ST551: HOMEWORK 4

Brian Cervantes Alvarez
November 6, 2023

Part A

```
v <- c(18,10,6,9,6,11)
```

```
person
```

```
function (given = NULL, family = NULL, middle = NULL, email = NULL,
    role = NULL, comment = NULL, first = NULL, last = NULL)
{
    args <- list(given = given, family = family, middle = middle,
        email = email, role = role, comment = comment, first = first,
        last = last)
    if (all(vapply(args, is.null, NA)))
        return(.person())
    args <- lapply(args, .listify)
    args_length <- lengths(args)
    if (!all(args_length_ok <- args_length %in% c(1L, max(args_length))))
        warning(gettextf("Not all arguments are of the same length, the following need to be r
            paste(names(args)[!args_length_ok], collapse = ", ")),
            domain = NA)
    args <- lapply(args, function(x) rep_len(x, max(args_length)))
    person1 <- function(given = NULL, family = NULL, middle = NULL,
        email = NULL, role = NULL, comment = NULL, first = NULL,
        last = NULL) {
        if (!.is_not_nonempty_text(first)) {
            if (!.is_not_nonempty_text(given))
                stop(gettextf("Use either %s or %s/%s but not both.",
                  sQuote("given"), sQuote("first"), sQuote("middle")),
                  domain = NA)
            warning(gettextf("It is recommended to use %s instead of %s.",
                sQuote("given"), sQuote("first")), domain = NA)
            given <- first
        }
        if (!.is_not_nonempty_text(middle)) {
            warning(gettextf("It is recommended to use %s instead of %s.",
                sQuote("given"), sQuote("middle")), domain = NA)
            given <- c(given, unlist(strsplit(middle, "[[:space:]]+")))
        }
        if (!.is_not_nonempty_text(last)) {
            if (!.is_not_nonempty_text(family))
                stop(gettextf("Use either %s or %s but not both.",
                  sQuote("family"), sQuote("last")), domain = NA)
            warning(gettextf("It is recommended to use %s instead of %s.",
                sQuote("family"), sQuote("last")), domain = NA)
            family <- last
        }
        if (.is_not_nonempty_text(given))
            given <- NULL
        if (.is_not_nonempty_text(family))
            family <- NULL
        if (.is_not_nonempty_text(email))
            email <- NULL
        if (.is_not_nonempty_text(role)) {
            if (!is.null(role))
                warning(sprintf(ngettext(length(role), "Invalid role specification: %s.",
                  "Invalid role specifications: %s."), paste(sQuote(role),
                  collapse = ", ")), domain = NA)
            role <- NULL
        }
        if (.is_not_nonempty_text(comment))
            comment <- NULL
        if (length(role))
            role <- .canonicalize_person_role(role)
        if (length(comment)) {
            ind <- grepl(sprintf("^https?://orcid.org/%s$", tools:::.ORCID_iD_regexp),
                comment)
            if (any(ind)) {
                if (is.null(names(comment)))
                  names(comment) <- ifelse(ind, "ORCID", "")
                else names(comment)[ind] <- "ORCID"
            }
        }
        rval <- list(given = given, family = family, role = role,
            email = email, comment = comment)
        if (any(ind <- (lengths(rval) == 0L)))
            rval[ind] <- vector("list", length = sum(ind))
        if (all(vapply(rval, is.null, NA)))
            NULL
        else rval
    }
    force(person1)
    rval <- lapply(seq_along(args$given), function(i) with(args,
        person1(given = given[[i]], family = family[[i]], middle = middle[[i]],
            email = email[[i]], role = role[[i]], comment = comment[[i]],
            first = first[[i]], last = last[[i]])))
    .person(rval[!vapply(rval, is.null, NA)])
}
<bytecode: 0x13a1db6c0>
<environment: namespace:utils>
```