

# Package: acledR (via r-universe)

February 26, 2025

**Type** Package

**Title** Manipulate ACLED Data

**Version** 0.1.0

**Description** Tools working with data from ACLED (Armed Conflict Location and Event Data). Functions include simplified access to ACLED's API (<<https://apidocs.acleddata.com/>>), methods for keeping local versions of ACLED data up-to-date, and functions for common ACLED data transformations.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, methods, httr, lubridate, stringr, tidyr, magrittr, purrr, rlang, utils

**RoxygenNote** 7.3.2

**Depends** R (>= 3.5.0)

**URL** <https://dtacled.github.io/acledR/>

**BugReports** <https://github.com/dtacled/acledR/issues>

**Suggests** knitr, janitor, rmarkdown, readr, kableExtra, ggplot2, covr, here, secret, sf, raster, forcats, igraph, sjmisc, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Config/pak/sysreqs** libicu-dev libssl-dev

**Repository** <https://dtacled.r-universe.dev>

**RemoteUrl** <https://github.com/dtacled/acledr>

**RemoteRef** HEAD

**RemoteSha** 3d5000aa9b317be6fa388f7c0b45daeda95f4fc6

## Contents

acled_access . . . . .	2
acled_api . . . . .	3
acled_codebook . . . . .	5
acled_countries . . . . .	6
acled_deletions_api . . . . .	6
acled_event_categories . . . . .	7
acled_interaction_codes . . . . .	8
acled_multipliers . . . . .	9
acled_old_deletion_dummy . . . . .	9
acled_old_dummy . . . . .	11
acled_regions . . . . .	12
acled_rounding . . . . .	13
acled_transform_interaction . . . . .	14
acled_transform_longer . . . . .	15
acled_transform_wider . . . . .	16
acled_update . . . . .	17
<b>Index</b>	<b>19</b>

---

acled_access	<i>Store your ACLED access information into your session.</i>
--------------	---

---

### Description

Simple function to authenticate and store (through `Sys.setenv()`) ACLED access key for the `acled_api()` function. If email and key is stored via `acled_access`, the email and key arguments for `acled_api` can be NULL.

### Usage

```
acled_access(email, key)
```

### Arguments

email	This is the email that you register in the ACLED Access portal ( <a href="https://developer.acleddata.com/">https://developer.acleddata.com/</a> )
key	This is the key generated by the ACLED Access portal.

### Value

Returns a success message if ACLED credentials are successfully authorized

### See Also

ACLED API Access guide <https://acleddata.com/download/35300/>  
 Other API and Access: `acled_api()`, `acled_deletions_api()`, `acled_update()`

**Examples**

```
## Not run:
acled_access(email = "your_email", key = "your_key")
Sys.getenv("acled_email")
Sys.getenv("acled_key")

## End(Not run)
```

acled\_api

*Request data from ACLED API***Description**

This function allows users to easily request data from the ACLED API. Users can include variables such as country, regions, dates of interest and the format (monadic or dyadic). The function returns a tibble of the desired ACLED events.

**Usage**

```
acled_api(
  email = NULL,
  key = NULL,
  country = NULL,
  regions = NULL,
  start_date = floor_date(Sys.Date(), "year") - years(1),
  end_date = Sys.Date(),
  timestamp = NULL,
  event_types = NULL,
  population = "none",
  inter_numeric = FALSE,
  monadic = FALSE,
  ...,
  acled_access = TRUE,
  prompt = TRUE,
  log = FALSE
)
```

**Arguments**

email	character string. Email associated with your ACLED account registered at <a href="https://developer.acleddata.com">https://developer.acleddata.com</a> .
key	character string. Access key associated with your ACLED account registered at <a href="https://developer.acleddata.com">https://developer.acleddata.com</a> .
country	character vector. Default is NULL, which will return events for all countries. Pass a vector of countries names to retrieve events from specific countries. The list of ACLED countries. names may be found via <code>acledR::acled_countries</code> .

regions	vector of region names (character) or region codes (numeric). Default is NULL, which will return events for all regions. Pass a vector of regions names or codes to retrieve events from countries. within specific regions. The list of ACLED regions may be found via <code>acledR::acled_regions</code> .
start_date	character string. Format 'yyyy-mm-dd'. The earliest date for which to return events. The default is 1997-01-01, which is the earliest date available.
end_date	character string. Format 'yyyy-mm-dd'. The latest date for which to return events. The default is <code>Sys.Date()</code> , which is the most present date.
timestamp	numerical or character string. Provide a date or datetime written as either a character string of yyyy-mm-dd or as a numeric Unix timestamp to access all events added or updated after that date.
event_types	vector of one or more event types (character). Default is NULL, which will return data for all event types. To return data for only specific event types, request one or more of the following options (not case sensitive): Battles, Violence against civilians, Protests, Riots, Strategic Developments, and Explosions/Remote violence.
population	character. Specify whether to return population estimates for each event. It accepts three options: "none" (default), "best", and "full".
inter_numeric	logical. If FALSE (default), interaction code columns (inter1, inter2, and interaction) returned as strings describing the actor types/interactions. If TRUE, the values are returned as numeric values.
monadic	logical. If FALSE (default), returns dyadic data. If TRUE, returns monadic actor1 data.
...	string. Any additional parameters that users would like to add to their API calls (e.g. interaction or ISO)
acled_access	logical. If TRUE (default), you have used the <code>acled_access</code> function and the email and key arguments are not required.
prompt	logical. If TRUE (default), users will receive an interactive prompt providing information about their call (countries requested, number of estimated events, and number of API calls required) and asking if they want to proceed with the call. If FALSE, the call continues without warning, but the call is split and returns a message specifying how many calls are being made.
log	logical. If TRUE, it provides a dataframe with the countries and days requested, and how many calls it entails. The dataframe is provided INSTEAD of the normal ACLED dataset.

### Value

Returns a tibble of of ACLED events.

### See Also

- ACLED API guide. <https://apidocs.acleddata.com/>

Other API and Access: `acled_access()`, `acled_deletions_api()`, `acled_update()`

## Examples

```
## Not run:

# Get all the events coded by ACLED in Argentina from 01/01/2022 until 02/01/2022
# in dyadic-wide form
argen_acled <- acled_api(
  email = "your_email", key = "your_key",
  country = "Argentina", start_date = "2022-01-01", end_date = "2022-02-01",
  acled_access = FALSE
)

# tibble with all the events from Argentina where each row is one event.
argen_acled

# Get all events coded by ACLED in the Caribbean from 01/01/2022 to 10/01/2022
# in monadic-long form using email and key saved in environment

acled_access(email = "your_email", key = "your_key")
carib_acled <- acled_api(
  regions = "Caribbean", start_date = "2022-01-01",
  end_date = "2022-01-10", monadic = TRUE, acled_access = TRUE
)

## Tibble with all the events from the Caribbean where each row is one actor
carib_acled

## End(Not run)
```

---

acled\_codebook

*ACLED Codebook*

---

## Description

Codebook for ACLED data

## Usage

```
acled_codebook
```

## Format

A data frame:

**Variable** Variable names

**Description** Text description of each variable

**Values** Text description of values for each variable

**See Also**

Other Data: [acled\\_countries](#), [acled\\_event\\_categories](#), [acled\\_interaction\\_codes](#), [acled\\_multipliers](#), [acled\\_old\\_deletion\\_dummy](#), [acled\\_old\\_dummy](#), [acled\\_regions](#)

---

acled_countries	<i>ACLED Countries</i>
-----------------	------------------------

---

**Description**

ACLED country names, regions, and coding start year

**Usage**

```
acled_countries
```

**Format**

A data frame:

**country** Country names

**region** Region names

**start\_year** First year coded by ACLED

**See Also**

Other Data: [acled\\_codebook](#), [acled\\_event\\_categories](#), [acled\\_interaction\\_codes](#), [acled\\_multipliers](#), [acled\\_old\\_deletion\\_dummy](#), [acled\\_old\\_dummy](#), [acled\\_regions](#)

---

acled_deletions_api	<i>Request data from the ACLED Deletions API</i>
---------------------	--

---

**Description**

This function allows users to pull deleted ACLED event IDs from the Deletions API.

**Usage**

```
acled_deletions_api(  
  email = NULL,  
  key = NULL,  
  date_deleted = NULL,  
  acled_access = TRUE,  
  log = FALSE  
)
```

## Arguments

email	character string. Email associated with your ACLED account registered at <a href="https://developer.acleddata.com">https://developer.acleddata.com</a> .
key	character string. Access key associated with your ACLED account registered at <a href="https://developer.acleddata.com">https://developer.acleddata.com</a> .
date_deleted	character string. Format 'yyyy-mm-dd' or Unix timestamp. The query will return all deleted events including and after the requested date/timestamp.
acled_access	logical. If TRUE it means that you have utilized the acled_access function and there is no need for the email and key arguments.
log	Only for testing purposes: you can use this to check if all the variables in your call were handled properly.

## Value

Returns a tibble of ACLED data with columns for event\_id\_cnty and deleted\_timestamp.

## See Also

- [ACLED API guide](#)
- [Keeping ACLED data up to date guide](#)

Other API and Access: [acled\\_access\(\)](#), [acled\\_api\(\)](#), [acled\\_update\(\)](#)

## Examples

```
## Not run:  
  
# Request deleted ACLED events since January 1, 2022  
acled_deletions_api(date_deleted = "2022-01-01", acled_access = TRUE)  
  
## End(Not run)
```

---

acled\_event\_categories  
*ACLED Event Categories*

---

## Description

ACLED event and sub-event types, grouped by category

## Usage

```
acled_event_categories
```

**Format**

A data frame:

**event\_type** ACLED event type

**sub\_event\_type** ACLED sub-event type

**political\_violence** Dummy indicator for whether sub-event type falls within political violence

**organized\_political\_violence** Dummy indicator for whether sub-event type falls within organized political violence

**disorder** Dummy indicator for whether sub-event type falls within disorder

**demonstrations** Dummy indicator for whether sub-event type falls within demonstrations

**See Also**

Other Data: [acled\\_codebook](#), [acled\\_countries](#), [acled\\_interaction\\_codes](#), [acled\\_multipliers](#), [acled\\_old\\_deletion\\_dummy](#), [acled\\_old\\_dummy](#), [acled\\_regions](#)

---

acled\_interaction\_codes

*ACLED interaction codes*

---

**Description**

ACLED interaction and actor types

**Usage**

acled\_interaction\_codes

**Format**

A data frame:

**Inter1/Inter2** Actor type

**Numeric Code** Numeric equivalent found in the inter1 and inter2 column.

**See Also**

Other Data: [acled\\_codebook](#), [acled\\_countries](#), [acled\\_event\\_categories](#), [acled\\_multipliers](#), [acled\\_old\\_deletion\\_dummy](#), [acled\\_old\\_dummy](#), [acled\\_regions](#)



---

acled_multipliers	<i>ACLED Multipliers</i>
-------------------	--------------------------

---

**Description**

A dataframe with additional information for each country, only for the purpose of estimating events.

**Usage**

```
acled_multipliers
```

**Format**

A data frame:

**country** Country names

**bin** Bin of event frequency

**year** Year corresponding to the bin

**avg\_month\_bin** Average monthly of the bin

**See Also**

Other Data: [acled\\_codebook](#), [acled\\_countries](#), [acled\\_event\\_categories](#), [acled\\_interaction\\_codes](#), [acled\\_old\\_deletion\\_dummy](#), [acled\\_old\\_dummy](#), [acled\\_regions](#)

---

acled\_old\_deletion\_dummy

*Second dummy data frame of ACLED events emulating an old format, used in [acled\\_deletion\\_api](#) Vignette*

---

**Description**

Large dataset of multiple regions and countries, purposefully including deleted/modified events.

**Usage**

```
acled_old_deletion_dummy
```

**Format**

A data frame:

**event\_id\_cnty** An unique individual identifier by number and country acronym (updated annually)

**event\_date** The day, month and year on which an event took place

**year** The year in which an event took place

**time\_precision** A numeric code indicating the level of certainty of the date coded for the event

**disorder\_type** Type of disorder associated with the event and sub event type

**event\_type** The type of event

**sub\_event\_type** The type of sub-event

**actor1** The named actor involved in the event. Note: Actor 1 and Actor 2 do not imply directionality (e.g. attacker or defender)

**assoc\_actor\_1** The named actor associated with or identifying actor1

**inter1** A numeric code indicating the type of actor1

**actor2** The named actor involved in the event. Note: Actor 1 and Actor 2 do not imply directionality (e.g. attacker or defender)

**assoc\_actor\_2** The named actor associated with or identifying actor1

**inter2** A numeric code indicating the type of actor1

**interaction** A numeric code indicating the interaction between types of actor1 and actor2

**civilian\_targeting** Column referencing the presence of civilian targeting

**iso** A numeric code for each individual country

**region** The region of the world where the event took place

**country** The country in which the event took place

**admin1** The largest sub-national administrative region in which the event took place

**admin2** The second largest sub-national administrative region in which the event took place

**admin3** The third largest sub-national administrative region in which the event took place

**location** The location in which the event took place

**latitude** The latitude of the location

**longitude** The longitude of the location

**geo\_precision** A numeric code indicating the level of certainty of the location coded for the event

**source** The source of the event report

**source\_scale** The scale (local, regional, national, international) of the source

**notes** A short description of the event

**fatalities** The number of reported fatalities which occurred during the event

**tags** Tags associated with the event.

**timestamp** Numeric code of time

**See Also**

Other Data: [acled\\_codebook](#), [acled\\_countries](#), [acled\\_event\\_categories](#), [acled\\_interaction\\_codes](#), [acled\\_multipliers](#), [acled\\_old\\_dummy](#), [acled\\_regions](#)

---

acled_old_dummy	<i>A dummy data frame of ACLED events emulating an old format, used in "Keeping your dataset updated" Vignette</i>
-----------------	--

---

### Description

Small dataset of events in Argentina, purposefully including events which are currently deleted/modified.

### Usage

acled\_old\_dummy

### Format

A data frame:

**event\_id\_cnty** An unique individual identifier by number and country acronym (updated annually)

**event\_date** The day, month and year on which an event took place

**year** The year in which an event took place

**time\_precision** A numeric code indicating the level of certainty of the date coded for the event

**disorder\_type** Type of disorder associated with the event and sub event type

**event\_type** The type of event

**sub\_event\_type** The type of sub-event

**actor1** The named actor involved in the event. Note: Actor 1 and Actor 2 do not imply directionality (e.g. attacker or defender)

**assoc\_actor\_1** The named actor associated with or identifying actor1

**inter1** A numeric code indicating the type of actor1

**actor2** The named actor involved in the event. Note: Actor 1 and Actor 2 do not imply directionality (e.g. attacker or defender)

**assoc\_actor\_2** The named actor associated with or identifying actor1

**inter2** A numeric code indicating the type of actor1

**interaction** A numeric code indicating the interaction between types of actor1 and actor2

**civilian\_targeting** Column referencing the presence of civilian targeting

**iso** A numeric code for each individual country

**region** The region of the world where the event took place

**country** The country in which the event took place

**admin1** The largest sub-national administrative region in which the event took place

**admin2** The second largest sub-national administrative region in which the event took place

**admin3** The third largest sub-national administrative region in which the event took place

**location** The location in which the event took place

**latitude** The latitude of the location

**longitude** The longitude of the location

**geo\_precision** A numeric code indicating the level of certainty of the location coded for the event

**source** The source of the event report

**source\_scale** The scale (local, regional, national, international) of the source

**notes** A short description of the event

**fatalities** The number of reported fatalities which occurred during the event

**tags** Tags associated with the event.

**timestamp** Numeric code of time

### See Also

Other Data: [acled\\_codebook](#), [acled\\_countries](#), [acled\\_event\\_categories](#), [acled\\_interaction\\_codes](#), [acled\\_multipliers](#), [acled\\_old\\_deletion\\_dummy](#), [acled\\_regions](#)

---

acled\_regions

*ACLED Regions*

---

### Description

ACLED region names, region numbers, and coding start dates

### Usage

acled\_regions

### Format

A data frame:

**region** Region number

**region\_name** Region names

**first\_event\_date** First date (yyyy-mm-dd) coded by ACLED

### See Also

Other Data: [acled\\_codebook](#), [acled\\_countries](#), [acled\\_event\\_categories](#), [acled\\_interaction\\_codes](#), [acled\\_multipliers](#), [acled\\_old\\_deletion\\_dummy](#), [acled\\_old\\_dummy](#)

---

acled_rounding	<i>Rounding function</i>
----------------	--------------------------

---

### Description

This function addresses some of the conflicts of rounding in R, especially when trying to round up.

### Usage

```
acled_rounding(num, digits = 0)
```

### Arguments

num	int. This is the number we are trying to round.
digits	int. Where do we want to round up. It accepts 0 (whole number), 1 (tenth place), 2 (hundredths), etc.

### Details

This function is meant to address the problem of rounding in R where the approach is always round to even. The function is meant to round things following the simple rule. If the decimal is 5+ then round up, if not round down. With the 'digits' argument, one can set up the specificity of the rounding, 0= whole number, 1 = tenth place, 2=hundreds place, and so on.

### Value

A rounded numeric value

### Examples

```
x1 <- 1.569
x2 <- 104.530
x3 <- 54.430
x4 <- 205.49999
acled_rounding(x1)
acled_rounding(x2)
acled_rounding(x3)
acled_rounding(x4)
```

---

`acled_transform_interaction`*Change interaction codes from numeric labels to string labels*

---

### Description

This function allows users to change from numeric interaction codes (i.e. 1, 2, 3, etc) to string interaction codes (i.e. State Forces, Rebel Group, etc.)

### Usage

```
acled_transform_interaction(df, only_inters = FALSE)
```

### Arguments

`df` dataframe. ACLED data including at least `inter1`, `inter2` columns. If `only_inters` is TRUE, it also requires `interaction` column.

`only_inters` boolean. Option whether to include the `interaction` column in the transformation (if TRUE) or to only use `inter1` and `inter2` (if FALSE).

### Value

Returns a tibble of of ACLED events with modified `inter1`, `inter2` and potentially `interaction` columns .

### See Also

Other Data Manipulation: [acled\\_transform\\_longer\(\)](#), [acled\\_transform\\_wider\(\)](#)

### Examples

```
## Not run:  
  
# Load data frame  
argen_acled <- acled_api(  
  email = "your_email", key = "your_key",  
  country = "Argentina", start_date = "2022-01-01", end_date = "2022-02-01",  
  acled_access = FALSE  
)  
  
# Transform the interactions  
argen_acled_transformed <- acled_transformation_interaction(argen_acled, only_inters = F)  
  
## End(Not run)
```

---

`acled_transform_longer`*Transform ACLED data from wide to long*

---

## Description

Function to convert your ACLED's API calls (if dyadic) into desired monadic forms.

## Usage

```
acled_transform_longer(data, type = "full_actors")
```

## Arguments

<code>data</code>	dataframe or tibble containing your dataset.
<code>type</code>	character string. One of five types: <code>full_actors</code> , <code>main_actors</code> , <code>assoc_actors</code> , <code>source</code> , or <code>all</code> . <ul style="list-style-type: none"><li>• <code>full_actors</code>: All actor and associated actor columns</li><li>• <code>main_actors</code>: Actor 1 and Actor 2 columns</li><li>• <code>assoc_actors</code>: All associated actor columns</li><li>• <code>source</code>: The source column becomes monadic</li></ul>

## Value

A tibble with the data transformed into long form.

## See Also

Other Data Manipulation: [acled\\_transform\\_interaction\(\)](#), [acled\\_transform\\_wider\(\)](#)

## Examples

```
## Not run:
# argen_acled <- acled_api(country = "Argentina", start_date = "2022-01-01",
#                          end_date="2022-02-01", acled_access = T, prompt = F)

# argen_acled_long_actors <- acled_transform_wide_to_long(argen_acled,
#                                                       type = "full_actor") # Transforming the data

# nrow(argen_acled_long_actors) # Number of rows in the dataset
# [1] 263 # Long form

# nrow(argen_acled) ) # Number of rows in the dataset
# [1] 145 # Wide form

## End(Not run)
```

---

acled\_transform\_wider *Reverse Transform ACLED Data from Long to Wide*

---

### Description

Function to convert your ACLED's API calls (if monadic) back into the original dyadic forms.

### Usage

```
acled_transform_wider(data, type = "full_actors")
```

### Arguments

data	a dataframe or tibble containing your dataset.
type	a character string. One of five types: full_actors, main_actors, assoc_actors, source, or all. <ul style="list-style-type: none"> <li>full_actors: All actor and associated actor columns</li> <li>main_actors: Actor 1 and Actor 2 columns</li> <li>assoc_actors: All associated actor columns</li> <li>source: The source column becomes dyadic</li> <li>api_monadic: Use this option for data that is the output of the API's monadic option.</li> </ul>

### Value

A tibble with the data transformed back into wide form.

### See Also

Other Data Manipulation: [acled\\_transform\\_interaction\(\)](#), [acled\\_transform\\_longer\(\)](#)

### Examples

```
## Not run:
# argen_acled <- acled_api(country = "Argentina", start_date = "2022-01-01",
#                          end_date="2022-02-01", acled_access = T, prompt = F)
# argen_acled_long_actors <- acled_transform_longer(argen_acled,
#                                                  type = "full_actor") # Transforming the data to long form

# argen_acled_wide <- acled_transform_wider(argen_acled_long_actors,
#                                          type = "full_actor") # Transforming the data back to wide form

# nrow(argen_acled_wide) # Number of rows in the dataset
# [1] 145 # Wide form

# nrow(argen_acled_long_actors) # Number of rows in the dataset
# [1] 263 # Long form

## End(Not run)
```



---

acled_update	<i>Updating your ACLED dataset</i>
--------------	------------------------------------

---

### Description

This function is meant to help you keep your dataset updated, by automatically checking for new and modified events, as well as deleted events (if `deleted = TRUE`). Note: The function makes new API calls to gather new and modified events.

### Usage

```
acled_update(
  df,
  start_date = min(df$event_date),
  end_date = max(df$event_date),
  additional_countries = "current countries",
  regions = NULL,
  event_types = NULL,
  acled_access = TRUE,
  email = NULL,
  key = NULL,
  inter_numeric = FALSE,
  deleted = TRUE,
  prompts = TRUE
)
```

### Arguments

<code>df</code>	The dataframe to update, it has to have the same structure as ACLED's dyadic dataframe (i.e. the result of <code>acled_api()</code> )
<code>start_date</code>	The first date of events you want to update from.. These are the ceiling and floor of <i>event_date</i> , not of <i>timestamp</i> .
<code>end_date</code>	The last date of events you want to update from. These are the ceiling and floor of <i>event_date</i> , not of <i>timestamp</i> .
<code>additional_countries</code>	string. Additional <code>additional_countries</code> to update your dataset. It defaults to "current countries", which includes all the <code>additional_countries</code> inside your dataset.
<code>regions</code>	string. The regions for which you would like events in your dataset updated.
<code>event_types</code>	string. The event types for which you would like events in your dataset updated.
<code>acled_access</code>	logical. If you have already used <code>acled_access()</code> , you can set this option as <code>TRUE</code> (default) to avoid having to input your email and access key.
<code>email</code>	character string. Email associated with your ACLED account registered at <a href="https://developer.acleddata.com">https://developer.acleddata.com</a> .
<code>key</code>	character string. Access key associated with your ACLED account registered at <a href="https://developer.acleddata.com">https://developer.acleddata.com</a> .

inter_numeric	logical. If FALSE (default), interaction code columns (inter1, inter2, and interaction) returned as strings describing the actor types/interactions. If TRUE, the values are returned as numeric values. Must match the inter type (numeric or string) in the dataframe being updated.
deleted	logical. If TRUE (default), the function will also remove deleted events using <code>acled_deletions_api()</code> .
prompts	logical. If TRUE (default), users will receive an interactive prompt providing information about their call (additional_countries requested, number of country-days, and number of API calls required) and asking if they want to proceed with the call. If FALSE, the call continues without warning, but the call is split and returns a message specifying how many calls are being made.

### Value

Tibble with updated ACLED data and a newer timestamp.

### See Also

- ACLED Keeping your dataset updated guide. <https://acleddata.com/download/35179/>  
Other API and Access: `acled_access()`, `acled_api()`, `acled_deletions_api()`

### Examples

```
## Not run:
# Updating dataset to include newer data from Argentina

acledR::acled_access(email = "your_email", key = "your_key")

new_argen_dataset <- acled_update(acledR::acled_old_dummy,
  additional_countries = "Argentina",
  acled_access = TRUE,
  prompts = FALSE
)

## End(Not run)
```

# Index

## \* API and Access

acled\_access, 2  
acled\_api, 3  
acled\_deletions\_api, 6  
acled\_update, 17

## \* Data Manipulation

acled\_transform\_interaction, 14  
acled\_transform\_longer, 15  
acled\_transform\_wider, 16

## \* Data

acled\_codebook, 5  
acled\_countries, 6  
acled\_event\_categories, 7  
acled\_interaction\_codes, 8  
acled\_multipliers, 9  
acled\_old\_deletion\_dummy, 9  
acled\_old\_dummy, 11  
acled\_regions, 12

## \* Helpers

acled\_rounding, 13

## \* datasets

acled\_codebook, 5  
acled\_countries, 6  
acled\_event\_categories, 7  
acled\_interaction\_codes, 8  
acled\_multipliers, 9  
acled\_old\_deletion\_dummy, 9  
acled\_old\_dummy, 11  
acled\_regions, 12

acled\_access, 2, 4, 7, 18  
acled\_api, 2, 3, 7, 18  
acled\_codebook, 5, 6, 8–10, 12  
acled\_countries, 6, 6, 8–10, 12  
acled\_deletions\_api, 2, 4, 6, 18  
acled\_event\_categories, 6, 7, 8–10, 12  
acled\_interaction\_codes, 6, 8, 8, 9, 10, 12  
acled\_multipliers, 6, 8, 9, 10, 12  
acled\_old\_deletion\_dummy, 6, 8, 9, 9, 12  
acled\_old\_dummy, 6, 8–10, 11, 12

acled\_regions, 6, 8–10, 12, 12  
acled\_rounding, 13  
acled\_transform\_interaction, 14, 15, 16  
acled\_transform\_longer, 14, 15, 16  
acled\_transform\_wider, 14, 15, 16  
acled\_update, 2, 4, 7, 17