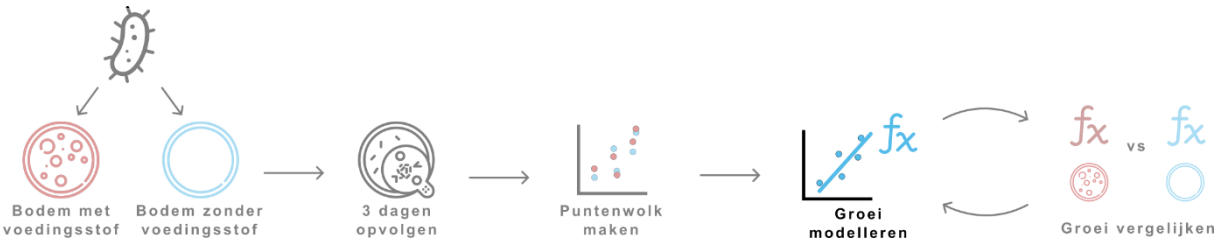
Deelprobleem 4

Welk wiskundig model hoort bij de opgemeten data en hoe kan je hieruit een maat halen om bacteriëngroei te vergelijken?



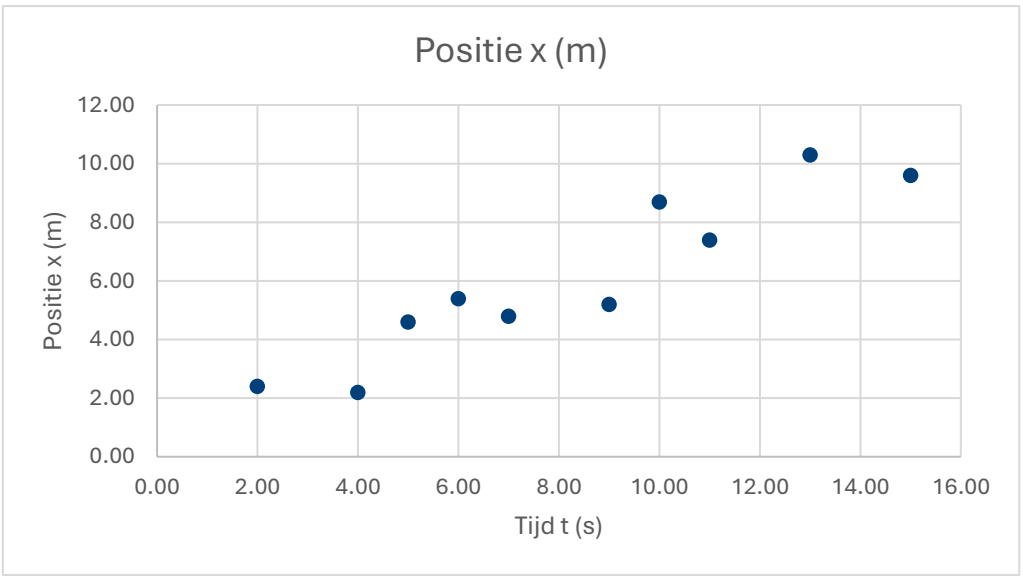
SPREIDINGSDIAGRAMMEN EN TRENDLIJNEN

Instap: variabiliteit in wetenschappelijke experimenten

Wanneer je een wetenschappelijk experiment uitvoert, meet je vaak meerdere variabelen en ga je op zoek naar het verband ertussen. Dit kan dan leiden tot een formule voor dat verband. Bijvoorbeeld, men weet dat bij een *eenparig rechtlijnige beweging* de positie $x(t)$ recht evenredig afhangt van de tijd, volgens de formule

$$x = x_0 + v \cdot t.$$

Rune wil nagaan of die wet wel klopt, en stelt een experiment op. Ze heeft een speciale opstelling waarmee ze een voorwerp aan een constante snelheid in een rechte lijn kan laten bewegen. Voor dit experiment kiest ze $v = 0{,}65$ m/s en $x_0 = 0{,}80$ m. Ze zet de experimentele data uit in een puntenwolk en bekomt de grafiek in Figuur 1.



Figuur 1: puntenwolk van opgemeten data experiment rollend balletje.

Hoe verwacht je dat de grafiek eruit ziet?

Waarom wijkt puntenwolk in Figuur 1 hiervan af?

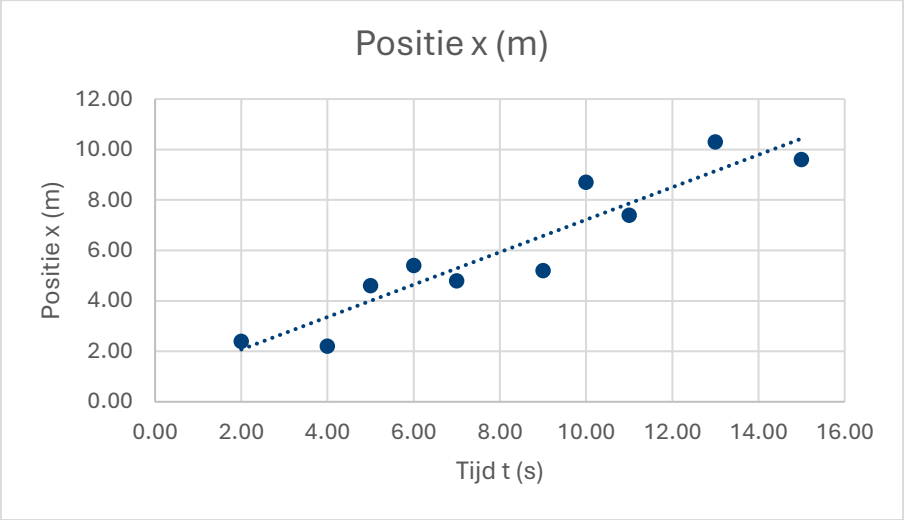
De variabiliteit die Rune observeert in het experiment is heel normaal in de wetenschap. Gelukkig biedt de statistiek wiskundige methodes om voor deze variabiliteit te compenseren.

Een trendlijn bepalen: lineaire regressie

Lineaire regressie

Lineaire regressie is een methode om een rechte te vinden die een gegeven puntenwolk het best beschrijft.

Toepassing op voorbeeld eenparig rechtlijnige beweging



Met behulp van ICT bepaalt Rune de trendlijn, zo bekomt ze

$$x = 0{,}79 + 0{,}64 \cdot t.$$

Wat is de fysische betekenis van de getallen $0{,}79$ en $0{,}64$?

Zijn dit de waarden die je verwacht? Waarom? Welke effecten spelen mee?

Hoe zou je de 'best passende rechte' door een puntenwolk zou definiëren?

Hoe goed beschrijft een rechte de data?

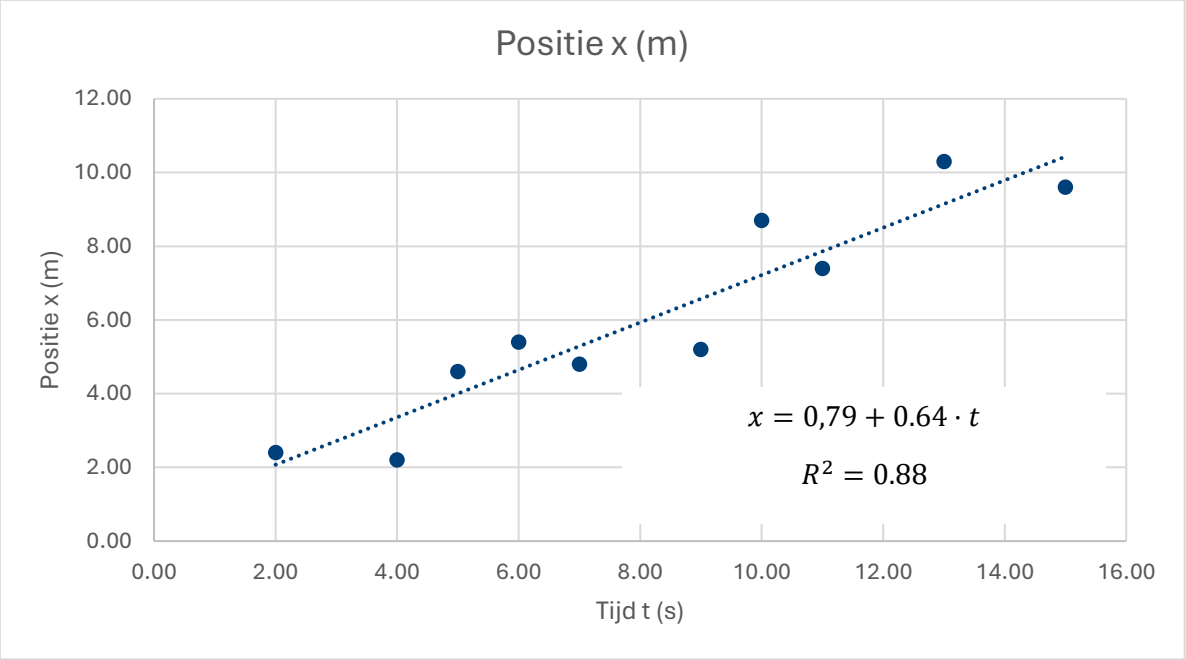
De kleinste kwadratenmethode laat toe om voor elke puntenwolk een 'best passende' rechte te vinden. De rechte is dan een 'model' voor je data. Om te weten hoe gepast een rechte is als model voor je data, kan je de determinatiecoëfficiënt gebruiken.

Determinatiecoëfficiënt

De determinatiecoëfficiënt R^2 is een getal tussen 0 en 1 die aangeeft hoe goed een puntenwolk beschreven kan worden door een lineair model (een rechte).

(*) De determinatiecoëfficiënt is het kwadraat van de Pearson correlatiecoëfficiënt.

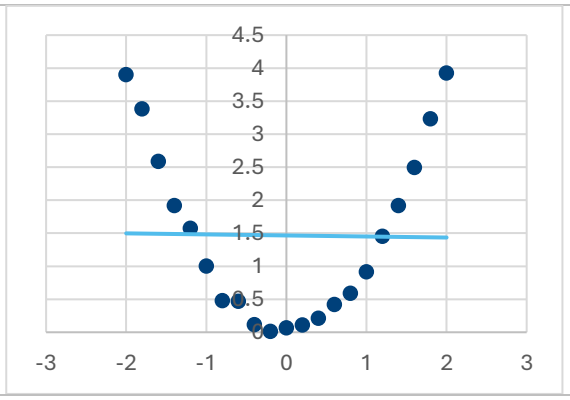
Toepassing op voorbeeld eenparig rechtlijnige beweging

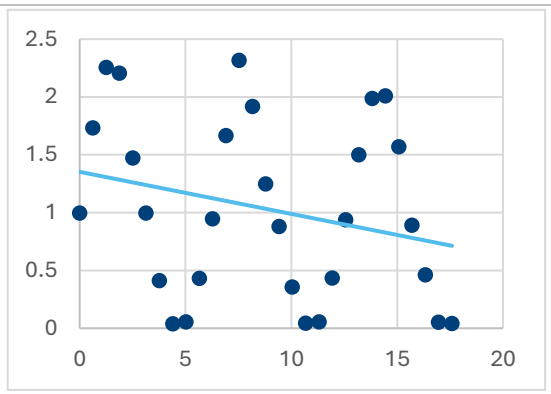


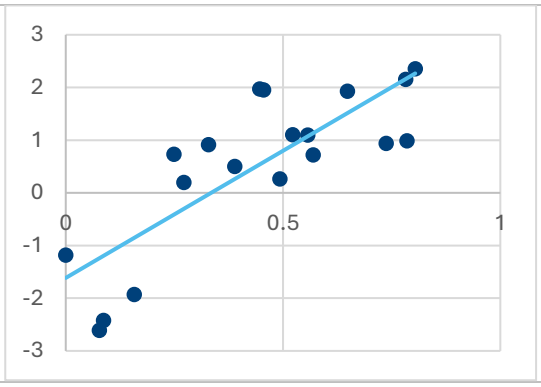
De determinatiecoëfficiënt ligt dicht bij 1, dus een rechte is een goed model voor de experimentele data.

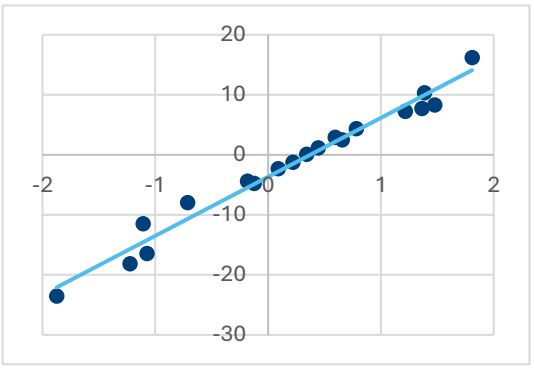
Oefening: Geef voor volgende puntenwolken en bijhorende lineaire trendlijn een schatting voor de waarde van de determinatiecoëfficiënt. Kies tussen $R^2 \approx 0$; $0{,}3 < R^2 < 0{,}7$ en $R^2 \approx 1$.

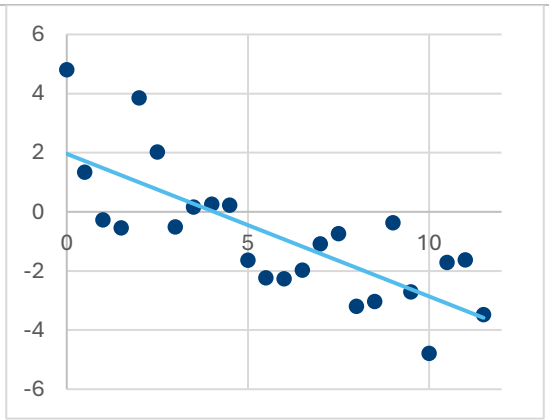
Opmerking: de grenzen $0{,}3$ en $0{,}7$ zijn willekeurig gekozen en betekenen vooral dat de determinatiecoëfficiënt niet bijna nul of bijna één is.

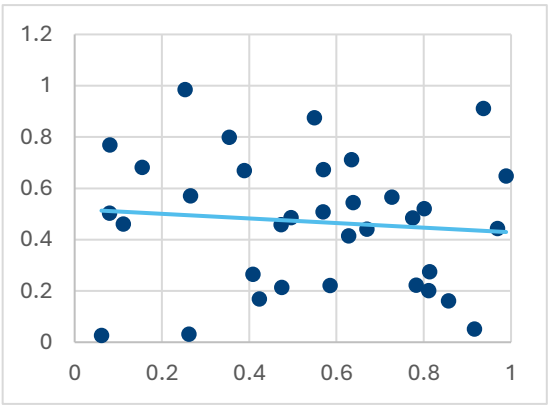












Een lineaire trendlijn en determinatiecoëfficiënt bepalen in Python

Je kan in Python eenvoudig een lineaire trendlijn bepalen. Volg deze stappen, waarbij je de code die je in deelprobleem 3 opstelde aanpast.

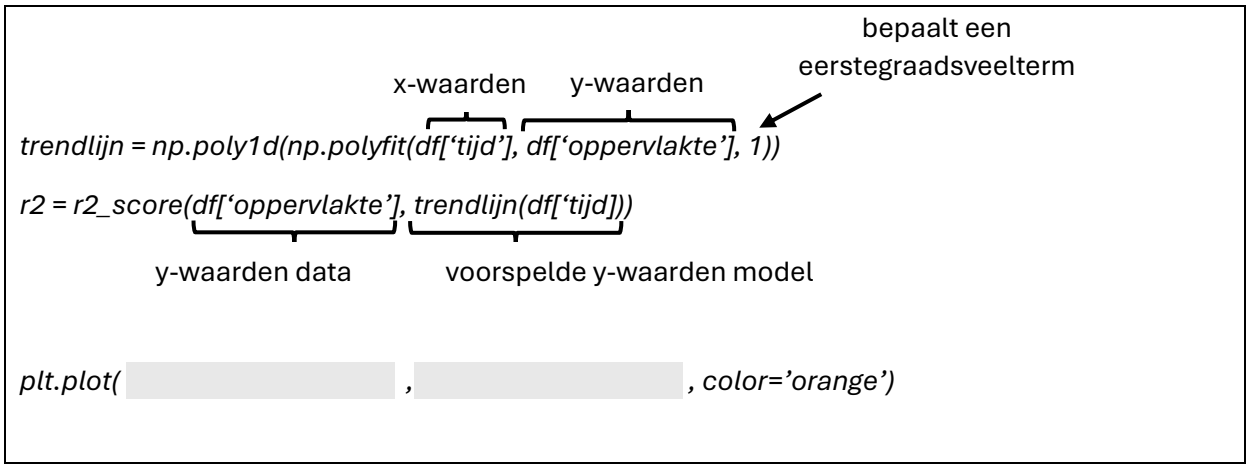
Stap 1

Voeg volgende softwarebibliotheken toe in de eerste lijnen van het bestand:

```
import numpy as np
from sklearn.metrics import r2_score
```

Stap 2

Voeg tussen de lijn met *df.plot(...)* en de lijn met *plt.show()* volgende code in:



Deze code bepaalt een functie *trendlijn* die de best passende rechte door de data

$$(x, y) = (df['tijd'], df['oppervlakte'])$$

weergeeft. De laatste lijn geeft de trendlijn weer in het oranje op de puntenwolk.

Merk op dat de kolommen 'tijd' en 'oppervlakte' in jouw code anders kunnen heten, afhankelijk van de namen die je ze gegeven hebt eerder in de code en in het csv-bestand.

Opdracht

De code hierboven is niet volledig. De eerste twee inputwaarden van *plt.plot* zijn immers niet gedefinieerd. Vervang deze door de gepaste commando's zodat de correcte trendlijn verschijnt op de figuur.

Weet je niet hoe hieraan te beginnen? Raadpleeg de tekst in volgende paragraaf. In principe kan het ook lukken met de informatie die hierboven staat!

(*) De code uit stap 2 ontleed

import numpy as np

In de eerste lijn extra code wordt de softwarebibliotheek **NumPy** ingeladen, waarbij we verwijzen naar deze bibliotheek met het verkorte *np*. Deze bevat functies om wetenschappelijke berekeningen te doen in Python. Zo kan je eenvoudig rekenen met lijsten met waarden – in NumPy heet een lijst overigens 'array'. De standaard ingebouwde Python *list()* laat immers niet toe om twee lijsten op te tellen. Ga dit maar eens na!

from sklearn.metrics import r2_score
Vervolgens laden we de methode *r2_score* in uit de softwarebibliotheek **scikit-learn**. Deze bibliotheek bevat een hele resem functies om data te analyseren. De specifieke functie *r2_score* zullen we gebruiken om de determinatiecoëfficiënt te bepalen, soms ook de R^2-score genoemd.

trendlijn = np.poly1d(np.polyfit(df['tijd'], df['oppervlakte'], 1))
Nu kunnen we de **trendlijn door de data** bepalen. Je merkt dat dit gebeurt met een geneste functie: *np.poly1d* wordt toegepast na *np.polyfit*. De functie *np.polyfit* heeft volgende (vereenvoudigde) syntax:

np.polyfit(x, y, deg)

De eerste twee argumenten zijn de x- en y-waarden van de data. Het derde argument *deg* is de graad van de gewenste veelterm. Door *deg=1* te kiezen, wordt lineaire regressie toegepast.

De output van *np.polyfit* is een lijst met de coëfficiënten van de best passende rechte. De functie *np.poly1d* gebruikt deze coëfficiënten om een functie te creëren die deze rechte voorstelt. Met andere woorden, de variabele *trendlijn* is nu een functie die je kan evalueren. Bijvoorbeeld, de code

trendlijn(df['tijd'])

geeft de y-waarden van de gefitte rechte in de x-waarden van de data. Dit heb je meteen nodig in de volgende lijn.

Voorbeeld

Stel dat de best passende rechte door gegeven data *(xdata, ydata)* gegeven is door $y = 5x - 3$. Dan zou onderstaande code volgende output hebben:

Code	**Output**
np.polyfit(xdata, ydata, 1)	*array([5., -3.])*
trendlijn = np.poly1d(array([5., -3.])	*trendlijn = poly1d([5., -3])*
trendlijn(4)	*17.0*

Zorg dat je het verband tussen de commando's en de output begrijpt!

Je kan dit voorbeeld zelf uitproberen in Thonny door bijvoorbeeld *xdata = np.array([0,1,2])* en *ydata = np.array([-3,2,7])* te nemen. De best passende rechte gaat dan exact door deze punten.

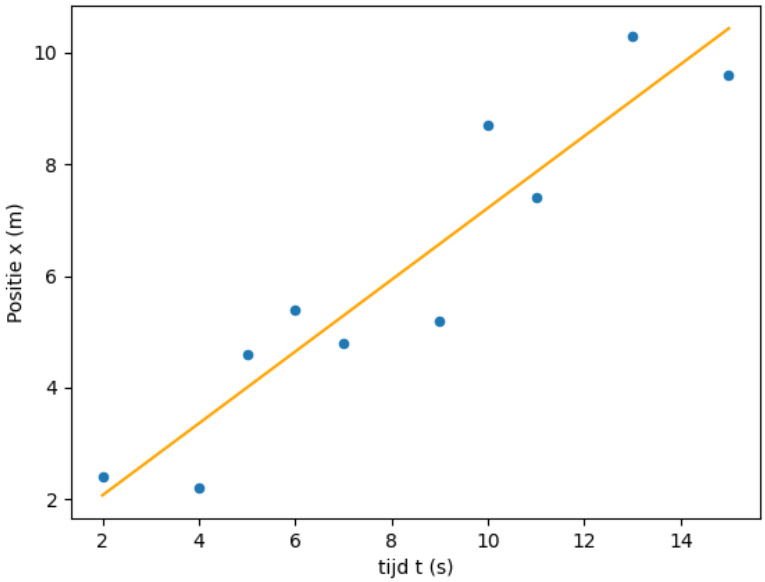
r2 = r2_score(df['oppervlakte'], trendlijn(df['tijd']))
De **determinatiecoëfficiënt** hangt af van de y-waarden van de data, in dit voorbeeld *df['oppervlakte']*, en de voorspelde y-waarden volgens het lineair model, wat we kunnen berekenen met *trendlijn(df['tijd])*. De uitkomst wordt opgeslagen in de variabele *r2*.

plt.plot(x, y, color='orange')
Tot slot voegen we de trendlijn toe aan de **figuur**. Dit doen we met behulp van de softwarebibliotheek **matplotlib**, hier afgekort tot *plt*. Het eerste argument van de functie zijn de x-waarden, het tweede argument zijn de y-waarden. Het effect van deze functie is dat een oranje lijn wordt toegevoegd aan de figuur.

Stap 3

Voer de code uit. Er verschijnt normaal een figuur. Lijkt de verkregen rechte inderdaad de best passende trendlijn? Zoniet, ga dan op zoek naar de fout in je code.

De trendlijn voor de data uit Figuur 1, bepaald met Python, ziet er bijvoorbeeld als volgt uit:



Stap 4

Bovenstaande code berekent de determinatiecoëfficiënt. Hoe kan je de waarde ervan opvragen? En hoe kan je de richtingscoëfficiënt en het intercept van de rechte opvragen?

Opdracht

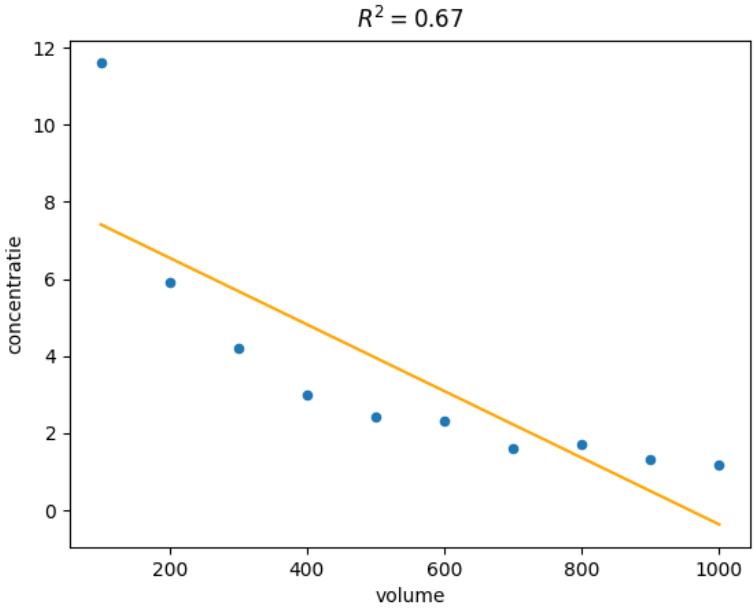
Gebruik Python om een trendlijn te bepalen voor een puntenwolk naar keuze die jullie gemaakt hebben op basis van de groeidata. Onderzoek de waarden voor het intercept, de richtingscoëfficiënt en de determinatiecoëfficiënt. Is een lineair model gepast voor de data? Waarom? Komt dit overeen met wat je verwacht na visuele inspectie van de puntenwolk?

TRANSFORMATIES

Lineaire regressie is een krachtige wiskundige methode om een model te bepalen dat bij gegeven data past. Een groot nadeel is dat het enkel een lineair model, een rechte dus, kan berekenen. Om een model te bepalen voor niet-lineaire data is nog een extra techniek nodig: **transformaties**.

Instap: concentratie in zoutoplossing

Eli maakt een oplossing van eenzelfde hoeveelheid zout in verschillende volumes water. Achteraf wordt de concentratie van de zoutoplossing gemeten. Eli verkrijgt volgende grafiek:



Is een rechte een goed model voor deze data? Hoe zie je dit?

Noteer hier wat je wil onthouden uit de inleiding van deze les.

Dit is de code die tijdens de inleiding gebruikt werd:

```
import numpy as np
import matplotlib.pyplot as plt

from data_analyse import inladen_data, fit_en_plot_rechte, plot_data_en_kromme

# data inladen
df = inladen_data("zoutconcentratie.csv")

# data omvormen via transformaties
df['concentratie (transf)'] = df['concentratie']

# lineaire regressie en figuur maken
trendlijn = fit_en_plot_rechte(df, 'volume', 'concentratie (transf)')

# grafiek toevoegen
df['model'] = trendlijn(df['volume'])
plot_data_en_kromme(df,'volume','concentratie (transf)')

plt.show()
```

Je hoeft deze code niet zelf te implementeren. Het dient als houvast tijdens de les en als inspiratie wanneer jullie je eigen model zullen implementeren.

Toepassing op bacteriële groeidata

In deelprobleem 2 verzamelden jullie groeidata van bacteriekolonies, waarvan sommigen groeiden op een substraat met een bepaalde voedingsstof, en anderen op een controlesubstraat. Het doel van dit project is om de groei van deze verschillende groepen te vergelijken. Maar daarvoor moet de groei beschreven worden met een **model**, die vervolgens **parameters** bevat met **biologische betekenis**.

Opdracht 1

Bedenk twee mogelijke modellen voor de groeidata. Doe dit op basis van de afgedrukte puntenwolken die jullie in het vorige deelprobleem maakten. Bepaal voor elk van de modellen de eventuele transformatie van de data die nodig is, en de inverse transformatie.

Vooraleer verder te gaan naar de volgende opdracht, stel je de twee modellen voor aan een leerkracht. Wanneer die groen licht geeft, kan je overgaan naar de implementatie.

Opdracht 2

Deel jullie groep op in twee deelgroepen. Per deelgroep implementeer je een van de voorgestelde modellen in Python. Werk hiervoor verder op de code uit deelprobleem 3 en het eerste deel van deelprobleem 4.

Sta telkens stil bij volgende vragen:

Geeft het gefitte model de informatie over de groei van de bacteriën die jullie verwachtten? Is de waarde van de parameter die jullie kozen als maat voor de groei van de bacteriën verschillend voor beide voedingsbodems? Is dit te verwachten?