Writing a simple R package in S3.

Ronald (Ryy) Glenn Thomas

2024-04-22

Table of contents

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.

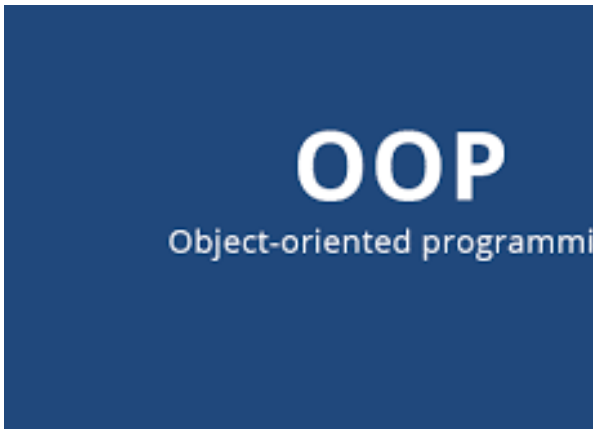

Figure 1: S3 OOP in R

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

p_load(tidyverse, dplyr, gapminder, thematic, palmerpenguins, tidyverse, knitr, lubridate, reac
knitr::opts_chunk$set(collapse = T)
set.seed(101)
dat <- palmerpenguins::penguins %>%
  fil(!is.na(sex))
dat1  <- slice_sample(dat, n=10) |>
sel(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, missing = FALSE, ...) {
  if (missing) {
    categs <- unique(as.character(x))
    categs[is.na(categs)] <- "missing"
  } else {
    categs <- unique(na.omit(as.character(x)))
  }
  nms <- cbind(variables = 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(variables = nm, code = 3)))
}

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

row_summary.character <- function(x, yy, totals = FALSE, missing = FALSE, ...) {
  df <- data.frame(x = x, y = yy)
  if (missing) {
    t1 <- df |> tabyl(x, y, show_na = TRUE, show_missing_levels = FALSE)
  } else {
    t1 <- df |>
      na.omit() |>
      tabyl(x, y, show_missing_levels = FALSE)
  }
  if (totals) {
    t1 <- t1 |> adorn_totals("col")
  }
  t1 <- t1 |>
    adorn_percentages("col") |>
    adorn_pct_formatting(digits = 0) |>
    adorn_ns(position = "front") |>
    select(-x)
  names(t1) <- gsub("_", "", names(t1))
  return(rbind("", t1))
}

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

row_summary.numeric <- function(x, yy, totals = FALSE, missing = FALSE, ...) {
  if (missing) {
    zz <- as.character(yy)
```

```
    zz[is.na(zz)] <- "NA"
    yy <- factor(zz)
  }
  sp <- split(x, yy)
  # add x to sp as a "k+1" element listing if totals=T
  if (totals) sp[["Total"]] <- x
  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, missing = FALSE, ...) {
  if (missing) {
    categs <- unique(as.character(x))
  } else {
    categs <- unique(na.omit(as.character(x)))
  }
  tab <- data.frame(x = x, y = yy) |>
    na.omit() |>
    tabyl(x, y, show_missing_levels = FALSE)
  if (!(nrow(tab) >= 2 & ncol(tab) >= 3)) {
    pv <- NA
  } else {
    pv <- janitor::fisher.test(tab, simulate.p.value = TRUE)$p.value |>
      round(4)
  }
  return(c(pv, rep("", length(categs))))
}
```

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

block <- function(indep, dep, grp, ...) {
  dd <- split(indep, grp)
  yy <- split(dep, grp)
  # print('inside block')
  tab3 <- map2(dd, yy, function(x, y) {
    build(indep = x, dep = y, ...)
  })
  tab4 <- bind_rows(tab3)
  new <- data.frame(matrix(NA, nrow = length(tab3), ncol = ncol(tab4)))
  names(new) <- names(tab4)
  new$variables <- names(tab3)
  new$code <- 4
  rr <- cumsum(c(1, rep(nrow(tab3[[1]]) + 1, length(tab3) - 1)))
  tab5 <- insertRows(tab4, rr, new, rcurr = F)
}



stripes <- function(tab5, digits = 3, theme = theme_bw, ...) {
  tab5 <- dplyr::mutate(tab5, variables = ifelse(code == 2,
    gsub("^", "\\\\quad ", variables), variables
  ))
  # tab6$vars = gsub("_","\_",tab6$vars)
  strp <- map(sort(unique(tab5$code)), function(x) {
```

```
    which(tab5$code == x)
  })
  myfcn <- function(x, i, theme = theme) {
    x <- x |> row_spec(strp[[i]],
      color = theme$foreground[i],
      background = theme$background[i]
    )
  }
  tab5 <- tab5 |>
    dplyr::select(-code)
  kk <- kbl(tab5, "latex",
    booktabs = T, linesep = "",
    escape = F, digits = digits
  )
  tab5plusstripes <- reduce(1:length(strp),
    ~ myfcn(.x, .y, theme = theme),
    .init = kk
  )
  return(tab5plusstripes)
}



build <- function(indep, dep, size = TRUE, ...) {
  # args <- list(...)
  # for (i in 1:length(args)) {
  #   assign(x = names(args)[i], value = args[[i]])
  # }
  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]], ...) |>
    bind_rows()
  if (size) {
```

```
    names(mid) <- paste0(names(mid), " (N = ", table(dep), ")")
  }
  tab <- bind_cols(left, mid, right)
  return(tab)
}
#' @export
#' @describeIn table1 interprets formula and yields publication tables
table1.formula <- function(form, data, pvalue = TRUE, totals = FALSE,
                           fname = "table1", layout = "console", ...) {
  # args <- list(...)
  # for (i in 1:length(args)) {
  #   assign(x = names(args)[i], value = args[[i]])
  # }
  # 1

  # need to figure out what to do when only when indep var and no group.
  # form[[c(3,1)]]
  formtest <- as.character(form)
  grptest <- str_detect(formtest, "\\|") |> any()
  vars <- all.vars(form)
  y_var <- deparse(form[[2]])
  g_bar <- 0
  if (grptest) 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)) {
    tab5 <- block(indep = data[x_vars], dep = data[y_var], grp = data[g_var], ...)
  } else {
    tab5 <- build(indep = data[x_vars], dep = data[y_var], ...)
  }
  if (!pvalue) {
    tab5 <- tab5 |>
      dplyr::select(-p.value)
  }
```

```
  if (!totals) {
    tab5 <- tab5 |>
      dplyr::select(-contains("Total"))
  }
  if (is.null(y_var)) {
    tab5 <- tab5 |>
      dplyr::select(variables, code, contains("Total"), p.value)
  }
  if (layout == "console") {
    return(tab5[-2])
  } else if (layout == "latex") {
    tablatex <- stripes(tab5, ...)
    write(tablatex, paste0("./tables/", fname, ".tex"))
    system(paste0("sh ~/shr/figurize.sh ./tables/", fname, ".tex"))
  } else if (layout == "html") {
    kk <- kbl(tab5[-2], "html",
      escape = F, digits = digits
    )
    tabhtml <- reduce(1:length(stripes),
      ~ myfcn(.x, .y, theme = theme),
      .init = kk
    )
  }
}
```

2 References

Also useful other references:

Introduction to Scientific Programming and Simulation using R. Jomes. 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.