Aula 1

Prof. Lucas Helal, MMSc, PhD

2023-08-07

Conhecendo a linguagem R

- R: linguagem de programação de alto nível e multiparadigma, orientada à objetos
- R: Otimizada para análise estatística em diversas áreas do conhecimento; C++, por exemplo, é otimizada para rápido processamento
- Raciocínio muito parecido com a linguagem de humanos (alto nível)
- Fraca tipagem: case sensitive, parênteses, espaços, letras erradas...

Fazendo conta de padeiro com a linguagem R

```r
# Adição
2 + 2
```

```
## [1] 4
```

```r
# Divisão
2/2
```

```
## [1] 1
```

Aumentando um pouco o nível...

```
# Logaritmos
log(2)
```

```
## [1] 0.6931472
```

```
log2(3)
```

```
## [1] 1.584963
```

```
# Exponenciais
2^8
```

```
## [1] 256
```

```
2^(1/4)
```

```
## [1] 1.189207
```

Um pouco mais...

```
# trigonometria
sin(60)
```

```
## [1] -0.3048106
```

```
tan(90)
```

```
## [1] -1.9952
```

Ainda falta bastante...

```
# Cálculo Dif-Integral
(fx <- deriv(y ~ b0 + b1 * 2^(-x/th), c("b0", "b1", "th"),
             function(b0, b1, th, x = 1:7){} ) )
```

```
## function (b0, b1, th, x = 1:7)
## {
##     .expr3 <- 2^(-x/th)
##     .value <- b0 + b1 * .expr3
##     .grad <- array(0, c(length(.value), 3L), list(NULL, c("b0",
##         "b1", "th")))
##     .grad[, "b0"] <- 1
##     .grad[, "b1"] <- .expr3
##     .grad[, "th"] <- b1 * (.expr3 * (log(2) * (x/th^2)))
##     attr(.value, "gradient") <- .grad
##     .value
## }
```

```
fx(2, 3, 4)
```

```
## [1] 4.522689 4.121320 3.783811 3.500000 3.261345 3.060660 2.89190
```

Slide with R Output

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Slide with Plot

```
plot(pressure)
```