3. Building a Table 1

Baseline characteristics, ready for a manuscript

Contents

Setup . . . 1
Anatomy of a summary table . . . 2
The smallest useful table . . . 2
Split the columns by baseline vision . . . 2
Add the full cohort and available counts . . . 3
Cross-tabulate two categories . . . 4
Choose labels and summary statistics . . . 4
Give the table a finished appearance . . . 5
Try it . . . 6
Do we need p-values? . . . 6
Optional: the same workflow for a model . . . 7

A Table 1 summarizes the people in a study. Here we use the public Moorfields cohort from Fasler et al. (2019) to describe patients by baseline visual acuity. This is a teaching analysis, rather than a copy of the paper's table; the data notes document its preparation.

Preview the finished table. Allow 35 minutes for the main examples, with the exercise if time permits. The optional regression-table extension adds another 5–10 minutes.

Setup

Run this cell first. Each page starts a fresh R session.

```
library(gtsummary)


# Read one baseline record per patient.
cohort <- read.csv("data/moorfields_baseline.csv", na.strings = c("", "NA"))
cohort$sex <- factor(cohort$sex, levels = c("Female", "Male"))
cohort$baseline_vision <- factor(
  cohort$baseline_vision,
  levels = c("Poor", "Intermediate", "Good")
)

str(cohort)
```

Visual acuity is measured in ETDRS letters, not percentages. Higher scores mean better vision. The paper's groups are Poor (35 or fewer), Intermediate (more than 35 but fewer than 70), and Good (70 or more).

Age was coarsened in the released data: ages at or below 65 became 60, and ages above 90 became 99. We therefore call it *Recorded baseline age*. The summary describes these released values, not exact ages.

Anatomy of a summary table

Arguments inside `tbl_summary()` choose what to summarize. Steps after `|>` add to, convert, or style the result. Read this example as a guide; we build it one piece at a time below.

```
tbl_summary(
  cohort,                                    # Data: one row per patient
  by = baseline_vision,                      # Columns: one per group
  include = c(age, sex, baseline_va),         # Rows: variables to summarize
  label = list(age ~ "Recorded baseline age")   # Settings: a label for age
) |>
  add_overall() |>                           # Add the full-cohort column
  as_gt() |>                                 # Convert to a gt display table
  gt::tab_header(title = "Table 1")           # Style the display
```

`by` creates the group columns. `include` selects data variables to summarize down the rows; a categorical variable can have several rows. Grouping, labels, and additions are optional. We use `include` to keep patient identifiers out of the summary.

Inside `label`, the rule `age ~ "Recorded baseline age"` pairs a variable on the left with its setting on the right. `list()` groups the rules. The same pattern sets statistics, digits, and variable types later.

Commas separate arguments inside a function call. Each `|>` passes the result to the next function. Follow the order: **summarize, add, convert, then style**. Apply gtsummary changes before `as_gt()`, and `gt::` styling after it.

Question: which setting would rename age without changing the values summarized?

The smallest useful table

`tbl_summary()` calculates summaries. `as_gt()` turns them into a display table.

```
# Calculate summaries, then format them for display.
tbl_summary(cohort, include = c(age, sex, baseline_va)) |>
  as_gt()
```

The defaults show median (interquartile range) for continuous variables and count (%) for categories, with a missing-value row when needed. The selected variables appear down the rows; patient IDs belong in joins, not in this summary.

Split the columns by baseline vision

The `by` argument creates a column for each group.

```
# Make one column for each baseline vision group.
tbl_summary(
  cohort,
  by = baseline_vision,
  include = c(age, sex, baseline_va)
) |>
  as_gt()
```

The factor levels keep Poor, Intermediate, and Good in that order. Each header shows its patient count. These are observational groups, not randomized arms: visual acuity differs between columns because it defines the groups. The table describes the cohort; it does not estimate a treatment effect.

Add the full cohort and available counts

```
tbl_summary(
  cohort,
  by = baseline_vision,
  include = c(age, sex, baseline_va)
) |>
  # Add the full cohort and the number of non-missing values.
  add_overall() |>
  add_n() |>
  as_gt()
```

`add_overall()` adds the full cohort. `add_n()` counts patients with a recorded value for each variable. That count can differ across rows when values are missing; it is not a count of visits.

Cross-tabulate two categories

`tbl_cross()` counts combinations of two categorical variables and adds totals. Here, rows are sex and columns are baseline vision groups.

Like `tbl_summary()` in the anatomy guide, it produces a gtsummary table. Use its own `row` and `col` arguments, then convert with `as_gt()` before applying `gt::` styling.

```
# Percentages use the total within each sex row.
tbl_cross(
  cohort,
  row = sex,
  col = baseline_vision,
  percent = "row",
  label = list(sex ~ "Sex", baseline_vision ~ "Baseline vision")
) |>
  as_gt()
```

Row percentages answer: "Among women, what proportion had poor vision?" The Table 1 above uses column percentages: "Among patients with poor vision, what proportion were women?" The denominator changes with the question.

Change `percent = "row"` to `"column"`. Which percentages add to 100%, excluding the totals? Do the patient counts change?

Choose labels and summary statistics

Save the table as `tbl`. This version uses mean (SD) for numeric variables and count (%) for sex.

```
# Save the calculations for styling and export.
tbl <- tbl_summary(
  cohort,
  by = baseline_vision,
  include = c(age, sex, baseline_va),
  type = list(sex ~ "categorical"),
  label = list(
    age ~ "Recorded baseline age, years",
    sex ~ "Sex",
    baseline_va ~ "Baseline visual acuity, ETDRS letters"
  ),
  statistic = list(
    all_continuous() ~ "{mean} ({sd})",
    all_categorical() ~ "{n} ({p}%)"
  ),
  digits = list(all_continuous() ~ 1),
  missing_text = "Missing"
) |>
  add_overall() |>
  add_n(col_label = "**Available N**") |>
  bold_labels() |>
  modify_header(label ~ "**Characteristic**")

tbl |> as_gt()
```

`all_continuous()` selects all continuous variables for the `~` rule. gtsummary fills placeholders such as `{mean}` and `{sd}` with calculated values.

Try changing the continuous statistic to `"{median} ({p25}, {p75})"`. What changes? SD describes variation between patients; the confidence intervals in the paper describe uncertainty in an estimate.

Give the table a finished appearance

Keep the calculations in `tbl` and the display in `table_gt`. Run the two steps in order, following the anatomy guide: convert with `as_gt()`, then add `gt::` styling.

Step 1: title, grouped header, and notes.

```
# Convert to gt, then add the title, grouped header, and source notes.
table_gt <- as_gt(tbl) |>
  gt::tab_header(
    title = gt::md("**Table 1. The Moorfields teaching cohort**"),
    subtitle = "Baseline characteristics by visual acuity"
  ) |>
  gt::tab_spanner(
    label = "Baseline visual acuity",
    columns = c(stat_1, stat_2, stat_3)
  ) |>
  gt::cols_align(align = "center", columns = c(n, stat_0, stat_1, stat_2, stat_3)) |>
  gt::tab_source_note(
    "Teaching analysis of public Moorfields data (Fasler et al., 2019). One baseline
     ↪  record per patient; age values are coarsened."
  ) |>
  gt::tab_source_note(
    "Vision groups: Poor  35; Intermediate >35 and <70; Good  70 ETDRS letters."
  )

table_gt
```

Change the spanning label to “Vision at baseline” and rerun the cell.

Step 2: colours, fonts, and stripes.

```
# Change appearance without recalculating the summaries.
table_gt <- table_gt |>
  gt::opt_row_striping() |>
  gt::tab_options(
    heading.align = "left",
    table.font.names = c("Arial", "sans-serif"),
    table.font.size = gt::px(13),
    table.font.color = "#243746",
    table.background.color = "#FFFFFF",
    column_labels.background.color = "#003B49",
    column_labels.border.bottom.color = "#00A6A6",
    row.striping.background_color = "#F0F7F7",
    data_row.padding = gt::px(8)
  ) |>
  gt::tab_style(
    style = gt::cell_text(color = "#FFFFFF", weight = "bold"),
    locations = gt::cells_column_labels()
  )

table_gt
```

Try another header background or remove `opt_row_striping()`. Check that the labels and notes remain readable.

Part 4 makes figures; part 5 exports them with this table. Fonts and spacing vary by format.

Try it

Build a summary of age and sex, split by baseline vision, with an overall column.

```
tbl_summary(
  cohort,
  by = ______,
  include = c(age, sex),
  label = list(age ~ "Recorded baseline age, years")
) |>
  ______()
```

> **Hint**
>
> The grouping column is `baseline_vision`. The full-cohort function starts with `add_`.

Solution.

> **Solution**
>
> ```
> tbl_summary(
>   cohort,
>   by = baseline_vision,
>   include = c(age, sex),
>   label = list(age ~ "Recorded baseline age, years")
> ) |>
>   add_overall()
> ```
>
> Try adding `bold_labels()` and then `as_gt()` to display it.

Do we need p-values?

We leave them out of Table 1. Testing visual acuity across groups defined by that score adds no useful information. Other comparisons need a scientific question and an analysis plan.

Optional: the same workflow for a model

Extension: a regression table

`tbl_regression()` builds a gtsummary table from a fitted model. It follows the same conversion and styling order, with model-specific settings. Fit the model first, then label and format its estimates. This example uses baseline visual acuity, recorded age, and sex from the same cohort.
In local R, install `broom` and `broom.helpers` once if needed. The browser loads these packages for this page.

```
# Fit a separate model; keep tbl and table_gt for the Table 1 examples.
model <- lm(baseline_va ~ age + sex, data = cohort)

tbl_regression(
  model,
  label = list(age ~ "Recorded baseline age (coarsened)", sex ~ "Sex")
) |>
  # Show estimates and confidence intervals without a p-value column.
  modify_column_hide(columns = p.value) |>
  bold_labels() |>
  as_gt()
```

The coefficient rows show adjusted differences in ETDRS letters, with 95% confidence intervals. Female is the reference group: the Male coefficient compares men with women at the same recorded age. The reference row has no coefficient.
Age is coarsened in the released data, so its coefficient should not be read as a change per actual year of age. Use this example to learn table formatting; these baseline associations do not establish causation.

For further examples, see the gtsummary tutorial and gt reference.

← 2. Downloading and loading data

4. Basic figures with ggplot2 →