Lektion 5 : Verwendung des Moduls ti_system

Übung 2 : Modellierung

In dieser zweiten Übung der Lektion 5 wird gezeigt, wie man das Ergebnis einer Modellierung unter Verwendung des Moduls **ti_system** importiert.

Lernziele :

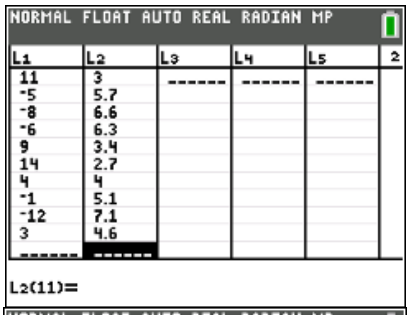
- Durchführung einer linearen Regression.
- Importieren der Ergebnisse in ein Python-Programm.

In dieser Lektion wird eine lineare Regression mit Daten durchgeführt, die zuvor in die Listen des Taschenrechners eingegeben wurden. Dann wird ein Programm geschrieben, um die Ergebnisse dieser Modellierung zu importieren, grafisch darzustellen, sie für eine Interpolation oder auch eine Extrapolation zu verwenden.

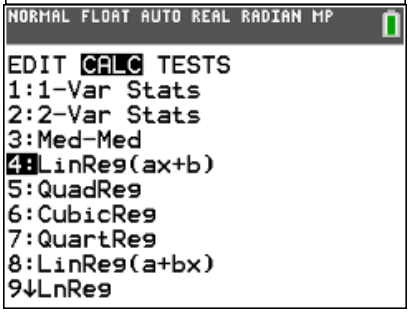
Die Aufgabe: An einem Tag erreicht der Stromverbrauch gegen 19:00 Uhr seinen Höhepunkt. Auf nationaler Ebene wurde am Mittwoch, den 15. Dezember 2019, um 19:02 Uhr ein Spitzenverbrauch von 96.350 Megawatt verzeichnet.
Nun soll ein lokales Kraftwerk und dessen nächste Umgebung betrachtet werden. An 10 Wochenenden hat man die Temperatur am Kraftwerk und die erzeugte Leistung gemessen. Die Messwerte sind in der Tabelle dargestellt. Eine Prognose für das nächste Wochenende soll daraus abgeleitet werden.

°C	11	-5	-8	-6	9	14	4	-1	-12	3
MW	3	5,7	6,6	6,3	3,4	2,7	4	5,1	7,1	4,6

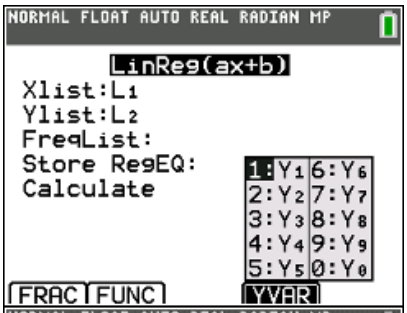
Die Messwerte werden in die zwei Listen $\mathbf{L_1}$ (Temperatur) und $\mathbf{L_2}$ (Verbrauch) eingetragen.



Unter stat **CALC** findet sich eine lineare Regression der Form $y = ax + b$.

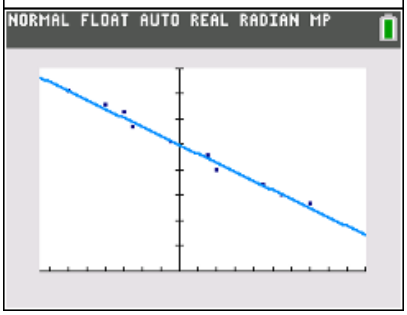


Die Maske wird entsprechend dem nebenstehenden Bild ausgefüllt und die Regression in Y_1 gespeichert.



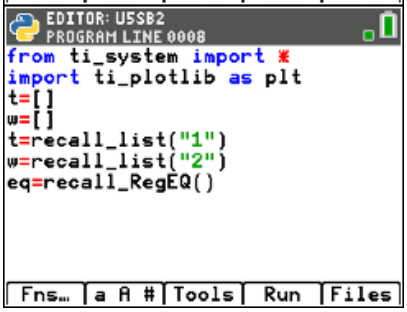
Lehrertipp : Mit alpha trace erhält man schnell den Funktionsnamen Y_1.

Die Abhängigkeit des Verbrauchs Wvon der Temperatur t lässt sich also beschreiben durch die lineare Funktion : $W = -0.18 \times t + 5.01$.

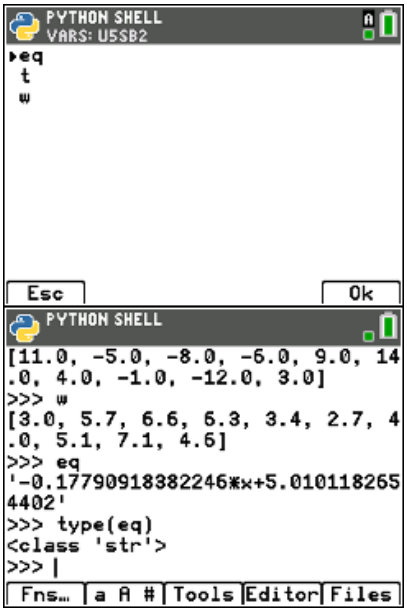


Verwendung der Modellierung in einem Python-Programm

- Anlegen eines neuen Programmes mit dem Namen U5SB2.
- Importieren der Module **ti_system** und **ti_plotlib**.
- Erzeugen zweier leerer Listen **t = [ ]** und **w = [ ]** .
- Füllen der Listen durch die Anweisung **var=recall_list(« name »)** aus dem Modul **ti_system** (s. Übung 5.1).
- Laden der Regressionsfunktion **eq** durch die Anweisung **var=recall_RegEQ()**



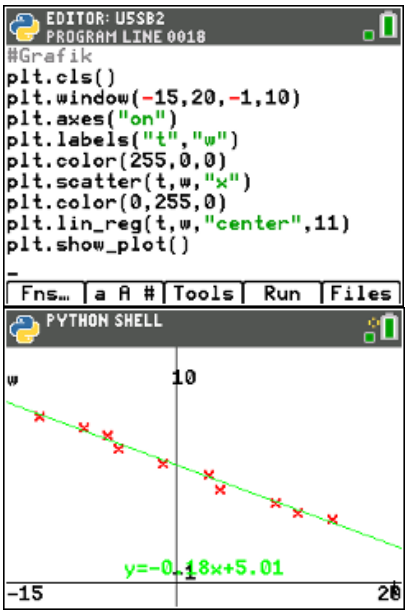
Jetzt kann man das Programm testen, indem man die einzelnen Variablen über vars aufruft.



Hinweis : Die Regressionsfunktion wird als Zeichenkette (String) importiert. Sie kann jedoch nicht als Funktion gezeichnet werden. Dazu benötigt man in Python eine Neuberechnung.

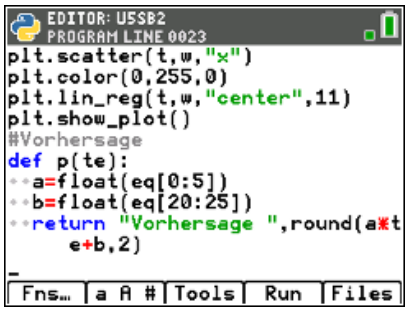
Zur grafischen Darstellung muss die Grafik wieder in der üblichen Weise eingerichtet werde.
Hinweis : Die Anweisung **plt.auto_window(x_list, y_list)** stellt die Fenstergrenzen automatisch nach den Extremwerten der Listen ein, funktioniert also ähnlich wie **ZoomStat**.

Lehrertipp : Die Anweisung **lin_reg(xliste, yliste, « disp », row)** beinhaltet die Möglichkeit einer zusätzlichen Angabe der Zeile, auf der die Regressionsgleichung dargestellt werden soll. Im Beispiel ist es die 11. Zeile..



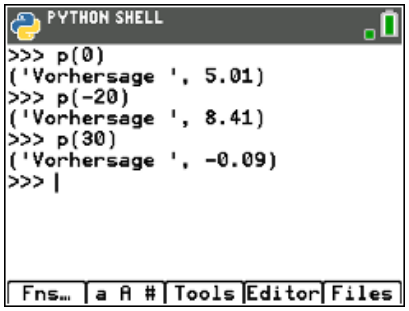
Ergänzung : Vorhersage für den Verbrauch bei einer bestimmten Temperatur

Dazu müssen die Koeffiezienten der Regressionsgleichung aus dem String extrahiert und in zwei Variablen a und b gespeichert werden. Dann kan man sie in einer Funktion verwenden, mit der man eine Extrapolation oder eine Interpolation der Leistungsaufnahme bei bekannter Temperatur durchführen kann.



Lehrertipp: Mit der Anweisung **v = float(eq[0: i+1])** kann man in der Variablen v die ersten i Elemente der Zeichenfolge eq speichern.

Damit kann man eine Funktion p(t) definieren, die den erwarteten Verbrauch für eine bestimmte Temperatur t liefert.



Man kann mit dem Programm einige Tests für verschiedene Temperaturen durchführen. So kann man die Gültigkeitsgrenzen des Modells festlegen.