The freqlist function

Tina Gunderson

29 December, 2016

Contents

Overview **2**
Sample dataset 2

The `freqlist` object **3**

Basic output using `summary` **4**

Rounding percentage digits or changing variable names for printing **6**

Additional examples **7**
Including combinations with frequencies of zero 7
Options for NA handling 8
Frequency counts and percentages subset by factor levels 9
Change labels on the fly 10
Using `xtable` to format and print `freqlist` results 11

Appendix: Notes regarding table options in R **13**
NAs 13
Table dimname names (dnn) 14

Overview

freqlist is a function meant to produce output similar to SAS's PROC FREQ procedure when using the /list option of the TABLE statement. freqlist provides options for handling missing or sparse data and can provide cumulative counts and percentages based on subgroups. It depends on the knitr package for printing.

Sample dataset

For our examples, we'll load the mockstudy data included with this package and use it to create a basic table. Because they have fewer levels, for brevity, we'll use the variables arm, sex, and mdquality.s to create the example table. We'll retain NAs in the table creation. See the appendix for notes regarding default NA handling and other useful information regarding tables in R.

```
# load the data
data(mockstudy)

# examine the data
str(mockstudy)
```

```
'data.frame':   1499 obs. of  14 variables:
 $ case       : int  110754 99706 105271 105001 112263 86205 99508 90158 88989 90515 ...
 $ age        : atomic  67 74 50 71 69 56 50 57 51 63 ...
  ..- attr(*, "label")= chr "Age in Years"
 $ arm        : atomic  F: FOLFOX A: IFL A: IFL G: IROX ...
  ..- attr(*, "label")= chr "Treatment Arm"
 $ sex        : Factor w/ 2 levels "Male","Female": 1 2 2 2 2 1 1 1 2 1 ...
 $ race       : atomic  Caucasian Caucasian Caucasian Caucasian ...
  ..- attr(*, "label")= chr "Race"
 $ fu.time    : int  922 270 175 128 233 120 369 421 387 363 ...
 $ fu.stat    : int  2 2 2 2 2 2 2 2 2 2 ...
 $ ps         : int  0 1 1 1 0 0 0 0 1 1 ...
 $ hgb        : num  11.5 10.7 11.1 12.6 13 10.2 13.3 12.1 13.8 12.1 ...
 $ bmi        : atomic  25.1 19.5 NA 29.4 26.4 ...
  ..- attr(*, "label")= chr "Body Mass Index (kg/m^2)"
 $ alk.phos   : int  160 290 700 771 350 569 162 152 231 492 ...
 $ ast        : int  35 52 100 68 35 27 16 12 25 18 ...
 $ mdquality.s: int  NA 1 1 1 NA 1 1 1 1 1 ...
 $ age.ord    : Ord.factor w/ 8 levels "10-19"<"20-29"<..: 6 7 4 7 6 5 4 5 5 6 ...
```

```
# retain NAs when creating the table using the useNA argument
tab.ex <- table(mockstudy[, c("arm", "sex", "mdquality.s")], useNA = "ifany")
```

The freqlist object

The freqlist function returns an object of class freqlist, which has three parts: freqlist, byVar, and labels.

- freqlist is a single data frame containing all contingency tables with calculated frequencies, cumulative frequencies, percentages, and cumulative percentages.
- byVar and labels are used in the summary method for subgroups and variable names, which will be covered in later examples.

```
noby <- freqlist(tab.ex)

str(noby)
```

```
List of 3
 $ freqlist:'data.frame':   18 obs. of  7 variables:
  ..$ arm        : Factor w/ 3 levels "A: IFL","F: FOLFOX",..: 1 1 1 1 1 1 2 2 2 2 ...
  ..$ sex        : Factor w/ 2 levels "Male","Female": 1 1 1 2 2 2 1 1 1 2 ...
  ..$ mdquality.s: Factor w/ 2 levels "0","1": 1 2 NA 1 2 NA 1 2 NA 1 ...
  ..$ Freq       : int [1:18] 29 214 34 12 118 21 31 285 95 21 ...
  ..$ cumFreq    : int [1:18] 29 243 277 289 407 428 459 744 839 860 ...
  ..$ freqPercent: num [1:18] 1.93 14.28 2.27 0.8 7.87 ...
  ..$ cumPercent : num [1:18] 1.93 16.21 18.48 19.28 27.15 ...
 $ byVar   : NULL
 $ labels  : NULL
 - attr(*, "class")= chr "freqlist"
```

```
# view the data frame portion of freqlist output
noby[["freqlist"]]
```

```
         arm    sex mdquality.s Freq cumFreq freqPercent cumPercent
1     A: IFL   Male           0   29      29        1.93       1.93
2     A: IFL   Male           1  214     243       14.28      16.21
3     A: IFL   Male        <NA>   34     277        2.27      18.48
4     A: IFL Female           0   12     289        0.80      19.28
5     A: IFL Female           1  118     407        7.87      27.15
6     A: IFL Female        <NA>   21     428        1.40      28.55
7  F: FOLFOX   Male           0   31     459        2.07      30.62
8  F: FOLFOX   Male           1  285     744       19.01      49.63
9  F: FOLFOX   Male        <NA>   95     839        6.34      55.97
10 F: FOLFOX Female           0   21     860        1.40      57.37
11 F: FOLFOX Female           1  198    1058       13.21      70.58
12 F: FOLFOX Female        <NA>   61    1119        4.07      74.65
13   G: IROX   Male           0   17    1136        1.13      75.78
14   G: IROX   Male           1  187    1323       12.47      88.26
15   G: IROX   Male        <NA>   24    1347        1.60      89.86
16   G: IROX Female           0   14    1361        0.93      90.79
17   G: IROX Female           1  121    1482        8.07      98.87
18   G: IROX Female        <NA>   17    1499        1.13     100.00
```

Basic output using `summary`

The `summary` method for `freqlist` relies on the `kable` function (in the `knitr` package) for printing. `knitr::kable` converts the output to markdown which can be printed in the console or easily rendered in Word, pdf, or html documents.

Note that you must supply `results="asis"` to properly format the markdown output.

```
summary(noby)
```

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	1.93	1.93
		1	214	243	14.28	16.21
		NA	34	277	2.27	18.48
	Female	0	12	289	0.80	19.28
		1	118	407	7.87	27.15
		NA	21	428	1.40	28.55
F: FOLFOX	Male	0	31	459	2.07	30.62
		1	285	744	19.01	49.63
		NA	95	839	6.34	55.97
	Female	0	21	860	1.40	57.37
		1	198	1058	13.21	70.58
		NA	61	1119	4.07	74.65
G: IROX	Male	0	17	1136	1.13	75.78
		1	187	1323	12.47	88.26
		NA	24	1347	1.60	89.86
	Female	0	14	1361	0.93	90.79
		1	121	1482	8.07	98.87
		NA	17	1499	1.13	100.00

Additional arguments (except digits) in the `kable` function can be passed through. Perhaps the most useful is `caption`.

```
summary(noby, caption = "Basic freqlist output")
```

Table 2: Basic freqlist output

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	1.93	1.93
		1	214	243	14.28	16.21
		NA	34	277	2.27	18.48
	Female	0	12	289	0.80	19.28
		1	118	407	7.87	27.15
		NA	21	428	1.40	28.55
F: FOLFOX	Male	0	31	459	2.07	30.62
		1	285	744	19.01	49.63
		NA	95	839	6.34	55.97
	Female	0	21	860	1.40	57.37
		1	198	1058	13.21	70.58
		NA	61	1119	4.07	74.65
G: IROX	Male	0	17	1136	1.13	75.78
		1	187	1323	12.47	88.26
		NA	24	1347	1.60	89.86

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
	Female	0	14	1361	0.93	90.79
		1	121	1482	8.07	98.87
		NA	17	1499	1.13	100.00

You can also easily pull out the `freqlist` data frame for more complicated formatting or manipulation (e.g. with another function such as `xtable` or `pander`). See below.

Rounding percentage digits or changing variable names for printing

The digits argument takes a single numeric value and controls the rounding of percentages in the output. The labelTranslations argument is a character vector whose length must be equal to the number of factors used in the table. Note: this does not change the names of the data frame in the freqlist object, only those used in printing. Both options are applied in the following example.

```r
withnames <- freqlist(tab.ex, labelTranslations = c("Treatment Arm", "Gender", "LASA QOL"),
    digits = 0)

summary(withnames)
```

Treatment Arm	Gender	LASA QOL	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	2	2
		1	214	243	14	16
		NA	34	277	2	18
	Female	0	12	289	1	19
		1	118	407	8	27
		NA	21	428	1	29
F: FOLFOX	Male	0	31	459	2	31
		1	285	744	19	50
		NA	95	839	6	56
	Female	0	21	860	1	57
		1	198	1058	13	71
		NA	61	1119	4	75
G: IROX	Male	0	17	1136	1	76
		1	187	1323	12	88
		NA	24	1347	2	90
	Female	0	14	1361	1	91
		1	121	1482	8	99
		NA	17	1499	1	100

Additional examples

Including combinations with frequencies of zero

The sparse argument takes a single logical value as input. The default option is FALSE. If set to TRUE, the sparse option will include combinations with frequencies of zero in the list of results. As our initial table did not have any such levels, we create a second table to use in our example.

```r
# we create a second table example to showcase the sparse argument
tab.sparse <- table(mockstudy[, c("race", "sex", "arm")])

nobysparse <- freqlist(tab.sparse, sparse = TRUE, digits = 1)
summary(nobysparse)
```

race	sex	arm	Freq	cumFreq	freqPercent	cumPercent
African-Am	Male	A: IFL	25	25	1.7	1.7
		F: FOLFOX	24	49	1.6	3.3
		G: IROX	16	65	1.1	4.4
	Female	A: IFL	14	79	0.9	5.3
		F: FOLFOX	25	104	1.7	7.0
		G: IROX	11	115	0.7	7.7
Asian	Male	A: IFL	0	115	0.0	7.7
		F: FOLFOX	10	125	0.7	8.4
		G: IROX	1	126	0.1	8.4
	Female	A: IFL	1	127	0.1	8.5
		F: FOLFOX	4	131	0.3	8.8
		G: IROX	2	133	0.1	8.9
Caucasian	Male	A: IFL	240	373	16.1	25.0
		F: FOLFOX	352	725	23.6	48.6
		G: IROX	195	920	13.1	61.7
	Female	A: IFL	131	1051	8.8	70.4
		F: FOLFOX	234	1285	15.7	86.1
		G: IROX	136	1421	9.1	95.2
Hawaii/Pacific	Male	A: IFL	1	1422	0.1	95.3
		F: FOLFOX	1	1423	0.1	95.4
		G: IROX	0	1423	0.0	95.4
	Female	A: IFL	0	1423	0.0	95.4
		F: FOLFOX	2	1425	0.1	95.5
		G: IROX	1	1426	0.1	95.6
Hispanic	Male	A: IFL	8	1434	0.5	96.1
		F: FOLFOX	17	1451	1.1	97.3
		G: IROX	12	1463	0.8	98.1
	Female	A: IFL	4	1467	0.3	98.3
		F: FOLFOX	11	1478	0.7	99.1
		G: IROX	2	1480	0.1	99.2
Native-Am/Alaska	Male	A: IFL	1	1481	0.1	99.3
		F: FOLFOX	0	1481	0.0	99.3
		G: IROX	2	1483	0.1	99.4
	Female	A: IFL	1	1484	0.1	99.5
		F: FOLFOX	1	1485	0.1	99.5
		G: IROX	0	1485	0.0	99.5
Other	Male	A: IFL	2	1487	0.1	99.7
		F: FOLFOX	2	1489	0.1	99.8
		G: IROX	1	1490	0.1	99.9

race	sex	arm	Freq	cumFreq	freqPercent	cumPercent
	Female	A: IFL	0	1490	0.0	99.9
		F: FOLFOX	2	1492	0.1	100.0
		G: IROX	0	1492	0.0	100.0

Options for NA handling

The various na.options allow you to include or exclude data with missing values for one or more factor levels in the counts and percentages as well as show the missing data but exclude it from the cumulative counts and percentages. The default option is to include all combinations with missing values.

```
summary(freqlist(tab.ex, na.options = "include"))
```

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	1.93	1.93
		1	214	243	14.28	16.21
		NA	34	277	2.27	18.48
	Female	0	12	289	0.80	19.28
		1	118	407	7.87	27.15
		NA	21	428	1.40	28.55
F: FOLFOX	Male	0	31	459	2.07	30.62
		1	285	744	19.01	49.63
		NA	95	839	6.34	55.97
	Female	0	21	860	1.40	57.37
		1	198	1058	13.21	70.58
		NA	61	1119	4.07	74.65
G: IROX	Male	0	17	1136	1.13	75.78
		1	187	1323	12.47	88.26
		NA	24	1347	1.60	89.86
	Female	0	14	1361	0.93	90.79
		1	121	1482	8.07	98.87
		NA	17	1499	1.13	100.00

```
summary(freqlist(tab.ex, na.options = "showexclude"))
```

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	2.33	2.33
		1	214	243	17.16	19.49
		NA	34	NA	NA	NA
	Female	0	12	255	0.96	20.45
		1	118	373	9.46	29.91
		NA	21	NA	NA	NA
F: FOLFOX	Male	0	31	404	2.49	32.40
		1	285	689	22.85	55.25
		NA	95	NA	NA	NA
	Female	0	21	710	1.68	56.94
		1	198	908	15.88	72.81
		NA	61	NA	NA	NA
G: IROX	Male	0	17	925	1.36	74.18
		1	187	1112	15.00	89.17
		NA	24	NA	NA	NA

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
	Female	0	14	1126	1.12	90.30
		1	121	1247	9.70	100.00
		NA	17	NA	NA	NA

```r
summary(freqlist(tab.ex, na.options = "remove"))
```

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	2.33	2.33
		1	214	243	17.16	19.49
	Female	0	12	255	0.96	20.45
		1	118	373	9.46	29.91
F: FOLFOX	Male	0	31	404	2.49	32.40
		1	285	689	22.85	55.25
	Female	0	21	710	1.68	56.94
		1	198	908	15.88	72.81
G: IROX	Male	0	17	925	1.36	74.18
		1	187	1112	15.00	89.17
	Female	0	14	1126	1.12	90.30
		1	121	1247	9.70	100.00

Frequency counts and percentages subset by factor levels

The groupBy argument internally subsets the data by the specified factor prior to calculating cumulative counts and percentages. By default, when used each subset will print in a separate table. Using the `single = TRUE` option when printing will collapse the subsetted result into a single table.

```r
withby <- freqlist(tab.ex, groupBy = c("arm", "sex"))
summary(withby)
```

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	10.47	10.47
		1	214	243	77.26	87.73
		NA	34	277	12.27	100.00

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Female	0	12	12	7.95	7.95
		1	118	130	78.15	86.09
		NA	21	151	13.91	100.00

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
F: FOLFOX	Male	0	31	31	7.54	7.54
		1	285	316	69.34	76.89
		NA	95	411	23.11	100.00

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
F: FOLFOX	Female	0	21	21	7.50	7.50
		1	198	219	70.71	78.21
		NA	61	280	21.79	100.00

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
G: IROX	Male	0	17	17	7.46	7.46
		1	187	204	82.02	89.47
		NA	24	228	10.53	100.00

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
G: IROX	Female	0	14	14	9.21	9.21
		1	121	135	79.61	88.82
		NA	17	152	11.18	100.00

```r
# using the single = TRUE argument will collapse results into a single table for
# printing
summary(withby, single = TRUE)
```

arm	sex	mdquality.s	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	10.47	10.47
		1	214	243	77.26	87.73
		NA	34	277	12.27	100.00
	Female	0	12	12	7.95	7.95
		1	118	130	78.15	86.09
		NA	21	151	13.91	100.00
F: FOLFOX	Male	0	31	31	7.54	7.54
		1	285	316	69.34	76.89
		NA	95	411	23.11	100.00
	Female	0	21	21	7.50	7.50
		1	198	219	70.71	78.21
		NA	61	280	21.79	100.00
G: IROX	Male	0	17	17	7.46	7.46
		1	187	204	82.02	89.47
		NA	24	228	10.53	100.00
	Female	0	14	14	9.21	9.21
		1	121	135	79.61	88.82
		NA	17	152	11.18	100.00

Change labels on the fly

At this time, the labels can be changed just for the variables (e.g. not the frequency columns).

```r
labels(noby) <- c("Arm", "Sex", "OtherThing")
summary(noby)
```

Arm	Sex	OtherThing	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	1.93	1.93
		1	214	243	14.28	16.21
		NA	34	277	2.27	18.48
	Female	0	12	289	0.80	19.28
		1	118	407	7.87	27.15
		NA	21	428	1.40	28.55
F: FOLFOX	Male	0	31	459	2.07	30.62
		1	285	744	19.01	49.63
		NA	95	839	6.34	55.97
	Female	0	21	860	1.40	57.37
		1	198	1058	13.21	70.58
		NA	61	1119	4.07	74.65
G: IROX	Male	0	17	1136	1.13	75.78
		1	187	1323	12.47	88.26
		NA	24	1347	1.60	89.86
	Female	0	14	1361	0.93	90.79
		1	121	1482	8.07	98.87
		NA	17	1499	1.13	100.00

You can also supply `labelTranslations` to `summary`.

```r
summary(noby, labelTranslations = c("Hi there", "What up", "Bye"))
```

Hi there	What up	Bye	Freq	cumFreq	freqPercent	cumPercent
A: IFL	Male	0	29	29	1.93	1.93
		1	214	243	14.28	16.21
		NA	34	277	2.27	18.48
	Female	0	12	289	0.80	19.28
		1	118	407	7.87	27.15
		NA	21	428	1.40	28.55
F: FOLFOX	Male	0	31	459	2.07	30.62
		1	285	744	19.01	49.63
		NA	95	839	6.34	55.97
	Female	0	21	860	1.40	57.37
		1	198	1058	13.21	70.58
		NA	61	1119	4.07	74.65
G: IROX	Male	0	17	1136	1.13	75.78
		1	187	1323	12.47	88.26
		NA	24	1347	1.60	89.86
	Female	0	14	1361	0.93	90.79
		1	121	1482	8.07	98.87
		NA	17	1499	1.13	100.00

Using `xtable` to format and print `freqlist` results

Fair warning: `xtable` has kind of a steep learning curve. These examples are given without explanation for more advanced users.

```r
require(xtable)
```

```
Loading required package: xtable
```

```r
# turn off xtable header
options(xtable.comment = FALSE)

# set up custom function for xtable text
italic <- function(x) {
    paste0("{\\emph{ ", x, "}}")
}
```

```r
xftbl <- xtable(noby[["freqlist"]], caption = "xtable formatted output of freqlist data frame",
    align = "|r|r|r|r|c|c|c|r|")

# change the column names
names(xftbl)[1:3] <- c("Arm", "Gender", "LASA QOL")

print(xftbl, sanitize.colnames.function = italic, include.rownames = FALSE)
```

Arm	*Gender*	*LASA QOL*	*Freq*	*cumFreq*	*freqPercent*	*cumPercent*
A: IFL	Male	0	29	29	1.93	1.93
A: IFL	Male	1	214	243	14.28	16.21
A: IFL	Male		34	277	2.27	18.48
A: IFL	Female	0	12	289	0.80	19.28
A: IFL	Female	1	118	407	7.87	27.15
A: IFL	Female		21	428	1.40	28.55
F: FOLFOX	Male	0	31	459	2.07	30.62
F: FOLFOX	Male	1	285	744	19.01	49.63
F: FOLFOX	Male		95	839	6.34	55.97
F: FOLFOX	Female	0	21	860	1.40	57.37
F: FOLFOX	Female	1	198	1058	13.21	70.58
F: FOLFOX	Female		61	1119	4.07	74.65
G: IROX	Male	0	17	1136	1.13	75.78
G: IROX	Male	1	187	1323	12.47	88.26
G: IROX	Male		24	1347	1.60	89.86
G: IROX	Female	0	14	1361	0.93	90.79
G: IROX	Female	1	121	1482	8.07	98.87
G: IROX	Female		17	1499	1.13	100.00

Table 17: xtable formatted output of freqlist data frame

Appendix: Notes regarding table options in R

NAs

There are several widely used options for basic tables in R. The `table` function in base R is probably the most common; by default it excludes NA values. You can change NA handling in `base::table` using the useNA or exclude arguments.

```
# base table default removes NAs
tab.d1 <- base::table(mockstudy[, c("arm", "sex", "mdquality.s")], useNA = "ifany")
tab.d1
```

```
, , mdquality.s = 0

           sex
arm          Male Female
  A: IFL       29     12
  F: FOLFOX    31     21
  G: IROX      17     14

, , mdquality.s = 1

           sex
arm          Male Female
  A: IFL      214    118
  F: FOLFOX   285    198
  G: IROX     187    121

, , mdquality.s = NA

           sex
arm          Male Female
  A: IFL       34     21
  F: FOLFOX    95     61
  G: IROX      24     17
```

`xtabs` is similar to `table`, but uses a formula-based syntax. However, there is not an option for retaining NAs in the `xtabs` function; instead, NAs must be added to each level of the factor where present using the `addNA` function.

```
# without specifying addNA
tab.d2 <- xtabs(formula = ~arm + sex + mdquality.s, data = mockstudy)
tab.d2
```

```
, , mdquality.s = 0

           sex
arm          Male Female
  A: IFL       29     12
  F: FOLFOX    31     21
  G: IROX      17     14

, , mdquality.s = 1

           sex
arm          Male Female
```

```
  A: IFL     214    118
  F: FOLFOX  285    198
  G: IROX    187    121
```

```r
# now with addNA
tab.d3 <- xtabs(~arm + sex + addNA(mdquality.s), data = mockstudy)
tab.d3
```

```
, , addNA(mdquality.s) = 0

           sex
arm          Male Female
  A: IFL       29     12
  F: FOLFOX    31     21
  G: IROX      17     14

, , addNA(mdquality.s) = 1

           sex
arm          Male Female
  A: IFL      214    118
  F: FOLFOX   285    198
  G: IROX     187    121

, , addNA(mdquality.s) = NA

           sex
arm          Male Female
  A: IFL       34     21
  F: FOLFOX    95     61
  G: IROX      24     17
```

Table dimname names (dnn)

Supplying a data.frame to the `table` function without giving columns individually will create a contingency table using all variables in the data.frame.

However, if the columns of a data.frame or matrix are supplied separately (i.e., as vectors), column names will not be preserved.

```r
# providing variables separately (as vectors) drops column names
tab.d4 <- base::table(mockstudy[, "arm"], mockstudy[, "sex"], mockstudy[, "mdquality.s"])
tab.d4
```

```
, ,  = 0


             Male Female
  A: IFL       29     12
  F: FOLFOX    31     21
  G: IROX      17     14

, ,  = 1
```

```
           Male Female
  A: IFL     214    118
  F: FOLFOX  285    198
  G: IROX    187    121
```

If desired, you can use the `dnn` argument to pass variable names.

```
# add the column name labels back using dnn option in base::table
tab.dnn <- base::table(mockstudy[, "arm"], mockstudy[, "sex"], mockstudy[, "mdquality.s"],
    dnn = c("Amy", "Susan", "George"))
tab.dnn
```

```
, , George = 0

           Susan
Amy          Male Female
  A: IFL       29     12
  F: FOLFOX    31     21
  G: IROX      17     14

, , George = 1

           Susan
Amy          Male Female
  A: IFL      214    118
  F: FOLFOX   285    198
  G: IROX     187    121
```

If using `freqlist`, you can provide the labels directly to `freqlist` or to `summary` using `labelTranslations`.