It is with great pleasure that we can announce the release of caracas version 1.0.1 to CRAN

The package enables user to make computer algebra from R using the Python library SymPy.

You can now install the caracas package as follows:

install.packages("caracas")

And then load it by:

library(caracas)

library(SumPy)

**Quick start**

x <- symbol('x')

eq <- 2\*x^2 - x

eq

## [caracas]: 2

## 2⋅x - x

as.character(eq)

## [1] "2\*x^2 - x"

as\_r(eq)

## expression(2 \* x^2 - x)

tex(eq) # $$`r tex(eq)`$$

## [1] "2 x^{2} - x"

\[2 x^{2} – x\]

solve\_sys(eq, x)

## Solution 1:

## x = 0

## Solution 2:

## x = 1/2

der(eq, x)

## [caracas]: 4⋅x - 1

subs(eq, x, "y")

## [caracas]: 2

## 2⋅y - y

**Linear algebra**

A <- matrix(c("x", 2, 0, "2\*x"), 2, 2)

B <- as\_symbol(A)

B

## [caracas]: ⎡x 0 ⎤

## ⎢ ⎥

## ⎣2 2⋅x⎦

determinant(B)

## [caracas]: 2

## 2⋅x

Binv <- inv(B) # or solve\_lin(B)

Binv

## [caracas]: ⎡ 1 ⎤

## ⎢ ─ 0 ⎥

## ⎢ x ⎥

## ⎢ ⎥

## ⎢-1 1 ⎥

## ⎢─── ───⎥

## ⎢ 2 2⋅x⎥

## ⎣ x ⎦

tex(Binv)

## [1] "\\left[\\begin{matrix}\\frac{1}{x} & 0\\\\- \\frac{1}{x^{2}} & \\frac{1}{2 x}\\end{matrix}\\right]"

\[\left[\begin{matrix}\frac{1}{x} & 0\\- \frac{1}{x^{2}} & \frac{1}{2 x}\end{matrix}\right]\]

eigen\_val(Binv)

## [[1]]

## [[1]]$eigval

## [caracas]: 1

## ─

## x

##

## [[1]]$eigmult

## [1] 1

##

##

## [[2]]

## [[2]]$eigval

## [caracas]: 1

## ───

## 2⋅x

##

## [[2]]$eigmult

## [1] 1

Matrix-vector multiplication with %\*%; subsetting with [, diag() etc. also works.

**Maximising the multinomial likelihood**

p <- as\_symbol(paste0("p", 1:3))

y <- as\_symbol(paste0("y", 1:3))

a <- as\_symbol("a")

l <- sum(y\*log(p))

l

## [caracas]: y₁⋅log(p₁) + y₂⋅log(p₂) + y₃⋅log(p₃)

L <- -l + a\*(sum(p) - 1)

L

## [caracas]: a⋅(p₁ + p₂ + p₃ - 1) - y₁⋅log(p₁) - y₂⋅log(p₂) - y₃⋅log(p₃)

g <- der(L, c(p, a))

g

## [caracas]: ⎡ y₁ y₂ y₃ ⎤

## ⎢a - ── a - ── a - ── p₁ + p₂ + p₃ - 1⎥

## ⎣ p₁ p₂ p₃ ⎦

sol <- solve\_sys(g, c(p, a))

sol

## Solution 1:

## p1 = y₁

## ────────────

## y₁ + y₂ + y₃

## p2 = y₂

## ────────────

## y₁ + y₂ + y₃

## p3 = y₃

## ────────────

## y₁ + y₂ + y₃

## a = y₁ + y₂ + y₃

sol[[1L]]$p1

## [caracas]: y₁

## ────────────

## y₁ + y₂ + y₃

tex(sol[[1L]]$p1)

## [1] "\\frac{y\_{1}}{y\_{1} + y\_{2} + y\_{3}}"

caracas objects can be turned into R objects using as\_r():

l

## [caracas]: y₁⋅log(p₁) + y₂⋅log(p₂) + y₃⋅log(p₃)

l\_e <- as\_r(l)

l\_e

## expression(y1 \* log(p1) + y2 \* log(p2) + y3 \* log(p3))

eval(l\_e, list(p1 = 0.2, p2 = 0.3, p3 = 0.5,

y1 = 18, y2 = 31, y3 = 51))

## [1] -101.6435