fars: A package to manipulate data of Administration's Fatality Analysis Reporting System

Carlos J. Dommar

2018-02-09

The functions provided in the package *fars* use data (provided with the package) from the US National Highway Traffic Safety Administration's Fatality Analysis Reporting System, which is a nationwide census providing the American public yearly data regarding fatal injuries suffered in motor vehicle traffic crashes. The current five functions of the package facilitate the manipulation, exploration and plotting of the data.

The functions contained in this packages are the following:

- *make_filename()*
- *fars_read()*
- *fars_read_years()*
- *fars_summarize_years()*
- *fars_map_state()*

make_filename()

The *make_filename()* function takes a four digit number representing a year and produces a string of the format *"accident_yyyy.csv.bz2"* where *yyyy* is the year. This string is used in other **fars** functions

Examples:

```
library(fars)
```

```
## Warning: replacing previous import 'magrittr::extract' by 'tidyr::extract'
## when loading 'fars'
```

```
make_filename(2015)
```

```
## [1] "accident_2015.csv.bz2"
```

```
make_filename(2001)
```

```
## [1] "accident_2001.csv.bz2"
```

fars_read()

The function *fars_read()* reads and loads a file of the format *"accident_yyyy.csv.bz2"* into a dplyr's data frame tbl, where *yyyy* is the year in four digits format. The location where the file is located must be provided as a path. This function returns a error message either if the file's name #' does not exits or a data frame table representing the data of the file. Example:

```
library("fars")
fars_read("~/accident_2015.csv.bz2")
```

```
## # A tibble: 32,166 x 52
##    STATE ST_CASE VE_TOTAL VE_FORMS PVH_INVL  PEDS PERNOTMVIT PERMVIT
##    <int>   <int>    <int>    <int>    <int> <int>      <int>   <int>
## 1      1   10001        1        1        0     0          0       1
## 2      1   10002        1        1        0     0          0       1
```

1

```
## 3     1   10003       1       1       0     0         0         2
## 4     1   10004       1       1       0     0         0         1
## 5     1   10005       2       2       0     0         0         2
## 6     1   10006       1       1       0     0         0         2
## 7     1   10007       1       1       0     0         0         2
## 8     1   10008       1       1       0     1         1         1
## 9     1   10009       1       1       0     0         0         1
## 10    1   10010       2       2       0     0         0         2
## # ... with 32,156 more rows, and 44 more variables: PERSONS <int>,
## #   COUNTY <int>, CITY <int>, DAY <int>, MONTH <int>, YEAR <int>,
## #   DAY_WEEK <int>, HOUR <int>, MINUTE <int>, NHS <int>, RUR_URB <int>,
## #   FUNC_SYS <int>, RD_OWNER <int>, ROUTE <int>, TWAY_ID <chr>,
## #   TWAY_ID2 <chr>, MILEPT <int>, LATITUDE <dbl>, LONGITUD <dbl>,
## #   SP_JUR <int>, HARM_EV <int>, MAN_COLL <int>, RELJCT1 <int>,
## #   RELJCT2 <int>, TYP_INT <int>, WRK_ZONE <int>, REL_ROAD <int>,
## #   LGT_COND <int>, WEATHER1 <int>, WEATHER2 <int>, WEATHER <int>,
## #   SCH_BUS <int>, RAIL <chr>, NOT_HOUR <int>, NOT_MIN <int>,
## #   ARR_HOUR <int>, ARR_MIN <int>, HOSP_HR <int>, HOSP_MN <int>,
## #   CF1 <int>, CF2 <int>, CF3 <int>, FATALS <int>, DRUNK_DR <int>
```

fars_read_years()

This function reads year(s) and returns a list with the corresponding months and year(s) out of the FARS data base. The function reads a vector or list 'years' with four digit number in the format yyyy (e.g., 2015, (2013:2015), c(2013, 2015)) and if the years exist/are correct the function wil returns a list with the correspondung month and year found in the data (magrittr needed for the operator %>%).

Example:

```r
library(magrittr)
fars_read_years(years = c(2013, 2015))
```

```
## [[1]]
## # A tibble: 30,202 x 2
##    MONTH  year
##    <int> <dbl>
## 1      1  2013
## 2      1  2013
## 3      1  2013
## 4      1  2013
## 5      1  2013
## 6      1  2013
## 7      1  2013
## 8      1  2013
## 9      1  2013
## 10     1  2013
## # ... with 30,192 more rows
##
## [[2]]
## # A tibble: 32,166 x 2
##    MONTH  year
##    <int> <dbl>
## 1      1  2015
## 2      1  2015
```

```
##  3      1  2015
##  4      1  2015
##  5      1  2015
##  6      1  2015
##  7      1  2015
##  8      1  2015
##  9      1  2015
## 10      1  2015
## # ... with 32,156 more rows
```

fars_summarize_years()

This function takes years in format yyyy and returns the number of observations per month per year

Examples:

```
fars_summarize_years(years = c(2013, 2015))
```

```
## # A tibble: 12 x 3
##     MONTH `2013` `2015`
##     <int>  <int>  <int>
## 1       1   2230   2368
## 2       2   1952   1968
## 3       3   2356   2385
## 4       4   2300   2430
## 5       5   2532   2847
## 6       6   2692   2765
## 7       7   2660   2998
## 8       8   2899   3016
## 9       9   2741   2865
## 10     10   2768   3019
## 11     11   2615   2724
## 12     12   2457   2781
```

```
fars_summarize_years(years = 2013:2015)
```

```
## # A tibble: 12 x 4
##     MONTH `2013` `2014` `2015`
##     <int>  <int>  <int>  <int>
## 1       1   2230   2168   2368
## 2       2   1952   1893   1968
## 3       3   2356   2245   2385
## 4       4   2300   2308   2430
## 5       5   2532   2596   2847
## 6       6   2692   2583   2765
## 7       7   2660   2696   2998
## 8       8   2899   2800   3016
## 9       9   2741   2618   2865
## 10     10   2768   2831   3019
## 11     11   2615   2714   2724
## 12     12   2457   2604   2781
```

```
fars_summarize_years(years = list(2013,2015))
```

```
## # A tibble: 12 x 3
##     MONTH `2013` `2015`
```

```
##     <int>  <int>  <int>
## 1      1   2230   2368
## 2      2   1952   1968
## 3      3   2356   2385
## 4      4   2300   2430
## 5      5   2532   2847
## 6      6   2692   2765
## 7      7   2660   2998
## 8      8   2899   3016
## 9      9   2741   2865
## 10    10   2768   3019
## 11    11   2615   2724
## 12    12   2457   2781
```

fars_map_state()

Plot a map with geolocated observations (cars accidents). This functions takes a valid state number and a valid year and produces a map with geolocated observations (accidents) plotted as dots on the map.

Example:

```
fars_map_state(51, 2015)
```