

Oracle R Enterprise: New Features in Release 1.3

- ✓ **New *ore.predict* package** allows you to use an existing R model to score data that is in a database table entirely in SQL. Supported Models include *lm()* (linear regression models), *glm()* (generalized linear models), *hclust()* (hierarchical clustering), *kmeans()* (k-Means clustering), *negbin()* (*glm.nb* - negative binomial generalized binomial models), *nnet::multinom* (multinomial log-linear model), *nnet::nnet* (neural network models) and *rpart::rpart* (recursive partitioning and regression tree models).
- ✓ **New In-Database Predictive Models package *OREdm*** designed to provide a standard R interface for in-Database predictive analytics and data mining functions. It supports the following Oracle Data Mining models: Attribute Importance, Decision Tree, Generalized Linear Models, k-Means, Naive Bayes and Support Vector Machine. Models created using *OREdm* are transient objects in the database, but can be saved and persisted, imported or exported to be accessed by other tools like the Oracle Data Mining GUI in SQL Developer or the PL/SQL language.
- ✓ **New Oracle R Enterprise support for persistence of R objects in the database** provides database storage to save and restore R and Oracle R Enterprise objects (for example, a model) across different R sessions. Specific use cases include ability to use and score predictive models in database embedded R execution and to enable application integration. ORE interfaces such as *ore.save()*, *ore.load()*, *ore.delete()*, *ore.datastore()* and *ore.datastoreSummary()* are available to interact with the data store in the database.
- ✓ **New support for all Oracle Date and Time data formats** and analytic capabilities that allow date arithmetic, aggregations, percentile calculations and moving window calculations to execute in-database. Collectively these analytic capabilities are critical for time series data analysis and data preparation enroute to using R's algorithms for time series forecasting.
- ✓ **New support for high-performance in-Database sampling and partitioning techniques:** A variety of sampling techniques to use against database resident data. Specifically, Simple random sampling, Systematic sampling, Stratified sampling, Cluster sampling, Quota sampling and Accidental sampling are supported.
- ✓ **New support for Long Names and Columns** allows for handling of R naming conventions for *ore.frame* columns, instead of the more restrictive Database names. The *ore.frame* column names can now be longer than 30 bytes, contain double quotes, and be non-unique.
- ✓ **New Embedded Execution Auto Connect for R Scripts** automatically enables database connectivity inside embedded R scripts, with the following features: Database Embedded R scripts can automatically connect to the database, using the same credentials as the session that invokes the embedded R SQL functions. Auto-connect is off by default, controlled by an *ore.connect* control argument.
- ✓ **New Neural Network Model: *ore.neural()*** builds a single layer feed-forward neural network for regression on Database-resident data, and uses the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method to solve the underlying unconstrained nonlinear optimization problem that results from fitting a neural network.
- ✓ **New Ordering and Indexing options** that allows better control in bridging the R semantics of ordered data frames with the database semantics of ordering via primary keys.