Extending a Lego package. An extension of the coin package for comparing interventions assigned by dynamic allocation

Johannes Hüsing
Coordination Center for Clinical Trials, Heidelberg Medical School

2008-08-13

Outline

History on Randomization

- Randomization initially had to find its place among systematic designs.
- Paul Martini, a forerunner of clinical trials in Germany, used alternation in an early form of cross-over trials (Shelley & Baur, 1999).
- In clinical trials, randomization had found its place at the top.
- Nevertheless, random allocation sometimes leads to unbalanced distribution between treatment arms.

Some Discussion of Randomized Designs

> *Those who habitually use random assignments may have thought that the issue was finally settled in favour of randomization, but the recent recrudescence of the dispute . . . make[s] it clear that there is still a considerable body of opinion which favour such arrangements.* Yates (1938)

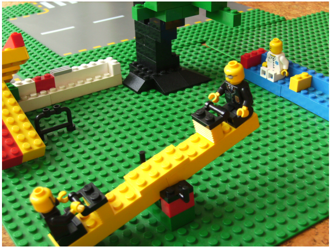
> *But in either case, as soon as the experiment is designed, [experimental conditions] are definitely known . . . , and not random any longer.* Jeffreys (1938)

New Discussion, new Variants

- biased coin (not `coin`!) EFRON (1971)
- minimization TAVES (1974)
- (optimal stratification) SEDRANSK (1974)
- adaptive allocation POCOCK & SIMON (1975)
- dynamic balanced randomization SIGNORINI (1993)
- generalization to continuous covariates incorporating Kullback-Leibler loss

New Discussion, new Variants

- biased coin (not `coin`!) Efron (1971)
- minimization Taves (1974)
- (optimal stratification) Sedransk (1974)
- adaptive allocation Pocock & Simon (1975)
- dynamic balanced randomization Signorini (1993)
- generalization to continuous covariates incorporating Kullback-Leibler loss



Idea: achieve balance

Reservations in Regulatory Bodies

Dynamic allocation is strongly discouraged. However, if it is used, then it is imperative that all factors used in the allocation scheme be included as covariates in the analysis. Even with this requirement, it remains controversial whether the analysis adequately reflect the randomisation scheme.
EMEA (2003)

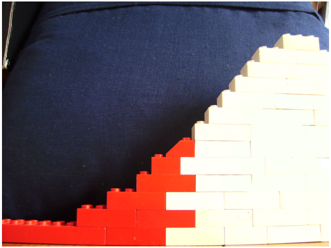
Outline

Reflect the Randomization Scheme

Can be accomplished by re-randomization methods (Simon 1979).

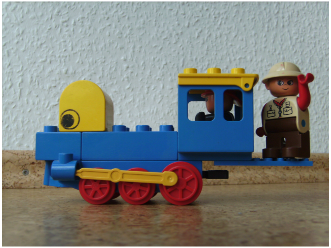
Idea: Repeat the same algorithm over original sequence of patients.

Obtain null distribution, define critical region and determine if test statistic is in it.



This idea is very similar to the permutation test.

Freeloading on an existing interface

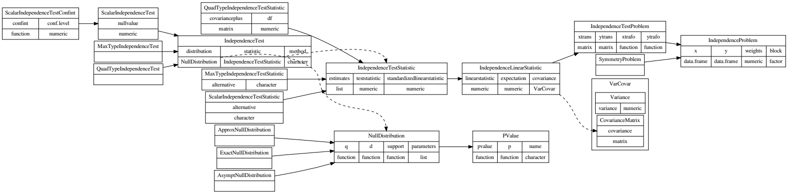


The package `coin` offers a common interface to permutation test methods. Choice between:

1. exact,
2. asymptotic (Strasser and Weber 1999),
3. approximate (Monte Carlo)

methods.
For re-randomization methods, Monte Carlo only choice.

Classes in the coin package

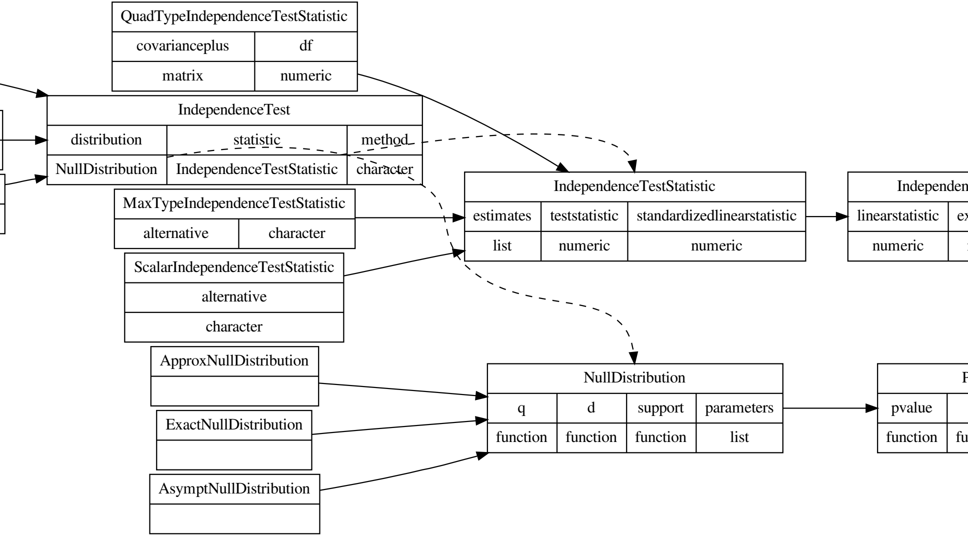


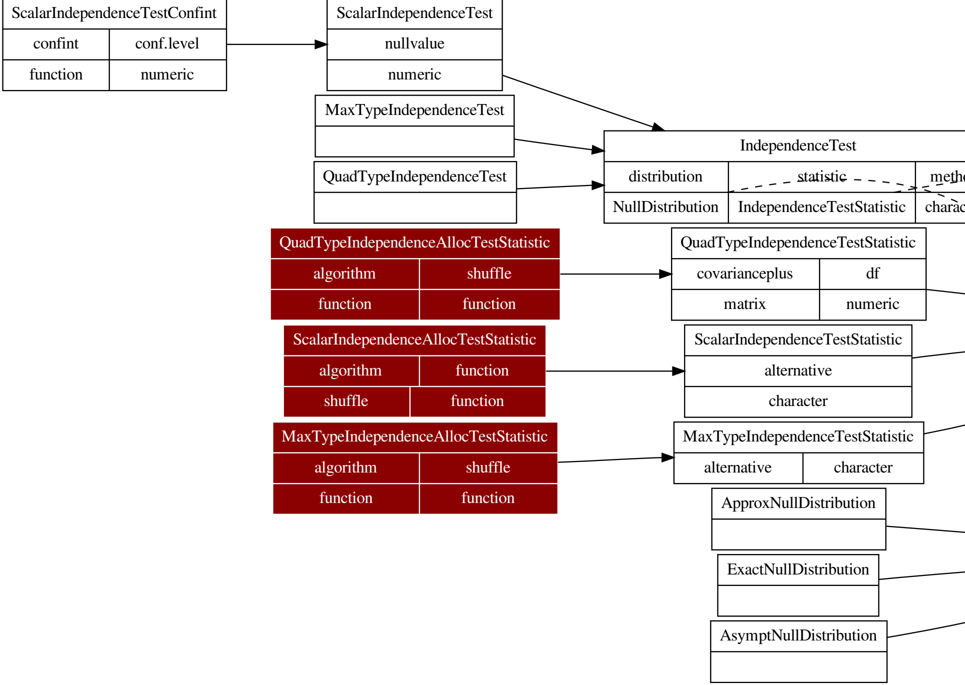
Want to mess with the approximate null distribution.

`ApproxNullDistribution` is a method on `ScalarIndependenceTestStatistic`.

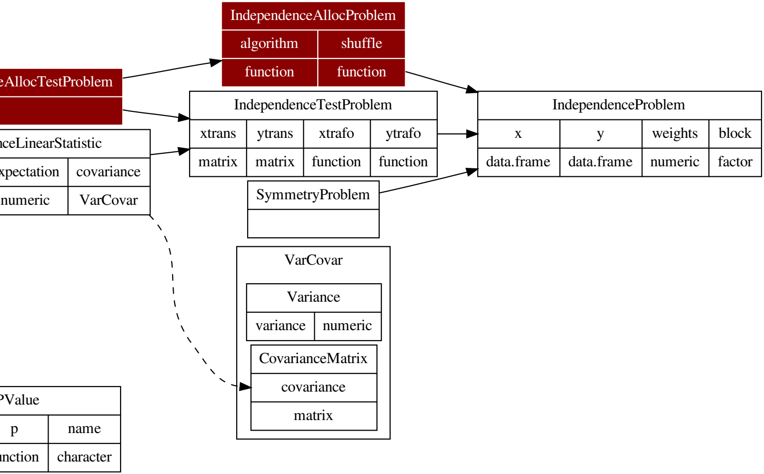
Need algorithm for reallocation and possibly a shuffling algorithm.

Classes in the coin package

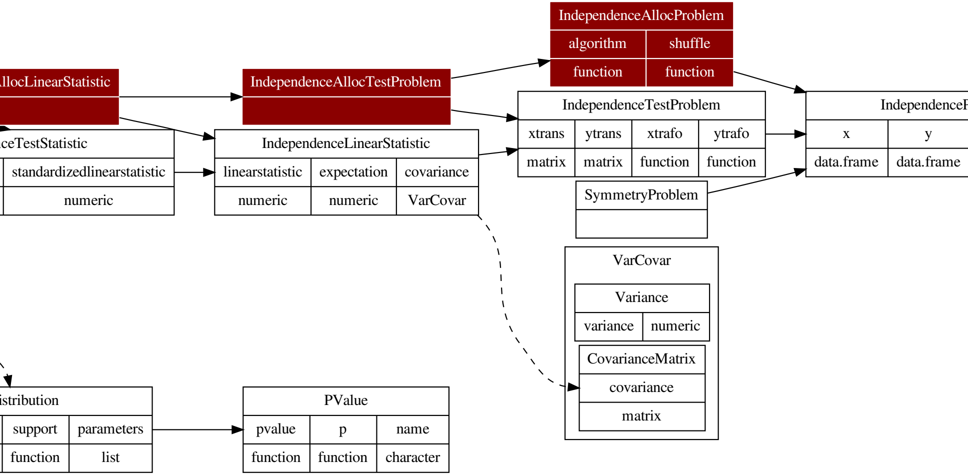



ScalarIndependenceTestConfint
confint
conf.level
function
numeric
ScalarIndependenceTest
nullvalue
numeric
MaxTypeIndependenceTest
QuadTypeIndependenceTest
IndependenceTest
distribution
statistic
NullDistribution
IndependenceTestStatistic
QuadTypeIndependenceAllocTestStatistic
algorithm
shuffle
function
function
QuadTypeIndependenceTestStatistic
covarianceplus
df
matrix
numeric
ScalarIndependenceAllocTestStatistic
algorithm
function
shuffle
function
ScalarIndependenceTestStatistic
alternative
character
MaxTypeIndependenceAllocTestStatistic
algorithm
shuffle
function
function
MaxTypeIndependenceTestStatistic
alternative
character
ApproxNullDistribution
ExactNullDistribution
AsymptNullDistribution

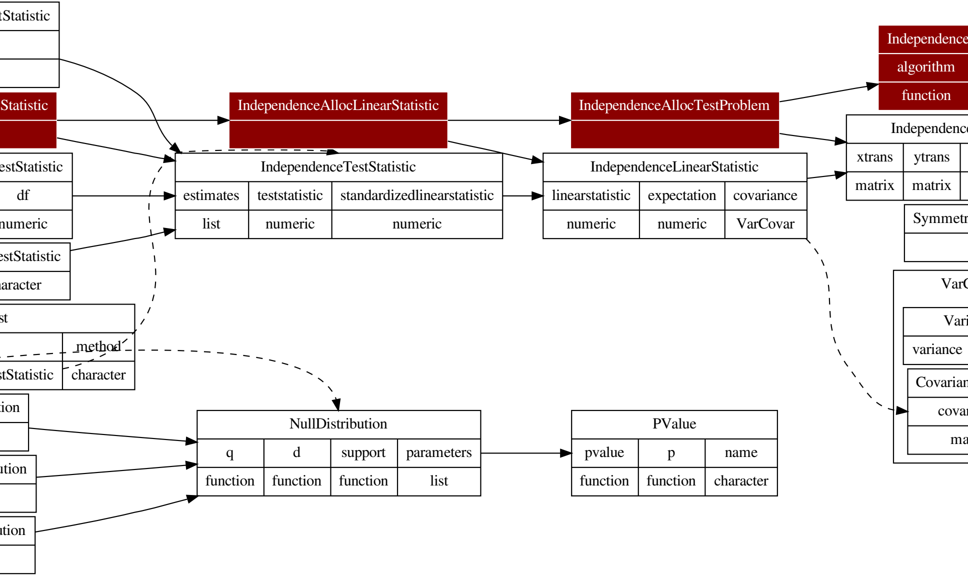
Messing with the `coin` package



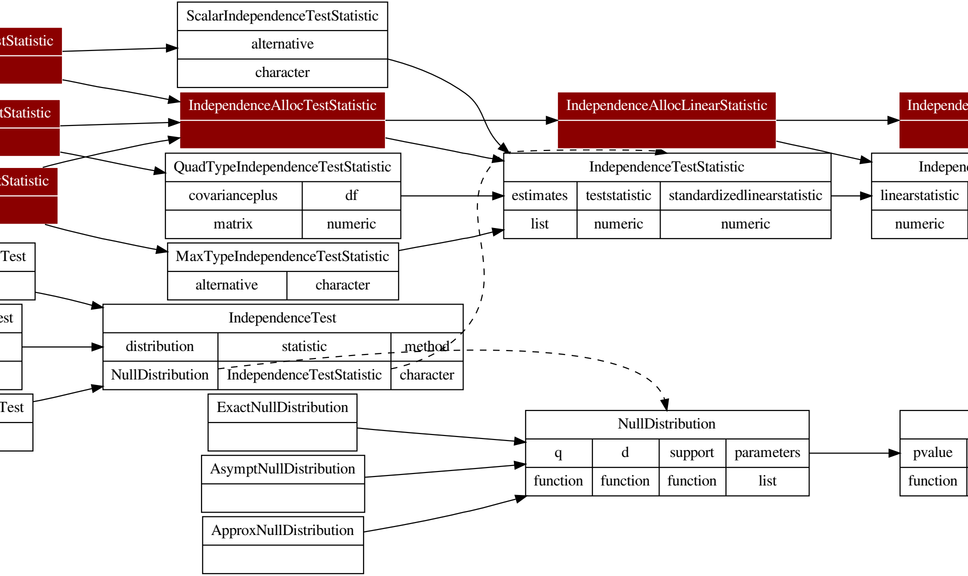
Messing with the coin package



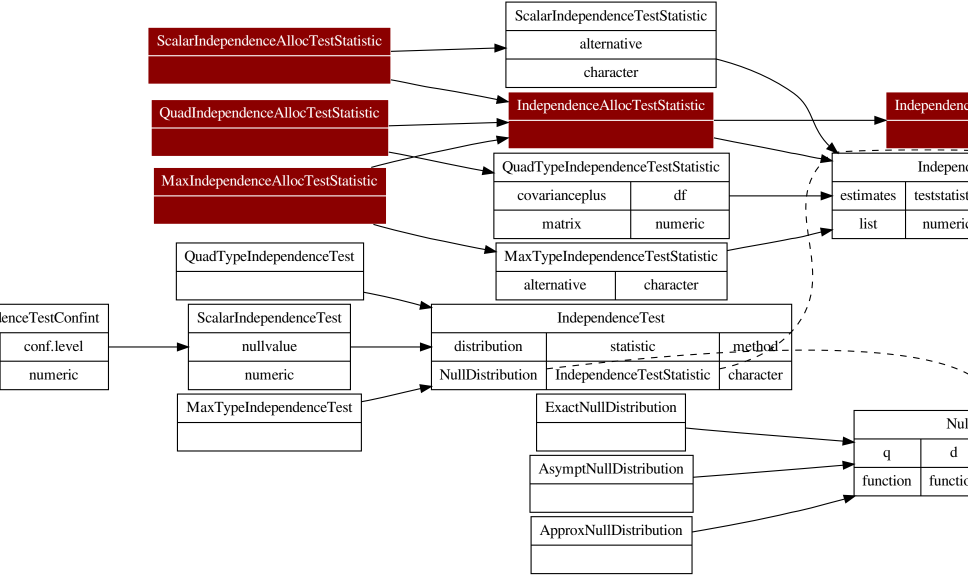
Messing with the `coin` package



Messing with the coin package



Messing with the coin package



Outline

More Lego bricks: goodarms

goodarms

- Takes the data frame of predictors and allocations up to now
- Returns treatment arms with the least loss

$$\text{lossfun()} \rightarrow \text{goodarms()} \rightarrow \text{arms with smallest loss}$$

More Lego pieces: sumloss

sumloss

- Takes the data frame of predictors and list of loss functions
- Returns sum of loss functions applied to data frame

$$
\begin{tikzcd}
 & & \text{contribloss}_{1\ldots p} \arrow[dl] \\
 & \text{sumloss}() \arrow[dl] \arrow[dr] & \\
\text{goodarms}() \arrow[dr] & & \sum^{p} \text{contribloss}(\text{dfr}) \\
 & \text{arms with smallest loss} &
\end{tikzcd}
$$

More Lego pieces: contribloss

factor, *dfr*, lossfun(table)

$\text{contribloss}_{1\ldots p}$

sumloss()

lossfun(*table vs treatment in dfr*)

goodarms()

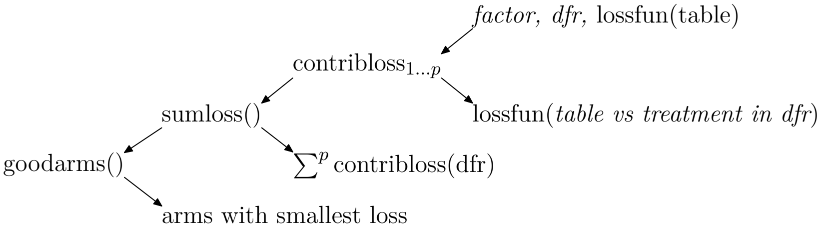
$\sum^{p}$ contribloss(dfr)

arms with smallest loss

contribloss

- Takes the data frame of predictors, factor and loss function on a contingency table
- Returns loss function applied to the data frame

More Lego pieces: contribloss



contribloss

- Takes the data frame of predictors, factor and loss function on a contingency table
- Returns loss function applied to the data frame

Example: Minimize X^2 loss over all marginal tables

```
X2loss ←function(conttab) prop.test(conttab+1)$statistic

X2alg ←function(preddf, xprev=NULL, factors=NULL) {
  current ←length(xprev)
  dataframe ←setupalc(preddf, levels(xprev))
  if (current == 0) dataframe$treatment[1] ←
    sample(levels(dataframe$treatment), 1)
  else {
    dataframe$treatment[1:current] ←xprev
    ga ←goodarms(dataframe, current,
                 function(dfr, crt) {
      sumloss(mapply(function(fac) {
        contribloss(X2loss, fac, dataframe)},
                 factors),
             dfr, crt)})
    dataframe$treatment[current + 1] ←sample(ga, 1)
  }
  dataframe
}
```

Usage of X2alg in coin package

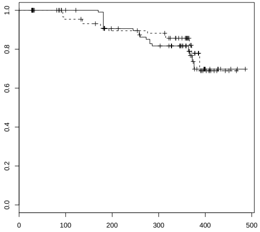
For instance, perform variation of Mann-Whitney U test on trial data.

```
independence_test(outcome ~ treatment, data = trial ,
                  teststat = "scalar", ytrafo = rank,
                  algorithm = X2alg, preddf = stratvars)
```

Outline

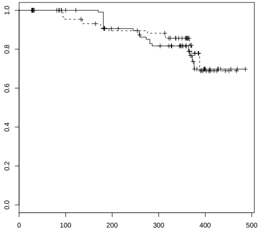
Re-analysis of clinical trial on Graves' disease

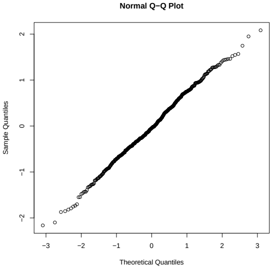
- Effect of levothyroxine on relapse of Graves' disease
- Relapse rates to be similar in the levothyroxine group (20% after 1 year) and the randomized controls (18%) ($p = 0.67$)
- If all (seven, including site!) factors are included in the model, no convergence.
- No effect with re-randomisation analysis ($p = 0.57$)
- standard error of Wald statistic of hazard ratio seems to be considerably lower than 1



Re-analysis of clinical trial on Graves' disease

- Effect of levothyroxine on relapse of Graves' disease
- Relapse rates to be similar in the levothyroxine group (20% after 1 year) and the randomized controls (18%) ($p = 0.67$)
- If all (seven, including site!) factors are included in the model, no convergence.
- No effect with re-randomisation analysis ($p = 0.57$)
- standard error of Wald statistic of hazard ratio seems to be considerably lower than 1





Collaboration with Randi2 team

- Project out of Medical Informatic thesis
- Online randomisation tool in Java
- R integration to interface with algorithms is discussed

The package is mostly vapour

This is work in progress. As of now, two different packages:

- dynamic allocation (including Wei's urn model, biased coin, big stick)
- extension of `coin`

Outline

Computing speed issues

- Focus was on clean rather than speedy implementation
- Heavy usage of data frames seems costly
- Usage of dreaded `for` loops seems unavoidable

Versatility of R

The object system and the possibility to pass functions as arguments allow the users to achieve big structures out of small bricks.

Thanks to:

- Herbert Hirche for using and advocating dynamic allocation.
- Rudolf Hörmann for letting me use his data.
- Ulla Roggenbuck for transferring and explaining the data.
- Elmar Hüsing for letting me use his bricks.