All About Linear Regression

D.G. Simpson, Ph.D.
Department of Physical Sciences and Engineering
Prince George's Community College

February 1, 2010

1 Introduction

Linear regression is a method for calculating the equation of the "best" straight line that passes through a set of points. By "best," we mean the "best fit" straight line—the one that passes as closely as possible to as many points as possible.

To calculate the equation of this best fit straight line, you begin with a set of N data points, (x_i, y_i), for $i = 1, 2, 3, \ldots, N$. You then calculate:

Sum of x_i: $\displaystyle\sum_{i=1}^{N} x_i$

Sum of y_i: $\displaystyle\sum_{i=1}^{N} y_i$

Sum of squares of x_i: $\displaystyle\sum_{i=1}^{N} x_i^2$

Sum of products $x_i y_i$: $\displaystyle\sum_{i=1}^{N} x_i y_i$

The equation of the "best fit" straight line will then have slope

$$m = \frac{N \displaystyle\sum_{i=1}^{N} x_i y_i - \left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} y_i\right)}{N \displaystyle\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}, \tag{1}$$

and y-intercept

$$b = \frac{\left(\sum_{i=1}^{N} y_i\right)\left(\sum_{i=1}^{N} x_i^2\right) - \left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} x_i y_i\right)}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}. \quad (2)$$

1.1 Example

Suppose we're given the following data, and wish to find the best-fit straight line through the points.

i	x_i	y_i
1	-3	-15
2	1	1
3	5	17
4	8	29
5	12	45

We begin by calculating the squares of the x_i's and the products $x_i y_i$ for each i:

i	x_i	y_i	x_i^2	$x_i y_i$
1	-3	-15	9	45
2	1	1	1	1
3	5	17	25	85
4	8	29	64	232
5	12	45	144	540

We then find the sums of each column:

i	x_i	y_i	x_i^2	$x_i y_i$
1	-3	-15	9	45
2	1	1	1	1
3	5	17	25	85
4	8	29	64	232
5	12	45	144	540
Sums	23	77	243	903

So that we have

$$\sum_{i=1}^{N} x_i = 23$$

$$\sum_{i=1}^{N} y_i = 77$$

$$\sum_{i=1}^{N} x_i^2 = 243$$

$$\sum_{i=1}^{N} x_i y_i = 903$$

(Note that $N = 5$, since we have five data points.) Substituting these into Eqn. (1) and (2), we get

$$m = \frac{(5 \times 903) - (23 \times 77)}{(5 \times 243) - 23^2} = 4$$

$$b = \frac{(77 \times 243) - (23 \times 903)}{(5 \times 243) - 23^2} = -3$$

The equation of the straight line that best fits these points is thus

$$y = 4x - 3 \tag{3}$$

You should work through this example carefully to be certain that you understand how this works.

2 Linear Regression on the Calculator

Many modern scientific calculators have linear regression analysis built in to their software. One enters data points into the calculator one at at time, and the calculator keeps track of the sums and performs the necessary calculations for linear regression.

This section will give instructions for performing linear regression calculations on specific calculators.

2.1 TI-83 Plus Calculator

Detailed instructions for peforming linear regression on the TI-82 and TI-83 calculators is available in the booklet *Data Handling and Analysis on the TI-82 and TI-83/83 Plus Graphing Calculators*. This booklet is available from the Physical Sciences Department office, or online at:

`http://academic.pgcc.edu/psc/TI83_booklet.pdf`

A summary is presented here.

To enter your data (x, y pairs), press **[STAT][ENTER]** to access the list editor. If any data is present in lists L1 or L2, you should clear it out by moving the cursor to the cursor over the label L1 or L2 (on the first line of the screen) and pressing **[CLEAR][ENTER]**. Then enter your x values into column L1 and y values into column L2.

Once your x and y data is entered into columns L1 and L2, press **[STAT]**, then press the right arrow to select the `CALC` menu, then press **[4]** to select `LinReg(ax+b)`. Press **[ENTER]** to perform the linear regression. The results will be displayed on the screen.

If you use this type of calculator, try entering the data in the previous example. At the end, the calculator should display

```
LinReg
 y=ax+b
 a=4
 b=-3
```

2.2 HP-35S Calculator

To perform a linear regression calculation on the HP-35S calculator, begin by clearing the statistical registers: **[g][CLEAR][Σ]**. You then enter each data (x_i, y_i) pair by pressing: y_i **[ENTER]** x_i **[$\Sigma+$]**. Once all data pairs are entered, press **[f][L.R.]**; select **m** for the slope, and **b** for the y-intercept.

2.3 HP-50g Calculator

Linear regression on the HP-50g is described in the HP-50g tutorial *Curve Fitting*, available for download at:

`http://h20331.www2.hp.com/Hpsub/downloads/50gCurve_Fitting.pdf`

3 Computer Programs for Linear Regression

Source code for computer programs to perform linear regression calculations is available in numerous programming languages on the course Web page; see the course Web page at

`http://www.pgccphy.net/Linreg/linreg.html`

for source code. A binary file is available for download there that will run on an IBM PC in a DOS window.

3.1 Fortran

Listed in Appendix A is a simple Fortran-77 program for linear regression calculations. When run, the program asks for (x_i, y_i) pairs; enter each pair on one line, separated by spaces. When done, enter `END`, and the program will display the slope and y-intercept of the best-fit straight line.

3.2 C

Listed in Appendix B is a C program that does the same thing as the Fortran program in the previous section. Enter each pair on one line, separated by spaces. When done, enter `end`, and the program will display the slope and y-intercept of the best-fit straight line.

4 Nonlinear Curve Fits

In general, nonlinear curve fitting is a complicated process. However, in some cases a nonlinear equation can be "linearized," and the above processes for linear curve fitting applied to the resulting equation.

4.1 Quadratic Fits

For example, suppose we know that the data we wish to fit should obey a quadratic (second-degree polynomial) equation of the form

$$y = Ax^2 + B, \tag{4}$$

where A and B are constants to be determined. Although this is not a linear equation, it can be easily *made* to be linear by making the simple substitution $u \equiv x^2$. We then do a linear regression fit to the equation $y = Au + B$.

4.1.1 Example: Quadratic Fit

As an example of quadratic curve fitting, suppose you are given the following data:

i	x_i	y_i
1	−3	28
2	1	4
3	−7	148
4	8	193
5	12	433

We wish to fit this to an equation of the form $y = Ax^2 + B$. We begin by defining $u \equiv x^2$, and list a column of u_i that are the squares of the numbers in the x_i column:

i	x_i	u_i	y_i
1	−3	9	28
2	1	1	4
3	−7	49	148
4	8	64	193
5	12	144	433

We then find squares of the u_i and the products $u_i y_i$:

i	x_i	u_i	y_i	u_i^2	$u_i y_i$
1	-3	9	28	81	252
2	1	1	4	1	4
3	-7	49	148	2401	7252
4	8	64	193	4096	12352
5	12	144	433	20736	62352

We then find the sums of the last four columns:

i	x_i	u_i	y_i	u_i^2	$u_i y_i$
1	-3	9	28	81	252
2	1	1	4	1	4
3	-7	49	148	2401	7252
4	8	64	193	4096	12352
5	12	144	433	20736	62352
Sums		267	806	27315	82212

Using these values in Eqs. (1) and (2), we find the equation of the best fit straight line:

$$A = \frac{5 \times 82212 - 267 \times 806}{5 \times 27315 - 267^2} = 3$$

$$B = \frac{806 \times 27315 - 267 \times 82212}{5 \times 27315 - 267^2} = 1$$

So the best-fit line is $y = 3x^2 + 1$.

4.2 Power Law Fits

As another case, suppose we wish to fit data to a "power law" equation of the form

$$y = Ax^C, \tag{5}$$

where A and C are constants to be determined. This equation can be linearized by taking logarithms of both sides:

$$\log y = \log A + C \log x \tag{6}$$

This is the equation of a straight line whose intependent variable is $\log x$, whose dependent variable is $\log y$, whose slope is C, and whose intercept is $\log A$.

4.2.1 Example: Power Law Fit

As an example of power law curve fitting, suppose you are given the following data:

i	x_i	y_i
1	-3	-135
2	1	5
3	-7	-1715
4	8	2560
5	12	8640

We begin by taking logarithms of the x and y values:

i	x_i	y_i	$\log_{10}(x_i)$	$\log_{10}(y_i)$
1	3	135	0.47712	2.13033
2	1	5	0	0.69897
3	7	1715	0.84510	3.23426
4	8	2560	0.90309	3.40824
5	12	8640	1.07918	3.93651

We then find the squares of the $log_{10}(x_i)$ and the products $(\log_{10} x_i)(\log_{10} y_i)$:

i	x_i	y_i	$\log_{10}(x_i)$	$\log_{10}(y_i)$	$[\log_{10}(x_i)]^2$	$(\log_{10} x_i)(\log_{10} y_i)$
1	3	135	0.47712	2.13033	0.22764	1.01643
2	1	5	0	0.69897	0	0
3	7	1715	0.84510	3.23426	0.71419	2.73327
4	8	2560	0.90309	3.40824	0.81557	3.07795
5	12	8640	1.07918	3.93651	1.16463	4.24821

We then find the sums of the last four columns:

i	x_i	y_i	$\log_{10}(x_i)$	$\log_{10}(y_i)$	$[\log_{10}(x_i)]^2$	$(\log_{10} x_i)(\log_{10} y_i)$
1	3	135	0.47712	2.13033	0.22764	1.01643
2	1	5	0	0.69897	0	0
3	7	1715	0.84510	3.23426	0.71419	2.73327
4	8	2560	0.90309	3.40824	0.81557	3.07795
5	12	8640	1.07918	3.93651	1.16463	4.24821
Sums			3.30449	13.40832	2.92204	11.07586

Using these values in Eqs. (1) and (2), we can find the equation of the straight line that best fits the logarithms of the x and y values:

$$C = \frac{(5 \times 11.07586) - (3.30449 \times 13.40832)}{(5 \times 2.92204) - 3.30449^2} = 3$$

$$\log_{10} A = \frac{(13.40832 \times 2.92204) - (3.30449 \times 11.07586)}{(5 \times 2.92204) - 3.304492^2} = 0.69897$$

Since $log_{10} A = 0.69897$, we have $A = 10^{0.69897} = 5$. Since we have now found that $C = 3$ and $A = 5$, the equation of the best-fit line is

$$y = 5x^3 \tag{7}$$

5 The Correlation Coefficient r

The computer programs given here (and some calculators) will calculate a *correlation coefficient* (r) when you perform a linear regression. It is given by

$$r = \frac{\sum_{i=1}^{N} x_i y_i - \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)\left(\sum_{i=1}^{N} y_i\right)}{\sqrt{\left[\sum_{i=1}^{N} x_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)^2\right]\left[\sum_{i=1}^{N} y_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} y_i\right)^2\right]}} \tag{8}$$

This is just a number which gives you an idea of how closely your curve fit fits the data. Values of r close to $+1$ or -1 indicate a good fit; values of r near 0 indicate a poor fit. The sign of r is of no real significance; it simply indicates the sign of the slope of the linear regression line. Some authors use r^2 instead of r to represent how well the line fits the data, so that the sign is discarded.

6 Appendix A.
Fortran Linear Regression Program

```
C ***************************************************************************
C
C                              L I N R E G
C
C  PROGRAM:      LINREG
C
C  PROGRAMMER:   DR. DAVID G. SIMPSON
C                DEPARTMENT OF PHYSICAL SCIENCE
C                PRINCE GEORGE'S COMMUNITY COLLEGE
C                LARGO, MARYLAND  20774
C
C  DATE:         JANUARY 21, 2002
C
C  LANGUAGE:     ANSI STANDARD FORTRAN-77
C
C  DESCRIPTION:  THIS PROGRAM PERFORMS A LINEAR REGRESSION ANALYSIS FOR
C                A SET OF DATA GIVEN AS (X,Y) PAIRS.  THE OUTPUT FROM
C                THE PROGRAM IS THE SLOPE AND Y-INTERCEPT OF THE LEAST-
C                SQUARES BEST FIT STRAIGHT LINE THROUGH THE DATA POINTS.
C
C ***************************************************************************


C ***************************************************************************
C  MAIN PROGRAM
C ***************************************************************************

      PROGRAM LINREG

C ***************************************************************************
C  VARIABLE DECLARATIONS
C
C  B        Y-INTERCEPT OF LEAST-SQUARES BEST FIT LINE
C  M        SLOPE OF LEAST-SQUARES BEST FIT LINE
C  N        NUMBER OF DATA POINTS
C  R        CORRELATION COEFFICIENT
C  STR      INPUT STRING
C  SUMX     SUM OF X
C  SUMX2    SUM OF X**2
C  SUMXY    SUM OF X * Y
C  SUMY     SUM OF Y
C  SUMY2    SUM OF Y**2
C  X        INPUT X DATA
C  Y        INPUT Y DATA
C ***************************************************************************

      DOUBLE PRECISION  B
      DOUBLE PRECISION  M
      DOUBLE PRECISION  N
      DOUBLE PRECISION  R
      CHARACTER         STR*80
      DOUBLE PRECISION  SUMX
      DOUBLE PRECISION  SUMX2
      DOUBLE PRECISION  SUMXY
      DOUBLE PRECISION  SUMY
      DOUBLE PRECISION  SUMY2
      DOUBLE PRECISION  X
      DOUBLE PRECISION  Y

C
C     PRINT INTRODUCTORY MESSAGE
C

      WRITE (UNIT=*, FMT='(A)') ' LINREG - PERFORM LINEAR REGRESSION'
      WRITE (UNIT=*, FMT='(A/)') '   (ENTER ''END'' TO STOP DATA '//
     $      'ENTRY AND COMPUTE LINEAR REGRESSION.)'

C
C     INITIALIZE SUMS
C

      N = 0.0D0
      SUMX  = 0.0D0
      SUMX2 = 0.0D0
      SUMXY = 0.0D0
      SUMY  = 0.0D0
      SUMY2 = 0.0D0

C
C     START OF DATA ENTRY LOOP
C

   10 WRITE (UNIT=*, FMT='(A)') ' ENTER X Y:  '
      READ (UNIT=*, FMT='(A)') STR
      IF ((STR .EQ. 'END') .OR. (STR .EQ. 'end')) GO TO 20
```

```
      READ (UNIT=STR, FMT=*) X, Y

C
C     COMPUTE SUMS
C

      N = N + 1.0D0
      SUMX  = SUMX + X
      SUMX2 = SUMX2 + X * X
      SUMXY = SUMXY + X * Y
      SUMY  = SUMY + Y
      SUMY2 = SUMY2 + Y * Y
      GO TO 10

C
C     COMPUTE SLOPE (M), Y-INTERCEPT (B), AND CORRELATION
C     COEFFICIENT (R)
C

   20 M = (N * SUMXY  -  SUMX * SUMY) / (N * SUMX2 - SUMX**2)
      B = (SUMY * SUMX2  -  SUMX * SUMXY) / (N * SUMX2  -  SUMX**2)
      R = (SUMXY - SUMX * SUMY / N) /
     $               SQRT((SUMX2 - SUMX**2/N) * (SUMY2 - SUMY**2/N))

C
C     PRINT RESULTS
C

      WRITE (UNIT=*, FMT='(/A,1P,E15.6)') ' SLOPE        M = ', M
      WRITE (UNIT=*, FMT='(A, 1P,E15.6)') ' Y-INTERCEPT  B = ', B
      WRITE (UNIT=*, FMT='(A, 1P,E15.6)') ' CORRELATION  R = ', R

      STOP
      END
```

7 Appendix B.
C Linear Regression Program

```
/*************************************************************************/
/*                                                                       */
/*                          L I N R E G                                  */
/*                                                                       */
/*  Program:      LINREG                                                 */
/*                                                                       */
/*  Programmer:   Dr. David G. Simpson                                   */
/*                Department of Physical Science                         */
/*                Prince George's Community College                      */
/*                Largo, Maryland  20774                                 */
/*                                                                       */
/*  Date:         January 21, 2002                                       */
/*                                                                       */
/*  Language:     ANSI Standard C                                        */
/*                                                                       */
/*  Description:  This program performs a linear regression analysis for a   */
/*                set of data given as (x,y) pairs.  The output from the     */
/*                program is the slope and y-intercept of the least-squares  */
/*                best fit straight line through the data points.            */
/*                                                                       */
/*************************************************************************/


/*************************************************************************/
/*  #includes                                                            */
/*************************************************************************/

#include <stdio.h>                      /* standard i/o                  */
#include <string.h>                     /* string functions              */
#include <math.h>                       /* math functions                */


/*************************************************************************/
/*  function prototypes                                                  */
/*************************************************************************/

void   chop (char *str);                /* remove \n from end of string  */
double sqr (double x);                  /* compute the square of a number*/


/*************************************************************************/
/*  global variables                                                     */
/*************************************************************************/

double   b;                             /* y-intercept of best fit line  */
double   m;                             /* slope of best fit line        */
double   n = 0.0;                       /* number of data points         */
double   r;                             /* correlation coefficient       */
char     str[81];                       /* input string                  */
double   sumx = 0.0;                    /* sum of x                      */
double   sumx2 = 0.0;                   /* sum of x**2                   */
double   sumxy = 0.0;                   /* sum of x * y                  */
double   sumy = 0.0;                    /* sum of y                      */
double   sumy2 = 0.0;                   /* sum of y**2                   */
double   x;                             /* input x data                  */
double   y;                             /* input y data                  */



/*************************************************************************/
/*  main()                                                               */
/*************************************************************************/

int main (void)
{

/*-----------------------------------------------------------------------*/
/*  Print introductory message.                                          */
/*-----------------------------------------------------------------------*/

   printf("LINREG - Perform linear regression\n");
   printf("  (Enter END to stop data entry and "
          "compute linear regression.)\n\n");

/*-----------------------------------------------------------------------*/
/*  Enter data and accumulate sums.                                      */
/*-----------------------------------------------------------------------*/

   for (;;)                             /* loop for all data points      */
      {
      printf ("Enter x y:  ");          /* prompt for x and y            */
```

```
      fgets (str, 80, stdin);              /* read x and y into string str  */
      chop (str);                          /* remove trailing \n            */
      if (!strcmp(str,"end") ||            /* if no more data..             */
          !strcmp(str,"END"))
         break;                            /* ..then go compute least sqrs  */
      sscanf (str, "%lf %lf", &x, &y);     /* else read data from string    */

      n += 1.0;                            /* increment num of data points  */
      sumx  += x;                          /* compute sum of x              */
      sumx2 += x * x;                      /* compute sum of x**2           */
      sumxy += x * y;                      /* compute sum of x * y          */
      sumy  += y;                          /* compute sum of y              */
      sumy2 += y * y;                      /* compute sum of y**2           */
      }                                    /* loop again for more data      */

/*---------------------------------------------------------------------------*/
/*  Compute least-squares best fit straight line.                            */
/*---------------------------------------------------------------------------*/

   m = (n * sumxy  -  sumx * sumy) /       /* compute slope                 */
       (n * sumx2 - sqr(sumx));

   b = (sumy * sumx2  -  sumx * sumxy) /   /* compute y-intercept           */
       (n * sumx2  -  sqr(sumx));

   r = (sumxy - sumx * sumy / n) /         /* compute correlation coeff     */
            sqrt((sumx2 - sqr(sumx)/n) *
            (sumy2 - sqr(sumy)/n));

/*---------------------------------------------------------------------------*/
/*  Print results and return to operating system.                            */
/*---------------------------------------------------------------------------*/

   printf("\nSlope        m = %15.6e\n",m);
   printf("y-intercept  b = %15.6e\n",b);
   printf("Correlation  r = %15.6e\n",r);

   return 0;                               /* return to operating system    */

}



/*****************************************************************************/
/*  chop()                                                                   */
/*                                                                           */
/*  Remove final \n from end of string.                                      */
/*****************************************************************************/

void chop (char *str)
{
   int  len;                               /* length of str (incl \n)       */

   len = strlen (str);                     /* get length of str incl \n     */
   if (str[len-1] == '\n')                 /* if final char is \n ..        */
      str[len-1] = '\0';                   /* ..then remove it              */
}




/*****************************************************************************/
/*  sqr()                                                                    */
/*                                                                           */
/*  Returns the square of a number.                                          */
/*****************************************************************************/

double sqr (double x)
{
   return x * x;                           /* compute square of argument    */
}
```