

Exploratory Data Analysis

Yijia Jiang

2022-11-30

1. Data Overview

In this study, the dataset we used is called ‘colon’, which is found in the ‘survival’ R package. These are data from one of the first successful trials of adjuvant chemotherapy for colon cancer. Levamisole is a low-toxicity compound previously used to treat worm infestations in animals; 5-FU is a moderately toxic (as these things go) chemotherapy agent. There are two records per person, one for recurrence and one for death. There are 1858 observations and 16 variables.

- id: patient id
- study: 1 for all patients
- rx: treatment - Obs(ervation), Lev(amisole), Lev(amisole)+5-FU
- sex: sex (0 = Female, 1 = Male)
- age: observation age in years
- obstruct: obstruction of colon by tumour (0 = No, 1 = Yes)
- perfor: perforation of colon (0 = No, 1 = Yes)
- adhere: adherence to nearby organs (0 = No, 1 = Yes)
- nodes: number of lymph nodes with detectable cancer
- time: days until event or censoring
- status: censoring status (0 = Censored, 1 = Event)
- differ: differentiation of tumour (1 = Well, 2 = Moderate, 3 = Poor)
- extent: extent of local spread (1 = Submucosa, 2 = Muscle, 3 = Serosa, 4 = Contiguous structures)
- surg: time from surgery to registration (0 = Short, 1 = Long)
- node4: more than 4 positive lymph nodes (0 = No, 1 = Yes)
- etype: event type (1 = recurrence, 2 = death)

The primary endpoints are the death of patients and the recurrence of patients. The type of censoring is right censoring, which means patients left the study before their death.

```
data(cancer, package = "survival")
colon_tb = as_tibble(colon)
colon_tb = colon_tb

colon_tb$status <- factor(colon_tb$status, levels = c(0,1), labels = c("Censored", "Event"))
colon_tb$sex <- factor(colon_tb$sex, levels = c(0,1), labels = c("Female", "Male"))
colon_tb$obstruct <- factor(colon_tb$obstruct, levels = c(0,1), labels = c("No", "Yes"))
colon_tb$perfor <- factor(colon_tb$perfor, levels = c(0,1), labels = c("No", "Yes"))
colon_tb$adhere <- factor(colon_tb$adhere, levels = c(0,1), labels = c("No", "Yes"))
colon_tb$differ <- factor(colon_tb$differ, levels = c(1,2,3), labels = c("Well", "Moderate", "Poor"))
colon_tb$extent <- factor(colon_tb$extent, levels = c(1,2,3,4),
                          labels = c("Submucosa", "Muscle", "Serosa", "Contiguous structures"))
```

```

colon_tb$surg <- factor(colon_tb$surg, levels = c(0,1), labels = c("Short", "Long"))
colon_tb$node4 <- factor(colon_tb$node4, levels = c(0,1), labels = c("No", "Yes"))

label(colon_tb$rx) <- "Treatment"
label(colon_tb$sex) <- "Sex"
label(colon_tb$age) <- "Age"
label(colon_tb$obstruct) <- "Obstruction of colon by tumour"
label(colon_tb$perfor) <- "Perforation of colon"
label(colon_tb$adhere) <- "Adherence to nearby organs"
label(colon_tb$status) <- "Censoring status"
label(colon_tb$differ) <- "Differentiation of tumour"
label(colon_tb$extent) <- "Extent of local spread"
label(colon_tb$surg) <- "Time from surgery to registration"
label(colon_tb$node4) <- "More than 4 positive lymph nodes"

units(colon_tb$age) <- "years"
units(colon_tb$time) <- "days"

my.render.cont <- function(x) {
  with(stats.apply.rounding(stats.default(x), digits = 3),
    c("", "Mean (SD)" = sprintf("%s (&plusmn;%s)", MEAN, SD)))
}
my.render.cat <- function(x) {
  c("", sapply(stats.default(x), function(y) with(y,
    sprintf("%d (%0.1f%%)", FREQ, PCT))))
}

# Baseline characteristic table for event death (with status)
output1 = table1(~ sex + age + obstruct + perfor + adhere + differ + extent + surg + node4 | status*rx,
  data = colon_tb %>% filter(etype == 2),
  render.continuous = my.render.cont,
  render.categorical = my.render.cat,
  render.missing = NULL, overall = FALSE)

# Baseline characteristic table for event recurrence (with status)
output2 = table1(~ sex + age + obstruct + perfor + adhere + differ + extent + surg + node4 | status*rx,
  data = colon_tb %>% filter(etype == 1),
  render.continuous = my.render.cont,
  render.categorical = my.render.cat,
  render.missing = NULL, overall = FALSE)

# Baseline characteristic table for event death (without status)
output3 = table1(~ sex + age + obstruct + perfor + adhere + differ + extent + surg + node4 | rx,
  data = colon_tb %>% filter(etype == 1),
  render.continuous = my.render.cont,
  render.categorical = my.render.cat,
  render.missing = NULL, overall = FALSE)

# Baseline characteristic table for event recurrence (without status)
output4 = table1(~ sex + age + obstruct + perfor + adhere + differ + extent + surg + node4 | rx,
  data = colon_tb %>% filter(etype == 1),
  render.continuous = my.render.cont,
  render.categorical = my.render.cat,

```

```

        render.missing = NULL, overall = FALSE)

save_table1 = function(output, file_name="temp.html"){
  css_path = system.file("table1_defaults_1.0/table1_defaults.css", package = "table1")
  css = sprintf('<style type="text/css">@import url(%s);</style>', css_path)
  cat(paste(css, output), file = file_name)
}

save_table1(output1, "./table/table1.html")
save_table1(output2, "./table/table2.html")
save_table1(output3, "./table/table3.html")
save_table1(output4, "./table/table4.html")

system("wkhtmltopdf --enable-local-file-access ./table/table1.html ./table/table1.pdf")
system("wkhtmltopdf --enable-local-file-access ./table/table2.html ./table/table2.pdf")
system("wkhtmltopdf --enable-local-file-access ./table/table3.html ./table/table3.pdf")
system("wkhtmltopdf --enable-local-file-access ./table/table4.html ./table/table4.pdf")

```

2. Overall Association Between Treatment Groups

```

colon_death = colon %>%
  subset(etype == 2) %>%
  mutate(death = ifelse(status == 1, 1, 0)) %>%
  mutate(tt_death = time, rx = as.factor(as.numeric(rx))) %>%
  select(id, rx, tt_death, death)

colon_recur = colon %>%
  subset(etype == 1) %>%
  mutate(recur = ifelse(status == 1, 1, 0)) %>%
  mutate(tt_recur = time, rx = as.factor(as.numeric(rx))) %>%
  select(id, rx, tt_recur, recur)

colon_ass = merge(colon_death, colon_recur, by = c("id", "rx")) %>%
  mutate(comp = ifelse(death == 1 || recur == 1, 1, 0)) %>%
  dplyr::arrange(id)

colon_ass$tt_comp = rje::rowMins(as.matrix((colon_ass[,c("tt_death", "tt_recur")])))

```

Right now, I failed to output the table using R code. Thus, it will be represented in another sas file.