Grid-based computing over joint probability distribution

FSTA 2024

Tomáš Bacigál

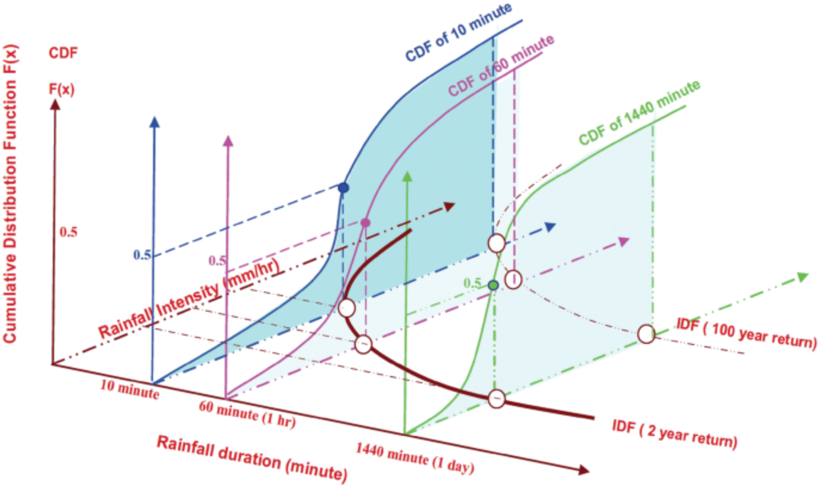
Department of Mathematics and Descriptive Geometry
Faculty of Civil Engineering
Slovak University of Technology in Bratislava
Slovakia

https://www.math.sk/bacigal

2024-02-01

Motivation

The initial impulse came from collaboration with hydrologists...



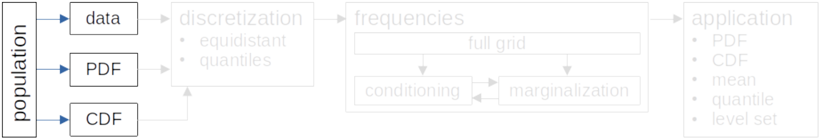
source: Sun et al. 2019: Deriving intensity–duration–frequency (IDF) curves ...

Motivation

When applying models of probability distribution we face up to:

- numerous classes and construction methods for mathematical models of joint probability distribution (multivariate extensions, decomposition to copula and marginals, nonparametric methods ...)
- PDF and CDF are rarely defined both at once in closed or computationally convenient form, sometimes just an effective random generator is available
- software implementations are *incomplete*, outdated and scattered across packages / products
  - missing application-related functions
  - models specialized either to categorical or continuous case
  - dimensionality limitations
- demand for unifying approach, simple and fast solution

Population distribution



- Sometimes we have plenty of observations, thus no parametric model is needed.
- But usually we need assumptions to create a useful representation of population.
- Some parametric models are expressed by PDF, some by CDF, others may be implicit yet easily providing random data.
- One of these three representations of probability distribution serves as input for the proposed grid-based approach to modeling and inference.

Population distribution

To *illustrate* the matter consider a probability distribution given by

$$F(x_1, x_2, x_3) = \Phi_{\mathbf{R}}\Big(\Phi^{-1}[F_{N(0,1)}(x_1)], \Phi^{-1}[F_{Exp(1)}(x_2)], \Phi^{-1}[F_{U(0,1)}(x_3)]\Big)$$

- F - joint (trivariate) cumulative distribution function (CDF),
- $\Phi_{\mathbf{R}}$ - trivariate CDF of standard normal distribution with correlation matrix

$$\mathbf{R} = \begin{pmatrix} 1 & 0.8 & 0 \\ 0.8 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Φ^{-1} - quantile function of univariate standard normal
- F_D - CDF of univariate distribution D

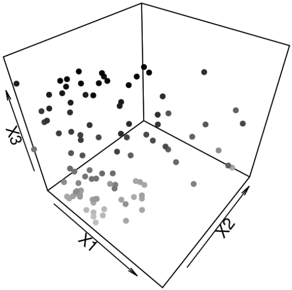
Population distribution

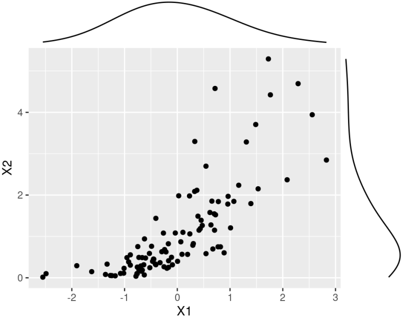
Example of specification and generating algorithm in R:

```
set.seed(1234)                              # random generator seed
dat <- c(0.8, 0, 0) |>                      # correlation coefficients
  copula::normalCopula(                     # dependence
    dim = 3,                                  # dimension
    dispstr = "un"                            # correlation structure
    ) |>
  copula::mvdc(                             # joint distribution
    margins = c("norm", "exp", "unif"),       # marginals families
    paramMargins = list(                      # marginals parameters
      list(mean = 0, sd = 1),                   # normal
      list(rate = 1),                           # exponential
      list(min = 0, max = 1))                   # uniform
  ) |>
  copula::rMvdc(n = 100)                    # generate random triplets
```

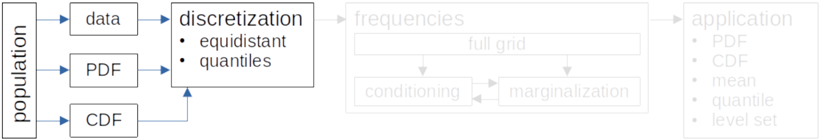
Population distribution

The outcome is 100 random triplets:





Discretization



Generated data enter the first stage, where

- values of continuous variables are categorized into *bins*,
- *breaks* can be chosen either as
  - equidistant (as it is usual in histogram) by number of bins, or
  - quantiles corresponding to given probabilities,
- every cell of the grid is characterized by,
  - *midpoint* (center of breaks) and
  - *size* (difference between breaks).

Discretization

Breaks (grid cells vertices, b) may be set as

- equidistant

$$b_{i\cdot\cdot} - b_{(i-1)\cdot\cdot} = \frac{\max(X_1) - \min(X_1)}{N_1}, \qquad i = 1, \dots, N_1$$

$$b_{\cdot j\cdot} - b_{\cdot(j-1)\cdot} = \frac{\max(X_2) - \min(X_2)}{N_2}, \qquad j = 1, \dots, N_2$$

$$b_{\cdot\cdot k} - b_{\cdot\cdot(k-1)} = \frac{\max(X_3) - \min(X_3)}{N_3}, \qquad k = 1, \dots, N_3$$

  with $b_{0\cdot\cdot} = \min(X_1)$, $b_{N_1\cdot\cdot} = \max(X_1)$, $b_{\cdot 0\cdot} = \min(X_2)$, ...

- quantiles for equidistant probabilities

$$Pr(b_{(i-1)\cdot\cdot} < X_1 \leq b_{i\cdot\cdot}) = \frac{1}{N_1}, \qquad i = 1, \dots, N_1$$

...

Discretization

Midpoints (grid cells centers, c) may be

- simple (average)

$$c_{i\cdot\cdot} = \frac{b_{i\cdot\cdot} + b_{(i-1)\cdot\cdot}}{2}$$

- probabilistic (local median)

$$Pr(b_{(i-1)\cdot\cdot} < X_1 \leq c_{i\cdot\cdot}) = Pr(c_{i\cdot\cdot} < X_1 \leq b_{i\cdot\cdot})$$

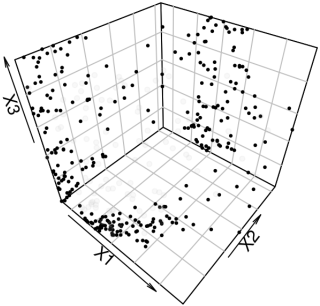
Differences (grid cells sizes, d) are defined as

$$d_{i\cdot\cdot} = b_{i\cdot\cdot} - b_{(i-1)\cdot\cdot}$$

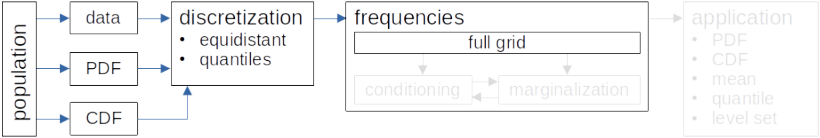
Discretization

```
breaks_eq_dat <- dat |>
  make_brea_data(bins = c(4, 3, 2))  # equidistant breaks

breaks_pr_dat <- dat |>
  make_brea_data(probs = list(       # equidistant probabilities
    seq(0, 1, by = 0.25),              # 4 bins
    seq(0, 1, length.out = 3+1),       # 3 bins
    seq(0, 1, by = 0.5))               # 2 bins
  )

mids_breaks_pr_dat <- make_mid_brea(breaks_pr_dat)  # simple midpoints
midp_breaks_pr_dat <- make_mid_brea(                # local medians
  breaks_pr_dat,
  probabilistic = TRUE,
  data = dat)

diff_breaks_pr_dat <- lapply(breaks_pr_dat, diff)   # differences
```

Discretization



Frequency



The second stage starts with

- counting absolute frequency in cells,
- making a one full frequency table in long format.

Frequency

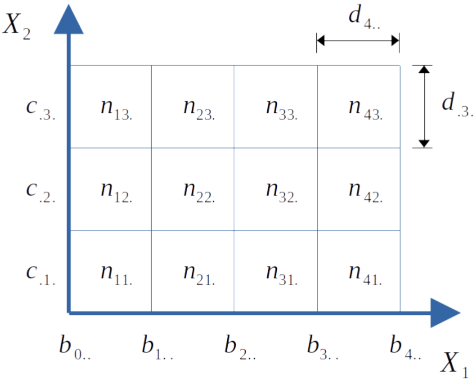
- Absolute frequency is defined as

$$n_{ijk} = \sum_{x_1} \sum_{x_2} \sum_{x_3} 1_{(b_{(i-1)\cdot\cdot}, b_{i\cdot\cdot}]}(x_1)\, 1_{(b_{\cdot(j-1)\cdot}, b_{\cdot j\cdot}]}(x_2)\, 1_{(b_{\cdot\cdot(k-1)}, b_{\cdot\cdot k}]}(x_3)$$

  where $1_{\mathcal{J}}(x) = 1$ if $x \in \mathcal{J}$ and 0 otherwise.
- Total number of observations will be simply

$$n = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} n_{ijk}$$

Frequency



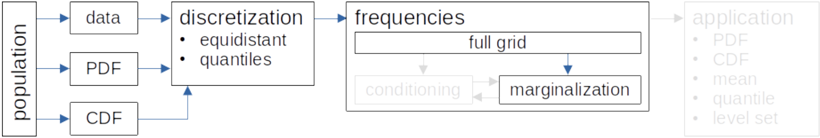
notation

X1	X2	X3	val
1	1	1	14
2	1	1	6
3	1	1	1
4	1	1	0
1	2	1	1
2	2	1	5
.	.	.	.

grid table

```
frequency_pr_dat <- make_freq_data(dat, breaks = breaks_pr_dat)
```

Marginalization



The second stage contains some optional steps. One of them is reduction of full grid (distribution) into a margin.

Marginalization is simply a summation over unwanted variables

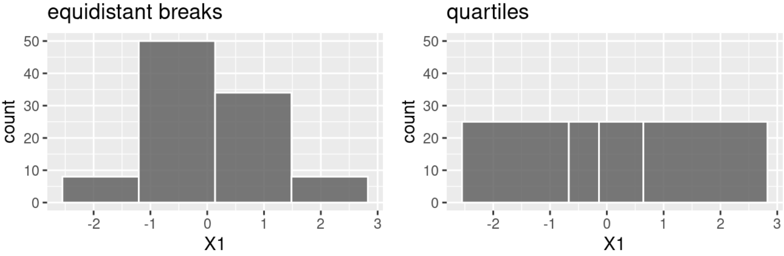
$$n_{i\cdot\cdot} = \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} n_{ijk}$$

$$n_{ij\cdot} = \sum_{k=1}^{N_3} n_{ijk}$$

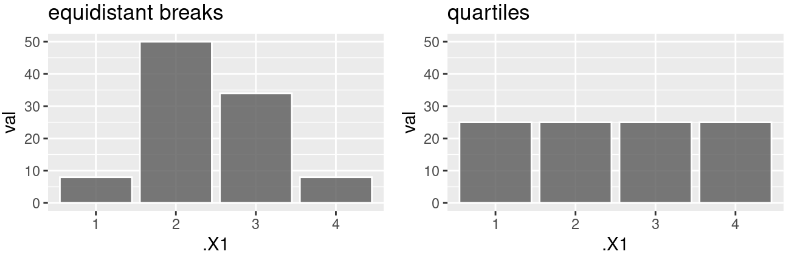
```
make_freq_marg(frequency_pr_dat, ind = 1)
make_freq_marg(frequency_pr_dat, ind = c(1,2))
```

Marginalization

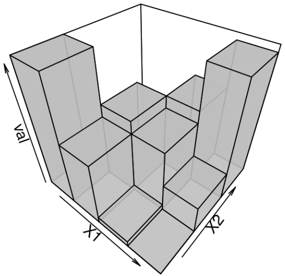
variable in real scale:

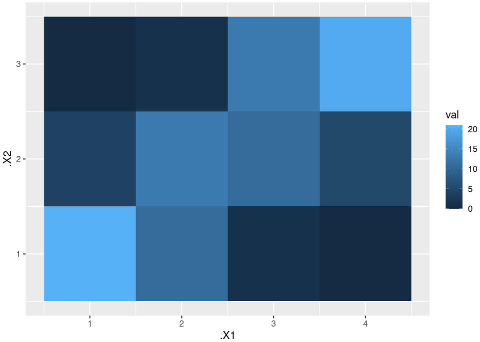


indexed bins:

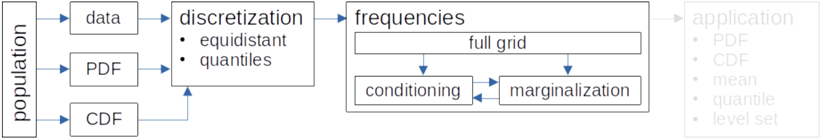


Marginalization





Conditioning



Another optional step within frequency stage is getting a conditional distribution

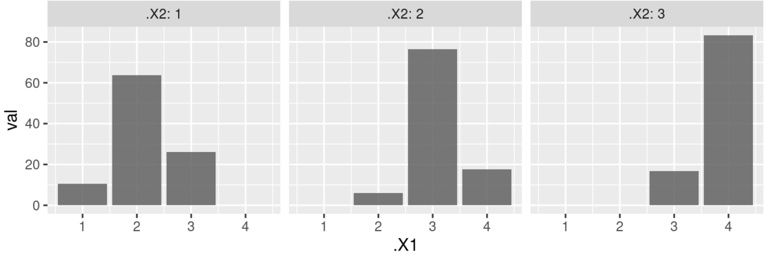
$$n_{i|jk} = n\frac{n_{ijk}}{n_{\cdot jk}}$$

$$n_{ij|k} = n\frac{n_{ijk}}{n_{\cdot\cdot k}}$$

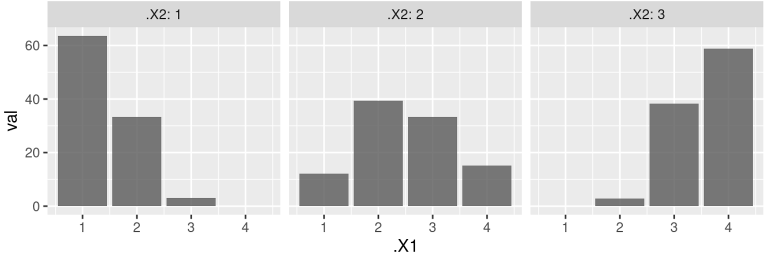
```
make_freq_cond(freq_pr_dat, ced = 1, cing = 2)
```

Conditioning

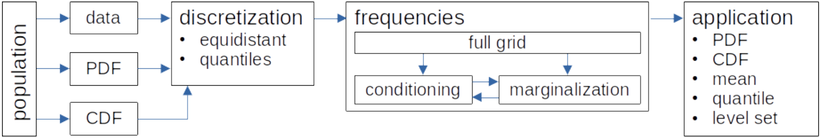
frequency of $X1|X2$ with equally spaced breaks:



frequency of $X1|X2$ with quantiles:



Application



Grid of frequency can be used to derive practical quantities related to joint distribution such as

- probability density function (PDF)
- cumulative distribution function (CDF)
- survival function
- mean values
- univariate quantiles (quantile function, QF)
- CDF level sets

PDF

One-dimensional via marginalization

$$Pr(b_{(i-1)\cdot\cdot} < X_1 \leq b_{i\cdot\cdot}) = \int_{b_{(i-1)\cdot\cdot}}^{b_{i\cdot\cdot}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2, x_3) dx_1 dx_2 dx_3$$

$$\frac{n_{i\cdot\cdot}}{n} = f_{i\cdot\cdot} d_{i\cdot\cdot}$$

$$\frac{n_{i\cdot\cdot}}{n\ d_{i\cdot\cdot}} = f_{i\cdot\cdot}$$

Higher-dimensional

$$f_{ijk} = \frac{n_{ijk}}{n\ d_i d_j d_k}$$

PDF

Conditional

$$f_{1|23}(x_1|x_2,x_3) = \frac{f(x_1,x_2,x_3)}{f_{23}(x_2,x_3)}$$

$$f_{i\cdot\cdot|\cdot jk} = \frac{n_{ijk}}{n_{\cdot jk}\, d_{i\cdot\cdot}} = \frac{n_{i\cdot\cdot|\cdot jk}}{n\ d_{i\cdot\cdot}}$$

Conditional and marginalized

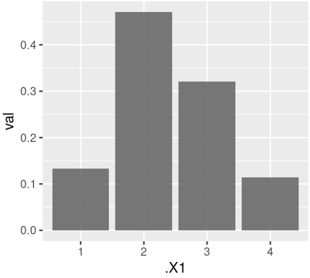
$$f_{ij\cdot|\cdot\cdot k} = \frac{n_{ijk}}{n_{\cdot\cdot k}\, d_{i\cdot\cdot} d_{\cdot j\cdot}} = \frac{n_{ij\cdot|\cdot\cdot k}}{n\ d_{i\cdot\cdot} d_{\cdot j\cdot}}$$

$$f_{i\cdot\cdot|\cdot\cdot k} = \frac{n_{i\cdot k}}{n_{\cdot\cdot k}\, d_{i\cdot\cdot}} = \frac{n_{i\cdot\cdot|\cdot\cdot k}}{n\ d_{i\cdot\cdot} d_{\cdot j\cdot}}$$

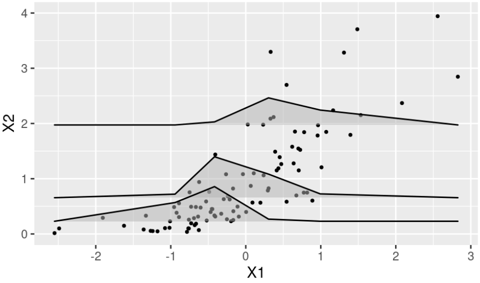
PDF

```
frequency_pr_dat |>                              # frequency full grid
  make_freq_marg(1) |>                           # marginalized frequency
  make_pdf(diffs = diff_breaks_pr_dat["X1"])     # PDF

frequency_pr_dat |>                              # frequency full grid
  make_freq_cond(ced = 1, cing = 2) |>           # conditioned frequency
  make_pdf(diffs = diff_breaks_pr_dat)           # PDF
```

marginal for X_1



conditional for $X_1|X_2$



CDF

Joint CDF

$$F(x_1, x_2, x_3) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \int_{-\infty}^{x_3} f(r, s, t)\, dr\, ds\, dt$$

$$F_{ijk} = \sum_{r=1}^{i} \sum_{s=1}^{j} \sum_{t=1}^{k} f_{rst}\, d_{r\cdot\cdot} d_{\cdot s\cdot} d_{\cdot\cdot t}$$

$$F_{ijk} = \frac{1}{n} \sum_{r=1}^{i} \sum_{s=1}^{j} \sum_{t=1}^{k} n_{rst}$$

CDF

Conditional CDF

$$F_{1|23}(x_1|x_2,x_3) \equiv Pr(X_1 \leq x_1|X_2 \leq x_2, X_3 \leq x_3)$$

$$F_{1|23}(x_1|x_2,x_3) = \int_{-\infty}^{x_1} f_{1|23}(r|x_2,x_3)\,dr$$

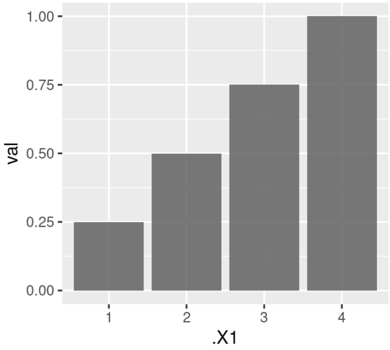
$$F_{i\cdot\cdot|\cdot jk} = \frac{1}{n_{\cdot jk}} \sum_{r=1}^{i} n_{rjk}$$

$$F_{ij\cdot|\cdot\cdot k} = \frac{1}{n_{\cdot\cdot k}} \sum_{r=1}^{i} \sum_{s=1}^{j} n_{rsk}$$

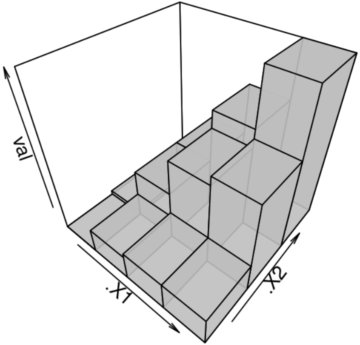
CDF

```
frequency_pr_dat |>            # frequency full grid
  make_freq_marg(ind = 1) |>  # marginalized frequency
  make_cdf()                   # CDF
```

marginal for X_1



conditional for $(X_1, X_2)|X_3$



Mean

Conditional

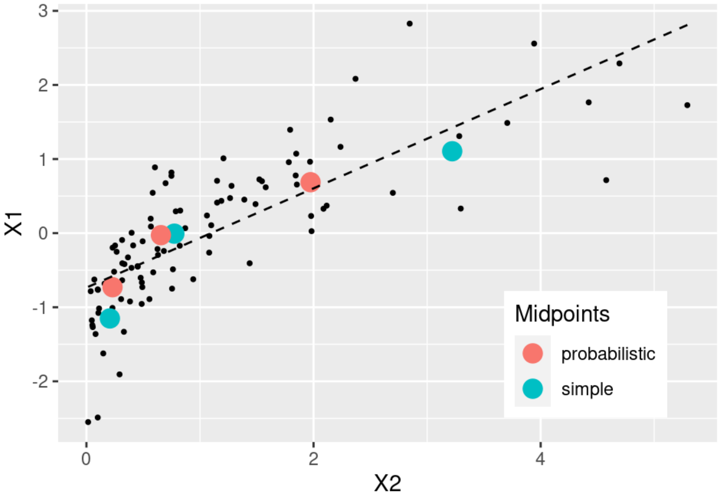
$$E(X_1|X_2 = x_2, X_3 = x_3) = \int_{-\infty}^{\infty} r\, f_{1|23}(x_1|x_2, x_3)\, dr$$

$$E_{1|\cdot jk} = \sum_{i=1}^{N_1} c_{i\cdot\cdot} \frac{n_{ijk}}{n_{\cdot jk}} = \frac{1}{n} \sum_{i=1}^{N_1} c_{i\cdot\cdot}\, n_{i\cdot\cdot|\cdot jk}$$

```
frequency_pr_dat |>                                  # frequency full grid
  make_freq_cond(ced = 1, cing = 2) |>               # conditional frequency
  make_mean(mids = midp_breaks_pr_dat)  # conditional mean
```

Mean

- calculated from median and average midpoints,
- comparison with OLS regression line



Quantile

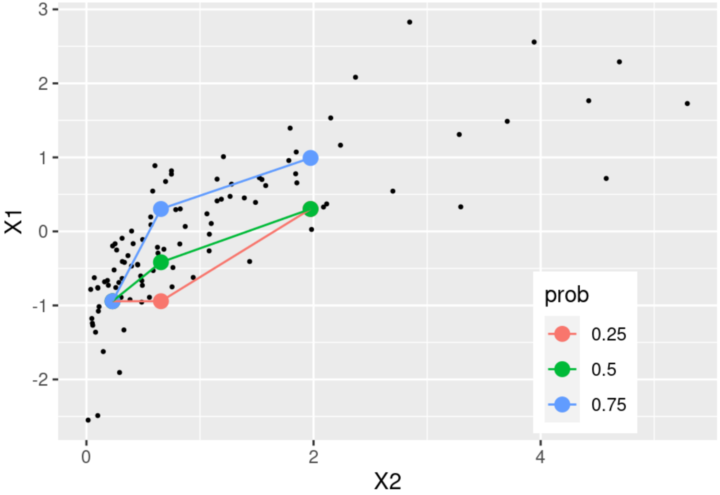
Quantile x_1 of X_1 conditional on values of X_2, X_3 and corresponding to probability p

$$F_{1|23}(x_1|x_2, x_3) = p$$
$$x_1 = F^{-1}_{1|23}(p|x_2, x_3)$$

```
frequency_pr_dat |>                               # frequency full grid
  make_freq_cond(ced = 1, cing = 2) |>            # conditional frequency
  make_quan(prob = c(0.25, 0.5, 0.75))  # conditional quantile
```

Quantile

- related to quantile regression



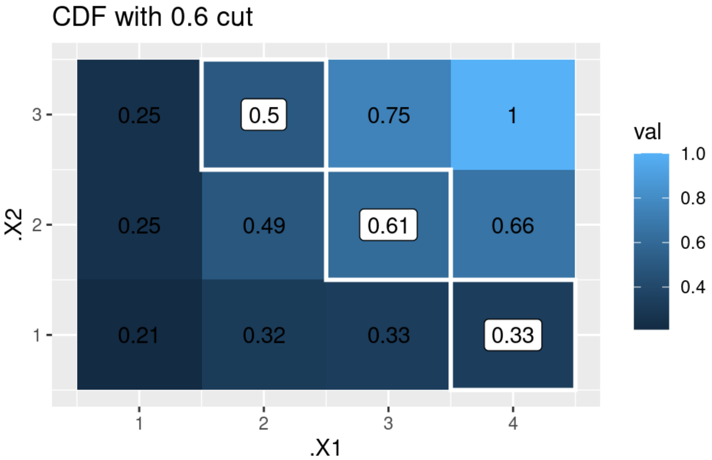
CDF level set

Pairs of quantiles (x_1, x_2) conditional on values of X_3 and corresponding to probability p:

$$\{(x_1, x_2) | F_{12|3}(x_1, x_2 | x_3) = p\}$$

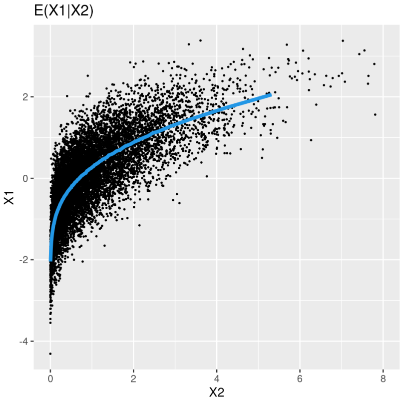
```
frequency_pr_dat |>              # frequency full grid
  make_freq_marg(ind = 1) |>    # conditional frequency
  make_cdf() |>                 # CDF
  cut_cdf(prob = c(0.6))        # cut CDF at given probability
```

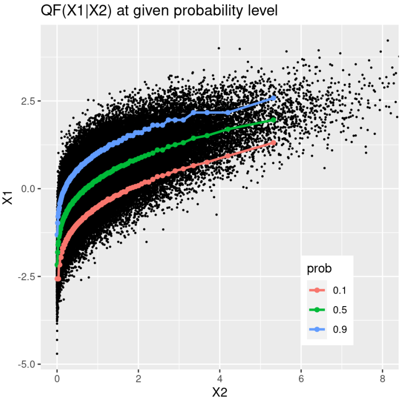
CDF level set



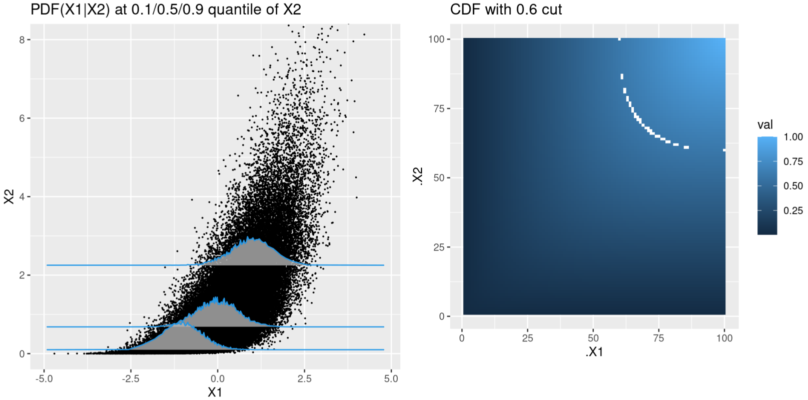
Finer grid

Let $n = 1 \cdot 10^6$ and $N_1 = N_2 = N_3 = 100$.





Finer grid


PDF(X1|X2) at 0.1/0.5/0.9 quantile of X2
X2
X1
8
6
4
2
0
-5.0
-2.5
0.0
2.5
5.0
CDF with 0.6 cut
.X2
.X1
100
75
50
25
0
val
1.00
0.75
0.50
0.25

Performance

Timing and memory consumption with 2021 processor and trivariate problem:

operation	N	n (10^6)	time (s)	volume (MB)
sample data	100	1	<1	24
		10	5	240
		100	40	2 400
breaks	100	1	<1	<1
		10	1.5	<1
		100	14	<1
	200	10	1.6	<1
frequency grid	100	1	1	16
		10	4	
		100	30	
	200	10	8	128

Conclusion

- distribution represented by counts in bins
- unifying approach
    - independent of model class
    - continuous with discrete RV
- scalable in
    - number of variables
    - precision (grid resolution)
- clear workflow
- modular in
    - input - distribution representation
    - output - application
- implemented in *R* using packages from *tidyverse* system

Future work

Things to finish, improve or add regarding to

- input: direct support for
  - models given by PDF and CDF
  - hybrid random vector (classification, clustering)
  - nonstandard models (factor copula)
- application:
  - survival function (probability of exceedance)
  - better level-set searching algorithm
  - easy replacement of indices by real values
  - refine with kernel smoothing on demand
- deployment:
  - available as package and in public
  - documentation
- optimization:
  - check for different data manipulation back-end (*data.table*)
  - parallelization wherever it makes sense (like in *make_freq*)
  - grid reduction to regions of interest (to save memory, especially in higher dimensions)

Thank you

and feel free to recall this presentation on

www.math.sk/bacigal