Chapter 1

Tables

Tabulation is a useful tool to describe the distribution of categorical data. R provides several possibilities to create tables. We use the data in *BackPain.csv* to demonstrate the concepts.

`table()` is the basic function for tabulation. It takes vectors as arguments to create tables, corresponding to the variables to be tabulated.

`xtabs()` is a convenient function for 1 and 2 way tabulations. It takes two arguments, the first starts with '∼' followed by the names of the columns to be tabulated. Multiple columns are separated by '+' . The second argument is the data that contains the columns. We focus on `xtabs()` as it often is more convenient than `table()`, but the two are largely similar.

First we load the *BackPain* dataset. Consider the following examples:

```
d <- read.csv("../data/BackPain.csv")
```

```
table(d$sex)

##
## Female   Male
##  18456  15666
```

This is eqaivalent to

```
xtabs(~sex, d)

## sex
## Female   Male
##  18456  15666
```

Cross tabulation of two variables in with `xtabs()` can be done like this:

```
xtabs(~sex+asthma, d) # two way with sex and asthma

##         asthma
## sex                no   yes
##   Female   415 16968  1073
##   Male     338 14279  1049
```

You may wonder what the ∼ means. This is called a *formula* and is a special way in R to specify variables. You will later encounter it in a statistical context as well. For now, it is enough to know that some functions like `xtabs()` work with formulas and that you use them to specify the variables. For `xtabs()`, the variables to be tabulated are specified after the ∼ sign and are separated by a +. Normally, these are the columns of a dataframe that is provided as the *data* argument of `xtabs()`.

Own experimentation

Perform the same crosstabulation of sex and asthma using the `table()` function.

1.1 Enveloping

1.1.1 Summary

`xtabs() results` can be inserted into the `summary()` function to give chi square statistics. Note how the the empty string "" is also counted (the leftmost column). We can exclude values using the exclude argument in `xtabs()`:

```
summary(xtabs(~sex+asthma+age, d, exclude = ""))    # calculate chi square statistics

## Call: xtabs(formula = ~sex + asthma + age, data = d, exclude = "")
## Number of cases in table: 33369
## Number of factors: 3
## Test for independence of all factors:
##  Chisq = 369.6, df = 178, p-value = 1.649e-15
##  Chi-squared approximation may be incorrect
```

Convenienty, the `summary()` function also computes a chi-square test of independence for the provided variables. The null hypothesis in that case is that the variables are independent. Because of the high *Chisq* the null hypothesis may be rejected. You will later learn much more on how to calculate test statistics with R.

1.1.2 Proportion

It can also be inserted into the `prop.table()` function to display proportions. Because in this example the numbers are that large, we use the round function with 2 decimal places to round the proportions:

```
# report proportions instead of frequencies
round(prop.table(xtabs(~sex+asthma, d, exclude = "")),2)

##         asthma
## sex        no  yes
##   Female 0.51 0.03
##   Male   0.43 0.03

# gives percentages
round(prop.table(xtabs(~sex+asthma, d, exclude = "")),2)*100

##         asthma
## sex      no yes
##   Female 51   3
##   Male   43   3
```

To get row and column totals, we can use `addmargins()` function:

```
addmargins(xtabs(~sex+asthma, d, exclude = "")) # add both row and column totals

##         asthma
## sex         no   yes   Sum
##   Female 16968  1073 18041
##   Male   14279  1049 15328
##   Sum    31247  2122 33369
```

Own experimentation

Use the prop.table function to create a table of percentages of `physical` and `country`, rounded to two places. How could you check that the valules sum up to 100%?

1.1.3 Tabulation of more variables

2-way or crass tabulation is the most straight-forward way of tabulation, given it has the same dimensions as the output (paper and computerscreens). However, its possibly to tabulate according to an arbitrary number of variables. Tables with more than two dimensions need to be collapsed to two dimensions for display.

`xtabs` can create 3-way contingency table:

```
xtabs(~sex+asthma+diabetes, d)

## , , diabetes =
##
##         asthma
## sex               no   yes
##   Female   317    14     1
##   Male     249    18     0
##
## , , diabetes = no
##
##         asthma
## sex               no   yes
##   Female    91 15500   932
##   Male      82 13313   959
##
## , , diabetes = yes
##
##         asthma
## sex               no   yes
##   Female     7  1454   140
##   Male       7   948    90
```

However, the output is not nicely formatted, or put otherwise, the collapsing is not very sophisticated. Feeding `xtabs()` to `ftable()` can solve this:

```
ftable(xtabs(~sex+asthma+wealthQ,d))

##               wealthQ      Q1 poorest   Q2   Q3   Q4 Q5 richest
## sex    asthma
## Female                   2        111   91   68   72         71
##        no               67       3342 3340 3350 3443       3426
##        yes               7        227  240  242  185        172
## Male                        2          76   67   62   55          76
##        no                  51        2497 2727 2708 3052        3244
##        yes                  3         241  230  215  202         158
```

Its also possible to directly use a formula and `ftable()`. `ftable()` expects variables on the left-hand side of the formula also, they indicate along what variables the table columns will be constructed. The right-hand side variables in the formula indicate the variables to contain the row categories:

```
ftable(sex+asthma~wealthQ+diabetes,d)

##                     sex    Female           Male
##                     asthma         no  yes         no  yes
## wealthQ    diabetes
##                              0    0    1    1    0    0
##            no                2   64    6    1   46    2
##            yes               0    3    0    0    5    1
## Q1 poorest                  92    1    0   65    5    0
##            no               19 3148  206   11 2420  227
##            yes               0  193   21    0   72   14
## Q2                          72    1    0   47    2    0
##            no               17 3105  221   17 2601  217
##            yes               2  234   19    3  124   13
## Q3                          50    5    0   41    7    0
##            no               17 3028  201   19 2534  191
##            yes               1  317   41    2  167   24
## Q4                          51    4    0   39    2    0
##            no               19 3106  160   15 2797  179
##            yes               2  333   25    1  253   23
## Q5 richest                  52    3    0   56    2    0
##            no               17 3049  138   19 2915  143
##            yes               2  374   34    1  327   15
```

Own experimentation

Try to use the `kable()` function in the *knitr* package to format 3 way tables generated using `xtabs()`. Unfortunately, `kable()` is not compatible with `ftable()`.