Using GenForward data in R

Kumar Ramanathan

2023-12-12

This document is a brief guide on how to read and wrangle GenForward data files in R.

GenForward data files are provided in the Stata file format, with file extension `.dta`. There are multiple packages available to read Stata files. We will use the `haven` package.

Before you use the package for the first time, you will need to install it. You can do so using the code `install.packages("haven")`.

Let's load the package.

```
library(haven)
```

The function to read Stata files in the package is `read_dta()`. Let's load an example file:

```
survey <- read_dta("genforward_2022-07_unembargoed.dta")
```

Let's take a look at the imported data frame:

```
survey
```

```
# A tibble: 4,201 x 204
   PANEL_TYPE     GENFRACE WEIGHT_TN Q1A     Q1       Q2       Q3A      Q3
   <dbl+lbl>      <dbl+lb>     <dbl> <dbl+l> <dbl+lb> <dbl+lb> <dbl+lb> <dbl+lb>
 1 21 [Non-proba~ 2 [Whit~     1.05  1 [App~  1 [Ver~  2 [Som~  1 [Ver~ 24 [Fre~
 2 21 [Non-proba~ 2 [Whit~     2.76  5 [Dis~  2 [Som~  4 [Ver~  1 [Ver~ 11 [Eco~
 3 21 [Non-proba~ 2 [Whit~     3.27  1 [App~  1 [Ver~  1 [Ver~  1 [Ver~ 26 [Too~
 4 21 [Non-proba~ 2 [Whit~     1.41  2 [App~  4 [Ver~  1 [Ver~  4 [Ver~  9 [Inc~
 5 21 [Non-proba~ 2 [Whit~     2.82  5 [Dis~  2 [Som~  4 [Ver~  1 [Ver~ 11 [Eco~
 6 21 [Non-proba~ 2 [Whit~     0.354 1 [App~  3 [Som~  1 [Ver~  2 [Som~  5 [Gun~
 7 21 [Non-proba~ 2 [Whit~     3.18  5 [Dis~  1 [Ver~  4 [Ver~  1 [Ver~ 21 [Ter~
 8 21 [Non-proba~ 2 [Whit~     2.98  5 [Dis~ 98 [SKI~ 98 [SKI~  3 [Som~ 18 [Cri~
```

```
 9 21 [Non-proba~ 2 [Whit~     0.946 1 [App~  1 [Ver~  1 [Ver~  1 [Ver~  5 [Gun~
10 21 [Non-proba~ 2 [Whit~     0.821 1 [App~  4 [Ver~  1 [Ver~ 77 [Don~  1 [The~
# i 4,191 more rows
# i 196 more variables: Q4 <dbl+lbl>, Q5 <dbl+lbl>, Q6 <dbl+lbl>, Q7 <dbl+lbl>,
#   Q8 <dbl+lbl>, Q9 <dbl+lbl>, Q12 <dbl+lbl>, Q13 <dbl+lbl>, Q14 <dbl+lbl>,
#   Q15 <dbl+lbl>, Q16 <dbl+lbl>, Q17 <dbl+lbl>, Q18 <dbl+lbl>, Q19 <dbl+lbl>,
#   Q20 <dbl+lbl>, Q21 <dbl+lbl>, Q22 <dbl+lbl>, Q23 <dbl+lbl>, Q24 <dbl+lbl>,
#   Q25 <dbl+lbl>, Q25_OE <chr>, Q26 <dbl+lbl>, Q27 <dbl+lbl>, Q28_1 <dbl+lbl>,
#   Q28_2 <dbl+lbl>, Q28_3 <dbl+lbl>, Q28_4 <dbl+lbl>, Q28_5 <dbl+lbl>, ...
```

You'll notice that `read_dta()` imports non-numeric variables in a special class called labelled variables. This class stores the variable and value labels (the `lbl` part of `dbl+lbl`). To learn more about this format and some of its uses, read the documentation for `read_dta()` and `labelled()`.

To convert labelled variables into the more familiar factor variables, we can use the `as_factor()` function. You can use this function on a single variable or the whole data frame.

```
as_factor(survey)
```

```
# A tibble: 4,201 x 204
   PANEL_TYPE GENFRACE WEIGHT_TN Q1A   Q1    Q2    Q3A   Q3    Q4    Q5    Q6
   <fct>      <fct>        <dbl> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <fct>
 1 Non-proba~ White, ~     1.05  Appr~ Very~ Some~ "Ver~ Free~ Off ~ Very~ Neit~
 2 Non-proba~ White, ~     2.76  Disa~ Some~ Very~ "Ver~ Econ~ Off ~ Very~ Some~
 3 Non-proba~ White, ~     3.27  Appr~ Very~ Very~ "Ver~ Too ~ Off ~ Very~ Very~
 4 Non-proba~ White, ~     1.41  Appr~ Very~ Very~ "Ver~ Inco~ Off ~ Neit~ Some~
 5 Non-proba~ White, ~     2.82  Disa~ Some~ Very~ "Ver~ Econ~ Off ~ Very~ Very~
 6 Non-proba~ White, ~     0.354 Appr~ Some~ Very~ "Som~ Gun ~ Gene~ Some~ Very~
 7 Non-proba~ White, ~     3.18  Disa~ Very~ Very~ "Ver~ Terr~ Off ~ Some~ Neit~
 8 Non-proba~ White, ~     2.98  Disa~ SKIP~ SKIP~ "Som~ Crime Off ~ Neit~ Neit~
 9 Non-proba~ White, ~     0.946 Appr~ Very~ Very~ "Ver~ Gun ~ Off ~ Very~ Very~
10 Non-proba~ White, ~     0.821 Appr~ Very~ Very~ "Don~ The ~ Gene~ Very~ Very~
# i 4,191 more rows
# i 193 more variables: Q7 <fct>, Q8 <fct>, Q9 <fct>, Q12 <fct>, Q13 <fct>,
#   Q14 <fct>, Q15 <fct>, Q16 <fct>, Q17 <fct>, Q18 <fct>, Q19 <fct>,
#   Q20 <fct>, Q21 <fct>, Q22 <fct>, Q23 <fct>, Q24 <fct>, Q25 <fct>,
#   Q25_OE <chr>, Q26 <fct>, Q27 <fct>, Q28_1 <fct>, Q28_2 <fct>, Q28_3 <fct>,
#   Q28_4 <fct>, Q28_5 <fct>, Q28_DK <fct>, Q28_SKP <fct>, Q28_REF <fct>,
#   Q29 <fct>, RND_01 <fct>, INS1_Q30 <fct>, INS2_Q30 <fct>, Q30 <fct>, ...
```

Alternatively, you can convert all these labelled variables to numeric format using the `zap_labels()` function.

```
zap_labels(survey)
```

```
# A tibble: 4,201 x 204
   PANEL_TYPE GENFRACE WEIGHT_TN   Q1A    Q1    Q2   Q3A    Q3    Q4    Q5    Q6
        <dbl>    <dbl>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1         21        2     1.05      1     1     2     1    24     1     1     3
 2         21        2     2.76      5     2     4     1    11     1     5     4
 3         21        2     3.27      1     1     1     1    26     1     1     1
 4         21        2     1.41      2     4     1     4     9     1     3     4
 5         21        2     2.82      5     2     4     1    11     1     5     5
 6         21        2     0.354     1     3     1     2     5     2     2     1
 7         21        2     3.18      5     1     4     1    21     1     4     3
 8         21        2     2.98      5    98    98     3    18     1     3     3
 9         21        2     0.946     1     1     1     1     5     1     1     1
10         21        2     0.821     1     4     1    77     1     2     5     1
# i 4,191 more rows
# i 193 more variables: Q7 <dbl>, Q8 <dbl>, Q9 <dbl>, Q12 <dbl>, Q13 <dbl>,
#   Q14 <dbl>, Q15 <dbl>, Q16 <dbl>, Q17 <dbl>, Q18 <dbl>, Q19 <dbl>,
#   Q20 <dbl>, Q21 <dbl>, Q22 <dbl>, Q23 <dbl>, Q24 <dbl>, Q25 <dbl>,
#   Q25_OE <chr>, Q26 <dbl>, Q27 <dbl>, Q28_1 <dbl>, Q28_2 <dbl>, Q28_3 <dbl>,
#   Q28_4 <dbl>, Q28_5 <dbl>, Q28_DK <dbl>, Q28_SKP <dbl>, Q28_REF <dbl>,
#   Q29 <dbl>, RND_01 <dbl>, INS1_Q30 <dbl>, INS2_Q30 <dbl>, Q30 <dbl>, ...
```

For full variable and value labels, make sure to consult the codebook file provided alongside each GenForward dataset.