qad package

Magdalena Tovilo, Martin Resch

PLUS

May 23, 2022

Content

- What is qad?
- Advantages
- Properties of qad
- Background and implementation
- Functions
- Exercises

PARIS LODRON UNIVERSITÄT SALZBURG

What is qad?

- most common measures of dependence: Pearson's and Spearman's correlation coefficients
- Pearson's ρ measures linear dependence of two random variables X and Y via

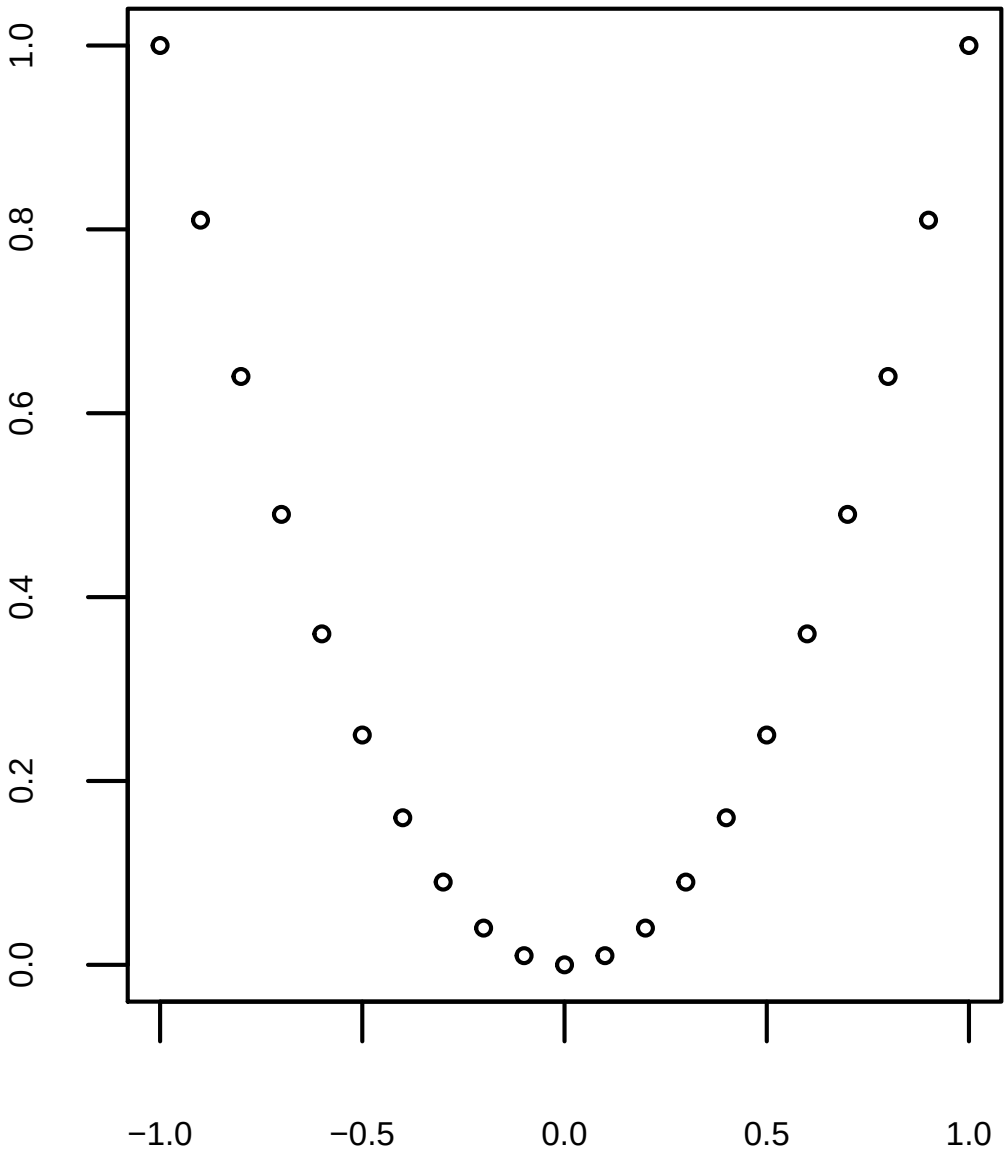
$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}}$$

- Spearman's r measures monotonic dependence and is calculated via the same formula BUT with ranks instead of the actual values of X and Y
- covariance is symmetric $\Rightarrow$ correlation coefficients are both symmetric, i.e.

$$\rho(X, Y) = \rho(Y, X) \text{ and } r(X, Y) = r(Y, X)$$

PARIS LODRON UNIVERSITÄT SALZBURG

What is qad?



PARIS LODRON UNIVERSITÄT SALZBURG

What is qad?

- $\rho(X, Y) = 1.2163073 \times 10^{-16} = \rho(Y, X)$
- $r(X, Y) = 0.041599 = r(Y, X)$

- qad package: provides another new (asymmetric!) dependence measure q_n
- the asymmetry gives the package its name
- qad is short for "quantification of asymmetric dependence"
- $q_n(X, Y) = 0.7232143$ and $q_n(Y, X) = 0.3490489$

PARIS LODRON UNIVERSITÄT SALZBURG

Advantages

- example illustrates the importance of introducing a directed dependency value (X has more influence on Y than vice versa)
- shows that while Pearson and Spearman fail (can only detect a certain type of dependence!), qad can detect any kind of dependence

PARIS LODRON UNIVERSITÄT SALZBURG

Properties of qad

- q_n is a strongly consistent estimator of a copula-based dependence measure q and therefore has the same properties for large enough sample size n

Properties of q:

- q can be calculated for all (continuous) random variables X and Y (without any knowledge of the distribution)
- $q(X, Y) \in [0, 1]$
- $q(X, Y) = 0 \Leftrightarrow X$ and Y are independent
- $q(X, Y) = 1 \Leftrightarrow Y$ is a function of X, i.e. $Y = f(X)$
- not necessarily: $q(X, Y) = q(Y, X)$ (asymmetry)
- $q(X, Y)$ is scale-invariant, i.e. $q(aX, aY) = C(a)q(X, Y)$ with $a, C(a) \in \mathbb{R}$ ($C(a)$ is a constant value dependent on a)

PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

Example:

	x_1	x_2	x_3	x_4
sample	2	7	5	3
ranks	1	4	3	2
norm. ranks	$\frac{1}{4}$	$\frac{4}{4}$	$\frac{3}{4}$	$\frac{2}{4}$
ecdf	$\frac{1}{4}$	$\frac{4}{4}$	$\frac{3}{4}$	$\frac{2}{4}$

$$\text{(ecdf) } \hat{F}_n(t) = \frac{\text{number of elements in the sample} \leq t}{n}.$$

PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

Copula

Copulas are functions which help analyze dependencies of random variables and are here used in the bivariate setting. Analytically a copula is a function $C : [0,1]^2 \to [0,1]$ with the following properties:

- $\forall x \in [0,1] : C(x,1) = C(1,x) = x, C(x,0) = C(0,x) = 0.$
- For $0 \leq x_1 \leq x_2 \leq 1$ and $0 \leq y_1 \leq y_2 \leq 1$ $C(x_2,y_2) - C(x_1,y_2) - C(x_2,y_1) + C(x_1,y_1) \geq 0$ holds.

Sklar's Theorem

For $(X,Y) \sim H$ with $X \sim F$ and $Y \sim G$ exists a copula C, so that for all $x, y \in \mathbb{R}$ the following equation holds:

$$H(x,y) = C(F(x), G(y)).$$

If F and G are continuous, the copula C is unique.

Background and implementation

Since qad works with samples, the underlying distribution functions of the data might not be known. Because of that, qad uses the so called empirical copula.

Empirical copula

Let $(x_k, y_k)_{k=1}^n$ denote a sample size of n from a continuous bivariate distribution. The empirical copula is the function C_n given by

$$C_n\left(\frac{i}{n}, \frac{j}{n}\right) = \frac{\text{number of pairs } (x, y) \text{ such that } x \leq x_{(i)} \text{ and } y \leq y_{(j)}}{n},$$

where $x_{(i)}$ and $y_{(j)}$, $1 \leq i, j \leq n$ denote the order statistics from the sample.

PARIS LODRON UNIVERSITÄT SALZBURG

Empirical copula frequency

The empirical copula frequency c_n is given by

$$c_n\left(\frac{i}{n}, \frac{j}{n}\right) = \begin{cases} \frac{1}{n}, & \text{if } (x_{(i)}, y_{(j)}) \text{ is an element of the sample} \\ 0, & \text{otherwise.} \end{cases}$$

Obviously it holds, that

$$C_n\left(\frac{i}{n}, \frac{j}{n}\right) = \sum_{p=1}^{i} \sum_{q=1}^{j} c_n\left(\frac{p}{n}, \frac{q}{n}\right).$$

PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

$$C_n\left(\frac{i}{n}, \frac{j}{n}\right) = \frac{\text{number of pairs } (x, y) \text{ such that } x \leq x_{(i)} \text{ and } y \leq y_{(j)}}{n}$$

i/j	x_i	y_j	$x_{(i)}$	$y_{(j)}$
1	0.95	0.24	0.19	0.16
2	0.53	0.16	0.32	0.24
3	0.77	0.56	0.53	0.33
4	0.19	0.33	0.77	0.56
5	0.32	0.80	0.95	0.80

Table: Sample and order statistics

$C_5(x, y)$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$	$\frac{5}{5}$
$\frac{1}{5}$	0	0	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$
$\frac{2}{5}$	0	0	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$
$\frac{3}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{3}{5}$
$\frac{4}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
$\frac{5}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$	$\frac{5}{5}$

Table: Empirical copula

Background and implementation

$$c_n\left(\frac{i}{n}, \frac{j}{n}\right) = \begin{cases} \frac{1}{n}, & \text{if } (x_{(i)}, y_{(j)}) \text{ is an element of the sample} \\ 0, & \text{otherwise.} \end{cases}$$

i/j	x_i	y_j	$x_{(i)}$	$y_{(j)}$
1	0.95	0.24	0.19	0.16
2	0.53	0.16	0.32	0.24
3	0.77	0.56	0.53	0.33
4	0.19	0.33	0.77	0.56
5	0.32	0.80	0.95	0.80

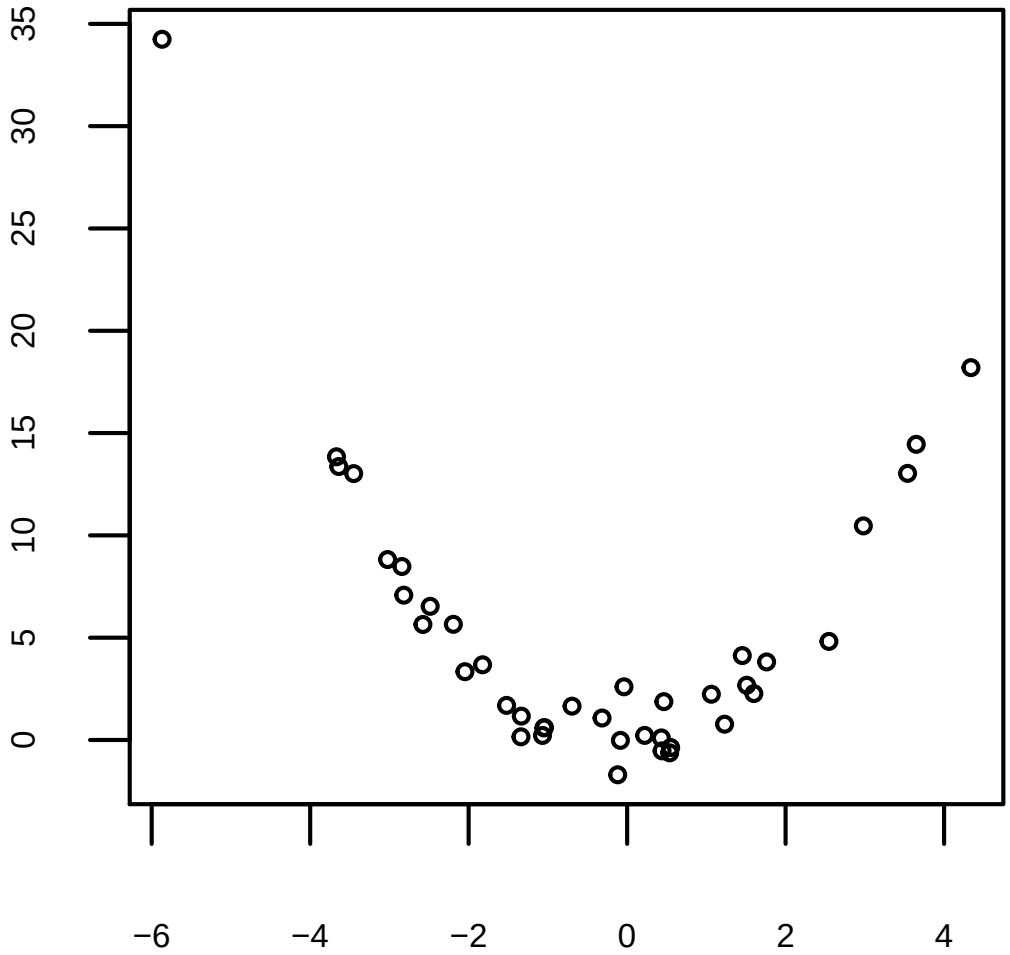
Table: Sample and order statistics

$c_5(x, y)$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$	$\frac{5}{5}$
$\frac{1}{5}$	0	0	$\frac{1}{5}$	0	0
$\frac{2}{5}$	0	0	0	0	$\frac{1}{5}$
$\frac{3}{5}$	$\frac{1}{5}$	0	0	0	0
$\frac{4}{5}$	0	0	0	$\frac{1}{5}$	0
$\frac{5}{5}$	0	$\frac{1}{5}$	0	0	0

Table: Empirical copula frequency

Background and implementation

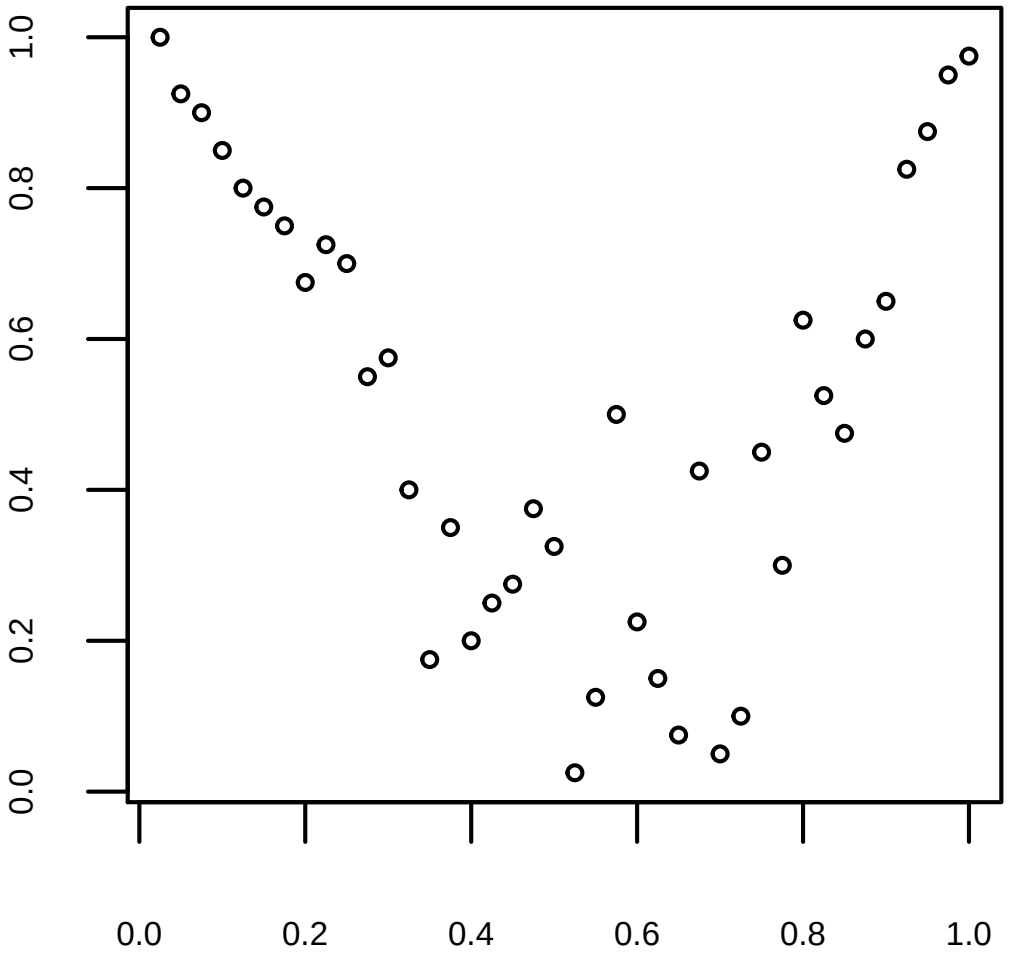
Draw a sample $(x_1, y_1), \ldots, (x_n, y_n)$ with $n = 40$ from the model $Y = X^2 + \varepsilon$ with $X \sim \mathcal{N}(0, 2)$ and $\varepsilon \sim \mathcal{N}(0, 1)$ pictured in the plot:



PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

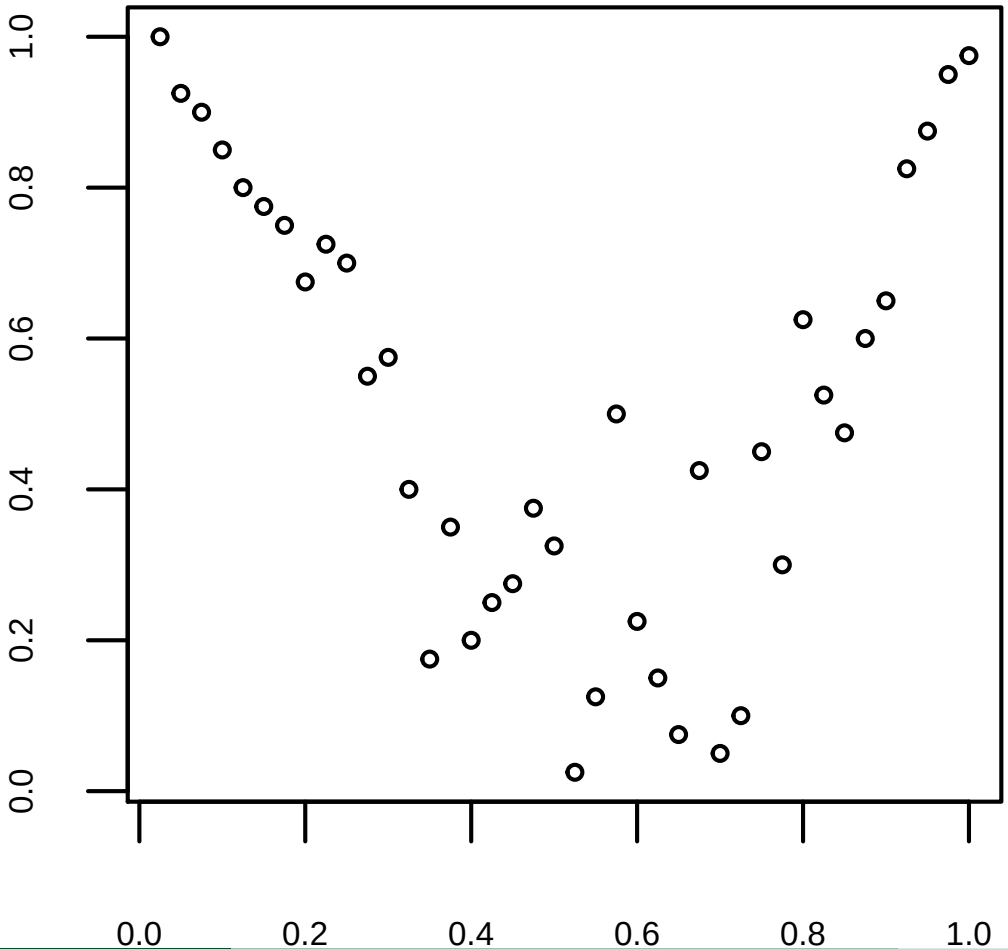
Plotting the normalized ranks of $x_1, \ldots, x_n$ against the normalized ranks of $y_1, \ldots, y_n$ yields:



PARIS LODRON UNIVERSITÄT SALZBURG

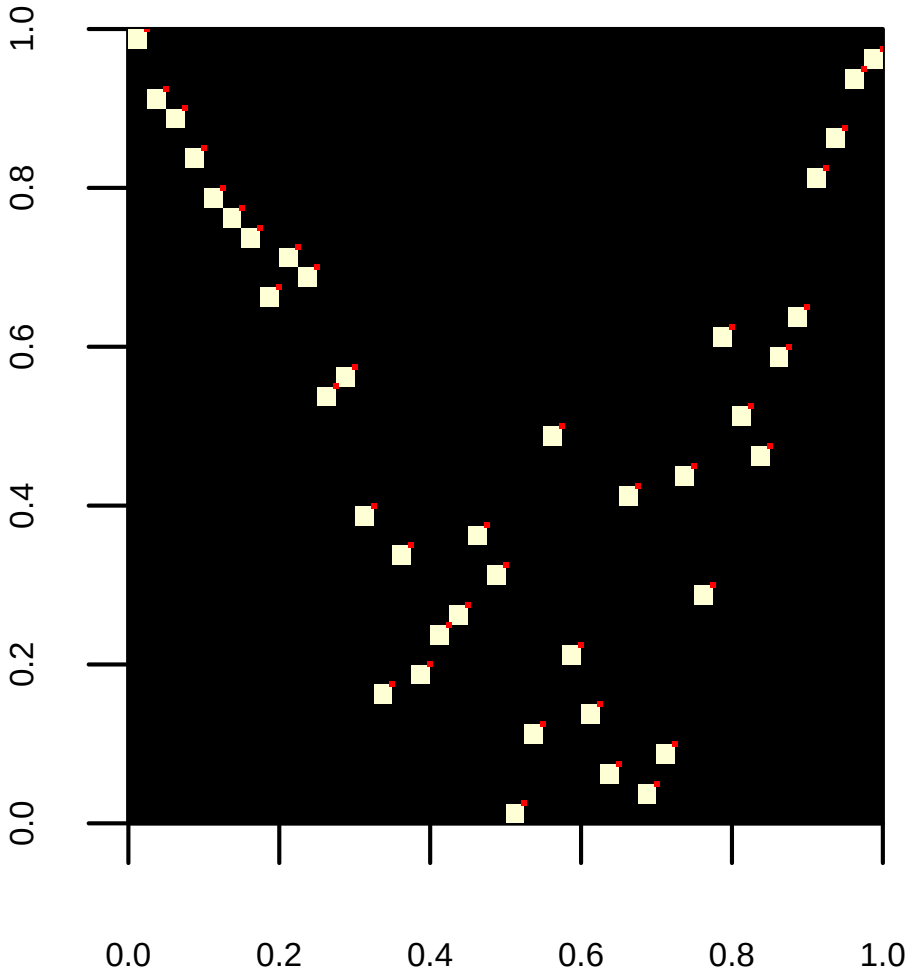
Background and implementation

The same result is given by plotting the values of the empirical cumulative distribution function (ecdf) of $x_1, \ldots, x_n$ $(y_1, \ldots, y_n)$ evaluated at $x_1, \ldots, x_n$ $(y_1, \ldots, y_n)$.



Background and implementation

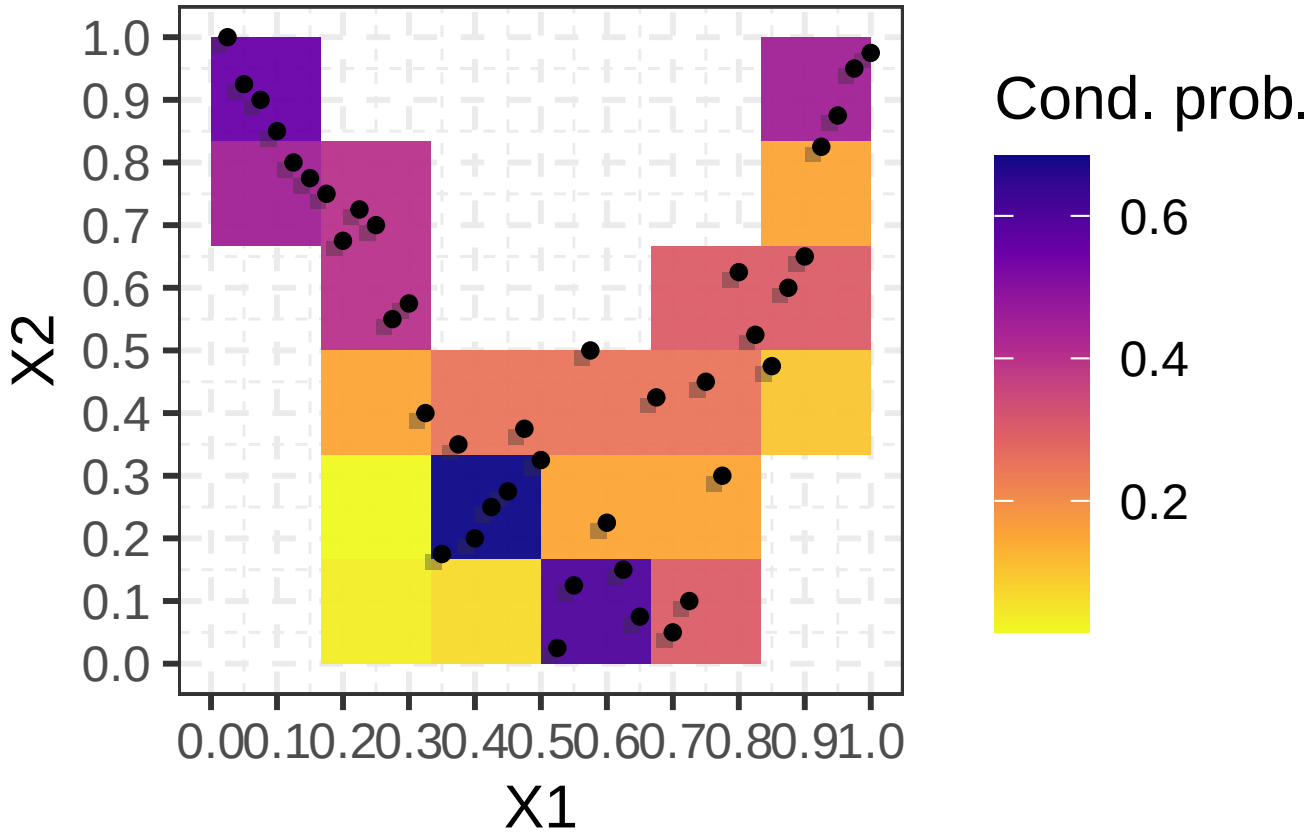
The empirical copula frequency plotted looks like:



PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

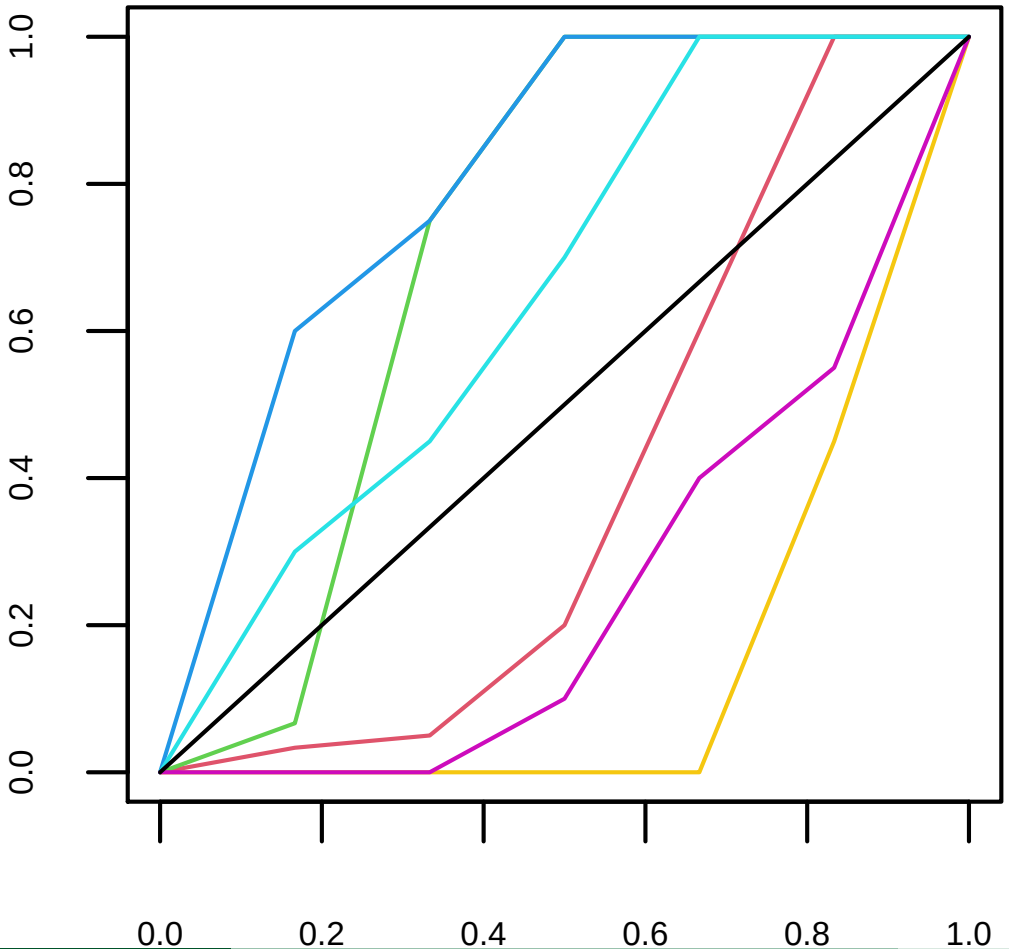
Qad plots the same image, but further divides $[0, 1]^2$ into a so-called checkerboard grid.



PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

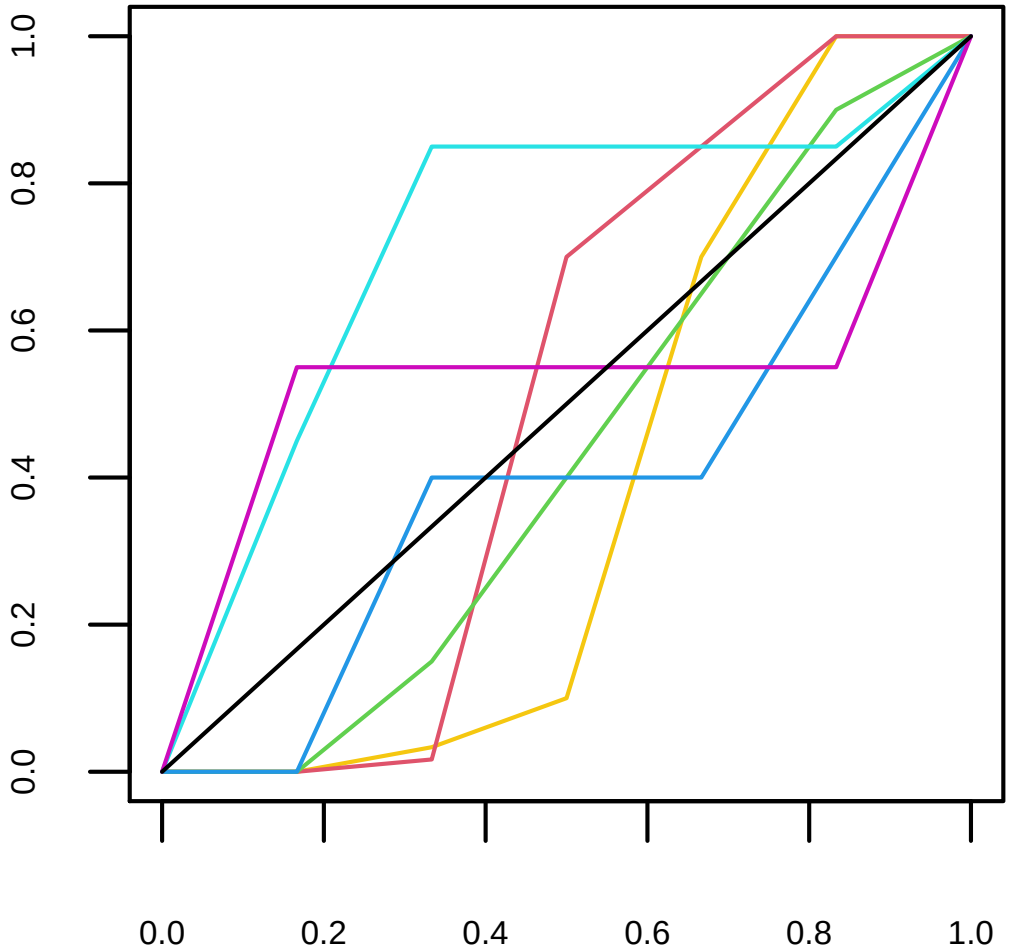
Vertical strips: The area between each function and the uniform distribution (depicted in black) is calculated and added up. This gives the value for $q_n(X, Y)$.



PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

The same is done for the horizontal strips returning the value of $q_n(Y, X)$.



PARIS LODRON UNIVERSITÄT SALZBURG

Background and implementation

The theory is based on continuous random variables with unique ranks. Nevertheless, qad gives reliable results for data with ties but the precision of the estimator may decrease.

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

The package "qad" contains 28 functions

- D1()
- D1.ECBC()
- zeta1()
- ECBC()
- ECBC.eval()
- bilinear_interpolation()
- build_checkerboard_weight()
- local_kernel_integral()
- D1_Pi()
- summary.qad()
- coef.qad()
- emp_c_copula()
- emp_c_copula_eval()
- plot.qad()
- plot_density()
- heatmap.qad()
- prepare_data_long()
- predict.qad()
- qad()
- qad.data.frame()
- qad.numeric()
- pairwise.qad()
- pqad()
- qqad()
- .r_local_kernel_integral_AB()
- .r_markov_kernel()
- .r_zeta1()
- .predict_qad_copula()

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

The package "qad" contains 28 functions

- D1()
- D1.ECBC()
- zeta1()
- ECBC()
- ECBC.eval()
- bilinear_interpolation()
- build_checkerboard_weight()
- local_kernel_integral()
- D1_Pi()
- summary.qad()
- coef.qad()
- emp_c_copula()
- emp_c_copula_eval()
- plot.qad()
- plot_density()
- heatmap.qad()
- prepare_data_long()
- predict.qad()
- qad()
- qad.data.frame()
- qad.numeric()
- pairwise.qad()
- pqad()
- qqad()
- .r_local_kernel_integral_AB()
- .r_markov_kernel()
- .r_zeta1()
- .predict_qad_copula()

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

These functions remain

- D1()
- D1.ECBC()
- zeta1()
- ECBC()
- ECBC.eval()
- summary.qad()
- coef.qad()
- plot.qad()
- heatmap.qad()
- predict.qad()
- qad()
- pairwise.qad()
- pqad()
- qqad()

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

We will take a look at these

- D1()
- D1.ECBC()
- zeta1()
- ECBC()
- ECBC.eval()
- summary.qad()
- coef.qad()
- plot.qad()
- heatmap.qad()
- predict.qad()
- qad()
- pairwise.qad()
- pqad()
- qqad()

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

$$
\begin{aligned}
&qad(x,\\
&\quad resolution = NULL,\\
&\quad p.value = TRUE, nperm = 1000,\\
&\quad p.value\_asymmetry = FALSE, nboot = 1000,\\
&\quad print = FALSE,\\
&\quad remove.00 = FALSE,\\
&\quad \ldots\\
&\quad)
\end{aligned}
$$

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

- **x:** data frame with two columns;
  alternatively a pair x, y of numeric vectors can be given
- **resolution:** number of strips in the checkerboard grid;
  default is set to NULL which gives $\lfloor n^{\frac{1}{2}} \rfloor$ where n is the minimum number of unique values
- **p.value:** a logical indicating whether to return a p-value of rejecting independence (based on permutation)
- **nperm:** an integer indicating the number of permutation runs (if p.value = TRUE)

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

- **p.value_asymmetry:** a logical indicating whether to return a p-value for the measure of asymmetry (based on bootstrap)
- **nboot:** an integer indicating the number of bootstrapping runs
- **print:** a logical indicating whether the result of qad is printed
- **remove.00:** a logical indicating whether double 0 entries should be excluded (default = FALSE)

PARIS LODRON UNIVERSITÄT SALZBURG

```
qad(sample, resolution = NULL,
  p.value = TRUE, nperm = 1000,
  p.value_asymmetry = TRUE, nboot = 1000,
  print = TRUE, remove.00 = FALSE)

##
## quantification of asymmetric dependence:
##
## Data: x1 := x
##       x2 := y
##
## Sample Size: 40
## Number of unique ranks: x1: 40
##                         x2: 40
##                     (x1,x2): 40
## Resolution: 6 x 6
##
## Dependence measures:
##                       q p.values
##  q(x1,x2)       0.720    0.000
##  q(x2,x1)       0.456    0.012
##  max.dependence 0.720    0.000
##
##                        a p.values
##  asymmetry       0.263    0.046
```

Functions

qad returns an object of class qad containing the following components:

- **data:** a data frame containing the input data
- **q(X,Y):** influence of X on Y
- **q(Y,X):** influence of Y on X
- **max.dependence:** maximal dependence
- **results:** a data frame containing the results
- **mass_matrix:** a matrix containing the mass distribution of the empirical checkerboard copula
- **resolution:** resolution of the checkerboard aggregation as an integer
- **n:** sample size

PARIS LODRON UNIVERSITÄT SALZBURG

```
fit <- qad(sample, print = FALSE)
fit$mass_matrix
```

```
##              [,1]        [,2]       [,3]       [,4]       [,5]       [,6]
## [1,] 0.000000000 0.000000000 0.00000000 0.00000000 0.07500000 0.09166667
## [2,] 0.005555556 0.002777778 0.02500000 0.06666667 0.06666667 0.00000000
## [3,] 0.011111111 0.113888889 0.04166667 0.00000000 0.00000000 0.00000000
## [4,] 0.100000000 0.025000000 0.04166667 0.00000000 0.00000000 0.00000000
## [5,] 0.050000000 0.025000000 0.04166667 0.05000000 0.00000000 0.00000000
## [6,] 0.000000000 0.000000000 0.01666667 0.05000000 0.02500000 0.07500000
```

```
fit$results
```

```
##                       coef p.values
## 1        q(x1,x2) 0.7198669    0.000
## 2        q(x2,x1) 0.4564470    0.006
## 3  max.dependence 0.7198669    0.000
## 4 asymmetry       0.2634198       NA
```

```
fit$resolution
```

```
## [1] 6
```

```
fit$max.dependence
```

```
## [1] 0.7198669
```

Functions

Functions that can be applied to a qad object are:

- **plot():** additional parameters are copula = TRUE\FALSE and addSample = TRUE\FALSE
- **summary():** prints the results
- **coef():** a named vector containing the results
- **predict()**

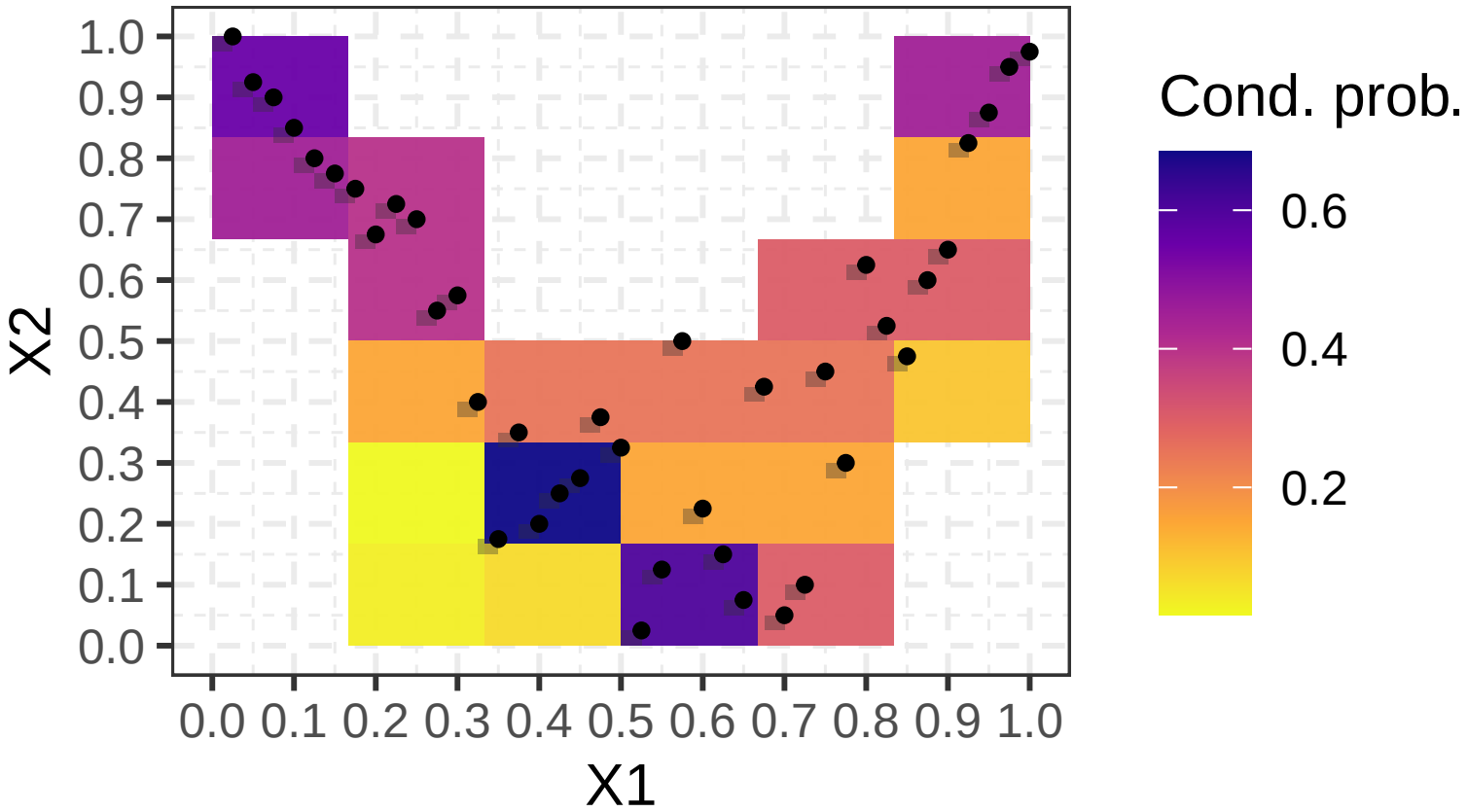
PARIS LODRON UNIVERSITÄT SALZBURG

Functions

$$
\begin{aligned}
&plot.qad(x,\\
&\qquad addSample = FALSE,\\
&\qquad copula = FALSE,\\
&\qquad density = FALSE,\\
&\qquad margins = FALSE, title = "",\\
&\qquad x.axis = "X1", y.axis = "X2",\\
&\qquad point.size = 0.9, panel.grid = TRUE,\\
&\qquad color = "plasma", rb\_values = c(10, 0.315, 0.15),\\
&\qquad \ldots\\
&\qquad)
\end{aligned}
$$

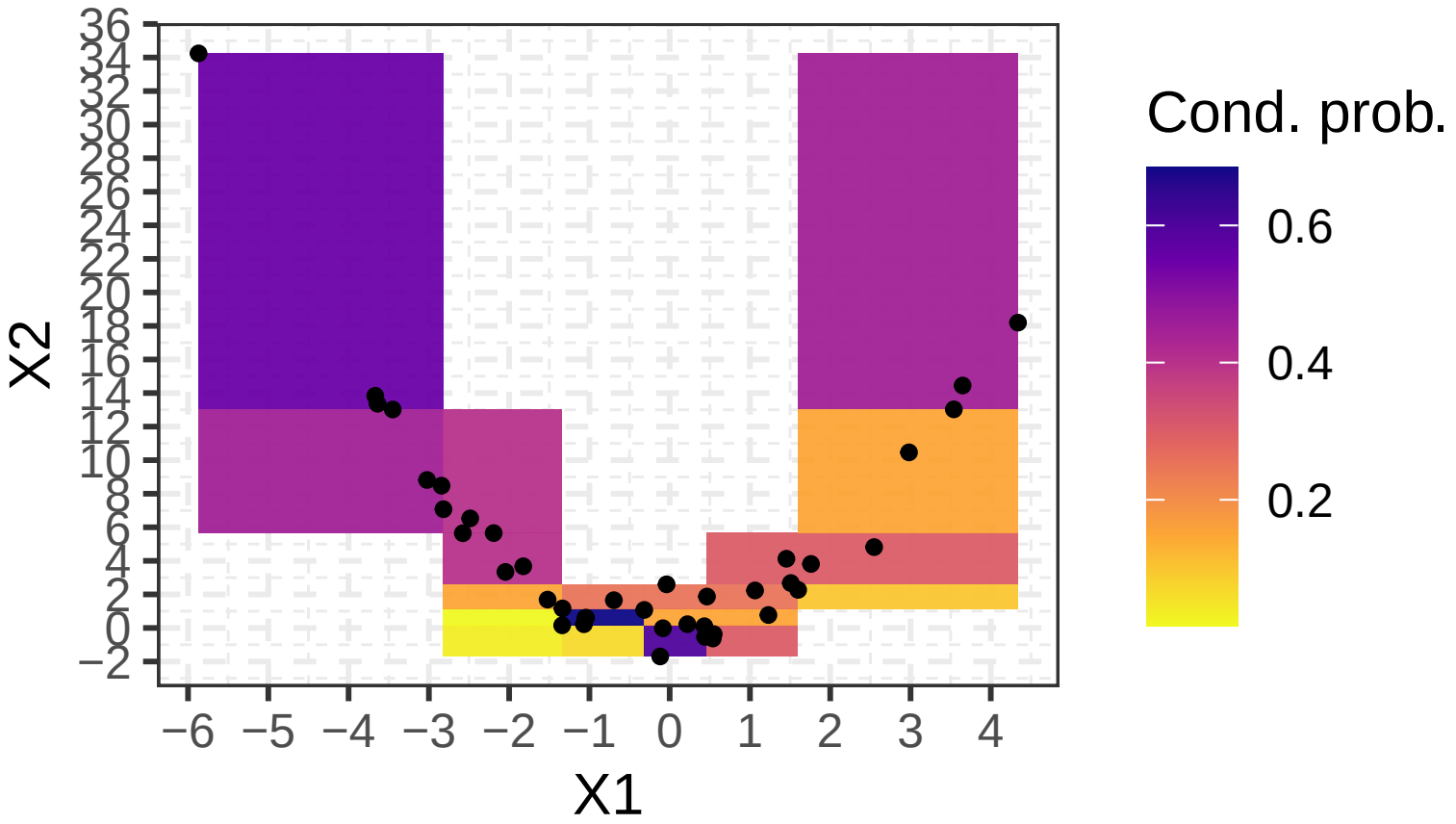
PARIS LODRON UNIVERSITÄT SALZBURG

```
fit <- qad(sample, print = FALSE)

plot(fit, copula = TRUE, addSample = TRUE)
```



PARIS LODRON UNIVERSITÄT SALZBURG

```
fit <- qad(sample, print = FALSE)

plot(fit, copula = FALSE, addSample = TRUE)
```



```r
fit <- qad(sample, print = FALSE)
summary(fit)
```

```
##
## quantification of asymmetric dependencies:
##
##
## Sample Size: 40
## Number of unique ranks: x1: 40
##                         x2: 40
##                     (x1,x2): 40
## Resolution: 6 x 6
##
## Dependence measures:
##                        q p.values
##  q(x1,x2)         0.7199    0.000
##  q(x2,x1)         0.4564    0.007
##  max.dependence 0.7199    0.000
##
##                        a p.values
##  asymmetry         0.263       NA
```

LODRON UNIVERSITÄT SALZBURG

```
fit <- qad(sample, print = FALSE)
coef(fit)

##         q(x1,x2)         q(x2,x1)   max.dependence         asymmetry
##        0.7198669        0.4564470        0.7198669        0.2634198
##       p.q(x1,x2)       p.q(x2,x1) p.max.dependence      p.asymmetry
##        0.0000000        0.0090000        0.0000000               NA

coef(fit)[2]

## q(x2,x1)
## 0.456447

coef(fit, select = c('q(x1,x2)','p.q(x1,x2)'))

##   q(x1,x2) p.q(x1,x2)
##  0.7198669  0.0000000
```

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

For the sake of completeness

$$summary.qad(object, \dots)$$

$$\begin{aligned}
&coef.qad(object,\\
&\quad select = c('q(x1, x2)', ' q(x2, x1)',\\
&\quad 'max.dependence', ' asymmetry', ' p.q(x1, x2)',\\
&\quad 'p.q(x2, x1)', ' p.max.dependence', ' p.asymmetry'),\\
&\quad \dots\\
&\quad)
\end{aligned}$$

Exercise 1

Create a datasample with $x \leftarrow rnorm(100, 0, 2)$ and a function of your choice as y dependent of x (not $y = x^2$).
Calculate the dependence measures and plot your result with the copula and the datasample.
Then display only the results for q and give a short interpretation.

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

$$
\begin{aligned}
&predict.qad(object,\\
&\qquad values,\\
&\qquad conditioned = "x1",\\
&\qquad nr\_intervals = NULL,\\
&\qquad prediction\_interval = NULL,\\
&\qquad copula = FALSE,\\
&\qquad pred\_plot = FALSE,\\
&\qquad panel.grid = TRUE,\\
&\qquad \ldots\\
&\qquad)
\end{aligned}
$$

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

- **object:** an object of class 'qad', as reference for the prediction
- **values:** a vector containing the x or y values for which the probabilities should be predicted
- **conditioned:** a character specifying on which variable is conditioned ("x1" or "x2")
- **nr_intervals:** an integer, which determines a different number of intervals for the prediction (only possible in the copula setting)

PARIS LODRON UNIVERSITÄT SALZBURG

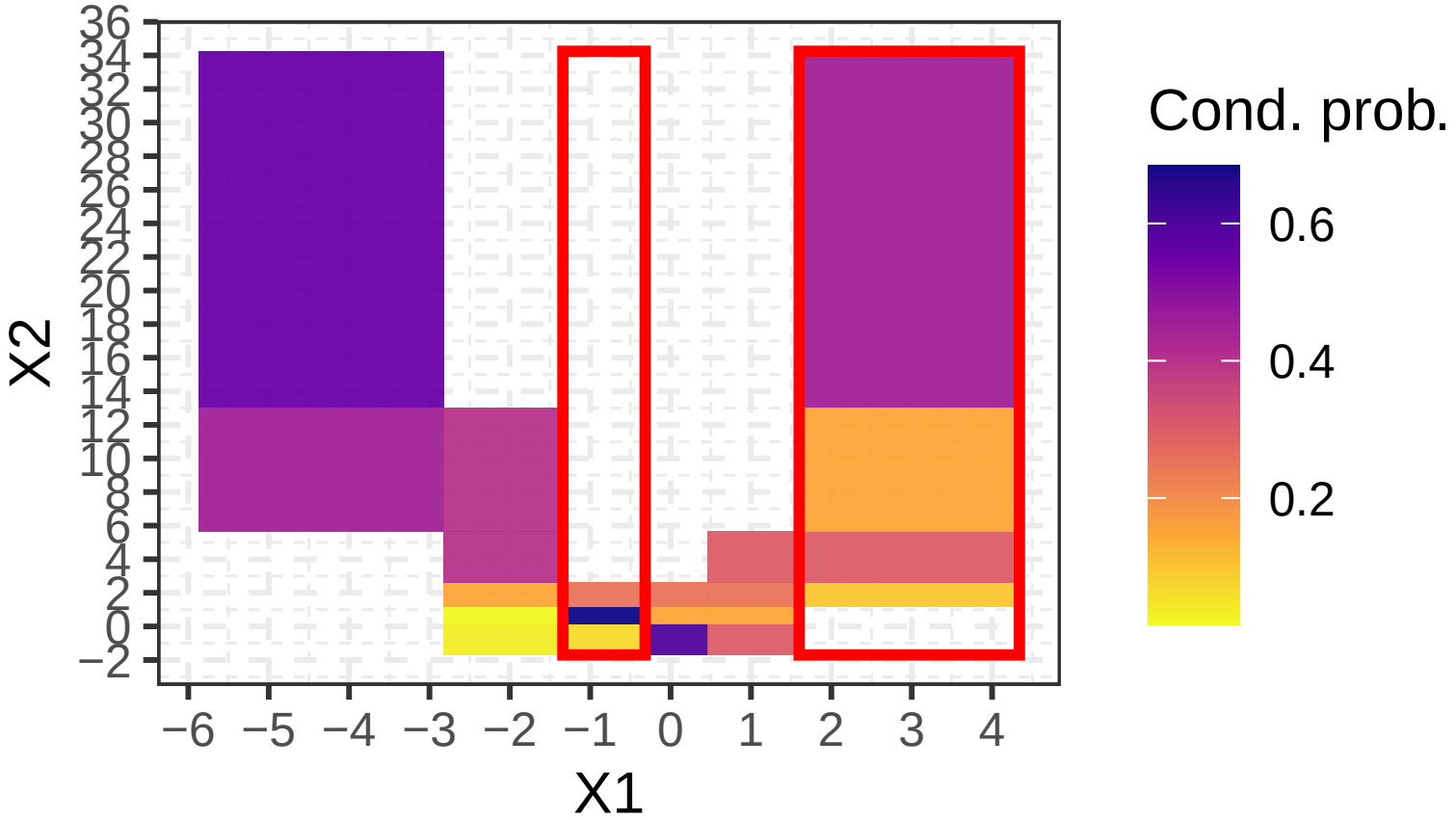
Functions

- **prediction_interval:** if NULL, predictions are made for the intervalls described by the resolution seperators, otherwise a vector c(lower_boundary, upper_boundary) has to be specified
- **copula:** a logical determining whether the empirical checkerboard copula is used or the retransformed data
- **pred_plot:** a logical indicating if the conditional probabilites are plotted
- **panel.grid:** a logical indicating whether the panel.grid is plotted.

PARIS LODRON UNIVERSITÄT SALZBURG

```r
fit <- qad(sample, print = FALSE)
predict.qad(fit, values = c(-0.5,2), conditioned = "x1")

## Intervall definition:
##   Interval lowerBound upperBound
## 1       I1 -1.6996562  0.1555202
## 2       I2  0.1555202  1.1640893
## 3       I3  1.1640893  2.5999350
## 4       I4  2.5999350  5.6455520
## 5       I5  5.6455520 13.0200838
## 6       I6 13.0200838 34.2460431
##
## Prediction probabilities:
##                  I1           I2   I3  I4   I5   I6
## x1=-0.5 0.06666667 6.833333e-01 0.25 0.0 0.00 0.00
## x1=2    0.00000000 3.330669e-17 0.10 0.3 0.15 0.45
```

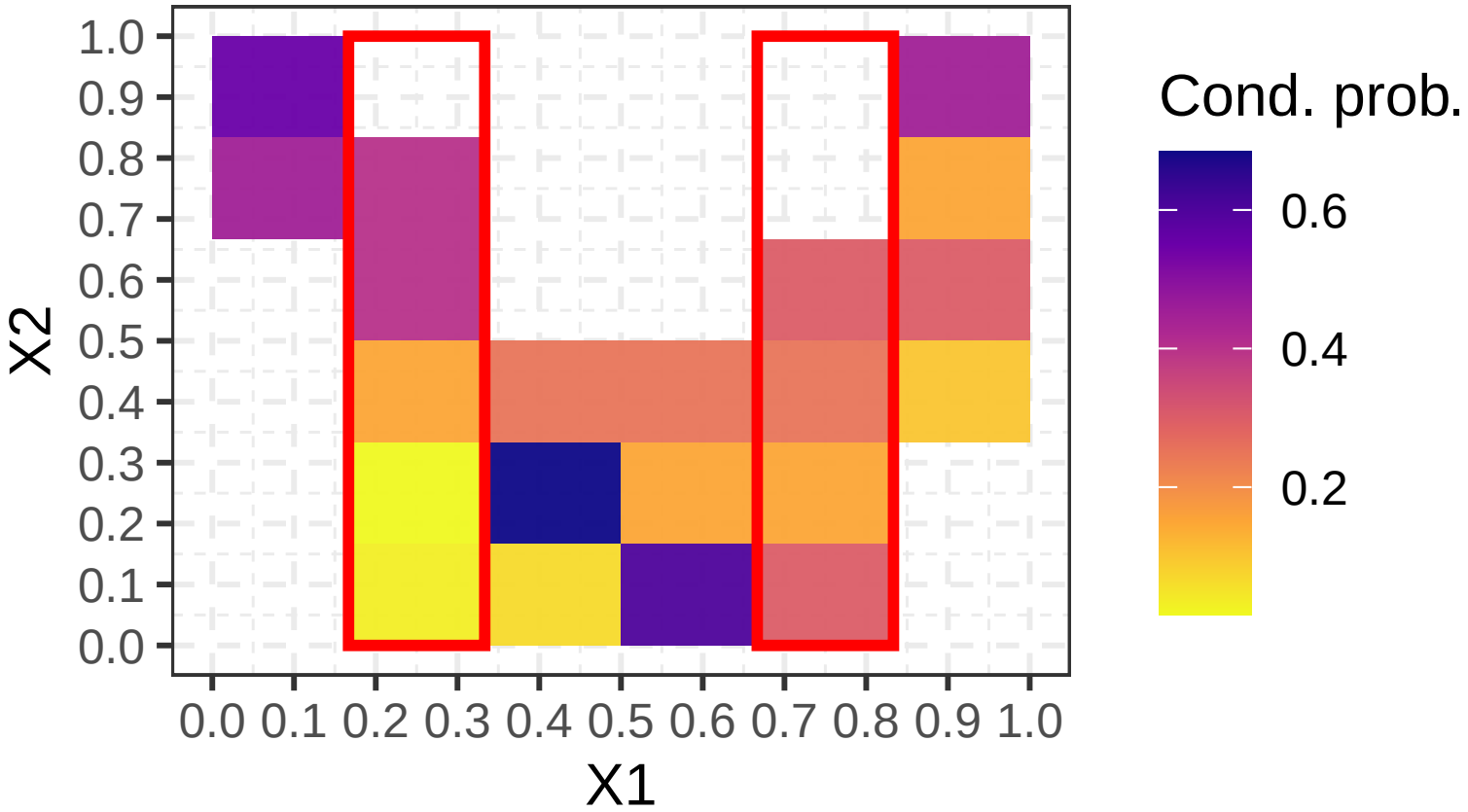
PARIS LODRON UNIVERSITÄT SALZBURG



PARIS LODRON UNIVERSITÄT SALZBURG

```
fit <- qad(sample, print = FALSE)
pred <- predict.qad(fit, values = c(0.3,0.8), nr_intervals = 4, copula = TRUE)

## Intervall definition:
##   Interval lowerBound upperBound
## 1       I1       0.00       0.25
## 2       I2       0.25       0.50
## 3       I3       0.50       0.75
## 4       I4       0.75       1.00
##
## Prediction probabilities:
##                 I1        I2  I3  I4
## x1=0.3 0.04166667 0.1583333 0.6 0.2
## x1=0.8 0.37500000 0.3250000 0.3 0.0
```

PARIS LODRON UNIVERSITÄT SALZBURG



PARIS LODRON UNIVERSITÄT SALZBURG

Exercise 2

Based upon the given dataset of x and y predict and plot for the values $y = 0.15$ and $y = 0.65$ the probabilities that x is in one of five intervalls of length 0.2.

$x \leftarrow rnorm(100, 0, 2)$

$y \leftarrow cos(x) + rnorm(100, 0, 0.1)$

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

$pqad(q, n, R = 1000, resolution = NULL)$
... calculates the value of the distribution function of qad at a given quantile. The calculation of qad is performed R times on independent datasamples with size n. Essentially it calculates $p = P(qad \leq q)$ under the assumption of independence.

$qqad(p, n, R = 1000, resolution = NULL)$
... is the quasi inverse function of pqad, calculating the quantile for a given probability, i.e. $F(q) = P(qad \leq q) = p \Leftrightarrow q = F^{-}(p)$

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

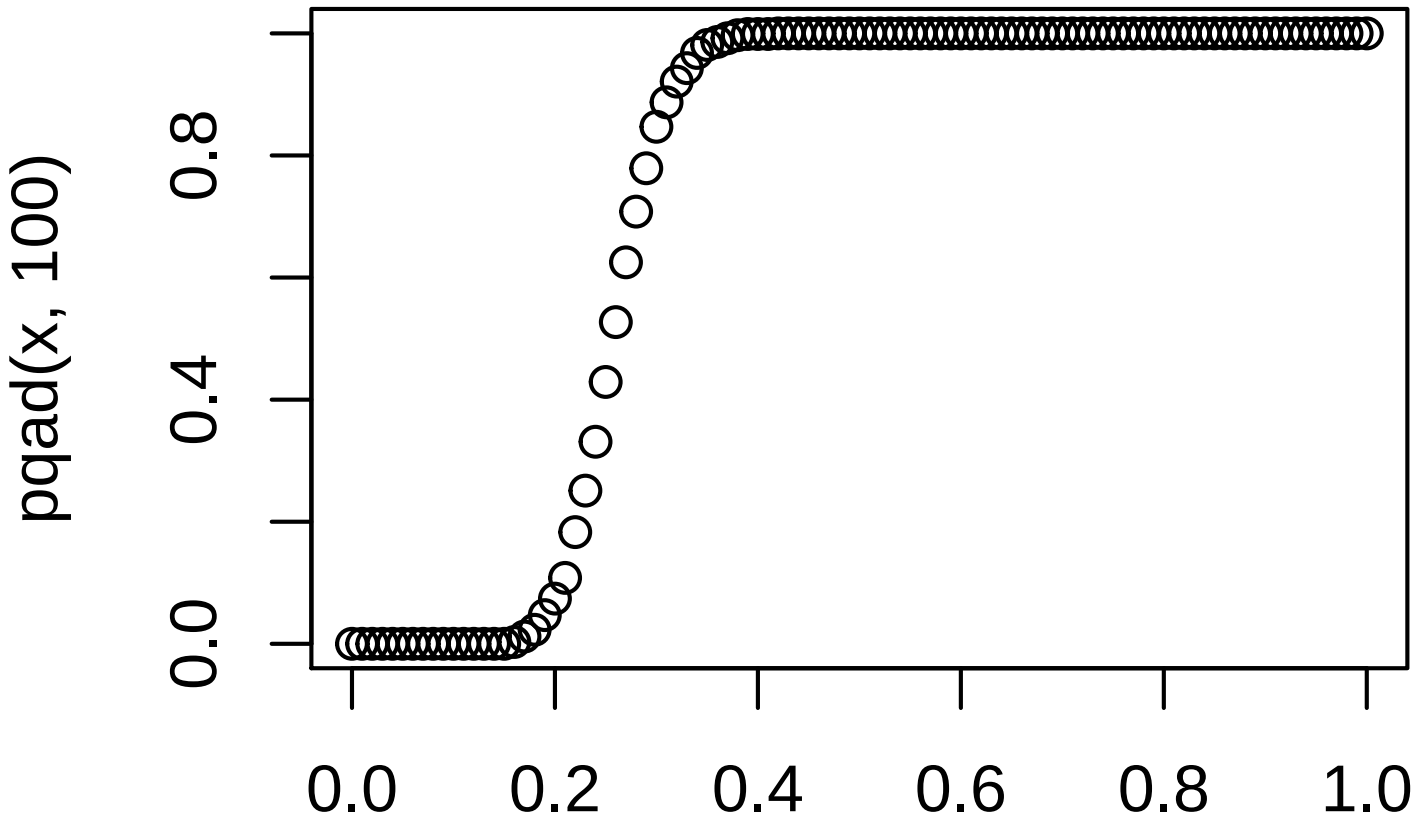
- **p:** vector of probabilities
- **q:** vector of quantiles
- **n:** size of sample used in calculation
- **R:** number of repetitions
- **resolution:** resolution of checkerboard copula

PARIS LODRON UNIVERSITÄT SALZBURG

```
pqad(0.25, 100)

## [1] 0.424

p <- pqad(0.25, 100)
qqad(p, 100)

## 42.4%
## 0.247

qqad(0.5, 100)

##    50%
## 0.2575
```

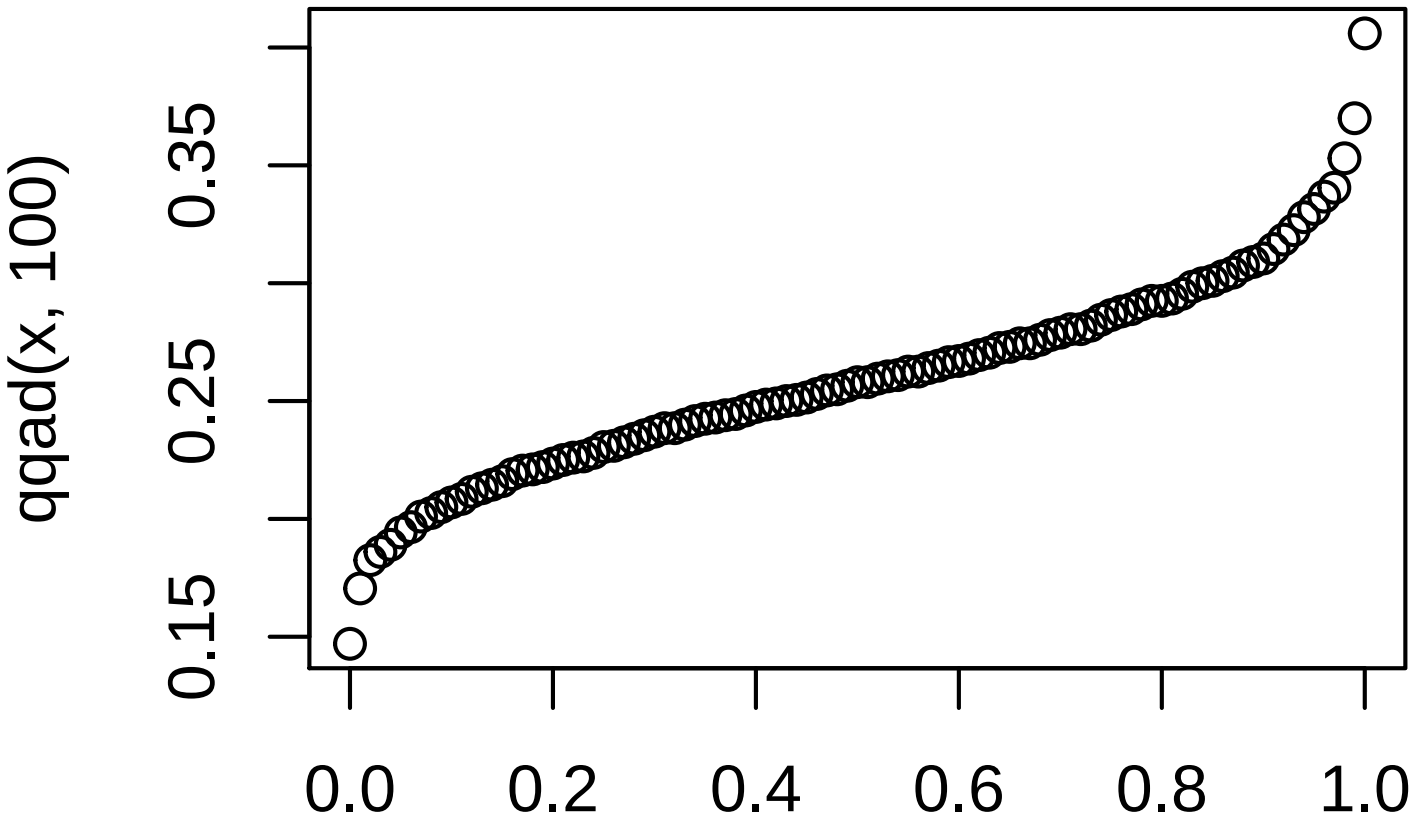
PARIS LODRON UNIVERSITÄT SALZBURG

```r
x <- seq(0, 1, by = 0.01)
plot(x, pqad(x, 100))
```



PARIS LODRON UNIVERSITÄT SALZBURG

```r
x <- seq(0, 1, by = 0.01)
plot(x, qqad(x, 100))
```



PARIS LODRON UNIVERSITÄT SALZBURG

Exercise 3

Plot the distribution function of qad for n=1000 (instead of n=100 as we have seen just now).
Calculate the smallest value at which the distribution function obtains 1 and compare it to the result from before. What does this mean?

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

$$
\begin{aligned}
&pairwise.qad(data\_df,\\
&\qquad remove.00 = FALSE,\\
&\qquad min.res = 3,\\
&\qquad p.value = TRUE,\\
&\qquad nperm = 1000,\\
&\qquad p.value\_asymmetry = FALSE,\\
&\qquad nboot = 1000\\
&\qquad)
\end{aligned}
$$

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

- **data_df:** a data frame containing numeric columns with the observations of the sample
- **remove.00:** a logical indicating whether double 0 entries should be excluded (default = FALSE)
- **min.res:** an integer indicating the necessary minimum resolution of the checkerboard grid to compute qad, otherwise the result is NA
- **p.value:** a logical indicating whether to return a p-value of rejecting independence (based on permutation)

PARIS LODRON UNIVERSITÄT SALZBURG

Functions

- **nperm:** an integer indicating the number of permutation runs
- **p.value_asymmetry:** a logical indicating whether a p-value (based on bootstrap) is computed for the measure of asymmetry
- **nboot:** an integer indicating the number of bootstrapping runs

PARIS LODRON UNIVERSITÄT SALZBURG

```r
n <- 40
x1 <- runif(n, 0, 1)
x2 <- x1^2 + rnorm(n, 0, 0.1)
x3 <- rnorm(n, 0, 1)
x4 <- x3 + rnorm(n, 0, 0.1)
sample <- data.frame(x1,x2,x3,x4)
```

```r
pair <- pairwise.qad(sample)
```

```r
pair$q
```

```
##           x1        x2        x3        x4
## x1        NA 0.7478155 0.1911869 0.1961458
## x2 0.7713274        NA 0.2566667 0.2573958
## x3 0.2316491 0.2573214        NA 0.8541715
## x4 0.2400298 0.2502083 0.8541715        NA
```

```r
pair$max.dependence
```

```
##           x1        x2        x3        x4
## x1        NA 0.7713274 0.2316491 0.2400298
## x2 0.7713274        NA 0.2573214 0.2573958
## x3 0.2316491 0.2573214        NA 0.8541715
## x4 0.2400298 0.2573958 0.8541715        NA
```

```
pair$asymmetry

##            x1             x2             x3          x4
## x1         NA -0.0235119048 -0.0404622114 -0.04388393
## x2 0.02351190             NA -0.0006547619  0.00718750
## x3 0.04046221  0.0006547619             NA  0.00000000
## x4 0.04388393 -0.0071875000  0.0000000000          NA

pair$q_p.values

##       x1    x2    x3    x4
## x1    NA 0.000 0.956 0.943
## x2 0.000    NA 0.684 0.679
## x3 0.809 0.681    NA 0.000
## x4 0.793 0.709 0.000    NA

pair$max.dependence_p.values

##       x1    x2    x3    x4
## x1    NA 0.000 0.933 0.915
## x2 0.000    NA 0.846 0.837
## x3 0.933 0.846    NA 0.000
## x4 0.915 0.837 0.000    NA
```

SALZBURG

```
pair$asymmetry_p.values

##    x1 x2 x3 x4
## x1 NA NA NA NA
## x2 NA NA NA NA
## x3 NA NA NA NA
## x4 NA NA NA NA

pair$resolution

##    x1 x2 x3 x4
## x1 NA  6  6  6
## x2  6 NA  6  6
## x3  6  6 NA  6
## x4  6  6  6 NA

pair$n_removed_00

##    x1 x2 x3 x4
## x1 NA NA NA NA
## x2 NA NA NA NA
## x3 NA NA NA NA
## x4 NA NA NA NA
```

Functions

```
heatmap.qad(pw_qad,
            select = c('dependence','max.dependence','asymmetry'),
            fontsize = 4,
            significance = FALSE,
            sign.level = 0.05,
            scale = "abs",
            color = "plasma",
            rb_values = c(10, 0.315, 0.15),
            title = ""
            )
```

Functions

- **pw_qad:** output of the function pairwise.qad()
- **select:** a character indicating which dependence value is plotted. Options are c("dependence", "max.dependence", "asymmetry")
- **fontsize:** a numeric specifying the font size of the values
- **significance:** a logical indicating whether significant values (with respect to the qad p.values) are denoted by a star
- **sign.level:** numeric value indicating the significance level

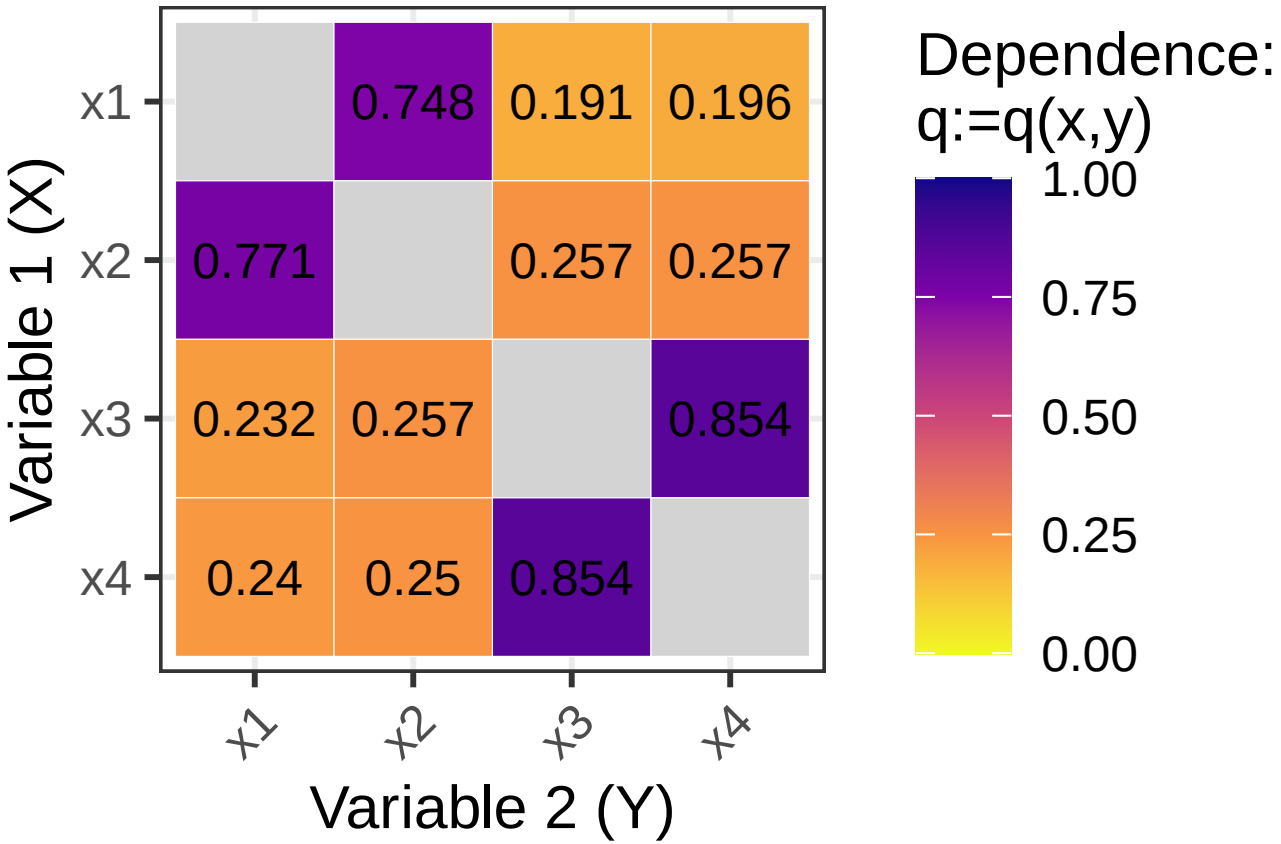
PARIS LODRON UNIVERSITÄT SALZBURG

Functions

- **scale:** character indicating whether the heatmap uses a relative or absolute scale. Options are "rel" or "abs" (default)
- **color:** Select the color palette. Options are c("plasma", "viridis", "inferno", "magma", "cividis")
- **rb_values:** a vector of size 3 with number of values, start value and end value in the rainbow colors space
- **title:** text for the title

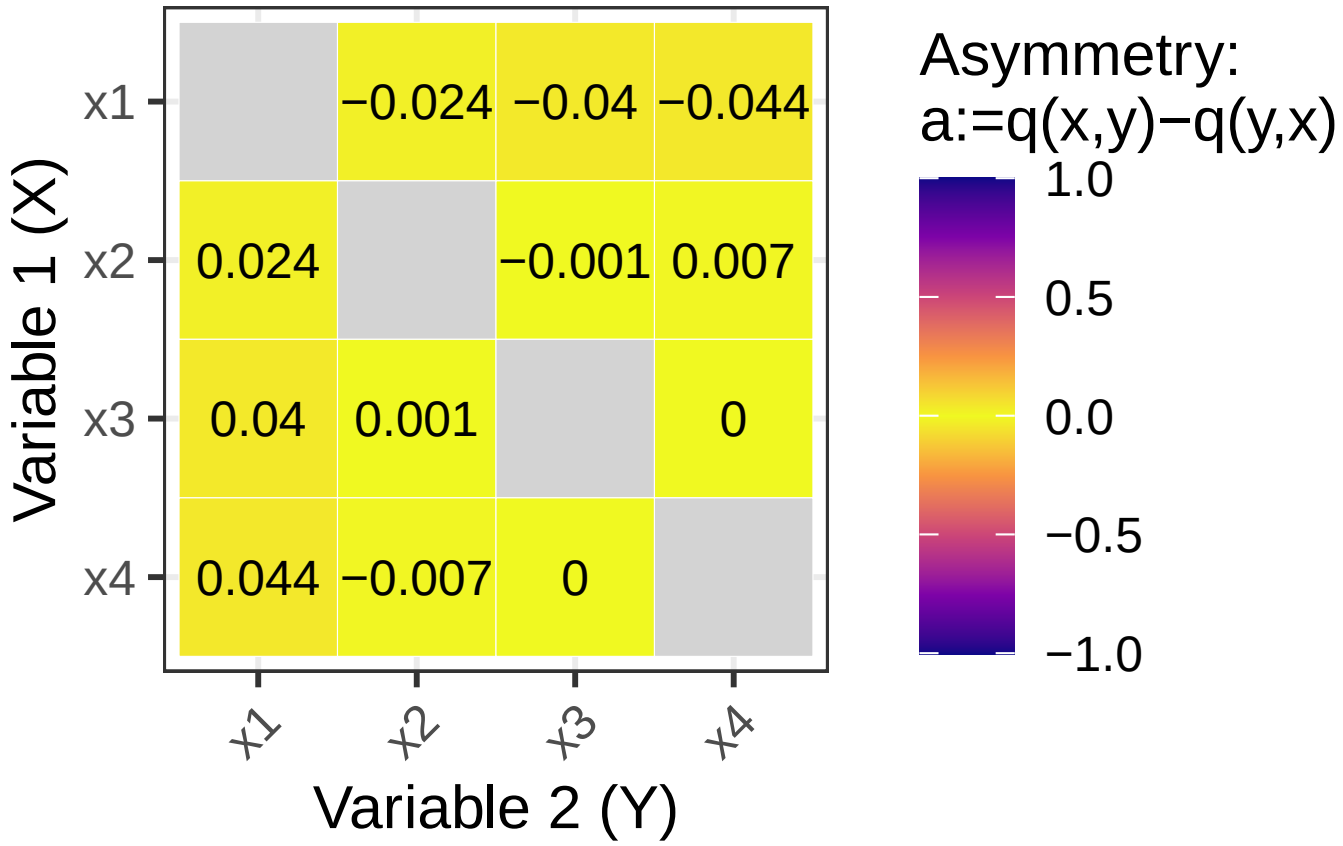
PARIS LODRON UNIVERSITÄT SALZBURG

```
heatmap.qad(pair, select = "dependence", fontsize = 3)
```



PARIS LODRON UNIVERSITÄT SALZBURG

```
heatmap.qad(pair, select = "asymmetry", fontsize = 3)
```



Exercise 4

Take the iris data set below where the column of the species was removed.
With the help of pairwise.qad display only the values of q.
Then plot the results in a heatmap.

PARIS LODRON UNIVERSITÄT SALZBURG

Sources

- http://www.trutschnig.net/software.html
- https://trutschnig.net/statistik_intro.pdf
- Nelsen, R. B. (1999): An introduction to Copulas, New York: Springer Science + Business Media New York
- https://www.deep-mind.org/2017/09/24/empirical_copula/
- https://github.com/griefl/qad
- https://www.rdocumentation.org/packages/qad/versions/1.0.1/

PARIS LODRON UNIVERSITÄT SALZBURG