RanCh: A package for abstract discrete *Ran*dom *Ch*oice analysis

William McCausland

2019-09-08

Introduction

This package provides tools for a research project whose purpose is to help us better understand the foundations of stochastic discrete choice. It includes datasets compiled from the context effects literature, the stochastic intransitivity literature, and from some recent experiments where we observe choices from all doubleton and larger subsets of some universe of objects. It provides graphical tools illustrating likelihood function and posterior density contours, as well as regions, in the space of choice probabilities, defined by various stochastic choice axioms, context effects and other conditions. Finally, it provides tools for parametric and non-parametric inference subject to various combinations of discrete choice axioms, as well as the testing of said axioms.

Data sets

We provide some datasets from the context effects literature, the stochastic intransitivity literature and from some recent experiments. The data sets are organized in similar ways, allowing for the use of a common set of analytical tools, described below.

Data sets from the context effects literature

One of the data sets in this package is the choice count data from the experiments described in Amos Tversky (1972), which introduces the similarity effect. The $3 \times 8 \times 7 \times 3$ array `T_1972_counts` contains the data for the three choice domains ("Dots", "Gambles" and "Applicants") and eight subjects of that experiment. The first dimension of the array is indexed by choice domain, either as an integer from one to three, or as one of the strings "Dots", "Gambles" or "Applicants". The second dimension is indexed by subject identifier, an integer from one to eight. The third dimension is indexed by choice set and the fourth by choice object.

Here and in many other contexts, choice sets and choice objects each have different kinds of indices. In this dataset, the object names are x, y and z, as in the paper; they can be indexed using the strings `"x"`, `"y"` and `"z"` or the integers 1, 2, and 3, respectively. Choice sets, subsets of the choice universe $\{x, y, z\}$, can be indexed using the strings `"x"`, `"y"`, `"xy"`, ..., `"xyz"` or by integers we will call set indices. Each digit (or bit) in the binary representation of a set index in an inclusion indicator: object $i \in \{1, \ldots, 3\}$ is in the set if and only if the i'th digit from the right is 1. For example, the sets $\{x, z\}$, $\{y, z\}$ and $\{x, y, z\}$ have indices that are binary 101, 110 and 111, or decimal 5, 6 and 7. This convention for indexing sets has some important computational advantages. The singleton set with object i is represented as $2^i - 1$. The bitwise "or" of the indices of two sets gives the index of their union; the bitwise "and", the index of their intersection.

Thus the number of times subject five chose object x from choice set $\{x, y, z\}$ of domain `Dots` is `T_1972_counts["Dots", 5, "xyz", "x"]` or `T_1972_counts[1, 5, 7, 1]`; names and integer indices can be mixed. Counts of impossible events, such as choosing x from $\{y, z\}$ have the value `NA` rather than zero. This is deliberate; accessing information that doesn't make sense should lead to an immediate (thus relatively easy to find) error.

Here is a cross section of counts for the "Gambles" domain and the first subject, showing the choice sets (the only choice sets used in the experimental design) $\{x, z\}$, $\{y, z\}$ and $\{x, y, z\}$. For example, the number of times this subject chose gamble x when presented with a choice from $\{x, y, z\}$ is 2.

```r
print(T_1972_counts['Gambles', 1, c('xz', 'yz', 'xyz'), ], na.print='-')
        Object
  Subset x  y  z
     xz  7  - 13
     yz  - 10 10
    xyz  2 13 15
```

Data sets from the stochastic intransitivity literature

Another data set in the package is the count data from the experiment described in Regenwetter, Dana, and Davis-Stober (2011). The $3 \times 18 \times 31 \times 5$ `RDS_2011_counts` contains the data for the three choice domains ("Cash 1", "Cash 2" and "Noncash") and 18 subjects of that experiment. The data is organized in a similar way to that of `T_1972_counts`, although the universe of choice objects has five objects instead of three, and their names are a, b, c, d and e. The four dimensions are indexed by domain, subject number, subset and choice object. Thus the number of times subject 5 chose object a from choice set $\{a, b\}$ of domain `Cash 1` is `RDS_2011_counts['Cash 1', 5, 'ab', 'a']` or `RDS_2011_counts[1, 5, 3, 1]`. Again, counts of impossible events have the value `NA`.

Here is a cross section of counts for the "Cash 1" domain and the fifth subject, showing binary choice sets (the only choice sets used in the experimental design) only. For example, the number of times this subject chose gamble c when presented with a choice between b and c is 18.

```r
print(RDS_2011_counts['Cash 1', 5, c(3, 5, 6, 9, 10, 12, 17, 18, 20, 24), ], na.print='-')
        Object
  Subset a  b  c  d  e
     ab  1 19  -  -  -
     ac  2  - 18  -  -
     bc  -  2 18  -  -
     ad  1  -  - 19  -
     bd  -  0  - 20  -
     cd  -  -  0 20  -
     ae  0  -  -  - 20
     be  -  0  -  - 20
     ce  -  -  0  - 20
     de  -  -  -  1 19
#n_objects = 5
#print(RDS_2011_counts['Cash 1', n_objects, doubletons[1:choose(n_objects, 2)], ], na.print='-')
```

For this partcular domain and subject, all choice proportions are consistent with weak stochastic transitivity: relative to alphabetical order, the subject always chose the lower ranked object much more often than the higher ranked object.

Data sets where all subsets of the choice universe are presented

We provide data from three experiments in which individuals, or a sample from a population of individuals, make repeated choices from all doubleton and larger subsets of a universe of related choice objects. (In the experiment described in Amos Tversky (1972), the set $\{x, y\}$ is never presented; in the experiment in Regenwetter, Dana, and Davis-Stober (2011), only doubleton subsets of the universe are presented.)

For each of the three experiments, we provide four data objects. Here, we will take the population choice experiment described in McCausland, Marley and Davis-Stober (no paper yet) as an example. `MMS_2019_trials` shows all responses of all subjects in a discrete choice experiment. The first few variables are the name of the choice domain (a universe of five related choice objects), the subject identifier and the trial index giving the order in which a subject responded to the various choice sets presented to her. The next variable, `set`, is a

string showing which choice objects were presented at a given trial, the objects in alphabetical order. This is followed by `choice`, a letter indicating which object the subject chose from the given choice set. The variable `set_perm` shows the configuration of the objects as seen on screen by the subject. The subset and choice information is repeated in the values of the variables `set_index` and `choice_int`. The variable `set_index` is the set index, a binary representation of the choice set, as described above. Take for example the first row of the tqble below. The decimal value 30 is binary 11110, which indicates that the 2nd, 3rd, 4th and 5th elements of the set $\{a, b, c, d, e\}$ are in the choice set. This corresponds to the string `bcde` that is the value of the `set` variable.

```
head(MMS_2019_trials[c('domain', 'subject', 'trial', 'set', 'choice', 'set_perm', 'set_index', 'choice_
  # A tibble: 5 x 8
    domain        subject trial set   choice set_perm set_index choice_int
    <fct>           <int> <int> <fct> <fct>  <chr>        <int>      <int>
  1 Dots                1     1 bcde  e      ecbd            30          5
  2 Films               1     2 ad    d      ad               9          4
  3 Beer                1     3 abcde b      cdbae           31          2
  4 Female stars        1     4 cde   d      ecd             28          4
  5 Phone plans         1     5 ad    d      ad               9          4
```

The remaining variables are revealed preference indicators for given pairs of objects. The value of variable `ab` is 1 if object a is revealed preferred to object b, -1 if object b is revealed preferred to object a and zero otherwise. So for example, the choice of e from $\{b, c, d, e\}$ reveals e preferred to c, d, and b; see the first row of the following table, which repeats the first five observations of the variables `set` and `choice`, and includes the first five observations of the revealed preference indicators.

```
MMS_2019_trials %>% select(set, choice, matches("^[a-e]{2}$")) %>% head(5)
  # A tibble: 5 x 12
    set   choice    ab    ac    bc    ad    bd    cd    ae    be    ce    de
    <fct> <fct>  <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
  1 bcde  e          0     0     0     0     0     0     0    -1    -1    -1
  2 ad    d          0     0     0    -1     0     0     0     0     0     0
  3 abcde b         -1     0     1     0     1     0     0     1     0     0
  4 cde   d          0     0     0     0     0    -1     0     0     0     1
  5 ad    d          0     0     0    -1     0     0     0     0     0     0
```

One possible application is aggregation of preferences. Aggregating revealed preference indicators for the `Beer` domain gives the following table. The regular expression `^[a-e]{2}$` matches variable names that are strings of length two formed from the letters a through e.

```
MMS_2019_trials %>% filter(domain=="Beer") %>% select(matches("^[a-e]{2}$")) %>% colSums
   ab    ac    bc    ad    bd    cd    ae    be    ce    de
  -36    52   123    -9    -4  -132   -52   -36  -167   -52
```

In terms of net revealed preference, all values are consistent with the preference ranking $e \succ d \succ b \succ a \succ c$. That is, the number of times a higher ranked object is revealed preferred to a lower ranked object is always greater than the number of times the lower ranked object is revealed preferred to the higher ranked object.

Choice counts for the experiment are in the $32 \times 31 \times 5$ array `MMS_2019_counts`. The first dimension is indexed by choice domain, the second by choice subset and the third by realized choice. For example, the number of times any of the subjects choose object a from choice set $\{a, b\}$ of domain `Beer` is `MMS_2019_counts['Beer', 'ab', 'a']` or `MMS_2019_counts['Beer', 3, 1]`. Again, counts of impossible events have the value `NA`.

We can take cross sections of count data using the function `marginalize`. Here is a cross section of counts for the 2nd, 3rd and 5th objects of the `Beer` domain. For example, the number of times any subject choose beer b when presented with a choice between beers b and c is 36.

```
N_bce <- marginalize(MMS_2019_counts['Beer', , ], c(2, 3, 5))
print(N_bce, na.print='-')
```

```
       Object
  Subset  b c  e
     b     0 -  -
     c     - 0  -
    bc    36 4  -
     e     - -  0
    be    24 - 17
    ce     - 3 37
   bce    10 1 29
```

Note that the original object names are preserved. We will be using this very small count matrix as an example several times below.

Demographic information for the subjects in the same experiment is found in the data frame `MMS_2019_demographics`. Here is the demographic information for the first five subjects.

```
head(MMS_2019_demographics, 5)
  # A tibble: 5 x 3
    sex     age location
    <fct>  <int> <fct>
  1 Female    27 Quebec
  2 Male      35 Ontario
  3 Female    38 British Columbia
  4 Female    41 British Columbia
  5 Male      53 Ontario
```

In the YG experiment, each subject is presented with two waves of 16 trials. In each wave, the subject sees one choice set from each of the 16 choice domains. We can partition the complete dataset into two waves and test to see if choice probabilities are plausibly the same in both waves.

In the following code, we compute for every domain and every subset of size two or more, the p-value for the Pearson two-sample chi-squared test of the hypothesis that the counts in the two waves are drawn from the same multinomial distribution.

```
my.chisq.test <- function(A)
{
  x <- na.omit(t(A))
  if (nrow(x) > 1) {
    chi2 <- chisq.test(x)
    chi2$p.value
  } else NA
}
M <- apply(MG_2019_counts, c(1, 3), my.chisq.test)
hist(M, 20)
```
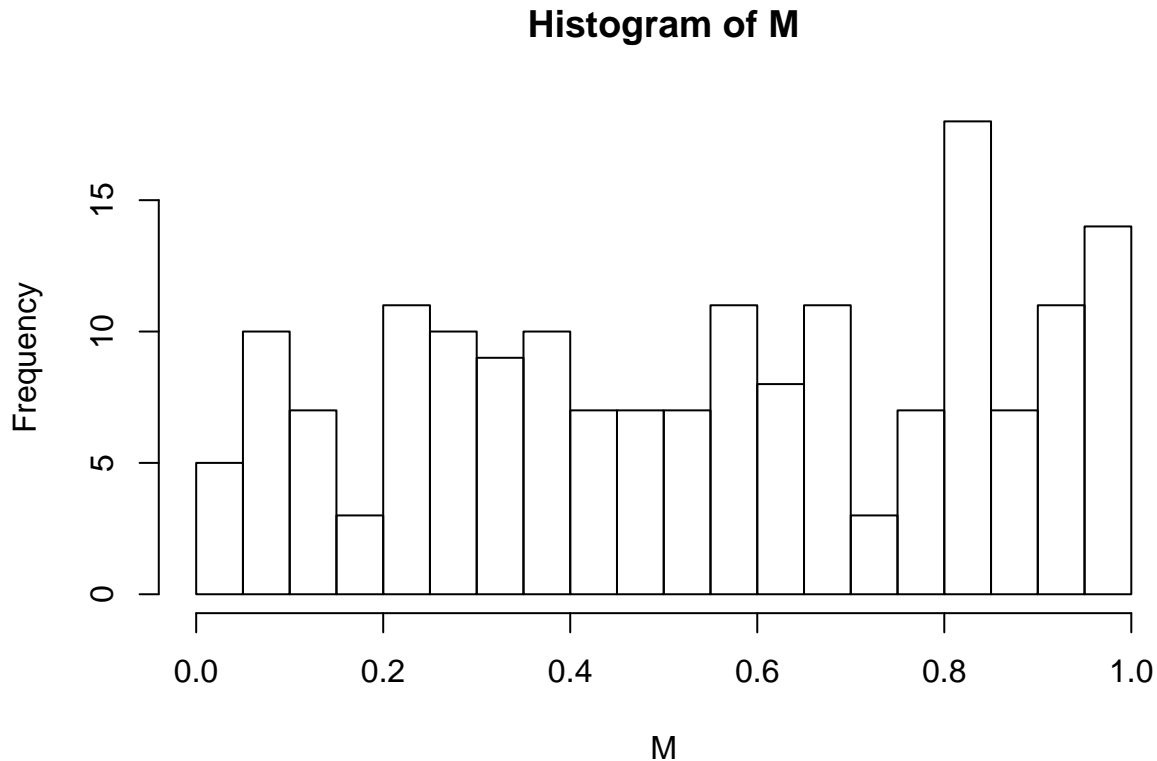
Histogram of M

Under the null hypotheses that the distributions are the same, the 176 different p-values are independent and uniformly distributed on $[0, 1]$. The histogram shows the realized p-values. The number of rejections at the five percent level is given by the height of the first histogram bar. Five rejections out of 176 is less than the 8.8 we would expect, on average, under the null.

Random Choice Structures

To make sense of most of the functions in this package, we need to introduce the concept of a Random Choice Structure (RCS), which organizes choice probabilities.

Let $T = (x_1, \ldots, x_n)$ be a universe of choice objects. When faced with a non-empty choice set $A \subseteq T$, a decision maker (DM) chooses a single object from A. The probability that the DM chooses $x \in A$ is denoted $P_A(x)$. A *Random Choice Structure* (RCS) is the complete specification of the $P_A(x)$, $x \in A \subseteq T$, and is denoted P.

The function `proportions` computes choice proportions from choice counts. The following code computes and displays the choice proportions that correspond to the cross section `N_bce` of counts constructed above. These choice proportions are an example of a RCS for the choice universe $\{b, c, e\}$.

```
P_bce <- proportions(N_bce)   # Compute proportions from count data
print(P_bce, na.print='-')    # and display.
        Object
  Subset         b      c         e
      b   1.0000000      -         -
      c           - 1.000         -
     bc   0.9000000 0.100         -
      e           -      - 1.0000000
     be   0.5853659      - 0.4146341
     ce           - 0.075 0.9250000
    bce   0.2500000 0.025 0.7250000
```

Note that we are using the convention that for singleton sets, the choice proportions for the only element of the set are equal to 1, even if the set is never presented. This convention makes it easier to check various choice axioms, by avoiding the need for special cases.

A big part of this package is a collection of tools for plotting binary and ternary probabilities in barycentric coordinate systems. The Wikipedia page on ternary plots gives a nice overview of plotting ternary probabilities or proportions in the regular 2-simplex, an equililateral triangle. The page on Barycentric coordinate systems gives more detail and generality.

The following code creates a figure displaying four points in barycentric coordinates, correponding to the three binary choice probability vectors and the ternary choice probability vector of the RCS `P_bce` we just constructed.

```
triplot(label=c('b', 'c', 'e')) # Set up ternary plot, with labels and grid
plot_P3(P_bce)                   # Plot points
```
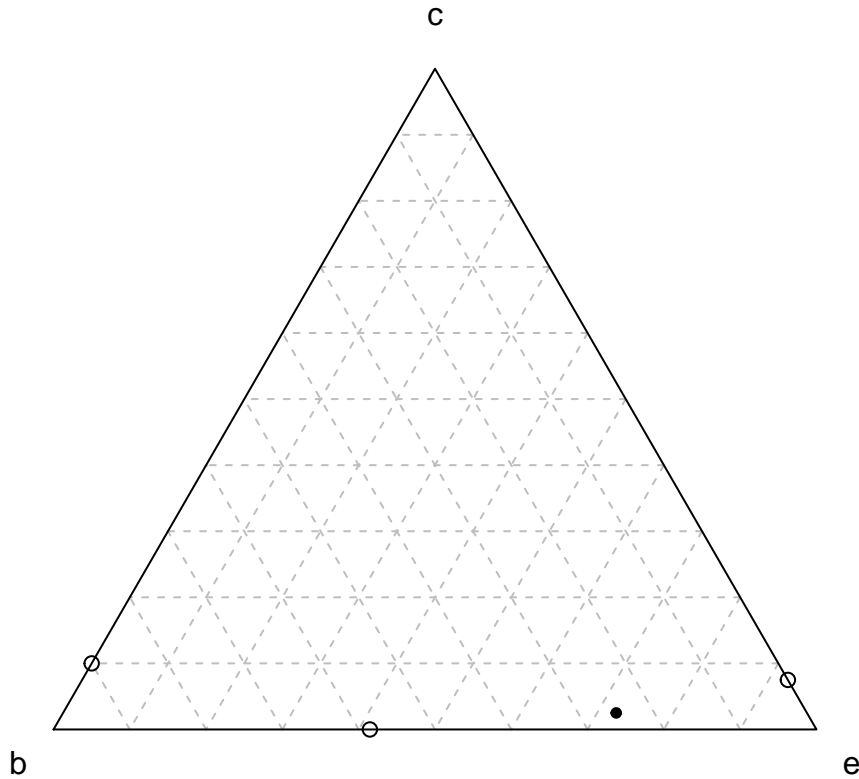

Each point in the triangle bce is a unique convex combination $\lambda_b b + \lambda_c c + \lambda_e e$ of the vertices b, c and e, and the vector $(\lambda_b, \lambda_c, \lambda_e)$ gives the coordinates of that point in Barycentric coordinates. Thus, the vertices b, c and e have Barycentric coordinates $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, respectively. We will interpret a point $(\lambda_b, \lambda_c, \lambda_e)$ as giving choice probabilities of b, c and e when the choice set $\{b, c, e\}$ is presented. Since triangle bce is equilateral, the distances of point $(\lambda_b, \lambda_c, \lambda_e)$ to the sides ce, be and bc of the triangle are fractions λ_b, λ_c and λ_e, respectively, of the height of the triangle.

Each of the rows 3, 5, 6 and 7 in the table above is represented by a point in the figure above. The hollow dot on the left side of the triangle gives the choice probabilities $p(b, c) = 0.9$ and $p(c, b) = 1 - p(b, c) = 0.1$. It is the convex combination $p(b, c)b + p(c, b)c$ of vertices b and c; it is also the point $(p(b, c), p(c, b), 0)$ in three dimensional barycentric coordinates. Similarly, the hollow point on the right side gives the choice probabilities $p(c, e)$ and $p(e, c)$; the hollow point on the base gives $p(b, e)$ and $p(e, b)$.

The ternary probability vector $P_{\{b,c,e\}}(\cdot)$ given in the final row of the table is indicated by a solid dot in the interior of the triangle. This point has Barycentric coordinates $(P_{\{b,c,e\}}(b), P_{\{b,c,e\}}(c), P_{\{b,c,e\}}(e)) = (0.25, 0.025, 0.725)$. We adopt the convention that binary probabilities are indicated by hollow dots and

ternary probability vectors by solid dots. In this way, we can tell the difference between a binary probability and a ternary probability vector that happens to be on the boundary of the triangle.

Axioms

Let Δ be the space of RCSs consistent with the axioms of probability; Δ is a Cartesian product of unit simplexes of various dimensions.

Various axioms, conditions, properties and hypotheses about probabilistic choice behaviour can be expressed as restrictions over the various choice probabilities of a RCS. Henceforth, we will use the term axiom as a generic term to include all such restrictions. Each restriction defines a subset of Δ.

Examples include weak, moderate and strong stochastic transitivity, regularity, the triangle inequality, random utility (the Block-Marshak inequalities) and the multiplicative inequality. Falmagne (1978) showed that the Block-Marshak inequalities are necessary and sufficient for random utility. A. Tversky (1972) and Sattath and Tversky (1976) establish that the Block-Marshak inequalities, moderate stochastic transitivity and the multiplicative inequality are all necessary conditions for, and thus testable implications of, the Elimination by Aspects model (EBA) introduced by Amos Tversky (1972).

Displaying cross sections of regions in barycentric coordinates

A random choice structure P satisfies

- *regularity* if for all $x \in A \subseteq B \subseteq T$, $P_A(x) \geq P_B(x)$,
- the *multiplicative inequality* if for all $x \in A, B \subseteq T$, $P_{A \cup B} \geq P_A(x)P_B(x)$,
- *random utility* if and only if for all $x \in A \subseteq T$,

$$\sum_{B \,:\, A \subseteq B \subseteq T} (-1)^{|B \setminus A|} P_B(x) \geq 0.$$

The various terms on the left hand side are the *Block-Marschak* terms; these can be computed using the function `BM_terms`.

The next code excerpt displays cross sections of the regularity and multiplicative inequality regions, both subsets of Δ. The blue triangle (pointing down) shows the region of ternary probabilities that is consistent with both regularity and the binary choice proportions given by the observed counts in `N_bce`: those binary choice proportions are displayed, as before, as hollow dots on the three sides of the triangle bce. Notice that the top side of the blue triangle is aligned with the lower of the two hollow dots on the left and right sides of the triangle bce; the left and right sides of the blue triangle meet at the hollow dot on the base of bce.

The red triangle (pointing up) shows the region of ternary probabilities consistent with the multiplicative inequality and the same binary choice proportions.

```
triplot(label=c('b', 'c', 'e'))  # Set up ternary plot, with labels and grid
plot_P3(P_bce)                    # Plot points
polygon(tritrafo(regularity_X3(P_bce)), border='blue')
polygon(tritrafo(multiplicative_X3(P_bce)), border='red')
```
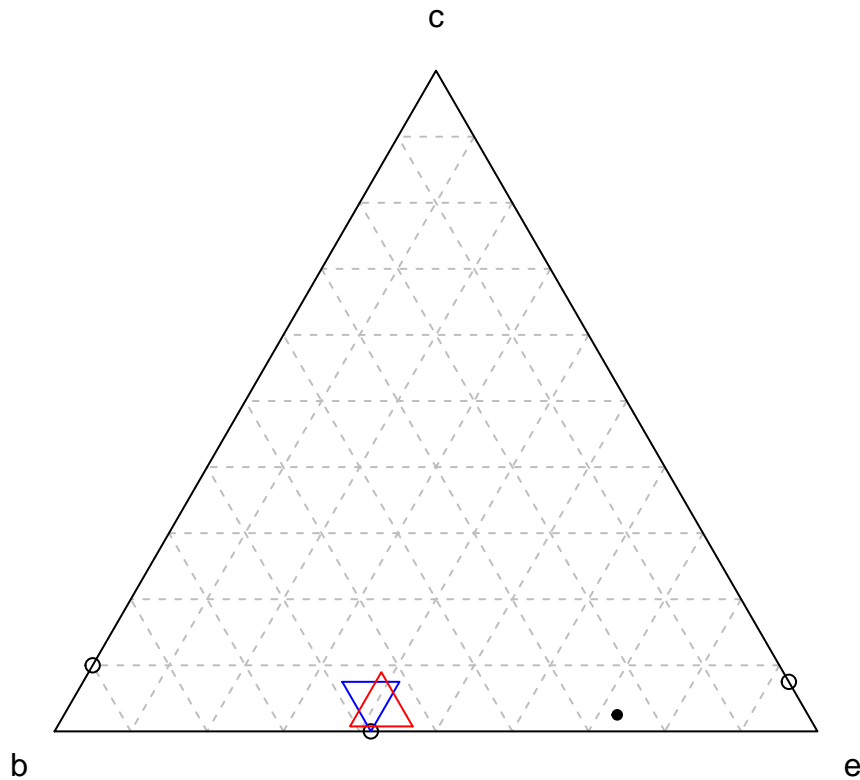
For a universe of size three, regularity and the Block-Marshak conditions are equivalent, so the blue triangle also gives the set of ternary probabilities consistent with the Block-Marschak inequalities. Notice that the ternary choice probability lies outside the two cross sections: the solid dot is not inside either the blue or red triangles. We can conclude that the RCE `Pbce` does not satisfy regularity, the multiplicative inequality, or the Block-Marschak inequalities.

Checking whether a given Random Choice Structure satisfies a given axiom

We just saw visually that for the RCS `P_bce`, the ternary choice probability vector was not consistent with the binary probabilities, under regularity. This can be checked directly using the function `regularity`, which works for RCSs with up to 6 choice objects:

```
print(regularity(P_bce))
  [1] FALSE
```

However, the RCS generated by a Luce model, with weights 2, 3 and 5 for objects 'x', 'y' and 'z' is consistent with regularity:

```
P_xyz <- P_Luce(c(2, 3, 5))
  Warning in v * member_table[i, ]: longer object length is not a multiple of
  shorter object length

  Warning in v * member_table[i, ]: longer object length is not a multiple of
  shorter object length

  Warning in v * member_table[i, ]: longer object length is not a multiple of
  shorter object length

  Warning in v * member_table[i, ]: longer object length is not a multiple of
  shorter object length
```

```
  Warning in v * member_table[i, ]: longer object length is not a multiple of
  shorter object length

  Warning in v * member_table[i, ]: longer object length is not a multiple of
  shorter object length

  Warning in v * member_table[i, ]: longer object length is not a multiple of
  shorter object length
print(P_xyz, na.print='-')
            [,1]   [,2]      [,3]
  [1,] 1.0000000     -         -
  [2,]         - 1.000        -
  [3,] 0.4000000 0.600        -
  [4,]         -     - 1.0000000
  [5,] 0.2857143     - 0.7142857
  [6,]         - 0.375 0.6250000
  [7,] 0.2000000 0.300 0.5000000
print(regularity(P_xyz))
  [1] TRUE
```

Context effects

Three context effects pertaining to stochastic discrete choice have attracted a lot of attention in Psychology, Economics, Marketing and other fields: the similarity effect, the compromise effect and the asymmetric dominance effect. See McCausland, Marley, Davis-Stober and Turner (no paper yet) for details.

Displaying context effects

Just as we can display cross-sections of a RCS consistent with regularity and the multiplicative inequality, we can display cross-sections of regions defined by various context effects.

Suppose we have similar objects x and y and an object z that is dissimilar to both x and y. The function `similarity_X3` constructs cross sections of six regions associated with binary-ternary similarity effects. The function takes as arguments the binary choice probabilities $p(x, z)$ and $p(y, z)$. For each of six types of similarity effect, it returns a region (in the form of a polygon) defining the set of ternary chocie probabilities consistent with $p(x, z)$, $p(y, z)$ and the type of similarity effect in question.

The following code computes the six regions for $p(x, z) = 0.6$ and $p(y, z) = 0.4$. It then displays the regions `So`, where there is no similarity effect; `Sx`, where there is a one-sided similarity effect (with x as target, y as decoy and z as competitor); `Sy`, where there is the other one-sided similarity effect (with x as target, y as decoy and z as competitor); and `Sxy` where there are both similarity effects. Note that `Sx` and `Sy` are both exclusive; this is what we mean by one-sided, as opposed to two-sided effects. The regions `Sxyz` and `Syxz` are non-exclusive: thus `Sxyz` is the region where the similarity effect with x as target, y as decoy and z as competitor occurs, regardless of whether or not the effect with x as decoy, y as target and z as competitor occurs.

```
S <- similarity_X3(pxz = 0.6, pyz = 0.4)
triplot(label=c('x', 'y', 'z')) # Set up ternary plot, with labels and grid
polygon(tritrafo(S$So), col=grey(0.95)); text(tritrafo(colMeans(S$So)), 'So')
polygon(tritrafo(S$Sx), col=grey(0.9)); text(tritrafo(colMeans(S$Sx)), 'Sx')
polygon(tritrafo(S$Sy), col=grey(0.9)); text(tritrafo(colMeans(S$Sy)), 'Sy')
polygon(tritrafo(S$Sxy), col=grey(0.8)); text(tritrafo(colMeans(S$Sxy)), 'Sxy')
```
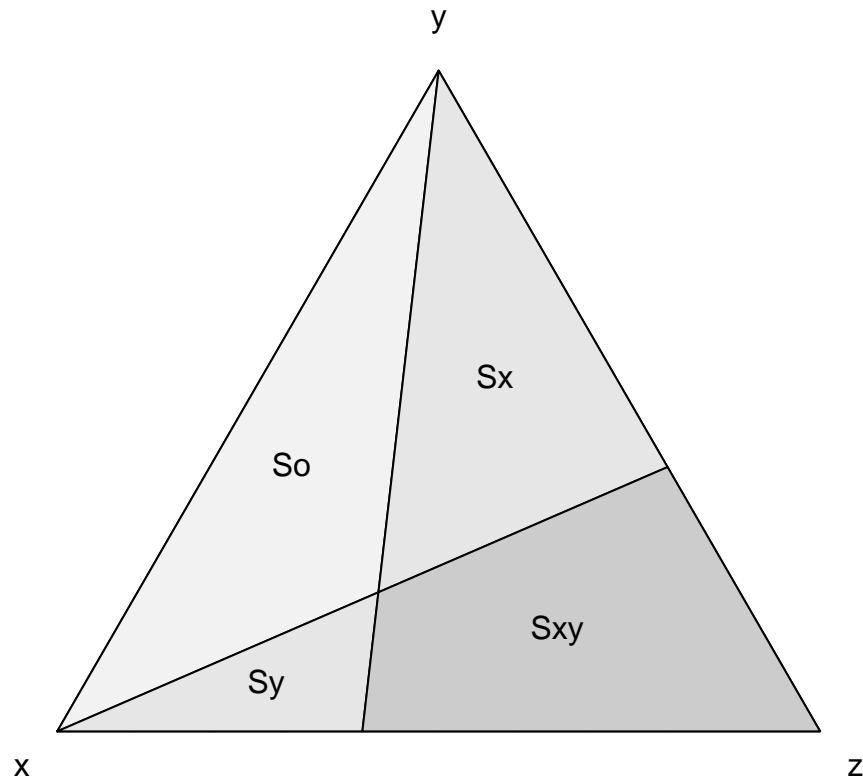
If the objects x, y and z do not correpond to the first, second and third baycentric coordinates that we wish to plot, we can always permute the regions returned by `similarity_X3` accordingly. Suppose we want to plot in a barycentric coordinate system where the three coordinates correspond to the ternary choices probabilities of a, b and c, but here b is the dissimilar object and we want to show the region where there is an effect with a as target, c as decoy and b as competitor.

```r
triplot(label=c('a', 'b', 'c')) # Set up ternary plot, with labels and grid
Sacb <- S$Sxyz[, c(1, 3, 2)]
polygon(tritrafo(Sacb), col=grey(0.95))
text(tritrafo(colMeans(Sacb)), 'Sacb')
```
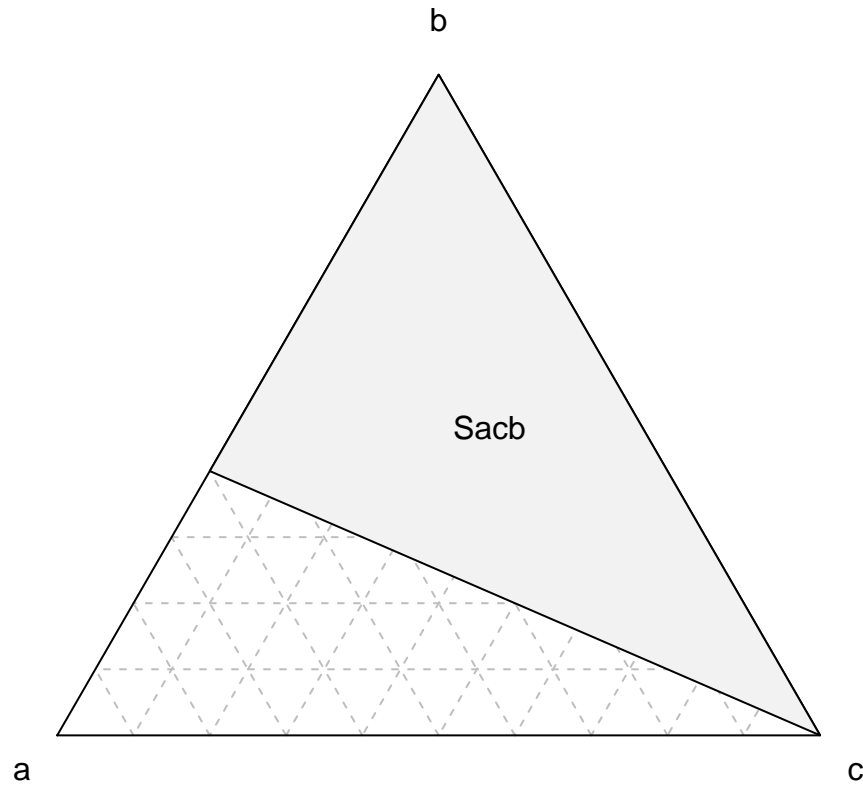
We now show some compromise effect regions. Suppose that x and z are extreme objects and that y is a between (or compromise) object. The following code computes six cross-sections of compromise regions and displays the regions `Co`, where there is no compromise effect; `Cx`, where there is a one-sided compromise effect with y as target, x as competitor and z as decoy; `Cz`, where there is a one-sided compromise effect with y as target; z as competitor and x as decoy; `Cxz`, where there is a two-sided compromise effect.

```r
C <- compromise_X3(pyx = 0.6, pyz = 0.4)
triplot(label=c('x', 'y', 'z')) # Set up ternary plot
polygon(tritrafo(C$Co), col=grey(0.95)); text(tritrafo(colMeans(C$Co)), 'Co')
polygon(tritrafo(C$Cx), col=grey(0.9)); text(tritrafo(colMeans(C$Cx)), 'Cx')
polygon(tritrafo(C$Cz), col=grey(0.9)); text(tritrafo(colMeans(C$Cz)), 'Cz')
polygon(tritrafo(C$Cxz), col=grey(0.8)); text(tritrafo(colMeans(C$Cxz)), 'Cxz')
```
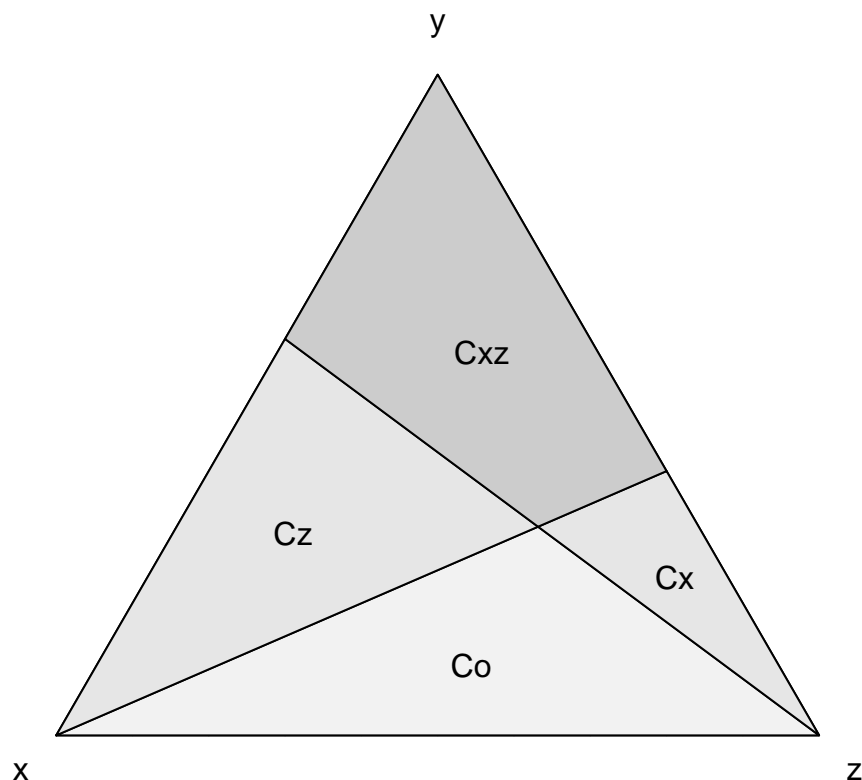
Checking for context effects

We can also check specific RCDs, arising from direct specification, simulation, or observed choice proportions, to see if they exhibit context effects. Below, we simulate 500 variates from the uniform distribution on the 2-simplex. For each draw, we construct a RCD by setting the ternary choice probability to the random draw and supplying the (constant) binary choice probabilities used in the examples illustrating `similarity_X3` and `compromise_X3`—the probabilities in the two examples are consistent with each other. Then we create two graphics. In the first, we plot only those ternary probabilities that, together with the fixed binary probabilities, satisfy the two-sided similarity effect. In the second, we plot the points that satisfy the two-sided compromise effect. We note that the retained points fall in the regions `Sxy` and `Cxz` from the `similarity_X3` and `compromise_X3` examples, respectively.

```
n <- 500
filt1 <- vector('logical', n); filt2 <- vector('logical', n)
p <- rDirichlet(n, c(1, 1, 1)) # Uniform distribution on 2-simplex
for (i in 1:n) {
  P <- create_P3(0.4, 0.4, 0.6, p[i,1], p[i,2])
  filt1[i] <- similarity(P, target=1, decoy=2, competitor=3, two_sided=TRUE)
  filt2[i] <- compromise(P, target=2, decoy=3, competitor=1, two_sided=TRUE)
}
triplot(label=c('x', 'y', 'z'))
points(tritrafo(p[filt1, ]), pch=20)
```
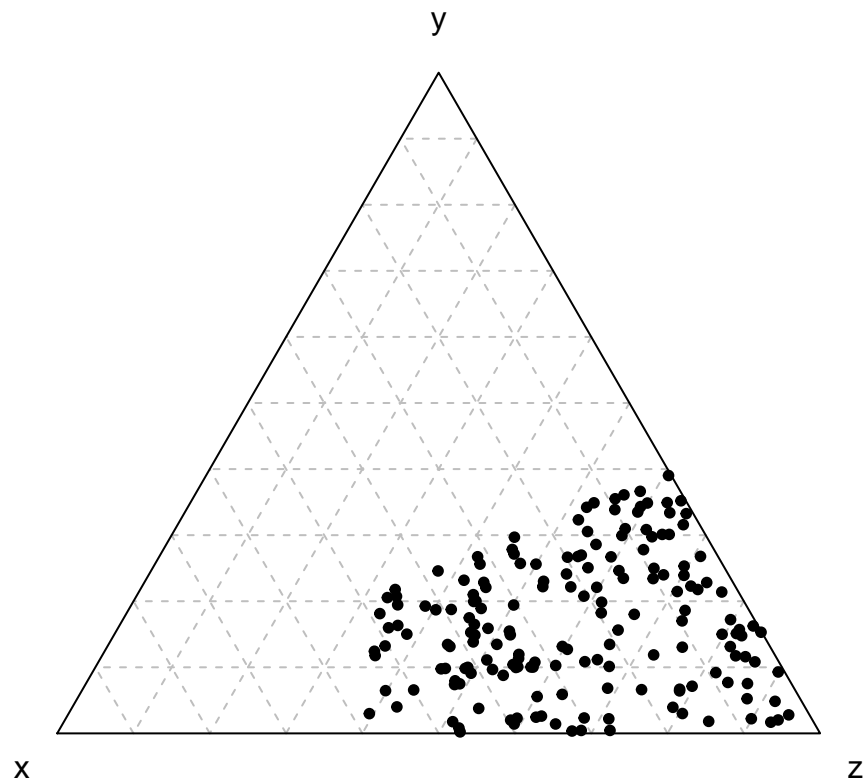
```
triplot(label=c('x', 'y', 'z'))
points(tritrafo(p[filt2, ]), pch=20)
```
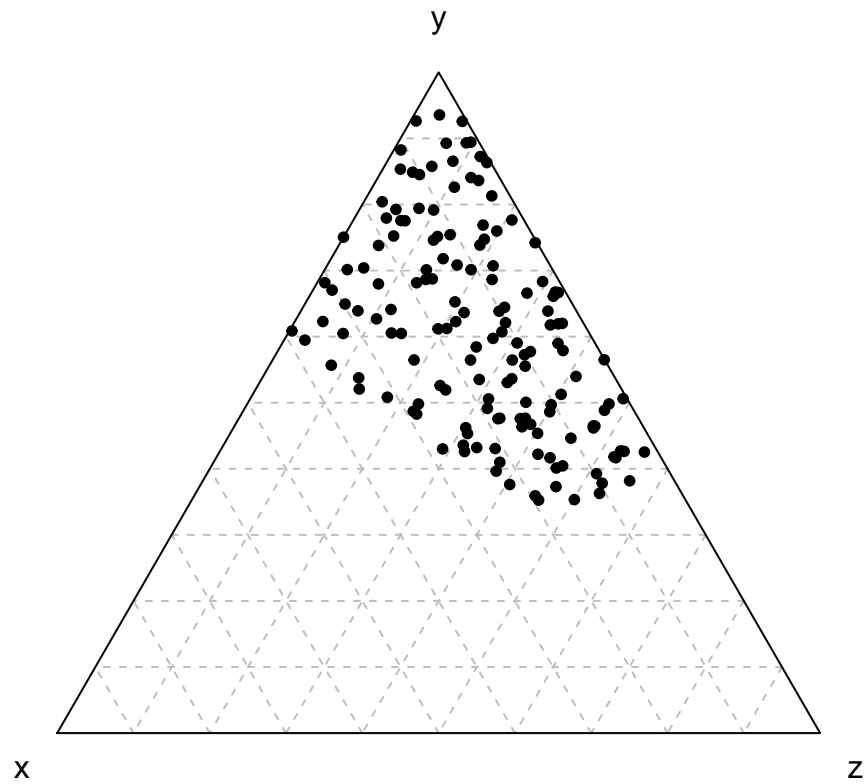
Inference

Suppose we have a count matrix N for a discrete choice experiment and an RCS matrix P with the same dimensions as N. Then the likelihood function for data N, evaluated at P, is a product of multinomial likelihoods, each likelihood factor corresponding to a particular choice set. Each choice set corresponds in turn to a particular row of both N and P.

We can complete a Bayesian model by providing a conjugate prior, where the rows of P are independent Dirichlet. This leads to closed form solutions to the posterior distribution of P and the marginal likelihood, which is the integral of the likelihood with respect to the prior distribution of P. The marginal likelihood is also the marginal probability of seeing the data N; it is, in a sense, an out-of-sample prediction record of the model for the data N.

In general, we can specify the conjugate prior in the form of a matrix of Dirichlet parameters, with the same dimensions as P and N. The simplest way of doing this is to set, for each choice set A, all Dirichlet parameters to $\alpha/|A|$. See McCausland and Marley (2013) for some justification for this choice. The function `DCS_scalar_alpha_prior` does just this, setting `prior_Alpha`. Here, $\alpha = 2$, which implies that all binary choice probabilities are uniform on $[0, 1]$.

The posterior distribution takes the same form as the prior: rows are *a posteriori* independent and Dirichlet. Here the matrix of Dirichlet parameters is `post_Alpha`, the sum of `prior_Alpha` and `N_bce`; see the Wikipedia page on the Dirichlet distribution for details.

High posterior density curves

The code below constructs and then displays 0.90 High Posterior Density (HPD) regions for four choice probability vectors. Regions for the three binary choice probabilities are the thick-line intervals on the three sides of the triangle. The region for the ternary choice probability vector is the green shape in the interior of the triangle.

```
library(klaR)
prior_Alpha <- RCS_scalar_alpha_prior(2.0, ncol(N_bce))
post_Alpha <- prior_Alpha + N_bce
HD3 <- Dir2_3_HD_region(post_Alpha, 0.9, c(1,2,3))
triplot(label=c('b', 'c', 'e'))              # Set up ternary plot
lines(tritrafo(HD3$HD12), lwd=4)             # Plot three binaries
lines(tritrafo(HD3$HD23), lwd=4)
lines(tritrafo(HD3$HD13), lwd=4)
polygon(tritrafo(HD3$HD123), border='lightgreen') # Plot ternary
plot_P3(P_bce)                               # Plot proportions from data
```
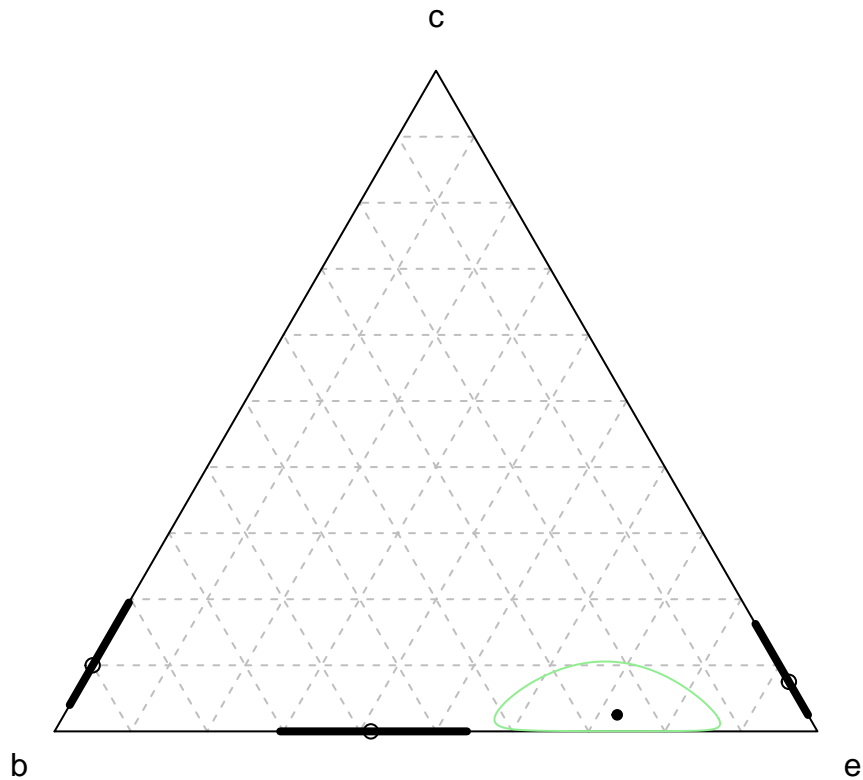
Marginal likelihoods for some simple models

We can also compute the log marginal likelihood for various simple models, as follows.

```
N = MMS_2019_counts['Colours',,]
n_objects = ncol(N)

# Zero parameter models
print(log_L_DCE_multinomial(P_uniform(n_objects), N, categorical=TRUE))
  [1] -1060.825
print(log_L_DCE_Dir_mult(RCS_uniform_prior(n_objects), N, categorical=TRUE))
  [1] -947.5373

# One parameter models
print(log_L_DCE_Dir_mult(RCS_scalar_alpha_prior(2.0, n_objects), N,
                         categorical=TRUE))
  [1] -955.2024

# n-parameter models
print(log_L_DCE_Dir_mult(RCS_vector_alpha_prior(4.0, c(1.0, 2.0, 1.0, 2.0, 1.0)),
                         N, categorical=TRUE))
  [1] -939.3222
```

These marginal likelihoods serve as useful benchmarks.

References

Falmagne, J. C. 1978. "A Representation Theorem for Finite Random Scale Systems." *Journal of Mathematical Psychology* 18: 52–72.

McCausland, William J., and A. A. J. Marley. 2013. "Prior Distributions for Random Choice Structures." *Journal of Mathematical Psychology* 57: 78–93.

Regenwetter, M., J. Dana, and C. P. Davis-Stober. 2011. "Transitivity of Preferences." *Psychological Review* 118: 42–56.

Sattath, S., and A. Tversky. 1976. "Unite and Conquer: A Multiplicative Inequality for Choice Probabilities." *Econometrica* 44: 79–89.

Tversky, A. 1972. "Choice by Elimination." *Journal of Mathematical Psychology* 9: 341–67.

Tversky, Amos. 1972. "Elimination by Aspects: A Theory of Choice." *Psychological Review* 79 (4): 281–99.