

### 1. Cystic Fibrosis dataset

On the class website you will find the file `cystfibr.txt` which contains a set of measurements on a set of individuals with cystic fibrosis. Save the file locally onto your computer.

- (a) Take a look at this file using a text editor like Wordpad or Notepad. Read the data into R using `read.table()`; call the resulting dataframe `cf`.

Check that the data have been formatted correctly by typing:

```
> head(cf)
```

- (b) Obtain the following:

- (i) the number of individuals in the dataset;
- (ii) the number of variables measured on each individual;
- (iii) a vector of the names of the variables measured on each individual;
- (iv) the mean, median, standard deviation and range of each of the variables (use `apply()`);

- (c) Create the following data frames and for each one calculate the mean of each of the variables

- (i) a new data frame containing just individuals older than 15
- (ii) a new data frame containing just individuals with `bmp` in the interval `[70,90]`
- (iii) a new data frame containing just individuals with either `FEV1` greater than 30 or `RV` greater than 300.

### 2. Data Frames

Load the `MASS` library and take a look at the dataset called `survey`.

```
> library(MASS)
> head(survey)
```

You can look at the documentation:

```
> ?survey
```

and get a brief summary of each variable:

```
> summary(survey)
```

- (a) Find the mean pulse rate of the students. What goes wrong here?

The vector of pulses stored in `survey` contains some entries which are labelled `NA`. This is used in R to represent **missing data**.

- (b) Try looking at the documentation for `mean()` to see how to get around this.
- (c) The ages are recorded as fractions representing a number of months. Change that columns of the data frame so that it contains whole years (the `floor()` command may be useful).

(d) Find the mean pulse rate for students under 20.

**Subsetting.** Suppose I want to obtain the records of students who are over 190 cm tall. Since data frames allow me to subset just like a matrix, I might just think of typing:

```
> survey[survey$Height > 190, ]
```

What goes wrong here? Try instead the following:

```
> subset(survey, Height > 190)
```

The subset function ignores missing values, which is usually the behaviour we would prefer. We can also select only some of the fields of the data frame if we prefer:

```
> subset(survey, Height > 190, select = c("Pulse", "Clap"))
```

Recall that the & operator does a point-wise logical ‘and’ comparison.

```
> subset(survey, (Pulse > 70) & (Smoke == "Heavy"))
```

Similarly, | is for ‘or’, and ! for ‘not’.

```
> with(survey, (Pulse > 70) | (Pulse < 45))
> !(survey$Age > 30)
```

- (e) Find the mean age of students who write with their right hand.
- (f) What proportion of left handers do not clap with their left hand on top?
- (g) Using the plot() command, plot the pulse of the subjects against their age. Try subtracting 10 from the age and taking the logarithm (using the function log()), to obtain a slightly clearer picture.

### 3. Factors

Take a look at the birthwt data from the MASS package.

- (a) How is race stored in these data? Is this sensible?
- (b) Turn this into a factor with level names as indicated in the documentation. To do this, look at the documentation for the function factor().
- (c) Use the table() command to count the number of babies of each race.
- (d) Try the following command:

```
> tab = with(birthwt, table(smoke, low))
```

What do the results in tab suggest?

- 4. **Tables** An 18×8 foot area of field planted with maize was split into 3×1 foot rectangles. The beetlelarva.txt data record the  $n = 48$  counts of Japanese beetle larvae found in the top foot of soil of each rectangle. Load it using the connection

```
> df <- read.table("http://www.stats.ox.ac.uk/~nicholls/PartASSP/beetlelarva.txt")
```

The beetles are spread by the planting and ploughing process. In which direction, **northing** or **easting**, would you say this field was ploughed? In order to answer this question convert the data to a table with

```
> tb <- xtabs(count ~ easting + northing, data = df)
```

Look at `tb` and try plotting `df` and `tb`. Can you guess?

5. **\*Gaussian Random Walk.** Write a function which simulates a Gaussian random walk with `n` steps. (In other words,  $X_0 = 0$  and  $X_i - X_{i-1} \sim N(0, 1)$  independently.)

Generate and plot the walk for `n = 1000` (use `type="l"`).

Try to vectorise your code (if it is already vectorised, try constructing a naïve version with a `for()` loop). Compare the speed of the vectorised and unvectorised versions by generating a random walk of length 50,000