Working with Different Object Types and Themes

Ronald G. Thomas

zztab2fig v0.2.0 introduced S3 method dispatch for `t2f()`, enabling direct conversion of statistical objects to publication-quality tables. This vignette demonstrates how different R object types are handled and how themes affect table appearance.

```r
library(zztab2fig)
```

Object Types

Data Frames

Data frames are the most common input type. The `t2f.data.frame()` method handles standard data frame objects with full formatting support.

```r
df <- data.frame(
  Variable = c("Age", "Weight", "Height"),
  Mean = c(45.2, 72.5, 168.3),
  SD = c(12.1, 15.8, 9.2),
  N = c(150, 148, 150)
)

t2f(df, filename = "descriptive_stats", sub_dir = output_dir,
    caption = "Descriptive Statistics")
```

Table 1: Descriptive Statistics

Variable	Mean	SD	N
Age	45.2	12.1	150
Weight	72.5	15.8	148
Height	168.3	9.2	150

Data frames from dplyr pipelines work seamlessly:

```r
library(dplyr)

summary_table <- mtcars |>
  group_by(cyl) |>
  summarise(
    N = n(),
    `Mean MPG` = round(mean(mpg), 1),
    `SD MPG` = round(sd(mpg), 1),
```

```
    `Mean HP` = round(mean(hp), 0),
    .groups = "drop"
  )

t2f(summary_table,
    filename = "cylinder_summary",
    caption = "Vehicle Statistics by Cylinder Count",
    sub_dir = output_dir)
```

Table 1: Vehicle Statistics by Cylinder Count

cyl	N	Mean_MPG	SD_MPG	Mean_HP
4	11	26.7	4.5	83
6	7	19.7	1.5	122
8	14	15.1	2.6	209

Matrices

Matrices are converted to data frames with optional row names as the first column via the `t2f.matrix()` method.

```
cor_mat <- cor(mtcars[, c("mpg", "cyl", "disp", "hp", "wt")])
cor_mat <- round(cor_mat, 2)

t2f(cor_mat,
    filename = "correlation_matrix",
    caption = "Correlation Matrix of Vehicle Characteristics",
    rownames = TRUE,
    sub_dir = output_dir)
```

Table 1: Correlation Matrix of Vehicle Characteristics

	rowname	mpg	cyl	disp	hp	wt
mpg	mpg	1.00	-0.85	-0.85	-0.78	-0.87
cyl	cyl	-0.85	1.00	0.90	0.83	0.78
disp	disp	-0.85	0.90	1.00	0.79	0.89
hp	hp	-0.78	0.83	0.79	1.00	0.66
wt	wt	-0.87	0.78	0.89	0.66	1.00

Contingency Tables

Base R `table` objects are supported directly via the `t2f.table()` method.

```
gear_cyl_table <- table(mtcars$gear, mtcars$cyl)

t2f(gear_cyl_table,
    filename = "gear_cylinder_crosstab",
    caption = "Cross-tabulation of Gears and Cylinders",
    sub_dir = output_dir)
```

Table 1: Cross-tabulation of Gears and Cylinders

	rowname	X4	X6	X8
3	3	1	2	12
4	4	8	4	0
5	5	2	1	2

Linear Models

The `t2f.lm()` method extracts coefficient tables from linear model objects. Control which statistics appear using the `include` parameter.

```
model <- lm(mpg ~ cyl + hp + wt, data = mtcars)

t2f(model,
    filename = "regression_table",
    caption = "Multiple Regression: MPG Predictors",
    include = c("estimate", "std.error", "p.value"),
    digits = 3,
    sub_dir = output_dir)
```

Table 1: Multiple Regression: MPG Predictors

Term	Estimate	Std_Error	p_value
(Intercept)	38.752	1.787	<0.001
cyl	-0.942	0.551	0.098
hp	-0.018	0.012	0.140
wt	-3.167	0.741	<0.001

Add confidence intervals (code only - full statistics):

```
t2f(model,
    filename = "regression_with_ci",
    include = c("estimate", "std.error", "conf.int", "p.value"),
    conf.level = 0.95,
    sub_dir = output_dir)
```

Generalized Linear Models

The `t2f.glm()` method supports logistic, Poisson, and other GLM families. Use `exponentiate = TRUE` to display odds ratios or rate ratios.

```
logit_model <- glm(am ~ mpg + hp + wt,
                data = mtcars,
                family = binomial)

t2f(logit_model,
    filename = "logistic_odds_ratios",
    caption = "Logistic Regression: Odds Ratios",
    exponentiate = TRUE,
    include = c("estimate", "conf.int", "p.value"),
    sub_dir = output_dir)
```

```
#> Waiting for profiling to be done...
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
#> Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Table 1: Logistic Regression: Odds Ratios

Term	OR	p_value	CI_Lower	CI_Upper
(Intercept)	0.000	0.694	0.000	2.51488e+18
mpg	3.419	0.437	0.620	1.71009e+03
hp	1.088	0.308	1.014	1.55300e+00
wt	0.001	0.038	0.000	1.16000e-01

ANOVA Tables

Both `aov` and `anova` objects are supported via `t2f.aov()` and `t2f.anova()`.

```
aov_model <- aov(mpg ~ factor(cyl) + factor(gear), data = mtcars)

t2f(aov_model,
    filename = "anova_table",
    caption = "Two-way ANOVA: MPG by Cylinder and Gear",
```

```
    sub_dir = output_dir)
```

Table 1: Two-way ANOVA: MPG by Cylinder and Gear

Source	Df	Sum_Sq	Mean_Sq	F_value	Pr__F_
factor(cyl)	2	824.785	412.392	38.001	<0.001
factor(gear)	2	8.252	4.126	0.380	0.687
Residuals	27	293.011	10.852	NA	

Hypothesis Tests

The `t2f.htest()` method handles output from t-tests, chi-square tests, and other hypothesis testing functions.

```
ttest <- t.test(mpg ~ am, data = mtcars)

t2f(ttest,
    filename = "ttest_result",
    caption = "Two-sample t-test: MPG by Transmission",
    sub_dir = output_dir)
```

Table 1: Two-sample t-test: MPG by Transmission

Statistic	Value	df	p_value	CI_Lower	CI_Upper	mean_in_group_0	mean_in_group_
t	-3.767	18.332	0.001	-11.28	-3.21	17.147	24.39

Regression Model Comparisons

The `t2f_regression()` function creates side-by-side model comparison tables with significance stars.

```
m1 <- lm(mpg ~ wt, data = mtcars)
m2 <- lm(mpg ~ wt + hp, data = mtcars)
m3 <- lm(mpg ~ wt + hp + cyl, data = mtcars)

t2f_regression(
  `(1)` = m1,
  `(2)` = m2,
  `(3)` = m3,
  stars = TRUE,
  digits = 3,
  filename = "model_progression",
  sub_dir = output_dir
)
```

Term	X_1_	X_2_	X_3_
(Intercept)	37.285* (1.878)	37.227* (1.599)	38.752* (1.787)
wt	-5.344* (0.559)	-3.878* (0.633)	-3.167* (0.741)
hp		-0.032* (0.009)	-0.018 (0.012)
cyl			-0.942 (0.551)
N	32	32	32
R-squared	0.753	0.827	0.843
Adj. R-squared	0.745	0.815	0.826

Additional Object Types via broom

zztab2fig v0.2.0 adds support for 13 additional statistical object types through broom integration. These methods use `broom::tidy()` to extract model information, then format the results.

Survival Analysis (survival package)

```r
library(survival)

cox_model <- coxph(Surv(time, status) ~ age + sex, data = lung)
t2f(cox_model, exponentiate = TRUE, filename = "cox_hr")

survreg_model <- survreg(Surv(time, status) ~ age + sex, data = lung)
t2f(survreg_model, filename = "parametric_survival")

km_fit <- survfit(Surv(time, status) ~ sex, data = lung)
t2f(km_fit, filename = "kaplan_meier")
```

Mixed Effects Models (lme4, nlme packages)

```r
library(lme4)

lmer_model <- lmer(mpg ~ cyl + (1 | gear), data = mtcars)
t2f(lmer_model, filename = "mixed_effects")

library(nlme)
lme_model <- lme(mpg ~ cyl, random = ~ 1 | gear, data = mtcars)
t2f(lme_model, filename = "nlme_model")
```

Other Statistical Models

```r
nls_model <- nls(mpg ~ a * exp(b * wt), data = mtcars,
                 start = list(a = 40, b = -0.5))
t2f(nls_model, filename = "nonlinear")

library(MASS)
polr_model <- polr(factor(gear) ~ mpg + hp, data = mtcars)
t2f(polr_model, filename = "ordinal_logistic")
```

Exploratory Analysis

```r
pca_result <- prcomp(mtcars[, 1:5], scale. = TRUE)
t2f(pca_result, matrix = "rotation", filename = "pca_loadings")
t2f(pca_result, matrix = "summary", filename = "pca_summary")

km_result <- kmeans(mtcars[, 1:5], centers = 3)
t2f(km_result, matrix = "centers", filename = "cluster_centers")
```

Complete List of broom-Based Methods

Object Class	Package	Method	Key Parameters
coxph	survival	t2f.coxph()	exponentiate, conf.int
survreg	survival	t2f.survreg()	digits
survfit	survival	t2f.survfit()	times
survdiff	survival	t2f.survdiff()	digits
Arima	stats	t2f.Arima()	digits
nls	stats	t2f.nls()	conf.int
polr	MASS	t2f.polr()	exponentiate, conf.int
multinom	nnet	t2f.multinom()	exponentiate, conf.int
prcomp	stats	t2f.prcomp()	matrix (rotation/summary)
kmeans	stats	t2f.kmeans()	matrix (centers/summary)
lmerMod	lme4	t2f.lmerMod()	effects, conf.int
glmerMod	lme4	t2f.glmerMod()	effects, exponentiate
lme	nlme	t2f.lme()	effects

Note: These methods require the `broom` package (or `broom.mixed` for lme4/nlme models) and the respective model packages to be installed.

Theme Demonstrations

Each built-in theme provides distinct styling appropriate for different publication contexts. Below are side-by-side comparisons using the same data.

```
demo_data <- mtcars[1:6, c("mpg", "cyl", "hp", "wt")]
```

Default Theme

The default styling uses blue row shading with booktabs rules.

```
t2f(demo_data,
    filename = "theme_default",
    caption = "Default Theme",
    sub_dir = output_dir)
```

Table 1: Default Theme

	mpg	cyl	hp	wt
Mazda RX4	21.0	6	110	2.620
Mazda RX4 Wag	21.0	6	110	2.875
Datsun 710	22.8	4	93	2.320
Hornet 4 Drive	21.4	6	110	3.215
Hornet Sportabout	18.7	8	175	3.440
Valiant	18.1	6	105	3.460

NEJM Theme

The New England Journal of Medicine theme provides clean, professional medical journal styling with light yellow shading and sans-serif fonts.

```
t2f(demo_data,
    filename = "theme_nejm",
    caption = "NEJM Theme",
    theme = "nejm",
    sub_dir = output_dir)
```

Table 1: NEJM Theme

	mpg	cyl	hp	wt
Mazda RX4	21.0	6	110	2.620
Mazda RX4 Wag	21.0	6	110	2.875
Datsun 710	22.8	4	93	2.320
Hornet 4 Drive	21.4	6	110	3.215
Hornet Sportabout	18.7	8	175	3.440
Valiant	18.1	6	105	3.460

NEJM theme characteristics:

- Light yellow alternating row colors
- Sans-serif font (Helvetica)
- Booktabs horizontal rules
- Compact column spacing (4pt)
- Footnotesize font

APA Theme

The American Psychological Association theme follows APA 7th edition table guidelines.

```
t2f(demo_data,
    filename = "theme_apa",
    caption = "APA Theme",
    theme = "apa",
    sub_dir = output_dir)
```

Table 1: APA Theme

	mpg	cyl	hp	wt
Mazda RX4	21.0	6	110	2.620
Mazda RX4 Wag	21.0	6	110	2.875
Datsun 710	22.8	4	93	2.320
Hornet 4 Drive	21.4	6	110	3.215
Hornet Sportabout	18.7	8	175	3.440
Valiant	18.1	6	105	3.460

APA theme characteristics:

- No row shading
- Times Roman font
- Letter paper with 1-inch margins
- Booktabs rules

Nature Theme

The Nature journal theme uses Helvetica with subtle gray shading.

```
t2f(demo_data,
    filename = "theme_nature",
    caption = "Nature Theme",
    theme = "nature",
    sub_dir = output_dir)
```

Table 1: Nature Theme

	mpg	cyl	hp	wt
Mazda RX4	21.0	6	110	2.620
Mazda RX4 Wag	21.0	6	110	2.875
Datsun 710	22.8	4	93	2.320
Hornet 4 Drive	21.4	6	110	3.215
Hornet Sportabout	18.7	8	175	3.440
Valiant	18.1	6	105	3.460

Nature theme characteristics:

- Light gray row shading (gray!8)
- Sans-serif font (Helvetica)
- Small font size
- 15mm margins

Minimal Theme

The minimal theme provides clean, unobtrusive styling suitable for general use.

```
t2f(demo_data,
    filename = "theme_minimal",
    caption = "Minimal Theme",
    theme = "minimal",
    sub_dir = output_dir)
```

Table 1: Minimal Theme

	mpg	cyl	hp	wt
Mazda RX4	21.0	6	110	2.620
Mazda RX4 Wag	21.0	6	110	2.875
Datsun 710	22.8	4	93	2.320
Hornet 4 Drive	21.4	6	110	3.215
Hornet Sportabout	18.7	8	175	3.440
Valiant	18.1	6	105	3.460

Custom Themes

Create custom themes for specific requirements using `t2f_theme()`.

```r
presentation_theme <- t2f_theme(
  name = "presentation",
  scolor = "blue!15",
  header_bold = TRUE,
  font_size = "large",
  striped = TRUE,
  extra_packages = list(
    geometry(margin = "10mm"),
    "\\renewcommand{\\arraystretch}{1.3}"
  )
)

t2f(demo_data,
    filename = "theme_presentation",
    caption = "Custom Presentation Theme",
    theme = presentation_theme,
    sub_dir = output_dir)
```

Table 1: Custom Presentation Theme

	mpg	cyl	hp	wt
Mazda RX4	21.0	6	110	2.620
Mazda RX4 Wag	21.0	6	110	2.875
Datsun 710	22.8	4	93	2.320
Hornet 4 Drive	21.4	6	110	3.215
Hornet Sportabout	18.7	8	175	3.440
Valiant	18.1	6	105	3.460

Global Theme Setting

Set a theme globally for consistent styling across multiple tables in an analysis.

```
t2f_theme_set("nejm")

t2f(mtcars[1:5, 1:4], filename = "table1", sub_dir = "output")
t2f(iris[1:5, ], filename = "table2", sub_dir = "output")

model <- lm(mpg ~ wt + hp, data = mtcars)
t2f(model, filename = "table3", sub_dir = "output")

t2f(mtcars[1:5, 1:4],
    filename = "table4_apa",
    theme = "apa",
    sub_dir = "output")

t2f_theme_set(NULL)
```

Summary

Object Type Support (Base Methods)

Object Class	Method	Key Parameters
data.frame	t2f.data.frame()	Standard t2f parameters
matrix	t2f.matrix()	rownames
table	t2f.table()	Standard t2f parameters
lm	t2f.lm()	digits, include, conf.level
glm	t2f.glm()	digits, include, exponentiate, conf.level
aov	t2f.aov()	digits
anova	t2f.anova()	digits

Object Class	Method	Key Parameters
htest	t2f.htest()	digits

Object Type Support (via broom)

Object Class	Package	Key Parameters
coxph, survreg, survfit, survdiff	survival	exponentiate
nls	stats	conf.int
polr, multinom	MASS, nnet	exponentiate
prcomp, kmeans	stats	matrix
lmerMod, glmerMod	lme4	effects
lme	nlme	effects
Arima	stats	digits

Theme Comparison

Theme	Row Shading	Font	Use Case
Default	blue!10	Serif	General purpose
NEJM	yellow!8	Helvetica	Medical journals
APA	None	Times	Psychology/social sciences
Nature	gray!8	Helvetica	Scientific journals
Minimal	gray!5	Serif	Clean, simple tables

Theme Functions

Function	Purpose
t2f_theme()	Create custom theme
t2f_theme_set()	Set global theme
t2f_theme_get()	Get current global theme
t2f_theme_nejm()	NEJM preset
t2f_theme_apa()	APA preset
t2f_theme_nature()	Nature preset
t2f_theme_minimal()	Minimal preset
t2f_list_themes()	List available themes