwijl een functie wel meerdere kan hebben?
2. Hieronder zie je een Matrix, hoeveel rijen en kolommen heeft deze matrix?



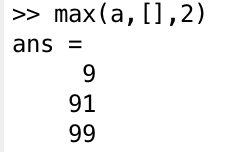
1. Bereken het gemiddelde van bovenstaande matrix voor iedere kolom.
2. Voer de a = [ 9 8 13; 4 3 15; 6 1 19] in, in Matlab. Hoe kan je het element 15 opvragen in Matlab? (Je mag niet spieken!)

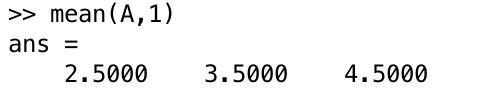
## Antwoorden en uitwerkingen

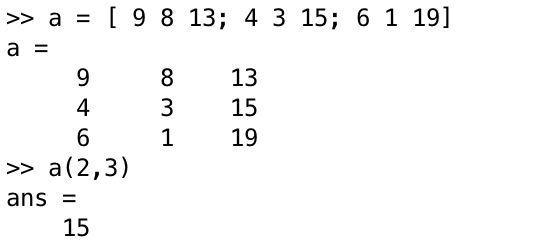
1. Juist
2. Onjuist, het juiste antwoord is: A(rijen,kolommen). Dus andersom!
3. Juist
4. Juist
5. Onjuist, hiermee laat je ze oplopen over de rijen.

Open vragen:

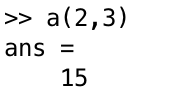
1. In de help functie zie je bij syntax twee hoe je het moet aanroepen voor een matrix:



1. Dan neemt de functie gewoon de default voor deze inputs.
2. Deze matrix heeft 2 rijen en 3 kolommen.
3. Wanneer je het gemiddelde van elke kolom wilt weten krijg je dus een rijvector. 
4. Wanneer de code wordt ingevoerd geeft matlab de onderstaande matrix a terug.



We zien dus dat 15 in de tweede rij staat en in de derde kolom. Weet je nog dat een matrix dimensies heeft? De eerste dimensie is voor rijen, de tweede dimensie zijn de kolommen.   
Het element kan dus worden opgevraagd aan de hand van a(2,3)



# Ingewikkeldere functies maken

Nu gaat het echt spannend worden (#FISSA), want we gaan een ingewikkeldere functie maken. We hebben net geleerd dat een standaard functie namelijk meerdere inputs en meerdere outputs kan hebben. Bij zelf gemaakte functies kan dat ook!

Met Kinovea kan je de kniehoek berekenen van een fietser. We gaan nu een functie maken waaruit we meteen het gemiddelde, de mediaan, de maximale kniehoek en de minimale kniehoek kunnen bepalen van een fietser. De input data in je functie is de kniehoek in graden en je output data is je gemKniehoek, mediaanKniehoek, maxKniehoek en minKniehoek in graden. Voordat je een functie gaat aanmaken, is het handig om een schematisch beeld te maken van hoe je functie eruit komt te zien.

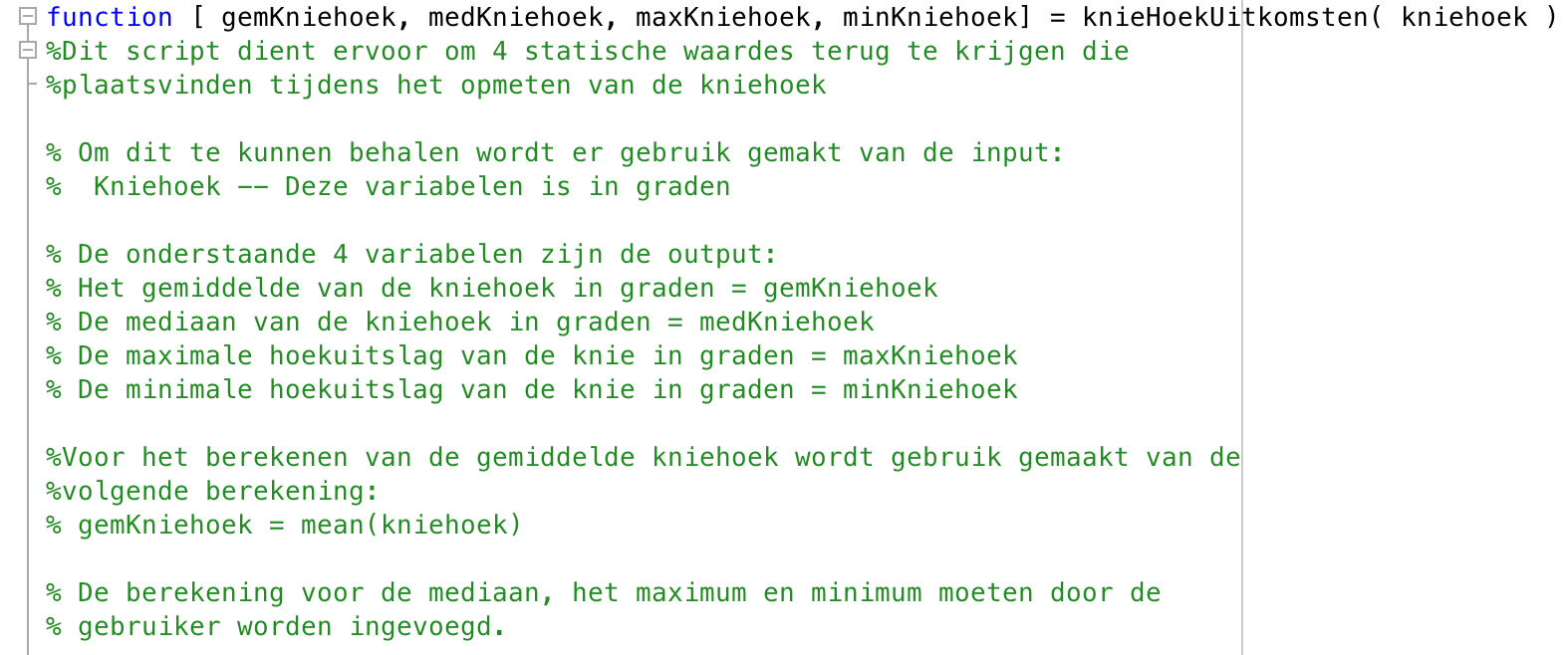
* 1. **Teken het onderstaande figuur na en vul in de figuur je input en je output variabelen in.**



* 1. **Maak een nieuwe functie aan met de naam KniehoekUitkomsten.m en sla deze op.**

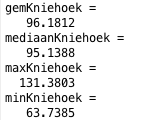
(Weet je niet meer hoe je een functie aanmaakt lees dit dan nog eens terug in de reader wk 1.2.)

* 1. De eerste regels zullen er ongeveer zo uitzien, de berekening van het gemiddelde is al gegeven



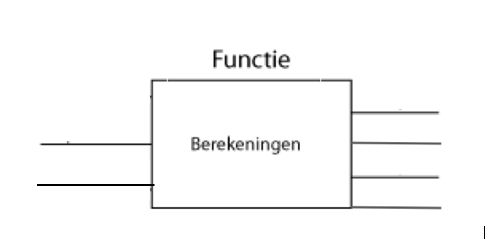
Figuur

* 1. **Voeg nu de code toe om de mediaan, maximale kniehoek en minimale kniehoek te berekenen toe.**
  2. **Open het bestand fietsenkniehoek.mat of sleep deze naar je workspace. Hierin staat de variabele kniehoek gegeven in graden en de variabele kniehoekSnelheid gegeven in graden/sec. (Weet je niet meer hoe je een mat bestand opent kijk dan nog eens naar hoofdstuk 3 in deze reader)**
  3. **Roep nu je functie aan in je Command Window of met een script. Komen je antwoorden overeen met hieronder? Zo ja, dan heb je het goed gedaan, anders moet je je functie nog aanpassen**

****

Stel we willen de maximale kniehoek snelheid weten. Daarvoor hebben we de kniehoek snelheid nodig en dus moeten we een extra input meegeven in de functie en ook een extra output.

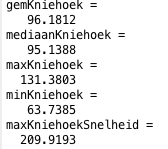
* 1. **Voeg de kniehoek snelheid toe aan je schematische figuur, zodat je het overzicht behoud.**



1. **Je functie zal er nu zo uitkomen te zien.**

function [ gemKniehoek, mediaanKniehoek, maxKniehoek, minKniehoek, maxKniehoekSnelheid ] = KniehoekUitkomsten( kniehoek, kniehoekSnelheid )

1. **Vul je functie verder met de code om de maximale kniehoek snelheid te berekenen.**
2. **Klopt je antwoord met hieronder wanneer je de functie opnieuw aanvraagt?**

****

## Vragen en opdrachten

Beantwoord de volgende vragen met juist of onjuist:

1. Wanneer je een functie maakt kan je zoveel inputs meegeven als je wilt.
2. Je kan een functie in een functie gebruiken.
3. Je kan een functie alleen aanroepen met je ‘command window’ en niet in een script

Open vragen:

1. Waarom gebruikt een BT’er standaardfuncties in Matlab?
2. Waarom maak je als BT’er je eigen functies?
3. Type in Matlab de volgende code: VectorA = [1 3 15; 12 4 6; 3 5 7]. Roep nu de hulpfunctie voor max aan. Met behulp van welke code kan je het hoogste getal van VectorA oproepen, het getal 15?

## Antwoorden en uitwerkingen

Antwoorden gesloten vragen:

1. Juist
2. Juist, dat heb je net gedaan bij het maken van je eigen functie, bijvoorbeeld doordat je in je functie de functie mean hebt gebruikt
3. Onjuist, je kan juist ook een functie aanvragen in je script, dit zorgt ervoor dat je netjes programmeert.

Antwoorden open vragen:

1. Standaardfuncties besparen programmeurs en bewegingstechnologen veel tijd in het ontwikkelen van hun algoritmes. Dat is dus erg prettig.
2. BT’er komt het vaak voor dat je iets wilt berekenen, maar er geen standaard functie voor is. Daarom maak je je eigen functie om zo netjes te programmeren en dat je snel veel berekeningen kan uitvoeren in plaats van alles op papier uit te rekenen.
3. Als je de functie aanroept zie je dat hij veel verteld over het berekenen van het grootste getal, echter in Matlab staat niet uitgelegd wat de code is voor het grootste getal in de matrix. Je zal vast gedacht hebben: “ah ik moet vast de reference page for max gebruiken”. Onder het kopje ‘largest element in matrix’ is de code te vinden: *If you only need the maximum value of A and not its index then call the max function twice:* ***Het antwoord is max(max(VectorA)),***

# Specifieke terminologie en eigenschappen van Matlab

We hebben nu veel geleerd over functies. In dit hoofdstuk verklaren we een aantal Matlab termen en functies nader die het eenvoudiger maken om een functie te schrijven of data te analyseren. Probeer ze allemaal uit in Matlab zodat je precies ziet wat ze doen.

### Dimensie

Een vector of matrix heeft bepaalde dimensies, of afmetingen. In Matlab wordt met de eerste en tweede (en verdere) dimensies van een matrix specifiek de rijen (eerste dimensie) en kolommen (tweede dimensie) bedoeld.

### end

Hiermee geef je het laatste element van een vector aan. Voer het volgende prompt in in Matlab:

v= [ 3 5 6 8]

Dan is het resultaat van het commando v(end) gelijk aan 8. Dit is buitengewoon handig als je niet van tevoren weet hoe lang een vector zal zijn. Weet je nog wat je krijgt als je v(2) doet? Probeer het eens uit.

### Semicolon (;) puntkomma

Met het semicolon kan je ervoor zorgen dat de Command Window je geen output geeft. Probeer eens het volgende:

**Voer de onderstaande code in.**

Clear all

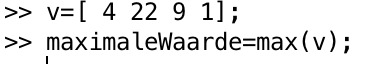
v=[ 4 22 9 1]

maximaleWaarde=max(v)

Je ziet nu dat er maximaleWaarde = 22 komt te staan in je Command Window.

**Voer nu de bovenstaande code opnieuw uit maar zet achter alle regels een semicolon.**

Nu is er geen output gekomen in je Command Window zoals je hieronder kan zien. Het gebruik van het semicolon is heel nuttig want dit zorgt ervoor dat je programma snel blijft!



**Een ander gebruik van de semicolon**



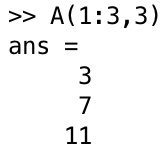
A=[ 1 2 3 4; 5 6 7 8; 9 10 11 12]

* 1. Maak de matrix A aan in Matlab. Een matrix aanmaken doe je door een nieuwe rij te beginnen met een puntkomma(;). Dus zoals je hierboven kan zien komt er na de 4 en de 8 een puntkomma.

### Colon (:) (de dubbele punt)

Om van een matrix een hele rij of een hele kolom op te vragen kun je een dubbele punt gebruiken.

**Als we de derde kolom op willen vragen dan kunnen we dat doen met het commando A(1:3,3), rijen 1 t/m 3 van de derde kolom**. Controleer je antwoord met hieronder:



* 1. Het volgende commando werkt ook: A(1:end,3). Hiermee vraag je rijen 1 t/m de laatste van de derde kolom op.
  2. Maar je kun dit nog eenvoudiger doen met het volgende commando: A(:,3). Dit geeft alle elementen uit kolom 3. Krijg je weer dezelfde antwoorden als hierboven? Dan gaat het goed! Probeer het nu eens voor de tweede kolom.

### Index

Deze term is uitgelegd in het hoofdstuk: ‘functie met meerdere inputs’

## Vragen en opdrachten

Waar onwaar vragen:

1. Met de puntkomma achter een statement zorg je ervoor dat je geen waardes krijg te zien in je Command Window maar wel in je Workspace.
2. Wanneer je van de onderstaande matrix A het commando A(:,:) uitvoert krijg je de hele matrix te zien
3. Wanneer je van de Matrix A het commando A(:,1:2) uitvoert krijg je de waardes van alle rijen en de eerste en derde kolom eruit.



Openvragen:

v=[ 1 2 4 6 9 2 ]

* + - 1. Maak van de bovenstaande vector (v) een 2 bij 3 matrix en noem deze A.
      2. Zorg er nu voor dat je geen Matrix ziet in je je Workspace.



* + - 1. Laat zien hoe je de laatste waarde 2 uit de Matrix krijgt door het commando end te gebruiken.
      2. Haal de waarde 3 uit de Matrix.

## Antwoorden en uitwerkingen

Antwoorden waar onwaar vragen:

* + - 1. Waar
      2. Waar, want met de dubbele punt krijg je alles te zien
      3. Onwaar, je krijg hier de waardes van alle rijen met de eerste en tweede kolom uit.

Antwoorden openvragen:

* + - * 1. Dit kan je doen door achter de 3 een puntkomma (;) te zetten.

Antwoord: v=[1 2 4; 6 9 2]

* + - * 1. Dit doe je door een puntkomma achter je statement te zetten.

Antwoord: A=[1 2 4; 6 9 2];

* + - * 1. Dit doe je met A(end) of nog netter met A(end,end)
        2. Dit doe je met A(1,end) of door A(1,3)

# Handige Matlab functies

Hieronder volgt een beschrijving van een aantal handige Matlab functies, er zijn er nog veel meer functies en deze zal je tijdens je opleiding allemaal nog gaan ontdekken, degene die hier besproken worden zal je het meeste gaan gebruiken.

Maak eerste een matrix(A) en een vector (v) aan in matlab, zoals je hieronder kan zien:

We gaan nu op deze data wat functies loslaten, voer voor elke functie de A en de v in. Erachter staat beschreven wat de functie doet.

### size()

Dit commando geeft de afmetingen van een vector of matrix. Voor de matrix geeft het commando size(A) de vector [2 4] terug. De matrix heeft dus 2 rijen en 4 kolommen. Voor de vector geeft size(v) de vector [1 4] terug; een rij en 4 kolommen. Je kan dit controleren door te kijken naar je workspace, komen ze overeen? Neem nu de transponent van A met een ’en bereken de size, wat is deze nu?

Antwoord: 3-bij-4

### isempty()

Dit commando geeft aan of een variabele leeg is. Als de variabele leeg is, krijg je de logische waarde 1 terug. Als de variabele *niet* leeg is, krijg je een logische nul terug.

### numel()

Dit commando geeft het aantal elementen van een variabele aan. Hoeveel is dat voor de matrix A en hoeveel voor die van v? Weet je het antwoord niet uit je hoofd? Geen probleem voer het in in Matlab!

### find()

Find is een hele handige functie, omdat je hiermee de indices kan achterhalen van elementen met een specifieke inhoud. **Vul in** find(v==8). Dit geeft alle indices van de vector v waar een 8 staat. Stel je wilt alle waardes groter dan 6 weten. Wat krijg je wanneer je find(v>6) invult? **Controleer je antwoord door het in te vullen in matlab.**

### strcmp() en strcmpi()

Met deze commando’s kun je bepalen of twee strings aan elkaar gelijk zijn. Maak twee variable aan in de vorm van een string met je naam met een hoofdletter en de andere met een kleine letter.

A= ‘Bart’

B= ‘bart’

**Run nu de volgende twee codes.**

strcmp(A,B) en strcmpi(A,B)

**Wat is het verschil dat je eruit krijgt? En wat betekent het? Weet je het niet zoek het op in de help functie.**

Het commando strcmp() houdt wel rekening met hoofdletters, het commando strcmpi() is ongevoelig (*insensitive*) voor hoofdletters.

## Vragen en opdrachten

1. Maak in Matlab een nieuwe variabele aan S1 = ‘gastrocnemiusLateralis’ en een S2 = ‘gastrocnemiusMedialis’. Roep de strcmp() functie aan zodanig dat je de string S1 vergelijkt met S2. Wat krijg je als antwoord terug? Wat betekent het getal dat Matlab teruggeeft?
2. Maak nu van S2=’GastrocnemiusLateralis’ en vergelijk S1 en S2 opnieuw. Wat geeft matlab je terug?
3. Vergelijk nu S1 en S2 met de functie strcmpi, wat betekent het getal nu dat Matlab teruggeeft?
4. Size() geeft de afmetingen van een vector of matrix. Stel de functie geeft het volgende terug ans = 2 3. Stelling: dit betekend dat de matrix 2 kolommen en 3 rijen heeft. Waar/ niet waar.
5. Maak een string aan met een anatomisch naam, bijvoorbeeld str = ‘tuberositas’; Bepaal de afmetingen en het aantal elementen.
6. Maak een variabele v aan met v=[]; wat komt er uit wanneer je de functie isempty(v) gebruikt en wat betekend deze waarde?
7. Bekijk de help tekst van het commando find en bepaal op deze manier waar in de vector (x) de getallen met waarde groter dan 10 zitten.

x =[18 8 10 5 16 18 19 12 12 3 18 10 5 18 16 18 6 14 14 3]

1. Zoek nu de index waarde van 19 van de vector x.
2. Hoeveel elementen heeft de vector x? Gebruik een van de functies die je hierboven hebt geleerd.

## Antwoorden en uitwerkingen

1. Strcmp() geeft een 0 terug, oftewel een 0 is false, de variabele zijn niet gelijk aan elkaar.
2. Strcmp() geeft opnieuw een 0 terug. Dit komt omdat de S1 met een kleine letter is geschreven en S2 met een hoofdletter.
3. Strcmpi() geeft een 1 terug, oftewel het is nu true. Nu wordt er geen rekening gehouden met hoofdletters.
4. Dit is onjuist. In H5 wordt uitgelegd dat de eerste dimensie de rijen zijn en de tweede dimensie de kolommen. Ans = 2 3 betekend dat de matrix 2 rijen heeft en 3 kolommen.
5. Als het goed is, krijg je voor elke letter een cijfer, dus voor tuberositas krijg je 1 bij 11 eruit.
6. Hier komt een 1 uit. Dit betekend dat het waar(True) is dat de variabele leeg is.
7. Dit kan je doen met het statement find(X>10)
8. Dit kan je doen met het statement find(X==10), hierover leer je meer in de volgende reader!
9. Met de functie numel() vind je dat het antwoord 20 is.

# Combinatie opdrachten

Dit deel bevat een combinatie van opdrachten van al het bovenstaande wat je hebt geleerd.

## Opdracht

Je hebt op twee tijdstippen de versnellingen gemeten van rigide segmenten door middel van versnelling sensoren. Namelijk van de voet, het onderbeen en het bovenbeen. Je krijgt voor beide tijdstippen een vector, zoals je hieronder kan zien.

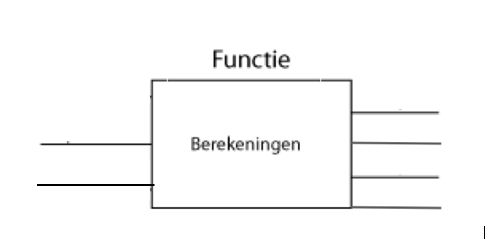
**Voer de versnellingsvectoren in in matlab:**

**versnelling1=[5 8 4]**

**versnelling2=[3 7 6]**

Je wilt van de twee versnellingsvectoren een matrix maken met de naam totaleVersnelling. Daarnaast wil je van de nieuwe aangemaakt matrix het aantal elementen mee geven als output, deze geef je mee met de naam aantalElementen.

**Maak een schets van hoe je functie eruit komt te zien:**

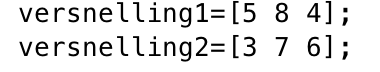


**Maak nu de functie aan met de naam vectorToMatrix.**

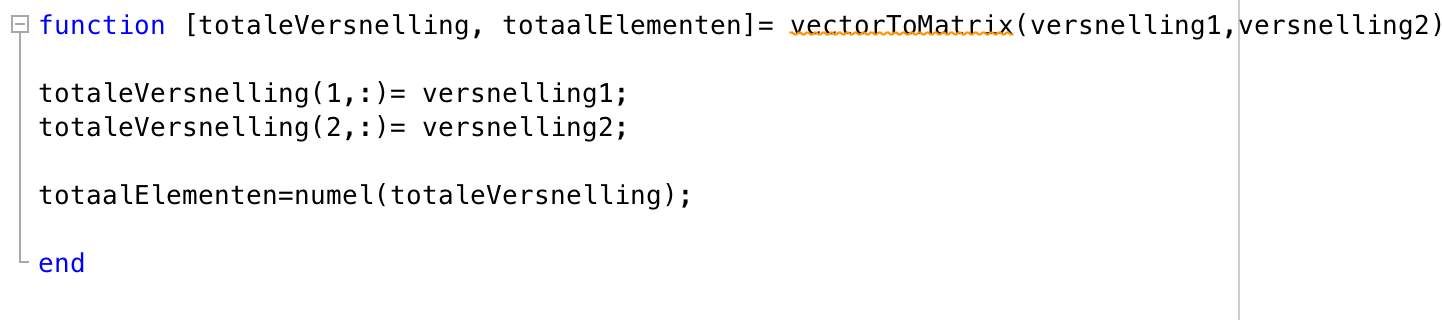
**Wanneer je de functie hebt aangemaakt vraag deze aan in je CommandWindow.**

## Uitwerking opdracht 8.1

Het invoeren van de vectoren in je Command Window:

****

Zo komt je functie eruit te zien:



Op de volgende manier vraag je je functie aan.

