



Unit 4: Het gebruik van de ti_plotlib module

Oefenblad 3: Puntenwolken tekenen

In deze les tekenen we een puntenwolk waarna we met lineaire regressie de best passende lijn erbij zoeken.

Doelen :

- Het tekenen van een puntenwolk.
- Lineaire regressie uitvoeren.

We bekijken het volgende probleem.

Na een overstroming in een gebied wordt elk uur de waterhoogte gemeten.

De resultaten staan in de volgende tabel:

t(uren)	0	1	2	3	4	5	6	7	8	9	10	11	12
H(cm)	130	127	123	118	116	111	105	103	101	95	86	80	71

We maken een programma dat een grafiek tekent van deze metingen en onderzoeken het verband tussen t en H.

Open een nieuw Python programma en voeg ti_plotlib module in.

Als eerste maken we twee lijsten met de gegevens uit de tabel.

```

1.1 1.2 *Doc RAD
*Waterhoogte.py 5/5
import ti_plotlib as plt

t=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
H=[130,127,123,118,116,111,105,103,101,95,86,

```

Voor de grafiek hebben we een geschikt assenstelsel nodig.

Met de opdracht plt.window() kun je de vensterinstellingen aanpassen.

De functie **plt.scatter (t,H,"x")** tekent de puntenwolk.

Hierbij is t de lijst met de horizontale gegevens, H de lijst met de verticale en "x" het symbool voor de punten.

(De functie scatter() is te vinden in het ti_plotlib menu onder tabblad Draw).

```

1.1 1.2 *Doc RAD
*Waterhoogte.py 10/10
import ti_plotlib as plt

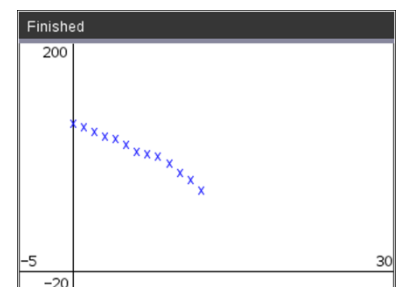
t=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
H=[130,127,123,118,116,111,105,103,101,95,86,

plt.window(-5,30,-20,200)
plt.axes("on")
plt.color(0,0,255)
plt.scatter(t,H,"x")

```

Voer het programma uit.

Je ziet nu de grafiek met de punten uit de tabel weergegeven met een x.





De punten lijken redelijk op een lijn te liggen.
De beste rechte door de punten heet de regressie lijn.

We kunnen deze lijn tekenen met de opdracht `plt.lin_reg(t,H,"center")`.
Het woord "center" geeft aan dat de bijbehorende vergelijking gecentreerd op het scherm wordt weergegeven.
In het voorbeeld hiernaast zijn nog wat extra opdrachten toegevoegd om het plaatje nog wat aan te passen

```

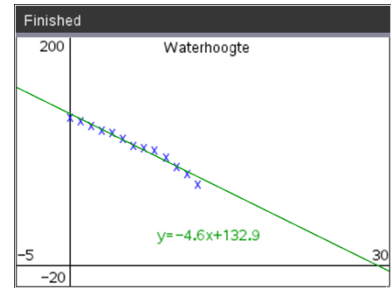
1.1 1.2 *Doc RAD 11/13
*Waterhoogte.py
import tiplotlib as plt

t=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
H=[130,127,123,118,116,111,105,103,101,95,86,
plt.window(-5,30,-20,200)
plt.axes("on")
plt.title("Waterhoogte")
plt.color(0,0,255)
plt.scatter(t,H,"x")
plt.color(0,155,0)
plt.lin_reg(t,H,"center")

```

Het tijdstip waarop de waterhoogte 0 is kun je nu berekenen met de vergelijking $-4.6x + 132.9 = 0$

De getallen -4.6 en 132.9 zijn afgerond op een decimaal.
De onafgeronde waarden van deze getallen zijn opgeslagen in de variabelen `plt.m` en `plt.b` die je kunt vinden in het `tiplotlib` menu bij Properties.



Als je in de shell deze variabelen intypt kun je de waardes bekijken.

```

1.1 1.2 *Doc RAD 7/7
Python Shell
>>>#Running Waterhoogte.py
>>>from Waterhoogte import *
>>>plt.m
-4.637362637362638
>>>plt.b
132.9010989010989
>>>

```