Function design

https://design.tidyverse.org

July 2022

Hadley Wickham
@hadleywickham
Chief Scientist, **RStudio**

Five reasonably complete chapters about defaults

1. Required arguments shouldn't have defaults
   https://design.tidyverse.org/def-required.html
2. Enumerate possible options
   https://design.tidyverse.org/def-enum.html
3. Avoid magical defaults
   https://design.tidyverse.org/def-magical.html
4. Keep defaults short and sweet
   https://design.tidyverse.org/def-short.html
5. Explain important defaults
   https://design.tidyverse.org/def-inform.html

Enumerate possible options

```
rank ← function(x, ties.method =
  c("average", "first", "last", "random", "max", "min")) {

  ties.method ← match.arg(ties.method)
}
```

Enumerate possible options

```
library(rlang)

rank ← function(x, ties.method =
  c("average", "first", "last", "random", "max", "min")) {

  ties.method ← arg_match(ties.method)
}
```

Documentation

```
#' @param ties.method For each set of ties:
#' * `"average"` (default): replaces by mean.
#' * `"first"`: break ties with index.
#' * `"last"`: break ties with -index.
#' * `"random"`: put in random order.
#' * `"max"`: replace with maximum.
#' * `"min"`: replace with minimum.
#'     Typical sports ranking.
```

Keep defaults short and sweet

```
rank ← function(x, ties.method = "average") {
  ties.method ← arg_match(ties.method, possible_ties)
}
possible_ties ← c(
  "average", "first", "last", "random", "max", "min"
)
```

```
rank ← function(x, ties.method = NULL) {
  ties.method ← arg_match(ties.method, possible_ties)
}
possible_ties ← c(
  "average", "first", "last", "random", "max", "min"
)
```

```
?reshape
```

Instead use NULL

```
reshape ← function( ... , split = NULL)

  if (is.null(split)) {
    if (sep == "") {
      split ← list(regexp = "[A-Za-z][0-9]", include = TRUE)
    } else {
      split ← list(regexp = sep, include = FALSE, fixed = TRUE)
    }
  }
  ...
}
```

Or make a helper

```
reshape ← function( ... , split = reshape_split(sep))
    ...
}

reshape_split ← function(sep) {
  if (is.null(split)) {
    if (sep == "") {
      list(regexp = "[A-Za-z][0-9]", include = TRUE)
    } else {
      list(regexp = sep, include = FALSE, fixed = TRUE)
    }
  }
}

# I'd probably go even further here since the split argument
# has a very specific form
```