IS5 in R: Relationships Between Categorical Variables–Contingency Tables (Chapter 3)

Margaret Chien and Nicholas Horton (nhorton@amherst.edu)

July 17, 2018

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fifth Edition of *Intro Stats* (2018) by De Veaux, Velleman, and Bock. More information about the book can be found at http://wps.aw.com/aw_deveaux_stats_series. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at http://nhorton.people.amherst.edu/is5.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (http://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

Chapter 3: Relationships Between Categorical Variables–Contingency Tables

Section 3.1: Contingency Tables

XX note:Code chunk was flowing out of chunk space in pdf so I fixed that

```
library(mosaic)
library(readr)
library(janitor)
OKCupid <-
  read_csv("http://nhorton.people.amherst.edu/is5/data/OKCupid_CatsDogs.csv", skip = 1) %>%
  clean_names()
```

```
## Parsed with column specification:
## cols(
##   CatsDogsBoth = col_character(),
##   Gender = col_character(),
##   `drugsY/N` = col_character(),
##   `smokesY/N` = col_character()
## )
```

```
names(OKCupid)
```

```
## [1] "cats_dogs_both" "gender"         "drugs_y_n"      "smokes_y_n"
```

By default, `read_csv()` prints the variable names. These messages can be suppressed using the `message=FALSE` code chunk option to save space and improve readability.
Here we use the `clean_names()` function from the `janitor` package to sanitize the names of the columns (which would otherwise contain special characters or whitespace). You can use the `names()` function to check the cleaned names.
We use `skip = 1` because the first line in the original data set is `Col1`, `Col2`, etc.

XX A reference saying how we are replicating Table 3.1-3.4 starting at pg. 65 would be helpful

```
## XX Consider using useNA = "no" (in the first tally command for example) to match the table in page65

tally(~ cats_dogs_both + gender, margin = TRUE, data = OKCupid)
```

```
##                gender
## cats_dogs_both     F     M Total
##       Has Both   897   577  1474
##       Has cats  3412  2388  5800
##       Has dogs  3431  3587  7018
##       <NA>     16377 29274 45651
##       Total    24117 35826 59943
```

```
tally(~ cats_dogs_both + gender, format = "percent", margin = TRUE, data = OKCupid)
```

```
##                gender
## cats_dogs_both           F           M       Total
##       Has Both   1.4964216   0.9625811   2.4590027
##       Has cats   5.6920741   3.9837846   9.6758587
##       Has dogs   5.7237709   5.9840182  11.7077891
##       <NA>      27.3209549  48.8363946  76.1573495
##       Total     40.2332216  59.7667784 100.0000000
```

```
tally(~ cats_dogs_both | gender, format = "percent", margin = TRUE, data = OKCupid)
```

```
##                gender
## cats_dogs_both          F          M
##       Has Both   3.719368   1.610562
##       Has cats  14.147697   6.665550
##       Has dogs  14.226479  10.012282
##       <NA>      67.906456  81.711606
##       Total    100.000000 100.000000
```

```
tally(~ gender | cats_dogs_both, format = "percent", margin = TRUE, data = OKCupid)
```

```
##         cats_dogs_both
## gender   Has Both  Has cats  Has dogs      <NA>
##   F      60.85482  58.82759  48.88857  35.87435
##   M      39.14518  41.17241  51.11143  64.12565
##   Total 100.00000 100.00000 100.00000 100.00000
```

Example 3.1: Exploring Marginal Distributions

```
SuperBowl <-
  read_csv("http://nhorton.people.amherst.edu/is5/data/Watch_the_Super_bowl.csv",
           skip = 1)
```

```
## Parsed with column specification:
## cols(
##   Plan = col_character(),
##   Sex = col_character()
## )
```

```
tally(~ Plan + Sex, data = SuperBowl)
```

```
##               Sex
## Plan          Female Male
##   Commercials    156   81
##   Game           200  279
##   Wont Watch     160  132
```

Example 3.2: Exploring Percentages: Children and First-Class Ticket Holders First?

XX Consider using `y ~ x` format instead of using `~ y | x` in `tally()` calls.

```
Titanic <- read_csv("http://nhorton.people.amherst.edu/is5/data/Titanic.csv")
```

```
## Parsed with column specification:
## cols(
##   Name = col_character(),
##   Survived = col_character(),
##   Boarded = col_character(),
##   Class = col_character(),
##   MWC = col_character(),
##   Age = col_double(),
##   Adut_or_Chld = col_character(),
##   Sex = col_character(),
##   Paid = col_double(),
##   Ticket_No = col_character(),
##   Boat_or_Body = col_character(),
##   Job = col_character(),
##   Class_Dept = col_character(),
##   Class_Full = col_character()
## )
```

```
tally(~ Class + Survived, format = "percent", margin = TRUE, data = Titanic)
```

```
##        Survived
## Class         Alive       Dead      Total
##   1        9.103261   5.570652  14.673913
##   2        5.389493   7.518116  12.907609
##   3        8.152174  24.003623  32.155797
##   Crew     9.601449  30.661232  40.262681
##   Total   32.246377  67.753623 100.000000
```

```
tally(~ Survived | Class, format = "percent", margin = TRUE, data = Titanic)
```

```
##         Class
## Survived         1         2         3      Crew
##    Alive  62.03704  41.75439  25.35211  23.84702
##    Dead   37.96296  58.24561  74.64789  76.15298
##    Total 100.00000 100.00000 100.00000 100.00000
```

```
tally(~ Class | Survived, format = "percent", margin = TRUE, data = Titanic)
```

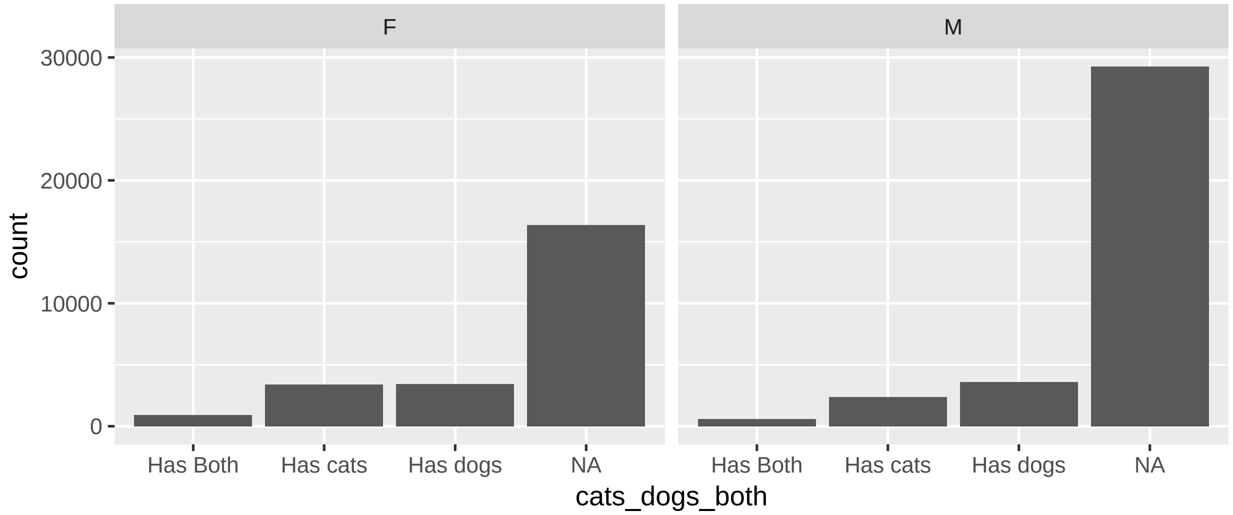
```
##        Survived
## Class         Alive       Dead
##   1       28.230337   8.221925
##   2       16.713483  11.096257
##   3       25.280899  35.427807
##   Crew    29.775281  45.254011
##   Total  100.000000 100.000000
```

Section 3.2: Conditional Distributions

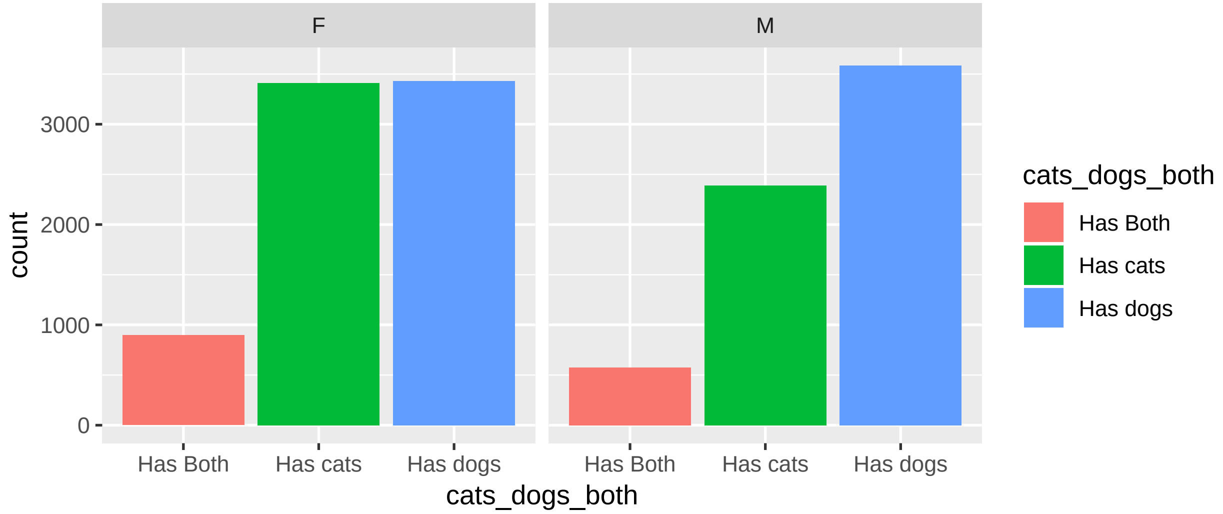
See displays on 68-69.

XX The colored bar graph is a bit confusing because the book's y axis is different so I don't know what can be done here.

```r
gf_bar(~ cats_dogs_both | gender, data = OKCupid)
```



```r
# There are many who don't own either (Figure 3.2, page 69)
gf_bar(~ cats_dogs_both | gender, fill = ~ cats_dogs_both,
  data = filter(OKCupid, cats_dogs_both != "NA"))
```



Example 3.3: Finding Conditional Distributions: Watching the Super Bowl

```r
tally(~ Plan + Sex, margin = TRUE, data = SuperBowl)
```

```
##                Sex
## Plan            Female Male Total
##   Commercials      156   81   237
##   Game          200  279   479
##   Wont Watch    160  132   292
##   Total         516  492  1008
```

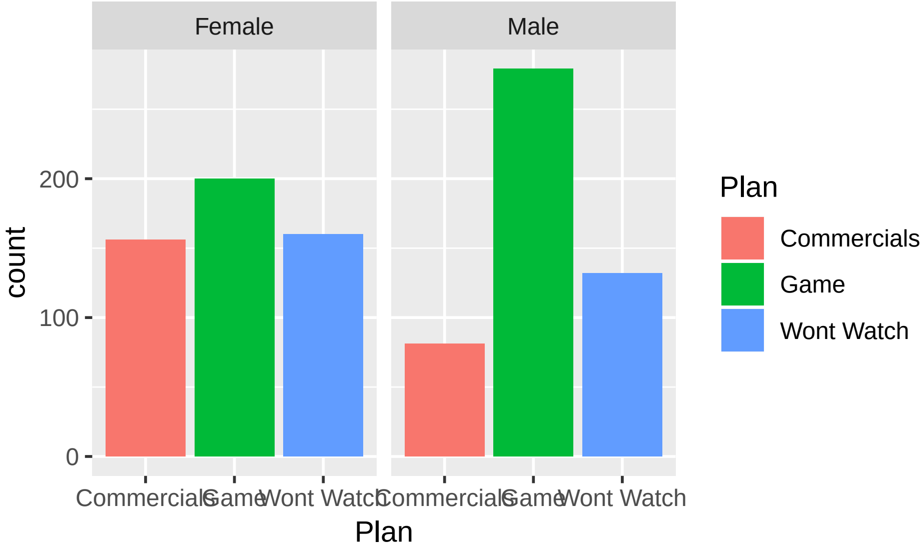
```
tally(~ Plan | Sex, format = "percent", data = SuperBowl)
```

```
##              Sex
## Plan             Female     Male
##   Commercials 30.23256 16.46341
##   Game        38.75969 56.70732
##   Wont Watch  31.00775 26.82927
```

Example 3.4: Looking for Associations Between Variables: Still Watching the Super Bowl

XX Consider this: `gf_bar(~ Plan, fill = ~ Sex, data = SuperBowl, position = "dodge")` XX The barplot still doesn't have percentages like the book(it has counts) but it is equally readable than the current version with the added bonus that it looks like the chart in the book.

```
gf_bar(~ Plan | Sex, fill = ~ Plan, format = "percent", data = SuperBowl)
```



Examining Contingency Tables

See displays on page 72.

XX Why not use `margins = TRUE` in this `tally()` call?

```
FishDiet <- read_csv("http://nhorton.people.amherst.edu/is5/data/Fish_diet.csv", skip = 1) %>%
  clean_names()
```

```
## Parsed with column specification:
## cols(
##   `Diet:Counts` = col_character(),
##   `Cancer:Counts` = col_character()
## )
```

```
tally(~ diet_counts + cancer_counts, data = FishDiet)
```

```
##            cancer_counts
## diet_counts   No  Yes
##    Large      507   42
##    Moderate 2769  209
##    Never      110   14
##    Small     2420  201
```

Random Matters

See display on page 74.

XX Consider changing the order in the `tally()` call into `~ Dream + Side` and adding `margins = TRUE`

```
Nightmares <- read_csv("http://nhorton.people.amherst.edu/is5/data/Nightmares.csv", skip = 1)
```

```
## Parsed with column specification:
## cols(
##   Side = col_character(),
##   Dream = col_character()
## )
```

```
Nightmares <- Nightmares %>%
  mutate(Dream = ifelse(Dream == "N", "Nightmare", "SweetDreams"))
tally(~ Side + Dream, data = Nightmares)
```

```
##      Dream
## Side Nightmare SweetDreams
##    L         9          13
##    R         6          35
```

Section 3.3: Displaying Contingency Tables

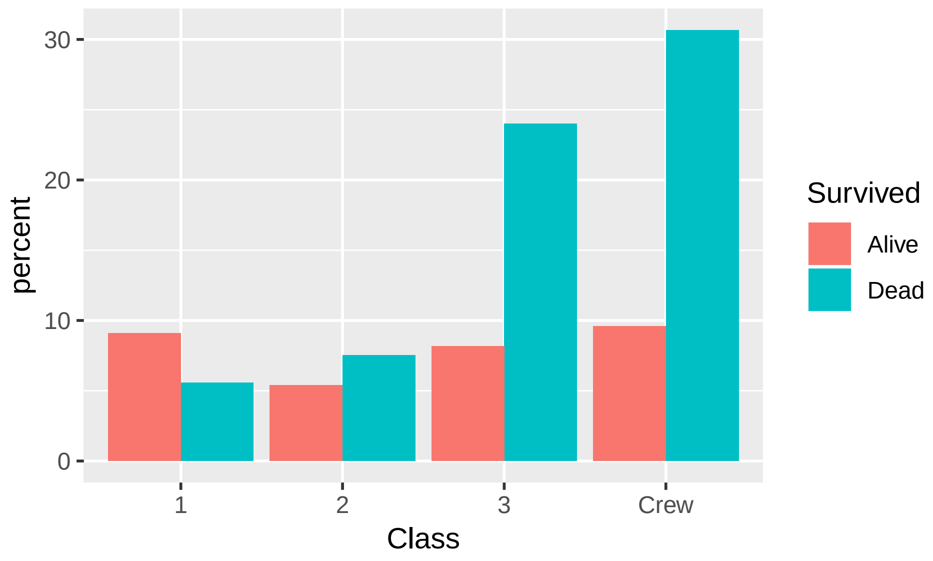
```
tally(~ Class + Survived, format = "count", data = Titanic)
```

```
##       Survived
## Class  Alive Dead
##   1      201  123
##   2      119  166
##   3      180  530
##   Crew   212  677
```

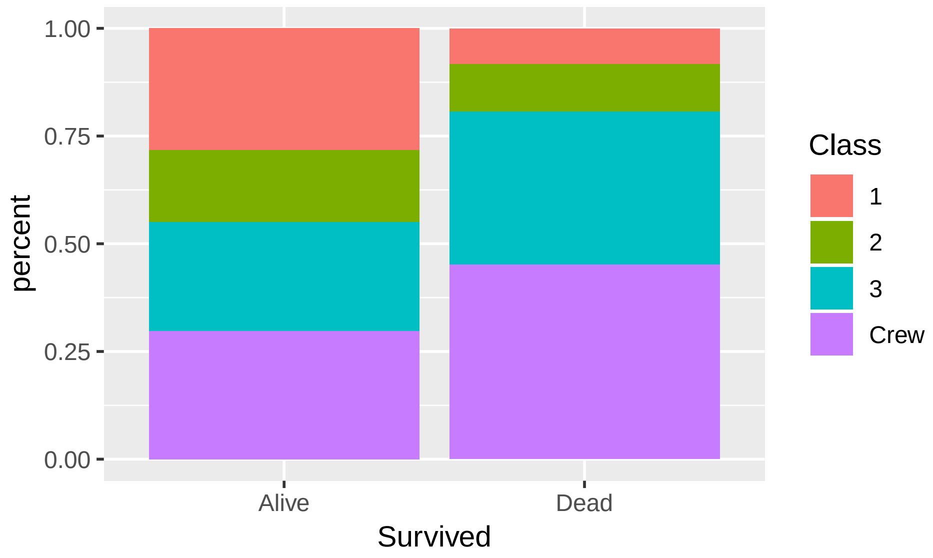
```
tally(~ Class + Survived, format = "percent", data = Titanic)
```

```
##       Survived
## Class      Alive      Dead
##   1     9.103261  5.570652
##   2     5.389493  7.518116
##   3     8.152174 24.003623
##   Crew  9.601449 30.661232
```

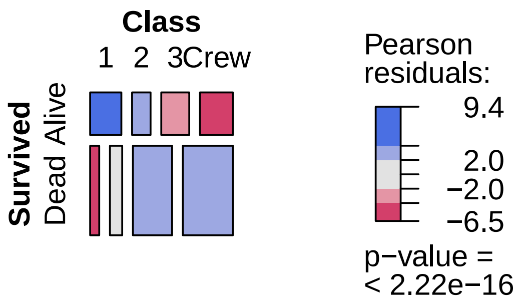
```
# Figure 3.4, page 75
gf_percents(~ Class, fill = ~ Survived, position = position_dodge(), data = Titanic)
```



```
# Figure 3.5
gf_percents(~ Survived, fill = ~ Class, position = "fill", data = Titanic)
```



```
# Figure 3.6, page 76
vcd::mosaic(tally(~ Survived + Class, data = Titanic),
           main = "Mosaic plot of Class by Survival",
           shade = TRUE)
```



Mosaic plot of Class by Survival

See the mosaic plots on page 77.

Section 3.4: Three Categorical Variables

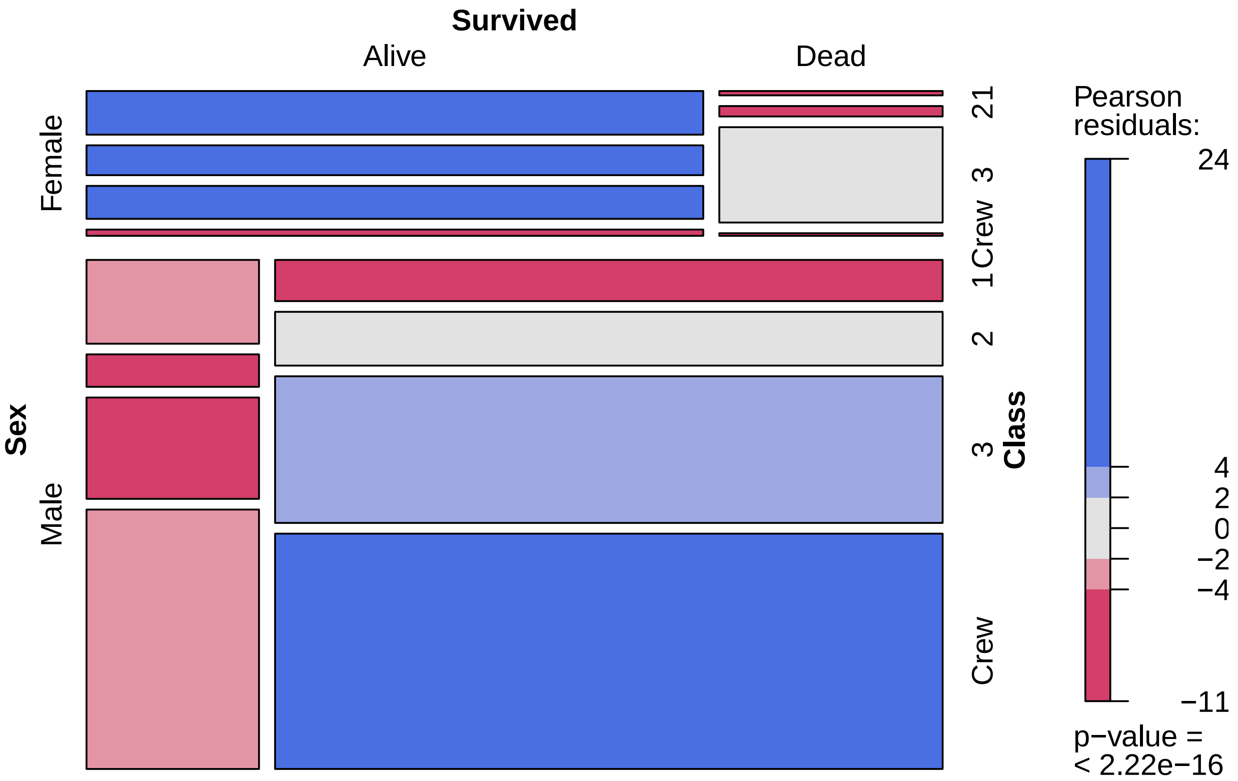
Consider using `useNA = "no"` option

```
tally(~ gender + cats_dogs_both + drugs_y_n, format = "percent", data = OKCupid)
```

```
## , , drugs_y_n = No
##
##         cats_dogs_both
## gender    Has Both   Has cats   Has dogs        <NA>
##      F   1.0243064  3.4199156  3.9437466 18.0187845
##      M   0.5922293  2.0819779  3.7769214 30.0719016
##
## , , drugs_y_n = Yes
##
##         cats_dogs_both
## gender    Has Both   Has cats   Has dogs        <NA>
##      F   0.2085314  0.8941828  0.6272626  2.9794972
##      M   0.1901807  0.8658225  0.9041923  6.9132342
##
## , , drugs_y_n = NA
##
##         cats_dogs_both
## gender    Has Both   Has cats   Has dogs        <NA>
##      F   0.2635837  1.3779757  1.1527618  6.3226732
##      M   0.1801712  1.0359842  1.3029044 11.8512587
```

Example 3.7: Looking for Associations Among Three Variables at Once

```
vcd::mosaic(tally(~ Sex + Survived + Class, data = Titanic), shade = TRUE)
```



Example 3.8: Simpson's Paradox: Gender Discrimination?

Here we demonstrate how to generate one of the tables on page 80.
XX Why do we have to have 825-512 can we just not put the difference?

```
# Create a dataframe from the counts
# http://mathemathinking.blogspot.com/2012/06/simpsons-paradox.html
Berk <- rbind(
  do(512) *       data.frame(admit = TRUE,  sex = "M", school = "A"),
  do(825 - 512) * data.frame(admit = FALSE, sex = "M", school = "A"),
  do(89) *        data.frame(admit = TRUE,  sex = "F", school = "A"),
  do(19) *        data.frame(admit = FALSE, sex = "F", school = "A")
)
```

In this case, `do(n)` creates `n` observations with the specified values in `data.frame()`. The `rbind()` function can then be used to combine the data frames into one.

```
tally(~ sex + admit, data = Berk)
```

```
##    admit
## sex TRUE FALSE
##   M  512   313
##   F   89    19
```

```
tally(~ admit | sex, format = "percent", data = Berk)
```

```
##        sex
## admit          M        F
##   TRUE  62.06061 82.40741
##   FALSE 37.93939 17.59259
```