Computer algebra systems in R

COMPSTAT 2023 London, UK

Mikkel Meyer Andersen and Søren Højsgaard

8/24/23

Table of contents I

Take-home message

Take-home message

- The caracas package for R provides computer algebra / symbolic math
  - At your fingertips...
  - within R...
  - using R syntax
  - Calculus: derivatives, integrals, sums etc.
  - Linear algebra
  - Solving equations
- The caracas package can easily be extended
- Easy transition from symbolic expression to numerical expressions
- Easy generation of math expressions for documents (used in this presentation).
- See https://r-cas.github.io/caracas/ for vignettes and other info.

caracas

caracas

- Initiated in 2019 by **Søren Højsgaard** and **Mikkel Meyer Andersen**
- Supported by a grant from the R Consortium
- Based on SymPy (large computer algebra library for Python), using `reticulate` package in R.
- '*cara*': face in Spanish (Castellano) / '*cas*': computer algebra system
- Links
  - Stable version: https://CRAN.R-project.org/package=caracas
  - Development version: https://github.com/r-cas/caracas/
  - Online documentation: http://r-cas.github.io/caracas/

Getting started

Installation

```
#devtools::install_github("r-cas/caracas")
install.packages("caracas")
```

```
library(caracas)
packageVersion("caracas")
```

```
[1] '2.0.1.9001'
```

Symbols

```
def_sym(x, y)
p <- x^2 + 3*x + 4*y + y^4
p
```

```
[c]:  2           4
     x  + 3*x + y  + 4*y
```

```
str(x)
```

```
List of 1
 $ pyobj:x
 - attr(*, "class")= chr "caracas_symbol"
```

```
str(p)
```

```
List of 1
 $ pyobj:x**2 + 3*x + y**4 + 4*y
 - attr(*, "class")= chr "caracas_symbol"
```

Documents with mathematical contents

Write the following in LaTeX:

```
$$
p = `r tex(p)`
$$
```

Gives:

$$p = x^2 + 3x + y^4 + 4y$$

Used throughout this presentation :)

From symbols to R expressions and numerical evaluations

```r
p_ <- as_expr(p); p_
```

```
expression(x^2 + 3 * x + y^4 + 4 * y)
```

```r
eval(p_, list(x = 1, y = 1))
```

```
[1] 9
```

```r
p_fn <- as_func(p); p_fn
```

```
function (x, y)
{
    x^2 + 3 * x + y^4 + 4 * y
}
<environment: 0x55d59e8d59b0>
```

```r
p_fn(x = 1, y = 1)
```

```
[1] 9
```

Mathematical examples

Linear algebra

```
as_sym() # Converts R object to caracas symbol
```

```r
A <- matrix_(c(2, 1, 4, "x"), 2, 2)
## A <- as_sym(matrix(c(2, 1, 4, "x"), 2, 2)) ## Same
A
```

$$\begin{bmatrix} 2 & 4 \\ 1 & x \end{bmatrix}$$

```r
t(A)
```

$$\begin{bmatrix} 2 & 1 \\ 4 & x \end{bmatrix}$$

```r
det(A)
```

$$2x - 4$$

```r
A[2,]
```

$$\begin{bmatrix} 1 \\ x \end{bmatrix}$$

```r
A %*% A[2,]
```

$$\begin{bmatrix} 4x + 2 \\ x^2 + 1 \end{bmatrix}$$

```r
Ai  <- inv(A) |> simplify()
Ai
```

$$\begin{bmatrix} \frac{x}{2(x-2)} & -\frac{2}{x-2} \\ -\frac{1}{2x-4} & \frac{1}{x-2} \end{bmatrix}$$

Solving equations

```
# Solve Ax = b; also inv(A) for inverse of A
solve_lin(A, b)
# Solve lhs = rhs for vars; rhs omitted finds roots
solve_sys(lhs, rhs, vars)
```

```
def_sym(x, y)
lhs <- cbind(3 * x * y - y, x)
rhs <- cbind(-5 * x, y + 4)
```

$$\begin{bmatrix} 3xy - y \\ x \end{bmatrix} = \begin{bmatrix} -5x \\ y + 4 \end{bmatrix}$$

```r
sol <- solve_sys(lhs, rhs, list(x, y))
sol
```

```
Solution 1:
  x =  2/3
  y =  -10/3
Solution 2:
  x =  2
  y =  -2
```

Derivatives - gradient and Hessian

```
gp <- der(p, c(x, y))
gp
```

$$\begin{bmatrix} 2x+3 & 4y^3+4 \end{bmatrix}$$

```
H <- der2(p, c(x, y)) # Hessian
```

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 12y^2 \end{bmatrix}$$

Sums

```
sum_(expr, var, [from, to], doit = TRUE)
```

Find $\sum_{k=0}^{n} k^2$.

```
def_sym(k)
s1 <- sum_(k^2, k, 0, "n", doit = FALSE)
s2 <- doit(s1)
s3 <- s2 |> simplify()
```

$$s1 = \sum_{k=0}^{n} k^2; \quad s2 = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}; \quad s3 = \frac{n\left(2n^2 + 3n + 1\right)}{6}$$

Integration

```
int(expr, var, [from, to], doit = TRUE)
```

Upper half of unit circle: $y = \sqrt{1 - x^2}$ for $-1 \leq x \leq 1$.

```
def_sym(x, y)
y <- sqrt(1 - x^2)
s1 <- int(y, x)
s2 <- int(y, x, -1, 1)
```

$$y = \sqrt{1 - x^2};\ s1 = \frac{x\sqrt{1 - x^2}}{2} + \frac{\operatorname{asin}(x)}{2};\ s2 = \frac{\pi}{2}$$

Variance of the average of correlated data

Variance of the average of correlated data

Consider random variables $x_1, \dots, x_n$ where $\mathbf{Var}(x_i) = v$ and $\mathbf{Cov}(x_i, x_j) = vr$ for $i \neq j$, where $0 \leq |r| \leq 1$. For $n = 3$, the covariance matrix of $(x_1, \dots, x_n)$ is therefore

$$V = vR = v \begin{bmatrix} 1 & r & r \\ r & 1 & r \\ r & r & 1 \end{bmatrix}. \qquad (1)$$

Let $\bar{x} = \sum_i x_i / n$ denote the average.

- What is $\mathbf{Var}(\bar{x})$, when n goes to infinity for fixed r?
- What is $\mathbf{Var}(\bar{x})$, when r goes 0 and 1 for fixed n?
- How many independent observations do n correlated observations correspond to (in terms of the same variance of the averages)?

We need the variance of a sum $x. = \sum_i x_i$ which is

$$\mathbf{Var}(x.) = \sum_i \mathbf{Var}(x_i) + 2 \sum_{ij:i<j} \mathbf{Cov}(x_i, x_j) \tag{2}$$

$$= v(n + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} r) \tag{3}$$

(i.e., the sum of the elements of the covariance matrix). We can do this in caracas as follows:

```
def_sym(v, r, n, j, i)
s1 <- sum_(r, j, i+1, n)
s2 <- sum_(s1, i, 1, n-1)
var_sum <- v*(n + 2 * s2) |> simplify()
var_avg <- var_sum / n^2
```

$$\texttt{s1} = r\left(-i + n\right); \quad \texttt{s2} = nr\left(n - 1\right) - r\left(\frac{n^2}{2} - \frac{n}{2}\right) \tag{4}$$

$$\mathbf{Var}(x.) = nv\left(r\left(n-1\right)+1\right), \quad \mathbf{Var}(\bar{x}) \quad = \frac{v\left(r\left(n-1\right)+1\right)}{n}.$$

From hereof, we can study the limiting behavior of the variance $\mathbf{Var}(\bar{x})$ in different situations:

```
l_1 <- lim(var_avg, n, Inf)          ## n -> infinity
l_2 <- lim(var_avg, r, 0, dir='+') ## r -> 0
l_3 <- lim(var_avg, r, 1, dir='-') ## r -> 1
```

$$l_1 = rv, \quad l_2 = \frac{v}{n}, \quad l_3 = v,$$

For a given correlation r, investigate how many independent variables k_n the n correlated variables correspond to (in the sense of the same variance of the average).

Moreover, study how k_n behaves as function of n when $n \to \infty$. That is we must

1. solve $v(1+(n-1)r)/n = v/k_n$ for k_n and
2. find $\lim_{n\to\infty} k$:

```
def_sym(k_n)
k_n <- solve_sys(var_avg - v / k_n, k_n)[[1]]$k_n
l_k <- lim(k_n, n, Inf)
```

The findings above are:

$$k_n = \frac{n}{nr - r + 1}, \quad l_k = \frac{1}{r}.$$

It is illustrative to supplement the symbolic computations above with numerical evaluations.

```
dat <- expand.grid(r=c(.1, .2, .5), n=c(10, 50))
k_fun <- as_func(k_n)
dat$k_n <- k_fun(r=dat$r, n=dat$n)
dat$l_k <- 1/dat$r
dat
```

```
    r  n  k_n l_k
1 0.1 10 5.26  10
2 0.2 10 3.57   5
3 0.5 10 1.82   2
4 0.1 50 8.47  10
5 0.2 50 4.63   5
6 0.5 50 1.96   2
```

Shows that even a moderate correlation reduces the effective sample size substantially

Extending caracas

Extending caracas

Only small part of Sympy is interfaced from `caracas` but it is easy to extend `caracas`. For example: polynomial division

```
def_sym(x)
f = 5 * x^2 + 10 * x + 3
g = 2 * x + 2
```

$$f = 5x^2 + 10x + 3; \; g = 2x + 2$$

Find f/g; that is find q and r such that

$$f = qg + r$$

The Sympy function for polynomial division is `div` and it can be invoked via the `caracas` function `sympy_func`.

```
v <- sympy_func(f, "div", g)
v
```

```
[[1]]
[c]: 5*x   5
     --- + -
      2    2


[[2]]
[c]: -2
```

```
(v[[1]] * g + v[[2]]) |> simplify()
```

```
[c]:          2
     5*(x + 1)  - 2
```

Wrapping up

Wrapping up

- The caracas package for R provides computer algebra / symbolic math
  - At your fingertips within R using R syntax
  - For example: derivatives, integration, sums, limits, linear algebra, solving equations
- Package can easily be extended
- Easy transition from symbolic expression to numerical expressions
- Easy generation of math expressions for documents (used in this presentation).
- See https://r-cas.github.io/caracas/ for vignettes and other info.
- Thank you for your attention!