

# Writing a simple R package in S3.

Ronald (Ryy) Glenn Thomas

2024-03-09

## Table of contents

|          |                     |          |
|----------|---------------------|----------|
| <b>1</b> | <b>Introduction</b> | <b>1</b> |
| <b>2</b> | <b>References</b>   | <b>8</b> |

## 1 Introduction

If you, like me, feel its time to expand your R programming armamentarium to include S3 methods. This blog may help.

Where to start?

In this post we'll walk through an example of a simple "table 1" function using S3 methods.

We'll start with the 'raw' data from a sample of the Penguins data set and return a dataframe with summary measures.

S3 methods allow coders to write functions that perform differently for different classes of objects.

In our project we want to build a function that creates a row in the 'Table 1' table for each variable in the formula regardless of the class of the variable.

Now reading Nick Tierney R journal paper.

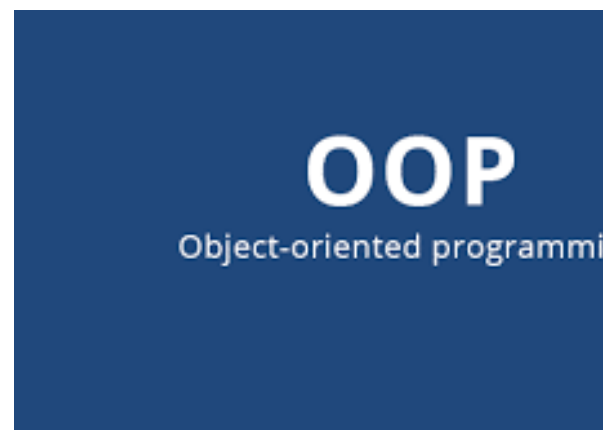


Figure 1: S3 OOP in R

```

source("~/shr/zz.tools.R")
library(pacman)

p_load(tidyverse, dplyr, gapminder, thematic, palmerpenguins, tidyverse, knitr, lubridate, readr)
knitr::opts_chunk$set(collapse = T)
set.seed(101)
dat <- palmerpenguins::penguins %>%
  filter(!is.na(sex))
dat1 <- slice_sample(dat, n=10) |>
  select(species, island, bill_length_mm)

```

```

#' Table one summaries
#'
#' Summarizes baseline trial results by treatment
#' @param data dataframe
#' @param form formula y ~ x1 + x2
#' @param ... extra parameters passed through to speciality functions
#' @return a dataframe
#' @examples
#' table1(dat2, form = arm ~ sex + age, annot = FALSE)

#' @export
table1 <- function(form, data, ...) {
  UseMethod("table1")
}

row_name <- function(x, nm, ...) {
  UseMethod("row_name")
}

row_name.character <- function(x, nm) {
  categs <- unique(na.omit(x))
  nms <- cbind(vars = c(nm, categs), code = c(1, rep(2, length(categs))))
  return(as.data.frame(nms))
}

row_name.factor <- row_name.character

row_name.logical <- row_name.character

```

```

row_name.numeric <- function(x, nm, ...) {
  return(as.data.frame(cbind(vars = nm, code = 3)))
}

row_summary <- function(x, yy) {
  UseMethod("row_summary")
}

row_summary.character <- function(x, yy) {
  df <- data.frame(x = x, y = yy) |> na.omit()
  t1 <- df |>
    tabyl(x, y) |>
    adorn_percentages("col") |>
    adorn_pct_formatting(digits = 0) |>
    adorn_ns(position = "front") |>
    select(-x)
  return(rbind("", t1))
}

row_summary.factor <- row_summary.character
row_summary.logical <- row_summary.character

row_summary.numeric <- function(x, yy) {
  sp <- split(x, yy)
  mm <- sp |>
    map_vec(mean, na.rm = TRUE) |>
    round(2)
  ss <- sp |>
    map_vec(sd, na.rm = TRUE) |>
    round(2) |>
    paste0("(", x = _, ")")
  out <- paste(mm, ss)
  names(out) <- names(sp)
  return(out)
}

row_pv <- function(x, yy) {
  UseMethod("row_pv")
}

```

```

row_pv.character <- function(x, yy) {
  tab <- data.frame(x = x, y = yy) |>
    na.omit() |>
    tabyl(x, y, show_missing_levels = FALSE)
  if (!(nrow(tab) >= 2 & ncol(tab) >= 3)) {
    return(NA)
  }
  pv <- janitor::fisher.test(tab, simulate.p.value = TRUE)$p.value |>
    round(4)
  return(c(pv, rep("", nrow(tab))))
}

row_pv.factor <- row_pv.character
row_pv.logical <- row_pv.character
row_pv.factor <- row_pv.character
row_pv.logical <- row_pv.character

row_pv.numeric <- function(x, yy) {
  df <- data.frame(x = x, y = yy)
  u <- unique(yy)
  categs <- u[!is.na(u)]
  if (!(length(categs) > 1)) {
    return(NA)
  }
  pv <- tidy(anova(lm(x ~ y, data = df)))$p.value[1] |>
    round(4)
  return(pv)
}

bygroup <- function(indep, dep) {
  left <- indep |>
    imap(row_name) |>
    bind_rows()
  right <- indep |>
    map(row_pv, yy = dep[[1]]) |>
    unlist() |>
    enframe(name = NULL) |>
    setNames("p.value")
  mid <- indep |>
    map(row_summary, yy = dep[[1]])

```

```

mid <- bind_rows(mid)
tab <- bind_cols(left, mid, right)
return(tab)
}
#' @export
#' @describeIn table1 interprets formula and yields publication tables
table1.formula <- function(form, data, ...) {
  args <- list(...)
  for (i in 1:length(args)) {
    assign(x = names(args)[i], value = args[[i]])
  }
  vars <- all.vars(form)
  y_var <- deparse(form[[2]])
  g_bar <- deparse(form[[c(3, 1)]])
  g_var <- NULL
  if (g_bar == "|") {
    x_vars <- all.vars(form[[c(3, 2)]])
    g_var <- all.vars(form[[c(3, 3)]])
    group <- data[g_var]
  } else {
    x_vars <- all.vars(form)[-1]
  }

  if (!is.null(g_var)) {
    dd <- split(data[x_vars], data[g_var])
    yy <- split(data[y_var], data[g_var])
    tab3 <- map2(dd, yy, ~ bygroup(indep = .x, dep = .y))
    tab4 <- bind_rows(tab3)
    # browser()
  } else {
    tab5 <- bygroup(indep = data[x_vars], dep = data[y_var])
  }

  if (!is.null(g_var)) {
    new <- data.frame(matrix(NA, nrow = length(tab3), ncol = ncol(tab4)))
    names(new) <- names(tab4)
    new$vars <- names(tab3)
    new$code <- 4
    # browser()
    rr <- cumsum(c(1, rep(nrow(tab3)[[1]]) + 1, length(tab3) - 1)))
    tab5 <- insertRows(tab4, rr, new, rcurr = F)
  }
}

```

```

# browser()
}
stripes <- map(unique(tab5$code), function(x) {
  which(tab5$code == x)
})

# browser()
myfcn <- function(x, i, theme = theme) {
  x <- x |> row_spec(stripes[[i]],
    color = theme$foreground[i],
    background = theme$background[i]
  )
}
browser()
tab5 = dplyr::mutate(tab5, vars = ifelse(code == 2, gsub("^", "\\quad ", vars), vars))
kk <- kbl(tab5[-2], "latex", booktabs = T, linesep = "",
  escape = F, digits = digits)
tablatex <- reduce(1:length(stripes), ~ myfcn(.x, .y, theme = theme), .init = kk)

write(tablatex, paste0("./tables/", fname, ".tex"))
system(paste0("sh ~/shr/figurize.sh ./tables/", fname, ".tex"))
# browser()

if (layout == "html") {
  stripes1 <- (1:length(var_categ_stripes))[which(var_categ_stripes == 1)]
  stripes2 <- (1:length(var_categ_stripes))[which(var_categ_stripes == 2)]
  stripes3 <- (1:length(var_categ_stripes))[which(var_categ_stripes == 3)]
  kableExtra::kbl(tab, "html") |>
    kable_paper() %>%
    row_spec(stripes1,
      bold = T, color = theme$color1,
      background = theme$background1
    ) |>
    row_spec(stripes2,
      bold = T, color = theme$color2,
      background = theme$background2
    ) |>
    row_spec(stripes3,
      bold = T, color = theme$color3,
      background = theme$background3
    )

```

```

    )
  }
  return(tab5)
}

p_load(atable, purrr, kableExtra, tibble, janitor, broom, palmerpenguins, dplyr)
options(knitr.kable.NA = "")
p1 <- sample_n(penguins, 300) |>
  dplyr::select(
    species, flipper_length_mm, sex,
    body_mass_g, bill_length_mm, island
  ) |>
  dplyr::mutate(flp = flipper_length_mm > 197)

theme_npg <- list(
  foreground = c("black", "black", "black", "black"),
  background = c("#E64B35B2", "#4DBBD5B2", "#00A087B2", "#3C5488B2", "#F39B7FB2")
)
theme_nejm <- list(
  foreground = c("black", "black", "black", "black"),
  background = c("#BC3C29FF", "#0072B5FF", "#E18727FF", "#20854EFF")
)

# tab1 <- table1(sex ~ flp + body_mass_g + bill_length_mm + island,
#   data = p1,
#   theme = theme_nejm, layout = "console", fname = "peng_table", digits = 3
# )
tab2 <- table1(sex ~ flp + body_mass_g + bill_length_mm | island,
  data = p1,
  theme = theme_nejm, layout = "console", fname = "peng_table", digits = 3
)
tab3 <- table1(sex ~ flp + body_mass_g + bill_length_mm | island,
  data = p1,
  theme = theme_nejm, layout = "latex", fname = "peng_table", digits = 3
)
tab4 <- table1(sex ~ flp + body_mass_g + bill_length_mm | island,
  data = p1,
  theme = theme_nejm, layout = "html", fname = "peng_table", digits = 3
)

```

## 2 References

Also useful other references:

Introduction to Scientific Programming and Simulation using  
R. Jones. Maillardet, Robinson.

[1608.07161] A Simple Guide to S3 Methods <https://arxiv.org/abs/1608.07161>

Why your S3 method isn't working | R-bloggers

Dealing with S3 methods in R with a simple example | R-  
bloggers

Video on S3 Classes in R by Dr Andrew Robinson | R-  
bloggers

Unexported S3 Methods and R Packages | R-bloggers

Simple Guide to S3 Methods | R-bloggers

The S3 OOP system | R-bloggers

Nick Tierney R journal paper.