SAS Simulation for the Connor–Davidson Resilience Scale (CD - RISC)

Purpose

This script simulates CD-RISC item responses on the 0–4 Likert scale and computes total scores for the 25-, 10-, and 2-item versions. It supports:

- Pre/post designs with a user-defined effect size on resilience.
- Optional correlated external variables (e.g., stress) for validity checks.
- Reliability summaries (Cronbach's α) and distribution diagnostics.

How the simulation works (brief)

Each subject has a latent resilience trait $\theta \sim \mathcal{N}(\mu, \sigma^2)$. Item j draws a continuous response

$$y_{ij} = a_j\,\theta_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \tau_j^2),$$

which is discretized into 5 ordered categories (0–4) via fixed cutpoints. Discriminations a_j vary modestly across items. Post–intervention shifts apply $\theta_{post} = \theta_{pre} + \Delta$, where Δ sets the standardized effect size.

Plug-and-Play SAS Code

1) Master macro: simulate data and score CD-RISC

```
%macro simulate_cdrisc(
    n               = 500,          /* sample size per arm/time point
*/
    seed            = 20250823,     /* RNG seed
*/
    mu              = 0,            /* mean latent resilience at baseline
*/
    sigma           = 1,            /* SD of latent resilience
*/
```

```
    effect_size     = 0.35,        /* standardized pre->post shift (Cohen d)*/
    design          = prepost,     /* none | prepost | twoarm_prepost
*/
    miss_rate       = 0.03,        /* MCAR missingness at item level
*/
    add_validity    = YES,         /* add negative stress, positive wellbeing
    out             = work.cdrisc /* output table
*/
  );
/*------------------------------ Item blueprints ------------------------------
/* Item counts for versions */
%let k25 = 25;
%let k10 = 10;
%let k02 = 2;

/* Index sets for CD-RISC-10 and CD-RISC-2 (using common published subsets;
   item text is not reproduced; only indices are used for scoring).
   Adjust indices if your institution uses different subsets.
*/
%let idx10 = 1 4 6 7 8 11 14 16 17 19;   /* 10 distinct items from the 25 */
%let idx02 = 1 8;                        /* 2-item form (adapt/bounce back)*

/* Fixed cutpoints (0..4) applied to continuous y:
   category 0: (-inf, -1.5], 1: (-1.5,-0.5], 2: (-0.5,0.5], 3: (0.5,1.5], 4
data _cutpoints;
  c1 = -1.5; c2 = -0.5; c3 = 0.5; c4 = 1.5;
run;

/* Item discriminations a_j and residual scales tau_j.
   We vary a_j modestly to create realistic item-total spread. */
data _items;
  call streaminit(&seed);
  do j = 1 to &k25;
    a  = rand("uniform")*0.6 + 0.6; /* a_j in [0.6,1.2] */
    t  = sqrt( max(0.25, 1.0 - a*a) ); /* residual SD; floor to avoid too-lo
    output;
  end;
run;

/* Helper: create long design frame depending on requested design */
data _design;
  length arm $8 time $6;
  if upcase("&design") = "NONE" then do;
    arm = "single"; time="single";
    do id = 1 to &n; output; end;
```

```
  end;
  else if upcase("&design") = "PREPOST" then do;
    arm = "single";
    do id = 1 to &n;
      time="pre";  output;
      time="post"; output;
    end;
  end;
  else if upcase("&design") = "TWOARM_PREPOST" then do;
    do arm = "control","treat";
      do id = 1 to &n;
        time="pre";  output;
        time="post"; output;
      end;
    end;
  end;
  else do;
    put "ERROR: Unknown design=&design";
    stop;
  end;
run;

/* Simulate latent theta by arm/time, applying effect_size where appropriate
proc sql noprint;
  select c1, c2, c3, c4 into :c1,:c2,:c3,:c4 from _cutpoints;
quit;

data _sim_long;
  merge _design;
  by id;
  call streaminit(&seed);
  length version $8;
  /* baseline latent */
  theta_base = rand("normal", &mu, &sigma);
  /* apply design-specific shifts */
  delta = 0;
  select (upcase("&design"));
    when ("NONE")           delta = 0;
    when ("PREPOST")        delta = (time="post")*&effect_size;
    when ("TWOARM_PREPOST") delta = (time="post" and arm="treat")*&effect_s
    otherwise delta = 0;
  end;
  theta = theta_base + delta;

  /* simulate 25 items -> discretize -> missingness */
```

```
array itemc[&k25] 8.; /* continuous */
array item [&k25] 8.; /* 0..4 categorical */
/* load item params */
if _n_ = 1 then do;
  dcl hash H(dataset:"_items"); H.defineKey("j"); H.defineData("a","t"); H
end;
do j=1 to &k25;
  rc=H.find();
  /* continuous response */
  itemc[j] = a*theta + rand("normal", 0, t);
  /* discretize */
  if      itemc[j] <= &c1 then item[j]=0;
  else if itemc[j] <= &c2 then item[j]=1;
  else if itemc[j] <= &c3 then item[j]=2;
  else if itemc[j] <= &c4 then item[j]=3;
  else                         item[j]=4;
  /* MCAR missingness */
  if rand("uniform") < &miss_rate then item[j]=.;
end;

/* ---- Totals by version ---- */
/* CD-RISC-25 */
total25 = 0; nobs25=0;
do j=1 to &k25; if item[j] ne . then do; total25+item[j]; nobs25+1; end; e
range25 = 4*&k25; /* 0..100 */
score25 = total25; /* 0..100 */

/* CD-RISC-10 subset */
total10 = 0; nobs10=0;
%local i val;
%let i=1;
%do %while(%scan(&idx10,&i) ne );
  %let val=%scan(&idx10,&i);
  if item[&val] ne . then do; total10+item[&val]; nobs10+1; end;
  %let i=%eval(&i+1);
%end;
range10 = 4*&k10; /* 0..40 */
score10 = total10;

/* CD-RISC-2 subset */
total02 = 0; nobs02=0;
%let i=1;
%do %while(%scan(&idx02,&i) ne );
  %let val=%scan(&idx02,&i);
  if item[&val] ne . then do; total02+item[&val]; nobs02+1; end;
```

```
    %let i=%eval(&i+1);
  %end;
  range02 = 4*&k02; /* 0..8 */
  score02 = total02;

  /* Optional validity variables: stress (neg), wellbeing (pos) */
  %if %upcase(&add_validity)=YES %then %do;
    stress     = rand("normal", 0, 1) - 0.60*theta;     /* higher theta -> l
    wellbeing  = rand("normal", 0, 1) + 0.55*theta;     /* higher theta -> l
*/
  %end;

  /* Keep wide item responses too */
  %do j=1 %to &k25; item&j = item[&j]; %end;

  /* labels */
  label
    theta    = "Latent resilience"
    score25  = "CD-RISC-25 total (0-100)"
    score10  = "CD-RISC-10 total (0-40)"
    score02  = "CD-RISC-2 total (0-8)"
    nobs25   = "# observed items (25)"
    nobs10   = "# observed items (10)"
    nobs02   = "# observed items (2)"
    stress   = "External stress (lower is better)"
    wellbeing= "External wellbeing (higher is better)"
  ;
  output;
  drop j rc a t itemc:;
run;

/* Output */
data &out; set _sim_long; run;
%mend simulate_cdrisc;
```

2) Example: simple cross-sectional simulation

```
/* One-wave sample, moderate reliability, small missingness */
%simulate_cdrisc(n=600, design=none, miss_rate=0.02, effect_size=0, out=work

/* Distribution summaries */
proc means data=work.cdrisc1 n mean std min p25 median p75 max;
  var score25 score10 score02;
run;
```

```
/* Reliability (Cronbach's alpha) for CD-RISC-25 */
proc corr data=work.cdrisc1 alpha nomiss;
  var item1-item25;
run;

/* Convergent validity checks */
proc corr data=work.cdrisc1;
  var score25 score10 score02 stress wellbeing;
run;
```

3) Example: pre/post single-arm with improvement

```
%simulate_cdrisc(n=300, design=prepost, effect_size=0.45, out=work.cdrisc2)

/* Wide-to-long for plotting means by time */
proc means data=work.cdrisc2 n mean std;
  class time;
  var score25 score10 score02;
run;

/* Paired t-test for pre vs post (25-item) */
proc sort data=work.cdrisc2; by id time; run;
data prepost;
  merge work.cdrisc2(where=(time="pre")  rename=(score25=pre25))
        work.cdrisc2(where=(time="post") rename=(score25=post25));
  by id;
run;

proc ttest data=prepost;
  paired pre25*post25;
run;
```

4) Example: two-arm pre/post randomized design

```
%simulate_cdrisc(n=250, design=twoarm_prepost, effect_size=0.35, out=work.c

/* Check group x time means */
proc means data=work.cdrisc3 mean std;
  class arm time;
  var score25;
run;
```

```
/* Difference-in-differences using PROC GLM */
proc glm data=work.cdrisc3;
  class arm time;
  model score25 = arm|time;
  lsmeans arm*time / pdiff cl;
run; quit;
```

5) Notes and tuning tips

- **Item difficulty**: Adjust cutpoints `c1--c4` to shift endorsement rates.
- **Reliability**: Increase discriminations a_j or decrease residual SD τ_j to raise internal consistency.
- **Missingness**: `miss_rate` currently MCAR; extend to MAR by making missingness a function of θ if desired.
- **Subsets**: If your CD-RISC-10/2 use different item indices, update `%let idx10` and `%let idx02`.
- **Scaling**: Raw totals already match common score ranges (0–100, 0–40, 0–8). Apply rescaling only if you change item counts.

Output tables you will get

- Descriptive stats for each version's total score.
- Cronbach's α for the 25-item pool.
- Correlations with external validity variables (`stress`, `wellbeing`).
- For pre/post: paired tests and/or group×time contrasts.

Attribution

This code simulates anonymized item responses consistent with the CD-RISC structure without reproducing proprietary item content. For the official scales and permissions, visit the CD-RISC website.