

Chapter 1

First chapter

1.1 Exercises

Exercise 1.1 The data set *UCBAdmissions* is a 3-way table of frequencies classified by **Admit**, **Gender**, and **Dept**.

- (a) Find the total number of cases contained in this table.
Use `sum(UCBAdmissions)`
- (b) For each department, find the total number of applicants.
- (c) For each department, find the overall proportion of applicants who were admitted.
- (d) Construct a tabular display of department (rows) and gender (columns), showing the proportion of applicants in each cell who were admitted relative to the total applicants in that cell.

Answer:

- (a) Use `sum(UCBAdmissions)`.
- (b) Use `margin.table(UCBAdmissions, 3)` to find the marginal total for the third dimension (`dept`).
- (c)

Exercise 1.2 The data set *DanishWelfare* in `vcd (?)` gives a 4-way, $3 \times 4 \times 3 \times 5$ table as a data frame in frequency form, containing the variable **Freq** and four factors, **Alcohol**, **Income**, **Status**, and **Urban**. The variable **Alcohol** can be considered as the response variable, and the others as possible predictors.

- (a) Find the total number of cases represented in this table.
- (b) In this form, the variables **Alcohol** and **Income** should arguably be considered *ordered* factors. Change them to make them ordered.
- (c) Convert this data frame to table form, **DanishWelfare.tab**, a 4-way array containing the frequencies with appropriate variable names and level names.
- (d) The variable **Urban** has 5 categories. Find the total frequencies in each of these. How would you collapse the table to have only two categories, **City**, **Non-city**?
- (e) Use `strutable()` or `ftable()` to produce a pleasing flattened display of the frequencies in the 4-way table. Choose the variables used as row and column variables to make it easier to compare levels of **Alcohol** across the other factors.

Hint:

- (a) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin.
- (b) Use `ordered()` or `as.ordered()` on the factor variable. `str()` will then show them as `Ord.factor`.
- (c) Use `xtabs()` with **Freq** as the response.

```
DanishWelfare.tab <- xtabs(Freq ~ ., data = DanishWelfare)
str(DanishWelfare.tab)
```