

# Package ‘hint’

February 20, 2015

**Type** Package

**Title** Tools for hypothesis testing based on Hypergeometric Intersection distributions

**Version** 0.1-1

**Date** 2013-05-16

**Author** Alex T. Kalinka

**Maintainer** Alex T. Kalinka <alex.t.kalinka@gmail.com>

## Description

Hypergeometric Intersection distributions are a broad group of distributions that describe the probability of picking intersections when drawing independently from two (or more) urns containing variable numbers of balls belonging to the same  $n$  categories.

**License** GPL ( $\geq 2$ )

**LazyLoad** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2013-07-24 11:31:45

## R topics documented:

hint-package . . . . .	2
add.distr . . . . .	3
Binomialintersection . . . . .	4
hint.dist.test . . . . .	5
hint.test . . . . .	7
Hyperdistinct . . . . .	8
Hyperintersection . . . . .	10
overlay.sim . . . . .	12
plot.hint.test . . . . .	13
plotDistr . . . . .	14
print.hint.test . . . . .	15
sim.hydist . . . . .	16
sim.hypint . . . . .	17

---

hint-package	<i>The hint package</i>
--------------	-------------------------

---

## Description

Tools for hypothesis testing based on Hypergeometric Intersection distributions.

## Details

Hypergeometric Intersection distributions are a broad group of distributions that describe the probability of picking intersections when drawing independently from two (or more) urns containing variable numbers of balls belonging to the same  $n$  categories.

## Author(s)

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

## References

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](https://arxiv.org/abs/1305.0717)

## See Also

[Hyperintersection](#), [hint.test](#), [Binomialintersection](#), [Hyperdistinct](#).

## Examples

```
## Generate a hypergeometric intersection distribution and plot it.
dd <- dhint(110, c(65, 71))
plotDistr(dd)

## Allow duplicates in the second urn and add the distributions to the existing plot.
dd <- dhint(110, c(65, 71), 82)
dd2 <- dhint(59, c(44, 28), 47)
add.distr(dd, dd2, cols = c("blue", "darkgreen"))

## Test whether the intersection of categories in data sets is significant and plot the result:
dd <- data.frame(letters[1:20], rep(1,20), rep(1,20))
tt <- hint.test(dd, letters[1:9], letters[4:15], alternative = "greater")
plot(tt)

## Allow duplicates in the second urn.
dd <- data.frame(letters[1:20], rep(1,20), c(rep(1,4),rep(2,16)))
tt <- hint.test(dd, letters[1:9], letters[9:14], alternative = "less")
plot(tt)
```

---

`add.distr`*Add Distributions to a Plot*

---

## Description

This function will add one or more distributions or hypothesis tests to an existing plot.

## Usage

```
add.distr(..., cols = "blue", test.cols = "red")
```

## Arguments

<code>...</code>	One or more distributions or objects of class <code>hint.test</code> .
<code>cols</code>	A character string vector naming the colours of the distributions. If <code>length(cols)</code> is less than the number of distributions, the colours will be recycled. Defaults to "blue".
<code>test.cols</code>	A character string vector naming the colours to use for the regions in which the cumulative probability of the hypothesis test was derived (if it exists). If <code>length(test.cols)</code> is less than the number of distributions, the colours will be recycled. Defaults to "red".

## Value

Plots to the current device.

## Author(s)

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

## See Also

[plotDistr](#), [plot.hint.test](#).

## Examples

```
dd <- dhint(25, c(12, 9))
plotDistr(dd)
dd1 <- dhint(12, c(4, 7))
dd2 <- dhint(8, c(4, 2))
add.distr(dd1, dd2, cols = c("blue", "darkgreen"))
```

**Description**

Density, distribution function, quantile function and random generation for the binomial intersection distribution.

**Usage**

```
dbint(n, A, range = NULL, log = FALSE)
pbint(n, A, vals, upper.tail = TRUE, log.p = FALSE)
qbint(p, n, A, upper.tail = TRUE, log.p = FALSE)
rbint(num = 5, n, A)
```

**Arguments**

n	An integer specifying the number of categories in the urns.
A	A vector of integers specifying the numbers of balls drawn from each urn. The length of the vector equals the number of urns.
p	A probability between 0 and 1.
num	An integer specifying the number of random numbers to generate. Defaults to 5.
range, vals	A vector of integers specifying the intersection sizes for which probabilities (dhint) or cumulative probabilities (phint) should be computed (can be a single number). If range is NULL (default) then probabilities will be returned over the entire range of possible values.
log, log.p	Logical. If TRUE, probabilities p are given as log(p). Defaults to FALSE.
upper.tail	Logical. If TRUE, probabilities are P(X >= v), else P(X <= v). Defaults to TRUE.

**Details**

The binomial intersection distribution is given by

$$P(X = v|N) = \binom{b}{v} \left( \prod_{i=1}^{N-1} p_i \right)^v \left( 1 - \prod_{i=1}^{N-1} p_i \right)^{b-v}$$

where b gives the sample size which is smallest. This is an approximation for the hypergeometric intersection distribution when n is large and b is small relative to the samples taken from the N - 1 other urns.

**Value**

dbint, pbint, and qbint return a data frame with two columns: v, the intersection size, and p, the associated p-values. rbint returns an integer vector of random samples based on the binomial intersection distribution.

**Author(s)**

Alex T. Kalinka <alex.t.kalinka@gmail.com>

**References**

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](#)

**See Also**

[Hyperintersection](#), [hint.test](#), [hint.dist.test](#), [plotDistr](#), [Hyperdistinct](#).

**Examples**

```
## Generate the distribution of intersections sizes:
dd <- dbint(20, c(10, 12, 11, 14))
## Restrict the range of intersections.
dd <- dbint(20, c(10, 12), range = 0:5)
## Generate cumulative probabilities.
pp <- pbint(29, c(15, 8), vals = 5)
pp <- pbint(29, c(15, 8), vals = 2, upper.tail = FALSE)
## Extract quantiles:
qq <- qbint(0.15, 23, c(12, 10))
## Generate random samples from Binomial intersection distributions.
rr <- rbint(num = 10, 18, c(9, 14))
```

---

hint.dist.test

*To Test for Significant Distances between pairs of Intersection Sizes*

---

**Description**

Tests whether the absolute distance between two intersection sizes would be expected by chance, i.e. whether they fall into opposite tails of their respective Hypergeometric Intersection distributions.

**Usage**

```
hint.dist.test(d, n1, A1, n2, A2, q1 = 0, q2 = 0, alternative = "greater")
```

**Arguments**

d	A positive integer specifying the observed distance to be tested.
n1	An integer specifying the number of categories in the urns for the first distribution.
A1	An integer vector specifying the number of balls drawn from urns for the first distribution.
n2	An integer specifying the number of categories in the urns for the second distribution.

A2	An integer vector specifying the number of balls drawn from the urns for the second distribution.
q1	An integer specifying the number of categories with duplicates in the second urn of the first distribution. If 0 then the symmetric, singleton case is computed, otherwise the asymmetric, duplicates case is computed (see <a href="#">Hyperintersection</a> ).
q2	An integer specifying the number of categories with duplicates in the second urn of the second distribution. If 0 then the symmetric, singleton case is computed, otherwise the asymmetric, duplicates case is computed (see <a href="#">Hyperintersection</a> ).
alternative	A character string specifying the hypothesis to be tested. Can be one of "greater", "less", or "two.sided".

### Details

The distribution of absolute distances between two hypergeometric intersection sizes is given by

$$P(X = d) = \sum_{\{v_1, v_2\}_i \in D_d}^{|D_d|} P(v_{1_i} | n_1, a_1, b_1, \dots) \cdot P(v_{2_i} | n_2, a_2, b_2, \dots)$$

where  $D_d$  is the set of pairs of intersection sizes,  $\{v_1, v_2\}$ , with absolute differences of size  $d$ .

### Value

An object of class `hint.test`, which is a list containing the following components:

parameters	An integer vector giving the parameter values.
p.value	A numerical value giving the p-value associated with the test.
alternative	A character string naming the hypothesis that was tested.

### Author(s)

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

### References

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](#)

### See Also

[phint](#), [hint.test](#), [plotDistr](#), [plot.hint.test](#).

### Examples

```
hint.dist.test(4, 20, c(11, 9), 24, c(14, 17), alternative = "greater")
```

---

`hint.test`*A Hypergeometric Intersection Test*

---

**Description**

Apply the hypergeometric intersection test to categorical data to test for enrichment or depletion of intersections between two samples.

**Usage**

```
hint.test(cats, draw1, draw2, alternative = "greater")
```

**Arguments**

<code>cats</code>	A data frame or matrix with 3 columns; the first gives the category identifier, and the second and third give the number of balls belonging to this category in the first and second urns respectively.
<code>draw1</code>	A vector of objects corresponding to the categories given in <code>cats</code> drawn from the first urn.
<code>draw2</code>	A vector of objects corresponding to the categories given in <code>cats</code> drawn from the second urn.
<code>alternative</code>	A character string specifying the hypothesis to be tested. Can be one of "greater", "less", or "two.sided".

**Details**

The hypergeometric intersection distributions describe the distribution of intersection sizes when sampling without replacement from two separate urns in which reside balls belonging to the same  $n$  object categories (see [Hyperintersection](#)).

**Value**

An object of class `hint.test`, which is a list containing the following components:

<code>parameters</code>	An integer vector giving the parameter values.
<code>p.value</code>	A numerical value giving the p-value associated with the test.
<code>alternative</code>	A character string naming the hypothesis that was tested.

**Author(s)**

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

**References**

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](#)

**See Also**

[phint](#), [hint.dist.test](#), [plotDistr](#), [plot.hint.test](#).

**Examples**

```
## Build data input:
dd <- data.frame(letters[1:20], rep(1,20), rep(1,20))
hint.test(dd, letters[1:9], letters[4:14], alternative = "greater")

dd <- data.frame(letters[1:20], rep(1,20), c(rep(1,4),rep(2,16)))
hint.test(dd, letters[1:9], letters[4:14], alternative = "less")
```

---

Hyperdistinct

*Drawing Distinct Categories from a Single Urn*

---

**Description**

Density, distribution function, quantile function and random generation for the distribution of distinct categories drawn from a single urn in which there are duplicates in  $q$  of the categories.

**Usage**

```
dhydist(n, a, q, range = NULL, log = FALSE)
phydist(n, a, q, vals, upper.tail = TRUE, log.p = FALSE)
qhydist(p, n, a, q, upper.tail = TRUE, log.p = FALSE)
rhydist(num = 5, n, a, q)
```

**Arguments**

<code>n</code>	An integer specifying the number of categories in the urn.
<code>a</code>	An integer specifying the number of balls drawn from the urn.
<code>q</code>	An integer specifying the number of categories in the urn which have duplicate members.
<code>p</code>	A probability between 0 and 1.
<code>num</code>	An integer specifying the number of random numbers to generate. Defaults to 5.
<code>range, vals</code>	A vector of integers specifying the intersection sizes for which probabilities (dhydist) or cumulative probabilities (phydist) should be computed (can be a single number). If range is NULL (default) then probabilities will be returned over the entire range of possible values.
<code>log, log.p</code>	Logical. If TRUE, probabilities $p$ are given as $\log(p)$ . Defaults to FALSE.
<code>upper.tail</code>	Logical. If TRUE, probabilities are $P(X \geq c)$ , else $P(X \leq c)$ . Defaults to TRUE.

**Details**

The distribution of the number of distinct categories drawn when sampling without replacement from a single urn containing duplicates in  $q$  of its  $n$  categories is given by

$$P(X = c) = \binom{q}{a-c} \sum_{j=0}^q \binom{q-a+c}{j} \binom{n-a+c-j}{2c-a-j} / \binom{n+q}{a}$$

When all of the  $n$  categories contain duplicates, this can be expressed in a closed form:

$$P(X = c) = \binom{n}{c} \binom{c}{a-c} 2^{2c-a} / \binom{2n}{a}$$

**Value**

`dhydist`, `phydist`, and `qhydist` return a data frame with two columns: `c`, the number of distinct categories drawn, and `p`, the associated p-values. `rhydist` returns an integer vector of random samples based on the distribution of distinct categories when sampling from a single urn containing  $q$  duplicates in  $n$  categories.

**Author(s)**

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

**References**

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](https://arxiv.org/abs/1305.0717)

**See Also**

[Hyperintersection](#), [plotDistr](#).

**Examples**

```
## Generate the distribution of distinct categories drawn from a single urn.
dd <- dhydist(20, 10, 12)
## Restrict the range of intersections.
dd <- dhydist(20, 10, 12, range = 5:10)
## Generate cumulative probabilities.
pp <- phydist(29, 15, 8, vals = 5)
pp <- phydist(29, 15, 8, vals = 2, upper.tail = FALSE)
## Extract quantiles:
qq <- qhydist(0.15, 23, 12, 10)
## Generate random samples based on this distribution.
rr <- rhydist(num = 10, 18, 9, 12)
```

**Description**

Density, distribution function, quantile function and random generation for the hypergeometric intersection distribution.

**Usage**

```
dhint(n, A, q = 0, range = NULL, approx = FALSE, log = FALSE, verbose = TRUE)
phint(n, A, q = 0, vals, upper.tail = TRUE, log.p = FALSE)
qhint(p, n, A, q = 0, upper.tail = TRUE, log.p = FALSE)
rhint(num = 5, n, A, q = 0)
```

**Arguments**

n	An integer specifying the number of categories in the urns.
A	A vector of integers specifying the numbers of balls drawn from each urn. The length of the vector equals the number of urns.
q	An integer specifying the number of categories in the second urn which have duplicate members. If q is 0 (default) then the symmetrical, singleton case is computed, otherwise the asymmetrical, duplicates case is computed (see Details).
p	A probability between 0 and 1.
num	An integer specifying the number of random numbers to generate. Defaults to 5.
range, vals	A vector of integers specifying the intersection sizes for which probabilities (dhint) or cumulative probabilities (phint) should be computed (can be a single number). If range is NULL (default) then probabilities will be returned over the entire range of possible values.
approx	Logical. If TRUE, a binomial approximation will be used to generate the distribution.
log, log.p	Logical. If TRUE, probabilities p are given as log(p). Defaults to FALSE.
verbose	Logical. If TRUE, progress of calculation in the asymmetrical, duplicates case is printed to the screen.
upper.tail	Logical. If TRUE, probabilities are $P(X \geq v)$ , else $P(X \leq v)$ . Defaults to TRUE.

**Details**

The hypergeometric intersection distributions describe the distribution of intersection sizes when sampling without replacement from two separate urns in which reside balls belonging to the same

$n$  object categories. In the simplest case when there is exactly one ball in each category in each urn (symmetrical, singleton case), then the distribution is hypergeometric:

$$P(X = v) = \frac{\binom{a}{v} \binom{n-a}{b-v}}{\binom{n}{b}}$$

When there are three urns, the distribution is given by

$$P(X = v) = \frac{\binom{a}{v} \sum_i \binom{a-v}{i} \binom{n-a}{b-v-i} \binom{n-v-i}{c-v}}{\binom{n}{b} \binom{n}{c}}$$

If, however, we allow duplicates in  $q \leq n$  of the categories in the second urn, then the distribution of intersection sizes is described by the following variant of the hypergeometric:

$$P(X = v) = \sum_{m=0}^{\alpha} \sum_{l=0}^{\beta} \sum_{j=0}^l \binom{n-q}{v-l} \binom{q}{l} \binom{q-l}{m} \binom{n-v-q+l}{a-v-m} \binom{l}{j} \binom{n+q-a-m-j}{b-v} / \binom{n}{a} \binom{n+q}{b}$$

### Value

`dhint`, `phint`, and `qhint` return a data frame with two columns:  $v$ , the intersection size, and  $p$ , the associated  $p$ -values. `rhint` returns an integer vector of random samples based on the hypergeometric intersection distribution.

### Author(s)

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

### References

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](https://arxiv.org/abs/1305.0717)

### See Also

[Binomialintersection](#), [hint.test](#), [hint.dist.test](#), [plotDistr](#), [Hyperdistinct](#).

### Examples

```
## Generate the distribution of intersections sizes without duplicates:
dd <- dhint(20, c(10, 12))
## Restrict the range of intersections.
dd <- dhint(20, c(10, 12), range = 0:5)
## Allow duplicates in q of the categories in the second urn:
dd <- dhint(35, c(15, 11), 22, verbose = FALSE)
## Generate cumulative probabilities.
pp <- phint(29, c(15, 8), vals = 5)
pp <- phint(29, c(15, 8), vals = 2, upper.tail = FALSE)
pp <- phint(29, c(15, 8), 23, vals = 2)
## Extract quantiles:
qq <- qhint(0.15, 23, c(12, 10))
qq <- qhint(0.15, 23, c(12, 10), 18)
```

```
## Generate random samples from Hypergeometric intersection distributions.  
rr <- rhint(num = 10, 18, c(9, 14))  
rr <- rhint(num = 10, 22, c(11, 17), 12)
```

---

overlay.sim

*Overlay Simulation Results on Distribution*

---

### Description

This function adds simulated data to a distribution plot.

### Usage

```
overlay.sim(sim, breaks, col = "red", pch = 1, lwd = 1)
```

### Arguments

sim	A vector of integers (simulated data). See <a href="#">sim.hypint</a> and <a href="#">sim.hydist</a> .
breaks	A vector of integers specifying the x-axis range for which simulated data values should be plotted.
col	A character string naming the colour of the data points. Defaults to "red".
pch	A number specifying the point type. Defaults to 1.
lwd	A number specifying the line width. Defaults to 1.

### Value

Simulated data will be added to an existing plot.

### Author(s)

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

### References

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](#)

### See Also

[sim.hypint](#), [sim.hydist](#).

**Examples**

```
## Generate exact distribution and plot:
dd <- dhint(108, c(54,76,89))
plot(dd[,1], dd[,2], col="blue")
## Simulate drawing from 3 urns (sims should be larger).
nn <- sim.hydist(108, c(54,76,89), sims = 10)
## Add data to existing distribution plot:
overlay.sim(nn, breaks = 0:54)
```

---

plot.hint.test            *Plot hint.test Objects.*

---

**Description**

This function visualises the results of a Hypergeometric Intersection test.

**Usage**

```
## S3 method for class 'hint.test'
plot(x, ...)
```

**Arguments**

x                    An object of class `hint.test`.  
...                  Additional arguments to be passed to `plot`.

**Details**

Plots the relevant Hypergeometric Intersection distribution as a segment plot, and highlights the region where the observed statistic falls, i.e. the region from which the probability is computed (two.sided tests are visualised in one tail, the one with the smallest density). This can be especially useful for pedagogical purposes.

**Value**

Plots to the current device.

**Author(s)**

Alex T. Kalinka <alex.t.kalinka@gmail.com>

**See Also**

[plotDistr](#), [hint.test](#), [hint.dist.test](#) [dhint](#).

**Examples**

```
## Build hint.test object:
dd <- data.frame(letters[1:20], rep(1,10), rep(1,10))
tt <- hint.test(dd, letters[1:9], letters[7:15], alternative = "greater")

## Plot the distribution with the test region highlighted in red.
plot(tt, col = "blue")
```

---

plotDistr

*Plot a Distribution*


---

**Description**

Plot a distribution or visualise the result of a hypothesis test.

**Usage**

```
plotDistr(distr, col = "black", test.col = "red", xlim = NULL, ylim = NULL,
          xlab = "Intersection size (v)", ylab = "Probability", add = FALSE, ...)
```

**Arguments**

distr	A data frame or matrix in which the first column gives random variable values, and the second gives probabilities. Can also be a vector (in which case random variables of $0:\text{length}(\text{distr})$ will be automatically assigned, or an object of class <code>hint.test</code> .
col	A character string naming the colour to use for the distribution. Defaults to "black".
test.col	A character string naming the colour to use for the region in which the cumulative probability of the hypothesis test was derived (if it exists). Defaults to "red".
xlim	A vector of two numbers giving the range for the x-axis. If NULL (default), then this is determined by the maximum and minimum values in <code>distr</code> .
ylim	A vector of two numbers giving the range for the y-axis. If NULL (default), then this is determined by the maximum and minimum values in <code>distr</code> .
xlab	A character string giving a label for the x-axis. Defaults to "Intersection size (v)".
ylab	A character string giving a label for the y-axis. Defaults to "Probability".
add	Logical. Whether the plot will be added to an existing plot or not. Defaults to FALSE.
...	Additional arguments to be passed to <code>plot</code> .

**Details**

Visualising the results of a hypothesis test may often be of interest, but can be especially useful for pedagogical purposes.

**Value**

Plots to the current device.

**Author(s)**

Alex T. Kalinka <alex.t.kalinka@gmail.com>

**See Also**

[plot.hint.test](#), [hint.test](#), [hint.dist.test](#).

**Examples**

```
dd <- dhint(25, c(12, 9))
plotDistr(dd)
```

---

`print.hint.test`      *Print a Summary of a hint.test Object*

---

**Description**

Prints a summary of a `hint.test` object to the screen.

**Usage**

```
## S3 method for class 'hint.test'
print(x, ...)
```

**Arguments**

`x`                    An object of class `hint.test`.  
`...`                  Additional arguments to be passed to `print`.

**Value**

Prints summary output to the screen.

**Author(s)**

Alex T. Kalinka <alex.t.kalinka@gmail.com>

**See Also**

[hint.test](#), [hint.dist.test](#).

## Examples

```
## Build test object.
dd <- data.frame(letters[1:20], rep(1,10), rep(1,10))
tt <- hint.test(dd, letters[1:9], letters[7:15], alternative = "greater")

## Print summary:
print(tt)
```

---

sim.hydist

*Simulating the Drawing of Distinct Categories*

---

## Description

This function simulates the drawing of balls from a single urn containing  $q$  duplicates in  $n$  categories to generate distributions of the number of distinct categories drawn.

## Usage

```
sim.hydist(n, a, sims = 10000, Na = rep(2, n))
```

## Arguments

n	An integer specifying the number of categories in the urn.
a	An integer specifying the number of balls drawn from the urn.
sims	An integer specifying the number of replicate draws for the simulation. Defaults to 10,000.
Na	A vector of integers specifying the number of balls in each category of the urn.

## Details

The simulation allows any number of balls in each of the  $n$  categories via the `Na` argument.

## Value

An integer vector of length `sims` giving the numbers of distinct categories for all replicates of the simulation.

## Author(s)

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

## References

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](https://arxiv.org/abs/1305.0717)

**See Also**

[Hyperdistinct](#).

**Examples**

```
## All n categories contain duplicates:
nn <- sim.hydist(108, 74, sims = 10)
## Duplicates in 78 of the categories:
nn <- sim.hydist(108, 74, sims = 10, Na = c(rep(1,(108-78)),rep(2,78)))
```

---

sim.hypint

*Simulating Intersection Sizes*

---

**Description**

This function simulates the drawing of balls from two urns to generate distributions of intersection sizes.

**Usage**

```
sim.hypint(n, A, sims = 10000, Na = NULL)
```

**Arguments**

n	An integer specifying the number of categories in the urns.
A	A vector of integers specifying the samples to be taken from each urn. The length of the vector equals the number of urns.
sims	An integer specifying the number of replicate draws for the simulation. Defaults to 10,000.
Na	A list of integer vectors specifying the number of balls in each category for each urn.

**Details**

The simulation allows any number of balls in each of the n categories via the Na argument.

**Value**

An integer vector of length sims giving intersection sizes for all replicates of the simulation.

**Author(s)**

Alex T. Kalinka <[alex.t.kalinka@gmail.com](mailto:alex.t.kalinka@gmail.com)>

**References**

Kalinka, A.T. (2013). The probability of drawing intersections: extending the hypergeometric distribution. [arXiv.1305.0717](#)

**See Also**

[Hyperintersection](#), [hint.test](#).

**Examples**

```
nn <- sim.hypint(108, c(54, 73), sims = 10)
```

# Index

## \*Topic **distribution**

- Binomialintersection, 4
- hint-package, 2
- hint.dist.test, 5
- hint.test, 7
- Hyperdistinct, 8
- Hyperintersection, 10
- plot.hint.test, 13
- plotDistr, 14

## \*Topic **htest**

- hint-package, 2
- hint.dist.test, 5
- hint.test, 7
- plot.hint.test, 13
- plotDistr, 14

## \*Topic **package**

- hint-package, 2

## \*Topic **univar**

- Binomialintersection, 4
- hint-package, 2
- Hyperdistinct, 8
- Hyperintersection, 10

add.distr, 3

Binomialintersection, 2, 4, 11

dbint (Binomialintersection), 4

dhint, 13

dhint (Hyperintersection), 10

dhydist (Hyperdistinct), 8

hint (hint-package), 2

hint-package, 2

hint.dist.test, 5, 5, 8, 11, 13, 15

hint.test, 2, 5, 6, 7, 11, 13, 15, 18

Hyperdistinct, 2, 5, 8, 11, 17

Hyperintersection, 2, 5–7, 9, 10, 18

overlay.sim, 12

pbint (Binomialintersection), 4

phint, 6, 8

phint (Hyperintersection), 10

phydist (Hyperdistinct), 8

plot.hint.test, 3, 6, 8, 13, 15

plotDistr, 3, 5, 6, 8, 9, 11, 13, 14

print.hint.test, 15

qbint (Binomialintersection), 4

qhint (Hyperintersection), 10

qhydist (Hyperdistinct), 8

rbint (Binomialintersection), 4

rhint (Hyperintersection), 10

rhydist (Hyperdistinct), 8

sim.hydist, 12, 16

sim.hypint, 12, 17