

Introduction to the `surveydata` package for working with survey data.

Andrie de Vries

October 22, 2013

`surveydata` is a package that makes it easy to work with typical survey data that originated in SPSS or other formats.

A `surveydata` object consists of:

- A data frame with a row for each respondent and a column for each question. Column names are typically names in the pattern `Q1`, `Q2_1`, `Q2_2`, `Q3` - where underscores separate the subquestions when these originated in a grid (array) of questions.
- Question metadata gets stored in the `variable.labels` attribute of the data frame. This typically contains the original questionnaire text for each question.
- Information about the subquestion separator (typically an underscore) is stored in the `patterns` attribute.

Data processing a survey file can be tricky, since the standard methods for dealing with data frames does not conserve the `variable.labels` attribute. The `surveydata` package defines a `surveydata` class and the following methods that knows how to deal with the `variable.labels` attribute:

- `as.surveydata`
- `[.surveydata`
- `[<-.surveydata`
- `$.surveydata`
- `$<-.surveydata`
- `merge.surveydata`

In addition, `surveydata` defines the following convenient methods for extracting and working with the variable labels:

- `varlabels`
- `varlabels<-`

1 Defining a surveydata object

First load the `surveydata` package.

```
> library(surveydata)
```

Next, create sample data. A data frame is the ideal data structure for survey data, and the convention is that data for each respondent is stored in the rows, while each column represents answers to a specific question.

```
> sdat <- data.frame(  
+   id     = 1:4,  
+   Q1     = c("Yes", "No", "Yes", "Yes"),  
+   Q4_1   = c(1, 2, 1, 2),  
+   Q4_2   = c(3, 4, 4, 3),  
+   Q4_3   = c(5, 5, 6, 6),  
+   Q10    = factor(c("Male", "Female", "Female", "Male")),  
+   crossbreak = c("A", "A", "B", "B"),  
+   weight  = c(0.9, 1.1, 0.8, 1.2)  
+ )  
>
```

The survey metadata consists of the questionnaire text. For example, this can be represented by a character vector, with an element for each question.

To assign this metadata to the survey data, use the `varlabels` function. This function assigns the questionnaire text to the `variable.labels` attribute of the data frame.

```
> varlabels(sdat) <- c(  
+   "RespID",  
+   "Question 1",  
+   "Question 4: red", "Question 4: green", "Question 4: blue",  
+   "Question 10",  
+   "crossbreak",  
+   "weight"  
+ )
```

Finally, create the `surveydata` object. To do this, call the `as.surveydata` function. The argument `renameVarlabels` controls whether the `varlabels` get renamed with the same names as the data. This is an essential step, and ensures that the question text remains in synch with the column names.

```
> sv <- as.surveydata(sdat, renameVarlabels=TRUE)
```

2 Extracting specific questions

It is easy to extract specific questions with the `[]` operator. This works very similar to extraction of data frames. However, there are two important differences:

- The extraction operators will always return a `surveydata` object, even if only a single column is returned. This is different from the behaviour of data frames, where a single column is simplified to a vector.

- Extracting a question with multiple subquestions, e.g. "Q4" returns multiple columns.

```
> sv[, "Q1"]
```

```
  Q1
1 Yes
2 No
3 Yes
4 Yes
```

```
> sv[, "Q4"]
```

```
  Q4_1 Q4_2 Q4_3
1     1     3     5
2     2     4     5
3     1     4     6
4     2     3     6
```

The extraction makes use of the underlying metadata, contained in the `varlabels` and `pattern` attributes:

```
> varlabels(sv)
```

id	Q1	Q4_1	Q4_2
"RespID"	"Question 1"	"Question 4: red"	"Question 4: green"
Q4_3	Q10	crossbreak	weight
"Question 4: blue"	"Question 10"	"crossbreak"	"weight"

```
> pattern(sv)
```

```
$sep
[1] "-"
```

```
$exclude
[1] "other"
```

3 Working with question columns

It is easy to query the `surveydata` object to find out which questions it contains, as well as which columns store the data for those questions.

```
> questions(sv)
```

```
[1] "id"      "Q1"      "Q4"      "Q10"     "crossbreak"
[6] "weight"
```

```
> which.q(sv, "Q1")
```

```
[1] 2
```

```
> which.q(sv, "Q4")
```

```
[1] 3 4 5
```

4 Reading the questionnaire text

The function `qText` gives access to the questionnaire text.

```
> qText(sv, "Q1")
[1] "Question 1"
> qText(sv, "Q4")
[1] "Question 4: red" "Question 4: green" "Question 4: blue"
> qTextCommon(sv, "Q4")
[1] "Question 4"
> qTextUnique(sv, "Q4")
[1] "red" "green" "blue"
```

5 Final thoughts

The last word.