Much of my programming time in R is spent on finding a suitable package doing the task I am looking for. But usually it does the work almost but not quite. One of the main experiences I have had with tidyverse purrr is that I have stopped looking for these packages and just programmed them exactly as I want them.

One example is to make Table 1 of a publication. The first table of a medical publication is usually a table of descriptives such as patient characteristics and baseline data. A quick internet search such as “rct table 1 in R” reveals several pages and packages suited for the purpose. But none of them is exactly what I want, especially since it is so easy to do using purrr.

**Functions**

First we need to make some functions taking in a dataset, a variable in the dataset and a grouping variable, and returning a correctly formatted character string. I make two functions as an example, one returning the mean and standard deviation, and one a count with percentage. I also make one just returning empty strings for headers. Note the use of quasiquotation functions (ensym and !!). I am not very proefficient with such functions, but this works somehow.

mean\_sd <- function(data, var, group, digits = 1) {

var <- ensym(var)

group <- ensym(group)

data %>%

group\_by(!!group) %>%

summarise(mean = mean(!!var, na.rm = TRUE), sd = sd(!!var, na.rm = TRUE), missing = sum(is.na(!!var))) %>%

mutate\_at(vars(mean, sd), ~round(., digits = digits)) %>%

mutate(txt = paste0(mean, " (", sd, ")")) %>%

select(group, txt) %>%

deframe

}

n\_pct <- function(data, var, group, level = 1) {

var <- ensym(var)

group <- ensym(group)

data %>%

filter(!is.na(!!var)) %>%

group\_by(!!group, !!var) %>%

summarise(n = n()) %>%

group\_by(!!group) %>%

mutate(tot = sum(n),

pct = round(n/tot\*100,digits = 1)) %>%

mutate(txt = paste0(n, " (", pct, "%)")) %>%

filter(!!var == !!level) %>%

ungroup %>%

select(group, txt) %>%

deframe

}

empty <- function(data, var, group, ...){

group <- ensym(group)

data %>%

group\_by(!!group) %>%

summarise(n = n()) %>%

mutate(txt = "") %>%

select(group,txt) %>%

deframe

}

**Table style tibble**

Then we need a tibble defining how the table should look like. The first column is the variable description, the next is the function to be used, then the variable name, and finally other arguments for the function call.

dm\_table <- tribble(

~text, ~f, ~var, ~param,

"\*\*Age at enrollment (years)\*\*", "mean\_sd", "age", list(digits = 1),

"\*\*Symptom duration, n (%)\*\*", "empty", "", list(),

"Less than 1 day", "n\_pct", "sympdurcat", list(level = "<1"),

" 1-2 days", "n\_pct", "sympdurcat", list(level = "1-2"),

" 2-7 days", "n\_pct", "sympdurcat", list(level = ">2-7"),

"More than 7 days", "n\_pct", "sympdurcat", list(level = ">7"),

"\*\*Signs and symptoms\*\*", "empty", "", list(),

"Dysuria, n(%)", "n\_pct", "dysblcat", list(level = "Yes"),

"Urinary urgency, n(%)", "n\_pct", "urgblcat", list(level = "Abnormal"),

"Urinary frequency, n(%)", "n\_pct", "polblcat", list(level = "Abnormal"),

"Visible hematuria, n(%)", "n\_pct", "hemblcat", list(level = "Yes"),

"Symptom severity sum score\*", "mean\_sd", "symptotscore", list(digits = 2),

"Dysuria symptom score", "mean\_sd", "dysbl", list(digits = 2),

"Urinary urgency symptom score", "mean\_sd", "urgbl", list(digits = 2),

"Urinary frequency symptom score", "mean\_sd", "polbl", list(digits = 2)

)

We need a wrapper to the rlang::exec-function to forward the other arguments to the function call:

stats\_exec <- function(f, data, var, group, ...){

exec(f, data, var, group, !!!(...))

}

**Add data**

Then we need some data. I was involved in a trial on the use of antibiotics in women with urinary tract infection (UTI), the [IMUTI](https://www.icostatistics.no/publication/vik-2018/) trial. In this trial the raw data are included in the publication for transparency. The baseline data in .rds format is stored [here](https://www.icostatistics.no/datasets/imutibl.rds)

imutibl <- readRDS("static/datasets/imutibl.rds") %>%

filter(fas == "Yes")

**Purrr it!**

Then everything is ready to use purrr to make the demographics and baseline characteristics table. I add the data as a list-column, and then use pmap to map the data, variable, grouping variable and arguments to the function using the stats\_exec function. Then some unnesting, pivoting and kabelinig, and we have a nice table!

dm\_table <- dm\_table %>%

mutate(data = list(imutibl),

group = "group") %>%

mutate(res = pmap(list(f, data, var, group, param), stats\_exec)) %>%

mutate(id = map(res,names)) %>%

unnest(c(res, id)) %>%

pivot\_wider(values\_from = res, names\_from = id) %>%

select(text, levels(imutibl$group))

knitr::kable(dm\_table)

| **text** | **Ibuprofen** | **Pivmecillinam** |
| --- | --- | --- |
| **Age at enrollment (years)** | 28.1 (8.5) | 28.5 (10.2) |
| **Symptom duration, n (%)** |  |  |
| Less than 1 day | 10 (5.5%) | 9 (5.1%) |
| 1-2 days | 89 (49.2%) | 84 (47.2%) |
| 2-7 days | 80 (44.2%) | 84 (47.2%) |
| More than 7 days | 2 (1.1%) | 1 (0.6%) |
| **Signs and symptoms** |  |  |
| Dysuria, n(%) | 152 (95.6%) | 152 (95.6%) |
| Urinary urgency, n(%) | 157 (98.7%) | 154 (96.9%) |
| Urinary frequency, n(%) | 157 (98.7%) | 156 (98.1%) |
| Visible hematuria, n(%) | 71 (44.4%) | 63 (39.6%) |
| Symptom severity sum score\* | 12.56 (3.43) | 12.29 (3.65) |
| Dysuria symptom score | 3.92 (1.49) | 4 (1.48) |
| Urinary urgency symptom score | 4.42 (1.32) | 4.3 (1.46) |
| Urinary frequency symptom score | 4.21 (1.36) | 3.99 (1.41) |

Then we can add some flourish, e.g. the number in each group in the header and footnotes:

header <- imutibl %>%

group\_by(group) %>%

summarise(n=n()) %>%

mutate(txt = paste0(group, "\n (N=", n, ")")) %>%

select(txt) %>%

deframe

header <- c("Characteristic", header)

dm\_table %>% knitr::kable(col.names = header) %>%

footnote(symbol = "Sum of day 0 symptopm scores of dysuria, urinary urgency and urinary frequency, range 0-18. The symptoms were given a value on a scale from 0 to 6 where 0 was 'normal/not affected' and 6 was 'as bad as it could be'.",

general = "SD, standard deviation",

footnote\_as\_chunk = T)

| **Characteristic** | **Ibuprofen (N=181)** | **Pivmecillinam (N=178)** |
| --- | --- | --- |
| **Age at enrollment (years)** | 28.1 (8.5) | 28.5 (10.2) |
| **Symptom duration, n (%)** |  |  |
| Less than 1 day | 10 (5.5%) | 9 (5.1%) |
| 1-2 days | 89 (49.2%) | 84 (47.2%) |
| 2-7 days | 80 (44.2%) | 84 (47.2%) |
| More than 7 days | 2 (1.1%) | 1 (0.6%) |
| **Signs and symptoms** |  |  |
| Dysuria, n(%) | 152 (95.6%) | 152 (95.6%) |
| Urinary urgency, n(%) | 157 (98.7%) | 154 (96.9%) |
| Urinary frequency, n(%) | 157 (98.7%) | 156 (98.1%) |
| Visible hematuria, n(%) | 71 (44.4%) | 63 (39.6%) |
| Symptom severity sum score\* | 12.56 (3.43) | 12.29 (3.65) |
| Dysuria symptom score | 3.92 (1.49) | 4 (1.48) |
| Urinary urgency symptom score | 4.42 (1.32) | 4.3 (1.46) |
| Urinary frequency symptom score | 4.21 (1.36) | 3.99 (1.41) |
| *Note:* SD, standard deviation |  |  |
| \* Sum of day 0 symptopm scores of dysuria, urinary urgency and urinary frequency, range 0-18. The symptoms were given a value on a scale from 0 to 6 where 0 was ‘normal/not affected’ and 6 was ‘as bad as it could be’. |  |  |