Customizing Statistics and Tests in zztable1

zztable1

2026-02-18

Contents

1 Introduction

The `zztable1` package provides powerful customization options for both **summary statistics** and **statistical tests** used in Table 1 generation. This vignette demonstrates how to:

- Customize numeric summary statistics with built-in and custom functions
- Select appropriate statistical tests for different data types
- Apply these customizations across different medical journal themes
- Handle real-world scenarios with varying data distributions

1.1 Key Customization Parameters

The package provides two main customization parameters that mirror each other in design:

- **`numeric_summary`**: Controls how continuous variables are summarized
- **`continuous_test`**: Controls statistical tests for continuous variables

- **`categorical_test`**: Controls statistical tests for categorical variables

2 Dataset Preparation

We'll use a simulated clinical trial dataset to demonstrate all customization options:

```
# Create comprehensive clinical trial dataset
set.seed(123)
n_per_group <- 40

clinical_data <- data.frame(
  # Treatment groups
  treatment = factor(rep(c("Placebo", "Low Dose", "High Dose"), each = n_per_group)),

  # Patient characteristics
  age = c(
    rnorm(n_per_group, mean = 65, sd = 12),    # Placebo
    rnorm(n_per_group, mean = 63, sd = 10),    # Low Dose
    rnorm(n_per_group, mean = 67, sd = 15)     # High Dose
  ),

  # Primary efficacy endpoint (with treatment effect)
  efficacy_score = c(
    rnorm(n_per_group, mean = 20, sd = 8),     # Placebo: lower scores
    rnorm(n_per_group, mean = 28, sd = 9),     # Low Dose: moderate improvement
    rnorm(n_per_group, mean = 35, sd = 7)      # High Dose: best improvement
  ),

  # Safety endpoint (non-normal distribution)
  biomarker_level = c(
    rexp(n_per_group, rate = 0.1),             # Placebo: exponential distribution
    rexp(n_per_group, rate = 0.08),            # Low Dose
```

```
    rexp(n_per_group, rate = 0.12)              # High Dose
  ),

  # Binary outcomes
  response = factor(c(
    sample(c("Responder", "Non-responder"), n_per_group, replace = TRUE, prob = c(0.3, 0.7)),
    sample(c("Responder", "Non-responder"), n_per_group, replace = TRUE, prob = c(0.6, 0.4)),
    sample(c("Responder", "Non-responder"), n_per_group, replace = TRUE, prob = c(0.8, 0.2))
  )),

  # Categorical safety outcome
  safety_grade = factor(c(
    sample(c("None", "Mild", "Moderate", "Severe"), n_per_group, replace = TRUE, prob = c(0.4, 0.4, 0.15
    sample(c("None", "Mild", "Moderate", "Severe"), n_per_group, replace = TRUE, prob = c(0.3, 0.45, 0.1
    sample(c("None", "Mild", "Moderate", "Severe"), n_per_group, replace = TRUE, prob = c(0.2, 0.5, 0.25
  ), levels = c("None", "Mild", "Moderate", "Severe"))
)

# Display sample data
knitr::kable(head(clinical_data),
             caption = "Sample of Clinical Trial Dataset")
```

Table 1: Sample of Clinical Trial Dataset

treatment	age	efficacy_score	biomarker_level	response	safety_grade
Placebo	58.27429	20.94117	15.9962002	Responder	None
Placebo	62.23787	12.42020	3.7548537	Non-responder	None
Placebo	83.70450	16.07554	0.8025235	Non-responder	None
Placebo	65.84610	17.95126	3.5644771	Non-responder	Moderate
Placebo	66.55145	34.75090	12.7150987	Non-responder	Moderate
Placebo	85.58078	14.78440	22.0726519	Non-responder	Mild

3 Customizing Summary Statistics

3.1 Built-in Summary Options

The package provides several built-in summary statistics optimized for different data types and journal requirements:

```
cat("**Available Built-in Summary Statistics:**\n\n")
```

Available Built-in Summary Statistics:

```
cat("- `mean_sd`: Mean +/- SD (default, most common)\n")
```

- `mean_sd`: Mean +/- SD (default, most common)

```
cat("- `median_iqr`: Median [Q1-Q3] (for non-normal data)\n")
```

- `median_iqr`: Median [Q1-Q3] (for non-normal data)

```
cat("- `median_range`: Median (min-max) (for small samples)\n")
```

- `median_range`: Median (min-max) (for small samples)

```
cat("- `mean_se`: Mean +/- SE (for experimental data)\n")
```

- `mean_se`: Mean +/- SE (for experimental data)

```
cat("- `mean_ci`: Mean (95% CI) (for effect estimates)\n\n")
```

- `mean_ci`: Mean (95% CI) (for effect estimates)

3.1.1 Comparison of Built-in Summaries

Let's see how different summary statistics present the same data:

```
cat("### Mean +/- SD (Default Clinical Standard)\n")
```

3.1.2 Mean +/- SD (Default Clinical Standard)

```
create_table(
  treatment ~ age + efficacy_score + biomarker_level,
  data = clinical_data,
  numeric_summary = "mean_sd",
  pvalue = TRUE,
  theme = "nejm"
)
```

variables	High Dose	Low Dose	Placebo	p.value
age	67.1 ± 12.7	62.9 ± 9.6	65.5 ± 10.8	0.0939
efficacy_score	35.1 ± 6.5	28.1 ± 8.9	19.2 ± 8.4	2e-04
biomarker_level	8 ± 7.5	10.4 ± 12.1	9.2 ± 8	0.259

```
cat("\n### Median [IQR] (For Non-Normal Data)\n")
```

3.1.3 Median [IQR] (For Non-Normal Data)

```
create_table(
  treatment ~ age + efficacy_score + biomarker_level,
  data = clinical_data,
  numeric_summary = "median_iqr",
  pvalue = TRUE,
  theme = "nejm"
)
```

variables	High Dose	Low Dose	Placebo	p.value
age	66.2 [57.9-74.9]	62.4 [57.5-67.8]	66.1 [58.3-73.3]	0.0939
efficacy_score	33.4 [30-38.9]	27 [22.2-32.1]	18.5 [12.4-25.5]	2e-04
biomarker_level	5.3 [3.2-9.9]	5.6 [1.7-15]	7.4 [3.4-11.6]	0.259

```
cat("\n### Median (Range) (For Small Samples)\n")
```

3.1.4 Median (Range) (For Small Samples)

```
create_table(
  treatment ~ age + efficacy_score + biomarker_level,
  data = clinical_data,
  numeric_summary = "median_range",
```

```
  pvalue = TRUE,
  theme = "nejm"
)
```

variables	High Dose	Low Dose	Placebo	p.value
age	66.2 (42-99.8)	62.4 (39.9-84.7)	66.1 (41.4-86.4)	0.0939
efficacy_score	33.4 (25.8-50.4)	27 (16.2-57.2)	18.5 (3.6-36.8)	2e-04
biomarker_level	5.3 (0-32.3)	5.6 (0.2-45.5)	7.4 (0.1-30.4)	0.259

```
cat("\n### Mean +/- SE (For Experimental Data)\n")
```

3.1.5 Mean +/- SE (For Experimental Data)

```
create_table(
  treatment ~ age + efficacy_score,
  data = clinical_data,
  numeric_summary = "mean_se",
  pvalue = TRUE,
  theme = "nejm"
)
```

variables	High Dose	Low Dose	Placebo	p.value
age	67.1 +/- 2	62.9 +/- 1.5	65.5 +/- 1.7	0.0939
efficacy_score	35.1 +/- 1	28.1 +/- 1.4	19.2 +/- 1.3	2e-04

3.2 Custom Summary Functions

You can create custom summary functions for specialized requirements:

```
# Example 1: Bootstrap confidence intervals
bootstrap_ci_summary <- function(x) {
  if (all(is.na(x))) return("N/A")

  # Bootstrap 95% CI for mean
  set.seed(42)  # For reproducibility in vignette
  n_boot <- 1000
  boot_means <- replicate(n_boot, {
    sample_data <- sample(x[!is.na(x)], replace = TRUE)
    mean(sample_data)
  })

  mean_est <- round(mean(x, na.rm = TRUE), 1)
  ci_lower <- round(quantile(boot_means, 0.025), 1)
  ci_upper <- round(quantile(boot_means, 0.975), 1)

  paste0(mean_est, " (", ci_lower, "-", ci_upper, ")")
}

# Example 2: Robust statistics (median with MAD)
robust_summary <- function(x) {
  if (all(is.na(x))) return("N/A")

  med <- round(median(x, na.rm = TRUE), 1)
```

```
  mad_val <- round(mad(x, na.rm = TRUE), 1)

  paste0(med, " [+/-", mad_val, "]")
}

# Example 3: Multi-line detailed summary
detailed_summary <- function(x) {
  if (all(is.na(x))) return("N/A")

  mean_val <- round(mean(x, na.rm = TRUE), 1)
  median_val <- round(median(x, na.rm = TRUE), 1)
  sd_val <- round(sd(x, na.rm = TRUE), 1)

  paste0(mean_val, " +/- ", sd_val, "\n", "(median: ", median_val, ")")
}

cat("### Custom Summary: Bootstrap 95% CI\n")
```

3.2.1 Custom Summary: Bootstrap 95% CI

```
create_table(
  treatment ~ efficacy_score,
  data = clinical_data,
  numeric_summary = bootstrap_ci_summary,
  pvalue = TRUE,
  theme = "jama"
)
```

variables	High Dose	Low Dose	Placebo	p.value
efficacy_score	35.1 (33.2-37)	28.1 (25.8-30.9)	19.2 (16.7-21.8)	2e-04

```
cat("\n### Custom Summary: Robust Statistics (Median +/- MAD)\n")
```

3.2.2 Custom Summary: Robust Statistics (Median +/- MAD)

```
create_table(
  treatment ~ biomarker_level,
  data = clinical_data,
  numeric_summary = robust_summary,
  pvalue = TRUE,
  theme = "jama"
)
```

variables	High Dose	Low Dose	Placebo	p.value
biomarker_level	5.3 [+/-5]	5.6 [+/-7.1]	7.4 [+/-6.1]	0.259

```
cat("\n### Custom Summary: Multi-line Detailed\n")
```

3.2.3 Custom Summary: Multi-line Detailed

```
create_table(
  treatment ~ efficacy_score,
```

```
  data = clinical_data,
  numeric_summary = detailed_summary,
  pvalue = TRUE,
  theme = "console"
)
```

variables	High Dose	Low Dose	Placebo	p.value
efficacy_score	35.1 +/- 6.5 (median: 33.4)	28.1 +/- 8.9 (median: 27)	19.2 +/- 8.4 (median: 18.5)	2e-04

4 Customizing Statistical Tests

4.1 Available Statistical Tests

The package supports multiple statistical tests appropriate for different data distributions and study designs:

```
cat("**Continuous Variable Tests:**\n")
```

Continuous Variable Tests:

```
cat("- `ttest`: Linear model t-test (default, robust for multiple groups)\n")
```

- `ttest`: Linear model t-test (default, robust for multiple groups)

```
cat("- `anova`: Traditional ANOVA F-test\n")
```

- `anova`: Traditional ANOVA F-test

```
cat("- `welch`: Welch's t-test (unequal variances, two groups only)\n")
```

- `welch`: Welch's t-test (unequal variances, two groups only)

```
cat("- `kruskal`: Kruskal-Wallis test (non-parametric)\n\n")
```

- `kruskal`: Kruskal-Wallis test (non-parametric)

```
cat("**Categorical Variable Tests:**\n")
```

Categorical Variable Tests:

```
cat("- `fisher`: Fisher's exact test (default, conservative)\n")
```

- `fisher`: Fisher's exact test (default, conservative)

```
cat("- `chisq`: Chi-square test (requires adequate cell counts)\n\n")
```

- `chisq`: Chi-square test (requires adequate cell counts)

4.2 Comparing Different Statistical Tests

Let's demonstrate how different tests can give different p-values for the same data:

```
cat("### Default Tests (ttest + fisher)\n")
```

4.2.1 Default Tests (ttest + fisher)

```
create_table(
  treatment ~ efficacy_score + response + safety_grade,
  data = clinical_data,
  pvalue = TRUE,
```

```
  theme = "console"
)
```

variables	High Dose	Low Dose	Placebo	p.value
efficacy_score	35.1 (6.5)	28.1 (8.9)	19.2 (8.4)	2e-04
response				
Non-responder	8 (20%)	18 (45%)	31 (78%)	0
Responder	32 (80%)	22 (55%)	9 (22%)	
safety_grade				
None	11 (28%)	17 (42%)	19 (48%)	0.0195
Mild	14 (35%)	17 (42%)	15 (38%)	
Moderate	15 (38%)	4 (10%)	4 (10%)	
Severe	0 (0%)	2 (5%)	2 (5%)	

```
cat("\n### ANOVA + Chi-square\n")
```

4.2.2 ANOVA + Chi-square

```
create_table(
  treatment ~ efficacy_score + response + safety_grade,
  data = clinical_data,
  pvalue = TRUE,
  continuous_test = "anova",
  categorical_test = "chisq",
  theme = "console"
)
```

variables	High Dose	Low Dose	Placebo	p.value
efficacy_score	35.1 (6.5)	28.1 (8.9)	19.2 (8.4)	0
response				
Non-responder	8 (20%)	18 (45%)	31 (78%)	0
Responder	32 (80%)	22 (55%)	9 (22%)	
safety_grade				
None	11 (28%)	17 (42%)	19 (48%)	0.0195
Mild	14 (35%)	17 (42%)	15 (38%)	
Moderate	15 (38%)	4 (10%)	4 (10%)	
Severe	0 (0%)	2 (5%)	2 (5%)	

```
cat("\n### Non-parametric Approach (Kruskal-Wallis + Fisher)\n")
```

4.2.3 Non-parametric Approach (Kruskal-Wallis + Fisher)

```
create_table(
  treatment ~ efficacy_score + biomarker_level + response,
  data = clinical_data,
  pvalue = TRUE,
  continuous_test = "kruskal",
  categorical_test = "fisher",
  theme = "console"
)
```

variables	High Dose	Low Dose	Placebo	p.value
efficacy_score	35.1 (6.5)	28.1 (8.9)	19.2 (8.4)	0
biomarker_level	8 (7.5)	10.4 (12.1)	9.2 (8)	0.8225
response				
Non-responder	8 (20%)	18 (45%)	31 (78%)	0
Responder	32 (80%)	22 (55%)	9 (22%)	

4.2.4 Manual Verification of Test Results

Let's manually verify the different test results:

```
cat("**Manual Statistical Test Verification:**\n\n")
```

Manual Statistical Test Verification:

```
# Test efficacy_score with different methods
lm_result <- lm(efficacy_score ~ treatment, data = clinical_data)
aov_result <- aov(efficacy_score ~ treatment, data = clinical_data)
kw_result <- kruskal.test(efficacy_score ~ treatment, data = clinical_data)

cat("Efficacy Score Tests:\n")
```

Efficacy Score Tests:

```
cat("- Linear model p-value:", round(summary(lm_result)$coefficients[2, 4], 4), "\n")
```

- Linear model p-value: 2e-04

```
cat("- ANOVA p-value:        ", round(summary(aov_result)[[1]]["treatment", "Pr(>F)"], 4), "\n")
```

- ANOVA p-value: 0

```
cat("- Kruskal-Wallis p-value:", round(kw_result$p.value, 4), "\n\n")
```

- Kruskal-Wallis p-value: 0

```
# Test categorical data
response_table <- table(clinical_data$treatment, clinical_data$response)
fisher_result <- fisher.test(response_table)
chisq_result <- chisq.test(response_table)

cat("Response Rate Tests:\n")
```

Response Rate Tests:

```
print(response_table)
```

```
        Non-responder Responder
```

High Dose 8 32 Low Dose 18 22 Placebo 31 9

```
cat("- Fisher's exact p-value:", round(fisher_result$p.value, 4), "\n")
```

- Fisher's exact p-value: 0

```
cat("- Chi-square p-value:    ", round(chisq_result$p.value, 4), "\n")
```

- Chi-square p-value: 0

4.3 Two-Group Comparisons

For two-group studies, Welch's t-test is often preferred when variances are unequal:

```
# Create two-group subset
two_group_data <- clinical_data[clinical_data$treatment %in% c("Placebo", "High Dose"), ]
two_group_data$treatment <- factor(two_group_data$treatment)

cat("### Two-Group Comparison: Welch's t-test vs Standard t-test\n")
```

4.3.1 Two-Group Comparison: Welch's t-test vs Standard t-test

```

cat("**Welch's t-test (unequal variances assumed):**\n")
```

Welch's t-test (unequal variances assumed):

```
create_table(
  treatment ~ age + efficacy_score + biomarker_level,
  data = two_group_data,
  pvalue = TRUE,
  continuous_test = "welch",
  theme = "nejm"
)
```

variables	High Dose	Placebo	p.value
age	67.1 ± 12.7	65.5 ± 10.8	0.5507
efficacy_score	35.1 ± 6.5	19.2 ± 8.4	0
biomarker_level	8 ± 7.5	9.2 ± 8	0.5013

```
cat("\n**Standard linear model t-test:**\n")
```

Standard linear model t-test:

```
create_table(
  treatment ~ age + efficacy_score + biomarker_level,
  data = two_group_data,
  pvalue = TRUE,
  continuous_test = "ttest",
  theme = "nejm"
)
```

variables	High Dose	Placebo	p.value
age	67.1 ± 12.7	65.5 ± 10.8	0.5506
efficacy_score	35.1 ± 6.5	19.2 ± 8.4	0
biomarker_level	8 ± 7.5	9.2 ± 8	0.5012

```
# Manual verification
cat("\n**Manual Verification for Efficacy Score:**\n")
```

Manual Verification for Efficacy Score:

```
welch_test <- t.test(efficacy_score ~ treatment, data = two_group_data, var.equal = FALSE)
standard_test <- t.test(efficacy_score ~ treatment, data = two_group_data, var.equal = TRUE)

cat("- Welch's t-test p-value: ", round(welch_test$p.value, 4), "\n")
```

- Welch's t-test p-value: 0

```
cat("- Standard t-test p-value:", round(standard_test$p.value, 4), "\n")
```

- Standard t-test p-value: 0

5 Real-World Clinical Scenarios

5.1 Scenario 1: Dose-Escalation Study

For dose-escalation studies, you might want non-parametric tests and robust summaries:

```
# Simulate dose-escalation data with safety focus
set.seed(456)
dose_data <- data.frame(
  dose_level = factor(c("Cohort 1 (1mg)", "Cohort 2 (3mg)", "Cohort 3 (10mg)", "Cohort 4 (30mg)"),
                      levels = c("Cohort 1 (1mg)", "Cohort 2 (3mg)", "Cohort 3 (10mg)", "Cohort 4 (30mg)
  # Efficacy increases with dose
  efficacy = c(rnorm(8, 15, 5), rnorm(8, 25, 6), rnorm(8, 35, 8), rnorm(8, 40, 10)),
  # Safety events increase with dose
  dlt_grade = c(rexp(8, 2), rexp(8, 1.5), rexp(8, 1), rexp(8, 0.8)),
  # Binary DLT outcome
  dlt = factor(c(
    sample(c("Yes", "No"), 8, replace = TRUE, prob = c(0.1, 0.9)),
    sample(c("Yes", "No"), 8, replace = TRUE, prob = c(0.2, 0.8)),
    sample(c("Yes", "No"), 8, replace = TRUE, prob = c(0.4, 0.6)),
    sample(c("Yes", "No"), 8, replace = TRUE, prob = c(0.6, 0.4))
  ))
)

dose_footnotes <- list(
  variables = list(
    efficacy = "Efficacy endpoint measured on 0-50 scale",
    dlt_grade = "Dose-limiting toxicity severity score",
    dlt = "Dose-limiting toxicity occurrence (binary)"
  ),
  general = c(
    "Phase I dose-escalation study with 3+3 design",
    "Non-parametric tests used due to small sample sizes",
    "Median with IQR preferred for safety data"
  )
)

cat("### Phase I Dose-Escalation Study Analysis\n")
```

5.1.1 Phase I Dose-Escalation Study Analysis

```
create_table(
  dose_level ~ efficacy + dlt_grade + dlt,
  data = dose_data,
  numeric_summary = "median_iqr",
  continuous_test = "kruskal",
  categorical_test = "fisher",
  pvalue = TRUE,
```

```
  footnotes = dose_footnotes,
  theme = "nejm"
)
```

variables	Cohort 1 (1mg)	Cohort 2 (3mg)	Cohort 3 (10mg)	Cohort 4 (30mg)	p.value
efficacy[1]	31.1 [26.1-42]	28.8 [20.5-35.7]	21.5 [18.9-34.9]	35.4 [28.7-36.7]	0.8927
dlt_grade[2]	0.5 [0.3-1]	0.5 [0.1-0.6]	0.6 [0.1-0.7]	1 [0.6-1.2]	0.316
dlt[3]					
No	3 (38%)	7 (88%)	5 (62%)	6 (75%)	0.2582
Yes	5 (62%)	1 (12%)	3 (38%)	2 (25%)	

[1] Efficacy endpoint measured on 0-50 scale
[2] Dose-limiting toxicity severity score
[3] Dose-limiting toxicity occurrence (binary)
- Phase I dose-escalation study with 3+3 design
- Non-parametric tests used due to small sample sizes
- Median with IQR preferred for safety data

5.2 Scenario 2: Bioequivalence Study

For bioequivalence studies, you might prefer confidence intervals and specific tests:

```
# Simulate crossover bioequivalence data
set.seed(789)
be_data <- data.frame(
  formulation = factor(rep(c("Reference", "Test"), each = 24)),
  # Primary PK parameters
  cmax = c(
    rnorm(24, mean = 100, sd = 20),    # Reference
    rnorm(24, mean = 105, sd = 18)     # Test (slight difference)
  ),
  auc = c(
    rnorm(24, mean = 500, sd = 80),    # Reference
    rnorm(24, mean = 495, sd = 75)     # Test
  ),
  tmax = c(
    rexp(24, rate = 0.5) + 1,          # Reference (non-normal)
    rexp(24, rate = 0.6) + 1           # Test
  )
)

# Custom summary for bioequivalence (geometric mean +/- %CV)
geometric_mean_summary <- function(x) {
  if (all(is.na(x))) return("N/A")

  # Remove zeros and negative values for log transformation
  x_pos <- x[x > 0 & !is.na(x)]
  if (length(x_pos) == 0) return("N/A")

  geom_mean <- round(exp(mean(log(x_pos))), 1)
  cv_percent <- round(100 * sqrt(exp(sd(log(x_pos))^2) - 1), 1)

  paste0(geom_mean, " (", cv_percent, "%CV)")
}
```

```r
be_footnotes <- list(
  variables = list(
    cmax = "Maximum plasma concentration (ng/mL)",
    auc = "Area under concentration-time curve (ng*h/mL)",
    tmax = "Time to maximum concentration (hours)"
  ),
  general = c(
    "Randomized crossover bioequivalence study",
    "Geometric mean and %CV shown for PK parameters",
    "Non-parametric tests used for tmax (non-normal distribution)"
  )
)

cat("### Bioequivalence Study Analysis\n")
```

5.2.1 Bioequivalence Study Analysis

```r
create_table(
  formulation ~ cmax + auc,
  data = be_data,
  numeric_summary = geometric_mean_summary,
  continuous_test = "welch",
  pvalue = TRUE,
  footnotes = be_footnotes,
  theme = "jama"
)
```

variables	Reference	Test	p.value
cmax[1]	93.3 (16.6%CV)	101.3 (21.8%CV)	0.0775
auc[2]	515 (14.1%CV)	479.7 (18.6%CV)	0.1621

[1] Maximum plasma concentration (ng/mL)
[2] Area under concentration-time curve (ng*h/mL)
• Randomized crossover bioequivalence study
• Geometric mean and %CV shown for PK parameters
• Non-parametric tests used for tmax (non-normal distribution)

```r
cat("\n### Non-parametric Analysis for Tmax\n")
```

5.2.2 Non-parametric Analysis for Tmax

```r
create_table(
  formulation ~ tmax,
  data = be_data,
  numeric_summary = "median_iqr",
  continuous_test = "kruskal",  # Non-parametric for non-normal tmax
  pvalue = TRUE,
  theme = "jama"
)
```

variables	Reference	Test	p.value
tmax	2.8 [1.6-4.2]	2.1 [1.6-3.1]	0.4213

5.3 Scenario 3: Multi-center Trial

For large multi-center trials, you might use different approaches:

```
# Simulate multi-center data
set.seed(101112)
center_data <- data.frame(
  center = factor(paste("Center", rep(1:4, each = 30))),
  treatment = factor(rep(rep(c("Active", "Control"), each = 15), 4)),
  # Primary endpoint with center effects
  primary_endpoint = c(
    # Center 1
    rnorm(15, mean = 75, sd = 12), rnorm(15, mean = 65, sd = 15),
    # Center 2
    rnorm(15, mean = 78, sd = 10), rnorm(15, mean = 68, sd = 12),
    # Center 3
    rnorm(15, mean = 72, sd = 14), rnorm(15, mean = 62, sd = 18),
    # Center 4
    rnorm(15, mean = 76, sd = 11), rnorm(15, mean = 66, sd = 13)
  ),
  # Secondary binary endpoint
  response = factor(c(
    # Center 1: Active vs Control
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.7, 0.3)),
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.4, 0.6)),
    # Center 2
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.75, 0.25)),
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.35, 0.65)),
    # Center 3
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.65, 0.35)),
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.45, 0.55)),
    # Center 4
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.8, 0.2)),
    sample(c("Success", "Failure"), 15, replace = TRUE, prob = c(0.3, 0.7))
  ))
)

multicenter_footnotes <- list(
  variables = list(
    primary_endpoint = "Primary efficacy endpoint (0-100 scale)",
    response = "Binary treatment response (success/failure)"
  ),
  general = c(
    "Multi-center randomized controlled trial",
    "ANOVA used to account for treatment and center effects",
    "Chi-square test for categorical outcomes (adequate sample size)"
  )
)

cat("### Multi-center Trial: Overall Treatment Comparison\n")
```

5.3.1 Multi-center Trial: Overall Treatment Comparison

```
create_table(
  treatment ~ primary_endpoint + response,
  data = center_data,
  continuous_test = "anova",       # ANOVA for multi-center
  categorical_test = "chisq",      # Chi-square for large samples
  pvalue = TRUE,
  totals = TRUE,
  footnotes = multicenter_footnotes,
  theme = "lancet"
)
```

variables	Active	Control	Total	p.value
primary_endpoint[1]	76.1 (12.5)	64.7 (16.9)	70.4 (15.9)	0
response[2]				
Failure	15 (25%)	30 (50%)	45 (38%)	0.0083
Success	45 (75%)	30 (50%)	75 (62%)	

[1] Primary efficacy endpoint (0-100 scale)
[2] Binary treatment response (success/failure)
- Multi-center randomized controlled trial
- ANOVA used to account for treatment and center effects
- Chi-square test for categorical outcomes (adequate sample size)

```
cat("\n### Multi-center Trial: Stratified by Center\n")
```

5.3.2 Multi-center Trial: Stratified by Center

```
create_table(
  treatment ~ primary_endpoint + response,
  data = center_data,
  strata = "center",
  continuous_test = "anova",
  categorical_test = "chisq",
  pvalue = TRUE,
  theme = "lancet"
)
```

variables	Active	Control	p.value
Center: Center 1			
primary_endpoint	77.3 (14.7)	71.5 (10.9)	0
response			
Failure	1 (6.7%)	8 (53.3%)	0.0083
Success	14 (93.3%)	7 (46.7%)	
Center: Center 2			
primary_endpoint	76.1 (11.2)	71.4 (15)	0
response			
Failure	2 (13.3%)	6 (40%)	0.0083
Success	13 (86.7%)	9 (60%)	
Center: Center 3			
primary_endpoint	74.9 (15.5)	55.5 (22.2)	0
response			
Failure	3 (20%)	7 (46.7%)	0.0083
Success	12 (80%)	8 (53.3%)	
Center: Center 4			
primary_endpoint	76.2 (8.4)	60.4 (12.5)	0
response			
Failure	9 (60%)	9 (60%)	0.0083
Success	6 (40%)	6 (40%)	

6 Best Practice Guidelines

6.1 Choosing Summary Statistics

```
cat("### Summary Statistic Selection Guidelines:\n\n")
#> ### Summary Statistic Selection Guidelines:
cat("**Mean +/- SD:** Use when data is approximately normal and for most clinical trials\n")
#> **Mean +/- SD:** Use when data is approximately normal and for most clinical trials
cat("- Standard for continuous endpoints in medical literature\n")
#> - Standard for continuous endpoints in medical literature
cat("- Allows readers to assess both central tendency and variability\n\n")
#> - Allows readers to assess both central tendency and variability

cat("**Median [IQR]:** Use for non-normal data or when outliers are present\n")
#> **Median [IQR]:** Use for non-normal data or when outliers are present
cat("- Biomarker levels, time-to-event data, cost data\n")
#> - Biomarker levels, time-to-event data, cost data
cat("- More robust to extreme values\n\n")
#> - More robust to extreme values

cat("**Median (range):** Use for small samples or ordinal data\n")
#> **Median (range):** Use for small samples or ordinal data
cat("- Phase I studies with small cohorts\n")
#> - Phase I studies with small cohorts
cat("- When full range of values is important\n\n")
#> - When full range of values is important

cat("**Mean +/- SE:** Use when emphasizing precision of the estimate\n")
#> **Mean +/- SE:** Use when emphasizing precision of the estimate
cat("- Experimental studies where precision matters\n")
#> - Experimental studies where precision matters
```

```
cat("- Generally not recommended for descriptive Table 1\n\n")
#> - Generally not recommended for descriptive Table 1

cat("### Statistical Test Selection Guidelines:\n\n")
#> ### Statistical Test Selection Guidelines:
cat("**Continuous Variables:**\n")
#> **Continuous Variables:**
cat("- `ttest` (default): Robust, works well for most scenarios\n")
#> - `ttest` (default): Robust, works well for most scenarios
cat("- `anova`: Traditional choice, equivalent to ttest for multiple groups\n")
#> - `anova`: Traditional choice, equivalent to ttest for multiple groups
cat("- `welch`: Two groups with potentially unequal variances\n")
#> - `welch`: Two groups with potentially unequal variances
cat("- `kruskal`: Non-parametric alternative for non-normal data\n\n")
#> - `kruskal`: Non-parametric alternative for non-normal data

cat("**Categorical Variables:**\n")
#> **Categorical Variables:**
cat("- `fisher` (default): Conservative, exact test, works with small samples\n")
#> - `fisher` (default): Conservative, exact test, works with small samples
cat("- `chisq`: Requires adequate cell counts (rule of thumb: all cells >= 5)\n\n")
#> - `chisq`: Requires adequate cell counts (rule of thumb: all cells >= 5)
```

7 Theme Integration

All customization options work seamlessly with medical journal themes:

```
cat("### NEJM Theme with Custom Bootstrap CI Summary\n")
```

7.0.1 NEJM Theme with Custom Bootstrap CI Summary

```
create_table(
  treatment ~ efficacy_score + response,
  data = clinical_data,
  numeric_summary = bootstrap_ci_summary,
  continuous_test = "anova",
  categorical_test = "fisher",
  pvalue = TRUE,
  totals = TRUE,
  theme = "nejm"
)
```

variables	High Dose	Low Dose	Placebo	Total	p.value
efficacy_score	35.1 (33.2-37)	28.1 (25.8-30.9)	19.2 (16.7-21.8)	27.5 (25.7-29.3)	0
response					
Non-responder	8 (20%)	18 (45%)	31 (78%)	57 (48%)	0
Responder	32 (80%)	22 (55%)	9 (22%)	63 (52%)	

```
cat("\n### JAMA Theme with Robust Statistics\n")
```

7.0.2 JAMA Theme with Robust Statistics

```
create_table(
  treatment ~ biomarker_level + safety_grade,
  data = clinical_data,
  numeric_summary = robust_summary,
  continuous_test = "kruskal",
  categorical_test = "chisq",
  pvalue = TRUE,
  totals = TRUE,
  theme = "jama"
)
```

variables	High Dose	Low Dose	Placebo	Total	p.value
biomarker_level	5.3 [+/-5]	5.6 [+/-7.1]	7.4 [+/-6.1]	6.3 [+/-6.8]	0.8225
safety_grade					
None	11 (28%)	17 (42%)	19 (48%)	47 (39%)	0.0195
Mild	14 (35%)	17 (42%)	15 (38%)	46 (38%)	
Moderate	15 (38%)	4 (10%)	4 (10%)	23 (19%)	
Severe	0 (0%)	2 (5%)	2 (5%)	4 (3%)	

8 Conclusion

The `zztable1` package provides comprehensive customization options that allow you to:

1. **Adapt to different data types** using appropriate summary statistics
2. **Choose statistically appropriate tests** for your study design
3. **Maintain journal formatting standards** across all customizations
4. **Handle complex study designs** like dose-escalation and multi-center trials

8.1 Key Features Demonstrated:

- **Built-in summaries**: mean_sd, median_iqr, median_range, mean_se, mean_ci
- **Custom summary functions**: Bootstrap CI, robust statistics, multi-line summaries
- **Statistical tests**: ttest, anova, welch, kruskal for continuous; fisher, chisq for categorical
- **Real-world scenarios**: Dose-escalation, bioequivalence, multi-center studies
- **Theme integration**: All customizations work with NEJM, JAMA, Lancet themes

The consistent parameter interface (`numeric_summary`, `continuous_test`, `categorical_test`) makes it easy to standardize analyses across studies while maintaining the flexibility to adapt to specific requirements.

Package Features Demonstrated:

- Flexible summary statistics with custom function support
- Comprehensive statistical test options
- Medical journal theme integration
- Real-world clinical trial scenarios
- Best practice guidelines for method selection
- Seamless parameter integration across all output formats