Package ‘intendo’

February 22, 2023

Title  A Group of Fun Datasets of Various Sizes and Differing Levels of Quality

Version  0.1

Description  Four datasets are provided here from the 'Intendo' game 'Super Jetroid'. It is data from the 2015 year of operation and it comprises a revenue table ('all_revenue'), a daily users table ('users_daily'), a user summary table ('user_summary'), and a table with data on all user sessions ('all_sessions'). These core datasets come in different sizes, and, each of them has a variant that was intentionally made faulty (totally riddled with errors and inconsistencies). This suite of tables is useful for testing with packages that focus on data validation and data documentation.

License  MIT + file LICENSE

Encoding  UTF-8

RoxygenNote  7.2.3

Suggests  DBI (>= 1.1.3), dplyr (>= 1.1.0), duckdb (>= 0.7.0), pointblank (>= 0.11.3), testthat, tibble (>= 3.1.8)

Depends  R (>= 3.5)

NeedsCompilation  no

Author  Richard Iannone [aut, cre] (<https://orcid.org/0000-0003-3925-190X>)

Maintainer  Richard Iannone <riannone@me.com>

Repository  CRAN

Date/Publication  2023-02-22 14:20:06 UTC

R topics documented:

  all_revenue ................................................. 2
  all_revenue_dd ............................................ 3
  all_sessions ................................................. 4
  all_sessions_dd ............................................ 5
 :intendo_metadata .......................................... 6
  users_daily .................................................. 6
  users_daily_dd ............................................. 7
This summary table provides revenue data for every in-app purchase and ad view for players of Super Jetroid in 2015.

Usage

```r
all_revenue(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb", "csv"),
  keep = FALSE
)
```

Arguments

- **size**: A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.

- **quality**: The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.

- **type**: The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the `duckdb` package installed, you can instead obtain the table as an in-memory DuckDB database table.

- **keep**: Should the downloaded data be stored on disk in the working directory? By default, this is FALSE. If the file is available in the next invocation then the data won't be downloaded again.

Value

A data table object, which could be a tibble (`tbl_df`) a data frame, or an in-memory DuckDB table (`tbl_dbi`). If a CSV is written then TRUE will be invisibly returned.
**all_revenue_dd**

**Examples**

```r
# Get a preview of the 'all_revenue' dataset
# with the 'preview' size option
all_revenue(size = "preview")
```

---

**Data Dictionary for all_revenue**

**Description**

The `all_revenue_dd()` function generates a data dictionary based on the `all_revenue` table.

**Usage**

```r
all_revenue_dd(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb")
)
```

**Arguments**

- `size`: A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.
- `quality`: The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.
- `type`: The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the `duckdb` package installed, you can instead obtain the table as an in-memory DuckDB database table.

**Value**

A `ptblank_informant` object.

**Examples**

```r
# Get a preview of the 'all_revenue' dataset
# with the 'preview' size option
all_revenue_dd(size = "preview")
```
Description

This table provides information on player sessions and summarizes the number of revenue events (ad views and IAP spends) and provides total revenue amounts (in USD) broken down by type for the session.

Usage

```r
all_sessions(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb", "csv"),
  keep = FALSE
)
```

Arguments

- **size**: A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.
- **quality**: The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.
- **type**: The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the `duckdb` package installed, you can instead obtain the table as an in-memory DuckDB database table.
- **keep**: Should the downloaded data be stored on disk in the working directory? By default, this is `FALSE`. If the file is available in the next invocation then the data won't be downloaded again.

Value

A data table object, which could be a tibble (`tbl_df`) a data frame, or an in-memory DuckDB table (`tbl_dbi`). If a CSV is written then `TRUE` will be invisibly returned.

Examples

```r
# Get a preview of the 'all_sessions' dataset
# with the 'preview' size option
all_sessions(size = "preview")
```
Description

The all_sessions_dd() function generates a data dictionary based on the all_sessions table.

Usage

```r
all_sessions_dd(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb")
)
```

Arguments

- **size**: A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.
- **quality**: The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.
- **type**: The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the `duckdb` package installed, you can instead obtain the table as an in-memory DuckDB database table.

Value

A `ptblank_informant` object.

Examples

```r
# Get a preview of the 'all_sessions' dataset
# with the 'preview' size option
all_sessions_dd(size = "preview")
```
 **intendo_metadata**  
*Get a data frame with metadata for all datasets*

**Description**

The `intendo_metadata()` function provides a data frame containing metadata for datasets in the `intendo` package.

**Usage**

```r
intendo_metadata(include = c("all", "perfect", "faulty"))
```

**Arguments**

- `include`  
  Should both perfect and faulty datasets be included in the metadata table? This is the default ("all"), otherwise one can use the "perfect" or "faulty" keywords to get a subset of the metadata table.

**Value**

A data frame.

**Examples**

```r
# Obtain metadata on all datasets in the package but
# only those in their 'perfect' form
intendo_metadata(include = "perfect")
```

---

**users_daily**  
*Daily users playing Super Jetroid*

**Description**

This summary table provides daily totals for every player that had at least one login/session in a day. We get measures such as daily sessions, time played, number of IAPs bought and ads viewed, revenue gained, progression info, and some segmentation categories.

**Usage**

```r
users_daily(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb", "csv"),
  keep = FALSE
)
```
**Arguments**

- **size**: A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.

- **quality**: The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.

- **type**: The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the duckdb package installed, you can instead obtain the table as an in-memory DuckDB database table.

- **keep**: Should the downloaded data be stored on disk in the working directory? By default, this is FALSE. If the file is available in the next invocation then the data won’t be downloaded again.

**Value**

A data table object, which could be a tibble (tbl_df) a data frame, or an in-memory DuckDB table (tbl_dbi). If a CSV is written then TRUE will be invisibly returned.

**Examples**

```r
# Get a preview of the 'users_daily' dataset
# with the 'preview' size option
users_daily(size = "preview")
```

**Description**

The `users_daily_dd()` function generates a data dictionary based on the users_daily table.

**Usage**

```r
users_daily_dd(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb")
)
```
Arguments

size
A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.

quality
The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.

type
The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the duckdb package installed, you can instead obtain the table as an in-memory DuckDB database table.

Value
A ptblank_informant object.

Examples

# Get a preview of the 'users_daily' dataset
# with the 'preview' size option
users_daily_dd(size = "preview")

user_summary

User summaries for Super Jetroid

Description

This summary table provides information on each user. We get information here such as the first login/session time and some information useful for segmentation.

Usage

user_summary(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb", "csv"),
  keep = FALSE
)

Arguments

size A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.

quality The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.

type The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the duckdb package installed, you can instead obtain the table as an in-memory DuckDB database table.

keep Should the downloaded data be stored on disk in the working directory? By default, this is FALSE. If the file is available in the next invocation then the data won’t be downloaded again.

Value

A data table object, which could be a tibble (tbl_df) a data frame, or an in-memory DuckDB table (tbl_dbi). If a CSV is written then TRUE will be invisibly returned.

Examples

# Get a preview of the 'user_summary' dataset
# with the 'preview' size option
user_summary(size = "preview")

Description

The user_summary_dd() function generates a data dictionary based on the user_summary table.

Usage

user_summary_dd(
  size = c("small", "medium", "large", "xlarge", "preview"),
  quality = c("perfect", "faulty"),
  type = c("tibble", "data.frame", "duckdb")
)
Arguments

size  A keyword that allows getting different variants of the table based on the size of player base. The default "small" table has the lowest number of players/records. Increasing in size, we can also opt for the "medium", "large", or "xlarge" versions.

quality  The data quality level of the returned dataset. There are two options: (1) "perfect" provides a pristine table with no errors at all and (2) "faulty" gives you a table with a multitude of errors.

type  The table return type. By default, this is a "tibble" but a "data.frame" can instead be returned if using that keyword. If you have the duckdb package installed, you can instead obtain the table as an in-memory DuckDB database table.

Value  A ptblank_informant object.

Examples

# Get a preview of the `user_summary` dataset
# with the 'preview' size option
user_summary_dd(size = "preview")
Index

all_revenue, 2
all_revenue_dd, 3
all_sessions, 4
all_sessions_dd, 5
intendo_metadata, 6
user_summary, 8
user_summary_dd, 9
users_daily, 6
users_daily_dd, 7