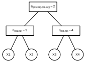
Hierarchical Archimedean Copulae: The HAC Package

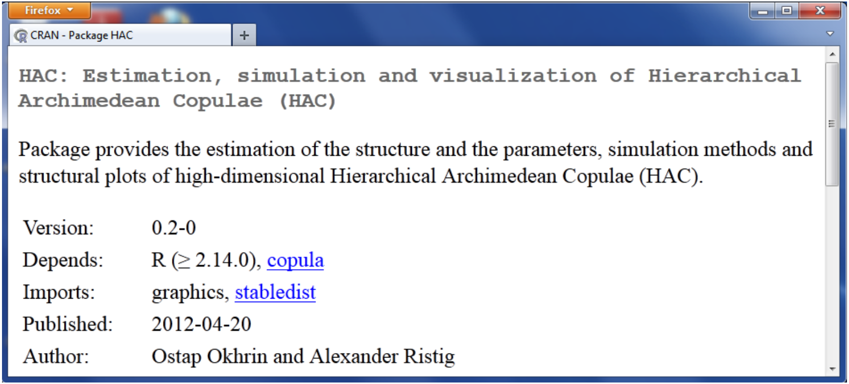
Ostap Okhrin
Alexander Ristig

Ladislaus von Bortkiewicz Chair of Statistics
C.A.S.E. – Center of Applied Statistics and Economics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de
http://www.case.hu-berlin.de

What is R?

- R is a language and environment for statistical computing and graphics.
- Can be easily extended by installing and loading packages.
- Popular copula packages: `copula` or `fCopulae`.
- `HAC` novel features:
  - estimation of the parameters and the structure of Hierarchical Archimedean Copula (HAC),
  - graphical visualization.





HAC: Estimation, simulation and visualization of Hierarchical Archimedean Copulae (HAC)

Package provides the estimation of the structure and the parameters, simulation methods and structural plots of high-dimensional Hierarchical Archimedean Copulae (HAC).

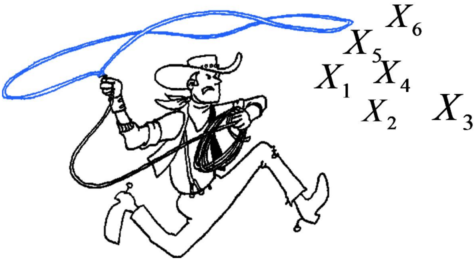
Version:	0.2-0
Depends:	R (≥ 2.14.0), copula
Imports:	graphics, stabledist
Published:	2012-04-20
Author:	Ostap Okhrin and Alexander Ristig

Figure 1: CRAN Website

Copula

Definition (Copula)

A d-dimensional copula C is a joint cumulative distribution function on $[0,1]^d$ with standard uniform marginal distributions.



Theorem (Sklar, 1959)

Let F be a joint distribution function with margins $F_1, \dots, F_d$. Then there exists a copula $C : [0,1]^d \rightarrow [0,1]$ such that, $\forall\, x_1, \dots, x_d \in \overline{\mathbb{R}} = [-\infty, \infty]$,

$$F(x_1, \dots, x_d) = C\{F_1(x_1), \dots, F_d(x_d)\}.$$

If the margins are continuous, then C is unique; otherwise C is uniquely determined on $\operatorname{Ran} F_1 \times \ldots \times \operatorname{Ran} F_d$*, where* $\operatorname{Ran} F_i = F_i\left(\overline{\mathbb{R}}\right)$ *denotes the range of F_i.*

Archimedean copula

Definition (Multivariate Archimedean copula)

A d-dimensional Archimedean copula $C : [0,1]^d \rightarrow [0,1]$ is defined as

$$C\left(u_1, \ldots, u_d\right) = \phi\left\{\phi^{-1}\left(u_1\right) + \cdots + \phi^{-1}\left(u_d\right)\right\},$$

where $\phi : [0,\infty) \rightarrow [0,1]$ is a completely monotone Archimedean copula generator with $\phi(0) = 1$, $\phi(\infty) = 0$.

Example 1

Family	$\phi(u,\theta)$	Parameter range	Independence
Gumbel	$\exp\left(u^{1/\theta}\right)$	$\theta \in [1,\infty)$	$\theta = 1$
Clayton	$(u+1)^{-1/\theta}$	$\theta \in (0,\infty)$	–

Gumbel, Emil Julius on BBI: 

Proposition (Genest and Rivest, 1993)

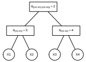
Let X_1 and X_2 be continuous r.v. with unique Archimedean copula C generated by ϕ. Then Kendall's τ is given by

$$\tau\left(X_1, X_2; \theta\right) = 1 + 4 \int_0^1 \frac{\phi^{-1}\left(t, \theta\right)}{\left(\phi^{-1}\right)'\left(t, \theta\right)} dt.$$

Kendall, Maurice George on BBI: 

Example 2

Gumbel: $\tau\left(\cdot, \cdot; \theta\right) = 1 - 1/\theta$
Clayton: $\tau\left(\cdot, \cdot; \theta\right) = \theta / \left(\theta + 2\right)$



HAC

Example 3

3-dimensional fully nested HAC:

$$C\left(u_1, u_2, u_3, \boldsymbol{\theta}\right) = \phi_{\theta_{(\mathbf{12})\mathbf{3}}}\left[\phi^{-1}_{\theta_{(\mathbf{12})\mathbf{3}}} \circ \phi_{\theta_{\mathbf{12}}}\left\{\phi^{-1}_{\theta_{\mathbf{12}}}\left(u_1\right) + \phi^{-1}_{\theta_{\mathbf{12}}}\left(u_2\right)\right\} + \phi^{-1}_{\theta_{(\mathbf{12})\mathbf{3}}}\left(u_3\right)\right]$$

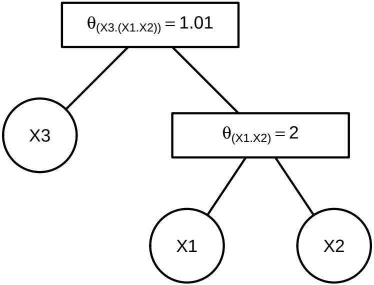


Figure 2: Structure of 3-dim HAC

Example 4

4-dimensional partially nested HAC:

$$C(u_1, \dots, u_4, \boldsymbol{\theta}) = \phi_{(12)(34)}[\phi^{-1}_{(12)(34)} \circ \phi_{12}\left\{\phi^{-1}_{12}(u_1) + \phi^{-1}_{12}(u_2)\right\} + \phi^{-1}_{(12)(34)} \circ \phi_{34}\left\{\phi^{-1}_{34}(u_3) + \phi^{-1}_{34}(u_4)\right\}]$$

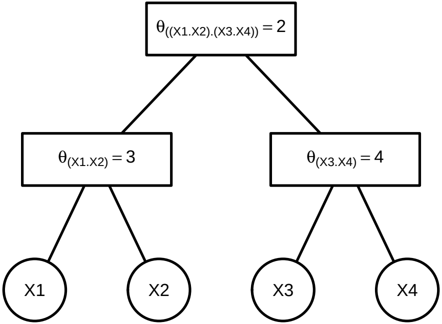


Figure 3: Structure of 4-dim HAC

Portfolio Management

- HAC can be applied to VaR estimation or assessing diversification effects.
- Four stocks: CVX, FP, RDSA and XOM.
- 2011-02-02 to 2012-03-19

```
> price = read.table("stocks")
> ret = diff(log(price), 1)
```

- Residuals of ARMA-GARCH models `res`
- Non-ellipticity? Joint extreme events?

```
> pairs(ret, pch = 20)
```

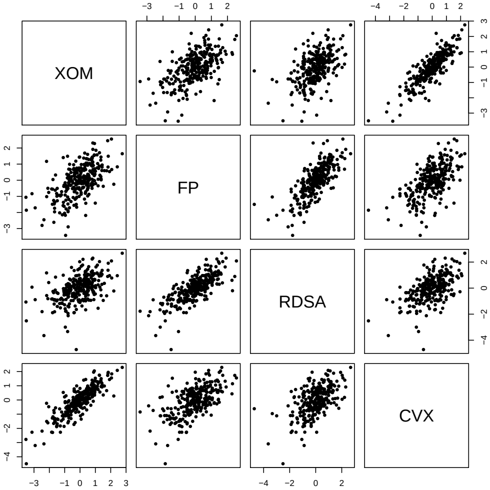


Figure 4: Dependencies of CVX, FP, RDSA and XOM

- Copula estimation based on uniformly distributed margins `ures`

```
> result = estimate.copula(ures)
> plot(result)
```

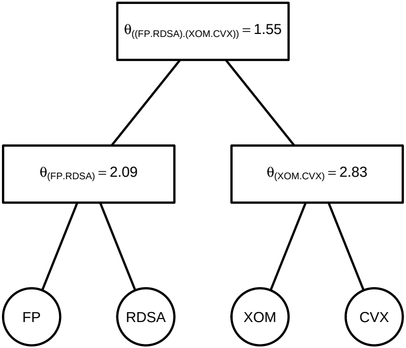
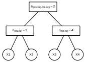


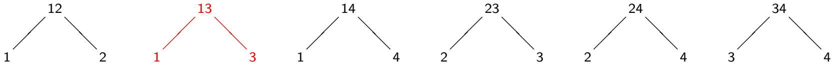
Figure 5: Estimated HAC of the portfolio



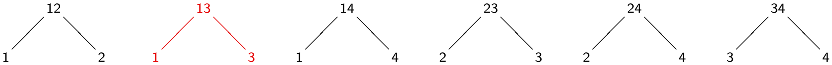
Outline

1. Motivation ✓
2. Estimation
3. `hac` object
4. Graphics
5. Simulation
6. ECDF
7. Density
8. Summary

Estimation procedure

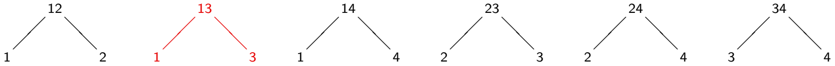


Estimation procedure

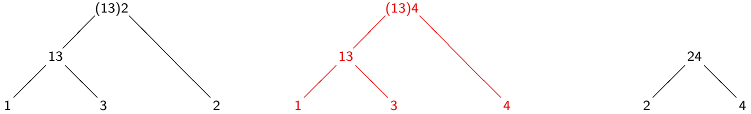


$$\max\{\hat{\theta}_{12}, \hat{\theta}_{13}, \hat{\theta}_{14}, \hat{\theta}_{23}, \hat{\theta}_{24}, \hat{\theta}_{34}\} = \hat{\theta}_{13} \quad \Rightarrow$$

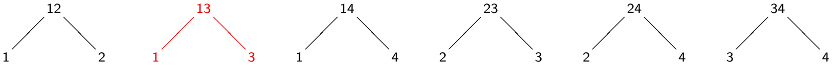
Estimation procedure



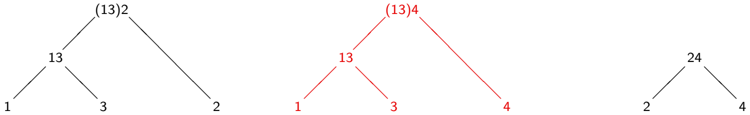
$$\max\{\hat{\theta}_{12}, \hat{\theta}_{13}, \hat{\theta}_{14}, \hat{\theta}_{23}, \hat{\theta}_{24}, \hat{\theta}_{34}\} = \hat{\theta}_{13} \quad \Rightarrow$$



Estimation procedure

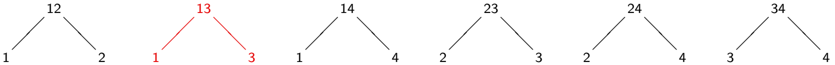


$$\max\{\hat{\theta}_{12}, \hat{\theta}_{13}, \hat{\theta}_{14}, \hat{\theta}_{23}, \hat{\theta}_{24}, \hat{\theta}_{34}\} = \hat{\theta}_{13} \quad \Rightarrow$$

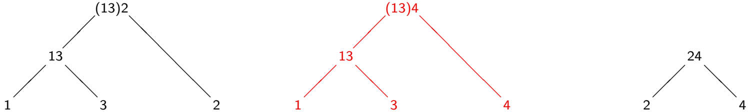


$$\max\{\hat{\theta}_{(13)2}, \hat{\theta}_{(13)4}, \hat{\theta}_{24}\} = \hat{\theta}_{(13)4} \quad \Rightarrow$$

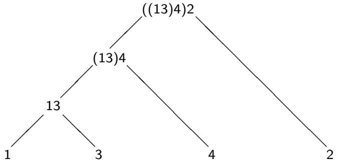
Estimation procedure



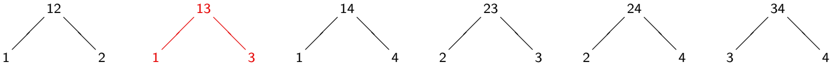
$$\max\{\hat{\theta}_{12}, \hat{\theta}_{13}, \hat{\theta}_{14}, \hat{\theta}_{23}, \hat{\theta}_{24}, \hat{\theta}_{34}\} = \hat{\theta}_{13} \quad \Rightarrow$$



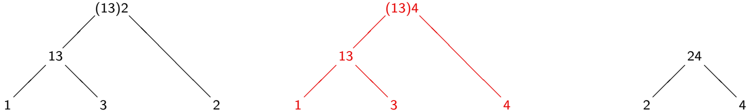
$$\max\{\hat{\theta}_{(13)2}, \hat{\theta}_{(13)4}, \hat{\theta}_{24}\} = \hat{\theta}_{(13)4} \quad \Rightarrow$$



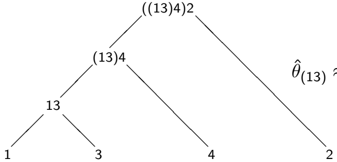
Estimation procedure



$$\max\{\hat{\theta}_{12}, \hat{\theta}_{13}, \hat{\theta}_{14}, \hat{\theta}_{23}, \hat{\theta}_{24}, \hat{\theta}_{34}\} = \hat{\theta}_{13} \quad \Rightarrow$$

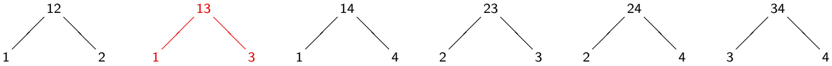


$$\max\{\hat{\theta}_{(13)2}, \hat{\theta}_{(13)4}, \hat{\theta}_{24}\} = \hat{\theta}_{(13)4} \quad \Rightarrow$$

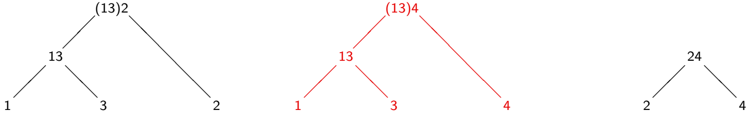


$$\hat{\theta}_{(13)} \approx \hat{\theta}_{((13)4)} \Rightarrow$$

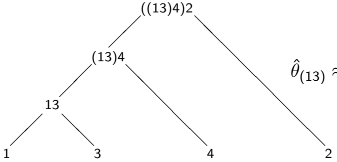
Estimation procedure



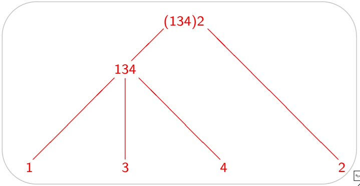
$$\max\{\hat{\theta}_{12}, \hat{\theta}_{13}, \hat{\theta}_{14}, \hat{\theta}_{23}, \hat{\theta}_{24}, \hat{\theta}_{34}\} = \hat{\theta}_{13} \quad \Rightarrow$$



$$\max\{\hat{\theta}_{(13)2}, \hat{\theta}_{(13)4}, \hat{\theta}_{24}\} = \hat{\theta}_{(13)4} \quad \Rightarrow$$

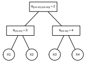


$$\hat{\theta}_{(13)} \approx \hat{\theta}_{((13)4)} \Rightarrow$$



Estimation

- 3 computational blocks:
  1. Specification of the margins
  2. Estimation of the parameters and the structure
  3. Optional aggregation of the binary HAC
- Two estimation procedures: QML and Kendall's τ.
- `estimate.copula` returns a `hac` object.



```r
> result1 = estimate.copula(res, margins = "edf")
> plot(result1)
```

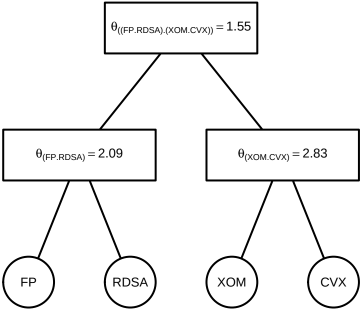


Figure 6: Estimation result

- Note, $C_{\theta_{\mathbf{1(23)}}}\left(C_{\theta_{\mathbf{23}}}(u_2, u_3), u_1\right) = C_{\theta_{\mathbf{123}}}(u_1, u_2, u_3)$, if $|\theta_{1(23)} - \theta_{23}| < \varepsilon, \varepsilon > 0$

- `epsilon = 0.3` leads to a non-binary structure

```
> result2 = estimate.copula(X = res,
+       type = 1, method = ML, epsilon = 0.3,
+       agg.method = "mean", margins = "edf")
> plot(result2)
```

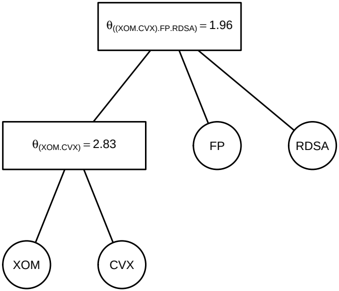


Figure 7: Results of the modified estimation

Objects of the class `hac`

- `hac` and `hac.full` create objects of the class `hac`.
- `hac.full` cannot construct partially nested AC.
- Consider a 5-dimensional fully nested Gumbel HAC:

```
> G1 = hac.full(type = 1,
+       y = c("X1", "X2", "X3", "X4", "X5"),
+       theta = c(1, 1.01, 2, 2.01))
> G1
Class: hac
Generator: Gumbel
((((X5.X4)_{2.01}.X3)_{2}.X2)_{1.01}.X1)_{1}
```

- It is smarter to aggregate the variables X1 and X2 in a first node and the variables X3, X4 and X5 in a second node.

```
> G2 = hac(type = 1,
+        tree = list(list("X3", "X4", "X5", 2.005),
+        "X2", "X1", 1.005))
```

- Substituting of variables for lists leads to arbitrary objects

```
> G3 = hac(type = 1,
+        tree = list(list("Y1", "Y2",
+        list("Z3", "Z4", 3), 2.5),
+        list("Z1", "Z2", 2),
+        list("X1", "X2", 2.4),
+        "X3", "X4", 1.5))
```

Graphics

```
> plot(G3)
```

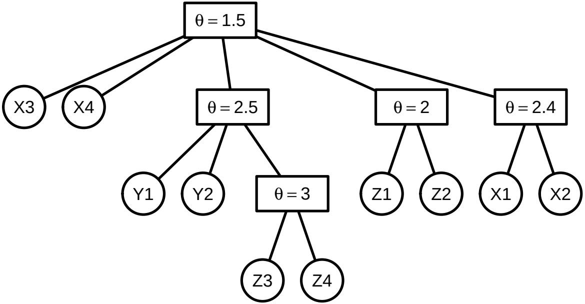


Figure 8: Structure of object G3

```
> plot(G3, digits = 2, theta = TRUE,
+          col = "blue3", fg = "red3",
+      bg = "white", col.t = "blue3")
```

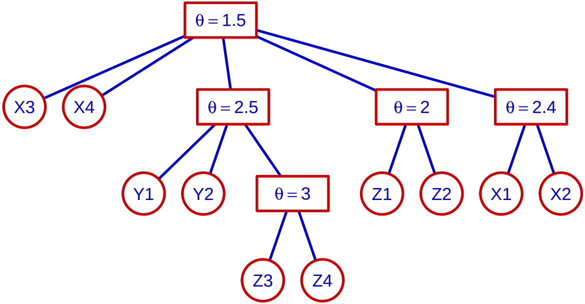


Figure 9: Colored structure of object G3

```
> tree2str(hac = G2, theta = TRUE
+          digits = 3)
[1] ``((X3.X4.X5)_{2.005}.X2.X1)_{1.005}''
> plot(G2, digits = 3, index = TRUE,
+     theta = FALSE)
```

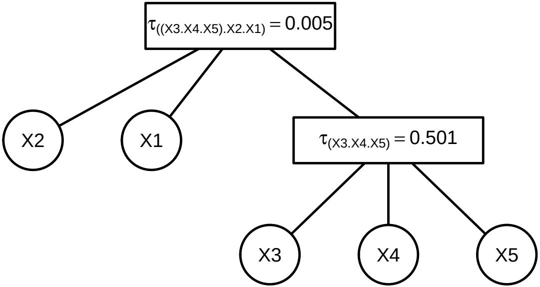


Figure 10: Structure of object G2

Simulation

- Simulation of HAC requires 2 arguments: the number of generated random vectors and a hac object.

```
> sample = rHAC(n = 1500, hac = G2)
```

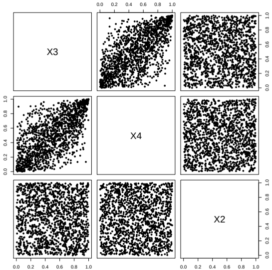
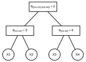


Figure 11: Scatterplot of sample



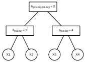
Distribution Functions

- pHAC computes the values of copulae.

```
> cf.values = pHAC(X = sample, hac = G2)
```

- emp.copula.self computes the empirical copula, i.e. $\widehat{C}\left(u_1, \ldots, u_d\right) = n^{-1} \sum_{i=1}^{n} \prod_{j=1}^{d} \mathbf{I}\left\{\widehat{F}_j\left(X_{ij}\right) \leq u_j\right\}$.

```
> ecf.values = emp.copula.self(x = sample,
+         proc = "M", sort = "none", na.rm = FALSE)
```

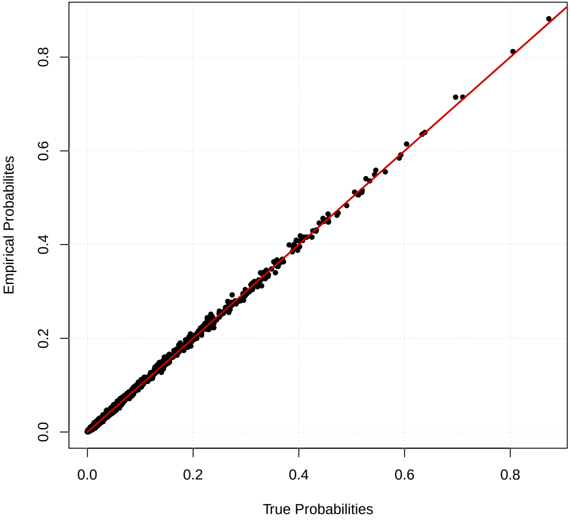


Figure 12: Values of `cf.values` on the x-axis against the values of the `ecf.values`

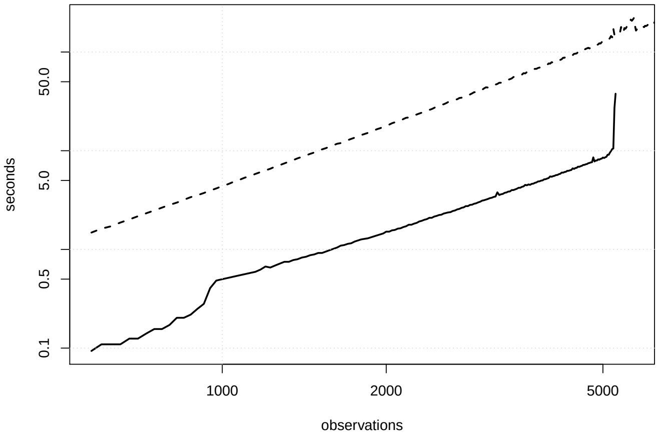


Figure 13: Runtimes of `emp.copula.self` for an increasing sample-size but fixed dimension $d = 5$ plotted on a log-log-scale

Density Functions

- d-dimensional copula density

$$c(u_1, \ldots, u_d) = \frac{\partial^d C(u_1, \ldots, u_d)}{\partial u_1 \cdots \partial u_d}.$$

- `dHAC` returns the values of the analytical density.
  - Requires a data matrix and a `hac` object as arguments.
- Construction of Likelihood functions by `to.logLik`.
- Random sampling using conditional inverse method.

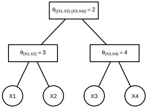
Conclusions

- `HAC` provides simple methods for applying hierarchical Archimedean copulae with usual `R`-syntax.
  - e.g. `dHAC`, `pHAC` and `rHAC` in combination with a `hac` object.
- Innovative functions `estimate.copula` and `plot.hac`.
- We are thankful for critique and suggestions.

Hierarchical Archimedean Copulae: The `HAC` Package

Ostap Okhrin
Alexander Ristig

Ladislaus von Bortkiewicz Chair of Statistics
C.A.S.E. – Center of Applied Statistics and Economics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de
http://www.case.hu-berlin.de

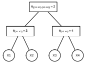


For Further Reading

Marius Hofert (2011)
Efficiently Sampling Nested Archimedean Copulas
Computational Statistics & Data Analysis 55, 57–70

Harry Joe (1997)
Multivariate Models and Dependence Concepts
Chapman & Hall

Harry Joe (2005)
Asymptotic Efficiency of the Two-Stage Estimation Method for Copula-Based Models
Journal of Multivariate Analysis, 94, 401–419



Alexander J. McNeil (2008)
Sampling Nested Archimedean Copulas
Journal of Statistical Computation and Simulation 78, 567–581

Roger B. Nelsen (2006)
An Introduction to Copulas – 2nd ed.
Springer

Ostap Okhrin, Yarema Okhrin and Wolfgang Schmid (2013)
On the Structure and Estimation of Hierarchical Archimedean Copulas
Journal of Econometrics, 173, 189–204

Ostap Okhrin and Alexander Ristig (2014)
Estimation, Simulation and Visualization of Hierarchical Archimedean Copulae (HAC)
available on www.CRAN.R-project.org