Contingency Tables

library(apa7)  
library(dplyr)  
library(tidyr)  
library(flextable)  
library(ftExtra)

# Data

d <- mtcars |>  
 select(Gears = gear, Transmission = am) |>  
 mutate(Transmission = factor(Transmission, labels = c("Automatic", "Manual")))

# The apa\_chisq contingency table function

The default output of the apa\_chisq function is a contingency table with a chi-square test of independence in the table note. The output is a flextable, which can be modified with flextable commands.

apa\_chisq(d)

| Transmission | 3 | | 4 | | 5 | |
| --- | --- | --- | --- | --- | --- | --- |
| *n* | % | *n* | % | *n* | % |
| Automatic | 15 | 100.0% | 4 | 33.3% | 0 | 0.0% |
| Manual | 0 | 0.0% | 8 | 66.7% | 5 | 100.0% |
| *Note*. *χ*2(2) = 20.94, *p* < .001, Adj. Cramer’s *V* = .78 | | | | | | |

# Styling Options

The table can be styled in many ways:

apa\_chisq(  
 d,  
 font\_size = 16,  
 line\_spacing = 3,  
 text\_color = "darkred",  
 border\_width = 2,  
 border\_color = "gray",  
 font\_family = "Arial"  
)

| Transmission | 3 | | 4 | | 5 | |
| --- | --- | --- | --- | --- | --- | --- |
| *n* | % | *n* | % | *n* | % |
| Automatic | 15 | 100.0% | 4 | 33.3% | 0 | 0.0% |
| Manual | 0 | 0.0% | 8 | 66.7% | 5 | 100.0% |
| *Note*. *χ*2(2) = 20.94, *p* < .001, Adj. Cramer’s *V* = .78 | | | | | | |

# No note  
apa\_chisq(d, note = NA)

| Transmission | 3 | | 4 | | 5 | |
| --- | --- | --- | --- | --- | --- | --- |
| *n* | % | *n* | % | *n* | % |
| Automatic | 15 | 100.0% | 4 | 33.3% | 0 | 0.0% |
| Manual | 0 | 0.0% | 8 | 66.7% | 5 | 100.0% |

# Custom note  
apa\_chisq(  
 d,  
 note = "This is a \*custom\* note written in \*\*markdown\*\* $x > \\omega$."  
)

| Transmission | 3 | | 4 | | 5 | |
| --- | --- | --- | --- | --- | --- | --- |
| *n* | % | *n* | % | *n* | % |
| Automatic | 15 | 100.0% | 4 | 33.3% | 0 | 0.0% |
| Manual | 0 | 0.0% | 8 | 66.7% | 5 | 100.0% |
| *Note*. This is a *custom* note written in **markdown** *x* > *ω*. | | | | | | |

# What if I want something completely different?

There are a lot of options out there. The flextable package has the proc\_freq function, which has the ability to include row, column, and total percentages in the table.

mtcars %>%  
 proc\_freq(row = "gear", col = "vs") %>%  
 theme\_apa()

| gear |  | vs | | |
| --- | --- | --- | --- | --- |
| 0 | 1 | Total |
| 3 | Count | 12 (37.5%) | 3 (9.4%) | 15 (46.9%) |
| Mar. pct (1) | 66.7% ; 80.0% | 21.4% ; 20.0% |  |
| 4 | Count | 2 (6.2%) | 10 (31.2%) | 12 (37.5%) |
| Mar. pct | 11.1% ; 16.7% | 71.4% ; 83.3% |  |
| 5 | Count | 4 (12.5%) | 1 (3.1%) | 5 (15.6%) |
| Mar. pct | 22.2% ; 80.0% | 7.1% ; 20.0% |  |
| Total | Count | 18 (56.2%) | 14 (43.8%) | 32 (100.0%) |
| (1) Columns and rows percentages | | | | |

These can be turned off selectively:

mtcars %>%  
 proc\_freq(  
 row = "gear",  
 col = "vs",  
 include.row\_percent = FALSE,  
 include.column\_percent = FALSE,  
 include.table\_percent = FALSE  
 ) %>%  
 apa\_style(table\_width = .5)

| gear | vs | | |
| --- | --- | --- | --- |
| 0 | 1 | Total |
| 3 | 12 | 3 | 15 |
| 4 | 2 | 10 | 12 |
| 5 | 4 | 1 | 5 |
| Total | 18 | 14 | 32 |

The flextable package’s tabulator function has considerable power in creating a wide variety of descriptive tables. In general, the tabulator function requires that you calculate the statistics first, and then you specify where they should go.

Here I calculate the means, standard deviations, and sample sizes within each cell of a contingency table.

warpbreaks |>  
 summarise(  
 Mean = mean(breaks, na.rm = TRUE),  
 stdev = sd(breaks, na.rm = TRUE),  
 sample\_size = sum(!is.na(breaks)),  
 .by = c(wool, tension)  
 ) %>%  
 rename(Wool = wool) |>  
 mutate(  
 tension = factor(  
 tension,  
 labels = paste(c("Low", "Medium", "High"), "Tension")  
 )) |>  
 flextable::tabulator(  
 rows = "Wool",  
 columns = "tension",  
 M = as\_paragraph(Mean),  
 SD = as\_paragraph(stdev),  
 n = as\_paragraph(sample\_size)  
 ) |>  
 flextable::as\_flextable() |>  
 italic(i = 2, part = "header") |>  
 align(j = 1, align = "center") |>   
 theme\_apa()

| Wool |  | Low Tension | | |  | Medium Tension | | |  | High Tension | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *M* | *SD* | *n* |  | *M* | *SD* | *n* |  | *M* | *SD* | *n* |
| A |  | 44.6 | 18.1 | 9 |  | 24.0 | 8.7 | 9 |  | 24.6 | 10.3 | 9 |
| B |  | 28.2 | 9.9 | 9 |  | 28.8 | 9.4 | 9 |  | 18.8 | 4.9 | 9 |

# Processing Data Yourself

What if you want it to look a little different? Unfortunately, you might have to do some of the heavy lifting yourself. The tidyr::pivot\_wider function can save a lot of time, if you know how to use the names\_glue argument to get the column names in the right order and names\_vary argument to get the columns in the right order.

A subtle difference in this version of the table is that the columns are decimal aligned, which is evident in the standard deviation column.

Also, whereas previously the Wool column header was middle aligned, it is now top aligned, which is consistent with comparable examples in the *APA Style Manual*. I am not sure if this is a real rule in APA style, and I actually prefer middle alignment for situations like this.

warpbreaks %>%  
 summarise(  
 M = mean(breaks, na.rm = TRUE),  
 SD = sd(breaks, na.rm = TRUE),  
 n = sum(!is.na(breaks)),  
 .by = c(wool, tension)  
 ) %>%  
 rename(Wool = wool) %>%  
 mutate(  
 tension = factor(  
 tension,  
 levels = c("L", "M", "H"),  
 labels = paste(c("Low", "Medium", "High"), "Tension")  
 )  
 ) |>  
 pivot\_wider(  
 values\_from = c(M, SD, n),  
 names\_from = tension,  
 names\_glue = "{tension}\_{.value}",  
 names\_vary = "slowest"  
 ) |>  
 add\_break\_columns(ends\_with("\_n"), omit\_last = TRUE) |>  
 apa\_flextable(column\_formats = column\_formats(accuracy = .1)) |>  
 align(j = 1, align = "center")

| Wool | Low Tension | | |  | Medium Tension | | |  | High Tension | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *M* | *SD* | *n* | *M* | *SD* | *n* | *M* | *SD* | *n* |
| A | 44.6 | 18.1 | 9 |  | 24.0 | 8.7 | 9 |  | 24.6 | 10.3 | 9 |
| B | 28.2 | 9.9 | 9 |  | 28.8 | 9.4 | 9 |  | 18.8 | 4.9 | 9 |

library(tables)  
tab <- tabular(  
 (Species + 1) ~ (n = 1) + Format(digits = 2) \*  
 (Sepal.Length + Sepal.Width) \* (mean + sd),  
 data = iris  
 )  
as.matrix(tab)   
#> [,1] [,2] [,3] [,4] [,5] [,6]   
#> [1,] "" "" "Sepal.Length" "" "Sepal.Width" ""   
#> [2,] "Species" "n" "mean" "sd" "mean" "sd"   
#> [3,] "setosa" " 50" "5.01" "0.35" "3.43" "0.38"  
#> [4,] "versicolor" " 50" "5.94" "0.52" "2.77" "0.31"  
#> [5,] "virginica" " 50" "6.59" "0.64" "2.97" "0.32"  
#> [6,] "All" "150" "5.84" "0.83" "3.06" "0.44"  
library(tinytable)  
   
library(crosstable)  
ct2 <- crosstable(mtcars2, c(starts\_with("cy"), ends\_with("at")), by=c(am, vs),   
 label=FALSE, num\_digits=3, funs=c(mean, quantile),   
 funs\_arg=list(probs=c(.25,.75))) %>%   
 as\_flextable(compact=TRUE, header\_show\_n=1:2)  
ct2 |>   
 apa\_style() |>   
 surround()

| **Engine** | **straight (N=14)** | | **vshaped (N=18)** | |
| --- | --- | --- | --- | --- |
| **Transmission** | **auto (N=7)** | **manual (N=7)** | **auto (N=12)** | **manual (N=6)** |
| **cyl** |  |  |  |  |
| 4 | 3 (27.27%) | 7 (63.64%) | 0 (0%) | 1 (9.09%) |
| 6 | 4 (57.14%) | 0 (0%) | 0 (0%) | 3 (42.86%) |
| 8 | 0 (0%) | 0 (0%) | 12 (85.71%) | 2 (14.29%) |
| **drat** |  |  |  |  |
| mean | 3.570 | 4.149 | 3.121 | 3.935 |
| quantile 25% | 3.385 | 3.965 | 3.052 | 3.690 |
| quantile 75% | 3.920 | 4.165 | 3.165 | 4.140 |