Subject: REGRESS Question
Posted by David Fanning on Wed, 04 Sep 2002 21:21:49 GMT
View Forum Message <> Reply to Message

Folks,

I have a client who has asked me to create a pixel density function between two images and then perform a regression analysis on the resulting distribution. No problem doing all this, but she finds that the results of my regression analysis differ from the same analysis performed in other statistics packages. In fact, three different packages give the same answer, and then there is IDL. :-(

For example, if the other packages calculate a "goodness of fit" of 0.95, IDL might report 0.97.

Here is my question. Are there any known problems with REGRESS? Or, can I assume that this problem comes from my own mathematical ignorance?

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: REGRESS Question
Posted by William Clodius on Thu, 05 Sep 2002 17:31:02 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Folks,
>
> I have a client who has asked me to create a pixel density
> function between two images and then perform a regression
> analysis on the resulting distribution. No problem doing all
> this, but she finds that the results of my regression analysis
> differ from the same analysis performed in other statistics
> packages. In fact, three different packages give the same
> answer, and then there is IDL. :-(

>
> For example, if the other packages calculate a "goodness
> of fit" of 0.95, IDL might report 0.97.
>
> Here is my question. Are there any known problems with REGRESS?
> Or, can I assume that this problem comes from my own mathematical
> ignorance?
>
> Cheers,
>
> David

<snip>

Almost any package can have problems, but the original REGRESS in Bevington has stood the test of time. IDL's version works for me, but it is possible that the introduced some problems. One thing that bothers me is tha 0.95 is to a good approximation 0.97^2. Could you be fitting the square root of the customer's data.

Subject: Re: REGRESS Question
Posted by julia[1] on Thu, 05 Sep 2002 18:31:55 GMT
View Forum Message <> Reply to Message

David --

I ran into problems with the regress routine a few years ago, trying to regress large amounts of data. The problem is that regress.pro calls the total routine, which is called in floating point precision. I had more obvious problems than the 2% difference in goodness of fit that your reporting, but I found I had to modify regress.pro to call total in double precision.

Julia

David Fanning <david@dfanning.com> wrote in message news:<MPG.17e027561a59e932989994@news.frii.com>...
> Folks,
>
> I have a client who has asked me to create a pixel density
> function between two images and then perform a regression
> analysis on the resulting distribution. No problem doing all
> this, but she finds that the results of my regression analysis
> differ from the same analysis performed in other statistics
> packages. In fact, three different packages give the same

> answer, and then there is IDL. :-(
>
> For example, if the other packages calculate a "goodness
> of fit" of 0.95, IDL might report 0.97.
>
> Here is my question. Are there any known problems with REGRESS?
> Or, can I assume that this problem comes from my own mathematical
> ignorance?
>
> Cheers,
>
> David

Subject: Re: REGRESS Question
Posted by Mike Alport on Fri, 06 Sep 2002 07:26:14 GMT
View Forum Message <> Reply to Message

I think Bill may have a point - either R or R^2 is sometimes used as a measure of "Goodness of Fit". One way to check this would be to compare this quantity from both programs to eg 5 dec places and see if one is the SQRT of the other.
Mike

"Bill" <wclodius@lanl.gov> wrote in message
news:3D7794C1.17DD18AB@lanl.gov...
>
>
> David Fanning wrote:
>
>> Folks,
>>
>> I have a client who has asked me to create a pixel density
>> function between two images and then perform a regression
>> analysis on the resulting distribution. No problem doing all
>> this, but she finds that the results of my regression analysis
>> differ from the same analysis performed in other statistics
>> packages. In fact, three different packages give the same
>> answer, and then there is IDL. :-(
>>
>> For example, if the other packages calculate a "goodness
>> of fit" of 0.95, IDL might report 0.97.
>>
>> Here is my question. Are there any known problems with REGRESS?
>> Or, can I assume that this problem comes from my own mathematical
>> ignorance?
>>
>> Cheers,

>>
>> David
>
> <snip>
>
> Almost any package can have problems, but the original REGRESS in
> Bevington has stood the test of time. IDL's version works for me, but it
> is possible that the introduced some problems. One thing that bothers
> me is tha 0.95 is to a good approximation 0.97^2. Could you be fitting
> the square root of the customer's data.
>

Subject: Re: REGRESS Question
Posted by Chris Lee on Wed, 22 Oct 2003 10:44:50 GMT
View Forum Message <> Reply to Message

In article <932b9720.0310210627.f93c6f2@posting.google.com>, "Kevin M. Lausten" <kevinlausten@hotmail.com> wrote:

> I am having difficulty working with the REGRESS function. I continually
> get values <1 for my slope when doing a regression between two vectors.
> When I do a regression mapping y to x (slope = regress(x, y, const =
> const)) and when I do a regression mapping x to y (slope = regress(y, x,
> const = const) I get a slope<1 for both calculations. Shouldn't the
> y=mx+b of these two regressions be inverses of each other (leading to
> one slope>1, and one<1?) Maybe I am misunderstanding regressions?
> Thanks,
> kevin

Hi,

If you try the regression with the simplest possible straight line

y = mx +c

where m=1 and c=0 , so

y=x

if you regress with y=f(x), you get a value of 1 (and a constant of 0)
if you regress with x=f(y), you get a value of 1, again.

if the gradient is negative for y=f(x), it has to be negative for x=f(y).
The two equations you are assuming in the regressions are

y= mx + c OR x = (y-c)/m = ny + d

n=1/m, so sign is preserved. (and d=-c/m = -cn)

HTH

Chris.

Subject: Re: REGRESS Question
Posted by wmconnolley on Wed, 22 Oct 2003 13:03:57 GMT
View Forum Message <> Reply to Message

"Christopher Lee" <cl> wrote:
> "Kevin M. Lausten" <kevinlausten@hotmail.com> wrote:

>> I am having difficulty working with the REGRESS function. I continually
>> get values <1 for my slope when doing a regression between two vectors.
>> When I do a regression mapping y to x (slope = regress(x, y, const =
>> const)) and when I do a regression mapping x to y (slope = regress(y, x,
>> const = const) I get a slope<1 for both calculations. Shouldn't the
>> y=mx+b of these two regressions be inverses of each other (leading to
>> one slope>1, and one<1?) Maybe I am misunderstanding regressions?

You've certainly misunderstood some basic maths: the inverse (as in
reciprocal) of -1 is -1, not 1. If the regression of y against x
has a negative slope, then you would expect the regression of x against y
to have too.

OTOH the relation is *not* reciprocal anyway, unless the fit is perfect.
(probably because the fit is asymmetric: y values are assumed uncertain,
x values exact).

-W.

--
William M Connolley | wmc@bas.ac.uk | http://www.antarctica.ac.uk/met/wmc/
Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself
I'm a .signature virus! copy me into your .signature file & help me spread!

Subject: Re: REGRESS Question
Posted by justspam03 on Wed, 22 Oct 2003 14:01:37 GMT
View Forum Message <> Reply to Message

Hej Kevin,

you may mix up 'regress' and 'linfit' - at least your
argument rather seems to relate to the latter.
Cheers
 Oliver

Subject: Re: regress question
Posted by Wout De Nolf on Thu, 27 Nov 2008 10:47:35 GMT
View Forum Message <> Reply to Message

On Thu, 27 Nov 2008 01:23:06 -0800 (PST), russ <rlayberry@hotmail.com>
wrote:

> Hi
>
> I'm using multiple linear regression using the REGRESS function. This
> gives me
>
> y = c + a1x1 + a2 x2 ...+ anxn
>
> with the coefficents a1,a2 etc.
>
> What I want to do is the above but force the constant to be zero. ie
> find the coeffcients that give the best linear fit whilst the function
> goes through thr origin (which it should do for physical reasons).
>
> Any ideas?
>
> Thanks
>
> Russ

You can create the design-matrix yourself and then use some
factorization like LU, SVD, Cholesky, QR,... (is your linear system
over/under determined?) The example below uses SVD. First it solves a
system not going through the origin by REGRESS and then by SVD.
Finally SVD is used for a system that goes through the origin.

```
X1 = [1.0, 2.0, 4.0, 8.0, 16.0, 32.0]
X2 = [0.0, 1.0, 2.0, 3.0, 4.0, 5.0]
X = transpose([[X1],[X2]])
Y = 3*X1 - 4*X2 + 5
Yorg = 3*X1 - 4*X2

; Regress
result1=regress(X,Y,const=const)
```

```
result1=[reform(result1),const]

; SVD (concat. X with 1's for the const)
SVDC, [X,replicate(1,1,n_elements(Y))], W, U, V
result2=reform(SVSOL(U, W, V, Y))

; SVD (origin)
SVDC, X, W, U, V
result3=reform(SVSOL(U, W, V, Yorg))

print,'Regress: ',result1
print,'SVD: ',result2
print,'SVD(origin): ',result3
```

Subject: Re: regress question
Posted by Brian Larsen on Thu, 27 Nov 2008 15:37:22 GMT
View Forum Message <> Reply to Message

Russ,

this has been discussed on this newsgroup for y=mx+b before, I have turkey on the brain now and not regression but extending this idea to multiple is probably not too bad (if it turns out to be the right thing). And if this is not easy to do it is an interesting thread that is good to remind oneself of.

This is the thread: http://tinyurl.com/2bfhl9
Here's a nice summary: http://tinyurl.com/2aqlgx

Cheers,

Brian

---------- ----------
Brian Larsen
Boston University
Center for Space Physics

Subject: Re: regress question
Posted by Kenneth P. Bowman on Sat, 29 Nov 2008 15:58:54 GMT
View Forum Message <> Reply to Message

In article

<7bfe8515-07f4-4c20-ad19-e2de871e3cc7@x38g2000yqj.googlegroups.com>,
Brian Larsen <balarsen@gmail.com> wrote:

> Russ,
>
> this has been discussed on this newsgroup for y=mx+b before, I have
> turkey on the brain now and not regression but extending this idea to
> multiple is probably not too bad (if it turns out to be the right
> thing). And if this is not easy to do it is an interesting thread
> that is good to remind oneself of.
>
>
> This is the thread: http://tinyurl.com/2bfhl9
> Here's a nice summary: http://tinyurl.com/2aqlgx

Like Brian, being too lazy to work this out myself, it occurred to me that you could use MPFIT to fit a general linear function and put very tight constraints on the intercept. Because the problem is linear, it should converge almost instantaneously.

Ken Bowman

Subject: Re: regress question
Posted by mccreigh on Mon, 01 Dec 2008 07:42:38 GMT
View Forum Message <> Reply to Message

I have some vague recollection of doing this once within an IDL function. A quick look turned up this, looks promising and like something i've seen before:

Curvefit(X, Y, Weights, A [, Sigma] [, CHISQ=variable] [, /DOUBLE] [, FITA=vector] [, FUNCTION_NAME=string] [, ITER=variable] [, ITMAX=value] [, /NODERIVATIVE] [, STATUS={0 | 1 | 2}] [, TOL=value] [, YERROR=variable])
A
A vector with as many elements as the number of terms in the user-supplied function, containing the initial estimate for each parameter. On return, the vector A contains the fitted model parameters.

FITA
Set this keyword to a vector, with as many elements as A, which contains a zero for each fixed parameter, and a non-zero value for elements of A to fit. If not supplied, all parameters are taken to be non-fixed.

Subject: Re: regress question
Posted by Wout De Nolf on Mon, 01 Dec 2008 08:38:36 GMT
View Forum Message <> Reply to Message

On Sun, 30 Nov 2008 23:42:38 -0800 (PST), James McCreight
<mccreigh@gmail.com> wrote:

> I have some vague recollection of doing this once within an IDL
> function. A quick look turned up this, looks promising and like
> something i've seen before:
>
> Curvefit(X, Y, Weights, A [, Sigma] [, CHISQ=variable] [, /DOUBLE] [,
> FITA=vector] [, FUNCTION_NAME=string] [, ITER=variable] [,
> ITMAX=value] [, /NODERIVATIVE] [, STATUS={0 | 1 | 2}] [, TOL=value] [,
> YERROR=variable])
> A
> A vector with as many elements as the number of terms in the user-
> supplied function, containing the initial estimate for each parameter.
> On return, the vector A contains the fitted model parameters.
>
> FITA
> Set this keyword to a vector, with as many elements as A, which
> contains a zero for each fixed parameter, and a non-zero value for
> elements of A to fit. If not supplied, all parameters are taken to be
> non-fixed.

Why using a non-linear least squares fitting algorithm for a linear
problem? Fixing parameters is not all that difficult using the linear
algorithms (i.e. orthogonal decomposition methods like SVD), although
you have to do it yourself.

Suppose y=a.x1+b.x2+c then you find the least squares solution by (X1
and X2 column vectors)
SVDC, [X1,X2,replicate(1,1,n_elements(X1))], W, U, V
result=SVSOL(U, W, V, Y) ; gives LSSol. [a,b,c]

Suppose I want to fix b=3 then you would do this
SVDC, [X1,replicate(1,1,n_elements(X1))], W, U, V
result=SVSOL(U, W, V, Y-3*X2); gives LSSol. [a,c]