

Kapitel 4 : Mer om ti-plotlib-modulen

Övning 3: Spridningsdiagram

Här skall vi titta på hur man ritar ett spridningsdiagram och hur man finner en regressionslinje.

Mål:

- Att rita spridningsdiagram
- Att hitta regressionslinjer

Vi ska analysera data från en översvämning. Under en översvämning i ett område mäts vattennivån varje timme. Resultaten visas i följande tabell:

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

Här är t tiden i timmar och h djupet i cm.

Vi ska göra ett program som skall rita ett spridningsdiagram över ovanstående data och sedan finna den räta linje som passar punkterna bäst.

Öppna ett nytt Python-program(meny>Åtgärder>Nytt..) och lägg till ti_plotlib-modulen.

Skapa två listor med värdena från t och D.



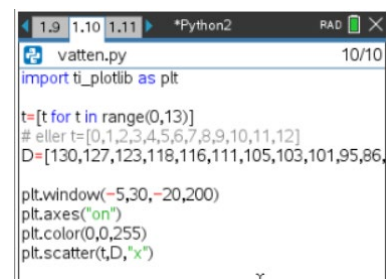
```
1.9 1.10 1.11 *Python2 RAD 2/6
*vatten.py
import ti_plotlib as plt

t=[t for t in range(0,13)]
# eller t=[0,1,2,3,4,5,6,7,8,9,10,11,12]
D=[130,127,123,118,116,111,105,103,101,95,86,
```

För diagrammet behöver vi ett lämpligt koordinatsystem.

Med kommandot plt.window () kan du justera fönsterinställningarna. Lägg även till att axlarna skall synas. Med plt.color kan du ställa in färgen som punkterna skall ha. Vi skall ha blå punkter, så vi använder plt.color(0,0,255).

Slutligen så använder vi funktionen **plt.scatter(t,D,"x")** för att skapa vårt spridningsdiagram. Parametern "x" i slutet markerar att punkterna skall visas som små kryss. Du hittar scatter under meny>TI PlotLib>Draw

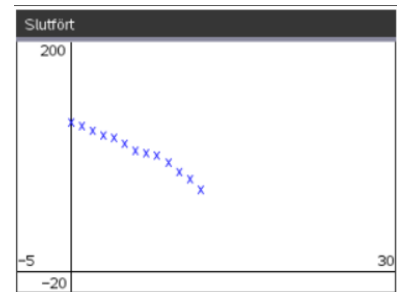


```
1.9 1.10 1.11 *Python2 RAD 10/10
vatten.py
import ti_plotlib as plt

t=[t for t in range(0,13)]
# eller t=[0,1,2,3,4,5,6,7,8,9,10,11,12]
D=[130,127,123,118,116,111,105,103,101,95,86,

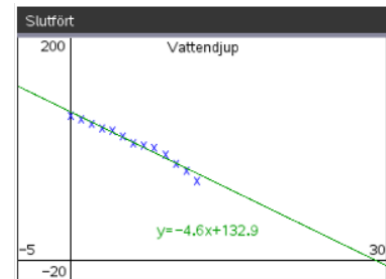
plt.window(-5,30,-20,200)
plt.axes("on")
plt.color(0,0,255)
plt.scatter(t,D,"x")
```

Om du nu kör programmet borde du ha fått fram en graf liknande den till höger.



Punkterna verkar ligga ganska nära en rät linje. Den linje som passar bäst enligt den så kallade minstakvadratmetoden kallas för regressionslinjen. Vi kan rita denna linje med kommandot `plt.lin_reg(t, D, "center")`.

Ordet "center" indikerar att ekvationen vi får fram placeras centrerad (i horisontalled) på skärmen.



I exemplet har några extra kommandon lagts till.

```
1.9 1.10 1.11 *Python2 RAD 2/12
*vatten.py
import tiplotlib as plt

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

Vi får nu fram att regressionslinjen har ekvationen

$$y = -4,6x + 132,9$$

där x är tiden och y är vattendjupet.

Vi kan använda denna ekvation till att exempelvis beräkna tiden tills vattendjupet är 0 med ekvationen

$$-4,6x + 132,9 = 0$$

Talen $-4,6$ och $132,9$ har avrundats till en decimal. Dessa bägge värden lagras i variablerna `plt.m` och `plt.b` som du hittar under meny>Ti Plolib>Egenskaper.

Om du tar fram dessa variabler i konsolen så kan du där se dem med fler decimaler. Du kan också använda dem för vidare beräkningar i programmet såväl som i konsolen.

```
1.9 1.10 1.11 *Python2 RAD 7/7
Python Shell
>>>#Running vatten.py
>>>from vatten import *
>>>plt.m
-4.637362637362638
>>>plt.b
132.9010989010989
>>>
```