# Package ‘completejourney’

October 12, 2022

**Title** Retail Shopping Data

**Version** 1.1.0

**Description** Retail shopping transactions for 2,469 households over one year. Originates from the 84.51° Complete Journey 2.0 source files <https://www.8451.com/area51> which also includes useful metadata on products, coupons, campaigns, and promotions.

**License** CC0

**LazyData** true

**Depends** R (>= 2.10)

**Imports** curl, dplyr, tibble, progress, stringr, zeallot

**Suggests** lubridate, knitr, rmarkdown, testthat

**URL** https://github.com/bradleyboehmke/completejourney

**BugReports** https://github.com/bradleyboehmke/completejourney/issues

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Brad Boehmke [aut, cre] (<https://orcid.org/0000-0002-3611-8516>), Steven M. Mortimer [aut]

**Maintainer** Brad Boehmke <bradleyboehmke@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-09-28 18:30:02 UTC

## R topics documented:

## `campaigns` *Campaigns to household data.*

### Description

Data on the campaigns received by each household in the Complete Journey study. Each household received a different set of marketing campaigns.

### Usage

```
campaigns
```

### Format

A data frame with 6,589 rows and 2 variables

- campaign_id: Uniquely identifies each campaign; Ranges 1-27
- household_id: Uniquely identifies each household

### Value

| | |
|---|---|
| `campaigns` | a tibble |

### Source

84.51°, Customer Journey study, http://www.8451.com/area51/

### Examples

```
# full data set
campaigns

# Join household demographics metadata to campaigns dataset
require("dplyr")
campaigns %>%
  left_join(demographics, "household_id")
```

## `campaign_descriptions` *Campaign metadata.*

### Description

Campaign metadata for all campaigns run for the Customer Journey study. This dataset gives the length of time for which a campaign runs. So, any coupons received as part of a campaign are valid within the dates contained in this dataset.

### Usage

```
campaign_descriptions
```

### Format

A data frame with 27 rows and 4 variables

- campaign_id: Uniquely identifies each campaign; Ranges 1-27
- campaign_type: Type of campaign (Type A, Type B, Type C)
- start_date: Start date of campaign
- end_date: End date of campaign

### Value

`campaign_descriptions`
a tibble

### Source

84.51°, Customer Journey study, http://www.8451.com/area51/

### Examples

```
# full data set
campaign_descriptions

# Join product campaign metadata to campaign_table dataset
require("dplyr")
campaigns %>%
  left_join(campaign_descriptions, "campaign_id")
```

## `completejourney` `completejourney` *package*

### Description

Retail shopping transactions for 2,469 households over one year

### Details

Learn more here: GitHub

### Author(s)

**Maintainer**: Brad Boehmke `<bradleyboehmke@gmail.com>` (0000-0002-3611-8516)

Authors:

- Steven M. Mortimer `<reportmort@gmail.com>`

### See Also

Useful links:

- `https://github.com/bradleyboehmke/completejourney`
- Report bugs at `https://github.com/bradleyboehmke/completejourney/issues`

## `coupons` *Coupon metadata.*

### Description

Coupon metadata for all coupons used in campaigns advertised to households participating in the Customer Journey study.

### Usage

```
coupons
```

### Format

A data frame with 116,204 rows and 3 variables

- coupon_upc: Uniquely identifies each coupon (unique to household and campaign)
- product_id: Uniquely identifies each product
- campaign_id: Uniquely identifies each campaign

### Value

| | |
|---|---|
| `coupons` | a tibble |

### Source

84.51°, Customer Journey study, http://www.8451.com/area51/

### Examples

```
# full data set
coupons

# Join product metadata to coupon dataset
require("dplyr")
coupons %>%
  left_join(products, "product_id")
```

## `coupon_redemptions` *Coupon redemption data.*

### Description

Coupon data identifying the coupons that each household redeemed in the Complete Journey study.

### Usage

```
coupon_redemptions
```

### Format

A data frame with 2,102 rows and 4 variables

- household_id: Uniquely identifies each household
- coupon_upc: Uniquely identifies each coupon (unique to household and campaign)
- campaign_id: Uniquely identifies each campaign
- redemption_date: Date when the coupon was redeemed

### Source

84.51°, Customer Journey study, http://www.8451.com/area51/

### Examples

```
# full data set
coupon_redemptions

# Join coupon metadata to coupon_redempt dataset
require("dplyr")
coupon_redemptions %>%
  left_join(coupons, "coupon_upc")
```

## `demographics` *Household demographic metadata.*

### Description

Household demographic metadata for households participating in the Customer Journey study. Due to nature of the data, the demographic information is not available for all households.

### Usage

```
demographics
```

### Format

A data frame with 801 rows and 8 variables

- household_id: Uniquely identifies each household
- age: Estimated age range
- income: Household income range
- home_ownership: Homeowner status (Homeowner, Renter, Unknown)
- marital_status: Marital status (Married, Single, Unknown)
- household_size: Size of household up to 5+
- household_comp: Household composition description
- kids_count: Number of children present up to 3+

### Value

| | |
|---|---|
| `demographics` | a tibble |

### Source

84.51°, Customer Journey study, http://www.8451.com/area51/

### Examples

```
# full data set
demographics

# Transaction line items that don't have household metadata
require("dplyr")
transactions_sample %>%
  anti_join(demographics, "household_id")
```

---

## get_data *Download full promotions and transactions data simultaneously.*

---

### Description

The promotions and transactions data sets are too large to be contained within the package. `get_data()` is a convenience function to download both full promotions and transactions data sets simultaneously from the source GitHub repository. An internet connection is required.

### Usage

```
get_data(which = "both", verbose = TRUE)
```

### Arguments

| | |
|---|---|
| `which` | Character string of one or more data sets to be downloaded. Can be one of the following; default is `"both"`:<br>• "both"<br>• "promotions"<br>• "transactions" |
| `verbose` | Logical indicator whether or not to download silently. |

### Value

Downloading a single data set will result in a tibble whereas downloading multiple data sets will return a list containing each tibble. For specific details on a given data set see the data sets respective help file (i.e. `?transactions_sample`).

### Source

Downloading from `https://github.com/bradleyboehmke/completejourney/tree/master/data`. Data originated from 84.51°, Customer Journey study, `http://www.8451.com/area51/` and were processes for analysis.

### See Also

Use %<-% for unpacking a list with multiple tibbles to their own global environment tibble. You can also download a single data set with get_promotions and get_transactions.

### Examples

```
# download transactions and promotions data sets
# requires internet connection
c(promotions, transactions) %<-% get_data(which = 'both')
```

---

## get_promotions *Get full Complete Journey promotions data set.*

---

### Description

The complete promotions data set for the Complete Journey is too large to be contained within the package. `get_promotions()` provides an efficient method for downloading the full data set from the source GitHub repository.

### Usage

```
get_promotions(verbose = FALSE)
```

### Arguments

| | |
|---|---|
| verbose | Logical indicator whether or not to download silently. |

### Value

A data frame with 20,940,529 rows and 5 variables

### Source

Downloading from https://github.com/bradleyboehmke/completejourney/tree/master/data. Data originated from 84.51°, Customer Journey study, http://www.8451.com/area51/ and were processes for analysis.

### See Also

promotions_sample for details regarding the variables.

### Examples

```
# requires internet connection
promotions <- get_promotions()
```

## get_transactions *Get full Complete Journey transactions data set.*

### Description

The complete transactions data set for the Complete Journey is too large to be contained within the package. get_transactions() provides an efficient method for downloading the full data set from the source GitHub repository.

### Usage

```
get_transactions(verbose = FALSE)
```

### Arguments

| | |
|---|---|
| verbose | Logical indicator whether or not to download silently. |

### Value

A data frame with 1,469,307 rows and 5 variables

### Source

Downloading from https://github.com/bradleyboehmke/completejourney/tree/master/data. Data originated from 84.51°, Customer Journey study, http://www.8451.com/area51/ and were processes for analysis.

### See Also

transactions_sample for details regarding the variables.

### Examples

```
# requires internet connection
transactions <- get_transactions()
```

products *Product metadata.*

## Description

Product metadata for all products purchased by households participating in the Customer Journey study.

## Usage

```
products
```

## Format

A data frame with 92,331 rows and 7 variables

- product_id: Uniquely identifies each product
- manufacturer_id: Uniquely identifies each manufacturer
- department: Groups similar products together
- brand: Indicates Private or National label brand
- product_category: Groups similar products together at lower level
- product_type: Groups similar products together at lowest level
- package_size: Indicates package size (not available for all products)

## Value

| | |
|---|---|
| `products` | a tibble |

## Source

84.51°, Customer Journey study, http://www.8451.com/area51/

## Examples

```
# full data set
products

# Transaction line items that don't have product metadata
require("dplyr")
transactions_sample %>%
  anti_join(products, "product_id")
```

## `promotions_sample` *Sampling of the full promotions data set.*

### Description

A sampling of the promotions data from the Complete Journey study signifying whether a given product was featured in the weekly mailer or was part of an in-store display (other than regular product placement).

### Usage

```
promotions_sample
```

### Format

A data frame with 360,535 rows and 5 variables

- product_id: Uniquely identifies each product
- store_id: Uniquely identifies each store
- display_location: Display location (see details for range of values)
- mailer_location: Mailer location (see details for range of values)
- week: Week of the transaction; Ranges 1-53

### Value

`promotions_sample`
a tibble

### Display Location Codes

- 0 - Not on Display
- 1 - Store Front
- 2 - Store Rear
- 3 - Front End Cap
- 4 - Mid-Aisle End Cap
- 5 - Rear End Cap
- 6 - Side-Aisle End Cap
- 7 - In-Aisle
- 9 - Secondary Location Display
- A - In-Shelf

### Mailer Location Codes

- 0 - Not on ad
- A - Interior page feature
- C - Interior page line item
- D - Front page feature
- F - Back page feature
- H - Wrap from feature
- J - Wrap interior coupon
- L - Wrap back feature
- P - Interior page coupon
- X - Free on interior page
- Z - Free on front page, back page or wrap

### Source

84.51°, Customer Journey study, http://www.8451.com/area51/

### See Also

Use get_promotions to download the entire promotions data containing all 20,940,529 rows.

### Examples

```
# sampled promotions data set
promotions_sample

# Join promotions to transactions to analyze
# product promotion/location
require("dplyr")
transactions_sample %>%
  left_join(promotions_sample,
            c("product_id", "store_id", "week"))
```

## transactions_sample *Sampling of the full Complete Journey transactions.*

### Description

A sampling of all products purchased by households within the Complete Journey study. Each line found in this table is essentially the same line that would be found on a store receipt. This is only a subsample of the complete data set to keep package size manageable.

### Usage

```
transactions_sample
```

### Format

A data frame with 75,000 rows and 11 variables

**household_id** Uniquely identifies each household

**store_id** Uniquely identifies each store

**basket_id** Uniquely identifies a purchase occasion

**product_id** Uniquely identifies each product

**quantity** Number of the products purchased during the trip

**sales_value** Amount of dollars retailer receives from sale

**retail_disc** Discount applied due to retailer's loyalty card program

**coupon_disc** Discount applied due to manufacturer coupon

**coupon_match_disc** Discount applied due to retailer's match of manufacturer coupon

**week** Week of the transaction; Ranges 1-53

**transaction_timestamp** Date and time of when the transaction occurred

### Value

`transactions_sample`
a tibble

### Source

84.51°, Customer Journey study, `http://www.8451.com/area51/`

### See Also

Use `get_transactions` to download the entire transactions data containing all 1,469,307 rows.

### Examples

```
transactions_sample
```

## %<-% *Assign values to names*

### Description

See %<-% for more details.

### Usage

```
x %<-% value
```

### Arguments

| | |
|---|---|
| x | A name structure. |
| value | A list of values, vector of values, or R objects to assign. |

# Index