

Relevant SAS Code for Creating IPIP Sub-Scales and Scale Totals

```
/* Raw data is in a long-format with one row per response, where qnum = question number and  
qresponse = response value */
```

```
Data ipip1;  
Set raw.ipip  
sf_num = qnum*1;  
if qresponse ne "Not Answered" then numresp = qresponse*1;  
run;
```

```
proc sort data=ipip;  
by sf_num;  
run;
```

```
/* Sheet 1 of the IPIP question groupings document indicates which questions require reverse scoring. It  
also assigns each question one of the 5 IPIP letters (A, E, C, O, N) and a sub-scale number (subnum) */
```

```
PROC IMPORT OUT= WORK.revscorelist  
    DATAFILE= "Y:\IPIP question groupings.xlsx"  
    DBMS=XLSX REPLACE;  
    SHEET="SHEET1";  
RUN;
```

```
proc sort data=revscorelist nodupkey;  
by sf_num;  
run;
```

```
data ipip2;  
    merge ipip1 (in=a) revscorelist (in=b);  
    by sf_num;  
    if a and b;  
  
    if reverse eq "R" then respval = 6 - numresp;  
    else respval = numresp;  
run;
```

/* Sheet 3 of the IPIP question groupings document assigns a value of 1 to 30 to each of the subscales */

```
PROC IMPORT OUT= WORK.names
    DATAFILE= "Y:\IPIP question groupings.xlsx"
    DBMS=XLSX REPLACE;
    SHEET="SHEET3";
RUN;
```

/* This macro uses imputation to assign values to the 30 IPIP sub-scales */

%macro subscales;

```
data _null_;
    set names end=last;
        call symput(trim(left('ssname')) || trim(left(_n_)),trim(left(combo_NAME))) ;
        call symput(trim(left('ssnum')) || trim(left(_n_)),trim(left(subscale_num))) ;
        if last then call symput ('last',_n_);
run;
```

```
proc sort data=ipip2;
by pid letter subnum;
run;
```

```
data ipip3;
set ipip2;
by pid letter subnum;
array lets(5) $ let1-let5;
array letnumnames(30) $ letnum1-letnum30;
```

```
let1 = "A";
let2 = "C";
let3 = "E";
let4 = "N";
let5 = "O";
```

```
do i = 1 to 5;
    if letter eq lets(i) then letnum = i;
end;
```

```
letnumcombo = (((letnum-1)*6) + subnum);
retain temp_count numnonmiss
    %do j = 1 %to &last;
        &&ssname&j
    %end;
;
```

```

if first.subnum then do;
    temp_count = 0;
    numnonmiss = 0;
end;

if respval ne . then do;
    numnonmiss = numnonmiss + 1;
    temp_count = temp_count + respval;
end;

if last.subnum then do;
    %do i = 1 %to &last;
        if letnumcombo eq &&ssnum&i then do;
            if numnonmiss eq 4 then &&ssname&i = temp_count;
            else if numnonmiss eq 3 then &&ssname&i = temp_count*4/3;
            else if numnonmiss lt 3 then &&ssname&i = .;
        end;
    %end;
end;

if last.pid then do;
    output;
    %do k = 1 %to &last;
        &&ssname&k = .;
    %end;
end;
keep pid
    %do q = 1 %to &last;
        &&ssname&q
    %end;
;

run;

%mend;

%subscals;

data ipip4;
set ipip3;
IPIP_A = IPIP_A1 + IPIP_A2 + IPIP_A3 + IPIP_A4 + IPIP_A5 + IPIP_A6;
IPIP_C = IPIP_C1 + IPIP_C2 + IPIP_C3 + IPIP_C4 + IPIP_C5 + IPIP_C6;
IPIP_E = IPIP_E1 + IPIP_E2 + IPIP_E3 + IPIP_E4 + IPIP_E5 + IPIP_E6;
IPIP_N = IPIP_N1 + IPIP_N2 + IPIP_N3 + IPIP_N4 + IPIP_N5 + IPIP_N6;
IPIP_O = IPIP_O1 + IPIP_O2 + IPIP_O3 + IPIP_O4 + IPIP_O5 + IPIP_O6;
run;

```