T³ EUROPE

Workshops zum TI-83 PLUS

Beiträge von T³ Flandern / Belgien

Ein Unterrichtsbehelf zum Einsatz moderner Technologien
im Mathematikunterricht

T³ Österreich / ACDCA am PI-Niederösterreich, Hollabrunn

Vorwort

Anlässlich unserer gemeinsamen Arbeit in der CAS-Focus-Group erhielt ich von Guido Herweyers die Unterlagen zum 29. Kongress der belgischen Mathematiklehrer, der im August 2003 in Brüssel abgehalten wurde.

Alle vier Beiträge befassen sich mit dem grafischen Taschenrechner TI-83 PLUS. Die belgischen Kollegen haben schon ein lange Erfahrung mit diesem Gerät und nach Durchsicht der Workshops fand ich es sehr sinnvoll, diese Papiere auch unseren deutschsprachigen Kolleginnen und Kollegen zugänglich zu machen.

Ich danke Guido für die Erlaubnis zur Übersetzung ins Deutsche. Ich hoffe, dass dies der Anfang einer Kooperation ist, die vielleicht überhaupt Schule machen könnte. T^3-Europa stellt so viele erstklassige Unterrichtsmaterialien in den verschiedenen Ländern her und leider werden die Synergien nur sehr spärlich genutzt.

Guido hat seinerseits bereits Interesse an einigen von unseren Papieren angemeldet.

Ich möchte die Gelegenheit auch nützen, alle belgischen T^3-Freunde herzlich zu grüßen und zu einer weiteren intensiven und fruchtbaren Zusammenarbeit einzuladen.

Der erste Workshop von Hans Bekaert gibt einen Überblick über die Zusammenarbeit zwischen den möglichen Plattformen, wie GTR und PC, wobei auch TI-InterActive und TI-Connect vorgestellt werden.

Guido Herweyers hat anlässlich seines Besuchs beim Seminar im Frühjahr 2003 seine hohe Kompetenz im Umgang mit Statistik gezeigt. Hier dringt er noch ein wenig tiefer in die Materie ein und demonstriert eindrucksvoll das Zusstandekommen und die Bedeutung von Korrelationskoeffizient und Bestimmtheitsmaß.

Im dritten Beitrag zeigt Koen Stulens, wie einfach Simulationen am TI-83+ nicht nur durchgeführt, sondern auch ausgewertet werden können. Ich habe mir erlaubt, diesen Beitrag um zwei Beispiele zu erweitern und eine recht brauchbare Applikation für den TI-83+ vorzustellen.

Und schließlich endet Regis Ockermann mit einer sehr verständlichen und behutsamen Einführung in die Programmiertechnik am TI-83+. Ich habe absichtlich die Texte im Programm in ihrem originalen holländischen Wortlaut belassen. Die deutsche Fassung kann jeder selbst erzeugen.

Ich wünsche allen Lesern viel Ertrag beim Durchlesen oder –arbeiten dieser T^3-Unterlage.

Josef Böhm
T^3-Östereich

INHOUD
INHALT

Workshops

De grafische rekenmachine koppelen aan de computer
Der grafische Taschenrechner und der Computer
Hans Bekaert

Regressie, correlatie en modelvorming met de TI-83 Plus
Regression, Korrelation und Modellbildung mit dem TI-83 Plus
Guido Herweyers

Simulatie van kansexperimenten met de TI-83 Plus
Simulation von Zufallsexperimenten mit dem TI-83 Plus
Koen Stulens (& Ergänzung von Josef Böhm)

Inleiding tot het programmeren met de TI-83 Plus
Einführung in das Programmieren mit dem TI-83 Plus
Regis Ockerman

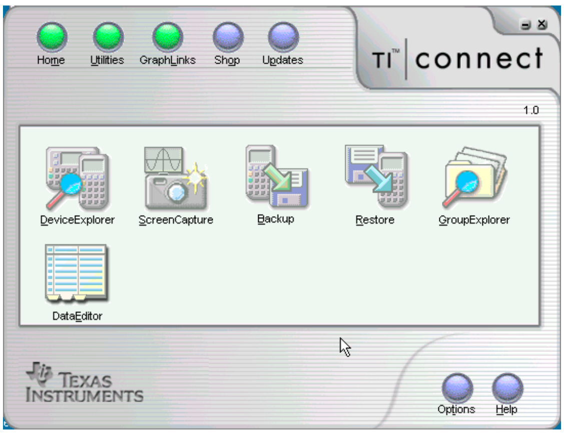
Der graphische Taschenrechner und der Computer

Hans Bekaert
Limburgs Universitair Centrum
Diepenbeek

TI Connect™

Um Daten aus dem GTR (TI-83+) auf dem Computer zu bearbeiten oder um ein Backup einer Liste oder einer Matrix vom PC auf den GTR zu übertragen, kann man TI Connect einsetzen. Dieses Programm ist eine interessante Weiterentwicklung der TI-GraphLink™ Software.

Kurze Übersicht



Mit dem **DeviceExplorer** sendet man Daten vom Rechner zum Computer.

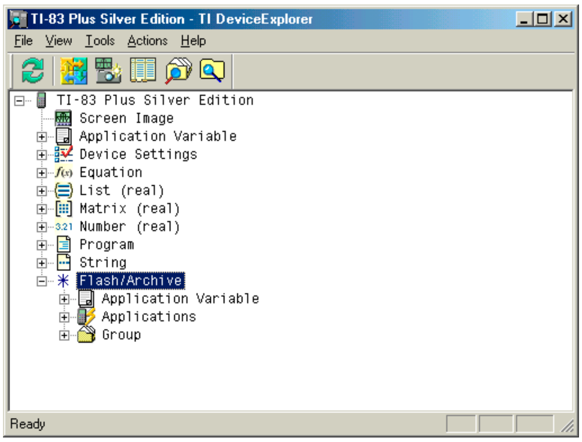
Mittels **ScreenCapture** kopiert man den Bildschirm des Rechners in die Zwischenablage oder speichert das Bild als eine Grafikdatei.

Mittels **Backup** kann man die Daten des Rechners extern (auf Diskette, Festplatte, ...) sichern und über **Restore** kann man sie dann auf dem Rechner wieder herstellen.

Mit dem **GroupExplorer** kann man logisch zusammengehörige Dateien zu einer Gruppe (gruppierte Datei) auf dem Computer zusammen fassen.

Mit dem **DataEditor** kann man Listen, Zahlen oder Matrizen, die vom GTR stammen auf dem Computer bearbeiten. Man kann diese aber auch am Computer erzeugen und dann zum Rechner senden.

DeviceExplorer



Der *DeviceExplorer* gibt ermöglicht die Übersicht über die auf dem Taschenrechner vorhandenen Dateien und Daten.

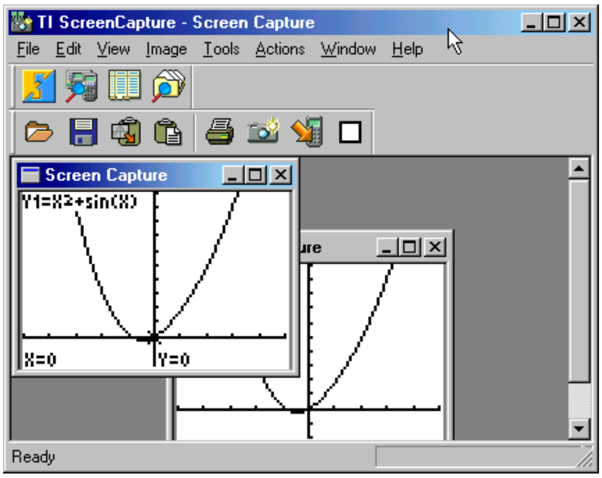
Klicke auf das +-Zeichen um die gewünschten Daten betrachten zu können.

Wähle ein Objekt aus und aktiviere im ACTIONS-Menü die Option COPY TO PC. Nun fragt das Programm, wohin die Daten übertragen werden sollen.

Im WINDOWS-Explorer findet man die TI-83+ -Dateien wieder. Sie sind an den Dateiendungen .8xl , .8xg , .8xk, ... zu erkennen.

Für den Datentransfer vom PC zum Rechner arbeitet man am besten mit dem WINDOWS-Explorer. Suche die entsprechende Datei auf, klicke mit der rechten Maustaste drauf und wähle SENDEN AN und dann CONNECTED TI DEVICE. Das lässt sich auch auf mehrere gemeinsam ausgewählte Dateien anwenden.

ScreenCapture



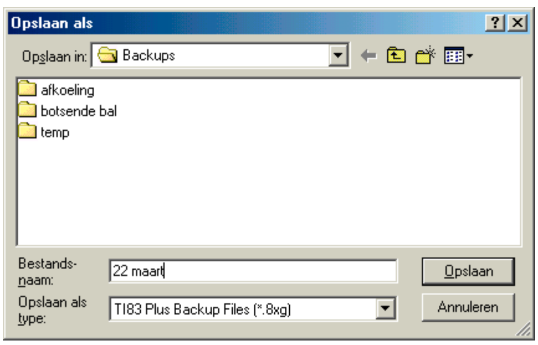
Mit *ScreenCapture* kann eine Kopie des Rechnerschirms erzeugt werden, die als Grafik in Dokumente oder Tabellen eingebunden werden können.

Mit dem Icon GET SCREEN wird die Kopie erzeugt. Es lassen sich mehrere Kopien gleichzeitg betrachten.

Wähle die gewünschte Kopie aus und kopiere sie mit dem COPY-Icon in die Zwischenablage. (Mit der „Diskette“ kann die Grafik auf dem PC gespeichert werden.)

Wechsle nun in die Anwendung, in der die Abbildung verwendet wird und füge sie aus der Zwischenablage ein.

Backup & Restore



Mit *Backup* lässt sich ein vollständiges Sicherungsbackup des Rechners durchführen. Das Backup besteht aus einer Gruppe aller Dateien. Das Programm fragt, wo das Backup am PC gespeichert werden soll.

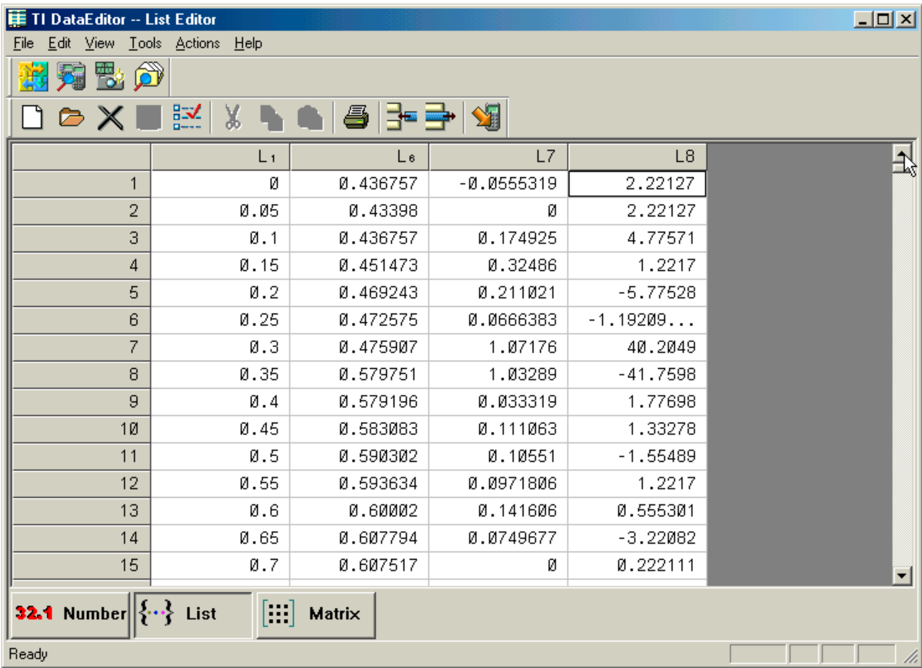
Mit *Restore* werden die gesicherten Daten wieder in ihrer ursprünglichen Form und Anordnung auf den Rechner rückübertragen.

GroupExplorer

Mit Hilfe des *GroupExplorers* kann man Dateien, die zusammengehören, in einer Gruppendatei auf dem Computer zusammenfassen. Damit erfolgt das Kopieren und Verschieben dieser zusammengehörigen Dateien rascher und fehlerfreier. Markiere die entsprechenden Dateien im *GroupExplorer*-Fenster und wähle über die rechte Maustaste CREATE GROUP. Sobald eine Gruppe markiert ist, wähle wieder über die rechte Maustaste die Option SEND TO DEVICE. Die Dateien aus der Gruppe werden eine nach der anderen zum Rechner geschickt. Auf dem Rechner ist die Gruppe nicht als solche zu erkennen.

Eine Gruppe kann auch auf dem Rechner (über 2nd [MEM] - 8:GROUP) zusammengestellt werden. Das macht die Weitergabe von Dateien über den *DeviceExplorer* wesentlich einfacher. Die Dateien erscheinen dann auch als Gruppe am PC. Wenn man nun die Gruppe markiert und über die rechte Maustaste die Option EXTRACT ALL wählt, dann lassen sich die Einzeldateien auch am PC nutzen, wie z.B. über den *DataEditor*.

DataEditor



	L1	L6	L7	L8
1	0	0.436757	-0.0555319	2.22127
2	0.05	0.43398	0	2.22127
3	0.1	0.436757	0.174925	4.77571
4	0.15	0.451473	0.32486	1.2217
5	0.2	0.469243	0.211021	-5.77528
6	0.25	0.472575	0.0666383	-1.19209...
7	0.3	0.475907	1.07176	40.2049
8	0.35	0.579751	1.03289	-41.7598
9	0.4	0.579196	0.033319	1.77698
10	0.45	0.583083	0.111063	1.33278
11	0.5	0.590302	0.10551	-1.55489
12	0.55	0.593634	0.0971806	1.2217
13	0.6	0.60002	0.141606	0.555301
14	0.65	0.607794	0.0749677	-3.22082
15	0.7	0.607517	0	0.222111

- Im *DataEditor* kann man Daten betrachten und in andere Anwendungen übernehmen. Drei Datenarten können so behandelt werden: Zahlen, Listen und Matrizen. Man muss zuerst die entsprechende Wahl (siehe Leiste unten) treffen, bevor man die Datei öffnen kann  .
- Nun lassen sich allfällige Änderungen vornehmen und die geänderten Daten wieder auf den Rechner übertragen.

  Man markiert den Spaltenkopf der zu übernehmenden Liste und drückt auf .
  So lassen sich auch mehrere über § selektierte Listen gleichzeitig übertragen.
- Die markierten Listen können über die Zwischenablage auch nach Excel übertragen werden. Beachte dabei, dass der *DataEditor* einen *Dezimalpunkt* verwendet. Das kann beim Kopieren in eine Excel-Tabelle Probleme machen. Man muss daher vor dem Kopiervorgang in den generellen WINDOWS-Einstellungen in der Systemsteuerung dafür sorgen, dass in den Ländereinstellungen als Dezimaltrennzeichen ein Punkt (anstelle des Komma) angegeben wird. (Dies gilt auch für den Einsatz von TI-InterActive!)

- Will man aber die Listen sofort als Exceldokument nützen, kann man im Menü FILE die Optionen EXPORT oder SPECIAL LIST EXPORT anwenden. Mit EXPORT spricht man jede einzelne Liste an, während SPECIAL LIST EXPORT alle Listen in eine gemeinsame Text- oder Exceldatei umwandelt und als solche speichert.

Ein ausführliches Beispiel für den Datentransfer nach Excel

1. Sammle die Messdaten. Halte fest, in welchen Listen die entsprechenden Daten zu finden sind.

2. Verbinde den Rechner mit dem PC (*GraphLink*-Kabel).

3. Starte *TI Connect.*

4. Aktiviere den *DeviceExplorer*

5. Markiere die gewünschten Listen und wähle im ACTIONS-menu die Option COPY TO PC.

6. Rufe den *DataEditor* auf.

7. In der unteren Leiste wähle das „Listen"-Icon.

8. Öffne die Listen.

9. Falls notwendig ändere die Ländereinstellungen auf den Dezimalpunkt. Ein Computerneustart ist i.a. nicht notwendig.

10. Markiere die Listen, die nach Excel übertragen werden sollen.

11. Klicke auf KOPIEREN.

12. Öffne Excel, markiere die gewünschte Zellenposition und klicke auf EINFÜGEN.

- Mit numerischen Daten und mit Matrizen kann man völlig analog verfahren. Eine Matrix kann auch über EXPORT nach Excel exportiert werden.

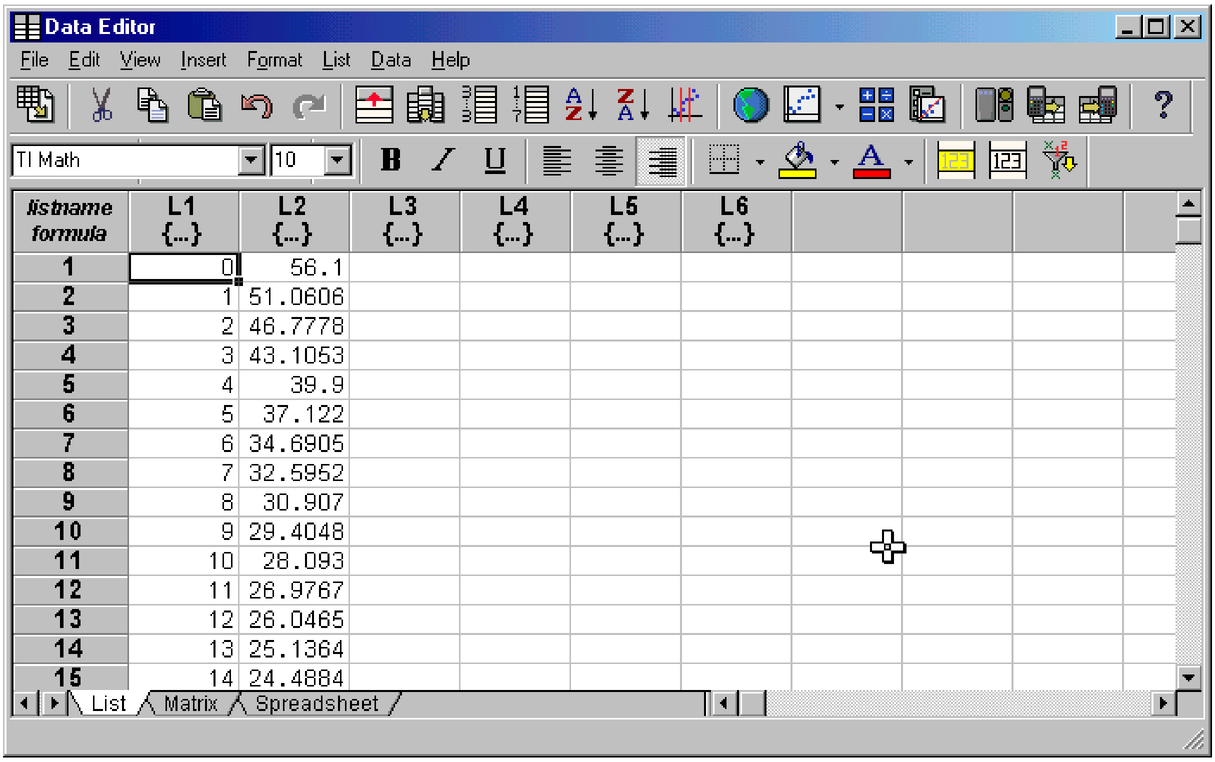
Tipp: Informiere Dich über Updates von TI Connect und über allfällige Updates des Betriebssystems Deines Rechners.

TI InterActive!™

Daten aus dem TI-83+ lassen sich direkt in TI InterActive öffnen. Mit diesem Programm lassen sich die Daten dann weiter bearbeiten, wie z.B. Regressionsrechnung, verschiedene grafische Darstellungen erzeugen usw.

Liste öffnen und bearbeiten

Öffne die List-Registerkarte (links unten) um Liste erzeugen, öffnen oder importieren zu können. Auf dem Bildschirm öffnet sich der *Data Editor*.

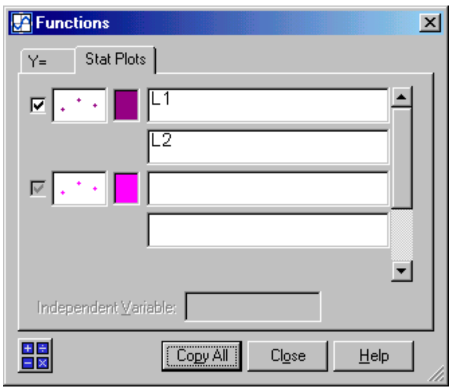


Nun können Listen neu eingegeben oder von, im Computer, bzw. TI-83+ vorliegenden Dateien importiert werden. Dazu klickt man im FILE-Menu auf die Option IMPORT.

Sobald alle gewünschten Listen geöffnet sind, werden sie über in den Dateneditor kopiert.

Statistikdiagramme und Funktionsgraphen

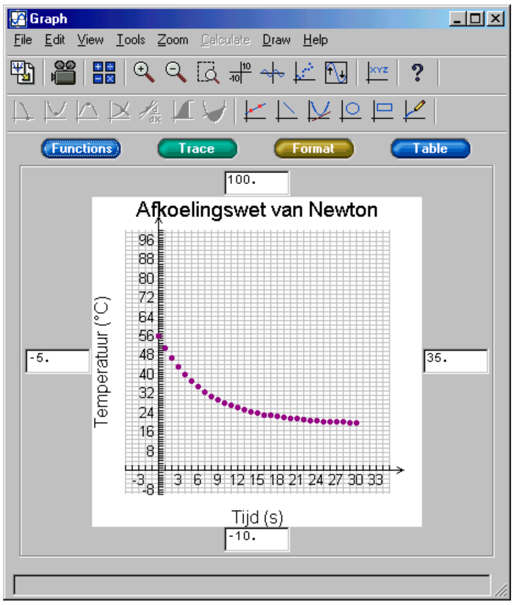
Auf Grundlage der importierten Daten lassen sich statistische Diagramme erzeugen..



Klicke auf das Icon für das Zeichenwerkzeug .

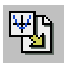
Die abgebildete Eingabemaske erscheint. Sie ist nun entsprechend den Daten auszufüllen. Durch Anklicken der Markierungsart und der Farbe können diese Einstellungen angepasst werden.

Bestätige mit dem Schaltknopf COPY ALL.



Man erhält ein Resultat wie das nebenstehende Bild. Der Bereich von x- und y-Achse kann angepasst werden, so dass die erwünschte Darstellung erreicht wird.

Klicke auf FORMAT um die Achsen richtig zu benennen und skalieren.

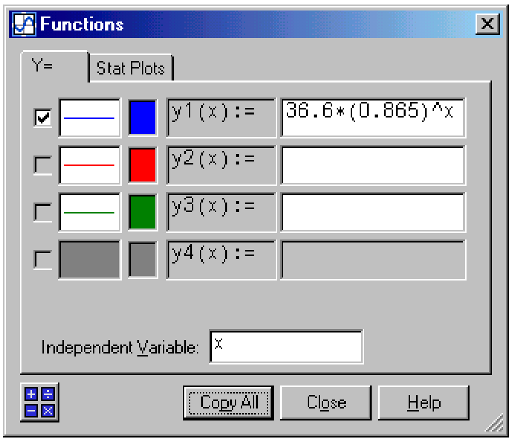


Mit dem abgebildeten Icon überträgt man die Grafik auf das Arbeitsblatt..

Das Experiment wurde bei einer Umgebungstemperatur von 19,5°C ausgeführt. Für den Abschnitt auf der y-Achse entnimmt man der Tabelle den Wert von 56,1°C. Daraus folgt, dass

$$f(x) = 36{,}6\, q^x + 19{,}5.$$

Aus den anderen Messdaten kann man mit einer einfachen Gleichung ableiten, dass q = 0,865.

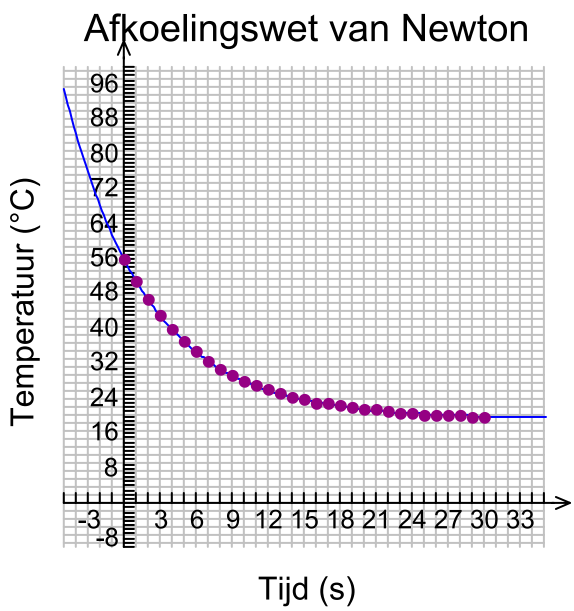


Um einen Funktionsgraphen durch die Messpunkte zu zeichnen, führen wir einen Doppelklick auf die bereits vorhandene Grafik aus.
Auf dem Schirm öffnet sich das nebenstehende Fenster:

Man gibt die entsprechende Funktionsvorschrift ein und bestätigt mit COPY ALL.

Der Funktionsgraph wird gezeichnet und mit  gelangt man wieder zurück ins Arbeitsblatt.

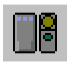
Damit sollte sich die unten abgebildete Figur ergeben. Diese Grafik läßt sich leicht in jede andere Windows-Anwendung (Textverarbeitung, Tabellenkalkulation, ...) kopieren.

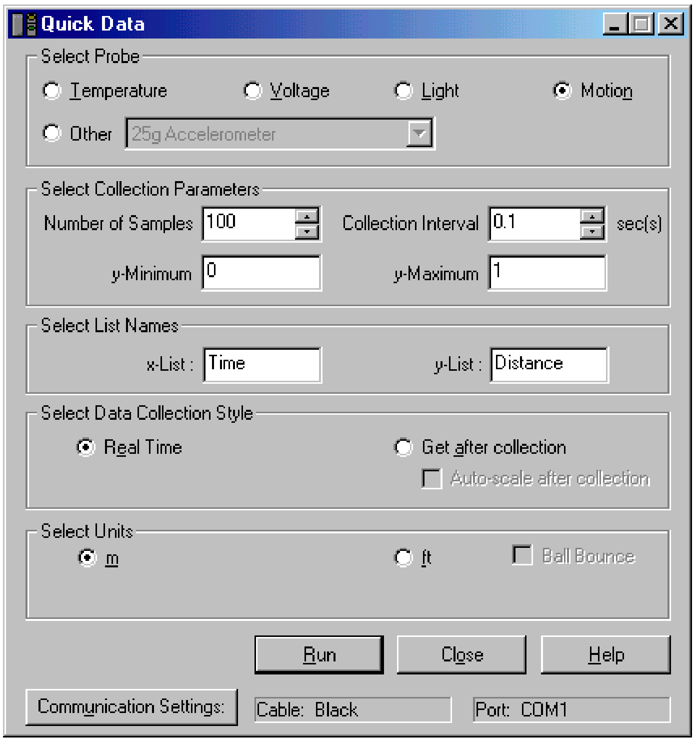


Einsatz von Sensoren

Mit TI InterActive!™ lässt sich auch eine Reihe von Sensoren steuern, so dass jene Messungen, die man i.a. mit einem GTR durchführt auch über den PC durchgeführt werden können.

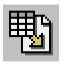
Zu diesem Zweck schließt man den Sensor über das CBL 2™ an den Computer an.

Klicke dann auf  um das QUICK DATA TOOL zu starten, mit dem die Messung durchgeführt wird.

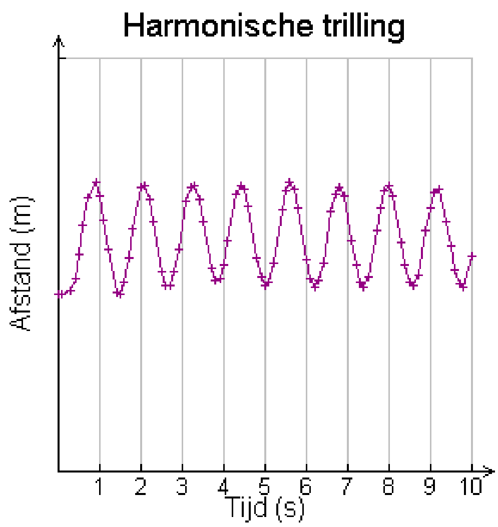


Passe die Einstellungen den Messungen an. Selektiere den richtigen Sensor, die Messintervalle, die Dauer des Experiments und sorge dafür, dass das Experiment beginnen kann. Klicke dann auf RUN um den Messvorgang zu starten. Auf dem Bildschirm kann man den Fortgang der Messung verfolgen.

Passe eventuell die Grafik an und kehre mit  wieder zurück zum Arbeitsblatt

Schließe auch den Data-Editor mit  .

Es könnte sich eine Grafik ergeben wie rechts gezeigt:



Zusammenfassung

Die Programme TI Connect™ und TI InterActive!™ bieten interessante Möglichkeiten, um die Technologie des grafischen Taschenrechners mit dem Einsatz des Computers zu verbinden. Damit können schöne Arbeitsblätter für die Schüler erzeugt werden. Die grafischen Darstellungen sind von einer weit besseren Qualität.

Das Zusammenspiel von GTR und PC erfolgt auf eine schnelle und unkomplizierte Art und Weise. Darüber hinaus können dank dieser Programme Daten vom GTR importiert und in andere typische Computeranwendungen wie Textverarbeitung, Grafikprogramme, Tabellenkalkulation u.a., importiert werden.

Das Programm TI InterActive hat aber neben den hier gezeigten Möglichkeiten noch viel mehr zu bieten. Es kombiniert die Funktionen eines GTR mit den Eigenschaften eines Computeralgebra Systems.

Außerdem gestattet es, interaktive Arbeitsblätter mit einem gefälligen Layout zu gestalten.

Regression, Korrelation und Modellbildung mit dem TI-83 PLUS

Guido Herweyers
KHBO Campus Oostende
guido.herweyers@khbo.be
K.U.Leuven
guido.herweyers@wis.kuleuven.ac.be

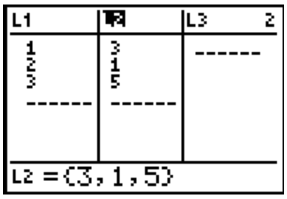
1. Einführendes Beispiel

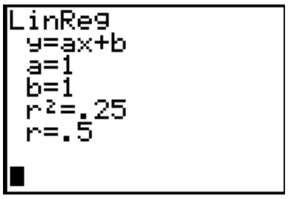
Die Punkte (1,3), (2,1) *und* (3,5) *liegen nicht auf einer Geraden. Wie findet man mit einem TI-83+ jene Gerade, die sich „am besten“ an die vorliegenden Punkte anpasst?*

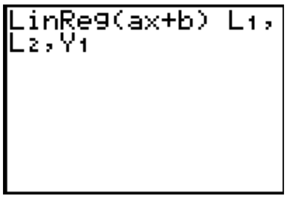
Übertrage zuerst die Koordinaten in die Listen `L1` en `L2` über [STAT] `1:Edit` und rufe dann über [STAT] `<CALC> 4:LinReg(ax+b)` die lineare Regression auf.. Auf dem Bildschirm erscheint `LinReg(ax+b)`.

Ergänze den Befehl mit [2nd] `[L1]` [,] [2nd] `[L2]` [,] [VARS] `<Y-VARS> 1:Function 1:Y1`.
Damit wird die Gleichung der „besten“ Geraden als Funktion `Y1` gespeichert.

Neuerliches Drücken auf [ENTER] liefert diese Gerade $y = x + 1$.



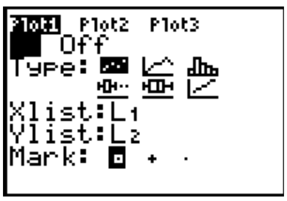


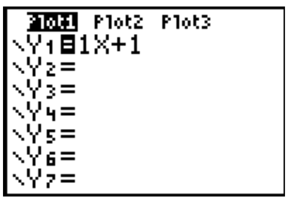


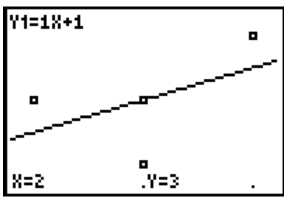
Die zusätzlichen Daten `r` und `r`^{2} erhält man erst nachdem man in [2nd] `[CATALOG] DiagnosticOn` gesetzt hat..

Mit [2nd] `[STAT PLOT] 1:Plot1` wird das erste Diagramm `Plot1` definiert wie unten zu sehen ist.
Über [Y=] können die Funktionen, deren Graphen gezeichnet werden sollen, überprüft werden. Falls der Funktionsterm im Grafikschirm nicht erscheint, muss über [2nd] `[FORMAT]` die Option `ExprOn` aktiviert werden. Über [ZOOM] `9:ZoomStat` erhält man sofort die Grafiken der Punktwolke (hier ein Wölkchen) und der Anpassungs- oder Ausgleichsgeraden.

Über [TRACE] kann man die Gerade durchlaufen und die zu den Argumenten *x* gehörigen Funktionswerte *y* beobachten.

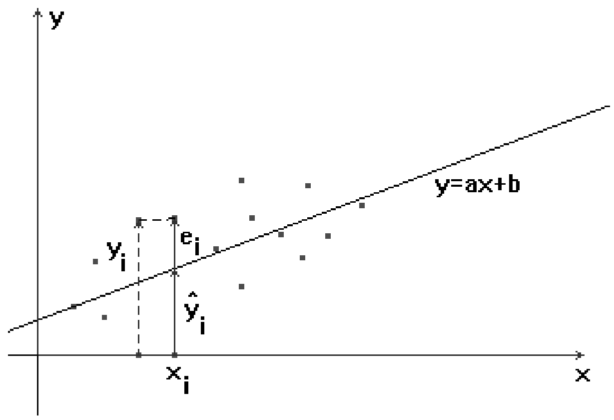






2. Wie findet man die Anpassungsgerade?

Gegeben sei eine *Punktwolke* oder eine *Streudiagramm* von *n* Punkten $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, die einen mehr oder weniger linearen Trend zeigen (siehe Abbildung).



Wir betrachten nun eine Gerade $y = ax + b$ „durch” diese Punkte, mit der wir die *y*-Werte zu gegebenen *x*-Werten vorhersagen wollen.

Den zum Argument x_i gehörigen prognostizierten (theoretischen) Wert von *y* bezeichnen wir mit $\hat{y}_i$. Damit gilt für die Gerade die Gleichung $\hat{y}_i = ax_i + b$.

Für jeden Punkt wird das *Residuum* e_i (die Abweichung, der Fehler) wie folgt berechnet:

$$\boxed{\begin{aligned} e_i &= \textit{beobachteter Wert} - \textit{vorhergesagter Wert} \\ &= y_i - \hat{y}_i \\ &= y_i - (ax_i + b) \end{aligned}}$$

Beachte, dass das Residuum positiv ist, wenn der Punkt über Geraden liegt und dass es negativ ist, wenn der Punkt unterhalb der Geraden liegt.

Um nun die „beste“ Gerade durch die Punktwolke zu legen, wenden wir das *Prinzip der kleinsten Quadrate* an: Bestimme a und b so, dass $\sum_{i=i}^{n} e_i^2$ (die Summe der Residuenquadrate) minimal wird.

Diese beste Gerade nennt man die *Regressionsgerade* oder *lineare Regression* von y bezüglich x, wobei y de abhängige oder *erklärte* Variable und x die unabhängige oder *erklärende* Variable darstellen.

Um die besten Werte für a und b zu bestimmen, verwenden wir die Abweichungen

$$\boxed{u_i = x_i - \overline{x} \text{ und } v_i = y_i - \overline{y}}$$

von den Durchschnittswerten (arithmetische Mittelwerte) $\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ en $\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n}$.

Für die Residuen gilt:

$$\begin{aligned} e_i &= y_i - ax_i - b \\ &= (y_i - \overline{y}) - a(x_i - \overline{x}) - (b - \overline{y} + a\overline{x}) \\ &= v_i - au_i - (b - \overline{y} + a\overline{x}) \end{aligned}$$

Damit wird die Summe der Quadrate der Residuen:

$$\begin{aligned} \sum_{i=i}^{n} e_i^{\,2} &= \sum_{i=1}^{n} ((v_i - au_i) - (b - \overline{y} + a\overline{x}))^2 \\ &= \sum_{i=1}^{n} (v_i - au_i)^2 + \sum_{i=1}^{n} (b - \overline{y} + a\overline{x})^2 \quad \text{(wobei gilt } \sum_{i=1}^{n} u_i = 0 \text{ und } \sum_{i=1}^{n} v_i = 0\text{)} \\ &= \sum_{i=1}^{n} (v_i - au_i)^2 + n(b - \overline{y} + a\overline{x})^2 \end{aligned}$$

Jetzt ist $\sum_{i=i}^{n} e_i^{\,2}$ als Summe von zwei positiven Summanden dargestellt. Diese Summe wird dann minimal, wenn beide Summanden möglichst klein werden.

Zuerst bestimmen wir a so, dass der erste Term minimal wird:

$$\sum_{i=1}^{n} (v_i - au_i)^2 = \sum_{i=1}^{n} u_i^{\,2}.a^2 - 2\sum_{i=1}^{n} u_i v_i.a + \sum_{i=1}^{n} v_i^{\,2}$$

Das ist ein quadratischer Term in a der minimal wird, sobald $a = \dfrac{\sum_{i=1}^{n} u_i v_i}{\sum_{i=1}^{n} u_i^{\,2}}$.

Der zweite Term wird minimal für $b = \overline{y} - a\overline{x}$, denn dann ergibt sich $n(b - \overline{y} + a\overline{x})^2 = 0$.

Zusammenfassung:

Für die „beste" Gerade $y = ax + b$ durch die Punkte $(x_1, y_1), (x_2, y_2), \ldots\ldots (x_n, y_n)$ gilt :

$$\boxed{a = \frac{\sum_{i=1}^{n} u_i v_i}{\sum_{i=1}^{n} u_i^{\,2}} \quad \text{und} \quad b = \overline{y} - a\overline{x}}$$

Wir berechnen nun diese Ausgleichsgerade durch die Punkte (1,3)*,* (2,1) *und* (3,5) *aus dem einführenden Beispiel:*

	x_i	y_i	$u_i = x_i - \overline{x}$	$v_i = y_i - \overline{y}$	$u_i v_i$	$u_i^{\ 2}$
	1	3	-1	0	0	1
	2	1	0	-2	0	0
	3	5	1	2	2	1
$\sum_i$	6	9	0	0	2	2

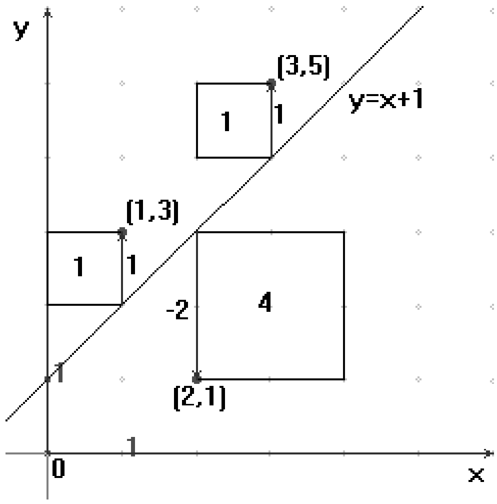
$\overline{x} = 2$	$\overline{y} = 3$

$a = 2/2 = 1$
$b = \overline{y} - a\overline{x} = 1$

Es ist somit klar, dass $y = x + 1$ die beste Gerade - die lineare Regression - darstellt.

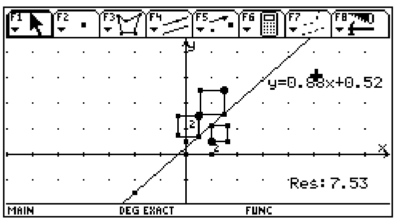
Die Summe der Residuenquadrate, $\sum_{i=1}^{3} e_i^{\ 2} = 6$, ist für diese Gerade minimal.

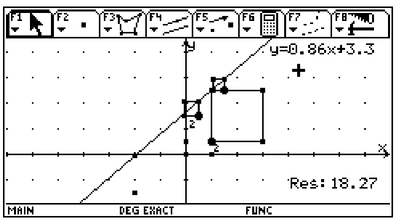
Das lässt eine graphische Interpretation zu:

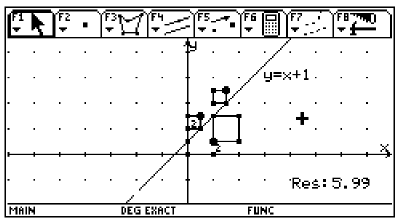


Die Gerade ist so durch die Punkte zu legen, dass die Summe der - vertikalen - Abstandsquadrate minimal wird.

Mit einem dynamischen Geometrieprogramm wie etwa Cabri, kann man die beste Gerade auch durch Probieren finden (lassen). Hier wurde die Cabri-Applikation auf dem Voyage 200 verwendet.

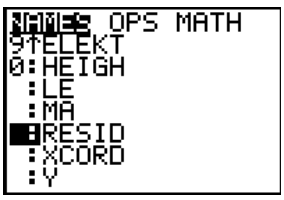


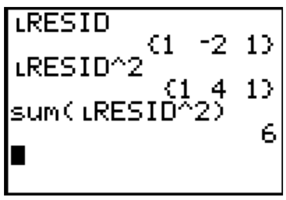




Der **TI-83+** erzeugt bei der Regressionsrechnung automatisch die Liste der Residuen. Diese Liste findet man über $\boxed{\text{2nd}}$ `[LIST] RESID`.

Die Summe der Residuenquadrate berechnet sich dann leicht mit $\boxed{\text{2nd}}$ `[LIST]<MATH> 5:sum(`.



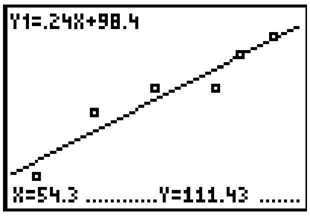


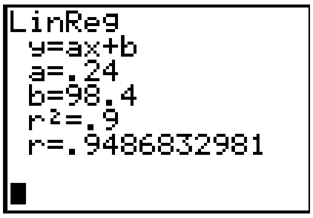
3. Die Korrelation

Die Länge l eines Metallstabes wird bei verschiedenen Temperaturen t gemessen:

t (°C)	20	30	40	50	60
l (mm)	102	107	109	109	113

In der Physik wird der Zusammenhang zwischen l und t durch die lineare Funktion $l = a.t + b$ beschrieben. Wir suchen auf Grundlage der vorliegenden Daten die Regressionsgerade. (Rechne auch per Hand und verwende eine Tabelle nach dem Muster auf der vorigen Seite.) Ersetze t und l durch x und y.



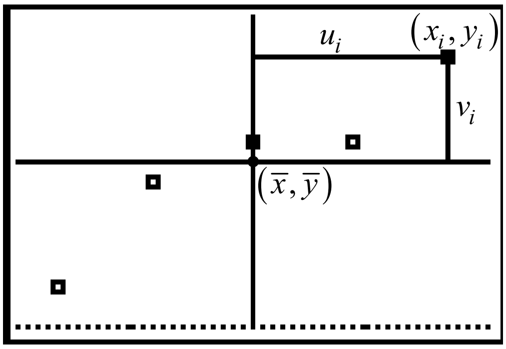


Welche Bedeutung hat nun der *Korrelationskoeffizient r*?

Um dieser Frage nachzugehen zeichnen wir zuerst die beiden Geraden $y = \overline{y}$ (mit $\overline{y} = 108$) und $x = \overline{x}$ (mit $\overline{x} = 40$).

Die achsenparallelen Geraden werden im Rechenfenster über $\boxed{\text{2nd}}$ `[DRAW]4:Vertical 40`, bzw. `3:Horizontal 108` eingezeichnet.

Betrachten wir nun die senkrechte Abweichung $v_i = y_i - \overline{y}$ und die waagrechte Abweichung $u_i = x_i - \overline{x}$ eines beliebigen Punktes (x_i, y_i) vom Schwerpunkt $(\overline{x}, \overline{y})$ der Punktwolke (siehe Abbildung):



Bei der vorliegenden Punktwolke sprechen wir von einer *positiven Korrelation* zwischen den Größen x und y angesichts der Tatsache, dass die meisten Punkte (u_i, v_i) im ersten oder dritten Quadranten bezüglich eines Achsensystems mit $(\overline{x}, \overline{y})$ als Koordinatenursprung liegen. Die Regressionsgerade hat einen positiven Richtungskoeffizienten (Anstieg).

Wir halten fest, dass die Regressionsgerade $y = a \cdot x + b$ immer durch den Schwerpunkt $(\overline{x}, \overline{y})$ der Punktwolke verläuft, da $\overline{y} = a.\overline{x} + b$. Wir können das mit dem TI-83+ sofort kontrollieren, indem man den Schnittpunkt zwischen der Regressionsgeraden und der Geraden $y = \overline{y}$ aufsucht.

Wenn die meisten Punkte im zweiten oder vierten Quadranten liegen, dann spricht man von einer negativen Korrelation. Die Regressionsgerade weist dann einen negativen Anstieg auf.

Sind hingegen die Punkte über alle vier Quadranten verteilt, dann besteht kein *linearer* Zusammenhang zwischen den Größen x und y. Eine Regressionsgerade kann wohl berechnet werden, sie ergibt aber kaum einen Sinn.

Aus unserer Abbildung wird deutlich, dass $\sum_{i=1}^{n} u_i v_i$ angesichts der Beiträge, die die Punkte im ersten und dritten Quadranten liefern, immer positiv ist..

Die Summe $\sum_{i=1}^{n} u_i v_i$ ist allerdings von den für x und y gewählten Einheiten abhängig. Wir kommen zur folgenden Definition des *Korrelationskoeffizienten* r, die unabhängig von den Einheiten für x und y ist:

$$\boxed{r = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\left(\sum_{i=1}^{n} u_i^2\right) \cdot \left(\sum_{i=1}^{n} v_i^2\right)}}}$$

Diese Formel ist symmetrisch in x und y und macht daher keinen Unterschied zwischen der abhängigen und der unabhängigen Variablen.

Wir berechnen r für die vorliegenden Daten mit Hilfe der folgenden Tabelle. Vergleiche mit dem Ergebnis des Rechners.

x_i	y_i	$u_i = x_i - 40$	$v_i = y_i - 108$	$u_i . v_i$	u_i^2	v_i^2
20	102	-20	-6	120	400	36
30	107	-10	-1	10	100	1
40	109	0	1	0	0	1
50	109	10	1	10	100	1
60	113	20	5	100	400	25

$\sum_i$	240	1000	64

$$\boxed{r = \frac{240}{\sqrt{1000 \cdot 64}} \approx 0,9487}$$

Im folgenden wird gezeigt, dass $r^2 \leq 1$ und daher weiters gilt: $-1 \leq r \leq 1$.

Der Korrelationskoeffizient r ist ein Maß für die Qualität des *linearen* Zusammenhangs zwischen den Größen x und y. Je näher der Absolutbetrag von r bei 1 liegt, desto besser ist der lineare Zusammenhang.

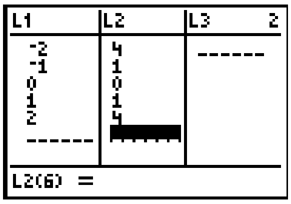
Beachte, dass r und die Steigung a der Regressionsgeraden das Vorzeichen der Summe $\sum_{i=1}^{n} u_i v_i$ haben.

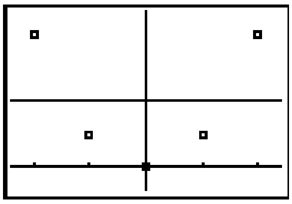
Bei positivem r ist die Korrelation auch positiv, bei negativem r ist die Korrelation negativ.

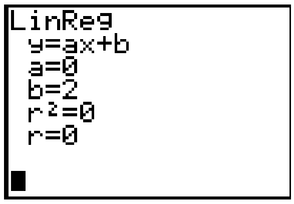
Wenn aber $r = 1$ oder $r = -1$, dann liegen alle Punkte der Punktwolke auf der Regressionsgeraden $y = a \cdot x + b$.

Für Werte von r nahe bei 0 besteht praktisch kein *linearer* Zusammenhang. Es kann aber sehr wohl einen anderen funktionalen Zusammenhang zwischen x und y geben.

Betrachten wir dazu die folgenden Daten, die von Punkten stammen, die auf der Parabel $y = x^2$ liegen. Als Regressionsgerade erhält man $y = 2$. Diese lineare Anpassung ergibt überhaupt keinen Sinn.







Zwischen der Steigung a der Regressionsgeraden und dem Korrelationskoeffizienten r besteht ein rechnerischer Zusammenhang:

$$a = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} = \frac{\sum_{i=1}^{n} u_i v_i}{\sum_{i=1}^{n} u_i^2} \quad \text{und} \quad r = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\left(\sum_{i=1}^{n} u_i^2\right)\cdot\left(\sum_{i=1}^{n} v_i^2\right)}} \; .$$

Daher gilt weiter: $$a = \frac{\sum_{i=1}^{n} u_i v_i}{\sum_{i=1}^{n} u_i^2} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\left(\sum_{i=1}^{n} u_i^2\right)\cdot\left(\sum_{i=1}^{n} v_i^2\right)}} \cdot \frac{\sqrt{\sum_{i=1}^{n} v_i^2}}{\sqrt{\sum_{i=1}^{n} u_i^2}} = r \cdot \frac{\sqrt{\frac{\sum_{i=1}^{n} v_i^2}{n-1}}}{\sqrt{\frac{\sum_{i=1}^{n} u_i^2}{n-1}}} = r \cdot \frac{s_y}{s_x} \, .$$

Dabei ist s_y die Stichprobenstandardabweichung (n-1-Gewichtung) der Daten y_i.
Analoges gilt für s_x.

4. Modellbildung.

4.1 Das Bestimmtheitsmaß

Das *Bestimmtheitsmaß* R^2 ist ein Maß für die Qualität eines Regressionsmodells, das *nicht notwendigerweise linear* ist.

Als Einführung nehmen wir die Datenpunkte von vorhin, (1,3), (2,1) und (3,5), für die wir $y = x + 1$ als Regressionsgerade erhalten haben.

Wir erzeugen die nächste Tabelle mit $\hat{y}_i = x_i + 1$, den nach dem Regressionsmodell theoretischen Funktionswerten.

x_i	y_i	$\hat{y}_i$	$(y_i - \overline{y})^2$	$(y_i - \hat{y}_i)^2$
1	3	2	0	1
2	1	3	4	4
3	5	4	4	1
	$\sum_i$		8	6

Wenn zwischen den Variablen x und y kein Zusammenhang bestehen sollte, dann sollten aufeinander folgende Werte y_i willkürlich um den Mittelwert $\overline{y}$ schwanken.

Für die totale Varianz der Werte y_i bezüglich $\overline{y}$ nehmen wir $\sum_{i=1}^{n}(y_i - \overline{y})^2$ als Maßzahl. Sie gibt an, in welchem Maß die Punktwolke vertikal von der Geraden $y = \overline{y}$ abweicht.

Man sieht, dass das Regressionsmodell besser geeignet ist, die y-Werte vorherzusagen, da die Summe der Residuenquadrate (oder die Varianz der y-Werte bezüglich des Regressionsmodells) $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ minimal ist. Daher folgt, dass $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \le \sum_{i=1}^{n}(y_i - \overline{y})^2$.

Es gilt:

$$\sum_{i=1}^{n}(y_i - \overline{y})^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \left(\sum_{i=1}^{n}(y_i - \overline{y})^2 - \sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)$$

totale Varianz = nicht erklärte Varianz +

(durch die Regression) erklärte Varianz

Das Regressionsmodell ist dann gut, wenn die nicht erklärte Varianz viel kleiner ist als die totale Varianz. Man kann es auch so sagen: wenn die erklärte Varianz nicht viel kleiner ist als die totale Varianz.

Das Bestimmtheitsmaß R^2 ist jener Anteil der Varianz der y_i bezüglich $\overline{y}$, der durch das Regressionsmodell erklärt wird:

$$R^2 = \frac{\text{erklärte Varianz}}{\text{nicht erklärte Varianz}}$$

$$\boxed{R^2 = \frac{\sum_{i=1}^{n}(y_i-\overline{y})^2 - \sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\overline{y})^2} = 1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\overline{y})^2}}$$

Im obigen Beispiel ist $R^2 = 1-\frac{6}{8}=\frac{1}{4}$, d.h., dass nur 25% der Varianz der Daten y_i bezüglich des Mittelwerts $\overline{y}$ durch das Regressionsmodell erklärt werden.

Für den Korrelationskoeffizienten $r = 0,5$ gilt, dass $r^2 = R^2$.

Das ist kein Zufall, denn bei der *linearen* Regression gilt immer, dass $r^2 = R^2$. Daher nennt man r^2 auch das *lineare Bestimmtheitsmaß*.

Gib zu einer linearen Regression immer r^2 als Maß dafür an, wie geeignet das Modell zur Prognose von Funktionswerte y ist.

Wir wollen nun allgemein zeigen, dass $r^2 = R^2$.

Man beginnt mit
$$R^2 = \frac{\sum_{i=1}^{n} v_i^2 - \sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n} v_i^2} \qquad (1)$$

Aus $y_i = ax_i + b + e_i$ und der Substitution $b = \overline{y} - a\overline{x}$

folgt $y_i - \overline{y} = a(x_i - \overline{x}) + e_i$ oder of $v_i = au_i + e_i$,

so dass
$$\sum_{i=1}^{n} v_i^2 = a^2 \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} e_i^2 + 2a\sum_{i=1}^{n} u_i e_i \qquad (2)$$

Der letzte Summand von (2) verschwindet.

$$\sum_{i=1}^{n} u_i e_i = \sum_{i=1}^{n} u_i(v_i - au_i) = \sum_{i=1}^{n} u_i v_i - a\sum_{i=1}^{n} u_i^2$$
$$= \sum_{i=1}^{n} u_i v_i - \frac{\sum_{i=1}^{n} u_i v_i}{\sum_{i=1}^{n} u_i^2} \cdot \sum_{i=1}^{n} u_i^2 = 0$$

Daher gilt
$$\sum_{i=1}^{n} v_i^2 = a^2 \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} e_i^2 \qquad (3):$$

Aus (1) und (3) folgt schließlich $R^2 = \dfrac{a^2 \sum_{i=1}^{n} u_i^2}{\sum_{i=1}^{n} v_i^2} = \dfrac{\left(\sum_{i=1}^{n} u_i v_i\right)^2}{\left(\sum_{i=1}^{n} u_i^2\right) \cdot \left(\sum_{i=1}^{n} v_i^2\right)} = r^2.$ w.z.b.w.

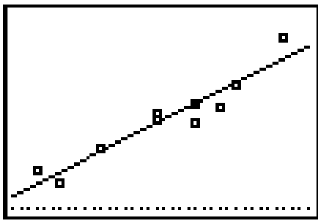
Da $R^2 \le 1$ muss auch gelten $r^2 \le 1$.

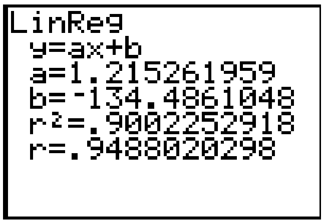
4.2 Der Residuenplot

Als Beispiel betrachten wir die Körpergröße x [cm] und die Masse y [kg] von 10 zufällig gewählten Schülern einer Klasse.:

x_i	163	185	180	175	168	175	191	180	160	183
y_i	60	90	78	81	71	79	104	84	64	83

Wir schreiben die Daten in die Listen `L1` und `L2` und zeichnen das Streudiagramm. Das lineare Modell scheint eine gute Näherung zu bilden.

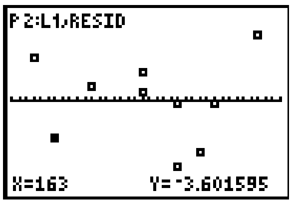


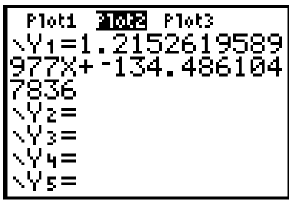


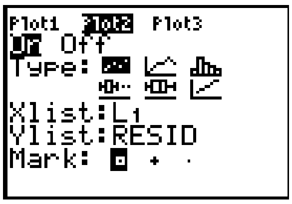
Bei linearer Regression ist die Summe der Residuen immer 0:

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n}\left(y_i - \hat{y}_i\right) = \sum_{i=1}^{n}\left(y_i - a x_i - b\right) = n\left(\overline{y} - a\overline{x} - b\right) = 0$$

Bei einem guten Modell muss der *Residuenplot*, d.h. die Grafik der Residuen in Abhängigkeit von den Daten x_i, eine zufällige Verteilung um die x-Achse aufweisen. Die x-Achse stellt hier die Regressionsgerade dar.





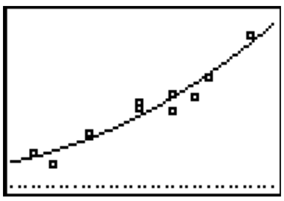


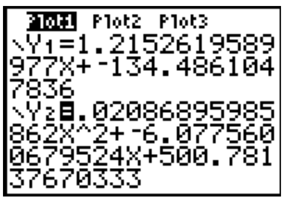
Das Bestimmtheitsmaß r^2 ist 0,9 . Damit werden 90% der Varianz der y_i-Werte gegenüber $\overline{y}$ durch die lineare Regression erklärt.

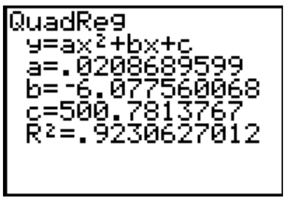
Mit `sum(LRESID^2)` wird die Summe der Quadrate der Residuen ausgegeben: $\sum_{i=1}^{10} e_i^2 = 143{,}7$.

(Überprüfe das).

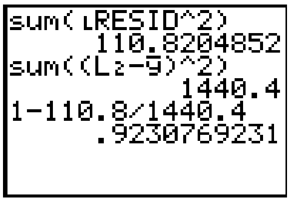
Wir wollen noch schauen, ob eine *quadratische Regression* ein besseres Ergebnis liefert.

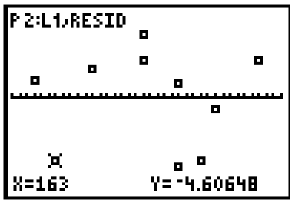






Wir erzeugen wieder den Residuenplot und berechnen die Summe der Residuenquadrate. Eine einfache Rechnung liefert den Wert 0,923 für das Bestimmtheitsmaß R^2. Das quadratische Modell erweist sich damit als geringfügig besser als das lineare.





Wenn wir auch noch alle übrigen Regressionsmodelle versuchen, die der TI-83+ anbietet, lässt sich die folgende Übersicht zusammenstellen:

Modell	R^2	r^2	r
LinReg($ax+b$)		0,900	0,949
QuadReg (ax^2+bx+c)	0,923		
CubicReg (ax^3+bx^2+cx+d)	0,938		
QuartReg ($ax^4+bx^3+cx^2+dx+e$)	0,963		
LnReg ($a+b.\ln(x)$)		0,892	0,945
ExpReg ($a.b^x$)		0,915	0,957
PwrReg ($a.x^b$)		0,911	0,955

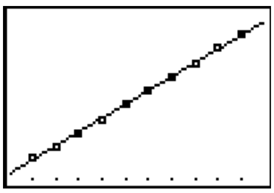
Die *logarithmische Regression* `LnReg` wird auf die lineare Regression der Daten $\left(\ln(x_i), y_i\right)$, die *exponentielle Regression* `ExpReg` auf die lineare Regression der Daten $\left(x_i, \ln(y_i)\right)$ und die *Potenzlinienregression* `PwrReg` auf die lineare Regression der Daten $\left(\ln(x_i), \ln(y_i)\right)$ zurückgeführt.

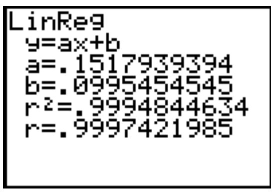
Dass eine lineare Regression verwendet wird, erkennt man an der Anzeige eines Korrelationskoeffizienten und der Ausgabe des Bestimmtheitsmaßes. Erkläre die notwendigen Transformationen der Daten.

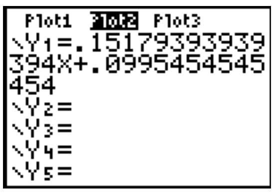
Als letztes Beispiel betrachten wir eine Anzahl regelmäßiger Vielecke mit Seitenlänge 1 und dem Radius des jeweils zugehörigen umschriebenen Kreises.

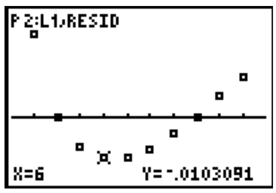
Seiten-l anzahl	3	4	5	6	7	8	9	10	11	12
Radius	0,577	0,707	0,851	1,000	1,152	1,306	1,462	1,618	1,775	1,932

Wir übertragen diese Daten in die Listen `L1` en `L2` und zeichnen das zugehörige Streudiagramm. Das lineare Modell scheint perfekt geeignet zu sein (vergleiche das r^2). Der Residuenplot wirft aber ein ganz anderes Licht auf den Sachverhalt.









Die Residuen zeigen ein deutliches Muster. Das lässt die Existenz eines besseren Modells vermuten. Es sei dem Leser überlassen, dieses besser geeignete Modell zu suchen.

5. Beschreibung des Newtonschen Abkühlungsgesetzes

Nach dem Abkühlungsgesetz von Newton ist die Geschwindigkeit mit der die Temperatur eines Körpers abnimmt proportional zum Unterschied der Körpertemperatur und der Umgebungstemperatur.

Wir führen das folgende Experiment durch und verweisen dazu auf den Workshop von Hans Bekaert „Grafischer Taschenrechner und Computer“ im ersten Teil des Skriptums.

Der Temperaturfühler des CBL 2™ wird in einen Behälter mit warmer Flüssigkeit getaucht und dann wieder herausgenommen, um ihn abkühlen zu lassen. Während der Abkühlungsphase werden Zeit x [sec] und Temperatur y [°C] gemessen. Die Umgebungstemperatur ist die Raumtemperatur, und diese beträgt 19,5°C.

Hier sind die gemessenen Daten:

Zeit	Temperatur	Zeit	Temperatur	Zeit	Temperatur
0	56,1	11	26,9767	21	21,4048
1	51,0606	12	26,0465	22	21,119
2	46,7778	13	25,1364	23	20,8372
3	43,1053	14	24,4884	24	20,6512
4	39,9	15	23,8333	25	20,4651
5	37,122	16	23,1667	26	20,3721
6	34,6905	17	22,7955	27	20,186
7	32,5952	18	22,3409	28	20,093
8	30,907	19	21,9762	29	20
9	29,4048	20	21,5952	30	19,9048
10	28,093				

Aus dem Newtonschen Gesetz folgt (durch Lösung einer Differentialgleichung) das mathematische Modell:

$$y = a \cdot e^{-k \cdot x} + 19{,}5 \qquad \text{wobei } k > 0)$$

Das lässt sich auch in der Form $y = a \cdot b^x + 19{,}5$ mit $b = e^{-k}$ schreiben.

Die Konstanten a und b lassen sich bestimmen, indem man vorgibt, dass der Funktionsgraph durch zwei Messpunkte verlaufen soll, etwa $(0;\ 56{,}1)$ und $(10; 28{,}093)$.

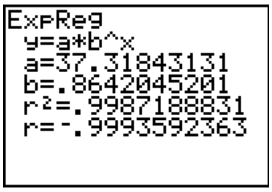
Damit ergibt sich eine Modellfunktion $y = 36{,}6 \cdot 0{,}865^x + 19{,}5$.

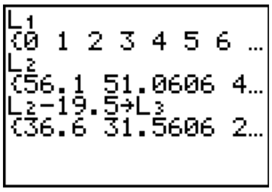
Dieses Modell beruht aber nur auf zwei Messpunkten. Durch eine exponentielle Regression erhalten wir ein (besseres) Modell, das alle vorliegenden Daten berücksichtigt.

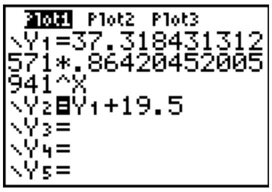
Das Modell kann umgeschrieben werden als $y - 19{,}5 = a \cdot b^x$ oder $Y = a \cdot b^x$ mit $Y = y - 19{,}5$.

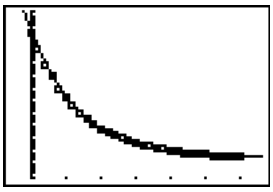
Für die exponentielle Regression müssen wir nun die Daten x_i und $Y_i = y_i - 19{,}5$ verwenden.

Die Werte x_i sind in der Liste `L1` und die Werte y_i in Liste `L2`. Wir ziehen von Liste `L2` den Wert 19,5 ab und erhalten somit in Liste `L3` die Zahlen Y_i.









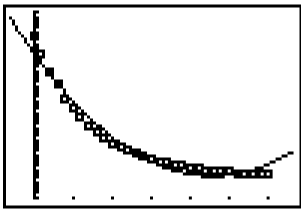
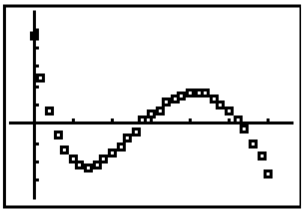
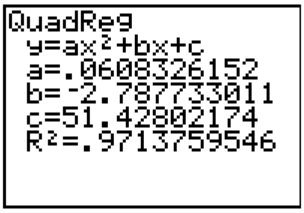
Die exponentielle Regression liefert $a = 37{,}32$ und $b = 0{,}864$.

Übungen:

1) Aus $Y = a \cdot b^x$ folgt $\ln Y = \ln a + x \ln b$ oder $Z = A + B$ mit $Z = \ln Y$, $A = \ln a$, $B = \ln b$.

   Zeige, dass die Ergebnisse der exponentiellen Regression aus einer linearen Regression der Datenpaare $(x_i, Z_i) = (x_i, \ln Y_i)$ abgeleitet werden können; die lineare Regression liefert die Zahlen A und B, woraus $a = e^A$ und $b = e^B$ folgt. Beachte dabei, dass das Bestimmtheitsmaß r^2 von der linearen Regression stammt.

2) Wende auf die ursprünglichen Daten (x_i, y_i) eine quadratische Regression an und erzeuge den Residuenplot. Die Residuen zeigen ein deutliches Muster und weisen damit auf die Existenz eines besseren Modellfunktion hin.

6. Literatur

[1] G. Herweyers, K. Stulens, *Statistiek met een grafisch rekentoestel*, Acco, Leuven, 2000.

[2] D.S.Yates, D.S. Moore, G.P. McCabe, *The Practice of Statistics, TI-83 Graphing Calculator Enhanced*, W.H. Freeman and Company, New York, 1999.

[3] Hackl – Katzenbeisser, *Statistik für Sozial- und Wirtschaftswissenschaften,* Oldenburg, 2000

[4] Markus Paul, *Beschreibende Statistik und explorative Datenanalyse,* T³-Österreich, 2002

Simulation von Zufallsxperimenten
mit dem `TI-83Plus`

Koen Stulens
Limburgs Universitair Centrum
Diepenbeek

1. Einleitung

Der **TI-83 Plus (Silver Edition)** stellt ein hervorragendes Werkzeug dar, um grundsätzliche Ideen über Wahrscheinlichkeiten zu demonstrieren und diese mit der theoretischen Wahrscheinlichkeitsrechnung zu verbinden.

Jacob Bernoullis Werk *Ars Conjectandi* war ein Meilenstein in der Entwicklung der Wahrscheinlichkeitsrechnung. Es wurde erst nach seinem Tod im Jahre 1713 durch seinen Neffen Nikolaus Bernoulli veröffentlicht.

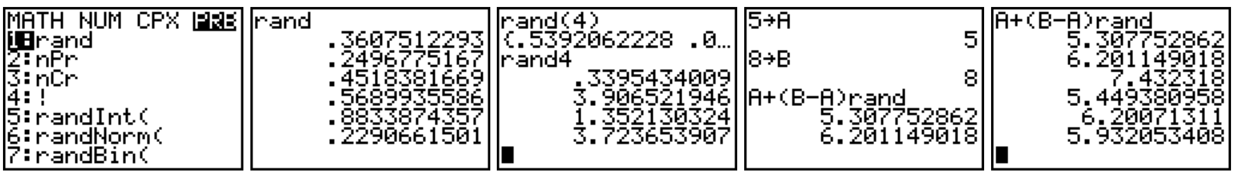
In diesem Buch wurde das erste Mal vom *Gesetz der großen Zahlen* gesprochen: die relative Häufigkeit für das Eintreten eines Ereignisses ist ein *Näherungswert* für die theoretische Wahrscheinlichkeit für dieses Eintreten, wenn man das entsprechende Experiment ausreichend oft durchführt.

Diesem Prinzip wollen wir in den folgenden Simulationen folgen.

2. Zufallszahlen

Mit dem Befehl `rand` ([MATH] `<PRB> 1:rand`) kann man auf die folgenden verschiedenen Arten Zufallszahlen erzeugen:

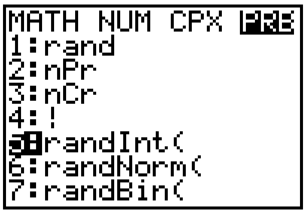
`rand`	→	eine Zufallszahl x zwischen 0 und 1 ($0 < x < 1$)
`rand(4)`	→	eine Liste von 4 Zufallszahlen zwischen 0 und 1
`rand4`	→	eine Zufallszahl x zwischen 0 und 4 ($0 < x < 4$)
`A+(B-A)rand`	→	eine Zufallszahl zwischen A und B ($A < x < B$)

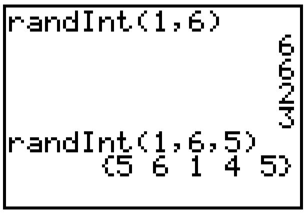


Mit `rand` werden Zufallszahlen generiert, bei denen der Startwert des Zufallsgenerators standardmäßig auf 0 gesetzt ist. Wenn man immer vom gleichen Startwert ausgeht, erhält man immer die gleiche Folge von Zufallszahlen. Um eine andere Folge von Zufallszahlen zu erzeugen, kann man den Startwert für `rand` von außen festlegen. Das funktioniert dann so: `144 ü rand`. Anstelle von 144 kann jede andere Zahl genommen werden.

Die Funktion `randInt` (MATH `<PRB> 5:randInt`) generiert die folgenden Zufallszahlen:

`randInt(1,6)` → eine ganzzahlige Zufallszahl x zwischen 1 und 6 ($1 \le x \le 6$)
`randInt(1,6,5)` → eine Liste von 5 ganzzahligen Zufallszahlen zwischen 1 und 6



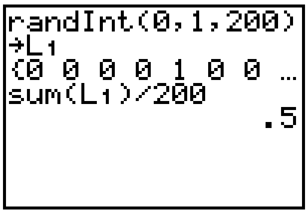


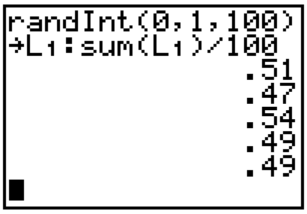
3. Der Wurf einer Münze

Für die Simulation eines Münzwurfs verschlüsseln wir "Kopf" mit **1** und "Zahl" mit **0**.

a. Simuliere das zweihundertmalige Werfen einer Münze: `randInt(0,1,200)` ü L_1.

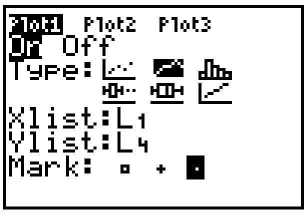
b. Ermittle die absolute und relative Häufigkeit des Ereignisses "Kopf". Die Bilder unten zeigen, dass die relative Häufigkeit für das Eintreten des Ereignisses "Kopf" bei vielen durchgeführten Würfen nahe bei ½ liegt.

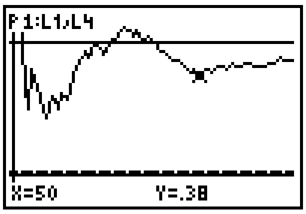


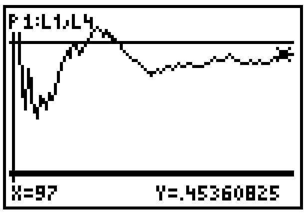


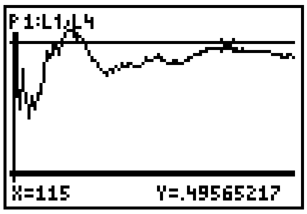
Um dieses Ergebnis zu visualisieren definieren wir die folgenden Listen:

L_1 = `seq(X,X,1,200)`
L_2 = `randInt(0,1,200)`
L_3 = `cumSum(L2)`
L_4 = `L3/L1`
Y_1 = `1/2`







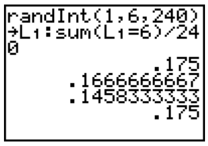


Die abgebildeten Diagramme der kumulierten Häufigkeiten zeigen, dass mehr durchgeführte Experimente nicht automatisch zu einer Verbesserung des Ergebnisses führen. Wenn wir fortfahren, die Münze zu werfen, können wir die relative Häufigkeit von 0,5 so gut approximieren wie wir wollen, aber niemand kann sagen, wie oft wir dafür die Münze werfen müssen.

Klassenauftrag

Jeder Schüler soll 100 Würfe simulieren. Berechne dann die relative Häufigkeit für "Kopf" für die ganze Klasse. Jeder Schüler soll einen zufälligen Startwert für den Zufallsgenerator wählen. Welches Ergebnis kann man erwarten?

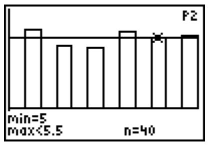
4. Der Wurf mit Würfeln



a. Das 240malige Werfen eines Würfels wird simuliert mit `randInt(1,6,240) ü L1`.

b. Berechne die relative Häufigkeit des Ereignisses “6 Augen”.

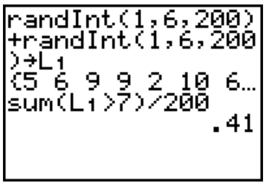
Das abgebildete Histogramm ist wieder eine Visualizierung von Bernoullis Gesetz der großen Zahlen.

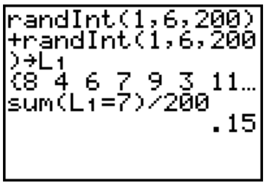


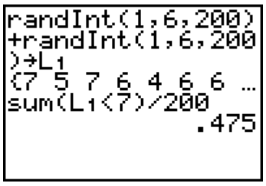
c. Der Aufruf `randInt(1,6)+randInt(1,6)` simuliert die Augensumme für den Wurf mit zwei Würfeln.

Dice 1	Dice 2	Dice 1	Dice 2	Dice 1	Dice 2	Dice 1	Dice 2	Dice 1	Dice 2	Dice 1	Dice 2
1	1	2	1	3	1	4	1	5	1	6	1
1	2	2	2	3	2	4	2	5	2	6	2
1	3	2	3	3	3	4	3	5	3	6	3
1	4	2	4	3	4	4	4	5	4	6	4
1	5	2	5	3	5	4	5	5	5	6	5
1	6	2	6	3	6	4	6	5	6	6	6

Verwende das [2nd][TEST] Menü und führe den gezeigten Versuch durch? Um welche Experimente (Ereignisse) handelt es sich dabei?







5. Aufgabe

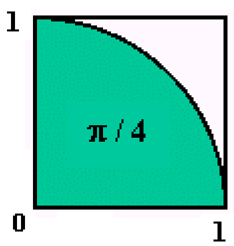
Erzeuge mittels Simulation einen Näherungswert für die Zahl π = 3.14159265358979… (Monte Carlo-Methode).

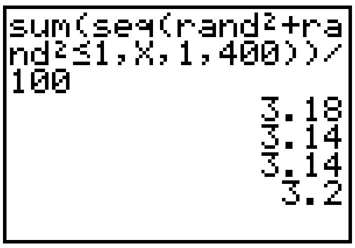
a. Überprüfe unter Einsatz der `rand`-Funktion, ob ein zufälliger Punkt `(rand,rand)` aus dem Intervall [0,1]×[0,1] eine Entfernung von höchstens 1 vom Ursprung aufweist.

b. Führe diesen Test 400 mal durch. Verwende eine Folge mit dem `seq`-Befehl.

c. Wie viele Punkte haben vom Ursprung höchstens den Abstand 1?

d. Unter Verwendung des Ergebnisses von c lässt sich ein Näherungswert für π angeben.





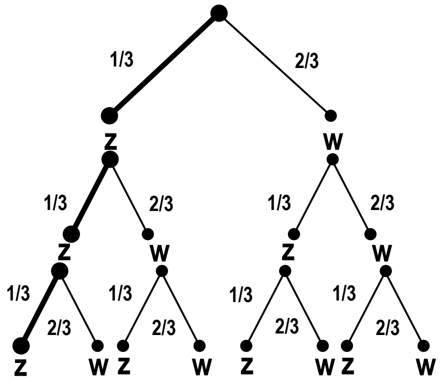
6. Ziehungen

Eine große Anzahl von Problemen der Wahrscheinlichkeitsrechnung lässt sich erklären als Ziehung von Kugeln aus einer Urne. Dabei werden zwei Arten von Ziehungen unterschieden: Ziehung mit Zurücklegen und Ziehung ohne Zurücklegen.

Wir wollen annehmen, dass sich in einer Urne 10 weiße und 5 schwarze Kugeln befinden. Wir stellen uns vor, dass die weißen Kugeln von 1 bis 10 durchnumeriert sind, die schwarzen von 11 bis 15.

Nun wird z.B. die Frage nach der Wahrscheinlichkeit gestellt, drei schwarze Kugeln beim dreimaligen Ziehen einer Kugel zu erhalten.

6.1 Ziehen mit Zurücklegen

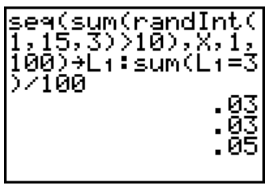


Entscheidungs- oder Wahrscheinlichkeitsbaum

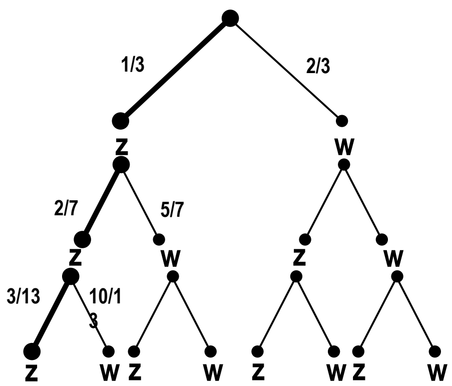
$$P(\,3\,mal\ schwarz) = \frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{3} \approx 0.037$$

Das Ziehen von drei Kugeln kann in diesem Fall einfach durch `randInt(1,15,3)` modelliert werden.

`sum(randInt(1,15,3)>10)` zählt, wie viele von den drei gezogenen Kugeln schwarz sind und legt die Ergebnisse in der Liste `L1` ab. Anschließend werden die Fälle gezählt, bei denen genau drei schwarze Kugeln gezogen worden sind.



6.2 Ziehen ohne Zurücklegen



$$P(\,3\,mal\ schwarz) = \frac{1}{3}\cdot\frac{2}{7}\cdot\frac{3}{13} \approx 0.022$$

Für die Simulation des Ziehens mit Zurücklegen müssen wir die Prozedur von vorhin verfeinern.

Das folgende kurze Programm simuliert das Ziehen ohne Zurücklegen von 3 Zahlen aus einem Vorrat von 15 Zahlen.

```
ClearList L1,L2
Repeat sum(¾List(L2)=0)=0
randInt(1,15,3)üL1
L1üL2
SortA(L2)
End
```

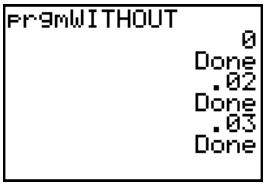
Das Ziehen von drei Kugeln (`randInt(1,15,3)üL1`) wird so oft wiederholt, bis die Bedingung der `Repeat`-Schleife (`sum(¾List(L2)=0)=0`) erfüllt ist. Dann sind alle Elemente der Differenzliste `¾List(L2)=0` gleich Null. Mit anderen Worten gesagt, wenn kein Element von `¾List(L2)` verschwindet, dann sind alle Elemente von `L1` verschieden.

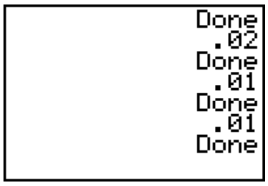
Für die Ausführung der Simulation wird das obige Programm in eine `For`-Schleife eingebaut. Das Programm erhält den Namen `WITHOUT`.

```
ClearList L1,L2,L3
For (I,1,100)

Repeat sum(¾List(L2)=0)=0
randInt(1,15,3)üL1
L1üL2
SortA(L2)
End

sum(L1>10)üL3(I)
End
sum(L3=3)/100üN
Disp N
```





Ein ausführlicheres Programm für die Simulation von Ziehungen finden Sie im Anhang (`DRAWING`).

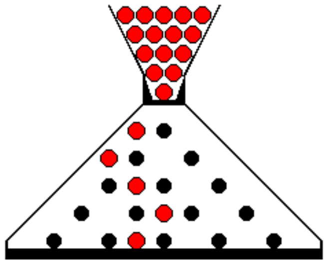
7. Aufgaben

1. Simuliere: Man wirft drei Münzen. Wie groß ist die Wahrscheinlichkeit, mindestens ein mal “Zahl” zu werfen?
2. In einer Packung befinden sich 20 Glühbirnen, von denen 5 kaputt sind. Es werden zufällig 3 Lampen entnommen. Wie groß ist die Wahrscheinlichkeit, dass
   (a) alle drei Lampen schlecht sind?
   (b) die drei Lampen funktionieren?
   (c) genau eine Lampe schlecht ist?

   Ändern sich die Lösungen, wenn eine Lampe nach der anderen aus der Packung genommen wird?

   Führe für alle Experimente die entsprechende Simulation durch.
3. Bei einer Glücksmaschine werden zufällig drei Zahlen zwischen 0 und 99 gezeigt, wobei sich die Zahlen wiederholen können. Man gewinnt, wenn die Summe der Zahlen durch 5 teilbar ist. Ermittle eine Näherung für die Gewinnchance durch eine Simulation.
4. Finde durch Monte-Carlo-Simulation einen Näherungswert für $\int_0^1 e^{-x^2}\,dx$.

8. Das Galtonbrett

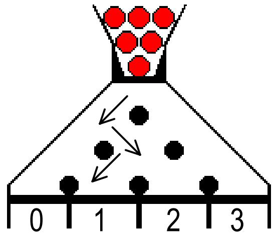


Das Galtonbrett ist eine Vorrichtung mit der man eine Kugel durch Reihen von Nägel rollen lassen kann. Die Nägel sind nach nebenstehendem Muster angebracht.

Sobald die Kugel auf einen Nagel trifft, ist die Wahrscheinlichkeit gleich, nach links oder nach rechts zu fallen Nachdem die Kugel alle Nagelreihen passiert hat, landet sie in einem der am Ende des Bretts angebrachten Fächer.

Wir stellen uns zuerst die Frage, auf wieviele verschiedene Arten eine Kugel den Weg in ein bestimmtes Fach finden kann.

a. Betrachten wir ein Galtonbrett mit drei Nagelreihen.



Auf wie viele Arten kann jedes der Zielfächer erreicht werden?
(der hier gezeigte Weg kan mit LRL beschrieben werden)

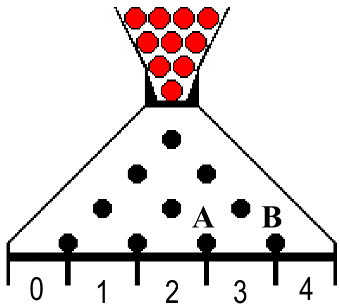
Fach *x*	0	1	2	3
Mögliche Wege nach *x*				

Um im Fach *x* zu landen muss die Kugel auf ihrem Weg ins Ziel genau *x* mal nach rechts fallen. Damit ist die Zahl aller möglichen Wege die Anzahl der Möglichkeiten sich bei 3 Nagelreihen *x* mal für R(echts) zu entscheiden. Diese Zahl wird mit $\binom{3}{x}$ berechnet.

Da die Wahrscheinlichkeit für jede Wegwahl gleich ist, ist nach dem Laplace-Modell die Wahrscheinlichkeit, dass eine Kugel im Fach *x* landet $\frac{\#\text{ Wege nach } x}{\#\text{ Wege gesamt}}$.

Welche Bedeutung hat das nun für ein Galtonbrett mit drei Nagelreihen?

b. Wir fügen eine weitere Nagelreihe hinzu.



Wie viele Wege führen ins Fach **3**?

$\rightarrow$ Anzahl Wege zum Punkt **A** ……

$\rightarrow$ Anzahl Wege zum Punkt **B** ……

$\rightarrow$ Anzahl Wege ins Fach **3** ……

Fach *x*	0	1	2	3	4
Mögliche Wege nach *x*					

Wie groß ist nun die Wahrscheinlichkeit, dass die Kugel im Fach *x* landet?

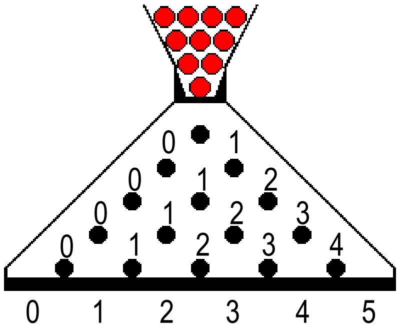
c. Verallgemeinerung

Wie groß ist bei einem Galtonbrett mit *n* Nagelreihen die Wahrscheinlichkeit im Fach *x* zu landen?

d. **Die Simulation des Galtonbretts**

Im folgenden wird ein Galtonbrett mit 5 Nagelreihen simuliert.

Die Entscheidung für “rechts” wird mit 1 kodiert, der Weg nach “links” mit 0. Das nächste kurze Programm modelliert den Weg einer Kugel.

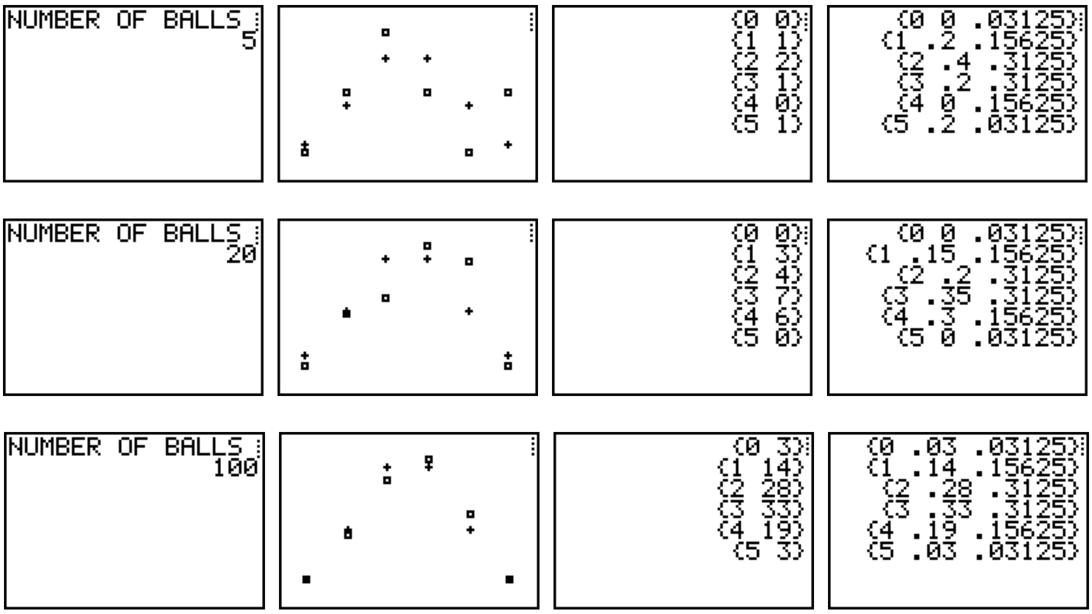



```
0üP
For(J,1,5)
P+randInt(0,1)üP
End
```

Wenn wir mehrere Kugeln rollen oder fallen lassen wollen, dann müssen wir dieses kurze Programm wieder in eine Schleife einbauen.

(Siehe im Anhang das Programm `GALTON`.)

2:GRAPH 3:FREQUENCY 4:REL FREQUENCY



Anhang : TI-83 Plus Programme

a. Galton's pinball machine

```
ClrList L1,L2,L3,L4,L5
Input "NUMBER OF ROWS   ",R
Input "NUMBER OF BALLS  ",N
seq(X,X,0,R)üL1
For(I,1,R+1)
0üL2(I)
End
NüM
Lbl H
For(I,1,N)
0üP
For(J,1,R)
P+randInt(0,1)üP
End
PüL5(I):L2(P+1)+1üL2(P+1)
End
L2/MüL3
binompdf(R,.5)üL4
Lbl G
ClrHome
Menu("GALTON","EXTRA BALLS",A,"GRAPH",B,"FREQUENCY",C,"REL FRE-
QUENCY",D,"TOTAL OF BALLS",E,"END",F)
Lbl A
Input "NUMBER OF BALLS  ",N
ClrList L5
M+NüM
Goto H
Lbl B
PlotsOff
FnOff
Plot1(Scatter,L1,L3,Ð):Plot2(Scatter,L1,L4,Ñ)
AxesOff:ZoomStat:DispGraph
Pause
Goto G
Lbl C
For(K,1,R+1)
Disp {L1(K),L2(K)}
Pause
End
Goto G
Lbl D
For(K,1,R+1)
Disp {L1(K),L3(K),L4(K)}
Pause
End
Goto G
Lbl E
Disp "NUMBER OF BALLS",M
Pause
Goto G
Lbl F
Stop
```

b. Drawing with or without replacement

```
Menu("REPLACEMENT","WITHOUT",A,"WITH",B)
Lbl A
Disp "DRAW"
Input R
Disp "OUT OF"
Input N
Repeat sum(¾List(L2)=0)=0
randInt(1,N,R)üL1
L1üL2
SortA(L2)
End
Disp L1
Stop
Lbl B
Disp "DRAW"
Input R
Disp "OUT OF"
Input N
randInt(1,N,R)üL1
Disp L1
Stop
```

Literatur

a. *Handbuch TI-83*, Texas Instruments (education.ti.com/guides), 1996

b. *Statistiek met een grafisch rekentoestel*, Guido Herweyers – Koen Stulens, Acco Leuven, 2000

c. *The Practice of Statistics, TI-83 Graphing Calculator Enhanced*, D.S.Yates, D.S. Moore, G.P. McCabe, W.H. Freeman and Company, New York, 1999.

Ergänzungen zum Beitrag von Koen Stulens

Josef Böhm

Das folgende Problem eignet sich als Einstiegsaufgabe in die Wahrscheinlichkeitsrechnung und Beschreibende Statistik.

Ein Würfelspiel

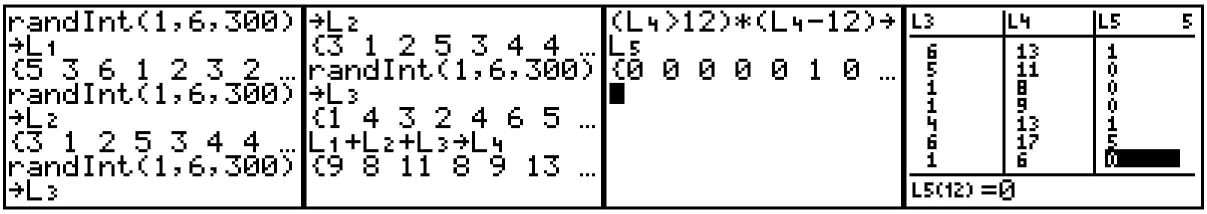
Es wird mit drei Würfeln geworfen. Jede Augensumme über 12 gewinnt, und zwar:

13 Augen bringen den Gewinn von 1 €,
14 Augen bringen 2 €,
....
18 Augen bringen den Höchstgewinn von 6€.

Welchen Einsatz wären Sie bereit für ein Spiel zu geben?

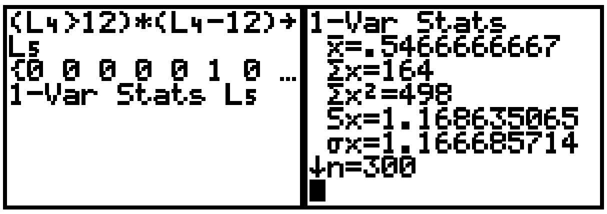
Welchen Einsatz muss der Bankhalter verlangen, um auf lange Sicht nichts zu verlieren?

Zuerst soll eine Simulation eine ungefähre Anwort geben. Die Berechnung des Erwartungswertes gibt dann das "exakte" Ergebnis. Verschiedene statistische Ausarbeitungen können folgen (Varianz, andere Kennzahlen, diverse Diagramme). Tipp: Leere vorerst alle Listen, sonst kann es zu einem Memory-Full-Error kommen.



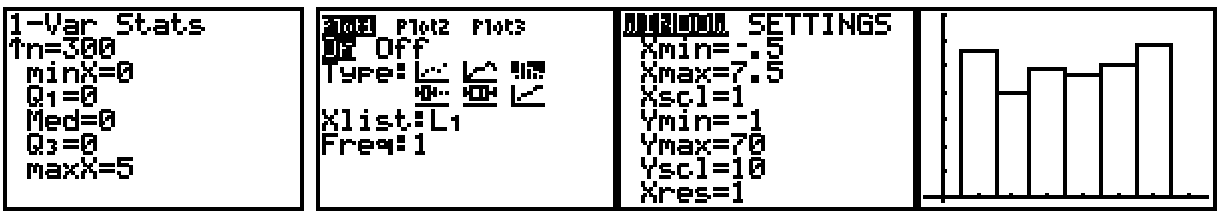
Für jeden Würfel wird eine Liste von 300 Würfen erzeugt. In Liste `L4` wird die Augensumme gebildet. Die Gewinnliste `L4` verdient etwas mehr Aufmerksamkeit: Die Relation `L4 > 12` hat den Wert logischen Wert 1, wenn sie wahr ist und den Wert 0 wenn sie wahr ist. Für alle Summen über 12 wird daher der Gewinn als Augensumme – 12 gebildet und in die Liste geschrieben. Ein Blick in den Listeneditor überzeugt uns von der Richtigkeit unseres Vorgehens.

Über STAT <CALC> wird der Befehl `1-Var Stats` aufgerufen und auf die Liste der Gewinne angewendet, denn wir wollen natürlich sofort die Antwort auf unsere Frage erhalten:

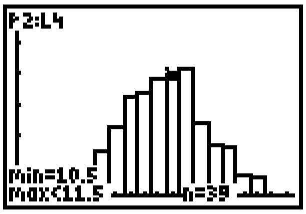


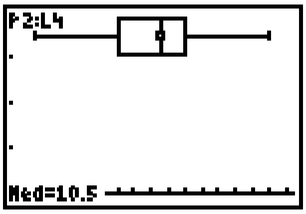
Der Mittelwert liegt bei 0,547 und die Standardabweichung hat die Wert 1,1686 (n-1-Gewichtung) und 1,1667 (bei n-Gewichtung). Außerdem werden der Median und die Quartile angegeben.

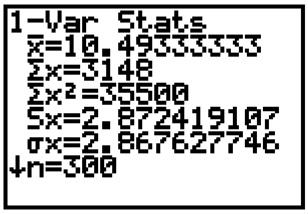
Bei einer geeigneten Einstellung der WINDOW-Werte liefert die Liste der Würfe des ersten Würfels ein schönes Histogramm für eine angenäherte Gleichverteilung:



Die nächsten beiden Plots sind Histogramm und Kastendiagramm (BoxPlot) der Augensummen. Wir haben 39 mal die Augensumme 11 erreicht. Der Median liegt bei 10,5, während der Mittelwert der Stichprobe bei 10,49 liegt.

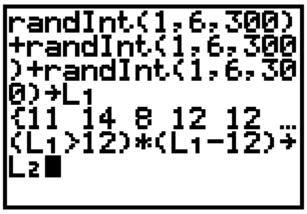


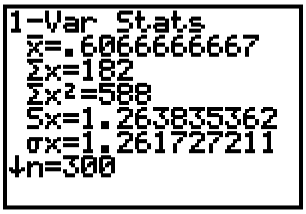


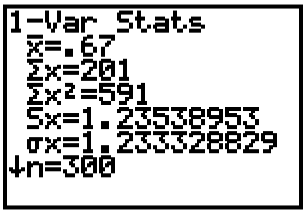


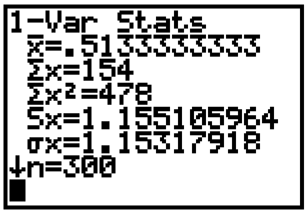
Nun kann man daran denken, die theoretischen Werte für Erwartungswert und Varianz der Zufallsvariablen *Spielgewinn* zu berechnen.

Die Simulation lässt sich bechleunigen, wie unten gezeigt. mit [ENTRY] können Befehle aus der “Vergangenheit” wieder auf den Schirm gebracht werden. Damit erspart man sich die Arbeit des Tippens. Es wurden weitere drei Simulationen durchgeführt.









Dabei ergeben sich die Mittelwerte 0,61, 0,67 und 0,51.

Eine Übung für die Schüler wäre, die Wahrscheinlichkeiten für die Gewinne von $X = 0, 1, 2, 3, 4, 5, 6$ durch die entsprechenden relativen Häufigkeiten aus der Simulation näherungsweise zu finden – und dann exakt nachzurechnen.

Lösung:

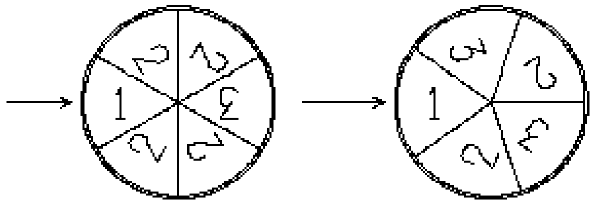
Der Erwartungswert $E(X) = \frac{126}{216} \approx$ 0,583, die n-gewichtete Standardabweichung ist $\frac{\sqrt{203}}{12} \approx 1{,}1873$.

Die Wahrscheinlichkeiten für die Gewinne 0, 1, 2, 3, 4, 5, 6 sind $\frac{90}{216}, \frac{21}{216}, \frac{15}{216}, \frac{10}{216}, \frac{6}{216}, \frac{3}{216}, \frac{1}{216}$.

Mit der nächsten Aufgaben sollen zusammengesetzte und bedingte Wahrscheinlichkeiten simuliert und berechnet werden.

Auf nach Las Vegas mit dem TI-83+

oder nur nach Baden bei Wien, wenn das Geld nicht reicht.



Eine Glücksmaschine – Slot Machine – besteht aus zwei von einander unabhängig rotierenden Scheiben. Auf beiden Scheiben sind Zahlen auf jeweils gleich großen Sektoren zu lesen. Die Scheiben bleiben zufällig so stehen, dass die Pfeile auf Ziffern zeigen.

Simuliere eine größere Anzahl von Spielen und bestimme daraus Näherungswerte für die Wahrscheinlichkeit des Eintreffens der folgenden Zufallsereignisse. Versuche dann diese Ergebnisse durch Mittel der Wahrscheinlichkeitsrechnung zu bestätigen. (Du kannst auch umgekehrt vorgehen, d.h. zuerst rechnen und dann simulieren.)

a) beide Pfeile zeigen auf **2**,

b) kein Pfeil zeigt auf **1**,

c) beide Pfeile zeigen auf gleiche Ziffern,

d) die Summe der beiden Ziffern ist 4,

e) mindestens ein Pfeil zeigt auf **3**,

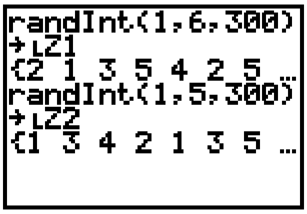
f) dreimal hintereinander zeigen beide Pfeile auf verschiedene Ziffern,

g) der linke Pfeil zeigt auf **1**, wenn die Summe beider Ziffern 2 ist,

h) die Summe der beiden Ziffern ist 4, wenn der linke Pfeil auf **2** zeigt.

Wir wollen wieder n = 300 Versuche durchführen. Der TI-83+ stößt dann bald an seine Speicherkapazitäten.

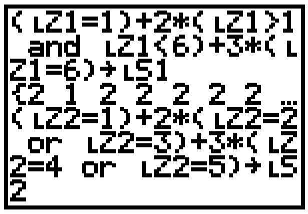
Die folgende Vorgangsweise kann gewählt werden:

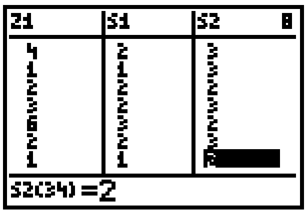


Wir erzeugen zwei Listen Z1 und Z2 mit gleich verteilten ganzen Zufallszahlen von 1-6, bzw. von 1-5. Sie stellen die Basis für unsere Untersuchung dar.

(Das Listensymbol ʟ wird über `[LIST] <OPS> B` erzeugt.)

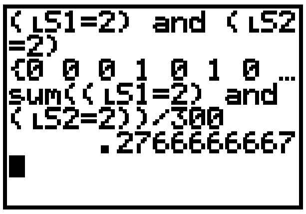
Die Ziffern der ersten Scheibe ergeben sich so: {1} → 1, {2,3,4,5} → 2 und {6} → 3. Bei der zweiten Scheibe ähnlich: {1} → 1, {2 ,3} → 2 und {4, 5} → 3. Das muss nun entsprechend kodiert werden und ergibt die Listen S1 und S2.





Im Listeneditor können wir uns davon überzeugen, dass unsere Prozedur richtig war. Im weiteren Verlauf der Aufgabe werden nur mehr die Listen S1 und S2 benötigt. Der TI-83+ speichert nämlich (leider) nicht dynamisch die Entstehung von S1 und S2 – d.h., bei neuen Listen Z1 und Z2 werden S1 und S2 nicht automatisch angepasst.

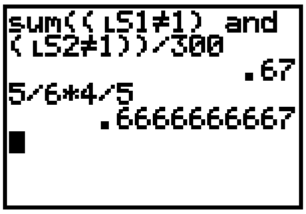
a) Wir beschreiben das Ereignis, dass beide Pfeile auf **2** zeigen .....



...... und berechnen die Wahrscheinlichkeit dafür:

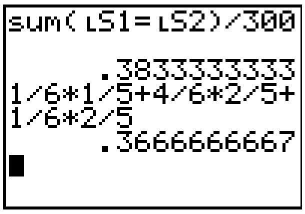
$$\frac{4}{6} \times \frac{2}{5} = \frac{4}{15} \approx 26{,}67\%$$

b) Kein Pfeil soll auf **1** zeigen .....

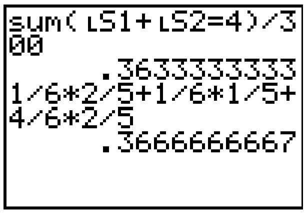


......Simulation und Ergebnis.

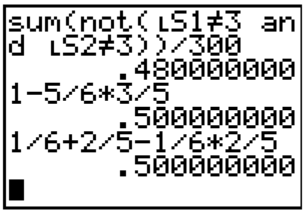
c) Beide Pfeile zeigen auf die gleichen Ziffern .....



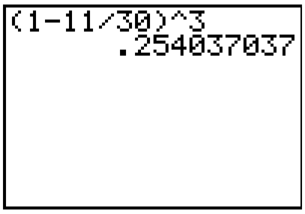
d) Die Summe der beiden Ziffern ist 4 .....



e) Mindestens 1 Pfeil zeigt auf **3** .....



f) Dreimal hintereinander zeigen beide Pfeile auf verschiedene Ziffern

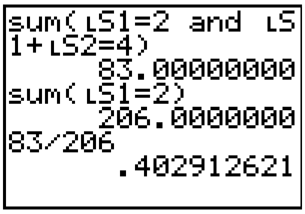


Über die Gegenwahrscheinlichkeit von Aufgabe c). Die - längere - Simulation wird unten gezeigt.

g) Links zeigt auf **1**, wenn die Summe **2** ist

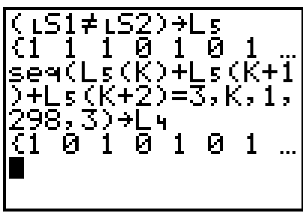
Hier sollte keine Simulation notwendig sein, um zu erkennen, dass dies ein sicheres Ereignis darstellt. Die Antwort muss daher 1 oder 100% lauten.

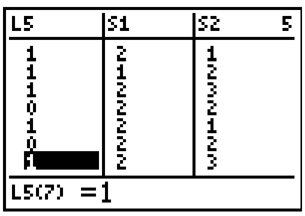
h) Die Summe ist **4**, wenn links **2** auftritt

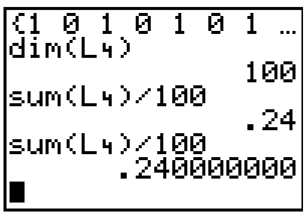


$$\frac{\frac{4}{6}\cdot\frac{2}{5}}{\frac{4}{6}} = \frac{2}{5} = 40\%$$

zu f) In L5 haben alle Versuche mit ungleichen Ziffern den Wert 1. L5 wird in Dreiergruppen zerlegt und die Summe der Elemente gebildet. Falls die Summe 3 ergibt, dann ist das gewünschte Ereignis eingetroffen.

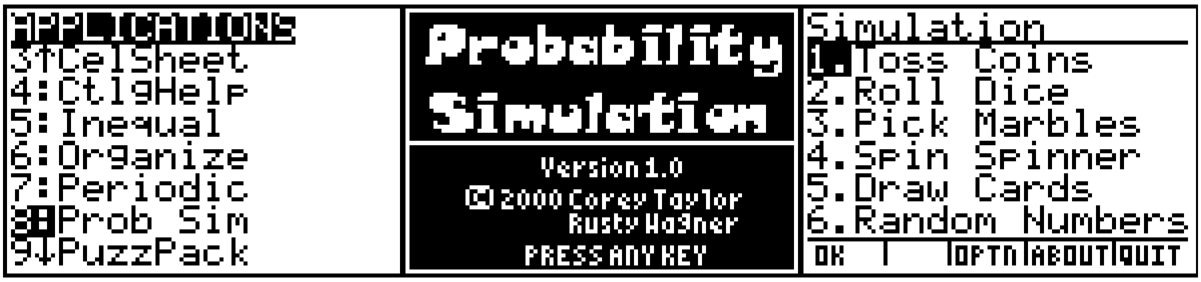






Hinweis auf eine nette TI-83+ Applikation

Für den TI-83+ steht eine recht brauchbare Applikation zur Simulation von Zufallsexperimenten zur Verfügung. Sie kann eine sinnvolle Ergänzung zu den gezeigten Beispielen sein. Für den Unterricht wertvoller ist sicher die mit den Schülern erzeugte Modellierung der Experimente. Zusätzliche spielerische Beispiele lockern aber den Unterricht auf.



Sowohl das Programm als auch die kurze Dokumentation sind nur in englischer Sparche verfügbar und können von der TI-Homepage heruntergeladen werden.

Die folgenden Simulationen lassen sich durchführen:

- Der Wurf mit bis zu 3 Münzen, wobei die Seiten der Münzen gewichtet werden können.
- Der Wurf mit bis zu 3 Würfeln, die bis zu 10 “Seitenflächen” mit unterschiedlicher Gewichtung aufweisen können.
- Das Ziehen von Kugeln (maximal 5 verschiedene Farben) mit oder ohne Zurücklegen. Die Anzahl der Kugeln von jeder Art kann frei festgelegt werden.
- Das Drehen eines Glücksrads mit bis zu 8 Sektoren, die unterschiedlich gewichtet werden können.
- Das Ziehen von Karten (32 Blatt oder 52 Blatt) mit und ohne Zurücklegen.
- Die Erzeugung von Serien von ganzen Zufallszahlen (1 – 6) aus einem zu definierenden Bereich, mit und ohne Wiederholung.

Bei einigen Experimenten werden Häufigkeitsdiagramme ausgegeben.

Alle Daten können in Listen exportiert und dann weiterbearbeitet werden.

Einführung in das Programmieren
mit einemTI-83 Plus

Regis Ockerman

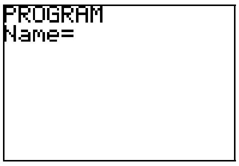
1. Programm 1

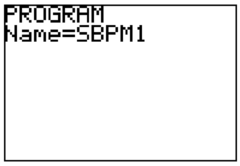
Ein Programm ist dann besonders nützlich, wenn man immer wieder den gleichen Prozess, aber mit anderen Daten ablaufen lassen muss.

In diesem Workshop beginnen wir mit einem einfachen Beispiel: mit der Berechnung des „Body Mass Index“, BMI.. Die Formel für den BMI ist (Gewicht in Kilogram) /(Körpergröße in Meter)^2 .

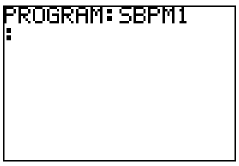
Man benötigt immer die gleiche Formel, aber für jede Person gelten andere Daten.

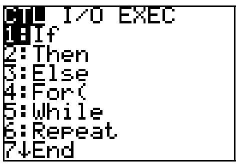
Um ein Programm zu erzeugen, gehen wir nun vor wie folgt: Nach dem Einschalten der Maschine drücken wir auf die Taste PRGM. Falls noch keine Programme gespeichert sind, sieht das so aus:

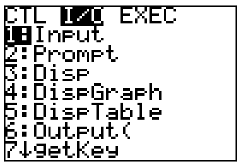




Wir wählen <NEW>, bestätigen mit ENTER und erhalten einen neuen Schirm. Nach der Bestätigung mit ENTER werden wir um den Namen des neuen Programms gefragt. `[A-LOCK]` ist bereits aktiviert. Damit kann der Programmname eingegeben werden, etwa `SBPM1`.
Für den Einser muss `[A-LOCK]` wieder ausgeschaltet werden. Damit ergeben sich die Bildschirme wie unten ganz links abgebildet. Jetzt können die Programmzeilen geschrieben werden.





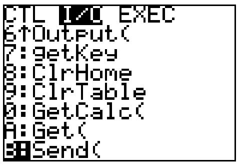


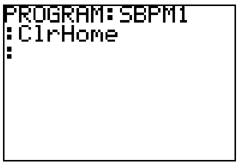
Über `[QUIT]` kann jederzeit das Programmierfenster verlassen werden und man findet sich wieder im Hauptbildschirm (Rechenfenster, Homescreen). Ein neuerlicher Aufruf von PRGM führt wieder ins Programmierfenster und über `<EDIT>` ENTER sind wir wieder zurück im Editierfenster.

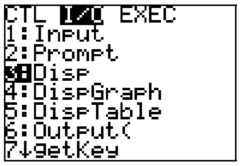
Nun können wir beginnen, Anweisungen (Befehle) zu schreiben.

Es ist immer günstig, die Arbeit mit einem leeren Bildschirm zu beginnen.

Wir drücken nochmals auf PRGM und es erscheint eine neue Auswahl, in der wir das Untermenü <I/O> wählen. Der Pfeil bei 7↓ weist darauf hin, dass es noch weitere Optionen gibt. Hier geht es weiter bis zum Buchstaben B.







Wir brauchen die Anweisung [ClrHome], daher gehen wir entweder mit dem Cursor bis zu diesem Befehl und drücken ENTER oder wir geben die entsprechende Nummer - hier 8 - ein. Damit wird die Anweisung ins Programm übertragen. Mit einem nachfolgenden ENTER wird die nächste Programmzeile mit dem Doppelpunkt [:] eingeleitet.

Unser Programm soll mit einem Titel beginnen. Dazu benötigen wir wieder das <I/O]> - Untermenü und wählen den Befehl [Disp] aus.

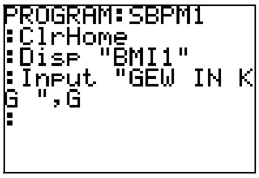
Damit lässt sich Text oder der Wert einer Variablen auf den Ausgabeschirm bringen.

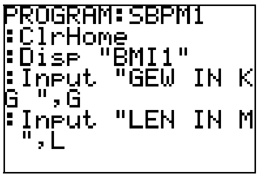
Hier handelt es sich um einen Text und dieser muss zwischen Doppelhochkomma " " geschrieben werden. Dieses Sonderzeichen finden wir am Rechner unter ALPHA +, der Zwischenraum - das Zeichen ␣ - wird über ALPHA 0angesprochen.

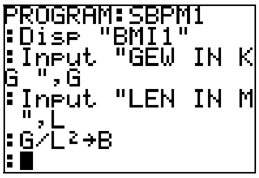
Beachte, dass ausschließlich Großbuchstaben verwendet werden können. Als Titel wählen wir ganz einfach "BMI1".

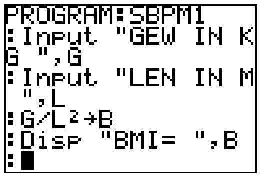
Nun werden die Daten zur Berechnung des BMI abgefragt.

Die Datenabfrage erfolgt durch die Anweisung [Input]. Sie bietet auch die Möglichkeit, eine Eingabeabfrage anzubringen.









Der Text steht zwischen " " und nachher folgt der Name der, der Eingabe zugewiesenen Variablen, getrennt durch ein Komma. Der TI-83 Plus kennt nur einbuchstabige Variable!

Zuerst fragen wir nach dem Gewicht und verwenden die Variablenbezeichnung G.

Auf die gleiche Weise fragen wir um die Körpergröße (Länge) und verwenden die Variable L.

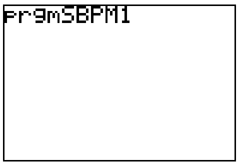
Dann wird der BMI berechnet und unter der Bezeichnung B gespeichert.

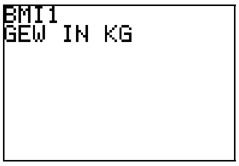
Zum Schluß wird das Ergebnis gemeinsam mit einer kurzen Erklärung mit der Anwesiung [Disp] ausgegeben.

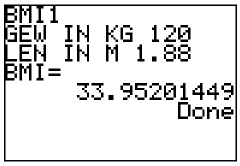
Das =-Zeichen wird aus dem Menü [TEST] <TEST> geholt.

Nun soll unser erstes Programm getestet werden. Der Editor wird mit [QUIT] geschlossen und mit PRGM sehen wir einen uns schon bekannten Schirm.

Wir gehen folgendermaßen vor:







Mit ENTER wird der Programmname auf den Schirm gebracht. Man muss im Hauptbildschirm mit dem Cursor in in einer neuen leeren Zeile stehen, sonst wird der Programmname einfach wo angefügt und man erhält eine Fehlermeldung.
Nochmaliges ENTER lässt das Programm starten. Wir testen für eine Person von 1,88 m Körpergröße und 120 kg Gewicht. Das ergibt die oben abgebildeten Ausgaben.

Wenn wir nach Ablauf des Programms nochmals ENTER drücken, dann beginnt es erneut mit der Abfrage nach den Eingabedaten.

2. Programm 2

Wir sehen, dass ein Programm aus drei Teilen besteht: Eingabe, Verarbeitung und Ausgabe – EVA-Prinzip. In unserem Beispiel war die Verarbeitung sehr kurz.

Hier erfolgte die Eingabe der Daten über `[Input]`. Dafür gibt es auch die `[Prompt]`-Anweisung. Mit ihr kann nur der Wert einer Variablen ohne Begleittext abgefragt werden. Manchmal kombiniert man `[Disp]` mit `[Prompt]`. Um einen Text an eine bestimmte Stelle des Ausgabeschrims zu setzen steht die Anweisung `[Output]` zur Verfügung. Nähres dazu findet man im Handbuch.

Man kann sich fragen, was geschieht, wenn jemand bei Ausführung unseres Programms z.B. ein negatives Gewicht oder eine andere sinnlose Eingabe macht?

Daher stellt sich weiters die Frage, ob man die Eingabe nicht absichern kann. Wir wollen eine Prüfung auf Plausibilität der Eingabe einbauen.

Dies leistet etwa die Anweisung `[Repeat]`.

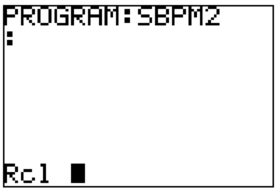
Manche werden sich möglicherweise daran erinnern, dass es in PASCAL eine REPEAT ... UNTIL-Konstruktion gibt. Das bedeutet, dass ein Block von Befehlen solange wiederholt wird, bis eine bestimmte Bedingung erfüllt ist.

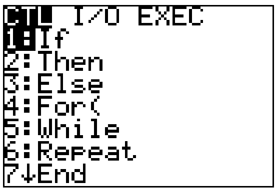
Wir sollten daher die Frage nach der Eingabe des Gewichts so lange wiederholen, bis mit einer positiven Zahl gentwortet wird. In TI-BASIC sieht diese Konstruktion etwas anders aus:

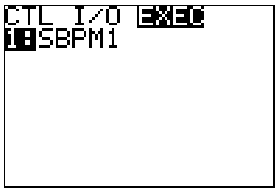
Die entsprechende Syntax lautet: `Repeat` *Bedingung Anweisung(en)* `End.`

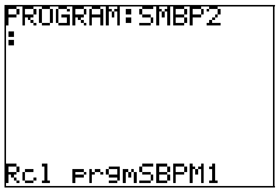
(Eine Übersicht findet sich unter 6.)

Das wird nun in einer verbeserten Programmversion `SBPM2` durchgeführt.







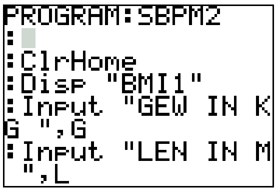


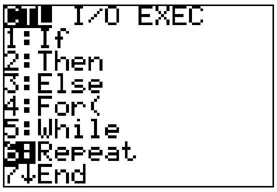
Über die Tastenfolge `[RCL]` [PRGM] erscheint das bekannte Teilmenü (2. Bild).
Wir wählen `<EXEC>` und sehen Bild 3 mit allen vorhandenen Programmen. Mit zweimaligem [ENTER] werden die Programmzeilen übernommen. Diese können nun erweitert und/oder verändert werden. Auf diese Art und Weise ersparen wir uns eine Menge Tipparbeit.
So lässt sich auch ein Programmname verändern!

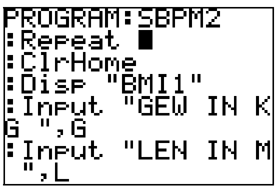
Wir gehen ganz an den Anfang des Programms und fügen mit `[INS]` [ENTER] eine leere Programmzeile ein, um den Beginn der REPEAT-Schleife einzufügen. Dazu holt man die Anweisung `[Repeat]` aus dem `CTL`-Untermenü (`Repeat-End` ist eine Kontrollstruktur, daher ist sie im `CTL`-Untermenü zu finden). Unmittelbar daran schreiben wir die Bedingung, die erfüllt werden muss, bevor mit der Durchführung des Programms fortgefahren wird.

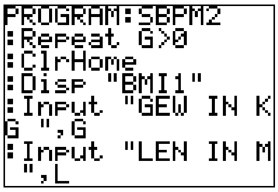
Die Bedingung heißt hier: `G > 0`. Das Relationszeichen > finden wir in `[TEST]`.

Damit sollten sich die folgenden Bildschirme ergeben:



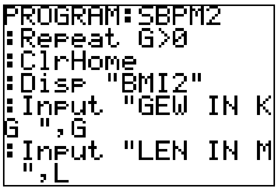


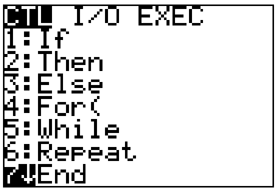


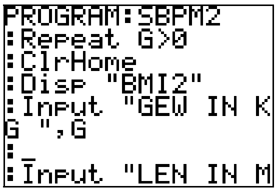


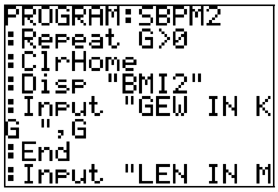
Zuerst verändern wir die Überschrift von `BMI1` in `BMI2`.

Zwischen den beiden `[Input]`-Anweisungen wird eine Leerzeile eingefügt, in die die `[End]`-Anweisung geschrieben wird, die wieder aus dem `CTL`-Untermenü abgeholt wird.







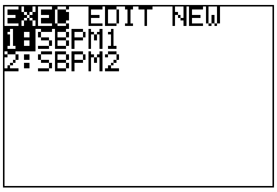


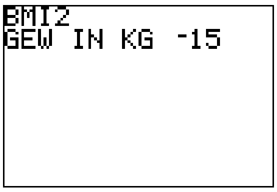
Nun kann das Programm ausgeführt werden. Jetzt stehen schon zwei Programme zur Auswahl.

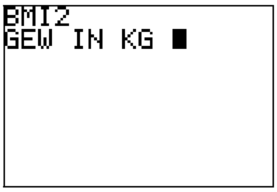
Mit dem Cursor wählen wir `SBPM2` an und starten mit [ENTER] die Ausführung.

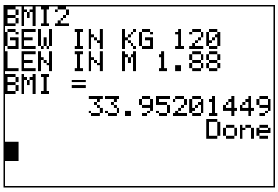
Sollte jemand -15 als Gewicht eingeben, dann wird das Programm nicht weiter ausgeführt, sondern die Frage wird erneut gestellt, und zwar so lange, bis eine positive Zahl eingegeben worden ist.

Ansonsten läuft das Programm ganz nach dem Muster des ersten ab.









Das Abfangen von sinnlosen Daten ist ein wesentliches Merkmal für die Benützerfreundlichkeit eines Programms.

Denk dabei z.B. an die Auflösung einer quadratischen Gleichung. Die Lösungsformel macht nur einen Sinn, wenn der Koeffizient des quadratischen Glieds von Null verschieden ist.

Das Programm lässt sich natürlich noch anpassen, wenn man bedenkt, dass Körpergewichte über 200 kg wohl auch nicht sinnvoll sind. Für die Körpergröße kann vollkommen analog vorgegangen werden.

Wir sehen, dass jede Anweisung in einer neuen Zeile steht, die mit einem Doppelpunkt `[:]` eingeleitet wird.

Das ist nicht unbedingt notwendig. Mehrere Anweisungen lassen sich in eine Programmzeile schreiben. Sie müssen durch einen Doppelpunkt getrennt werden.

Vergiss nicht, dass alle Anweisungen nur über die Menüs in den Programmcode gebracht werden können. Kein einziger Befehl lässt sich über die alphabetische Tastatur eintippen.

3. Programm 3

Bis jetzt haben unsere Programme nur numerische Ergebnisse ausgegeben. Wir wollen nun zusätzlich eine Interpretation des Resultats angeben: Grob lässt sich sagen, dass ein BMI zwischen 20 und 25 ein „gutes“ Ergebnis ist. Wir bauen in das Programm einen Test des Resultats ein, der für die Ausgabe „gut“ oder „nicht gut“ sorgen soll.

Das Programmkonstrukt, das dafür in Frage kommt ist natürlich das `[If]` im `CTL`-Untermenü.

Diese If-Abfrage gibt es in verschiedenen Varianten, von denen hier die einfachste vollkommen ausreicht. Liegt der BMI zwischen 20 und 25, dann erscheint "`GOED`" (= GUT) auf dem Schirm, andernfalls "`NIET GOED`" (= NICHT GUT).

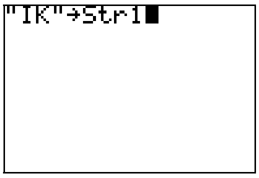
Wir setzen sogenannte *Stringvariable* (Zeichenkettenvariable) ein. Wenn wir einen Text in einem String speichern wollen, dann geschieht das folgendermaßen:

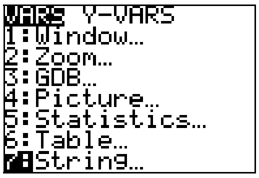
Im Hauptschirm schreibt man z.B.: `"IK"→Str1`.

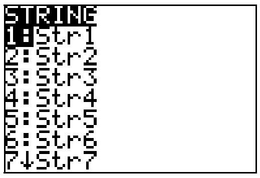
Anführungszeichen und Buchstaben werden wie üblich geschrieben. Der Pfeil steht für den Speicherauftrag [STO▸] und die vom Rechner vorgeschriebene Stringvariable `Str1` müssen wir aus dem [VARS]-Menü holen.

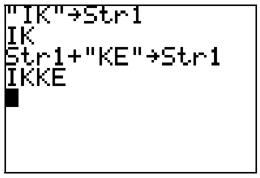
Über die Taste [VARS] erhalten wir einen Schirm mit zwei Untermenüs. In <VARS> selektieren wir Option 7 und mit [ENTER] wird der angewählte String in die Ausgabe übertragen.

Strings lassen sich zusammenhängen und man kann an Strings vorne und hinten etwas anfügen. Eine Stringvariable wird geleert, indem man sie mit dem „Leerstring“ überschreibt: `""→Str1`.



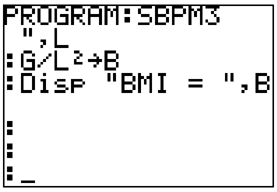


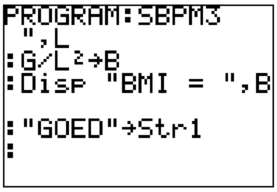


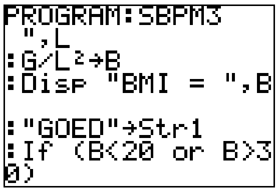


Kehren wir zurück und erzeugen wir ein neues Programm SBPM3, wobei wir das vorige voll übernehmen. Der Titeltext wird in BMI3 abgeändert. Dann gehen wir ans Ende des vorliegenden Programmcodes und fügen ein paar Leerzeilen an.

Zuerst speichern wir wie eben gezeigt einen Text in einer Stringvariablen.



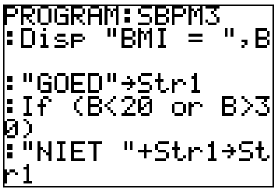


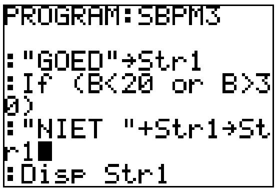


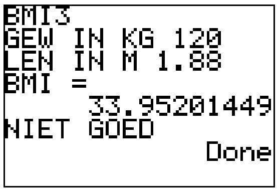
Im Anschluss daran folgt der Test mit einer Fallunterscheidung. Da es sich um eine zusammengesetzte Bedingung handelt, müssen wir die Klammern `()` verwenden. Die logische Verknüpfung `[or]` holen wir aus dem `[TEST] <LOGIC>`-Menü.

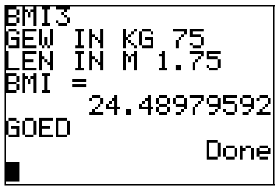
Zum Schluss vervollständigen wir die Zeile nach der Fallunterscheidung. Hier wird beim Zutreffen der entsprechenden Bedingung die Meldung `GOED` abgeändert in `NIET GOED`. Dieser Text wird schlussendlich mit `[Disp]` zur Ausgabe gebracht.

Wenn wir nun das Programm mit verschiedenen Daten laufen lassen, dann erhalten wir die jeweils passende Auswertung.









Wir haben hier das Beispiel einer `[If]`-Konstruktion ohne `[Then]` und auch ohne `[Else]`. Wenn der Test positiv verläuft (die Bedingung wird wahr), dann wird die nächste, dem `If` folgende Anweisung ausgeführt, wenn nicht, dann wird diese Anweisung übersprungen und mit der übernächsten fortgesetzt. Das geht nur, wenn der Unterscheidung eine einzige Anweisung folgen soll.

Wenn mehrere Anweisungen bei Eintreffen der Bedingung ausgeführt werden sollen, dann muss mit `[Else]` gearbeitet werden. Die Syntax sieht dann anders aus.

4. Programm 4

Informiert man sich weiter über den BMI, dann kann man folgendes finden:

BMI < 20 geht in die Richtung Anorexie (Appetitlosigkeit) und man kann schon von Untergewicht sprechen,

20 ≤ BMI < 25 ist der „Idealwert“,

25 ≤ BMI ≤ 30 bedeutet Übergewicht,

BMI > 30 stellt bereits einen deutlichen Risikofaktor dar.

Wenn wir das in unserem Programm berücksichtigen wollen, haben wir es mit einer komplexen If-Konstruktion zu tun. Nach der Plausibilitätskonstrolle könnte das Programm dann so lauten:

```
If B<20
Then
Disp "TE LICHT"
    Else
        If B<25
        Then
        Disp "PROFICIAT"
            Else
                If B<30
                Then
                Disp "OVERGEWICHT"
                    Else
                    Disp "GEVAARLIJK"
```

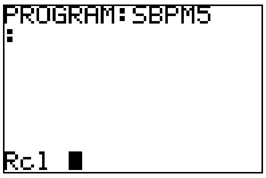
Die Einrückungen sollen die Struktur des Programms übersichtlich machen. Auf dem TI-83+ werden die Programmzeilen nicht so aussehen. Das Programm kann nun einige Testläufe absolvieren.

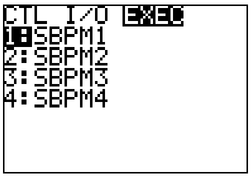
5. Programm 5

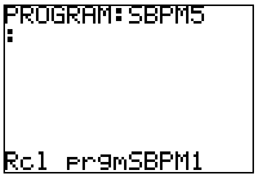
Zum Schluss wollen wir für eine Gruppe von Personen (z.B. die Schüler einer Klasse) den BMI berechnen und die Werte für späteren Gebrauch in eine Liste schreiben.

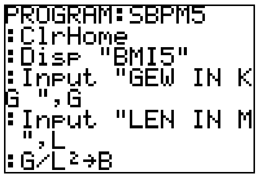
Dafür schreiben wir ein neues Programm `SBPM5`, wobei wir von der Version `SBPM1` ausgehen.

Der Bildschirm sollte das folgende Aussehen gewinnen:

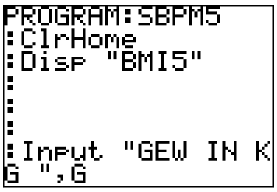


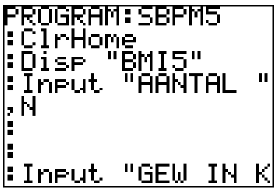


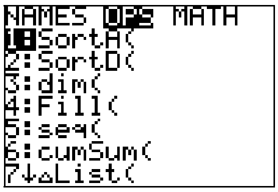


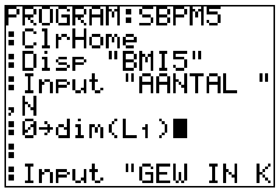


Nach dem Programmtitel fügen wir einige Leerzeilen ein. Dann stellen wir die Frage nach der Anzahl der Personen und speichern diese Zahl in der Variablen `N`. Anschließend müssen wir dafür sorgen, dass die Liste `L1` existiert und leer ist. `[dim]` finden wir im Menü `[LIST]` im Untermenü `<OPS>`.









Jetzt wird eine Schleife mit der Anweisung `[For]` programmiert.

Die vollständige Syntax dafür lautet:

`For(SchleifenVariable,Anfangswert,Endwert,[Schrittweite])`.

Dabei ist die Schrittweite optional. Ihr Standardwert ist 1. Die `[For]`-Anweisung findet man in der oberen Hälfte des `CTL`-Menüs. Hier lassen wir auch nochmals den Schirm löschen. Bei der Anzeige des BMI zeigen wir zusätzlich die Nummer der entsprechenden Person.

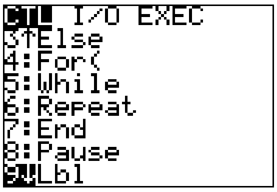
Dann wird der Wert in die Liste übertragen. Um die Daten auch vom Schirm lesen zu können, fügen wir vor der Abfrage nach den nächsten Daten eine `[Pause]` (aus `<CTL>`) ein.

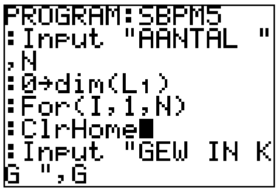
Anstelle von `[Pause]` ließe sich auch eine `[For]`-Warteschleife verwenden. Der Wert, wie lange -- im Hintergrund - gezählt werden soll, dass man die Ergebnisse bequem lesen kann, muss ausgetestet werden. Ein brauchbarer Wert ist `For(X,1,200) : End`

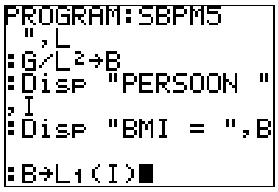
Die `[For]`-Schleife ist mit der `[End]`-Anweisung zu schließen.

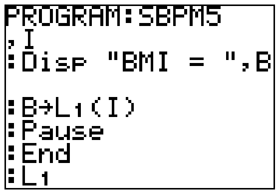
Die Liste `L1` kann nun betrachtet werden. Und das führen wir dann mit den dafür besser geeigneten Listenwerkzeugen durch: STAT <EDIT> .....

Natürlich könnte man bei dieser Gelegenheit die Körpergewichte und -größen auch gleich in Listen übertragen und für statistische Auswertungen bereit stellen.

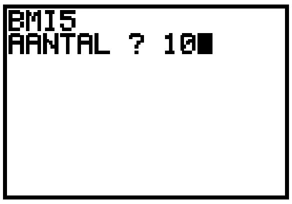


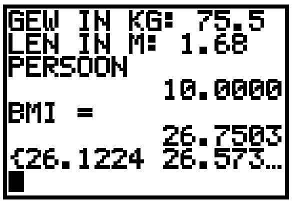


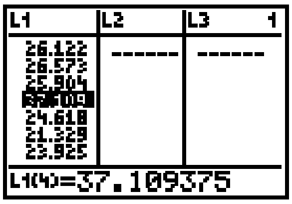


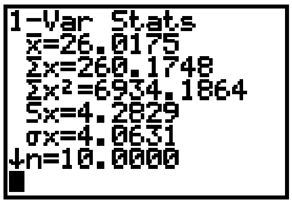


Ein Testlauf für 10 Personen (inkl. statistischer Auswertung).









6. Übersicht Kontrollstrukturen

TURBO PASCAL	TI-BASIC

Repeat
Anweisung(en)
Until (Bedingung)
	`Repeat(Bedingung)`
	`Anweisung(en)`
	`End`

While (Bedingung) Do
Begin
Anweisung(en)
End
	`While (Bedingung(en))`
	`Anweisung(en)`
	`End`

If …Then …Else...;
 	
If …then
Begin
Anweisung(en);
end Else
begin
Anweisung(en);
end;
	`If(Bed)`
	`eine Anweisung`
	
	`If(Bed)`
	`Then`
	`Anweisungen`
	`End`
	
	`If (Bedingung(en))`
	`Then`
	`Anweisungen`
	`Else`
	`Anweisungen (End)`

For ….To…..Do (For…Downto…Do)
Begin
Anweisung(en)
End;
	`For(VAriable,Anf,Ende,(Step))`
	`Anweisungen`
	`End`

7. Nützliche Anweisungen (Befehle)

Ein Programm **erzeugen**	[PGRM] NEW 1: Create New
Eine **Zeile einfügen**	[PGRM] EDIT Cursor an die Stelle setzen, wo die Zeile eingefügt werden soll 2nd INS und ENTER
Eine **Programmzeile löschen**	[PGRM] EDIT Cursor an die Stelle setzen, wo die Zeile verschwinden soll CLEAR (Eventuell gegefolgt von DEL um auch die leere Zeile zu löschen)
Ein Programm **aufrufen**	[PGRM] EXE …
Ein Programm **unterbrechen**	[ON] auf dem Schirm erscheint ERR:BREAK Mit 1 kehrt man in den Rechenschirm zurück, Mit 2 gelangt man zu der Programmstelle, wo das Programm unterbrochen wurde
Ein Programm **ändern**	[PGRM] EDIT …
Ein Programm **löschen**	2nd [MEM] 2: Mem Mgmt/Del 7:Prgm … Das Programm wählen ... [DEL]
Ein Programm **archivieren**	wie oben – wähle das Programm, dann [ENTER]; beachte das Sternchen
Ein Programm **dearchivieren**	wie archivieren – das Sternchen verschwindet
Ein Programm **von einem anderen Gerät empfangen** oder **auf ein anderes Gerät übertragen**	Beide Geräte einschalten und mit dem Übertragungskabel ordentlich verbinden **Empfänger**: [LINK] wähle RECEIVE …. `Waiting` erscheint **Sender**: [LINK] wähle SEND 3: Prgm selektiere das Programm (auch mehrere) die Auswahl wird durch ein kleines Viereck angezeigt gehe zu TRANSMIT um die Auswahl zu senden

8. Die Programme aus dem Skriptum

Programm SBPM1

```
ClrHome
Disp "   BMI1"
Input "GEW IN KG ",G
Input "LEN IN M ",L
G/LÜüB
Disp "BMI = ",B
```

Programma SBPM2

```
Repeat G>0
ClrHome
Disp " BMI2 "
Input "GEW IN KG ",G
End
Input "LEN IN M ",L
G/LÜüB
Disp "BMI = ",B
```

Programm SBPM3

```
Repeat G>0
ClrHome
Disp " BMI3 "
Input "GEW IN KG ",G
End
Input "LEN IN M ",L
G/LÜüB
Disp "BMI = ",B
"GOED"üStr1
If (B<20  or  B>25)
"NIET "+Str1üStr1
Disp Str1
```

Programm SBPM4

```
Repeat G>0
ClrHome
Disp " BMI4 "
Input "GEW IN KG ",G
End
Input "LEN IN M ",L
G/LÜüB
Disp "BMI = ",B
If B<20
Then
Disp "TE LICHT"
Else
If (B<25)
Then
Disp "PROFICIAT"
Else
If (B<30)
Then
Disp "OVERGEWICHT"
Else
Disp "GEVAARLIJK"
```

Programm SBPM5

```
ClrHome
Disp " BMI5 "
Input "AANTAL ? ",N
0üdim(L )
For(I,1,N)
ClrHome
Input "GEW IN KG ",G
Input "LEN IN M ",L
G/LÜüB
Disp "PERSOON  ",I
Disp "BMI = ",B
BüL (I)
Pause         oder           For(X,1,200) : End
End
L
```

9. Literaturverzeichnis

-) Guidebook voor TI-83 Plus, hoofdstuk 16 (education.ti.com/guides)

-) Hilde De Maesschalk, Pedagogische begeleiding Bisdom Gent
“ Waarom werken met programma’s voor een GRM in de tweede graad?”

-) Luc Gheysens “Exploreren uitgaande van een Internettoepassing”
6° T [3] Symposium Leuven 15 maart 2003

-) www.slankadvies.nl/ideale-lichaamsgewicht.html#top

-) www.henkshoekje.com
Website von Henk Pfaltzgraff
Zeigt u.a. wie man mit kleineren Zeichen am Schirm arbeiten kann

-) www.pandd.demon.nl
Website von Dick Klingens
Ein Angebot von über 140 Programmen

-) users.pandora.be/wiskunde
Portal für Mathematik