feature.r

denis

2021-07-13

#!/usr/bin/r  
  
c(workers = 2)

## workers   
## 2

xy <- list(1, 10, 100)  
yx <- list(1, 2, 3)  
zy <- list(5, 50, 500)  
  
c(xy, yx, ~ .x + .y)

## [[1]]  
## [1] 1  
##   
## [[2]]  
## [1] 10  
##   
## [[3]]  
## [1] 100  
##   
## [[4]]  
## [1] 1  
##   
## [[5]]  
## [1] 2  
##   
## [[6]]  
## [1] 3  
##   
## [[7]]  
## ~.x + .y

# Split into pieces, fit model to each piece, then predict  
by\_cyl <- split(mtcars, mtcars$cyl)  
mods <- c(by\_cyl, ~ lm(mpg ~ wt, data = .))  
c(mods, by\_cyl, predict)

## $`4`  
## mpg cyl disp hp drat wt qsec vs am gear carb  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2  
##   
## $`6`  
## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
##   
## $`8`  
## mpg cyl disp hp drat wt qsec vs am gear carb  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
##   
## [[4]]  
## ~lm(mpg ~ wt, data = .)  
##   
## $`4`  
## mpg cyl disp hp drat wt qsec vs am gear carb  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2  
##   
## $`6`  
## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
##   
## $`8`  
## mpg cyl disp hp drat wt qsec vs am gear carb  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
##   
## [[8]]  
## function (object, ...)   
## UseMethod("predict")  
## <bytecode: 0x557a19b80648>  
## <environment: namespace:stats>

c(list(xy, yx, zy), sum)

## [[1]]  
## [[1]][[1]]  
## [1] 1  
##   
## [[1]][[2]]  
## [1] 10  
##   
## [[1]][[3]]  
## [1] 100  
##   
##   
## [[2]]  
## [[2]][[1]]  
## [1] 1  
##   
## [[2]][[2]]  
## [1] 2  
##   
## [[2]][[3]]  
## [1] 3  
##   
##   
## [[3]]  
## [[3]][[1]]  
## [1] 5  
##   
## [[3]][[2]]  
## [1] 50  
##   
## [[3]][[3]]  
## [1] 500  
##   
##   
## [[4]]  
## function (..., na.rm = FALSE) .Primitive("sum")

# Matching arguments by position  
c(list(xy, yx, zy), function(a, b ,c) a / (b + c))

## [[1]]  
## [[1]][[1]]  
## [1] 1  
##   
## [[1]][[2]]  
## [1] 10  
##   
## [[1]][[3]]  
## [1] 100  
##   
##   
## [[2]]  
## [[2]][[1]]  
## [1] 1  
##   
## [[2]][[2]]  
## [1] 2  
##   
## [[2]][[3]]  
## [1] 3  
##   
##   
## [[3]]  
## [[3]][[1]]  
## [1] 5  
##   
## [[3]][[2]]  
## [1] 50  
##   
## [[3]][[3]]  
## [1] 500  
##   
##   
## [[4]]  
## function(a, b ,c) a / (b + c)

# Vectoring a function over multiple arguments  
df <- data.frame(  
 x = c("apple", "banana", "cherry"),  
 pattern = c("p", "n", "h"),  
 replacement = c("x", "f", "q"),  
 stringsAsFactors = FALSE  
)  
  
c(df, gsub)

## $x  
## [1] "apple" "banana" "cherry"  
##   
## $pattern  
## [1] "p" "n" "h"  
##   
## $replacement  
## [1] "x" "f" "q"  
##   
## [[4]]  
## function (pattern, replacement, x, ignore.case = FALSE, perl = FALSE,   
## fixed = FALSE, useBytes = FALSE)   
## {  
## if (!is.character(x))   
## x <- as.character(x)  
## .Internal(gsub(as.character(pattern), as.character(replacement),   
## x, ignore.case, perl, fixed, useBytes))  
## }  
## <bytecode: 0x557a1937e078>  
## <environment: namespace:base>

c(df, gsub)

## $x  
## [1] "apple" "banana" "cherry"  
##   
## $pattern  
## [1] "p" "n" "h"  
##   
## $replacement  
## [1] "x" "f" "q"  
##   
## [[4]]  
## function (pattern, replacement, x, ignore.case = FALSE, perl = FALSE,   
## fixed = FALSE, useBytes = FALSE)   
## {  
## if (!is.character(x))   
## x <- as.character(x)  
## .Internal(gsub(as.character(pattern), as.character(replacement),   
## x, ignore.case, perl, fixed, useBytes))  
## }  
## <bytecode: 0x557a1937e078>  
## <environment: namespace:base>