Name: _

1 Objective: Least squares curve fitting

A procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the errors (or offsets or residuals) of the points from the curve. Review the lecture notes.

2 Key Words

`proc reg`, noisy data, regression,

3 Lab Steps

[A] Data step 1: Build 31 data points `(x, y)` and some noise `z`, so that

[1] `x` goes from 0 to 3 in steps of size 1/10,

[2] `y = 5 + 3*x - 4*x^2 + x^3` [`y` is a cubic in `x`]

[3] `z` = iid normal mean 0 variance 1 noise.

[B] Add noise: It is fairly amazing that a computer program can be given any four of the data points `(x, y)`, do some calculations and then report that the points all lie on the cubic of [A2]. What is even more amazing is that if the computer is given NOISY DATA : `(x, y + a*z)`, `"a"` not too large then it can strip off the noise and still come close to finding the cubic [A2]. In addition, it can guess the magnitude of the scale factor `"a"`. How is this possible? The computer uses method of LEAST SQUARES.

[C] Data step 2: Build 31 data points `(x, x^2, x^3, w)`, `w:= y + a*z`, where columns are labeled `x1 x2 x3 w`. Do this in one of two ways:

[1] edit [A]

[2] preferred: write a macro with input: `seed a`.

[D] PROC reg step: Now use `proc reg data = step2;` to check that claim [B] is true. Here is the code:

```
PROC reg data = step2;
        MODEL w = x1;                  /* fit line, output predicted values yhat in data set lin */
        output out = lin p = plin;      /* name of column containing predicted yhat */

        MODEL w = x1 x2;        /* fit quadratic, output predicted values yhat in data set quad */
        output out = quad p = pquad;    /* name of column containing predicted yhat */

        MODEL w = x1 x2 x3;     /* fit cubic, output predicted values yhat in data set cub */
        output out = cub p = pcub;      /* name of column containing predicted yhat */
run;
```

Merge the data, predicted values:

```
DATA all;                           /* for plotting make data set "all" which contains x1, x2, x3, w, */
                                    /* and various predicted values plin pquad pcub */
        MERGE lin quad cub;
run;
```

[1] Run `proc reg data = step2;`

[2] Look at its output. Find and list below

[a] Your value of "`a`" _________________. Root MSE : __________________ the estimate "`a`"

[b] The Parameter Estimates of

Intercept: _____________ x1: _____________ x2: ______________ x3: ______________

which should be close to the coefficients of the non-noisy cubic in [A2].

[E] Plotting the results:

Now `gplot` your results (see Lab 01 and Lab 02). For example:

```
options reset = global gunit = pct border
        ftext = swissb htitle = 4 htext = 3

        hsize = 8 in vsize = 5 in
        cback = white;

        symbol1 v = dot h=2 c = black;
        symbol2 v = circle h=2 i= join c=black;
        symbol3 v = square h=2 i = spline c = black;
        symbol4 v = triangle h=4 i = spline c = black;
run;

PROC gplot data = all;

title1 ’CUBIC␣CURVE␣FITTING’;
footnote j=1 ’curve’
             j=r ’MAT␣4672␣Lab␣07’;
plot w∗x1=1 plin∗x1=2
        pquad∗x1=3 pcub∗x1=4 / overlay

        frame

        haxis = 0 to 3 by 1
        vaxis = 2 to 7 by 1

        hminor = 3
        vminor = 3;
run;
quit;
```

Try above code using "`PROC sgplot`" command for better graphs.

[F] TURN IN: For TWO values of "`a`", in ranges [.05 to .25] and [1.0 to 1.3]:

- [1] Sketches of `gplot`, making sure that the changes in "`a`" are clear.
- [2] Hand written values found in [D2a, D2b] for each value of "`a`".
- [3] Answer the following questions in a short paragraph:
  - [a] Can least squares find the cubic in the presence of noise? Explain.
  - [b] Does its performance decrease as noise increases? Explain.
  - [c] Will a polynomial of any order necessarily fit any set of data? Explain.