On some copula related software tools

Tomáš Bacigál

Slovak University of Technology in Bratislava
2012

Outline

- Overview of software for modelling with copulas
  - commercial
  - R packages
- Some copula constructions
  - Archimax
  - distorted univariate conditioning stable
- Implementation in our package
- Development environments

alternatives

- commercial

  Mathematica 8

  - copulas: Gumbel-Hougaard, Clayton, Frank, Farlie-Gordon-Morgenstern, Ali-Mikhail-Haq, normal, t-copula

alternatives

- commercial

  Mathematica 8

  Matlab

    part of in-built Statistics Toolbox

alternatives

- commercial

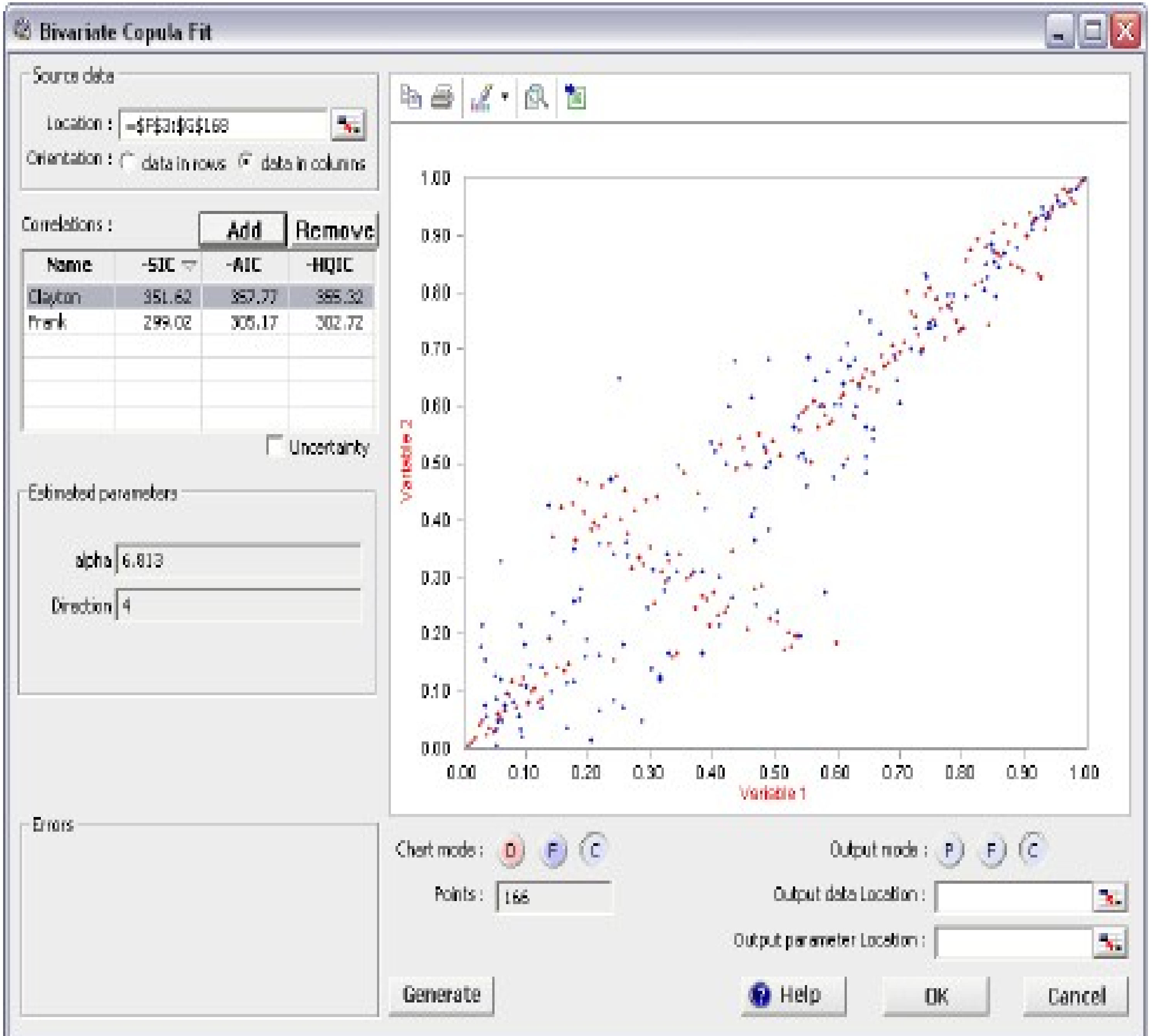
  Mathematica 8

  Matlab

  Excel

  - Hoadley Finance Add-in
  - Vose Model Risc

Model Risk 4


Bivariate Copula Fit
Source data
Location : =$F$3:$G$168
Orientation : data in rows data in columns
Correlations :
Add
Remove
Name
-SIC
-AIC
-HQIC
Clayton 351.62 357.77 355.32
Frank 299.02 305.17 302.72
Uncertainty
Estimated parameters
alpha 6.813
Direction 4
Errors
Variable 2
Variable 1
Chart mode :
Output mode :
Points : 166
Output data Location :
Output parameter Location :
Generate
Help
OK
Cancel

alternatives

- commercial

  Mathematica 8

  Matlab

  Excel

  S-plus

  - S+ Finmetrics / EVANESCE

alternatives

- commercial

  Mathematica 8

  Matlab

  Excel

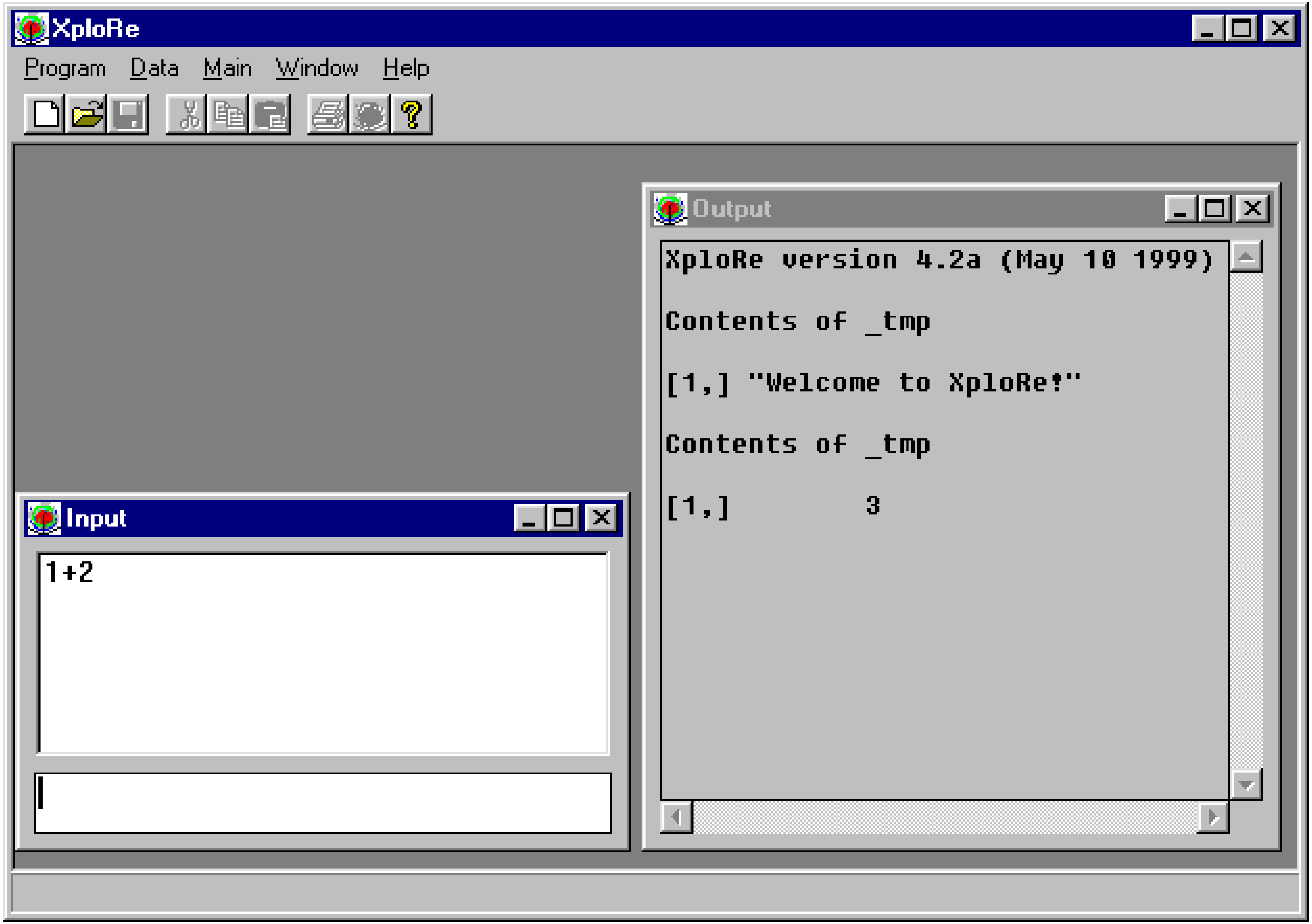
  S-plus

- „free”

  XploRe

  - abandoned in 2007, modelling VaR and extreme value

XploRe 4.2



XploRe
Program Data Main Window Help
Output
XploRe version 4.2a (May 10 1999)
Contents of _tmp
[1,] "Welcome to XploRe!"
Contents of _tmp
[1,] 3
Input
1+2

R-packages

- copula

  authors: I.Kojadinovic, J.Yan

  copulas: elliptical (normal and t), Archimedean (Clayton, Gumbel, Frank, and Ali-Mikhail-Haq), extreme value (Gumbel, Husler-Reiss,Galambos, Tawn, and t-EV), and other families (Plackett and Farlie-Gumbel-Morgenstern).

  description: Methods for density, distribution,random number generation, bi-variate dependence measures,perspective and contour plots. Functions for fitting copula models with variance estimate. Independence tests among random variables and random vectors. Serial independence tests for univariate and multivariate continuous time series. Goodness-of-fit tests for copulas based on multipliers and on the parametric bootstrap. Tests of extreme-value dependence.

R-packages

- copula
- CDVine

  authors: U.Schepsmeier, E.C.Brechmann

  description: This package provides functions for statistical inference of canonical vine (C-vine) and D-vine copulas. It contains tools for bivariate exploratory data analysis and for bivariate as well as vine copula selection. Models can be estimated either sequentially or by joint maximum likelihood estimation. Sampling algorithms and plotting methods are also included.

R-packages

- copula
- CDVine
- copulaedas

  authors: Y.González-Fernández, M.Soto

  description: Estimation of Distribution Algorithms Based on Copula Theory. This package contains implementations of various classes of Estimation of Distribution Algorithms (EDAs) based on copula theory: Copula EDAs and Vine EDAs. In this package, EDAs are implemented using S4 classes with generic functions for its main parts: seeding, selection, learning, sampling, replacement, local optimization, termination, and reporting. The package also includes the implementation of a group of well-known optimization test problems and utility functions to study the behavior of EDAs.

R-packages

- copula
- CDVine
- copulaedas
- copBasic

  author: W.H.Asquith

  description: Survival, dual, co-copula; level curves, inverses and derivatives for generating rv's; diagonal sections; Plackett copula; composition of 2 copulas (or 1 leading to asymmetric cop.; measures of association (Kendall's Tau, Spearman's Rho, Gini's Gamma, Blomqvist's Beta,bSchweizer and Wolff's Sigma), tail dependence.

R-packages

- copula
- CDVine
- copulaedas
- copBasic
- fCopulae (Rmetrics)

  authors: Diethelm Wuertz and many others

  descript.: Environment for teaching "Financial Engineering and Computational Finance". Archimedean, eliptical, EV c., interactive plots.

R-packages

- copula
- CDVine
- copulaedas
- copBasic
- fCopulae
- fgac

  author: V.A.Gonzalez-Lopez

  descript.: 7 families of copulas (Generalized Archimedean Copulas)

R-packages

- copula
- CDVine
- copulaedas
- copBasic
- fCopulae
- fgac
- HAC

  authors: O.Okhrin and A.Ristig

  description: Estimation of the structure and the parameters, simulation methods and structural plots of high-dimensional Hierarchical Archimedean Copulae

R-packages

- copula
- CDVine
- copulaedas
- copBasic
- fCopulae
- fgac
- HAC
- nacopula

  authors: M.Hofert, M.Maechler

  description: Nested Archimedean copulas. Procedures for computing function values and cube volumes, characteristics such as Kendall's tau and tail dependence coefficients, efficient sampling algorithms, various estimators, and goodness-of-fit tests. Also contains related univariate distributions and special functions such as the Sibuya distribution, the polylogarithm, Stirling and Eulerian numbers.

R-packages

- copula
- CDVine
- copulaedas
- copBasic
- fCopulae
- fgac
- HAC
- nacopula
- pencopula

  author: Ch.Schellhase

  description: Flexible copula density Estimation with penalized hierarchical B-Splines

R-packages

- copula
- CDVine
- copulaedas
- copBasic
- fCopulae
- fgac
- HAC
- nacopula
- pencopula
- sbgcop

  author: P.Hoff

  description: Semiparametric Bayesian Gaussian copula estimation and imputation

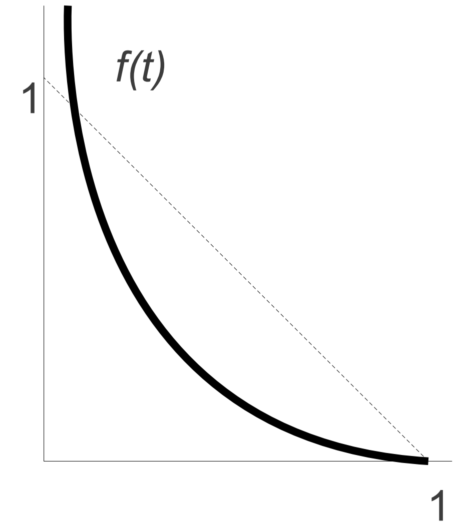
Copula constructs

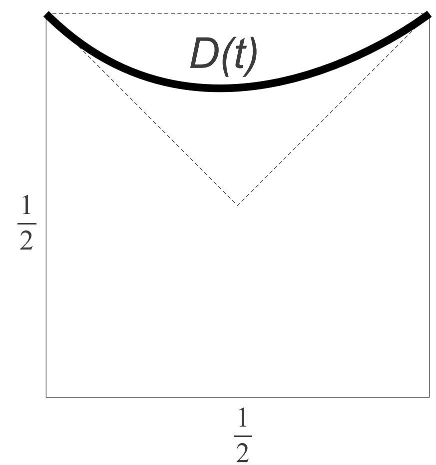
- Archimax

$$C_{f,D}(x,y) = f^{(-1)}\left[(f(x)+f(y))\, D\left(\frac{f(x)}{f(x)+f(y)} \right) \right]$$

where

- a convex continuous decreasing function f : [0, 1] → [0, ∞), $f(1) = 0$ is called a ***generator***, with pseudo-inverse $f^{(-1)}(x) = f^{-1}(\min(f(0), x))$,
- a convex function D : [0, 1] → [0, 1], max(t, 1 − t) ≤ $D(t)$ ≤ 1 for all t ∈ [0, 1], is called a ***dependence function***





Reference: (Capéraà et al. 2000)

Observe that Archimax copulas contains as special subclasses

- all Archimedean copulas (then $D \equiv 1$) and
- all extreme value copulas (then $f(t) = -\log(t)$).

Copula constructs

- Archimax

$$C_{f,D}(x,y) = f^{(-1)}\left[(f(x)+f(y))\, D\left(\frac{f(x)}{f(x)+f(y)} \right) \right]$$

where

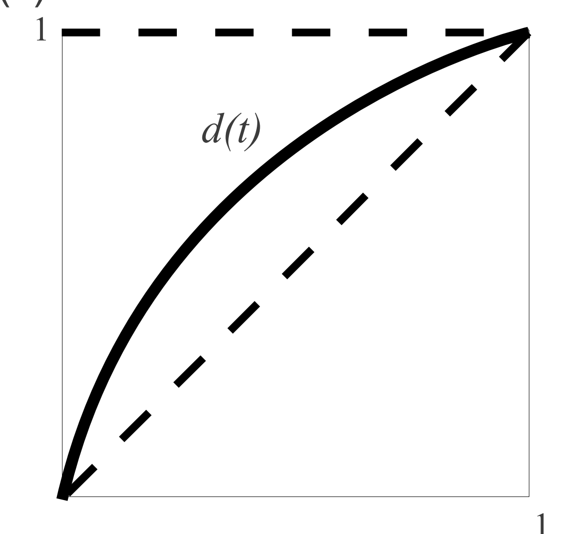
  - a convex continuous decreasing function f : [0, 1] → [0, ∞), $f(1) = 0$ is called a ***generator***, with pseudo-inverse $f^{(-1)}(x) = f^{-1}(\min(f(0), x))$,
  - a convex function D : [0, 1] → [0, 1], $\max(t, 1 - t) \leq D(t) \leq 1$ for all $t \in [0, 1]$, is called a ***dependence function***

- Distorted univariate conditioning stable (DUCS)

$$C_{(f,d)}(x,y) = x\, f^{(-1)}\left(\frac{f(y)}{d(x)} \right)$$

where

  - a non-decreasing function d : [0, 1] → [0, 1] is called ***distortion*** which needs to cooexist with a function $đ$: [0, 1] → [0, 1] such that $đ(x)d(x) = x$



Note that DUCS copulas can be seen as particular case of distortion

of general copulas $C_{(d)}(x, y) = đ(x) \cdot C(d(x), y)$

Reference: (Mesiar – Pekárová, 2010)

Copula constructs

- Archimax

$$C_{f,D}(x,y)=f^{(-1)}\left[(f(x)+f(y))\,D\left(\frac{f(x)}{f(x)+f(y)}\right)\right]$$

where

- a convex continuous decreasing function f : [0, 1] → [0, ∞), $f(1) = 0$ is called a ***generator***, with pseudo-inverse $f^{(-1)}(x) = f^{-1}(\min(f(0), x))$,
- a convex function D : [0, 1] → [0, 1], $\max(t, 1 - t) \le D(t) \le 1$ for all $t \in [0, 1]$, is called a ***dependence function***

- Distorted univariate conditioning stable (DUCS)

- Construction methods for compounds
  - generators f_i or it's inverse (Bacigál – Juráňová – Mesiar, 2010)

$$f=\sum_i a_i f_i$$

$$f=\left(\sum_i a_i f_i^{(-1)}\right)^{(-1)}$$

  - dependence functions (Bacigál – Jágr – Mesiar, 2010)

$$D(x)=\sum_i (a_i x+b_i(1-x))\,D_i\left(\frac{a_i x}{a_i x+b_i(1-x)}\right) \qquad \sum_i a_i=\sum_i b_i=1$$

$$D(x)=\left(\sum_i a_i B_i^{(-1)}\right)^{(-1)}(x)+1-x \quad \text{with} \quad B_i(x)=D_i(x)-1+x$$

Copula constructs

- Archimax

$$C_{f,D}(x,y)=f^{(-1)}\left[(f(x)+f(y))\,D\left(\frac{f(x)}{f(x)+f(y)}\right)\right]$$

where

- a convex continuous decreasing function f : [0, 1] → [0, ∞), f (1) = 0 is called a ***generator***, with pseudo-inverse $f^{(-1)}(x) = f^{-1}(\min(f(0), x))$,
- a convex function *D* : [0, 1] → [0, 1], max(*t*, 1 − *t*) ≤ *D*(*t*) ≤ 1 for all t ∈ [0, 1], is called a ***dependence function***

- Distorted univariate conditioning stable (DUCS)
- Construction methods for compounds
- d-dimensional Archimax (Jágr – Mesiar, 2012)

$$C_{f,L}(x_1,\dots,x_d)=f^{(-1)}\left[L\left(f(x_i)+\dots+f(x_d)\right)\right]$$

where

- *L* denotes tail dependence function $L(x_1,\dots,x_d)=(x_1,\dots,x_d)\,D\left(\frac{x_1}{\sum_i x_i},\dots,\frac{x_d}{\sum_i x_i}\right)$
- D is again (Pickand's) dependence function

Construction of d-dimensional dependence function (Jágr – Mesiar, 2012)

$$L(x_1,\dots,x_d)=\sum_j^n L_j(\alpha_{j1}x_1,\dots,\alpha_{jd}x_d)\ \text{ with }\ \sum_j^n \alpha_{ji}=1,\ \alpha_{ji}\geqslant 0$$

Implementation in our R package

```
> source(utils.R)
> source(archimax_functions.R)
> pCAX(0.5, 0.5, archimedean=amedProduct, dependence=depfu1)
[1] 0.25
> dCAX(0.5, 0.5, arch=amedGumbel, dep=depfuGalambos, apar=1, dpar=0)
[1] 1
```

Distribution related functions:

pCAX(u,v) - copula CDF
pCAXd(u,v,var="u") - part.deriv.
dCAX(u,v) - density
rCAX(n) - simulation
eCAX(data) - estimation
gofCAX(data) – GOF test

Archimedean generators:

amedProduct	$f(t) = -\log(t)$
AmedGumbel	$f(t) = [-\log(t)]^p$
AmedClayton	$f(t) = t^{-p} - 1$
AmedFrank	
AmedJoe	
amedBB1	

Dependence func.:

depfu1	D(t) = 1
depfuGumbel	
depfuMixed	
depfuGalambos	
depfuHuslerReiss	
depfuTawn	

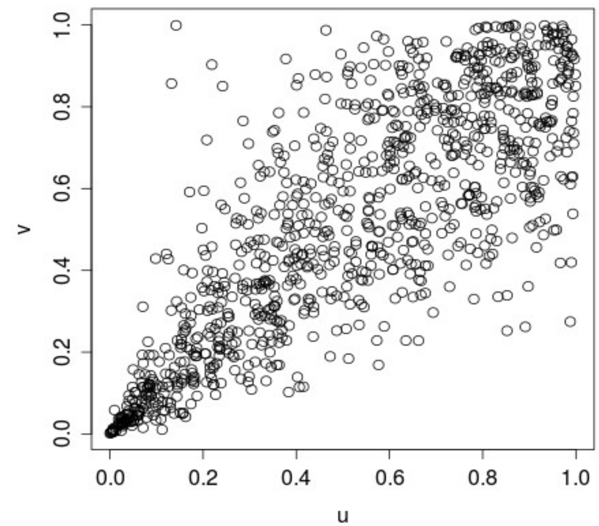
Implementation in our R package

```
amedGumbel <- list(

  parameters = c(2),

  gen = function(t, pars) (-log(t))^pars[1],

  gen.der = function(t, pars) -pars[1]*(-log(t))^(pars[1]-1)/t,

  gen.der2 = function(t, pars) pars[1]*(-log(t))^(pars[1]-2)*(pars[1]-1-log(t))/t^2,

  gen.inv = function(t, pars) exp(-t^(1/pars[1])),

  gen.inv.der = function(t, pars) -exp(-t^(1/pars[1]))*t^(1/pars[1]-1)/pars[1],

  gen.inv.der2 = function(t, pars) exp(-t^(1/pars[1]))*t^(1/pars[1]-2)*(pars[1]+t^(1/pars[1])-1)/pars[1]^2,

  lower = 1,          #Pi, g(t)=-ln(t)

  upper = Inf         #M

)
```

```
> amedGumbel$gen(0,1)

[1] Inf
```

Implementation in our R package

```
> amedCC <- famedCC( archimedean1=amedGumbel, archimedean2=amedClayton, pars1=5,pars2=3,inverse=FALSE)

> plot(rCAX(1000,arch=amedCC,apars=0.8))

> eCAX <- (data, arch=amedCC, alimits=list(0,1),technique="ML”,procedure=”optim", method="default")
```



eCAX parameters with values:

- data: (data frame or Nx2 matrix)
- archimedean
- dependence
- procedure: "optim", "nlminb", "nls", "grid"
- technique: "ML","LS"
- method: (depends on procedure)
- grid: number of points dividing parameters interval
- alimits: list of upper and lower bounds for parameters of archimedean parameters
- dlimits: the same but for dependence function
- aparameters: archimedean parameters set to be estimated over
- dparameters: the same for dependence functions

Integrated Development Environments

- for Windows

  Tinn-R, Notepad++ (NpptoR), RevolutionR

- for Linux

  Gedit or Kate (with plugins), **RKWard**

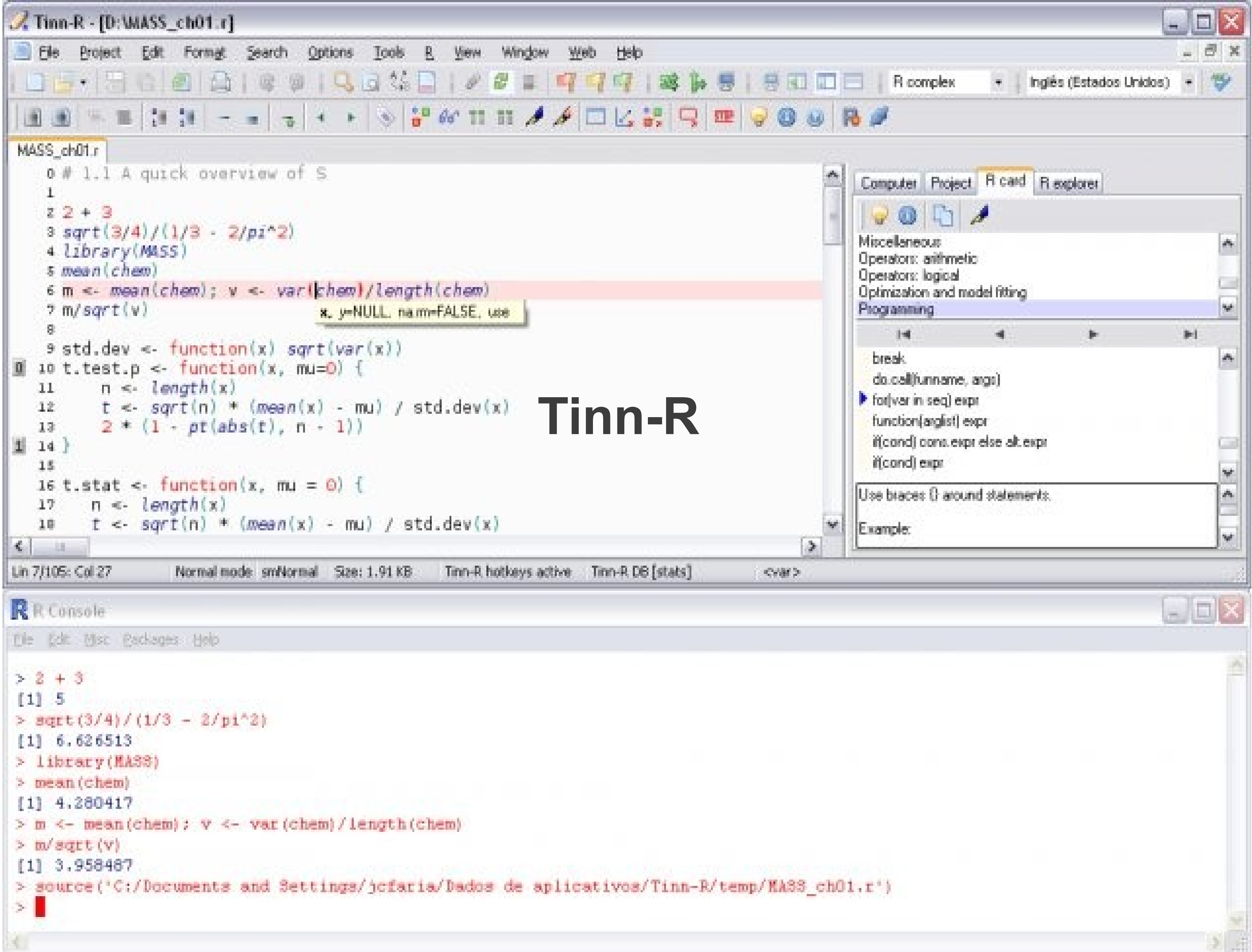
- multiplatform

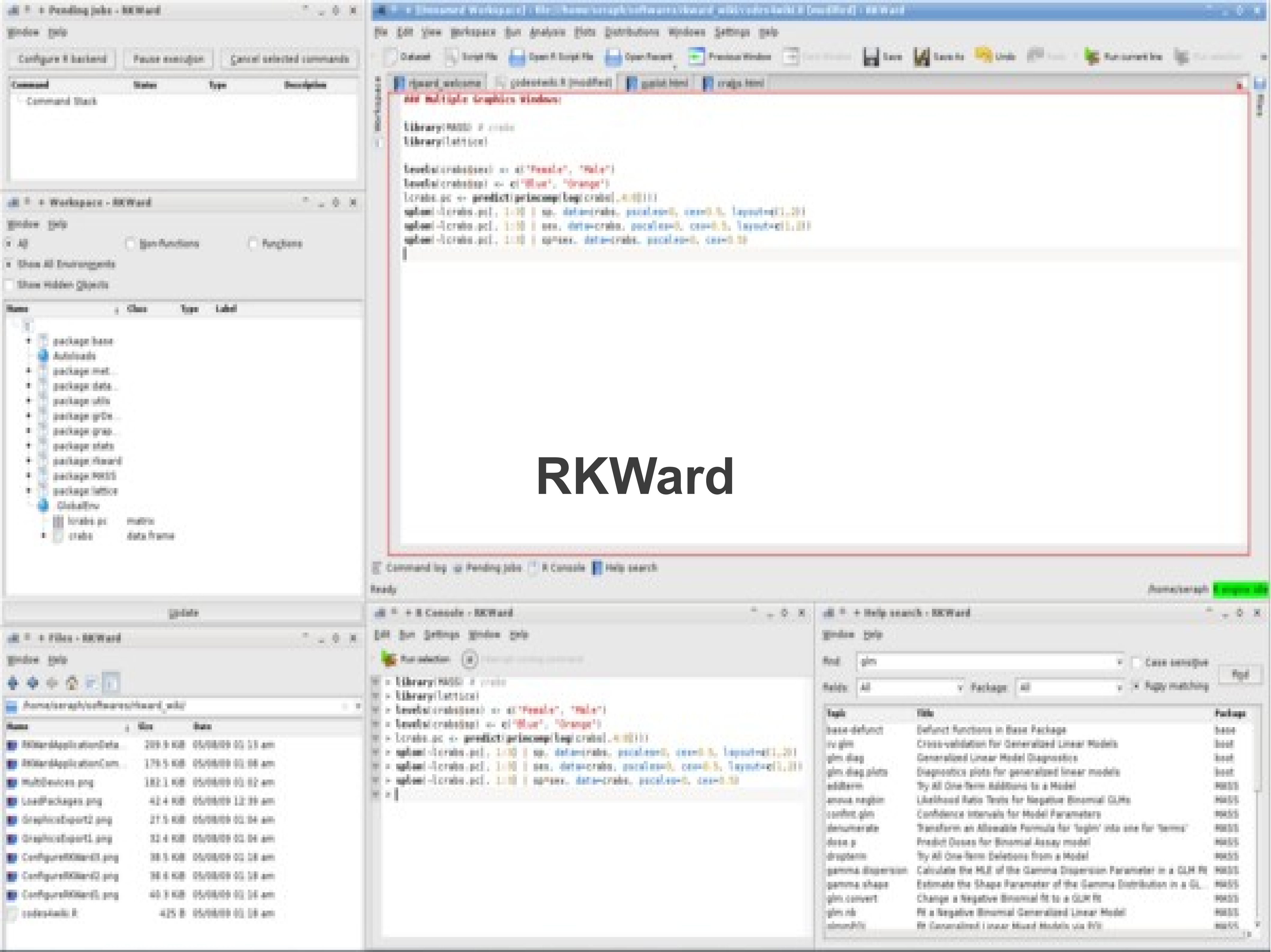
  Rstudio

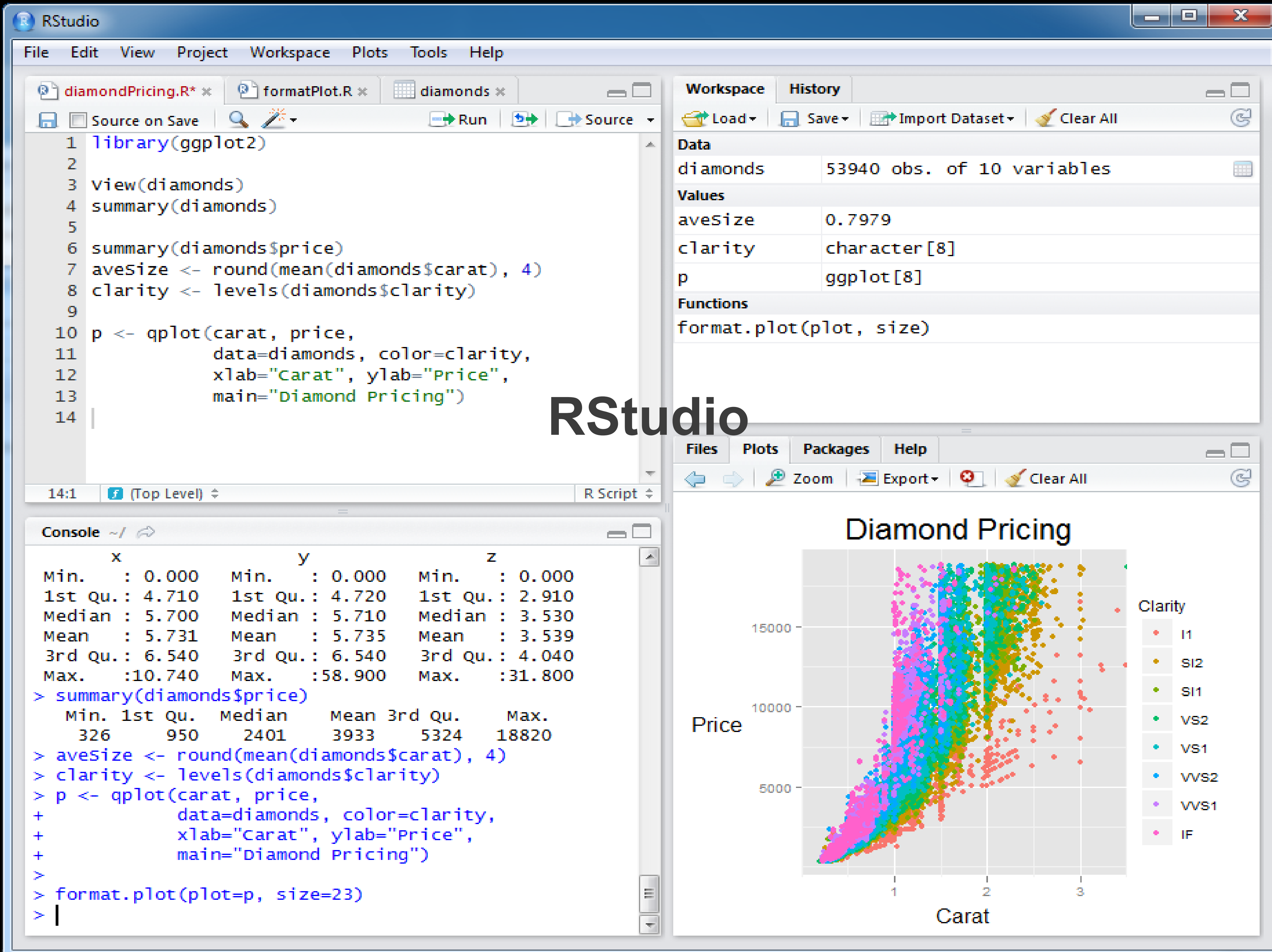
  Emacs or Eclipse (with plugins)

- package:

  Rcommander


Tinn-R - [D:\MASS_ch01.r]
File Project Edit Format Search Options Tools R View Window Web Help
R complex
Inglês (Estados Unidos)
MASS_ch01.r
1.1 A quick overview of S
2 + 3
sqrt(3/4)/(1/3 - 2/pi^2)
library(MASS)
mean(chem)
m <- mean(chem); v <- var(chem)/length(chem)
m/sqrt(v)
x, y=NULL, na.rm=FALSE, use
std.dev <- function(x) sqrt(var(x))
t.test.p <- function(x, mu=0) {
n <- length(x)
t <- sqrt(n) * (mean(x) - mu) / std.dev(x)
2 * (1 - pt(abs(t), n - 1))
}
t.stat <- function(x, mu = 0) {
n <- length(x)
t <- sqrt(n) * (mean(x) - mu) / std.dev(x)
Tinn-R
Computer Project R card R explorer
Miscellaneous
Operators: arithmetic
Operators: logical
Optimization and model fitting
Programming
break
do.call(funname, args)
for(var in seq) expr
function(arglist) expr
if(cond) cons.expr else alt.expr
if(cond) expr
Use braces {} around statements.
Example:
Lin 7/105: Col 27
Normal mode smNormal
Size: 1.91 KB
Tinn-R hotkeys active
Tinn-R DB [stats]
<var>
R Console
File Edit Misc Packages Help
> 2 + 3
[1] 5
> sqrt(3/4)/(1/3 - 2/pi^2)
[1] 6.626513
> library(MASS)
> mean(chem)
[1] 4.280417
> m <- mean(chem); v <- var(chem)/length(chem)
> m/sqrt(v)
[1] 3.958487
> source('C:/Documents and Settings/jcfaria/Dados de aplicativos/Tinn-R/temp/MASS_ch01.r')
>


RKWard


RStudio
File Edit View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save
Run
Source
library(ggplot2)
View(diamonds)
summary(diamonds)
summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)
p <- qplot(carat, price,
data=diamonds, color=clarity,
xlab="Carat", ylab="Price",
main="Diamond Pricing")
14:1
(Top Level)
R Script
Workspace History
Load Save Import Dataset Clear All
Data
diamonds 53940 obs. of 10 variables
Values
aveSize 0.7979
clarity character[8]
p ggplot[8]
Functions
format.plot(plot, size)
RStudio
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
15000
10000
5000
Carat
1
2
3
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
Console ~/
x y z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean : 5.731 Mean : 5.735 Mean : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max. :10.740 Max. :58.900 Max. :31.800
> summary(diamonds$price)
Min. 1st Qu. Median Mean 3rd Qu. Max.
326 950 2401 3933 5324 18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+ data=diamonds, color=clarity,
+ xlab="Carat", ylab="Price",
+ main="Diamond Pricing")
>
> format.plot(plot=p, size=23)
>