

Inleiding Statistiek

Practicum 1

Op dit practicum herhalen we wat Matlab. Vervolgens illustreren we het schatten van een parameter en het toetsen van een hypothese met een klein simulatie experiment.

Het is de bedoeling dat je de onderstaande commando's zelf intypt, of met copy-paste overbrengt, en kijkt wat er gebeurt. Maak een verslag van de antwoorden op de opgaven die in de tekst staan.

1 Matlab

Start Matlab op. Je krijgt een window met een prompt `>>`. Hier kun je commando's typen. Zoals

```
>> 2+4
ans =
     6
```

of

```
>> 3*5
>> 2^3
>> sqrt(9)
>> sin(pi)           % Matlab kent pi
ans =
 1.2246e-16          % nou ja, eigenlijk 0!
```

Soms is het antwoord van een berekening WAAR of NIET WAAR

```
>> 2>4          % is 2 groter dan 4?  
ans =  
    0          % NIET WAAR = 0
```

andere voorbeelden:

```
>> 3==3         % let op de dubbele ==  
>> 3~=3         % ~= staat voor ongelijkheid  
>> (3>2)&(3>4)   % & staat voor het logische "EN"  
>> (3>2)|(3>4)   % | staat voor het logische "OF"
```

1.1 variabelen

Het volgende commando maakt een variabele a met waarde 3.162.

```
>> a = sqrt(10)
```

We kunnen natuurlijk ook rekenen met variabelen.

```
>> b = a+5
```

maakt een variabele b met waarde 8.162. Herhaal het bovenstaande commando, maar sluit af met een punt-komma. Wat valt je op?

Een logische variabele neemt de waarde WAAR of NIET WAAR aan. Als je met logische variabelen rekent, correspondeert WAAR met 1 en NIET WAAR met 0. Toch is een logische variabele niet hetzelfde als een gewone variabele met de waarde 0 of 1.

```

>> c = (3<4);
>> c
c =
     1
>> islogical(c)
ans =
     1           % c is een logische variabele
>> d=1;
>> islogical(d)
ans =
     0           % d is niet een logische variabele

```

Logische variabelen zijn bijzonder belangrijk, zoals spoedig duidelijk zal worden.

1.2 vectoren

Vectoren zijn het belangrijkste onderdeel van Matlab. Probeer de volgende voorbeelden een voor een uit

```

>> x = [1 3 2 6 0]
>> x = 3*ones(1,10)
>> x = 1:20
>> x = 1:3:20
>> x = 20:-1:1
>> x = []           % een lege vector

```

We kunnen ook rekenen met vectoren

```

>> x = 1:10;
>> y = 3*x - 12
y =
    -9    -6    -3     0     3     6     9    12    15    18
>> z = (y<0)
z =
     1     1     1     0     0     0     0     0     0     0

```

opgave 1.1 Het commando `x = unidrnd(10,1,10000)` produceert een rij van tienduizend toevallig gekozen getallen tussen 1 en 10. Voer het commando uit, en tel hoeveel vijven er in de rij voorkomen.

1.3 subscripting

Subscripting is het selecteren van elementen van een vector.

```
>> y(3)
>> y(3:5)
```

We kunnen ook selecteren met behulp van een Boolese vector. We selecteren alle elementen van `y` die groter zijn dan 0 door

```
>> y(y>0)
```

Dit is een zo'n belangrijk gebruik van logische variabelen waar we het eerder over hadden.

We kunnen natuurlijk ook rekenen met vectoren

```
>> x = 1:10;
>> y = 1:2:20;
>> x+y
>> x.*y      % puntsgewijze vermenigvuldiging
>> x*y'      % inproduct (waarom werkt x*y niet?)
```

1.4 matrices

We maken een matrix `A` (probeer de onderstaande commando's een voor een uit)

```
>> A = ones(2,3)
```

```

>> A = eye(4)
>> A = [1 2 3; 4 5 6; 7 8 9; 10 11 12] % punt-komma's scheiden de rijen.
>> A = A' % transpositie
>> A = reshape(A,6,2)
>> A = repmat(A,2,3)

```

We kunnen natuurlijk elementen van de matrix selecteren

```

>> A(1,2)
>> A(1:2,2:3)
>> A(2,:) % selecteer de tweede rij
>> A(:,3) % selecteer de derde kolom

```

opgave 1.2 Beschouw het volgende lineaire probleem

$$3x_1 + 2x_2 + x_3 = 39$$

$$2x_1 + 3x_2 + x_3 = 34$$

$$x_1 + 2x_2 + 3x_3 = 26$$

Los dit probleem op met Matlab. Gebruik het commando `inv()` om de matrix te inverteren.

1.5 functies

We hebben al een aantal functies gebruikt: `sqrt`, `sin` en `inv`. Matlab heeft bijzonder veel ingebouwde functies. De uitleg van een functie is de vinden door

```

>> help inv

```

Hoe weet je nou of Matlab een ingebouwde functie heeft voor (bijvoorbeeld) de cosinus? De beste methode is proberen te raden

```
>> cosine(pi)
??? Undefined command/function 'cosine'.
>> cos(pi)
ans =
    -1          % Aha!
```

Je kunt ook nog altijd Google “Matlab cosine” proberen.

Sommige functies hebben twee of meer “outputs”.

```
>> x = [2 7 3 4 3 1 4];
>> sort(x)                % de gesorteerde rij
>> [y i]=sort(x)         % de gesorteerde rij met indices
>> x(i)                  % de gesorteerde rij (hoezo?!)
```

opgave 1.3 Gebruik Matlab om de waarde tot op 3 decimalen te benaderen waar de functie $\sin(x)$ zijn maximum bereikt op het interval $[0, 2\pi]$.

1.6 plotten

Je kunt de functies `plot()` gebruiken om plaatjes te maken.

```
>> x = 0:0.1:2*pi;
>> plot(x,sin(x));
```

of

```
> plot(x,sin(x), 'r*')
```

Bekijk `help plot` voor alle mogelijkheden. Het commando `hold on` zorgt ervoor dat je iets aan een bestaand plaatje kunt toevoegen. Na het commando `hold off` wordt alles eerst gewist.

opgave 1.4 Plot de functie $f(x) = x^2$ op het interval $[-5, 5]$. Breng vervolgens de functie $g(x) = |5x|$ aan in je grafiek.

1.7 scripting

Een script is een lijstje met commando's. Hier is een voorbeeld van een scriptje dat de kwadraten van de getallen 1 t/m 10 uitrekent. We gebruiken een for-lus.

```
for i = 1:10
    disp([i i^2]);
end;
```

Hier is een script dat de if-constructie demonstreert.

```
for i = 1:10
    disp([i i^2]);
    if (i == 5)
        disp('halverwege');
    end;
end; % sluit de if-constructie af
% sluit de for-lus af
```

opgave 1.5 Een for-lus is vaak niet nodig, en kost dan nodeloos veel rekestijd. Bepaal de kwadraten van de getallen 1 t/m 10 met behulp van een enkel commando.

Als je zelf een script moet schrijven, is het handig om een editor te gebruiken. Geef het bestand een naam met een .m extensie, bijvoorbeeld `kwadraten.m`. Als je nu bij de Matlab prompt het commando `kwadraten` geeft, wordt je script uitgevoerd.

opgave 1.6 In de Fibonacci rij is ieder getal gelijk aan de som van de twee voorafgaande getallen: 1, 1, 2, 3, 5, 8, 13, 21, ... Gebruik Matlab om de eerste 20 Fibonacci getallen te bepalen.

2 Schatten

Stel we hebben een munt met onbekende kop-kans p . We gooien n keer met deze munt, en noemen het aantal koppen X . Nu is X een stochastische grootte met de binomiale kansverdeling met

parameters n en (de onbekende) p . Er geldt

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, \dots, n.$$

De meest voor de hand liggende schatter van p is natuurlijk

$$\hat{p} = X/n.$$

opgave 2.1 Hoezo ligt deze schatter voor de hand?

Merk op dat \hat{p} een stochastische grootheid is met een eigen kansverdeling die weer van de onbekende p afhangt!

$$P(\hat{p} = k/n) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, \dots, n.$$

De verwachting van \hat{p} is

$$E(\hat{p}) = E\left(\frac{X}{n}\right) = \frac{1}{n}EX = \frac{np}{n} = p,$$

en de variantie van \hat{p} is

$$\text{Var}(\hat{p}) = \text{Var}\left(\frac{X}{n}\right) = \frac{1}{n^2}\text{Var}(X) = \frac{np(1-p)}{n^2} = \frac{p(1-p)}{n}.$$

Hoe groter n , hoe kleiner deze variantie. Dus: hoe vaker we met de munt gooien hoe nauwkeuriger we p kunnen schatten.

opgave 2.2 Doe 1000 trekkingen uit de binomiale verdeling met parameters $n = 10$ en $p = 0.6$ en maak een histogram van de 1000 schatters.

```
n = 10;  
p = 0.6;  
X = binornd(n,p,1,1000);  
hist(X/n,0:1/n:1);
```

Kies $n = 3, 10, 20, 100, 10000$. Begrijp je het?

opgave 2.3 Voer de onderstaande commando's een aantal keer uit.

```
p = 0.6;
X = binornd(1,p,1,1000);
M = cumsum(X)./(1:1000);
plot(M)
```

Welke Stelling uit Kansrekening en Statistiek 1 zie je hier in actie? Wat heeft dat met schatten te maken?

3 Toetsen

Gooi met een munt met kop-kans p . Als kop valt krijg je een euro, als munt valt ben je een euro kwijt. De meeste mensen zullen dit spel alleen willen spelen als $p > 1/2$. Stel nu dat je eerst 10 keer met de munt mag gooien om vast te stellen of p groter dan $1/2$ is.

Zij X het aantal koppen in de eerste 10 worpen. Het ligt voor de hand om de volgende regel te hanteren:

Kies een zekere K . Als $X > K$, dan zijn we bereid het spel te spelen.

Het is nu nog niet duidelijk hoe we K moeten kiezen, maar daar komen we later op terug.

opgave 3.1 Waarom ligt deze regel voor de hand?

In het jargon van de statistiek hebben we de volgende situatie. Stel X is verdeeld volgens de Binomiale verdeling met parameters $n = 10$ en onbekende p . We toetsen de *nulhypothese*

$$H_0 : p \leq \frac{1}{2}$$

versus het *alternatief*

$$A : p > \frac{1}{2}.$$

We besluiten de nulhypothese te *verwerpen* als $X > K$. We noemen X de *toetsingsgrootte* en K de *kritieke waarde*

De nulhypothese is de “grondtoestand”. We verwerpen de nulhypothese pas als we echt genoeg reden om eraan te twijfelen. Als het experiment geen duidelijke uitslag oplevert, dan krijgt de nulhypothese het voordeel van de twijfel.

Nog een keer voor alle duidelijkheid: Als we de nulhypothese niet kunnen verwerpen, betekent dat *niet* dat de nulhypothese waar is. Het betekent alleen maar dat we (nog) niet genoeg reden hebben om aan de nulhypothese te twijfelen.

opgave 3.2 Waarom toetsen we hier niet de nulhypothese $H_0 : p > 1/2$ versus het alternatief $A : p \leq 1/2$?

Het volgende script genereert 1000 trekkingen uit de binomiaal($n = 10, p = 0.6$) verdeling. In dit voorbeeld is de nulhypothese dus **niet** waar. Als het aantal koppen groter dan $K = 5$ is, verwerpen we de nulhypothese.

```
p = 0.6;
K = 5;
X = binornd(10,p,1,1000);
verwerp = (X>K);
verwerp = num2str(sum(verwerp));
disp(['We verwerpen de nulhypothese in ' verwerp ' uit 1000 keer.']);
```

opgave 3.3 Voer het script uit. De nulhypothese is hier niet waar. Wordt deze dan ook altijd verworpen? Probeer K zo in te stellen dat de nulhypothese altijd wordt verworpen.

opgave 3.4 Verander het script zodat $p = 0.5$ en $K = 5$. Nu is de nulhypothese wèl waar. Wordt deze dan ook nooit verworpen? Probeer K zo in te stellen dat de nulhypothese nooit wordt verworpen.

Zoals je ziet, kun je twee soorten fouten maken bij het toetsen van een hypothese:

1. fout van de eerste soort: nulhypothese is waar, maar wordt toch verworpen.
2. fout van de tweede soort: nulhypothese is niet waar, maar wordt niet verworpen.

Door K aan te passen kunnen we de kans op een fout van de ene soort klein krijgen. Helaas gaat daardoor de kans op een fout van de andere soort omhoog.

De algemeen geaccepteerde oplossing voor dit dilemma is van te voren vast te stellen hoe groot de kans op een fout van de *eerste* soort mag zijn. Dit heet het *significantieniveau* α . Met andere woorden, het significantie niveau is de kans dat de nulhypothese wordt verworpen, terwijl deze juist is. Gebruikelijke waarden voor α zijn 1% of 5%.

opgave 3.5 Stel $\alpha=5\%$. Gebruik het simulatie script om K zo te kiezen dat de nulhypothese in ongeveer 5% van de gevallen wordt verworpen als $p = 1/2$.

Bij de vorige opgave vond je dat $K = 7$ een toets oplevert die in ongeveer 5% van de gevallen de nulhypothese verwerpt, wanneer deze in feite juist is. Met andere woorden $P(X > 7) \approx 0.05$ als de kop-kans $p = 1/2$.

We hebben de functie `binornd` gebruikt voor het trekken van een steekproef uit de binomiale verdeling. De functies `binopdf`, `binocdf` en `binoinv` corresponderen met de kansmassafunctie, de verdelingsfunctie en de kwantiel functie van de binomiale verdeling. Zie de `help` voor uitleg.

opgave 3.6 Gebruik Matlab om $P(X > 7)$ te bepalen, onder de aanname dat $p = 1/2$. Dit is het exacte significantie niveau van de toets.

De zogeheten P -waarde lijkt erg op het significantie niveau, maar mag daarmee niet verward worden. De P -waarde is de kans, onder de nulhypothese, dat de toetsingsgrootte een waarde aanneemt die tenminste zo onwaarschijnlijk is onder de nulhypothese als de geobserveerde waarde.

Bijvoorbeeld: Stel dat we 9 koppen in 10 worpen waarnemen. De P -waarde is $P(X \geq 9) = 0.011$. De P -waarde geeft aan hoe onwaarschijnlijk de observatie is, als de nulhypothese waar zou zijn. Een kleine P -waarde betekent dus dat de observatie erg onwaarschijnlijk is onder de nulhypothese—reden om deze te verwerpen. Er geldt:

De nulhypothese wordt verworpen dan en slechts dan als de P -waarde van de observatie kleiner is dan het significantie niveau.

opgave 3.7 De P -waarde is de kans is dat de nulhypothese waar is. Is deze bewering juist of niet?