

Appendix C

Functions

Appendix Table of Contents

INTRODUCTION	1841
FUNCTION DESCRIPTIONS	1842
AOQ2 Function	1842
ASN2 Function	1843
ATI2 Function	1845
BAYESACT Call	1846
C4 Function	1848
CUSUMARL Function	1849
D2 Function	1850
D3 Function	1851
EWMAARL Function	1852
PROBACC2 Function	1853
PROBBNML Function	1854
PROBHYPYR Function	1856
PROBMED Function	1858
STDMED Function	1859
DETAILS	1861
Types of Sampling Plans	1861
Evaluating Single-Sampling Plans	1861
Evaluating Double-Sampling Plans	1863
Deriving Control Chart Constants	1863

Appendix C

Functions

Introduction

SAS/QC software provides specialized DATA step functions for computations related to control chart analysis, for Bayes analysis of screening designs, and for sampling plan evaluation. You can use these functions in DATA step programming statements. The following lists summarize these functions:

Table C.1. Functions for Control Chart Analysis

Function	Description
C4	expected value c_4 of the standard deviation of a sample from a normal population with unit standard deviation
CUSUMARL	average run length for a cumulative sum control chart scheme
D2	expected value d_2 of the range of a sample from a normal population with unit standard deviation
D3	standard deviation d_3 of the range of a sample from a normal population with unit standard deviation
EWMAARL	average run length for an EWMA scheme
PROBMED	cumulative distribution function of sample median
STDMED	standard deviation of median of a standard normal sample

Table C.2. Function for Bayes Analysis of Screening Designs

Function	Description
BAYESACT	posterior probabilities of variance contamination

Table C.3. Functions for Sampling Plan Evaluation

Function	Description
AOQ2	average outgoing quality for double-sampling plan
ASN2	average sample number for double-sampling plan
ATI2	average total inspection for double-sampling plan
PROBACC2	acceptance probability for double-sampling plan

In addition, the PROBBNML and PROBHYPR functions, which are provided in base SAS software, are useful when evaluating single-sampling plans.

The twelve SAS/QC functions, together with the PROBBNML and PROBHYPR functions, are described in the next section, “Function Descriptions.” The “Details” section on page 1861, summarizes types of sampling plans and gives additional definitions.

Function Descriptions

This section describes the twelve SAS/QC functions and the related functions PROBBNML and PROBHYPR in alphabetical order.

AOQ2 Function

computes average outgoing quality for a double-sampling plan.

Syntax

AOQ2(*replacement*, *N*, *a*₁, *r*₁, *a*₂, *n*₁, *n*₂, *p*)

where

<i>replacement</i>	has the value 'REP' or 'NOREP', respectively, depending on whether nonconforming items are replaced with conforming items.
<i>N</i>	is the lot size, where $N \geq 2$.
<i>a</i> ₁	is the acceptance number for the first sample, where $a_1 \geq 0$.
<i>r</i> ₁	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
<i>a</i> ₂	is the acceptance number for the second sample, where $a_2 \geq a_1$.
<i>n</i> ₁	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
<i>n</i> ₂	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
<i>p</i>	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The AOQ2 function returns the average outgoing quality for a Type B double-sampling plan in which nonconforming items are replaced with conforming items (*replacement* is 'REP') or not replaced (*replacement* is 'NOREP'). For details on Type B double-sampling plans, see "Types of Sampling Plans" on page 1861.

For replacement, the average outgoing quality is

$$\text{AOQ} = \frac{pP_{a_1}(N - n_1) + pP_{a_2}(N - n_1 - n_2)}{N}$$

and for no replacement, the average outgoing quality is

$$\text{AOQ} = \frac{pP_{a_1}(N - n_1)}{N - n_1p} + \frac{pP_{a_2}(N - n_1 - n_2)}{N - n_1p - n_2p}$$

where, in both situations,

$$P_{a_1} = \sum_{d=0}^{a_1} f(d|n)$$

= probability of acceptance for first sample

$$P_{a_2} = \sum_{d=a_1+1}^{r_1-1} f(d|n_1)F(a_2 - d|n_2)$$

= probability of acceptance for second sample

and

$$f(d|n) = \binom{n}{d} p^d (1-p)^{n-d}$$

= binomial probability that the number of nonconforming items in a sample of size n is exactly d

$$F(a|n) = \sum_{d=0}^a f(d|n)$$

= probability that the number of nonconforming items is less than or equal to a

Examples

The first set of statements results in a value of 0.0148099904. The second set of statements results in a value of 0.0144743043.

```
data;
  aoq=aoq2('norep',120,0,2,1,13,13,0.18);
  put aoq;
run;
```

```
data;
  aoq=aoq2('rep',120,0,2,1,13,13,0.18);
  put aoq;
run;
```

ASN2 Function

computes the average sample number for a double-sampling plan.

Syntax

ASN2(*mode*, *a*₁, *r*₁, *a*₂, *n*₁, *n*₂, *p*)

where

<i>mode</i>	identifies whether sampling is under full inspection (<i>mode</i> is 'FULL') or semicurtailed inspection (<i>mode</i> is 'SEMI').
<i>a</i> ₁	is the acceptance number for the first sample, where $a_1 \geq 0$.
<i>r</i> ₁	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
<i>a</i> ₂	is the acceptance number for the second sample, where $a_2 \geq a_1$.
<i>n</i> ₁	is the size of the first sample, where $n_1 \geq 1$.
<i>n</i> ₂	is the size of the second sample, where $n_2 \geq 1$.
<i>p</i>	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The ASN2 function returns the average sample number for a Type B double-sampling plan under full inspection (*mode* is 'FULL') or semicurtailed inspection (*mode* is 'SEMI'). For details on Type B double-sampling plans, see “Types of Sampling Plans” on page 1861.

For full inspection, the average sample number is

$$ASN = n_1 + n_2[F(r_1 - 1|n_1) - F(a_1|n_1)]$$

and for semicurtailed inspection, the average sample number is

$$ASN = n_1 + \sum_{d=a_1+1}^{r_1-1} f(d|n_1) \left(n_2 F(a_2 - d|n_2) + \frac{r_2 - d}{p} [1 - F(r_2 - d|n_2 + 1)] \right)$$

where

$$f(d|n) = \binom{n}{d} p^d (1 - p)^{n-d}$$

= binomial probability that the number of nonconforming items in a sample of size n is exactly d

$$F(a|n) = \sum_{d=0}^a f(d|n)$$

= probability that the number of nonconforming items is less than or equal to a

Examples

The first set of statements results in a value of 15.811418112. The second set of statements results in a value of 14.110408695.

```
data;
  asn=asn2('full',0,2,1,13,13,0.18);
  put asn;
run;
```

```
data;
  asn=asn2('semi',0,2,1,13,13,0.18);
  put asn;
run;
```

ATI2 Function

computes the average total inspection for a double-sampling plan.

Syntax

ATI2($N, a_1, r_1, a_2, n_1, n_2, p$)

where

N	is the lot size, where $N \geq 2$.
a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 \geq a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The ATI2 function returns the average total inspection for a Type B double-sampling plan. For details on Type B double-sampling plans, see “Types of Sampling Plans” on page 1861.

The average total inspection is

$$ATI = n_1 P_{a_1} + (n_1 + n_2) P_{a_2} + N(1 - P_{a_1} - P_{a_2})$$

where

$$P_{a_1} = \sum_{d=0}^{a_1} f(d|n)$$

= probability of acceptance for first sample

$$P_{a_2} = \sum_{d=a_1+1}^{r_1-1} f(d|n_1) F(a_2 - d|n_2)$$

= probability of acceptance for second sample

and

$$f(d|n) = \binom{n}{d} p^d (1-p)^{n-d}$$

= binomial probability that the number of nonconforming items in a sample of size n is exactly d

$$F(a|n) = \sum_{d=0}^a f(d|n)$$

= probability that the number of nonconforming items is less than or equal to a

Examples

The following statements result in a value of 110.35046381:

```
data;
  ati=ati2(120,0,2,1,13,13,0.18);
  put ati;
run;
```

BAYESACT Call

computes posterior probabilities that observations are contaminated with a larger variance.

Syntax

CALL BAYESACT(*k*, *s*, *df*, $\alpha_1, \dots, \alpha_n$, y_1, \dots, y_n , β_1, \dots, β_n , *p*₀);

where

<i>k</i>	is the contamination coefficient, where $k \geq 1$.
<i>s</i>	is an independent estimate of σ , where $s \geq 0$.
<i>df</i>	is the number of degrees of freedom for <i>s</i> , where $df \geq 0$.
α_i	is the prior probability of contamination for the <i>i</i> th observation in the sample, where $i = 1, \dots, n$ and <i>n</i> is the number of observations in the sample. Note that $0 \leq \alpha_i \leq 1$.
y_i	is the <i>i</i> th observation in the sample, where $i = 1, \dots, n$ and <i>n</i> is the number of observations in the sample. When the BAYESACT call is used to perform a Bayes analysis of designs (see “Description” below), the y_i s are estimates for effects.
β_i	is the variable that contains the returned posterior probability of contamination for the <i>i</i> th observation in the sample, where $i = 1, \dots, n$ and <i>n</i> is the number of observations in the sample.
<i>p</i> ₀	is the variable that contains the posterior probability that the sample is uncontaminated.

Description

The BAYESACT call computes posterior probabilities (β_i) that observations in a sample are *contaminated* with a larger variance than other observations and computes the posterior probability (*p*₀) that the entire sample is uncontaminated.

Specifically, the BAYESACT call assumes a normal random sample of *n* independent observations, with a mean of 0 (a centered sample) where some of the observations may have a larger variance than others:

$$\text{Var}(y_i) = \begin{cases} \sigma^2 & \text{with probability } 1 - \alpha_i \\ k^2\sigma^2 & \text{with probability } \alpha_i \end{cases}$$

where $i = 1, \dots, n$. The parameter *k* is called the *contamination coefficient*. The value of α_i is the *prior probability* of contamination for the *i*th observation. Based on the prior probability of contamination for each observation, the call gives the posterior probability of contamination for each observation and the posterior probability that the entire sample is uncontaminated.

Box and Meyer (1986) suggest computing posterior probabilities of contamination for the analysis of saturated orthogonal factorial designs. Although these designs give uncorrelated estimates for effects, the significance of effects cannot be tested in an analysis of variance since there are no degrees of freedom for error. Box and Meyer suggest computing posterior probabilities of contamination for the effect estimates. The prior probabilities (α_i) give the likelihood that an effect will be significant, and the contamination coefficient (k) gives a measure of how large the significant effect will be. Box and Meyer recommend using $\alpha = 0.2$ and $k = 10$, implying that about 1 in 5 effects will be about 10 times larger than the remaining effects. To adequately explore posterior probabilities, examine them over a range of values for prior probabilities and a range of contamination coefficients.

If an independent estimate of σ is unavailable (as is the case when the y_i s are effects from a saturated orthogonal design), use 0 for s and df in the BAYESACT call. Otherwise, the call assumes s is proportional to the square root of a χ^2 random variable with df degrees of freedom. For example, if the y_i s are estimated effects from an orthogonal design that is not saturated, then use the BAYESACT call with s equal to the estimated standard error of the estimates and df equal to the degrees of freedom for error.

From Bayes' theorem, the posterior probability that y_i is contaminated is

$$\beta_i(\sigma) = \frac{\alpha_i f(y_i; 0, k^2 \sigma^2)}{\alpha_i f(y_i; 0, k^2 \sigma^2) + (1 - \alpha_i) f(y_i; 0, \sigma^2)}$$

for a given value of σ , where $f(x; \mu, \sigma)$ is the density of a normal distribution with mean μ and variance σ^2 .

The probability that the sample is uncontaminated is

$$p = \prod_{i=1}^n (1 - \beta_i(\sigma))$$

Posterior probabilities that are independent of σ are derived by integrating $\beta_i(\sigma)$ and p over a noninformative prior for σ . If an estimate of σ is available (when $df > 0$), it is appropriately incorporated. Refer to Box and Meyer (1986) for details.

Examples

The statements

```

data;
  retain post1-post7 postnone;
  call bayesact(10,0,0,
    0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
    -5.4375,1.3875,8.2875,0.2625,1.7125,-11.4125,1.5875,
    post1, post2, post3, post4, post5, post6, post7,
    postnone);
run;

```

return the following posterior probabilities:

POST1	0.42108
POST2	0.037412
POST3	0.53438
POST4	0.024679
POST5	0.050294
POST6	0.64329
POST7	0.044408
POSTNONE	0.28621

The probability that the sample is uncontaminated is 0.28621. A situation where this BAYESACT call would be appropriate is a saturated 2^7 design in 8 runs, where the estimates for main effects are as shown in the function above (-5.4375, 1.3875, . . . , 1.5875).

C4 Function

computes the expected value of the standard deviation of n independent normal random variables.

Syntax

C4(n)

where n is the sample size, with $n \geq 2$.

Description

The C4 function returns the expected value of the standard deviation of n independent, normally distributed random variables with the same mean and with standard deviation of 1. This expected value is referred to as the control chart constant c_4 .

The value c_4 is calculated as

$$c_4 = \frac{\Gamma(\frac{n}{2})\sqrt{2/(n-1)}}{\Gamma(\frac{n-1}{2})}$$

where $\Gamma(\cdot)$ is the gamma function. As n grows, c_4 is asymptotically equal to $(4n - 4)/(4n - 3)$.

For more information, refer to the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Montgomery (1996), and Wadsworth and others (1986).

In other chapters, c_4 is written as $c_4(n)$ to emphasize the dependence on n .

You can use the constant c_4 to calculate an unbiased estimate ($\hat{\sigma}$) of the standard deviation σ of a normal distribution from the sample standard deviation of n observations:

$$\hat{\sigma} = (\text{sample standard deviation})/c_4$$

where the sample standard deviation is calculated using $n - 1$ in the denominator. In the SHEWHART procedure, c_4 is used to calculate control limits for s charts, and it is used in the estimation of the process standard deviation based on subgroup standard deviations.

Examples

The following statements result in a value of 0.939985603:

```
data;
  constant=c4(5);
  put constant;
run;
```

CUSUMARL Function

computes the average run length for a cumulative sum control chart scheme.

Syntax

CUSUMARL(*type*, δ , *h*, *k* <,*headstart*>)

where

<i>type</i>	indicates a one-sided or two-sided scheme. Valid values are 'ONESIDED' or 'O' for a one-sided scheme, and 'TWO SIDED' or 'T' for a two-sided scheme.
δ	is the shift to be detected, expressed as a multiple of the process standard deviation (σ).
<i>h</i>	is the decision interval (one-sided scheme) or the vertical distance between the origin and the upper arm of the V-mask (two-sided scheme), each time expressed as a positive value in standard units (a multiple of σ/\sqrt{n} , where <i>n</i> is the subgroup sample size).
<i>k</i>	is the reference value (one-sided scheme) or the slope of the lower arm of the V-mask (two-sided scheme), each time expressed as a positive value in standard units (a multiple of σ/\sqrt{n} , where <i>n</i> is the subgroup sample size).
<i>headstart</i>	is the headstart value (optional) expressed in standard units (a multiple of σ/\sqrt{n} , where <i>n</i> is the subgroup sample size). The default <i>headstart</i> is zero. For details, refer to Lucas and Crosier (1982).

Description

The CUSUMARL function returns the average run length of one-sided and two-sided cumulative sum schemes with parameters as described above. The notation is consistent with that used in the CUSUM procedure.

For a one-sided scheme, the average run length is calculated using the integral equation method (with 24 Gaussian points) described by Goel and Wu (1971) and Lucas and Crosier (1982).

For a two-sided scheme with no *headstart*, the average run length (ARL) is calculated using the fact that

$$(\text{ARL})^{-1} = (\text{ARL}_+)^{-1} + (\text{ARL}_-)^{-1}$$

where ARL_+ and ARL_- denote the average run lengths of the equivalent one-sided schemes for detecting a shift of the same magnitude in the positive direction and in the negative direction, respectively.

For a two-sided scheme with a nonzero *headstart*, the ARL is calculated by combining average run lengths for one-sided schemes as described in Appendix A.1 of Lucas and Crosier (1982, 204).

For a specified shift δ , you can use the CUSUMARL function to design a cusum scheme by first calculating average run lengths for a range of values of h and k and then choosing the combination of h and k that yields a desired average run length.

You can also use the CUSUMARL function to interpolate published tables of average run lengths.

Examples

The following three sets of statements result in the values 4.1500826715, 4.1500836225, and 4.1061588131, respectively.

```
data;
  arl=cusumarl('twosided',2.5,8,0.25);
  put arl;
run;
```

```
data;
  arl=cusumarl('onesided',2.5,8,0.25);
  put arl;
run;
```

```
data;
  arl=cusumarl('o',2.5,8,0.25,0.1);
  put arl;
run;
```

D2 Function

computes the expected value of the sample range.

Syntax

$D2(n)$

where n is the sample size, with $2 \leq n \leq 25$.

Description

The D2 function returns the expected value of the sample range of n independent, normally distributed random variables with the same mean and a standard deviation of 1. This expected value is referred to as the control chart constant d_2 . The values returned by the D2 function are accurate to ten decimal places.

The value d_2 can be expressed as

$$d_2 = \int_{-\infty}^{\infty} [1 - (1 - \Phi(x))^n - (\Phi(x))^n] dx$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. Refer to Tippett (1925). In other chapters, d_2 is written as $d_2(n)$ to emphasize the dependence on n .

In the SHEWHART procedure, d_2 is used to calculate control limits for r charts, and it is used in the estimation of the process standard deviation based on subgroup

ranges. Also refer to the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Kume (1985), Montgomery (1996), and Wadsworth and others (1986).

You can use the constant d_2 to calculate an unbiased estimate ($\hat{\sigma}$) of the standard deviation σ of a normal distribution from the sample range of n observations:

$$\hat{\sigma} = (\text{sample range})/d_2$$

Note that the statistical efficiency of this estimate relative to that of the sample standard deviation decreases as n increases.

Examples

The following statements result in a value of 2.3259289473:

```
data;
  constant=d2(5);
  put constant;
run;
```

D3 Function

computes the standard deviation of the range of n independent normal random variables.

Syntax

D3(n)

where n is the sample size, with $2 \leq n \leq 25$.

Description

The D3 function returns the standard deviation of the range of n independent, normally distributed random variables with the same mean and with unit standard deviation. The standard deviation returned is referred to as the control chart constant d_3 . The values returned by the D3 function are accurate to ten decimal places.

The value d_3 can be expressed as

$$d_3 = \sqrt{2 \int_{-\infty}^{\infty} \int_{-\infty}^y f(x, y) dx dy - d_2^2}$$

where

$$f(x, y) = 1 - (\Phi(y))^n - (1 - \Phi(x))^n + (\Phi(y) - \Phi(x))^n$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function and d_2 is the expected range. Refer to Tippett (1925).

In other chapters d_3 is written as $d_3(n)$ to emphasize the dependence on n .

In the SHEWHART procedure, d_3 is used to calculate control limits for r charts, and it is used in the estimation of the process standard deviation based on subgroup ranges.

For more information, refer to the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Montgomery (1996), and Wadsworth and others (1986).

You can use the constant d_3 to calculate an unbiased estimate ($\hat{\sigma}$) of the standard deviation σ_R of the range of a sample of n normally distributed observations from the sample range of n observations:

$$\hat{\sigma}_R = (\text{sample range})(d_3/d_2)$$

You can use the D2 function to calculate d_2 .

Examples

The following statements result in a value of 0.8640819411:

```
data;  
  constant=d3(5);  
  put constant;  
run;
```

EWMAARL Function

computes the average run length for an exponentially weighted moving average.

Syntax

EWMAARL(δ , r , k)

where

- δ is the shift to be detected, expressed as a multiple of the process standard deviation (σ), where $\delta \geq 0$.
- r is the weight factor for the current subgroup mean in the EWMA, where $0 < r \leq 1$. If $r = 1$, the EWMAARL function returns the average run length for a Shewhart chart for means. Refer to Wadsworth and others (1986). If $r \leq 0.05$, $k \geq 3$, and $\delta < 0.10$, the algorithm used is unstable. However, note that the EWMA behaves like a cusum when $r \rightarrow 0$, and in this case the CUSUMARL function is applicable.
- k is the multiple of σ used to define the control limits, where $k \geq 0$. Typically $k = 3$.

Description

The EWMAARL function computes the average run length for an exponentially weighted moving average (EWMA) scheme using the method of Crowder (1987a,b). The notation used in the preceding list is consistent with that used in the MACONTROL procedure.

For a specified shift δ , you can use the EWMAARL function to design an exponentially weighted moving average scheme by first calculating average run lengths for a range of values of r and k and then choosing the combination of r and k that yields a desired average run length.

Examples

The following statements specify a shift of 1σ , a weight factor of 0.25, and 3σ control limits. The EWMAARL function returns an average run length of 11.154267016.

```
data;  
  arl=ewmaarl(1.00,0.25,3.0);  
  put arl;  
run;
```

PROBACC2 Function

computes the acceptance probability for a double-sampling plan.

Syntax

PROBACC2($a_1, r_1, a_2, n_1, n_2, D, N$)

PROBACC2($a_1, r_1, a_2, n_1, n_2, p$)

where

a_1	is the acceptance number for the first sample, where $a_1 \geq 0$.
r_1	is the rejection number for the first sample, where $r_1 > a_1 + 1$.
a_2	is the acceptance number for the second sample, where $a_2 > a_1$.
n_1	is the size of the first sample, where $n_1 \geq 1$ and $n_1 + n_2 \leq N$.
n_2	is the size of the second sample, where $n_2 \geq 1$ and $n_1 + n_2 \leq N$.
D	is the number of nonconforming items in the lot, where $0 \leq D \leq N$.
N	is the lot size, where $N \geq 2$.
p	is the proportion of nonconforming items produced by the process, where $0 < p < 1$.

Description

The PROBACC2 function returns the acceptance probability for a double-sampling plan of Type A if you specify the parameters D and N , and it returns the acceptance probability for a double-sampling plan of Type B if you specify the parameter p . For details on Type A and Type B double-sampling plans, see “Types of Sampling Plans” on page 1861.

For either type of sampling plan, the acceptance probability is calculated as

$$P_{a_1} + P_{a_2}$$

where

$$\begin{aligned}
 P_{a_1} &= \sum_{d=0}^{a_1} f(d|n) \\
 &= \text{probability of acceptance for first sample} \\
 P_{a_2} &= \sum_{d=a_1+1}^{r_1-1} f(d|n_1)F(a_2 - d|n_2) \\
 &= \text{probability of acceptance for second sample}
 \end{aligned}$$

and

$$\begin{aligned}
 f(d|n) &= \binom{n}{d} p^d (1-p)^{n-d} \\
 &= \text{binomial probability that the number of nonconforming items} \\
 &\quad \text{in a sample of size } n \text{ is exactly } d
 \end{aligned}$$

$$F(a|n) = \sum_{d=0}^a f(d|n)$$

= probability that the number of nonconforming items is less than or equal to a

These probabilities are determined from either the hypergeometric distribution (Type A sampling) or the binomial distribution (Type B sampling).

Examples

The first set of statements results in a value of 0.2396723824. The second set of statements results in a value of 0.0921738126.

```
data;
  prob=probacc2(1,4,3,50,100,10,200);
  put prob;
run;

data;
  prob=probacc2(0,2,1,13,13,0.18);
  put prob;
run;
```

PROBBNML Function

computes the probability that an observation from a binomial(n, p) distribution will be less than or equal to m .

Syntax

PROBBNML(p, n, m)

where

- p is the probability of success for the binomial distribution, where $0 \leq p \leq 1$. In terms of acceptance sampling, p is the probability of selecting a nonconforming item.
- n is the number of independent Bernoulli trials in the binomial distribution, where $n \geq 1$. In terms of acceptance sampling, n is the number of items in the sample.
- m is the number of successes, where $0 \leq m \leq n$. In terms of acceptance sampling, m is the number of nonconforming items.

Description

The PROBBNML function returns the probability that an observation from a binomial distribution (with parameters n and p) is less than or equal to m . To compute the probability that an observation is equal to a given value m , compute the difference of two values for the cumulative binomial distribution.

In terms of acceptance sampling, the function returns the probability of finding m or fewer nonconforming items in a sample of n items, where the probability of a nonconforming item is p . To find the probability that the sample contains exactly m nonconforming items, compute the difference between $\text{PROBBNML}(p, n, m)$ and $\text{PROBBNML}(p, n, m - 1)$.

In addition to using the PROBBNML function to return the probability of acceptance, the function can be used in calculations for average sample number, average outgoing quality, and average total inspection in Type B single-sampling. See “Evaluating Single-Sampling Plans” on page 1861 for details.

The PROBBNML function computes

$$\sum_{j=0}^m \binom{n}{j} p^j (1-p)^{n-j}$$

where m , n , and p are defined in the preceding list.

Examples

The following statements compute the probability that an observation from a binomial distribution with $p = 0.05$ and $n = 10$ is less than or equal to 4:

```
data;
  prob=probbnml(0.05,10,4);
  put prob;
run;
```

These statements result in the value 0.9999363102. In terms of acceptance sampling, for a sample of size 10 where the probability of a nonconforming item is 0.05, the probability of finding 4 or fewer nonconforming items is 0.9999363102.

The following statements compute the probability that an observation from a binomial distribution with $p = 0.05$ and $n = 10$ is exactly 4:

```
data;
  p=probbnml(0.05,10,4) - probbnml(0.05,10,3);
  put p;
run;
```

These statements result in the value 0.0009648081.

For additional information on probability functions, refer to *SAS Language Reference: Dictionary*.

PROBHYPR Function

computes the probability that an observation from a hypergeometric distribution is less than or equal to x .

Syntax

PROBHYPR($N, K, n, x <, r >$)

where

N	is the population size for a hypergeometric distribution. In terms of acceptance sampling, N is the lot size.
K	is the number of items in the category of interest in the population. In terms of acceptance sampling, K is the number of nonconforming items in a lot.
n	is the sample size for a hypergeometric distribution. In terms of acceptance sampling, n is the sample size.
x	is the number of items from the category of interest in the sample. In terms of acceptance sampling, x is the number of nonconforming items in the sample.
r	is optional and gives the odds ratio for the extended hypergeometric distribution. For the standard hypergeometric distribution, $r = 1$; this value is the default. In acceptance sampling, typically $r = 1$.

Restrictions on items in the syntax are given in the following equations:

$$\begin{aligned}
 1 &\leq N \\
 0 &\leq K \leq N \\
 0 &\leq n \leq N \\
 \max(0, K + n - N) &\leq x \leq \min(K, n) \\
 N, K, n \text{ and } x &\text{ are integers}
 \end{aligned}$$

Description

The PROBHYPR function returns the probability that an observation from an extended hypergeometric distribution with parameters N , K and n and an odds ratio of r is less than or equal to x . The default for r is 1 and leads to the usual hypergeometric distribution.

In terms of acceptance sampling, if $r = 1$, the PROBHYPR function gives the probability of x or fewer nonconforming items in a sample of size n taken from a lot containing N items, K of which are nonconforming, when sampling is done without replacement. Typically $r = 1$ in acceptance sampling.

For example, suppose an urn contains red and white balls, and you are interested in the probability of selecting a white ball. If $r = 1$, the function returns the probability of selecting x white balls when given the population size (number of balls in the urn), sample size (number of balls taken from the urn), and number of white balls in the population (urn).

If, however, the probability of selecting a white ball differs from the probability of selecting a red ball, then $r \neq 1$. Suppose an urn contains one white ball and one red ball, and the probability of choosing the red ball is higher than the probability of choosing the white ball. This might occur if the red ball were larger than the white ball, for example. Given the probabilities of choosing a red ball and a white ball when an urn contains one of each, you calculate r and use the value in the PROBHYPR function. Returning to the case where an urn contains many balls with $r \neq 1$, the function gives the probability of selecting x white balls when given the number of balls in the urn, the number of balls taken from the urn, the number of white balls in the urn, and the relative probability of selecting a white ball or a red ball.

The PROBHYPR function is used to evaluate Type A single-sampling plans. See “Evaluating Single-Sampling Plans” on page 1861 for details.

If $r = 1$ (the default), the PROBHYPR function calculates probabilities from the usual hypergeometric distribution:

$$\Pr[X \leq x] = \sum_{i=0}^x P_i$$

where

$$P_i = \begin{cases} \frac{\binom{K}{i} \binom{N-K}{n-i}}{\binom{N}{n}} & \text{if } \max(0, K+n-N) \leq i \leq \min(K, n) \\ 0 & \text{otherwise} \end{cases}$$

The PROBHYPR function accepts values other than 1 for r , and in these cases, it calculates the probability for the extended hypergeometric distribution:

$$\Pr[X_1 \leq x | X_1 + X_2 = n] = \sum_{i=0}^x P_i$$

where

$$P_i = \begin{cases} \frac{\binom{K}{i} \binom{N-K}{n-i} r^i}{\sum_{j=0}^n \binom{K}{j} \binom{N-K}{n-j} r^j} & \text{if } \max(0, K+n-N) \leq i \leq \min(K, n) \\ 0 & \text{otherwise} \end{cases}$$

where

- X_1 is binomially distributed with parameters K and p_1 .
- X_2 is binomially distributed with parameters $N-K$ and p_2 .
- $q_1 = 1 - p_1$
- $q_2 = 1 - p_2$
- $r = (p_1 q_2) / (p_2 q_1)$

For details on the extended hypergeometric distribution, refer to Johnson and Kotz (1969).

Examples

Suppose you take a sample of size 10 (without replacement) from an urn that contains 200 balls, 50 of which are white. The remaining 150 balls are red. The following statements calculate the probability that your sample contains 2 or fewer white balls:

```
data;  
  y=probypr(200,50,10,2);  
  put y;  
run;
```

These statements result in a value of 0.5236734081. Now, suppose the probability of selecting a red ball does not equal the probability of selecting a white ball. Specifically, suppose the probability of choosing a red ball is $p_2 = 0.4$ and the probability of choosing a white ball is $p_1 = 0.2$. Calculate r as

$$r = \frac{p_1 q_2}{p_2 q_1} = \frac{(0.2)(0.6)}{(0.4)(0.8)} = 0.375$$

With $r = 0.375$, the probability of choosing 2 or fewer white balls from an urn that contains 200 balls, 50 of which are white, is calculated using the following statements:

```
data;  
  y=probypr(200,50,10,2,0.375);  
  put y;  
run;
```

These statements return a value of 0.9053936127. See “Evaluating Single-Sampling Plans” on page 1861 for another example.

For additional information on probability functions, refer to *SAS Language Reference: Dictionary*.

PROBMED Function

computes cumulative probabilities for the sample median.

Syntax

PROBMED(n, x)

where

- n is the sample size.
- x is the point of interest; that is, the PROBMED function calculates the probability that the median is less than or equal to x .

Description

The PROBMED function computes the probability that the sample median is less than or equal to x for a sample of n independent, standard normal random variables (mean 0, variance 1).

Let n represent the sample size and $X_{(i)}$ represent the i th order statistic. Then, when n is odd, the function calculates

$$\Pr[X_{((n+1)/2)} \leq x] = I_{\Phi(x)}\left(\frac{n+1}{2}, \frac{n+1}{2}\right)$$

where

$$I_p(a, b) = \frac{1}{B(a, b)} \int_0^p t^{a-1} (1-t)^{b-1} dt$$

and $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$, where $\Gamma(\cdot)$ is the gamma function. If n is even, the PROBMEDE function calculates

$$\Pr\left[\frac{X_{(n/2)} + X_{((n/2)+1)}}{2} \leq x\right] = \frac{2}{B(\frac{n}{2}, \frac{n}{2})} \int_{-\infty}^x \left\{ [1 - \Phi(u)]^{n/2} - [1 - \Phi(2x - u)]^{n/2} \right\} [\Phi(u)]^{(n/2)-1} \phi(u) du$$

where $B(n/2, n/2) = [\Gamma(n/2)]^2/\Gamma(n)$ and $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal cumulative distribution function and density function, respectively.

For more information, refer to David (1981).

Examples

The statements

```
data;
  b=probmed(5, -0.1);
  put b;
run;
```

result in a value of 0.4256380897.

STDMED Function

computes the standard deviation of a sample median.

Syntax

STDMED(n)

where n is the sample size.

Description

The STDMED function gives the standard deviation of the median of a normally distributed sample with a mean of 0 and a variance of 1. This function gives the standard error used to determine the width of the control limits for charts produced by the MCHART and MRCHART statements in PROC SHEWHART.

Let n represent the sample size and $X_{(i)}$ represent the i th order statistic. Then, when n is odd, the STDMED function calculates $\sqrt{\text{Var}(X_{((n+1)/2})}}$, where

Part 10. The CAPABILITY Procedure

$$\text{Var}(X_{((n+1)/2)}) = \frac{1}{B\left(\frac{n+1}{2}, \frac{n+1}{2}\right)} \int_{-\infty}^{\infty} x^2 [\Phi(x)]^{(n-1)/2} [1 - \Phi(x)]^{(n-1)/2} \phi(x) dx$$

where $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ and $\Gamma(\cdot)$ is the gamma function, $\Phi(\cdot)$ is the standard normal cumulative distribution function, and $\phi(\cdot)$ is the corresponding density function.

If n is even, the function calculates the square root of the following:

$$\text{Var} \left[\frac{X_{(n/2)} + X_{((n/2)+1)}}{2} \right] =$$

$$(1/4) \left[E(X_{(n/2)}^2) + E(X_{((n/2)+1)}^2) + 2E(X_{(n/2)}X_{((n/2)+1)}) \right]$$

where

$$E(X_{(n/2)}^2) = \frac{2}{B\left(\frac{n}{2}, \frac{n}{2}\right)} \int_{-\infty}^{\infty} x^2 [\Phi(x)]^{(n/2)-1} [1 - \Phi(x)]^{n/2} \phi(x) dx$$

$$E(X_{((n/2)+1)}^2) = \frac{2}{B\left(\frac{n}{2}, \frac{n}{2}\right)} \int_{-\infty}^{\infty} x^2 [\Phi(x)]^{n/2} [1 - \Phi(x)]^{(n/2)-1} \phi(x) dx$$

$$E(X_{(n/2)}X_{((n/2)+1)}) = \frac{n}{B\left(\frac{n}{2}, \frac{n}{2}\right)} \int_{-\infty}^{\infty} \int_{-\infty}^y xy [\Phi(x)]^{(n/2)-1} [1 - \Phi(y)]^{(n/2)-1} \phi(x) \phi(y) dx dy$$

For more details, refer to David (1981), Kendall and Stuart (1977, 252), and Sarhan and Greenberg (1962).

Examples

These statements use a loop to calculate the standard deviation of the median for sample sizes from 6 to 12:

```
data;
  do n=6 to 12;
    s=stdmed(n);
    put s;
    output;
  end;
run;
```

The statements produce these values:

```
0.4634033519
0.4587448763
0.410098592
0.4075552495
0.3719226208
0.3703544701
0.3428063408
```

Details

Types of Sampling Plans

In single sampling, a random sample of n items is selected from a lot of size N . If the number d of nonconforming (defective) items found in the sample is less than or equal to an acceptance number c , the lot is accepted. Otherwise, the lot is rejected.

In double sampling, a sample of size n_1 is drawn from the lot, and the number d_1 of nonconforming items is counted. If d_1 is less than or equal to an acceptance number a_1 , the lot is accepted, and if d_1 is greater than or equal to a rejection number r_1 , the lot is rejected. Otherwise, if $a_1 < d_1 < r_1$, a second sample of size n_2 is taken, and the number of nonconforming items d_2 is counted. Then if $d_1 + d_2$ is less than or equal to an acceptance number a_2 , the lot is accepted, and if $d_1 + d_2$ is greater than or equal to a rejection number $r_2 = a_2 + 1$, the lot is rejected. This notation follows that of Schilling (1982). Note that some authors, including Montgomery (1996), define the first rejection number using a strict inequality.

In *Type A sampling*, the sample is intended to represent a single, finite-sized lot, and the characteristics of the sampling plan depend on D , the number of nonconforming items in the lot, as well as N , n , and c .

In *Type B sampling*, the sample is intended to represent a series of lots (or the lot size is effectively infinite), and the characteristics of the sampling plan depend on p , the proportion of nonconforming items produced by the process, as well as n and c .

A hypergeometric model is appropriate for Type A sampling, and a binomial model is appropriate for Type B sampling.

Evaluating Single-Sampling Plans

You can use the base SAS functions PROBBNML and PROBHYP to evaluate single-sampling plans. Measures of the performance of single-sampling plans include

- the probability of acceptance P_a
- the average sample number ASN
- the average outgoing quality AOQ
- the average total inspection ATI

Probability of Acceptance

Since P_a is the probability of finding c or fewer defectives in the sample, you can calculate the acceptance probability using the function PROBHYP(N, D, n, c) for Type A sampling and the function PROBBNML(p, n, c) for Type B sampling.

For example, the following statements calculate P_a for the plan $n = 20$, $c = 1$ when sampling from a single lot of size $N = 120$ that contains $D = 22$ nonconforming items, resulting in a value of 0.0762970752:

Part 10. The CAPABILITY Procedure

```
data;
  prob=probhypr(120,22,20,1);
  put prob;
run;
```

Similarly, the following statements calculate P_a for the plan $n = 20, c = 1$ when sampling from a series of lots for which the proportion of nonconforming items is $p = 0.18$, resulting in a value of 0.1018322793:

```
data;
  prob=probbnml(0.18,20,1);
  put prob;
run;
```

Other Measures of Performance

The measures ASN, AOQ, and ATI are meaningful only for Type B sampling and can be calculated using the PROBBNML function. For reference, the following equations are provided.

Average sample number: Following the notation of Schilling (1982), let $F(c|n)$ denote the probability of finding c or fewer nonconforming items in a sample of size n . Note that $F(c|n)$ is equivalent to $\text{PROBBNML}(p, n, c)$. Then, depending on the mode of inspection, the average sample number can be expressed as shown in the following table:

Mode of Inspection	ASN
Full	n
Semicurtailed	$nF(c n) + \frac{(c+1)(1 - F(c+1 n+1))}{p}$
Fully curtailed	$\frac{(n-c)F(c n+1)}{1-p} + \frac{(c+1)(1 - F(c+1 n+1))}{p}$

Average outgoing quality can be expressed as

$$\text{AOQ} = \frac{p(N - n)F(c|n)}{N}$$

if the nonconforming items found are replaced with conforming items, and as

$$\text{AOQ} = \frac{p(N - n)F(c|n)}{N - np}$$

if the nonconforming items found are not replaced.

Average total inspection can be expressed as

$$\text{ATI} = n + (1 - F(c|n))(N - n)$$

Evaluating Double-Sampling Plans

The following list gives some measures for double-sampling plans. The formula for each measure is given in the section describing the corresponding function.

- the probability of acceptance, P_a , calculated with the PROBACC2 function
- the average sample number, ASN, calculated with the ASN2 function
- the average outgoing quality, AOQ, calculated with the AOQ2 function
- the average total inspection, ATI, calculated with the ATI2 function

Deriving Control Chart Constants

You can use the functions D2, D3, and C4 to calculate standard control chart constants that are derived from d_2 , d_3 and c_4 . For reference, the following equations for some of these constants are provided:

$$\begin{aligned}
 A_2 &= k/(d_2\sqrt{n}) \\
 A_3 &= k/(c_4\sqrt{n}) \\
 B_3 &= \max(0, 1 - (k/c_4)\sqrt{1 - c_4^2}) \\
 B_4 &= 1 + (k/c_4)\sqrt{1 - c_4^2} \\
 B_5 &= \max(0, c_4 - k\sqrt{1 - c_4^2}) \\
 B_6 &= c_4 + k\sqrt{1 - c_4^2} \\
 c_5 &= \sqrt{1 - c_4^2} \\
 D_1 &= \max(0, d_2 - kd_3) \\
 D_2 &= d_2 + kd_3 \\
 D_3 &= \max(0, 1 - kd_3/d_2) \\
 D_4 &= 1 + kd_3/d_2 \\
 E_2 &= k/d_2 \\
 E_3 &= k/c_4
 \end{aligned}$$

In the preceding equations, k is the multiple of standard error ($k = 3$ in the case of 3σ limits), and n is the subgroup sample size. The use of these control chart constants is discussed in the *ASQC Glossary and Tables for Statistical Quality Control*, the *ASTM Manual on Presentation of Data and Control Chart Analysis*, Montgomery (1996), and Wadsworth and others (1986).

Although you do not ordinarily need to calculate control chart constants when using the SHEWHART procedure, you may find the D2, D3, and C4 functions useful for creating LIMITS= data sets that contain control limits to be read by the SHEWHART procedure.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/QC[®] User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1994 pp.

SAS/QC[®] User's Guide, Version 8

Copyright © 1999 SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-493-4

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, by any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS[®] and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute in the USA and other countries. [®] indicates USA registration.

IBM[®], ACF/VTAM[®], AIX[®], APPN[®], MVS/ESA[®], OS/2[®], OS/390[®], VM/ESA[®], and VTAM[®] are registered trademarks or trademarks of International Business Machines Corporation. [®] indicates USA registration.

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

The Institute is a private company devoted to the support and further development of its software and related services.