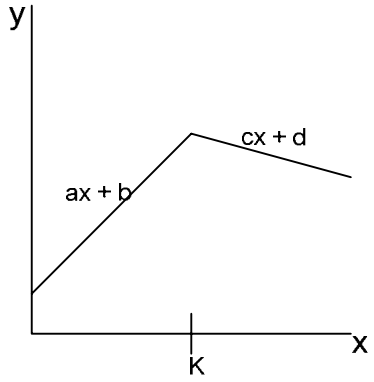
Problem:

Variables y and x are related as shown:



Our model is a continuous function f(x):

$$y = f(x) = \begin{Bmatrix} ax + b & x \le K \\ cx + d & x \ge K \end{Bmatrix} + e$$ [1]

Objective

Estimate all parameters including the break point K[2] with confidence intervals.

Reformulate

If the two lines should meet at x = K, then f(x) can be reformed[3]:

$$f(x) = (ax + b) * [x<=K] + (c(x-k) + ak + b) * [x>=K] + e$$

where [ ] is the indicator function:

$$[L] = \begin{cases} 1 & L = \text{true} \\ 0 & L = \text{false} \end{cases}$$

Wording

The problem, we are going to solve should more precisely be described as a "**segmented regression problem**" solved by means of nonlinear fitting.

[1] e is normally distributed
[2] knot
[3] see proof below

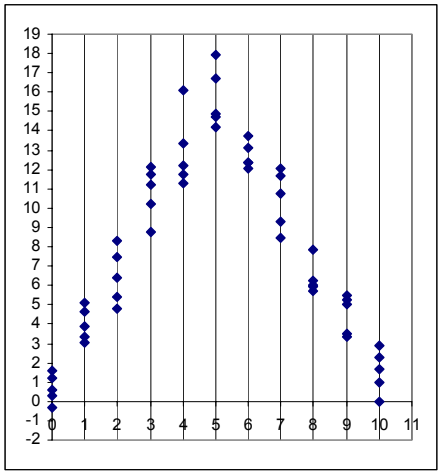
Simulating an example

We simulate via SplineLin.xls (green area was chosen)

Slope 1	**3**	Equation up to change at x = 5,0	Y = 1,0 + 3,0 * X
Intersection 1	**1**		
Turning point is at X-value	**5**		
Slope 2	**-3**	Equation from change at x = 5,0	Y = 16,0 -3,0 * (X - 5,0)
Intersection 2 (calculated)	16		

Data Y = (ax + b) * [if x<=k] + (c(x-k) + ak + b) * [if x>=k] + Normal(0,1)

X	Y
0	1,232278253
0	0,603233409
0	1,571885945
0	0,282846817
0	-0,287596229
1	5,112261196
1	3,361909558
1	4,638542298
1	3,02068829
1	3,902428606
2	7,469328173
2	4,773730474
2	8,301793235
2	6,366609545
2	5,429732445
3	11,71854443
3	12,16343367
3	11,17503268
3	8,769224057
3	10,19877767
4	16,09476285
4	12,22007441
4	11,77922753
4	13,33791882
4	11,27767793
5	14,6909043
5	16,69642444
5	14,90564868
5	17,96115771
5	14,21237218
6	13,7055562
6	13,102971
6	12,38364092
6	12,38459026
6	12,07103302
7	11,63822609
7	12,06801012
7	8,463592537
7	10,78336346
7	9,293967821
8	5,742664636
8	5,995921319
8	5,969546043
8	6,269699694
8	7,866197506
9	5,051103518
9	5,515355306
9	5,224646645
9	3,333581615
9	3,485534381
10	2,852426847
10	0,963189417
10	1,663694029
10	-0,044572181
10	2,288131804



SPSS 13

Substituting the indicator function

Unfortunately there is no indicator function in SPSS so that we have to use a trick by using the range-function:

range(x,0,k) gives 1 for x<=k else 0
range(x,k,max(x)) gives 1 for x>=k else 0

The next disadvantage is, that we also have to substitute the max()-function, because there is no such function in SPSS.

So we have to insert the maximum of x into the formula.
In this example it could be any value above 10 because our x-data range from 0 to 10

The complete formula:

Y = (ax + b) * range(x,0,k) + (c(x-k) + ak + b) * range(x,k,10)

SPSS-Syntax

```
MODEL PROGRAM A=0.1 B=0.1 K=1 C=0.1 .
COMPUTE PY = (a*x+b)*range(x,0,k-0.001)+(c*(x-K) +a*K+b)*range(x,k+0.001,10).
CNLR y
/OUTFILE='Spline1.TMP'
/PRED PY
/BOUNDS A >= 0; B >= 0; K >= 0; C < 0
/SAVE PRED RES(ry41)
/CRITERIA ITER 100 STEPLIMIT 2 ISTEP 1E+20 .
```

The little correction "-0.001" resp. "+0.001" is also essential for the algorithm to give it a range to variate.
If you want to plot the result use

```
GRAPH
  /SCATTERPLOT(OVERLAY)=x x  WITH y py  (PAIR)
  /MISSING=LISTWISE .
```

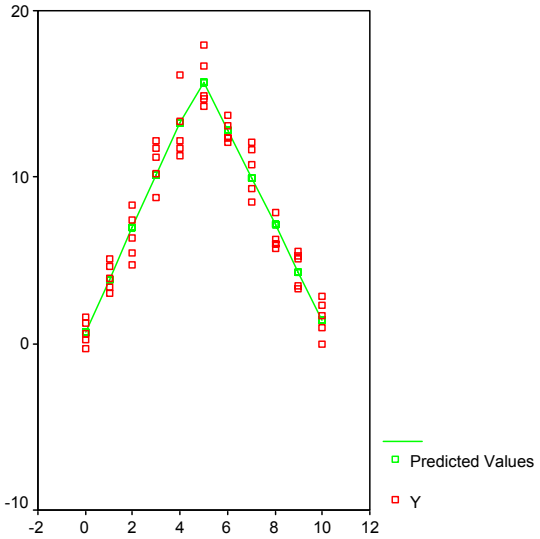
Remark

If you want to run the program with new data you have to delete all variables except X and Y before you run this program. Because SPSS always creates new predicted values with new names.

Result

Parameter	Estimate	Asymptotic Std. Error	Asymptotic 95 % Confidence Interval Lower	Upper
A	3,132066142	,174671733	2,781398006	3,482734278
B	,716441394	,427856618	-,142516608	1,575399396
K	4,879106983	,116530933	4,645161372	5,113052593
C	-2,841337504	,132039419	-3,106417699	-2,576257309

Parameter	Estimated	Confidence	Real was
A	3.1	[2.8 ; 3.5]	3
B	0.7	[-0.1 ; 1.6]	1
K	4.9	[4.6 ; 5.1]	5
C	-2.8	[-3.1; -2.6]	-3



STATA 9.2

As we have an indicator function in STATA, we can perform the segmented regression in one line.

The COND() function

We can use the cond-function in STATA as an indicator function.
COND(L,a,b) is defined as:

$$\text{COND}(L,a,b) := \begin{cases} a & L \text{ is true} \\ b & L \text{ is false} \end{cases}$$

Example:
COND(x<5, 1, 0) would give 1 for if x<5 and 0 for x>=5
There is an extension COND(x<5, 1, 0, -1) which would operate like the one before, but which moreover would output -1 if x is missing.

Using COND as an indicator-function the whole syntax would be one line

Syntax

nl (y = cond(x <= {k}, {a}*x + {b}, {c}*x + {k}*({a} - {c}) + {b})) , initial (a 1 b 1 c 1 k 1)

where
nl ( ) stands for nonlinear regression
{ } marks a parameter to be estimated
initial gives for each parameter a guess in which range he should look for a solution
here: start with a=1, b=1, c=1 and k=1
i.e. it's not around 100 or 1,000,000

Result

```
      Source |       SS       df       MS                      
-------------+------------------------------           Number of obs =        55
       Model |  1230.53872     3  410.179575           R-squared     =    0.9405
    Residual |  77.8010355    51   1.5255105           Adj R-squared =    0.9370
-------------+------------------------------           Root MSE      =  1.235116
       Total |  1308.33976    54  24.2285141           Res. dev.     =  175.1584

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          /k |   4.879108    .116531    41.87   0.000     4.645162    5.113054
          /a |   3.132064   .1746717    17.93   0.000     2.781396    3.482732
          /b |   .7164454   .4278566     1.67   0.100    -.1425125    1.575403
          /c |  -2.841337   .1320394   -21.52   0.000    -3.106417   -2.576257
------------------------------------------------------------------------------
* (SEs, P values, CIs, and correlations are asymptotic approximations)
  Parameter b taken as constant term in model & ANOVA table
```

Extension

With just one more option, we can also perform a bootstrap of 50 complete draws of our sample, to check for robustness of the result:

```
nl  (  y =  cond( x <= {k},  {a}*x + {b},  {c}*x + {k}*( {a} - {c}) + {b} )  )  , initial (a 1 b 1 c 1 k 1) vce(bootstrap)
```

Bootstrap result

Bootstrap provides a more robust result of the estimators.
Standard errors are more precise, and confidence interval more correct.

Moreover we can see, that the parameter b now has a significant contribution with a smaller confidence interval than in the standard procedure.

```
Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
Nonlinear regression                              Number of obs =        55
                                                  R-squared     =    0.9405
                                                  Adj R-squared =    0.9370
                                                  Root MSE      =  1.235116
                                                  Res. dev.     =  175.1584
Bootstrap results
------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          /k |   4.879108   .1288064    37.88   0.000     4.626652    5.131564
          /a |   3.132064   .1770563    17.69   0.000      2.78504    3.479088
          /b |   .7164454   .3099072     2.31   0.021      .1090385    1.323852
          /c |  -2.841337   .1106353   -25.68   0.000    -3.058178   -2.624496
------------------------------------------------------------------------------
* (SEs, P values, CIs, and correlations are asymptotic approximations)
  Parameter b taken as constant term in model
```

Remark

STATA 9 provides various post-testing routines and methods to achieve more robust and reliable estimators.
Also, we can process more complicated models by using own macro functions.

SAS 9.1

SAS provides the indicator function IFN(). Moreover, we can immediately write inbuild functions quite easily.
Both methods are demonstrated here.

The IFN() function

We can use the IFN-function in SAS as an indicator function.
IFN(L,a,b) is defined as:

$$\mathrm{IFN}(L,a,b) := \begin{cases} a & L \text{ is true} \\ b & L \text{ is false} \end{cases}$$

Example:
IFN(x<5, 1, 0) would give 1 for if x<5 and 0 for x>=5

One line apporach

```
proc nlin;
     parms a=1 b=1 c=1 k=1;
     model y =  ifn( x<=k,  a*x + b,  c*x + k*(a-c)+ b );
run;
```

Inbuild function approach

```
proc nlin data=SplinLin;
     parms a=1 b=1 c=1 k=1;

     if x<=k then do;
        model y=a*x + b;
     end;
     else do;
        model y=c*x + k*(a-c)+b;
     end;
run;
```

Result (for both approaches)

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	3	1230.5	410.2	268.88	<.0001
Error	51	77.8010	1.5255		
Corrected Total	54	1308.3			

The NLIN Procedure

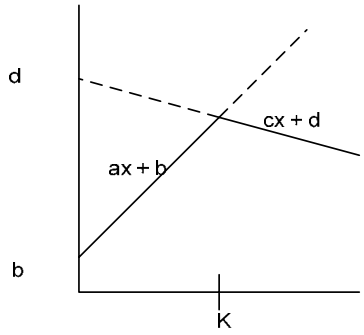
Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
a	3.1321	0.1747	2.7814	3.4827
b	0.7164	0.4279	-0.1425	1.5754
c	-2.8413	0.1320	-3.1064	-2.5763
k	4.8791	0.1165	4.6452	5.1131

Comparison

		SPSS 13 STATA 9.2 SAS 9.1	
Parameter	**Real**	**Estimate**	**Confidence Interval**
A	3	3.1	[2.8 ; 3.5]
B	1	0.7	[-0.1 ; 1.6](*)
K	5	4.9	[4.6 ; 5.1]
C	-3	-2.8	[-3.1; -2.6]

(*) H_0: B=0 could not be rejected with all three standard procedures
STATA Bootstrap rejects H_0: B=0 (p=0.02) and provides a confidence interval B ε [0.1 ; 1.3]

Proof:



$$f(x) = \begin{cases} ax + b & x \le K \\ cx + d & x > K \end{cases}$$

At K :

$$aK + b = cK + d$$

$$aK + b - cK = d$$

$$K(a - c) + b = d$$

so that :

$$f(x) = \begin{cases} ax + b & x \le K \\ cx + K(a - c) + b & x \ge K \end{cases}$$

which is :

$$f(x) = \begin{cases} ax + b & x \le K \\ c(x - K) + Ka + b & x \ge K \end{cases}$$