

Advanced Features: zztab2fig

Ronald G. Thomas

This vignette demonstrates the advanced table formatting capabilities in zztab2fig v0.2.0, including table footnotes, spanning headers, multi-row cells, decimal alignment, and LaTeX figure placement helpers.

Setup

```
library(zztab2fig)
```

1. Table Footnotes

Professional tables often require footnotes to explain abbreviations, provide context, or indicate significance levels. zztab2fig provides full footnote support via the `t2f_footnote()` function.

Basic Footnotes

```
# Create sample data
results <- data.frame(
  Variable = c("Age", "BMI", "Systolic BP", "HbA1c"),
  Treatment = c("52.3 (8.2)", "28.1 (4.3)", "142.5 (18.3)", "7.2 (1.1)"),
  Control = c("51.8 (7.9)", "27.9 (4.1)", "140.2 (17.1)", "7.4 (1.2)"),
  `P-value` = c("0.542", "0.631", "0.048", "0.089"),
  check.names = FALSE
)

# Define footnotes
fn <- t2f_footnote(
  general = c(
    "Values are mean (SD) unless otherwise noted.",
    "BMI = body mass index; BP = blood pressure; HbA1c = hemoglobin A1c."
  )
)

t2f(results,
  filename = "baseline_characteristics",
  caption = "Baseline Characteristics by Treatment Group",
  footnote = fn,
  theme = "nejm")
```

Multiple Footnote Types (APA Style)

APA format requires specific footnote organization: general notes, then specific notes, then probability notes.

```

# Data with significance markers
anova_results <- data.frame(
  Source = c("Treatment", "Time", "Treatment x Time", "Error"),
  SS = c(245.32, 189.45, 52.18, 1024.56),
  df = c(2, 3, 6, 108),
  MS = c(122.66, 63.15, 8.70, 9.49),
  F = c(12.93, 6.66, 0.92, NA),
  p = c("<0.001", "0.002", "0.485", NA)
)

# Mark significant results with footnote markers
anova_results$F[1] <- t2f_mark("12.93", 1, "symbol")
anova_results$F[2] <- t2f_mark("6.66", 2, "symbol")

fn_apa <- t2f_footnote(
  general = "SS = sum of squares; MS = mean square.",
  symbol = c("p < .001", "p < .01"),
  title_general = "Note.",
  title_symbol = NULL,
  footnote_as_chunk = TRUE
)

t2f(anova_results,
  filename = "anova_table",
  caption = "Analysis of Variance Results",
  footnote = fn_apa,
  theme = "apa")

```

Numbered Footnotes for Specific Cells

```

# Clinical trial endpoints
endpoints <- data.frame(
  Endpoint = c("Primary: ADAS-Cog14", "Secondary: CDR-SB",
              "Secondary: ADCS-ADL", "Exploratory: Brain Volume"),
  Difference = c("-0.45", "-0.21", "0.18", "-2.1%"),
  CI = c("(-0.82, -0.08)", "(-0.39, -0.03)", "(-0.15, 0.51)",
         "(-3.8%, -0.4%)"),
  P = c("0.018", "0.024", "0.285", "0.016")
)

# Add footnote markers
endpoints$Endpoint[1] <- t2f_mark(endpoints$Endpoint[1], 1, "number")
endpoints$Difference[4] <- t2f_mark(endpoints$Difference[4], 2, "number")

fn_numbered <- t2f_footnote(
  number = c(
    "Co-primary endpoint with CDR-SB.",
    "Percentage change from baseline in whole brain volume."
  )
)

t2f(endpoints,
  filename = "endpoints_table",

```

```

caption = "Efficacy Endpoints at Week 78",
footnote = fn_numbered,
theme = "nejm")

```

2. Spanning Headers (Grouped Columns)

Spanning headers group related columns under a common label, essential for complex tables comparing multiple conditions or timepoints.

Basic Spanning Header

```

# Treatment comparison data
comparison <- data.frame(
  Measure = c("Score A", "Score B", "Score C"),
  T_Mean = c(23.4, 18.7, 42.1),
  T_SD = c(5.2, 3.8, 8.9),
  C_Mean = c(21.2, 19.1, 38.5),
  C_SD = c(4.9, 4.1, 9.2)
)
names(comparison) <- c("Measure", "Mean", "SD", "Mean", "SD")

# Create spanning header
hdr <- t2f_header_above(
  " " = 1,           # Empty cell above "Measure"
  "Treatment" = 2,   # Spans Mean and SD
  "Control" = 2      # Spans Mean and SD
)

t2f(comparison,
  filename = "treatment_comparison",
  caption = "Treatment vs Control Outcomes",
  header_above = hdr)

```

Multiple Header Rows

```

# Complex longitudinal data
longitudinal <- data.frame(
  Variable = c("Cognitive", "Functional", "Behavioral"),
  W12_T = c(-1.2, -0.8, -0.3),
  W12_C = c(-0.9, -0.6, -0.2),
  W24_T = c(-2.1, -1.5, -0.5),
  W24_C = c(-1.4, -1.0, -0.3),
  W52_T = c(-3.8, -2.8, -0.9),
  W52_C = c(-2.2, -1.8, -0.5)
)
names(longitudinal) <- c("Domain", "Trt", "Ctrl", "Trt", "Ctrl", "Trt", "Ctrl")

# Two levels of headers
headers <- list(
  t2f_header_above(
    " " = 1,

```

```

    "Week 12" = 2,
    "Week 24" = 2,
    "Week 52" = 2
),
t2f_header_above(
  " " = 1,
  "Change from Baseline" = 6,
  line = FALSE
)
)

t2f(longitudinal,
  filename = "longitudinal_outcomes",
  caption = "Change from Baseline by Timepoint",
  header_above = headers,
  theme = "nature")

```

3. Multi-Row Cells (Collapsed Rows)

When data has hierarchical structure, collapsing repeated values into multi-row cells improves readability.

Basic Row Collapsing

```

# Hierarchical data
subgroup <- data.frame(
  Category = c("Age", "Age", "Sex", "Sex", "Race", "Race", "Race"),
  Subgroup = c("<65", ">=65", "Male", "Female", "White", "Black", "Asian"),
  N = c(245, 189, 218, 216, 312, 67, 55),
  Effect = c(0.42, 0.38, 0.41, 0.39, 0.40, 0.44, 0.36),
  CI = c("(0.28, 0.56)", "(0.22, 0.54)", "(0.27, 0.55)", "(0.24, 0.54)",
         "(0.28, 0.52)", "(0.24, 0.64)", "(0.14, 0.58)")
)

t2f(subgroup,
  filename = "subgroup_analysis",
  caption = "Subgroup Analysis of Treatment Effect",
  collapse_rows = t2fCollapse_rows(
    columns = 1,
    valign = "top",
    latex_hline = "major"
))

```

Multiple Column Collapse

```

# Multi-level hierarchy
adverse_events <- data.frame(
  System = c(rep("Nervous System", 3), rep("Gastrointestinal", 3),
             rep("General", 2)),
  Category = c("Headache", "Dizziness", "Somnolence",
              "Nausea", "Diarrhea", "Vomiting",
              "Fatigue", "Pyrexia"),
  Treatment = c(45, 32, 18, 28, 22, 15, 38, 12),

```

```

Placebo = c(41, 28, 14, 25, 19, 12, 35, 10)
)

t2f(adverse_events,
  filename = "adverse_events",
  caption = "Adverse Events by System Organ Class",
  collapse_rows = t2fCollapseRows(
    columns = 1,
    valign = "middle",
    latex_hline = "major"
  ),
  theme = "nejm")

```

4. Decimal Alignment

Proper decimal alignment makes numeric tables easier to read and compare. `ztab2fig` provides `siunitx` integration for precise alignment.

Basic Decimal Alignment

```

# Numeric results requiring alignment
stats <- data.frame(
  Parameter = c("Mean", "Median", "SD", "Min", "Max"),
  Treatment = c(127.45, 125.00, 18.234, 89.1, 178.92),
  Control = c(124.82, 123.50, 17.891, 91.3, 172.45)
)

# Create decimal-aligned columns
t2f(stats,
  filename = "summary_stats",
  caption = "Summary Statistics by Group",
  align = list(
    "l",                      # Parameter: left
    t2f_decimal(3, 2),        # Treatment: up to 999.99
    t2f_decimal(3, 2)         # Control: up to 999.99
  ))

```

Mixed Alignment with Custom Formats

```

# Regression coefficients
coefs <- data.frame(
  Term = c("(Intercept)", "Age", "Sex (Male)", "BMI", "Smoking"),
  Estimate = c(2.345, 0.082, -0.451, 0.123, 0.892),
  SE = c(0.234, 0.015, 0.198, 0.028, 0.245),
  t = c(10.02, 5.47, -2.28, 4.39, 3.64),
  p = c("<0.001", "<0.001", "0.024", "<0.001", "<0.001")
)

# Different precision for different columns
t2f(coefs,
  filename = "regression_coefs",

```

```

caption = "Regression Coefficients",
align = list(
  "l",
  t2f_siunitx(table_format = "1.3"),    # Estimate: X.XXX
  t2f_siunitx(table_format = "1.3"),    # SE: X.XXX
  t2f_siunitx(table_format = "2.2"),    # t: XX.XX
  "r"                                    # p: right-aligned text
))

```

5. Short Captions

For documents with a List of Tables, short captions provide concise entries while the full caption remains with the table.

```

detailed_results <- data.frame(
  Outcome = c("Primary", "Secondary 1", "Secondary 2"),
  Estimate = c(0.45, 0.32, 0.28),
  CI = c("(0.22, 0.68)", "(0.15, 0.49)", "(0.08, 0.48)"),
  P = c("0.001", "0.012", "0.042")
)

t2f(detailed_results,
  filename = "efficacy_results",
  caption = paste(
    "Efficacy Results from the Phase 3 Randomized Controlled Trial",
    "of Drug X versus Placebo in Patients with Moderate-to-Severe",
    "Condition Y (Modified Intention-to-Treat Population)"
  ),
  caption_short = "Efficacy Results from Phase 3 Trial",
  label = "tab:efficacy")

```

6. Combining Advanced Features

Real-world tables often require multiple advanced features simultaneously.

Publication-Ready Clinical Table

```

# Complex clinical trial table
clinical <- data.frame(
  Endpoint = c("ADAS-Cog14", "ADAS-Cog14", "CDR-SB", "CDR-SB"),
  Timepoint = c("Week 52", "Week 78", "Week 52", "Week 78"),
  N_T = c(423, 398, 423, 398),
  N_C = c(421, 395, 421, 395),
  Diff = c(-0.31, -0.45, -0.15, -0.21),
  CI_Low = c(-0.58, -0.82, -0.28, -0.39),
  CI_High = c(-0.04, -0.08, -0.02, -0.03),
  P = c("0.024", "0.018", "0.025", "0.024")
)

# Add significance markers
clinical$Diff <- sapply(seq_len(nrow(clinical)), function(i) {

```

```

if (as.numeric(clinical$P[i]) < 0.05) {
  t2f_mark(as.character(clinical$Diff[i]), 1, "symbol")
} else {
  as.character(clinical$Diff[i])
}
})

# Spanning header
hdr <- t2f_header_above(
  " " = 2,
  "N" = 2,
  "Treatment Effect" = 4
)

# Footnotes
fn <- t2f_footnote(
  general = "Difference is treatment minus placebo (negative favors treatment).",
  symbol = "p < 0.05 vs placebo.",
  title_general = "Note:"
)

t2f(clinical,
  filename = "clinical_efficiency",
  caption = "Efficacy Outcomes in the Modified ITT Population",
  caption_short = "Efficacy Outcomes",
  label = "tab:efficiency",
  header_above = hdr,
  footnote = fn,
  collapse_rows = t2f_collapse_rows(columns = 1, valign = "top"),
  theme = "nejm")

```

7. LaTeX Figure Placement Helpers

After generating tables, use these helpers to include them in LaTeX documents with precise placement control.

Standard Figure Float

```

# Generate the table first
t2f(mtcars[1:5, 1:4], filename = "demo_table", sub_dir = "tables")

# Include as figure (in R Markdown with results='asis')
t2f_include("tables/demo_table",
            caption = "Motor Trend Car Data (First 5 Rows)",
            label = "fig:mtcars",
            position = "htbp")

```

Exact Placement (No Floating)

```

t2f_include_inline("tables/demo_table",
                   width = "0.8\\textwidth",
                   vspace = "1em")

```

Side-by-Side Tables

```
# Generate two tables
t2f(mtcars[1:5, 1:3], filename = "table_a", sub_dir = "tables")
t2f(mtcars[1:5, 4:6], filename = "table_b", sub_dir = "tables")

# Include side by side
t2f_include_sidebyside(
  "tables/table_a", "tables/table_b",
  caption1 = "(a) Performance",
  caption2 = "(b) Design",
  main_caption = "Motor Trend Car Data by Category",
  main_label = "fig:mtcars_combined"
)
```

Margin Placement

```
# For documents with wide margins (Tufte style, etc.)
t2f_include_margin("tables/demo_table",
  caption = "Summary",
  method = " sidenotes")
```

Text Wrapping

```
t2f_include_wrap("tables/demo_table",
  placement = "r",
  wrap_width = "0.45\\textwidth",
  caption = "Car data summary")
```

8. Cross-References

Generate LaTeX cross-reference commands from R:

```
# In text, reference the table
cat("As shown in ")
t2f_ref("fig:mtcars")
cat(", the fuel efficiency varies considerably.")

# Or build the reference into a string
ref_text <- sprintf("Results are presented in %s.",
  t2f_ref("tab:efficacy", cat = FALSE))
cat(ref_text)
```

Summary

ztab2fig v0.2.0 provides comprehensive support for publication-quality tables:

- **Footnotes:** General, numbered, alphabetic, and symbol notation
- **Spanning headers:** Single or multiple levels of grouped columns
- **Multi-row cells:** Automatic collapsing of hierarchical data
- **Decimal alignment:** siunitx integration for precise number formatting
- **Short captions:** For List of Tables entries
- **Figure placement:** Float, inline, margin, wrap, and side-by-side options

These features can be combined to create tables meeting the most demanding journal requirements while maintaining a clean R-based workflow.