A quick guide to caracas

Mikkel Meyer Andersen (mikl@math.aau.dk, Dept Math Sci, Aalborg U, Denmark) and **Søren Højsgaard** (sorenh@math.aau.dk, Dept Math Sci, Aalborg U, Denmark)

What is **caracas**?

`caracas` is an R package that gives symbolic mathematics in R. `caracas` is based on SymPy (a computer algebra system for Python).
Function names are kept the same as in R if the function does the same, but have been given a postfix `_` if the functionality is different (e.g. `sum_()`).

Creating symbols

`k <- symbol("k")`	`k`
`def_sym(a, b)`	`a` `b`
`def_sym_vec(c("a", "b"))`	`a` `b`
`v <- vector_sym(2, "v")`	`[[v1, v2]]^T`
`M <- matrix_sym(2, 2, "m")`	`[[m11, m12],` ` [m21, m22]]`
`D <- matrix_sym_diag(2)`	`[[v1, 0],` ` [ 0, v2]]`

Coerce R objects to symbols

`T2 <- matrix(c("a", "b", "b", "a"), nrow = 2)`	
`T3 <- toeplitz(c("a", "b", "0"))`	
`T2 <- as_sym(T2)`	`[[a, b],` ` [b, a]]`

`T3 <- as_sym(T3)`	`[[a, b, 0],`
	` [b, a, b],`
	` [0, b, a]]`

Standard R functions

`c(v, v)`	*output omitted*
`cbind(v)`	*output omitted*
`rbind(v)`	*output omitted*
`sum(v)`	`v1 + v2`
`cumsum(v)`	`[[v1, v1 + v2]]^T`
`rep(v, times = 2)`	*output omitted*
`rep(v, each = 2)`	*output omitted*
`rev(v)`	`[[v2, v1]]^T`

Algebra

`simplify(cos(a)^2 + sin(a)^2)`	`1`
`solve_sys(a^2, -1, a)`	`a = -1i` `a = 1i`
`inv(T2)`	*output omitted*
`solve(T2)`	*output omitted*
`factor_(a^3 - a^2 + a - 1)`	`(a - 1)*(a^2 + 1)`
`expand((a - 1) * (a^2 + 1))`	`a^3 - a^2 + a - 1`

Calculus

`der(3 * a + a^2, a)`	`2*a + 3`
`sum_(1/a^2, a, 1, Inf)`	`pi^2/6`
`s <- sum_(1/a^2, a, 1, Inf, doit = FALSE)`	
`s`	$\sum_{a=1}^{\infty} \frac{1}{a^2}$
`doit(s)`	`pi^2/6`
`lim((1 + a)^(1/a), a, 0)`	`exp(1)`
`f <- taylor(cos(a), x0 = 0, n = 3 + 1)`	
`drop_remainder(f)`	`1 - a^2/2`

Subsetting

`T3[1:2, 2:3]`	`[[b, 0],` ` [a, b]]`
`T3[1:2]`	`[[a, b]]^T`
`T3[2]`	`b`
`T3[2, ]`	`[[b, a, b]]^T`

Linear algebra

`rankMatrix_(T2)`	`2`

`rref(T2)`	`$mat`
	`[[1, 0],`
	` [0, 1]]`
	`$pivot_vars`
	`[1] 1 2`

`T2i <- solve(T2)`	

`scale_matrix(T2i, det(T2i))`	`1/(a^2 - b^2)*[`
	`[ a, -b],`
	`[-b, a]]`

`QRdecomposition(D)`	*output omitted*
`LUdecomposition(D)`	*output omitted*
`chol(D, hermitian = FALSE)`	*output omitted*
`svd_(D)`	*output omitted*

Substitution and evaluation

`subs(T2, "b", "b-k")`	`[[ a, b - k],` ` [b - k, a]]`
`subs(T2, c("a", "b"), c(1, 2))`	`[[1, 2],` ` [2, 1]]`

Coercion to R objects

```
T2e <- as.expression(T2) # or as_expr()
T2e
```

```
## expression(matrix(c(a, b, b, a), nrow = 2))
```

```
T2f <- as.function(T2) # or as_func()
```

`eval(T2e, list(a = 1, b = 2))`	*output omitted*
`T2f(a = 1, b = 2)`	*output omitted*
`T2f2 <- as.function(T2, vec_arg = TRUE)`	
`T2f2(c(a = 1, b = 2))`	*output omitted*

Extending caracas

SymPy documentation at https://docs.sympy.org/.

With helper function `sympy_func()`:

`sympy_func(T2, "inverse_BLOCK")`	*output omitted*
`sympy_func(T2, "upper_triangular")`	*output omitted*

Calling SymPy directly via `reticulate`:

```
get_sympy()$diff("2*a*x**2", "x") |> as.character()
```

```
## [1] "4*a*x"
```

Output

Functions: `tex()`, `print(..., method = "prettyascii")` and others.

Chunk type `rtex` for e.g. `rmarkdown`/Quarto.

Bridging a gap

caracas

SYMBOLIC MATH IN R

Online docs
https://r-cas.github.io/caracas/

Updated Nov 2023
caracas v. 2.1.1