Tabling Challenges: pander vs zztab2fig

Ronald G. Thomas

This vignette presents a series of tabling challenges of varying difficulty, showing solutions in both pander and zztab2fig. Each challenge demonstrates the relative strengths and approaches of each package.

Setup

```
suppressPackageStartupMessages(library(zztab2fig))
```

Standard Difficulty Challenges

Challenge 1: Basic Data Frame

Task: Display a simple data frame with default formatting.

pander Solution

```
df <- mtcars[1:5, 1:4]

pander::pander(df)
```

Output (Markdown):

```
------------------------------------------
                  mpg   cyl   disp   hp
-------------------- ----- ----- ------ ----
   **Mazda RX4**       21     6    160   110

 **Mazda RX4 Wag**     21     6    160   110

  **Datsun 710**      22.8    4    108    93
------------------------------------------
```

zztab2fig Solution

```
df <- mtcars[1:5, 1:4]

t2f(df, filename = "basic_table", sub_dir = output_dir)
```

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110
Hornet Sportabout	18.7	8	360	175

Comparison: pander outputs markdown for inline display; zztab2fig produces publication-ready PDF files.

Challenge 2: Custom Column Alignment

Task: Create a table with left-aligned text and right-aligned numbers.

pander Solution

```
df <- data.frame(
  Name = c("Alice", "Bob", "Charlie"),
  Score = c(95.5, 87.2, 91.8),
  Rank = c(1, 3, 2)
)

pander::pander(df, justify = c("left", "right", "right"))
```

zztab2fig Solution

```
df <- data.frame(
  Name = c("Alice", "Bob", "Charlie"),
  Score = c(95.5, 87.2, 91.8),
  Rank = c(1, 3, 2)
)

t2f(df, filename = "aligned_table", sub_dir = output_dir,
    align = c("l", "r", "r"))
```

Name	Score	Rank
Alice	95.5	1
Bob	87.2	3
Charlie	91.8	2

Comparison: Both handle basic alignment. zztab2fig auto-detects based on column type.

Challenge 3: Table with Caption

Task: Add a caption to the table.

pander Solution

```
df <- iris[1:5, ]

pander::pander(df, caption = "First Five Rows of Iris Dataset")
```

Output: Caption appears below table with "Table:" prefix.

zztab2fig Solution

```
df <- iris[1:5, ]

t2f(df,
    filename = "iris_sample",
    sub_dir = output_dir,
    caption = "First Five Rows of Iris Dataset",
    label = "tab:iris")
```

Table 1: First Five Rows of Iris Dataset

Sepal_Length	Sepal_Width	Petal_Length	Petal_Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

Comparison: zztab2fig adds cross-reference labels; pander does not.

Challenge 4: Regression Model Output

Task: Display linear regression coefficients.

pander Solution

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

pander::pander(model)
```

Output:

```
-------------------------------------------------------------
                Estimate   Std. Error   t value   Pr(>|t|)
----------------- ---------- ------------ --------- ----------
 **(Intercept)**     38.75       1.787       21.69    3.043e-18

     **cyl**        -0.9416      0.5509     -1.709     0.0985
-------------------------------------------------------------
```

zztab2fig Solution

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

t2f(model,
    filename = "regression",
    sub_dir = output_dir,
    include = c("estimate", "std.error", "p.value"),
    caption = "Linear Regression Results")
```

Table 1: Linear Regression Results

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

Comparison: Both handle lm objects natively. zztab2fig allows selecting which statistics to include.

Challenge 5: Highlighting Cells

Task: Emphasize specific cells in a table.

pander Solution

```
df <- mtcars[1:5, 1:4]

pander::pander(df, emphasize.strong.rows = 2, emphasize.italics.cols = 3)
```

zztab2fig Solution

```
df <- mtcars[1:5, 1:4]

t2f(df,
    filename = "emphasized",
    sub_dir = output_dir,
    formatting = list(t2f_bold_col(1)))
```

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110
Hornet Sportabout	18.7	8	360	175

Comparison: pander uses row/column indices; zztab2fig supports conditional formatting with LaTeX colors.

Advanced Challenges

Challenge 6: Table Footnotes

Task: Add footnotes explaining abbreviations and significance levels.

pander Solution

```
df <- data.frame(
  Variable = c("BMI*", "SBP", "DBP"),
  Mean = c(27.5, 142.3, 88.2),
  SD = c(4.2, 18.5, 11.3)
)

pander::pander(df, caption = "Clinical Measurements")
# Footnote must be added as separate text outside the table
```

Limitation: pander has no native table footnote support. Footnotes must be added manually as separate text.

zztab2fig Solution

```
df <- data.frame(
  Variable = c("BMI", "SBP", "DBP"),
  Mean = c(27.5, 142.3, 88.2),
  SD = c(4.2, 18.5, 11.3)
)

df$Variable[1] <- t2f_mark("BMI", 1, "symbol")

fn <- t2f_footnote(
  general = "SBP = Systolic Blood Pressure; DBP = Diastolic Blood Pressure.",
  symbol = "Body Mass Index (kg/m2).",
  threeparttable = TRUE
)

t2f(df,
    filename = "clinical_footnotes",
    sub_dir = output_dir,
    caption = "Clinical Measurements",
    footnote = fn)
```

Table 1: Clinical Measurements

Variable	Mean	SD
BMI*	27.5	4.2
SBP	142.3	18.5
DBP	88.2	11.3

Note:
SBP = Systolic Blood Pressure; DBP = Diastolic Blood Pressure.
* Body Mass Index (kg/m2).

Advantage: zztab2fig produces proper LaTeX footnotes within the table environment using threeparttable.

Challenge 7: Spanning Column Headers

Task: Create grouped column headers (e.g., "Treatment" spanning two columns).

pander Solution

```
df <- data.frame(
  Outcome = c("Score A", "Score B"),
  T_Mean = c(45.2, 38.7),
  T_SD = c(8.3, 6.2),
  C_Mean = c(42.1, 37.9),
  C_SD = c(7.9, 5.8)
)

# pander cannot create spanning headers
# Best approximation: rename columns
names(df) <- c("Outcome", "Treatment Mean", "Treatment SD",
               "Control Mean", "Control SD")
pander::pander(df)
```

Limitation: pander does not support spanning (multi-column) headers.

zztab2fig Solution

```
df <- data.frame(
  Outcome = c("Score A", "Score B"),
  T_Mean = c(45.2, 38.7),
  T_SD = c(8.3, 6.2),
  C_Mean = c(42.1, 37.9),
  C_SD = c(7.9, 5.8)
```

```
)
names(df) <- c("Outcome", "Mean", "SD", "Mean", "SD")

hdr <- t2f_header_above(
  " " = 1,
  "Treatment" = 2,
  "Control" = 2
)

t2f(df,
    filename = "spanning_header",
    sub_dir = output_dir,
    caption = "Outcomes by Group",
    header_above = hdr)
```

Table 1: Outcomes by Group

	Treatment		Control	
Outcome	**Mean**	**SD**	**Mean**	**SD**
Score A	45.2	8.3	42.1	7.9
Score B	38.7	6.2	37.9	5.8

Advantage: zztab2fig creates proper LaTeX multicolumn headers with cmidrule separators.

Challenge 8: Multi-Row Cells (Row Spanning)

Task: Merge cells vertically for hierarchical data.

pander Solution

```
df <- data.frame(
  Category = c("Treatment", "Treatment", "Control", "Control"),
  Subgroup = c("Male", "Female", "Male", "Female"),
  N = c(45, 52, 48, 49),
  Response = c("72%", "68%", "45%", "42%")
)

# pander cannot merge rows - display as-is with repeated values
pander::pander(df)
```

Limitation: pander does not support multi-row (vertically merged) cells.

zztab2fig Solution

```
df <- data.frame(
  Category = c("Treatment", "Treatment", "Control", "Control"),
  Subgroup = c("Male", "Female", "Male", "Female"),
  N = c(45, 52, 48, 49),
  Response = c("72%", "68%", "45%", "42%")
```

```
)

t2f(df,
    filename = "multirow",
    sub_dir = output_dir,
    caption = "Response by Group and Sex",
    collapse_rows = t2f_collapse_rows(
      columns = 1,
      valign = "middle",
      latex_hline = "major"
    ))
```

Table 1: Response by Group and Sex

Category	Subgroup	N	Response
Treatment	Male	45	72%
Treatment	Female	52	68%
Control	Male	48	45%
Control	Female	49	42%

Advantage: zztab2fig automatically merges repeated values using LaTeX multirow.

Challenge 9: Decimal Point Alignment

Task: Align numbers on the decimal point for easy comparison.

pander Solution

```
df <- data.frame(
  Item = c("A", "B", "C"),
  Value1 = c(1.5, 123.45, 12.345),
  Value2 = c(0.001, 10.1, 1000.01)
)

# pander uses right-alignment for numbers
# No decimal alignment available
pander::pander(df, justify = c("left", "right", "right"))
```

Limitation: pander supports left/center/right alignment only, not decimal alignment.

zztab2fig Solution

```
df <- data.frame(
  Item = c("A", "B", "C"),
  Value1 = c(1.5, 123.45, 12.345),
  Value2 = c(0.001, 10.1, 1000.01)
)

t2f(df,
```

```
    filename = "decimal_aligned",
    sub_dir = output_dir,
    align = list(
      "l",
      t2f_siunitx(table_format = "3.3"),
      t2f_siunitx(table_format = "4.2")
    ))
```

Item	Value1	Value2
A	1.500	0.001
B	123.450	10.100
C	12.345	1000.010

Advantage: zztab2fig uses siunitx for true decimal alignment.

Challenge 10: Multi-Page Tables

Task: Create a table that spans multiple pages with repeated headers.

pander Solution

```
df <- mtcars[rep(1:32, 3), ]  # 96 rows

# pander splits wide tables but not long tables
# For long tables, output continues without repeated headers
pander::pander(df)
```

Limitation: pander splits wide tables horizontally but does not handle multi-page tables with repeated headers.

zztab2fig Solution

```
df <- mtcars[rep(1:10, 2), 1:4]  # Smaller example for vignette

t2f(df,
    filename = "multipage",
    sub_dir = output_dir,
    caption = "mtcars Data (Repeated)",
    longtable = TRUE)
```

Table 1: mtcars Data (Repeated)

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160.0	110
Mazda RX4 Wag	21.0	6	160.0	110
Datsun 710	22.8	4	108.0	93
Hornet 4 Drive	21.4	6	258.0	110
Hornet Sportabout	18.7	8	360.0	175
Valiant	18.1	6	225.0	105
Duster 360	14.3	8	360.0	245
Merc 240D	24.4	4	146.7	62
Merc 230	22.8	4	140.8	95
Merc 280	19.2	6	167.6	123
Mazda RX4.1	21.0	6	160.0	110
Mazda RX4 Wag.1	21.0	6	160.0	110
Datsun 710.1	22.8	4	108.0	93
Hornet 4 Drive.1	21.4	6	258.0	110
Hornet Sportabout.1	18.7	8	360.0	175
Valiant.1	18.1	6	225.0	105
Duster 360.1	14.3	8	360.0	245
Merc 240D.1	24.4	4	146.7	62
Merc 230.1	22.8	4	140.8	95
Merc 280.1	19.2	6	167.6	123

Advantage: zztab2fig uses LaTeX longtable for proper page breaks with repeated headers.

Challenge 11: Journal-Specific Formatting

Task: Format table according to NEJM (New England Journal of Medicine) style guidelines.

pander Solution

```
df <- data.frame(
  Characteristic = c("Age", "Male sex", "BMI"),
  Treatment = c("65.2 (8.4)", "58%", "27.3 (4.1)"),
  Placebo = c("64.8 (8.1)", "56%", "27.1 (3.9)")
)

# pander has no journal-specific themes
# Manual formatting required
pander::panderOptions("table.style", "rmarkdown")
pander::pander(df, caption = "Baseline Characteristics")
```

zztab2fig Solution

```
df <- data.frame(
  Characteristic = c("Age", "Male sex", "BMI"),
  Treatment = c("65.2 (8.4)", "58%", "27.3 (4.1)"),
  Placebo = c("64.8 (8.1)", "56%", "27.1 (3.9)")
```

```
)

t2f(df,
    filename = "nejm_table",
    sub_dir = output_dir,
    caption = "Baseline Characteristics",
    theme = "nejm")
```

Table 1: Baseline Characteristics

Characteristic	Treatment	Placebo
Age	65.2 (8.4)	64.8 (8.1)
Male sex	58%	56%
BMI	27.3 (4.1)	27.1 (3.9)

Advantage: zztab2fig includes built-in themes for NEJM, APA, and Nature journals.

Challenge 12: Model Comparison Table

Task: Display multiple regression models side by side with significance stars.

pander Solution

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

# pander displays one model at a time
pander::pander(m1)
pander::pander(m2)
pander::pander(m3)

# Side-by-side comparison requires manual construction
```

Limitation: pander has no built-in model comparison functionality.

zztab2fig Solution

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

t2f_regression(
  Model1 = m1,
  Model2 = m2,
  Model3 = m3,
  stars = TRUE,
  filename = "model_comparison",
```

```
  sub_dir = output_dir
)
```

Term	Model1	Model2	Model3
(Intercept)	37.885* (2.074)	36.908* (2.191)	38.752* (1.787)
cyl	-2.876* (0.322)	-2.265* (0.576)	-0.942 (0.551)
hp		-0.019 (0.015)	-0.018 (0.012)
wt			-3.167* (0.741)
N	32	32	32
R-squared	0.726	0.741	0.843
Adj. R-squared	0.717	0.723	0.826

Advantage: zztab2fig provides `t2f_regression()` for publication-ready model comparison tables.

Challenge 13: Figure Placement in LaTeX

Task: Include a table as a float with specific positioning.

pander Solution

```
df <- mtcars[1:5, 1:4]

# pander outputs markdown - LaTeX float control not applicable
pander::pander(df)

# For LaTeX output, user must manually wrap in figure environment
```

Limitation: pander targets markdown; LaTeX float control requires manual intervention.

zztab2fig Solution

```
df <- mtcars[1:5, 1:4]

t2f(df, filename = "float_demo", sub_dir = output_dir)

# In LaTeX documents, use:
# t2f_include("float_demo", caption = "Sample Data",
#             label = "fig:sample", position = "htbp")
```

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110
Hornet Sportabout	18.7	8	360	175

Advantage: zztab2fig provides helper functions for all LaTeX placement options (float, inline, margin, wrap).

Challenge 14: Combining Multiple Features

Task: Create a complex table with footnotes, spanning headers, multi-row cells, and NEJM styling.

pander Solution

```
# This combination is not achievable in pander
# Would require extensive manual LaTeX coding
```

Limitation: pander cannot combine these advanced features.

zztab2fig Solution

```
df <- data.frame(
  Endpoint = c("Primary", "Primary", "Secondary", "Secondary"),
  Timepoint = c("Week 26", "Week 52", "Week 26", "Week 52"),
  N = c(245, 232, 245, 232),
  Difference = c(-0.42, -0.58, -0.28, -0.35),
  P = c("0.008", "0.002", "0.045", "0.018")
)

df$Difference <- sapply(seq_len(nrow(df)), function(i) {
  p <- as.numeric(df$P[i])
  if (p < 0.01) t2f_mark(as.character(df$Difference[i]), 2, "symbol")
  else if (p < 0.05) t2f_mark(as.character(df$Difference[i]), 1, "symbol")
  else as.character(df$Difference[i])
})

hdr <- t2f_header_above(" " = 2, "Results" = 3)

fn <- t2f_footnote(
  general = "Negative values favor treatment.",
  symbol = c("p < 0.05", "p < 0.01")
)

t2f(df,
    filename = "complex_table",
    sub_dir = output_dir,
    caption = "Efficacy Results by Endpoint and Timepoint",
    caption_short = "Efficacy Results",
    header_above = hdr,
    collapse_rows = t2f_collapse_rows(1, valign = "top"),
    footnote = fn,
    theme = "nejm")
```

Table 1: Efficacy Results by Endpoint and Timepoint

		Results		
Endpoint	**Timepoint**	**N**	**Difference**	**P**
Primary	Week 26	245	-0.42†	0.008
Primary	Week 52	232	-0.58†	0.002
Secondary	Week 26	245	-0.28*	0.045
Secondary	Week 52	232	-0.35*	0.018

Note:
Negative values favor treatment.
* p ¡ 0.05
† p ¡ 0.01

Summary Comparison

Feature	pander	zztab2fig
Basic tables	Yes	Yes
Column alignment	L/C/R	L/C/R + decimal
Captions	Yes	Yes + short captions
Cross-reference labels	No	Yes
Table footnotes	No	Yes (4 notation types)
Spanning headers	No	Yes
Multi-row cells	No	Yes
Multi-page tables	No	Yes (longtable)
Journal themes	No	Yes (NEJM, APA, Nature)
Model comparison	No	Yes
LaTeX float control	No	Yes
Margin placement	No	Yes
Output formats	Markdown	PDF, PNG, SVG, TEX

When to Use Each Package

Choose pander when:

- Working in R Markdown targeting HTML/Word output
- Need quick inline display of many R object types
- Document will be processed through Pandoc
- Simple tables without advanced formatting

Choose zztab2fig when:

- Creating publication-ready tables for journals
- Need footnotes, spanning headers, or multi-row cells
- Working with LaTeX documents

- Require decimal alignment
- Need consistent journal-specific styling
- Creating cropped PDF tables for inclusion in documents