SESUG Paper 181-2019

Using SAS® Hash and Hiter Objects to Compute the Ability Levels that Correspond to the Rasch Model's Response Probabilities

Imelda C. Go, Questar Assessment, Inc.

ABSTRACT

This paper illustrates how to use hash and hiter objects to determine the ability level that corresponds to an item's particular response probability (RP) under the Rasch Model. As an example, the Rasch model's RP50 (i.e., RP of 0.50) is identical to the item's difficulty parameter.

This paper does not provide details of hash and hiter object basics, but papers by Paul Dorfman and other authors on the topic are highly recommended.

Let us suppose you have a data set named *sample* with item *ID* and Rasch model *a* and *b* parameters as variables. (The *a* parameter (a.k.a. item discrimination parameter) is the slope of the item characteristic curve (ICC). The *b* parameter is the item difficulty parameter.)

sample data set

id	a	b
A	1	3.975
B	2	4
C	0.5	3

The response probability (RP) of 0.50 was chosen as the example for this paper so that it would be easy to validate the results. For the Rasch model, the ability level that corresponds to an item's RP50 is the item difficulty. By definition, RP50 is the person ability level (i.e., theta) where the probability of the person answering the item correctly is 0.50.

The code on the next page shows you the use of hash and hiter objects to illustrate computationally that the RP50 for the Rasch model is the *b* parameter value. The macro is written in such a way that the target response probability of 0.50 is a macro parameter. If RP67 is of interest then the macro can be invoked by using 0.67 as a parameter.

To keep the example simple, the example uses a theta range of [-6, 6]. Because all the *b* values in the example are in that range, we know that this theta range will work. In practice this range may need to be adjusted depending on target RP, and the *a* and *b* parameter values.

A more efficient variation of this program when the target RP is not 0.50 is based on the ICC being monotonically increasing. For example, we know that the RP at the item difficulty level is 0.50. Therefore, RP67 will be an ability level that is greater than the item difficulty and that theta values less than the item difficulty cannot produce an RP of 0.67 because the ICC is a monotonically increasing curve. By using this property of the ICC, we can use a smaller theta range in the DO loop, which will reduce the program's run time.

```
%macro rp(dsn,prob);
	data _null_;
		if 0 then
				set sample nobs=n;
		call symputx('nrows',n);
		stop;
	run;

	%do i=1 %to &nrows;

		data record;
			set sample;
			if _n_=&i;
			call symputx('id',id);
			call symputx('a',a);
			call symputx('b',b);

		data rasch;
			rpref=&prob;
			id="&id";
			do theta=-6 to 6 by .00001;
				rp=exp(&a*(theta-&b))/(1+exp(&a*(theta-&b)));
				diff=abs((rp-rpref));
				if rp<&prob then
					output;
				if rp>=&prob then
					stop;
			end;
		run;

		data values;
			if _n_=1 then
				do;
					if 0 then
						set rasch (keep=id diff theta rp);
					declare hash calcs(dataset:'rasch',ordered:'ascending');
					calcs.definekey('diff','theta');
					calcs.definedata('id','diff', 'theta','rp');
					calcs.definedone();
					declare hiter c('calcs');
				end;
			c.first();
			output values;
		run;

		data values; set values; a=&a; b=&b;

		proc append base=rpvalues data=values;
	%end;
%mend;

proc datasets lib=work;
	delete rpvalues;

%rp(sample,.5);
```

PROGRAMMING CODE EXPLAINED

SAS PROGRAMMING STATEMENTS — **DESCRIPTION**

```
%macro rp(dsn,prob);
```

The RP macro has two parameters: the dataset name (*dsn*) and response probability of interest (*prob*).

```
    data _null_;
        if 0 then
                set sample nobs=n;
        call symputx('nrows',n);
        stop;
    run;
```

This code puts the number of observations in data set *sample* into the macro variable *nrows*.

```
    %do i=1 %to &nrows;
```

The hash and hiter objects will be created for each item in the sample data set by using this %DO loop.

```
    data record;
        set sample;
        if _n_=&i;
        call symputx('id',id);
        call symputx('a',a);
        call symputx('b',b);
```

The *record* data set keeps the ith record (isolated by the condition `if _n_=&i`) in the sample data set.

The *id*, *a*, and *b* values are assigned to macro variables with the same name.

```
    data rasch;
            id="&id";
```

The data set *rasch* will be created for each item. The *ID* variable is given the ith record's ID value.

```
        do theta=-6 to 6 by .00001;
```

A DO loop with the *theta* (ability) values from -6 to 6 is created. (This range of theta values can be increased, if necessary.)

```
        rp=exp(&a*(theta-&b))/(1+exp(&a*(theta-&b)));
                diff=abs((rp-&prob));
```

In this DO loop, the RP is calculated given the *a*, *b*, and specific *theta* values.

The *diff* value is the difference between the calculated *rp* value and *prob* value.

```
                if rp<&prob then
                    output;
                if rp>=&prob then
                    stop;
            end;
        run;
```

We check that the RP value is still less than the *prob* value before we output the record to the data set.

If the *rp* value meets or exceeds the *prob* value, then we do not need that record and subsequent rp values generated by subsequent theta values in the DO loop. Hence, the loop processing is stopped.

```
        data values;
        if _n_=1 then
            do;
            if 0 then set rasch (keep=id diff theta rp);
```

If it is the first record in the data set, then do the following.

This next statement is a way to initialize the variables needed for defining the hash object. The variables take on the attributes of the variables in data set *rasch*.

In lieu of this statement, the following can be used instead:

```
call missing( id, diff,theta,rp);
```

```
            declare hash calcs(dataset:'rasch',ordered:'ascending');
```

This defines the hash object named *calcs.* It uses the variables in the *rasch* data set with the data set rows ordered in ascending key order.

```
                calcs.definekey('diff','theta');
```

The keys for the *calcs* hash object are the *diff* and *theta* variables.

```
               calcs.definedata('id','diff', 'theta','rp');
```

The *calcs* hash object will contain the *id, diff, theta,* and *rp* variables as values.

```
               calcs.definedone();
```

This statement means the definition for the hash object *calcs* is done/complete.

```
               declare hiter c('calcs');
         end;
```

The hiter object named *c* is based on the *calcs* hash object.

```
         c.first();
         output values;
      run;
```

The hiter value of interest is the first record of the hiter object *c.* That record is output to data set *values.*

```
      data values; set values; a=&a; b=&b;
```

These statements add the *a* and *b* parameter values to this data set using variables with the same name;

```
      proc append base=rpvalues data=values;
```

The data set *rpvalues* will be the accumulation of the data sets (named *values*) produced from each record in the *sample* data set.

```
   %end;
%mend;
```

This marks the end of the macro %DO loop.

```
proc datasets lib=work;
   delete rpvalues;
```

This makes sure the *rpvalues* data set is empty before it accumulates the desired records.

```
%rp(sample,.5);
```

This macro invocation generates the RP50 values using the items and Rasch model parameters in the *sample* data set.

The *rpvalues* data set is:

id	theta	rp	diff	a	b
A	3.975	0.5	1.01011E-11	1	3.975
B	4	0.5	2.01203E-11	2	4
C	3	0.5	5.84899E-12	0.5	3

The *diff* values show how close the computed *rp* value is to the target RP value of 0.50. The example illustrates that the RP50 value is the item difficulty parameter.

CONCLUSION

There are a number of different ways to do anything in SAS. Hash and hiter objects are powerful tools in one's SAS programming toolkit.

CONTACT INFORMATION

Imelda C. Go, Ph.D.
igo@questarai.com
Working remotely from Columbia, SC

TRADEMARK NOTICE