

# Package ‘RMM’

October 12, 2022

**Type** Package

**Title** Revenue Management Modeling

**Version** 0.1.0

**Date** 2022-04-29

**Maintainer** Chul Kim <statkim7578@yonsei.ac.kr>

**Description** The RMM fits Revenue Management Models using the RDE(Robust Demand Estimation) method introduced in the paper by <doi:10.2139/ssrn.3598259>, one of the customer choice-based Revenue Management Model. Furthermore, it is possible to select a multinomial model as well as a conditional logit model as a model of RDE.

**License** GPL (>= 2.0)

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.1.0)

**Imports** tibble, magrittr, rlang, dplyr, numDeriv, Rcpp

**RoxygenNote** 7.1.2

**LinkingTo** Rcpp

**NeedsCompilation** yes

**Author** Chul Kim [aut, cre],  
Sanghoon Cho [aut],  
Jongho Im [aut]

**Repository** CRAN

**Date/Publication** 2022-05-09 11:30:05 UTC

## R topics documented:

Choice_Set	2
Hotel_Long	2
Hotel_Wide	3
predict.rmm	5
rmm	6
rmm_reshape	7

**Index** **9**


---

Choice_Set	<i>Collects a choice set exposed to individuals.</i>
------------	--

---

**Description**

Collects a choice set exposed to individuals. Internal function that users should not call directly.

**Usage**

```
Choice_Set(df, num_id, uniq_id, idvar)
```

**Arguments**

df	a long format tibble.
num_id	Number of unique Booking_IDs appearing in transaction data
uniq_id	Unique Booking_ID in transaction data.
idvar	Variable name representing customer id (Booking_ID).

**Value**

Returns a list containing the values required for calculation within the [rmm\\_reshape](#) function.

---

Hotel_Long	<i>Data from a Major Hotel Chain</i>
------------	--------------------------------------

---

**Description**

'Hotel\_Long', a 'Long format', is a preprocessing data of the publicly available 'Hotel 1' data introduced in Bodea et al. (2009).

**Usage**

```
Hotel_Long
```

**Format**

'Hotel\_Long': A data frame with 8,318 rows and 11 variables:

**Booking\_ID** ID associated with a booking. Begins at one for each hotel property.

**Purchase** Indicator variable equal to one if the product identified by product ID is purchased, zero otherwise.

**Room\_Type** Code describing the room type associated with the product ID.

**Price** The average nightly rate the customer pays in USD (e.g., \$199.99). Note that the average nightly rate will not match the rate of any available product rates if an upsell occurs at time of check-in, if the customer requests a specific discount rate at time of check-in, etc.

**Party\_Size** Number of adults and children associated with the booking.

**Membership\_Status** Status in rewards program (0—not a member, 1—basic, 2—elevated, 3—premium).

**VIP\_Membership\_Status** Membership status of a VIP rewards program member (0—not a VIP, 1—basic VIP, 2—premium VIP member).

**Booking\_Date** Date the booking was created (e.g., 20070303 = March 3, 2007).

**Check\_In\_Date** Check-in date (e.g., 20070307 = March 7, 2007).

**Check\_Out\_Date** Check-out date (e.g., 20070310 = March 10, 2007).

**Length\_of\_Stay** Length of stay/number of nights (e.g., three).

## Details

'Hotel 1' data contains information on the available alternatives, i.e., choice sets and the associated prices at the time of each customer's booking decision. We preprocessed 'Hotel 1' data and provide it in two types of data format, 'Hotel\_Long' and 'Hotel\_Wide'.

The following are the preprocessing of 'Hotel 1' data.

1. Customers' booking transactions that had only one room type available in their choice set were removed as our methods require at least two different products in each choice set.
2. Duplicate records was removed.
3. Choice sets with less than 30 observations, representing rare case were removed.

## Source

[doi:10.1287/msom.1080.0231](https://doi.org/10.1287/msom.1080.0231)

---

Hotel\_Wide

*Data from a Major Hotel Chain*

---

## Description

'Hotel\_Wide', a 'Wide format', is a preprocessing data of the publicly available 'Hotel 1' data introduced in Bodea et al. (2009).

## Usage

Hotel\_Wide

## Format

'Hotel\_Wide': A data frame with 1,100 rows and 22 variables:

**Booking\_ID** ID associated with a booking. Begins at one for each hotel property.

**Party\_Size** Number of adults and children associated with the booking.

**Membership\_Status** Status in rewards program (0—not a member, 1—basic, 2—elevated, 3—premium).

**VIP\_Membership\_Status** Membership status of a VIP rewards program member (0—not a VIP, 1—basic VIP, 2—premium VIP member).

**Booking\_Date** Date the booking was created (e.g., 20070303 = March 3, 2007).

**Check\_In\_Date** Check-in date (e.g., 20070307 = March 7, 2007).

**Check\_Out\_Date** Check-out date (e.g., 20070310 = March 10, 2007).

**Length\_of\_Stay** Length of stay/number of nights (e.g., three).

**Room\_Type** Code describing the room type associated with the product ID.

**Purchased\_Prod\_Code** Unique number of products associated with room type.

**Exposed\_Choice\_Set** Choice set exposed to individual

**Exposed\_Choice\_Set\_Code** Unique number of choice sets associated with Exposed\_Choice\_Set

**Price\_1 ~ Price\_10** The average nightly rate the customer pays in USD (e.g., \$199.99). Note that the average nightly rate will not match the rate of any available product rates if an upsell occurs at time of check-in, if the customer requests a specific discount rate at time of check-in, etc.

## Details

'Hotel 1' data contains information on the available alternatives, i.e., choice sets and the associated prices at the time of each customer's booking decision. We preprocessed 'Hotel 1' data and provide it in two types of data format, 'Hotel\_Long' and 'Hotel\_Wide'.

The following are the preprocessing of 'Hotel 1' data.

1. Customers' booking transactions that had only one room type available in their choice set were removed as our methods require at least two different products in each choice set.
2. Duplicate records was removed.
3. Choice sets with less than 30 observations, representing rare case were removed.

## Source

[doi:10.1287/msom.1080.0231](https://doi.org/10.1287/msom.1080.0231)

---

 predict.rmm

*Predict method for Revenue Management Model Fits*


---

## Description

Predicted values based on RMM object

## Usage

```
## S3 method for class 'rmm'
predict(object, newdata, Rem_Choice_Set, Choice_Set_Code, fixed = TRUE, ...)
```

## Arguments

object	Object of class inheriting from "rmm"
newdata	A data frame in which to look for variables with which to predict.
Rem_Choice_Set	List of choice sets remaining in the data.
Choice_Set_Code	Specifies the choice set of newdata.
fixed	If fixed=TRUE, the alternative with the highest prediction probability is determined as decision. Otherwise (fixed=FALSE), one of the alternatives is determined in proportion to the predictive probability.
...	further arguments passed to or from other methods.

## Value

preict.rmm produces a list of predictions, which contains decisions and probabilities.

## Examples

```
data(Hotel_Long)

# Before using the rmm function, the user must first use the rmm_shape function.
rst_reshape <- rmm_reshape(data=Hotel_Long, idvar="Booking_ID",
  alts="Room_Type", asv="Price", resp="Purchase", min_obs=30)

# Fitting a model
rst_rmm <- rmm(rst_reshape, prop=0.7, model="c1")

# Predictions
Rem_Choice_Set <- rst_reshape$Rem_Choice_Set

newdata1 <- data.frame(Price_1=c(232, 122, 524), Price_3=c(152, 531, 221),
  Price_4=c(163, 743, 192), Price_5=c(132, 535, 325),
  Price_7=c(136, 276, 673), Price_8=c(387, 153, 454),
  Price_9=c(262, 163, 326), Price_10=c(421, 573, 472))
```

```

predict(rst_rmm, newdata=newdata1, Rem_Choice_Set=Rem_Choice_Set,
        Choice_Set_Code=3, fixed=TRUE)

newdata2 <- data.frame(Price_1=c(521, 321, 101, 234, 743),
                      Price_5=c(677, 412, 98, 321, 382),
                      Price_8=c(232, 384, 330, 590, 280))

predict(rst_rmm, newdata=newdata2, Rem_Choice_Set=Rem_Choice_Set,
        Choice_Set_Code=7, fixed=FALSE)

```

---

rmm

*Fitting Revenue Management Models*


---

## Description

`rmm` is used to fit Revenue Management Models. Users can specify `cl` (conditional logit model) and `ml` (multinomial logit model) as RMM model.

## Usage

```
rmm(rmm_data, prop = 0.7, model = "cl")
```

## Arguments

<code>rmm_data</code>	an object of class "rmm_data", a output of <code>rmm_reshape</code> function.
<code>prop</code>	numeric, user assumed market share.
<code>model</code>	character, specify fitting method ("cl" or "ml"). "cl" (default) refers to the Conditional Logit Model, and "ml" refers to the Multinomial Logit Model.

## Value

`rmm` returns an object of class inheriting from "rmm".

## See Also

`rmm` fits the model with the RDE method introduced in [doi:10.2139/ssrn.3598259](https://doi.org/10.2139/ssrn.3598259).

## Examples

```

data(Hotel_Long)

# Before using the rmm function, the user must first use the rmm_shape function.
rst_reshape <- rmm_reshape(data=Hotel_Long, idvar="Booking_ID", alts="Room_Type",
                          asv="Price", resp="Purchase", min_obs=30)

```

```
# Fitting a model
rst_rmm <- rmm(rst_reshape, prop=0.7, model="cl")
print(rst_rmm)
```

---

rmm\_reshape

*Reshape Long-Format Data*

---

## Description

This function reshapes a 'Long-Format' data (with the repeated measurements in separate rows) to 'Wide-Format' data (with repeated measurements in separate columns of the same row). The reshaped 'wide-format' data is an S3 class called 'rmm\_data' and contains information for fitting the model with the `rmm` function. Users who want to use the `rmm` function must first use the `rmm_reshape` function. The `rmm` function receives only S3 class 'rmm\_data' as input.

## Usage

```
rmm_reshape(data, idvar, resp, alts, asv, min_obs)
```

## Arguments

<code>data</code>	data frame, a 'Long-Format' transaction data.
<code>idvar</code>	character, variable name representing each individual's id in the transaction data.
<code>resp</code>	character, variable name representing result of a individual choice.
<code>alts</code>	character vector, variable names representing a alternatives.
<code>asv</code>	character vector, variable names representing a alternative specific variables.
<code>min_obs</code>	numeric, specify the minimum observation for each choice set in the transaction data.

## Value

The 'Wide-Format' data and various information required for the `rmm` function.

## See Also

`rmm` for estimating parameters.

## Examples

```
data(Hotel_Long)

rst_reshape <- rmm_reshape(data=Hotel_Long, idvar="Booking_ID",
  resp="Purchase", alts="Room_Type", asv="Price", min_obs=30)

class(rst_reshape) # "rmm_data"
```

```
ls(rst_reshape) # "Alts_Code_Desc" "ASV" "asv_name" "data_wide"  
                # "Rem_Choice_Set"   "Removed_Choice_Set"  
  
rst_reshape$data_wide # reshaped data
```



# Index

## \* datasets

Hotel\_Long, 2

Hotel\_Wide, 3

Choice\_Set, 2

Hotel\_Long, 2

Hotel\_Wide, 3

predict.rmm, 5

rmm, 6, 6, 7

rmm\_reshape, 2, 6, 7