

Untitled

Chao Cheng

2022-09-21

目录

R Markdown	1
Including Plots	2
appendix	3
Details of predefined functions	3

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:

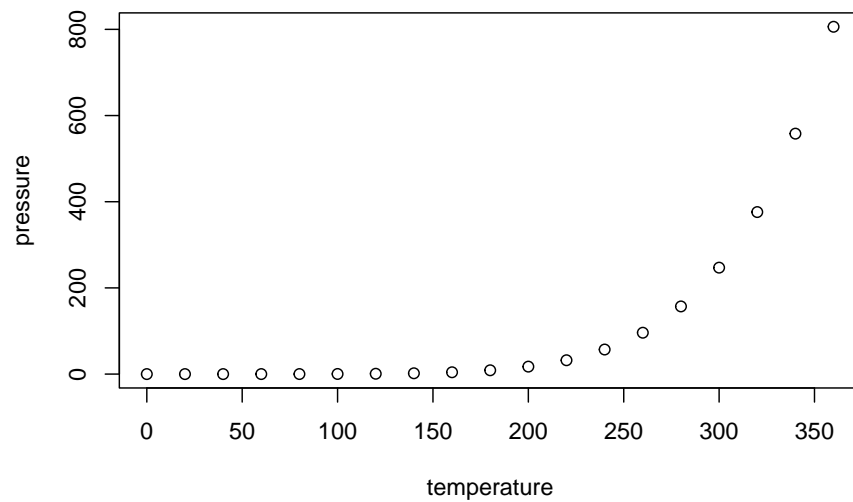
```
summary(cars)
```

```
##      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"
```

```
ls()
```

```
## character(0)
```

```

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"

ls()

## [1] "child_res"      "Form_Output"    "Gen_Str_Output"
## [4] "Gen_Str_Output_Atom"

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
      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"
```