

Introduction to conditioned Latin hypercube sampling with the `clhs` package

Pierre Roudier*

October 14, 2016

1 A simple example

```
> data(diamonds, package = 'ggplot2')
> diamonds <- data.frame(diamonds)
> head(diamonds)
```

	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
5	0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48

```
> nrow(diamonds)
```

```
[1] 53940
```

In this example we sample the `diamonds` data set and pick a subset of 100 individuals using the `cLHS` method. To reduce the length of the optimisation step to 1000 iterations to save computing time. This is controlled through the `iter` option. The progress bar is disabled because it doesn't renders well in the vignette. By default, the index of the selected individuals in the original object are returned.

```
> library(clhs)
> res <- clhs(diamonds, size = 100, progress = FALSE, iter = 1000)
> str(res)
```

```
int [1:100] 40866 51462 18309 46793 11972 38311 39728 16176 35433 15530 ...
```

*Landcare Research – Manaaki Whenua, Private Bag 11052, Manawatu Mail Centre, Palmerston North 4442, New Zealand. roudierp@landcareresearch.co.nz

2 Tweaking the parameters

(work in progress)

3 Cost-constrained implementation

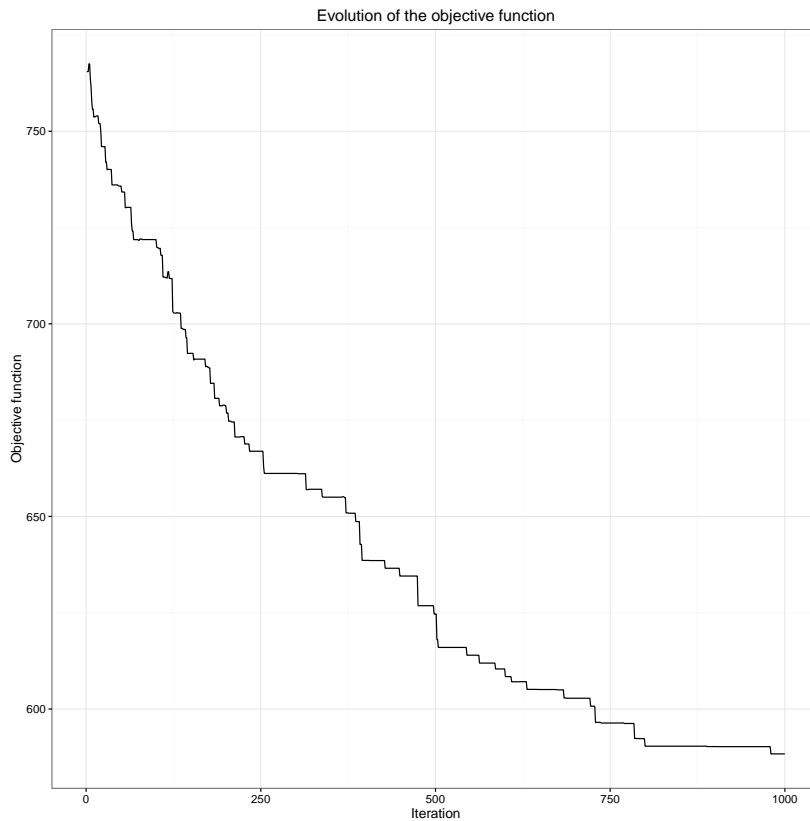
(work in progress)

```
> diamonds$cost <- runif(nrow(diamonds))
> res_cost <- clhs(diamonds, size = 100, progress = FALSE, iter = 1000, cost = 'cost')
```

4 Plotting the results

If you want to report on the cLHS results, e.g. plot the evolution of the objective function, or compare the distribution of attributes in the initial object and in the sampled subset, you need to switch the `simple` option to `FALSE`. Instead of simply returning a numeric vector giving the index of the sampled individuals in the original object, a specific, more complex object will be returned. This object can be handled by a specific `plot` method:

```
> res <- clhs(diamonds, size = 100, simple = FALSE, progress = FALSE, iter = 1000)
> plot(res)
```

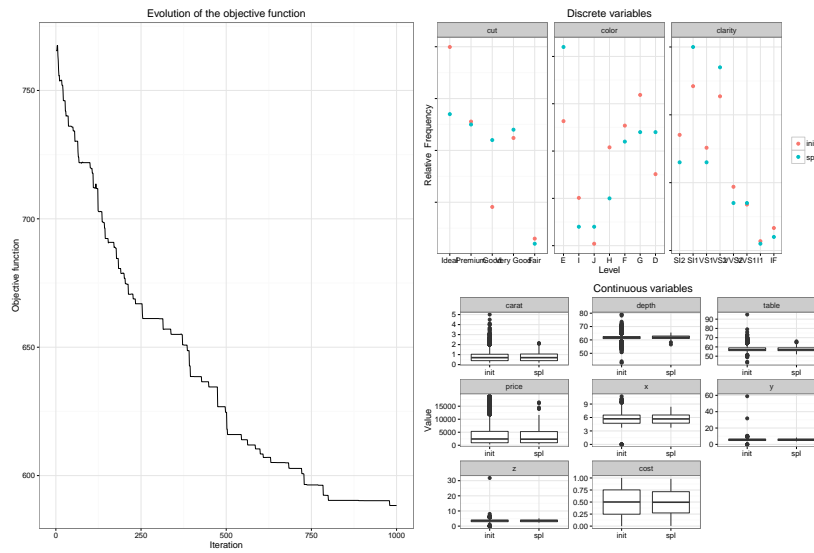


The default plotting method plots the evolution of the objective function with the number of iterations. However, you can get more details using the `modes` option, which controls which indicators are plotted. Three *modes* can be simultaneously plotted:

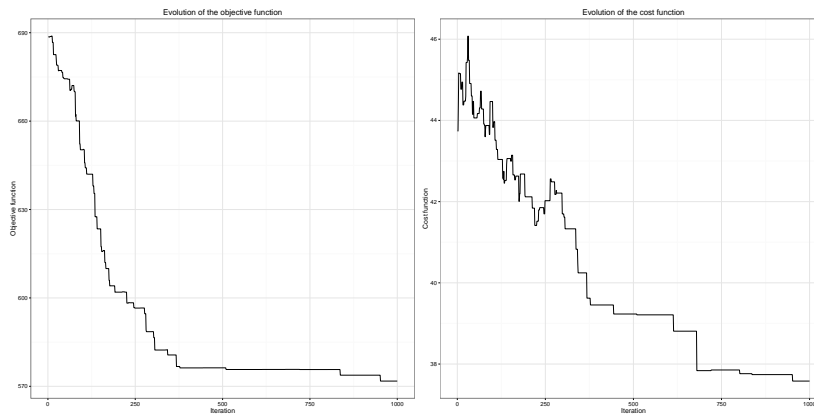
- `obj`: evolution of the objective function (default)
- `cost`: evolution of the cost function (if available)
- `dens` OR `box` OR `hist`: comparison of the distributions of each attribute in the original object and in the proposed sample, respectively using probability density functions, boxplots or histograms. Note that categorical attributes are always reported using dotplots.

These modes should be given as a vector of characters.

```
> plot(res, c('obj', 'box'))
```



```
> res_cost <- clhs(diamonds, size = 100, progress = FALSE, iter = 1000, cost = 'cost', simpl)
> plot(res_cost, c('obj', 'cost'))
```



```
> plot(res_cost, c('obj', 'cost', 'box'))
```

