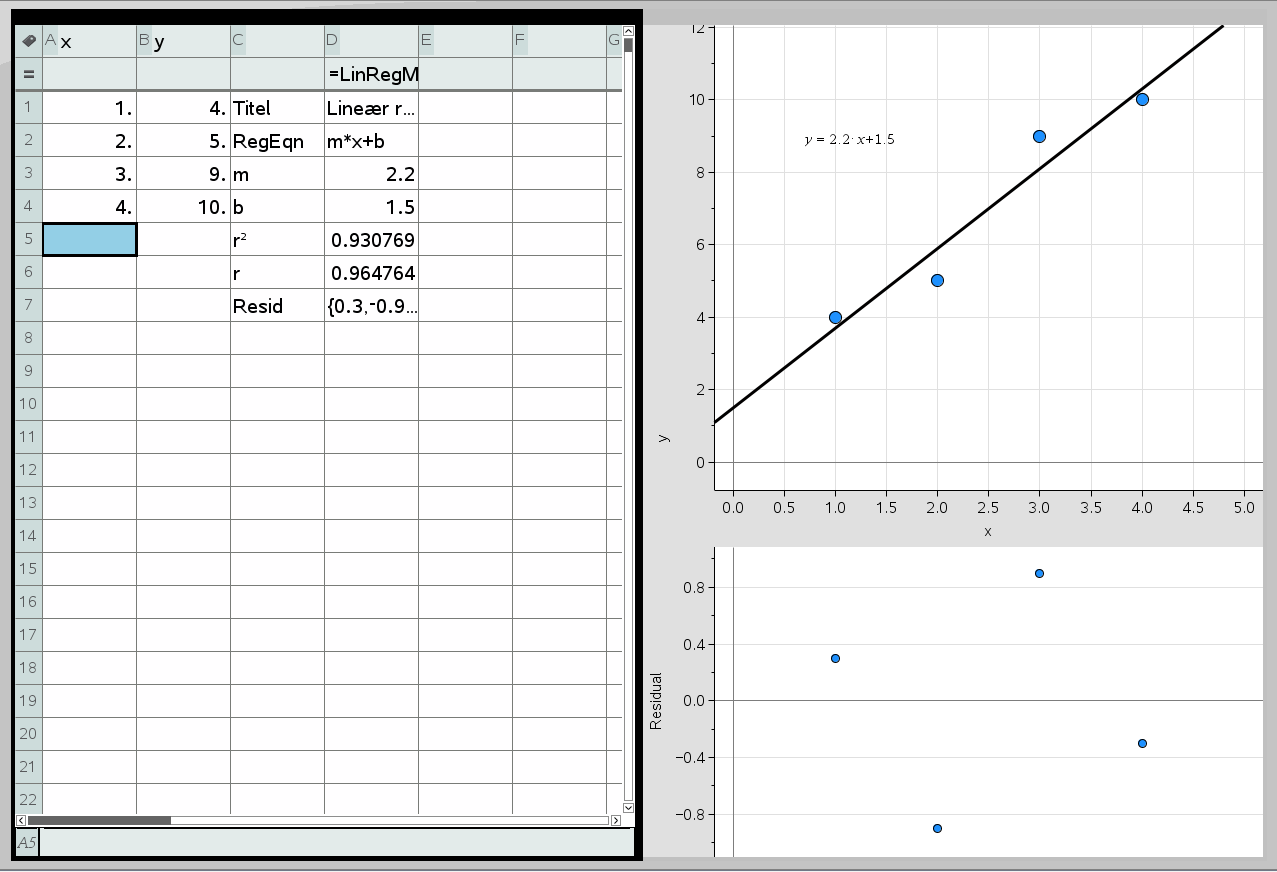
**Regression og forklaringsgrad i Nspire**

Som et eksempel vil vi se på sildebenet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | 1 | 2 | 3 | 4 |
| *y* | 4 | 5 | 9 | 10 |

hvor vi forventer en lineær sammenhæng. I Nspire laves de sædvanlige 4 ting:

(i) sildeben  
(ii) hurtiggraf med regression for at tegne tendenslinjen  
(iii) residualplot  
(iv) regression i regnearket for at beregne forklaringsgraden



Når man indtaster *x* og *y* i Nspire kommer sildebenet jo til at stå som lodrette kolonner. Inde i Nspire bliver *x* og *y* dog gemt som vandrette lister:

**Forklaringsgrad**. For at beregne forklaringsgraden skal vi først beregne gennemsnittet af *y*’erne. Gennemsnittet kaldes normalt for

Her har vi skrevet formlen som man skriver den i beregneren i Nspire: beregner summen af de fire sildebens-*y’*er

og beregner længden af *y*-listen som er 4. Ved at dividere de to tal fås et gennemsnit på

.

Dernæst beregner vi

I betegnelsen står *SS* for ”sum of squares” altså ”sum af kvadrater”. Et kvadrat betyder at sætte i anden. Vi ser at kvadraterne tages af forskellene mellem de enkelte sildebens-*y*’er og deres gennemsnit . Vi vil gerne have at alle forskelle giver positive bidrag, så derfor sættes de i anden. Til sidst summeres de kvadrerede forskelle. bliver på denne måde et mål for hvor meget sildebens-*y*’erne afviger fra deres gennemsnit. Skrevet på lang form er

**a)** Vis at .

Så beregner vi

hvor er den lineære sammenhæng som Nspire beregner ved lineær regression. Med kigger vi ikke længere på forskellene mellem sildebens-*y*’erne og deres gennemsnit, men i stedet for på forskellene mellem sildebens-*y*’erne og den lineære sammenhængs *y*’er. Disse forskelle kaldes for residualer og i Nspire ser vi dem i residualplottet. Vi har derfor sat et ”*res*” (residual) på *SS*. I summen er der for hvert sildebens-*y* et tilhørende lineær-sammenhængs-*y*, som er beregnet vha. det tilsvarende sildebens-*x*. Skrevet på lang form er

**b)** Vis at .

Til sidst er vi klar til at beregne forklaringsgraden

**c)** Vis at og sammenlign med forklaringsgraden i Nspire.

Filosofien bagved formlen for forklaringsgraden er følgende. Hvis der ikke er nogen sammenhæng mellem *x* og *y* så må tendenslinjen være vandret: selvom vi varierer *x* så ændrer *y* sig ikke, altså ingen sammenhæng. Man kan vise, se ekstraopgave 2, at den vandrette linje , der skærer *y*-aksen i gennemsnittet, er den linje som giver det mindst mulige for en ikke-sammenhæng (). Denne værdi kalder vi for . Man forsøger så om en sammenhæng () kan gøre mindre end . Jo mindre jo bedre. Kan komme helt ned på 0 bliver forklaringsgraden perfekt 1.

**Lineær regression**. Formålet med lineær regression er at gøre afvigelsen mellem de fire sildebens-*y*’er og den påståede lineære sammenhængs *y*’er så lille som muligt. Dermed bliver forklaringsgraden så stor som mulig.

**d)** Forklar vha. formlen for forklaringsgraden at et minimalt giver et maksimalt .

For at finde minimummet for har vi brug for at kunne arbejde med en funktion af to variable. Vi er vant til at en funktion kun afhænger af én variabel *x*. Hvis for eksempel , så er . Men funktioner kan også afhænge af to variable som for eksempel .

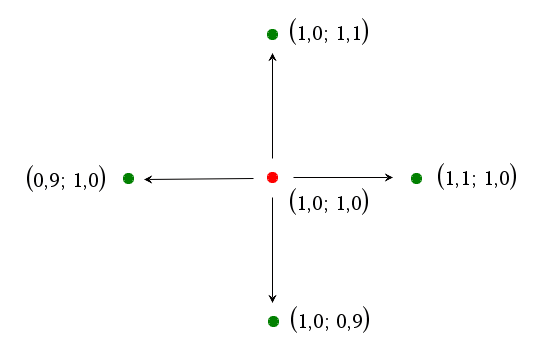
**e)** Vis at .

Lad os skrive afvigelsen på lang form

Vi ser at afhænger af to variable tal *a* og *b*. Vi skriver derfor funktionen som .

**f)** Vis at

Nspire forsøger at finde det *a* og *b* som giver det mindste . Mindsteværdien må altså være mindre end eller lig med 58. Nspires regressionsberegning viser at *a* = 2,2 og *b* = 1,5 giver det mindste . I opgave b) beregnede du denne værdi til 1,8. Hvordan finder Nspire frem til at ? Nspire prøver sig frem! I det følgende vil vi på tilsvarende måde lede efter minimummet. Lad os sige at vi vælger at starte i som vi opfatter som punktet i et almindeligt koordinatsystem, se billedet.



Fra (1,0; 1,0) prøver vi at gå enten

0,1 til højre hen til (1,1; 1,0) eller

0,1 op hen til (1,0; 1,1) eller

0,1 til venstre hen til (0,9; 1,0) eller

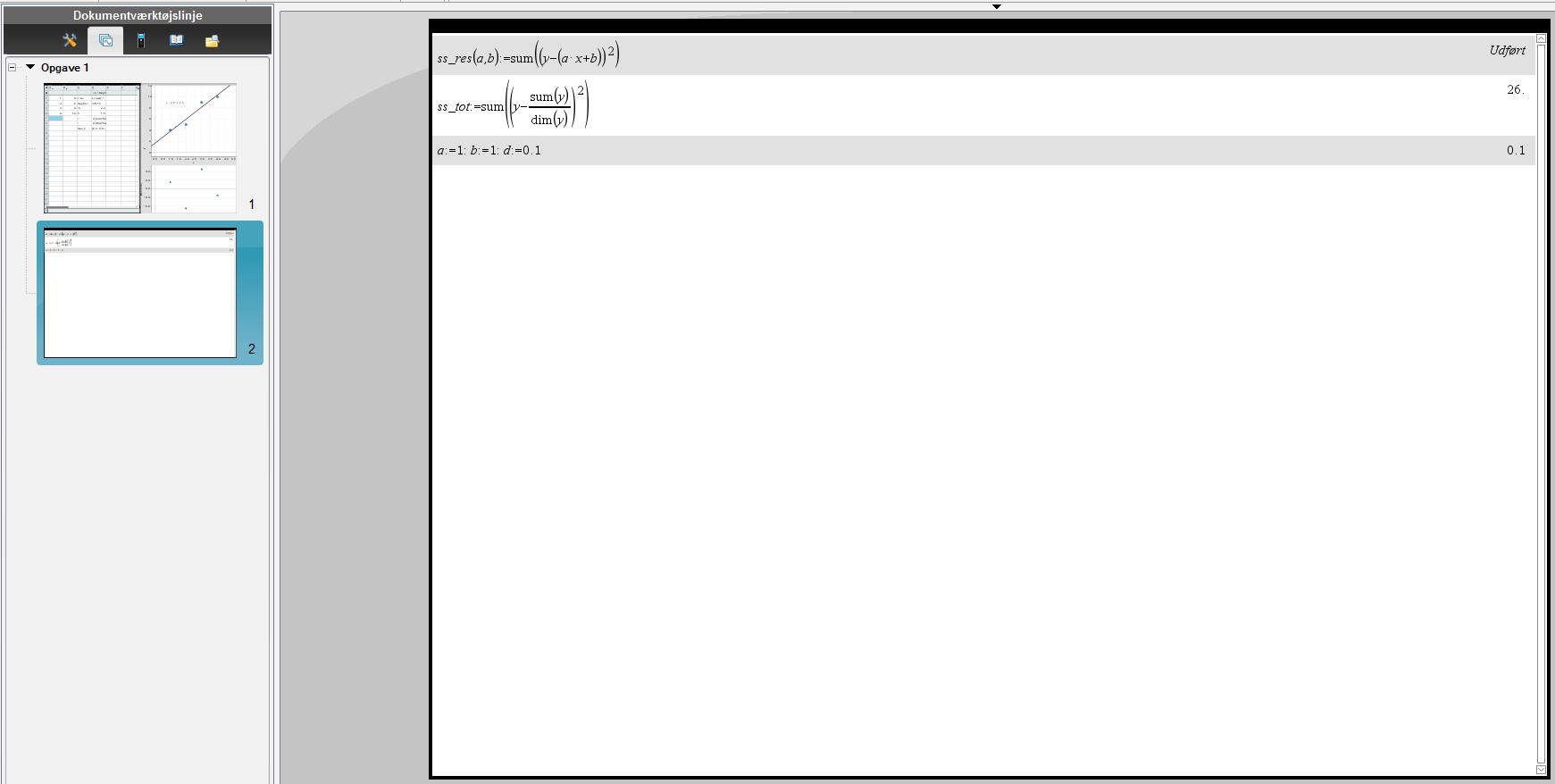
0,1 ned hen til (1,0; 0,9).

Ved vores bevægelse i ændrer sig. Tilvæksten er

**g)** Vis at

I g) ser vi at det bedste skridt er at gå 0,1 til højre fordi det bringer os længst ned i værdi. Dette ser vi ved at tilvæksten er mest negativ. Dermed bliver (1,1; 1,0) det nye forberede udgangspunkt for (*a*, *b*). Vi kunne gentage g) om og om igen og hver gang komme tættere på minimummet i. Hvis skridtlængden er for stor, så ingen af de fire tilvækster i g) bliver negativ, så halverer vi blot skridtlængden til og regner g) igen. Som du også fandt ud af i g) er hvert skridt temmelig besværligt at lave med en lommeregner. Men så er det godt at vi har computere der utrætteligt kan gentage g) så mange gange det skal være!

**Program for g)**. I Nspire kan vi skrive h) som et lille program. Først indsætter du en ny side i samme dokument som sildebenet og vælger beregner. I beregneren skal du definere , og startværdierne for *a* og *b* og skridtlængden *d:*

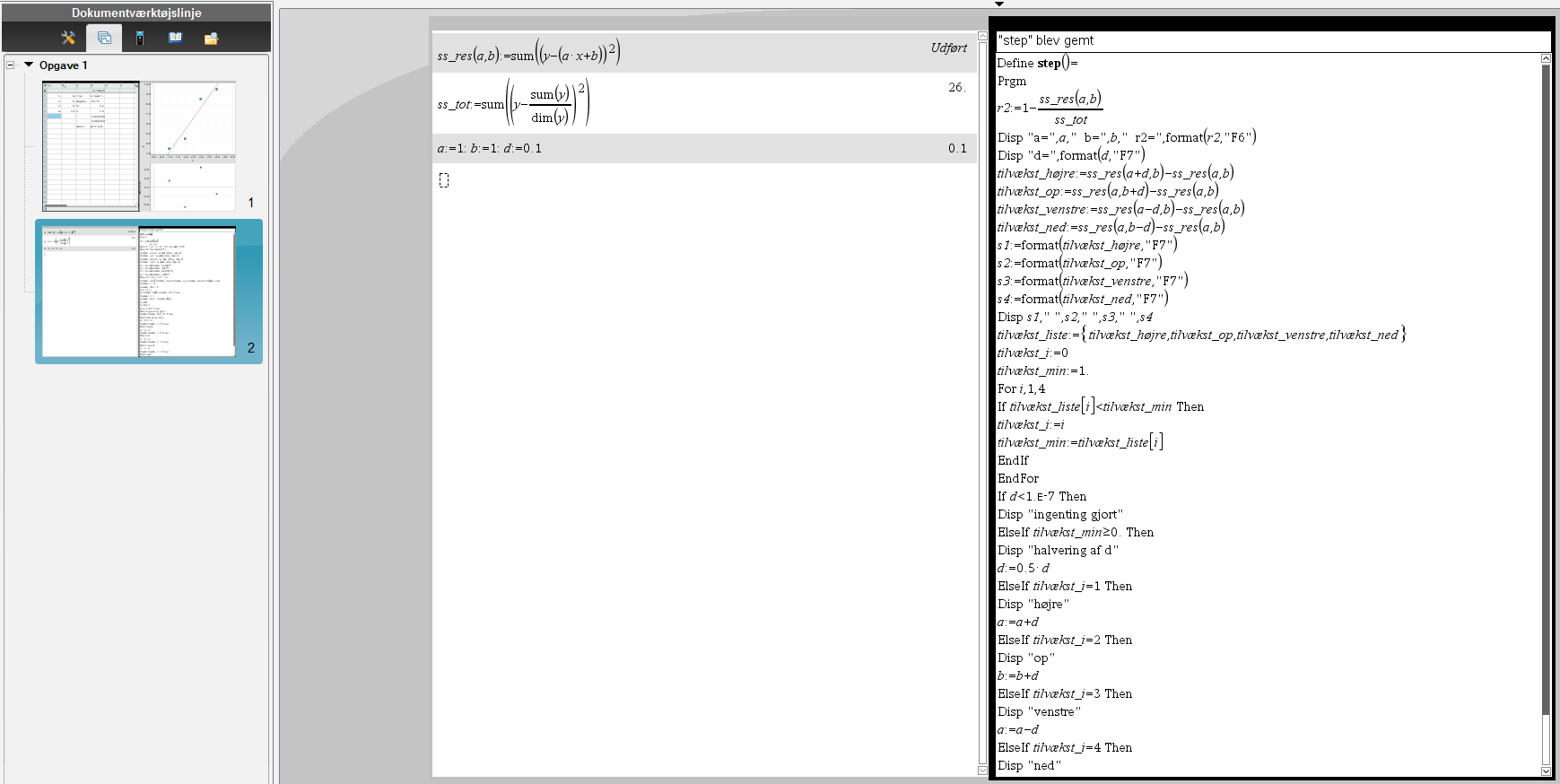


Her har vi altså valgt at sætte *a* = 1, *b* = 1 og *d* = 0,1. I Nspire bruges : som separator og det skrives som

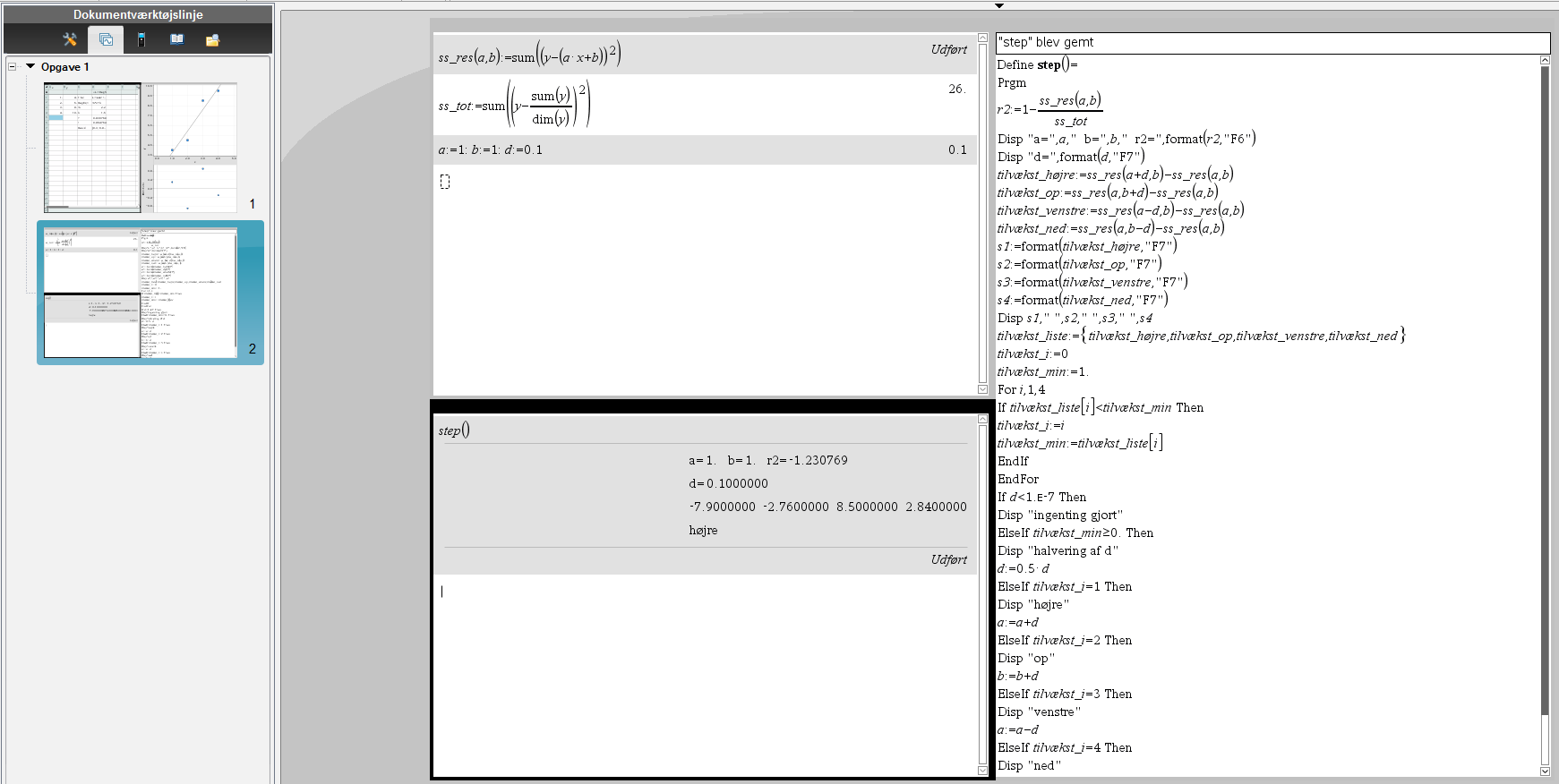
a:=1: b:=1: d:=0.1

Ude i værktøjskassen vælger du ”funktioner og programmer” -> ”programeditor” -> ”ny” og kalder programmet for step. Så indtaster du programmet (copy-paste nedenstående program). Højreklik til sidst og vælg ”kontroller syntaks og gem”. Dermed er programmet godkendt og gemt.

r2:=1-ss\_res(a,b)/ss\_tot  
:Disp "a=",a," b=",b," r2=",format(r2,"F6")  
:Disp "d=",format(d,"F7")  
:tilvækst\_højre:=ss\_res(a+d,b)-ss\_res(a,b)  
:tilvækst\_op:=ss\_res(a,b+d)-ss\_res(a,b)  
:tilvækst\_venstre:=ss\_res(a-d,b)-ss\_res(a,b)  
:tilvækst\_ned:=ss\_res(a,b-d)-ss\_res(a,b)  
:s1:=format(tilvækst\_højre,"F7")  
:s2:=format(tilvækst\_op,"F7")  
:s3:=format(tilvækst\_venstre,"F7")  
:s4:=format(tilvækst\_ned,"F7")  
:Disp s1," ",s2," ",s3," ",s4  
:tilvækst\_liste:={tilvækst\_højre,tilvækst\_op,tilvækst\_venstre,tilvækst\_ned}  
:tilvækst\_i:=0  
:tilvækst\_min:=1.0  
:For i,1,4  
:If tilvækst\_liste[i]<tilvækst\_min Then  
:tilvækst\_i:=i  
:tilvækst\_min:=tilvækst\_liste[i]  
:EndIf  
:EndFor  
:If d<0.0000001 Then  
:Disp "ingenting gjort"  
:ElseIf tilvækst\_min≥0. Then  
:Disp "halvering af d"  
:d:=0.5\*d  
:ElseIf tilvækst\_i=1 Then  
:Disp "højre"  
:a:=a+d  
:ElseIf tilvækst\_i=2 Then  
:Disp "op"  
:b:=b+d  
:ElseIf tilvækst\_i=3 Then  
:Disp "venstre"  
:a:=a-d  
:ElseIf tilvækst\_i=4 Then  
:Disp "ned"  
:b:=b-d  
:EndIf



Nu deler du venstre del af skærmen op i to og vælger igen beregner for det nye felt forneden. Her udfører du programmet step én gang ved at skrive step() og trykke på enter:

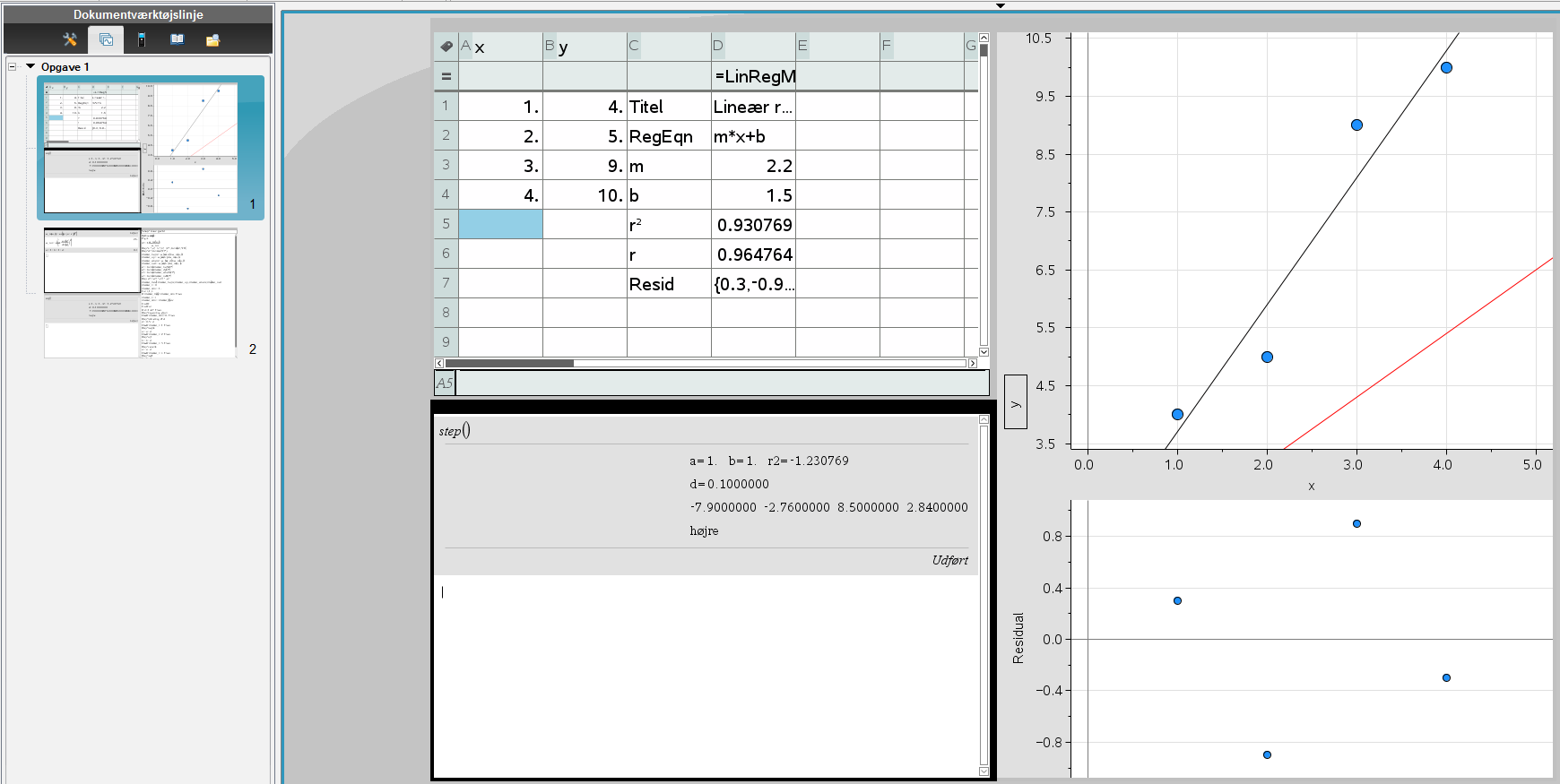


Man ser at programmet regner rigtigt! Trykker du på enter igen udføres programmet igen. Holder du enter nede gentages programmet igen og igen. Når *d* bliver mindre end 0,0000001 ændrer *a* og *b* sig ikke mere og programmet skriver ”ingenting gjort”. Nu skulle du gerne have fundet det optimale (*a*, *b*) og den tilhørende forklaringsgrad!

Fordelen ved at have to felter med beregner er at dine indtastninger i øverste felt ikke bliver skubbet opad når du i det nederste felt udfører step() mange gange. Hvis du vil starte forfra gentager du blot definitionen af *a*, *b* og *d* i øverste felt igen. Herefter vil step() i det nederste felt også starte forfra.

Hvis du forventer mere ekstreme værdier for *a* og *b*, som f.eks. *a* = 30000 og *b* = -50000, er det bedre at lade begyndelsesværdierne være tilsvarende ekstreme, f.eks. *a* = 10000, *b* = -10000 og *d* = 1000. Find også stedet i programmet hvor afskæringsværdien for *d* er sat til 0,0000001, og sæt denne til f.eks. 0,001.

Du kan tegne den fundne tendenslinje ved at vende tilbage til side 1. Ude i værktøjskassen vælger du ”undersøg data” -> ”plot funktion” og indtaster . Linjen kan f.eks. farves rød. Derefter deler du den venstre del af skærmen op i to og vælg igen beregner for det nye felt forneden. Her kan du så steppe *a* og *b* samtidigt med at du ser hvordan den røde linje nærmer sig Nspires tendenslinje. Husk at sætte *a* = 1, *b* = 1 og *d* = 0,1 på side 2 før du begynder at steppe *a* og *b*.



**Ekstraopgave 1 der kræver logaritmer.**

Nspire benytter en tilsvarende minimumsmetode for eksponentiel og potensiel regression, dog tages logaritmen af alle *y*’er. Det vil sige at

i) Eksponentiel regression

ii) Potensiel regression

Vis at minimumsmetoden for eksponentiel og potensiel regression på sildebenet fører til værdier for *a*, *b* og forklaringsgrad *r2*, der stemmer overens med regression i Nspire.

**Ekstraopgave 2 der kræver differentialregning.**

Som vi tidligere har set er afvigelsen mellem de fire sildebens-*y*’er og den lineære sammenhængs *y*’er

Her er de fire *x*’er og de fire *y*’er jo **faste** tal. Omvendt er *a* og *b* **variable** tal. Vi ser altså at er en funktion af de to variable *a* og *b*. For en ikke-sammenhæng () reducerer til

som på lang form er

De fire *y*’er er stadig **faste** tal og *b* er stadig et **variabelt** tal. Dermed er blevet til en funktion af kun *b*. Du skal nu bevise den tidligere påstand at for en ikke-sammenhæng er minimal når *b* er lig med gennemsnittet af *y*’erne:

Lav beviset i to trin:

**i)** Kig på det første led i og vis at

Hvis du ikke har lært at differentiere en sammensat funktion så lav udregningen ved at gange ud og derefter differentiere resultatet mht. *b*. Gør det tilsvarende for de tre andre led i .

**ii)** I minimummet på grafen for er tangenthældningen jo nul

Differentiér mht. *b* hvor du bruger dine resultater fra i). Sæt resultatet lig med nul og vis at løsningen bliver . Overvej hvorfor du med denne løsning har fundet et **minimum**. Dermed er beviset færdigt.