Algorithms of sampling
with equal or unequal probabilities

Yves Tillé
University of Neuchâtel

Euskal Estatistika Erakunde
XXIII Seminario Internacional de Estadística
November 2010
Table of Contents

1. General Ideas and Definitions
2. Expectation, Inclusion, Estimation
3. Sampling Algorithms
4. Simple Random Sampling
5. Basic Unequal Probability Methods
6. Unequal Probability Exponential Designs
7. The splitting method
Population

General Concepts
General Ideas

Three mains definitions.

1. Supports or set of samples (example all the samples with replacement with fixed sample size $n$)
2. Sampling design or multivariate discrete positive distribution.
3. Sampling algorithms (applicable to any support and any design), ex: sequential algorithms.

The application of a particular sampling algorithm on a sampling design defined on a particular support gives a sampling procedure.
Population

- Finite population, set of $N$ units $\{u_1, \ldots, u_k, \ldots, u_N\}$.
- Each unit can be identified without ambiguity by a label.
- Let

$$U = \{1, \ldots, k, \ldots, N\}$$

be the set of these labels.
Variable of Interest

- The total $Y = \sum_{k \in U} y_k$,
- The population size $N = \sum_{k \in U} 1$,
- The mean $\bar{Y} = \frac{1}{N} \sum_{k \in U} y_k$,
- The variance $\sigma_y^2 = \frac{1}{N} \sum_{k \in U} (y_k - \bar{Y})^2$.
- The corrected variance $V_y^2 = \frac{1}{N - 1} \sum_{k \in U} (y_k - \bar{Y})^2$. 
Sample Without Replacement

- A sample without replacement is denoted by a column vector
  \[ \mathbf{s} = (s_1 \cdots s_k \cdots s_N)' \in \{0, 1\}^N, \]
  where
  \[ s_k = \begin{cases} 
  1 & \text{if unit } k \text{ is in the sample} \\
  0 & \text{if unit } k \text{ is not in the sample}, 
  \end{cases} \]
  for all \( k \in U \).
- The sample size is \( n(\mathbf{s}) = \sum_{k \in U} s_k \).
Sample With Replacement

- Samples with replacement,

\[ s = (s_1 \, \cdots \, s_k \, \cdots \, s_N) \in \mathbb{N}^N, \]

where \( \mathbb{N} = \{0, 1, 2, 3, \ldots\} \)

and \( s_k \) is the number of times that unit \( k \) is in the sample.

- The sample size is

\[ n(s) = \sum_{k \in U} s_k, \]

and, in sampling with replacement, we can have \( n(s) > N \).
Definition
A support $Q$ is a set of samples.

Definition
A support $Q$ is said to be symmetric if, for any $s \in Q$, all the permutations of the coordinates of $s$ are also in $Q$. 
Particular symmetric supports 1

- The symmetric support without replacement: $S = \{0, 1\}^N$.
- Note that $\text{card}(S) = 2^N$. 

![Diagram of a cube with labeled vertices](image)
The symmetric support without replacement with fixed sample size
\[ S_n = \left\{ s \in S \middle| \sum_{k \in U} s_k = n \right\}. \]
The symmetric support with replacement $\mathcal{R} = \mathbb{N}^N$, 

![Cube diagram with labeled vertices]
Particular symmetric supports 4

The symmetric support with replacement of fixed size $n$

$$\mathcal{R}_n = \{ s \in \mathcal{R} \mid \sum_{k \in U} s_k = n \}.$$
General Ideas and Definitions

Properties

1. $S, S_n, R, R_n$, are symmetric,
2. $S \subset R$,
3. The set $\{S_0, \ldots, S_n, \ldots, S_N\}$ is a partition of $S$,
4. The set $\{R_0, \ldots, R_n, \ldots, R_N, \ldots\}$ is an infinite partition of $R$,
5. $S_n \subset R_n$, for all $n = 0, \ldots, N$. 
Sampling Design and Random Sample

Definition
A sampling design \( p(\cdot) \) on a support \( Q \) is a multivariate probability distribution on \( Q \); that is, \( p(\cdot) \) is a function from support \( Q \) to \([0, 1]\) such that \( p(s) > 0 \) for all \( s \in Q \) and

\[
\sum_{s \in Q} p(s) = 1.
\]

Remark
Because \( S \) can be viewed as the set of all the vertices of a hypercube, a sampling design without replacement is a probability measure on all these vertices.
Random Sample

Definition

A random sample $\mathbf{S} \in \mathbb{R}^N$ with the sampling design $p(.)$ is a random vector such that

$$\Pr(\mathbf{S} = \mathbf{s}) = p(\mathbf{s}), \text{ for all } \mathbf{s} \in Q,$$

where $Q$ is the support of $p(.)$. 
Expectation and variance

**Definition**

The expectation of a random sample $S$ is

$$\mu = E(S) = \sum_{s \in Q} p(s)s.$$ 

The joint expectation

$$\mu_{k\ell} = \sum_{s \in Q} p(s)s_k s_\ell.$$ 

The variance-covariance operator

$$\Sigma = [\Sigma_{k\ell}] = \text{var}(S) = \sum_{s \in Q} p(s)(s - \mu)(s - \mu)' = [\mu_{k\ell} - \mu_k \mu_\ell].$$
Inclusion probabilities

Definition
The first-order inclusion probability is the probability that unit \( k \) is in the random sample

\[
\pi_k = \Pr(S_k > 0) = \mathbb{E}[r(S_k)],
\]

where \( r(.) \) is the reduction function

\[
r(S_k) = \begin{cases} 
1 & \text{if } S_k > 0 \\
0 & \text{if } S_k = 0 
\end{cases}
\]

\( \pi = (\pi_1 \cdots \pi_k \cdots \pi_N)' \).
Inclusion probabilities

Definition

The joint inclusion probability is the probability that unit \( k \) and \( \ell \) are together in the random sample

\[
\pi_{k\ell} = \Pr(S_k > 0 \text{ and } S_\ell > 0) = \mathbb{E}[r(S_k)r(S_\ell)],
\]

with \( \pi_{kk} = \pi_k, k \in U \).

Let \( \Pi = [\pi_{k\ell}] \) be the matrix of joint inclusion probabilities. Moreover, we define

\[
\Delta = \Pi - \pi \pi'.
\]
Inclusion probabilities

Result

\[ \sum_{k \in U} \pi_k = E \{ n[r(S)] \} , \]

and

\[ \sum_{k \in U} \Delta_{k\ell} = E \{ n[r(S)] (r(S_{\ell}) - \pi_{\ell}) \} , \text{ for all } \ell \in U. \]

Moreover, if \( \text{var} \{ n[r(S)] \} = 0 \) then

\[ \sum_{k \in U} \Delta_{k\ell} = 0 , \text{ for all } \ell \in U. \]
Computation of the Inclusion Probabilities

- Auxiliary variables $x_k > 0, k \in U$.
- First, compute the quantities

$$\frac{n x_k}{\sum_{\ell \in U} x_{\ell}},$$

$k = 1, \ldots, N$.

- For units for which these quantities are larger than 1, set $\pi_k = 1$.

Next, the quantities are recalculated using (1) restricted to the remaining units.
Characteristic Function

The characteristic function $\phi(t)$ from $\mathbb{R}^N$ to $\mathbb{C}$ of a random sample $S$ with sampling design $p(.)$ on $Q$ is defined by

$$\phi_S(t) = \sum_{s \in Q} e^{it's} p(s), \; t \in \mathbb{R}^N,$$

where $i = \sqrt{-1}$, and $\mathbb{C}$ is the set of the complex numbers.

$$\phi'(0) = i \mu, \quad \text{and} \quad \phi''(0) = -(\Sigma + \mu \mu').$$
The Hansen-Hurwitz estimator (see Hansen and Hurwitz, 1943) of $Y$ is defined by

$$\hat{Y}_{HH} = \sum_{k \in U} \frac{S_k y_k}{\mu_k},$$

where $\mu_k = \mathbb{E}(S_k)$, $k \in U$.

**Result**

*If $\mu_k > 0$, for all $k \in U$, then $\hat{Y}_{HH}$ is an unbiased estimator of $Y$.***
The Horvitz-Thompson estimator (see Horvitz and Thompson, 1952) is defined by

$$\hat{Y}_{HT} = \sum_{k \in U} \frac{r(S_k) y_k}{\pi_k},$$

where

$$r(S_k) = \begin{cases} 
0 & \text{if } S_k = 0 \\
1 & \text{if } S_k > 0.
\end{cases}$$
Population

Sampling Algorithms
Definition

A sampling algorithm is a procedure allowing the selection of a random sample.

An algorithm must be a shortcut that avoid the combinatorial explosion.
**Algorithm** Enumerative algorithm

1. First, construct a list \( \{s_1, s_2, \ldots, s_j, \ldots, s_J\} \) of all possible samples with their probabilities.

2. Next, generate a random variable \( u \) with a uniform distribution in \([0,1]\).

3. Finally, select the sample \( s_j \) such that

\[
\sum_{i=1}^{j-1} p(s_i) \leq u < \sum_{i=1}^{j} p(s_i).
\]
### Table: Sizes of symmetric supports

<table>
<thead>
<tr>
<th>Support $Q$</th>
<th>$\text{card}(Q)$</th>
<th>$N = 100, n = 10$</th>
<th>$N = 300, n = 30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{R}$</td>
<td>$\infty$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\mathcal{R}_n$</td>
<td>$\binom{N+n-1}{n}$</td>
<td>$5.1541 \times 10^{13}$</td>
<td>$3.8254 \times 10^{42}$</td>
</tr>
<tr>
<td>$S$</td>
<td>$2^N$</td>
<td>$1.2677 \times 10^{30}$</td>
<td>$2.0370 \times 10^{90}$</td>
</tr>
<tr>
<td>$S_n$</td>
<td>$\binom{N}{n}$</td>
<td>$1.7310 \times 10^{13}$</td>
<td>$1.7319 \times 10^{41}$</td>
</tr>
</tbody>
</table>
Sequential Algorithms

A sequential procedure is a method that is applied to a list of units sorted according to a particular order denoted $1, \ldots, k, \ldots, N$.

**Definition**

A sampling procedure is said to be weakly sequential if at step $k = 1, \ldots, N$ of the procedure, the decision concerning the number of times that unit $k$ is in the sample is definitively taken.

**Definition**

A sampling procedure is said to be strictly sequential if it is weakly sequential and if the decision concerning unit $k$ does not depend on the units that are after $k$ on the list.
Algorithm Standard sequential procedure

1. Let \( p(s) \) be the sampling design and \( Q \) the support. First, define

\[
q_1(s_1) = \Pr(S_1 = s_1) = \sum_{s \in Q | S_1 = s_1} p(s), \ s_1 = 0, 1, 2, \ldots
\]

2. Select the first unit \( s_1 \) times according to the distribution \( q_1(s_1) \).

3. For \( k = 2, \ldots, N \) do

   1. Compute

   \[
   q_k(s_k) = \Pr(S_k = s_k | S_{k-1} = s_{k-1}, \ldots, S_1 = s_1)
   = \frac{\sum_{s \in Q | S_k = s_k, S_{k-1} = s_{k-1}, \ldots, S_1 = s_1} p(s)}{\sum_{s \in Q | S_{k-1} = s_{k-1}, \ldots, S_1 = s_1} p(s)}, \ s_k = 0, 1, 2, \ldots
   \]

   2. Select the \( k \)th unit \( s_k \) times according to the distribution \( q_k(s_k) \);

EndFor.
Draw by draw Algorithms

The draw by draw algorithms are restricted to designs with fixed sample size. We refer to the following definition.

**Definition**

A sampling design of fixed sample size $n$ is said to be draw by draw if, at each one of the $n$ steps of the procedure, a unit is definitively selected in the sample.
Standard Draw by Draw Algorithm

**Algorithm** Standard draw by draw algorithm

1. Let \( p(s) \) be a sampling design and \( Q \subseteq \mathcal{R}_n \) the support. First, define \( p^{(0)}(s) = p(s) \) and \( Q(0) = Q \). Define also \( \mathbf{b}(0) \) as the null vector of \( \mathbb{R}^N \).

2. **For** \( t = 0, \ldots, n - 1 \) **do**
   
   1. Compute \( \nu(t) = \sum_{s \in Q(t)} sp(t)(s) \);
   2. Select randomly one unit from \( U \) with probabilities \( q_k(t) \), where
      
      \[
      q_k(t) = \frac{\nu_k(t)}{\sum_{\ell \in U} \nu_\ell(t)} = \frac{\nu_k(t)}{n - t}, \quad k \in U;
      \]
      
      The selected unit is denoted \( j \);
   3. Define \( \mathbf{a}_j = (0 \cdots 0 1 \cdots 0) \); Execute \( \mathbf{b}(t + 1) = \mathbf{b}(t) + \mathbf{a}_j \);

3. Define \( Q(t + 1) = \{ \tilde{s} = s - \mathbf{a}_j, \text{ for all } s \in Q(t) \text{ such that } s_j > 0 \} \);

4. Define, for all \( \tilde{s} \in Q(t + 1) \),
   
   \[
   p^{(t+1)}(\tilde{s}) = \frac{s_j p(t)(s)}{\sum_{s \in Q(t)} s_j p(t)(s)}, \quad \text{where } s = \tilde{s} + \mathbf{a}_j;
   \]

5. The selected sample is \( \mathbf{b}(n) \).
Algorithm Standard draw by draw algorithm for sampling without replacement

1. Let \( p(s) \) be a sampling design and \( \mathcal{Q} \in \mathcal{S} \) the support.

2. Define \( \mathbf{b} = (b_k) = 0 \in \mathbb{R}^N \).

3. For \( t = 0, \ldots, n - 1 \) do
   
   - select a unit from \( U \) with probability
     \[
     q_k = \begin{cases} 
     \frac{1}{n-t} & \mathbb{E}(S_k | S_i = 1 \text{ for all } i \text{ such that } b_i = 1) \\
     0 & \text{if } b_k = 1 \\
     \end{cases}
     \]
   
   If unit \( j \) is selected, then \( b_j = 1 \);
Other Algorithms

- Eliminatory algorithms (Chao, Tillé)
- Splitting methods
- Rejective algorithms
- Systematic algorithms
- Others algorithms (Sampford)
Population

Simple Random Sampling
Simple Random Sampling

Definition

A sampling design $p_{\text{SIMPLE}}(\cdot, \theta, Q)$ of parameter $\theta \in \mathbb{R}^*$ on a support $Q$ is said to be simple, if

(i) Its sampling design can be written

$$p_{\text{SIMPLE}}(s, \theta, Q) = \frac{\theta^n(s) \prod_{k \in U} 1/s_k!}{\sum_{s \in Q} \theta^n(s) \prod_{k \in U} 1/s_k!}, \quad \text{for all } s \in Q.$$ 

(ii) Its support $Q$ is symmetric (see Definition 2, page 9).
Simple Random Sampling

- Support $S$ Bernoulli sampling
- Support $S_n$ Simple Random Sampling Without Replacement
- Support $R$ Bernoulli sampling With replacement
- Support $R_n$ Simple Random Sampling With Replacement
Links between simple designs

Figure: Links between the main simple sampling designs
## Main simple random sampling designs

<table>
<thead>
<tr>
<th>Notation</th>
<th>BERNWR</th>
<th>SRSWR</th>
<th>BERN</th>
<th>SRSWOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(s)$</td>
<td>$\frac{\mu^n(s)}{e^{N\mu} \prod_{k \in U} \frac{1}{s_k!}}$</td>
<td>$\frac{n!}{N^n} \prod_{k \in U} \frac{1}{s_k!}$</td>
<td>$\pi^n(s)(1 - \pi)^{N-n(s)}$</td>
<td>$\binom{N}{n}^{-1}$</td>
</tr>
<tr>
<td>$Q$</td>
<td>$\frac{\mu^n(s)}{e^{N\mu} \prod_{k \in U} \frac{1}{s_k!}}$</td>
<td>$\frac{n!}{N^n} \prod_{k \in U} \frac{1}{s_k!}$</td>
<td>$\prod_{k \in U} \left(1 + \pi \left(e^{it_k} - 1\right)\right)$</td>
<td>$\binom{N}{n}^{-1} \sum_{s \in S_n} e^{it's}$</td>
</tr>
<tr>
<td>$\phi(t)$</td>
<td>$\exp\left{\mu \sum_{k \in U} (e^{it_k} - 1)\right}$</td>
<td>$\left(\frac{1}{N} \sum_{k \in U} e^{it_k}\right)^n$</td>
<td>$\prod_{k \in U} {1 + \pi \left(e^{it_k} - 1\right)}$</td>
<td>$\binom{N}{n}^{-1} \sum_{s \in S_n} e^{it's}$</td>
</tr>
<tr>
<td>WOR/WR</td>
<td>with repl.</td>
<td>with repl.</td>
<td>without repl.</td>
<td>without repl.</td>
</tr>
<tr>
<td>$n(S)$</td>
<td>random</td>
<td>fixed</td>
<td>random</td>
<td>fixed</td>
</tr>
<tr>
<td>$\mu_k$</td>
<td>$\mu$</td>
<td>$\frac{n}{N}$</td>
<td>$\pi$</td>
<td>$\frac{n}{N}$</td>
</tr>
<tr>
<td>$\pi_k$</td>
<td>$1 - e^{-\mu}$</td>
<td>$1 - \left(\frac{N - 1}{N}\right)^n$</td>
<td>$\pi$</td>
<td>$\frac{n}{N}$</td>
</tr>
</tbody>
</table>
Sequential procedure on Bernoulli sampling

**Algorithm** Bernoulli sampling without replacement

**Definition**

\[
\text{k : Integer;}
\]

\[
\text{For } k = 1, \ldots, N \text{ do with probability } \pi \text{ select unit } k; \text{ EndFor.}
\]
Algorithm Draw by draw procedure for SRSWOR

**Definition** \( j : \text{INTEGER}; \)

**For** \( t = 0, \ldots, n - 1 \) **do**

select a unit \( k \) from the population with probability

\[
q_k = \begin{cases} 
\frac{1}{N-t} & \text{if } k \text{ is not already selected} \\
0 & \text{if } k \text{ is already selected}
\end{cases}
\]

**EndFor**.
Fan et al. (1962)

**Algorithm** Selection-rejection procedure for SRSWOR

**Definition**\[ k, j : \text{Integer}; \]
\[ j = 0; \]
\[ \text{FOR } k = 1, \ldots, N \text{ DO} \]
\[ \quad \text{with probability } \frac{n - j}{N - (k - 1)} \text{ THEN select unit } k; \]
\[ \quad j = j + 1; \]
\[ \text{ENDFOR.} \]
Algorithm Draw by Draw Procedure for SRSWR

**Definition** \( j : \text{Integer}; \)  
**For** \( j = 1, \ldots, n \) **do**  
   a unit is selected with equal probability \( 1/N \) from the population \( U \);  
**EndFor.**
Sequential Procedure for SRSWR

Algorithm Sequential procedure for SRSWR

**Definition**  $k, j : \text{Integer};$

$j = 0;$

**For** $k = 1, \ldots, N \text{ do}$

- select the $k$th unit $s_k$ times according to the binomial distribution

$$B\left(n - \sum_{i=1}^{k-1} s_i, \frac{1}{N - k + 1}\right);$$

**EndFor.**
Unequal Probability Sampling
Selection of 2 units with unequal probability

\[ p_k = \frac{x_k}{\sum_{\ell \in U} x_\ell}, \ k \in U. \]

The generalization is the following:
- At the first step, select a unit with unequal probability \( p_k, \ k \in U \).
- The selected unit is denoted \( j \).
- The selected unit is removed from \( U \).
- Next we compute

\[ p_k^j = \frac{p_k}{1 - p_j}, \ k \in U \setminus \{j\}. \]
Select again a unit with unequal probabilities \( p_k^j \), \( k \in U \), amongst the \( N - 1 \) remaining units, and so on. This method is wrong. We can see it by taking \( n = 2 \).
Why the problem is complex? 3

In this case,

\[ Pr(k \in S) = Pr(k \text{ be selected at the first step}) + Pr(k \text{ be selected at the second step}) = p_k + \sum_{j \in U \backslash \{k\}} p_j p_k \]

\[ = p_k \left(1 + \sum_{j \in U \backslash \{k\}} \frac{p_j}{1 - p_j}\right). \]  

(3)

We should have \( \pi_k = 2p_k, k \in U. \)
Why the problem is complex? 4

We could use modified values $p_k^*$ for the $p_k$ in such a way that the inclusion probabilities is equal to $\pi_k$.
In the case where $n = 2$, we should have $p_k^*$ such that

$$p_k^* \left( 1 + \sum_{j \in U, j \neq k} \frac{p_j^*}{1 - p_j^*} \right) = \pi_k, \ k \in U.$$

This method is known as the Nairin procedure (see also Horvitz and Thompson, 1952; Yates and Grundy, 1953; Brewer and Hanif, 1983, p.25)
Madow (1949)
Fixed sample size and exact method.
We have \(0 < \pi_k < 1, k \in U\) with
\[
\sum_{k \in U} \pi_k = n.
\]

Define \(V_k = \sum_{\ell=1}^{k} \pi_\ell\), for all \(k \in U\), with \(V_o = 0\). A uniform random number is generated in \([0, 1]\).
- the first unit selected \(k_1\) is such that \(V_{k_1-1} \leq u < V_{k_1}\),
- the second unit selected is such that \(V_{k_2-1} \leq u + 1 < V_{k_2}\) and
- the \(j\)th unit selected is such that \(V_{k_j-1} \leq u + j - 1 < V_{k_j}\). 


Example

Suppose that $N = 6$ and $n = 3$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_k$</td>
<td>0.07</td>
<td>0.17</td>
<td>0.41</td>
<td>0.61</td>
<td>0.83</td>
<td>0.91</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$V_k$</td>
<td>0.07</td>
<td>0.24</td>
<td>0.65</td>
<td>1.26</td>
<td>2.09</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Suppose also that the value taken by the uniform random number is $u = 0.354$. The rules of selections are:

- Because $V_2 \leq u < V_3$, unit 3 is selected;
- Because $V_4 \leq u < V_5$, unit 5 is selected;
- Because $V_5 \leq u < V_6$, unit 6 is selected.

The sample selected is thus $s = (0, 0, 1, 0, 1, 1)$. 
Systematic sampling 4

\[ \pi = \begin{pmatrix}
0.07 & 0 & 0 & 0.07 & 0.07 & 0 \\
0 & 0.17 & 0 & 0.17 & 0.02 & 0.15 \\
0 & 0 & 0.41 & 0.02 & 0.39 & 0.41 \\
0.07 & 0.17 & 0.02 & 0.61 & 0.44 & 0.52 \\
0.07 & 0.02 & 0.39 & 0.44 & 0.83 & 0.74 \\
0 & 0.15 & 0.41 & 0.52 & 0.74 & 0.91 \\
\end{pmatrix}. \]
**Algorithm** Systematic sampling

**Definition** \(a, b, u\) real; \(k\) Integer;

\[u \in \mathcal{U}[0, 1];\]

\[a = -u;\]

For \(k = 1, \ldots, N\) do

\[b = a;\]

\[a = a + \pi_k;\]

If \(\lfloor a \rfloor \neq \lfloor b \rfloor\) then select \(k\)

EndFor.
Systematic sampling 6

Problem: most of the joint inclusion probabilities are equal to zero. Matrix of the joint inclusion probabilities:

\[
\begin{bmatrix}
- & 0 & 0.2 & 0.2 & 0 & 0 \\
0 & - & 0.5 & 0.2 & 0.4 & 0.3 \\
0.2 & 0.5 & - & 0.3 & 0.4 & 0.2 \\
0.2 & 0.2 & 0.3 & - & 0 & 0.3 \\
0 & 0.4 & 0.4 & 0 & - & 0 \\
0 & 0.3 & 0.2 & 0.3 & 0 & -
\end{bmatrix}
\]
Systematic sampling

- The sampling design depends on the order of the population.
- When the variable of interest depends on the order of the file, the variance is reduced.
- **Random systematic sampling:** The file is sorted randomly before applying random systematic sampling.
Exponential family

Definition

A sampling design \( p_{\text{EXP}}(.) \) on a support \( Q \) is said to be exponential if it can be written

\[
p_{\text{EXP}}(s, \lambda, Q) = g(s) \exp \left[ \lambda' s - \alpha(\lambda, Q) \right],
\]

where \( \lambda \in \mathbb{R}^N \) is the parameter,

\[
g(s) = \prod_{k \in U} \frac{1}{s_k!},
\]

and \( \alpha(\lambda, Q) \) is called the normalizing constant and is given by

\[
\alpha(\lambda, Q) = \log \sum_{s \in Q} g(s) \exp \lambda' s.
\]
The expectation

\[ \mu(\lambda) = \sum_{s \in Q} s p_{\text{EXP}}(s, \lambda, Q) \]

- The function \( \mu(\lambda) \) is bijective. (fundamental result on exponential families).
- The most important in an exponential family is its parameter and not \( \mu \) or \( \pi \).
### Main exponential designs

<table>
<thead>
<tr>
<th>Notation</th>
<th>POISSWR</th>
<th>MULTI</th>
<th>POISSWOR</th>
<th>CPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(s)$</td>
<td>$\prod_{k \in U} \frac{\mu_k^{s_k} e^{-\mu_k}}{s_k!}$</td>
<td>$\frac{n!}{n^n} \prod_{k \in U} \frac{\mu_k^{s_k}}{s_k!}$</td>
<td>$\prod_{k \in U} \left[ \pi_k^{s_k} (1 - \pi_k)^{1-s_k} \right]$</td>
<td>$\sum_{s \in S_n} \exp[\lambda' s - \alpha(\lambda, S_n)]$</td>
</tr>
<tr>
<td>$Q$</td>
<td>$\mathcal{R}$</td>
<td>$\mathcal{R}_n$</td>
<td>$S$</td>
<td>$S_n$</td>
</tr>
<tr>
<td>$\alpha(\lambda, Q)$</td>
<td>$\sum_{k \in U} \exp \lambda_k$</td>
<td>$\log \frac{1}{n!} \left( \sum_{k \in U} \exp \lambda_k \right)^n$</td>
<td>$\log \prod_{k \in U} (1 + \exp \lambda_k)$</td>
<td>difficult</td>
</tr>
<tr>
<td>$\phi(t)$</td>
<td>$\exp \sum_{k \in U} \mu_k (e^{it_k} - 1)$</td>
<td>$\left( \frac{1}{n} \sum_{k \in U} \mu_k \exp it_k \right)^n$</td>
<td>$\prod_{k \in U} \left{ 1 + \pi_k (\exp it_k - 1) \right}$</td>
<td>not reducible</td>
</tr>
<tr>
<td>WOR/WR</td>
<td>with repl.</td>
<td>with repl.</td>
<td>without repl.</td>
<td>without repl.</td>
</tr>
<tr>
<td>$n(S)$</td>
<td>random</td>
<td>fixed</td>
<td>random</td>
<td>fixed</td>
</tr>
<tr>
<td>$\mu_k$</td>
<td>$\mu_k = \exp \lambda_k$</td>
<td>$\mu_k = \frac{n \exp \lambda_k}{\sum_{k \in U} \exp \lambda_k}$</td>
<td>$\pi_k = \frac{\exp \lambda_k}{1+\exp \lambda_k}$</td>
<td>$\pi_k(\lambda, S_n)$ difficult</td>
</tr>
<tr>
<td>$\pi_k$</td>
<td>$1 - e^{-\mu_k}$</td>
<td>$1 - (1 - \mu_k/n)^n$</td>
<td>$\pi_k$</td>
<td>$\pi_k(\lambda, S_n)$</td>
</tr>
</tbody>
</table>
Algorithm Sequential procedure for multinomial design

**Definition** \( k : \text{Integer}; \)

**For** \( k = 1, \ldots, N \) **do**

select the \( k \)th unit \( s_k \) times according to the binomial distribution

\[
\mathcal{B} \left( n - \sum_{\ell=1}^{k-1} s_\ell, \frac{\mu_k}{n - \sum_{\ell=1}^{k-1} \mu_\ell} \right);
\]

**EndFor.**
Draw by draw procedure for multinomial design

**Algorithm** Draw by draw procedure for multinomial design

**Definition** \( j : \text{INTEGER} ; \)

**For** \( j = 1, \ldots, n \) **do**

a unit is selected with probability \( \mu_k / n \) from the population \( U \);

**EndFor.**
Algorithm Sequential procedure for POISSWOR

**Definition** $k$: Integer;
For $k = 1, \ldots, N$, do select the $k$th unit with probability $\pi_k$;
EndFor.
Conditional Poisson Sampling (CPS)

- CPS = Exponential design on $S_n$ (or maximum entropy design)
- Chen et al. (1994) and Deville (2000)

$$p_{CPS}(s, \lambda, n) = p_{\text{EXP}}(s, \lambda, S_n) = \frac{\exp \lambda' s}{\sum_{s \in S_n} \exp \lambda' s}$$

The relation between $\lambda$ and $\pi$ is complex, but there exists the recursive relation:

$$\pi_k(\lambda, S_n) = n \frac{\exp \lambda_k [1 - \pi_k(\lambda, S_{n-1})]}{\sum_{\ell \in U} \exp \lambda_{\ell} [1 - \pi_{\ell}(\lambda, S_{n-1})]} \quad (\text{with } \pi_{\ell}(\lambda, S_0) = 0)$$

- For obtaining $\lambda$ from $\pi$, the Newton method can be used.
Rejective procedure

- For example, select poisson samples until obtaining a fixed sample size.  
  \[ p_{\text{CPS}}(s, \lambda, n) = p_{\text{EXP}}(s, \lambda, S_n) = \frac{p_{\text{EXP}}(s, \lambda, S)}{\sum_{s \in S_n} p_{\text{EXP}}(s, \lambda, S)} \]
  
  - \( p_{\text{EXP}}(s, \lambda, S) \) is a Poisson design
  - \( p_{\text{EXP}}(s, \lambda, S_n) \) is a conditional Poisson design

- Warning: the use of a rejective procedure changes the inclusion probabilities.

- The parameter of the exponential design remains the same.
Idea of implementations: Rejective procedure

- The $\pi_k$ of the CPS are given.
- Compute $\lambda$ from the $\pi_k$ by the Newton method.
- Compute the inclusion probabilities of the Poisson design
  \[ \tilde{\pi}_k = \exp(\lambda_k + C)/[1 + \exp(\lambda_k + C)]. \]
- Select Poisson samples until obtaining the good sample size $n$. 
Implementation of CPS

- Sequential procedure
- Draw by draw procedure
- Poisson rejective procedure
- Multinomial rejective procedure

For all these procedures, $\lambda$ must first be computed from $\pi$. Next, the implementation becomes relatively simple.
Link between the exponential methods

- **POISSWR**: conditioning on $S$
- **POISSWOR**: conditioning on $S_n$
- **MULTI**: conditioning on $S_n$ with $0 \leq n \leq N$
- **CPS**: conditioning on $S_n$

Reduction
The splitting method

Splitting Method
Basic splitting method

Deville and Tillé (1998)

\( \pi_k \) is split into two parts \( \pi_k^{(1)} \) and \( \pi_k^{(2)} \) that must satisfy:

\[
\pi_k = \lambda \pi_k^{(1)} + (1 - \lambda) \pi_k^{(2)}; \quad (4)
\]

\[
0 \leq \pi_k^{(1)} \leq 1 \quad \text{and} \quad 0 \leq \pi_k^{(2)} \leq 1, \quad (5)
\]

\[
\sum_{k \in U} \pi_k^{(1)} = \sum_{k \in U} \pi_k^{(2)} = n, \quad (6)
\]

where \( \lambda \) can be chosen freely provided that \( 0 < \lambda < 1 \). The method consists of drawing \( n \) units with unequal probabilities

\[
\left\{ \begin{array}{l}
\pi_k^{(1)}, k \in U, \quad \text{with a probability } \lambda \\
\pi_k^{(2)}, k \in U, \quad \text{with a probability } 1 - \lambda.
\end{array} \right.
\]
Basic splitting method

\[ \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_k \\ \vdots \\ \pi_N \end{bmatrix} \]

\[ \begin{bmatrix} \lambda \\ 1 - \lambda \end{bmatrix} \]

\[ \begin{bmatrix} \pi_1^{(1)} \\ \vdots \\ \pi_k^{(1)} \\ \vdots \\ \pi_N^{(1)} \end{bmatrix} \quad \begin{bmatrix} \pi_1^{(2)} \\ \vdots \\ \pi_k^{(2)} \\ \vdots \\ \pi_N^{(2)} \end{bmatrix} \]

**Figure:** Splitting into two parts
Splitting method into \( M \) parts

Construct the \( \pi^{(j)}_k \) and the \( \lambda_j \) in such a way that

\[
\sum_{j=1}^{M} \lambda_j = 1, \\
0 \leq \lambda_j \leq 1 \quad (j = 1, \ldots, M),
\]

\[
\sum_{j=1}^{M} \lambda_j \pi^{(j)}_k = \pi_k, \\
0 \leq \pi^{(j)}_k \leq 1 \quad (k \in U, j = 1, \ldots, M),
\]

\[
\sum_{k \in U} \pi^{(j)}_k = n \quad (j = 1, \ldots, M).
\]
The splitting method

Splitting method into \( M \) parts

\[
\begin{bmatrix}
\pi_1 \\
\vdots \\
\pi_k \\
\vdots \\
\pi_N
\end{bmatrix}
\]

\[
\begin{bmatrix}
\lambda_1 \\
\lambda_j \\
\lambda_M
\end{bmatrix}
\]

Figure: Splitting into \( M \) parts
Denote by $\pi(1), \ldots, \pi(k), \ldots, \pi(N)$ the ordered inclusion probabilities. Next, define

$$\lambda = \min\{1 - \pi(N-n), \pi(N-n+1)\},$$

$$\pi^{(1)}(k) = \begin{cases} 0 & \text{if } k \leq N - n \\ 1 & \text{if } k > N - n, \end{cases}$$

$$\pi^{(2)}(k) = \begin{cases} \frac{\pi(k)}{1 - \lambda} & \text{if } k \leq N - n \\ \frac{\pi(k) - \lambda}{1 - \lambda} & \text{if } k > N - n. \end{cases}$$
Example: Splitting tree for the minimal support design

(0.07, 0.17, 0.41, 0.61, 0.83, 0.91)

0.59

0.41

(0, 0, 0, 1, 1, 1)

0.585

(0.171, 0.415, 1, 0.049, 0.585, 0.780)

(0, 0, 1, 0, 1, 1)

0.415

(0.471, 1, 1, 0.118, 0, 0.471)

(0, 1, 1, 0, 0, 1)

0.471

0.529

(0.778, 1, 1, 0.222, 0, 0)

(0, 1, 1, 0, 0, 1)

0.778

0.222

(1, 1, 1, 0, 0, 0)

(0, 1, 1, 1, 0, 0)
Splitting into simple random sampling

\[ \lambda = \min \left\{ \pi(1) \frac{N}{n}, \frac{N}{N-n} (1 - \pi(N)) \right\}, \quad (7) \]

and compute, for \( k \in U \),

\[ \pi^{(1)}(k) = \frac{n}{N}, \quad \pi^{(2)}(k) = \frac{\pi_k - \lambda \frac{n}{N}}{1 - \lambda}. \]

If \( \lambda = \pi(1) N/n \), then \( \pi^{(2)}(1) = 0 \); if \( \lambda = (1 - \pi(N)) N/(N - n) \), then \( \pi^{(2)}(N) = 1 \). At the next step, the problem is thus reduced to a selection of a sample of size \( n - 1 \) or \( n \) from a population of size \( N - 1 \). In at most \( N - 1 \) steps, the problem is solved.
Splitting tree for splitting into simple random sampling

(0.07, 0.17, 0.41, 0.61, 0.83, 0.91)

0.14 0.86

(0.5, 0.5, 0.5, 0.5, 0.5, 0.5)

0.058 0.932

(0, 0.116, 0.395, 0.628, 0.884, 0.977)

0.086 0.914

(0, 0.086, 0.383, 0.630, 0.901, 1)

0.045 0.954

(0, 0, 0.358, 0.657, 0.985, 1)

0.688 0.312

(0, 0, 0.667, 0.667, 0.667, 1)

(0, 0, 0.667, 0.667, 1, 1)

(0, 0, 0.5, 0.5, 0.5, 1)

(0, 0, 0.5, 0.5, 1, 1)

(0, 0, 0, 1, 1, 1)
Pivotal Method

At each step, only two units are modified: $i$ and $j$.

Two cases: If $\pi_i + \pi_j > 1$, then

$$\lambda = \frac{1 - \pi_j}{2 - \pi_i - \pi_j},$$

$$\pi^{(1)}_{k} = \begin{cases} 
\pi_k & k \in U \setminus \{i,j\} \\
1 & k = i \\
\pi_i + \pi_j - 1 & k = j,
\end{cases}$$

$$\pi^{(2)}_{k} = \begin{cases} 
\pi_k & k \in U \setminus \{i,j\} \\
\pi_i + \pi_j - 1 & k = i \\
1 & k = j.
\end{cases}$$
Pivotal Method

If \( \pi_i + \pi_j < 1 \), then

\[
\lambda = \frac{\pi_i}{\pi_i + \pi_j},
\]

\[
\pi_k^{(1)} = \begin{cases} 
\pi_k & k \in U \setminus \{i, j\} \\
\pi_i + \pi_j & k = i \\
0 & k = j,
\end{cases}
\]

and \( \pi_k^{(2)} = \begin{cases} 
\pi_k & k \in U \setminus \{i, j\} \\
0 & k = i \\
\pi_i + \pi_j & k = j.
\end{cases} \)
Brewer’s Method

Brewer and Hanif (1983, p.26)
Brewer (1975)
draw by draw procedure

\[
\lambda_j = \left\{ \sum_{z=1}^{N} \frac{\pi_z(n - \pi_z)}{1 - \pi_z} \right\}^{-1} \frac{\pi_j(n - \pi_j)}{1 - \pi_j}.
\]

Next, we compute

\[
\pi_k^{(j)} = \begin{cases} 
\frac{\pi_k(n - 1)}{n - \pi_j} & \text{if } k \neq j \\
1 & \text{if } k = j.
\end{cases}
\]
Brewer’s Method

The validity derives from the following result:

**Theorem**

\[ \sum_{j=1}^{N} \lambda_j \pi_k^{(j)} = \pi_k, \]

for all \( k = 1, \ldots, N, \)


