

Kapitel 4: Använda biblioteksmodulen ti-plotlib

Övning 3: Grafiskt representera en funktion

I den tredje övningen i kapitel 4 får du lära dig hur du grafiskt representerar en uppsättning data och analyserar dessa data, och sedan beräknar en matematisk modell som anpassar sig till datamängden.

Scenario: Under en översvämning registrerade myndigheterna varje timme förändringen av den maximala vattennivån i förhållande till en referenspunkt.

Datan finns i tabellen nedan. Du ska representera data med hjälp av `ti_plotlib`-biblioteket och sedan leta efter en matematisk modell för att göra en extrapolering. Skälet till detta är att kunna förutsäga när vattennivån blir noll i förhållande till en referenspunkt.

Syfte:

- grafiskt representera en uppsättning av data
- Undersöka om det finns någon bra regressionsmodell
- Använda listor i Python



T(h)	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

Implementering: Starta ett nytt skript och namnge det +AP4OVN3.

1. Importera modulen `ti_plotlib`.
2. Spara data i två listor, "t" och "h".
3. Förbered sedan den grafiska representationen med bland annat följande:
 - Rensa skärmen `plt.cls()`.
 - Ställ sedan in det grafiska fönstret:
xmin -5, xmax 35, ymin -25, ymax 250
genom att skriva `plt.window(-5, 35, -25, 250)`
 - Visa axlar: `plt.axes("on")`
 - Visa rutnät: `plt.grid(5,50,"dot")`
 - Visa den grafiska representationen av data som ett spridningsdiagram med instruktionen `plt.scatter(t,h,"+")`.

```

EDITOR: KAP4OVN3
PROGRAM LINE 0010
t=[0,1,2,3,4,5,6,7,8,9,10,11,12]
h=[130,127,123,118,116,111,105,103,101,95,86,80,71]
plt.cls()
plt.window(-5,35,-25,250)
plt.axes("on")
plt.grid(5,50,"dot")
plt.title("vattennivå")
plt.scatter(t,h,"+")
plt.show_plot()_
Fns... a A # Tools Run Files

```

Instruktionerna för fönsterinställning av grafik finns i modulen `ti_plotlib` på menyn **Setup**. Instruktioner för plottning och datavisning finns på menyn **Draw**.

Lärarkommentar: För att slippa skriva in data 1, 2 ...12 så kan du göra en avgränsad loop enligt följande:

```
PYTHON SHELL
>>> lt=[i for i in range(13)]
>>> lt
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
>>> |
```

Observera att värdet noll alltid kommer med.

Kör skriptet och observera den grafiska representationen. Om allt gått bra bör du ha fått skärmen här till höger.

Verkar alla punkter ligga någorlunda i linje?

Du ska nu söka efter den matematiska modellen med en metod man kallar för linjär anpassning eller linjär regression. Vi ska söka en linje som så bra som möjligt ligger nära datapunkterna.

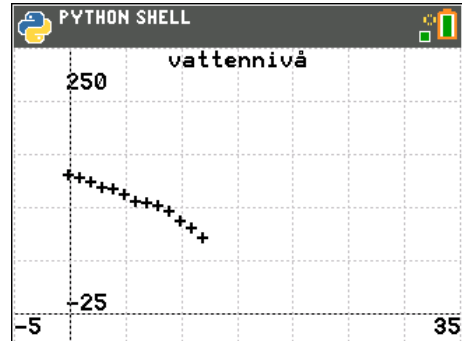
För att göra detta måste du lägga till raden

```
plt.lin_reg(t, h, "center", 11)
```

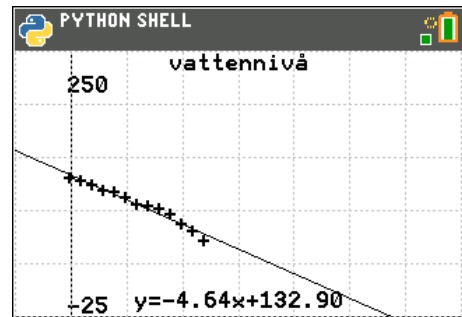
i skriptet så att regressionslinjen beräknas från listorna t och h och sedan visas och centreras på linje 11 på skärmen.

Till höger ser vi resultatet. Längst ner på skärmen visas den beräknade regressionslinjen. Vi vill nu veta när den skär x-axeln (nivån=0). Det sker när $y=0$ dvs vid $x=132,90/4,64 \approx 28,6$.

Genom att lägga till två instruktioner för färg i skriptet kan vi få det ännu snyggare. Vi vill få datapunkterna inkl. grafens titel i en färg och regressionslinjen inkl. den beräknade ekvationen i en annan färg. Se skriptet här till höger och grafen nedan.



```
EDITOR: KAP4OVN3
PROGRAM LINE 0012
h=[130,127,123,118,116,111,105,103,101,95,86,80,71]
plt.cls()
plt.window(-5,35,-25,250)
plt.axes("on")
plt.grid(5,50,"dot")
plt.title("vattennivå")
plt.scatter(t,h,"+")
plt.lin_reg(t,h,"center",11)
plt.show_plot()
_
```



```
EDITOR: KAP4OVN3
PROGRAM LINE 0014
plt.cls()
plt.window(-5,35,-25,250)
plt.axes("on")
plt.color(255,0,0)
plt.grid(5,50,"dot")
plt.title("vattennivå")
plt.scatter(t,h,"+")
plt.color(0,0,255)
plt.lin_reg(t,h,"center",11)
plt.show_plot()
```

