6. Now you try: a clinical data project

Table 1, figures, and exports from the Mayo Clinic PBC study

Contents

The clinical study . . . 1
Your brief . . . 2
Setup . . . 2
Task 1: check the cohort . . . 2
Task 2: build Table 1 . . . 3
Task 3: make a bilirubin histogram . . . 5
Task 4: show baseline stage by treatment . . . 5
Task 5: export your report assets . . . 6
Download files from the browser . . . 7
Check your work . . . 8

Put sections 2–5 together with a **different, published clinical dataset**. You are preparing a short baseline report for a research meeting. Your question is: **Who entered the Mayo Clinic PBC study, and how were their baseline characteristics distributed across randomized treatment groups?**

Allow **35–45 minutes**, after the main seminar or as a take-home project. Try each task before opening its hint or worked solution. There is room for your own labels, colours, and layout; the checks focus on the data and meaning of the outputs.

The clinical study

The data come from Mayo Clinic studies of **primary biliary cholangitis (PBC)** conducted in 1974–1984. The older paper uses the name "primary biliary cirrhosis": Dickson et al. (1989), *Prognosis in primary biliary cirrhosis: Model for decision making*. The public data are distributed as `survival::pbc`; see the official dataset documentation.

The full file has 418 patients: 312 randomized participants and 106 patients who were not randomized. **Use only the 312 randomized patients** for this project. The supplied baseline CSV already applies that restriction and has one row per patient. The data notes and preparation rules explain the source, recoding, and missing values.

You will create new teaching outputs from the released baseline data. The paper develops a survival model; this project describes the trial cohort. You do not need to reproduce its tables or fit a survival model.

Column	Meaning
`patient_id`	Source case identifier; use for checks, not as a summary variable.
`treatment`	Randomized assignment: D-penicillamine or Placebo.
`age`	Age at entry, years.
`sex`	Recorded sex: Female or Male.

Column	Meaning
`bilirubin`	Baseline serum bilirubin, mg/dL.
`albumin`	Baseline serum albumin, g/dL.
`cholesterol`	Baseline serum cholesterol, mg/dL; some values are missing.
`stage`	Baseline histologic disease stage, coded 1–4.

Your brief

Prepare these deliverables:

1. A **Table 1** by randomized treatment, with an overall column, available counts, readable units, and visible missingness.
2. **Figure 1:** a histogram of baseline bilirubin, with a stated bin width.
3. **Figure 2:** proportional bars of baseline histologic stage by treatment, with group sizes and a percentage axis.
4. Both figures as **300-dpi PNGs and vector SVGs**, plus an editable RTF table.
5. Two or three sentences describing the cohort and one limitation.

Use Table 1, ggplot2, and exporting as references. The final checklist is your definition of done.

Setup

This page has a fresh R session. Run the cells in order; objects created here stay available to later cells on this page. The blank workspaces and the worked solutions use the same session. If you change an earlier object, rerun the later code that uses it.

```
library(gtsummary)
library(ggplot2)

pbc <- read.csv("data/pbc_baseline.csv", na.strings = c("", "NA"))
pbc$treatment <- factor(pbc$treatment,
                        levels = c("D-penicillamine", "Placebo"))
pbc$sex <- factor(pbc$sex, levels = c("Female", "Male"))
pbc$stage <- factor(pbc$stage, levels = 1:4,
                    labels = c("I", "II", "III", "IV"))

str(pbc)
```

Working locally? Download the CSV into `data/` inside your R project and run the same code. Install `gtsummary`, `gt`, `ggplot2`, and `svglite` once if needed. The complete local solution script writes its outputs to `output/`; keep it for after your first attempt.

Task 1: check the cohort

Before making anything, check the row count, unique patient IDs, treatment counts, and missing values. Why should the number of rows equal the number of distinct IDs here? Which variable has fewer available observations?

```
# Your workspace: inspect pbc, its patient IDs, groups, and missingness.
```

Hint: checking the import

Use `nrow()`, `anyDuplicated()`, `table()`, and `colSums(is.na(...))`. Use `useNA = "ifany"` when counting categories so missing groups are visible.

Worked solution: cohort checks

```
nrow(pbc)
anyDuplicated(pbc$patient_id)
table(pbc$treatment, useNA = "ifany")
colSums(is.na(pbc))
```

Expect **312 patients**, no duplicated IDs (`anyDuplicated()` returns 0), **158** assigned D-penicillamine and **154** assigned placebo. Cholesterol is missing for **28 patients**, leaving **284 available measurements**. The other teaching variables are complete. Each row is a baseline patient, not a visit.
Keep all 312 patients in Table 1. Applying `na.omit(pbc)` to the entire dataset would unnecessarily remove patients from summaries of their recorded age, bilirubin, and disease stage.

Task 2: build Table 1

Save the calculations as `tbl`, then the styled table as `table_gt`.

- Group by `treatment`; include age, sex, bilirubin, albumin, cholesterol, and stage. Treat stage as categorical.
- Use **median (Q1, Q3)** for continuous measurements and **n (%)** for categorical variables. State that percentages are within treatment columns.
- Add the overall column and available N for each variable. Show missing counts, including cholesterol's 28 missing measurements.
- Add a title, units, and a source note. Keep the table descriptive.

```
# Your workspace: build tbl, then convert and style it as table_gt.
# Start with tbl_summary(pbc, by = treatment, include = ...).
```

Hint: adapting the earlier table

Set `statistic = list(all_continuous() ~ "{median} ({p25}, {p75})")`. Use `type = list(stage ~ "categorical", sex ~ "categorical")` so both sex levels and all four stages appear. Chain `add_overall()` and `add_n()` before converting with `as_gt()`.

Worked solution: Table 1

```
tbl <- tbl_summary(
  pbc,
  by = treatment,
  include = c(age, sex, bilirubin, albumin, cholesterol, stage),
  type = list(sex ~ "categorical", stage ~ "categorical"),
  label = list(
    age ~ "Age, years",
    sex ~ "Recorded sex",
    bilirubin ~ "Serum bilirubin, mg/dL",
    albumin ~ "Serum albumin, g/dL",
    cholesterol ~ "Serum cholesterol, mg/dL",
    stage ~ "Histologic stage"
  ),
  statistic = list(
    all_continuous() ~ "{median} ({p25}, {p75})",
    all_categorical() ~ "{n} ({p}%)"
  ),
  digits = list(all_continuous() ~ 1),
  percent = "column",
  missing = "ifany",
  missing_text = "Missing",
  missing_stat = "{N_miss}"
) |>
  add_overall() |>
  add_n(col_label = "**Available N**") |>
  bold_labels() |>
  modify_header(label ~ "**Characteristic**")

table_gt <- as_gt(tbl) |>
  gt::tab_header(
    title = "Table 1. Mayo Clinic PBC trial participants",
    subtitle = "Baseline characteristics by randomized treatment"
  ) |>
  gt::opt_row_striping() |>
  gt::tab_options(table.font.size = gt::px(13), heading.align = "left") |>
  gt::tab_source_note(
    "Median (Q1, Q3) or n (%). Percentages use non-missing values within each
     ↪  column."
  ) |>
  gt::tab_source_note(
    "Source: survival::pbc; Dickson et al. (1989). Teaching analysis of 312
     ↪  randomized patients."
  )

table_gt
```

The headers count all patients in each arm; available N can differ by variable. For cholesterol, available N plus missing N should equal 312 in the overall column. Stage is a category, even though the source uses numbers to record it.

We omit baseline p-values: this report describes entry characteristics. A baseline comparison alone does not answer whether treatment improves outcomes.

Task 3: make a bilirubin histogram

Save a plot called `bilirubin_hist`. Show baseline bilirubin in mg/dL on the x-axis and patient counts on the y-axis. Choose a bin width, give the figure a title, and include the source and number of available observations in the caption. Try a second bin width and describe what changes.

```
# Your workspace: create and display bilirubin_hist.
# Use geom_histogram() and make the bin width an explicit choice.
```

Hint: figure-specific missing values

Subset only on the variable used in this plot: `bilirubin_data <- pbc[!is.na(pbc$bilirubin), ]`. Map `x = bilirubin` and try `binwidth = 1`. Keep the full observed range.

Worked solution: bilirubin distribution

```
bilirubin_data <- pbc[!is.na(pbc$bilirubin), ]

bilirubin_hist <- ggplot(bilirubin_data, aes(x = bilirubin)) +
  geom_histogram(binwidth = 1, boundary = 0,
                 fill = "#17324D", colour = "white") +
  labs(
    title = "Baseline serum bilirubin",
    subtitle = "Mayo Clinic PBC trial | Bin width: 1 mg/dL",
    x = "Serum bilirubin (mg/dL)", y = "Number of patients",
    caption = paste0(
      "Source: survival::pbc; Dickson et al. (1989). Teaching analysis.\n",
      "Available bilirubin: n = ", nrow(bilirubin_data),
      "; missing: ", sum(is.na(pbc$bilirubin)), "."
    )
  ) +
  theme_classic(base_size = 12) +
  theme(plot.caption = element_text(hjust = 0))

bilirubin_hist
```

All 312 patients have bilirubin recorded. The right-skewed distribution gives us a reason to prefer median and quartiles over mean and SD in this project. Changing the bin width changes the visual grouping, not the underlying values.

Task 4: show baseline stage by treatment

Save a second plot called `stage_plot`. Make one bar per randomized treatment, with histologic stages I–IV filling the bar. Each bar should total 100%. Put the number of patients contributing to each bar in its axis label, keep stages in order, and provide a readable legend and source caption.

```
# Your workspace: create and display stage_plot.
# Check table(pbc$stage, pbc$treatment) before drawing the bars.
```

Hint: counts versus proportions

Use `aes(x = treatment, fill = stage)` and `geom_bar(position = "fill")`. The y-axis values are proportions from 0 to 1; format them as percentages. Count patients after excluding any missing stage values, so the label matches the bar's denominator.

Worked solution: stage proportions

```
stage_data <- pbc[!is.na(pbc$stage) & !is.na(pbc$treatment), ]
stage_n <- table(stage_data$treatment)
stage_labels <- setNames(
  paste0(names(stage_n), "\n(n = ", as.integer(stage_n), ")"),
  names(stage_n)
)

stage_plot <- ggplot(stage_data, aes(x = treatment, fill = stage)) +
  geom_bar(position = "fill", width = 0.65, colour = "white") +
  scale_fill_manual(values = c(
    "I" = "#E1ECEB", "II" = "#A2C6C2",
    "III" = "#507E89", "IV" = "#17324D"
  ), drop = FALSE) +
  scale_x_discrete(labels = stage_labels) +
  scale_y_continuous(
    labels = function(x) paste0(round(100 * x), "%"),
    breaks = seq(0, 1, 0.25),
    expand = expansion(mult = c(0, 0.02))
  ) +
  labs(
    title = "Histologic stage at study entry",
    subtitle = "Mayo Clinic PBC trial | Baseline measurements",
    x = NULL, y = "Patients with recorded stage (%)", fill = "Stage",
    caption = paste0(
      "Source: survival::pbc; Dickson et al. (1989). Teaching analysis.\n",
      "Within-arm percentages; ", nrow(pbc) - nrow(stage_data),
      " patients excluded for missing stage or treatment."
    )
  ) +
  theme_classic(base_size = 12) +
  theme(legend.position = "bottom", plot.caption = element_text(hjust = 0))

stage_plot
```

Check the source counts: stages I–IV contain **12, 35, 56, 55** patients in the D-penicillamine arm, and **4, 32, 64, 54** in the placebo arm. There are no missing stages in this cohort. Your bars use denominators 158 and 154.
These are **baseline** measurements. Differences between these bars cannot demonstrate a response to treatment.

Task 5: export your report assets

Use your `table_gt`, `bilirubin_hist`, and `stage_plot` objects. Export the table as RTF for Word. Save **each figure as a PNG and an SVG**, using distinct filenames, a white background, and a final size of **7 × 5 inches**. Set the PNGs to **300 dpi** and always pass the intended `plot` explicitly.

```
# Your workspace: create output/ and save your table and both figures.
# Use writeLines(gt::as_rtf(...), ...) and ggplot2::ggsave(...).
```

Hint: exporting both figures

For PNG use `device = grDevices::png`; for SVG use `device = svglite::svglite`. A 7 × 5 inch PNG at 300 dpi should contain **2100 × 1500 pixels**. SVG does not need a DPI setting for these vector plots.

Worked solution: save the table and figures

```
dir.create("output", showWarnings = FALSE)
writeLines(gt::as_rtf(table_gt), "output/pbc_table1.rtf")
writeLines(gt::as_raw_html(table_gt), "output/pbc_table1.html")

plots <- list(pbc_bilirubin = bilirubin_hist, pbc_stage = stage_plot)
for (name in names(plots)) {
  ggsave(
    paste0("output/", name, ".png"), plot = plots[[name]],
    device = grDevices::png, width = 7, height = 5, units = "in",
    dpi = 300, bg = "white"
  )
  ggsave(
    paste0("output/", name, ".svg"), plot = plots[[name]],
    device = svglite::svglite, width = 7, height = 5, units = "in",
    bg = "white"
  )
}

list.files("output", pattern = "^pbc_")
```

The loop applies the same settings to each named plot. You can also write four separate `ggsave()` calls, following section 5.

Download files from the browser

Browser R writes to its own temporary filesystem. After running your export code, run this supplied helper to get download links. If you chose other filenames, update the paths below. Locally, the files are already in `output/`.

```
download_button <- function(path) {
  mime <- switch(tools::file_ext(path),
                 rtf = "application/rtf", html = "text/html",
                 png = "image/png", svg = "image/svg+xml",
                 "application/octet-stream")
  if (!file.exists(path)) stop("Export this file first: ", path)
  htmltools::tags$a(
    href = paste0("data:", mime, ";base64,", base64enc::base64encode(path)),
    download = basename(path), class = "btn btn-primary",
    paste("Download", basename(path))
  )
}
```

```
"Download helper ready."
```

```
htmltools::tagList(
  download_button("output/pbc_table1.rtf"),
  download_button("output/pbc_bilirubin.png"),
  download_button("output/pbc_bilirubin.svg"),
  download_button("output/pbc_stage.png"),
  download_button("output/pbc_stage.svg")
)
```

Download your files before leaving or refreshing this page. Open the RTF in Word and insert a saved PNG or SVG into a report or slide. Check the labels at their intended size; compare the saved image with your browser plot.

Check your work

- ☐ Table 1 includes 312 patients overall, with treatment Ns of 158 and 154.
- ☐ Cholesterol shows available N = 284 and missing N = 28 overall.
- ☐ Stage is categorical; continuous measurements have units and defined statistics.
- ☐ Both figures have titles, readable axes, a source, and the correct sample sizes.
- ☐ Stage bars sum to 100% within each treatment arm and match Table 1.
- ☐ Both PNGs are 2100 × 1500 pixels; both SVGs open and remain sharp when resized.
- ☐ The RTF table opens with editable text and cells in Word.
- ☐ Your script reruns from a fresh session with the supplied CSV.
- ☐ Your two or three sentences describe baseline data and state a limitation.

> Example interpretation
>
> The report describes 312 randomized patients: 158 assigned D-penicillamine and 154 assigned placebo. Baseline bilirubin is right-skewed, and cholesterol is available for 284 patients, so its summary does not cover the entire cohort. These baseline figures do not assess treatment response or survival.

For an extension, add an albumin-versus-bilirubin scatterplot, label both axes with units, and colour points by stage. Explain why an association between two baseline measurements does not show that one causes the other. Export it using a new filename.

← 5. Exporting to Word and PowerPoint Lesson 1 overview →