

## 15.1: Random Selection

### Introduction

The usual treatments deal with a single random variable or a fixed, finite number of random variables, considered jointly. However, there are many common applications in which we select at random a member of a class of random variables and observe its value, or select a random number of random variables and obtain some function of those selected. This is formulated with the aid of a *counting or selecting* random variable  $N$ , which is nonnegative, integer valued. It may be independent of the class selected, or may be related in some sequential way to members of the class. We consider only the independent case. Many important problems require *optional* random variables, sometimes called *Markov times*. These involve more theory than we develop in this treatment.

Some common examples:

- Total demand of  $N$  customers—  $N$  independent of the individual demands.
- Total service time for  $N$  units—  $N$  independent of the individual service times.
- Net gain in  $N$  plays of a game—  $N$  independent of the individual gains.
- Extreme values of  $N$  random variables—  $N$  independent of the individual values.
- Random sample of size  $N$ —  $N$  is usually determined by properties of the sample observed.
- Decide when to play on the basis of past results—  $N$  dependent on past

### A useful model—random sums

As a basic model, we consider the sum of a random number of members of an iid class. In order to have a concrete interpretation to help visualize the formal patterns, we think of the demand of a random number of customers. We suppose the number of customers  $N$  is independent of the individual demands. We formulate a model to be used for a variety of applications.

A *basic sequence*  $\{X_n : 0 \leq n\}$  [Demand of  $n$  customers]

An *incremental sequence*  $\{Y_n : 0 \leq n\}$  [Individual demands]

These are related as follows:

$$X_n = \sum_{k=0}^n Y_k \text{ for } n \geq 0 \text{ and } X_n = 0 \text{ for } n < 0 \quad Y_n = X_n - X_{n-1} \text{ for all } n$$

A *counting random variable*  $N$ . If  $N = n$  then  $n$  of the  $Y_k$  are added to give the *compound demand*  $D$  (the random sum)

$$D = \sum_{k=0}^N Y_k = \sum_{k=0}^{\infty} I_{[N \geq k]} X_k = \sum_{k=0}^{\infty} I_{\{k\}}(N) X_k$$

*Note.* In some applications the counting random variable may take on the idealized value  $\infty$ . For example, in a game that is played until some specified result occurs, this may never happen, so that no finite value can be assigned to  $N$ . In such a case, it is necessary to decide what value  $X_{\infty}$  is to be assigned. For  $N$  independent of the  $Y_n$  (hence of the  $X_n$ ), we rarely need to consider this possibility.

### Independent selection from an iid incremental sequence

We assume throughout, unless specifically stated otherwise, that:

$$X_0 = Y_0 = 0$$

$\{Y_k : 1 \leq k\}$  is iid

$\{N, Y_k : 0 \leq k\}$  is an independent class

We utilize repeatedly two important propositions:

$$E[h(D)|N=n] = E[h(X_n)], n \geq 0$$

$$M_D(s) = g_N[M_Y(s)]. \text{ If the } Y_n \text{ are nonnegative integer valued, then so is } D \text{ and } g_D(s) = g_N[g_Y(s)]$$

#### DERIVATION

We utilize properties of generating functions, moment generating functions, and conditional expectation.

$$E[I_{\{n\}}(N)h(D)] = E[h(D)|N=n]P(N=n) \text{ by definition of conditional expectation, given an event, Now,}$$

$$I_{\{n\}}(N)h(D) = I_{\{n\}}(N)h(X_n) \text{ and } E[I_{\{n\}}(N)h(X_n)] = P(N=n)E[h(X_n)]. \text{ Hence}$$

$$E[h(D)|N=n]P(N=n) = P(N=n)E[h(X_n)]. \text{ Division by } P(N=n) \text{ gives the desired result.}$$

By the law of total probability (CE1b),  $M_D(s) = E[e^{sD}] = E\{E[e^{sD}|N]\}$ . By proposition 1 and the product rule for moment generating functions,

$$E[e^{sD} | N = n] = E[e^{sX_n}] = \prod_{k=1}^n E[e^{sY_k}] = M_Y^n(s)$$

Hence

$$M_D(s) = \sum_{n=0}^{\infty} M_Y^n(s) P(N = n) = g_N[M_Y(s)]$$

A parallel argument holds for  $g_D$

—□

*Remark.* The result on  $M_D$  and  $g_D$  may be developed without use of conditional expectation.

in the integer-valued case.

$$\begin{aligned} M_D(s) &= E[e^{sD}] = \sum_{k=0}^{\infty} E[I_{\{N=n\}} e^{sX_n}] = \sum_{k=0}^{\infty} P(N = n) E[e^{sX_n}] \\ &= \sum_{k=0}^{\infty} P(N = n) M_Y^n(s) = g_N[M_Y(s)] \end{aligned}$$

—□

### Example 15.1.1 A service shop

Suppose the number  $N$  of jobs brought to a service shop in a day is Poisson (8). One fourth of these are items under warranty for which no charge is made. Others fall in one of two categories. One half of the arriving jobs are charged for one hour of shop time; the remaining one fourth are charged for two hours of shop time. Thus, the individual shop hour charges  $Y_k$  have the common distribution

$$Y = [0 \ 1 \ 2] \text{ with probabilities } PY = [1/4 \ 1/2 \ 1/4]$$

Make the basic assumptions of our model. Determine  $P(D \leq 4)$ .

**Solution**

$$g_N(s) = e^{8(s-1)} g_Y(s) = \frac{1}{4}(1 + 2s + s^2)$$

According to the formula developed above,

$$g_D(s) = g_N[g_Y(s)] = \exp((8/4)(1 + 2s + s^2) - 8) = e^{4s} e^{2s^2} e^{-6}$$

Expand the exponentials in power series about the origin, multiply out to get enough terms. The result of straightforward but somewhat tedious calculations is

$$g_D(s) = e^{-6} (1 + 4s + 10s^2 + \frac{56}{3}s^3 + \frac{86}{3}s^4 + \dots)$$

Taking the coefficients of the generating function, we get

$$P(D \leq 4) \approx e^{-6} (1 + 4 + 10 + \frac{56}{3} + \frac{86}{3}) = e^{-6} \frac{187}{3} \approx 0.1545$$

### Example 15.1.2 A result on Bernoulli trials

Suppose the counting random variable  $N \sim \text{binomial}(n, p)$  and  $Y_i = I_{E_i}$ , with  $P(E_i) = p_0$ . Then

$$g_N = (q + ps)^n \text{ and } g_Y(s) = q_0 + p_0 s$$

By the basic result on random selection, we have

$$g_D(s) = g_N[g_Y(s)] = [q + p(q_0 + p_0 s)]^n = [(1 - pp_0) + pp_0 s]^n$$

so that  $D \sim \text{binomial}(n, pp_0)$ .

In the next section we establish useful m-procedures for determining the generating function  $g_D$  and the moment generating function  $M_D$  for the compound demand for simple random variables, hence for determining the complete distribution. Obviously, these will not work for all problems. It may helpful, if not entirely sufficient, in such cases to be able to determine the mean value  $E[D]$  and variance  $\text{Var}[D]$ . To this end, we establish the following expressions for the mean and variance.

### Example 15.1.3 Mean and variance of the compound demand

$$E[D] = E[N]E[Y] \text{ and } \text{Var}[D] = E[N]\text{Var}[Y] + \text{Var}[N]E^2[Y]$$

DERIVATION

$$\begin{aligned} E[D] &= E\left[\sum_{n=0}^{\infty} I_{\{N=n\}} X_n\right] = \sum_{n=0}^{\infty} P(N=n)E[X_n] \\ &= E[Y] \sum_{n=0}^{\infty} nP(N=n) = E[Y]E[N] \\ E[D^2] &= \sum_{n=0}^{\infty} P(N=n)E[X_n^2] = \sum_{n=0}^{\infty} P(N=n)\{\text{Var}[X_n] + E^2[X_n]\} \\ &= \sum_{n=0}^{\infty} P(N=n)\{n\text{Var}[Y] + n^2 E^2[Y]\} = E[N]\text{Var}[Y] + E[N^2]E^2[Y] \end{aligned}$$

Hence

$$\text{Var}[D] = E[N]\text{Var}[Y] + E[N^2]E^2[Y] - E[N]^2 E^2[Y] = E[N]\text{Var}[Y] + \text{Var}[N]E^2[Y]$$

### Example 15.1.4 Mean and variance for Example 15.1.1

$E[N] = \text{Var}[N] = 9$ . By symmetry  $E[Y] = 1$ .  $\text{Var}[Y] = 0.25(0+2+4) - 1 = 0.5$ . Hence,

$$E[D] = 8 \cdot 1 = 8, \text{Var}[D] = 8 \cdot 0.5 + 8 \cdot 1 = 12$$

### Calculations for the compound demand

We have m-procedures for performing the calculations necessary to determine the distribution for a composite demand  $D$  when the counting random variable  $N$  and the individual demands  $Y_k$  are simple random variables with not too many values. In some cases, such as for a Poisson counting random variable, we are able to approximate by a simple random variable.

*The procedure gend*

If the  $Y_i$  are nonnegative, integer valued, then so is  $D$ , and there is a generating function. We examine a strategy for computation which is implemented in the m-procedure *gend*. Suppose

$$g_N(s) = p_0 + p_1 s + p_2 s^2 + \cdots p_n s^n$$

$$g_Y(s) = \pi_0 + \pi_1 s + \pi_2 s^2 + \cdots \pi_m s^m$$

The coefficients of  $g_N$  and  $g_Y$  are the probabilities of the values of  $N$  and  $Y$ , respectively. We enter these and calculate the coefficients for powers of  $g_Y$ :

$gN = [p_0 \ p_1 \ \cdots \ p_n]$	$1 \times (n+1)$	Coefficients of $g_N$
$y = [\pi_0 \ \pi_1 \ \cdots \ \pi_m]$	$1 \times (m+1)$	Coefficients of $g_Y$
$\dots$		
$y2 = \text{conv}(y, y)$	$1 \times (2m+1)$	Coefficients of $g_Y^2$
$y3 = \text{conv}(y, y2)$	$1 \times (3m+1)$	Coefficients of $g_Y^3$
$\dots$		
$yn = \text{conv}(y, y(n-1))$	$1 \times (nm+1)$	Coefficients of $g_Y^n$

We wish to generate a matrix  $P$  whose rows contain the joint probabilities. The probabilities in the  $i$ th row consist of the coefficients for the appropriate power of  $g_Y$  multiplied by the probability  $N$  has that value. To achieve this, we need a matrix, each of whose  $n+1$  rows has  $nm+1$  elements, the length of  $yn$ . We begin by “preallocating” zeros to the rows. That is, we set  $P = \text{zeros}(n+1, n * m + 1)$ . We then replace the appropriate elements of the successive rows. The replacement probabilities for the  $i$ th row are obtained by the convolution of  $g_Y$  and the power of  $g_Y$  for the previous row. When the matrix  $P$  is completed, we remove zero rows and columns, corresponding to missing values of  $N$  and  $D$  (i.e., values with zero probability). To orient the joint probabilities as on the plane, we rotate  $P$  ninety degrees counterclockwise. With the joint distribution, we may then calculate any desired quantities.

### Example 15.1.5 A compound demand

The number of customers in a major appliance store is equally likely to be 1, 2, or 3. Each customer buys 0, 1, or 2 items with respective probabilities 0.5, 0.4, 0.1. Customers buy independently, regardless of the number of customers. First we determine the matrices representing  $g_N$  and  $g_Y$ . The coefficients are the probabilities that each integer value is observed. Note that the zero coefficients for any missing powers must be included.

```
gN = (1/3)*[0 1 1 1]; % Note zero coefficient for missing zero power
gY = 0.1*[5 4 1]; % All powers 0 thru 2 have positive coefficients
gend
Do not forget zero coefficients for missing powers
Enter the gen fn COEFFICIENTS for gN gN % Coefficient matrix named gN
Enter the gen fn COEFFICIENTS for gY gY % Coefficient matrix named gY
Results are in N, PN, Y, PY, D, PD, P
May use jcalc or jcalcf on N, D, P
To view distribution for D, call for gD
disp(gD) % Optional display of complete distribution
    0    0.2917
    1.0000    0.3667
    2.0000    0.2250
    3.0000    0.0880
    4.0000    0.0243
    5.0000    0.0040
    6.0000    0.0003
EN = N*PN'
EN = 2
EY = Y*PY'
EY = 0.6000
ED = D*PD'
ED = 1.2000 % Agrees with theoretical EN*EY
P3 = (D>=3)*PD'
P3 = 0.1167
[N,D,t,u,PN,PD,PL] = jcalcf(N,D,P);
EDn = sum(u.*P)./sum(P);
disp([N;EDn]')
    1.0000    0.6000 % Agrees with theoretical E[D|N=n] = n*EY
    2.0000    1.2000
    3.0000    1.8000
VD = (D.^2)*PD' - ED^2
VD = 1.1200 % Agrees with theoretical EN*VY + VN*EY^2
```

### Example 15.1.6 A numerical example

$$g_N(s) = \frac{1}{5}(1 + s + s^2 + s^3 + s^4) \quad g_Y(s) = 0.1(5s + 3s^2 + 2s^3)$$

Note that the zero power is missing from  $g_Y$ . corresponding to the fact that  $P(Y = 0) = 0$ .

```
gN = 0.2*[1 1 1 1 1];
gY = 0.1*[0 5 3 2];      % Note the zero coefficient in the zero position
gend
Do not forget zero coefficients for missing powers
Enter the gen fn COEFFICIENTS for gN  gN
Enter the gen fn COEFFICIENTS for gY  gY
Results are in N, PN, Y, PY, D, PD, P
May use jcalc or jcalcf on N, D, P
To view distribution for D, call for gD
disp(gD)                  % Optional display of complete distribution
```

0	0.2000
1.0000	0.1000
2.0000	0.1100
3.0000	0.1250
4.0000	0.1155
5.0000	0.1110
6.0000	0.0964
7.0000	0.0696
8.0000	0.0424
9.0000	0.0203
10.0000	0.0075
11.0000	0.0019
12.0000	0.0003

```
p3 = (D == 3)*PD'          % P(D=3)
P3 = 0.1250
P4_12 = ((D >= 4)&(D <= 12))*PD'
P4_12 = 0.4650             % P(4 <= D <= 12)
```

#### Example 15.1.7 Number of successes for random number $N$ of trials.

We are interested in the number of successes in  $N$  trials for a general counting random variable. This is a generalization of the Bernoulli case in [Example 15.1.2](#). Suppose, as in [Example 15.1.2](#), the number of customers in a major appliance store is equally likely to be 1, 2, or 3, and each buys at least one item with probability  $p = 0.6$ . Determine the distribution for the number  $D$  of buying customers.

##### Solution

We use  $gN$ ,  $gY$ , and  $gend$ .

```
gN = (1/3)*[0 1 1 1]; % Note zero coefficient for missing zero power
gY = [0.4 0.6];       % Generating function for the indicator function
gend
Do not forget zero coefficients for missing powers
Enter gen fn COEFFICIENTS for gN  gN
Enter gen fn COEFFICIENTS for gY  gY
Results are in N, PN, Y, PY, D, PD, P
May use jcalc or jcalcf on N, D, P
```

To view distribution for  $D$ , call for  $gD$   
`disp(gD)`

0	0.2080
1.0000	0.4560
2.0000	0.2640
3.0000	0.0720

The procedure `gend` is limited to simple  $N$  and  $Y_k$ , with nonnegative integer values. Sometimes, a random variable with unbounded range may be approximated by a simple random variable. The solution in the following example utilizes such an approximation procedure for the counting random variable  $N$ .

#### Example 15.1.8 Solution of the shop time Example 15.1.1

The number  $N$  of jobs brought to a service shop in a day is Poisson (8). The individual shop hour charges  $Y_k$  have the common distribution  $Y = [0 \ 1 \ 2]$  with probabilities  $PY = [1/4 \ 1/2 \ 1/4]$ .

Under the basic assumptions of our model, determine  $P(D \leq 4)$ .

#### Solution

Since Poisson  $N$  is unbounded, we need to check for a sufficient number of terms in a simple approximation. Then we proceed as in the simple case.

```
pa = cpoisson(8,10:5:30)      % Check for sufficient number of terms
pa = 0.2834    0.0173    0.0003    0.0000    0.0000
p25 = cpoisson(8,25)          % Check on choice of n = 25
p25 = 1.1722e-06
gN = ipoisson(8,0:25);        % Approximate gN
gY = 0.25*[1 2 1];
gend
Do not forget zero coefficients for missing powers
Enter gen fn COEFFICIENTS for gN  gN
Enter gen fn COEFFICIENTS for gY  gY
Results are in N, PN, Y, PY, D, PD, P
May use jcalc or jcalcf on N, D, P
To view distribution for D, call for gD
disp(gD(D<=20,:))            % Calculated values to D = 50
    0    0.0025                % Display for D <= 20
    1.0000    0.0099
    2.0000    0.0248
    3.0000    0.0463
    4.0000    0.0711
    5.0000    0.0939
    6.0000    0.1099
    7.0000    0.1165
    8.0000    0.1132
    9.0000    0.1021
   10.0000    0.0861
   11.0000    0.0684
   12.0000    0.0515
```

```

13.0000    0.0369
14.0000    0.0253
15.0000    0.0166
16.0000    0.0105
17.0000    0.0064
18.0000    0.0037
19.0000    0.0021
20.0000    0.0012

sum(PD)                                % Check on sufficiency of approximation
ans = 1.0000
P4 = (D<=4)*PD'
P4 = 0.1545                            % Theoretical value (4 places) = 0.1545
ED = D*PD'
ED = 8.0000                            % Theoretical = 8 (Example 15.1.4)
VD = (D.^2)*PD' - ED^2
VD = 11.9999                           % Theoretical = 12 (Example 15.1.4)

```

### The m-procedures mgd and jmgd

The next example shows a fundamental limitation of the gend procedure. The values for the individual demands are not limited to integers, and there are considerable gaps between the values. In this case, we need to implement the moment generating function  $M_D$  rather than the generating function  $g_D$ .

In the generating function case, it is as easy to develop the joint distribution for  $\{N, D\}$  as to develop the marginal distribution for  $D$ . For the moment generating function, the joint distribution requires considerably more computation. As a consequence, we find it convenient to have two m-procedures: *mgd* for the marginal distribution and *jmgd* for the joint distribution.

Instead of the convolution procedure used in gend to determine the distribution for the sums of the individual demands, the m-procedure mgd utilizes the m-function mgsum to obtain these distributions. The distributions for the various sums are concatenated into two row vectors, to which csort is applied to obtain the distribution for the compound demand. The procedure requires as input the generating function for  $N$  and the actual distribution,  $Y$  and  $PY$ , for the individual demands. For  $gN$ , it is necessary to treat the coefficients as in gend. However, the actual values and probabilities in the distribution for  $Y$  are put into a pair of row matrices. If  $Y$  is integer valued, there are no zeros in the probability matrix for missing values.

#### Example 15.1.9 Noninteger values

A service shop has three standard charges for a certain class of warranty services it performs: \$10, \$12.50, and \$15. The number of jobs received in a normal work day can be considered a random variable  $N$  which takes on values 0, 1, 2, 3, 4 with equal probabilities 0.2. The job types for arrivals may be represented by an iid class  $\{Y_i : 1 \leq i \leq 4\}$ , independent of the arrival process. The  $Y_i$  take on values 10, 12.5, 15 with respective probabilities 0.5, 0.3, 0.2. Let  $C$  be the total amount of services rendered in a day. Determine the distribution for  $C$ .

#### Solution

```

gN = 0.2*[1 1 1 1 1];                % Enter data
Y = [10 12.5 15];
PY = 0.1*[5 3 2];
mgd                                    % Call for procedure
Enter gen fn COEFFICIENTS for gN gN
Enter VALUES for Y Y
Enter PROBABILITIES for Y PY
Values are in row matrix D; probabilities are in PD.

```

To view the distribution, call for mD.

```
disp(mD)                                % Optional display of distribution
      0      0.2000
    10.0000    0.1000
    12.5000    0.0600
    15.0000    0.0400
    20.0000    0.0500
    22.5000    0.0600
    25.0000    0.0580
    27.5000    0.0240
    30.0000    0.0330
    32.5000    0.0450
    35.0000    0.0570
    37.5000    0.0414
    40.0000    0.0353
    42.5000    0.0372
    45.0000    0.0486
    47.5000    0.0468
    50.0000    0.0352
    52.5000    0.0187
    55.0000    0.0075
    57.5000    0.0019
    60.0000    0.0003
```

We next recalculate [Example 15.1.6](#), above, using mgd rather than gend.

#### Example 15.1.10 Recalculation of Example 15.1.6

In Example 15.1.6, we have

$$g_N(s) = \frac{1}{5}(1 + s + s^2 + s^3 + s^4) \quad g_Y(s) = 0.1(5s + 3s^2 + 2s^3)$$

The means that the distribution for  $Y$  is  $Y = [1 \ 2 \ 3]$  and  $PY = 0.1 * [5 \ 3 \ 2]$ .

We use the same expression for  $gN$  as in Example 15.1.6.

```
gN = 0.2*ones(1,5);
Y = 1:3;
PY = 0.1*[5 3 2];
mgd
Enter gen fn COEFFICIENTS for gN  gN
Enter VALUES for Y  Y
Enter PROBABILITIES for Y  PY
Values are in row matrix D; probabilities are in PD.
To view the distribution, call for mD.
disp(mD)
      0      0.2000
     1.0000    0.1000
     2.0000    0.1100
```

```

3.0000    0.1250
4.0000    0.1155
5.0000    0.1110
6.0000    0.0964
7.0000    0.0696
8.0000    0.0424
9.0000    0.0203
10.0000   0.0075
11.0000   0.0019
12.0000   0.0003
P3 = (D==3)*PD'
P3 =    0.1250
ED = D*PD'
ED =    3.4000
P_4_12 = ((D>=4)&(D<=12))*PD'
P_4_12 =    0.4650
P7 = (D>=7)*PD'
P7 =    0.1421

```

As expected, the results are the same as those obtained with `gend`.

If it is desired to obtain the joint distribution for  $\{N, D\}$ , we use a modification of `mgd` called `jmgd`. The complications come in placing the probabilities in the  $P$  matrix in the desired positions. This requires some calculations to determine the appropriate size of the matrices used as well as a procedure to put each probability in the position corresponding to its  $D$  value. Actual operation is quite similar to the operation of `mgd`, and requires the same data format.

A principle use of the joint distribution is to demonstrate features of the model, such as  $E[D|N=n] = nE[Y]$ , etc. This, of course, is utilized in obtaining the expressions for  $M_D(s)$  in terms of  $g_N(s)$  and  $M_Y(s)$ . This result guides the development of the computational procedures, but these do not depend upon this result. However, it is usually helpful to demonstrate the validity of the assumptions in typical examples.

*Remark.* In general, if the use of `gend` is appropriate, it is faster and more efficient than `mgd` (or `jmgd`). And it will handle somewhat larger problems. But both m-procedures work quite well for problems of moderate size, and are convenient tools for solving various “compound demand” type problems.

This page titled [15.1: Random Selection](#) is shared under a [CC BY 3.0](#) license and was authored, remixed, and/or curated by [Paul Pfeiffer](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.