

Stata and SPSS files keep their labels

haven, the labelled class, and why a survey firm's .dta should be read in R even when the analysis happens elsewhere — plus the two ways to lose a codebook without noticing.

Pierre Robentz Cassion

Lesson 3 of 8

R and the Tidyverse for Programme Data — Data that keeps its meaning

What this lesson covers

- What a CSV throws away
- The labelled class
- Converting deliberately
- The two ways to lose a codebook
- User-defined missing values
- SPSS and the other formats
- Where this fits with the Python course
- What comes next

What a CSV throws away — In R

```
library(haven)
```

```
survey <- read_dta(here("data", "raw", "household-survey.dta"))
```

The labelled class — In R

```
class(survey$consent)
#> [1] "haven_labelled" "vctrs_vctr" "double"

attr(survey$consent, "labels")
#>      Yes      No Don't know
#>      1      2      9

attr(survey$consent, "label")
#> [1] "Household head consented to interview"
```

The labelled class — In R

```
mean(survey$consent, na.rm = TRUE)  
#> [1] 1.7
```

Converting deliberately

- A categorical variable becomes a factor

Converting deliberately — In R

```
library(dplyr)

survey <- survey |>
  mutate(consent = as_factor(consent))

levels(survey$consent)
#> [1] "Yes" "No" "Don't know"
```

Converting deliberately

- A genuine number loses its labels

Converting deliberately — In R

```
survey <- survey |> mutate(hh_size = zap_labels(hh_size))
```

Converting deliberately

- Everything at once, when the file is uniformly categorical

Converting deliberately — In R

```
survey <- read_dta(path) |> as_factor()
```

The two ways to lose a codebook

- Reading with the labels already applied

The two ways to lose a codebook — In R

```
survey <- read_dta(path, .name_repair = "unique")  
survey <- as_factor(survey)           # values are gone; only labels remain
```

The two ways to lose a codebook — In R

```
survey <- read_dta(path)

codes <- survey |> mutate(across(where(is.labelled), zap_labels))
labels <- survey |> as_factor()
```

The two ways to lose a codebook

- Writing the CSV and moving on

The two ways to lose a codebook — In R

```
readr::write_csv(as_factor(survey), "survey.csv")
```

The two ways to lose a codebook — In R

```
codebook <- tibble::tibble(  
  variable = names(survey),  
  label    = vapply(survey, function(x) attr(x, "label") %||% NA_character_, character(1))  
)  
  
readr::write_csv(codebook, "survey-codebook.csv")
```

User-defined missing values — In R

```
survey$income  
#> <labelled<double>[3]>  
#> [1] 4500 NA(a) NA(b)
```

User-defined missing values — In R

```
survey <- survey |> mutate(income = zap_missing(income))
```

SPSS and the other formats — In R

```
read_sav(path)    # SPSS .sav  
read_por(path)    # SPSS portable  
read_dta(path)    # Stata  
read_xpt(path)    # SAS transport
```

```
write_dta(survey, "cleaned.dta")
```

Where this fits with the Python course

- The *Python for Programme Data* course says, in its lesson on codes and sentinels, that this is the one place where the answer is "use the other language".

What comes next

- The labels are attached and converted into factors.

Read the full lesson, with runnable code

`https:`

`//data-analysis.cassion.dev/courses/r-for-programme-data/r-03-labelled-data`