Supplement of Hydrol. Earth Syst. Sci., 21, 571–588, 2017
http://www.hydrol-earth-syst-sci.net/21/571/2017/
doi:10.5194/hess-21-571-2017-supplement

Hydrology and Earth System Sciences
Open Access

Supplement of

Quantifying uncertainty on sediment loads using bootstrap confidence intervals

Johanna I. F. Slaets et al.

Correspondence to: Johanna I. F. Slaets (hanna.slaets@gmail.com)

The objective is to simulate pairs of discharge (Q) and water level (h) that result in a stage-discharge rating curve with a fixed realized R^2 (for example, 95%). We develop the theoretical framework and provide code in SAS to obtain this result.

```
/*
In the original dataset, Q is discharge and h is the water height.
Simulate a dataset of log(Q) and log(h) that:
-has the same size as the original dataset (i=1 to n_original)
-is simulated from the original stage-discharge equation (enter
parameters for logQ=a*logh + b)
-has residual errors drawn randomly for a normal distribution with any
variance, as the variance will be rescaled later
*/

data Q_simset;
do i=1 to 20;
  logh=-1.0727485+4.1546429*rannor(-1);
  logQ_perf=0.0466*logh + 0.1254;
  logQ=logQ_perf+ (0.1*rannor(-1));
      *add any variance, then rescale;
  output;
end;
run;

/*
This simulated dataset is the input dataset for the macro which
rescales the errors so we obtain the desired R-square.
The macro variables are
sim= the simulated dataset from the first step
Rsquare_desired= the required R2, for example 0.95
*/
%macro Rsquare(sim=, Rsquare_desired=);

      proc mixed data=&sim;
      model logQ1= logp1 /outpm=pred;
      run;
      quit;

      data pred;
      set pred;
      resid_sq=resid**2;
      run;

      proc univariate data=pred noprint;
      var resid resid_sq logQ1 pred;
      output out=sum_resid sum=sum_resid sum_resid_sq sumQ1 sumpred
      mean=mean_resid mean_resid_sq mean_logQ1 mean_pred;
      run;

      data sum_resid;
      set sum_resid;
      call symput('SS_resid',trim(left(sum_resid_sq)));
      call symput('mean_resid',trim(left(mean_resid)));
      call symput('mean_logQ1',trim(left(mean_logQ1)));
```

```
call symput('mean_pred',trim(left(mean_pred)));
run;

data data_sim;
set pred;
z2_pred=(pred-&mean_pred);
z2_err=(resid-&mean_resid);
run;

data data_sim;
set data_sim;
z1_calc=(pred-&mean_pred)**2;
z2_calc=z2_pred*z2_err;
z3_calc=(resid-&mean_resid)**2;
run;

proc univariate data=data_sim noprint;
var z1_calc z2_calc z3_calc;
output out=explained_error sum=sum_z1 sum_z2 sum_z3;
run;

data explained_error;
set explained_error;
z1=sum_z1;
z2=2*sum_z2;
z3=sum_z3;
run;


data explained_error;
set explained_error;
call symput('z1',trim(left(z1)));
call symput('z2',trim(left(z2)));
call symput('z3',trim(left(z3)));
run;


*solve for c: Ac2 + bc + C = 0 with
C= (Rsquare_desired-1)*z1
B= (Rsquare_desired-1)*z2
A= SSe + (Rsquare_desired-1)*z3

-> solution
x1,2= ( -B +/- Sqrt( B**2 - 4*A*C) ) / (2*A)
;

data rescale;
set explained_error;
A= &SS_resid + ((&Rsquare_desired-1)*&z3);
B= (&Rsquare_desired-1)*&z2;
C= (&Rsquare_desired-1)*&z1;
run;

data rescale;
set rescale;
D=(B**2) - (4*A*C);
run;
```

```
        data rescale;
        set rescale;
        sqrtD=sqrt(D);
        run;

        data rescale;
        set rescale;
        c1= (-B + sqrtD)/(2*A);
        c2= (-B - sqrtD)/(2*A);
        run;

        data rescale;
        set rescale;
        call symput('c1',trim(left(c1)));
        call symput('c2',trim(left(c2)));
        run;

        data data_sim;
        set data_sim;
        c1=&c1;
        c2=&c2;
        SS_error=&SS_resid;
        run;

        data data_sim;
        set data_sim;
        resid_rescaled1=c1*resid;
        resid_rescaled2=c2*resid;
        run;

        data data_sim;
        set data_sim;
        logQ1_new_1=pred+resid_rescaled1;
        logQ1_new_2=pred+resid_rescaled2;
        run;

        data data_sim;
        set data_sim;
        keep logQ1_new_1 logQ1_new_2 logp1;
        run;


%mend;

%Rsquare(sim=Q_simset, Rsquare_desired=0.95);


/*
The final dataset is called data_sim.
As there are two distinct solutions, there are two rescaled variables
that both give the desired Rsquare:
logQ_new_1 and logQ_new_2
The desired Rsquare can be demonstrated in PROC GLM
*/

proc glm data=data_sim;
```

```
model logQ_new_1=logh;
run;
quit;

proc glm data=data_sim;
model logQ_new_2=logh;
run;
```