

Lektion 4 : Verwendung des Moduls tiplotlib

Übung 3 : Punktwolke und Regression

In dieser dritten Übung der Lektion 4 wird vermittelt, wie man grafisch eine Liste von Messpunkten darstellt und es wird ein mathematisches Modell vorgestellt, das sich dieser Punktwolke optimal anpasst.

Lernziele :

- Darstellung einer Punktwolke.
- Verwendung einer geeigneten Regression.
- Verwendung von Listen in Python.

Das Problem :

Um die Bevölkerung während eines Hochwassers mit praktischen Informationen zu versorgen, zeichnen die Behörden stündlich (t/h) ab Beginn des sinkenden Wasserspiegels den maximalen Wasserstand (h/cm) in Bezug auf einen Referenzpunkt auf. Die Daten sind in der nachstehenden Tabelle aufgeführt. Man möchte die Punktwolke mit Hilfe der **tiplotlib**-Bibliothek darstellen und dann nach einem mathematischen Modell zur Extrapolation suchen, um den Verlauf des sinkenden Wasserstandes vorherzusagen.



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

Das Programm :

- 1) Ein neues Programm mit dem Namen **U4SB3** anlegen.
- 2) Das Modul **tiplotlib** importieren.
- 3) Messwerte in die Listen « **lt** » und « **lh** » eintragen. Lange Listen ragen dabei rechts aus dem Bildschirm heraus, bleiben aber editierbar.
- 4) Die grafische Darstellung vorbereiten :
 - Den Bildschirm löschen : **plt.cls()**.
 - Die Fenstereinstellungen $x_{min} = -2$; $x_{max} = 20$; $y_{min} = -20$; $y_{max} = 200$ durch **plt.window(-2, 20, -20, 200)** vornehmen.
 - Koordinatenachsen hinzufügen : **plt.axes()**.
- 5) Die Messwerte als Punktwolke darstellen : **plt.scatter()**.

Die Einstellungen unter 4) findet man im Menü **SetUp**, die Syntax zur Darstellung der Punktwolke unter 5) im Menü **Draw**.

```

1.1 *Dok RAD 4/4
*U4SB3.py
import tiplotlib as plt
lt=[0,1,2,3,4,5,6,7,8,9,10,11,12]
lh=[130,127,123,118,116,111,105,103,101,95,86,

```

```

1.1 *Dok RAD 8/8
*U4SB3.py
import tiplotlib as plt
lt=[0,1,2,3,4,5,6,7,8,9,10,11,12]
lh=[130,127,123,118,116,111,105,103,101,95,86,
plt.cls()
plt.window(-2,20,-20,200)
plt.axes("on")
plt.scatter(lt,lh,"+")
plt.show_plot()

```

Lehertipp : Die Liste der Zeiten kann auch durch eine For-Schleife erzeugt werden.

```

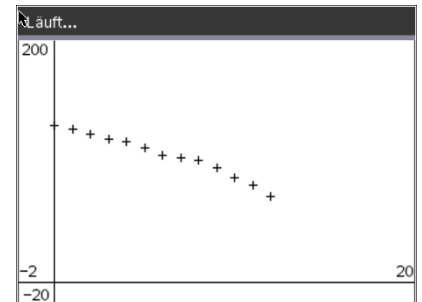
1.1 1.2 *Dok RAD 4/4
Python-Shell
>>>lt=[i for i in range(13)]
>>>lt
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
>>>

```

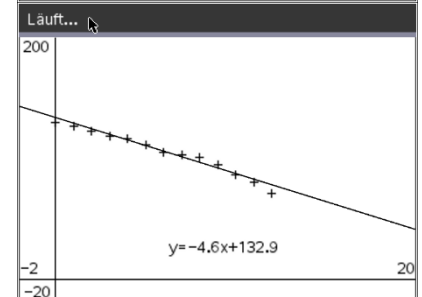
Das Programm sollte bei fehlerfreier Ausführung das nebenstehende Bild erzeugen..

Es sieht so aus, als würde sich der Verlauf durch eine lineare Funktion, die sich der Punktwolke anpasst, gut beschreiben lassen können. Das geeignete Modell dafür ist die **lineare Regression**, die sich in Python einfach durchführen lässt.

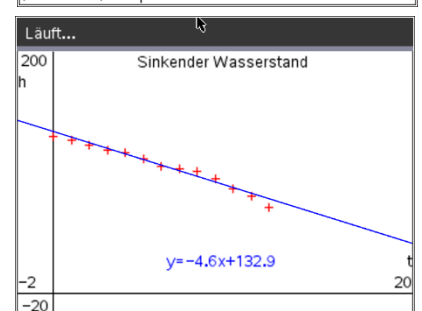
Dazu wird die Zeile `plt.lin_reg(lt,lh, « center »,11)` in das Programm eingefügt. Die Regressionsgleichung wird dann mittig (**center**) in **Zeile 11** auf dem Bildschirm eingefügt. Die Zeilenangabe ist optional, Zeile 11 ist die Voreinstellung. Um den Zeitpunkt zu bestimmen, wo der Wasserstand wieder die Höhe des Referenzpunktes hat, muss dann lediglich die lineare Gleichung $-4.64x + 132.90 = 0$ gelöst werden ($x = 28,6$).



```
U4SB3.py 9/10
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.cls()
plt.window(-2,20,-20,200)
plt.axes("on")
plt.scatter(t,h,"+")
plt.lin_reg(t,h,"center",11)
plt.show_plot()
```



```
U4SB3.py 14/14
plt.cls()
plt.window(-2,20,-20,200)
plt.color(0,0,0)
plt.axes("on")
plt.labels("t","h",11,2)
plt.title("Sinkender Wasserstand")
plt.cjlor(255,0,0)
plt.scatter(t,h,"+")
plt.color(0,0,255)
plt.lin_reg(t,h,"center",11)
plt.show_plot()
```



Verbesserung der grafischen Darstellung

Das betrifft die farbliche Ausgestaltung. So könnte man z.B. die Achsen schwarz (`plt.color(0,0,0)`), die Punktwolke rot (`plt.color(255,0,0)`) und die Regression blau darstellen (`plt.color(0,0,255)`).

Man kann auch noch einen **Titel** und die **Beschriftung der Achsen** hinzufügen.

Lehertipp : Um Überschneidungsprobleme bei der Anzeige zu vermeiden, sollten kurze Namen für die Achsenbeschriftungen gewählt werden.

Die Beschriftungen für die "t"- und "h"-Achse können mit der Beschriftungsanweisung ("x-label", "y-label", x, y) beliebig eingestellt werden, wobei x und y die Linie darstellen, auf der diese Beschriftungen geschrieben werden. Standardmäßig sind diese Linien 12 bzw. 2 für x bzw. y sowie links- bzw. rechtsbündig.

In ähnlicher Weise kann die Regressionsgleichung mit dem Befehl `lin_reg(xlist, ylist, "disp", row)` an der gewünschten Stelle angezeigt werden. Standardmäßig wird diese Gleichung in Zeile 11 angezeigt