1 Building a simple data package for R

Suppose that we wish to make a package containing data sets only available in-house or on CRAN. This is often done for the data sets in the examples and exercises of a text book. For illustration, I will consider various data sets used in the Statistics Department's Masters exams over the years.

The simplest approach is to create the data files and then use the function `package.skeleton` to create the package. Even if you only make data sets available within your own research group, it will have the advantage that others can quickly install the package and have access to the data in the "usual" way.

2 Data files

2.1 `bmd` data

One of the simplest data files is `bmd.dat` in Fall 2003 master's exam, stored as tab-separated values in the directory `/p/stat/Data/MS.exam/f03/` on the AFS file system. We use the `read.delim` function to read the data. I usually check the structure immediately and decide if I need to post-process some of the columns.

```
> dir <- "/p/stat/Data/MS.exam"
> str(bmd <- read.delim(file.path(dir, "f03", "bmd.dat")))
```

```
'data.frame':        126 obs. of  15 variables:
 $ ID     : Factor w/ 125 levels "","F02","F03",..: 2 3 4 5 6 7 8 9 10 11 ...
 $ Init   : Factor w/ 123 levels "","AES","ALP",..: 93 84 81 95 94 64 45 65 30 92 ...
 $ Age    : int  64 78 59 74 51 68 57 57 83 50 ...
 $ Gender : Factor w/ 3 levels "","F","M": 2 2 2 2 2 2 2 2 2 2 ...
 $ Height : num  161 169 169 163 168 ...
 $ Weight : int  251 181 165 122 129 197 131 157 148 117 ...
 $ L1.L4T : num  2.6 1.8 0.8 1.4 -1.6 -2.6 -1.6 1.8 -0.6 -2.3 ...
 $ X      : logi  NA NA NA NA NA NA ...
 $ INeckT : num  0.1 -1 0.1 0.2 -1.2 -0.5 0 1 -1.6 -1.3 ...
 $ ITrochT: num  0.3 -0.6 0.6 -0.2 -2 -0.1 -1 1.9 -1.6 -0.8 ...
 $ ITotalT: num  0.8 -0.8 0.2 0.1 -1.7 -0.5 -0.7 1 -1.8 -1.1 ...
 $ DNeckT : num  0.1 -0.9 0.1 0.1 -1.4 -0.4 0.1 1.1 -1.6 -1.6 ...
 $ DTrochT: num  0.6 -0.6 0.8 -0.1 -1.9 -0.2 -0.8 1.8 -1.5 -0.7 ...
 $ DTotalT: num  0.8 -0.8 0.2 0.1 -1.7 -0.5 -0.7 1 -1.8 -1.1 ...
 $ X.1    : logi  NA NA NA NA NA NA ...
```

Immediately we can see some problems, although fewer problems than usually encountered. The factors all have a level of `""` indicating that there are probably some completely blank lines in the data. Also there are two superfluous variables called `"X"` and `"X.1"` which are likely the result of extra tab characters on the lines.

First we handle the blank columns.

```
> bmd <- bmd[, names(bmd)[!names(bmd) %in% c("X","X.1")]]
```

To assure ourselves that the lines with missing `ID` are indeed blank we consider the subset of the data frame corresponding to these rows.

```
> subset(bmd, Gender=="")

   ID Init Age Gender Height Weight L1.L4T INeckT ITrochT ITotalT DNeckT DTrochT
73          NA             NA     NA     NA     NA      NA      NA     NA      NA
74          NA             NA     NA     NA     NA      NA      NA     NA      NA
   DTotalT
73      NA
74      NA
```

They are indeed blank in all the variables so we remove them

```
> str(bmd <- subset(bmd, Gender != ""))

'data.frame':        124 obs. of  13 variables:
 $ ID     : Factor w/ 125 levels "","F02","F03",..: 2 3 4 5 6 7 8 9 10 11 ...
 $ Init   : Factor w/ 123 levels "","AES","ALP",..: 93 84 81 95 94 64 45 65 30 92 ...
 $ Age    : int  64 78 59 74 51 68 57 57 83 50 ...
 $ Gender : Factor w/ 3 levels "","F","M": 2 2 2 2 2 2 2 2 2 2 ...
 $ Height : num  161 169 169 163 168 ...
 $ Weight : int  251 181 165 122 129 197 131 157 148 117 ...
 $ L1.L4T : num  2.6 1.8 0.8 1.4 -1.6 -2.6 -1.6 1.8 -0.6 -2.3 ...
 $ INeckT : num  0.1 -1 0.1 0.2 -1.2 -0.5 0 1 -1.6 -1.3 ...
 $ ITrochT: num  0.3 -0.6 0.6 -0.2 -2 -0.1 -1 1.9 -1.6 -0.8 ...
 $ ITotalT: num  0.8 -0.8 0.2 0.1 -1.7 -0.5 -0.7 1 -1.8 -1.1 ...
 $ DNeckT : num  0.1 -0.9 0.1 0.1 -1.4 -0.4 0.1 1.1 -1.6 -1.6 ...
 $ DTrochT: num  0.6 -0.6 0.8 -0.1 -1.9 -0.2 -0.8 1.8 -1.5 -0.7 ...
 $ DTotalT: num  0.8 -0.8 0.2 0.1 -1.7 -0.5 -0.7 1 -1.8 -1.1 ...
```

Notice that the factors still have the blank level, even though we have eliminated the observations at that level. To clean things up we apply `factor` to those variables.

```
> str(bmd <- within(bmd,
+               {
+                   ID <- factor(ID)
+                   Init <- factor(Init)
+                   Gender <- factor(Gender)
+               }))

'data.frame':        124 obs. of  13 variables:
 $ ID     : Factor w/ 124 levels "F02","F03","F04",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Init   : Factor w/ 122 levels "AES","ALP","ALS",..: 92 83 80 94 93 63 44 64 29 9..
```

```
 $ Age    : int  64 78 59 74 51 68 57 57 83 50 ...
 $ Gender : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
 $ Height : num  161 169 169 163 168 ...
 $ Weight : int  251 181 165 122 129 197 131 157 148 117 ...
 $ L1.L4T : num  2.6 1.8 0.8 1.4 -1.6 -2.6 -1.6 1.8 -0.6 -2.3 ...
 $ INeckT : num  0.1 -1 0.1 0.2 -1.2 -0.5 0 1 -1.6 -1.3 ...
 $ ITrochT: num  0.3 -0.6 0.6 -0.2 -2 -0.1 -1 1.9 -1.6 -0.8 ...
 $ ITotalT: num  0.8 -0.8 0.2 0.1 -1.7 -0.5 -0.7 1 -1.8 -1.1 ...
 $ DNeckT : num  0.1 -0.9 0.1 0.1 -1.4 -0.4 0.1 1.1 -1.6 -1.6 ...
 $ DTrochT: num  0.6 -0.6 0.8 -0.1 -1.9 -0.2 -0.8 1.8 -1.5 -0.7 ...
 $ DTotalT: num  0.8 -0.8 0.2 0.1 -1.7 -0.5 -0.7 1 -1.8 -1.1 ...
```

```
> summary(bmd)
```

```
      ID           Init          Age        Gender     Height          Weight
 F02    :  1   MJS    :  2   Min.   :50.00   F:72   Min.   :149.1   Min.   : 69.0
 F03    :  1   RLS    :  2   1st Qu.:63.00   M:52   1st Qu.:161.4   1st Qu.:136.8
 F04    :  1   AES    :  1   Median :73.00          Median :167.0   Median :158.5
 F05    :  1   ALP    :  1   Mean   :70.73          Mean   :167.6   Mean   :159.9
 F06    :  1   ALS    :  1   3rd Qu.:78.00          3rd Qu.:173.8   3rd Qu.:178.2
 F07    :  1   BAC    :  1   Max.   :93.00          Max.   :188.2   Max.   :286.0
 (Other):118   (Other):116
    L1.L4T            INeckT          ITrochT          ITotalT
 Min.   :-4.5000   Min.   :-3.700   Min.   :-3.400   Min.   :-3.2000
 1st Qu.:-1.9250   1st Qu.:-1.900   1st Qu.:-1.325   1st Qu.:-1.6000
 Median :-0.5500   Median :-1.200   Median :-0.500   Median :-0.8000
 Mean   :-0.4363   Mean   :-1.119   Mean   :-0.504   Mean   :-0.7718
 3rd Qu.: 0.7250   3rd Qu.:-0.300   3rd Qu.: 0.300   3rd Qu.: 0.0250
 Max.   : 4.6000   Max.   : 1.400   Max.   : 2.600   Max.   : 1.6000
    DNeckT          DTrochT          DTotalT
 Min.   :-3.700   Min.   :-3.200   Min.   :-3.2000
 1st Qu.:-1.900   1st Qu.:-1.200   1st Qu.:-1.6000
 Median :-1.350   Median :-0.300   Median :-0.8000
 Mean   :-1.154   Mean   :-0.346   Mean   :-0.7718
 3rd Qu.:-0.400   3rd Qu.: 0.400   3rd Qu.: 0.0250
 Max.   : 1.200   Max.   : 3.000   Max.   : 1.6000
```

It looks like one of those "mixed units" studies where the height is recorded in cm. and the weight is recorded in pounds.

2.2 `feltys` data

Another straightforward data set is `feltys.txt` from fall 2005. This is also a tab-separated values file.

```
> str(feltys <- read.delim(file.path(dir, "f05", "feltys.txt")))

'data.frame':        1115 obs. of  4 variables:
 $ ScrSSN  : int  22941661 43531010 47721177 57508613 104124546 111621660 121426267..
 $ Gender  : Factor w/ 2 levels "FEMALE","MALE": 2 2 2 2 2 2 2 2 2 2 ...
 $ Age     : int  56 76 54 63 52 61 62 84 69 79 ...
 $ YearHosp: int  4 4 4 4 4 4 4 4 4 4 ...

> summary(feltys)

     ScrSSN             Gender          Age           YearHosp
 Min.   :   662337   FEMALE:  34   Min.   :20.00   Min.   : 0.00
 1st Qu.:222908921   MALE  :1081   1st Qu.:62.00   1st Qu.:86.00
 Median :478593206                 Median :67.00   Median :89.00
 Mean   :486819846                 Mean   :66.63   Mean   :79.87
 3rd Qu.:757649932                 3rd Qu.:73.00   3rd Qu.:93.00
 Max.   :999768223                 Max.   :91.00   Max.   :99.00
```

The `ScrSSN` columns is a case identifier (presumably a scrambled social security number) and the `YearHosp` appears to be a year without the century, meaning that 2000 sorts before 1999. Consider the unique values

```
> xtabs(~ YearHosp, feltys)

YearHosp
  0   1   2   3   4  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99
 30  30  19  27  27 105 123  85  86  83  67  64  59  50  59  40  45  50  35  31
```

We repair those variables

```
> str(feltys <- within(feltys,
+                  {
+                      ScrSSN <- factor(ScrSSN)
+                      YearHosp <- ordered(ifelse(YearHosp < 85,
+                                                 2000, 1900) + YearHosp)
+                  }))

'data.frame':        1115 obs. of  4 variables:
 $ ScrSSN  : Factor w/ 686 levels "662337","2861467",..: 17 32 42 47 79 85 92 104 1..
 $ Gender  : Factor w/ 2 levels "FEMALE","MALE": 2 2 2 2 2 2 2 2 2 2 ...
 $ Age     : int  56 76 54 63 52 61 62 84 69 79 ...
 $ YearHosp: Ord.factor w/ 20 levels "1985"<"1986"<..: 20 20 20 20 20 20 20 20 20 2..

> summary(feltys)
```

```
       ScrSSN        Gender          Age           YearHosp
 143196520:  10   FEMALE:  34   Min.   :20.00   1986   :123
 545443700:   9   MALE  :1081   1st Qu.:62.00   1985   :105
 40239436 :   8                 Median :67.00   1988   : 86
 812344720:   8                 Mean   :66.63   1987   : 85
 3996116  :   7                 3rd Qu.:73.00   1989   : 83
 201004127:   7                 Max.   :91.00   1990   : 67
 (Other)  :1066                                 (Other):566
```

```
> subset(feltys, ScrSSN == 143196520)
```

```
       ScrSSN Gender Age YearHosp
170 143196520   MALE  74     1998
208 143196520   MALE  74     1997
303 143196520   MALE  72     1995
346 143196520   MALE  71     1994
402 143196520   MALE  70     1993
452 143196520   MALE  69     1992
513 143196520   MALE  68     1991
579 143196520   MALE  67     1990
648 143196520   MALE  65     1989
730 143196520   MALE  65     1988
```

An associated data set is

```
> str(raprev <- read.delim(file.path(dir, "f05", "raprev.txt")))
```

```
'data.frame':        20 obs. of  12 variables:
 $ Year              : int  2004 2003 2002 2001 2000 1999 1998 1997 1996 1995 ...
 $ Hosp_Persons      : int  366948 359247 359617 358573 353249 359608 371273 406556..
 $ Hosp_Women        : int  17134 15971 15740 15355 14488 14103 13741 14268 15974 1..
 $ RA_Persons        : int  3073 3091 3006 3037 2944 3037 3085 3291 3652 3923 ...
 $ RA_Women          : int  230 193 192 208 171 173 180 198 211 216 ...
 $ Felty_Persons     : int  27 27 19 30 30 31 35 50 45 40 ...
 $ Felty_Women       : int  0 1 1 1 0 1 1 3 0 0 ...
 $ Pleuritis_Persons : int  118 115 119 112 88 122 132 121 127 121 ...
 $ ILD_Persons       : int  6 4 6 7 7 7 3 7 7 7 ...
 $ Vasculitis_Persons: int  69 64 64 72 156 160 200 203 184 192 ...
 $ Vasculitis_Women  : int  7 8 2 5 2 6 9 9 7 4 ...
 $ Carditis_P        : int  16 18 23 22 22 32 25 33 26 28 ...
```

3 Producing a package skeleton

Now we produce a package skeleton.

```
> package.skeleton("WiscMSexam", c("bmd", "feltys", "raprev"))
> tree <- system("tree WiscMSexam", intern=TRUE)
> str(tree)

 chr [1:14] "WiscMSexam" "|-- data" "|   |-- bmd.rda" "|   |-- feltys.rda" ...

> cat(paste(tree, collapse="\n"), "\n")

WiscMSexam
|-- data
|   |-- bmd.rda
|   |-- feltys.rda
|   `-- raprev.rda
|-- DESCRIPTION
|-- man
|   |-- bmd.Rd
|   |-- feltys.Rd
|   |-- raprev.Rd
|   `-- WiscMSexam-package.Rd
`-- Read-and-delete-me
2 directories, 9 files
```