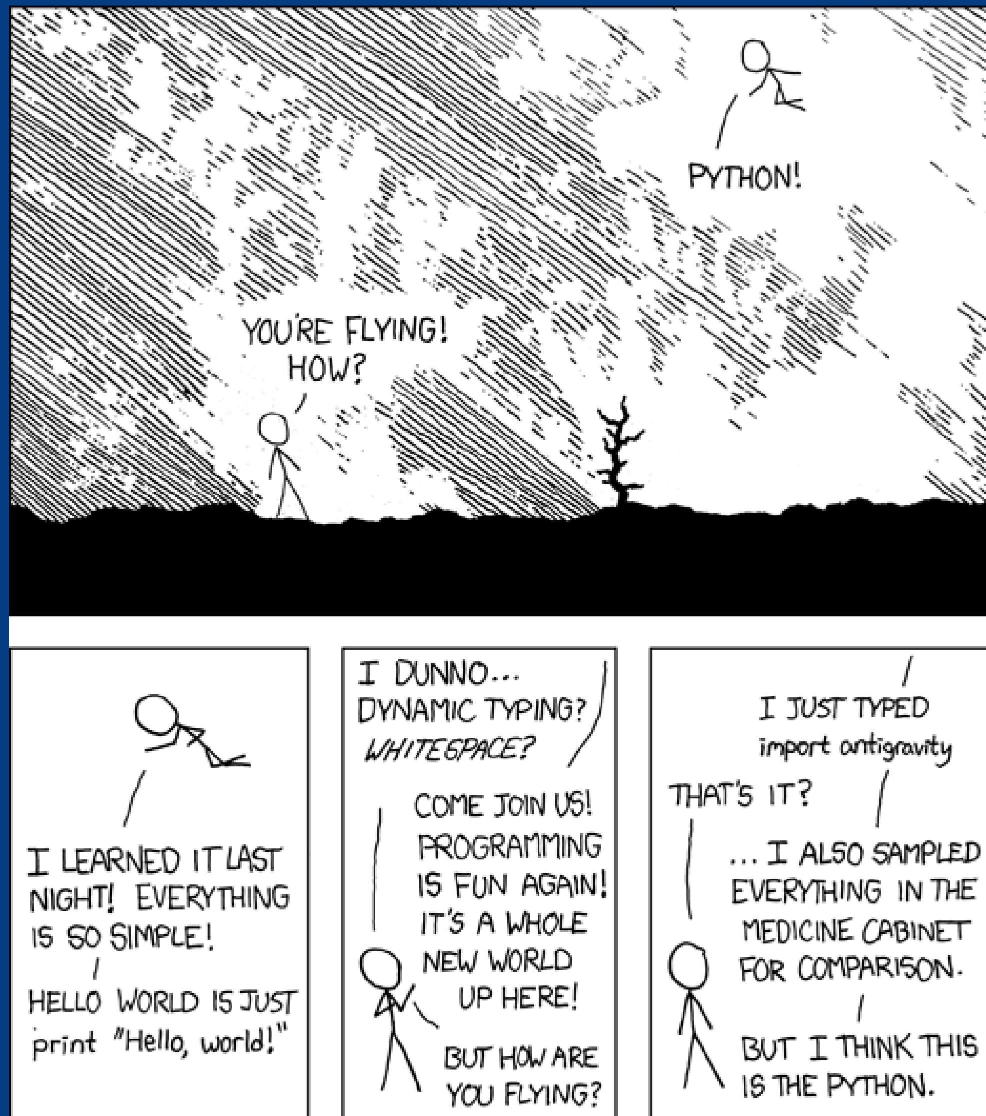
Fitten met PHYTON



Inhoud

Wat gaan we doen?

- PYTHON:
  - Wat?
  - Waarom?
  - Waar?
  - 'opwarmoefeningen'
    - Variabelen declareren
    - Workspace
    - Wiskundige operaties
    - Plotten
    - Flow control
    - Functies
    - Data inladen
- Fitten met LS-methode
  - Theorie
  - Klassikaal voorbeeld
  - Taak

PHYTHON
Wat?

- Probleem → Computergesteunde oplossing

vb. ‘fitten’ → Implementeren van een fitalgoritme op basis van LS-methode

- PHYTON

general purpose/multi-paradigm programmeertaal

i.h.b. gebruiken we de volgende pakketten (libraries):

NumPy – (numerieke) wiskunde, arrays & matrices

SciPy - \`science and engineering (v.b. Integratie, optimalisatie)

Matplotlib – plotten, gericht op wetenschap

....

De keuze voor programma (of algoritme) is niet bindend, **zolang het fitten maar correct gebeurt**
Wij vragen wel om de eigengeschreven code voor the Python-taak en practicum 4 te gebruiken

Een van de doelen van deze zitting is immers beter begrijpen wat (een voorbeeld van) een fitalgoritme doet, a.d.h.v. de kleinste-kwadratenmethode zoals gezien in de theorieles

PHYTHON

Waar?

'hetzelfde' als in de ICTS PC-klas is te vinden op (freeware):

https://www.anaconda.com/download/

'Anaconda, the most popular Python Data Science Platform'

- Keuze tussen Python 2 en 3
- Anaconda navigator:
  Overzicht van **programmeeromgevingen** waarin je Python op een gebruiksvriendelijke manier kunt gebruiken. Wij zullen zoals vorig jaar **Jupyter Notebook** gebruiken voor deze les.

‘Opwarmoefeningen’

Vind pythonles.ipynb op Toledo, en open dit in Jupyter Notebooks

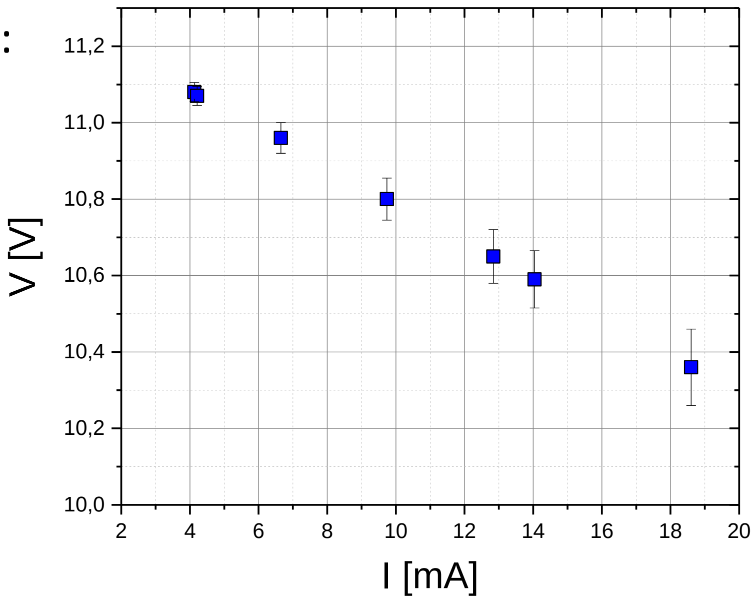
Theorie:

Waarin zijn we geïnteresseerd?

Meetresultaten: x_i , y_i met een onzekerheid σ_i

We willen deze set data beschrijven: $y_i = f\left(x_i \middle| \vec{\theta}\right)$

Bijvoorbeeld:



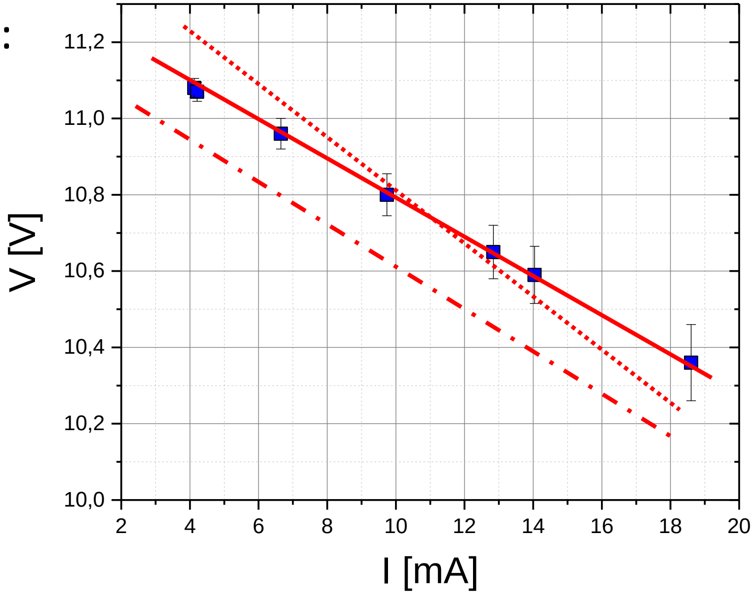
Theorie:

Waarin zijn we geïnteresseerd?

Meetresultaten: x_i , y_i met een onzekerheid σ_i

We willen deze set data beschrijven: $y_i = f\left(x_i \middle| \vec{\theta}\right)$

Bijvoorbeeld:



$f(x_i \mid \vec{\theta}) = f(x_i \mid [a, b]) = a x_i + b$

Hoe goed beschrijft $\vec{\theta}$ de data?
Welke $\vec{\theta}$ beschrijft de data het beste?

Beste beschrijving: *Likelihood waarde*

Theorie: Likelihood waarde

Veronderstel onafhankelijke datapunten

(x_i, y_i, σ_i) hangt niet af van (x_j, y_j, σ_j)

Likelihood:

$$L\left(\vec{\theta}\middle|x,y,\sigma\right) = \prod_i p\left(\vec{\theta}\middle|x_i,y_i,\sigma_i\right)$$

Aangezien datapunten onafhankelijk:
Likelihood is het product van de individuele kansen *p* om *(x_i, y_i, σ_i)* te ‘verkrijgen’ bij parameterset $\vec{\theta}$

Grotere Likelihood, betere beschrijving van de dataset:

zoek $\vec{\theta}$ die de likelihood maximaliseert!

Theorie: Loglikelihood

Voor veelal praktische redenen is het ‘eenvoudiger’ om het negatief logaritme van de likelihood te minimaliseren, in plaats van de likelihood the maximaliseren:

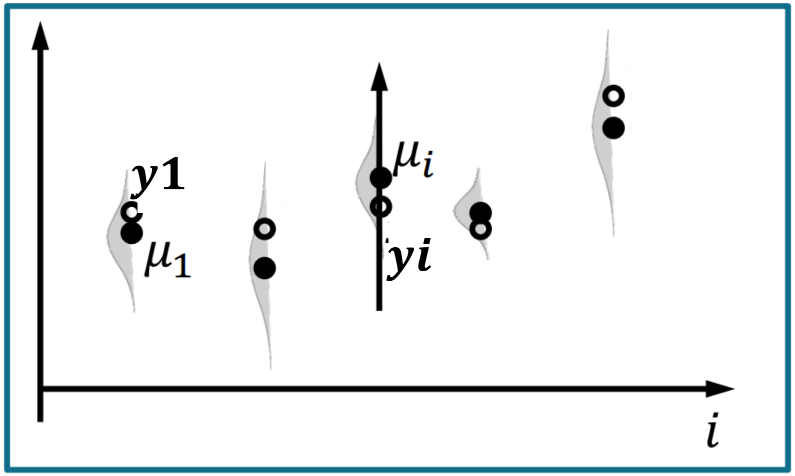
(product wordt som, maximum wordt minimum)

$$\mathcal{L}(\vec{\theta} \mid x, y, \sigma) = -\ln L(\vec{\theta} \mid x, y, \sigma) = -\sum_i p(\vec{\theta} \mid x_i, y_i, \sigma_i)$$

“ Ok – maar wat is nu $\boldsymbol{p(\vec{\theta} \mid x_i, y_i, \sigma_i)}$, wat heeft dat met ons model $\boldsymbol{f(x_i \mid \vec{\theta})}$ te maken, hoe minimaliseren we dat?”

Theorie: Least squares LS

Vaak gebruikt: **Neem aan** dat de datapunten y_i uit verschillende normale verdelingen komen met gemiddelde $\mu_i(x_i \mid \vec{\theta}) = f(x_i \mid \vec{\theta})$ en standaarddeviatie σ_i. Dan is $p\left(\vec{\theta} \mid x_i, y_i, \sigma_i\right)$ dus:



$$p\left(\vec{\theta} \mid x_i, y_i, \sigma_i\right) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[\frac{-(y_i - f(x_i \mid \vec{\theta}))^2}{2{\sigma_i}^2}\right]$$

"<u>Wat is hier σ_i?</u>"

De beste benadering voor de standaarddeviatie van de veronderstelde normale verdeling:

Vb. Meetfout op y_i na foutenpropagatie, $\Delta_{syst}/\sqrt{12}$ als y_i een direct gemeten grootheid is, populatie standaarddeviatie s_i (spreiding) als y_i het resultaat is van een herhaalde meting en de spreiding groter is dan de systematische fout,....

Theorie: Least squares LS $\longrightarrow$ χ^2

Dus, ons minimalisatieprobleem wordt:

$$\mathcal{L}(\vec{\theta} \mid x, y, \sigma) = -\sum_i p(\vec{\theta} \mid x_i, y_i, \sigma_i) = \cdots$$

Zie les voor verdere aannamens, uitwerking, subtiliteiten,… … …

Minimaliseren van:

$$LS(\vec{\theta}) \equiv \sum_i \frac{(y_i - f(x_i \mid \vec{\theta}))^2}{\sigma_i^2} \equiv \chi^2(\vec{\theta})$$

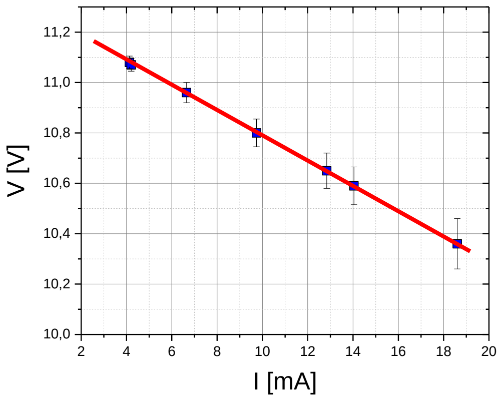
→ Levert beste fit $\vec{\theta}^*$

→ ‘programmeren’

! intuitief

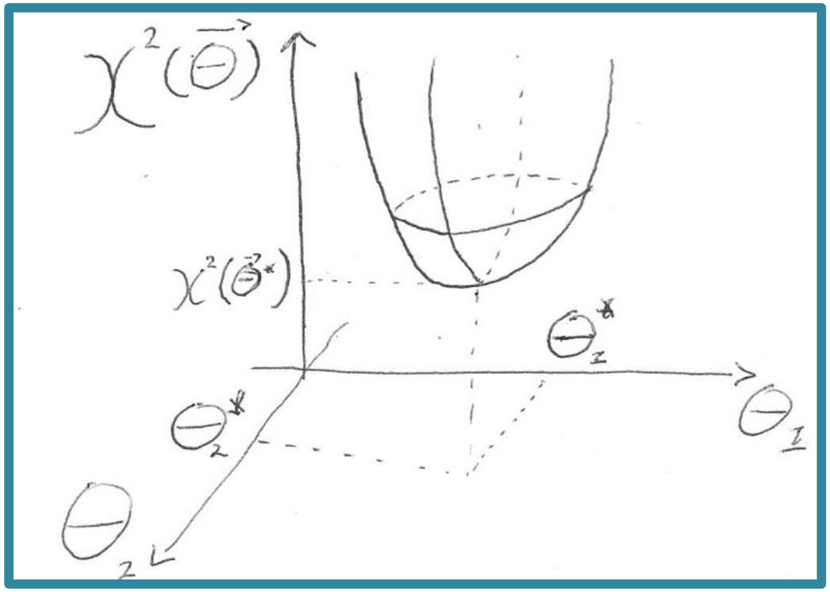
Least squares Fit - Ingrediënten:

- ✔ Minimaliseren $LS(\vec{\theta}) \equiv \chi^2(\vec{\theta})$ levert beste fit $\vec{\theta}^*$ = $(\theta_1^*, \ldots, \theta_n^*)$
- ✔ Plot oplossing (data + $f(x_i\,|\vec{\theta}^*)$)



- ✘ Fout σ_{θ_i}op θ_i^* ?
- ✘ Kwaliteit van de fit (goodness-of-fit) ?

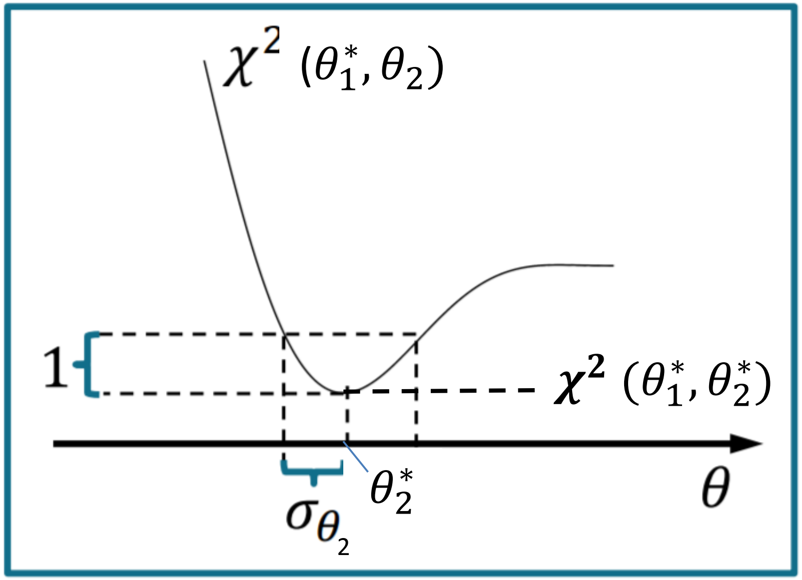
- *Fout op fitparameters σ_{θ_i}*



bv.

neem
$\theta_1 = \theta_1^*$
vast

Varieer θ_2



In de les gezien / bewezen (Taylorexpansie rond optimum $\vec{\theta}^*$):

$$\chi^2(\theta_2^* \pm \sigma_{\theta_2}) = \chi^2(\theta_2^*) + 1$$

→ Oplossen om σ_{θ_2} te vinden

→ Niet per se symmetrisch rond θ_2^*, $\theta_2^{*\,+\,bovengrens}_{\;\;-\,ondergrens}$

Goodness-of-fit

De aanname is dat de termen $\frac{(y_i - f(x_i\,|\,\vec{\theta}))^2}{\sigma_i^2}$ normaal verdeeld zijn → hun som is χ^2 verdeeld

Fit is 'goed' als de teststatistiek $LS(\vec{\theta}) \equiv \sum_i \frac{(y_i - f(x_i\,|\,\vec{\theta}))^2}{\sigma_i^2} \equiv \chi^2(\vec{\theta})$ dicht bij zijn verwachte waarde ligt, die afhangt van het aantal vrijheidsgraden ν = N – p (aantal datapunten – aantal fitparameters)

Voor ν groot, verwachtingswaarde ≈ ν, dus:

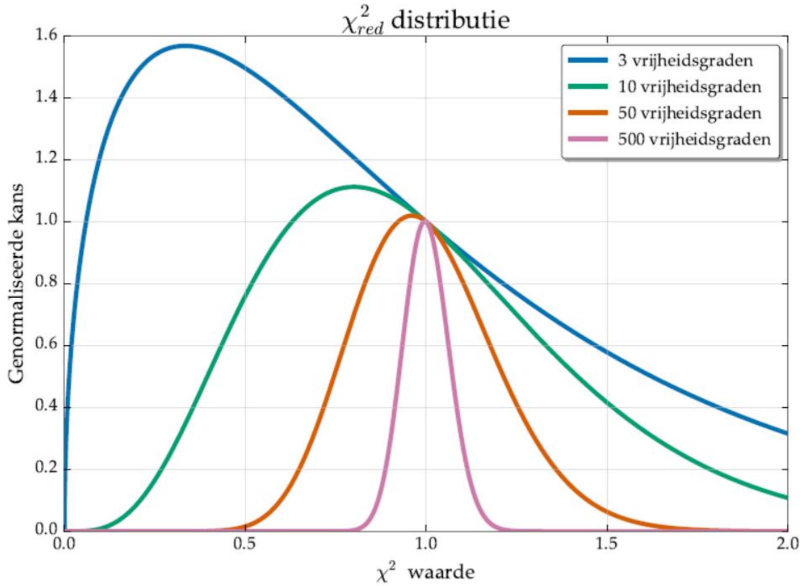
$$\chi^2 \approx N - p \quad \text{of} \quad \chi^2/(N-p) \approx 1$$

Maar:

? Wat voor "kleine ν"

? Wanneer ligt χ^2 'ver' van de verwachte waarde (is 0.6 'dicht' bij 1 of niet?)

→ **χ^2 toets**



Goodness-of-fit → χ^2 **toets**

- Teststatistiek: $LS(\vec{\theta}) \equiv \sum_i \frac{(y_i - f(x_i\,|\,\vec{\theta}))^2}{\sigma_i^2} \equiv \chi^2(\vec{\theta})$
- **H$_0$** hypothese in dit geval: de Teststatistiek voldoet aan de χ_v^2 -distributie.
- Aantal vrijheidsgraden ν = N – p
- Kies significantieniveau (α = 1% of 5% typisch)
- Vergelijk teststatistiek met de kritische waarde van de χ_v^2 -distributie en het geselecteerde betrouwbaarheidsniveau (meestal eenzijdig rechts):

p(χ_v^2 >$LS(\vec{\theta})$) →

> α. **H$_0$** niet verwerpen.

< α. **H$_0$** verwerpen op significantieniveau α. Wat is er dan mis?

p-waarde ' te klein' -> $LS(\vec{\theta})$ bijvoorbeeld 'te groot'?

-> Model beschrijft data niet goed?

-> onze aannamen voor $p(\vec{\theta}\mid x_i, y_i, \sigma_i)$ zijn verkeerd (misschien niet normaal verdeeld?)

-> de fouten σ_i misschien onderschat?

<u>Zie volgende slide:</u>
in realiteit kan de p-waarde van deze test ook ' te groot' (i.e. $LS(\vec{\theta})$ 'te klein')

KU LEUVEN

Even tussendoor: χ^2 toets / kwantielen met PYTHON:

Herinner je uit de les (3):

- p-waarde (11 vrijheidsgraden, eenzijdig rechts):
  $p(Q_{11}^2 > 14{,}9) = 18{,}6\% > 5\%$ dus ok (hypothese niet verwerpen)

erimentele basistechnieken in de natuurkunde – 2016/2017 07/11/2016 – Les 3

Opzoeken in tabel…… ->
met Statistiek pakket van SciPy:

$P(\chi^2_{\nu=11} > 14.9) = 1 - p(\chi^2_{\nu=11} < 14.9) = \ldots$

```
In [9]:  import scipy.stats as stat

         pwaarde = 1- stat.chi2.cdf(14.9 , 11)
         print(pwaarde*100, '%')

18.7120909047 %
```

In realiteit kan de χ^2 waarde ook ‘te klein’ zijn: p-waarde ‘te groot’ :

$P(\chi^2_{\nu=11} > 3{,}2) = 1 - p(\chi^2_{\nu=11} < 3{,}2) = \ldots$

```
In [14]:  import scipy.stats as stat

          pwaarde = 1- stat.chi2.cdf(3.2, 11)
          print(pwaarde*100, '%')

98.7809830974 %
```

Concreet (neem b.v. het 5% significantieniveau):

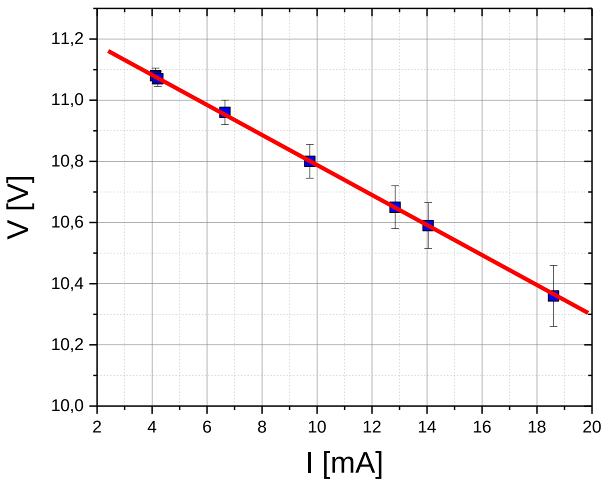
De p-waarde van deze test moet tussen 5% en 95% liggen om de hypothese niet te verwerpen

Least squares Fit - Ingrediënten:

- Minimaliseren $LS(\vec{\theta}) \equiv \chi^2(\vec{\theta})$ levert beste fit $\vec{\theta}^* = (\theta_1^*, \ldots, \theta_n^*)$

  !- θ_i^* hebben eenheden,...

- Plot oplossing (data + $f(x_i\,|\vec{\theta}^*)$)



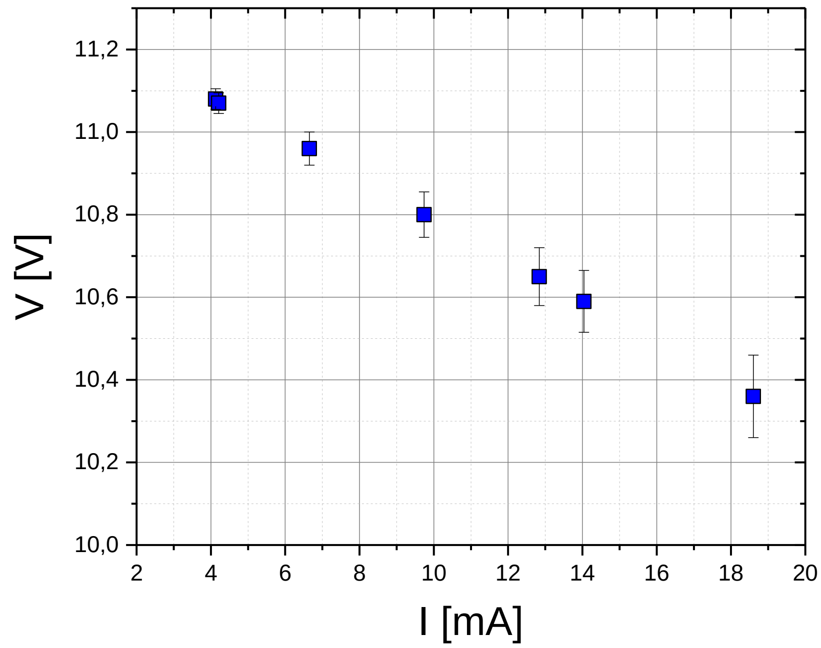
- Fout σ_{θ_i}op θ_i^* → $\chi^2(\theta_2^* \pm \sigma_{\theta_2}) = \chi^2(\theta_2^*) + 1$ oplossen levert $\theta_2^*\,{}^{+\,bovengrens}_{-\,ondergrens}$

- Kwaliteit van de fit (goodness-of-fit) → χ^2 toets

  Maar natuurlijk geeft de plot ook veel informatie over de kwaliteit van de fit!

- *Klassikaal voorbeeld: Gebruik meting.txt*

I(mA)	ΔI (mA)	V_A(V)	ΔV_A(V)
4,13	0,05	11,08	0,12
4,21	0,05	11,07	0,12
6,65	0,08	10,96	0,12
9,74	0,11	10,80	0,12
12,84	0,14	10,65	0,12
14,04	0,15	10,59	0,12
18,6	0,20	10,36	0,11



$$R = \frac{V_0 - V_A}{I} \longrightarrow V_A = -RI - V_0$$

Nog wat laatste opmerkingen….

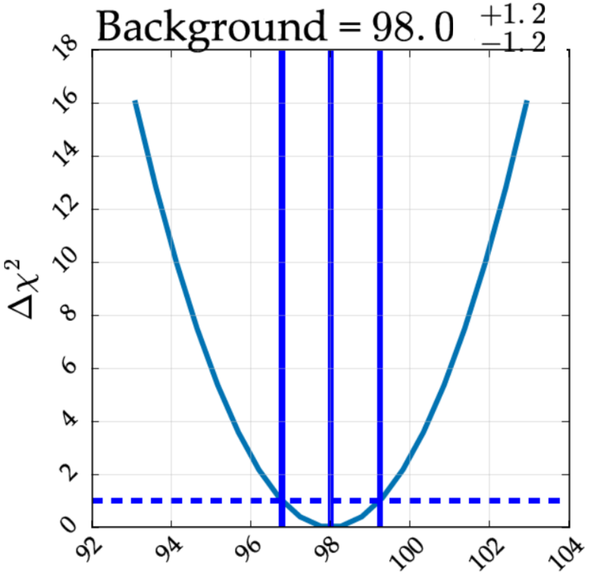
1) **Alle** elementen behandelen bij een fit… (zie 2 slides terug!)

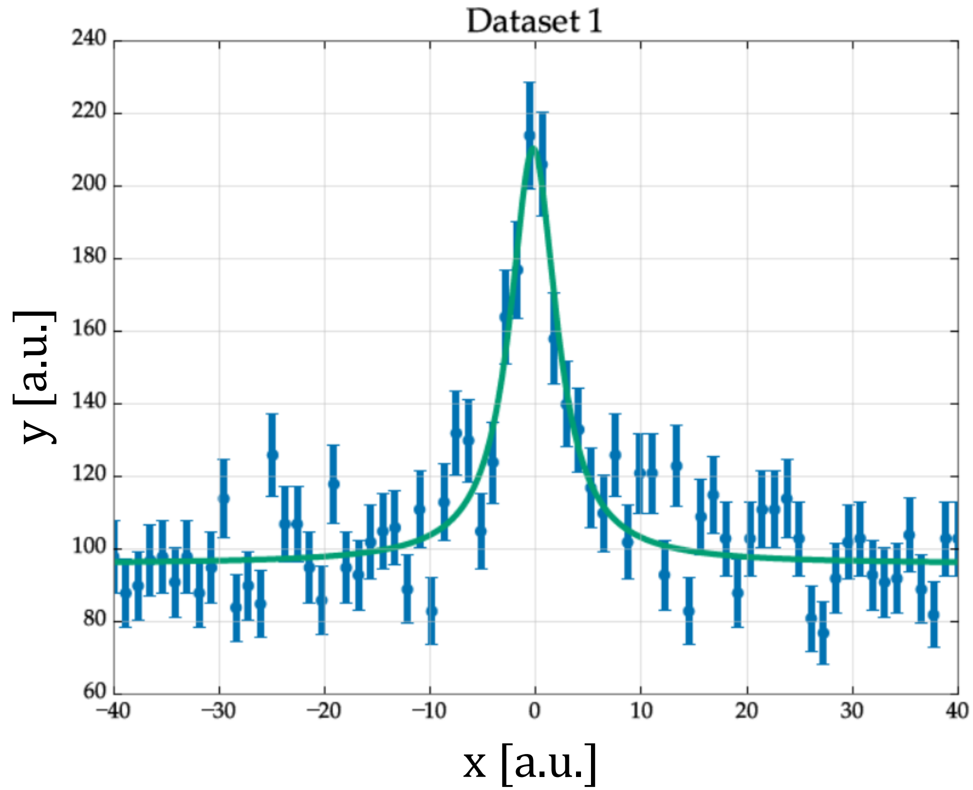
2) Maak **duidelijk leesbare plots** (labels van assen, symbolen)
   - Alles op de plot moet benoemd zijn (data, fit,….)

3) De hier gepresenteerde methode is zeker niet 'uniek'
   - LS methode houd bijvoorbeeld geen rekening met eventuele correlatie tussen de fitparameters,
   - LS methode houd geen rekening met fout op x_i
   - Voor dit vak: gebruik LS-methode

4) Ook de implementatie is zeker niet uniek:
   - PYTHON heeft ook een 'fit(…)'commando, en bij Graphical Analysis of 'Excel' gebruik(te) je ook een 'black box' fit methode.
   - Voor de taak een practicum 4, gebruik je eigen geschreven implementatie van de LS methode. Voor het vervolg, ben je vrij om te gebruiken wat je wilt – zolang je maar voldoet aan stap 1) en 2) hier!

- *Taak*





Goodness of fit = ?

Taak

Concreet willen we, van iedereen individueel (dus niet per groep) een verslag (.pdf) van deze opdracht zien die de volgende elementen bevat:

(i) Eén figuur met een plot van de data (met error bars) én de plot van de beste fit (i.e. vergelijking (1) met de best fittende parameters).

(ii) De gevonden beste fitwaarden met hun fouten.

(iii) de reduced χ^2-waarde en een kleine bespreking daarvan (kwaliteit van de fit / het fitmodel).

(iv) Plots van de χ^2 waarde rond het minimum voor elke fitparameter (μ, γ, etc.) afzonderlijk, waarbij je laat zien voor welke waarden van deze parameter de χ^2-waarde exact 1 groter is dan het minimum. Dit is dus de illustratie van de methode om de fouten op de fitwaarden af te schatten.

(v) De gebruikte matlabcode moet ook in de .pdf .

N.B.: Merk op dat stappen (i)-(iii) de elementen zijn die je (zeker voor dit vak) *altijd* moet behandelen bij het maken van het fit. (iv)-(v) moeten niet altijd expliciet gemaakt en getoond worden (mogen eventueel in een bijlage).