**CS 634 Final Term Project**

-Alfred Zane Rajan Velladurai

UCID: ar2257

Email address: [ar2257@njit.edu](mailto:ar2257@njit.edu)

**Option no: 1**

Dataset:  https://www.kaggle.com/mlg-ulb/creditcardfraud#creditcard.csv

Classification methods used: Linear Regression and SVM models in R

**Tools:**

R Studio, R packages

**My Code:**

library(ISLR)

library(e1071)

library(caret)

setwd("~/Data Science/Sem 3/Data Mining/Final Project")

MyData <- read.csv(file="creditcardfraud/creditcard.csv", header=TRUE, sep=",")

set.seed(10) # setting seed to reproduce results of random sampling

trainingRowIndex <- sample(1:nrow(MyData), 0.7\*nrow(MyData)) # row indices for training data

trainingData <- MyData[trainingRowIndex, ]

testData <- MyData[-trainingRowIndex, ]

linearMod <- lm(Class ~ V3+V4+V7+V9+V10+V11+V12+V14+V16+V17+V18, data=trainingData)

Pred <- predict(linearMod, testData)

Pred[Pred<0.1] = 0

Pred[Pred>0] = 1

sum(Pred==testData[31])/length(Pred)

confusionMatrix(as.factor(Pred), as.factor(testData$Class))

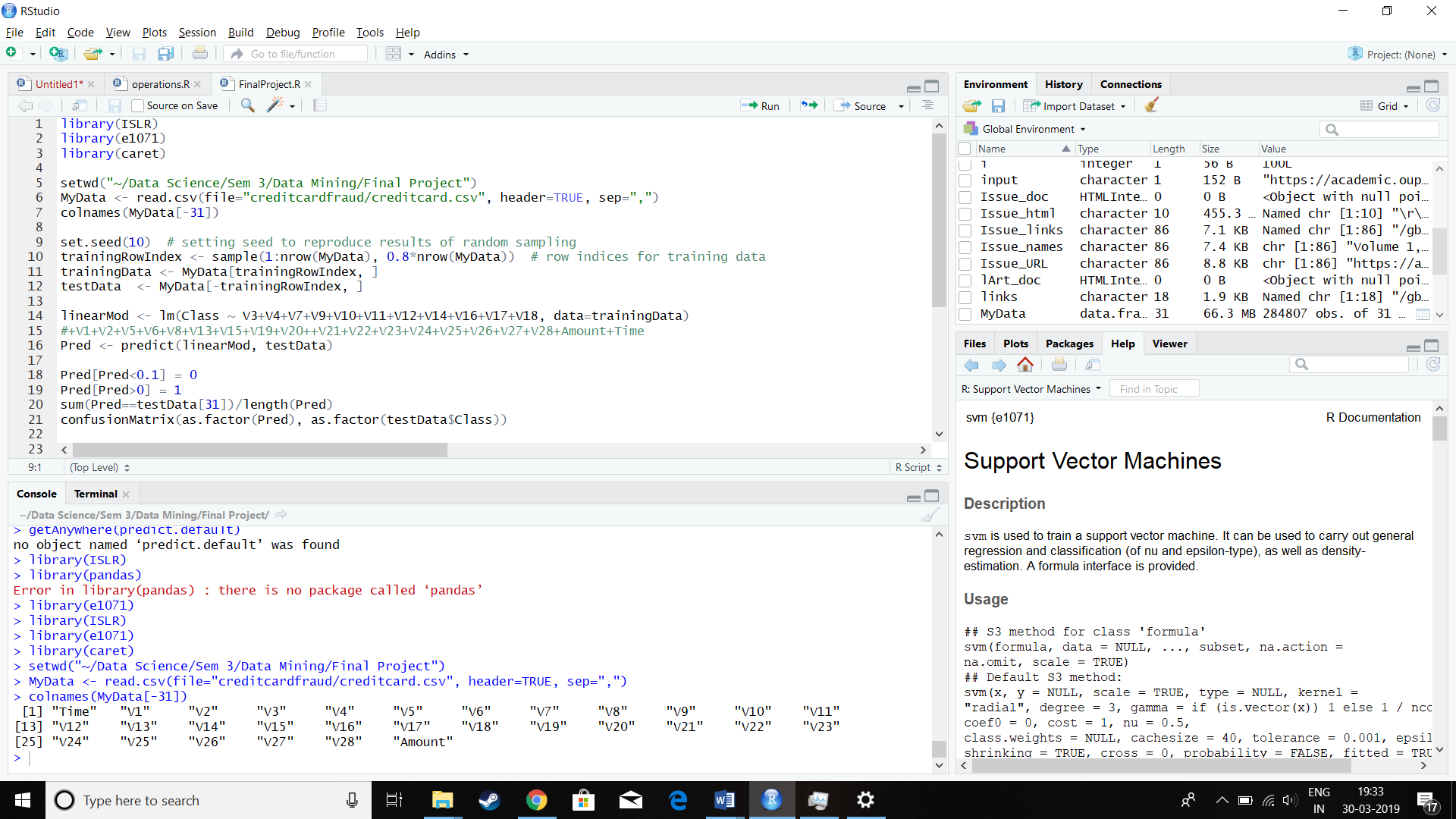
trainingData$Class = as.factor(trainingData$Class)

SVMmod <- svm(trainingData[c(4,5,8,10,11,12,13,15,17,18,19)], trainingData$Class, type = "C" ,kernel = "linear", tolerance = 0.01)

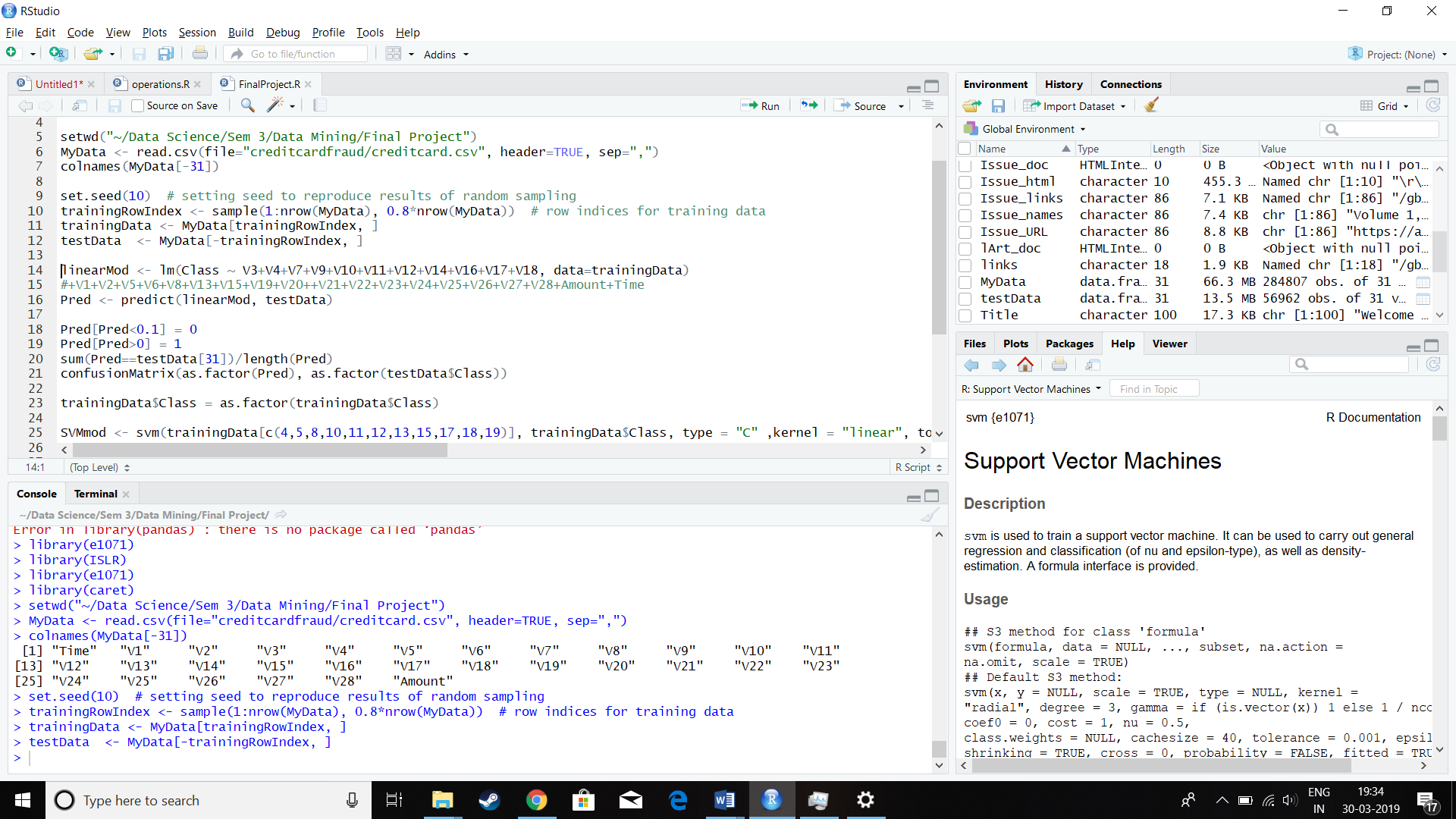
Pred1 <- predict(SVMmod, testData[c(4,5,8,10,11,12,13,15,17,18,19)])

confusionMatrix(Pred1, as.factor(testData$Class))

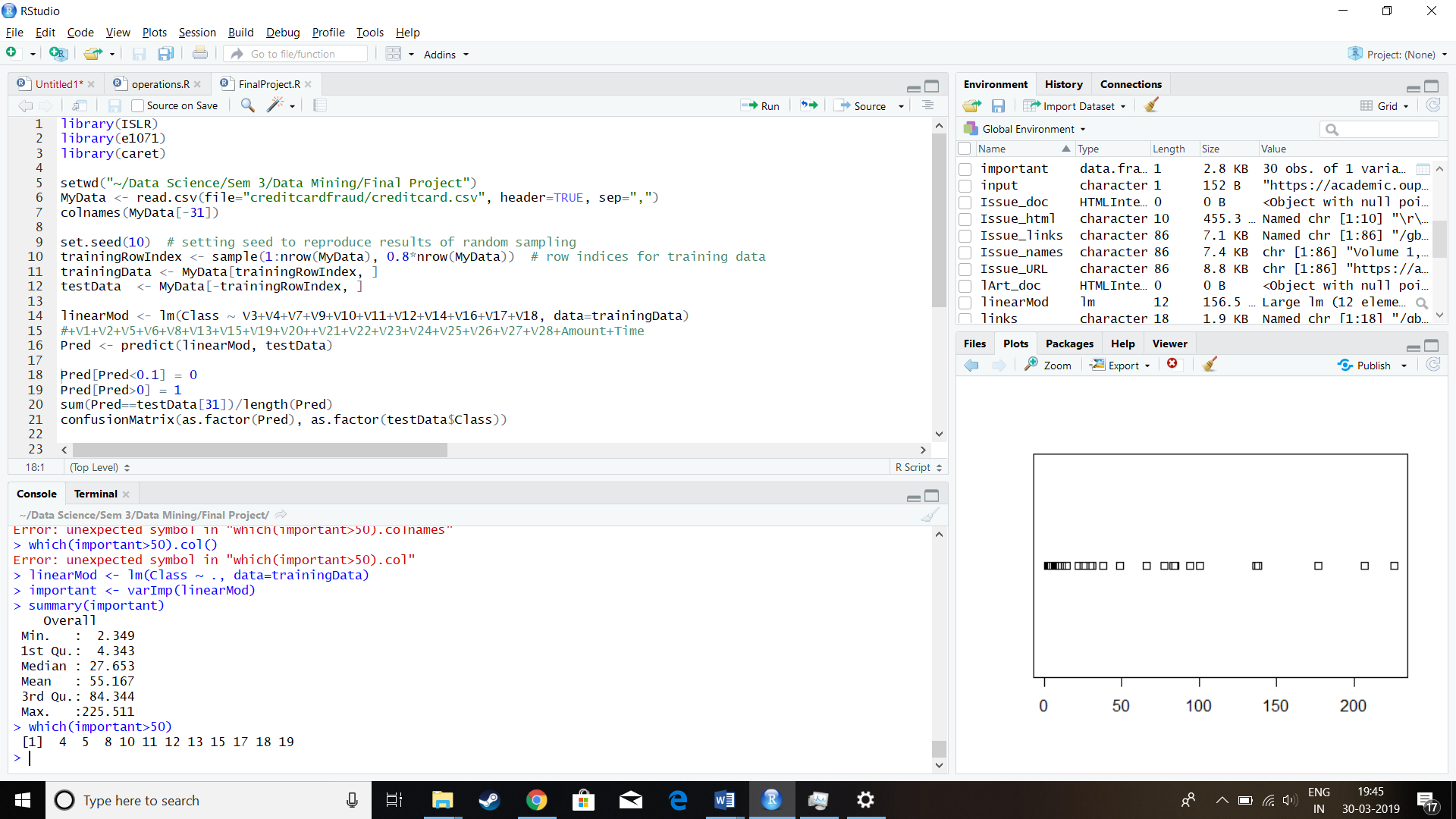
**Step by step:**



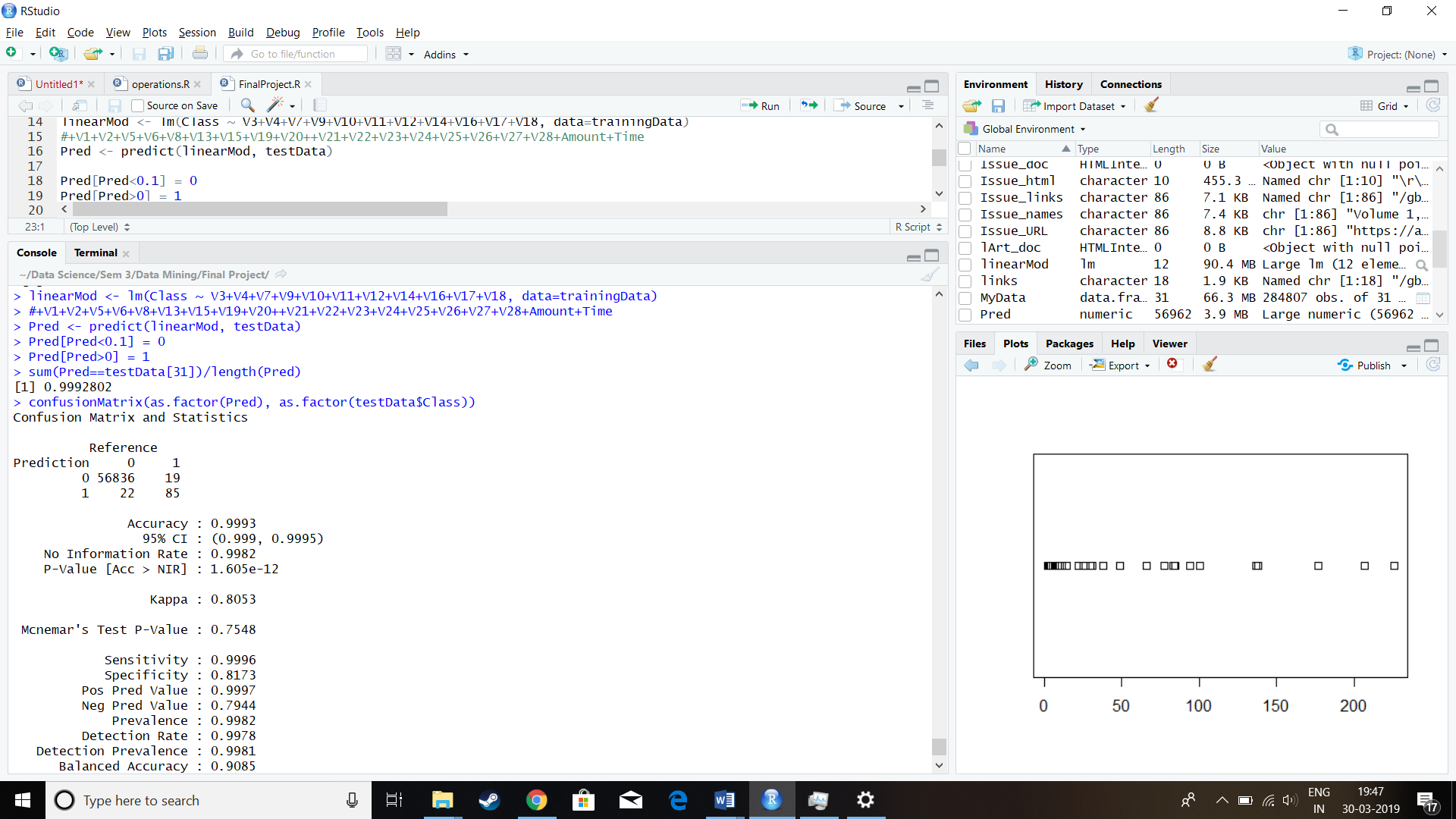
Reading data



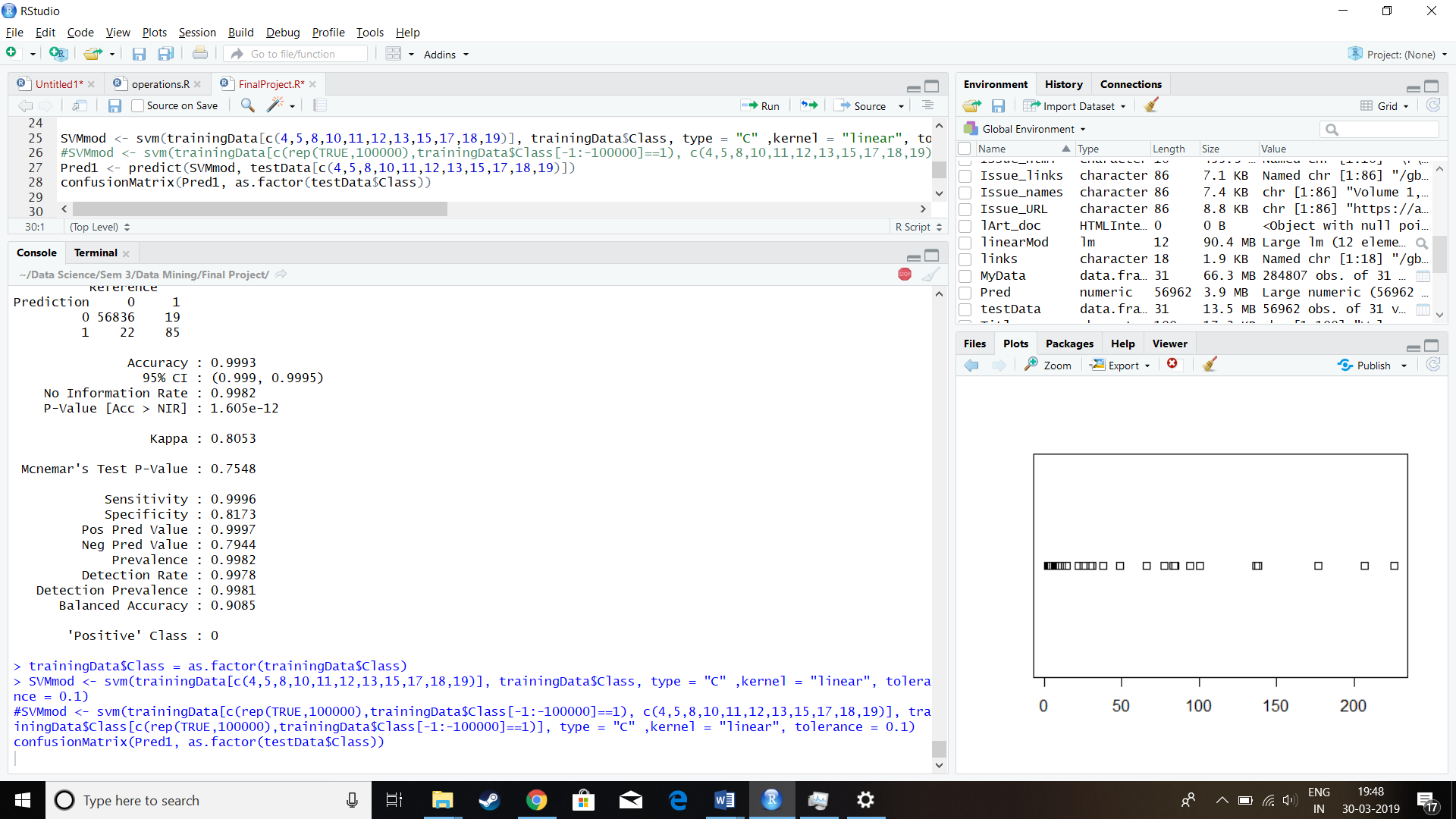
Splitting Dataset



Feauture Selection



Linear Model



SVM

**Description:**

The Data was read into a Data Frame.

It was then split into train and test data with a 70:30 ratio.

Linear model was first built with all the features.

Then, varImp(linearMod) was used to find the important features.

summary(varImp(linearMod))

Overall

Min. : 2.349

1st Qu.: 4.343

Median : 27.653

Mean : 55.167

3rd Qu.: 84.344

Max. :225.511

The mean(~50) was used to qualify the feature as important. And only features with importance above 50 was selected.

**Linear Regeression:**

The model was built again with the selected features and predictions were made on the test set.

The prediction output being a continuous regression was converted to classification by selecting a split.

Initially 0.5 was used as the split(>=0.5 being Class 1 and <0.5 being Class 0)

> confusionMatrix(as.factor(Pred), as.factor(testData$Class))

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 56850 59

1 8 45

In this case, although it had a high accuracy(99.88%), its specificity was low as the dataset had far more datapoints with Class 0 than Class 1.

So different splits were tried.

> Pred[Pred<0.2] = 0

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 85290 27

1 21 105

> Pred[Pred<0.1] = 0

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 85288 26

1 23 106

> Pred[Pred<0.05] = 0

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 85267 25

1 44 107

The balanced accuracy was the same in all these cases(~90%) but we have to choose between misclassifying less frauds as genuine and misclassifying less genuine transactions as frauds.

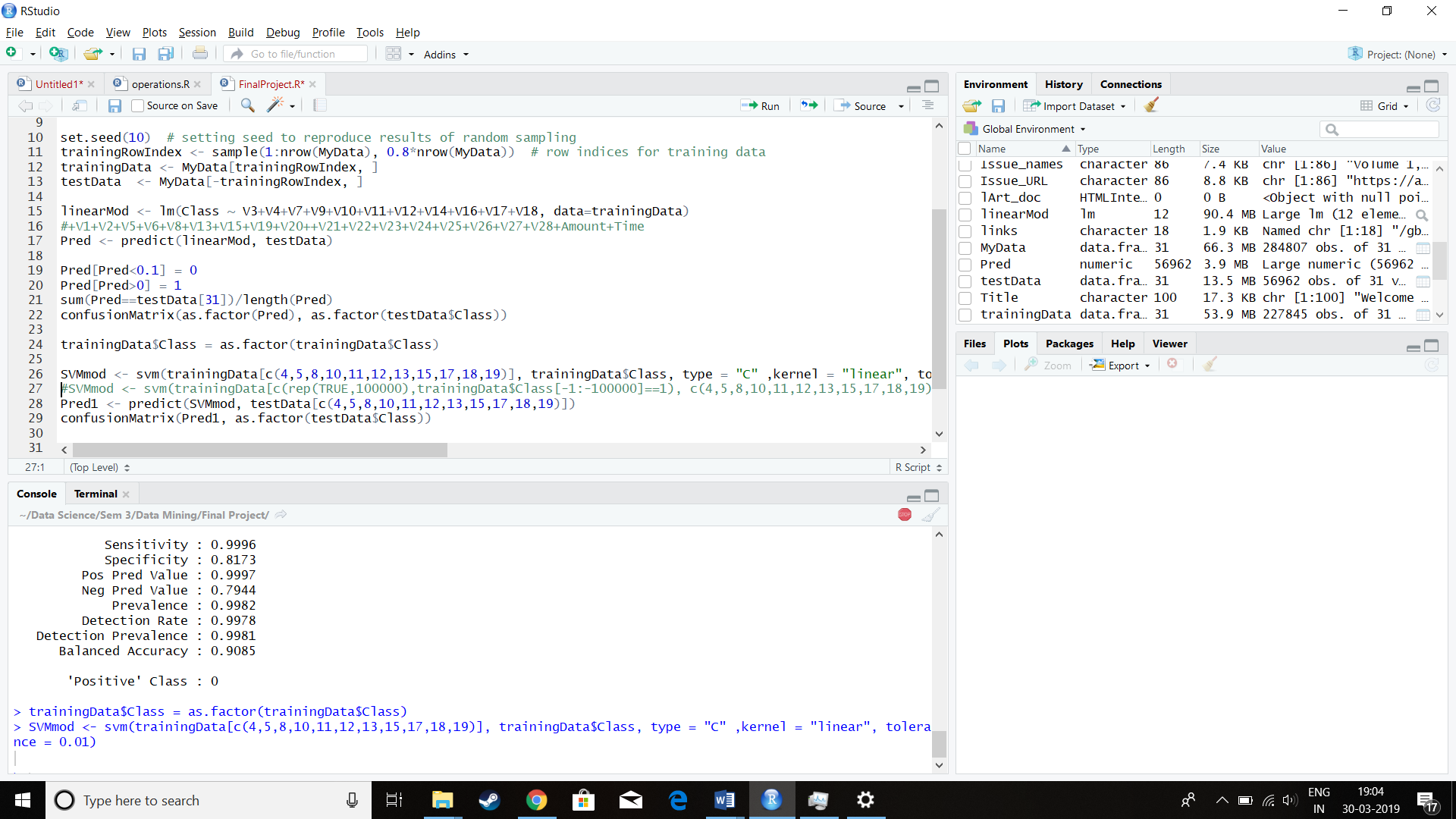
**SVM:**

Now in SVM we can set the model itself to give the output as classification instead of regression by giving either a factor as response variable or setting the type = “C”.

Although in case of SVM we have to decide the tolerance value which will decide how much it reduces the error but also exponentially increases the time taken to train the model.

First SVM was tried with 0.1 tolerance which trained the model for several minutes and yet during predictions, the balanced accuracy was 88%, slightly lesser than linear regression.

Next, 0.01 tolerance was tried which took a few hours.

But the balanced accuracy improved to 90%.

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 142128 57

1 30 189

Accuracy : 0.9994

95% CI : (0.9992, 0.9995)

No Information Rate : 0.9983

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.8126

Mcnemar's Test P-Value : 0.005312

Sensitivity : 0.9998

Specificity : 0.7683

Pos Pred Value : 0.9996

Neg Pred Value : 0.8630

Prevalence : 0.9983

Detection Rate : 0.9981

Detection Prevalence : 0.9985

Balanced Accuracy : 0.8840

The confusion matrix tells that the model focuses on catching more fraud transactions at the cost of misclassifying genuine transactions.

**Conclusion:**

Lesser tolerances give better accuracies with longer run times. So choosing between Linear regression and SVM is simply choosing between efficiency and accuracy.

**Source codes for the tools:**

**Linear regression:**

**lm():** <https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R>

*#' @include utilities.R*

*#' @include data\_prep.R*

*#' @include FLTable.R*

[**NULL**](https://rdrr.io/r/base/NULL.html)

*## move to file datamining.R*

*#' An S4 class to represent FLLinRegr*

*#'*

*#' @slot offset this can be used to specify a priori known component to be included in the*

*#' linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases*

*#' An S4 class to represent objects returned by Data-Mining functions*

*#'*

*#' @slot deeptable A character vector containing a deeptable (either conversion from a*

*#' widetable or input deeptable)*

*#' @slot AnalysisID An output character ID from call to data-mining function*

*#' @slot wideToDeepAnalysisID An output character ID from FLRegrDataPrep*

*#' @slot mapTable A character string name for the mapping table in-database if input is wide-table, generated by FLRegrDataPrep*

*#' @slot results cache list of results computed*

*#' @slot table Input data object*

*#' @export*

[setClass](https://rdrr.io/r/methods/setClass.html)("FLDataMining",

slots=[list](https://rdrr.io/r/base/list.html)(AnalysisID="character",

wideToDeepAnalysisID="character",

[table](https://rdrr.io/r/base/table.html)="FLTable",

results="list",

deeptable="FLTable",

mapTable="character"))

*#' An S4 class to represent objects returned by Regression Functions*

*#'*

*#' @slot formula an object of class 'formula': Model Formula*

*#' @slot scoreTable Name of the in-database table where scoring results are stored*

*#' @export*

[setClass](https://rdrr.io/r/methods/setClass.html)("FLRegr",

contains="FLDataMining",

slots=[list](https://rdrr.io/r/base/list.html)([formula](https://rdrr.io/r/stats/formula.html)="formula",

scoreTable="character",

RegrDataPrepSpecs="list"))

*#' An S4 class to represent output from Linear Regression(lm) on in-database Objects*

*#'*

*#' @slot offset column name used as offset*

*#' @slot vfcalls information about system tables*

*#' @method print FLLinRegr*

*#' @method coefficients FLLinRegr*

*#' @method residuals FLLinRegr*

*#' @method influence FLLinRegr*

*#' @method lm.influence FLLinRegr*

*#' @method plot FLLinRegr*

*#' @method summary FLLinRegr*

*#' @method predict FLLinRegr*

*#' @export*

[setClass](https://rdrr.io/r/methods/setClass.html)(

"FLLinRegr",

contains="FLRegr",

slots=[list](https://rdrr.io/r/base/list.html)([offset](https://rdrr.io/r/stats/offset.html)="character",

vfcalls="character"))

*#' @export*

[setClass](https://rdrr.io/r/methods/setClass.html)(

"FLLinRegrMD",

contains="FLLinRegr")

*#' @export*

[setClass](https://rdrr.io/r/methods/setClass.html)(

"FLLinRegrSF",

contains="FLRegr",

slots=[list](https://rdrr.io/r/base/list.html)([offset](https://rdrr.io/r/stats/offset.html)="character",

vfcalls="character"))

*#' @export*

[setClass](https://rdrr.io/r/methods/setClass.html)(

"FLLogRegrSF",

contains="FLRegr",

slots=[list](https://rdrr.io/r/base/list.html)([offset](https://rdrr.io/r/stats/offset.html)="character",

vfcalls="character"))

*#' @export*

[setClass](https://rdrr.io/r/methods/setClass.html)(

"FLRobustRegr",

contains="FLLinRegr")

*#' Robust Regression.*

*#'*

*#' performs robust regression*

*#' @examples*

*#' Example for deeptbl:*

*#' library(MASS)*

*#' options(debugSQL =TRUE)*

*#' table <- FLTable(getTestTableName("tblRobustRegr"), "ObsID","VarID", "Num\_Val")*

*#' flmod <- rlm(a~., data = table)*

*#' predict(flmod)*

*#' residuals(flmod)*

*#' flmod$fitted.values*

*#' summary(flmod)*

*#' @section Constraints:*

*#' plot method not supported*

*#' Example for widetable:*

*#' widetbl <- FLTable(getTestTableName("tblautompg"), "ObsID")*

*#' flmod <- rlm(Weight~ Acceleration , data = widetbl)*

*#' summary(flmod)*

*#' coefficients(flmod)*

*#' residuals(flmod)*

*#' @export*

[rlm](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/rlm.html) <- [**function**](https://rdrr.io/r/base/function.html) ([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html)=[list](https://rdrr.io/r/base/list.html)(),psi, [**...**](https://rdrr.io/r/base/dots.html)) {

[UseMethod](https://rdrr.io/r/base/UseMethod.html)("rlm", [data](https://rdrr.io/r/utils/data.html))

}

*## move to file rlm.R*

*#' @export*

rlm.default <- MASS::[rlm](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/rlm.html)

*## move to file rlm.R*

*#' @export*

[rlm.FLpreparedData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html),psi = "psi.huber", [**...**](https://rdrr.io/r/base/dots.html))

{

vcallObject <- [match.call](https://rdrr.io/r/base/match.call.html)()

[**return**](https://rdrr.io/r/base/function.html)([lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)([formula](https://rdrr.io/r/stats/formula.html)=[formula](https://rdrr.io/r/stats/formula.html),

[data](https://rdrr.io/r/utils/data.html)=[data](https://rdrr.io/r/utils/data.html),

callObject=vcallObject,

familytype="robust",

psi = psi,

[**...**](https://rdrr.io/r/base/dots.html)))

}

*## move to file rlm.R*

*#' @export*

rlm.FLTable <- [rlm.FLpreparedData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*## move to file rlm.R*

*#' @export*

rlm.FLTableMD <- [rlm.FLpreparedData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*#' @export*

rlm.FLTableDeep <- [rlm.FLpreparedData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*## move to file lm.R*

*#' Linear Regression.*

*#'*

*#' \code{lm} performs linear regression on FLTable objects.*

*#'*

*#' The DB Lytix function called is FLLinRegr. Performs Linear Regression and*

*#' stores the results in predefined tables.*

*#'*

*#' @seealso \code{\link[stats]{lm}} for R reference implementation.*

*#' @param formula A symbolic description of model to be fitted*

*#' @param data An object of class FLTable or FLTableMD*

*#' @param catToDummy Transform categorical variables to numerical values*

*#' either using dummy variables or by using Empirical*

*#' Logit. If the value is 1, transformation is done using*

*#' dummy variables, else if the value is 0,*

*#' transformation is done using Empirical Logit.*

*#' @param performNorm 0/1 indicating whether to perform standardization of data.*

*#' @param performVarReduc 0/1. If the value is 1,*

*#' the stored procedure eliminates variables based on standard deviation and*

*#' correlation.*

*#' @param makeDataSparse If 0,Retains zeroes and NULL values*

*#' from the input table. If 1, Removes zeroes and NULL. If 2,Removes zeroes*

*#' but retains NULL values.*

*#' @param minStdDev Minimum acceptable standard deviation for*

*#' elimination of variables. Any variable that has a*

*#' standard deviation below this threshold is*

*#' eliminated. This parameter is only consequential if*

*#' the parameter PerformVarReduc = 1. Must be >0.*

*#' @param maxCorrel Maximum acceptable absolute correlation between*

*#' a pair of columns for eliminating variables. If the*

*#' absolute value of the correlation exceeds this*

*#' threshold, one of the columns is not transformed.*

*#' Again, this parameter is only consequential if the*

*#' parameter PerformVarReduc = 1. Must be >0 and <=1.*

*#' @param classSpec list describing the categorical dummy variables.*

*#' @param whereconditions takes the where\_clause as a string.*

*#' @section Constraints:*

*#' The anova method is not yet available for FLLinRegr*

*#' If \code{data} is FLTableMD, only single formula is accepted.*

*#' So input deeptable or deeptable produced after data preparation*

*#' should have same VarIDs'.*

*#' For FLTableMD data object, only coefficients and summary*

*#' methods are defined.Predict method on \code{FLTableMD}*

*#' \code{newdata} is not supported.*

*#' Properties like \code{print(x),model,plot} might take time as they*

*#' have to fetch data*

*#' @return \code{lm} returns an object of class \code{FLLinRegr}*

*#' @examples*

*#' widetable <- FLTable(getTestTableName("tblAbaloneWide"), "ObsID")*

*#' lmfit <- lm(Rings~Height+Diameter,widetable)*

*#' lmfit$coefficients*

*#' lmfit$fitted.values*

*#' plot(lmfit)*

*#' mu <- predict(lmfit,newdata=widetable)*

*#' deeptable <- FLTable(getTestTableName("myLinRegrSmall"),"ObsID","VarID","Num\_Val")*

*#' lmfit <- lm(NULL,deeptable)*

*#' summary(lmfit)*

*#' flMDObject <- FLTableMD(table=getTestTableName("tblAutoMPGMD"),*

*#' group\_id\_colname="GroupID",*

*#' obs\_id\_colname="ObsID",group\_id = c(2,4))*

*#' vformula <- MPG~HorsePower+Displacement+Weight+Acceleration*

*#' lmfit <- lm(vformula,*

*#' data=flMDObject)*

*#' coeffList <- coef(lmfit)*

*#' summaryList <- summary(lmfit)*

*#' @export*

[lm](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/lm.html) <- [**function**](https://rdrr.io/r/base/function.html) ([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html)=[list](https://rdrr.io/r/base/list.html)(),[**...**](https://rdrr.io/r/base/dots.html)) {

[UseMethod](https://rdrr.io/r/base/UseMethod.html)("lm", [data](https://rdrr.io/r/utils/data.html))

}

*## move to file lm.R*

*#' @export*

lm.default <- [stats](https://rdrr.io/r/stats/stats-package.html)::[lm](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/lm.html)

*## move to file lm.R*

*#' @export*

[lm.FLpreparedData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html),[**...**](https://rdrr.io/r/base/dots.html))

{

vcallObject <- [match.call](https://rdrr.io/r/base/match.call.html)()

[**return**](https://rdrr.io/r/base/function.html)([lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)([formula](https://rdrr.io/r/stats/formula.html)=[formula](https://rdrr.io/r/stats/formula.html),

[data](https://rdrr.io/r/utils/data.html)=[data](https://rdrr.io/r/utils/data.html),

callObject=vcallObject,

[**...**](https://rdrr.io/r/base/dots.html)))

}

*## move to file lm.R*

*#' @export*

[lm.FLTable](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html),[**...**](https://rdrr.io/r/base/dots.html))

{

vcallObject <- [match.call](https://rdrr.io/r/base/match.call.html)()

[data](https://rdrr.io/r/utils/data.html) <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)([data](https://rdrr.io/r/utils/data.html),"")

[**return**](https://rdrr.io/r/base/function.html)([lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)([formula](https://rdrr.io/r/stats/formula.html)=[formula](https://rdrr.io/r/stats/formula.html),

[data](https://rdrr.io/r/utils/data.html)=[data](https://rdrr.io/r/utils/data.html),

callObject=vcallObject,

[**...**](https://rdrr.io/r/base/dots.html)))

}

*#' @export*

lm.FLTableMD <- [lm.FLTable](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*## move to file step.R*

*#' Choose a model.*

*#'*

*#' \code{steps} performs linear regression on FLTable objects.*

*#' Choose a formula based model by p-values and R-Squared Values.*

*#'*

*#' @seealso \code{\link[stats]{step}} for R reference implementation.*

*#'*

*#' @param object An object of class FLTable*

*#' @param scope A symbolic description of model to be fitted.*

*#' \code{scope} can be a list with upper and lower components*

*#' or a formula. For a widetable, upper and lower should be formulas*

*#' describing the range of models. If a formula is given instead of list*

*#' it will be treated as upper. For a deeptable, upper and lower should*

*#' be vectors with variable ids'.Provide empty list for deeptable if*

*#' nothing is to be specified.*

*#' @param scale currently not used.*

*#' @param direction character.Must be one of backward,*

*#' Fbackward,UFbackward,forward.*

*#' @param trace if positive, information is printed out during the*

*#' running of the steps.*

*#' @param catToDummy Transform categorical variables to numerical values*

*#' either using dummy variables or by using Empirical*

*#' Logit. If the value is 1, transformation is done using*

*#' dummy variables, else if the value is 0,*

*#' transformation is done using Empirical Logit.*

*#' @param performNorm 0/1 indicating whether to perform standardization of data.*

*#' @param performVarReduc 0/1. If the value is 1,*

*#' the stored procedure eliminates variables based on standard deviation and*

*#' correlation.*

*#' @param makeDataSparse If 0,Retains zeroes and NULL values*

*#' from the input table. If 1, Removes zeroes and NULL. If 2,Removes zeroes*

*#' but retains NULL values.*

*#' @param minStdDev Minimum acceptable standard deviation for*

*#' elimination of variables. Any variable that has a*

*#' standard deviation below this threshold is*

*#' eliminated. This parameter is only consequential if*

*#' the parameter PerformVarReduc = 1. Must be >0.*

*#' @param maxCorrel Maximum acceptable absolute correlation between*

*#' a pair of columns for eliminating variables. If the*

*#' absolute value of the correlation exceeds this*

*#' threshold, one of the columns is not transformed.*

*#' Again, this parameter is only consequential if the*

*#' parameter PerformVarReduc = 1. Must be >0 and <=1.*

*#' @param classSpec list describing the categorical dummy variables.*

*#' @param whereconditions takes the where\_clause as a string.*

*#' @param highestpAllow1 All the variables whose p-value exceed the value*

*#' specified by HighestpAllow1 are dropped in one go.*

*#' Typical value for HighestProbAllow1 could be 0.50. Must be >0 and < 1.*

*#' Not applicable for forward.*

*#' @param highestpAllow2 Only one variable is dropped at a time*

*#' till all the p-Values are below the HighestpAllow2.*

*#' Typical value could be 0.10. Must be >0 and < 1.*

*#' Not applicable for forward and backward.*

*#' @param stepWiseDecrease The StepwiseDecrease is used to*

*#' decrease the p-Value at each stage. In first step,*

*#' all variables having pValue exceeding HighestpValue1 are*

*#' dropped. Then the HighestpValue1 is*

*#' reduced by StepwiseDecreasepValue*

*#' and the process is repeated until all*

*#' the variables have p-value less than HighestpValue2.*

*#' Must be >0 and <1. Used only for UFbackward.*

*#' @section Constraints:*

*#' The anova method is not yet available for FLLinRegr.*

*#' Properties like \code{print(fit$x),model,plot} might take time as they*

*#' have to fetch data*

*#'*

*#' @return \code{step} performs linear regression and replicates equivalent R output.*

*#' @examples*

*#' widetable <- FLTable(getTestTableName("tblAbaloneWide"), "ObsID")*

*#' s <- step(widetable,*

*#' scope=list(lower=Rings~Height+Diameter),*

*#' direction = "UFbackward")*

*#' plot(s)*

*#' s$coefficients*

*#' s <- step(widetable,*

*#' scope=list(lower=Rings~Height+Diameter,*

*#' upper=Rings~Height+Diameter+Sex+Num\_Length),*

*#' direction = "UFbackward")*

*#' plot(s)*

*#' s$coefficients*

*#' s <- step(widetable,*

*#' scope=list(lower=Rings~Num\_Length),*

*#' direction = "UFbackward",*

*#' performNorm=1,performVarReduc=1,maxCorrel=0.6)*

*#' plot(s)*

*#' s$coefficients*

*#' s <- step(widetable,*

*#' scope=list(upper=Rings~Height+Diameter+Sex+Num\_Length+DummyCat),*

*#' direction = "Fbackward")*

*#' plot(s)*

*#' s$coefficients*

*#' s <- step(widetable,*

*#' scope=Rings~Height+Diameter+Sex+Num\_Length+DummyCat,*

*#' direction = "forward")*

*#' plot(s)*

*#' s$coefficients*

*#' s <- step(widetable,*

*#' scope=Rings~Height+Diameter+Sex+Num\_Length+DummyCat,*

*#' direction = "Fbackward")*

*#' plot(s)*

*#' s$coefficients*

*#' s <- step(widetable,*

*#' scope=list(upper=Rings~Height+Diameter+Sex+Num\_Length+DummyCat),*

*#' direction = "forward")*

*#' plot(s)*

*#' s$coefficients*

*#' deeptable <- FLTable(getTestTableName("myLinRegrSmall"),"ObsID","VarID","Num\_Val")*

*#' s <- step(deeptable,*

*#' scope=list(upper=c("-1","0","1")),*

*#' direction = "backward")*

*#' s <- step(deeptable,*

*#' scope=list(upper=c("1","2"),lower=c("1")),*

*#' direction = "Fbackward")*

*#' s <- step(deeptable,*

*#' scope=list(lower=c("2")),*

*#' direction = "UFbackward")*

*#' s <- step(deeptable,*

*#' scope=list(),*

*#' direction = "forward")*

*#' deeptable1 <- FLTable(getTestTableName("tblLogRegr"),*

*#' "ObsID","VarID","Num\_Val",*

*#' whereconditions=c("ObsID < 7001","VarID<5"))*

*#' s <- step(deeptable1,*

*#' scope=list(lower=c("2")),*

*#' direction = "UFbackward",familytype = "logistic")*

*#' s <- step(deeptable1,*

*#' scope=list(),*

*#' direction = "forward",familytype="logistic")*

*#' plot(s)*

*#' s <- step(deeptable1,*

*#' scope=list(upper=c("-1","0","1","2","3")),*

*#' direction = "backward",*

*#' familytype="multinomial",pRefLevel=1)*

*#' s <- step(deeptable1,*

*#' scope=list(upper=c("1","2","3"),lower=c("2")),*

*#' direction = "Fbackward",familytype="multinomial",pRefLevel=1)*

*#' deeptable2 <- FLTable(getTestTableName("tblLogRegrMN10000"),*

*#' "ObsID","VarID","Num\_Val",*

*#' whereconditions=c("ObsID < 7001","VarID<5"))*

*#' s <- step(deeptable2,*

*#' scope=list(lower=c("2")),*

*#' direction = "UFbackward",familytype = "multinomial",pRefLevel=1)*

*#' summary(s)*

*#' @export*

[step](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/step.html) <- [**function**](https://rdrr.io/r/base/function.html) (object,scope,[**...**](https://rdrr.io/r/base/dots.html)){

[UseMethod](https://rdrr.io/r/base/UseMethod.html)("step", object)

}

*## move to file step.R*

*#' @export*

step.default <- [stats](https://rdrr.io/r/stats/stats-package.html)::[step](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/step.html)

*## move to file step.R*

*#' @export*

[step.FLTable](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object, scope, [scale](https://rdrr.io/r/base/scale.html) = 0,

direction = "forward",

[trace](https://rdrr.io/r/base/trace.html) = 1,

familytype="linear",

[**...**](https://rdrr.io/r/base/dots.html)){

[**if**](https://rdrr.io/r/base/Control.html) (!direction %in% [c](https://rdrr.io/r/base/c.html)("forward","Fbackward","backward","UFbackward","sf"))

[stop](https://rdrr.io/r/base/stop.html)("direction must be in c(forward,Fbackward,backward,UFbackward)")

[**if**](https://rdrr.io/r/base/Control.html)(![is.list](https://rdrr.io/r/base/list.html)(scope) && ![class](https://rdrr.io/r/base/class.html)(scope)=="formula")

[stop](https://rdrr.io/r/base/stop.html)("scope argument must be a list or formula.\n",

" empty list accepted for deeptable.\n")

[**if**](https://rdrr.io/r/base/Control.html)(!familytype %in% [c](https://rdrr.io/r/base/c.html)("linear","logistic","multinomial","linearSF","logisticSF"))

[stop](https://rdrr.io/r/base/stop.html)("familytype argument must be one of linear,logistic or multinomial",

"in step.FLTable\n")

[**if**](https://rdrr.io/r/base/Control.html)(familytype=="multinomial" && direction=="forward")

[stop](https://rdrr.io/r/base/stop.html)("forward not supported in multinomial logistic regr currently")

vupperformula <- ""

[**if**](https://rdrr.io/r/base/Control.html)([class](https://rdrr.io/r/base/class.html)(scope)=="formula")

{

[**if**](https://rdrr.io/r/base/Control.html)([isDotFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(scope))

scope <- [genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(pColnames=[setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object),

getVariables(object)[["obs\_id\_colname"]]),

pDepColumn=[ifelse](https://rdrr.io/r/base/ifelse.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object),

[**NULL**](https://rdrr.io/r/base/NULL.html),

[all.vars](https://rdrr.io/r/base/allnames.html)(scope)[1]))

vupperformula <- scope

}

object <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)(object,"")

vinclude <- [c](https://rdrr.io/r/base/c.html)()

vexclude <- [c](https://rdrr.io/r/base/c.html)()

[**if**](https://rdrr.io/r/base/Control.html)([is.list](https://rdrr.io/r/base/list.html)(scope))

{*#browser()*

vlower <- scope[["lower"]]

vupper <- scope[["upper"]]

*## To account for . in formula*

[**if**](https://rdrr.io/r/base/Control.html)([isDotFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(vlower))

vlower <- [genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(pColnames=[setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object),

getVariables(object)[["obs\_id\_colname"]]),

pDepColumn=[ifelse](https://rdrr.io/r/base/ifelse.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object),

[**NULL**](https://rdrr.io/r/base/NULL.html),

[all.vars](https://rdrr.io/r/base/allnames.html)(vlower)[1]))

[**if**](https://rdrr.io/r/base/Control.html)([isDotFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(vupper))

vupper <- [genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(pColnames=[setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object),

getVariables(object)[["obs\_id\_colname"]]),

pDepColumn=[ifelse](https://rdrr.io/r/base/ifelse.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object),

[**NULL**](https://rdrr.io/r/base/NULL.html),

[all.vars](https://rdrr.io/r/base/allnames.html)(vupper)[1]))

*##If only lower is given. Upper includes all.*

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(vupper) && ![is.null](https://rdrr.io/r/base/NULL.html)(vlower)){

[**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object)){

[**if**](https://rdrr.io/r/base/Control.html)([class](https://rdrr.io/r/base/class.html)(vlower)!="formula") [stop](https://rdrr.io/r/base/stop.html)("for wide table scope should have formula as components\n")

vupperformula <- [formula](https://rdrr.io/r/stats/formula.html)([paste0](https://rdrr.io/r/base/paste.html)([all.vars](https://rdrr.io/r/base/allnames.html)(vlower)[1],"~",

[paste0](https://rdrr.io/r/base/paste.html)([setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object),

[c](https://rdrr.io/r/base/c.html)([all.vars](https://rdrr.io/r/base/allnames.html)(vlower)[1],

getVariables(object)[["obs\_id\_colname"]])),

collapse="+")))

vinclude <- [all.vars](https://rdrr.io/r/base/allnames.html)(vlower)[2:[length](https://rdrr.io/r/base/length.html)([all.vars](https://rdrr.io/r/base/allnames.html)(vlower))]

}

[else](https://rdrr.io/r/base/Control.html){

[**if**](https://rdrr.io/r/base/Control.html)(![is.vector](https://rdrr.io/r/base/vector.html)(vlower)) [stop](https://rdrr.io/r/base/stop.html)("for deep table scope should have vectors as components\n")

vinclude <- vlower

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(vlower) && ![is.null](https://rdrr.io/r/base/NULL.html)(vupper)){

[**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object)){

[**if**](https://rdrr.io/r/base/Control.html)([class](https://rdrr.io/r/base/class.html)(vupper)!="formula") [stop](https://rdrr.io/r/base/stop.html)("for wide table scope should have formula as components\n")

vupperformula <- vupper

}

[else](https://rdrr.io/r/base/Control.html){

[**if**](https://rdrr.io/r/base/Control.html)(![is.vector](https://rdrr.io/r/base/vector.html)(vupper)) [stop](https://rdrr.io/r/base/stop.html)("for deep table scope should have vectors as components\n")

vexclude <- [setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object),vupper)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(vupper) && ![is.null](https://rdrr.io/r/base/NULL.html)(vlower)){

[**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object)){

[**if**](https://rdrr.io/r/base/Control.html)([class](https://rdrr.io/r/base/class.html)(vupper)!="formula" || [class](https://rdrr.io/r/base/class.html)(vlower)!="formula")

[stop](https://rdrr.io/r/base/stop.html)("for wide table scope should have formula as components\n")

vupperformula <- vupper

vinclude <- [all.vars](https://rdrr.io/r/base/allnames.html)(vlower)[2:[length](https://rdrr.io/r/base/length.html)([all.vars](https://rdrr.io/r/base/allnames.html)(vlower))]

}

[else](https://rdrr.io/r/base/Control.html){

[**if**](https://rdrr.io/r/base/Control.html)(![is.vector](https://rdrr.io/r/base/vector.html)([class](https://rdrr.io/r/base/class.html)(vupper)) || ![is.vector](https://rdrr.io/r/base/vector.html)([class](https://rdrr.io/r/base/class.html)(vlower)))

[stop](https://rdrr.io/r/base/stop.html)("for deep table scope should have vectors as components\n")

vinclude <- vlower

vexclude <- [setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object),vupper)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object)) [stop](https://rdrr.io/r/base/stop.html)("scope cannot be empty list for widetable")

}

vinclude <- [setdiff](https://rdrr.io/r/base/sets.html)(vinclude,[c](https://rdrr.io/r/base/c.html)("-1"))

vexclude <- [setdiff](https://rdrr.io/r/base/sets.html)(vexclude,[c](https://rdrr.io/r/base/c.html)("0","-1"))

[**if**](https://rdrr.io/r/base/Control.html)(![length](https://rdrr.io/r/base/length.html)(vinclude)>0) vinclude <- [**NULL**](https://rdrr.io/r/base/NULL.html)

[**if**](https://rdrr.io/r/base/Control.html)(![length](https://rdrr.io/r/base/length.html)(vexclude)>0) vexclude <- [**NULL**](https://rdrr.io/r/base/NULL.html)

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(vinclude) || ![is.null](https://rdrr.io/r/base/NULL.html)(vexclude))

specID <- [list](https://rdrr.io/r/base/list.html)(include=vinclude,

exclude=vexclude)

[**else**](https://rdrr.io/r/base/Control.html) specID <- [list](https://rdrr.io/r/base/list.html)()

vcallObject <- [match.call](https://rdrr.io/r/base/match.call.html)()

[**return**](https://rdrr.io/r/base/function.html)([lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)([formula](https://rdrr.io/r/stats/formula.html)=vupperformula,

[data](https://rdrr.io/r/utils/data.html)=object,

callObject=vcallObject,

familytype=familytype,

specID=specID,

direction=direction,

[trace](https://rdrr.io/r/base/trace.html)=[trace](https://rdrr.io/r/base/trace.html),

[**...**](https://rdrr.io/r/base/dots.html)))

}

*## move to file lmGeneric.R*

[lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html),

callObject=[**NULL**](https://rdrr.io/r/base/NULL.html),

familytype="linear",

specID=[list](https://rdrr.io/r/base/list.html)(),

direction="",

[trace](https://rdrr.io/r/base/trace.html)=1,

[**...**](https://rdrr.io/r/base/dots.html))

{

[**if**](https://rdrr.io/r/base/Control.html)([inherits](https://rdrr.io/r/base/class.html)([data](https://rdrr.io/r/utils/data.html),"FLTable"))

prepData <- [prepareData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html),

callObject=callObject,

familytype=familytype,

specID=specID,

direction=direction,

[trace](https://rdrr.io/r/base/trace.html)=[trace](https://rdrr.io/r/base/trace.html),

[**...**](https://rdrr.io/r/base/dots.html))

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([inherits](https://rdrr.io/r/base/class.html)([data](https://rdrr.io/r/utils/data.html),"FLpreparedData")){

prepData <- [data](https://rdrr.io/r/utils/data.html)

[data](https://rdrr.io/r/utils/data.html) <- prepData$wideTable

}

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) [names](https://rdrr.io/r/base/names.html)(prepData))

[assign](https://rdrr.io/r/base/assign.html)(i,prepData[[i]])

deepx <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)(deepx,"")

deeptable <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(deepx)

*#for more generic output:*

mod <- [c](https://rdrr.io/r/base/c.html)(FLCoeffCorrelWithRes="CORRELWITHRES",

FLCoeffNonZeroDensity="NONZERODENSITY",

FLCoeffTStat="TSTAT",

FLCoeffStdErr="STDERR",

FLCoeffPValue="PVALUE",

nCoeffEstim = "COEFFVALUE",

nID = "COEFFID"

)

*## todo: create a list for this lookup*

[**if**](https://rdrr.io/r/base/Control.html)(familytype=="linear"){

[**if**](https://rdrr.io/r/base/Control.html)(direction=="sf") vfcalls<-[c](https://rdrr.io/r/base/c.html)(functionName="FLLinRegrSF",

infotableName="fzzlLinRegrInfo",

note="SingleFactorLinRegr",

coefftablename="fzzlLinRegrCoeffs",

statstablename="fzzlLinRegrStats")

[else](https://rdrr.io/r/base/Control.html)

vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName=[ifelse](https://rdrr.io/r/base/ifelse.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([data](https://rdrr.io/r/utils/data.html)),

"FLLinRegrMultiDataSet",

"FLLinRegr"),

infotableName="fzzlLinRegrInfo",

Note="linregr",

coefftablename="fzzlLinRegrCoeffs",

statstablename="fzzlLinRegrStats",

valcolnamescoretable="Y",

scoretablename="FLLinRegrScore")}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype=="logistic"){

[**if**](https://rdrr.io/r/base/Control.html)(direction=="sf") vfcalls<-[c](https://rdrr.io/r/base/c.html)(functionName="FLLogRegrSF",

infotableName="fzzlLogRegrInfo",

note="SingleFactorLogRegr",

coefftablename="fzzlLogRegrCoeffsSF",

statstablename="fzzlLogRegrStatsSF")

[else](https://rdrr.io/r/base/Control.html)

vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName=[ifelse](https://rdrr.io/r/base/ifelse.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([data](https://rdrr.io/r/utils/data.html)),

"FLLogRegrMultiDataSet",

"FLLogRegr"),

infotableName="fzzlLogRegrInfo",

Note="logregr",

coefftablename="fzzlLogRegrCoeffs",

statstablename="fzzlLogRegrStats",

valcolnamescoretable="Y",

scoretablename="FLLogRegrScore")}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype=="poisson") vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName="FLPoissonRegr",

infotableName="fzzlPoissonRegrInfo",

Note="poissonregr",

coefftablename="fzzlPoissonRegrCoeffs",

statstablename="fzzlPoissonRegrStats",

valcolnamescoretable="Mu",

scoretablename="FLPoissonRegrScore")

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype=="logisticwt"){

vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName="FLLogRegrWt",

infotableName="fzzlLogRegrInfo",

Note="logregrwt",

coefftablename="fzzlLogRegrCoeffs",

statstablename="fzzlLogRegrStats",

valcolnamescoretable="Y",

scoretablename="FLLogRegrScore")

*# vtemp <- pThreshold*

*# pThreshold <- maxiter*

*# maxiter <- vtemp*

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype=="multinomial") vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName="FLLogRegrMN",

infotableName="fzzlLogRegrMNInfo",

Note="logregrMN",

coefftablename="fzzlLogRegrMNCoeffs",

statstablename="fzzlLogRegrMNStats",

valcolnamescoretable="Y",

scoretablename="FLLogRegrScore")

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype == "robust") vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName="FLRobustRegr",

infotableName="fzzlRobustRegrInfo",

Note="robustregr",

coefftablename="fzzlRobustRegrCoeffs",

statstablename="fzzlRobustRegrStats",

cortablename = "fzzlRobustRegrVarCov",

scoretablename="FLLinRegrScore"

)

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype == "pls") vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName="FLPLSRegr",

infotableName="fzzlPLSRegrInfo",

note="plsregr",

coefftablename="fzzlPLSRegrCoeffs",

statstablename="fzzlPLSRegrConvVec",

scoretablename="FLLinRegrScore"

)

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype == "opls") vfcalls <- [c](https://rdrr.io/r/base/c.html)(functionName="FLOPLSRegr",

infotableName="fzzlPLSRegrnfo",

Note="oplsregr",

coefftablename="fzzlPLSRegrCentCoeffs",

statstablename="fzzlOPLSRegrConvVec",

scoretablename="FLLinRegrScore",

rcoeff = "fzzlOPLSRegrFactorFit"

)

functionName <- vfcalls["functionName"]

infotableName <- vfcalls["infotableName"]

vnote <- [genNote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(vfcalls["note"])

coefftablename <- vfcalls["coefftablename"]

statstablename <- vfcalls["statstablename"]

vinputCols <- [list](https://rdrr.io/r/base/list.html)()

[**if**](https://rdrr.io/r/base/Control.html)(functionName %in% [c](https://rdrr.io/r/base/c.html)("FLLinRegrMultiDataSet",

"FLLogRegrMultiDataSet"))

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

TableName=deeptable,

GroupIDCol=getGroupIdSQLExpression(deepx),

ObsIDCol=getObsIdSQLExpression(deepx),

VarIDCol=getVarIdSQLExpression(deepx),

ValueCol=getValueSQLExpression(deepx)

)

[**else**](https://rdrr.io/r/base/Control.html) vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

TableName=deeptable,

ObsIDCol=getObsIdSQLExpression(deepx),

VarIDCol=getVarIdSQLExpression(deepx),

ValueCol=getValueSQLExpression(deepx)

)

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% [c](https://rdrr.io/r/base/c.html)("multinomial"))

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

pRefLevel=pThreshold)

[**if**](https://rdrr.io/r/base/Control.html)(!familytype %in% [c](https://rdrr.io/r/base/c.html)("linear", "robust", "pls", "opls") && direction!="forward")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

MaxIterations=maxiter)

[**if**](https://rdrr.io/r/base/Control.html)([base](https://rdrr.io/r/base/base-package.html)::[grepl](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/grepl.html)("logistic",familytype)

&& direction!="forward")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

pThreshold=pThreshold)

[**if**](https://rdrr.io/r/base/Control.html)(direction==""){

vfuncName=functionName

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% "logisticwt"){

vinputCols <- [as.list](https://rdrr.io/r/base/list.html)(vinputCols)

vinputCols[["MaxIterations"]] <- [**NULL**](https://rdrr.io/r/base/NULL.html)

vinputCols <- [unlist](https://rdrr.io/r/base/unlist.html)(vinputCols)

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

MaxIterations=maxiter,

EVENTWEIGHT=eventweight,

NONEVENTWEIGHT=noneventweight)

}

}

[**if**](https://rdrr.io/r/base/Control.html)(direction %in% [c](https://rdrr.io/r/base/c.html)("backward","Fbackward","UFbackward")){

vfuncName <- [paste0](https://rdrr.io/r/base/paste.html)(functionName,"BW")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

SPECID=vspecID,

HIGHESTPALLOW1=highestpAllow1)

}

[**if**](https://rdrr.io/r/base/Control.html)(direction %in% [c](https://rdrr.io/r/base/c.html)("Fbackward","UFbackward")){

vfuncName <- [paste0](https://rdrr.io/r/base/paste.html)(functionName,"FB")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

HIGHESTPALLOW2=highestpAllow2)

}

[**if**](https://rdrr.io/r/base/Control.html)(direction %in% [c](https://rdrr.io/r/base/c.html)("UFbackward")){

vfuncName <- [paste0](https://rdrr.io/r/base/paste.html)(functionName,"UFB")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

STEPWISEDECREASE=stepWiseDecrease)

}

[**if**](https://rdrr.io/r/base/Control.html)(direction %in% [c](https://rdrr.io/r/base/c.html)("forward")){

vfuncName <- [paste0](https://rdrr.io/r/base/paste.html)(functionName,"SW")

[**if**](https://rdrr.io/r/base/Control.html)(!familytype %in% "linear")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

pThreshold=pThreshold)

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% "logistic")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

MaxIterations=maxiter)

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

TOPN=topN,

HIGHESTPALLOW1=highestpAllow1)

}

[**if**](https://rdrr.io/r/base/Control.html)(direction %in% "sf"){

vfuncName <- [paste0](https://rdrr.io/r/base/paste.html)(functionName)

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

MaxIterations=maxiter,

pThreshold=pThreshold)

}

*##for rlm defining psi and tuning constant:*

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% "robust")

{

weightfn = "huber"

[**if**](https://rdrr.io/r/base/Control.html)([list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$psi == "psi.bisquare" )

{weightfn <- "bisquare"}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$psi == "psi.hampel")

[print](https://rdrr.io/r/base/print.html)("dont compute rlm for hampel function currently computing it for huber")

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$psi %in% [c](https://rdrr.io/r/base/c.html)("cauchy", "fair","logistic", "talwar", "andrews", "welsch")

)

weightfn <- [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$psi

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)([list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$u))

{

tunconst <- .5

}

[else](https://rdrr.io/r/base/Control.html)

tunconst <- [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$u

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

WeightFn = weightfn,

TuneConstant= tunconst,

MaxIterations =maxiter

)

functionName <- "FLRobustRegr"

mod <- [c](https://rdrr.io/r/base/c.html)(FLCoeffCorrelWithRes="",

FLCoeffNonZeroDensity="",

FLCoeffTStat="T\_VAL",

FLCoeffStdErr="STDDEV",

FLCoeffPValue="P\_VAL",

nCoeffEstim = "EST",

nID = "VARID"

) }

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% "pls")

{

functionName <- "FLLinRegr"

[**if**](https://rdrr.io/r/base/Control.html)(![list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$nfactor )

{[print](https://rdrr.io/r/base/print.html)([list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$nfactor)

nfactor <- 15}

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

NumOfFactors = [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$nfactor)

mod <- [c](https://rdrr.io/r/base/c.html)(mod, ncomp = [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$nfactor)

}

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% "opls")

{

functionName <- "FLLinRegr"

[**if**](https://rdrr.io/r/base/Control.html)(![list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$nfactor)

{[print](https://rdrr.io/r/base/print.html)("Number of Component is missing insterting default value of 4 ")

nfactor <- 4

}

[**if**](https://rdrr.io/r/base/Control.html)(![list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$Northo)

{

[print](https://rdrr.io/r/base/print.html)("Number of Ortho is missing insterting default value of 3")

northo <- 3

}

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

NumOfFactors = [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$nfactor,

NumOfOrtho = [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$Northo)

mod <- [c](https://rdrr.io/r/base/c.html)(mod, ncomp = [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$nfactor, northo = [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$Northo)

}

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

Note=vnote)

retobj <- [sqlStoredProc](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlStoredProc.html)(getFLConnection(),

vfuncName,

outputParameter=[c](https://rdrr.io/r/base/c.html)(AnalysisID="a"),

pInputParams=vinputCols

)

retobj <- [checkSqlQueryOutput](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(retobj)

AnalysisID <- [as.character](https://rdrr.io/r/base/character.html)(retobj[1,1])

*##Find the max modelID to avoid joins later.*

*##For forward find best fit model id.*

vmaxModelID <- [**NULL**](https://rdrr.io/r/base/NULL.html)

vmaxLevelID <- [**NULL**](https://rdrr.io/r/base/NULL.html)

[**if**](https://rdrr.io/r/base/Control.html)(direction=="" && familytype!="poisson"

&&![is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([data](https://rdrr.io/r/utils/data.html))){

vmaxModelID <- 1

vmaxLevelID <- 1

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(!direction %in% [c](https://rdrr.io/r/base/c.html)("forward","sf") && familytype!="poisson" && ![is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([data](https://rdrr.io/r/utils/data.html))){

vsqlstr <- [paste0](https://rdrr.io/r/base/paste.html)("SELECT MAX(ModelID) AS modelid",

[ifelse](https://rdrr.io/r/base/ifelse.html)(familytype=="multinomial",",MAX(LevelID) AS levelid ",""),

" FROM ",coefftablename," WHERE AnalysisID=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID))

vtemp <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),vsqlstr)

vmaxModelID <- vtemp[["modelid"]]

vmaxLevelID <- vtemp[["levelid"]]

}

[**if**](https://rdrr.io/r/base/Control.html)([trace](https://rdrr.io/r/base/trace.html)>0 && !direction %in% [c](https://rdrr.io/r/base/c.html)("","forward","sf"))

{

*# vsqlstr <- paste0("SELECT a.coeffid,c.\* \n",*

*# " FROM ",coefftablename," a,",statstablename," c \n",*

*# " WHERE NOT EXISTS(SELECT 1 FROM ",coefftablename," b ",*

*# " WHERE b.analysisid=a.analysisid AND b.modelid=a.modelid+1 \n",*

*# " AND a.coeffid = b.coeffid ",ifelse(!is.null(vmaxLevelID),*

*# " AND a.LevelID = b.LevelID ",""),")\n",*

*# " AND a.analysisid=",fquote(AnalysisID)," AND c.analysisid=a.analysisid \n",*

*# " AND a.modelid<>",vmaxModelID," AND c.modelid=a.modelid\n",*

*# ifelse(!is.null(vmaxLevelID),paste0(" AND a.LevelID = ",vmaxLevelID),""),*

*# " \n UNION ALL\n",*

*# " SELECT 0,a.\* FROM ",statstablename," a \n",*

*# " WHERE a.AnalysisID=",fquote(AnalysisID),*

*# " AND a.ModelID=",vmaxModelID,"\n",*

*# " ORDER BY 3")*

vsqlstr <- [paste0](https://rdrr.io/r/base/paste.html)("SELECT a.coeffid,c.\* \n",

" FROM (SELECT DISTINCT AnalysisID,modelid,coeffid from ",coefftablename,

" WHERE analysisid=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)," \n EXCEPT \n ",

" SELECT DISTINCT AnalysisID,modelid+1,coeffid from ",coefftablename,

" WHERE analysisid=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)," \n ",

") a,",statstablename," c \n",

" WHERE c.analysisid=a.analysisid \n"

)

d <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),vsqlstr)

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(d)<-[toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(d))

d[["ANALYSISID"]] <- [**NULL**](https://rdrr.io/r/base/NULL.html)

vdroppedCols <- [c](https://rdrr.io/r/base/c.html)()

[**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)([data](https://rdrr.io/r/utils/data.html)))vdroppedCols <- specID[["exclude"]]

[**if**](https://rdrr.io/r/base/Control.html)([nrow](https://rdrr.io/r/base/nrow.html)(d)>1){

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) [unique](https://rdrr.io/r/base/unique.html)([setdiff](https://rdrr.io/r/base/sets.html)(d[["MODELID"]],vmaxModelID)))

{

[**if**](https://rdrr.io/r/base/Control.html)(familytype=="linear")

[cat](https://rdrr.io/r/base/cat.html)("Step: RSQUARED = ",d[d[,"MODELID"]==i,"RSQUARED"][1],"\n")

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype=="logistic")

[cat](https://rdrr.io/r/base/cat.html)("Step: Gini Coefficient = ",d[d[,"MODELID"]==i,"GINICOEFF"][1],"\n")

*#browser()*

vdropped <- [as.numeric](https://rdrr.io/r/base/numeric.html)(d[d[,"MODELID"]==i,"COEFFID"])

vcolnames <- [names](https://rdrr.io/r/base/names.html)(vmapping)

vdroppedCols1 <- [sapply](https://rdrr.io/r/base/lapply.html)(vdropped,[**function**](https://rdrr.io/r/base/function.html)(x) vcolnames[[as.numeric](https://rdrr.io/r/base/numeric.html)(vmapping)==x])

[**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)([data](https://rdrr.io/r/utils/data.html)))vdroppedCols <- [c](https://rdrr.io/r/base/c.html)(vdroppedCols1,vdroppedCols)

[**if**](https://rdrr.io/r/base/Control.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)([data](https://rdrr.io/r/utils/data.html))){

vallVars <- [all.vars](https://rdrr.io/r/base/allnames.html)([formula](https://rdrr.io/r/stats/formula.html))

vfr <- [genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)([c](https://rdrr.io/r/base/c.html)(vdropped))

vdroppedCols <- [c](https://rdrr.io/r/base/c.html)(vdroppedCols,[all.vars](https://rdrr.io/r/base/allnames.html)(vfr)[-1])

vdroppedCols1 <- [all.vars](https://rdrr.io/r/base/allnames.html)([genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(specID[["exclude"]]))[-1]

vcolnames <- vallVars[!vallVars %in% vdroppedCols1]

}

[cat](https://rdrr.io/r/base/cat.html)(vallVars[1],"~",[paste0](https://rdrr.io/r/base/paste.html)(vcolnames[![toupper](https://rdrr.io/r/base/chartr.html)(vcolnames) %in% [c](https://rdrr.io/r/base/c.html)([toupper](https://rdrr.io/r/base/chartr.html)(vdroppedCols)

,[toupper](https://rdrr.io/r/base/chartr.html)(vallVars[1]))],

collapse=" + "),"\n")

vdataframe <- [rbind](https://rdrr.io/r/base/cbind.html)(d[d[,"MODELID"]==i,][1,],d[d[,"MODELID"]==i+1,][1,])

[rownames](https://rdrr.io/r/base/colnames.html)(vdataframe) <- [c](https://rdrr.io/r/base/c.html)(" - None",[paste0](https://rdrr.io/r/base/paste.html)(" - ",[paste0](https://rdrr.io/r/base/paste.html)(vdroppedCols,collapse=" + ")))

[print](https://rdrr.io/r/base/print.html)(vdataframe[,![colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vdataframe) %in% [c](https://rdrr.io/r/base/c.html)("COEFFID","BPSTAT","SIGBPSTAT")])

[cat](https://rdrr.io/r/base/cat.html)("\n\n\n")

}

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(direction %in% [c](https://rdrr.io/r/base/c.html)("forward"))

{

[**if**](https://rdrr.io/r/base/Control.html)(familytype=="linear")

vsqlstr <- [limitRowsSQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)([paste0](https://rdrr.io/r/base/paste.html)("SELECT a.\*,b.maxPValue \n",

" FROM ",statstablename," a,( \n",

" SELECT a.ModelID,",

" MAX(a.PValue) AS maxPValue \n",

" FROM ",coefftablename," a \n",

" WHERE a.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID),

" GROUP BY a.ModelID) AS b \n",

" WHERE b.ModelID = a.ModelID \n",

" AND a.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID),

" AND b.MaxPValue < 0.10 \n",

" ORDER BY 3 DESC, 2 \n"),1)

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(familytype=="logistic")

vsqlstr <- [limitRowsSQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)([paste0](https://rdrr.io/r/base/paste.html)("SELECT a.\*\n",

" FROM ",statstablename," a\n",

" WHERE a.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID),

" AND a.HighestPValue < 0.10 \n",

" ORDER BY 3 DESC, 2 \n"),1)

d <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),vsqlstr)

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(d) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(d))

d[["ANALYSISID"]] <- [**NULL**](https://rdrr.io/r/base/NULL.html)

vmaxModelID <- d[["MODELID"]]

[**if**](https://rdrr.io/r/base/Control.html)([trace](https://rdrr.io/r/base/trace.html)>0) [print](https://rdrr.io/r/base/print.html)(d)

}

vfuncName <- [ifelse](https://rdrr.io/r/base/ifelse.html)(familytype %in% [c](https://rdrr.io/r/base/c.html)("logisticwt","poisson"),

"FLLogRegr",functionName)

vfuncName <- [base](https://rdrr.io/r/base/base-package.html)::[gsub](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/gsub.html)("MultiDataSet","MD",vfuncName)

vfuncName <- [ifelse](https://rdrr.io/r/base/ifelse.html)(familytype %in% [c](https://rdrr.io/r/base/c.html)("pls", "opls"), "FLPLSRegr", vfuncName)

[**return**](https://rdrr.io/r/base/function.html)([new](https://rdrr.io/r/methods/new.html)(vfuncName,

[formula](https://rdrr.io/r/stats/formula.html)=[formula](https://rdrr.io/r/stats/formula.html),

AnalysisID=AnalysisID,

wideToDeepAnalysisID=wideToDeepAnalysisID,

[table](https://rdrr.io/r/base/table.html)=[data](https://rdrr.io/r/utils/data.html),

results=[list](https://rdrr.io/r/base/list.html)([call](https://rdrr.io/r/base/call.html)=callObject,

modelID=vmaxModelID,

mod = mod),

deeptable=deepx,

mapTable=mapTable,

scoreTable="",

vfcalls=vfcalls,

[offset](https://rdrr.io/r/stats/offset.html)=[as.character](https://rdrr.io/r/base/character.html)([offset](https://rdrr.io/r/stats/offset.html)),

RegrDataPrepSpecs=RegrDataPrepSpecs))

}

*#' @export*

[prepareData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html),[**...**](https://rdrr.io/r/base/dots.html)) [UseMethod](https://rdrr.io/r/base/UseMethod.html)("prepareData")

*#' @export*

[prepareData.FLpreparedData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html), [data](https://rdrr.io/r/utils/data.html), fetchIDs=[**FALSE**](https://rdrr.io/r/base/logical.html), outDeepTable="", [**...**](https://rdrr.io/r/base/dots.html)) {

template <- [formula](https://rdrr.io/r/stats/formula.html)

dataCopy <- [data](https://rdrr.io/r/utils/data.html)

vRegrDataPrepSpecs <- [setDefaultsRegrDataPrepSpecs](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(x=template$RegrDataPrepSpecs,

values=[list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html)))

deepx <- [FLRegrDataPrep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/FLRegrDataPrep.html)([data](https://rdrr.io/r/utils/data.html),depCol=vRegrDataPrepSpecs$depCol,

OutDeepTable=outDeepTable,

OutObsIDCol=vRegrDataPrepSpecs$outObsIDCol,

OutVarIDCol=vRegrDataPrepSpecs$outVarIDCol,

OutValueCol=vRegrDataPrepSpecs$outValueCol,

CatToDummy=vRegrDataPrepSpecs$catToDummy,

PerformNorm=vRegrDataPrepSpecs$performNorm,

PerformVarReduc=vRegrDataPrepSpecs$performVarReduc,

MakeDataSparse=vRegrDataPrepSpecs$makeDataSparse,

MinStdDev=vRegrDataPrepSpecs$minStdDev,

MaxCorrel=vRegrDataPrepSpecs$maxCorrel,

TrainOrTest=1,

ExcludeCols=vRegrDataPrepSpecs$excludeCols,

ClassSpec=vRegrDataPrepSpecs$classSpec,

WhereClause=vRegrDataPrepSpecs$whereconditions,

InAnalysisID=template$wideToDeepAnalysisID,

fetchIDs=fetchIDs)

[data](https://rdrr.io/r/utils/data.html) <- deepx

[data](https://rdrr.io/r/utils/data.html) <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)([data](https://rdrr.io/r/utils/data.html),"")

[data](https://rdrr.io/r/utils/data.html)

}

*## gk: todo: make this obsolete and refactor to use prepareData.FLpreparedData*

*#' @export*

[prepareData.FLRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html), [data](https://rdrr.io/r/utils/data.html), outDeepTableName="",

fetchIDs=[**FALSE**](https://rdrr.io/r/base/logical.html), [**...**](https://rdrr.io/r/base/dots.html)) {

[**if**](https://rdrr.io/r/base/Control.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)([data](https://rdrr.io/r/utils/data.html))) [**return**](https://rdrr.io/r/base/function.html)([data](https://rdrr.io/r/utils/data.html))

dataCopy <- [data](https://rdrr.io/r/utils/data.html)

vRegrDataPrepSpecs <- [setDefaultsRegrDataPrepSpecs](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(x=[formula](https://rdrr.io/r/stats/formula.html)@RegrDataPrepSpecs,

values=[list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html)))

vdepCol <- [formula](https://rdrr.io/r/stats/formula.html)@RegrDataPrepSpecs$depCol

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(vdepCol))

vdepCol <- "NULL"

deepx <- [FLRegrDataPrep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/FLRegrDataPrep.html)([data](https://rdrr.io/r/utils/data.html),depCol=vdepCol,

OutDeepTable=outDeepTableName,

OutObsIDCol=vRegrDataPrepSpecs$outObsIDCol,

OutVarIDCol=vRegrDataPrepSpecs$outVarIDCol,

OutValueCol=vRegrDataPrepSpecs$outValueCol,

CatToDummy=vRegrDataPrepSpecs$catToDummy,

PerformNorm=vRegrDataPrepSpecs$performNorm,

PerformVarReduc=vRegrDataPrepSpecs$performVarReduc,

MakeDataSparse=vRegrDataPrepSpecs$makeDataSparse,

MinStdDev=vRegrDataPrepSpecs$minStdDev,

MaxCorrel=vRegrDataPrepSpecs$maxCorrel,

TrainOrTest=1,

ExcludeCols=vRegrDataPrepSpecs$excludeCols,

ClassSpec=vRegrDataPrepSpecs$classSpec,

WhereClause=vRegrDataPrepSpecs$whereconditions,

InAnalysisID=[formula](https://rdrr.io/r/stats/formula.html)@wideToDeepAnalysisID,

fetchIDs=fetchIDs)

[data](https://rdrr.io/r/utils/data.html) <- deepx

[data](https://rdrr.io/r/utils/data.html) <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)([data](https://rdrr.io/r/utils/data.html),"")

[**if**](https://rdrr.io/r/base/Control.html)([formula](https://rdrr.io/r/stats/formula.html)@vfcalls["functionName"]=="FLPoissonRegr"){

*## Insert dependent and offset varids in deeptable*

vVaridCols <- [c](https://rdrr.io/r/base/c.html)(-2)

vcellValCols <- [ifelse](https://rdrr.io/r/base/ifelse.html)([formula](https://rdrr.io/r/stats/formula.html)@[offset](https://rdrr.io/r/stats/offset.html)!="",[formula](https://rdrr.io/r/stats/formula.html)@[offset](https://rdrr.io/r/stats/offset.html),0)

*## if(!all.vars(formula@formula)[1] %in% colnames(dataCopy))*

*## stop("dependent column ",all.vars(formula@formula)[1],*

*## " not in data \n ")*

vVaridCols <- [c](https://rdrr.io/r/base/c.html)(vVaridCols,-1)

vcellValCols <- [c](https://rdrr.io/r/base/c.html)(vcellValCols,[all.vars](https://rdrr.io/r/base/allnames.html)([formula](https://rdrr.io/r/stats/formula.html)@[formula](https://rdrr.io/r/stats/formula.html))[1])

*# vtablename <- getTableNameSlot(dataCopy)*

vtablename <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)([table](https://rdrr.io/r/base/table.html))

vtablename1 <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)([data](https://rdrr.io/r/utils/data.html))

vobsid <- getObsIdSQLExpression([formula](https://rdrr.io/r/stats/formula.html)@[table](https://rdrr.io/r/base/table.html))

sqlstr <- [paste0](https://rdrr.io/r/base/paste.html)(" SELECT ",vobsid," AS obs\_id\_colname, \n ",

vVaridCols," AS var\_id\_colname, \n ",

vcellValCols," AS cell\_val\_colname \n ",

" FROM ",vtablename,collapse=" UNION ALL ")

[t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html) <- [insertIntotbl](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)(pTableName=vtablename1,

pSelect=sqlstr)

[data](https://rdrr.io/r/utils/data.html)@Dimnames[[2]] <- [c](https://rdrr.io/r/base/c.html)("-1","-2",[data](https://rdrr.io/r/utils/data.html)@Dimnames[[2]])

}

[data](https://rdrr.io/r/utils/data.html)

}

*## move to file lmGeneric.R*

*#' @export*

[prepareData.formula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html),

callObject=[**NULL**](https://rdrr.io/r/base/NULL.html),

familytype="linear",

specID=[list](https://rdrr.io/r/base/list.html)(),

direction="",

[trace](https://rdrr.io/r/base/trace.html)=1,

catToDummy=0,

performNorm=0,

performVarReduc=0,

makeDataSparse=1,

minStdDev=0,

maxCorrel=1,

classSpec=[list](https://rdrr.io/r/base/list.html)(),

whereconditions="",

highestpAllow1=0.5,

highestpAllow2=0.1,

stepWiseDecrease=0.05,

topN=1,

pThreshold=0.1,

eventweight=0.8,

noneventweight=1,

maxiter=25,

[offset](https://rdrr.io/r/stats/offset.html)="",

pRefLevel=[**NULL**](https://rdrr.io/r/base/NULL.html),

fetchIDs=[**FALSE**](https://rdrr.io/r/base/logical.html),

outDeepTableName="",

[**...**](https://rdrr.io/r/base/dots.html)){

[data](https://rdrr.io/r/utils/data.html) <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)([data](https://rdrr.io/r/utils/data.html),"")

[**if**](https://rdrr.io/r/base/Control.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)([data](https://rdrr.io/r/utils/data.html))){

vallVars <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)([data](https://rdrr.io/r/utils/data.html))

*##For MultiDataset and deep data*

*##colnames is a list*

*##Assumption: All Models have same formula*

*## Meaning same varIDs*

[**if**](https://rdrr.io/r/base/Control.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([data](https://rdrr.io/r/utils/data.html))){

[**if**](https://rdrr.io/r/base/Control.html)(![length](https://rdrr.io/r/base/length.html)([unique](https://rdrr.io/r/base/unique.html)(vallVars))==1)

*# stop("Datasets should have same columns \n ")*

vallVars <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)([data](https://rdrr.io/r/utils/data.html))[[1]]

[**else**](https://rdrr.io/r/base/Control.html) vallVars <- vallVars[[1]][1]:vallVars[[1]][2]

}

[formula](https://rdrr.io/r/stats/formula.html) <- [genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(vallVars)

}

[else](https://rdrr.io/r/base/Control.html){

[**if**](https://rdrr.io/r/base/Control.html)([isDotFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([formula](https://rdrr.io/r/stats/formula.html))){

vexcludeCols <- [**NULL**](https://rdrr.io/r/base/NULL.html)

[**if**](https://rdrr.io/r/base/Control.html)("excludeCols" %in% [names](https://rdrr.io/r/base/names.html)([list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))))

vexcludeCols <- [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$excludeCols

[**if**](https://rdrr.io/r/base/Control.html)("ExcludeCols" %in% [names](https://rdrr.io/r/base/names.html)([list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))))

vexcludeCols <- [list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))$ExcludeCols

[formula](https://rdrr.io/r/stats/formula.html) <- [genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(pColnames=[setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)([data](https://rdrr.io/r/utils/data.html)),

[c](https://rdrr.io/r/base/c.html)(vexcludeCols,

getObsIdSQLExpression([data](https://rdrr.io/r/utils/data.html)))),

pDepColumn=[all.vars](https://rdrr.io/r/base/allnames.html)([formula](https://rdrr.io/r/stats/formula.html))[1])

}

vallVars <- [base](https://rdrr.io/r/base/base-package.html)::[all.vars](https://rdrr.io/r/base/allnames.html)([formula](https://rdrr.io/r/stats/formula.html))

vdependent <- vallVars[1]

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(vdependent))

vdependent <- "NULL"

vindependent <- vallVars[2:[length](https://rdrr.io/r/base/length.html)(vallVars)]

[checkValidFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)([formula](https://rdrr.io/r/stats/formula.html),[data](https://rdrr.io/r/utils/data.html))

}

vcolnames <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)([data](https://rdrr.io/r/utils/data.html))

wideToDeepAnalysisID <- ""

mapTable <- ""

[**if**](https://rdrr.io/r/base/Control.html)([offset](https://rdrr.io/r/stats/offset.html)!="" && ![toupper](https://rdrr.io/r/base/chartr.html)([offset](https://rdrr.io/r/stats/offset.html)) %in% [toupper](https://rdrr.io/r/base/chartr.html)(vcolnames))

[stop](https://rdrr.io/r/base/stop.html)("offset not in colnames of data")

check0To1 <- [**function**](https://rdrr.io/r/base/function.html)(pObject)

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.numeric](https://rdrr.io/r/base/numeric.html)(pObject) ||

pObject < 0 ||

pObject > 1)

[stop](https://rdrr.io/r/base/stop.html)([names](https://rdrr.io/r/base/names.html)(pObject)," should be >0 and <1\n")

}

checkSpecID <- [**function**](https://rdrr.io/r/base/function.html)(pObject,pAllVars)

{

pObject <- [c](https://rdrr.io/r/base/c.html)(pObject[["include"]],pObject[["exclude"]])

[**if**](https://rdrr.io/r/base/Control.html)([length](https://rdrr.io/r/base/length.html)(pObject)>0)

{

[sapply](https://rdrr.io/r/base/lapply.html)(pObject,[**function**](https://rdrr.io/r/base/function.html)(x)

[**if**](https://rdrr.io/r/base/Control.html)(!(x %in% pAllVars))

[stop](https://rdrr.io/r/base/stop.html)([paste0](https://rdrr.io/r/base/paste.html)(x,collapse=",")," specified in SpecID not in colnames of data\n"))

}

}

maxiter <- maxiter[1]

[**if**](https://rdrr.io/r/base/Control.html)(![is.numeric](https://rdrr.io/r/base/numeric.html)(maxiter) || maxiter<=0)

[stop](https://rdrr.io/r/base/stop.html)("maxiter should be >0")

maxiter <- [as.integer](https://rdrr.io/r/base/integer.html)(maxiter)

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% [c](https://rdrr.io/r/base/c.html)("logistic","logisticwt"))

{

check0To1([c](https://rdrr.io/r/base/c.html)(pThreshold=pThreshold))

[**if**](https://rdrr.io/r/base/Control.html)(familytype %in% "logisticwt"){

check0To1([c](https://rdrr.io/r/base/c.html)(eventweight=eventweight))

check0To1([c](https://rdrr.io/r/base/c.html)(noneventweight=noneventweight))

}

}

[**if**](https://rdrr.io/r/base/Control.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([data](https://rdrr.io/r/utils/data.html))){

[**if**](https://rdrr.io/r/base/Control.html)(!familytype %in% [c](https://rdrr.io/r/base/c.html)("logistic","linear"))

[stop](https://rdrr.io/r/base/stop.html)("only lm and glm with binomial family supported for MultiDataSet\n")

direction <- ""

}

[**if**](https://rdrr.io/r/base/Control.html)(direction=="UFbackward")

{

check0To1([c](https://rdrr.io/r/base/c.html)(highestpAllow1=highestpAllow1,

highestpAllow2=highestpAllow2,

stepWiseDecrease=stepWiseDecrease))

checkSpecID(specID,vallVars)

}

[**if**](https://rdrr.io/r/base/Control.html)(direction=="backward")

{

check0To1([c](https://rdrr.io/r/base/c.html)(highestpAllow1=highestpAllow1))

checkSpecID(specID,vallVars)

}

[**if**](https://rdrr.io/r/base/Control.html)(direction=="forward")

{

check0To1([c](https://rdrr.io/r/base/c.html)(highestpAllow1=highestpAllow1))

[**if**](https://rdrr.io/r/base/Control.html)(![is.numeric](https://rdrr.io/r/base/numeric.html)(topN) || [as.integer](https://rdrr.io/r/base/integer.html)(topN)<1 || [as.integer](https://rdrr.io/r/base/integer.html)(topN)>10)

[stop](https://rdrr.io/r/base/stop.html)("topN should be >0 and <=10")

topN <- [as.integer](https://rdrr.io/r/base/integer.html)(topN)

}

[**if**](https://rdrr.io/r/base/Control.html)(direction=="Fbackward")

{

check0To1([c](https://rdrr.io/r/base/c.html)(highestpAllow1=highestpAllow1,

highestpAllow2=highestpAllow2))

checkSpecID(specID,vallVars)

}

[**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)([data](https://rdrr.io/r/utils/data.html))){

*##browser()*

unused\_cols <- [setdiff](https://rdrr.io/r/base/sets.html)(vcolnames,[c](https://rdrr.io/r/base/c.html)([all.vars](https://rdrr.io/r/base/allnames.html)([formula](https://rdrr.io/r/stats/formula.html)),specID[["exclude"]]))

unused\_cols <- [setdiff](https://rdrr.io/r/base/sets.html)(unused\_cols,

[c](https://rdrr.io/r/base/c.html)(getGroupIdSQLExpression([data](https://rdrr.io/r/utils/data.html)),

getObsIdSQLExpression([data](https://rdrr.io/r/utils/data.html))))

*## Detect factors and assign classSpec*

vfirstRow <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[limitRowsSQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)([paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM (",

[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "),1))

vtblInfo <- [separateDBName](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)([getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)([data](https://rdrr.io/r/utils/data.html)))

vColInfo <- [c](https://rdrr.io/r/base/c.html)()

[**if**](https://rdrr.io/r/base/Control.html)([is.TD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/platforms.R)())

vColInfo <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT columnName FROM dbc.columns WHERE \n ",

"columnType = 'CV' AND databaseName= ",

[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(vtblInfo["vdatabase"])," \n ",

" AND tableName = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(vtblInfo["vtableName"])))[[1]]

vfactorCols <- [list](https://rdrr.io/r/base/list.html)()

*## apply(t,2,function(x){class(x[[1]])}) gives all character*

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) [setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vfirstRow),

[c](https://rdrr.io/r/base/c.html)(unused\_cols,[names](https://rdrr.io/r/base/names.html)(classSpec),

getGroupIdSQLExpression([data](https://rdrr.io/r/utils/data.html)),

getObsIdSQLExpression([data](https://rdrr.io/r/utils/data.html)),

"obs\_id\_colname",

getGroupIdSQLExpression([data](https://rdrr.io/r/utils/data.html)),

"group\_id\_colname",

[list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))[["doNotTransform"]]))){

*##browser()*

[**if**](https://rdrr.io/r/base/Control.html)([length](https://rdrr.io/r/base/length.html)(i)==0) [**break**](https://rdrr.io/r/base/Control.html);

[**if**](https://rdrr.io/r/base/Control.html)([is.factor](https://rdrr.io/r/base/factor.html)(vfirstRow[[i]])

|| [is.character](https://rdrr.io/r/base/character.html)(vfirstRow[[i]])

|| [is.logical](https://rdrr.io/r/base/logical.html)(vfirstRow[[i]])

|| (i %in% [sub](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sub.html)("\\s+$", "", vColInfo))){ *## remove trailing spaces*

*# if(is.logical(vfirstRow[[i]])){*

*# vtemp <- levels(sqlQuery(getFLConnection(),*

*# paste0("SELECT DISTINCT(",i,*

*# ") FROM(",constructSelect(data),") a "))[[1]])[1]*

*# names(vtemp) <- i*

*# classSpec <- c(classSpec,vtemp)*

*# }*

*# else{*

r<-[as.character](https://rdrr.io/r/base/character.html)(vfirstRow[[i]])

[names](https://rdrr.io/r/base/names.html)(r) <- i

vfactorCols <- [c](https://rdrr.io/r/base/c.html)(vfactorCols,r)

*# }*

}

}

[**if**](https://rdrr.io/r/base/Control.html)([length](https://rdrr.io/r/base/length.html)(vfactorCols)>0){

[**if**](https://rdrr.io/r/base/Control.html)([is.ODBC](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)())

vrefVars <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT ",

[paste0](https://rdrr.io/r/base/paste.html)("MIN(",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

") AS ",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

collapse=","),

" FROM (",[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "),

as.is=[**TRUE**](https://rdrr.io/r/base/logical.html))

[**else**](https://rdrr.io/r/base/Control.html) vrefVars <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT ",

[paste0](https://rdrr.io/r/base/paste.html)("MIN(",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

") AS ",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

collapse=","),

" FROM (",[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "))

vtempList <- [list](https://rdrr.io/r/base/list.html)()

vrefVarNames <- [names](https://rdrr.io/r/base/names.html)(vrefVars)

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vrefVars)){

*## Remove variables with NA*

[**if**](https://rdrr.io/r/base/Control.html)([is.na](https://rdrr.io/r/base/NA.html)(vrefVars[[i]]))

vrefVarNames <- [setdiff](https://rdrr.io/r/base/sets.html)(vrefVarNames,

i)

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([is.logical](https://rdrr.io/r/base/logical.html)(vrefVars[[i]]))

vtempList <- [c](https://rdrr.io/r/base/c.html)(vtempList,

[levels](https://rdrr.io/r/base/levels.html)([as.factor](https://rdrr.io/r/base/factor.html)([sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT DISTINCT(",i,

") FROM(",[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "))[[1]]))[1])

[**else**](https://rdrr.io/r/base/Control.html) vtempList <- [c](https://rdrr.io/r/base/c.html)(vtempList,[as.character](https://rdrr.io/r/base/character.html)(vrefVars[[i]]))

}

[names](https://rdrr.io/r/base/names.html)(vtempList) <- vrefVarNames

*# vfactorCols[names(vrefVars)] <- as.list(apply(vrefVars[names(vrefVars)],*

*# 2,function(x){*

*# browser()*

*# if(is.logical(x))*

*# return(levels(sqlQuery(getFLConnection(),*

*# paste0("SELECT DISTINCT(",names(x),*

*# ") FROM(",constructSelect(data),") a "))[[1]])[1])*

*# else as.character(x)*

*# }))*

classSpec <- [c](https://rdrr.io/r/base/c.html)(classSpec,vtempList)

}

*# vexcludeCols <- paste0(unused\_cols,collapse=",")*

vexcludeCols <- [setdiff](https://rdrr.io/r/base/sets.html)(unused\_cols,

getObsIdSQLExpression([data](https://rdrr.io/r/utils/data.html)))

}

vcallObject <- callObject

vRegrDataPrepSpecs <- [list](https://rdrr.io/r/base/list.html)()

[**if**](https://rdrr.io/r/base/Control.html)(![isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)([data](https://rdrr.io/r/utils/data.html)))

{

deepx <- [FLRegrDataPrep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/FLRegrDataPrep.html)([data](https://rdrr.io/r/utils/data.html),depCol=vdependent,

OutDeepTable=outDeepTableName,

OutObsIDCol="obsid",

OutVarIDCol="varid",

OutValueCol="numval",

CatToDummy=catToDummy,

PerformNorm=performNorm,

PerformVarReduc=performVarReduc,

MakeDataSparse=makeDataSparse,

MinStdDev=minStdDev,

MaxCorrel=maxCorrel,

TrainOrTest=0,

ExcludeCols=vexcludeCols,

ClassSpec=classSpec,

WhereClause=whereconditions,

InAnalysisID="",

fetchIDs=fetchIDs)

vRegrDataPrepSpecs <- [list](https://rdrr.io/r/base/list.html)(

outObsIDCol="obsid",

outVarIDCol="varid",

outValueCol="numval",

catToDummy=catToDummy,

performNorm=performNorm,

performVarReduc=performVarReduc,

makeDataSparse=makeDataSparse,

minStdDev=minStdDev,

maxCorrel=maxCorrel,

trainOrTest=0,

excludeCols=vexcludeCols,

classSpec=classSpec)

wideToDeepAnalysisID <- deepx@wideToDeepAnalysisID

deepx <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)(deepx,"")

whereconditions <- ""

mapTable <- [getRemoteTableName](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)(tableName=[getSystemTableMapping](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/platforms.R)("fzzlRegrDataPrepMap"),

temporaryTable=[**FALSE**](https://rdrr.io/r/base/logical.html))

[**if**](https://rdrr.io/r/base/Control.html)(familytype=="poisson")

{

vtablename <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(deepx)

vtablename1 <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)([data](https://rdrr.io/r/utils/data.html))

vobsid <- getObsIdSQLExpression([data](https://rdrr.io/r/utils/data.html))

sqlstr <- [paste0](https://rdrr.io/r/base/paste.html)(" SELECT ",vobsid," AS obs\_id\_colname,","\n ",

" -2 AS var\_id\_colname,","\n ",

[ifelse](https://rdrr.io/r/base/ifelse.html)([offset](https://rdrr.io/r/stats/offset.html)!="",[offset](https://rdrr.io/r/stats/offset.html),0)," AS cell\_val\_colname","\n ",

" FROM ",vtablename1)

[t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html) <- [insertIntotbl](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)(pTableName=vtablename,

pSelect=sqlstr)

deepx@Dimnames[[2]] <- [c](https://rdrr.io/r/base/c.html)("-2",deepx@Dimnames[[2]])

}

*##Get Mapping Information for specID*

[**if**](https://rdrr.io/r/base/Control.html)(![is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)([data](https://rdrr.io/r/utils/data.html))){

vmapping <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT a.Column\_name AS colname,\n",

" a.Final\_VarID AS varid\n",

" FROM ",mapTable," AS a\n",

" WHERE a.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(wideToDeepAnalysisID),

" AND a.Final\_VarID IS NOT NULL \n",

" ORDER BY a.Final\_VarID\n"))

vtemp <- vmapping[["varid"]]

[names](https://rdrr.io/r/base/names.html)(vtemp) <- vmapping[["colname"]]

vmapping <- vtemp

vallVars <- [setdiff](https://rdrr.io/r/base/sets.html)(vallVars,specID[["exclude"]])

}

[**else**](https://rdrr.io/r/base/Control.html) vmapping <- ""

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([class](https://rdrr.io/r/base/class.html)([data](https://rdrr.io/r/utils/data.html)@select)=="FLTableFunctionQuery")

{

*#sqlstr <- paste0("CREATE VIEW ",getOption("ResultDatabaseFL"),*

*# ".",deeptablename," AS ",constructSelect(data))*

*#sqlSendUpdate(connection,sqlstr)*

deeptablename <- [createView](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/createView.html)(pViewName=[gen\_view\_name](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(""),

pSelect=[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)))

*#sqlstr <- paste0("CREATE VIEW ",getOption("ResultDatabaseFL"),".",deeptablename1,*

*# " AS SELECT \* FROM ",getOption("ResultDatabaseFL"),".",deeptablename,*

*# constructWhere(whereconditions))*

*#t <- sqlSendUpdate(connection,sqlstr)*

deeptablename1<-[createView](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/createView.html)(pViewName=[gen\_view\_name](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)("New"),

pSelect=[paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM ",deeptablename,

constructWhere(whereconditions)))

deepx <- [FLTable](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/FLTable.html)(deeptablename1,

getObsIdSQLExpression([data](https://rdrr.io/r/utils/data.html)),

getVarIdSQLExpression([data](https://rdrr.io/r/utils/data.html)),

getValueSQLExpression([data](https://rdrr.io/r/utils/data.html))

)

deepx <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)([setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R),"")

whereconditions <- ""

vmapping <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(deepx)

[names](https://rdrr.io/r/base/names.html)(vmapping) <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(deepx)

}

[else](https://rdrr.io/r/base/Control.html)

{

deepx <- [data](https://rdrr.io/r/utils/data.html)

[data](https://rdrr.io/r/utils/data.html)@select@whereconditions <- [c](https://rdrr.io/r/base/c.html)([data](https://rdrr.io/r/utils/data.html)@select@whereconditions,whereconditions)

[**if**](https://rdrr.io/r/base/Control.html)([length](https://rdrr.io/r/base/length.html)([data](https://rdrr.io/r/utils/data.html)@select@whereconditions)>0 &&

[data](https://rdrr.io/r/utils/data.html)@select@whereconditions!=""){

*## Hadoop does not support \_ in column names*

*## for FLLinRegrMultiDataSet*

[**if**](https://rdrr.io/r/base/Control.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(deepx)){

[data](https://rdrr.io/r/utils/data.html) <- setIndexSQLName([data](https://rdrr.io/r/utils/data.html),1,"groupid")

[data](https://rdrr.io/r/utils/data.html) <- setIndexSQLName([data](https://rdrr.io/r/utils/data.html),2,"obsid")

[data](https://rdrr.io/r/utils/data.html) <- setIndexSQLName([data](https://rdrr.io/r/utils/data.html),3,"varid")

[data](https://rdrr.io/r/utils/data.html) <- setIndexSQLName([data](https://rdrr.io/r/utils/data.html),4,"numval")

}

*#sqlstr <- paste0("CREATE VIEW ",getOption("ResultDatabaseFL"),".",*

*# deeptablename," AS ",constructSelect(data))*

*#t <- sqlSendUpdate(connection,sqlstr)*

deeptablename<-[createView](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/createView.html)(pViewName=[gen\_view\_name](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)("New"),

pSelect=[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)))

[**if**](https://rdrr.io/r/base/Control.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(deepx))

deepx <- [FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/FLTableMD.html)(deeptablename,

getGroupIdSQLName([data](https://rdrr.io/r/utils/data.html)),

getObsIdSQLName([data](https://rdrr.io/r/utils/data.html)),

getVarIdSQLName([data](https://rdrr.io/r/utils/data.html)),

getValueSQLName([data](https://rdrr.io/r/utils/data.html)))

[**else**](https://rdrr.io/r/base/Control.html)

deepx <- [FLTable](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/FLTable.html)(deeptablename,

getObsIdSQLName([data](https://rdrr.io/r/utils/data.html)),

getVarIdSQLName([data](https://rdrr.io/r/utils/data.html)),

getValueSQLName([data](https://rdrr.io/r/utils/data.html)))

deepx <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)(deepx,"")

}

whereconditions <- ""

vmapping <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(deepx)

[names](https://rdrr.io/r/base/names.html)(vmapping) <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(deepx)

}

*## Set RefLevel for Multinomial*

[**if**](https://rdrr.io/r/base/Control.html)(familytype=="multinomial"){

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(pRefLevel)){

pRefLevel <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT cell\_val\_colname FROM (",

[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a \n ",

"WHERE obs\_id\_colname = 1 \n AND ",

" var\_id\_colname = -1 \n "))[1,1]

}

pThreshold <- pRefLevel

}

vinclude <- [**NULL**](https://rdrr.io/r/base/NULL.html)

vexclude <- [**NULL**](https://rdrr.io/r/base/NULL.html)

*##Insert SpecID*

vspecID <- "NULL"

[**if**](https://rdrr.io/r/base/Control.html)([is.list](https://rdrr.io/r/base/list.html)(specID) && [length](https://rdrr.io/r/base/length.html)(specID)>0

&& direction %in% [c](https://rdrr.io/r/base/c.html)("UFbackward","Fbackward","backward"))

{

vspecID <- [genRandVarName](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)()

vdf <- [**NULL**](https://rdrr.io/r/base/NULL.html)

vspecIDTable <- [getRemoteTableName](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)(tableName=[ifelse](https://rdrr.io/r/base/ifelse.html)(familytype=="linear","fzzlLinRegrModelVarSpec",

"fzzlLogRegrModelVarSpec"),temporaryTable=[**FALSE**](https://rdrr.io/r/base/logical.html))

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(specID[["include"]]))

{

*#browser()*

vinclude <- vmapping[[charmatch](https://rdrr.io/r/base/charmatch.html)(specID[["include"]],[names](https://rdrr.io/r/base/names.html)(vmapping))]

vinclude <- vinclude[![is.na](https://rdrr.io/r/base/NA.html)(vinclude)]

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(vinclude) || [length](https://rdrr.io/r/base/length.html)(vinclude) < 1)

[stop](https://rdrr.io/r/base/stop.html)("columns in lower are not in deeptable.",

" Might be due to variable reduction during data preparation")

*# sqlstr <- c(sqlstr,paste0("INSERT INTO ", vspecIDTable," VALUES(",fquote(vspecID),",",*

*# vinclude,",","'I')"))*

vdf <- [rbind](https://rdrr.io/r/base/cbind.html)(vdf,[data.frame](https://rdrr.io/r/base/data.frame.html)(vspecID,vinclude,"I"))

}

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(specID[["exclude"]]))

{

vexclude <- vmapping[[charmatch](https://rdrr.io/r/base/charmatch.html)(specID[["exclude"]],[names](https://rdrr.io/r/base/names.html)(vmapping))]

vexclude <- vexclude[![is.na](https://rdrr.io/r/base/NA.html)(vexclude)]

[**if**](https://rdrr.io/r/base/Control.html)([length](https://rdrr.io/r/base/length.html)(vexclude)>0 && vexclude!="")

*# sqlstr <- c(sqlstr,paste0("INSERT INTO ",vspecIDTable," VALUES(",fquote(vspecID),",",*

*# vexclude,",","'X')"))*

vdf <- [rbind](https://rdrr.io/r/base/cbind.html)(vdf,[data.frame](https://rdrr.io/r/base/data.frame.html)(vspecID,vexclude,"X"))

}

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(vdf))

[t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html) <- [insertIntotbl](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)(pTableName=vspecIDTable,

pValues=vdf)

*# t <- sqlSendUpdate(getFLConnection(),paste0(sqlstr,collapse=";"))*

}

result <- [list](https://rdrr.io/r/base/list.html)(deepx=deepx,

wideTable=[data](https://rdrr.io/r/utils/data.html), *## todo: in case a deep table was passed this is not a wideTable*

wideToDeepAnalysisID=wideToDeepAnalysisID,

[formula](https://rdrr.io/r/stats/formula.html)=[formula](https://rdrr.io/r/stats/formula.html),

vmapping=vmapping,

mapTable=mapTable,

vspecID=vspecID,

highestpAllow1=highestpAllow1,

highestpAllow2=highestpAllow2,

topN=topN,

vexclude=vexclude,

vallVars=vallVars,

stepWiseDecrease=stepWiseDecrease,

eventweight=eventweight,

noneventweight=noneventweight,

maxiter=maxiter,

pThreshold=pThreshold,

[offset](https://rdrr.io/r/stats/offset.html)=[offset](https://rdrr.io/r/stats/offset.html),

RegrDataPrepSpecs=vRegrDataPrepSpecs)

[class](https://rdrr.io/r/base/class.html)(result) <- "FLpreparedData"

[**return**](https://rdrr.io/r/base/function.html)(result)

}

*#' @export*

prepareData.lmGeneric <- [prepareData.formula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*#' @export*

prepareData.NULL <- [prepareData.formula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*#' @export*

prepareData.character <- [prepareData.formula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*## move to file lm.R*

*#' @export*

`$.FLLinRegr`<-[**function**](https://rdrr.io/r/base/function.html)(object,property){

*#parentObject <- deparse(substitute(object))*

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),

"(",fixed=T))[2],",",fixed=T))[1]

[**if**](https://rdrr.io/r/base/Control.html)(property=="coefficients"){

coefficientsvector <- [coefficients](https://rdrr.io/r/stats/coef.html)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(coefficientsvector)

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html) (property=="residuals"){

residualsvector <- [residuals](https://rdrr.io/r/stats/residuals.html)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(residualsvector)

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="fitted.values")

{

fitvector <- [fitted.values.FLGAM](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGAM.R)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(fitvector)

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="FLCoeffStdErr")

{

coeffVector <- [coefficients](https://rdrr.io/r/stats/coef.html)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(object@results[["FLCoeffStdErr"]])

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="FLCoeffTStat")

{

coeffVector <- [coefficients](https://rdrr.io/r/stats/coef.html)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(object@results[["FLCoeffTStat"]])

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="FLCoeffPValue")

{

coeffVector <- [coefficients](https://rdrr.io/r/stats/coef.html)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(object@results[["FLCoeffPValue"]])

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="FLCoeffNonZeroDensity")

{

coeffVector <- [coefficients](https://rdrr.io/r/stats/coef.html)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(object@results[["FLCoeffNonZeroDensity"]])

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="FLCoeffCorrelWithRes")

{

coeffVector <- [coefficients](https://rdrr.io/r/stats/coef.html)(object)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(object@results[["FLCoeffCorrelWithRes"]])

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property == "s")

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["s"]]))

{

[**return**](https://rdrr.io/r/base/function.html)(object@results[["s"]])

}

[str](https://rdrr.io/r/utils/str.html) <- [paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM ",object@vfcalls["statstablename"],

" a WHERE a.AnalysisID = '",object@AnalysisID,"'")

[t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html) <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)([connection](https://rdrr.io/r/base/connections.html), [str](https://rdrr.io/r/utils/str.html))

val <- [t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html)[[t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html)$Notation == "Mad\_S",3]

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results, [list](https://rdrr.io/r/base/list.html)(s = val))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(val)

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="call")

{

[**return**](https://rdrr.io/r/base/function.html)(object@results[["call"]])

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="FLLinRegrStats")

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["FLLinRegrStats"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["FLLinRegrStats"]])

[else](https://rdrr.io/r/base/Control.html)

{

sqlstr <- [paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM ",object@vfcalls[["statstablename"]]," \n",

" WHERE AnalysisID=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@AnalysisID),

" \nAND ModelID=",object@results[["modelID"]])

statsdataframe <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),sqlstr)

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)(FLLinRegrStats=statsdataframe))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(statsdataframe)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="df.residual")

{

[**if**](https://rdrr.io/r/base/Control.html)(object@vfcalls["functionName"] == "FLRobustRegr")

[**return**](https://rdrr.io/r/base/function.html)([**NULL**](https://rdrr.io/r/base/NULL.html))

[else](https://rdrr.io/r/base/Control.html)

{

statsdataframe <- object$FLLinRegrStats

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(statsdataframe) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(statsdataframe))

dfResidualVector <- statsdataframe[["DFRESIDUAL"]]

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)([df.residual](https://rdrr.io/r/stats/df.residual.html)=dfResidualVector))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(dfResidualVector)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="model")

{

*## The Column order may not be same as*

*## in formula object because add. columns*

*## may be added by categorical trans.*

*##This might stop any parent script!!*

*##Need something that has wait time and*

*## Default value.*

modelframe <- [model.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(object)

*## Do not store. Better to fetch each time as*

*## it saves memory and not much time loss in*

*## Fetching.*

*##object@results <- c(object@results,list(model=modelframe))*

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(modelframe)

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="x")

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["x"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["x"]])

modelframe <- [getXMatrix](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(object,

pDropCols=[c](https://rdrr.io/r/base/c.html)(-1))

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)(x=modelframe))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(modelframe)

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="y")

{

*##This is safer from simple subsetting of*

*## WideTable as whereConditions may exist*

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["y"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["y"]])

[else](https://rdrr.io/r/base/Control.html)

{

vtablename <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(object@deeptable)

obs\_id\_colname <- getObsIdSQLExpression(object@deeptable)

var\_id\_colname <- getVarIdSQLExpression(object@deeptable)

cell\_val\_colname <- getValueSQLExpression(object@deeptable)

sqlstr <- [paste0](https://rdrr.io/r/base/paste.html)("SELECT '%insertIDhere%' AS vectorIdColumn,\n",

obs\_id\_colname," AS vectorIndexColumn,\n",

cell\_val\_colname," AS vectorValueColumn\n",

" FROM ",vtablename,

" \nWHERE ",var\_id\_colname," = -1 \n")

tblfunqueryobj <- [new](https://rdrr.io/r/methods/new.html)("FLTableFunctionQuery",

connectionName = [getFLConnectionName](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/platforms.R)(),

variables = [list](https://rdrr.io/r/base/list.html)(

obs\_id\_colname = "vectorIndexColumn",

cell\_val\_colname = "vectorValueColumn"),

whereconditions="",

[order](https://rdrr.io/r/base/order.html) = "",

SQLquery=sqlstr)

yvector <- [newFLVector](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(

select = tblfunqueryobj,

Dimnames = [list](https://rdrr.io/r/base/list.html)(object@deeptable@Dimnames[[1]],

"vectorValueColumn"),

dims = [as.integer](https://rdrr.io/r/base/integer.html)([c](https://rdrr.io/r/base/c.html)([nrow](https://rdrr.io/r/base/nrow.html)(object@deeptable),1)),

[isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R) = [**FALSE**](https://rdrr.io/r/base/logical.html))

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)(y=yvector))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(yvector)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="qr" || property=="rank")

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["qr"]]))

{

[**if**](https://rdrr.io/r/base/Control.html)(property=="qr")

[**return**](https://rdrr.io/r/base/function.html)(object@results[["qr"]])

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="rank")

[**return**](https://rdrr.io/r/base/function.html)(object@results[["qr"]]$[rank](https://rdrr.io/r/base/rank.html))

}

[else](https://rdrr.io/r/base/Control.html)

{

modelmatrix <- object$x

[**if**](https://rdrr.io/r/base/Control.html)([nrow](https://rdrr.io/r/base/nrow.html)(modelmatrix)>700

|| [ncol](https://rdrr.io/r/base/nrow.html)(modelmatrix)>700)

modelmatrix <- [as.matrix](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/as.matrix.html)(modelmatrix)

*# modelmatrix <- as.matrix(object$x)*

*# qrList <- base::qr(modelmatrix)*

qrList <- [qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)(modelmatrix)

vrank <- qrList$[rank](https://rdrr.io/r/base/rank.html)

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)([qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)=qrList))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**if**](https://rdrr.io/r/base/Control.html)(property=="qr")

[**return**](https://rdrr.io/r/base/function.html)(qrList)

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="rank") [**return**](https://rdrr.io/r/base/function.html)(vrank)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="terms")

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["terms"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["terms"]])

[else](https://rdrr.io/r/base/Control.html)

{

coeffVector <- object$[coefficients](https://rdrr.io/r/stats/coef.html)

vallVars <- [all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))

vcolnames <- object@results[["modelColnames"]][-1]

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(vcolnames))

vcolnames <- [names](https://rdrr.io/r/base/names.html)(coeffVector)[2:[length](https://rdrr.io/r/base/length.html)(coeffVector)]

vterms <- [terms](https://rdrr.io/r/stats/terms.html)([formula](https://rdrr.io/r/stats/formula.html)([paste0](https://rdrr.io/r/base/paste.html)(vallVars[1],"~",

[paste0](https://rdrr.io/r/base/paste.html)(vcolnames,collapse="+"))))

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)([terms](https://rdrr.io/r/stats/terms.html)=vterms))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(vterms)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="xlevels")

{

[cat](https://rdrr.io/r/base/cat.html)("categorical variables are Transformed \n ")

[**return**](https://rdrr.io/r/base/function.html)([list](https://rdrr.io/r/base/list.html)())

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="assign")

{

[**return**](https://rdrr.io/r/base/function.html)([c](https://rdrr.io/r/base/c.html)(0,[rep](https://rdrr.io/r/base/rep.html)(1,[length](https://rdrr.io/r/base/length.html)([all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html)))-1)))

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="formula")

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["formula"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["formula"]])

[else](https://rdrr.io/r/base/Control.html)

{

coeffVector <- object$[coefficients](https://rdrr.io/r/stats/coef.html)

vallVars <- [all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))

vcolnames <- object@results[["modelColnames"]][-1]

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(vcolnames))

vcolnames <- [names](https://rdrr.io/r/base/names.html)(coeffVector)[2:[length](https://rdrr.io/r/base/length.html)(coeffVector)]

vterms <- [terms](https://rdrr.io/r/stats/terms.html)([formula](https://rdrr.io/r/stats/formula.html)([paste0](https://rdrr.io/r/base/paste.html)(vallVars[1],"~",

[paste0](https://rdrr.io/r/base/paste.html)(vcolnames,collapse="+"))))

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)([terms](https://rdrr.io/r/stats/terms.html)=vterms))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(vterms)

}

}

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(property=="anova") [stop](https://rdrr.io/r/base/stop.html)("This feature is not available yet.")

[**else**](https://rdrr.io/r/base/Control.html) [stop](https://rdrr.io/r/base/stop.html)("That's not a valid property \n ")

}

[setMethod](https://rdrr.io/r/methods/setMethod.html)("names", [signature](https://rdrr.io/r/methods/GenericFunctions.html)("FLRobustRegr"), [**function**](https://rdrr.io/r/base/function.html)(x) [c](https://rdrr.io/r/base/c.html)("coefficients",

"residuals",

"fitted.values",

"x",

"y",

"call" ))

*#' @export*

[setMethod](https://rdrr.io/r/methods/setMethod.html)("names", [signature](https://rdrr.io/r/methods/GenericFunctions.html)("FLLinRegr"), [**function**](https://rdrr.io/r/base/function.html)(x) [c](https://rdrr.io/r/base/c.html)("anova", "formula", "assign",

"xlevels","y","x","model",

"df.residual","FLLinRegrStats",

"call","s","FLCoeffCorrelWithRes"

,"FLCoeffNonZeroDensity",

"FLCoeffPValue","FLCoeffTStat",

"FLCoeffStdErr","fitted.values",

"residuals","coefficients" ))

*#' @export*

[coefficients](https://rdrr.io/r/stats/coef.html)<-[**function**](https://rdrr.io/r/base/function.html)([table](https://rdrr.io/r/base/table.html)){

[UseMethod](https://rdrr.io/r/base/UseMethod.html)("coefficients",[table](https://rdrr.io/r/base/table.html))

}

*#' @export*

coefficients.default <- [stats](https://rdrr.io/r/stats/stats-package.html)::[coefficients](https://rdrr.io/r/stats/coef.html)

*## move to file lm.R*

*#' @export*

[coefficients.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)<-[**function**](https://rdrr.io/r/base/function.html)(object){

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)(

[as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),

"(",fixed=T))[2],")",fixed=T))[1]

[**if**](https://rdrr.io/r/base/Control.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(object@[table](https://rdrr.io/r/base/table.html)))

coeffVector <- [coefficients.FLLinRegrMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(object)

[**else**](https://rdrr.io/r/base/Control.html)

coeffVector <- [coefficients.lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(object,

FLCoeffStats=[c](https://rdrr.io/r/base/c.html)(object@results$mod["FLCoeffStdErr"],

object@results$mod["FLCoeffPValue"],

object@results$mod["FLCoeffTStat"],

object@results$mod["FLCoeffCorrelWithRes"],

object@results$mod["FLCoeffNonZeroDensity"]))

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(coeffVector)

}

*## move to file lmGeneric.R*

*#' @export*

[coefficients.lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <-[**function**](https://rdrr.io/r/base/function.html)(object,

FLCoeffStats=[c](https://rdrr.io/r/base/c.html)(),

pIntercept=[**TRUE**](https://rdrr.io/r/base/logical.html),

[**...**](https://rdrr.io/r/base/dots.html)){

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["coefficients"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["coefficients"]])

[else](https://rdrr.io/r/base/Control.html)

{

*## Since Currently only 1000 Columns are supported*

*## by FLLinRegr, fetch them.*

*#browser()*

*# vmapping <- NULL*

vID <- object@results$mod[["nID"]]

vfcalls <- object@vfcalls

vcoeffnames <- [**NULL**](https://rdrr.io/r/base/NULL.html)

vmodelnames <- [**NULL**](https://rdrr.io/r/base/NULL.html)

[**if**](https://rdrr.io/r/base/Control.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object@[table](https://rdrr.io/r/base/table.html)))

coeffVector <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM ",vfcalls["coefftablename"],

" where AnalysisID=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@AnalysisID),

[ifelse](https://rdrr.io/r/base/ifelse.html)([length](https://rdrr.io/r/base/length.html)(object@results[["modelID"]])>0 &&

object@vfcalls[["functionName"]] != "FLRobustRegr" &&

object@vfcalls["functionName"] != "FLPLSRegr",

[paste0](https://rdrr.io/r/base/paste.html)(" AND ModelID=",object@results[["modelID"]]),""),

" ORDER BY ",vID))

[else](https://rdrr.io/r/base/Control.html){

*#browser()*

vcoeffframe <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT a.\*,b.\* \n",

" FROM ",[getSystemTableMapping](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/platforms.R)("fzzlRegrDataPrepMap")," AS a, \n ",

vfcalls["coefftablename"]," AS b \n",

" WHERE a.Final\_VarID = b.",vID," \n",

" AND a.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@wideToDeepAnalysisID),

"\n AND b.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@AnalysisID),

[ifelse](https://rdrr.io/r/base/ifelse.html)([length](https://rdrr.io/r/base/length.html)(object@results[["modelID"]])>0 &&

object@vfcalls[["functionName"]] != "FLRobustRegr" &&

object@vfcalls[["functionName"]] != "FLPLSRegr",

[paste0](https://rdrr.io/r/base/paste.html)("\n AND b.ModelID = ",object@results[["modelID"]]),""),

"\n ORDER BY b.",vID))

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vcoeffframe) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vcoeffframe))

*# vcolumnnames <- unique(as.character(vcoeffframe[["COLUMN\_NAME"]]))*

*# vcolumnnames <- vcolumnnames[2:length(vcolumnnames)]*

*# vmodelnames <- c(all.vars(object@formula)[1],*

*# vcolumnnames)*

coeffVector <- vcoeffframe

vcolumnnames <- vcoeffframe[["COLUMN\_NAME"]]

vcolumnnames <- vcolumnnames[2:[length](https://rdrr.io/r/base/length.html)(vcolumnnames)]

want <- [all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))

want <- want[2:[length](https://rdrr.io/r/base/length.html)(want)]

[q](https://rdrr.io/r/base/quit.html) <- [unlist](https://rdrr.io/r/base/unlist.html)([sapply](https://rdrr.io/r/base/lapply.html)(want,

[**function**](https://rdrr.io/r/base/function.html)(x){

[which](https://rdrr.io/r/base/which.html)([toupper](https://rdrr.io/r/base/chartr.html)(vcolumnnames) %in% [toupper](https://rdrr.io/r/base/chartr.html)(x))}

))+1

coeffVector <- coeffVector[[c](https://rdrr.io/r/base/c.html)(1,[q](https://rdrr.io/r/base/quit.html)), ]

vmodelnames <- [c](https://rdrr.io/r/base/c.html)([all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))[1],

[unique](https://rdrr.io/r/base/unique.html)([as.character](https://rdrr.io/r/base/character.html)(coeffVector[["COLUMN\_NAME"]])[-1]))

vVarnames <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object@[table](https://rdrr.io/r/base/table.html))

vmapNames <- [**function**](https://rdrr.io/r/base/function.html)([t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html))

{

vindex <- [match](https://rdrr.io/r/base/match.html)([tolower](https://rdrr.io/r/base/chartr.html)([t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html)),[tolower](https://rdrr.io/r/base/chartr.html)(vVarnames))

[**if**](https://rdrr.io/r/base/Control.html)([is.na](https://rdrr.io/r/base/NA.html)(vindex))

vnames <- [t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html)

[**else**](https://rdrr.io/r/base/Control.html) vnames <- vVarnames[vindex]

vnames

}

vcoeffnames <- [as.vector](https://rdrr.io/r/base/vector.html)([apply](https://rdrr.io/r/base/apply.html)(coeffVector,1,

[**function**](https://rdrr.io/r/base/function.html)(x){

[**if**](https://rdrr.io/r/base/Control.html)([tolower](https://rdrr.io/r/base/chartr.html)(x["VAR\_TYPE"])%in%[c](https://rdrr.io/r/base/c.html)("category","varchar"))

[**return**](https://rdrr.io/r/base/function.html)([paste0](https://rdrr.io/r/base/paste.html)(vmapNames(x["COLUMN\_NAME"]),x["CATVALUE"]))

[**else**](https://rdrr.io/r/base/Control.html) [**return**](https://rdrr.io/r/base/function.html)(vmapNames(x["COLUMN\_NAME"]))}))

*# vcoeffnames <- sapply(vcoeffnames,vmapNames)*

}

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(coeffVector) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(coeffVector))

coeffVector1 <- coeffVector[[object@results$mod[["nCoeffEstim"]]]]

*# vmapping <- as.FLVector(unique(c(-2,-1,coeffVector[["COEFFID"]])))*

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(vcoeffnames)){

[**if**](https://rdrr.io/r/base/Control.html)(!pIntercept)

[names](https://rdrr.io/r/base/names.html)(coeffVector1) <- [as.character](https://rdrr.io/r/base/character.html)(vcoeffnames)

[else](https://rdrr.io/r/base/Control.html)

[names](https://rdrr.io/r/base/names.html)(coeffVector1) <- [c](https://rdrr.io/r/base/c.html)("(Intercept)",

[as.character](https://rdrr.io/r/base/character.html)(vcoeffnames)[2:[length](https://rdrr.io/r/base/length.html)(vcoeffnames)])

}

[else](https://rdrr.io/r/base/Control.html){

vallVars <- [all.vars](https://rdrr.io/r/base/allnames.html)([genDeepFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(coeffVector[[vID]]))

[names](https://rdrr.io/r/base/names.html)(coeffVector1) <- [c](https://rdrr.io/r/base/c.html)("(Intercept)",vallVars[2:[length](https://rdrr.io/r/base/length.html)(vallVars)])

[**if**](https://rdrr.io/r/base/Control.html)(!pIntercept)

[names](https://rdrr.io/r/base/names.html)(coeffVector1) <- vallVars

}

*# to remove null values.*

FLCoeffStats <- FLCoeffStats[FLCoeffStats!= ""]

FLCoeffStats <- [lapply](https://rdrr.io/r/base/lapply.html)(FLCoeffStats,

[**function**](https://rdrr.io/r/base/function.html)(x){

[t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html)<-coeffVector[[x]]

[names](https://rdrr.io/r/base/names.html)([t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html))<-[names](https://rdrr.io/r/base/names.html)(coeffVector1)

[t](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/t.html)

})

vcolnames <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object@deeptable)

droppedCols <- vcolnames[!vcolnames %in% [c](https://rdrr.io/r/base/c.html)("-1",coeffVector[[vID]])]

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,

[list](https://rdrr.io/r/base/list.html)([coefficients](https://rdrr.io/r/stats/coef.html)=coeffVector1,

droppedCols=droppedCols),

FLCoeffStats)

object@results[["modelColnames"]]<-vmodelnames

object@results[[vID]] <- [as.numeric](https://rdrr.io/r/base/numeric.html)(coeffVector[[vID]])

*# object@results[["varIDMapping"]] <- vmapping*

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)(

[as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(coeffVector1)

}

}

*## move to file lm.R*

*#' @export*

[residuals.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)<-[**function**](https://rdrr.io/r/base/function.html)(object)

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["residuals"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["residuals"]])

[else](https://rdrr.io/r/base/Control.html)

{

residualsvector <- [calcResiduals](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(object=object)

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)([residuals](https://rdrr.io/r/stats/residuals.html)=residualsvector))

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)(

[as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(residualsvector)

}

}

*## move to file lm.R*

*#' @export*

[model.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,[**...**](https://rdrr.io/r/base/dots.html))

{

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["model"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["model"]])

[else](https://rdrr.io/r/base/Control.html)

{

coeffVector <- object$[coefficients](https://rdrr.io/r/stats/coef.html)

vallVars <- [all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))

vcolnames <- [**NULL**](https://rdrr.io/r/base/NULL.html)

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["modelColnames"]])){

vcolnames <- object@results[["modelColnames"]]

modelframe <- object@[table](https://rdrr.io/r/base/table.html)

modelframe@Dimnames[[2]] <- vcolnames

}

[else](https://rdrr.io/r/base/Control.html){

vdroppedCols <- object@results[["droppedCols"]]

modelframe <- object@deeptable

modelframe@select@whereconditions <- [c](https://rdrr.io/r/base/c.html)(modelframe@select@whereconditions,

[paste0](https://rdrr.io/r/base/paste.html)(getVarIdSQLExpression(object@deeptable),

" NOT IN ","(",[paste0](https://rdrr.io/r/base/paste.html)([c](https://rdrr.io/r/base/c.html)(0,-2,vdroppedCols),

collapse=","),

")"))

[**if**](https://rdrr.io/r/base/Control.html)([is.matrix](https://rdrr.io/r/base/matrix.html)(coeffVector))

vcolnames <- [c](https://rdrr.io/r/base/c.html)(vallVars[1],

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(coeffVector)[2:[ncol](https://rdrr.io/r/base/nrow.html)(coeffVector)])

[**else**](https://rdrr.io/r/base/Control.html)

vcolnames <- [c](https://rdrr.io/r/base/c.html)(vallVars[1],

[names](https://rdrr.io/r/base/names.html)(coeffVector)[2:[length](https://rdrr.io/r/base/length.html)(coeffVector)])

}

*## Have to implement names for FLTable*

*# modelframe@Dimnames <- list(modelframe@Dimnames[[1]],*

*# vcolnames)*

*# return(modelframe)*

modelframe <- [as.data.frame](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/as.data.frame.html)(modelframe)

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)([names](https://rdrr.io/r/base/names.html)(vcolnames)))

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(modelframe) <- [names](https://rdrr.io/r/base/names.html)(vcolnames)

[**else**](https://rdrr.io/r/base/Control.html) [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(modelframe) <- vcolnames

object@results <- [c](https://rdrr.io/r/base/c.html)(object@results,[list](https://rdrr.io/r/base/list.html)(model=modelframe))

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)(

[as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(modelframe)

}

}

*#' @export*

[summary.FLRobustRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object, [**...**](https://rdrr.io/r/base/dots.html)){

[str](https://rdrr.io/r/utils/str.html) <- [paste0](https://rdrr.io/r/base/paste.html)("SELECT a.StdDev, a.T\_Val FROM ",object@vfcalls["coefftablename"]," a

WHERE a.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@AnalysisID),"

")

[df](https://rdrr.io/r/stats/Fdist.html) <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)([connection](https://rdrr.io/r/base/connections.html), [str](https://rdrr.io/r/utils/str.html))

vcoeff <- [data.frame](https://rdrr.io/r/base/data.frame.html)([coefficients](https://rdrr.io/r/stats/coef.html)(object), [df](https://rdrr.io/r/stats/Fdist.html))

[names](https://rdrr.io/r/base/names.html)(vcoeff) <- [c](https://rdrr.io/r/base/c.html)("Value", "Std.Error", "t Value")

vresiduals <- [as.vector](https://rdrr.io/r/base/vector.html)(object$[residuals](https://rdrr.io/r/stats/residuals.html))

vdf <- [c](https://rdrr.io/r/base/c.html)([length](https://rdrr.io/r/base/length.html)(object@deeptable@Dimnames[[2]]),

[length](https://rdrr.io/r/base/length.html)(object@deeptable@Dimnames[[1]]) + 1 - [length](https://rdrr.io/r/base/length.html)(object@deeptable@Dimnames[[2]]),

[length](https://rdrr.io/r/base/length.html)(object@deeptable@Dimnames[[2]]))

reqList <- [list](https://rdrr.io/r/base/list.html)([call](https://rdrr.io/r/base/call.html) = object$[call](https://rdrr.io/r/base/call.html),

[residuals](https://rdrr.io/r/stats/residuals.html)=vresiduals,

[coefficients](https://rdrr.io/r/stats/coef.html) = vcoeff,

[sigma](https://rdrr.io/r/stats/sigma.html) = object$s,

stddev = [**NA**](https://rdrr.io/r/base/NA.html),

[df](https://rdrr.io/r/stats/Fdist.html) = vdf,

r.squared = [**NA**](https://rdrr.io/r/base/NA.html),

cov.unscaled = [**NA**](https://rdrr.io/r/base/NA.html),

[terms](https://rdrr.io/r/stats/terms.html) = [**NA**](https://rdrr.io/r/base/NA.html)

)

*#print(reqList)*

[class](https://rdrr.io/r/base/class.html)(reqList) <- "summary.rlm"

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)

([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(reqList)

}

*## move to file lm.R*

*#' @export*

[summary.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,

[calcResiduals](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)=[**FALSE**](https://rdrr.io/r/base/logical.html)){

[**if**](https://rdrr.io/r/base/Control.html)([is.FLTableMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(object@[table](https://rdrr.io/r/base/table.html)))

reqList <- [summary.FLLinRegrMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(object)

[else](https://rdrr.io/r/base/Control.html)

{

stat <- object$FLLinRegrStats

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(stat) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(stat))

coeffframe <- [data.frame](https://rdrr.io/r/base/data.frame.html)(object$[coefficients](https://rdrr.io/r/stats/coef.html),

object$FLCoeffStdErr,

object$FLCoeffTStat,

object$FLCoeffPValue)

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(coeffframe)<-[c](https://rdrr.io/r/base/c.html)("Estimate","Std. Error","t value","Pr(>|t|)")

*#put rowname*

*# rname <- all.vars(object@formula)*

*# rownames(coeffframe) <- c(rownames(coeffframe)[1], rname[2:length(rname)])*

[rownames](https://rdrr.io/r/base/colnames.html)(coeffframe) <- [names](https://rdrr.io/r/base/names.html)(object$[coefficients](https://rdrr.io/r/stats/coef.html))

[**if**](https://rdrr.io/r/base/Control.html)([calcResiduals](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R))

vresiduals <- [as.vector](https://rdrr.io/r/base/vector.html)(object$[residuals](https://rdrr.io/r/stats/residuals.html))

[**else**](https://rdrr.io/r/base/Control.html) vresiduals <- [**NULL**](https://rdrr.io/r/base/NULL.html)

reqList <- [list](https://rdrr.io/r/base/list.html)([call](https://rdrr.io/r/base/call.html) = [as.call](https://rdrr.io/r/base/call.html)(object@[formula](https://rdrr.io/r/stats/formula.html)),

[residuals](https://rdrr.io/r/stats/residuals.html) = vresiduals,

[coefficients](https://rdrr.io/r/stats/coef.html) = [as.matrix](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/as.matrix.html)(coeffframe),

[sigma](https://rdrr.io/r/stats/sigma.html) = stat$STDERR,

[df](https://rdrr.io/r/stats/Fdist.html) = [as.vector](https://rdrr.io/r/base/vector.html)([c](https://rdrr.io/r/base/c.html)((stat$DFREGRESSION + 1),stat$DFRESIDUAL, (stat$DFREGRESSION + 1))),

r.squared = stat$RSQUARED,

adj.r.squared = stat$ADJRSQUARED,

fstatistic = [c](https://rdrr.io/r/base/c.html)(stat$FSTAT, stat$DFREGRESSION, stat$DFRESIDUAL ),

aliased = [**FALSE**](https://rdrr.io/r/base/logical.html)

)

[class](https://rdrr.io/r/base/class.html)(reqList) <- "summary.lm"

reqList

}

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)

([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(reqList)

}

*## move to file lm.R*

*## Add deep statment, also problem can be of vobsid*

*## Use FLSUMPROD: usemethod dispatch, create new class.*

*#' @export*

[predict.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,

newdata=object@[table](https://rdrr.io/r/base/table.html),

scoreTable="",

[**...**](https://rdrr.io/r/base/dots.html)){

[**return**](https://rdrr.io/r/base/function.html)([predict.lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(object,newdata=newdata,

scoreTable=scoreTable,

[**...**](https://rdrr.io/r/base/dots.html)))

}

[predict.FLRobustRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,

newdata=object@[table](https://rdrr.io/r/base/table.html),

scoreTable="",

[**...**](https://rdrr.io/r/base/dots.html))

{

ObsID <- getVariables(object@deeptable)$obs\_id\_colname

VarID <- getVariables(object@deeptable)$var\_id\_colname

Num\_Val <- getVariables(object@deeptable)$cell\_val\_colname

[str](https://rdrr.io/r/utils/str.html) <- [paste0](https://rdrr.io/r/base/paste.html)(" SELECT '%insertIDhere%' AS vectorIdColumn,

b.",ObsID," AS VectorIndexColumn,

FLSUMPROD(b.",Num\_Val,",a.Est) AS vectorValueColumn FROM ",

object@vfcalls["coefftablename"]," a,",

[getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(object@deeptable)," b

WHERE a.VarID = b.",VarID," AND a.AnalysisID = '",object@AnalysisID,"'

GROUP BY b.",ObsID,"")

tblfunqueryobj <- [new](https://rdrr.io/r/methods/new.html)("FLTableFunctionQuery",

connectionName = [getFLConnectionName](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/platforms.R)(),

variables = [list](https://rdrr.io/r/base/list.html)(

obs\_id\_colname = "vectorIndexColumn",

cell\_val\_colname = "vectorValueColumn"),

whereconditions="",

[order](https://rdrr.io/r/base/order.html) = "",

SQLquery=[str](https://rdrr.io/r/utils/str.html))

flv <- [newFLVector](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(

select = tblfunqueryobj,

Dimnames = [list](https://rdrr.io/r/base/list.html)([rownames](https://rdrr.io/r/base/colnames.html)(object@[table](https://rdrr.io/r/base/table.html)),

"vectorValueColumn"),

dims = [as.integer](https://rdrr.io/r/base/integer.html)([c](https://rdrr.io/r/base/c.html)(newdata@dims[1],1)),

[isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R) = [**FALSE**](https://rdrr.io/r/base/logical.html))

[**return**](https://rdrr.io/r/base/function.html)(flv)

}

*## move to file lmGeneric.R*

*#' @export*

[predict.lmGeneric](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,

newdata=object@[table](https://rdrr.io/r/base/table.html),

scoreTable="",

[type](https://rdrr.io/r/base/typeof.html)="response",[**...**](https://rdrr.io/r/base/dots.html)){

[**if**](https://rdrr.io/r/base/Control.html)(![is.FLTable](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/is.FLTable.html)(newdata) && [class](https://rdrr.io/r/base/class.html)(newdata) != "FLpreparedData") [stop](https://rdrr.io/r/base/stop.html)("scoring allowed on FLTable only")

vfcalls <- object@vfcalls

[**if**](https://rdrr.io/r/base/Control.html)([class](https://rdrr.io/r/base/class.html)(newdata) == "FLpreparedData"){

newdata <- newdata$deepx

}

[else](https://rdrr.io/r/base/Control.html){

newdata <- [prepareData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(object,newdata,outDeepTableName="", [**...**](https://rdrr.io/r/base/dots.html)) }

newdata <- [setAlias](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLSubsetting.R)(newdata,"")

[**if**](https://rdrr.io/r/base/Control.html)(scoreTable=="")

*# scoreTable <- paste0(getOption("ResultDatabaseFL"),".",*

*# gen\_score\_table\_name(getTableNameSlot(object@table)))*

scoreTable <- [gen\_score\_table\_name](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)([getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(object@[table](https://rdrr.io/r/base/table.html)))

*# else if(!grep(".",scoreTable))*

*# scoreTable <- paste0(getOption("ResultDatabaseFL"),".",scoreTable)*

vinputTable <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(newdata)

vtable <- [getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(newdata)

vobsid <- getObsIdSQLExpression(newdata)

vvarid <- getVarIdSQLExpression(newdata)

vvalue <- getValueSQLExpression(newdata)

vinputCols <- [list](https://rdrr.io/r/base/list.html)()

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

TableName=vtable,

ObsIDCol=vobsid,

VarIDCol=vvarid,

ValCol=vvalue

)

[**if**](https://rdrr.io/r/base/Control.html)(!object@vfcalls["functionName"]=="FLPoissonRegr")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

WhereClause="NULL")

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

RegrAnalysisID=object@AnalysisID,

ScoreTable=scoreTable)

[**if**](https://rdrr.io/r/base/Control.html)(![is.Hadoop](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/platforms.R)())

vinputCols <- [c](https://rdrr.io/r/base/c.html)(vinputCols,

Note=[genNote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)([paste0](https://rdrr.io/r/base/paste.html)("Scoring ",vfcalls["note"])))

AnalysisID <- [sqlStoredProc](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlStoredProc.html)(getFLConnection(),

vfcalls["scoretablename"],

outputParameter=[c](https://rdrr.io/r/base/c.html)(AnalysisID="a"),

pInputParams=vinputCols)

AnalysisID <- [checkSqlQueryOutput](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)

[**if**](https://rdrr.io/r/base/Control.html)([type](https://rdrr.io/r/base/typeof.html) %in% "link"){

[sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),[paste0](https://rdrr.io/r/base/paste.html)("alter table ",scoreTable," add logit float"))

[sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),[paste0](https://rdrr.io/r/base/paste.html)("update ",scoreTable," set logit = -ln(1/Y - 1) where Y<1"))

object@vfcalls["valcolnamescoretable"]<-"logit"

}

sqlstr <- [getFittedValuesLogRegrSQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)(object,newdata,scoreTable)

*# sqlstr <- paste0(" SELECT '%insertIDhere%' AS vectorIdColumn,",*

*# vobsid," AS vectorIndexColumn,",*

*# vfcalls["valcolnamescoretable"]," AS vectorValueColumn",*

*# " FROM ",scoreTable)*

tblfunqueryobj <- [new](https://rdrr.io/r/methods/new.html)("FLTableFunctionQuery",

connectionName = [getFLConnectionName](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/platforms.R)(),

variables = [list](https://rdrr.io/r/base/list.html)(

obs\_id\_colname = "vectorIndexColumn",

cell\_val\_colname = "vectorValueColumn"),

whereconditions="",

[order](https://rdrr.io/r/base/order.html) = "",

SQLquery=sqlstr)

flv <- [newFLVector](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(

select = tblfunqueryobj,

Dimnames = [list](https://rdrr.io/r/base/list.html)([rownames](https://rdrr.io/r/base/colnames.html)(newdata),

"vectorValueColumn"),

dims = [as.integer](https://rdrr.io/r/base/integer.html)([c](https://rdrr.io/r/base/c.html)(newdata@dims[1],1)),

[isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R) = [**FALSE**](https://rdrr.io/r/base/logical.html))

[**return**](https://rdrr.io/r/base/function.html)(flv)

}

*#' Print FLLinRegr Object*

*#'*

*#' Printing of output from Linear Regression*

*#'*

*#' @title Print FLLinRegr output Info*

*#' @method print FLLinRegr*

*#' @param object prints results of FLLinRegr on FL objects*

*#' @method coefficients FLLinRegr*

*#' @param object returns coefficient vector of the object*

*#' @method residuals FLLinRegr*

*#' @param object the residuals, that is response minus fitted values.*

*#' @method influence FLLinRegr*

*#' @param object returns the basic quantities which are used in forming a wide variety of diagnostics for checking the quality of regression fits.*

*#' @method lm.influence FLLinRegr*

*#' @param object returns the basic quantities which are used in forming a wide variety of diagnostics for checking the quality of regression fits.*

*#' @method plot FLLinRegr*

*#' @param object plots the results of FLLinRegr on FL objects.*

*#' @method summary FLLinRegr*

*#' @method predict FLLinRegr*

*#' @export*

[print.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/print.FLLinRegr.html)<-[**function**](https://rdrr.io/r/base/function.html)(object){

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)(

[as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

reqList <- [list](https://rdrr.io/r/base/list.html)([call](https://rdrr.io/r/base/call.html)=object$[call](https://rdrr.io/r/base/call.html),

[coefficients](https://rdrr.io/r/stats/coef.html)=object$[coefficients](https://rdrr.io/r/stats/coef.html))

[class](https://rdrr.io/r/base/class.html)(reqList) <- "lm"

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[print](https://rdrr.io/r/base/print.html)(reqList)

}

*## move to file lm.R*

*#' @export*

[setMethod](https://rdrr.io/r/methods/setMethod.html)("show","FLLinRegr",[print.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/print.FLLinRegr.html))

*## move to file lm.R*

*#' @export*

[plot.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,method="R",limit=4000,[**...**](https://rdrr.io/r/base/dots.html))

{

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)(

[as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[**if**](https://rdrr.io/r/base/Control.html)(method=="R"){

vqr <- object$[qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)

vqr <- [list](https://rdrr.io/r/base/list.html)([qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)=[as.matrix](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/as.matrix.html)(vqr$[qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)),

[rank](https://rdrr.io/r/base/rank.html)=[as.integer](https://rdrr.io/r/base/integer.html)([as.vector](https://rdrr.io/r/base/vector.html)(vqr$[rank](https://rdrr.io/r/base/rank.html))),

qraux=[as.numeric](https://rdrr.io/r/base/numeric.html)([as.vector](https://rdrr.io/r/base/vector.html)(vqr$qraux)),

pivot=[as.integer](https://rdrr.io/r/base/integer.html)([as.vector](https://rdrr.io/r/base/vector.html)(vqr$pivot)))

[class](https://rdrr.io/r/base/class.html)(vqr)<-"qr"

reqList <- [list](https://rdrr.io/r/base/list.html)([residuals](https://rdrr.io/r/stats/residuals.html)=[as.vector](https://rdrr.io/r/base/vector.html)(object$[residuals](https://rdrr.io/r/stats/residuals.html)),

[coefficients](https://rdrr.io/r/stats/coef.html)=object$[coefficients](https://rdrr.io/r/stats/coef.html),

[df.residual](https://rdrr.io/r/stats/df.residual.html)=object$[df.residual](https://rdrr.io/r/stats/df.residual.html),

[qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)=vqr,

[rank](https://rdrr.io/r/base/rank.html)=vqr$[rank](https://rdrr.io/r/base/rank.html),

[call](https://rdrr.io/r/base/call.html)=object$[call](https://rdrr.io/r/base/call.html),

xlevels=object$xlevels,

model=object$model,

[terms](https://rdrr.io/r/stats/terms.html)=object$[terms](https://rdrr.io/r/stats/terms.html))

[class](https://rdrr.io/r/base/class.html)(reqList) <- "lm"

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[plot](https://rdrr.io/r/graphics/plot.html)(reqList,[**...**](https://rdrr.io/r/base/dots.html))

}

[else](https://rdrr.io/r/base/Control.html){

vObsIdColname <- getVariables(object@deeptable)[["obs\_id\_colname"]]

vVarIdColname <- getVariables(object@deeptable)[["var\_id\_colname"]]

vCellValColname <- getVariables(object@deeptable)[["cell\_val\_colname"]]

p <- [min](https://rdrr.io/r/base/Extremes.html)(limit,[length](https://rdrr.io/r/base/length.html)(object$[fitted.values](https://rdrr.io/r/stats/fitted.values.html)))/[length](https://rdrr.io/r/base/length.html)(object$[fitted.values](https://rdrr.io/r/stats/fitted.values.html))

sqlstr <- [paste0](https://rdrr.io/r/base/paste.html)("SELECT \n ",

" b.",vCellValColname," AS y, \n ",

" a.y AS yhat, \n ",

" (b.",vCellValColname," - a.y) AS residual \n ",

" FROM ",object@scoreTable," a, \n ",

[getTableNameSlot](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/getTableNameSlot.html)(object@deeptable)," b \n ",

" WHERE b.",vObsIdColname,"=a.",vObsIdColname,

" AND b.",vVarIdColname,"=-1 ",

" AND FLSimUniform(",

[getNativeRandFunction](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(pArg1=1,pArg2=10000),

", 0.0, 1.0) < ",p)

vdf <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),sqlstr)

vfit <- vdf[["yhat"]]

vresid <- vdf[["residual"]]

vactual <- vdf[["y"]]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[plot](https://rdrr.io/r/graphics/plot.html)(vfit,vresid,xlab="fitted.values",ylab="residuals",main="residual plot")

[readline](https://rdrr.io/r/base/readline.html)("Hit <Return> to see next plot:")

[plot](https://rdrr.io/r/graphics/plot.html)(vactual,vfit,xlab="actual values",ylab="fitted.values",main="Actual vs Fitted")

}

}

*## move to file lm.R*

*#' @export*

[influence.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(model,[**...**](https://rdrr.io/r/base/dots.html)){

reqList <- [list](https://rdrr.io/r/base/list.html)([residuals](https://rdrr.io/r/stats/residuals.html)=[as.vector](https://rdrr.io/r/base/vector.html)(model$[residuals](https://rdrr.io/r/stats/residuals.html)),

[coefficients](https://rdrr.io/r/stats/coef.html)=model$[coefficients](https://rdrr.io/r/stats/coef.html),

[df.residual](https://rdrr.io/r/stats/df.residual.html)=model$[df.residual](https://rdrr.io/r/stats/df.residual.html),

[qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)=model$[qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html),

[rank](https://rdrr.io/r/base/rank.html)=model$[rank](https://rdrr.io/r/base/rank.html),

[call](https://rdrr.io/r/base/call.html)=model$[call](https://rdrr.io/r/base/call.html),

xlevels=model$xlevels,

model=model$model,

termsz=model$[terms](https://rdrr.io/r/stats/terms.html))

[class](https://rdrr.io/r/base/class.html)(reqList) <- "lm"

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)

([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,model,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)([stats](https://rdrr.io/r/stats/stats-package.html)::[influence](https://rdrr.io/r/stats/lm.influence.html)(reqList,[**...**](https://rdrr.io/r/base/dots.html)))

}

*#' @export*

[lm.influence](https://rdrr.io/r/stats/lm.influence.html) <- [**function**](https://rdrr.io/r/base/function.html)(model,do.coef=[**TRUE**](https://rdrr.io/r/base/logical.html),[**...**](https://rdrr.io/r/base/dots.html)){

[UseMethod](https://rdrr.io/r/base/UseMethod.html)("lm.influence",model)

}

*#' @export*

lm.influence.default <- [stats](https://rdrr.io/r/stats/stats-package.html)::[lm.influence](https://rdrr.io/r/stats/lm.influence.html)

*#' @export*

[lm.influence.FLLinRegr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(model,do.coef=[**TRUE**](https://rdrr.io/r/base/logical.html),[**...**](https://rdrr.io/r/base/dots.html)){

reqList <- [list](https://rdrr.io/r/base/list.html)([residuals](https://rdrr.io/r/stats/residuals.html)=[as.vector](https://rdrr.io/r/base/vector.html)(model$[residuals](https://rdrr.io/r/stats/residuals.html)),

[coefficients](https://rdrr.io/r/stats/coef.html)=model$[coefficients](https://rdrr.io/r/stats/coef.html),

[df.residual](https://rdrr.io/r/stats/df.residual.html)=model$[df.residual](https://rdrr.io/r/stats/df.residual.html),

[qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html)=model$[qr](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/qr.html),

[rank](https://rdrr.io/r/base/rank.html)=model$[rank](https://rdrr.io/r/base/rank.html),

[call](https://rdrr.io/r/base/call.html)=model$[call](https://rdrr.io/r/base/call.html),

xlevels=model$xlevels,

model=model$model,

[terms](https://rdrr.io/r/stats/terms.html)=model$[terms](https://rdrr.io/r/stats/terms.html))

[class](https://rdrr.io/r/base/class.html)(reqList) <- "lm"

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)

([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

[assign](https://rdrr.io/r/base/assign.html)(parentObject,model,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)([stats](https://rdrr.io/r/stats/stats-package.html)::[lm.influence](https://rdrr.io/r/stats/lm.influence.html)(reqList,do.coef,[**...**](https://rdrr.io/r/base/dots.html)))

}

*##Return list of coefficients vectors*

[coefficients.FLLinRegrMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object){

[**if**](https://rdrr.io/r/base/Control.html)(![is.null](https://rdrr.io/r/base/NULL.html)(object@results[["coefficients"]]))

[**return**](https://rdrr.io/r/base/function.html)(object@results[["coefficients"]])

[else](https://rdrr.io/r/base/Control.html)

{

[**if**](https://rdrr.io/r/base/Control.html)([isDeep](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLconstructSQL.R)(object@[table](https://rdrr.io/r/base/table.html))){

coeffVector <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM ",object@vfcalls["coefftablename"],

" where AnalysisID=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@AnalysisID),

" AND ModelID IN(",[paste0](https://rdrr.io/r/base/paste.html)([unlist](https://rdrr.io/r/base/unlist.html)(object@deeptable@Dimnames[[1]]),collapse=","),

") ORDER BY ModelID,CoeffID"))

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(coeffVector) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(coeffVector))

vcoeffnames <- [as.vector](https://rdrr.io/r/base/vector.html)([apply](https://rdrr.io/r/base/apply.html)(coeffVector,1,

[**function**](https://rdrr.io/r/base/function.html)(x){

[**if**](https://rdrr.io/r/base/Control.html)([as.numeric](https://rdrr.io/r/base/numeric.html)(x[["COEFFID"]])=="0")

[**return**](https://rdrr.io/r/base/function.html)("(Intercept)")

[**else**](https://rdrr.io/r/base/Control.html) [**return**](https://rdrr.io/r/base/function.html)([paste0](https://rdrr.io/r/base/paste.html)("Var",x[["COEFFID"]]))

}))

}

[else](https://rdrr.io/r/base/Control.html){

vcoeffframe <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT a.\*,b.\* \n",

" FROM fzzlRegrDataPrepMDMap AS a, \n ",

object@vfcalls["coefftablename"]," AS b \n",

" WHERE a.Final\_VarID = b.CoeffID \n",

" AND a.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@wideToDeepAnalysisID),

"\n AND b.AnalysisID = ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@AnalysisID),

"\n AND a.groupID = b.modelID ",

"\n AND b.ModelID IN(",

[paste0](https://rdrr.io/r/base/paste.html)([unlist](https://rdrr.io/r/base/unlist.html)(object@deeptable@Dimnames[[1]]),

collapse=","),

")\n ORDER BY ModelID,CoeffID"))

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vcoeffframe) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vcoeffframe))

want <- [all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))

want <- want[2:[length](https://rdrr.io/r/base/length.html)(want)]

[q](https://rdrr.io/r/base/quit.html) <- dlply(vcoeffframe,"MODELID",[**function**](https://rdrr.io/r/base/function.html)(x){

[c](https://rdrr.io/r/base/c.html)(1,[unlist](https://rdrr.io/r/base/unlist.html)([sapply](https://rdrr.io/r/base/lapply.html)(want,

[**function**](https://rdrr.io/r/base/function.html)(y)

[which](https://rdrr.io/r/base/which.html)(

[as.character](https://rdrr.io/r/base/character.html)(x[["COLUMN\_NAME"]][-1]) %in% y)

))+1)

})

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) 2:[length](https://rdrr.io/r/base/length.html)([q](https://rdrr.io/r/base/quit.html))){

[q](https://rdrr.io/r/base/quit.html)[[i]] <- [q](https://rdrr.io/r/base/quit.html)[[i]]+[length](https://rdrr.io/r/base/length.html)([q](https://rdrr.io/r/base/quit.html)[[i-1]])

}

coeffVector <- vcoeffframe[[unlist](https://rdrr.io/r/base/unlist.html)([q](https://rdrr.io/r/base/quit.html)),]

vcoeffnames <- [as.vector](https://rdrr.io/r/base/vector.html)([apply](https://rdrr.io/r/base/apply.html)(coeffVector,1,

[**function**](https://rdrr.io/r/base/function.html)(x){

[**if**](https://rdrr.io/r/base/Control.html)(x[["COLUMN\_NAME"]]=="INTERCEPT")

[**return**](https://rdrr.io/r/base/function.html)("(Intercept)")

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([tolower](https://rdrr.io/r/base/chartr.html)(x["VAR\_TYPE"])=="category")

[**return**](https://rdrr.io/r/base/function.html)([paste0](https://rdrr.io/r/base/paste.html)(x["COLUMN\_NAME"],x["CATVALUE"]))

[**else**](https://rdrr.io/r/base/Control.html) [**return**](https://rdrr.io/r/base/function.html)(x["COLUMN\_NAME"])

}))

}

coeffVector[["COEFFNAMES"]] <- vcoeffnames

vcoeffList <- dlply(coeffVector,"MODELID",

[**function**](https://rdrr.io/r/base/function.html)(x){

vcoeff <- x[["COEFFVALUE"]]

[names](https://rdrr.io/r/base/names.html)(vcoeff) <- x[["COEFFNAMES"]]

[**return**](https://rdrr.io/r/base/function.html)(vcoeff)

})

[names](https://rdrr.io/r/base/names.html)(vcoeffList) <- [paste0](https://rdrr.io/r/base/paste.html)("Model",[unlist](https://rdrr.io/r/base/unlist.html)(object@deeptable@Dimnames[[1]]))

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)

([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

object@results[["coefficients"]] <- vcoeffList

object@results[["coeffframe"]] <- coeffVector

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(vcoeffList)

}

}

*#' @export*

[summary.FLLinRegrMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object){

vcoeffList <- object$[coefficients](https://rdrr.io/r/stats/coef.html)

coeffframe <- object@results[["coeffframe"]]

[**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(object@results[["statsframe"]]))

statsframe <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM ",object@vfcalls["statstablename"],

" WHERE AnalysisID=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(object@AnalysisID),

" ORDER BY MODELID "))

[**else**](https://rdrr.io/r/base/Control.html) statsframe <- object@results[["statsframe"]]

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(statsframe) <- [toupper](https://rdrr.io/r/base/chartr.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(statsframe))

vresList <- [lapply](https://rdrr.io/r/base/lapply.html)([unlist](https://rdrr.io/r/base/unlist.html)(object@deeptable@Dimnames[[1]]),

[**function**](https://rdrr.io/r/base/function.html)(x){

vtemp <- coeffframe[coeffframe[,"MODELID"]==x,]

vrownames <- vtemp[["COEFFNAMES"]]

vtemp <- vtemp[,[c](https://rdrr.io/r/base/c.html)("COEFFVALUE","STDERR","TSTAT","PVALUE","NONZERODENSITY","CORRELWITHRES")]

vcoeffmat <- [as.matrix](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/as.matrix.html)(vtemp)

[rownames](https://rdrr.io/r/base/colnames.html)(vcoeffmat) <- vrownames

[colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vcoeffmat) <- [c](https://rdrr.io/r/base/c.html)("Estimate","Std.Err","t-stat",

"p-value","non-zero Density","Correl With Residual")

vtemp <- statsframe[statsframe[,"MODELID"]==x,]

vsummaryList <- [list](https://rdrr.io/r/base/list.html)(coeffframe=vcoeffmat,

statsframe=vtemp,

[call](https://rdrr.io/r/base/call.html)=object$[call](https://rdrr.io/r/base/call.html))

[class](https://rdrr.io/r/base/class.html)(vsummaryList) <- "summary.FLLinRegrMD"

[**return**](https://rdrr.io/r/base/function.html)(vsummaryList)

})

[names](https://rdrr.io/r/base/names.html)(vresList) <- [paste0](https://rdrr.io/r/base/paste.html)("Model",[unlist](https://rdrr.io/r/base/unlist.html)(object@deeptable@Dimnames[[1]]))

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)

([sys.call](https://rdrr.io/r/base/sys.parent.html)()),"(",fixed=T))[2],")",fixed=T))[1]

object@results[["statsframe"]] <- statsframe

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(vresList)

}

*#' @export*

[print.summary.FLLinRegrMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object){

ret <- object$statsframe

[cat](https://rdrr.io/r/base/cat.html)("Call:\n")

[cat](https://rdrr.io/r/base/cat.html)([paste0](https://rdrr.io/r/base/paste.html)(object$[call](https://rdrr.io/r/base/call.html)),"\n")

[cat](https://rdrr.io/r/base/cat.html)("\n\nCoefficients:\n")

[print](https://rdrr.io/r/base/print.html)(object$coeffframe)

[cat](https://rdrr.io/r/base/cat.html)("\n---\n")

[cat](https://rdrr.io/r/base/cat.html)("Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 '' 1\n")

[cat](https://rdrr.io/r/base/cat.html)("Residual standard error: ",ret[["MSRESIDUAL"]]," on ",ret[["DFRESIDUAL"]]," degrees of freedom\n\n")

[cat](https://rdrr.io/r/base/cat.html)("Multiple R-squared: ",ret[["RSQUARED"]]," , Adjusted R-squared: ",ret[["ADJRSQUARED"]],"\n")

FStatPVal<-[pf](https://rdrr.io/r/stats/Fdist.html)(ret[["FSTAT"]],ret[["DFREGRESSION"]],ret[["DFRESIDUAL"]],lower.tail=[**FALSE**](https://rdrr.io/r/base/logical.html))

[cat](https://rdrr.io/r/base/cat.html)("F-statistic: ",ret[["FSTAT"]]," on ",ret[["DFREGRESSION"]]," and ",ret[["DFRESIDUAL"]]

,"DF , p-value: ",FStatPVal,"\n")

[cat](https://rdrr.io/r/base/cat.html)("MSRegression: ",ret[["MSREGRESSION"]]," , SigFStat: ",ret[["SIGFSTAT"]],"\n")

[cat](https://rdrr.io/r/base/cat.html)("DWStat: ",ret[["DWSTAT"]]," , ResidualAutoCorrel: ",ret[["RESIDUALAUTOCORREL"]],"\n")

[cat](https://rdrr.io/r/base/cat.html)("BPStat: ",ret[["BPSTAT"]]," , SigBPStat: ",ret[["SIGBPSTAT"]],"\n")

}

*#' @export*

print.FLLinRegrMD <- [summary.FLLinRegrMD](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

*#' @export*

`[[.FLLinRegr`<-[**function**](https://rdrr.io/r/base/function.html)(object,property){

*#parentObject <- deparse(substitute(object))*

parentObject <- [unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([unlist](https://rdrr.io/r/base/unlist.html)([strsplit](https://rdrr.io/r/base/strsplit.html)([as.character](https://rdrr.io/r/base/character.html)([sys.call](https://rdrr.io/r/base/sys.parent.html)()),

"(",fixed=T))[2],",",fixed=T))[1]

vresult <- `$.FLLinRegr`(object=object,property=property)

[assign](https://rdrr.io/r/base/assign.html)(parentObject,object,envir=[parent.frame](https://rdrr.io/r/base/sys.parent.html)())

[**return**](https://rdrr.io/r/base/function.html)(vresult)

}

[setDefaultsRegrDataPrepSpecs](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(x,values){

x <- [as.list](https://rdrr.io/r/base/list.html)(x)

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) [c](https://rdrr.io/r/base/c.html)("depCol", "catToDummy","performNorm",

"performVarReduc","minStdDev",

"maxCorrel","makeDataSparse",

"excludeCols","classSpec")){

[**if**](https://rdrr.io/r/base/Control.html)(i %in% [names](https://rdrr.io/r/base/names.html)(values))

x[[i]] <- values

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([is.null](https://rdrr.io/r/base/NULL.html)(x[[i]])){

[**if**](https://rdrr.io/r/base/Control.html)(i %in% [c](https://rdrr.io/r/base/c.html)("maxCorrel","makeDataSparse"))

x[[i]] <- 1

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(i %in% [c](https://rdrr.io/r/base/c.html)("excludeCols","depCol",

"outDeepTableName","outObsIDCol",

"outVarIDCol","outValueCol",

"whereconditions"))

x[[i]] <- ""

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)(i %in% "classSpec")

x[[i]] <- [list](https://rdrr.io/r/base/list.html)()

[**else**](https://rdrr.io/r/base/Control.html) x[[i]] <- 0

}

}

x[["depCol"]] <- ""

x

}

[getFittedValuesLogRegrSQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,newdata,scoreTable){

[UseMethod](https://rdrr.io/r/base/UseMethod.html)("getFittedValuesLogRegrSQL",newdata)

}

[getFittedValuesLogRegrSQL.FLTable.Hadoop](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,newdata,scoreTable){

vobsid <- "ObsID"

vfcalls <- object@vfcalls

[getFLVectorTableFunctionQuerySQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(indexColumn=vobsid,

valueColumn=vfcalls["valcolnamescoretable"],

FromTable=scoreTable)

}

getFittedValuesLogRegrSQL.FLTableDeep.Hadoop <- [getFittedValuesLogRegrSQL.FLTable.Hadoop](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

[getFittedValuesLogRegrSQL.FLTable.TDAster](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,newdata,scoreTable){

vobsid <- "obsid"

vfcalls <- object@vfcalls

[getFLVectorTableFunctionQuerySQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(indexColumn=vobsid,

valueColumn=vfcalls["valcolnamescoretable"],

FromTable=scoreTable)

}

getFittedValuesLogRegrSQL.FLTableDeep.TDAster <- [getFittedValuesLogRegrSQL.FLTable.TDAster](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)

[getFittedValuesLogRegrSQL.default](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,newdata,scoreTable){

vobsid <- getObsIdSQLExpression(newdata)

vfcalls <- object@vfcalls

[getFLVectorTableFunctionQuerySQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(indexColumn=vobsid,

valueColumn=vfcalls["valcolnamescoretable"],

FromTable=scoreTable)

}

[getFittedValuesLogRegrSQL.FLpreparedData](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)(object,newdata,scoreTable){

vobsid <- newdata$deepx@select@variables$obs\_id\_colname

vfcalls <- object@vfcalls

[getFLVectorTableFunctionQuerySQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLGet.R)(indexColumn=vobsid,

valueColumn=vfcalls["valcolnamescoretable"],

FromTable=scoreTable)

}

*#' @export*

[summary.FLLinRegrSF](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)<-[**function**](https://rdrr.io/r/base/function.html)(object,modelid=1){

AnalysisID<-object@AnalysisID

statstablename<-object@vfcalls["statstablename"]

query<-[paste0](https://rdrr.io/r/base/paste.html)("Select \* from ",statstablename, " Where AnalysisID = ",

[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)," And modelid =",modelid)

x<-[sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),query)

coeff<-[sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),[paste0](https://rdrr.io/r/base/paste.html)("Select \* from ",object@vfcalls["coefftablename"],

" Where AnalysisID=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)," And modelid=coeffid"))

coeffframe <- [data.frame](https://rdrr.io/r/base/data.frame.html)([coefficients](https://rdrr.io/r/stats/coef.html)(object),

t\_stat=coeff$TSTAT,

p\_value=coeff$PVALUE)

reqList <- [list](https://rdrr.io/r/base/list.html)([call](https://rdrr.io/r/base/call.html) = [as.call](https://rdrr.io/r/base/call.html)(object@[formula](https://rdrr.io/r/stats/formula.html)),

[residuals](https://rdrr.io/r/stats/residuals.html) = [**NULL**](https://rdrr.io/r/base/NULL.html),

[coefficients](https://rdrr.io/r/stats/coef.html) = [as.matrix](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/as.matrix.html)(coeffframe),

[sigma](https://rdrr.io/r/stats/sigma.html) = x$STDERR,

[df](https://rdrr.io/r/stats/Fdist.html) = [as.vector](https://rdrr.io/r/base/vector.html)([c](https://rdrr.io/r/base/c.html)((x$DFREGRESSION + 1),x$DFRESIDUAL, (x$DFREGRESSION + 1))),

*#r.squared = x$RSQUARED,*

*#adj.r.squared = x$ADJRSQUARED,*

*#fstatistic = c(x$FSTAT, x$DFREGRESSION, x$DFRESIDUAL ),*

aliased = [**FALSE**](https://rdrr.io/r/base/logical.html)

)

[class](https://rdrr.io/r/base/class.html)(reqList) <- "summary.lm"

reqList

}

*#' @export*

`$.FLLinRegrSF`<-[**function**](https://rdrr.io/r/base/function.html)(object,property){

[**if**](https://rdrr.io/r/base/Control.html)(property=="coefficients")

[**return**](https://rdrr.io/r/base/function.html)([coefficients](https://rdrr.io/r/stats/coef.html)(object))

}

*#' @export*

[coefficients.FLLinRegrSF](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R)<-[**function**](https://rdrr.io/r/base/function.html)(object){

AnalysisID<-object@AnalysisID

coefftablename<-object@vfcalls["coefftablename"]

statstablename<-object@vfcalls["statstablename"]

query1<-[paste0](https://rdrr.io/r/base/paste.html)("Select a.modelid, a.coeffvalue From ",coefftablename,

" a Where AnalysisID= ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)," And a.modelid=a.coeffid ORDER BY 1")

query2<-[paste0](https://rdrr.io/r/base/paste.html)("Select a.modelid, a.coeffvalue From ",coefftablename,

" a Where AnalysisID= ",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)," And a.modelid!=a.coeffid ORDER BY 1")

query3<-[paste0](https://rdrr.io/r/base/paste.html)("Select a.ModelID, a.AdjRSquared, a.RSquared, a.StdErr, a.FStat from ",statstablename,

" a Where AnalysisID=",[fquote](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/utilities.R)(AnalysisID)," Order By 1")

a<-[sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),query1)

b<-[sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),query2)

[c](https://rdrr.io/r/base/c.html)<-[sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),query3)

ret<-[data.frame](https://rdrr.io/r/base/data.frame.html)(ModelID=a$MODELID,

Intercept=b$COEFFVALUE,

Coeff=a$COEFFVALUE,

AdjRSquared=[c](https://rdrr.io/r/base/c.html)$ADJRSQUARED,

RSquared=[c](https://rdrr.io/r/base/c.html)$RSQUARED,

StdErr=[c](https://rdrr.io/r/base/c.html)$STDERR,

FStat=[c](https://rdrr.io/r/base/c.html)$FSTAT)

[**if**](https://rdrr.io/r/base/Control.html)(![isDotFormula](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)(object@[formula](https://rdrr.io/r/stats/formula.html))) [rownames](https://rdrr.io/r/base/colnames.html)(ret)<-[setdiff](https://rdrr.io/r/base/sets.html)([all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html)),[all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))[1])

[**else**](https://rdrr.io/r/base/Control.html) [rownames](https://rdrr.io/r/base/colnames.html)(ret)<- [setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(object@[table](https://rdrr.io/r/base/table.html)),[all.vars](https://rdrr.io/r/base/allnames.html)(object@[formula](https://rdrr.io/r/stats/formula.html))[1])

[**return**](https://rdrr.io/r/base/function.html)([data.matrix](https://rdrr.io/r/base/data.matrix.html)(ret))

}

[getReferenceCategories](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLLinRegr.R) <- [**function**](https://rdrr.io/r/base/function.html)([data](https://rdrr.io/r/utils/data.html),pExcludeCols="",

classSpec=[list](https://rdrr.io/r/base/list.html)(),

[**...**](https://rdrr.io/r/base/dots.html)){

*## browser()*

vcolnames <- [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)([data](https://rdrr.io/r/utils/data.html))

unused\_cols <- [c](https://rdrr.io/r/base/c.html)(pExcludeCols,

getObsIdSQLExpression([data](https://rdrr.io/r/utils/data.html)),

getGroupIdSQLExpression([data](https://rdrr.io/r/utils/data.html)))

*## Detect factors and assign classSpec*

vfirstRow <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[limitRowsSQL](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/constructAbstractSQL.R)([paste0](https://rdrr.io/r/base/paste.html)("SELECT \* FROM (",

[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "),1))

vfactorCols <- [list](https://rdrr.io/r/base/list.html)()

*## apply(t,2,function(x){class(x[[1]])}) gives all character*

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) [setdiff](https://rdrr.io/r/base/sets.html)([colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vfirstRow),

[c](https://rdrr.io/r/base/c.html)(unused\_cols,[names](https://rdrr.io/r/base/names.html)(classSpec),

[list](https://rdrr.io/r/base/list.html)([**...**](https://rdrr.io/r/base/dots.html))[["doNotTransform"]],

"obs\_id\_colname",

"group\_id\_colname"))){

[**if**](https://rdrr.io/r/base/Control.html)([length](https://rdrr.io/r/base/length.html)(i)==0) [**break**](https://rdrr.io/r/base/Control.html);

[**if**](https://rdrr.io/r/base/Control.html)([is.factor](https://rdrr.io/r/base/factor.html)(vfirstRow[[i]])

|| [is.character](https://rdrr.io/r/base/character.html)(vfirstRow[[i]])

|| [is.logical](https://rdrr.io/r/base/logical.html)(vfirstRow[[i]])){

r<-[as.character](https://rdrr.io/r/base/character.html)(vfirstRow[[i]])

[names](https://rdrr.io/r/base/names.html)(r) <- i

vfactorCols <- [c](https://rdrr.io/r/base/c.html)(vfactorCols,r)

}

}

[**if**](https://rdrr.io/r/base/Control.html)([length](https://rdrr.io/r/base/length.html)(vfactorCols)>0){

[**if**](https://rdrr.io/r/base/Control.html)([is.ODBC](https://rdrr.io/github/Fuzzy-Logix/AdapteR/src/R/FLIs.R)())

vrefVars <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT ",

[paste0](https://rdrr.io/r/base/paste.html)("MIN(",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

") AS ",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

collapse=","),

" FROM (",[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "),

as.is=[**TRUE**](https://rdrr.io/r/base/logical.html))

[**else**](https://rdrr.io/r/base/Control.html) vrefVars <- [sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT ",

[paste0](https://rdrr.io/r/base/paste.html)("MIN(",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

") AS ",[names](https://rdrr.io/r/base/names.html)(vfactorCols),

collapse=","),

" FROM (",[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "))

vtempList <- [list](https://rdrr.io/r/base/list.html)()

vrefVarNames <- [names](https://rdrr.io/r/base/names.html)(vrefVars)

[**for**](https://rdrr.io/r/base/Control.html)(i [**in**](https://rdrr.io/r/base/Control.html) [colnames](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/colnames.html)(vrefVars)){

*## Remove variables with NA*

[**if**](https://rdrr.io/r/base/Control.html)([is.na](https://rdrr.io/r/base/NA.html)(vrefVars[[i]]))

vrefVarNames <- [setdiff](https://rdrr.io/r/base/sets.html)(vrefVarNames,

i)

[**else**](https://rdrr.io/r/base/Control.html) [**if**](https://rdrr.io/r/base/Control.html)([is.logical](https://rdrr.io/r/base/logical.html)(vrefVars[[i]]))

vtempList <- [c](https://rdrr.io/r/base/c.html)(vtempList,

[levels](https://rdrr.io/r/base/levels.html)([sqlQuery](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/sqlQuery.html)(getFLConnection(),

[paste0](https://rdrr.io/r/base/paste.html)("SELECT DISTINCT(",i,

") FROM(",[constructSelect](https://rdrr.io/github/Fuzzy-Logix/AdapteR/man/constructSelect.html)([data](https://rdrr.io/r/utils/data.html)),") a "))[[1]])[1])

[**else**](https://rdrr.io/r/base/Control.html) vtempList <- [c](https://rdrr.io/r/base/c.html)(vtempList,[as.character](https://rdrr.io/r/base/character.html)(vrefVars[[i]]))

}

[names](https://rdrr.io/r/base/names.html)(vtempList) <- vrefVarNames

[**return**](https://rdrr.io/r/base/function.html)([c](https://rdrr.io/r/base/c.html)(classSpec,vtempList))

}

[**return**](https://rdrr.io/r/base/function.html)(classSpec)

}

[setMethod](https://rdrr.io/r/methods/setMethod.html)("names",[signature](https://rdrr.io/r/methods/GenericFunctions.html)("FLLinRegr"),

[**function**](https://rdrr.io/r/base/function.html)(x) [c](https://rdrr.io/r/base/c.html)("coefficients",

"residuals",

"fitted.values",

"FLCoeffStdErr",

"FLCoeffTStat",

"FLCoeffPValue",

"FLCoeffNonZeroDensity",

"FLCoeffCorrelWithRes",

"s",

"call",

"FLLinRegrStats",

"df.residual",

"model",

"x",

"y",

"qr",

"rank",

"terms",

"xlevels",

"assign",

"formula",

"anova"

))

**SVM:**

function (x, y = NULL, scale = TRUE, type = NULL, kernel = "radial",

degree = 3, gamma = if (is.vector(x)) 1 else 1/ncol(x), coef0 = 0,

cost = 1, nu = 0.5, class.weights = NULL, cachesize = 40,

tolerance = 0.001, epsilon = 0.1, shrinking = TRUE, cross = 0,

probability = FALSE, fitted = TRUE, ..., subset, na.action = na.omit)

{

yorig <- y

if (inherits(x, "Matrix")) {

loadNamespace("SparseM")

loadNamespace("Matrix")

x <- as(x, "matrix.csr")

}

if (inherits(x, "simple\_triplet\_matrix")) {

loadNamespace("SparseM")

ind <- order(x$i, x$j)

x <- new("matrix.csr", ra = x$v[ind], ja = x$j[ind],

ia = as.integer(cumsum(c(1, tabulate(x$i[ind])))),

dimension = c(x$nrow, x$ncol))

}

if (sparse <- inherits(x, "matrix.csr"))

loadNamespace("SparseM")

if (is.null(degree))

stop(sQuote("degree"), " must not be NULL!")

if (is.null(gamma))

stop(sQuote("gamma"), " must not be NULL!")

if (is.null(coef0))

stop(sQuote("coef0"), " must not be NULL!")

if (is.null(cost))

stop(sQuote("cost"), " must not be NULL!")

if (is.null(nu))

stop(sQuote("nu"), " must not be NULL!")

if (is.null(epsilon))

stop(sQuote("epsilon"), " must not be NULL!")

if (is.null(tolerance))

stop(sQuote("tolerance"), " must not be NULL!")

xhold <- if (fitted)

x

else NULL

x.scale <- y.scale <- NULL

formula <- inherits(x, "svm.formula")

if (is.null(type))

type <- if (is.null(y))

"one-classification"

else if (is.factor(y))

"C-classification"

else "eps-regression"

type <- pmatch(type, c("C-classification", "nu-classification",

"one-classification", "eps-regression", "nu-regression"),

99) - 1

if (type > 10)

stop("wrong type specification!")

kernel <- pmatch(kernel, c("linear", "polynomial", "radial",

"sigmoid"), 99) - 1

if (kernel > 10)

stop("wrong kernel specification!")

nac <- attr(x, "na.action")

if (sparse) {

scale <- rep(FALSE, ncol(x))

if (!is.null(y))

na.fail(y)

x <- SparseM::t(SparseM::t(x))

}

else {

x <- as.matrix(x)

if (!formula) {

if (!missing(subset)) {

x <- x[subset, ]

y <- y[subset]

if (!is.null(xhold))

xhold <- as.matrix(xhold)[subset, ]

}

if (is.null(y))

x <- na.action(x)

else {

df <- na.action(data.frame(y, x))

y <- df[, 1]

x <- as.matrix(df[, -1], rownames.force = TRUE)

nac <- attr(x, "na.action") <- attr(y, "na.action") <- attr(df,

"na.action")

}

}

if (length(scale) == 1)

scale <- rep(scale, ncol(x))

if (any(scale)) {

co <- !apply(x[, scale, drop = FALSE], 2, var)

if (any(co)) {

warning(paste("Variable(s)", paste(sQuote(colnames(x[,

scale, drop = FALSE])[co]), sep = "", collapse = " and "),

"constant. Cannot scale data."))

scale <- rep(FALSE, ncol(x))

}

else {

xtmp <- scale(x[, scale])

x[, scale] <- xtmp

x.scale <- attributes(xtmp)[c("scaled:center",

"scaled:scale")]

if (is.numeric(y) && (type > 2)) {

yorig <- y

y <- scale(y)

y.scale <- attributes(y)[c("scaled:center",

"scaled:scale")]

y <- as.vector(y)

}

}

}

}

nr <- nrow(x)

if (cross > nr)

stop(sQuote("cross"), " cannot exceed the number of observations!")

ytmp <- y

attributes(ytmp) <- NULL

if (!is.vector(ytmp) && !is.factor(y) && type != 2)

stop("y must be a vector or a factor.")

if (type != 2 && length(y) != nr)

stop("x and y don't match.")

if (cachesize < 0.1)

cachesize <- 0.1

if (type > 2 && !is.numeric(y))

stop("Need numeric dependent variable for regression.")

lev <- NULL

weightlabels <- NULL

if (type == 2)

y <- rep(1, nr)

else if (is.factor(y)) {

lev <- levels(y)

y <- as.integer(y)

}

else {

if (type < 3) {

if (any(as.integer(y) != y))

stop("dependent variable has to be of factor or integer type for classification mode.")

y <- as.factor(y)

lev <- levels(y)

y <- as.integer(y)

}

else lev <- unique(y)

}

if (type < 3 && !is.null(class.weights)) {

if (is.character(class.weights) && class.weights == "inverse")

class.weights <- 1/table(y)

if (is.null(names(class.weights)))

stop("Weights have to be specified along with their according level names !")

weightlabels <- match(names(class.weights), lev)

if (any(is.na(weightlabels)))

stop("At least one level name is missing or misspelled.")

}

nclass <- 2

if (type < 2)

nclass <- length(lev)

if (type > 1 && length(class.weights) > 0) {

class.weights <- NULL

warning(sQuote("class.weights"), " are set to NULL for regression mode. For classification, use a \_factor\_ for ",

sQuote("y"), ", or specify the correct ", sQuote("type"),

" argument.")

}

err <- empty\_string <- paste(rep(" ", 255), collapse = "")

if (is.null(type))

stop("type argument must not be NULL!")

if (is.null(kernel))

stop("kernel argument must not be NULL!")

if (is.null(degree))

stop("degree argument must not be NULL!")

if (is.null(gamma))

stop("gamma argument must not be NULL!")

if (is.null(coef0))

stop("coef0 seed argument must not be NULL!")

if (is.null(cost))

stop("cost argument must not be NULL!")

if (is.null(nu))

stop("nu argument must not be NULL!")

if (is.null(cachesize))

stop("cachesize argument must not be NULL!")

if (is.null(tolerance))

stop("tolerance argument must not be NULL!")

if (is.null(epsilon))

stop("epsilon argument must not be NULL!")

if (is.null(shrinking))

stop("shrinking argument must not be NULL!")

if (is.null(cross))

stop("cross argument must not be NULL!")

if (is.null(sparse))

stop("sparse argument must not be NULL!")

if (is.null(probability))

stop("probability argument must not be NULL!")

cret <- .C(R\_svmtrain, as.double(if (sparse) x@ra else t(x)),

as.integer(nr), as.integer(ncol(x)), as.double(y), as.integer(if (sparse) x@ia else 0),

as.integer(if (sparse) x@ja else 0), as.integer(type),

as.integer(kernel), as.integer(degree), as.double(gamma),

as.double(coef0), as.double(cost), as.double(nu), as.integer(weightlabels),

as.double(class.weights), as.integer(length(class.weights)),

as.double(cachesize), as.double(tolerance), as.double(epsilon),

as.integer(shrinking), as.integer(cross), as.integer(sparse),

as.integer(probability), nclasses = integer(1), nr = integer(1),

index = integer(nr), labels = integer(nclass), nSV = integer(nclass),

rho = double(nclass \* (nclass - 1)/2), coefs = double(nr \*

(nclass - 1)), sigma = double(1), probA = double(nclass \*

(nclass - 1)/2), probB = double(nclass \* (nclass -

1)/2), cresults = double(cross), ctotal1 = double(1),

ctotal2 = double(1), error = err)

if (cret$error != empty\_string)

stop(paste(cret$error, "!", sep = ""))

cret$index <- cret$index[1:cret$nr]

ret <- list(call = match.call(), type = type, kernel = kernel,

cost = cost, degree = degree, gamma = gamma, coef0 = coef0,

nu = nu, epsilon = epsilon, sparse = sparse, scaled = scale,

x.scale = x.scale, y.scale = y.scale, nclasses = cret$nclasses,

levels = lev, tot.nSV = cret$nr, nSV = cret$nSV[1:cret$nclasses],

labels = cret$labels[1:cret$nclasses], SV = if (sparse) SparseM::t(SparseM::t(x[cret$index])) else t(t(x[cret$index,

, drop = FALSE])), index = cret$index, rho = cret$rho[1:(cret$nclasses \*

(cret$nclasses - 1)/2)], compprob = probability,

probA = if (!probability) NULL else cret$probA[1:(cret$nclasses \*

(cret$nclasses - 1)/2)], probB = if (!probability) NULL else cret$probB[1:(cret$nclasses \*

(cret$nclasses - 1)/2)], sigma = if (probability) cret$sigma else NULL,

coefs = if (cret$nr == 0) NULL else t(matrix(cret$coefs[1:((cret$nclasses -

1) \* cret$nr)], nrow = cret$nclasses - 1, byrow = TRUE)),

na.action = nac)

if (cross > 0)

if (type > 2) {

scale.factor <- if (any(scale))

crossprod(y.scale$"scaled:scale")

else 1

ret$MSE <- cret$cresults \* scale.factor

ret$tot.MSE <- cret$ctotal1 \* scale.factor

ret$scorrcoeff <- cret$ctotal2

}

else {

ret$accuracies <- cret$cresults

ret$tot.accuracy <- cret$ctotal1

}

class(ret) <- "svm"

if (fitted) {

ret$fitted <- na.action(predict(ret, xhold, decision.values = TRUE))

ret$decision.values <- attr(ret$fitted, "decision.values")

attr(ret$fitted, "decision.values") <- NULL

if (type > 1)

ret$residuals <- yorig - ret$fitted

}

ret

}

<bytecode: 0x000000001b900180>

<environment: namespace:e1071>