Untitled

Chao Cheng

2022-09-21

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

knitr::all\_labels()

## [1] "setup" "cars" "pressure" "all\_lab1"   
## [5] "load\_child" "all\_lab2" "unnamed-chunk-1" "show\_child"   
## [9] "all\_lab3" "all\_lab4"

child\_res <- lapply(c("gen\_str\_output.Rmd", "form\_output.Rmd"),  
 knitr::knit\_child,   
 quiet = TRUE,   
 envir = environment())

knitr::all\_labels()

## [1] "setup" "cars" "pressure"   
## [4] "all\_lab1" "load\_child" "all\_lab2"   
## [7] "unnamed-chunk-1" "show\_child" "all\_lab3"   
## [10] "all\_lab4" "fun\_gen\_str\_output" "fun\_form\_output"

Gen\_Str\_Output(1 : 10, digit = 2)

## [1] "1.00" "2.00" "3.00" "4.00" "5.00" "6.00" "7.00" "8.00" "9.00"   
## [10] "10.00"

# appendix

## Details of predefined functions

Gen\_Str\_Output 用来产生数值结果的字符串:

* in\_num: 数值型向量
* digit：输出结果的小数点位数
* pct：输入的数值是否是百分数，默认为TRUE。当输入数值代表百分数时，0和100 将会直接输出，而不添加小数点和小数位数。

Gen\_Str\_Output\_Atom <- function(in\_num, digit = 1, pct = TRUE){  
 # Generate string output from numeric input  
 digit <- min(digit, 4)  
   
 if(pct){ # rule for `percentage` output  
 if((in\_num == 0) | (in\_num == 100)){  
 res <- as.character(in\_num)  
 }else{  
 res <- sprintf(paste0("%.", digit, "f"), in\_num)  
 }  
 }else{ # rule for other output  
 res <- sprintf(paste0("%.", digit, "f"), in\_num)  
 }  
 return(res)  
}  
  
Gen\_Str\_Output <- function(in\_num, digit = 1, pct = TRUE){  
 res <- mapply(Gen\_Str\_Output\_Atom, in\_num = in\_num, digit = digit, pct = pct)  
 return(res)  
}

Form\_Output 将总结好的结果转换为TFL中要求的按列呈现的形式

* df\_long: 待输出的数据，可参考Summary\_Perct的结果，主要需包含
* by\_var\_name：提供pivot\_wider时的names\_from。其内容一般是剂量组（字符串或factor），在最终结果表中是列名（A组、B组……）
* col\_name: 该列保存计数结果
* {col\_name}\_pct\_str: 格式处理过后的百分比数值，（字符串格式）。
* group\_var\_name: 若非空，说明df\_long中数据是按照(by\_var\_name, group\_var\_name)这样的双层结构进行计数的。一般该列内容是各分组结果，如原因1，原因2，……
* by\_var\_name, col\_name, {col\_name}\_pct\_str：已在之前解释

Form\_Output <- function(df\_long,   
 by\_var\_name = "arm\_fct",   
 col\_name = "trt\_num",   
 group\_var\_name = NULL,   
 out\_1st\_name = NULL,   
 out\_1st\_val = NULL){  
 res <- df\_long %>%  
 mutate(out\_str = str\_c(.data[[col\_name]],   
 "(",   
 .data[[glue::glue("{var\_name}\_pct\_str",   
 var\_name = col\_name)]],   
 "%)"))  
 if(is.null(group\_var\_name)){  
 res <- res %>%  
 select(all\_of(by\_var\_name), out\_str) %>%  
 pivot\_wider(names\_from = all\_of(by\_var\_name),   
 values\_from = out\_str)   
 if(!is.null(out\_1st\_name)){  
 res <- res %>%  
 mutate("{out\_1st\_name}" := out\_1st\_val, .before = 1)  
 }else{  
 res <- res %>%  
 mutate("{col\_name}" := " ", .before = 1)  
 }  
 }else{  
 grp\_lvls <- levels(df\_long %>% pull(all\_of(group\_var\_name)))  
   
 res <- res %>%  
 pivot\_wider(id\_cols = .data[[group\_var\_name]],   
 names\_from = all\_of(by\_var\_name),   
 values\_from = out\_str) %>%  
 arrange(factor(.data[[group\_var\_name]], levels = grp\_lvls)) %>% # 确保输出行的顺序与原始`group\_var\_name`的level一致  
 mutate("{group\_var\_name}" := as.character(.data[[group\_var\_name]]))  
 if(!is.null(out\_1st\_name)){  
 res <- res %>%  
 rename("{out\_1st\_name}" := all\_of(group\_var\_name))  
 }  
 }  
   
 return(res)  
}

knitr::all\_labels()

## [1] "setup" "cars" "pressure"   
## [4] "all\_lab1" "load\_child" "all\_lab2"   
## [7] "unnamed-chunk-1" "show\_child" "all\_lab3"   
## [10] "all\_lab4" "fun\_gen\_str\_output" "fun\_form\_output"

Gen\_Str\_Output\_Atom <- function(in\_num, digit = 1, pct = TRUE){  
 # Generate string output from numeric input  
 digit <- min(digit, 4)  
   
 if(pct){ # rule for `percentage` output  
 if((in\_num == 0) | (in\_num == 100)){  
 res <- as.character(in\_num)  
 }else{  
 res <- sprintf(paste0("%.", digit, "f"), in\_num)  
 }  
 }else{ # rule for other output  
 res <- sprintf(paste0("%.", digit, "f"), in\_num)  
 }  
 return(res)  
}  
  
Gen\_Str\_Output <- function(in\_num, digit = 1, pct = TRUE){  
 res <- mapply(Gen\_Str\_Output\_Atom, in\_num = in\_num, digit = digit, pct = pct)  
 return(res)  
}

knitr::all\_labels()

## [1] "setup" "cars" "pressure"   
## [4] "all\_lab1" "load\_child" "all\_lab2"   
## [7] "unnamed-chunk-1" "show\_child" "all\_lab3"   
## [10] "all\_lab4" "fun\_gen\_str\_output" "fun\_form\_output"