

Package: crashapi (via r-universe)

October 12, 2026

Type Package

Title CrashAPI

Version 0.1.2

Description Get Fatality Analysis Reporting System (FARS) data with the FARS API from the U.S. National Highway Traffic Safety Administration (NHTSA).

License MIT + file LICENSE

URL <https://github.com/elipousson/crashapi>,
<https://elipousson.github.io/crashapi/>

BugReports <https://github.com/elipousson/crashapi/issues>

Depends R (>= 4.1.0)

Imports cli, dplyr (>= 1.1.0), glue, httr2 (>= 1.0.0), nanoparquet, rlang, stats, stringr, utils, vctrs

Suggests covr, haven, httptest2, knitr, readr, rmarkdown, sf, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libicu-dev libssl-dev

Repository <https://elipousson.r-universe.dev>

Date/Publication 2026-10-12 00:27:19 UTC

RemoteUrl <https://github.com/elipousson/crashapi>

RemoteRef HEAD

RemoteSha 16d0d4dd27a4b536ba3a0a526a3e05730cc1c15e

Contents

crash_data_index	2
decode_fars	3
fars_terms	5
fars_value_labels	5
fars_variables	7
fars_vars	8
fars_vars_labels	9
format_crashes	10
get_crss_zip	10
get_fars	12
get_fars_crash_persons	14
get_fars_crash_vehicles	15
get_fars_crashes	16
get_fars_zip	19
mmucc_codes	21
read_crashapi	22
Index	23

crash_data_index	<i>U.S. vehicular crash data index (city, county, regional, and state)</i>
------------------	--

Description

This index include identified data from cities, counties, or regional entities in 43 of 50 U.S. states. Statewide data sources are included from 33 states. In 4 states (NE, OH, ME, and CT), the only identified statewide data sources allow limited access through a web-based public query form. In 1 state (MN), data only available through restricted access mapping/query tool. Not all statewide data sources include all crashes (some include only cyclist/pedestrian crashes or fatal crashes) and the structure and format of the crash data provided varies considerably.

Usage

crash_data_index

Format

- A data frame with 75 rows and 22 variables:
- name Name of data set from provider.
 - level Geographic scope/level (e.g. city, county, region, state, national)
 - city City name
 - county County name
 - region logical COLUMN_DESCRIPTION
 - state_name U.S. state name

state_abb U.S. state abbreviation
 info_url Informational URL (e.g. informational page about file download options)
 data_url Data URL (e.g. direct link to ArcGIS FeatureServer layer)
 format Data format (e.g. Socrata, CKAN, ArcGIS MapServer, etc.)
 statewide_yn Yes for data with statewide geographic scope; NA for data from city, county, or regional level providers
 batch_download_yn Yes for data where batch download is possible
 start_year Earliest year for crashes in dataset
 end_year Latest year for crashes in dataset
 publisher Agency/organization responsible for publishing the data online
 description Description of the dataset from provider
 bike_ped_only Yes for data that only includes bike/ped involved crashes (common for Vision Zero programs)
 rolling_window Description of rolling time window if data is only available within a rolling window
 fatal_severe_only Yes for data that only includes fatal/severe crashes (common for Vision Zero programs)
 date_note Note on the dates for the crash data
 updates Information on update schedule if available
 note General notes

Details

This index was compiled by Eli Pousson between October 2021 and February 2022 with additional contributions from Mae Hanzlik.

Added: March 27 2022 Updated: June 05 2022

Corrections, updates, or additional sources should be added to this public Google Sheet: <https://docs.google.com/spreadsheets/d/1rmn6GbHNkfWLLDEEmA87iuy2yHdh7hBybCTZiQJEY0k/edit?usp=sharing>

decode_fars

Add decoded value columns to FARS or CRSS data

Description

decode_fars() adds a column with decoded values (e.g. INJ_SEVNAME for INJ_SEV) after each coded variable in a FARS or CRSS data table, using the value labels from [fars_value_labels\(\)](#) for the year of each row. Variables that already have a decoded column (like the . . . NAME columns in data from the CrashAPI or FARS zipped CSV files from 2015 onwards) are not changed unless overwrite = TRUE.

Usage

```
decode_fars(
  x,
  table,
  year = NULL,
  decode = c("names", "replace", "labelled"),
  source = c("FARS", "CRSS"),
  suffix = "NAME",
  overwrite = FALSE
)
```

Arguments

x	A data frame with a FARS or CRSS data table, e.g. from get_fars() , get_fars_zip() , or get_crss_zip() .
table	Name of the FARS or CRSS data table (e.g. "accident", "person", or "vehicle").
year	Year of the data. If NULL (default), uses the CASEYEAR, CaseYear, or YEAR column of x. Can be a single year or one year for each row of x.
decode	How to decode values: • "names" (default) adds a column with decoded values after each coded variable (e.g. INJ_SEVNAME for INJ_SEV), matching the . . . NAME columns in data from the CrashAPI and FARS zipped CSV files from 2015 onwards. • "replace" replaces the codes with a factor of decoded values (with levels in code order) and removes any existing . . . NAME column. Variables with values that don't have value labels (e.g. AGE, where only special codes like "Not Reported" have labels) are not replaced. • "labelled" keeps the codes and adds value labels with haven::labelled(). A labelled vector has a single set of value labels so the value labels from the latest year in the data are used (with a warning if the value labels differ between years). Requires the haven package.
source	Data source: "FARS" (default) or "CRSS".
suffix	Suffix for the names of the decoded columns if decode = "names". Default: "NAME".
overwrite	If TRUE and decode = "names", replace existing decoded columns. Default: FALSE.

Details

[get_fars\(\)](#) uses `decode_fars()` to add decoded columns to data from zipped FARS data files before 2015.

Value

x with additional columns of decoded values (decode = "names"), factors of decoded values (decode = "replace"), or labelled vectors (decode = "labelled").

Examples

```
person <- data.frame(CASEYEAR = c(1985, 2024), INJ_SEV = c(3, 3))
decode_fars(person, table = "person")

decode_fars(person, table = "person", decode = "replace")
```

fars_terms	<i>NHSTA Terms and Definitions</i>
------------	------------------------------------

Description

FARS-related terms defined by the National Highway Traffic Safety Administration based on ANSI D16.1-1996: Manual on Classification of Motor Vehicle Traffic Accidents.

Usage

```
fars_terms
```

Format

A data frame with 66 rows and 2 variables:

```
term character Term
```

```
definition character Term definition
```

Details

Added: October 25 2021 Updated: October 25 2021

Source

[NHTSA FARS Terms](#)

fars_value_labels	<i>Get value labels for FARS and CRSS variables</i>
-------------------	---

Description

fars_value_labels() returns the value labels (the decoded value for each code) for variables in the FARS or CRSS data files by year. The value labels are built from the SAS format programs and data files that NHTSA publishes with the zipped SAS data for each year (see data-raw/fars_value_labels.R).

Usage

```
fars_value_labels(
  year = NULL,
  table = NULL,
  variable = NULL,
  source = c("FARS", "CRSS")
)
```

Arguments

<code>year</code>	One or more years. If NULL (default), returns value labels for all years.
<code>table</code>	One or more FARS or CRSS data table names (e.g. "accident", "person", or "vehicle"). Not case sensitive. If NULL (default), returns value labels for all tables.
<code>variable</code>	One or more variable names (e.g. "INJ_SEV"). Not case sensitive. If NULL (default), returns value labels for all variables.
<code>source</code>	Data source: "FARS" (default) or "CRSS".

Details

Value labels change over time, so the same code can have a different label in different years. For example, the INJ_SEV code 3 is labelled "Incapacitating" in early FARS data and "Suspected Serious Injury (A)" in recent years.

The CrashAPI and FARS zipped CSV files from 2015 onwards include . . . NAME columns with decoded values. Use [decode_fars\(\)](#) to add these columns to older FARS data or CRSS data.

Labels from the SAS formats can differ slightly in wording from the . . . NAME columns in the CSV files (e.g. "Curve - Right" instead of "Curve Right" for VALIGN). A few variables use SAS formats that NHTSA doesn't publish (e.g. a state variable in some 2006-2015 tables) and are not included. Value labels for the FARS vPIC variables (e.g. VPICMAKE and VPICMODEL, used from 2020 onwards) are not included since FARS data already includes . . . NAME columns for these variables; they are included for CRSS data.

Value

A data frame with columns `source`, `year`, `table`, `variable`, `format` (the SAS format name, without the year), `code` (as written in the SAS format, e.g. "01"), `code_min` and `code_max` (numeric codes or ranges), and `value` (the label).

Examples

```
# Injury severity labels for the person table in 1985 and 2024
fars_value_labels(
  year = c(1985, 2024),
  table = "person",
  variable = "INJ_SEV"
)
```

fars_variables	<i>Get variable labels for FARS and CRSS variables</i>
----------------	--

Description

fars_variables() returns the variables in each FARS or CRSS data table by year with the variable label and SAS format name from the SAS data files that NHTSA publishes with the zipped SAS data for each year (see data-raw/fars_value_labels.R). Variable labels can change over time and include variables that are not in [fars_vars_labels](#) (e.g. variables added in recent years or only used in CRSS).

Usage

```
fars_variables(  
  year = NULL,  
  table = NULL,  
  variable = NULL,  
  source = c("FARS", "CRSS")  
)
```

Arguments

year	One or more years. If NULL (default), returns value labels for all years.
table	One or more FARS or CRSS data table names (e.g. "accident", "person", or "vehicle"). Not case sensitive. If NULL (default), returns value labels for all tables.
variable	One or more variable names (e.g. "INJ_SEV"). Not case sensitive. If NULL (default), returns value labels for all variables.
source	Data source: "FARS" (default) or "CRSS".

Value

A data frame with columns source, year, table, variable, label (the variable label), and format (the SAS format name without the year, or NA for variables without value labels).

Examples

```
fars_variables(year = 2024, table = "person", variable = c("AGE", "INJ_SEV"))
```

fars_vars	<i>Get variables and variable attributes for the Fatality Analysis Reporting System (FARS) API</i>
-----------	--

Description

By default, this function returns the list of variables for the data year specified. If var is "make", "model", or "bodytype", the function returns a list of variable attributes for the specified variable name or attributes for make model and body type specified in the FARS dataset.

Usage

```
fars_vars(  
  year = getOption("crashapi.year", 2024),  
  var = NULL,  
  make = NULL,  
  model = NULL  
)
```

Arguments

year	One or more case years between 2010 and 2024. If more than one year is provided, the results for each year are combined. Ignored if var is "bodytype" since body type attributes are not available by year. Defaults to getOption("crashapi.year", 2024).
var	Default NULL. Supported values are "make", "model", and "bodytype". Using the var parameter returns variable attributes for the specified variable name or attributes for make model and body type specified in the dataset.
make	Integer. Make ID number. Required to return variables for "model" and "bodytype". Get a list of make ID numbers using the "make" var for the selected year, e.g. <code>fars_vars(year = 2010, var = "make")</code> .
model	Integer. Model ID number. Required to return variables for "bodytype". Get a list of model ID numbers using the "model" var for the selected year with a valid make ID number, e.g. <code>fars_vars(year = 2010, var = "model", make = 37)</code>

Value

A data frame with a year column (except when var is "bodytype").

Examples

```
head(fars_vars(year = 2022, var = "make"), 5)  
  
head(fars_vars(year = 2022, var = "model", make = 12), 5)  
  
fars_vars(year = 2022, var = "bodytype", make = 12, model = 37)
```

fars_vars_labels	<i>FARS variable names and labels</i>
------------------	---------------------------------------

Description

A table of FARS table variable names extracted from the Fatality Analysis Reporting System (FARS) Analytical User's Manual, 1975-2019, documentation of the SAS format data files.

Usage

```
fars_vars_labels
```

Format

A data frame with 749 rows and 15 variables:

```
name character Variable name
label character Variable label
order double Sort order
data_file character SAS data file name
data_file_id double SAS data file ID
file_id character File ID
key logical Indicator for key variables
location double Location in SAS data file
mmuc_equivalent logical Equivalent term in MMUC (placeholder)
discontinued logical Indicator for discontinued variables
api_only logical Indicator for variables only used by API
api character Name(s) of corresponding CrashAPI service (e.g. "Details" for GetCaseDetails or
"FARS data" for GetFARSData)
name_var logical Indicator for "NAME" variable returned by API
nm Short version of the variable name
api_list_col logical Indicator for list columns returned by API
```

Details

This table is used for column order, CrashAPI metadata, and variable labels for columns that are only returned by the CrashAPI. Use [fars_variables\(\)](#) for complete, year-specific variable labels from the SAS data files that NHTSA publishes for each year.

The source data is installed with the package as a CSV file (`system.file("extdata", "fars_vars_labels.csv", package = "crashapi")`) with column definitions in `fars_vars_labels_definitions.csv`.

Added: January 31 2022 Updated: March 27 2022

format_crashes	<i>Format crash data</i>
----------------	--------------------------

Description

Reorder columns to match the order documented in Fatality Analysis Reporting System (FARS) Analytical User's Manual, 1975-2019, convert coordinate and count columns to numeric and integer values, and append derived columns for date, time, and datetime. Column names are kept as returned by the API.

Usage

```
format_crashes(x, details = TRUE, var_labels = c("attribute", "none"))
```

Arguments

x	Data frame with crash data.
details	If TRUE, append date, time, datetime columns to formatted crash data; defaults to TRUE
var_labels	How to add variable labels from fars_vars_labels : "attribute" (default) adds a "label" attribute to each column, which is shown by View() in RStudio and Positron and used by packages such as gt and gtsummary, and "none" doesn't add variable labels.

get_crss_zip	<i>Download CRSS data files as zipped CSV or SAS files</i>
--------------	--

Description

The **Crash Report Sampling System (CRSS)** is a national sample of police-reported motor vehicle crashes of all severities, from minor property damage to fatal crashes, and replaced the General Estimates System (GES) in 2016. This function is similar to [get_fars_zip\(\)](#) to download CRSS files directly from the NHTSA downloads site. If read is TRUE, the function returns a list containing data frames for each table available in the selected year.

Usage

```
get_crss_zip(
  year = getOption("crashapi.year", 2024),
  format = "csv",
  path = NULL,
  aux = FALSE,
  read = TRUE,
  geometry = FALSE,
  overwrite = FALSE,
```

```

    keep_files = FALSE,
    var_labels = c("attribute", "none")
  )

```

Arguments

year	One or more years of data from 2016 to 2024. Each year is downloaded separately and, if read = TRUE, returned as a list named by year. Defaults to <code>getOption("crashapi.year", 2024)</code> .
format	Format of zipped data tables ('csv' or 'sas'). Default: 'csv'. read = TRUE and geometry = TRUE are only supported if format is "csv".
path	Directory for downloaded and cached files. If NULL (default), uses <code>getOption("crashapi.cache_dir")</code> or the user cache directory from <code>tools::R_user_dir("crashapi", "cache")</code> .
aux	If TRUE, download auxiliary CRSS datasets.
read	If TRUE (default), read the cached parquet files into a list of tables with each list item corresponding to one data file.
geometry	Not supported for CRSS data, which does not include coordinates. If TRUE, a warning is shown and the accident table is returned as a data frame.
overwrite	If TRUE, download the data and replace any cached files. Default: FALSE
keep_files	If TRUE, keep the downloaded zip file and the extracted CSV files in the cache folder. Default: FALSE
var_labels	How to add variable labels: "attribute" (default) adds a "label" attribute to each column with a matching variable in fars_vars_labels and "none" doesn't add variable labels. Labels are taken from the FARS Analytical User's Manual so variables only used in CRSS are not labelled.

Details

CRSS is a sample, so national estimates should use the sampling weight (WEIGHT) and survey design variables (e.g. PSU_VAR and PSUSTRAT) as described in the CRSS Analytical User's Manual. CRSS data does not include crash locations beyond broad geographic categories (e.g. REGION and URBANICITY).

Zipped CSV files are converted to parquet files and cached in a folder named after the zip file within path (by default, the user cache directory from `tools::R_user_dir()`), and the zip and CSV files are removed unless `keep_files = TRUE`. Later calls read the cached parquet files without downloading the data again unless `overwrite = TRUE`.

Value

If read = TRUE, a list of data frames with one item for each table (with the accident table converted to an sf object if geometry = TRUE). If read = FALSE, the cached parquet file paths named by table (or the zip file path if format = "sas") invisibly. If more than one year is provided, returns a list named by year.

get_fars

*Get Fatality Analysis Reporting System (FARS) data tables***Description**

get_fars() returns a FARS data table (e.g. accident, person, or vehicle) for one or more states, an optional county, and one or more years, using the CrashAPI or the zipped data files from the NHTSA downloads site:

Usage

```
get_fars(
  state = NULL,
  year = getOption("crashapi.year", 2024),
  type = "accident",
  county = NULL,
  source = c("auto", "api", "zip"),
  ...,
  geometry = FALSE,
  crs = NULL,
  decode = c("names", "replace", "labelled", "none"),
  var_labels = c("attribute", "none")
)
```

Arguments

state	One or more state names, abbreviations, or FIPS numbers. If NULL (default), returns national data. Use "PR" for Puerto Rico.
year	One or more years from 1975 to 2024 (or from 2010 for the CrashAPI). The minimum and maximum values are used as the start and end year, with a message if year has gaps. Defaults to getOption("crashapi.year", 2024).
type	Name of the FARS data table, e.g. "accident" (default), "person", or "vehicle". Lowercase or uppercase values are permitted. The available tables vary by year and source; see get_fars_year() for tables available from the CrashAPI.
county	County name or FIPS number. Requires a single state.
source	Source for the data: "auto" (default), "api" for the CrashAPI, or "zip" for zipped data files.
...	Additional parameters passed to get_fars_zip() when data comes from zipped data files, e.g. path, overwrite, or keep_files.
geometry	If TRUE, return an sf object. Only supported for tables with LATITUDE and LONGITUD columns (e.g. accident).
crs	Coordinate reference system to use if geometry is TRUE.

decode	How to decode coded values using <code>decode_fars()</code> : "names" (default) adds . . . NAME columns with decoded values where they are missing (data from the CrashAPI and FARS zipped CSV files from 2015 onwards already include them), "replace" replaces codes with factors of decoded values, "labelled" adds value labels with <code>haven::labelled()</code> , and "none" doesn't decode values. See <code>decode_fars()</code> for details.
var_labels	How to add variable labels from <code>fars_vars_labels</code> : "attribute" (default) adds a "label" attribute to each column, which is shown by <code>View()</code> in RStudio and Positron and used by packages such as <code>gt</code> and <code>gtsummary</code> , and "none" doesn't add variable labels.

Details

- With `source = "auto"` (default), data for a state from 2010 onwards comes from the CrashAPI with `get_fars_year()`. National data (`state = NULL`), data for Puerto Rico (`state = "PR"`), and data before 2010 comes from zipped data files with `get_fars_zip()`, which are converted to parquet files and cached so later calls do not download the data again.
- Data from zipped files is filtered to the requested states and, if needed, a `CASEYEAR` column is added so the data matches the CrashAPI.
- If `county` is provided, tables with a `COUNTY` column (accident and person) are filtered by county and other tables are filtered to crashes in the county from the accident table (matching `ST_CASE` and `CASEYEAR`).

Use `get_fars_crashes()`, `get_fars_cases()`, `get_fars_crash_list()`, or `get_fars_summary()` to access other CrashAPI endpoints.

Value

A data frame, or an `sf` object if `geometry = TRUE`. If tables for multiple years from zipped data files can't be combined, a list of data frames named by year.

Examples

```
# Person table for Maryland from the CrashAPI
head(get_fars(state = "MD", year = 2022, type = "person"), 5)

# Accident table for Baltimore City
get_fars(state = "MD", county = "Baltimore city", year = 2022)

# National accident table from the zipped data files
get_fars(year = 2024)
```

get_fars_crash_persons

Get Crashes By Occupant

Description

This function returns a list of people involved in fatal crashes that match a specific age, sex, seat position, and injury severity using the CrashAPI GetCrashesByPerson endpoint. The returned data has one row for each person.

Usage

```
get_fars_crash_persons(
  state,
  age,
  sex,
  seat,
  injury,
  year = getOption("crashapi.year", 2024),
  var_labels = c("attribute", "none")
)
```

Arguments

state	Required. State name, abbreviation, or FIPS number.
age	Required. Age in years.
sex	Required. Code or name for the person's sex: 1 ("Male"), 2 ("Female"), 8 ("Not Reported"), or 9 ("Reported as Unknown"). "m", "f", and "unknown" are also accepted.
seat	Required. Code or name for the person's seat position, e.g. 11 or "Front Seat, Left Side" ("front left" is also accepted). Use 0 or "Not a Motor Vehicle Occupant" for non-occupants. See the FARS Analytical User's Manual for all seat position codes.
injury	Required. Code or name for the person's injury severity: 0 ("No Apparent Injury (O)"), 1 ("Possible Injury (C)"), 2 ("Suspected Minor Injury (B)"), 3 ("Suspected Serious Injury (A)"), 4 ("Fatal Injury (K)"), 5 ("Injured, Severity Unknown"), 6 ("Died Prior to Crash"), or 9 ("Unknown/Not Reported"). Short names (e.g. "fatal") and KABCO letters (e.g. "K") are also accepted.
year	numeric vector. One or more years from 2010 to 2024. The minimum and maximum values are used as the start and end year. Defaults to getOption("crashapi.year", 2024).
var_labels	How to add variable labels from fars_vars_labels : "attribute" (default) adds a "label" attribute to each column, which is shown by View() in RStudio and Positron and used by packages such as gt and gtsummary, and "none" doesn't add variable labels.

API behavior

NHTSA's documentation says this endpoint accepts "one or more" occupant details for age, sex, seat position, and injury severity. However, testing in October 2026 found that the API does not behave this way: the API returns an error (HTTP 404) unless all four are provided, an error message if any are blank, and only returns people who match all four values exactly. For this reason, age, sex, seat, and injury are all required.

The endpoint's `includeOccupants` and `includeNonOccupants` parameters are not supported since the required seat position already determines whether occupants or non-occupants are returned (use `seat = 0` for people who were not motor vehicle occupants).

To find people using other criteria (e.g. a range of ages or any injury severity), download the person table with `get_fars_year()` using `type = "person"` and filter the data.

Examples

```
# Fatally injured 30 year old female drivers in Maryland in 2022
get_fars_crash_persons(
  state = "MD",
  age = 30,
  sex = "female",
  seat = "front left",
  injury = "fatal",
  year = 2022
)
```

`get_fars_crash_vehicles`*Get Crashes By Vehicle*

Description

This function returns a list of fatal crashes involving a specific vehicle make, model, model year, and body type using the CrashAPI `GetCrashesByVehicle` endpoint. The make, model, and body type can be names or ID numbers matching the options returned by `fars_vars()`. Names are converted to ID numbers for use in the API query.

Usage

```
get_fars_crash_vehicles(
  state,
  make,
  model,
  model_year,
  body_type,
  year = getOption("crashapi.year", 2024),
  var_labels = c("attribute", "none")
)
```

Arguments

state	Required. State name, abbreviation, or FIPS number.
make	Required. Make name or ID, e.g. "Ford" or 12. Valid options are returned by <code>fars_vars(var = "make")</code> for the earliest year in year.
model	Required. Model name or ID. Valid options are returned by <code>fars_vars(var = "model", make = make)</code> .
model_year	Required. Vehicle model year.
body_type	Required. Body type name or ID. Valid options are returned by <code>fars_vars(var = "bodytype", make = make, model = model)</code> .
year	numeric vector. One or more years from 2010 to 2024. The minimum and maximum values are used as the start and end year. Defaults to <code>getOption("crashapi.year", 2024)</code> .
var_labels	How to add variable labels from fars_vars_labels : "attribute" (default) adds a "label" attribute to each column, which is shown by <code>View()</code> in RStudio and Positron and used by packages such as <code>gt</code> and <code>gtsummary</code> , and "none" doesn't add variable labels.

API behavior

NHTSA's documentation says this endpoint accepts "one or more" vehicle details for make, model, model year, and body type. However, testing in October 2026 found that the API does not behave this way: the API returns an error (HTTP 404) unless all four are provided and an error message if any are blank. For this reason, `make`, `model`, `model_year`, and `body_type` are all required.

To find crashes using other vehicle criteria, download the vehicle table with `get_fars_year()` using `type = "vehicle"` and filter the data.

Examples

```
get_fars_crash_vehicles(
  state = "CA",
  make = "Ford",
  model = 37,
  model_year = 2012,
  body_type = 4,
  year = 2022
)
```

Description

These functions access specific endpoints of DOT NHTSA's CrashAPI:

- `get_fars_crashes` returns a list of fatal crashes in a county.
- `get_fars_cases` returns details for one or more fatal crashes by case number.
- `get_fars_crash_list` returns a list of fatal crashes that have occurred in one or more states with a range of vehicles involved.
- `get_fars_summary` provides a count of crashes and fatalities by year for a state.
- `get_fars_year` provides a FARS data table for a state.

Both `get_fars_crash_list` and `get_fars_crashes` limit the returned data to 5000 records so consider limiting the range of years requested if data exceeds that threshold.

Use `get_fars()` to get FARS data tables for any state, county, or year (including national data and years before 2010) from the CrashAPI or the NHTSA data downloads site.

Usage

```
get_fars_crashes(  
  state,  
  county,  
  year = getOption("crashapi.year", 2024),  
  details = FALSE,  
  geometry = FALSE,  
  crs = NULL,  
  var_labels = c("attribute", "none")  
)
```

```
get_fars_cases(  
  state,  
  cases,  
  year = getOption("crashapi.year", 2024),  
  details = FALSE,  
  geometry = FALSE,  
  crs = NULL,  
  var_labels = c("attribute", "none")  
)
```

```
get_fars_crash_list(  
  state,  
  year = getOption("crashapi.year", 2024),  
  vehicles = c(1, 50),  
  var_labels = c("attribute", "none")  
)
```

```
get_fars_summary(  
  state,  
  year = getOption("crashapi.year", 2024),
```

```

  var_labels = c("attribute", "none")
)

get_fars_year(
  state,
  year = getOption("crashapi.year", 2024),
  type = "accident",
  format = "json",
  path = NULL,
  geometry = FALSE,
  crs = NULL,
  download = FALSE,
  var_labels = c("attribute", "none"),
  call = caller_env()
)

```

Arguments

state	Required. State name, abbreviation, or FIPS number. <code>get_fars_crash_list</code> supports multiple states.
county	County name or FIPS number. Required for <code>get_fars_crashes</code> .
year	numeric vector. One or more years. For <code>get_fars_crashes</code> , <code>get_fars_crash_list</code> , <code>get_fars_summary</code> , and <code>get_fars_year</code> , the minimum and maximum values are used as the start and end year, so <code>year = 2015:2020</code> and <code>year = c(2015, 2020)</code> both request data from 2015 through 2020 (with a message if year has gaps). For <code>get_fars_cases</code> , year must be a single year or one year for each case in cases. The CrashAPI supports the years from 2010 through 2024. Defaults to <code>getOption("crashapi.year", 2024)</code> so the default year can be set for all functions with <code>options(crashapi.year = 2022)</code> .
details	For <code>get_fars_crashes</code> , if TRUE, detailed case data (excluding event and vehicle data) is joined to the crash data. For <code>get_fars_cases</code> , if TRUE, event and vehicle data are included as list columns, or use "events" or "vehicles" to return the event or vehicle data for all cases. Default: FALSE.
geometry	If TRUE, return sf object. Optional for <code>get_fars_crashes</code> .
crs	Coordinate reference system to return for <code>get_fars_crashes</code> if geometry is TRUE.
var_labels	How to add variable labels from <code>fars_vars_labels</code> : "attribute" (default) adds a "label" attribute to each column, which is shown by <code>View()</code> in RStudio and Positron and used by packages such as <code>gt</code> and <code>gtsummary</code> , and "none" doesn't add variable labels.
cases	One or more FARS case numbers. Required for <code>get_fars_cases</code> .
vehicles	numeric vector with the minimum and maximum number of vehicles, e.g. <code>c(1, 2)</code> for minimum of 1 vehicle and maximum of 2. Required for <code>get_fars_crash_list</code> .
type	Name of the FARS data table. Supported values include "ACCIDENT", "CEVENT", "DAMAGE", "DISTRACT", "DRIMPAIR", "DRUGS", "FACTOR", "MANEUVER", "NMCRASH", "NMIMPAIR", "NMPRIOR", "PARKWORK", "PBTYPE",

	"PERSON", "SAFETYEQ", "VEHICLE", "VEVENT", "VINDECODE", "VIN-DERIVED", "VIOLATION", "VISION", and "VSOE". "TRAILERVINDERIVED" is supported for 2016 onwards, "NMDISTRACT" and "RACE" for 2019 onwards, and "CRASHRF", "DRIVERRF", "PERSONRF", "PVEHICLESF", "VEHICLESF", and "WEATHER" for 2020 onwards for get_fars_year. Lower-case or mixed case values are permitted.
format	Format to request from the CrashAPI for get_fars_year ("json" or "csv"). Default: "json".
path	Directory to save the file if download is TRUE for get_fars_year.
download	logical. If TRUE, get_fars_year downloads the data to a file instead of returning a data frame. Default FALSE.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

Examples

```
head(get_fars_crashes(state = "MD", county = "Baltimore city"), 5)

get_fars_cases(state = "MD", cases = "240274")

get_fars_crash_list(state = "MD", vehicles = 5)

get_fars_summary(state = "MD")

head(get_fars_year(state = "MD", type = "PERSON"), 5)
```

get_fars_zip	Download FARS data files as zipped CSV or SAS files
--------------	---

Description

This function provides an alternative to [get_fars_year\(\)](#) that downloads files directly from the NHTSA downloads site. If read is TRUE, the function returns a list containing data frames for each table available in the selected year. If geometry is TRUE, the accident table is converted to an sf object.

Usage

```
get_fars_zip(
  year = getOption("crashapi.year", 2024),
  format = "csv",
  path = NULL,
  pr = FALSE,
  aux = FALSE,
  read = TRUE,
```

```

    geometry = FALSE,
    overwrite = FALSE,
    keep_files = FALSE,
    var_labels = c("attribute", "none")
  )

```

Arguments

year	One or more years of data from 1975 to 2024. Each year is downloaded separately and, if read = TRUE, returned as a list named by year. Defaults to <code>getOption("crashapi.year", 2024)</code> .
format	Format of zipped data tables ('csv' or 'sas'). Default: 'csv'. read = TRUE and geometry = TRUE are only supported if format is "csv".
path	Directory for downloaded and cached files. If NULL (default), uses <code>getOption("crashapi.cache_dir")</code> or the user cache directory from <code>tools::R_user_dir("crashapi", "cache")</code> .
pr	If TRUE, download FARS data for Puerto Rico. Puerto Rico data is available for 1978 and from 1980 onwards. Default: FALSE
aux	If TRUE and year is 1982 or later, download auxiliary FARS datasets that "contain data derived from the regular FARS/GES variables using NCSA analytical data classifications." In 2010, the NHTSA explained: "These classifications are widely used in NCSA publications and research. Many definitions such as "speeding-related" or "distracted driving" comprise a combination of variables whose names or attributes have changed over time. The derived variables in the auxiliary files incorporate these nuances, thus simplifying the use of standard classifications in any traffic safety research." Learn more from the FARS and GES Auxiliary Datasets Q & A: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811364
read	If TRUE (default), read the cached parquet files into a list of tables with each list item corresponding to one data file.
geometry	If TRUE, convert the accident table to a sf object.
overwrite	If TRUE, download the data and replace any cached files. Default: FALSE
keep_files	If TRUE, keep the downloaded zip file and the extracted CSV files in the cache folder. Default: FALSE
var_labels	How to add variable labels from <code>fars_vars_labels</code> : "attribute" (default) adds a "label" attribute to each column, which is shown by <code>View()</code> in RStudio and Positron and used by packages such as gt and gtsummary, and "none" doesn't add variable labels.

Details

Zipped CSV files are converted to parquet files and cached in a folder named after the zip file within path (by default, the user cache directory from `tools::R_user_dir()`), and the zip and CSV files are removed unless `keep_files = TRUE`. Later calls read the cached parquet files without downloading the data again unless `overwrite = TRUE`.

Value

If read = TRUE, a list of data frames with one item for each table (with the accident table converted to an sf object if geometry = TRUE). If read = FALSE, the cached parquet file paths named by table (or the zip file path if format = "sas") invisibly. If more than one year is provided, returns a list named by year.

mmucc_codes

*Model Minimum Uniform Crash Criteria (MMUCC) codes (simple)***Description**

A collection of the 73 unique codes identified as simple codes: <https://release.niem.gov/niem/codes/mmucc/4.1/mmucc.xsd>

See the MMUCC Guideline Fifth Edition (2017) for more information: <https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812433>

About MMUC from NHTSA: <https://www.nhtsa.gov/mmucc-1>

To encourage greater uniformity, the National Highway Traffic Safety Administration (NHTSA) and the Governors Highway Safety Association (GHSA) cooperatively developed a voluntary data collection guideline in 1998. The MMUCC guideline identifies a minimum set of motor vehicle crash data elements and their attributes that States should consider collecting and including in their State crash data system.

The MMUCC 5th Edition is the result of an 18-month collaboration between NHTSA, the Federal Highway Administration (FHWA), the Federal Motor Carrier Safety Administration (FMCSA), the National Transportation Safety Board (NTSB), the GHSA, and subject matter experts from State DOTs, local law enforcement, emergency medical services, safety organizations, industry partners, and academia. The traffic records community and general public also contributed through external forums (Federal Register) and at the 2016 Traffic Records Forum.

- Added: March 26 2022
- Updated: March 26 2022

Usage

mmucc_codes

Format

A data frame with 700 rows and 6 variables:

code Attribute code

name Attribute code name

type Attribute code type

definition Code definition

restriction_id Restriction id number

restriction Restriction value

read_crashapi	<i>Read data from the CrashAPI using a url template</i>
---------------	---

Description

An updated utility function using the `httr2` package to read data from the CrashAPI using the API URL templates listed on the NHSTA website: <https://crashviewer.nhtsa.dot.gov/crashviewer/CrashAPI>

Usage

```
read_crashapi(
  url = "https://crashviewer.nhtsa.dot.gov/crashviewer/CrashAPI",
  data = "crashes",
  type = NULL,
  format = "json",
  results = TRUE,
  ...,
  cookie_file = "crashapi.cookies",
  call = caller_env()
)
```

Arguments

<code>url</code>	Base url for CrashAPI.
<code>data</code>	Data (crashes, analytics, or fars), Default: 'crashes'
<code>type</code>	Type of API to use, Default: NULL
<code>format</code>	Format to return, Default: 'json'
<code>results</code>	If FALSE, return formatted url, Default: TRUE
<code>...</code>	Additional parameters used in template (varies by type).
<code>cookie_file</code>	File name passed to <code>tempfile()</code> for the temporary cookie file used to preserve cookies between requests.
<code>call</code>	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

Data frame with requested data or a formatted url (if `results = FALSE`)

Index

- * **datasets**
 - crash_data_index, 2
 - fars_terms, 5
 - fars_vars_labels, 9
 - mmucc_codes, 21
- abort(), 19, 22
- crash_data_index, 2
- decode_fars, 3
- decode_fars(), 6, 13
- fars_terms, 5
- fars_value_labels, 5
- fars_value_labels(), 3
- fars_variables, 7
- fars_variables(), 9
- fars_vars, 8
- fars_vars(), 15
- fars_vars_labels, 7, 9, 10, 11, 13, 14, 16, 18, 20
- format_crashes, 10
- get_crss_zip, 10
- get_crss_zip(), 4
- get_fars, 12
- get_fars(), 4, 17
- get_fars_cases (get_fars_crashes), 16
- get_fars_cases(), 13
- get_fars_crash_details
 - (get_fars_crashes), 16
- get_fars_crash_list (get_fars_crashes), 16
- get_fars_crash_list(), 13
- get_fars_crash_persons, 14
- get_fars_crash_vehicles, 15
- get_fars_crashes, 16
- get_fars_crashes(), 13
- get_fars_summary (get_fars_crashes), 16
- get_fars_summary(), 13
- get_fars_year (get_fars_crashes), 16
- get_fars_year(), 12, 13, 15, 16, 19
- get_fars_zip, 19
- get_fars_zip(), 4, 10, 12, 13
- get_vars (fars_vars), 8
- haven::labelled(), 4, 13
- mmucc_codes, 21
- read_crashapi, 22
- tempfile(), 22
- tools::R_user_dir(), 11, 20
- View(), 10, 13, 14, 16, 18, 20