# Package 'CADF'

October 31, 2024

**Encoding** UTF-8

**Title** Customer Analytics Data Formatting

**Version** 0.1

**Description** Converts customer transaction data (ID, purchase date) into a R6 class called customer. The class stores various customer analytics calculations at the customer level. The package also contains functionality to convert data in the R6 class to data.frames that can serve as inputs for various customer analytics models.

**License** GPL-3

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.3.1

**Imports** R6

**Suggests** knitr, rmarkdown, lubridate, markovchain, utils, survival

**VignetteBuilder** knitr

**Maintainer** Ludwig Steven <steven.ludwig@u.northwestern.edu>

**NeedsCompilation** no

**Author** Ludwig Steven [aut, cre]

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2024-10-31 14:10:02 UTC

## Contents

## annualhalfingmodel *Annual Halfing Model*

### Description

A recency-frequency model used in non-contractual situations. Model assumptions: 1.) Increasing recency leads to higher probability of quitting. 2.) Frequency is related to exponential learning curves Reference: Segmentation and Lifetime Value Modeling in SAS (Edward Malthouse)

### Usage

```
annualhalfingmodel(cadf.data, starting.values)
```

### Arguments

| | |
|---|---|
| cadf.data | cadf-formatted dataset |
| starting.values | parameter starting values for model |

### Value

Returns model parameters

### Examples

```
dta <- lapply(CADF::cadf.data.sample, function(x) tail(x$data, 1))
dta <- do.call(rbind, dta)
starting.values <- c(.5,.9,.2,-.9)
annualhalfingmodel(cadf.data.sample, starting.values)
```

## annualhalfing_LL *Likelihood maximization for annual halfing customer retention model*

### Description

Likelihood maximization for annual halfing customer retention model

### Usage

```
annualhalfing_LL(grid, dta)
```

### Arguments

| | |
|---|---|
| grid | model parameters |
| dta | dataset |

## Value

Annual halfing Likelihood in optimization routine

---

`bass.answeringmachines` *Answering machine data*

---

## Description

Answering machine data

## Format

A data frame with 9 rows and two columns

---

`bigT_expand_via_apply` *bigT_expand_via_apply*

---

## Description

bigT_expand_via_apply

## Usage

```
bigT_expand_via_apply(x)
```

## Arguments

| | |
|---|---|
| `x` | vector containing bigT, cancel and count |

## Examples

```
x <- c(3, 1, 5)
bigT_expand_via_apply(x)
```

---

`billionaire` *Billionaires*

---

## Description

Billionaires

## Format

data frame

## cadf *cadf.*

### Description

cadf.

## cadf.data.sample *CADF-formatted sample data*

### Description

CADF-formatted sample data

### Format

List with 2,185 customers, in CADF format

## CADF_to_annualhalfing_data *Convert CADF dataset into annualhalfing model dataset*

### Description

Converts CADF output to dataset for annual halfing model

### Usage

```
CADF_to_annualhalfing_data(cadf.data)
```

### Arguments

| | |
|---|---|
| cadf.data | CADF dataset |

`CADF_to_btyd_pareto_nbd` *CADF to btyd pareto nbd model*

## Description

Converts a CADF dataset to a dataset for btyd pareto nbd modeling

## Usage

```
CADF_to_btyd_pareto_nbd(cadf.data)
```

## Arguments

| | |
|---|---|
| `cadf.data` | CADF-formatted dataset |

`CADF_to_logistic_regression` *CADF to logistic regression*

## Description

Convert a CADF dataset to a dataset for logistic regression

## Usage

```
CADF_to_logistic_regression(CADF)
```

## Arguments

| | |
|---|---|
| `CADF` | CADF-formatted dataset |

CADF_to_migration_model *CADF_to_migration_model converts CADF data to migration model data*

## Description

Builds transition matrix for a migration model. T is the maximum time cutoff which defaults to 5. The output will be a transition matrix.

## Usage

```
CADF_to_migration_model(cadf.data, maxT = 5)
```

## Arguments

| | |
|---|---|
| cadf.data | Data in R list format processed by CADF functions |
| maxT | If time is greater than maxT it will be converted into a + category |

## Examples

```
tmatrix <- CADF_to_migration_model(cadf.data.sample)
```

CADF_to_nth_purchase *CADF_to_nth_purchase*

## Description

CADF_to_nth_purchase

## Usage

```
CADF_to_nth_purchase(cadf.data, n)
```

## Arguments

| | |
|---|---|
| cadf.data | Data in R list format processed by CADF functions |
| n | the nth purchase you want to analyze |

CADF_to_nth_purchase_allrows *CADF_to_nth_purchase_allrows inputs CADF data and the desired purchase number that you want to count the nth result of.*

## Description

CADF_to_nth_purchase_allrows inputs CADF data and the desired purchase number that you want to count the nth result of.

## Usage

```
CADF_to_nth_purchase_allrows(cadf.data, n)
```

## Arguments

| | |
|---|---|
| cadf.data | Data in R list format processed by CADF functions |
| n | the nth purchase |

ca_SRM *ca_SRM*

## Description

ca_SRM

## Usage

```
ca_SRM(df_logistic)
```

## Arguments

| | |
|---|---|
| df_logistic | data frame containing the data for logistic regression |

## Examples

```
customertype1 <- c(3, 1, 5)
customertype2 <- c(12, 0, 3)
cust1 <- bigT_expand_via_apply(customertype1)
cust2 <- bigT_expand_via_apply(customertype2)
df_logistic <- rbind(cust1, cust2)
model <- ca_SRM(df_logistic)
```

ca_SRM_time_varying *Time varying Simple retention model Estimates retention rate using logistic regression and the simple regression model Mostly used for contractual models where there are clear opportunities for cancellation. Could be used in non-contractional situations although the cancellation opportunities should be defined. Not recommended for use with services that consumers use rotating-door style. Use the migration model there.*

## Description

Time varying Simple retention model Estimates retention rate using logistic regression and the simple regression model Mostly used for contractual models where there are clear opportunities for cancellation. Could be used in non-contractional situations although the cancellation opportunities should be defined. Not recommended for use with services that consumers use rotating-door style. Use the migration model there.

## Usage

```
ca_SRM_time_varying(df_logistic, reference_level = 12, maxT = 12)
```

## Arguments

| | |
|---|---|
| df_logistic | A data frame, formatted for logistic regression. 1 row for each customer id/timeperiod. 1/0 for purchase. |
| reference_level | All coefficients will be judged relevant to the reference level. It defaults to time period 12. (Note interpretation will change based on how T is formulated.) |
| maxT | The number of timeperiods to build. |

## Value

Returns logistic model results (the glm model)

## Examples

```
library(stats)
x <- c(3, 1, 5)
df_logistic <- bigT_expand_via_apply(x)
model <- ca_SRM_time_varying(df_logistic, reference_level = 3)
```

`ca_to_ps_matrix` *CADF to purchase string Extracts purchase strings from the CADF and formats as a R matrix.*

## Description

CADF to purchase string Extracts purchase strings from the CADF and formats as a R matrix.

## Usage

```
ca_to_ps_matrix(ca.data, maxT)
```

## Arguments

| | |
|---|---|
| `ca.data` | Data in the CADF format generated by the CADF _to_CADF functions and Customer class. |
| `maxT` | Number of columns in the matrix |

## Details

Output is a matrix. Rows are number of customers; columns = maxT

## Value

Matrix with dimensions C x maxT (number of customers by maxT) library(CADF) data("transactions")
customer <- subset(transactions, transactions$ID == 40) today.study.cutoff <- max(customer$PURCHASE_DATE)
customer.40.CADF <- list(Customer$new(customer, today.study.cutoff)) psmatrix <- customer.40.CADF$purchase_string_as
psmatrix2 <- ca_to_ps_matrix(customer.40.CADF, 15)

`create.purchase.string` *Function called during Customer$new() (the Customer R6 class) to create purchase string for the customer.*

## Description

Function called during Customer$new() (the Customer R6 class) to create purchase string for the customer.

## Usage

```
create.purchase.string(x, id.column, date.column, return.mode = "")
```

### Arguments

| | |
|---|---|
| x | Transactional data associated with customer id. |
| id.column | Description goes here. |
| date.column | Description goes here. |
| return.mode | Set to matrix if you want result returned as a matrix |

### Value

purchase string in 0/1 format. Returned as string.

### Examples

```
data("transactions")
customer <- subset(transactions, transactions$ID == 5)
create.purchase.string(customer, "ID", "PURCHASE_DATE")
```

---

## create.recency.string *create_recency_string*

---

### Description

Tracks cumulative recency

### Usage

```
create.recency.string(x)
```

### Arguments

| | |
|---|---|
| x | vector of zeros and ones |

### Examples

```
head(cadf.data.sample)
```

`Customer` *R6 Class representing a customer. Otherwise known as the CADF.*

## Description

A short description...

## Details

Call Customer$new() to convert transactional data to CADF format

## Public fields

`output` Stores all information in R format at the customer level.

`payload` Stores all computed customer information in JSON format for integration into other systems. This is not quite an API but designed so that customer information can be imported to other formats and systems.

`data` a data frame that stores purchase information for a single customer. Input data for various calculations in initialize (df_customer)

`id` The customer id. This will be the same ID as provided in the input transaction file.

`study_name` A name to associate with the cohort study. #The name can be whatever is easiest to associate with the set of customer id and dates included in the analysis.

`study_begin_date` Begin date of the customer study. In theory this should be min(TRANSACTION_DATE) for each customer in the dataset.

`timing` Monthly timing computes T as months. Most commonly utilized and is the default.

`transaction_dates` All transaction dates for the customer

`transaction_months` All YYYY_MM transaction dates for the customer

`first_purchase_date` First purchase date for the customer.

`last_purchase_date` Last purchase date for the customer. #' @field repeat_customer repeat_customer if the following conditions are true. The customer has more than one transaction. The second transaction date is greater than the first transaction date.

`repeat_customer_by_day` description

`today` today #' @field T a measure of time between first date of activity and purchase.

`T_ss` T_ss

`transaction_range_complete` shows a consecutive sequence usually beginning at 1

`purchase_count` purchase count

`purchase_string` description

`purchase_string_as_matrix` purchase string as matrix

`recency_string_as_matrix` recency string as matrix

`Freq` frequency count

`logistic_modeling_matrix` Stores customer's logistic modeling matrix. (One row for each time period (T), 1 = purchase; 0 = no purchase)

`logistic_modeling_matrix_ss` logistic_modeling_matrix_ss

`logistic_modeling_matrix_custom` logistic_modeling_matrix_custom

`survival_modeling_matrix` Stores customer's modeling matrix for survival analysis. For survival analysis '1' means that the customer has stopped being a customer. '0' means that the customer is continuing to be a customer.

`survival_modeling_matrix_ss` survival_modeling_matrix_ss

`survival_modeling_matrix_custom` survival_modeling_matrix_custom

`repeat_customer` This can be used to filter out repeat customers from analysis. Repeat customer based on YYYY_MM. (Customer with only two purchases in January would not be a repeat customer) however it's by day instead of YYYY_MM. PURCHASE STRINGS purchase_string Utilizes the 'create.purchase.string' function to create a purchase string. "1" if purchase was made during the purchase period; "0" otherwise. No special rules are applied and the purchase string reflects true purchase history. df_customer: data frame for single customer, id column, purchase date column

`T` T is a cancellation time. CADF offers different ways to estimate the cancellation time strict_quitter: Customer leaves after first period of inactivity. Example purchase string 11001. T=3 strict_stayer: T is the last period of transaction in the purchase string. 11001. T=5 As T becomes longer strict_quitter will have a tendancy to underestimate retention. Strict_stayer will have a tendancey to overestimate If you know your customers come and go at free will you can utilize a Migration model or choose T between strict quitter and strict stayer

T_ss T_ss

`T_custom` T_custom logistic_modeling_matrix Stores rows for the customer that contribute to a logistic modeling matrix. Assumes strict/perm cancellations. Customer relationship starts at time 1 and ends at time N (with perm cancellation and no pauses in between) This is usually known as a contractual relationship logistic_modeling_matrix_sc Assumes strict stayer assumption $field logistic_modeling_matrix_custom survival_modeling_matrix Stores rows for the customer that contribute to a survival modeling matrix. $field logistic_modeling_matrix_custom cleanup and data storage empty working df_customer data frame and place the result in the class, name it 'data'

## Methods

### Public methods:

- `Customer$new()`
- `Customer$clone()`

**Method** `new()`**:** Creates a CADF profile for a given customer based on the input transactional data usually an R list

*Usage:*

```
Customer$new(df_customer = NA, today = NA)
```

*Arguments:*

`df_customer` description

`today`

*Returns:* A new ‘Customer‘ object. Converted transactional data to CADF format. To access cadf[[1]], etc... Represents customer data (for a particular id) in the "CADF" format df_customer$Tdays df_customer data frame column: to compute "days from first purchase" df_customer$month_yr date converted to YYYY_MM format df_customer$Tmonths Number of months between purchase date and first purchase date. Rounded up to nearest month id the customerid which identifies the customer in the CADF class. transaction_dates All unique transaction dates for customer All unique YYYY_MM combinations for customer transactions. This is used for building purchase strings.

**Method** `clone()`**:** The objects of this class are cloneable with this method.

*Usage:*

```
Customer$clone(deep = FALSE)
```

*Arguments:*

`deep` Whether to make a deep clone.

## Examples

```
library(CADF)
data("transactions")
customer <- subset(transactions, transactions$ID == 40)
today.study.cutoff <- max(customer$PURCHASE_DATE)
customer.40.CADF <- Customer$new(customer, today.study.cutoff)
```

---

`discretechoice` *Discrete choice*

---

## Description

Discrete choice

## Format

##’discretechoice’

---

`exceldata` *Excel data*

---

## Description

Excel data

## Format

Data frame with 50 rows and 9 columns

## fp *Health Data*

### Description

Health Data

### Format

data frame with 5,432 rows and 36 columns

## frequency_from_ps *Purchase string to frequency count*

### Description

Purchase string to frequency count

### Usage

```
frequency_from_ps(x)
```

### Arguments

| | |
|---|---|
| x | rle object |

## frequency_from_rle *RLE object to frequency count*

### Description

RLE object to frequency count

### Usage

```
frequency_from_rle(x)
```

### Arguments

| | |
|---|---|
| x | rle object |

### Examples

```
# example code
x <- c(1,1,0,1,0,0,1,0,0,0)
x.rle <- rle(x)
frequency_from_rle(x.rle)
```

`f_CustomerModelingMatrix` *For each customer, return a modeling matrix that is utilized for logistic regression*

## Description

'f_CustomerModelingMatrix' inputs are cancellation_time.

## Usage

```
f_CustomerModelingMatrix(cancellation_time)
```

## Arguments

`cancellation_time`
= cancellation time

## Details

Description here

## Examples

```
f_CustomerModelingMatrix(10)
```

`f_CustomerSurvivalModelingMatrix` *For each customer, return a survival modeling matrix that is utilized for survival analysis*

## Description

'f_CustomerSurvivalModelingMatrix' inputs are T.

## Usage

```
f_CustomerSurvivalModelingMatrix(cancellation_time)
```

## Arguments

`cancellation_time`
cancellation time

## Details

Description here

## Examples

```
f_CustomerSurvivalModelingMatrix(10)
```

---

`f_intMonths` *Compute the months between two purchase dates*

---

## Description

Compute the months between two purchase dates

## Usage

```
f_intMonths(a, b)
```

## Arguments

| | |
|---|---|
| a | starting date |
| b | ending date |
| | Description here |

---

`gammagamma` *Gamma gamma spend model data*

---

## Description

Gamma gamma spend model data

## Format

data frame with 2,357 rows and 6 columns

---

`generate_date_template` *generate_date_template*

---

## Description

generate_date_template

## Usage

```
generate_date_template()
```

## Examples

```
dates <- generate_date_template()
```

## id_to_CADF *Convert to CADF for a single customer id*

### Description

'id_to_CADF' inputs is coming from a lapply operation on a split customer dataset. If variable a is the split customer dataset then a$'1' is customer with ID 1

### Usage

```
id_to_CADF(data, today.study.cutoff)
```

### Arguments

| | |
|---|---|
| data | Transactional Data for one customerid |
| today.study.cutoff | Separate data an holdout |

### Details

Description here

## ld_sample_customer_matrix *LD functions are utilized for learning and diagnostic use.*

### Description

LD functions are utilized for learning and diagnostic use.

### Usage

```
ld_sample_customer_matrix(numCustomers, maxT, purchaseAtT0 = TRUE)
```

### Arguments

| | |
|---|---|
| numCustomers | number of customers to simulate |
| maxT | number of timeperiods |
| purchaseAtT0 | by default sets first column of matrix to 1 |

## ltv.transactions *LTV transactions data*

### Description

LTV transactions data

### Format

data frame with 53,998 rows and 4 columns

## modeling.annualhalfing.likelihood *Likelihood function for annual halfing model*

### Description

Likelihood function for annual halfing model

### Usage

```
modeling.annualhalfing.likelihood(grid2, rec, freq, targetBuy)
```

### Arguments

| | |
|---|---|
| grid2 | Modeling parameters |
| rec | recency |
| freq | frequency |
| targetBuy | indicator if purchase was made in holdout period |

## modeling.LL.gamma_spend *LL function for the gamma gamma spend model*

### Description

LL function for the gamma gamma spend model

### Usage

```
modeling.LL.gamma_spend(p, q, gamma, y = data)
```

### Arguments

| | |
|---|---|
| p | p |
| q | q |
| gamma | gamma |
| y | data |

---

## pdf_gamma *PDF probability function for gamma distribution*

---

### Description

PDF probability function for gamma distribution

### Usage

```
pdf_gamma(x, r, a)
```

### Arguments

| | |
|---|---|
| x | between 0 and 1 for pdf |
| r | shape parameter |
| a | scale parameter |

---

## pdf_gamma2 *Probability density function for gamma distribution*

---

### Description

Probability density function for gamma distribution

### Usage

```
pdf_gamma2(x, shape, scale)
```

### Arguments

| | |
|---|---|
| x | x |
| shape | shape parameter |
| scale | scale parameter |

## print.glossary *The glossary for the CADF data format*

### Description

The glossary for the CADF data format

### Usage

```
## S3 method for class 'glossary'
print()
```

## psmatrix_to_psstring *psmatrix_to_psstring*

### Description

psmatrix_to_psstring

### Usage

```
psmatrix_to_psstring(psmatrix)
```

### Arguments

| | |
|---|---|
| psmatrix | purchase string of 1's and 0's in matrix format |

### Examples

```
cadf.data.sample[[4]]$purchase_string_as_matrix
```

## psmatrix_to_recency_attimeof_matrix *accepts a psmatrix converts 1/0 purchase strings to recency at timeof*

### Description

accepts a psmatrix converts 1/0 purchase strings to recency at timeof

### Usage

```
psmatrix_to_recency_attimeof_matrix(psmatrix)
```

### Arguments

| | |
|---|---|
| psmatrix | a psmatrix |

`ps_to_T_custom` *Calculates T from a purchase string. Custom.*

## Description

Calculates T from a purchase string. Custom.

## Usage

```
ps_to_T_custom(ps, skips = 2)
```

## Arguments

| | |
|---|---|
| `ps` | Purchase string. |
| `skips` | Number of non purchase periods that the customer is still considered a customer for. |

## Value

The sum of `x` and `y`.

`ps_to_T_strict_quitter` *Calculates T from a purchase string*

## Description

Calculates T from a purchase string

## Usage

```
ps_to_T_strict_quitter(ps)
```

## Arguments

| | |
|---|---|
| `ps` | Purchase string. |

## Value

The sum of `x` and `y`.

`ps_to_T_strict_stayer` *Calculates T from a purchase string under the "strict stayer" assumption.*

## Description

Calculates T from a purchase string under the "strict stayer" assumption.

## Usage

```
ps_to_T_strict_stayer(ps)
```

## Arguments

| | |
|---|---|
| `ps` | Purchase string. |

## Value

The numeric value for T, which is the position of the last 1 in the purchase string

`qc_transactional_data` *The customer analytics data format (CADF) relays heavily on correct input data. Transactional data must: 1.) be a data frame with two columns 2.) Column one is the customer id 3.) Column 2 is the transaction date. Column 2 must be formatted as a date object in R.*

## Description

The customer analytics data format (CADF) relays heavily on correct input data. Transactional data must: 1.) be a data frame with two columns 2.) Column one is the customer id 3.) Column 2 is the transaction date. Column 2 must be formatted as a date object in R.

## Usage

```
qc_transactional_data(x)
```

## Arguments

| | |
|---|---|
| `x` | R dataframe representing .. |

## Value

A number representing whether it passes or not.

## segltv *Segmentation and LTV data*

### Description

Segmentation and LTV data

### Format

A data frame with 53998 rows and 4 columns

## simple_migration *Simple Migration*

### Description

Function used for simulation and scenario planning

### Usage

```
simple_migration(num.customers, pct.buy.buy, pct.nobuy.buy, n.periods)
```

### Arguments

| | |
|---|---|
| num.customers | Number of customers for the simulation. |
| pct.buy.buy | percentage of customers that buy in the nxt period |
| pct.nobuy.buy | percentage of non buyers that convert over to buyers |
| n.periods | number of periods |

### Examples

```
simple_migration(200, .80, .20, 12)
```

`split.transaction.file_to_CADF` *Create a CADF dataset from a dataframe*

## Description

Create a CADF dataset from a dataframe

## Usage

```
## S3 method for class 'transaction.file_to_CADF'
split(data, today.study.cutoff)
```

## Arguments

| | |
|---|---|
| `data` | data frame for a single customer id |
| `today.study.cutoff` | separate analysis and holdout data |

`srm_data` *#' Simple retention model data*

## Description

#' Simple retention model data

## Format

A data frame with 5828 rows and two columns

**bigT** Time period

**cancel** Whether or not there was a cancellation in the time period ...

`srm_summaries` *SRM model data*

## Description

SRM model data

## Format

Data frame with 22 rows and 3 columns

## stocks *Stockmarket put/call data*

### Description

Stockmarket put/call data

### Format

A data frame with 770 rows and 20 columns

## transactions *Transactions data*

### Description

Transactions data

### Format

data frame with 69659 rows and 4 columns

## transactions.merged *#' Transaction data*

### Description

#' Transaction data

### Format

A data frame with 67,944 rows and 4 columns

**ID** Customer ID

**PURCHASE_DATE** Purchase date

**NUM_ITEMS** Number of items purchased

**TOTAL** Total transaction amount ...

`transitions` *Calculate transition periods between two timeperiods*

## Description

Calculate transition periods between two timeperiods

## Usage

```
transitions(timeperiod0, timeperiod1, buyvar = "Y", nobuyvar = "N")
```

## Arguments

| | |
|---|---|
| `timeperiod0` | Column representing the 'from' side of the transition probability |
| `timeperiod1` | Column representing the 'to' side of the transition probability |
| `buyvar` | field value that represents a buy, defaults to Y |
| `nobuyvar` | field value that represents not buy, defaults to N |

## Value

2 x 2 transaction matrix

## Examples

```
timeperiod0 <- c("Y", "Y", "Y", "Y", "Y")
timeperiod1 <- c("N", "Y", "N", "Y", "N")
transitions(timeperiod0, timeperiod1)
```

# Index