Package ‘excluder’

January 13, 2024

Title Checks for Exclusion Criteria in Online Data

Version 0.5.1

Description Data that are collected through online sources such as Mechanical Turk may require excluding rows because of IP address duplication, geolocation, or completion duration. This package facilitates exclusion of these data for Qualtrics datasets.

License GPL (>= 3)


BugReports https://github.com/ropensci/excluder/issues/

Depends R (>= 3.5.0)

Imports cli, curl, dplyr, ipaddress, janitor, lubridate, magrittr, maps, rlang, stringr, tidyr, tidyselect

Suggests covr, knitr, lifecycle, readr, rmarkdown, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Jeffrey R. Stevens [aut, cre, cph] (<https://orcid.org/0000-0003-2375-1360>), Joseph O’Brien [rev] (<https://orcid.org/0000-0001-9851-5077>), Julia Silge [rev] (<https://orcid.org/0000-0002-3671-836X>)

Maintainer Jeffrey R. Stevens <jeffrey.r.stevens@protonmail.com>

Repository CRAN

Date/Publication 2024-01-13 17:50:02 UTC
R topics documented:

- check_duplicates
- check_duration
- check_ip
- check_location
- check_preview
- check_progress
- check_resolution
- deidentify
- exclude_duplicates
- exclude_duration
- exclude_ip
- exclude_location
- exclude_preview
- exclude_progress
- exclude_resolution
- mark_duplicates
- mark_duration
- mark_ip
- mark_location
- mark_preview
- mark_progress
- mark_resolution
- qualtrics_fetch
- qualtrics_fetch2
- qualtrics_numeric
- qualtrics_raw
- qualtrics_text
- remove_label_rows
- rename_columns
- unite_exclusions
- use_labels

Index

check_duplicates Check for duplicate IP addresses and/or locations

Description

The `check_duplicates()` function subsets rows of data, retaining rows that have the same IP address and/or same latitude and longitude. The function is written to work with data from Qualtrics surveys.
check_duplicates

Usage

check_duplicates(
  x,
  id_col = "ResponseId",
  ip_col = "IPAddress",
  location_col = c("LocationLatitude", "LocationLongitude"),
  rename = TRUE,
  dupl_ip = TRUE,
  dupl_location = TRUE,
  include_na = FALSE,
  keep = FALSE,
  quiet = FALSE,
  print = TRUE
)

Arguments

x          Data frame (preferably imported from Qualtrics using {qualtRics}).
id_col     Column name for unique row ID (e.g., participant).
ip_col     Column name for IP addresses.
location_col Two element vector specifying columns for latitude and longitude (in that order).
rename     Logical indicating whether to rename columns (using rename_columns()).
dupl_ip     Logical indicating whether to check IP addresses.
dupl_location Logical indicating whether to check latitude and longitude.
include_na Logical indicating whether to include rows with NAs for IP address and location
              as potentially excluded rows.
keep        Logical indicating whether to keep or remove exclusion column.
quiet       Logical indicating whether to print message to console.
print       Logical indicating whether to print returned tibble to console.

Details

To record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the qualtRics::fetch_survey(). By default, IP address and location are both checked, but they can be checked separately with the dupl_ip and dupl_location arguments.

The function outputs to console separate messages about the number of rows with duplicate IP addresses and rows with duplicate locations. These counts are computed independently, so rows may be counted for both types of duplicates.
check_duration

Description

The check_duration() function subsets rows of data, retaining rows that have durations that are too fast or too slow. The function is written to work with data from Qualtrics surveys.

Usage

check_duration(
  x,
  min_duration = 10,
  max_duration = NULL,
  id_col = "ResponseId",
  duration_col = "Duration (in seconds)",
  rename = TRUE,
  keep = FALSE,
check_duration

quiet = FALSE,
print = TRUE
)

Arguments

x      Data frame (preferably imported from Qualtrics using {qualtRics}).
min_duration Minimum duration that is too fast in seconds.
max_duration Maximum duration that is too slow in seconds.
id_col  Column name for unique row ID (e.g., participant).
duration_col Column name for durations.
rename Logical indicating whether to rename columns (using rename_columns())
keep Logical indicating whether to keep or remove exclusion column.
quiet Logical indicating whether to print message to console.
print Logical indicating whether to print returned tibble to console.

Details

Default column names are set based on output from the qualtRics::fetch_survey(). By default, minimum durations of 10 seconds are checked, but either minima or maxima can be checked with the min_duration and max_duration arguments. The function outputs to console separate messages about the number of rows that are too fast or too slow.

This function returns the fast and slow rows.

Value

An object of the same type as x that includes the rows with fast and/or slow duration. For a function that marks these rows, use mark_duration(). For a function that excludes these rows, use exclude_duration().

See Also

Other duration functions: exclude_duration(), mark_duration()
Other check functions: check_duplicates(), check_ip(), check_location(), check_preview(), check_progress(), check_resolution()

Examples

# Check for durations faster than 100 seconds
data(qualtrics_text)
check_duration(qualtrics_text, min_duration = 100)

# Remove preview data first
qualtrics_text %>%
  exclude_preview() %>%
  check_duration(min_duration = 100)
# Check only for durations slower than 800 seconds
qualtrics_text %>%
  exclude_preview() %>%
  check_duration(max_duration = 800)

# Do not print rows to console
qualtrics_text %>%
  exclude_preview() %>%
  check_duration(min_duration = 100, print = FALSE)

# Do not print message to console
qualtrics_text %>%
  exclude_preview() %>%
  check_duration(min_duration = 100, quiet = TRUE)

---

check_ip  
Check for IP addresses from outside of a specified country.

**Description**

The check_ip() function subsets rows of data, retaining rows that have IP addresses from outside the specified country. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
cHECK_ip(
  x,
  id_col = "ResponseId",
  ip_col = "IPAddress",
  rename = TRUE,
  country = "US",
  include_na = FALSE,
  keep = FALSE,
  quiet = FALSE,
  print = TRUE
)
```

**Arguments**

- **x**: Data frame or tibble (preferably imported from Qualtrics using `qualtRics`).
- **id_col**: Column name for unique row ID (e.g., participant).
- **ip_col**: Column name for IP addresses.
- **rename**: Logical indicating whether to rename columns (using `rename_columns()`).
- **country**: Two-letter abbreviation of country to check (default is "US").
- **include_na**: Logical indicating whether to include rows with NA in IP address column in the output list of potentially excluded data.
check_ip

keep Logical indicating whether to keep or remove exclusion column.
quiet Logical indicating whether to print message to console.
print Logical indicating whether to print returned tibble to console.

Details

to record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the qualtrics::fetch_survey(). The function uses ipaddress::country_networks() to assign IP addresses to specific countries using ISO 3166-1 alpha-2 country codes.

The function outputs to console a message about the number of rows with IP addresses outside of the specified country. If there are NAs for IP addresses (likely due to including preview data—see check_preview()), it will print a message alerting to the number of rows with NAs.

Value

An object of the same type as x that includes the rows with IP addresses outside of the specified country. For a function that marks these rows, use mark_ip(). For a function that excludes these rows, use exclude_ip().

Note

This function requires internet connectivity as it uses the ipaddress::country_networks() function, which pulls daily updated data from https://www.iwik.org/ipcountry/. It only updates the data once per session, as it caches the results for future work during the session.

See Also

Other ip functions: exclude_ip(), mark_ip()

Other check functions: check_duplicates(), check_duration(), check_location(), check_preview(), check_progress(), check_resolution()

Examples

# Check for IP addresses outside of the US
data(qualtrics_text)
check_ip(qualtrics_text)

# Remove preview data first
qualtrics_text %>%
  exclude_preview() %>%
  check_ip()

# Check for IP addresses outside of Germany
qualtrics_text %>%
  exclude_preview() %>%
  check_ip(country = "DE")
check_location

Check for locations outside of the US

Description

The check_location() function subsets rows of data, retaining rows that have locations outside of the US. The function is written to work with data from Qualtrics surveys.

Usage

check_location(
  x,
  id_col = "ResponseId",
  location_col = c("LocationLatitude", "LocationLongitude"),
  rename = TRUE,
  include_na = FALSE,
  keep = FALSE,
  quiet = FALSE,
  print = TRUE
)

Arguments

x Data frame (preferably imported from Qualtrics using the qualRics package).

id_col Column name for unique row ID (e.g., participant).

location_col Two element vector specifying columns for latitude and longitude (in that order).

rename Logical indicating whether to rename columns (using rename_columns()).

include_na Logical indicating whether to include rows with NA in latitude and longitude columns in the output list of potentially excluded data.

keep Logical indicating whether to keep or remove exclusion column.

quiet Logical indicating whether to print message to console.

print Logical indicating whether to print returned tibble to console.
check_preview

Details

To record this information in your Qualtrics survey, you must ensure that **Anonymize responses is disabled**.

Default column names are set based on output from the `qualtrics::fetch_survey()`. The function only works for the United States. It uses the `maps::map.where()` to determine if latitude and longitude are inside the US.

The function outputs to console a message about the number of rows with locations outside of the US.

Value

The output is a data frame of the rows that are located outside of the US and (if `include_na == FALSE`) rows with no location information. For a function that marks these rows, use `mark_location()`. For a function that excludes these rows, use `exclude_location()`.

See Also

Other location functions: `exclude_location()`, `mark_location()`

Other check functions: `check_duplicates()`, `check_duration()`, `check_ip()`, `check_preview()`, `check_progress()`, `check_resolution()`

Examples

```r
# Check for locations outside of the US
data(qualtrics_text)
check_location(qualtrics_text)

# Remove preview data first
qualtrics_text %>%
  exclude_preview() %>%
  check_location()

# Do not print rows to console
qualtrics_text %>%
  exclude_preview() %>%
  check_location(print = FALSE)

# Do not print message to console
qualtrics_text %>%
  exclude_preview() %>%
  check_location(quiet = TRUE)
```
Description

The `check_preview()` function subsets rows of data, retaining rows that are survey previews. The function is written to work with data from Qualtrics surveys.

Usage

```r
check_preview(
  x,
  id_col = "ResponseId",
  preview_col = "Status",
  rename = TRUE,
  keep = FALSE,
  quiet = FALSE,
  print = TRUE
)
```

Arguments

- `x` : Data frame (preferably imported from Qualtrics using `{qualtrics}`).
- `id_col` : Column name for unique row ID (e.g., participant).
- `preview_col` : Column name for survey preview.
- `rename` : Logical indicating whether to rename columns (using `rename_columns()`).
- `keep` : Logical indicating whether to keep or remove exclusion column.
- `quiet` : Logical indicating whether to print message to console.
- `print` : Logical indicating whether to print returned tibble to console.

Details

Default column names are set based on output from the `{qualtrics}`::`fetch_survey()`. The preview column in Qualtrics can be a numeric or character vector depending on whether it is exported as choice text or numeric values. This function works for both.

The function outputs to console a message about the number of rows that are survey previews.

Value

The output is a data frame of the rows that are survey previews. For a function that marks these rows, use `mark_preview()`. For a function that excludes these rows, use `exclude_preview()`.

See Also

Other preview functions: `exclude_preview()`, `mark_preview()`

Other check functions: `check_duplicates()`, `check_duration()`, `check_ip()`, `check_location()`, `check_progress()`, `check_resolution()`
Examples

# Check for survey previews
data(qualtrics_text)
check_preview(qualtrics_text)

# Works for Qualtrics data exported as numeric values, too
qualtrics_numeric %>%
  check_preview()

# Do not print rows to console
qualtrics_text %>%
  check_preview(print = FALSE)

# Do not print message to console
qualtrics_text %>%
  check_preview(quiet = TRUE)

check_progress

Check for survey progress

Description

The `check_progress()` function subsets rows of data, retaining rows that have incomplete progress. The function is written to work with data from Qualtrics surveys.

Usage

```r
check_progress(
  x, 
  min_progress = 100, 
  id_col = "ResponseId", 
  finished_col = "Finished", 
  progress_col = "Progress", 
  rename = TRUE, 
  keep = FALSE, 
  quiet = FALSE, 
  print = TRUE
)
```

Arguments

- `x`      Data frame (preferably imported from Qualtrics using `{qualtRics}`).
- `min_progress` Amount of progress considered acceptable to include.
- `id_col`   Column name for unique row ID (e.g., participant).
- `finished_col` Column name for whether survey was completed.
- `progress_col` Column name for percentage of survey completed.
rename Logical indicating whether to rename columns (using `rename_columns()`)
keep Logical indicating whether to keep or remove exclusion column.
quiet Logical indicating whether to print message to console.
print Logical indicating whether to print returned tibble to console.

Details
Default column names are set based on output from the `qualtrics::fetch_survey()`. The default requires 100% completion, but lower levels of completion maybe acceptable and can be allowed by specifying the `min_progress` argument. The finished column in Qualtrics can be a numeric or character vector depending on whether it is exported as choice text or numeric values. This function works for both.

The function outputs to console a message about the number of rows that have incomplete progress.

Value
The output is a data frame of the rows that have incomplete progress. For a function that marks these rows, use `mark_progress()`. For a function that excludes these rows, use `exclude_progress()`.

See Also
Other progress functions: `exclude_progress()`, `mark_progress()`
Other check functions: `check_duplicates()`, `check_duration()`, `check_ip()`, `check_location()`, `check_preview()`, `check_resolution()`

Examples

```r
# Check for rows with incomplete progress
data(qualtrics_text)
check_progress(qualtrics_text)

# Remove preview data first
qualtrics_text %>%
  exclude_preview() %>%
  check_progress()

# Include a lower acceptable completion percentage
qualtrics_numeric %>%
  exclude_preview() %>%
  check_progress(min_progress = 98)

# Do not print rows to console
qualtrics_text %>%
  exclude_preview() %>%
  check_progress(print = FALSE)

# Do not print message to console
qualtrics_text %>%
  exclude_preview() %>%
  check_progress(quiet = TRUE)
```
check_resolution

Check screen resolution

Description

The check_resolution() function subsets rows of data, retaining rows that have unacceptable screen resolution. This can be used, for example, to determine data collected via phones when desktop monitors are required. The function is written to work with data from Qualtrics surveys.

Usage

check_resolution(
    x,  
    res_min = 1000,  
    width_min = 0,  
    height_min = 0,  
    id_col = "ResponseId",  
    res_col = "Resolution",  
    rename = TRUE,  
    keep = FALSE,  
    quiet = FALSE,  
    print = TRUE
)

Arguments

x Data frame (preferably imported from Qualtrics using {qualtrics}).
res_min Minimum acceptable screen resolution (width and height).
width_min Minimum acceptable screen width.
height_min Minimum acceptable screen height.
id_col Column name for unique row ID (e.g., participant).
res_col Column name for screen resolution (in format widthxheight).
rename Logical indicating whether to rename columns (using rename_columns()).
keep Logical indicating whether to keep or remove exclusion column.
quiet Logical indicating whether to print message to console.
print Logical indicating whether to print returned tibble to console.

Details

To record this information in your Qualtrics survey, you must insert a meta info question. Default column names are set based on output from the qualtrics::fetch_survey().

The function outputs to console a message about the number of rows with unacceptable screen resolution.
Value

The output is a data frame of the rows that have unacceptable screen resolutions. This includes new columns for resolution width and height. For a function that marks these rows, use `mark_resolution()`. For a function that excludes these rows, use `exclude_resolution()`.

See Also

Other resolution functions: `exclude_resolution()`, `mark_resolution()`
Other check functions: `check_duplicates()`, `check_duration()`, `check_ip()`, `check_location()`, `check_preview()`, `check_progress()`

Examples

```r
# Check for survey previews
data(qualtrics_text)
check_resolution(qualtrics_text)

# Remove preview data first
qualtrics_text %>%
  exclude_preview() %>%
  check_resolution()

# Do not print rows to console
qualtrics_text %>%
  exclude_preview() %>%
  check_resolution(print = FALSE)

# Do not print message to console
qualtrics_text %>%
  exclude_preview() %>%
  check_resolution(quiet = TRUE)
```

---

**deidentify**

*Remove columns that could include identifiable information*

Description

The `deidentify()` function selects out columns from Qualtrics surveys that may include identifiable information such as IP address, location, or computer characteristics.

Usage

`deidentify(x, strict = TRUE)`

Arguments

- **x**: Data frame (downloaded from Qualtrics).
- **strict**: Logical indicating whether to use strict or non-strict level of deidentification. Strict removes computer information columns in addition to IP address and location.
**Details**

The function offers two levels of deidentification. The default strict level removes columns associated with IP address and location and computer information (browser type and version, operating system, and screen resolution). The non-strict level removes only columns associated with IP address and location.

Typically, deidentification should be used at the end of a processing pipeline so that these columns can be used to exclude rows.

**Value**

An object of the same type as x that excludes Qualtrics columns with identifiable information.

**Examples**

```r
names(qualtrics_numeric)

# Remove IP address, location, and computer information columns
deid <- deidentify(qualtrics_numeric)
names(deid)

# Remove only IP address and location columns
deid2 <- deidentify(qualtrics_numeric, strict = FALSE)
names(deid2)
```

**exclude_duplicates**

Exclude rows with duplicate IP addresses and/or locations

**Description**

The exclude_duplicates() function removes rows of data that have the same IP address and/or same latitude and longitude. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
exclude_duplicates(
x,
id_col = "ResponseId",
ip_col = "IPAddress",
location_col = c("LocationLatitude", "LocationLongitude"),
rename = TRUE,
dupl_ip = TRUE,
dupl_location = TRUE,
include_na = FALSE,
quiet = TRUE,
print = TRUE,
silent = FALSE)
```
Arguments

- **x**: Data frame (preferably imported from Qualtrics using `qualtrics`).
- **id_col**: Column name for unique row ID (e.g., participant).
- **ip_col**: Column name for IP addresses.
- **location_col**: Two element vector specifying columns for latitude and longitude (in that order).
- **rename**: Logical indicating whether to rename columns (using `rename_columns()`).
- **dupl_ip**: Logical indicating whether to check IP addresses.
- **dupl_location**: Logical indicating whether to check latitude and longitude.
- **include_na**: Logical indicating whether to include rows with NAs for IP address and location as potentially excluded rows.
- **quiet**: Logical indicating whether to print message to console.
- **print**: Logical indicating whether to print returned tibble to console.
- **silent**: Logical indicating whether to print message to console. Note this argument controls the exclude message not the check message.

Details

To record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the `qualtrics::fetch_survey()`. By default, IP address and location are both checked, but they can be checked separately with the `dupl_ip` and `dupl_location` arguments.

The function outputs to console separate messages about the number of rows with duplicate IP addresses and rows with duplicate locations. These counts are computed independently, so rows may be counted for both types of duplicates.

Value

An object of the same type as `x` that excludes rows with duplicate IP addresses and/or locations. For a function that just checks for and returns duplicate rows, use `check_duplicates()`. For a function that marks these rows, use `mark_duplicates()`.

See Also

Other duplicates functions: `check_duplicates()`, `mark_duplicates()`

Other exclude functions: `exclude_duration()`, `exclude_ip()`, `exclude_location()`, `exclude_preview()`, `exclude_progress()`, `exclude_resolution()`

Examples

```r
# Exclude duplicate IP addresses and locations
data(qualtrics_text)
df <- exclude_duplicates(qualtrics_text)

# Remove preview data first
```
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_duplicates()

# Exclude only for duplicate locations
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_duplicates(dupl_location = FALSE)

exclude_duration

Exclude rows with minimum or maximum durations

Description

The `exclude_duration()` function removes rows of data that have durations that are too fast or too slow. The function is written to work with data from Qualtrics surveys.

Usage

```
exclude_duration(
  x,
  min_duration = 10,
  max_duration = NULL,
  id_col = "ResponseId",
  duration_col = "Duration (in seconds)",
  rename = TRUE,
  quiet = TRUE,
  print = TRUE,
  silent = FALSE
)
```

Arguments

- **x**: Data frame (preferably imported from Qualtrics using `{qualtrics}`).
- **min_duration**: Minimum duration that is too fast in seconds.
- **max_duration**: Maximum duration that is too slow in seconds.
- **id_col**: Column name for unique row ID (e.g., participant).
- **duration_col**: Column name for durations.
- **rename**: Logical indicating whether to rename columns (using `rename_columns()`).
- **quiet**: Logical indicating whether to print message to console.
- **print**: Logical indicating whether to print returned tibble to console.
- **silent**: Logical indicating whether to print message to console. Note this argument controls the exclude message not the check message.
Details

Default column names are set based on output from the `qualtrics::fetch_survey()`. By default, minimum durations of 10 seconds are checked, but either minima or maxima can be checked with the `min_duration` and `max_duration` arguments. The function outputs to console separate messages about the number of rows that are too fast or too slow.

This function returns the fast and slow rows.

Value

An object of the same type as `x` that excludes rows with fast and/or slow duration. For a function that checks for these rows, use `check_duration()`. For a function that marks these rows, use `mark_duration()`.

See Also

Other duration functions: `check_duration()`, `mark_duration()`

Other exclude functions: `exclude_duplicates()`, `exclude_ip()`, `exclude_location()`, `exclude_preview()`, `exclude_progress()`, `exclude_resolution()`

Examples

```r
# Exclude durations faster than 100 seconds
data(qualtrics_text)
df <- exclude_duration(qualtrics_text, min_duration = 100)

# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_duration()

# Exclude only for durations slower than 800 seconds
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_duration(max_duration = 800)
```

exclude_ip

Exclude IP addresses from outside of a specified country.

Description

The `exclude_ip()` function removes rows of data that have IP addresses from outside the specified country. The function is written to work with data from Qualtrics surveys.
exclude_ip

Usage

exclude_ip(
  x,
  id_col = "ResponseId",
  ip_col = "IPAddress",
  rename = TRUE,
  country = "US",
  include_na = FALSE,
  quiet = TRUE,
  print = TRUE,
  silent = FALSE
)

Arguments

x Data frame or tibble (preferably imported from Qualtrics using {qualtrics}).
id_col Column name for unique row ID (e.g., participant).
ip_col Column name for IP addresses.
rename Logical indicating whether to rename columns (using rename_columns()).
country Two-letter abbreviation of country to check (default is "US").
include_na Logical indicating whether to include rows with NA in IP address column in the output list of potentially excluded data.
quiet Logical indicating whether to print message to console.
print Logical indicating whether to print returned tibble to console.
silent Logical indicating whether to print message to console. Note this argument controls the exclude message not the check message.

Details

To record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the qualtrics::fetch_survey(). The function uses ipaddress::country_networks() to assign IP addresses to specific countries using ISO 3166-1 alpha-2 country codes.

The function outputs to console a message about the number of rows with IP addresses outside of the specified country. If there are NAs for IP addresses (likely due to including preview data—see check_preview()), it will print a message alerting to the number of rows with NAs.

Value

An object of the same type as x that excludes rows with IP addresses outside of the specified country. For a function that checks these rows, use check_ip(). For a function that marks these rows, use mark_ip().
**Note**

This function **requires internet connectivity** as it uses the `ipaddress::country_networks()` function, which pulls daily updated data from [http://www.iwik.org/ipcountry/](http://www.iwik.org/ipcountry/). It only updates the data once per session, as it caches the results for future work during the session.

**See Also**

Other ip functions: `check_ip()`, `mark_ip()`

Other exclude functions: `exclude_duplicates()`, `exclude_duration()`, `exclude_location()`, `exclude_preview()`, `exclude_progress()`, `exclude_resolution()`

**Examples**

```r
# Exclude IP addresses outside of the US
data(qualtrics_text)
df <- exclude_ip(qualtrics_text)

# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_ip()

# Exclude IP addresses outside of Germany
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_ip(country = "DE")
```

---

**exclude_location**  
*Exclude locations outside of US*

**Description**

The `exclude_location()` function removes rows that have locations outside of the US. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
exclude_location(
  x,
  id_col = "ResponseId",
  location_col = c("LocationLatitude", "LocationLongitude"),
  rename = TRUE,
  include_na = FALSE,
  quiet = TRUE,
  print = TRUE,
  silent = FALSE
)
```
Arguments

- **x**: Data frame (preferably imported from Qualtrics using `qualtrics`).
- **id_col**: Column name for unique row ID (e.g., participant).
- **location_col**: Two element vector specifying columns for latitude and longitude (in that order).
- **rename**: Logical indicating whether to rename columns (using `rename_columns()`).
- **include_na**: Logical indicating whether to include rows with NA in latitude and longitude columns in the output list of potentially excluded data.
- **quiet**: Logical indicating whether to print message to console.
- **print**: Logical indicating whether to print returned tibble to console.
- **silent**: Logical indicating whether to print message to console. Note this argument controls the exclude message not the check message.

Details

To record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the `qualtrics::fetch_survey()`. The function only works for the United States. It uses the `maps::map.where()` to determine if latitude and longitude are inside the US.

The function outputs to console a message about the number of rows with locations outside of the US.

Value

An object of the same type as `x` that excludes rows that are located outside of the US and (if `include_na == FALSE`) rows with no location information. For a function that checks for these rows, use `check_location()`. For a function that marks these rows, use `mark_location()`.

See Also

Other location functions: `check_location()`, `mark_location()`

Other exclude functions: `exclude_duplicates()`, `exclude_duration()`, `exclude_ip()`, `exclude_preview()`, `exclude_progress()`, `exclude_resolution()`

Examples

```r
# Exclude locations outside of the US
data(qualtrics_text)
df <- exclude_location(qualtrics_text)

# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_location()
```
exclude_preview

Description

The `exclude_preview()` function removes rows that are survey previews. The function is written to work with data from Qualtrics surveys.

Usage

```
exclude_preview(
  x,
  id_col = "ResponseId",
  preview_col = "Status",
  rename = TRUE,
  quiet = TRUE,
  print = TRUE,
  silent = FALSE
)
```

Arguments

- `x` Data frame (preferably imported from Qualtrics using `{qualtrics}`).
- `id_col` Column name for unique row ID (e.g., participant).
- `preview_col` Column name for survey preview.
- `rename` Logical indicating whether to rename columns (using `rename_columns()`).
- `quiet` Logical indicating whether to print message to console.
- `print` Logical indicating whether to print returned tibble to console.
- `silent` Logical indicating whether to print message to console. Note this argument controls the exclude message not the check message.

Details

Default column names are set based on output from the `qualtrics::fetch_survey()`. The preview column in Qualtrics can be a numeric or character vector depending on whether it is exported as choice text or numeric values. This function works for both.

The function outputs to console a message about the number of rows that are survey previews.

Value

An object of the same type as `x` that excludes rows that are survey previews. For a function that checks for these rows, use `check_preview()`. For a function that marks these rows, use `mark_preview()`.
exclude_progress

See Also

Other preview functions: `check_preview()`, `mark_preview()`

Other exclude functions: `exclude_duplicates()`, `exclude_duration()`, `exclude_ip()`, `exclude_location()`, `exclude_progress()`, `exclude_resolution()`

Examples

```r
# Exclude survey previews
data(qualtrics_text)
df <- exclude_preview(qualtrics_text)

# Works for Qualtrics data exported as numeric values, too
df <- qualtrics_numeric %>%
  exclude_preview()

# Do not print rows to console
df <- qualtrics_text %>%
  exclude_preview(print = FALSE)
```

---

### exclude_progress

**Exclude survey progress**

**Description**

The `exclude_progress()` function removes rows that have incomplete progress. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
exclude_progress(
  x,
  min_progress = 100,
  id_col = "ResponseId",
  finished_col = "Finished",
  progress_col = "Progress",
  rename = TRUE,
  quiet = TRUE,
  print = TRUE,
  silent = FALSE
)
```

**Arguments**

- `x`: Data frame (preferably imported from Qualtrics using `{qualtrics}`).
- `min_progress`: Amount of progress considered acceptable to include.
- `id_col`: Column name for unique row ID (e.g., participant).
- `finished_col`: Column name for whether survey was completed.
exclude_progress

progress_col  Column name for percentage of survey completed.
rename        Logical indicating whether to rename columns (using rename_columns())
quiet         Logical indicating whether to print message to console.
print         Logical indicating whether to print returned tibble to console.
silent        Logical indicating whether to print message to console. Note this argument controls the exclude message not the check message.

Details
Default column names are set based on output from the qualtrics::fetch_survey(). The default requires 100% completion, but lower levels of completion maybe acceptable and can be allowed by specifying the min_progress argument. The finished column in Qualtrics can be a numeric or character vector depending on whether it is exported as choice text or numeric values. This function works for both.

The function outputs to console a message about the number of rows that have incomplete progress.

Value
An object of the same type as x that excludes rows that have incomplete progress. For a function that checks for these rows, use check_progress(). For a function that marks these rows, use mark_progress().

See Also
Other progress functions: check_progress(), mark_progress()
Other exclude functions: exclude_duplicates(), exclude_duration(), exclude_ip(), exclude_location(), exclude_preview(), exclude_resolution()

Examples
# Exclude rows with incomplete progress
data(qualtrics_text)
df <- exclude_progress(qualtrics_text)

# Remove preview data first
df <- qualtrics_text %>%
     exclude_preview() %>%
     exclude_progress()

# Include a lower acceptable completion percentage
df <- qualtrics_numeric %>%
     exclude_preview() %>%
     exclude_progress(min_progress = 98)

# Do not print rows to console
df <- qualtrics_text %>%
     exclude_preview() %>%
     exclude_progress(print = FALSE)
**Description**

The `exclude_resolution()` function removes rows that have unacceptable screen resolution. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
exclude_resolution(
  x,
  res_min = 1000,
  width_min = 0,
  height_min = 0,
  id_col = "ResponseId",
  res_col = "Resolution",
  rename = TRUE,
  quiet = TRUE,
  print = TRUE,
  silent = FALSE
)
```

**Arguments**

- `x` Data frame (preferably imported from Qualtrics using `{qualtrics}`).
- `res_min` Minimum acceptable screen resolution (width and height).
- `width_min` Minimum acceptable screen width.
- `height_min` Minimum acceptable screen height.
- `id_col` Column name for unique row ID (e.g., participant).
- `res_col` Column name for screen resolution (in format widthxheight).
- `rename` Logical indicating whether to rename columns (using `rename_columns()`)
- `quiet` Logical indicating whether to print message to console.
- `print` Logical indicating whether to print returned tibble to console.
- `silent` Logical indicating whether to print message to console. Note this argument controls the exclude message not the check message.

**Details**

To record this information in your Qualtrics survey, you must insert a meta info question. Default column names are set based on output from the `qualtrics::fetch_survey()`.

The function outputs to console a message about the number of rows with unacceptable screen resolution.
**Value**

An object of the same type as `x` that excludes rows that have unacceptable screen resolutions. For a function that checks for these rows, use `check_resolution()`. For a function that marks these rows, use `mark_resolution()`.

**See Also**

Other resolution functions: `check_resolution()`, `mark_resolution()`

Other exclude functions: `exclude_duplicates()`, `exclude_duration()`, `exclude_ip()`, `exclude_location()`, `exclude_preview()`, `exclude_progress()`

**Examples**

```r
# Exclude low screen resolutions
data(qualtrics_text)
df <- exclude_resolution(qualtrics_text)

# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  exclude_resolution()
```

---

**mark_duplicates**

*Mark duplicate IP addresses and/or locations*

**Description**

The `mark_duplicates()` function creates a column labeling rows of data that have the same IP address and/or same latitude and longitude. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
mark_duplicates(
  x,
  id_col = "ResponseId",
  ip_col = "IPAddress",
  location_col = c("LocationLatitude", "LocationLongitude"),
  rename = TRUE,
  dupl_ip = TRUE,
  dupl_location = TRUE,
  include_na = FALSE,
  quiet = FALSE,
  print = TRUE
)
```
Arguments

- `x`: Data frame (preferably imported from Qualtrics using `qualtrics`).
- `id_col`: Column name for unique row ID (e.g., participant).
- `ip_col`: Column name for IP addresses.
- `location_col`: Two element vector specifying columns for latitude and longitude (in that order).
- `rename`: Logical indicating whether to rename columns (using `rename_columns()`).
- `dupl_ip`: Logical indicating whether to check IP addresses.
- `dupl_location`: Logical indicating whether to check latitude and longitude.
- `include_na`: Logical indicating whether to include rows with NAs for IP address and location as potentially excluded rows.
- `quiet`: Logical indicating whether to print message to console.
- `print`: Logical indicating whether to print returned tibble to console.

Details

To record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the `qualtrics::fetch_survey()`. By default, IP address and location are both checked, but they can be checked separately with the `dupl_ip` and `dupl_location` arguments.

The function outputs to console separate messages about the number of rows with duplicate IP addresses and rows with duplicate locations. These counts are computed independently, so rows may be counted for both types of duplicates.

Value

An object of the same type as `x` that includes a column marking rows with duplicate IP addresses and/or locations. For a function that just checks for and returns duplicate rows, use `check_duplicates()`. For a function that excludes these rows, use `exclude_duplicates()`.

See Also

Other duplicates functions: `check_duplicates()`, `exclude_duplicates()`

Other mark functions: `mark_duration()`, `mark_ip()`, `mark_location()`, `mark_preview()`, `mark_progress()`, `mark_resolution()`

Examples

```r
# Mark duplicate IP addresses and locations
data(qualtrics_text)
df <- mark_duplicates(qualtrics_text)

# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_duplicates()
```
# Mark only for duplicate locations

df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_duplicates(dupl_location = FALSE)

mark_duration

## Mark minimum or maximum durations

**Description**

The `mark_duration()` function creates a column labeling rows with fast and/or slow duration. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
mark_duration(
  x,
  min_duration = 10,
  max_duration = NULL,
  id_col = "ResponseId",
  duration_col = "Duration (in seconds)",
  rename = TRUE,
  quiet = FALSE,
  print = TRUE
)
```

**Arguments**

- `x` Data frame (preferably imported from Qualtrics using `qualRics`).
- `min_duration` Minimum duration that is too fast in seconds.
- `max_duration` Maximum duration that is too slow in seconds.
- `id_col` Column name for unique row ID (e.g., participant).
- `duration_col` Column name for durations.
- `rename` Logical indicating whether to rename columns (using `rename_columns()`).
- `quiet` Logical indicating whether to print message to console.
- `print` Logical indicating whether to print returned tibble to console.

**Details**

Default column names are set based on output from the `qualRics::fetch_survey()`. By default, minimum durations of 10 seconds are checked, but either minima or maxima can be checked with the `min_duration` and `max_duration` arguments. The function outputs to console separate messages about the number of rows that are too fast or too slow.

This function returns the fast and slow rows.
Value

An object of the same type as \( x \) that includes a column marking rows with fast and slow duration. For a function that checks for these rows, use `check_duration()`. For a function that excludes these rows, use `exclude_duration()`.

See Also

Other duration functions: `check_duration()`, `exclude_duration()`

Other mark functions: `mark_duplicates()`, `mark_ip()`, `mark_location()`, `mark_preview()`, `mark_progress()`, `mark_resolution()`

Examples

```r
# Mark durations faster than 100 seconds
data(qualtrics_text)
df <- mark_duration(qualtrics_text, min_duration = 100)

# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_duration()

# Mark only for durations slower than 800 seconds
df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_duration(max_duration = 800)
```

mark_ip

Mark IP addresses from outside of a specified country.

Description

The `mark_ip()` function creates a column labeling rows of data that have IP addresses from outside the specified country. The function is written to work with data from Qualtrics surveys.

Usage

```r
mark_ip(
  x,
  id_col = "ResponseId",
  ip_col = "IPAddress",
  rename = TRUE,
  country = "US",
  include_na = FALSE,
  quiet = FALSE,
  print = TRUE
)
```
Arguments

- `x`: Data frame or tibble (preferably imported from Qualtrics using `qualtRics`).
- `id_col`: Column name for unique row ID (e.g., participant).
- `ip_col`: Column name for IP addresses.
- `rename`: Logical indicating whether to rename columns (using `rename_columns()`).
- `country`: Two-letter abbreviation of country to check (default is "US").
- `include_na`: Logical indicating whether to include rows with NA in IP address column in the output list of potentially excluded data.
- `quiet`: Logical indicating whether to print message to console.
- `print`: Logical indicating whether to print returned tibble to console.

Details

To record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the `qualtRics::fetch_survey()`. The function uses `ipaddress::country_networks()` to assign IP addresses to specific countries using ISO 3166-1 alpha-2 country codes.

The function outputs to console a message about the number of rows with IP addresses outside of the specified country. If there are NAs for IP addresses (likely due to including preview data—see `check_preview()`), it will print a message alerting to the number of rows with NAs.

Value

An object of the same type as `x` that includes a column marking rows with IP addresses outside of the specified country. For a function that checks these rows, use `check_ip()`. For a function that excludes these rows, use `exclude_ip()`.

Note

This function requires internet connectivity as it uses the `ipaddress::country_networks()` function, which pulls daily updated data from `https://www.iwik.org/ipcountry/`. It only updates the data once per session, as it caches the results for future work during the session.

See Also

Other ip functions: `check_ip()`, `exclude_ip()`
Other mark functions: `mark_duplicates()`, `mark_duration()`, `mark_location()`, `mark_preview()`, `mark_progress()`, `mark_resolution()`

Examples

```r
# Mark IP addresses outside of the US
data(qualtrics_text)
df <- mark_ip(qualtrics_text)
```
# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_ip()

# Mark IP addresses outside of Germany
df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_ip(country = "DE")

---

**mark_location**  
*Mark locations outside of US*

**Description**

The `mark_location()` function creates a column labeling rows that have locations outside of the US. The function is written to work with data from Qualtrics surveys.

**Usage**

```r
mark_location(
  x,
  id_col = "ResponseId",
  location_col = c("LocationLatitude", "LocationLongitude"),
  rename = TRUE,
  include_na = FALSE,
  quiet = FALSE,
  print = TRUE
)
```

**Arguments**

- `x`  
  Data frame (preferably imported from Qualtrics using `{qualtrics}`).
- `id_col`  
  Column name for unique row ID (e.g., participant).
- `location_col`  
  Two element vector specifying columns for latitude and longitude (in that order).
- `rename`  
  Logical indicating whether to rename columns (using `rename_columns()`).
- `include_na`  
  Logical indicating whether to include rows with NA in latitude and longitude columns in the output list of potentially excluded data.
- `quiet`  
  Logical indicating whether to print message to console.
- `print`  
  Logical indicating whether to print returned tibble to console.
To record this information in your Qualtrics survey, you must ensure that Anonymize responses is disabled.

Default column names are set based on output from the `qualtrics::fetch_survey()`. The function only works for the United States. It uses the `maps::map.where()` to determine if latitude and longitude are inside the US.

The function outputs to console a message about the number of rows with locations outside of the US.

An object of the same type as `x` that includes a column marking rows that are located outside of the US and (if `include_na == FALSE`) rows with no location information. For a function that checks for these rows, use `check_location()`. For a function that excludes these rows, use `exclude_location()`.

Other location functions: `check_location()`, `exclude_location()`

Other mark functions: `mark_duplicates()`, `mark_duration()`, `mark_ip()`, `mark_preview()`, `mark_progress()`, `mark_resolution()`

# Mark locations outside of the US
```
data(qualtrics_text)
df <- mark_location(qualtrics_text)
```

# Remove preview data first
```
df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_location()
```

To mark survey previews, use the `mark_preview()` function.

The `mark_preview()` function creates a column labeling rows that are survey previews. The function is written to work with data from Qualtrics surveys.
Usage

mark_preview(
  x,
  id_col = "ResponseId",
  preview_col = "Status",
  rename = TRUE,
  quiet = FALSE,
  print = TRUE
)

Arguments

x  Data frame (preferably imported from Qualtrics using \{qualtRics\}).
id_col Column name for unique row ID (e.g., participant).
preview_col Column name for survey preview.
rename Logical indicating whether to rename columns (using rename_columns()).
quiet Logical indicating whether to print message to console.
print Logical indicating whether to print returned tibble to console.

Details

Default column names are set based on output from the \texttt{qualtRics::fetch_survey()}. The preview column in Qualtrics can be a numeric or character vector depending on whether it is exported as choice text or numeric values. This function works for both.

The function outputs to console a message about the number of rows that are survey previews.

Value

An object of the same type as \texttt{x} that includes a column marking rows that are survey previews. For a function that checks for these rows, use \texttt{check_preview()}. For a function that excludes these rows, use \texttt{exclude_preview()}.  

See Also

Other preview functions: \texttt{check_preview()}, \texttt{exclude_preview()}
Other mark functions: \texttt{mark_duplicates()}, \texttt{mark_duration()}, \texttt{mark_ip()}, \texttt{mark_location()}, \texttt{mark_progress()}, \texttt{mark_resolution()}

Examples

# Mark survey previews
data(qualtrics_text)
df <- mark_preview(qualtrics_text)

# Works for Qualtrics data exported as numeric values, too
df <- qualtrics_numeric %>%
  mark_preview()
mark_progress

Mark survey progress

Description

The `mark_progress()` function creates a column labeling rows that have incomplete progress. The function is written to work with data from Qualtrics surveys.

Usage

```r
mark_progress(
  x,
  min_progress = 100,
  id_col = "ResponseId",
  finished_col = "Finished",
  progress_col = "Progress",
  rename = TRUE,
  quiet = FALSE,
  print = TRUE
)
```

Arguments

- `x` Data frame (preferably imported from Qualtrics using `{qualtrics}`).
- `min_progress` Amount of progress considered acceptable to include.
- `id_col` Column name for unique row ID (e.g., participant).
- `finished_col` Column name for whether survey was completed.
- `progress_col` Column name for percentage of survey completed.
- `rename` Logical indicating whether to rename columns (using `rename_columns()`)
- `quiet` Logical indicating whether to print message to console.
- `print` Logical indicating whether to print returned tibble to console.

Details

Default column names are set based on output from the `qualtrics::fetch_survey()`. The default requires 100% completion, but lower levels of completion maybe acceptable and can be allowed by specifying the `min_progress` argument. The finished column in Qualtrics can be a numeric or character vector depending on whether it is exported as choice text or numeric values. This function works for both.

The function outputs to console a message about the number of rows that have incomplete progress.

Value

An object of the same type as `x` that includes a column marking rows that have incomplete progress. For a function that checks for these rows, use `check_progress()`. For a function that excludes these rows, use `exclude_progress()`.
See Also

Other progress functions: `check_progress()`, `exclude_progress()`
Other mark functions: `mark_duplicates()`, `mark_duration()`, `mark_ip()`, `mark_location()`, `mark_preview()`, `mark_resolution()`

Examples

```r
# Mark rows with incomplete progress
data(qualtrics_text)
df <- mark_progress(qualtrics_text)

# Remove preview data first
df <- qualtrics_text %>%
    exclude_preview() %>%
    mark_progress()

# Include a lower acceptable completion percentage
df <- qualtrics_numeric %>%
    exclude_preview() %>%
    mark_progress(min_progress = 98)
```

---

**mark_resolution**  
Mark unacceptable screen resolution

Description

The `mark_resolution()` function creates a column labeling rows that have unacceptable screen resolution. The function is written to work with data from Qualtrics surveys.

Usage

```r
mark_resolution(
    x,
    res_min = 1000,
    width_min = 0,
    height_min = 0,
    id_col = "ResponseId",
    res_col = "Resolution",
    rename = TRUE,
    quiet = FALSE,
    print = TRUE
)
```

Arguments

- **x**  
  Data frame (preferably imported from Qualtrics using `qualtrics()`).
- **res_min**  
  Minimum acceptable screen resolution (width and height).
mark_resolution

width_min  Minimum acceptable screen width.
height_min Minimum acceptable screen height.
id_col     Column name for unique row ID (e.g., participant).
res_col    Column name for screen resolution (in format widthxheight).
rename     Logical indicating whether to rename columns (using rename_columns()).
quiet      Logical indicating whether to print message to console.
print      Logical indicating whether to print returned tibble to console.

Details

To record this information in your Qualtrics survey, you must insert a meta info question. Default column names are set based on output from the qualTRics::fetch_survey(). The function outputs to console a message about the number of rows with unacceptable screen resolution.

Value

An object of the same type as x that includes a column marking rows that have unacceptable screen resolutions. For a function that checks for these rows, use check_resolution(). For a function that excludes these rows, use exclude_resolution().

See Also

Other resolution functions: check_resolution(), exclude_resolution()
Other mark functions: mark_duplicates(), mark_duration(), mark_ip(), mark_location(), mark_preview(), mark_progress()

Examples

# Mark low screen resolutions
data(qualtrics_text)
df <- mark_resolution(qualtrics_text)

# Remove preview data first
df <- qualtrics_text %>%
  exclude_preview() %>%
  mark_resolution()
Example numeric metadata imported with `qualtrics::fetch_survey()` from simulated Qualtrics study

Description

A dataset containing the metadata from a standard Qualtrics survey with browser metadata collected and exported with "Use numeric values". The data were imported using `qualtrics::fetch_survey()`. These data were randomly generated using `iptools::ip_random()` and `rgeolocate::ip2location()` functions.

Usage

`qualtrics_fetch`

Format

A data frame with 100 rows and 17 variables:

- `StartDate` date and time data collection started, in ISO 8601 format
- `EndDate` date and time data collection ended, in ISO 8601 format
- `Status` numeric flag for preview (1) vs. implemented survey (0) entries
- `IPAddress` participant IP address (truncated for anonymity)
- `Progress` percentage of survey completed
- `Duration (in seconds)` duration of time required to complete survey, in seconds
- `Finished` numeric flag for whether survey was completed (1) or progress was < 100 (0)
- `RecordedDate` date and time survey was recorded, in ISO 8601 format
- `ResponseId` random ID for participants
- `LocationLatitude` latitude geolocated from IP address
- `LocationLongitude` longitude geolocated from IP address
- `UserLanguage` language set in Qualtrics
- `Q1_Browser` user web browser type
- `Q1_Version` user web browser version
- `Q1_Operating System` user operating system
- `Q1_Resolution` user screen resolution
- `Q2` response to question about whether the user liked the survey (1 = Yes, 0 = No)

See Also

Other data: `qualtrics_fetch2`, `qualtrics_numeric`, `qualtrics_raw`, `qualtrics_text`
qualtrics_fetch2  Example numeric metadata imported with qualtrics::fetch_survey() from simulated Qualtrics study but with labels included as column names

Description

A dataset containing the metadata from a standard Qualtrics survey with browser metadata collected and exported with "Use numeric values". The data were imported using qualtrics::fetch_survey(), and then the secondary labels were assigned as column names with sjlabelled::get_label(). These data were randomly generated using iptools::ip_random() and rgeolocate::ip2location() functions.

Usage

qualtrics_fetch2

Format

A data frame with 100 rows and 17 variables:

Start Date  date and time data collection started, in ISO 8601 format
End Date  date and time data collection ended, in ISO 8601 format
Response Type  numeric flag for preview (1) vs. implemented survey (0) entries
IP Address  participant IP address (truncated for anonymity)
Progress  percentage of survey completed
Duration (in seconds)  duration of time required to complete survey, in seconds
Finished  numeric flag for whether survey was completed (1) or progress was < 100 (0)
Recorded Date  date and time survey was recorded, in ISO 8601 format
Response ID  random ID for participants
Location Latitude  latitude geolocated from IP address
Location Longitude  longitude geolocated from IP address
User Language  language set in Qualtrics
Click to write the question text - Browser  user web browser type
Click to write the question text - Version  user web browser version
Click to write the question text - Operating System  user operating system
Click to write the question text - Resolution  user screen resolution
like  response to question about whether the user liked the survey (1 = Yes, 0 = No)

See Also

Other data: qualtrics_fetch, qualtrics_numeric, qualtrics_raw, qualtrics_text
Example numeric metadata from simulated Qualtrics study

Description

A dataset containing the metadata from a standard Qualtrics survey with browser metadata collected and exported with "Use numeric values". These data were randomly generated using ip-tools::ip_random() and rgeolocate::ip2location() functions.

Usage

qualtrics_numeric

Format

A data frame with 100 rows and 16 variables:

- **StartDate**: date and time data collection started, in ISO 8601 format
- **EndDate**: date and time data collection ended, in ISO 8601 format
- **Status**: numeric flag for preview (1) vs. implemented survey (0) entries
- **IPAddress**: participant IP address (truncated for anonymity)
- **Progress**: percentage of survey completed
- **Duration (in seconds)**: duration of time required to complete survey, in seconds
- **Finished**: numeric flag for whether survey was completed (1) or progress was < 100 (0)
- **RecordedDate**: date and time survey was recorded, in ISO 8601 format
- **ResponseId**: random ID for participants
- **LocationLatitude**: latitude geolocated from IP address
- **LocationLongitude**: longitude geolocated from IP address
- **UserLanguage**: language set in Qualtrics
- **Browser**: user web browser type
- **Version**: user web browser version
- **Operating System**: user operating system
- **Resolution**: user screen resolution

See Also

Other data: qualtrics_fetch2, qualtrics_fetch, qualtrics_raw, qualtrics_text
A dataset containing the metadata from a standard Qualtrics survey with browser metadata collected and exported with "Use choice text". These data were randomly generated using `iptools::ip_random()` and `rgeolocate::ip2location()` functions. This dataset includes the two header rows of with column information that is exported by Qualtrics.

Usage

```r
qualtrics_raw
```

Format

A data frame with 102 rows and 16 variables:

- **StartDate**: date and time data collection started, in ISO 8601 format
- **EndDate**: date and time data collection ended, in ISO 8601 format
- **Status**: flag for preview (Survey Preview) vs. implemented survey (IP Address) entries
- **IPAddress**: participant IP address (truncated for anonymity)
- **Progress**: percentage of survey completed
- **Duration (in seconds)**: duration of time required to complete survey, in seconds
- **Finished**: logical for whether survey was completed (TRUE) or progress was < 100 (FALSE)
- **RecordedDate**: date and time survey was recorded, in ISO 8601 format
- **ResponseId**: random ID for participants
- **LocationLatitude**: latitude geolocated from IP address
- **LocationLongitude**: longitude geolocated from IP address
- **UserLanguage**: language set in Qualtrics
- **Browser**: user web browser type
- **Version**: user web browser version
- **Operating System**: user operating system
- **Resolution**: user screen resolution

See Also

Other data: `qualtrics_fetch2`, `qualtrics_fetch`, `qualtrics_numeric`, `qualtrics_text`
qualtrics_text

Example text-based metadata from simulated Qualtrics study

Description

A dataset containing the metadata from a standard Qualtrics survey with browser metadata collected and exported with "Use choice text". These data were randomly generated using iptools::ip_random() and rgeolocate::ip2location() functions.

Usage

qualtrics_text

Format

A data frame with 100 rows and 16 variables:

- **StartDate**: date and time data collection started, in ISO 8601 format
- **EndDate**: date and time data collection ended, in ISO 8601 format
- **Status**: flag for preview (Survey Preview) vs. implemented survey (IP Address) entries
- **IPAddress**: participant IP address (truncated for anonymity)
- **Progress**: percentage of survey completed
- **Duration (in seconds)**: duration of time required to complete survey, in seconds
- **Finished**: logical for whether survey was completed (TRUE) or progress was < 100 (FALSE)
- **RecordedDate**: date and time survey was recorded, in ISO 8601 format
- **ResponseId**: random ID for participants
- **LocationLatitude**: latitude geolocated from IP address
- **LocationLongitude**: longitude geolocated from IP address
- **UserLanguage**: language set in Qualtrics
- **Browser**: user web browser type
- **Version**: user web browser version
- **Operating System**: user operating system
- **Resolution**: user screen resolution

See Also

Other data: qualtrics_fetch2, qualtrics_fetch, qualtrics_numeric, qualtrics_raw
remove_label_rows \hspace{1cm} \textit{Remove two initial rows created in Qualtrics data}

\textbf{Description}

The \texttt{remove_label_rows()} function filters out the initial label rows from datasets downloaded from \texttt{Qualtrics} surveys.

\textbf{Usage}

\begin{verbatim}
remove_label_rows(x, convert = TRUE, rename = FALSE)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{1cm} Data frame (downloaded from Qualtrics).
  \item \texttt{convert} \hspace{1cm} Logical indicating whether to convert/coerce date, logical and numeric columns from the metadata.
  \item \texttt{rename} \hspace{1cm} Logical indicating whether to rename columns based on first row of data.
\end{itemize}

\textbf{Details}

The function (1) checks if the data set uses Qualtrics column names, (2) checks if label rows are already used as column names, (3) removes label rows if present, and (4) converts date, logical, and numeric metadata columns to proper data type. Datasets imported using \texttt{qualRics::fetch_survey()} should not need this function.

The \texttt{convert} argument only converts the \texttt{StartDate}, \texttt{EndDate}, \texttt{RecordedDate}, \texttt{Progress}, \texttt{Finished}, \texttt{Duration (in seconds)}, \texttt{LocationLatitude}, and \texttt{LocationLongitude} columns. To convert other data columns, see \texttt{dplyr::mutate()}.

\textbf{Value}

An object of the same type as \texttt{x} that excludes Qualtrics label rows and with date, logical, and numeric metadata columns converted to the correct data class.

\textbf{Examples}

\begin{verbatim}
# Remove label rows
data(qualtrics_raw)
df <- remove_label_rows(qualtrics_raw)
\end{verbatim}
rename_columns

Rename columns to match standard Qualtrics names

Description

The rename_columns() function renames the metadata columns to match standard Qualtrics names.

Usage

rename_columns(x, alert = TRUE)

Arguments

x 
Data frame (preferably imported from Qualtrics using {qualtrics}).

alert 
Logical indicating whether to alert user to the fact that the columns do not match the secondary labels and therefore cannot be renamed.

Details

When importing Qualtrics data using qualtrics::fetch_survey(). labels entered in Qualtrics questions are saved as 'subtitles' for column names. Using sjlabelled::get_label() can make these secondary labels be the primary column names. However, this results in a different set of names for the metadata columns than is used in all of the mark_(), check_(), and exclude_() functions. This function renames these columns to match the standard Qualtrics names.

Value

An object of the same type as x that has column names that match standard Qualtrics names.

See Also

Other column name functions: use_labels()

Examples

# Rename columns
data(qualtrics_fetch)
qualtrics_renamed <- qualtrics_fetch %>%
  rename_columns()
names(qualtrics_fetch)
names(qualtrics_renamed)

# Alerts when columns cannot be renamed
data(qualtrics_numeric)
rename_columns(qualtrics_numeric)

# Turn off alert
rename_columns(qualtrics_numeric, alert = FALSE)
unite_exclusions  

Unite multiple exclusion columns into single column

Description

Each of the mark_*( ) functions appends a new column to the data. The unite_exclusions() function unites all of those columns in a single column that can be used to filter any or all exclusions downstream. Rows with multiple exclusions are concatenated with commas.

Usage

unite_exclusions(  
  x,  
  exclusion_types = c( "duplicates", "duration", "ip", "location", "preview", "progress",  
                       "resolution" ),  
  separator = ",",  
  remove = TRUE  
)

Arguments

x  
Data frame or tibble (preferably exported from Qualtrics).

exclusion_types  
Vector of types of exclusions to unite.

separator  
Character string specifying what character to use to separate multiple exclusion types

remove  
Logical specifying whether to remove united columns (default = TRUE) or leave them in the data frame (FALSE)

Value

An object of the same type as x that includes the all of the same rows but with a single exclusion column replacing all of the specified exclusion_* columns.

Examples

# Unite all exclusion types
df <- qualtrics_text %>%  
  mark_duplicates() %>%  
  mark_duration(min_duration = 100) %>%  
  mark_ip() %>%  
  mark_location() %>%  
  mark_preview() %>%  
  mark_progress() %>%  
  mark_resolution()

df2 <- df %>%
use_labels

unite_exclusions()

# Unite subset of exclusion types
df2 <- df %>%
  unite_exclusions(exclusion_types = c("duplicates", "duration", "ip"))

use_labels
Use Qualtrics labels as column names

Description
The `use_labels()` function renames the columns using the labels generated in Qualtrics. Data must be imported using `qualtRics::fetch_survey()`.

Usage
`use_labels(x)`

Arguments

  x Data frame imported using `qualtRics::fetch_survey()`.

Value
An object of the same type as `x` that has column names using the labels generated in Qualtrics.

See Also
Other column name functions: `rename_columns()`

Examples

# Rename columns
data(qualtrics_fetch)
qualtrics_renamed <- qualtrics_fetch %>%
  use_labels()
names(qualtrics_fetch)
names(qualtrics_renamed)
Index

* check functions
  check_duplicates, 2
  check_duration, 4
  check_ip, 6
  check_location, 8
  check_preview, 9
  check_progress, 11
  check_resolution, 13

* column name functions
  rename_columns, 43
  use_labels, 45

* datasets
  qualtrics_fetch, 37
  qualtrics_fetch2, 38
  qualtrics_numeric, 39
  qualtrics_raw, 40
  qualtrics_text, 41

* data
  qualtrics_fetch, 37
  qualtrics_fetch2, 38
  qualtrics_numeric, 39
  qualtrics_raw, 40
  qualtrics_text, 41

* duplicates functions
  check_duplicates, 2
  exclude_duplicates, 15
  mark_duplicates, 26

* duration functions
  check_duration, 4
  exclude_duration, 17
  mark_duration, 28

* exclude functions
  exclude_duplicates, 15
  exclude_duration, 17
  exclude_ip, 18
  exclude_location, 20
  exclude_preview, 22
  exclude_progress, 23
  exclude_resolution, 25

* helper
  deidentify, 14
  remove_label_rows, 42
  rename_columns, 43
  unite_exclusions, 44
  use_labels, 45

* ip functions
  check_ip, 6
  exclude_ip, 18
  mark_ip, 29

* location functions
  check_location, 8
  exclude_location, 20
  mark_location, 31

* mark functions
  mark_duplicates, 26
  mark_duration, 28
  mark_ip, 29
  mark_location, 31
  mark_preview, 32
  mark_progress, 34
  mark_resolution, 35

* preview functions
  check_preview, 9
  exclude_preview, 22
  mark_preview, 32

* progress functions
  check_progress, 11
  exclude_progress, 23
  mark_progress, 34

* resolution functions
  check_resolution, 13
  exclude_resolution, 25
  mark_resolution, 35

check_duplicates, 2, 5, 7, 9, 10, 12, 14, 16, 27
check_duplicates(), 16, 27
check_duration, 4, 4, 7, 9, 10, 12, 14, 18, 29
check_duration(), 18, 29
INDEX

check_ip, 4, 5, 6, 9, 10, 12, 14, 20, 30
check_ip(), 19, 30
check_location, 4, 5, 7, 8, 10, 12, 14, 21, 32
check_location(), 21, 32
check_preview, 4, 5, 7, 9, 12, 14, 23, 33
check_preview(), 7, 19, 22, 30, 33
check_progress, 4, 5, 7, 9, 10, 11, 14, 24, 35
check_progress(), 24, 34
check_resolution, 4, 5, 7, 9, 10, 12, 13, 26, 36
check_resolution(), 26, 36
deidentify, 14
dplyr::mutate(), 42
exclude_duplicates, 4, 15, 18, 20, 21, 23, 24, 26, 27
exclude_duplicates(), 4, 27
exclude_duration, 5, 16, 17, 20, 21, 23, 24, 26, 29
exclude_duration(), 5, 29
exclude_ip, 7, 16, 18, 21, 23, 24, 26, 30
exclude_ip(), 7, 30
exclude_location, 9, 16, 18, 20, 23, 24, 26, 32
exclude_location(), 9, 32
exclude_preview, 10, 16, 18, 20, 21, 22, 24, 26, 33
exclude_preview(), 10, 33
exclude_progress, 12, 16, 18, 20, 21, 23, 23, 26, 35
exclude_progress(), 12, 34
exclude_resolution, 14, 16, 18, 20, 21, 23, 24, 25, 36
exclude_resolution(), 14, 36
ipaddress::country_networks(), 7, 19, 20, 30
maps::map.where(), 9, 21, 32
mark_duplicates, 4, 16, 26, 29, 30, 32, 33, 35, 36
mark_duplicates(), 4, 16
mark_duration, 5, 18, 27, 28, 30, 32, 33, 35, 36
mark_duration(), 5, 18
mark_ip, 7, 20, 27, 29, 32, 33, 35, 36
mark_ip(), 7, 19
mark_location, 9, 21, 27, 29, 30, 31, 33, 35, 36
mark_location(), 9, 21
mark_preview, 10, 23, 27, 29, 30, 32, 32, 35, 36
mark_preview(), 10, 22
mark_progress, 12, 24, 27, 29, 30, 32, 33, 34, 36
mark_progress(), 12, 24
mark_resolution, 14, 26, 27, 29, 30, 32, 33, 35, 35
mark_resolution(), 14, 26
qualtrics_fetch, 37, 38–41
qualtrics_fetch2, 37, 38, 39–41
qualtrics_numeric, 37, 38, 39, 40, 41
qualtrics_raw, 37–39, 40, 41
qualtrics_text, 37–40, 41
remove_label_rows, 42
rename_columns, 43, 45
rename_columns(), 3, 5, 6, 8, 10, 12, 13, 16, 17, 19, 22, 24, 25, 27, 28, 30, 31, 33, 34, 36
unite_exclusions, 44
use_labels, 43, 45