Package: sqlm (via r-universe)

October 6, 2026

Type Package

Title SQL-Backed Linear Regression

Version 1.0.0

Description Fits linear regression models on datasets residing in SQL databases without pulling data into R memory. Computes sufficient statistics inside the database engine via a single aggregation query and solves the normal equations in R.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, dbplyr, DBI, glue, purrr, S7, MASS, broom, tibble, stats, utils

Suggests testthat (>= 3.0.0), duckdb, orbital, withr, knitr, rmarkdown, quarto, mirai, parallel

Config/testthat/edition 3

RoxygenNote 7.3.3

Config/pak/sysreqs libicu-dev

Repository https://usrbinr.r-universe.dev

Date/Publication 2026-02-05 12:18:40 UTC

RemoteUrl https://github.com/usrbinr/sqlm

RemoteRef HEAD

RemoteSha 21e6f2ecdd07fd3f218e3f9084e0b9896a098687

Contents

glance.lm_sql_result 2
lm_sql 2
orbital.lm_sql_result 3
print.lm_sql_result 4
tidy.lm_sql_result 5

Index **6**

glance.lm_sql_result *Glance at an lm_sql_result*

Description

Extract a single-row tibble of model-level summary statistics from a fitted SQL linear model.

Usage

```
## S3 method for class 'lm_sql_result'
glance(x, ...)
```

Arguments

x	An ‘lm_sql_result‘ object.
...	Not used.

Details

Returns R-squared, adjusted R-squared, residual standard error, F-statistic and its p-value, model degrees of freedom, log-likelihood, AIC, BIC, number of observations, and residual degrees of freedom.

Value

A single-row tibble with columns ‘r.squared‘, ‘adj.r.squared‘, ‘sigma‘, ‘statistic‘, ‘p.value‘, ‘df‘, ‘logLik‘, ‘AIC‘, ‘BIC‘, ‘nobs‘, and ‘df.residual‘.

lm_sql *SQL-Backed Linear Regression*

Description

Fits a linear regression model using SQL aggregation on a remote database table. The data never leaves the database — only sufficient statistics (sums and cross-products) are returned to R.

Usage

```
lm_sql(formula, data, tol = 1e-07)
```

Arguments

formula	A formula object (e.g., price ~ x + cut).
data	A tbl_sql object (from **dbplyr**).
tol	Tolerance for detecting linear dependency.

Details

The function computes the $X^T X$ and $X^T y$ matrices entirely inside the database engine via a single SQL aggregation query, then solves the normal equations in R using Cholesky decomposition (falling back to Moore-Penrose pseudoinverse for rank-deficient designs).

Supported formula features:

- Numeric and categorical (character/factor) predictors with automatic dummy encoding via ‘CASE WHEN‘.
- Interaction terms (‘*‘ and ‘:‘) including numeric × categorical and categorical × categorical cross-products.
- Dot expansion (‘y ~ .‘) to all non-response columns.
- Transforms: ‘I()‘, ‘log()‘, and ‘sqrt()‘ translated to SQL equivalents (‘POWER‘, ‘LN‘, ‘SQRT‘).
- Date and datetime predictors automatically cast to numeric in SQL.
- No-intercept models (‘y ~ 0 + x‘).

For grouped data (via [dplyr::group_by()]), a single ‘GROUP BY‘ query is executed and one model per group is returned in a tibble with a ‘model‘ list-column.

NA handling uses listwise deletion: rows with ‘NULL‘ in any model variable are excluded via a ‘WHERE ... IS NOT NULL‘ clause.

Value

An S7 object of class `lm_sql_result`, or a tibble with a `model` list-column if the data is grouped.

`orbital.lm_sql_result` *Convert an lm_sql_result to an orbital object*

Description

Creates an orbital object from a fitted SQL linear model, enabling in-database predictions without pulling data into R.

Usage

```
orbital.lm_sql_result(x, ..., prefix = ".pred")
```

Arguments

`x`	An ‘lm_sql_result‘ object.
`...`	Not used.
`prefix`	Column name for predictions. Defaults to ‘".pred"‘.

Details

Builds a single prediction expression by combining the fitted coefficients with the R expressions stored in ‘term_expressions‘. For categorical predictors, the expression includes ‘ifelse()‘ calls that dbplyr translates to SQL ‘CASE WHEN‘. The resulting ‘orbital_class‘ object can be used with [orbital::predict()] to get predictions or [orbital::augment()] to append a ‘.pred‘ column to a database table.

Value

An ‘orbital_class‘ object.

`print.lm_sql_result` *Print an lm_sql_result*

Description

Display a concise summary of a fitted SQL linear model.

Usage

```
## S3 method for class 'lm_sql_result'
print(x, ...)
```

Arguments

`x`	An ‘lm_sql_result‘ object.
`...`	Not used.

Details

Prints the original function call and the named coefficient vector.

Value

Invisibly returns ‘x‘.

`tidy.lm_sql_result` *Tidy an lm_sql_result*

Description

Extract a tidy tibble of per-term coefficient statistics from a fitted SQL linear model.

Usage

```
## S3 method for class 'lm_sql_result'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

Arguments

`x`	An ‘lm_sql_result‘ object.
`conf.int`	Logical. If ‘TRUE‘, include confidence interval columns ‘conf.low‘ and ‘conf.high‘. Defaults to ‘FALSE‘.
`conf.level`	Confidence level for the interval. Defaults to ‘0.95‘.
`...`	Not used.

Details

Returns one row per model term with the estimate, standard error, t-statistic, and p-value. When ‘conf.int = TRUE‘, confidence intervals are computed using the t-distribution with ‘df_residual‘ degrees of freedom.

Value

A tibble with columns ‘term‘, ‘estimate‘, ‘std.error‘, ‘statistic‘, and ‘p.value‘. If ‘conf.int = TRUE‘, also ‘conf.low‘ and ‘conf.high‘.

Index

glance.lm_sql_result, 2

lm_sql, 2

orbital.lm_sql_result, 3

print.lm_sql_result, 4

tidy.lm_sql_result, 5