

Package: olr (via r-universe)

May 31, 2026

Title Optimal Linear Regression

Version 1.2

Date 2025-05-05

Description The `olr` function systematically evaluates multiple linear regression models by exhaustively fitting all possible combinations of independent variables against the specified dependent variable. It selects the model that yields the highest adjusted R-squared (by default) or R-squared, depending on user preference. In model evaluation, both R-squared and adjusted R-squared are key metrics: R-squared measures the proportion of variance explained but tends to increase with the addition of predictors—regardless of relevance—potentially leading to overfitting. Adjusted R-squared compensates for this by penalizing model complexity, providing a more balanced view of fit quality. The goal of `olr` is to identify the most suitable model that captures the underlying structure of the data while avoiding unnecessary complexity. By comparing both metrics, it offers a robust evaluation framework that balances predictive power with model parsimony. Example Analogy: Imagine a gardener trying to understand what influences plant growth (the dependent variable). They might consider variables like sunlight, watering frequency, soil type, and nutrients (independent variables). Instead of manually guessing which combination works best, the `olr` function automatically tests every possible combination of predictors and identifies the most effective model—based on either the highest R-squared or adjusted R-squared value. This saves the user from trial-and-error modeling and highlights only the most meaningful variables for explaining the outcome. A Python version is also available at <https://pypi.org/project/olr>.

License GPL-3

Encoding UTF-8

Depends R (>= 2.10)

LazyData true

Imports plyr, utils, stats, readxl, htmltools

Suggests knitr, rmarkdown, ggplot2

VignetteBuilder knitr

RoxygenNote 7.2.3

URL https://github.com/MathHatter/olr_r, <https://pypi.org/project/olr/>

BugReports https://github.com/MathHatter/olr_r/issues

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

Date/Publication 2025-05-10 00:52:47 UTC

RemoteUrl https://github.com/mathatter/olr_r

RemoteRef HEAD

RemoteSha 9620ab7e0441c6ecc9e0b905882c41a35ea72e59

Contents

load_custom_data	2
olr	3
Index	6

load_custom_data	<i>Load custom data from inst/extdata or a user-specified path</i>
------------------	--

Description

This function loads custom data from the inst/extdata directory of the package or a designated path provided by the user.

Usage

```
load_custom_data(
  data = "crudeoildata.csv",
  custom_path = NULL,
  exclude_first_column = FALSE
)
```

Arguments

data	The name of the data file to load (default: "crudeoildata.csv").
custom_path	An optional custom file path. If provided, it overrides the default file path.
exclude_first_column	Logical value indicating whether to exclude the first column from the loaded data (default: FALSE).

Format

A data frame with 55 weekly observations and 19 columns, including headers. The first column represents the date in the format MM/DD/YYYY, while all other columns display weekly percentage changes with five-decimal precision.

Value

A data frame containing the loaded data.

Source

Example dataset compiled from the following public sources: - Crude oil supply/demand metrics (FieldProduction, RefinerNetInput, OperableCapacity, Imports, StocksExcludingSPR): <https://www.eia.gov>
- API gravity and Rig Count: Extracted from industry reports - S&P 500 Index (SPX): <https://fred.stlouisfed.org/series/SP500>
- CFTC positioning data (NonCommercialLong, CommercialShort, OpenInterest, etc.): <https://www.cftc.gov> (NYMEX short format) - TotalLong and TotalShort = NonCommercial (L/S) + Spread (L/S) + Commercial (L/S)

Examples

```
## Not run:  
# Load custom data with default options  
df <- load_custom_data()  
  
# Load data from a custom file path and exclude the first column (e.g., a Date column)  
df <- load_custom_data(  
  data = "crudeoildata.csv",  
  custom_path = "path/to/custom/crudeoildata.csv",  
  exclude_first_column = TRUE  
)  
  
# Load default custom data and exclude the first column  
df <- load_custom_data(exclude_first_column = TRUE)  
  
## End(Not run)
```

Description

The `olr` function systematically evaluates multiple linear regression models by exhaustively fitting all possible combinations of independent variables against the specified dependent variable. It selects the model that yields the highest adjusted R-squared (by default) or R-squared, depending on user preference. In model evaluation, both R-squared and adjusted R-squared are key metrics: R-squared measures the proportion of variance explained but tends to increase with the addition of

predictors—regardless of relevance—potentially leading to overfitting. Adjusted R-squared compensates for this by penalizing model complexity, providing a more balanced view of fit quality. The goal of `olr` is to identify the most suitable model that captures the underlying structure of the data while avoiding unnecessary complexity. By comparing both metrics, it offers a robust evaluation framework that balances predictive power with model parsimony. Example Analogy: Imagine a gardener trying to understand what influences plant growth (the dependent variable). They might consider variables like sunlight, watering frequency, soil type, and nutrients (independent variables). Instead of manually guessing which combination works best, the `olr` function automatically tests every possible combination of predictors and identifies the most effective model—based on either the highest R-squared or adjusted R-squared value. This saves the user from trial-and-error modeling and highlights only the most meaningful variables for explaining the outcome.

Usage

```
olr(dataset, responseName = NULL, predictorNames = NULL, adjr2 = TRUE)
```

```
olrmodels(dataset, responseName = NULL, predictorNames = NULL)
```

```
olrformulas(dataset, responseName = NULL, predictorNames = NULL)
```

```
olrformulasorder(dataset, responseName = NULL, predictorNames = NULL)
```

```
adjr2list(dataset, responseName = NULL, predictorNames = NULL)
```

```
r2list(dataset, responseName = NULL, predictorNames = NULL)
```

Arguments

<code>dataset</code>	is defined by the user and points to the name of the dataset that is being used.
<code>responseName</code>	the response variable name defined as a string. For example, it represents a header in the data table.
<code>predictorNames</code>	the predictor variable or variables that are the terms that are to be regressed against the <code>responseName</code> . Place desired headers from the dataset in here as a character vector.
<code>adjr2</code>	<code>adjr2 = TRUE</code> returns the regression summary for the maximum adjusted R-squared term. <code>adjr2 = FALSE</code> returns the regression summary for the maximum R-squared term.

Details

Complementary functions below follow the format: `function(dataset, responseName = NULL, predictorNames = NULL)`

olrmodels: Returns the list of all evaluated models. Use `summary(olrmodels(dataset, responseName, predictorNames)[, x])` to inspect a specific model, where `x` is the model index.

olrformulas: Returns the list of all regression formulas generated by `olr()`, each representing a unique combination of specified predictor variables regressed on the dependent variable, in the

order created.

olrformulasorder: Returns the same set of regression formulas as `olrformulas`, but sorted alphabetically by variable names within each formula. This helps users more easily locate or compare specific combinations of predictors.

adjr2list: Returns adjusted R-squared values for all models.

r2list: Returns R-squared values for all models.

Tip: To avoid errors from non-numeric columns (e.g., dates), remove them using `dataset <- dataset[, -1]`. Or use `load_custom_data(..., exclude_first_column = TRUE)`.

When `responseName` and `predictorNames` are `NULL`, the function will treat the first column of the dataset as the response variable and all remaining columns as predictors. **If the first column contains non-numeric or irrelevant data (e.g., a Date column), you must exclude it manually:** `dataset <- crudeoildata[, -1]`.

Otherwise, you can utilize `load_custom_data(data = "crudeoildata.csv", custom_path = NULL, exclude_first_column = TRUE)`, a custom function that allows you to load the data (`crudeoildata`) automatically without the first column.

Value

Returns the best-fitting linear model object based on either adjusted R-squared (default) or R-squared. Call `summary()` on the result to view full regression statistics.

Examples

```
# Please allow time for rendering after clicking "Run Examples"
crudeoildata <- read.csv(system.file("extdata", "crudeoildata.csv", package = "olr"))
dataset <- crudeoildata[, -1]

responseName <- 'CrudeOil'
predictorNames <- c('RigCount', 'API', 'FieldProduction', 'RefinerNetInput',
  'OperableCapacity', 'Imports', 'StocksExcludingSPR', 'NonCommercialLong',
  'NonCommercialShort', 'CommercialLong', 'CommercialShort', 'OpenInterest')

olr(dataset, responseName, predictorNames, adjr2 = TRUE)
```

Index

- * **adjusted**
 - olr, 3
- * **combination**
 - olr, 3
- * **linear-model**
 - olr, 3
- * **linear**
 - olr, 3
- * **model-selection**
 - olr, 3
- * **model**
 - olr, 3
- * **olr**
 - olr, 3
- * **optimal**
 - olr, 3
- * **r-squared**
 - olr, 3
- * **regression**
 - olr, 3

adjr2list (olr), 3

load_custom_data, 2

olr, 3

olrformulas (olr), 3

olrformulasorder (olr), 3

olrmodels (olr), 3

r2list (olr), 3