How to use DAP .RData

2021-03-25

Contents

1 Introduction

This document introduces you to the structure of the DAP .RData format. The data were created using the `haven()` package in R in order to append (i) labels to the raw values of the variable and (ii) the question text to the variable. The labelled variables in the .Rdata may be a new and unfamiliar data format, so this document will provide some examples of how to use the labelled variable format, as well as code to transform the labelled variables into integer or factor variables. The code below is applicable to all DAP .RData files, but for simplicity here we illustrate examples with the HLES_dog_owner table.

Let's first load necessary libraries and the DAP_2020_HLES_dog_owner_v1.0.RData. **NOTE: In order to use the DAP data in R, you will need to install and load the `haven` package.**

```
library(haven)
library(ggplot2)
load("DAP_2020_HLES_dog_owner_v1.0.RData")
```

2 Using the labelled variables

Let's first take a look at the structure of the HLES_dog_owner data. It is a dataframe with 27,542 rows and 652 columns.

```
class(HLES_dog_owner)
```

```
## [1] "data.frame"
```

```
dim(HLES_dog_owner)
```

```
## [1] 27542   652
```

2.1 How to check the labels of a labelled integer?

Let's focus on one variable: *dd_breed_pure_or_mixed*, which is a `haven_labelled integer`.

```
class(HLES_dog_owner$dd_breed_pure_or_mixed)
```

```
## [1] "haven_labelled" "vctrs_vctr"     "integer"
```

To view the labels of a specific variable, use any of the following three functions: `head()`, `attributes()`, `haven::print_labels()`.

As shown below, *dd_breed_pure_or_mixed* is either 1 or 2 in the dataframe, which corresponds to the following labels Purebred and Mixed breed, respectively.

```
head(HLES_dog_owner$dd_breed_pure_or_mixed)
```

```
## <labelled<integer>[6]>: Is your dog a purebred or a mixed breed?
## [1] 2 2 1 2 1 1
##
## Labels:
##  value       label
##      1    Purebred
##      2 Mixed breed
```

```
attributes(HLES_dog_owner$dd_breed_pure_or_mixed)
```

```
## $labels
##    Purebred Mixed breed
##           1           2
##
## $label
## [1] "Is your dog a purebred or a mixed breed?"
##
## $class
## [1] "haven_labelled" "vctrs_vctr"     "integer"
```

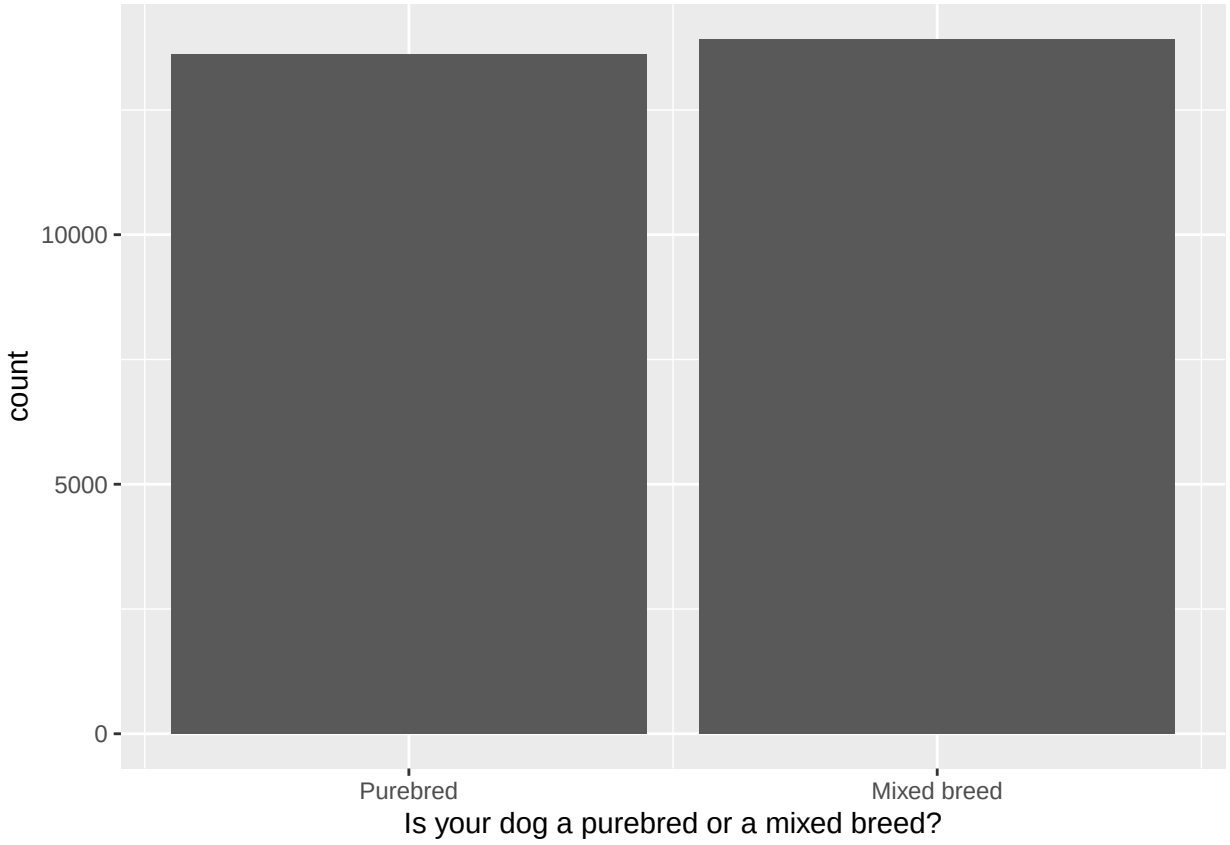
```
haven::print_labels(HLES_dog_owner$dd_breed_pure_or_mixed)
```

```
##
## Labels:
##  value       label
##      1    Purebred
##      2 Mixed breed
```

2.2 How to use labelled variables in exploratory analysis (such as ggplot)?

Below is how to use a labelled variable to produce a plot using ggplot().

```
ggplot(data=HLES_dog_owner,aes(x=dd_breed_pure_or_mixed)) +
  geom_bar() +
  scale_x_continuous(attributes(HLES_dog_owner$dd_breed_pure_or_mixed)$label,
                     breaks=attributes(HLES_dog_owner$dd_breed_pure_or_mixed)$labels,
                     labels=names(attributes(HLES_dog_owner$dd_breed_pure_or_mixed)$labels))
```



3 Transforming the labelled variables

Below is sample code for transforming the labelled variables into either integer values or factor variables.

3.1 Transforming to integer values

Let's use the example of *oc_household_adult_count* which is the number of adults in the household. As shown below the values for this labelled variable in the dataframe range from 1 to 10 and the labels go from 1 to 10+.

```
class(HLES_dog_owner$oc_household_adult_count)
```

```
## [1] "haven_labelled" "vctrs_vctr"     "integer"
```

```
head(HLES_dog_owner$oc_household_adult_count)
```

```
## <labelled<integer>[6]>: Number of adults (age 18 and over) in your household
## [1] 2 3 1 3 2 5
##
## Labels:
##  value label
##      1     1
##      2     2
##      3     3
##      4     4
##      5     5
##      6     6
##      7     7
##      8     8
##      9     9
##     10   10+
```

To remove the labels and transform the *oc_household_adult_count* variable into an integer variable, use the following function: `haven::zap_labels()`.

```
HLES_dog_owner$oc_household_adult_count <- zap_labels(HLES_dog_owner$oc_household_adult_count)
```

The *oc_household_adult_count* variable is now an integer variable.

```
class(HLES_dog_owner$oc_household_adult_count)
```

```
## [1] "integer"
```

```
head(HLES_dog_owner$oc_household_adult_count)
```

```
## [1] 2 3 1 3 2 5
```

3.2 Transforming to a factor variable

Let's use the example of *dd_sex* which is the sex of the dog. As shown below, the values for this labelled variable in the dataframe are 1 and 2 which correspond to Male and Female, respectively.

```
class(HLES_dog_owner$dd_sex)
```

```
## [1] "haven_labelled" "vctrs_vctr"     "integer"
```

```
head(HLES_dog_owner$dd_sex)
```

```
## <labelled<integer>[6]>: What is the sex of your dog?
## [1] 1 1 2 1 2 2
##
## Labels:
##  value  label
##      1   Male
##      2 Female
```

To change the *dd_sex* variable into a factor, use the following function: `haven::as_factor()`.

```
HLES_dog_owner$dd_sex <- as_factor(HLES_dog_owner$dd_sex)
```

The *dd_sex* variable is now a factor variable.

```
class(HLES_dog_owner$dd_sex)
```

```
## [1] "factor"
```

```
head(HLES_dog_owner$dd_sex)
```

```
## [1] Male   Male   Female Male   Female Female
## Levels: Male Female
```