

Package: turfLP (via r-universe)

October 12, 2026

Title TURF Analysis with Integer Linear Programming

Version 0.2.0

Description Finds product portfolios that maximize TURF (total unduplicated reach and frequency) with integer linear programming, following Serra (2013) [<doi:10.1016/j.foodqual.2012.10.001>](https://doi.org/10.1016/j.foodqual.2012.10.001). The maximum reach problem is the maximal covering location problem of Church and ReVelle (1974) [<doi:10.1007/BF01942293>](https://doi.org/10.1007/BF01942293). The package solves it as an integer linear program, so it finds exact optima without enumerating every portfolio. Ties on reach are broken by frequency and then by the harmonic mean of the individual product reaches. The package also finds the smallest portfolio that reaches every reachable respondent. For related work on TURF for large data sets, see Ennis, Fayle, and Ennis (2012) [<doi:10.1016/j.foodqual.2011.06.004>](https://doi.org/10.1016/j.foodqual.2011.06.004).

License MIT + file LICENSE

Copyright See the file COPYRIGHTS.

URL <https://github.com/aigorahub/turfLP>

BugReports <https://github.com/aigorahub/turfLP/issues>

Depends R (>= 3.5.0)

Imports highs (>= 1.14.0), Matrix, stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

LazyData true

Config/pak/sysreqs cmake pkg-config

Repository <https://aigorahub.r-universe.dev>

Date/Publication 2026-10-12 21:20:54 UTC
RemoteUrl <https://github.com/aigorahub/turflip>
RemoteRef HEAD
RemoteSha 9e6e7e569c07c53f338b7da817cdb59880110959

Contents

cafe	2
chips	3
coffee	4
ham	5
icecream	6
lunchbags	7
pies	8
turf	9
turf_min_cover	10
turf_simulate	11
turf_sizes	12
Index	13

cafe	<i>Simulated cafe drink data</i>
------	----------------------------------

Description

Simulated answers from 2500 customers who marked which of 40 cafe drinks they would order. A 1 means that the customer would order the drink. This is a large data set, and it is already a reach matrix.

Usage

cafe

Format

A data frame with 2500 rows (customers) and 40 integer columns (drinks), with values 0 and 1 and no missing values.

Details

The data comes from a latent class model with six customer segments (coffee purists, everyday milk coffee, sweet drinks, tea, wellness, and family). The segments make the answers for similar drinks correlated. Some customers would order no drink. The data is simulated and does not describe real customers or products.

Run time depends on the portfolio size, the solver version, and the computer. Start with a small portfolio, such as the size 2 example below.

Source

Simulated with `data-raw/simulated.R` in the package source.

See Also

[icecream](#) and [chips](#) for smaller data sets.

Examples

```
sort(colMeans(cafe), decreasing = TRUE)[1:10]  
turf(cafe, size = 2)
```

chips

Simulated potato chip purchase intent data

Description

Simulated purchase intent from 600 consumers for 24 potato chip flavors on a 5-point scale (1 = definitely would not buy, 3 = might or might not buy, 5 = definitely would buy). This is a medium-size data set.

Usage

```
chips
```

Format

A data frame with 600 rows (consumers) and 24 integer columns (flavors), with answers from 1 to 5 and no missing values.

Details

The data comes from a latent class model with five consumer segments (classic, heat seekers, tangy, cheese, and foodie). The segments make the ratings of similar flavors correlated. The data is simulated and does not describe real consumers or products.

A common way to define reach from purchase intent is the top-2 box: a flavor reaches a consumer who answers 4 or 5. Use `chips >= 4` as the reach matrix.

Source

Simulated with `data-raw/simulated.R` in the package source.

See Also

[icecream](#) for a smaller data set and [cafe](#) for a larger one.

Examples

```
reach <- chips >= 4
sort(colMeans(reach), decreasing = TRUE)
turf_sizes(reach, sizes = 1:4)
```

coffee

Black coffee liking data

Description

Liking of 27 black coffee brews by 118 consumers on the 9-point hedonic scale. The brews come from a full factorial design with three brew temperatures (87, 90, and 93 degrees Celsius), three strengths (1, 1.25, and 1.5 percent total dissolved solids), and three percent extractions (16, 20, and 24 percent). The column names give these three values: for example, 87-1.0-16 is the brew at 87 degrees, 1 percent total dissolved solids, and 16 percent extraction.

Usage

```
coffee
```

Format

A data frame with 118 rows (consumers) and 27 integer columns (brews), with ratings from 1 to 9 and no missing values.

Details

The products here are recipes of one coffee, not market products. A TURF question for this data is which few brew recipes a cafe should offer so that most consumers find one they like.

Use `coffee >= 8` (the top-2 box) as the reach matrix. With `coffee >= 7`, two or three brews reach most consumers.

Source

The liking column of Ristenpart, W., Cotter, A. R., & Guinard, J.-X. (2023). Consumer preference data for black coffee [Data set]. Dryad. [doi:10.25338/B8993H](https://doi.org/10.25338/B8993H). Dedicated to the public domain under CC0 1.0. `data-raw/real.R` in the package source makes the data set.

References

Cotter, A. R., Batali, M. E., Ristenpart, W. D., & Guinard, J.-X. (2021). Consumer preferences for black coffee are spread over a wide range of brew strengths and extraction yields. *Journal of Food Science*, 86(1), 194-205. [doi:10.1111/17503841.15561](https://doi.org/10.1111/17503841.15561)

Examples

```
turf_sizes(coffee >= 8, sizes = 1:4)
```

ham	<i>Cured ham liking data</i>
-----	------------------------------

Description

Blind liking of 8 cured hams by 127 Norwegian consumers on a 9-point hedonic scale, where 9 is the highest liking. The consumers tasted the hams with no information about them.

Usage

ham

Format

A data frame with 127 rows (consumers) and 8 integer columns (hams), with ratings from 1 to 9 and no missing values.

Details

The columns are the product labels of the source data:

Ham	Style	Price range	Origin
N1	Norwegian	Economy	Norway
N2	Norwegian	Economy	Norway
N3	Norwegian	Premium	Norway
N4	Norwegian	Premium	Norway
S1	Serrano	Economy	Spain
S2	Serrano	Premium	Norway
S3	Serrano	Premium	Norway
S4	Serrano	Premium	Spain

Use `ham >= 7` (the top-3 box) as the reach matrix. With `ham >= 8`, 23 consumers like no ham enough to count as reached.

Source

The blind liking sheet of Berget, I. (2024). Dataset on rapid sensory methods for cured hams and associations with consumer values in the Schwartz model [Data set]. Zenodo. [doi:10.5281/zenodo.10996096](https://doi.org/10.5281/zenodo.10996096). Licensed under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>). Changes: the data was reshaped to one row per consumer, and consumer IDs and all other sheets were removed. `data-raw/real.R` in the package source makes the data set.

Examples

```
turf(ham >= 7, size = 2)
```

`icecream`*Simulated ice cream liking data*

Description

Simulated ratings from 120 consumers who each rated 10 ice cream flavors on the 9-point hedonic scale (1 = dislike extremely, 5 = neither like nor dislike, 9 = like extremely). This is a small data set for first examples.

Usage

```
icecream
```

Format

A data frame with 120 rows (consumers) and 10 integer columns (flavors), with ratings from 1 to 9 and no missing values.

Details

The data comes from a latent class model with four consumer segments (traditional, chocolate lovers, fruit lovers, and adventurous). The segments make the ratings of similar flavors correlated, as in real consumer data. The data is simulated and does not describe real consumers or products.

A common way to define reach from hedonic ratings is the top-2 box: a flavor reaches a consumer who rates it 8 or 9. Use `icecream >= 8` as the reach matrix. A lower threshold such as `icecream >= 7` gives more reach per flavor.

Source

Simulated with `data-raw/simulated.R` in the package source.

See Also

[chips](#) and [cafe](#) for larger data sets.

Examples

```
reach <- icecream >= 8
colMeans(reach)
turf(reach, size = 3)
```

lunchbags*Lunch bag purchase data*

Description

Which of 14 lunch bag designs each of 1175 customers of a UK online retailer bought from December 2010 to December 2011. A 1 means that the customer bought the design at least once. It is already a reach matrix.

Usage

lunchbags

Format

A data frame with 1175 rows (customers) and 14 integer columns (designs), with values 0 and 1 and no missing values.

Details

A TURF question for this data is which few designs the retailer should keep so that most of these customers can still buy a design they bought before. Many of the customers are wholesalers.

Source

Chen, D. (2015). Online Retail [Data set]. UCI Machine Learning Repository. doi:10.24432/C5BW33. Licensed under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>). Changes: only purchases of lunch bags by known customers were kept, cancellations were removed, and the purchases were coded as 0 and 1 for each customer and design. Customer IDs were removed. data-raw/real.R in the package source makes the data set.

References

Chen, D., Sain, S. L., & Guo, K. (2012). Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management*, 19(3), 197-208. doi:10.1057/dbm.2012.17

Examples

```
turf(lunchbags, size = 3)
```

pies

Thanksgiving pie data

Description

The pies that 980 US respondents said are typically served at their Thanksgiving dinner, from a FiveThirtyEight survey in November 2015. A 1 means that the pie is served. Only respondents who celebrate Thanksgiving are included. It is already a reach matrix.

Usage

pies

Format

A data frame with 980 rows (respondents) and 10 integer columns (pies), with values 0 and 1 and no missing values.

Details

The question is about the pies served in the respondent's household, not about the pies that the respondent likes. A TURF question for this data is which few pies a bakery should make so that most households get a pie they serve.

Source

The pie questions of FiveThirtyEight (2015). thanksgiving-2015 [Data set]. <https://github.com/fivethirtyeight/data/tree/master/thanksgiving-2015>. Licensed under CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>). Changes: the answers were coded as 0 and 1, the "None" and "Other" answers and all other questions were removed, and only respondents who celebrate Thanksgiving were kept. data-raw/real.R in the package source makes the data set.

Examples

```
colMeans(pies)
turf(pies, size = 3)
```

turf

*Find the product portfolio with maximum reach***Description**

turf() selects size products that together reach the largest number of respondents. Ties on reach are broken by the criteria in tiebreak, in the order given.

Usage

```
turf(reach, size, tiebreak = c("frequency", "penetration"))
```

Arguments

reach	A matrix or data frame with one row per respondent and one column per product. A value of 1 (or TRUE) means that the product reaches the respondent. All other values must be 0 (or FALSE). Column names are used as product names.
size	The number of products in the portfolio.
tiebreak	A character vector of tie-break criteria to apply after reach, in order. Use any of "frequency" and "penetration", or character(0) for reach only.

Value

An object of class turf_portfolio, a list with these elements:

products Column indices of the selected products.

names Names of the selected products.

size Number of selected products.

reach Number of respondents that at least one selected product reaches.

reach_prop reach divided by the number of respondents.

frequency Sum of the individual reaches of the selected products.

penetration Harmonic mean of the individual reaches of the selected products.

respondents Number of respondents (rows of reach).

Model

Let $a_{ij} = 1$ when product j reaches respondent i , and let $r_j = \sum_i a_{ij}$ be the individual reach of product j . The binary variable x_j selects product j . The continuous variable $z_i \geq 0$ counts respondent i as not reached. The constraints are

$$z_i + \sum_j a_{ij} x_j \geq 1 \quad \text{for each respondent } i,$$

$$\sum_j x_j = k.$$

For integer x , the smallest feasible z_i is 0 when a selected product reaches respondent i and 1 when none does. Only the product variables need to be integer, which keeps the solve fast.

The stages solve in sequence. Each later stage keeps the earlier criteria at their optimal values.

- Reach: minimize $\sum_i z_i$, which maximizes the number of respondents reached.
- "frequency": maximize $\sum_j r_j x_j$, the total number of product and respondent pairs where the product reaches the respondent.
- "penetration": minimize $\sum_j x_j / r_j$, which maximizes the harmonic mean of the individual reaches of the selected products. When reach and frequency are fixed, this prefers portfolios whose products have similar individual reach.

If portfolios still tie after the last stage, the solver returns one of them.

References

- Serra, D. (2013). Implementing TURF analysis through binary linear programming. *Food Quality and Preference*, 28(1), 382-388. doi:10.1016/j.foodqual.2012.10.001
- Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association*, 32(1), 101-118. doi:10.1007/BF01942293
- Ennis, J. M., Fayle, C. M., & Ennis, D. M. (2012). eTURF: A competitive TURF algorithm for large datasets. *Food Quality and Preference*, 23(1), 44-48. doi:10.1016/j.foodqual.2011.06.004
- Miaoulis, G., Free, V., & Parsons, H. (1990). TURF: A new planning approach for product line extensions. *Marketing Research*, 2(1), 28-40.

See Also

`turf_sizes()` to solve several portfolio sizes, `turf_min_cover()` for the smallest portfolio that reaches every reachable respondent.

Examples

```
set.seed(1234)
reach <- turf_simulate(n_respondents = 300, n_products = 15)
turf(reach, size = 4)

# Reach only, with no tie-break
turf(reach, size = 4, tiebreak = character(0))
```

turf_min_cover

Find the smallest portfolio that reaches every reachable respondent

Description

`turf_min_cover()` solves the set cover problem: it finds the fewest products such that every respondent that some product reaches is reached by at least one selected product. Larger portfolios cannot add reach, so for reach this is the largest size worth passing to `turf()`. Larger portfolios can still add frequency.

Usage

```
turf_min_cover(reach)
```

Arguments

reach A matrix or data frame with one row per respondent and one column per product. A value of 1 (or TRUE) means that the product reaches the respondent. All other values must be 0 (or FALSE). Column names are used as product names.

Value

An object of class `turf_portfolio`. See [turf\(\)](#) for its elements. When several portfolios of the minimum size exist, the solver returns one of them.

Examples

```
set.seed(1234)
reach <- turf_simulate()
turf_min_cover(reach)
```

turf_simulate	<i>Simulate a reach matrix</i>
---------------	--------------------------------

Description

`turf_simulate()` makes a random reach matrix for examples and tests. Each product gets a reach probability drawn uniformly from 0 to `max_prob`. The number of respondents that the product reaches is drawn from a binomial distribution with that probability, and those respondents are chosen at random. Products are generated independently. Sample correlations between products can differ from zero.

Usage

```
turf_simulate(n_respondents = 1000, n_products = 30, max_prob = 0.5)
```

Arguments

n_respondents Number of respondents (rows).
n_products Number of products (columns).
max_prob Largest possible reach probability for a product, from 0 to 1.

Details

Call `set.seed()` first for reproducible results.

Value

A numeric matrix of 0 and 1 with n_respondents rows and n_products columns named P1, P2, and so on.

Examples

```
set.seed(1234)
reach <- turf_simulate(n_respondents = 200, n_products = 10)
colSums(reach)
```

turf_sizes	<i>Find the best portfolio for each of several sizes</i>
------------	--

Description

turf_sizes() calls turf() once for each value in sizes and returns the results as one table.

Usage

```
turf_sizes(reach, sizes = NULL, tiebreak = c("frequency", "penetration"))
```

Arguments

- reach A matrix or data frame with one row per respondent and one column per product. A value of 1 (or TRUE) means that the product reaches the respondent. All other values must be 0 (or FALSE). Column names are used as product names.
- sizes The portfolio sizes to solve. The default is every size from 1 to the size from turf_min_cover().
- tiebreak A character vector of tie-break criteria to apply after reach, in order. Use any of "frequency" and "penetration", or character(0) for reach only.

Value

A data frame with one row per size and the columns size, reach, reach_prop, frequency, penetration, and products. The products column holds the names of the selected products, separated by commas.

Examples

```
set.seed(1234)
reach <- turf_simulate(n_respondents = 300, n_products = 15)
turf_sizes(reach, sizes = 1:4)
```

Index

* datasets

- cafe, [2](#)
- chips, [3](#)
- coffee, [4](#)
- ham, [5](#)
- icecream, [6](#)
- lunchbags, [7](#)
- pies, [8](#)

cafe, [2](#), [3](#), [6](#)
chips, [3](#), [3](#), [6](#)
coffee, [4](#)

ham, [5](#)

icecream, [3](#), [6](#)

lunchbags, [7](#)

pies, [8](#)

set.seed(), [11](#)

turf, [9](#)
turf(), [10–12](#)
turf_min_cover, [10](#)
turf_min_cover(), [10](#), [12](#)
turf_simulate, [11](#)
turf_sizes, [12](#)
turf_sizes(), [10](#)