# CDVine: An R-package for statistical inference of C- and D-vines

Ulf Schepsmeier & Eike Christian Brechmann

Technische Universität München

May 11, 2011

# Available software for vines

In general...

- "Uncertainty analysis with Correlations" (UNICORN, TU Delft) includes some functionality for vines.

In R...

- Packages for bivariate and multivariate copulas (`copula`, `fCopulae`, `QRMlib`, `nacopula`,...), but nothing regarding vines.
- Daniel Berg (U Oslo/NR): `copulaGOF`/`CopulaLib`.

The package `CDVine` fills this gap!

# Available software for vines

In general...

- "Uncertainty analysis with Correlations" (UNICORN, TU Delft) includes some functionality for vines.

In R...

- Packages for bivariate and multivariate copulas (`copula`, `fCopulae`, `QRMlib`, `nacopula`,...), but nothing regarding vines.
- Daniel Berg (U Oslo/NR): `copulaGOF`/`CopulaLib`.

The package `CDVine` fills this gap!

# Available software for vines

In general...

- "Uncertainty analysis with Correlations" (UNICORN, TU Delft) includes some functionality for vines.

In R...

- Packages for bivariate and multivariate copulas (`copula`, `fCopulae`, `QRMlib`, `nacopula`,...), but nothing regarding vines.
- Daniel Berg (U Oslo/NR): `copulaGOF`/`CopulaLib`.

The package `CDVine` fills this gap!

# Available software for vines

In general...

- "Uncertainty analysis with Correlations" (UNICORN, TU Delft) includes some functionality for vines.

In R...

- Packages for bivariate and multivariate copulas (`copula`, `fCopulae`, `QRMlib`, `nacopula`,...), but nothing regarding vines.
- Daniel Berg (U Oslo/NR): `copulaGOF`/`CopulaLib`.

The package `CDVine` fills this gap!

# Scope of CDVine

CDVine provides functions for bivariate analysis...

- graphical tools
- analytical tools
- selection and estimation of bivariate copulas
- simulation of bivariate copulas

BiCop...

...and for multivariate analysis using canonical (C-) and D-vines.

- sequential and joint maximum likelihood estimation
- simulation of vine copulas
- model selection
- illustration of vine trees

CDVine...

General assumption: Data lies in the unit hypercube $[0,1]^d$.

# Scope of CDVine

CDVine provides functions for bivariate analysis...

- graphical tools
- analytical tools
- selection and estimation of bivariate copulas
- simulation of bivariate copulas

BiCop...

...and for multivariate analysis using canonical (C-) and D-vines.

- sequential and joint maximum likelihood estimation
- simulation of vine copulas
- model selection
- illustration of vine trees

CDVine...

General assumption: Data lies in the unit hypercube $[0,1]^d$.

# Scope of CDVine

CDVine provides functions for bivariate analysis...

- graphical tools
- analytical tools
- selection and estimation of bivariate copulas
- simulation of bivariate copulas

BiCop...

...and for multivariate analysis using canonical (C-) and D-vines.

- sequential and joint maximum likelihood estimation
- simulation of vine copulas
- model selection
- illustration of vine trees

CDVine...

General assumption: Data lies in the unit hypercube $[0,1]^d$.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:
  `family` = 1 Gaussian copula
  `family` = 2 Student-t copula
- One parameter Archimedean copulas:
- Two parameter Archimedean copulas:
  `family` = 7 Clayton-Gumbel (BB1) copula
  `family` = 9 Joe-Clayton (BB7) copula

The independence copula is denoted by 0.

Density, distribution & *h*-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:

|  |  |  |
|---|---|---|
| `family` = | 1 | Gaussian copula |
| `family` = | 2 | Student-t copula |

- One parameter Archimedean copulas:

|  |  |  |
|---|---|---|
| `family` = | 3 | Clayton copula |
| `family` = | 4 | Gumbel copula |
| `family` = | 5 | Frank copula |
| `family` = | 6 | Joe copula |

- Two parameter Archimedean copulas:

|  |  |  |
|---|---|---|
| `family` = | 7 | Clayton-Gumbel (BB1) copula |
| `family` = | 9 | Joe-Clayton (BB7) copula |

The independence copula is denoted by 0.

Density, distribution & $h$-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:

| | | |
|---|---|---|
| `family` = | 1 | Gaussian copula |
| `family` = | 2 | Student-t copula |

- One parameter Archimedean copulas:

| | | |
|---|---|---|
| `family` = | 3 | Clayton copula |
| `family` = | 4 | Gumbel copula |
| `family` = | 5 | Frank copula |
| `family` = | 6 | Joe copula |

- Two parameter Archimedean copulas:

| | | |
|---|---|---|
| `family` = | 7 | Clayton-Gumbel (BB1) copula |
| `family` = | 9 | Joe-Clayton (BB7) copula |

The independence copula is denoted by 0.

Density, distribution & $h$-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:

  `family` = 1 Gaussian copula

  `family` = 2 Student-t copula

- One parameter Archimedean copulas:

  `family` = 3 Clayton copula

  `family` = 4 Gumbel copula

  `family` = 5 Frank copula

  `family` = 6 Joe copula

- Two parameter Archimedean copulas:

  `family` = 7 Clayton-Gumbel (BB1) copula

  `family` = 9 Joe-Clayton (BB7) copula

The independence copula is denoted by 0.

Density, distribution & *h*-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:

| | |
|---|---|
| `family` = 1 | Gaussian copula |
| `family` = 2 | Student-t copula |

- One parameter Archimedean copulas:

| | |
|---|---|
| `family` = 13 | survival Clayton copula |
| `family` = 14 | survival Gumbel copula |
| `family` = 5 | Frank copula |
| `family` = 16 | survival Joe copula |

- Two parameter Archimedean copulas:

| | |
|---|---|
| `family` = 7 | Clayton-Gumbel (BB1) copula |
| `family` = 9 | Joe-Clayton (BB7) copula |

The independence copula is denoted by 0.

Density, distribution & $h$-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:

| | | |
|---|---|---|
| `family` = | 1 | Gaussian copula |
| `family` = | 2 | Student-t copula |

- One parameter Archimedean copulas:

| | | |
|---|---|---|
| `family` = | 23 | rotated Clayton copula (90 degrees) |
| `family` = | 24 | rotated Gumbel copula (90 degrees) |
| `family` = | 5 | Frank copula |
| `family` = | 26 | rotated Joe copula (90 degrees) |

- Two parameter Archimedean copulas:

| | | |
|---|---|---|
| `family` = | 7 | Clayton-Gumbel (BB1) copula |
| `family` = | 9 | Joe-Clayton (BB7) copula |

The independence copula is denoted by 0.

Density, distribution & $h$-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:

  `family` = 1 Gaussian copula  
  `family` = 2 Student-t copula

- One parameter Archimedean copulas:

  `family` = 33 rotated Clayton copula (270 degrees)  
  `family` = 34 rotated Gumbel copula (270 degrees)  
  `family` = 5 Frank copula  
  `family` = 36 rotated Joe copula (270 degrees)

- Two parameter Archimedean copulas:

  `family` = 7 Clayton-Gumbel (BB1) copula  
  `family` = 9 Joe-Clayton (BB7) copula

The independence copula is denoted by 0.

Density, distribution & *h*-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# The building blocks: Bivariate copula families

Each family is denoted by a number to shorten notation.

- Elliptical copulas:
  `family` = 1 Gaussian copula
  `family` = 2 Student-t copula
- One parameter Archimedean copulas:
  `family` = 33 rotated Clayton copula (270 degrees)
  `family` = 34 rotated Gumbel copula (270 degrees)
  `family` = 5 Frank copula
  `family` = 36 rotated Joe copula (270 degrees)
- Two parameter Archimedean copulas:
  `family` = 7 Clayton-Gumbel (BB1) copula
  `family` = 9 Joe-Clayton (BB7) copula

The independence copula is denoted by 0.

Density, distribution & *h*-functions: `BiCopPDF`, `BiCopCDF` & `BiCopHfunc`.

# Rotation of copulas

- Rotate asymmetric Archimedean copulas to capture negative dependence: if $(U_1, U_2) \sim C_{90^\circ}$, then $(1 - U_1, U_2) \sim C_{0^\circ}$.
- Survival copulas correspond to rotation by 180 degrees.

### Clayton copulas rotated by 0, 90, 180 and 270 degrees

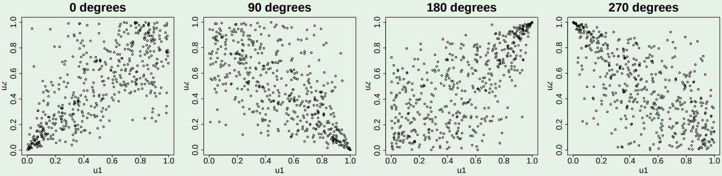
```
> dat0 = BiCopSim(N = 500, family = 3, par = 2)
> dat90 = BiCopSim(N = 500, family = 23, par = -2)
> dat180 = BiCopSim(N = 500, family = 13, par = 2)
> dat270 = BiCopSim(N = 500, family = 33, par = -2)
```

# Rotation of copulas

- Rotate asymmetric Archimedean copulas to capture negative dependence: if $(U_1, U_2) \sim C_{90^\circ}$, then $(1 - U_1, U_2) \sim C_{0^\circ}$.
- Survival copulas correspond to rotation by 180 degrees.

## Clayton copulas rotated by 0, 90, 180 and 270 degrees

```
> dat0 = BiCopSim(N = 500, family = 3, par = 2)
> dat90 = BiCopSim(N = 500, family = 23, par = -2)
> dat180 = BiCopSim(N = 500, family = 13, par = 2)
> dat270 = BiCopSim(N = 500, family = 33, par = -2)
```



# Two parameter Archimedean copulas

- The BB1 and BB7 copulas can model asymmetric dependence with different non-zero lower and upper tail dependence.
- Density expressions and derivatives are however numerically involved.

### Clayton-Gumbel (BB1) copula

```
> BiCopPar2Tau(family = 7,
+     par = 0.25, par2 = 2.5)

[1] 0.64

> BiCopPar2TailDep(family = 7,
+     par = 0.25, par2 = 2.5)

$lower
[1] 0.33

$upper
[1] 0.68
```

# Two parameter Archimedean copulas

- The BB1 and BB7 copulas can model asymmetric dependence with different non-zero lower and upper tail dependence.
- Density expressions and derivatives are however numerically involved.

### Clayton-Gumbel (BB1) copula

```
> BiCopPar2Tau(family = 7,
+     par = 0.25, par2 = 2.5)

[1] 0.64

> BiCopPar2TailDep(family = 7,
+     par = 0.25, par2 = 2.5)

$lower
[1] 0.33

$upper
[1] 0.68
```

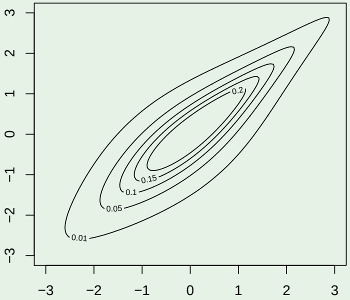
# Two parameter Archimedean copulas

- The BB1 and BB7 copulas can model asymmetric dependence with different non-zero lower and upper tail dependence.
- Density expressions and derivatives are however numerically involved.

## Clayton-Gumbel (BB1) copula

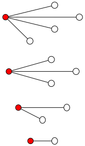
```
> BiCopPar2Tau(family = 7,
+      par = 0.25, par2 = 2.5)

[1] 0.64

> BiCopPar2TailDep(family = 7,
+      par = 0.25, par2 = 2.5)

$lower
[1] 0.33

$upper
[1] 0.68
```

```
> BiCopMetaContour(family = 7,
+      par = 0.25, par2 = 2.5)
```



# Order of variables in C- and D-vines

- C-vine: specify characteristic order of root nodes for each tree.

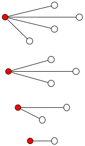


- D-vine: specify characteristic order of nodes in the first tree.

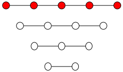
In CDVine it is assumed that this order is given by the order of the variables in the data set under investigation.

# Order of variables in C- and D-vines

- C-vine: specify characteristic order of root nodes for each tree.



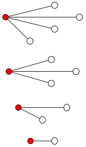
- D-vine: specify characteristic order of nodes in the first tree.



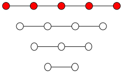
In CDVine it is assumed that this order is given by the order of the variables in the data set under investigation.

# Order of variables in C- and D-vines

- C-vine: specify characteristic order of root nodes for each tree.



- D-vine: specify characteristic order of nodes in the first tree.



In CDVine it is assumed that this order is given by the order of the variables in the data set under investigation.

# Specification of C- and D-vine copula models

Copula families (`family`) and parameters (`par` and `par2`) are specified as vectors of length $d(d-1)/2$ ($d = \#$variables). Entries:

- C-vine:

$$\underbrace{(1,2),(1,3),(1,4),\ldots,(1,d)}_{\text{1st tree}},\underbrace{(2,3|1),(2,4|1),\ldots,(2,d|1)}_{\text{2nd tree}},$$

$$\underbrace{(3,4|1,2),(3,5|1,2),\ldots,(3,d|1,2)}_{\text{3rd tree}},\ldots,\underbrace{(d-1,d|1,\ldots,d-2)}_{(d-1)\text{-th tree}}.$$

- D-vine:

$$\underbrace{(1,2),(2,3),(3,4),\ldots,(d-1,d)}_{\text{1st tree}},\underbrace{(1,3|2),(2,4|3),\ldots,(d-2,d|d-1)}_{\text{2nd tree}},$$

$$\underbrace{(1,4|2,3),(2,5|3,4),\ldots,(d-3,d|d-2,d-1)}_{\text{3rd tree}},\ldots,\underbrace{(1,d|2,\ldots,d-1)}_{(d-1)\text{-th tree}}.$$

# Specification of C- and D-vine copula models

Copula families (`family`) and parameters (`par` and `par2`) are specified as vectors of length $d(d-1)/2$ ($d = \#$variables). Entries:

- C-vine:

$$\underbrace{(1,2),(1,3),(1,4),...,(1,d)}_{\text{1st tree}},\underbrace{(2,3|1),(2,4|1),...,(2,d|1)}_{\text{2nd tree}},$$

$$\underbrace{(3,4|1,2),(3,5|1,2),...,(3,d|1,2)}_{\text{3rd tree}},...,\underbrace{(d-1,d|1,...,d-2)}_{(d-1)\text{-th tree}}.$$

- D-vine:

$$\underbrace{(1,2),(2,3),(3,4),...,(d-1,d)}_{\text{1st tree}},\underbrace{(1,3|2),(2,4|3),...,(d-2,d|d-1)}_{\text{2nd tree}},$$

$$\underbrace{(1,4|2,3),(2,5|3,4),...,(d-3,d|d-2,d-1)}_{\text{3rd tree}},...,\underbrace{(1,d|2,...,d-1)}_{(d-1)\text{-th tree}}.$$

# Specification of C- and D-vine copula models

Copula families (`family`) and parameters (`par` and `par2`) are specified as vectors of length $d(d-1)/2$ ($d = \#$variables). Entries:

- C-vine:

$$\underbrace{(1,2),(1,3),(1,4),...,(1,d)}_{\text{1st tree}},\underbrace{(2,3|1),(2,4|1),...,(2,d|1)}_{\text{2nd tree}},$$

$$\underbrace{(3,4|1,2),(3,5|1,2),...,(3,d|1,2)}_{\text{3rd tree}},...,\underbrace{(d-1,d|1,...,d-2)}_{(d-1)\text{-th tree}}.$$

- D-vine:

$$\underbrace{(1,2),(2,3),(3,4),...,(d-1,d)}_{\text{1st tree}},\underbrace{(1,3|2),(2,4|3),...,(d-2,d|d-1)}_{\text{2nd tree}},$$

$$\underbrace{(1,4|2,3),(2,5|3,4),...,(d-3,d|d-2,d-1)}_{\text{3rd tree}},...,\underbrace{(1,d|2,...,d-1)}_{(d-1)\text{-th tree}}.$$

# Example

Transformed residuals of daily log returns of major world stock indices in 2009 and 2010 (396 observations):

- S&P 500
- Nikkei 225
- SSE Composite Index
- DAX
- CAC 40
- FTSE 100 Index

Load into workspace:

```
> data(worldindices)
```

# Copula selection

- Manually using tools for bivariate analysis (e.g., contour plots or goodness-of-fit tests).
- Automatically using AIC or BIC: BiCopSelect (bivariate) or CDVineCopSelect (multivariate).

```
> DV = CDVineCopSelect(data = worldindices[, c(1,4,5,6,2,3)],
+      familyset = NA, type = "DVine", selectioncrit = "AIC")

$family
 [1]  9  2  2  9  9  1  2 14 14  3 24  5  5  2  6

$par
 [1]  1.91  0.96  0.94  1.17  1.15  0.15  0.07  1.09
 [9]  1.08  0.13 -1.08 -0.39 -0.27 -0.01  1.04

$par2
 [1]  1.21 14.93 13.76  0.22  0.32  0.00  8.71  0.00
[12]  0.00  0.00  0.00  0.00  0.00 11.28  0.00
```

# Copula selection

- Manually using tools for bivariate analysis (e.g., contour plots or goodness-of-fit tests).
- Automatically using AIC or BIC: `BiCopSelect` (bivariate) or `CDVineCopSelect` (multivariate).

```
> DV = CDVineCopSelect(data = worldindices[, c(1,4,5,6,2,3)],
+      familyset = NA, type = "DVine", selectioncrit = "AIC")

$family
 [1]  9  2  2  9  9  1  2 14 14  3 24  5  5  2  6

$par
 [1]  1.91  0.96  0.94  1.17  1.15  0.15  0.07  1.09
 [9]  1.08  0.13 -1.08 -0.39 -0.27 -0.01  1.04

$par2
 [1]  1.21 14.93 13.76  0.22  0.32  0.00  8.71  0.00
[12]  0.00  0.00  0.00  0.00  0.00 11.28  0.00
```

# Copula selection

- Manually using tools for bivariate analysis (e.g., contour plots or goodness-of-fit tests).
- Automatically using AIC or BIC: `BiCopSelect` (bivariate) or `CDVineCopSelect` (multivariate).

```
> DV = CDVineCopSelect(data = worldindices[, c(1,4,5,6,2,3)],
+     familyset = NA, type = "DVine", selectioncrit = "AIC")
```

```
$family
 [1]  9  2  2  9  9  1  2 14 14  3 24  5  5  2  6

$par
 [1]  1.91  0.96  0.94  1.17  1.15  0.15  0.07  1.09
 [9]  1.08  0.13 -1.08 -0.39 -0.27 -0.01  1.04

$par2
 [1]  1.21 14.93 13.76  0.22  0.32  0.00  8.71  0.00
[12]  0.00  0.00  0.00  0.00  0.00 11.28  0.00
```

# Copula selection

- Manually using tools for bivariate analysis (e.g., contour plots or goodness-of-fit tests).
- Automatically using AIC or BIC: `BiCopSelect` (bivariate) or `CDVineCopSelect` (multivariate).

```
> DV = CDVineCopSelect(data = worldindices[, c(1,4,5,6,2,3)],
+      familyset = NA, type = "DVine", selectioncrit = "AIC")

$family
 [1]  9  2  2  9  9  1  2 14 14  3 24  5  5  2  6

$par
 [1]  1.91  0.96  0.94  1.17  1.15  0.15  0.07  1.09
 [9]  1.08  0.13 -1.08 -0.39 -0.27 -0.01  1.04

$par2
 [1]  1.21 14.93 13.76  0.22  0.32  0.00  8.71  0.00
[12]  0.00  0.00  0.00  0.00  0.00 11.28  0.00
```

# Parameter estimation

- Sequential estimation (based on `BiCopEst`)
  - either using bivariate inversion of Kendall's $\tau$:

    ```
    > CDVineSeqEst(data, family, type, method="itau")
    ```
  - or bivariate maximum likelihood estimation:

    ```
    > CDVineSeqEst(data, family, type, method="mle")
    ```
- Maximum likelihood estimation of all parameters jointly (log-likelihood computation: `CDVineLogLik`).

```
> DV.mle = CDVineMLE(data = worldindices[, c(1,4,5,6,2,3)],
+      family = DV$family, start = DV$par, start2 = DV$par2, type = "DVine")

$par
 [1]  1.91  0.96  0.94  1.15  1.13  0.16  0.07  1.10 ...


$par2
 [1]  1.22 14.93 13.76  0.23  0.32  0.00  8.71  0.00 ...


$loglik
[1] 1193

...
```

# Parameter estimation

- Sequential estimation (based on `BiCopEst`)
  - either using bivariate inversion of Kendall's $\tau$:

    ```
    > CDVineSeqEst(data, family, type, method="itau")
    ```
  - or bivariate maximum likelihood estimation:

    ```
    > CDVineSeqEst(data, family, type, method="mle")
    ```
- Maximum likelihood estimation of all parameters jointly (log-likelihood computation: `CDVineLogLik`).

```
> DV.mle = CDVineMLE(data = worldindices[, c(1,4,5,6,2,3)],
+      family = DV$family, start = DV$par, start2 = DV$par2, type = "DVine")

$par
 [1]  1.91  0.96  0.94  1.15  1.13  0.16  0.07  1.10 ...

$par2
 [1]  1.22 14.93 13.76  0.23  0.32  0.00  8.71  0.00 ...

$loglik
[1] 1193
...
```

# Parameter estimation

- Sequential estimation (based on `BiCopEst`)
  - either using bivariate inversion of Kendall's $\tau$:
    `> CDVineSeqEst(data, family, type, method="itau")`
  - or bivariate maximum likelihood estimation:
    `> CDVineSeqEst(data, family, type, method="mle")`
- Maximum likelihood estimation of all parameters jointly (log-likelihood computation: `CDVineLogLik`).

```
> DV.mle = CDVineMLE(data = worldindices[, c(1,4,5,6,2,3)],
+     family = DV$family, start = DV$par, start2 = DV$par2, type = "DVine")

$par
 [1]  1.91  0.96  0.94  1.15  1.13  0.16  0.07  1.10 ...

$par2
 [1]  1.22 14.93 13.76  0.23  0.32  0.00  8.71  0.00 ...

$loglik
[1] 1193

...
```

# Parameter estimation

- Sequential estimation (based on `BiCopEst`)
  - either using bivariate inversion of Kendall's $\tau$:

    ```
    > CDVineSeqEst(data, family, type, method="itau")
    ```

  - or bivariate maximum likelihood estimation:

    ```
    > CDVineSeqEst(data, family, type, method="mle")
    ```

- Maximum likelihood estimation of all parameters jointly (log-likelihood computation: `CDVineLogLik`).

```
> DV.mle = CDVineMLE(data = worldindices[, c(1,4,5,6,2,3)],
+     family = DV$family, start = DV$par, start2 = DV$par2, type = "DVine")

$par
 [1]  1.91  0.96  0.94  1.15  1.13  0.16  0.07  1.10 ...

$par2
 [1]  1.22 14.93 13.76  0.23  0.32  0.00  8.71  0.00 ...

$loglik
[1] 1193
...
```

# Discrimination among vine models

- AIC and BIC: `CDVineAIC` and `CDVineBIC`.
- Tests by Vuong (1989) and by Clarke (2007) for non-nested comparisons of two models.

```
> CDVineVuongTest(worldindices,
+     Model1.order = c(1,4,5,6,2,3), Model2.order = c(5,2,6,1,4,3),
+     Model1.family = DV$family, Model2.family = CV$family, ...)

$statistic
[1] 0.05

$statistic.Akaike
[1] -0.20

$statistic.Schwarz
[1] -0.70

$p.value
[1] 0.96

...
```

# Discrimination among vine models

- AIC and BIC: `CDVineAIC` and `CDVineBIC`.
- Tests by Vuong (1989) and by Clarke (2007) for non-nested comparisons of two models.

```
> CDVineVuongTest(worldindices,
+     Model1.order = c(1,4,5,6,2,3), Model2.order = c(5,2,6,1,4,3),
+     Model1.family = DV$family, Model2.family = CV$family, ...)

$statistic
[1] 0.05

$statistic.Akaike
[1] -0.20

$statistic.Schwarz
[1] -0.70

$p.value
[1] 0.96

...
```

# Discrimination among vine models

- AIC and BIC: `CDVineAIC` and `CDVineBIC`.
- Tests by Vuong (1989) and by Clarke (2007) for non-nested comparisons of two models.

```
> CDVineVuongTest(worldindices,
+     Model1.order = c(1,4,5,6,2,3), Model2.order = c(5,2,6,1,4,3),
+     Model1.family = DV$family, Model2.family = CV$family, ...)

$statistic
[1] 0.05

$statistic.Akaike
[1] -0.20

$statistic.Schwarz
[1] -0.70

$p.value
[1] 0.96

...
```

# Discrimination among vine models

- AIC and BIC: `CDVineAIC` and `CDVineBIC`.
- Tests by Vuong (1989) and by Clarke (2007) for non-nested comparisons of two models.

```
> CDVineVuongTest(worldindices,
+     Model1.order = c(1,4,5,6,2,3), Model2.order = c(5,2,6,1,4,3),
+     Model1.family = DV$family, Model2.family = CV$family, ...)

$statistic
[1] 0.05

$statistic.Akaike
[1] -0.20

$statistic.Schwarz
[1] -0.70

$p.value
[1] 0.96

...
```

# Illustrating C- and D-vine trees

Respective first trees of the C- and D-vine copulas models:

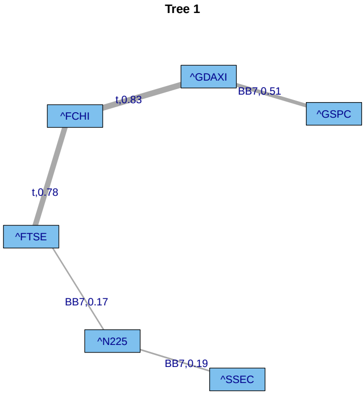
```
> CDVineTreePlot(family = DV$family, ..., tree = 1,
+     names = colnames(worldindices[ ,c(1,4,5,6,2,3)]),
+     edge.labels = c("family", "theotau"))
> CDVineTreePlot(family = CV$family, ...)
```

# Illustrating C- and D-vine trees

Respective first trees of the C- and D-vine copulas models:

```
> CDVineTreePlot(family = DV$family, ..., tree = 1,
+      names = colnames(worldindices[ ,c(1,4,5,6,2,3)]),
+      edge.labels = c("family", "theotau"))
```

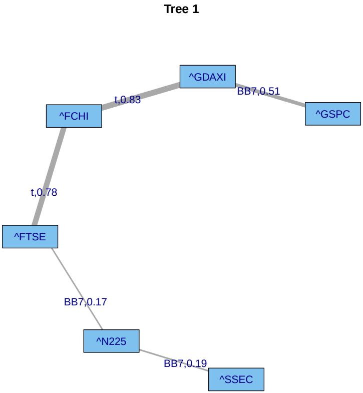
```
> CDVineTreePlot(family = CV$family, ...)
```



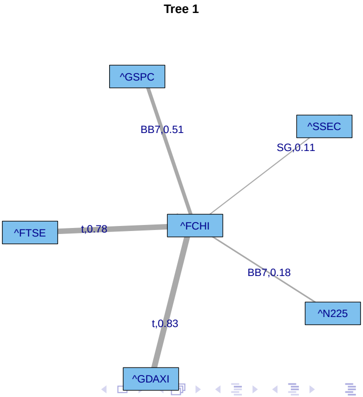
# Illustrating C- and D-vine trees

Respective first trees of the C- and D-vine copulas models:

```
> CDVineTreePlot(family = DV$family, ..., tree = 1,
+     names = colnames(worldindices[ ,c(1,4,5,6,2,3)]),
+     edge.labels = c("family", "theotau"))
> CDVineTreePlot(family = CV$family, ...)
```



Tree 1



Tree 1

# Summary & conclusion

- Strong (positive) dependence among European indices.
- European indices central to explaining overall dependence.
- Indication of medium to strong tail dependence.
- Asymmetric (conditional) dependencies.
- No significant difference between C- and D-vine copula models.
  → Different interpretations possible based on tree structures.

# Summary & conclusion

- Strong (positive) dependence among European indices.
- European indices central to explaining overall dependence.
- Indication of medium to strong tail dependence.
- Asymmetric (conditional) dependencies.
- No significant difference between C- and D-vine copula models.
  → Different interpretations possible based on tree structures.

# Summary & conclusion

- Strong (positive) dependence among European indices.
- European indices central to explaining overall dependence.
- Indication of medium to strong tail dependence.
- Asymmetric (conditional) dependencies.
- No significant difference between C- and D-vine copula models.  
  → Different interpretations possible based on tree structures.

# Summary & conclusion

- Strong (positive) dependence among European indices.
- European indices central to explaining overall dependence.
- Indication of medium to strong tail dependence.
- Asymmetric (conditional) dependencies.
- No significant difference between C- and D-vine copula models.
  → Different interpretations possible based on tree structures.

# Summary & conclusion

- Strong (positive) dependence among European indices.
- European indices central to explaining overall dependence.
- Indication of medium to strong tail dependence.
- Asymmetric (conditional) dependencies.
- No significant difference between C- and D-vine copula models.
  → Different interpretations possible based on tree structures.

# Bibliography

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009).
Pair-copula constructions of multiple dependence.
*Insurance: Mathematics and Economics 44*(2), 182–198.

Clarke, K. A. (2007).
A simple distribution-free test for nonnested model selection.
*Political Analysis 15*(3), 347–363.

Czado, C., U. Schepsmeier, and A. Min (2011).
Maximum likelihood estimation of mixed C-vines with application to exchange rates.
*To appear in Statistical Modelling*.

Vuong, Q. H. (1989).
Ratio tests for model selection and non-nested hypotheses.
*Econometrica 57*(2), 307–333.

**Thank you very much for your attention!**

Visit: `http://cran.r-project.org/web/packages/CDVine/`