ZZTABLE1 User Guide

Publication-Ready Summary Tables for Clinical Research

Ronald G. Thomas

February 2025

Contents

ZZTABLE1 User Guide

Publication-Ready Summary Tables for Clinical Research

Version 0.1.0

Table of Contents

Introduction

zztable1 is an R package for creating publication-ready summary tables (Table 1) commonly used in biomedical research and clinical trials. The package provides a formula-based interface for specifying table structure with support for multiple journal formatting styles.

Key Features

- Formula-based table specification
- Medical journal themes (NEJM, Lancet, JAMA, BMJ)
- Multiple output formats (console, HTML, LaTeX)
- Configurable statistical tests
- Stratified (subgroup) analysis
- Flexible footnote system
- Missing data reporting
- Lazy evaluation architecture for efficiency

Architecture

zztable1 uses a blueprint-based architecture with lazy evaluation:

1. **Blueprint Creation**: `table1()` creates a blueprint object containing table structure and computation instructions
2. **Lazy Evaluation**: Computations are stored as expressions, not results
3. **Rendering**: `render_html()`, `render_latex()`, or `render_console()` evaluates the blueprint and produces formatted output

This architecture provides memory efficiency (60-80% reduction) and flexibility in output format selection.

Installation

From GitHub

```r
# Install devtools if needed
install.packages("devtools")

# Install zztable1
devtools::install_github("rgt47/zztable1")
```

Load the Package

```r
library(zztable1)
```

Quick Start

Basic Example

```
# Prepare data
data(mtcars)
mtcars$transmission <- factor(
  ifelse(mtcars$am == 1, "Manual", "Automatic")
)

# Create table
bp <- table1(transmission ~ mpg + hp + wt, data = mtcars)

# Display
print(bp)
```

With Journal Theme and P-values

```
bp <- table1(
  transmission ~ mpg + hp + wt + cyl,
  data = mtcars,
  theme = "nejm",
  pvalue = TRUE
)

render_html(bp)
```

Core Concepts

Formula Syntax

The formula specifies the table structure:

```
grouping_variable ~ variable1 + variable2 + variable3
```

- **Left side**: Grouping variable (creates columns)
- **Right side**: Variables to summarize (creates rows)
- **Tilde (~)**: Separates grouping from summary variables

Examples

```
# Two-group comparison
treatment ~ age + sex + bmi

# Multiple groups
arm ~ age + sex + race + baseline_score

# No grouping (descriptive statistics only)
~ age + sex + bmi
```

Variable Types

zztable1 automatically detects variable types:

R Type	Table Treatment	Default Summary
numeric	Continuous	Mean (SD)
integer	Continuous	Mean (SD)
factor	Categorical	N (%)
character	Categorical	N (%)
logical	Categorical	N (%)

Blueprint Object

The `table1()` function returns a `table1_blueprint` object:

```
bp <- table1(arm ~ age + sex, data = trial_data)
class(bp)
#> [1] "table1_blueprint"
```

The blueprint contains:

- Table dimensions (rows, columns)
- Cell computation instructions
- Metadata (theme, options, footnotes)
- Sparse storage environment for efficiency

The table1() Function

Syntax

```
table1(
  formula,
  data,
  strata = NULL,
  missing = FALSE,
  pvalue = TRUE,
  size = FALSE,
  totals = FALSE,
  layout = "console",
  numeric_summary = "mean_sd",
  theme = "console",
  continuous_test = "ttest",
  categorical_test = "fisher",
  footnotes = NULL,
  ...
)
```

Parameters

Parameter	Type	Default	Description
`formula`	formula	required	Table structure specification
`data`	data.frame	required	Data containing all variables
`strata`	character	NULL	Stratification variable name
`missing`	logical	FALSE	Show missing value counts
`pvalue`	logical	TRUE	Include p-values
`size`	logical	FALSE	Show group sizes in header
`totals`	logical	FALSE	Include totals column

Parameter	Type	Default	Description
`layout`	character	"console"	Output format hint
`numeric_summary`	character	"mean_sd"	Summary type for numeric variables
`theme`	character	"console"	Journal theme name
`continuous_test`	character	"ttest"	Test for continuous variables
`categorical_test`	character	"fisher"	Test for categorical variables
`footnotes`	list	NULL	Footnote specifications

Return Value

Returns a `table1_blueprint` S3 object with:

- `$nrows`: Number of rows
- `$ncols`: Number of columns
- `$cells`: Environment containing cell data (sparse storage)
- `$row_names`: Row labels
- `$col_names`: Column labels
- `$metadata`: List of options and computed information

Output Formats

Console Output

```
bp <- table1(transmission ~ mpg + hp, data = mtcars)
print(bp)
# or
render_console(bp)
```

HTML Output

```
bp <- table1(transmission ~ mpg + hp, data = mtcars, theme = "nejm")
render_html(bp)
```

For R Markdown documents, HTML output integrates automatically with `results='asis'` chunk option.

LaTeX Output

```
bp <- table1(transmission ~ mpg + hp, data = mtcars, theme = "lancet")
render_latex(bp)
```

LaTeX output includes necessary packages (`booktabs`, `colortbl`, `threeparttable`) and color definitions for themes.

Display Function

The `display_table()` function provides a convenient wrapper:

```
bp <- table1(transmission ~ mpg + hp, data = mtcars)
display_table(bp, mtcars, format = "console")
display_table(bp, mtcars, format = "html")
display_table(bp, mtcars, format = "latex")
```

Journal Themes

zztable1 includes themes matching major medical journal styles.

Available Themes

Theme	Description	Characteristics
`console`	Default	Monospace, minimal styling
`nejm`	New England Journal of Medicine	Cream row striping (#fefcf0), minimal borders
`lancet`	The Lancet	Clean white, horizontal borders only
`jama`	JAMA	Minimal formatting, lettered footnotes
`bmj`	British Medical Journal	Blue header accents
`simple`	Basic styling	Clean, no special formatting

Using Themes

```r
# NEJM style
bp <- table1(arm ~ age + sex, data = trial_data, theme = "nejm")

# Lancet style
bp <- table1(arm ~ age + sex, data = trial_data, theme = "lancet")

# JAMA style
bp <- table1(arm ~ age + sex, data = trial_data, theme = "jama")
```

Listing Available Themes

```r
list_available_themes()
```

Getting Theme Details

```r
theme <- get_theme("nejm")
print(theme)
```

Statistical Tests

Continuous Variables

Test	Parameter Value	Use Case
Student's t-test	`"ttest"`	Two groups, normal distribution
ANOVA	`"anova"`	Three+ groups, normal distribution
Welch's t-test	`"welch"`	Two groups, unequal variances
Kruskal-Wallis	`"kruskal"`	Non-parametric, any number of groups

Categorical Variables

Test	Parameter Value	Use Case
Fisher's exact	`"fisher"`	Small expected cell counts
Chi-squared	`"chisq"`	Large sample sizes

Specifying Tests

```r
# Non-parametric tests
bp <- table1(
  arm ~ age + sex + race,
  data = trial_data,
  continuous_test = "kruskal",
  categorical_test = "chisq"
)

# Welch's t-test for unequal variances
bp <- table1(
  arm ~ age + bmi,
  data = trial_data,
  continuous_test = "welch"
)
```

Disabling P-values

```r
bp <- table1(arm ~ age + sex, data = trial_data, pvalue = FALSE)
```

Numeric Summaries

Built-in Summary Types

Type	Parameter Value	Output Format
Mean (SD)	`"mean_sd"`	25.3 (4.2)
Median [IQR]	`"median_iqr"`	24.0 [21.0, 28.5]
Mean (95% CI)	`"mean_ci"`	25.3 (23.1, 27.5)
Median (Range)	`"median_range"`	24.0 (18.0, 35.0)

Using Summary Types

```r
# Median and IQR (common for skewed data)
bp <- table1(
  arm ~ age + los + cost,
  data = trial_data,
  numeric_summary = "median_iqr"
)

# Mean with 95% CI
bp <- table1(
  arm ~ change_score,
  data = trial_data,
  numeric_summary = "mean_ci"
)
```

Custom Summary Functions

You can provide a custom function for numeric summaries:

```r
# Custom function returning formatted string
my_summary <- function(x) {
  sprintf("%.1f [%.1f]", median(x, na.rm = TRUE), IQR(x, na.rm = TRUE))
}

bp <- table1(
  arm ~ age + bmi,
  data = trial_data,
  numeric_summary = my_summary
)
```

Stratified Analysis

Stratified analysis creates separate tables for each level of a stratification variable, useful for multi-center trials or subgroup analyses.

Basic Stratification

```r
# Stratify by study site
bp <- table1(
  arm ~ age + sex + bmi,
  data = trial_data,
  strata = "site"
)
```

Example: Multi-Center Trial

```r
# Create sample data
trial_data <- data.frame(
  site = rep(c("Site A", "Site B", "Site C"), each = 100),
  arm = rep(c("Treatment", "Placebo"), 150),
  age = rnorm(300, 55, 10),
  sex = factor(sample(c("Male", "Female"), 300, replace = TRUE)),
  response = rbinom(300, 1, 0.6)
)

# Stratified Table 1
bp <- table1(
  arm ~ age + sex + response,
  data = trial_data,
  strata = "site",
  theme = "nejm"
)

render_html(bp)
```

Footnotes

The footnote system allows adding explanatory notes to specific variables, columns, or the entire table.

Footnote Types

Type	Target	Example
`variables`	Row labels	Explain variable definition
`columns`	Column headers	Explain group definition
`general`	Table footer	General notes, abbreviations

Specifying Footnotes

```r
bp <- table1(
 arm ~ age + sex + egfr + bmi,
  data = trial_data,
  theme = "nejm",
  footnotes = list(
    variables = list(
      egfr = "Estimated glomerular filtration rate (mL/min/1.73m2)",
      bmi = "Body mass index (kg/m2)"
    ),
    columns = list(
      Treatment = "Active drug 100mg daily",
      Placebo = "Matching placebo"
    ),
    general = c(
      "Values are mean (SD) for continuous variables and N (%) for categorical.",
      "P-values from t-test (continuous) or Fisher's exact test (categorical)."
    )
  )
)
```

Theme-Specific Footnote Styles

Different themes render footnotes differently:

- **NEJM**: Superscript numbers
- **JAMA**: Superscript letters
- **Lancet**: Symbols (*, dagger, etc.)
- **Console**: Bracketed numbers

Missing Data

Showing Missing Counts

```r
bp <- table1(
  arm ~ age + sex + lab_value,
  data = trial_data,
  missing = TRUE
)
```

When `missing = TRUE`, each variable with missing values gets an additional row showing the count of missing observations per group.

Output Format

```
Variable           Treatment    Placebo     P-value
                   (N=150)      (N=150)
Age, mean (SD)     55.2 (10.1)  54.8 (9.8)  0.72
```

```
  Missing         2 (1.3%)     1 (0.7%)
Lab Value         ...          ...          ...
  Missing         15 (10.0%)   12 (8.0%)
```

R Markdown Integration

HTML Documents

~~~
```{r table1, results='asis'}
library(zztable1)

bp <- table1(
  arm ~ age + sex + bmi,
  data = trial_data,
  theme = "nejm"
)

render_html(bp)
```
~~~

PDF Documents (LaTeX)

~~~
```{r table1, results='asis'}
library(zztable1)

bp <- table1(
  arm ~ age + sex + bmi,
  data = trial_data,
  theme = "lancet"
)

render_latex(bp)
```
~~~

Required YAML for PDF

For PDF output with colored themes, include these LaTeX packages:

```yaml
output:
  pdf_document:
    extra_dependencies:
      - colortbl
      - booktabs
      - threeparttable
      - xcolor
```

Automatic Format Detection

For vignettes or documents that may render to multiple formats, use a helper function:

```r
display_auto <- function(bp) {
  if (knitr::is_latex_output()) {
    render_latex(bp)
  } else if (knitr::is_html_output()) {
    render_html(bp)
```

```
  } else {
    render_console(bp)
  }
}
```

Advanced Usage

Accessing Blueprint Internals

```
bp <- table1(arm ~ age + sex, data = trial_data)

# Dimensions
bp$nrows
bp$ncols

# Row and column names
bp$row_names
bp$col_names

# Metadata
bp$metadata$theme
bp$metadata$options
```

Memory Information

```
bp <- table1(arm ~ age + sex + race + bmi, data = large_dataset)
blueprint_memory_info(bp)
```

Subsetting Blueprints

```
# Access specific cells
bp[1, 2]  # Row 1, Column 2
bp[1:5, ] # First 5 rows
```

Clearing Cache

For large tables, clear the computation cache to free memory:

```
clear_cell_cache(bp)
```

Troubleshooting

Common Issues

"Variables not found in data"

```
# Error: Variables not found in data: age, sex
```

Solution: Check variable names match exactly (case-sensitive):

```
names(data)  # List available variables
```

“Unknown theme”

```
# Warning: Unknown theme 'minimal', using 'console'
```

Solution: Use a valid theme name:

```
list_available_themes()  # See available themes
```

LaTeX Color Errors

```
! Package xcolor Error: Undefined color 'nejmstripe'
```

Solution: Add required LaTeX packages to YAML header:

```
extra_dependencies:
  - colortbl
  - xcolor
```

Large Table Performance For tables with many variables or large datasets:

1. Use `missing = FALSE` unless needed
2. Consider `pvalue = FALSE` for descriptive tables
3. Use stratification judiciously

Getting Help

```
# Function documentation
?table1
?render_html
?render_latex

# List available themes
list_available_themes()
```

Function Reference

Primary Functions

Function	Purpose
`table1()`	Create table blueprint
`render_html()`	Render to HTML
`render_latex()`	Render to LaTeX
`render_console()`	Render to console
`display_table()`	Convenience display function
`print.table1_blueprint()`	Print method for blueprints

Theme Functions

Function	Purpose
`list_available_themes()`	List all available themes
`get_theme()`	Get theme configuration

Utility Functions

Function	Purpose
`blueprint_memory_info()`	Memory usage statistics
`clear_cell_cache()`	Clear computation cache
`validate_inputs()`	Validate table1 inputs

Examples

Clinical Trial Table 1

```r
# Simulated clinical trial data
set.seed(42)
n <- 300
trial <- data.frame(
  arm = factor(rep(c("Treatment", "Placebo"), each = n/2)),
  age = rnorm(n, 58, 12),
  sex = factor(sample(c("Male", "Female"), n, replace = TRUE)),
  race = factor(sample(c("White", "Black", "Asian", "Other"), n,
                       replace = TRUE, prob = c(0.6, 0.2, 0.15, 0.05))),
  bmi = rnorm(n, 28, 5),
  diabetes = factor(sample(c("Yes", "No"), n, replace = TRUE,
                           prob = c(0.3, 0.7))),
  baseline_score = rnorm(n, 50, 15)
)

# Create publication-ready table
bp <- table1(
  arm ~ age + sex + race + bmi + diabetes + baseline_score,
  data = trial,
  theme = "nejm",
  pvalue = TRUE,
  numeric_summary = "mean_sd",
  footnotes = list(
    variables = list(
      bmi = "Body mass index (kg/m2)",
      baseline_score = "Baseline assessment score (0-100)"
    ),
    general = c(
      "Values are mean (SD) or N (%).",
      "P-values from t-test or Fisher's exact test."
    )
  )
)

# Render for journal submission
render_latex(bp)
```

Observational Study

```r
# Using mtcars as example
data(mtcars)
```

```r
mtcars$efficiency <- factor(
  ifelse(mtcars$mpg > median(mtcars$mpg), "High", "Low")
)
mtcars$cylinders <- factor(mtcars$cyl)

bp <- table1(
  efficiency ~ hp + wt + qsec + cylinders,
  data = mtcars,
  theme = "lancet",
  continuous_test = "wilcox",
  footnotes = list(
    variables = list(
      hp = "Gross horsepower",
      wt = "Weight (1000 lbs)",
      qsec = "1/4 mile time (seconds)"
    )
  )
)

render_html(bp)
```

Version History

- **0.1.0**: Initial release with blueprint architecture
  - Formula-based interface
  - NEJM, Lancet, JAMA, BMJ themes
  - HTML, LaTeX, console output
  - Stratified analysis
  - Footnote system

License

GPL-3

Author

Ronald G. Thomas

Citation

```
Thomas RG (2024). zztable1: Publication-Ready Summary Tables for
Clinical Research. R package version 0.1.0.
```

Document generated: February 2025