A Macro-Based Approach for Calculating Binomial Probabilities

Daniel R. Bretheim, William M. Mercer, Inc.

I. INTRODUCTION

Calculating the binomial probabilities from the formula

\[ P(r) = \binom{n}{r} p^r q^{(n-r)} \]

would be quite time consuming if \( n \) were very large. While comprehensive tables are available, their use is cumbersome and not well suited to computer-based Monte Carlo simulations that use the binomial distribution.

This paper outlines a SAS System macro-based approach for use in calculating binomial probabilities. The sections that follow describe the methodology, code, and sample output related to a macro designed to accomplish this task.

II. THE BINOMIAL PROBABILITY FORMULA

In general, the probability of \( r \) successes in \( n \) trials is:

\[ \frac{n!}{r!(n-r)!} p^r q^{(n-r)} \]

where \( r \) is the number of successes; \( n \) is the size of the sample; \( p \) is the probability of a success; \( q = (1 - p) \) is the probability of a failure; and \( P(r) \) = probability of exactly \( r \) successes.

Using the bent coin toss example (Spurr & Bonini, "Statistical Analysis For Business Decisions", Irwin, 1973), we can determine the probability of getting 3 heads and 2 tails using a bent coin that turns up heads 60 percent of the time.

\[ n = 5 \text{ flips, the sample size} \]
\[ r = 3 \text{ heads, the number of successes} \]
\[ n - r = 2 \text{ tails, the number of failures} \]
\[ p = .6, \text{ the probability of a head (success)} \]
\[ q = 1 - p = .4, \text{ the probability of a tail (failure)} \]

\[ P(r) = \frac{5!}{3!2!} (.6)^3(.4)^2 = 10*.034 = .34 \]

Carrying out this calculation for all possible outcomes yields the following results:

- Probability of 0 heads = \( P(0) = .01 \)
- Probability of 1 heads = \( P(1) = .08 \)
- Probability of 2 heads = \( P(2) = .23 \)
- Probability of 3 heads = \( P(3) = .34 \)
- Probability of 4 heads = \( P(4) = .26 \)
- Probability of 5 heads = \( P(5) = .08 \)

Total = 1.00

III. CAN THESE CALCULATIONS BE "AUTOMATED"?

Since the formula for \( P(r) \) defines a whole family of distributions of \( r \), one for each combination of the values of \( n \) and \( p \), an easy-to-use method for generating these distributions would be a handy tool for applications that use the binomial distribution.

The SAS System provides a starting point for computing binomial distributions through use of the PROBBNML function. This function uses values supplied in three parameters to return the probability that an observation from a binomial distribution with parameters \( n \) and \( p \) is less than or equal to a specified outcome \( r \). Using the PROBBNML function, we could compute probability values from a binomial distribution and, with additional programming, generate a cumulative probability distribution as well. However, because I wanted the flexibility of adding additional terms to the basic formula, I opted to develop code to replicate the PROBBNML function. While less elegant than simply using the function, the approach outlined below does provide the flexibility that I desired. (An example of why such flexibility is valuable is provided in Appendix 2).

IV. A MACRO-BASED APPROACH

In this section, each piece of a macro named DIST will be built and described. Examples of output produced by the macro are also displayed.

Step 1: Compute Factorials

The first section of code computes \( n! \).

```sas
%macro dist(n,p);

* Step 1: Compute factorials;
array f{*} facto-fact&s (1);
do i = 1 to &n;
    f{i+1} = f{i}*(i);
end;
%mend dist;
```

As the macro processor executes the macro with \( n=5 \), SAS sees

```
ARRAY F{*} FACT0-FACT5 (1);
DO I = 1 TO 5;
    F[I+1] = F[I]*(I);
END;
```

The array statement shows six array elements named FACT0 through FACT5, where FACT0 will be assigned an
initial value of 1. The array statement uses an asterisk as the array subscript value. This indicates that the SAS System is to determine the subscript by counting the variables in the array, thereby making the array statement flexible for any value of n. The DO loop will increment five times computing 5 factorial (5!) and assigning each successive value to FACT1 through FACTS. The values for FACTO through FACT5 are produced as follows:

**Factorial**

- FACTO = 1
- FACT1 = 1
- FACT2 = 2
- FACT3 = 6
- FACT4 = 24
- FACT5 = 120

Note: Since \( n! = \text{GAMMA}(n+1) \), the GAMMA function could also be used to achieve the same result.

Step 2: Compute Probabilities

The following section of code is added to the macro to compute the probability of each possible outcome, given the values of n and p.

```sas
* Step 2: Compute probabilities;
array p{*} pO-p&n;
doi = 1 to &n+1:
p{i} = (fact&n/(f{i} "f{&n+2-i}))*(&p"*(i-1))*((1-
\_p)**(&n+1-i));
end;
```

As the macro processor executes the macro with \( p=.6 \) and \( n=5 \), SAS sees

```sas
ARRAY P{*} PO-P5;
DO I = 1 TO 5+1;
P{I} = (FACT5*f{I} "f{5+2-I})*(.6"*(9-1))*((1-
.6)**(5+1-I));
END;
```

The DO loop will execute the formula the required number of times. For each value of \( n \), there are \( n+1 \) possible outcomes. For example, if \( n=5 \), the possible outcomes are 0, 1, 2, 3, 4, or 5. Therefore, the formula will be executed 6 times, i.e., from 1 to (5 + 1). The values for P0 through P5 are produced as follows:

<table>
<thead>
<tr>
<th>Prob.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>0.01024</td>
</tr>
<tr>
<td>P1</td>
<td>0.07680</td>
</tr>
<tr>
<td>P2</td>
<td>0.23040</td>
</tr>
<tr>
<td>P3</td>
<td>0.34560</td>
</tr>
<tr>
<td>P4</td>
<td>0.25920</td>
</tr>
<tr>
<td>P5</td>
<td>0.07776</td>
</tr>
</tbody>
</table>

Step 3: Compute Cumulative Probabilities

Finally, adding the last section of code computes the cumulative probability distribution.

```sas
* Step 3: Compute probability distribution;
retain t;
array c{*} cO-c&n;
doi = 1 to &n+1:
t + p{i};
c{i} = t;
cp = c{i};
pp = p{i};
outcome = i+1;
output;
end;
```

As the macro processor executes the macro with \( n=5 \) and \( p=.6 \), SAS sees

```sas
RETAIN T;
ARRAY C{*} CO-CS;
DO I = 1 TO 5+1;
T + P{I};
C{i} = T;
CP = C{i};
PP = P{I};
OUTCOME = I+1;
OUTPUT;
END;
```

Three new variables are created (CP, PP, and OUTCOME) simply for use in the PROC PRINT that follows the macro (see Appendix 1). The values for C0 through C5 are produced as follows:

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C0</td>
</tr>
<tr>
<td>C1</td>
</tr>
<tr>
<td>C2</td>
</tr>
<tr>
<td>C3</td>
</tr>
<tr>
<td>C4</td>
</tr>
<tr>
<td>C5</td>
</tr>
</tbody>
</table>

Step 4: Invoking the Macro

Invoke the macro from within a DATA step by specifying values for the macro parameters. For example, values for the macro parameters of \( n=5 \) and \( p=.6 \) produce the distribution displayed in Exhibit 1.

**EXHIBIT 1**

\( (n=5 \text{ and } p=.6) \)

<table>
<thead>
<tr>
<th>Probability Distribution of Possible Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outcome</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

Values for the macro parameters of \( n=50 \) and \( p=.5 \) produce the distribution displayed in Exhibit 2.
EXHIBIT 2

(n = 50 and p = .5)

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Probability</th>
<th>Cumulative Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0000</td>
<td>0.000000000000001</td>
</tr>
<tr>
<td>1</td>
<td>0.000000000000044</td>
<td>0.000000000000045</td>
</tr>
<tr>
<td>2</td>
<td>0.000000000001088</td>
<td>0.000000000001133</td>
</tr>
<tr>
<td>3</td>
<td>0.000000000017408</td>
<td>0.000000000018542</td>
</tr>
<tr>
<td>4</td>
<td>0.000000000204547</td>
<td>0.000000000223089</td>
</tr>
<tr>
<td>5</td>
<td>0.000000000881837</td>
<td>0.000000001104337</td>
</tr>
<tr>
<td>6</td>
<td>0.000000014113775</td>
<td>0.000000015538112</td>
</tr>
<tr>
<td>7</td>
<td>0.000000088715169</td>
<td>0.000000108033336</td>
</tr>
<tr>
<td>8</td>
<td>0.000000476844031</td>
<td>0.000000584888067</td>
</tr>
<tr>
<td>9</td>
<td>0.00001225212144</td>
<td>0.000002307005047</td>
</tr>
<tr>
<td>10</td>
<td>0.00009126157932</td>
<td>0.00001930656388</td>
</tr>
<tr>
<td>11</td>
<td>0.00033176784697</td>
<td>0.000451074509636</td>
</tr>
<tr>
<td>12</td>
<td>0.00010245050266</td>
<td>0.000152932000802</td>
</tr>
<tr>
<td>13</td>
<td>0.000315179454624</td>
<td>0.000468111454426</td>
</tr>
<tr>
<td>14</td>
<td>0.00083397427395</td>
<td>0.001095085728361</td>
</tr>
<tr>
<td>15</td>
<td>0.001899312653544</td>
<td>0.003294326880405</td>
</tr>
<tr>
<td>16</td>
<td>0.004373114932910</td>
<td>0.007673338916315</td>
</tr>
<tr>
<td>17</td>
<td>0.011546229856819</td>
<td>0.019219607036833</td>
</tr>
<tr>
<td>18</td>
<td>0.027864975500200</td>
<td>0.047084552525521</td>
</tr>
<tr>
<td>19</td>
<td>0.069798746465053</td>
<td>0.116118160178772</td>
</tr>
<tr>
<td>20</td>
<td>0.1788265700670390</td>
<td>0.328938284651646</td>
</tr>
<tr>
<td>21</td>
<td>0.335961686035319</td>
<td>0.564904231713171</td>
</tr>
<tr>
<td>22</td>
<td>0.5501379668737709</td>
<td>0.773377854839863</td>
</tr>
<tr>
<td>23</td>
<td>1.000000000000000</td>
<td>1.000000000000000</td>
</tr>
</tbody>
</table>

V. SUMMARY

The ability to dynamically create a probability distribution based on varying values for p and n adds considerable flexibility to data-driven applications. The macro will accommodate any value and compute the appropriate probabilities and cumulative distribution. The availability of a cumulative probability distribution is particularly useful in applications that use Monte Carlo simulation techniques.

For further information, please contact:

Dan Bretheim
William M. Mercer, Inc.
1417 Lake Cook Road
Deerfield, Illinois 60015
(708) 317-7723

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ®indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The complete macro and all related code used to produce the output above is displayed in Appendix 1. Appendix 2 displays an example of how the basic concepts used to create the DIST macro can be expanded to accommodate more complex formulas, such as the one displayed below.

\[
\sum_{t=0}^{N_{\text{c}}} \binom{N_{\text{c}}}{t} p^t (1-p)^{N_{\text{c}}-t} \text{ for } N_{\text{c}}=50, p=0.5
\]

For further information, please contact:

Dan Bretheim
William M. Mercer, Inc.
1417 Lake Cook Road
Deerfield, Illinois 60015
(708) 317-7723

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ®indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.
%macro dist(n,p);
* Step 1: Compute factorials ;
  array f[*] fact0-fact&n (1);
  do i = 1 to &n;
    f[i+1] = f[i]*i;
  end;
* Step 2: Compute probabilities ;
  array p[*] p0-p&n;
  do i = 1 to &n+1;
    p[i] = (fact&n!*(fact(i)*fact(&n+2-i))*((1-p)**(i-1))*p**((1-p)**(i-n+1)));  
  end;
* Step 3: Compute probability distribution :
  retain t;
  array c[*] c0-c&n;
  do i = 1 to &n+1;
    t + p[i];
    c[i]=t;
    cp=c[i];
    pp=p[i];
    outcome = i-1;
  end;
  output;
%mend dist;
* Enter values for two macro variables .
* The first value represents the size of the sample .
* The second value represents the probability of a success .
data dist;
%dist(5,.6);
run;

proc print data = dist split = ' ' noobs;
  var outcome pp cp;
  format cp pp 16.15;
  sum pp;
  label outcome = 'Outcome'
    pp = 'Probability'
    cp = 'Cumulative Probability'
  ;
  title 'Probability Distribution of Possible Outcomes';
run;
APPENDIX 2

- Program Name : TEST4.SAS
- Author : D.Bretheim
- Creation Data : 08-19-93
- Last Update :
- Description : Step 1 - Compute cumulative probability dist.;

Step 2 - Assign random numbers to events. :

Step 3 - Simulate outcomes. :

Step 4: cumulative probability distribution:
array c(*) c0-c&end;
array pd(*) pd0-pd&end;
retain t;
do i = (1 + &x) to finish:
pd(i) = pr(i)/denom;
t + pd(i);
c(i) = t;
pp = pd(i);
outcome = i-1;
output;
end;
finetot = sum(of pd0-pd&end);
call symput('nchild',left(&x));

%macro dist (end,p,n,x,function);
  * Step 1: compute factorials :
  array W{1} nactO-Iact&end (1);
do i = 1 to &end;
    I{i}={i}*(1);
  end;
  * Step 2: compute probabilities:
  array p{'} p0-p&end;
  finish = &end + 1;
do i = (1 +&x) to finish;
    p{i} = (I{i}/lf{i-1}**(n+x)**(i-1-x)***(1-x)*
      (1-2**n)**x);
  end;
  tot = sum(of p0-p&end);
  * Step 3: apply function:
  array fu{'} f0-f&end;
  %if &function=yes %then %do;
    set function;
    %end;
  %else %do;
    do i = 1 to finish;
      fu{i}=1;
    end;
  %end;
  array pr{'} pr0-pr&end;
do i = (1 +&x) to finish;
  pr{i} = p{i}*fu{i};
  end;
  denom = sum(of pr0-pr&end);

  %step 4: cumulative probability distribution:
  array c(*) c0-c&end;
  array pd(*) pd0-pd&end;
  retain t;
do i = (1 + &x) to finish:
pd(i) = pr(i)/denom;
t + pd(i);
c(i) = t;
cp = c(i);
pp = pd(i);
outcome = i-1;
output;
end;
finetot = sum(of pd0-pd&end);
call symput('nchild',left(&x));

%mend dist;

* enter values for five macro variables.
* the first value represents the number of potential addl.
  children.;
* the second value represents the probability of a
  covered child not incurring a claim in one month.;
* the third value represents the number of months of
  exposure. ;
* the fourth value represents the number of children
  with claims. ;
* the fifth value represents whether a function is used.

data dat. dist:
%dist(1 0 .. 78,2,0,yes);
%dist(10,.78,15,3,yes);
%dist(1 0 .. 78,5,0,yes);
rn;

proc print data=dat.dist split='*' noobs;
var outcome pp cp;
format pp cp 16.15;
sum pp;
label outcome = 'Outcome'
  pp = 'Probability'
  cp = 'Cumulative Probability'
; title1 'Probability Distribution of Possible Outcomes';
title2 'PD(X) For the Total Number of Children';
title3 'Where X Represents the Number of Children With Claims';
title4 "In This Example, X = &nchild";
title5 'Outcome Represents the Total Number of Children';
data one;
set dat.dist end = last;
if last;
* create cumulative probability distribution
array c(*) c0-c10;
do i = 1 to finish:
  c{i} = c{i-1}+
  if i < = (finish - &nchild) then do;
  c{i} = c{i-1}+
  end;
else c{i} = 1;

801
end;

* generate a random number between 0 and 1 to
  * determine then number of addl. children.
  prob = ranuni(0);

* assign random numbers to events and
  * determine outcomes.
  if prob = c10 then extchild = 10;
  else
    if prob > = c9 then extchild = 10;
    else
      if prob > = c8 then extchild = 9;
      else
        if prob > = c7 then extchild = 8;
        else
          if prob > = c6 then extchild = 7;
          else
            if prob > = c5 then extchild = 6;
            else
              if prob > = c4 then extchild = 5;
              else
                if prob > = c3 then extchild = 4;
                else
                  if prob > = c2 then extchild = 3;
                  else
                    if prob > = c1 then extchild = 2;
                    else
                      if prob > = c0 then extchild = 1;
                      else
                        extchild = 0;
                        run;

proc print data=one;
  var c0-c10
  prob
  extchild;
  title1 'Cumulative Distribution For the Number of Children';
  title2 'That Did Not File a Claim';
  title3 "Given That &nchild Children Did File a Claim";
run;