

Writing a simple R package in S3.

Ronald (Ryy) Glenn Thomas

2024-02-29

Table of contents

| | | |
|----------|---------------------|----------|
| 1 | Introduction | 1 |
| 2 | References | 6 |

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.

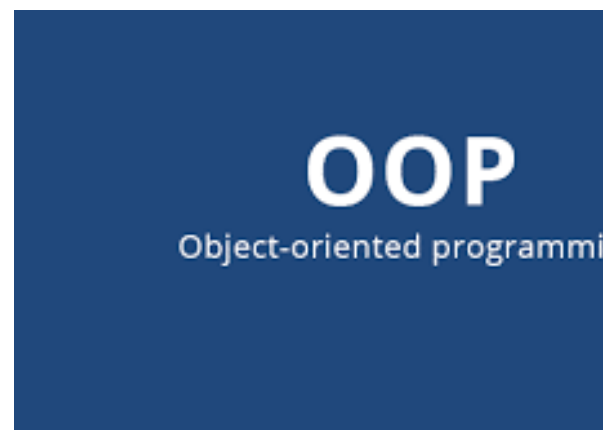


Figure 1: S3 OOP in R

```

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, ...) {
  # browser()
  u <- unique(x)
  cats <- paste(" ", u[!is.na(u)])
  return(c(nm, cats))
}

row_name.factor <- row_name.character

```

```

row_name.logical <- row_name.character

row_name.numeric <- function(x, nm, ...) {
  return(nm)
}

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

row_summary.character <- function(x) {
  # browser()
  df <- data.frame(x = x, y = dep) |> na.omit()
  t1 <- df |>
    tabyl(x, y) |>
    adorn_percentages("col") |>
    adorn_pct_formatting(digits = 0) |>
    adorn_ns(position = "front") |>
    select(-x)
  # browser()
  t1 <- as_tibble(t1)
  t2 <- table(df$x, df$y) |> as.data.frame.matrix()
  rbind("", t1)
}

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

row_summary.numeric <- function(x) {
  sp <- split(x, dep)
  nms <- names(sp)
  mm <- sp |>
    map_vec(mean, na.rm = TRUE) |>
    round(2) |>
    as.character() |>
    matrix(1)
  ss <- sp |>
    map_vec(sd, na.rm = TRUE) |>
    round(2) |>
    paste0("(", x = _, ")") |>

```

```

    matrix(1)
    bb <- paste(unlist(mm), unlist(ss)) |> matrix(nrow = nrow(mm))
    colnames(bb) <- nms
    bb <- bb |> as_tibble()
    bb
  }

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

row_pv.character <- function(x) {
  tab <- data.frame(x = x, y = dep) |>
    na.omit() |>
    tabyl(x, y)
  if (!(nrow(tab) >= 2 & ncol(tab) >= 2)) {
    return(NA)
  }
  # browser()
  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.numeric <- function(x) {
  df <- data.frame(x = x, y = dep)
  tab <- table(x, dep)
  pv <- ifelse((nrow(tab) >= 2 & ncol(tab) >= 2),
    stats::fisher.test(tab, simulate.p.value = TRUE)$p.value, NA
  ) |>
    round(4)
  return(c(pv, rep("", nrow(tab))))
}

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

```

```

row_pv.numeric <- function(x) {
  df <- data.frame(x = x, y = dep)
  pv <- tidy(anova(lm(x ~ y, data = df)))$p.value[1] |>
    round(4)
  return(pv)
}

#' @export
#' @describeIn table1 interprets formula and yields publication tables
table1.formula <- function(form, data, ...) {
  vars <- all.vars(form)
  # dep <- data[[vars[1]]]
  indep <- data[vars[-1]]

  y_var <- deparse(form[[2]])
  dep <- data[y_var]
  g_bar <- form[[c(3, 1)]]
  if (g_bar == "|") {
    x_vars <- all.vars(form[[c(3, 2)]]])
    g_vars <- all.vars(form[[c(3, 3)]]])
    group <- data[g_vars]
  } else {
    x_vars <- all.vars(form)[-1]
  }
  indep <- data[x_vars]
  browser()
  left <- indep |>
    imap(row_name, ...) |>
    unlist() |>
    enframe(name = NULL) |>
    setNames("variable")
  right <- indep |>
    map(row_pv) |>
    unlist() |>
    enframe(name = NULL) |>
    setNames("p-value")
  mid <- indep |>
    map_dfr(row_summary) |>
    identity()
  mid <- bind_rows(mid)

```

```

# browser()
bind_cols(left, mid, right)
}

p_load(palmerpenguins, dplyr)
p1 <- sample_n(penguins, 300) |>
  dplyr::select(species, sex, body_mass_g, island)
# p1 <- pp |> dplyr::mutate(sex1 = sex)
# table1(species ~ sex + body_mass_g, data = p1)
  table1(species ~ sex + body_mass_g | island, data = p1)

```

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.