



HAL
open science

Dirichlet and multinomial distributions: properties and uses in Jags

Isabelle Albert, Jean-Baptiste Denis

► **To cite this version:**

Isabelle Albert, Jean-Baptiste Denis. Dirichlet and multinomial distributions: properties and uses in Jags. 2012. hal-01263601

HAL Id: hal-01263601

<https://hal.science/hal-01263601>

Submitted on 6 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Dirichlet and multinomial distributions: properties and uses in Jags

Isabelle Albert¹

Jean-Baptiste Denis²

Rapport technique 2012-5, 28 pp.

Unité Mathématiques et Informatique Appliquées

INRA

Domaine de Vilvert

78352 Jouy-en-Josas cedex

France

¹Isabelle.Albert@Paris.Inra.Fr

²Jean-Baptiste.Denis@Jouy.Inra.Fr

© 2012 INRA

Previous versions of this technical report have been elaborated when preparing the paper *A Bayesian Evidence Synthesis for Estimating Campylobacteriosis Prevalence* published in Risk Analysis, 2011, (31, 1141-1155).

In this second edition some typos were corrected and a new proposal in §4.5 for the use of gamma distributions was introduced.

The last section of this report about JAGS implementation gained benefit from exchanges with Sophie ANCELET, Natalie COMMEAU and Clémence RIGAUX.

(24 juin 2012)
(30 juillet 2012)

ABSTRACT

Some properties of the Dirichlet and multinomial distributions are provided with a focus towards their use in Bayesian statistical approaches, particularly with the JAGS software.

Key Words Dirichlet - multinomial - probability - constraints - Bayesian

RÉSUMÉ

Quelques propriétés des distributions de Dirichlet et multinomiales sont explorées dans l'optique de leur utilisation en statistique bayésienne, notamment avec le logiciel JAGS.

Mots clef Dirichlet - multinomiale - probabilité - contraintes - bayésien

Contents

| | | |
|----------|--|-----------|
| 1 | The Dirichlet distribution | 7 |
| 1.1 | Definition | 7 |
| 1.2 | Moments | 8 |
| 1.3 | Marginal distributions (and distribution of sums) | 8 |
| 1.4 | Conditional distributions | 9 |
| 1.5 | Construction | 9 |
| 1.5.1 | With independent Gamma distributions | 9 |
| 1.5.2 | With independent Beta distributions | 10 |
| 2 | The multinomial distribution | 10 |
| 2.1 | Definition | 10 |
| 2.2 | Moments | 11 |
| 2.3 | Marginal distributions (and distributions of sums) | 11 |
| 2.4 | Conditional distributions | 12 |
| 2.5 | Construction | 12 |
| 2.5.1 | With independent Poisson distributions | 12 |
| 2.5.2 | With conditional Binomial distributions | 12 |
| 3 | A global scheme | 13 |
| 3.1 | Standard results of conjugation | 13 |
| 3.2 | Global presentation | 13 |
| 3.3 | More intricate situations | 15 |
| 4 | Implementation with JAGS | 15 |
| 4.1 | Simulating a multinomial vector | 15 |
| 4.2 | Simulating a Dirichlet vector | 16 |
| 4.3 | Inferring from multinomial vectors with fixed sizes | 18 |
| 4.4 | Inferring from multinomial vectors with random sizes | 18 |
| 4.4.1 | Every multinomial has got a non null size | 18 |
| 4.4.2 | At least one of the multinomial data is null size | 20 |
| 4.5 | Inferring from a Dirichlet vector | 22 |
| A | Main R script | 24 |

Notations

p, q, r refers to probabilities.

u, v, w refers to random counts.

m, n, N, M refers to sample sizes.

i, j refers to generic subscripts and h, k to their upper value.

$Di(), Be(), Ga(), Mu(), Bi(), Po()$ will respectively denote the Dirichlet, Beta, Gamma, Multinomial, Binomial and Poisson probability distributions.

1 The Dirichlet distribution

1.1 Definition

(p_1, \dots, p_k) is said to follow a Dirichlet distribution with parameters (a_1, \dots, a_k)

$$(p_1, \dots, p_k) \sim Di(a_1, \dots, a_k)$$

if the following two conditions are satisfied:

1. $p_1 + p_2 + \dots + p_k = 1$ and all p_i are non negative.
2. $[p_1, p_2, \dots, p_{k-1}] = \frac{\Gamma(\sum_{i=1}^k a_i)}{\prod_{i=1}^k \Gamma(a_i)} \left[\prod_{i=1}^{k-1} p_i^{a_i-1} \right] \left[1 - \sum_{i=1}^{k-1} p_i \right]^{a_k-1}$

It is important to underline some points:

- Due to the functional relationship between the k variables (summation to one), their joint probability distribution is degenerated. This is why the density is proposed on the first $k - 1$ variables, the last one being given as $p_k = 1 - \sum_{i=1}^{k-1} p_i$.
- But, there is still a complete symmetry¹ between the k couples (p_i, a_i) and the density could have been defined on any of the k subsets of size $k - 1$.
- All a_i are strictly positive values².
- When $k = 2$, it is easy to see that $p_1 \sim Be(a_1, a_2)$ ³, this is why Dirichlet distributions are one of the possible generalisations of the family of Beta distributions. We will see later on that there are other links between these two distributions.
- A helpful interpretation of this distribution is to see the p_i variables as the length of successive segments of a rope of size unity, randomly cutted at $k - 1$ points with different weights (a_i) for each. The most straightforward case being when all a_i are equal to a : then all random segments have the same marginal distribution and greater is a less random they are.
- When all $a_i = 1$, the Dirichlet distribution reduces to the uniform distribution onto the simplex defined by $\sum_i p_i = 1; p_i > 0$ into R^k .

Possibly a super-parameterization of this distribution could be convenient for a better proximity with the multinomial

$$(a_1, \dots, a_k) \iff (A, \pi_1, \dots, \pi_k)$$

where $A = \sum_i a_i$ and $\pi_i = \frac{a_i}{A}$.

¹Indeed notice that $\left[1 - \sum_{i=1}^{k-1} p_i \right]$ is p_k .

²If not the Γ function are not defined.

³and that $p_2 \sim Be(a_2, a_1)$ plus $p_1 + p_2 = 1$.

1.2 Moments

Let denote $A = \sum_{i=1}^k a_i$.

$$E(p_i) = \frac{a_i}{A}$$

$$V(p_i) = \frac{a_i(A - a_i)}{A^2(1 + A)}$$

$$Cov(p_i, p_j) = \frac{-a_i a_j}{A^2(1 + A)} \quad \forall i \neq j$$

as a consequence

$$Cor(p_i, p_j) = -\sqrt{\frac{a_i a_j}{(A - a_i)(A - a_j)}} \quad \forall i \neq j$$

and

$$V(\mathbf{p}) = \frac{1}{A(1 + A)} \left(\text{diag}(\mathbf{a}) - \frac{1}{A} \mathbf{a} \mathbf{a}' \right) \quad (1)$$

as a consequence, one can check that $V(\sum_i p_i) = 0$ as expected.

Also when all a_i are greater than one, the vector $(\frac{a_i - 1}{A - k})$ is the mode of the distribution.

When r_i are non negative integers, a general formula for the moment is given by:

$$E\left(\prod_{i=1}^k (p_i)^{r_i}\right) = \frac{\prod_{i=1}^k \binom{a_i}{r_i}}{A^{\sum_{i=1}^k r_i}}$$

where the bracketted power is a special notation for

$$\begin{aligned} p^{[n]} &= \prod_{j=0}^{n-1} (p + j) \\ &= p(p + 1)(p + 2) \dots (p + n - 1) \end{aligned}$$

1.3 Marginal distributions (and distribution of sums)

When $h < k$, if $(p_1, \dots, p_k) \sim Di(a_1, \dots, a_k)$ then

$$(p_1, \dots, p_h, (p_{h+1} + \dots + p_k)) \sim Di(a_1, \dots, a_h, (a_{h+1} + \dots + a_k))$$

As a consequence:

- Generally

$$((p_1 + \dots + p_h), (p_{h+1} + \dots + p_k)) \sim Di((a_1 + \dots + a_h), (a_{h+1} + \dots + a_k)) \quad (2)$$

a Dirichlet of dimension two, and $S_h = p_1 + \dots + p_h$ follows a Beta distribution with the same parameters.

- As a particular case: $p_i \sim Be(a_i, A - a_i)$.

1.4 Conditional distributions

When $h < k$, if $(p_1, \dots, p_k) \sim Di(a_1, \dots, a_k)$ then

$$\left(\frac{1}{\sum_{i=1}^h p_i} (p_1, \dots, p_h) \mid p_{h+1}, \dots, p_k \right) \sim Di(a_1, \dots, a_h)$$

An important consequence of this formulae is that $\frac{1}{\sum_{i=1}^h p_i} (p_1, \dots, p_h)$ is independent of (p_{h+1}, \dots, p_k) since the distribution doesn't depend on its value. So we can also state that

$$\left(\frac{1}{\sum_{i=1}^h p_i} (p_1, \dots, p_h) \right) \sim Di(a_1, \dots, a_h)$$

with the special case of the marginal distribution of the proportion of the sum of $g < h$ components among the h :

$$\frac{\sum_{i=1}^g p_i}{\sum_{i=1}^h p_i} \sim Be((a_1 + \dots + a_g), (a_{g+1} + \dots + a_h)) \quad (3)$$

A specific property of this kind is sometimes expressed as the *complete neutral property* of a Dirichlet random vector. It states that whatever is j

$$(p_1, p_2, \dots, p_j) \text{ and } \frac{1}{1 - (p_1 + \dots + p_j)} (p_{j+1}, \dots, p_k) \text{ are independent.}$$

Then $p_1, \frac{p_2}{1-p_1}, \frac{p_3}{1-p_1-p_2}, \dots, \frac{p_{k-1}}{1-p_1-p_2-\dots-p_{k-2}}$ are mutually independent random variables and

$$\frac{p_j}{1 - \sum_{u=1}^{j-1} p_u} \sim Be\left(a_j, \sum_{u=1}^{j-1} a_u\right).$$

Of course, any order of the k categories can be used for the construction of this sequence of independent variables.

1.5 Construction

1.5.1 With independent Gamma distributions

Let $\lambda_i, i = 1, \dots, k$ a series of independently distributed $Ga(a_i, b)$ variables⁴. Then

$$(p_1, \dots, p_k) = \frac{1}{\sum_{i=1}^k \lambda_i} (\lambda_1, \dots, \lambda_k) \sim Di(a_1, \dots, a_k)$$

⁴Here we use the parameterization of BUGS, the second parameter is the rate, not the scale so that $E(\lambda_i) = \frac{a_i}{b}$ and $V(\lambda_i) = \frac{a_i}{b^2}$. Remind that a $Ga(a, 1)$ distributed variable is no more than a $\frac{1}{2}\chi_{2a}^2$. Note that the second parameter is a scaling parameter, so it can be set to any common values to all λ s, in particular it can be imposed to be unity ($b = 1$) without any loss of generality.

1.5.2 With independent Beta distributions

Let $q_i, i = 1, \dots, k - 1$ a series of independently distributed $Be(\alpha_i, \beta_i)$ distributions. Let us define recursively from them the series of random variables

$$\begin{aligned} p_1 &= q_1 \\ p_i &= q_i \prod_{j=1}^{i-1} (1 - q_j) \quad i = 2, \dots, k - 1 \\ p_k &= \prod_{j=1}^{k-1} (1 - q_j) \end{aligned}$$

then

$$(p_1, \dots, p_k) \sim Di(a_1, \dots, a_k)$$

when for $i = 1, \dots, k - 1$:

$$\begin{aligned} \alpha_i &= a_i \\ \beta_i &= \sum_{u=i+1}^k a_u. \end{aligned}$$

That implies that the following equalities must be true:

$$\beta_i = \beta_{i-1} + \alpha_i \quad \forall i = 2, \dots, k - 1.$$

2 The multinomial distribution

In a Bayesian statistical framework, the Dirichlet distribution is often associated to multinomial data sets for the prior distribution⁵ of the probability parameters, this is the reason why we will describe it in this section, in a very similar way.

2.1 Definition

(u_1, \dots, u_k) is said to follow a multinomial distribution with parameters (N, p_1, \dots, p_k)

$$(u_1, \dots, u_k) \sim Mu(N, p_1, \dots, p_k)$$

if the following conditions are satisfied:

1. $p_1 + p_2 + \dots + p_k = 1$ and all p_i are non negative.
2. $u_1 + u_2 + \dots + u_k = N$ and all u_i are non negative integers.
3. $[u_1, u_2, \dots, u_{k-1}] = \frac{\Gamma(N+1)}{\prod_{i=1}^k \Gamma(u_i+1)} \left(\prod_{i=1}^{k-1} p_i^{u_i} \right) (p_k)^{N - \sum_{i=1}^{k-1} u_i}$

It is important to underline some points:

⁵for conjugation properties, cf. §3.1

- Due to the functional relationship between the k variables (summation to N , a known value), their joint probability distribution is degenerated. This is why the density is proposed on the first $k - 1$ variables, the last one being given as $u_k = N - \sum_1^{k-1} u_i$.
- There is a complete symmetry between the k couples (u_i, p_i) .
- When $P = 2$, it is easy to see that $u_1 \sim Bi(N, p_1)$, this is why the multinomial distribution is a generalisation of the binomial distributions. We will see later on that there are other links between these two distributions.

2.2 Moments

The expectations are given by

$$E(u_i) = Np_i$$

The variances are given by

$$V(u_i) = Np_i(1 - p_i)$$

The covariances are given by

$$Cov(u_i, u_j) = -Np_i p_j$$

So the variance matrix reads

$$V(\mathbf{u}) = N(\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}')$$

and $V(\sum_i u_i) = 0$ as it must be since $\sum_i u_i = N$, a constant. Finally, the correlation is

$$Cor(u_i, u_j) = \frac{-\sqrt{p_i p_j}}{\sqrt{(1 - p_i)(1 - p_j)}} \text{ for } i \neq j$$

2.3 Marginal distributions (and distributions of sums)

When $h < k$, if $(u_1, \dots, u_k) \sim Mu(N, p_1, \dots, p_k)$ then

$$(u_1, \dots, u_h, (u_{h+1} + \dots + u_k)) \sim Mu(N, p_1, \dots, p_h, (p_{h+1} + \dots + p_k))$$

As a consequence:

- Generally

$$((u_1 + \dots + u_h), (u_{h+1} + \dots + u_k)) \sim Mu(N, (p_1 + \dots + p_h), (p_{h+1} + \dots + p_k)) \quad (4)$$

a multinomial of size two, and $S_h = u_1 + \dots + u_h$ follows a binomial distribution with the parameters $(N, (p_1 + \dots + p_h))$.

- As a particular case: $u_i \sim Bi(N, p_i)$.

2.4 Conditional distributions

When $h < k$, if $(u_1, \dots, u_k) \sim Mu(N, p_1, \dots, p_k)$ then

$$((u_1, \dots, u_h) \mid u_{h+1}, \dots, u_k) \sim Mu\left(N - \sum_{i=h+1}^k u_i, \frac{p_1}{\sum_{i=1}^h p_i}, \dots, \frac{p_h}{\sum_{i=1}^h p_i}\right) \quad (5)$$

An important consequence of this formulae is that (u_1, \dots, u_h) depends on (u_{h+1}, \dots, u_k) only through their sum, $(u_{h+1} + \dots + u_k)$. From this, it is not difficult to see that when $g < h$:

$$\left(\sum_{i=1}^g u_i \mid \sum_{i=1}^h u_i\right) \sim Bi\left(\sum_{i=1}^h u_i, \frac{\sum_{i=1}^g p_i}{\sum_{i=1}^h p_i}\right) \quad (6)$$

It can also be shown that $u_1, (u_2 \mid u_1), (u_3 \mid u_1 + u_2), \dots, (u_{k-1} \mid u_1 + \dots + u_{k-2})$ are independent and follow binomial distributions:

$$\left(u_j \mid \sum_{u=1}^{j-1} u_u\right) \sim Bi\left(N - \sum_{u=1}^{j-1} u_u, \frac{p_j}{1 - \sum_{u=1}^{j-1} p_u}\right). \quad (7)$$

2.5 Construction

2.5.1 With independent Poisson distributions

Let $w_i, i = 1, \dots, k$ a series of independently distributed $Po(\lambda_i)$ variables, then

$$\left(w_1, \dots, w_k \mid \sum_{i=1}^k w_i = N\right) \sim Mu\left(N, \frac{\lambda_1}{\sum_{i=1}^k \lambda_i}, \dots, \frac{\lambda_k}{\sum_{i=1}^k \lambda_i}\right).$$

So the (u_i) can be seen as a set of Poisson variables constrained to sum to N and having parameters proportional to the probabilities p_i .

Notice that this kind of condition (the sum be equal to a given value) is not affordable with BUGS softwares.

2.5.2 With conditional Binomial distributions

From (7), let $u_i, i = 1, \dots, k - 1$ a series of successive conditional random variables distributed as

$$\begin{aligned} u_1 &\sim Bi(N, p_1) \\ u_2 \mid u_1 &\sim Bi\left(N - u_1, \frac{p_2}{1 - p_1}\right) \\ &\dots \\ u_i \mid u_1, u_2, \dots, u_{i-1} &\sim Bi\left(N - u_1 - \dots - u_{i-1}, \frac{p_i}{1 - p_1 - \dots - p_{i-1}}\right) \end{aligned}$$

then

$$(u_1, \dots, u_k) \sim Mu(N, p_1, \dots, p_k).$$

Notice that, with a positive probability, N can be attained with less than k positive u_i , so the last binomial can be of size zero. This limit case is perfectly handled in JAGS.

Table 1: Probability distributions associated to the dag of Figure 1.

| | | |
|-------|---------------------------------|--|
| (1.a) | $(a) \mapsto (p)$ | $(p) \sim Di(a)$ |
| (1.a) | $((p), N) \mapsto (u)$ | $(u) \sim Mu(N, p)$ |
| (1.b) | $(a) \mapsto (\lambda)$ | $\lambda_i \sim Ga(a_i, 1)$ |
| (1.b) | $(\lambda) \dashrightarrow (p)$ | $p_i = \frac{\lambda_i}{\sum \lambda_u}$ |
| (1.b) | $((p), N) \mapsto (u)$ | $(u) \sim Mu(N, p)$ |
| (1.c) | $(a) \mapsto (\lambda)$ | $\lambda_i \sim Ga(a_i)$ |
| (1.c) | $(\lambda) \dashrightarrow (p)$ | $p_i = \frac{\lambda_i}{\sum \lambda_u}$ |
| (1.c) | $((\lambda), N) \mapsto (u)$ | $(u) \sim Mu\left(N, \left(\frac{\lambda_i}{\sum \lambda_u}\right)\right)$ |
| (1.d) | $(a) \mapsto (\lambda)$ | $\lambda_i \sim Ga(a_i 1)$ |
| (1.d) | $(\lambda) \dashrightarrow (p)$ | $p_i = \frac{\lambda_i}{\sum \lambda_u}$ |
| (1.d) | $(\lambda) \mapsto (w)$ | $w_i \sim Po(\lambda_i)$ |
| (1.d) | $(w) \dashrightarrow (u)$ | $(w) = (u)$ |
| (1.d) | $(w) \dashrightarrow N$ | $N = \sum w_u$ |

3 A global scheme

3.1 Standard results of conjugation

Beta-Binomial $p \sim Be(a_1, a_2)$ and $(u | N, p) \sim Bi(N, p)$ imply that

$$(p | N, u) \sim Be(a_1 + u, a_2 + N - u).$$

Dirichlet-Multinomial $(p_1, \dots, p_k) \sim Di(a_1, \dots, a_k)$ and $(u_1, \dots, u_k | N, p_1, \dots, p_k) \sim Mu(N, p_1, \dots, p_k)$ imply that

$$(p_1, \dots, p_k | N, u_1, \dots, u_k) \sim Di(a_1 + u_1, \dots, a_k + u_k).$$

Gamma-Poisson $\lambda \sim Ga(a, b)$ and $(w | \lambda) \sim Po(\lambda)$ imply that

$$(\lambda | w) \sim Ga(a + w, b + 1).$$

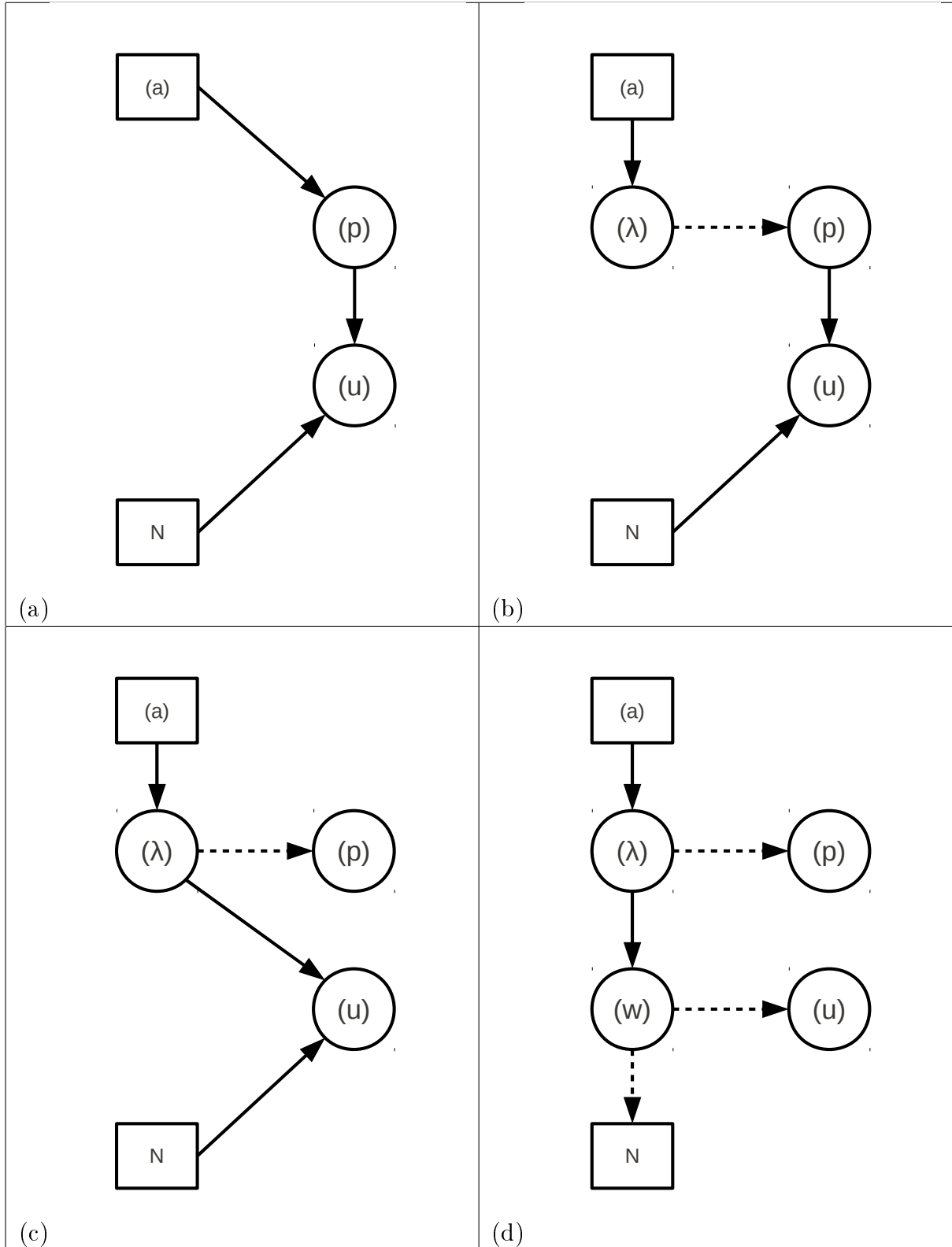
These results emphasize the relationships between the probability distributions we have previously considered.

3.2 Global presentation

Indeed, we can embed all the previously described distributions in the same framework allowing a better view of the relationships; four ways to attain the joint distribution of $((p), (u))$ are given in Figure 1 and detailed in Table 1.

The construction 1.a is the direct construction while 1.d is associated to the Gamma-Poisson modelling conditioned with the total number (N) giving rise with simple derivations to the desired random variables.

Figure 1: Different constructions of the joint density of $((p), (u))$. Fixed values are in squared boxes, random variables in circles; parentheses indicate vectors; dashed arcs indicate deterministic links, solid arcs random ones. Relationships are provided in Table 1



3.3 More intricated situations

It can occur that the multinomial vector is partially observed. Let us consider the case of $k = 3$, and suppose that we only know $D = (u_1, u_2 + u_3)$ that is we are not able to distinguish between the last two categories. The point is to derive the posterior. The model is :

prior $(p_1, p_2, p_3) \sim Di(a_1, a_2, a_3)$

likelihood $((u_1, u_2 + u_3) | (p_1, p_2, p_3)) \sim Mu(N, p_1, p_2 + p_3)$

Standard posterior computation provides

$$\begin{aligned} [(p_1, p_2, p_3) | (u_1, N)] &\propto \left(\prod_{i=1}^3 p_i^{a_i-1} \right) p_1^{u_1} (p_2 + p_3)^{N-u_1} \\ &= p_1^{u_1+a_1-1} \left(\sum_{i=0}^{N-u_1} \frac{(N-u_1)!}{i!(N-u_1-i)!} p_2^{i+a_2-1} p_3^{N-u_1-i+a_3-1} \right) \\ &= \sum_{i=0}^{N-u_1} \frac{(N-u_1)!}{i!(N-u_1-i)!} \{ p_1^{u_1+a_1-1} p_2^{i+a_2-1} p_3^{N-u_1-i+a_3-1} \} \end{aligned}$$

The posterior is a mixture of Dirichlet distributions, $Di(u_1 + a_1 - 1, i + a_2 - 1, N - u_1 - i + a_3 - 1)$, for $i = 0, \dots, N - u_1$ with binomial coefficients $\frac{(N-u_1)!}{i!(N-u_1-i)!}$. This could have been seen directly since it is a mixture of the completely observed cases associated to the partial observation, each being weighted with the probability of its occurrence:

$$[(p_1, p_2, p_3) | (u_1, N)] = \sum_{i=0}^{N-u_1} [(u_2 = i, u_3 = N - u_1 - i) | u_1] [(p_1, p_2, p_3) | (u_1, u_2 = i, u_3 = N - u_1 - i)]$$

and

$$[(u_2 = i, u_3 = N - u_1 - i) | u_1] = \frac{(N - u_1)!}{i!(N - u_1 - i)!}.$$

4 Implementation with JAGS

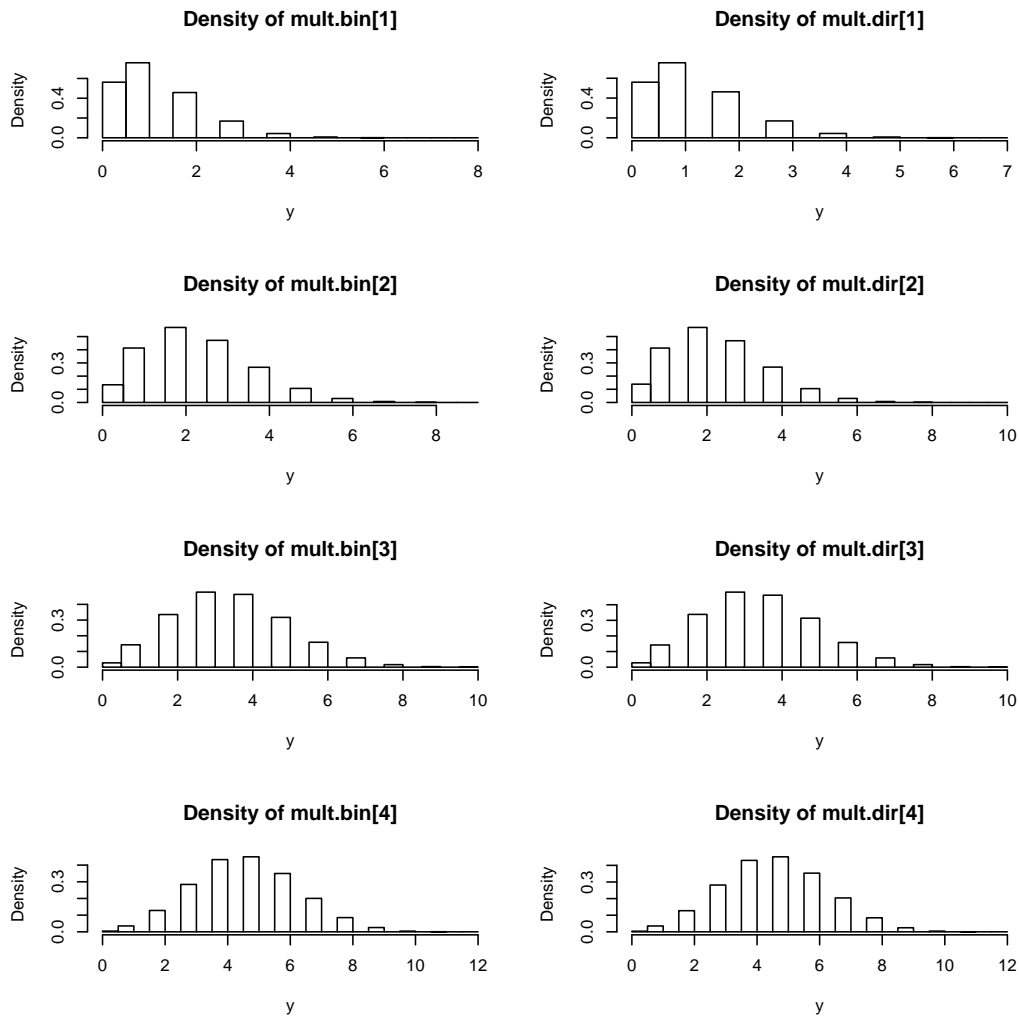
Here, we will consider how the above considerations could be used with JAGS to perform some statistical computations. The main R script used to obtain the presented results is provided in §A. There are two series of calculations : simulations and inferences. For the sake of the number of files, inferences have been performed in the same run which constrains to duplicate the data to get independent inferences.

4.1 Simulating a multinomial vector

In BUGS the multinomial distribution is available with `dmulti` and it can be also obtained with nested binomial (see §2.5.2) as exemplified in the following model code.

```
model {
  # (1) direct simulation
  mult.dir ~ dmulti(p[],N);
```


Figure 2: Simulating a multinomial vector in two different ways. First column with $Bi()$ and second column with $Mu()$.



```

# (2) simulation with binomial distributions
pp[1] <- p[1];
for (i in 2:k) { pp[i] <- p[i]/(1-sum(p[1:(i-1)]));}
mult.bin[1] ~ dbin(pp[1],N);
for (i in 2:(k-1)) {
  mult.bin[i] ~ dbin(pp[i],N-sum(mult.bin[1:(i-1)]));
}
mult.bin[k] <- N - sum(mult.bin[1:(k-1)]);
}

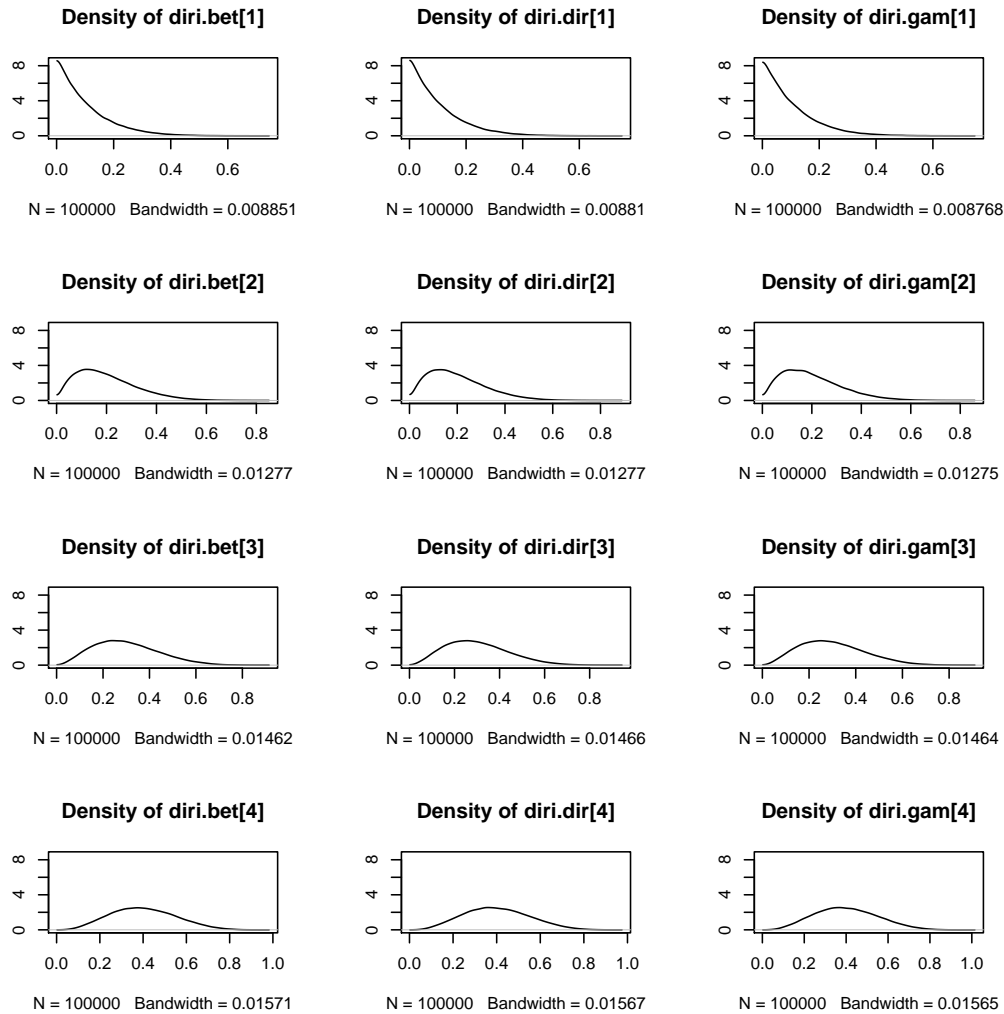
```

The binomial way requires some care. The obtained results are proposed in Figure 2: the two ways of simulation are not distinguishable.

4.2 Simulating a Dirichlet vector

In BUGS the Dirichlet distribution is available with `ddirch` and it can be also obtained with percentages of independent Gamma (see §1.5.1) or nested beta (see §1.5.2) as exemplified in the following model code.

Figure 3: Simulating a Dirichlet vector in three different ways. First column: with $Be()$; second column with $Di()$ and third column with $Ga()$.



```

model {
  # (1) direct simulation
  diri.dir ~ ddirch(a[]);
  # (2) simulation through Gamma distributions
  for (i in 1:k) {
    gam[i] ~ dgamma(a[i],1);
    diri.gam[i] <- gam[i] / sgam;
  }
  sgam <- sum(gam[]);
  # (3) simulation through Beta distributions
  for (i in 1:(k-1)) { b[i] <- sum(a[(i+1):k]);}
  for (i in 1:(k-1)) { q[i] ~ dbeta(a[i],b[i]);}
  diri.bet[1] <- q[1];
  for (i in 2:(k-1)) { diri.bet[i] <- q[i]*prod(1-q[1:(i-1)]);}
  diri.bet[k] <- prod(1-q[1:(k-1)]);
}

```

The beta way is quite tricky to implement. The obtained results are proposed in Figure 3: the three ways gave very similar results.

4.3 Inferring from multinomial vectors with fixed sizes

Using the different ways of simulating multinomial distributions, ones can try some inference about the probability parameter when some multinomial data have been observed. Here is the proposed BUGS model when the size of the multinomial is considered as fixed.

```
model {
  # (0) prior on the probabilities
  p1 ~ ddirch(a[]); p2 ~ ddirch(a[]);
  # (1) inference with dmult
  for (ii in 1:nbds) { YY1[ii,] ~ dmulti(p1[],nb[ii]);}
  # (2) inference with dbin
  pp[1] <- p2[1];
  for (i in 2:k) { pp[i] <- p2[i]/(1-sum(p2[1:(i-1)]));}
  for (ii in 1:nbds) {
    YY2[ii,1] ~ dbin(pp[1],nb[ii]);
    for (jj in 2:(k-1)) {
      YY2[ii,jj] ~ dbin(pp[jj],nb[ii]-sum(YY2[ii,1:(jj-1)]));
    }
  }
}
```

Results are graphically presented in Figure 4. The posterior distributions looks identical.

4.4 Inferring from multinomial vectors with random sizes

The same exercise can be made when the size of the multinomial data is supposed random (not been monitored by the experimenter). To do so, to comply BUGS requirements, the sizes are duplicated to give direct likelihood to the multinomial vectors and to the sizes; this has no consequence on the statistical analysis as far as no stochastic dependence is introduced between the parameter of the two parts.

We will distinguish two cases because the results are different.

4.4.1 Every multinomial has got a non null size

Here is the BUGS model code.

```
model {
  # (0.a) prior on the parameters
  p1 ~ ddirch(a[]);          p2 ~ ddirch(a[]);
  lambda1 ~ dunif(lin,lax); lambda2 ~ dunif(lin,lax);
  # (0.b) likelihood of the sizes
  for (ii in 1:nbds) { nb1[ii] ~ dpois(lambda1);}
  for (ii in 1:nbds) { nb2[ii] ~ dpois(lambda2);}
  # (1) inference with dmulti
  for (ii in 1:nbds) { YY1[ii,] ~ dmulti(p1[],nb1[ii]);}
  # (1) inference with dbin
  pp[1] <- p2[1];
  for (i in 2:k) { pp[i] <- p2[i]/(1-sum(p2[1:(i-1)]));}
}
```

Figure 4: Inferring the probability vector of a multinomial either with `dmult` (first column) or with nested binomials (second column).

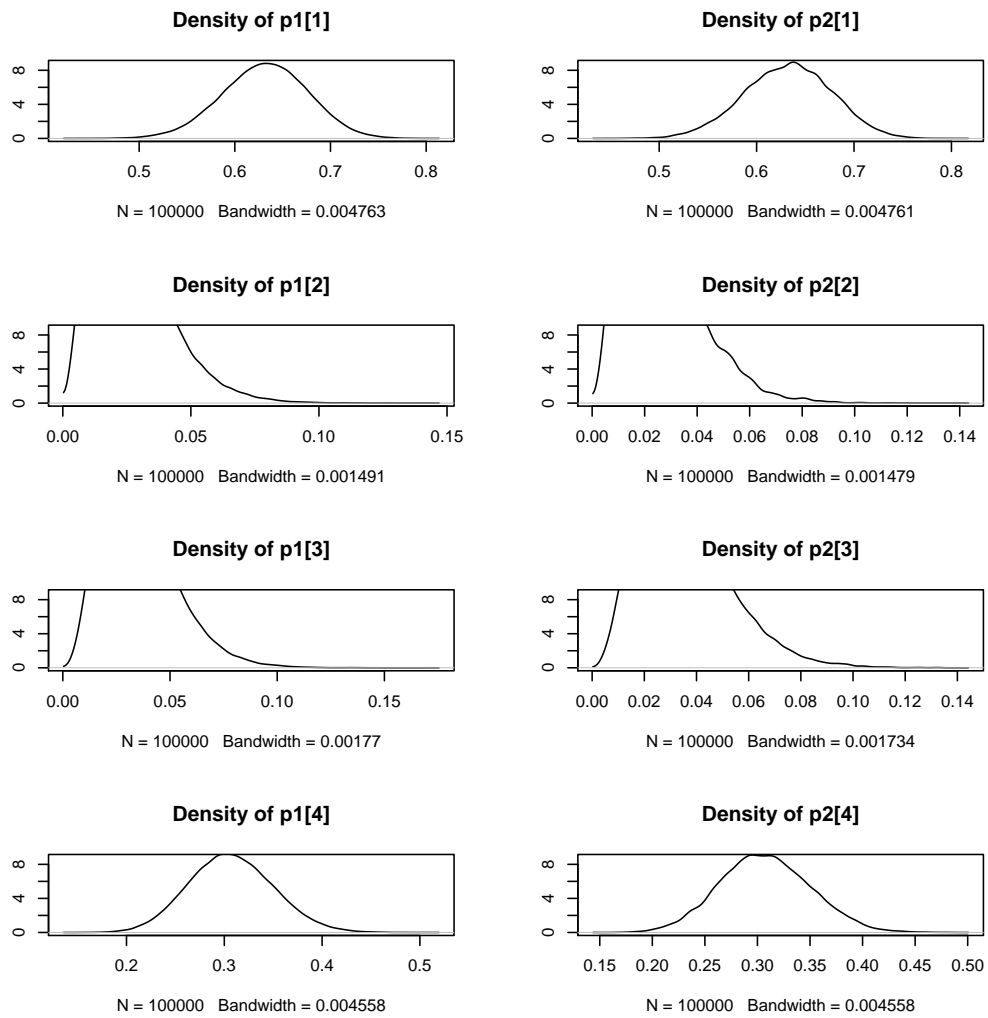
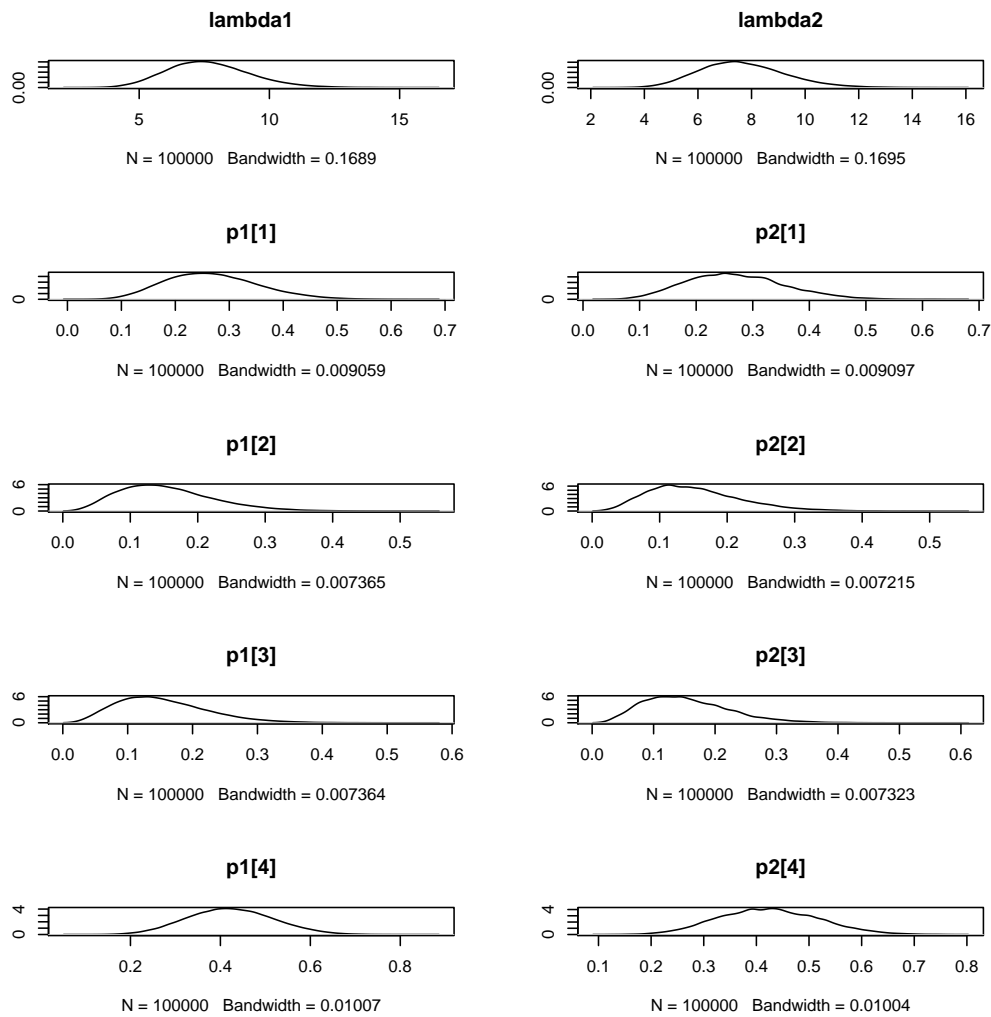


Figure 5: Inferring the probability vector of multinomial data in two different ways.



```

for (ii in 1:nbds) {
  YY2[ii,1] ~ dbin(pp[1],nb2[ii]);
  for (jj in 2:(k-1)) {
    YY2[ii,jj] ~ dbin(pp[jj],nb2[ii]-sum(YY2[ii,1:(jj-1)]));
  }
}
}

```

Results are proposed in Figure 5. Both ways produce the same inference.

4.4.2 At least one of the multinomial data is null size

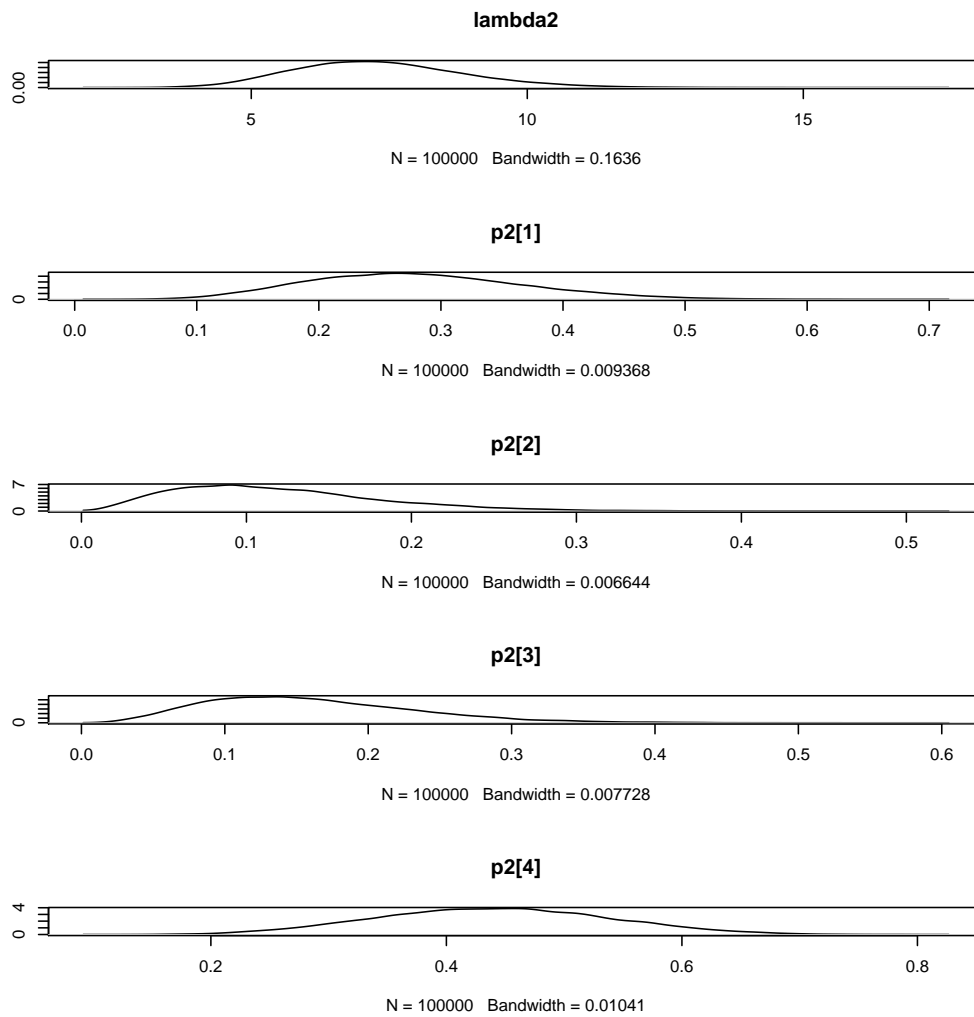
When the multinomial size are random, one cannot exclude the possibility of a null size. In that case, only the binomial way works; using `dmult` produces an error. The obtained results for the binomial tricks with some small data are shown in Figure 6 associated to the following code:

```

model {
  #

```

Figure 6: Inferring the probability vector of multinomial data when one of them is of size zero.



```

# prior on the parameters
p2 ~ ddirch(a[]);
lambda2 ~ dunif(lin,lax);
# likelihood of the data
for (ii in 1:nbds) {
  nb2[ii] ~ dpois(lambda2);
}
pp[1] <- p2[1];
for (i in 2:k) { pp[i] <- p2[i]/(1-sum(p2[1:(i-1)]));}
#
for (ii in 1:nbds) {
  YY2[ii,1] ~ dbin(pp[1],nb2[ii]);
  for (jj in 2:(k-1)) {
    YY2[ii,jj] ~ dbin(pp[jj],nb2[ii]-sum(YY2[ii,1:(jj-1)]));
  }
}
#
}

```

Notice that no information is born by empty data set and that a natural solution is to exclude

them from the statistical analysis, making the `dmult` solution always available.

4.5 Inferring from a Dirichlet vector

Even if not a so frequent situation, we can check what occurs when the observations are at the level of the Dirichlet variables. This implies an additional level in the model to focus the inference on the a parameter. An example is given in the following BUGS model code considering the three identified ways : (i) the direct `ddirch` use, (ii) the `dbeta` construction theoretically exact and (iii) a `dgamma` approximation.

The `dbeta` construction implies the computation of q_i , direct transformations of the observed p_i (see §1.5.2) and the code in the Appendix for the details.

The `dgamma` construction is only an approximation. There is no way for a transformation of the observations as done for the `dbeta` construction⁶. Here the parameterization of the gammas variables is defined such that their expectations and variances be equal to those of a Dirichlet vector, forgetting the existing covariance of (1). More precisely, with the notations of §1.1 and the definition of gamma distributions in Footnote 4 we use

$$Ga\left(\frac{a_i(1+A)}{A-a_i}, \frac{A(1+A)}{A-a_i}\right)$$

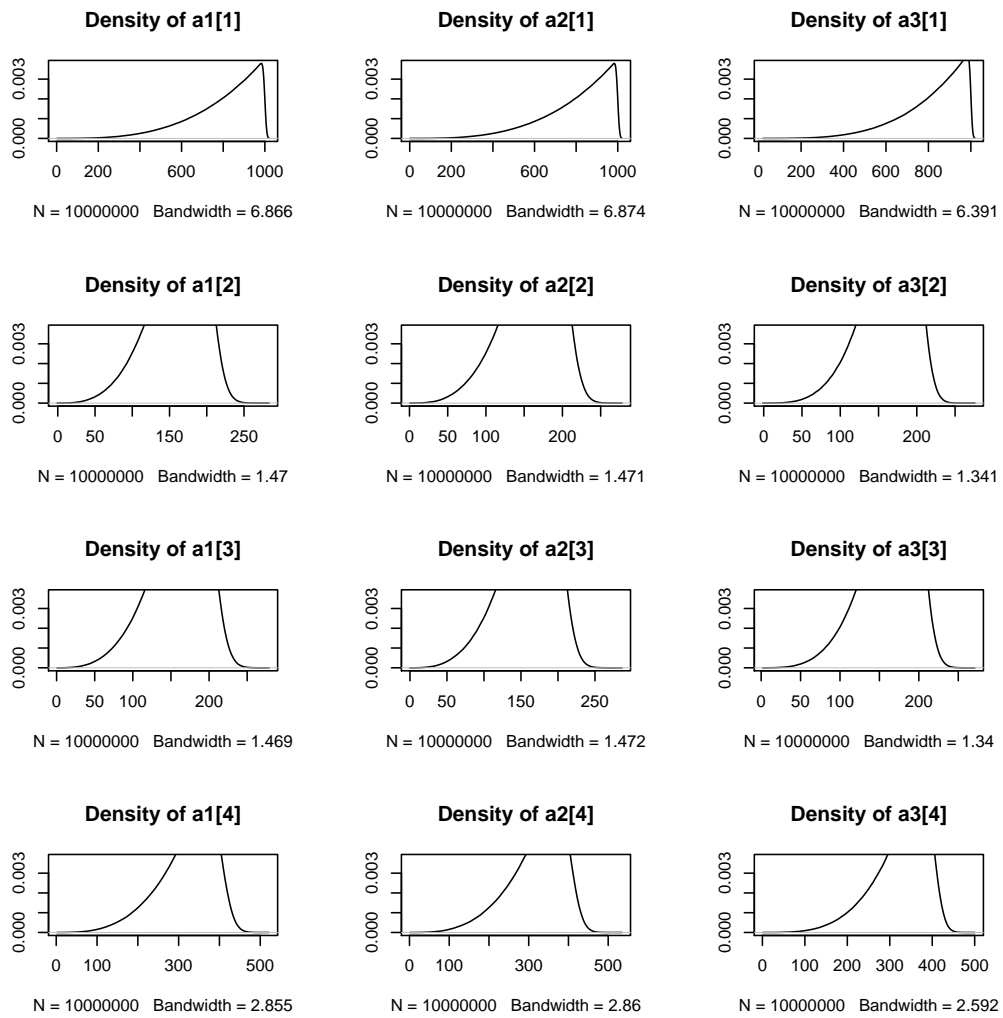
for the i th probability. Higher will be the dimension k of the Dirichlet, better we can imagine will be the approximation.

```
model {
  # (0) priors
  for (ii in 1:k) {
    a1[ii] ~ dunif(lin,lax);
    a2[ii] ~ dunif(lin,lax);
    a3[ii] ~ dunif(lin,lax);
  }
  # (1) likelihood with dirichlet
  p1 ~ ddirch(a1[]);
  # (2) likelihood with betas
  for (i in 1:(k-1)) {
    b[i] <- sum(a2[(i+1):k]);
    q2[i] ~ dbeta(a2[i],b[i]);
  }
  # (3) likelihood with gammas
  A <- sum(a3[]);
  for (i in 1:k) {
    p3[i] ~ dgamma(a3[i]*(1+A)/(A-a3[i]),
                  A*(1+A)/(A-a3[i]));
  }
}
```

The obtained results are displayed in Figure 7 , it can be seen that (i) as expected `ddirch` and `dbeta` ways are very similar and (ii) the `dgamma` approximation is quite good even if the dimension of the Dirichlet ($k = 4$) is not very high.

⁶If the sum of the gamma variables was known, we could multiply the p_i by it, but it is not the case.

Figure 7: Inferring Dirichlet parameter in three ways: (first column) `ddirch`, (second column) `dbeta` construction and (third column) `dgamma` approximation.



References

The most complete derivations were found in Johnson and Kotz, *Distribution in Statistics: Continuous Multivariate Distributions* (1972, John Wiley) §40.5, pages 231-235.

It is also of interest to consult some articles of the Wikipedia, for instance http://en.wikipedia.org/wiki/Neutral_vector.

Jags is a similar software to WinBugs available at <http://www-fis.iarc.fr/~martyn/software/jags/>. Contrary to WinBugs, it can be run under Linux (an also MSWindows). It has got extended facilities as vector/matrix/array definition of the nodes (to avoid loops).

A Main R script

Here is the complete R script used for all calculations related to §4. The called files to describe the model were given in the text above: `dirimul.mulsim.jam` in §4.1, `dirimul.dirsim.jam` in §4.2, `dirimul.mulinf1.jam` in §4.3, `dirimul.mulinf2.jam` in §4.4.1, `dirimul.mulinf3.jam` in §4.4.2 and `dirimul.dirinf1.jam` in §4.5,

```
# 12_06_06 12_06_07 12_06_20 12_06_21 12_07_18
library(rjags)

# initialization
nbsimu <- 100000; nburnin <- 10000;
#nbsimu <- 1000; nburnin <- 100;
files <- TRUE;
fitext <- "dirimul.res.txt";
if (files) { sink(fitext); sink();}
fifi <- function(nom) {
  res <- rep(NA,2);
  res[1] <- paste("dirimul",nom,"jam",sep=".");
  res[2] <- paste("dirimul",nom,"pdf",sep=".");
  res;
}

# Simulating a multinomial vector
".RNG.state" <- c(19900, 14957, 25769);
set.seed(1234);
nom <- "mulsim"; ff <- fifi(nom);
p <- 1:4/sum(1:4);
k <- length(p);
N <- 12;
tomonitor <- c("mult.dir","mult.bin");
constants <- list(p=p,k=k,N=N);
modele <- jags.model(file=ff[1],data=constants);
update(modele,10);
resultats <- coda.samples(modele,tomonitor,n.iter=nbsimu);
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
```

```

print(summary(resultats));
if (files) { sink();}
if (files) { pdf(ff[2]);}
par(mfcol=c(4,2));
densplot(resultats,show.obs=FALSE);
if (files) { dev.off();}
cat(nom,": <Enter> to continue\n");
if (!files) { scan();}

# Simulating a Dirichlet vector
".RNG.state" <- c(19900, 14957, 25769);
set.seed(1234);
nom <- "dirsim"; ff <- fifi(nom);
a <- 1:4;
k <- length(a);
tomonitor <- c("diri.dir","diri.gam","diri.bet");
constants <- list(a=a,k=k);
modele <- jags.model(file=ff[1],data=constants);
update(modele,10);
resultats <- coda.samples(modele,tomonitor,n.iter=nbsimu);
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(summary(resultats));
if (files) { sink();}
if (files) { pdf(ff[2]);}
par(mfcol=c(4,3));
densplot(resultats,show.obs=FALSE);
if (files) { dev.off();}
cat(nom,": <Enter> to continue\n");
if (!files) { scan();}

# Inferring multinomial vectors of fixed sizes
".RNG.state" <- c(19900, 14957, 25769);
set.seed(1234);
nom <- "mulinf1"; ff <- fifi(nom);
a <- rep(1,4);
k <- length(a);
tomonitor <- c("p1","p2");
YY1 <- matrix(c(1,0,2,7,
               70,2,1,27),byrow=TRUE,2,k);
nbds <- nrow(YY1);
nb <- apply(YY1,1,sum);
YY2 <- YY1;
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(YY1);
if (files) { sink();}
constants <- list(a=a,k=k,nbds=nbds,nb=nb,

```

```

        YY1=YY1,YY2=YY2);
modele <- jags.model(file=ff[1],data=constants);
update(modele,nburnin);
resultats <- coda.samples(modele,tomonitor,n.iter=nbsimu);
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(summary(resultats));
if (files) { sink();}
if (files) { pdf(ff[2]);}
par(mfcol=c(4,2));
densplot(resultats,show.obs=FALSE);
if (files) { dev.off();}
cat(nom,": <Enter> to continue\n");
if (!files) { scan();}

# Inferring multinomial vectors with positive random sizes
".RNG.state" <- c(19900, 14957, 25769);
set.seed(1234);
nom <- "mulinf2"; ff <- fifi(nom);
a <- rep(1,4);
k <- length(a);
lin <- 1; lax <- 100;
tomonitor <- c("p1","lambda1","p2","lambda2");
YY1 <- matrix(c(1,0,2,7,
               0,1,0,0,
               5,2,1,3),byrow=TRUE,3,k);
nbds <- nrow(YY1);
nb1 <- apply(YY1,1,sum);
YY2 <- YY1; nb2 <- nb1;
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(YY1);
if (files) { sink();}
constants <- list(a=a,k=k,nbds=nbds,lin=lin,lax=lax,
                 nb1=nb1,YY1=YY1,nb2=nb2,YY2=YY2
                 );
modele <- jags.model(file=ff[1],data=constants);
update(modele,nburnin);
resultats <- coda.samples(modele,tomonitor,n.iter=nbsimu);
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(summary(resultats));
if (files) { sink();}
if (files) { pdf(ff[2]);}
par(mfcol=c(5,2));
for (ip in c(1,3:6,2,7:10)) {

```

```

densplot(resultats[[1]][,ip],show.obs=FALSE,
          main=dimnames(resultats[[1]])[[2]][ip]);
}
if (files) { dev.off();}
cat(nom,": <Enter> to continue\n");
if (!files) { scan();}

# Inferring multinomial vectors with positive and nought random sizes
".RNG.state" <- c(19900, 14957, 25769);
set.seed(1234);
nom <- "mulinf3"; ff <- fifi(nom);
a <- rep(1,4);
k <- length(a);
lin <- 1; lax <- 100;
tomonitor <- c("p2","lambda2");
YY1 <- matrix(c(1,0,2,7,
                0,0,0,0,
                5,2,1,3),byrow=TRUE,3,k);
nbds <- nrow(YY1);
nb1 <- apply(YY1,1,sum);
YY2 <- YY1; nb2 <- nb1;
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(YY1);
if (files) { sink();}
constants <- list(a=a,k=k,nbds=nbds,lin=lin,lax=lax,
                 nb2=nb2,YY2=YY2
                 );
modele <- jags.model(file=ff[1],data=constants);
update(modele,nburnin);
resultats <- coda.samples(modele,tomonitor,n.iter=nbsimu);
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(summary(resultats));
if (files) { sink();}
if (files) { pdf(ff[2]);}
par(mfcol=c(5,1));
for (ip in 1:5) {
  densplot(resultats[[1]][,ip],show.obs=FALSE,
            main=dimnames(resultats[[1]])[[2]][ip]);
}
if (files) { dev.off();}
cat(nom,": <Enter> to continue\n");
if (!files) { scan();}

# Inferring Dirichlet vectors
".RNG.state" <- c(19900, 14957, 25769);
set.seed(1234);

```

```

nom <- "dirinf1"; ff <- fifi(nom);
p <- c(5,1,1,2);
k <- length(p);
lin <- 1; lax <- 1000;
tomonitor <- c("a1","a2","a3");
p <- p/sum(p);
p1 <- p2 <- p3 <- p;
q2 <- p2[-4];
q2[2] <- q2[2]/(1-q2[1]);
q2[3] <- q2[3]/(1-q2[1])/(1-q2[2]);
constants <- list(k=k,lin=lin,lax=lax,p1=p1,q2=q2,p3=p3);
modele <- jags.model(file=ff[1],data=constants);
update(modele,nbburnin*100);
resultats <- coda.samples(modele,tomonitor,n.iter=nbsimu*100);
if (files) {
  sink(fitext,append=TRUE);
  cat("\n\n\n",nom,"\n\n");
}
print(summary(resultats));
if (files) { sink();}
if (files) { pdf(ff[2]);}
par(mfcol=c(4,3));
densplot(resultats,show.obs=FALSE);
if (files) { dev.off();}
cat(nom,": <Enter> to continue\n");
if (!files) { scan();}

# The end
cat("C'est FINI !\n");

```