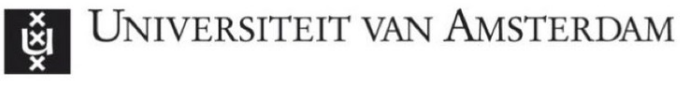
Data Analyse en Statistiek

Bachelor Natuur- en Sterrenkunde

studiejaar 2021-2022



dr H.L. Snoek en M.F.L. Schut MSc
2022.v1

Inhoud

MODULE I

Statistische data analyse is een belangrijk onderdeel in vele werk- en onderzoeksvelden. Als student, en later wellicht ook als wetenschapper, zul je te maken krijgen met het verzamelen en interpreteren van data bij het het prakticum, bij het doen van onderzoek, of juist bij het begrijpen van de interpretatie van andermans resultaten.

- Wanneer kun je zeggen dat een hypothese moet worden verworpen of bewijs je juist dat deze correct is?
- Hoe moet je inschatten of je meetnauwkeurigheid goed genoeg is? Wanneer heb je eigenlijk genoeg data verzameld?
- Hoe kun je een experiment zo ontwerpen dat je een hypothese kunt onderzoeken.
- Hoe kom je erachter wat jouw hypothese toetsbaar maakt - in welke observabele onderscheidt zij zich voldoende van andere hypotheses?

Alle kennis die we tot nu toe hebben over de Natuur- en Sterrenkunde is tot stand gekomen met het uitvoeren van experimenten en het analyseren van de uitkomsten hiervan. Voor het bestuderen van Natuurkundige en Sterrenkundige theorieën is niet persé kennis nodig van de statistiek en van data analyse technieken. Voor het uitvoeren van wetenschap, het vinden van bewijzen voor nieuwe theorieën is kennis hiervan echter essentieel.

Bij het presenteren van onderzoeksresultaten is het belangrijk om helder uit te kunnen leggen hoe het onderzoek precies is uitgevoerd, hoe de metingen zijn verkregen en wat de resultaten zijn. Vaak maken we hierbij gebruik van histogrammen, grafieken en tabellen. Om een hypothese te toetsen moeten we metingen ook kunnen interpreteren. Hiervoor zijn verschillende methodes, bijvoorbeeld kunnen we de data proberen te 'fitten' met een functie, een wiskundige vergelijking. Bij al deze methodes speelt statistiek een belangrijke rol.

In deze cursus zullen we vaardigheden gaan leren voor data analyse en statistiek. We beginnen deze week met een aantal basisbegrippen (Hfdst. 1) in de beschrijvende statistiek. We gaan kijken naar het gemiddelde, variantie, de standaardafwijking, en coëfficiënt van variantie. We leren over hoe we meetresultaten moeten presenteren, het gebruik van de wetenschappelijke notatie (Hfdst. 2) en hoe we ze kunnen visualiseren (Hfdst. 3). We gaan in op het begrip meetonzekerheid (Hfdst. 4). Ook maken we een begin met kansrekening (Hfdst. 5) en kansdichtheidsverdelingen (Hfdst. 6). Niet elk van deze onderwerpen is even moeilijk. Let goed op dat je genoeg tijd overhoudt om de introductie van de kanstheorie te bestuderen.

MODULE I Hoofdstuk 1

Basisbegrippen in de statistiek

1.1 Datasets beschrijven

Als we een set metingen (data) hebben verzameld kunnen we deze op verschillende manieren gebruiken. Vaak willen we bepaalde kenmerken van de dataset weten. Stel we hebben een dataset met de temperatuur op elk van de 37 meetpunten van het KNMI in Nederland in de afgelopen twintig jaar. Het is dan niet zo inzichtelijk om dit aan medewetenschappers te presenteren d.m.v. een enorme tabel (elke 10 minuten wordt een meting gedaan door de weerstations) met de mededeling 'dit was de temperatuur in de afgelopen twintig jaar'. Uit deze dataset kun je natuurlijk een enorme hoeveelheid informatie halen. Bijvoorbeeld wat is de koudste temperatuur die in de afgelopen 20 jaar in Nederland is gemeten. Maar ook: Wat is de gemiddelde temperatuur in de maand Juli. Of: Hoeveel kouder zijn de winters in het binnenland ten opzichte van de regio's aan de kust.

In de secties hieronder behandelen we verschillende veelvoorkomende definities van kenmerken van data.

1.1.1 Populatie en steekproef

Voordat we het gaan hebben over de kenmerken van data is het belangrijk om te kijken naar de data zelf. Waar komt die vandaan? We maken hierbij onderscheid tussen de **populatie** en een **steekproef**.

Een **populatie** bestaat uit alle personen/dieren/objecten binnen de groep waarin we geïnteresseerd zijn. Dit zouden bijvoorbeeld *alle* mensen in Nederland kunnen zijn tussen de 30 en 40 jaar, of *alle* lieveheersbeestjes die in Noorwegen leven. Nu is het zo dat het vaak lastig is om van *alle* personen/dieren/objecten (hierna uniform aangeduid met 'elementen') van een groep gegevens te verzamelen. Het kost bijvoorbeeld erg veel tijd (en geld) om data te verzamelen over alle personen tussen de 30 en 40 jaar in Nederland (of om alle lieveheersbeestjes in Noorwegen te vangen). Het is dan veel makkelijker om data over een deel van deze groep te verzamelen om zo toch iets te kunnen zeggen over de gehele doelgroep. Zo zouden we bijvoorbeeld data kunnen verzamelen van een willekeurige selectie

van 200 personen in Nederland tussen de 30 en 40 jaar. Dit wordt een *steekproef* genoemd, de deelgroep wordt in het Engels vaak aangeduid met een *sample*. Een steekproef is dus een gedeelte van de populatie. Vaak is het trouwens zelfs helemaal niet mogelijk om de hele populatie te meten. Denk bijvoorbeeld maar eens aan de gemiddelde massa van een ster. Dan zouden we deze meting moeten verrichten voor alle sterren in het universum.

We maken onderscheid in de namen en de notatie van de kenmerken van data. Kenmerken van meetgegevens (data) van een populatie noemen we **parameters**, kenmerken van steekproeven noemen we **statistieken**. Het is belangrijk om onderscheid te maken. Als we bijvoorbeeld de gemiddelde leeftijd willen weten van alle eerstejaars Natuur- en Sterrenkunde studenten in Amsterdam dan maakt het uit of we de gegevens hebben verzameld van alle eerstejaars of dat we de gemiddelde leeftijd inschatten door de gegevens te noteren van de studenten uit je eigen werkgroep. In het eerste geval hebben we gegevens van de hele populatie en spreken we van een parameter en weten we de uitkomst exact. In het tweede geval hebben we een steekproef gedaan van een selectie van de eerstejaars, we spreken dan van een statistiek en op deze statistiek komt een onzekerheid. We hebben immers niet alle informatie van de populatie en het kan zijn dat het gemiddelde van de steekproef afwijkt van het gemiddelde van de gehele populatie. Het is dus belangrijk om je te realiseren of je de gegevens bekijkt van een steekproef of een populatie als je de resultaten interpreteert.

Als je een steekproef neemt is het belangrijk om op twee dingen goed te letten: de grootte van de steekproef en hoe representatief deze is. Je kunt je voorstellen dat als we de lengte van drie mensen in Nederland meten, we nog niet zoveel kunnen zeggen over de lengte van de gehele populatie die bestaat uit alle mensen in Nederland. Als we de lengte van 1000 mensen zouden meten dan krijgen we al een beter beeld van de verdeling van lichaamslengte in Nederland, en kiezen we 100.000 mensen dan krijgen we een nog veel beter beeld van de verdeling. Hoe groter de steekproef, hoe nauwkeuriger de statistiek is die we willen weten. (We zeggen dan vaak dat we *meer statistiek* hebben.)

Ook is het belangrijk hoe we de steekproef nemen. Als we bijvoorbeeld de lengte gegevens van 1000 mensen nemen dan krijgen we een vertekend beeld als we hiervoor de leden van de Nederlandse Basketbal vereniging uitnodigen, of de gegevens van 1000 kleuters hiervoor gebruiken. Je moet dus altijd goed kijken of de steekproef de je neemt wel representatief is voor de hele groep.

1.2 Veel gebruikte parameters en statistieken

1.2.1 Het gemiddelde

Het gemiddelde van een dataset geeft een maat voor het centrum van de waarden die de dataset aanneemt. We onderscheiden het populatiegemiddelde (parameter) en het steekproefgemiddelde (statistiek). Hoe groter de steekproef hoe meer het gemiddelde van de steekproef overeenkomt met het populatiegemiddelde.

Het gemiddelde kun je berekenen door alle waardes in de dataset te sommeren en te delen door de grootte van de dataset. We maken onderscheid in de notatie voor het gemiddelde van een steekproef en die van het populatiegemiddelde.

Het steekproef gemiddelde $\bar{x}$ (x-streep of in het Engels: x-bar) van een dataset is de som van de waarden $x_1, \ldots, x_n$ in de set gedeeld door het aantal datapunten in de steekproef: n:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1.1}$$

Het steekproef gemiddelde wordt zo vaak gebruikt dat dit veelal wordt aangeduid als 'het gemiddelde'. Voor het gemiddelde wordt ook vaak de 'vishaak-notatie' gebruikt: $< x >$.

Het populatiegemiddelde wordt als volgt genoteerd:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1.2}$$

Hierbij is N het aantal elementen in de populatie, en zijn $x_i, \ldots, x_N$ de waardes van de grootheid in de populatie. Let op dat voor de steekproefgrootte n wordt gebruikt en voor de populatiegrootte N. Een andere veel gebruikte notatie voor het populatiegemiddelde is $E(x)$ waar de E van het Engelse woord *expectation* komt. Ook kun je een subscript toevoegen om aan te geven van welke grootheid je het gemiddelde berekent, bijvoorbeeld hier μ_x.

Je ziet dat het steekproef gemiddelde erg lijkt op de uitdrukking voor het populatiegemiddelde. Het verschil is dat het steekproefgemiddelde niet persé gelijk is aan de verwachtingswaarde van de populatie. Het is wel zo dat, hoe beter de steekproef overeenkomt met de populatie, des te dichter komt het steekproef gemiddelde bij de verwachtingswaarde van de populatie. Met behulp van een goed uitgevoerde steekproef kan het statistische gedrag van een populatie dus benaderd worden.

Voorbeeld Stel je voor dat we de volgende steekproef hebben:

$$X = \{-5, 1, 14, 12, 0\}.$$

De gemiddelde waarde voor de data is nu dus

$$\bar{x} = \frac{1}{5} \cdot (-5 + 1 + 14 + 12 + 0) = \frac{1}{5} \cdot 22 = 4.4$$

1.2.2 De mediaan

De mediaan is een maat voor het midden van de elementen in een gesorteerde dataset of verdeling. De mediaan is zo gedefinieerd dat je precies 50% kans hebt om een waarde te

vinden die lager is dan de mediaan en 50% kans om een waarde te vinden die hoger is dan de mediaan.

Als we alle datapunten in een dataset sorteren van lage naar hoge waarde, dan is de mediaan de waarde van het element in het midden van de set. Is er sprake van een even aantal elementen dan is de mediaan de gemiddelde waarde van de twee elementen in het midden van de set.

Voorbeeld Stel dat we de volgende dataset hebben:

$$X = \{13, 11, 10, 14, 12, 9\}.$$

Het eerste wat we moeten doen om de mediaan te vinden is de dataset sorteren:

$$\{9, 10, 11, 12, 13, 14\}.$$

We hebben een dataset met een even aantal datapunten, de mediaan ligt hier dus tussen twee waardes in.

$$\text{de mediaan is : } \quad \frac{(11 + 12)}{2} = 11.5.$$

De mediaan en het gemiddelde *kunnen* dezelfde waarde hebben, maar dat hoeft niet zo te zijn. Voor het voorbeeld hierboven is dat wel het geval (reken maar na). Maar voor de dataset uit het voorbeeld voor het berekenen van het gemiddelde is dit niet zo. Kijk maar!

Voorbeeld We bekijken de steekproef

$$X = \{-5, 1, 14, 12, 0\}.$$

Het gemiddelde was berekend op 4.4. We gaan nu kijken waar de mediaan ligt. Eerst sorteren we de dataset:

$$\{-5, 0, 1, 12, 14\}.$$

Dit is een oneven dataset en de mediaan ligt dus op de middelste waarde van de gesorteerde dataset: 1.

Voor symmetrische datasets zijn het gemiddelde en de mediaan altijd gelijk aan elkaar, voor asymmetrische datasets is dit niet het geval. Bij een symmetrische dataset is de data precies gespiegeld rond het gemiddelde. Dit is makkelijker uit te leggen aan de hand van datadistributies. We komen hier later op terug.

1.2.3 De modus

De modus van een dataset is de waarde die met de grootste frequentie in de dataset voorkomt. Hebben we bijvoorbeeld de dataset

$$2, 2, 3, 4, 7, 7, 7, 9 \tag{1.3}$$

dan komen de 3, de 4 en de 9 elk één keer voor, het getal 2 komt twee keer voor en het getal 7 komt drie keer voor. Het meest voorkomende getal is dus de 7 en dit is de modus van de dataset. Als een dataset één modus heeft dan wordt deze *unimodaal* genoemd.

Het komt ook voor dat er twee of meer getallen zijn die vaker voorkomen dan andere waardes. Een dataset met twee getallen als modus wordt ook wel *bimodaal* genoemd, een dataset met meer dan twee getallen als modus wordt *multimodaal* genoemd.

Een voorbeeld van een bimodale dataset is:

$$1, 2, 3, 3, 4, 4, 4, 5, 6, 11, 11, 11, 15 \tag{1.4}$$

zowel het getal 4 als het getal 11 komen drie keer voor in de set. De set is dus bimodaal met modus 4 en modus 11.

Bij sommige soorten dataverdelingen is het gebruikelijker om over de modus te praten dan over het gemiddelde of de mediaan. Een voorbeeld hiervan is de Landau distributie die een slecht gedefinieerd gemiddelde of mediaan kent door een lange staart in de distributie.

Voor unimodale symmetrische distributies ligt het gemiddelde, de mediaan en de modus precies op dezelfde plek.

1.3 Spreiding van data

De spreiding geeft een beeld van de mate waarin datapunten in een set verspreid zijn. Er zijn verschillende maten om de spreiding van een dataset mee aan te geven. Hieronder zullen we **de spreidingsbreedte** (ook wel de *range*), **de variantie**, **coëfficiënt van variantie** en **de standaardafwijking** (ook wel de *standaarddeviatie*) bespreken.

1.3.1 Spreidingsbreedte (range)

De range is de afstand tussen de hoogste en de laagste waarde in een dataset. Hebben we bijvoorbeeld de dataset

$$50, 70, 72, 76, 76, 80, 120 \tag{1.5}$$

dan is de range van deze dataset gelijk aan $120 - 50 = 70$.

De range geeft dus aan hoe breed de dataset in totaliteit is. De range is niet altijd een handige maat voor de spreiding van een dataset. Zo zouden we bijvoorbeeld de volgende dataset kunnen hebben:

$$1, 2, 3, 4, 5, 5, 5, 6, 6, 6, 7, 7, 10 \tag{1.6}$$

De range is in dit geval $10 - 1 = 9$. Maar stel dat we een foutieve meting doen (of we maken een typefout in het overnemen van de data), en we hebben de volgende dataset:

$$1, 2, 3, 4, 5, 5, 5, 6, 6, 6, 7, 7, 10, 30 \tag{1.7}$$

De range wordt nu $30 - 1 = 29$. Dus onder invloed van één foutief datapunt geeft de range nu een veel grotere mate van spreiding aan.

1.3.2 Standaardafwijking en variantie

De standaardafwijking geeft aan in welke mate de data verspreid is rondom het gemiddelde van de dataset. Dit geeft met name ook een maat voor de spreiding van de datapunten onderling. Hoe groter de standaardafwijking des te groter is de spreiding tussen de afzonderlijke punten. De standaardafwijking voor de populatie wordt aangeduid met σ, voor een steekproef noteren we dit met s.

De variantie, *var*, is direct gerelateerd aan de standaardafwijking, namelijk de variantie is gelijk aan de standaardafwijking in het kwadraat. Voor de populatie geldt dus var $= \sigma^2$. De variantie van een steekproef noteren we met s^2.

De variantie en standaardafwijking van een populatie kunnen worden berekend met de volgende formule:

$$var = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{1.8}$$

of in het geval van de steekproef:

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{1.9}$$

Let op dat de eenheid van de variantie het kwadraat is van de eenheid van x. In het geval dat je bijvoorbeeld lengtes van luciferstokjes hebt opgemeten, dan zullen de waardes in cm zijn genoteerd. De variantie heeft dan de eenheid cm^2. Dat kan soms best onhandig zijn, vandaar dat we vaker de standaardafwijking gebruiken. De standaardafwijking heeft altijd dezelfde eenheid als de originele elementen van de dataset.

Je kan wiskundig aantonen dat je voor het berekenen van de variantie ook de volgende formule mag gebruiken:

$$s_x^2 = \overline{x^2} - \overline{x}^2. \tag{1.10}$$

Soms is deze formule makkelijker in het gebruik.

1.3.3 Variatiecoëfficiënt

De variatiecoëfficiënt wordt ook wel de relatieve standaardafwijking genoemd. De coëfficiënt van variatie geeft, net zoals de standaardafwijking en de variantie, een maat voor de spreiding van de populatie of dataset.

De variatiecoëfficiënt wordt gegeven door de verhouding tussen de standaardafwijking en het gemiddelde. Voor een populatie is de coëfficiënt van variantie c_v dan:

$$c_v = \frac{\sigma}{\mu}. \tag{1.11}$$

Met σ de standaardafwijking van de populatie en μ het populatiegemiddelde.

De steekproef variantie $\hat{c}_v$ wordt gegeven door:

$$\hat{c}_v = \frac{s}{\bar{x}}. \tag{1.12}$$

Met s de standaardafwijking van de steekproef en $\bar{x}$ het steekproef gemiddelde.

Het verschil met de variantie en de standaardafwijking is dat de variatiecoëfficiënt dimensieloos is. Dit is bijvoorbeeld handig als er meerdere datasets vergeleken moeten worden die verschillende eenheden hebben. Ook als de gemiddelde waarden van verschillende datasets erg uiteen liggen is het beter om de variatiecoëfficiënt te gebruiken i.p.v. de standaardafwijking.

Een nadeel van het gebruik van de variatiecoëfficiënt is dat er gedeeld wordt door het gemiddelde. Als dit gemiddelde een heel kleine waarde heeft, dicht bij nul, dan is de variatiecoëfficiënt slecht gedefinieerd.

1.4 Samenvatting

kenmerk	populatie (*parameter*)	steekproef (*statistiek*)
grootte	N	n
gemiddelde	$\mu = \frac{1}{N}\sum_i^N x_i$	$\bar{x} = \frac{1}{n}\sum_i^n x_i$
standaardafwijking	$\sigma = \sqrt{\frac{1}{N}\sum_i^N (x_i - \mu)^2}$	$s = \sqrt{\frac{1}{n}\sum_i^n (x_i - \bar{x})^2}$
variantie	$var = \sigma^2$	s^2
variatiecoëfficiënt	$c_v = \frac{\sigma}{\mu}$	$\hat{c}_v = \frac{s}{\bar{x}}$

1.5 Voorbeelden

We berekenen de eigenschappen van een aantal datasets als voorbeeld.

Voorbeeld: Een populatie
We hebben de volgende dataset van een populatie:

$$Y = \{285, -20, 31, 60, 12, 53, 133\}.$$

We bepalen nu hieronder de verschillende *parameters* die horen bij deze populatie.

- De grootte is dus N = 7.
- Om de mediaan te bepalen sorteren we eerst de datapunten van klein naar groot:

  $$\{-20, 12, 31, 53, 60, 133, 285\}.$$

  Het is een even aantal datapunten en de mediaan ligt tussen 53 en 60 in. Dit komt dan uit op 56.5.
- De spreidingsbreedte:

  $$285 - (-20) = 305.$$

- Het gemiddelde:

  $$\mu_Y = \frac{1}{7} \cdot (285 - 20 + 31 + 60 + 12 + 53 + 133) = 79.1$$

- De standaardafwijking is:
  $\sigma_Y^2 = \frac{1}{7} \cdot \Big[(285 - 79.1)^2 + (-20 - 79.1)^2$
  $+ (31 - 79.1)^2 + (60 - 79.1)^2 + (12 - 79.1)^2$
  $+ (53 - 79.1)^2 + (133 - 79.1)^2\Big] = 8997.6$
  geeft $\sigma_Y = 94.9$.
- De variantie $var_Y = 8997.6$.
- De variatiecoëfficiënt $c_v = 1.20$.

Voorbeeld: Een steekproef
Stel we hebben een steekproef gedaan van de lengte van eerstejaars studenten. De volgende dataset is hiervoor verzameld:

$$L = \{1.90 \text{ m}; 1.72 \text{ m}; 1.61 \text{ m}; 1.84 \text{ m}; 1.79 \text{ m}\}.$$

Hieronder bepalen we de *statistieken* voor deze steekproef.

- De grootte van de steekproef: $n = 5$.
- De spreidingsbreedte is 1.90 m-1.61 m = 39 cm.
- De mediaan ligt in het midden van de gesorteerde dataset. Dit is 1.79 m.
- Het gemiddelde $\bar{L} = 1.77$ m.
- De variantie is:
$$\begin{aligned} s^2 =& \frac{1}{5} \cdot \Big[(1.90 - 1.77)^2 + (1.72 - 1.77)^2 \\ &+ (1.61 - 1.77)^2 + (1.84 - 1.77)^2 + (1.79 - 1.77)^2\Big] \\ =& \ 0.0100 \text{ m}^2 \end{aligned}$$
- De standaardafwijking is $s = 0.10$ m.
- De variatiecoëfficiënt is $\hat{c_v} = 0.0057$d

MODULE I Hoofdstuk 2

Het correct noteren van resultaten

Voordat we verder gaan is het belangrijk om even in te gaan in het onderwerp significantie en de wetenschappelijke notatie. Dit gaat over hoe we een resultaat noteren. Het is goed om hier even bij stil te staan.

Stel dat we een lang meetlint hebben met een millimeter verdeling. We meten de lengte van een lange plank op. We noteren 253.3 cm. We hebben de plank goed kunnen opmeten en er staat een millimeter verdeling op het meetlint. We hebben de meting tot op de millimeter nauwkeurig gedaan. Stel nu dat we met hetzelfde meetlint de hoogte van een struik opmeten. Is het dan oké voor deze gemeten hoogte ook de millimeters te noteren? Het opmeten zal waarschijnlijk wel lastig worden. Waar begint bijvoorbeeld de stam van het struikje? De aarde zal wel niet helemaal glad zijn. En lukt het wel om loodrecht op de aarde te meten?

De hoeveelheid getallen die we noteren zegt vaak iets over nauwkeurig we denken het resultaat te weten. Meer hierover komt later terug in het stukje over meetonzekerheid (Hfdst. 4).

Een ander voorbeeld is als we de gemiddelde lengte van drie stokken willen uitrekenen. De stokken zijn 45, 50 en 54 cm lang. We rekenen het gemiddelde uit met onze rekenmachine en we kopiëren het resultaat: 49.66666666 cm. Het lijkt nu of we het resultaat super-nauwkeurig weten terwijl we voor de stoklengtes alleen de centimeters hebben genoteerd. Dat klopt natuurlijk niet!

Voor het noteren van wetenschappelijke resultaten maken we in dit vak nu afspraken die jullie ook zuleln toepassen in de overige bachelorvakken. Hetzelfde geldt voor het visualiseren van data, daarvoor maken we in het volgende hoofdstuk afspraken. Het is goed om je te realiseren dat er soms wat kleine verschillen kunnen zijn in de afspraken omtrent de visualisatie en de notatie. Als je later in je bachelor een project gaat doen kan het zijn dat de consensus over het presenteren van resultaten net iets anders ligt. Voor nu spreken we de regels af zoals die hieronder volgen.

We beginnen met het uitleggen van wat begrippen die we nodig hebben om de afspraken uit te kunnen leggen.

2.1 Significantie en precisie

Meetwaardes moeten met de juiste **significantie** worden genoteerd. De *significantie* is de nauwkeurigheid waarmee een getal/waarde wordt weergegeven. Vaak wordt gedacht dat het aantal decimale cijfers de nauwkeurigheid aangeeft, maar dit is technisch gezien de *precisie* waarmee de (meet)waarde wordt aangegeven. De nauwkeurigheid (significantie) van een getal zegt welke cijfers in het getal er iets toe doen. Cijfers zonder betekenis tellen we niet mee bij de significantie.

Om de significantie en de precisie te bepalen is het belangrijk om op de nullen te letten en de positie van de punt.

Voor de **significantie** geldt:

- Nullen aan de linkerkant doen niet mee. Het getal 0.0056 heeft bijvoorbeeld twee significante cijfers.
- Nullen aan de linkerkant voorafgegaan door een getal doen wel mee met de significantie. Het getal 100.004 heeft zes significante cijfers.
- Nullen aan de rechterkant doen wel mee met de significantie. Zo heeft 10.34000 zeven significante cijfers.
- Een uitzondering op de tweede regel zijn getallen zoals 300, 4000, 570 etc. Deze getallen zijn weergeven zonder decimalen waardoor het onduidelijk is of daadwerkelijk de waarde van 300 respectievelijk 4000 en 570 is gemeten, of dat dit met een hogere of juist lagere precisie is gebeurd. De afspraak is dat als een getal op deze manier wordt weergegeven met nullen rechts, deze nullen niet meedoen met de nauwkeurigheid. De getallen 300 en 4000 hebben bijvoorbeeld allebei een significantie van 1. Het getal 570 heeft twee significante cijfers. Om deze getallen met een ander aantal significante cijfers weer te geven wordt vaak de *wetenschappelijke notatie* gebruikt. Hier komen we later op terug.

De **precisie** van een getal wordt gegeven door het aantal cijfers achter de punt.

Een aantal voorbeelden

- Het getal 7.134 heeft in totaal 4 significante cijfers, de precisie is 3.
- Het getal 0.576 heeft 3 significante cijfers, de precisie is ook 3.
- 0.001 heeft 1 significant cijfer, de precisie is 3.
- 1.001 heeft 4 significante cijfers, de precisie is 3.
- 2.4500 heeft 5 significante cijfers, de precisie is 4.

In het voorbeeld hierboven zie je dat de getallen (bijna) allemaal dezelfde precisie hebben, maar wel een variatie aan significante cijfers.

2.2 Wetenschappelijke notatie

Een veel gebruikte manier om getallen en meetresultaten weer te geven is met behulp van de wetenschappelijke notatie. Bij de wetenschappelijke notatie wordt elk getal in de vorm $A \times 10^n$ opgeschreven. Een voordeel van deze notatie is dat je hiermee ook hele kleine getallen en hele grote getallen op een makkelijke manier op kunt schrijven. We geven een voorbeeld:

Voorbeeld klein getal We willen het getal 0.000000000004563 opschrijven met twee significante cijfers. Nu kunnen we natuurlijk 0.0000000000046 opschrijven maar als we dat vaak moeten doen kost dat veel ruimte (en werk). In de wetenschappelijke notatie ziet dit getal met twee significante cijfers er als volgt uit:

$$0.000000000004563 = 4.6 \cdot 10^{-12}$$

In het voorbeeld hierboven mag je natuurlijk zowel $4.6 \cdot 10^{-12}$ als 0.000000000004563 schrijven. Dat maakt niet uit. Bij grote ronde getallen is het vaak niet duidelijk hoe groot de significantie is. Met de wetenschappelijk notatie kunnen we dit duidelijk maken.

Voorbeeld groot getal Stel dat je het aantal knikkers in een pot hebt geschat op 2500. De onzekerheid is alleen in het laatste getal, maar dat kan je op deze manier niet zien. Je kan dit getal dan beter met de wetenschappelijk notatie schrijven. Bijvoorbeeld:

$$2.50 \times 10^3$$

of 25.0×10^2
Op zich mag je ook schrijven 250×10^1 maar in de praktijk doet niemand dit (10^1 gebruiken) en bovendien blijft bij dit voorbeeld dan nog steeds onduidelijk wat de significantie is.

In het algemeen geldt voor de wetenschappelijke notatie het volgende:

- Je schuift de decimale punt op zodat er een getal staat dat in absolute waarde groter is dan 1 en kleiner dan 10. Dit is het getal A.
- Heb je de decimale punt hierbij n plaatsen naar links verschoven dan vermenigvuldig je het getal A met 10^n. Heb je de decimale punt n plaatsen naar rechts verschoven dan vermenigvuldig je A met een factor 10^{-n}.
- Daarna rond je af op het gewenste aantal significante cijfers.

Hieronder een aantal voorbeelden:

Getal	Gewenste significantie	Wetenschappelijke notatie
0.00343	1 cijfer	$3 \cdot 10^{-3}$
0.00343	2 cijfers	$3.4 \cdot 10^{-3}$
0.00343	3 cijfers	$3.43 \cdot 10^{-3}$
10.7	2 cijfers	$1.1 \cdot 10^{1}$
255	2 cijfers	$2.6 \cdot 10^{2}$
34590	2 cijfers	$3.5 \cdot 10^{4}$

Let op! Bij natuurkundige resultaten is het vaak netter om het getal aan te passen aan een eenheid. Stel dat je een lengte meet, dan kan het netjes zijn om in plaats van 9.2×10^2 meter, 0.92 km te schrijven. De significantie blijft in dit geval hetzelfde. Gebruik de instructies hierboven als richtlijnen en niet als regels. Soms is het beter om ervan af te wijken, maar denk er wel over na!

2.3 Hoeveel significante cijfers noteren?

Het is dus belangrijk om niet te veel en niet te weinig **significante getallen** gebruiken als je een resultaat noteert.

Voor het noteren van een meetresultaat hanteren we de volgende regel:

- Als er **geen** meetonzekerheid op het resultaat bekend is dan noteren we het meetresultaat met 2 significante cijfers.
- Als er **wel** een meetonzekerheid op het resultaat bekend is, dan noteren we de onzekerheid met 2 significante cijfers en noteren we het meetresultaat met dezelfde precisie.

Voorbeeld

Resultaat	Onzekerheid	Notatie
2.515	0.2142	2.52 ± 0.21
2.515	onbekend	2.5
2515	241	$(2.52 \pm 0.24) \cdot 10^3$
2515	onbekend	$2.5 \cdot 10^3$
0.0471	0.12	0.05 ± 0.12
0.00148	10.38	0 ± 10
0.00148	onbekend	0.0015 of $1.5 \cdot 10^{-3}$
24018.2184	1.2125	24018.2 ± 1.2

NB. Als we teruggaan naar het voorbeeld met opmeten van de plank met het meetlint waarbij we hebben gemeten dat de plank 253.3 cm lang is, hebben we 4 significante cijfers

genoteerd. Het resultaat is genoteerd zonder meetfout. Toch is dit de juiste notatie geweest. De ingeschatte fout is immers in de orde van een millimeter. In de tabel hierboven wordt steeds aangegeven dat de onzekerheid onbekend is, in zeker zin is die bij de meting van de lengte van de plank *wel* bekend. Meer hierover volgt in de sectie over meetonzekerheid (Hfdst. 4).

2.4 Significantie en berekeningen

Voor het kiezen van het juiste aantal significante cijfers zijn er een aantal regels.

- Bij het vermenigvuldigen of delen van getallen krijgt het resultaat de significantie van het oorspronkelijke getal dat de laagste significantie had.

Voorbeeld Vermenigvuldigen we bijvoorbeeld 20.5 (drie significante cijfers) met 3.5 (twee significante cijfers) dan is het resultaat gelijk aan $20.5 \times 3.5 = 72$ (twee significante cijfers).

- Bij het optellen of aftrekken van getallen heeft het resultaat niet meer cijfers achter de decimale punt dan het gegeven met het minste aantal cijfers achter de decimale punt.

Voorbeeld Tellen we bijvoorbeeld 1.23 op bij 0.1 dan is het resultaat

$$1.23 + 0.1 = 1.3. \tag{2.1}$$

- Bij het vermenigvuldigen of delen van constante waardes (zoals $\pi, e, 100\%$ of 1) dan verandert de significantie niet. En bij optellen of aftrekken bij constante waardes verandert het aantal decimalen niet.

Voorbeeld Ongeveer 7.8% van de mannen in Nederland is kleurenblind. Dat betekend dat het percentage van de mannen die niet kleurenblind is gelijk is aan:
$100\% - 5.3\% = 94.7\%$. (Want, het is een aftreksom en we behouden dus het aantal decimalen.)

Voorbeeld We meten de straal van een cirkel $r = 3.15$ m en berekenen de omtrek.
omtrek $= 2 \cdot \pi r = 2 \cdot \pi \cdot 3.15$ m $= 19.8$ m
(Want, r heeft 3 significante getallen, dan moet de omtrek dat ook hebben.)

MODULE I Hoofdstuk 3

Data visualiseren

In dit deel bekijken we de verschillende manieren om data visueel te presenteren. Aan bod komen grafieken en scatterplots, staafdiagrammen en histogrammen. We laten ook zien hoe je deze met behulp van python kan maken.

Als je data visualiseert dan is het de bedoeling dat iemand anders deze goed kan begrijpen. Er zijn wel een aantal richtlijnen, maar het meest belangrijke is dat de data overzichtelijk is. Dat trends, of juist afwijkingen daarvan, goed zichtbaar worden gemaakt.

De richtlijnen zijn geen regels. Er zijn altijd uitzonderlijke datasets die erom vragen om af te wijken van de richtlijnen. Blijf dus altijd goed nadenken over wat je doet en waarom.

Afhankelijk van wat voor soort metingen je hebt genomen kies je uit een grafiek, een scatterplot, een staafdiagram of een histogram. Elk van deze data visualisatie methodes worden hieronder besproken.

3.1 Grafieken & Scatterplots

Grafieken en scatterplots zijn twee vormen van een diagram die veel op elkaar lijken. Ze verschillen wel op een paar punten.

Bij **scattterplots**,

- kunnen voor een ingestelde/gekozen waarde meer dan één gemeten waarden bestaan. Een voorbeeld zou zijn als je een meting doet waarbij je de lengte van mensen opmeet en tegen hun leeftijd uitzet.
- verbind je *nooit* punten met lijnen. Dat zou ook erg verwarrend zijn omdat je de dataset niet altijd logisch kan ordenen. In grafieken is dat vaak trouwens ook onwenselijk.

Bij **grafieken**,

- kies je meestal voor een van de twee variabelen een meetpunt of stel je een waarde in. De gemeten waarde laten we zien op de verticale as en de gekozen waarde op de

horizontale as.

3.1.1 Richtlijnen voor de opmaak van diagrammen

Met behulp van voorbeelden laten we zien wat de richtlijnen zijn en waar je op moet letten.

Stel bijvoorbeeld dat we naar de gemiddelde dagtemperatuur in de maand december 2019 in de Bilt kijken. Hier , in Fig. 3.1 een plot met een lijn tussen elk datapunt (Bron: KNMI, gehomogeniseerde data).

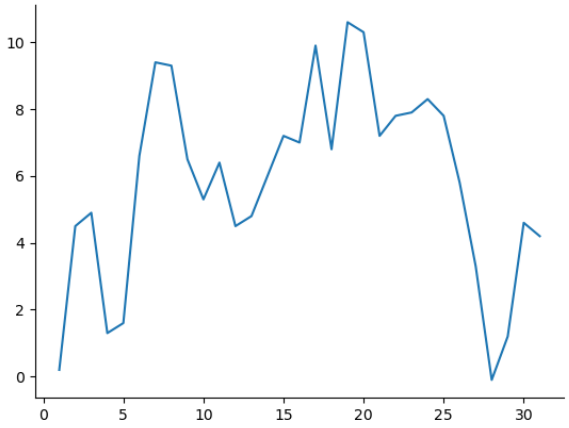


Figure 3.1: Plot met lijn tussen de datapunten.

Je ziet dat dit niet erg duidelijk is. Het is bijvoorbeeld niet precies te zien waar de gemeten punten zitten, we hebben wel een vermoeden doordat er punten zijn waarop de lijn abrupt van richting verandert, maar wie weet zitten er nog wel meer datapunten tussen.

Laten we dezelfde data eens plotten zonder lijnen maar alleen met punten.

Vanuit deze grafiek , Fig. 3.2 zien we waar de datapunten zijn. Dat konden we in de lijnplot niet goed zien. We kunnen nu helaas de trend niet meer goed waarnemen. Omdat er op een dag maar één gemiddelde gemeten temperatuur kan bestaan, is het toch beter deze als een grafiek weer te geven. We kiezen ervoor om zowel een lijn als markers te gebruiken.

De plot kan echter netter. Zo staan er geen labels op de assen. Nu kunnen we in dit geval wel raden welke as het jaar aangeeft en welke as de temperatuur, maar in veel gevallen is dat niet zo duidelijk. Om die reden moeten er altijd **labels op de assen** staan, zie het volgende figuur 3.3 .

Zoals je ziet hebben we het formaat van de grafiek ook aangepast zodat de distributie iets natuurlijker overkomt.

Een andere conventie is dat grafieken doorgaans **beginnen bij de oorsprong, tenzij de data dan onvolledig of onleesbaar wordt**. In het geval van het weergeven van de

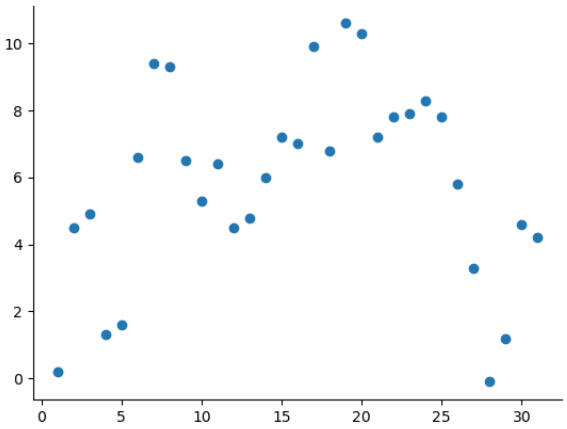


Figure 3.2: Plot met alleen de datapunten.

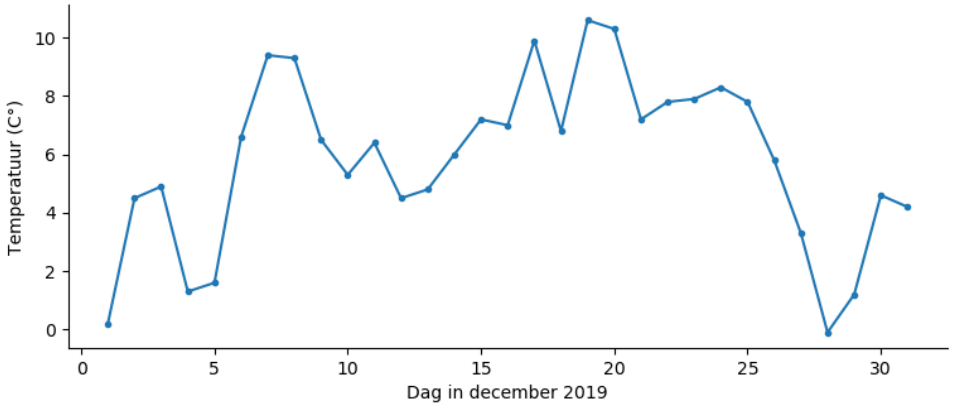


Figure 3.3: Plot met lijnen en datapunten en aslabels.

temperaturen wordt de data bijvoorbeeld onvolledig als we de temperatuur bij nul laten beginnen, we hebben immers ook temperaturen onder het vriespunt. In dit geval kunnen we de horizontale as wel bij nul laten beginnen, al is niet voor alle datums zo.

De assen kunnen nog wat netter. Zo eindigt de verticale as net voor de waarde 0, maar het is niet helemaal duidelijk bij welke waarde precies. De horizontale as begint een klein stukje voor 0 en eindigt een klein stukje na 30. Conventie is om assen te laten **beginnen en eindigen op een maatstreepje met een getal**. In ons geval laten we het beginnen op de eerste dag van de maand en de laatste dag, daarnaast laten we de temperatuur beginnen op -2 °C en eindigen op 16 °C. Dit tonen we in Fig. 3.4.

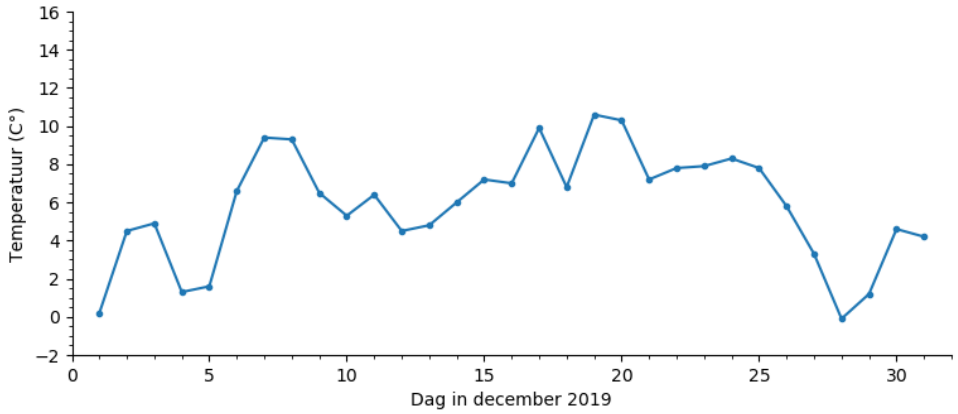


Figure 3.4: Plot met correct as-bereik.

Stel we willen de temperatuur in de Bilt nu weergeven naast de temperaturen gemeten in Vlissingen en Maastricht. De grafiek ziet er dan zo uit in Fig.3.5 .

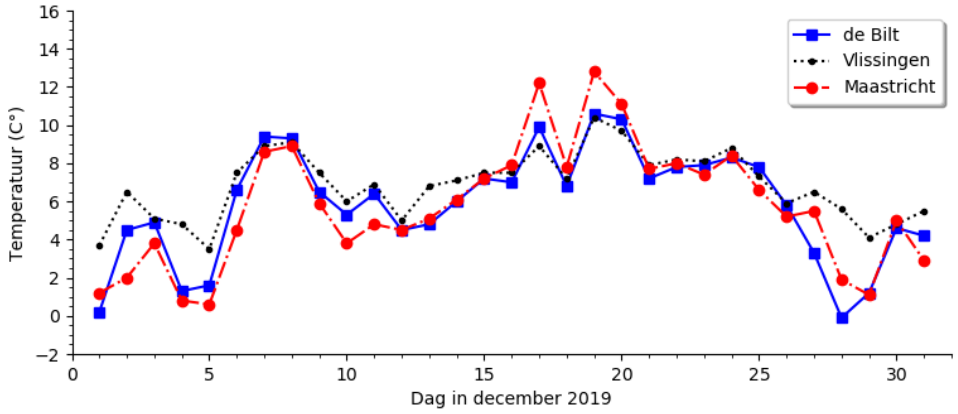


Figure 3.5: Plot met verschillende weerstations.

We hebben in Fig.3.5 ook een legenda toegevoegd zodat duidelijk is welke lijn bij welk weerstation hoort.

Tot nu toe hebben we nog geen titels toegevoegd aan de plots. Dit komt omdat dat voor verslagen en wetenschappelijke artikelen ongebruikelijk is, daar moet het onderschrift namelijk al vertellen wat er te zien is in de grafiek. In webteksten, lesteksten en presentaties kan het echter voorkomen dat een grafiek wel een titel heeft, omdat er in die context vaak geen onderschrift toegevoegd kan worden.

Samengevat:

- Een grafiek van een dataset wordt geplot met punten en eventueel lijnen.
- Het resultaat van een fit of een theoretisch verband wordt met een gladde lijn geplot.
- Bij een enkele dataset wordt geen legenda gebruikt. Als er meerdere datasets in één grafiek worden weergegeven dan is een legenda noodzakelijk.
- Aslabels geven weer wat elke as representeert (inclusief eenheden!).
- Assen beginnen in principe bij de oorsprong. Een uitzondering kan zijn als de data heel erg ver van de oorsprong af zit.
- Een as begint en eindigt op een groot maatstreepje met een waarde ('major tick') en niet op een klein maatstreepje of een maatstreep zonder getal. Tenzij er een heel goede reden is om hiervan af te wijken. (Zoals in het voorbeeld hierboven.)
- Een grafiek voor een wetenschappelijk artikel of een verslag heeft geen titel. Een grafiek voor webteksten of lesmateriaal heeft over het algemeen wel een titel.
- Als je de onzekerheid op de variabelen kent, dan is het goed om deze ook weer te geven in je plot. Tenzij deze heel onoverzichtelijk wordt (zoals in een scatterplot met heel veel punten).

Let op! Dit zijn weer richtlijnen en geen regels. Denk altijd goed na over wat je doet en waarom. Het eindresultaat moet goed begrijpbaar zijn en daarvoor is het soms nodig om van de richtlijnen af te wijken.

3.2 Staafdiagrammen & Histogrammen

Staafdiagrammen en histogrammen worden allebei typisch gebruikt om frequenties van meetwaardes aan te geven.

Hier zie je voorbeelden van een staafdiagram en een histogram.

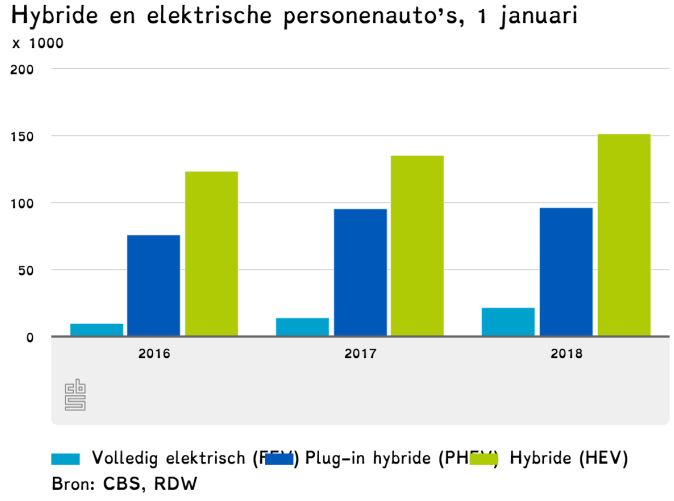


Figure 3.6: Een voorbeeld van een staafdiagram.

Hier , in Fig. 3.6 zie je een **staafdiagram** die de hoeveelheid auto's in Nederland laat zien over drie verschillende jaren opgesplitst naar drie auto categorieën.

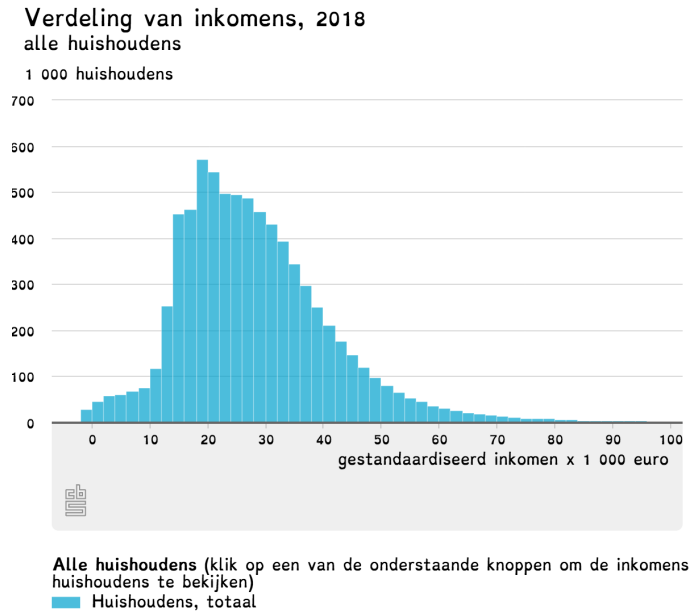


Figure 3.7: Een voorbeeld van een histogram.

Hierboven , in Fig. 3.7 zie je een **histogram** die de inkomensverdeling in Nederland laat zien.

Er is een belangrijk verschil tussen een staafdiagram en een histogram. Een staafdiagram laat de frequentie zien voor *gecategoriseerde* verdelingen. Een histogram wordt gebruikt om het resultaat van een *numeriek sorteerbare* verdeling mee weer te geven. In het geval van een histogram gaat het vaak om data met een continue variabele, zoals bijvoorbeeld bij het opmeten van lengte of gewicht. In dat geval sorteer je de data per interval.

Bij het weergeven van data in een histogram wordt de data gegroepeerd in intervallen. De breedte van de staven (in het vervolg 'bins' genoemd) geeft de breedte van de intervallen.

Bij een staafdiagram kun je de frequentie direct aflezen; voor één categorie lees je op de as af hoe vaak deze voorkomt. Voor een histogram is de frequentie gelijk aan de oppervlakte van de balken, en dus afhankelijk van de bin breedte.

3.2.1 Breedte van de bins bij een histogram

Voor een histogram is de breedte van de intervallen van belang. Als we te weinig bins kiezen dan worden de intervallen erg groot (/breed) en is er minder te zeggen over het gedrag van de data. Als we te veel bins kiezen dan fluctueert de hoogte van de (smalle) bins onderling erg en is het ook lastiger om de trend in de data goed in te schatten.

Dit bekijken we aan de hand van een voorbeeld , zie Fig. 3.8 . Zo zou het kunnen zijn dat

het ideale plaatje bij een gegeven dataset het volgende is.

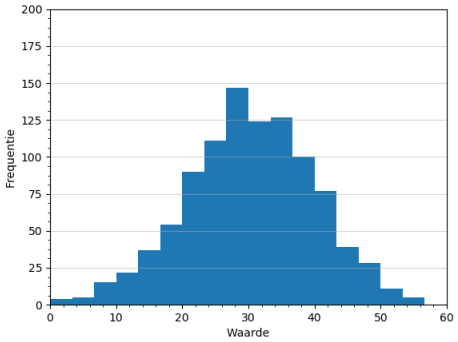


Figure 3.8: Een histogram met een goede bin en range keuze.

Als we te brede bins kiezen dan wordt de data afgevlakt en kunnen we het bovenstaande gedrag niet meer herkennen , zie Fig. 3.9 .

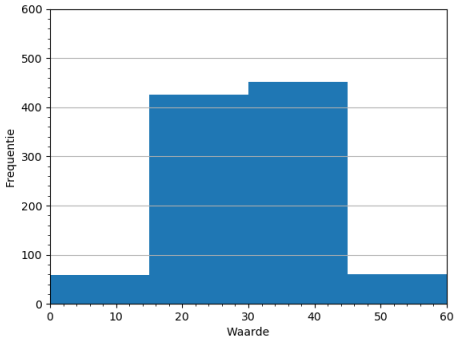


Figure 3.9: Een histogram met een te grove binning.

Kiezen we juist te smalle bins, ,zoals hieronder in Fig. 3.10 dan kunnen we het gedrag van de data nog wel herkennen (in dit geval) maar er is veel fluctuatie in de hoogte van de bins.

Met het kiezen van te veel bins hebben we dus visuele ruis geïntroduceerd, dit maakt het moeilijker om het gedrag op het oog te herkennen.

Bij het bepalen van het optimale aantal bins en de optimale bin breedte is het belangrijkste dat het gedrag van de data goed zichtbaar is. Er zijn verschillende formules (bijvoorbeeld de square of de Sturges formule) ontwikkeld waarmee je het aantal bins dat je nodig hebt kunt berekenen. Echter, geen van die formules kun je blind toepassen. Het is veel beter om gewoon goed naar je dataset te kijken en een inschatting te maken van de bin breedte.

Bij het maken van een histogram moet je goed letten op het volgende:

- Het bereik (de range) die je kiest op de horizontale as. Van waar tot waar plot je de data? Meestal wil je de gehele dataset laten zien, maar soms wil je juist inzoomen

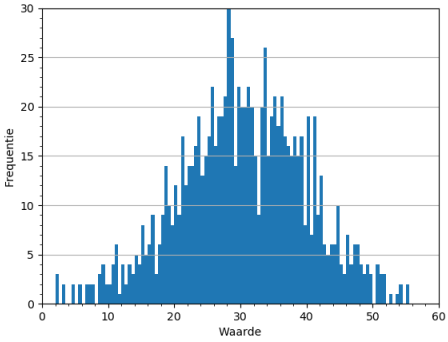


Figure 3.10: Een histogram met een te fijne binning.

op een kleiner stukje.

- De bin breedte. Meestal kies je voor het hele histogram dezelfde bin breedte, in sommige gevallen kun je verschillende bin breedtes kiezen. In elk geval geldt dat het histogram goed 'leesbaar' moet zijn. Het moet duidelijk blijven hoe de data gedistribueerd is. Wat is de trend? Zijn er afwijkingen van die trend.
- Let bij histogrammen erg goed op waar de grens van een bin ligt. Vooral als je een dataset met natuurlijke getallen weergeeft is het belangrijk dat de bin grenzen netjes *tussen* de natuurlijke getallen ligt. Anders kan de distributie van de data verkeerd gerepresenteerd worden.
- Het histogram is makkelijker leesbaar als de bins een natuurlijk interval hebben. Als je range van 0 tot 10 loopt is het heel gek om deze te verdelen in 7 bins.

Voor zowel histogrammen als staafdiagrammen geldt:

- Natuurlijk moeten bij histogrammen en staafdiagrammen ook netjes aslabels worden gebruikt.
- Als je meer dan één dataset laat zien maak dan gebruik van een legenda.

3.3 Wanneer gebruik je wat?

1. Als je de incidentie (of frequentie) van meetwaarden wil laten zien dan gebruik je een histogram of staafdiagram.
   - Een staafdiagram gebruik je als de meetwaarden discreet zijn gecategoriseerd, bijvoorbeeld in het soort auto of per kleur.
   - Een histogram gebruik je voor variabelen die numeriek geordend kunnen worden, zoals bijvoorbeeld variabelen met integer of continue waardes.
2. Als je de relatie tussen twee variabelen wilt tonen kies je voor een grafiek of een scatterplot.
   - Je gebruikt een grafiek als de afhankelijke variabele (meestal de langs de x-as) unieke waardes kent. Dus voor een bepaalde waarde van x is maar één uitkomst van y mogelijk. Andersom zijn er wellicht meerdere waardes voor x voor een

bepaalde gemeten grootheid y. Bijvoorbeeld de gemeten temperatuur om 12 uur 's middags op een bepaalde locatie. Er kan maar 1 gemeten temperatuur bestaan.

- Je gebruikt een scatterplot als er geen unieke waarde is per afhankelijke variabele. Bijvoorbeeld als je de lengte van een student meet in relatie met de leeftijd. Er zijn waarschijnlijk meerdere studenten met dezelfde leeftijd in de groep die hoogstwaarschijnlijk in lengte van elkaar verschillen.

3.4 Data plotten met Python

Om in Python te kunnen plotten moeten we als eerste een library importeren die ingebouwde functies heeft voor het visueel weergeven van data. Een populair pakket is Matplotlib, deze zullen we in dit vak dan ook gebruiken (er zijn ook andere geschikte pakketten zoals Seaborn, geplot en Plotly). We importeren de `pyplot` functie vanuit Matplotlib en geven deze de naam 'plt' met het volgende commando:

```
import matplotlib.pyplot as plt
```

De naamgeving `plt` met het commando `as plt` is optioneel, maar wel handig omdat we deze functie over het algemeen vaak zullen gebruiken (dat scheelt typen).

3.4.1 Voorbeeld: een grafiek plotten

Stel we hebben de hoogte van een vallende bal gemeten als functie van de tijd. In de tabel is de gemeten data weergegeven:

t (s)	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0
h(cm)	180.0	178.8	175.1	169.0	160.4	149.3	135.9	120.0	102.0	80.7	57.4	31.6	3.4

Nu maken we een lijst `t_data` aan voor de tijd en een lijst `h_data` voor de hoogte van de bal:

```
t_data = [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0]
h_data = [180.0, 178.8, 175.1, 169.0, 160.4, 149.3, 135.9, 120.0, 102.0,
    80.7, 57.4, 31.6, 3.4]
```

Daarna roepen we het `plot` commando uit matplotlib.pyplot aan:

```
plt.plot(t_data, h_data, 'ro')
```

Met `'ro'` geven we aan dat we rode gevulde punten in de plot willen. De plot ziet er nu als volgt (zie Fig. 3.11) uit.

Je ziet dat de assen automatisch vanaf de laagste waarde tot aan de hoogste waarden gaan, en hierbij niet eindigen op een maatstreepje. Daarnaast willen we graag labels op de assen.

De limiet van de assen kunnen we aangeven met de commando's `plt.xlim` en `plt.ylim`:

```
plt.xlim(0,7)
plt.ylim(0,200)
```

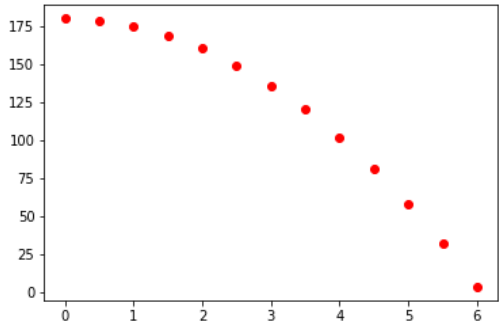


Figure 3.11: Plot met rode punten.

Labels voor de assen kunnen we als volgt specificeren:

```
plt.xlabel('t (s)')
plt.ylabel('h (cm)')
```

Het resultaat is (Fig. 3.12) .

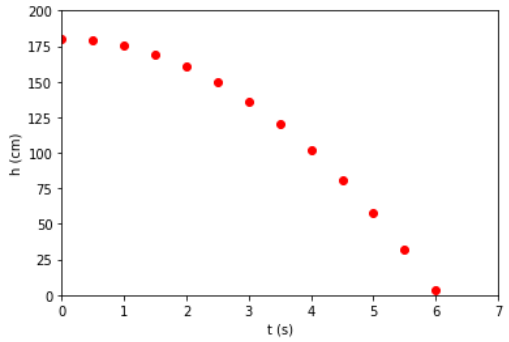


Figure 3.12: Plot met aslabels en -ranges.

De volledige code tot nu toe is:

```
# dataset in lijsten zetten
t_data = [0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5,5.5,6]
h_data =
    [180,178.8,175.1,169.0,160.4,149.3,135.9,120,102,80.7,57.4,31.6,3.4]

# data plotten, as-limieten instellen, as-labels instellen
plt.plot(t_data, h_data, 'ro')
plt.xlim(0,7)
plt.ylim(0,200)
plt.xlabel('t (s)')
plt.ylabel('h (cm)')
```

Als we nu nog een dataset hebben, bijvoorbeeld van dezelfde bal die vanaf een hoogte van 160 cm valt in plaats van een hoogte van 180 cm:

```
# tweede dataset in lijsten zetten
t_data2 = [0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5,5.5]
h_data2 = [160,158.8,155.1,149.0,140.4,129.3,115.9,100,82,60.7,37.4,11.6]
```

Deze dataset kunnen we in de grafiek van de eerste plotten door twee keer het plot commando achter elkaar te gebruiken: daarna gebruiken we weer dezelfde eigenschappen voor de limieten en de aslabels:

```
plt.plot(t_data, h_data, 'ro')
plt.plot(t_data2, h_data2, 'bo')
```

Daarna gebruiken we weer dezelfde eigenschappen voor de as-limieten en de as-labels:

```
plt.xlim(0,7)
plt.ylim(0,200)
plt.xlabel('t (s)')
plt.ylabel('h (cm)')
```

De plot ziet er dan als volgt uit (Fig. 3.13) .

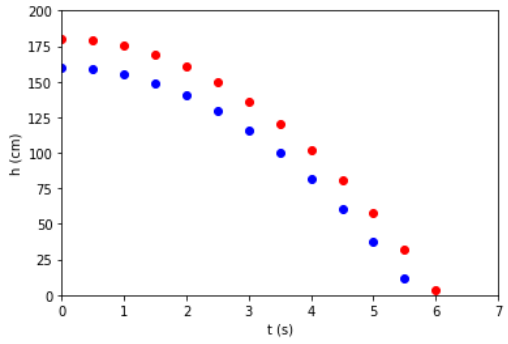


Figure 3.13: Twee datasets.

Omdat er meerdere datasets in één grafiek zijn weergegeven is het noodzakelijk om hier een legenda bij te plaatsen. Een legenda kan op meerdere plaatsen in de figuur neergezet worden. Voordat we de legenda kunnen toevoegen moeten we de plots eerst labelen. Dit doen we door `label = "naam"` achteraan in de `plot` commando's toe te voegen:

```
plt.plot(t_data, h_data, 'ro' , label='h(0) = 180 cm')
plt.plot(t_data2, h_data2, 'bo', label='h(0) = 160 cm')
```

Nu kunnen we de legenda als volgt toevoegen (hier kiezen we ervoor om de legenda in de rechterbovenhoek neer te zetten zodat er geen overlap is met de grafieken zelf):

```
plt.legend(loc='upper right', shadow=True, ncol=1)
```

De grafiek is nu als volgt (Fig. 3.14) .

De volledige code tot nu toe is:

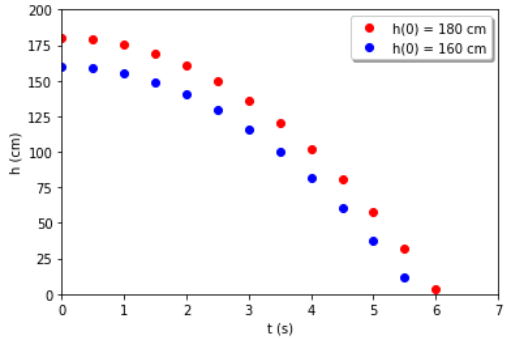


Figure 3.14: Plot met legenda.

```
# dataset in lijsten zetten
t_data = [0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5,5.5,6]
h_data =
    [180,178.8,175.1,169.0,160.4,149.3,135.9,120,102,80.7,57.4,31.6,3.4]

# tweede dataset in lijsten zetten
t_data2 = [0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5,5.5]
h_data2 = [160,158.8,155.1,149.0,140.4,129.3,115.9,100,82,60.7,37.4,11.6]

# data plotten, as-limieten instellen, as-labels instellen
plt.plot(t_data, h_data, 'ro' , label='h(0) = 180 cm')
plt.plot(t_data2, h_data2, 'bo', label='h(0) = 160 cm')
plt.xlim(0,7)
plt.ylim(0,200)
plt.xlabel('t (s)')
plt.ylabel('h (cm)')

# legenda toevoegen
plt.legend(loc='upper right', shadow=True, ncol=1)
```

Een ander voorbeeld is dat we lijnen willen plotten, bijvoorbeeld van een theoretisch verband. (De conventie is dat data altijd met punten wordt uitgebeeld.) Dit kan je als volgt doen:

```
# datasets in lijsten
x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y1 = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21] #2x+1
y2 = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] #x+2

# datasets plotten als 'solid' lijnen
plt.plot(x, y1, 'r-' , label='dataset 1')
plt.plot(x, y2, 'b-', label='dataset 2')
plt.xlim(0,7)
plt.ylim(0,20)
plt.xlabel('x')
plt.ylabel('y')

# legenda toevoegen
plt.legend(loc='upper right', shadow=True, ncol=1)
```

De bijbehorende plot (Fig. 3.15) .

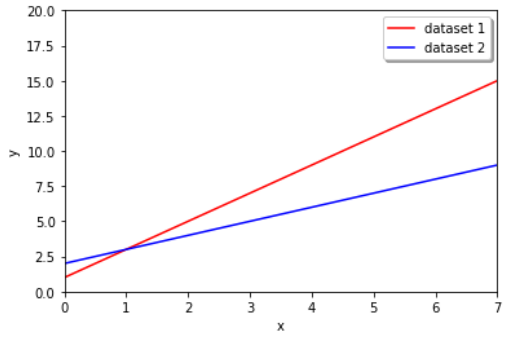


Figure 3.15: Een plot met twee lijnen en legenda.

3.4.2 Voorbeeld een histogram plotten

In de allereerste opgave M1.1 ga je een histogram plotten. In die opgave staat stap voor stap uitgelegd hoe je dat moet doen.

Maar kijk vooral ook in de online manual van matplotlib.

Succes!

MODULE I Hoofdstuk 4

Metingen en onzekerheid

4.1 Fouten en onzekerheden

Als we iets meten is dat meestal omdat we een bepaalde grootheid willen weten. Voorbeelden hiervan zijn

- Het aantal knikkers in een pot.
- De lengte van een blokje hout.
- De levensduur van Cesium-131.
- Het aantal sterren in een bolhoop.
- De snelheid van het licht.

We doen ook metingen om bepaalde hypothesis te ontkrachten of juist te bevestigen. Maar ook als we een hypothese toetsen zijn er onderliggende grootheden (vaak meerdere tegelijk) die we meten en vergelijken met voorspelde waardes.

Het is belangrijk om te begrijpen dat we meestal te maken hebben met een fout op de gemeten grootheid. **Hoe groot die fout is kunnen we niet weten, wel kunnen we de meetonzekerheid in kaart proberen te brengen.** Die onzekerheid kan grofweg twee oorzaken hebben. Onzekerheden die komen door de meetmethode en onzekerheden met een natuurlijke oorzaak. Technisch gezien is er dus een verschil tussen de fout en de onzekerheid van de meting. In de praktijk gebruiken we vaak het woord fout ook voor de onzekerheid. Toch is het goed om even stil te staan bij dit verschil.

4.1.1 Meetfouten

De eerste oorzaak is dat we een fout maken bij de meting. Een fout betekent hier niet dat we iets verkeerds doen. Het betekent dat we met onze meting een waarde vinden die afwijkt van de echte waarde. De afwijking noemen we een meetfout. Helaas weten we nooit hoe groot de meetfout exact is, maar vaak kunnen we hem wel goed inschatten. De meetfout wordt beïnvloed door de meetmethode. Voorbeelden hiervan zijn:

- **Het aantal knikkers in een pot.** Stel dat we de pot leeg kieperen en we tellen

ze allemaal met de hand, dan kan het zo zijn dat we een knikker kwijtraken of een telfout maken. Als we het netjes doen en we doen verschillende hertellingen is de kans groot dat we hierbij geen meetfout maken (zeker als het maar een paar knikkers zijn).

- **De lengte van een blokje hout.** Hier zou je bijvoorbeeld een liniaal voor kunnen gebruiken. Allereerst moeten we het blokje netjes langs de liniaal leggen. Zie hieronder in Fig. 4.1, een schets van de opstelling. Ligt de 0 wel echt netjes langs de rand?

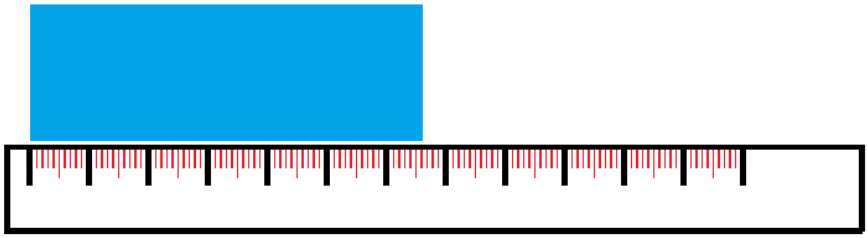


Figure 4.1: Schets van een meetopstelling

Dan moeten we de waarde op de lineaal afmeten. Als we naar de bovenstaande situatie kijken dan zou het blokje 7.6 cm lang kunnen zijn. Maar het is niet helemaal goed af te lezen. Zo zou het blokje ook 7.7 cm lang kunnen zijn als we de linkerkant van het blokje aan de binnenkant van de eerste zwarte streep leggen, en het kan 7.5 cm zijn als we het blokje aan de buitenkant van de eerste zwarte streep leggen. Omdat er geen streepjes tussen de rode streepjes zitten, kunnen we slechts op een mm nauwkeurig zeggen wat de lengte is van het blokje. Er is dus sprake van een meetonzekerheid. In dit geval zouden we bijvoorbeeld noteren dat het blokje een lengte heeft van 7.6 ± 0.1 cm.

- **De levensduur van Cesium-131.** Meestal meten we dit met behulp van een Geiger-Müller telbuis. We meten bijvoorbeeld het aantal tellingen in 5 minuten en deze metingen herhalen we een paar keer met steeds een even groot tijdsinterval. Meetfouten komen door de nauwkeurigheid van de twee soorten intervallen, het meetinterval en de tussenliggende periode. Ook zullen er, hopelijk veel kleinere, onzekerheden komen doordat de telbuizen niet altijd precies hetzelfde functioneren.
- **Het aantal sterren in een bolhoop.** Het aantal sterren in een bolhoop kunnen we niet met de hand tellen zoals we met de knikkers in een pot doen. Bolhopen kunnen 100.000 of zelfs wel meer dan een miljoen sterren bevatten. De telling van het aantal sterren in een bolhoop is vaak een combinatie van meerdere methodes, die allemaal hun onnauwkeurigheden kennen.
- **De snelheid van het licht.** De lichtsnelheid kun je op verschillende manieren meten. Maar hoe je het ook meet, je zal zeer nauwkeurige tijds- en lengtemetingen nodig hebben. Als je de lichtsnelheid in vacuüm wilt bepalen zal je ook nog eens zeer goede conditie van je vacuüm nodig hebben òf hiervoor moeten corrigeren. Er

zijn vele bronnen van onnauwkeurigheid in dit experiment die je goed zal moeten controleren.

4.1.2 Onzekerheden met natuurlijke oorzaak

De tweede categorie onzekerheden hebben een andere oorzaak. De grootheid zelf kan ook een spreiding kennen. Je kan hierbij denken aan toevalligheden in een productieproces, maar ook door bijvoorbeeld kans processen in de kwantummechanica. We bekijken nogmaals de voorbeelden.

- **Het aantal knikkers in een pot.** Stel dat de pot een onneembaar aantal knikkers heeft. Dan wordt het een monnikenwerk om ze allemaal met de hand te tellen. Een goede methode zou dan zijn om de pot te wegen. Als we nu het gewicht van de pot en het gewicht van een knikker weten dan kunnen we uitrekenen hoeveel knikkers er in de pot zitten. Behalve de meetfouten die we maken bij de weging kan er ook een onzekerheid komen in de meting van het aantal, doordat er een spreiding is in knikker gewichten. Niet iedere knikker is precies even zwaar.
- **De lengte van een blokje hout.** Stel dat we de nauwkeurigheid van de lengtemeting helemaal onder controle hebben, dan kan het nog steeds zo zijn dat het blokje zelf niet overal precies dezelfde lengte heeft. Vooral bij houten blokjes zal er zo nu en dan variatie in zitten door nerven en knoesten.
- **De levensduur van Cesium-131.** Kwantummechanische kans processen spelen hier een grote rol. De kans dat een Cesium-131 vervalt binnen een bepaalde tijd is volkomen gedreven door de kwantummechanica (anders zouden ze natuurlijk allemaal tegelijk vervallen). Deze onzekerheid resulteert uiteindelijk in een variatie in het aantal tellingen dat je in een van de tijdsintervallen meet. Deze onzekerheid volgt de Poisson statistiek, waar we later meer over zullen lezen.
- **Het aantal sterren in een bolhoop.** Een van de methodes om het aantal sterren te meten in een bolhoop berust op het meten van de dichtheid. Deze dichtheid neemt meestal af over de straal van de bolhoop. In de berekeningen ga je er meestal van uit dat dit homogeen afneemt. Dat wil zeggen dat op een afstand r van het centrum van de bolhoop overal dezelfde dichtheid voorkomt. In het echt zijn er fluctuaties in de dichtheid. De fluctuaties zorgen uiteindelijk ook voor een onzekerheid in de telling.
- **De snelheid van het licht.** Ook hier spelen kans processen een rol. Als je opstelling een roterend radartje bevat, dan zijn er hoogstwaarschijnlijk variaties in de tanden van het radartje. Kijk bijvoorbeeld naar het experiment dat hier beschreven staat.

4.2 Reduceren van meet-onnauwkeurigheden

Goed nadenken over de opzet van een experiment is belangrijk en kan grote onnauwkeurigheden voorkomen. Nog belangrijker is het om alle onzekerheden goed in kaart te brengen. Alleen zo kun je inschatten wat de waarde is van een meting.

Een voorbeeld is een keukeninstallateur die een werkblad voor een keuken moet opleveren. De installateur zal een goede meting moeten doen van de lengte van het werkblad die hij nodig heeft. Als hij een werkblad aanlevert dat uiteindelijk 2 cm te kort is, past het blad niet, dit is niet meer op te vullen met een kit randje. Als het keukenblad 3 mm te lang is zal het natuurlijk ook niet passen.

Zo werkt het ook bij natuur- en sterrenkundige experimenten. Als je een meting wilt doen, zul je eerst goed moeten kijken hoe nauwkeurig het resultaat moet zijn. Wil je een hypothese weerleggen die voorspelt dat een hyperfijnstructuur in de spectraallijn van een atoom 1 nm vergroot, dan zal je ook de nauwkeurigheid moeten bereiken om dat te kunnen meten.

We onderscheiden **systematische onzekerheid**, **statistische onzekerheid** en **theoretische onzekerheid**.

Systematische onzekerheden hangen af van de meetopstelling en zijn niet te voorkomen. We kunnen hem soms wel reduceren door de meetopstelling te verbeteren. Een enkele bron van een systematische fout is eenzijdig, dat wil zeggen dat de gemeten waarde consequent te hoog of te laag uitvalt door bijvoorbeeld kalibratie fouten van de meetinstrumenten. Vaak zijn er in een experiment combinaties van systematische onzekerheden waardoor de gemeten waarde zowel te hoog als te laag kan uitvallen. Systematische onzekerheden zijn lastig te vinden in opstellingen en zijn vooral te voorkomen door kritisch te kijken en na te denken over de meetopstelling. Een systematische fout kunnen we bijvoorbeeld verbeteren door het blokje hout uit het voorbeeld met een schuifmaat te meten, zo kunnen we de meetonzekerheid verkleinen tot een tiende van een millimeter. We kunnen zorgen dat de tanden van de radartjes uit het voorbeeld van de meting van de lichtsnelheid heel gelijk zijn.

Statistische onzekerheden zijn reduceerbaar door het experiment te herhalen. Bijvoorbeeld kunnen we tijdens een langer interval het aantal tellingen meten in het Cesium-131 levensduur experiment. Ook kunnen we meer meetpunten verzamelen. De relatieve fout op de telling zal dan kleiner worden en daarmee ook de uiteindelijk onzekerheid op de levensduur meting.

Theoretische onzekerheden kunnen voorkomen als we gebruik maken van aannames met theoretische grondslag. Als we de onzekerheden in deze aannames kunnen kwantificeren hebben we een maat voor de theoretische onzekerheid. Soms kunnen theoretische onzekerheden worden verkleind door bijvoorbeeld meer berekeningen uit te voeren.

4.3 Onderliggende verdelingen

Je begrijpt nu dat veel metingen wel herhaalbaar zijn, maar dat je niet altijd precies dezelfde resultaat verwacht te meten. Het gevolg hiervan is dat je een verdeling of distributie krijgt van je meetresultaten. Van deze verdeling kunnen we bepaalde eigenschappen uitrekenen. Meer hierover kun je vinden in het hoofdstuk basisbegrippen (Hfdst. 1). Als je de onderliggende verdeling zou kennen (soms is dat zo, maar soms ook niet) dan hoor de meetonzekerheid overeen te komen met de standaardafwijking van de onderliggende

verdeling van resultaten. Het is belangrijk om de verdelingen goed te presenteren, meer daarover kun je hier (Hfdst. 3) lezen.

4.4 Meetonzekerheden noteren

We kunnen de meetonzekerheden op verschillende manieren noteren.

Het meetresultaat zelf noemen we de **centrale waarde** en de meetonzekerheid heet ook wel de **absolute fout**. We noteren dit als volgt:

$$\text{gemeten waarde van } x = x_{\text{centraal}} \pm \Delta x \tag{4.1}$$

.

Waarbij Δx de meetonzekerheid is.

Wat je ook tegen kunt komen is dat de fout tussen haakjes wordt gezet achter de decimalen waar de fout van invloed op is. Hebben we bijvoorbeeld

$$1.456 \pm 0.004 \tag{4.2}$$

dan kunnen we dit ook noteren als:

$$1.456(4) \tag{4.3}$$

Dit wordt met name vaak gebruikt als een meetwaarde met meetonzekerheid in de wetenschappelijke notatie wordt weergegeven. Het tussen haakjes zetten van de meetonzekerheid is dan namelijk korter dan de notatie met een plusminus.

We kunnen in de wetenschappelijke notatie bijvoorbeeld $(4.51 \pm 0.27) \cdot 10^3$ schrijven. Dit kunnen we ook als $4.51 \cdot 10^3 \pm 0.27 \cdot 10^3$ schrijven (minder gebruikelijk). Als we de fout echter tussen haakjes zetten wordt dit een stuk korter en schrijven we:

$$4.51(27) \cdot 10^3 \tag{4.4}$$

Hieronder de verschillende schrijfwijzen naast elkaar gezet in een tabel voor diverse meetwaarden met meetonzekerheden.

Meetwaarde	Notatie met $\pm$	Notatie met haakjes
100.5 ± 1.8	$(1.005 \pm 0.018) \cdot 10^2$	$1.005(18) \cdot 10^2$
0.0045 ± 0.0006	$(4.5 \pm 0.6) \cdot 10^{-3}$	$4.5(6) \cdot 10^{-3}$
300.0 ± 40	$(3.0 \pm 0.4) \cdot 10^2$	$3.0(4) \cdot 10^2$
56934 ± 160	$(5.693 \pm 0.016) \cdot 10^4$	$5.693(16) \cdot 10^4$

Soms is het nuttig om de **relatieve fout** te gebruiken. Deze wordt gegeven door de waarde van de absolute fout te delen door de centrale waarde.

$$\text{relatieve fout} = \frac{\Delta x}{x_{\text{centraal}}} \tag{4.5}$$

De relatieve fout is onder andere handig als er meetwaarden vergeleken moeten worden die in een heel andere orde van grootte zitten. Zo zouden we bijvoorbeeld de gemeten snelheid van een vliegtuig kunnen vergelijken met de gemeten snelheid van een hardloper. Stel de gemeten snelheid van een vliegtuig is $v_{vliegtuig} = 803 \pm 3\text{km/h}$. De gemeten snelheid van een hardloper is

$$v_{hardloper} = 18.3 \pm 0.2 \tag{4.6}$$

km/h. Welk van de metingen heeft met een grotere precisie plaatsgevonden?

Dit is niet direct uit de absolute fout te zien, maar wel vanuit de relatieve fout. De relatieve fout behorende bij de snelheid van het vliegtuig is

$$\frac{3}{803} = 0.004. \tag{4.7}$$

De relatieve fout behorende bij de snelheidsmeting van de hardloper is $\frac{0.2}{18.3} = 0.01$. Dit betekent dat de snelheidsmeting van het vliegtuig met een grotere precisie heeft plaatsgevonden.

Soms werkt een relatieve fout ook juist weer niet. Bijvoorbeeld als je heel nauwkeurig een faseverschil probeert te meten tussen twee golven. Een faseverschil van bijna 0° is ook een meting en een relatieve meetfout zegt hierover bijna niets.

MODULE I

Hoofdstuk 5

Kanstheorie

In dit hoofdstuk leren we over kanstheorie en kansdichtheidsfuncties. Kanstheorie speelt een belangrijke rol in het begrijpen en bepalen van meetonzekerheden. Zoals in het hoofdstuk over meetonzekerheden (Hfdst. 4) is uitgelegd kunnen meetonzekerheden verschillende oorzaken hebben. Bij elk van die oorzaken hoort een bepaalde waarschijnlijkheidsverdeling en deze zijn verbonden aan kansprocessen.

Vaak willen we metingen gebruiken om voorspellingen te doen of hypotheses te toetsen. Als we hiervoor een bepaalde serie meetgegevens willen gebruiken, dan is het belangrijk om te weten wat de meetonzekerheden zijn. Deze kunnen we vervolgens gebruiken om te kijken hoe goed ze passen bij een bepaald voorspeld patroon of hoe goed ze een theorie bevestigen of juist weerleggen.

Om die stap later te kunnen maken, moeten we eerst meer leren over kanstheorie en hierna over kansdichtheidsfuncties. In dit hoofdstuk maken we daar een begin mee.

5.1 Definitie van Kans

Waarschijnlijk is iedereen wel bekend met het concept van kans. We gebruiken het vaak. Wat is de kans dat het regent? Wat is de kans om de loterij te winnen?

Wiskundig is een kans gedefinieerd als een getal tussen de 0 en de 1 dat aangeeft hoe waarschijnlijk het is dat een bepaalde gebeurtenis zal plaatsvinden. Een kans van 1 zegt dat het **zeker** zal gebeuren en een kans van 0 dat het **zeker niet** zal gebeuren. Een kans van 0.5 geeft aan dat in 50% van de gevallen de gebeurtenis zal plaatsvinden.

Voorbeeld We kijken naar een dobbelsteen. Wat is de kans dat je een 4 gooit als je de dobbelsteen 1 keer gooit? Voor een normale dobbelsteen kunnen we deze kans uitrekenen met behulp van de volgende formule:

$$P(\text{uitkomst is 4}) = \frac{\text{aantal uitkomsten met een 4}}{\text{totaal aantal uitkomsten}} = \frac{1}{6}.$$

Dit is de kans voor een normale eerlijke dobbelsteen. Met eerlijk bedoelen we hier dat de dobbelsteen niet gemanipuleerd is en dat elk vlak van de dobbelsteen evenveel kans heeft om boven te eindigen.

Stel nu dat we een speciale, maar wel eerlijke, dobbelsteen zouden hebben met de volgende vlakken: {1,2,2,3,4,4}. De mogelijke uitkomsten bij een dobbelsteenworp zijn nu: {1,2,3,4}. Dit noemen we ook de **uitkomstenverzameling** waarbij alle elementen uniek zijn, en dus maar 1 keer voorkomt.

De kans om nu een 4 te gooien is groter dan met een normale eerlijke dobbelsteen. Dit werken we uit in het volgende voorbeeld.

Voorbeeld Als we de kans nu berekenen voor de speciale dobbelsteen met vlakken {1,2,2,3,4,4} dan is de kans om vier te gooien:

$$P(\text{uitkomst is 4}) = \frac{\text{aantal uitkomsten met een 4}}{\text{totale aantal uitkomsten}} = \frac{2}{6}.$$

En stel nu dat we een normale dobbelsteen hebben die gemanipuleerd is? Dan zal de kans om een 4 te gooien anders zijn. Een goede manier om dan de kans te bepalen is met behulp van de **Frequentist** formule. De algemene formule voor de **Frequentist definitie** van kans is:

$$P(\text{uitkomst A}) = \lim_{n\to\infty} \frac{\text{aantal uitkomsten A}}{n}. \tag{5.1}$$

Waarbij we het aantal malen dat we uitkomst A krijgen uit ons experiment delen door het totaal aantal metingen, n, dat we hebben verricht.

In het geval van de dobbelsteen wordt deze formule dan:

$$P(4) = lim_{n\to\infty} \frac{\text{uitkomst is 4}}{\text{totaal aantal worpen}}. \tag{5.2}$$

Hoe vaker we de dobbelsteen gooien des te nauwkeuriger kunnen we de kans dat we een 4 gooien, bepalen.

De Frequentist definitie voor kans is een goede manier om kansen te berekenen. Het kent echter twee grote beperkingen. De eerste is dat we eigenlijk nooit een oneindig aantal metingen kunnen doen. Dit is goed te benaderen door gewoon een heel groot aantal metingen te doen. De tweede beperking is dat niet alle experimenten herhaalbaar zijn.

5.1.1 Frequentist versus Bayesiaanse methode

Het zal je dan misschien niet verbazen dat er nog een andere methode bestaat die wel werkt voor experimenten die niet herhaalbaar zijn of een beperkte statistiek hebben. Deze manier noemen we ook wel de Bayesiaanse (spreek uit: Beej-sie-jaanse) methode (Engels: Bayesian).

De frequentist methode wordt in het algemeen als objectieve methode gezien en de Bayesiaanse methode een subjectieve manier. Het geeft aan wat je denkt dat de waarschijnlijkheid is. Dat klinkt misschien niet erg wetenschappelijk maar in de praktijk is dit misschien wel de meest gebruikte methode. Vooral omdat je hem ook kan gebruiken als het experiment niet herhaalbaar is. De bayesiaanse methode zegt eigenlijk dat je nooit helemaal zeker kunt stellen wat de grootte van een kans is. Dat voelt misschien wat gek, maar het enige wat het zegt is dat ook bij een berekende kanswaarde er een mate van onzekerheid is. Ook daar is er sprake van een 'meetonzekerheid'.

Een voorbeeld In een wielerronde staat een bergklassieker op het programma van vandaag. De wedstrijd is nog niet gestart. Er staan twee sterke renners, Verstappen en Onana, op de gedeelde eerste plaats van het klassement en de voorsprong met de derde wielrenner is meer dan 20 minuten. Het lijkt dus waarschijnlijk dat aan het einde van de dag Verstappen of Onana op de eerste plaats in het klassement zal staan. Op bergetappes wint Onana 9 van de 10 keer met een flinke voorsprong van Verstappen. Wie denk je dat er vandaag wint?

We kunnen het experiment natuurlijk niet herhalen maar het lijkt zeer waarschijnlijk dat Onana aan het einde van de dag op nummer 1 zal eindigen. Hier maken we gebruik van de subjectieve methode van Bayes. Om het te kwantificeren kunnen we misschien zelfs wel zeggen dat de kans 0.9 is.

Maar nu zitten we aan het ontbijt en we zien dat Onana geen hap door zijn keel krijgt. Hij is duidelijk erg ziek. Verstappen daarentegen ziet er fris en sterk uit.
Hoe waarschijnlijk denk je nu dat het is dat Onana zal winnen?

Het lijkt nu toch een stuk minder waarschijnlijk dat Onana zal winnen. Misschien schat je nu de kansen lager in dan de 0.9 waarmee je begon. Misschien heb je zelfs wel informatie uit het verleden waaruit je weet hoeveel langzamer renners zijn als ze er zo ziek uitzien als Onana, wat voor impact dat heeft op hun performance. Dan zouden

we onze kans van 0.9 kunnen 'updaten' met de nieuwe informatie. Dat is typisch een Bayesiaanse methode om kansen uit te rekenen.

Beide methodes worden dus gebruikt, maar de Bayesiaanse methode, of zelfs een hybride methode vindt vooral zijn toepassing in complexe modellen en voorspellingen. In dit vak zullen we echter vooral werken met de frequentist methode. Wat in elk geval belangrijk is, is om altijd heel precies te vermelden wat de voorwaardes zijn geweest waaronder de kans is uitgerekend. Ook bij de frequentist methode!

5.2 Rekenen met kansen

Er zijn een paar basisregels waar kansen aan voldoen.

1. **Behoud van kans:** Een gebeurtenis, A, kan plaatsvinden, of het kan niet plaatsvinden. De kans is behouden en dat betekent dat:

$$P(A) + P(\text{niet A}) = 1 \tag{5.3}$$

2. **Complementregel:** Een direct gevolg hiervan is dat $P(\text{niet A})$ het complement is van $P(A)$ ofwel:

$$P(\text{niet A}) = 1 - P(A). \tag{5.4}$$

3. Als de uitkomst B *bestaat* dan geldt:

$$0 < P(B) \leq 1. \tag{5.5}$$

   Een kans moet dus altijd groter zijn dan nul voor alle elementen in de uitkomstenverzameling.

4. **De *of* Regel**: Als de uitkomsten A en B *wederzijds uitsluitend* zijn, ofwel als A plaatsvindt, dan kan B nooit plaats vinden, dan geldt:

$$P(A \text{ of } B) \equiv P(A \cup B) = P(A) + P(B). \tag{5.6}$$

   We mogen in dit geval de kansen dus optellen.

5. **De *en* regel**: Als de uitkomsten A en B onafhankelijk zijn, dus als je A een uitkomst is dan zegt dat niets over de kans op B, dan geldt:

$$P(A \text{ en } B) = P(A) \cdot P(B). \tag{5.7}$$

We gaan voor elk van deze regels een voorbeeld geven. We kijken hiervoor naar een kaartendek. De uitkomstenverzameling van een kaartendek is:

{*1♡,2♡,3♡,4♡,5♡,6♡,7♡,8♡,9♡,H♡,D♡,K♡,A♡,*
1♢,2♢,3♢,4♢,5♢,6♢,7♢,8♢,9♢,H♢,D♢,K♢,A♢,
1♠,2♠,3♠,4♠,5♠,6♠,7♠,8♠,9♠,H♠,D♠,K♠,A♠,
1♣,2♣,3♣,4♣,5♣,6♣,7♣,8♣,9♣,H♣,D♣,K♣,A♣}

Dit zijn in totaal 52 kaarten verdeeld over 2 kleuren: rood en zwart. We trekken in de volgende voorbeelden steeds 1 kaart.

Voorbeeld 1 - behoud van kans/complement regel:

- De kans om een harten 5 uit een dek kaarten te trekken is precies: P(*5♡*)= $\frac{1}{52}$.
- De kans om een *andere kaart dan een harten 5* te trekken is gelijk aan: 1-P(*5♡*) $= 1 - \frac{1}{52} = \frac{51}{52}$.
- De kans om een rode kaart te trekken is precies $\frac{26}{52} = \frac{1}{2}$ en is precies gelijk aan de kans om een zwarte kaart te trekken ($1 - \frac{1}{2} = \frac{1}{2}$).

Voorbeeld 2 - groter dan nul:

- Voor elke kaart in het dek is er een kans dat je hem trekt.

Voorbeeld 3 - de of-regel:

- De kans dat je een 3 of een 5 trekt is gelijk aan P(3)+P(5) $= \frac{1}{13} + \frac{1}{13} = \frac{2}{13}$.
- De kans dat je een 3 of een rode kaart trekt kunnen we niet zomaar optellen. Er bestaan ook rode kaarten met een 3.

Voorbeeld 4 - de en-regel:

- De kans dat je een 3 trekt die ook een rode kaart is kunnen we uitrekenen met:

$$P(\text{rood en } 3) = P(\text{rood}) \cdot P(3) = 1/2 \cdot 4/52 = 2/52.$$

  Er zijn maar twee rode 3 kaarten in het dek, dus dat klopt. Er zijn evenveel rode drie kaarten als zwarte drie kaarten en daarom mag je ze in dit geval vermenigvuldigen. De uitkomsten zijn onafhankelijk.

- De kans dat je een 9♡ en een A♣ trekt. Deze kansen zijn niet onafhankelijk. Als je een 9♡ trekt, zegt dat al direct iets over de kans dat deze kaart ook een A♣ is (die is namelijk gereduceerd tot 0). Hier kunnen we de en-regel dus niet toepassen.

MODULE I Hoofdstuk 6

Kansdichtheidsfuncties

We gaan nu kijken naar kansverdelingen. In het voorbeeld van de simpele dobbelsteen zou je kunnen kijken hoe de kansen verdeeld zijn over de verschillende uitkomsten. Voor een normale dobbelsteen is dit misschien een beetje saai, voor elke uitkomst verwacht je een andere waarde. Voor de speciale dobbelsteen die we eerder beschreven ziet het er al wat interessanter uit.

Om over kansverdelingen te kunnen leren moeten we eerst weten wat stochasten zijn. Daarna introduceren we enkele veelgebruikte kansdichtheidsverdelingen.

6.1 Wat is een stochast?

Een **stochast** is een (meetbare) variabele waarvan de waarde van een kans proces afhangt. Bijvoorbeeld de uitkomst van het trekken van een kaart, dan is het getrokken kaart (de uitkomst van de trekking) een stochast. Je weet van tevoren niet welke kaart je gaat trekken en daarom is de uitkomst *stochastisch*. Of als je een met een dobbelsteen gooit dan is de uitkomst van de worp een stochast. Het Engelse woord (random variable) is misschien bekender.

6.2 Kansdichtheidsfuncties

Stochasten zijn een handig middel bij het beschrijven van experimenten. We gaan hieronder een aantal vaak voorkomende distributies van stochastische variabelen bekijken. De distributies laten zien wat de kans is dat een bepaalde stochastische waarde wordt gevonden. Het is dus een verdeling van kansen. Deze verdelingen noemen we **kansdichtheidsfuncties** (Engels: probability density function of PDF). Een kansdichtheidsfunctie, $f(x)$, zegt dat de kans dat een variabele x gevonden wordt in een gebied $[x, x + dx]$ gelijk is aan $f(x)dx$.
De kans dat we x terugvinden in een interval $[a, b]$ is gelijk aan:

$$P(a \leq x \leq b) = \int_a^b f(x)dx. \tag{6.1}$$

Er zijn **twee belangrijke voorwaardes** aan een kansdichtheidsfuncties die je misschien bekend zullen voorkomen:
1. De kans kan nergens kleiner dan nul zijn in het uitkomstengebied.
2. De kansdichtheidsdistributie moet genormaliseerd zijn op 1.
In formule notatie: $f(x) \geq 0$ en $\int_{-\infty}^{\infty} f(x)dx = 1$.

Wellicht komt dit allemaal wat abstract over en helpt het om wat concrete voorbeelden te zien. Hieronder definiëren we vier belangrijke kansdichtheidsfuncties. Er zijn veel meer kansdichtheidsfuncties gedefinieerd, kijk bijvoorbeeld maar eens naar deze lijst op Wikipedia.

Voor we gaan kijken naar de voorbeelden is het handig om uit te leggen hoe we de verwachtingswaarde en de standaardafwijking kunnen uitrekenen voor kansdichtheidsfuncties. De definities hiervan heb je gezien in het hoofdstuk Basisbegrippen (Hfdst. 1), voor dichtheidsfuncties zien de formules er net iets anders uit dan voor datasets.

6.3 Verwachtingswaarde en standaardafwijking

Voor **discrete** verdelingen gelden de volgende vergelijkingen:

- de verwachtingswaarde: $\mu = E(x) = \sum_{i=1}^{N} x_i P(x_i)$,
- de variantie: $\sigma^2 = \sum_{i=1}^{N} (x_i - E(x))^2 P(x_i)$.

Voor **continue** verdelingen maak je gebruik van de volgende vergelijkingen:

- de verwachtingswaarde: $\mu = E(x) = \int_{-\infty}^{\infty} x f(x)dx$,
- de variantie: $\sigma^2 = E(x^2) - E(x)^2 = \int_{-\infty}^{\infty} (x - E(x))^2 f(x)dx$.

NB Herinner je nog het verschil tussen parameters (voor de kenmerken van een populatie) en statistieken (voor de kenmerken van een steekproef). Afhankelijk van wat we beschrijven zijn verschillende schrijfwijze voor het gemiddelde $\mu, \bar{x}$ en $E(x)$. Het symbool μ is meestal voorbehouden aan het gemiddelde van de populatie, dat wil zeggen het *echte* gemiddelde. Het gemiddelde van de steekproef is $\bar{x}$, je hoopt dus dat die dicht bij het populatie gemiddelde μ ligt. De verwachtingswaarde $E(x)$ is de waarde die je verwacht te gaan meten. Deze kan je met simulaties benaderen. De verschillen worden pas echt duidelijk als je er al een tijdje mee werkt. We zullen het niet fout rekenen als je een vergissing maakt in de notatie, maar we proberen het hier wel netjes op te schrijven. In deze vergelijkingen is het in elk geval ook gewoon handiger om $E(x)$ of $\bar{x}$ te schrijven. $E(x)^2$ is, net als $\bar{x}^2$, het kwadraat van de verwachtingswaarde van x. $E(x^2)$ is, net als $\overline{x^2}$ de verwachtingswaarde van x^2.

6.4 Bekende kansdichtheidsfuncties

6.4.1 Uniform

De uniforme distributie is een vlakke kansverdeling. De kans op elk deel van de uitkomstenverzameling is gelijk. We hebben hier al een paar voorbeelden van gezien. Bijvoorbeeld bij de eerlijke dobbelsteen waarbij de kans op elk van de 6 uitkomsten precies gelijk is. De uitkomsten van een dobbelsteen zijn discreet. Voor **discrete uniforme** verdelingen van stochastische waarden kunnen we schrijven dat de kans op uitkomst van stochast i, $P(i)$, gevonden kan worden met de relatie: $P(i) = 1/N$.
Waarbij N de hoeveelheid mogelijke uitkomsten is. Dit ziet er grafisch als volgt uit (zie Fig. 6.1 .

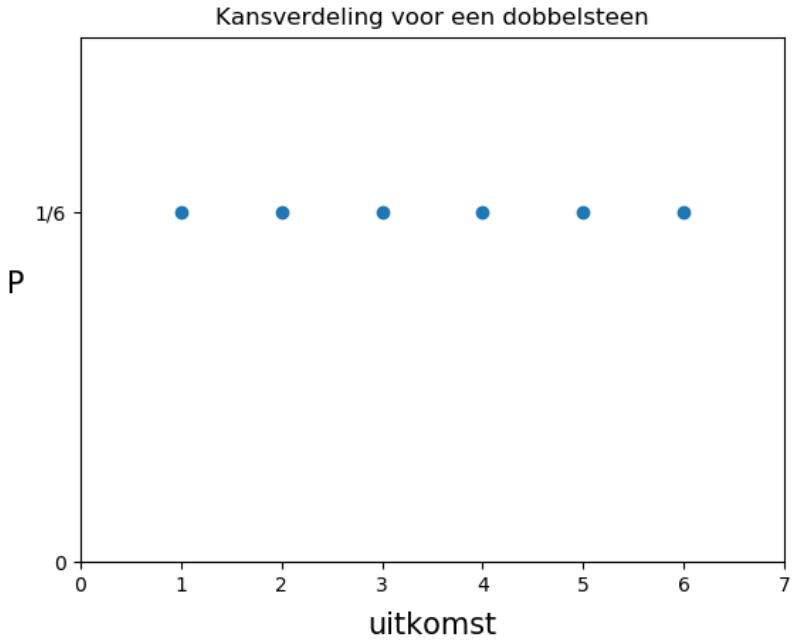


Figure 6.1: De kansverdeling van uitkomsten van een worp met een dobbelsteen.

Een algemene formule voor een **continue uniforme** verdeling is:

$$f(x; a, b) = \frac{1}{b-a} \quad \text{voor} \quad a \leq x \leq b. \tag{6.2}$$

Hierbij is $f(x)$ de kans dat je de waarde x vindt. De stochast is hier dus x. Hier in Fig. 6.2 zie je hoe de uniforme verdeling eruit ziet voor een continue verdeling.

De **verwachtingswaarde** kunnen we uitrekenen met behulp van de algemene formule:

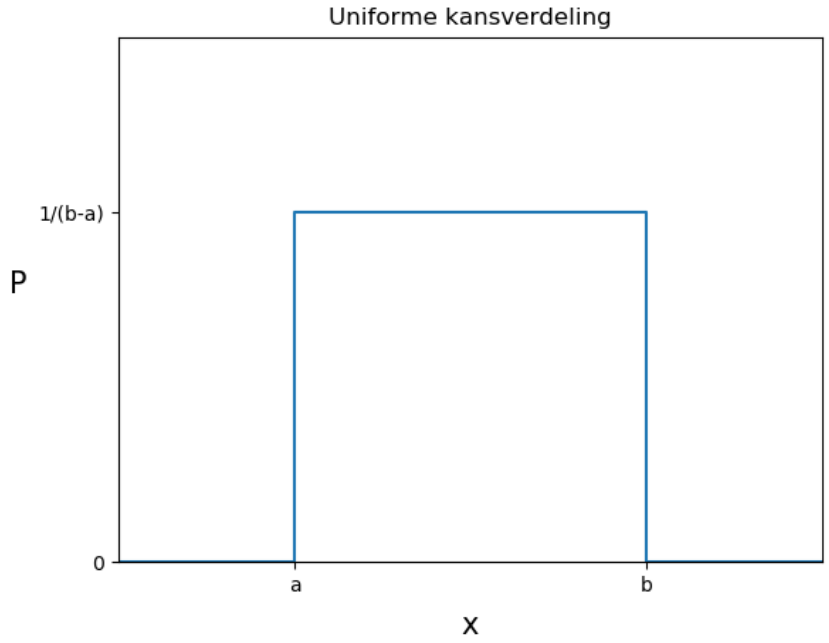


Figure 6.2: Een voorbeeld van de kansdichtheidsverdeling van een uniforme distributie.

$$\begin{aligned} E(x) &= \int_{-\infty}^{\infty} x f(x) dx \\ &= \int_{a}^{b} x \cdot \frac{1}{b-a} dx \\ &= \frac{1}{2} \frac{1}{(b-a)} x^2 \Big|_a^b \\ &= \frac{b^2 - a^2}{2(b-a)} \\ &= \frac{a+b}{2}. \end{aligned} \tag{6.3}$$

De **standaardafwijking** berekenen we met de formule:

$$\begin{aligned} \sigma^2 &= \int_{-\infty}^{\infty} (x - E(x))^2 f(x) dx = \int_{a}^{b} \left(x - \frac{a+b}{2} \right)^2 \cdot \frac{1}{b-a} dx \\ &= \frac{1}{12} \cdot \frac{(b-a)^3}{b-a} = \frac{(b-a)^2}{12}. \end{aligned} \tag{6.4}$$

Dit geeft de vergelijking voor de standaardafwijking:

$$\sigma = \frac{(b-a)}{\sqrt{12}}. \tag{6.5}$$

6.4.2 Binomiaal

Om de binomiale verdelingsfunctie uit te leggen beginnen we eerst met het Bernoulli-experiment. Dit is een experiment met maar twee uitkomsten, 'succes' en 'mislukking'. De kans op succes is p en de kans op mislukking q, is dan dus $q = 1 - p$.

Als we precies n onafhankelijke Bernoulli experimenten uitvoeren dan is de kans op een totaal aantal malen succes uit deze n experiment gedefinieerd als k. Dit wordt beschreven door de binomiale verdeling:

$$P(k; n, p) = \left(\begin{array}{c} n \\ k \end{array} \right) p^k (1-p)^{n-k} \equiv \frac{n!}{k!(n-k)!} p^k q^{n-k}. \tag{6.6}$$

Het gemiddelde en de standaardafwijking van de Binomiale verdeling zijn:

$$E(k) = np \tag{6.7}$$

en

$$\sigma = \sqrt{npq} \tag{6.8}$$

.

Voorbeeld Stel dat we een oneindige grote verzameling knikkers hebben waarvan 30% gele knikkers, alle andere knikkers zijn rood gekleurd. Als we een enkele knikker trekken hebben we dus precies 30% kans ($p = 0.3$) dat dit een gele knikker is.
Als we twee knikkers trekken hebben we een kans van $0.3 \cdot 0.3 = 0.09$ dat we precies twee gele knikkers hebben getrokken. Immers heeft de eerste trekking geen invloed op de tweede trekking en zijn de twee trekkingen onafhankelijk. We mogen dus de 'en'-regel gebruiken.
We hebben een kans van $(1 - 0.3 \times 0.3) = 0.91$ dat we minstens 1 rode knikker hebben, hier gebruiken we de complement regel.
De kans dat we twee rode knikkers hebben (en dus geen gele knikkers) is

$$(1 - 0.3) \times (1 - 0.3) = 0.49.$$

We kunnen nu ook redeneren dat de kans dat we 1 gele knikker en 1 rode knikker hebben getrokken precies gelijk is aan

$$0.91 - 0.49 = 0.42.$$

We kunnen deze kansen ook met de Binomiaal vergelijking uitrekenen:

- 2 trekkingen, 0 gele knikkers:

$$P(k; n, p) = p(0; 2, 0.3) = \frac{2!}{(0! \cdot 2!)} 0.3^0 \cdot 0.7^2 = 0.49$$

- 2 trekkingen, 1 gele knikker:

$$P(k;n,p) = p(1;2,0.3) = \frac{2!}{1! \cdot 1!} 0.3^1 \cdot 0.7^1 = 0.42$$

- 2 trekkingen, 2 gele knikkers:

$$P(k;n,p) = p(2;2,0.3) = \frac{2!}{2! \cdot 0!} 0.3^2 \cdot 0.7^0 = 0.09$$

Deze kansen staan ook uitgedrukt in de gele lijn in de figuur hieronder.

De binomiale verdeling is een discrete verdeling. Deze formule kunnen we niet toepassen op fractionele waardes. Dat is ook logisch want het Bernoulli experiment kunnen we niet een fractioneel aantal keer uitvoeren. De kansverdeling is asymmetrisch voor lage waardes van n en wordt voor grotere waardes van n steeds meer symmetrisch.

In het figuur hier , in Fig. 6.3, zie je een aantal Binomiaalverdelingen.

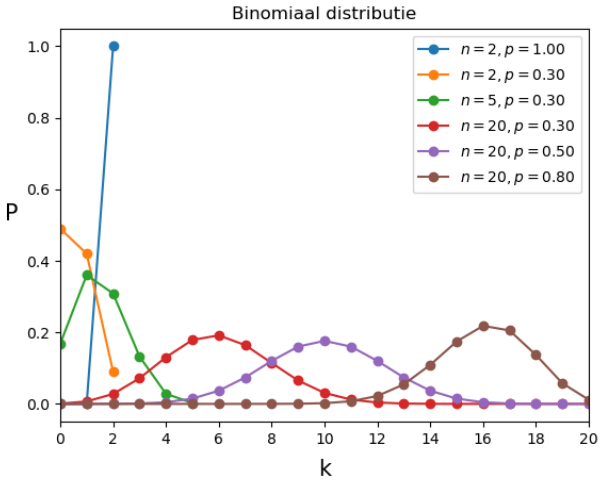


Figure 6.3: Binomiaal kansdichtheidsverdelingen.

Het voorbeeld van daarnet is uitgedrukt in de oranje lijn. Kijk ook eens goed naar de blauwe lijn. De kans $p = 1$ zegt dat de uitkomst altijd 'succes' is. Als je het experiment twee keer uitvoert, zijn ze dus gegarandeerd allebei succesvol. En de kans is 0 dat je maar 1 uit 2 ($n = 2, k = 1$) positieve uitslagen hebt. Dat kan immers ook niet, je kan alleen maar succes hebben, er bestaan in dit geval geen andere uitslagen van het experiment.

6.4.3 Poisson

De Poisson is een discrete verdelingsfunctie die, in veel gevallen, de onzekerheid weergeeft op telexperimenten. Het aantal geobserveerde gebeurtenissen (k) is gerelateerd aan het verwachte aantal gebeurtenissen (λ) via de Poissonverdeling:

$$P(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{6.9}$$

De Poisson kent, in tegenstelling tot de binomiaal dus maar 1 parameter. De verwachtingswaarde van de Poisson vergelijking (het gemiddelde) is λ en de variantie is ook λ. De onzekerheid op een stochast, als deze de Poisson statistiek volgt, is gelijk aan de standaardafwijking: $\sigma = \sqrt{\text{var}} = \sqrt{\lambda}$.

Het is dus een bijzondere vergelijking! In het figuur hier , Fig. 6.4 zie hoe de Poisson distributie eruit ziet voor verschillende waardes van λ.

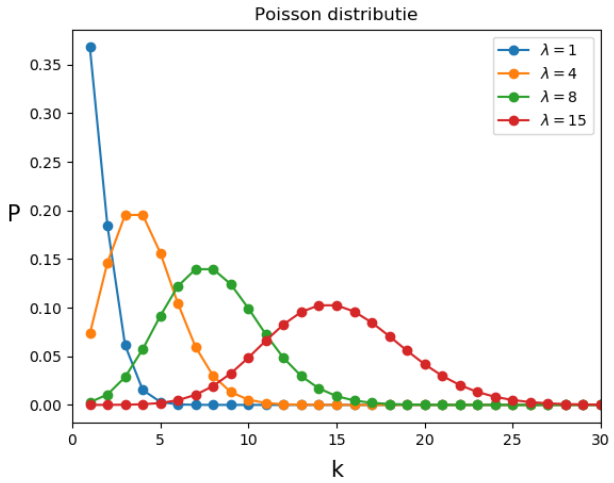


Figure 6.4: Poisson-verdeelde kansdichtheidsdistributie.

De Poisson verdeling is, net als de Binomiaal vergelijking asymmetrisch voor lage waardes van λ en wordt steeds meer symmetrisch voor hogere waardes van λ. Dat is ook geen toeval, de Poisson vergelijking is een speciale vorm van de Binomiaal. Als je hier meer over wilt weten kun je dit filmpje bekijken.

6.4.4 Normaal (ofwel Gauss)

Stochastische variabelen zijn Normaal-verdeeld (ook wel Gaussisch) als ze door de volgende functie worden beschreven:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}. \tag{6.10}$$

De functie heeft twee parameters, μ en σ, de notering is niet toevallig. De verwachtingswaarde van de normaal verdeling is precies μ en de standaardafwijking is precies σ.

Over de mathematische beginselen van de Normale verdelingsfunctie gaan we hier verder niet in. Het is wel goed om te weten dat de Normale verdelingsfunctie zonder twijfel de meest belangrijke functie is in de statische data analyse. De verdelingsfunctie komt erg vaak voor. Dat is geen toevalligheid, we zullen later in module 3 zien waarom dit zo is.

In het figuur hier , Fig. 6.5 zie je enkele voorbeelden van de Normale verdeling met verschillende waardes voor μ en σ.

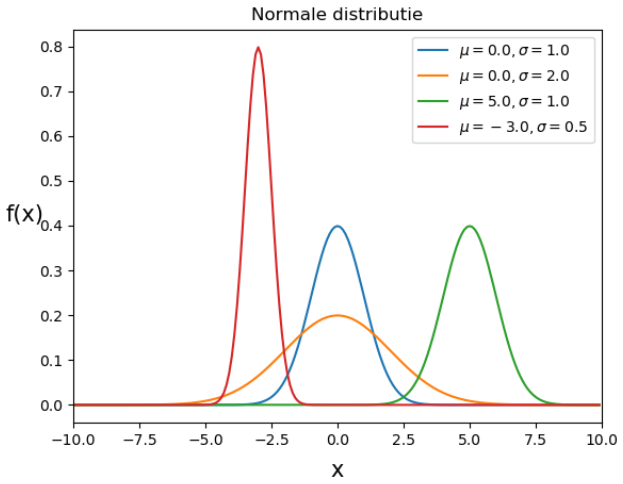


Figure 6.5: Normaal-verdeelde kansdichtheidsverdelingen.

Het is goed om op te merken dat de Normaalverdeling een symmetrische continue verdeling is. Bij de Normaalverdeling zijn de meeste uitkomst waardes gegroepeerd rond het gemiddelde en hoe meer we van het gemiddelde afwijken, des kleiner de kans is dat we een meetwaarde aantreffen.

Voorbeelden van Normaalverdelingen vinden we overal om ons heen. De verdeling van lichaamslengtes van mensen (of bijvoorbeeld olifanten), de grote van zandkorrels op een strand, de luminositeit van sterren in het melkwegstelsel.

MODULE I Hoofdstuk 7

Opdrachten Module 1

Tijdens de laptopcolleges in de eerste week werken we aan de vijf opdrachten van de eerste module.

- M1.1 Mooi Plotten * (Hfdst. 7.1)
- M1.2 Kansdichtheid-distributies ** (Hfdst. 7.2)
- M1.3 Eigenschappen van distributies ** (Hfdst. 7.3)
- M1.4 Grote Aantallen I *** (Hfdst. 7.4)
- M1.5 Halfwaardedikte I * (Hfdst. 7.5)

De sterretjes geven een indicatie voor hoeveel werk een opdracht is.

7.1 M1.1 - Mooi Plotten *

We beginnen dit vak met een eenvoudige opdracht. We gaan in deze opdracht een histogram goed leesbaar maken. Lees eerst het stuk je over Data visualiseren (Hfdst. 3), daar vind je ook de richtlijnen waaraan een goed histogram voldoet.

> Installeer de volgende twee bestanden naar een werkfolder op je computer `M1.1_MooiPlotten.py` en `DAS_DatasetGenerator.py`.

Open de bestanden in het VSCode programma en lees de code. We zullen hieronder wat uitleg over de code geven.

In `M1.1_MooiPlotten.py` wordt eerst een dataset aangemaakt met de volgende regel:

```
x = ds.DataSetMooiPlotten()
```

Helemaal boven in de code vind je de regel:

```
import DAS_DatasetGenerator as ds
```

en zo weet je dat `DataSetMooiPlotten()` een functie is die in `DAS_DatasetGenerator.py` is gedefinieerd. Om dit bestand te kunnen runnen moet je eerst je studentnummer invoeren in de DAS_DatasetGerator.

> Open het bestand `DAS_DatasetGenerator.py`, vind de **`student_nummer`** variabele in regel 12 en voer hier je studentnummer in.

Nu kun je het bestand `M1.1_MooiPlotten.py` runnen. Kijk goed naar je output.

Je hebt nu je eerste histogram gemaakt van jouw eigen dataset x. Een histogram is een manier om data te presenteren. Er bestaan 1-dimensionale, 2- en 3- dimensionale histogrammen. Lees hier (Hfdst. 3) meer over histogrammen.

We gaan nu het histogram zo maken dat de dataset x ook goed 'leesbaar' is. Vooralsnog is van het histogram niet heel duidelijk hoe de distributie van x eruit ziet. De bedoeling is dat als we naar het histogram kijken, we meteen een goed idee krijgen van de distributie van x.

> Voor we iets gaan veranderen in `M1.1_MooiPlotten.py`, voer eerst je naam en je studentnummer in de eerste regels van dat bestand.

Je moet de code uiteindelijk ook inleveren. We gebruiken deze bij het nakijken om, als er een fout is gemaakt in de opdracht te kunnen bekijken waar die vandaan komt. Het is natuurlijk ook de bedoeling dat de resultaten die je inlevert overeenkomen met de resultaten uit je programmaatje. Daar kijken we ook naar.

In `M1.1_MooiPlotten.py` vind je de volgende regel code:

```
plt.hist(x, bins=31, range=(-100,100))
```

Dit is de regel code waar het histogram wordt aangemaakt. Als je helemaal boven in de code kijkt zie je de volgende regel:

```
import matplotlib.pyplot as plt
```

De **`hist`** functie wordt dus gedefinieerd in de **`matplotlib`** library. Dat is handig om te weten als je meer over deze library te weten wilt komen. Je kan bijvoorbeeld veel vinden over de verschillende plot mogelijkheden door op "matplotlib" en "hist" te zoeken op het web.

Je ziet dat behalve de dataset x er ook twee opties worden meegegeven (**`bins`** en **`range`**). De range geeft aan welk bereik de x-as van het histogram heeft. De andere variabele, bins, geeft aan in hoeveel delen deze x-as is opgedeeld.
Deze twee opties kun je eventueel weglaten. In dat geval zoekt python zelf, met behulp van een algoritme een range en aantal bins uit. Je zult zien dat dat niet altijd optimaal werkt.

Probeer nu de default binning van de hist functie uit door de `bins` en `range` opties weg te laten. Kijk goed naar de waardes op de x-as en waar de kolommen precies starten en ophouden.

Ook in dit geval is de representatie van de dataset niet optimaal. We gaan dus de range en de binning zelf optimaliseren. De bedoeling is dat het histogram goed interpreteerbaar wordt. Door te fijne binning (veel bins in een kleine range) wordt de dataset heel grillig, het wordt dan lastig om de distributie te herkennen en trends goed te kunnen zien. Hetzelfde gebeurt als de binning te grof is. Er bestaat meestal niet een enkele goede instelling maar een gebied waarin het goed uitpakt.

Pas nu de binning en de range aan zodat de distributie van x goed zichtbaar is. Let hierbij goed op of de binning niet te grof of te fijn is.

Het is makkelijker om eerst de range goed af te stellen en dan pas de binning waarde te veranderen. Als je meer controle wilt hebben over waar de assen beginnen en eindigen kun je gebruik maken van de `plt.xlim(min,max)` functie.

Voeg de `plt.xlim(min,max)` functie toe aan je code (voor het `show` commando) en kies geschikte waardes. Stem hierna nogmaals de binning af.

Als je een goede binning en range combinatie hebt gevonden waarin de kenmerken van de distributie goed zichtbaar zijn, is het goed om nog een keer te kijken naar de leesbaarheid van de x-as. Het is prettig als de bins een eenvoudig te lezen fractie hebben van de streepjes op de x-as. Dus als je een range hebt van 2 tot 12, is het onhandig om die in 13 stukjes op te delen. Prettiger is bijvoorbeeld 2, 4, 5 of 10. Dat is het histogram eenvoudiger leesbaar. Misschien heb jij wel veel meer bins nodig, of een grotere range.

Pas als laatste nog de binning aan zodat ook de bin-breedtes eenvoudig zijn af te lezen - maar zorg dat de kenmerken van de distributie van x ook goed zichtbaar blijven.

TIP: Als je het lastig vindt om te begrijpen wat 'kenmerken' zijn van de distributie is het het handigste om even te spelen met de waardes en goed naar de data te kijken. Het is nooit voorspelbaar hoe een dataset eruitziet (en dus wat de belangrijke kenmerken zijn). Typisch wil je wel weten hoe breed de verdeling is, waar hij begint en waar hij ophoudt.

Ook als deze bijvoorbeeld asymmetrisch is, is het wel belangrijk dat dat zichtbaar is in het histogram.

Als je tevreden bent met het resultaat kun je het histogram opslaan door de laatste regel code te activeren:

```
plt.savefig('M1.1_MooiPlotten.png')
```

Open nu het inlevertemplate voor Module 1 en beantwoord de volgende twee vragen.
M1.1a) Plaats hier je histogram.
M1.1b) Valt je iets op aan de data? Probeer het te omschrijven.

Als je alle andere opdrachten voor module 1 ook hebt gemaakt kun je de opdrachten en code inleveren via de ANS website. Bij de laatste opgave voor module 1 vind je hier meer informatie over. Zorg in elk geval dat je de code die je hier hebt gemaakt bewaard.

Let op! Je mag dus niet de automatische binning gebruiken van `matplotlib`; Je moet expliciet de `bin` en `range` opties gebruiken in je code.

7.2 M1.2 - Kansdichtheid distributies **

We gaan in deze opgave kijken naar kansdichtheid distributie. Lees hier (Hfdst. 6) meer over kansdichtheidsdistributies. Er zijn een paar belangrijke en bekende distributies. We gaan in deze opgave aan de slag met de poissonen uniformedistributies.

7.2.1 Poisson distributie

M1.2a) Reken (met de hand) de volgende Poisson kansen uit: $P(k = 1, \lambda = 3)$**,** $P(k = 2, \lambda = 3)$ **en** $P(k = 3, \lambda = 3)$**. Kijk goed wat** λ **en** k **eigenlijk betekenen en wat de verwachtingswaarde is, en wat de geobserveerde waarde. Schrijf niet alleen het antwoord op maar begin bij de formule en werk het dan uit. Let ook op de regels van de notatie. Bekijk hiervoor het stukje over significantie in het hoofdstuk notatie (Hfdst. 2).**

De Poisson distributie is één van de belangrijkste distributies. We zullen hem vaak tegen gaan komen. We gaan nu Poisson distributies met python grafisch weergeven. Download het bestand `M1.2_Distributies.py`.

Maak eerst een functie die de Poisson kans uitrekent. De bedoeling is dat je de functie k en λ meegeeft en deze de Poisson kans teruggeeft. In het bestand vind je al een lege functie die je kunt invullen.

TIP De macht, de exponentieel en de faculteit die in de formule voorkomen kun je makkelijk uitrekenen met het `math` pakket in python.

```
import math as math

math.pow(lamda,k)   ## dit geeft lamda^k
math.exp(-lamda)    ## geeft e^{-lamda}

math.factorial(k)   ## geeft k!
```

- Reken nu de kansen $P(k = 1, \lambda = 3)$, $P(k = 2, \lambda = 3)$ en $P(k = 3, \lambda = 3)$, uit met je python functie. Komt het overeen met de waardes die je eerder met de hand berekende? Check het resultaat.
- Maak vervolgens een Poisson-kansdichtheidsdistributie voor $\lambda = 5$. Doe dit door eerst een lijst aan te maken met x waardes tussen 1 en 40 (met stapjes van 1) en vervolgens voor elk punt de kans uit te rekenen met de functie en deze op te slaan in een lijst. Maak vervolgens een grafiek met de resultaten. Zorg dat je grafiek er netjes uitziet. Ziet de grafiek eruit zoals je had verwacht?
- **M1.2b) Maak nu een grafiek waarin je de Poisson distributies voor $\lambda = 2, 5, 10$ en 20 laat zien. Maak 1 grafiek met de 4 resultaten. Zorg dat je de grafiek leesbaar maakt, bekijk hiervoor de richtlijnen in het hoofdstuk data visualisatie (Hfdst. 3).**
- **M1.2c) Lees af in je grafiek hoe groot de kans is om een waarde van 4 te vinden voor de vier verschillende verwachtingswaarden. NB je kan het natuurlijk ook uitrekenen met je functie.**

7.2.2 Uniforme distributie

We gaan nu kijken naar de uniforme distributie en simuleren een experiment dobbelstenen gooien. We gaan ervan uit dat we een zuivere dobbelsteen hebben die precies gelijke kansen heeft om op een willekeurige vlak terecht te komen.

Beantwoord de volgende vragen:

- **M1.2d) Als je een dobbelsteen eenmaal gooit, wat is dan de kans dat je een 1 gooit?**

En wat is de kans dat je een 4 of lager gooit?

- **M1.2e) Stel dat je 30 keer met de dobbelsteen gooit. Wat is dan het verwachte aantal keren dat je 3 gooit?**
- **M1.2f) Als je dit experiment doet en je gooit wel 10 keer een 3, kun je daaruit dan concluderen dat je een niet eerlijke dobbelsteen hebt?**

We gaan dit experiment nu simuleren. Om stochastische (ook wel toevallige of random) getallen te genereren maken we gebruik van de random nummer generator in het `numpy` pakket. Er zijn verschillende manieren om een dataset te maken die het experiment simuleert. In elk geval heb je de volgende pakketten nodig:

```
import numpy as np
import random
from random import seed
```

Zet nu de `seed` van de random nummer generator op 1. Dit kan je zien als het startpunt waarvandaan het algoritme begint. Zo is de simulatie herhaalbaar onder steeds dezelfde condities. Dat wil zeggen dat wanneer het programma opnieuw runt, elke keer dezelfde random getallen worden gegenereerd

```
np.random.seed(1)    # dit zet de seed in de random generator op 1.
```

Er zijn verschillende functies die je kan gebruiken. Kijk eens naar de `uniform()` en de `randint()` functies van random en bedenk een manier om je simulatie te schrijven.

- Genereer nu een dataset waarin je simuleert dat je 30 keer met een dobbelsteen gooit.
  TIP Als je het nodig hebt: gebruik `(int)` om naar een natuurlijk getal af te ronden. Controleer dat je alle getallen in de set {1,2,3,4,5,6} kunt maken.

- **M1.2g) Plot de waarden in je dataset in een histogram. Kijk naar de code in opgave M1.1 om te zien hoe je een histogram maakt. Let goed op de binning en range van je histogram. Als deze niet in orde zijn krijg je de verkeerde indruk van de dataset. Controleer je histogram desnoods door je dataset uit te printen en met de hand te tellen of je de juiste hoeveelheid in de juiste bin hebt. Kijk goed naar de richtlijnen en maak je histogram helemaal netjes.**
- **M1.2h) Komt de distributie overeen met je verwachting? Motiveer dit.**

- **M1.2i) Waarschijnlijk komt niet elke mogelijke uitkomst evenveel keer voor in je dataset. Hoe verwacht je dat de uitkomsten verdeeld zijn? Als je het experiment opnieuw zou doen, wat is dan de kans dat je maar 1 keer een 6 zult gooien. Gebruik hiervoor de Poisson kans(deze kans moet je berekenen en kun je niet aflezen uit je histogram).**

Verzamel je antwoorden op het template voor module 1, zorg dat je de code ook bewaart, deze moet je later inleveren.

7.3 M1.3 - Eigenschappen van distributies **

In deze opdracht gaan we kijken naar de eigenschappen (Hfdst. 1) van distributies en hoe deze veranderen als je een translatie of vermenigvuldiging toepast. We kijken naar de Normaal en Poisson distributies.

Download voor deze opdracht het bestand `M1.3_Eigenschappen.py` zorg dat deze in dezelfde folder staat as het `DAS_DatasetGenerator.py` bestand.

7.3.1 Normale distributie

We beginnen met het maken van een Gaussische dataset dg(x) met 500 punten. Deze maken we aan met de functie `genereerDistributieDG(N)` waarbij `N` het aantal datapunten is die we willen genereren. We kiezen voor een dataset met 500 punten.

```
    dg = ds.genereerDistributieDG(500)
```

Deze regel code vind je in het `M1.3_Eigenschappen.py` bestand.

Schrijf een functie die de volgende statistieken uitrekent voor de dataset:
- het gemiddelde
- de mediaan
- de variantie
- de standaardafwijking

NB: Het is de bedoeling dat je de formules zelf programmeert. Je mag geen gebruik maken van standaard functies van python die dit direct voor je teruggeven. Uiteraard mag je wel gebruiken maken van functies als `len()` en `sort()`.
Let wel op dat als je de functie `sort()` gebruikt bij een lijst, dat de oorspronkelijke volgorde van de lijst daarna weg is. Dat is vaak niet zo handig als je met data bezig bent. Gebruik daarom liever de `sorted()` functie en maak een kopie van de data:

```
p = [3,2,4,1]
l = sorted(p)
```

`l` is nu de gesorteerde lijst `[1,2,3,4]`.

M1.3a) Stel nu dat je de dataset vergroot en dat je niet 500 maar 1000 meetwaardes hebt in je set. Wat denk je dan dat er gebeurt met elk van deze statistieken? Schrijf hier eerst op wat je verwacht, kwantificeer het resultaat waar het kan.

We gaan nu kijken naar het effect van een translatie van de dataset.

- Kopieer de originele dataset en manipuleer de waardes in de dataset met de volgende translatie:
$$x' = x + 2$$

- Plot daarna de originele en de getransleerde dataset over elkaar heen. Controleer of de punten inderdaad zijn opgeschoven.

- Welke van de eigenschappen verwacht je dat er veranderen? Controleer dit door voor de originele en getransleerde dataset alle variabelen uit te rekenen.

Nu gaan we kijken naar het effect van een vermenigvuldiging van x.

- Kopieer de originele dataset en manipuleer de waardes met de volgende multiplicatie:
$x' = 2x$

- Plot nu de gemultipliceerde dataset toe aan je plot zodat je de originele, de translatie en multiplicatie in 1 figuur ziet.

- **M1.3b) Maak nu één plot waar de drie histogrammen voor de normaalverdeling te zien zijn. De originele, de translatie en de multiplicatie. Zorg dat de histogram goed leesbaar is en kijk hiervoor nog eens naar de richtlijnen.**
TIP: gebruik de plot optie `alpha` om de histogrammen doorzichtig te maken.

Dit helpt bij het zichtbaar maken van de overlappende gebieden. `plt.hist(dg, alpha=0.6)`

- **M1.3c) Maak een tabel met de vier berekende statistieken voor de 3 normaalverdelingen. Let goed op de notatie.**
- **M1.3d) Welke van de statistieken veranderen en hoe?**

7.3.2 Poisson

We gaan nu kijken wat het effect is van translatie en multiplicatie op een Poisson distributie $dp(k)$. De Poisson distributie krijg je door de volgende functie aan te roepen.

```
dp = ds.genereerDistributieDP(500)
```

Herhaal nu de vragen b-d voor de Poisson verdeling:

- **M1.3e) Maak nu een plot waar de drie histogrammen voor de Poisson verdeling te zien zijn. De originele, de translatie en de multiplicatie. Zorg dat de histogram goed leesbaar is en kijk hiervoor nog eens naar de richtlijnen.**
- **M1.3f) Maak een tabel met de vier berekende statistieken voor de 3 Poisson verdelingen. Let goed op de notatie.**
- **M1.3g) Welke van de statistieken veranderen en hoe?**

7.4 M1.4 - Grote Aantallen ***

We hebben een enorme ton kogeltjes en we willen weten hoe zwaar een enkele kogel uit de ton is. De kogels zijn, door variaties in het productieproces, niet allemaal precies even zwaar. De massa's van de kogels zijn **Normaal** ofwel **Gaussisch** verdeeld. We willen graag weten wat de ***typische*** massa is van een kogel uit deze ton. Er zit ook een onzekerheid op de meting, maar die is veel kleiner dat de variatie in de kogelmassa's en mogen we negeren.

Het is te veel werk om alle kogels apart te wegen, dus we nemen een steekproef. We nemen eerst een enkele kogel en wegen die. Omdat we niet weten wat de spreiding is in de massa van de kogels, kunnen we nu ook nog niet weten hoe representatief de massa van deze enkele kogel is voor de gemiddelde massa.

We doen daarom nog een tweede meting. Nu kunnen we de resultaten van deze twee metingen vergelijken en een eerste schatting doen van de onzekerheid op de gemeten waardes. Deze schatting op de spreiding van kogel massa's is natuurlijk nog erg onnauwkeurig. We weten niet hoe groot de fout is op de grootheid die we willen meten, namelijk de ***typische*** massa.

We herhalen het experiment daarom nog een paar keer en elke keer kijken we naar het gemiddelde van de metingen die we hebben gedaan. Uiteindelijk kunnen we een de distributie van de metingen bekijken en bepalen wat de standaardafwijking is van de verdeling. Nu hebben we eindelijk een maat voor de nauwkeurigheid van een enkele meting.

Doordat we nu eigenlijk heel veel metingen hebben genomen is de nauwkeurigheid op het gemiddelde, en zo de nauwkeurigheid van de grootheid die we wilden bepalen een heel stuk verbeterd. Intuïtief snappen we dat hoe meer kogeltjes we wegen uit de ton hoe beter we weten wat het gemiddelde is van de kogeltjes in de ton.

De Wet van de Grote Aantallen zegt dat als we een verdeling hebben van random (stochastische) waardes, en deze verdeling een mathematisch goed gedefinieerd gemiddelde heeft, dat het gemiddelde van een steeds grotere dataset uiteindelijk convergeert. Dit betekent dus dat als de dataset aan de voorwaarde voldoet, we een steeds nauwkeuriger beeld hebben van wat het gemiddelde van de data is. We komen hier later nog uitgebreid op terug.

We gaan onze metingen nu simuleren om een gevoel te krijgen hoe de wet van grote aantallen werkt.

Download het volgende bestand in je werkfolder op de computer: `M1.4_GroteAantallen.py`. Zorg dat dit bestand in dezelfde folder staat als de `DAS_DatasetGenerator.py` file die je in opgave M1.1 al hebt gebruikt.

In `M1.4_GroteAantallen.py` bestand zie je eerst een aantal functies (`berekenGemiddelde()`, `maakSetGemiddelde()`) die gaan we **later pas** gebruiken.

Eerst kijken we naar de regel:

```
set_gauss = ds.DataSetGroteAantallen(),
```

Hier wordt de dataset met de **gemeten kogel massa (in grammen)** aangemaakt waarbij de elementen een normaalverdeling volgen. We gaan eerst kijken naar de gehele dataset.

- **M1.4a) Laat zien dat de waardes in de dataset een normaalverdeling volgen. Doe dit door de waardes te plotten in een *histogram*. Zorg dat het histogram er netjes uitziet en dat je de as-labels ook aanmaakt. Kijk eventueel naar de code van opgave M1.1 om te zien hoe je dat moet doen.**

- **M1.4b) Reken het gemiddelde, $\bar{m}$, en de standaardafwijking, s, uit van de gehele set metingen. Let bij het noteren van het resultaat op de notatieregels. Programmeer dit zelf uit (en maak dus geen gebruik**

van functies in python libraries die dit voor je kunnen doen). Je kan je functie uit opdracht M1.3 natuurlijk weer opnieuw gebruiken.

We gaan nu simuleren dat we steeds meer datapunten hebben in onze dataset. We willen weten wat het gevonden kogel massa is na $1, 2, 3...n$ metingen. Hoe verandert het gevonden gemiddelde van de dataset als we nog een extra kogel massa eraan toevoegen?

Voltooi nu eerst de functie `berekenGemiddelde(dataset,n)`. Als je de functie aanroept met n=2 dan is de bedoeling dat functie het gemiddelde uitrekent over de *eerste twee punten in de dataset*, bij n=3 over de eerste drie datapunten etc.

TIP: Controleer of de functie goed werkt door het resultaat van bijvoorbeeld n=4 met de hand na te rekenen.

Nu gaan we de functie `maakSetGemiddeldes()` afmaken. Deze functie geeft twee lijsten terug: ***n*** en ***gemiddeldes***. **n** loopt van 1 tot het aantal punten in de originele set_gauss dataset en ***gemiddeldes*** waarin steeds het gemiddelde over de eerste **n** meetwaardes in de dataset wordt berekend. Controleer of de functie goed werkt door bijvoorbeeld de gevonden gemiddeldes uit te printen.

- **M1.4c) Maak nu een grafiek met op de horizontale as n en op de verticale as de bijbehorende berekende gemiddelde waarde, $\bar{m}_n$. Als je nu de grafiek afleest bij n=2 dan is het de bedoeling dat je op de verticale as het gemiddelde afleest over de eerste twee punten van de dataset. Let goed op het goed leesbaar maken van de grafiek.**

- **M1.4d) Beschrijf in de grafiek wat er gebeurt. Is dit wat je verwacht had? Waarom wel of waarom niet?**

7.5 M1.5 - Halfwaardedikte I *

We voeren een experiment uit waarbij de halfwaardedikte van lood wordt bepaald voor een bepaalde gamma-bron. De bron produceert gamma straling met een onbekende energie. Een halfwaardedikte is gedefinieerd als precies de energie die je nodig hebt om de intensiteit van de bron te halveren. Dit is een belangrijke grootheid om te weten als je bijvoorbeeld ziekenhuispersoneel wil beschermen tegen straling die bij het maken van een röntgenfoto vrijkomt.

We hebben een opstelling gemaakt waarmee we steeds per tijdsinterval van 2 minuten de hoeveelheid straling meten met een Geiger-Müller telbuis. In de opstelling kunnen we plakken lood tussen de bron en de telbuis plaatsen. De plakken lood hebben een dikte van 0.3 cm. Als we twee loodplakken plaatsen is de totale dikte dus in totaal 0.6 cm lood. De afstand tussen de bron en de telbuis is constant. De achtergrondstraling is te verwaarlozen in dit experiment, net als de onzekerheid op de dikte van de loodplakken.

- **M1.5a) Welke kansverdeling volgt de onzekerheid op de telling van de Geiger-Müller telbuis? Als we bijvoorbeeld N counts hebben gemeten, hoe groot is dan de onzekerheid op de centrale waarde N? Geef de formule en beredeneer je antwoord.**

De intensiteit van de γ-bron, $I(d)$, hangt af van de dikte van het lood (in cm) dat tussen de bron en de telbuis is geplaatst. Deze volgt de volgende vergelijking:

$$I(d) = I_0 \times \left(\frac{1}{2}\right)^{d/d_{half}} \tag{7.1}$$

Hierbij is de intensiteit I gelijk aan het aantal counts per seconde:

$$I = \frac{N}{\Delta T} \tag{7.2}$$

We gaan het experiment nu simuleren. Download het bestand `M1.5_Halfwaardedikte.py` en zorg dat deze in dezelfde folder staat als het `DAS_DatasetGenerator.py` bestand.

De volgende regel in de code maakt de dataset aan:

```
counts,diktes = ds.DataSetHalfwaardeDikte()
```

Deze lijsten bevatten de meetwaardes (in counts) en de diktes lood (in cm) die tussen de bron en de telbuis zijn geplaatst. We gaan eerst de meetwaardes met foutenvlaggen in een grafiek plotten.

- Voor het plotten maak je een lijst aan met voor elk punt de onzekerheden op de gemeten aantal counts. De onzekerheden moet je dus zelf berekenen.
  Je kan nu met de volgende code de foutenvlaggen plotten.

plt.errorbar(diktes,counts, yerr=fouten, fmt = 'o', label='"data"')

- **M1.5b) Maak de grafiek met meetwaardes en foutenvlaggen. Let goed op de leesbaarheid van de grafiek, gebruik hiervoor de richtlijnen.**

We gaan nu de halfwaardedikte bepalen met de volgende methode. We kijken eerst naar het punt N_0 dus het aantal counts als er geen loodplakken zijn geplaatst. Nu zoeken we de eerste dikte, d, in de grafiek waarvoor geldt dat $N \leq 0.5 \times N_0$.

- **M1.5c) Bepaal nu met de hierboven beschreven methode de halfwaardedikte (in cm). Dit is natuurlijk makkelijk met de hand te doen maar programmeer het ook, dat hebben we in een latere opdracht nog nodig.**

Beantwoord nu de volgende vragen:

- **M1.5d) Hoe groot denk je dat de onzekerheid is op de bepaalde halfwaardedikte? Probeer dit te kwantificeren, schrijf niet alleen de geschatte waarde op maar leg ook uit hoe je tot die waarde bent gekomen.**

- **M1.5e) Kun je bedenken wat voor soort kans distributie zou de onzekerheid op de halfwaardedikte beschrijven? Leg uit hoe je tot je antwoord komt en als je het niet weet, beredeneer dan waarom je het niet weet.**

- **M1.5f) Is de methode om de halfwaardedikte te meten zuiver (Engels: unbiased), dat wil zeggen vind je niet steeds juist een te hoge of te lage waarde? Zo nee, waarom denk dat je dat dit niet zo is. Zo ja, kun je een manier bedenken om de onzuiverheid te verminderen?**

- **M1.5g) Stel dat de halfwaardedikte veel kleiner is dan de waarde die je nu gevonden hebt. Zou dit experiment dan nog hebben gewerkt? Wanneer wordt dit een probleem, kwantificeer je antwoord?**

- **M1.5h) Hoe zou je dit experiment willen verbeteren. Dit kunnen verbeteringen zijn aan de kant van de opstelling maar ook aan de kant van de data analyse. Noem een verbetering voor de opstelling en een voor de data analyse.**

MODULE II

Deze week verdiepen we ons in het concept meetonzekerheid. We leren hoe we onzekerheden kunnen doorrekenen (Hfdst. 8) in vergelijkingen. We zien hoe onzekerheden veranderen (Hfdst. 9) als we meer meetpunten aan onze dataset toevoegen. We kijken ook naar relaties tussen onzekerheden in meerdimensionale datasets (Hfdst. 10), en we introduceren extra kans rekenregels (Hfdst. 11) die vooral voor multidimensionale data belangrijk zijn. Daarna introduceren we de definitie van schatmethodes (Hfdst. 12)

MODULE II

Hoofdstuk 8

Foutenpropagatie

Vaak kunnen we de grootheid die we willen weten niet direct meten, maar meten we een observabele die zich via een bepaalde functie verhoudt tot de gezochte grootheid. Of meten we zelfs twee of meer variabelen die we nodig hebben om de gewilde grootheid te bepalen.

Dit is bijvoorbeeld het geval als we de gemiddelde snelheid van een auto willen bepalen. Dit zouden we kunnen doen door de tijd te meten die de auto nodig heeft om een bepaald traject af te leggen. We meten dan de door de auto gebruikte tijd, T en de lengte van het traject, L, en die zetten we dan om in snelheid via de bekende formule $v = L/T$. Of we bepalen bijvoorbeeld de massa van een elementair deeltje (in rust) en willen dit omzetten naar de energie van het deeltje via de formule $E = mc^2$.

Als we de onzekerheid (de standaardafwijking) weten op een gemeten grootheden dan kunnen we deze omzetten naar de grootheid die we eigenlijk willen bepalen. Dit noemen we het **propageren** van fouten. In dit hoofdstuk leren we je de basisregels voor het propageren van **ongecorreleerde** fouten. Dat wil zeggen dat als er meerdere onzekerheden worden gepropageerd deze onzekerheden onafhankelijk zijn; De meting van de ene observabele heeft geen invloed op de meting van de andere observabele; de fout die we maken in het meten van de ene grootheid hangt niet af van de fout die we maken op de andere gemeten grootheid.

Het is goed om alvast te beseffen dat er ook gecorreleerde fouten bestaan. Er zijn twee oorzaken voor het ontstaan van gecorreleerde fouten:

- Doordat er in de meting een correlatie is. Een voorbeeld van een gecorreleerde fout is als we een oppervlakte van een tafel willen weten, en we meten de lengte en de breedte met hetzelfde meetlint op. Als het meetlint een afwijking heeft waardoor we de lengte te groot opmeten, dan zullen we waarschijnlijk ook de breedte te groot opmeten.
- Doordat er een onderliggende parameter is waar beide gemeten grootheden vanaf hangen.

Hier behandelen we dus alleen ongecorreleerde fouten.

8.1 Basisregel

We beginnen met de **algemene regel voor het propageren van ongecorreleerde fouten**. Daarna zullen we laten zien hoe deze regel eruitziet voor eenvoudige relaties. Deze zou je apart kunnen leren, maar je kunt ook altijd de basisregel gebruiken. Het resultaat behoort hetzelfde te zijn. We noteren de onzekerheid op variabele x in dit hoofdstuk met Δx waar we eerder ook wel σ_x of s_x hebben gebruikt.

Als $q = q(x, y, z, \dots)$ een functie is met meerdere ongecorreleerde variabelen, dan wordt de onzekerheid op q gegeven door:

$$\Delta q = \sqrt{\left(\frac{\delta q}{\delta x}\Delta x\right)^2 + \left(\frac{\delta q}{\delta y}\Delta y\right)^2 + \left(\frac{\delta q}{\delta z}\Delta z\right)^2 + \dots} \tag{8.1}$$

Hierbij zijn $\frac{\delta q}{\delta x}$, $\frac{\delta q}{\delta y}$ etc. de partiële afgeleiden van q naar de betreffende variabele.

We zullen laten zien hoe deze formule werkt aan de hand van een paar voorbeelden.

Voorbeeld 1: Factor
Stel we hebben een vergelijking $y = c \cdot x$ met een standaardafwijking op x van Δx. Dan is de standaardafwijking op y, (Δy), gelijk aan:

$$\Delta y = \sqrt{\left(\frac{\delta y}{\delta x}\Delta x\right)^2} = c \cdot \Delta x.$$

In dit geval schaalt de onzekerheid op x (Δx) dus met dezelfde factor c tot de onzekerheid op y (Δy). In de grafiek hieronder (Fig. 8.1) wordt voor een willekeurige waarde x_i het effect van de propagatie van Δx rond de waarde x_i naar de fout Δy rond y_i visueel weergegeven. Je kunt duidelijk zien dat de grootte van Δy veranderd is met de factor c.

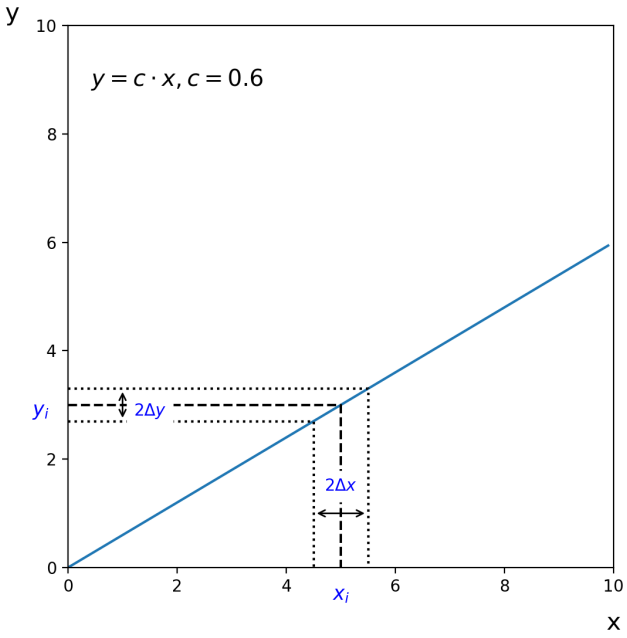


Figure 8.1: Visualisatie van de propagatie van Δx naar Δy voor een lineaire functie.

De onzekerheid op een meting is direct gerelateerd aan de standaardafwijking van de verwachtingswaarde van de te meten grootheid (de stochast). De variantie van de is zoals gebruikelijk het kwadraat van de standaardafwijking. In dit geval is dus de variantie op y:

$$Var(y) = (\Delta y)^2 = (c \cdot \Delta x)^2 = c^2 \cdot Var(x).$$

Voorbeeld 2: Translatie
Stel we hebben een vergelijking $y = x + a$ met een standaardafwijking op x van Δx. Dan is de standaardafwijking op y, (Δy), gelijk aan:

$$\Delta y = \sqrt{\left(\frac{\delta y}{\delta x}\Delta x\right)^2} = \Delta x.$$

Wederom geven we het effect van de foutenpropagatie van Δx rond x_i naar Δy rond y_i grafisch weer in de grafiek hieronder (Fig. 8.2) . Je ziet dat de translatie geen effect heeft op de grootte van de onzekerheid.

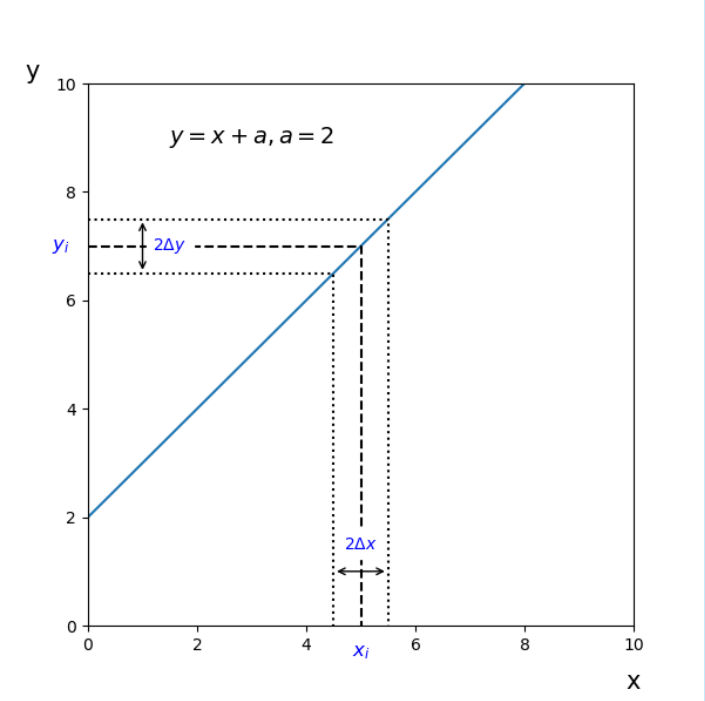


Figure 8.2: Visualisatie van de propagatie van Δx naar Δy bij een translatie.

Voorbeeld 3: Macht
Stel we hebben een vergelijking $y = x^3$ met een standaardafwijking op x van Δx. Dan is de standaardafwijking op y, (Δy), gelijk aan:
$\Delta y = \sqrt{\left(\frac{\delta y}{\delta x}\Delta x\right)^2} = 3x^2 \cdot \Delta x.$
Het effect van de foutenpropagatie volgens deze formule van Δx rond x_i naar Δy rond y_i wordt weer grafisch weergegeven in het plaatje hieronder (Fig 8.3) . Je kunt zien dat de mate waarin de grootte van Δx verandert afhangt van de gekozen waarde van x_i, op sommige plekken is hij kleiner geworden, op andere plekken groter.

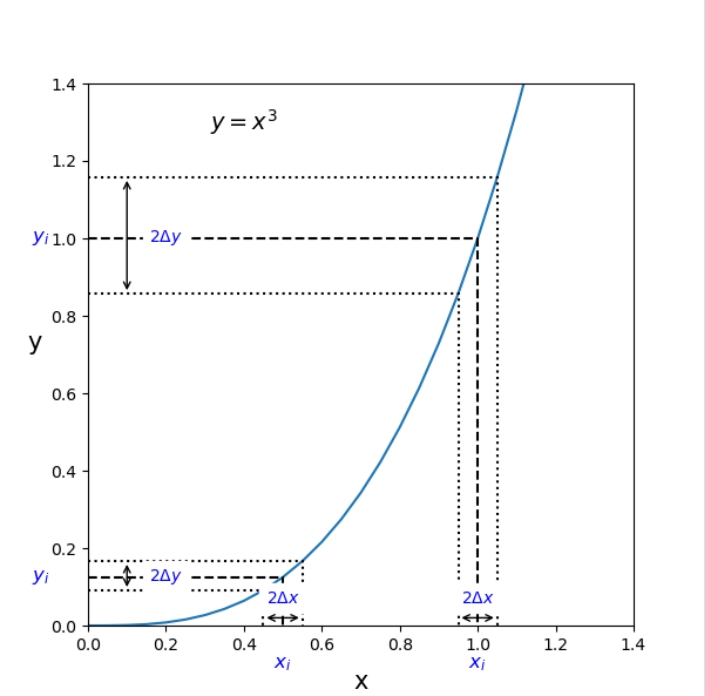


Figure 8.3: Visualisatie van de propagatie van Δx naar Δy bij een macht.

Voorbeeld 4: Parabool
Stel we hebben een vergelijking $y = ax + bx^2 + c$ met een standaardafwijking op x van Δx. Dan is de standaardafwijking op y, (Δy), gelijk aan:

$$\Delta y = \sqrt{\left(\frac{\delta y}{\delta x}\Delta x\right)^2} = (a + 2bx)\Delta x.$$

In het plaatje hieronder (Fig. 8.4) geven we nu voor verschillende waardes x_i de foutenpropagatie van Δx naar Δy de grafische interpretatie. We zien dat het niet alleen de relatieve grootte van Δy afhangt van de gekozen waarde van x_i maar dat op sommige plaatsen de boven en ondergrens van de onzekerheid zijn geïnverteerd.

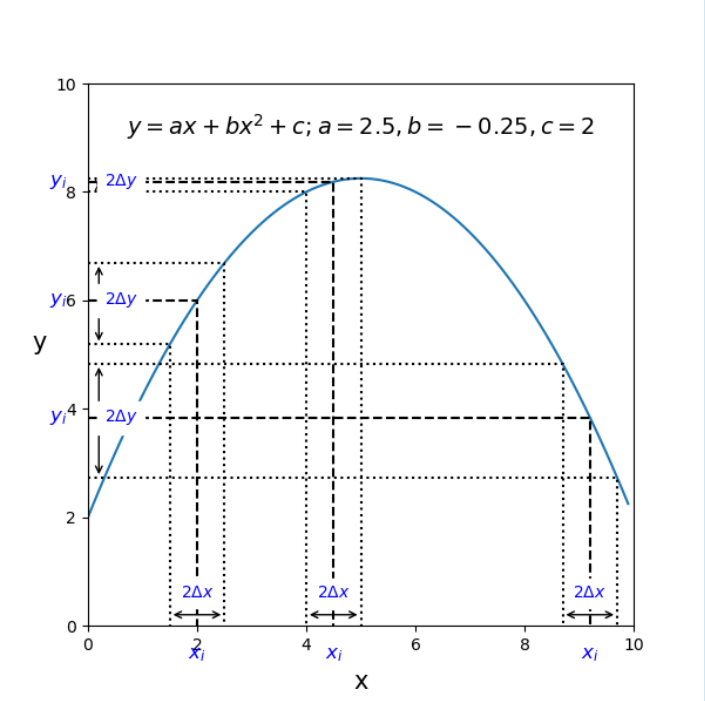


Figure 8.4: Visualisatie van de propagatie van Δx naar Δy voor een kwadratische vergelijking.

Voorbeeld 5: Twee dimensionaal
Stel we hebben een vergelijking $z = ax + y^2$ met standaardafwijkingen Δx en Δy . Dan is de standaardafwijking op z, (Δz), gelijk aan:

$$\Delta z = \sqrt{\left(\frac{\delta z}{\delta x}\Delta x\right)^2 + \left(\frac{\delta z}{\delta y}\Delta y\right)^2} = \sqrt{(a\Delta x)^2 + (2y\Delta y)^2}.$$

Voorbeeld 6: 2 dimensionaal
Stel we hebben een vergelijking $z = ax + y^2 + 2xy$ met standaardafwijkingen Δx en Δy . Dan is de standaardafwijking op z, (Δz), gelijk aan:

$$\Delta z = \sqrt{\left(\frac{\delta z}{\delta x}\Delta x\right)^2 + \left(\frac{\delta z}{\delta y}\Delta y\right)^2} = \sqrt{((a + 2y) \cdot \Delta x)^2 + ((2y + 2x) \cdot \Delta y)^2}.$$

8.2 Som en verschil

De algemene regel kan eenvoudig worden uitgeschreven naar de regel voor som en verschil. Als $q = x + y$ of $q = x - y$ dan wordt de onzekerheid op q gegeven door:

$$\Delta q = \sqrt{\left(\frac{\delta q}{\delta x}\Delta x\right)^2 + \left(\frac{\delta q}{\delta y}\Delta y\right)^2} = \sqrt{(\Delta x)^2 + (\Delta y)^2}. \tag{8.2}$$

We mogen de varianties $(\Delta x)^2$ en $(\Delta y)^2$ in het geval van een vergelijking met enkel sommen en/of verschillen dus optellen.

8.3 Vermenigvuldigen met constante

Als q het exacte veelvoud c is van de gemeten waarde x, dus $q = c \cdot x$, dan geldt:

$$\Delta q = \sqrt{\left(\frac{\delta q}{\delta x}\Delta x\right)^2} = |c|\Delta x. \tag{8.3}$$

De onzekerheid op q is dus gelijk aan de onzekerheid op x geschaald met dezelfde factor c.

8.4 Vermenigvuldigen en delen met variabelen

Als q een vermenigvuldiging is van meerdere variabelen, dus bijvoorbeeld $q = x \cdot y \cdot z$ dan geldt:

$$\Delta q = \sqrt{\left(\frac{\delta q}{\delta x}\Delta x\right)^2 + \left(\frac{\delta q}{\delta y}\Delta y\right)^2 + \left(\frac{\delta q}{\delta z}\Delta z\right)^2} = \sqrt{\left(\frac{q}{x}\Delta x\right)^2 + \left(\frac{q}{y}\Delta y\right)^2 + \left(\frac{q}{z}\Delta z\right)^2}. \tag{8.4}$$

Dit kan je eenvoudiger schrijven als:

$$\frac{\Delta q}{q} = \sqrt{\left(\frac{\Delta x}{x}\right)^2 + \left(\frac{\Delta y}{y}\right)^2 + \left(\frac{\Delta z}{z}\right)^2}. \tag{8.5}$$

Ofwel de relatieve fout $\frac{\Delta q}{q}$ is gelijk aan de kwadratische som van de variabelen.

Voorbeeld: foutenpropagatie en afronding van de getallen
Stel dat we de lengte van het blokje hebben gemeten en we lezen de volgende waarde af:

- De lengte (l) $= 7.60 \pm 0.10$ cm
- De breedte (b) $= 4.10 \pm 0.20$ cm

- De hoogte (h) $= 2.00 \pm 0.20$ cm

Het volume van het blokje wordt gegeven door:

$$V = l \cdot b \cdot h = 7.60 \cdot 4.10 \cdot 2.00 = 62.32 \text{ cm}^3$$

We gebruiken de regel dat als $q = x \cdot y \cdot \ldots$ dan:

$$\frac{\Delta q}{|q|} = \sqrt{\left(\frac{\Delta x}{x}\right)^2 + \left(\frac{\Delta y}{y}\right)^2 + \left(\frac{\Delta z}{z}\right)^2}$$

Dus:

$$\begin{aligned}\frac{\Delta V}{|V|} &= \sqrt{\left(\frac{\Delta l}{l}\right)^2 + \left(\frac{\Delta b}{b}\right)^2 + \left(\frac{\Delta h}{h}\right)^2} \\ &= \sqrt{\left(\frac{0.1}{7.6}\right)^2 + \left(\frac{0.2}{4.1}\right)^2 + \left(\frac{0.2}{2.0}\right)^2} \\ &= 0.01255\ldots\end{aligned}$$

We ronden dit nog niet af, dat doen we pas als we de absolute fout hebben:

$$\begin{aligned}\Delta V &= \frac{\Delta V}{|V|} \cdot |V| \\ &= 0.01255 \cdot \cdots \cdot 62.32 \\ &= 0.78228\ldots \\ &\approx 0.78\end{aligned}$$

Het gemeten volume van het blokje is dus $V = 62.32 \pm 0.78$ cm^3

MODULE II Hoofdstuk 9

Wet van Grote Aantallen

In opgave M1.4 hebben we gezien hoe de spreiding van het steekproef gemiddelde steeds kleiner wordt als we meer data gebruiken om het gemiddelde te bepalen. De gemeten waardes liggen steeds dichter bij elkaar. Dit is een belangrijke observatie. Het geeft aan dat hoe meer data we hebben, hoe nauwkeuriger we ons resultaat weten. Je voelt misschien al aan dat dit niet altijd op gaat. Wanneer dit wel en wanneer dit niet opgaat zullen we hier bespreken.

We bespreken hier twee regels, of wetten, de $\sqrt{n}$-wet en de wet van grote aantallen. De eerste wet zegt dat we een gemiddelde, onder bepaalde voorwaarden, steeds beter kennen als we meer datapunten meenemen. De tweede wet zegt dat het steekproef gemiddelde langzaam zal convergeren naar het gemiddelde van de populatie naarmate de steekproef steeds groter wordt.

9.1 De $\sqrt{n}$-wet

Stel dat we een grootheid willen weten die de som is van twee onafhankelijke variabelen die beide stochastisch verdeeld zijn. Dit betekent dat beide variabelen gemeten kunnen worden maar ook dat de waardes die we meten een onderliggende verdeling volgen die afhangt van een kansproces. We hebben hier dus twee onafhankelijke stochasten, die we X en Y noemen. De verwachtingswaarde van de som $X + Y$ is gelijk aan:

$$E(X + Y) = E(X) + E(Y). \tag{9.1}$$

Ofwel de verwachtingswaarde van de som is gelijk aan de verwachtingswaarde van X plus de verwachtingswaarde van Y. De verwachtingswaarde is hier niets anders dan het steekproefgemiddelde. Dus

$$E(X) = \frac{1}{n} \sum_{i}^{n} X_i. \tag{9.2}$$

We kunnen ook naar de spreiding van waardes van de som $(X + Y)$ kijken. Als X en Y onafhankelijk zijn dan geldt ook:

$$Var(X + Y) = Var(X) + Var(Y). \tag{9.3}$$

Het ziet er misschien ingewikkeld uit, maar het enige wat we doen is een nieuwe variabele definiëren die de som is van twee variabelen. De som van stochasten is zelf ook een stochast. De variantie op de som vinden we via de regels (Hfdst. 8) van de foutenpropagatie.

Stel nu dat we dit uitbreiden. En we nemen de som van n onafhankelijk stochasten, $X_1, X_2, ..., X_n$ die elk *dezelfde* onderliggende verdeling kennen. Dat wil zeggen dat ze allemaal dezelfde verwachtingswaarde en dezelfde variantie hebben. Je kan dit bijvoorbeeld zien als n onafhankelijke metingen van eenzelfde grootheid van een steekproef.

De formule voor de som S_n, kunnen we nu schrijven als:

$$S_n = X_1 + X_2 + ... + X_n. \tag{9.4}$$

En de verwachtingswaarde van S_n is dan:

$$E(S_n) = E(X_1 + X_2 + ... + X_n) = E(X_1) + E(X_2) + ... + E(X_n). \tag{9.5}$$

Omdat we eerder stelden dat elke stochast *dezelfde* onderliggende verdeling betekent dit dat

$$\mu_{X_1} = \mu_{X_2} = ... = \mu_{X_n} \equiv \mu. \tag{9.6}$$

Als de verwachtingswaarde (steekproefgemiddelde) van een enkele stochast $E(X_i)$ gelijk is aan het populatiegemiddelde μ dan geldt nu voor de verwachtingswaarde van de som:

$$E(S_n) = \mu \cdot n. \tag{9.7}$$

En als de variantie van de steekproef is gelijk aan de variantie van de populatie $Var(X_i) = \sigma^2$, dan geldt

$$Var(S_n) = n \cdot \sigma^2. \tag{9.8}$$

ofwel de standaardafwijking van de som is gelijk aan:

$$s_{(S_n)} = \sqrt{n} \cdot \sigma. \tag{9.9}$$

In plaats van naar de eigenschappen van de som S_n te kijken, kunnen we ook naar de eigenschappen van het gemiddelde van de stochasten X_i kijken. We hoeven hiervoor alleen maar de waarde van de som te delen door het aantal metingen n.

Behalve de som S_n kunnen we ook het gemiddelde van de stochasten, G_n, definieren. Dit gemiddelde is gedefinieerd als:

$$G_n = \frac{S_n}{n}. \tag{9.10}$$

De variabele n mogen we zien als een constante en daarom kunnen we gebruik maken van de regels die we in het vorige hoofdstuk hebben afgeleid (in het eerste voorbeeld)

$$E(cX) = c \cdot E(x), Var(cX) = c^2 \cdot Var(X). \tag{9.11}$$

De verwachtingswaarde van het gemiddelde G_n is dus gelijk aan:

$$E\left(G_n\right) = E\left(\frac{S_n}{n}\right) = \frac{E\left(S_n\right)}{n} = \frac{(n \cdot \mu)}{n} = \mu. \tag{9.12}$$

Precies wat we verwachten. De verwachtingswaarde van de steekproef is gelijk aan de verwachtingswaarde van de populatie. Voor de standaardafwijking vinden we

$$Var(G_n) = Var\left(\frac{S_n}{n}\right) = \frac{Var\left(S_n\right)}{n^2} = \frac{n \cdot \sigma^2}{n^2} = \frac{\sigma^2}{n}. \tag{9.13}$$

Dit betekent dat **de standaardafwijking voor het gemiddelde** G_n kan worden geschreven als

$$s_{(G_n)} = \frac{\sigma}{\sqrt{n}}.$$

Dit is een belangrijk resultaat. Het zegt dat we het gemiddelde van een steekproef steeds beter kennen als we meer metingen verrichten.

Denk bijvoorbeeld aan de ton met N kogels waarvan de massa's van de kogels een Normale distributie hebben met een gemiddelde μ en een standaardafwijking σ, de onzekerheid op het bepaalde gemiddelde massa van een steekproef gelijk is aan $\sigma/\sqrt{n}$. Hoe meer kogels we wegen en meenemen in het berekende steekproefgemiddelde, hoe nauwkeuriger we dit gemiddelde kennen.

9.2 De wet van Grote Aantallen

Intuïtief voelen we aan dat hoe meer metingen we doen, hoe meer informatie we hebben, en hoe nauwkeuriger ons resultaat is. We hebben in de $\sqrt{n}$-wet al gezien dat de standaardafwijking op een gemeten stochast afneemt met $1/\sqrt{n}$. We laten nu zien dat we, in de meeste gevallen, ook kunnen verwachten dat de gemeten steekproefgemiddelde steeds meer in de buurt komt van het populatiegemiddelde.

De **wet van grote aantallen** zegt dat het berekende steekproef gemiddelde, $\overline{X}$, van een distributie met een eindige variantie, convergeert naar het populatie gemiddelde μ voor steeds grote steekproeven:

$lim_{n\to\infty} P(\mid \overline{X} - \mu \mid > \epsilon) = 0$
Ofwel de kans dat het steekproef gemiddelde meer afwijkt van het populatie gemiddelde dan e en heel klein getal, convergeert naar 0 voor oneindig grote steekproeven.

Voor eindige populaties is dit natuurlijk zeker waar. Maar denk hier ook aan oneindig grote, of nagenoeg oneindig grote populaties, zoals bijvoorbeeld als je de gemiddelde massa van het electron wilt bepalen.

Tip: In deze video wordt de wet van grote aantallen nogmaals duidelijk uitgelegd.

Als je de wet goed leest zie je dat er een voorwaarde aan vast zit. Namelijk dat de variantie van de stochast eindig moet zijn, en dat dus de verwachtingswaarde van de stochast bepaald is. Er bestaan distributies, zoals de Cauchy of de Landau distributie waarvoor dit dus niet geldt. Deze distributies hebben oneindig lange staarten. In het figuur hier , in Fig. 9.1 zie je hoe de Cauchy distributie eruit ziet.

Wiskundig gezien kan de wet van grote aantallen dus weleens voor problemen zorgen. In Natuurkundige experimenten zijn verdelingen uiteindelijk vaak beknot door bijvoorbeeld de eindigheid van energie. Voor Natuurkundige experimenten gaat de wet van grote aantallen eigenlijk altijd wel op.

Overigens noemen we deze wet van grote aantallen de *zwakke* wet van grote aantallen, er bestaat ook een *sterke* wet. We gaan hier niet in op de kleine verschillen tussen deze twee wetten, online kun je er eventueel genoeg over vinden.

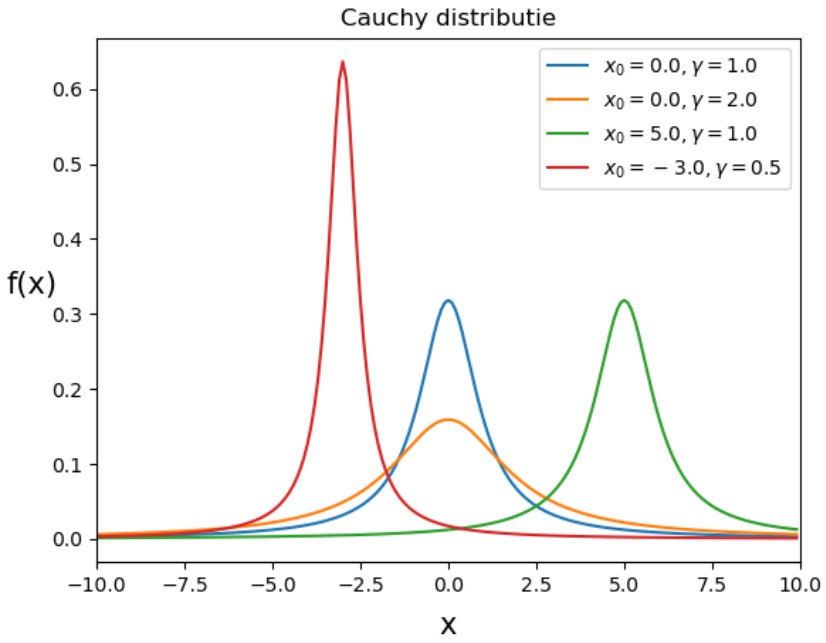


Figure 9.1: Cauchy verdeelde kansdistributies.

MODULE II Hoofdstuk 10

Meerdimensionale datasets

Het komt vaak voor dat we datasets hebben waarbij we meerdere variabelen tegelijkertijd hebben gemeten. Bijvoorbeeld als we een steekproef doen onder de bevolking waarbij we allerlei gegevens tegelijkertijd opvragen zoals leeftijd, inkomen, gezinssamenstelling etc. We kunnen dan niet alleen naar verdelingen kijken van bijvoorbeeld alleen het inkomen, maar we kunnen ook naar het inkomen kijken *afhankelijk* van de leeftijd. Dit levert dus meer informatie op dan als we deze gegevens afzonderlijk zouden hebben verzameld. Ook in natuurkundige experimenten komen multidimensionale datasets veel voor.

Voor elke afzonderlijke variabele kunnen we bijvoorbeeld het gemiddelde en de standaardafwijking berekenen met behulp van de formules die we in 'Basisbegrippen' (Hfdst. 1) hebben geïntroduceerd. Maar we kunnen nu ook kijken of de waarde van een observabele afhangt van een andere observabele in de dataset. Dit noemen we correlatie. Ook kunnen we berekenen of een spreiding in een variabele afhangt van de waarde van een andere variabele. We noemen die covariantie. Hieronder introduceren we eerst covariantie, daaronder komt correlatie aan bod.

10.1 Variantie en covariantie

De variantie geeft zoals eerder besproken (onder 'Basisbegrippen' (Hfdst. 1)) een maat voor de spreiding van een dataset aan. Bij een 2D dataset waarbij een variabele wordt aangegeven op de x-as en een andere variabelen op de y-as wordt de mate van spreiding o.a. aangegeven met de *covariantie*.

De covariantie bij een 2D dataset geeft aan in welke mate de data verspreid is over het twee dimensionale vlak.

Voor twee variabelen x en y wordt de covariantie aangeduid met $cov(x, y)$ en gegeven door:

$$cov(x, y) = E\left((x - E_x)(y - E_y)\right) \tag{10.1}$$

Hier staat E voor de *verwachtingswaarde*. De verwachtingswaarde voor x en y worden

respectievelijk aangegeven met E_x en E_y. De formule geeft dus aan dat de covariantie gelijk is aan de verwachtingswaarde van het verschil tussen de waarde van de variabele x en de verwachtingswaarde van x vermenigvuldigd met het verschil tussen de variabele y en de verwachtingswaarde van y.

Als bovengemiddelde waardes van x overwegend samen gaan met relatief hoge waardes van y, dan hebben we te maken met een positieve waarde voor de covariantie. Als bij bovengemiddelde waardes van x de waardes van y voornamelijk onder het gemiddelde van y liggen, dan is de covariantie negatief. Als de covariantie gelijk is aan nul dan is er, gemiddeld over de hele dataset, geen afhankelijkheid. Het kan zijn dat voor delen van de dataset wel degelijk een positieve covariantie bestaat, deze wordt dan opgeheven door een ander gedeelte met een negatieve covariantie. Als de covariantie nul is, betekent dat dus niet persé dat er geen afhankelijkheid is van x ten opzichte van y.

Met de volgende formules kun je de covariantie van een steekproef uitrekenen:

- Voor discrete verdelingen geldt:

$$cov(x, y) = \frac{1}{n} \sum_{i}^{n} (x_i - \bar{x}) \cdot (y_i - \overline{y}). \tag{10.2}$$

- Voor continue verdelingen geldt:

$$cov(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x\ y \cdot f(x, y)\ dy\ dx \tag{10.3}$$

De covariantie geeft dus aan in hoeverre waarden van de ene variabele toenemen/afnemen bij toenemende waarden van de andere variabele. De covariantie is een heel nuttige maat maar lastig te interpreteren vanwege de dimensies die, net als bij de variantie, niet dezelfde zijn als de variabelen zelf. Eenvoudiger is om naar de correlatiecoëfficiënt ρ te kijken.

Hier kun je een filmpje zien die covariantie uitlegt.

10.2 Correlatie

De correlatiecoëfficiënt ρ van een populatie is gedefinieerd als:

$$\rho_{x,y} = \frac{cov(x, y)}{\sigma_x \cdot \sigma_y}. \tag{10.4}$$

Hierbij is $cov(x, y)$ de covariantie tussen variabele x en variabele y, en zijn σ_x en σ_y de standaardafwijkingen van variabele x en y respectievelijk. Deze reken je dus uit met de formule die hierboven is gedefinieerd.

Voor een steekproef gebruiken we de notatie r_{xy} en de volgende vergelijking:

$$r_{xy} = \frac{cov(x, y)}{s_x \cdot s_y}. \tag{10.5}$$

De correlatiecoëfficiënt kan een waarde aannemen tussen de -1 en 1.

Als er geen correlatie is tussen de twee variabelen, dan is correlatiecoëfficiënt gelijk aan nul. Is de correlatiecoëfficiënt tussen de twee variabelen gelijk aan 1 of aan -1 dan zijn de twee variabelen maximaal afhankelijk. In het geval van een correlatiecoëfficiënt gelijk aan 1 is dit een positief lineair verband, in het geval van een correlatiecoëfficiënt gelijk aan -1 is dit een lineair verband met negatieve helling.

In het figuur, in figuur 10.1 zijn een aantal 2D datasets weergegeven met verschillende correlatiecoëfficiënten.

Hoe dichter de correlatiecoëfficiënt bij een waarde van 1 of -1 zit des te groter is de afhankelijkheid van de variabelen. Hoe dichter de correlatiecoëfficiënt bij nul zit des te kleiner is de correlatie tussen de variabelen.

In het figuur, in figuur 10.2 zie je een aantal voorbeelden van datasets die allen een correlatiecoëfficiënt hebben met waarde $\rho = 0$. Zoals je ziet hoeft een waarde van 0 niet te betekenen dat er helemaal geen verband is tussen de waardes van x en y. Wat het wel zegt is dat er over de gehele dataset even veel bovengemiddelde punten van x met een bovengemiddeld punt y corresponderen als het omgekeerde.

Je kunt hier een filmpje vinden waarin correlatie ook wordt uitgelegd. Er zijn meerdere 'spelletjes' op internet waarbij je kunt oefenen met het herkennen en raden van de correlatiecoëfficiënt van twee variabelen. Kijk bijvoorbeeld eens bij Geogebra-Correlatie game of Guess the correlation.

10.3 Correlatie en causaliteit

Soms betekent correlatie dat er oorzakelijk verband is tussen de twee observabelen. Dat wil zeggen dat de ene observabele invloed heeft op de andere observabele.

Een voorbeeld hiervan is als je kijkt naar de ijsverkoop en de buitentemperatuur. Omdat het warm is buiten hebben mensen meer trek in een ijsje. Het is dus niet zo gek dat je er een verband tussen vindt. Dit verband noemen we een **causaal** verband. Iets wordt veroorzaakt door iets anders.

In wetenschappelijk onderzoek zijn we altijd op zoek naar correlaties. Immers, die kunnen wijzen op onbekende wetten of onderliggende, nog onbekende fenomenen. Toch moet je behoorlijk oppassen om meteen een conclusie te trekken. Niet alle observabelen die gecorreleerd zijn, hebben een causaal verband. Het kan ook toeval zijn, als je maar genoeg variabelen tegen elkaar uitzet zal je er altijd wat vinden die toevallig een correlatie vertonen. Het kan ook komen door een verborgen parameter. Dit wordt ook wel Simpsons paradox genoemd.

Een bekend voorbeeld van een Simpsons paradox is een onderzoek naar veiligheid op de scheepvaart. Er is gebleken dat er een positieve correlatie is tussen het dragen van reddingsvesten en het aantal ongevallen waarbij mensen verdronken zijn. Dit is natuurlijk niet wat je verwacht! Voordat je adviseert om alle reddingsvesten weg te laten gooien is het

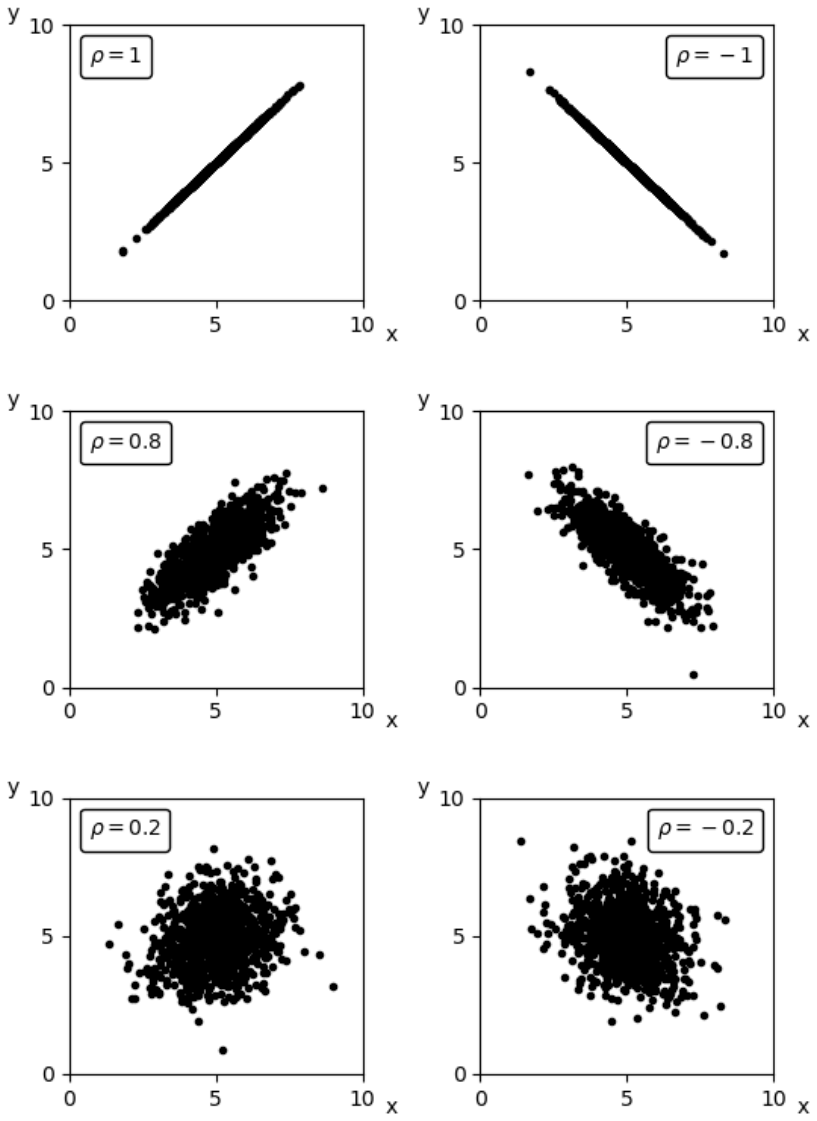


Figure 10.1: Tweedimensionale datasets met verschillende correlatiecoëfficiënten.

goed om nog iets verder onderzoek te plegen. Wat blijkt, de reddingsvesten worden alleen aangetrokken bij slecht weer op zee. De verborgen parameter is dus het weer. Als we de data nog een keer goed bekijken en nu kijken naar alleen de categorie slecht weer dan zien we dat de overlevingskans juist vele malen hoger is als een reddingsvest wordt gedragen.

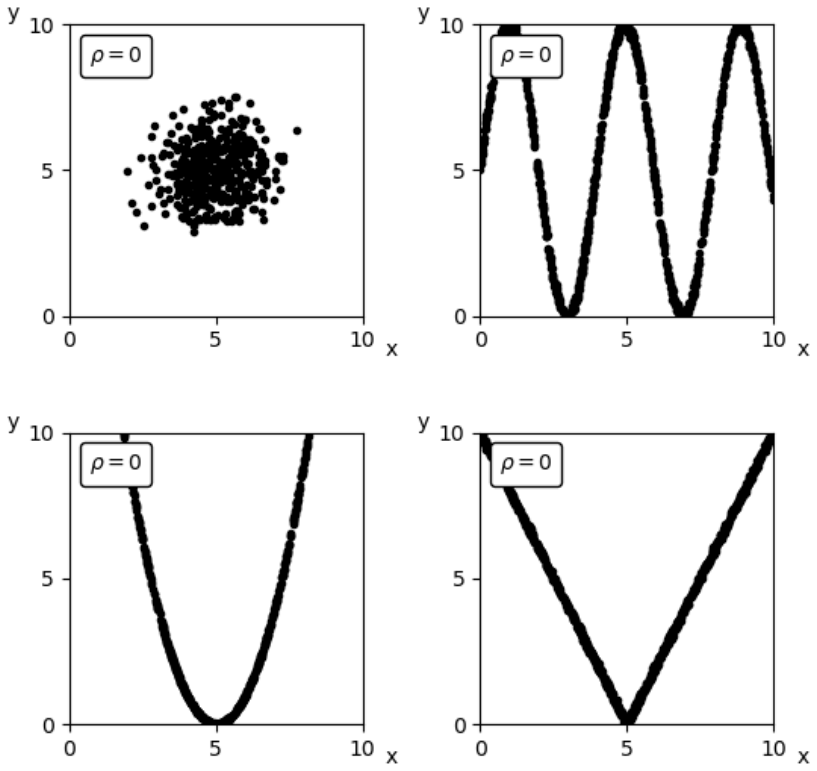


Figure 10.2: Tweedimensionale datasets met correlatiecoëfficiënt $\rho = 0$.

De les die je hieruit moet leren is dat je altijd heel goed moet nadenken over wat een verborgen parameter zou kunnen zijn en niet zomaar de conclusie trekken dat een correlatie ook causaliteit impliceert. Het is goed om zo'n conclusie eerst te onderbouwen met een plausibele verklaring.

MODULE II Hoofdstuk 11

Extra kans rekenregels

In module 1 (Hfdst. 5) hebben we de complement-regel, de en-regel en de of-regel geleerd voor het rekenen met kansen. Aan deze regels waren enkele voorwaarden verbonden.

De of-regel geldt alleen als de metingen A en B wederzijds uitsluitend zijn. Dat betekent dat een meting A niet kan voorkomen als B gemeten is.

> Een voorbeeld van kansen die niet wederzijds uitsluitend zijn is, als we weer kijken naar een set kaarten waar A bijvoorbeeld de kleur rood is en B het getal 4. Er bestaan rode kaarten met getal vier en in dit geval mogen we de kansen dus niet optellen.
>
> $$P(\text{rood of } 4) \neq P(\text{rood}) + P(4)$$

We breiden de regels hier verder uit en gaan kijken naar het combineren van kansen die niet wederzijds uitsluitend zijn. We kijken ook naar het begrip conditionele kans en introduceren Bayes theorema die gebruikt kan worden om informatie van kansen om te rekenen.

We introduceren eerst de begrippen die we nodig hebben in dit hoofdstuk.

De **vereniging**, ook wel de unie, van A en B wordt genoteerd met $A \cup B$ en is de verzameling van alle elementen van A en B.

De term (A en B) noemen we ook wel de **doorsnede**, of intersectie, van A en B. Het is het overlappende deel van elementen in de verzameling. De doorsnede wordt ook wel genoteerd met $A \cap B$.

Het **complement** van A wordt genoteerd met A^c en is het deel van de uitkomstenverzameling dat *niet* in A ligt.

Bovenstaande definities kunnen we ook visueel weergeven in Venn diagrammen. Deze vind je in Fig. 11.1.

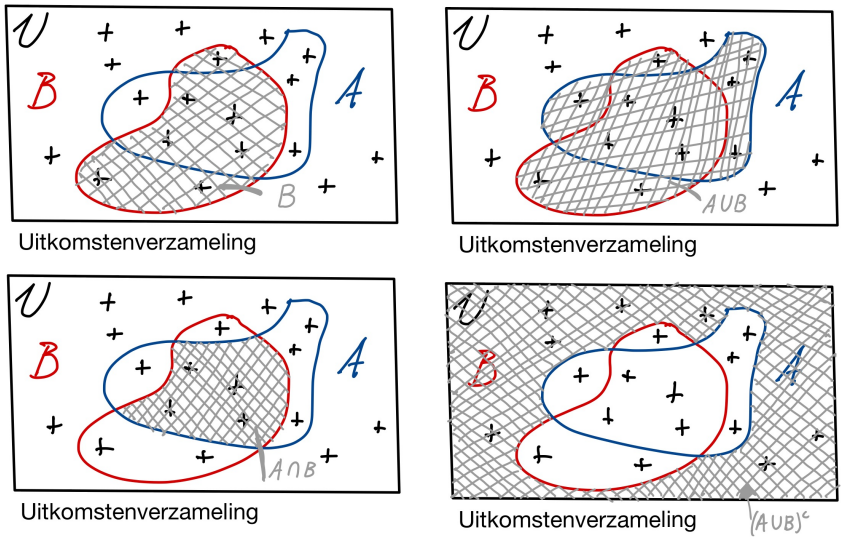


Figure 11.1: Venn diagrammen.

11.1 De of regel wanneer A en B niet wederzijds uitsluitend zijn

In het geval A en B niet wederzijds uitsluitend zijn dan:

$$P(\text{A en B}) \equiv P(A \cap B) > 0. \tag{11.1}$$

De kans dat A of B gemeten wordt is dan:

$$P(\text{A of B}) = P(A) + P(B) - P(\text{A en B}). \tag{11.2}$$

Voorbeeld De kans dat een kaart rood is en een vier als uitkomst heeft is $\frac{2}{52}$. De kans dat een kaart rood is of een vier is nu gelijk aan

$$P\left(\frac{1}{2}\right) + P\left(\frac{4}{52}\right) - P\left(\frac{2}{52}\right) = \frac{28}{52}.$$

11.2 Conditionele kans

Een conditionele kans wordt geschreven als $P(A \mid B)$ en kun je lezen als “Wat is de kans op meting A gegeven dat B is gemeten”. Let op dat je een conditionele kans niet zomaar

kan omkeren: $P(A \mid B) \neq P(B \mid A)$!
Een sprekend voorbeeld hiervan is de volgende. De kans dat een persoon zwanger is gegeven dat de persoon een vrouw is, $P(\text{zwanger} \mid \text{vrouw})$, is niet gelijk aan de kans dat iemand een vrouw is gegeven dat de persoon zwanger is, $P(\text{vrouw} \mid \text{zwanger})$. De laatste kans is duidelijk gelijk aan 1, als je zwanger bent ben je zeker een vrouw. De eerste kans is een stuk kleiner!

De conditionele kans kunnen we berekenen met:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}. \tag{11.3}$$

De noemer in deze vergelijking, $P(B)$, noemen we ook wel een normalisatie term. De kans $P(A \cap B)$ moet genormaliseerd worden naar de kans $P(B)$, immers het is al een gegeven dat B waar is.

Visueel is dit wellicht het meest eenvoudige om te zien. Als het gegeven is dat de uitkomst in het deelgebied B ligt, dan is de kans dat het ook de waarde A bezit gelijk aan het oppervlak van de overlap tussen A en B gedeeld door het oppervlak van B. Dat de uitkomst in B ligt weten we al, dus we moeten de kans 'normaliseren' naar B.

11.3 Bayes theorema

Met behulp van de conditionele kans formule kunnen we nu Bayes theorema afleiden.

Een belangrijke stap is om te realiseren dat de doorsnede van A en B natuurlijk precies hetzelfde is als de doorsnede van B en A. En dus geldt:

$$P(A \cap B) \equiv P(B \cap A). \tag{11.4}$$

Als we de formule van de conditionele kans nu anders opschrijven vinden we de vergelijking

$$P(A \cap B) = P(A \mid B) \cdot P(B). \tag{11.5}$$

Deze vergelijking combineren we nu met de vergelijking voor $P(B \cap A)$:

$$P(B \mid A) \cdot P(A) \equiv P(A \mid B) \cdot P(B), \tag{11.6}$$

ofwel

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}. \tag{11.7}$$

Deze formule heet het Bayes theorema en blijkt een van de meest krachtige formules om kansen te berekenen. Het beste is om dit te demonstreren met een voorbeeld.

Voorbeeld Een patiënt komt de huisartsenpost met pijnklachten in de buik, de doktersassistent vermoedt een blaasontsteking en onderzoekt de urine met een zogeheten combinatietest.
De test is relatief betrouwbaar. Slechts in 5% van de tests volgt er een positieve testuitslag terwijl de patiënt niet ziek is. Dit noem je een fout-positieve uitslag en noteren we hier als $P(+ \mid \text{niet ziek})$. In 3% van de gevallen is de testuitslag fout-negatief; de patiënt heeft een blaasontsteking maar de uitslag is toch negatief. De fout-negatieve kans noteren we met $P(- \mid \text{ziek})$.
Voordat de testuitslag bekend is weet de assistent al wat de *voorafkans* is, de voorafkans is de kans dat een patiënt met die type klachten een blaasontsteking heeft. Dit weet men door jarenlange ervaring in de praktijk. De voorafkans is verschillend voor kinderen (0.20) en volwassen (0.60). Kinderen hebben ook vaak om andere reden buikpijn.
De testuitslag is positief. **Wat is nu de kans dat de patiënt daadwerkelijk een blaasonsteking heeft?**
Wat we dus willen weten is de kans $P(\text{ziek} \mid +)$, namelijk wat is de kans dat de patiënt ziek is gegeven de positieve testuitslag.
Hiervoor gebruiken we Bayes theorema om dit te berekenen.

$$P(\text{ziek} \mid +) = \frac{P(+ \mid \text{ziek}) \cdot P(\text{ziek})}{P(+)}.$$

We kennen $P(+ \mid \text{niet ziek}) = 0.05$, namelijk dit is de fout-positief en we kennen de fout-negatief $P(- \mid \text{ziek}) = 0.03$. De voorafkans is $P(\text{ziek})$ hangt af van de leeftijd van de patiënt. Voor Bayes theorema moeten we ook nog de kans op uberhaupt een positieve testuitslag weten, dit is $P(+)$. Deze kunnen we berekenen met de volgende formule:

$$P(+) = P(+ \mid \text{ziek}) \cdot P(\text{ziek}) + P(+ \mid \text{niet ziek}) \cdot P(\text{niet ziek}).$$

Namelijk, er zijn twee opties. Je krijg een positieve uitslag en je bent inderdaad ziek. Of je krijgt een positieve uitslag terwijl je helemaal niet ziek bent. In beide gevallen moet je dit vermenigvuldigen met de kans op de bijbehorende toestand (ziek of niet ziek). We zijn er hierbij vanuit gegaan dat een testuitslag altijd positief of negatief is. De kans $P(+ \mid \text{niet ziek})$ hebben we al gezien, dat is de fout-positief. De kans $P(+ \mid \text{ziek})$ is gelijk aan het complement van de fout-negatief dus $P(+ \mid \text{ziek}) = 1 - P(- \mid \text{ziek})$.
Invullen voor kinderen geeft:

$$P(+) = (1 - 0.03) \cdot 0.20 + 0.05 \cdot (1 - 0.20) = 0.234 \ .$$

We vullen dit in in Bayes theorema:

$$P(\text{ziek} \mid +) = \frac{P(+ \mid \text{ziek}) \cdot P(\text{ziek})}{P(+)} = \frac{(1 - 0.03) \cdot 0.20}{0.234} = 0.83.$$

Van de kinderen met een positieve test uitslag heeft dus 83% ook daadwerkelijk een blaasontsteking. Dit is een stuk lager dan we misschien zouden verwachten. De test is namelijk betrouwbaar, in 97% van de gevallen met blaasonsteking geeft de test immers het juiste resultaaat. Deze afwijking heeft te maken met de lage voorkans bij kinderen; het is nog redelijk waarschijnlijk dat het kind niet ziek is maar een fout-positieve uitslag heeft.
De kans dat het kind niet ziek is bij een positieve uitslag is dus $1 - 0.83 = 0.17 = 17\%$. Reken nu zelf de kans $P(\text{ziek} \mid +)$ uit voor een volwassene en controleer dat dit gelijk is aan 0.97. Deze kans is veel groter dan bij de kinderen. Dit heeft alles te maken met de voorkans.

We hebben in deze twee voorbeelden gezien hoe we informatie over conditionele kansen kunnen omzetten. Het theorema van Bayes maakt het mogelijk om nieuwe informatie te gebruiken. De nakans wordt berekend met een test uit voorkans (ook wel prior), een testuitslag en een normalisatie. De normalisatie is in het geval van het voorbeeld de kans $P(+)$, de kans dat er uberhaupt een positieve uitslag volgt.

Voordat de patiënt de test afnam konden we alleen afgaan op de praktijkervaring van de assistent. Een blaasontsteking bij een kind is onwaarschijnlijk (slechts 20%) en bij een volwassene waarschijnlijk (60%). Na het uitvoeren van de test hebben we meer informatie, maar nog steeds is het belangrijk om de ervaring van de assistent mee te nemen (de voorkans), maar ook mee tenemen hoe groot de kans is op een positieve testuitslag (de normalisatie). Dat een patiënt daadwerkelijk een blaasontsteking heeft is in beide gevallen (kind/volwassene) waarschijnlijk, maar bij een kind is het misschien goed om ook nog even wat andere oorzaken uit te sluiten.

MODULE II

Hoofdstuk 12

Schatmethodes

In dit hoofdstuk leggen we uit wat een **schatter** is.

In een experiment willen we met een meting een bepaalde grootheid te weten komen. Soms kunnen we die direct opmeten, maar meestal hebben we een methode of een recept nodig om dit te doen. Denk bijvoorbeeld bij de proef met de halfwaardedikte. We nemen eerst een set metingen en vervolgens hebben we een recept om hieruit de halfwaardedikte te bepalen. Deze halfwaardedikte *schatten* we met behulp van de methode die we een *schatter* noemen (Engels: estimator).

Een **schatter** is een steekproefgrootheid (stochast) die als doel heeft een populatieparameter in te schatten.

Het woord schatten is in het Nederlands vaak geassocieerd met een wat slordige manier om iets te bepalen. "Even schatten hoeveel knikkers er in de pot zitten in plaats ze nauwkeurig te tellen." Het begrip **schatter** in de statistiek is veel algemener en probeert juist uit te gaan van nauwkeurigheid.

Zonder dat we dit expliciet benoemd hebben zijn we al veel schatters tegengekomen. Het steekproefgemiddelde zelf is bijvoorbeeld ook een schatter.

Voorbeeld Het gemiddelde van een steekproef met waardes x_i en grootte n is gedefinieerd als:
$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.
Het steekproefgemiddelde $\overline{x}$ is berekend met een formule (recept) van metingen. Het steekproefgemiddelde $\overline{x}$ noemen we hierom ook wel een **schatter**. De grootheid die we proberen in te schatten is het populatiegemiddelde.

Het doel van de schattingstheorie is een functie of methode te vinden die een goede schatting oplevert van de grootheid die we willen meten. Dat we deze grootheid nooit exact kunnen bepalen is inmiddels duidelijk, alle metingen zijn onderhevig aan meetfouten en onzekerheden.

Er zijn 3 criterea waarmee we evalueren waaraan een goede schatter voldoet:

- de schatter moet **zuiver** zijn,
- de schatter moet **consistent** zijn,
- de schatter moet **relatief efficïent** zijn.

Als we verschillende schatters definïeren bieden de criterea een houvast om te bepalen wat de beste schatter is. We gaan kort op deze 3 criterea in.

12.1 Zuiver

De zuiverheid van een schatter wordt gedefinieerd als:

$$b = E(\bar{x}) - \mu \tag{12.1}$$

Waarbij b de onzuiverheid is, $E(\bar{x})$ de verwachtingswaarde van de te schatten grootheid en μ het populatiegemiddelde van de te schatten grootheid. In het Engels noemen we deze afwijking de bias, vandaar de letter b. Voor het woord zuiverheid gebruiken we ook wel eens het woord vertekend.

Een goede schatter is zuiver. Dat wil zeggen dat de verwachtingswaarde van de schatter gelijk is aan de populatiegrootheid.

Voorbeeld Het steekproefgemiddelde is zuiver als de verwachtingswaarde van het steekproefgemiddelde $E(\bar{x})$ gelijk is aan het populatiegemiddelde μ.

12.2 Consistent

Naarmate de steekproef groter wordt benaderen we de populatiegrootheid met de schatter. Met andere woorden, de schatter voldoet aan de wet van grote aantallen.

Voorbeeld Naarmate we meer data nemen benadert het steekproefgemiddelde (de schatter) het populatiegemiddelde (de populatiegrootheid).

12.3 Efficïent

Als er meerdere schatters kunnen worden gedefinïeerd dan heeft een goede schatter relatief de kleinste variantie. Anders gezegd, als we meerdere recepten kunnen bedenken om tot een inschatting te komen van de populatiegrootheid die we willen meten dan heeft de beste schatter de kleinste meetonzekerheid.

Voorbeeld We hebben twee schatters, θ_1 en θ_2, die allebei het doel hebben dezelfde populatiegrootheid μ te bepalen. De variantie op de schatters zijn s_1^2 en s_2^2 waarbij $s_1 < s_2$. Beide schatters voldoen aan de zuiverheid en consistentheid criterea. De beste schatter is degene met de kleinste variantie. In dit voorbeeld is dat θ_1.

12.4 Voorbeelden

We bekijken een aantal voorbeelden waar het misgaat met de schatmethode.

Onzuiver: Een ongeijkte meetlat Je meet in een experiment de gemiddelde lengte van lucifers. Elke lucifer leg je langs een nauwkeurige lineaal en je noteert heel precies de lengte in millimeters. De meetlat zelf is helaas ongeijkt en 1 millimeter op de schaal van de meetlat is in werkelijkheid 10% groter. Hierdoor ontstaat er een vertekend beeld van de lengte van de lucifers. De verwachtingwaarde van het gemiddelde is dat die 10% te groot zal uitvallen. De relatieve onzuiverheid zal dan ook precies +10% zijn. Als we de het gemeten gemiddelde kennen kunnen we deze omrekenen naar de absolute onzuiverheid b.

Onzuiver: Een school vissen Je bestudeert een school vissen. Sommige vissen hebben 2 zwarte strepen en sommige 3. Het doel van je onderzoek is om de verhouding te bepalen tussen het aantal vissen met 2 strepen en het aantal vissen met 3 strepen. Je telt in totaal 150 vissen, 50 vissen hebben twee strepen, 70 vissen hebben 3 strepen. Van 40 vissen kun je niet zien hoeveel strepen ze hebben omdat er toevallig net steeds een andere vis voor zwemt. Je besluit de 40 vissen eerlijk te verdelen over de twee groepen. Je meet nu 70 vissen met twee strepen en 90 vissen met drie strepen en vindt een verhouding van 7:9.
Deze schatmethode levert een vertekend beeld op en is dus **onzuiver**. Het zou beter zijn om de 40 ongeïdentificeerde weg te laten uit de berekening en de verhouding op 5:7 te bepalen. Immers van de groep ongeïdentificeerde vissen mag je aannemen dat ze dezelfde verhouding hebben als de vissen die je wel goed hebt kunnen zien. Wel levert

de groep van 40 een extra onzekerheid in de bepaalde verhouding. In dit voorbeeld heeft de onzuiverheid niet te maken met een ongeijkt meetinstrument maar met een onzuivere schatmethode.
Het zou goed zijn om nog iets meer te weten over dit onderzoek. Kan het zo zijn dat bij sommige vissen waarbij maar twee strepen te zien zijn de derde streep er wel was maar niet zichtbaar was? Ook dat kan een vertekening van het het eindresultaat opleveren.

Niet consistent Het meest bekende voorbeeld van een schatter die niet consistent is is de volgende. Je neemt een steekproef met n waardes van x. Je steekproef ziet er als volgt uit: $\{x_1, x_2, .., x_n\}$. We willen met de steekproef het populatiegemiddelde μ_x bepalen en gebruiken hiervoor de eerste waarde van de steekproef, x_1. Op zich is dit een zuivere schatter, de waarde van x_1 geeft wellicht niet een heel nauwkeurige schatting van μ_x, maar hij is niet vertekend. De schatmethode is echter niet consistent met de wet van grote aantallen. De nauwkeurigheid van de schatting wordt niet beter als we de steekproef vergroten en de schatmethode noemen we dus *niet consistent*.

Efficïent Stel we hebben een verzameling van dinosaurusbotten en we weten dat de botten afkomstig zijn van twee soorten dinosaurussen. We gaan de botten klasificeren. We kunnen hierbij gebruik maken van de kleur van de botten. De botten van de ene dinosaurus lijken iets lichter van kleur te zijn dan de andere. Er is wel een grijs gebied, er zijn botten die we niet kunnen klassificeren omdat ze niet heel licht en ook niet heel donker van kleur zijn. Deze botten vallen er precies tussen in.
Een andere methode is om te kijken naar de dichtheid van de botten. Hier zit een groot verschil in. De botten van de ene soort dino's hebben een veel grotere massadichtheid dan de andere. We kunnen door het volume op te meten en het massa van de botten het soortelijk gewicht bepalen. Het verschil in massadichtheid is veel groter dan de onzekerheid van de meting en ook zit er maar een heel kleine spreiding in de massadichteid voor de verschillende exemplaren dino's van dezelfde soort.
Dit tweede methode is heel omslachtig maar wel zeer efficïent. Voor alle botten kun je uitsluitsel geven. Omdat er een alternatieve methode bestaat die beter is, noemen we de klassificatie methode met behulp van kleur *niet efficïent*.

MODULE II Hoofdstuk 13

Opdrachten module 2

Tijdens laptopcolleges 3 en 4 werken we aan het de opdrachten van module 2. In deze module gaan we werken aan de volgende opdrachten.

- M2.1 Grote Aantallen II ** (Hfdst. 13.1)
- M2.2 Meesjes **** (Hfdst. 13.2)
- M2.3 Halfwaardedikte II *** (Hfdst. 13.3)

De sterren geven een indicatie voor hoeveel werk een opdracht is. Let op dat je deze week goed plant!

13.1 Opdracht M2.1 Grote Aantallen II **

We gaan in deze opdracht verder kijken naar de ton met kogels uit opgave M1.4. In die opgave begonnen we met een ton met 80 kogels en berekenden we het gemiddelde, $g_n = \overline{m_n}$ over de eerste n kogels van de set. Zo kregen we de distributie van g_n versus n. ***Voordat je met deze opdracht begint, controleer eerst even in ANS of je dit goed hebt gedaan en corrigeer eventueel je fouten.***

We gaan nu naar meerdere tonnen kijken, steeds met 80 kogels en uit dezelfde fabriek. Voor elke ton berekenen we de waardes van g_n opnieuw. Dit doen we om het effect van de grootte van een steekproef bekijken. Eerst nemen we een steekproef van 10 kogels uit elke ton en gaan we daar het gemiddelde van bepalen. Daarna vergroten we de steekproef en kijken we hoe het gemiddelde verandert. Hierbij zijn we vooral benieuwd hoe groot de steekproef moet zijn om dicht bij het populatiegemiddelde te komen.

We gaan er in deze opgave stap voor stap doorheen.

- Maak eerst 100 verschillende datasets. Elke dataset is een ton met 80 kogels. Dit kan je doen door steeds een andere *seed* mee te geven aan de dataset generator:

```
datasets = [ds.DataSetGroteAantallen(i) for i in range(0,100)]
```

De seed zorgt ervoor dat de datasets verschillend zijn, het is als het ware het startpunt van de random nummer generator. Als je steeds dezelfde seed meegeeft krijg je steeds dezelfde random dataset. (Probeer maar eens.)

We hebben hier gebruik gemaakt van een *list comprehension*. Dat is een verkorte manier van het toepassen van een *for* statement waarbij het resultaat direct de gewenste lijst oplevert zonder dat de commando's *extend* of *append* nodig zijn. Als je hier meer over wilt weten dan vind je op het internet veel voorbeelden, bijvoorbeeld hier.

Je hebt nu een `list` die `datasets` heet met 100 items. Elke item in `datasets` is op zichzelf een lijst met 80 meetwaardes.

- **M2.1a) Maak nu eerst een histogram van *alle eerste* elementen, m_1, van de 100 datasets. Zorg dat je histogram er netjes uitziet.**
  Tip: Kijk eens of je hier een *list comprehension* voor kunt gebruiken. Welke index heeft het eerste element van een lijst in python?

- **M2.1b) Wat is het gemiddelde, g_1, en de standaardafwijking s_1 van dit histogram? Denk bij het noteren aan de eenheden en de juiste notatie!** Kijk naar je histogram uit M2.1a en controleer of je resultaat overeen komt met je verwachting.

We nemen nu uit elke ton een steekproef van 10 kogels. We berekenen eerst het gemiddelde g_{10} per steekproef, daarna bepalen we het gemiddelde van het gemiddelde.

- **M2.1c) Bereken voor elk van de 100 datasets het gemiddelde over de eerste 10 metingen en laat de distributie van deze gemiddeldes g_{10} zien in een histogram.**
  Tip: Denk hierbij aan de functie die je al voor M1.4 hebt geschreven.

Je ziet nu duidelijk dat niet elke steekproef hetzelfde gemiddelde oplevert. Er is een bepaalde spreiding van de gemiddeldes g_{10} die we hebben berekend. De distributie van gemiddeldes heeft dus zelf ook weer een gemiddelde en een spreiding.

- **M2.1d) Bereken van deze distributie het gemiddelde $\overline{g_{10}}$, dit is het gemiddelde van de gemiddeldes g_{10}. Bereken ook de standaardafwijking van de gemiddeldes: $s_{g_{10}}$.**

De spreiding van de berekende gemiddeldes kun je in dit geval zien als een maat voor de meetonzekerheid. Immers, we zien dat de berekende grootheid (g_{10}) niet altijd hetzelfde resultaat geeft. Het berekende resultaat *varieert*. Als mate van de meetonzekerheid nemen we de standaardafwijking van de distributie.

We gaan dit nu herhalen voor met verschillende groottes van de steekproef n.

Maak een functie die voor een steekproefgrootte n de standaardafwijking van de gemiddeldes g_n terug geeft. Bereken hiervoor in de functie voor elk van de 100 datasets het gemiddelde over de eerste n metingen.
Roep nu de functie aan voor de volgende waardes van n: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80. Controleer of de punten voor $n = 1$ en $n = 10$ dezelfde resultaten opleveren als dat je net had.

- **M2.1e) Maak nu een grafiek waarin je de berekende standaardafwijking s_{g_n} uitzet tegen de grootte van de steekproef, n.**
- **M2.1f) Maak een nieuwe grafiek waarin je de berekende s_{g_n} uitzet tegen $1/\sqrt{n}$.**

- **M2.1g) Kun je iets zeggen over de grafieken? Beschrijf wat je ziet en probeer daar een conclusie uit te trekken.**

Wat we hebben gedaan in deze opdracht is illustreren wat er gebeurd als we een steeds grotere steekproef nemen.

13.2 M2.2 Meesjes ****

Je vindt helaas een dood meesje in de tuin. Het lijkt op een koolmeesje maar het zou ook een pimpelmeesje kunnen zijn. Deze twee vogeltjes lijken erg veel op elkaar. Er zijn manieren om pimpelmeesjes van koolmeesjes te onderscheiden met behulp van uiterlijke kenmerken. Maar je bent een Natuurkundige en geen Bioloog. Online vind je een dataset met informatie over het massa en de spanwijdte van beide soorten meesjes.

Voordat we aan deze opdracht beginnen moeten we eerst een nieuwe versie downloaden van de `DAS_DatasetGenerator.py`. Zonder de nieuwe versie werkt deze opgave niet. Down-

load ook het bestand `M2.2_Meesjes.py` en zorg dat deze in dezelfde folder staat als het `DAS_DatasetGenerator.py` bestand.

We genereren eerst twee datasets met behulp van de volgende regel code:

```
m_km, span_km, m_pm, span_pm = ds.datasetVogeltjes()
```

De variabelen hebben de volgende betekenis:

```
m_km    : de massa van een koolmeesje in gram
span_km : de spanwijdte van een koolmeesje in cm
```

De laatste twee variabelen zijn de datapunten voor pimpelmeesjes. De twee variabelen van de koolmeesjes horen bij elkaar. Van elk meesje in de dataset zijn zowel de massa als de spanwijdte gemeten. De dataset is zo geordend dat het n-de punt uit de `m_km`-lijst bij het n-de punt uit de `span_km`-lijst hoort. Dit zijn de gegevens van het n-de meesje. Let er dus goed op dat je de lijsten in de juiste volgorde houdt! Voor de twee variabelen van de pimpelmeesjes geldt precies hetzelfde.

We gaan eerst naar de twee massaverdelingen van de meesjes kijken.

- **M2.2a) Plot de massaverdelingen van beide meesjes in een histogram. Maak een apart histogram waarin je spanwijdtes van de twee soorten meesjes plot.** Laat in een legenda zien welke meesje bij welke kleur hoort. Maak de twee histogrammen netjes af en zorg dat duidelijk is welke distributie bij welk soort meesje hoort.

  TIP: Gebruik de plot optie `alpha=0.8` zodat je histogrammen wat doorzichtig worden. Zo kan je het achterste histogram ook nog altijd goed zien.

- **M2.2b) Maak een tabel waarin je voor beide soorten meesjes de gemiddeldes, de standaardafwijkingen en de varianties noteert.** Let goed op de notatie en denk ook even aan de eenheden.

We meten nu de massa op van het meesje dat je gevonden hebt. Gebruik de volgende regel code om dat te doen:

```
    mees_m_laag, mees_m_hoog = ds.meetMassaMeesje()
```

Je krijgt nu een onderwaarde `mees_m_laag` en een bovenwaarde `mees_m_hoog` terug. Deze geven de onzekerheid op de meting aan. Het gemiddelde van deze twee is de gemeten massa, de centrale waarde. De waarde van de massa van de mees ligt **zeker** tussen de boven- en onderwaarde in.

Met deze informatie kunnen we nu met de Frequentist Methode de kans uitrekenen dat onze mees een koolmeesje is.

- **M2.2c) Gebruik de dataset `m_km` om de kans uit te rekenen dat je een koolmeesje vindt die een massa heeft die in het gebied `mees_m_laag` en `mees_m_hoog` in ligt. Herhaal dit voor het pimpelmeesje, bereken dus ook $P(m_{\mathbf{obs}} \mid \mathbf{pimpelmees})$.**
  Dit noem je ook wel de voorwaardelijke kans $P(\text{mees\_m\_laag} < m < \text{mees\_m\_hoog} \mid \text{koolmees})$. Voor het gemak noteren we dit even als $P(m_{\text{obs}} \mid \text{koolmees})$. Zie ook het hoofdstuk over Extra Kansrekenregels (Hfdst. 11) over voorwaardelijke kansen.
  Tip Bedenk dat je voor de dataset van de pimpelmeesjes altijd zeker weet dat het om een pimpelmeesje gaat en dat dus per definitie $P(\text{pimpelmees}) \equiv 1$.

- **M2.2d) Als je kijkt naar de uitkomst van M2.2c), wat voor soort vogeltje denk je dat het is? Beredeneer je antwoord.**

De frequentist methode, zoals we die hierboven gebruiken, is uiteindelijk een ratio tussen twee getallen. Deze twee getallen hebben een onzekerheid volgens de Poisson verdeling.

- **M2.2e) Schrijf de formule uit hoe de onzekerheden van de noemer en teller zich propageren naar de onzekerheid op de uitgerekende kans. Noteer deze formule en bereken met behulp van deze formule de onzekerheden uit op de kansen die je in M2.2c) hebt berekend.**

Je besluit ook de spanwijdte van de mees op te meten. Misschien geeft dat wel meer uitsluitsel.

```
mees_span_laag, mees_span_hoog = ds.meetLengteMeesje()
```

De output volgt dezelfde logica als hiervoor.

- **M2.2f) Gebruik dezelfde methode als hiervoor om beide kansen $P(w_{\mathbf{obs}} \mid \mathbf{koolmees})$ en $P(w_{\mathbf{obs}} \mid \mathbf{pimpelmees})$ uit te rekenen maar nu door (alleen) gebruik te maken van de informatie van de spanwijdtes. Noteer ook de onzekerheden op de uitgerekende kansen.**

- **M2.2g) Op basis van deze informatie, wat denk je nu dat het voor vogeltje is? Beredeneer je antwoord.**

We kunnen nu natuurlijk ook de gecombineerde informatie gebruiken. Hiervoor gaan we

eerst de data visualiseren.

- **M2.2h) Maak een tweedimensionale scatterplot die de tweedimensionale dataset van de massa versus de spanwijdte voor zowel de pimpelmezen als de koolmezen.**
  TIP gebruik de opties `'o',markersize=3` in de plot functie. Zorg dat beide datasets weer hun eigen kleur hebben en vergeet de legenda niet.

Het valt misschien op dat er een verband lijkt te zijn tussen beide variabelen. We gaan daar eerst naar kijken naar de covariantie (Hfdst. 10) en de correlatie tussen de massa en de spanwijdte voor beide vogelsoorten.

- **M2.2i) Bereken de covariantie en de correlatie tussen de massa en de spanwijdte voor zowel de koolmeesje als de pimpelmeesjes meetgegevens.**

- **M2.2j) Als je naar de berekende correlaties kijkt wat valt dan op, wat voor verband zit er tussen de twee variabelen? Als je toch even als een Bioloog nadenkt, is dit dan wat je verwacht? Leg uit.**

We gaan terug naar de kansberekeningen.

- **M2.2k) Combineer nu de gegevens en bereken de kansen** $P(m_{\mathbf{obs}}$ **en** $w_{\mathbf{obs}} \mid$ **koolmees)** **en** $P(m_{\mathbf{obs}}$ **en** $w_{\mathbf{obs}} \mid$ **pimpelmees).**

- **M2.2l) Welk vogeltje denk je nu dat het is? Beredeneer je antwoord.**

Na al deze berekeningen lopen we een eindje in de tuin. Op de plek waar we eerder het meesje aantroffen zit nu een ander meesje hartstochtelijk te zingen. Aan de zang hoor je direct dat dit een pimpelmeesje is. Je schat in dat er een kans is van 90% dat dit pimpelmeesje bij het andere meesje hoorde, en dat dat dus ook een pimpelmees is.

- **M2.2m) Bereken nu de kans dat het inderdaad een pimpelmeesje is geweest: P(pimpelmees $\mid$ m_{obs} en w_{obs}). Bereken hier alleen de centrale waarde.**

TIP: Maak hierbij gebruik van de vergelijking (Hfdst. 11) van Bayes. Om $P(m_{\text{obs}}$ en $w_{\text{obs}})$ te berekenen kun je gebruiken maken van de volgende formule:

$$P(C) = P(C \mid D) \cdot P(D) + P(C \mid \text{niet } D) \cdot P(\text{niet } D)$$

.

13.3 M2.3 Halfwaardedikte II ***

We gaan nu terug naar het experiment uit opgave M1.5 waarbij we de halfwaardedikte van lood onderzoeken voor een bepaalde gamma-bron. We onderzoeken in deze opdracht de onzekerheid op het meetresultaat.

In opgave M1.5 gebruikten we een methode om de halfwaardedikte te bepalen. Bij deze methode zochten we naar de eerste dikte, d, in de grafiek waarvoor geldt dat $N \leq 0.5 \times N_0$. Hiervoor wordt steeds een ratio, R, berekend:

$$R = \frac{N_d}{N_0}. \tag{13.1}$$

Zodra deze ratio onder de 0.5 komt nemen we d als de halfwaardedikte. Hierbij geldt dan $N_d \equiv N_{\text{half}}$.

- **M2.3a) Wat is de onzekerheid op de ratio R? Bereken deze door gebruik te maken van de onzekerheden op N_0 en N_{half} en de regels voor propagatie van ongecorreleerde fouten (Deze kan je hier vinden (Hfdst. 8)). Schrijf je berekening helemaal uit.**

We maken bij het bepalen van de halfwaardedikte gebruik van een recept. Zo'n recept om de waarde van een parameter te bepalen noemen we ook wel een **schatter**. De parameter die we hier willen bepalen is de halwaardedikte van lood (voor de energie van onze bron). De schatter is in dit geval:
d_{half} = kleinste waarde van d waarvoor geldt dat $R = \frac{N_d}{N_0} < 0.5$.

We gaan het experiment nu 50 keer herhalen en kijken naar de distributie van de gevonden halfwaardediktes. Uit deze distributie bepalen we de standaardafwijking en gebruiken dit als onzekerheid op de gevonden dikte d_{half}.

- Schrijf een loop waarin je 50 maal een nieuwe dataset genereert. Uit elk van deze datasets bepaal je een de halfwaardedikte met de ratio-methode. Om 50 unieke dataset te maken moet je steeds de *seed* veranderen. Dat kan je doen door deze mee te geven aan de DAS dataset generator:

```
for j in range(0,50) :
    counts, diktes = ds.DataSetHalfwaardeDikte(j)
```

- Met bovenstaande loop maak je 50 unieke datasets aan waarbij de counts die gemeten worden steeds worden gevarieerd volgens de Poisson statistiek. Bereken nu, binnen de loop, voor elk van deze dataset de halfwaardedikte met de ratio-methode. Zorg dat je dit getal bewaart in een lijst.
- **M2.3b) Maak een histogram waarin je de gevonden halfwaardediktes van de 50 verschillende experimenten laat zien.** Zorg dat het histogram de distributie netjes laat zien en dat de as-labels goed zijn aangemaakt.
  TIP: De binning in het histogram luistert nauw doordat er alleen bepaalde uitkomsten van de halfwaardedikte mogelijk zijn. Reken precies uit wat de range en de binning moet zijn in het histogram om te voorkomen dat je lege bins midden in de distributie krijgt.

- **M2.3c) Ziet de distributie eruit zoals je verwacht had? Beredeneer je antwoord.**

- **M2.3d) Bepaal nu het gemiddelde van de meetuitkomsten en de standaardafwijking van de distributie.**
- **M2.3e) Zeggen deze getallen ook iets of de gemeten waardes systematisch te hoog of te laag uitkomen. Beredeneer je antwoord.**

We gaan nu bekijken hoe zuiver de meting is. Lees hiervoor eerst het hoofdstuk schatmethodes (Hfdst. 12).

De zuiverheid is gedefinieerd als als het verschil tussen de verwachtingswaarde van een schatter en de 'echte' waarde van de te meten parameter. Het symbool voor de zuiverheid is b (van het Engelse bias). De formule van de onzuiverheid is:

$$b = \text{gemeten waarde} - \text{echte waarde}. \tag{13.2}$$

Bijvoorbeeld als het gemiddelde meten van een parameter is de onzuiverheid gedefinieerd als:

$$b = \bar{x} - \mu. \tag{13.3}$$

Waarbij $\bar{x}$ het steekproefgemiddelde en μ het populatiegemiddelde is.

Hoe groter dit verschil, hoe meer onzuiver de meting is. In ons geval is het het dus het verschil tussen de gemiddelde gemeten d_{half} en de 'echte' halfwaardedikte. Bij gesimuleerde data kunnen we dit onderzoeken. We kunnen de verwachtingswaarde van de schatter vergelijken met de initiële waardes die we hebben gebruikt om in de simulatie de dataset aan te maken.

Om de zuiverheid van ons experiment te bepalen gaan we dus de bepaalde halfwaardedikte te vergelijken met de initiële halfwaardedikte die gebruikt is om de data te simuleren. Roep hiervoor de volgende functie aan in de dataset generator: metingen, diktes, d_true = ds.DataSetHalfwaardeDikteVariatie(s,d_input)
Je geeft twee variabelen mee aan de functie: de seed (`s`) en een waarde(`d_input`). We komen er zo op terug wat deze variabelen betekenen. De functie geeft drie objecten terug. De eerste twee zijn de lists met de counts en de looddikte (zoals je eerder ook terugkreeg), de derde variabele is halfwaardedikte die gebruikt is als input voor de simulatie. Dit noemen we meestal de *true* waarde in simulaties vandaar dat we hem `d_true` noemen. Met de variabele `d_input` kunnen we nu de input waarde van de simulatie controleren. In principe is `d_input` gelijk aan `d_true`, tenzij je de waarde -1 kiest.
Met deze dataset generator gaan we nu de zuiverheid van onze meting bestuderen.
* Kijk eerst eens naar wat de *true* waarde was in je datasets die je hierboven hebt gebruikt! Als je voor `d_input` nu -1 invult krijg je de halfwaardedikte terug die gebruikt is voor het genereren van de 50 datasets die je eerder in deze opdracht hebt gebruikt. Het maakt hierbij niet uit wat voor waarde je aan de seed meegeeft, maar je moet wel iets meegeven, gebruik bijvoorbeeld `s=1`.

- **M2.3f) Hoe groot is de onzuiverheid van ons experiment? Vergelijk hiervoor de gemiddelde bepaalde halfwaardediktes van de 50 experimenten met de `d_true`.**

Nu kun je het gedrag bekijken over meerdere waardes rond de `d_true` waarde.

Pas nu je code aan en varieer de `d_input` waarde bijvoorbeeld met 5 of 10 procent rond je aanvankelijke waarde. Voor elke setting van `d_input` bepaal je over 50 experimenten het gemiddelde van de bepaalde waardes van d_{half}.

- **M2.3g) Zet de gevonden gemiddelde waardes in een grafiek uit tegen de gekozen waardes van `d_input`. Let goed op de leesbaarheid van je grafiek.**
  Tip: Bedenk hoe je de lezer helpt om makkelijk af te lezen waar de zuivere

meting zou liggen (dus als d_half = d_input).

- **M2.3h) Is de onzuiverheid altijd constant of varieert die afhankelijk van de waarde van de halfwaardedikte?**

- **M2.3i) In dit geval simuleren we het experiment. Zou je een methode kunnen bedenken om de onzuiverheid van je experiment te onderzoeken bij een echte meting?**

MODULE III

In deze module bekijken we eerst de Centrale Limietstelling (Hfdst. 14) die verklaart waarom de meeste verdelingen die we in de natuur tegenkomen Normaal verdeeld zijn. Hierna bekijken we in meer detail de Normale verdeling (Hfdst. 15).

Daarna gaan we kijken naar de kleinste kwadraten methode (Hfdst. 16) en de chi-kwadraat (Hfdst. 17) verdeling. Uiteindelijk introduceren we ook methodes om hypotheses (Hfdst. 18) te toetsen.

MODULE III Hoofdstuk 14

De Centrale Limietstelling

Stel dat we een schatter hebben die afhangt van meerdere grootheden. Deze metingen van deze grootheden hebben elk hun eigen onzekerheden en kunnen bijvoorbeeld uniform of Poisson verdeeld zijn. Hoe ziet de verwachte kansverdeling van de schatter er dan uit? In dit hoofdstuk zullen we zien dat schatters die van veel grootheden afhangen vaak de Normale verdeling volgen.

Voorbeeld Twee dobbelstenen gooien. We gooien eerst een enkele (eerlijke) dobbelsteen. De uitkomstenverzameling is $\mathscr{U} = \{1, 2, 3, 4, 5, 6\}$. De kans op elk van deze uitkomsten is gelijk, dat wil zeggen:

$$P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = \frac{1}{6}$$

De uitkomsten verdeling noemen we dus uniform.

We werpen nu twee dobbelstenen en kijken naar de uitkomsten van de twee dobbelstenen samen. De som is gedefinieerd als:

$$W_2 = D_1 + D_2.$$

Waarbij we de uitkomst van elke individuele dobbelsteen hebben weergegeven als D_1 en D_2. De uitkomst van de worp met twee dobbelstenen noteren we hier met W_2.
Je zou misschien denken dat de kansverdeling van de uitkomstenverzameling van het experiment met twee dobbelstenen ook weer uniform is. Immers zijn de onderliggende kansdistributies van D_1 en D_2 beiden uniform. Toch is dat niet zo.
De uitkomsten verzameling van W_2 is:

$$\mathscr{U}_2 = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}.$$

De kans op elk van deze uitkomsten is niet gelijk. Er bestaat bijvoorbeeld maar één manier om de uitkomst $W_2 = 12$ te krijgen, namelijk met $D_1 = 6$ *en* $D_2 = 6$.

Maar bijvoorbeeld de uitkomst $W_2 = 3$ kunnen we op twee manieren verkrijgen: $D_1 = 1, D_2 = 2$ en $D_1 = 2, D_2 = 1$.
De kansverdeling van W_2 is dus niet uniform.

We zien in het voorbeeld met de dobbelstenen, dat de als we twee metingen combineren die elk uniform verdeeld zijn de som van de metingen duidelijk niet uniform verdeeld is.

We werken het voorbeeld van de dobbelstenen nog verder uit.

Voorbeeld Veel dobbelstenen gooien We bekijken in dit voorbeeld hoe de kansdistributies eruitzien van experimenten waarbij we met meerdere dobbelstenen tegelijk gooien. Als we een enkele dobbelsteen gooien dan weten we dat we een uniforme distributie verwachten. Gooien we met twee dobbelstenen dan hebben we in het vorige experiment al gezien dat de kansdistributie van de uitkomstenverzameling van de worp zeker niet uniform is.
Hieronder zie je de kansdistributies voor experimenten waarbij met 1,2,3 en 4 dobbelstenen tegelijk wordt geworpen.

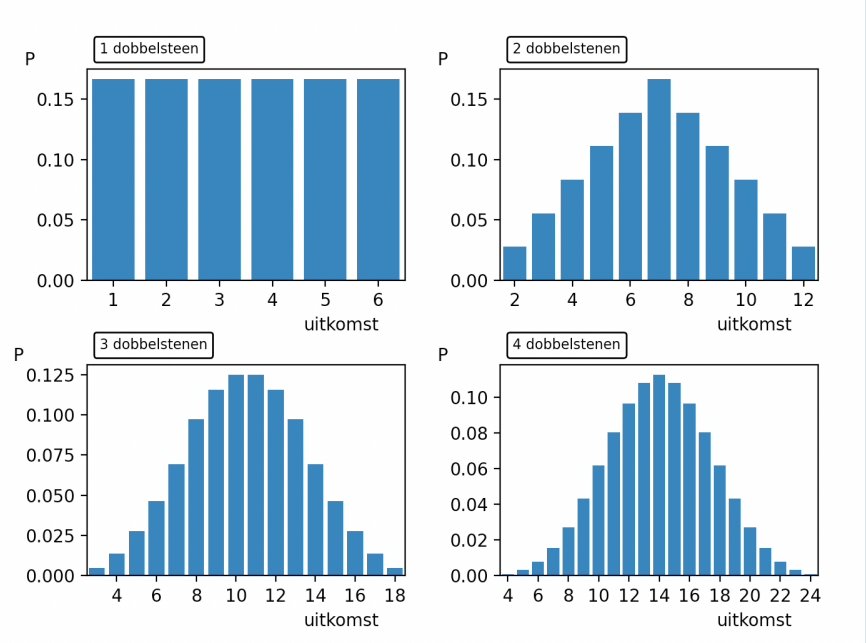


Figure 14.1: De kansdistributie van de uikomstenverzameling van experimenten met 1,2,3 of 4 dobbelstenen.

We zien dat kansdistributie van de uitkomsten voor het experiment met 1 dobbelsteen uniform is. Voor het experiment met 2 dobbelstenen lijkt het op een piramidevorm. Daarna zien we dat de vorm steeds meer een Normaalverdeling benadert.

We hebben in het voorbeeld hierboven gekeken naar het combineren van metingen waarvan de uitkomsten uniform verdeeld zijn. Maar eenzelfde soort resultaat krijg je ook als je bijvoorbeeld meetwaardes die exponentieel verdeeld zijn, of als de meetwaardes elk hun eigen onderliggende verdeling hebben. Zoals bijvoorbeeld als je uniforme kansen combineert met exponentiële.

De **Centrale Limietstelling** zegt dat als je n onafhankelijk stochasten x_j hebt, waarvan elke stochast zijn eigen verdeling heeft met gemiddelde μ_j en variantie σ_j^2, **de som van deze stochasten $\sum_j^n x_j$ een Normaalverdeling zal volgen** met het gemiddelde $\sum_j^n \mu_j$ en de variantie $\sum_j^n \sigma_j^2$, als $n \to \infty$. Hierbij hoeven de populatiegemiddeldes van de stochasten, noch de varianties hiervan (μ_j en σ_j^2) dezelfde te zijn.

De Centrale Limietstelling (Engels: Central Limit Theorem of CLT) zegt dat als we $n \to \infty$ stochasten optellen, de som van deze stochasten een Normaalverdeling zullen volgen. Het maakt hierbij niet uit wat voor vorm de kansverdelingen de stochasten hebben, ze kunnen exponentieel, uniform, Normaal verdeeld zijn om welke vorm dan ook hebben.

Er is één voorwaarde en dat is dat de onderliggende verdelingen een gedefinieerd gemiddelde en eindige variantie moeten hebben. Dat is een belangrijke voorwaarde. Wiskundig kun je laten zien dat bijvoorbeeld stochasten die volgens de Cauchy of Landau verdeeld zijn bij combinatie geen Normaalverdeling opleveren. Toch is die beperking niet heel groot. In de natuur zijn praktisch alle stochastische verdelingen beperkt en voldoen dus aan de Centrale Limietstelling.

De convergentiesnelheid van de distributie naar de Normaalverdeling hangt af van de onderliggende stochastische verdelingen. Bijvoorbeeld hoef je minder uniform verdeelde stochasten bij elkaar op te tellen om een Normaalcurve te benaderen dan stochasten die van zichzelf exponentieel verdeeld zijn.

De Centrale Limietstelling is zonder meer de meest belangrijke stelling in de statistiek. De Centrale Limietstelling verklaart waarom we in de natuur zoveel parameters vinden die Normaal zijn verdeeld.

Het bewijs van deze stelling is bijzonder ingewikkeld en zullen we hier niet behandelen. Eventueel kun je hier verder lezen over de bewijs stelling.

Twee leuke video's die de Centrale Limietstelling illustreren kun je hier en hier vinden.

Meestal is er in de natuur of in experimenten sprake van een combinatie van een grote hoeveelheid toevalligheden die een rol speelt bij de onzekerheid van een meting. Hetzelfde geldt vaak voor de eigenschappen van een populatie, de natuurlijke verdeling van deze eigenschappen zijn vaak ook Normaal verdeeld om dezelfde reden.

Denk maar eens aan de vorming van een zandkorrel of van een ster. Het is dan begrijpelijk dat de sterren in een bolhoop een Normale massa verdeling kennen. Of de grootte van de zandkorrel op een strand. Bij de vorming van een ster of zandkorrel zijn er vele toevalligheden die invloed hebben op bijvoorbeeld de grootte van zo'n object.

14.1 Overeenkomsten tussen de Poisson en de Normale verdeling

Als laatste bekijken we nogmaals de Poisson verdeling en laten we in het voorbeeld hieronder zien dat voor grotere waardes van λ de Poisson steeds meer overeenkomt met een Normale verdeling.

De Poisson onzekerheid zien we veel terug omdat het de onzekerheid op telexperimenten beschrijft, en veel van de metingen die we uitvoeren zijn telexperimenten. Voor een verwachtingswaarde van λ vinden we een standaardafwijking van $\sqrt{\lambda}$ en zoals we al eerder hebben gezien mogen we deze bij het uitvoeren van een experiment vaak zien als de onzekerheid op de verwachtingswaarde zelf.

We herhalen de formule van de Poisson vergelijking hier:

$$P(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{14.1}$$

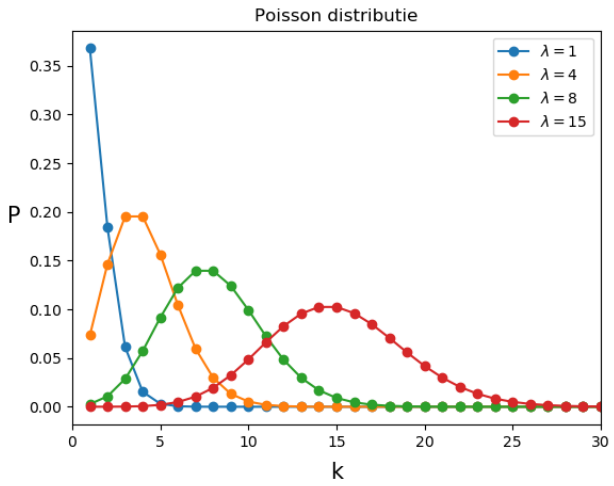


Figure 14.2: De Poisson distributie.

Zoals we in Fig. 14.2 zien is de Poisson verdeling asymmetrisch. Vooral voor lage waardes van λ is dit goed zichtbaar. Voor grotere waardes van λ zien we dat de verdeling steeds meer symmetrisch is en ook steeds meer overeenkomsten vertoont met een Normale verdeling. Dit komt door de Centrale Limietstelling. **We verwachten voor grotere waardes van λ dat de Poisson de Normale verdeling zal gaan volgen.** We leggen dit uit aan de hand van een voorbeeld.

Voorbeeld Stel dat we in een experiment tellingen uitvoeren per tijdsinterval. De meetwaarde die we vinden in één tijdsinterval, k, is het aantal waarnemingen per

tijdsinterval Δt. We zouden het tijdsinterval ook kunnen opdelen in bijvoorbeeld 100 stukjes. We verwachten nu $k_{\text{kort}} = k/100$ waarnemingen te meten per tijdsinterval van $\Delta t_{\text{kort}} = 1/100 \times \Delta t$. Deze kortere tijdsintervallen volgen ook de Poisson statistiek, immers we hebben de voorwaardes van het experiment veranderd, maar het is nog steeds een telexperiment.

Rachid en Belia voeren een telexperiment tegelijkertijd op twee manieren uit. Rachid noteert 100 uitkomsten elke 10 secondes en Belia wacht precies 1000 secondes en noteert precies 1 uitkomst. De metingen van Rachid noemen we r_i en de meting van Belia noemen we b. De 100 metingen van Rachid zijn 100 onafhankelijke stochasten. Dit omdat je verwacht dat de meetwaardes statistische fluctuaties hebben (stochastisch) en omdat de ene meting geen invloed heeft op de volgende meting (onafhankelijk).

De fluctuaties volgen de Poisson statistiek (telexperiment). Als de verwachting op 1 enkele meting gelijk is aan λ_r, dan is de spreiding (variantie) van de punten ook gelijk aan λ_r. De standaardafwijking is dus gelijk aan $\sqrt{\lambda_r}$. Kijk hiervoor ook nog eens naar het hoofdstuk Kansdichtheidsfuncties (Hfdst. 6). Voor de meting van Belia verwachten we als uitkomst λ_b met een standaardafwijking $\sqrt{\lambda_b}$.

We weten in dit experiment dat precies geldt dat $\sum_i^{100} r_i \equiv b$. Immers Rachid en Belia zitten naast elkaar en nemen dezelfde getallen waar. Rachid telt zijn 100 uitkomsten bij elkaar op en de Centrale Limietstelling zegt nu dat de som van deze 100 stochasten de Normale verdeling zal benaderen. Ofwel de verwachting van de som van de 100 uitkomsten van Rachid heeft de waarde $\sum_i^{100} \lambda_r$ en de standaardafwijking op deze verwachting is, redelijk, Normaal verdeeld (niet exact omdat n is groot maar nog geen ∞).

De verwachtingswaarde van de uitkomst van het experiment van Belia zoals eerder gezegd gelijk aan λ_b en deze moet overeenkomen met de verwachtingswaarde van de som van de 100 uitkomsten van Rachid. De onzekerheid op de verwachte uitkomst van het experiment van Belia hoort de Poisson verdeling te volgen (want telexperiment), maar we zien nu ook dat de som van de uitkomsten van Rachid de Normaalcurve zal benaderen. Hieruit kunnen we concluderen dat voor stochasten die de Poisson verdeling volgen we verwachten dat de Poisson verdeling steeds meer de Normaalcurve zal benaderen voor hogere waardes van λ.

We illustreren de gelijkenis van de Poisson en Normale verdeling door de twee functies over

elkaar heen weer te geven voor een waarde van $\lambda = 60$. Deze vergelijken we nu met de Normaalverdeling met $\mu = 60$ en $\sigma = \sqrt{60}$. Zie figuur 14.3.

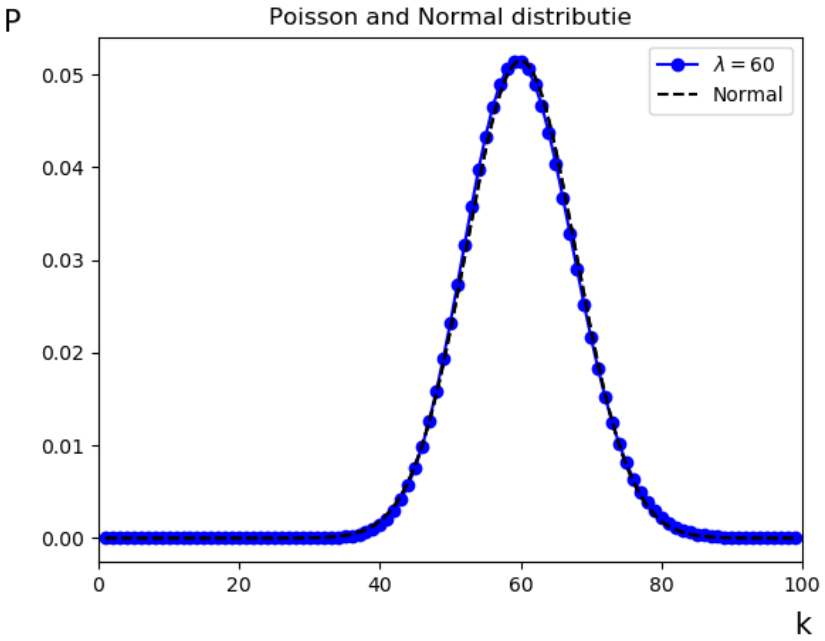


Figure 14.3: Vergelijking tussen de Poisson en de Normaal distributie.

Het is duidelijk dat de overeenkomst tussen de Poisson verdeling en de Normale verdeling bij deze waarde van λ al heel groot is.

Er blijven natuurlijk verschillen, zo is de Poisson verdeling een discrete verdeling een standaardafwijking die afhangt van de verwachtingswaarde. Maar de grote gelijkenis verklaart wel waarom we, voor grotere waardes van λ, gebruik mogen maken van vergelijkingen die eigenlijk alleen voor de Normale verdeling gelden. Zoals bijvoorbeeld de regels voor de foutenpropagatie.

MODULE III Hoofdstuk 15

De Normaalverdeling

We hebben in het hoofdstuk De Centrale Limietstelling (Hfdst. 14) gezien waarom onzekerheden op metingen zo vaak Normaal zijn verdeeld. Het is nu duidelijk dat de Normaalverdeling een belangrijke rol speelt in de statiek. In dit hoofdstuk bekijken we nogmaals de Normaalverdeling en introduceren we de zogeheten z-score methode die we later gaan toepassen bij de χ^2-methode en bij het toetsen van hypotheses.

15.1 De Normaalverdeling

Allereerst herhalen we de formule die jullie ook al in Module 1 hebben gezien. De normaalverdeling is gedefinieerd als:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}. \tag{15.1}$$

De functie heeft twee parameters, μ en σ. De verwachtingswaarde van de normaal verdeling is precies μ en de standaardafwijking is precies gelijk aan σ. (De notering is niet toevallig!)

In de figuur hieronder, 15.1, zie je enkele voorbeelden van de Normale verdeling voor verschillende waardes voor μ en σ.

Er is geen relatie tussen de het gemiddelde μ en de standaardafwijking σ, lage waardes van μ kunnen een grotere of kleinere standaardafwijking hebben. (Anders dan bij de Poisson verdeling.) We zien dat voor hogere waardes voor σ de datapunten meer verspreid zijn.

Voorbeeld Stel nu dat we een meting doen L en we kennen het populatiegemiddelde $\mu_L = 10.0$ cm met een spreiding van $\sigma_L = 2.0$ cm. De kans dat we een meting doen die $L = 4.0$ cm oplevert is dan niet zo groot. Als de spreiding rond het populatiegemiddelde daarentegen groter is, bijvoorbeeld $\sigma = 5.0$ cm dan is de kans veel groter dat de meting een waarde van $L = 4.0$ cm oplevert.

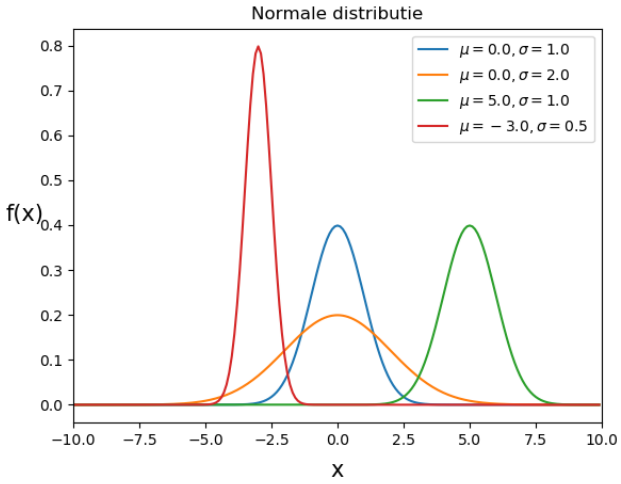


Figure 15.1: De Normaalverdeling.

Als de uitkomsten uit een experiment Normaal verdeeld zijn, en we kennen μ en σ, dan kunnen we de kans op een bepaalde uitkomst exact berekenen. Hoe grotere de afstand met μ, hoe kleiner de kans op dat meetresultaat. Hoe groter de standaardafwijking σ, hoe meer verpreid de meetuikomsten zijn en hoe groter de kans op een grotere afstand ten op zichte van het gemiddelde μ. Deze kansen kunnen we exact berekenen met behulp van de Normaalverdeling.

15.2 Z-score en waarschijnlijkheden

Om de kans op een bepaalde meetuikomst uit te drukken maken we gebruik van de oppervlaktes onder de Normaalverdeling. Dit kunnen we schematisch weergeven.

Het oppervlak onder de Normaalkromme behorende bij de kans om een waarde $X < x$ te vinden, kun je als volgt schematisch weergeven , zie figuur 15.2 . Dit noemen we ook wel de *linkszijdige overschrijding* en we berekenen de *onderkans*.

Het oppervlak onder de normaalkromme behorende bij de kans om een waarde $X > x$ te vinden, is hier schematisch weergegeven , zie figuur 15.3 . Dit noemen we ook wel de *rechtszijdige overschrijding* en we berekenen de *bovenkans*.

Het oppervlak onder de kromme van een Normaalverdeling is lastig uit te rekenen, zie bijvoorbeeld de uitleg op wikipedia. We maken hierom een tussenstap en berekenen eerst de zogenoemde z-score. Stel een dataset is Normaal verdeeld met gemiddelde μ en standaardafwijking σ, de z-score, voor een bepaalde gemeten waarde x, is dan gelijk aan:

$$Z = \frac{x - \mu}{\sigma}. \tag{15.2}$$

Het oppervlak onder de Normaalkromme, behorende bij de kans op een bepaalde waarde,

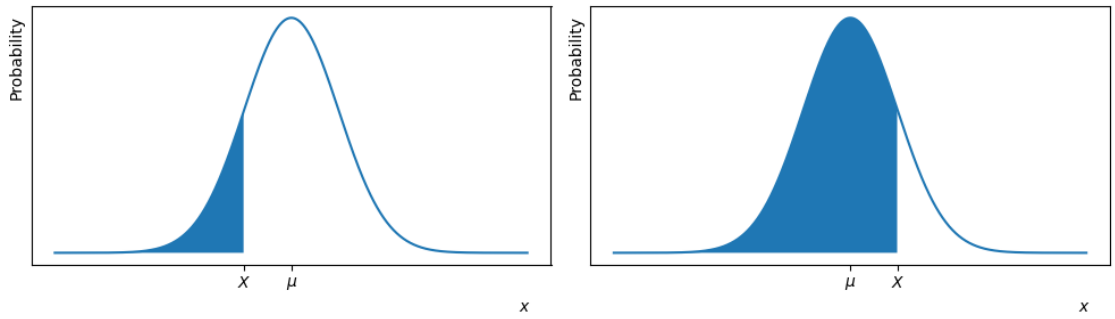


Figure 15.2: Het oppervlak onder de normaalkromme behorende bij de kans om een waarde $X < x$ te vinden.

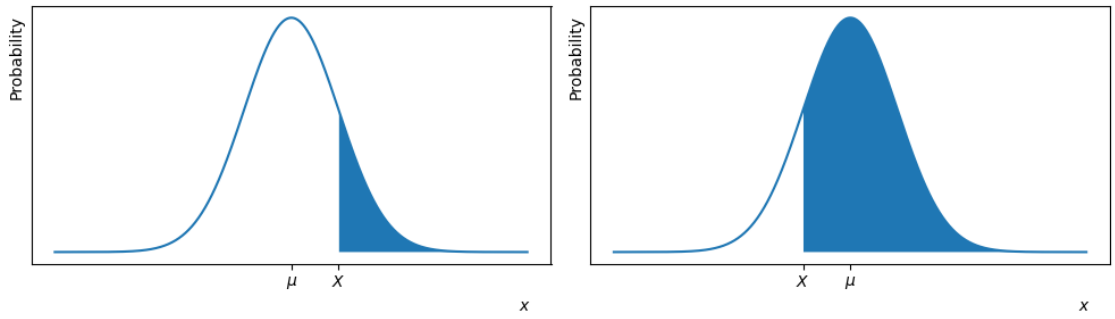


Figure 15.3: Het oppervlak onder de normaalkromme behorende bij de kans om een waarde $X < x$ te vinden

hangt op de volgende manier van de z-score af.

De kans om een waarde $X < x$ te vinden is gelijk aan:

$$P(X < x) = P\left(Z < \frac{x - \mu}{\sigma}\right) \tag{15.3}$$

De kans om een waarde $X > x$ te vinden is gelijk aan:

$$P(X > x) = 1 - P(X < x) = 1 - P\left(Z < \frac{x - \mu}{\sigma}\right) \tag{15.4}$$

Dit kun je zelf nagaan door schetsen te maken van de bijbehorende oppervlakken onder de normaalkromme.

Als je de z-score hebt berekend, kun je uit een voorberekende tabel aflezen wat de bijbehorende overschrijdingskans is.

Hieronder laten we in twee voorbeelden zien hoe je deze methode toepast.

Voorbeeld 1: Een stochast X is Normaal verdeeld met gemiddelde $\mu = 20$ en standaardafwijking $\sigma = 2$. De kans op een waarde $X < 16$ is nu gelijk aan

$$\begin{aligned} P(X < 16) &= P\left(Z < \frac{x-\mu}{\sigma}\right) \\ &= P\left(Z < \frac{16-20}{2}\right) \\ &= P(Z < -2). \end{aligned}$$

Dit is een linkszijdige overschrijding. In de z-score tabel kunnen we nu de bijbehorende kans waarde opzoeken. Dit is een waarde van 0.02275. We schrijven dus

$$P(X < 16) = P\left(Z < \frac{x-\mu}{\sigma}\right) = 0.02275.$$

Er is in dit geval dus een kans van 0.02 dat we bij de gegeven dataset een waarde onder de 15 zullen vinden.

Voorbeeld 2: Een stochast X is Normaal verdeeld met gemiddelde $\mu = 20$ en standaardafwijking $\sigma = 2$, de kans op een waarde $X > 22$ is nu gelijk aan

$$\begin{aligned} P(X > 22) &= 1 - P(X < 22 >) \\ &= 1 - P\left(Z < \frac{x-\mu}{\sigma}\right) \\ &= 1 - P\left(Z < \frac{22-20}{2}\right) \\ &= 1 - P(Z < 1). \end{aligned}$$

Dit is een rechtszijdige overschrijding. In de z-score tabel kunnen we nu de bijbehorende kans waarde opzoeken. Dit is een waarde van 0.84134. We schrijven dus

$$P(X > 22) = 1 - P\left(Z < \frac{x-\mu}{\sigma}\right) = 1 - 0.84134 = 0.15866.$$

Er is in dit geval dus een kans van 0.16 dat we bij de gegeven dataset een waarde boven de 22 zullen vinden.

MODULE III Hoofdstuk 16

De Kleinste-Kwadraten Methode

In labexperimenten meten we vaak een bepaalde grootheid waarbij we een andere grootheid variëren. Zo krijgen we bijvoorbeeld datapunten van een observabele y waarbij we een andere grootheid x variëren. Meestal doen we dit als we een bepaald verband verwachten tussen de twee grootheden.

In dit hoofdstuk introduceren we een krachtige methode om onbekende parameters te schatten uit het verband van de gevonden meetwaardes.

Deze methode heet de kleinste kwadraten methode en wordt ook wel lineaire regressie, of χ^2-fitten genoemd. De methode kan wiskundig worden afgeleid met behulp met 'maximale waarschijnlijkheid principes'.

16.1 De kleinste-kwadraten methode

Een van de meest krachtige schatters is de methode van de kleinste-kwadraten. Met de kleinste kwadraten methode minimaliseren we het kwadratisch verschil tussen een set metingen en de voorspelde waardes op die metingen, waarbij de voorspelling afhangt van één of meerdere parameters. De voorspelling kunnen we uitdrukken in een functie.

Voorbeeld: Meetwaardes We beginnen met een voorbeeld. Stel dat we een set metingen hebben die er als volgt (Fig. 16.1 uitziet.

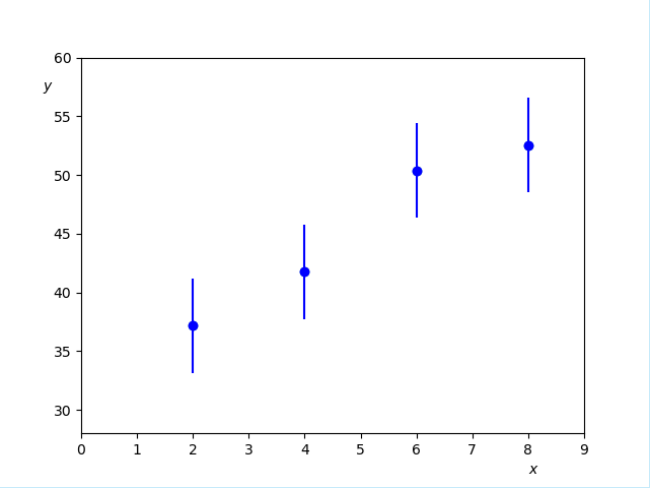


Figure 16.1: Een set metingen.

We vermoeden een lineair verband tussen de grootheden x en y met parameters a en b.

De parameters van de functie θ_i willen we nu afschatten met behulp van de metingen. De geschatte waardes van θ_i noemen we $\hat{\theta}_i$. Oftewel, $\hat{\theta}_i$ zijn de waardes van parameters θ waarbij de functie de meetwaardes optimaal beschrijft.

Voorbeeld: Lineair verband In het figuur 16.2, zien we twee voorbeelden van oplossingen van een functie $y = a + b \cdot x$, de rode lijn en de gestreepte zwarte lijn. De vraag is nu hoe bepaal je welke combinatie van de paramaters a en b het beste de data beschrijft?

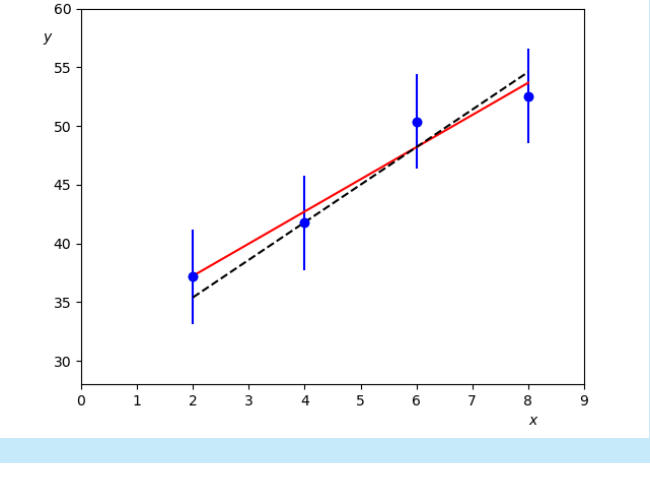


Figure 16.2: Een set metingen met twee lijnen.

Als we een schatter kunnen definïeren voor onze functie dan vinden we de geschatte optimale waardes van a en b . Deze noemen we $\hat{a}$ en $\hat{b}$. Oftewel, $\hat{a}$ en $\hat{b}$ zijn de waardes van a en b waarbij de lineaire functie onze dataset optimaal beschrijft. Dit schatten van de waardes van $\hat{a}$ en $\hat{b}$ noemen we ook wel fitten.

We introduceren nu een definitie van de χ^2 schatter. Deze methode noemen we ook wel de kleinste kwadraten methode.

Stel dat we een functie $f(x;\theta)$ hebben die waardes van y voorspelt. En we hebben een dataset met n waardes voor $x : x_1, x_2, ..., x_n$ met corresponderende waardes voor $y : y_1, y_2, ..., y_n$ waarbij elke waarde van y gemeten is met precisie σ_i. Nu kunnen we de som nemen van het kwadratische verschil van alle punten in de dataset met de voorspelde waardes $f(x_i;\theta)$, geschaald met de onzekerheden σ_i. Deze som noemen we χ^2 :

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i;\theta)}{\sigma_i} \right)^2 . \tag{16.1}$$

Dit is de definitie van de χ^2 schatter.

De meest optimale waarde voor θ geeft de kleinste χ^2. Door de χ^2 te minimaliseren vinden we de optimale schatting $\hat{\theta}$.

Door het kwadraat te gebruiken en niet het absolute verschil tussen de datapunten en de voorspelling geven we meer waarde aan de punten die ver van de voorspelling afliggen.

Voorbeeld: Afstanden In onderstaande figuur is elk datapunt van de meetset apart weergeven. Voor elke meetwaarde zien we een normaalverdeling die gecentreerd is op de meetwaarde met als breedte de betreffende meetfout. Ook zijn in de grafiek de voorspelde waardes weergegeven met pijlen die volgen uit de functie met de gepostuleerde waardes van a en b. De rode (zwart gestreepte) pijlen horen bij de rode (zwart gestreepte) lijn uit het figuur (16.2) uit het vorige voorbeeld.

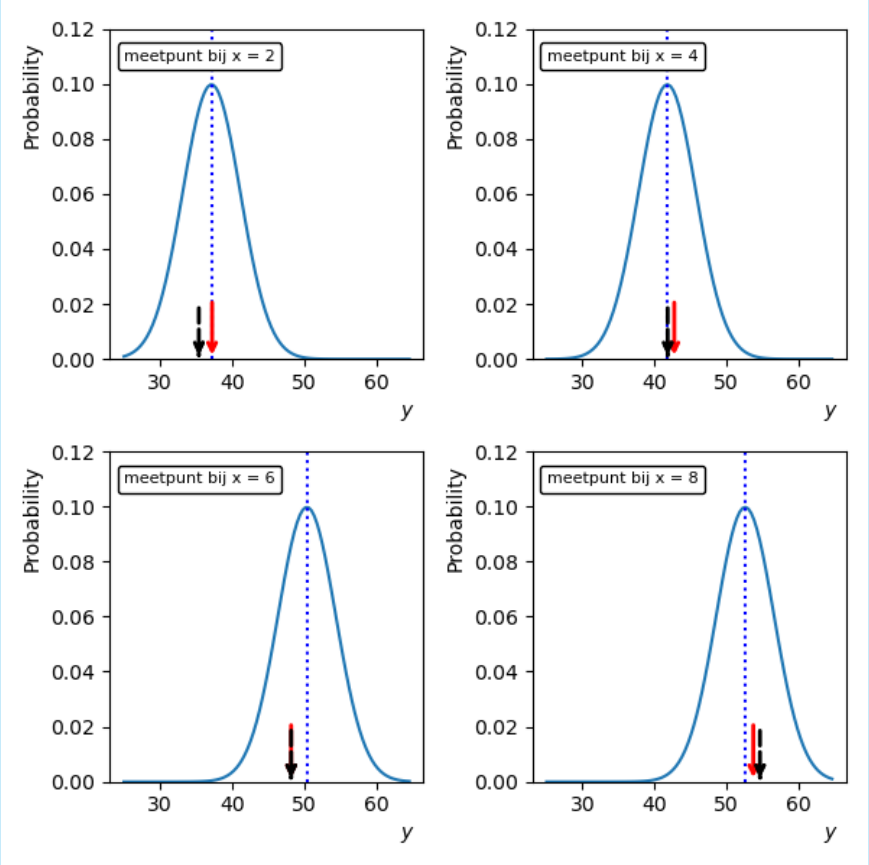


Figure 16.3: Een set metingen met twee lijnen.

De afstanden tussen de voorspelde waardes en de gemeten waardes zijn de ingredïenten van de kleinste kwadraten methode.

In de meeste gevallen kunnen we het minimum van de χ^2 algebraïsch vinden. Als we nu kijken naar een functie die afhangt van slechts één parameter a dan kunnen we het minimum vinden op het punt dat de afgeleide gelijk is aan nul:

$$\frac{\partial \chi^2}{\partial a} = 0. \tag{16.2}$$

Dit geeft:

$$\frac{\partial \chi^2}{\partial a} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \frac{\partial f(x_i; a)}{\partial a} \left(y_i - f\left(x_i; a\right)\right) = 0. \tag{16.3}$$

De betreffende waarde van a waarvoor dit geldt is de schatter van a, genoteerd als $\hat{a}$.

In de vergelijkingen hierboven hebben we maar één afhankelijke parameter gezien maar dit principe kun je ook toepassen op functies met meerder afhankelijke parameters die je dan tegelijkertijd oplost.

Een andere, niet analytische oplossing kan gevonden worden met een computerprogramma door de χ^2 voor veel waardes van a en b uit te rekenen en uit deze set van waardes het punt met de laagste χ^2 te bepalen. Uiteraard werkt dat ook voor functies met meerder onbekende (ook wel vrije) parameters kennen.

Twee filmpjes die het principe van de kleinste kwadraten goed illustreren vind je hier en hier.

Om in te schatten **hoe goed** je fit gelukt is moeten we eerst meer weten over de χ^2-distributie. Daar gaat het volgende hoofdstuk over.

In opgave M3.1 ga je het principe van de kleinste kwadraten toepassen.

MODULE III Hoofdstuk 17

De χ^2 distributie

We hebben in het vorige hoofdstuk over de kleinste-kwadraten methode de definitie van de χ^2 schatter gezien. De χ^2 is een maat voor het verschil tussen de voorspelde en de gemeten waardes. Als een functie f de data goed beschrijft voor de geoptimaliseerde parameters van de functie, dan zal de χ^2 klein zijn. Als de χ^2 dus groot blijft na het optimaliseren van de parameters van f dan is er iets misgegaan. Het kan zijn dat de functie f de datapunten niet goed *kan* beschrijven, maar het kan ook zijn dat als je minimalisatie uitvoert met een computer, deze het minimum niet goed heeft weten te vinden.

Als daarentegen de χ^2 heel klein is gaat er ook iets mis. Waarschijnlijk heb je de onzekerheden op de datapunten heel erg overschat.

Maar wat is precies heel groot of heel klein? Wat is de verwachtingswaarde van de χ^2? Deze vragen gaan we in dit hoofdstuk beantwoorden.

17.1 De χ^2-toets

We hebben gezien in het hoofdstuk over de kleinste kwadraten methode, dat de χ^2 gedefinieerd is als het kwadratische gewogen verschil tussen de meetwaardes en de voorspelde waardes:

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i; \hat{a}, \hat{b}, ..)}{\sigma_i} \right)^2 . \tag{17.1}$$

Let op dat we hier de geoptimaliseerde parameters $(\hat{a}, \hat{b}, ...)$ van de functie hebben ingevuld. Deze waarde voor χ^2 is dus al geminimaliseerd voor de parameters van f.

De χ^2 verdeling zelf is een kansdichtheidsverdeling, en voldoet dus ook aan de voorwaardes hiervan. Dat wil zeggen dat het oppervlakte onder de χ^2-curve is genormaliseerd. De functie ziet er als volgt uit:

$$P(\chi^2;\nu) = \frac{2^{-\nu/2}}{\Gamma(\nu/2)}\chi^{n-2}e^{-\chi^2/2}. \tag{17.2}$$

De Γ in de noemer is een speciale wiskundige functie. Deze zal pas in jullie tweede jaar volledig worden uitgelegd. Op dit moment kun je hem simpelweg interpreteren als een functie waar een normalisatie term uitkomt. Het is best een gekke functie, voorbeelden van uitkomsten: $\Gamma(1/2) = \sqrt{\pi}$, $\Gamma(1) = 1$ en $\Gamma(3/2) = 1/2\sqrt{\pi}$. Als je al meer wilt weten over de Γ-functie dan kun je daar bijvoorbeeld hier meer over lezen.

Zoals je ziet hangt de χ^2 kans ook af van een parameter ν, dit is het aantal meetpunten, n, gereduceerd met het aantal parameters van de functie f. We noemen ν het aantal **vrijheidsgraden** (Engels: degrees of freedom).

Voorbeeld: Bepalen van het aantal vrijheidsgraden: Stel we hebben 10 meetwaardes en we gebruiken de kleinste kwadraten methode om 2 parameters van een functie f te optimaliseren. We hebben dan $\nu = 10 - 2 = 8$ vrijheidsgraden.

Hier in figuur 17.1 zie je hoe de χ^2-curve eruit ziet voor verschillende waardes van ν.

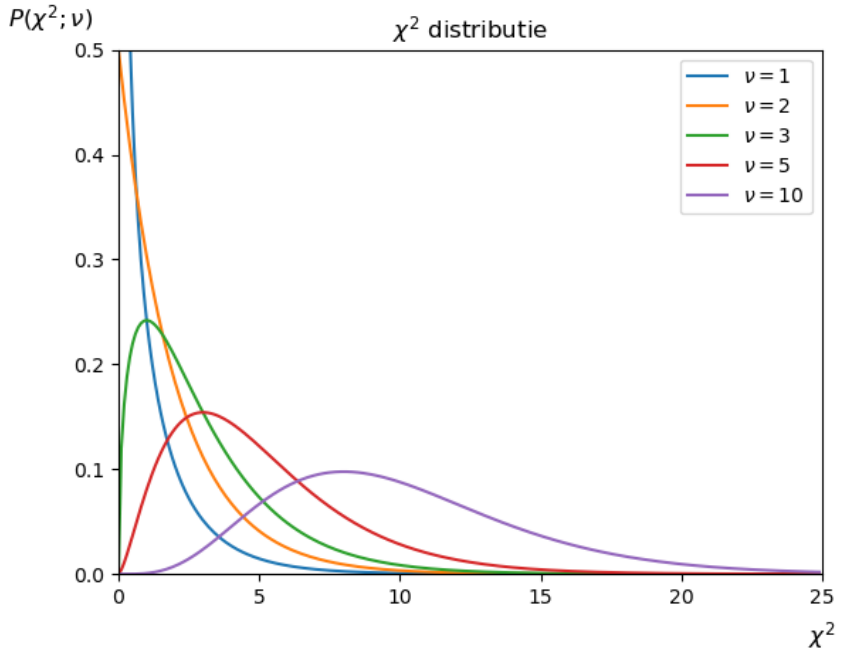


Figure 17.1: De χ^2 verdeling.

De χ^2 distributie heeft een verwachtingswaarde $\mu = \nu$ en een variantie van $var(\chi^2) = 2 \cdot \nu$.

Voor een gefitte functie met ν vrijheidsgraden verwachten we dus een waarde voor de χ^2 te vinden die gelijk is aan het aantal vrijheidsgraden.

Met behulp van de χ^2-curve kunnen we de overschrijdingskansen uitrekenen en aangeven hoe waarschijnlijk het is dat een functie f met geoptimaliseerde parameters $\hat{a}, \hat{b}, ...$ de waarnemingen uit het experiment beschrijft. Je kan nu de overschrijdingskansen voor verschillende waardes van χ^2 en vrijheidsgraden ν bepalen, bijvoorbeeld met behulp van deze tabel (Hfdst. A).

Het is gemakkelijker om de waarde van de χ^2 direct te delen door het aantal vrijheidsgraden. De verwachtingswaarde voor de ratio χ^2/ν is dan altijd gelijk aan 1 en de variantie is gelijk aan $var(\chi^2/\nu) = 2/\nu$. (Deze laatste stap kan je controleren door toepassing van de regels van de foutenpropagatie.) We definïeren de **gereduceerde** χ^2 als:

$$\chi^2_\nu = \frac{\chi^2}{\nu}. \tag{17.3}$$

De gereduceerde χ^2 wordt ook wel geschreven als χ^2/df, χ^2_{red} of $\tilde{\chi}^2$. Je kan met de gereduceerde χ^2_ν ook zonder de tabel al makkelijk inschatten of de fit aan de χ^2 toets voldoet. Namelijk je verwacht dan een waarde van ongeveer 1.

Als nu χ^2_ν veel afwijkt van 1 dan is het waarschijnlijk dat er een probleem is met de fit. Het kan zijn dat de functie de relatie tussen de datapunten niet goed beschrijft, of dat er iets mis is met de onzekerheden op de datapunten.

Doorgaans betekent een veel te kleine gereduceerde chi-kwadraat ($\chi^2_\nu \ll 1$) dat de onzekerheden op de meetwaardes overschat zijn. Een te grote waarde ($\chi^2 \gg 1$) betekent meestal dat de functie de datapunten niet goed kan beschrijven of dat de onzekerheden zijn onderschat.

17.2 Akaike Informatie Criterium

Stel dat je een dataset hebt waarvan je niet zeker weet door welke functie deze wordt beschreven. Je probeert twee functies uit, f_1 en f_2. Voor beide functies schat je de beste waardes voor de parameters waar de functies van afhangen. De geschatte χ^2 waardes noemen we dan χ^2_1 en χ^2_2. Als algemene vuistregel geldt dat de functie met de kleinste geminimaliseerde χ^2_ν de data het beschrijft. Als in dat geval de betreffende χ^2_ν dicht bij 1 ligt werkt deze vuistregel goed.

Voorbeeld 1 Stel dat we een dataset hebben met 10 gemeten waardes. We proberen twee functies uit:

$$f_1(x; a, b) = a \cdot x + b$$

en

$$f_2(x; a) = a \cdot x.$$

Als geminimaliseerde χ^2 voor de twee functies vinden we: $\chi^2_1 = 4.0$ en $\chi^2_2 = 13.0$.
De χ^2 per vrijheidsgraad is voor de twee functies:

$$\chi^2_1/\text{vrijheidsgraad} = 4.0/(10-2) = 0.5$$

en

$$\chi^2_2/\text{vrijheidsgraad} = 13.0/(10-1) = 1.44.$$

Op basis van de vuistregel zou je functie f_1 kiezen.

Voorbeeld 2 Stel dat we een dataset hebben met 10 gemeten waardes. We proberen twee functies uit:

$$f_1(x; a, b) = a \cdot x + b$$

en

$$f_2(x; a) = a \cdot x$$

Als geminimaliseerde χ^2 voor de twee functies vinden we: $\chi^2_1 = 6.0$ en $\chi^2_2 = 9.0$.
De χ^2 per vrijheidsgraad is voor de twee functies:

$$\chi^2_1/\text{vrijheidsgraad} = 6.0/(10-2) = 0.75$$

en

$$\chi^2_2/\text{vrijheidsgraad} = 9.0/(10-1) = 1.0.$$

Op basis van de vuistregel zou je functie f_1 kiezen.

Als de gereduceerde chi-kwadraat echter veel kleiner is dan 1 dan kun je betwijfelen of de bijbehorende functie wel echt de beste is. Je zou dan de waarde kunnen kiezen die het dichtste bij 1 bevindt. Meestal komt die wel goed uit, maar het hangt erg af van de verschillen tussen de twee functies, voor het aantal vrijheidsgraden speelt hier een rol.

Beter is om dan het Akaike Informatie Criterium kun je gebruiken om uit te vinden welke functie het beste aan een dataset fit. Stel dat je een dataset hebt waarbij je n meetwaardes hebt die je beschreven hebt met een functie met p vrije parameters met een geminimaliseerde χ^2. Dan heeft het Akaike Informatie Criterium de volgende waarde:

$$AIC = \chi^2 + 2p + \frac{2p(p+1)}{n-p-1}. \tag{17.4}$$

Als we deze AIC berekenen voor beide functies dan is de functie met de laagste AIC de meest optimale.

Voorbeeld 1 Stel dat we een dataset hebben met 10 gemeten waardes. We proberen twee functies uit:

$$f_1(x; a, b) = a \cdot x + b$$

en

$$f_2(x; a) = a \cdot x.$$

Als geminimaliseerde χ^2 voor de twee functies vinden we: $\chi_1^2 = 4.0$ en $\chi_2^2 = 13.0$.
De AIC waarde voor de twee functies zijn nu:

- $AIC_1 = 4.0 + 4 + 12/7 = 9.7$
- $AIC_2 = 13.0 + 2 + 4/8 = 15.5$.

Op basis van het Akaike Informatie criterium zou je functie f_1 kiezen.

Voorbeeld 2 Stel dat we een dataset hebben met 10 gemeten waardes. We proberen twee functies uit:

$$f_1(x; a, b) = a \cdot x + b$$

en

$$f_2(x; a) = a \cdot x.$$

Als geminimaliseerde χ^2 voor de twee functies vinden we: $\chi_1^2 = 6.0$ en $\chi_2^2 = 9.0$.
De χ^2 per vrijheidsgraad is voor de twee functies:

- $AIC_1 = 6.0 + 4 + 12/7 = 11.8$
- $AIC_2 = 9.0 + 2 + 4/8 = 11.5$

Op basis van de vuistregel zou je functie f_2 kiezen.

MODULE III Hoofdstuk 18

Hypothese toetsen

Als je een steekproef hebt genomen en je wil hiermee iets kunnen zeggen over de populatie dan moet er ook nagegaan worden in hoeverre de steekproef ons idee over de populatie ondersteund.

Dit wordt hypothese toetsen genoemd. Bij hypothese toetsen doorloop je de volgende stappen:

1. Hypothese opstellen
2. Significantieniveau kiezen (*Let op! Dit is iets anders dan de significantie waarin je een meetwaarde noteert.*)
3. p-waarde bepalen
4. Conclusie trekken

Deze stappen worden hieronder toegelicht.

18.1 Hypothese opstellen

Een hypothese is een uitspraak over een bepaalde eigenschap van een populatie. Je weet nog niet of deze uitspraak correct is. Een hypothese wordt geformuleerd als stelling.

Voorbeelden van hypotheses

- 20% van de auto's in Nederland is blauw.
- 50% van de Nederlanders heeft blauwe ogen
- De valversnelling heeft als waarde in Nederland 9.81 ms^{-2}
- Studenten in Amsterdam halen hogere cijfers dan studenten in Groningen

Bij hypothese toetsen is er sprake van twee hypotheses. De zogenoemde *nulhypothese* en de *alternatieve hypothese*.

Bij hypothese toetsen wordt eerst aangenomen dat de eigenschap die onderzocht wordt niet waar is. Dit wordt de *nulhypothese* genoemd. De stelling dat de gewenste eigenschap wel waar is wordt de *alternatieve hypothese* genoemd. De nulhypotese wordt aangegeven met H_0, de alternatieve hypothese met H_α (ook H_1 is veelvoorkomend).

De procedure bij hypothese toetsen is dat je in eerste instantie aanneemt dat de eigenschap niet waar is (dus je houdt de nulhypothese aan) en dan onderzoekt of dit stand houdt in het kader van de gevonden resultaten. Uiteindelijk hoop je dat je de nulhypothese kunt verwerpen waardoor de alternatieve hypothese (en dus de gewenste waarde van de eigenschap) kunt aannemen.

De term nulhypothese komt overigens uit het Engels van de 'null hypothesis' en de naamgeving slaat op de hypothese die verworpen (oftewel 'nullified') moet worden.

Dus:

- Alternatieve hypothese H_α: De hypothese die zegt wat je verwacht te vinden in de dataset
- Nulhypothese: Het omgekeerde van de alternatieve hypothese.

Onderstaand de eerdere hypothesen met bijbehorende nulhypothesen:

Voorbeelden van alternatieve hypothese en nulhypothese

- H_α: 20% van de auto's in Nederland is blauw.
- H_0: Het percentage blauwe auto's in Nederland is geen 20%.

- H_α: Meer dan 20% van de auto's in Nederland is blauw.
- H_0: Minder dan 20% van de auto's in Nederland is blauw.

- H_α: Het percentage Nederlanders met blauwe ogen is 50%.
- H_0: Het percentage Nederlanders met blauwe ogen is geen 50%.

- H_α: De valversnelling heeft als waarde in Nederland 9.81 ms^{-2}.
- H_0: De valversnelling in Nederland is niet gelijk aan 9.81 ms^{-2}.

- H_α: Studenten in Amsterdam halen hogere cijfers dan studenten in Groningen
- H_0: De studenten in Amsterdam halen lagere cijfers dan de studenten in Groningen.

- H_α: Het aantal katten in Nederland is groter dan 20 000
- H_0: Het aantal katten in Nederland is kleiner of gelijk aan 20 000

- H_α: Het percentage mensen over de gehele wereld met een hond is kleiner dan 40%
- H_0: Het percentage mensen over de gehele wereld met een hond is groter of gelijk aan 40%

In alle bovenstaande gevallen is het dus de procedure om te kijken of we genoeg bewijs hebben om de nulhypothese te kunnen verwerpen zodat we de alternatieve hypothese kunnen aannemen.

18.2 Significantieniveau kiezen

De volgende stap in hypothese toetsen is het kiezen van het significantieniveau. Dit houdt in dat we bepalen hoe zeker we ervan willen zijn dat we de correcte conclusie trekken, zonder precisie te verliezen.

Niet elke steekproef zal daadwerkelijk iets kunnen zeggen over de bijbehorende populatie. Als we bijvoorbeeld willen weten of het klopt dat 20% van de auto's in Nederland de kleur blauw heeft, maar in de steekproef kiezen we toevallig alleen auto's met een andere kleur, dan zouden we als conclusie kunnen trekken dat er in Nederland geen blauwe auto's rondrijden. Dit klopt echter niet met de daadwerkelijke populatie. Als je op de weg rijdt zie je namelijk wel degelijk blauwe auto's voorbij komen.

In het bovenstaande geval is de alternatieve hypothese dat 20% van de Auto's in Nederland blauw is maar we trekken de conclusie dat de nulhypothese (het percentage blauwe auto's in Nederland is geen 20%) correct is.

Er bestaat dus de kans dat we de berekeningen en statistiek op de juiste manier uitvoeren, maar alsnog de verkeerde conclusie trekken doordat de steekproef niet representatief is.

Er zijn twee manieren waarop de juiste conclusie wordt getrokken:

- De nulhypothese is correct en we concluderen ook daadwerkelijk vanuit de data dat deze correct is.

- De nulhypothese is niet correct en we concluderen ook daadwerkelijk vanuit de data dat we deze mogen verwerpen.

Omdat we de eigenschap alleen van de steekproef bekijken en niet van de gehele populatie weten we nooit helemaal zeker of we wel de juiste conclusie hebben getrokken (je weet immers niet of de nulhypothese in het echt correct/incorrect is).

Het zogenoemde *significantieniveau* α geeft aan welk risico we willen lopen dat we de nulhypothese foutief verwerpen (d.w.z. de nulhypothese is eigenlijk wel waar maar we concluderen vanuit de data dat deze niet waar is).

Doorgaans wordt er voor het significantieniveau gekozen uit de volgende drie waarden:

- $\alpha = 10\%$
- $\alpha = 5\%$
- $\alpha = 1\%$

Als de waargenomen kans (zie p-waarde hierna) kleiner is of gelijk aan het gekozen significantieniveau a dan verwerpen we de nulhypothese. Is de waargenomen kans groter dan α dan verwerpen we de nulhypothese niet.

Kiezen we bijvoorbeeld een significantieniveau van $\alpha = 5\%$ dan verwerpen we de nulhypothese zodra de waargenomen kans kleiner is dan 5%. Is de waargenomen kans groter dan 5%, dan verwerpen we de nulhypothese niet.

Hoe kleiner de kans is op de nulhypothese des te zekerder we ervan kunnen zijn dat we deze rechtmatig verwerpen. In principe wil je het significantieniveau daarom zo laag mogelijk kiezen. Maar het kiezen van $\alpha = 1\%$ heeft een nadeel. Hoe de lager we het significantieniveau kiezen, hoe meer gegevens we nodig hebben om de nulhypothese te kunnen verwerpen. Dit betekend dat je langer moet meten of meer tests moet uitvoeren. In sommige onderzoeken is dit een groter probleem dan in andere.

Denk maar eens aan de effectiviteit bepalen van een bepaald medicijn. Je moet dan veel patiënten vinden waar het medicijn voor zou moeten helpen. Als je de werking van paracetamol bij hoofdpijn wil onderzoeken is dit misschien geen groot probleem, maar wil je de werking van een medicijn testen die werkt bij een zeer zeldzame ziekte, dan is het gewoon erg lastig zo niet onmogelijk om hele grote groepen te testen.

Ook binnen de natuur- en sterrenkunde verschilt het nogal of je grote datasets kunt verkrijgen. In een experiment in de deeltjesfysica is het bijvoorbeeld relatief eenvoudig om een grote dataset te verkrijgen in een bostingsexperiment zoals op CERN. Maar het onderzoek naar de smaak van neutrino die bij supernova's worden geproduceerd is lastig, er zijn niet heel veel supernova's.

H Het is dus altijd een afweging tussen het zo zeker mogelijk zijn van correctheid van het verwerpen van de nulhypothese, en de uitvoerbaarheid van het onderzoek.

18.3 p-Waarde bepalen

Na het kiezen van het significantieniveau, bepalen (of meten) we de *p-waarde* behorende bij de nulhypothese. De p-waarde is de kans om de geobserveerde meetwaarden te vinden onder de aanname dat de nulhypothese correct is.

Stel we hebben de nulhypothese dat het percentage blauwe auto's in Nederland geen 20% is. We doen een meting waarbij we gedurende een dag het aantal blauwe auto's tellen die op de A6 voorbij komen. De kans dat we een uitkomst kunnen hebben van 25% blauwe auto's, **onder de aanname dat de nulhypothese correct is** (geen 20% blauwe auto's), is de p-waarde. Hoe kleiner de p-waarde die we vinden des te meer grond we hebben om de nulhypothese te verwerpen.

Er zijn verscheidene methodes voor het hypothese toetsen. In deze sectie behandelen we het bepalen van de p-waarde voor een normaal verdeelde dataset, middels de zogenoemde *z-toets*.

Ook voor data met een andere distributie kan de p-waarde bepaald worden via de z-toets voor een normale verdeling. Wel moet er dan een voldoende aantal metingen gedaan zijn zodat de **wet van grote aantallen** toegepast kan worden, en de data benaderd kan worden met een normale verdeling.

Afhankelijk van de manier waarop de nulhypothese en alternatieve hypothese opgesteld zijn, bepalen we de *eenzijdige overschrijdingskans* of de *tweezijdige overschrijdingskans*. Is de nulhypothese opgesteld met de formulering 'is gelijk aan' of 'is ongelijk aan', dan bepalen we de tweezijdige overschrijdingskans. Is de nulhypothese opgesteld met de formulering 'groter/kleiner dan' of 'groter/kleiner of gelijk aan' dan is het noodzakelijk om de eenzijdige overschrijdingskans te bepalen. Dus:

H_0 met	H_α met	type overschrijding
$=$	$\neq$	tweezijdig
$\neq$	$=$	tweezijdig
$\leq$	$>$	eenzijdig
$\geq$	$<$	eenzijdig

Voorbeelden van nulhypothesen waarbij er sprake is van het bepalen van de tweezijdige overschrijdingskans:

- H_0: Het percentage blauwe auto's in Nederland is geen 20%.
- H_0: Het percentage Nederlanders met blauwe ogen is 50%.

Voorbeelden van nulhypothesen waarbij er sprake is van het bepalen van

de eenzijdige overschrijdingskans:

- H_0: De studenten in Amsterdam halen lagere cijfers dan de studenten in Groningen.
- H_0: Het aantal katten in Nederland is kleiner of gelijk aan 20 000
- H_0: Het percentage mensen over de gehele wereld met een hond, is groter of gelijk aan 40%

Zoals eerder vermeld geeft de p-waarde de kans dat waargenomen uitkomst gevonden kan worden onder de aanname dat de nulhypothese correct is. De p-waarde is dus gelijk aan een zeker oppervlak onder de normaalkromme. Deze kun je berekenen met de z-score.

18.4 Conclusie trekken

Tot nu toe hebben we de nulhypothese en de alternatieve hypothese opgesteld. Daarna hebben we bepaald welk significantieniveau we zullen aanhouden. Vervolgens hebben we de z-score en daarmee de p-waarde bepaald. Maar hoe trek je aan de hand hiervan nu een conclusie over de nulhypothese?

Dit bekijken we aan de hand van een paar voorbeelden:

Voorbeeld 1: We onderzoeken de staartlengte van volgroeide lapjeskatten in Nederland, en stellen de volgende hypotheses op:

- H_α: De lengte van de staart van een volgroeide lapjeskat in Nederland is groter dan 10 cm.
- H_0: De lengte van de staart van een volgroeide lapjeskat in Nederland is kleiner of gelijk aan 10 cm.

Bij voorbaat kiezen we als significantieniveau $\alpha = 5\%$.
We meten de staartlengte van 300 lapjeskatten in Nederland (met alle gevolgen van dien voor de onderzoekers), en zetten het resultaat uit in een histogram. Dit resulteert in een normale verdeling met gemiddelde $\mu = 25$ cm en een standaardafwijking 5 cm. De nulhypothese stelde dat de lengte van de staart van een volgroeide lapjeskat kleiner is dan 10 cm. We bepalen dus de p-waarde die hierbij hoort:

$$\begin{aligned} P(X < 10) &= P\left(Z < \frac{10 - 25}{5}\right) \\ &= P(Z < -3) \end{aligned}$$

Als we in de tabel kijken dan hoort er een waarde van 0.00135 bij deze Z-score. Dus:

$$P(X < 10) = P(Z < -3) = 0.00135$$

De p-waarde is dus 0.14%. Op grond van het eerder gekozen significantieniveau van 5% verwerpen we de nulhypothese. In dit geval is het zo dat we de nulhypothese ook

hadden verworpen als we $\alpha = 10\%$ of $\alpha = 1\%$ hadden gekozen.

Is de p-waarde kleiner dan het gekozen significantieniveau dan verwerpen we de nulhypothese. Is de p-waarde groter dan het gekozen significantieniveau dan verwerpen we de nulhypothese niet.

Het is goed om te beseffen dat we **niet** kunnen zeggen dat onze alternatieve hypothese correct is of dat de nulhypothese fout is. De p-waarde geeft namelijk geen bewijs. Wel hebben we met de p-waarde een onderbouwing om de nulhypothese, met inachtname van het gekozen significantieniveau, wel/niet te verwerpen.

Voorbeeld 2: We onderzoeken de gemiddelde lengte van alle vrouwen (> 18 jaar) in Nederland, en stellen de volgende hypotheses op:

- H_α: De gemiddelde lengte van alle vrouwen boven de 18 jaar is hoger dan 180 cm.
- H_0: De gemiddelde lengte van alle vrouwen boven de 18 jaar is lager dan of gelijk aan 180 cm.

Bij voorbaat kiezen we als significantieniveau $\alpha = 5\%$.
We meten de lengte van 500 Nederlandse vrouwen boven de 18 jaar. De resultaten volgen een normale verdeling met gemiddelde $\mu = 165$ cm en een standaardafwijking 10 cm.
De nulhypothese stelde dat de gemiddelde lengte van de Nederlandse vrouwen hoger is dan 180 cm. We bepalen dus de p-waarde die hierbij hoort:

$$\begin{aligned} P(X > 180) &= 1 - P(X < 180) \\ &= 1 - P(Z < \frac{180 - 165}{10}) \\ &= 1 - P(Z < 1.5) \end{aligned}$$

Als we in de tabel kijken dan hoort er een waarde van 0.93319 bij deze Z-score. Dus:

$$P(X > 180) = 1 - P(Z < 1.5) = 0.06681$$

De p-waarde is dus 6.7%. Op grond van het $\alpha = 5\%$ significantieniveau verwerpen we de nulhypothese dus niet.

MODULE III Hoofdstuk 19

Opdrachten module 3

Tijdens laptopcolleges 5 en 6 werken we aan het de opdrachten in module 3. In deze module gaan we werken aan twee opdrachten.

- M3.1 Grote Aantallen III **** (Hfdst. 19.1)
- M3.2 Halfwaardedikte III *** (Hfdst. 19.2)

De sterren geven een indicatie voor hoeveel werk een opdracht is.

19.1 M3.1 Grote Aantallen III ****

In deze opdracht gaan we het eindresultaat van M2.1 'fitten' met de kleinste kwadraten methode.

We hebben gezien dat er verband is tussen de grootte van onze steekproef en de onzekerheid op het bepaalde gemiddelde. Deze volgt de $\sqrt{n}$-wet (Hfdst. 9). We gaan in deze opdracht een lineaire regressie (ofwel een fit) aan de data punten maken met behulp van de kleinste kwadraten methode.

We gaan eerst even terug naar het experiment om te kijken wat we ook alweer aan het bepalen waren. We hebben een ton met kogels en een heel nauwkeurige weegschaal. We kunnen ons verschillende vragen stellen over de massa van de kogels.

- Als we een kogel uit de ton pakken: "Wat is de massa van deze kogel?". De massa van een enkele kogel weten we in dit experiment met bijna oneindige precisie. In ons voorbeeld zelfs bijvoorbeeld: $m_{kogel} = 85.07426079254506 \pm 0.0000000000001$ gram.
- Wat is de massa van een ***typische*** kogel. Wat we hiermee bedoelen is: Als ik een *willekeurige* kogel uit de ton pak, wat is dan de massa? Het antwoord op deze vraag kun je vinden als je het gemiddelde weet van de kogels in de ton en de spreiding (standaardafwijking) van de kogelmassa's. Stel dat het gemiddelde van de populatie 25.0 gram is en de standaardafwijking 2.5 gram dan zeg je in dat geval dat een **typische** massa: $m_{\text{kogel}} = 25.0 \pm 2.5$ gram is. Je moet dan dus wel het gemiddelde

en de spreiding weten, of bepalen. De standaardafwijking is hier dus een maat voor de onzekerheid.

- Om de bovenstaande vraag te kunnen beantwoorden moet je dus weten wat het gemiddelde van de kogel massa's is en wat de spreiding op deze massa's is. De derde vraag die je kunt stellen is dus: Wat is het gemiddelde van de kogel massa's. Om die vraag te beantwoorden kunnen we een steekproef nemen uit de ton. We zien dan al snel een spreiding ontstaan. Bij de eerste kogel kunnen we nog heel weinig zeggen over het populatie gemiddelde en zeker niets over de spreiding. Bij twee kogels heb je al wat meer informatie. Mocht je de standaardafwijking σ_m kennen dan kun je uitrekenen wat de onzekerheid is op het steekproef gemiddelde met de $\sqrt{n}$ wet. Als je steekproef redelijk groot is, dan kun je ook de spreiding s_n hiervoor gebruiken. De onzekerheid waar we hier over hebben gaat dus niet over de onzekerheid op de massa van een enkele kogel, maar over de onzekerheid op de centrale waarde van het kogelmassagemiddelde zelf.

We willen dus weten wat de een **typische** kogel uit de ton weegt, we nemen een steekproef om het gemiddelde van de kogels in de ton te bepalen en we onderzoeken hoe de onzekerheid op de centrale waarde van dit gemiddelde afhangt van de grootte van de steekproef. We focussen dus op de spreiding van de bepaalde gemiddeldes. In M2.1 hebben we een lineair verband gezien tussen de spreiding van de bepaalde gemiddeldes en de grootte van de steekproef. In deze opdracht gaan we deze nu 'fitten' met behulp van de kleinste kwadraten methode..

We gaan eerst datapunten fitten met gelijke fouten. Later kijken we naar meer realistische onzekerheden op de datapunten. Met de volgende instructie kun je de datapunten opvragen:

```
inv_sqrt_n,std_n,std_n_err = ds.GroteAantallenFitSetGenerator()
```

De `inv_sqrt_n` punten zijn de waardes van $1/\sqrt{n}$ waarbij n de grootte is van de steekproef zoals je die in M2.1 hebt gedaan, `std_n` is de onzekerheid s_{g_n} en `std_n_err` zijn de onzekerheden op de waardes van s_{g_n}. In deze opdracht noteren we s_{g_n} als s_n en de onzekerheid op deze standaardafwijking als Δs_n.

Voor deze dataset zijn de waardes van Δs_n dus nog allemaal gelijk, later in deze opdracht zullen we met meer realistische onzekerheden gaan werken. Maar eerst gaan we de fit opzetten.

Naar aanleiding van de $\sqrt{n}$-wet verwachten dat de relatie tussen n en s_n er als volgt uitziet:

$$s_n = \sigma/\sqrt{n}. \tag{19.1}$$

De parameter σ is nu de standaardafwijking van de originele verdeling van de massa van de kogels, dus van de gehele populatie. De variabele $\hat{\sigma}$ is de geschatte waarde van σ die we proberen te vinden met de fit.

- Maak eerst een grafiek waarbij je `std_n` tegen `inv_sqrt_n` uitzet met de foutenvlaggen. Gebruik hier niet de code voor uit M2.1 maar maak gebruik van de dataset die je met het commando dat hier boven beschreven staat verkrijgt. Kijk goed naar de punten en probeer alvast voor jezelf in te schatten welke waarde je verwacht voor σ.
- Vind nu de meest optimale waarde van σ door gebruik te maken van de kleinste-kwadraten (Hfdst. 16) methode.
  1. Schrijf eerst een functie die voor een waarde van `inv_sqrt_n` en een gegeven waarde voor σ een waarde teruggeeft voor de voorspelling van `std_n`. Gebruik hierbij de formule die hierboven gegeven is.
  2. Schrijf een functie die de χ^2 uitrekent volgens de formule die je vindt in het hoofdstuk de kleinste-kwadraten (Hfdst. 16).
  3. Schrijf een loop die over verschillende waardes van σ loopt voor het optimalisatie proces en voor elke waarde van σ de χ^2 uitrekent.
  4. Vind nu voor welke waarde van σ de laagste waarde van χ^2 voorkomt. Dit is je schatting $\hat{\sigma}$.
     Tip: Weet je zeker dat de waarde van σ in het gebied ligt waar je probeert te optimaliseren? Probeer met de grafiek die je eerder maakte af te schatten welke waarde voor σ je verwacht te vinden.

- **M3.1a) Welke waarde voor σ geeft de beste fit? Met andere woorden wat is, na het optimaliseren met de kleinste kwadraten methode, je geschatte $\hat{\sigma}$?**
- **M3.1b) Wat is de waarde voor de geminimaliseerde χ^2? Noteer ook hoeveel vrijheidsgraden er zijn en bereken de χ^2_ν.**
- **M3.1c) Maak een grafiek waarin je de waarde voor χ^2 uitzet tegen σ.**
- **M3.1d) Maak een grafiek met de datapunten, de foutenvlaggen en het fit resultaat.**
  Tip: De gefitte functie kun je het makkelijkste plotten door met behulp van de `inv_sqrt_n` lijst een bijbehorende lijst te maken met behulp van de functie die je in stap 1 hebt gemaakt.

Met de functie:

```
s_true = ds.GroteAantallenStdTrue()
```

Kun je de werkelijke 'true'-waarde van σ terugvragen.

- **M3.1e) Controleer of jouw gefitte waarde van $\hat{\sigma}$ overeen komt met je uitkomst met je uitkomst voor `s_true`. Je verwacht altijd nog wel wat**

verschillen te zien - vooral omdat de onzekerheden op de waardes van `s_n` niet realistisch waren.

We gaan nu de fit uitvoeren met realistische onzekerheden op de datapunten. Deze datapunten genereer je met de volgende functie:

```
inv_sqrt_n,std_n,std_n_err = ds.GroteAantallenStdGenerator()
```

- **M3.1f) Vind nu de meest optimale waarde van $\hat{\sigma}$ door gebruik te maken van de realistische foutenvlaggen. Bij welke χ^2 ligt deze optimale waarde?**
- **M3.1g) Maak nu een grafiek met de datapunten, de foutenvlaggen en het fit resultaat voor de dataset met reële foutenvlaggen.**
- **M3.1h) Vergelijk nu de gevonden $\hat{\sigma}$ met de 'true' waarde van σ. Komt deze nu meer of minder overeen in vergelijking met je eerste fit?**
- **M3.1i) Bereken nu de gereduceerde χ^2, dat wil zeggen corrigeer de gevonden χ^2 voor het aantal vrijheidsgraden van de fit. Is deze beter of slechter dan de gevonden waarde in opgave b. Geef hiervoor een verklaring.**

19.2 M3.2 Halfwaardedikte III ***

In opgave M2.3 hebben we gezien dat de meetmethode die we gebruikten om de halfwaardedikte te bepalen niet optimaal was. Er was zeker sprake van een onzuivere meting doordat we stelselmatig een te hoge waarde van d_{half} terugkregen.

In deze opgave zullen we zien dat de onzuiverheid te maken heeft met de methode waarop we de halfwaardedikte hebben bepaald. Het heeft niets te maken met de opstelling van de meting of met de verzamelde datapunten. Het is de analyse techniek die zorgt voor de onzuiverheid.

In deze opdracht gaan we een fit gebruiken om de waarde van d_{half} te achterhalen. In opdracht M3.1 hebben we onze eigen lineaire regressie methode geprogrammeerd met behulp van de kleinste kwadraten methode. In deze opdracht gebruiken we een fit pakket `lmfit`. Dit programma rekent de χ^2 uit en minimaliseert deze voor ons. Dat scheelt op zich een hoop werk, maar je zult in deze opdracht zien dat het toch ook weer niet helemaal vanzelf gaat.

Om dit fit pakket te kunnen gebruiken moet het volgende import statement gebruiken:

```
from lmfit import models
```

Maak nu een dataset aan met de standaard waardes zoals je dat in M2.3 ook hebt gedaan:

```
counts, diktes, dtrue = ds.DataSetHalfwaardeDikteVariatie()
```

We hebben eerst een functie nodig die de datapunten beschrijft en een set met startwaardes voor de parameters. We kijken eerst nog even naar de formule die in M1.1 is gegeven:

$$I(d; N_0, d_{half}) = I_0 \times \left(\frac{1}{2}\right)^{d/d_{half}} \tag{19.2}$$

We zien dat de functie uiteindelijk afhangt van twee parameters: I_0 en d_{half}. De waardes I_0 en $I(d)$ zijn natuurlijk direct gerelateerd aan de gemeten waardes voor N_0 en $N(d)$. Met de vergelijking $I = N/(\Delta T)$. In principe is N_0 en dus I_0 gemeten, maar hier zit een (bekende) onzekerheid op en om die reden wil je hem 'vrijlaten' in de fit.

De functie `functie` die we straks gebruiken voor de fit ziet er in het algemeen als volgt uit:

```
def functie(x,par1,par2) :
    y = een formule
    return y
```

De parameters die hier worden meegegeven zijn de parameters die worden geschat (of geoptimaliseerd) in de fit. Voor deze twee waardes zullen we straks ook de startwaardes moeten meegeven.

- Schrijf nu eerst de code voor de functie `functie(d, N0, dhalf)` die de relatie tussen dikte d en de counts aangeeft. Controleer of die goed werkt.
- Voor de fit hebben we ook een lijst met gewichten nodig. Deze gewichten zijn gelijk zijn gelijk aan de reciproke waardes van de fouten op de counts. Het is de deler in de χ^2 vergelijking. Noem deze gewichten `N_inv_err` en maak hiervoor een lijst aan. Als de onzekerheid op N, ΔN is, dan is het gewicht dus $1/\Delta N$.

Als we onze functie en de lijst met gewichten hebben gedefinieerd dan kunnen we de fit uitvoeren.

```
ons_model = models.Model(functie)
result= ons_model.fit(counts, d=diktes, weights =N_inv_err,
    N0=startwaarde,dhalf=startwaarde)
```

We definiëren eerst `ons_model` en vervolgens fitten we deze. Je moet een aantal opties meegeven:

```
result   : deze vangt het fit resultaat op
counts   : de lijst met counts
d=diktes : d is de eerste parameter van functie, diktes is de lijst met
    diktes
weights = N_inv_err : hier geef je de lijst met gewichten mee
N0= startwaarde : hier moet je de startwaarde voor de fit meegeven op N0
dhalf = startwaarde : hier moet je de startwaarde voor dhalf meegeven
```

Je ziet dat je nog zelf twee startwaardes mee moet geven voordat de fit kan werken. Je kan eventueel eerst even de data plotten om zo de startwaardes voor N_0 en d_{half}. Met het volgende commando kun je de fitresultaten uitprinten:

```
print(result.fit_report())
```

- **M3.2a) Voer de fit uit en bekijk het fitresultaat. Als je tevreden bent met de fit kopieer dan je resultaat op het inlevertemplate. Het kan zijn dat je de startwaardes van de parameters nog iets moet aanpassen als de fit niet convergeert.**

De gefitte curve kunnen we ook weergeven in een grafiek. Maak zoals gebruikelijk een grafiek met foutenvlaggen. Het fitresultaat kun je dan als volgt toevoegen:

```
plt.plot(diktes, result.init_fit, 'k--', label='initial fit')
plt.plot(diktes, result.best_fit, 'r-', label='best fit')
plt.legend(loc='best')
```

TIP Pas op de je de juiste waardes kiest voor de foutenvlaggen, deze zijn dus niet hetzelfde als de gewichten die je gebruikt hebt in de fit.

- **M3.2b) Maak een grafiek met de datapunten, foutenvlaggen en het gefitte resultaat. Maak de grafiek netjes af.**
- **M3.2c) Bekijk de gereduceerde χ^2_ν. Ziet deze waarde er goed uit? Beredeneer je antwoord. Wat is het aantal vrijheidsgraden in de fit?**
- **M3.2d) Wat is de geschatte waarde $\hat{d}_{half}$? Vergelijk deze met de 'true' waarde 'dtrue'.**
- **M3.2e) De correlatiecoëfficiënt r wordt ook uitgeprint. Hoe groot is deze en wat zegt dat?** Bedenk goed voor welke parameters deze correlatiecoëfficiënt berekend is.

Definieer nu een polynoom met de volgende code:

```
def poly(d,N0,a,b) :
    y = N0 + a*d + b*d*d
    return y
```

Fit deze functie aan de datapunten, zorg dat de startwaardes zo worden ingesteld dat de fit convergeert. Meestal kies je voor de startwaardes eerst 1 en als dat niet convergeert, probeer dan wat andere waardes.

- **M3.2f) Maak een grafiek met de datapunten, foutenvlaggen en het gefitte resultaat. Maak de grafiek netjes af.**
- **M3.2g) Presenteer de fitresultaten van de poly fit op het inlevertemplate.**
- **M3.2h) Vergelijk nu de twee fits met elkaar. Bekijk de uitkomsten van de gefitte exponentiele functie met de gefitte polynoom. Welke functie beschrijft de data het beste? Op basis van welke variabelen trek je deze conclusie? Beargumenteer je antwoorden.**

MODULE III APPENDIX A

De χ^2 -toets tabel

In deze kansen kan je voor verschillende overschrijdingskansen aan de bovenzijde (0.995, 0.99, ..., 0.005) de χ^2 waardes aflezen per hoeveelheid vrijheidsgraden ν.

Bijvoorbeeld als je 5 vrijheidsgraden hebt in je fit, dan ligt slechts 10% van de fits hoger dan een χ^2 van 15.086.

De onderkansen kun je vinden door het complement te nemen. Slechts in 5% van de gevallen vind je een waarde van $\chi^2 \leq 0.412$ bij $\nu = 5$.

ν	0.995	0.990	0.975	0.950	0.900	0.100	0.050	0.025	0.010	0.005
1	—	—	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	4.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	5.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.034	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932	41.401

ν	0.995	0.990	0.975	0.950	0.900	0.100	0.050	0.025	0.010	0.005
22	8.643	9.542	10.982	12.338	14.041	30.813	33.924	36.781	40.289	42.796
23	9.260	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638	44.181
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.559
25	10.520	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314	46.928
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.808	12.879	14.573	16.151	18.114	36.741	40.113	43.195	46.963	49.645
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.993
29	13.121	14.256	16.047	17.708	19.768	39.087	42.557	45.722	49.588	52.336
30	13.787	14.953	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
40	20.707	22.164	24.433	26.509	29.051	51.805	55.758	59.342	63.691	66.766
50	27.991	29.707	32.357	34.764	37.689	63.167	67.505	71.420	76.154	79.490
60	35.534	37.485	40.482	43.188	46.459	74.397	79.082	83.298	88.379	91.952
70	43.275	45.442	48.758	51.739	55.329	85.527	90.531	95.023	100.425	104.215
80	51.172	53.540	57.153	60.391	64.278	96.578	101.879	106.629	112.329	116.321
90	59.196	61.754	65.647	69.126	73.291	107.565	113.145	118.136	124.116	128.299
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.169

MODULE IV

Deze week sluiten we het vak af. We gaan kijken naar hoe we gebruik kunnen maken van de geminimaliseerde χ^2 waardes van een fit bij het testen van een hypothese. Dit doen we met behulp van de Wald test die in hoofdstuk Hypothese toetsen II (Hfdst. 20) bespreken.

MODULE IV Hoofdstuk 20

Hypothese toetsen II

We hebben in module 3 een eerste stap gemaakt met hypothese toetsen. We hebben gezien dat er vier belangrijke stappen zijn. Eerst stellen we de hypothese op die we willen toetsen en ook de nulhypothese. Vervolgens is het belangrijk om een statistiek te vinden die gevoelig is voor de stelling. Met andere woorden, een statistiek waarmee we de hypothese kunnen toetsen. We kiezen van tevoren een significantieniveau (p-waarde) waarbij we H_0 of H_α kunnen verwerpen. We hebben ook gezien wat de z-score betekend.

In dit hoofdstuk leggen we nu een bijzondere vorm van hypothese toetsen uit waarbij we gebruik maken van de kleinste kwadraten methode en de daarbij berekende χ^2.

20.1 De Wald test

De Wald test is een bijzondere test die kan worden gebruikt om met behulp van de kleinste kwadraten methode een hypothese te toetsen.

Het idee is om aan een set meetwaardes twee functies te fitten. De eerste functie, f_0, beschrijft de dataset onder de hypothese H_0, de tweede functie, f_α, beschrijft de dataset onder de alternatieve hypothese H_α. Het verschil in de geminimaliseerde χ^2 voor beide functies wordt gedefinieerd als

$$\Delta\chi^2 = \chi_0^2 - \chi_\alpha^2. \tag{20.1}$$

Dit chi-kwadraat verschil ($\Delta\chi^2$) kan direct worden gebruikt om een p-waarde te berekenen met behulp van een opzoektabel.

Er zijn hierbij strikte voorwaardes voor het opstellen van de twee functies. De functies mogen slechts in 1 parameter verschillen, verder moeten ze geheel identiek zijn. De H_0 hypothese wordt hierbij beschreven met het *minste* aantal vrije parameters. Alle parameters die H_0 kents, kent H_α ook.

Voorbeeld Als de nulhypothese wordt beschreven door een functie $f_0(x; a, b)$ dan wordt de alternatieve hypothese beschreven door een functie $f_\alpha(x; a, b, c)$ waarbij de parameters a en b identiek zijn en ook de relatie tussen x en deze twee parameters gelijk is.

Alleen als aan de bovengenoemde voorwaarde wordt voldaan dan wordt de $\Delta\chi^2$ beschreven door een χ^2 functie met vrijheidsgraad $n = 1$. En zoals we in module 3 hebben beschreven is de χ^2 zelf een kansdichtheidsverdeling. We kunnen in dat geval de $\Delta\chi^2$ direct omrekenen naar een waarschijnlijkheid en deze is gelijk aan de p-waarde.

Voorbeeld Wald test Stel dat we een chemisch element willen traceren en gebruik maken van spectroscopie. Als het chemische element X aanwezig is dan verwachten we een verhoogde intensiteit te zien bij de emissielijn van het specifieke element. We verwachten ook een achtergrondspectrum te zien. Dat wil zeggen we meten over alle golflengtes normaal gesproken een bepaalde intensiteit, ook zonder dat het chemische element aanwezig is. We kunnen nu de twee functies opstellen. Stel dat de achtergrond een lineaire functie volgt

$$I_0(\lambda; a, b) = a + b \cdot \lambda.$$

Waarbij λ de golflengte is.
De emissielijn van X, verwachten we rond 930 nm en de resolutie van de spectroscoop is 1 nm. De intensiteit van de emissielijn wordt dan beschreven door:

$$I_\alpha(\lambda; J, \lambda_0, \sigma) = J \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\lambda-\lambda_0}{\sigma}\right)^2}$$

We zien dat in principe er geen vrije parameters zijn in deze fit, behalve een schaalfactor J die de hoeveelheid intensiteit van het signaal schaalt.

De functie f_0 wordt in dit geval gelijk gesteld aan de functie die de achtergrond (of nulhypothese) beschrijft: $f_0 = I_0$. De vrije parameters in deze fit zijn a en b.

De functie f_α die de alternatieve hypothese beschrijft is nu gelijk aan de achtergrond, plus het signaal: $f_\alpha = I_0 + I_\alpha$. De vrije parameters in deze fit zijn a, b en J. We voldoen dus aan het criterium van de Wald methode.

Het verschil in de geoptimaliseerde χ^2's voor de nul- en de alternatieve hypothese is gelijk aan $\Delta\chi^2 = \chi^2_0 - \chi^2_\alpha$.

We spreken af dat we de nulhypthese mogen verwerpen als de p-waarde kleiner is dan $1 \cdot 10^{-6}$.

We gaan even naar de data kijken. We hebben het spectrum waargenomen dat hier in figuur 20.1 wordt getoond.

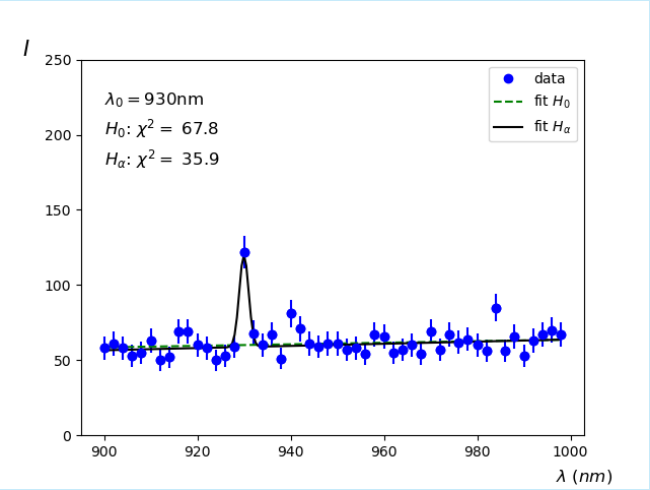


Figure 20.1: Het waargenomen spectrum met de gefitte lijn.

In de grafiek zien we een duidelijk piekje rond 930 nm, precies waar we het signaal van het chemische element X door H_α voorspeld is. De fit resultaten van beide hypotheses zijn in het plaatje weergegeven. Met het verschil in χ^2 kunnen we nu een p-waarde uitrekenen. In dit geval is die gelijk aan $1.6 \cdot 10^{-8}$. Het is dus uitermate waarschijnlijk dat we het chemische element X hebben aangetoond in de spectraal analyse.

De berekende p-waarde wordt vaak weer omgerekend in een z-score. De enige reden waarom dit gedaan wordt is omdat de waardes van de z-score over het algemeen wat makkelijker liggen. Het is zeg maar een handigere manier om een kans uit te drukken. In het voorbeeld hierboven komt de kans van $1.6 \cdot 10^{-8}$ overeen met een z-score van 5.5. Ga maar na, het laatste spreekt een stuk makkelijker uit.

In het voorbeeld hierboven hebben we een nulhypothese (waarbij alleen een achtergrond-spectrum aanwezig is) vergeleken met alternatieve hypothese waarbij een element X bestaat en we de emissielijn meten in het spectrum. We hebben een kans gevonden (de p-waarde) van $1.6 \cdot 10^{-8}$ dat de geobserveerde dataset past bij de nulhypothese. Dit is een uitermate kleine kans en omdat deze kleiner is dan het vooraf afgesproken significantieniveau mogen we de nulhypothese verwerpen.

20.2 p-Waarde scan

In het voorbeeld hierboven is er een duidelijk stelling over de golflengte van de emissielijn van het element X, namelijk $\lambda_0 = 930$ nm. Stel nu dat dat niet zo is, dan zouden we een extra vrije parameter hebben in de functie die H_α beschrijft. In dat geval kunnen we de Wald methode niet toepassen. Wat we in dat geval wèl kunnen doen is een zogeheten p-waarde scan uitvoeren. We fixeren dan telkens de waarde van de golflengte van de emissielijn en berekenen voor elk van deze golflengtes de p-waarde. Als er een emissielijn aanwezig is die sterk genoeg is zullen we op die locatie een dip zien in de p-waarde.

We moeten ook bij deze toets van te voren bepalen bij welke p-waarde we de nulhypothese verwerpen.

Voorbeeld p-waarde scan We gaan terug naar ons experiment met de spectraalfit. In dit experiment is er een deeltje Y dat wel wellicht kunnen waarnemen. Echter, in dit geval weten is er geen voorspelde waarde van de golflengte λ_0, ook kennen we de verwachte intensiteit niet. We kunnen de Wald test hierdoor niet zomaar uitvoeren. Immers moeten we precies één vrije parameter extra fitten in de H_α hypothese ten opzichte van de H_0 hypothese.
De oplossing vinden we door 1 variabele te fixeren, deze houden we constant. We kunnen dan de p-waarde scannen als functie van de gefixeerde parameter. In het voorbeeld hier scannen we over de golflengte λ_0.
We laten de fit nu voor vier waardes van λ_0 in figuur 20.2 zien. Voor elke waarde van λ_0 fitten we nu de twee functies alsof we weten dat de spectraallijn van element Y zich precies daar bevindt.

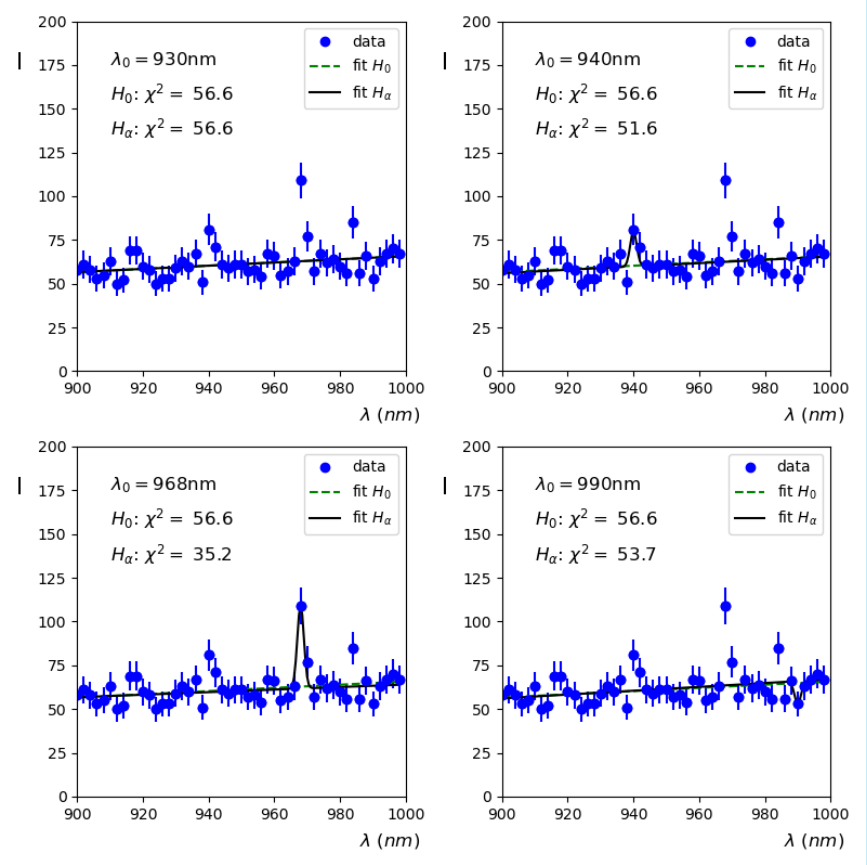


Figure 20.2: Het waargenomen spectrum met de gefitte lijn voor $\lambda = 932$ nm.

We zien voor de fit met waarde $\lambda_0 = 930$ nm dat de $\Delta\chi^2$ gelijk is aan 0. Maar als we goed kijken in het plaatje zien we ook geen enkel piekje bij $\lambda = 930$ nm. Bij de waardes van $\lambda = 940$ en 980 nm zien we wel een klein piekje. Maar vooral bij de waarde van $\lambda = 968$ nm is een echte piek te vinden.
Als we alle p-waardes van de scan (maar dan over alle waardes van λ_0) nu grafisch weergeven dan krijgen we het volgende <!–FIG, in figuur 20.3 resultaat.

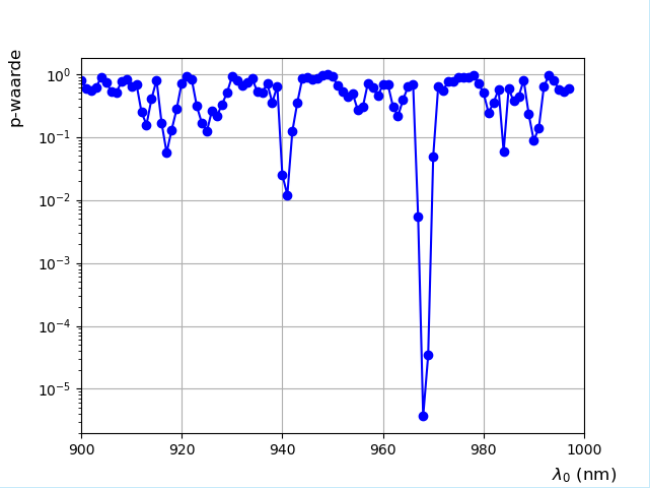


Figure 20.3: De p-waarde scan van de emissiedata.

Je ziet nu dat er op een aantal plekken in het spectrum een kleine afwijking van de H_0 hypothese te zien is. Deze komen allemaal overeen met golflengtes waar op het oog een piekje zichtbaar lijkt. Op slechts 1 locatie is er een heel duidelijke afwijking zichtbaar. Precies bij 968 nm. De $\Delta\chi^2$ is op dat punt 21.4 en dit kunnen we omrekenen naar een p-waarde van $3.7 \cdot 10^{-6}$.

Eigenlijk hadden we van tevoren ook bij dit experiment van te voren een significantieniveau moeten afspreken waarbij we de aanwezigheid van het chemische element kunnen aantonen. Zodra de gemeten p-waarde onder deze afgesproken significantie zakt in de p-waarde scan kunnen we claimen het element te hebben aangetoond. Als we hem weer bij een waarde van $1 \cdot 10^{-6}$ hadden afgesproken hadden we ook in de p-waarde scan het element Y kunnen claimen.

De Wald test is een krachtige methode om hypotheses te toetsen. We gaan hem in opdracht M4.1 toepassen.

MODULE IV Hoofdstuk 21

Opdrachten module 4

Tijdens het laatste laptopcollege werken we aan het de opdracht in module 4. In deze module gaan we werken aan één opdracht.

- M4.1 Een nieuw deeltje *** (Hfdst. 21.1)

De sterren geven een indicatie voor hoeveel werk een opdracht is.

21.1 M4.1 Een Nieuw Deeltje ***

We gaan op zoek naar een nieuw elementair deeltje X. Dit deeltje is gepostuleerd door een groep Natuurkundigen die daarmee het bestaan van de Donkere Materie in het heelal denken te kunnen verklaren.

We kunnen het deeltje misschien maken met botsingen bij de Large Hadron Collider op het CERN onderzoekscentrum. Als deze deeltjes X bestaan dan zouden we dit moeten kunnen zien in de gereconstrueerde massaverdeling van de restproducten van de X-deeltjes in een heel precies geselecteerde dataset.

We hebben een beschrijving van hoe de massaverdeling eruit zou zien *zonder* het bestaan van het deeltje, dit noemen we de achtergrond. En we weten hoe een massaverdeling eruit zou zien als er *wel* X-deeltjes zouden bestaan. Alleen hebben de theoreten geen enkel idee van de exacte massa van het X deeltje. We gaan in een massagebied de p-waarde scannen met behulp van de Wald (Hfdst. **??**) test. We berekenen in de massa scan voor elke waarde van m_X de p-waarde en kijken of we een significante afwijking vinden.

In het vakgebied van de deeltjesfysica hanteren we de volgende normen voor de berekende p-waarde in de Wald test.

z-score	p-waarde	statement
$\geq 3\ \sigma$	≤ 0.003	observatie van afwijking
$\geq 5\ \sigma$	$\leq 2.87 \times 10^{-7}$	ontdekking

- **M4.1a) Wat is H_0 en wat is de H_α hypothese in dit onderzoek? Postuleer de stellingen.**

Haal de dataset op met het volgende statement

```
m,events,events_err = ds.DeeltjesDataset()
```

De drie `list`'s die worden teruggegeven zijn:

```
m - de berekende massa van de restproducten uitgedrukt in proton massa's
    (pm)
events - het aantal botsingen in de geselecteerde data
events_err - de onzekerheid op het aantal events
```

Het aantal *events* (aantal evenementen) is het geobserveerde aantal botsingen die we in onze dataset hebben voor de specifieke massabin m.
De achtergrond data wordt beschreven met de volgende functie:

$$f(m; N_0, c) = N_0 \cdot \left(\frac{1}{2}\right)^{m/c} \tag{21.1}$$

De massaverdeling van deeltje X ziet er zo uit:

$$g(m; m_0, \sigma, N_X) = N_X \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{m-m_0}{\sigma})^2} \tag{21.2}$$

We weten niet wat de waarde is van m_0 en ook niet hoeveel X-deeltjes we zouden kunnen hebben in onze dataset (N_X). Wel kennen we de resolutie van onze meting ($\sigma = 5$ pm). Hiermee bedoelen we dat we op de massapiek van het X deeltje een spreiding verwachten van ($\sigma = 5$ pm). De eenheid van m drukken we uit in het aantal proton massa's omdat het anders onhandig wordt met de eenheid (1 proton massa is gelijk aan 1.6726×10^{-27} kg). Voor de Wald test mogen we maar 1 vrije parameter verschil hebben tussen de achtergrond (H_0) en de deeltjeshypothese (H_α), maar we hebben dus twee onbekende parameters die we moeten schatten. We gebruiken hier de methode van de p-waarde scan en scannen de massa m, zo hebben we maar 1 vrije parameter in de fit en dat is de hoeveelheid deeltjes, N_x.

We gaan weer fitten met het pakket **lmfit**.

- Schrijf eerst een functie die je `achtergrond_functie(x,N0,c)` noemt. Het is misschien logischer om in plaats de `x` variabele `m` te gebruiken, maar later in het programma maken we gebruik van een standaard functie van `models` en hiervoor is het nodig om het nu alvast `x` te noemen.

Nadat je de functie hebt geschreven die de achtergrond beschrijft maken we het model

voor de fit en die noemen we `achtergrond_model`.

```
achtergrond_model = models.Model(achtergrond_functie)
```

- **M4.1b) Zet nu eerst een fit op waarbij je het achtergrond model fit. Maak een grafiek waarbij je de datapunten, de onzekerheden op de datapunten en de gefitte curve laat zien.**

```
plt.plot(m, result.init_fit, 'k--', label='initial fit')
```

Tip 1. Zoals je ziet lijkt de functie erg op de functie die gefit moest worden in opgave M3.2, je kan een deel van je code dus hergebruiken.
Tip 2. Het is voor deze fit erg belangrijk om de startwaardes goed mee te geven. Bekijk eerst eens de data goed en probeer af te schatten welke startwaardes voor `N0` en `c` je moet meegeven. De functie met startwaardes kun je op de volgende manier visualiseren. (Hiervoor moet je wel eerst het fit statement hebben uitgevoerd maar ook als de fit niet goed werkt zie je nog steeds of de startwaardes goed zijn ingeschat.)
Tip 3 Voor de fit moet je de gewichten van de datapunten mee geven. In module 3 heb je al gezien dat deze niet precies hetzelfde zijn als de onzekerheden (`events_err`) maar dat je ze eerst even moet omrekenen met de formule $w = 1/\sigma$.
Tip 4 Kijk nog even goed naar de fit in opgave M3.2 en op welke plek je welke informatie moet meegeven.

- **M4.1c) Hoeveel vrijheidsgraden, ν heeft deze fit?**

- **M4.1d) Wat is de χ^2 en de χ^2_ν voor deze fit?**

De massaverdeling van deeltje X wordt beschreven met een normaalverdeling. Als het deeltje X bestaat, dan ligt deze normaalverdeling als het ware op de achtergrond. We zullen dus een model moeten programmeren die de som is van de normaal verdeling die het 'signaal' beschrijft en de achtergrond functie. We beginnen eerst met het opzetten van een functie die alleen de signaal component beschrijft.

- De functie die de massaverdeling van het deeltje X beschrijft is een normaalverdeling. We kunnen hiervoor een eigen functie programmeren maar we kunnen ook gebruik maken van de standaard functies die het `lmfit` pakket bevat. Deze kunnen we aanroepen met:

```
normaal_model = models.GaussianModel()
```

Als je een standaardfunctie gebruikt is het altijd even goed om uit te zoeken hoe die precies werkt. Je kan bijvoorbeeld even op de website van lmfit kijken en zoeken naar `GaussianModel()`.

We bekijken in elk geval even hoe de vrije parameters en de variabele heten in dit model. Dit kun je doen met het volgende statement.

```
print('De parameters: ',normaal_model.param_names,' de variabele: ',
    normaal_model.independent_vars)
```

We zien nu dat er drie variabelen zijn `amplitude`, `center` en `sigma`. Vergelijk de functie op de website met de normale verdeling voor het deeltje X. We zien nu dat de variabele `x` de variabele m is.

Nu maken we eerst het model voor de H_α hypothese.

- We gaan nu model voor het signaal maken dat dus bestaat uit de achtergrond component en de normaal component:

```
signaal_model = normaal_model + achtergrond_model
```

Met de informatie die we hierboven gegeven hebben weten we dat we een van deze parameters (namelijk de standaardafwijking van de normaalfunctie) moeten *fixeren*. We bedoelen hiermee dat deze niet een vrije parameter in de fit mag zijn, hij moet constant worden gehouden in de optimalisatie van de χ^2.

We kunnen een parameter fixeren met het volgende statement:
signaal_model.set_param_hint("par_name", vary=False)
waarbij `par_name` dus de naam van de variabele is waarvoor we dat willen doen. Dit statement zegt dat de variabele `par_name` niet wordt geoptimaliseerd in de fit procedure, de variabele wordt gefixeerd op de waarde die je startwaarde meegeeft als je de fit uitvoert.

We zijn nu klaar om de zogeheten p-waarde scan uitvoeren. We kiezen steeds een waarde van m_0 en laten alleen de volgende parameters vrij in de fit: N_0, c en N_x. De andere parameter σ wordt vastgezet op 5 en m_0 wordt steeds op een andere gekozen waarde gefixeerd in **het gehele gebied die door de dataset wordt beschreven met stapjes**

van 1 proton massa.

- Fit nu voor elke integer waarde van m_0 in het massagebied van m de functie en bereken de χ^2 van de fit met het signaal model. Controleer of alle parameters die moeten worden gefixeerd in de fit, dat ook daadwerkelijk zijn. Kijk hiervoor naar het fit resultaat.

  Tip 1: Zorg dat je de juiste startwaardes meegeeft.
  Tip 2: Je kunt de χ^2 opvragen van het fit resultaat met het statement:

  ```
  result.chisqr
  ```

- **M4.1e) Hoeveel vrijheidsgraden heeft de signaal fit? Schrijf de formule helemaal uit.**

Voor elke waarde van m_0 kunnen we nu het verschil in χ^2 tussen de fit met het achtergrond model en het de χ^2 en de fit van het signaal model bij die waarde van m_0. Dit verschil noteren we als:

$$\Delta\chi^2 = \chi^2_a - \chi^2_s \tag{21.3}$$

Waarbij we $\Delta\chi^2$ kunnen we omrekenen naar een p-waarde. Lees hierover meer in het hoofdstuk Hypothese toetsen II (Hfdst. 20).

- Gebruik de volgende functie uit het `scipy.stats` pakket om de p-waarde te berekenen:

  ```
  from scipy import stats
  p_value = stats.chi2.sf(Delta_chisquare, 1)
  ```

- **M4.1f) Bereken voor *elke waarde* van m_0 nu de p-waarde en representeer deze in een grafiek waarbij je de p-waarde uitzet tegen m_0.**
  Tip: Gebruik hiervoor de volgende plot opties om de grafiek duidelijker te maken:

  ```
  plt.yscale('log')
  plt.grid(True)
  ```

- **M4.1g) Bij welke waarde van m_0 vind je de beste p-waarde in jouw massa gebied?**

- **M4.1h) Maak een grafiek met de dataset en de gefitte modellen (achtergrond en signaal) voor deze waarde van $\hat{m}_0$.**

- **M4.1i) Bereken voor $\hat{m}_0$ de p-waarde en de z-score.** De z-score kun je met het volgende statement uitrekenen:

```
z_waarde = -stats.norm.ppf(p_waarde)
```

- **M4.1j) Denk je dat je de achtergrond hypothese kunt verwerpen. Zo ja, redeneer waarom. Zo nee redeneer waarom niet.** Kijk even naar de afspraken die hierover zijn gemaakt in de deeltjefysica.