R for DS - 3/08/17

Pipes

Chilren's story…

```r
bop(
  scoop(
    hop(foo_foo, through = forest),
    up = field_mice
  ),
  on = head
)
```

Final version:

```r
library(magrittr)

foo_foo %>%
  hop(through = forest) %>%
  scoop(up = field_mouse) %>%
  hop(through = forest) %>%
  scoop(up = field_mouse) ->
through_forest

through_forest %>%
  hop(through = forest) %>%
  scoop(up = field_mouse) %>%
  hop(through = forest) %>%
  scoop(up = field_mouse) %>%
  bop(on = head)

foo_foo %>%
  hop(through = forest) %>%
  scoop(up = field_mouse) %>%
  bop(on = head) ->
ans
```

```r
library(magrittr)

# wrong
"my string" %>%
  gsub("string", "character")
```

```
## [1] "character"
```

```r
# correct behavior
"my string" %>%
  gsub("string", "character", x = .)
```

```
## [1] "my character"
```

```r
"my string" %>%
  gsub("string", "character", .)
```

```
## [1] "my character"
```

```r
"my string" %>%
  gsub(x = ., "string", "character")
```

```
## [1] "my character"
```

Functions

```r
df <- tibble::tibble(
  a = rnorm(10),
  b = rnorm(10),
  c = rnorm(10),
  d = rnorm(10)
)
df
```

```
## # A tibble: 10 × 4
##              a          b          c           d
##          <dbl>      <dbl>      <dbl>       <dbl>
## 1  -1.75273317  1.5215680 -1.3511604  1.36558182
## 2  -1.21719048  0.1234797 -1.0343316 -0.07672493
## 3  -2.22602988 -0.6509647 -1.3131585  0.96036477
## 4   0.06161627 -0.6618843  0.2566936 -0.15213864
## 5   0.14487444  1.9654451 -0.4865959  0.79019658
## 6   0.00542494 -0.6382923  0.8884249 -1.46989522
## 7  -1.45820819  0.3631996  1.1061912 -0.82931249
## 8   0.05797202 -1.9565139 -0.7136450  0.04137858
## 9   0.52426397 -0.1046669  1.1263203  0.07218637
## 10 -0.20358472 -1.0682561  1.0480330  1.24735180
```

```r
df$a <- (df$a - min(df$a, na.rm = TRUE)) /
  (max(df$a, na.rm = TRUE) - min(df$a, na.rm = TRUE))
df$b <- (df$b - min(df$b, na.rm = TRUE)) /
  (max(df$b, na.rm = TRUE) - min(df$a, na.rm = TRUE))
df$c <- (df$c - min(df$c, na.rm = TRUE)) /
  (max(df$c, na.rm = TRUE) - min(df$c, na.rm = TRUE))
df$d <- (df$d - min(df$d, na.rm = TRUE)) /
  (max(df$d, na.rm = TRUE) - min(df$d, na.rm = TRUE))
df
```

```
## # A tibble: 10 × 4
##            a         b          c         d
##        <dbl>     <dbl>      <dbl>     <dbl>
## 1  0.1720895 1.7696154 0.00000000 1.0000000
## 2  0.3668115 1.0582812 0.12788347 0.4913354
## 3  0.0000000 0.6642511 0.01533896 0.8570903
## 4  0.8317824 0.6586953 0.64898751 0.4647389
## 5  0.8620549 1.9954559 0.34896920 0.7970764
## 6  0.8113514 0.6706987 0.90397687 0.0000000
## 7  0.2791781 1.1802484 0.99187516 0.2259171
## 8  0.8304574 0.0000000 0.25732405 0.5329875
## 9  1.0000000 0.9422023 1.00000000 0.5438526
## 10 0.7353560 0.4519372 0.96840045 0.9583033
```

```r
rescale01 <- function(x) {
  rng <- range(x, na.rm = TRUE)
  (x - rng[1]) / (rng[2] - rng[1])
}
rescale01(c(0, 5, 10))
```

```
## [1] 0.0 0.5 1.0
```

```r
rescale01(c(0, 5, 100))
```

```
## [1] 0.00 0.05 1.00
```

‘...’: all ‘other’ arguments

```r
string_to_character <- function(x, pattern, replacement, ...) {
  gsub(pattern = pattern, replacement = replacement, x = x, ...)
}
string_to_character(
  "my string",
  "string",
  "character"
)
```

```
## [1] "my character"
```

```r
string_to_character(
  "my String",
  "string",
  "character",
  ignore.case = TRUE
)
```

```
## [1] "my character"
```

```r
df$a <- rescale01(df$a)
df$b <- rescale01(df$b)
df$c <- rescale01(df$c)
df$d <- rescale01(df$d)
df
```

```
## # A tibble: 10 × 4
##            a         b          c         d
##        <dbl>     <dbl>      <dbl>     <dbl>
## 1  0.1720895 0.8868226 0.00000000 1.0000000
## 2  0.3668115 0.5303456 0.12788347 0.4913354
## 3  0.0000000 0.3328819 0.01533896 0.8570903
## 4  0.8317824 0.3300977 0.64898751 0.4647389
## 5  0.8620549 1.0000000 0.34896920 0.7970764
## 6  0.8113514 0.3361130 0.90397687 0.0000000
## 7  0.2791781 0.5914681 0.99187516 0.2259171
## 8  0.8304574 0.0000000 0.25732405 0.5329875
## 9  1.0000000 0.4721740 1.00000000 0.5438526
## 10 0.7353560 0.2264832 0.96840045 0.9583033
```

```r
op_fn <- function(x, y, op) {
 switch(op,
   plus = x + y,
   minus = x - y,
   times = x * y,
   divide = x / y,
   stop("Unknown op!")
 )
}
op_fn(4, 7, "plus")
```

```
## [1] 11
```

```
add_or_divide_only <- function(x, y, op) {
 switch(op,
   plus = x + y,
   minus = ,
   times = ,
   divide = x / y,
   stop("Unknown op!")
 )
}
add_or_divide_only(4, 7, "plus")
```

```
## [1] 11
```

```
add_or_divide_only(4, 7, "minus")
```

```
## [1] 0.5714286
```

```
add_or_divide_only(4, 7, "times")
```

```
## [1] 0.5714286
```

```
add_or_divide_only(4, 7, "divide")
```

```
## [1] 0.5714286
```

```
letter_grade <- function(any_grade) {
  ans <- switch(any_grade,
    "A+" = ,
    "A" = ,
    "A-" = "A",
    "B+" = ,
    "B" = ,
    "B-" = "B",
    "F"
  )

  ans
}
lapply(
  c("A", "A+", "B+", "A-", "C"),
  letter_grade
) %>%
  unlist()
```

```
## [1] "A" "A" "B" "A" "F"
```

function scope

```r
f <- function(x) {
  f <- function(x) {
    f <- function(x) {
      x + 5
    }
    f(x * 2)
  }
  f(x + 100)
}
f(10)
```

```
## [1] 225
```

Parallel

```r
library(plyr)

my_func <- function(i) {
  print(i)
  Sys.sleep(2)
}
my_func(1)
```

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

```r
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```r
registerDoParallel(4)

llply(1:8, my_func, .parallel = FALSE)
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
```

```r
llply(1:40, my_func, .parallel = TRUE)
```

```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## [[6]]
## NULL
##
## [[7]]
## NULL
##
## [[8]]
## NULL
##
## [[9]]
## NULL
##
## [[10]]
## NULL
##
## [[11]]
## NULL
##
## [[12]]
## NULL
##
## [[13]]
## NULL
##
```

```
## [[14]]
## NULL
##
## [[15]]
## NULL
##
## [[16]]
## NULL
##
## [[17]]
## NULL
##
## [[18]]
## NULL
##
## [[19]]
## NULL
##
## [[20]]
## NULL
##
## [[21]]
## NULL
##
## [[22]]
## NULL
##
## [[23]]
## NULL
##
## [[24]]
## NULL
##
## [[25]]
## NULL
##
## [[26]]
## NULL
##
## [[27]]
## NULL
##
## [[28]]
## NULL
##
## [[29]]
## NULL
##
## [[30]]
## NULL
##
## [[31]]
## NULL
##
```

```
## [[32]]
## NULL
##
## [[33]]
## NULL
##
## [[34]]
## NULL
##
## [[35]]
## NULL
##
## [[36]]
## NULL
##
## [[37]]
## NULL
##
## [[38]]
## NULL
##
## [[39]]
## NULL
##
## [[40]]
## NULL
```

```r
library(memoise)

fib <- function(n) {
  if (n < 2) {
    return(1)
  }
  fib(n - 1) + fib(n - 2)
}
fib(10)
```

```
## [1] 89
```

```r
fib(32)
```

```
## [1] 3524578
```

```r
# fib_m(300) # take too long!!

library(memoise)
fib_m <- memoise(function(n) {
  if (n < 2) {
    return(1)
  }
  fib_m(n - 1) + fib_m(n - 2)
})
fib_m(32)
```

```
## [1] 3524578
```

```r
fib_m(300)
```

```
## [1] 3.595793e+62
```